

致尊敬的顾客

关于产品目录等资料中的旧公司名称

NEC电子公司与株式会社瑞萨科技于2010年4月1日进行业务整合（合并），整合后的新公司暨“瑞萨电子公司”继承两家公司的所有业务。因此，本资料中虽还保留有旧公司名称等标识，但是并不妨碍本资料的有效性，敬请谅解。

瑞萨电子公司网址：<http://www.renesas.com>

2010年4月1日
瑞萨电子公司

【发行】瑞萨电子公司（<http://www.renesas.com>）

【业务咨询】<http://www.renesas.com/inquiry>

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
 - “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
 - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
 - “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

H8S/2218 群、H8S/2212 群

瑞萨 16 位单片机

H8S 族 / H8S/2200 系列

H8S/2218	HD64F2218
	HD64F2218U
	HD6432217
H8S/2212	HD64F2212
	HD64F2212U
	HD64F2211
	HD64F2211U
	HD6432211
	HD6432210

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (<http://www.renesas.com>)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
 - (1) artificial life support devices or systems
 - (2) surgical implantations
 - (3) healthcare intervention (e.g., excision, administration of medication, etc.)
 - (4) any other purposes that pose a direct threat to human lifeRenesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

注意

本文只是参考译文，前页所载英文版“Cautions”具有正式效力。

关于利用本资料时的注意事项

1. 本资料是为了让用户根据用途选择合适的本公司产品的参考资料，对于本资料中所记载的技术信息，并非意味着对本公司或者第三者的知识产权及其他权利做出保证或对实施权力进行的承诺。
2. 对于因使用本资料所记载的产品数据、图、表、程序、算法及其他应用电路例而引起的损害或者对第三者的知识产权及其他权利造成侵犯，本公司不承担任何责任。
3. 不能将本资料所记载的产品和技术用于大规模破坏性武器的开发等目的、军事目的或其他的军需用途方面。另外，在出口时必须遵守日本的《外汇及外国贸易法》及其他出口的相关法令并履行这些法令中规定的必要手续。
4. 本资料所记载的产品数据、图、表、程序、算法以及其他应用电路例等所有信息均为本资料发行时的内容，本公司有可能在未做事先通知的情况下，对本资料所记载的产品或者产品规格进行更改。所以在购买和使用本公司的半导体产品之前，请事先向本公司的营业窗口确认最新的信息并经常留意本公司通过公司主页（<http://www.renesas.com>）等公开的最新信息。
5. 对于本资料中所记载的信息，制作时我们尽力保证出版时的精确性，但不承担因本资料的叙述不当而致使顾客遭受损失等的任何相关责任。
6. 在使用本资料所记载的产品数据、图、表等所示的技术内容、程序、算法及其他应用电路例时，不仅要对所使用的技术信息进行单独评价，还要对整个系统进行充分的评价。请顾客自行负责，进行是否适用的判断。本公司对于是否适用不负任何责任。
7. 本资料中所记载的产品并非针对万一出现故障或是错误运行就会威胁到人的生命或给人体带来危害的机器、系统（如各种安全装置或者运输交通用的、医疗、燃烧控制、航天器械、核能、海底中继用的机器和系统等）而设计和制造的，特别是对于品质和可靠性要求极高的机器和系统等（将本公司指定用于汽车方面的产品用于汽车时除外）。如果要用于上述的目的，请务必事先向本公司的营业窗口咨询。另外，对于用于上述目的而造成的损失等，本公司概不负责。
8. 除上述第7项内容外，不能将本资料中记载的产品用于以下用途。如果用于以下用途而造成的损失，本公司概不负责。
 - 1) 生命维持装置。
 - 2) 植埋于人体使用的装置。
 - 3) 用于治疗（切除患处、给药等）的装置。
 - 4) 其他直接影响到人的生命的装置。
9. 在使用本资料所记载的产品时，对于最大额定值、工作电源电压的范围、放热特性、安装条件及其他条件请在本公司规定的保证范围内使用。如果超出了本公司规定的保证范围使用时，对于由此而造成的故障和出现的事故，本公司将不承担任何责任。
10. 本公司一直致力于提高产品的质量和可靠性，但一般来说，半导体产品总会以一定的概率发生故障、或者由于使用条件不同而出现错误运行等。为了避免因本公司的产品发生故障或者错误运行而导致人身事故和火灾或造成社会性的损失，希望客户能自行负责进行冗余设计、采取延烧对策及进行防止错误运行等的安全设计（包括硬件和软件两方面的设计）以及老化处理等，这是作为机器和系统的出厂保证。特别是单片机的软件，由于单独进行验证很困难，所以要求在顾客制造的最终的机器及系统上进行安全检验工作。
11. 如果把本资料所记载的产品从其载体设备上卸下，有可能造成婴儿误吞的危险。顾客在将本公司产品安装到顾客的设备上时，请顾客自行负责将本公司产品设置为不容易剥落的安全设计。如果从顾客的设备上剥落而造成事故时，本公司将不承担任何责任。
12. 在未得到本公司的事先书面认可时，不可将本资料的一部分或者全部转载或者复制。
13. 如果需要了解关于本资料的详细内容，或者有其他关心的问题，请向本公司的营业窗口咨询。

产品使用上的注意事项

1. NC管脚的处理

【注意】 NC管脚不连接任何电路。

NC (Non connection) 管脚不与任何内部电路连接，它是作为测试和降低噪声用的管脚。因此，在NC管脚上不连接任何电路。

2. 未使用输入管脚处理

【注意】 将未使用的输入管脚固定为高电平或低电平。

通常，CMOS产品的输入管脚为高阻抗输入管脚。如果将未使用的管脚置于开路状态，由于外围噪声的感应会产生中间电平，而在内部产生灌电流，引起故障的发生。

通过上拉或下拉输入将未使用的输入管脚固定为高电平或低电平。

3. 初始化前的处理

【注意】 第一次加载电源时，产品的状态不稳定。

从给所有电源管脚外加电压，到给复位管脚输入低电平为止，内部电路均为不确定状态，寄存器的设定、各管脚的输出状态均不定。请采用避免由此不定状态引起系统故障的策略进行系统设计。对于具有复位功能的产品，加载电源后首先要先执行复位操作。

4. 禁止访问未定义及保留地址

【注意】 禁止访问未定义及保留地址。

未定义及保留地址除了被用于将来的功能扩展外，也用于给测试寄存器所分配的地址。不要访问这些寄存器，如果这些寄存器被访问，将不能保证这些操作均被执行。

本手册的篇章结构

本手册包含以下条目：

1. 产品操作相关的一般注意事项
2. 本手册的篇章结构
3. 前言
4. 目录
5. 概要
6. 各功能模块的说明
 - CPU 及系统控制模块
 - 内部外围模块

各模块的功能结构描述针对不同的模块而会略有不同，但一般包含以下条目：
①特点、②输入 / 输出、③寄存器描述、④操作说明、⑤使用注意事项。

在设计采用本 LSI 的应用系统时，请事先充分确认注意事项。
请务必阅读各章节中有关说明的注意事项及各章节最后提示的注意事项。
(使用注意事项一般根据需要记载于本手册中。)

7. 寄存器一览表
8. 电特性
9. 附录
10. 此版本修订的主要内容 (仅适用于修订版)

修订目录是总结前版修订的主要内容，并没有记载所有的修订内容，有关详细内容请在
本手册的正文中查看。

11. 索引

前言

本 LSI 是以采用瑞萨科技原始体系结构的 H8S/2000 CPU 为核心，为集成系统构成所必需的外围功能的单片机。

H8S/2000CPU 为内部 32 位结构，具备 16 位×16 个的通用寄存器和以高速运行为目标的最佳简洁并适用的指令设置，可处理 16M 字节的线性地址空间。

它包括系统构成必需的外围功能，如：ROM、RAM、DMA 控制器（DMAC）的总线主控器、16 位定时器脉冲单元（TPU）、看门狗定时器（WDT）、实时时钟（RTC）、通用串行总线（USB）、串行通信接口（SCI）、A/D 转换器、I/O 端口等。内部 ROM 带有单一电源闪存（F-ZTAT™*）、掩模 ROM，可随时编程，迅速而灵活地适应规格流动性大的产品在量产初期到大规模量产过程中所发生的各种变化。

本手册所记载内容是关于本 LSI 的硬件。

【注】 * F-ZTAT（株）是瑞萨科技的商标。

对象 本手册在“应用系统设计中采用 H8S/2218 群及 H8S/2212 群”的用户为对象。
使用本手册的读者必须具备有关电路、逻辑电路及单片机的基础知识。

目的 本手册是以“让用户理解 H8S/2218 群、H8S/2212 群的硬件功能与电特性”为目的。关于指令执行的详细内容请参考“H8S/2600 系列、H8S/2000 系列软件手册”，敬请一并阅读。

阅读方法

- 希望了解全部芯片的功能时：
→请按照目录的顺序阅读。
本手册按照大分类，由 CPU、系统控制功能、外围功能、电特性等分类顺序构成。
- 希望了解 CPU 功能的详细内容时
→请参考另册《H8S/2600 系列、H8S/2000 系列软件手册》。
- 如果知道寄存器名称，而希望了解详细功能时
→本书后面附有“索引”。请从索引检索页码。
有关地址、位内容及初始化值，汇总在《第 21 章 寄存器一览》。

凡例 寄存器名称：16 位定时器脉冲单元、串行通信接口、多个通道带有相同或类似的功能时，将采用以下的标记方法 XXX_N（XXX 为基本寄存器名称、N 为通道编号）。

位标记顺序：左侧为高位、右侧为低位。

数字的标记：2 进制为 B'xxxx、16 进制为 H'xxxx。

信号的标记：低电平有效的信号带有上划线 xxxx。

相关资料一览表 在瑞萨公司网页刊登有最新的资料。请确认您索取的资料是否为最新版本。

(<http://www.renesas.com/>)

- H8S/2218 群、H8S/2212 群相关用户手册

资料名称	资料编号
H8S/2218 群、H8S/2212 群硬件手册	本手册
H8S/2600 系列、H8S/2000 系列软件手册	RJJ09B0143

- 开发工具相关用户手册

资料名称	资料编号
H8S、H8/300 系列 C/C++ 编译程序、汇编程序、优化连接编辑编译包用户手册	ADJ - 702 - 303
H8S、H8/300 系列仿真器 . 调试器 (Windows 版) 用户手册	ADJ - 702 - 355
H8S、H8/300 系列仿真器 . 调试器 (UNIX 版) 用户手册	ADJ - 702 - 163
H8S、H8/300 系列 High-performance Embedded Workshop、 High-performance Debugging Interface 指南	ADJ - 702 - 307
High-performance Embedded Workshop 用户手册	ADJ - 702 - 275

- 应用手册

资料名称	资料编号
H8S、H8/300 系列 C/C++ 编译程序包应用手册	ADJ - 502 - 051
F-ZTAT 单片机技术 Q & A	ADJ - 502 - 055

目 录

1. 概要	1
1.1 特点	1
1.2 内部框图	3
1.3 管脚排列图	7
1.4 运行模式其他管脚功能一览	15
1.5 管脚功能	20
2. CPU	27
2.1 特点	27
2.1.1 H8S/2600 CPU 与 H8S/2000 CPU 的区别	28
2.1.2 与 H8/300 CPU 的区别	28
2.1.3 与 H8/300H CPU 的区别	29
2.2 CPU 运行模式	29
2.2.1 普通模式	29
2.2.2 高级模式	31
2.3 地址空间	33
2.4 寄存器结构	34
2.4.1 通用寄存器	35
2.4.2 程序计数器 (PC)	36
2.4.3 扩展控制寄存器 (EXR)	36
2.4.4 条件码寄存器 (CCR)	37
2.4.5 CPU 内部寄存器初始值	38
2.5 数据格式	38
2.5.1 通用寄存器的数据格式	38
2.5.2 存储器的数据格式	40
2.6 指令系统	41
2.6.1 指令功能一览	42
2.6.2 基本指令格式	49
2.7 寻址模式和有效地址的计算方法	50
2.7.1 寄存器直接寻址 Rn	50
2.7.2 寄存器间接寻址 @ERn	50
2.7.3 带位移量的寄存器间接寻址 @ (d:16,ERn) / @ (d:32,ERn)	50
2.7.4 后增寄存器间接寻址 @ERn+ / 先减寄存器间接寻址 @-ERn	50
2.7.4.1 后增寄存器间接寻址 @ERn+	50
2.7.4.2 先减寄存器间接寻址 @-ERn	50
2.7.5 绝对地址 @aa:8 / @aa:16 / @aa:24 / @aa:32	51
2.7.6 立即数寻址 #xx:8 / #xx:16 / #xx:32	51
2.7.7 程序计数器相对寻址 @ (d:8,PC) / @ (d:16,PC)	51
2.7.8 存储器间接寻址 @@aa:8	52
2.7.9 有效地址的计算方法	53
2.8 处理状态	54
2.9 使用注意事项	55
2.9.1 TAS 操作指令使用注意事项	55
2.9.2 STM/LDM 操作指令使用注意事项	56
2.9.3 位操作指令使用注意事项	56
2.9.4 包括写专用位的寄存器存取方法	58
3. MCU 运行模式	60
3.1 工作模式的选择	60

3.2	寄存器说明	61
3.2.1	模式控制寄存器 (MDCR)	61
3.2.2	系统控制寄存器 (SYSCR)	62
3.3	各工作模式的说明	63
3.3.1	模式 4 (仅限于 H8S/2218 群)	63
3.3.2	模式 5 (仅限于 H8S/2218 群)	63
3.3.3	模式 6 (仅限于 H8S/2218 群)	63
3.3.4	模式 7	63
3.3.5	管脚功能	64
3.4	各工作模式的地址映射	65
4.	异常处理	68
4.1	异常处理种类和优先级	68
4.2	异常源与向量表	69
4.3	复位	70
4.3.1	复位种类	70
4.3.2	复位异常处理	71
4.3.3	复位后的中断	72
4.3.4	复位解除后的内置外围功能	72
4.4	跟踪异常处理	73
4.5	中断异常处理	73
4.6	陷阱指令异常处理	74
4.7	异常处理后的堆栈状态	75
4.8	使用注意事项	76
5.	中断控制器	77
5.1	特点	77
5.2	输入 / 输出管脚	78
5.3	寄存器的说明	78
5.3.1	中断优先级寄存器 A ~ G、J、K、M (IPRA ~ IPRG、IPRJ、IPRK、IPRM)	79
5.3.2	IRQ 允许寄存器 (IER)	80
5.3.3	IRQ 感知控制寄存器 H、L (ISCRH、ISCRL)	81
5.3.4	IRQ 状态寄存器 (ISR)	82
5.4	中断源	82
5.4.1	外部中断源	82
5.4.1.1	NMI 中断	82
5.4.1.2	IRQ7 ~ IRQ0 中断	83
5.4.2	内部中断	83
5.5	中断异常处理向量表	84
5.6	中断控制模式与中断运行	85
5.6.1	中断控制模式 0	85
5.6.2	中断控制模式 2	87
5.6.3	中断异常处理顺序	89
5.6.4	中断响应时间	90
5.6.5	通过中断启动 DMAC	91
5.6.5.1	中断源的选择	91
5.6.5.2	优先级判别	91
5.6.5.3	运行顺序	92
5.6.5.4	使用注意事项	92
5.7	使用注意事项	93
5.7.1	中断发生与禁止的竞争	93
5.7.2	禁止中断的指令	93

5.7.3	中断屏蔽期间	93
5.7.4	EEPMOV 指令执行中的中断	93
5.7.5	IRQ 中断	94
5.7.6	NMI 中断使用时的注意事项	94
6.	总线控制器 (BSC)	95
6.1	特点	95
6.2	输入 / 输出管脚	97
6.3	寄存器的说明	98
6.3.1	总线宽度控制寄存器 (ABWCR)	98
6.3.2	存取状态控制寄存器 (ASTCR)	98
6.3.3	等待控制寄存器 H、L (WCRH、WCRL)	99
6.3.4	总线控制寄存器 H (BCRH)	101
6.3.5	总线控制寄存器 L (BCRL)	102
6.3.6	管脚功能控制寄存器 (PFCR)	103
6.4	总线控制	104
6.4.1	区域分割	104
6.4.2	总线规格	105
6.4.2.1	总线宽度	105
6.4.2.2	存取状态数	105
6.4.2.3	程序等待状态数	105
6.4.3	各区域的总线接口	106
6.4.3.1	区域 0	106
6.4.3.2	区域 1 ~ 6	106
6.4.3.3	区域 7	106
6.4.4	片选信号	106
6.5	基本运行时序	107
6.5.1	内部存储器 (ROM、RAM) 存取时序	107
6.5.2	内部外围模块存取时序	108
6.5.3	外部地址空间存取时序	108
6.6	基本总线接口	109
6.6.1	数据长度与数据对准 (仅限于 H8S/2218 群使用)	109
6.6.1.1	8 位存取空间	109
6.6.1.2	16 位存取空间	109
6.6.2	有效选通	110
6.6.3	基本时序	111
6.6.3.1	8 位 2 态存取空间	111
6.6.3.2	8 位 3 态存取空间 (区域 6 以外)	112
6.6.3.3	8 位 3 态存取空间 (区域 6 及 RTC)	113
6.6.3.4	16 位 2 态存取空间	114
6.6.3.5	16 位 3 态存取空间	117
6.6.4	等待控制	120
6.6.4.1	程序等待的插入	120
6.6.4.2	管脚等待的插入	120
6.7	突发式 ROM 接口	121
6.7.1	基本时序	121
6.7.2	等待控制	122
6.8	空闲周期	123
6.8.1	不同区域间的连续读取	123
6.8.2	读取后的写入	124
6.8.3	片选 (\overline{CS}) 信号与读取 (\overline{RD}) 信号的关系	124
6.9	总线释放	126

6.9.1	使用总线权释放时的注意事项	127
6.10	总线仲裁	128
6.10.1	运行说明	128
6.10.2	总线权转移时序	128
6.10.2.1	CPU	128
6.10.2.2	DMAC	128
6.10.3	使用外部总线权释放注意事项	128
6.11	复位与总线控制器	129
7.	DMA 控制器 (DMAC)	130
7.1	特点	130
7.1.1	短地址模式	130
7.1.2	全地址模式	130
7.2	寄存器说明	132
7.3	各寄存器的说明	134
7.3.1	存储器地址寄存器 (MAR)	134
7.3.1.1	短地址模式	134
7.3.1.2	全地址模式	134
7.3.2	MI/O 地址寄存器 (IOAR)	134
7.3.2.1	短地址模式	134
7.3.2.2	全地址模式	134
7.3.3	传送计数寄存器 (ETCR)	135
7.3.3.1	短地址模式	135
7.3.3.2	全地址模式	135
7.3.4	DMA 控制寄存器 (DMACR)	136
7.3.5	DMA 带控制寄存器 (DMABCR)	140
7.4	工作说明	145
7.4.1	传送模式	145
7.4.2	顺序模式	145
7.4.3	地址模式	148
7.4.4	重复模式	150
7.4.5	正常模式	153
7.4.6	块传送模式	156
7.4.7	DMAC 启动源	161
7.4.7.1	通过内部中断激活	161
7.4.7.2	通过 USB 请求激活	161
7.4.7.3	通过自动请求激活	161
7.4.8	DMAC 基本总线周期	162
7.4.9	DMAC 总线周期 (双地址模式)	163
7.4.9.1	短地址模式	163
7.4.9.2	自由地址模式 (周期挪用模式)	164
7.4.9.3	全地址模式 (突发模式)	165
7.4.9.4	全地址模式 (块传送模式)	166
7.4.9.5	DREQ 电平激活时序 (正常模式)	167
7.4.10	DMAC 多个通道的工作	168
7.4.11	DMAC 和外部总线权请求的关系	168
7.4.12	NMI 中断和 DMAC	169
7.4.13	DMAC 工作的强制终止	169
7.4.14	自由地址模式解除	170
7.5	中断源	171
7.6	使用时的注意事项	172
7.6.1	工作中 DMAC 寄存器存取	172

7.6.2	模块停止	173
7.6.3	中速模式	173
7.6.4	启动源的接受	173
7.6.5	传送结束后的内部中断	173
7.6.6	通道的再次设定	173
8.	I/O 端口	174
8.1	端口 1	178
8.1.1	端口 1 数据方向寄存器 (P1DDR)	178
8.1.2	端口 1 数据寄存器 (P1DR)	178
8.1.3	端口 1 寄存器 (PORT1)	179
8.1.4	管脚功能	179
8.1.4.1	H8S/2218 群的管脚功能	179
8.1.4.2	H8S/2212 群的管脚功能	181
8.2	端口 3	184
8.2.1	端口 3 数据方向寄存器 (P3DDR)	184
8.2.2	端口 3 数据寄存器 (P3DR)	184
8.2.3	端口 3 寄存器 (PORT3)	185
8.2.4	端口 3 漏极开路控制寄存器 (P3ODR)	185
8.2.5	管脚功能	186
8.3	端口 4	187
8.3.1	端口 4 寄存器 (PORT4)	187
8.3.2	管脚功能	187
8.4	端口 7	188
8.4.1	端口 7 数据方向寄存器 (P7DDR)	188
8.4.2	端口 7 数据寄存器 (P7DR)	189
8.4.3	端口 7 寄存器 (PORT7)	190
8.4.4	管脚功能	191
8.4.4.1	H8S/2218 群的管脚功能	191
8.4.4.2	H8S/2212 群的管脚功能	191
8.5	端口 9	192
8.5.1	端口 9 寄存器 (PORT9)	192
8.5.2	管脚功能	192
8.6	端口 A	193
8.6.1	端口 A 数据方向寄存器 (PADDR)	193
8.6.2	端口 A 数据寄存器 (PADR)	193
8.6.3	端口 A 寄存器 (PORTA)	194
8.6.4	端口 A 上拉 MOS 控制寄存器 (PAPCR)	194
8.6.5	端口 A 漏极开路控制寄存器 (PAODR)	194
8.6.6	管脚功能	195
8.6.6.1	H8S/2218 群的管脚功能	195
8.6.6.2	H8S/2212 群的管脚功能	196
8.6.7	端口 A 输入上拉 MOS 的状态	197
8.7	端口 B (仅 H8S/2218 群)	198
8.7.1	端口 B 数据方向寄存器 (PBDDR)	198
8.7.2	端口 B 数据寄存器 (PBDR)	198
8.7.3	端口 B 寄存器 (PORTB)	199
8.7.4	端口 B 上拉 MOS 控制寄存器 (PBPCR)	199
8.7.5	管脚功能	200
8.7.6	端口 B 输入上拉 MOS 的状态	201
8.8	端口 C (仅 H8S/2218 群)	202
8.8.1	端口 C 数据方向寄存器 (PCDDR)	202

8.8.2	端口 C 数据寄存器 (PCDR)	202
8.8.3	端口 C 寄存器 (PORTC)	203
8.8.4	端口 C 上拉 MOS 控制寄存器 (PCPCR)	203
8.8.5	端口功能	204
8.8.6	端口 C 输入上拉 MOS 的状态	205
8.9	端口 D (仅 H8S/2218 群)	206
8.9.1	端口 D 数据方向寄存器 (PDDDR)	206
8.9.2	端口 D 数据寄存器 (PDDR)	206
8.9.3	端口 D 寄存器 (PORTD)	207
8.9.4	端口 D 上拉 MOS 控制寄存器 (PDPCR)	207
8.9.5	管脚功能	208
8.9.6	端口 D 输入上拉 MOS 的状态	209
8.10	端口 E	210
8.10.1	端口 E 数据方向寄存器 (PEDDR)	210
8.10.2	端口 E 数据寄存器 (PEDR)	210
8.10.3	端口 E 寄存器 (PORTE)	211
8.10.4	端口 E 上拉 MOS 控制寄存器 (PEPCR)	211
8.10.5	管脚功能	212
8.10.5.1	H8S/2218 群的管脚功能	212
8.10.5.2	H8S/2212 群的管脚功能	213
8.10.6	端口 E 输入上拉 MOS 的状态	214
8.11	端口 F	215
8.11.1	端口 F 数据方向寄存器 (PFDDR)	215
8.11.2	端口 F 数据寄存器 (PFDR)	215
8.11.3	端口 F 寄存器 (PORTF)	216
8.11.4	时钟输出控制寄存器 (OUTCR)	216
8.11.5	管脚功能	217
8.11.5.1	H8S/2218 群的管脚功能	217
8.11.5.2	H8S/2212 群的管脚功能	219
8.12	端口 G	220
8.12.1	端口 G 数据方向寄存器 (PGDDR)	220
8.12.2	端口 G 数据寄存器 (PGDR)	221
8.12.3	端口 G 寄存器 (PORTG)	221
8.12.4	管脚功能	222
8.12.4.1	H8S/2218 群的管脚功能	222
8.12.4.2	H8S/2212 群的管脚功能	223
8.13	未使用管脚的处理	223
9.	16 位定时器脉冲单元 (TPU)	224
9.1	特点	224
9.2	输入 / 输出管脚	226
9.3	寄存器的说明	227
9.3.1	定时器控制寄存器 (TCR)	228
9.3.2	定时器模式寄存器 (TMDR)	231
9.3.3	定时 I/O 控制寄存器 (TIOR)	232
9.3.4	定时器中断允许寄存器 (TIER)	241
9.3.5	定时状态寄存器 (TSR)	242
9.3.6	定时器寄存器 (TCNT)	244
9.3.7	定时器通用寄存器 (TGR)	244
9.3.8	定时器起始寄存器 (TSTR)	244
9.3.9	定时器单寄存器 (TSYR)	244
9.4	总线主控器接口	245

9.4.1	16 位寄存器	245
9.4.2	8 位寄存器	245
9.5	工作说明	247
9.5.1	基本工作	247
9.5.1.1	计数器操作	247
9.5.1.2	用比较匹配输出波形功能	249
9.5.1.3	输入捕捉功能	250
9.5.2	同步工作	252
9.5.2.1	同步操作的设定顺序例子	252
9.5.2.2	同步操作的例子	253
9.5.3	缓冲运行	253
9.5.3.1	缓冲操作的设定顺序例子	254
9.5.3.2	缓冲操作例子	255
9.5.4	PWM 模式	257
9.5.4.1	PWM 模式的设定顺序例子	258
9.5.4.2	PWM 模式操作例子	258
9.5.5	相位计数模式	261
9.5.5.1	相位计数模式的设定顺序例子	261
9.5.5.2	相位计数模式操作例子	262
9.6	中段源	266
9.6.1	中段源和优先顺序	266
9.6.1.1	输入捕捉 / 比较匹配中断	266
9.6.1.2	上溢中断	266
9.6.1.3	下溢中断	266
9.6.2	DMAC 的启动	267
9.6.3	A/D 转换器的启动	267
9.7	工作时序	268
9.7.1	输入 / 输出时序	268
9.7.1.1	TCNT 的计数时序	268
9.7.1.2	输出比较输出时序	269
9.7.1.3	输入捕捉信号时序	269
9.7.1.4	比较匹配 / 输入捕捉的计数器清除时序	270
9.7.1.5	缓冲运行时序	271
9.7.2	中断信号时序	272
9.7.2.1	比较匹配时 TGF 标志置位时序	272
9.7.2.2	输入捕捉时 TGF 标志置位时序	272
9.7.2.3	TCFV 标志 / TCFU 标志的置位时序	273
9.7.2.4	状态标志清除时序	274
9.8	使用时的注意事项	275
9.8.1	输入时钟的限制事项	275
9.8.2	周期设定的注意事项	275
9.8.3	TCNT 的写与清除的竞争	276
9.8.4	TCNT 的写与累加计数的竞争	276
9.8.5	TGR 的写入与比较匹配的竞争	277
9.8.6	缓冲寄存器的写与比较匹配的竞争	277
9.8.7	TGR 的读与输入捕捉的竞争	278
9.8.8	TGR 的写与输入捕捉的竞争	278
9.8.9	缓冲寄存器的写与输入捕捉的竞争	279
9.8.10	上溢 / 下溢与计数器清除的竞争	279
9.8.11	TCNT 的写与上溢 / 下溢的竞争	280
9.8.12	输入 / 输出管脚的兼用	280
9.8.13	模块停止时的中断	280

9.8.14	模块停止模式的设定	280
10.	看门狗定时器 (WDT)	281
10.1	特点	281
10.2	寄存器的说明	282
10.2.1	定时计数器 (TCNT)	282
10.2.2	定时器控制 / 状态寄存器 (TCSR)	283
10.2.3	置位控制 / 状态寄存器 (RSTCSR)	284
10.3	工作说明	285
10.3.1	看门狗定时器模式	285
10.3.2	看门狗定时器上溢标志 (WOVF) 的置位时序	286
10.3.3	间隔定时器模式	286
10.3.4	上溢标志 (OVF) 的置位时序	287
10.4	中断源	287
10.5	使用时的注意事项	288
10.5.1	寄存器存取时的注意事项	288
10.5.1.1	写入 TCNT、TCSR	288
10.5.1.2	写入 RSTCSR	288
10.5.1.3	从 TCNT、TCSR、RSTCSR 读	288
10.5.2	定时器计数器 (TCNT) 的写和累加计数的竞争	289
10.5.3	CKS2 ~ CKS0 位的改写	289
10.5.4	看门狗定时器模式和间接定时器模式的切换	289
10.5.5	看门狗定时器模式的内部复位	289
10.5.6	间隔定时器模式下的 OVF 标志清除	289
11.	实时时钟 (RTC)	290
11.1	特点	290
11.2	输入输出管脚	290
11.3	寄存器的说明	291
11.3.1	秒数据寄存器 (RSECDR)	291
11.3.2	分数据寄存器 (RMINDR)	292
11.3.3	时数据寄存器 (RHRDR)	292
11.3.4	星期数据寄存器 (RWKDR)	293
11.3.5	RTC 控制寄存器 1 (RTCCR1)	294
11.3.6	RTC 控制寄存器 2 (RTCCR2)	295
11.3.7	时钟源选择寄存器 (RTCCSR)	296
11.3.8	扩展模块待机寄存器 (EXMDLSTP)	296
11.4	RTC 的运行	297
11.4.1	加电后寄存器的初始设定及再设定顺序	297
11.4.2	读取时刻顺序	298
11.5	中断源	299
11.6	各模式下的运行状态	300
11.7	使用注意事项	300
11.7.1	使用仿真器时的注意事项	300
11.7.2	关于总线接口	300
11.7.3	通过开发板 D 寄存器 (PORTD) 读取管脚状态时的注意事项	300
12.	串行通信接口 (SCI)	301
12.1	特点	301
12.1.1	框图	302
12.2	输入 / 输出管脚	303
12.3	寄存器说明	304

12.3.1	接收移位寄存器 (RSR)	304
12.3.2	接收数据寄存器 (RDR)	304
12.3.3	发送数据寄存器 (TDR)	304
12.3.4	发送移位寄存器 (TSR)	304
12.3.5	串行通信模式寄存器 (SMR)	305
12.3.6	串行通信控制寄存器 (SCR)	308
12.3.7	串行通信状态寄存器 (SSR)	311
12.3.8	智能卡模式寄存器 (SCMR)	315
12.3.9	串行扩展模式寄存器 A_0 (SEMRA_0)	316
12.3.10	串行扩展模式寄存器 B_0 (SEMRB_0)	318
12.3.11	比特率寄存器 (BRR)	326
12.4	异步模式的运行	333
12.4.1	通信格式	334
12.4.2	异步模式下接收数据的采样时序和接收容限	335
12.4.3	时钟	336
12.4.4	SCI 的初始化 (异步模式)	337
12.4.5	串行数据发送 (异步模式)	338
12.4.6	串行数据接收 (异步模式)	340
12.5	多处理器通信功能	343
12.5.1	多处理器模式串行数据的发送	344
12.5.2	多处理器模式串行数据的接收	345
12.6	时钟同步模式的运行	348
12.6.1	时钟	348
12.6.2	SCI 的初始化 (时钟同步模式)	349
12.6.3	串行数据的发送 (时钟同步模式)	350
12.6.4	接收串行数据 (时钟同步模式)	352
12.6.5	发送 / 接收串行数据同时运行 (时钟同步模式)	354
12.7	智能卡接口操作说明	355
12.7.1	管脚连接图例	355
12.7.2	数据格式 (除数据块传送模式之外)	356
12.7.3	时钟	357
12.7.4	块传送模式	357
12.7.5	接收数据采样时序接收容限	358
12.7.6	初始设定	359
12.7.7	发送串行数据 (除数据块传送模式之外)	359
12.7.8	串行数据的接收 (除数据块传送模式之外)	362
12.7.9	时钟输出控制	364
12.7.9.1	加载电源时	364
12.7.9.2	从智能卡接口模式转换到软件待机模式时	364
12.7.9.3	从软件待机模式返回到智能卡接口模式时	365
12.8	SCI 选择功能 (时钟同步模式)	366
12.9	中断源	368
12.9.1	串行通信接口的中断	368
12.9.2	智能卡接口模式下的中断	368
12.10	使用注意事项	369
12.10.1	模块停止模式的设定	369
12.10.2	关于中止信号的检测和处理 (仅限异步模式)	369
12.10.3	MARK 状态和中止检测 (仅限异步模式)	369
12.10.4	接收错误标志和发送运行 (仅对时钟同步模式有效)	369
12.10.5	DMAC 使用注意事项	369
12.10.6	模式迁移	370
12.10.6.1	发送	370

12.10.6.2 接收	371
12.10.7 从 SCK 管脚向端口管脚切换注意事项	372
13. 边界扫描	374
13.1 特点	374
13.2 输入输出管脚	375
13.3 寄存器说明	376
13.3.1 指令寄存器 (INSTR)	376
13.3.1.1 EXTEST	376
13.3.1.2 SAMPLE/PRELOAD	377
13.3.1.3 CLAMP	377
13.3.1.4 HIGHZ	377
13.3.1.5 IDCODE	377
13.3.1.6 BYPASS	377
13.3.2 IDCODE 寄存器 (IDCODE)	377
13.3.3 BYPASS 寄存器 (BYPASS)	378
13.3.4 边界扫描寄存器 (BSCANR)	378
13.4 运行说明	386
13.4.1 TAP 控制器	386
13.5 使用注意事项	387
14. 通用串行总线 (USB)	389
14.1 特点	389
14.2 输入输出管脚	391
14.3 寄存器说明	392
14.3.1 USB 控制寄存器 (UCTLR)	393
14.3.2 USB DMAC 传送请求寄存器 (UDMAR)	394
14.3.3 USB 设备恢复寄存器 (UDRR)	395
14.3.4 USB 触发寄存器 0 (UTRG0)	396
14.3.5 USB FIFO 清除寄存器 0 (UFCLR0)	397
14.3.6 USB 终点挂起寄存器 0 (UESTL0)	398
14.3.7 USB 终点挂起寄存器 1 (UESTL1)	398
14.3.8 USB 终点数据寄存器 0s (UEDR0s)	399
14.3.9 USB 终点数据寄存器 0i (UEDR0i)	399
14.3.10 USB 终点数据寄存器 0o (UEDR0o)	399
14.3.11 USB 终点数据寄存器 3 (UEDR3)	399
14.3.12 USB 终点数据寄存器 1 (UEDR1)	400
14.3.13 USB 终点数据寄存器 2 (UEDR2)	400
14.3.14 USB 终点接收数据大小寄存器 0o (UESZ0o)	400
14.3.15 USB 终点接收数据长度寄存器 2 (UESZ2)	400
14.3.16 USB 中断标志寄存器 0 (UIFR0)	401
14.3.17 USB 中断标志寄存器 1 (UIFR1)	402
14.3.18 USB 中断标志寄存器 3 (UIFR3)	403
14.3.19 USB 中断允许寄存器 0 (UIER0)	404
14.3.20 USB 中断允许寄存器 1 (UIER1)	404
14.3.21 USB 中断允许寄存器 3 (UIER3)	405
14.3.22 USB 中断选择寄存器 0 (UISR0)	405
14.3.23 USB 中断选择寄存器 1 (UISR1)	406
14.3.24 USB 中断选择寄存器 3 (UISR3)	406
14.3.25 USB 数据状态寄存器 (UDSR)	407
14.3.26 USB 配置值寄存器 (UCVR)	407
14.3.27 USB 测试寄存器 0 (UTSTR0)	408

14.3.28	USB 测试寄存器 1 (UTSTR1)	409
14.3.29	USB 测试寄存器 2、A ~ F (UTSTR2、UTSTRA ~ UTSTRF)	409
14.3.30	模块暂停控制寄存器 B (MSTPCRB)	410
14.3.31	扩展模块待机寄存器 (EXMDLSTP)	410
14.4	中断源	411
14.5	通信运行说明	413
14.5.1	初始设定	413
14.5.2	USB 线的连接 / 断开	414
14.5.2.1	USB 线连接时 (没有使用 USB 模块停止、各种低功耗模式时)	414
14.5.2.2	USB 线连接时 (使用 USB 模块停止、各种低功耗模式时)	415
14.5.2.3	USB 线断开时 (没有使用 USB 模块停止、各种低功耗模式时)	416
14.5.2.4	USB 线断开时 (使用 USB 模块停止、各种低功耗模式时)	417
14.5.3	挂起 / 恢复	418
14.5.3.1	挂起 / 恢复时	418
14.5.3.2	挂起 / 恢复中断处理	419
14.5.3.3	挂起 / 远程唤醒处理	420
14.5.3.4	远程唤醒中断处理	421
14.5.4	控制传送	422
14.5.4.1	设置阶段	422
14.5.4.2	数据阶段 (control in 时)	423
14.5.4.3	数据阶段 (control out 时)	424
14.5.4.4	状态阶段 (control in 时)	425
14.5.4.5	状态阶段 (control out 时)	426
14.5.5	中断 in 传送 — 终点 3 —	427
14.5.6	Bulk in 传送 (2 面 FIFO) — 终点 1 —	428
14.5.7	Bulk out 传送 (2 面 FIFO) — 终点 2 —	429
14.5.8	USB 标准指令和 class/vender 指令的处理	430
14.5.8.1	通过控制传送将发送来的指令进行处理	430
14.5.9	挂起操作	430
14.5.9.1	概要	430
14.5.9.2	固件需要强制挂起时	431
14.5.9.3	USB 功能模块自动被挂起时	433
14.6	DMA 传送说明	434
14.6.1	通过 USB 请求的 DMA 传送	434
14.6.1.1	概要	434
14.6.1.2	内置 DMAC 的设定	434
14.6.1.3	关于 EP1 的 DMA 传送	434
14.6.1.4	关于 EP2 的 DMA 传送	435
14.6.2	通过自动请求的 DMA 传送	435
14.6.2.1	概要	435
14.6.2.2	内部 DMAC 的设定	436
14.6.2.3	EP0i、EP1、EP3 的 DMA 传送	436
14.6.2.4	EP0o、EP2 的 DMA 传送	437
14.7	USB 外部电路	438
14.8	使用注意事项	439
14.8.1	使用仿真器注意事项	439
14.8.2	关于总线接口	439
14.8.3	关于运行频率	439
14.8.4	接收设置数据	439
14.8.5	FIFO 的清除	439
14.8.6	IRQ6 中断	439
14.8.7	数据寄存器的通读 / 盖写	440

14.8.8	关于复位	440
14.8.9	EP0 相关中断源的分配	440
14.8.10	关于 VBUS、 $\overline{\text{IRQx}}$ 管脚的电平转移	440
14.8.11	USB 终点数据寄存器的读取 / 写入	441
14.8.12	向各低功耗模式转换和解除各低功耗模式时的限制事项	442
14.8.13	关于 USB 外部电路	443
14.8.14	未使用 USB 时的管脚处理	443
14.8.15	使用 TR 中断的注意事项	444
15.	A/D 转换器	445
15.1	特点	445
15.2	输入 / 输出管脚	447
15.3	寄存器说明	448
15.3.1	A/D 数据寄存器 A ~ D (ADDRA ~ ADDRD)	448
15.3.2	A/D 控制 / 状态寄存器 (ADCSR)	449
15.3.3	A/D 控制寄存器 (ADCR)	450
15.4	与总线主控器的接口	451
15.5	运行说明	452
15.5.1	单通道模式	452
15.5.2	扫描模式	453
15.5.3	输入采样和 A/D 转换时间	454
15.5.4	外部触发器输入时序	455
15.6	中断源	455
15.7	A/D 转换精度的定义	456
15.8	使用时的注意事项	458
15.8.1	模块停止模式的设定	458
15.8.2	关于容许信号源阻抗	458
15.8.3	对绝对精度的影响	458
15.8.4	模拟电源管脚其他的设定范围	458
15.8.5	电路板设计时的注意事项	459
16.	ROM	460
17.	闪存 (F-ZTAT 版)	461
17.1	特点	461
17.2	模式转移图	463
17.3	块结构	466
17.4	输入 / 输出管脚	468
17.5	寄存器的说明	469
17.5.1	闪存控制寄存器 1 (FLMCR1)	469
17.5.2	闪存控制寄存器 2 (FLMCR2)	470
17.5.3	擦除块指定寄存器 1 (EBR1)	470
17.5.4	擦除块指定寄存器 2 (EBR2)	470
17.5.5	RAM 仿真寄存器 (RAMER)	471
17.5.6	串行控制寄存器 X (SCRX)	471
17.6	单板上编程模式	472
17.6.1	SCI 引导模式 (HD64F2218、HD64F2212、HD64F2211)	472
17.6.2	USB 引导模式 (HD64F2218U、HD64F2212U、HD64F2211U)	475
17.6.2.1	特点	475
17.6.2.2	USB 引导模式运行注意事项	475
17.6.2.3	概要	476
17.6.3	用户编程模式	479

17.7	RAM 中的闪存仿真	480
17.8	闪存的编程 / 擦除	482
17.8.1	编程 / 编程验证	482
17.8.2	擦除 / 擦除验证	484
17.9	编程 / 擦除保护	486
17.9.1	硬件保护	486
17.9.2	软件保护	486
17.9.3	错误保护	486
17.10	编程 / 擦除闪存时的中断	486
17.11	编程器模式	487
17.12	闪存的低功耗状态注意事项	487
17.13	编程 / 擦除闪存时的注意事项	488
17.14	F-ZTAT 单片机的掩模型 ROM 化时的注意事项	493
18.	掩模型 ROM	494
18.1	特点	494
19.	时钟振荡器	495
19.1	寄存器的说明	496
19.1.1	系统时钟控制寄存器 (SCKCR)	496
19.1.2	低功率控制寄存器 (LPWRCR)	497
19.2	系统时钟振荡器	498
19.2.1	连接晶体振荡器的方法	498
19.2.2	输入外部时钟的方法	499
19.3	占空比调整电路	500
19.4	中速时钟分频器	500
19.5	总线主控器时钟发生电路	500
19.6	子时钟振荡器	501
19.6.1	连接 32.768KHz 连接晶体振荡器的方法	501
19.6.2	不使用子时钟的管脚处理	501
19.7	子时钟波形成形电路	502
19.8	USB 专用 PLL 电路	502
19.9	使用注意事项	502
19.9.1	振荡器注意事项	502
19.9.2	电路板设计注意事项	502
19.9.3	切换外部时钟注意事项	503
20.	低功耗状态	504
20.1	寄存器说明	508
20.1.1	待机控制寄存器 (SBYCR)	509
20.1.2	定时器控制 / 状态寄存器 (TCSR_1)	510
20.1.3	模块停止控制寄存器 A ~ C (MSTPCRA ~ MSTPCRC)	511
20.1.4	扩展模块停止寄存器 (EXMDLSTP)	513
20.2	中速模式	514
20.3	睡眠模式	515
20.3.1	向睡眠模式的转移	515
20.3.2	睡眠模式的解除	515
20.4	软件待机模式	516
20.4.1	向软件待机模式转移	516
20.4.2	软件待机模式的解除	516
20.4.3	软件待机模式解除后的振荡稳定时间的设置	517
20.4.4	软件待机模式的应用例	518

20.5	硬件待机模式	519
20.5.1	向硬件待机模式转移	519
20.5.2	硬件待机模式的解除	519
20.5.3	硬件待机模式的时序	519
20.5.4	有关硬件待机模式的转移时序	520
20.5.4.1	硬件待机模式的转移时序	520
20.5.4.2	从硬件待机模式恢复的时序	520
20.6	模块停止模式	521
20.7	监视模式	521
20.7.1	向监视模式转移	521
20.7.2	监视模式的解除	521
20.8	子睡眠模式	522
20.8.1	向子睡眠模式的转移	522
20.8.2	子睡眠模式的解除	522
20.9	子激活模式	523
20.9.1	向子激活模式的转移	523
20.9.2	子激活模式的解除	523
20.10	直接转移	523
20.10.1	从高速模式到子激活模式的直接转移	523
20.10.2	从子激活模式到高速模式的直接转移	523
20.11	ϕ 时钟输出控制	524
20.12	使用时的注意事项	524
20.12.1	I/O 端口的状态	524
20.12.2	振荡稳定待机中的消耗电流	524
20.12.3	闪存的模块停止	524
20.12.4	DMAC 的模块停止	524
20.12.5	内部外围模块的中断	524
20.12.6	子激活模式 / 监视模式转移和 DMAC 的模块停止	525
20.12.7	MSTPCR 的写入	525
21.	寄存器一览表	526
21.1	寄存器地址一览表 (按地址顺序)	527
21.2	寄存器位一览表	533
21.3	各工作模式下的寄存器状态	540
22.	电特性	546
22.1	绝对最大额定值	546
22.2	电源电压与运行频率范围	547
22.3	DC 特性	548
22.4	AC 特性	551
22.4.1	时钟时序	552
22.4.2	控制信号时序	554
22.4.3	总线时序	556
22.4.4	内部外围模块时序	561
22.5	USB 特性	565
22.6	A/D 转换特性	566
22.7	闪存特性	567
22.8	使用注意事项	568
22.8.1	设计印刷电路板注意事项	568
22.8.2	F-ZTAT 版与掩模型 ROM 版的特性	568
附录		569

附录 1. 各管脚状态中的 I/O 端口状态	569
附录 2. 型号一览.....	572
附录 3. 外形尺寸图.....	573
索引	577

H8S/2218 群、H8S/2212 群

瑞萨 16 位单片机

1. 概要

1.1 特点

- 16位高速H8S/2000CPU
与H8/300CPU、H8/300HCPU在目标级向上兼容
通用寄存器：16位×16个
基本指令：65种
- 丰富的外围功能
DMA控制器（DMAC）
16位定时器脉冲单元（TPU）
看门狗定时器（WDT）
实时时钟（RTC）
串行通信接口（SCI）
边界扫描
通用串行总线（USB）
10位A/D转换器
用户调试接口（H-UDI）
时钟振荡器
- On-chip存储器
H8S/2218群

ROM	型号名称	ROM	RAM	备注
闪存版	HD64F2218	128K 字节	12K 字节	SCI 引导版
	HD64F2218U	128K 字节	12K 字节	USB 引导版
掩膜 ROM 版	HD6432217	64K 字节	8K 字节	

H8S/2212 群

ROM	型号名称	ROM	RAM	备注
闪存版	HD64F2212	128K 字节	12K 字节	SCI 引导版
	HD64F2212U	128K 字节	12K 字节	USB 引导版
	HD64F2211	64K 字节	8K 字节	SCI 引导版
	HD64F2211U	64K 字节	8K 字节	USB 引导版
掩膜 ROM 版	HD6432211	64K 字节	8K 字节	
	HD6432210	32K 字节	4K 字节	

- 通用输入输出端口
输入输出端口
H8S/2218 群: 69 个
H8S/2212 群: 37 个
- 支持各种低功耗模式
- 小型封装

封装	(代码) *1	尺寸	管脚节距	备注
TQFP-100	TFP-100G、TFP-100GV	12.0 × 12.0mm	0.4mm	H8S/2218 群
P-LFBGA-112	BP-112、BP-112V	10.0 × 10.0mm	0.8mm	
LQFP-64*2	FP-64E、FP-64EV	10.0 × 10.0mm	0.5mm	H8S/2212 群
VQFN-64*2	TNP-64B、TNP-64BV	8.0 × 8.0mm	0.4mm	

【注】 *1 封装代码的末尾是 V 的封装为无铅产品。

*2 上述产品中包括开发及计划中的产品。
各产品的情况请向本公司销售人员确认。

1.2 内部框图

HD64F2218、HD64F2218U 的内部框图如图 1.1、HD6432217 的内部框图如图 1.2、HD64F2212、HD64F2212U、HD64F2211、HD64F2211U 的内部框图如图 1.3、HD6432211、HD6432210 的内部框图如图 1.4 所示。

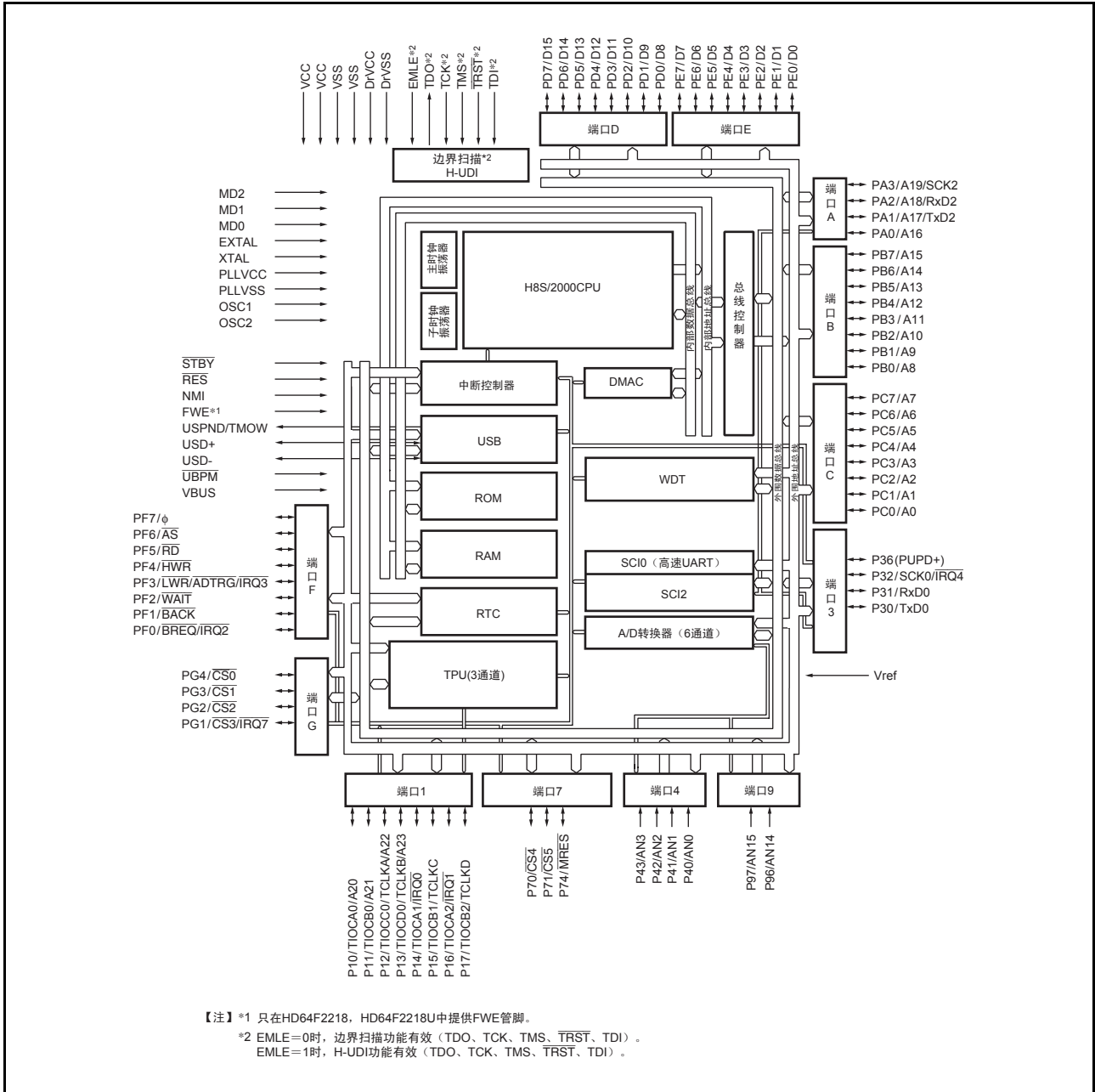


图 1.1 HD64F2218、HD64F2218U 内部框图

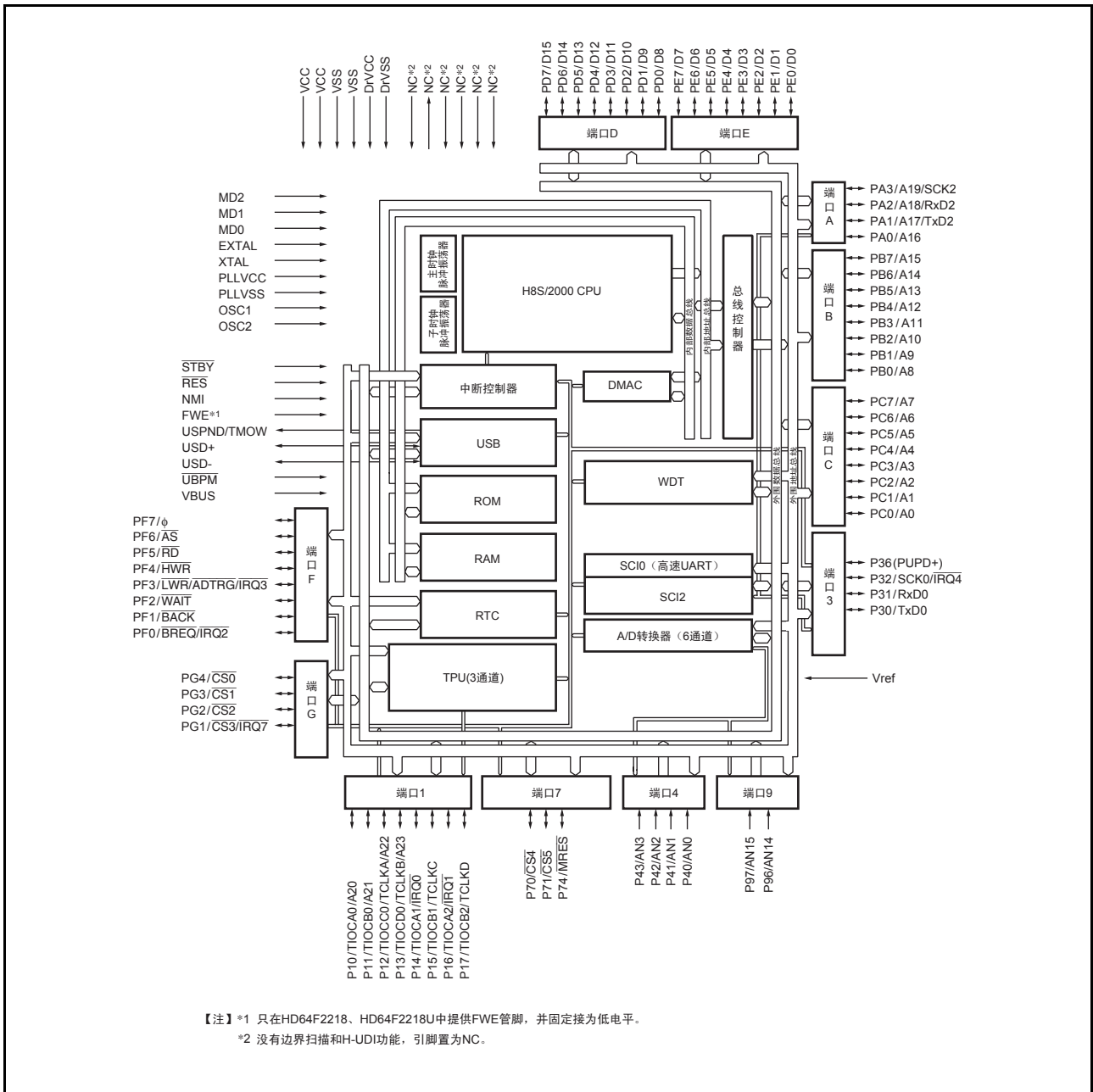


图 1.2 HD6432217 内部框图

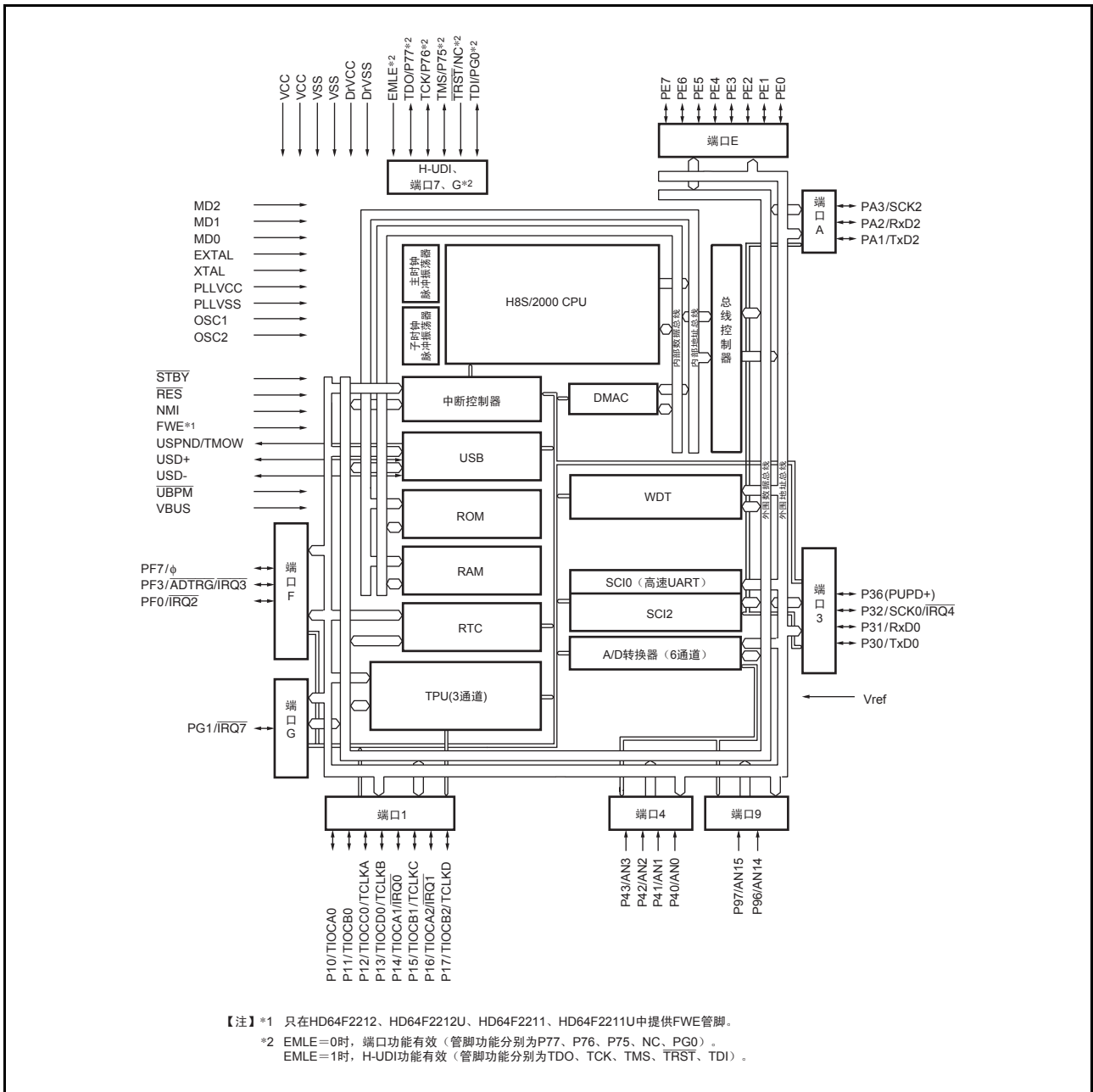


图 1.3 HD64F2212、HD64F2212U、HD64F2211、HD64F2211U 内部框图

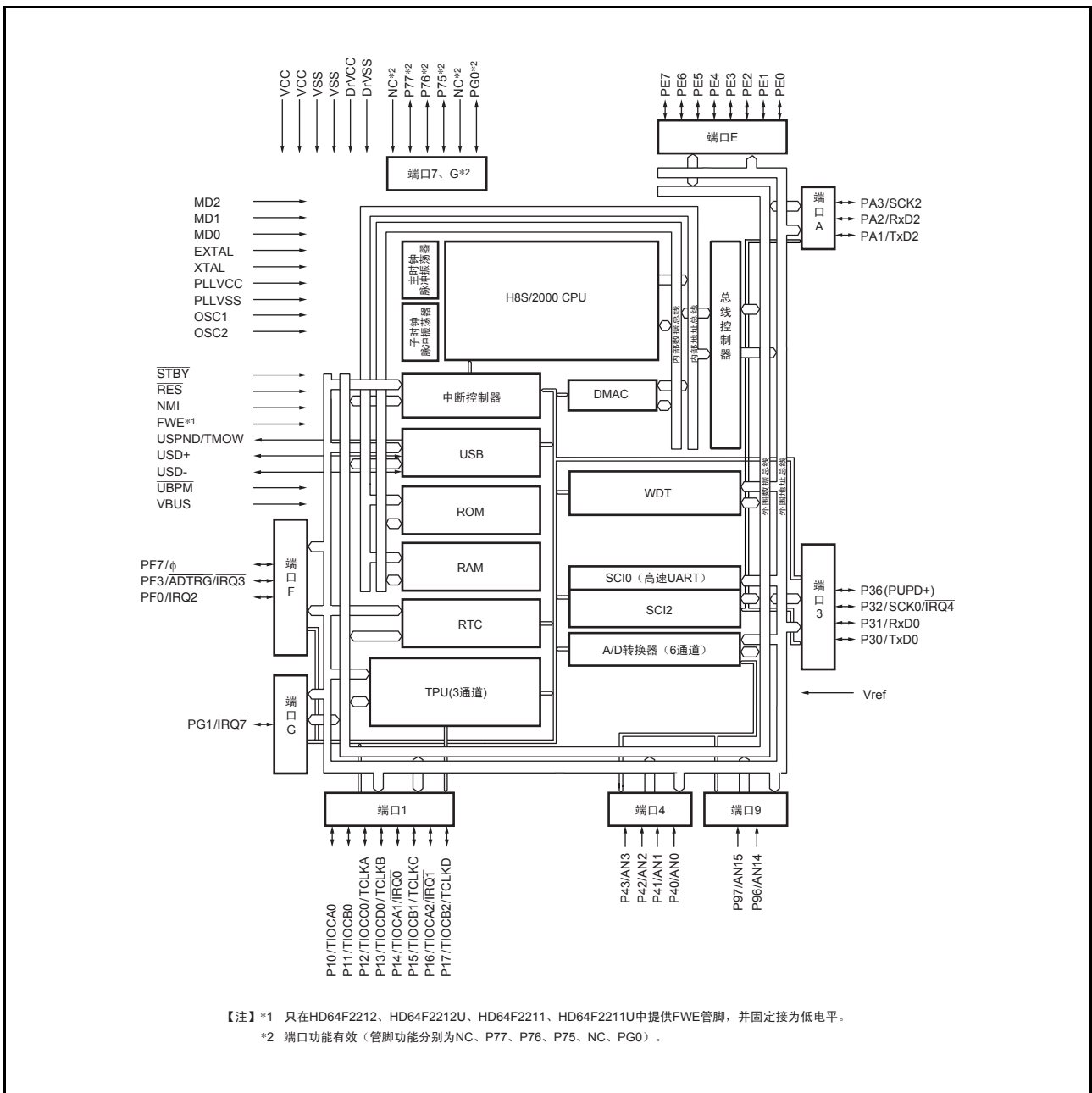


图 1.4 HD6432211、HD6432210 内部框图

	A	B	C	D	E	F	G	H	J	K	L
11	NC	PD3/D11	PD0/D8	PE5/D5	PE2/D2	P70/ $\overline{CS4}$	XTAL	\overline{STBY}	OSC1	PB7/A15	NC
10	PD5/D13	PD4/D12	PD2/D10	PE7/D7	PE3/D3	PE0/D0	EXTAL	P71/ $\overline{CS5}$	OSC2	PB6/A14	PB5/A13
9	FWE* ¹	PD7/D15	NC	PD1/D9	PE4/D4	VCC	VSS	P74/ \overline{MRES}	NC	PB4/A12	\overline{UBPM}
8	TDO* ²	EMLE* ²	NMI	PD6/D14	PE6/D6	PE1/D1	\overline{RES}	NC	PLLVCC	PLLVSS	P41/AN1
7	\overline{TRST} * ²	TDI* ²	TMS* ²	TCK* ²	BP-112 BP-112V (俯视图)			P40/AN0	P42/AN2	P43/AN3	NC
6	PF7/ ϕ	VSS	VCC	PF6/ \overline{AS}				Vref	PB1/A9	PB3/A11	PB2/A10
5	PF5/ \overline{RD}	PF4/ \overline{HWR}	PF3/ \overline{LWR} / \overline{ADTRG} / $\overline{IRQ3}$	PF1/ \overline{BACK}				D/VSS	P97/ AN15	PB0/A8	P96/ AN14
4	PF2/ \overline{WAIT}	PF0/ \overline{BREQ} / $\overline{IRQ2}$	PA2/A18/ Rx/D2	P11/ $\overline{TIOCB0}$ / A21	P17/ $\overline{TIOCB2}$ / \overline{TCLKD}	MD2	USPND/ \overline{TMOW}	PG4/ $\overline{CS0}$	D/VCC	USD+	USD-
3	PA3/A19/ SCK2	PA1/A17/ Tx/D2	NC	P14/ $\overline{TIOCA1}$ / $\overline{IRQ0}$	PC0/A0	PC3/A3	PC6/A6	P32/ $\overline{SCK0}$ / $\overline{IRQ4}$	NC	VBUS	P36 (PUDP+)
2	NC	PA0/A16	P12/ $\overline{TIOCC0}$ / \overline{TCLKA} / A22	P15/ $\overline{TIOCB1}$ / \overline{TCLKC}	PC2/A2	MD1	PC5/A5	P30/TxD0	PG1/ $\overline{CS3}$ / $\overline{IRQ7}$	PG2/ $\overline{CS2}$	PG3/ $\overline{CS1}$
1	NC	P10/ $\overline{TIOCA0}$ / A20	P13/ $\overline{TIOCD0}$ / \overline{TCLKB} / A23	P16/ $\overline{TIOCA2}$ / $\overline{IRQ1}$	PC1/A1	MD0	PC4/A4	PC7/A7	P31/RxD0	NC	NC
	A	B	C	D	E	F	G	H	J	K	L

【注】NC (No Connection) : 不连接任何电路, 请在Open状态使用。

*1 只在HD64F2218、HD64F2218U中提供FWE管脚。

*2 EMLE=0时, 边界扫描功能有效 (管脚功能分别为TDO、TCK、TMS、 \overline{TRST} 、TDI)
EMLE=1时, H-UDI功能有效 (管脚功能分别为TDO、TCK、TMS、 \overline{TRST} 、TDI)

图 1.6 HD64F2218、HD64F2218U 管脚排列图 (BP-112、BP-112V)

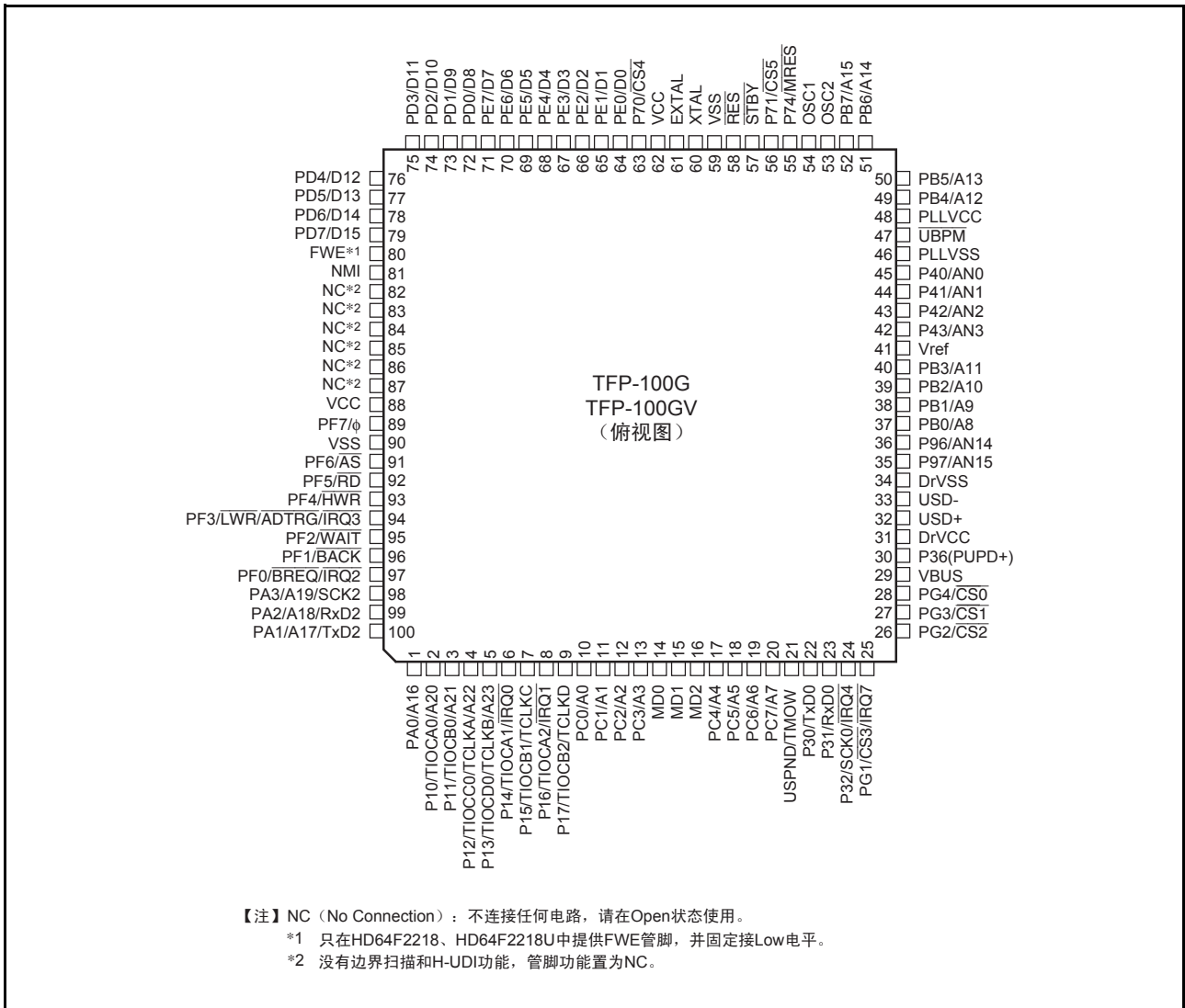


图 1.7 HD6432217 频配置图 (TFP-100G、TFP-100GV)

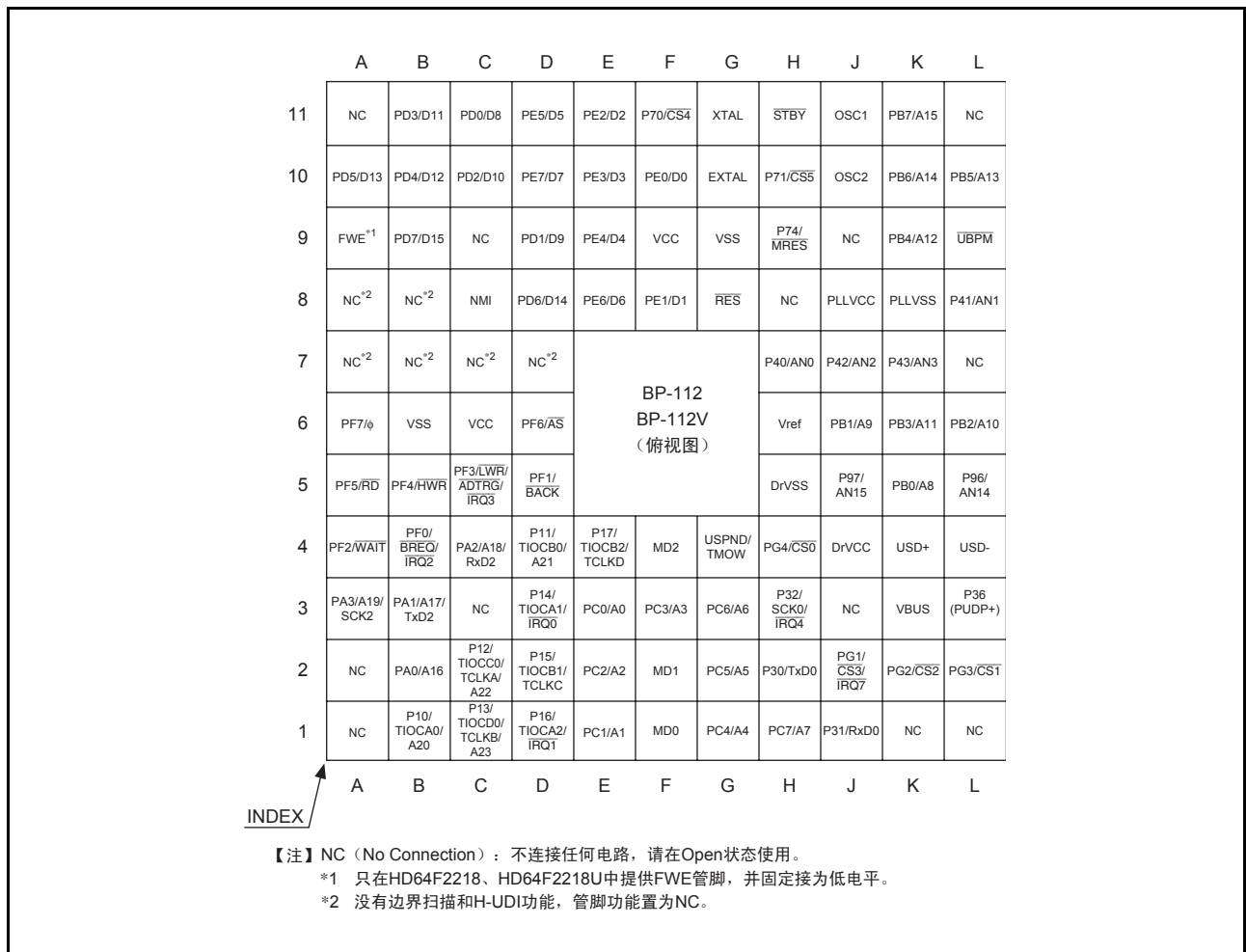


图 1.8 HD6432217 管脚排列图 (BP-112、BP-112V)

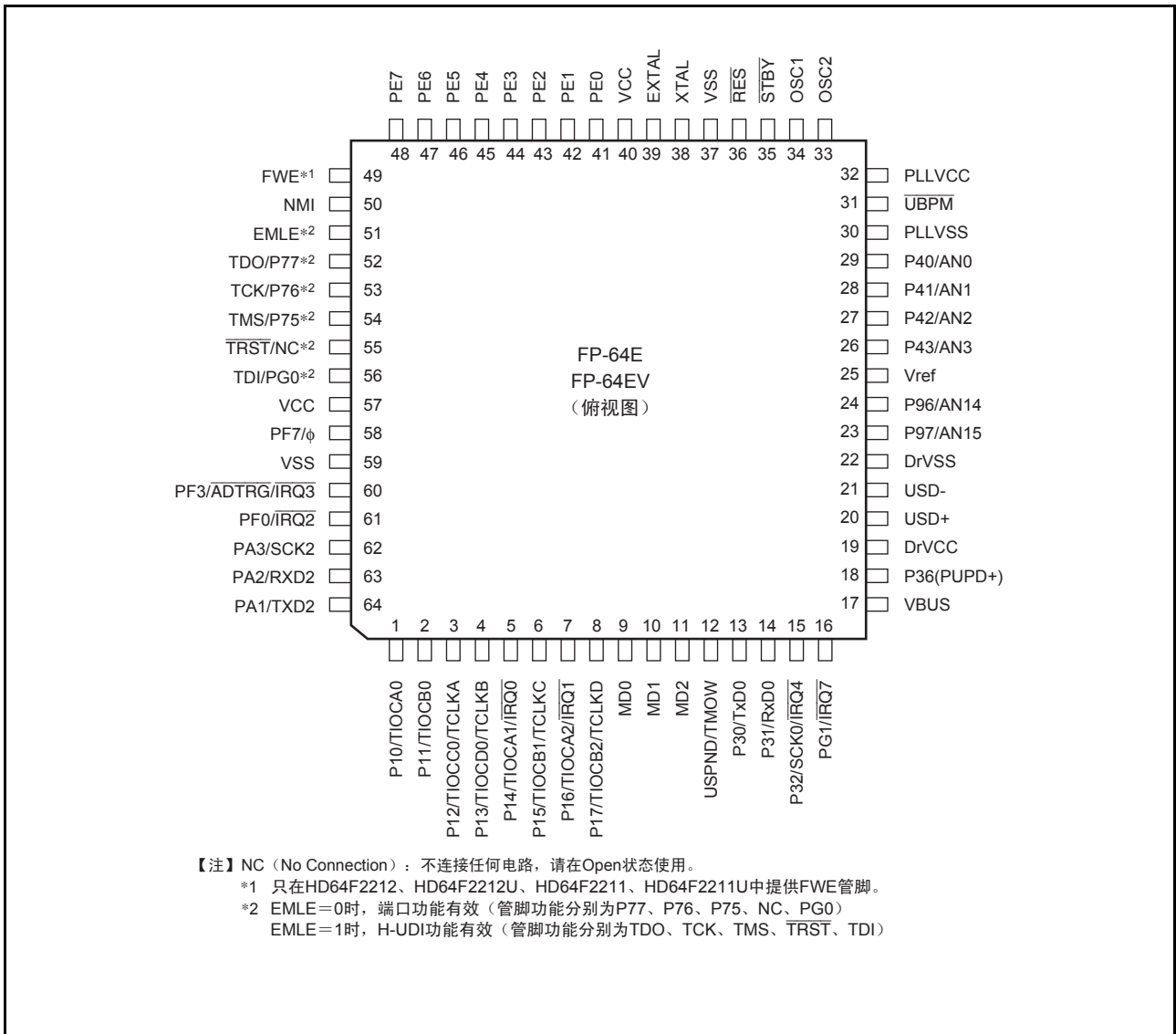


图 1.9 HD64F2212、HD64F2212U、HD64F2211、HD64F2211U 管脚排列图 (FP-64E、FP-64EV)

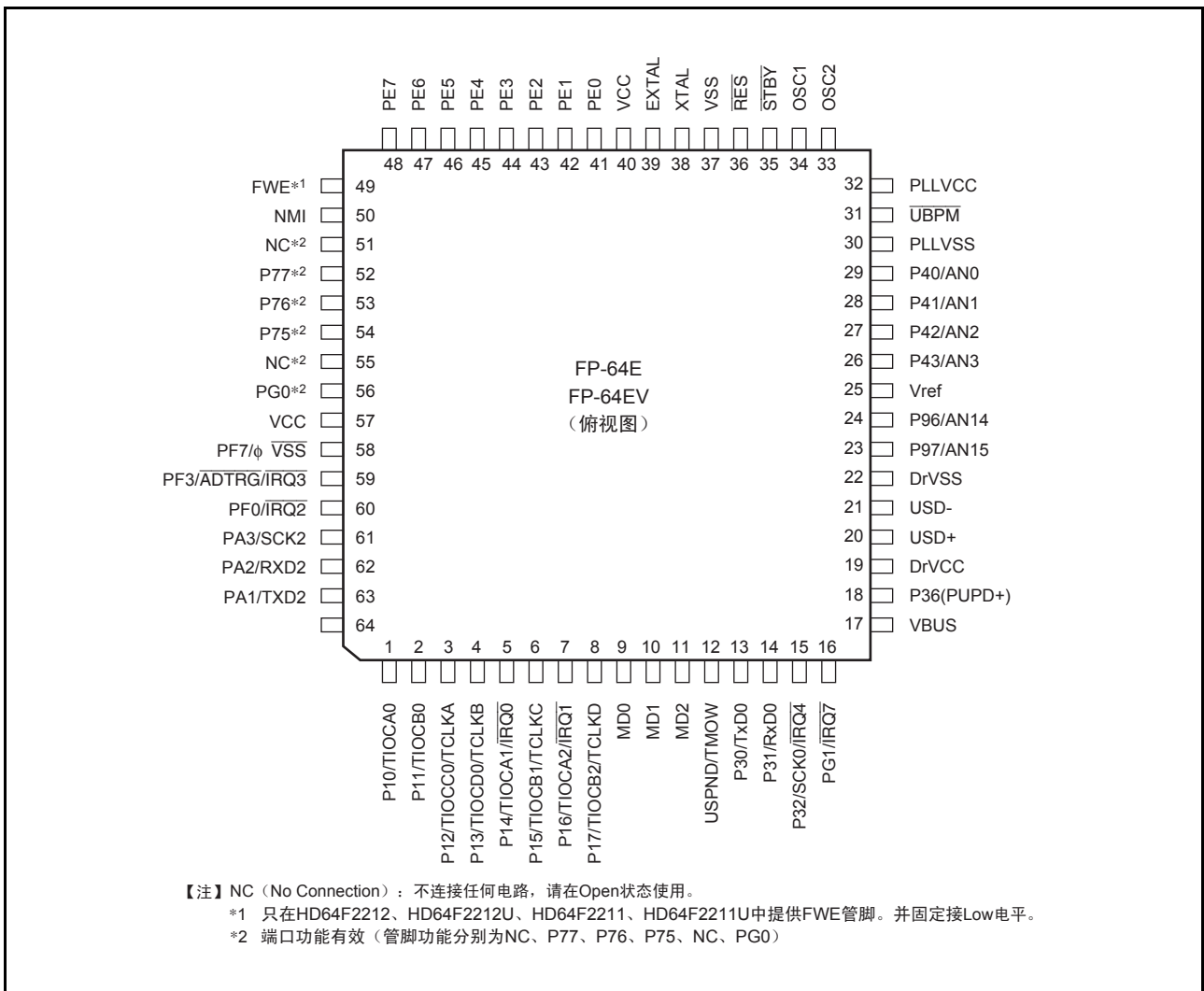


图 1.10 HD6432211、HD6432210 管脚排列图 (FP-64E、FP-64EV)

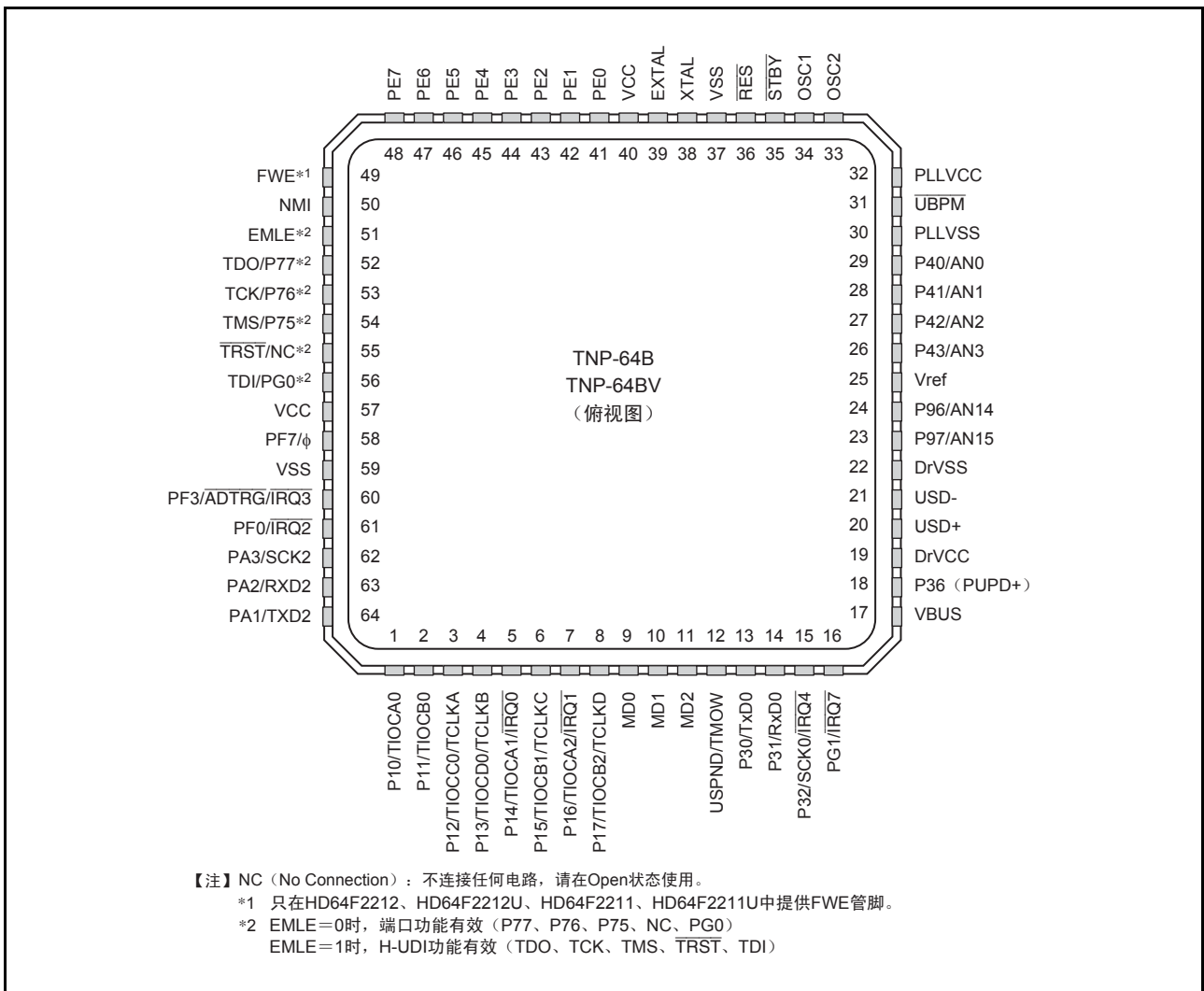


图 1.11 HD64F2212、HD64F2212U、HD64F2211、HD64F2211U 的管脚排列图 (TNP-64B、TNP-64BV)

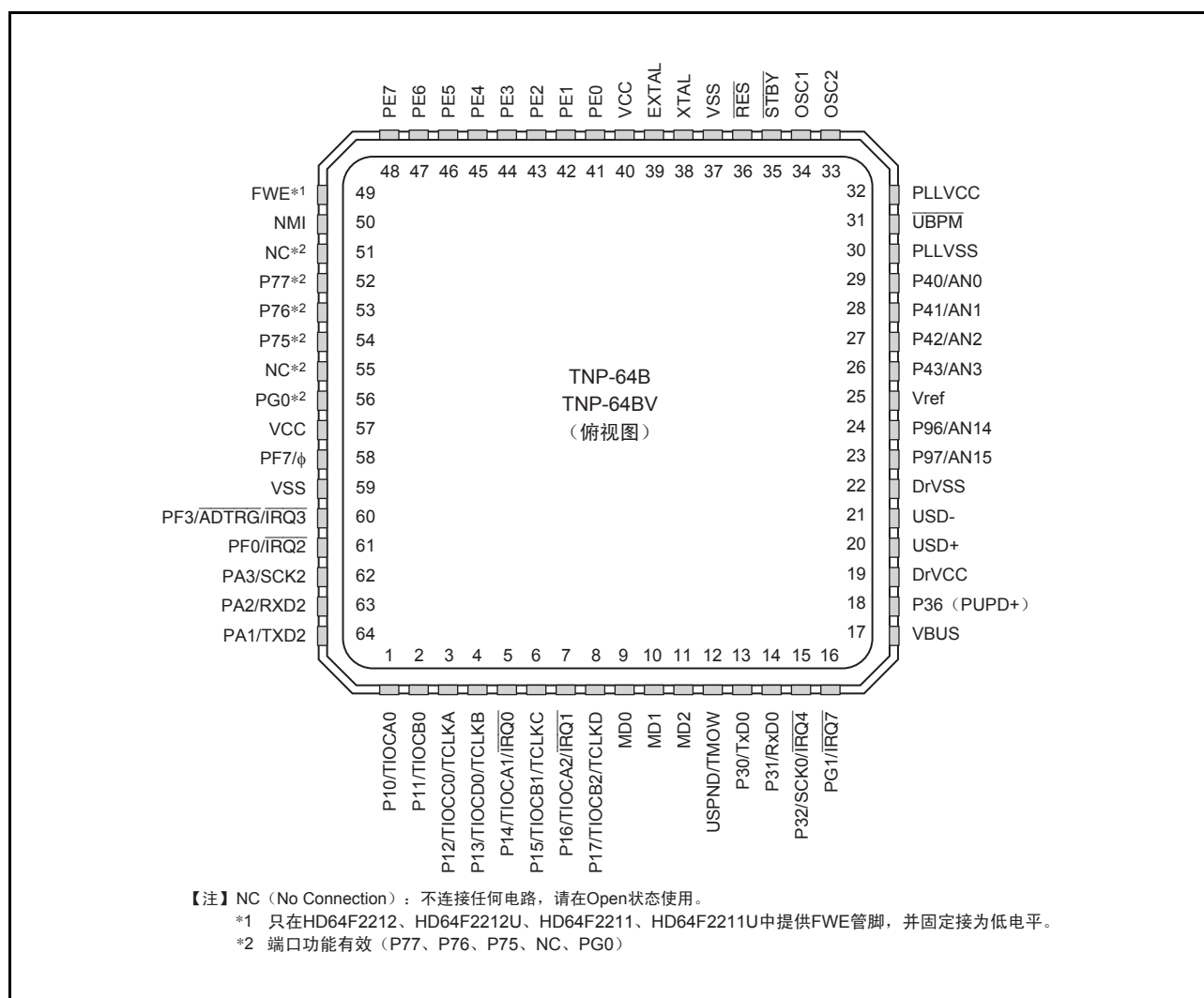


图 1.12 HD6432211、HD6432210 的管脚排列图 (TNP-64B、TNP-64BV)

1.4 运行模式其他管脚功能一览

H8S/2218 群的管脚功能一览如表 1.1，H8S/2212 群的管脚功能一览如表 1.2 所示。

表 1.1 H8S/2218 群的运行模式其他管脚功能一览

管脚号		管脚名称*			
TFP-100G TFP-100GV	BP-112 BP-112V	模式 4、5	模式 6	模式 7	闪存写入模式
1	B2	PA0/A16	PA0/A16	PA0	NC
2	B1	P10/TIOCA0/A20	P10/TIOCA0/A20	P10/TIOCA0	A2
3	D4	P11/TIOCB0/A21	P11/TIOCB0/A21	P11/TIOCB0	A3
4	C2	P12/TIOCC0/TCLKA/A22	P12/TIOCC0/TCLKA/A22	P12/TIOCC0/TCLKA	A4
5	C1	P13/TIOCD0/TCLKB/A23	P13/TIOCD0/TCLKB/A23	P13/TIOCD0/TCLKB	A5
6	D3	P14/TIOCA1/ $\overline{\text{IRQ0}}$	P14/TIOCA1/ $\overline{\text{IRQ0}}$	P14/TIOCA1/ $\overline{\text{IRQ0}}$	VSS
7	D2	P15/TIOCB1/TCLKC	P15/TIOCB1/TCLKC	P15/TIOCB1/TCLKC	$\overline{\text{WE}}$
8	D1	P16/TIOCA2/ $\overline{\text{IRQ1}}$	P16/TIOCA2/ $\overline{\text{IRQ1}}$	P16/TIOCA2/ $\overline{\text{IRQ1}}$	VSS
9	E4	P17/TIOCB2/TCLKD	P17/TIOCB2/TCLKD	P17/TIOCB2/TCLKD	$\overline{\text{CE}}$
10	E3	A0	PC0/A0	PC0	NC
11	E1	A1	PC1/A1	PC1	NC
12	E2	A2	PC2/A2	PC2	NC
13	F3	A3	PC3/A3	PC3	NC
14	F1	MD0	MD0	MD0	VSS
15	F2	MD1	MD1	MD1	VSS
16	F4	MD2	MD2	MD2	VSS
17	G1	A4	PC4/A4	PC4	NC
18	G2	A5	PC5/A5	PC5	NC
19	G3	A6	PC6/A6	PC6	NC
20	H1	A7	PC7/A7	PC7	NC
21	G4	USPND/TMOW	USPND/TMOW	USPND/TMOW	NC
22	H2	P30/TxD0	P30/TxD0	P30/TxD0	A10
23	J1	P31/RxD0	P31/RxD0	P31/RxD0	A11
24	H3	P32/SCK0/ $\overline{\text{IRQ4}}$	P32/SCK0/ $\overline{\text{IRQ4}}$	P32/SCK0/ $\overline{\text{IRQ4}}$	A12
25	J2	PG1/ $\overline{\text{CS3}}$ / $\overline{\text{IRQ7}}$	PG1/ $\overline{\text{CS3}}$ / $\overline{\text{IRQ7}}$	PG1/ $\overline{\text{IRQ7}}$	A15
26	K2	PG2/ $\overline{\text{CS2}}$	PG2/ $\overline{\text{CS2}}$	PG2	NC
27	L2	PG3/ $\overline{\text{CS1}}$	PG3/ $\overline{\text{CS1}}$	PG3	NC
28	H4	PG4/ $\overline{\text{CS0}}$	PG4/ $\overline{\text{CS0}}$	PG4	NC
29	K3	VBUS	VBUS	VBUS	VSS
30	L3	P36 (PUDP+)	P36 (PUDP+)	P36 (PUDP+)	A16

管脚号		管脚名称*			
TFP-100G	BP-112	模式 4、5	模式 6	模式 7	闪存写入模式
TFP-100GV	BP-112V				
31	J4	DrVCC	DrVCC	DrVCC	VCC
32	K4	USD+	USD+	USD+	NC
33	L4	USD-	USD-	USD-	NC
34	H5	DrVSS	DrVSS	DrVSS	VSS
35	J5	P97/AN15	P97/AN15	P97/AN15	A7
36	L5	P96/AN14	P96/AN14	P96/AN14	A6
37	K5	PB0/A8	PB0/A8	PB0	NC
38	J6	PB1/A9	PB1/A9	PB1	NC
39	L6	PB2/A10	PB2/A10	PB2	NC
40	K6	PB3/A11	PB3/A11	PB3	NC
41	H6	Vref	Vref	Vref	VCC
42	K7	P43/AN3	P43/AN3	P43/AN3	A14
43	J7	P42/AN2	P42/AN2	P42/AN2	A13
44	L8	P41/AN1	P41/AN1	P41/AN1	A9
45	H7	P40/AN0	P40/AN0	P40/AN0	A8
46	K8	PLLVSS	PLLVSS	PLLVSS	VSS
47	L9	$\overline{\text{UBPM}}$	$\overline{\text{UBPM}}$	$\overline{\text{UBPM}}$	A17
48	J8	PLLVCC	PLLVCC	PLLVCC	VCC
49	K9	PB4/A12	PB4/A12	PB4	NC
50	L10	PB5/A13	PB5/A13	PB5	NC
51	K10	PB6/A14	PB6/A14	PB6	NC
52	K11	PB7/A15	PB7/A15	PB7	NC
53	J10	OSC2	OSC2	OSC2	NC
54	J11	OSC1	OSC1	OSC1	VCC
55	H9	$\overline{\text{P74/MRES}}$	$\overline{\text{P74/MRES}}$	$\overline{\text{P74/MRES}}$	NC
56	H10	$\overline{\text{P71/CS5}}$	$\overline{\text{P71/CS5}}$	P71	NC
57	H11	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	VCC
58	G8	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$
59	G9	VSS	VSS	VSS	VSS
60	G11	XTAL	XTAL	XTAL	XTAL
61	G10	EXTAL	EXTAL	EXTAL	EXTAL
62	F9	VCC	VCC	VCC	VCC
63	F11	$\overline{\text{P70/CS4}}$	$\overline{\text{P70/CS4}}$	P70	NC
64	F10	PE0/D0	PE0/D0	PE0	D0
65	F8	PE1/D1	PE1/D1	PE1	D1

管脚号		管脚名称*			
TFP-100G	BP-112	模式 4、5	模式 6	模式 7	闪存写入模式
TFP-100GV	BP-112V				
66	E11	PE2/D2	PE2/D2	PE2/D2	D2
67	E10	PE3/D3	PE3/D3	PE3/D3	D3
68	E9	PE4/D4	PE4/D4	PE4/D4	D4
69	D11	PE5/D5	PE5/D5	PE5/D5	D5
70	E8	PE6/D6	PE6/D6	PE6	D6
71	D10	PE7/D7	PE7/D7	PE7	D7
72	C11	D8	D8	PD0	NC
73	D9	D9	D9	PD1	NC
74	C10	D10	D10	PD2	NC
75	B11	D11	D11	PD3	NC
76	B10	D12	D12	PD4	NC
77	A10	D13	D13	PD5	NC
78	D8	D14	D14	PD6	NC
79	B9	D15	D15	PD7	NC
80	A9	FWE	FWE	FWE	FWE
81	C8	NMI	NMI	NMI	VCC
82	B8	EMLE/NC	EMLE/NC	EMLE/NC	VSS
83	A8	TDO/NC	TDO/NC	TDO/NC	NC
84	D7	TCK/NC	TCK/NC	TCK/NC	VCC
85	C7	TMS/NC	TMS/NC	TMS/NC	VCC
86	A7	$\overline{\text{TRST}}/\text{NC}$	$\overline{\text{TRST}}/\text{NC}$	$\overline{\text{TRST}}/\text{NC}$	$\overline{\text{RES}}$
87	B7	TDI/NC	TDI/NC	TDI/NC	VSS
88	C6	VCC	VCC	VCC	VCC
89	A6	PF7/ ϕ	PF7/ ϕ	PF7/ ϕ	NC
90	B6	VSS	VSS	VSS	VSS
91	D6	$\overline{\text{AS}}$	$\overline{\text{AS}}$	PF6	NC
92	A5	$\overline{\text{RD}}$	$\overline{\text{RD}}$	PF5	NC
93	B5	HWR	HWR	PF4	NC
94	C5	$\overline{\text{PF3/LWR/ADTRG/IRQ3}}$	$\overline{\text{PF3/LWR/ADTRG/IRQ3}}$	$\overline{\text{PF3/LWR/ADTRG/IRQ3}}$	VCC
95	A4	$\overline{\text{PF2/WAIT}}$	$\overline{\text{PF2/WAIT}}$	PF2	NC
96	D5	$\overline{\text{PF1/BACK}}$	$\overline{\text{PF1/BACK}}$	PF1	NC
97	B4	$\overline{\text{PF0/BREQ/IRQ2}}$	$\overline{\text{PF0/BREQ/IRQ2}}$	$\overline{\text{PF0/IRQ2}}$	VCC
98	A3	PA3/A19/SCK2	PA3/A19/SCK2	PA3/SCK2	A1
99	C4	PA2/A18/RxD2	PA2/A18/RxD2	PA2/RxD2	A0
100	B3	PA1/A17/TxD2	PA1/A17/TxD2	PA1/TxD2	$\overline{\text{OE}}$

【注】 * 将 NC 置为开路状态。

表 1.2 H8S/2212 群的运行模式其他管脚功能一览

管脚号	管脚名称 *	
FP-64E FP-64EV TNP-64B TNP-64BV	模式 7	闪存写入模式
1	P10/TIOC	A2
2	P11/TIOCB0	A3
3	P12/TIOCC0/TCLKA	A4
4	P13/TIOCD0/TCLKB	A5
5	P14/TIOCA1/ $\overline{\text{IRQ0}}$	VSS
6	P15/TIOCB1/TCLKC	$\overline{\text{WE}}$
7	P16/TIOCA2/ $\overline{\text{IRQ1}}$	VSS
8	P17/TIOCB2/TCLKD	$\overline{\text{CE}}$
9	MD0	VSS
10	MD1	VSS
11	MD2	VSS
12	USPND/TMOW	NC
13	P30/TxD0	A10
14	P31/RxD0	A11
15	P32/SCK0/ $\overline{\text{IRQ4}}$	A12
16	PG1/ $\overline{\text{IRQ7}}$	A15
17	VBUS	VSS
18	P36 (PUPD+)	A16
19	DrVCC	VCC
20	USD+	NC
21	USD-	NC
22	DrVSS	VSS
23	P97/AN15	A7
24	P96/AN14	A6
25	Vref	VCC
26	P43/AN3	A14
27	P42/AN2	A13
28	P41/AN1	A9
29	P40/AN0	A8
30	PLLSS	VSS
31	$\overline{\text{UBPM}}$	A17
32	PLLCC	VCC

管脚号	管脚名称 *	
FP-64E FP-64EV TNP-64B TNP-64BV	模式 7	闪存写入模式
33	OSC2	NC
34	OSC1	VCC
35	$\overline{\text{STBY}}$	VCC
36	$\overline{\text{RES}}$	$\overline{\text{RES}}$
37	VSS	VSS
38	XTAL	XTAL
39	EXTAL	EXTAL
40	VCC	VCC
41	PE0	D0
42	PE1	D1
43	PE2	D2
44	PE3	D3
45	PE4	D4
46	PE5	D5
47	PE6	D6
48	PE7	D7
49	FWE	FWE
50	NMI	VCC
51	EMLE/NC	VSS
52	TDO/P77	NC
53	TCK/P76	VCC
54	TMS/P75	VCC
55	$\overline{\text{TRST/NC}}$	$\overline{\text{RES}}$
56	TDI/PG0	VSS
57	VCC	VCC
58	PF7/ ϕ	NC
59	VSS	VSS
60	PF3/ $\overline{\text{ADTRG/IRQ3}}$	VCC
61	PF0/ $\overline{\text{IRQ2}}$	VCC
62	PA3 /SCK2	A1
63	PA2/RxD2	A0
64	PA1/TxD2	$\overline{\text{OE}}$

【注】 * 将 NC 置为开路状态。

1.5 管脚功能

分类	记号	管脚编号			输入 / 输出	功能
		TFP-100G TFP-100GV	BP-112 BP-112V	FP-64E FP-64EV TNP-64B TNP-64BV		
电源	VCC	62 88	F9 C6	40 57	输入	本管脚是电源管脚。请连接在系统电源上。
	VSS	59 90	G9 B6	37 59	输入	本管脚为接地管脚。请连接在系统电源上 (0V)。
	PLLVC	48	J8	32	输入	本管脚为内置 PLL 振荡器用的电源管脚。请连接在系统电源上。
	PLLVS	46	K8	30	输入	本管脚为内置 PLL 振荡器用的接地管脚。
时钟	XTAL	60	G11	38	输入	本管脚连接晶体振荡器。连接晶体振荡器及外部时钟输入时的连接例请参考 19. 时钟振荡器。
	EXTAL	61	G10	39	输入	本管脚连接晶体振荡器。并且 EXTAL 管脚也可以输入外部时钟。连接晶体振荡器及外部时钟输入时的连接例请参考 19. 时钟振荡器。
	OSC1 OSC2	54 53	J11 J10	34 33	输入	本管脚连接 32.768KHz 的晶体振荡器。连接晶体振荡器时的连接例请参考 19. 时钟振荡器。
	ϕ	89	A6	58	输出	向外部设备供应系统时钟。
运行控制模式	MD2 MD1 MD0	16 15 14	F4 F2 F1	11 10 9	输入	设定运行模式。在运行中不能使这些管脚发生变化。 除模式切换外，模式管脚 (MD2 ~ MD0) 在断电之前，必须下拉或上拉，并固定电平。
系统控制	$\overline{\text{RES}}^*$	58	G8	36	输入	本管脚为复位管脚。如果本管脚变为低电平状态，则成为复位状态。
	$\overline{\text{STBY}}^*$	57	H11	35	输入	如果本管脚变为低电平状态，则转换为硬件待机模式。
	$\overline{\text{MRES}}$	55	H9	—	输入	如果本管脚变为低电平状态，则成为手动复位状态。(仅限 H8S/2218 群)
	$\overline{\text{BREQ}}$	97	B4	—	输入	对于本 LSI，外部总线主控制器发生总线请求。(仅限 H8S/2218 群)
	$\overline{\text{BACK}}$	96	D5	—	输出	表示将总线向外部总线主控制器开放。(仅限 H8S/2218 群)
	FWE	80	A9	49	输入	本管脚是闪存用的管脚。仅适用于闪存版。掩模 ROM 版中，将它连接在系统电源 (0V) 上。

分类	符号	管脚编号			输入 / 输出	功能
		TFP-100G TFP-100GV	BP-112 BP-112V	FP-64E FP-64EV TNP-64B TNP-64BV		
系统控制	EMLE	82	B8	51	输入	仿真器允许。E10A 未使用时，请连接在系统电源（0V）。 仅在 E10A 使用时，置为高电平状态。
中断	NMI	81	C8	50	输入	本管脚为不可屏蔽中断请求管脚。不使用时，固定为高电平。
	IRQ7	25	J2	16	输入	请求可屏蔽的中断。
	IRQ4	24	H3	15		
	IRQ3	94	C5	60		
	IRQ2	97	B4	61		
	IRQ1	8	D1	7		
	IRQ0	6	D3	5		
地址总线	A23	5	C1	—	输出	输出地址。 (仅限 H8S/2218 群)
	A22	4	C2	—		
	A21	3	D4	—		
	A20	2	B1	—		
	A19	98	A3	—		
	A18	99	C4	—		
	A17	100	B3	—		
	A16	1	B2	—		
	A15	52	K11	—		
	A14	51	K10	—		
	A13	50	L10	—		
	A12	49	K9	—		
	A11	40	K6	—		
	A10	39	L6	—		
	A9	38	J6	—		
	A8	37	K5	—		
	A7	20	H1	—		
	A6	19	G3	—		
	A5	18	G2	—		
	A4	17	G1	—		
	A3	13	F3	—		
A2	12	E2	—			
A1	11	E1	—			
A0	10	E3	—			

分类	符号	管脚编号			输入 / 输出	功能
		TFP-100G TFP-100GV	BP-112 BP-112V	FP-64E FP-64EV TNP-64B TNP-64BV		
数据总线	D15	79	B9	—	输入 / 输出	双向数据总线。 (仅针对 H8S/2218 群)
	D14	78	D8	—		
	D13	77	A10	—		
	D12	76	B10	—		
	D11	75	B11	—		
	D10	74	C10	—		
	D9	73	D9	—		
	D8	72	C11	—		
	D7	71	D10	—		
	D6	70	E8	—		
	D5	69	D11	—		
	D4	68	E9	—		
	D3	67	E10	—		
	D2	66	E11	—		
	D1	65	F8	—		
	D0	64	F10	—		
总线控制	$\overline{\text{CS5}}$	56	H10	—	输出	外部地址空间的分割区域 5 ~ 0 的选择信号。 (仅限 H8S/2218 群)
	$\overline{\text{CS4}}$	63	F11	—		
	$\overline{\text{CS3}}$	25	J2	—		
	$\overline{\text{CS2}}$	26	K2	—		
	$\overline{\text{CS1}}$	27	L2	—		
	$\overline{\text{CS0}}$	28	H4	—		
	$\overline{\text{AS}}$	91	D6	—	输出	本管脚为低电平状态时，表示地址总线上的地址输出有效。(仅限 H8S/2218 群)
	$\overline{\text{RD}}$	92	A5	—	输出	本管脚为低电平时，表示外部地址空间的读取有效状态。(仅限 H8S/2218 群)
	$\overline{\text{HWR}}$	93	B5	—	输出	表示写入外部地址空间时，数据总线的高位 (D15 ~ D8) 为有效选通信号。(仅限 H8S/2218 群)
	$\overline{\text{LWR}}$	94	C5	—	输出	表示写入外部地址空间时，数据总线的低位 (D7 ~ D0) 为有效选通信号。(仅限 H8S/2218 群)
$\overline{\text{WAIT}}$	95	A4	—	输入	访问外部 3 态地址空间时，请求在总线周期插入等待状态。(仅限 H8S/2218 群)	

分类	符号	管脚编号			输入 / 输出	功能
		TFP-100G TFP-100GV	BP-112 BP-112V	FP-64E FP-64EV TNP-64B TNP-64BV		
16 位定时器 脉冲单元 (TPU)	TCLKA	4	C2	3	输入	TPU 的外部时钟输入管脚。
	TCLKB	5	C1	4		
	TCLKC	7	D2	6		
	TCLKD	9	E4	8		
	TIOCA0	2	B1	1	输入 / 输出	TGRA_0 ~ TGRD_0 输入捕捉输入 / 输出比较输出 / PWM 输出管脚。
	TIOCB0	3	D4	2		
	TIOCC0	4	C2	3		
	TIOCD0	5	C1	4		
	TIOCA1	6	D3	5	输入 / 输出	TGRA_1、TGRB_1 的输入捕捉输入 / 输出比较输出 / PWM 输出管脚。
	TIOCB1	7	D2	6		
	TIOCA2	8	D1	7	输入 / 输出	TGRA_2、TGRB_2 的输入捕捉输入 / 输出比较输出 / PWM 输出管脚。
	TIOCB2	9	E4	8		
实时时钟 (RTC)	TMOW	21	G4	12	输出	分频时钟输出管脚
串行通信接口 (SCI)	TxD2	100	B3	64	输出	数据输出管脚。
	TxD0	22	H2	13		
	RxD2	99	C4	63	输入	数据输入管脚。
	RxD0	23	J1	14		
SCK2	98	A3	62	输入 / 输出	时钟输入输出管脚。	
SCK0	24	H3	15			
A/D 转换器	AN15	35	J5	23	输入	A/D 转换器的模拟输入管脚。
	AN14	36	L5	24		
	AN3	42	K7	26		
	AN2	43	J7	27		
AN1	44	L8	28			
AN0	45	H7	29			
	$\overline{\text{ADTRG}}$	94	C5	60	输入	A/D 转换器开始的外部触发器输入管脚。
	Vref	41	H6	25	输入	本管脚为 A/D 转换器的基准电压输入管脚。 不使用 A/D 转换器时，请连接在系统电源上 (VCC)。
边界扫描 (仅针对 HD64F2218、 HD64F2218U)	TMS	85	C7	54	输入	本管脚为边界扫描用的控制信号输入管脚。
	TCK	84	D7	53	输入	本管脚为边界扫描用的时钟输入管脚。
	TDO	83	A8	52	输出	本管脚为边界扫描用的数据输出管脚。
	TDI	87	B7	56	输入	本管脚为边界扫描用的数据输入管脚。
	$\overline{\text{TRST}}$	86	A7	55	输入	本管脚为 TAP 控制器的复位管脚。

分类	符号	管脚编号			输入 / 输出	功能
		TFP-100G TFP-100GV	BP-112 BP-112V	FP-64E FP-64EV TNP-64B TNP-64BV		
USB	DrVCC	31	J4	19	输出	本管脚为内置收发器的电源管脚。请连接在系统电源上。
	DrVSS	34	H5	22	输入	本管脚为内置收发器的接地管脚。
	USD+* USD-*	32 33	K4 L4	20 21	输入 / 输出	本管脚为 USB 数据输入输出管脚。
	VBUS*	29	K3	17	输入	本管脚为检查 USB 线连接 / 断开的输入管脚。
	USPND	21	G4	12	输出	USB 挂起输出 如果转换为挂起状态，则成为高电平状态。
	UBPM	47	L9	31	输入	本管脚为总线电源 / 自身电源模式设定输入管脚。 在总线电源模式下使用时请固定为 0。自身电源模式下使用时请固定为 1。
	P36 (PUPD+)	30	L3	18	输入 / 输出	请将本管脚作为 D+ 信号的上拉控制来使用。
I/O 端口	P17	9	E4	8	输入 / 输出	本管脚为 8 位的输入输出管脚。
	P16	8	D1	7		
	P15	7	D2	6		
	P14	6	D3	5		
	P13	5	C1	4		
	P12	4	C2	3		
	P11	3	D4	2		
	P10	2	B1	1		
	P36 P32 P31 P30	30 24 23 22	L3 H3 J1 H2	18 15 14 13	输入	本管脚为 4 位的输入输出管脚。 (请将 P36 作为 USB 的 D+ 信号上拉控制管脚使用。)
	P43 P42 P41 P40	42 43 44 45	K7 J7 L8 H7	26 27 28 29	输入	本管脚为 4 位的输入管脚。
	P77 P76 P75 P74 P71 P70	— — — 55 56 63	— — — H9 H10 F11	52 53 54 — — —	输入 / 输出	本管脚为 3 位的输入输出管脚。
	P97 P96	35 36	J5 L5	23 24	输入	本管脚为 2 位的输入管脚。

分类	符号	管脚编号			输入 / 输出	功能
		TFP-100G TFP-100GV	BP-112 BP-112V	FP-64E FP-64EV TNP-64B TNP-64BV		
I/O 端口	PA3	98	A3	62	输入 / 输出	H8S/2218 群中本管脚为 4 位的输入输出管脚。 H8S/2212 群中本管脚为 3 位的输入输出管脚。
	PA2	99	C4	63		
	PA1	100	B3	64		
	PA0	1	B2	—		
	PB7	52	K11	—	输入 / 输出	本管脚为 8 位的输入输出管脚。 (仅针对 H8S/2218 群)
	PB6	51	K10	—		
	PB5	50	L10	—		
	PB4	49	K9	—		
	PB3	40	K6	—		
	PB2	39	L6	—		
	PB1	38	J6	—		
	PB0	37	K5	—		
	PC7	20	H1	—	输入 / 输出	本管脚为 8 位的输入输出管脚。 (仅针对 H8S/2218 群)
	PC6	19	G3	—		
	PC5	18	G2	—		
	PC4	17	G1	—		
	PC3	13	F3	—		
	PC2	12	E2	—		
	PC1	11	E1	—		
	PC0	10	E3	—		
	PD7	79	B9	—	输入 / 输出	本管脚为 8 位的输入输出管脚。 (仅针对 H8S/2218 群)
	PD6	78	D8	—		
	PD5	77	A10	—		
	PD4	76	B10	—		
	PD3	75	B11	—		
	PD2	74	C10	—		
	PD1	73	D9	—		
	PD0	72	C11	—		
PE7	71	D10	48	输入 / 输出	本管脚为 8 位的输入输出管脚。	
PE6	70	E8	47			
PE5	69	D11	46			
PE4	68	E9	45			
PE3	67	E10	44			
PE2	66	E11	43			
PE1	65	F8	42			
PE0	64	F10	41			

分类	符号	管脚编号			输入 / 输出	功能
		TFP-100G TFP-100GV	BP-112 BP-112V	FP-64E FP-64EV TNP-64B TNP-64BV		
I/O 端口	PF7	89	A6	58	输入 / 输出	H8S/2218 群中本管脚为 8 位的输入输出管脚。 H8S/2212 群中本管脚为 3 位的输入输出管脚。
	PF6	91	D6	—		
	PF5	92	A5	—		
	PF4	93	B5	—		
	PF3	94	C5	60		
	PF2	95	A4	—		
	PF1	96	D5	—		
	PF0	97	B4	61		
	PG4	28	H4	—	输入 / 输出	H8S/2218 群中本管脚为 4 位的输入输出管脚。 H8S/2212 群中本管脚为 2 位的输入输出管脚。
	PG3	27	L2	—		
	PG2	26	K2	—		
	PG1	25	J2	16		
	PG0	—	—	56		

【注】 * 可能会成为误动作的原因，所以请设法防止噪音。

2. CPU

H8S/2000CPU 是一个 32 位的高速处理器，该处理器可以向上与 H8/300、H8/300H CPU 兼容。H8S/2000CPU 有 16 个 16 位的通用寄存器，可寻址 16M 字节线性地址空间，用于实时控制十分理想。本章对 H8S/2000CPU 进行说明。根据产品的不同，能够使用的模式和地址空间也有所不同，所以具体每一个产品请参考 3. MCU 运行模式。

2.1 特点

- 向上与 H8/300、H8/300H CPU 兼容
可执行 H8/300、H8/300H CPU 目标程序
- 通用寄存器的结构：16 位×16 个
也可作为 8 位×16 个、32 位×8 个使用
- 基本指令：65 种
8 / 16 / 32 位运算指令
乘除运算指令
强大的位操作指令
- 寻址方式：8 种
寄存器直接寻址 (Rn)
寄存器间接寻址 (@ERn)
带位移量寄存器间接寻址 (@(d:16,ERn) / @(d:32,ERn))
后增寄存器间接寻址 / 先减寄存器间接寻址 (@ERn+ / @-ERn)
绝对地址寻址 (@aa:8 / @aa:16 / @aa:24 / @aa:32)
立即寻址 (#xx:8 / #xx:16 / #xx:32)
程序计数器相对寻址 (@(d:8,PC) / @(d:16,PC))
存储器间接寻址 (@@aa:8)
- 地址空间：16M 字节
程序：16M 字节
数据：16M 字节
- 高速运行
全部常用指令在 1~2 状态下执行
8 / 16 / 32 位寄存器之间的加减运算：1 状态
8×8 位寄存器之间的乘法运算：12 状态
16÷8 位寄存器之间的除法运算：12 状态
16×16 位寄存器之间的乘法运算：20 状态
32÷16 位寄存器之间的除法运算：20 状态
- CPU 运行模式：2 种
普通模式*/高级模式

【注】 * 在本 LSI 不能使用普通模式。

- 低功耗状态
通过 SLEEP 指令转换为低功耗状态
可选择 CPU 运行时钟

2.1.1 H8S/2600 CPU 与 H8S/2000 CPU 的区别

H8S/2600 CPU 和 H8S/2000 CPU 的区别如下所示。

- 寄存器构成
H8S/2600 CPU 仅支持 MAC 寄存器。
- 基本指令
仅 H8S/2600 CPU 支持 MAC、CLRMAC、LDMAC、STMAC4 种指令。
- MULXU、MULXS 指令的执行状态数

指令	助记符	执行状态	
		H8S/2600	H8S/2000
MULXU	MULXU.B Rs,Rd	3	12
	MULXU.W Rs,ERd	4	20
MULXS	MULXS.B Rs,Rd	4	13
	MULXS.W Rs,ERd	5	21

此外，根据产品的不同，地址空间、CCR 和 EXR 寄存器的功能、低功耗模式等也有所不同。

2.1.2 与 H8/300 CPU 的区别

H8S/2000 CPU 相对于 H8/300 CPU 增加和扩展了以下功能。

- 更多的通用寄存器、控制寄存器
增加了 16 位×8 个的扩展寄存器及 8 位×1 个、32 位×2 个的控制寄存器
- 扩展了地址空间
在普通模式，可使用与 H8/300 CPU 相同的 64K 字节的地址空间
在高级模式，可使用最大为 16M 字节的地址空间
- 加强了寻址方式
能有效的使用 16M 字节的地址空间
- 加强了指令
加强了位操作指令的寻址方式
增加了带符号的乘除法指令
增加了 2 位移位、2 位循环指令
增加了多个寄存器的保存/恢复指令
增加了 test and set 指令
- 高速化
将基本指令的执行速率提高了 2 倍。

2.1.3 与 H8/300H CPU 的区别

H8S/2000 CPU 相对于 H8/300H CPU 增加和扩展了以下功能

- 扩展了控制寄存器
增加了8位×1个的控制寄存器
- 加强了指令
加强了位操作指令的寻址方式
增加了2位移位、2位循环指令
增加了多个寄存器的保存/恢复指令
增加了test and set指令
- 高速化
将基本指令2倍高速化

2.2 CPU 运行模式

在 H8S/2000 CPU 有普通模式和高级模式 2 种运行模式。在普通模式，可支持的地址空间最大为 64K 字节，在高级模式，可支持的地址空间最大为 16M 字节。根据模式管脚决定运行模式。

2.2.1 普通模式

在普通模式，异常处理向量表、堆栈的结构与 H8/300 CPU 相同。

- 地址空间
可线性存取最大为 64K 字节的地址空间。
- 扩展寄存器 (En)
扩展寄存器 (E0~E7) 可被用作为 16 位的寄存器、也可作为 32 位寄存器的高 16 位使用。
扩展寄存器 En 作为 16 位寄存器时，可设定任意值，即使对应的通用寄存器 Rn 作为地址寄存器使用时。(请注意，如果通用寄存器 Rn 在引用先减寄存器间接寻址方式 (@-ERn)、或后增寄存器间接寻址方式 (@ERn+) 时，会产生进位/借位，这样，则会对相应的扩展寄存器 En 的内容产生影响)。
- 指令系统
可使用所有指令及寻址方式。仅对有效地址 (EA) 的低 16 位有效。
- 异常处理向量表及存储器间接转移地址
在普通模式，将高位区从 H'0000 开始的区域分配给异常处理向量表，每 16 位保存一个转移地址。普通模式的异常处理向量表结构如图 2.1 所示。有关异常处理向量表详细情况请参照 4. 异常处理。存储器间接寻址 (@@aa:8) 可在 JMP 及 JSR 指令时使用。根据指令码包含的 8 位绝对地址，指定存储器上的操作数，将此内容作为转移处的地址。在普通模式，操作数为 16 位 (字)，则此 16 位成为转移处的地址。此外，可保存转移地址的范围为 H'0000~H'00FF，请注意，此区域也会用来存储异常处理向量表。
- 堆栈结构
在普通模式，子程序转移时的 PC 堆栈构造和异常处理时，PC 和 CCR、EXR 的堆栈构造如图 2.2 所示。EXR 在中断控制模式 0 时不会压入堆栈。中断控制模式的详细内容请参考 4. 异常处理。

【注】 在本 LSI 不能使用普通模式。

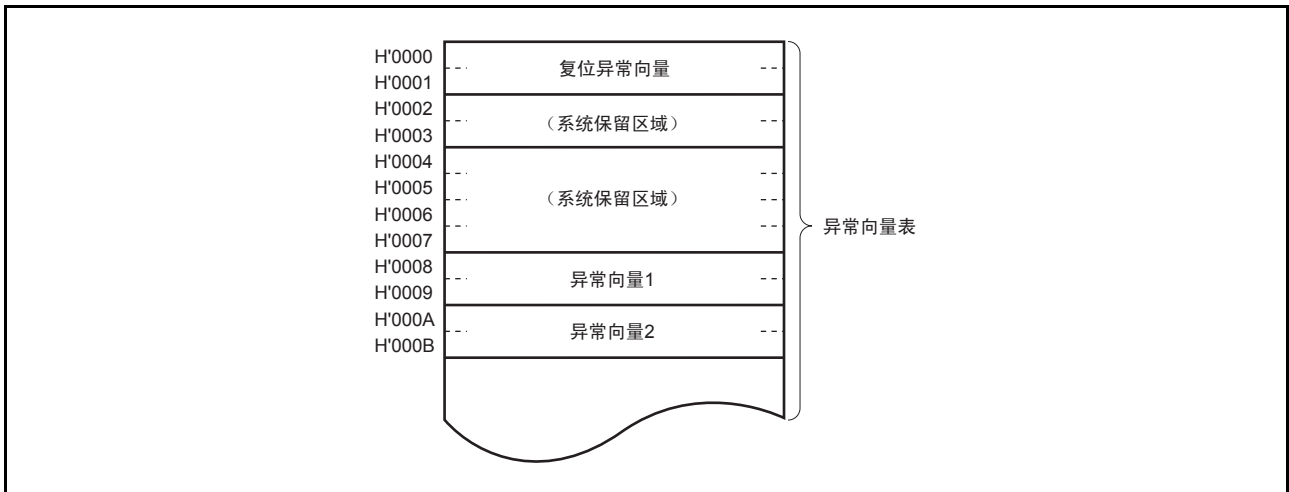


图 2.1 异常处理向量表（普通模式）

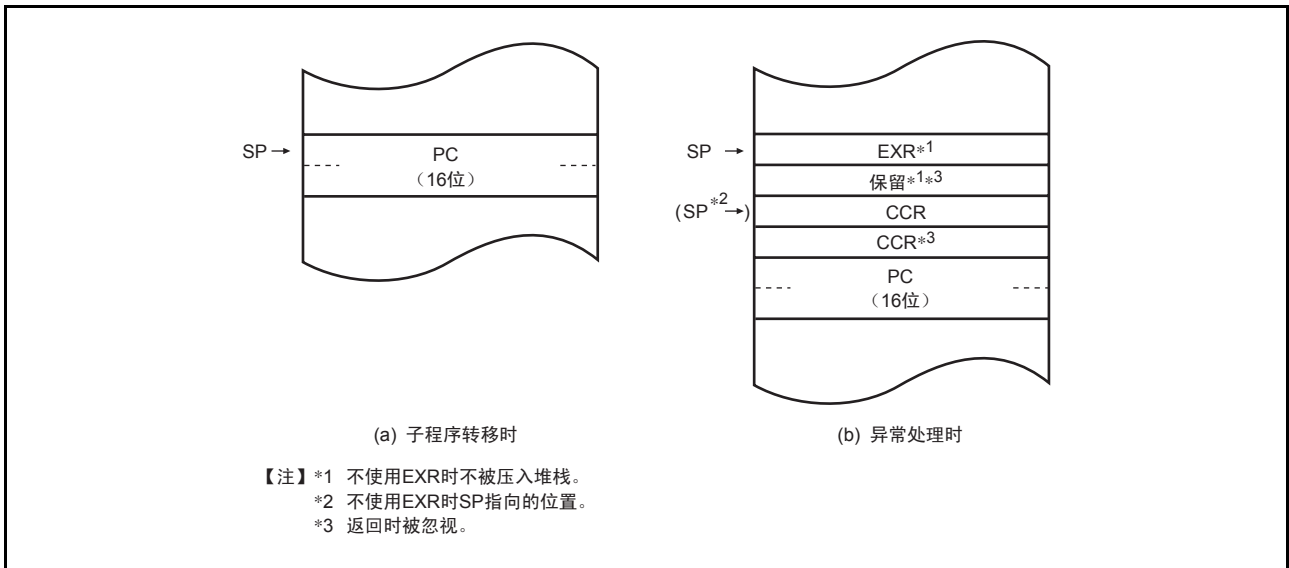


图 2.2 普通模式的堆栈构造

2.2.2 高级模式

- 地址空间
可线性存取最大为 16M 字节的地址空间。
- 扩展寄存器 (En)
扩展寄存器 (E0~E7) 可作为 16 位寄存器、32 位寄存器或地址寄存器的高 16 位使用。
- 指令设置
可使用所有指令及寻址方式。
- 异常处理向量表、存储器间接转移地址
在高级模式，以 32 位为单位，在高位区域从 H'00000000 开始分配给异常向量表。在每个 32 位的区域中，忽略高 8 位，将转移地址存储在低 24 位中（参考图 2.3）。有关异常向量表的细节请参考 4. 异常处理。

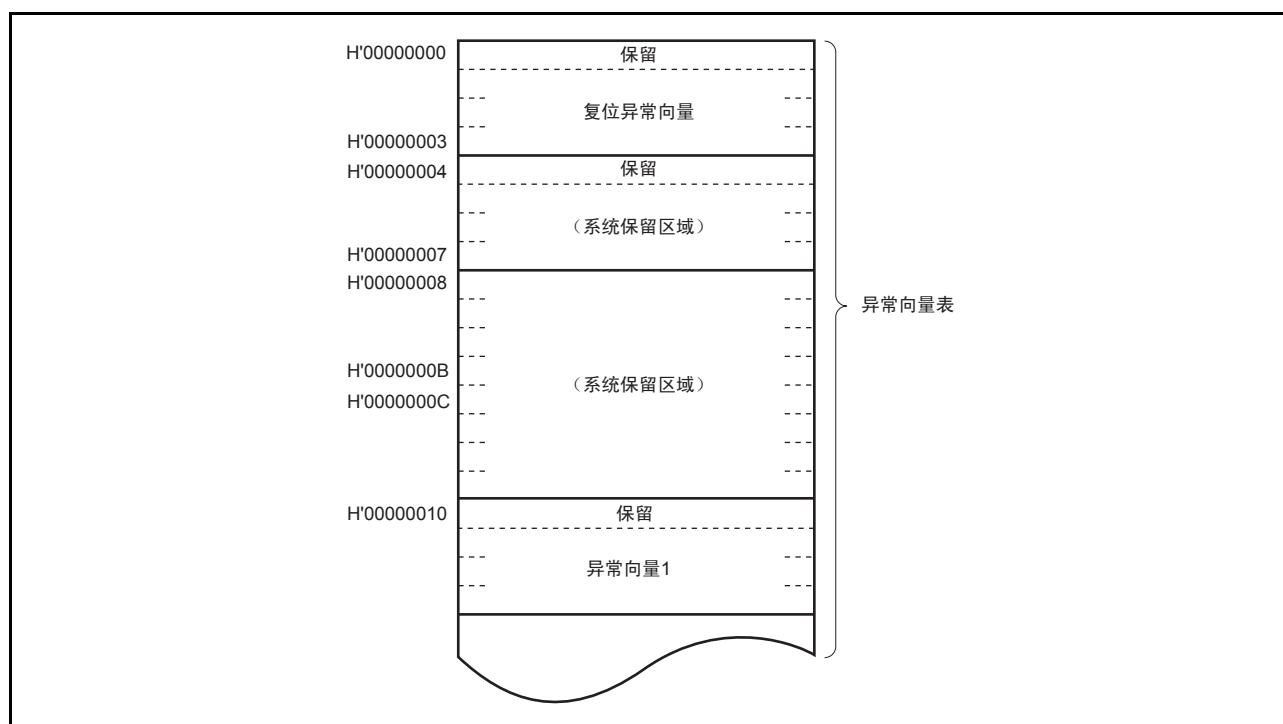


图 2.3 异常处理向量表（高级模式）

存储器间接寻址 (@@aa:8) 在 JMP 及 JSR 指令时使用。根据指令码包含的 8 位绝对地址来指定存储器上的操作数，将此内容作为转移处的地址。

在高级模式，操作数为 32 位（长字），则此 32 位成为转移处的地址。其中，高 8 位作为预留区域，被视为 H'00。此外，转移地址可保存的范围为 H'00000000 ~ H'000000FF，请注意，此区域也会用来存储异常处理向量表。

- 堆栈构成

在高级模式下，当响应子程序时，将 PC 入栈，和在异常处理时将 PC、CCR、EXR 入栈，它们的存储方式如图 2.4 所示。EXR 在中断控制模式 0 时不会压入堆栈。中断控制模式的详细内容请参考 4. 异常处理。

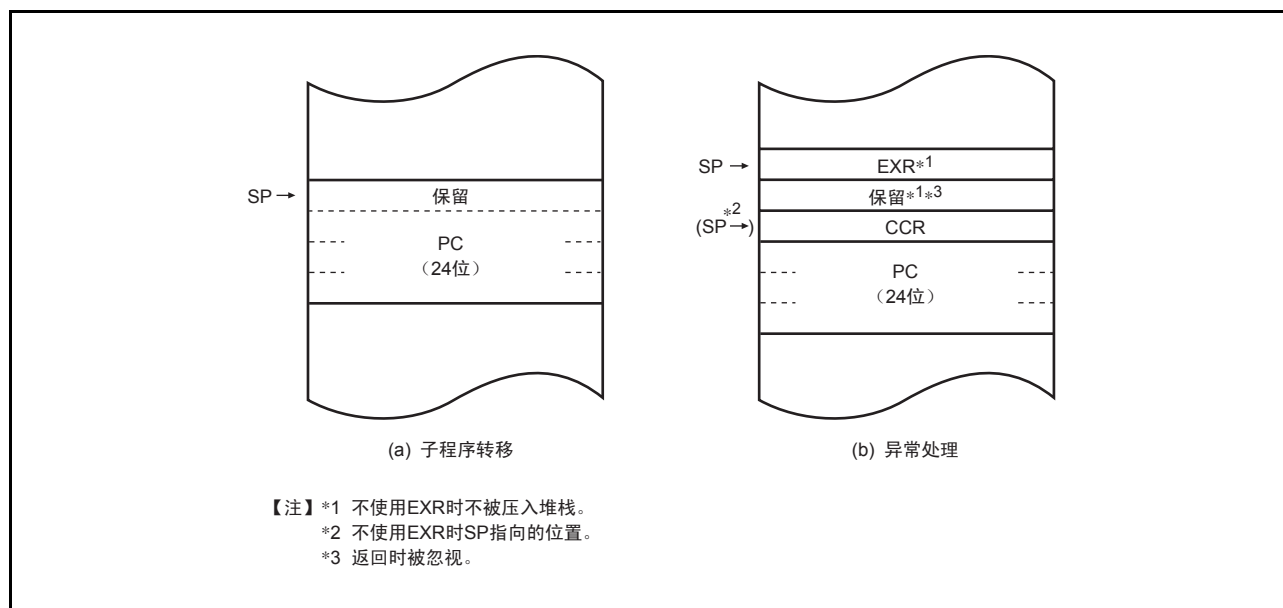


图 2.4 高级模式的堆栈构造

2.4 寄存器结构

H8S/2000 CPU 内部寄存器结构如图 2.6 所示。这些寄存器分为通用寄存器和控制寄存器二类。在控制寄存器中有 24 位程序计数器（PC）、8 位扩展控制寄存器（EXR）、8 位条件码寄存器（CCR）。

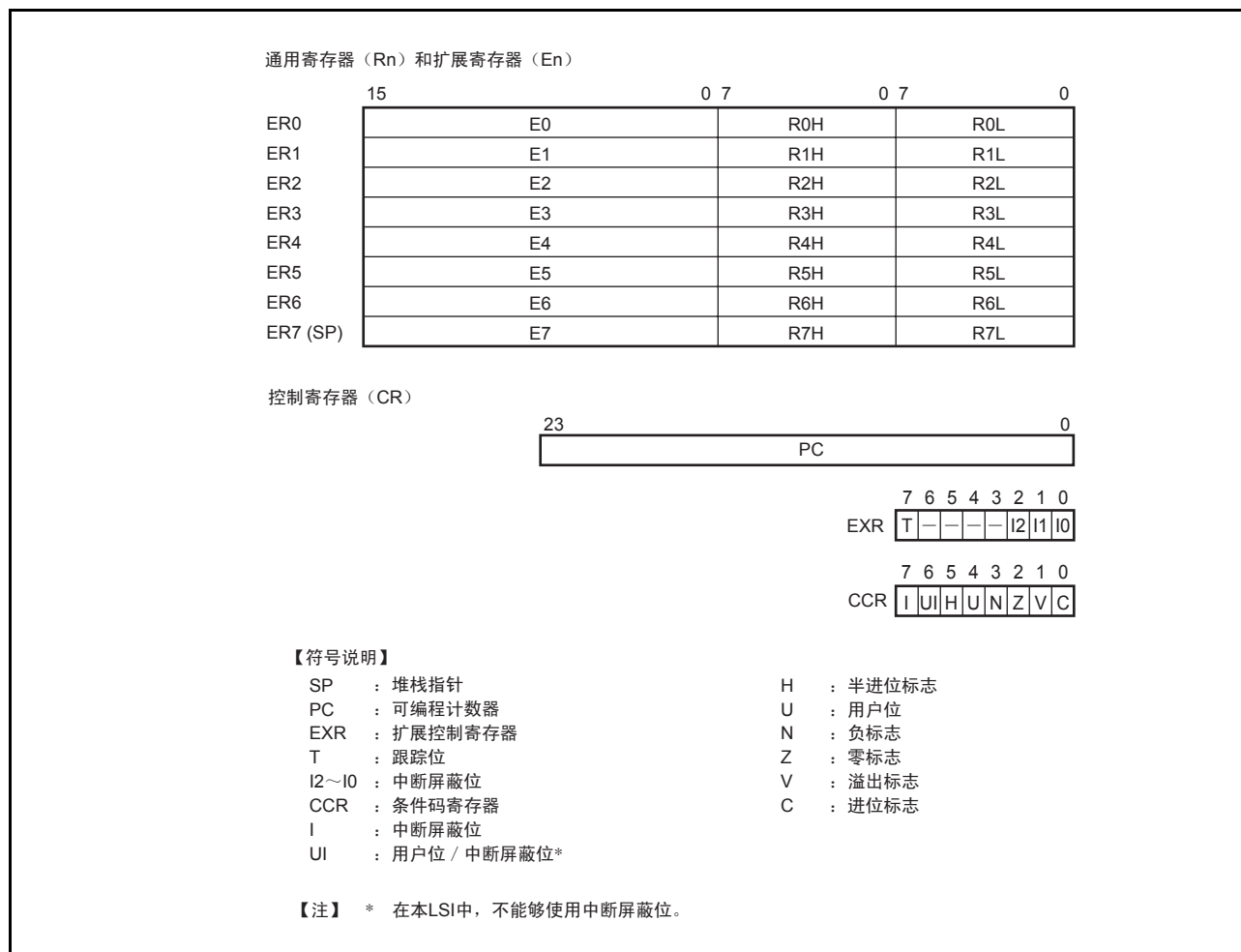


图 2.6 CPU 内部寄存器构成

2.4.1 通用寄存器

H8S/2000 CPU 有 8 个 32 位的通用寄存器。他们具有相同的功能，也可作为地址寄存器或数据寄存器。当作为数据寄存器使用时，可作为 32 位、16 位或 8 位寄存器使用。通用寄存器的使用方法如图 2.7 所示。

当作为地址寄存器或 32 位寄存器使用时，均指定为通用寄存器 ER (ER0 ~ ER7)。

作为 16 位寄存器使用时，通用寄存器 ER 分为通用寄存器 E (E0 ~ E7)、通用寄存器 R (R0 ~ R7)。它们具有相同功能，并可最多使用 16 个 16 位寄存器。并且，有时通用寄存器 E (E0 ~ E7) 会被归类为扩展寄存器。

作为 8 位寄存器使用时，通用寄存器 R 分为通用寄存器 RH (R0H ~ R7H)、通用寄存器 RL (R0L ~ R7L)。它们具有相同的功能，并可最多使用 16 个 8 位寄存器。

各寄存器可独立选择使用方法。

通用寄存器 ER7 除了通用寄存器功能外，还增加了堆栈指针 (SP) 被隐含地用在异常处理和子程序转移等。堆栈状态如图 2.8 所示。

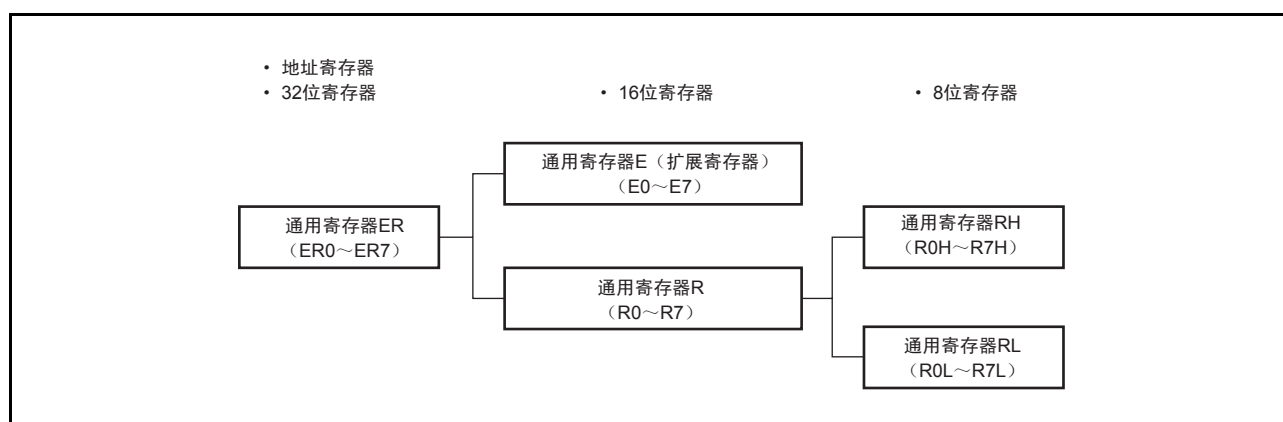


图 2.7 通用寄存器的使用方法

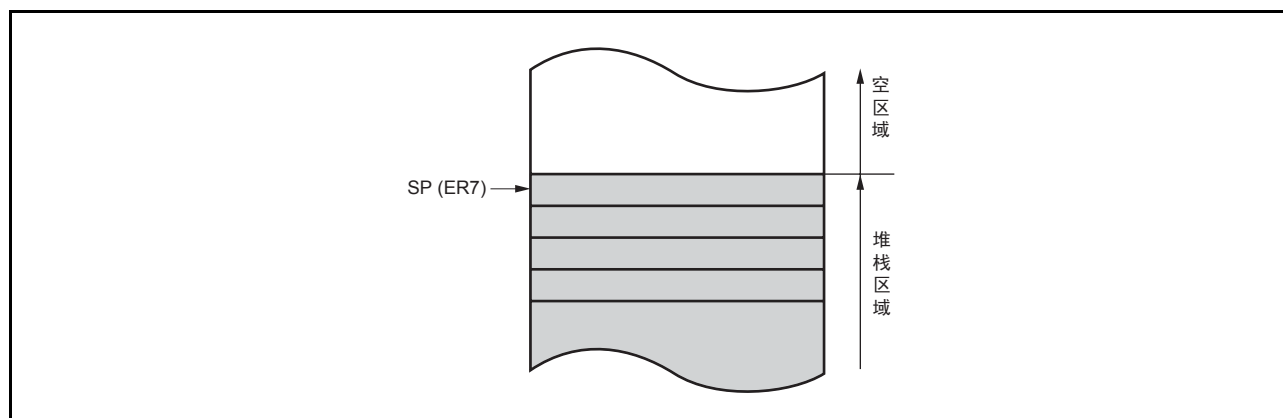


图 2.8 堆栈状态

2.4.2 程序计数器 (PC)

此寄存器为 24 位计数器，指定 CPU 将要执行下一条指令地址。由于 CPU 的指令都是以 2 字节（一个字）为单位，所以忽略最低位（在读指令时，最低位被视作 0）。

2.4.3 扩展控制寄存器 (EXR)

EXR 是一个处理 8 位，能执行 LDC、STC、ANDC、ORC、XORC 指令的寄存器。执行除 STC 之外的指令时，所有指令执行完成后的三种状态，包含 NMI 在内的所有中断将被屏蔽。

位	位名	初始值	R/W	说明
7	T	0	R/W	跟踪位 将此位置 1 时，每执行一条指令，就会产生跟踪异常处理。清 0 时，依次执行指令。
6~3	—	全为 1	—	保留位 读时通常读出 1。
2~0	I2 I1 I0	1 1 1	R/W R/W R/W	指定中断请求屏蔽级别（0 ~ 7）。详细内容请参考 5. 中断控制器。

2.4.4 条件码寄存器 (CCR)

此寄存器为 8 位的寄存器，表示 CPU 的内部状态信息。包含中断屏蔽位 (I)、半进位 (H)、负位 (N)、零位 (Z)、溢出位 (V)、进位 (C) 等 8 位标志位。LDC、STC、ANDC、ORC、XORC 等指令可操作 CCR 位。而 N、Z、V、C 等标志位可用作条件转移指令 (Bcc) 使用。

位	位名	初始值	R/W	说明
7	I	1	R/W	中断屏蔽位 如果本位置 1，中断被屏蔽。但是，不管 I 位是否设置 NMI 都不受其影响。开始执行异常处理时本位被置 1。详细内容请参照 5. 中断控制器。
6	UI	不定	R/W	用户位 / 中断屏蔽位 可通过软件 (LDC、STC、ANDC、ORC、XORC 指令) 进行读 / 写。在本 LSI 中，不可作为中断屏蔽位使用。
5	H	不定	R/W	半进位标志 执行 ADD.B、ADDX.B、SUB.B、SUBX.B、CMP.B、NEG.B 指令时，位 3 有进位或借位时置 1，否则清 0。执行 ADD.W、SUB.W、CMP.W、NEG.W 指令时，如果位 11 有进位或借位，标志位 H 置 1，否则清 0。或者执行 ADD.L、SUB.L、CMP.L、NEG.L 指令时，如果位 27 有进位或借位时置 1，否则清 0。
4	U	不定	R/W	用户位 可通过软件 (LDC、STC、ANDC、ORC、XORC 命令) 进行读 / 写。
3	N	不定	R/W	负标志 数据的最高位被视为符号位，保存最上位的值。
2	Z	不定	R/W	零标志 数据为零时置 1，否则为 0。
1	V	不定	R/W	溢出标志 执行算术运算指令时，溢出产生时置 1。否则为 0。
0	C	不定	R/W	进位标志 执行运算有进位产生时置 1，否则为 0。进位有以下种类： <ul style="list-style-type: none"> • 加法运算的进位 • 减法运算的借位 • 移位 / 循环的进位 另外，进位标志有位累加器功能，可用位操作指令执行。

2.4.5 CPU 内部寄存器初始值

在 CPU 内部寄存器中，PC 通过复位异常处理从向量地址装入起始地址进行初始化。并且，EXR 的 T 位清零，EXR、CCR 的 I 位置 1，但是，通用寄存器和 CCR 的其他位不初始化。SP (ER7) 的初始化也不定。因此在复位后，用 MOV.L 指令进行 SP 的初始化。

2.5 数据格式

H8S/2000 CPU 可处理 1 位、4 位 BCD、8 位（字节）、16 位（字）、及 32 位（长字）数据。

1 位数据通过位操作指令执行，以操作数（字节）的第 n 位 ($n = 0, 1, 2, \dots, 7$) 的格式进行访问。

此外，DAA 及 DAS 的 10 进制校正指令，字节数据成为 2 个 4 位 BCD 数据。

2.5.1 通用寄存器的数据格式

通用寄存器的数据格式如图 2.9 所示。

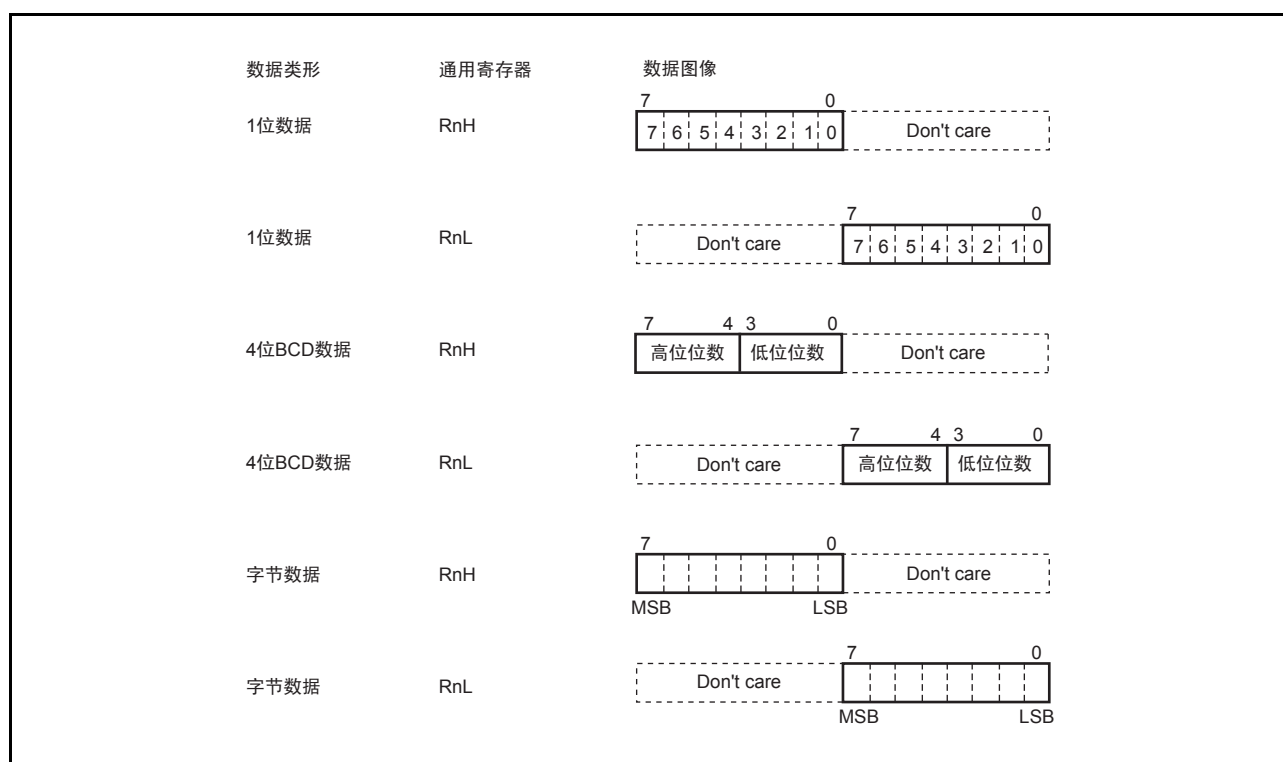


图 2.9 通用寄存器的数据格式 (1)

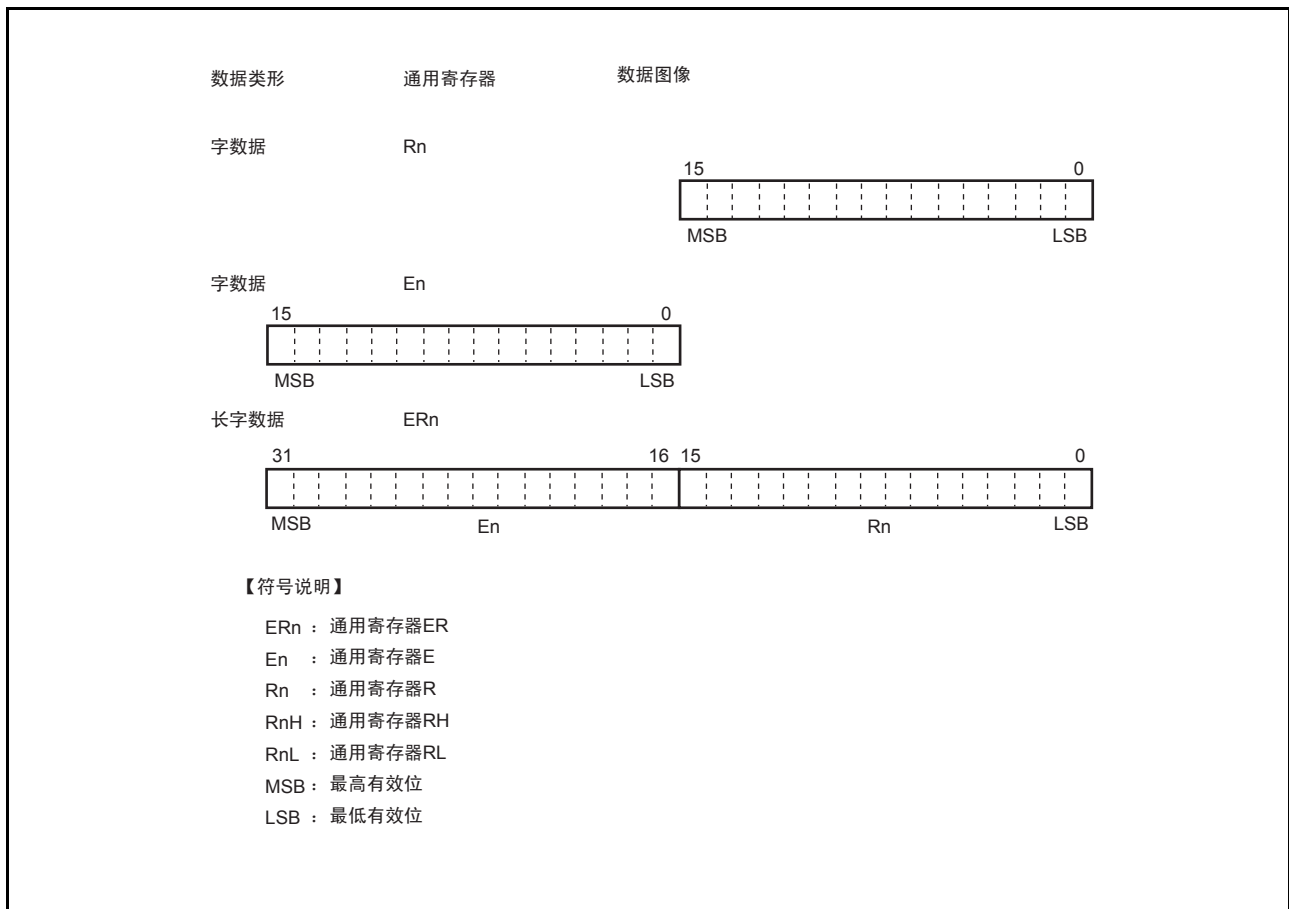


图 2.9 通用寄存器的数据格式 (2)

2.5.2 存储器的数据格式

存储器的数据格式如图 2.10 所示。

H8S/2000 CPU 可对存储器的字数据 / 长字数据进行存取操作。这些数据必须是从偶数地址开始的数据。如果存取从奇数地址开始的字数据 / 长字数据，地址的最低位被视为 0，存取从前一个地址开始的数据。此时，不产生地址操作错误。指令代码的获取也相同。

把 SP (ER7) 作为地址寄存器进行堆栈区域的存取操作时，必须以字大小或长字大小进行存取操作。

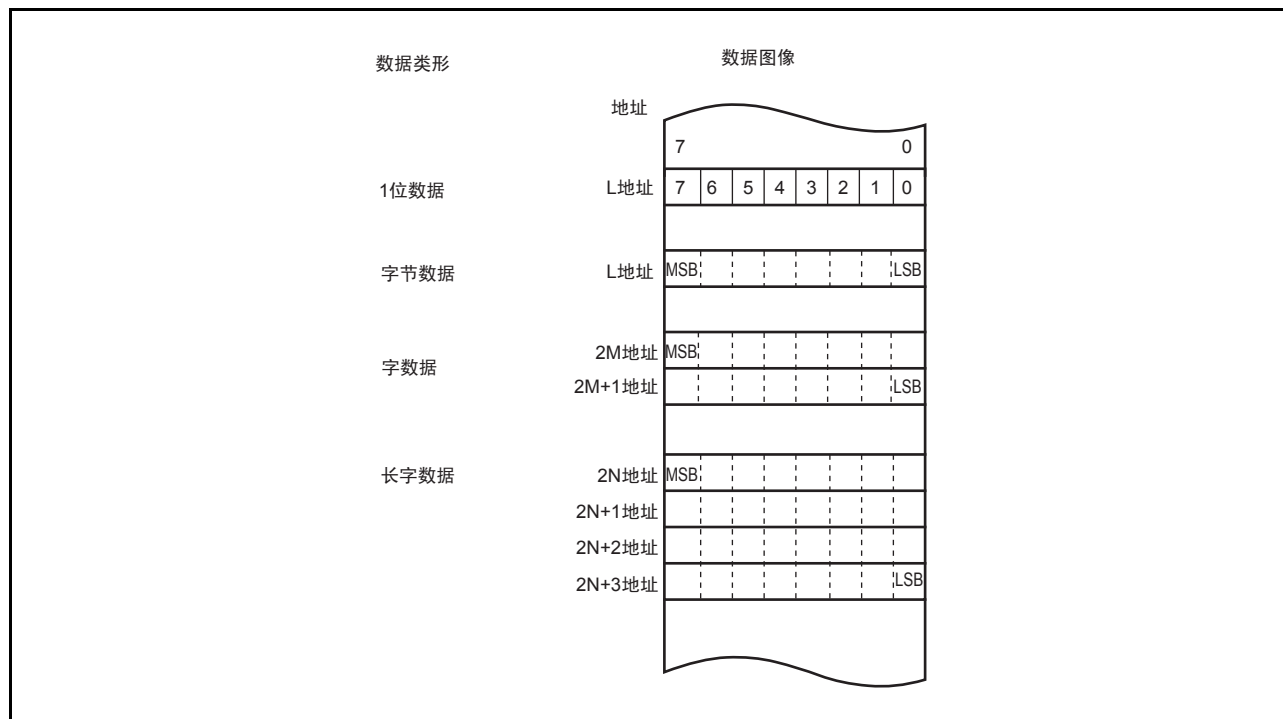


图 2.10 存储器的数据格式

2.6 指令系统

H8S/2000 CPU 共有 65 种指令，按照各指令的功能将指令分类，如表 2.1 所示。

表 2.1 指令分类

分类	指令	大小	种类
数据传送指令	MOV	B / W / L	5
	POP *1,PUSH *1	W / L	
	LDM*5,STM*5	L	
	MOVFPE*3,MOVTPE*3	B	
算术运算指令	ADD,SUB,CMP,NEG	B / W / L	19
	ADDX,SUBX,DAA,DAS	B	
	INC,DEC	B / W / L	
	ADDS,SUBS	L	
	MULXU,DIVXU,MULXS,DIVXS	B / W	
	EXTU,EXTS	W / L	
	TAS*4	B	
逻辑运算指令	AND,OR,XOR, NOT	B / W / L	4
移位指令	SHAL,SHAR,SHLL,SHLR,ROTL,ROTR,ROTXL,ROTXR	B / W / L	8
位操作指令	BSET,BCLR,BNOT,BTST,BLD,BILD,BST,BIST,BAND, BIAND,BOR,BIOR,BXOR,BIXOR	B	14
转移指令	Bcc*2, JMP, BSR, JSR, RTS	—	5
系统控制指令	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	—	9
块传送指令	EEPMOV	—	1

合计 65 种

【符号说明】

- B: 字节大小
- W: 字大小
- L: 长字大小

【注】 *1 POP.W Rn、PUSH.W Rn 分别和 MOV.W @SP+,Rn、MOV.W Rn,@-SP 相同。
并且、POP.L ERn、PUSH.L ERn 分别和 MOV.L @SP+,ERn、MOV.L ERn,@-SP 相同。

*2 Bcc 为条件转移指令的总称。

*3 不可用于本 LSI。

*4 使用 TAS 指令时，请使用寄存器 ER0、ER1、ER4、ER5。

*5 关于 STM/LDM 指令，因为 ER7 寄存器为堆栈指针，所以不可作为可进行保存（STM）/返回（LDM）寄存器使用。

2.6.1 指令功能一览

各类指令功能表 2.3 ~ 表 2.10 所示。各操作符号的意义如下表所示。

表 2.2 操作符号

符号	说明
Rd	通用寄存器（目标）*
Rs	通用寄存器（源）*
Rn	通用寄存器*
ERn	通用寄存器（32 位寄存器）
(EAd)	目标操作数
(EAs)	源操作数
EXR	扩展寄存器
CCR	条件码寄存器
N	CCR 的 N（负）标志
Z	CCR 的 Z（零）标志
V	CCR 的 V（溢出）标志
C	CCR 的 C（进位）标志
PC	程序计数器
SP	堆栈指针
#IMM	立即数据
disp	位移
+	加法
-	减法
×	乘法
÷	除法
∧	逻辑与
∨	逻辑或
⊕	逻辑异或
→	传送
~	非逻辑（逻辑补）
:8 / :16 / :24 / :32	8 / 16 / 24 / 32 长度

【注】 * 通用寄存器有 8 位（R0H ~ R7H、R0L ~ R7L）、16 位（R0 ~ R7、E0 ~ E7）、或 32 位寄存器（ER0 ~ ER7）。

表 2.3 数据传送指令

指令	大小*1	功能
MOV	B / W / L	(EAs) → Rd、Rs → (EAd) 在两个通用寄存器之间，或通用寄存器和存储器之间传输数据，或将立即数据传送至通用寄存器。
MOVFP	B	不可用于本 LSI。
MOVTPE	B	不可用于本 LSI。
POP	W / L	@SP+ → Rn 从堆栈弹出数据送往通用寄存器。 POP.W Rn 与 MOV.W @SP+, Rn 的功能一致，POP.L ERn 与 MOV.L @SP+, ERn 的功能一致。
PUSH	W / L	Rn → @-SP 将通用寄存器的内容保存到堆栈。 PUSH.W Rn 与 MOV.W Rn, @-SP 的功能相同。 PUSH.L ERn 与 MOV.L ERn, @-SP 的功能相同。
LDM*2	L	@SP+ → Rn (寄存器系列) 从堆栈返回数据到多个通用寄存器。
STM*2	L	Rn (寄存器系列) → @-SP 将多个通用寄存器的数据压入堆栈。

【注】 *1 指操作数大小。

B: 字节

W: 字

L: 长字

*2 在 STM/LDM 指令，ER7 寄存器是指堆栈指针，所以，不能作为可保存 (STM) / 恢复 (LDM) 的寄存器使用。

表 2.4 算数运算指令

指令	大小*1	功能
ADD SUB	B / W / L	$Rd \pm Rs \rightarrow Rd$ 、 $Rd \pm \#IMM \rightarrow Rd$ 在两个通用寄存器数据之间或在通用寄存器中立即数据和数据之间执行加减运算。（字节大小不可在通用寄存器中和立即数据之间进行减法运算。请使用 SUBX 指令或 ADD 指令）。
ADDX SUBX	B	$Rd \pm Rs \pm C \rightarrow Rd$ 、 $Rd \pm \#IMM \pm C \rightarrow Rd$ 在两个通用寄存器位数据之间或在通用寄存器中立即数据和数据之间执行带有进位的加减法。
INC DEC	B / W / L	$Rd \pm 1 \rightarrow Rd$ 、 $Rd \pm 2 \rightarrow Rd$ 通用寄存器执行加减 1 或 2。（通过字节大小只可加减 1）。
ADDS SUBS	L	$Rd \pm 1 \rightarrow Rd$ 、 $Rd \pm 2 \rightarrow Rd$ 、 $Rd \pm 4 \rightarrow Rd$ 将 32 位寄存器中的数据加减 1、2、或 4。
DAA DAS	B	$Rd(\text{十进制校正}) \rightarrow Rd$ 参考 CCR，将通用寄存器上的加减运算结果校正为 4 位 BCD 数据。
MULXU	B / W	$Rd \times Rs \rightarrow Rd$ 在两个寄存器之间执行不带符号的乘法运算。 可执行 8 位 \times 8 位 \rightarrow 16 位、16 位 \times 16 位 \rightarrow 32 位的乘法运算。
MULXS	B / W	$Rd \times Rs \rightarrow Rd$ 在两个通用寄存器之间执行带符号的乘法运算。 可执行 8 位 \times 8 位 \rightarrow 16 位、16 位 \times 16 位 \rightarrow 32 位的乘法运算。
DIVXU	B / W	$Rd \div Rs \rightarrow Rd$ 在两个通用寄存器之间执行不带符号的除法运算。 可执行 16 位 \div 8 位 \rightarrow 8 位商及 8 位余数。 32 位 \div 16 位 \rightarrow 16 位商及 16 位余数的除法运算。
DIVXS	B / W	$Rd \div Rs \rightarrow Rd$ 在两个通用寄存器之间执行带符号的除法运算。 16 位 \div 8 位 \rightarrow 8 位商余及 8 位余数 可执行 32 位 \div 16 位 \rightarrow 16 位商及 16 位余数的除法运算。
CMP	B / W / L	$Rd - Rs$ 、 $Rd - \#IMM$ 在两寄存器数据之间，或在通用寄存器数据和立即数据之间进行比较，其结果在 CCR 中反映。
NEG	B / W / L	$0 - Rd \rightarrow Rd$ 获取通用寄存器内容 2 的补码（算数补码）。
EXTU	W / L	$Rd(\text{零扩展}) \rightarrow Rd$ 对 16 位寄存器的低 8 位向字大小进行零扩展。或者，对 32 位寄存器的低 16 位向长字大小进行零扩展。
EXTS	W / L	$Rd(\text{符号扩展}) \rightarrow Rd$ 对 16 位寄存器的低 8 位向字大小进行符号扩展。或者，对 32 位寄存器的低 16 位向长字大小进行符号扩展。
TAS*2	B	$@ERd - 0, 1 \rightarrow (<\text{位 } 7 > \text{ of } @ERd)$ 测试存储器的内容后，将最高位（位 7）置 1。

【注】 *1 指操作数大小。

B: 字节

W: 字

L: 长字

*2 在执行 TAS 指令时, 请使用寄存器 ER0、ER1、ER4、ER5。

表 2.5 逻辑运算

指令	大小 *	功能
AND	B / W / L	$Rd \wedge Rs \rightarrow Rd$ 、 $Rd \wedge \#IMM \rightarrow Rd$ 在两个通用寄存器之间, 或在通用寄存器和立即数据之间进行逻辑与操作。
OR	B / W / L	$Rd \vee Rs \rightarrow Rd$ 、 $Rd \vee \#IMM \rightarrow Rd$ 在两个通用寄存器之间, 或在通用寄存器和立即数据之间进行逻辑或操作。
XOR	B / W / L	$Rd \oplus Rs \rightarrow Rd$ 、 $Rd \oplus \#IMM \rightarrow Rd$ 在两个通用寄存器之间, 或在通用寄存器和立即数据之间执行逻辑异或操作。
NOT	B / W / L	$\sim Rd \rightarrow Rd$ 获取通用寄存器内容 1 的补码。(逻辑补码)

【注】 * 长度指操作数长度。

B: 字节

W: 字

L: 长字

表 2.6 移位指令

指令	大小 *	功能
SHAL SHAR	B / W / L	$Rd(\text{移位处理}) \rightarrow Rd$ 对通用寄存器的内容执行算数移位。 可移动 1 位或 2 位。
SHLL SHLR	B / W / L	$Rd(\text{移位处理}) \rightarrow Rd$ 对通用寄存器的内容执行逻辑移位。 可移动 1 位或 2 位。
ROTL ROTR	B / W / L	$Rd(\text{循环处理}) \rightarrow Rd$ 对通用寄存器的内容执行循环移位。 可循环移动 1 位或 2 位。
ROTXL ROTXR	B / W / L	$Rd(\text{循环处理}) \rightarrow Rd$ 对通用寄存器的内容执行包含进位标志的循环移位。 可循环移动 1 位或 2 位。

【注】 * 长度指操作数长度。

B: 字节

W: 字

L: 长字

表 2.7 位操作指令

指令	大小*	功能
BSET	B	$1 \rightarrow (< \text{位编号} > \text{of} < \text{EAd} >)$ 将通用寄存器或存储器操作数中指定的某位置 1。 位编号由一个 3 位立即数据或一个通用寄存器的低 3 位来确定。
BCLR	B	$0 \rightarrow (< \text{位编号} > \text{of} < \text{EAd} >)$ 将通用寄存器或存储器的操作数中指定的某一位清零。 位编号由一个 3 位立即数据或一个通用寄存器的低 3 位来确定。
BNOT	B	$\sim (< \text{位编号} > \text{of} < \text{EAd} >) \rightarrow (< \text{位编号} > \text{of} < \text{EAd} >)$ 将通用寄存器或存储器操作数中指定的某一位取反。 位编号由一个 3 位立即数据或一个通用寄存器的低 3 位来确定。
BTST	B	$\sim (< \text{位编号} > \text{of} < \text{EAd} >) \rightarrow Z$ 测试通过寄存器或存储器操作数中指定的某一位并将结果反映到零标志位。 位编号由一个 3 位立即数据或一个通用寄存器的低 3 位来确定。
BAND	B	$C \wedge (< \text{位编号} > \text{of} < \text{EAd} >) \rightarrow C$ 对通用寄存器或存储器操作数中指定的某一位和进位标志进行逻辑与操作，并将结果保存在进位标志。
BIAND	B	$C \wedge (\sim (< \text{位编号} > \text{of} < \text{EAd} >)) \rightarrow C$ 将通用寄存器或存储器操作数中指定的某一位取反，并和进位标志进行逻辑与操作，将结果保存在进位标志位中。 位编号由一个 3 位立即数据来确定。
BOR	B	$C \vee (< \text{位编号} > \text{of} < \text{EAd} >) \rightarrow C$ 对通用寄存器或存储器操作数中指定的某一位和进位标志进行逻辑或操作，并将结果保存在进位标志。
BIOR	B	$C \vee (\sim (< \text{位编号} > \text{of} < \text{EAd} >)) \rightarrow C$ 将通用寄存器或存储器操作数中指定的某一位取反，并和进位标志进行逻辑或操作，将结果保存在进位标志位中。 位编号由一个 3 位立即数据来确定。
BXOR	B	$C \oplus (< \text{位编号} > \text{of} < \text{EAd} >) \rightarrow C$ 对通用寄存器或存储器操作数中指定的某一位和进位标志位进行逻辑异或操作， 并将结果保存在进位标志。
BIXOR	B	$C \oplus (\sim (< \text{位编号} > \text{of} < \text{EAd} >)) \rightarrow C$ 将通用寄存器或存储器操作数中指定的某一位取反，并与进位标志位进行逻辑异或运算，并将结果保存在进位标志位中。 位编号由一个 3 位立即数据来确定。

指令	大小*	功能
BLD	B	(< 位编号 >of<EAd>) → C 将通用寄存器或寄存器操作数中指定的某一位传送到进位标志位中。
BILD	B	~ (< 位编号 >of<EAd>) → C 将通用寄存器或寄存器操作数中指定的某一位取反，并传送到进位标志位中。 位编号由一个 3 位立即数据来确定。
BST	B	C → (< 位编号 >of<EAd>) 将进位标志位的内容传送到通用寄存器或寄存器操作数中指定的某一位。
BIST	B	~ C → (< 位编号 >of<EAd>) 将进位标志位的内容取反后，传送到通用寄存器或寄存器操作数中指定的某一位。 位编号由一个 3 位立即数确定。

【注】 * 长度指操作数长度。

B: 字节

表 2.8 转移指令

指令	类型	功能																																																			
Bcc	—	<p>当指定条件成立时，向已指定的地址转移，转移条件如下表所示。</p> <table border="1"> <thead> <tr> <th>助记符</th> <th>说明</th> <th>转移条件</th> </tr> </thead> <tbody> <tr> <td>BRA(BT)</td> <td>Always(True)</td> <td>Always</td> </tr> <tr> <td>BRN(BF)</td> <td>Never(False)</td> <td>Never</td> </tr> <tr> <td>BHI</td> <td>High</td> <td>C ∨ Z=0</td> </tr> <tr> <td>BLS</td> <td>Low or Same</td> <td>C ∨ Z=1</td> </tr> <tr> <td>BCC(BHS)</td> <td>Carry Clear(High or Same)</td> <td>C=0</td> </tr> <tr> <td>BCS(BLO)</td> <td>Carry Set(LOW)</td> <td>C=1</td> </tr> <tr> <td>BNE</td> <td>Not Equal</td> <td>Z=0</td> </tr> <tr> <td>BEQ</td> <td>Equal</td> <td>Z=1</td> </tr> <tr> <td>BVC</td> <td>oVerflow Clear</td> <td>V=0</td> </tr> <tr> <td>BVS</td> <td>oVerflow Set</td> <td>V=1</td> </tr> <tr> <td>BPL</td> <td>PLus</td> <td>N=0</td> </tr> <tr> <td>BMI</td> <td>Minus</td> <td>N=1</td> </tr> <tr> <td>BGE</td> <td>Greater or Equal</td> <td>N⊕V=0</td> </tr> <tr> <td>BLT</td> <td>Less Than</td> <td>N⊕V=1</td> </tr> <tr> <td>BGT</td> <td>Greater Than</td> <td>Z ∨ (N⊕V)=0</td> </tr> <tr> <td>BLE</td> <td>Less or Equal</td> <td>Z ∨ (N⊕V)=1</td> </tr> </tbody> </table>	助记符	说明	转移条件	BRA(BT)	Always(True)	Always	BRN(BF)	Never(False)	Never	BHI	High	C ∨ Z=0	BLS	Low or Same	C ∨ Z=1	BCC(BHS)	Carry Clear(High or Same)	C=0	BCS(BLO)	Carry Set(LOW)	C=1	BNE	Not Equal	Z=0	BEQ	Equal	Z=1	BVC	oVerflow Clear	V=0	BVS	oVerflow Set	V=1	BPL	PLus	N=0	BMI	Minus	N=1	BGE	Greater or Equal	N⊕V=0	BLT	Less Than	N⊕V=1	BGT	Greater Than	Z ∨ (N⊕V)=0	BLE	Less or Equal	Z ∨ (N⊕V)=1
助记符	说明	转移条件																																																			
BRA(BT)	Always(True)	Always																																																			
BRN(BF)	Never(False)	Never																																																			
BHI	High	C ∨ Z=0																																																			
BLS	Low or Same	C ∨ Z=1																																																			
BCC(BHS)	Carry Clear(High or Same)	C=0																																																			
BCS(BLO)	Carry Set(LOW)	C=1																																																			
BNE	Not Equal	Z=0																																																			
BEQ	Equal	Z=1																																																			
BVC	oVerflow Clear	V=0																																																			
BVS	oVerflow Set	V=1																																																			
BPL	PLus	N=0																																																			
BMI	Minus	N=1																																																			
BGE	Greater or Equal	N⊕V=0																																																			
BLT	Less Than	N⊕V=1																																																			
BGT	Greater Than	Z ∨ (N⊕V)=0																																																			
BLE	Less or Equal	Z ∨ (N⊕V)=1																																																			
JMP	—	无条件转移到指定地址。																																																			
BSR	—	转移到指定地址的子程序。																																																			
JSR	—	转移到指定地址的子程序。																																																			
RTS	—	从子程序返回。																																																			

表 2.9 系统控制指令

指令	大小*	功能
TRAPA	—	进行陷阱指令的异常处理。
RTE	—	从异常处理程序返回。
SLEEP	—	转移到低功耗模式。
LDC	B / W	(EAs) → CCR、(EAs) → EXR 将通用寄存器或存储器内容传送至 CCR、EXR 寄存器中。并且将立即数据传送到 CCR、EXR 寄存器中。虽然 CCR、EXR 寄存器为 8 位，但是通过字长度将 CCR、EXR 寄存器和存储器之间的数据传送。高 8 位为有效位。
STC	B / W	CCR → (EAd)、EXR → (EAd) 将 CCR、EXR 寄存器的内容传送到通用寄存器或存储器中。虽然 CCR、EXR 寄存器为 8 位，但是通过字长度将 CCR、EXR 寄存器和存储器之间的数据传送。高 8 位为有效位。
ANDC	B	CCR ∧ #IMM → CCR、EXR ∧ #IMM → EXR 对 CCR、EXR 和立即数进行逻辑与操作。
ORC	B	CCR ∨ #IMM → CCR、EXR ∨ #IMM → EXR 对 CCR、EXR 和立即数进行逻辑或操作。
XORC	B	CCR ⊕ #IMM → CCR、EXR ⊕ #IMM → EXR 对 CCR、EXR 和立即数进行逻辑异或操作。
NOP	—	PC+2 → PC 仅进行 PC 的增量。

【注】 * 长度指操作数长度。

B: 字节

W: 字

表 2.10 块传送指令

指令	大小	功能
EPPMOV.B	—	if R4L ≠ 0 then Repeat @ER5+ → @ER6+ R4L - 1 → R4L Until R4L = 0 else next;
EPPMOV.W	—	if R4 ≠ 0 then Repeat @ER5+ → @ER6+ R4 - 1 → R4 Until R4 = 0 else next;
块传送指令。从 ER5 所示的地址开始，将 R4L 或 R4 中所指定的字节数的数据传送到 ER6 所示的地址处。传送结束后，执行下一条指令。		

2.6.2 基本指令格式

H8S/2000 CPU 的指令以 2 字节（1 字）为单位。各指令由操作字段（op）、寄存器字段（r）、EA 扩展字段（EA）、及条件字段（cc）构成。

指令格式的例子如图 2.11 所示。

- 操作字段
表示指令功能、指定寻址方式和操作数的执行内容。通常包含指令的起始 4 位。有些指令有 2 个操作字段。
- 寄存器字段
指定通用寄存器。地址寄存器为 3 位，数据寄存器为 3 位或 4 位。有些指令有 2 个寄存器字段，有些没有寄存器字段。
- EA 有效扩展字段
8 位、16 位或 32 位指定立即数、绝对地址或偏移量。
- 条件字段
指定条件转移指令 Bcc 的转移条件。

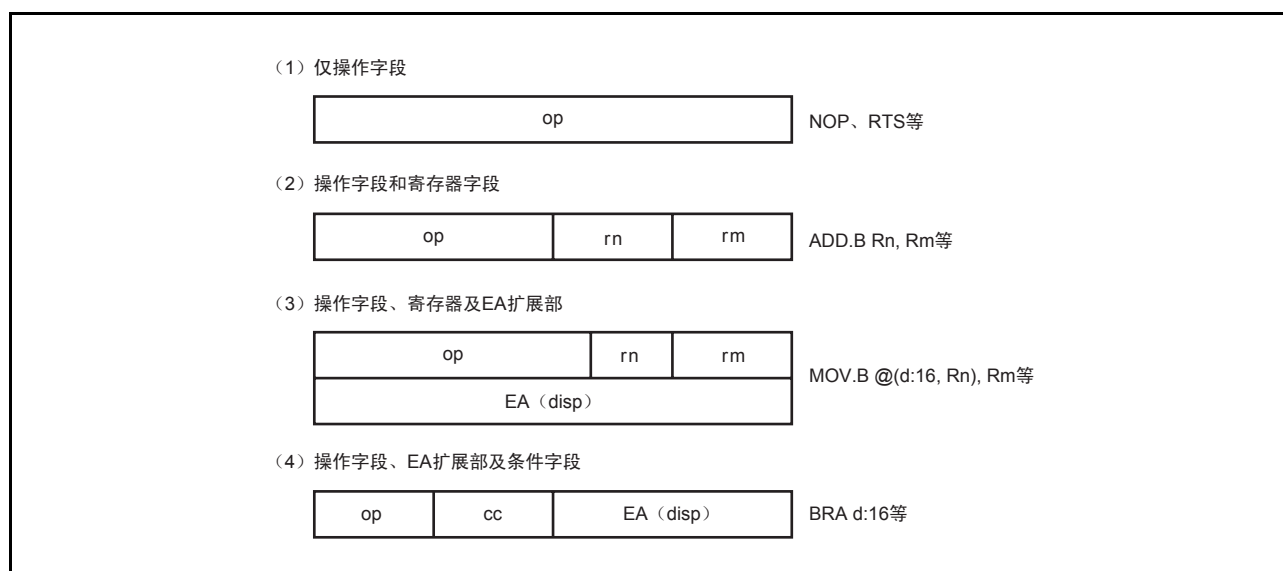


图 2.11 指令格式例子

2.7 寻址模式和有效地址的计算方法

H8S/2000 CPU 支持表 2.11 所示的 8 种寻址方式。根据各指令的不同，可使用的寻址方式也有所不同。

运算指令可使用寄存器直接及立即数寻址方式。传送指令除了程序计数器相对寻址和存储器间接寻址以外，所有的寻址方式都可使用。位操作指令可使用寄存器直接、寄存器间接以及绝对寻址，并且可使用寄存器直接（BSET、BCLR、BNOT、BTST 的各指令）和立即数（3 位）寻址方式来指定操作数中的位编号。

表 2.11 寻址模式一览表

No.	寻址模式	记号
1	寄存器直接寻址	Rn
2	寄存器间接寻址	@ERn
3	带位移量的寄存器间接寻址	@(d:16,ERn) / @(d:32,ERn)
4	后增寄存器间接寻址 先减寄存器间接寻址	@ERn+ @-ERn
5	绝对地址寻址	@aa:8 / @aa:16 / @aa:24 / @aa:32
6	立即寻址	#xx:8 / #xx:16 / #xx:32
7	程序计数器相对寻址	@(d:8,PC) / @(d:16,PC)
8	存储器间接寻址	@@aa:8

2.7.1 寄存器直接寻址 Rn

以指令码的寄存器字段指定的寄存器包含操作数（8 位、16 位或 32 位）。作为 8 位寄存器可指定 R0H ~ R7H、R0L ~ R7L。作为 16 位寄存器可指定 R0 ~ R7、E0 ~ E7。作为 32 位寄存器可指定 ER0 ~ ER7。

2.7.2 寄存器间接寻址 @ERn

以指令码的寄存器字段指定的地址寄存器（ERn）包含存储器操作数的地址。作为程序指令地址其低 24 位为有效位，高 8 位全部为 0（H'00）。

2.7.3 带位移量的寄存器间接寻址 @(d:16,ERn) / @(d:32,ERn)

将指令中包含的 16 位或 32 位位移量与指令码的寄存器字段指定的地址寄存器的内容相加，其结果作为存储器操作数的地址，当相加时，16 位位移量被符号扩展。

2.7.4 后增寄存器间接寻址 @ERn+ / 先减寄存器间接寻址 @-ERn

2.7.4.1 后增寄存器间接寻址 @ERn+

将指令码的寄存器字段指定的地址寄存器（ERn）的内容作为指定存储器上的操作数地址。在操作数存取后，地址寄存器的内容加 1、2 或 4，并将结果保存至地址寄存器中。其中字节大小加 1、字大小加 2、长字转移大小加 4。当转移指令是字大小或长字大小时，地址寄存器的内容应为偶数。

2.7.4.2 先减寄存器间接寻址 @-ERn

将指令码的寄存器字段指定的地址寄存器（ERn）内容减 1、减 2 或减 4，其结果作为存储器操作数的地址。该结果也存入地址寄存器。其中字节大小减 1、字大小减 2、长字大小减 4。当转移指令是字大小或长字大小时，地址寄存器的内容应为偶数。

2.7.5 绝对地址 @aa:8 / @aa:16 / @aa:24 / @aa:32

指令码中包含有一个存储器操作数的绝对地址。绝对地址可以是 8 位 (@aa:8)、16 位 (@aa:16)、24 位 (@aa:24) 或 32 位 (@aa:32)。可访问的绝对地址的范围如表 2.12 所示。

数据区域的操作中，其绝对地址应为 8 位 (@aa:8)、16 位 (@aa:16)、或 32 位 (@aa:32)。8 位绝对地址时，高 24 位全部为 1 (H'FFFF)。16 位绝对地址时，高 16 位为符号扩展。32 位绝对地址时，可访问整个地址空间。

对于程序区域的操作，使用 24 位 (@aa:24)。高 8 位全部为 0 (H'00)。

表 2.12 绝对地址访问范围

绝对地址		普通模式	高级模式
数据地址	8 位 (@aa:8)	H'FF00 ~ H'FFFF	H'FFFF00 ~ H'FFFFFF
	16 位 (@aa:16)	H'0000 ~ H'FFFF	H'000000 ~ H'007FFF、 H'FF8000 ~ H'FFFFFF
	32 位 (@aa:32)		H'000000 ~ H'FFFFFF
程序区域	24 位 (@aa:24)		

【注】 * 不可使用本 LSI。

2.7.6 立即数寻址 #xx:8 / #xx:16 / #xx:32

将指令码中含有 8 位 (#xx:8)、16 位 (#xx:16) 或 32 位 (#xx:32) 立即数直接作为操作数使用。

ADDS、SUBS、INC、DEC 指令在指令码中暗含了立即数。有些位操作指令在指令码中包含有指定位编号的 3 位立即数。TRAPA 指令在指令码中包含指定向量地址的 2 位立即数。

2.7.7 程序计数器相对寻址 @ (d:8, PC) / @ (d:16, PC)

这种寻址方式在 Bcc、BSR 指令中使用。指令码中包含的 8 位或 16 位位移量加上 24 位的 PC 内容生成一个 24 位转移地址。加法运算执行时，位移量被符号扩展为 24 位，加法运算结果的低 24 位有效，高 8 位全部为 0 (H'00)。与位移量相加后的 PC 内容是下一条指令的起始地址，所以可转移的范围为 -126 ~ +128 字节 (-63 ~ +64 字) 或者是 -32766 ~ +32768 字节 (-16383 ~ +16384 字)。此时，加法运算结果为偶数。

2.7.8 存储器间接寻址 @@aa:8

这种寻址方式在 JMP、JSR 指令时使用。指令码中包含一个指定存储器上的操作数的 8 位绝对地址。该操作数作为一个转移地址进行转移。由于 8 位绝对地址的高位全部为 0，因此转移地址的范围是 0 ~ 255（普通模式时为 H'0000 ~ H'00FF、高级模式时为 H'000000 ~ H'0000FF）。

在普通模式，存储器的操作数指定为字大小，而转移地址为 16 位。在高级模式，存储器操作数指定为长字操作数。操作数的首字节全部为 0（H'00）。

请注意，可保存转移地址的起始区域也是异常处理向量区域。详细内容请参考 4. 异常处理。

对于用字大小或长字大小指定存储器及指定转移地址时，存储器地址为奇数，那么最低位为 0，数据或指令码从前一个地址开始访问。（请参考 2.5.2 存储器的数据格式）

【注】 * 不可用于本 LSI。

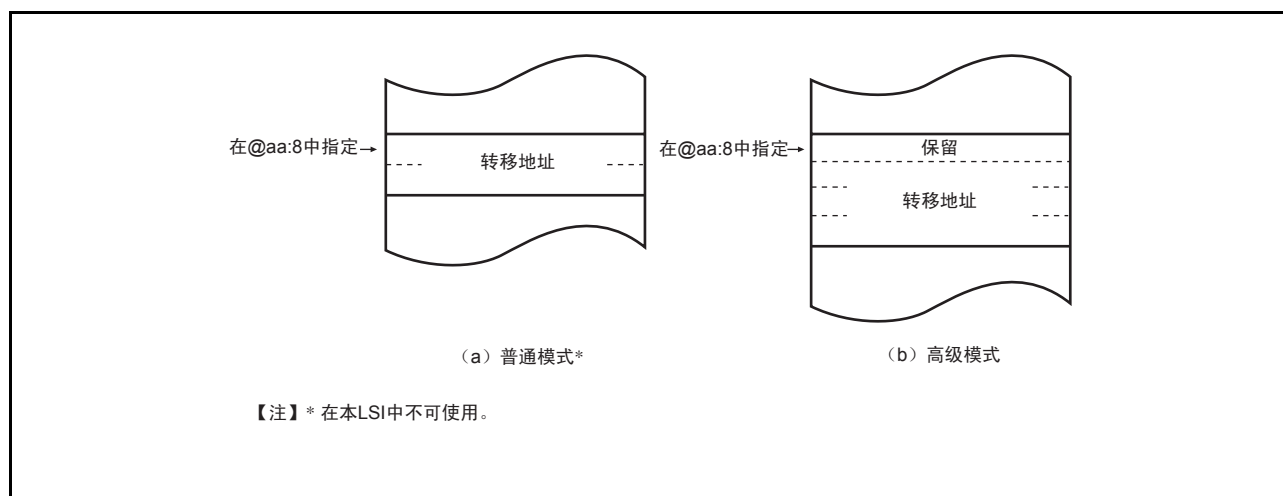


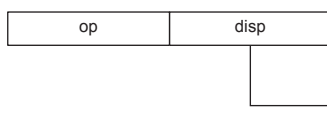
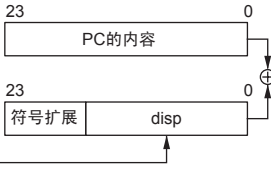

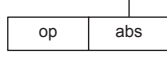
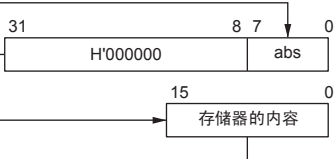
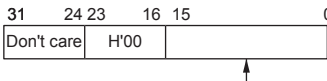
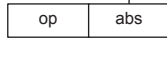
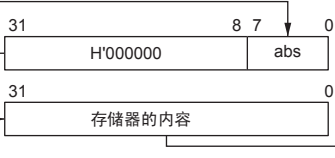
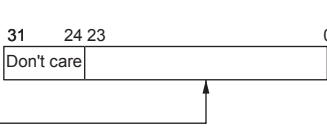
图 2.12 存储器间接转移地址规定

2.7.9 有效地址的计算方法

各种寻址方式的有效地址（EA: Effective Address）计算方法如表 2.13 所示。
 在普通模式，有效地址的高 8 位被忽略，生成一个 16 位地址。

表 2.13 有效地址的计算方法

No	寻址模式·命令格式	有效地址的计算方法	有效地址 (EA)								
1	寄存器直接寻址 (Rn) 		操作数是通用寄存器的内容。								
2	寄存器间接 (@ERn) 										
3	带位移量寄存器间接寻址 @(d:16,ERn) / @(d:32,ERn) 										
4	后增寄存器间接 / 先减寄存器间接寻址 • 后增寄存器间接寻址 @ERn+ • 先减寄存器间接寻址 @-ERn 	<table border="1" style="margin-top: 10px;"> <thead> <tr> <th>操作数长度</th> <th>加减运算得到的值</th> </tr> </thead> <tbody> <tr> <td>字节</td> <td>1</td> </tr> <tr> <td>字</td> <td>2</td> </tr> <tr> <td>长字</td> <td>4</td> </tr> </tbody> </table>	操作数长度	加减运算得到的值	字节	1	字	2	长字	4	
操作数长度	加减运算得到的值										
字节	1										
字	2										
长字	4										
5	绝对地址寻址 @aa:8 @aa:16 @aa:24 @aa:32 										
6	立即寻址 #xx:8 / #xx:16 / #xx:32 		操作数是立即数。								

No	寻址模式・指令格式	有效地址的计算方法	有效地址 (EA)
7	程序计数器相对寻址 @(d:8,PC) / @(d:16,PC) 		
8	存储器间接寻址@@aa:8 ・普通模式* 		
	・高级模式 		

【注】* 在本LSI中不能使用。

2.8 处理状态

H8S/2000 CPU 的处理状态有：复位状态、异常处理状态、程序执行状态、总线释放状态及低功耗状态 5 种。处理状态间的状态转移图如图 2.13 所示。

- 复位状态**
 CPU 以及内置外围模块全部初始化，成为停止状态。复位管脚为低电平时，中止所有执行中的处理，CPU 处于复位状态。复位状态屏蔽所有的中断操作。复位管脚从低电平到高电平时，开始执行复位异常处理。复位的详细信息请参考 4. 异常处理。内置看门狗定时器的产品，由于看门狗定时器的溢出也可进行复位操作。
- 异常处理状态**
 异常处理状态是当 CPU 因为异常源，比如复位、跟踪、中断或陷阱指令而改变正常程序执行流程，从异常处理向量表中获得起始地址（向量），并向其起始地址转移而产生的一个过渡状态。详细内容请参考 4. 异常处理。
- 程序执行状态**
 CPU 按顺序执行程序指令。
- 总线释放状态**
 对于内置 DMA 控制器（DMAC）和数据传输控制器（DTC）等，为响应 CPU 以外的主线主控器的总线权发出请求，就进入总线释放状态。处于总线释放状态时，CPU 操作停止。
- 低功耗状态**
 CPU 运行停止时，处于低功耗状态。当执行 SLEEP 指令或迁移到硬件待机模式，CPU 成为程序停止状态。详细内容请参照 20. 低功耗状态。

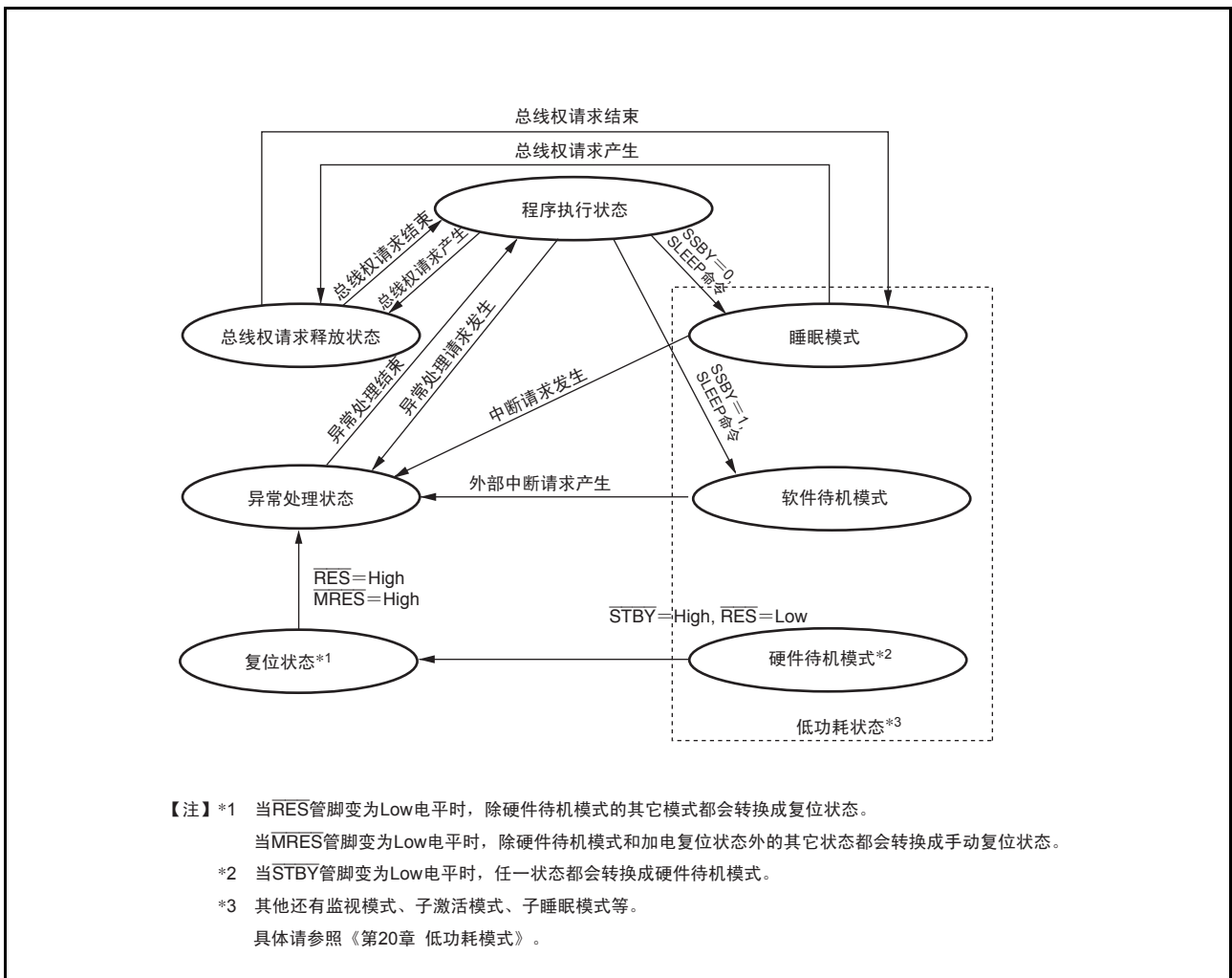


图 2.13 状态迁移图

2.9 使用注意事项

2.9.1 TAS 操作指令使用注意事项

执行 TAS 操作指令时，仅能使用寄存器 ER0、ER1、ER4、ER5。

并且，瑞萨科技 H8S、H8/300 系列 C/C++ 编译器不生成 TAS 操作指令。作为客户定义的嵌入函数使用 TAS 操作指令时，请仅使用寄存器 ER0、ER1、ER4、ER5。

2.9.2 STM/LDM 操作指令使用注意事项

使用 STM/LDM 操作指令，ER7 寄存器被用作堆栈指针，因此就不能作为允许保存（STM）/恢复（LDM）操作的寄存器使用。

使用单指令 STM 或 LDM，2 ~ 4 个寄存器能够被保存 / 恢复。

可使用的寄存器如下所示。

2 个：ER0 - ER1、ER2 - ER3、ER4 - ER5

3 个：ER0 - ER2、ER4 - ER6

4 个：ER0 - ER3

瑞萨科技的 H8S、H8/300 系列 C/C++ 编译器不能生成包括 ER7 的 STM/LDM 操作指令。

2.9.3 位操作指令使用注意事项

对含有只写位寄存器使用位操作指令时，会发生不按预期或在错误的位操作的情况。

从含有写专用位的寄存器中读的数据会返回为固定值或不定值。随之，使用读取的值进行运算的位操作指令（BNOT、BTST、BAND、BIAND、BOR、BIOR、BXOR、BIXOR、BLD、BILD）不会正确工作。

而且，基于数据值读取的伴随写的位操作指令（BSET、BCLR、BNOT、BST、BIST）会产生与位操作无关的位值变化。因此，对于包含只写位的寄存器，在使用位操作指令时请特别注意。

BSET、BCLR、BNOT、BST、BIST 的各指令按以下顺序执行：

1. 以字节单位读数据
2. 读到的数据执行位操作指令
3. 再次以字节单位写数据

例子：通过使用 BCLR 指令，仅对端口 1 的 P1DDR 的管脚 14 进行清零。

P1DDR 是包含写专用位的 8 位寄存器。用于指定端口 1 每个管脚的输入 / 输出设置。读定义为无效。读取后读出不定值。

此例中，表示通过 BCLR 操作指令将管脚 P14 设置为输入管脚。假定此时 P17 ~ P14 被设定成输出管脚，P13 ~ P10 被设定成输入管脚，则 P1DDR 的值是最初的 H'F0。

	P17	P16	P15	P14	P13	P12	P11	P10
输入 / 输出	输出	输出	输出	输出	输入	输入	输入	输入
P1DDR	1	1	1	1	0	0	0	0

P14 从输出管脚切换为输入管脚必须将 P1DDR 的位 4 的值从“1”变为“0”（H'F0 → H'E0）。现在，假定使用 BCLR 操作指令将 P1DDR 的位 4 清零。

BCLR #4, @P1DDR

但是，对于只写寄存器 P1DDR 进行上述位操作指令会产生如下问题。

BCLR 指令 首先以字节单位从 P1DDR 读数据，但此时读到的值是不定值。不定值在寄存器里可以是“0”或“1”，但是无法知道是哪个值。因为 P1DDR 全部为只写位，无论何位都读出不定值。此例中，P1DDR 的值为 H'F0，而当读取的寄存器的返回值为 H'F8 时，位 3 的值为 1。

	P17	P16	P15	P14	P13	P12	P11	P10
输入 / 输出	输出	输出	输出	输出	输入	输入	输入	输入
P1DDR	1	1	1	1	0	0	0	0
读取值	1	1	1	1	1	0	0	0

BCLR 指令对于读到的值执行位操作。本例中为 H'F8，它将位 4 清零。

	P17	P16	P15	P14	P13	P12	P11	P10
输入 / 输出	输出	输出	输出	输出	输入	输入	输入	输入
P1DDR	1	1	1	1	0	0	0	0
位操作后	1	1	1	0	1	0	0	0

执行位操作后，将其数据写入 P1DDR，结束 BCLR 操作指令。

	P17	P16	P15	P14	P13	P12	P11	P10
输入 / 输出	输出	输出	输出	输入	输出	输入	输入	输入
P1DDR	1	1	1	0	1	0	0	0
写入值	1	1	1	0	1	0	0	0

P1DDR 的内容本应被 H'E0 值覆盖，但是实际上 H'E8 已经被写入寄存器，使本应是输入管脚的 P13 变成输出管脚。本例中我们假定 P1DDR 中位 3 为 1，但由于读取 P1DDR 的 7~0 位时，读到的值全是不定值，就可能使每一个位操作指令结束后，从“0”变成“1”或从“1”变成“0”。为了避免上述情况的发生，在变更包括只写位寄存器的值时，请参考 2.9.4 包括写专用位的寄存器存取方法。

此外，使用 BCLR 操作指令可将内部 I/O 寄存器标志清“0”。在此情况下，只要预先明确标志已被中断处理或类似程序置“1”，就没必要读取相应的标志。

2.9.4 包括写专用位的寄存器存取方法

从包括只写位的寄存器发出数据传送指令或对包括写只写的寄存器执行位操作指令，会导致读出不确定值。为了防止读出不定值，请按照以下的方法存取包括只写位的寄存器。

写入包括只写位的寄存器时，在内置 RAM 等的存储器上设置另外的工作区域，将被操作的数据写入一次后。对存储器进行存取，请将存储器里的数据写入包括只写位的寄存器。

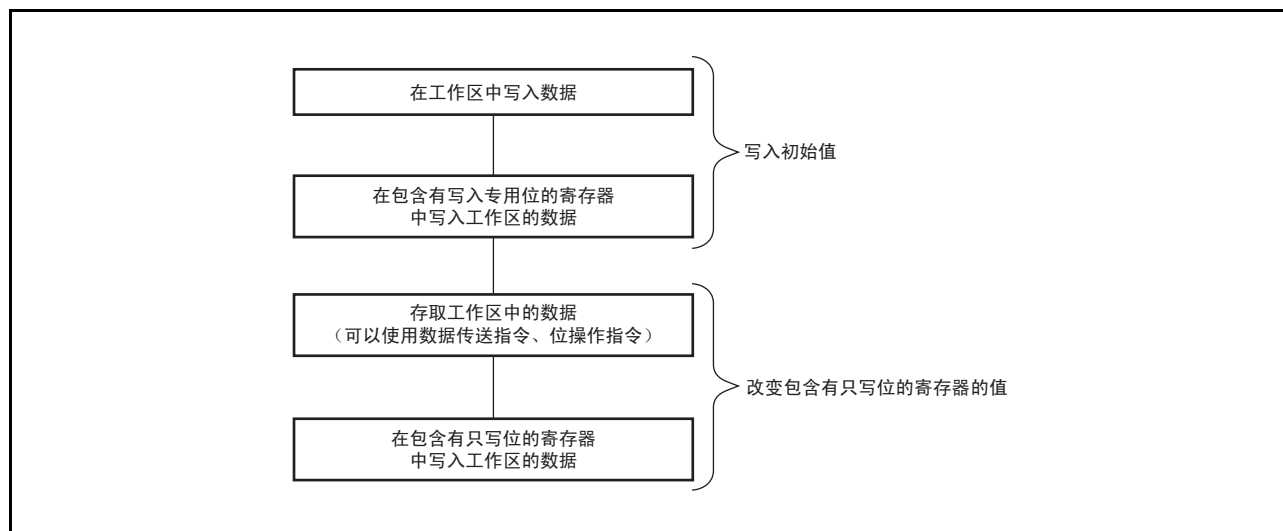


图 2.14 包括写专用位的寄存器存取方法流程图例子

例子： 仅对端口 1 的 P1DDR 的管脚 P14 进行清零

P1DDR 是包括只写位的 8 位寄存器，以用来指定端口 1 各管脚的输入 / 输出设置。读定义为无效。读后读出不定值。

本例中，BCLR 操作指令用来将 P14 管脚设置为输入端口。首先将写入 P1DDR 的初始值 H'F0 事先写入存储器里的工作区域（RAM0）。

```

MOV.B  # H'F0, R0L
MOV.B  R0L,  @RAM0
MOV.B  R0L,  @P1DDR
  
```

	P17	P16	P15	P14	P13	P12	P11	P10
输入 / 输出	输出	输出	输出	输出	输入	输入	输入	输入
P1DDR	1	1	1	1	0	0	0	0

RAM0	1	1	1	1	0	0	0	0
------	---	---	---	---	---	---	---	---

要将 P14 从输出管脚变为输入管脚，必须将 P1DDR 的 4 位的值从 “1” 变为 “0” (H'F0 → H'E0)。此处用 BCLR 指令将 P1DDR 的位 4 清零。

```
BCLR    #4, @RAM0
```

	P17	P16	P15	P14	P13	P12	P11	P10
输入 / 输出	输出	输出	输出	输出	输入	输入	输入	输入
P1DDR	1	1	1	1	0	0	0	0

RAM0	1	1	1	0	0	0	0	0
------	---	---	---	---	---	---	---	---

因为 RAM0 是存储器的一个可读 / 写区域，所以执行上述位操作指令后，仅将 RAM0 的位 4 清零。并将 RAM0 的值写入 P1DDR。

```
MOV.B   @RAM0, R0L
MOV.B   R0L,   @P1DDR
```

	P17	P16	P15	P14	P13	P12	P11	P10
输入 / 输出	输出	输出	输出	输入	输入	输入	输入	输入
P1DDR	1	1	1	0	0	0	0	0

RAM0	1	1	1	0	0	0	0	0
------	---	---	---	---	---	---	---	---

如果依上顺序对包括只写位的寄存器进行存取，可不依赖指令的种类进行编程。

3. MCU 运行模式

3.1 工作模式的选择

本 LSI 有 4 种工作模式（模式 4 ~ 7）。按照各工作模式，切换管脚功能。模式管脚（MD2 ~ MD0）的设置决定工作模式。模式 4 ~ 6 是可存取外部存储器及外围器件的外部扩展模式。程序开始运行后，外部扩展模式可以通过总线控制器给每个区分配 8 位或 16 位地址空间。另外，如果给任意一个区分配 16 位地址空间，则变为 16 位总线模式，如果给所有的区分配 8 位存取空间，则变为 8 位总线模式。

模式 7 不能使用外部地址空间。另外，在工作过程中不能使模式管脚发生变化。

H8S/2212 群只能使用模式 7。

表 3.1 MCU 工作模式的选择

MCU 工作模式	MD2	MD1	MD0	CPU 工作模式	内容	内置 ROM	外部数据	
							初始值	最大值
4	1	0	0	高级模式	On-chip ROM 无效扩展模式	无效	16 位	16 位
5	1	0	1	高级模式	On-chip ROM 无效扩展模式	无效	8 位	16 位
6	1	1	0	高级模式	On-chip ROM 有效扩展模式	有效	8 位	16 位
7	1	1	1	高级模式	单片模式	有效	—	—

【注】 使用 E6000 仿真器时

1. H8S/2218 群不能使用模式 7。（E6000 仿真器不支持模式 7。）
2. 模式 6 在使用 RTC 和 USB 时，有如下限制
 - 为了使 PB1 和 PB0 管脚输出 A9 和 A8，需设置 PFCR 寄存器。
 - 为了使 PC7~PC0 管脚输出 A7~A0，需在 PCDDR 寄存器设置 H'FF。

3.2 寄存器说明

与工作模式有关的寄存器有以下几种。

- 模式控制寄存器 (MDCR)
- 系统控制寄存器 (SYSCR)

3.2.1 模式控制寄存器 (MDCR)

MDCR 用于监控本 LSI 当前的工作模式。不能写入。

位	位名称	初始值	R/W	说明
7~4	—	不定	—	保留位 读取时，读出不定值。不能写入。
3	FWE	—*1	R	闪存写入允许 表示 FWE 管脚的输入电平。与 FLMCR1 寄存器的 FWE 位功能相同。
2	MDS2	—*1	R	模式选择 2 ~ 0 表示模式管脚 (MD2 ~ MD0) 的输入电平值 (当前运行模式)。 MDS2 ~ MDS0 位分别与 MD2 ~ MD0 管脚相对应。这些位专用于读取，写入定义为无效。读取 MDCR 时，模式管脚 (MD2 ~ MD0) 的输入电平被这些位锁存。锁存在加电复位时被解除，但在手动复位 *2 时被保持。
1	MDS1	—*1	R	
0	MDS0	—*1	R	

【注】 *1 由 FWE，MD2 ~ MD0 管脚的设置决定。

*2 仅限于 H8S/2218 群。

3.2.2 系统控制寄存器 (SYSCR)

SYSCR 执行中断控制模式的选择, NMI 检测边沿的选择, $\overline{\text{MRES}}$ 管脚 * 输入允许 / 禁止的选择, 内部 RAM 有效 / 无效的选择。

位	位名称	初始值	R/W	说明
7	—	0	R/W	保留位 写入时, 必须写入 0。
6	—	0	—	保留位 读取时, 总是读出 0。写入定义为无效。
5 4	INTM1 INTM0	0 0	R/W R/W	选择中断控制器的中断控制模式。关于中断控制模式, 请参照 5.6 中断控制模式与中断运行。 00: 中断控制模式 0 01: 禁止设置 10: 中断控制模式 2 11: 禁止设置
3	NMIEG	0	R/W	NMI 边沿选择 选择 NMI 管脚的输入边沿。 0: 在 NMI 输入的下沿, 发生中断请求 1: 在 NMI 输入的上沿, 发生中断请求
2	MRESE	0	R/W	手动复位选择位 选择允许 / 禁止 $\overline{\text{MRES}}$ 管脚 * 输入 0: 禁止手动复位 1: 允许手动复位 可以作为 $\overline{\text{MRES}}$ 输入管脚 * 使用
1	—	0	—	保留位 读取时, 总是读出 0。写入定义为无效。
0	RAME	1	R/W	RAM 允许 选择 On-chip RAM 有效或无效。复位解除时, 初始化 RAME 位。 0: On-chip RAM 无效 1: On-chip RAM 有效

【注】 * 仅限于 H8S/2218 群。

3.3 各工作模式的说明

3.3.1 模式 4（仅限于 H8S/2218 群）

CPU 在高级模式环境下，可存取 16M 字节的地址空间。On-chip ROM 无效。

P13 ~ P10 管脚，端口 A、B、C 为地址总线，端口 D、E 为数据总线，端口 F 的一部分传送总线控制信号。复位后，P13 ~ P11 管脚变为输入端口，P10 管脚，端口 A、B 变为地址（A20 ~ A8）输出。通过 PFCR 的 AE3 ~ AE0 位，可以设置允许 / 禁止地址（A23 ~ A21）输出，与相应 DDR 的值无关。相应的 DDR 置位时，P13 ~ P10 管脚，端口 A、B 中禁止地址输出的管脚变为端口输出。端口 C 通常为地址（A7 ~ A0）输出。

复位后，变为 16 位总线模式，所有区域变为 16 位存取空间。但是，通过总线控制器，将所有区域设置为 8 位存取空间时，变为 8 位总线模式。

3.3.2 模式 5（仅限于 H8S/2218 群）

CPU 在高级模式环境下，可存取 16M 字节的地址空间。On-chip ROM 无效。

P13 ~ P10 管脚，端口 A、B、C 为地址总线，端口 D、E 为数据总线，端口 F 的一部分传送总线控制信号。复位后，P13 ~ P11 管脚变为输入端口，P10 管脚，端口 A、B 变为地址（A20 ~ A8）输出。通过 PFCR 的 AE3 ~ AE0 位，可以设置允许 / 禁止地址（A23 ~ A21）输出，与相应的 DDR 的值无关。相应的 DDR 置位时，P13 ~ P10 管脚，端口 A、B 中禁止地址输出的管脚变为端口输出，端口 C 通常为地址（A7 ~ A0）输出。

复位后，变为 8 位总线模式，所有区域变为 8 位存取空间。但是，通过总线控制器，将任意区域设置为 16 位存取空间时，变为 16 位总线模式。端口 E 变为数据总线。

3.3.3 模式 6（仅限于 H8S/2218 群）

CPU 在高级模式环境下，可存取 16M 字节的地址空间。On-chip ROM 有效。复位后，P13 ~ P10 管脚，端口 A、B、C 变为输入端口。通过 PFCR 的 AE3 ~ AE0 位，可以设置允许 / 禁止地址（A23 ~ A8）输出，与相应的 DDR 的值无关。相应的 DDR 置位时，P13 ~ P10 管脚，端口 A、B 中禁止地址输出的管脚变为端口输出。

端口 C 在相应的 DDR 置位时，地址（A7 ~ A0）变为输出。

端口 D、E 变为数据总线，端口 F 的一部分传送总线控制信号

复位后，变为 8 位总线模式，所有区域变为 8 位存取空间。但是，通过总线控制器，将任意区域设置为 16 位存取空间时，变为 16 位总线模式，且端口 E 变为数据总线。

3.3.4 模式 7

CPU 在高级模式环境下，可存取 16M 字节的地址空间。On-chip ROM 有效，不能使用外部地址空间。所有的 I/O 端口都可作为输入 / 输出端口使用。

3.3.5 管脚功能

根据工作模式，来切换端口 1、A ~ F 的管脚功能。

模式 4 ~ 7 中的管脚功能一览表如表 3.2 所示。

表 3.2 各工作模式中的管脚功能

端口		模式 4	模式 5	模式 6	模式 7
端口 1	P11 ~ P13	P* / A	P* / A	P* / A	P
	P10	P / A*	P / A*	P* / A	P
端口 A	PA3 ~ PA0	P / A*	P / A*	P* / A	P
端口 B		P / A*	P / A*	P* / A	P
端口 C		A	A	P* / A	P
端口 D		D	D	D	P
端口 E		P / D*	P* / D	P* / D	P
端口 F	PF7	P / C*	P / C*	P / C*	P / C*
	PF6 ~ PF4	C	C	C	
	PF3	P / C*	P* / C	P* / C	
	PF2 ~ PF0	P* / C	P* / C	P* / C	

【符号说明】

P: 输入 / 输出端口

A: 地址总线输出

D: 数据总线输入 / 输出

C: 控制信号・时钟输入 / 输出

*: 复位后

3.4 各工作模式的地址映射

各产品的地址映射如图 3.1、~图 3.3 所示。

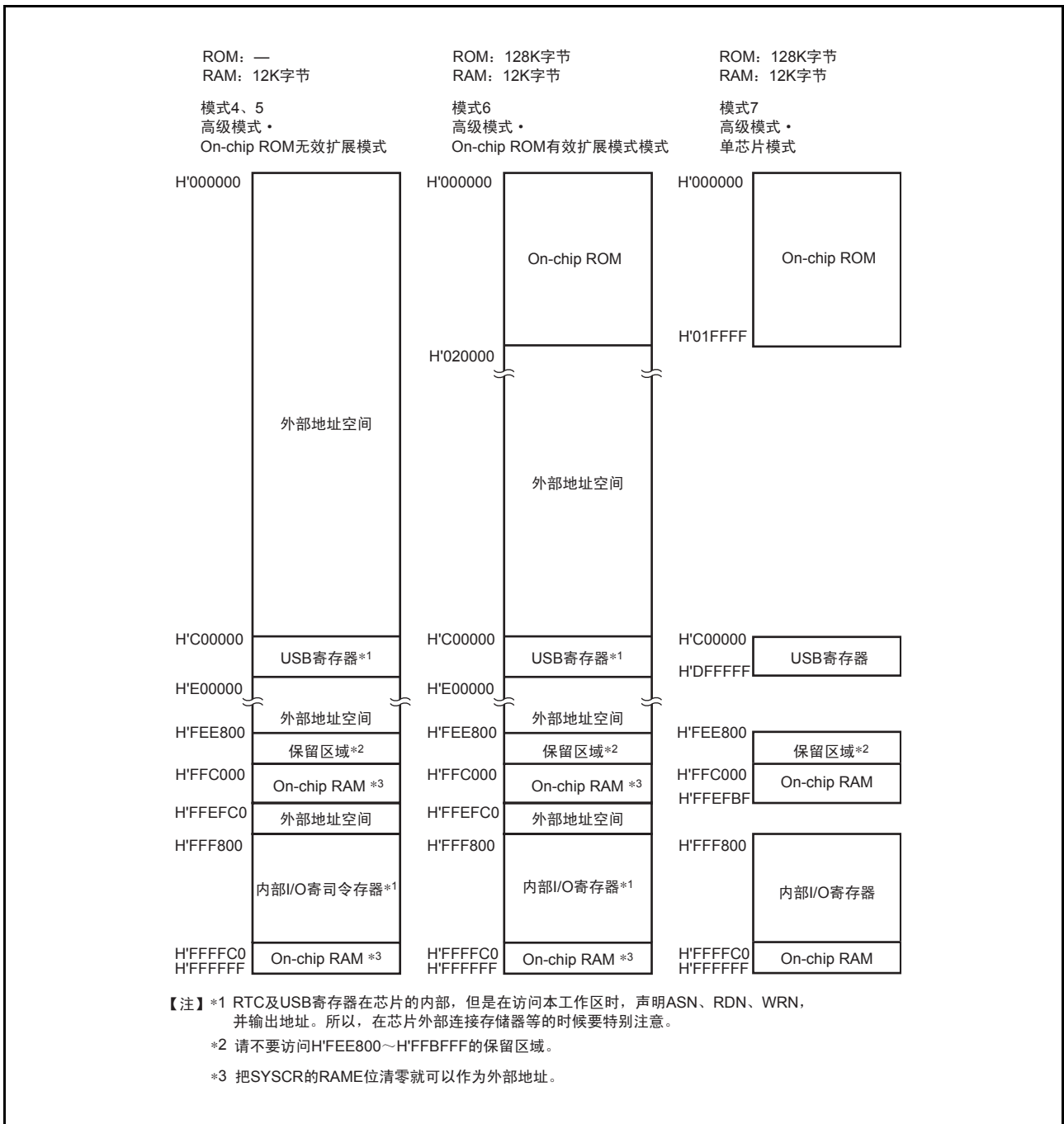


图 3.1 HD64F2218、HD64F2218U 的地址映射

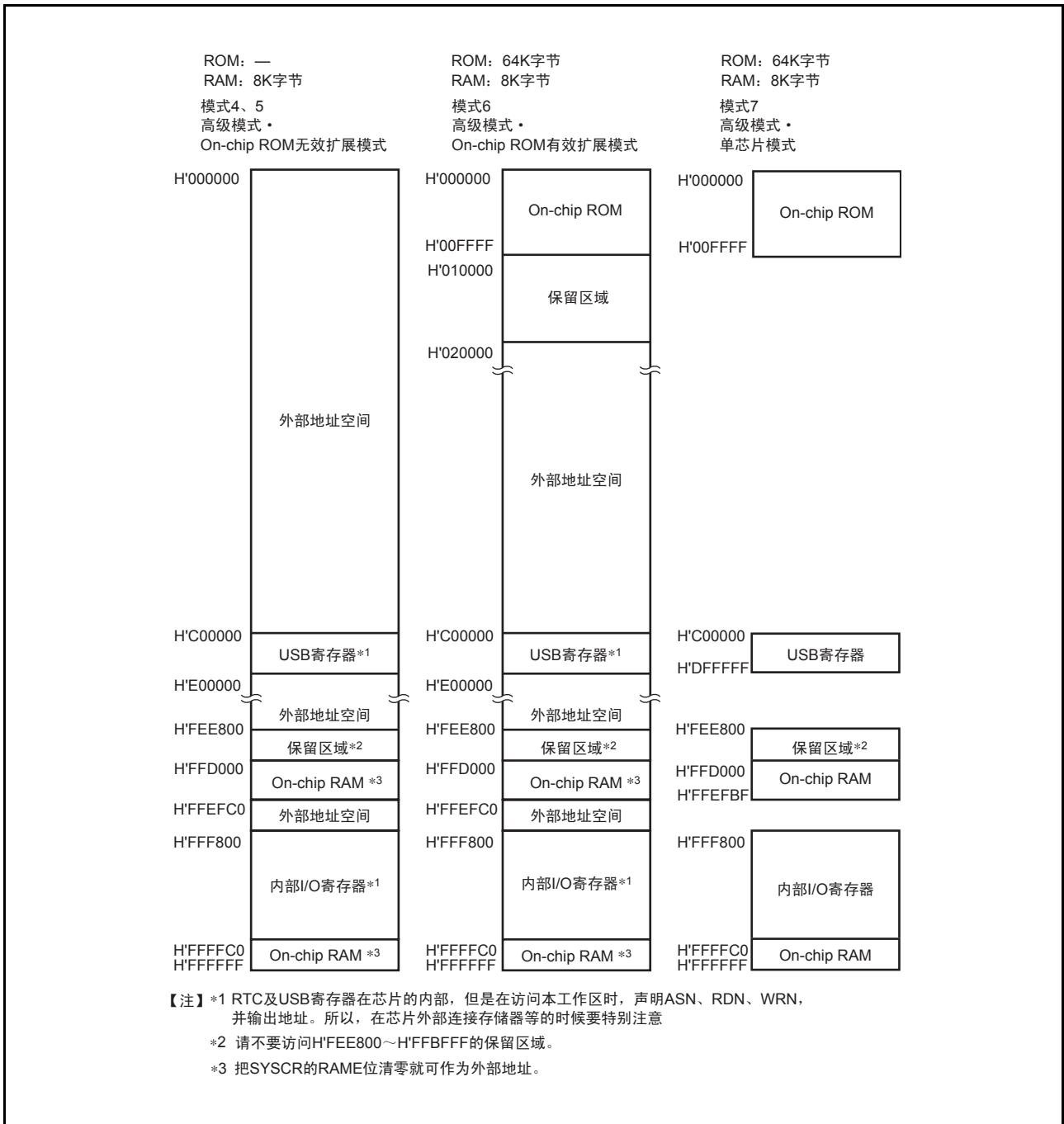


图 3.2 HD6432217 的地址映射

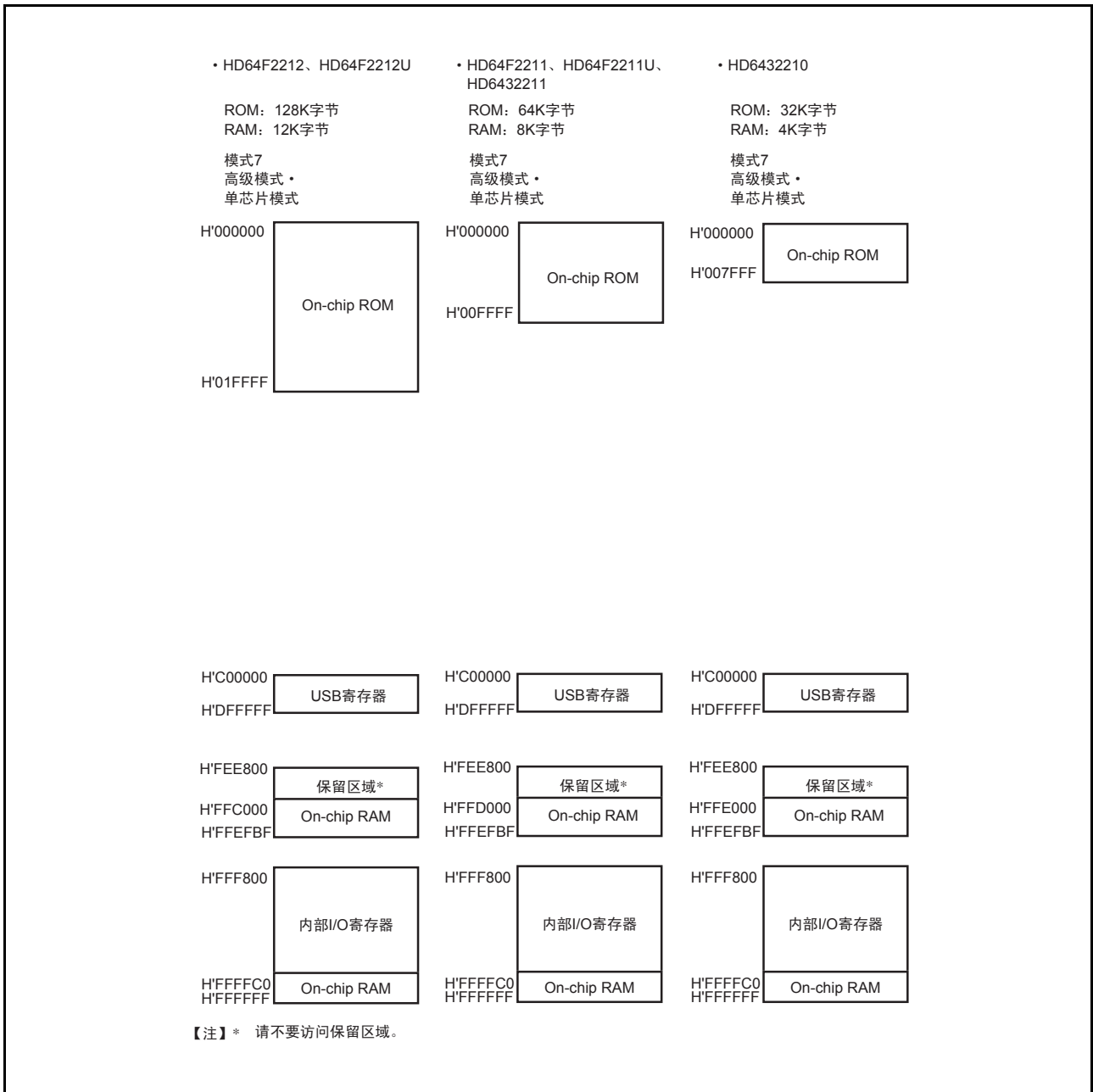


图 3.3 HD64F2212、HD64F2212U、HD64F2211、HD64F2211U、HD6432211、HD6432210 的地址映射

4. 异常处理

4.1 异常处理种类和优先级

如表 4.1 所示，异常源有复位、跟踪、中断及陷阱指令。这些异常源如表 4.1 所示，设有优先级、数个异常源同时发生时，可按此优先级进行处理。异常处理因中断控制模式不同而异常源、堆栈结构及 CPU 运行有所不同。中断控制模式的详细内容，请参考 5. 中断控制器。

表 4.1 异常处理的种类与优先级

优先级	异常处理种类	异常处理开始时许
高 ↑	复位	$\overline{\text{RES}}$ 管脚、 $\overline{\text{MRES}}$ 管脚 * 从低电平转换至高电平或由于看门狗定时器溢出，开始进行复位异常处理。 $\overline{\text{RES}}$ 管脚为低电平时，成为加电复位状态。 $\overline{\text{MRES}}$ 管脚 * 为低电平时，成为手动复位状态。
	跟踪	EXR 跟踪 (T) 位置位为 1 的状态下指令或异常处理结束时，开始进行跟踪异常处理。跟踪仅在中断控制模式 2 中有效。RTE 指令执行结束后，不进行跟踪异常处理。
	中断	发生中断请求时，指令或结束异常处理结束时，开始进行中断异常处理。但是，执行 ANDC、ORC、XORC、LDC 指令结束或复位异常处理结束时，不执行中断检测。
低	陷阱指令 (TRAPA)	通过执行陷阱 (TRAPA) 指令，开始进行陷阱指令异常处理。程序运行状态下，随时接受陷阱指令异常处理。

【注】 * 仅限 H8S/2218 群

4.2 异常源与向量表

异常源设定各种不同的向量地址。异常源与向量地址对照表如表 4.2 所示。因产品不同，可使用的模式有所不同，所以，每个产品的详细内容，请参考 3. MCU 运行模式。

表 4.2 异常源与向量地址的对照表

异常源	向量号	向量地址 *1	
		普通模式 *4	高级模式
加电复位	0	H'0000 ~ H'0001	H'0000 ~ H'0003
手动复位	1	H'0002 ~ H'0003	H'0004 ~ H'0007
系统保留	2	H'0004 ~ H'0005	H'0008 ~ H'000B
	3	H'0006 ~ H'0007	H'000C ~ H'000F
	4	H'0008 ~ H'0019	H'0010 ~ H'0013
跟踪	5	H'000A ~ H'000B	H'0014 ~ H'0017
直接转换 *2	6	H'000C ~ H'000D	H'0018 ~ H'001B
外部中断 NMI	7	H'000E ~ H'000F	H'001C ~ H'001F
陷阱指令 (#0)	8	H'0010 ~ H'0011	H'0020 ~ H'0023
陷阱指令 (#1)	9	H'0012 ~ H'0013	H'0024 ~ H'0027
陷阱指令 (#2)	10	H'0014 ~ H'0015	H'0028 ~ H'002B
陷阱指令 (#3)	11	H'0016 ~ H'0017	H'002C ~ H'002F
系统保留	12	H'0018 ~ H'0019	H'0030 ~ H'0033
	13	H'001A ~ H'001B	H'0034 ~ H'0037
	14	H'001C ~ H'001D	H'0038 ~ H'003B
	15	H'001E ~ H'001F	H'003C ~ H'003F
外部中断 IRQ0	16	H'0020 ~ H'0021	H'0040 ~ H'0043
外部中断 IRQ1	17	H'0022 ~ H'0023	H'0044 ~ H'0047
外部中断 IRQ2	18	H'0024 ~ H'0025	H'0048 ~ H'004B
外部中断 IRQ3	19	H'0026 ~ H'0027	H'004C ~ H'004F
外部中断 IRQ4	20	H'0028 ~ H'0029	H'0050 ~ H'0053
RTC 中断 IRQ5	21	H'002A ~ H'002B	H'0054 ~ H'0057
USB 中断 IRQ6	22	H'002C ~ H'002D	H'0058 ~ H'005B
外部中断 IRQ7	23	H'002E ~ H'002F	H'005C ~ H'005F
内部中断 *3	24	H'0030 ~ H'0031	H'0060 ~ H'0063
	 127	 H'00FE ~ H'00FF	 H'01FC ~ H'01FF

【注】 *1 表示起始地址低位 16 位。

*2 直接转换，请参考 20.10 直接转移。

*3 中断向量表请参照 5.5 中断异常处理向量表。

*4 本 LSI 环境中不可使用。

4.3 复位

复位为最高优先级异常处理。 $\overline{\text{RES}}$ 管脚、 $\overline{\text{MRES}}$ 管脚 * 为低电平时，可全部停止执行中的处理，本 LSI 成为复位状态。为确保本 LSI 复位，通电时，务请将 $\overline{\text{RES}}$ 管脚保持在最低 20ms 之间低电平。另外，运行中，务请将 $\overline{\text{RES}}$ 管脚或 $\overline{\text{MRES}}$ 管脚 * 保持在最低 20 个状态之间低电平。通过复位，CPU 内部状态与内部外围模块的寄存器进行初始化。另外，通过看门狗定时器的溢出，也可设定为复位状态。详细内容请参考 10. 看门狗定时器 (WDT)。复位后，中断控制模式为 0。

【注】 具有边界扫描功能的 HD64F2218 与 HD64F2218U，务请通电时，将 $\overline{\text{TRST}}$ 设定为低电平。详细内容请参考 13. 边界扫描。

* 仅限 H8S/2218 群。

4.3.1 复位种类

H8S/2218 群的复位有加电复位与手动复位 2 种。H8S/2212 群仅有加电复位 1 种。

复位种类如表 4.3 所示。通电时，务请设定为加电复位。

加电复位与手动复位均可进行 CPU 内部状态初始化。另外，相对于加电复位全面初始化内部外围模块寄存器，而手动复位则初始化总线控制器、I/O 端口以外的内部外围模块寄存器。保持总线控制器、I/O 端口的状态。

手动复位时，因初始化内部外围模块作为内部外围模块的输入 / 输出管脚使用的端口，切换成通过 DDR 和 DR 控制的输入 / 输出端口。

表 4.3 复位的种类

种类	向复位转移的条件		内部状态	
	$\overline{\text{MRES}}$	$\overline{\text{RES}}$	CPU	内置外围模块
加电复位	×	Low	初始化	初始化
手动复位	Low	High	初始化	除总线控制器，I/O 端口以外初始化

【符号说明】

×: Don't care

通过看门狗定时器产生的复位有加电复位，手动复位 2 种。

另外，使用 $\overline{\text{MRES}}$ 管脚 * 时，必须通过 SYSCR 的 $\overline{\text{MRES}}$ 位，将 $\overline{\text{MRES}}$ 管脚 * 设置为输入允许 ($\overline{\text{MRES}}=1$)。

【注】 * 仅限 H8S/2218 群。

4.3.2 复位异常处理

$\overline{\text{RES}}$ 管脚、 $\overline{\text{MRES}}$ 管脚 * 为低电平时，本 LSI 进入复位状态。

为确保本 LSI 复位，加电时， $\overline{\text{RES}}$ 必须保持最低 20ms 的低电平。另外，在运行过程中， $\overline{\text{RES}}$ 管脚或 $\overline{\text{MRES}}$ 管脚 * 在最低 20 个状态内，请保持为低电平。

$\overline{\text{RES}}$ 管脚、 $\overline{\text{MRES}}$ 管脚 * 在一定时间低电平后成为高电平时，开始复位异常处理，本 LSI 进行以下运行。

1. 初始化 CPU 内部状态与内部外围模块各寄存器，EXR 的 T 位清零，EXR 与 CCR 的 I 位置位为 1。
2. 读取复位异常处理向量地址并传送至 PC 后，从 PC 显示的地址开始执行程序。

【注】 * 仅限 H8S/2218 群。

复位顺序例如图 4.1，图 4.2 所示。

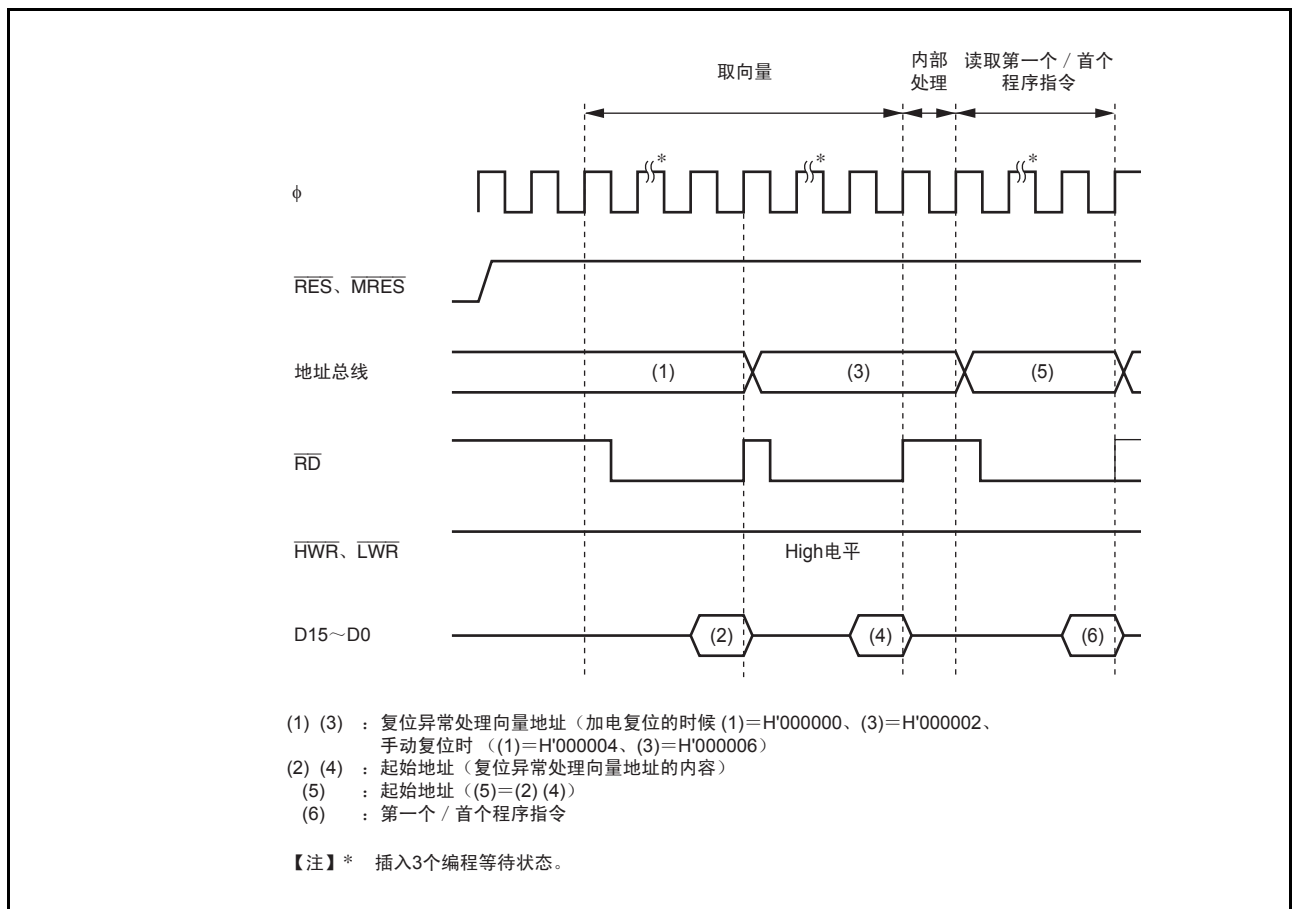


图 4.1 复位顺序 (模式 4)

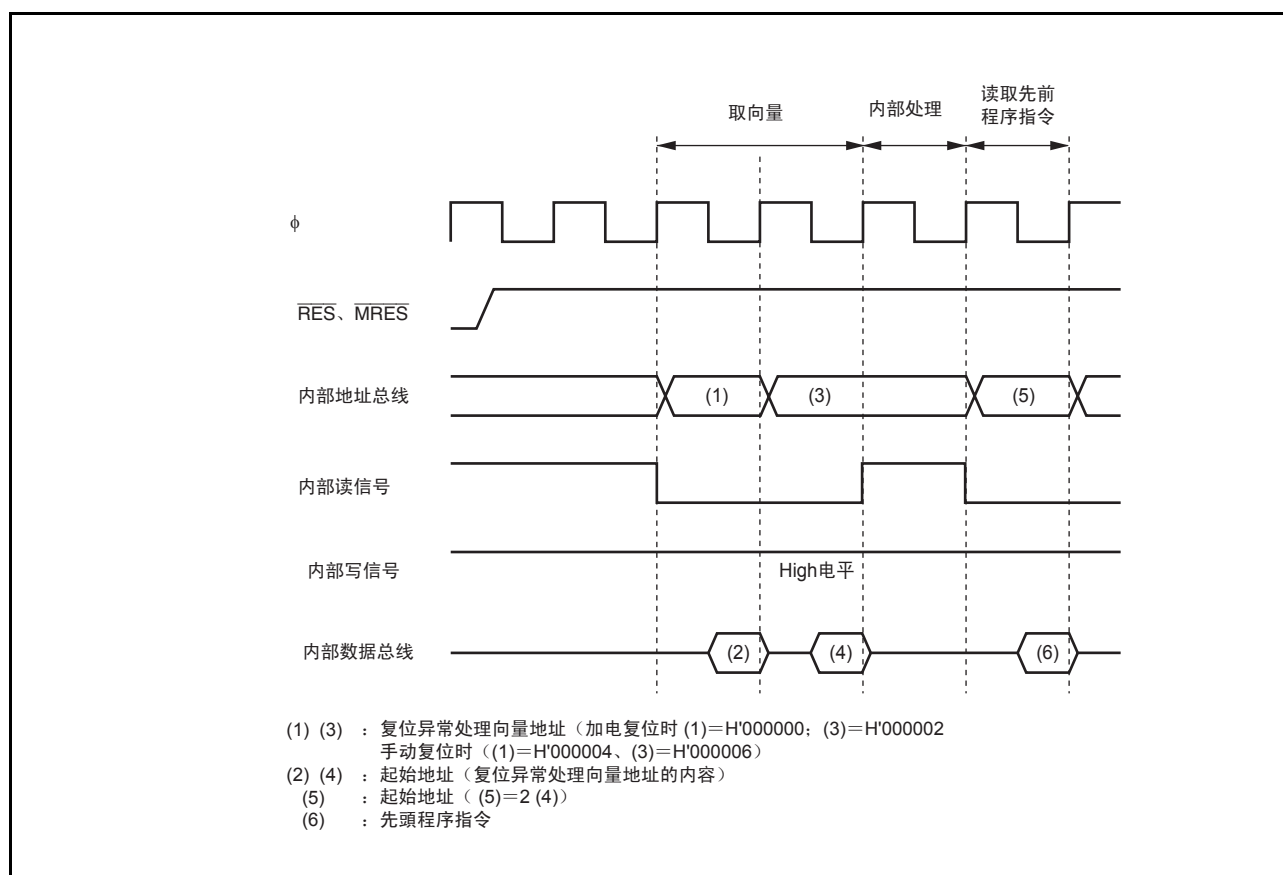


图 4.2 复位 (模式 6、7)

4.3.3 复位后的中断

复位后，堆栈指针 (SP) 初始化前接受中断时，PC 与 CCR 的保存就不能正常进行，因此，将发生程序失控。为防止上述情况发生，执行完复位异常处理后，禁止包含 NMI 在内的所有中断请求。即复位后必须执行程序的第一条指令，所以，程序的第一条指令务请设定为初始化 SP 的指令 (例: `MOVL # xx: 32SP`)。

4.3.4 复位解除后的内置外围功能

复位解除后，MSTPCRA, MSTPCRB, MSTPCRC 初始化，除闪存与 DMAC 以外的所有模块均变成模块停止模式。

因此，各内部外围模块寄存器不可读取 / 写入。通过解除模块停止模式，可进行寄存器的读取 / 写入。

4.4 跟踪异常处理

跟踪在中断控制模式 2 下有效。中断控制模式 0 下，无论 T 位的状态如何，均不会成为跟踪模式。中断控制模式请参考 5. 中断控制器。

EXR_T 位置位为 1 时，成为跟踪模式。跟踪模式下，CPU 执行指令，均开始跟踪异常处理。跟踪异常处理不受 CCR 中断屏蔽位的影响。跟踪异常处理后 CCR、EXR 的状态如表 4.4 所示。通过跟踪异常处理，EXR_T 位清零，解除跟踪模式，但保存在堆栈中的 T 位保持 1；通过 RTE 指令从跟踪异常处理程序返回后，再次成为跟踪模式。执行 RTE 指令进行跟踪异常处理。

即使跟踪异常处理程序也接受中断。

表 4.4 跟踪异常处理后的 CCR、EXR 状态

中断控制模式	CCR		EXR	
	I	UI	I2 ~ I0	T
0	不可使用跟踪异常处理			
2	1	—	—	0

【符号说明】

- 1：置位为 1。
- 0：清零。
- ：保持执行前的值。

4.5 中断异常处理

中断通过中断控制器进行控制。中断控制有 2 种控制模式，可向 NMI 以外的中断设定 8 个级别的优先级 / 屏蔽级，进行多路中断控制。因产品不同，开始中断异常处理的异常源与向量地址有所不同。详细内容请参考 5. 中断控制器。

中断异常处理运行如下：

1. 将程序计数器（PC）与条件码寄存器（CCR）、扩展寄存器（EXR）内容保存堆栈中。
2. 更新中断屏蔽位，将 T 位清零。
3. 生成中断源对应的向量地址，从向量表中将起始地址装入 PC，从此地址开始执行程序。

4.6 陷阱指令异常处理

陷阱指令异常处理执行 TRAPA 指令，开始进行异常处理。程序运行状态下，可随时进行陷阱指令异常处理。

陷阱指令异常处理的运行如下：

1. 将程序计数器（PC）与条件码寄存器（CCR）、扩展寄存器（EXR）内容保存至堆栈中。
2. 更新中断屏蔽位，将T位清零。
3. 生成中断源对应的向量地址，从向量表中将起始地址装入PC，从此地址开始执行程序。

TRAPA 指令从指令码中指定的 0 ~ 3 向量号相应的向量表中取出起始地址。

执行陷阱指令异常处理后 CCR、EXR 状态如表 4.5 所示。

表 4.5 陷阱指令异常处理后 CCR、EXR 状态

中断控制模式	CCR		EXR	
	I	UI	I2 ~ I0	T
0	1	—	—	—
2	1	—	—	0

【符号说明】

- 1：置位为 1。
- 0：清零。
- ：保持执行前的值

4.7 异常处理后的堆栈状态

陷阱指令异常处理及中断异常处理后的堆栈状态如图 4.3 所示。

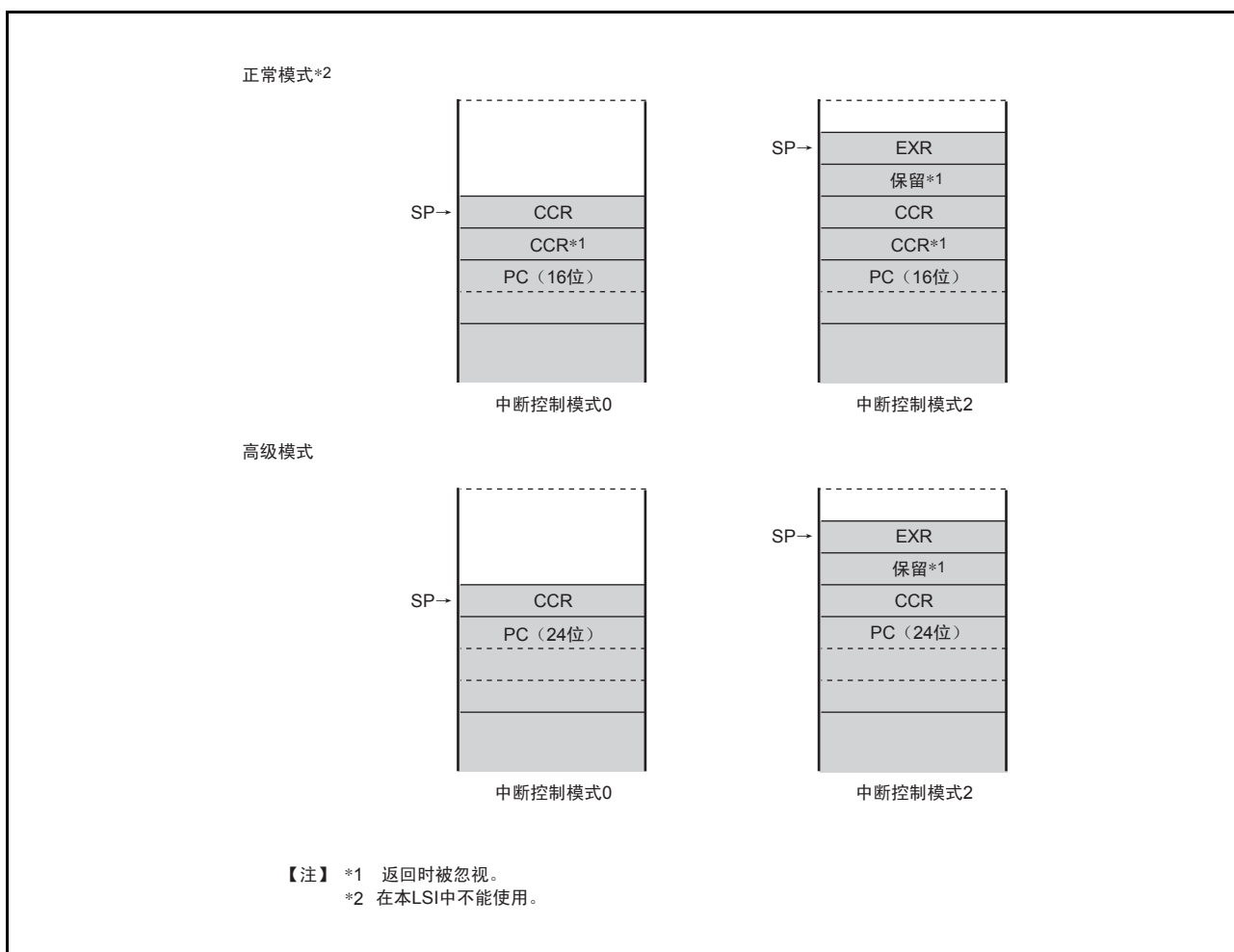


图 4.3 结束异常处理后的堆栈状态

4.8 使用注意事项

存取字数据或长字数据时，地址的最低有效位可视为 0。

堆栈区域的存取，常以字长度或长字节长度来进行，堆栈指针（SP: ER7）内容请勿设定为奇数。

即寄存器的保存请使用：

```
PUSH.W Rn(MOV.W Rn, @-SP)
```

```
PUSH.L ERn(MOV.L ERn, @-SP)
```

另外，寄存器的返回请使用：

```
POP.W Rn(MOV.W @SP+, Rn)
```

```
POP.L ERn(MOV.L @SP+, ERn)
```

SP 设定为奇数，成为产生误动作的原因。SP 设定为奇数时的运行例如图 4.4 所示。

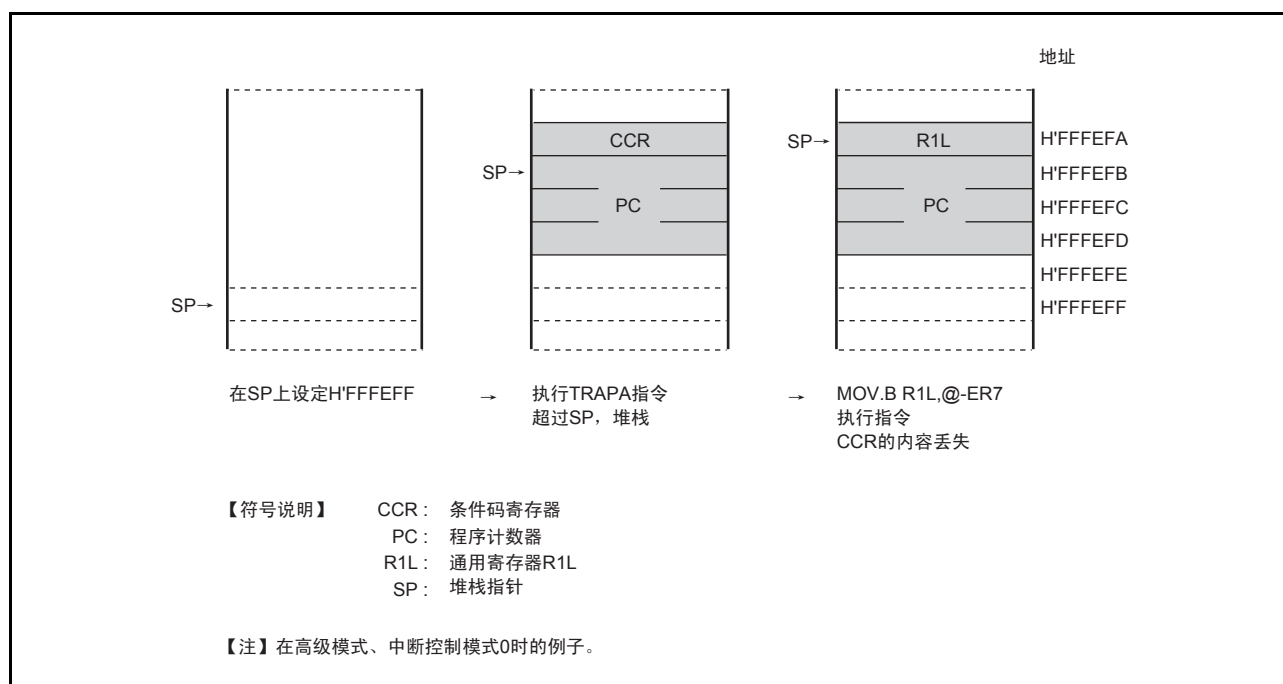


图 4.4 SP 设定为奇数时的运行

5. 中断控制器

5.1 特点

- 2种中断控制模式
通过系统控制寄存器（SYSCR）INTM1、INTM0位来设定2种中断控制模式。
- 可通过IPR来设定优先级
通过中断优先级寄存器（IPR），对每个模块的中断请求（除NMI外）设定8个级别的优先级。NMI作为8个级别中最高级的中断请求，随时都能被接受。
- 独立的向量地址
所有中断源分配了独立的向量地址，无需通过中断处理程序来判别中断源。
- 7个外部中断管脚（NMI、 $\overline{\text{IRQ7}}$ 、 $\overline{\text{IRQ4}} \sim \overline{\text{IRQ0}}$ ）
NMI以优先级最高的中断随时被接受。NMI可选择上升沿或者下降沿。 $\overline{\text{IRQ7}}$ 、 $\overline{\text{IRQ4}} \sim \overline{\text{IRQ0}}$ 可以独立地分别选择下降沿、上升沿、双边沿及检测级别。 $\overline{\text{IRQ6}}$ 为内部USB专用中断。 $\overline{\text{IRQ5}}$ 为内部RTC专用中断。
- DMAC控制
可通过中断请求启动DMAC。

中断控制器框图如图 5.1 所示。

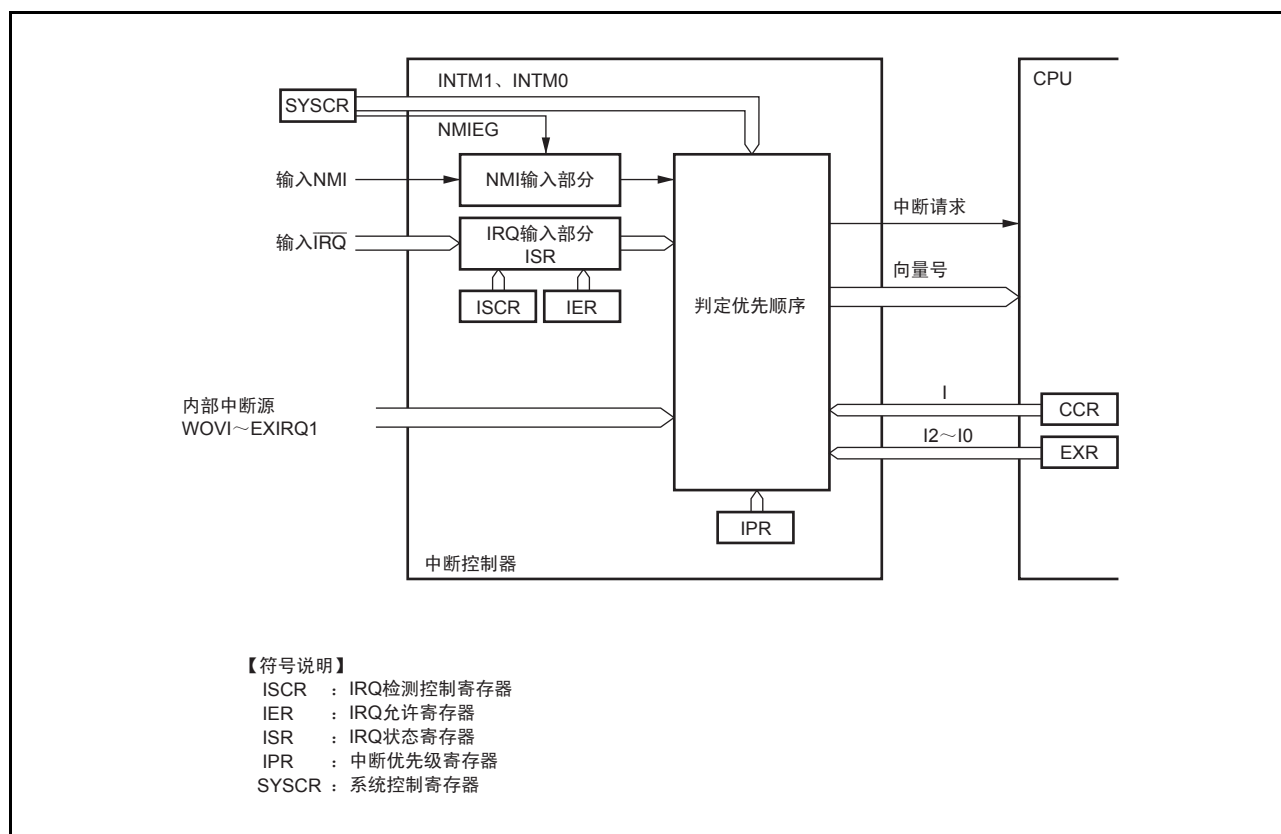


图 5.1 中断控制器框图

5.2 输入 / 输出管脚

中断控制器管脚结构如表 5.1 所示。

表 5.1 管脚结构

名称	输入 / 输出	功能
NMI	输入	非屏蔽外部中断管脚 可选择上升沿或下降沿
$\overline{\text{IRQ7}}$	输入	可屏蔽的外部中断管脚 可独立地选择下降沿、上升沿、双边沿及检测级别。（ $\overline{\text{IRQ6}}$ 为内部 USB 专用的中断信号。IRQ5 是内部 RTC 专用中断信号。）
$\overline{\text{IRQ4}}$	输入	
$\overline{\text{IRQ3}}$	输入	
$\overline{\text{IRQ2}}$	输入	
$\overline{\text{IRQ1}}$	输入	
$\overline{\text{IRQ0}}$	输入	

5.3 寄存器的说明

中断控制器有以下几种寄存器：关于系统控制寄存器请参考 3.2.2 系统控制寄存器（SYSCR）。

- 系统控制寄存器（SYSCR）
- IRQ 检测控制寄存器 H（ISCRH）
- IRQ 检测控制寄存器 L（ISCR L）
- IRQ 允许寄存器（IER）
- IRQ 状态寄存器（ISR）
- 中断优先级寄存器 A（IPRA）
- 中断优先级寄存器 B（IPRB）
- 中断优先级寄存器 C（IPRC）
- 中断优先级寄存器 D（IPRD）
- 中断优先级寄存器 E（IPRE）
- 中断优先级寄存器 F（IPRF）
- 中断优先级寄存器 G（IPRG）
- 中断优先级寄存器 J（IPRJ）
- 中断优先级寄存器 K（IPRK）
- 中断优先级寄存器 M（IPRM）

5.3.1 中断优先级寄存器 A ~ G、J、K、M (IPRA ~ IPRG、IPRJ、IPRK、IPRM)

IPR 设定除 NMI 以外的中断源优先级 (级别 7 ~ 0)。各中断源与 IPR 的对应关系请参考 5.5 中断异常处理向量表。通过对位 6 ~ 4, 位 2 ~ 0 的各 3 位设定 H'0 至 H'7 范围的值来决定相应中断请求优先级。

位	位名	初始化	R/W	说明
7	—	0	—	保留位 读取时, 常读取为 0。
6	IPR6	1	R/W	设定相应的中断源优先级。 000: 优先级 0 (最低) 001: 优先级 1 010: 优先级 2 011: 优先级 3 100: 优先级 4 101: 优先级 5 110: 优先级 6 111: 优先级 7 (最高)
5	IPR5	1	R/W	
4	IPR4	1	R/W	
3	—	0	—	
2	IPR2	1	R/W	设定相应的中断源优先级。 000: 优先级 0 (最低) 001: 优先级 1 010: 优先级 2 011: 优先级 3 100: 优先级 4 101: 优先级 5 110: 优先级 6 111: 优先级 7 (最高)
1	IPR1	1	R/W	
0	IPR0	1	R/W	

5.3.2 IRQ 允许寄存器 (IER)

IER 允许 IRQ7 ~ IRQ0 中断请求。

位	位名	初始化	R/W	说明
7	IRQ7E	0	R/W	IRQ7 允许 此位为 1 时, 允许 IRQ7 中断请求。
6	IRQ6E	0	R/W	IRQ6 允许 *1 此位为 1 时, 允许 IRQ6 中断请求。
5	IRQ5E	0	R/W	IRQ5 允许 *2 此位为 1 时, 允许 IRQ5 中断请求。
4	IRQ4E	0	R/W	IRQ4 允许 此位为 1 时, 允许 IRQ4 中断请求。
3	IRQ3E	0	R/W	IRQ3 允许 此位为 1 时, 允许 IRQ3 中断请求。
2	IRQ2E	0	R/W	IRQ2 允许 此位为 1 时, 允许 IRQ2 中断请求。
1	IRQ1E	0	R/W	IRQ1 允许 此位为 1 时, 允许 IRQ1 中断请求。
0	IRQ0E	0	R/W	IRQ0 允许 此位为 1 时, 允许 IRQ0 中断请求。

【注】 *1 IRQ6 为内部 USB 专用中断。

*2 IRQ5 为内部 RTC 专用中断。

5.3.3 IRQ 感知控制寄存器 H、L (ISCRH、ISCRL)

ISCR 从 $\overline{\text{IRQ7}} \sim \overline{\text{IRQ0}}$ 管脚中选择发生中断请求的中断源。

位	位名	初始化	R/W	说明
15 14	IRQ7SCB IRQ7SCA	0 0	R/W R/W	IRQ7 检测控制 B IRQ7 检测控制 A 00: $\overline{\text{IRQ7}}$ 输入的低电平, 发生中断请求。 01: $\overline{\text{IRQ7}}$ 输入的下降沿, 发生中断请求。 10: $\overline{\text{IRQ7}}$ 输入的上升沿, 发生中断请求。 11: $\overline{\text{IRQ7}}$ 输入的下降沿、上升沿双边沿, 均发生中断请求。
13 12	IRQ6SCB IRQ6SCA	0 0	R/W R/W	IRQ6*1 检测控制 B IRQ6*1 检测控制 A 00: 在使用内部 USB 的挂起 / 恢复中断时, 禁止设定。 01: 在 $\overline{\text{IRQ6}}$ 输入的下降沿, 发生中断请求。 1X: 禁止设定。
11 10	IRQ5SCB IRQ5SCA	0 0	R/W R/W	IRQ5*2 检测控制 B IRQ5*2 检测控制 A 00: 使用 RTC 中断时, 禁止设定。 01: $\overline{\text{IRQ5}}$ 输入的下降沿, 发生中断请求。 1X: 禁止设定
9 8	IRQ4SCB IRQ4SCA	0 0	R/W R/W	IRQ4 检测控制 B IRQ4 检测控制 A 00: $\overline{\text{IRQ4}}$ 输入的低电平, 发生中断请求。 01: $\overline{\text{IRQ4}}$ 输入的下降沿, 发生中断请求。 10: $\overline{\text{IRQ4}}$ 输入的上升沿, 发生中断请求。 11: $\overline{\text{IRQ4}}$ 输入的下降沿、上升沿双边沿, 均发生中断请求。
7 6	IRQ3SCB IRQ3SCA	0 0	R/W R/W	IRQ3 检测控制 B IRQ3 检测控制 A 00: $\overline{\text{IRQ3}}$ 输入的低电平, 发生中断请求。 01: $\overline{\text{IRQ3}}$ 输入的下降沿, 发生中断请求。 10: $\overline{\text{IRQ3}}$ 输入的上升沿, 发生中断请求。 11: $\overline{\text{IRQ3}}$ 输入的下降沿、上升沿双边沿, 均发生中断请求。
5 4	IRQ2SCB IRQ2SCA	0 0	R/W R/W	IRQ2 检测控制 B IRQ2 检测控制 A 00: $\overline{\text{IRQ2}}$ 输入的低电平, 发生中断请求。 01: $\overline{\text{IRQ2}}$ 输入的下降沿, 发生中断请求。 10: $\overline{\text{IRQ2}}$ 输入的上升沿, 发生中断请求。 11: $\overline{\text{IRQ2}}$ 输入的下降沿、上升沿双边沿, 均发生中断请求。
3 2	IRQ1SCB IRQ1SCA	0 0	R/W R/W	IRQ1 检测控制 B IRQ1 检测控制 A 00: $\overline{\text{IRQ1}}$ 输入的低电平, 发生中断请求。 01: $\overline{\text{IRQ1}}$ 输入的下降沿, 发生中断请求。 10: $\overline{\text{IRQ1}}$ 输入的上升沿, 发生中断请求。 11: $\overline{\text{IRQ1}}$ 输入的下降沿、上升沿双边沿, 发生中断请求。

位	位名	初始化	R/W	说明
1	IRQ0SCB	0	R/W	IRQ0 检测控制 B
0	IRQ0SCA	0	R/W	IRQ0 检测控制 A 00: $\overline{\text{IRQ0}}$ 输入的低电平, 发生中断请求。 01: $\overline{\text{IRQ0}}$ 输入的下降沿, 发生中断请求。 10: $\overline{\text{IRQ0}}$ 输入的上升沿, 发生中断请求。 11: $\overline{\text{IRQ0}}$ 输入的下降沿、上升沿双边沿, 均发生中断请求。

【符号说明】

X: Don't care

【注】 *1 IRQ6 为内部 USB 专用中断。

*2 IRQ5 为内部 RTC 专用中断。

5.3.4 IRQ 状态寄存器 (ISR)

ISR 为 IRQ7 ~ IRQ0 中断请求标志寄存器。

位	位名	初始化	R/W	说明
7	IRQ7F	0	R/(W)*	[置位条件]
6	IRQ6F	0	R/(W)*	在 ISCR 选择的中断源时。
5	IRQ5F	0	R/(W)*	[清除条件]
4	IRQ4F	0	R/(W)*	• 在读取 1 的状态后, 写入 0 时。
3	IRQ3F	0	R/(W)*	• 在设定为低电平检测状态并且 $\overline{\text{IRQn}}$ 输入为高电平状态下, 进行中断异常处理时。
2	IRQ2F	0	R/(W)*	
1	IRQ1F	0	R/(W)*	• 在设置为下降沿、上升沿、双边沿检测状态下, 进行 $\overline{\text{IRQn}}$ 中断异常处理时。
0	IRQ0F	0	R/(W)*	

【注】 * 仅限为清除标志。

5.4 中断源

5.4.1 外部中断源

外部中断有 NMI、IRQ7、IRQ4 ~ IRQ0, 7 个中断源。外部中断可用于软件待机模式的返回。虽然 IRQ5 为内部 RTC 专用中断、IRQ6 为内部 USB 专用中断, 但他们均可用于软件待机模式的返回。另外, 与其他的 IRQ7、IRQ4 ~ IRQ0 具有同等功能。

5.4.1.1 NMI 中断

非屏蔽中断请求 NMI 为最优先级最高的外部中断请求, 不管中断控制模式与 CPU 的中断屏蔽位的状态如何, 随时可被接受。可通过设定 SYSCR 的 NMIEG 位, 选择是否在 NMI 管脚的上升沿或下降沿, 发生中断请求。

5.4.1.2 IRQ7 ~ IRQ0 中断

IRQ7 ~ IRQ0 中断，通过 $\overline{\text{IRQ7}} \sim \overline{\text{IRQ0}}$ 管脚的输入信号发生中断请求。IRQ7 ~ IRQ0 中断有以下特点：

- 可通过 ISCR 选择是否在 $\overline{\text{IRQ7}} \sim \overline{\text{IRQ0}}$ 管脚的低电平、下降沿、上升沿及双边沿发生中断请求
- 可通过 IER 屏蔽 IRQ7 ~ IRQ0 中断请求。
- 可通过 IPR 设定中断优先级。
- ISR 表示 IRQ7 ~ IRQ0 中断请求的状态。可用软件清除 ISR 标志。

IRQn 中断的框图如图 5.2 所示。

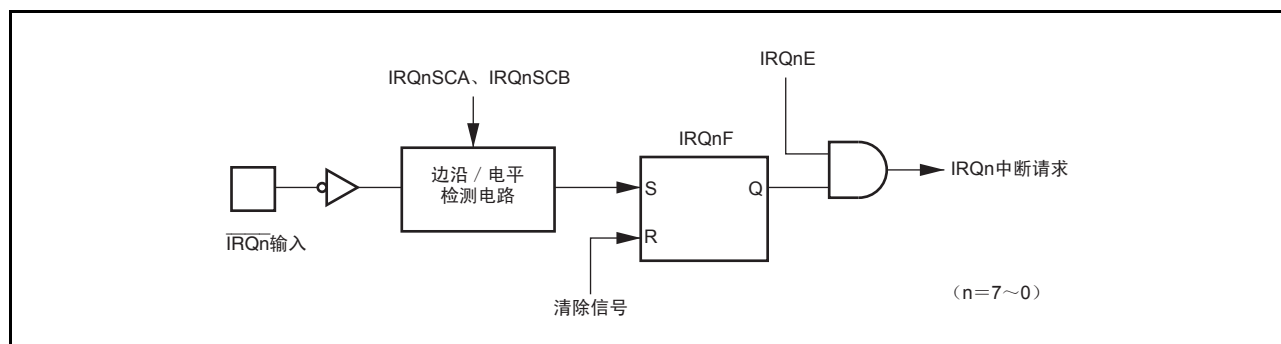


图 5.2 IRQn 中断的框图

IRQnF 的复位时序如图 5.3 所示。

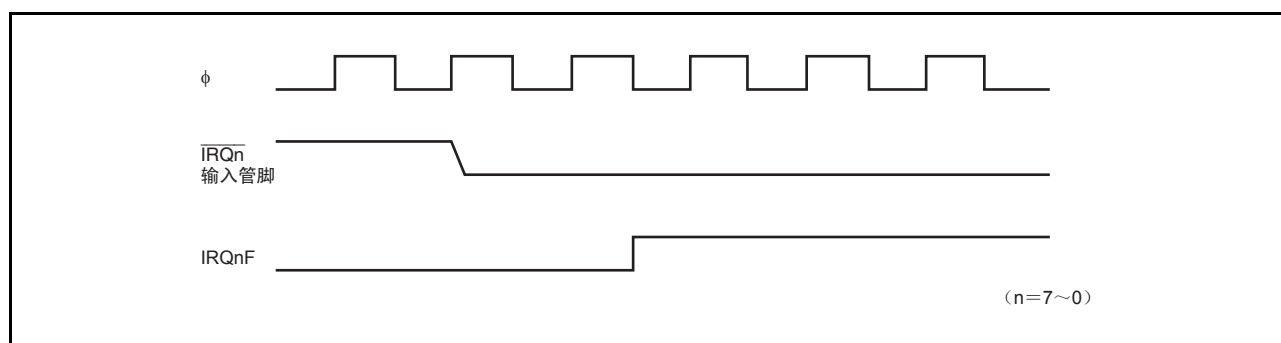


图 5.3 IRQnF 的复位时序

IRQn 中断的检测，不依赖于此管脚设定为输入还是输出。因此，作为外部中断输入管脚使用时，不能清除相应的 DDR，也不能作为其他功能的输入/输出管脚使用。另外，与 IER 的设定无关，中断请求标志 IRQnF 满足置位条件时，被置位，所以，仅参考必要的标志。

5.4.2 内部中断

内部外围模块的内部中断源有以下特点：

- 各内部外围模块，有表示中断请求状态的标志与允许位，并可独立屏蔽。如果允许位为 1 时，中断请求被发送至中断控制器。
- 可通过 IPR 设定中断优先级。
- 可通过 TPU、SCI 等的中断请求，启动 DMAC。
- 通过中断请求启动 DMAC 时，不受中断控制模式及 CPU 的中断屏蔽位的影响。

中断源	名称	向量号	向量地址 * 高级模式	IPR	优先级
SCI 通道 0	ERI0	80	H'0140	IPRJ2 ~ IPRJ0	高 ↑
	RXI0	81	H'0144		
	TXI0	82	H'0148		
	TEI0	83	H'014C		
SCI 通道 2	ERI2	88	H'0160	IPRK2 ~ IPRK0	↑
	RXI2	89	H'0164		
	TXI2	90	H'0168		
	TEI2	91	H'016C		
USB	EXIRQ0	104	H'01A0	IPRM6 ~ IPRM4	低 ↓
	EXIRQ1	105	H'01A4		

【注】 * 表示起始地址的低 16 位。

5.6 中断控制模式与中断运行

中断控制器有中断控制模式 0 与中断控制模式 2 两种模式，因中断控制模式不同而运行不同。通过 SYSCR 选择中断控制模式。中断控制模式 0 与中断控制模式 2 的不同点如表 5.3 所示。

表 5.3 中断控制模式

中断控制模式	中断优先级	中断屏蔽位	说明
0	默认值	I	各中断源的优先级在默认值中加以固定。 可通过 I 位，屏蔽除 NMI 以外的中断源。
2	IPR	I2 ~ I0	可通过 IPR 给除 NMI 以外的各中断源设定 8 个优先级。 通过 I2 ~ I0 位进行 8 个级别的中断屏蔽控制。

5.6.1 中断控制模式 0

中断控制模式 0 中，通过 CPU 的 CCR 的 I 位，屏蔽除 NMI 以外的所有中断请求。接受中断运行流程图如图 5.4 所示。

1. 中断允许位置位为 1 的中断源发生中断时，发送中断请求至中断控制器。
2. CPU 的 I 位置位为 1 时，中断控制器保留 NMI 以外的中断请求。清除 I 位时，接受中断请求。
3. 有数个中断请求时，中断控制器按优先级，选择优先级最高的中断请求，并向 CPU 请求中断处理，保留其他。
4. CPU 接受中断请求时，执行中的指令处理完毕后，开始中断异常处理。
5. 通过中断异常处理，将 PC 与 CCR 保存在堆栈区域。PC 保存返回后执行的第一条指令地址。
6. 将 CCR 的 I 位置位为 1。以此来屏蔽除 NMI 以外的中断。
7. CPU 生成接受中断请求相应的向量地址，并从向量表中读取中断程序开始地址，开始中断处理。

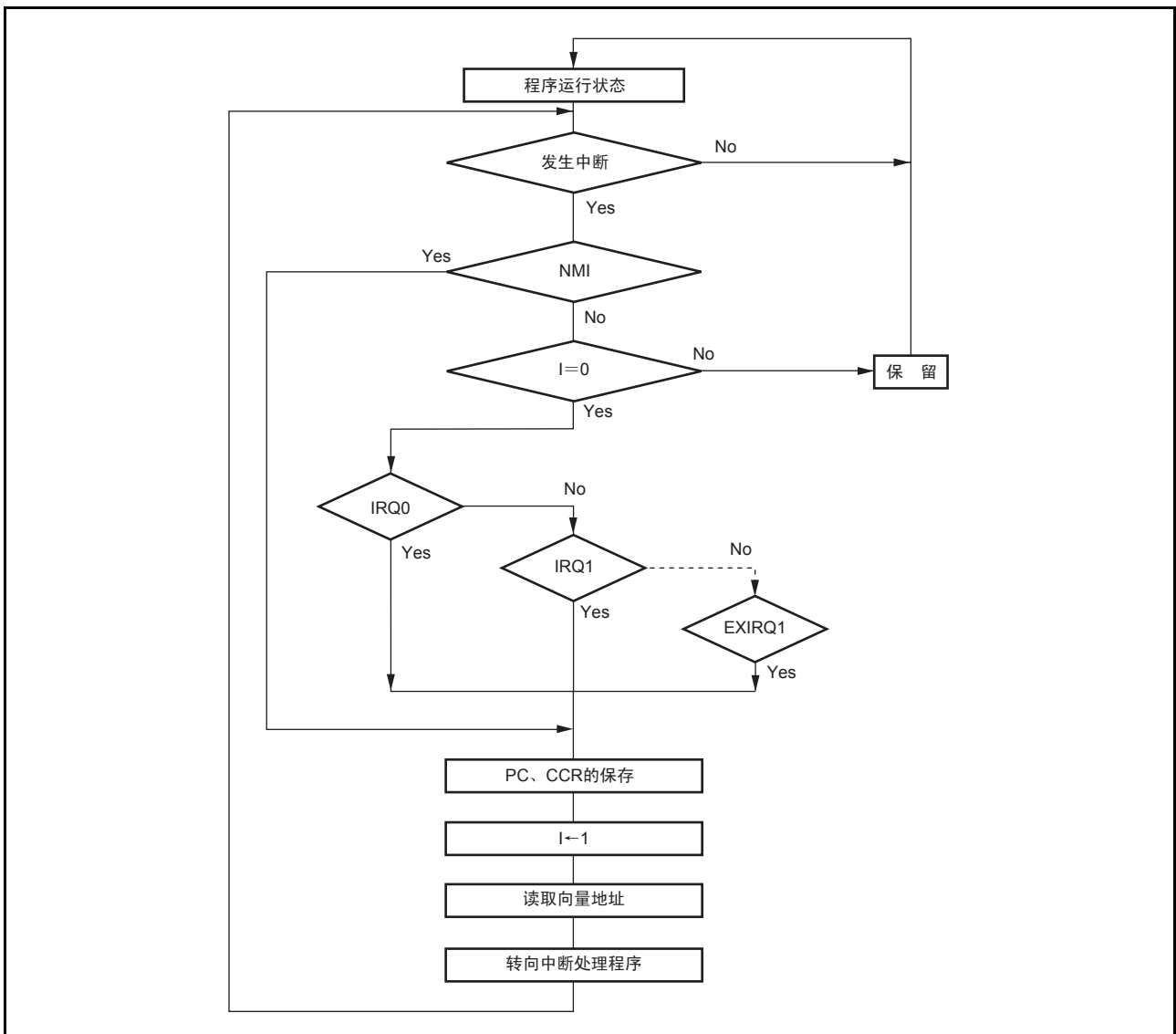


图 5.4 中断控制模式 0 的中断接受流程图

5.6.2 中断控制模式 2

中断控制模式 2 中，除 NMI 以外的中断请求通过 IPR 与 CPU 的 EXR 中断屏蔽级，来进行 8 级屏蔽控制。接受中断运行的流程如图 5.5 所示。

1. 中断允许位置位为 1 的中断源发生时，发送中断请求至中断控制器。
2. 有数个中断请求时，中断控制器按 IPR 所设定的中断优先级选择优先级最高的中断，保留比此低位的中断请求。优先级相同时，按照表 5.2 所示的默认优先级来选择中断请求。
3. 此后，比较已选择中断请求优先级与 EXR 中断屏蔽级。如为设置的屏蔽级，则保留，如优先级比中断屏蔽级高，则请求 CPU 进行中断处理。
4. CPU 接受中断请求时，执行中的指令后，开始中断异常处理。
5. 通过中断异常处理，将 PC、CCR 及 EXR 保存至堆栈区中。返回后将执行的第一条指令的地址保存于 PC。
6. EXR 的 T 位进行清除。可将中断屏蔽级改写为已接受的中断优先级。已接受的中断为 NMI 时，将中断屏蔽级设定为 H'7。
7. CPU 生成接受中断请求相对应向量地址，并从向量表中读取中断程序开始地址，开始进行中断处理。

5.6.3 中断异常处理顺序

中断异常处理顺序如图 5.6 所示。在高级模式下，中断控制模式 0、程序区及堆栈区均为内部存储器例。

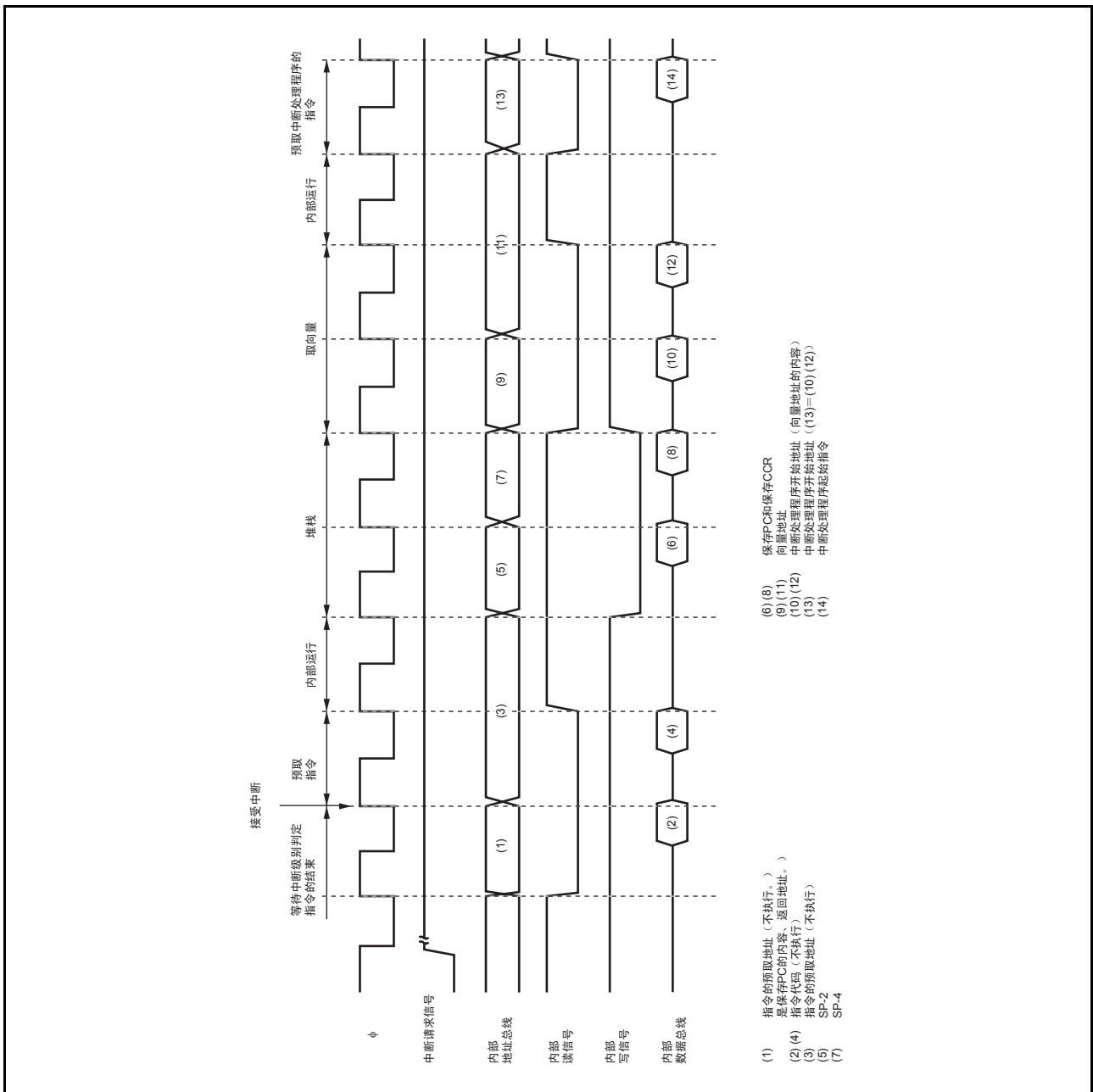


图 5.6 中断异常处理

5.6.4 中断响应时间

发生中断请求，至执行中断异常处理程序的第一条指令为止的中断响应时间如表 5.4 所示。表 5.4 执行状态符号请参考表 5.5。本 LSI 可对内部存储器进行高速字存取，所以将程序区设为内部 ROM、将堆栈区设为内部 RAM，以谋求提高处理速度。

表 5.4 中断响应时间

No.	执行状态	普通模式 *5		高级模式	
		中断控制模式 0	中断控制模式 2	中断控制模式 0	中断控制模式 2
1	中断优先级判别 *1	3			
2	至执行中指令退出的等待状态数 *2	$(1 \sim 19) + 2 \cdot S_I$			
3	PC、CCR 及 EXR 的堆栈	$2 \cdot S_K$	$3 \cdot S_K$	$2 \cdot S_K$	$3 \cdot S_K$
4	取向量	S_I		$2 \cdot S_I$	
5	取指令 *3	$2 \cdot S_I$			
6	内部处理 *4	2			
合计（使用内部存储器时）		11 ~ 31	12 ~ 32	12 ~ 32	13 ~ 33

【注】 *1 内部中断时为 2 个状态。

*2 表示 MULXS、DIVXS 指令。

*3 为接受中断后的预取与中断处理程序的预取。

*4 为接受中断后的内部处理与取向量后的内部处理，

*5 本 LSI 不可使用。

表 5.5 中断异常处理执行状态数

符号	内部存储器	存取对象			
		外部器件			
		8 位总线		16 总线	
		2 个状态存取	3 个状态存取	2 个状态存取	3 个状态存取
取指令 S_I	1	4	$6 + 2m$	2	$3 + m$
转移地址列表 S_J					
栈操作 S_K					

【符号说明】

m: 外部器件存取时的等待状态数

5.6.5 通过中断启动 DMAC

可通过中断启动 DMAC。此时，可进行以下选择。

1. 对CPU的中断请求
2. 对DMAC的启动请求
3. 1、2.全选

关于可启动 DMAC 的中断请求，请参考 7. DMA 控制器（DMAC）

DMAC 与中断控制器框图如图 5.7 所示。

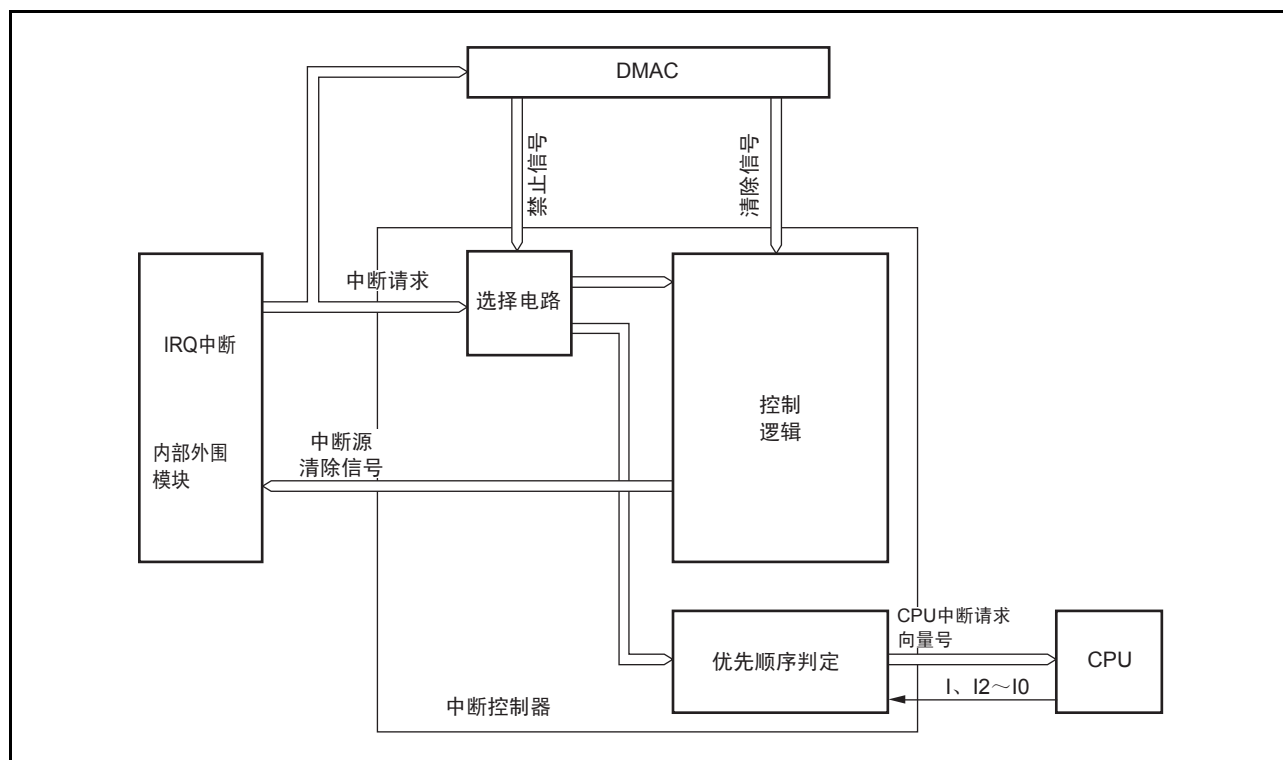


图 5.7 DMAC 与中断控制

5.6.5.1 中断源的选择

可将启动源直接输入 DMAC 各通道。通过 DMACR 的 DTF3 ~ DTF0 位，选择 DMAC 各通道启动源。可通过 DMABCR 的 DTA 位选择 DMAC 是否管理已选择的启动源。DTA 位置位为 1 时，成为 DMAC 启动源的中断源，不会成为 CPU 的中断源。

DMAC 管理的中断以外的中断源，均成为 CPU 中断请求。

5.6.5.2 优先级判别

将启动源直接输入 DMAC 各通道。

5.6.5.3 运行顺序

选择同一中断为 DMAC 启动源或 CPU 中断源时，这些均独立的进行运行状态及总线权优先级。
如表 5.6 所示为 DMAC 的 DMABCR 的 DTA 位设定的中断源选择与中断源清除控制。

表 5.6 中断源选择与清除控制

设置内容	中断源选择・清除控制	
DMAC		
DTA	DMAC	CPU
0	○	◎
1	◎	×

【符号说明】

- ◎：使用此中断。清除中断源。
(CPU 在中断处理程序中清除中断源标志。)
- ：使用此中断。不清除中断源。
- ×：不可使用此中断

5.6.5.4 使用注意事项

DMAC 读取 / 写入指定寄存器时，清除 SCI 及 A/D 转换器的中断源，与 DTA 位无关。

5.7 使用注意事项

5.7.1 中断发生与禁止的竞争

清除中断允许位、屏蔽中断请求时，中断屏蔽在结束执行其指令后有效。以 BCLR, MOV 等指令清除中断允许位、指令执行中发生中断请求时，指令执行退出时，该中断处于允许状态，所以，指令执行完成 / 结束后开始进行中断异常处理。但是，有比此中断优先级高的中断请求时，进行优先级比较高的中断异常处理，可忽视此中断。清零中断源标志时也相同。TPU 的 TIER_0 的 TGIEA 清零例如图 5.8 所示。屏蔽中断状态下，如果清零允许位或中断源标志，就不发生上述竞争。

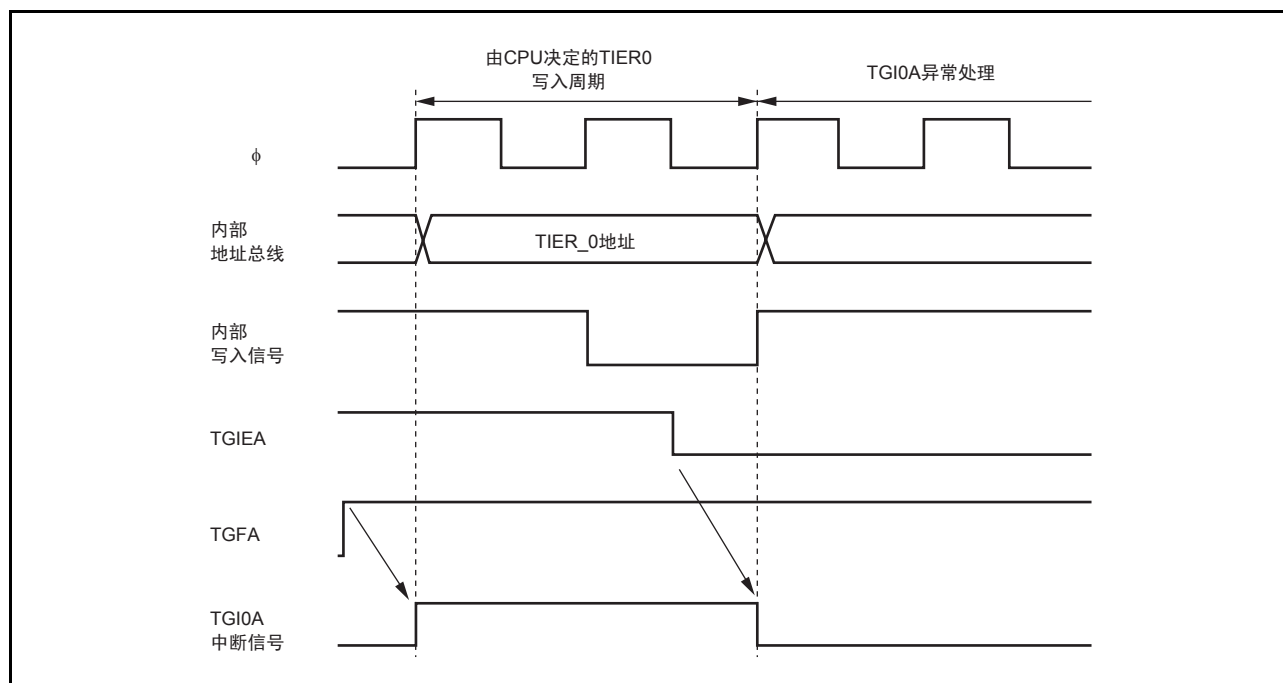


图 5.8 中断发生和中断屏蔽的竞争

5.7.2 禁止中断的指令

作为执行后不接受中断请求的指令有 LDC、ANDC、ORC、XORC 指令。这些指令执行完成 / 结束后，禁止中断（含 NMI 中断），必须执行下一条指令。通过这些指令设定 I 位时，指令执行完成 / 结束的 2 个状态后，新值有效。

5.7.3 中断屏蔽期间

中断控制器有禁止接受中断请求的期间。CPU 通过 LDC、ANDC、ORC、XORC 指令更新屏蔽级后的 3 个状态期间，中断控制器不接受中断请求。

5.7.4 EEPMOV 指令执行中的中断

EEPMOV.B 与 EEPMOV.W 指令时中断运行不同。

EEPMOV.B 指令时，即使传送中有含 NMI 的中断请求，传送结束不接受中断请求。

EEPMOV.W 指令时，传送中有中断请求的情况下，在传送周期的相接处开始进行中断异常处理。此时，堆栈的 PC 值成为下一条指令的地址。因此，在 EEPMOV.W 指令执行过程中发生中断时，务请设定以下程序：

```
L1: EEPMOV.W
    MOV.W    R4, R4
    BNE     L1
```

5.7.5 IRQ 中断

时钟运行时，IRQ 与时钟同步接受输入。

软件备用及监视模式时与异步接受输入。

输入条件请参考 "25A.3.2、25B.3.2、35C.3.2 控制信号中断"。

5.7.6 NMI 中断使用时的注意事项

NMI 中断为，在电特性规定条件下正常运行时，内置于本 LSI 的中断控制器与 CPU 共同执行的异常处理。因软件异常或对 LSI 管脚进行异常输入等而不能正常运行（失控状态）时，包括 NMI 中断在内的所有运行均无法保证。该情况下，通过外部复位可使 LSI 再次转换至正常的程序执行状态。

6. 总线控制器 (BSC)

本 LSI 内置总线控制器 (BSC)，将外部地址空间设定为 8 个区进行管理。另外，总线控制器具有总线仲裁功能，并控制内部总线主控器 CPU、DMA 控制器 (DMAC) 的运行。

6.1 特点

- 以区域为单位管理外部地址空间
将外部地址空间设定为以 2M 字节为单位的 8 个区进行管理
每个区设定总线规格
可设定突发式 ROM 接口
- 基本总线接口 *1
可向区域 0 ~ 5 输出片选 ($\overline{CS0} \sim \overline{CS5}$) *2
每个区域可选择 8 位存取空间 / 16 位存取空间
每个区域可选择 2 态存取空间 / 3 态存取空间
每个区域可插入程序等待状态
- 突发式 ROM 接口 *2
可向区域 0 设定突发式 ROM 接口
可选择突发存取的 1 态或 2 态
- 插入空闲周期 *2
不同区域间的外部读取周期时，可插入空闲周期
外部读取周期后的外部写入周期时，可插入空闲周期
- 总线权仲裁功能 (总线仲裁)
内置总线仲裁器，仲裁 CPU、DMAC 的总线权
- 其他
外部总线权释放功能 *2

【注】 *1 区域 6 的片选 $\overline{CS6}$ 为内部 USB 用，所以不可作为外部区域使用。

区域 6 务请设定为 8 位存取 / 3 态存取 / 无程序等待。

另外，向 RTC 相关的寄存器 (地址 = H'FFFF40 ~ H'FFFF5F) 的存取，按照区域 7 的设定。本芯片中，区域 7 务请设定为 8 位存取 / 3 态存取 / 无程序等待。

*2 H8S/2212 群不可使用这些功能。

总线控制器框图如图 6.1 所示。

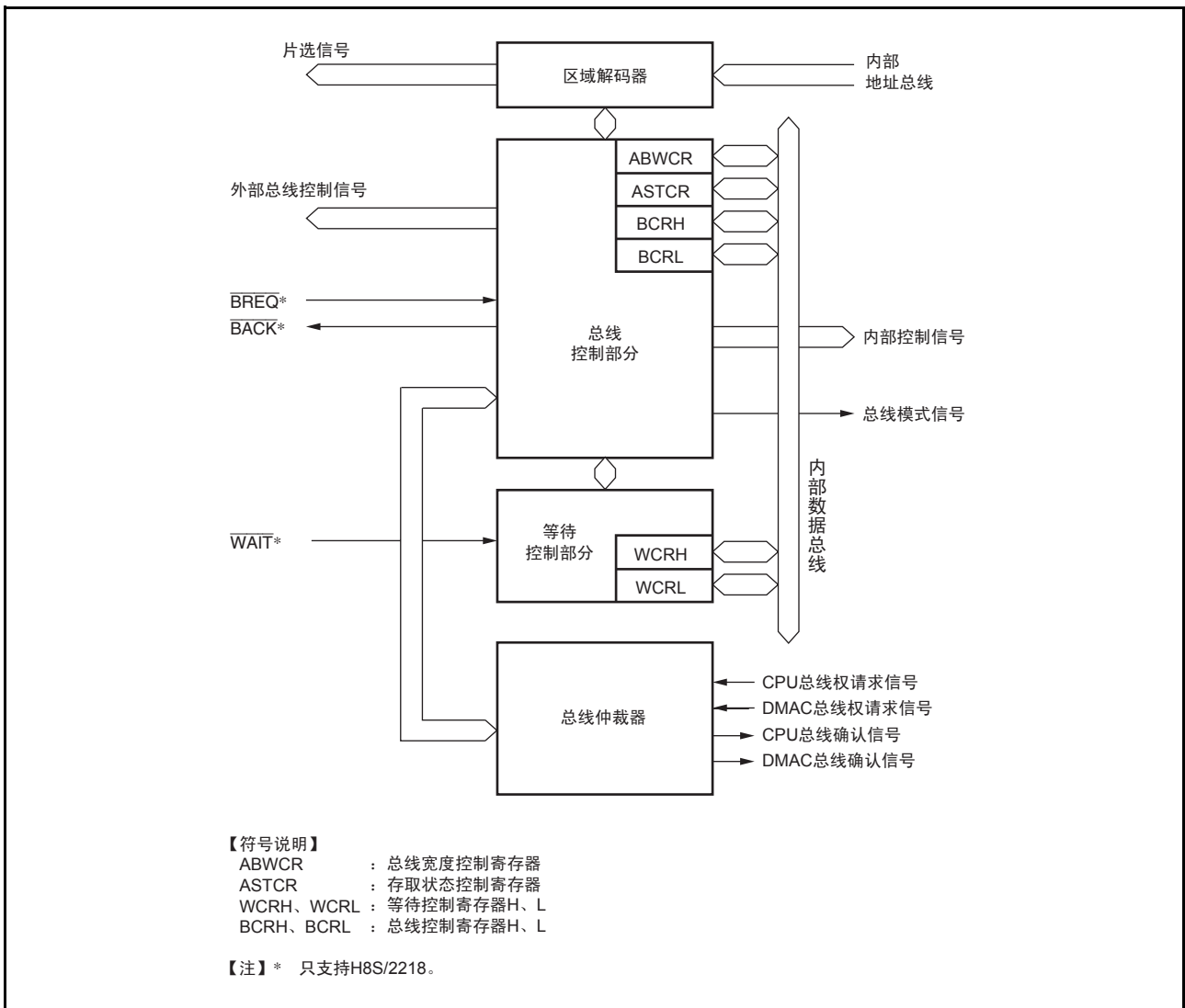


图 6.1 总线控制器框图

6.2 输入 / 输出管脚

总线控制器管脚结构如表 6.1 所示。
这些管脚仅限 H8S/2218 群。

表 6.1 管脚结构

名称	符号	输入 / 输出	功能
地址选通	\overline{AS}	输出	表示地址总线上的地址输出有效的选通信号。
读取	\overline{RD}	输出	表示读取外部地址空间的选通信号。
高位写入	\overline{HWR}	输出	表示写入外部地址空间、数据总线高位侧 (D15 ~ D8) 有效的选通信号。
低位写入	\overline{LWR}	输出	表示写入外部地址空间、数据总线低位侧 (D7 ~ D0) 有效的选通信号。
片选 0 ~ 5	$\overline{CS0} \sim \overline{CS5}$	输出	表示选择区域 0 ~ 5 的选通信号。
等待	\overline{WAIT}	输入	存取外部 3 态存取空间时的等待请求信号。
总线权请求	\overline{BREQ}	输入	总线权释放至外部的请求信号。
总线权请求应答	\overline{BACK}	输出	表示释放总线权的应答信号。

6.3 寄存器的说明

总线控制器的寄存器构成如下所示：

- 总线宽度控制寄存器 (ABWCR)
- 存取状态控制寄存器 (ASTCR)
- 等待控制寄存器H (WCRH)
- 等待控制寄存器L (WCRL)
- 总线控制寄存器H (BCRH)
- 总线控制寄存器L (BCRL)
- 管脚功能控制寄存器 (PFCR)

6.3.1 总线宽度控制寄存器 (ABWCR)

ABWCR 将各区域设定为 8 位存取空间或 16 位存取空间。

ABWCR 设定外部存储器空间数据总线宽度。USB 与 RTC 以外的内部存储器及内部 I/O 寄存器总线宽度均为固定的，与 ABWCR 值无关。

位	位名	初始值	R/W	说明
7	ABW7*2	1 / 0*1	R/W	区域 7 ~ 0 总线宽度控制 选择将相应的区域是设定为 8 位存取空间还是设置为 16 位存取空间。 0: 将区域 n 设定为 16 位存取空间 1: 将区域 n 设定为 8 位存取空间 【注】 n = 7 ~ 0
6	ABW6*2	1 / 0*1	R/W	
5	ABW5	1 / 0*1	R/W	
4	ABW4	1 / 0*1	R/W	
3	ABW3	1 / 0*1	R/W	
2	ABW2	1 / 0*1	R/W	
1	ABW1	1 / 0*1	R/W	
0	ABW0	1 / 0*1	R/W	

【注】 *1 模式 5 ~ 7 为 1、模式 4 为 0。H8S/2212 群中，务必将这些位设定为 1。

*2 因向区域 6 分配了内部 USB、向区域 7 分配了内部 RTC，所以务必设定 ABW7 = ABW6 = 1。

6.3.2 存取状态控制寄存器 (ASTCR)

ASTCR 将各区域设定为 2 态存取空间或 3 态存取空间。

ASTCR 设定外部存储器空间的存取状态数。内部 USB 以外的内部存储器及内部 I/O 寄存器的存取状态数均为固定，与 ASTCR 设定值无关。

位	位名	初始值	R/W	说明
7	AST7*	1	R/W	区域 7 ~ 0 存取状态控制 选择将相应的区域是设定为 2 态存取空间还是设定为 3 态存取空间。 同时，允许或禁止插入等待状态。 0: 将区 n 设定为 2 态存取空间 禁止向区域 n 的存取插入等待状态 1: 将区域 n 设定为 3 态存取空间 允许给区域 n 的存取插入等待状态 【注】 n = 7 ~ 0
6	AST6*	1	R/W	
5	AST5	1	R/W	
4	AST4	1	R/W	
3	AST3	1	R/W	
2	AST2	1	R/W	
1	AST1	1	R/W	
0	AST0	1	R/W	

【注】 * 因向区域 6 分配了内部 USB、向区域 7 分配了内部 RTC，所以务必设定 AST7 = AST6 = 1。

6.3.3 等待控制寄存器 H、L (WCRH、WCRL)

WCRH 与 WCRL 选择各区域的程序等待状态数。

不向内部存储器 (除内部 USB) 与内部 I/O 寄存器插入程序等待。

• WCRH

位	位名	初始值	R/W	说明
7 6	W71* W70*	1 1	R/W R/W	区域 7 等待控制 1、0 ASTCR 的 AST7 = 1 时, 选择存取区域 7 时的程序等待状态数。 00: 不插入程序等待 01: 插入 1 个程序等待状态 10: 插入 2 个程序等待状态 11: 插入 3 个程序等待状态
5 4	W61* W60*	1 1	R/W R/W	区域 6 等待控制 1、0 ASTCR 的 AST6 = 1 时, 存取区域 6 时的程序等待状态数。 00: 不插入程序等待 01: 插入 1 个程序等待状态 10: 插入 2 个程序等待状态 11: 插入 3 个程序等待状态
3 2	W51 W50	1 1	R/W R/W	区域 5 等待控制 1、0 选择在 ASTCR 的 AST5 = 1 时, 存取区域 5 时的程序等待状态数。 00: 不插入程序等待 01: 插入 1 个程序等待状态 10: 插入 2 个程序等待状态 11: 插入 3 个程序等待状态
1 0	W41 W40	1 1	R/W R/W	区 4 域等待控制 1、0 选择在 ASTCR 的 AST4 = 1 时, 选择存取区域 4 时的程序等待状态数。 00: 不插入程序等待 01: 插入 1 个程序等待状态 10: 插入 2 个程序等待状态 11: 插入 3 个程序等待状态

【注】 * 因向区域 6 分配了内部 USB、向区域 7 分配了内部 RTC, 所以务必设定 W71 = W70 = 0、W61 = W60 = 0。

• WCRL

位	位名	初始值	R/W	说明
7 6	W31 W30	1 1	R/W R/W	区域 3 等待控制 1、0 ASTCR 的 AST3 = 1 时, 选择存取区域 3 时的程序等待状态数。 00: 不插入程序等待 01: 插入 1 个程序等待状态 10: 插入 2 个程序等待状态 11: 插入 3 个程序等待状态
5 4	W21 W20	1 1	R/W R/W	区域 2 等待控制 1、0 ASTCR 的 AST2 = 1 时, 选择存取区域 2 时的程序等待状态数。 00: 不插入程序等待 01: 插入 1 个程序等待状态 10: 插入 2 个程序等待状态 11: 插入 3 个程序等待状态
3 2	W11 W10	1 1	R/W R/W	区域 1 等待控制 1、0 ASTCR 的 AST1 = 1, 选择存取区域 1 时的程序等待状态数。 00: 不插入程序等待 01: 插入 1 个程序等待状态 10: 插入 2 个程序等待状态 11: 插入 3 个程序等待状态
1 0	W01 W00	1 1	R/W R/W	区域 0 等待控制 1、0 ASTCR 的 AST0 = 1 时, 选择存取区域 0 时的程序等待状态数。 00: 不插入程序等待 01: 插入 1 个程序等待状态 10: 插入 2 个程序等待状态 11: 插入 3 个程序等待状态

6.3.4 总线控制寄存器 H (BCRH)

BCRH 进行空闲周期插入的允许或禁止、区域 0 的存储器接口的选择。

H8S/2212 群中，本寄存器保持初始值，请勿写入。

位	位名	初始值	R/W	说明
7	ICIS1	1	R/W	空闲周期插入 1 不同区域的外部读取周期连续时，选择是否在总线周期之间插入 1 个空闲周期状态。 0: 不同区域的外部读取周期连续时，不插入空闲周期 1: 不同区域的外部读取周期连续时，插入空闲周期
6	ICIS0	1	R/W	空闲周期插入 0 外部读取周期与外部写入周期连续时，选择是否在总线周期之间插入 1 个空闲周期状态。 0: 外部读取周期与外部写入周期连续时，不插入空闲周期 1: 外部读取周期与外部写入周期连续时，插入空闲周期
5	BRSTRM	0	R/W	突发式 ROM 允许 选择是否将区域 0 设定为突发式 ROM 接口。 0: 区域 0 为基本总线接口 1: 区域 0 为突发式 ROM 接口
4	BRSTS1	1	R/W	突发周期选择 1 选择突发式 ROM 接口的突发周期数。 0: 突发周期为 1 个状态 1: 突发周期为 2 个状态
3	BRSTS0	0	R/W	突发周期选择 0 选择突发式 ROM 接口的可突发存取字数。 0: 突发存取最大为 4 个字 1: 突发存取最大为 8 个字
2~0	—	全为 0	R/W	保留位 写入时务请写入 0。

6.3.5 总线控制寄存器 L (BCRL)

BCRL 进行外部总线释放状态协议的选择、 $\overline{\text{WAIT}}$ 管脚输入允许或禁止的选择。
本寄存器选择的功能仅 H8S/2218 群有效。H8S/2212 群中，本寄存器请勿写入。

位	位名	初始值	R/W	说明
7	BRLE*	0	R/W	总线释放允许 允许或禁止释放外部总线权。 0: 禁止释放外部总线权。 $\overline{\text{BREQ}}$ 、 $\overline{\text{BACK}}$ 可作为输入 / 输出端口使用。 1: 允许释放外部总线权。
6	—	0	R/W	保留位 写入时，必须写入 0。
5	—	0	—	保留位 读取时，常读取 0。写入无效。
4	—	0	R/W	保留位 写入时，务请写入 0。
3	—	1	R/W	保留位 写入时，务请写入 1。
2、1	—	全为 0	R/W	保留位 写入时，务请写入 0。
0	WAITE*	0	R/W	WAIT 管脚允许 选择允许或禁止通过 $\overline{\text{WAIT}}$ 管脚的等待输入 0: 禁止通过 $\overline{\text{WAIT}}$ 管脚的等待输入。 $\overline{\text{WAIT}}$ 管脚可作为输入 / 输出端口使用。 1: 允许通过 $\overline{\text{WAIT}}$ 管脚的等待输入。

【注】 * H8S/2212 群中，务请设定为 0。

6.3.6 管脚功能控制寄存器 (PFCR)

PFCR 进行外部扩展模式时的地址输出控制。

使用带有 E6000 仿真器的 USB 时，务请设定 AE3 ~ AE0 = 0010 等，允许 A8 与 A9 的输出。

位	位名称	初始值	R/W	说明
7 ~ 4	—	不定	R/W	保留位 写入时，务请写入 0。
3	AE3	1 / 0*	R/W	地址输出允许 3 ~ 0
2	AE2	1 / 0*	R/W	选择允许或禁止无 ROM 扩展模式与有 ROM 模式的地址输出 A8 ~ A23。允许
1	AE1	0	R/W	地址输出的管脚输出地址，与相应的 DDR 无关。禁止地址输出的管脚，将相应
0	AE0	1 / 0*	R/W	的 DDR 置位为 1 时，成为端口输出。 0000: 禁止 A8 ~ A23 输出 (模式 6, 7 的初始值) 0001: 允许 A8 输出。禁止 A9 ~ A23 输出。 0010: 允许 A8、A9 输出。禁止 A10 ~ A23 输出。 0011: 允许 A8 ~ A10 输出。禁止 A11 ~ A23 输出。 0100: 允许 A8 ~ A11 输出。禁止 A12 ~ A23 输出。 0101: 允许 A8 ~ A12 输出。禁止 A13 ~ A23 输出。 0110: 允许 A8 ~ A13 输出。禁止 A14 ~ A23 输出。 0111: 允许 A8 ~ A14 输出。禁止 A15 ~ A23 输出。 1000: 允许 A8 ~ A15 输出。禁止 A16 ~ A23 输出。 1001: 允许 A8 ~ A16 输出。禁止 A17 ~ A23 输出。 1010: 允许 A8 ~ A17 输出。禁止 A18 ~ A23 输出。 1011: 允许 A8 ~ A18 输出。禁止 A19 ~ A23 输出。 1100: 允许 A8 ~ A19 输出。禁止 A20 ~ A23 输出。 1101: 允许 A8 ~ A20 输出。禁止 A21 ~ A23 输出 (模式 4, 5 的初始值)。 1110: 允许 A8 ~ A21 输出。禁止 A22、A23 输出。 1111: 允许 A8 ~ A23 输出。

【注】 * 模式 4、5 中为 1；模式 6、7 中为 0。

6.4 总线控制

6.4.1 区域分割

高级模式下，总线控制器以 2M 字节为单位，将 16 M 字节的地址空间分割为 0 ~ 7 的 8 个区域，以区域为单位进行外部地址空间的总线控制。另外，普通模式 * 下，控制区域 0 的一部分 64K 字节的地址空间。存储器映射概要如图 6.2 所示。

可向 0 ~ 5 每个区域输出片选信号 ($\overline{CS0} \sim \overline{CS5}$)。

【注】 * 本 LSI 不可使用。

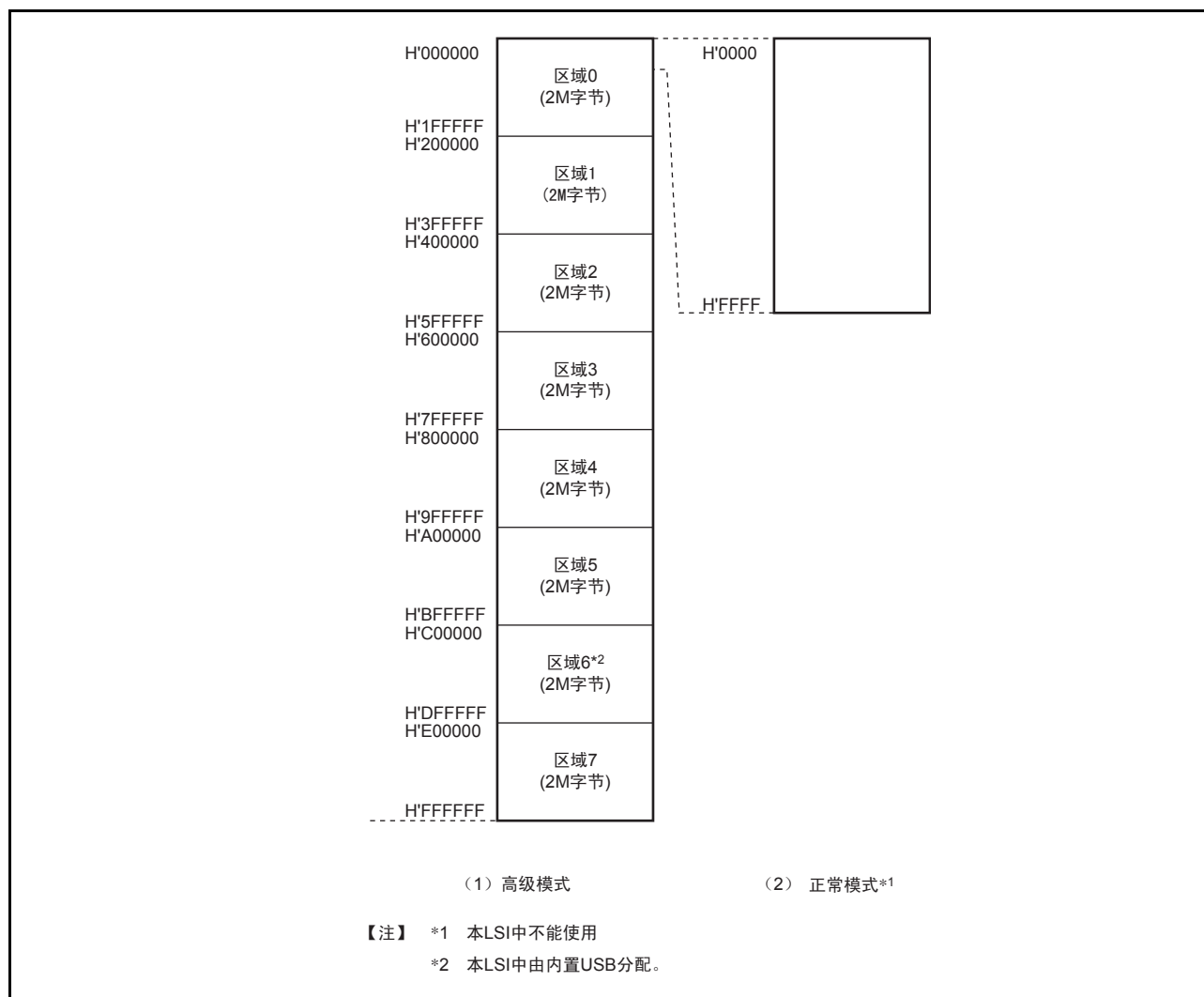


图 6.2 区域分割状态

6.4.2 总线规格

外部地址空间的总线规格由 (1) 总线宽度、(2) 存取状态数、(3) 程序等待状态数 3 个元素所构成。另外, 内部 USB 与 RTC 以外的存储器、内部 I/O 寄存器的总线宽度、存取状态数均为固定, 不受总线控制器的影响。

6.4.2.1 总线宽度

通过 ABWCR 来选择是 8 位还是 16 位的总线宽度。选择 8 位总线的区域为 8 位存取空间、选择 16 位总线的区域为 16 位存取空间。

将所有区域设定为 8 位存取空间时, 成为 8 位总线模式、将任意区域设定为 16 位存取空间时, 成为 16 位总线模式。设定为突发式 ROM 接口时, 通常成为 16 位总线模式。本 LSI 环境下, 务请将区域 6 及区域 7 设定为 8 位总线模式。

6.4.2.2 存取状态数

存取状态数通过 ASTCR 选择 2 态或 3 态。选择 2 态存取的区域为 2 态存取空间、选择 3 态存取的区域为 3 态存取空间。

突发式 ROM 接口有时不通过 ASTCR 来决定存取状态数。

设定为 2 态存取空间时, 禁止插入等待。

本 LSI 环境下, 务请将区域 6 及区域 7 设定为 3 态存取空间。

6.4.2.3 程序等待状态数

当通过 ASTCR 设定为 3 态存取空间时, 通过 WCRH、WCRL 自动选择插入的程序等待状态数。可选择 0 ~ 3 个程序等待状态。

本 LSI 环境下, 务请将区域 6 及区域 7 的程序等待数设定为 0 态。

表 6.2 各区域的总线规格 (基本总线接口)

ABWCR	ASTCR	WCRH、WCRL		总线规格 (基本总线接口)		
ABWn	ASTn	Wn1	Wn0	总线宽度	存取状态数	程序等待状态数
0	0	—	—	16	2	0
						1
	1	0	0		0	0
			1		0	1
1	0	—	—	8	2	0
						1
	1	0	0		0	0
			1		0	1

6.4.3 各区域的总线接口

各区域的初始状态为基本总线接口和 3 态存取空间。以运行模式选择总线宽度的初始状态。在此说明的总线规格仅为基本条款，所以请务必确认 6.6 基本总线接口、6.7 突发式 ROM 接口的各存储器接口条款。此外，在 H8S/2212 群中，ROM 常为有效，无外部扩展模式。

6.4.3.1 区域 0

区域 0 包含内部 ROM，在 ROM 无效扩展模式中，区域 0 的所有空间均为外部地址空间。在 ROM 有效扩展模式中，内部 ROM 以外的所有空间均为外部地址空间。

存取区域 0 的外部地址空间时，可输出 $\overline{CS0}$ 信号。

区域 0 可选择基本总线接口或突发式 ROM 接口。

6.4.3.2 区域 1 ~ 6

区域 1 ~ 6 在外部扩展模式时，区域 1 ~ 6 的所有空间均为外部地址空间。存取区域 1 ~ 5 的外部地址空间时，可分别输出 $\overline{CS1} \sim \overline{CS5}$ 管脚信号。区域 1 ~ 5 只可使用基本总线接口。区域 6 为内部 USB 专用区。详细内容请参考 14. 通用串行总线 (USB)。

6.4.3.3 区域 7

区域 7 包含内部 RAM 及内部 I/O 寄存器，在外部扩展模式时，除保留区域（详细内容请参考 3.4 各工作模式的地址映射）内部 RAM 及 I/O 寄存器（内部 RTC 以外）空间以外的空间，均为外部地址空间。另外，内部 RAM 将系统控制寄存器（SYSCR）的 RAME 位置位为 1 时有效。RAME 位清零时无效，相应的地址成为外部地址空间。区域 7 仅可使用基本总线接口。

6.4.4 片选信号

H8S/2218 群中，可向区 0 ~ 5 分别输出片选信号 ($\overline{CS0} \sim \overline{CS5}$)，存取该区域的外部地址空间时，输出低电平。 \overline{CSn} ($n = 0 \sim 5$) 信号输出时序如图 6.3 所示。通过设定各 \overline{CSn} 管脚相应端口的数据方向寄存器 (DDR) 来进行允许或禁止输出 \overline{CSn} 信号。

ROM 无效扩展模式中， $\overline{CS0}$ 管脚在加电复位后成为输出状态。因为加电复位后 $\overline{CS1} \sim \overline{CS5}$ 管脚成为输入状态，所以，在输出 $\overline{CS1} \sim \overline{CS5}$ 信号时，务请将相应的 DDR 置位为 1。

ROM 有效扩展模式中，因为加电复位后 $\overline{CS0} \sim \overline{CS5}$ 管脚均成为输入状态，所以，在输出 $\overline{CS0} \sim \overline{CS5}$ 信号时，务请将相应的 DDR 置位为 1。详细情况请参考 8. I/O 端口。

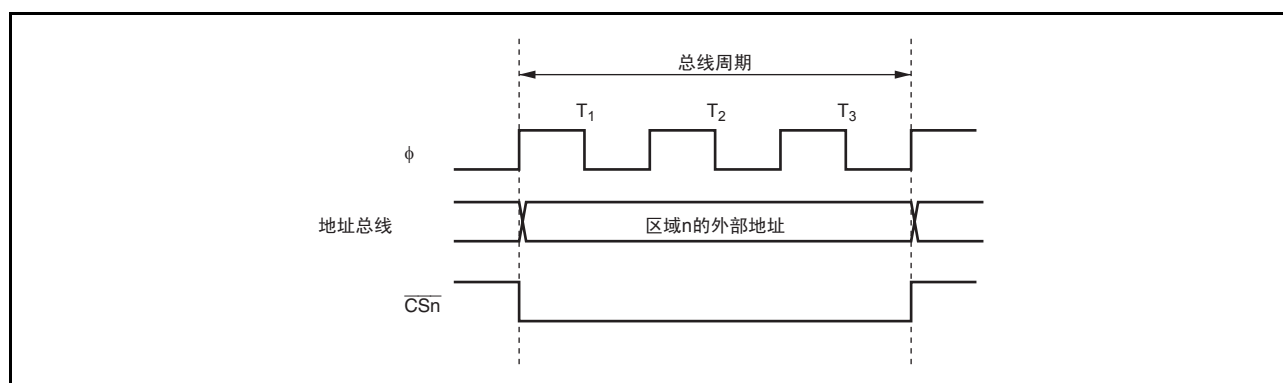


图 6.3 \overline{CSn} 信号输出时序 ($n=0 \sim 5$)

6.5 基本运行时序

本 CPU 的运行以系统时钟 (ϕ) 为标准。将 ϕ 的一个上升开始至下一个上升的 1 个单位称为 1 个状态。存储器周期或总线周期由 1, 2 或 3 个状态所构成, 因内部存储器、内部外围模块或外部地址空间不同, 进行各自不同的存取。

6.5.1 内部存储器 (ROM、RAM) 存取时序

内部存储器的存取进行 1 态存取。此时, 数据总线宽度为 16 位, 可存取字节及字长度。内部存储器的存取周期、管脚状态如图 6.4、图 6.5 所示。

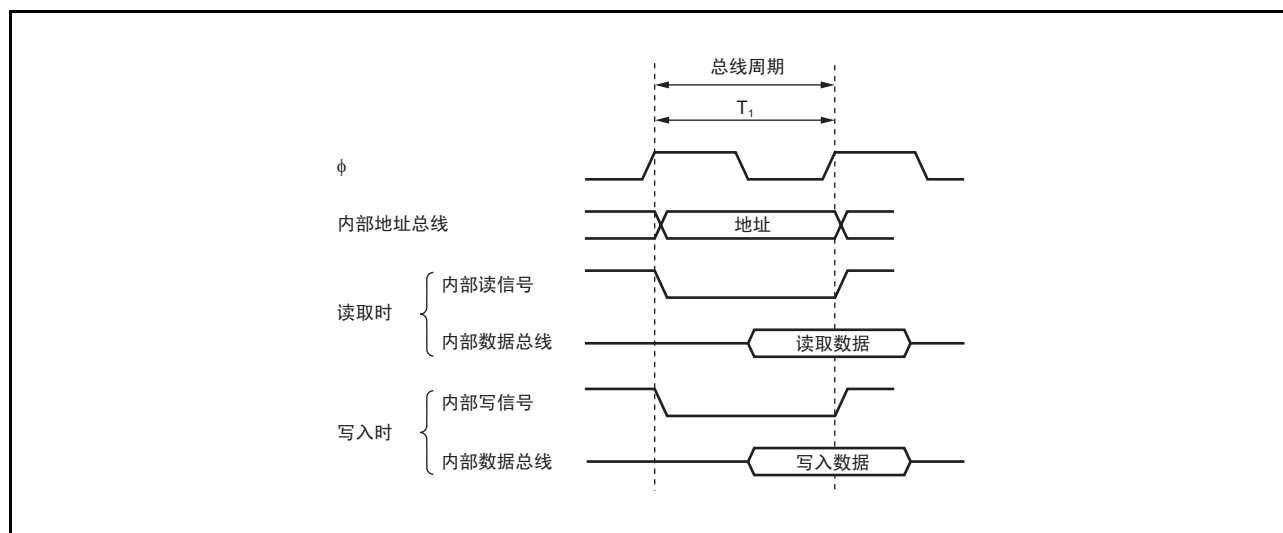


图 6.4 内部存储器存取周期

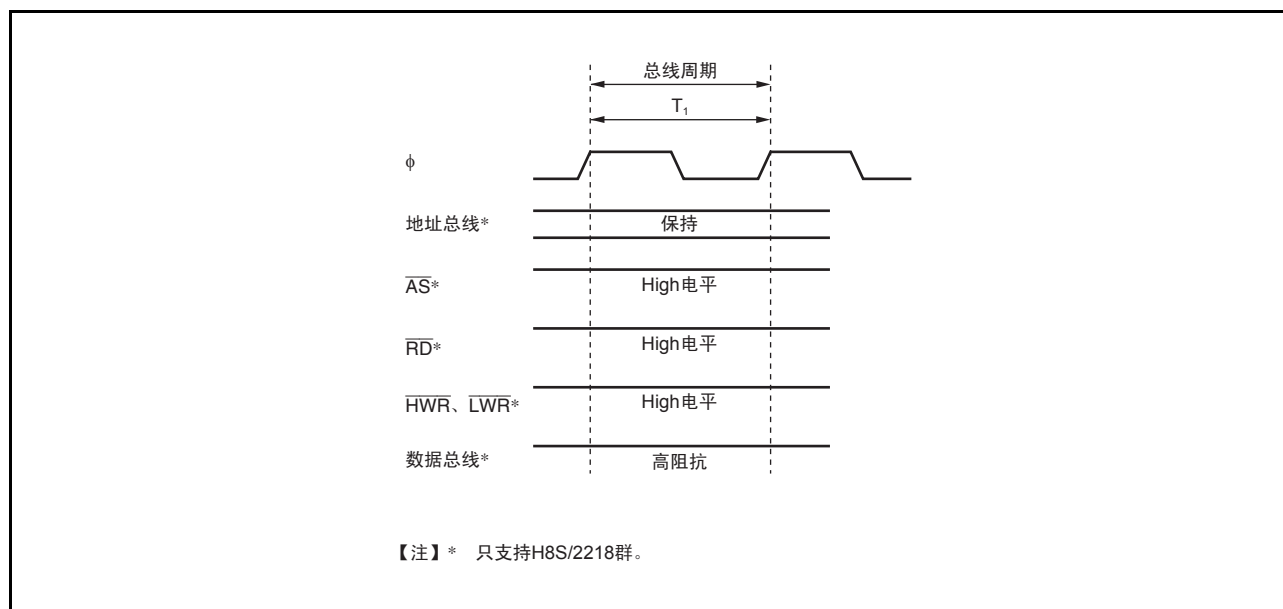


图 6.5 内部存储器存取是的管脚状态

6.5.2 内部外围模块存取时序

内部 USB、RTC 以外的内部外围模块均进行 2 态存取。此时，因内部 I/O 寄存器不同，数据总线宽度为 8 位或 16 位。内部外围模块的存取时序如图 6.6、管脚状态如图 6.7 所示。

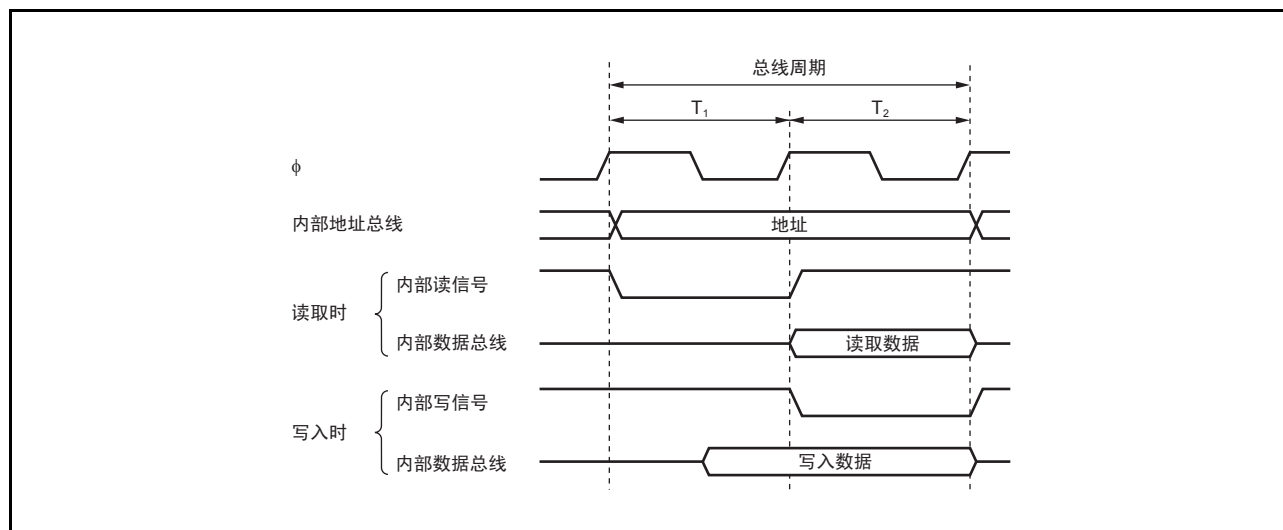


图 6.6 内部外围模块存取周期

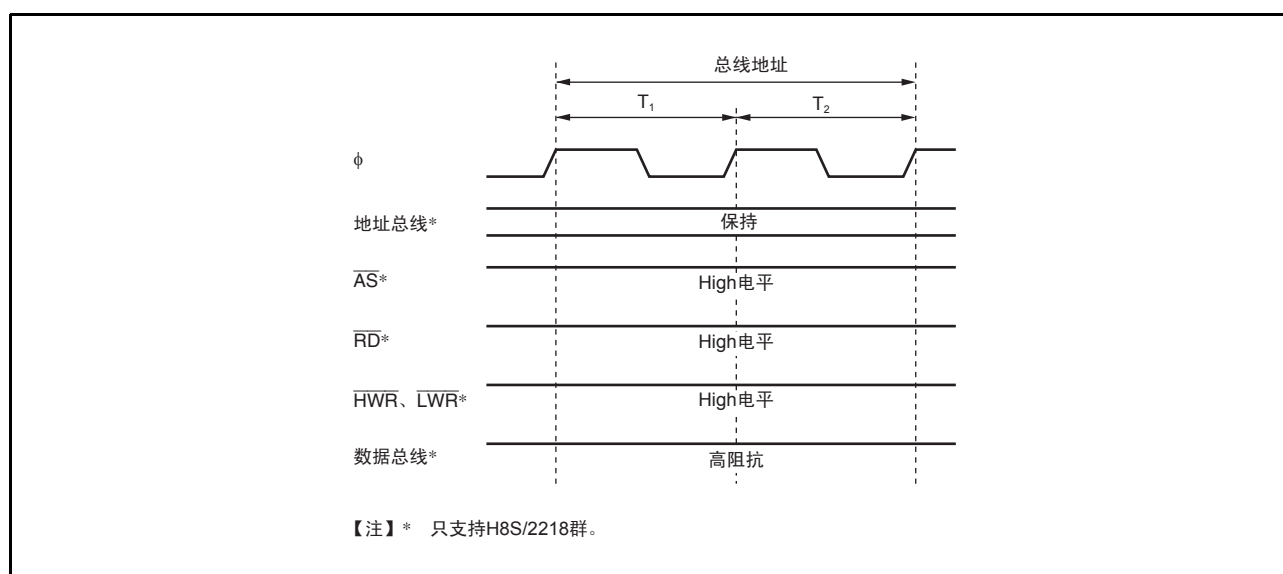


图 6.7 内部外围模块存取时的管脚状态

6.5.3 外部地址空间存取时序

进行外部地址空间存取时的数据总线宽度为 8 位或 16 位、总线周期为 2 态或 3 态。3 态存取可插入等待状态。详细内容请参考 6.6.3 基本时序。

6.6 基本总线接口

基本总线接口可直接连接 ROM, SRAM 等。

6.6.1 数据长度与数据对准 (仅限于 H8S/2218 群使用)

CPU 及其他内部总线主控器的数据长度有字节、字、长字。总线控制器拥有数据对准功能, 存取外部地址空间时可通过存取区域的总线规格 (8 位存取空间或 16 位存取空间) 与数据长度来控制是使用高位侧数据总线 (D15 ~ D8) 还是使用低位侧数据总线 (D7 ~ D0)。

6.6.1.1 8 位存取空间

8 位存取空间的数据对准控制如图 6.8 所示。8 位存取空间常使用高位侧数据总线 (D15 ~ D8) 进行存取。1 次可存取的数据量为 1 字节, 字存取进行 2 次字节存取, 长字存取进行 4 次字节存取。

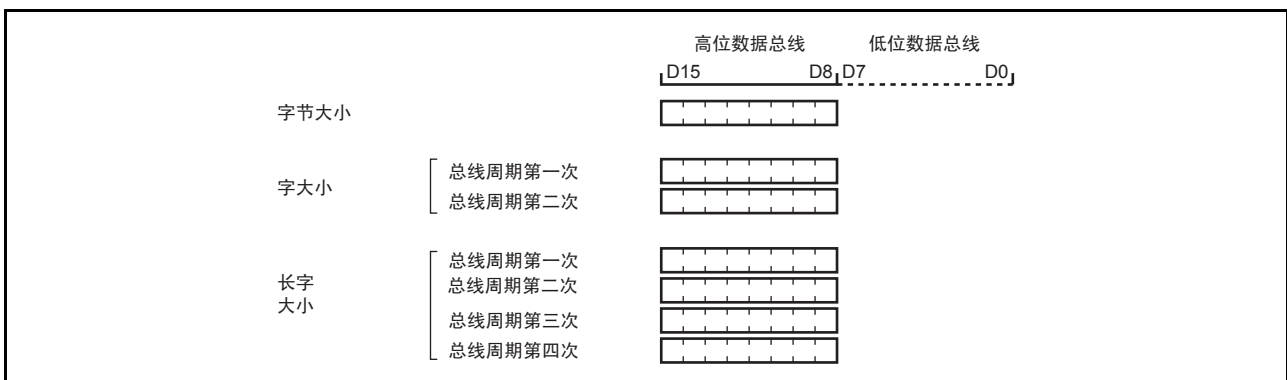


图 6.8 存取长度与数据对准控制 (8 位存取空间)

6.6.1.2 16 位存取空间

16 位存取空间的数据对准控制如图 6.9 所示。16 位存取空间使用高位侧数据总线 (D15 ~ D8) 及低位侧数据总线 (D7 ~ D0) 进行存取。一次可存取的数据量为 1 字节或 1 字, 长字存取进行 2 次字存取。

字节存取时, 通过地址的偶数 / 奇数来决定是使用高位侧数据总线还是使用低位侧数据总线。对偶数地址的字节存取, 使用高位侧数据总线; 对奇数地址的字节存取时, 使用低位侧数据总线。

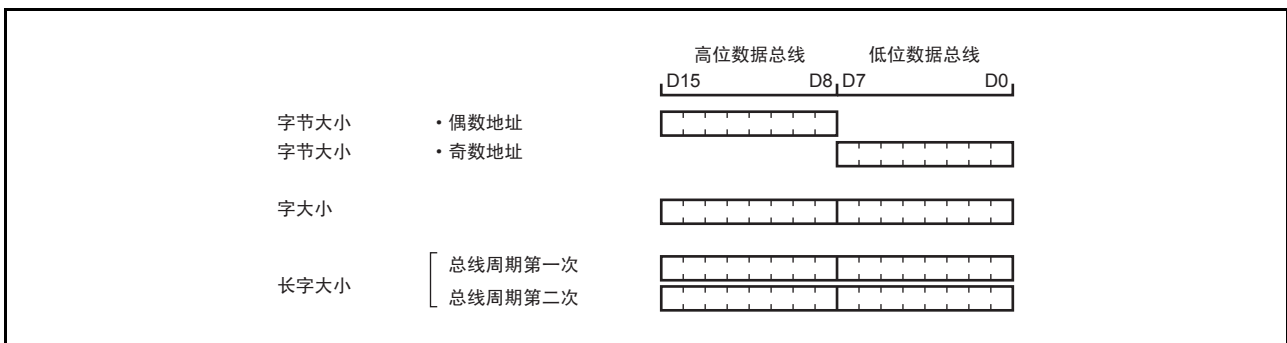


图 6.9 存取长度与数据对准控制 (16 位存取空间)

6.6.2 有效选通

H8S/2218 群的存取空间、使用的数据总线及有效选通如表 6.3 所示。

读取时， \overline{RD} 信号有效，无数据总线高位、低位的区别。

写入时，对于数据总线的高位侧， \overline{HWR} 信号有效；对于低位侧， \overline{LWR} 信号有效。

H8S/2212 群无 \overline{RD} 、 \overline{HWR} 、 \overline{LWR} 信号。

表 6.3 使用数据总线及有效选通

区域	存取长度	读取 / 写入	地址	有效选通	数据总线高位 (D15 ~ D8)	数据总线低位 (D7 ~ D0)
8 位 存取空间	字节	读取	—	\overline{RD}	有效	无效
		写入	—	\overline{HWR}	有效	Hi-Z
16 位 存取空间	字节	读取	偶数	\overline{RD}	有效	无效
			奇数	\overline{RD}	无效	有效
		写入	偶数	\overline{HWR}	有效	Hi-Z
			奇数	\overline{LWR}	Hi-Z	有效
	字	读取	—	\overline{RD}	有效	有效
		写入	—	\overline{HWR} 、 \overline{LWR}	有效	有效

【注】 Hi-Z: 为高阻抗状态。

无效: 为输入状态, 输入值忽视。

6.6.3 基本时序

6.6.3.1 8 位 2 态存取空间

H8S/2218 群的 8 位 2 态存取空间总线时序如图 6.10 所示。存取 8 位存取空间时，使用高位侧 (D15 ~ D8) 的数据总线。

不可插入等待状态。

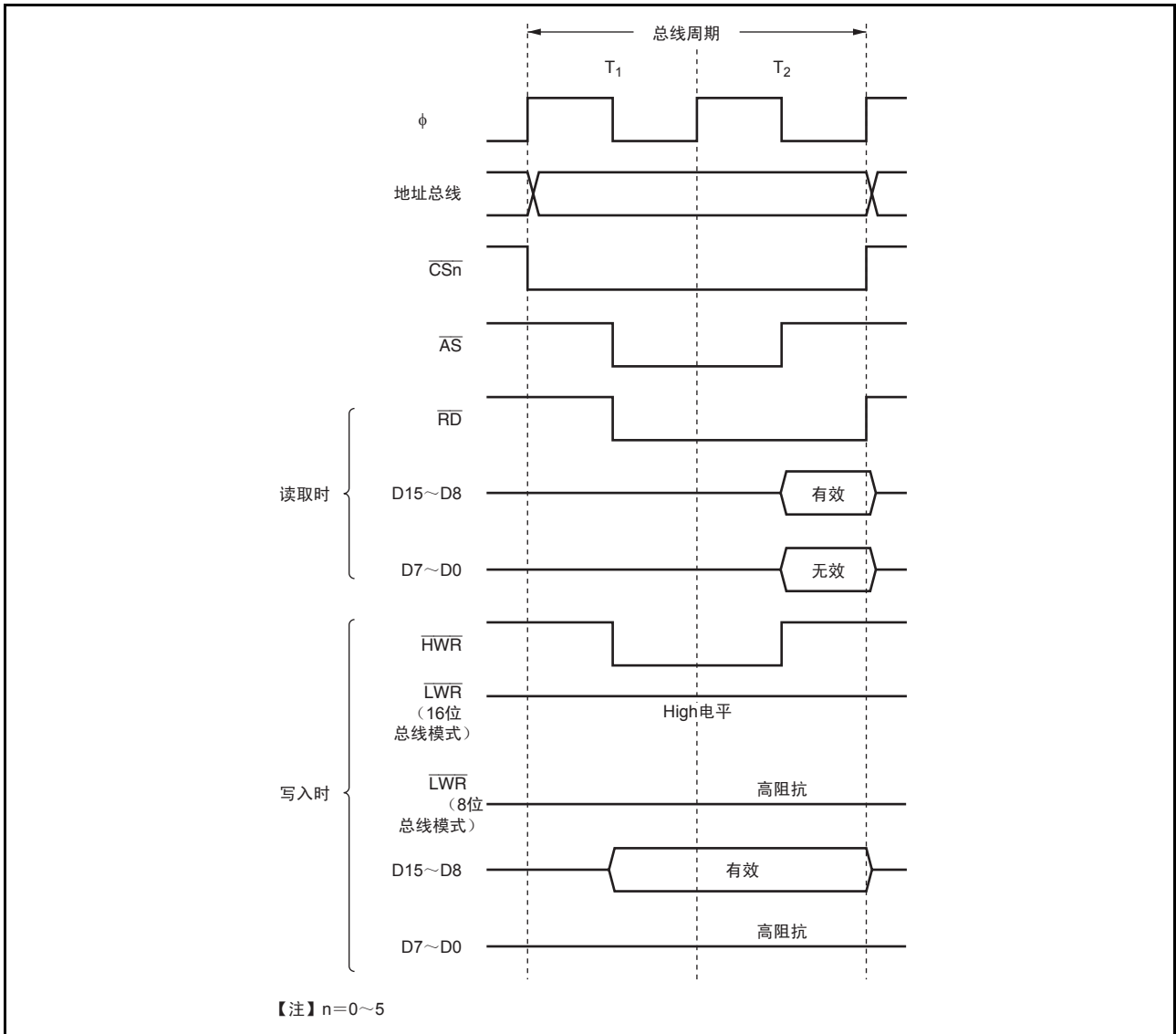


图 6.10 8 位 2 态存取空间总线时序

6.6.3.2 8 位 3 态存取空间 (区域 6 以外)

H8S/2218 群的 8 位 3 态存取空间总线时序如图 6.11 所示。存取 8 位存取空间时，使用高位 (D15 ~ D8) 的数据总线。

可插入等待状态。

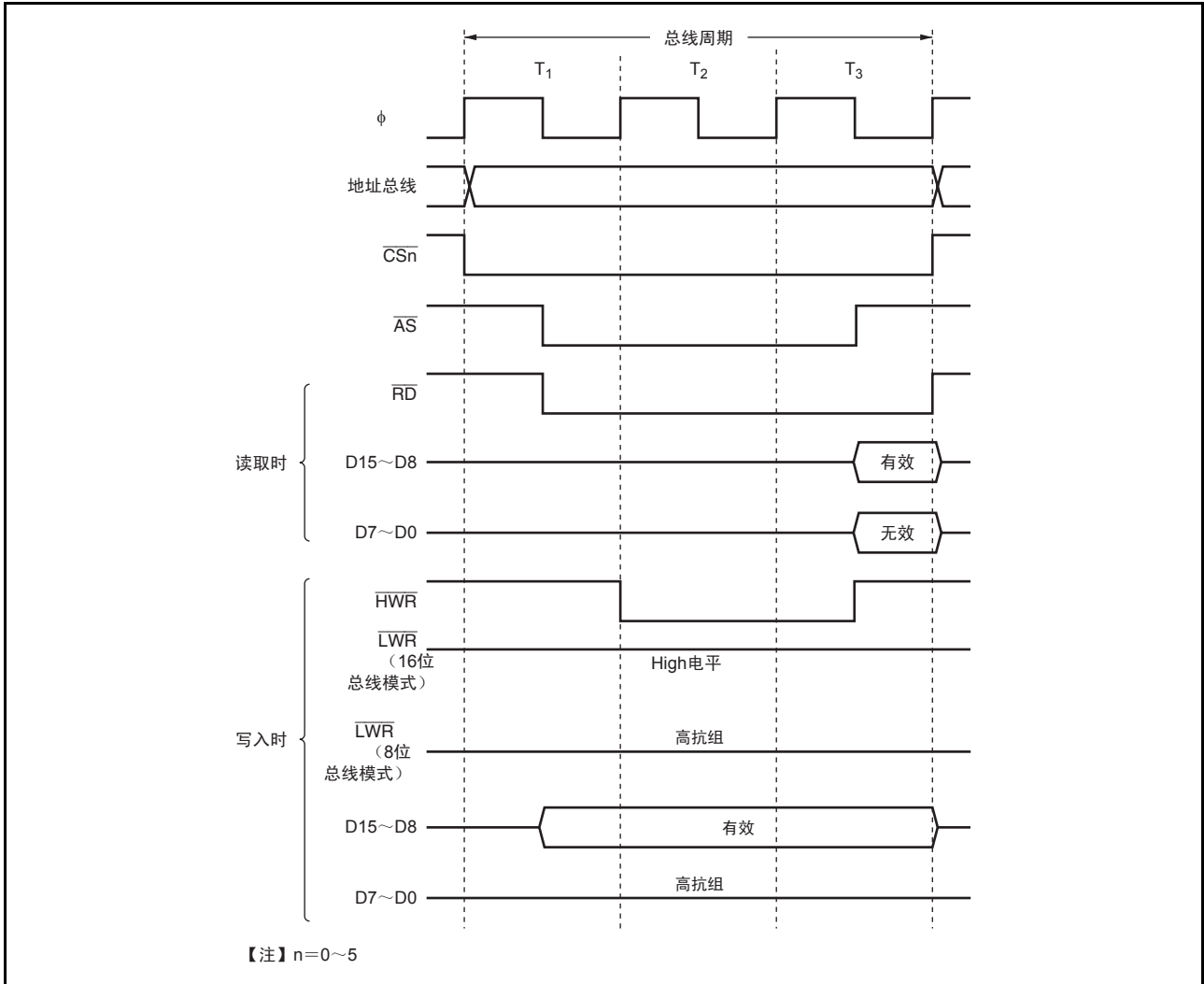


图 6.11 8 位 3 态存取空间总线 (区 6 以外)

6.6.3.3 8 位 3 态存取空间 (区域 6 及 RTC)

区域 6 及 RTC 区域 (地址 = H'FFFF40 ~ H'FFFF5F) 的总线时序如图 6.12 所示。存取这些区域时, 不可使用数据总线。

不可插入等待状态。

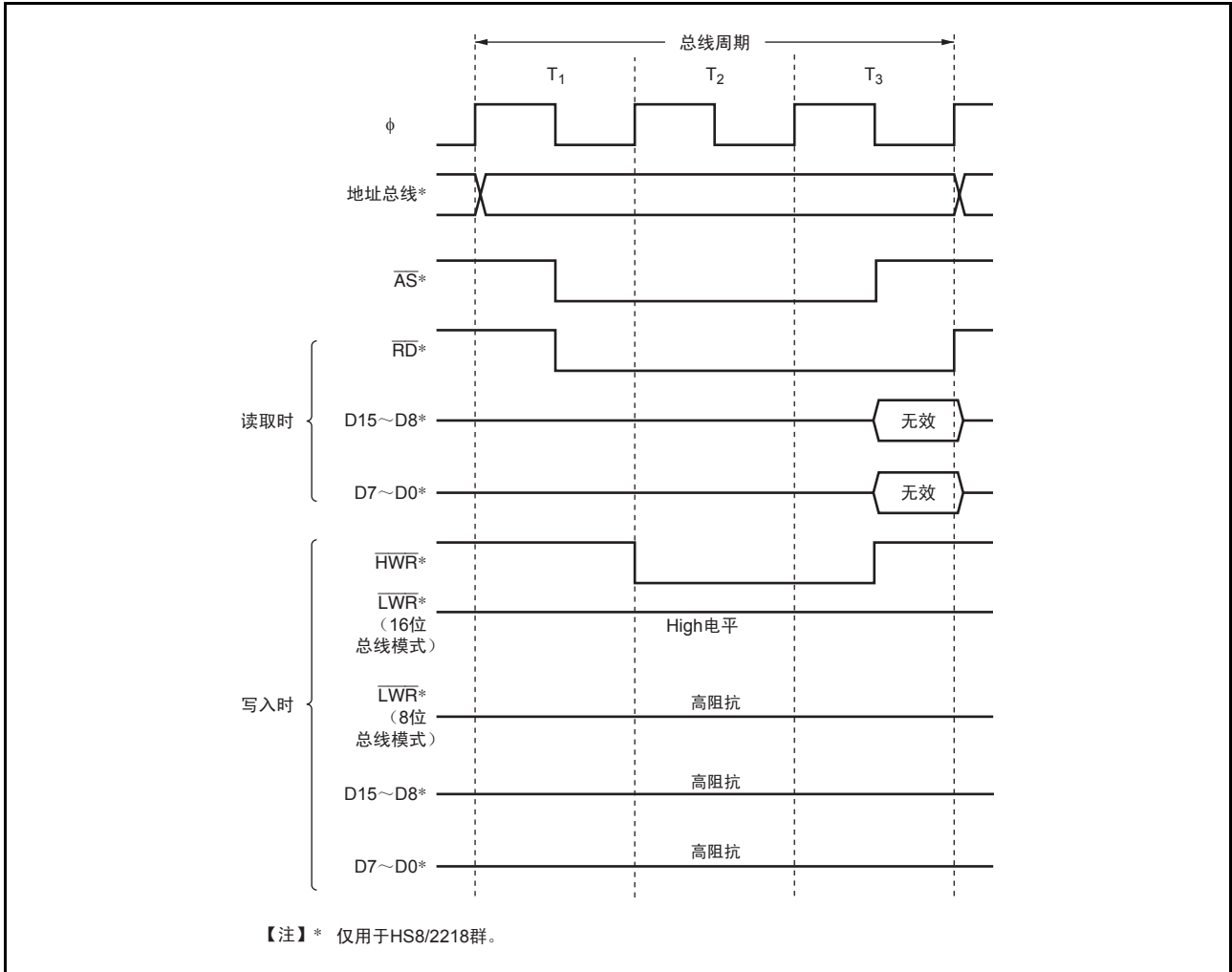


图 6.12 区域 6 及 RTC 的总线时序

6.6.3.4 16 位 2 态存取空间

H8S/2218 群的 16 位 2 态存取空间总线时序如图 6.13 ~ 图 6.15 所示。存取 16 位存取空间时，对偶数地址，使用高位侧数据总线 (D15 ~ D8)；对奇数地址，使用低位数据总线 (D7 ~ D0)。不可插入等待状态。

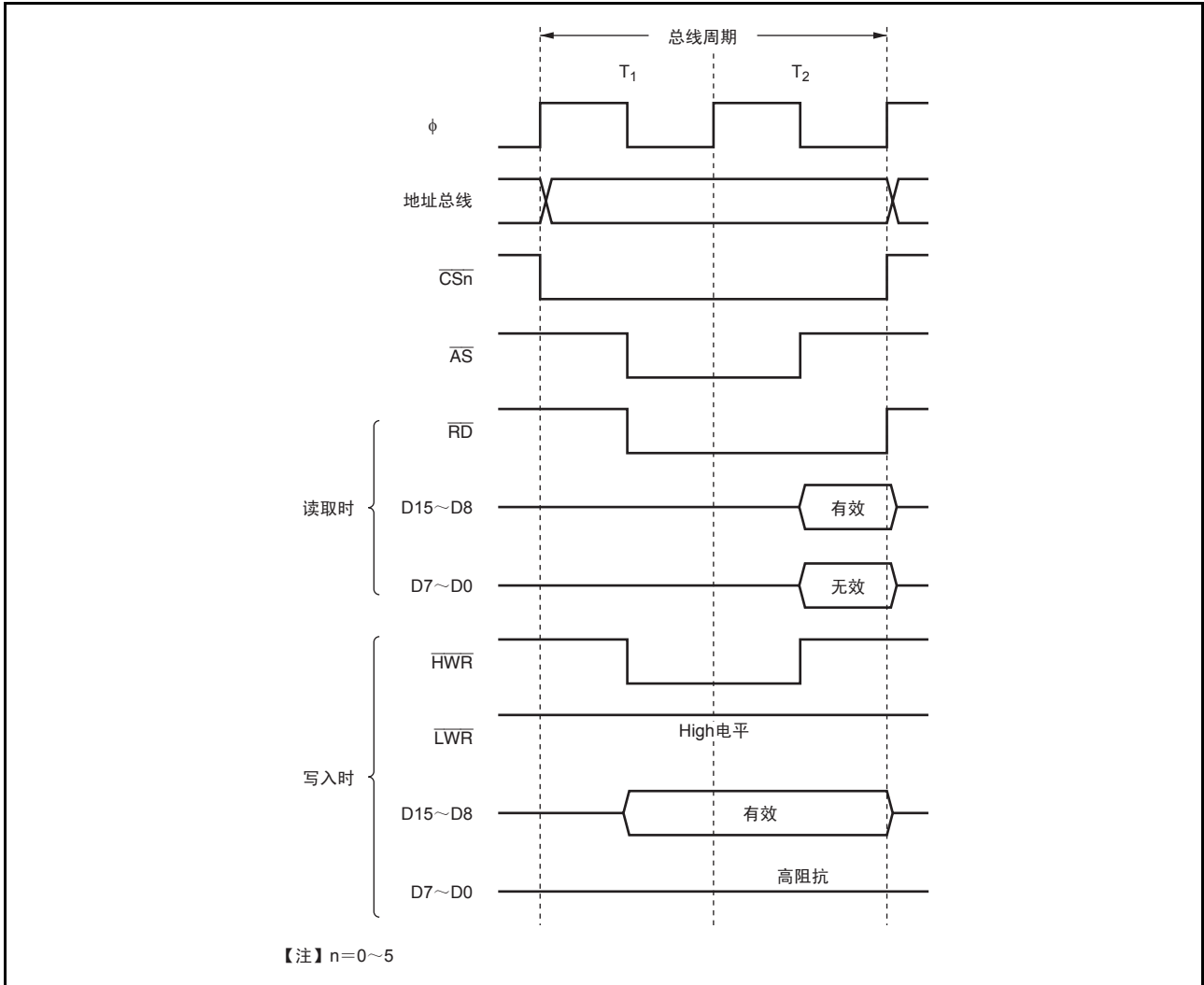


图 6.13 16 位 2 态存取空间的总线时序 (1) (偶数地址字节存取)

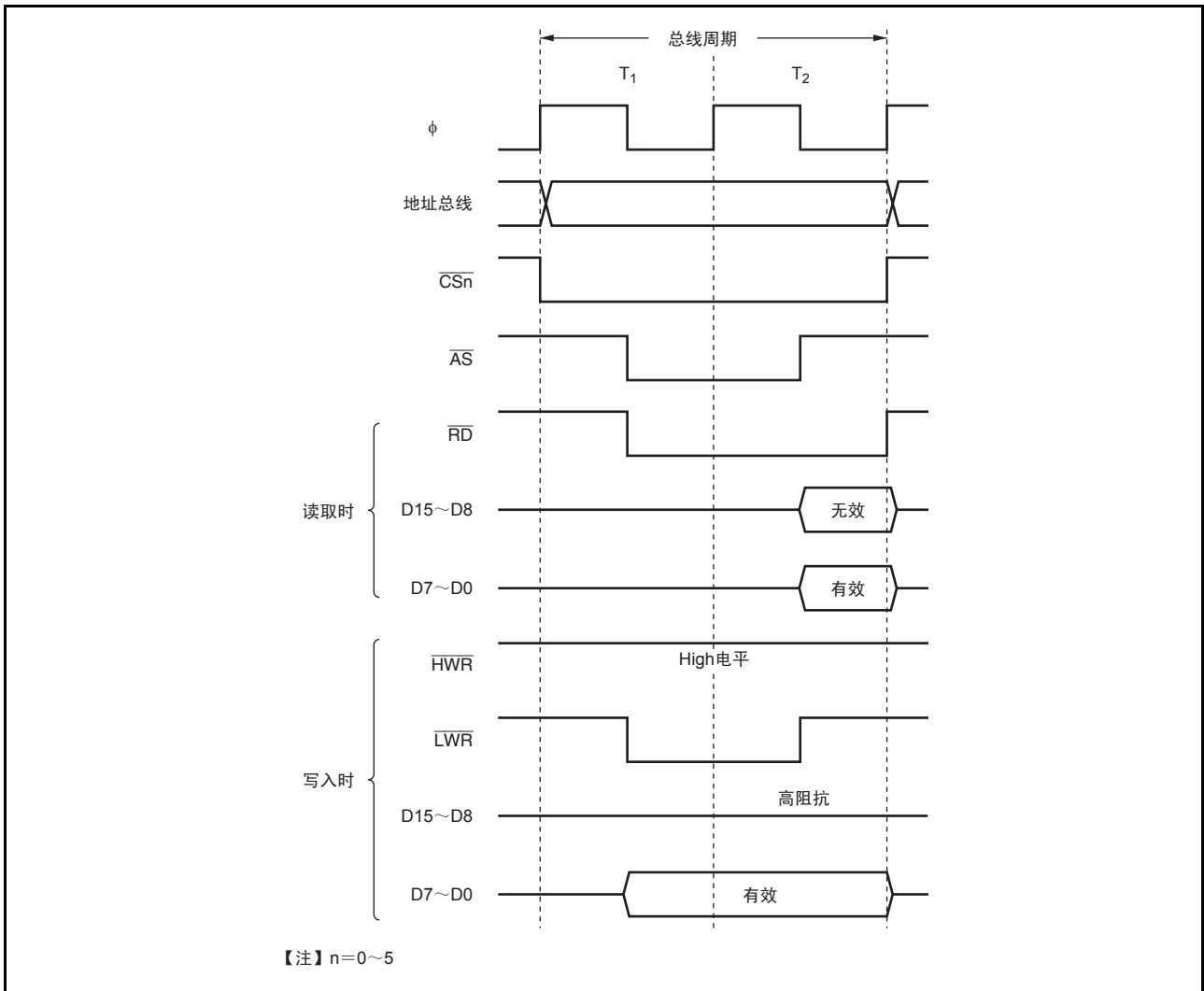


图 6.14 16 位 2 态存取空间的总线时序 (2) (奇数地址字节存取)

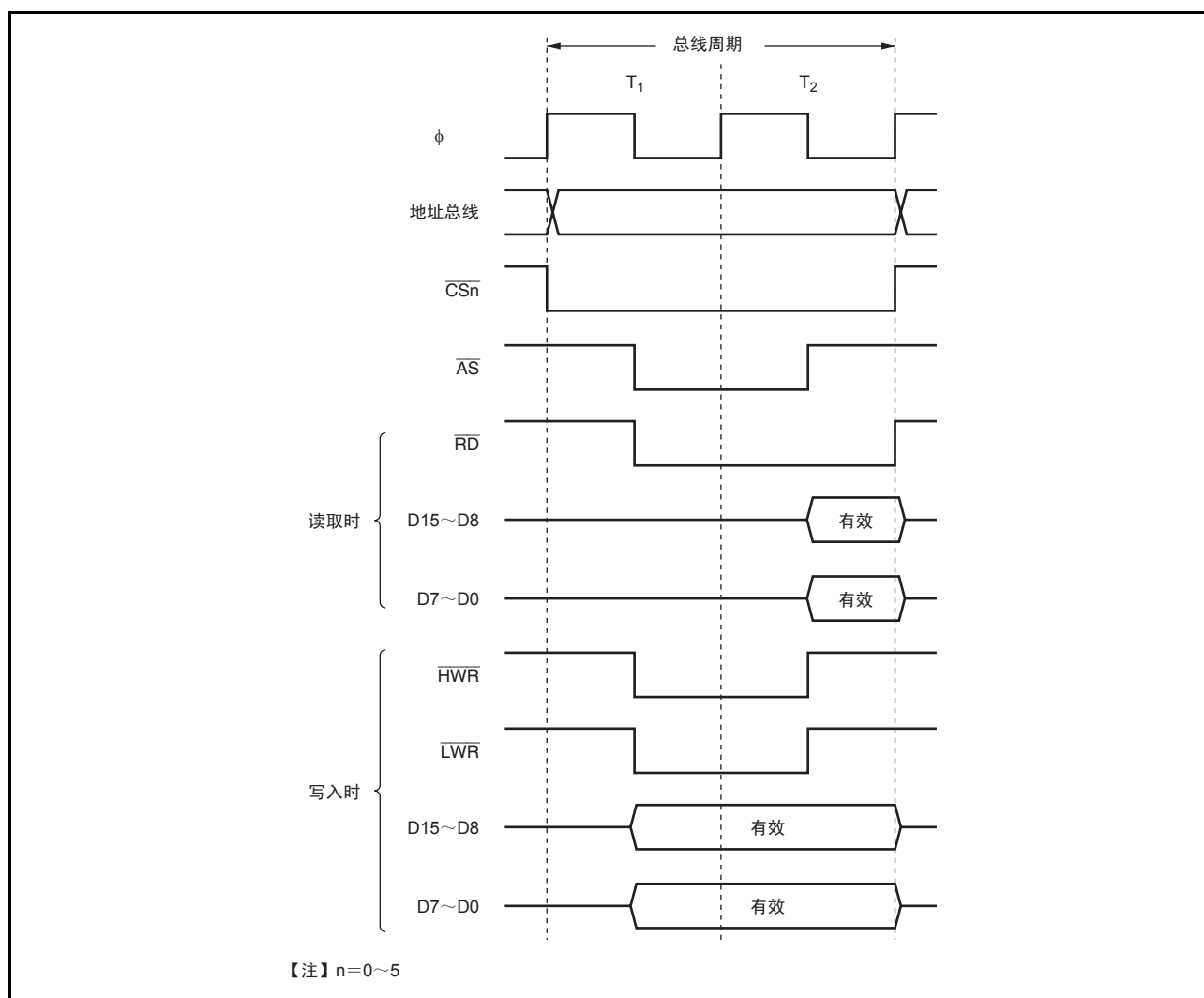


图 6.15 16 位 2 态存取空间的总线时序 (3) (字存取)

6.6.3.5 16 位 3 态存取空间

H8S/2218 群的 16 位 3 态存取空间总线时序如图 6.16 ~ 图 6.18 所示。存取 16 位存取空间时，对偶数地址，使用高位侧数据总线 (D15 ~ D8)，对奇数地址，使用低位侧数据总线 (D7 ~ D0)。可插入等待状态。

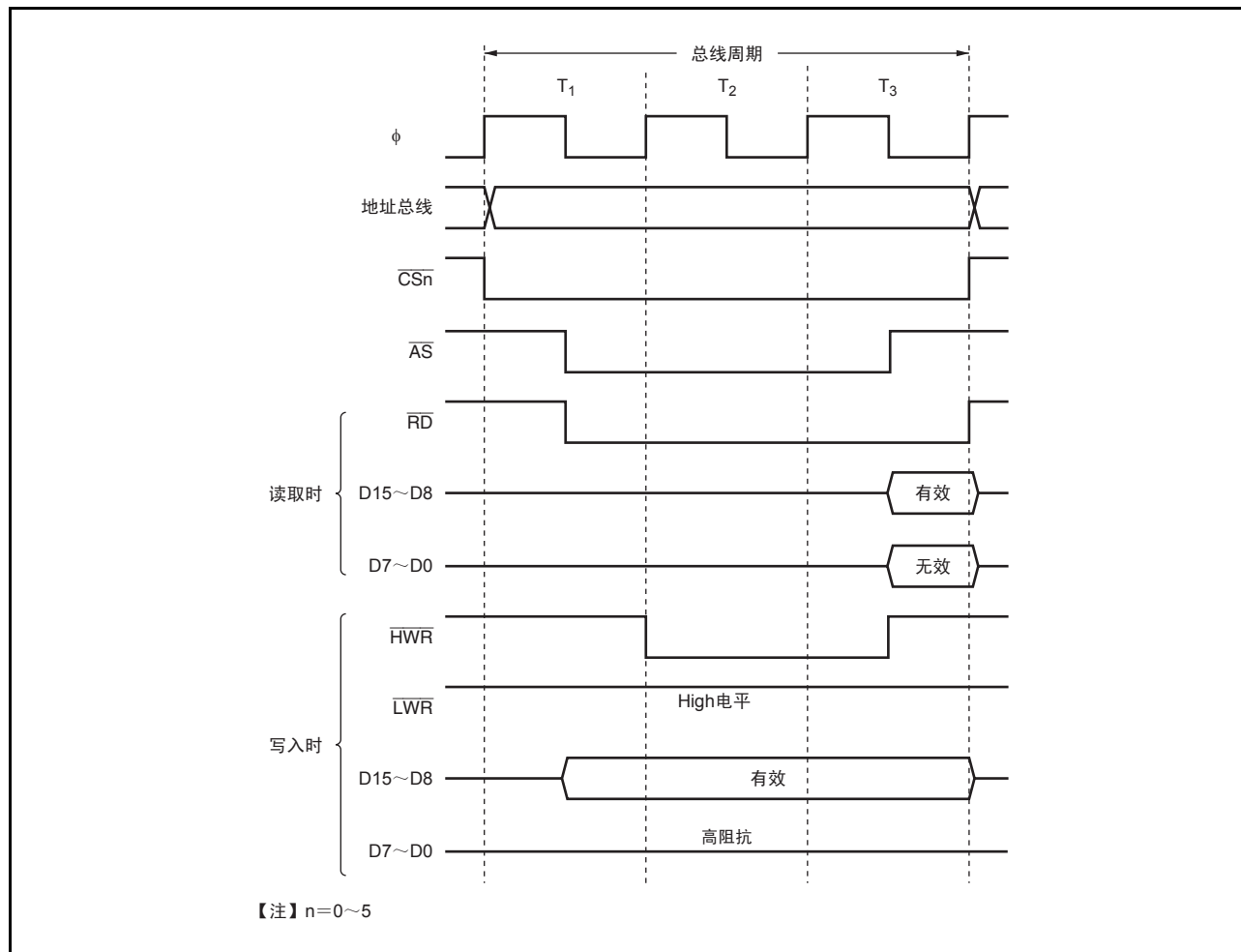


图 6.16 16 位 3 态存取空间的总线时序 (1) (偶数地址字节存取)

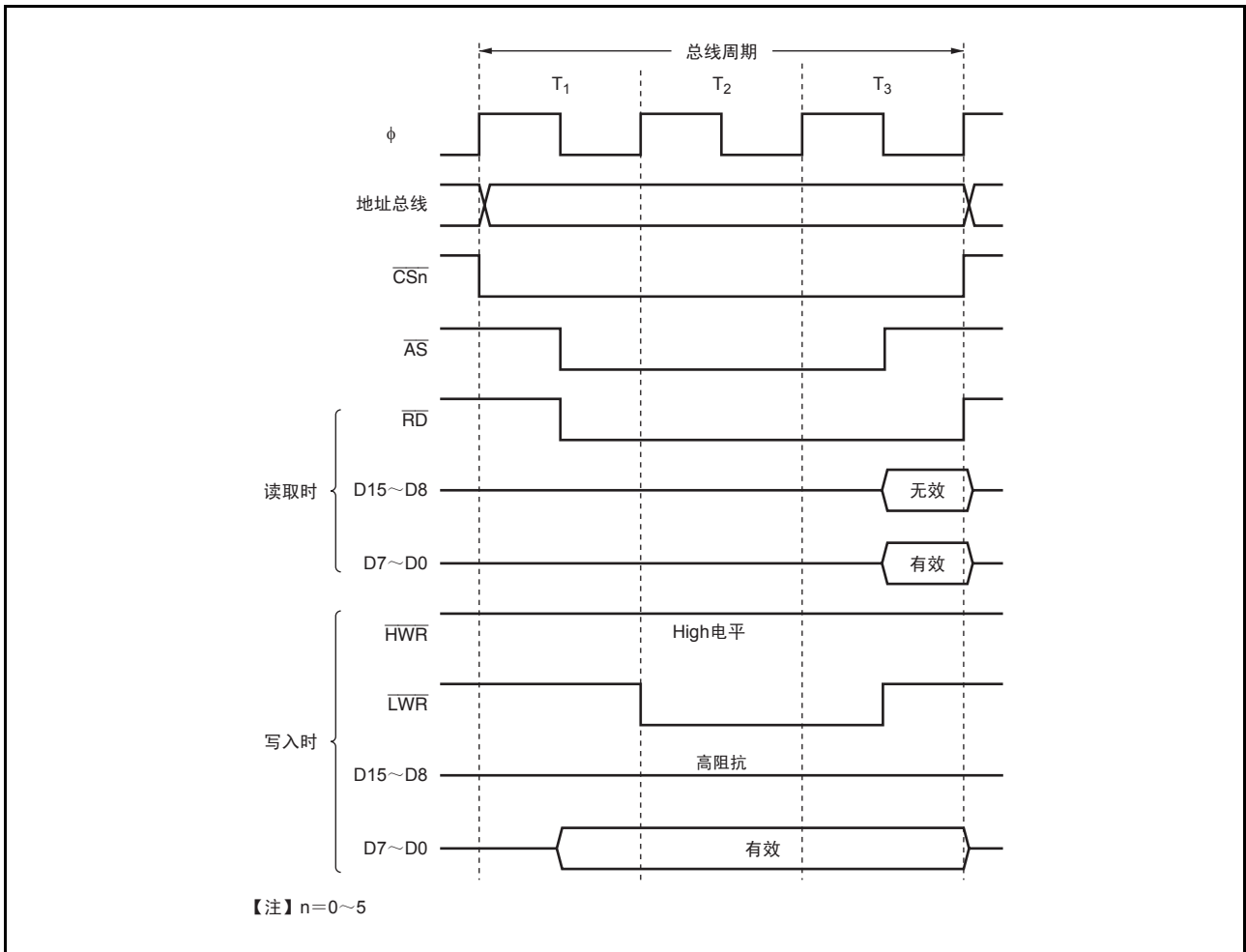


图 6.17 16 位 3 态存取空间的总线时序 (2) (奇数地址字节存取)

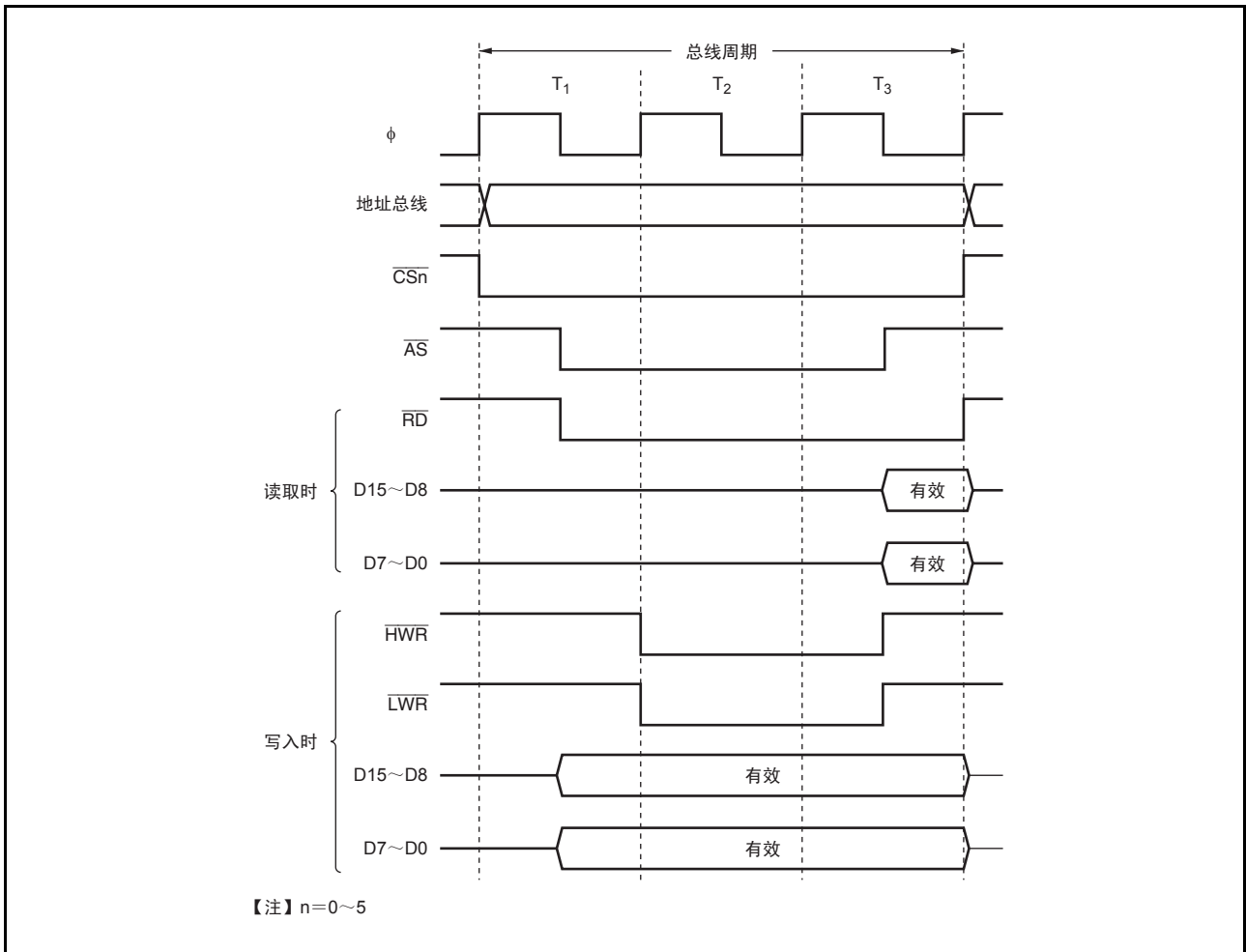


图 6.18 16 位 3 态存取空间的总线时序 (3) (字存取)

6.6.4 等待控制

存取外部地址空间时，本 LSI 可插入等待状态 (T_w) 来延长总线周期。

6.6.4.1 程序等待的插入

可通过设定 WCRH、WCRL，向 3 态存取空间，以区域为单位在 T2 状态与 T3 状态之间自动插入 0 ~ 3 个等待状态。

6.6.4.2 管脚等待的插入

H8S/2218 群中，将 BCRH 的 WAITE 位置位为 1 时，通过 $\overline{\text{WAIT}}$ 管脚的等待输入为有效。此状态下，存取外部地址空间时，首先按照 WCRH、WCRL 的设定来插入程序等待。接着，在 T_2 或 T_w 的最后状态的 ϕ 下降时序， $\overline{\text{WAIT}}$ 管脚为低电平时，插入 T_w 。 $\overline{\text{WAIT}}$ 管脚保持低电平时，插入 T_w 至 $\overline{\text{WAIT}}$ 管脚成为高电平。

等待状态插入时序例如图 6.19 所示。

H8S/2212 群中，请勿将 BCRH 的 WAITE 位置位为 1。

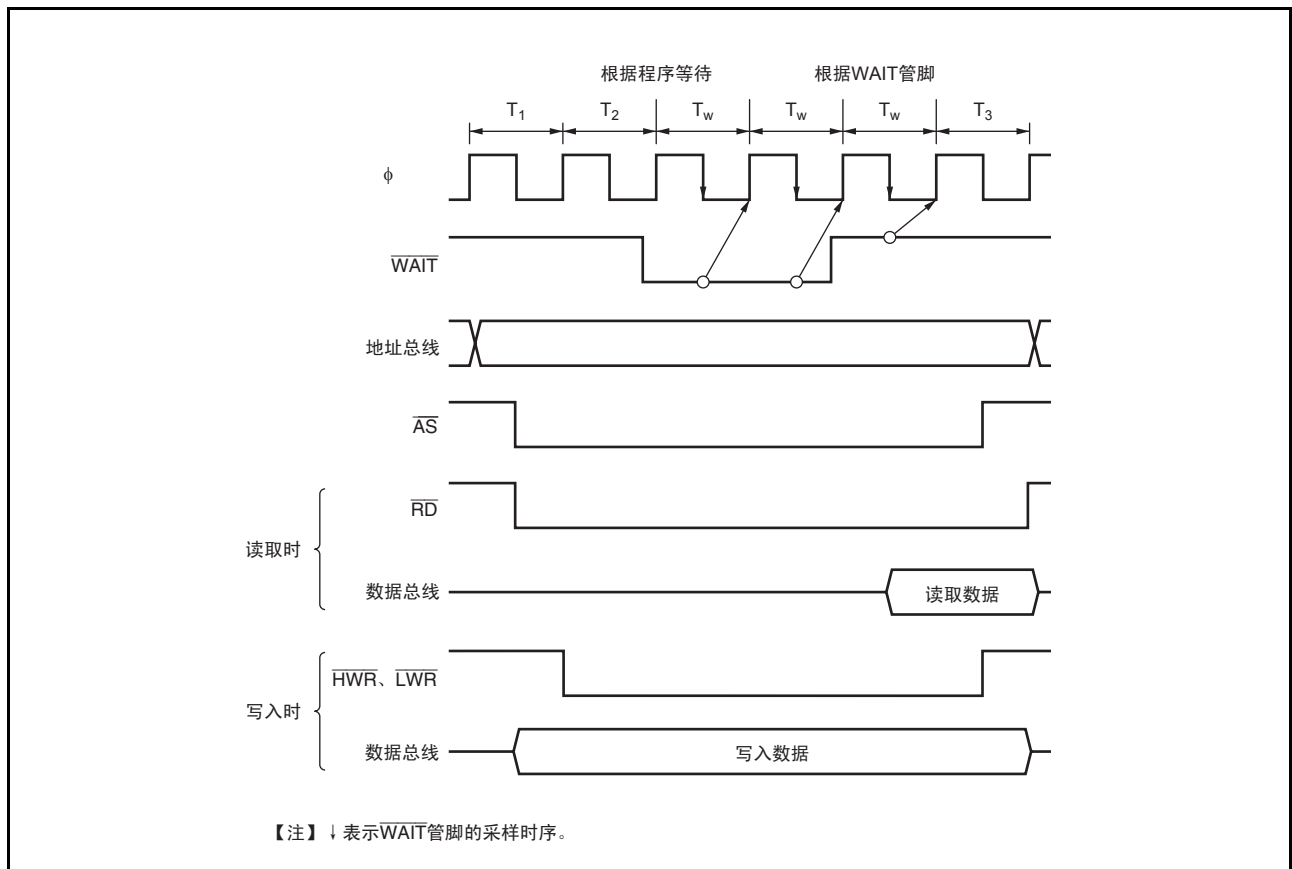


图 6.19 等待状态插入时序例

6.7 突发式 ROM 接口

H8S/2218 群中，可将区域 0 的外部地址空间设定为突发式 ROM 空间，执行突发式 ROM 接口。突发式 ROM 空间接口可高速存取 16 位结构可突发存取的 ROM。

通过 BCRH 的 BRSTRM 位，将区域 0 设定为突发式 ROM 空间。仅限于 CPU 取指令，可进行最大 4 字 /8 字的连续突发存取。突发存取可选择 1 态或 2 态。

6.7.1 基本时序

突发式 ROM 接口的初始周期（全部存取）存取状态数根据 ASTCR 的 AST0 位设定。另外，AST0 位置位为 1 时，可插入等待状态。通过设定 BCRH 的 BRSTS1 位，突发周期可选择 1 态或 2 态。不可插入等待状态。将区域 0 设定为突发式 ROM 空间时，至 ABWCR 的 ABW0 位设定无关，区域 0 成为 16 位存取空间。

另外，BCRH 的 BRSTS0 位清零时，至最大 4 字的突发存取。BRSTS0 位置位为 1 时，进行最大 8 字的突发存取。

突发式 ROM 空间的基本存取时序如图 6.20、图 6.21 所示。

AST0 位、BRSTS1 位设定为 1 例如图 6.20 所示。

AST0 位，BRSTS1 位设定为 0 例如图 6.21 所示。

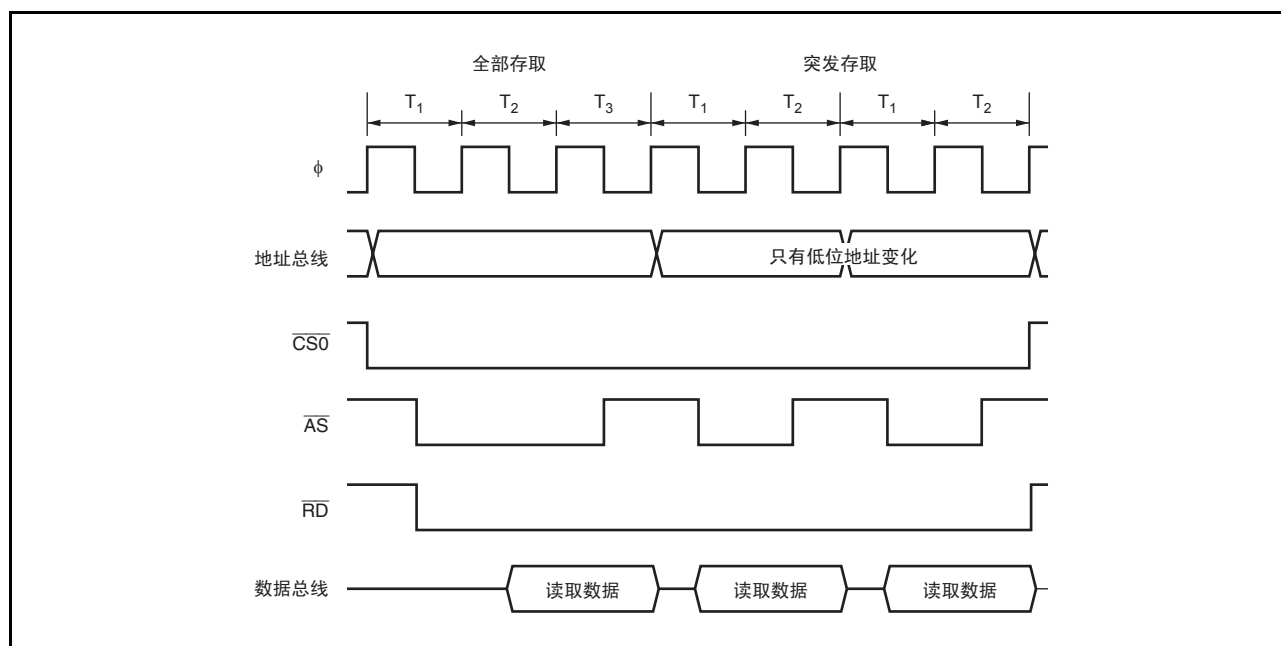


图 6.20 突发式 ROM 存取时序例 (AST0 = BRSTS1 = 1 时)

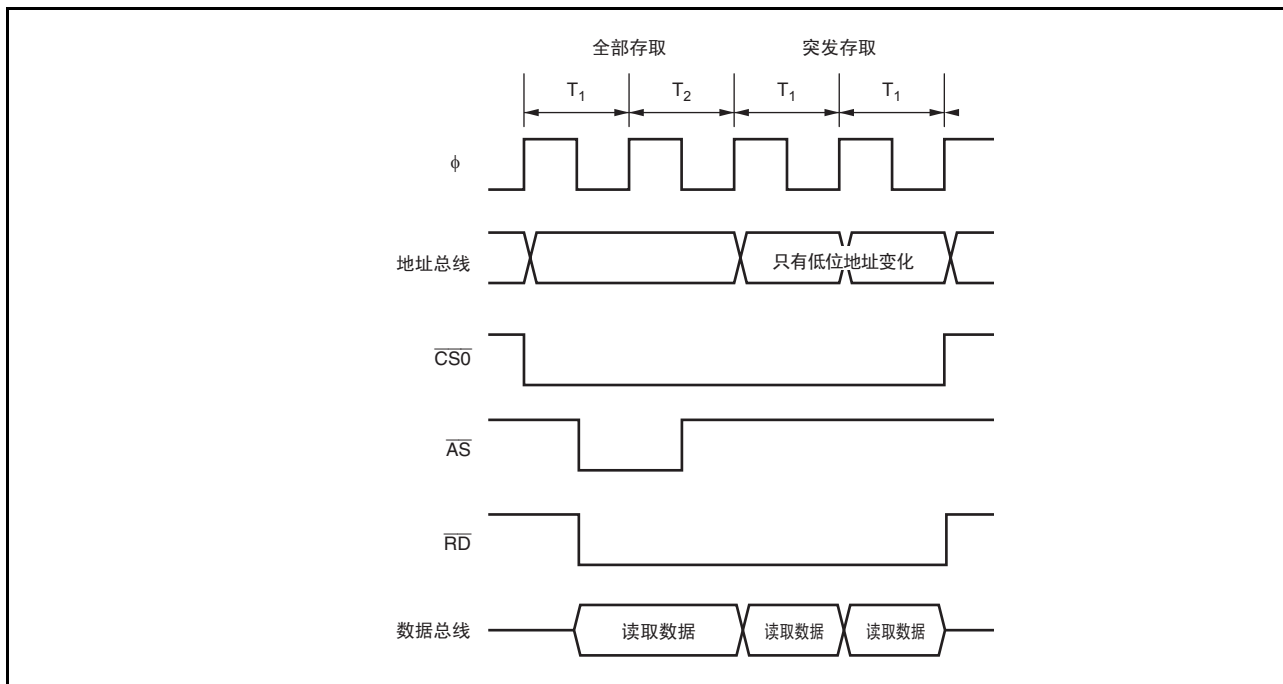


图 6.21 突发式 ROM 存取时序例 (AST0 = BRSTS1 = 0 时)

6.7.2 等待控制

突发式 ROM 接口的初始周期 (全部存取) 与基本总线接口相同均可 (1) 插入程序等待 (2) 插入 WAIT 管脚的管脚等待。请参考 6.6.4 等待控制。

突发周期不可插入等待状态。

6.8 空闲周期

H8S/2218 群中, 存取外部地址空间时, 在以下 2 种情况下, 可在总线周期与总线周期之间插入 1 态空闲周期 (T_1), 上述 2 种情况为 (1) 在不同区域间连续发生读取存取时, (2) 在读取周期后直接发生写入周期时。通过插入空闲周期, 可以防止诸如输出浮动时间长的 ROM 等与高速存储器、I/O 接口等之间的数据冲突。

6.8.1 不同区域间的连续读取

BCRH 的 ICIS1 位置位为 1 的状态下, 发生不同区域间的连续读取时, 第 2 次读取周期的开头, 插入空闲周期。

运行例如图 6.22 所示。此列为总线周期 A 为输出浮动时间较长的 ROM 的读取周期、总线周期 B 为 SRAM 的读取周期, 向各自不同区域分配的例。(a) 不插入空闲周期时, ROM 的读取数据与 SRAM 的读取数据, 将会在总线周期 B 中发生冲突。(b) 插入空闲周期将可避免数据冲突。

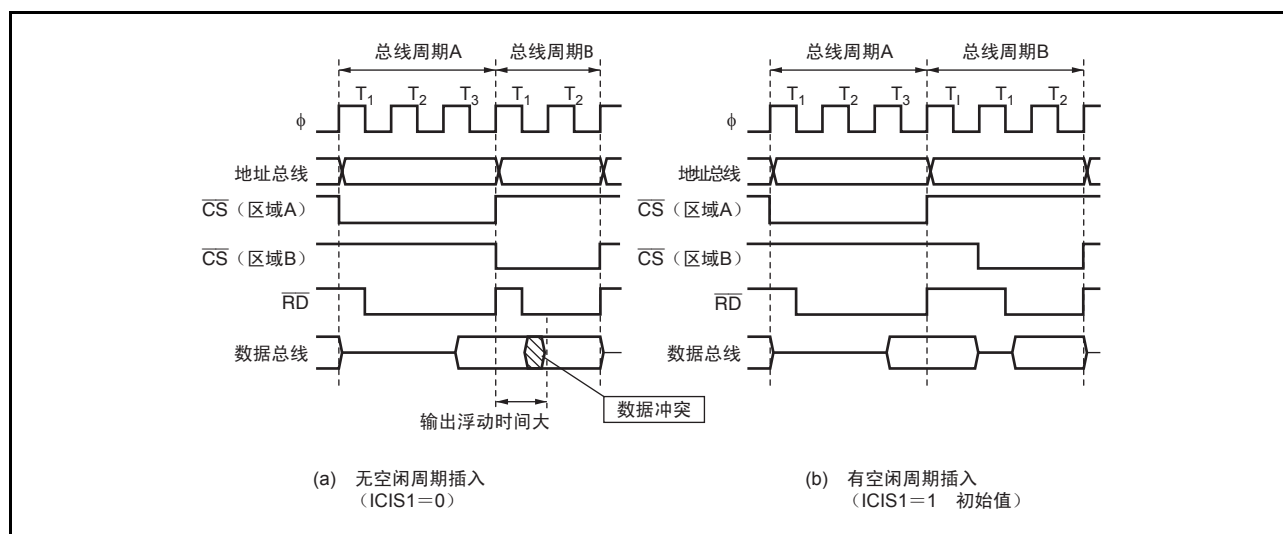


图 6.22 空闲周期运行例 (1)

6.8.2 读取后的写入

BCRH 的 ICIS0 位置位为 1 的状态下，外部读取后发生外部写入时，写入周期的开头插入空闲周期。

运行例如图 6.23 所示。此例为总线周期 A 为输出浮动时间较长的 ROM 的读取周期、总线周期 B 为 CPU 的写入周期例。(a) 不插入空闲周期时，ROM 的读取数据与 CPU 的写入数据，将会在总线周期 B 中发生冲突。(b) 插入空闲周期将避免数据冲突。

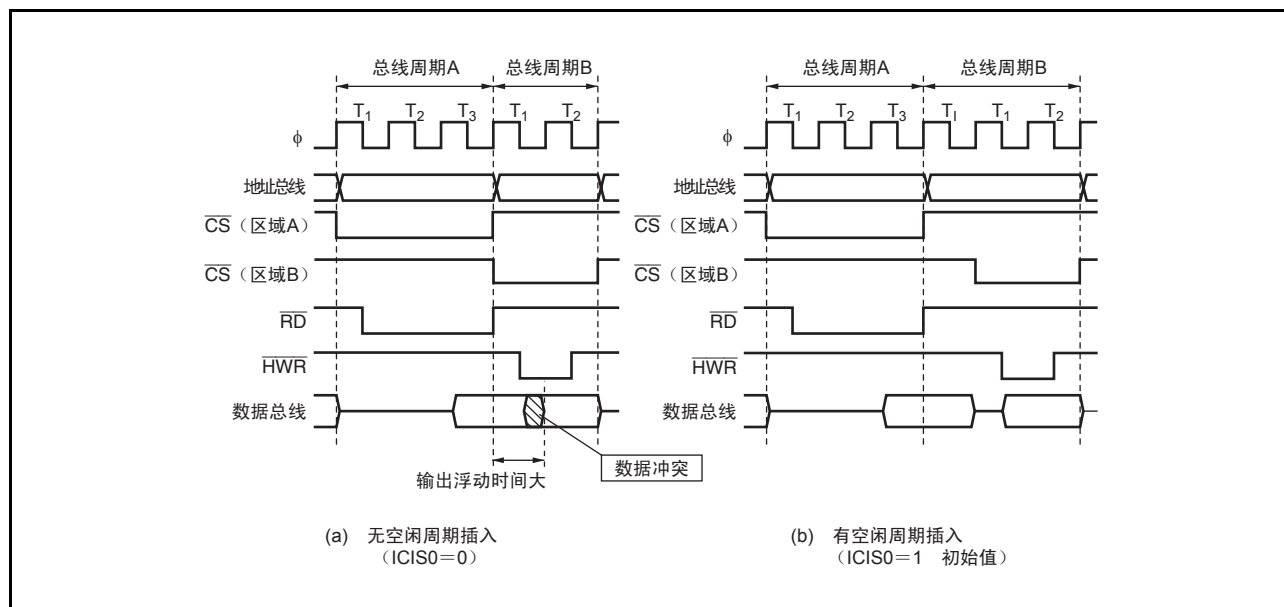


图 6.23 空闲周期运行例 (2)

6.8.3 片选 (CS) 信号与读取 (RD) 信号的关系

如图 6.24 所示。由于系统的负载条件，有时 RD 信号会比 CS 信号晚。在此情况下，不插入 (a) 的空闲周期的设定，总线周期 A 的 RD 信号与总线周期 B 的 CS 信号之间有可能有一段期间重叠。

对此，如果像 (b) 那样，设定插入空闲周期，可避免 RD 信号与 CS 信号之间的重叠期间。此外，复位解除后的初始状态，设定插入 (b) 空闲周期。

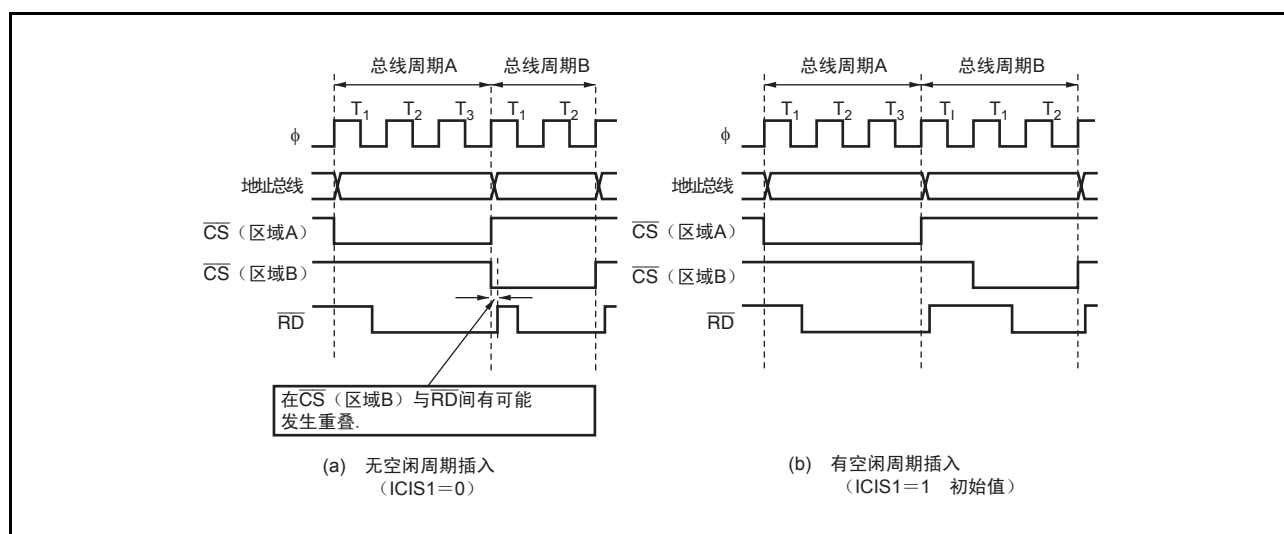


图 6.24 片选 (CS) 与读取 (RD) 的关系

空闲周期的管脚状态如表 6.4 所示。

表 6.4 空闲周期的管脚状态

管脚名称	管脚状态
A23 ~ A0	下一个总线周期的内容
D15 ~ D0	高阻抗
$\overline{\text{CSn}}$	高电平
$\overline{\text{AS}}$	高电平
$\overline{\text{RD}}$	高电平
$\overline{\text{HWR}}$	高电平
$\overline{\text{LWR}}$	高电平

6.9 总线释放

H8S/2218 群可通过外部总线权请求，释放外部总线。外部总线释放状态下，只要不发生外部存取，内部总线主控器就连续运行。

外部扩展模式中，将 BCRL 的 BRLE 位置位为 1，可将总线权释放给外部。通过将 $\overline{\text{BREQ}}$ 管脚设定为低电平来向本 LSI 请求外部总线权。对 $\overline{\text{BREQ}}$ 管脚进行采样时，在所定时序中，将 $\overline{\text{BACK}}$ 管脚被设定为低电平，将地址总线、数据总线、总线控制信号被设定为高阻抗，成为外部总线释放状态。

外部总线权释放状态下，内部总线主控器可进行使用内部总线的存取。内部总线主控器要进行外部存取时，如果保留总线周期的启动，就会等待撤回外部主控器的总线权请求。

将 $\overline{\text{BREQ}}$ 管脚设定为高电平时，在所定时序中，将 $\overline{\text{BACK}}$ 管脚设定为高电平，结束外部总线权释放状态。外部总线权释放请求与外部存取同时发生时的优先顺序：

(高) 外部总线权 > 内部总线主控器的外部存取 (低)

外部总线权释放状态下的管脚状态如表 6.5 所示。

H8S/2212 群请勿将 BCRL 的 BRLE 位置位为 1。

表 6.5 总线权释放状态下的管脚状态

管脚名称	管脚状态
A23 ~ A0	高阻抗
D15 ~ D0	高阻抗
$\overline{\text{CSn}}$	高阻抗
$\overline{\text{AS}}$	高阻抗
$\overline{\text{RD}}$	高阻抗
$\overline{\text{HWR}}$	高阻抗
$\overline{\text{LWR}}$	高阻抗

总线权释放状态转换时序如图 6.25 所示。

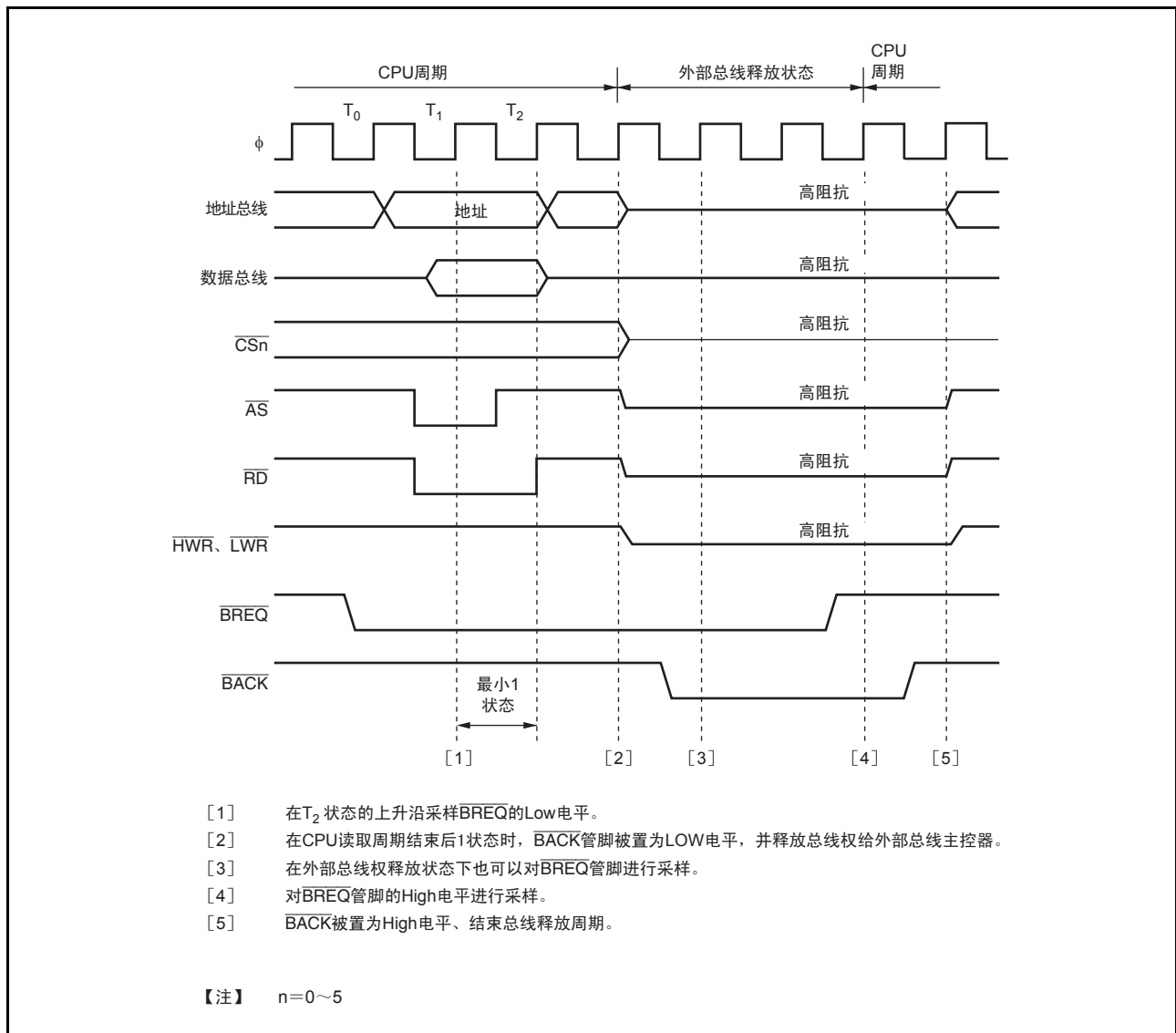


图 6.25 总线权释放状态换移时序

6.9.1 使用总线权释放时的注意事项

将 MSTPCR 设定为 H'FFFFFF，且在转换至睡眠模式的状态下，停止外部总线权释放功能。在睡眠模式下，使用外部总线全释放功能时，请勿将 MSTPCR 设定为 H'FFFFFF。

6.10 总线仲裁

本 LSI 内置仲裁总线主控器运行的总线仲裁器。

总线主控器有 CPU、DMAC 两种，占有总线权时，进行读取 / 写入操作。各总线主控器通过总线权请求信号来请求总线权。总线仲裁器以所定时序判定优先级，通过总线权请求应答信号允许使用总线。总线主控器获得总线权并进行运行。

6.10.1 运行说明

总线仲裁器检测总线主控器的总线权请求信号，如果请求总线权，就会向此总线主控器发出总线权请求应答信号。如果多个总线主控器提出总线权请求，就会向优先级最高的主控器发出总线权请求应答信号。接收到总线权请求应答信号的总线主控器随后，直至撤回此信号为止，一直占有总线。

总线主控器的优先级：

(高) DMAC > CPU (低)

另外，H8S/2218 群，可同时进行内部总线主控器的内部总线存取与外部总线权释放。

外部总线权释放请求与内部总线主控器的外部存取同时发生时的优先级：

(高) 外部总线权释放 > 内部总线主控器的外部存取 (低)

H8S/2212 群无外部总线权释放功能。

6.10.2 总线权转移时序

即使比获得总线权正在运行的总线主控器优先级高的总线主控器有总线权请求时，也不一定立即转移总线权。各总线主控器有可转让总线权的时序。

6.10.2.1 CPU

CPU 为优先级最低的总线主控器，DMAC 提出总线权请求时，总线仲裁器就会将总线权转移至有总线权请求的总线主控器。总线权转移时序如下：

1. 在总线周期的衔接处转移总线权。但是，长字长度的存取等、总线周期进行分割等时，分割的总线周期衔接处，不转移总线权。
2. CPU 为睡眠模式时，立即转移总线权。

6.10.2.2 DMAC

DMAC 提出启动请求时，向总线仲裁器请求总线权。

DMAC 为短地址模式、普通模式的 USB 请求周期 steal 模式时，完成一次传送后释放总线权。

块传送模式时，1 块传送后释放总线权。突发模式时，传送结束后释放总线权。

6.10.3 使用外部总线权释放注意事项

外部总线周期结束时，可进行 H8S/2218 群外部总线权的释放。

$\overline{\text{CS}}$ 信号输出低电平直至外部总线周期结束。

因此，进行外部总线权释放时，有时 $\overline{\text{CS}}$ 信号从低电平高阻抗状态转移。

6.11 复位与总线控制器

加电复位时，本 LSI，包括总线控制器，此时进入复位状态。运行中的总线周期中途可停止。

手动复位 * 时，保持总线控制器寄存器与内部状态。直至运行中的外部总线周期结束。此时，忽略 $\overline{\text{WAIT}}$ 输入。不保证写入数据。

【注】 * 仅限 H8S/2218 群。

7. DMA 控制器 (DMAC)

本 LSI 内置 DMA 控制器 (DMAC)。DMAC 可进行最大 4 个通道的数据传送。

7.1 特点

DMAC 有以下特点。

- 可选择短地址模式和全地址模式

7.1.1 短地址模式

可使用最多 4 个通道

可选择双地址模式运行

在双地址模式，传送源，传送目的的一方指定为 24 位，另一方指定为 16 位

在双地址模式，可选择顺序模式 / 空闲模式 / 重复模式

7.1.2 全地址模式

可使用最多 2 个通道

传送源，传送目的指定为 24 位

- 可选择正常模式/块传送模式可直接指定 16M 字节的地址空间
- 可将字节/字设置为传送单位
- 激活源为内部中断、USB 请求、自动请求 (取决于传送模式)
16 位定时器脉冲单元 (TPU) 的比较匹配/输入捕捉中断 × 3
串行通信接口 (SCI_0) 发送数据完全中断、接受数据完全中断
A/D 转换器的转换结束中断
USB 请求
自动请求
- 可设定模块停止模式

DMAC 的框图如图 7.1 所示。

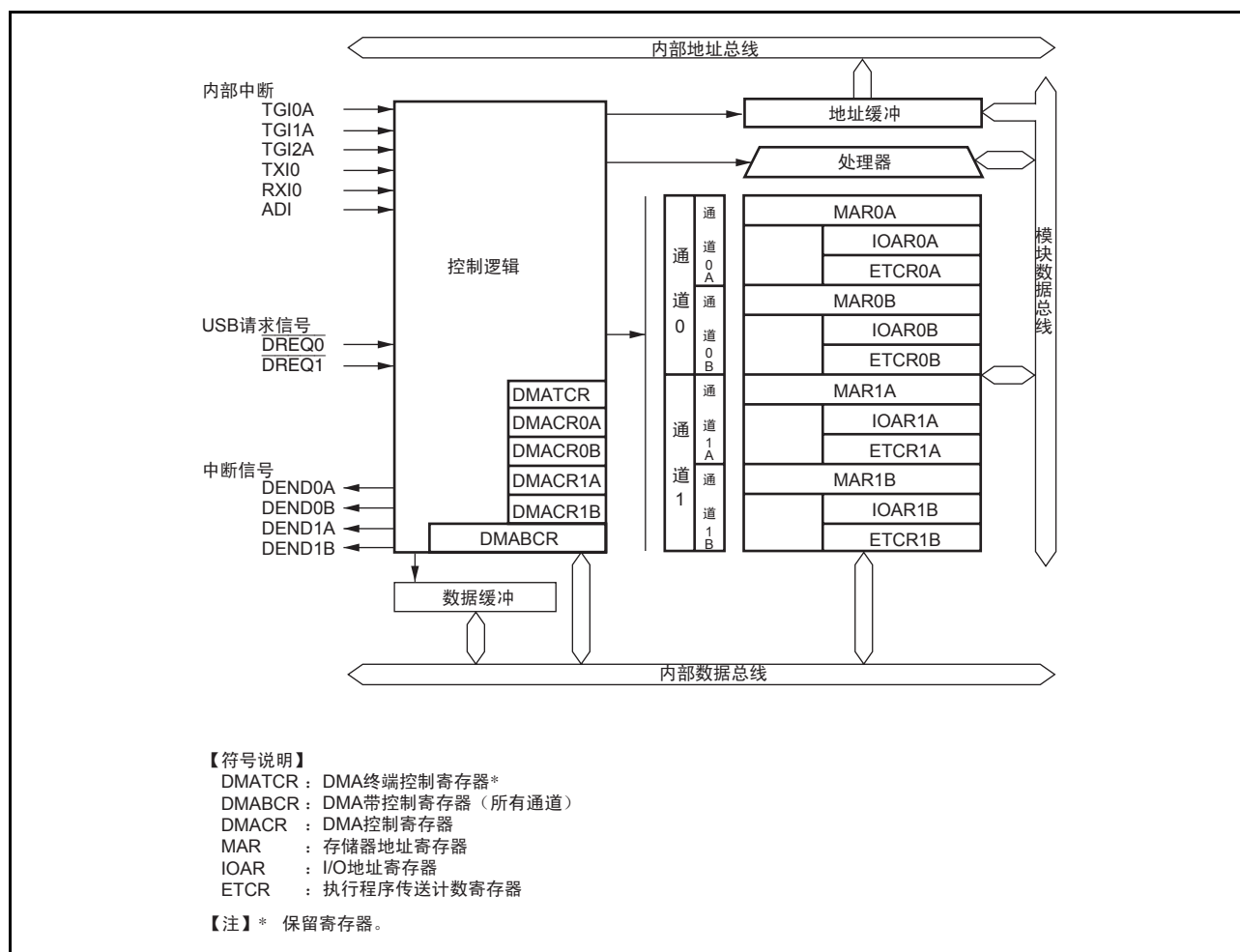


图 7.1 DMAC 的块图

7.2 寄存器说明

DMAC 有以下寄存器。

- 存储器地址寄存器 0A (MAR0A)
- I/O 地址寄存器 0A (IOAR0A)
- 传送计数寄存器 0A (ETCR0A)
- 存储器地址寄存器 0B (MAR0B)
- I/O 地址寄存器 0B (IOAR0B)
- 传送计数寄存器 0B (ETCR0B)
- 存储器地址寄存器 1A (MAR1A)
- I/O 地址寄存器 1A (IOAR1A)
- 传送计数寄存器 1A (ETCR1A)
- 存储器地址寄存器 1B (MAR1B)
- I/O 地址寄存器 1B (IOAR1B)
- 传送计数寄存器 1B (ETCR1B)
- DMA 控制寄存器 0A (DMACR0A)
- DMA 控制寄存器 0B (DMACR0B)
- DMA 控制寄存器 1A (DMACR1A)
- DMA 控制寄存器 1B (DMACR1B)
- DMA 带控制寄存器 (DMABCR)

DMAC 寄存器的功能依赖于寻址模式 (短地址模式、全地址模式) 而有所不同。所以这些功能在每个模式中进行分别的说明。如表 7.1 所示, 可通过 FAE1、FAE0 位将通道 1、0 独立, 选择短地址模式 / 全地址模式。

表 7.1 短地址模式和自由地址模式 (1 个通道: 通道 0 的例子)

FAEO	说 明																								
0	<p>指定短地址模式 (通道A、B独立工作)</p> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px; writing-mode: vertical-rl; text-orientation: upright;">通 道 A</div> <table border="1" style="margin-left: 10px;"> <tr><td colspan="2">MAR0A</td><td>← 指定传送源 / 传送目标地址</td></tr> <tr><td></td><td>IOAR0A</td><td>← 指定传送目标 / 送源地址</td></tr> <tr><td></td><td>ETCR0A</td><td>← 指定传送次数</td></tr> <tr><td></td><td>DMACR0A</td><td>← 指定传送大小、模式、激活源等</td></tr> </table> </div> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px; writing-mode: vertical-rl; text-orientation: upright;">通 道 B</div> <table border="1" style="margin-left: 10px;"> <tr><td colspan="2">MAR0B</td><td>← 指定传送源 / 传送目标地址</td></tr> <tr><td></td><td>IOAR0B</td><td>← 指定传送目标 / 传送源地址</td></tr> <tr><td></td><td>ETCR0B</td><td>← 指定传送次数</td></tr> <tr><td></td><td>DMACR0B</td><td>← 指定传送大小、模式、激活源等</td></tr> </table> </div>	MAR0A		← 指定传送源 / 传送目标地址		IOAR0A	← 指定传送目标 / 送源地址		ETCR0A	← 指定传送次数		DMACR0A	← 指定传送大小、模式、激活源等	MAR0B		← 指定传送源 / 传送目标地址		IOAR0B	← 指定传送目标 / 传送源地址		ETCR0B	← 指定传送次数		DMACR0B	← 指定传送大小、模式、激活源等
MAR0A		← 指定传送源 / 传送目标地址																							
	IOAR0A	← 指定传送目标 / 送源地址																							
	ETCR0A	← 指定传送次数																							
	DMACR0A	← 指定传送大小、模式、激活源等																							
MAR0B		← 指定传送源 / 传送目标地址																							
	IOAR0B	← 指定传送目标 / 传送源地址																							
	ETCR0B	← 指定传送次数																							
	DMACR0B	← 指定传送大小、模式、激活源等																							
1	<p>指定全地址模式 (通道A、B共同工作)</p> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px; writing-mode: vertical-rl; text-orientation: upright;">通 道 0</div> <table border="1" style="margin-left: 10px;"> <tr><td colspan="2">MAR0A</td><td>← 指定传送源地址</td></tr> <tr><td colspan="2">MAR0B</td><td>← 指定传送目标地址</td></tr> <tr><td></td><td>IOAR0A</td><td>← 未使用</td></tr> <tr><td></td><td>IOAR0B</td><td>← 未使用</td></tr> <tr><td></td><td>ETCR0A</td><td>← 指定传送次数</td></tr> <tr><td></td><td>ETCR0B</td><td>← 指定传送次数 (只在块传送时使用)</td></tr> <tr><td></td><td>DMACR0A</td><td>DMACR0B</td><td>← 指定传送大小、模式、激活源等</td></tr> </table> </div>	MAR0A		← 指定传送源地址	MAR0B		← 指定传送目标地址		IOAR0A	← 未使用		IOAR0B	← 未使用		ETCR0A	← 指定传送次数		ETCR0B	← 指定传送次数 (只在块传送时使用)		DMACR0A	DMACR0B	← 指定传送大小、模式、激活源等		
MAR0A		← 指定传送源地址																							
MAR0B		← 指定传送目标地址																							
	IOAR0A	← 未使用																							
	IOAR0B	← 未使用																							
	ETCR0A	← 指定传送次数																							
	ETCR0B	← 指定传送次数 (只在块传送时使用)																							
	DMACR0A	DMACR0B	← 指定传送大小、模式、激活源等																						

7.3 各寄存器的说明

7.3.1 存储器地址寄存器 (MAR)

7.3.1.1 短地址模式

MAR 是 32 位可读 / 写寄存器，指定传送的源地址或目的地址。MAR 的高 8 位是保留位。读后通常读出 0。写定义为无效。通过 DMACR 的 DTDIR 位，MAR 可选择作为源地址寄存器或作为目的地址寄存器。

MAR 在进行每次的 1 字节或 1 字传送时被增量 / 减量，通过 MAR 指定的地址不断的进行自动更新。详细的内容请参考 7.3.4 DMA 控制寄存器 (DMACR)。MDR 在重置和待机模式下不被初始化。

7.3.1.2 全地址模式

MAR 是 32 位可读 / 写的寄存器，MARA 作为传送的源地址寄存器，MARB 作为目的地址寄存器。

MAR 由 2 个 16 位 MARH、MARL 寄存器构成。MARH 的高 8 位是保留位。读时通常读出 0。写被定义为无效。MAR 在每次 1 字节传送或 1 字传送时被增量 / 减量，源或目的存储器地址自动更新。详细请参考 7.3.4 DMA 控制寄存器 (DMACR)。

7.3.2 MI/O 地址寄存器 (IOAR)

7.3.2.1 短地址模式

IOAR 是 16 位可读 / 写的寄存器，指定传送源地址或目的地址的低位为 16 位。传送地址的高 8 位将自动设置为 H'FF。通过 DMACR 的 DTDIR 位选择 IOAR 作为源地址寄存器或目的地址寄存器。

IOAR 在每次传送时不被增量 / 减量，通过 IOAR 指定的地址是固定的。IOAR 在重置和待机模式下不被初始化。

7.3.2.2 全地址模式

IOAR 不可使用在全地址模式传送。

7.3.3 传送计数寄存器 (ETCR)

7.3.3.1 短地址模式

ETCR 是 16 位可读 / 写的寄存器，设定传送数。此寄存器的顺序模式、空闲模式和重复模式的设定有所不同。

ETCR 在复位或待机模式时不被初始化。

顺序模式和空闲模式时，ETCR 作为 16 位传送计数器（计数范围 1 ~ 65536）使用，每进行 1 次传送 ETCR 减 1，当计数达到 H'0000 时，清除 DMABCR 的 DTE 位，结束传送。

在重复模式，ETCR 作为 8 位转送计数器（计数范围 1 ~ 256）ETCRL 和传送数存储寄存器 ETCRH 使用。每进行 1 次传送 ETCRL 减 1，当计数达到 H'00 时，载入 ETCRH 的值。此时，MAR 自动还原计数开始时它的值。DMABCR 的 DTE 位不被清除，此时，传送将反复执行，直到用户将 DTE 位清零为止。

7.3.3.2 全地址模式

ETCR 是 16 位可读 / 写的寄存器，设定传送数，它在正常模式和块传送模式下的功能有所不同。ETCR 在复位或待机模式时不被初始化。

在正常模式，ETCRA 作为 16 位传送计数器使用。每进行 1 次传送 ETCRA 减 1，计数达到 H'0000 时传送结束。

ETCRB 在正常模式不使用。

块传送模式，ETCRAL 作为 8 位的块大小计数器使用，ETCRAH 保留块大小。ETCRAL 在每次进行 1 字节或 1 字的传送时进行减 1，计数达到 H'00 时 ETCRAH 的值被载入。因此，通过 ETCRAH、ETCRAL 设定块大小，可进行任意字节数或字数所构成块的反复传送。

ETCRB 在块传送模式时，作为 16 位的块传送计数器使用。每进行 1 次块传送 ETCRB 减 1，达到 H'0000 时，传送结束。

7.3.4 DMA 控制寄存器 (DMACR)

DMACR 控制 DMAC 各通道的工作。

- 短地址模式 (DMACRA、DMACRB 通用)

位	位名	初始值	R/W	说明
7	DTSZ	0	R/W	数据传送大小 选择 1 次传送的数据大小。 0: 字节大小传送 1: 字大小传送
6	DTID	0	R/W	数据传送增量 / 减量 顺序模式或重复模式时, 每次传送数据时选择 MAR 的增量 / 减量。 在空闲模式, MAR 即不增量也不减量。 0: 数据传送后 MAR 增量 (1) DTSZ = 0 时, 传送后 MAR + 1 (2) DTSZ = 1 时, 传送后 MAR + 2 1: 数据传送后 MAR 减量 (1) DTSZ = 0 时, 传送后 MAR - 1 (2) DTSZ = 1 时, 传送后 MAR - 2
5	RPE	0	R/W	允许重复 与 DMABCR 的 DTIE 位组合, 在顺序模式、空闲模式、重复模式中, 选择以何种模式传送。 RPE DTIE 0 0: 用顺序模式传送 (无传送结束中断) 0 1: 用顺序模式传送 (有传送结束中断) 1 0: 用重复模式传送 (无传送结束中断) 1 1: 用空闲模式传送 (有传送结束中断) 【注】顺序模式、空闲模式、重复运行模式的工作请参考 7.4.2 顺序模式、7.4.3 地址模式、7.4.4 重复模式。
4	DTDIR	0	R/W	数据传输方向 指定数据传送方向 (源、目的)。 0: MAR 作为源地址传送, IOAR 作为目的地址传送 1: IOAR 作为源地址传送, MAR 作为目的地址传送

位	位名	初始值	R/W	说明
3	DTF3	0	R/W	数据传送因子
2	DTF2	0	R/W	选择数据传送因子 (激活源)
1	DTF1	0	R/W	0000: —
0	DTF0	0	R/W	0001: 通过 A/D 转化器的转换结束中断激活
				0010: —
				0011: —
				0100: 通过 SCI 通道 0 的发送数据空中断激活
				0101: 通过 SCI 通道 0 的接收数据全中断激活
				0110: —
				0111: —
				1000: 通过 TPU 通道 0 的比较匹配 / 输入捕捉 A 中断激活
				1001: 通过 TPU 通道 1 的比较匹配 / 输入捕捉 A 中断激活
				1010: 通过 TPU 通道 2 的比较匹配 / 输入捕捉 A 中断激活
				1011: —
				1100: —
				1101: —
				1110: —
				1111: —

多个通道可选择同一个因子。此时，根据通道间的优先顺序，激活从最高优先级的通道开始。通道的优先顺序请参考 7.4.10 DMAC 多个通道的工作。

• 全地址模式 (DMACRA)

位	位名	初始值	R/W	说明
15	DTSZ	0	R/W	数据传送大小 选择 1 次传送的数据大小。 0: 字节大小传送 1: 字大小传送
14 13	SAID SAIDE	0 0	R/W R/W	源地址增量 / 减量 允许源地址增量 / 减量 在传送数据时，指定源地址寄存器 MARA 增量、减量或不变。 00: MARA 固定 01: 数据传送后，MARA 增量 (1) DTSZ = 0 时，传送后 MARA + 1 (2) DTSZ = 1 时，传送后 MARA + 2 10: MARA 固定 11: 数据传送后，MARA 减量 (1) DTSZ = 0 时，传送后 MARA - 1 (2) DTSZ = 1 时，传送后 MARA - 2
12 11	BLKDIR BLKE	0 0	R/W R/W	块方向 允许块 BLKE 指定用正常模式还是块传送模式传送。当指定块传送模式时，BLKDIR 指定源一方还是目的一方为块区域。 00: 用正常模式传送 01: 用块传送模式传送，目的方为块区域 10: 用正常模式传送 11: 用块传送模式传送，源方为块区域 正常模式、块传送模式的工作请参考 7.4 工作说明。
10 ~ 8	—	全部为 0	R/W	保留位 可读 / 写，但只能写入 0。

• 全地址模式 (DMACRB)

位	位名	初始值	R/W	说明
7	—	0	R/W	保留位 可读 / 写，但只能写入 0。
6 5	DAID DAIDE	0 0	R/W R/W	目的地址增量 / 减量 允许目的地址增量 / 减量 在传送数据时，指定目的地址寄存器 MARB 增量、减量或不变。 00: MARB 固定 01: 数据传送后，MARB 增量 (1) DTSZ = 0 时，传送后 MARB + 1 (2) DTSZ = 1 时，传送后 MARB + 2 10: MARB 固定 11: 数据传送后 MARB 减量 (1) DTSZ = 0 时，传送后 MARB - 1 (2) DTSZ = 1 时，传送后 MARB - 2
4	—	0	R/W	保留位 可读 / 写，但只能写入 0。
3 2 1 0	DTF3 DTF2 DTF1 DTF0	0 0 0 0	R/W R/W R/W R/W	数据传送因子 选择数据传送因子 (激活源)。在正常模式和块传送模式所能指定的启动因素不同。 正常模式: 0000: — 0001: — 0010: — 0011: 通过从 USB 发出的 DREQ 信号的低电平输入 (USB 请求) 来激活 010X: — 0110: 自动请求 (周期挪用) 0111: 自动请求 (突发) 1XXX: — 块传送模式 0000: — 0001: 通过 A/D 转化器的转化结束中断来激活 0010: — 0011: — 0100: 通过 SCI 通道 0 的发送数据空中断来激活 0101: 通过 SCI 通道 0 的接收数据自由中断来激活 0110: — 0111: — 1000: 通过 TPU 通道 0 的比较匹配 / 输入捕捉 A 中断来激活 1001: 通过 TPU 通道 1 的比较匹配 / 输入捕捉 A 中断来激活 1010: 通过 TPU 通道 2 的比较匹配 / 输入捕捉 A 中断来激活 1011: — 11XX: — 多个通道可选择同一个因子。此时，按照通道间的优先顺序，激活从最高优先级开始。通道的优先顺序请参考 7.4.10 DMAC 多个通道的工作。 【符号说明】 X: Don't care

7.3.5 DMA 带控制寄存器 (DMABCR)

DMABCR 控制 DMAC 各通道的工作。

- 短地址模式

位	位名	初始值	R/W	说明
15	FAE1	0	R/W	允许全地址 1 指定使用通道 1 的短地址模式 / 全地址模式的位。 在短地址模式, 通道 1A、1B 用作独立的通道。 0: 短地址模式 1: 全地址模式
14	FAE0	0	R/W	允许全地址 0 指定使用通道 0 的短地址模式 / 满地址模式的位。 在短地址模式, 通道 0A、0B 用作独立的通道。 0: 短地址模式 1: 全地址模式
13、12	—	全部为 0	R/W	保留位 写时请写入 0。
11 10 9 8	DTA1B DTA1A DTA0B DTA0A	0 0 0 0	R/W R/W R/W R/W	数据传送应答 当 DMA 传送时, 允许或禁止清除由数据传送因子设置所选择的内部中断源。 在 DTE = 1 时 DTA = 1, 数据传送因子选择的内部中断源, 通过 DMA 传送被自动清除。 在 DTE = 1、DTA = 1 时, 数据传送因子选择的内部中断源不向 CPU 请求中断。 在 DTE = 1、DTA = 0 时, 数据传送因子选择的内部中断源, 在传送时不被清除, 同时可向 CPU 并列发布中断请求。此时, 通过 CPU 可清除中断源。 在 DTE = 0 时, 数据传送因子选择的内部中断不管 DTA 位的设置向 CPU 发布中断请求。 0: 禁止在 DMA 传送时清除选择的内部中断源。 1: 允许在 DMA 传送时清除选择的内部中断源。

位	位名	初始值	R/W	说明
7 6 5 4	DTE1B DTE1A DTE0B DTE0A	0 0 0 0	R/W R/W R/W R/W	<p>允许数据传输</p> <p>允许或禁止数据传送的位。</p> <p>DTE 位 = 0 时，数据传送是禁止状态，由数据传送因子设置而选择的激活源是被忽略的。如果激活源是内部中断，就会向 CPU 发布中断请求。</p> <p>当 DTE = 0 时，将 DTIE 位置 1，DMAC 就视此为传送结束的指示，并向 CPU 发布传送结束中断请求。</p> <p>DTE 位被清 0 的条件如下：</p> <ul style="list-style-type: none"> • 初始化时 • 在传送模式下（除重复模式）指定的传送数完成时 • 当向 DTE 位写 0，或因类似强行中止传送等原因时 <p>当 DTE = 1 时，数据传送被允许，由数据传送因子选择的激活源发布的请求为等待状态。当激活源发布了请求时，就执行 DMA 传送。</p> <p>DTE 位被置 1 的条件如下：</p> <ul style="list-style-type: none"> • 读 DTE = 0 后写入 DTE = 1 时 <p>0: 禁止数据传送 1: 允许数据传送</p>
3 2 1 0	DTIE1B DTIE1A DTIE0B DTIE0A	0 0 0 0	R/W R/W R/W R/W	<p>允许数据传送结束中断</p> <p>是传送结束时允许或禁止向 CPU 请求的中断的位。如果在 DTE = 0 时 DTIE 的位置 1，DMAC 就认为是传送结束的指示，就向 CPU 发布传送结束中断请求。</p> <p>解除传送结束中断，可用中断处理程序将 DTIE 位清 0 的方法或通过再次设定传送计数器和地址寄存器后将 DTE 位置 1，使传送得以继续。</p> <p>0: 禁止传送结束中断 1: 允许传送结束中断</p>

- 全地址模式

位	位名	初始值	R/W	说明
15	FAE1	0	R/W	允许全地址 1 指定通道 1 使用短地址模式或全地址模式。 在全地址模式，通道 1A、1B 在一起可作为通道 1 使用。 0: 短地址模式 1: 全地址模式
14	FAE0	0	R/W	允许全地址 0 指定通道 0 使用短地址模式或自由地址模式。 在全地址模式，通道 0A、0B 在一起可作为通道 0 使用。 0: 短地址模式 1: 全地址模式
13、12	—	全部为 0	R/W	保留位 可读 / 写，但只能写入 0。
				数据传送应答 当 DMA 传送时，是允许或禁止清除由数据传送因子设置所选择的内部中断源的位。在 DTE = 1 时 DTA = 1，数据传送因子选择的内部中断源，通过 DMA 传送被自动清除。 在 DTE = 1、DTA = 1 时，数据传送因子选择的内部中断源不向 CPU 请求中断。 在 DTE = 1 时 DTA = 0，数据传送因子选择的内部中断源，在传送时不被清除，同时可向 CPU 并列发布中断请求。此时，通过 CPU 可清除中断源。 在 DTE = 0 时，数据传送因子选择的内部中断源不管 DTA 位的设置向 CPU 发布中断请求。
11	—	0	R/W	DTME 位的状态不受前述运行影响。 数据传输应答 1 当 DMA 传送时，是允许或禁止清除由通道 1 数据传送因子设置所选择的内部中断源的位。 0: 禁止在 DMA 传送时清除选择的内部中断源。 1: 允许在 DMA 传送时清除选择的内部中断源。
10	—	0	R/W	保留位 可读 / 写，但只能写入 0。
9	DTA0	0	R/W	数据传输应答 0 当 DMA 传送时，是允许或禁止清除由通道 0 数据传送因子设置所选择的内部中断源的位。 0: 禁止在 DMA 传送时清除选择的内部中断源 1: 允许在 DMA 传送时清除选择的内部中断源
8	—	0	R/W	保留位 可读 / 写，但只能写入 0。

位	位名	初始值	R/W	说明
				<p>允许数据传送控制</p> <p>与 DTE 位一起，控制允许或禁止相关通道的数据传送。</p> <p>当 DTME 位和 DTE 位的都置 1 后，其通道成为传送允许状态。</p> <p>当相关通道处于突发模式传送中间，一旦产生 NMI 中断，DTME 位就被清除，传送被中断，总线权就移交给 CPU。随后，当 DTME 位再次置 1，中断了的传送又重新开始。但是，在块传送模式时，NMI 中断不会清除 DTME 位，传送也就不会被中断。</p> <p>DTME 位被清 0 的条件如下：</p> <ul style="list-style-type: none"> • 初始化时 • 在突发模式输入 NMI 时 • 向 DTME 位写入 0 时 <p>DTME 被置 1 的条件如下：</p> <ul style="list-style-type: none"> • 读 DTME 位 = 0 后向 DTME 位写入 1 时
7	DTME1	0	R/W	<p>允许数据传送控制 1</p> <p>是允许或禁止通道 1 的数据传送的位。</p> <p>0: 禁止数据传送。突发模式时，由 NMI 产生中断后清 0</p> <p>1: 允许数据传送</p>
				<p>允许数据传送</p> <p>DTE 位 = 0 时，数据传送是禁止状态，由数据传送因子设置所选择的激活源被忽略。如果激活源是内部中断，就向 CPU 发布中断请求。</p> <p>当 DTE = 0 时如果 DTIE 的位置 1，DMAC 就认为是传送结束的指示，就向 CPU 发布传送结束中断请求。</p> <p>DTE 的位清 0 的条件如下：</p> <ul style="list-style-type: none"> • 初始化时 • 指定的传送数完成时 • 当向 DTE 位写 0，或因类似的原因强行中止传送时 <p>DTE = 1 且 DTME = 1 时数据传送成为许可状态，由数据传送因子设置所选择的激活源发布的请求成为等待状态。当激活源发布请求后，就执行 DMA 传送。</p> <p>DTE 被置 1 的条件如下：</p> <ul style="list-style-type: none"> • 读 DTE = 0 后向 DTE 位写入 1 时
6	DTE1	0	R/W	<p>允许数据传送 1</p> <p>允许或禁止通道 1 的数据传送</p> <p>0: 禁止数据传送</p> <p>1: 允许数据传送</p>
5	DTME0	0	R/W	<p>允许数据传送控制 0</p> <p>允许或禁止通道 0 的数据传送</p> <p>0: 禁止数据传送。突发模式时，由 NMI 中断清 0</p> <p>1: 允许数据传送</p>

位	位名	初始值	R/W	说明
4	DTE0	0	R/W	<p>允许数据传输 0 允许或禁止通道 0 的数据传送 0: 禁止数据传送 1: 允许数据传送</p> <p>允许数据传送中断 B 在传送中断时, 是允许或禁止向 CPU 请求中断的位。如果在 DTME = 0 时 DTIEB 的位置 1, DMAC 视此为传送暂停的指示, 就向 CPU 发布传送暂停中断的请求。 解除传送暂停中断, 可将中断处理程序的 DTIEB 位清 0, 或将 DTME 位置 1, 使传送继续。</p>
3	DTIE1B	0	R/W	<p>允许数据传送中断 1B 是允许或禁止通道 1 传送暂停中断的位 0: 禁止传送暂停中断 1: 允许传送暂停中断</p> <p>允许数据传送结束中断 A 在传送结束时, 是允许或禁止向 CPU 请求中断的位。如果在 DTE = 0 时 DTIEA 的位置 1, DMAC 视此为传送结束的指示, 就向 CPU 发布传送结束中断的请求。 解除传送结束中断, 可将中断处理程序的 DTIEA 位清 0, 或将 DTE 位置 1, 使传送继续。</p>
2	DTIE1A	0	R/W	<p>允许数据传送结束中断 1A 是允许或禁止通道 1 的传送结束中断的位 0: 禁止传送结束中断 1: 允许传送结束中断</p>
1	DTIE0B	0	R/W	<p>允许数据传送中断 0B 是允许或禁止通道 0 的传送暂停中断的位 0: 禁止传送暂停中断 1: 允许传送暂停中断</p>
0	DTIE0A	0	R/W	<p>允许数据传送结束中断 0A 是允许或禁止通道 0 的传送结束中断的位。 0: 禁止传送结束中断 1: 允许传送结束中断</p>

7.4 工作说明

7.4.1 传送模式

DMAC 模式一览如表 7.2 所示。

表 7.2 DMAC 传送模式

传送模式			传送源	备注
短地址模式	双地址模式	1. 顺序模式 2. 地址模式 3. 重复模式	TPU 通道 0 ~ 2 的比较匹配 / 输入捕捉 A 中断 • SCI 发送数据的完成中断 • SCI 接收数据的完成中断 • A/D 转换器的转换结束中断	• 可进行最大 4 通道的独立工作
自由地址	4. 正常模式		• USB 请求 • 自动请求	• 将通道 A、B 组合, 可运行最大 2 个通道 • 通过自动请求可选择突发模式传送 / 周期挪用传送
	5. 块传送模式		• TPU 通道 0 ~ 2 的比较匹配 / 输入捕捉 A 中断 • SCI 发送的完成中断 • SCI 接收的完成中断 • A/D 转换器的结束中断	

7.4.2 顺序模式

可通过将 DMACR 的 RPE 位清 0 来指定顺序模式。在顺序模式, 每次为响应单一传送请求而传送 1 字节或 1 字后, MAR 都被更新, 这一过程按照 ETCR 指定的次数执行。地址的一方用 MAR 指定、另一方用 IOAR 指定。传送方向可通过 DMACR 的 DTDIR 位指定。顺序模式时, 寄存器的功能如表 7.3 所示。

表 7.3 顺序模式时的寄存器功能

对象寄存器	功能		初始设定值	工作
	DTDIR=0	DTDIR=1		
	源地址寄存器	目的地址寄存器	传送地址或传送源的起始地址	每次传送进行增量 / 减量
	目的地址寄存器	源地址寄存器	传送源或传送地址的起始地址	固定
	传送计数器		传送次数	每次传送进行减量 1。成为 H'0000 后, 结束传送

MAR 指定传送源或传送目的的起始地址为 24 位。每次在 1 字节或字传送时，MAR 增量 / 减量 1 或 2。IOAR 指定另一方地址的低 16 位。IOAR 高 8 位的值为 H'FF。顺序模式的工作如图 7.2 所示。

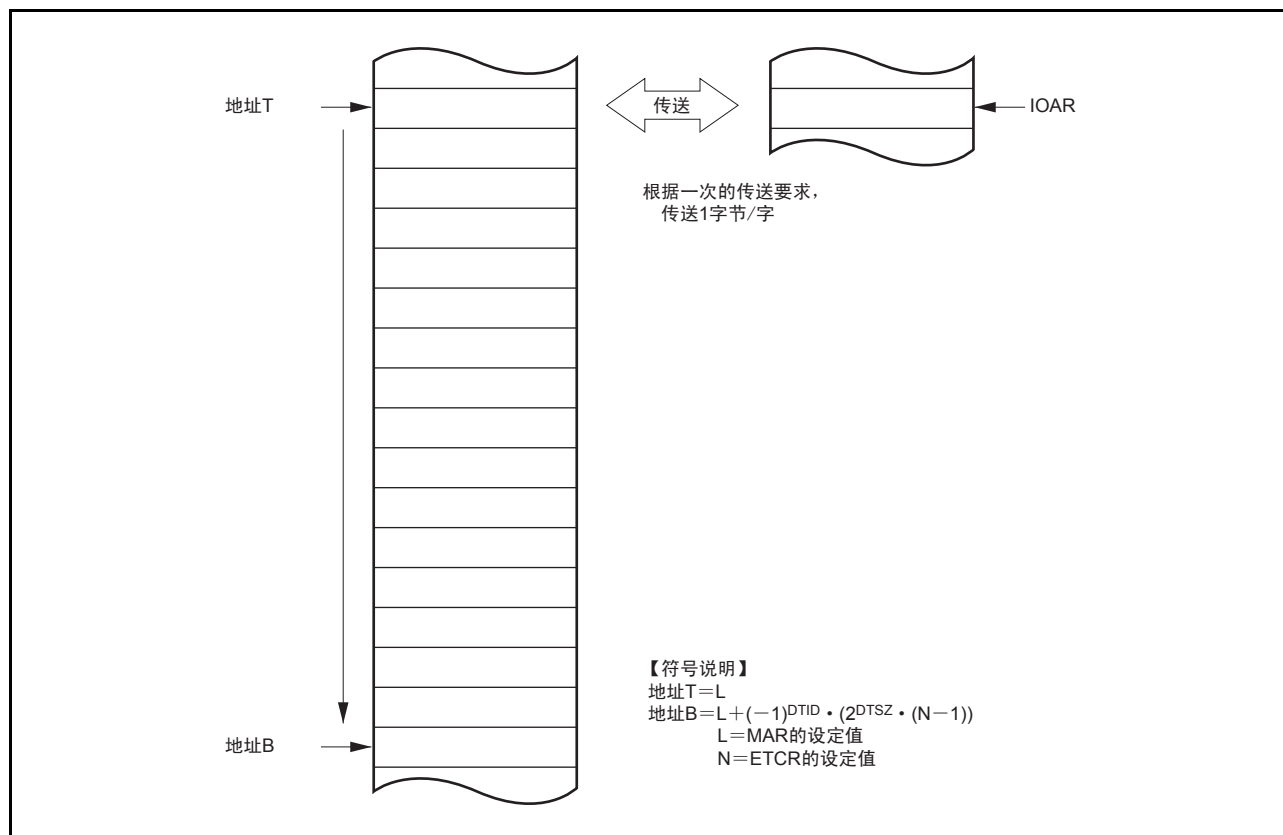


图 7.2 顺序模式的工作

传送数在 ETCR 被指定为 16 位。每次执行传送时 ETCR 减 1，当其值达到 H'0000 时，清除 DTE 位并结束传送。此时，如果 DTIE 位被置 1，就会向 CPU 发送一个中断请求。当 ETCR 设置了 H'0000 时，最大传送数为 65536。传送请求（激活源）包括 A/D 转换器的转换结束中断，SCI 的发送数据完成 / 接收数据完成中断，及 TPU 通道 0 ~ 2 的比较匹配 / 输入捕捉 A 中断。顺序模式的设定过程例子如图 7.3 所示。



图 7.3 顺序模式的设定顺序例子

7.4.3 地址模式

通过将 DMACR 的 RPE 位和 DTIE 位置 1 可指定空闲模式。空闲模式时，为响应单一的传送请求而传送一个 1 字节或 1 字，这一过程按照在 ETCR 中指定的次数执行。地址一方用 MAR 指定，另一方用 IOAR 指定。传送方向可通过 DMACR 的 DTDIR 位指定。空闲模式时的功能如表 7.4 表示。

表 7.4 地址模式时的寄存器功能

对象寄存器	功能		初始设定值	工作
	DTDIR=0	DTDIR=1		
	源地址寄存器	目的地址寄存器	传送地址或传送元的起始地址	固定
	目的地址寄存器	源地址寄存器	传送元或传送地址的起始地址	固定
	传送计数器		传送次数	每次传送进行减量，成为 H'0000 后结束传送

MAR 指定传送源或传送目的的起始地址为 24 位。每次传送字节或字时 MAR 不被增量、减量。IOAR 指定另一方地址为低 16 位。IOAR 高 8 位的值为 H'FF。空闲模式的工作如图 7.4 所示。

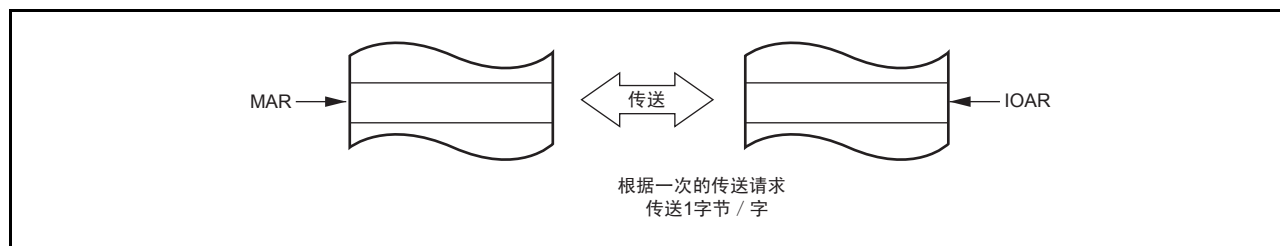


图 7.4 地址模式的工作

传送数在 ETCR 被指定为 16 位。每次执行传送时 ETCR 减 1，达到 H'0000 时将 DTE 位清除并结束传送。此时，如果 DTIE 位被置 1，就会向 CPU 发送一个中断请求。当 ETCR 设置了 H'0000 时，最大传送数为 65536。

传送请求（激活源）包括 A/D 转换器的转换结束中断，SCI 的发送数据完成 / 接收数据完成中断，及 TPU 通道 0 ~ 2 的比较匹配 / 输入捕捉 A 中断。空闲模式的设定过程例子如图 7.5 所示。

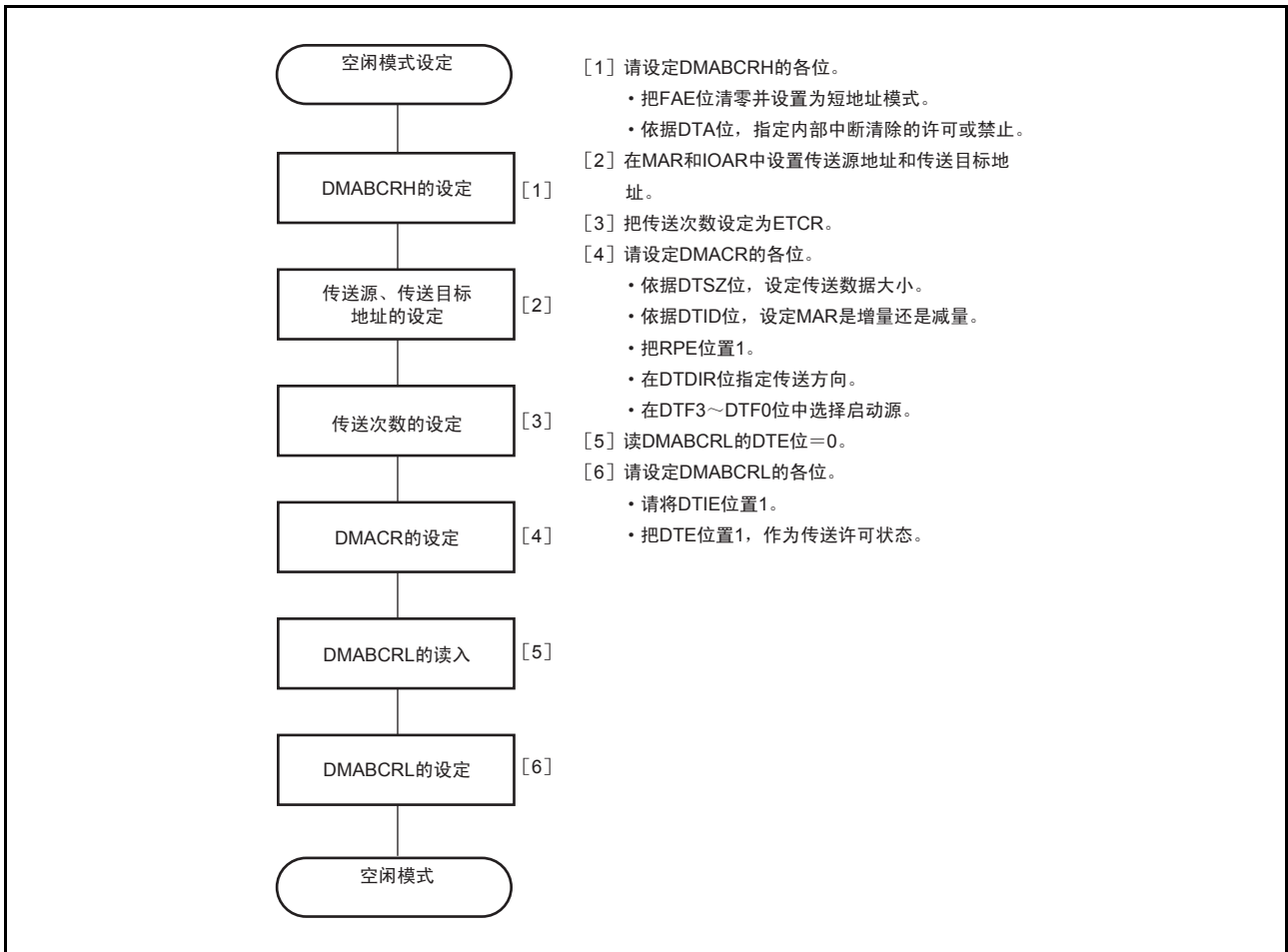

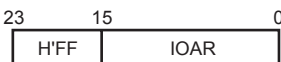
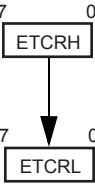


图 7.5 地址模式的设定顺序例子

7.4.4 重复模式

通过 DMACR 的 RPE 位置 1，DTIE 位清 0 来指定重复模式。在重复模式，每次为响应单一传送请求而传送 1 字节或 1 字后，MAR 都被更新，这一过程按照 ETCR 指定的次数执行。指定传送数完成时，MAR、ETCRL 自动还原最初的设置，继续运算。地址的一方用 MAR 指定，另一方用 IOAR 来指定。传送方向可通过 DMACR 的 DTDIR 位指定。重复模式时寄存器的功能如表 7.5 所示。

表 7.5 重复模式时的寄存器功能

对象寄存器	功能		初始设定值	工作
	DTDIR=0	DTDIR=1		
	源地址寄存器	目的地址寄存器	传送地址或传送源的起始地址	每次传送进行增量 / 减量成为 H'0000 后返回初始设定值
	目的地址寄存器	源地址寄存器	传送源或传送地址的起始地址	固定
	保持传送次数 传送计数器		传送次数 传送次数	固定 每次传送进行减量。成为 H'00 后，装入 ETCRH 的值

MAR 指定传送源或传送目的的起始地址为 24 位。每次传送字节或字时，MAR 增量 / 减量 1 或 2。IOAR 指定另一方地址为低 16 位。IOAR 高 8 位的值为 H'FF。传送数通过 ETCRH、ETCRL 指定为 8 位。当 ETCRH、ETCRL 都设定了 H'00 时，最大传送数为 256。

在重复模式，ETCRL 作为传送计数器，ETCRH 用于保留传送数。每执行一次传送，ETCRL 减 1，当其值达到 H'00 时载入 ETCRH 的值。此时，MAR 依照 DMACR 的 DTSZ、DTID 位的值还原设定值。MAR 的还原运算如下所示。

$$\text{MAR} = \text{MAR} - (-1)^{\text{DTID}} \cdot 2^{\text{DTSZ}} \cdot \text{ETCRH}$$

请将 ETCRH 和 ETCRL 设定为同一值。

在重复模式，运算继续直到 DTE 位被清除为止。因此，为了结束传送请将 DTE 位清 0。此时不向 CPU 发送传送结束中断请求。清除 DTE 位后，将 DTE 位再次置位，可使运算从 DTE 位被清 0 时终止的传送处重新开始。重复模式的工作如图 7.6 所示。

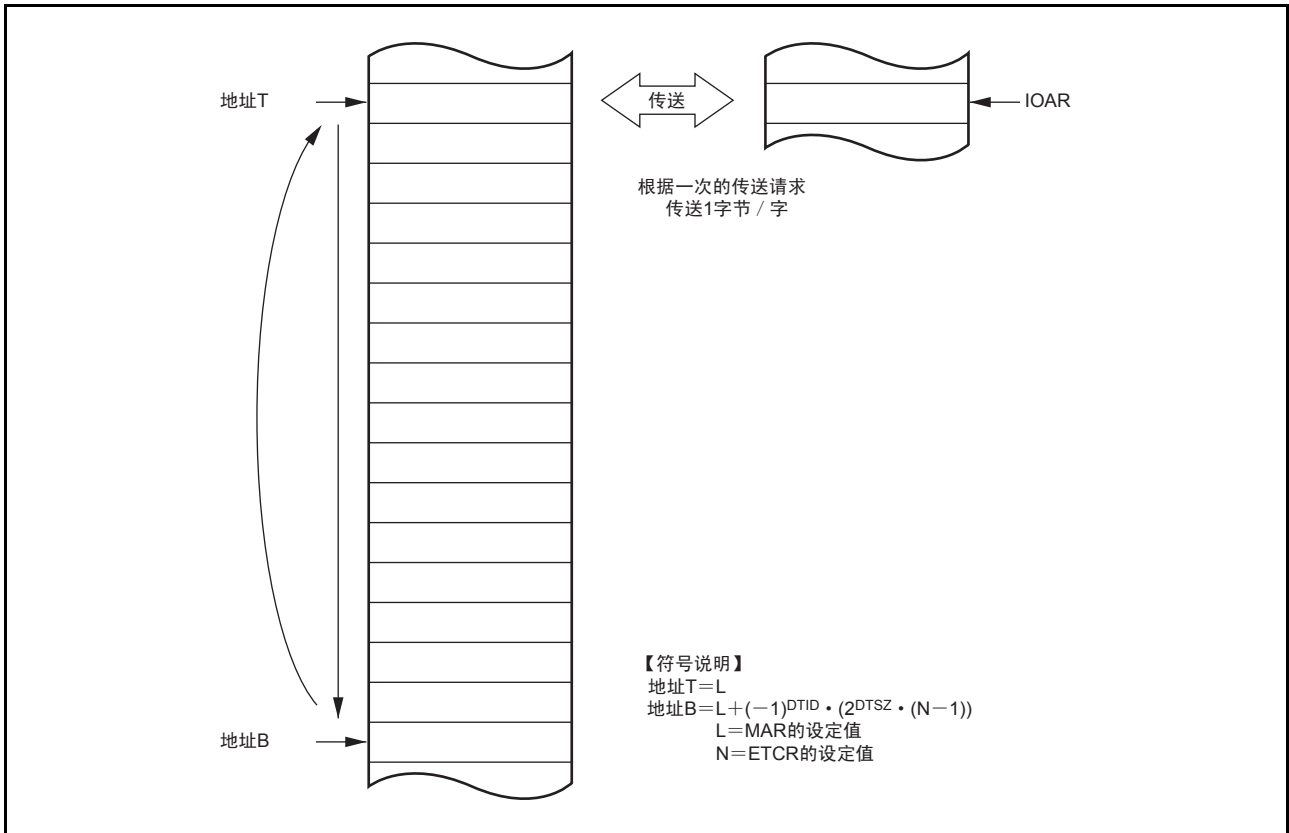


图 7.6 重复模式的工作图

传送请求（激活源）包括 A/D 转换器的转换结束中断，SCI 的发送数据完成 / 接收数据完成中断，及 TPU 通道 0 ~ 2 的比较匹配 / 输入捕捉 A 中断。重复模式的设定过程例子如图 7.7 所示。

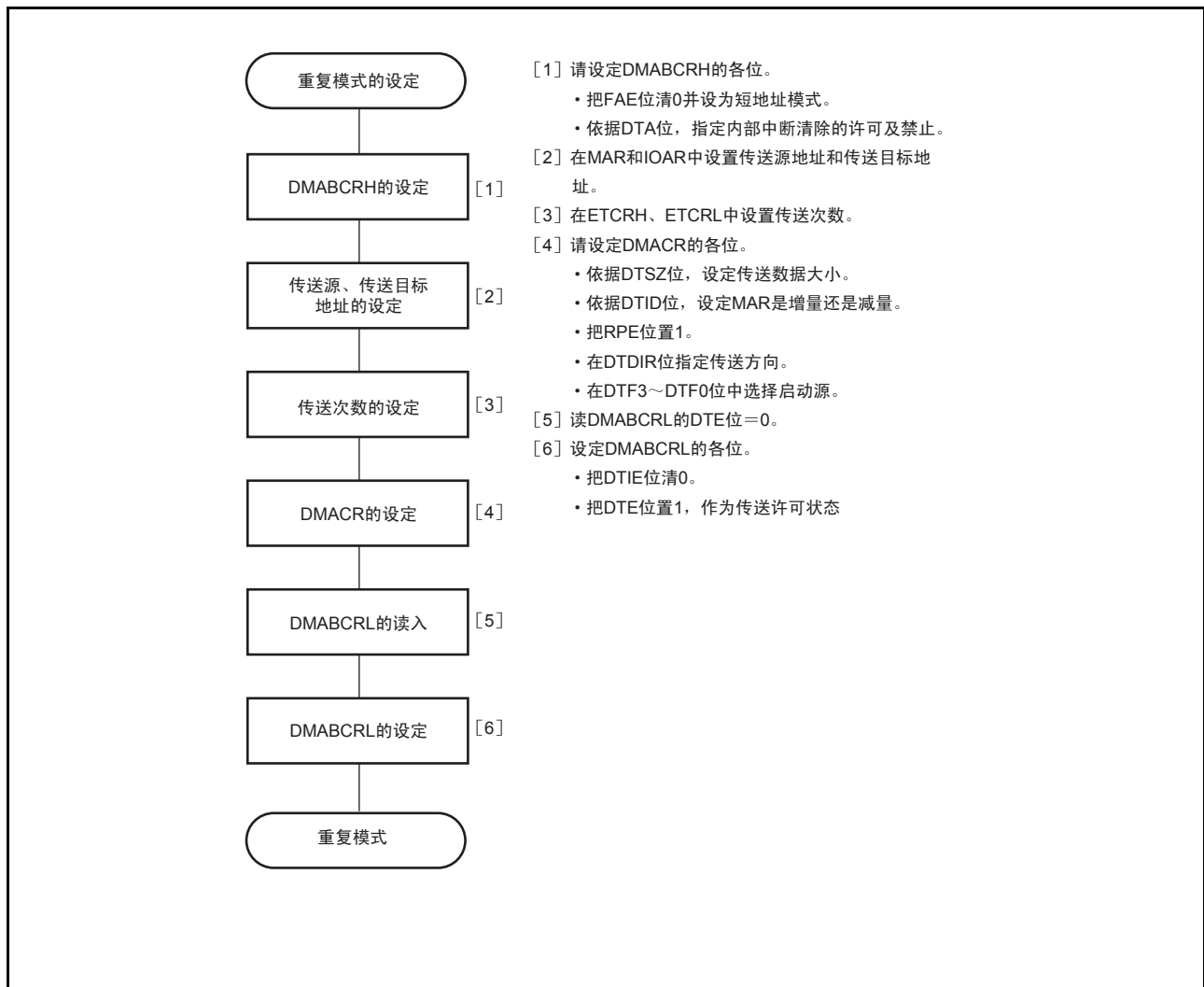





图 7.7 重复模式的设定顺序例子

7.4.5 正常模式

在正常模式，用 A、B 通道的组合进行传送。通过将 DMABCR 的 FAE 位置 1、DMACRA 的 BLKE 位清 0，可指定正常模式。在正常模式，每次为响应单一传送请求而传送字节或字后，MAR 都被更新，这一过程按照 ETCRA 指定的次数执行。用 MARA 指定传送源，用 MARB 指定传送目的。正常模式时寄存器的功能如表 7.6 所示。

表 7.6 正常模式时的寄存器功能

对象寄存器	功能	初始设定值	工作
	源地址寄存器	传送源起始地址	每次传送时进行增量 / 减、或固定
	目的地址寄存器	传送地址的起始地址	每次传送时进行增量 / 减量、或固定
	传送计数器	传送次数	每次传送时进行减量，成为 H'0000 后传送结束

MARA、MARB 分别指定传送源或传送目的的起始地址为 24 位。每次字节或字传送时，MAR 可增量 / 减量 1、2，或也可固定。可分别为 MARA、MARB 设定增量 / 减量 / 保持固定值。

传送数通过 ETCRA 指定为 16 位。每执行一次传送进行减 1，达到 H'0000 时清除 DTE 位并结束传送。此时，如果将 DTIE 位置 1，就向 CPU 发送一个中断请求。当 ETCRA 设定了 H'0000 时，最大传送数为 65536。

正常模式工作如图 7.8 所示。

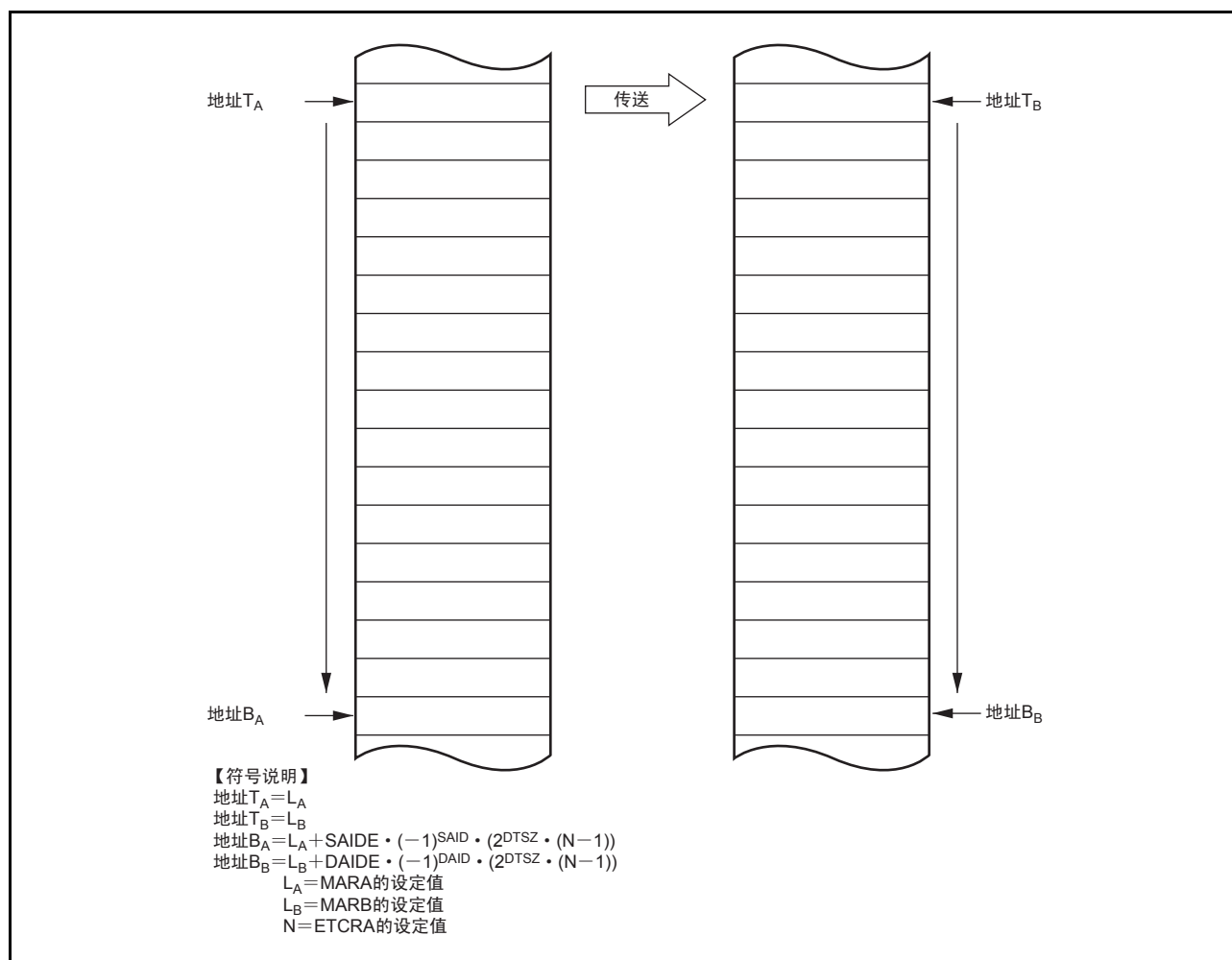


图 7.8 正常模式的工作

传送请求（激活源）包括 USB 请求和自动请求。只有通过寄存器的设定 DMAC 才能用自动请求激活，同时自动执行指定的传送数。可用自动请求选择周期挪用模式和突发模式。在周期挪用模式，每次传送时总线被释放到另一个总线主控器。在突发模式，总线持续占据到传送结束为止。

详细的设定请参考 7.3.4 DMA 控制寄存器 (DMACR)。

正常模式的设定过程例子如图 7.9 所示。



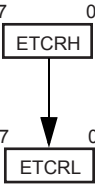



图 7.9 正常模式的设定顺序例子

7.4.6 块传送模式

在块传送模式，通过通道 A、B 的组合进行传送。通过将 DMABCR 的 FAE 位置 1，DMACRA 的 BLKE 位置 1，可指定块传送模式。在块传送模式，为响应单一传送请求而执行一个指定了块大小的传送，此过程按照指定的次数执行。传送源用 MARA 指定，传送目的用 MARB 指定。传送源或传送目的都可选择一个块区域（由许多字节 / 字构成的区域）。块传送模式时寄存器的功能如表 7.7 表示。

表 7.7 块传送模式时的寄存器功能

对象寄存器	功能	初始设定值	工作
	源地址寄存器	传送源起始地址	每次传送时进行增量 / 减量，或固定
	目的地址寄存器	传送地址的起始地址	每次传送时进行增量 / 减量，或固定
	保持块大小 块大小计数器	块大小 块大小	固定 每次传送时进行减量，成为 H'00 后复制 ETCRH 的值
	块传送计数器	块传送次数	每传送 1 块时进行减量，成为 H'0000 后传送结束

MARA、MARB 分别指定传送源及传送目的的起始地址为 24 位。每传送一次字节或字时，MAR 可增量 / 减量 1、2，或固定。可分别为 MARA、MARB 设定增量 / 减量 / 固定值。通过 DMACRA 的 BLKDIR 位指定将 MARA 和 MARB 哪一个设定为块。

传送数的指定：假如 1 块的大小为 M（M = 1 ~ 256），执行 N（N = 1 ~ 65536）次，则 M 在 ETCRAH、ETCRAL 中设定，N 在 ETCRB 中设定。

MARB 作为块区域时，块传送模式的工作如图 7.10 表示。

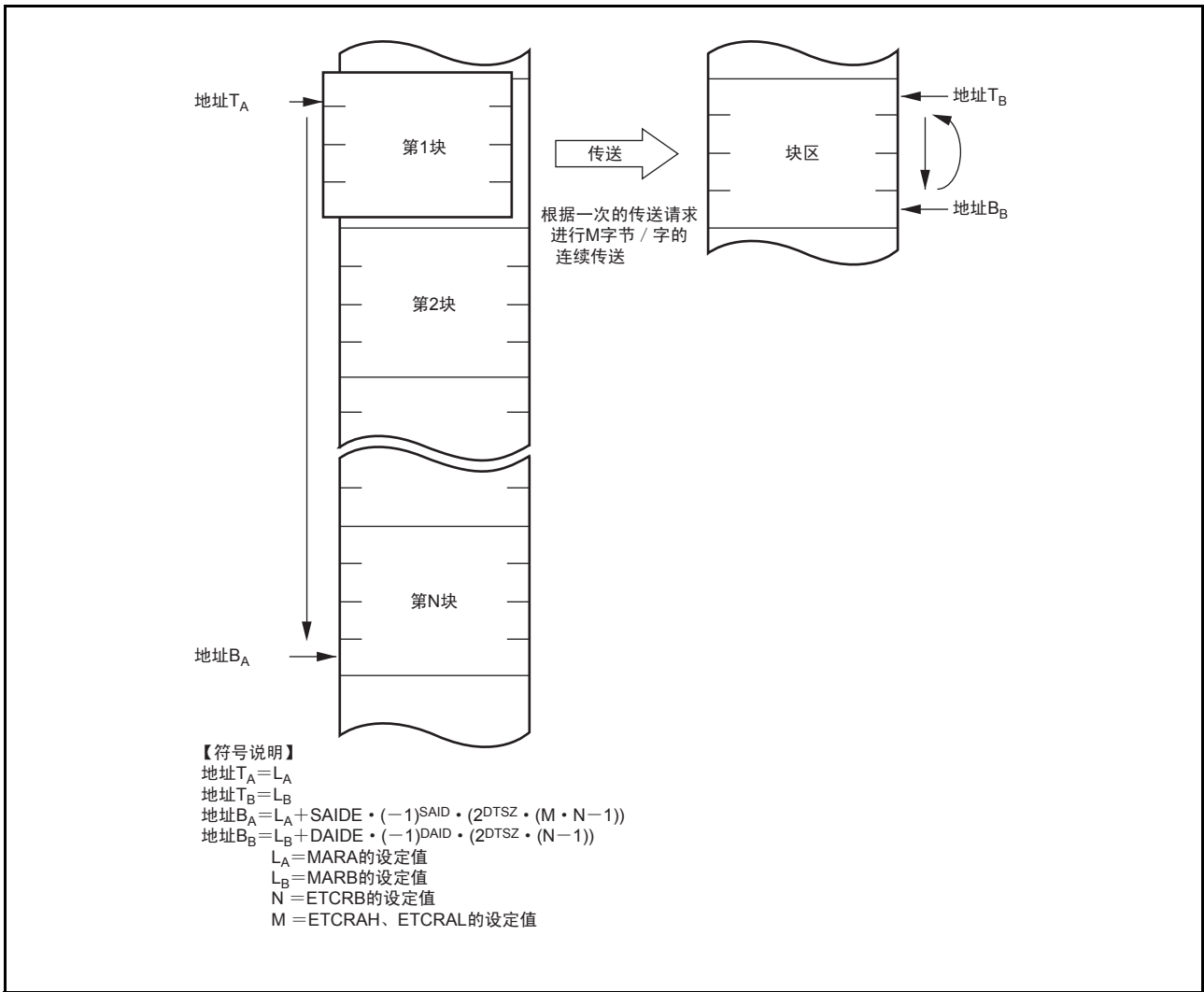


图 7.10 块传送模式的工作 (BLKDIR = 0)

MARA 作为块区域时，块传送模式的工作如图 7.11 所示。

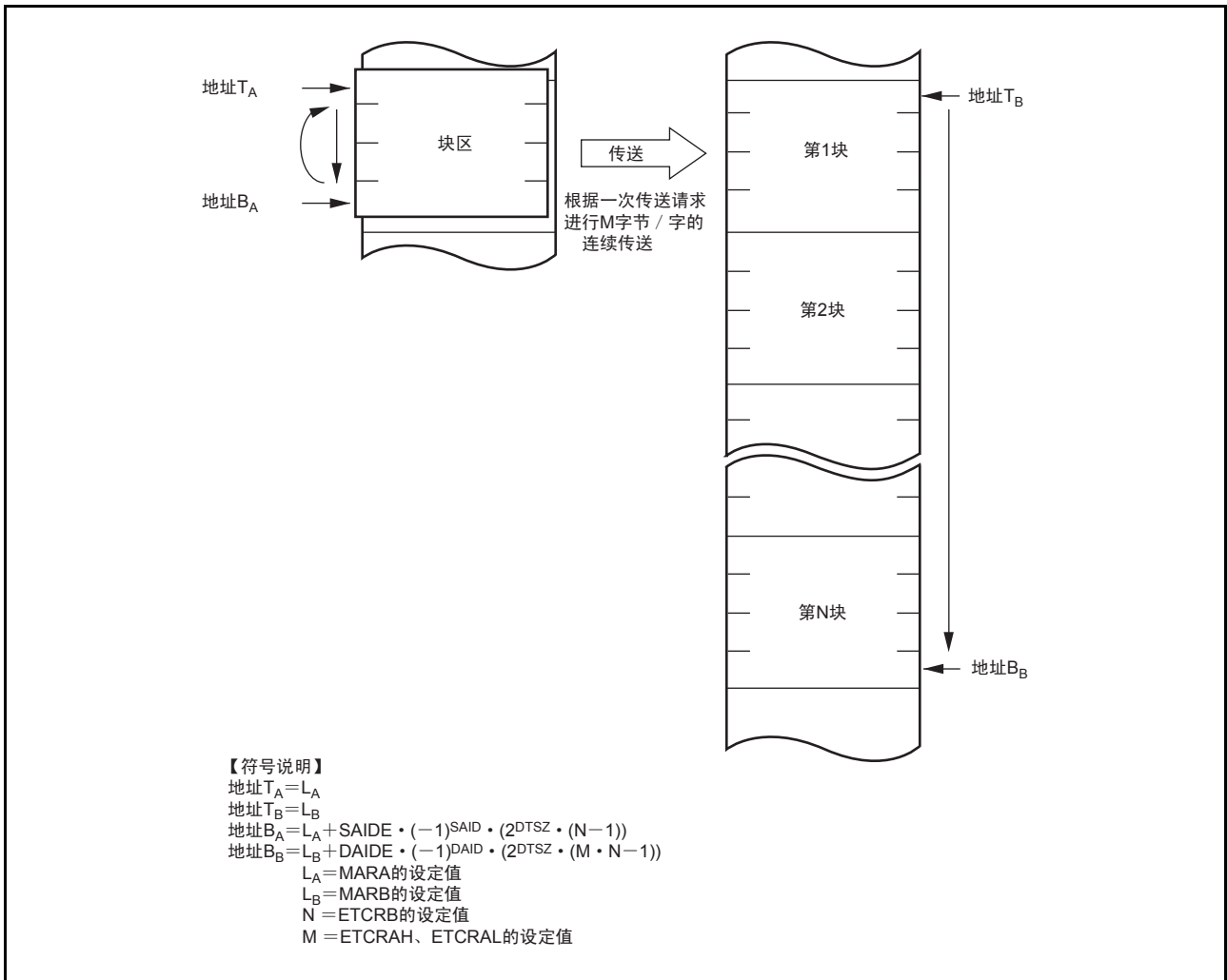


图 7.11 块传送模式的工作 (BLKDIR = 1)

在每传送一次字节或字，ETCRAL 减 1。以响应单一传送请求的方式执行突发传送直到 ETCRAL 的值达到 H'00 为止。然后 ETCRAL 载入 ETCRAH 的值。此时，被 DMACRA 的 BLKDIR 位指定为块的 MAR 中的值依照 DMACR 的 DTSZ 及 SAID/DAID、SAIDE/DAIDE 位被还原。

每次块传送，ETCRB 减 1，计数达到 H'0000 时将 DTE 位清除并结束传送。此时，如果将 DTIE 位置 1，就会向 CPU 发送中断请求。图 7.12 表示块传送模式的工作流程。

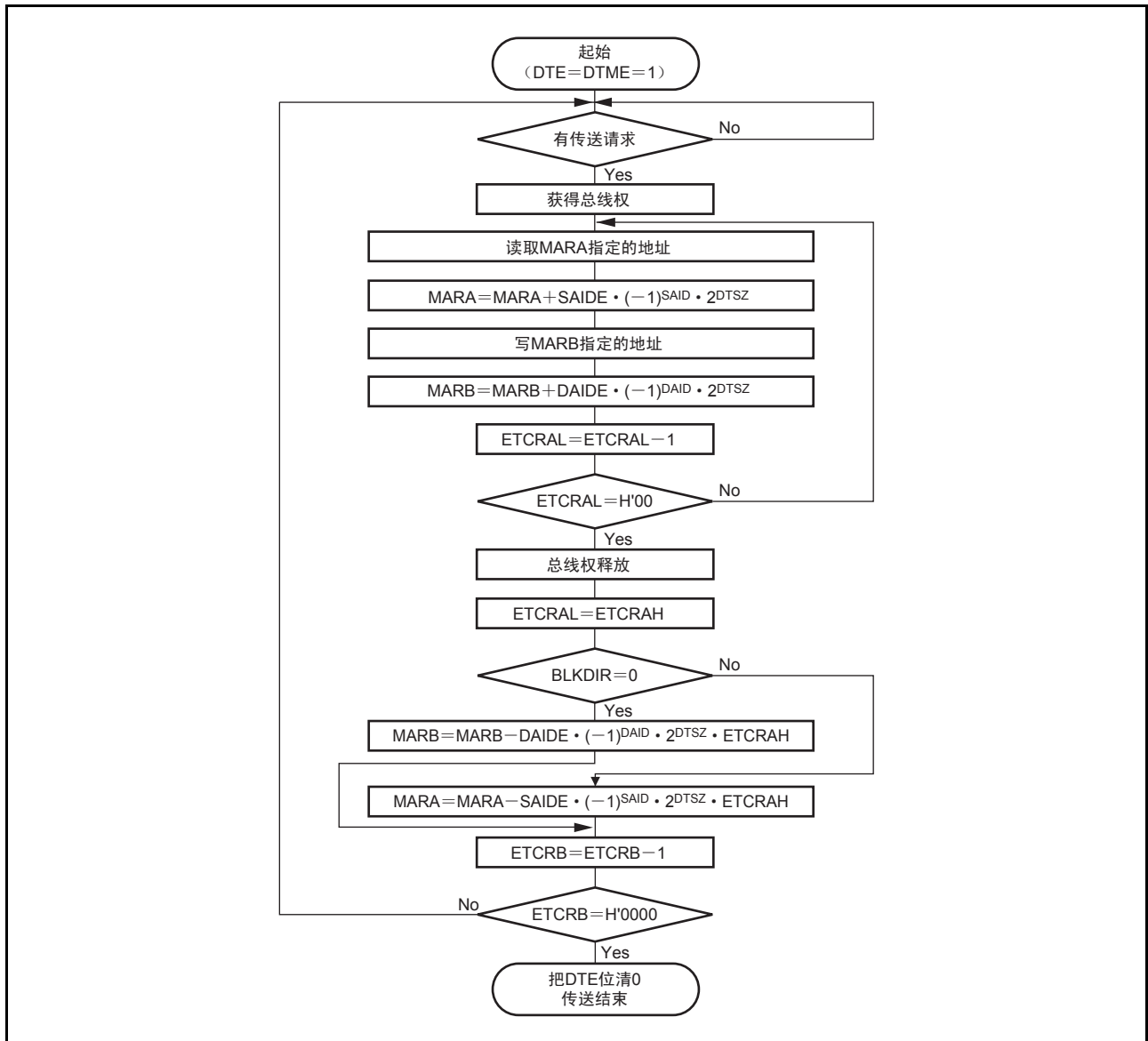


图 7.12 块传送模式的工作流程

传送请求 (启动源) 包括 A/D 转换器的转换结束中断, SCI 的发送数据空或接收数据满中断, 及 TPU 通道 0 ~ 2 的比较匹配 / 输入捕捉 A 中断。

详细设定请参照 7.3.4 DMA 控制寄存器 (DMACR)。块传送模式的设定过程例子如图 7.13 所示。



图 7.13 块传送模式的设定顺序例子

7.4.7 DMAC 启动源

DMAC 激活源有内部中断 USB 请求及自动请求。依据传送模式可指定的激活源如表 7.8 所示。

表 7.8 DMAC 启动源

启动源		短地址模式	自由地址模式	
			普通模式	块传送模式
内部中断	ADI	○	×	○
	TXI0	○	×	○
	RXI0	○	×	○
	TGI0A	○	×	○
	TGI1A	○	×	○
	TGI2A	○	×	○
USB 请求	DERQ 信号的 Low 电平输入	×	○	×
自动请求		×	○	×

【符号说明】

○：可指定

×：不可指定

7.4.7.1 通过内部中断激活

一个被选作 DMAC 的激活源的中断请求，也可同时向 CPU 发送。详细请参考 5. 中断控制器。

随着内部中断的激活，DMAC 接收中断控制器独立的请求。因此，中断控制器的优先级的设定是不受影响的。

如果 DMAC 被一个不用作 CPU 中断源 (DTA = 1) 的中断请求激活时，中断请求标志通过 DMA 传送自动被清除。但是，对于 ADI、TXI、RXI 中断，除非在 DMA 传送中指定的寄存器被访问，中断源标志是不会被清除的。如果同一个中断被用作多个通道的一个激活源时，在最高优先级的通道被首先激活时，中断请求标志就被清除了。其它通道的传送请求保留在 DMAC 中待决，激活依优先级次序进行。

当 DTE = 0 时，例如传送完成后，从选定的激活源发出的请求不管 DTA 位如何都不会发送给 DMAC。此时，相应的中断请求就会发送给 CPU。

当与 CPU 中断源重叠时 (DTA = 0)，中断请求标志不能通过 DMAC 清除。

7.4.7.2 通过 USB 请求激活

USB 请求 ($\overline{\text{DREQ}}$ 信号) 被指定为一个 DMAC 激活源。USB 请求由电平检测产生。全地址正常模式时的 USB 请求工作如下所示。

$\overline{\text{DREQ}}$ 信号在保持 High 电平时，DMAC 等待传送请求。 $\overline{\text{DREQ}}$ 信号保持 Low 电平时，DMAC 每释放一次总线就传送 1 个字节，且传送持续进行。在传送过程中当 $\overline{\text{DREQ}}$ 信号升为 High 电平时，传送停止，DMAC 等待传送请求。

7.4.7.3 通过自动请求激活

只有通过寄存器设定才能执行自动请求激活，此时传送一直继续到结束。自由请求激活可选择周期挪用模式和突发模式。

在周期挪用模式，DMAC 向另一个总线主控器释放总线，每释放一次传送 1 字节或 1 字。通常，DMA 周期和 CPU 周期是相互交替的。在突发模式，DMAC 一直占据总线到传送结束，传送持续进行。

7.4.8 DMAC 基本总线周期

DMAC 基本总线周期的时序例子如图 7.4 所示。这是一个字大小传送的例子（从 16 位 2 状态存取空间向 8 位 3 状态存取空间传送）。从 CPU 向 DMAC 转移总线权后，进行读取源地址和写入目标地址。在读取和写入工作之间，在响应其它总线权请求时不释放总线。DMAC 周期与 CPU 周期一样，服从总线控制器的设定。

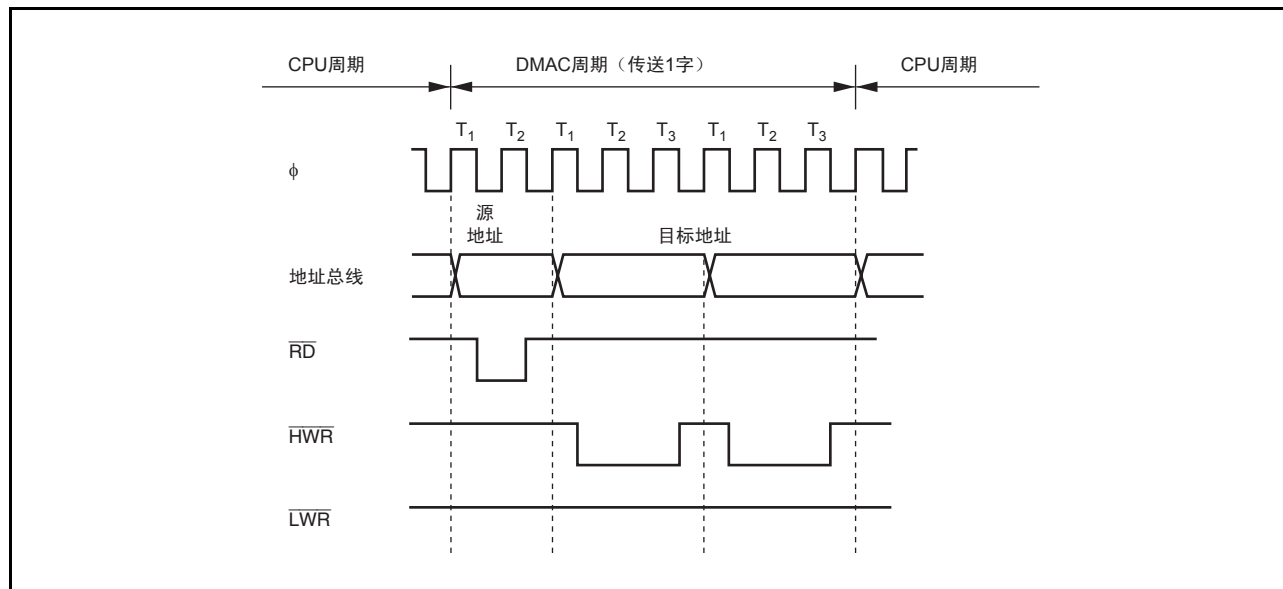


图 7.14 DMA 传送总线时序例子

在访问内部存储器或内部 I/O 寄存器时，地址不被输出到外部地址总线。

7.4.9 DMAC 总线周期 (双地址模式)

7.4.9.1 短地址模式

在图 7.15 表示一个传送例子。其中， $\overline{\text{TEND}}^*$ 输出被允许，进行从外部 8 位 2 状态存取空间向内部 I/O 空间的字节大小短地址模式 (顺序 / 空闲 / 重复模式) 的传送。

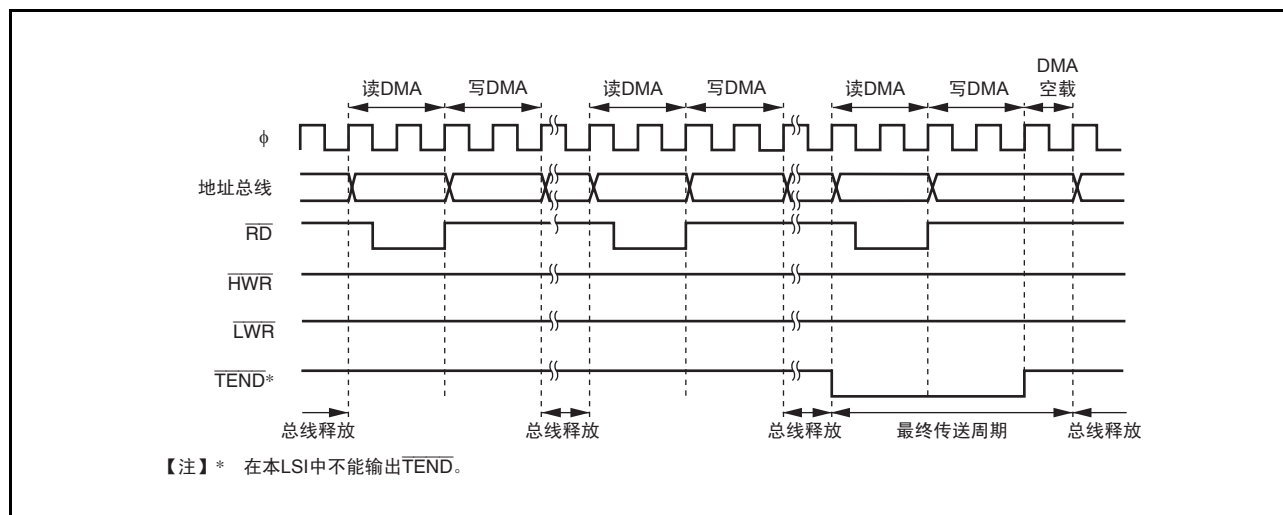


图 7.15 短地址模式传送例子

一次的传送请求进行 1 字节或 1 字传送，传送后，释放总线。总线释放期间，通过 CPU 将总线周期插入 1 次以上。

在传送结束周期 (传送计数器达 0 的周期) 中，在 DMA 写入周期后插入了一个 1 状态 DMA 静止周期。重复模式时，允许 $\overline{\text{TEND}}^*$ 输出后，在传送计数器达到 0 的传送周期中， $\overline{\text{TEND}}^*$ 输出变为 Low 电平。

7.4.9.2 自由地址模式 (周期挪用模式)

在图 7.16 表示一个传送例子。其中， $\overline{\text{TEND}}^*$ 输出被允许，进行从外部 16 位 2 状态存取空间向外部 16 位 2 状态存取空间的字大小全地址模式 (周期挪用模式) 的传送。

【注】* 本 LSI 不支持 $\overline{\text{TEND}}$ 输出。

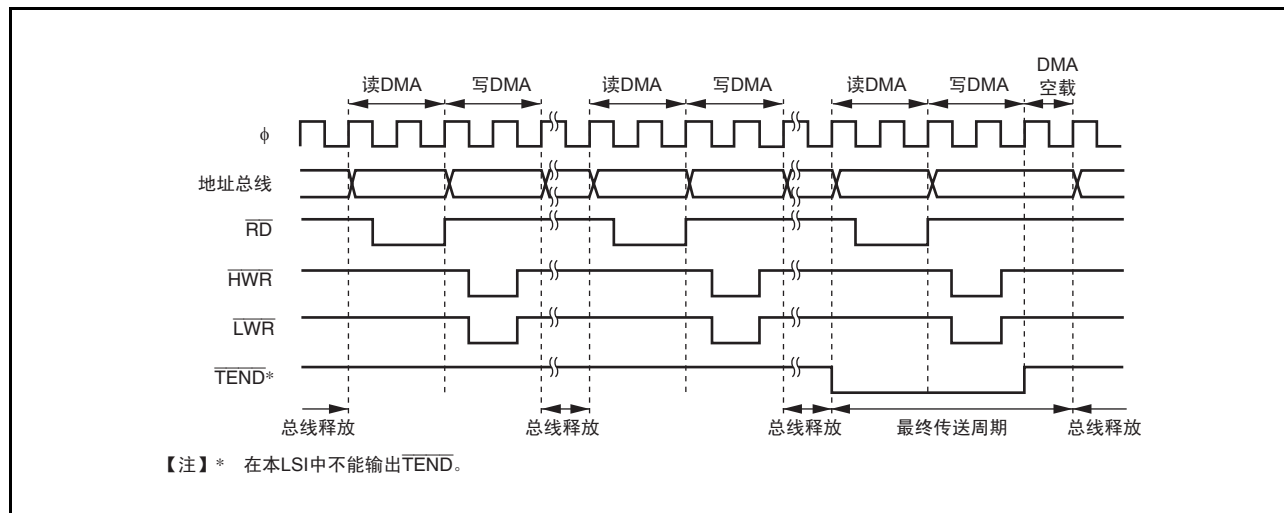


图 7.16 自由地址模式 (周期挪用) 传送例子

进行 1 字节或 1 字的传送，传送后将总线释放。总线释放期间通过 CPU 将总线周期插入一次。传送结束周期 (传送计数器达 0 周期) 在 DMA 写周期后插入一个 1 状态 DMA 静止周期。

7.4.9.3 全地址模式 (突发模式)

在图 7.17 表示一个传送例子。其中， $\overline{\text{TEND}}^*$ 输出被允许，进行从外部 16 位 2 状态存取空间向外部 16 位 2 状态存取空间的字大小全地址模式 (突发模式) 的传送。

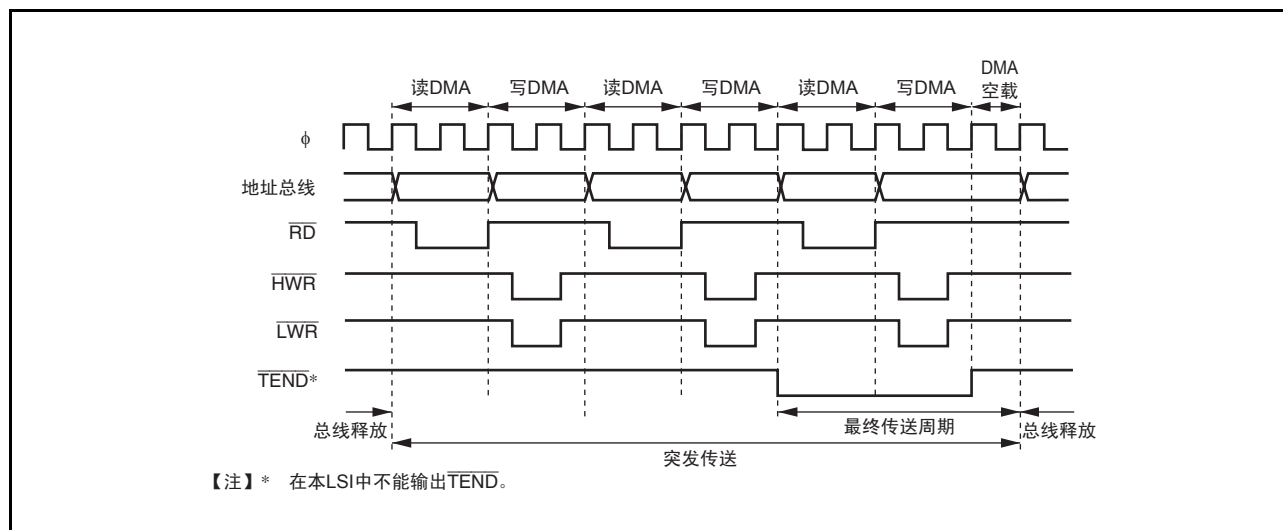


图 7.17 自由地址模式 (突发模式) 传送例子

在突发模式，持续进行 1 字节或 1 字的传送，直到传送结束。

在传送结束周期 (传送计数器达 0 的周期) 中，在 DMA 写周期后插入一个 1 状态 DMA 静止周期。

【注】* 本 LSI 不支持 $\overline{\text{TEND}}^*$ 输出。

突发传送开始后，即使由另一个更高优先级的通道产生请求，该通道还是必须等待，直到突发传送结束。

当设定为突发传送的通道处于传送许可状态时，如果产生了一个 NMI，则 DTME 位被清除，通道就变成传送禁止状态。如果突发传送已经在 DMAC 内部被激活，传送中，1 字节或 1 字的传送完成时，就释放总线并暂停突发传送。如果突发传送的最终传送周期已经在 DMAC 内部被激活，即使清除 DTME 位，传送将继续进行直到结束。

7.4.9.4 全地址模式 (块传送模式)

图 7.18 表示一个传送例子。其中， $\overline{\text{TEND}}^*$ 输出被允许，进行从内部 16 位 1 状态存取空间向外部 16 位 2 状态存取空间的字大小全模式 (块传送模式) 的传送。

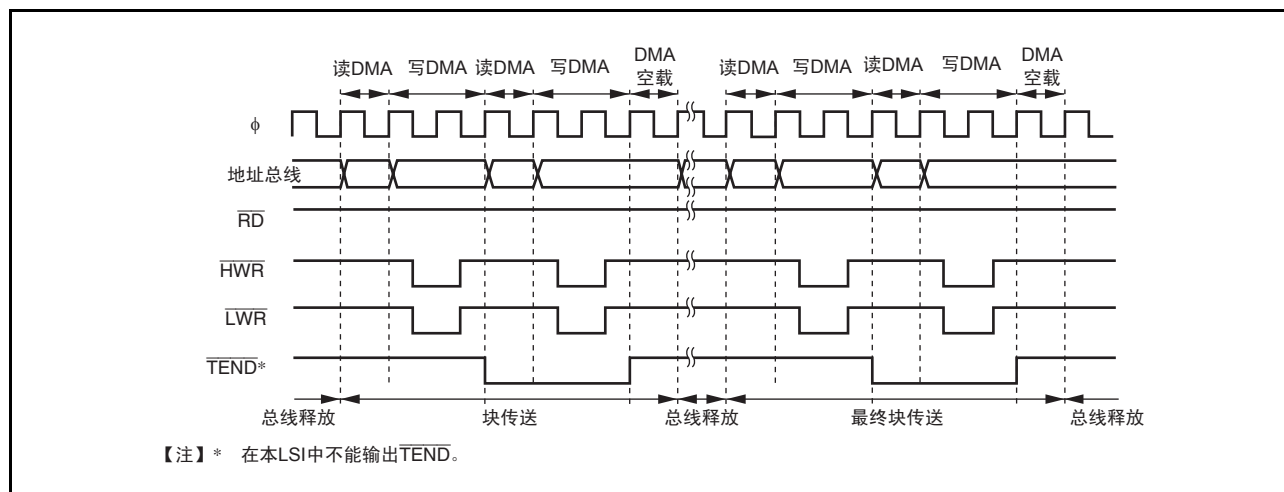


图 7.18 自由地址模式 (块传送模式) 运行例子

1 次的传送请求传送 1 个块，传送后，释放总线。在总线释放期间，通过 CPU 将总线周期插入 1 次以上。

在每块的传送结束周期 (传送计数器达到 0 周期) 中，在 DMA 写周期之后，插入一个 1 状态 DMA 静止周期。

1 个块的传送是不间断的。即使产生 NMI 也不影响块传送工作。

【注】* 本 LSI 不支持 $\overline{\text{TEND}}$ 输出。

7.4.9.5 $\overline{\text{DREQ}}$ 电平激活时序 (正常模式)

将选择 $\overline{\text{DREQ}}$ 信号通道的 DTA 置 1。图 7.19 表示 $\overline{\text{DREQ}}$ 电平激活的正常模式的传送例子。

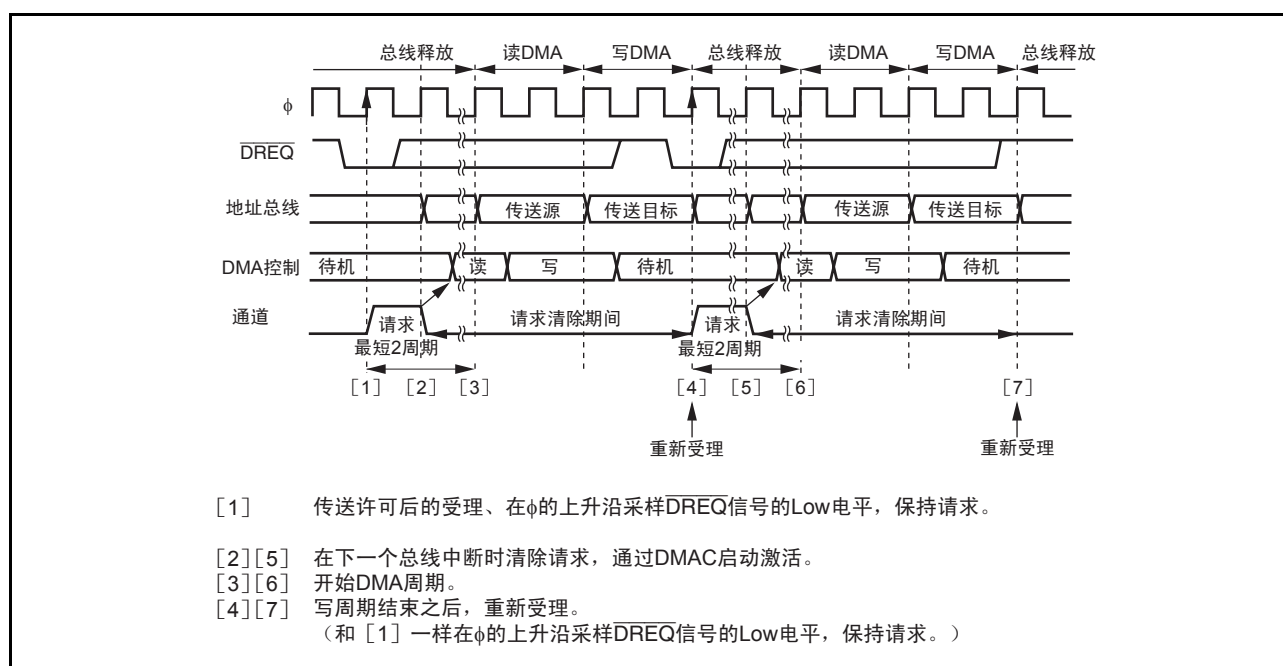


图 7.19 $\overline{\text{DREQ}}$ 电平启动的正常模式传送例子

传送许可状态的 DMABCR 写入周期结束后，下一个 ϕ 的上升沿为起点，在每一个周期进行 $\overline{\text{DREQ}}$ 信号的采样。

通过 $\overline{\text{DREQ}}$ 信号可接受的状态， $\overline{\text{DREQ}}$ 信号的 Low 电平被采样后，在 DMAC 内部保持请求。然后，在 DMAC 中激活，请求就被清除。写周期结束后接受再次开始， $\overline{\text{DREQ}}$ 信号的 Low 电平再次采样，到传送结束为止重复此工作。

【注】 本 LSI 的 $\overline{\text{DREQ}}$ 信号是内部信号，所以不能从管脚输出。

7.4.10 DMAC 多个通道的工作

DMAC 通道间的优先顺序是通道 0 > 通道 1, 通道 A > 通道 B。表 7.9 表示 DMAC 通道间的优先顺序。

表 7.9 DMAC 通道间的优先顺序

短地址模式	自由地址模式	优先顺序
通道 0A	通道 0	高
通道 0B		
通道 1A	通道 1	底
通道 1B		

如果在多个通道同时发布传送请求或在传送期间另一个通道发布传送请求，在释放总线时，DMAC 按照表 7.4 的优先顺序从发布请求的通道中，选择优先级最高的通道。在突发送期间或在块传送模式一个块被传送时，直到传送结束为止，通道切换但不传送。图 7.20 表示通道 0A、0B、1 的传送请求同时发布时的传送例子。

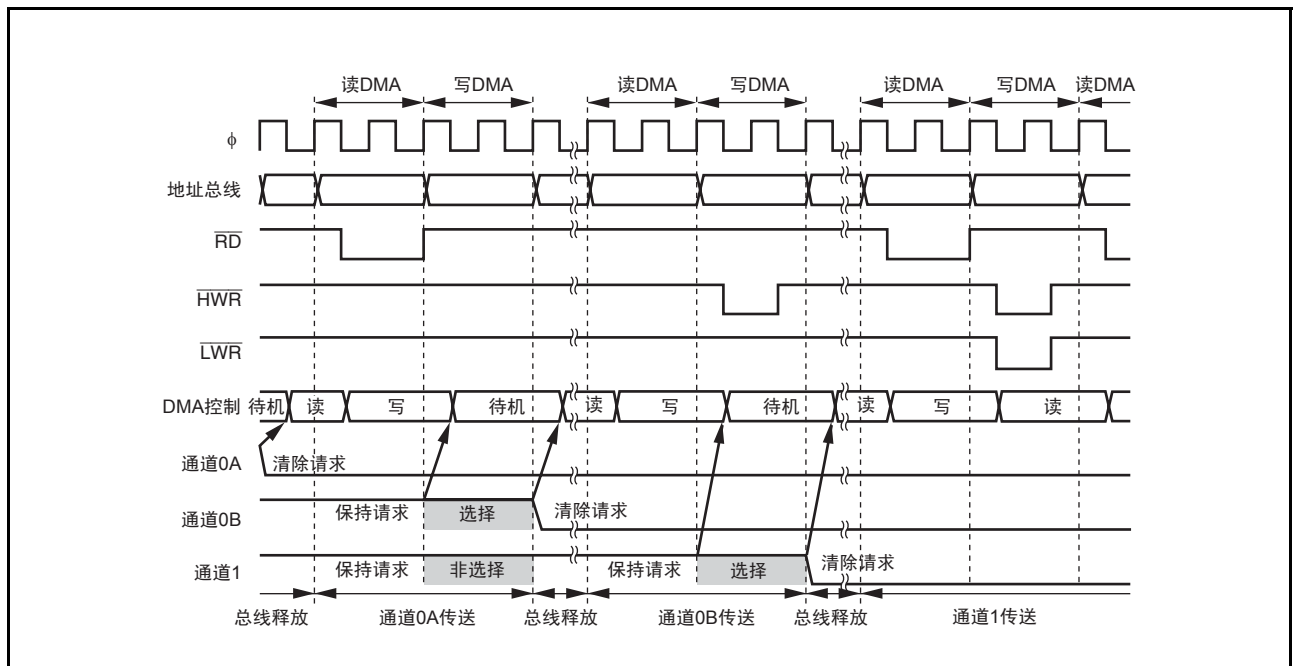


图 7.20 多个通道传送例子

7.4.11 DMAC 和外部总线权请求的关系

在 DMA 周期的读和写之间是没有中断的。这意味着 DMA 周期的外部读和外部写之间，不产生外部总线释放周期。

在连续进行读周期和写周期时，如突发送或块传送，在写周期后，会插入一个外部总线释放状态。

当 DMA 周期的读或写访问内置存储器或内部 I/O 寄存器时，这些 DMA 周期可以作为外部总线释放在同时执行。

7.4.12 NMI 中断和 DMAC

当一个 NMI 中断被请求后，全地址模式的突发模式传送会被中断。在其它模式，NMI 中断对 DMAC 的工作不产生影响。

在全地址模式，DTE 位和 DTME 位两者都置 1 时，其通道成为传送许可状态。随着突发模式的设定，在请求 NMI 中断时，DTME 位被清除。

如果在突发模式传送中 DTME 位被清除，DMAC 在完成正在进行的 1 字节或 1 字的传送时，中断传送后释放总线，向 CPU 转移总线权。

当再次开放传送被中断的通道时，请再次将 DTME 位置 1。图 7.21 所示当一个指定为突发模式传送的通道被一个 NMI 中断造成中断而继续传送的过程。

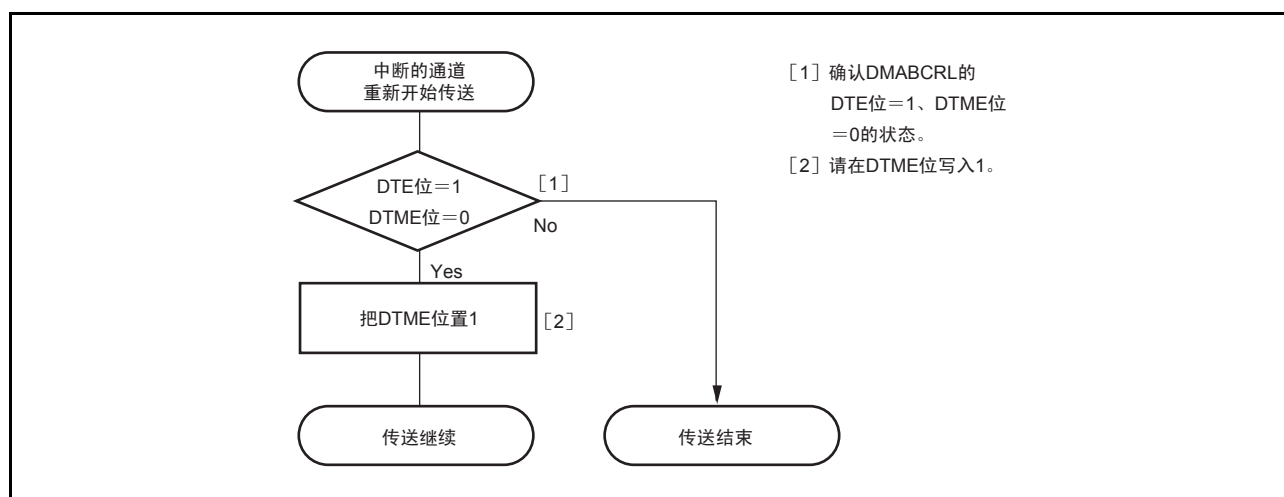


图 7.21 通过 NMI 继续中断通道传送的顺序例子

7.4.13 DMAC 工作的强制终止

如果当前工作中通道的 DTE 位清 0，DMAC 就停止正在进行的 1 字节或 1 字的传送。此后，当再次将 DTE 位置 1，DMAC 的工作重新开始。全地址模式时，DTME 位也是同样。DMAC 通过软件强行终止时的过程如图 7.22 所示。

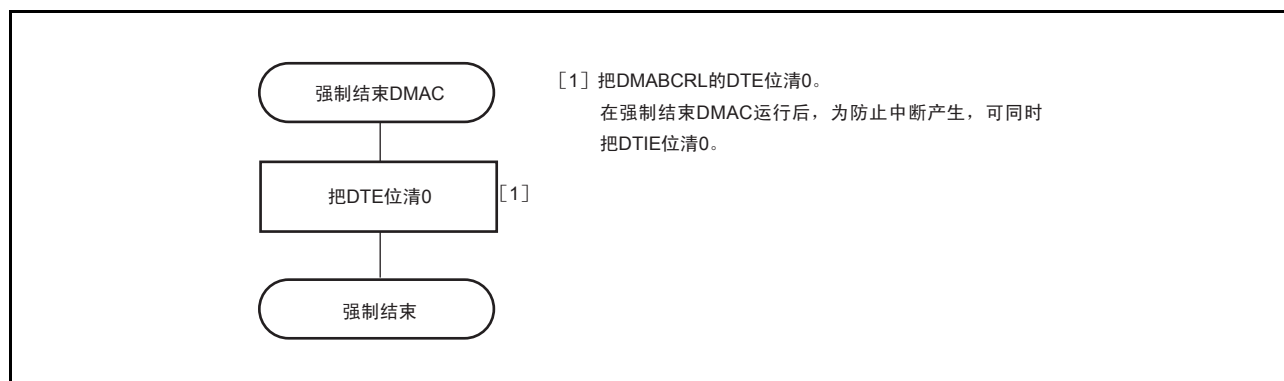


图 7.22 DMAC 强制结束工作的顺序例子

7.4.14 自由地址模式解除

图 7.23 所示为解除和初始化一个设定为全地址模式的通道的过程。当全地址模式已被清除，通道可以用适当的设置过程设定为另一个传送模式。

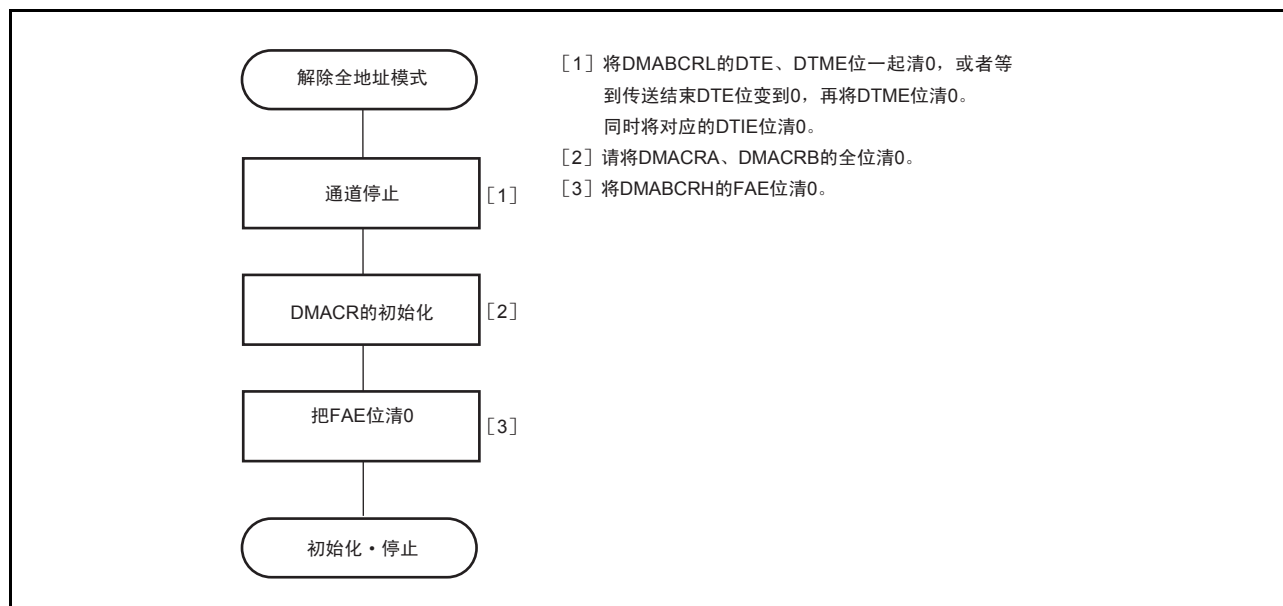


图 7.23 自由地址模式的解除顺序例子

7.5 中断源

DMAC 产生的中断源是传送结束和传送暂停。表 7.10 表示中断源和它们的优先度级序。

表 7.10 中断源和优先度

中断名称	中断源		中断 优先顺序
	短地址模式	自由地址模式	
DEND0A	通过通道 0A 的传送结束中断	通过通道 0 的传送结束中断	高 ↑ 低
DEND0B	通过通道 0B 的传送结束中断	通过通道 0 的传送中断来中断	
DEND1A	通过通道 1A 的传送结束中断	通过通道 1 的传送结束中断	
DEND1B	通过通道 1B 的传送结束中断	通过通道 1 的传送中断来中断	

各中断源通过 DMABCR 对应通道的 DTIE 位，设定为许可或禁止，每个源的中断分别送入中断控制器。每个通道传送结束中断的相对优先级由中断控制器决定，如表 7.10 所示。

传送结束 / 传送暂停中断的框图如图 7.24 所示。在 DTE = 0 的状态下，DTIE 位设定为 1 后，总是产生中断。

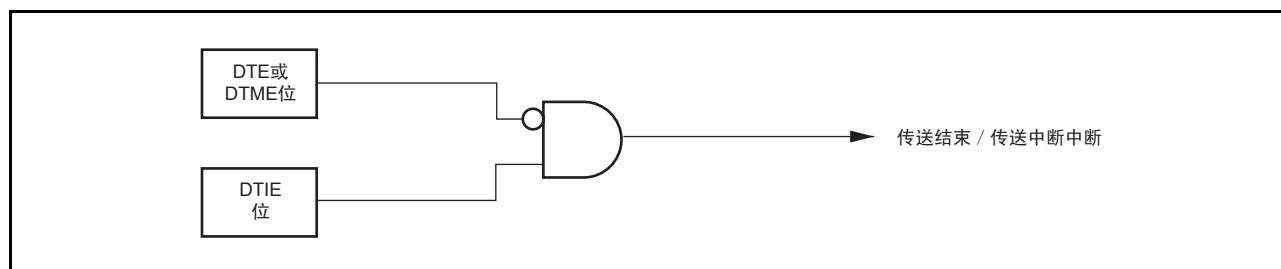


图 7.24 传送结束 / 传送中断的中断块图

在全地址模式，传送暂停中断在 DTIEB = 1 时 DTME 位清 0 后产生。短地址模式和全地址模式在设定的过程中，为了不出现构成中断条件的组合，请设定 DMABCR。

7.6 使用时的注意事项

7.6.1 工作中 DMAC 寄存器存取

除了强制终止以外，工作（包括传送等待状态）的通道的设置请不要变更。如要变更工作中通道的设置，必须在传送禁止状态进行。而且，在 DMA 传送时不要向 DMAC 寄存器写入。

工作（包括传送等待状态）中的 DMAC 寄存器的读取如下所示。

- 在总线周期之前DMAC控制开始一个周期，输出地址值。因此，MAR更新到DMAC传送前的总线周期。

图7.25表示在双地址传送模式下DMAC寄存器的更新时序。

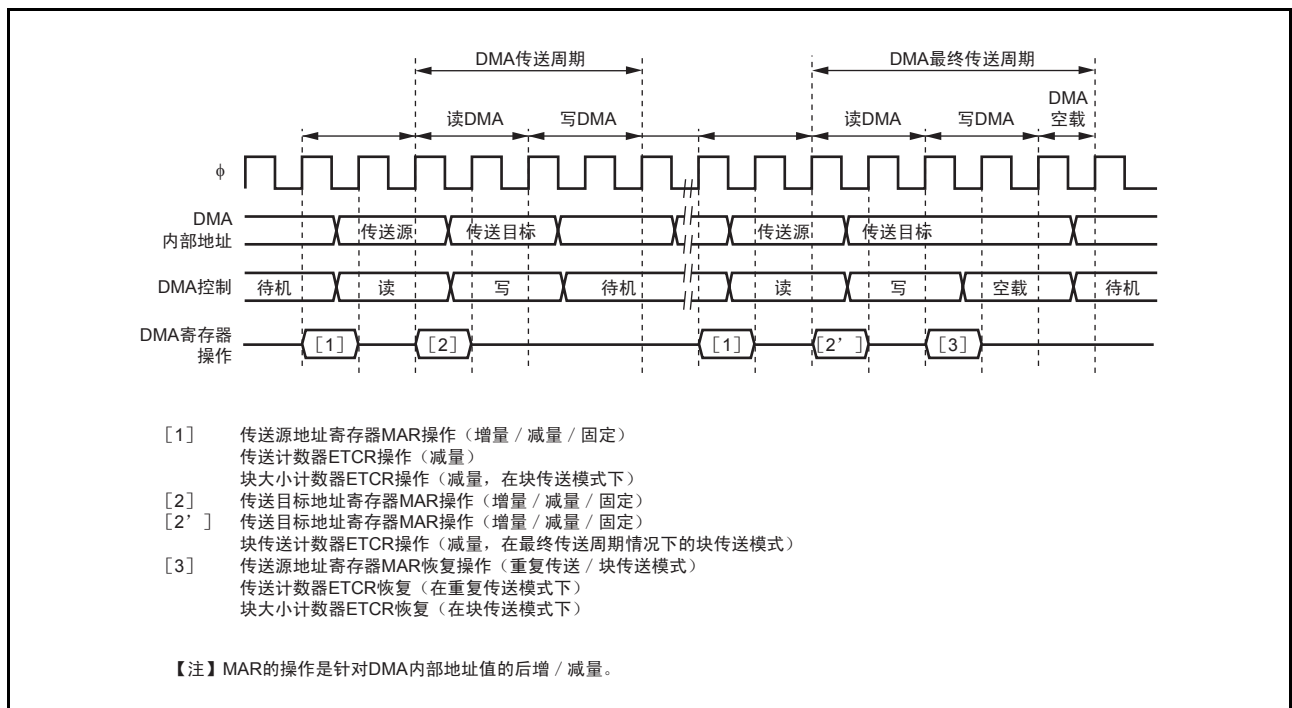


图 7.25 DMAC 寄存器的更新时序

- 如果DMAC寄存器读取周期后，立即发生DMAC的传送周期，DMAC寄存器被读取如图7.26所示。

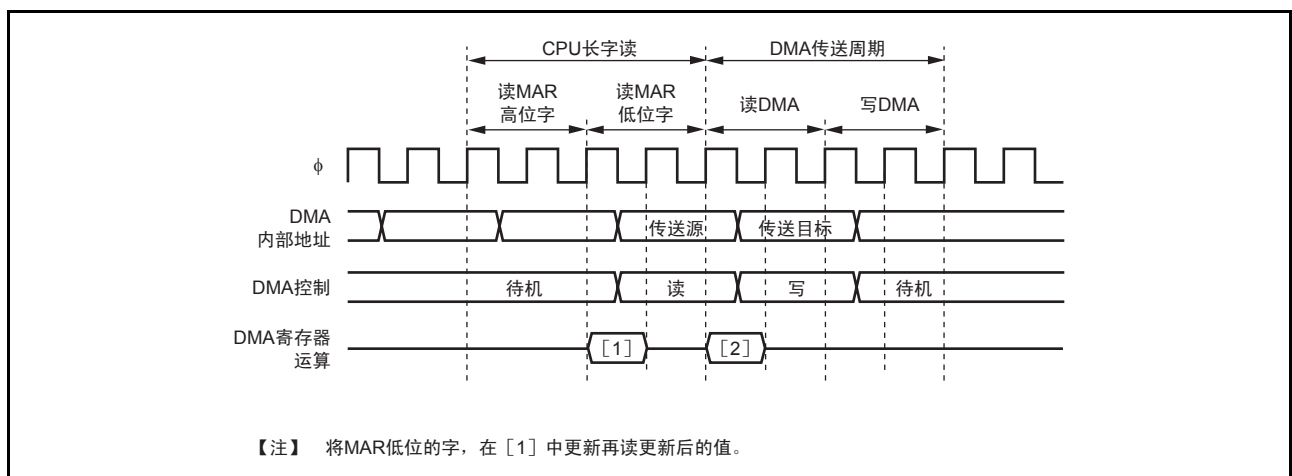


图 7.26 DMAC 寄存器更新和 CPU 读取的竞争

7.6.2 模块停止

MSTPCRA 的 MSTPA7 位置 1 后，DMAC 时钟停止，DMAC 成为模块停止状态。但是，如果任何 DMAC 的通道成为允许状态，MSTPA7 位就不能写入 1。请在 DMAC 工作停止状态下设定。

DMAC 时钟停止后，DMAC 的寄存器不能存取。因为以下的 DMAC 寄存器的设定，在模块停止状态也有效，所以根据需要，请将模块停止模式首先设置为无效。

- 传送结束/暂停中断 (DTE = 0 且 DTIE = 1)

7.6.3 中速模式

DTA 位清 0 时，被指定为 DMAC 传送源的内部中断信号检测出沿。

在中速模式，DMAC 是通过中速时钟工作，同时内置外围模块通过高速时钟工作。因此，如果相应的中断源被 CPU 或另一个 DMAC 通道清除，到产生下一个中断的周期相对于 DMAC 时钟（总线主控器时钟）少于 1 个状态，就不能检测沿，中断会被忽略。

7.6.4 启动源的接受

开始激活源的接受时，按 $\overline{\text{DREQ}}$ 信号按低电平读出都检测出一个 Low 电平。类似的，内部中断时，中断请求被检测出。随之，从一个内部中断或从 $\overline{\text{DREQ}}$ 管脚低电平发出的请求被接受。管脚低电平发生在 DMABCRL 写入变成传送许可状态的执行之前。

当 DMAC 被激活时，应采取任何必要的步骤以防止前一次传送时剩余内部中断或 $\overline{\text{DREQ}}$ 信号的 Low 电平。

7.6.5 传送结束后的内部中断

通过传送结束或中止将 DTE 位清 0 后，即使 DTA = 1 时，选择的内部中断请求还会向 CPU 发送。

同时，在工作被中止时，如果内部 DMAC 已经被激活，则传送被执行，但由于所选择的内部中断，即使在 DTA = 1 时，也不进行标志清除。

传送结束或中止后的内部中断请求，根据需要可通过 CPU 进行处理。

7.6.6 通道的再次设定

多个通道在传送许可状态下，为了多个通道再激活，使用传送结束中断的异处理以及对 DMABCR 的控制位操作进行异处理。

注意，特别是在进行 DMABCR 的读/写之间，产生了多重中断，而且在新的中断处理时执行了 DMABCR 操作，通过原始中断处理程序写入 DMABCR 的数据会变得不正确，多次中断操作也会使写的结果无效。应确保不要用多次中断执行 DMABCR 重叠的运算，而且使用位操作指令时，从读到写是无区别的。同时，当 DTE 及 DTME 位通过 DMAC 清除或写入 0 时，DTE/DTME = 0 的状态必须首先读到，否则就不能被 CPU 写入 1。

8. I/O 端口

H8S/2218 群的端口功能一览如表 8.1 所示，H8S/2212 群的端口功能一览如表 8.2 所示。各管脚是兼用内置外围模块的输入 / 输出管脚和外部中断输入管脚。各管脚由控制输入 / 输出的数据方向寄存器（DDR）、保存输出数据的数据寄存器（DR）和读取管脚状态的端口寄存器（PORT）构成。输入专用管脚没有 DR、DDR。

端口 A ~ E 内置输入上拉 MOS 和一个输入上拉 MOS 控制寄存器（PCR），用来控制输入上拉 MOS 的 ON/OFF 状态。端口 3 内置漏极开路控制寄存器（ODR），可选择输出缓冲 PMOS 的 ON/OFF 状态。

所有的 I/O 端口都可驱动 1 个 TTL 负载和 30pF 的容量负载。

表 8.1 H8S/2218 群的端口功能一览

端口名	概要状态	模式 4、5	模式 6	模式 7	输入 / 输出及其它
端口 1	与 TPU 输入 / 输出、中断输入、地址总线输出管脚兼用的通用输入 / 输出端口	P17/TIOCB2/TCLKD P16/TIOCA2/IRQ1 P15/TIOCB1/TCLKC P14/TIOCA1/IRQ0			施密特触发器 (IRQ1、IRQ0)
		P13/TIOCD0/TCLKB/A23		P13/TIOCD0/TCLKB	
		P12/TIOCC0/TCLKA/A22		P12/TIOCC0/TCLKA	
		P11/TIOCB0/A21		P11/TIOCB0	
		P10/TIOCA0/A20		P10/TIOCA0	
端口 3	与 SCI_0 输入 / 输出、中断输入管脚兼用的通用输入 / 输出端口	P36 P32/SCK0/IRQ4 P31/RxD0 P30/TxD0			可进行漏极开路输出 施密特触发器输入 (IRQ4)
端口 4	与 A/D 转换器的模拟输入管脚兼用的通用输入端口	P43/AN3 P42/AN2 P41/AN1 P40/AN0			
端口 7	与总线控制输出、手动复位输入管脚兼用的通用输入 / 输出端口	P74/MRES P71/CS5 P70/CS4		P74/MRES P71 P70	
端口 9	与 A/D 转换器的模拟输入管脚兼用的通用输入端口	P97/AN15 P96/AN14			
端口 A	与 SCI_2 的输入 / 输出管脚、地址总线输出管脚兼用的通用输入 / 输出端口	PA3/A19/SCK2 PA2/A18/RxD2 PA1/A17/TxD2 PA0/A16		PA3/SCK2 PA2/RxD2 PA1/TxD2 PA0	内置输入上拉 MOS 输入 可进行漏极开路输出

端口名	概要状态	模式 4、5	模式 6	模式 7	输入 / 输出及其它
端口 B	与地址总线输出管脚兼用的通用输入 / 输出端口	PB7/A15 PB6/A14 PB5/A13 PB4/A12 PB3/A11 PB2/A10 PB1/A9 PB0/A8		PB7 PB6 PB5 PB4 PB3 PB2 PB1 PB0	内置输入上拉 MOS 输入
端口 C	与地址总线输出管脚兼用的通用输入 / 输出端口	A7	DDR = 0 时 PC7 DDR = 1 时 A7	PC7	内置输入上拉 MOS 输入
		A6	DDR = 0 时 PC6 DDR = 1 时 A6	PC6	
		A5	DDR = 0 时 PC5 DDR = 1 时 A5	PC5	
		A4	DDR = 0 时 PC4 DDR = 1 时 A4	PC4	
		A3	DDR = 0 时 PC73 DDR = 1 时 A3	PC3	
		A2	DDR = 0 时 PC2 DDR = 1 时 A2	PC2	
		A1	DDR = 0 时 PC1 DDR = 1 时 A1	PC1	
		A0	DDR = 0 时 PC0 DDR = 1 时 A0	PC0	
端口 D	与数据总线输入 / 输出管脚兼用的通用输入 / 输出端口	D15 D14 D13 D12 D11 D10 D9 D8		PD7 PD6 PD5 PD4 PD3 PD2 PD1 PD0	内置上拉 MOS 输入

端口名	概要状态	模式 4、5	模式 6	模式 7	输入 / 输出及其它
端口 E	与数据总线输入 / 输出管脚兼用的通用输入 / 输出端口	8 位总线模式时 PE7 16 位总线模式时 D7		PE7	内置输入上拉 MOS
		8 位总线模式时 PE6 16 位总线模式时 D6		PE6	
		8 位总线模式时 PE5 16 位总线模式时 D5		PE5	
		8 位总线模式时 PE4 16 位总线模式时 D4		PE4	
		8 位总线模式时 PE3 16 位总线模式时 D3		PE3	
		8 位总线模式时 PE2 16 位总线模式时 D2		PE2	
		8 位总线模式时 PE1 16 位总线模式时 D1		PE1	
		8 位总线模式时 PE0 16 位总线模式时 D0		PE0	
端口 F	与总线控制信号输入 / 输出、中断输入管脚兼用的通用输入 / 输出端口	DDR = 0 时 PF7 DDR = 1 时 (复位后) ϕ		DDR = 0 时 (复位后) PF7 DDR = 1 时 ϕ	施密特触发器 输入 ($\overline{\text{IRQ3}}$ 、 $\overline{\text{IRQ2}}$)
		$\overline{\text{AS}}$		PF6	
		$\overline{\text{RD}}$		PF5	
		$\overline{\text{HWR}}$		PF4	
		8 位总线模式时 $\overline{\text{PF3/ADTRG/IRQ3}}$ 16 位总线模式时 $\overline{\text{LWR}}$		$\overline{\text{PF3/ADTRG/IRQ3}}$	
		WAITE = 0 时 (复位后) PF2 WAITE = 1 时 $\overline{\text{WAIT}}$		PF2	
		BRLE = 0 时 (复位后) PF1 BRLE = 1 时 $\overline{\text{BACK}}$		PF1	
		BRLE = 0 时 (复位后) $\overline{\text{PF0/IRQ2}}$ BRLE = 1 时 $\overline{\text{BREQ/IRQ2}}$		$\overline{\text{PF0/IRQ2}}$	
端口 G	与总线控制输出、中断输入管脚兼用的通用输入 / 输出端口	DDR = 0 时 $\overline{\text{PG4}}$ (模式 6 复位后) DDR = 1 时 $\overline{\text{CS0}}$ (模式 4、5 复位后)		$\overline{\text{PG4}}$	施密特触发器 输入 ($\overline{\text{IRQ7}}$)
		DDR = 0 时 $\overline{\text{PG3}}$ DDR = 1 时 $\overline{\text{CS1}}$		$\overline{\text{PG3}}$	
		DDR = 0 时 $\overline{\text{PG2}}$ DDR = 1 时 $\overline{\text{CS2}}$		$\overline{\text{PG2}}$	
		DDR = 0 时 $\overline{\text{PG1/IRQ7}}$ DDR = 1 时 $\overline{\text{CS3/IRQ7}}$		$\overline{\text{PG1/IRQ7}}$	

表 8.2 H8S/2212 群的管脚功能一览

端口名	概要	模式 7	输入 / 输出状态及其他
端口 1	与 TPU 输入 / 输出、中断输入管脚兼用的通用输入 / 输出端口	P17/TIOCB2/TCLKD P16 /TIOCA2/ $\overline{\text{IRQ1}}$ P15 /TIOCB1/TCLKC P14 /TIOCA1/ $\overline{\text{IRQ0}}$ P13 /TIOCD0/TCLKB P12 /TIOCC0/TCLKA P11/TIOCB0 P10/TIOCA0	施密特触发器输入 ($\overline{\text{IRQ1}}$ 、 $\overline{\text{IRQ0}}$)
端口 3	与 SCI_0 输入 / 输出、中断输入管脚兼用的通用输入 / 输出端口	P36 P32/SCK0/ $\overline{\text{IRQ4}}$ P31/RxD0 P30/TxD0	可进行漏极开路输出 施密特触发器输入 ($\overline{\text{IRQ4}}$)
端口 4	与 A/D 转换器的模拟输入管脚兼用的通用输入端口	P43/AN3 P42/AN2 P41/AN1 P40/AN0	
端口 7	通用输入 / 输出端口	P77* P76* P75*	
端口 9	与 A/D 转换器的模拟输入管脚兼用的通用输入端口	P97/AN15 P96/AN14	
端口 A	与 SCI_2 的输入 / 输出管脚兼用的通用输入 / 输出端口	PA3/SCK2 PA2/RxD2 PA1/TxD2	内置输入上拉 MOS 漏极开路输出
端口 E	通用输入 / 输出端口	PE7 PE6 PE5 PE4 PE3 PE2 PE1 PE0	内置输入上拉 MOS
端口 F	与中断输入管脚兼用的通用输入 / 输出端口	DDR = 0 时 (复位后) PF7 DDR = 1 时 ϕ PF3/ $\overline{\text{ADTRG}}$ / $\overline{\text{IRQ3}}$ PF0/ $\overline{\text{IRQ2}}$	施密特触发器输入 ($\overline{\text{IRQ3}}$ 、 $\overline{\text{IRQ2}}$)
端口 G	与中断输入管脚兼用的通用输入 / 输出端口	PG1/ $\overline{\text{IRQ7}}$ PG0*	施密特触发器输入 ($\overline{\text{IRQ7}}$)

【注】 * 仅在 EMLE = 0 时可使用。在使用 H-UDI 时不可使用。

8.1 端口 1

H8S/2218 群的端口 1 是兼用地址总线管脚、TPU 输入 / 输出管脚、外部中断输入管脚的 8 位输入 / 输出端口。H8S/2212 群的端口 1 是兼容 TPU 输入 / 输出管脚、外部中断输入管脚的 8 位输入 / 输出端口。端口 1 有以下寄存器。

- 端口 1 数据方向寄存器 (P1DDR)
- 端口 1 数据寄存器 (P1DR)
- 端口 1 寄存器 (PORT1)

8.1.1 端口 1 数据方向寄存器 (P1DDR)

P1DDR 以每位指定端口 1 管脚的输入 / 输出。

本寄存器是写专用寄存器，所以不能使用位操作指令进行写入。

位	位名	初始值	R/W	说 明
7	P17DDR	0	W	(H8S/2218 群) 在模式 4 ~ 6 时 当用 PFCR 的 AE3 ~ AE0 位允许地址输出时，P13 ~ P10 管脚成为地址输出。P17 ~ P14 管脚和地址输出禁止时的 P13 ~ P10 管脚将相应 P1DDR 位置 1 后就成为输出端口，清 0 相应 P1DDR 位后成则为为输入端口。
6	P16DDR	0	W	
5	P15DDR	0	W	
4	P14DDR	0	W	
3	P13DDR	0	W	
2	P12DDR	0	W	
1	P11DDR	0	W	
0	P10DDR	0	W	
				(H8S/2212 群) 将 P1DDR 位置 1 后对应端口 1 管脚成为输出端口，清 0 后成为输入端口。

8.1.2 端口 1 数据寄存器 (P1DR)

P1DR 保存端口 1 的输出数据。

位	位名	初始值	R/W	说 明
7	P17DR	0	R/W	给管脚保存输出数据以作为通用输出端口使用。
6	P16DR	0	R/W	
5	P15DR	0	R/W	
4	P14DR	0	R/W	
3	P13DR	0	R/W	
2	P12DR	0	R/W	
1	P11DR	0	R/W	
0	P10DR	0	R/W	

8.1.3 端口 1 寄存器 (PORT1)

PORT1 表示的是端口 1 的管脚状态。

位	位名	初始值	R/W	说明
7	P17	— *	R	P1DDR 置 1 时，读取端口 1，则读取 P1DR 的值。若 P1DDR 清 0 时，读取端口 1，则读取管脚状态。
6	P16	— *	R	
5	P15	— *	R	
4	P14	— *	R	
3	P13	— *	R	
2	P12	— *	R	
1	P11	— *	R	
0	P10	— *	R	

【注】 * 由 P17 ~ P10 的管脚状态决定。

8.1.4 管脚功能

8.1.4.1 H8S/2218 群的管脚功能

端口 1 管脚兼用地址总线 (A23 ~ A20) 输出管脚、TPU 输入 / 输出管脚、外部中断输入 ($\overline{\text{IRQ0}}$ 、 $\overline{\text{IRQ1}}$) 管脚。寄存器的设定值和管脚功能的关系如下所示。

表 8.3 P17 的管脚功能

TPU 通道 2 的设定 *	输出设定	输出设定或初始值	
P17DDR	—	0	1
管脚功能	TIOCB2 输出管脚	P17 输入管脚	P17 输出管脚
		TIOCB2 输入管脚	
	TCLKD 输入管脚		

【注】 * TPU 通道的设定请参考 9. 16 位定时器脉冲单元 (TPU)。

表 8.4 P16 的管脚功能

TPU 通道 2 的设定 *1	输出设定	输出设定或初始值	
P16DDR	—	0	1
管脚功能	TIOCA2 输出管脚	P16 输入管脚	P16 输出管脚
		TIOCA2 输入管脚	
	$\overline{\text{IRQ1}}$ 输入管脚 *2		

【注】 *1 TPU 通道的设定请参考 9. 16 位定时器脉冲单元 (TPU)。

*2 作为外部中断管脚使用时，该管脚不可兼用。

表 8.5 P15 的管脚功能

TPU 通道 1 的设定 *	输出设定	输出设定或初始值	
P15DDR	—	0	1
管脚功能	TIOCB1 输出管脚	P15 输入管脚	P15 输出管脚
		TIOCB1 输入管脚	
	TCLKC 输入管脚		

【注】 * TPU 通道的设定请参考 9. 16 位定时器脉冲单元 (TPU)。

表 8.6 P14 的管脚功能

TPU 通道 1 的设定 *1	输出设定	输出设定或初始值	
P14DDR	—	0	1
管脚功能	TIOCA1 输出管脚	P14 输入管脚	P14 输出管脚
		TIOCA1 输入管脚	
	IRQ0 输入管脚 *2		

【注】 *1 TPU 通道的设定请参考 9. 16 位定时器脉冲单元 (TPU)。

*2 作为外部中断管脚使用时, 该管脚不可兼用。

表 8.7 P13 的管脚功能

AE3 ~ AE0*2	(B'1111) 以外			B'1111
TPU 通道 0 的设定 *1	输出设定	输入设定或初始值		—
P13DDR	—	0	1	—
管脚功能	TIOCD0 输出管脚	P13 输入管脚	P13 输出管脚	A23 输出管脚 *2
		TIOCD0 输入管脚		
	TCLKB 输入管脚			

【注】 *1 TPU 通道的设定请参考 9. 16 位定时器脉冲单元 (TPU)。

*2 模式 4、5、6 时有效。

表 8.8 P12 的管脚功能

AE3 ~ AE0*2	(B'1111) 以外			B'1111
TPU 通道 0 的设定 *1	输出设定	输入设定或初始值		—
P12DDR	—	0	1	—
管脚功能	TIOCC0 输出管脚	P12 输入管脚	P12 输出管脚	A22 输出管脚 *2
		TIOCC0 输出管脚		
	TCLKA 输入管脚			

【注】 *1 TPU 通道的设定请参考 9. 16 位定时器脉冲单元 (TPU)。

*2 模式 4、5、6 时有效。

表 8.9 P11 的管脚功能

AE3 ~ AE0*2	(B'1110 ~ B'1111) 以外		B'1110 ~ B'1111
TPU 通道 0 的设定 *1	输出设定	输入设定或初始值	
P11DDR	—	0	1
管脚功能	TIOCB0 输出管脚	P11 输入管脚	P11 输出管脚
		TIOCB0 输入管脚	
			A21 输出管脚 *2

表 8.10 P10 的管脚功能

AE3 ~ AE0*2	(B'1101 ~ B'1111) 以外		B'1101 ~ B'1111
TPU 通道 0 的设定 *1	输出设定	输入设定或初始值	
P10DDR	—	0	1
管脚功能	TIOCA0 输出管脚	P10 输入管脚	P10 输出管脚
		TIOCA0 输入管脚	
			A20 输出管脚 *2

【注】 *1 TPU 通道的设定请参考 9. 16 位定时器脉冲单元 (TPU)。

*2 模式 4、5、6 时有效。

8.1.4.2 H8S/2212 群的管脚功能

端口 1 兼用 TPU 输入 / 输出，外部中断输入 ($\overline{\text{IRQ0}}$ 、 $\overline{\text{IRQ1}}$)。寄存器的设定值和端口功能的关系如下所示。

表 8.11 P17 的管脚功能

TPU 通道 2 的设定 *	输出设定	输入设定或初始值	
P17DDR	—	0	1
管脚功能	TIOCB2 输出管脚	P17 输入管脚	P17 输出管脚
		TIOCB2 输入管脚	
		TCLKD 输入管脚	

【注】 * TPU 通道的设定请参考 9. 16 位定时器脉冲单元 (TPU)。

表 8.12 P16 的管脚功能

TPU 通道 2 的设定 *1	输出设定	输入设定或初始值	
P16DDR	—	0	1
管脚功能	TIOCA2 输出管脚	P16 输入管脚	P16 输出管脚
		TIOCA2 输入管脚	
		IRQ1 输入管脚 *2	

【注】 *1 TPU 通道的设定请参考 9. 16 位定时器脉冲单元 (TPU)。

*2 作为外部中断管脚使用时，该管脚不可兼用。

表 8.13 P15 的管脚功能

TPU 通道 1 的设定 *	输出设定	输入设定或初始值	
P15DDR	—	0	1
管脚功能	TIOCB1 输出管脚	P15 输入管脚	P15 输出管脚
		TIOCB1 输入管脚	
	TCLKC 输入管脚		

【注】 * TPU 通道的设定请参考 9. 16 位定时器脉冲单元 (TPU)。

表 8.14 P14 的管脚功能

TPU 通道 1 的设定 *1	输出设定	输入设定或初始值	
P14DDR	—	0	1
管脚功能	TIOCA1 输出管脚	P14 输入管脚	P14 输出管脚
		TIOCA1 输入管脚	
	IRQ0 输入管脚 *2		

【注】 *1 TPU 通道的设定请参考 9. 16 位定时器脉冲单元 (TPU)。

*2 作为外部中断管脚使用时, 该管脚不可兼用。

表 8.15 P13 的管脚功能

TPU 通道 0 的设定 *	输出设定	输入设定或初始值	
P13DDR	—	0	1
管脚功能	TIOCD0 输出管脚	P13 输入管脚	P13 输出管脚
		TIOCD0 输入管脚	
	TCLKB 输入管脚		

【注】 * TPU 通道的设定请参考 9. 16 位定时器脉冲单元 (TPU)。

表 8.16 P12 的管脚功能

TPU 通道 0 的设定 *	输出设定	输入设定或初始值	
P12DDR	—	0	1
管脚功能	TIOCC0 输出管脚	P12 输入管脚	P12 输出管脚
		TIOCC0 输入管脚	
	TCLKA 输入管脚		

【注】 * TPU 通道的设定请参考 9. 16 位定时器脉冲单元 (TPU)。

表 8.17 P11 的管脚功能

TPU 通道 0 的设定 *	输出设定	输入设定或初始值	
P11DDR	—	0	1
管脚功能	TIOCB0 输出管脚	P11 输入管脚	P11 输出管脚
		TIOCB0 输入管脚	

表 8.18 P10 的管脚功能

TPU 通道 0 的设定 *	输出设定	输入设定或初始值	
P10DDR	—	0	1
管脚功能	TIOCA0 输出管脚	P10 输入管脚	P10 输出管脚
		TIOCA0 输入管脚	

【注】 * TPU 通道的设定请参考 9. 16 位定时器脉冲单元 (TPU)。

8.2 端口 3

端口 3 是兼用 SCI 输入 / 输出管脚、外部中断输入（ $\overline{\text{IRQ4}}$ ）管脚的 4 位的输入 / 输出端口。H8S/2218 群的端口 3 和 H8S/2212 群的端口 3 具有同样功能。端口 3 包括以下寄存器。

- 端口 3 数据方向寄存器（P3DDR）
- 端口 3 数据寄存器（P3DR）
- 端口 3 寄存器（PORT3）
- 端口 3 漏极开路控制寄存器（P3ODR）

8.2.1 端口 3 数据方向寄存器（P3DDR）

P3DDR 以每位指定端口 3 的各管脚的输入 / 输出。

本寄存器是写专用寄存器所以不能使用位操作指令写入。

位	位名	初始值	R/W	说 明
7	—	不定	—	保留位 该位未定义且不能被修改
6	P36DDR	0	W	P3DDR 置 1 时对应端口 3 的管脚为输出管脚，清 0 时为输入管脚。
5~3	—	不定	—	保留位 该位未定义且不能被修改
2	P32DDR	0	W	P3DDR 置 1 时对应端口 3 的管脚为输出管脚，清 0 时为输入管脚。
1	P31DDR	0	W	
0	P30DDR	0	W	

8.2.2 端口 3 数据寄存器（P3DR）

P3DR 保存端口 3 的各管脚输出数据。

位	位名	初始值	R/W	说 明
7	—	不定	—	保留位 该位未定义且不能被修改。
6	P36DDR	0	R/W	给管脚保存输出数据以作为通用输出端口使用。
5~3	—	不定	—	保留位 该位未定义且不能被修改。
2	P32DDR	0	R/W	给管脚保存输出数据以作为通用输出端口使用。
1	P31DDR	0	R/W	
0	P30DDR	0	R/W	

8.2.3 端口 3 寄存器 (PORT3)

PORT3 表示端口 3 的管脚的状态。

位	位名	初始值	R/W	说明
7	—	不定	—	保留位 该位未定义。
6	P36	— *	R	P3DDR 置 1 时, 读取端口 3, 则读取 P3DR 的值。若 P3DDR 清 0 时, 读取端口 3, 则读取管脚状态。
5~3	—	不定	—	保留位 该位未定义。
2	P32	— *	R	P3DDR 置 1 时, 读取端口 3, 则读取 P3DR 的值。若 P3DDR 清 0 时, 读取端口 3, 则读取管脚状态。
1	P31	— *	R	
0	P30	— *	R	

【注】 * 根据 P36、P32 ~ P30 管脚的状态决定。

8.2.4 端口 3 漏极开路控制寄存器 (P3ODR)

P3ODR 控制端口 3 的各管脚 PMOS 的 ON/OFF。

位	位名	初始值	R/W	说明
7	—	不定	—	保留位 该位未定义且不能被修改。
6	P36ODR	0	R/W	P3ODR 置 1 时相应端口 3 的各管脚为 NMOS 漏极开路输出管脚, 清 0 时为 CMOS 输出管脚。
5~3	—	不定	—	保留位 该位未定义且不能被修改。
2	P32ODR	0	R/W	P3ODR 置 1 时相应端口 3 的各管脚为 NMOS 漏极开路输出管脚, 清 0 时为 CMOS 输出管脚。
1	P31ODR	0	R/W	
0	P30ODR	0	R/W	

8.2.5 管脚功能

端口 3 管脚兼用 SCI 输入 / 输出管脚、外部中断输入（ $\overline{\text{IRQ4}}$ ）管脚。寄存器的设定值和管脚功能的关系如下所示。P36 管脚必须用作 USB 的 D+ 上拉控制输出管脚。详情请参考 9. 16 位定时器脉冲单元（TPU）。

表 8.19 P36 的管脚构成

P36DDR	0	1
管脚功能	P36 输入管脚	P36 输出管脚 (USB 的 D+ 上拉控制输出)

表 8.20 P32 的管脚功能

SCR_0 的 CKE1	0				1
SMR_0 的 $\overline{\text{C/A}}$	0			1	—
SCR_0 的 CKE0	0		1	—	—
P32DDR	0	1	—	—	—
管脚功能	P32 输入管脚	P32 输出管脚	$\overline{\text{SCK0}}$ 输出管脚	SCK0 输出管脚	SCK0 输出管脚
	$\overline{\text{IRQ4}}$ 输入管脚 *				

【注】 * 作为外部中断管脚使用时，该管脚不可兼用。

表 8.21 P31 的管脚功能

SCR_0 的 RE	0		1
P31DDR	0	1	—
管脚功能	P31 输入管脚	P31 输出管脚	RxD0 输入管脚

表 8.22 P30 的管脚功能端口 4

SCR_0 的 TE	0		1
P30DDR	0	1	—
管脚功能	P30 输入管脚	P30 输出管脚	TxD0 输入管脚

8.3 端口 4

端口 4 是兼用 A/D 转换器模拟输入管脚的 4 位输入端口。H8S/2218 群的端口 4 和 H8S/2212 群的端口 4 具有相同功能。端口 4 包括以下寄存器。

- 端口 4 寄存器 (PORT4)

8.3.1 端口 4 寄存器 (PORT4)

PORT4 表示端口 4 的管脚的状态。

位	位名	初始值	R/W	说 明
7~4	—	不定	—	保留位 该位未定义。
3	P43	— *	R	读取该位得到管脚状态。
2	P42	— *	R	
1	P41	— *	R	
0	P40	— *	R	

【注】 * 由 P43 ~ P40 管脚状态决定。

8.3.2 管脚功能

端口 4 兼用 A/D 转换器的模拟输入 (AN3 ~ 0) 管脚。

8.4 端口 7

H8S/2218 群的端口 7 是兼用总线控制输出管脚、手动复位输入管脚的 3 位的输入 / 输出端口。

H8S/2212 群的端口 7 是与 H-UDI 管脚兼用的 3 位的输入 / 输出端口。端口 7 有以下寄存器。

- 端口 7 数据方向寄存器 (P7DDR)
- 端口 7 数据寄存器 (P7DR)
- 端口 7 寄存器 (PORT7)

8.4.1 端口 7 数据方向寄存器 (P7DDR)

P7DDR 是以每位指定端口 7 各管脚的输入 / 输出。

本寄存器是写专用寄存器所以不能使用位操作指令进行写入。

详细请参考 2.9.4 包括写专用位的寄存器存取方法。

位	位名	初始值	R/W	说 明
7	P77DDR	0	W	(H8S/2218 群)
6	P76DDR	0	W	保留位
5	P75DDR	0	W	该位未定义且不能被修改。 (H8S/2212 群) EMLE = 1 时, P77 ~ P75 管脚用作 H-UDI 管脚 (TDO、TMS、TCK)。 EMLE = 0 时, P77 ~ P75 管脚在 P7DDR 置 1 时为输出端口, 清 0 时为输入端口。
4	P74DDR	0	W	(H8S/2218 群) P7DDR 置 1 时对应的端口 7 的各管脚为输出管脚, 清 0 时为输入管脚。 (H8S/2212 群) 保留位 该位未定义且不能被修改。
3、2	—	不定	—	保留位 该位未定义且不能被修改。
1	P71DDR	0	W	(H8S/2218 群)
0	P70DDR	0	W	P7DDR 置 1 时对应端口 7 的各管脚为输出管脚, 清 0 时为输入管脚。 (H8S/2212 群) 保留位 该位未定义且不能被修改。

8.4.2 端口 7 数据寄存器 (P7DR)

P7DR 保存端口 7 的各管脚的输出数据。

位	位名	初始值	R/W	说 明
7	P77DR	0	R/W	(H8S/2218 群)
6	P76DR	0	R/W	保留位
5	P75DR	0	R/W	该位未定义且不能被修改。 (H8S/2212 群) 保存端口 7 各管脚的输出数据。
4	P74DR	0	R/W	(H8S/2218 群) 保存端口 7 各管脚的输出数据。 (H8S/2212 群) 保留位 该位未定义且不能被修改。
3、2	—	不定	—	保留位 该位未定义且不能被修改。
1	P71DD	0	R/W	(H8S/2218 群)
0	P70DD	0	R/W	保存端口 7 各管脚的输出数据。 (H8S/2212 群) 保留位 该位未定义且不能被修改。

8.4.3 端口 7 寄存器 (PORT7)

PORT7 表示端口 7 管脚的状态。

位	位名	初始值	R/W	说 明
7 6 5	P77 P76 P75	— * — * — *	— — —	(H8S/2218 群) 保留位 该位未定义且不能被修改。 (H8S/2212 群) P7DDR 置 1, 读取 P7DR 的值。 P7DDR 清 0, 读取管脚的状态。
4	P74	— *	R	(H8S/2218 群) P7DDR 置 1 时, 读取端口 7, 则读取 P7DR 的值。P7DDR 清 0 时, 读取端口 7 则读取管脚的状态。 (H8S/2212 群) 保留位 该位未定义且不能被修改。
3、2	—	不定	—	保留位 该位未定义且不能被修改。
1 0	P71 P70	— * — *	R R	(H8S/2218 群) P7DDR 置 1 时, 读取端口 7, 则读取 P7DR 的值。P7DDR 清 0 时, 读取端口 7 则读取管脚的状态。 (H8S/2212 群) 保留位 该位未定义且不能被修改。

【注】 * 根据 P77、P76、P75、P74、P71、P70 管脚的状态决定。

8.4.4 管脚功能

8.4.4.1 H8S/2218 群的管脚功能

端口 7 兼用总线控制输出，手动置位输入。寄存器的设定值与管脚的功能的关系如下所示。

表 8.23 P74 的管脚功能

MRESE	0		1
P74DDR	0	1	—
管脚功能	P74 输入管脚	P74 输出管脚	$\overline{\text{MRES}}$ 输入管脚

表 8.24 P71 的管脚功能

工作模式	模式 4 ~ 6		模式 7	
P71DDR	0	1	0	1
管脚功能	P71 输入管脚	$\overline{\text{CS5}}$ 输出管脚	P71 输入管脚	P71 输出管脚

表 8.25 P70 的管脚功能

工作模式	模式 4 ~ 6		模式 7	
P70DDR	0	1	0	1
管脚功能	P70 输入管脚	$\overline{\text{CS4}}$ 输出管脚	P70 输入管脚	P70 输出管脚

8.4.4.2 H8S/2212 群的管脚功能

端口 7 与 H-UDI 管脚兼用。寄存器的设定值与管脚功能的关系如下所示。

表 8.26 P77 的管脚功能

EMLE	0		1
P77DDR	0	1	—
管脚功能	P77 输入管脚	P77 输出管脚	TDO 输出管脚

表 8.27 P76 的管脚功能

EMLE	0		1
P76DDR	0	1	—
管脚功能	P76 输入管脚	P76 输出管脚	TCK 输入管脚

表 8.28 P75 的管脚功能

EMLE	0		1
P75DDR	0	1	—
管脚功能	P75 输入管脚	P75 输出管脚	TMS 输入管脚

8.5 端口 9

端口 9 是兼用 A/D 转换器模拟输入管脚的 2 位输入端口。H8S/2218 群的端口 9 和 H8S/2212 群的端口 9 具有相同功能。

- 端口 9 寄存器 (PORT9)

8.5.1 端口 9 寄存器 (PORT9)

PORT9 表示端口 9 的管脚状态。

位	位名	初始值	R/W	说 明
7	P97	— *	R	读此寄存器后通常读出管脚的状态。
6	P96	— *	R	
5~0	—	不定	—	保留位 读后读出不定值。

【注】 * 根据 P97、P96 管脚的状态决定。

8.5.2 管脚功能

端口 9 兼用 A/D 转换器的模拟输入 (AN15、AN14)。

8.6 端口 A

H8S/2218 群的端口 A 是地址总线（A19 ~ A16）输出管脚、SCI 输入 / 输出管脚兼用的 4 位输入 / 输出端口。H8S/2212 群的端口 A 是兼用 SCI 输入 / 输出管脚的 3 位输入 / 输出端口。端口 A 有以下寄存器。

- 端口 A 数据方向寄存器（PADDR）
- 端口 A 数据寄存器（PADR）
- 端口 A 寄存器（PORTA）
- 端口 A 上拉 MOS 控制寄存器（PAPCR）
- 端口 A 漏极开路控制寄存器（PAODR）

8.6.1 端口 A 数据方向寄存器（PADDR）

PADDR 是以每位指定端口 A 管脚的输入 / 输出。

本寄存器是写专用寄存器，所以不能通过位操作指令进行写入。

详细请参考 2.9.4 包括写专用位的寄存器存取方法。

位	位名	初始值	R/W	说 明
7~4	—	不定	—	读此寄存器后通常读出管脚的状态。
3	PA3DDR	0	W	（H8S/2218 群） 模式 4 ~ 6 时 用 PFCR 的 AE3 ~ AE0 位允许地址输出时，对应端口 A 的各管脚为地址输出管脚。禁止地址输出时，将 PADDR 值 1，对应端口 A 的各管脚为输出端口。清 0 则为输入端口。 模式 7 时 将此位置 1，对应管脚为输出端口，清 0 则为输入端口。 （H8S/2212 群） 将此位置 1，对应管脚为输出端口，清 0 则为输入端口。
2	PA2DDR	0	W	
1	PA1DDR	0	W	
0	PA0DDR*	0	W	

【注】 * H8S/2212 群是保留位。读后读出不定值。写定义为无效。

8.6.2 端口 A 数据寄存器（PADR）

PADR 保存端口 A 各管脚的输出数据。

位	位名	初始值	R/W	说 明
7~4	—	不定	—	保留位 读后读出不定值。写定义为无效。
3	PA3DR	0	R/W	给管脚保存输出数据以作为通用输出端口使用。
2	PA2DR	0	R/W	
1	PA1DR	0	R/W	
0	PA0DR*	0	R/W	

【注】 * H8S/2212 群是保留位。读后读出不定值。写定义为无效。

8.6.3 端口 A 寄存器 (PORTA)

PORTA 反映端口 A 的管脚的状态。

位	位名	初始值	R/W	说 明
7~4	—	不定	—	保留位 该位未定义。
3	PA3	— *1	R	将 PADDR 置 1，读取端口 A 后读取的是 PADDR 的值。PADDR 请 0 时，读端口 A 后读出管脚的状态。
2	PA2	— *1	R	
1	PA1	— *1	R	
0	PA0*2	— *1	R	

【注】 *1 根据 PA3 ~ PA0 管脚的状态决定。

*2 H8S/2212 群是保留位。读后读出不定值。

8.6.4 端口 A 上拉 MOS 控制寄存器 (PAPCR)

PAPCR 控制端口 A 的输入上拉 MOS 的 ON/OFF。对于端口输入、SCI 输入管脚有效。

位	位名	初始值	R/W	说 明
7~4	—	不定	—	保留位 该位未定义且不能被修改。
3	PA3PCR	0	R/W	管脚在输入状态时，寄存器对应位置 1 则为该管脚开启输入上拉 MOS 设置为 ON。
2	PA2PCR	0	R/W	
1	PA1PCR	0	R/W	
0	PA0PCR*	0	R/W	

【注】 * H8S/2212 群是保留位。读取该位读出未定义值且该位不能被修改。

8.6.5 端口 A 漏极开路控制寄存器 (PAODR)

PAODR 规定端口 A 的输出类型。PAODR 对于端口输出、SCI 输出管脚有效。

位	位名	初始值	R/W	说 明
7~4	—	不定	—	保留位 该位未定义且不能被修改。
3	PA3ODR	0	R/W	将 PAODR 置 1 端口 A 的各管脚为 NMOS 漏极开路输出，清 0 为 CMOS 输出。
2	PA2ODR	0	R/W	
1	PA1ODR	0	R/W	
0	PA0ODR*	0	R/W	

【注】 * H8S/2212 群为保留位。读取该位读出未定义值且该位不能被修改。

8.6.6 管脚功能

8.6.6.1 H8S/2218 群的管脚功能

端口 A 管脚兼用地址总线 (A19 ~ A16) 输出管脚和 SCI_2 输入 / 输出管脚。寄存器的设定值和管脚功能的关系如下所示。

表 8.29 PA3 的管脚功能

工作模式	模式 4 ~ 6						模式 7				
AE3 ~ AE0	B'11xx	(B'11xx) 以外					—				
SCR_2 的 CK1	—	0			1		0			1	
SMR_2 的 C/A	—	0		1		—		0		1	
SCR_2 的 CK0	—	0		1		—		0		1	
PA3DDR	—	0	1	—	—	—	0	1	—	—	—
管脚功能	A19 输出 管脚	PA3 输入 管脚	PA3 输出 管脚	SCK2 输出 管脚	SCK2 输出 管脚	SCK2 输入 管脚	PA3 输入 管脚	PA3 输出 管脚	SCK2 输出 管脚	SCK2 输出 管脚	SCK2 输入 管脚

表 8.30 PA2 的管脚功能

工作模式	模式 4 ~ 6				模式 7		
AE3 ~ AE0	B'1011 或 B'11xx	(B'1011 或 B'11xx) 以外			—		
SCR_2 的 RE	—	0		1	0		1
PA2DDR	—	0	1	—	0	1	—
管脚功能	A18 输出管脚	PA2 输入管脚	PA2 输出管脚	RxD2 输入管脚	PA2 输入管脚	PA2 输出管脚	RxD2 输入管脚

【符号说明】

×: Don't care

表 8.31 PA1 的管脚功能

工作模式	模式 4 ~ 6				模式 7		
AE3 ~ AE0	B'101x 或 B'11xx	(B'101x 或 B'11xx) 以外			—		
SCR_2 的 TE	—	0		1	0		1
PA1DDR	—	0	1	—	0	1	—
管脚功能	A17 输出管脚	PA1 输入管脚	PA1 输出管脚	TxD2 输出管脚	PA1 输入管脚	PA1 输出管脚	TxD2 输出管脚

表 8.32 PA0 的管脚功能

工作模式	模式 4 ~ 6			模式 7	
AE3 ~ AE0	(B'0xxx 或 B'1000) 以外	B'0xxx 或 B'1000		—	
PA0DDR	—	0	1	0	1
管脚功能	A16 输出管脚	PA0 输入管脚	PA0 输出管脚	PA0 输入管脚	PA0 输出管脚

【符号说明】

X: Don't care

8.6.6.2 H8S/2212 群的管脚功能

端口 A 管脚兼用 SCI_2 输入 / 输出管脚。寄存器的设定值和管脚功能的关系如下所示。

表 8.33 PA3 的管脚功能

SCR_2 的 CKE1	0			1	
SMR_2 的 C/A	0		1	—	
SCR_2 的 CKE0	0		1	—	
PA3DDR	0	1	—	—	—
管脚功能	PA3 输入管脚	PA3 输出管脚	SCK2 输出管脚	SCK2 输出管脚	SCK2 输入管脚

表 8.34 PA2 的管脚功能

SCR_2 的 RE	0		1		
PA2DDR	0	1	—		
管脚功能	PA2 输入管脚		PA2 输出管脚		RxD2 输入管脚

表 8.35 PA1 的管脚功能

SCR_2 的 TE	0		1		
PA1DDR	0	1	—		
管脚功能	PA1 输入管脚		PA1 输出管脚		TxD2 输出管脚

8.6.7 端口 A 输入上拉 MOS 的状态

端口 A 内置可通过程序控制的输入上拉 MOS。输入上拉 MOS 通过每个位可指定 ON/OFF 状态。
输入上拉 MOS 的状态如表 8.36 所示。

表 8.36 端口 A 输入上拉 MOS 的状态

管脚	加电复位	硬件待机模式	手动复位	软件待机模式	其它工作时
地址输出、端口输出、 SCI 输出	OFF		OFF		
端口输入、SCI 输入			ON/OFF		

【符号说明】

OFF : 输入上拉 MOS 通常为 OFF 状态。

ON/OFF: PADDR = 0 且 PAPCR = 1 时为 ON 状态, 其它为 OFF 状态。

8.7 端口 B（仅 H8S/2218 群）

端口 B 是兼用地址总线（A15 ~ A8）输出管脚的 8 位输入 / 输出端口。端口 B 有以下寄存器。

【注】 使用 E6000 仿真程序的 USB 时，为从 PB1 和 PB0 输出地址 A9、A8，请设定 PFCR 的 AE3 ~ AE0 位。这一点适用于 H8S/2218 群、H8S/2212 群。

- 端口 B 数据方向寄存器（PBDDR）
- 端口 B 数据寄存器（PBDR）
- 端口 B 寄存器（PORTB）
- 端口 B 上拉 MOS 控制寄存器（PBPCR）

8.7.1 端口 B 数据方向寄存器（PBDDR）

PBDDR 是以每位指定端口 B 各管脚的输入 / 输出。

本寄存器是写专用寄存器，所以不能通过位操作指令写入。

详细请参考 2.9.4 包括写专用位的寄存器存取方法。

位	位名	初始值	R/W	说明
7	PB7DDR	0	W	模式 4 ~ 6 时 通过 PFCR 的 AE3 ~ AE0 位允许地址输出时，对应的端口 B 各管脚为地址输出管脚。禁止地址输出时，若 PBDDR 置 1，对应端口 B 各管脚为输出端口，清 0 则为输入端口。
6	PB6DDR	0	W	
5	PB5DDR	0	W	
4	PB4DDR	0	W	
3	PB3DDR	0	W	
2	PB2DDR	0	W	
1	PB1DDR	0	W	
0	PB0DDR	0	W	模式 7 时 PBDDR 位置 1，对应端口 B 的管脚为输出端口，清 0 则为输入端口。

8.7.2 端口 B 数据寄存器（PBDR）

PBDR 保存端口 B 各管脚的输出数据。

位	位名	初始值	R/W	说明
7	PB7DR	0	R/W	给管脚保存输出数据以作为通用输出端口使用。
6	PB6DR	0	R/W	
5	PB5DR	0	R/W	
4	PB4DR	0	R/W	
3	PB3DR	0	R/W	
2	PB2DR	0	R/W	
1	PB1DR	0	R/W	
0	PB0DR	0	R/W	

8.7.3 端口 B 寄存器 (PORTB)

PORTB 表示端口 B 管脚的状态。

位	位名	初始值	R/W	说明
7	PB7	— *	R	PBDDR 的位置 1, 读端口 B 时读取 PBDR 的值。 PBDDR 清 0 时, 读取端口 B, 则读取管脚的状态。
6	PB6	— *	R	
5	PB5	— *	R	
4	PB4	— *	R	
3	PB3	— *	R	
2	PB2	— *	R	
1	PB1	— *	R	
0	PB0	— *	R	

【注】 * 根据 PB7 ~ PB0 管脚的状态决定。

8.7.4 端口 B 上拉 MOS 控制寄存器 (PBPCR)

PBPCR 控制端口 B 的输入上拉 MOS 的 ON/OFF 状态。PBPCR 对于端口输入管脚有效。

位	位名	初始值	R/W	说明
7	PB7PCR	0	R/W	管脚在输入状态时, 对应寄存器的位置 1, 则开启该管脚的输入上拉 MOS。
6	PB6PCR	0	R/W	
5	PB5PCR	0	R/W	
4	PB4PCR	0	R/W	
3	PB3PCR	0	R/W	
2	PB2PCR	0	R/W	
1	PB1PCR	0	R/W	
0	PB0PCR	0	R/W	

8.7.5 管脚功能

端口 B 管脚兼用地址总线（A15 ~ A9）输出管脚。寄存器的设定值和管脚功能的关系如下所示。

【注】 使用 E6000 仿真程序的 USB 时，请将 A9、A8 设定为地址总线输出管脚。

表 8.37 PB7 的管脚状态

工作模式	模式 4 ~ 6			模式 7	
AE3 ~ AE0	B'1xxx	(B'1xxx) 以外		—	
PA7DDR	—	0	1	0	1
管脚功能	A15 输出管脚	PB7 输入管脚	PB7 输出管脚	PB7 输入管脚	PB7 输出管脚

表 8.38 PB6 管脚功能

工作模式	模式 4 ~ 6			模式 7	
AE3 ~ AE0	B'0111 或 B'1xxx	(B'0111 或 B'1xxx) 以外		—	
PB6DDR	—	0	1	0	1
管脚功能	A14 输出管脚	PB6 输入管脚	PB6 输出管脚	PB6 输入管脚	PB6 输出管脚

表 8.39 PB5 管脚功能

工作模式	模式 4 ~ 6			模式 7	
AE3 ~ AE0	B'011x 或 B'1xxx	(B'011x 或 B'1xxx) 以外		—	
PB5DDR	—	0	1	0	1
管脚功能	A13 输出管脚	PB5 输入管脚	PB5 输出管脚	PB5 输入管脚	PB5 输出管脚

表 8.40 PB4 的管脚功能

工作模式	模式 4 ~ 6			模式 7	
AE3 ~ AE0	(B'0100 或 B'00xx) 以外	B'0100 或 B'00xx		—	
PB4DDR	—	0	1	0	1
管脚功能	A12 输出管脚	PB4 输入管脚	PB4 输出管脚	PB4 输入管脚	PB4 输出管脚

表 8.41 PB3 的管脚状态

工作模式	模式 4 ~ 6			模式 7	
AE3 ~ AE0	(B'00xx) 以外	B'00xx		—	
PB3DDR	—	0	1	0	1
管脚功能	A11 输出管脚	PB3 输入管脚	PB3 输出管脚	PB3 输入管脚	PB3 输出管脚

【符号说明】

×: Don't care

表 8.42 PB2 的管脚功能

工作模式	模式 4 ~ 6			模式 7	
AE3 ~ AE0	(B'0010 或 B'000x) 以外	B'0010 或 B'000x		—	
PB2DDR	—	0	1	0	1
管脚功能	A10 输出管脚	PB2 输入管脚	PB2 输出管脚	PB2 输入管脚	PB2 输出管脚

表 8.43 PB1 的管脚功能

工作模式	模式 4 ~ 6			模式 7	
AE3 ~ AE0	(B'000x) 以外	B'000x		—	
PB1DDR	—	0	1	0	1
管脚功能	A9 输出管脚	PB1 输入管脚	PB1 输出管脚	PB1 输入管脚	PB1 输出管脚

表 8.44 PB0 的管脚功能

工作模式	模式 4 ~ 6			模式 7	
AE3 ~ AE0	(B'0000) 以外	B'0000		—	
PB0DDR	—	0	1	0	1
管脚功能	A8 输出管脚	PB0 输入管脚	PB0 输出管脚	PB0 输入管脚	PB0 输出管脚

【符号说明】

×: Don't care

8.7.6 端口 B 输入上拉 MOS 的状态

端口 B 内置可通过程序控制的输入上拉 MOS。输入上拉 MOS 可用每个位指定 ON/OFF 状态。

输入上拉 MOS 的状态如表 8.45 所示。

表 8.45 端口 B 输入上拉 MOS 的状态

管脚	加电复位	硬件待机模式	手动复位	软件待机模式	其它工作时
地址输出、端口输出、	OFF		OFF		
端口输入	ON/OFF				

【符号说明】

OFF : 输入上拉 MOS 通常为 Off 状态。

ON/OFF : PBDDR = 0 且 PBPCR = 1 时为 ON 状态, 其它为 OFF 状态。

8.8 端口 C（仅 H8S/2218 群）

端口 C 是兼用地址总线（A7 ~ A0）输出管脚的 8 位的输入 / 输出端口。端口 C 有以下寄存器。

【注】 使用 E6000 仿真程序的 RTC 和 USB 时，为从 PC7 ~ PC0 输出地址 A7 ~ A0，请设定 PCDDR。这一点适用于 H8S/2218 群、H8S/2212 群。

- 端口 C 数据方向寄存器（PCDDR）
- 端口 C 数据寄存器（PCDR）
- 端口 C 寄存器（PORTC）
- 端口 C 上拉 MOS 控制寄存器（PCPCR）

8.8.1 端口 C 数据方向寄存器（PCDDR）

PCDDR 是以每位指定端口 C 的输入 / 输出。

本寄存器是写专用寄存器所以不能通过位操作指令写入。

详细请参考 2.9.4 包括写专用位的寄存器存取方法。

位	位名	初始值	R/W	说明
7	PC7DDR	0	W	模式 4、5 时
6	PC6DDR	0	W	对应端口 C 管脚为地址输出管脚。
5	PC5DDR	0	W	模式 6 时
4	PC4DDR	0	W	此位置 1，对应端口 C 管脚为地址输出管脚，清 0 则为
3	PC3DDR	0	W	输入端口。
2	PC2DDR	0	W	模式 7 时
1	PC1DDR	0	W	此位置 1，对应端口 C 管脚为地址输出管脚，清 0 则为
0	PC0DDR	0	W	输入端口。

8.8.2 端口 C 数据寄存器（PCDR）

PCDR 保存端口 C 的输出数据。

位	位名	初始值	R/W	说明
7	PC7DR	0	R/W	给管脚保存输出数据以作为通用输出端口使用。
6	PC6DR	0	R/W	
5	PC5DR	0	R/W	
4	PC4DR	0	R/W	
3	PC3DR	0	R/W	
2	PC2DR	0	R/W	
1	PC1DR	0	R/W	
0	PC0DR	0	R/W	

8.8.3 端口 C 寄存器 (PORTC)

PORTC 表示端口 C 管脚的状态。

位	位名	初始值	R/W	说明
7	PC7	— *	R	PCDDR 的位置 1, 读端口 C, 读取 PCDR 的值。 PCDDR 清 0, 读端口 C, 读取管脚的状态。
6	PC6	— *	R	
5	PC5	— *	R	
4	PC4	— *	R	
3	PC3	— *	R	
2	PC2	— *	R	
1	PC1	— *	R	
0	PC0	— *	R	

【注】 * 根据 PC7 ~ PC0 管脚的状态决定。

8.8.4 端口 C 上拉 MOS 控制寄存器 (PCPCR)

PCPCR 控制端口 C 的输入上拉 MOS 的 ON/OFF 状态。

位	位名	初始值	R/W	说明
7	PC7PCR	0	R/W	管脚在输入状态时, 将此寄存器的位置 1, 则开启管脚的输入上拉 MOS。
6	PC6PCR	0	R/W	
5	PC5PCR	0	R/W	
4	PC4PCR	0	R/W	
3	PC3PCR	0	R/W	
2	PC2PCR	0	R/W	
1	PC1PCR	0	R/W	
0	PC0PCR	0	R/W	

8.8.5 端口功能

端口 C 兼用地址总线 (A7 ~ A0) 输出管脚。寄存器的设定值和管脚功能的关系如下所示。

【注】 使用带有 E6000 仿真程序的 RTC 和 USB 时, 请将地址 A7 ~ A0 设定为地址总线输出管脚。

表 8.46 PC7 的管脚功能

工作模式	模式 4、5	模式 6		模式 7	
PC7DDR	—	0	1	0	1
管脚功能	A7 输出管脚	PC7 输入管脚	A7 输出管脚	PC7 输入管脚	PC7 输出管脚

表 8.47 PC6 的管脚功能

工作模式	模式 4、5	模式 6		模式 7	
PC6DDR	—	0	1	0	1
管脚功能	A6 输出管脚	PC6 输入管脚	A6 输出管脚	PC6 输入管脚	PC6 输出管脚

表 8.48 PC5 的管脚功能

工作模式	模式 4、5	模式 6		模式 7	
PC5DDR	—	0	1	0	1
管脚功能	A5 输出管脚	PC5 输入管脚	A5 输出管脚	PC5 输入管脚	PC5 输出管脚

表 8.49 PC4 的管脚功能

工作模式	模式 4、5	模式 6		模式 7	
PC4DDR	—	0	1	0	1
管脚功能	A4 输出管脚	PC4 输入管脚	A4 输出管脚	PC4 输入管脚	PC4 输出管脚

表 8.50 PC3 的管脚功能

工作模式	模式 4、5	模式 6		模式 7	
PC3DDR	—	0	1	0	1
管脚功能	A3 输出管脚	PC3 输入管脚	A3 输出管脚	PC3 输入管脚	PC3 输出管脚

表 8.51 PC2 的管脚功能

工作模式	模式 4、5	模式 6		模式 7	
PC2DDR	—	0	1	0	1
管脚功能	A2 输出管脚	PC2 输入管脚	A2 输出管脚	PC2 输入管脚	PC2 输出管脚

表 8.52 PC1 的管脚功能

工作模式	模式 4、5	模式 6		模式 7	
PC1DDR	—	0	1	0	1
管脚功能	A1 输出管脚	PC1 输入管脚	A1 输出管脚	PC1 输入管脚	PC1 输出管脚

表 8.53 PC0 的管脚功能

工作模式	模式 4、5	模式 6		模式 7	
PC0DDR	—	0	1	0	1
管脚功能	A0 输出管脚	PC0 输入管脚	A0 输出管脚	PC0 输入管脚	PC0 输出管脚

8.8.6 端口 C 输入上拉 MOS 的状态

端口 C 内置可用程序控制的输入上拉 MOS。此输入上拉 MOS 可用于模式 6、7，通过每个位指定 ON/OFF 状态。

输入上拉 MOS 的状态如表 8.54 所示。

表 8.54 端口 C 输入上拉 MOS 的状态

管脚	加电复位	硬件待机模式	手动复位	软件待机模式	其它工作时
地址输出 (模式 4、5)	OFF		OFF		
端口输出 (模式 6、7)					
端口输入 (模式 6、7)			ON/OFF		

【符号说明】

OFF : 输入上拉 MOS 通常是 OFF 状态。

ON/OFF : PCDDR = 0 且 PCPCR = 1 时为 ON 状态，否则为 OFF 状态。

8.9 端口 D（仅 H8S/2218 群）

端口 D 是兼用数据总线（D15 ~ D8）输入 / 输出管脚的 8 位输入 / 输出端口。端口 D 有以下寄存器。

- 端口 D 数据方向寄存器（PDDDR）
- 端口 D 数据寄存器（PDDR）
- 端口 D 寄存器（PORTD）
- 端口 D 上拉 MOS 控制寄存器（PDPCR）

8.9.1 端口 D 数据方向寄存器（PDDDR）

PDDDR 是以每位指定端口 D 管脚的输入 / 输出。

本寄存器是写专用寄存器，所以不能用位操作指令来写。

详细请参考 2.9.4 包括写专用位的寄存器存取方法。

位	位名	初始值	R/W	说明
7	PD7DDR	0	W	模式 4 ~ 6 忽视用 PDDDR 的输入 / 输出方向，端口 D 自动成为数据输入 / 输出管脚。
6	PD6DDR	0	W	
5	PD5DDR	0	W	
4	PD4DDR	0	W	模式 7 此位置 1，对应端口 D 的管脚为输出端口，清 0 则为输入端口。
3	PD3DDR	0	W	
2	PD2DDR	0	W	
1	PD1DDR	0	W	
0	PD0DDR	0	W	

8.9.2 端口 D 数据寄存器（PDDR）

PDDR 保存端口 D 的输出数据。

位	位名	初始值	R/W	说明
7	PD7DR	0	R/W	给管脚保存输出数据以作为通用输出端口使用。
6	PD6DR	0	R/W	
5	PD5DR	0	R/W	
4	PD4DR	0	R/W	
3	PD3DR	0	R/W	
2	PD2DR	0	R/W	
1	PD1DR	0	R/W	
0	PD0DR	0	R/W	

8.9.3 端口 D 寄存器 (PORTD)

PORTD 表示端口 D 管脚的状态。

位	位名	初始值	R/W	说明
7	PD7	— *	R	PDDDR 的位置 1, 读端口 D 则读取 PDDDR 的值。 PDDDR 清 0, 读端口 D 则读取管脚的状态。 【注】 存取 EXMDLSTP 或 RTC 寄存器 (地址范围 H'FFFF40 ~ H'FFFF5F) 后, 对 H'FFFF40 ~ H'FFFF5F 范围以外的外部地址空间 (如 H'FFEF00 ~ H'FFF7FF 等) 进行一次虚读后, 再读 PORTD。
6	PD6	— *	R	
5	PD5	— *	R	
4	PD4	— *	R	
3	PD3	— *	R	
2	PD2	— *	R	
1	PD1	— *	R	
0	PD0	— *	R	

【注】 * 根据 PD7 ~ PD0 管脚状态决定。

8.9.4 端口 D 上拉 MOS 控制寄存器 (PDPCR)

PDPCR 控制端口 D 输入上拉 MOS 的 ON/OFF。

位	位名	初始值	R/W	说明
7	PD7PCR	0	R/W	管脚在输入状态时, 此寄存器的位置 1, 则开启该管脚的输入上拉 MOS。
6	PD6PCR	0	R/W	
5	PD5PCR	0	R/W	
4	PD4PCR	0	R/W	
3	PD3PCR	0	R/W	
2	PD2PCR	0	R/W	
1	PD1PCR	0	R/W	
0	PD0PCR	0	R/W	

8.9.5 管脚功能

端口 D 兼用数据总线 (D15 ~ D8) 输入 / 输出管脚。
寄存器的设定值和管脚功能的关系如下所示。

表 8.55 PD7 的管脚功能

工作模式	模式 4 ~ 6	模式 7	
PD7DDR	—	0	1
管脚功能	D15 输入 / 输出管脚	PD7 输入管脚	PD7 输出管脚

表 8.56 PD6 的管脚功能

工作模式	模式 4 ~ 6	模式 7	
PD6DDR	—	0	1
管脚功能	D14 输入 / 输出管脚	PD6 输入管脚	PD6 输出管脚

表 8.57 PD5 的管脚功能

工作模式	模式 4 ~ 6	模式 7	
PD5DDR	—	0	1
管脚功能	D13 输入 / 输出管脚	PD5 输入管脚	PD5 输出管脚

表 8.58 PD4 的管脚功能

工作模式	模式 4 ~ 6	模式 7	
PD4DDR	—	0	1
管脚功能	D12 输入 / 输出管脚	PD4 输入管脚	PD4 输出管脚

表 8.59 PD3 的管脚功能

工作模式	模式 4 ~ 6	模式 7	
PD3DDR	—	0	1
管脚功能	D11 输入 / 输出管脚	PD3 输入管脚	PD3 输出管脚

表 8.60 PD2 的管脚功能

工作模式	模式 4 ~ 6	模式 7	
PD2DDR	—	0	1
管脚功能	D10 输入 / 输出管脚	PD2 输入管脚	PD2 输出管脚

表 8.61 PD1 的管脚功能

工作模式	模式 4 ~ 6	模式 7	
PD1DDR	—	0	1
管脚功能	D9 输入 / 输出管脚	PD1 输入管脚	PD1 输出管脚

表 8.62 PD0 的管脚功能

工作模式	模式 4 ~ 6	模式 7	
PD0DDR	—	0	1
管脚功能	D8 输入 / 输出管脚	PD0 输入管脚	PD0 输出管脚

8.9.6 端口 D 输入上拉 MOS 的状态

端口 D 内置可用程序控制的输入上拉 MOS。此输入上拉 MOS 可用于模式 7，通过每个位指定 ON/OFF 状态。

输入上拉 MOS 的状态如表 8.63 所示。

表 8.63 端口 D 输入上拉的状态

管脚	加电复位	硬件待机模式	手动复位	软件待机模式	其它工作时
数据输入 / 输出 (模式 4 ~ 6)	OFF		OFF		
端口输出 (模式 7)					
端口输入 (模式 7)			ON/OFF		

【符号说明】

OFF : 输入上拉 MOS 通常为 OFF 状态。

ON/OFF : PDDDR = 0 且 PDPCR = 1 时为 ON 状态，其它为 OFF 状态。

8.10 端口 E

端口 E 是兼用数据总线（D7 ~ D0）输入 / 输出管脚的 8 位输入 / 输出端口。端口 E 有以下寄存器。

- 端口 E 数据方向寄存器（PEDDR）
- 端口 E 数据寄存器（PEDR）
- 端口 E 寄存器（PORTE）
- 端口 E 上拉 MOS 控制寄存器（PEPCR）

8.10.1 端口 E 数据方向寄存器（PEDDR）

PEDDR 是以每位指定端口 E 各管脚的输入 / 输出。

本寄存器是写专用寄存器，所以不能通过位操作指令写入。

详细请参考 2.9.4 包括写专用位的寄存器存取方法。

位	位名	初始值	R/W	说明
7	PE7DDR	0	W	(H8S/2218 群) 模式 4 ~ 6 时 选择 8 位总线模式时，端口 E 作为输入 / 输出端口使用。 将 PEDDR 置 1，对应端口 E 的各管脚为输出端口，清 0 则为输入端口。 选择 16 位总线模式时，可忽视 PEDDR 输入 / 输出的方向设定，端口 E 管脚自动成为数据输入 / 输出管脚。 关于 8 位 / 16 位总线模式请参考 6. 总线控制器（BSC）。 模式 7 时： PEDDR 置 1，对应端口 E 的各管脚为输出端口，清 0 则为输入端口。 (H8S/2212 群) PEDDR 置 1，对应端口 E 的各管脚为输出端口，清 0 则为输入端口。
6	PE6DDR	0	W	
5	PE5DDR	0	W	
4	PE4DDR	0	W	
3	PE3DDR	0	W	
2	PE2DDR	0	W	
1	PE1DDR	0	W	
0	PE0DDR	0	W	

8.10.2 端口 E 数据寄存器（PEDR）

PEDR 保存端口 E 的各管脚的输出数据。

位	位名	初始值	R/W	说明
7	PE7DR	0	R/W	给管脚保存输出数据以作为通用输出端口使用。
6	PE6DR	0	R/W	
5	PE5DR	0	R/W	
4	PE4DR	0	R/W	
3	PE3DR	0	R/W	
2	PE2DR	0	R/W	
1	PE1DR	0	R/W	
0	PE0DR	0	R/W	

8.10.3 端口 E 寄存器 (PORTE)

PORTE 表示端口 E 管脚的状态。

位	位名	初始值	R/W	说明
7	PE7	— *	R	PEDDR 的位置 1, 读端口 E, 读取 PEDR 的值。 PEDDR 清 0 时, 读端口 E 则读取管脚的状态。
6	PE6	— *	R	
5	PE5	— *	R	
4	PE4	— *	R	
3	PE3	— *	R	
2	PE2	— *	R	
1	PE1	— *	R	
0	PE0	— *	R	

【注】 * 根据 PE7 ~ PE0 管脚的状态决定。

8.10.4 端口 E 上拉 MOS 控制寄存器 (PEPCR)

PEPCR 控制端口 E 输入上拉 MOS 的 ON/OFF。

位	位名	初始值	R/W	说明
7	PE7PCR	0	R/W	管脚为输入状态时, 此寄存器的位置 1 则开启该管脚的 输入上拉 MOS。
6	PE6PCR	0	R/W	
5	PE5PCR	0	R/W	
4	PE4PCR	0	R/W	
3	PE3PCR	0	R/W	
2	PE2PCR	0	R/W	
1	PE1PCR	0	R/W	
0	PE0PCR	0	R/W	

8.10.5 管脚功能

8.10.5.1 H8S/2218 群的管脚功能

端口 E 兼用数据总线 (D7 ~ D0) 输入 / 输出管脚。寄存器的设定值和管脚功能关系如下所示。

表 8.64 PE7 的管脚功能

工作模式	模式 4 ~ 6			模式 7	
总线模式	8 位总线模式		16 位总线模式	—	
PE7DDR	0	1	—	0	1
管脚功能	PE7 输入管脚	PE7 输出管脚	D7 输入 / 输出管脚	PE7 输入管脚	PE7 输出管脚

表 8.65 PE6 的管脚功能

工作模式	模式 4 ~ 6			模式 7	
总线模式	8 位总线模式		16 位总线模式	—	
PE6DDR	0	1	—	0	1
管脚功能	PE6 输入管脚	PE6 输出管脚	D6 输入 / 输出管脚	PE6 输入管脚	PE6 输出管脚

表 8.66 PE5 的管脚功能

工作模式	模式 4 ~ 6			模式 7	
总线模式	8 位总线模式		16 位总线模式	—	
PE5DDR	0	1	—	0	1
管脚功能	PE5 输入管脚	PE5 输出管脚	D5 输入 / 输出管脚	PE5 输入管脚	PE5 输出管脚

表 8.67 PE4 的管脚功能

工作模式	模式 4 ~ 6			模式 7	
总线模式	8 位总线模式		16 位总线模式	—	
PE4DDR	0	1	—	0	1
管脚功能	PE4 输入管脚	PE4 输出管脚	D4 输入 / 输出管脚	PE4 输入管脚	PE4 输出管脚

表 8.68 PE3 的管脚功能

工作模式	模式 4 ~ 6			模式 7	
总线模式	8 位总线模式		16 位总线模式	—	
PE3DDR	0	1	—	0	1
管脚功能	PE3 输入管脚	PE3 输出管脚	D3 输入 / 输出管脚	PE3 输入管脚	PE3 输出管脚

表 8.69 PE2 的管脚功能

工作模式	模式 4 ~ 6			模式 7	
总线模式	8 位总线模式		16 位总线模式	—	
PE2DDR	0	1	—	0	1
管脚功能	PE2 输入管脚	PE2 输出管脚	D2 输入 / 输出管脚	PE2 输入管脚	PE2 输出管脚

表 8.70 PE1 的管脚功能

工作模式	模式 4 ~ 6			模式 7	
总线模式	8 位总线模式		16 位总线模式	—	
PE1DDR	0	1	—	0	1
管脚功能	PE1 输入管脚	PE1 输出管脚	D1 输入 / 输出管脚	PE1 输入管脚	PE1 输出管脚

表 8.71 PE0 的管脚功能

工作模式	模式 4 ~ 6			模式 7	
总线模式	8 位总线模式		16 位总线模式	—	
PE0DDR	0	1	—	0	1
管脚功能	PE0 输入管脚	PE0 输出管脚	D0 输入 / 输出管脚	PE0 输入管脚	PE0 输出管脚

8.10.5.2 H8S/2212 群的管脚功能

在 H8S/2212 群将端口 E 作为通用输入 / 输出端口。寄存器的设定值和端口功能的关系如下所示。

表 8.72 PE7 的管脚状态

PE7DDR	0	1
管脚功能	PE7 输入管脚	PE7 输出管脚

表 8.73 PE6 的管脚功能

PE6DDR	0	1
管脚功能	PE6 输入管脚	PE6 输出管脚

表 8.74 PE5 的管脚功能

PE5DDR	0	1
管脚功能	PE5 输入管脚	PE5 输出管脚

表 8.75 PE4 的管脚功能

PE4DDR	0	1
管脚功能	PE4 输入管脚	PE4 输出管脚

表 8.76 PE3 的管脚功能

PE3DDR	0	1
管脚功能	PE3 输入管脚	PE3 输出管脚

表 8.77 PE2 的管脚功能

PE2DDR	0	1
管脚功能	PE2 输入管脚	PE2 输出管脚

表 8.78 PE1 的管脚功能

PE1DDR	0	1
管脚功能	PE1 输入管脚	PE1 输出管脚

表 8.79 PE0 的管脚功能

PE0DDR	0	1
管脚功能	PE0 输入管脚	PE0 输出管脚

8.10.6 端口 E 输入上拉 MOS 的状态

端口 E 内置可用程序控制的输入上拉 MOS。此输入上拉 MOS 可用于模式 4 ~ 6，8 位总线模式或模式 7，且可通过每个位指定 ON/OFF 状态。输入上拉 MOS 的状态如表 8.80 所示。

表 8.80 端口 E 输入上拉的状态

管脚	加电复位	硬件待机模式	手动复位	软件待机模式	其它工作时
数据输入 / 输出（模式 4 ~ 6 的 16 位总线） 端口输出（模式 4 ~ 6 的 8 位总线、模式 7）	OFF		OFF		
端口输入（模式 4 ~ 6 的 8 位总线、模式 7）	ON/OFF				

【符号说明】

OFF：输入上拉 MOS 通常为 OFF 状态。

ON/OFF：PEDDR = 0 且 PEPCR = 1 时为 ON 状态，否则为 OFF 状态。

8.11 端口 F

在 H8S/2218 群，端口 F 是兼用外部中断输入（ $\overline{\text{IRQ2}}$ 、 $\overline{\text{IRQ3}}$ ）管脚、总线控制信号输入 / 输出管脚、系统时钟输出管脚的 8 位输入 / 输出端口。在 H8S/2212 群，端口 F 是兼用外部中断输入（ $\overline{\text{IRQ2}}$ 、 $\overline{\text{IRQ3}}$ ）管脚、系统时钟输出管脚的 3 位输入 / 输出端口。端口 F 有以下寄存器。

- 端口 F 数据方向寄存器（PFDDR）
- 端口 F 数据寄存器（PFDR）
- 端口 F 寄存器（PORTF）

8.11.1 端口 F 数据方向寄存器（PFDDR）

PFDDR 是以每位指定端口 F 各管脚的输入 / 输出。

本寄存器是写专用寄存器，所以不能用位操作指令来写入。

详细请参考 2.9.4 包括写专用位的寄存器存取方法。

位	位名	初始值	R/W	说明
7	PF7DDR	1/0*1	W	(H8S/2218 群) 模式 4 ~ 6 时 PF7 管脚在 PF7DDR 置 1 时为 ϕ 输出管脚，清 0 则为输入端口。PF6 ~ PF3 管脚自动被指定为总线控制输出管脚。PF2 ~ PF0 管脚通过总线控制器的设定成为总线控制输入 / 输出管脚。此外，PFDDR 置 1，对应管脚为输出端口，清 0 则为输入端口。 模式 7 时 此位置 1，端口 F 管脚 PF7 为 ϕ 输出管脚，PF6 ~ PF0 为输出端口，清 0 后，则均为输入端口。 (H8S/2212 群) 此位置 1，端口 F 管脚 PF7 为 ϕ 输出管脚，PF6 ~ PF0 为输出端口；清 0 后，则均为输入端口。
6	PF6DDR*2	0	W	
5	PF5DDR*2	0	W	
4	PF4DDR*2	0	W	
3	PF3DDR	0	W	
2	PF2DDR*2	0	W	
1	PF1DDR*2	0	W	
0	PF0DDR	0	W	

【注】 *1 模式 4 ~ 6 时为 1、模式 7 时为 0。

*2 H8S/2212 群成为保留位。读取该位读出未定义值且该位不能被修改。

8.11.2 端口 F 数据寄存器（PFDR）

PFDR 保存端口 F 各管脚的输出数据。

位	位名	初始值	R/W	说明
7	PF7DR	0	R/W	给管脚保存输出数据以作为通用输出端口使用。
6	PF6DR*	0	R/W	
5	PF5DR*	0	R/W	
4	PF4DR*	0	R/W	
3	PF3DR	0	R/W	
2	PF2DR*	0	R/W	
1	PF1DR*	0	R/W	
0	PF0DR	0	R/W	

【注】 * 在 H8S/2212 群为保留位。读取该位读出未定义值且该位不能被修改。

8.11.3 端口 F 寄存器 (PORTF)

PORTF 表示端口 F 管脚的状态。

位	位名	初始值	R/W	说明
7	PF7	— *1	R	PFDDR 的位置 1, 读端口 F, 读取 PFDR 的值。 PFDDR 清 0, 读取端口 F 则读取管脚的状态。
6	PF6*2	— *1	R	
5	PF5*2	— *1	R	
4	PF4*2	— *1	R	
3	PF3	— *1	R	
2	PF2*2	— *1	R	
1	PF1*2	— *1	R	
0	PF0	— *1	R	

【注】 *1 根据 PF7 ~ PF0 管脚状态决定。

*2 在 H8S/2212 群为保留位。读后读出不定值。

8.11.4 时钟输出控制寄存器 (OUTCR)

OUTCR 是指定从 PF7 管脚输出时钟的频率。

位	位名	初始值	R/W	说明
7~3	—	不定	R/W	保留位 写时请写入 0。
2	PF7OUT2	0	R/W	PF7 管脚输出选择 2 ~ 0 000: 主振荡时钟 001: 输出主振荡时钟的 2 分频时钟 010: 输出主振荡时钟的 3 分频时钟 011: 输出主振荡时钟的 4 分频时钟 1XX: 保留 E6000 仿真程序不支持该功能。22. 电特性的 ϕ 在 PF7OUT2 ~ 0 = 000 时即为此种情况。
1	PF7OUT1	0	R/W	
0	PF7OUT0	0	R/W	

8.11.5 管脚功能

8.11.5.1 H8S/2218 群的管脚功能

端口 F 是兼用外部中断输入管脚 ($\overline{\text{IRQ2}}$ 、 $\overline{\text{IRQ3}}$)、总线控制信号输入 / 输出管脚及系统时钟输出 (ϕ) 管脚的 8 位输入 / 输出端口。寄存器的设定值和管脚功能的关系如下所示。

表 8.81 PF7 的管脚功能

PF7DDR	0	1			
PF7OUT2 ~ 0	—	B'000	B'001	B'010	B'011
管脚功能	PF7 输入管脚	ϕ 输出管脚	$\phi/2$ 输出管脚	$\phi/3$ 输出管脚	$\phi/4$ 输出管脚

表 8.82 PF6 的管脚的功能

工作模式	模式 4 ~ 6	模式 7	
PF6DDR	—	0	1
管脚功能	$\overline{\text{AS}}$ 输出管脚	PF6 输入管脚	PF6 输出管脚

表 8.83 PF5 的管脚功能

工作模式	模式 4 ~ 6	模式 7	
PF5DDR	—	0	1
管脚功能	$\overline{\text{RD}}$ 输出管脚	PF5 输入管脚	PF5 输出管脚

表 8.84 PF4 的管脚功能

工作模式	模式 4 ~ 6	模式 7	
PF4DDR	—	0	1
管脚功能	$\overline{\text{HWR}}$ 输出管脚	PF4 输入管脚	PF4 输出管脚

表 8.85 PF3 的管脚功能

工作模式	模式 4 ~ 6		模式 7		
总线模式	16 位总线模式	8 位总线模式		—	
PF3DDR	—	0	1	0	1
管脚功能	$\overline{\text{LWR}}$ 输出管脚	PF3 输入管脚	PF3 输出管脚	PF3 输入管脚	PF3 输出管脚
		$\overline{\text{ADTRG}}$ 输入管脚 *1			
		$\overline{\text{IRQ3}}$ 输入管脚 *2			

【注】 *1 TRGS0 = TRGS1 = 1 时成为 $\overline{\text{ADTRG}}$ 输入管脚。

*2 作为外部中断输入管脚使用时，不能作为其它功能的输入 / 输出管脚兼用。

表 8.86 PF2 的管脚功能

工作模式	模式 4 ~ 6			模式 7	
WAITE	0		1	—	
PF2DDR	0	1	—	0	1
管脚功能	PF2 输入管脚	PF2 输出管脚	$\overline{\text{WAIT}}$ 输入管脚	PF2 输入管脚	PF2 输出管脚

表 8.87 PF1 的管脚功能

工作模式	模式 4 ~ 6			模式 7	
BRLE	0		1	—	
PF1DDR	0	1	—	0	1
管脚功能	PF1 输入管脚	PF1 输出管脚	$\overline{\text{BACK}}$ 输出管脚	PF1 输入管脚	PF1 输出管脚

表 8.88 PF0 的管脚功能

工作模式	模式 4 ~ 6			模式 7	
BRLE	0		1	—	
PF0DDR	0	1	—	0	1
管脚功能	PF0 输入管脚	PF0 输出管脚	$\overline{\text{BREQ}}$ 输入管脚	PF0 输入管脚	PF0 输出管脚
	$\overline{\text{IRQ2}}$ 输入管脚 *				

【注】 * 作为外部中断输入管脚使用时，不能作为其它功能的输入 / 输出管脚兼用。

8.11.5.2 H8S/2212 群的管脚功能

端口 F 是兼用外部中断输入管脚 ($\overline{\text{IRQ2}}$ 、 $\overline{\text{IRQ3}}$) 及系统时钟输出 (ϕ) 管脚的 3 位的输入 / 输出端口。寄存器的设定值和管脚功能的关系如下所示。

表 8.89 PF7 的管脚功能

PF7DDR	0	1			
PF7OUT2 ~ 0	—	B'000	B'001	B'010	B'011
管脚功能	PF7 输入管脚	ϕ 输出管脚	$\phi/2$ 输出管脚	$\phi/3$ 输出管脚	$\phi/4$ 输出管脚

表 8.90 PF3 的管脚功能

PF3DDR	0	1
管脚功能	PF3 输入管脚	PF3 输出管脚
	$\overline{\text{ADTRG}}$ 输入管脚 *1	
	$\overline{\text{IRQ3}}$ 输入管脚 *2	

【注】 *1 TRGS0 = TRGS1 = 1 时， $\overline{\text{ADTRG}}$ 成为输入管脚。

*2 作为外部中断输入管脚使用时，不能作为其它功能的输入 / 输出管脚兼用。

表 8.91 PF0 的管脚功能

PF0DDR	0	1
管脚功能	PF0 输入管脚	PF0 输出管脚
	$\overline{\text{IRQ2}}$ 输入管脚 *	

【注】 * 作为外部中断输入端口使用时，不能作为其它功能的输入 / 输出管脚兼用。

8.12 端口 G

在 H8S/2218 群端口 G 是兼用外部中断输入管脚 ($\overline{\text{IRQ7}}$) 和总线控制输出管脚 ($\overline{\text{CS0}} \sim \overline{\text{CS3}}$) 的 4 位输入 / 输出端口。在 H8S/2212 群, 端口 G 是兼用外部中断输入管脚 ($\overline{\text{IRQ7}}$) 和 H-UDI (TDI) 的 2 位输入 / 输出端口。端口 G 有以下寄存器。

- 端口 G 数据方向寄存器 (PGDDR)
- 端口 G 数据寄存器 (PGDR)
- 端口 G 寄存器 (PORTG)

8.12.1 端口 G 数据方向寄存器 (PGDDR)

PGDDR 是以每位指定各端口的输入 / 输出。

本寄存器是写专用寄存器, 所以不能用位操作指令来写入。

详细请参考 2.9.4 包括写专用位的寄存器存取方法。

位	位名	初始值	R/W	说明
7~5	—	不定	—	保留位 该位未定义且不能被修改。
4	PG4DDR*2	0/1*1	W	(H8S/2218 群)
3	PG3DDR*2	0	W	模式 4 ~ 6 时
2	PG2DDR*2	0	W	PGDDR 置 1, PG4 ~ PG1 管脚为总线控制信号输出管脚, 清 0 则为输入端口。
1	PG1DDR	0	W	模式 7 时 PGDDR 置 1, 对应端口 G 管脚为输出端口, 清 0 后成为输入端口。 (H8S/2212 群) 将 PG1DDR 置 1 对应端口 G 管脚后成为输出端口, 清 0 则为输入端口
0	PG0DDR*3	0	W	(H8S/2212 群) EMLE = 1 时, PG0 管脚成为 H-UDI 管脚 (TDI)。 EMLE = 0 时, PG0DDR 置 1, PG0 管脚为输出端口, 清 0 则为输入端口。

【注】 *1 模式 4、5 时为 1、模式 6、7 时为 0。

*2 在 H8S/2212 群成为保留位。读取该位后读出未定义值且该位不能被修改。

*3 在 H8S/2218 群成为保留位。读取该位后读出未定义值且该位不能被修改。

8.12.2 端口 G 数据寄存器 (PGDR)

PGDR 保存端口 G 各管脚的输出数据。

位	位名	初始值	R/W	说明
7~5	—	不定	—	保留位 读后读出不定值。写定义为无效。
4	PG4DR*1	0	R/W	给管脚保存输出数据以作为通用输出端口使用。
3	PG3DR*1	0	R/W	
2	PG2DR*1	0	R/W	
1	PG1DR	0	R/W	
0	PG0DR*2	0	R/W	

【注】 *1 在 H8S/2212 群是保留位。读取该位后读出未定义值且该位不能被修改。

*2 在 H8S/2218 群是保留位。读取该位后读出未定义值且该位不能被修改。

8.12.3 端口 G 寄存器 (PORTG)

PORTG 表示端口 G 管脚的状态。

位	位名	初始值	R/W	说明
7~5	—	不定	—	保留位 读后读出不定值。写定义为无效。
4	PG4*2	— *1	R	PGDDR 置 1 的位，读端口 G 后读出 PGDR 的值。 PGDDR 清 0 时，读端口 G 后读出管脚的状态。
3	PG3*2	— *1	R	
2	PG2*2	— *1	R	
1	PG1	— *1	R	
0	PG0*3	— *1	R	

【注】 *1 根据 PG4 ~ PG0 管脚状态来决定。

*2 在 H8S/2212 群是保留位。读取该位后读出未定义值。

*3 在 H8S/2218 群是保留位。读取该位后读出未定义值。

8.12.4 管脚功能

8.12.4.1 H8S/2218 群的管脚功能

端口 G 兼用外部中断输入 ($\overline{\text{IRQ7}}$) 管脚和总线控制信号输出 ($\overline{\text{CS0}} \sim \overline{\text{CS3}}$) 管脚。寄存器的设定值和管脚功能的关系如下所示。

表 8.92 PG4 的管脚功能

工作模式	模式 4 ~ 6		模式 7	
PG4DDR	0	1	0	1
管脚功能	PG4 输入管脚	$\overline{\text{CS0}}$ 输出管脚	PG4 输入管脚	PG4 输出管脚

表 8.93 PG3 的管脚功能

工作模式	模式 4 ~ 6		模式 7	
PG3DDR	0	1	0	1
管脚功能	PG3 输入管脚	$\overline{\text{CS1}}$ 输出管脚	PG3 输入管脚	PG3 输出管脚

表 8.94 PG2 的管脚功能

工作模式	模式 4 ~ 6		模式 7	
PG2DDR	0	1	0	1
管脚功能	PG2 输入管脚	$\overline{\text{CS2}}$ 输出管脚	PG2 输入管脚	PG2 输出管脚

表 8.95 PG1 的管脚功能

工作模式	模式 4 ~ 6		模式 7	
PG1DDR	0	1	0	1
管脚功能	PG1 输入管脚	$\overline{\text{CS3}}$ 输出管脚	PG1 输入管脚	PG1 输出管脚
	$\overline{\text{IRQ7}}$ 输入管脚*			

【注】 * 作为外部中断输入管脚使用时，不能作为其它功能的输入 / 输出管脚兼用。

8.12.4.2 H8S/2212 群的管脚功能

端口 G 兼用外部中断输入 ($\overline{\text{IRQ7}}$) 管脚和 H-UDI (TDI) 管脚。寄存器的设定值和管脚功能的关系如下所示。

表 8.96 PG1 的管脚功能

PG1DDR	0	1
管脚功能	PG1 输入管脚	PG1 输出管脚
	$\overline{\text{IRQ7}}$ 输入管脚*	

【注】 * 作为外部中断输入管脚使用时，不能作为其它功能的输入 / 输出管脚兼用。

表 8.97 PG0 的管脚功能

EMLE	0		1
PG0DDR	0	1	—
管脚功能	PG0 输入管脚	PG0 输出管脚	TDI 输入管脚

8.13 未使用管脚的处理

请将未使用的输入管脚固定为高电平或低电平。

CMOS 产品的输入管脚一般为高阻抗输入。在开路状态下，运行未使用管脚时，会出现由于外围噪音感应，产生中间电平、并在内部产生通过电流引起误动的现象。未使用输入管脚的处理例如表 8.98 所示。

另外，边界扫描的专用管脚未使用时的处理，请参考 13.2 输入输出管脚、13.5 使用注意事项。请将 NC 管脚置于开路状态。

USB 专用管脚未使用时的处理，请参考 14.8.14 未使用 USB 时的管脚处理。

表 8.98 未使用输入管脚的处理例子

端口名	管脚处理例子
端口 1	通过电阻将每个管脚连接至 Vcc (上拉) 或至 Vss (下拉)。
端口 3	
端口 4	通过电阻将每个管脚连接至 AVcc (上拉) 或至 AVss (下拉)。
端口 7	通过电阻将每个管脚连接至 Vcc (上拉) 或至 Vss (下拉)。
端口 9	通过电阻将每个管脚连接至 AVcc (上拉) 或至 AVss (下拉)。
端口 A	通过电阻将每个管脚连接至 Vcc (上拉) 或至 Vss (下拉)。 * 端口 B, C 和 D 仅限于 H8S/2218 群使用。
端口 B*	
端口 C*	
端口 D*	
端口 E	
端口 F	
端口 G	

9. 16 位定时器脉冲单元 (TPU)

本 LSI 内置由 3 通道的 16 位定时器构成的 16 位定时器脉冲单元 (TPU)。16 位定时器脉冲单元的框图如图 9.1 所示，功能一览如表 9.1 所示。

9.1 特点

- 最大可输入/输出 8 个脉冲
- 每个通道可选择 8 种计数器输入时钟
- 各通道都设定为以下的工作

在比较匹配、输入捕捉功能、计数器清除操作输出波形，同时向多个定时器计数器 (TCNT) 写入，同时清除比较匹配或输入捕捉，同时计数器的同步运行向单个寄存器同时输入/输出，用户定义占空比的 PWM 输出和同步运行的组合产生最大 7 相的 PWM 输出。

- 通道 0 设定为缓冲运行
- 通道 1、2 设定为独立的相位计数模式
- 通过 16 位内部总线进行高速存取
- 13 种的中断源
- 进行寄存器数据的自动传送
- 生成 A/D 转换器的转换起始触发
- 设定模块停止模式
- 生成 SCI0 的波特率时钟

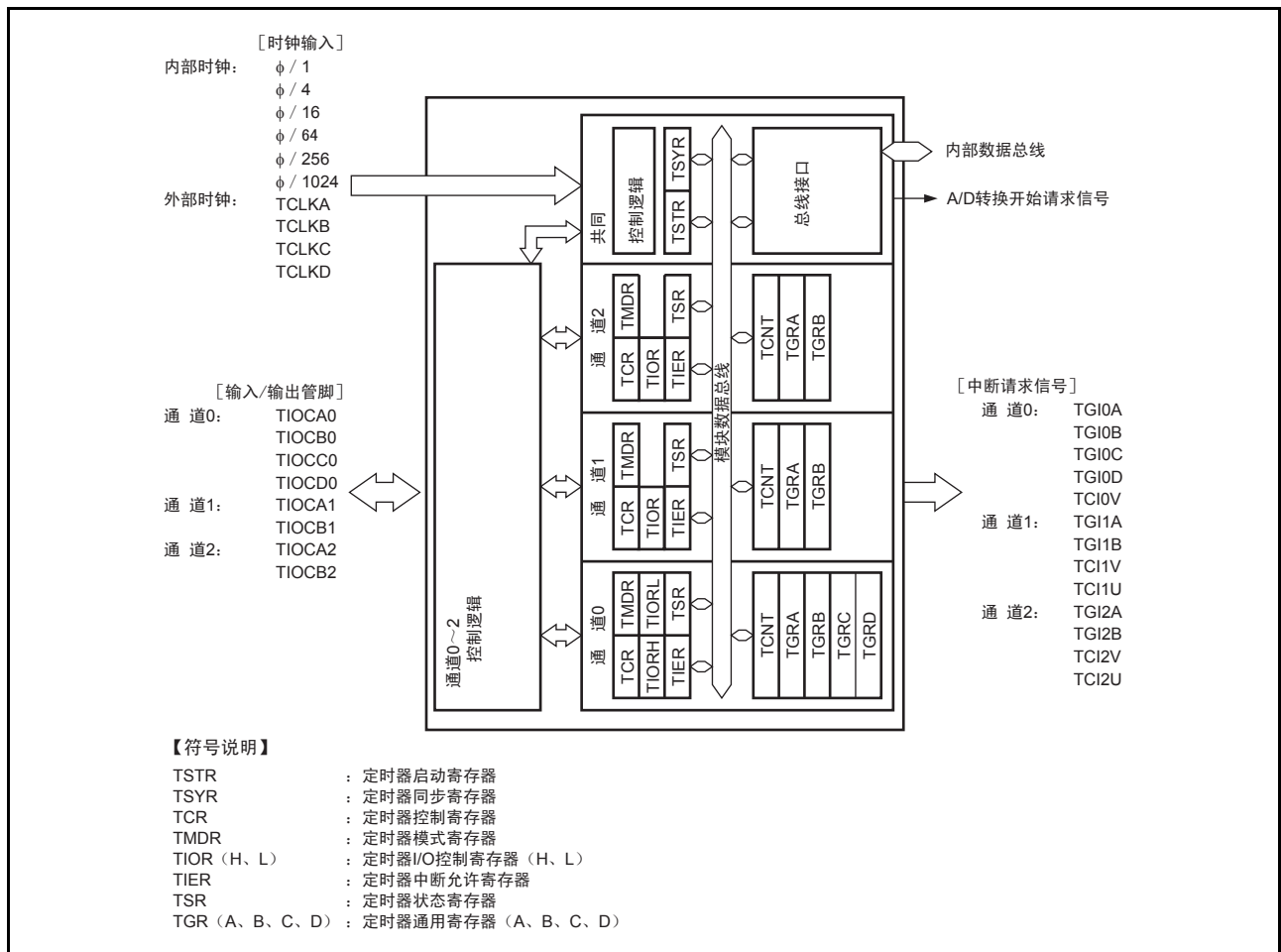


图 9.1 TPU 的框图

表 9.1 TPU 的功能一览

项目	通道 0	通道 1	通道 2
计数时钟	$\phi / 1$ $\phi / 4$ $\phi / 16$ $\phi / 64$ TCLKA TCLKB TCLKC TCLKD	$\phi / 1$ $\phi / 4$ $\phi / 16$ $\phi / 64$ $\phi / 256$ TCLKA TCLKB	$\phi / 1$ $\phi / 4$ $\phi / 16$ $\phi / 64$ $\phi / 1024$ TCLKA TCLKB TCLKC
通用寄存器 (TGR)	TGRA_0 TGRB_0	TGRA_1 TGRB_1	TGRA_2 TGRB_2
通用寄存器 / 缓冲寄存器	TGRC_0 TGRD_0	—	—
输入 / 输出管脚	TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2
计数器清除功能	TGR 比较匹配或输入捕捉	TGR 的比较匹配或输入捕捉	TGR 比较匹配或输入捕捉
比较匹配 输出	0 输出	○	○
	1 输出	○	○
	触发输出	○	○
输入捕捉功能	○	○	○
同步工作	○	○	○
PWM 模式	○	○	○
相位计数模式	—	○	○
缓冲运行	○	—	—
DMAC 的启动	TGRA_0 比较匹配或输入捕捉	TGRA_1 比较匹配或输入捕捉	TGRA_2 比较匹配或输入捕捉
A/D 转换开始触发	TGRA_0 比较匹配或输入捕捉	TGRA_1 比较匹配或输入捕捉	TGRA_2 比较匹配或输入捕捉
中断源	5 源 <ul style="list-style-type: none"> • 比较匹配 / 输入捕捉 0A • 比较匹配 / 输入捕捉 0B • 比较匹配 / 输入捕捉 0C • 比较匹配 / 输入捕捉 0D • 上溢 	4 源 <ul style="list-style-type: none"> • 比较匹配 / 输入捕捉 1A • 比较匹配 / 输入捕捉 1B • 上溢 • 下溢 	4 源 <ul style="list-style-type: none"> • 比较匹配 / 输入捕捉 2A • 比较匹配 / 输入捕捉 2B • 上溢 • 下溢

【符号说明】

○: 可能

—: 不可

9.2 输入 / 输出管脚

表 9.2 TPU 的管脚构成

通道	名称	输入 / 输出	功 能
公共	TCLKA	输入	外部时钟 A 输入管脚 (通道 1 的相位计数模式输入 A 相)
	TCLKB	输入	外部时钟 B 输入管脚 (通道 1 的相位计数模式输入 B 相)
	TCLKC	输入	外部时钟 C 输入管脚 (通道 2 的相位计数模式输入 A 相)
	TCLKD	输入	外部时钟 D 输入管脚 (通道 2 的相位计数模式输入 B 相)
0	TIOCA0	输入 / 输出	TGRA_0 的输入捕捉输入 / 输出比较输出 / PWM 输出管脚
	TIOCB0	输入 / 输出	TGRB_0 的输入捕捉输入 / 输出比较输出 / PWM 输出管脚
	TIOCC0	输入 / 输出	TGRC_0 的输入捕捉输入 / 输出比较输出 / PWM 输出管脚
	TIOCD0	输入 / 输出	TGRD_0 的输入捕捉输入 / 输出比较输出 / PWM 输出管脚
1	TIOCA1	输入 / 输出	TGRA_1 的输入捕捉输入 / 输出比较输出 / PWM 输出管脚
	TIOCB1	输入 / 输出	TGRB_1 的输入捕捉输入 / 输出比较输出 / PWM 输出管脚
2	TIOCA2	输入 / 输出	TGRA_2 的输入捕捉输入 / 输出比较输出 / PWM 输出管脚
	TIOCB2	输入 / 输出	TGRB_2 的输入捕捉输入 / 输出比较输出 / PWM 输出管脚

9.3 寄存器的说明

TPU 各通道有以下寄存器。

- 定时器控制寄存器_0 (TCR_0)
- 定时器模式寄存器_0 (TMDR_0)
- 定时器I/O控制寄存器H_0 (TIORH_0)
- 定时器I/O控制寄存器L_0 (TIORL_0)
- 定时器中断允许寄存器_0 (TIER_0)
- 定时器状态寄存器_0 (TSR_0)
- 定时器计数器_0 (TCNT_0)
- 定时器通用寄存器A_0 (TGRA_0)
- 定时器通用寄存器B_0 (TGRB_0)
- 定时器通用寄存器C_0 (TGRC_0)
- 定时器通用寄存器D_0 (TGRD_0)
- 定时器控制寄存器_1 (TCR_1)
- 定时器模式寄存器_1 (TMDR_1)
- 定时器I/O控制寄存器_1 (TIOR_1)
- 定时器中断允许寄存器_1 (TIER_1)
- 定时器状态寄存器_1 (TSR_1)
- 定时器计数器_1 (TCNT_1)
- 定时器通用寄存器A_1 (TGRA_1)
- 定时器通用寄存器B_1 (TGRB_1)
- 定时器控制寄存器_2 (TCR_2)
- 定时器模式寄存器_2 (TMDR_2)
- 定时器I/O控制寄存器_2 (TIOR_2)
- 定时器中断允许寄存器_2 (TIER_2)
- 定时器状态寄存器_2 (TSR_2)
- 定时器寄存器_2 (TCNT_2)
- 定时器通用寄存器A_2 (TGRA_2)
- 定时器通用寄存器B_2 (TGRB_2)

公共寄存器

- 定时器起始寄存器 (TSTR)
- 定时器单寄存器 (TSYR)

9.3.1 定时器控制寄存器 (TCR)

TCR 控制各通道的 TCNT 操作。在 TPU，通道 0 ~ 2 各有 1 个，共计有 3 个 TCR。TCR 应在 TCNT 运行停止状态设定。

位	位名	初始值	R/W	说明
7	CCLR2	0	R/W	计数器清除 2 ~ 0 选择 TCNT 的计数器清除源。详细请参考表 9.3、表 9.4。
6	CCLR1	0	R/W	
5	CCLR0	0	R/W	
4	CKEG1	0	R/W	时钟沿 1、0 该位选择输入时钟沿。内部时钟在用两沿计数时，输入时钟的周期为 1 / 2 (例: $\phi / 4$ 的两沿 = $\phi / 2$ 上升沿)。通道 1、2 为相位计数模式时，忽视本设定，优先相位计数的模式。输入时钟在 $\phi / 4$ 或更晚选择内部时钟沿时有效。输入时钟选择 $\phi / 1$ 时，忽略此设定，选择下降沿计数。 00: 用上升沿计数 01: 用下降沿计数 1x: 用两沿计数 【符号说明】 x: Don't care
3	CKEG0	0	R/W	
2	TPSC2	0	R/W	定时预定标器 2 ~ 0 该位选择 TCNT 的计数器时钟。各通道可独立选择时钟源。详细请参考表 9.5 ~ 表 9.7 TPSC2 ~ TPSC0。
1	TPSC1	0	R/W	
0	TPSC0	0	R/W	

表 9.3 CCLR2 ~ CCLR0 (通道 0)

通道	位 7 CCLR2	位 6 CCLR1	位 5 CCLR0	说明
0	0	0	0	禁止清除 TCNT
	0	0	1	用 TGRA 的比较匹配 / 输入捕捉清除 TCNT
	0	1	0	用 TGRB 的比较匹配 / 输入捕捉清除 TCNT
	0	1	1	用同步清除 / 同步工作的其它通道的清除计数器来清除 TCNT*1
	1	0	0	禁止清除 TCNT
	1	0	1	用 TGRC 的比较匹配 / 输入捕捉清除 TCNT*2
	1	1	0	用 TGRD 的比较匹配 / 输入捕捉清除 TCNT*2
	1	1	1	用同步清除 / 同步工作的其它通道的清除计数器来清除 TCNT*1

【注】 *1 同步运行的设定是通过 TSYR 的 SYNC 位置 1 来进行。

*2 TGRC 或 TGRD 作为缓冲寄存器使用时，优先缓冲寄存器的设定，为了不产生比较匹配 / 输入捕捉，不清除 TCNT。

表 9.4 CCLR2 ~ CCLR0 (通道 1、2)

通道	位 7	位 6	位 5	说明
	保留 *2	CCLR1	CCLR0	
1、2	0	0	0	禁止清除 TCNT
	0	0	1	用 TGRA 的比较匹配 / 输入捕捉清除 TCNT
	0	1	0	用 TGRB 的比较匹配 / 输入捕捉清除 TCNT
	0	1	1	用同步清除 / 同步工作的其它通道的清除计数器来清除 TCNT*1

【注】 *1 用 TSYR 的 SYNC 位置 1 设定同步操作。

*2 在通道 1、2 位 7 是保留位。读后通常读出 0。写定义为无效。

表 9.5 TPSC2 ~ TPSC0 (通道 0)

通道	位 2	位 1	位 0	说明
	TPSC2	TPSC1	TPSC0	
0	0	0	0	内部时钟：用 ϕ / 1 计数
	0	0	1	内部时钟：用 ϕ / 4 计数
	0	1	0	内部时钟：用 ϕ / 16 计数
	1	1	1	内部时钟：用 ϕ / 64 计数
	1	0	0	外部时钟：用 TCLKA 管脚输入计数
	1	0	1	外部时钟：用 TCLKB 管脚输入计数
	1	1	0	外部时钟：用 TCLKC 管脚输入计数
	1	1	1	外部时钟：用 TCLKD 管脚输入计数

表 9.6 TPSC2 ~ TPSC0 (通道 1)

通道	位 2	位 1	位 0	说明
	TPSC2	TPSC1	TPSC0	
1	0	0	0	内部时钟：用 ϕ / 1 计数
	0	0	1	内部时钟：用 ϕ / 4 计数
	0	1	0	内部时钟：用 ϕ / 16 计数
	0	1	1	内部时钟：用 ϕ / 64 计数
	1	0	0	外部时钟：用 TCLKA 管脚输入计数
	1	0	1	外部时钟：用 TCLKB 管脚输入计数
	1	1	0	内部时钟：用 ϕ / 256 计数
	1	1	1	禁止设定

【注】 通道 1 在相位计数模式时，此设定无效。

表 9.7 TPSC2 ~ TPSC0 (通道 2)

通道	位 2	位 1	位 0	说明
	TPSC2	TPSC1	TPSC0	
2	0	0	0	内部时钟: 用 ϕ / 1 计数
	0	0	1	内部时钟: 用 ϕ / 4 计数
	0	1	0	内部时钟: 用 ϕ / 16 计数
	0	1	1	内部时钟: 用 ϕ / 64 计数
	1	0	0	外部时钟: 用 TCLKA 管脚输入计数
	1	0	1	外部时钟: 用 TCLKB 管脚输入计数
	1	1	0	外部时钟: 用 TCLKC 管脚输入计数
	1	1	1	内部时钟: 用 ϕ / 1024 计数

【注】 通道 2 在相位计数模式时。此设定无效。

9.3.2 定时器模式寄存器 (TMDR)

TMDR 设定各通道的运行模式。TPU 的各通道有 1 个，共计有 3 个 TMDR。TMDR 在 TCNT 运行停止状态设定。

位	位名	初始值	R/W	说明
7、6	—	全为 1	—	保留位 读后通常读出 1。写定义为无效。
5	BFB	0	R/W	缓冲操作 B 设定 TGRB 正常运行或 TGRB 和 TGRD 的组合进行缓冲操作。TGRD 作为缓冲寄存器时，不产生 TGRD 的输入捕捉 / 输出比较。 没有 TGRD 的通道 1、2，此位成为保留位。读后通常读出 0。写定义为无效。 0: TGRB 正常运行 1: TGRB 和 TGRD 一起用于缓冲操作
4	BFA	0	R/W	缓冲操作 A 设定 TGRA 正常运行或 TGRA 和 TGRC 的组合进行缓冲操作。TGRC 作为缓冲寄存器时，不产生 TGRC 的输入捕捉 / 输出比较。 没有 TGRC 的通道 1、2，此位成为保留位。读后通常读出 0。写定义为无效。 0: TGRA 正常运行 1: TGRA 和 TGRC 一起用于缓冲操作
3	MD3	0	R/W	模式 3 ~ 0
2	MD2	0	R/W	MD3 ~ MD0 设定定时器的运行模式。
1	MD1	0	R/W	MD3 是保留位写时通常写入 0。
0	MD0	0	R/W	详细请参考表 9.8。

表 9.8 MD3 ~ MD0

位 3 MD3*1	位 2 MD2*2	位 1 MD1	位 0 MD0	说明
0	0	0	0	正常工作
0	0	0	1	保留
0	0	1	0	PWM 模式 1
0	0	1	1	PWM 模式 2
0	1	0	0	相位计数模式 1
0	1	0	1	相位计数模式 2
0	1	1	0	相位计数模式 3
0	1	1	1	相位计数模式 4
1	×	×	×	—

【符号说明】

×: Don't care

【注】 *1 MD3 是保留位。写时通常写入 0。

*2 通道 0 不可设定为相位计数模式。MD2 通常写入 0。

9.3.3 定时 I/O 控制寄存器 (TIOR)

TIOR 控制 TGR。TPU 的通道 0 有 2 个，通道 1、2 各 1 个，共计有 4 个 TIOR。TIOR 受 TMDR 设定的影响，请注意。

在计数器停止 (TSTR 的 CST 位清 0) 状态，TIOR 指定的初始输出有效。PWM 模式 2 时，指定计数器在清 0 时输出。TGRC 或 TGRD 设定为缓冲操作时，本设定无效，作为缓冲寄存器运行。

- TIORH_0、TIOR_1、TIOR_2

位	位名	初始值	R/W	说 明
7	IOB3	0	R/W	I/O 控制 B3 ~ B0 设定 TGRB 的功能。
6	IOB2	0	R/W	
5	IOB1	0	R/W	
4	IOB0	0	R/W	
3	IOA3	0	R/W	I/O 控制 A3 ~ A0 设定 TGRA 的功能。
2	IOA2	0	R/W	
1	IOA1	0	R/W	
0	IOA0	0	R/W	

- TIORL_0

位	位名	初始值	R/W	说 明
7	IOD3	0	R/W	I/O 控制 D3 ~ D0 设定 TGRD 的功能。
6	IOD2	0	R/W	
5	IOD1	0	R/W	
4	IOD0	0	R/W	
3	IOC3	0	R/W	I/O 控制 C3 ~ C0 设定 TGRC 的功能。
2	IOC2	0	R/W	
1	IOC1	0	R/W	
0	IOC0	0	R/W	

表 9.9 TIORH_0 (通道 0)

位 7	位 6	位 5	位 4	说 明	
IOB3	IOB2	IOB1	IOB0	TGRB_0 的功能	TIOCB0 管脚的功能
0	0	0	0	输出比较寄存器	禁止输出
0	0	0	1		初始输出为 0 用比较匹配输出 0
0	0	1	0		初始输出为 0 输出 用比较匹配输出 1
0	0	1	1		初始输出为 0 输出 用比较匹配交替输出
0	1	0	0		禁止输出
0	1	0	1		初始输出为 1 输出 用比较匹配输出 0
0	1	1	0		初始输出为 1 输出 用比较匹配输出 1
0	1	1	1		初始输出为 1 输出 用比较匹配交替输出
1	0	0	0		输入捕捉寄存器
1	0	0	1	捕捉输入源是 TIOCB0 管脚 用下降沿输入捕捉	
1	0	1	×	捕捉输入源是 TIOCB0 管脚 用两沿输入捕捉	
1	1	×	×	禁止设定	

【符号说明】

×: Don't care

表 9.10 TIORH_0 (通道 0)

位 3	位 2	位 1	位 0	说 明	
IOA3	IOA2	IOA1	IOA0	TGRA_0 的功能	TIOCA0 管脚的功能
0	0	0	0	输出比较寄存器	禁止输出
0	0	0	1		初始输出为 0 用比较匹配输出 0
0	0	1	0		初始输出为 0 输出 用比较匹配输出 1
0	0	1	1		初始输出为 0 输出 用比较匹配交替输出
0	1	0	0		禁止输出
0	1	0	1		初始输出为 1 输出 用比较匹配输出 0
0	1	1	0		初始输出为 1 输出 用比较匹配输出 1
0	1	1	1		初始输出为 1 输出 用比较匹配交替输出
1	0	0	0		输入捕捉寄存器
1	0	0	1	捕捉输入源是 TIOCA0 管脚 用下降沿输入捕捉	
1	0	1	×	捕捉输入源是 TIOCA0 管脚 用两沿输入捕捉	
1	1	×	×	禁止设定	

【符号说明】

×: Don't care

表 9.11 TIORL_0 (通道 0)

位 7	位 6	位 5	位 4	说 明	
IOD3	IOD2	IOD1	IOD0	TGRD_0 的功能	TIOCD0 管脚的功能
0	0	0	0	输出比较寄存器 *	禁止输出
0	0	0	1		初始输出为 0 用比较匹配输出 0
0	0	1	0		初始输出为 0 输出 用比较匹配输出 1
0	0	1	1		初始输出为 0 输出 用比较匹配交替输出
0	1	0	0		禁止输出
0	1	0	1		初始输出为 1 输出 用比较匹配输出 0
0	1	1	0		初始输出为 1 输出 用比较匹配输出 1
0	1	1	1		初始输出为 1 输出 用比较匹配交替输出
1	0	0	0	输入捕捉寄存器 *	捕捉输入源是 TIOCD0 管脚 用上升沿输入捕捉
1	0	0	1		捕捉输入源是 TIOCD0 管脚 用下降沿输入捕捉
1	0	1	×		捕捉输入源是 TIOCD0 管脚 用两沿输入捕捉
1	1	×	×		禁止设定

【符号说明】

×: Don't care

【注】 * TMDR_0 的 BFB 位置 1, TGRD_0 作为缓冲寄存器使用时, 本设定无效, 不产生输入捕捉 / 输出比较。

表 9.12 TIORL_0 (通道 0)

位 3 IOC3	位 2 IOC2	位 1 IOC1	位 0 IOC0	说 明	
				TGRC_0 的功能	TIOCC0 管脚的功能
0	0	0	0	输出比较寄存器 *	禁止输出
0	0	0	1		初始输出为 0 用比较匹配输出 0
0	0	1	0		初始输出为 0 输出 用比较匹配输出 1
0	0	1	1		初始输出为 0 输出 用比较匹配交替输出
0	1	0	0		禁止输出
0	1	0	1		初始输出为 1 输出 用比较匹配输出 0
0	1	1	0		初始输出为 1 输出 用比较匹配输出 1
0	1	1	1		初始输出为 1 输出 用比较匹配交替输出
1	0	0	0	输入捕捉寄存器 *	捕捉输入源是 TIOCC0 管脚 用上升沿输入捕捉
1	0	0	1		捕捉输入源是 TIOCC0 管脚 用下降沿输入捕捉
1	0	1	×		捕捉输入源是 TIOCC0 管脚 用两沿输入捕捉
1	1	×	×		禁止设定

【符号说明】

×: Don't care

【注】 * TMDR_0 的 BFA 位置 1, TGRC_0 作为缓冲寄存器使用时, 本设定无效, 不产生输入捕捉 / 输出比较。

表 9.13 TIOR_1 (通道 1)

位 7	位 6	位 5	位 4	说 明	
IOB3	IOB2	IOB1	IOB0	TGRB_1 的功能	TIOCB1 管脚的功能
0	0	0	0	输出比较寄存器 *	禁止输出
0	0	0	1		初始输出为 0 用比较匹配输出 0
0	0	1	0		初始输出为 0 输出 用比较匹配输出 1
0	0	1	1		初始输出为 0 输出 用比较匹配交替输出
0	1	0	0		禁止输出
0	1	0	1		初始输出为 1 输出 用比较匹配输出 0
0	1	1	0		初始输出为 1 输出 用比较匹配输出 1
0	1	1	1		初始输出为 1 输出 用比较匹配交替输出
1	0	0	0		输入捕捉寄存器
1	0	0	1	捕捉输入源是 TIOCB1 管脚 用下降沿输入捕捉	
1	0	1	×	捕捉输入源是 TIOCB1 管脚 用两沿输入捕捉	
1	1	×	×	禁止设定	

【符号说明】

×: Don't care

表 9.14 TIOR_1 (通道 1)

位 3	位 2	位 1	位 0	说 明	
IOA3	IOA2	IOA1	IOA0	TGRA_1 的功能	TIOCA1 管脚的功能
0	0	0	0	输出比较寄存器	禁止输出
0	0	0	1		初始输出为 0 用比较匹配输出 0
0	0	1	0		初始输出为 0 输出 用比较匹配输出 1
0	0	1	1		初始输出为 0 输出 用比较匹配交替输出
0	1	0	0		禁止输出
0	1	0	1		初始输出为 1 输出 用比较匹配输出 0
0	1	1	0		初始输出为 1 输出 用比较匹配输出 1
0	1	1	1		初始输出为 1 输出 用比较匹配交替输出
1	0	0	0		输入捕捉寄存器
1	0	0	1	捕捉输入源是 TIOCA1 管脚 用下降沿输入捕捉	
1	0	1	×	捕捉输入源是 TIOCA1 管脚 用两沿输入捕捉	
1	1	×	×	禁止设定	

【符号说明】

×: Don't care

表 9.15 TIOR_2 (通道 2)

位 7	位 6	位 5	位 4	说 明	
IOB3	IOB2	IOB1	IOB0	TGRB_2 的功能	TIOCB2 管脚的功能
0	0	0	0	输出比较寄存器	禁止输出
0	0	0	1		初始输出为 0 用比较匹配输出 0
0	0	1	0		初始输出为 0 输出 用比较匹配输出 1
0	0	1	1		初始输出为 0 输出 用比较匹配交替输出
0	1	0	0		禁止输出
0	1	0	1		初始输出为 1 输出 用比较匹配输出 0
0	1	1	0		初始输出为 1 输出 用比较匹配输出 1
0	1	1	1		初始输出为 1 输出 用比较匹配交替输出
1	×	0	0		输入捕捉寄存器
1	×	0	1	捕捉输入源是 TIOCB2 管脚 用下降沿输入捕捉	
1	×	1	×	捕捉输入源是 TIOCB2 管脚 用两沿输入捕捉	

【符号说明】

×: Don't care

表 9.16 TIOR_2 (通道 2)

位 3	位 2	位 1	位 0	说 明	
IOA3	IOA2	IOA1	IOA0	TGRA_2 的功能	TIOCA2 管脚的功能
0	0	0	0	输出比较寄存器	禁止输出
0	0	0	1		初始输出为 0 用比较匹配输出 0
0	0	1	0		初始输出为 0 输出 用比较匹配输出 1
0	0	1	1		初始输出为 0 输出 用比较匹配交替输出
0	1	0	0		禁止输出
0	1	0	1		初始输出为 1 输出 用比较匹配输出 0
0	1	1	0		初始输出为 1 输出 用比较匹配输出 1
0	1	1	1		初始输出为 1 输出 用比较匹配交替输出
1	×	0	0		输入捕捉寄存器
1	×	0	1	捕捉输入源是 TIOCA2 管脚 用下降沿输入捕捉	
1	×	1	×	捕捉输入源是 TIOCA2 管脚 用两沿输入捕捉	

【符号说明】

×: Don't care

9.3.4 定时器中断允许寄存器 (TIER)

TIER 控制各通道中断请求的允许或禁止。TPU 各通道有 1 个，共计有 3 个 TIER。

位	位名	初始值	R/W	说明
7	TTGE	0	R/W	允许 A/D 转换开始请求 禁止或允许用 TGRA 的输入捕捉 / 比较匹配产生 A/D 转换开始请求。 0: 禁止 A/D 转换开始请求的产生 1: 允许 A/D 转换开始请求的产生
6	—	1	—	保留位 读后读出 1。写定义为无效。
5	TCIEU	0	R/W	允许下溢中断 在通道 1、2，TSR 的 TCFU 标志置 1 时，允许或禁止用 TCFU 标志产生中断请求 (TCIU)。 在通道 0 是保留位。 读后通常读出 0。写定义为无效。 0: 用 TCFU 禁止中断请求 (TCIU) 1: 用 TCFU 允许中断请求 (TCIU)
4	TCIEV	0	R/W	允许上溢中断 TSR 的 TCFV 标志置 1 时，允许或禁止用 TCFV 标志产生中断请求 (TCIV)。 0: 用 TCFV 禁止中断请求 (TCIV) 1: 用 TCFV 允许中断请求 (TCIV)
3	TGIED	0	R/W	TGR 中断允许 D 在通道 0，TSR 的 TGFD 位置 1 时，允许或禁止用 TGFD 位产生中断请求 (TGID)。在通道 1、2 是保留位。读后通常读出 0。写定义为无效。 0: 用 TGFD 位禁止中断请求 (TGID) 1: 用 TGFD 位允许中断请求 (TGID)
2	TGIEC	0	R/W	TGR 中断允许 C 在通道 0，TSR 的 TGFC 位置 1 时，允许或禁止用 TGFC 位产生中断请求 (TGIC)。 在通道 1、2 是保留位。读后通常读出 0。写定义为无效。 0: 用 TGFC 位禁止中断请求 (TGIC) 1: 用 TGFC 位允许中断请求 (TGIC)
1	TGIEB	0	R/W	TGR 中断允许 B TSR 的 TGFB 位置 1 时，允许或禁止用 TGFB 位产生中断请求 (TGIB)。 0: 用 TGFB 位禁止中断请求 (TGIB) 1: 用 TGFB 位允许中断请求 (TGIB)
0	TGIEA	0	R/W	TGR 中断允许 A TSR 的 TGFA 位置 1 时，禁止或允许用 TGFA 位产生中断请求 (TGIA)。 0: 用 TGFA 位禁止中断请求 (TGIA) 1: 用 TGFA 位允许中断请求 (TGIA)

9.3.5 定时状态寄存器 (TSR)

TSR 表示各通道的状态。TPU 的各通道有 1 个，共计有 3 个 TSR。

位	位名	初始值	R/W	说明
7	TCFD	1	R	计数方向标志 表示通道 1、2 的 TCNT 计数方向的状态标志。 在通道 0 是保留位。读后通常读出 1。写定义为无效。 0: TCNT 递减计数 1: TCNT 递增计数
6	—	1	—	保留位 读后读出 1。写定义为无效。
5	TCFU	0	R/(W)*	下溢标志 通道 1、2 在相位计数模式时，表示 TCNT 下溢产生的状态标志。 在通道 0 是保留位。读后通常读出 0。写定义为无效。 [置位条件] TCNT 的值在下溢 (H'0000 → H'FFFF) 时 [清除条件] 在 TCFU = 1 状态，读 TCFU 后，向 TCFU 写入 0 时
4	TCFV	0	R/(W)*	上溢标志 表示 TCNT 上溢产生的状态标志。 [置位条件] TCNT 的值在上溢 (H'FFFF → H'0000) 时 [清除条件] 在 TCFV = 1 的状态，读 TCFV 后，向 TCFV 写 0 时
3	TGFD	0	R/(W)*	输入捕捉 / 输出比较标志 D 表示通道 0 的 TGRD 输入捕捉或比较匹配产生的状态标志。 在通道 1、2 是保留位。读后通常读出 0。写定义为无效。 [置位条件] • TGRD 作为输出比较寄存器使用，TCNT = TGRD • TGRD 作为输入捕捉使用时，用输入捕捉信号将 TCNT 的值向 TGRD 传送时 [清除条件] TGFD = 1 的状态读 TGFD 后，向 TGFD 写 0 时
2	TGFC	0	R/(W)*	输入捕捉 / 输出比较标志 C 通道 0 的 TGRC 输入捕捉或比较匹配产生的状态标志。 在通道 1、2 是保留位。读后通常读出 0。写定义为无效。 [置位条件] • TGRC 作为输出比较寄存器使用，TCNT = TGRC 时 • TGRC 作为输入捕捉使用时，通过输入捕捉信号将 TCNT 的值向 TGRC 传送时 [清除条件] 在 TGFC = 1 状态读 TGFC 后，向 TGFC 写 0 时

位	位名	初始值	R/W	说明
1	TGFB	0	R/(W)*	输入捕捉 / 输出比较标志 B 表示 TGRB 的输入捕捉或比较匹配产生的状态标志。 [置位条件] <ul style="list-style-type: none"> • TGRB 作为输出比较寄存器使用, TCNT = TGRB 时 • TGRB 作为输入捕捉使用时, 通过输入捕捉信号将 TCNT 的值传送到 TGRB 时 [清除条件] 在 TGFB = 1 状态读 TGFB 后, 向 TGFB 写 0 时
0	TGFA	0	R/(W)*	输入捕捉 / 输出比较标志 A 表示 TGRA 输入捕捉或比较匹配产生的状态标志。 [置位条件] <ul style="list-style-type: none"> • TGRA 作为输出比较寄存器, TCNT = TGRA 时 • TGRA 作为输入捕捉, 通过输入捕捉信号将 TCNT 的值向 TGRA 传送时 [清除条件] <ul style="list-style-type: none"> • 通过 TGIA 中断启动 DMAC, DMAC 的 DMABCR 的 DTA 位是 1 时 • TGFA = 1 的状态读 TGFA 后, 向 TGFA 写入 0 时

【注】 * 为清除标志, 仅可写入 0。

9.3.6 定时器寄存器 (TCNT)

TCNT 是 16 位可读 / 写的计数器。各通道有 1 个, 共计有 3 个 TCNT。

TCNT 在置位或硬件待机模式时, 初始化为 H'0000。

TCNT 禁止通过 8 位单位存取。必须用 16 位单位存取。

9.3.7 定时器通用寄存器 (TGR)

TGR 是兼用输出比较和输入捕捉的 16 位寄存器, 可进行 16 位的读取 / 写入。在通道 0 各有 4 个, 通道 1、2 各有 2 个, 共计有 8 个 16 位 TGR 寄存器。通道 0 的 TGRC 和 TGRD 可设定为缓冲寄存器运行。TRG 在置位或硬件待机模式时, 初始化为 H'FFFF。禁止通过 TGR 的 8 位单位存取。必须用 16 位单位存取。TGR 和缓冲寄存器的组合为 TGRA - TGRC、TGRB - TGRD。

9.3.8 定时器起始寄存器 (TSTR)

TSTR 选择通道 0 ~ 2 的 TCNT 操作 / 停止的寄存器。TSTR 对应位置 1 通道的 TCNT 进行计数操作。先停止 TCNT 的计数器操作后, 才能在 TMDR 设定操作模式或在 TCR 设定计数时钟。

位	位名	初始值	R/W	说 明
7~3	—	全部为 0	—	保留位 写入值常为 0。
2	CST2	0	R/W	计数器起始 2 ~ 0 选择 TCNT 的运行或停止。 在 TIOC 管脚输出状态运行时, 向 CST 位写 0 时停止计数器, 但是保持 TIOC 管脚输出比较的输出电平。 CST 位在 0 状态向 TIOR 写入时, 管脚输出电平将变更为初始输出值的设置。 0: 停止 TCNTn 的计数操作 1: TCNTn 是执行计数操作 (n = 2 ~ 0)
1	CST1	0	R/W	
0	CST0	0	R/W	

9.3.9 定时器单寄存器 (TSYR)

TSYR 选择通道 0 ~ 2 的 TCNT 独立操作或同步操作。TSYR 中对应位置 1 的通道进行同步操作。

位	位名	初始值	R/W	说 明
7~3	—	全部为 0	—	保留位 写入值常为 0。
2	SYNC2	0	R/W	定时器同步 2 ~ 0 该位选择与其它通道的独立操作或同步操作。 通过多个 TCNT 的同步预置和其它通道的计数器清除, 选择同步操作时可同步清除。 在设定同步操作时, 需要将最少 2 个通道的 SYNC 位置 1。在设定同步清除时, 除 SYNC 位以外, 需要用 TCR 的 CCLR2 ~ CCLR0 位设定 TCNT 的清除源。 0: TCNTn 是单独操作 (TCNT 的预置 / 清除与其它通道无关) 1: TCNTn 能进行同步操作 TCNT 的同步预置 / 同步清除 (n = 2 ~ 0)
1	SYNC1	0	R/W	
0	SYNC0	0	R/W	

9.4 总线主控器接口

9.4.1 16 位寄存器

TCNT、TGR 是 16 位寄存器。与总线主控器之间的数据总线是 16 位宽，所以可用 16 位单位读出 / 写入。不可用 8 位单位读出 / 写入。必须用 16 位单位存取。

16 位寄存器的存取操作例子如图 9.2 所示。

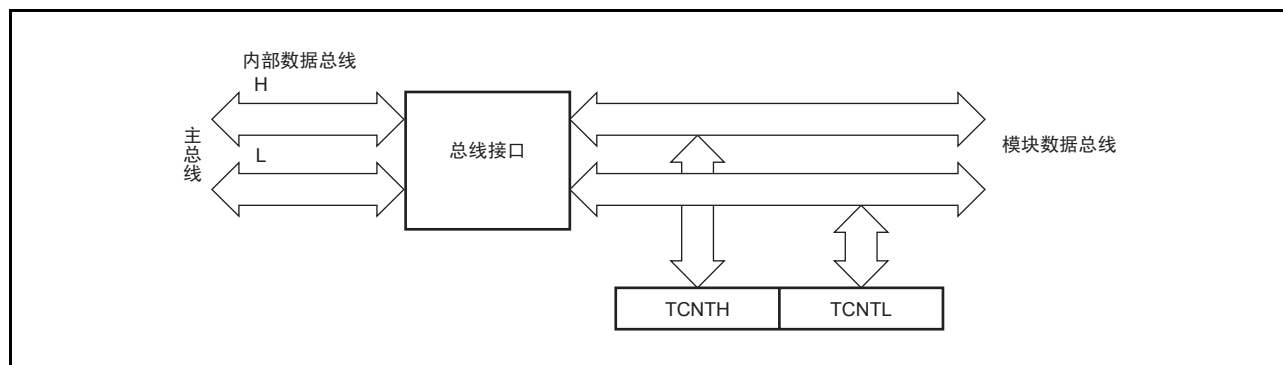


图 9.2 16 位寄存器的存取工作 (总线主控器 ↔ TCNT (16 位))

9.4.2 8 位寄存器

TCNT、TGR 以外的寄存器是 8 位寄存器。与总线主控器之间的数据总线是 16 位宽，所以能通过 16 位单位读出 / 写入。而且，这些寄存器通过 8 位单位也能读出 / 写入。

8 位寄存器的存取操作例子如图 9.3、~图 9.5 所示。

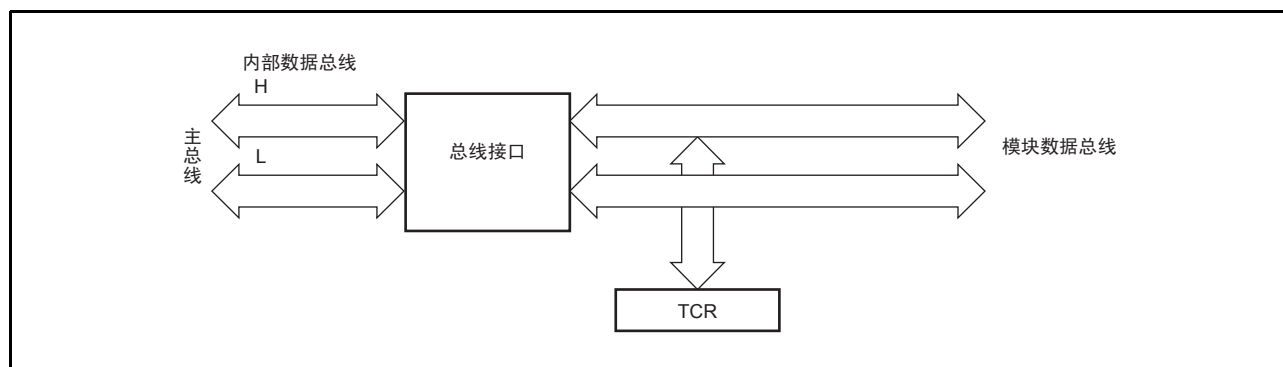


图 9.3 8 位寄存器的存取工作 (总线主控器 ↔ TCR (高 8 位))

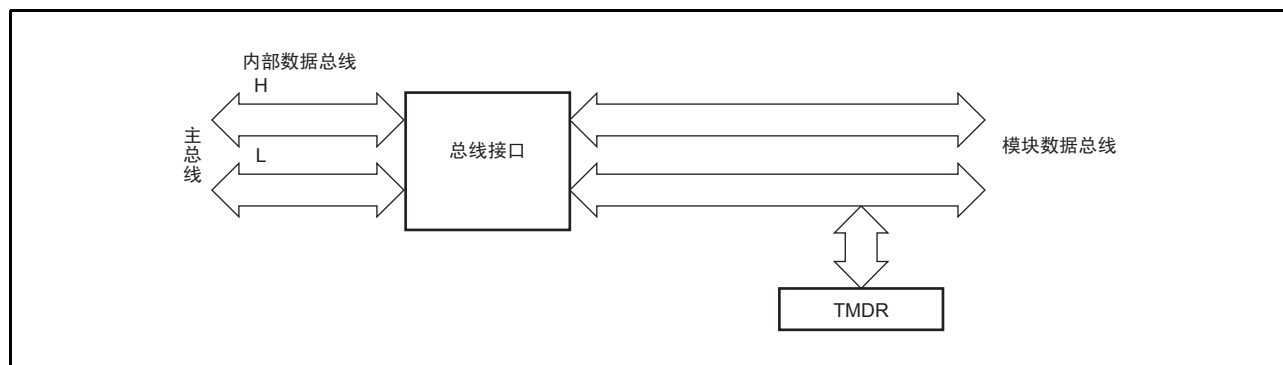


图 9.4 8 位寄存器的存取工作 (总线主控器 ↔ TMDR (低 8 位))

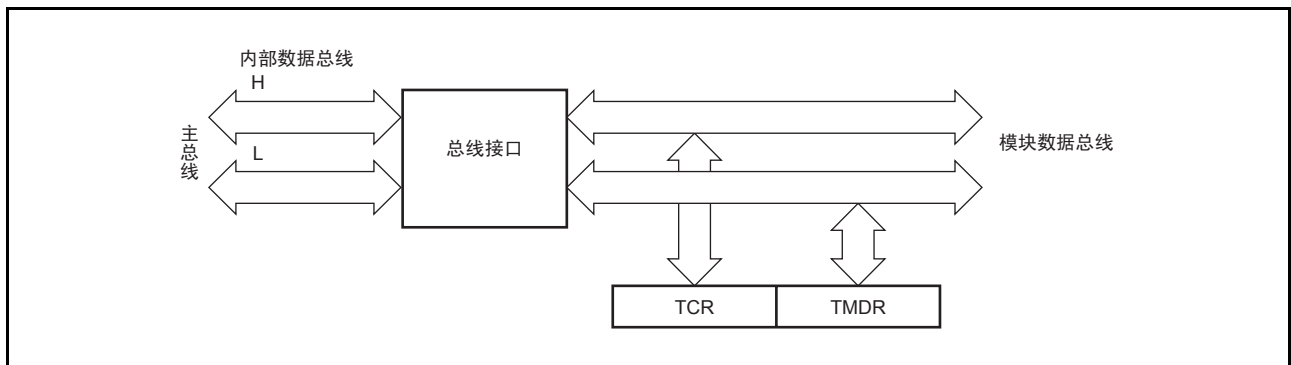


图 9.5 8 位寄存器的存取工作 (总线主控器 ↔ TCR、TMDR (16 位))

9.5 工作说明

9.5.1 基本工作

各通道都有 TCNT 和 TGR。TCNT 进行递增计数，能自由运行操作、周期计数运行、或外部事件计数。TGR 能分别作为输入捕捉寄存器或输出比较寄存器使用。

9.5.1.1 计数器操作

TSTR 的 CST0 ~ CST2 位置 1 后，对应通道的 TCNT 计数器开始计数。作为自由运行计数器、周期计数器等操作。

(1) 计数操作设定顺序的例子

计数操作设定顺序的例子如图 9.6 所示。

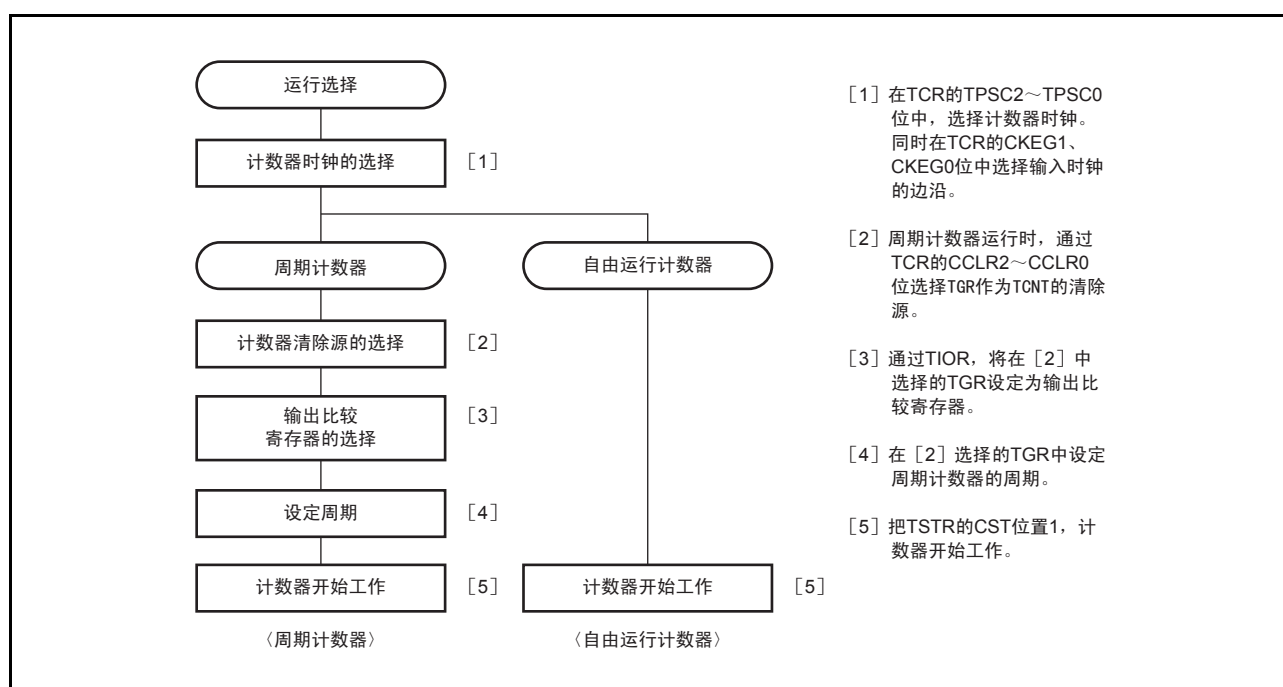


图 9.6 计数器工作设定顺序例子

(2) 自由运行计数操作和周期计数操作

TPU 的 TCNT 计数器在复位后，全部用作自由运行计数器，TSTR 对应位置 1 后作为自由运行计数器，对应的 TCNT 计数器开始递增计数操作。TCNT 上溢 (H'FFFF → H'0000) 后，TSR 的 TCFV 位置 1。此时，对应 TIER 的 TCIEV 位如果是 1，TPU 请求中断。TCNT 上溢后，从 H'0000 开始，继续递增计数运行。

自由运行计数器操作如图 9.7 所示。

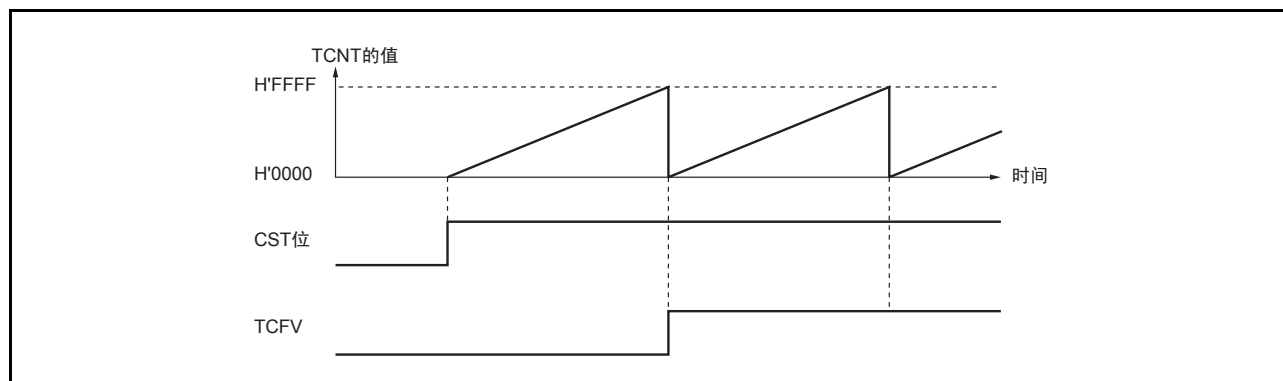


图 9.7 自由运行计数器的工作

当选择比较匹配作为 TCNT 清除源时，对应通道的 TCNT 将进行周期计数操作。用于周期设定的 TGR 被设定为输出比较寄存器，并以 TCR 的 CCLR2~CCLR0 位选择比较匹配清除计数器。设定后，TSTR 对应的位置 1 时，作为周期计数器开始递增计数器操作。计数值与 TGR 的值一致后，TSR 的 TGF 位置 1，TCNT 清除为 H'0000。

此时，对应 TIER 的 TGIE 位如为 1，TPU 请求中断。比较匹配后，TCNT 从 H'0000 开始继续递增计数运行。

周期计数器操作如图 9.8 所示。

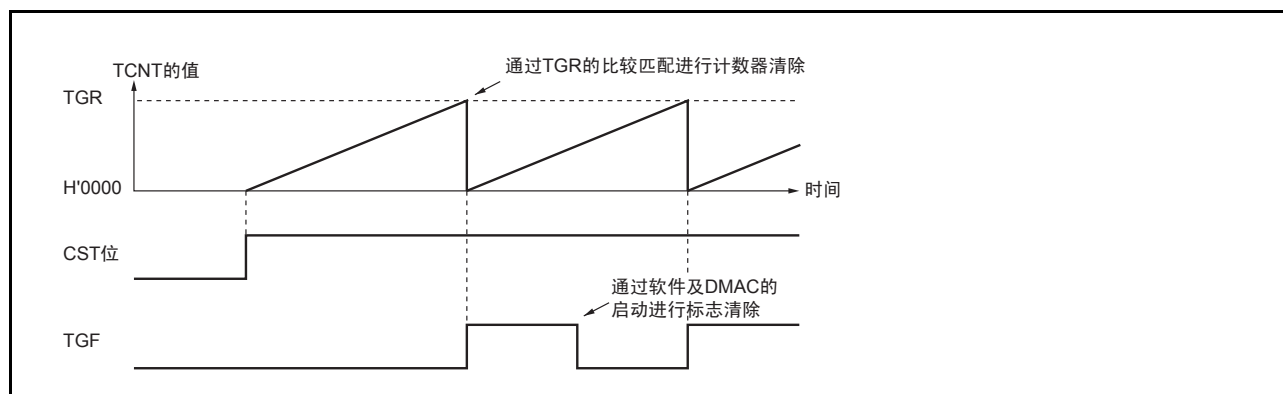


图 9.8 周期计数器的工作

9.5.1.2 用比较匹配输出波形功能

TPU 能用比较匹配从对应的输出管脚进行 0 输出 / 1 输出 / 交替输出。

(1) 用比较匹配输出波形的设定顺序的例子

用比较匹配输出波形的设定顺序的例子如图 9.9 所示。

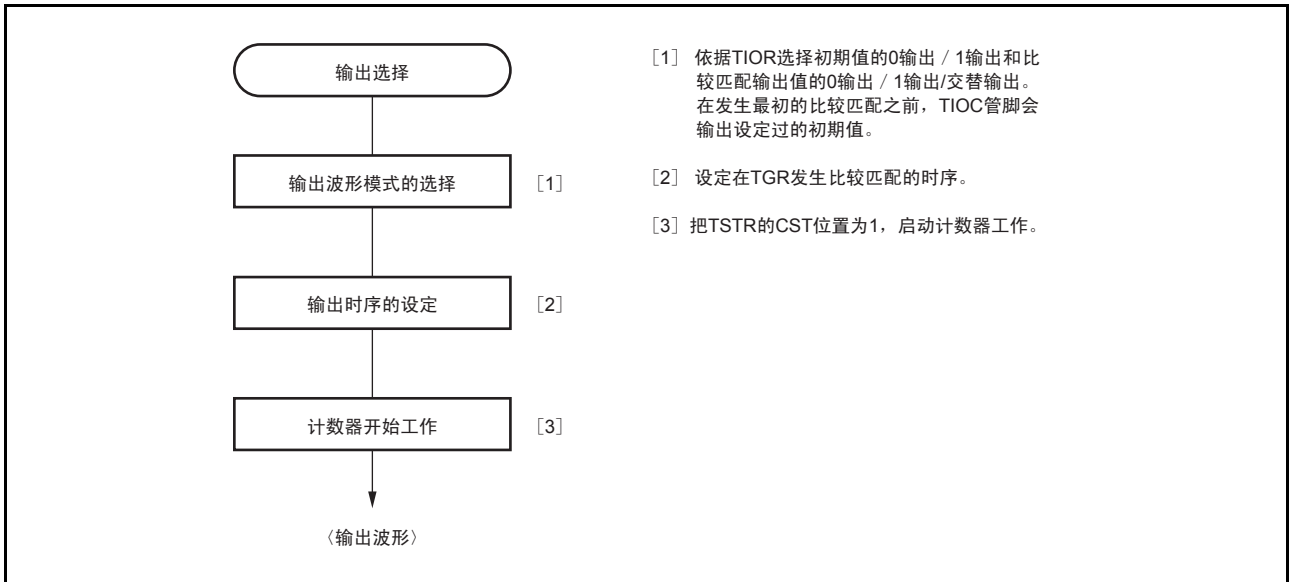


图 9.9 通过比较匹配输出波形的工作例子

(2) 波形输出操作例子

0 输出 / 1 输出例子如图 9.10 所示。

TCNT 设定为自由运行计数工作时的例子，用比较匹配 A 输出 1，用比较匹配 B 输出 0。设定电平与管脚的电平一致时，管脚的电平不变化。

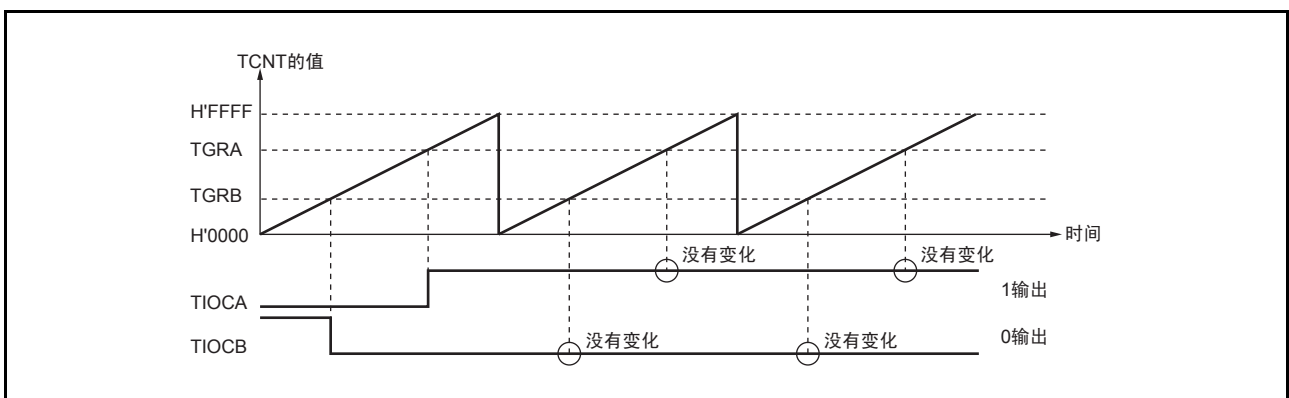


图 9.10 0 输出 / 1 输出的工作例子

交替输出的例子如图 9.11 所示。

TCNT 设定为周期计数工作时的例子（通过比较匹配 B 计数器清除），用比较匹配 A、比较匹配 B 交替输出。

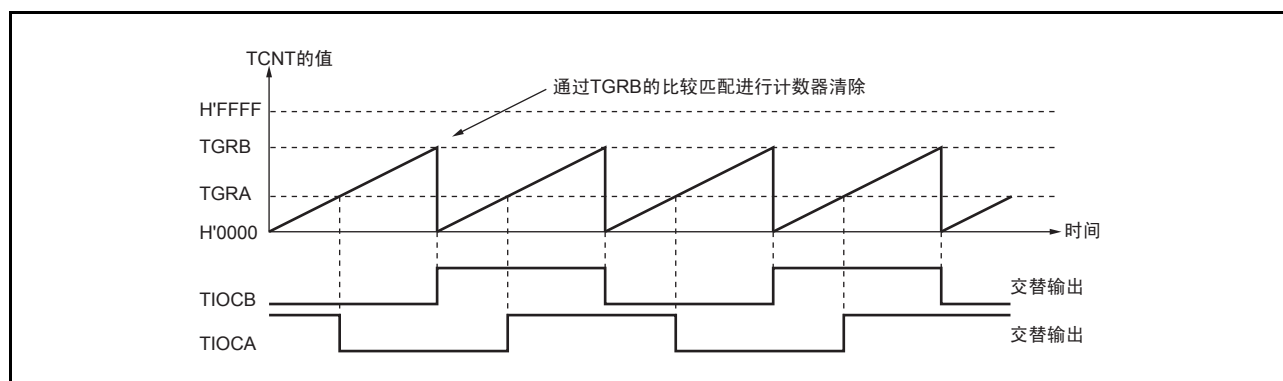


图 9.11 交替输出的工作例子

9.5.1.3 输入捕捉功能

检测出 TIOC 管脚的输入沿，能将 TCNT 的值向 TGR 传送。

检测沿可从上升沿 / 下降沿 / 两沿选择。

(1) 输入捕捉操作的设定顺序例子

输入捕捉操作的设定顺序例子如图 9.12 所示。

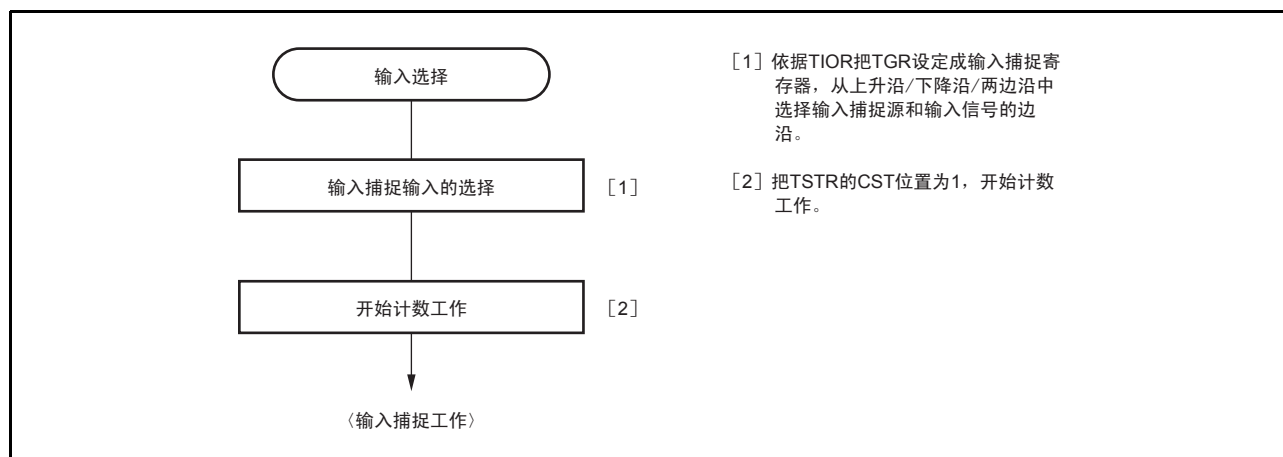


图 9.12 输入捕捉工作的设定例子

(2) 输入捕捉操作例子

输入捕捉操作例子如图 9.13 所示。

从上升沿 / 下降沿的两沿选择 TIOCA 管脚输入捕捉的输入沿，TIOCB 管脚的输入捕捉的输入沿选为下降沿，TCNT 用 TGRB 的输入捕捉设定计数器清除时的例子。

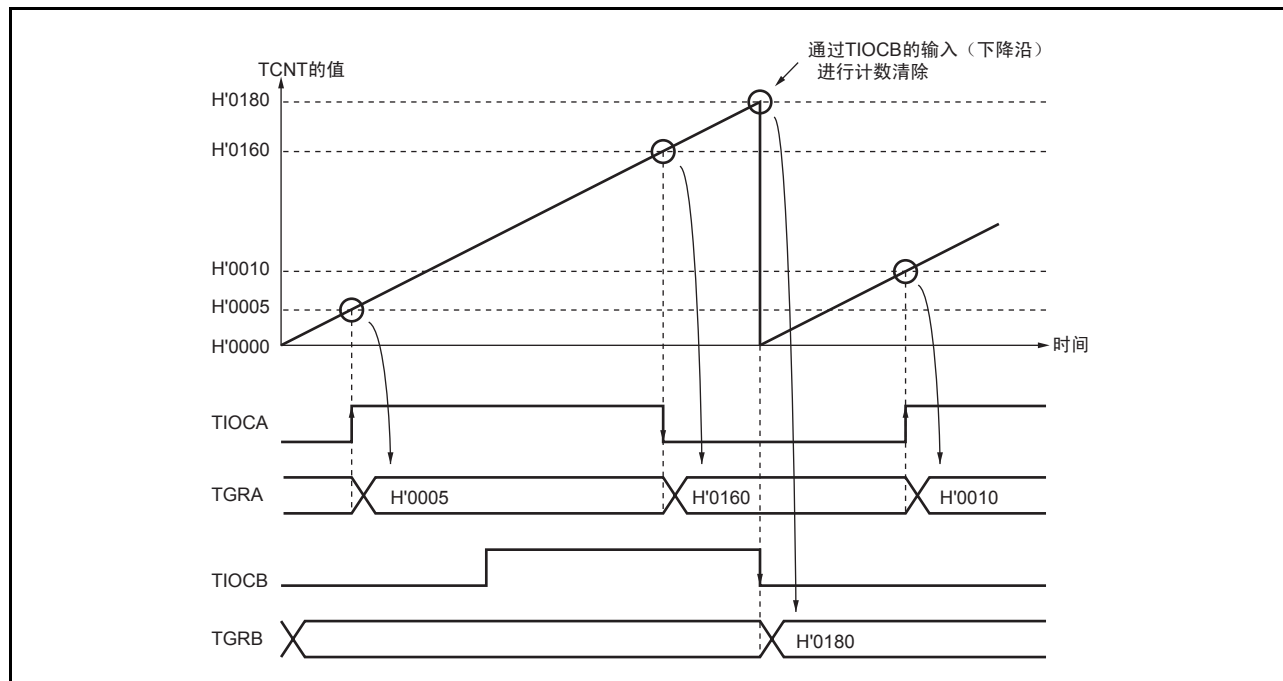


图 9.13 输入捕捉工作例子

9.5.2 同步工作

同步操作可同时改写多个 TCNT 的值（同步预置）。而且，用 TCR 的设定可同时清除多个 TCNT（同步清除）。

用同步操作，能对 1 个时基增加 TGR。

通道 0 ~ 2 可全部设定为同步操作。

9.5.2.1 同步操作的设定顺序例子

同步操作的设定顺序例子如图 9.14 所示。

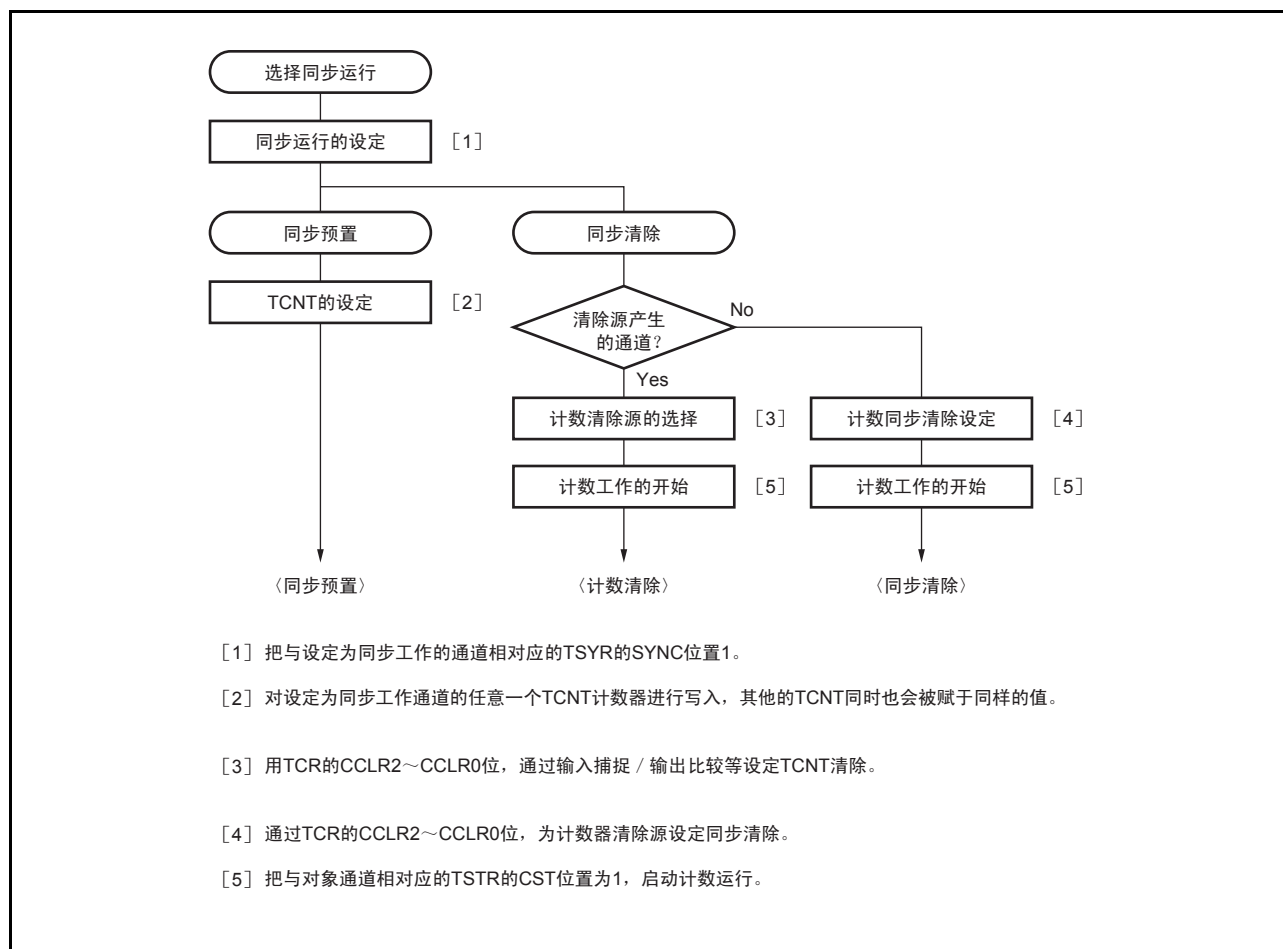


图 9.14 同步工作的设定顺序例子

9.5.2.2 同步操作的例子

同步操作的例子如图 9.15 所示。

通道 0 ~ 2 设定为同步操作和 PWM 模式 1，通道 0 的计数器清除源设定为 TGRB_0 的比较匹配，通道 1、2 的计数器清除源设定为同步清除时的例子。

从 TIOCA0、TIOCA1、TIOCA2 管脚输出 3 相的 PWM 波形。此时，通道 0 ~ 2 的 TCNT 通过同步预置、TGRB_0 的比较匹配同步清除，TGRB_0 设定的数据设置成 PWM 周期。

PWM 模式请参考 9.5.4 PWM 模式。

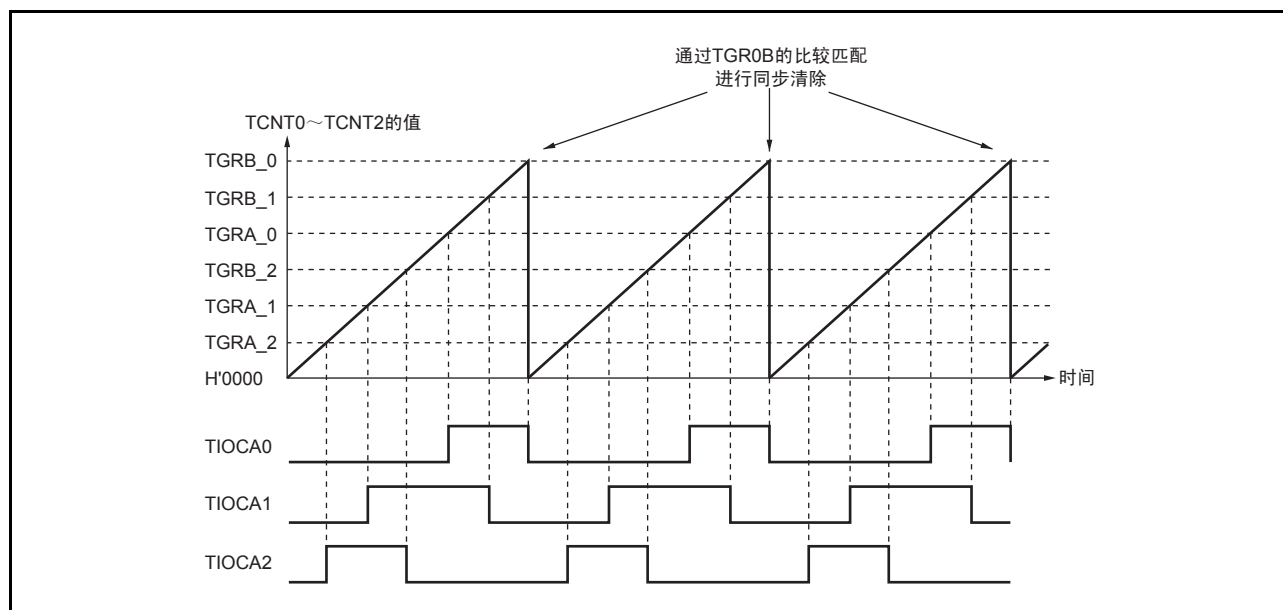


图 9.15 同步工作的工作例子

9.5.3 缓冲运行

通道 0 具有缓冲操作功能。TGRC 和 TGRD 可作为缓冲寄存器。将 TGR 设定为输入捕捉寄存器或比较匹配寄存器时，缓冲操作内容也有所不同。表 9.17 表示缓冲操作时的寄存器组合。

表 9.17 寄存器的组合

通道	定时通用寄存器	缓冲寄存器
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0

1. TGR 作为输出比较寄存器

产生比较匹配时，对应通道缓冲寄存器的值传送到定时器通用寄存器。此操作如图 9.16 所示。

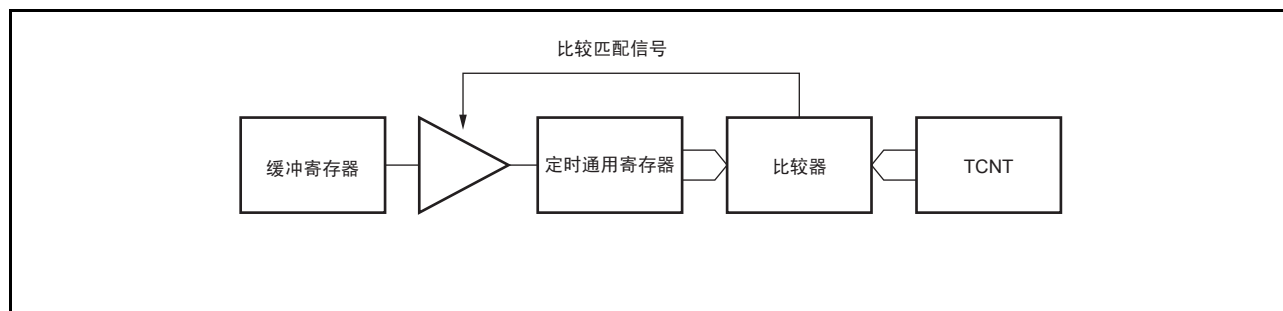


图 9.16 比较匹配缓冲运行

2. TGR 作为输入捕捉寄存器

产生输入捕捉后，TCNT 的值向 TGR 传送的同时，到此时为止保存在定时器通用寄存器的值传送到缓冲寄存器。

此操作如图 9.17 所示。

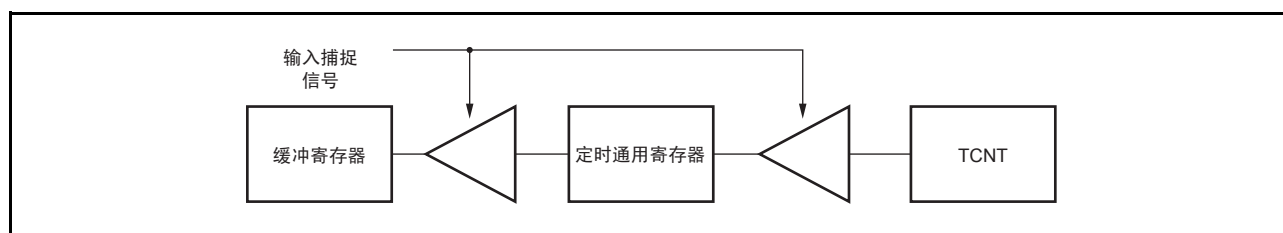


图 9.17 输入捕捉缓冲运行

9.5.3.1 缓冲操作的设定顺序例子

缓冲操作的设定顺序例子如图 9.18 所示。

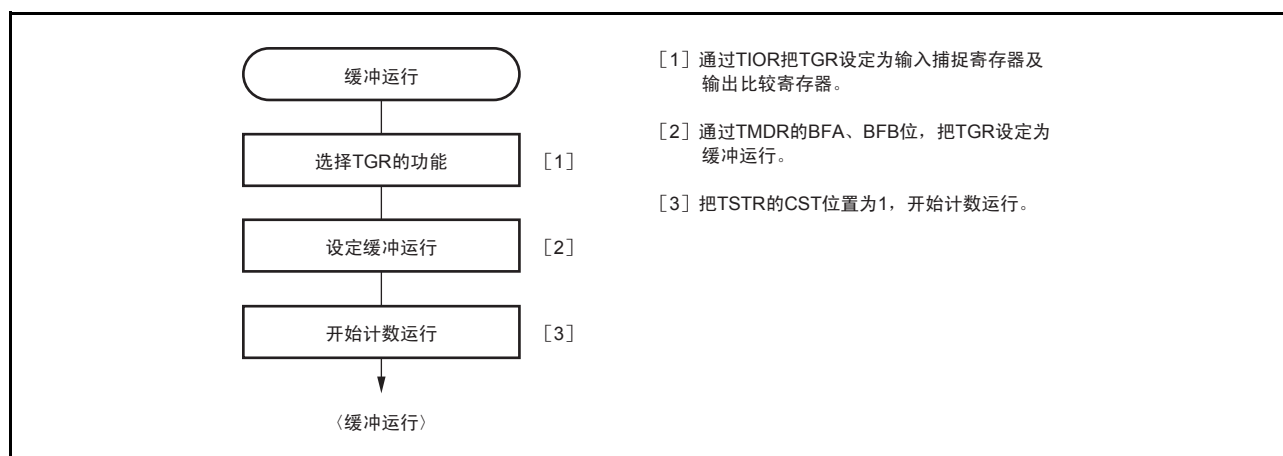


图 9.18 缓冲工作的设定顺序例子

9.5.3.2 缓冲操作例子

(1) TGR 作为输出比较寄存器

通道 0 设定为 PWM 模式 1，TGRA 和 TGRC 设定为缓冲操作时的例子如图 9.19 所示。通过比较匹配 B 设定 TCNT 清除、用比较匹配 A 输出 1、用比较匹配 B 输出 0 的例子。

因设定了缓冲运行，在比较匹配 A 输出产生变化的同时，缓冲寄存器 TGRC 的值传送到定时器通用寄存器 TGRA。此操作在每次产生比较匹配 A 时重复。

PWM 模式请参考 9.5.4 PWM 模式。

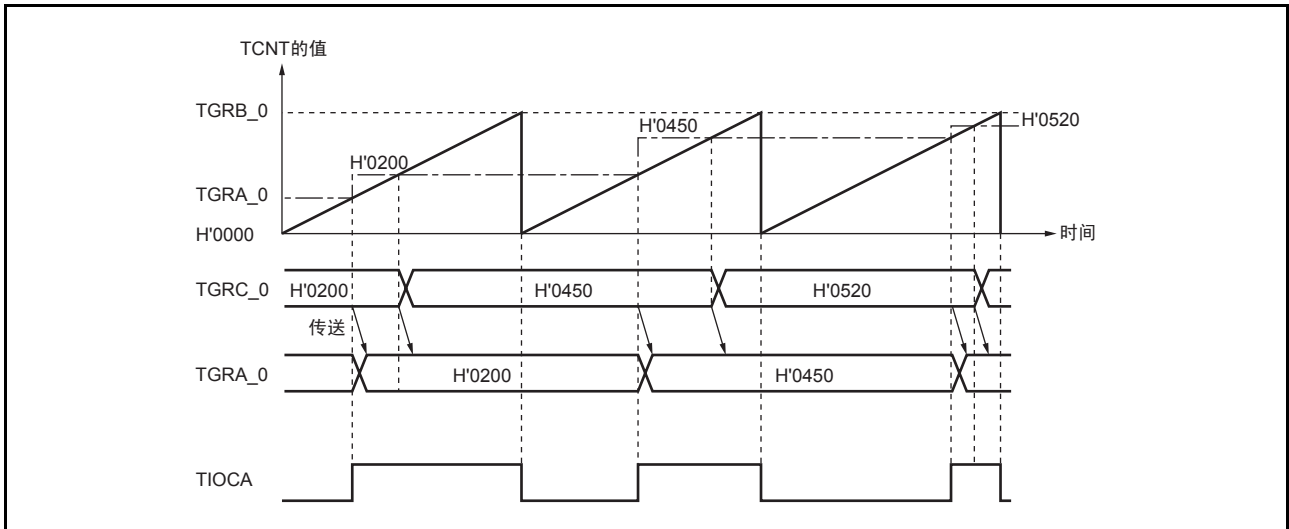


图 9.19 缓冲运行例子 (1)

(2) TGR 作为输入捕捉寄存器

TGRA 设定为输入捕捉寄存器，TGRA 和 TGRC 设定为缓冲操作时的例子如图 9.20 所示。

TCNT 通过 TGRA 的输入捕捉进行计数器清除，TIOCA 管脚的输入捕捉的输入沿从上升沿 / 下降沿的两沿选择。

因设定为缓冲操作，所以通过输入捕捉 A 将 TCNT 的值保存到 TGRA 的同时，先前保存在 TGRA 的值同时传送到 TGRC。

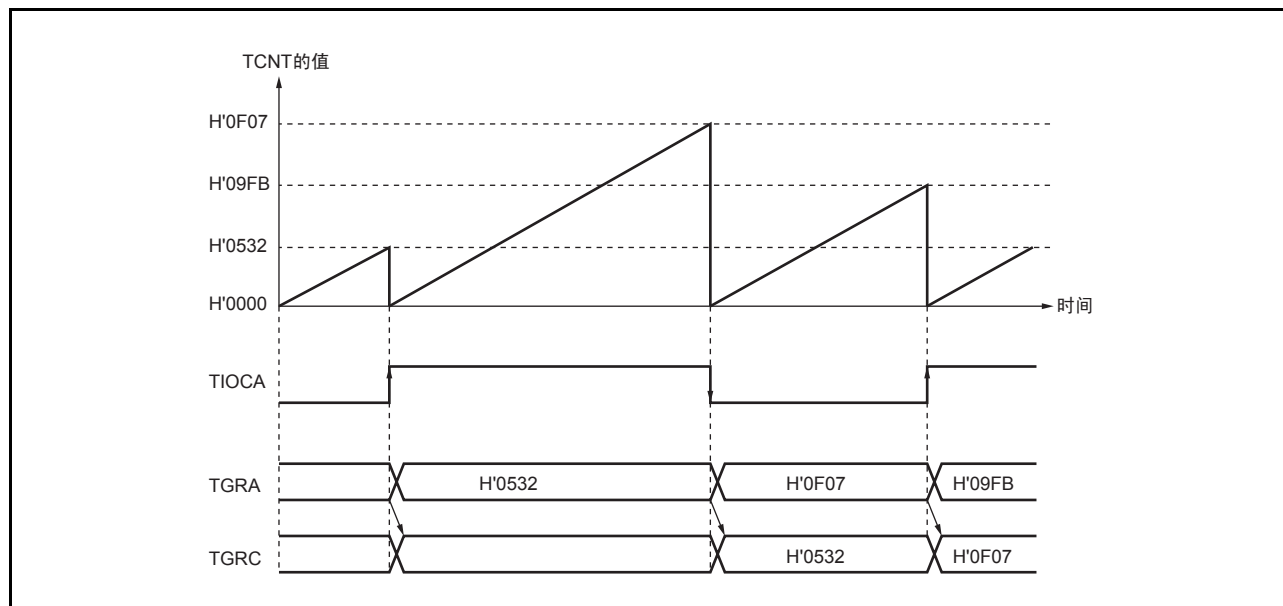


图 9.20 缓冲工作例子 (2)

9.5.4 PWM 模式

在 PWM 模式，通过输出管脚输出 PWM 波形。符合各 TGR 比较匹配的输出电平可从 0 输出 / 1 输出 / 交替输出中选择。

通过各 TGR 的设定，可输出占空比 0 ~ 100% 的 PWM 波形。

TGR 作为比较匹配的计数器清除源，可设定其寄存器的周期。全部通道可独立设定为 PWM 模式。也可同步运行。

PWM 模式如下所示有 2 种。

1. PWM 模式 1

TGRA 和 TGRB、TGRC 和 TGRD 成对使用，从 TIOCA、TIOCC 管脚生成 PWM 输出。从 TIOCA、TIOCC 管脚通过比较匹配 A、C，用 TIOR 的 IOA3 ~ IOA0、IOC3 ~ IOC0 位指定输出，且通过比较匹配 B、D，用 TIOR 的 IOB3 ~ IOB0、IOD3 ~ IOD0 位指定输出。初始输出值是 TGRA、TGRC 设定的值。成对使用 TGR 的设定值在两值相同时，即使产生比较匹配，输出值也不变化。在 PWM 模式 1，可进行最大 4 相的 PWM 输出。

2. PWM 模式 2

1 个 TGR 作为周期寄存器，其他的 TGR 作为占空比寄存器使用，生成 PWM 输出。用比较匹配进行 TIOR 指定的输出。而且，用同步寄存器的比较匹配进行计数器清除时，各管脚的输出值通过 TIOR 设定为初始值。周期寄存器和占空比寄存器的设定值一样时，即使产生比较匹配，输出值也不产生变化。

在 PWM 模式 2，通过共用同步操作可进行最大 7 相的 PWM 输出。

PWM 输出管脚和寄存器的对应如表 9.18 所示。

表 9.18 各 PWM 输出的寄存器和输出管脚

通道	寄存器	输出管脚	
		PWM 模式 1	PWM 模式 2
0	TGRA_0	TIOCA0	TIOCA0
	TGRB_0		TIOCB0
	TGRC_0	TIOCC0	TIOCC0
	TGRD_0		TIOCD0
1	TGRA_1	TIOCA1	TIOCA1
	TGRB_1		TIOCB1
2	TGRA_2	TIOCA2	TIOCA2
	TGRB_2		TIOCB2

【注】 PWM 模式 2 时，即使设定了周期，TGR 也不可能有 PWM 输出。

9.5.4.1 PWM 模式的设定顺序例子

PWM 模式的设定顺序例子如图 9.21 所示。

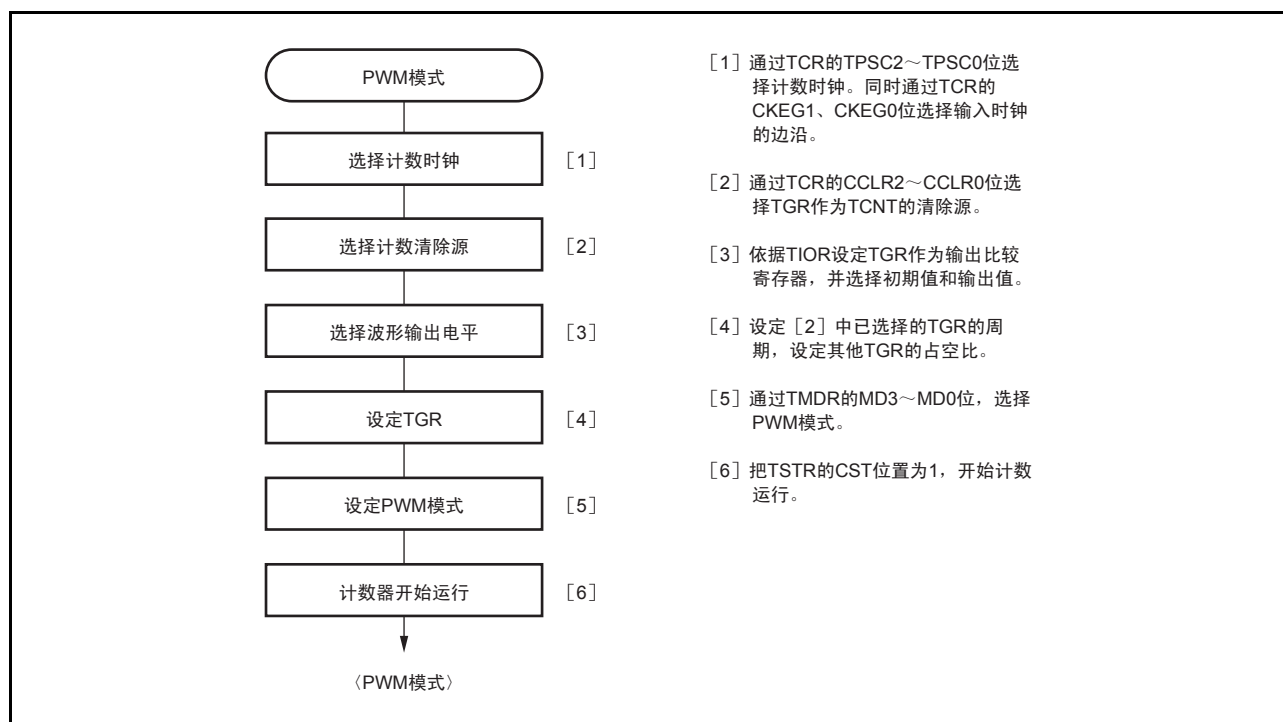


图 9.21 PWM 模式的设定顺序例子

9.5.4.2 PWM 模式操作例子

PWM 模式 1 操作例子如图 9.22 所示。

此图，将 TGRA 比较匹配作为 TCNT 的清除源，TGRA 的初始输出值和输出值设定为 0、TGRB 的输出值设定为 1 时的例子。

此时，TGRA 设定的值为周期，TGRB 设定的值为占空比。

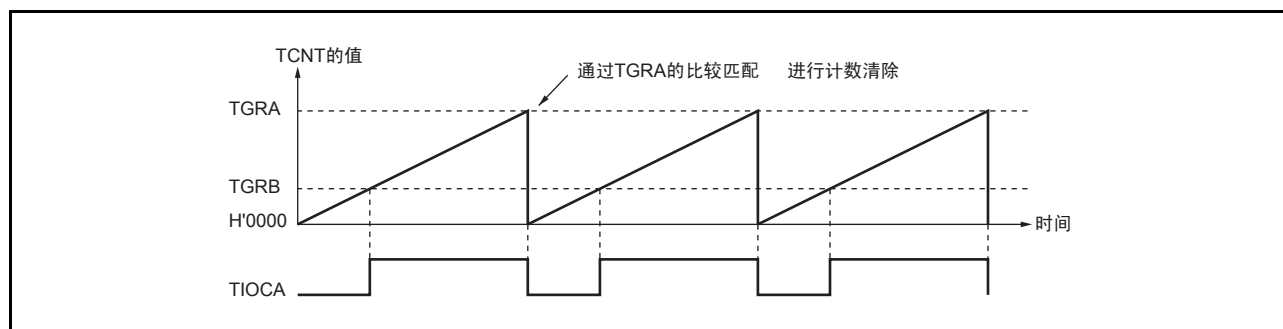


图 9.22 PWM 模式的工作例子 (1)

PWM 模式 2 操作例子如图 9.23 所示。

此图，通道 0 和 1 同步操作，TGRB_1 的比较匹配作为 TCNT 的清除源。其它 TGR (TGRA_0 ~ TGRD_0、TGRA_1) 的初始输出值设定为 0，输出值设定为 1，输出 5 相的 PWM 波形时的例子。

此时，TGRB_1 设定的值为周期，其它的 TGR 设定的值为占空比。

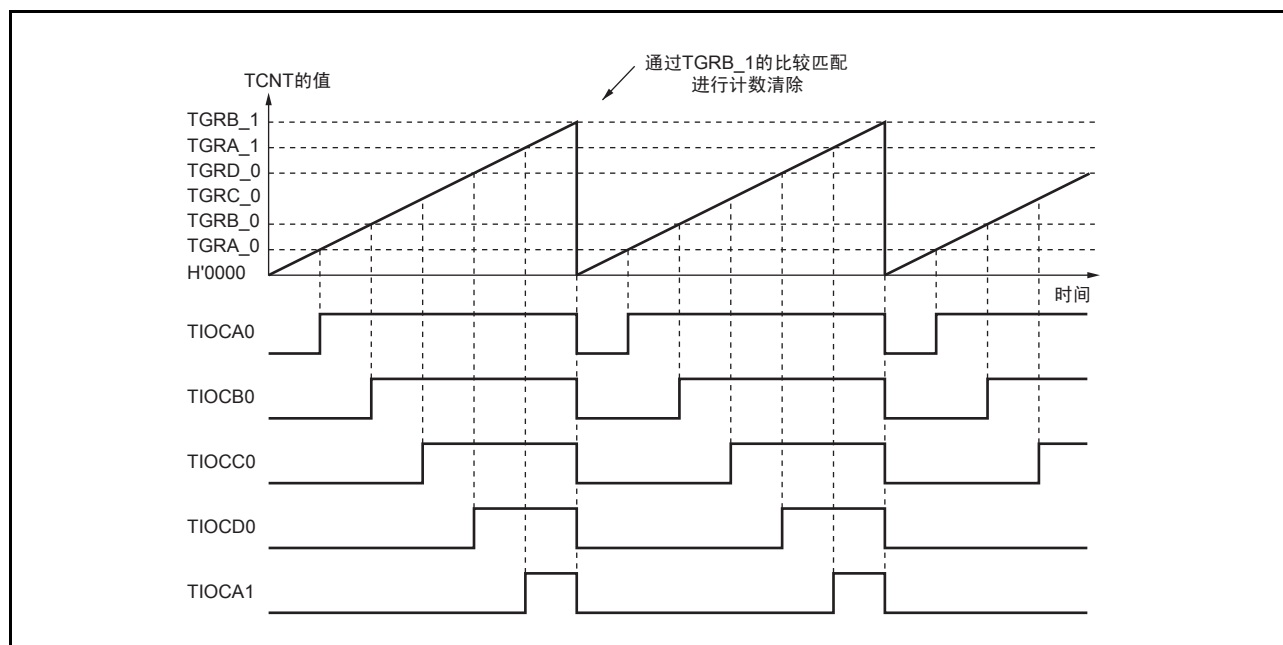


图 9.23 PWM 模式的工作例子 (2)

在 PWM 模式，输出占空比 0%、占空比 100% 的 PWM 波形的例子图 9.24 所示。

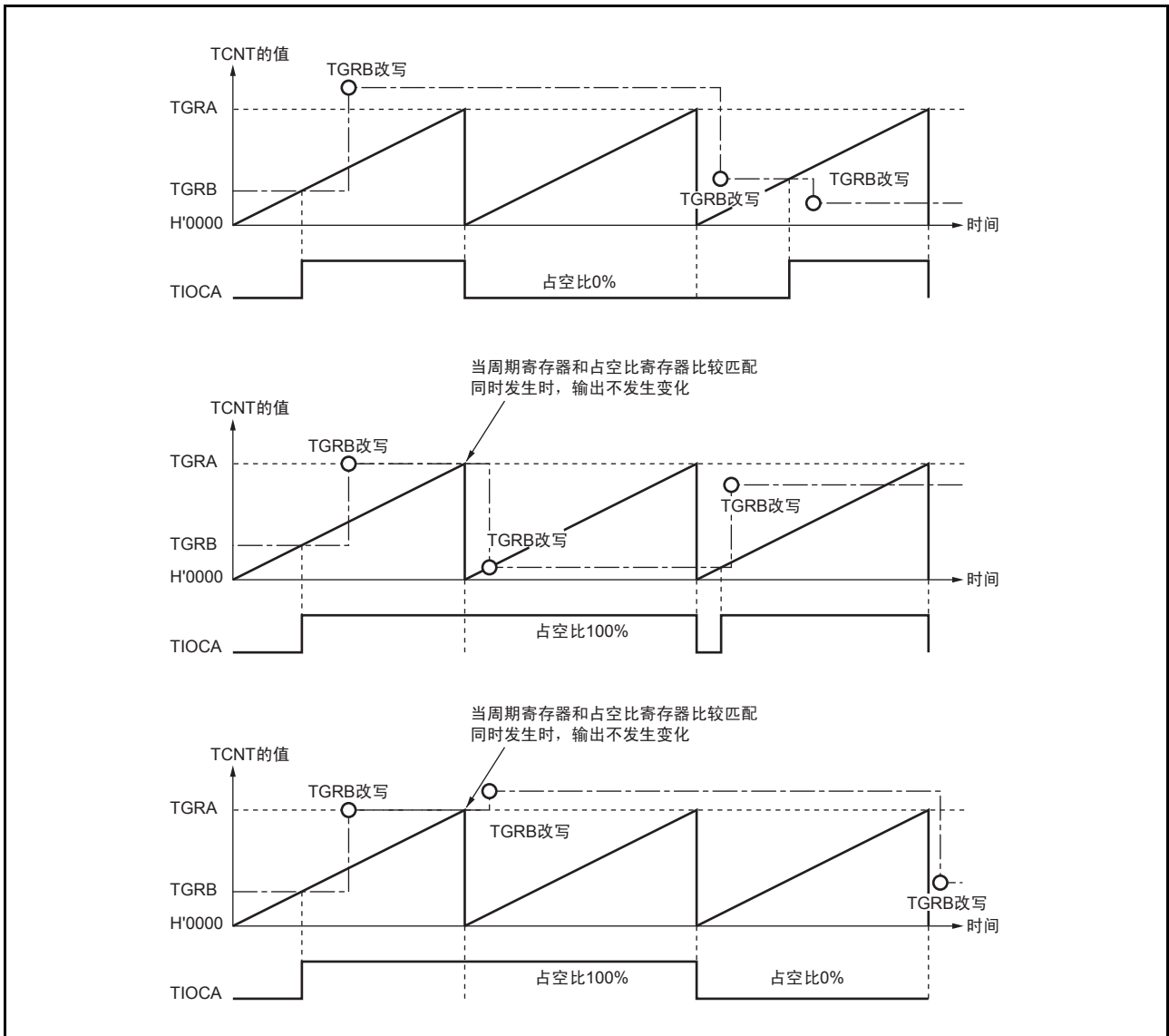


图 9.24 PWM 模式工作例子 (3)

9.5.5 相位计数模式

在相位计数模式，用通道 1、2 的设定，检测 2 个外部时钟的相位差，TCNT 进行递增 / 递减计数。

设定为相位计数模式时，无论 TCR 的 TPSC2 ~ TPSC0 位，CKEG1、CKEG0 位的设定，外部时钟选定为计数器输入时钟，TCNT 作为递增 / 递减计数器运行。但是，TCR 的 CCLR1、CCLR0 位、TIOR、TIER、TGR 的功能有效，所以可使用输入捕捉 / 比较匹配功能和中断功能。

TCNT 在递增计数时，如果产生上溢，将 TSR 的 TCFV 标志置位。而递减计数产生下溢时，TCFU 标志置位。

TSR 的 TCFD 位是计数方向标志。通过读取 TCFD 标志，可判定 TCNT 是递增计数或递减计数。

表 9.19 表示外部时钟管脚和通道的对应关系。

表 9.19 相位计数模式时钟输入管脚

通道	外部时钟管脚	
	A 相	B 相
通道 1 在相位计数模式时	TCLKA	TCLKB
通道 2 在相位计数模式时	TCLKC	TCLKD

9.5.5.1 相位计数模式的设定顺序例子

相位计数模式的设定顺序例子如图 9.25 所示。

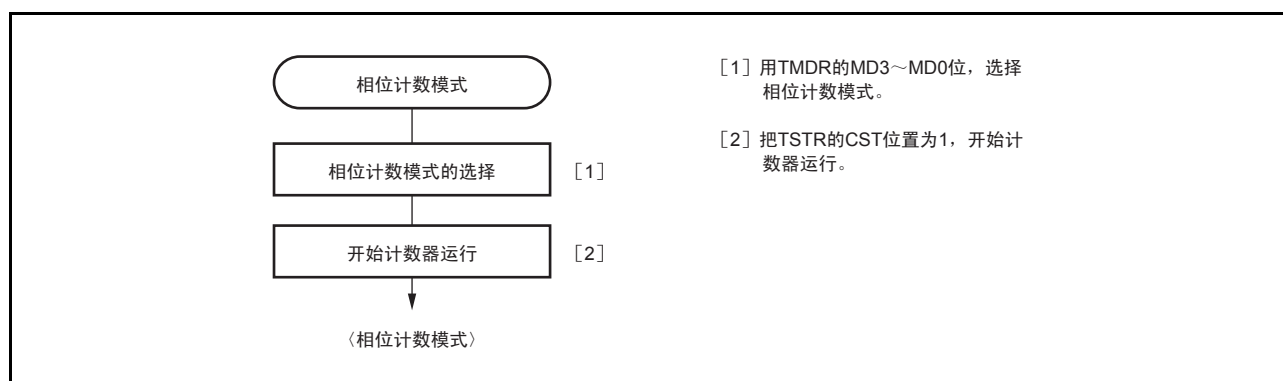


图 9.25 相位计数模式的设定顺序例子

9.5.5.2 相位计数模式操作例子

在相位计数模式，通过 2 个外部时钟的相位差，TCNT 进行递增 / 递减计数。而且，根据计数条件产生 4 种模式。

(1) 相位计数模式 1

相位计数模式 1 操作例子如图 9.26 所示，TCNT 的递增 / 递减计数条件如表 9.20 所示。

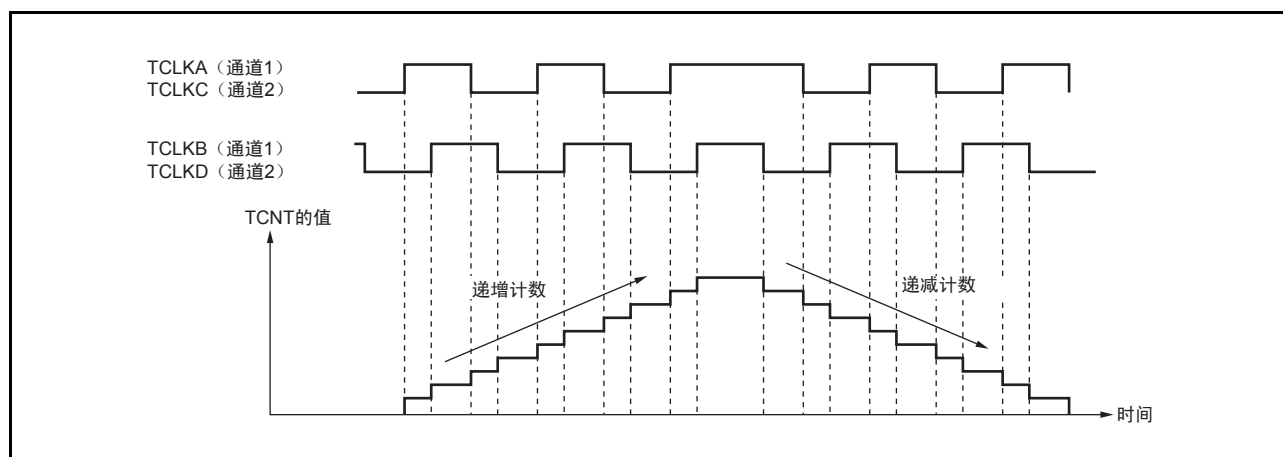


图 9.26 相位计数模式 1 的工作例子

表 9.20 相位计数模式 1 的递增 / 递减计数条件

TCLKA (通道 1) TCLKC (通道 2)	TCLKB (通道 1) TCLKD (通道 2)	工作内容
High 电平	↑	递增计数
Low 电平	↓	
↑	Low 电平	
↓	High 电平	
High 电平	↓	递减计数
Low 电平	↑	
↑	High 电平	
↓	Low 电平	

【符号说明】

↑ : 上升沿

↓ : 下降沿

(2) 相位计数模式 2

相位计数模式 2 操作例子如图 9.27 所示，TCNT 的递增 / 递减计数条件如表 9.21 所示。

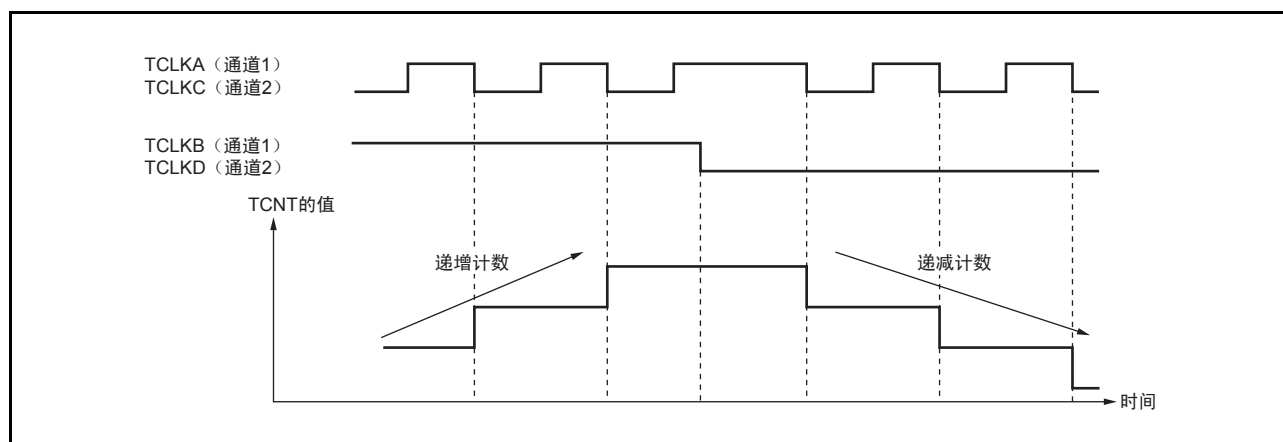


图 9.27 相位计数模式 2 的工作例子

表 9.21 相位计数模式 2 的递增 / 递减计数条件

TCLKA (通道 1) TCLKC (通道 2)	TCLKB (通道 1) TCLKD (通道 2)	工作内容
High 电平	↑	Don't care
Low 电平	↓	Don't care
↑	Low 电平	Don't care
↓	High 电平	递减计数
High 电平	↓	Don't care
Low 电平	↑	Don't care
↑	High 电平	Don't care
↓	Low 电平	递增计数

【符号说明】

↑：上升沿

↓：下降沿

(3) 相位计数模式 3

相位计数模式 3 操作例子如图 9.28 所示，TCNT 的递增 / 递减计数条件如表 9.22 所示。

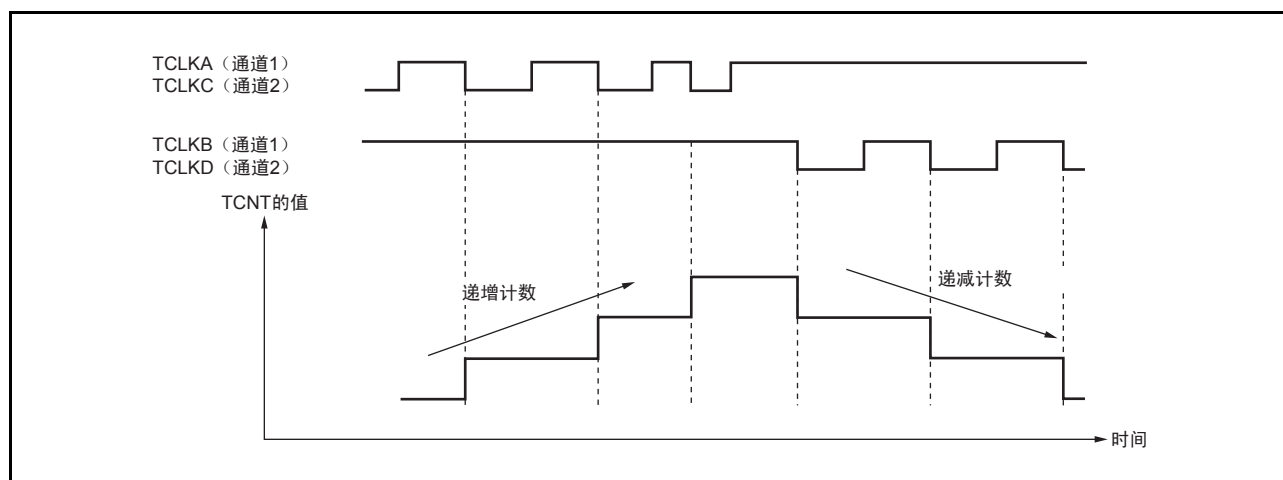


图 9.28 相位计数模式 3 的工作例子

表 9.22 相位计数模式 3 的递增 / 递减计数条件

TCLKA (通道 1) TCLKC (通道 2)	TCLKB (通道 1) TCLKD (通道 2)	工作内容
High 电平	↑	Don't care
Low 电平	↓	Don't care
↑	Low 电平	Don't care
↓	High 电平	递增计数
High 电平	↓	递减计数
Low 电平	↑	Don't care
↑	High 电平	Don't care
↓	Low 电平	Don't care

【符号说明】

↑: 上升沿

↓: 下降沿

(4) 相位计数模式 4

相位计数模式 4 操作例子如图 9.29 所示，TCNT 的递增 / 递减计数条件如表 9.23 所示。

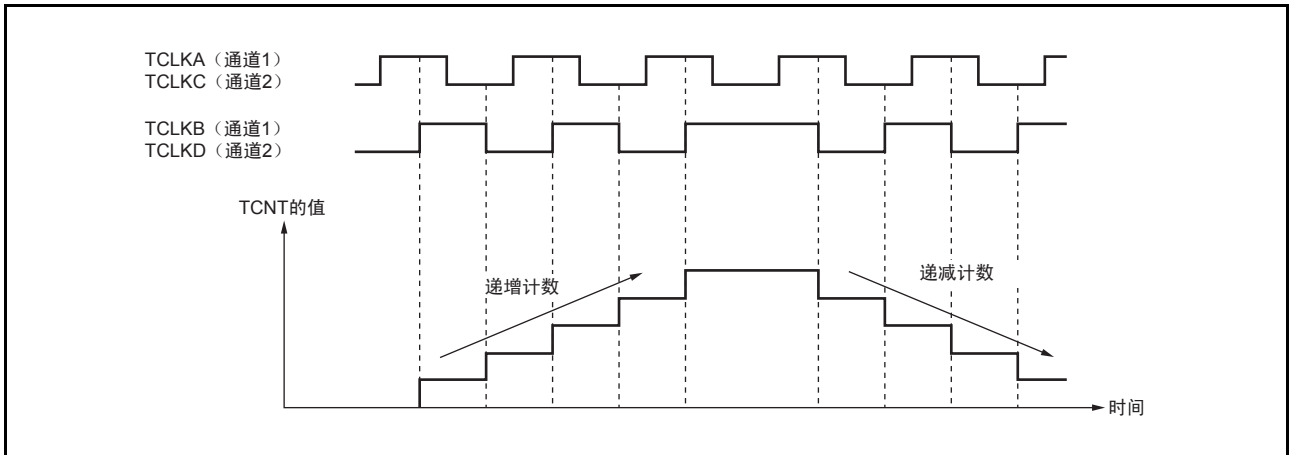


图 9.29 相位计数模式 4 的工作例子

表 9.23 相位计数模式 4 的递增 / 递减计数条件

TCLKA (通道 1) TCLKC (通道 2)	TCLKB (通道 1) TCLKD (通道 2)	工作内容
High 电平	↑	递增计数
Low 电平	↓	
↑	Low 电平	Don't care
↓	High 电平	
High 电平	↓	递减计数
Low 电平	↑	
↑	High 电平	Don't care
↓	Low 电平	

【符号说明】

- ↑ : 上升沿
- ↓ : 下降沿

9.6 中段源

9.6.1 中段源和优先顺序

TPU 的中断源有 TGR 的输入捕捉 / 比较匹配、TCNT 的上溢、下溢 3 种。各中断源因具有各自专用的状态标志和允许 / 禁止位，所以可独立的允许或禁止中断请求信号的产生。

产生中断源时，TSR 的对应状态标志置 1。此时，如果 TIER 对应的允许 / 禁止位置 1，则请求中断。状态标志通过清 0，解除中断请求。

通道间的优先权，用中断控制器可进行变更。通道内的优先权是固定的。详细的内容请参考 5. 中断控制器。

表 9.24 表示 TPU 的中断源一览。

表 9.24 TPU 中断一览

通道	名称	中断源	中断标志	DMAC 的启动	优先顺序*
0	TGI0A	TGRA_0 的输入捕捉 / 比较匹配	TGFA	可	高  低
	TGI0B	TGRB_0 的输入捕捉 / 比较匹配	TGFB	不可	
	TGI0C	TGRC_0 的输入捕捉 / 比较匹配	TGFC	不可	
	TGI0D	TGRD_0 的输入捕捉 / 比较匹配	TGFD	不可	
	TCI0V	TCNT_0 的上溢	TCFV	不可	
1	TGI1A	TGRA_1 的输入捕捉 / 比较匹配	TGFA	可	
	TGI1B	TGRB_1 的输入捕捉 / 比较匹配	TGFB	不可	
	TCI1V	TCNT_1 的上溢	TCFV	不可	
	TCI1U	TCNT_1 的下溢	TCFU	不可	
2	TGI2A	TGRA_2 的输入捕捉 / 比较匹配	TGFA	可	
	TGI2B	TGRB_2 的输入捕捉 / 比较匹配	TGFB	不可	
	TCI2V	TCNT_2 的上溢	TCFV	不可	
	TCI2U	TCNT_2 的下溢	TCFU	不可	

【注】 * 表示复位后的初始状态。通过中断控制器可进行通道间优先权的变更。

9.6.1.1 输入捕捉 / 比较匹配中断

通过各通道 TGR 的输入捕捉 / 比较匹配的产生，在 TSR 的 TGF 标志置 1 时，TIER 的 TGIE 位如果置 1，则请求中断。TGF 标志通过清 0 解除中断请求。TPU 的通道 0 各有 4 个，通道 1、2 各有 2 个，共计有 8 个输入捕捉 / 比较匹配中断。

9.6.1.2 上溢中断

通过产生各通道 TCNT 上溢，在 TSR 的 TCFV 标志置 1 时，如果 TIER 的 TCIEV 位置 1，则请求中断。TCFV 标志通过清 0，解除中断请求。TPU 的各通道有 1 个，共计有 3 个上溢中断。

9.6.1.3 下溢中断

通过产生各通道 TCNT 下溢，在 TSR 的 TCFU 标志置 1 时，如果 TIER 的 TCIEU 位置 1，则请求中断。TCFU 标志通过清 0，解除中断请求。TPU 通道 1、2 各有 1 个，共计有 2 个下溢中断。

9.6.2 DMAC 的启动

通过各通道 TGRA 的输入捕捉 / 比较匹配中断可启动 DMAC。详细请参考 7. DMA 控制器 (DMAC)。TPU 各通道有 1 个。共计有 3 个 TGRA 的输入捕捉 / 比较匹配中断可作为 DMAC 的启动源。

9.6.3 A/D 转换器的启动

通过各通道 TGRA 的输入捕捉 / 比较匹配可启动 A/D 转换器。

通过 TGRA 的输入捕捉 / 比较匹配的产生, TSR 的 TGFA 标志置 1 时, 如果 TIER 的 TTGE 位置 1, 则向 A/D 转换器请求开始 A/D 转换。此时在 A/D 转换器, 如果选择 TPU 的转换开始触发, 则开始 A/D 转换。

TPU 的各通道有 1 个。共计有 3 个 TGRA 输入捕捉 / 比较匹配中断, 可作为 A/D 转换器的转换开始源使用。

9.7 工作时序

9.7.1 输入 / 输出时序

9.7.1.1 TCNT 的计数时序

内部时钟运行时，TCNT 的计数时序如图 9.30 所示。而且，外部时钟运行时，TCNT 的计数时序如图 9.31 所示。

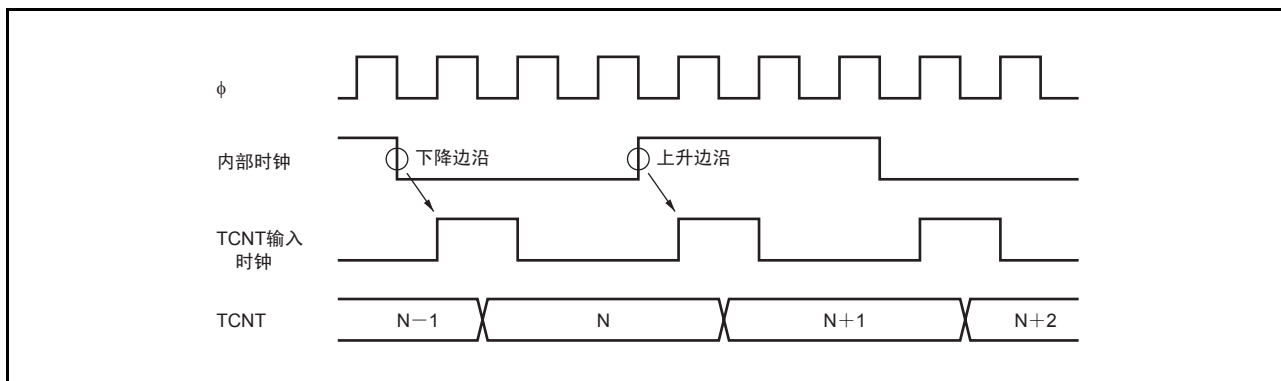


图 9.30 内部时钟工作时的计数时序

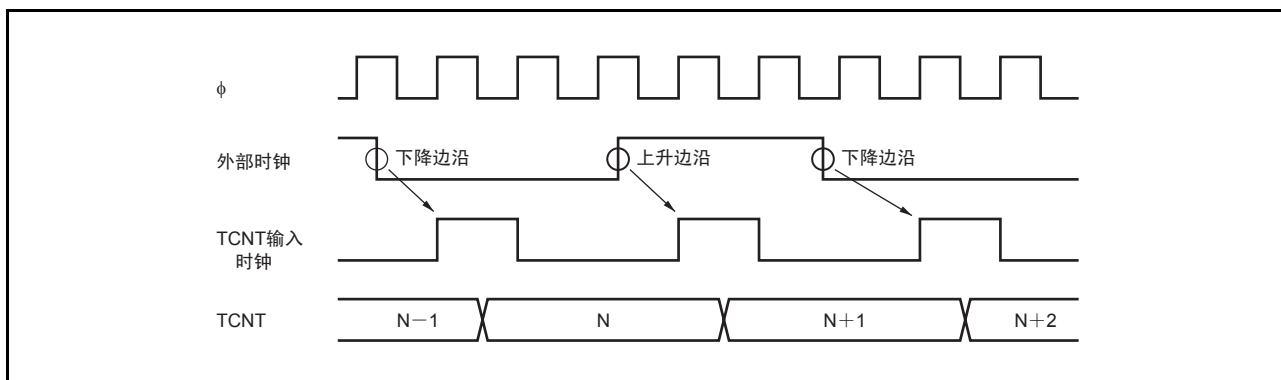


图 9.31 外部时钟工作时的计数时序

9.7.1.2 输出比较输出时序

在 TCNT 和 TGR 一致的最后状态（更新 TCNT 一致的计数值的时序）产生比较匹配信号。产生比较匹配信号时，TIOR 设定的输出值从输出比较管脚（TIOC 管脚）输出。TCNT 和 TGR 一致后，到产生 TCNT 输入时钟为止，不产生比较匹配信号。

比较匹配输出时序如图 9.32 所示。

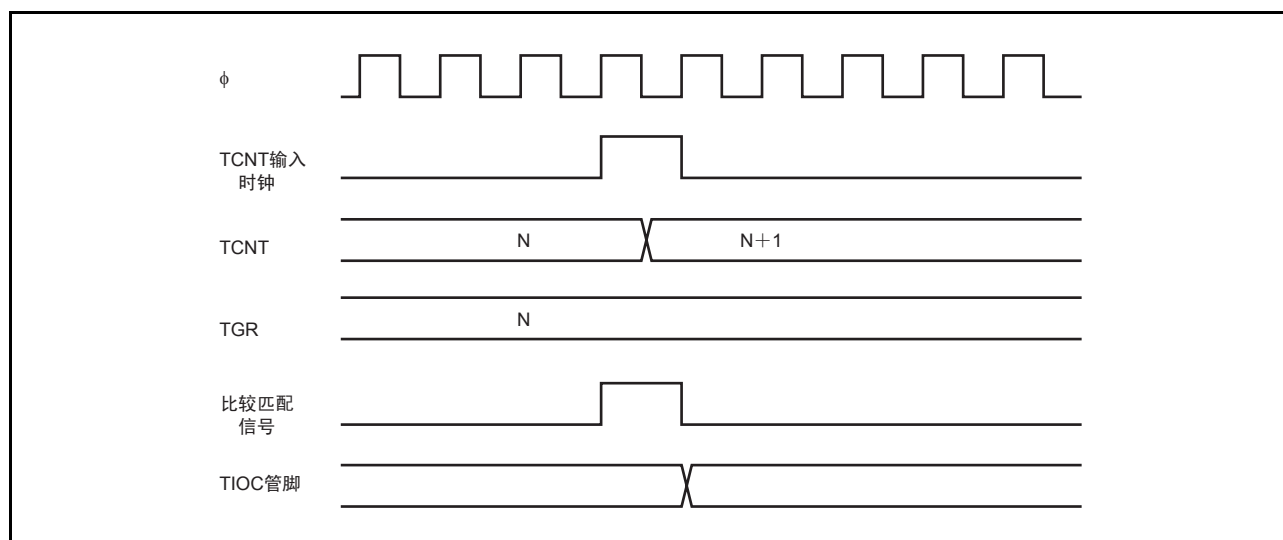


图 9.32 输出比较输出时序

9.7.1.3 输入捕捉信号时序

输入捕捉时序如图 9.33 所示。

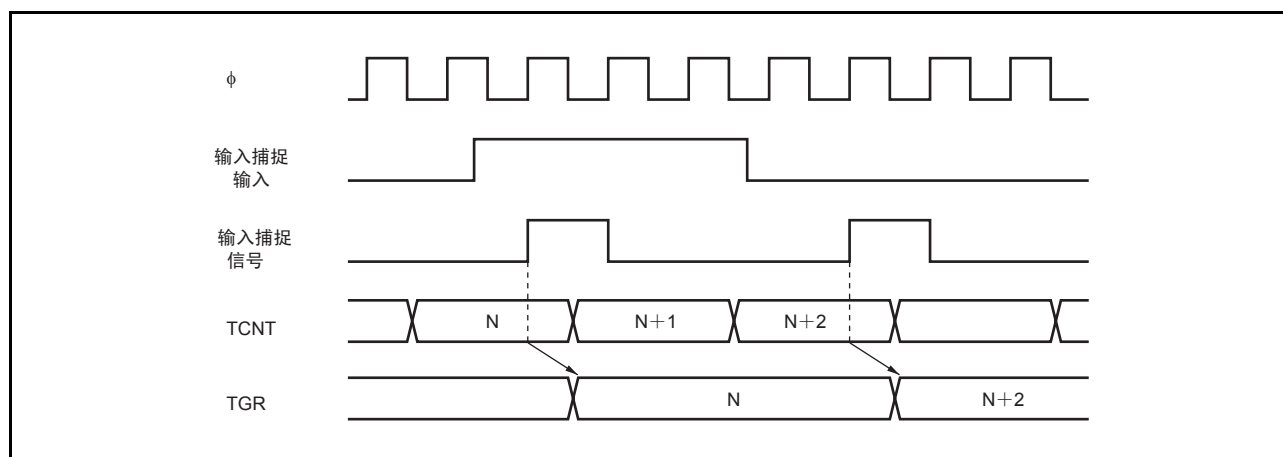


图 9.33 输入捕捉输入信号时序

9.7.1.4 比较匹配 / 输入捕捉的计数器清除时序

通过比较匹配的产生指定计数器清除时的时序如图 9.34 所示。

通过输入捕捉的产生指定计数器清除时的时序如图 9.35 所示。

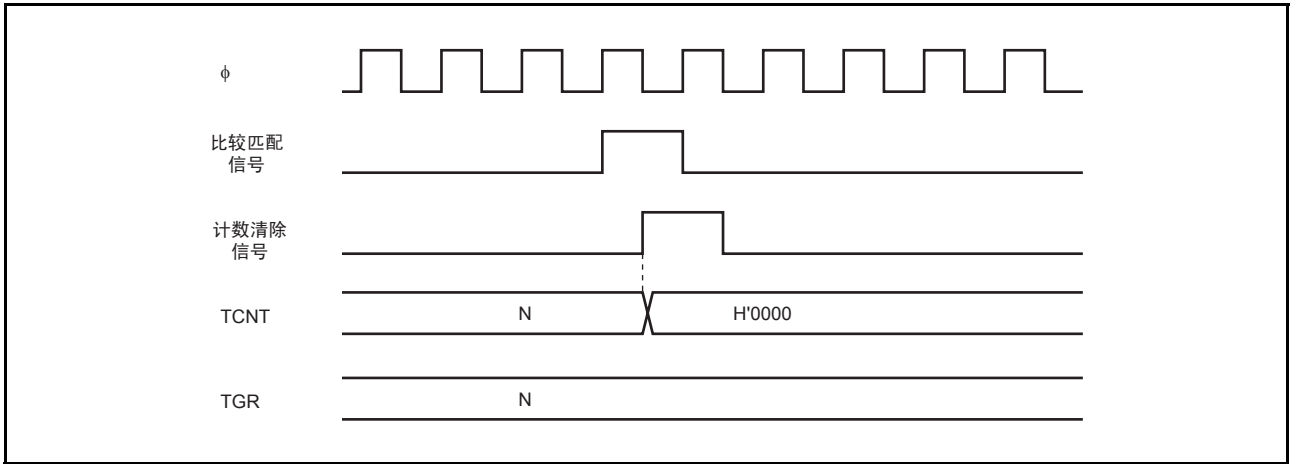


图 9.34 计数器清除时序 (比较匹配)

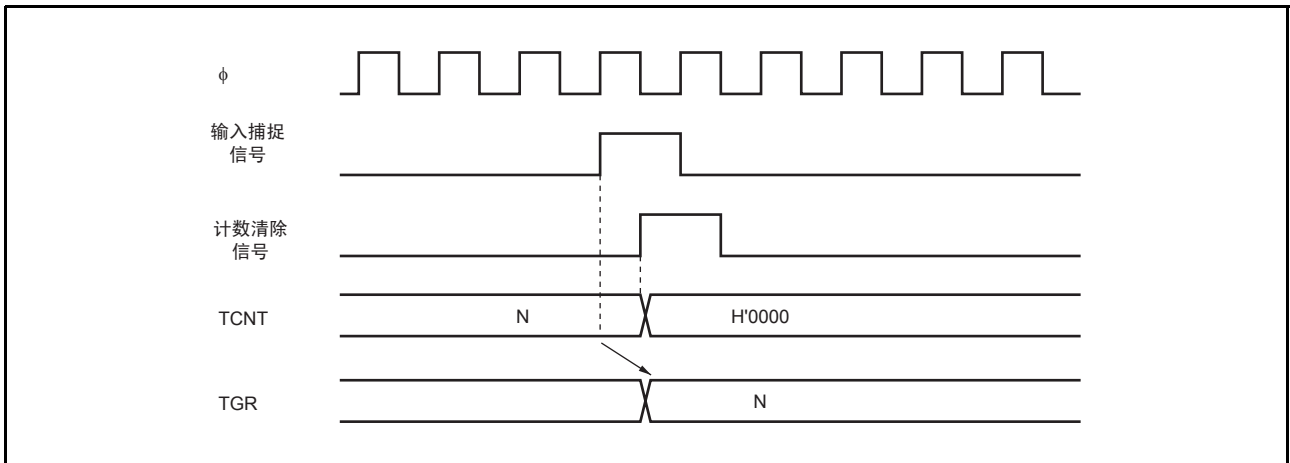


图 9.35 计数器清除时序 (输入捕捉)

9.7.1.5 缓冲运行时序

缓冲运行时的时序如图 9.36、图 9.37 所示

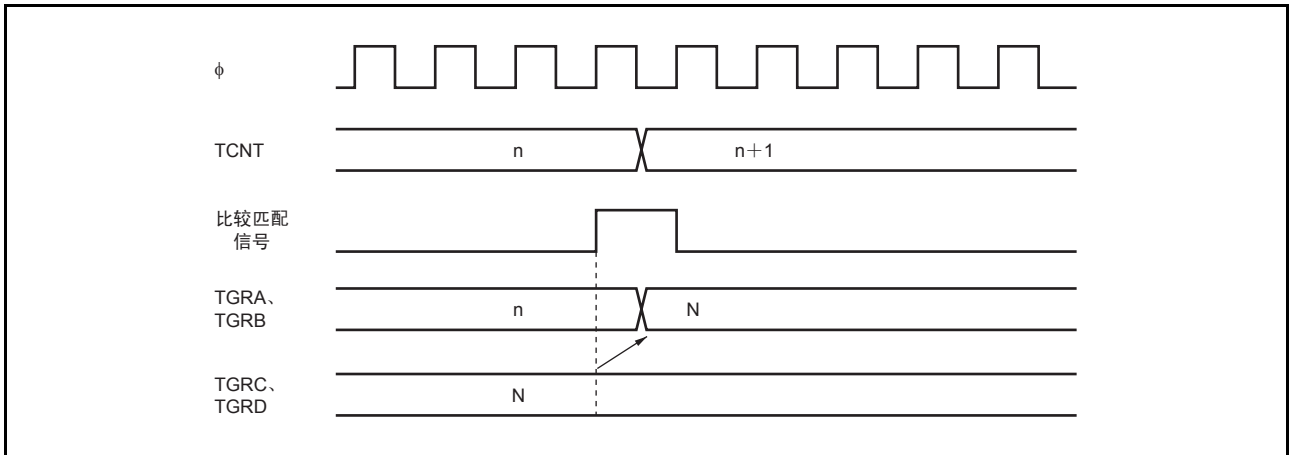


图 9.36 缓冲运行时序 (比较匹配)

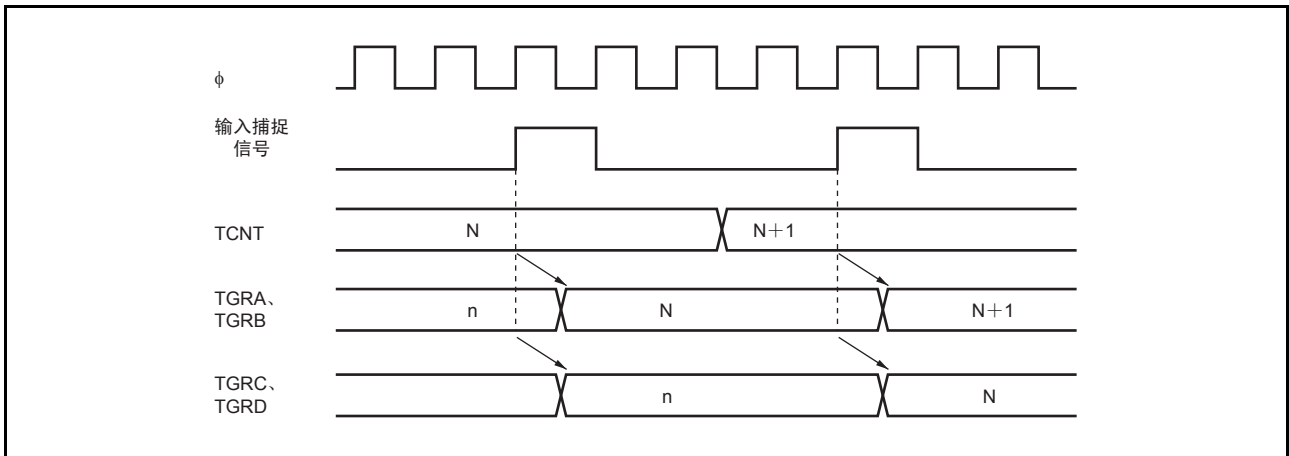


图 9.37 缓冲运行时序 (输入捕捉)

9.7.2 中断信号时序

9.7.2.1 比较匹配时 TGF 标志置位时序

比较匹配产生 TSR 的 TGF 标志的置位时序和 TGI 中断请求信号的时序如图 9.38 所示。

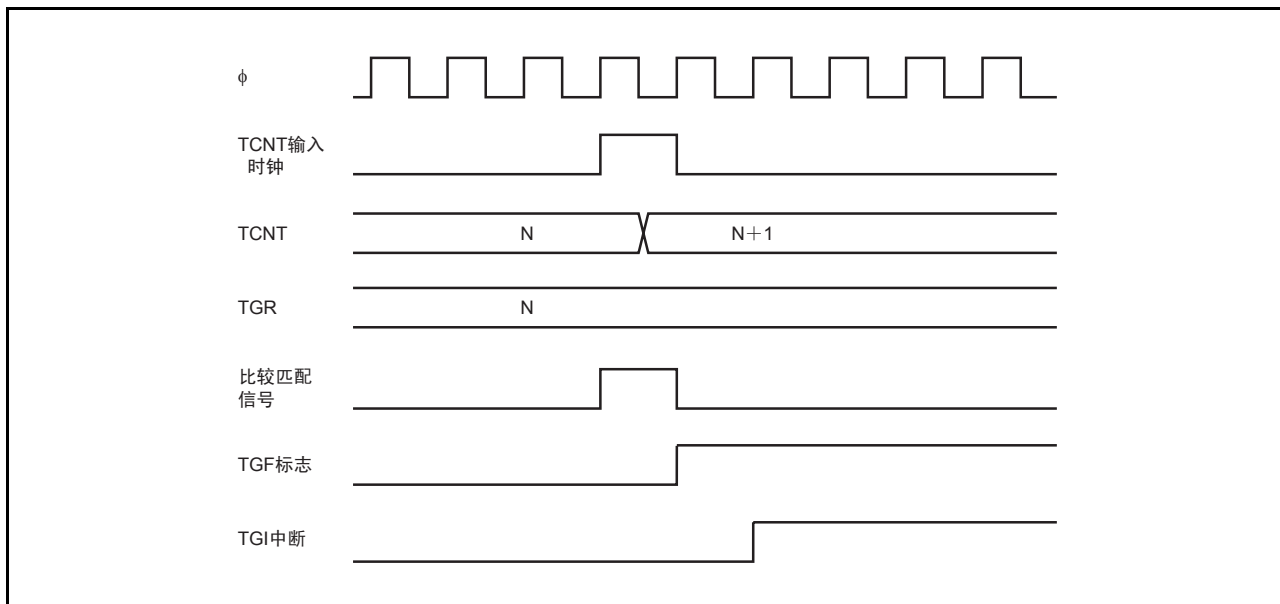


图 9.38 TGI 中断时序 (比较匹配)

9.7.2.2 输入捕捉时 TGF 标志置位时序

输入捕捉产生 TSR 的 TGF 标志置位时序和 TGI 中断请求信号的时序如图 9.39 所示。

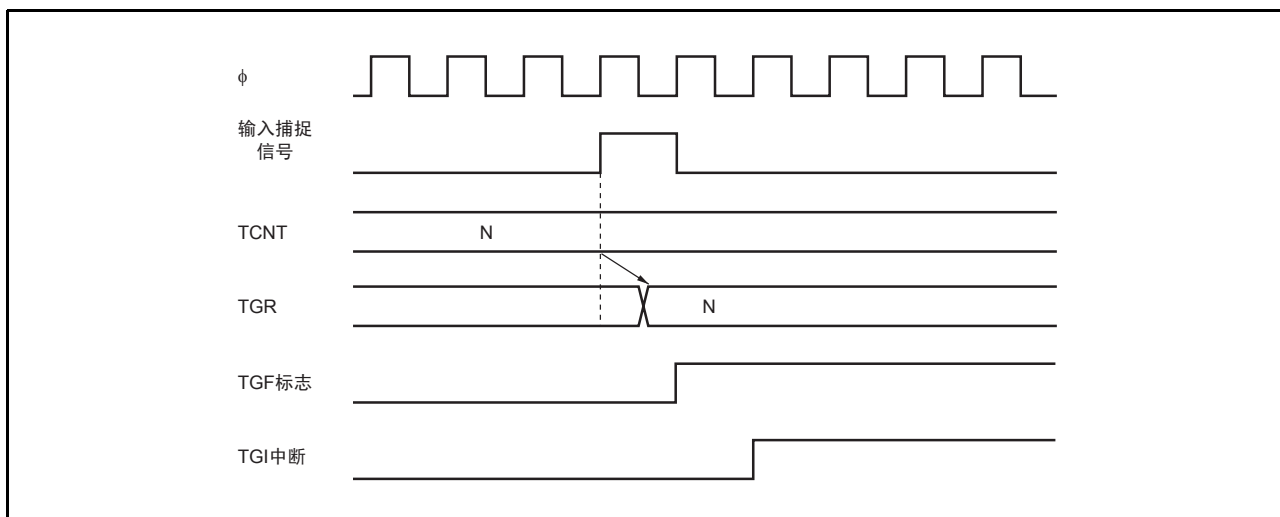


图 9.39 TGI 中断时序 (输入捕捉)

9.7.2.3 TCFV 标志 /TCFU 标志的置位时序

通过产生上溢，TSR 的 TCFV 标志置位时序和 TCIV 中断请求信号时序如图 9.40 所示。

通过产生下溢，TSR 的 TCFU 标志的置位时序和 TCIU 中断请求信号时序如图 9.41 所示。

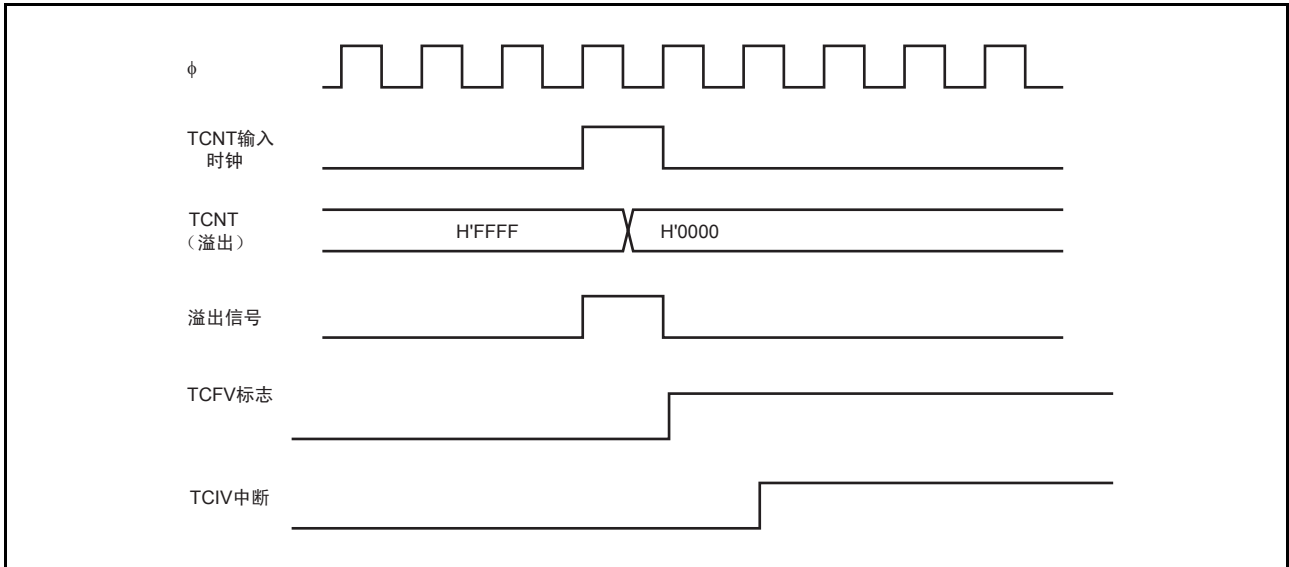


图 9.40 TCIV 中断的置位时序

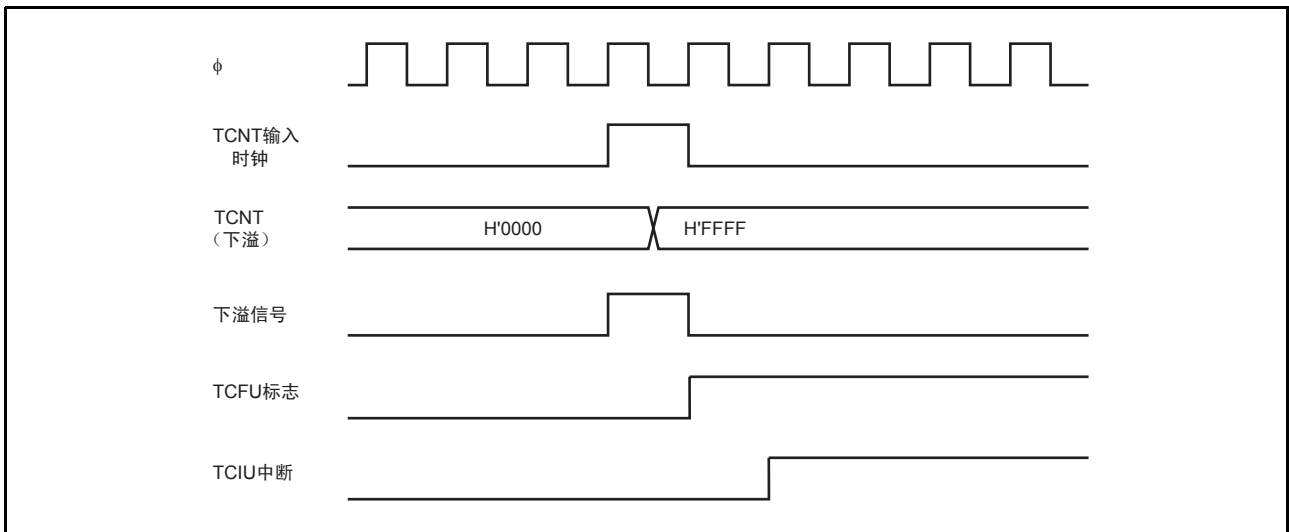


图 9.41 TCIU 中断的置位时序

9.7.2.4 状态标志清除时序

CPU 在状态标志读取 1 的状态后，如果写 0，则状态标志清除。启动 DMAC 时，状态标志也可自动清除。用 CPU 的状态标志清除时序如图 9.42 所示，用 DMAC 状态标志进行清除的时序如图 9.43 所示。

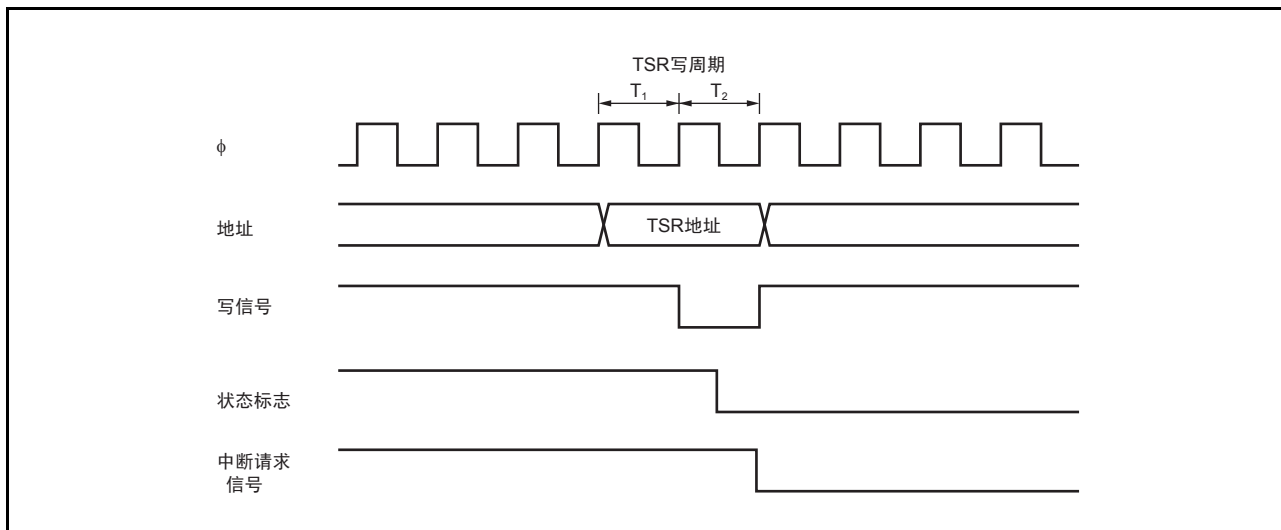


图 9.42 通过 CPU 的状态标志清除时序

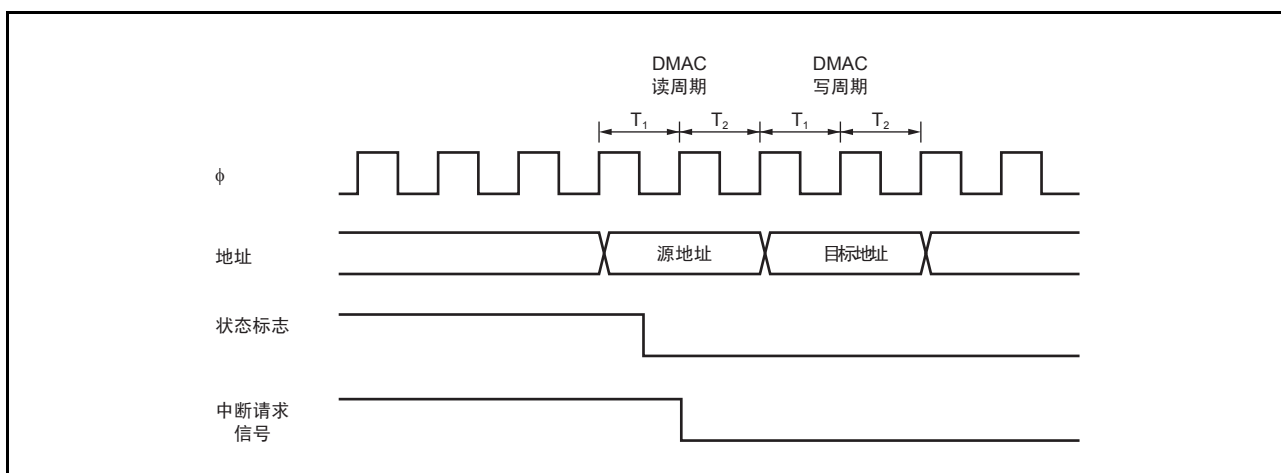


图 9.43 通过 DMAC 的启动，状态标志的清除时序

9.8 使用时的注意事项

9.8.1 输入时钟的限制事项

输入时钟的脉冲宽度在单沿检测时，必须为 1.5 状态时钟以上，两沿检测时必须为 2.5 状态以上。在这以下的脉冲宽度不能正常运行，请注意。

在相位计数模式时，2 个输入时钟的相位差及重叠，必须分别在 1.5 状态以上，脉冲宽度必须在 2.5 状态以上。相位计数模式的输入时钟条件如图 9.44 所示。

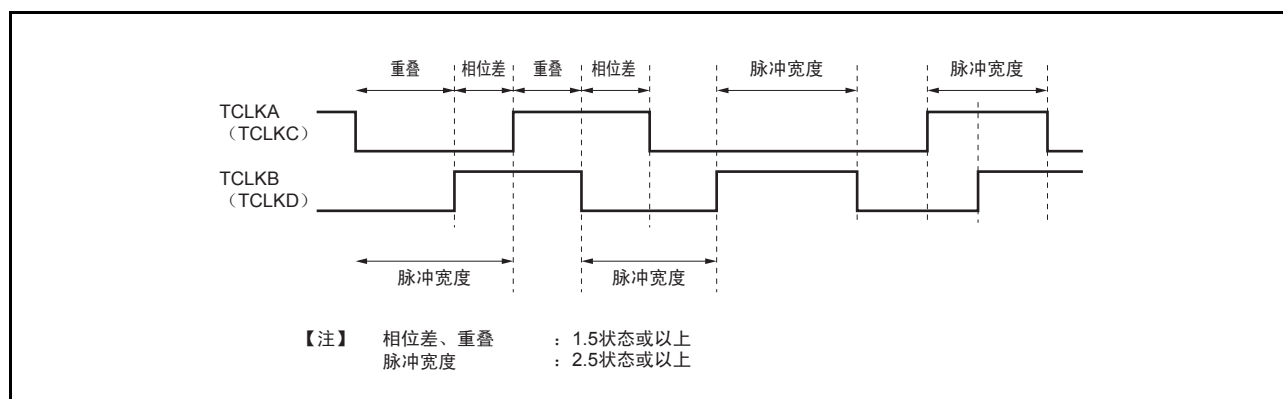


图 9.44 相位计数模式时的相位差、重叠、及脉冲宽

9.8.2 周期设定的注意事项

用比较匹配设定计数器清除时，TCNT 在与 TGR 的值一致的最后状态（当计数值与 TCNT 更新时一致的时候）清除。因此，实际的计数器的频率如下所示。

$$f = \frac{\phi}{(N+1)}$$

f : 计数器频率

ϕ : 运行频率

N : TGR 的设定值

9.8.3 TCNT 的写与清除的竞争

若 TCNT 写周期中的 T2 状态产生计数器清除信号，则优先 TCNT 清除，且不向 TCNT 写入。此时序如图 9.45 所示。

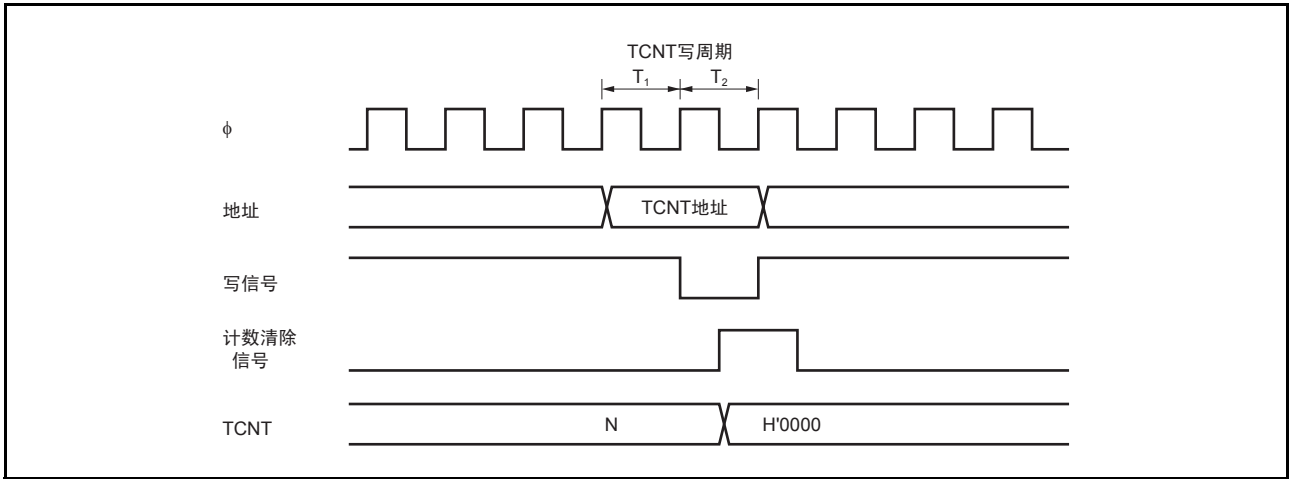


图 9.45 TCNT 的写和清除的竞争

9.8.4 TCNT 的写与累加计数的竞争

若 TCNT 写周期中 T2 状态产生累加计数，则优先向 TCNT 写入，且不进行累加计数。此时序如图 9.46 所示。

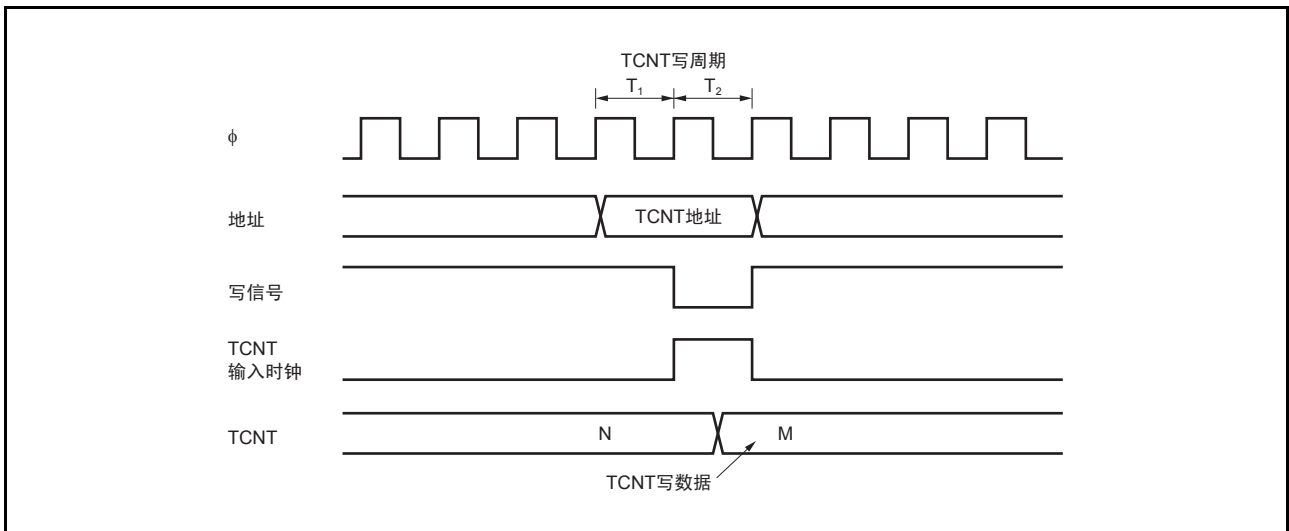


图 9.46 TCNT 的写入和累加计数的竞争

9.8.5 TGR 的写入与比较匹配的竞争

若 TGR 写周期中 T2 状态产生比较匹配，则优先向 TGR 写入，禁止比较匹配信号。即使写入与前一次相同的值，也不产生比较匹配。此时序如图 9.47 所示。

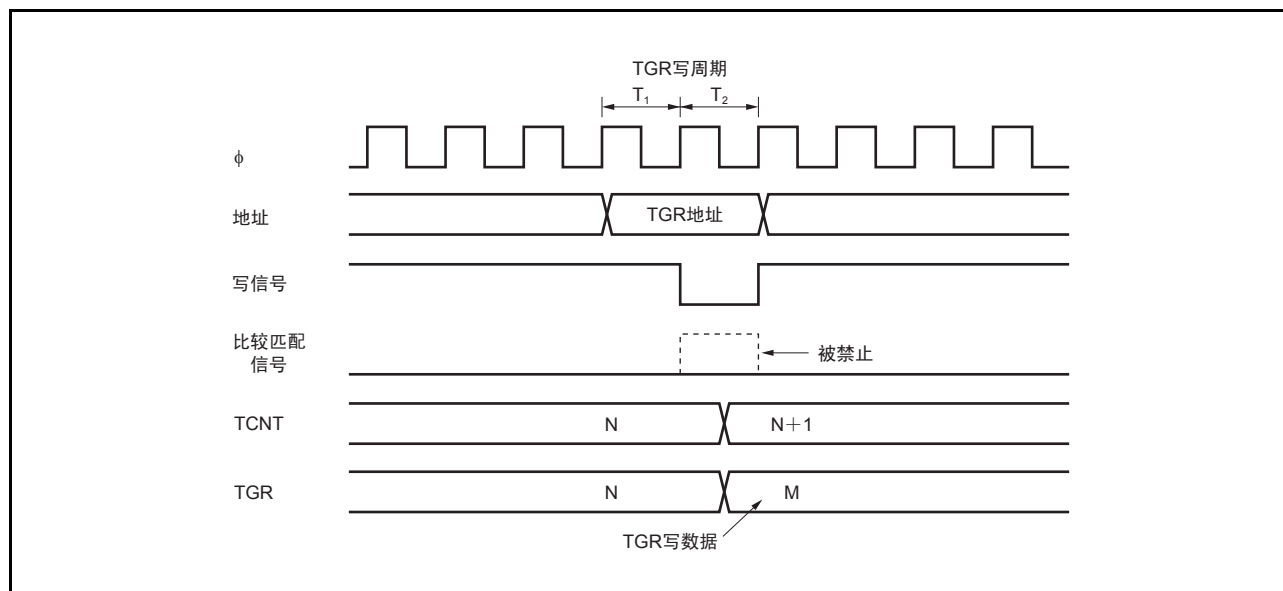


图 9.47 TGR 写和比较匹配的竞争

9.8.6 缓冲寄存器的写与比较匹配的竞争

若 TGR 写周期中 T2 状态产生比较匹配，则用缓冲运行向 TGR 传送的数据成为预先写数据。此时序如图 9.48 所示。

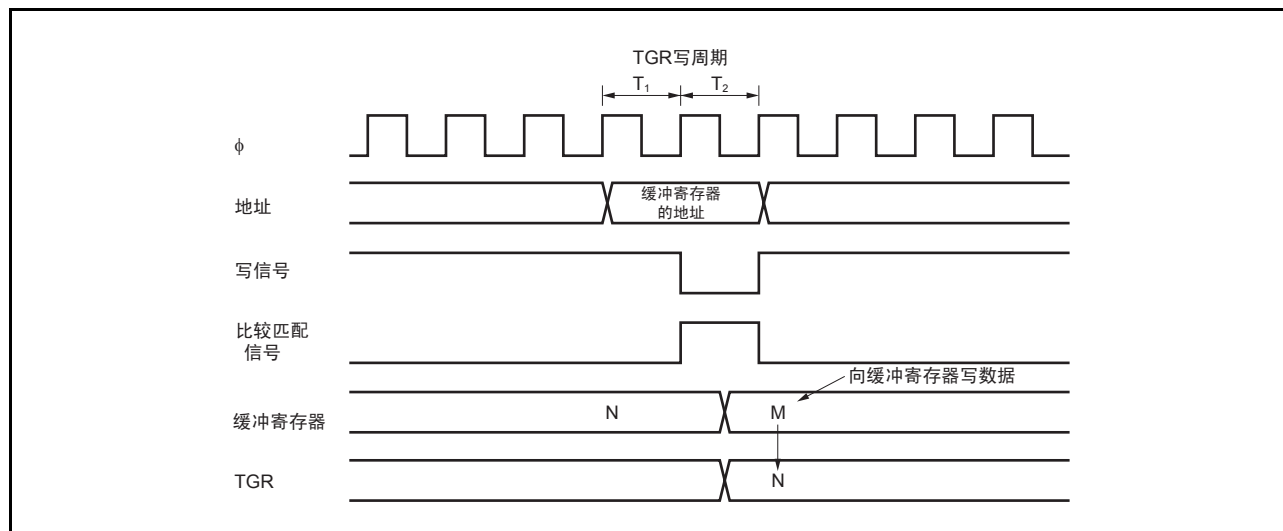


图 9.48 缓冲寄存器的写和比较匹配的竞争

9.8.7 TGR 的读与输入捕捉的竞争

若 TGR 读周期中 T1 状态产生输入捕捉信号，则读取的数据成为输入捕捉传送后的数据。此时序如图 9.49 所示。

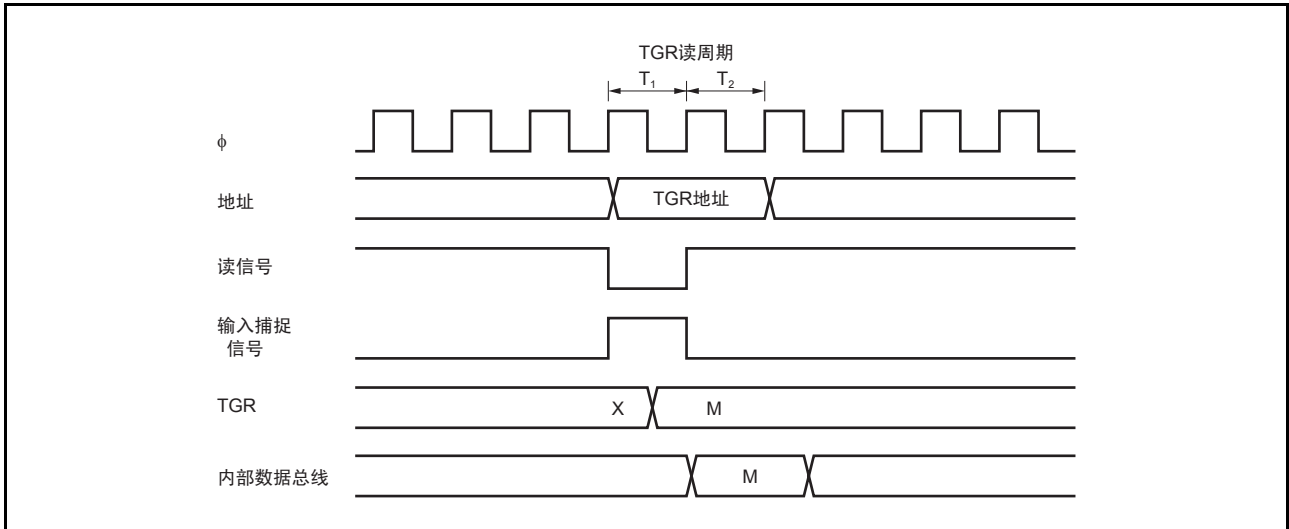


图 9.49 TGR 的写和输入捕捉的竞争

9.8.8 TGR 的写与输入捕捉的竞争

若 TGR 写周期中 T2 状态产生输入捕捉，则优先输入捕捉，且不向 TGR 写入。此时序如图 9.50 所示。

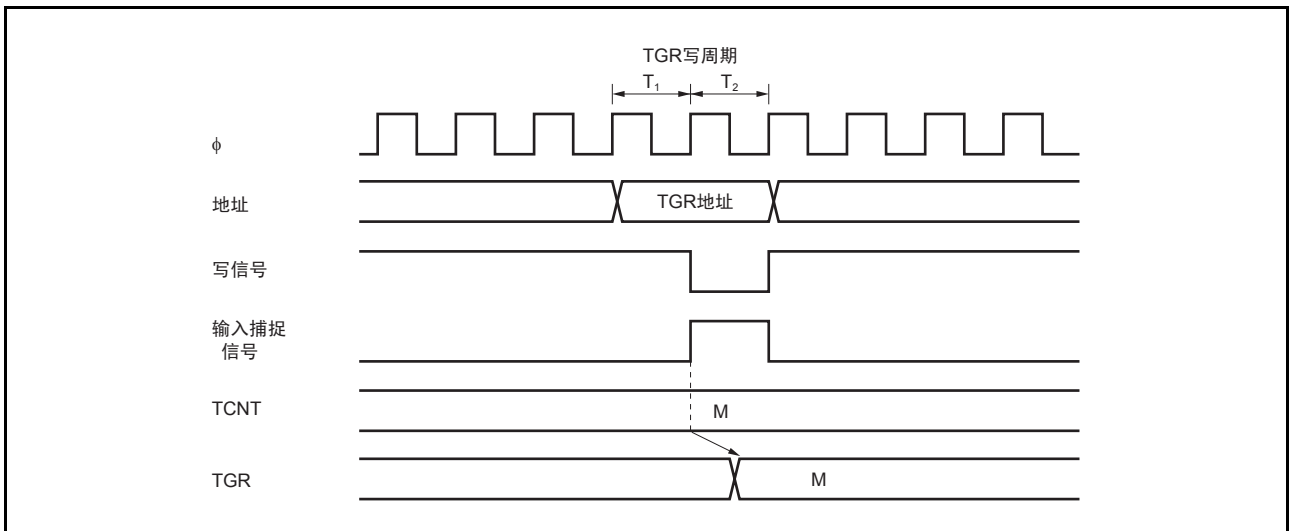


图 9.50 TGR 的写和输入捕捉的竞争

9.8.9 缓冲寄存器的写与输入捕捉的竞争

若缓冲寄存器写周期中 T2 状态产生输入捕捉信号，则优先缓冲运行，且不向缓冲寄存器写入。此时序如图 9.51 所示。

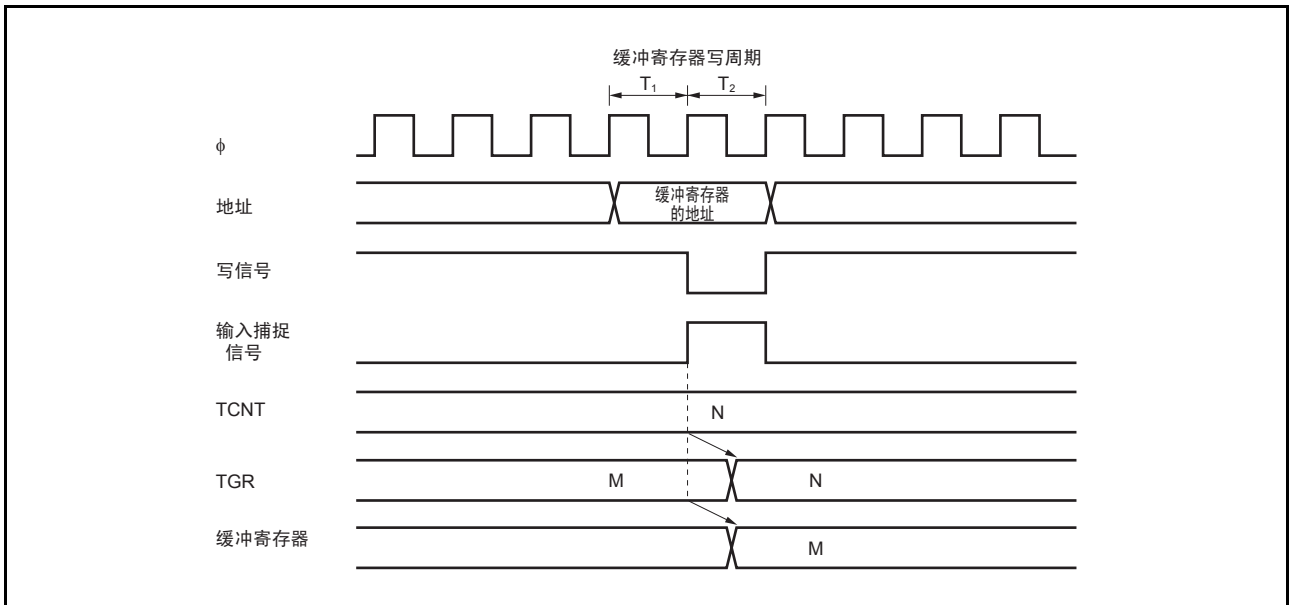


图 9.51 缓冲寄存器的写和输入捕捉的竞争

9.8.10 上溢 / 下溢与计数器清除的竞争

若上溢 / 下溢与计数器清除同时产生，则 TSR 的 TCFV/TCFU 标志不置位，优先 TCNT 的清除。作为 TGR 比较匹配的清除源，向 TGR 设定 H'FFFF 时的运行时序如图 9.52 上溢和计数器清除的竞争所示。

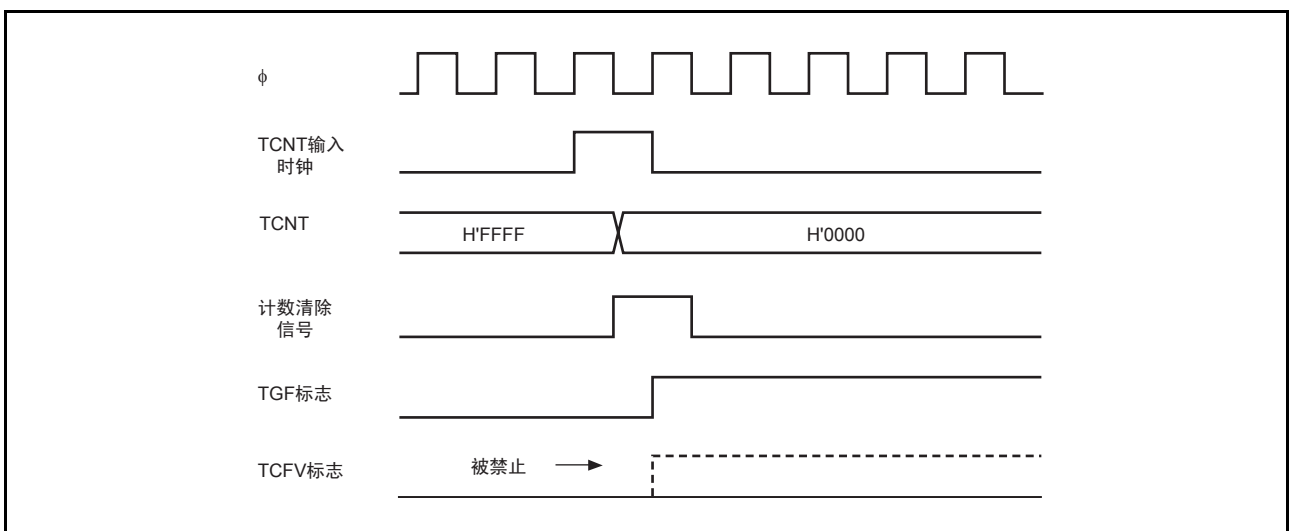


图 9.52 上溢和计数器清除的竞争

9.8.11 TCNT 的写与上溢 / 下溢的竞争

若 TCNT 写周期中 T2 状态产生累加计数 / 递减计数，即使产生上溢 / 下溢，优先向 TCNT 写入，TSR 的 TCFV/TCFU 标志不置位。

TCNT 的写与上溢竞争时的运行时序如图 9.53 所示。

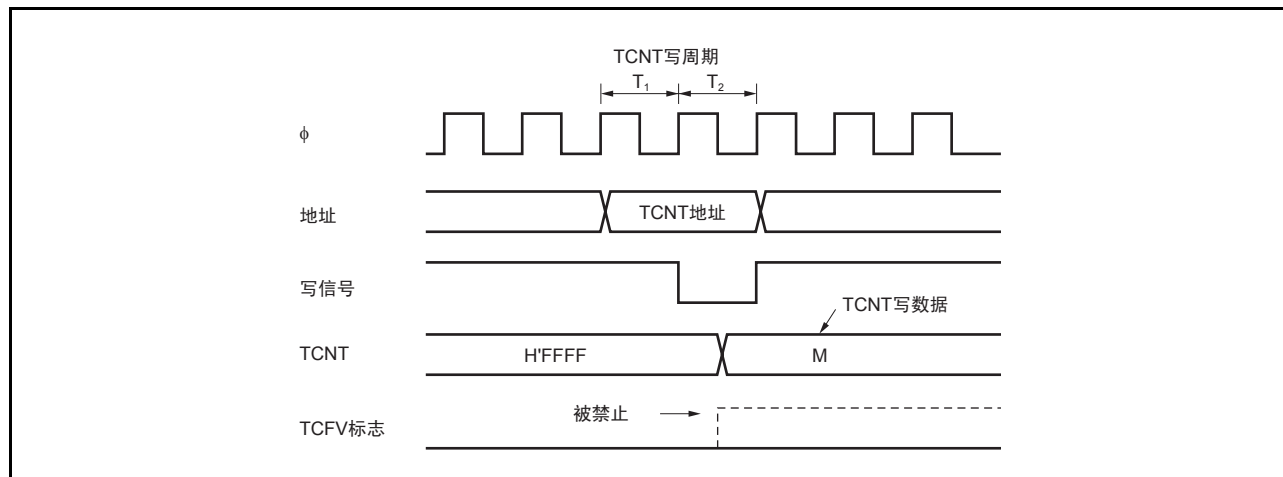


图 9.53 TCNT 的写和上溢的竞争

9.8.12 输入 / 输出管脚的兼用

在本 LSI，TCLKA 输入和 TIOCC0 输入 / 输出，TCLKB 输入和 TIOCD0 输入 / 输出，TCLKC 输入和 TIOCB1 输入 / 输出，TCLKD 输入和 TIOCB2 输入 / 输出的管脚都互相兼用。输入外部时钟时，从兼用管脚不进行比较匹配输出。

9.8.13 模块停止时的中断

若请求中断状态设定为模块停止模式，不能清除 CPU 的中断源或 DMAC 的启动源，禁止中断后才进入模块停止模式。

9.8.14 模块停止模式的设定

可通过模块停止控制寄存器，设定 TPU 的运行禁止 / 允许。为初始值时，TPU 的运行停止。通过解除模块停止模式，可进行寄存器的存取。详细内容请参考 20. 低功耗状态。

10. 看门狗定时器 (WDT)

看门狗定时器 (WDT) 是 8 位的定时器，因系统的失控而不改写计数器的值，产生上溢后在本 LSI 内部能进行置位。

不作为看门狗定时器时，可作为间隔定时器使用。作为间隔定时器时，计数器在每次上溢时产生间隔定时器中断。

WDT 的框图如图 10.1 所示。

10.1 特点

- 可选择 8 种计数器输入时钟
- 可切换看门狗定时器模式和间隔定时器模式

看门狗定时器模式

- 计数器产生上溢后，可选择是否在本 LSI 内部置位。

间隔定时器模式

- 计数器产生上溢后，产生间隔定时器中断 (WOVI)

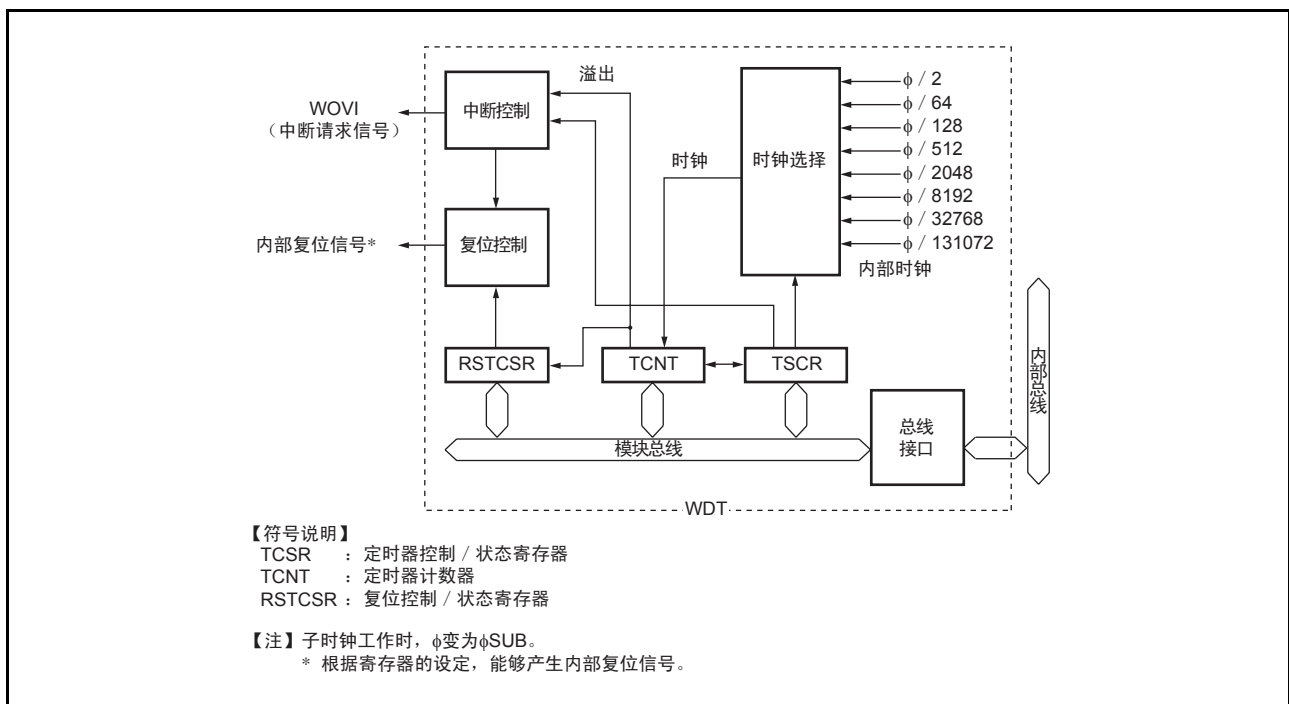


图 10.1 WDT 的框图

10.2 寄存器的说明

WDT 有以下寄存器。为了 TCSR、TCNT、RSTCSR 不容易被改写，写入的方法与一般的寄存器有所不同。详细请参考 10.5.1 寄存器存取时的注意事项。

- 定时计数器 (TCNT)
- 定时控制/状态寄存器 (TCSR)
- 复位控制/状态寄存器 (RSTCSR)

10.2.1 定时计数器 (TCNT)

TCNT 可读 / 写 8 位递增计数器。TCNT 在定时器控制 / 状态寄存器 (TCSR) 的 TME 位为 0 时，初始化为 H'00。

10.2.2 定时器控制 / 状态寄存器 (TCSR)

TCSR 进行向 TCNT 输入时钟、模式等的选择。

位	位名	初始值	R/W	说明
7	OVF	0	R/(W)*	上溢标志 表示 TCNT 上溢。 [置位条件] TCNT 上溢 (H'FF → H'00) 时 但是, 在看门狗定时器模式, 选择内部复位请求时, 置位后, 通过内部置位自动清除。 [清除条件] 在 OVF = 1 状态, 读取 TCSR 后, 向 OVF 写入 0 时 禁止间隔定时器中断, 查询 OVF 时, 读取 OVF = 1 状态 2 次以上。
6	WT/IT	0	R/W	定时器模式选择 选择作为看门狗定时器使用或间接定时器使用。 0: 间接定时器模式 1: 看门狗定时器模式
5	TME	0	R/W	定时器允许 将此位置 1 后 TCNT 开始计数。清除后 TCNT 停止计数运行, 初始化为 H'00。
4、3	—	全为 1	—	保留位 读后通常读出 1。不能被修改。
2 1 0	CKS2 CKS1 CKS0	0 0 0	R/W R/W R/W	时钟选择 2 ~ 0 选择向 TCNT 输入时钟。标志 () 内 $\phi = 16\text{MHz}$ 时的上溢 周期。溢出周期为从 TCNT=H'00 开始累加计数到发生溢出的 时间。 000: 时钟 /2 (周期 32.0 μs) 001: 时钟 /64 (周期 1.0ms) 010: 时钟 /128 (周期 2.0ms) 011: 时钟 /512 (周期 8.2ms) 100: 时钟 /2048 (周期 32.8ms) 101: 时钟 /8192 (周期 131.1ms) 110: 时钟 /32768 (周期 524.3ms) 111: 时钟 /131072 (周期 2.1s)

【注】 * 为清除标志仅能写 0。

10.2.3 置位控制 / 状态寄存器 (RSTCSR)

RSTCSR 通过 TCNT 的上溢控制内部复位信号的产生, 选择内部复位信号的种类。通过 $\overline{\text{RES}}$ 管脚的复位信号将 RSTCSR 初始化为 H'1F。不能用 WDT 上溢产生的内部复位信号进行初始化。

位	位名	初始值	R/W	说明
7	WOVF	0	R/(W)*1	看门狗定时器上溢标志 在看门狗定时器模式, TCNT 上溢后置位。在间隔定时器模式不被复位。 [置位条件] 在看门狗定时器模式, TCNT 上溢 (H'FF → H'00) 时 [清除条件] 在 1 的状态, 读取 RSTCSR 后, 向 WOVF 写 0 时
6	RSTE	0	R/W	复位允许 在看门狗定时器模式, 通过 TCNT 的上溢选择是否在 LSI 内部复位。 0: 即使 TCNT 上溢, 在内部不被复位 (LSI 内部不复位, 但是, WDT 内部的 TCNT、TCSR 复位。) 1: TCNT 上溢后内部复位
5	RSTS	0	R/W	置位选择 在看门狗定时器模式 TCNT 产生上溢, 选择内部置位的种类。 0: 加电复位 *2 1: 手动复位 *3
4~0	—	全为 1	—	保留位 读后通常读出 1。写定义为无效。

【注】 *1 为清除标志, 仅能写 0。

*2 用 WDT 的加电复位, USB 的寄存器不被初始化, 请注意。详细请参考 14.8.8 关于复位。

*3 仅支持 H8S/2218 群。

10.3 工作说明

10.3.1 看门狗定时器模式

作为看门狗定时器模式使用时，请设定 TCSR 的 $\overline{WT/IT}$ 位 = 1、TME 位 = 1。

TCNT 上溢前必须将 TCNT 的值改写（正常写 H'00），为不产生上溢请编程。因此，在系统正常工作期间，不产生 TCNT 的上溢。

RSTCSR 的 RSTE 位置 1 后，作为看门狗定时器工作时，因系统的失控而不能改写 TCNT 的值，上溢时，产生将本 LSI 内部复位的信号。复位通过 RSTCSR 的 RSTS 位设定，可选择加电复位或手动复位。内部复位信号连续输出 518 个状态。

从 \overline{RES} 管脚的输入信号进行复位和通过 WDT 上溢同时产生时，通过 \overline{RES} 管脚优先复位，RSTCSR 的 WOVF 位清 0。

在看门狗定时器模式，TCNT 上溢后，RSTCSR 的 WOVF 位置 1。而且，RSTCSR 的 RSTE 位置 1 后，在 TCNT 上溢时，本 LSI 全部产生内部复位信号。

【注】 * 仅支持 H8S/2218 群。

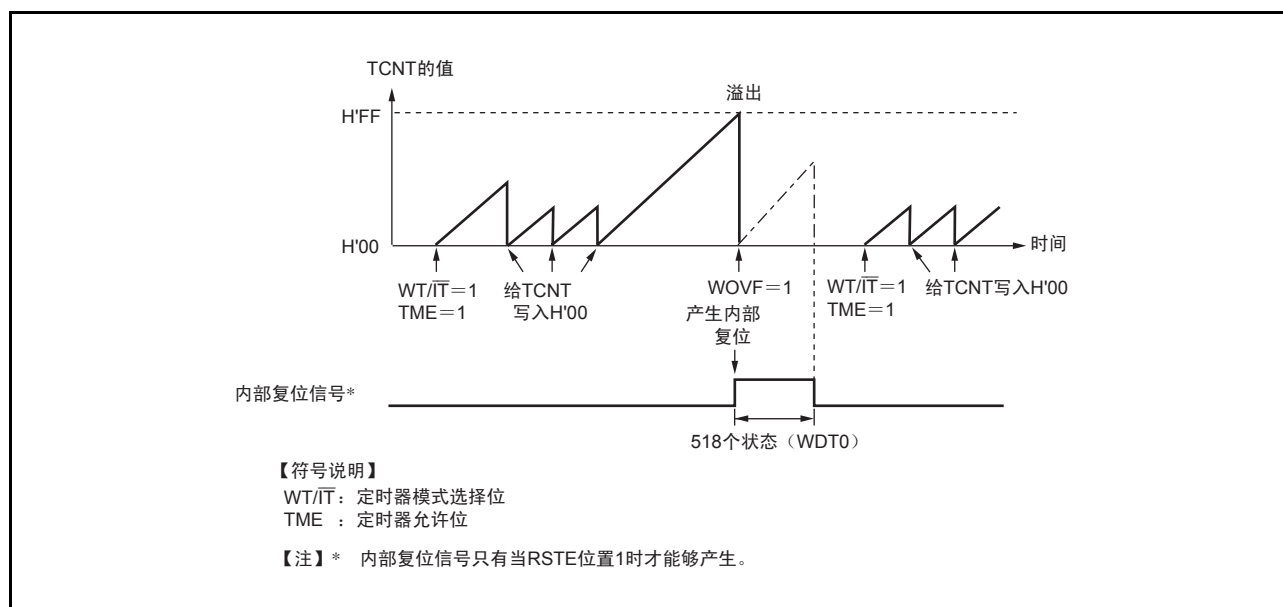


图 10.2 看门狗定时器模式时的工作

10.3.2 看门狗定时器上溢标志 (WOVF) 的置位时序

在看门狗定时器模式，TCNT 上溢后，RSTCSR 的 WOVI 位置 1。而且，RSTCSR 的 RSTE 位置 1 后 TCNT 上溢时，本 LSI 全部生成内部复位信号。此时序如图 10.3 所示。

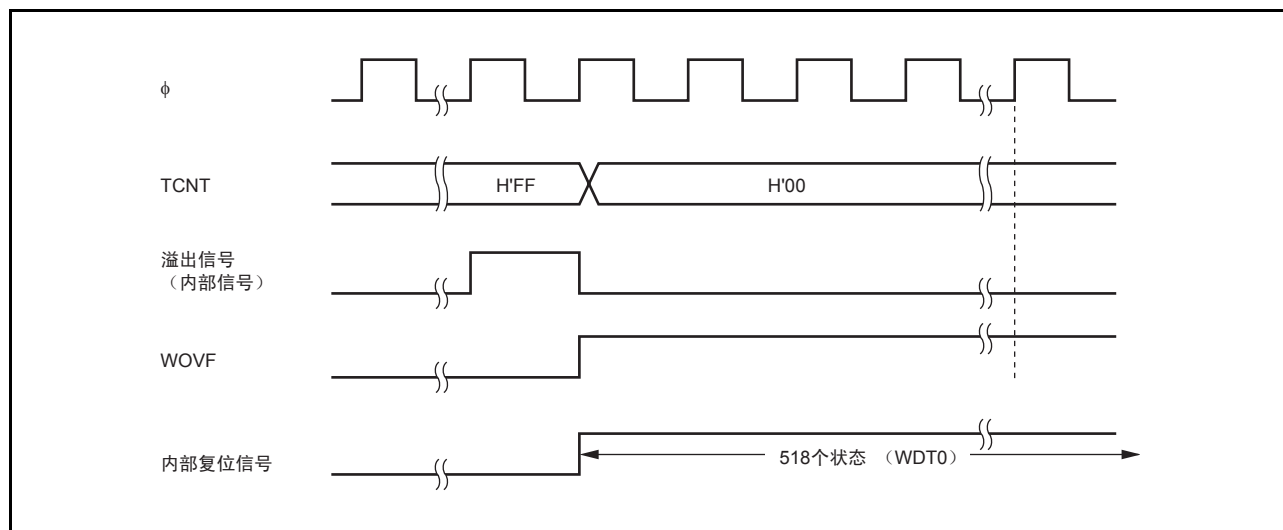


图 10.3 WOVI 的置位时序

10.3.3 间隔定时器模式

作为间隔定时器模式使用时，设定 TCSR 的 $\overline{WT/IT}$ 位 = 0，TME 位 = 1。

作为间隔定时器工作时，TCNT 在每次上溢时产生间隔定时器中断 (WOVI)。因此，每隔一定时间能生成中断。

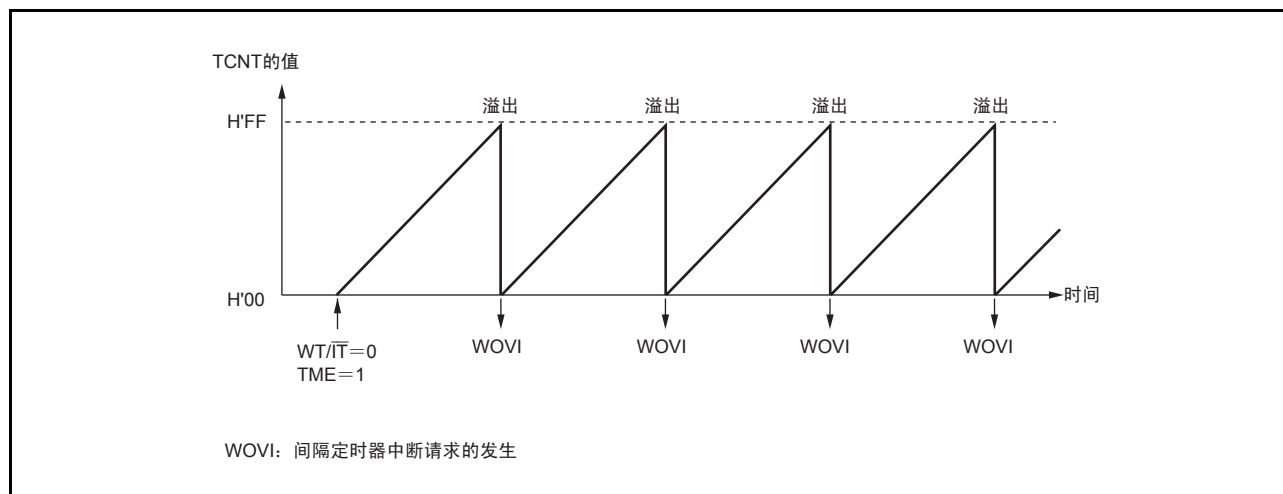


图 10.4 间隔定时器模式的工作

10.3.4 上溢标志 (OVF) 的置位时序

在间隔定时器模式，TCNT 溢出后，TCSR 的 OVF 位置 1，同时请求间隔定时器中断 (WOVI)。此时序如图 10.5 所示。

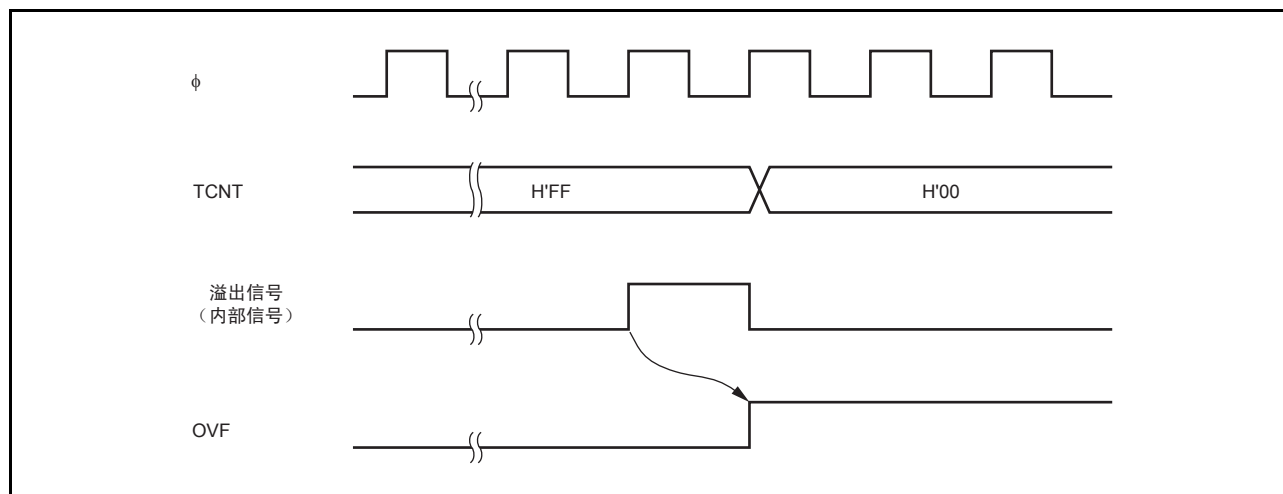


图 10.5 OVF 的置位时序

10.4 中断源

在间隔定时器模式时，通过上溢产生间隔定时器中断 (WOVI)。当 TCSR 的 OVF 标志置 1 时，总会生成间隔定时器中断。务必在中断处理子程序中将 OVF 清 0。

名称	中断源	中断标志
WOVI	TCNT 的上溢	WOVF

10.5 使用时的注意事项

10.5.1 寄存器存取时的注意事项

为了使 TCNT、TCSR、RSTCSR 不能轻易的被改写，写入方法与一般的寄存器有所不同。用以下的方法，进行读 / 写。

10.5.1.1 写入 TCNT、TCSR

向 TCNT、TCSR 写时，务必使用字传送指令，而使用字节传送指令时则无效。如图 10.6 所示。

写时，TCNT 和 TCSR 赋值到同一个地址。为此，写入 TCNT 时，将高位字节作为 H'5A，低位字节作为写数据传送。写入 TCSR 时，将高位字节作为 H'A5，低位字节作为写数据传送。如此传送后，低位字节的数据写入到 TCNT 或 TCSR。

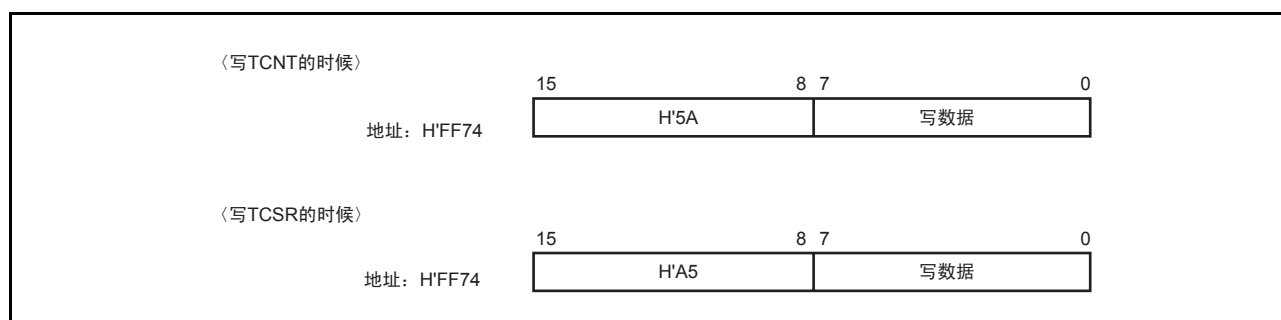


图 10.6 向 TCNT、TCSR 写

10.5.1.2 写入 RSTCSR

写入 RSTCSR 时，请进行字传送，不能以字节传送指令写。如图 10.7 所示。

向 WOVF 位写 0 与向 RSTE 位和 RSTS 位的写入方法不同。

WOVF 位写 0 时，将高位字节作为 H'A5，低位字节作为 H'00 传送。如此，WOVF 位清 0。此时，RSTE、RSTS 位不受影响。写入 RSTE、RSTS 位时，将高位字节作为 H'5A，低位字节作为写数据传送。如此，低位字节的位 6 和位 5 的值分别写入 RSTE 位和 RSTS 位。此时，WOVF 位不受影响。

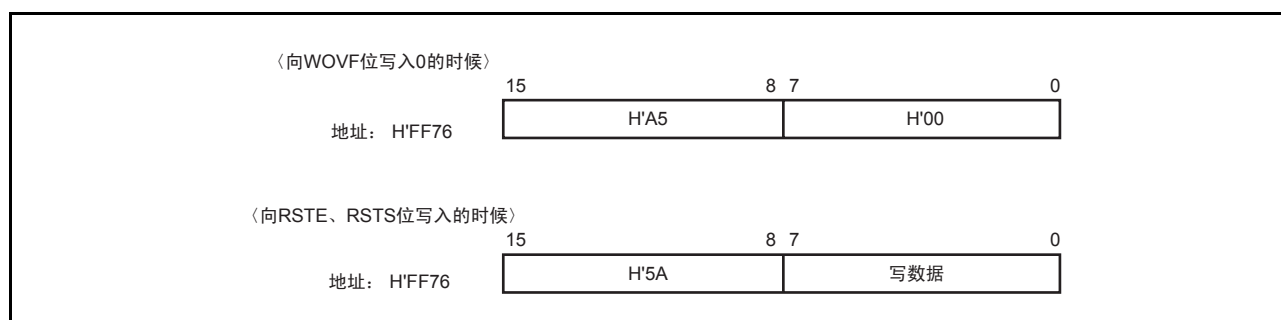


图 10.7 向 RSTCSR 写入

10.5.1.3 从 TCNT、TCSR、RSTCSR 读

读后，可使用与一般寄存器同样的方法。TCSR 赋值到地址 H'FF74，TCNT 赋值到地址 H'FF75，RSTCSR 赋值到地址 H'FF77。

10.5.2 定时器计数器 (TCNT) 的写和累加计数的竞争

在 TCNT 的写周期 T_2 状态即使产生累加计数也不进行累加计数，优先写入 TCNT 计数器。如图 10.8 所示。

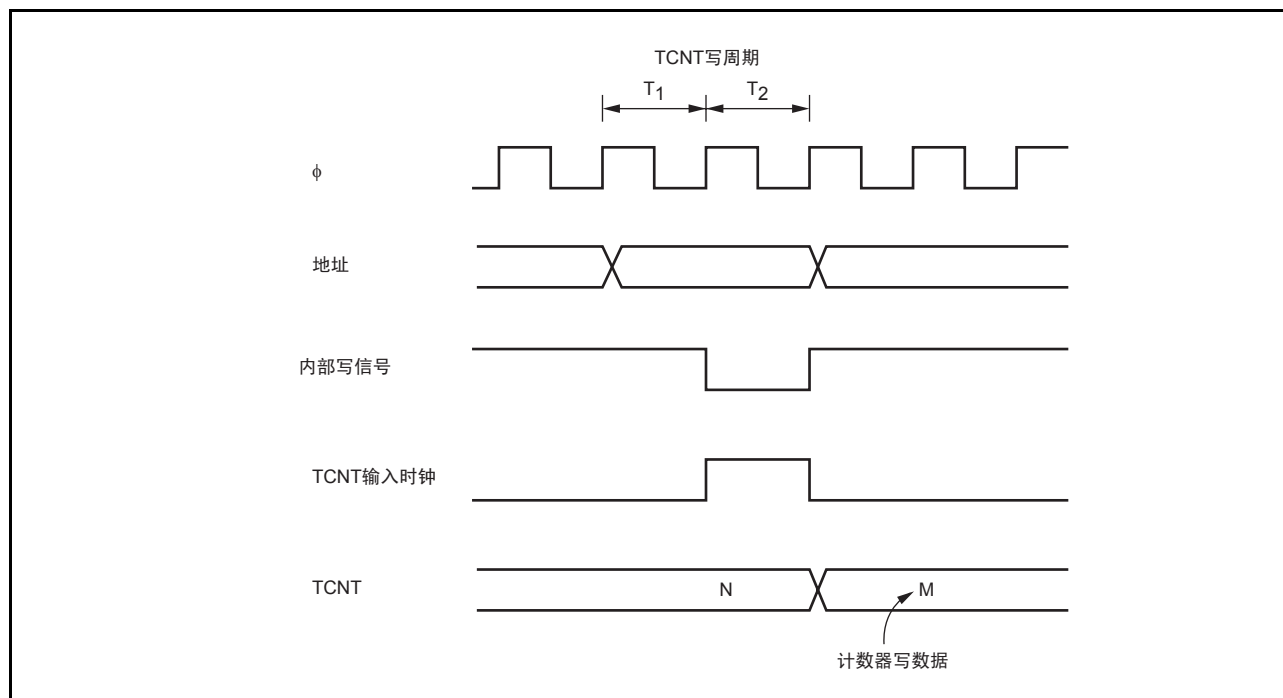


图 10.8 TCNT 的写和累加计数的竞争

10.5.3 CKS2 ~ CKS0 位的改写

在 WDT 工作中改写 TCSR 的 CKS2 ~ CKS0 位后，会不能正确进行累加计数。改写 CKS2 ~ CKS0 位时，务必在停止 WDT 后（从 TME 位清 0 后）进行。

10.5.4 看门狗定时器模式和间接定时器模式的切换

在 WDT 工作中，切换看门狗定时器模式于间隔定时器模式后，会产生不能正常工作的情况，定时器模式的切换，务必在停止 WDT 后（从 TME 位清 0 开始）进行。

10.5.5 看门狗定时器模式的内部复位

在看门狗定时器模式时将 RSTE 位置 0 后，即使 TCNT 上溢，本 LSI 内部也不复位，但是 WDT 的 TCNT、TCSR 进行复位。

从产生上溢开始的 132 个状态期间，向 TCNT、TCSR、RSTCR 不能写入。而且，在此期间也不能识别 WOVF 标志的读取。因此，WOVF 标志的清除从产生上溢开始到等待 132 个状态后，请向 WOVF 标志写入 0。

10.5.6 间隔定时器模式下的 OVF 标志清除

进入间隔定时器模式时，当 OVF 标志的设置与 OVF 标志的读取发生竞争时，即使读取 OVF=1 后，如果 OVF 写入 0，也不能清除标志。禁止间隔定时器中断、查询 OVF 标志等，OVF 标志的设置与读取可能会发生竞争时，如果清除标志，至少要读取 2 次以上 OVF = 1 的状态后，再给 OVF 写入 0。

11. 实时时钟 (RTC)

实时时钟 (RTC: Real Time Clock) 是可计算 1 秒到 1 周时间的定时器。RTC 框图如图 11.1 所示。

11.1 特点

- 计时秒、分、时及星期
- 开始/停止功能
- 复位功能
- 可通过BCD代码读/写的秒、分、时及星期计数器
- 周期 (秒、分、时、日、周) 中断
- 8位自由运行计数器
- 时钟源的选择
- 采用外部总线接口方式的8位总线3周期存取时序

RTC 寄存器被分配到外部地址区域 7 的一部分 (H'FFFF48 ~ H'FFFF4F)。

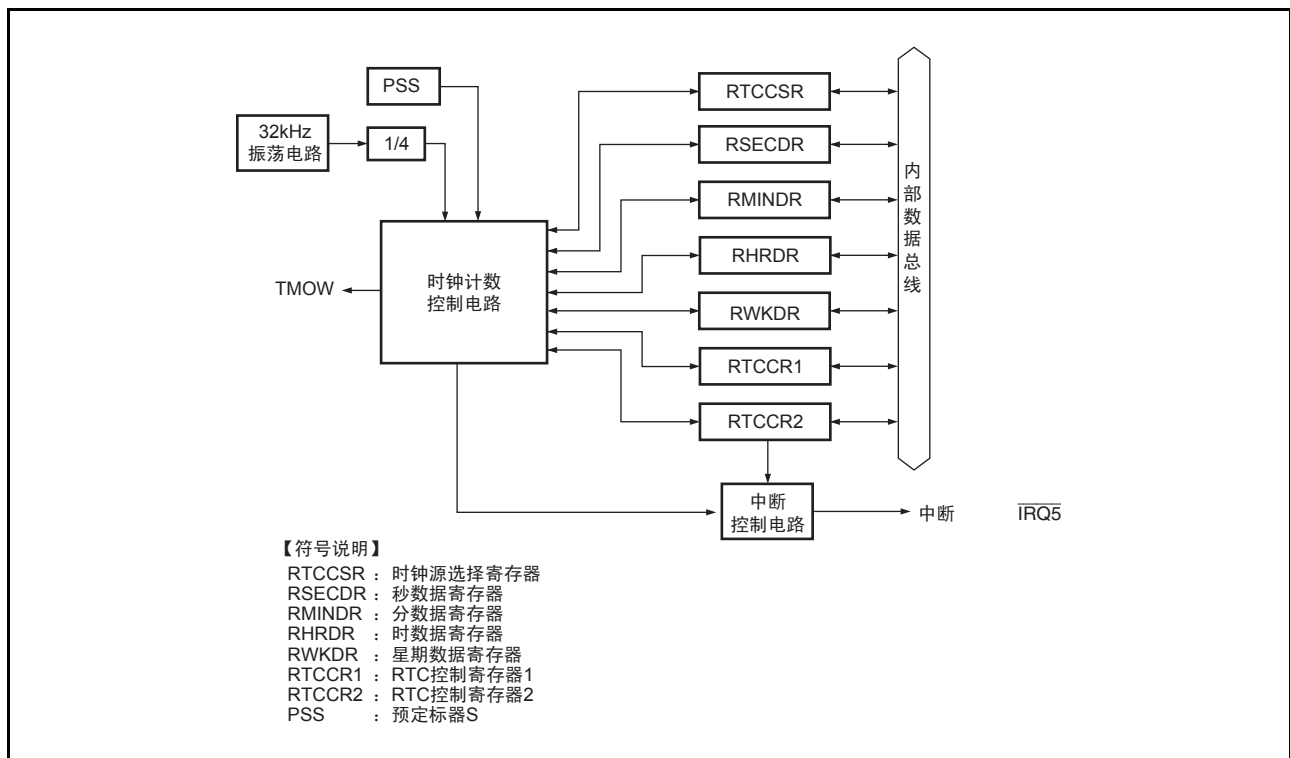


图 11.1 RTC 框图

11.2 输入输出管脚

RTC 的输入输出管脚如表 11.1 所示。

表 11.1 管脚结构

名称	简称	输入输出	功能
时钟输出	TMOW	输出	RTC 分频时钟输出管脚

11.3 寄存器的说明

RTC 中有以下寄存器。

- 秒数据寄存器 (RSECDR)
- 分数据寄存器 (RMINDR)
- 时数据寄存器 (RHRDR)
- 星期数据寄存器 (RWKDR)
- RTC控制寄存器1 (RTCCR1)
- RTC控制寄存器2 (RTCCR2)
- 时钟源选择寄存器 (RTCCSR)
- 扩展模块停止寄存器 (EXMDLSTP)

11.3.1 秒数据寄存器 (RSECDR)

RSECDR 用于计算秒。本寄存器通过 $\overline{\text{STBY}}$ 输入或者 RTCCR1 的 RST 位被初始化为 H'00，但不能通过 RES 输入初始化。RSECDR 以 BCD 代码表示，进行从 0 到 59 的计算。关于秒、分、时及星期的读取请参考 11.4.2 读取时刻顺序。

位	位名	初始值*	R/W	说明
7	BSY	—	R	RTC 忙 当 RTC 更新秒、分、时、星期数据寄存器的值时，此位被置为 1。此位为 0 时，可使用秒、分、时及星期数据寄存器的值。
6	SC12	—	R/W	秒十位计数 秒十位计数 0 到 5，进行 60 秒计数。
5	SC11	—	R/W	
4	SC10	—	R/W	
3	SC03	—	R/W	秒一位计数 秒一位每秒计数 0 到 9。如果发生进位，则秒十位加 1。
2	SC02	—	R/W	
1	SC01	—	R/W	
0	SC00	—	R/W	

【注】 * $\overline{\text{RES}}$ 后的初始值。

11.3.2 分数据寄存器 (RMINDR)

RMINDR 一旦有 RSECDR 的进位，进行分计数。本寄存器可通过 $\overline{\text{STBY}}$ 输入或者 RTCCR1 的 RST 位被初始化为 H'00，但是不能通过 RES 输入进行初始化。RMINDR 以 BCD 码表示，进行从 0 到 59 的计数。

位	位名	初始值*	R/W	说明
7	BSY	—	R	RTC 忙 当 RTC 更新秒、分、时及星期数据寄存器的值时，此位被置 1。此位为 0 时，可使用秒、分、时及星期数据寄存器的值。
6	MN12	—	R/W	分十位计数 分十位从 0 到 5 计数，进行 60 分的计数。
5	MN11	—	R/W	
4	MN10	—	R/W	
3	MN03	—	R/W	分一位计数 分一位每分从 0 到 9 计数。如果发生进位，分十位加 1。
2	MN02	—	R/W	
1	MN01	—	R/W	
0	MN00	—	R/W	

【注】 * $\overline{\text{RES}}$ 后的初始值。

11.3.3 时数据寄存器 (RHRDR)

RHRDR 如果发生 RMINDR 进位，则进行时刻计数。本寄存器可通过 $\overline{\text{STBY}}$ 输入或者 RTCCR1 的 RST 位初始化为 H'00，但是不能通过 RES 输入进行初始化。RHRDR 以 BCD 码表示，通过 RTCCR1 的 12/24 位的选择来计算 0 到 11 或者计算 0 到 23。

位	位名称	初始值*	R/W	说明
7	BSY	—	R	RTC 忙 当 RTC 更新秒、分、时及星期数据寄存器的值时，此位被置为 1。此位为 0 时，可使用秒、分、时及星期数据寄存器的值。
6	—	0	—	保留位 读取时常读出 0。
5	HR11	—	R/W	时十位计数 时十位从 0 到 2 计数。
4	HR10	—	R/W	
3	HR03	—	R/W	时一位计数 时一位每小时从 0 到 9 计数。如果发生进位，时十位加 1。
2	HR02	—	R/W	
1	HR01	—	R/W	
0	HR00	—	R/W	

【注】 * $\overline{\text{RES}}$ 后的初始值。

11.3.4 星期数据寄存器 (RWKDR)

RWKDR 如果发生 RHRDR 的进位，则进行星期的计算。本寄存器可通过 $\overline{\text{STBY}}$ 输入或者 RTCCR1 的 RST 位被初始化为 H'00，但是不能通过 RES 输入进行初始化。通过 WK2 ~ WK0 位以 2 进制代码从 0 到 6 表示星期。

位	位名	初始值*	R/W	说明
7	BSY	—	R	RTC 忙 当 RTC 更新 (运算中) 秒、分、时及星期数据寄存器的值时，此位被置为 1。此位为 0 时，可使用秒、分、时及星期数据寄存器的值。
6~3	—	全部为 0	—	保留位 读取时通常读出 0。
2	WK2	—	R/W	星期计数 以 2 进制代码表示星期。 000: 星期日 001: 星期一 010: 星期二 011: 星期三 100: 星期四 101: 星期五 110: 星期六 111: 保留 (请不要设定。)
1	WK1	—	R/W	
0	WK0	—	R/W	

【注】 * $\overline{\text{RES}}$ 后的初始值。

11.3.5 RTC 控制寄存器 1 (RTCCR1)

RTCCR1 控制时钟定时器的运行开始 / 停止及复位。本寄存器的位 7 ~ 5 可通过 \overline{STBY} 输入或本寄存器的 RST 位被初始化为 H'00，但是不能通过 \overline{RES} 输入初始化位 7 ~ 5。时刻显示的定义请参考图 11.2。

位	位名称	初始值*	R/W	说明
7	RUN	—	R/W	RTC 运行开始 0: RTC 或自由运行计数器停止运行 1: RTC 或自由运行计数器开始运行
6	12/24	—	R/W	运行模式 0: RTC 以 12 小时模式运行。RHRDR 进行 0 ~ 11 的计数。 1: RTC 以 24 小时模式运行。RHRDR 进行 0 ~ 23 的计数。
5	PM	—	R/W	上午 / 下午 0: RTC 在 12 小时模式时有效，表示上午。 1: RTC 在 12 小时模式时有效，表示下午。
4	RST	0	R/W	复位 0: 正常运行 1: 除 RTCCSR 及此位外，复位全部寄存器、控制电路。置位为 1 后，请务必将此位清 0。
3~0	—	全部为 0	—	保留位 读取时常读出 0。

【注】 * \overline{RES} 后的初始值。

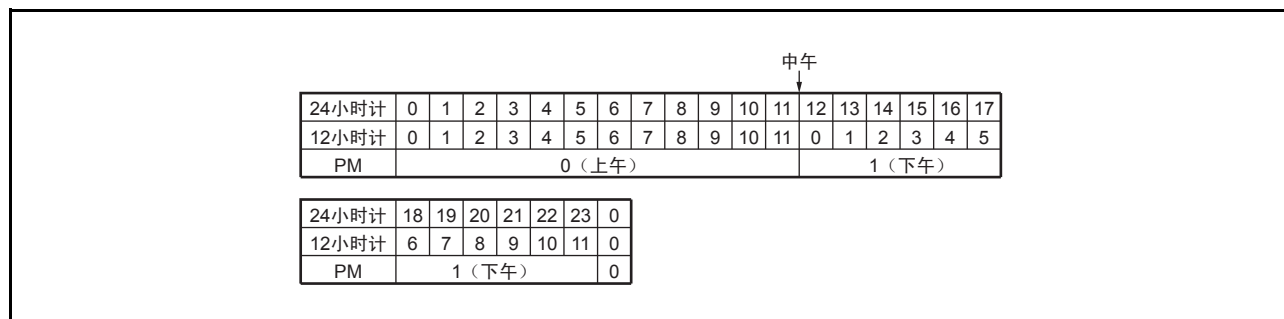


图 11.2 时刻显示的定义

11.3.6 RTC 控制寄存器 2 (RTCCR2)

RTCCR2 控制周、日、时、分及秒的 RTC 周期中断。本寄存器可通过 $\overline{\text{STBY}}$ 输入或者 RTCCR1 的 RST 位初始化为 H'00，但不能通过 $\overline{\text{RES}}$ 输入进行初始化。如果允许周、日、时、分及秒的各中断，发生中断时，IRQ 状态寄存器 (ISR) 的 IRQ5F 标志会被置为 1。同时，RTC 作为自由运行计数器运行时，控制自由运行计数器的溢出中断。

位	位名	初始值*	R/W	说明
7、6	—	全部为 0	—	保留位 读取时常读出 0。
5	FOIE	—	R/W	自由运行计数器溢出中断允许 0: 禁止溢出中断 1: 允许溢出中断
4	WKIE	—	R/W	周周期中断允许 0: 禁止周周期中断 1: 允许周周期中断
3	DYIE	—	R/W	日周期中断允许 0: 禁止日周期中断 1: 允许日周期中断
2	HRIE	—	R/W	时周期中断允许 0: 禁止时周期中断 1: 允许时周期中断
1	MNIE	—	R/W	分周期中断允许 0: 禁止分周期中断 1: 允许分周期中断
0	SEIE	—	R/W	秒周期中断允许 0: 禁止秒周期中断 1: 允许秒周期中断

【注】 * $\overline{\text{RES}}$ 后的初始值。

11.3.7 时钟源选择寄存器 (RTCCSR)

RTCCSR 进行时钟源的选择。本寄存器可通过 $\overline{\text{STBY}}$ 输入或者 $\overline{\text{RES}}$ 输入初始化为 H'08。自由运行计数器通过 RTCCR1 的 RUN 位控制计数器的运行开始 / 停止。如果选择 32.768kHz 以外的时钟，RTC 变为无效，而作为 8 位的自由运行计数器运行。同时，如果 RTCCR2 的 FOIE 位置 1，允许自由运行计数器的溢出中断就可以发生中断。将系统时钟 32、16、8、4 分频的时钟，会在高速模式、中速模式、睡眠模式、子激活模式、子睡眠模式下被输出。

位	位名	初始值	R/W	说明
7	—	0	—	保留位 读取时常读出 0。
6	RCS6	0	R/W	时钟输出选择 UCTLR 的 TMOWE 位置 1 时，选择从 TMOW 管脚输出的时钟。 00: $\phi/4$ 01: $\phi/8$ 10: $\phi/16$ 11: $\phi/32$
5	RCS5	0	R/W	
4	—	0	—	保留位 读取时常读出 0。
3	RCS3	1	R/W	时钟源选择 0000: $\phi/8$ 自由运行计数器运行 0001: $\phi/32$ 自由运行计数器运行 0010: $\phi/128$ 自由运行计数器运行 0011: $\phi/256$ 自由运行计数器运行 0100: $\phi/512$ 自由运行计数器运行 0101: $\phi/2048$ 自由运行计数器运行 0110: $\phi/4096$ 自由运行计数器运行 0111: $\phi/8192$ 自由运行计数器运行 1000: 32.768kHz..... RTC 运行
2	RCS2	0	R/W	
1	RCS1	0	R/W	
0	RCS0	0	R/W	

11.3.8 扩展模块待机寄存器 (EXMDLSTP)

EXMDLSTP 控制 RTC 及 USB 的时钟供应。

【注】 通过端口 D 寄存器 (PORTD) 读取管脚状态时，访问 EXMDLSTP 后，先虚读一次 H'FFFF40 ~ H'FFFF5F 范围外的外部地址空间 (H'FFFEFC0 ~ H'FFF7FF 等)，再读取 PORTD。

位	位名	初始值	R/W	说明
7~2	—	不定	—	保留位 读取值不定。写入定义为无效。
1	RTCSTOP	0	R/W	RTC 模块停止 0: RTC 模块停止解除 1: RTC 模块停止
0	USBSTOP1	0	R/W	USB 模块停止 0: USB 模块停止部分解除 1: USB 模块完全停止

11.4 RTC 的运行

11.4.1 加电后寄存器的初始设定及再设定顺序

RTC 虽然可通过 $\overline{\text{STBY}}$ 输入复位保存秒、分、时、星期、运行模式、上午 / 下午信息的寄存器，但是不能通过 $\overline{\text{RES}}$ 输入复位这些寄存器。因此，加电及输入后，要初始设定所有寄存器。初始设定和再次设定 RTC 的顺序如图 11.3 所示。之后与 $\overline{\text{RES}}$ 输入无关，只要供给电源，就可以获取正确时间。

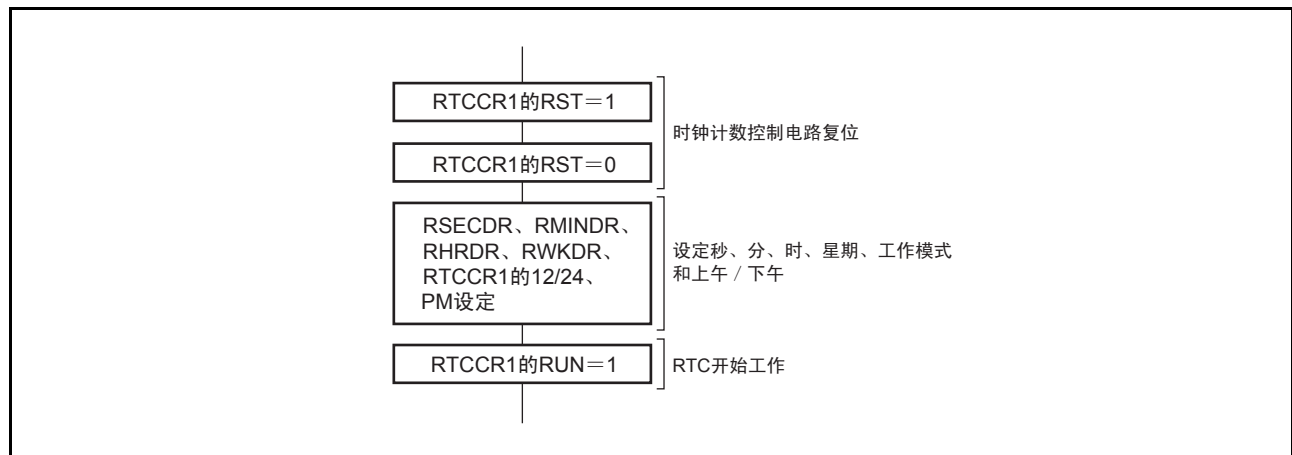


图 11.3 初始设定顺序

11.4.2 读取时刻顺序

如果在读取时刻期间更新秒、分、时、星期数据，就不能获取正确时间，因此需要再次读取。不能获取正确时间的例如图 11.4 所示。在此例中，由于仅 RSECDR 是在数据更新后读取，所以大约会有 1 分钟的不一致。

读取正确时间的方法有 3 种。

1. 判断 BSY 位，BSY 位从 1 变为 0 后，读取表示秒、分、时、星期的寄存器。BSY 位被置 1 约 62.5ms 后，进行寄存器的更新，BSY 位被清 0。
2. 使用中断，如果 ISR 的 IRQ5F 标志被置 1，确认 BSY 位为 0 后，读取表示秒、分、时、星期的寄存器。
3. 连续 2 次读取表示秒、分、时、星期的寄存器，如果读取的数据没有变化，就使用该数据。

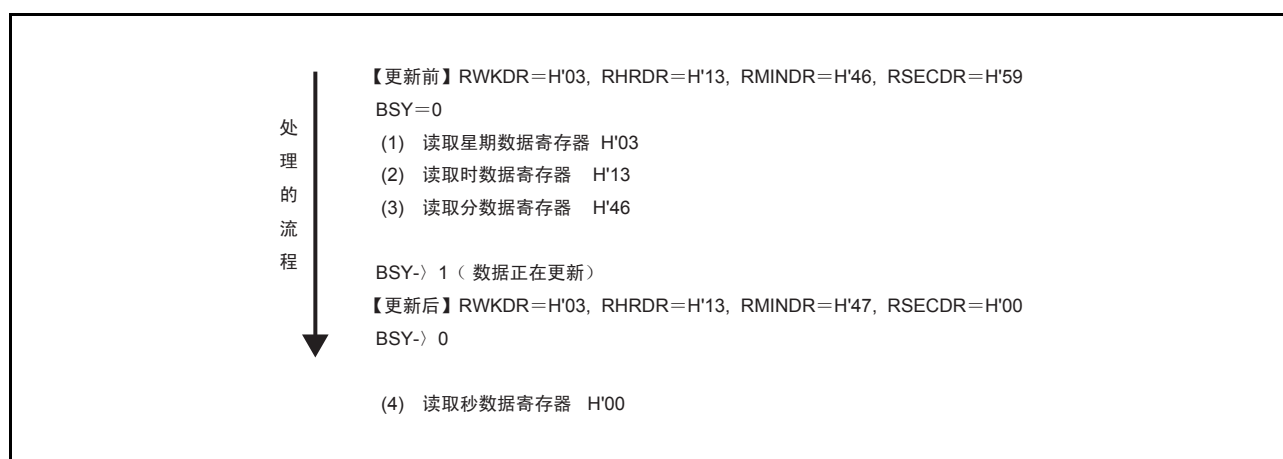


图 11.4 不能获得正确时间的例

11.5 中断源

RTC 的中断源如表 11.2 所示。RTC 的中断源有周、日、时、分、秒 5 种。

使用中断时，在其它寄存器的设定（包含中断控制器的 ISCRH、IER）完成后，最后启动 RTC。并且，请不要同时将 RTCCR2 中多个中断允许位置 1。

如果发生 RTC 的中断请求，ISR 的 IRQ5F 标志就被置为 1。清除标志时，读取一次标志 1 后，请写入 0。

图 11.5、图 11.6 分别表示的是使用 RTC 中断时初始化设定顺序和 RTC 中断处理程序例。

表 11.2 中断源

中断源名称	中断源	中断允许位
上溢中断	在上溢时发生自由运行计数器。	FOIE
周周期中断	星期数据寄存器的值变为 0 时，在 1 周周期发生中断。	WKIE
日周期中断	每次计算星期数据寄存器时，在 1 日周期发生中断。	DYIE
时周期中断	每次计算时数据寄存器时，在 1 小时周期发生中断。	HRIE
分周期中断	每次计算分数数据寄存器时，在 1 分周期发生中断。	MNIE
秒周期中断	每次计算秒数据寄存器时，在 1 秒周期发生中断。	SEIE

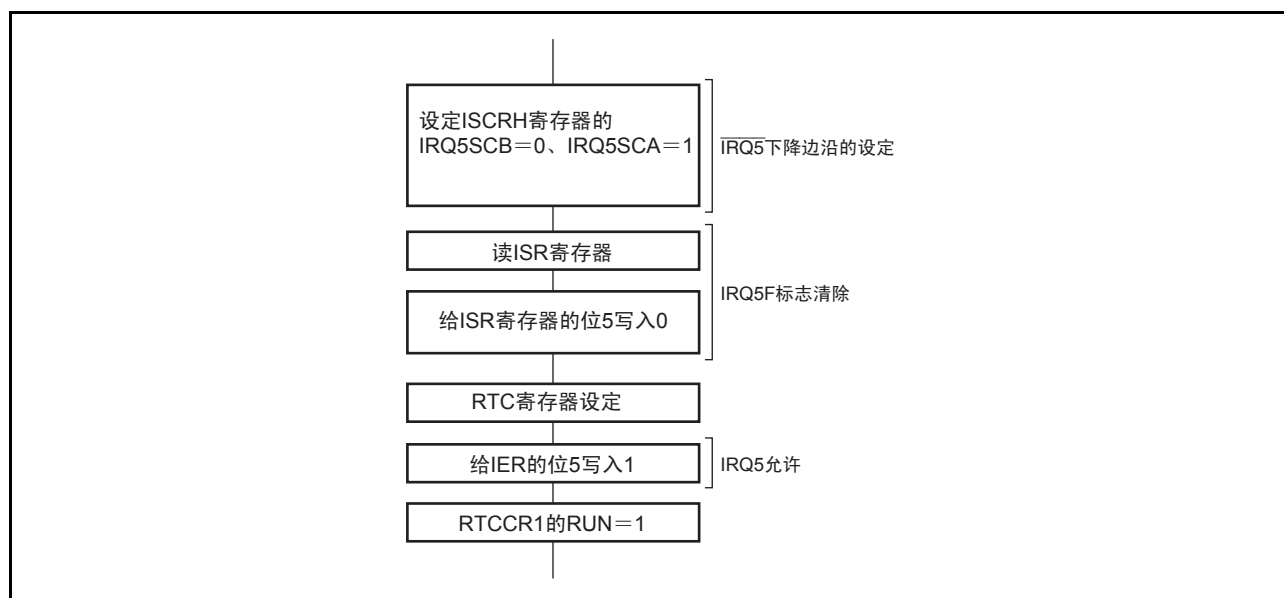


图 11.5 使用 RTC 中断时初始化顺序

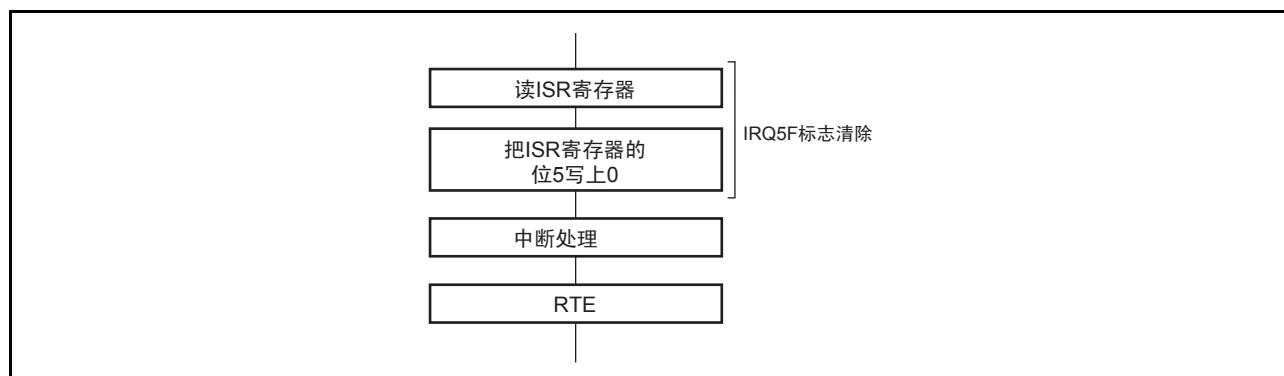


图 11.6 RTC 中断处理程序例

11.6 各模式下的运行状态

将 RTC 设定为时钟运行和设定为自由运行定时器运行时，分别在各模式下的运行状态如表 11.3 所示。由于时钟在低功耗模式下也可以继续运行，因此不需要时钟运行时，请务必通过 EXMDLSTP 启动模块停止。

表 11.3 各模式下的运行状态

	高速	中速	睡眠	模块停止	监视	子激活	子睡眠	软件待机	硬件待机
时钟运行	子时钟运行			停止 (保持)	子时钟运行			停止 (复位)	
自由运行定时器运行	运行			停止 (保持)	停止 (保持)			停止 (复位)	

11.7 使用注意事项

11.7.1 使用仿真器时的注意事项

在 E6000 中，RTC 模块已经被实装在外扩展板上，在外扩展板处理中需要访问，因此有以下制约。
E10A 和产品芯片就没有以下制约。

- 不支持 H8S/2218 群的模式 7 (单芯片模式) 的 RTC 运行。
- 在 H8S/2218 群的模式 6 (On-chip ROM 模式) 和 H8S/2212 群的模式 7 (单芯片模式) 下使用时，在初始状态由于 A7~A0 变为输入管脚，因此在访问本模块前设定 PC7DDR~PC0DDR = H'FF，使 A7~A0 变为输出管脚。
- 在 H8S/2218 群的模式 4、5 (On-chip ROM 无效模式) 下使用时，由于 A7~A0 是输出管脚，不需要设定。

11.7.2 关于总线接口

本模块的总线接口遵循外部区域 7 的总线规格。因此访问本模块前，请务必在总线控制器的寄存器中将区域 7 设定为总线宽 8 位，3 态访问。

11.7.3 通过开发板 D 寄存器 (PORTD) 读取管脚状态时的注意事项

访问 EXMDLSTP 和 RTC 寄存器 (地址范围 H'FFFF40 ~ H'FFFF5F) 后，请虚读一次 H'FFFF40 ~ H'FFFF5F 范围外的外部地址空间 (H'FFFEFC0 ~ H'FFF7FF 等) 后，再读取 PORTD。

12. 串行通信接口 (SCI)

本 LSI 配备了独立的双通道串行通信接口 (SCI: Serial Communication Interface), SCI 可进行异步和同步两种方式的串行通信。异步方式下, 可与通用异步收发器 (UART) 和异步通讯接口适配器 (ACIA) 等标准异步式通信芯片进行串行通信。而且在异步模式还配备有多个处理器间的串行通信功能 (多处理器通信功能)。除此之外, SCI 作为异步模式的扩展功能, SCI 支持 ISO/IEC7816-3 (Identification Card) 标准的智能卡 (IC 卡) 接口。

12.1 特点

- 可将串行数据通信方式设定为异步或时钟同步模式
- 可执行全双工通信。
发送和接收功能相互独立, 并具有双缓冲结构。因此, 可实现同时连续的发送和接收。
- 内部波特率发生器允许选择任意比特率。
作为发送或接收时钟源也可选择来自波特率生成器的内部时钟或者来自 SCK 管脚的外部时钟。
- 可选择 LSB 优先 / MSB 优先 (除异步模式下的 7 位数据之外)
- 中断源: 4 种
发送结束、发送数据空、接收数据满、接收错误 4 种中断源。并且可通过发送数据空、接收数据满中断源启动 DMAC。
- 可设定为模块停止模式

异步模式

- 数据长度: 可选择 7 位 / 8 位。
- 停止位长度: 可选择 1 位 / 2 位。
- 奇偶校验性: 可选择偶校验 / 奇校验 / 无奇偶校验。
- 接收错误检测: 奇偶校验错误、溢出错误、帧错误。
- 检测中止: 发生帧错误时, 直接读取 RxD 管脚的电平, 就可检测出中止。
- 平均传送率生成器 (SCI_0):
可选择以 16MHz 运行时的 921.569kbps/720kbps/460.784kbps/115.196kbps, 以 24MHz 运行时 921.053kbps/720kbps/460.526kbps/115.132kbps。
- TPU 的传送率生成器 (SCI_0)
- 可进行多处理器间的通信

时钟同步模式

- 数据长度: 8 位。
- 检测接收错误: 溢出错误。
- SCI 选择功能 (SCI_0): $\overline{\text{IRQ7}} = 1$ 时可选择 TxD0 = 高阻抗、SCK0 = 固定输入高电平
- 可与具有时钟同步模式通信功能的其它 LSI 进行串行数据通信

智能卡接口

- 接收时, 如检测出奇偶校验错误, 会自动发送错误信号。
- 接收到接收/发送错误信号会再次自动发送数据。
- 支持直接转换/反码转换。

12.1.1 框图

图 12.1 所示为 SCI_0 框图, 图 12.2 所示为 SCI_2 框图。

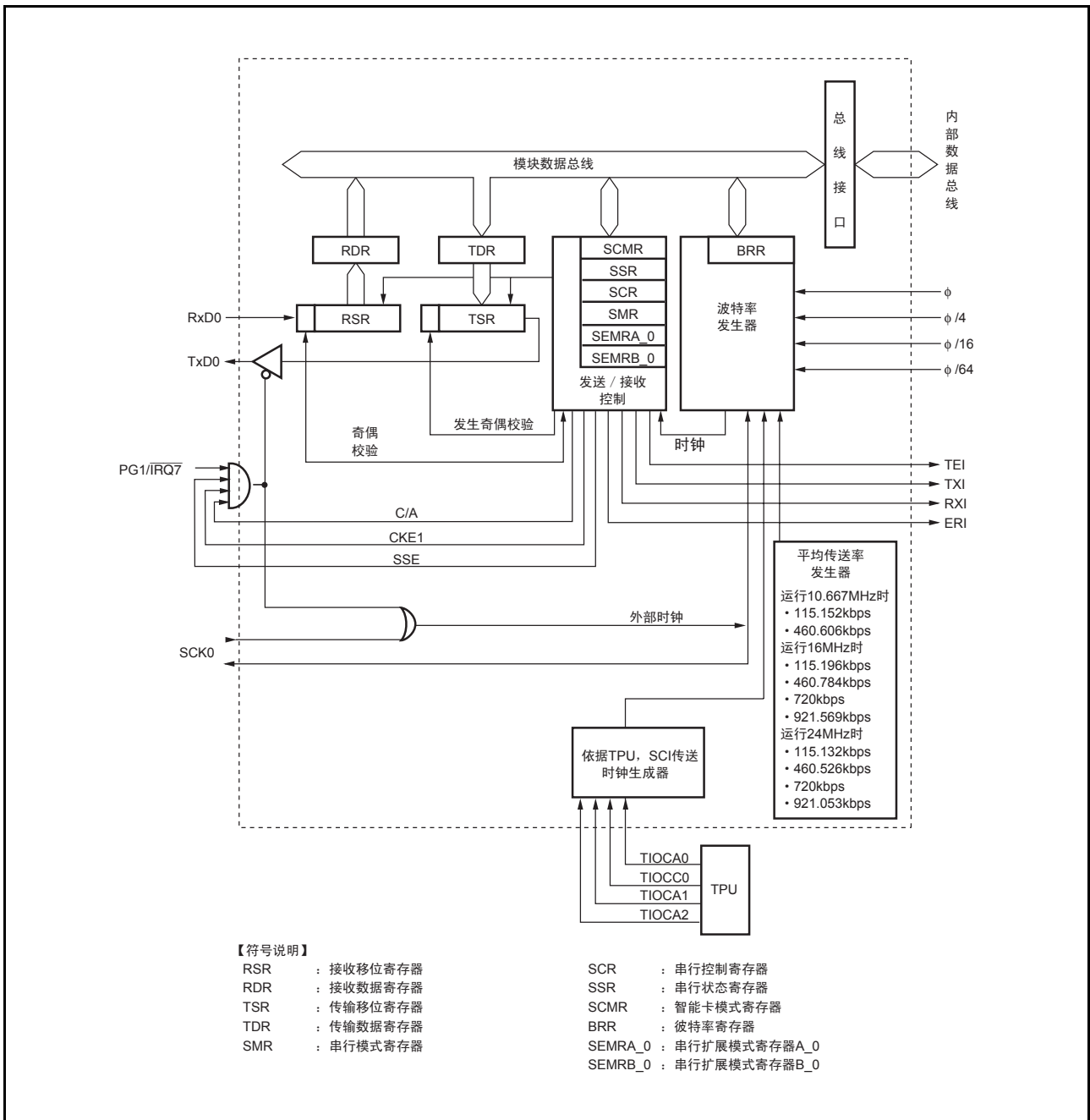


图 12.1 SCI_0 框图

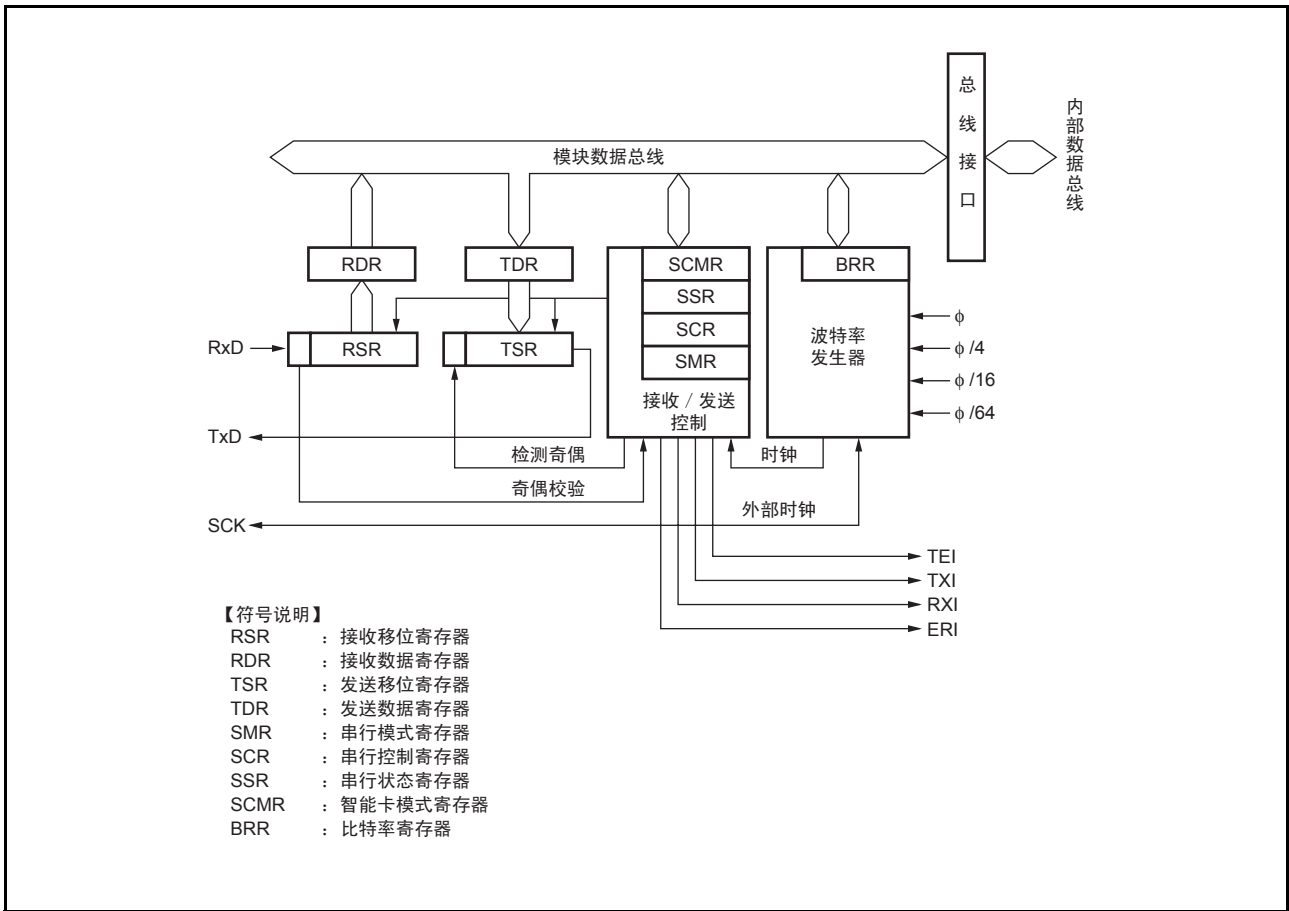


图 12.2 SCI_2 框图

12.2 输入 / 输出管脚

SCI 中有如表 12.1 所示的输入 / 输出管脚。

表 12.1 管脚结构

通道	管脚名称 *	输入 / 输出	功能
0	SCK0	输入 / 输出	通道 0 的时钟输入输出管脚
	RxD0	输入	通道 0 的发送数据输入管脚
	TxD0	输出	通道 0 的接收数据输出管脚
2	SCK2	输入 / 输出	通道 2 的时钟输入输出管脚
	RxD2	输入	通道 2 的发送数据输入管脚
	TxD2	输出	通道 2 的接收数据输出管脚

【注】 * 本文中省略了通道名，管脚均以 SCK、RxD、TxD 简称。

12.3 寄存器说明

SCI 的每个通道都有以下寄存器。并且串行通信模式寄存器 (SMR)、串行通信状态寄存器 (SSR)、串行通信控制寄存器 (SCR) 在通常串行通信接口模式和智能卡接口模式中有部分位的功能不同, 详情另有记载。

- 接收移位寄存器 (RSR)
- 接收数据寄存器 (RDR)
- 送数据寄存器 (TDR)
- 发送移位寄存器 (TSR)
- 串行通信模式寄存器 (SMR)
- 串行通信控制寄存器 (SCR)
- 串行通信状态寄存器 (SSR)
- 智能卡模式寄存器 (SCMR)
- 串行扩展模式寄存器 A_0 (SEMR A_0) [仅对通道0]
- 串行扩展模式寄存器 B_0 (SEMR B_0) [仅对通道0]
- 比特率寄存器 (BRR)

12.3.1 接收移位寄存器 (RSR)

RSR 是用来接收, 并将从 RxD 管脚输入的串行数据进行并行转换的移位寄存器。当接收 1 帧的数据后, 数据就自动传送到 RDR。此寄存器不能通过 CPU 直接访问。

12.3.2 接收数据寄存器 (RDR)

RDR 是保存接收数据的 8 位寄存器。当接收到 1 帧数据, 接收数据从 RSR 传送到 RDR 寄存器, RSR 就可接收下一个数据。RSR 和 RDR 是双缓冲结构, 因此可连续接收信息。确认 SSR 的 RDRF 被置为 1 后, 只需读取一次 RDR。不能通过 CPU 对 RDR 进行写入操作。

复位、待机模式、监视模式、子激活模式、子睡眠模式或模块停止模式时, RDR 初始化为 H'00。

12.3.3 发送数据寄存器 (TDR)

TDR 是保存发送数据的 8 位寄存器。如果检测到 TSR 为空, 写入 TDR 的发送数据被传送到 TSR, 并开始发送数据。TDR 和 TSR 是双缓冲结构, 可连续发送数据。发送完 1 帧数据后, 如果将下一次要发送的数据写入 TDR, 则会把它传送到 TSR 继续发送。TDR 通常可通过 CPU 进行读 / 写操作, 为了正确进行串行发送, 必须在确认 SSR 的 TDRE 被置为 1 后, 仅对 TDR 写入 1 次发送数据。

复位、待机模式、监视模式、子激活模式、子睡眠模式或模块停止模式时, TDR 初始化为 H'FF。

12.3.4 发送移位寄存器 (TSR)

TSR 是发送串行数据的移位寄存器。写入到 TDR 的发送数据被自动传送到 TSR, 通过传送到 TxD 管脚, 进行串行数据的发送。不能直接通过 CPU 访问 TSR。

12.3.5 串行通信模式寄存器 (SMR)

SMR 是用于选择通信形式和内部波特率发生器时钟源的寄存器。SMR 在普通模式和智能卡接口模式，有部分位的功能不同。

- 普通串行通信接口模式 (SCMR 的 SMIF = 0 时)

位	位名	初始值	R/W	说明
7	C/A	0	R/W	通信模式 0: 在异步模式下运行。 1: 在时钟同步模式下运行。
6	CHR	0	R/W	字符长度 (仅对异步模式有效) 0: 以 8 位数据长度发送接收。 1: 以 7 位数据长度发送接收。发送时不发送 TDR 的 MSB (位 7) 时钟同步模式下, 与 CHR 的设定无关, 数据长度固定为 8 位。
5	PE	0	R/W	奇偶校验允许 (仅对异步模式有效) 当此位为 1, 发送时附加奇偶校验位, 接收时要检查奇偶校验。在多处理器模式中, 奇偶校验位的添加和检查与此位的设置无关。
4	O/E	0	R/W	奇偶模式 (仅在异步模式中 PE = 1 时有效) 0: 对发送和接收数据进行偶校验。 1: 对发送和接收数据进行奇校验。
3	STOP	0	R/W	停止位的长度 (仅在异步模式下有效) 选择发送时停止位的长度。 0: 1 停止位 1: 2 停止位 接收时与此位的设定无关, 只检查停止位的第 1 位, 第 2 位为 0 时, 看作是下一帧的起始位。
2	MP	0	R/W	多处理器模式 (仅对异步模式有效) 此位为 1 时, 可实现多处理器通信功能。 在多处理器模式下, PE、O/E 位的设定无效。 详情请参考 12.5 多处理器通信功能。
1 0	CKS1 CKS0	0 0	R/W R/W	选择时钟 1 ~ 0 选择内部波特率发生器的时钟源。 00: ϕ 时钟 (n=0) 01: $\phi/4$ 时钟 (n=1) 10: $\phi/16$ 时钟 (n=2) 11: $\phi/64$ 时钟 (n=3) 此位的设定值和波特率的关系, 请参考 12.3.11 比特率寄存器 (BRR)。n 是用 10 进制表示的设定值, 表示 12.3.11 比特率寄存器 (BRR) 中 n 的值。

• 智能卡接口模式 (SCMR 的 SMIF = 1 时)

位	位名	初始值	R/W	说明
7	GM	0	R/W	<p>GSM 模式</p> <p>当此位置为 1, SCI 在 GSM 模式下工作。GSM 模式下 TEND 的设定时间从开始前推到 11.0etu, 并追加时钟输出控制功能。详细请参考 12.7.9 时钟输出控制。</p> <p>0: 普通智能卡接口模式的运行 (初始值)</p> <p>a. TEND 标志从起始开始 12.5etu (数据块传送模式时为 11.5etu) 的时间, 才会发生。</p> <p>b. 仅控制时钟输出的 ON/OFF。</p> <p>1: GSM 模式的智能卡接口模式的运行</p> <p>a. TEND 标志从开始位的起始开始 11.0etu, 智能卡接口模式运行。</p> <p>b. 除了控制时钟输出的 ON/OFF, 也可固定控制高/低电平 (用 SCR 设定)。</p>
6	BLK	0	R/W	<p>将此位置为 1, SCI 在数据块传送模式下工作。数据块传送模式的详细请参考 12.7.4 块传送模式。</p> <p>0: 普通智能卡接口模式的运行</p> <p>a. 进行错误信号的发送、检测再次自动发送数据</p> <p>b. 由 TEND 标志发生 TXI 中断</p> <p>c. TEND 标志的设定时间是从开始发送到 12.5etu 后 (GSM 模式下是 11.0etu 后)</p> <p>1: 在数据块传送模式下运行</p> <p>a. 不进行错误信号的发送、检测和数据的再次自动发送</p> <p>b. 由 TDRE 标志发生 TXI 中断</p> <p>c. TEND 标志的设定时间是从开始发送到 11.5etu 后 (GSM 模式下是 11.0etu 后)</p>
5	PE	0	R/W	<p>奇偶校验允许</p> <p>当此位置为 1 时, 发送前发送数据附加奇偶位, 接收后检查接收数据的奇偶位。在智能卡接口模式中请把此位置 1 后再使用。</p>
4	O/\bar{E}	0	R/W	<p>奇偶校验模式 (只有 PE = 1 时有效)</p> <p>0: 对发送和接收数据进行偶校验。</p> <p>1: 对发送和接收数据进行奇校验。</p> <p>关于智能卡接口模式中此位的设置方法请参考 12.7.2 数据格式 (除数据块传送模式之外)。</p>
3 2	BCP1 BCP0	0 0	R/W R/W	<p>基本时钟脉冲 1 ~ 0</p> <p>智能卡接口模式下, 选择传送 1 位数据的基本时钟数。</p> <p>00: 32 时钟 (S=32)</p> <p>01: 64 时钟 (S=64)</p> <p>10: 372 时钟 (S=372)</p> <p>11: 256 时钟 (S=256)</p> <p>详细内容请参考 12.7.5 接收数据采样时序接收容限。S 是 12.3.11 比特率寄存器 (BRR) 中所示的 S 的值。</p>

位	位名	初始值	R/W	说明
1	CKS1	0	R/W	时钟选择 1、0 选择内部波特率发生器的时钟源。 00: ϕ 时钟 (n=0) 01: $\phi/4$ 时钟 (n=1) 10: $\phi/16$ 时钟 (n=2) 11: $\phi/64$ 时钟 (n=3) 关于此位设定值和波特率的关系, 请参考 12.3.11 比特率寄存器 (BRR)。n 是用 10 进制表示的设定值, 表示 12.3.11 比特率寄存器 (BRR) 中 n 的值。
0	CKS0	0	R/W	

12.3.6 串行通信控制寄存器 (SCR)

SCR 是进行发送 / 接收控制、中断控制及发送 / 接收时钟源选择的寄存器。关于各中断请求细节请参考 12.9 中断源。SCR 在串行接口模式和智能卡接口模式下有部分位的功能不同。

- 普通串行通信接口模式 (SCMR 的 SMIF = 0 时)

位	位名	初始值	R/W	说明
7	TIE	0	R/W	发送中断允许 当该位置为 1, 允许 TXI 中断请求。 可通过从 TDRE 标志读取 1 后, 清 0 或者将 TIE 位清 0 来解除 TXI 中断请求。
6	RIE	0	R/W	接收中断允许 当该位置为 1, 允许 RXI 及 ERI 中断请求。 可通过从 RDRF 或 FER、PER、ORER 的各标志读取 1 后, 清 0 或者将 RIE 位清 0 来解除 RXI 及 ERI 中断请求。
5	TE	0	R/W	发送允许 当该位置为 1, 可进行发送。在此状态, 向 TDR 写入发送数据, 一旦将 SSR 的 TDRE 标志清 0, 则开始串行发送。然而, 将 TE 位置 1 前, 请务必通过 SMR 设定为发送格式。 此位清 0, 禁止发送, SSR 的 TDRE 标志会被固定为 1。
4	RE	0	R/W	接收允许 当该位置为 1, 可进行接收。在此状态, 异步模式下检测出开始位, 时钟同步模式下检测同步时钟输入后, 就可以开始串行接收。然而, 将 RE 位置为 1 前务必要通过 SMR 设定接收格式。 将此位清 0, 禁止接收, 也不会影响 RDRF、或 FER、PER、ORER 的各标志, 并保持其状态。
3	MPIE	0	R/W	多处理器中断允许 (异步模式下 SMR 的 MP = 1 时有效) 此位置为 1 时, 跳过多处理器位为 0 时的接收数据, 禁止设定 SSR 的 RDRF、FER、ORER 各状态标记。如果接收到多处理器位为 1 时的数据, 该位会被自动清除, 而返回正常接收状态。详细内容请参考 12.5 多处理器通信功能。 接收包含 SSR 的 MPB = 0 的数据时, 从 RSR 向 RDR 传送接收数据及检测到接收错误, SSR 的 RDRF、FER、ORER 各标志都不会被置位。如果接收包含 MPB = 1 的接收数据, 将 SSR 的 MPB 置位为 1, 将 MPIE 位自动清 0, 就会允许 RXI、ERI 中断请求 (SCR 的 TIE、RIE 位被置 1 时) 和 FER、ORER 标志的设置。
2	TEIE	0	R/W	发送结束中断允许 此位置为 1 时, 则允许 TEI 中断请求。可通过从 TDRE 标志读取 1 后, 清 0, 将 TEND 标志清 0 或者将 TEIE 位清 0, 来解除 TEI 中断请求。

位	位名	初始值	R/W	说明
1	CKE1	0	R/W	时钟允许 1、0 选择时钟源及 SCK 管脚的功能。 异步模式 00: 内置波特率生成器 (SCK 管脚可当作 I/O 端口使用) 01: 内置波特率生成器 (从 SCK 管脚输出与比特率相同频率的时钟) 1X: 外部时钟 (从 SCK 管脚输入频率为比特率 16 倍的时钟。) 时钟同步模式 0X: 内部时钟 (SCK 管脚作为时钟输出管脚。) 1X: 外部时钟 (SCK 管脚作为时钟输入管脚。)
0	CKE0	0	R/W	

【符号说明】

X: Don't care

• 智能卡接口模式 (SCMR 的 SMIF = 1 时)

位	位名	初始值	R/W	说明
7	TIE	0	R/W	<p>传送中断允许</p> <p>当该位置为 1, 则允许 TXI 中断请求。</p> <p>可通过从 SSR 的 TDRE 标志读取 1 后, 清 0 或者将 TIE 位清 0 来解除 TXI 中断请求。</p>
6	RIE	0	R/W	<p>接收中断允许</p> <p>当该位置为 1, 则允许 RXI 及 ERI 中断请求。</p> <p>可通过从 SSR 的 RDRF、FER、PER、ORER 的各标志读取 1 后, 清 0, 或者将 RIE 位清 0 来解除 RXI 及 ERI 中断请求。</p>
5	TE	0	R/W	<p>发送允许</p> <p>当该位置为 1, 则可发送。在此状态, 向 TDR 中写入发送数据, 将 SSR 的 TDRE 标志清 0, 就会开始串行发送。然而, 在置位为 1 前, 必须要通过 SMR 设定发送格式。</p> <p>如果将此位清 0 发送动作被禁止, SSR 的 TDRE 标志就会被固定为 1。</p>
4	RE	0	R/W	<p>接收允许</p> <p>当该位置为 1, 则可接收。在此状态, 如果检测到异步模式时的开始位或时钟同步模式时的同步时钟输入, 就会开始串行接收。然而, 在置位为 1 前, 必须要通过 SMR 设定接收格式。</p> <p>当该位清 0, 接收动作被禁止。即使清 0, RDRF、FER、PER、ORER 的各标志也不会受影响, 仍保持原状态。</p>
3	MPIE	0	R/W	<p>多处理器中断允许 (在异步模式下 SMR 的 MP = 1 时有效)</p> <p>智能卡接口模式下, 此位写入 0 后才可使用。</p> <p>接收包含 MPB = 0 的接收数据时, 如果检测到从 RSR 向 RDR 传送接收数据及检测到接收错误, 则不设置 SSR 的 RDRF、FER、ORER 各标志。如果接收包含 MPB = 1 的接收数据, SSR 的 MPB 位置 1, 将 MPIE 位自动清 0, 就会允许发生 RXI、ERI 中断请求 (SCR 的 TIE、RIE 位被置 1 时), 并允许 FER、ORER 标志置位。</p>
2	TEIE	0	R/W	<p>发送结束中断允许</p> <p>智能卡接口模式下, 此位写入 0 后才可使用。</p> <p>可通过从 SSR 的 TDRE 标志读取 1 后, 清 0 或者将 TEND 标志清 0 或将 TEIE 位清 0 来解除 TEI 中断请求。</p>
1 0	CKE1 CKE0	0 0	R/W R/W	<p>时钟允许 1~0</p> <p>该位控制 SCK 管脚的时钟输出。在 GSM 模式中, 可以把时钟输出切换为动态。详细请参考 12.7.9 时钟输出控制。</p> <p>SMR 的 GM=0 时</p> <ul style="list-style-type: none"> 00: 禁止输出 (SCK 管脚可当作 I/O 端口使用) 01: 时钟输出 1x: 保留 <p>SMR 的 GM=1 时</p> <ul style="list-style-type: none"> 00: 固定为低电平输出 01: 时钟输出 10: 固定为高电平输出 11: 时钟输出

【符号说明】 x: Don't care

12.3.7 串行通信状态寄存器 (SSR)

SSR 由 SCI 的状态标志和发送接收多处理器位构成。TDRE、RDRF、ORER、PER、FER 只能被清 0。SSR 在普通模式和智能卡接口模式有部分位的功能不同。

- 普通串行通信接口模式 (SCMR 的 SMIF = 0 时)

位	位名	初始值	R/W	说明
7	TDRE	1	R/(W)*	<p>发送数据寄存器为空 表示 TDR 内是否有发送数据。</p> <p>[置位条件]</p> <ul style="list-style-type: none"> • SCR 的 TE 位为 0 时 • 从 TDR 向 TSR 传送数据时可以向 TDR 写入数据时 <p>[清除条件]</p> <ul style="list-style-type: none"> • 当读取 TDRE = 1 的状态后, 置 TDRE = 0。 • 通过 TXI 中断请求, 用 DMAC 向 TDR 传送发送数据时
6	RDRF	0	R/(W)*	<p>接受数据寄存器满 表示 RDR 内是否有接收数据。</p> <p>[置位条件]</p> <ul style="list-style-type: none"> • 接收正常结束, 将接收数据从 RSR 传送到 RDR <p>[清除条件]</p> <ul style="list-style-type: none"> • 当读取 RDRF = 1 后, 将 RDRF 写入 0 • 由于 RXI 中断, 用 DMAC 从 RDR 传送接收数据时即使清除 SCR 的 RE, RDR 及 RDRF 不受影响, 仍保持原状态。如果 RDRF 标志保持被置为 1 的状态, 完成接收数据, 会发生溢出错误, 请注意, 接收数据会丢失。
5	ORER	0	R/(W)*	<p>溢出错误 表示接收时发生溢出错误, 而异常结束。</p> <p>[置位条件]</p> <ul style="list-style-type: none"> • 在 RDRF = 1 的状态接收下一个数据时在 RDR 中, 保持发生溢出错误前的接收数据, 后面接收的数据会丢失。在 ORER 标志被置 1 的状态, 不能继续之后的串行接收。然而, 在时钟同步模式下, 也不能继续串行发送。 <p>[清除条件]</p> <ul style="list-style-type: none"> • 读取 ORER = 1 的状态后, 写入 0 时即使清除 SCR 的 RE, ORER 也不会受影响, 仍保持原状态。

位	位名	初始值	R/W	说明
4	FER	0	R/(W)*	<p>帧错误</p> <p>异步模式下接收时，会发生帧错误，而异常结束。</p> <p>[置位条件]</p> <ul style="list-style-type: none"> 停止位为 0 时 <p>当为 2 停止模式时，只判断第一个停止位是否是 1，不检查第二个停止位。发生帧错误时的接收数据虽然会被传送到 RDR，但是 RDRF 标志不会被置位。甚至在 FER 标志被置 1 的状态，也不能继续之后的串行接收。在时钟同步模式下，也不能继续串行发送。</p> <p>[清除条件]</p> <ul style="list-style-type: none"> 读取 1 的状态后，写入 0 时 <p>即使清除 SCR 的 RE，FER 也不会受影响，仍保持以前的状态。</p>
3	PER	0	R/(W)*	<p>奇偶校验错误</p> <p>表示在异步模式下，发生奇偶校验错误，异常结束。</p> <p>[置位条件]</p> <ul style="list-style-type: none"> 接收过程中，检测出奇偶校验错误时 <p>发生奇偶校验错误时的接收数据虽然被传送到 RDR，但是 RDRF 标志不会被置位。然而在 PER 标志被置 1 的状态，不能继续以后的串行接收。在时钟同步模式下，也不能继续串行发送。</p> <p>[清除条件]</p> <ul style="list-style-type: none"> 读取 PER = 1 的状态后，写入 0 时 <p>即使清除 SCR 的 RE，PER 也不受影响，仍保持以前的状态。</p>
2	TEND	1	R	<p>发送结束</p> <p>表示完成发送。</p> <p>[置位条件]</p> <ul style="list-style-type: none"> SCR 的 TE 为 0 时 发送字符的最末尾位时，TDRE 为 1 时 <p>[清除条件]</p> <ul style="list-style-type: none"> 读取 TDRE = 1 的状态后，向 TDRE 中写入 0 时 通过 TXI 中断请求，用 DMAC 向 TDR 传送发送数据时
1	MPB	0	R	<p>多处理器位</p> <p>在接收数据时，MPB 保存多处理器位的值。当 SCR 中的 RE 清 0 时，仍保持以前的状态，不改变。</p>
0	MPBT	0	R/W	<p>多处理器位传送</p> <p>设定附加在发送数据中多处理器位的值。</p>

【注】 * 只能在要清除标志中写入 0。

• 智能卡接口模式 (SCMR 的 SMIF = 1 时)

位	位名	初始值	R/W	说明
7	TDRE	1	R/(W)*	<p>传输数据寄存器为空 表示 TDR 内是否有发送数据。</p> <p>[置位条件]</p> <ul style="list-style-type: none"> 当 SCR 中的 TE 位为 0 时 将数据从 TDR 传送到 TSR 时 <p>[清除条件]</p> <ul style="list-style-type: none"> 读取 TDRE = 1 的状态后, 将 TDRE 写入 0 由于 TXI 中断请求, 用 DMAC 向 TDR 传送发送数据时
6	RDRF	0	R/(W)*	<p>接收数据寄存器满 表示 RDR 内是否有接收数据。</p> <p>[置位条件]</p> <p>接收正常结束, 从 RSR 向 RDR 传送接收数据时</p> <p>[清除条件]</p> <ul style="list-style-type: none"> 读取 1 的状态后, 写入 0 时 由于 RXI 中断, 用 DMAC 从 RDR 传送数据时 <p>即使清除 SCR 的 RE, RDRF 仍然保持不受影响的状态。 保持 RDRF 标志被置为 1 的状态, 接收完下一个数据后, 就会发生溢出错误, 接收数据会丢失, 请注意。</p>
5	ORER	0	R/(W)*	<p>溢出错误 表示接收时发生溢出错误而异常结束。</p> <p>[置位条件]</p> <ul style="list-style-type: none"> 在 RDRF = 1 的状态接收下一个数据时 <p>RDR 中, 在发生溢出错误前保持接收数据, 之后接收的数据会丢失。甚至, 在 ORER = 1 的状态, 不能继续以后的串行接收。而在时钟同步模式下, 也不能继续串行发送。</p> <p>[清除条件]</p> <ul style="list-style-type: none"> 读取 1 的状态后, 写入 0 时 <p>即使清除 SCR 的 RE, ORER 不受影响, 仍保持以前的状态。</p>
4	ERS	0	R/(W)*	<p>错误信号状态 表示在发送时, 从接收方返回的错误信号的状态。</p> <p>[置位条件]</p> <ul style="list-style-type: none"> 采样到错误信号低电平时 <p>[清除条件]</p> <ul style="list-style-type: none"> 读取 1 的状态后, 写入 0 时 <p>即使清除 SCR 的 TE, ERS 仍保持不受影响的状态。</p>

位	位名	初始值	R/W	说明
3	PER	0	R/(W)*	<p>奇偶校验错误</p> <p>该位表示在异步模式下，进行附加奇偶校验的接收时，发生奇偶校验错误，异常结束。</p> <p>[置位条件]</p> <ul style="list-style-type: none"> 接收中检测出奇偶校验错误时 发生奇偶校验错误时的接收数据虽然被传送到 RDR，但是 RDRF 标志不会被置位。甚至，在 PER = 1 的状态，不能继续以后的串行接收。且在时钟同步模式下，也不能继续串行发送。 <p>[清除条件]</p> <ul style="list-style-type: none"> 读取 1 的状态后，写入 0 时 即使清除 SCR 的 RE，PER 也不受影响，仍保持以前的状态。
2	TEND	1	R	<p>发送结束</p> <p>没有来自接收方的错误信号的应答，直到可以向 TDR 传送下一个发送数据时，该位就会被置位。</p> <p>[置位条件]</p> <ul style="list-style-type: none"> SCR 的 TE = 0、ERS = 0 时 发送 1 字节的数据，一定时间后，ERS = 0、TDRE = 1 时。 根据寄存器设定的不同，设置时序也有以下不同。 GM = 0、BLK = 0 时，从发送开始 12.5etu 后 GM = 0、BLK = 1 时，从发送开始 11.5etu 后 GM = 1、BLK = 0 时，从发送开始 11.0etu 后 GM = 1、BLK = 1 时，从发送开始 11.0etu 后 <p>[清除条件]</p> <ul style="list-style-type: none"> 读取 TDRE = 1 的状态后，在 TDRE 标志写入 0 时 由于 TXI 中断请求，用 DMAC 向 TDR 传送发送数据时
1	MPB	0	R	<p>多处理器位</p> <p>该位在智能卡接口模式中不能使用。</p>
0	MPBT	0	R/W	<p>多处理器位传送</p> <p>智能卡接口中，在此位写入 0 后才可使用。</p>

【注】 * 只能在要清除标志中写入 0。

12.3.8 智能卡模式寄存器 (SCMR)

SCMR 是选择智能卡接口及其通信数据格式的寄存器。

位	位名	初始值	R/W	说明
7~4	—	全部为 1	—	保留位 读取时常读出 1。
3	DIR	0	R/W	数据传输方向 选择串行 / 并行转换方向。 0: 以 LSB 优先发送接收 1: 以 MSB 优先发送接收 只有在发送接收格式为 8 位数据时才有效。
2	INV	0	R/W	数据反转 反转发送 / 接收数据的逻辑层。INV 位不影响奇偶校验位数据的逻辑层。 反转奇偶校验位时, 请反转 SMR 的 O/\bar{E} 位。 0: 直接发送 TDR 的内容, 把接收数据直接保存到 RDR 中 1: 反转 TDR 的内容发送, 反转接收数据, 保存到 RDR 中
1	—	1	—	保留位 读取时常读出 1。写入定义为无效。
0	SMIF	0	R/W	智能卡接口 在智能卡接口模式下, 要使其运行就要置位为 1。 0: 普通异步模式或时钟同步模式 1: 智能卡接口模式

12.3.9 串行扩展模式寄存器 A_0 (SEMRA_0)

SEMRA_0 可进行时钟同步模式时的 SCI_0 选择功能的选择、异步模式时的基本时钟设定、时钟源的选择、传送率的自动设定。选择平均传送率时的基本时钟例如图 12.3 所示，选择 TPU 时钟输入时的设定例如图 12.4 所示。

位	位名	初始值	R/W	说明																																				
7	SSE	0	R/W	<p>SCI_0 选择允许</p> <p>时钟同步模式下，可选择外部时钟输入时的 SCI_0 选择功能。SSE 的设定，在时钟同步模式下 (SMR 寄存器的 C/A 位 = 1)，外部时钟输入 (SCR 寄存器的 CK1 位 = 1) 时有效。</p> <p>0: SCI_0 选择功能无效 1: SCI_0 选择功能有效</p> <p>SCI_0 选择功能有效时，如果向 PG1/IRQ7 管脚输入 1，TxD0 输出就会变为高阻抗状态，SCK0 时钟输入变为固定高电平输入。</p>																																				
6 5 4	TCS2 TCS1 TCS0	0 0 0	R/W R/W R/W	<p>TPU 时钟选择</p> <p>异步模式下，时钟源选择 TPU 时钟输入 (ACS3 ~ 0 = B'0100) 时，通过 TPU 时钟的组合就会生成串行传送时钟。</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th></th> <th>基本时钟</th> <th>时钟允许</th> <th>TCLKA</th> <th>TCLKB</th> <th>TCLKC</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>TIOCA1</td> <td>TIOCA2</td> <td>如左所记基本时钟</td> <td>管脚输入</td> <td>管脚输入</td> </tr> <tr> <td>001</td> <td>TIOCA0 TIOCC0</td> <td>TIOCA1</td> <td>管脚输入</td> <td>如左所记基本时钟</td> <td>管脚输入</td> </tr> <tr> <td>010</td> <td>TIOCA0</td> <td>TIOCA1&TIOCA2</td> <td>管脚输入</td> <td>如左所记基本时钟</td> <td>管脚输入</td> </tr> <tr> <td>011</td> <td>TIOCA0 TIOCC0</td> <td>TIOCA1&TIOCA2</td> <td>管脚输入</td> <td>如左所记基本时钟</td> <td>管脚输入</td> </tr> <tr> <td>1xx</td> <td colspan="5">保留 (禁止设定)</td> </tr> </tbody> </table> <p>【注】在 E6000 仿真器中不支持本功能。图 12.4 所示的是其设定例子。以上表中的运算符号分别表示 & 为 AND (逻辑与)， 为 OR (逻辑或)。</p>		基本时钟	时钟允许	TCLKA	TCLKB	TCLKC	000	TIOCA1	TIOCA2	如左所记基本时钟	管脚输入	管脚输入	001	TIOCA0 TIOCC0	TIOCA1	管脚输入	如左所记基本时钟	管脚输入	010	TIOCA0	TIOCA1&TIOCA2	管脚输入	如左所记基本时钟	管脚输入	011	TIOCA0 TIOCC0	TIOCA1&TIOCA2	管脚输入	如左所记基本时钟	管脚输入	1xx	保留 (禁止设定)				
	基本时钟	时钟允许	TCLKA	TCLKB	TCLKC																																			
000	TIOCA1	TIOCA2	如左所记基本时钟	管脚输入	管脚输入																																			
001	TIOCA0 TIOCC0	TIOCA1	管脚输入	如左所记基本时钟	管脚输入																																			
010	TIOCA0	TIOCA1&TIOCA2	管脚输入	如左所记基本时钟	管脚输入																																			
011	TIOCA0 TIOCC0	TIOCA1&TIOCA2	管脚输入	如左所记基本时钟	管脚输入																																			
1xx	保留 (禁止设定)																																							
3	ABCS	0	R/W	<p>异步基本时钟选择</p> <p>选择异步模式中 1 位期间的基本时钟。</p> <p>ABCS 的设定在异步模式 (SMR 寄存器的 C/A 位 = 0) 时有效。</p> <p>0: 以频率为 16 倍传送率的基本时钟运行 1: 以频率为 8 倍传送率的基本时钟运行</p>																																				

位	位名	初始值	R/W	说明
2	ACS2	0	R/W	选择异步时钟源 通过与 SEMRB_0 (串行扩展模式寄存器 B_0) 的位 7 (ACS3) 组合, 在异步模式下, 选择时钟源。选择平均传送率时, 与 ABCS 的设定无关, 基本时钟会被自动设定。并且, 平均传送率支持 10.667MHz、16MHz 及 24MHz 的运行频率。其他不支持。 请在异步模式下 (SMR 寄存器的 C/A 位 = 0), 外部时钟输入时 (SCR 寄存器的 CK1 位 = 1) 时设定 ACS3 ~ 0。图 12.3、图 12.4 所示的是设定例。 ACS 3 2 1 0 0 0 0 0: 外部时钟输入 0 0 0 1: 选择 $\phi = 10.667\text{MHz}$ 专用的平均传送率 115.152kbps (以频率为 16 倍的传送率的基本时钟运行) 0 0 1 0: 选择 $\phi = 10.667\text{MHz}$ 专用平均传送率 460.606kbps (以频率为 8 倍传送率的基本时钟运行) 0 0 1 1: 选择 $\phi = 16\text{MHz}$ 专用平均传送率 921.569kbps (以频率为 8 倍传送率的基本时钟运行) 0 1 0 0: TPU 时钟输入 此信号由 TIOCA0、TIOCC0、TIOCA1 和 TIOCA2 生成, 可用作基本时钟, 作为 TPU_0~TPU_2 的比较匹配、PWM 输出等。但是 TIOCA1 管脚和 TIOCA2 管脚会成为输出管脚, 所以 IRQ0 和 IRQ1 输入就不能使用, 请注意。 0 1 0 1: 选择 $\phi = 16\text{MHz}$ 专用平均传送率 115.196kbps (以频率为 16 倍传送率的基本时钟运行) 0 1 1 0: 选择 $\phi = 16\text{MHz}$ 专用平均传送率 460.784kbps (以频率为 8 倍传送率的基本时钟运行) 0 1 1 1: 选择 $\phi = 16\text{MHz}$ 专用平均传送率 720kbps (以频率为 8 倍传送率的基本时钟运行) 1 0 0 0: 选择 $\phi = 24\text{MHz}$ 专用平均传送率 115.132kbps * (以频率为 16 倍传送率的基本时钟运行) 1 0 0 1: 选择 $\phi = 24\text{MHz}$ 专用平均传送率 460.526kbps * (以频率为 16 倍传送率的基本时钟运行) 1 0 1 0: 选择 $\phi = 24\text{MHz}$ 专用平均传送率 720kbps * (以频率为 8 倍传送率的基本时钟运行) 1 0 1 1: 选择 $\phi = 24\text{MHz}$ 专用平均传送率 921.053kbps * (以频率为 8 倍传送率的基本时钟运行) 1 1 X X: 保留 (禁止设定)
1	ACS1	0	R/W	
0	ACS0	0	R/W	

【注】 * E6000 仿真器不支持 24MHz 专用的平均传送率选择功能 (ACS3 ~ 0 = B'10XX)。

12.3.10 串行扩展模式寄存器 B_0 (SEMRB_0)

可以通过 SEMRB_0 和 SEMRA_0 组合, 选择时钟源、自动设定传送率及控制 TPU 生成传送时钟时的端口 I 管脚 (P16、P14、P12、P10)。

【注】 在 E6000 仿真器中不支持 SEMRB_0。

位	位名	初始值	R/W	说明
7	ACS3	0	R/W	异步时钟源选择 通过与 ACS2 ~ ACS0 (SEMRA_0 位 2 ~ 0) 的组合, 选择异步模式中的时钟源。详情请参考 12.3.9 串行扩展模式寄存器 A_0 (SEMRA_0)。
6~4	—	不定	—	保留位 写入时请写入 0。
3	TIOCA2E	1	R/W	TIOCA2 输出允许 控制向 P16 管脚输出的 TIOCA2 的输出。 仅将 TPU 的 TIOCA2 输出用于生成传送时钟时; 将本位设定为 0, 就可以把 P16 管脚作为其他用途使用。 0: 禁止 TPU 的 TIOCA2 输出 1: 允许 TPU 的 TIOCA2 输出
2	TIOCA1E	1	R/W	TIOCA1 输出允许 控制向 P14 管脚输出的 TIOCA1 的输出。 仅将 TPU 的 TIOCA1 输出用于生成传送时钟时; 将本位设定为 0, 就可以把 P14 管脚作为其他用途使用。 0: 禁止 TPU 的 TIOCA1 输出 1: 允许 TPU 的 TIOCA1 输出
1	TIOCC0E	1	R/W	TIOCC0 输出允许 控制向 P12 管脚输出的 TIOCC0 的输出。 仅将 TPU 的 TIOCC0 输出用于生成传送时钟时; 将本位设定为 0, 就可以把 P12 管脚作为其他用途使用。 0: 禁止 TPU 的 TIOCC0 输出 1: 允许 TPU 的 TIOCC0 输出
0	TIOCA0E	1	R/W	TIOCA0 输出允许 控制向 P10 管脚输出的 TIOCA0 的输出。 仅将 TPU 的 TIOCA0 输出用于生成传送时钟时; 将本位设定为 0, 就可以把 P10 管脚作为其他用途使用。 0: 禁止 TPU 的 TIOCA0 输出 1: 允许 TPU 的 TIOCA0 输出

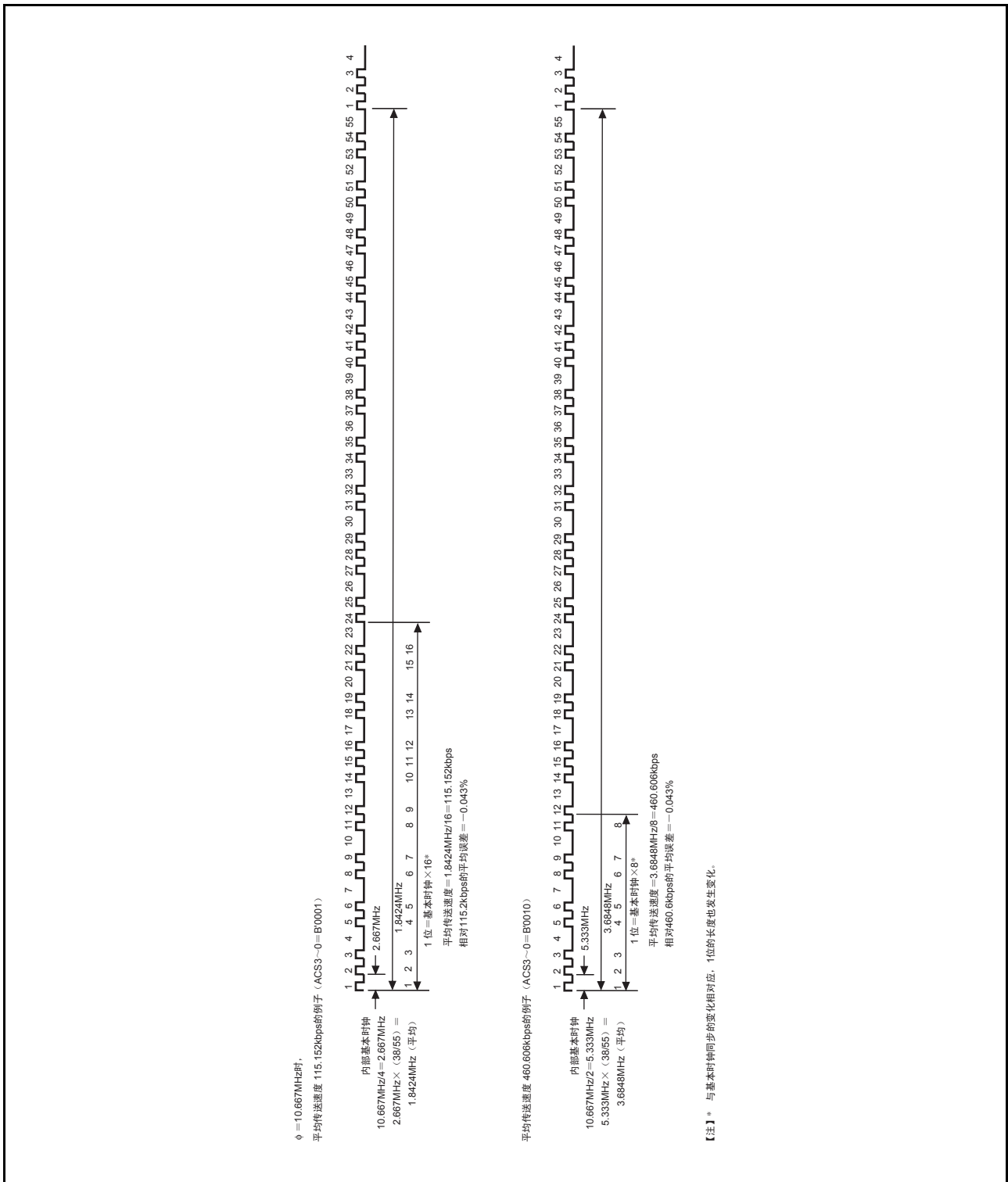


图 12.3 选择平均传送率时的基本时钟例 (1)

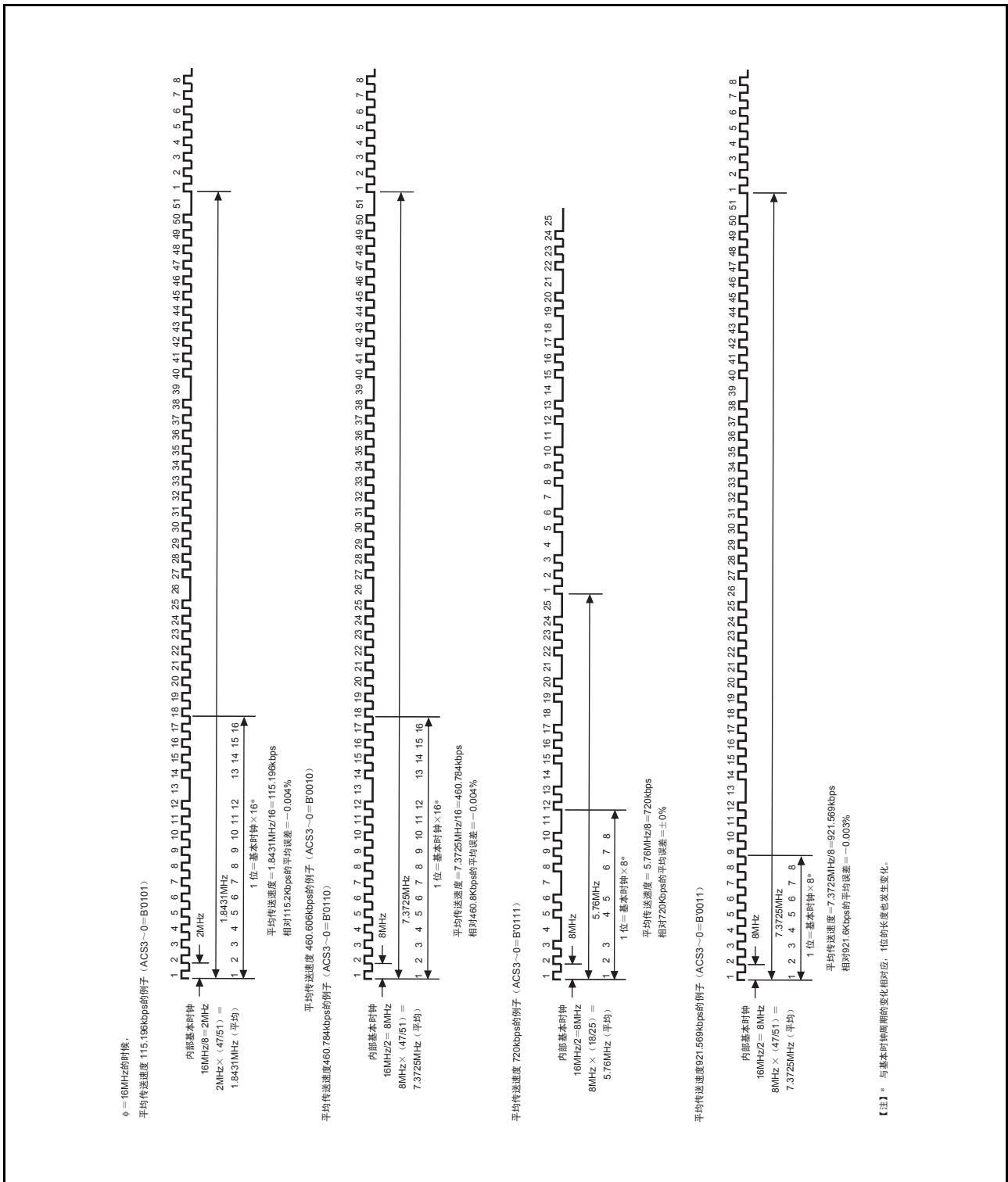


图 12.3 选择平均传送率时的基本时钟例 (2)

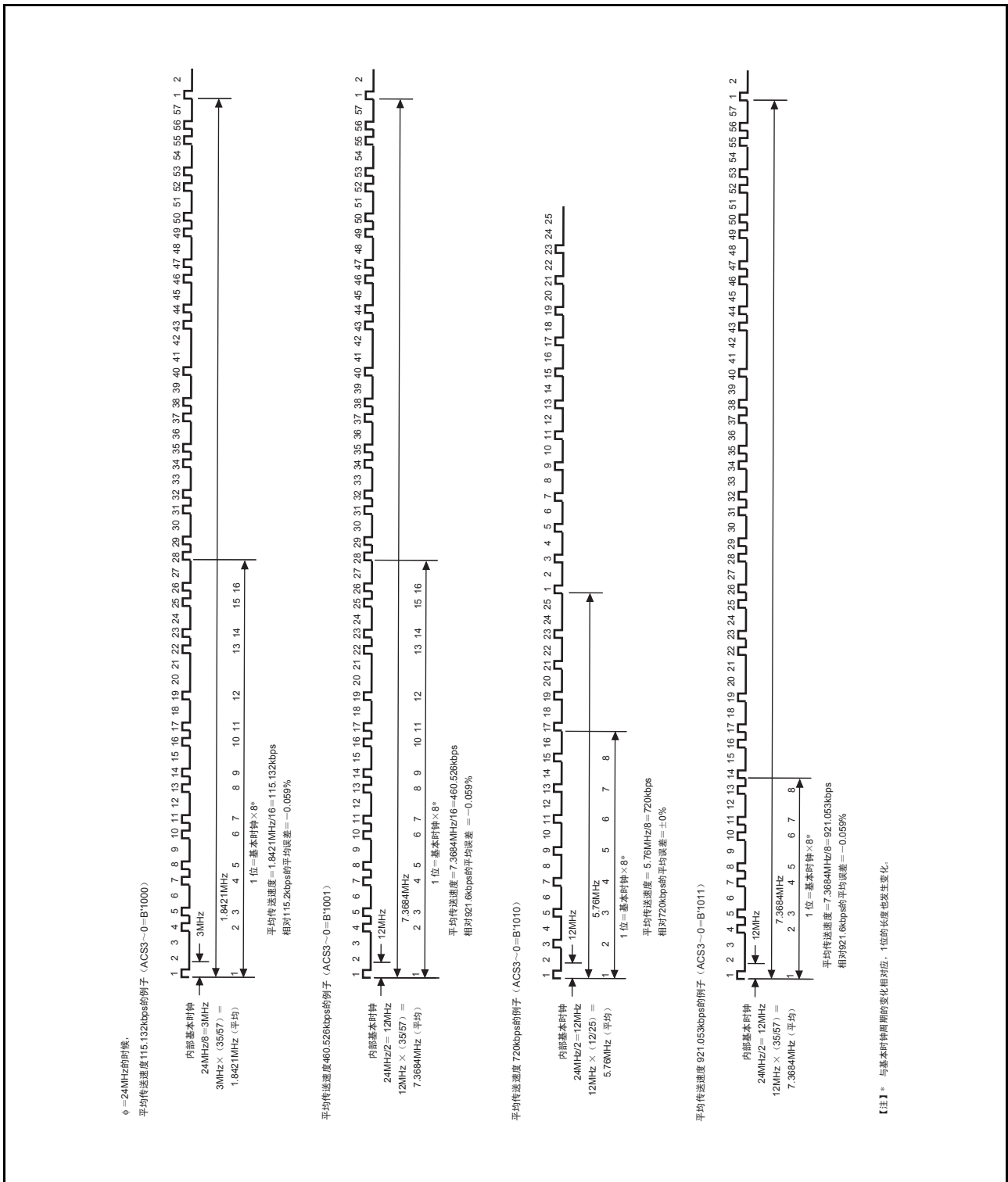


图 12.3 选择平均传送率时的基本时钟例 (3)

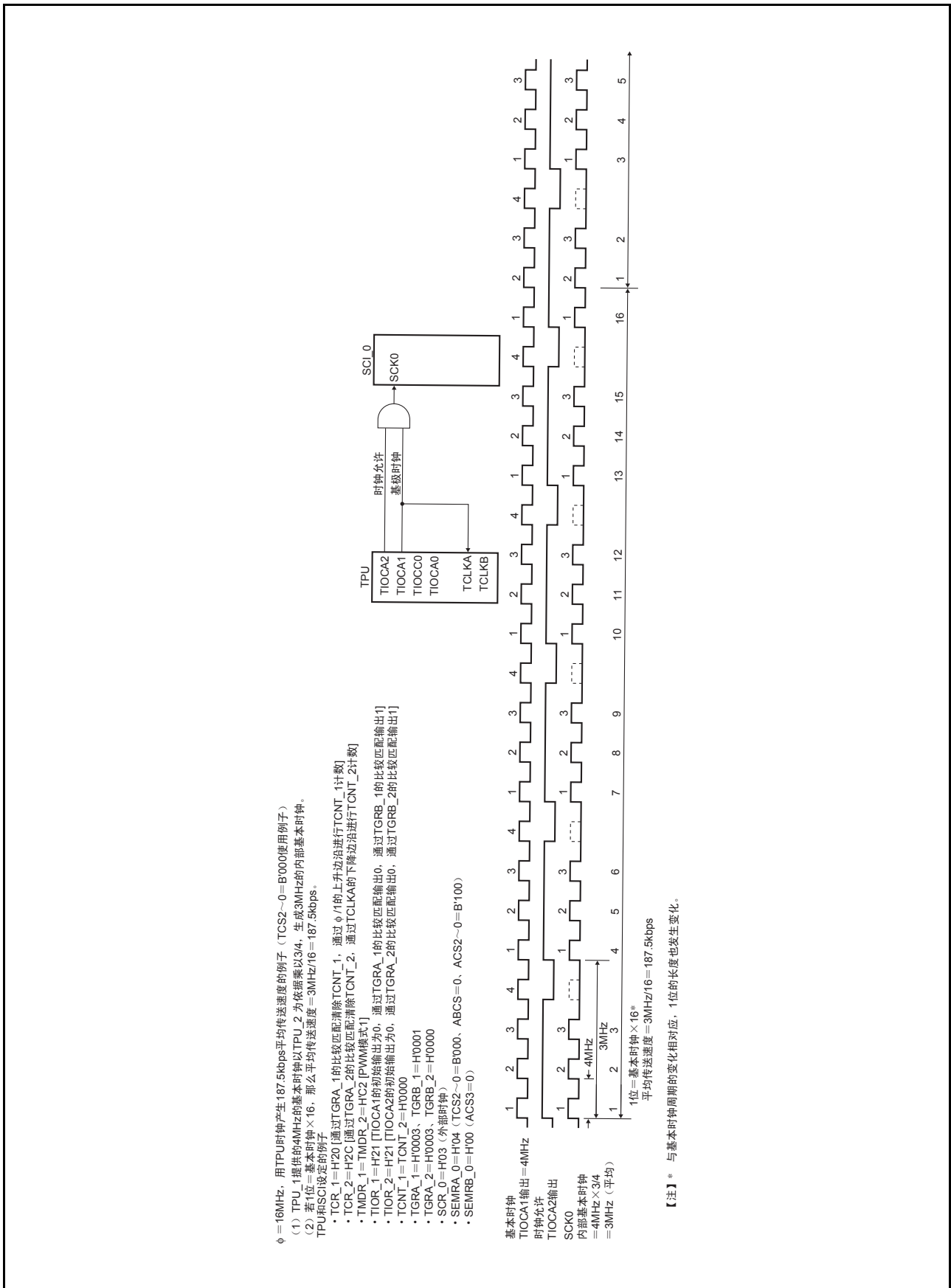


图 12.4 TPU 时钟输入时的平均传输率设定例 (1)

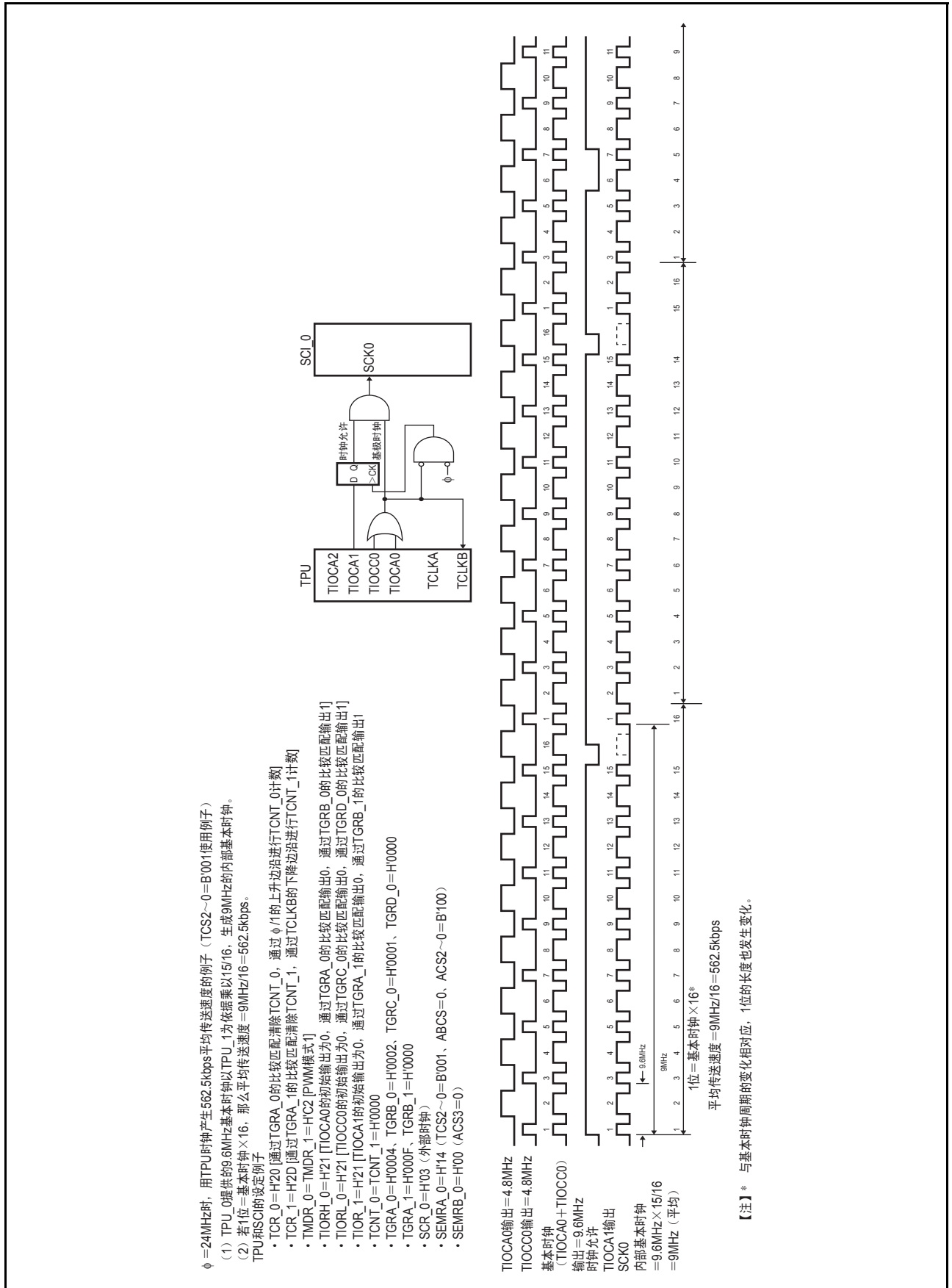


图 12.4 TPU 时钟输入时的平均传送率设定例 (2)

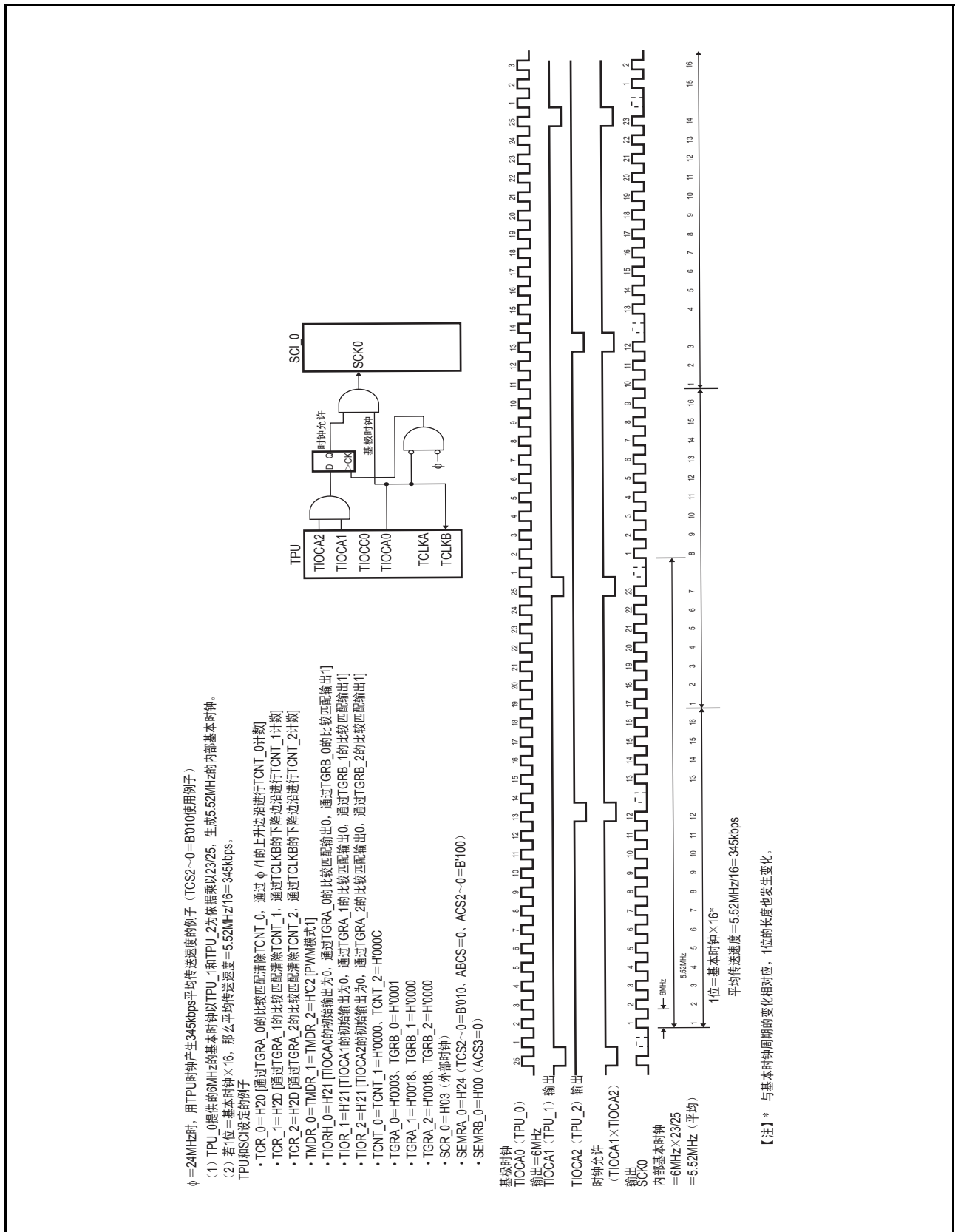


图 12.4 TPU 时钟输入时的平均传输率设定例 (3)

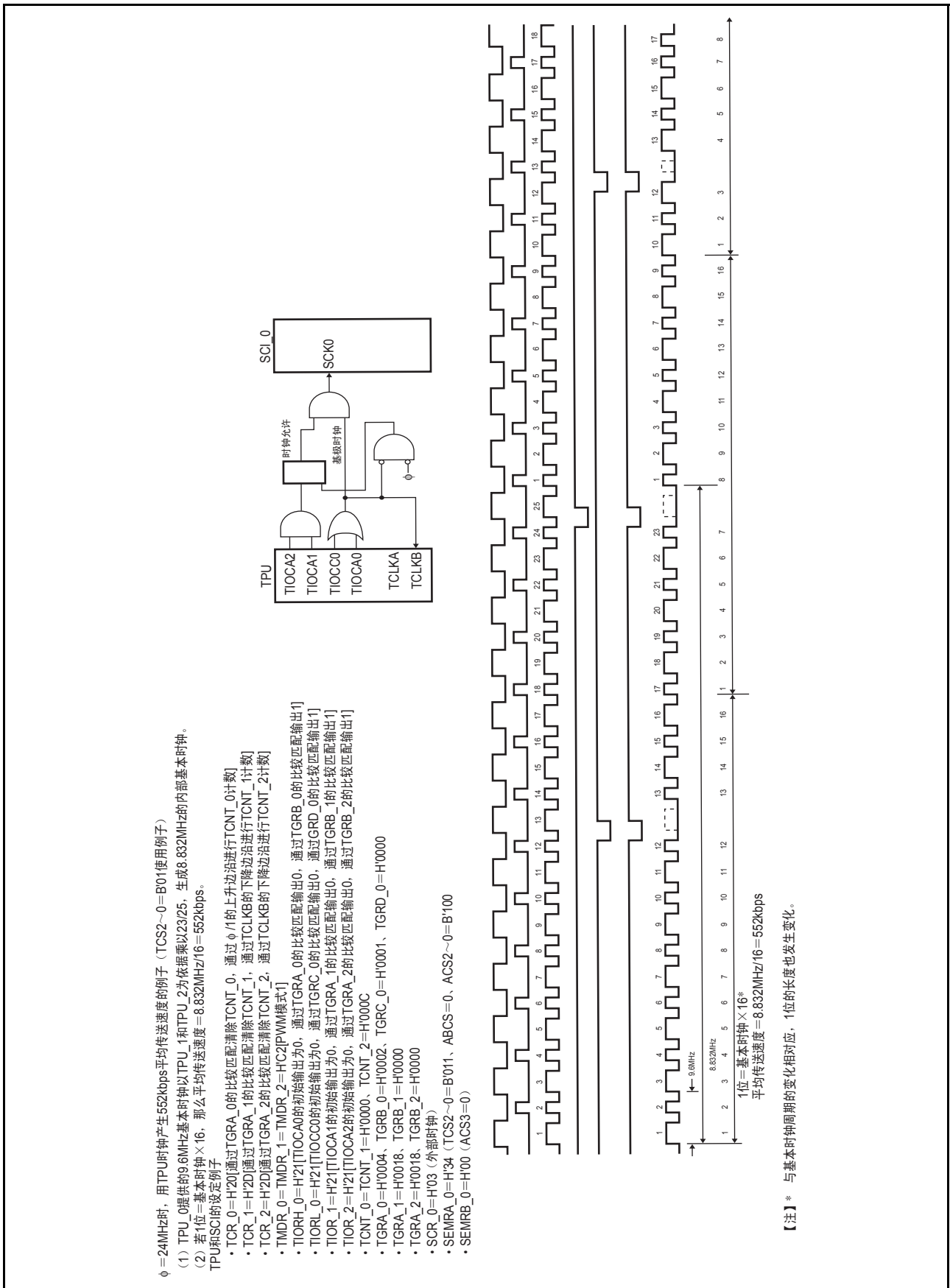


图 12.4 TPU 时钟输入时的平均传送率设定例 (4)

12.3.11 比特率寄存器 (BRR)

BRR 是调整比特率的 8 位寄存器。由于 SCI 每个通道的波特率发生器是相互独立的，因此每个通道可设定不同的比特率。在普通异步模式、时钟同步模式、智能卡接口模式下 BRR 的设定值 N 和比特率的关系如表 12.2 所示。BRR 的初始值是 H'FF，可随时通过 CPU 进行读 / 写。

表 12.2 BRR 的设定值 N 和比特率 B 的关系

通信模式	ABCS 位	比特率	误差
异步式	0	$B = \frac{\phi \times 10^6}{64 \times 2^{2n-1} \times (N+1)}$	误差 (%) = $\left\{ \frac{\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	1	$B = \frac{\phi \times 10^6}{32 \times 2^{2n-1} \times (N+1)}$	误差 (%) = $\left\{ \frac{\phi \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
时钟同步式	×	$B = \frac{\phi \times 10^6}{8 \times 2^{2n-1} \times (N+1)}$	—————
智能卡接口	×	$B = \frac{\phi \times 10^6}{S \times 2^{2n+1} \times (N+1)}$	误差 (%) = $\left\{ \frac{\phi \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1 \right\} \times 100$

【符号说明】

B: 比特率 (bps)

N: 波特率寄存器 BRR 的设定值 ($0 \leq N \leq 255$)

ϕ : 运行频率 (MHz) [16MHz 时 $\phi = 16$]

n 和 S: 如下表，由 SMR 的设定值决定。

×: Don't care

SMR的设定值		时钟源	n
CKS1	CKS0		
0	0	ϕ	0
0	1	$\phi / 4$	1
1	0	$\phi / 16$	2
1	1	$\phi / 64$	3

SMR的设定值		S
BCP1	BCP0	
0	0	32
0	1	64
1	0	372
1	1	256

普通异步模式 BRR 的值 N 的设定例如表 12.3，各运行频率中可设定的最大比特率如表 12.4 所示。时钟同步模式 BRR 的值 N 的设定例如表 12.6，智能卡接口 BRR 的值 N 的设定例如表 12.8 所示。在智能卡接口模式下可选择传送 1 位数据的基本时钟数 S。详情请参考 12.7.5 接收数据采样时序接收容限。表 12.5、表 12.7 所示的是外部时钟输入时的最大比特率。

异步模式下，将 SCI_0 的串行扩展模式寄存器 A_0 (SEMRA_0) 的 ABCS 位置 1 时的比特率变为表 12.3 的 2 倍。

表 12.3 比特率对应的 BRR 的设定例 (异步模式)

比特率 (bps)	运行频率 ϕ (MHz)											
	2			2.097152			2.4576			3		
	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)
110	1	141	0.03	1	148	-0.04	1	174	-0.26	1	212	0.03
150	1	103	0.16	1	408	0.21	1	127	0.00	1	155	0.16
300	0	207	0.16	0	217	0.21	0	255	0.00	0	77	0.16
600	0	103	0.16	0	108	0.21	0	127	0.00	0	155	0.16
1200	0	51	0.16	0	54	-0.70	0	63	0.00	0	77	0.16
2400	0	25	0.16	0	26	1.14	0	31	0.00	0	38	0.16
4800	0	12	0.16	0	13	-2.48	0	15	0.00	0	19	-2.34
9600	—	—	—	0	6	-2.48	0	7	0.00	0	9	-2.34
19200	—	—	—	—	—	—	0	3	0.00	0	4	-2.34
31250	0	1	0.00	—	—	—	—	—	—	0	2	0.00
38400	—	—	—	—	—	—	1	1	0.00	—	—	—

比特率 (bps)	运行频率 ϕ (MHz)											
	3.6864			4			4.9152			5		
	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)
110	2	64	0.70	2	70	0.03	2	86	0.31	2	88	-0.25
150	1	191	0.00	1	207	0.16	1	255	0.00	2	64	0.16
300	1	95	0.00	1	103	0.16	0	127	0.00	1	129	0.16
600	0	191	0.00	0	207	0.16	0	255	0.00	1	64	0.16
1200	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
2400	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
4800	0	23	0.00	0	25	0.16	0	31	0.00	0	32	-1.36
9600	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73
19200	0	5	0.00	—	—	—	0	7	0.00	0	7	1.73
31250	—	—	—	0	3	0.00	0	4	-1.70	0	4	0.00
38400	0	2	0.00	—	—	—	0	3	0.00	0	3	1.73

比特率 (bps)	运行频率 ϕ (MHz)											
	6			6.144			7.3728			8		
	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)
110	2	106	-0.44	2	108	0.08	2	130	-0.07	2	141	0.03
150	2	77	0.16	2	79	0.00	2	95	0.00	2	103	0.16
300	1	155	0.16	1	159	0.00	1	191	0.00	1	207	0.16
600	0	77	0.16	1	79	0.00	1	95	0.00	1	103	0.16
1200	0	155	0.16	0	159	0.00	0	191	0.00	0	207	0.16
2400	0	77	0.16	0	79	0.00	0	95	0.00	0	103	0.16
4800	0	38	0.16	0	39	0.00	0	47	0.00	0	51	0.16
9600	0	19	-2.34	0	19	0.00	0	23	0.00	0	25	0.16
19200	0	9	-2.34	0	9	0.00	0	11	0.00	0	12	0.16
31250	—	5	0.00	0	5	2.40	—	—	—	0	7	0.00
38400	0	4	-2.34	0	4	0.00	0	5	0.00	—	—	—

比特率 (bps)	运行频率 ϕ (MHz)											
	9.8304			10			12			12.288		
	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)
110	2	174	-0.26	2	177	-0.25	2	212	0.03	2	217	0.08
150	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	31	0.00	0	32	-1.36	0	38	0.16	0	39	0.00
19200	0	15	0.00	0	15	1.73	0	19	-2.34	0	19	0.00
31250	0	9	-1.70	0	9	0.00	—	11	0.00	0	11	2.40
38400	0	7	0.00	0	7	1.73	0	9	-2.34	0	9	0.00

比特率 (bps)	运行频率 ϕ (MHz)											
	14			14.7456			16			17.2032		
	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)
110	2	248	-0.17	3	64	0.70	3	70	0.03	3	75	0.48
150	2	181	0.16	2	191	0.00	2	207	0.16	2	223	0.00
300	2	90	0.16	2	95	0.00	2	103	0.16	2	111	0.00
600	1	181	0.16	1	191	0.00	1	207	0.16	1	223	0.00
1200	1	90	0.16	1	95	0.00	1	103	0.16	1	111	0.00
2400	0	181	0.16	0	191	0.00	0	207	0.16	0	223	0.00
4800	0	90	0.16	0	95	0.00	0	103	0.16	0	111	0.00
9600	0	45	-0.93	0	47	0.00	0	51	0.16	0	55	0.00
19200	0	22	-0.93	0	23	0.00	0	25	0.16	0	27	0.00
31250	0	13	0.00	0	14	-1.70	0	15	0.00	0	16	1.20
38400	—	—	—	0	11	0.00	0	12	0.16	0	13	0.00

比特率 (bps)	运行频率 ϕ (MHz)											
	18			19.6608			20			24		
	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)
110	3	79	-0.12	3	86	0.31	3	88	-0.25	3	106	-0.44
150	2	233	0.16	2	255	0.00	3	64	0.16	3	77	0.16
300	2	116	0.16	2	127	0.00	2	129	0.16	2	155	0.16
600	1	233	0.16	1	255	0.00	2	64	0.16	2	77	0.16
1200	1	116	0.16	1	127	0.00	1	129	0.16	1	155	0.16
2400	0	233	0.16	0	255	0.00	1	64	0.16	1	77	0.16
4800	0	116	0.16	0	127	0.00	0	129	0.16	0	155	0.16
9600	0	58	-0.69	0	63	0.00	0	64	0.16	0	77	0.16
19200	0	28	1.02	0	31	0.00	0	32	-1.36	0	38	0.16
31250	0	17	0.00	0	19	-1.70	0	19	0.00	0	13	0.00
38400	0	14	-2.34	0	15	0.00	0	15	1.73	0	19	-2.34

【注】 是 SEMRA_0 的 ABCS = 0 时的例。ABCS = 1 时，比特率变为 2 倍。

本 LSI 中，请将运行频率设定为 $\phi \geq 6\text{MHz}$ 。

表 12.4 各运行频率中最大比特率 (异步模式)

ϕ (MHz)	最大比特率 (kbps)		n	N	ϕ (MHz)	最大比特率 (kbps)		n	N
	ABCS = 0	ABCS = 1							
2	62.5	125.0	0	0	9.8304	307.2	614.4	0	0
2.097152	65.536	131.027	0	0	10	312.5	625.0	0	0
2.4576	76.8	153.6	0	0	12	375.0	750.0	0	0
3	93.75	187.5	0	0	12.288	384.0	768.0	0	0
3.6864	115.2	230.4	0	0	14	437.5	875.0	0	0
4	125.0	250.0	0	0	14.7456	460.8	921.6	0	0
4.9152	153.6	307.2	0	0	16	500.0	1000.0	0	0
5	156.25	312.5	0	0	17.2032	537.6	1075.2	0	0
6	187.5	375.0	0	0	18	562.5	1125.0	0	0
6.144	192.0	384.0	0	0	19.6608	614.4	1228.8	0	0
7.3728	230.4	460.8	0	0	20	625.0	1250.0	0	0
8	250.0	500.0	0	0	24	750.0	1500.0	0	0

表 12.5 外部时钟输入时的最大比特率 (异步模式)

ϕ (MHz)	外部输入时钟 (MHz)	最大比特率 (kbps)		ϕ (MHz)	外部输入时钟 (MHz)	最大比特率 (kbps)	
		ABCS = 0	ABCS = 1			ABCS = 0	ABCS = 1
2	0.5000	31.25	62.5	9.8304	2.4576	153.6	307.2
2.097152	0.5243	327.68	65.536	10	2.5000	156.25	312.5
2.4576	0.6144	38.4	76.8	12	3.0000	187.5	375.0
3	0.7500	46.875	93.75	12.288	3.0720	192.0	384.0
3.6864	0.9216	57.6	115.2	14	3.5000	218.75	437.0
4	1.0000	62.5	125.0	14.7456	3.6864	230.4	460.8
4.9152	1.2288	76.8	153.6	16	4.0000	250.0	500.0
5	1.2500	78.125	156.25	17.2032	4.3008	268.8	537.6
6	1.5000	93.75	187.5	18	4.5000	281.25	562.5
6.144	1.5360	96.0	192.0	19.6608	4.9152	307.2	614.4
7.3728	1.8432	115.2	230.4	20	5.0000	312.5	625.0
8	2.0000	125.0	250.0	24	6.0000	375.0	750.0

【注】 本 LSI 中，对应的运行频率为 $\phi \cong 6\text{MHz}$ 。

表 12.6 比特率对应的 BRR 的设定例 (时钟同步模式)

比特率 (bps)	运行频率 ϕ (MHz)															
	2		4		6		8		10		16		20		24	
	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N
110	3	70	—	—												
250	2	124	2	249			3	124	—	—	3	249				
500	1	249	2	124			2	249	—	—	3	124	—	—	—	—
1k	1	124	1	249			2	124	—	—	2	249	—	—	—	—
2.5k	0	199	1	99	1	149	1	199	1	249	2	99	2	124	2	149
5k	0	99	0	199	1	74	1	99	1	124	2	199	1	249	2	74
10k	0	49	0	99	0	149	0	199	0	249	1	99	1	124	1	149
25k	0	19	0	39	0	59	0	79	0	99	1	159	0	199	0	239
50k	0	9	0	19	0	29	0	39	0	49	0	79	0	99	0	119
100k	0	4	0	9	0	14	0	19	0	24	0	39	0	49	0	59
250k	0	1	0	3	0	5	0	7	0	9	0	15	0	19	0	23
500k	0	0*	0	1	0	2	0	3	0	4	0	7	0	9	0	11
1M			0	0*			0	1			0	3	0	4	0	5
2M							0	0*			0	1			0	2
2.5M									0	0*			0	1	—	—
4M											0	0*				
5M													0	0*	—	—
6M															0	0*

【符号说明】

空栏 : 不可设定。

— : 虽然可以设定, 但是有误差。

* : 不可连续发送 / 连续接收。

表 12.7 外部时钟输入时的最大比特率 (时钟同步模式)

ϕ (MHz)	外部输入时钟 (MHz)	最大比特率 (Mbps)	ϕ (MHz)	外部输入时钟 (MHz)	最大比特率 (Mbps)
2	0.333	0.333	14	2.333	2.333
4	0.667	0.667	16	2.667	2.667
6	1.000	1.000	18	3.000	3.000
8	1.333	1.333	20	3.333	3.333
10	1.667	1.667	24	4.000	4.000
12	2.000	2.000			

【注】 本 LSI 中, 对应的运行频率为 $\phi \cong 6\text{MHz}$ 。

表 12.8 比特率对应的 BRR 的设定例

(智能卡接口模式下 $n = 0$ 、 $S = 372$ 时)

比特率 (bps)	运行频率 ϕ (MHz)											
	5.00		7.00		7.1424		10.00		10.7136		13.00	
	N	误差 (%)	N	误差 (%)	N	误差 (%)	N	误差 (%)	N	误差 (%)	N	误差 (%)
6720	0	0.01	1	30.00	1	28.57	1	0.01	1	7.14	2	13.33
9600	0	30.00	0	1.99	0	0.00	1	30.00	1	25.00	0	8.99

比特率 (bps)	运行频率 ϕ (MHz)									
	14.2848		16.00		18.00		20.00		24.00	
	N	误差 (%)	N	误差 (%)	N	误差 (%)	N	误差 (%)	N	误差 (%)
6720	2	4.76	2	6.67	3	0.01	3	0.01	4	3.99
9600	1	30.00	1	12.01	2	15.99	2	6.66	2	12.01

表 12.9 各运行频率中最大比特率

(智能卡接口模式)

ϕ (MHz)	最大比特率 (bps)				n	N
	S = 32	S = 64	S = 256	S = 372		
5.00	78125	39063	9766	6720	0	0
6.00	93750	46875	11719	8065	0	0
7.00	109375	54688	13672	9409	0	0
7.1424	111600	55800	13950	9600	0	0
10.00	156250	78125	19531	13441	0	0
10.7136	167400	83700	20925	14400	0	0
13.00	203125	101563	25391	17473	0	0
14.2848	223200	111600	27900	19200	0	0
16.00	250000	125000	31250	21505	0	0
18.00	281250	140625	35156	24194	0	0
20.00	312500	156250	39063	26882	0	0
24.00	375000	187500	46875	32258	0	0

【注】 本 LSI 中，对应的运行频率为 $\phi \cong 6\text{MHz}$ 。

12.4 异步模式的运行

异步模式串行通信的一般格式如图 12.5 所示。一帧数据是从起始位（低电平）开始，按照接收 / 发送数据、奇偶校验位、最终停止位（高电平）的顺序构成。异步模式串行通信中，通信线路通常保持 MARK 状态（高电平）。SCI 监视通信线路，一旦检测出 SPACE 状态（低电平），则视为起始位，开始串行通信。在 SCI 内部，发送和接收部是独立的，可进行全双工通信。且发送和接收部都采用了双缓冲结构，在发送及接收中可读取 / 写入数据，也可连续发送和接收数据。

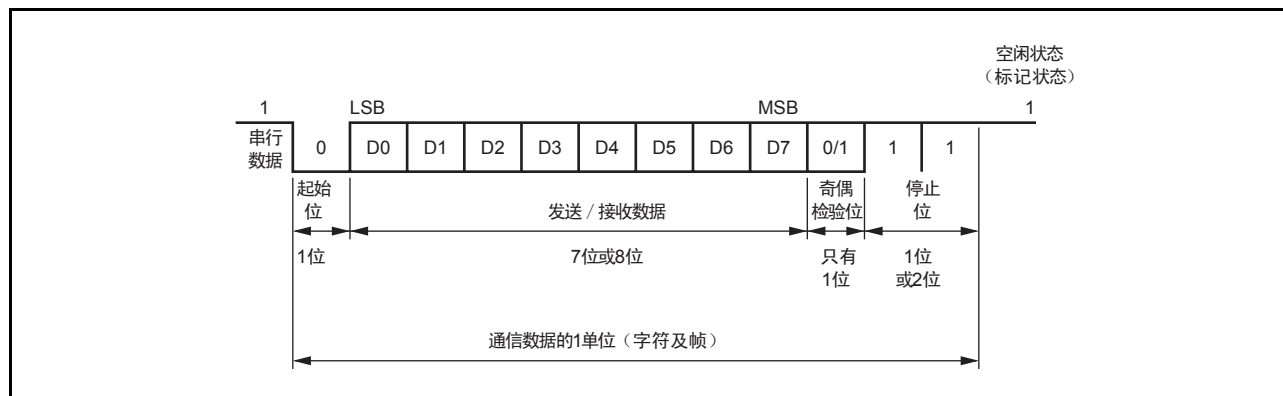


图 12.5 异步模式通信的数据格式（8 位数据 / 有奇偶校验 / 2 停止位的例）

12.4.1 通信格式

在异步模式下，可设定的发送 / 接收信息格式如表 12.10 所示。格式有 12 种，可根据 SMR 的设置来选择。关于多处理器位，请参考 12.5 多处理器通信功能。

表 12.10 串行发送 / 接收格式 (异步模式)

SMR 的设定				串行发送 / 接收格式和帧长度													
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12		
0	0	0	0	S	8位数据								STOP				
0	0	0	1	S	8位数据								STOP	STOP			
0	1	0	0	S	8位数据								P	STOP			
0	1	0	1	S	8位数据								P	STOP	STOP		
1	0	0	0	S	7位数据							STOP					
1	0	0	1	S	7位数据							STOP	STOP				
1	1	0	0	S	7位数据							P	STOP				
1	1	0	1	S	7位数据							P	STOP	STOP			
0	—	1	0	S	8位数据								MPB	STOP			
0	—	1	1	S	8位数据								MPB	STOP	STOP		
1	—	1	0	S	7位数据							MPB	STOP				
1	—	1	1	S	7位数据							MPB	STOP	STOP			

【符号说明】

S : 开始位

STOP : 停止位

P : 奇偶位

MPB : 多处理器位

12.4.2 异步模式下接收数据的采样时序和接收容限

异步模式下，SCI 运行的基本时钟的频率为比特率的 16 倍。接收数据时用基本时钟采样起始位的下降沿，与内部同步。如图 12.6 所示，在基本时钟的第 8 个脉冲上升沿采样接收数据，在各位中央读出数据。因此在异步模式的接收容限如公式 (1) 所示。

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100 (\%) \quad \dots \text{公式 (1)}$$

M: 接收容限 (%)

N: 时钟对应比特率的比值 (ABCS = 0 时 N = 16、ABCS = 1 时 N = 8)

D: 时钟的占空比 (D = 0 ~ 1.0)

L: 帧长度 (L = 9 ~ 12)

F: 时钟频率偏差的绝对值

在公式 (1) 中，F (时钟频率偏差的绝对值) = 0、D (时钟的占空比) = 0.5、N (时钟对应比特率的比值) = 16

$$M = \left\{ 0.5 - \frac{1}{(2 \times 16)} \right\} \times 100 (\%) = 46.875\%$$

但是，此值完全是理论上的估算值，在系统设计时请保持 20 ~ 30% 的余地。

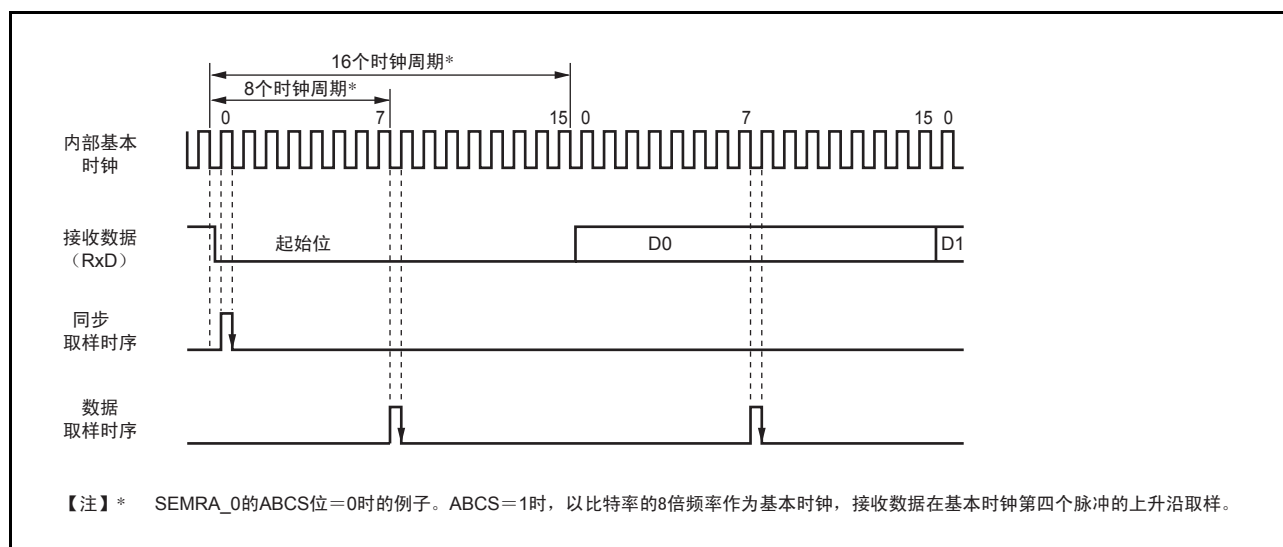


图 12.6 异步模式的接收数据采样时序

12.4.3 时钟

通过设定 SMR 的 $\overline{C/A}$ 和 SCR 的 CKE1、CKE0 位，可任意选择 On-chip 波特率发生器生成的内部时钟或从 SCK 管脚输入的外部时钟，作为 SCI 的发送接收时钟。当使用外部时钟输入时，请在 SCK 管脚输入频率为 16 倍 (ABCS = 0 时)、8 倍 (ABCS = 1 时) 比特率的时钟。同时，选择外部时钟时，也可通过设定 SEMR_0 的 ACS2 ~ ACS0 位，选择平均传送率或者来自 TPU 的基本时钟。

当 SCI 使用内部时钟运行时，如果 CKE1 = 0、CKE0 = 1，就可从 SCK 管脚输出时钟。此时输出的时钟频率与比特率相等，发送数据时的相位如图 12.7 所示，时钟上升沿在发送数据的中央。

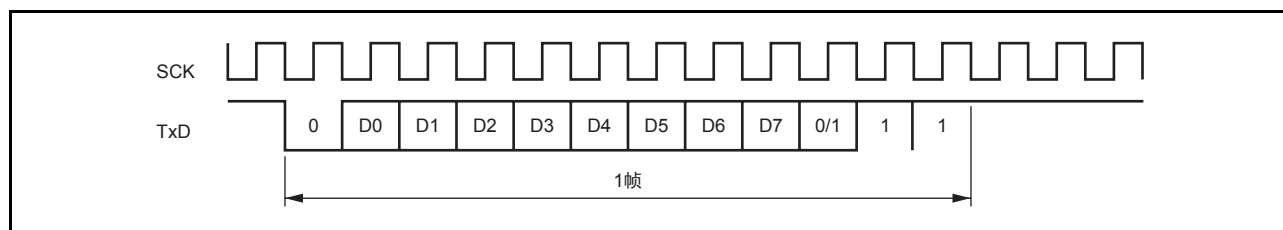


图 12.7 输出时钟和发送数据的相位关系 (异步模式)

12.4.4 SCI 的初始化 (异步模式)

发送 / 接收数据前, 首先把 SCR 的 TE、RE 位清零, 然后按照图 12.8 的流程进行初始化。必须将 TE 位及 RE 位清零后, 再变更运行模式、通信格式等。应特别注意的是, 如果 TE 清零, TDRE 就被置为 1。即使将 RE 也清零, RDRF、PER、FER、ORER 的各标志及 RDR 的内容也不会被初始化。在异步模式下使用外部时钟时, 必须在初始化期间持续保持时钟。

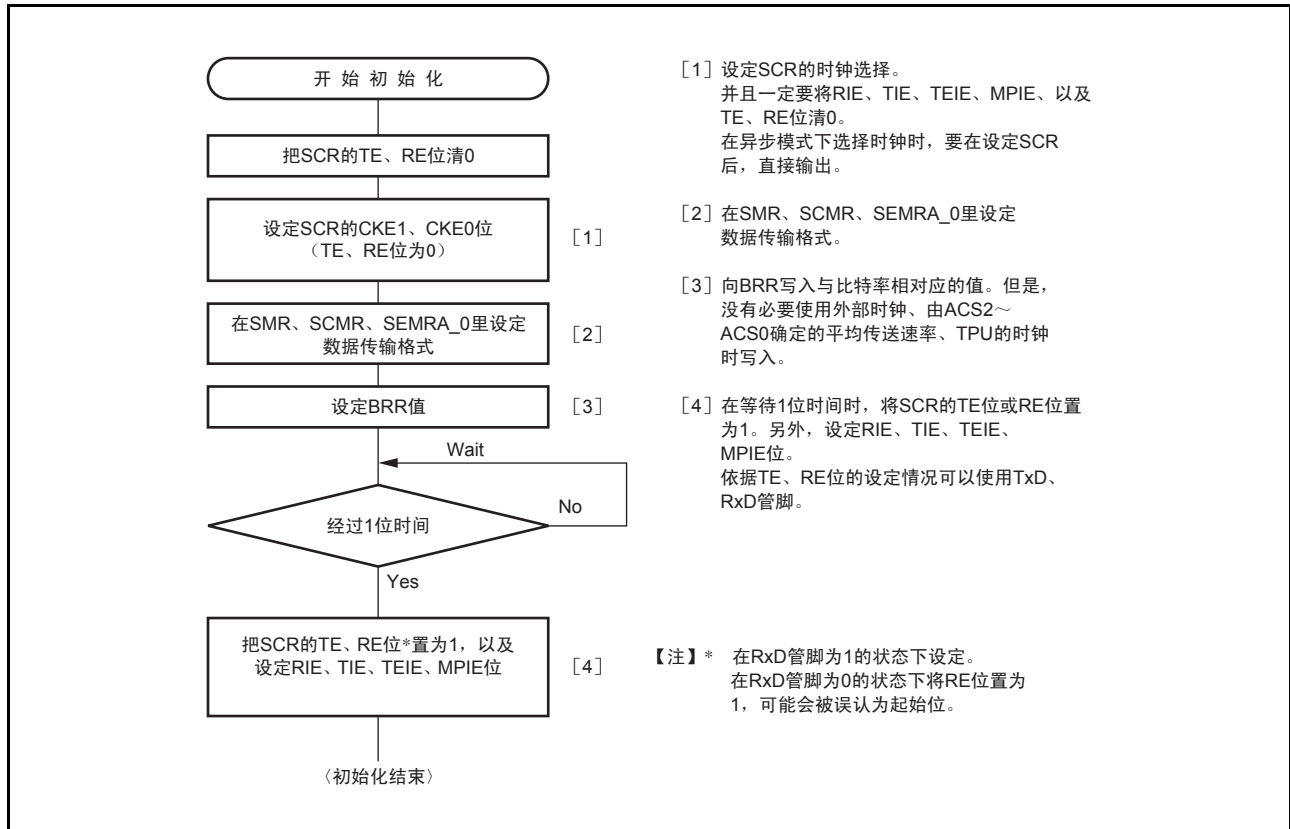


图 12.8 SCI 的初始化流程图

12.4.5 串行数据发送 (异步模式)

图 12.9 所示的是异步模式发送的运行例。发送数据时，SCI 运行步骤如下：

1. SCI 监视 SSR 的 TDRE，如果被清零，则认为数据已被写入 TDR，将数据从 TDR 传送到 TSR 中。
2. 将数据从 TDR 传送到 TSR 后，SCI 把 TDRE 置为 1，开始发送数据。这时，如果 SCR 的 TIE 被置为 1，就会发生 TXI 中断请求。在 TXI 中断程序中，前一个传送的数据发送完成前，在 TDR 中写入下一个发送数据，就可以连续发送信息。
3. 从 TxD 管脚按照起始位、发送数据、奇偶校验位或者多处理器位（根据格式决定该位有无）、停止位的顺序发送。
4. 在发送出停止位的同时 SCI 检查 TDRE。
5. 如果 TDRE 为 0，将发送数据从 TDR 传送到 TSR，直到停止位发送完成后，就会开始下一帧的发送
6. 如果 TDRE 为 1，把 SSR 的 TEND 置为 1，停止位送出后，输出 1，变为 MARK 状态。此时如果 SCR 的 TEIE 被置为 1，就发生 TEI 中断请求。

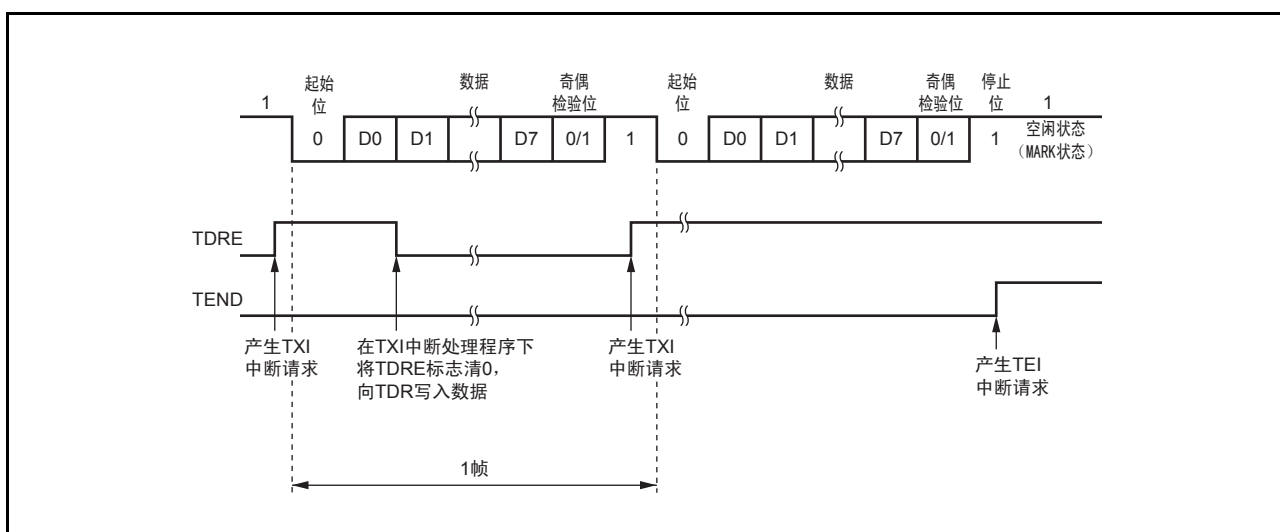


图 12.9 异步模式下发送时的运行例 (8 位数据 / 有奇偶性 / 1 停止位)

图 12.10 串行发送流程图所示的是数据发送流程。

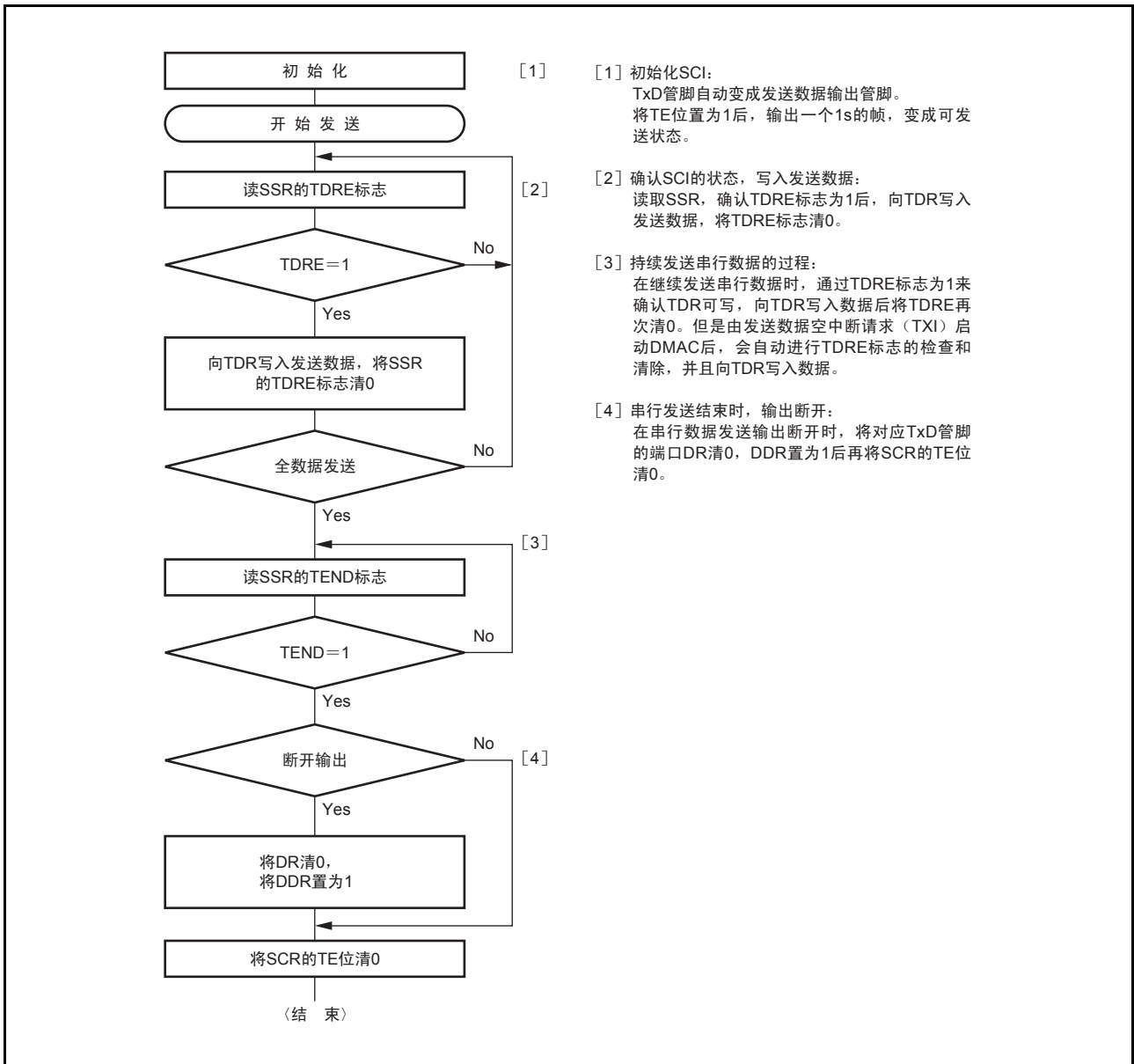


图 12.10 串行发送流程图

12.4.6 串行数据接收 (异步模式)

图 12.11 所示的是异步模式下接收运行例。接收数据时 SCI 运行步骤如下：

1. 监视通信线路，如果检测出起始位，SCI 使内部同步，将接收数据传送到 RSR，并检查奇偶校验位和停止位。
2. 发生溢出错误时（在 SSR 的 RDRF 被置为 1 的状态下，完成下一个数据的接收），要设置 SSR 的 OREER 为 1。此时，如果 SCR 的 RIE 被置为 1，就会发生 ERI 中断请求。接收数据不会被传送到 RDR，且 RDRF 保持置 1 的状态。
3. 检测出奇偶校验错误时，要将 SSR 的 PER 置 1，将接收数据传送到 RDR。此时，如果 SCR 的 RIE 被置为 1，就发生 ERI 中断请求。
4. 检测出帧错误时（停止位为 0），SSR 的 FER 位被置 1，将接收数据传送到 RDR。此时，如果 SCR 的 RIE 被置为 1，就发生 ERI 中断请求。
5. 正常接收时，SSR 的 RDRF 被置位，并将接收数据传送到 RDR。此时，如果 SCR 的 RIE 被置为 1，就发生 RXI 中断请求。可以通过该 RXI 中断处理程序，在下一个数据接收完成前，读取传送到 RDR 中的接收数据，就可连续接收信息。

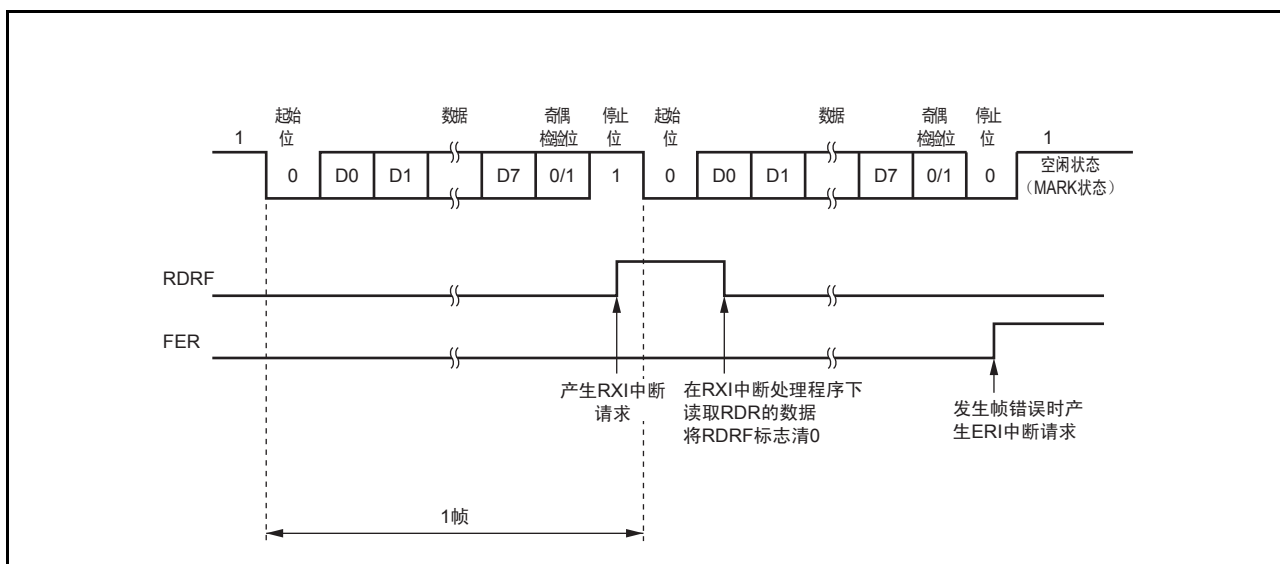


图 12.11 SCI 接收数据时的运行例 (8 位数据 / 有奇偶性 / 1 停止位)

在检测到接收错误时，SSR 各状态标志的状态和接收数据的处理如表 12.11 所示。一旦检测出接收错误，RDRF 就保持接收数据前的状态。在接收错误标志被置位的状态，不能进行以后的接收操作。因此，在继续接收数据前，必须将 ORER、FER、PER 及 RDRF 清零。图 12.12 所示的是接收数据的流程图。

表 12.11 SSR 的状态标志的状态和接收数据的处理

SSR 的状态标志				接收数据	接收错误的状态
RDRF*	ORER	FER	PER		
1	1	0	0	丢失	溢出错误
0	0	1	0	传送到 RDR	帧错误
0	0	0	1	传送到 RDR	奇偶校验错误
1	1	1	0	丢失	溢出错误 + 帧错误
1	1	0	1	丢失	溢出错误 + 奇偶校验错误
0	0	1	1	传送到 RDR	帧错误 + 奇偶校验错误
1	1	1	1	丢失	溢出错误 + 帧错误 + 奇偶校验错误

【注】 * RDRF 保持接收数据前的状态。

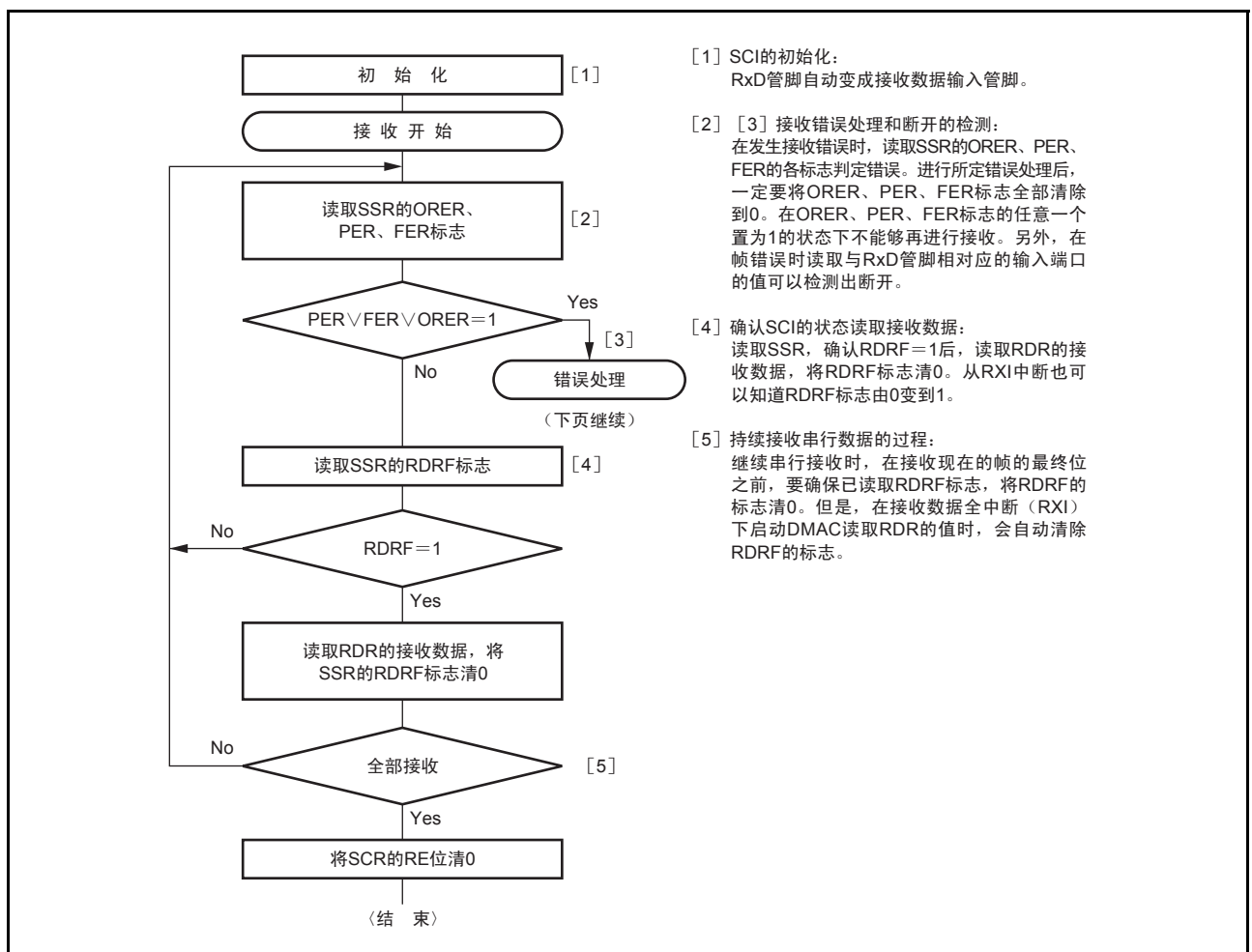


图 12.12 串行接收数据流程图例 (1)

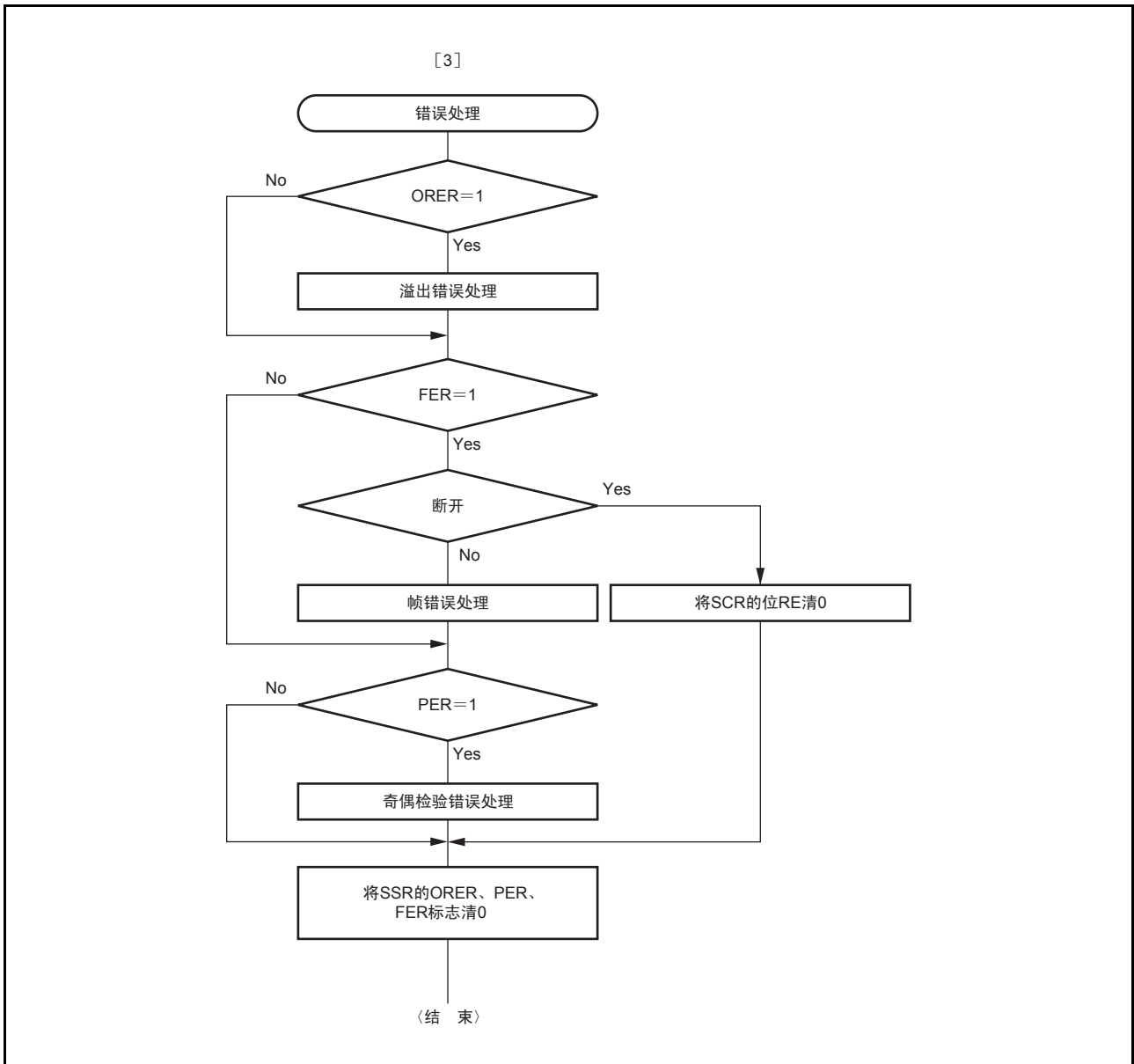


图 12.12 串行接收数据流程图例 (2)

12.5 多处理器通信功能

如果使用多处理器通信功能，就能通过附加了多处理器位的异步串行通信，在多个处理器之间共用通信线路，进行数据的发送和接收。多处理器通信时，给各接收站分别分配了其固有的 ID 代码。串行通信周期由指定接收站的 ID 发送周期和指定接收站对应的数据发送周期构成。ID 发送周期和数据发送周期的区别在于是否以多处理器位运行。多处理器位为 1 时，是 ID 发送周期，多处理器位为 0 时，是数据发送周期。图 12.13 所示的是使用了多处理器格式的处理器之间通信例。发送站首先发送接收站 ID 代码后附加多处理器位为 1 的通信数据，接着发送将多处理器位为 0 的数据附加于发送数据的通信数据。接收站接收到多处理器位为 1 的通信数据后，就会与本站 ID 进行比较，一致时继续接收被发送来的通信数据。不一致时，就在再次接收到多处理器位为 1 的通信数据前，略过通信数据。

SCI 为支持此功能，在 SCR 中配置了 MPIE 位。如果将 MPIE 置为 1，在接收到多处理器位为 1 的数据前，禁止从 RSR 向 RDR 传送接收数据、禁止检测接收错误和设定 SSR 的 RDRF、FER、ORER 各状态标志。接收到多处理器位为 1 的接收字符后，SSR 的 MPB 被置为 1，同时 MPIE 被自动清零，并返回普通的接收状态。此时如果 SCR 的 RIE 被置 1，就发生 RXI 中断。

指定了多处理器格式时，再指定奇偶校验位是无效的。其它位的设置都和普通异步模式相同。进行多处理器通信时的时钟设置也和普通异步模式的时钟相同。

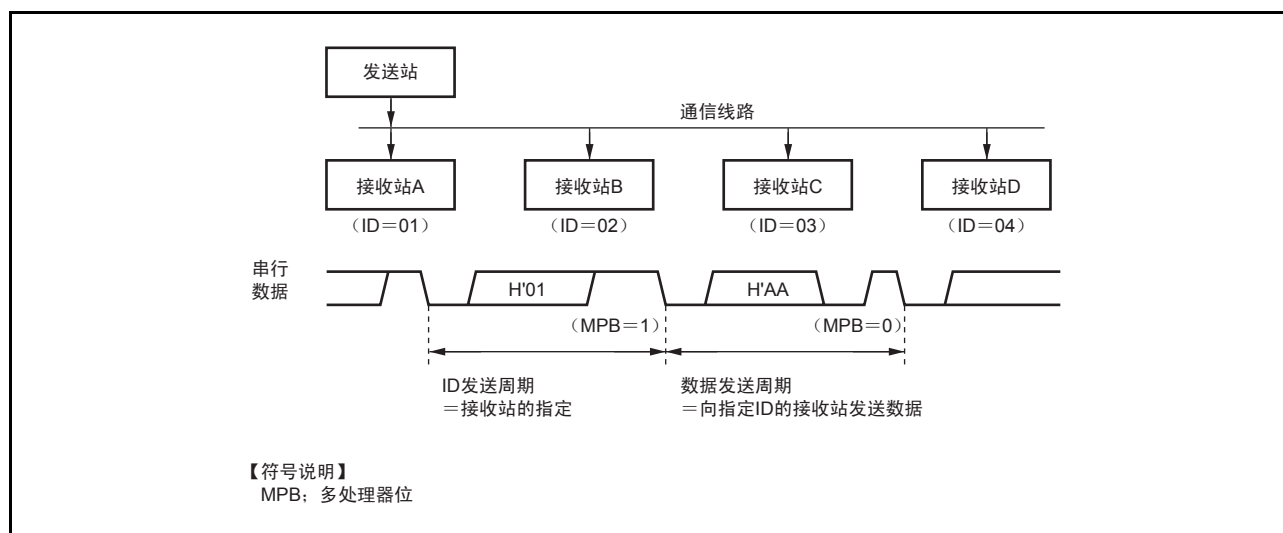


图 12.13 使用了多处理器格式的通信样例（向接收站 A 发送数据 H'AA）

12.5.1 多处理器模式串行数据的发送

图 12.14 所示的是多处理器模式数据处理的流程图例。在 ID 发送周期将 SSR 的 MPBT 置为 1，再发送数据。在数据发送周期将 SSR 的 MPBT 清零，再发送。其它的 SCI 操作均与异步模式的操作相同。

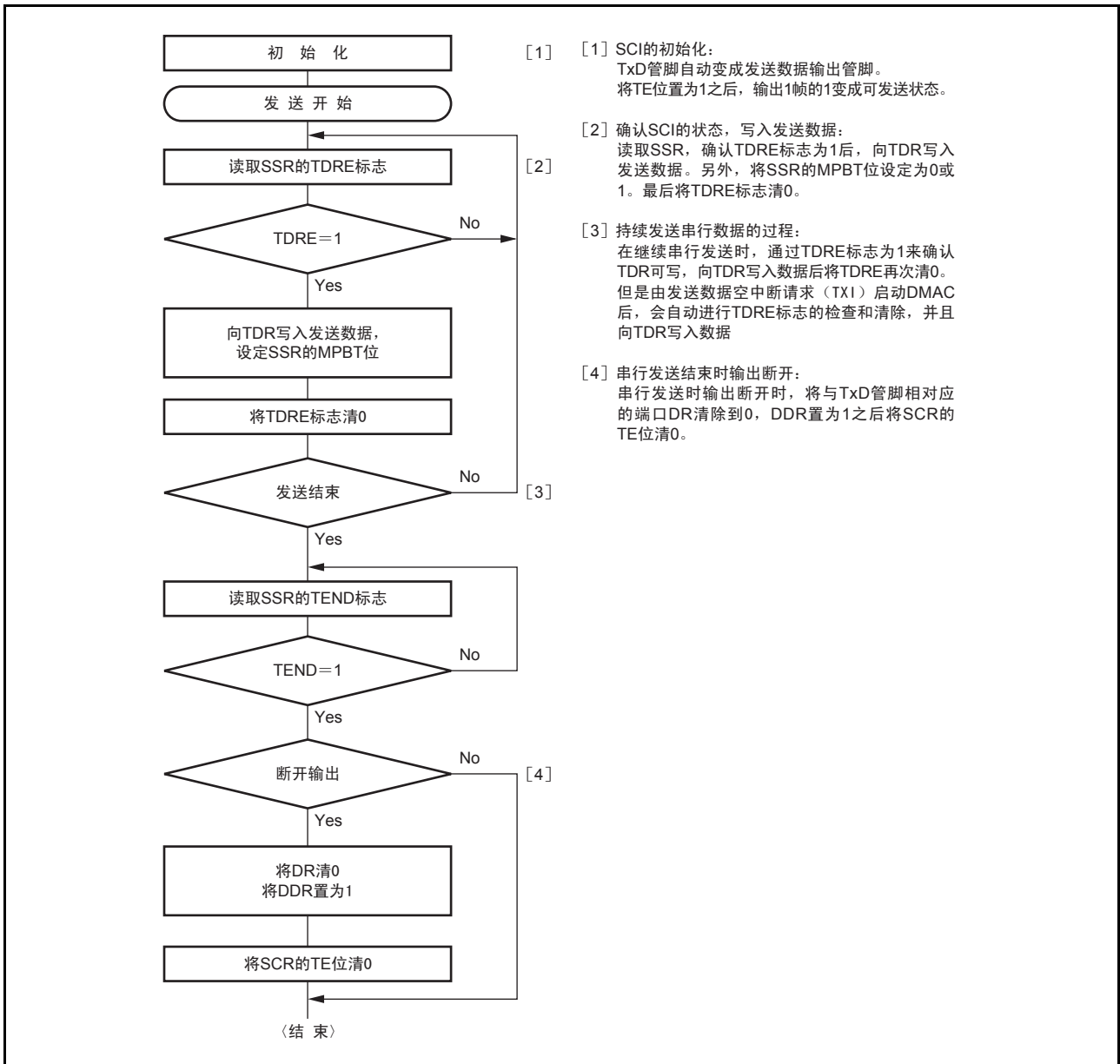


图 12.14 多处理器串行数据发送流程图

12.5.2 多处理器模式串行数据的接收

图 12.16 所示的是多处理器模式串行数据接收流程图例。如果将 SCR 的 MPIE 置 1，在接收到多处理器位为 1 的通信数据前，略过通信数据。接收到多处理器位为 1 的通信数据后，将接收数据传送到 RDR。此时会发生 RXI 中断请求。其它的 SCI 操作均与异步模式的操作相同。图 12.15 所示的 SCI 多处理器接收数据时的操作运行例。

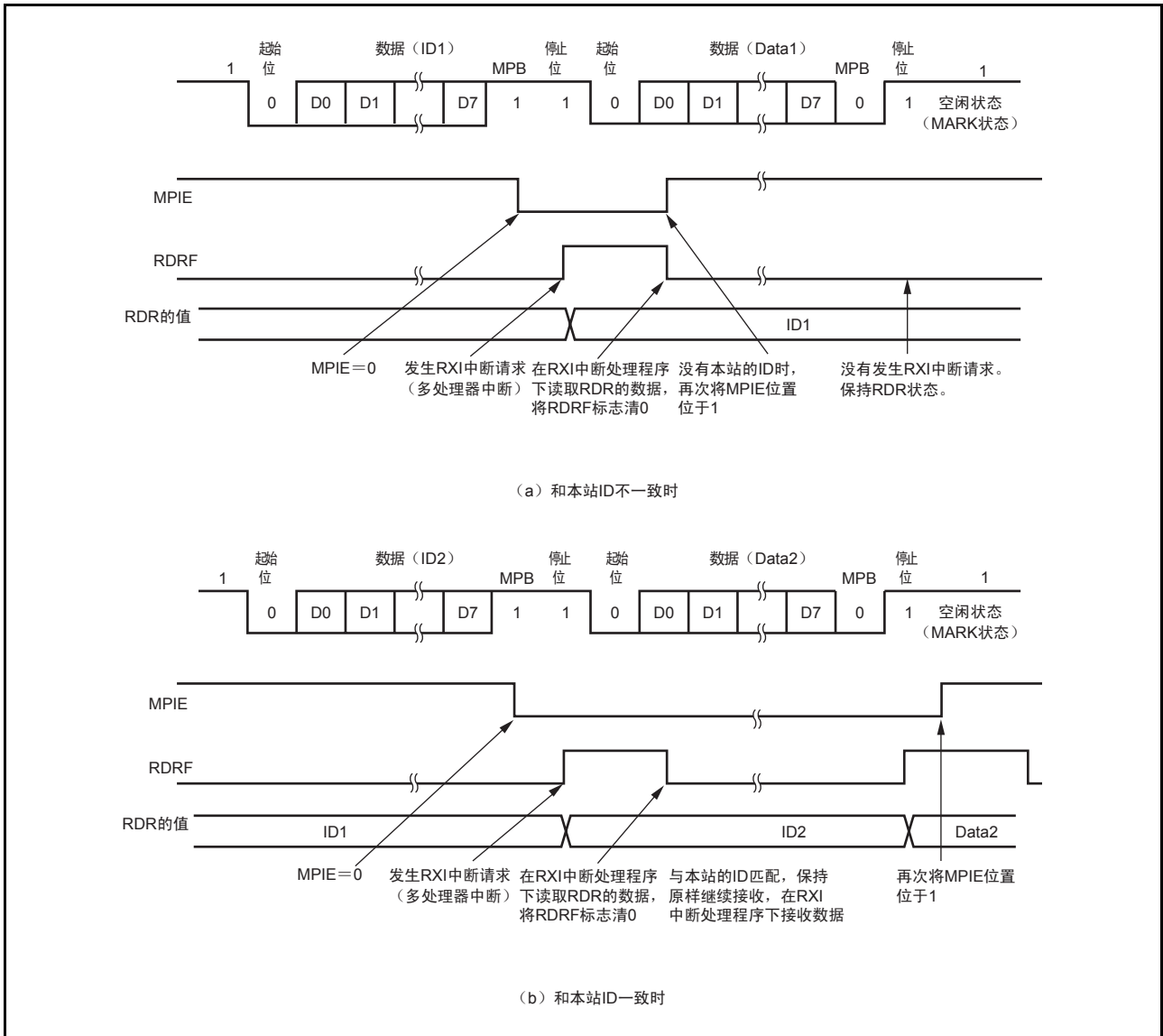


图 12.15 SCI 接收的运行 (8 位数据 / 有多处理器位 / 1 停止位的例)

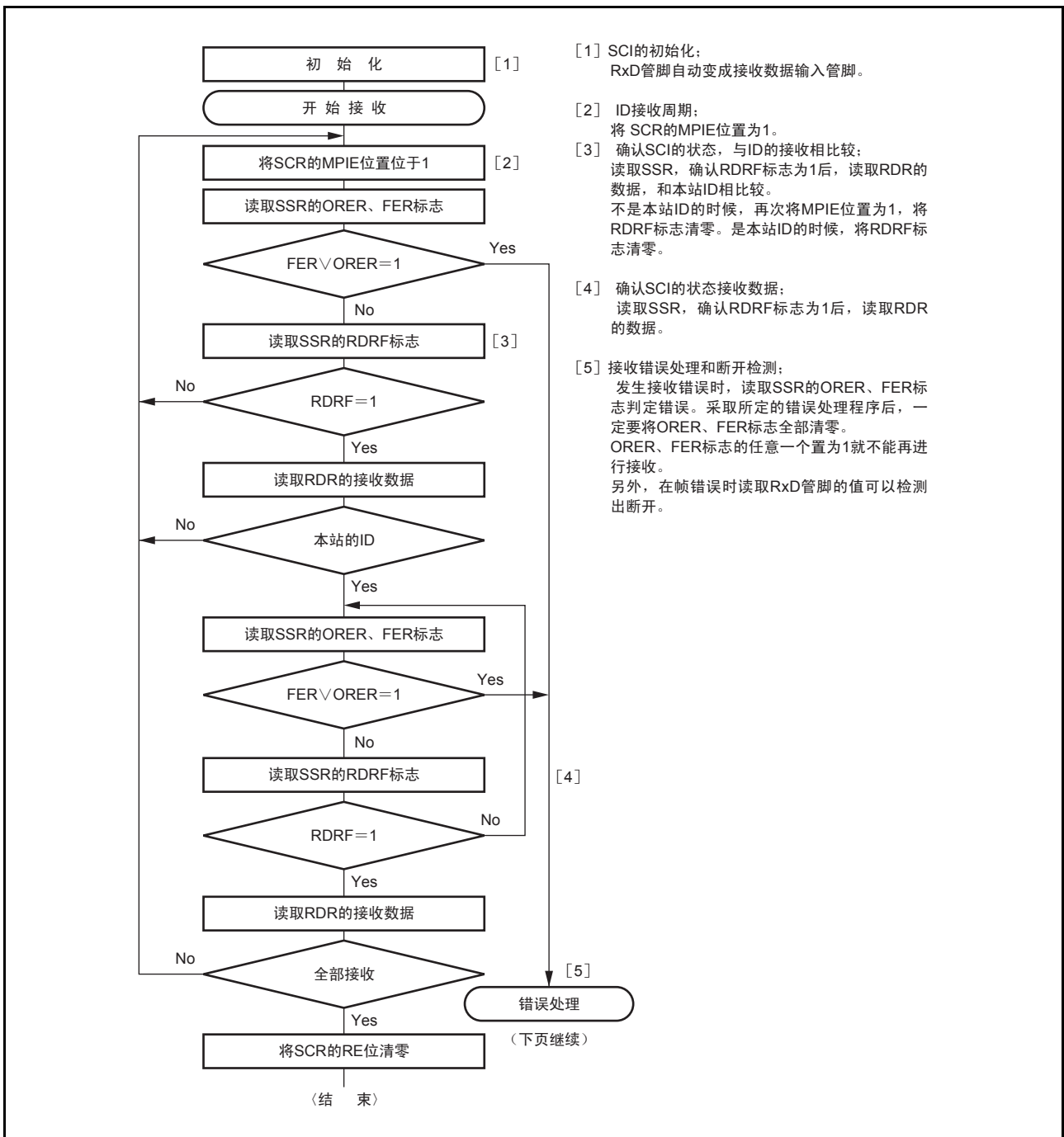


图 12.16 多处理器串行接收流程图例 (1)

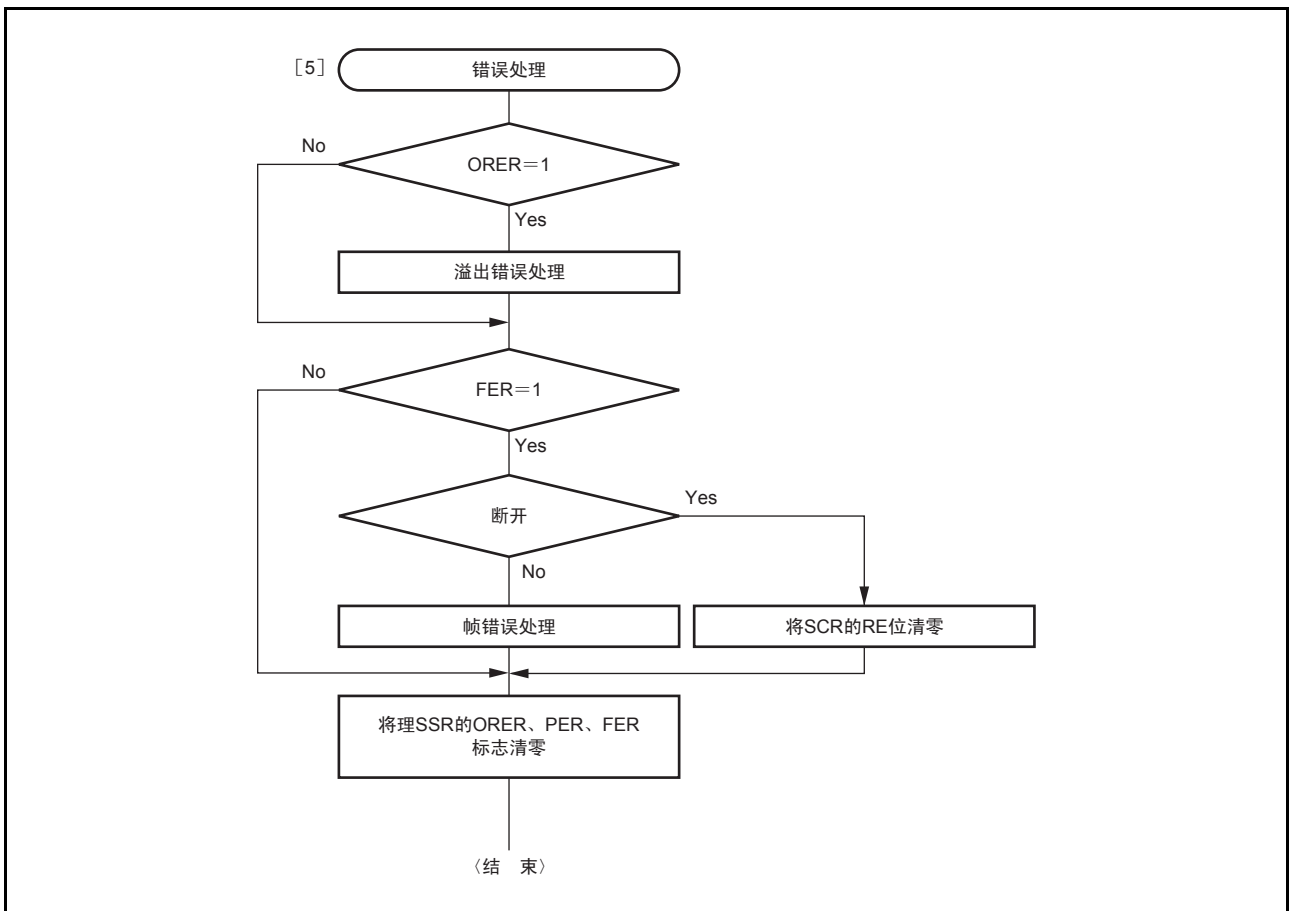


图 12.16 多处理器串行接收流程图例 (2)

12.6 时钟同步模式的运行

时钟同步模式的通信数据格式如图 12.17 所示。在时钟同步模式下，与时钟脉冲同步发送和接收数据。通信数据的 1 个字符由 8 位数据构成。SCI 在数据发送时从同步时钟的一个下降沿开始到下一个下降沿为止发送数据。接收数据时，与同步时钟的上升沿同步接收数据。传输 8 位数据后通信线路保持最终位输出状态。时钟同步模式下，不能附加奇偶校验位和多处理器位。在 SCI 内部发送和接收单元是相互独立的，可共用时钟进行全双工通信。由于发送 / 接收单元采用的是双缓冲结构，发送中写入下一个发送数据，接收中读取前一个接收数据，就可实现连续接收和发送。

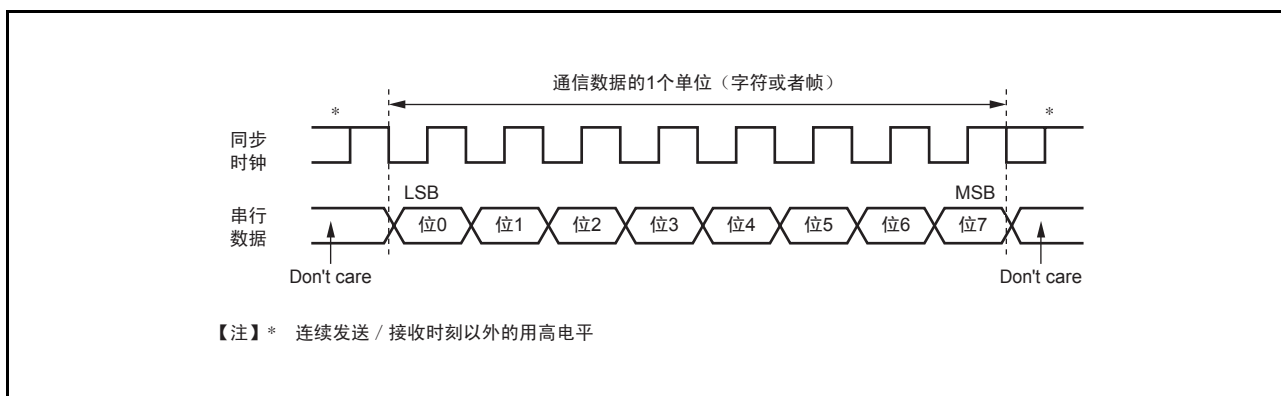


图 12.17 时钟同步模式通信的数据格式 (LSB 优先)

12.6.1 时钟

通过设定 SCR 的 CKE1、CKE0，可选择内部波特率发生器产生的内部时钟或从 SCK 管脚输入的外部同步时钟。当内部时钟运行时，可从 SCK 管脚输出同步时钟。同步时钟发送和接收 1 个字符，将要输出 8 个持续脉冲，不进行发送 / 接收时，时钟被固定为高电平。

12.6.2 SCI 的初始化 (时钟同步模式)

发送 / 接收数据前, 将 SCR 的 TE、RE 位清 0 后, 按照图 12.18 的流程图例对 SCI 进行初始化。务必将 TE 位及 RE 位清零后, 再变更运行模式、通信格式等。将 TE 清零后, TDRE 被置为 1, 但是, 请注意, 即使 RE 也被清零, RDRF、PER、FER、ORER 的各标志及 RDR 也不会被初始化。

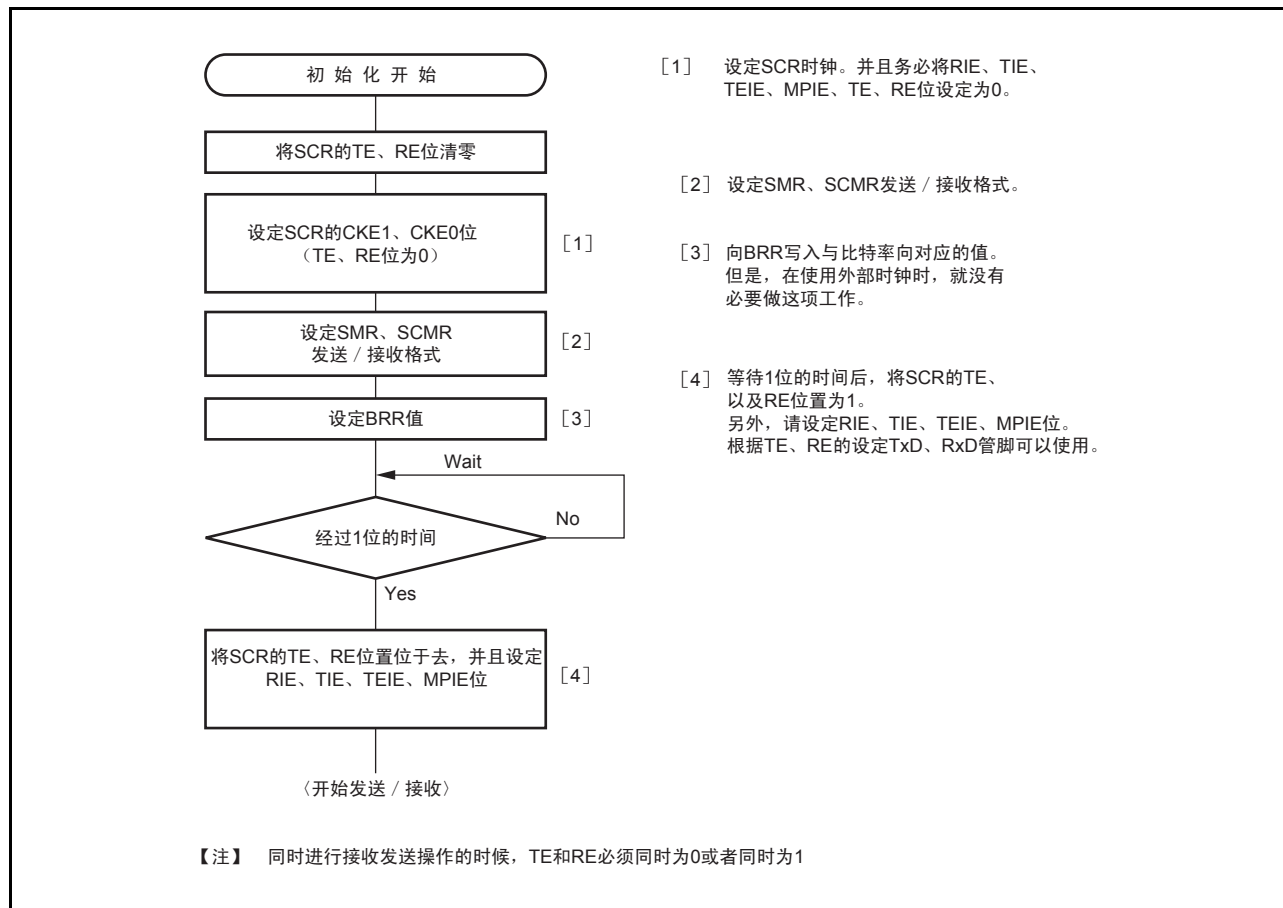


图 12.18 SCI 的初始化流程图例

12.6.3 串行数据的发送 (时钟同步模式)

图 12.19 所示的是时钟同步模式下发送数据的运行例。发送数据时 SCI 运行步骤如下：

1. SCI 监视 SSR 的 TDRE，如果被清零，SCI 则判定已经在 TDR 中写入数据，将数据从 TDR 传送到 TSR。
2. 数据从 TDR 传送到 TSR 后，将 TDRE 置为 1，开始发送数据。此时如果 SCR 的 TIE 置为 1，会发生 TXI 中断请求。在 TXI 中断程序中，发送完上一个传送的数据前，给 TDR 写入下一个数据，就可连续发送。
3. 如果设定输出时钟模式时，从 TXD 管脚输出与输出时钟同步的 8 位数据；如果设定为外部时钟时，与输入时钟同步，从 TXD 管脚输出 8 位数据。
4. 发送出最终位的同时检查 TDRE。
5. 如果 TDRE 为 0，下一个发送数据从 TDR 传送到 TSR，开始发送下一帧。
6. 如果 TDRE 为 1，将 SSR 的 TEND 置 1，保持最终位输出状态。此时 SCR 的 TEIE 置为 1 后，会发生一个 TEI 中断请求，SCK 管脚固定为高电平。

图 12.20 所示的是串行数据发送的流程图例。在接收错误标志 (ORER、FER、PER) 置为 1 的状态，即使将 TDRE 清 0，也不开始发送数据。发送开始前，必须事先将接收错误标志清零。注意：仅清除 RE 位，接收错误标志是不会被清除的。

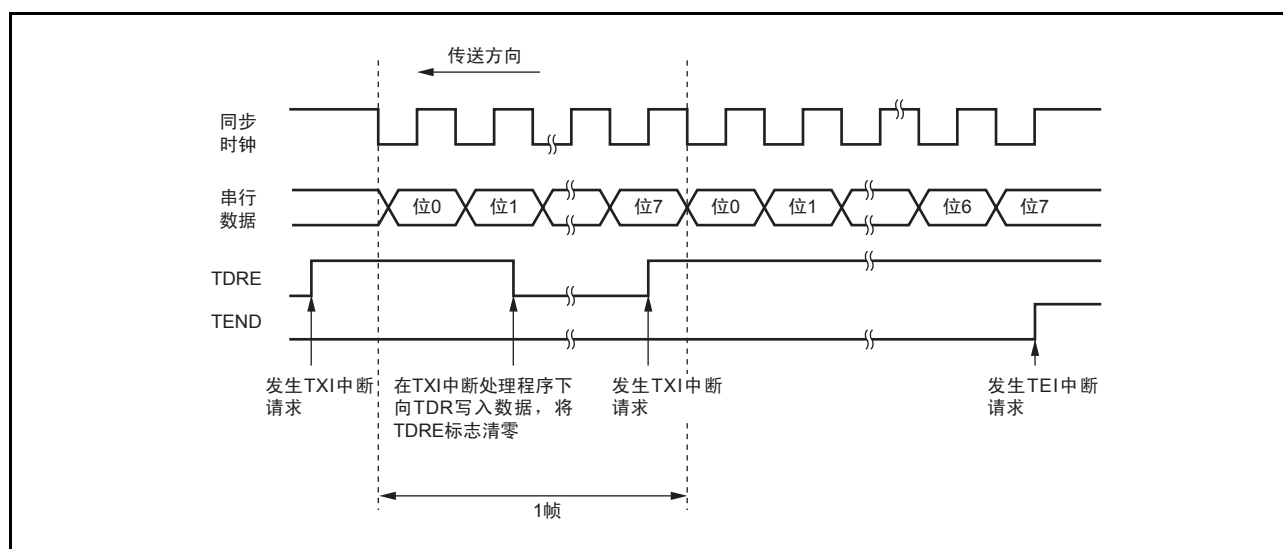


图 12.19 时钟同步模式发送数据的运行例

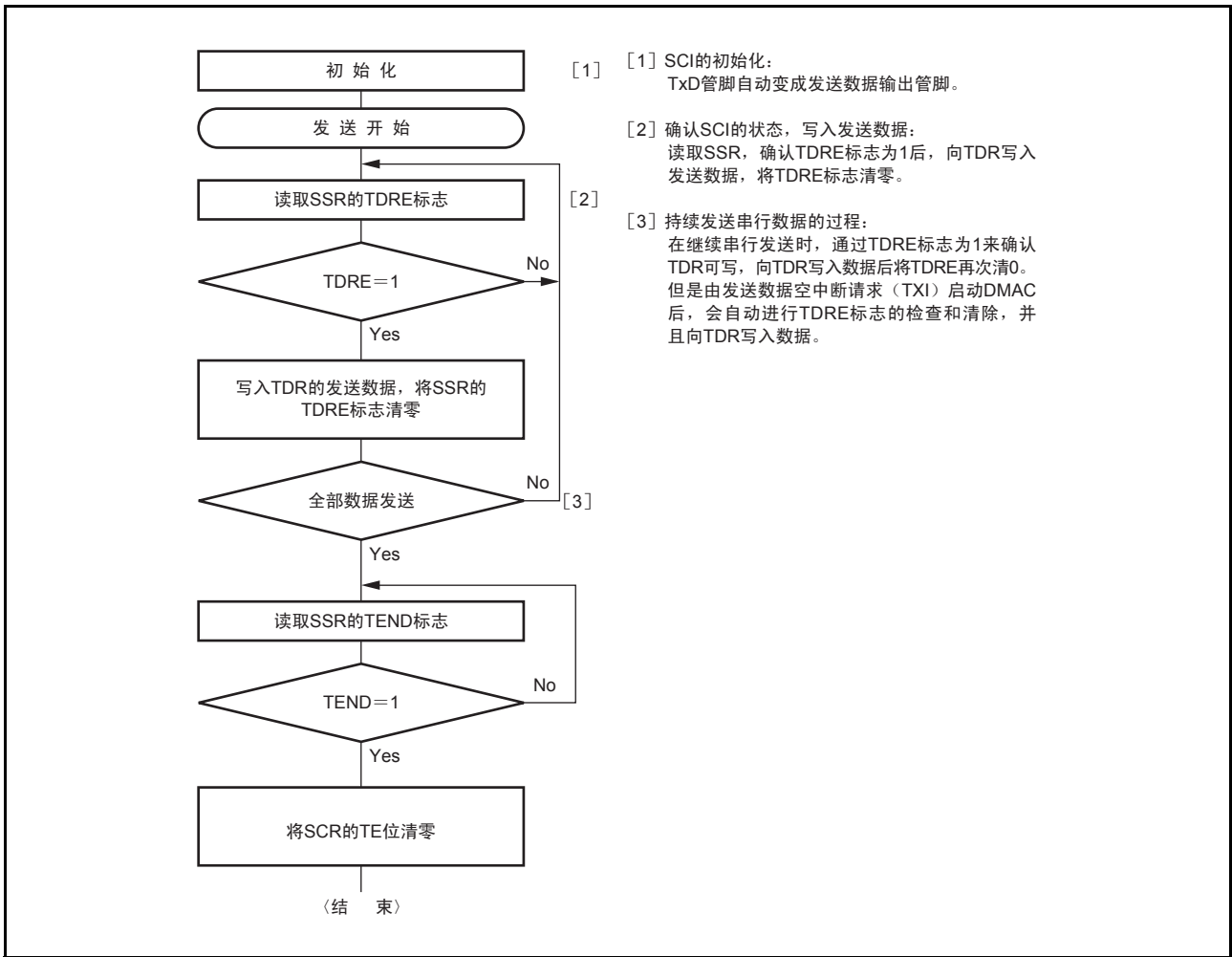


图 12.20 行数据发送的流程图例

12.6.4 接收串行数据 (时钟同步模式)

图 12.21 所示的是时钟同步模式下接收数据的运行例。接收数据时 SCI 运行步骤如下：

1. SCI 执行内部初始化与同步时钟的输入或输出同步后，开始接收数据，并将接收数据放入 RSR。
2. 发生溢出错误时（在 SSR 的 RDRF 置为 1 的状态，接收到下一个数据时），要把 SSR 的 ORER 置 1。这时如果 SCR 的 RIE 置为 1，会产生 ERI 中断请求。接收数据将不会传送到 RDR，并保持 RDRF 置为 1 的状态。
3. 如果接收时，SSR 的 RDRF 位置 1，将接收数据传送到 RDR。此时如果 SCR 的 RIE 置为 1，会发生 RXI 中断请求。在 RXI 中断处理程序中，接收下一个数据完成前，读取传送到 RDR 的接收数据，就可连续接收。

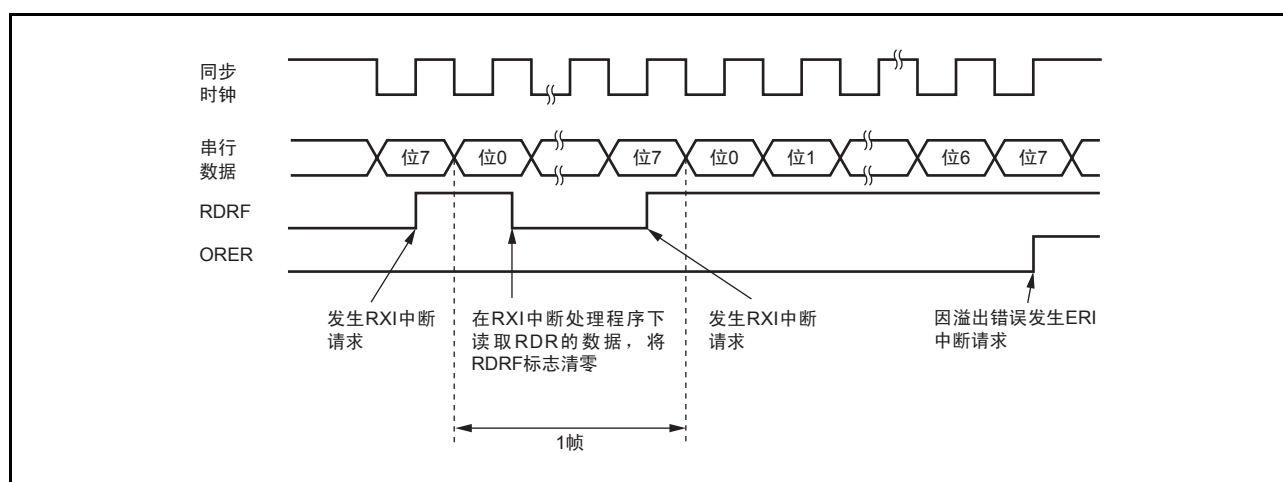


图 12.21 SCI 接收时的运行例

在接收错误标志被置 1 的状态，不能进行以后的接收运行。因此，在继续接收前请务必将 ORER、FER、PER 及 RDRF 清零。图 12.22 所示的是接收数据的流程图例。

选择内部时钟，接收运行时，在发生溢出错误或者 RE 位清 0 后，输出同步时钟。想要进行 1 帧单位的接收时，请同时进行发送仅 1 帧的虚数据。

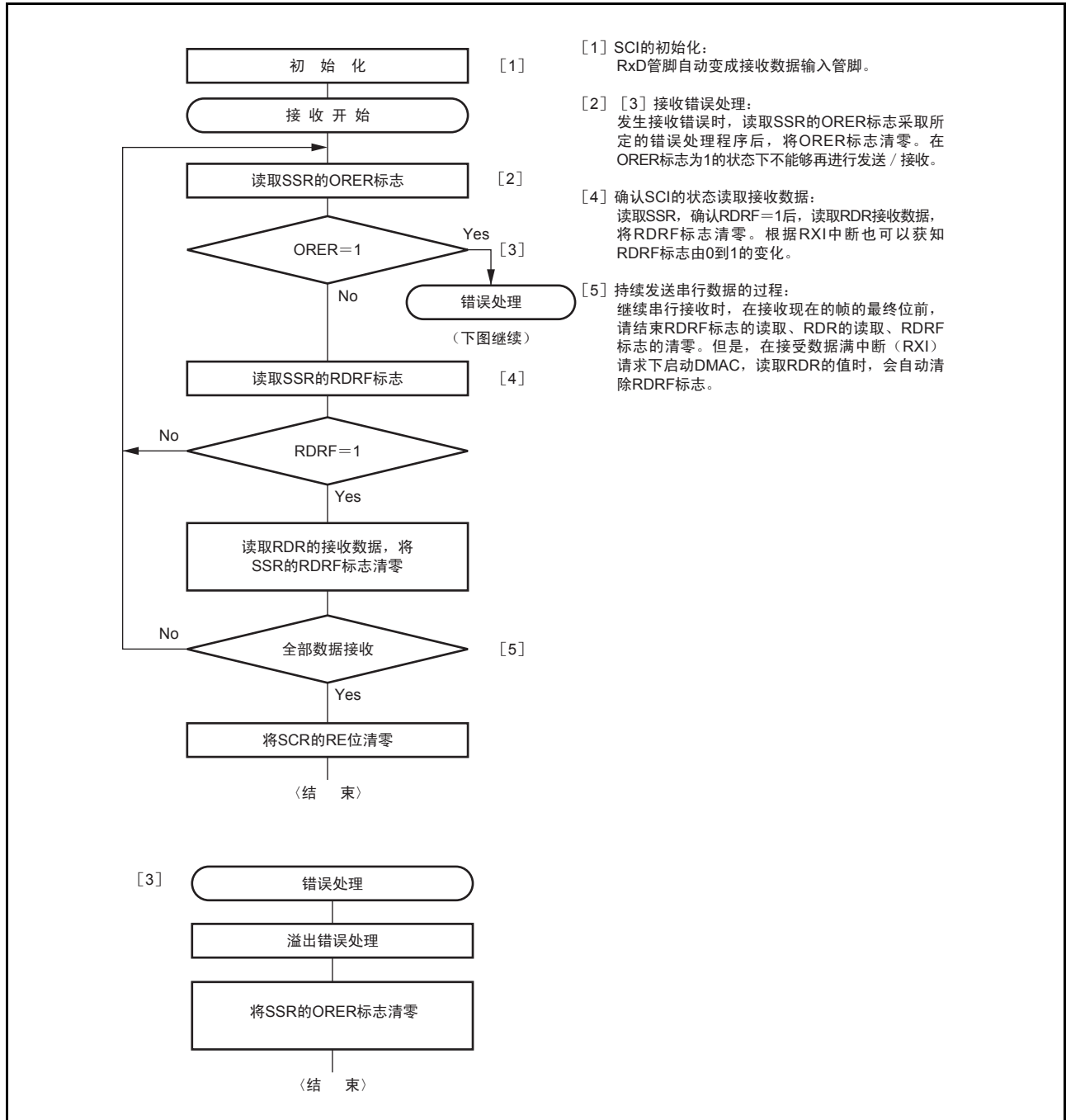


图 12.22 接收串行数据的流程图例

12.6.5 发送 / 接收串行数据同时运行 (时钟同步模式)

图 12.23 所示的是发送 / 接收同步运行的流程图例。在 SCI 初始化后, 按照以下顺序, 同时发送 / 接收数据。从发送切换到同时发送 / 接收时, 确认 SCI 为发送完成状态, TDRE 及 TEND 被置为 1, TE 清零后, 将 TE 及 RE 根据单一指令同时置为 1。从接收模式切换到同时发送 / 接收模式时, 确认 SCI 为接收完成状态, 将 RE 清零后, 确认 RDRF 及接收错误标志 (ORER、FER、PER) 都被清零后, 根据单一指令将 TE 及 RE 同时置为 1。

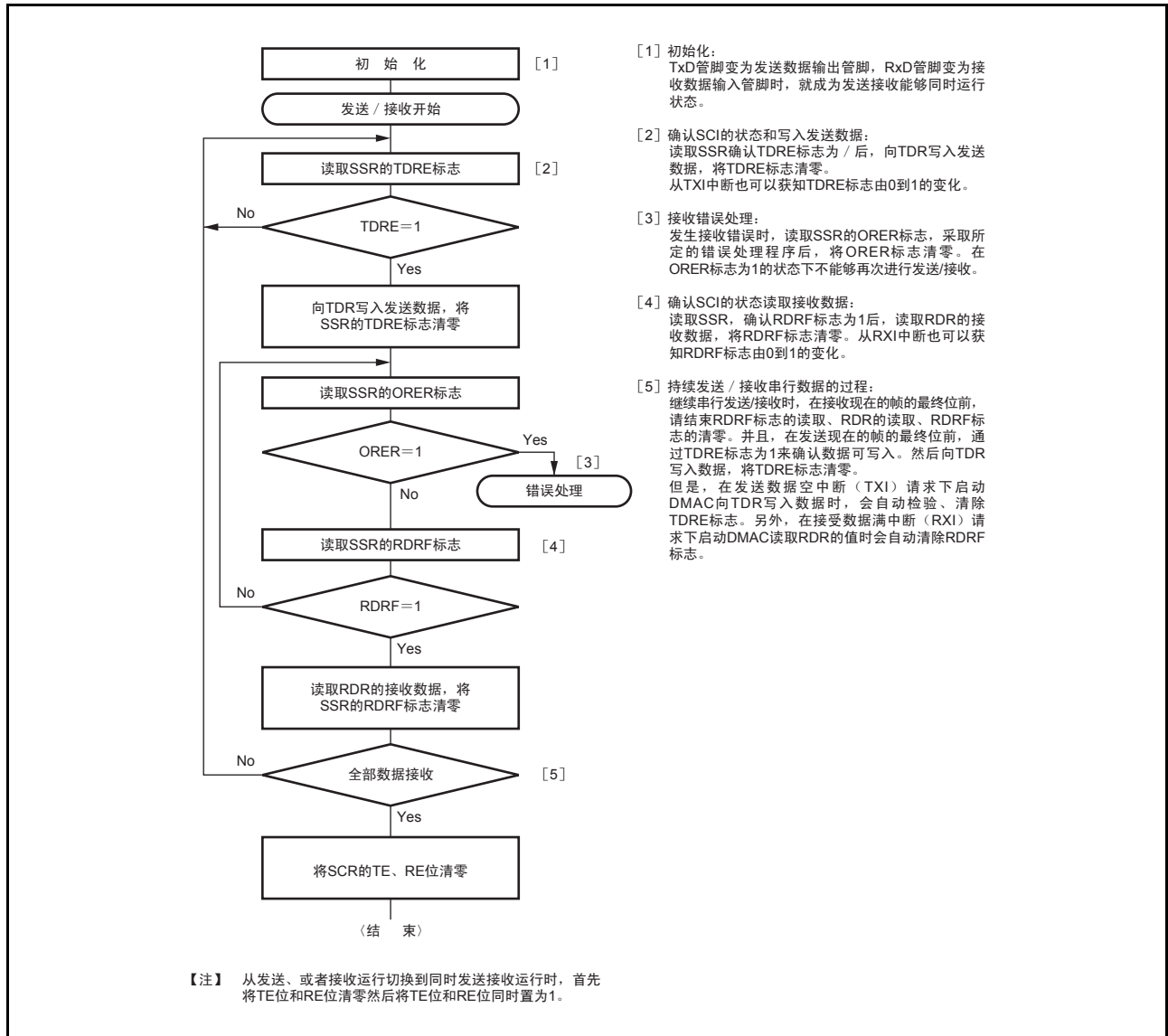


图 12.23 串行发送 / 接收同时运行的流程图例

12.7 智能卡接口操作说明

SCI 支持 ISO/IEC 7816-3 (Identification Card) 标准的 IC 卡 (智能卡) 接口, 并将其作为串行通信接口的扩展功能。可通过设置寄存器的方法进行常规的串行通信接口模式和智能卡接口模式的切换。

12.7.1 管脚连接图例

图 12.24 是与智能卡连接的图例。在 IC 卡通信过程中, 用一根数据传输线接收 / 发送数据, 连接时将 TxD 管脚和 RxD 管脚短接, 在数据传输线上用电阻向电源 VCC 上拉。如果在没有连接 IC 卡的状态, 设定 RE = TE = 1, 就变为封闭的发送 / 接收, 可进行自诊断。要将 SCI 产生的时钟提供给 IC 卡时, 请把 SCK 管脚的输出连接到 IC 卡的 CLK 管脚。IC 卡中, 使用内部时钟时, 不用连接。IC 卡的复位信号可与本 LSI 的输出端口连接。除此之外, 作为管脚需要连接电源和地。

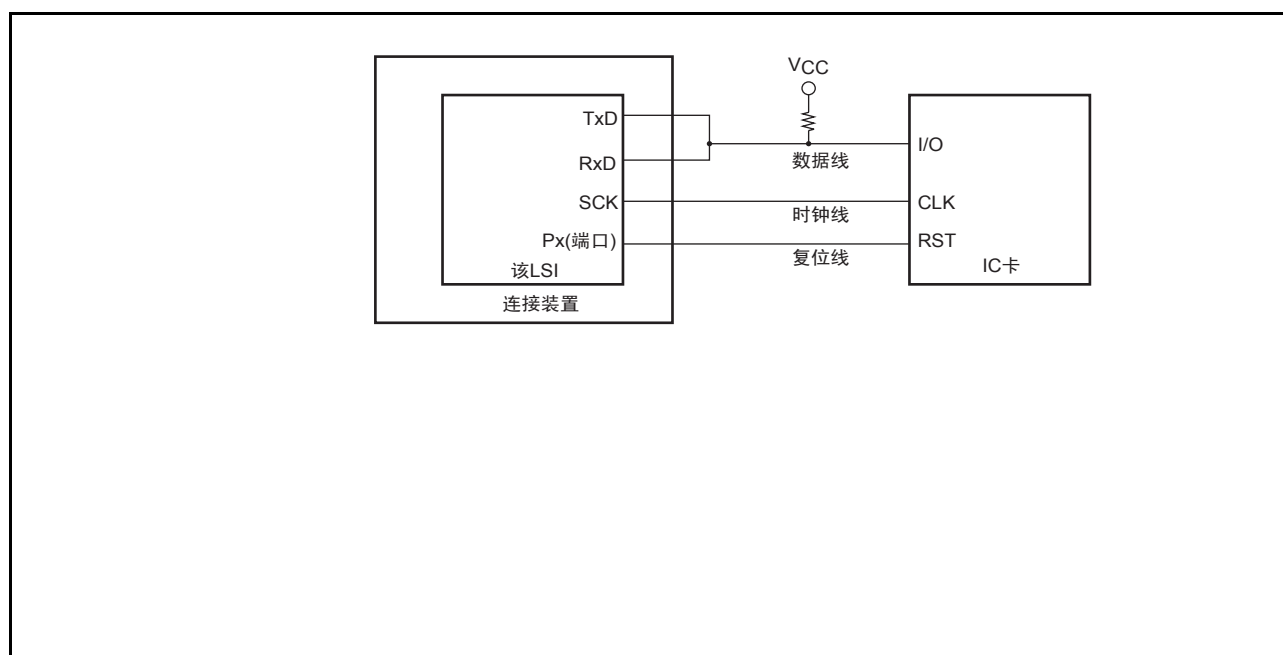


图 12.24 智能卡接口管脚连接概要

12.7.2 数据格式 (除数据块传送模式之外)

图 12.25 所示的是智能卡接口模式下的发送 / 接收格式。

- 异步模式下，1 帧数据由 8 位数据和奇偶校验位构成。
- 发送时，从奇偶校验位结束到下一帧开始，要留 2etu (Elementary Time Unit: 传送 1 位) 以上的保护时间。
- 接收时，检测出奇偶校验错误，从起始位经过 10.5etu 后，错误信号的低电平将在 1etu 时间输出。
- 发送时，如果采样到错误信号，经过 2etu 以上后，会自动再次发送相同数据。

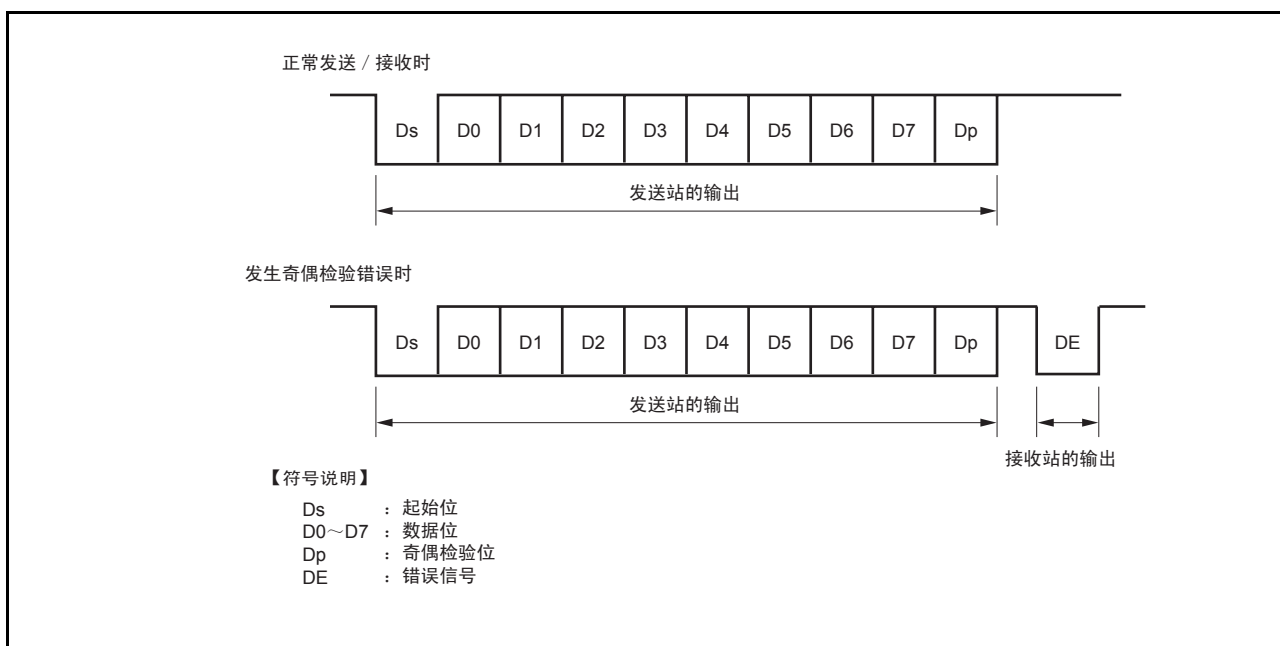


图 12.25 通常智能卡接口的数据格式

如下与直接转换及反码转换 2 种类型的 IC 卡进行发送 / 接收。

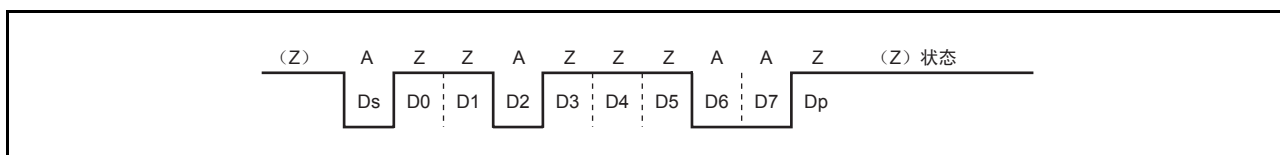


图 12.26 直接转换 (SDIR = SINV = $\overline{O/E} = 0$)

如下图进行字符直接转换例，直接转换类型逻辑 1 与状态 Z 对应，逻辑 0 与状态 A 对应，以 LSB 优先格式发送 / 接收。以上开始字符中，数据为 H'3B。在直接转换类型中，请将 SCMR 的 SDIR 和 SINV 位同时置为 0。根据智能卡规范，要变为偶校验，请将 SMR 的 $O\bar{E}$ 位置为 0。

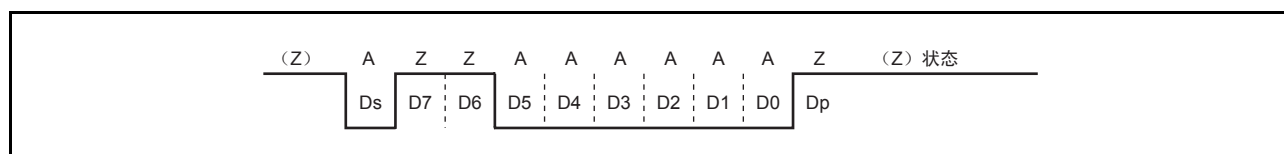


图 12.27 反码转换 ($SDIR = SINV = O\bar{E} = 1$)

反码转换类型，逻辑 1 与状态 A 对应，逻辑 0 与状态 Z 对应，以 MSB 优先格式发送 / 接收。以上开始字符中，数据为 H'3F。反码转换类型中，请将 SCMR 的 SDIR 位、SINV 位均置为 1。根据智能卡规范，奇偶校验位若是偶校验则为逻辑 0，与状态 Z 对应。在本 LSI 中，SINV 位只能反转数据位 D7 ~ D0，因此，发送 / 接收都要将 SMR 的 $O\bar{E}$ 位置为 1，才可反转奇偶位。

12.7.3 时钟

发送接收时钟只能使用内部波特率生成器生成的内部时钟。此时，如果用 $CKE0 = 1$ 选择时钟输出，会从 SCK 管脚输出频率为比特率的 S^* 倍的时钟。

【注】 * 符号 S 表示的是 12.3.11 比特率寄存器 (BRR) 中 S 的值。

12.7.4 块传送模式

数据块传送模式与普通的智能卡接口相比，有以下不同点：

- 接收时，虽然进行奇偶校验检查，即使检查到错误也不会输出错误信号。由于 SSR 的 PER 被置 1，因此接收下一帧的奇偶校验位之前请将其清除。
- 发送时，从奇偶校验位结束到下一帧开始的保护时间最少为 1etu 以上。
- 发送时，不执行再次发送，所以从发送开始后 11.5etu 的时间，将 TEND 标志置位。
- 与通常的智能卡接口相同，ERS 标记表示错误信号的状态，但由于不发送和接收错误信号，所以通常为 0。

12.7.5 接收数据采样时序接收容限

在智能卡接口模式下，可使用的发送接收时钟只有 On-chip 波特率生成器生成的内部时钟。通过设定 BCP1、BCP0，SCI 以频率为波特率的 32 倍、64 倍、372 倍、256 倍（普通异步模式中固定为 16 倍）的基本时钟运行。接收时，用基本时钟采样起始位的下降沿，并使内部同步。如图 12.28 所示，分别在基本时钟脉冲的第 16、32、186、128 个上升沿进行采样，在各位的中间放入数据。这时的接收容限可用下面的公式表示。

$$M = \left| -\frac{1}{2N} - F(L - 0.5) - (F+1) \frac{|D - 0.5|}{N} + 0.5 \right| \times 100\%$$

M: 接收容限 (%)

N: 时钟与波特率的比值 (N = 32、64、372、256)

D: 时钟占空比 (D = 0 ~ 1.0)

L: 帧长度 (L = 10)

F: 时钟频率偏差的绝对值

以上的公式中，如果 F = 0、D = 0.5、N = 372，接收容限变为以下所示。

$$\begin{aligned} M &= (0.5 - 1/2 \times 372) \times 100\% \\ &= 49.866\% \end{aligned}$$

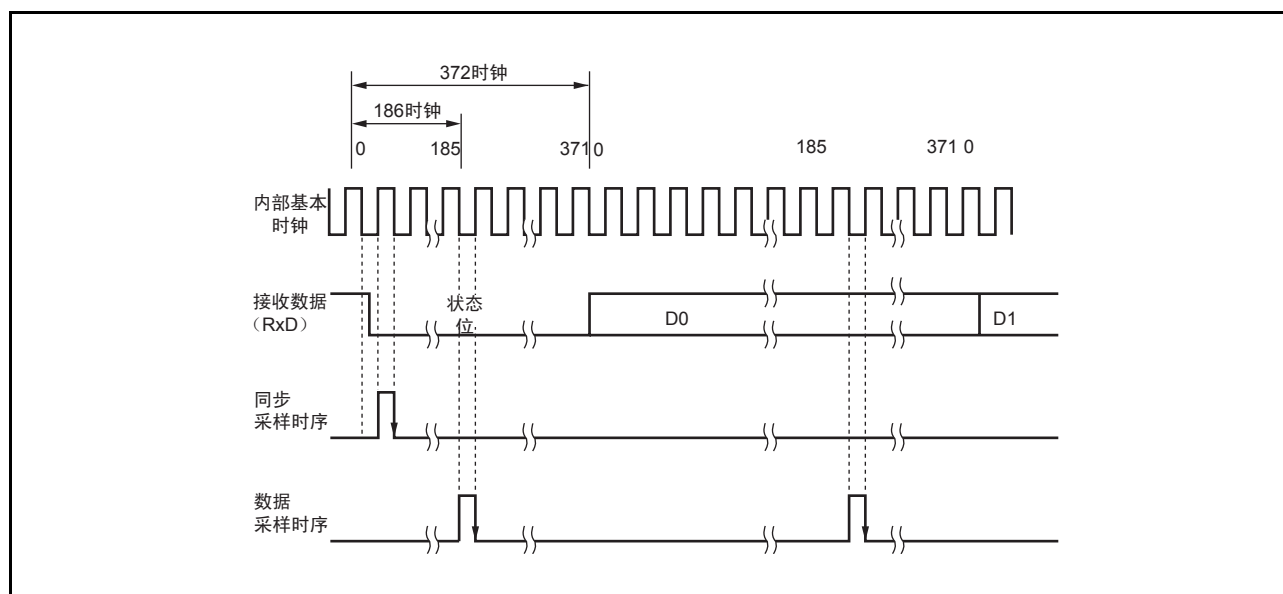


图 12.28 智能卡接口模式下的接收数据采样时序 (使用 372 倍的时钟时)

12.7.6 初始设定

发送 / 接收数据前, 按以下顺序初始化 SCI。无论从发送模式转换为接收模式, 还是从接收模式转换为发送模式, 都需要初始化。

1. 将 SCR 的 TE、RE 位清 0。
2. 将 SSR 的错误标志 ERS、PER、ORER 清 0。
3. 设置 SMR 的 GM、BLK、O/E、BCP1、BCP0、CKS1、CKS0 位。此时请将 PE 位置为 1。
4. 设置 SCMR 的 SMIF、SDIR、SINV 位。
将 SMIF 位置为 1 后, TxD 管脚及 RxD 管脚同时从端口切换到 SCI 管脚, 从而置为高阻抗状态。
5. 在 BRR 中设置与波特率相对应的值。
6. 设定 SCR 的 CKE1、CKE0 位。将 TIE、RIE、TE、RE、MPIE、TEIE 位均置为 0。
如果将 CKE0 位置为 1, 将从 SCK 管脚输出时钟。
7. 至少等候 1 位的间隔时间, 然后设定 SCR 的 TIE、RIE、TE、RE 位。除过自诊断外, 请不要同时设置 TE 位和 RE 位。

从接收模式向发送模式切换时, 确认 SCI 接收已经完成, 从初始化开始, 设定 RE = 0、TE = 1。可根据 RDRF、PER 或 ORER 标志确认 SCI 接收是否完成。从发送模式向接收模式切换时, 确认 SCI 发送完成, 从初始化开始, 设定 TE = 0、RE = 1。可根据 TEND 标志确认 SCI 发送是否完成。

12.7.7 发送串行数据 (除数据块传送模式之外)

智能卡模式下的数据发送具有错误信号的采样和再发送处理, 因此与普通串行通信接口的运行不同 (除数据块传送模式之外)。发送数据的再传送运行如图 12.29 所示。

1. 发送 1 帧数据完成后, 如果检测到接收方的错误信号, SSR 的 ERS 位被置为 1。这时如果 SCR 的 RIE 位被置为 1, 会产生 ERI 中断请求。在采样下一个奇偶校验位前, 请将 ERS 清零。
2. 在接收到错误信号的帧中, 表明 SSR 的 TEND 未被置位, 会再次将数据从 TDR 传送到 TSR, 数据会自动被再次发送。
3. 如果没有从接收方返回错误信号时, 表明 SSR 的 ERS 位未被置位, 确定包括再次传送数据的一个帧数据已经发送完成, 则 SSR 的 TEND 会被置 1, 这时如果 SCR 的 TIE 被置位, 就产生 TXI 中断请求, 此时将发送数据写到 TDR, 下一个数据就会被发送。

发送处理流程例如图 12.31 所示。这些一连串的处理, 通过 TXI 中断源启动 DMAC, 就可以自动执行。在发送过程中, SSR 的 TEND 标志被置为 1, 同时 TDRE 标志也被置位, 如果事先设定 SCR 的 TIE = 1, 就会发生 TXI 中断请求。如果事先设定 TXI 请求作为 DMAC 启动源, 就会通过 TXI 请求, 启动 DMAC, 执行发送数据的传送。TDRE 及 TEND 标志通过 DMAC 传送数据时, 会被自动清 0。当发生错误时, SCI 会自动再次发送相同数据, 在此期间 TEND 保持为 0, DMAC 不能启动。因此, 对于发生错误时的再次发送, SCI 和 DMAC 会自动发送指定的字节数。但是发生错误时, ERS 标志不会被自动清除, 所以要事先将 RIE 位置为 1, 发生错误时, ERI 发生中断请求, 同时请清除 ERS 标志。

而使用 DMAC 发送接收信息时, 务必事先设定 DMAC, 允许状态后, 请进行 SCI 设定。DMAC 的设定方法请参考 7. DMA 控制器 (DMAC)。

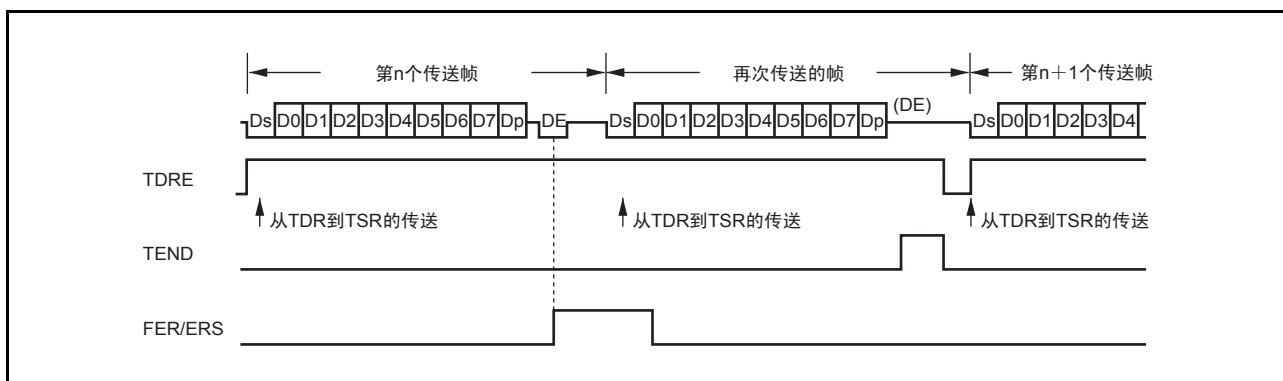


图 12.29 SCI 发送模式时的再传送运行

而根据 SMR 的 GM 位设定的不同，TEND 标志的设置时序也不同。图 12.30 所示的是 TEND 标志发生时序。

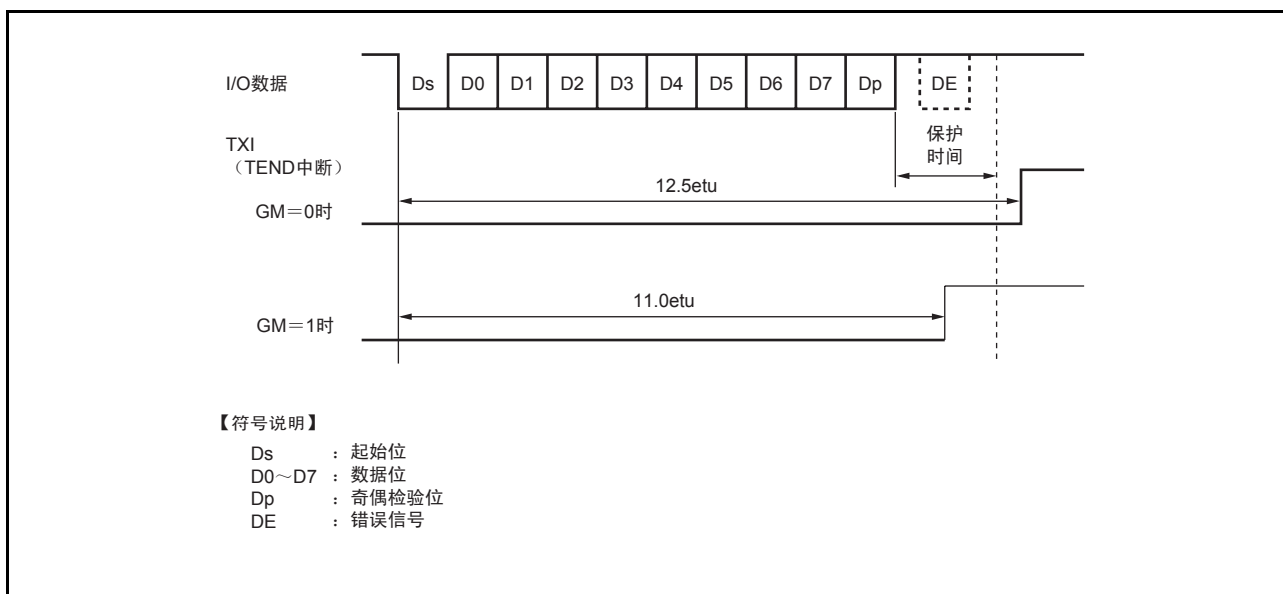


图 12.30 发送运行时的 TEND 标志发生时序

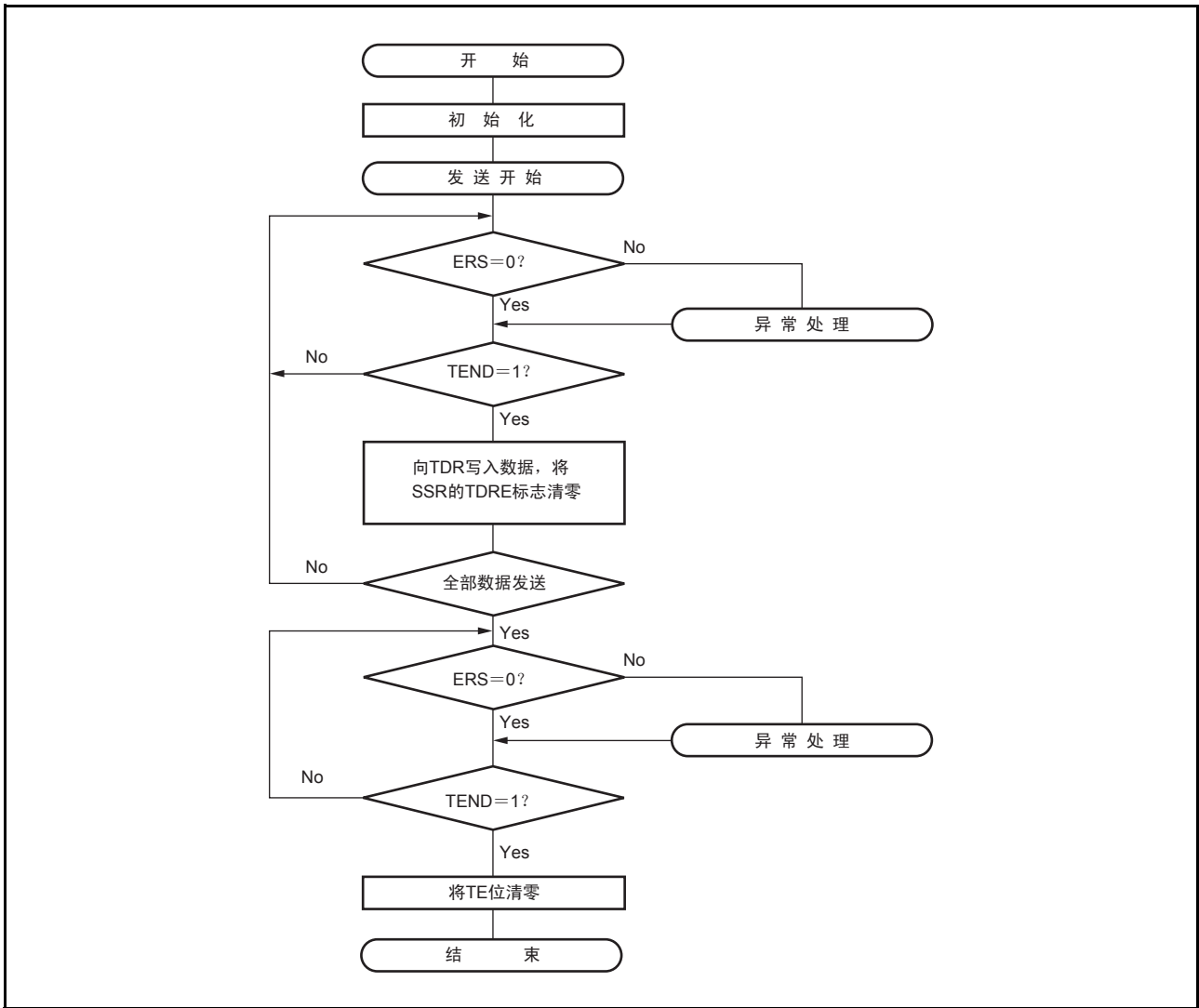


图 12.31 发送处理流程例

12.7.8 串行数据的接收 (除数据块传送模式之外)

智能卡接口模式下的接收数据处理顺序与普通串行通信接口模式的相同。SCI 的接收模式的再次传送运行如图 12.32 所示。

1. 接收数据时, 如果检测出奇偶校验, SSR 的 PER 位自动置为 1。这时, 如果 SCR 的 RIE 被置位, 就会发生 ERI 中断请求。在检测到下一个奇偶校验位前, 请保持 PER 位清零状态。
2. 检测出奇偶校验错误时, SSR 的 RDRF 位不被置位。
3. 没有检测出奇偶校验错误时, SSR 的 PER 位未被置为 1。判断其正常完成接收后, SSR 的 RDRF 自动置 1。这时, 如果 SCR 的 RIE 位被置位, 则会发生 RXI 中断请求。

接收数据流程例如图 12.33 所示。这些一连串的处理, 通过 RXI 中断源启动 DMAC, 就可以自动执行。在接收过程中, 当预先将 RIE 位置 1, RDRF 标志被置 1, 就会发生 RXI 中断请求。如果事先设定 RXI 请求作为 DMAC 的启动源, 就会通过 RXI 请求启动 DMAC, 来发送数据, 如果通过 DMAC 传送数据, RDRF 标志会被自动清除。而在接收时发生错误, 则 ORER 或 PER 标志的任一个被置为 1, 就会发生接收 / 发送错误中断 (ERI) 请求, 所以请及时将错误标志清除。发生错误时, 不启动 DMAC, 接收数据会被略过, 所以只传送 DMAC 中设定的字节数。

即使在接收时发生奇偶校验错误, PER 被置 1, 接收的数据被传送 RDR, 因此可以读取这些数据。

【注】 数据块传送模式时请参考 12.4 异步模式的运行。

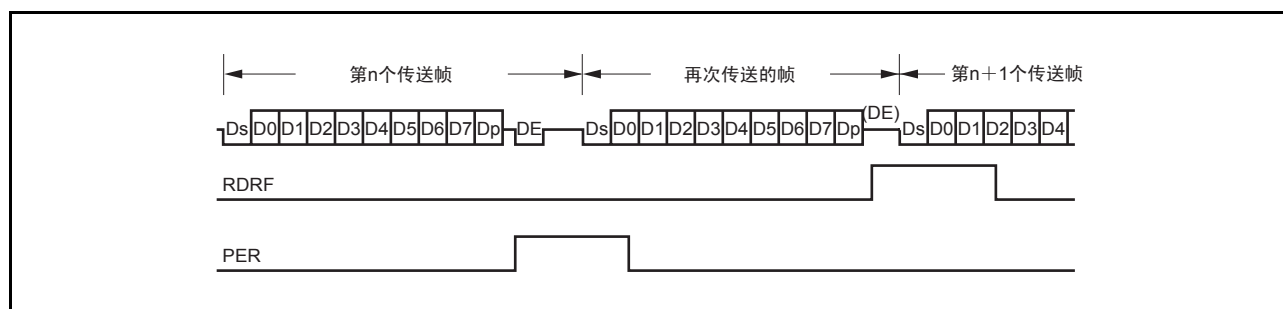


图 12.32 SCI 接收模式时的再次传送运行

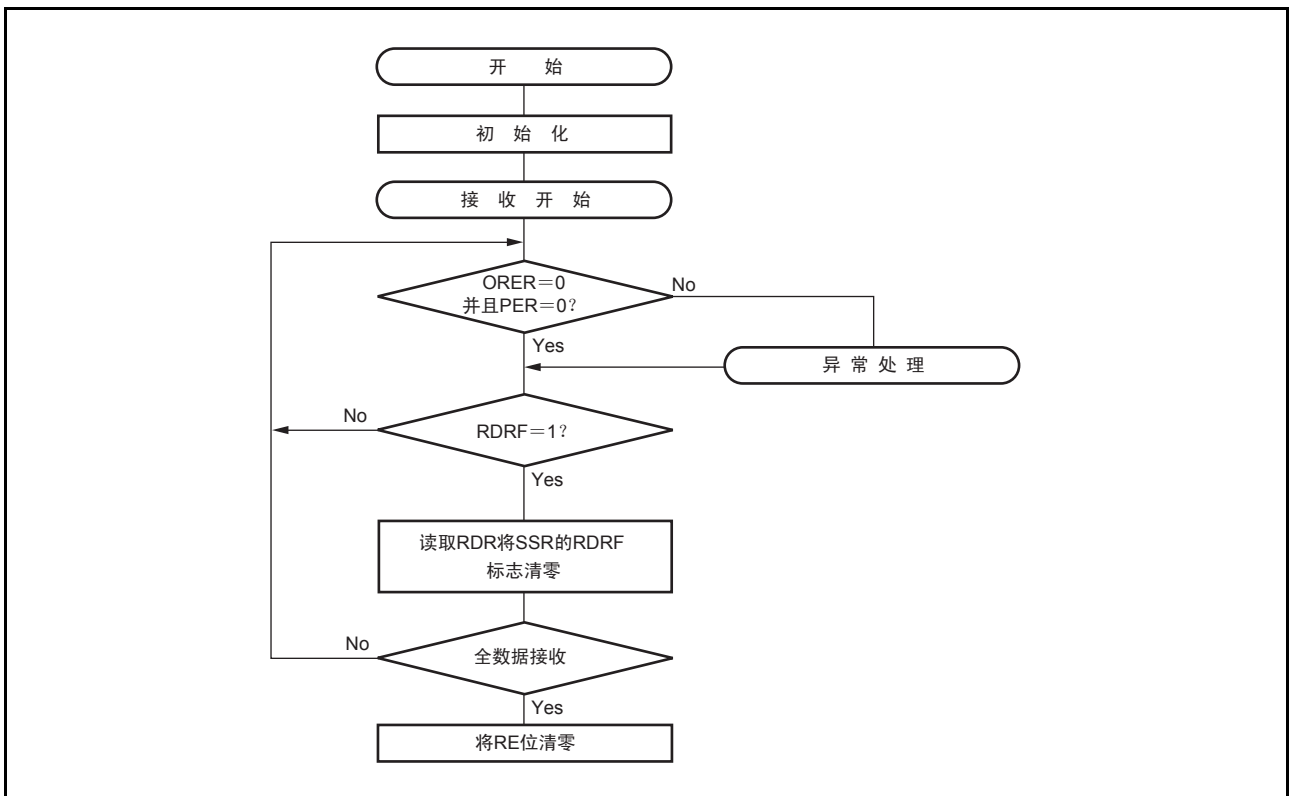


图 12.33 接收流程例

12.7.9 时钟输出控制

当 SMR 的 GM 位被置为 1 时, 根据 SCR 的 CKE1、CKE0 位可固定时钟输出。此时可将时钟脉冲的最小宽度定为指定宽度。

图 12.34 所示的是时钟输出的固定时序。在这个例子中 GM = 1、CKE1 = 0 时, 控制 CKE0 位。

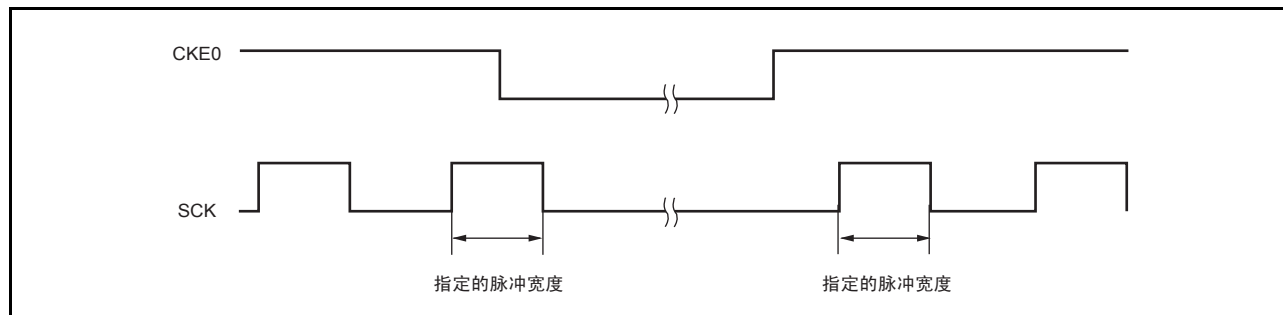


图 12.34 时钟输出固定时序

当加载电源时或在智能卡接口模式和软件待机模式之间切换时, 为了确保时钟的占空比, 请按以下顺序处理:

12.7.9.1 加载电源时

从加载电源时就要确保时钟占空比, 按照以下的切换顺序进行处理。

1. 初始状态时输入端口保持高阻抗状态。使用上拉电阻 / 下拉电阻来固定电压。
2. 用 SCR 的 CKE1 位将 SCK 管脚固定为指定的输出。
3. 设置 SMR 和 SCMR, 切换为智能卡模式。
4. 将 SCR 的 CKE0 位设定为 1, 开始时钟输出。

12.7.9.2 从智能卡接口模式转换到软件待机模式时

1. 将 SCK 管脚对应的数据寄存器 (DR) 和数据方向寄存器 (DDR) 设定为软件待机模式的输出固定状态值。
2. 在 SCR 的 TE 位和 RE 位写入 0, 使发送 / 接收停止。
同时也将 CKE1 位设定为软件待机时的输出固定状态值。
3. 在 SCR 的 CKE0 位写入 0, 使时钟输出停止。
4. 等待一个串行时钟周期。
在此期间, 保持占空比, 用指定电平固定时钟输出。
5. 转换到软件待机状态。

12.7.9.3 从软件待机模式返回到智能卡接口模式时

1. 请解除软件待机状态。
2. 在SCR的CKE0位写入1，输出时钟。根据正常的占空比开始发生信号。

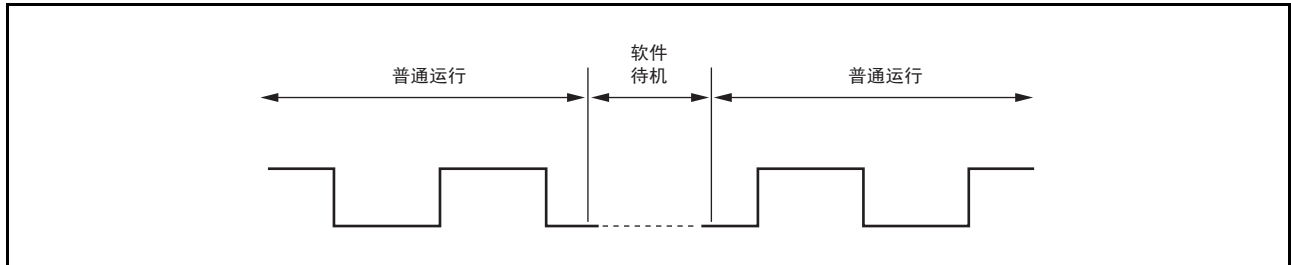


图 12.35 时钟停止重启顺序

12.8 SCI 选择功能 (时钟同步模式)

SCI_0 中有 SCI 选择功能，主 LSI 与多个从属 LSI (本 LSI) 间可进行 1 对多的时钟同步通信。图 12.36 所示的是通过 SCI 选择功能的通信例，图 12.37 SCI 选择功能的运行概要所示的是运行概要。

主 LSI 通过将 $\overline{\text{SEL_A}}$ 信号进行低电平输出、将 $\overline{\text{SEL_B}}$ 信号进行高电平输出，也可以与 LSI_A 进行 1 对 1 通信。此时从属 LSI_B 的 TxD0_B 管脚为高阻抗状态、内部 SCK0_A 信号固定为高电平，通信就会停止。主 LSI 通过将 $\overline{\text{SEL_A}}$ 信号置为高电平、将 $\overline{\text{SEL_B}}$ 信号置为低电平，就可以与从属 LSI_B 进行 1 对 1 通信。*

从属 LSI 检测出通过 IRQ7 的低电平输入中断进行选择时，可以立即执行发送接收数据处理。

【注】 * 主 LSI 选择信号 ($\overline{\text{SEL_A}}$ 、 $\overline{\text{SEL_B}}$ 等) 的切换，在发送数据的最终位输出后，串行时钟 (M_SCK) = High 时进行。并且，只有 1 个选择信号置为低电平输出。

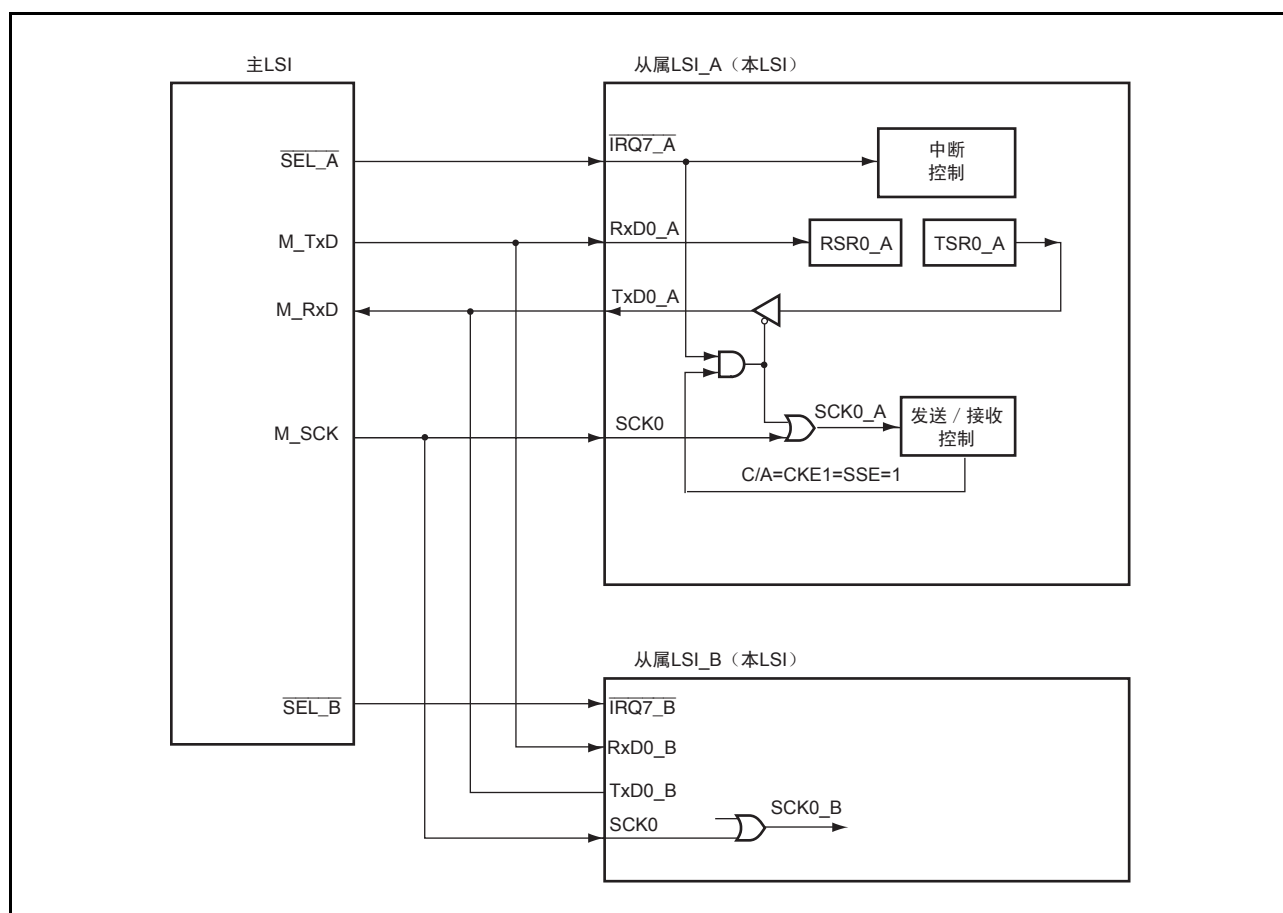


图 12.36 SCI 选择功能的通信例

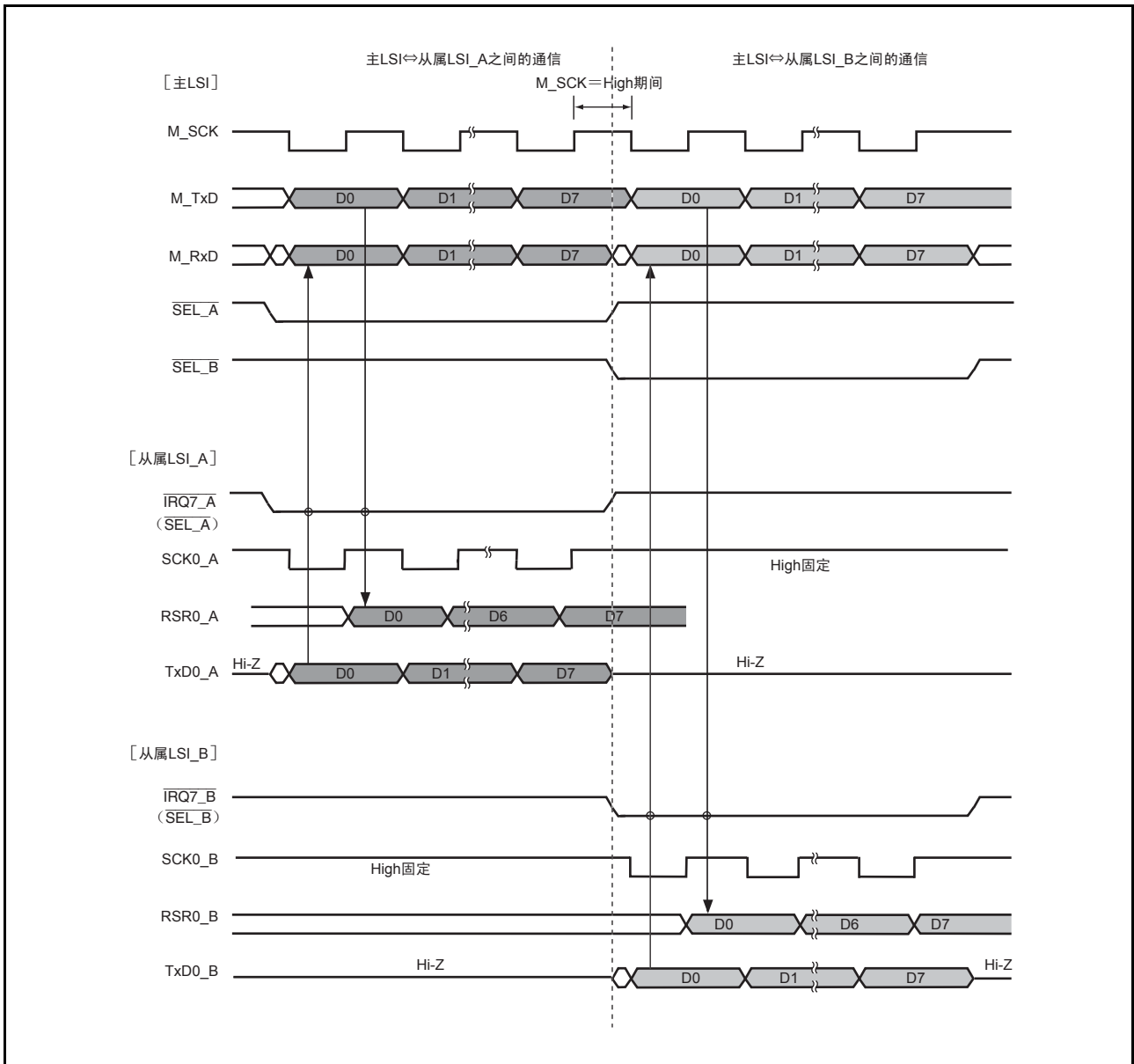


图 12.37 SCI 选择功能的运行概要

12.9 中断源

12.9.1 串行通信接口的中断

表 12.12 所示的是普通串行通信接口模式的中断源。给各中断源分配了不同的中断向量，可通过 SCR 的允许位进行独立允许操作。

当 SSR 的 TDRE 标志被置为 1，会发生 TXI 中断请求。当 SSR 的 TEND 标志置为 1，会发生 TEI 中断请求。通过 TXI 中断请求启动 DMAC，就可以传送数据。通过 DMAC 传送数据时，TDRE 标志会被自动清 0。

当 SSR 的 RDRF 标志置为 1，会发生 RXI 中断请求。当 SSR 的 ORER、PER、FER 标志任意一个被置为 1，均会发生 ERI 中断请求。通过 RXI 中断请求启动 DMAC，就可以进行数据传送。通过 DMAC 传送数据时，RDRF 会被清 0。

在 TEIE 位置为 1 的状态，TEND 标志被置为 1 时将发生 TEI 中断。同时发生 TEI 中断和 TXI 中断的状态时，先接受 TXI 中断。在 TXI 中断程序中，如果要同时清除 TDRE 标志和 TEND 标志，请注意 SCI 不能转移至 TEI 中断程序了。

表 12.12 SCI 中断源

通道	名称	中断源	中断标志	DMAC 启动	优先级*
0	ERI0	接收错误	ORER、FER、PER	不可	高 ↑ 低
	RXI0	接收数据满	RDRF	可	
	TXI0	接收数据空	TDRE	可	
	TEI0	发送完成	TEND	不可	
2	ERI2	接收错误	ORER、FER、PER	不可	
	RXI2	接收数据满	RDRF	不可	
	TXI2	接收数据空	TDRE	不可	
	TEI2	发送完成	TEND	不可	

【注】* 表示复位后的初始状态。通道间的优先级顺序可通过中断控制器变更。

12.9.2 智能卡接口模式下的中断

在智能卡接口模式，有如表 12.13 中的中断源。发送完成中断 (TEI) 请求在此模式下不能使用。数据块传送模式时，请参考 12.9.1 串行通信接口的中断。

表 12.13 智能卡接口模式的中断源

通道	名称	中断源	中断标志	DMAC 启动	优先级*
0	ERI0	检测出接收错误、错误信号	ORER、PER、ERS	不可	高 ↑ 低
	RXI0	接收数据满	RDRF	可	
	TXI0	发送数据空	TEND	可	
2	ERI2	检测出接收错误、错误信号	ORER、PER、ERS	不可	
	RXI2	接收数据满	RDRF	不可	
	TXI2	发送数据空	TEND	不可	

【注】* 表示复位后的初始状态。通道间优先级顺序可通过中断控制器进行变更。

12.10 使用注意事项

12.10.1 模块停止模式的设定

可通过模块停止控制寄存器，设定 SCI 的运行禁止 / 允许。为初始值时，SCI 的运行停止。通过解除模块停止模式，可进行寄存器的存取。详细内容请参考 20. 低功耗状态。

12.10.2 关于中止信号的检测和处理（仅限异步模式）

在检测帧错误时，通过直接读取 RxD 管脚值就可检测出中止状态。中止状态期间，RxD 管脚的输入全部为 0，因此 FER 和 PER 都被置位。SCI 接收到中止状态信号后，还可以继续接收。因此，请注意，即使将 FER 清零，也会被再次置为 1。

12.10.3 MARK 状态和中止检测（仅限异步模式）

TE 为 0 时，TxD 管脚成为通过 DDR 和 DR 决定输入输出方向和电平的 I/O 端口。使用它将 TxD 管脚置为 MARK 状态或在发送数据时送出中止信号。为了在 TE 置为 1 前，使通信线路保持 MARK 状态（1 的状态），设定 DDR = 1、DR = 1。这时，TE 被清零，TxD 管脚变为 I/O 端口并输出 1。另一方面，发送数据时，如果希望发送中止信号时，设定 DDR = 1、DR = 0 后，将 TE 清零。一旦 TE 清零，发送功能被初始化与当前发送状态无关，TxD 管脚变为 I/O 端口，从 TxD 管脚输出 0。

12.10.4 接收错误标志和发送运行（仅对时钟同步模式有效）

在接收错误标志（ORER、PER、FER）被置为 1 的状态，即使将 TDRE 清零，也不能开始发送。必须在发送开始时事先将接收错误标志清零。另外，请注意，即使将 RE 清零，也不能将接收错误标志清零。

12.10.5 DMAC 使用注意事项

- 使用外部时钟作为串行时钟时，在通过 DMAC 更新 TDR 后，发送时钟须经过 5ϕ 时钟以后方能输入。更新 TDR 后，如果输入小于 4 时钟的发送时钟，有时会发生误动作。（图 12.38）。
- 通过 DMAC 读取 RDR 时，必须设定与 SCI 的接受数据满（RXI）相关的启动源。

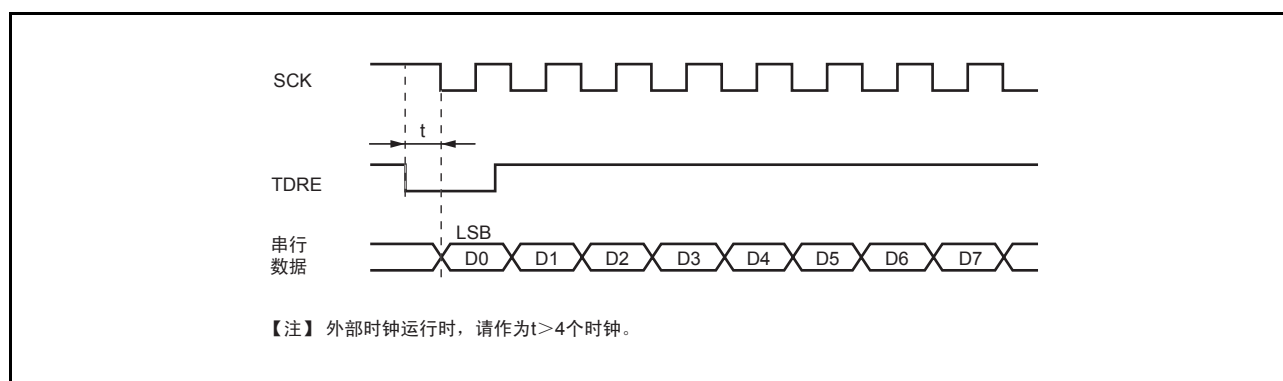


图 12.38 通过 DMAC 进行时钟同步模式发送时

12.10.6 模式迁移

12.10.6.1 发送

停止运行后 (TE=TIE=TEIE=0) 才可向模块停止模式、软件待机模式、监视模式、子激活模式或子睡眠模式转移。并且 TSR、TDR 及 SSR 被复位。依据端口的设定决定模块处于停止模式、软件待机模式、监视模式、子激活模式或子睡眠模式期间的输出状态，模式解除后成为高电平输出。如果发送数据时发生转移，正在发送中的数据变的不确定。

模式解除后，不改变发送模式发送时，返回到 TE=1，按照“SSR 读取→TDR 写入→TDRE 清除为 0”的顺序，开始发送。解除后，改变发送模式发送时，首先要对 SCI 进行出初始化。图 12.39 中所示的是发送模式迁移流程图例。图 12.40、图 12.41 所示的是端口管脚状态。

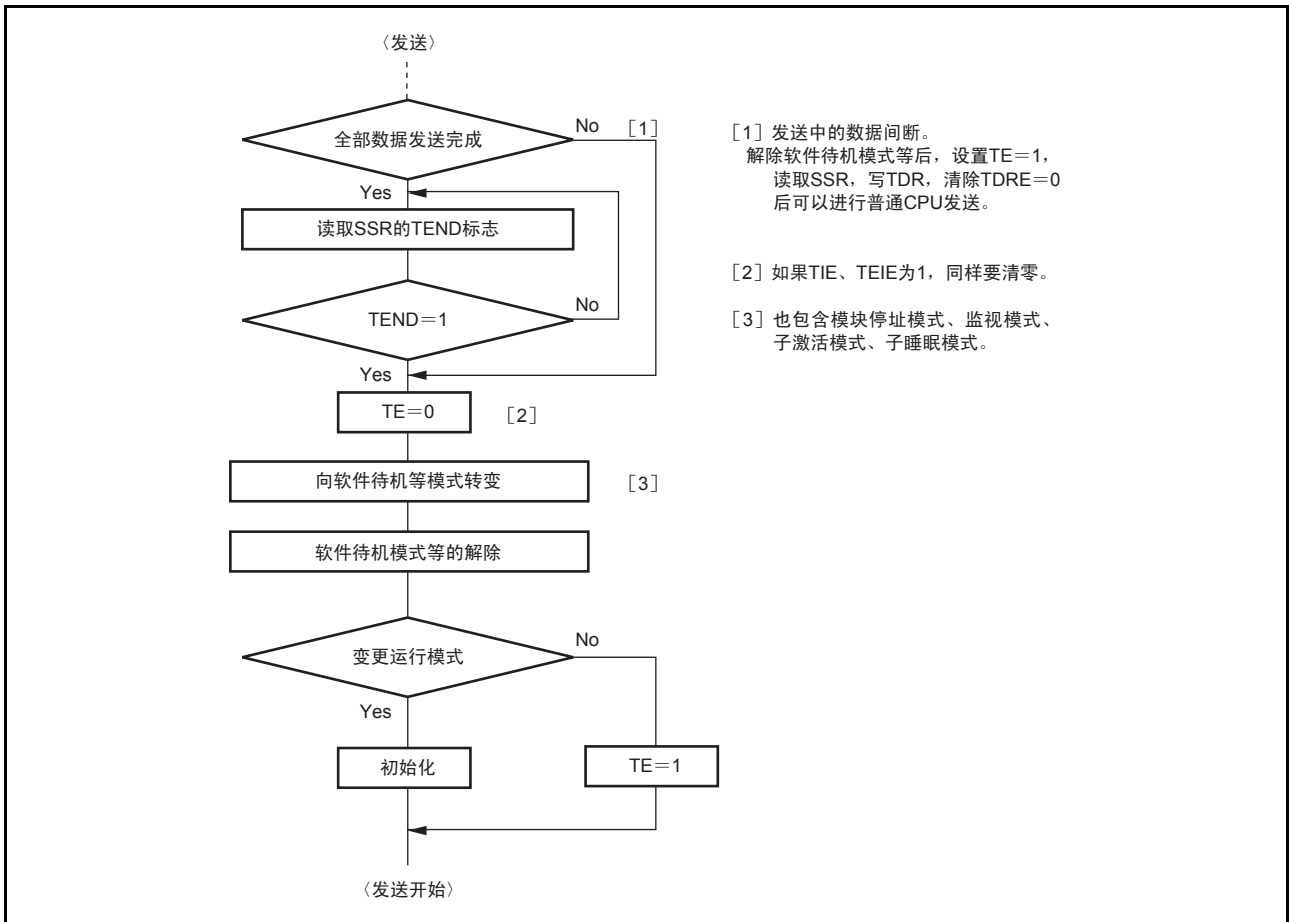


图 12.39 发送时的模式转换流程图例

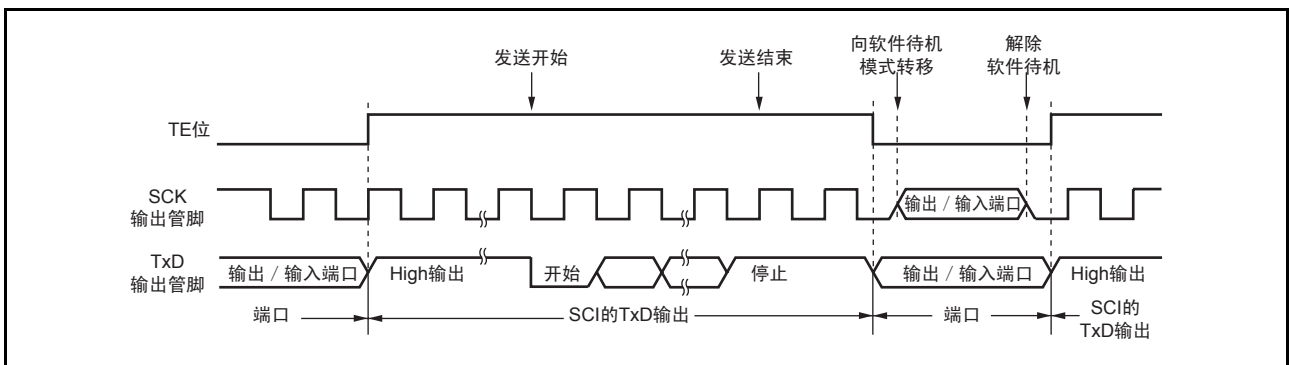


图 12.40 内部时钟、异步模式发送时的端口管脚状态

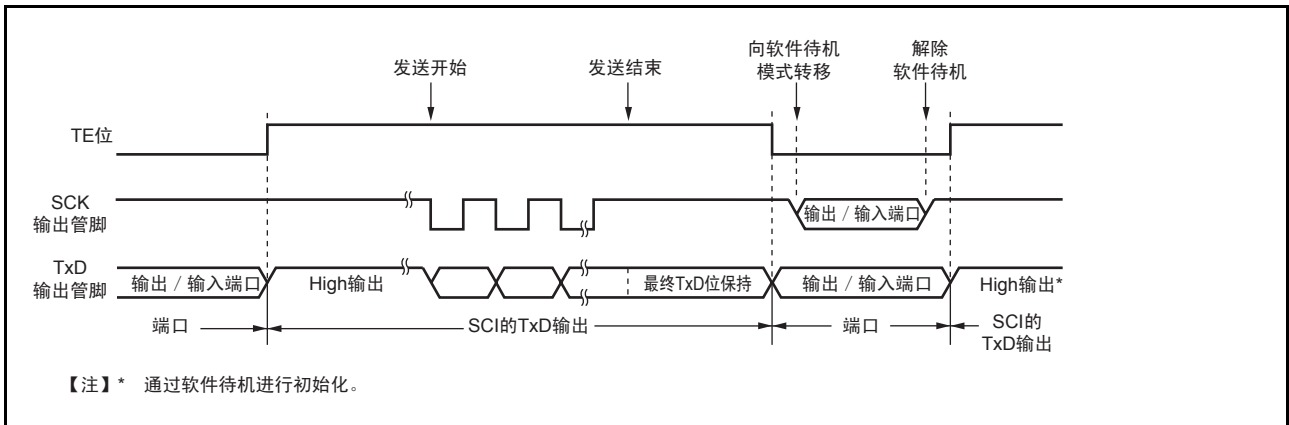


图 12.41 内部时钟、时钟同步模式发送时的端口管脚状态

12.10.6.2 接收

接收运行停止 (RE=0) 后才可向模块停止模式、软件待机模式、监视模式、子激活模式或子睡眠模式转移。并且 RSR、RDR 及 SSR 被复位。如果没有停止发生转换时, 正在接收的数据将是无效的。

模式解除后, 不改变接收模式接收数据时, 设定 RE=1 后, 开始接收。改变接收模式接收数据时, 首先要对 SCI 进行初始化。

图 12.42 所示的是接收时的模式转换流程图。

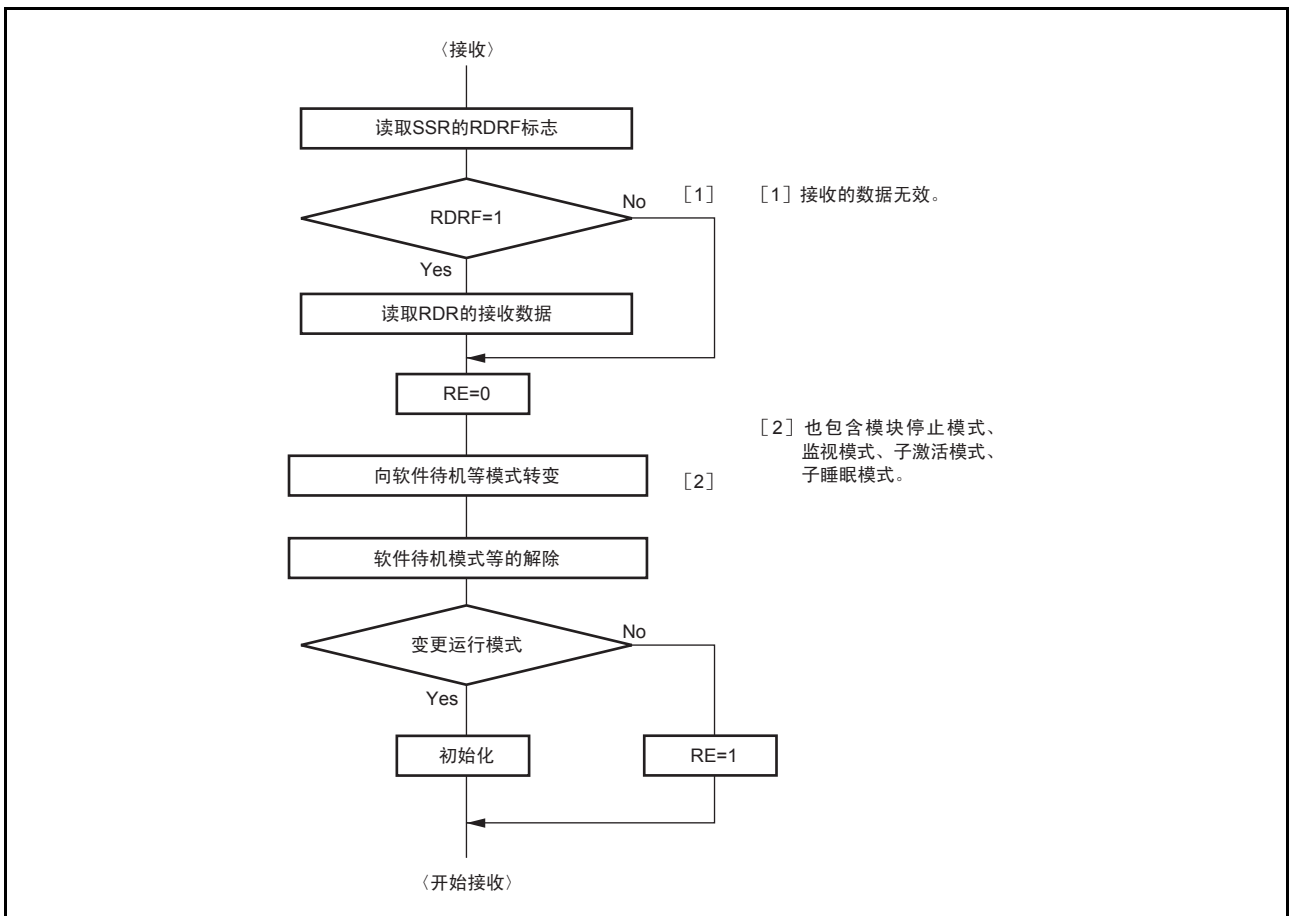


图 12.42 接收时的模式转换流程图例

12.10.7 从 SCK 管脚向端口管脚切换注意事项

设定 $DDR=1$ 、 $DR=1$ ，使用同步时钟 SCI 时钟输出，在发送完成状态将 SCK 管脚切换到端口时，经过半个周期的低电平输出后，成为端口输出。

在 $DDR=1$ 、 $DR=1$ 、 $\overline{C/A}=1$ 、 $CKE1=0$ 、 $CKE0=0$ 、 $TE=1$ 的状态，根据以下设定切换到端口时，会发生半个周期的低电平输出。

1. 发送串行数据完成
2. TE 位 = 0
3. $\overline{C/A}$ 位 = 0...切换到端口输出
4. 发生低电平输出 (参考图 12.43)

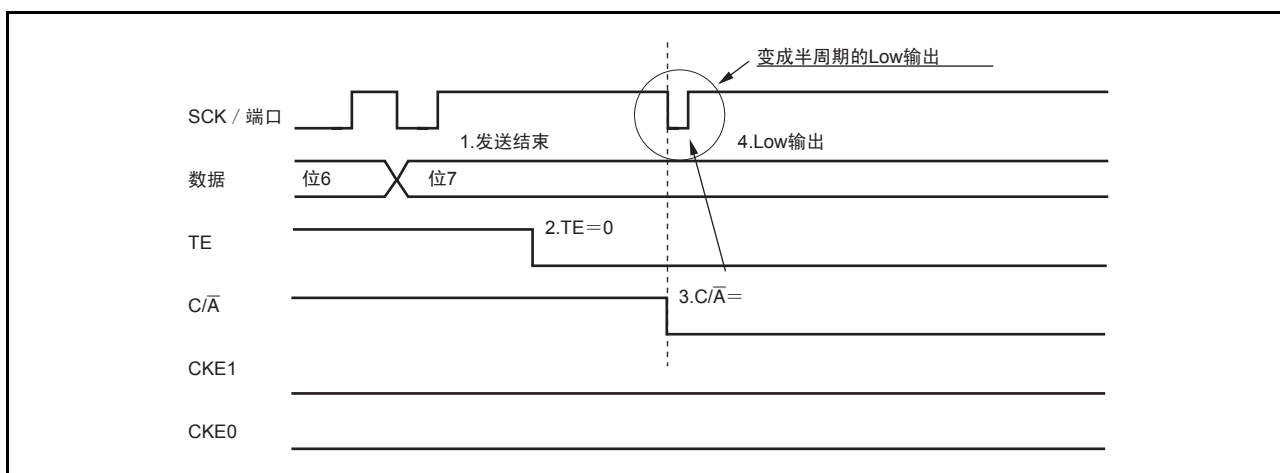


图 12.43 从 SCK 管脚向端口管脚切换时的操作时序

为避免将 SCK 管脚切换为端口时，发生低电平输出，请按以下的顺序进行：

由于这个顺序是将 SCK 管脚作为一次输入状态，所以请事先使用外部电路上拉 SCK / 端口管脚。在 $DDR=1$ 、 $DR=1$ 、 $\overline{C/A}=1$ 、 $CKE1=0$ 、 $CKE0=0$ 、 $TE=1$ 的状态开始按照 1 ~ 5 的顺序进行设定。

1. 串行数据发送完成。
2. TE 位 = 0
3. $\overline{CKE1}$ 位 = 1
4. $\overline{C/A}$ 位 = 0...切换到端口输出
5. $\overline{CKE1}$ 位 = 0

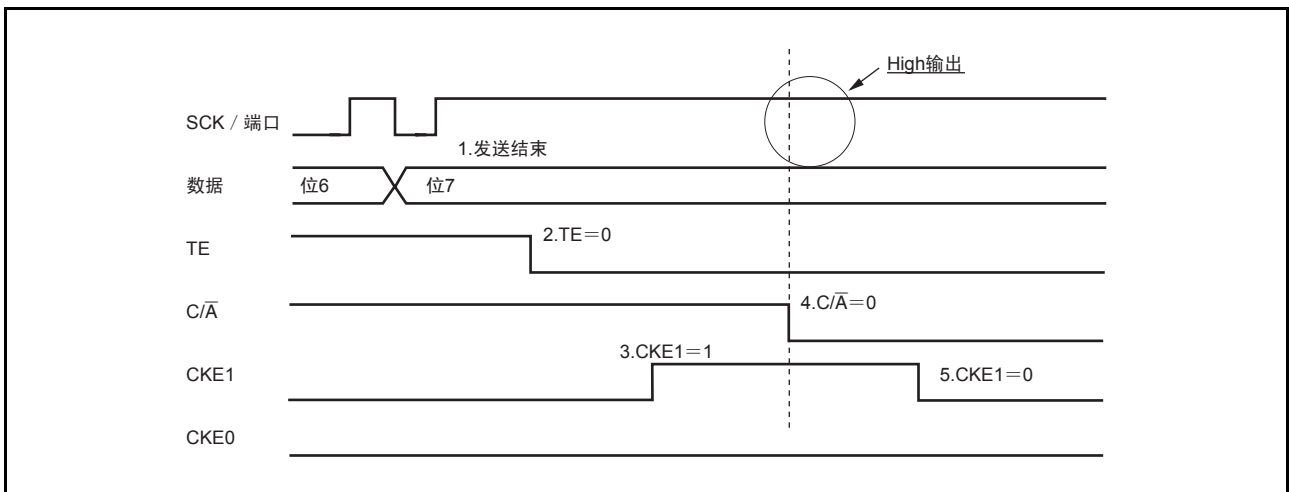


图 12.44 从 SCK 管脚向端口管脚切换时的操作时序 (避免低电平输出样例)

13. 边界扫描

HD64F2218、HD64F2218U 内置边界扫描功能。边界扫描是遵循 JTAG（Joint Test Action Group, IEEEStd. 1149.1 and IEEE Standard Test Access Port and Boundary-Scan Architecture）标准的串行输入输出接口。边界扫描的框图如图 13.1 所示。

13.1 特点

- 5个测试信号
TCK、TDI、TDO、TMS、 $\overline{\text{TRST}}$
- 支持6种测试模式
BYPASS、SAMPLE/PRELOAD、EXTEST、CLAMP、HIGHZ、IDCODE
- 以下管脚不支持边界扫描
电源相关管脚（VCC、VSS、Vref、PLL VCC、PLL VSS、DrVCC、DrVSS）
时钟相关信号（EXTAL、XTAL、OSC2、OSC1）
模拟相关信号（P40~P43、P96、P97、USD+、USD-）
边界扫描相关信号（TCK、TDI、TDO、TMS、 $\overline{\text{TRST}}$ ）
H-UDI控制信号（EMLE）

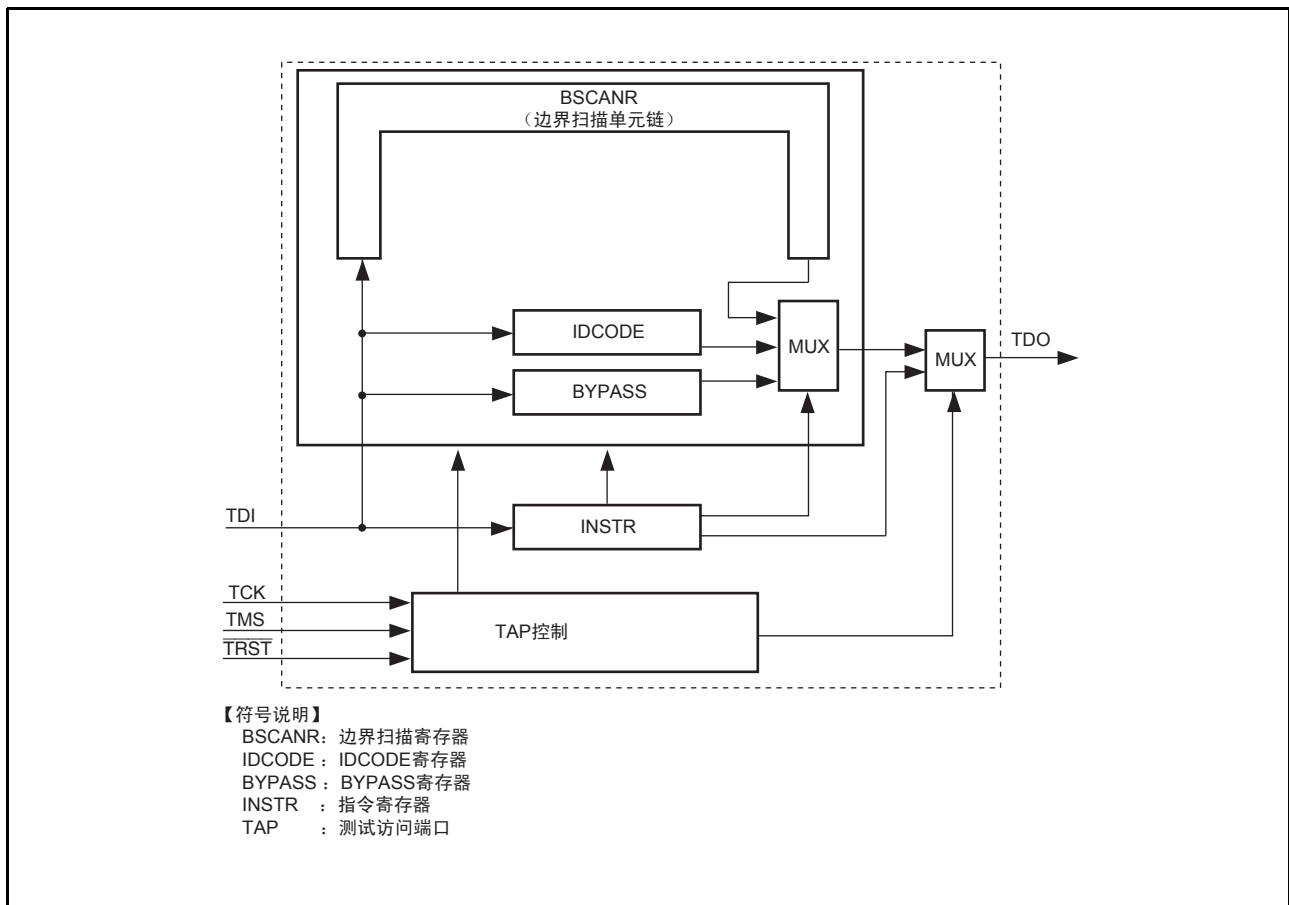


图 13.1 边界扫描的框图

13.2 输入输出管脚

在边界扫描下使用的管脚如表 13.1 所示。

表 13.1 管脚结构

管脚名称	输入输出	功能
TMS	输入	TEST MODE SELECT TAP 控制器（拥有 16 状态的有限状态机（Finite State Machine））的控制信号。 TCK 上升沿的 TMS 输入值决定 TAP 控制器的状态转移方向。不使用边界扫描功能时，要固定为 1。 协议遵守 JTAG 标准（IEEE Std. 1149.1）。 本管脚带上拉电阻。
TCK	输入	TEST CLOCK 是边界扫描用的时钟信号。 使用边界扫描功能时，请输入占空比 50% 的时钟波形。 本管脚带上拉电阻。
TDI	输入	TEST DATA INPUT 是边界扫描用的数据输入信号。 在 TCK 的上升沿取得输入数据。 不使用边界扫描功能时固定为 1。 本管脚附带上拉电阻。
TDO	输出	TEST DATA OUTPUT 边界扫描用的数据输出信号。 在 TCK 的下降沿输出数据。 输出驱动器仅在需要的时候（仅 Shift-IR、Shift-DR 状态）驱动，其他时候都是高阻抗状态。
$\overline{\text{TRST}}$	输入	TEST RESET $\overline{\text{TRST}} = \text{Low}$ 电平，TAP 控制器不同步复位。 用户在插入电源时，给边界扫描专用的加电复位信号加电（请参考 13.5 使用注意事项）。 本管脚附带上拉电阻。

13.3 寄存器说明

边界扫描中有以下寄存器。

内部 CPU 不能访问这些寄存器。

- 指令寄存器 (INSTR)
- IDCODE 寄存器 (IDCODE)
- BYPASS 寄存器 (BYPASS)
- 边界扫描寄存器 (BSCANR)

13.3.1 指令寄存器 (INSTR)

指令寄存器是 3 位寄存器。在初始状态此寄存器变为 IDCODE 模式。当 $\overline{\text{TRST}}$ 为低电平状态或者 TAP 控制器的 Test-Logic-Reset 状态时，本寄存器就会被初始化。本寄存器可通过来自 TDI 的串行数据输入进行写入。如果从 TDI 输入超过 3 位的指令，会在指令寄存器中保存串行数据的最后 3 位。

本寄存器不能保证设定预约指令时的运行。

位	位名	初始值	R/W	说明
2	TI2	1	—	测试指令位 表 13.2 所示的是指令结构。
1	TI1	0	—	
0	TI0	1	—	

表 13.2 指令结构

位 2	位 1	位 0	指 令
TI2	TI1	TI0	
0	0	0	EXTEST
0	0	1	SAMPLE/PRELOAD
0	1	0	CLAMP
0	1	1	HIGHZ
1	0	0	保留
1	0	1	IDCODE (初始值)
1	1	0	保留
1	1	1	BYPASS

13.3.1.1 EXTEST

将本 LSI 实装到印刷电路板时，EXTEST 命令用于测试外部电路。运行本命令时，输出管脚将测试数据（已经用 SAMPLE/PRELOAD 命令设定）从扫描边界寄存器输出到印刷电路板，输入管脚将测试结果从印刷电路板提取到扫描边界寄存器。

13.3.1.2 SAMPLE/PRELOAD

SAMPLE/PRELOAD 指令用于从本 LSI 的内部电路向边界扫描寄存器输入数据，从扫描路径输出数据，将数据再装载到扫描路径的指令。当本指令在运行中，输入信号直接输入本 LSI，而输出信号也直接输出外部电路。本指令运行，本 LSI 的系统电路就不会受任何影响。

SAMPLE 运行中，从输入管脚传送到内部电路的值及从内部电路传送到输出管脚的值（snap shot），锁存至边界扫描寄存器，再从扫描路径读出。Snap shot 数据的锁存与 Capture-DR 状态的 TCK 上升同步进行。Snap shot 数据的锁存不影响本 LSI 的正常运行。

PRELOAD 运行中，优先于 EXTEST 指令，从扫描路径向边界扫描寄存器的并行输出锁存器设定初始值。如果 PRELOAD 没有运行，执行 EXTEST 指令时，在最初的扫描顺序结束前（传送到输出锁存器），从输出管脚输出不定值（通过 EXTEST 指令，并行锁存器通常输出到输出管脚）。

13.3.1.3 CLAMP

如果选择 CLAMP 指令，输出管脚提前输出用 SAMPLE/PRELOAD 指令设定的边界扫描寄存器的值。选择 CLAMP 指令期间，与 TAP 控制器的状态无关，边界扫描寄存器保持之前的状态。TDI-TDO 之间被 BYPASS 寄存器连接，与选择 BYPASS 指令时的运行相同。

13.3.1.4 HIGHZ

如果选择 HIGHZ 指令，所有的输出管脚变为高阻态。选择 HIGHZ 指令期间，与 TAP 控制器的状态无关，边界扫描寄存器保持之前的状态。TDI-TDO 之间被 BYPASS 寄存器连接，与选择 BYPASS 指令时的运行相同。

13.3.1.5 IDCODE

如果选择 IDCODE 指令，在 TAP 控制器的 Shift-DR 状态时，将 IDCODE 寄存器的值通过 LSB 从 TDO 输出。此指令运行中，测试电路对系统电路没有任何影响。在 TAP 控制器的 Test-Logic-Reset 状态下，指令寄存器被初始化为 IDCODE 命令。

13.3.1.6 BYPASS

BYPASS 指令是使 BYPASS 寄存器运行所必需的标准指令。此指令用于缩短扫描途径，提高对印刷电路板上其他 LSI 的串行数据的传送速度。此指令运行中，测试电路对系统电路没有任何影响。

13.3.2 IDCODE 寄存器（IDCODE）

IDCODE 寄存器是 32 位的寄存器。如果将指令寄存器设定为 IDCODE 模式，IDCODE 寄存器就会被连接在 TDI 和 TDO 之间，从 TDO 输出固定代码（HD64F2218 和 HD64F2218U 是 H'002A200F）。不能通过 TDI 把串行数据写入到 IDCODE 寄存器。表 13.3 所示的是 IDCODE 寄存器构成。

表 13.3 IDCODE 寄存器结构

位	31……28	27……12	11……1	0
HD64F2218、 HD64F2218U 的代码	0000	0000 0010 1010 0010	0000 0000 111	1
内容	版本 (4 位)	零件号码 (16 位)	制造者号码 (11 位)	固定代码 (1 位)

13.3.3 BYPASS 寄存器（BYPASS）

BYPASS 寄存器是 1 位的寄存器。如果将指令寄存器设定为 BYPASS 模式、CLAMP 模式、HIGHZ 模式的任意一个，BYPASS 寄存器就会被连接在 TDI 和 TDO 之间。

13.3.4 边界扫描寄存器（BSCANR）

边界扫描寄存器是为了控制输入输出管脚，在管脚上配置的 199 位的移位寄存器。LSI 的输入输出管脚由 3 位（IN、Control、OUT）、输入专用管脚 1 位（IN）、输出专用管脚 1 位（OUT）的移位寄存器构成。

使用表 13.2 的指令，可以进行遵循 JTAG 标准的边界扫描测试。表 13.4 所示的是本 LSI 的管脚和边界扫描寄存器的相互对照（表中的 Control 是 High active 的信号。将 Control 设定为 High，用 OUT 的值驱动相关的管脚。图 13.2 所示的是边界扫描寄存器的简略图）。

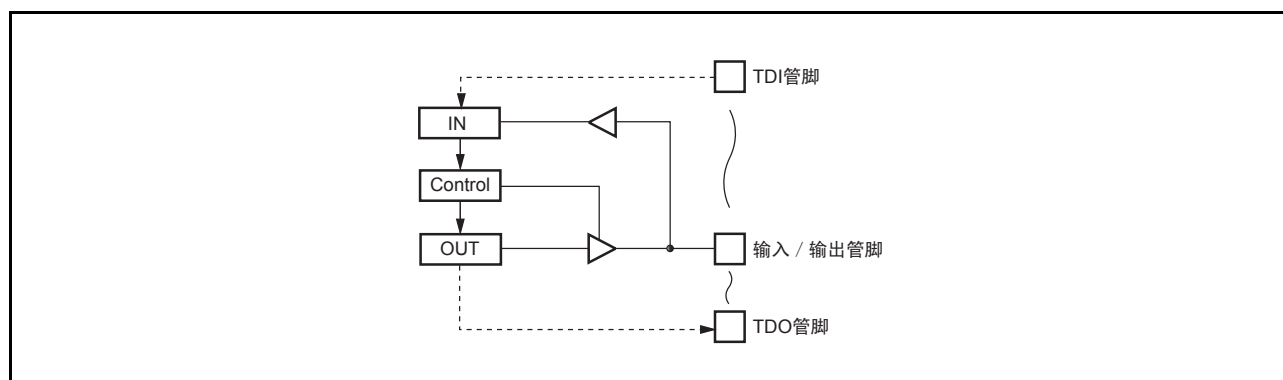


图 13.2 边界扫描寄存器的简略图（输入输出管脚的例）

表 13.4 本 LSI 的管脚和边界扫描寄存器的对应

TFP-100G TFP-100GV Pin No	BP-112 BP-112V Pin No	管脚名称	输入输出	位名
From TDI				
89	A6	PF7/ ϕ	IN	198
			Control	197
			OUT	196
91	D6	PF6/ $\overline{\text{AS}}$	IN	195
			Control	194
			OUT	193
92	A5	PF5/ $\overline{\text{RD}}$	IN	192
			Control	191
			OUT	190
93	B5	PF4/ $\overline{\text{HWR}}$	IN	189
			Control	188
			OUT	187
94	C5	PF3/ $\overline{\text{LWR}}/\overline{\text{ADTRG}}/\overline{\text{IRQ3}}$	IN	186
			Control	185
			OUT	184
95	A4	PF2/ $\overline{\text{WAIT}}$	IN	183
			Control	182
			OUT	181
96	D5	PF1/ $\overline{\text{BACK}}$	IN	180
			Control	179
			OUT	178
97	B4	PF0/ $\overline{\text{BREQ}}/\overline{\text{IRQ2}}$	IN	177
			Control	176
			OUT	175
98	A3	PA3/A19/SCK2	IN	174
			Control	173
			OUT	172
99	C4	PA2/A18/RxD2	IN	171
			Control	170
			OUT	169
100	B3	PA1/A17/TxD2	IN	168
			Control	167
			OUT	166

TFP-100G TFP-100GV Pin No	BP-112 BP-112V Pin No	管脚名称	输入输出	位名
1	B2	PA0/A16	IN	165
			Control	164
			OUT	163
2	B1	P10/TIOCA0/A20	IN	162
			Control	161
			OUT	160
3	D4	P11/TIOCB0/A21	IN	159
			Control	158
			OUT	157
4	C2	P12/TIOCC0/TCLKA/A22	IN	156
			Control	155
			OUT	154
5	C1	P13/TIOCD0/TCLKB/A23	IN	153
			Control	152
			OUT	151
6	D3	P14/TIOCA1/ $\overline{\text{IRQ0}}$	IN	150
			Control	149
			OUT	148
7	D2	P15/TIOCB1/TCLKC	IN	147
			Control	146
			OUT	145
8	D1	P16/TIOCA2/ $\overline{\text{IRQ1}}$	IN	144
			Control	143
			OUT	142
9	E4	P17/TIOCB2/TCLKD	IN	141
			Control	140
			OUT	139
10	E3	PC0/A0	IN	138
			Control	137
			OUT	136
11	E1	PC1/A1	IN	135
			Control	134
			OUT	133

TFP-100G TFP-100GV Pin No	BP-112 BP-112V Pin No	管脚名称	输入输出	位名
12	E2	PC2/A2	IN	132
			Control	131
			OUT	130
13	F3	PC3/A3	IN	129
			Control	128
			OUT	127
14	F1	MD0	IN	126
15	F2	MD1	IN	125
16	F4	MD2	IN	124
17	G1	PC4/A4	IN	123
			Control	122
			OUT	121
18	G2	PC5/A5	IN	120
			Control	119
			OUT	118
19	G3	PC6/A6	IN	117
			Control	116
			OUT	115
20	H1	PC7/A7	IN	114
			Control	113
			OUT	112
21	G4	USPND/TMOW	OUT	111
22	H2	P30/TxD0	IN	110
			Control	109
			OUT	108
23	J1	P31/RxD0	IN	107
			Control	106
			OUT	105
24	H3	P32/SCK0/IRQ4	IN	104
			Control	103
			OUT	102
25	J2	PG1/CS3/IRQ7	IN	101
			Control	100
			OUT	99

TFP-100G TFP-100GV Pin No	BP-112 BP-112V Pin No	管脚名称	输入输出	位名
26	K2	PG2/ $\overline{\text{CS2}}$	IN	98
			Control	97
			OUT	96
27	L2	PG3/ $\overline{\text{CS1}}$	IN	95
			Control	94
			OUT	93
28	H4	PG4/ $\overline{\text{CS0}}$	IN	92
			Control	91
			OUT	90
29	K3	VBUS		89
30	L3	P36	IN	88
			Control	87
			OUT	86
37	K5	PB0/A8	IN	85
			Control	84
			OUT	83
38	J6	PB1/A9	IN	82
			Control	81
			OUT	80
39	L6	PB2/A10	IN	79
			Control	78
			OUT	77
40	K6	PB3/A11	IN	76
			Control	75
			OUT	74
47	L9	$\overline{\text{UBPM}}$	IN	73
49	K9	PB4/A12	IN	72
			Control	71
			OUT	70
50	L10	PB5/A13	IN	69
			Control	68
			OUT	67

TFP-100G TFP-100GV Pin No	BP-112 BP-112V Pin No	管脚名称	输入输出	位名
51	K10	PB6/A14	IN	66
			Control	65
			OUT	64
52	K11	PB7/A15	IN	63
			Control	62
			OUT	61
55	H9	P74/MRES	IN	60
			Control	59
			OUT	58
56	H10	P71/CS5	IN	57
			Control	56
			OUT	55
57	H11	STBY	IN	54
58	G8	RES	IN	53
63	F11	P70/CS4	IN	52
			Control	51
			OUT	50
64	F10	PE0/D0	IN	49
			Control	48
			OUT	47
65	F8	PE1/D1	IN	46
			Control	45
			OUT	44
66	E11	PE2/D2	IN	43
			Control	42
			OUT	41
67	E10	PE3/D3	IN	40
			Control	39
			OUT	38
68	E9	PE4/D4	IN	37
			Control	36
			OUT	35

TFP-100G TFP-100GV Pin No	BP-112 BP-112V Pin No	管脚名称	输入输出	位名
69	D11	PE5/D5	IN	34
			Control	33
			OUT	32
70	E8	PE6/D6	IN	31
			Control	30
			OUT	29
71	D10	PE7/D7	IN	28
			Control	27
			OUT	26
72	C11	PD0/D8	IN	25
			Control	24
			OUT	23
73	D9	PD1/D9	IN	22
			Control	21
			OUT	20
74	C10	PD2/D10	IN	19
			Control	18
			OUT	17
75	B11	PD3/D11	IN	16
			Control	15
			OUT	14
76	B10	PD4/D12	IN	13
			Control	12
			OUT	11
77	A10	PD5/D13	IN	10
			Control	9
			OUT	8
78	D8	PD6/D14	IN	7
			Control	6
			OUT	5
79	B9	PD7/D15S	IN	4
			Control	3
			OUT	2
80	A9	FWE	IN	1

TFP-100G TFP-100GV Pin No	BP-112 BP-112V Pin No	管脚名称	输入输出	位名
81	C8	NMI	IN	0
to TDO				

13.4 运行说明

13.4.1 TAP 控制器

图 13.3 所示的是遵循 JTAG 规定 TAP 控制器的状态转换图。

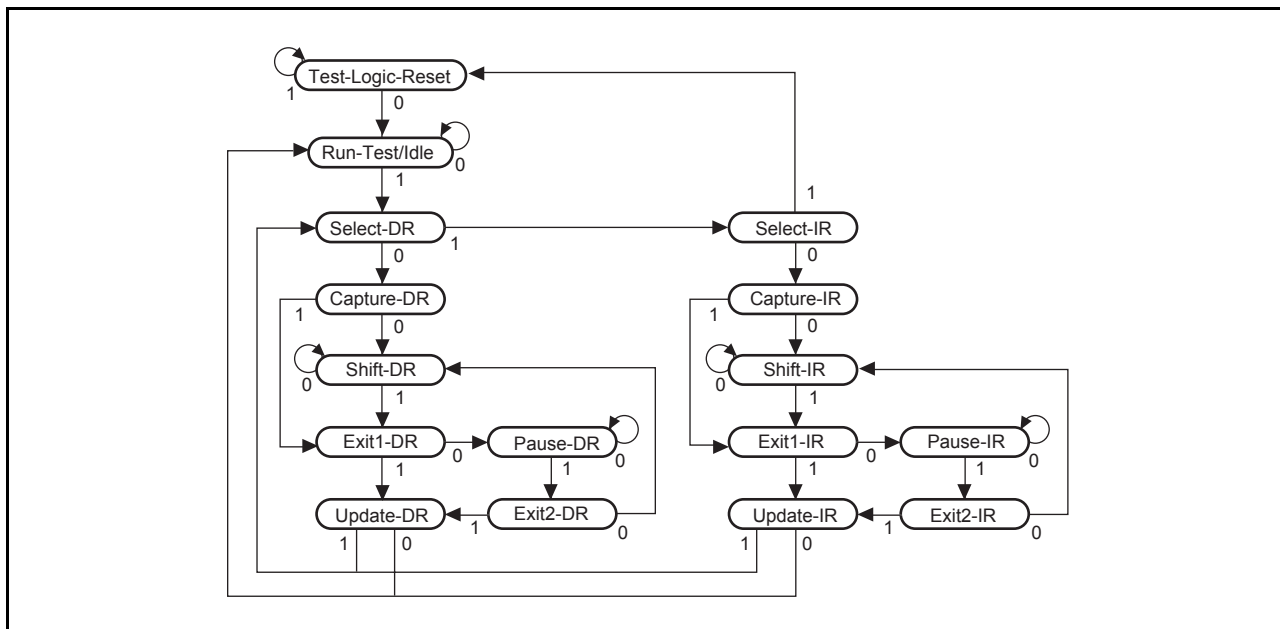


图 13.3 TAP 控制器状态转换图

【注】 转换条件：TMS 值是 TCK 上升沿的值。在 TCK 的上升沿采样 TDI 值，在 TCK 的下降沿移位。TDO 值在 TCK 的下降沿发生变化。并且，TDO 除了 Shift-DR、Shift-IR 状态外都是高阻态。TRST=0 时，与 TCK 异步转换至 Test-Logic-Reset 状态。

13.5 使用注意事项

- 使用边界扫描功能，插入电源时 $\overline{\text{TRST}} = 0$ ，经过 t_{RESW} 期间后，使 $\overline{\text{TRST}} = 1$ ，适当设定TCK、TMS、TDI。
不使用边界扫描正常运行时，将TCK、TMS、TDI设定为Hi-Z，插入电源时 $\overline{\text{TRST}} = 0$ ，经过 t_{RESW} 期间后，请设定为 $\overline{\text{TRST}} = 1$ 或者Hi-Z。由于这些管脚在芯片内被上拉，如果设定为1时的管脚输入电压值和电源电压 V_{cc} 之间产生电位差，会有灌电流流通，因此在待机时请注意。
- 给 $\overline{\text{TRST}}$ 输入的加电复位信号，请考虑以下几点。
 - 插入电源时，请务必输入复位信号。
 - 为了使板上测试器的 $\overline{\text{TRST}}$ 信号不影响LSI的系统运行，请将电路分离。
 - 相反，为了使LSI的系统复位不影响板上测试器的 $\overline{\text{TRST}}$ 信号，请分离电路。图13.4中所示的是不相互干扰的复位信号的设计例。

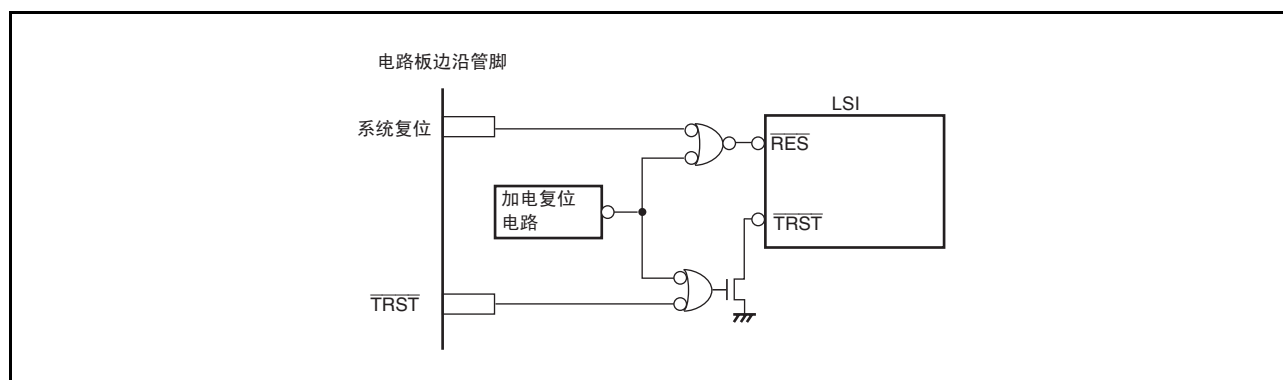


图 13.4 互相不干扰的复位信号的设计例

- 要使TCK的时钟速度低于系统时钟频率。
- 从LSB开始串行传送时的数据输入输出。图13.5所示的是串行数据输入输出。

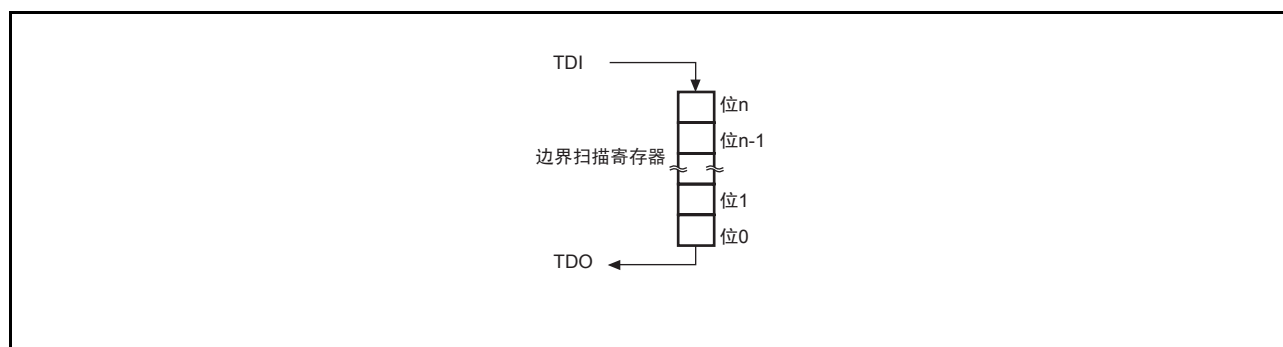


图 13.5 串行数据输入输出

5. 在带有上拉功能的管脚，上拉功能有效的状态，如果SAMPLE，就可以在对应的IN寄存器看到1。此时要将对应的控制寄存器置为0。
6. 带有漏极开路功能的管脚在漏极开路功能有效的状态，如果对应的OUT寄存器为1时SAMPLE，控制寄存器变为0（此时管脚状态为Hi-Z）。

同样，如果对应的OUT寄存器为0时SAMPLE，控制寄存器变为1（此时管脚状态为0输出）。

7. 如果迁移至EXTEST、CLAMP、HIGHZ模式，本LSI变为启动保护的状态（ $\overline{\text{RES}} = \overline{\text{STBY}} = 0$ 的硬件待机状态）。请将 $\overline{\text{RES}}$ 、 $\overline{\text{STBY}}$ 、FEW、MD2~MD0管脚设定为希望使其运行的模式后，再进行从EXTEST、CLAMP、HIGHZ向其他模式迁移（作为正常芯片运行的模式）。
8. 请将EMLE管脚置为0。如果置为1，本芯片就成为用户调试接口（H-UDI）。

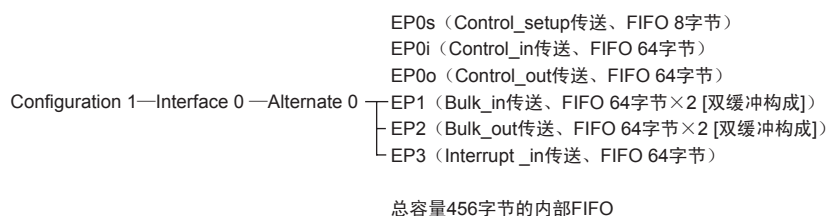
EMLE 管脚	芯片状态
0	正常芯片运行、边界扫描
1	用户调试接口（H-UDI）

14. 通用串行总线 (USB)

本 LSI 内置遵循 USB 标准的 USB 功能模块。USB 框图如图 14.1 所示。

14.1 特点

- 支持 USB 规格 Ver2.0 的全速模式 (12Mbps)
- 可通过专用管脚 (UBPM) 选择总线电源模式和自身电源模式
- 内置 USB 运行时钟生成用的 PLL 电路 (24MHz × 2 = 48MHz 或 16MHz × 3 = 48MHz)
- 内置总线收发器电路
- 可通过硬件自动处理标准指令
Set_Descriptor、Get_Descriptor、Class/VendorCommand、SynchFrame 指令必须通过软件处理。
- 可通过 Set_Configuration 中断, 检查当前的 Configuration 值
- 支持 3 种传送模式 (Control 传送、Bulk 传送、Interrupt 传送)
- 由 EP0、EP1、EP2、EP3 四个终点构成



- 中断源: 16 种
可在 IRQ6 中设定 suspend/resume 中断源
除过 suspend/resume 以外的各中断源可通过寄存器设定为 EXIRQ0 或者 EXIRQ1 中的任意一个
- DMA 传送接口
EP1 和 EP2 的 Bulk 传送数据可进行 DMA 传送
- 使用外部总线接口方式的 8 位总线 3 周期存取时序
内置寄存器被分配到外部地址的区域 6 (H'C00000 ~ H'DFFFFFF)。
地址 H'C00100 ~ H'DFFFFFF 为 USB 保留区域, 是禁止访问区域。

【注】 在本章, 把监视模式、子激活模式、子睡眠模式、软件待机模式总称为各种低功耗模式。

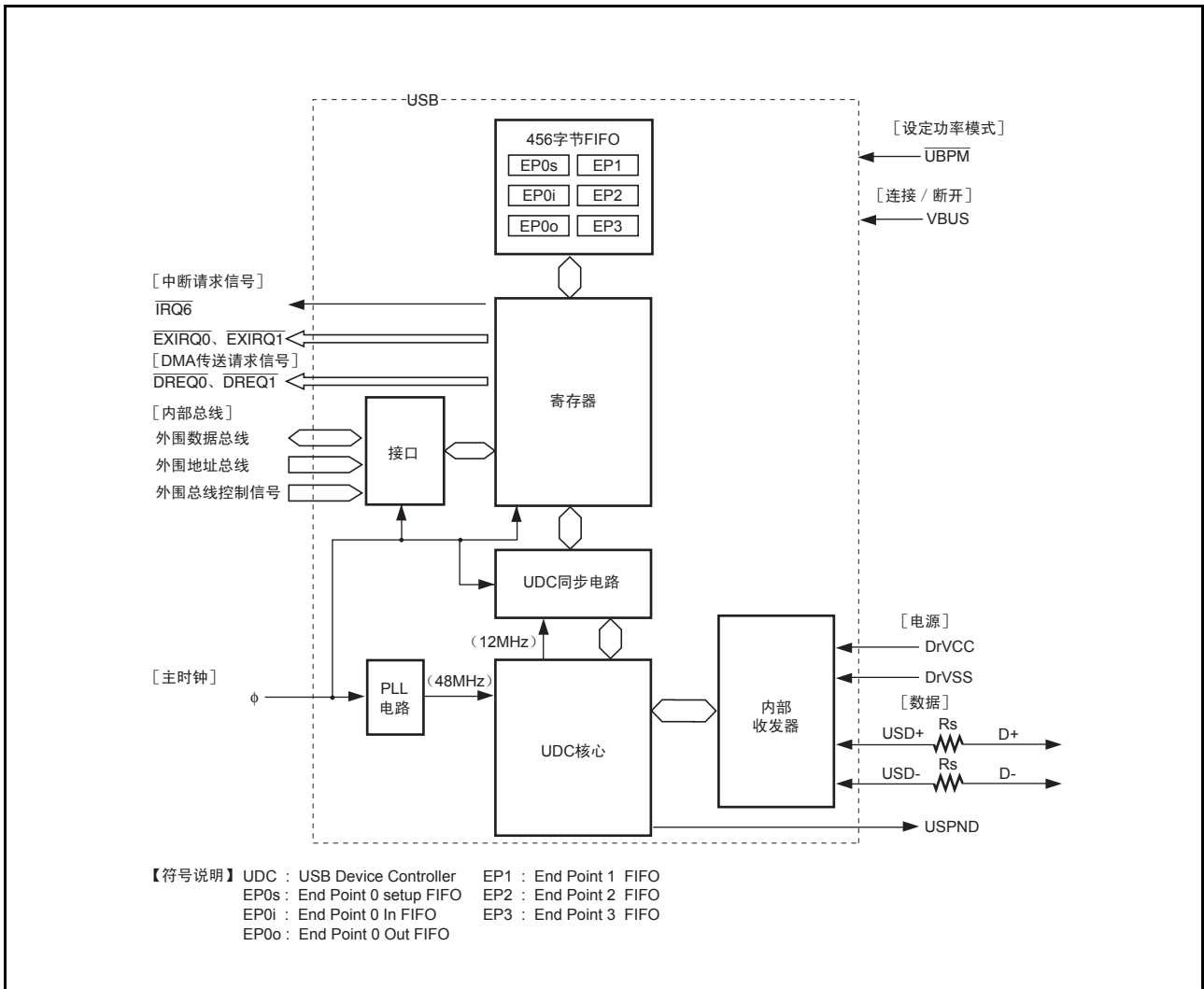


图 14.1 USB 框图

14.2 输入输出管脚

USB 的管脚结构如表 14.1 所示。

表 14.1 管脚结构

管脚名称	输入输出	功能
USD+	输入输出	USB 数据的输入输出管脚
USD-		
DrVCC	输入	USB 内置收发器电源管脚
DrVSS	输入	USB 内置收发器的接地管脚
VBUS	输入	USB 线的连接 / 断开检测信号管脚
UBPM	输入	USB 总线电源 / 自身电源模式设定管脚 在总线电源模式下使用时请固定为 0。 在自身电源模式下使用时请固定为 1。
USPND	输出	USB 挂起输出管脚 一旦转换至 suspend 状态, 会变为高电平状态。

14.3 寄存器说明

USB 中有以下寄存器。

- USB 控制寄存器 (UCTLR)
- USB DMAC 传送请求寄存器 (UDMAR)
- USB 设备恢复寄存器 (UDRR)
- USB 触发器寄存器0 (UTRG0)
- USB FIFO 清除寄存器0 (UFCLR0)
- USB 终点挂起寄存器0 (UESTL0)
- USB 终点挂起寄存器1 (UESTL1)
- USB 终点数据寄存器0s (UEDR0s) [用于接收 Setup 数据]
- USB 终点数据寄存器0i (UEDR0i) [用于发送 Control_in 数据]
- USB 终点数据寄存器0o (UEDR0o) [用于接收 Control_out 数据]
- USB 终点数据寄存器3 (UEDR3) [用于发送 Interrupt_in 数据]
- USB 终点数据寄存器1 (UEDR1) [用于发送 Bulk_in 数据]
- USB 终点数据寄存器2 (UEDR2) [用于接收 Bulk_out 数据]
- USB 终点接收数据大小寄存器0o (UESZ0o) [用于接收 Control_out 数据]
- USB 终点接收数据大小寄存器2 (UESZ2) [用于接收 Bulk_out 数据]
- USB 中断标志寄存器0 (UIFR0)
- USB 中断标志寄存器1 (UIFR1)
- USB 中断标志寄存器3 (UIFR3)
- USB 中断允许寄存器0 (UIER0)
- USB 中断允许寄存器1 (UIER1)
- USB 中断允许寄存器3 (UIER3)
- USB 中断选择寄存器0 (UISR0)
- USB 中断选择寄存器1 (UISR1)
- USB 中断选择寄存器3 (UISR3)
- USB 数据状态寄存器 (UDSR)
- USB 配置值寄存器 (UCVR)
- USB 测试寄存器0 (UTSTR0)
- USB 测试寄存器1 (UTSTR1)
- USB 测试寄存器2、A~F (UTSTR2、A~F)
- 模块停止控制寄存器B (MSTPCRB)
- 扩展模块停止寄存器 (EXMDLSTP)

14.3.1 USB 控制寄存器 (UCTLR)

本寄存器用于控制 USB 运行时钟的选择及 USB 模块内的复位。在 USB 模块停止 2 (MSTPCRB 寄存器 MSTPB0) = 1 的状态也可以读取 / 写入。UCTLR 的设定流程请参考 14.5 通信运行说明。

位	位名	初始值	R/W	说明
7	—	0	R/W	保留位 写入时请写入 0。
6	TMOWE	0	R/W	TMOW 管脚允许位 0: USPND/TMOW 管脚为 USB 的 USPND 输出 1: USPND/TMOW 管脚为 RTC 的 TMOW 输出
5 4 3 2	UCKS3 UCKS2 UCKS1 UCKS0	0 0 0 0	R/W R/W R/W R/W	USB 运行时钟选择 3 ~ 0 该位用于控制生成 USB 运行时钟 (48MHz) 的内置 PLL。UCKS3 ~ UCKS0 = 0000 时, PLL 停止, 因此必须与时钟源进行核对, 选择 USB 运行时钟。 内置 PLL 从解除 USB 模块停止 2 后才开始运行。并且, 等待 USB 运行时钟稳定后, 才向 UDC 内核提供 USB 运行时钟。USB 运行时钟稳定等待时间完成时序可通过 UIFR3 寄存器的 CK48READY 标志检测出。请在 USB 模块停止 2 (MSTPB0) 为 1 时进行 UCKS3 ~ 0 的写入。 0000: USB 运行时钟停止 (PLL 运行停止) 0001: 保留 001x: 保留 010x: 保留 0110: 使用通过 PLL 将 24MHz 主振荡提高 2 倍 (48MHz) 的时钟 0111: 使用通过 PLL 将 16MHz 主振荡提高 3 倍 (48MHz) 的时钟 1xxx: 保留 USB 运行时钟稳定等待时间全部为 2ms。 【符号说明】 x: Don't care
1	UIFRST	1	R/W	USB 接口软件复位 UIFRST 位控制 USB 模块内的复位。如果将 UIFRST 位置为 1, 除了 UCTLR、UIER3 寄存器及 UIFR3 寄存器的 CK48READY 位, USB 模块内全部被复位。在初始设定时, 解除 USB 模块停止 2, 等待 USB 运行时钟 (48MHz) 稳定时间后, 将 UIFRST 位清 0。 0: 将 USB 模块设定为运行状态 (初始设定时, 请在等待 USB 运行时钟稳定时间后, 清 0) 1: 除 UCTLR 和 UIER3 及 UIFR3 寄存器的 CK48READY 位外, 将 USB 模块内设定为复位状态。将 UIFRST 位清 0 后, 再次置位为 1 时, 请务必同时将 UDCRST 位也置为 1。

位	位名	初始值	R/W	说明
0	UDCRST	1	R/W	<p>UDC 内核软件复位</p> <p>UDCRST 位控制 USB 模块内的 UDC 内核部的复位。如果将 UDCRST 位置 1，UDC 内核被复位，USB 总线同步运行停止。初始设定时，将 UIFRST 位清 0，通过端口 (P36) 控制上拉 D+ 后，将 UDCRST 位清 0。但是，在 suspend 时，需要保持 UDC 内核的状态，因此必须保持 UDCRST 位为 0，设定 USB 模块停止 2 后，请转换至各低功耗模式。并且，检测出 VBUS 断开后请置位为 1。</p> <p>0: 将 USB 模块内的 UDC 内核设定为运行状态 (初始设定时，将 UIFRST 位清 0，通过端口控制上拉 D+ 后，请清 0。)</p> <p>1: 将 USB 模块内的 UDC 内核设定为复位状态 (suspend 时禁止置位为 1。检测出 VBUS 断开后，请置位为 1。)</p>

14.3.2 USB DMAC 传送请求寄存器 (UDMAR)

通过内置 DMAC 的 USB 请求传送数据时设定用于 Bulk 传送的 EP1、EP2 的各数据寄存器 (UEDR1、UEDR2)。请将 DREQ0 和 DREQ1 分开，进行 DMAC 传送设定。重复设定 DREQ 时，不能保证该寄存器的运行。关于 DMAC 传送的详细运行请参考 14.6 DMA 传送说明。

【注】 通过内置 DMAC 的 USB 请求传送数据时，因为不使用 DREQ0，所以请设定 UDMAR=H'00。

位	位名	初始值	R/W	说明
7~4	—	全为 0	R	保留位 读取时通常 0。写入无效。
3	EP2T1	0	R/W	EP2 DMAC 传送请求选择 1、0 00: 没有 EP2 的 DMAC 传送请求。 01: 保留。 10: 在 <u>DREQ0</u> 请求 EP2 的 DMAC 传送。 11: 在 <u>DREQ1</u> 请求 EP2 的 DMAC 传送。
2	EP2T0	0	R/W	
1	EP1T1	0	R/W	EP1 DMAC 传送请求选择 1、0 00: 没有 EP1 的 DMAC 传送请求。 01: 保留。 10: 在 <u>DREQ0</u> 请求 EP1 的 DMAC 传送。 11: 在 <u>DREQ1</u> 请求 EP1 的 DMAC 传送。
0	EP1T0	0	R/W	

14.3.3 USB 设备恢复寄存器 (UDRR)

UDRR 中包含了表示主机远程唤醒的禁止 / 允许状态的状态位和执行 USB 模块 suspend 状态的远程唤醒的位。

位	位名	初始值	R/W	说明
7~2	—	全部为 0	R	保留位 读取时通常读出 0。写入定义为无效。
1	RWUPs	0	R	远程唤醒状态 该位是表示从主机进行远程唤醒的禁止 / 允许状态的状态位。RWUPs 是状态位，不能写入。通过 Set_Feature/Clear_Feature 请求，用 Device_Remote_Wakeup，禁止从主机远程唤醒，RWUPs 位会被清 0。如果远程唤醒被允许，RWUPs 位会被置为 1。 0: 远程唤醒禁止状态。 1: 远程唤醒允许状态。
0	DVR	0	W	设备恢复 该位是可以解除 suspend 状态的（执行远程唤醒）位。 DVR 位只能写入 1。读取时通常读出 0。执行远程唤醒前，需要向 USB 模块提供时钟，因此请解除各种低功耗模式和 USB 模块停止模式。 0: 不执行任何操作。 1: 可以解除 suspend 状态。 (运行远程唤醒)

14.3.4 USB 触发寄存器 0 (UTRG0)

本寄存器是触发 EP0 ~ EP3 各终点 FIFO 的单触发寄存器。

位	位名	初始值	R/W	说明
7、6	—	全部为 0	R	保留位 读取时通常读出 0。写入定义为无效。
5	EP2RDFN	0	W	EP2 读取完成 0: 不执行任何操作。 1: 完成对 EP2 的 OUT FIFO 的数据读取后, 写入 1。EP2 的 FIFO 是 2 面结构。该触发器对当前有效的面进行读取。
4	EP1PKTE	0	W	EP1 包允许 0: 不执行任何操作。 1: 对 EP1 的 IN FIFO 授与执行发送允许的触发器。 EP1 的 FIFO 是 2 面结构。该触发器对当前有效的面进行读取。
3	EP3PKTE	0	W	EP3 包允许 0: 不执行任何操作。 1: 对 EP3 的 IN FIFO 授与执行发送允许的触发器。
2	EP0oRDFN	0	W	EP0o 读取完成 0: 不执行任何操作。 1: 对 EP0o 的 OUT FIFO 进行数据读取完成后, 写入 1。通过该触发器 EP0o 就可以接收下一个数据包。
1	EP0iPKTE	0	W	EP0i 包允许 0: 不执行任何操作。 1: 对 EP0i 的 IN FIFO 授与执行发送允许的触发器。
0	EP0sRDFN	0	W	EP0s 读取完成 0: 不执行任何操作。在写入 1 之前, 对于数据阶段的发送接收请求, 返回 NAK 信号交换。 1: 对 EP0s 的指令用 FIFO 完成数据读取后, 写入 1。接收设置指令后, 通过该触发器, EP0i、EP0o 就可以接收数据阶段的信息包。与此触发器无关, EP0s 通常可以通过盖写接收信息。

14.3.5 USB FIFO 清除寄存器 0 (UFCLR0)

该寄存器是清除 EP0 ~ EP3 的各终点的 FIFO 的单触发寄存器。给各位写入 1，就可以清除对应的 FIFO 内的数据。

如果是 IN FIFO，写入数据后，没有对 UTRG0 寄存器的 PKTE 写 1 的数据或者由于对 UTRG0 寄存器的 PKTE 写 1 而变为有效的数据，就会被清除。

如果是 OUT FIFO，还在接收中的不确定的数据和接收完成的数据（没有对 UTRG0 寄存器的 RDFN 位写 1 的数据）被清除。因此请不要进行正在发送 / 接收中的清除。对于 EP1、EP2 的 2 面结构的 FIFO，2 面同时都会被清除。请注意，在该触发器中，不会清除对应的中断标志。

位	位名	初始值	R/W	说明
7、6	—	全部为 0	R	保留位 读取时通常读出 0。写入定义为无效。
5	EP2CLR	0	W	EP2 清除 0: 不执行任何操作。 1: 清除 EP2 的 OUT FIFO。
4	EP1CLR	0	W	EP1 清除 0: 不执行任何操作。 1: 清除 EP1 的 IN FIFO。
3	EP3CLR	0	W	EP3 清除 0: 不执行任何操作。 1: 清除 EP3 的 IN FIFO。
2	EP0oCLR	0	W	EP0o 清除 0: 不执行任何操作。 1: 清除 EP0o 的 OUT FIFO。
1	EP0iCLR	0	W	EP0i 清除 0: 不执行任何操作。 1: 清除 EP0i 的 IN FIFO。
0	—	0	R	保留位 读取时通常读出 0。写入定义为无效。

14.3.6 USB 终点挂起寄存器 0 (UESTL0)

该寄存器是使 EP0 ~ EP3 的各终点挂起的寄存器。当该寄存器置位为 1，从下一次的传送开始，被设置为 1 的终点，对主机返回挂起信号交换。通过函数接收指令解码的 8 字节数据时，终点 0 的挂起位被自动解除，EPOSTL 位被清 0。并且，UIFR0 寄存器的 SetupTS 标志被置位为 1 时，对 EPOSTL 位写入 1 被忽略。详细请参考 14.5.9 挂起操作。

位	位名	初始值	R/W	说明
7、6	—	全部为 0	R	保留位 读取时通常读出 0。写入定义为无效。
5	EP2STL	0	R/W	EP2 挂起 0: 解除 EP2 的挂起状态。 1: 将 EP2 设定为挂起状态。
4	EP1STL	0	R/W	EP1 挂起 0: 解除 EP1 的挂起状态。 1: 将 EP1 设定为挂起状态。
3	EP3STL	0	R/W	EP3 挂起 0: 解除 EP3 的挂起状态。 1: 将 EP3 设定为挂起状态。
2、1	—	全部为 0	R	保留位 读取时通常读出 0。写入定义为无效。
0	EPOSTL	0	R/W	EP0 挂起 0: 解除 EP0 的挂起状态。 1: 将 EP0 设定为挂起状态。

14.3.7 USB 终点挂起寄存器 1 (UESTL1)

本寄存器是控制所有终点的挂起解除的寄存器。

位	位名	初始值	R/W	说明
7	SCME	0	R/W	挂起解除模式允许 SCME 为控制挂起解除模式的位。若将 SCME 置位为 1，置位为 1 的 EPnSTL 位在返回挂起信号交换后，自动清零。所有终点的挂起解除模式的控制是通用的。不可进行各终点的个别控制。 SCME 为 0 时，写入 EPnSTL = 1 的 EPnSTL 位不会被自动清零。为了解除 EPn 的挂起状态，必须写 EPnSTL = 0。 0: 所有终点 (EP0 ~ EP3) 的挂起解除模式无效。 1: 所有终点 (EP0 ~ EP3) 的挂起解除模式有效。
6~0	—	全部为 0	R	保留位 读取时通常读出 0。写入定义为无效。

14.3.8 USB 终点数据寄存器 0s (UEDR0s)

UEDR0s 是保存终点 0 (用于 Control_setup 传送) 的设置指令的寄存器。在设置阶段, 从主机发送的 8 字节的指令数据被保存。

读取保存数据中, 开始下一个设置阶段的数据接收时的运行, 请参考 14.8 使用注意事项。UEDR0s 是 1 字节的寄存器, 但是 1 个寄存器分配了 4 字节的地址, 因此可通过字传送 / 长字传送连续读取 2 字节 / 4 字节。

位	位名	初始值	R/W	说明
7~0	D7 ~ D0	—	R	本寄存器是保存 Control_out 传送时的设置指令的寄存器。

14.3.9 USB 终点数据寄存器 0i (UEDR0i)

UEDR0i 是终点 0 (用于 Control_in 传送) 的数据寄存器。写入向主机待发送数据。连续写入的数据数必须小于最大信息包的大小。UEDR0i 虽然是 1 字节的寄存器, 但是 1 个寄存器分配了 4 字节的地址, 因此可通过字传送 / 长字传送连续写入 2 字节 / 4 字节。

位	位名	初始值	R/W	说明
7~0	D7 ~ D0	全部为 0	W	Control_in 传送用的数据寄存器。

14.3.10 USB 终点数据寄存器 0o (UEDR0o)

UEDR0o 是终点 0 (用于 Control_out 传送) 的数据寄存器。保存来自主机的接收数据。读取时的数据数必须是连续读取 UESZ0o 寄存器指定的字节数。每次读取 1 字节 UEDR0o, UESZ0o 会被减 1。UEDR0o 虽然是 1 字节的寄存器, 但是 1 个寄存器分配了 4 字节的地址, 因此可通过字传送 / 长字传送连续读取 2 字节 / 4 字节。

位	位名	初始值	R/W	说明
7~0	D7 ~ D0	—	R	Control_out 传送用的数据寄存器

14.3.11 USB 终点数据寄存器 3 (UEDR3)

UEDR3 是终点 3 (Interrupt_in 传送用) 的数据寄存器。写入向主机待发送数据。连续写入的数据数必须小于最大信息包大小。UEDR3 虽然是 1 字节的寄存器, 但是 1 个寄存器分配了 4 字节的地址, 因此通过字传送 / 长字传送就可以连续写入 2 字节 / 4 字节。

位	位名	初始值	R/W	说明
7~0	D7 ~ D0	全部为 0	W	Interrupt_in 传送用的数据寄存器

14.3.12 USB 终点数据寄存器 1 (UEDR1)

UEDR1 是终点 1 (Bulk_in 传送用) 的数据寄存器。写入向主机待发送数据。连续写入的数据数必须小于最大信息包的大小。UEDR1 虽然是 1 字节的寄存器，但是 1 个寄存器分配了 4 字节的地址，因此可通过字传送 / 长字传送连续写入 2 字节 / 4 字节。

位	位名	初始值	R/W	说明
7~0	D7 ~ D0	全部为 0	W	Bulk_in 传送用的数据寄存器

14.3.13 USB 终点数据寄存器 2 (UEDR2)

UEDR2 是终点 2 (Bulk_out 传送用) 的数据寄存器。保存来自主机的接收数据。读取时的数据数必须为 UESZ2 寄存器指定的字节数的连续读取。每读取 1 字节 UEDR2，UESZ2 会被减 1。UEDR2 虽然是 1 字节的寄存器，但是 1 个寄存器分配了 4 字节的地址，因此可通过字传送 / 长字传送连续读取 2 字节 / 4 字节。

位	位名	初始值	R/W	说明
7~0	D7 ~ D0	—	R	Bulk_out 传送用的数据寄存器

14.3.14 USB 终点接收数据大小寄存器 0o (UESZ0o)

UESZ0o 是终点 0 (Control_out 传送用) 的接收数据大小寄存器。表示从主机接收的数据的字节数。每读取 1 字节 UEDR0o，会被减 1，请注意。

位	位名	初始值	R/W	说明
7	—	—	R	保留位
6~0	D6 ~ D0	—	R	Control_out 传送数据的接收大小寄存器

14.3.15 USB 终点接收数据长度寄存器 2 (UESZ2)

UESZ2 是终点 2 (Bulk_out 传送用) 的接收数据大小寄存器。表示从主机接收的数据字节数。每读取 1 字节 UEDR2，会被减 1，请注意。

终点 2 的 Bulk_out 传送用的 FIFO 是 2 面结构。寄存器中表示的接收数据大小是当前可读取的面对应的大小。

位	位名	初始值	R/W	说明
7	—	—	R	保留位
6~0	D6 ~ D0	—	R	Bulk_out 传送数据的接收大小寄存器

14.3.16 USB 中断标志寄存器 0 (UIFR0)

该寄存器是接收设置指令、接收发送 EP0、EP3、总线复位的中断标志寄存器。

如果各标志置位为 1，会输出对应的 EXIRQ0、EXIRQ1 中断请求。写入 0，进行清除标志。写入 1 无效，因此不执行任何操作。仅仅清除某一标志时，须把适当位作为 0、其他位作为 1 写入（仅清除位 5 时，写入 H'DF）。位清除指令为读取 / 修改 / 写入指令，在读取与写入之间，产生新标志时，可能会发生误清除。因此，进行本中断标志寄存器的清除时，请勿使用位清除指令。

位	位名	初始值	R/W	说明
7	BRST	0	R/(W)*	总线复位 在 USB 总线上检测出总线复位信号时，改为被置 1。对应的中断输出是 EXIRQ0 或 EXIRQ1。 连接 USB 线时，不上拉 D+ 时，会变为 BRST = 1，请注意。
6	—	0	R	保留位 读取时通常读出 0。写入定义为无效。
5	EP3TR	0	R/(W)*	EP3 传送请求 从主机向 EP3 发送 IN 权标，FIFO 为空时该位会被置 1。对应的中断输出是 EXIRQ0 或 EXIRQ1。
4	EP3TS	0	R/(W)*	EP3 发送完成 在 EP3 写入向主机发送的数据后，正常向主机发送数据，ACK 信号交换返回时，该位被置位为 1。对应的中断输出是 EXIRQ0 或 EXIRQ1。
3	EP0oTS	0	R/(W)*	EP0o 接收完成 在 EP0o 正常接收来自主机的数据，ACK 交换信号返回到主机时，该位被置位为 1。对应的中断输出是 EXIRQ0 或 EXIRQ1。
2	EP0iTR	0	R/(W)*	EP0i 传送请求 从主机向 EP0i 发送 IN 权标，FIFO 为空时 1 该位会被置为 1。对应的中断输出是 EXIRQ0 或 EXIRQ1。
1	EP0iTS	0	R/(W)*	EP0i 发送完成 在 EP0i 写入向主机发送的数据后，数据被正常发送到主机，ACK 信号交换返回时该位被置 1。对应的中断输出是 EXIRQ0 或 EXIRQ1。
0	SetupTS	0	R/(W)*	设置指令接收完成 在 EP0s 正常接收用函数进行指令解码的 8 字节数据，ACK 信号交换返回到主机时该位被置位为 1。对应的中断输出是 EXIRQ0 或 EXIRQ1。

【注】 * 要清除标志，只能写入 0。

14.3.17 USB 中断标志寄存器 1 (UIFR1)

该寄存器是表示 EP1、EP2 状态的中断标志寄存器。

如果各标志被置 1，就会输出对应的 EXIRQ0、EXIRQ1 中断请求。

EP1TR 的标志清除，通过写入 0 来进行。写入 1 无效，不执行任何操作。仅仅清除某一标志时，须把适当位作为 0、其他位作为 1 写入 (仅清除位 1 时，写入 H'FD)。位清除指令为读取 / 修改 / 写入指令，在读取与写入之间，产生新标志时，可能会发生误清除。因此，进行本中断标志寄存器的清除时，请勿使用位清除指令。

EP1EMPTY、EP2READY、EP1ALLEMPTY 的各状态是表示 EP1、EP2 的 FIFO 状态的状态位，因此不能清除。

位	位名	初始值	R/W	说明
7~4	—	全部为 0	R	保留位 读取时通常读出 0。写入定义为无效。
3	EP1ALL EMPTYs	1	R	EP1 FIFO 全空状态 EP1 的 FIFO 是 2 面结构。此时，2 面都为空时置位为 1。(相当于 UDSR/ EP1DE 的负极性信号) 不能在 EP1ALLEMPTYs 请求中断。
2	EP2READY	0	R	EP2 数据就绪 EP2 的 FIFO 是 2 面结构。此时，至少在 1 面的 FIFO 内保存有效数据时， 该位会被置为 1。2 面都为空时，被清 0。由于该位是状态位，不能清除。 对应的中断输出是 <u>EXIRQ0</u> 或 <u>EXIRQ1</u>
1	EP1TR	0	R/(W)*	EP1 传送请求 从主机向 EP1 发送 IN 权标，2 面 FIFO 同时都为空时，该位被置 1。对应 的中断输出是 <u>EXIRQ0</u> 或 <u>EXIRQ1</u>
0	EP1EMPTY	1	R	EP1 FIFO 空 EP1 的 FIFO 是 2 面结构。此时，至少 1 面以上 FIFO 为空时该位被置 1。 2 面同时 FULL 时，会被清 0。由于该位是状态位，不能清除。对应的中断 输出是 <u>EXIRQ0</u> 或 <u>EXIRQ1</u>

【注】 * 要清除标志只能写入 0。

14.3.18 USB 中断标志寄存器 3 (UIFR3)

本寄存器是表示 USB 状态的中断标志寄存器。如果各标志被置位为 1，会输出对应的 $\overline{\text{EXIRQ0}}$ 、 $\overline{\text{EXIRQ1}}$ 、 $\overline{\text{IRQ6}}$ 中断请求。写入 0 来清除 VBUSi 、 SPRSi 、 SETC 、 SOF 、 CK48READY 的各标志。写入 1 无效。仅仅清除某一标志时，须把适当位作为 0、其他位作为 1 写入（仅清除位 5 时，写入 H'DF）。位清除指令为读取 / 修改 / 写入指令，在读取与写入之间，产生新标志时，可能会发生误清除。因此，进行本中断标志寄存器的清除时，请勿使用位清除指令。由于 VBUSs 、 SPRSs 是状态位，不能清除。

位	位名	初始值	R/W	说明
7	CK48READY	0	R(W)*	USB 运行时钟 (48MHz) 稳定检测 自动计算 USB 模块停止模式解除后的 USB 运行时钟 (48MHz) 稳定时间后，该位被置 1。 对应的中断输出是 $\overline{\text{EXIRQ0}}$ 或 $\overline{\text{EXIRQ1}}$ 。 CK48READY 在 USB 接口软件复位状态 (UCTLR 寄存器 UIFRST = 1) 也可以运行。详细请参考 14.3.1 USB 控制寄存器 (UCTLR) 的 UCKS3 ~ UCKS0 位。
6	SOF	0	R(W)*	Start of Frame 信息包检测 检测出 Start of Frame (SOF) 包时，该位会被置 1。 对应的中断输出是 $\overline{\text{EXIRQ0}}$ 或 $\overline{\text{EXIRQ1}}$ 。
5	SETC	0	R(W)*	Set_Configuration 指令检测 检测出 Set_Configuration 指令时，该位被置 1。 对应的中断输出是 $\overline{\text{EXIRQ0}}$ 或 $\overline{\text{EXIRQ1}}$ 。
4	—	0	R	保留位 读取时通常读出 0。写入定义为无效。
3	SPRSs	0	R	挂起 / 恢复状态 表示挂起 / 恢复状态。 0: 表示总线状态为正常状态。 1: 表示总线状态为挂起状态。
2	SPRSi	0	R(W)*	挂起 / 恢复中断 从正常状态向挂起状态转换，或者从挂起状态向正常状态转换时，该位被置 1。对应的中断输出是 $\overline{\text{IRQ6}}$ 。可用于恢复各种低功耗模式状态的解除。
1	VBUSs	0	R	VBUS 状态 该位是通过 USB 线的连接 / 断开表示 VBUS 状态的状态位。不能请求在 VBUSs 的中断。 0: 表示 VBUS (USB 线) 断开状态。 1: 表示 VBUS (USB 线) 连接状态。
0	VBUSi	0	R(W)*	VBUS 中断 通过 USB 线的连接 / 断开，改变 VBUS 状态时，该位被置 1。对应的中断输出是 $\overline{\text{EXIRQ0}}$ 或 $\overline{\text{EXIRQ1}}$ 。

【注】 * 要清除标志只能写入 0。

14.3.19 USB 中断允许寄存器 0 (UIER0)

该寄存器是允许中断标志寄存器 0 (UIFR0) 的各中断的寄存器。

UIER0 的各位被置 1 时，如果对应的中断标志被置 1，对应的 $\overline{\text{EXIRQ0}}$ 、 $\overline{\text{EXIRQ01}}$ 被断言，就会请求中断。请通过中断选择寄存器 0 (UISR0) 进行 $\overline{\text{EXIRQ0}}$ 、 $\overline{\text{EXIRQ01}}$ 的选择。

位	位名	初始值	R/W	说明
7	BRSTE	0	R/W	BRST 中断允许。
6	—	0	R	保留位 读取时通常读出 0。
5	EP3TRE	0	R/W	EP3TR 中断允许。
4	EP3TSE	0	R/W	EP3TS 中断允许。
3	EP0oTSE	0	R/W	EP0oTS 中断允许。
2	EP0iTRE	0	R/W	EP0iTR 中断允许。
1	EP0iTSE	0	R/W	EP0iTS 中断允许。
0	SetupTSE	0	R/W	SetupTS 中断允许。

14.3.20 USB 中断允许寄存器 1 (UIER1)

本寄存器是允许中断标志寄存器 1 (UIFR1) 的各中断的寄存器。

UIER1 的各位被置位为 1 时，如果对应的中断标志被置 1，对应的 $\overline{\text{EXIRQ0}}$ 、 $\overline{\text{EXIRQ01}}$ 被断言，就会请求中断。请通过中断选择寄存器 1 (UISR1) 进行 $\overline{\text{EXIRQ0}}$ 、 $\overline{\text{EXIRQ01}}$ 的选择。

位	位名	初始值	R/W	说明
7~4	—	全部为 0	R	保留位 读取时通常读出 0。
3	—	0	R/W	保留位 写入时请写入 0。
2	EP2READYE	0	R/W	EP2READY 中断允许。
1	EP1TRE	0	R/W	EP1TR 中断允许。
0	EP1EMPTYE	0	R/W	EP1EMPTY 中断允许。

14.3.21 USB 中断允许寄存器 3 (UIER3)

本寄存器是允许中断标志寄存器 3 (UIFR3) 各中断的寄存器。

即使在 USB 模块停止 2 (MSTPCRB 寄存器 MSTPB0) = 1 的状态, 也可进行读 / 写操作。

UIER3 的各位被置为 1 时, 如果对应的中断标志置 1, 对应的 $\overline{\text{EXIRQ0}}$ 、 $\overline{\text{EXIRQ01}}$ 被断言, 就会请求中断。通过中断选择寄存器 3 (UISR3) 进行 $\overline{\text{EXIRQ0}}$ 、 $\overline{\text{EXIRQ01}}$ 的选择。但是, SPRSiE 位是 $\overline{\text{IRQ6}}$ 专用的中断允许位, 因此不能通过 UISR3 寄存器选择。

位	位名	初始值	R/W	说明
7	CK48READYE	1	R/W	CK48READY 中断允许。
6	SOFE	0	R/W	SOF 中断允许。
5	SETCE	0	R/W	SETC 中断允许。
4	—	0	R	保留位
3	—	0	R	读取时通常读出 0。
2	SPRSiE	0	R/W	SPRSi 中断允许 ($\overline{\text{IRQ6}}$ 专用)。
1	—	0	R	保留位 读取时通常读出 0。
0	VBUSiE	0	R/W	VBUSi 中断允许。

14.3.22 USB 中断选择寄存器 0 (UISR0)

本寄存器是用于设定输出中断标志寄存器 0 (UIFR0) 各中断请求的 $\overline{\text{EXIRQ}}$ 的寄存器。如果将 UISR0 寄存器对应的位清 0, 会从 $\overline{\text{EXIRQ0}}$ 输出中断请求, 如果置位为 1, 会从 $\overline{\text{EXIRQ1}}$ 输出中断请求。

位	位名	初始值	R/W	说明
7	BRSTS	0	R/W	BRST 中断允许。
6	—	0	R	保留位 读取时通常读出 0。
5	EP3TRS	0	R/W	EP3TR 中断允许。
4	EP3TSS	0	R/W	EP3TS 中断允许。
3	EP0oTSS	0	R/W	EP0oTS 中断允许。
2	EP0iTRS	0	R/W	EP0iTR 中断允许。
1	EP0iTSS	0	R/W	EP0ITS 中断允许。
0	SetupTSS	0	R/W	SetupTS 中断允许。

14.3.23 USB 中断选择寄存器 1 (UISR1)

本寄存器用于设定输出中断标志寄存器 1 (UIFR1) 各中断请求的 $\overline{\text{EXIRQ}}$ 的寄存器。如果将对应 UISR1 寄存器对应的位清 0, 会从 $\overline{\text{EXIRQ0}}$ 输出中断请求, 如果置位为 1, 会从 $\overline{\text{EXIRQ1}}$ 输出中断请求。

位	位名	初始值	R/W	说明
7~4	—	全部为 0	R	保留位 读取时通常读出 0。
3	—	0	R/W	保留位 写入时请写 0。
2	EP2READYS	0	R/W	EP2READY 中断选择。
1	EP1TRS	0	R/W	EP1TR 中断选择。
0	EP1EMPTY	0	R/W	EP1EMPTY 中断选择。

14.3.24 USB 中断选择寄存器 3 (UISR3)

本寄存器是用于设定输出中断标志寄存器 3 (UIFR3) 各中断请求的 $\overline{\text{EXIRQ}}$ 的寄存器。如果将 UISR3 寄存器对应的位清 0, 会从 $\overline{\text{EXIRQ0}}$ 输出中断请求, 如果置位为 1, 会从 $\overline{\text{EXIRQ1}}$ 输出中断请求。

位	位名	初始值	R/W	说明
7	CK48READYS	0	R/W	CK48READY 中断选择。
6	SOFS	0	R/W	SOF 中断选择。
5	SETCS	0	R/W	SETC 中断选择。
4~1	—	全为 0	R	保留位 读取时通常读出 0。
0	VBUSiS	0	R/W	VBUSi 中断选择。

14.3.25 USB 数据状态寄存器 (UDSR)

本寄存器是表示 IN FIFO 数据寄存器 (EP0i、EP1、EP3) 内有无有效数据的状态寄存器。

写入 IN FIFO 的数据, 因 UTRG0 寄存器的 PKTE 位写 1 而变为有效时, 会被置位为 1。并且, 将有效数据全部传送到主机时, 会被清 0。如果是 EP1 的 2 面结构时, 2 面同时没有数据时显示 0。

位	位名	初始值	R/W	说明
7~3	—	全部为 0	R	保留位 读取时通常读出 0。写入定义为无效。
2	EP1DE	0	R	EP1 数据允许 0: EP1 中没有有效数据时。 1: EP1 中有有效数据时。 相当于 UIFR1/EP1ALLEMTYs 的负极性信号。
1	EP3DE	0	R	EP3 数据允许 0: EP3 中没有有效数据时。 1: EP3 中有有效数据时。
0	EP0iDE	0	R	EP0i 数据允许 0: EP0i 中没有有效数据时。 1: EP0i 中有有效数据时。

14.3.26 USB 配置值寄存器 (UCVR)

本寄存器是保存从主机接收 Set_Configuration 指令时的配置值的寄存器。

位	位名	初始值	R/W	说明
7、6	—	全部为 0	R	保留位 读取时通常读出 0。写入定义为无效。
5	CNFV0	0	R	配置值 0 该位保存接收 Set_Configuration 指令时的配置值。CNFV0 位的更新时序是 UIFR3 寄存器的 SETC = 1 设置时。
4~0	—	全部为 0	R	保留位 读取时通常读出 0。写入定义为无效。

14.3.27 USB 测试寄存器 0 (UTSTR0)

本寄存器是可控制内置收发器输出信号的测试寄存器。设定 UCTLR/UIFRST = 0、UDCRST = 0 后，通过设定 PTSTE = 1，就可以任意设定收发器输出信号 (USD+、USD-)。UTSTR0 设定和管脚输出值的关系如表 14.2 所示。

位	位名	初始值	R/W	说明
7	PTSTE	0	R/W	管脚测试允许 该位是使内置收发器输出管脚 (USD+、USD-) 和 USPND 管脚的测试控制有效的位。
6~4	—	全部为 0	R	保留位 读取时通常读出 0。写入定义为无效。
3	SUSPEND	0	R/W	内置收发器输出信号设定位 SUSPEND：设定内置收发器的 USPND 管脚信号。 \overline{OE} ：设定内置收发器的输出允许 (\overline{OE}) 信号。 FSE0：设定内置收发器的 Single-ended 0 (FSE0) 信号。 VPO：设定内置收发器的 USD+ (VPO) 信号。
2	\overline{OE}	1	R/W	
1	FSE0	0	R/W	
0	VPO	0	R/W	

表 14.2 UTSTR0 设定和管脚输出值的关系

寄存器设定值			管脚输出值	管脚输入					管脚输出值	
UCTLR/ TMOWE	PTSTE	SUSPEND	USPND/ TMOW	VBUS	PTSTE	OE	FSE0	VPO	USD+	USD-
1	×	×	—	0	×	×	×	×	Hi-Z	Hi-Z
0	0	×	—	1	0	×	×	×	—	—
0	1	0	0	1	1	0	0	0	0	1
0	1	1	1	1	1	0	0	1	1	0
				1	1	0	1	×	0	0
				1	1	1	×	×	Hi-Z	Hi-Z

【符号说明】

×：Don't care

—：不可控制。通过当时的 USB 运行状态和端口设定，显示正常运行的管脚状态。

14.3.28 USB 测试寄存器 1 (UTSTR1)

本寄存器是 USB 控制管脚和内置收发器输入信号为可监视的测试寄存器。管脚输入值和 UTSTR1 监视器的关系如表 14.3 所示。

位	位名	初始值	R/W	说明
7	$\overline{\text{VBUS}}$	— *	R	内置收发器输入信号监视位 VBUS: VBUS 管脚监视位。 $\overline{\text{UBPM}}$: $\overline{\text{UBPM}}$ 管脚监视位。
6	$\overline{\text{UBPM}}$	— *	R	
5~3	—	全部为 0	R	保留位 读取时通常读出 0。写入定义为无效。
2	RCV	— *	R	内置收发器输入信号监视位 RCV: 内置收发器的差动输入级别 (RCV) 信号监视位 VP: 内置收发器的 USD+ (VP) 信号监视位 VM: 内置收发器的 USD- (VM) 信号监视位
1	VP	— *	R	
0	VM	— *	R	

【注】* 由 VBUS、 $\overline{\text{UBPM}}$ 、USB+、USB- 管脚的状态决定。

表 14.3 管脚输入值和 UTSTR1 监视器的关系

管脚输入值		UTSTR1 监视器值		寄存器设定值		管脚输入值			UTSTR1 监视器值		
VBUS	$\overline{\text{UBPM}}$	VBUS	$\overline{\text{UBPM}}$	UTSTR0/ PTSTE	UTSTR0/ SUSPEND	VBUS	USD+	USD-	RCV	VP	VM
0/1	×	0/1	×	×	×	0	×	×	0	0	0
×	0/1	×	0/1	0	×	1	0	0	×	0	0
				0	×	1	0	1	0	0	1
				0	×	1	1	0	1	1	0
				0	×	1	1	1	×	1	1
				1	0	1	0	0	×	0	0
				1	0	1	0	1	0	0	1
				1	0	1	1	0	1	1	0
				1	0	1	1	1	×	1	1
				1	1	1	0	0	0	0	0
				1	1	1	0	1	0	0	1
				1	1	1	1	0	0	1	0
				1	1	1	1	1	0	1	1

【符号说明】

× : Don't care

0/1: 管脚输入值 = UTSTR1 监视器值的组合

14.3.29 USB 测试寄存器 2、A ~ F (UTSTR2、UTSTRA ~ UTSTRF)

本寄存器为测试寄存器。禁止写入。

14.3.30 模块暂停控制寄存器 B (MSTPCRB)

位	位名	初始值	R/W	说明
7	MSTPB7	1	R/W	模块停止位 详细请参考 20.1.3 模块停止控制寄存器 A ~ C (MSTPCRA ~ MSTPCRC)。
6	MSTPB6	1	R/W	
5	MSTPB5	1	R/W	
4	MSTPB4	1	R/W	
3	MSTPB3	1	R/W	
2	MSTPB2	1	R/W	
1	MSTPB1	1	R/W	
0	MSTPB0	1	R/W	USB 模块停止 2 0: 完全解除 USB 模块停止 向 USB 模块提供时钟。 请务必在本位清 0 前, 将 EXMDLSTP 寄存器的 USBSTOP1 清 0: 清除本位后, 内置 PLL 电路开始运行。等待 USB 运行时钟的稳定时间后, (设定 UIFR3 寄存器 CK48READY = 1), 再访问 USB 模块内寄存器。 1: USB 模块停止 停止内置 PLL 电路及大部分 USB 模块内的时钟。但是, 可保持 USB 模块内寄存器值。

14.3.31 扩展模块待机寄存器 (EXMDLSTP)

位	位名	初始值	R/W	说明
7~2	—	不定	—	保留位 读取不定。写入定义为无效。
1	RTCSTOP	0	R/W	RTC 模块停止 0: RTC 模块停止解除 1: RTC 模块停止
0	USBSTOP1	0	R/W	USB 模块停止 1 0: USB 模块停止一部分解除 向 USB 模块的一部分提供时钟。 清除本位后, 仅能访问 USB 模块内的 UCTLR 和 UIER3 寄存器。 为了要访问其他寄存器, 请将 MSTPCRB 寄存器的 MSTPB0 清 0。 1: USB 模块完全停止 USB 模块内的时钟完全停止。 但是, 仍保持 USB 模块内寄存器值。

- 【注】
1. USB 模块停止模式解除流程请参考 14.5 通信运行说明。
 2. 通过端口 D 寄存器 (PORTD) 读取管脚状态时, 在访问 EXMDLSTP 后, 虚读一次 H'FFFF40 ~ H'FFFF5F 范围外的外部地址空间 (H'FFEFC0 ~ H'FFF7FF), 再读取 PORTD。

14.4 中断源

本模块有 3 个中断信号。各中断源和请求信号的对照如表 14.4 所示。 $\overline{\text{EXIRQ}}$ 中断在低电平被触发。 $\overline{\text{EXIRQ}}$ 中断检测只能以电平检测。关于挂起 / 恢复中断 $\overline{\text{IRQ6}}$ ，请通过中断控制器的寄存器设定设置为下降沿检测。

表 14.4 中断信号一览

寄存器	位	传送模式	中断源	说明	中断请求信号	通过 USB 请求, 启动 DMAC 启动 *5
UIFR0	0	Control 传送 (EP0)	SetupTS*1	Setup 指令接收完成	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	1		EP0iTS*1	EP0i 发送完成	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	2		EP0iTR*1	EP0i 发送请求	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	3		EP0oTS*1	EP0o 接收完成	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	4	Interrupt_in 传送 (EP3)	EP3TS	EP3 发送完成	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	5		EP3TR	EP3 发送请求	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	6	—	保留	—	—	—
	7	(Status)	BRST	总线复位	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
UIFR1	0	Bulk_in 传送 (EP1)	EP1EMPTY	EP1 FIFO 空	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	$\overline{\text{DREQ0}}$ or $\overline{\text{DREQ1}}$ *2
	1		EP1TR	EP1 发送请求	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	2	Bulk_out 传送 (EP2)	EP2READY	EP2 数据就绪	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	$\overline{\text{DREQ0}}$ or $\overline{\text{DREQ1}}$ *3
	3	Bulk_in 传送 (EP1)	(EP1ALLEMP TYs)	EP1 FIFO 全空状态	×	×
	4	—	保留	—	—	—
	5					
	6					
7						

寄存器	位	传送模式	中断源	说明	中断请求信号	通过 USB 请求, 启动 DMAC 启动 *5
UIFR3	0	— (Status)	VBUSi	VBUS 中断	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	1		(VBUSs)	VBUS 状态	×	×
	2		SPRSi	挂起 / 恢复中断	$\overline{\text{IRQ6}}^{*4}$	×
	3		(SPRSs)	挂起 / 恢复状态	×	×
	4		保留	—	—	—
	5		SETC	Set_Configuration 检测	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	6		SOF	Start of Frame 包检测	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×
	7		CK48READY	USB 运行时钟稳定检测	$\overline{\text{EXIRQ0}}$ or $\overline{\text{EXIRQ1}}$	×

【注】 *1 EPO 相关中断源请分配到相同的中断请求信号。

*2 EP1 的 USB 请求的 DMA 传送请求通过 UDMAR 寄存器的 EP1T1、EP1T0 位设定。

*3 EP2 的 USB 请求的 DMA 传送请求通过 UDMAR 寄存器的 EP2T1、EP2T0 位设定。

*4 请将挂起 / 恢复中断请求 $\overline{\text{IRQ6}}$ 设定为下降沿检测 (ISCRH 寄存器 IRQ6SCB、IRQ6SCA = 01)。

*5 自动请求时, 不使用 DREQ 信号. 所有的标志或中断均可通过 CPU 启动 DMAC。

- $\overline{\text{EXIRQ0}}$ 信号

本信号是对于由于中断选择寄存器 0~3 (UISR0~3) 被清 0 的中断源的中断请求信号。分配到 $\overline{\text{EXIRQ0}}$ 信号的中断源中, 即使是 1 个, 也会在相应中断标志寄存器的位被置 1 时断言。

- $\overline{\text{EXIRQ1}}$ 信号

本信号是对于由于中断选择寄存器 0~3 (UISR0~3) 而被置 1 的中断源的中断请求信号。分配到 $\overline{\text{EXIRQ1}}$ 信号的中断源中, 即使是 1 个, 也会在相应中断标志寄存器的位被置 1 时断言。

- $\overline{\text{IRQ6}}$ 信号

本信号是挂起 / 恢复中断源专用的信号。从挂起状态转换或者从恢复状态转换时, 会输出下降沿。

14.5 通信运行说明

14.5.1 初始设定

请按照以下流程进行初始设定处理。

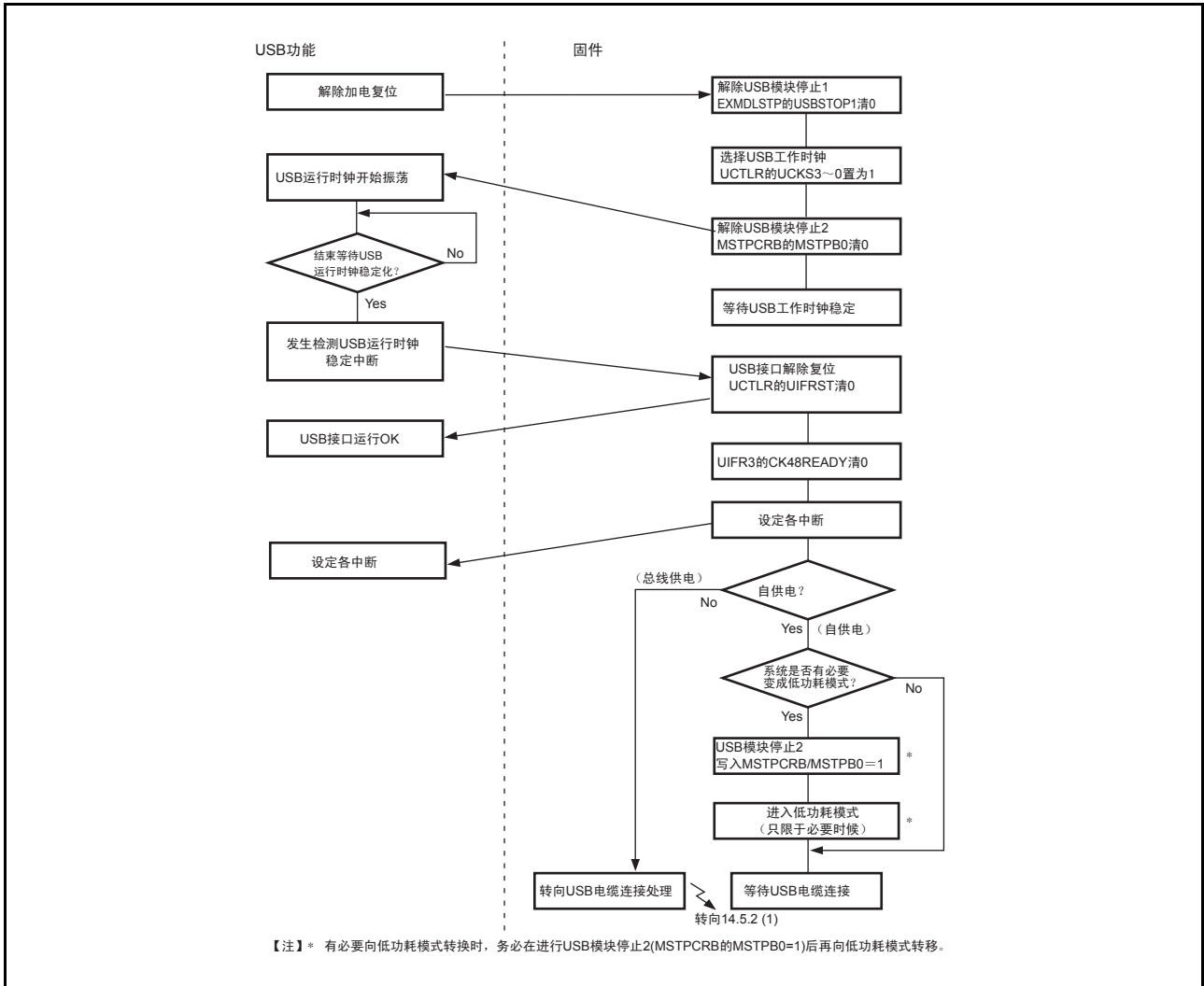


图 14.2 初始设定的运行

14.5.2 USB 线的连接 / 断开

14.5.2.1 USB 线连接时 (没有使用 USB 模块停止、各种低功耗模式时)

没有使用 USB 模块停止或者各种低功耗模式的应用程序 (自身电源) 时, USB 线从断开状态到连接状态, 请按照以下流程进行处理:

并且, 在总线电源功能下, 请按照以下【注】*2 进行处理。

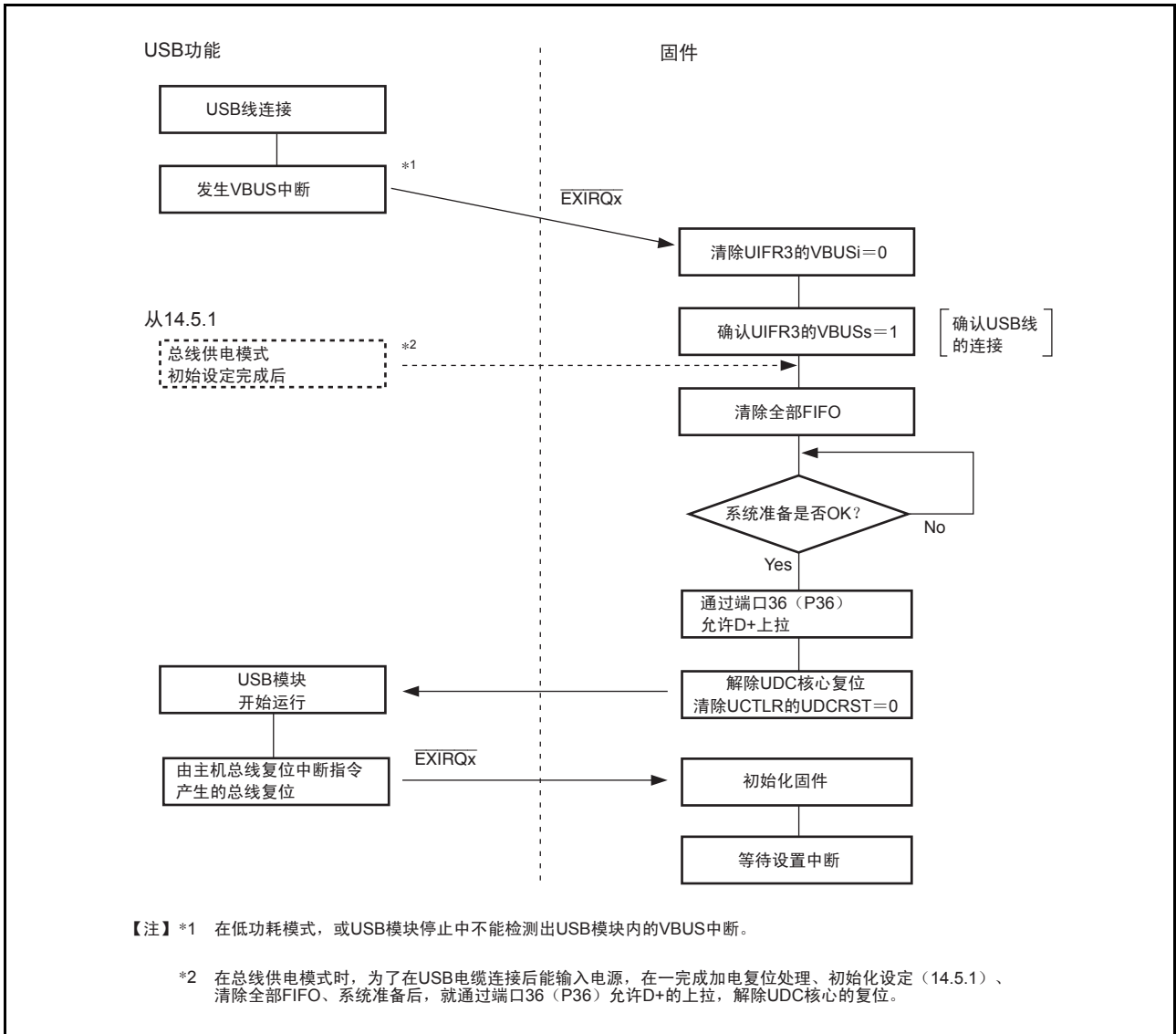


图 14.3 USB 线连接时的运行 (没有使用 USB 模块待机、各种低功耗模式时)

14.5.2.2 USB 线连接时 (使用 USB 模块停止、各种低功耗模式时)

在使用 USB 模块停止或者各种低功耗模式的应用程序 (自身电源) 时, 请按照以下流程进行 USB 线从断开状态到连接状态的处理。

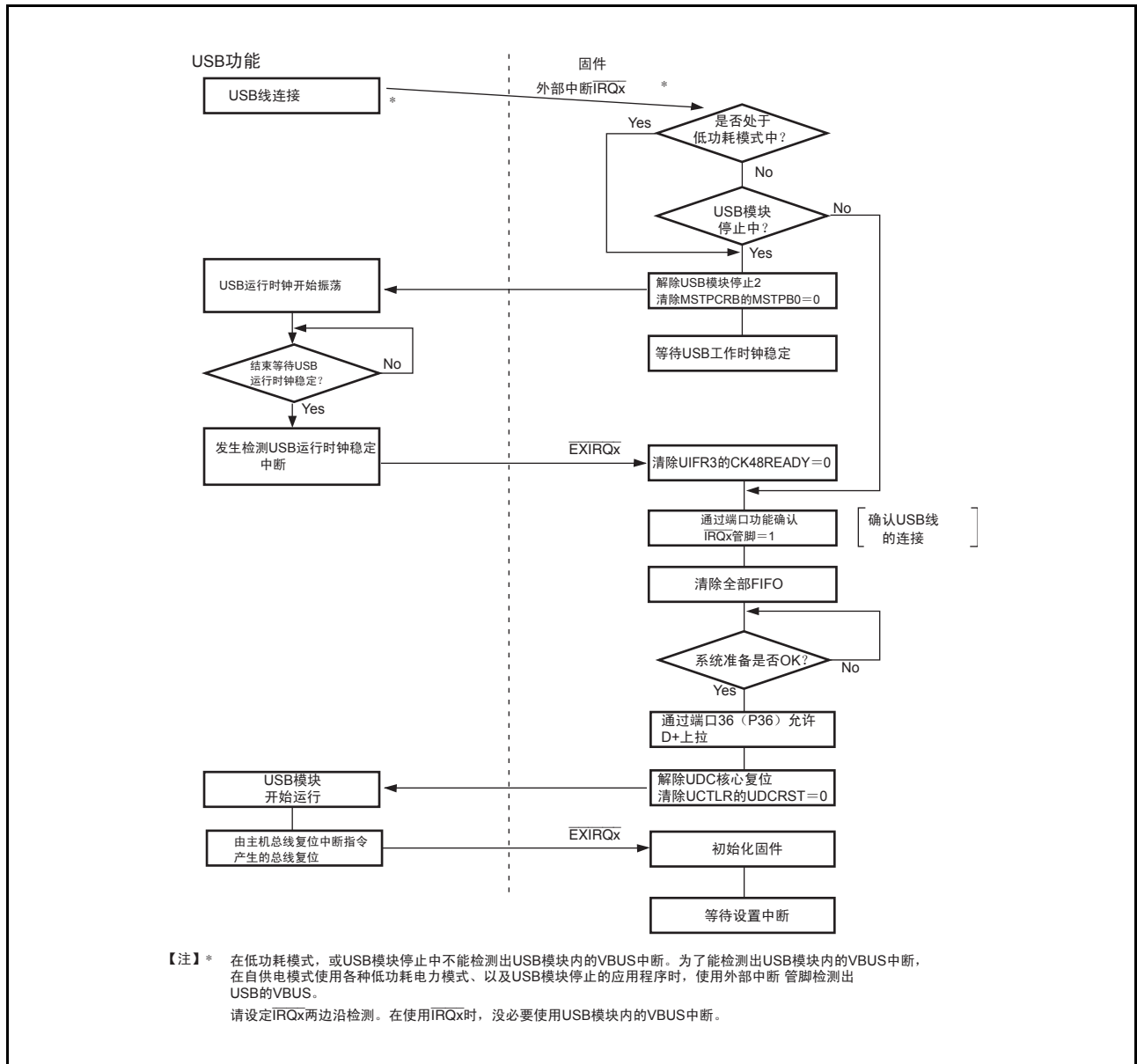


图 14.4 USB 线连接时的运行 (使用 USB 模块待机、各种低功耗模式时)

14.5.2.3 USB 线断开时 (没有使用 USB 模块停止、各种低功耗模式时)

在没有使用 USB 模块停止或者各种低功耗模式的应用程序 (自身电源) 时, USB 线从连接状态到断开状态处理, 请按照以下流程进行。

同时, 在总线电源功能下, 断开 USB 线, 电源会 OFF, 因此不需要以下处理。

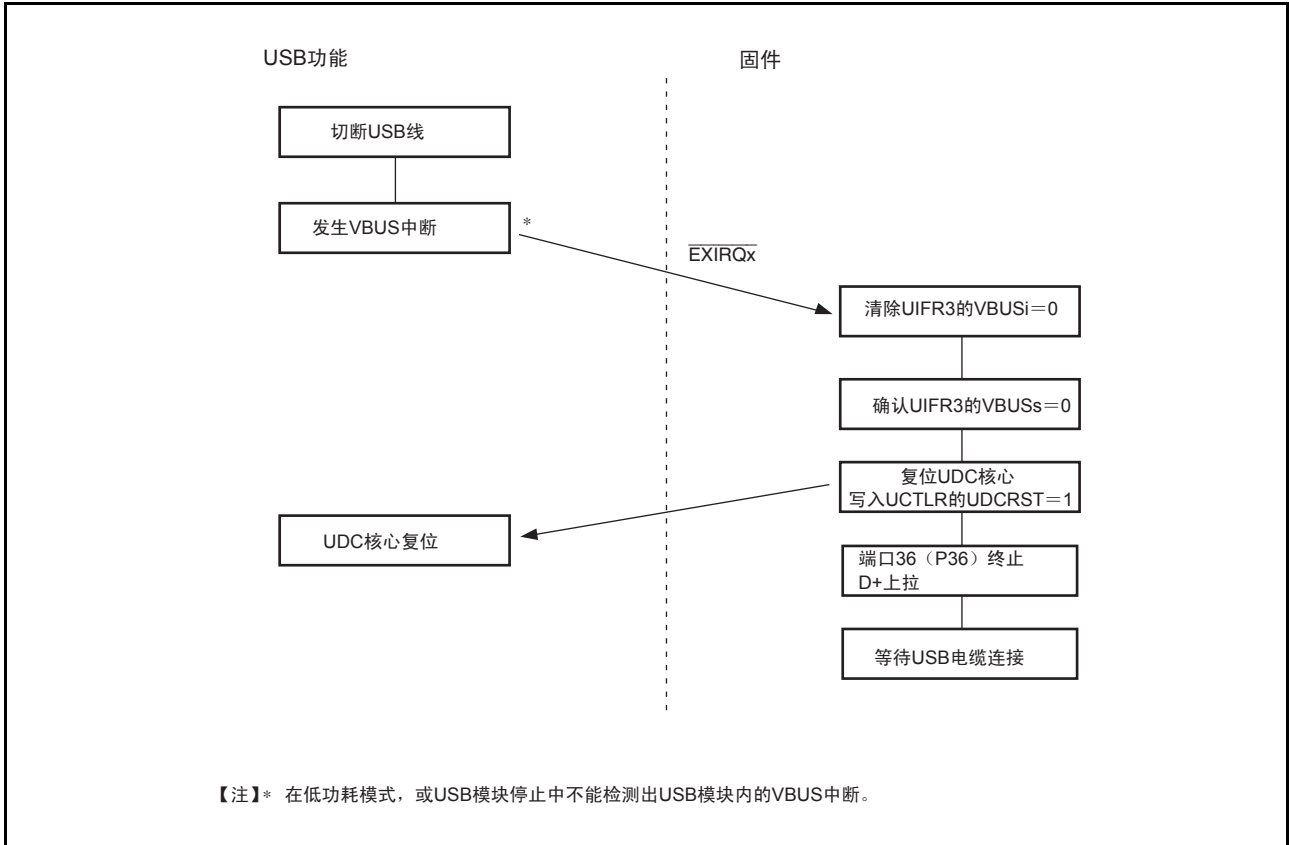


图 14.5 USB 线断开时的运行 (没有使用 USB 模块待机、各种低功耗模式时)

14.5.2.4 USB 线断开时 (使用 USB 模块停止、各种低功耗模式时)

在使用 USB 模块待机或者各种低功耗模式的应用程序 (自身电源) 时, USB 线从连接状态到断开状态的处理, 请按照以下流程进行:

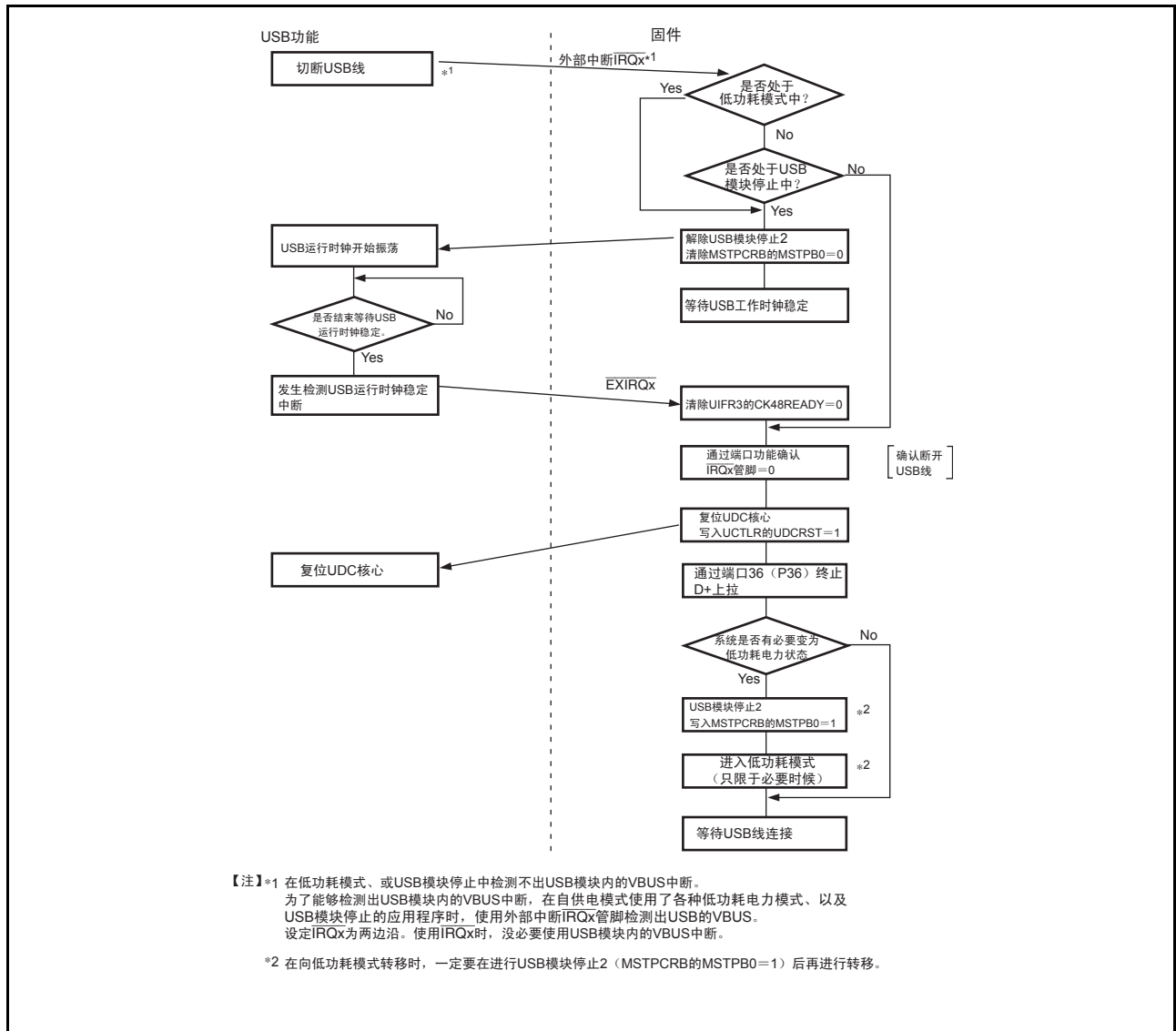


图 14.6 USB 线断开时的运行 (使用 USB 模块待机、各种低功耗模式时)

14.5.3 挂起 / 恢复

14.5.3.1 挂起 / 恢复时

图 14.7 和图 14.8 所示的是挂起 / 恢复处理的流程图。USB 总线从非挂起状态转换至挂起状态时；或者通过 up-stream 的恢复信号，USB 总线从挂起状态转换至非挂起状态时，请按照以下流程处理：

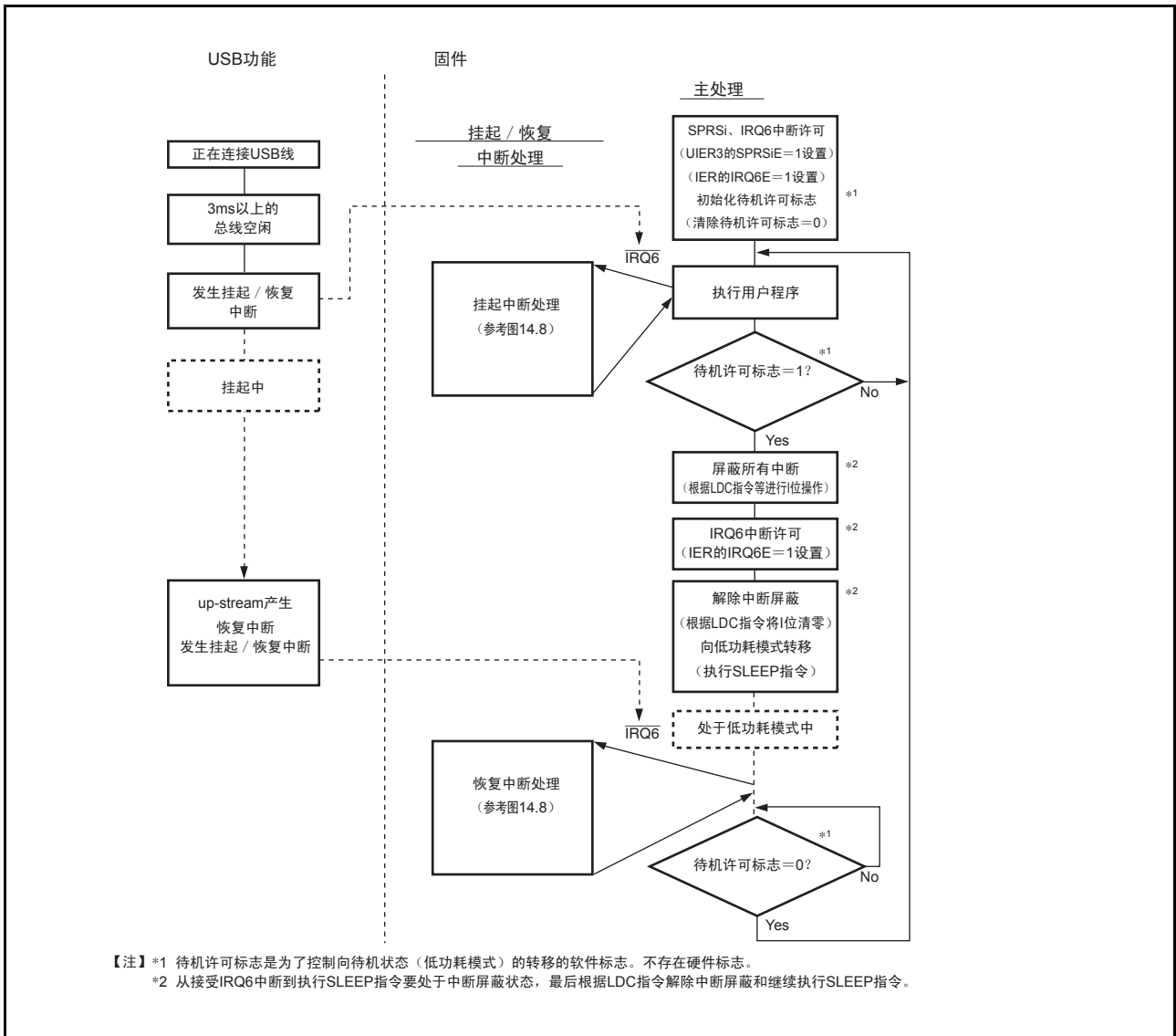


图 14.7 挂起 / 恢复处理流程图

14.5.3.2 挂起 / 恢复中断处理

图 14.8 所示的是挂起 / 恢复中断处理流程图。

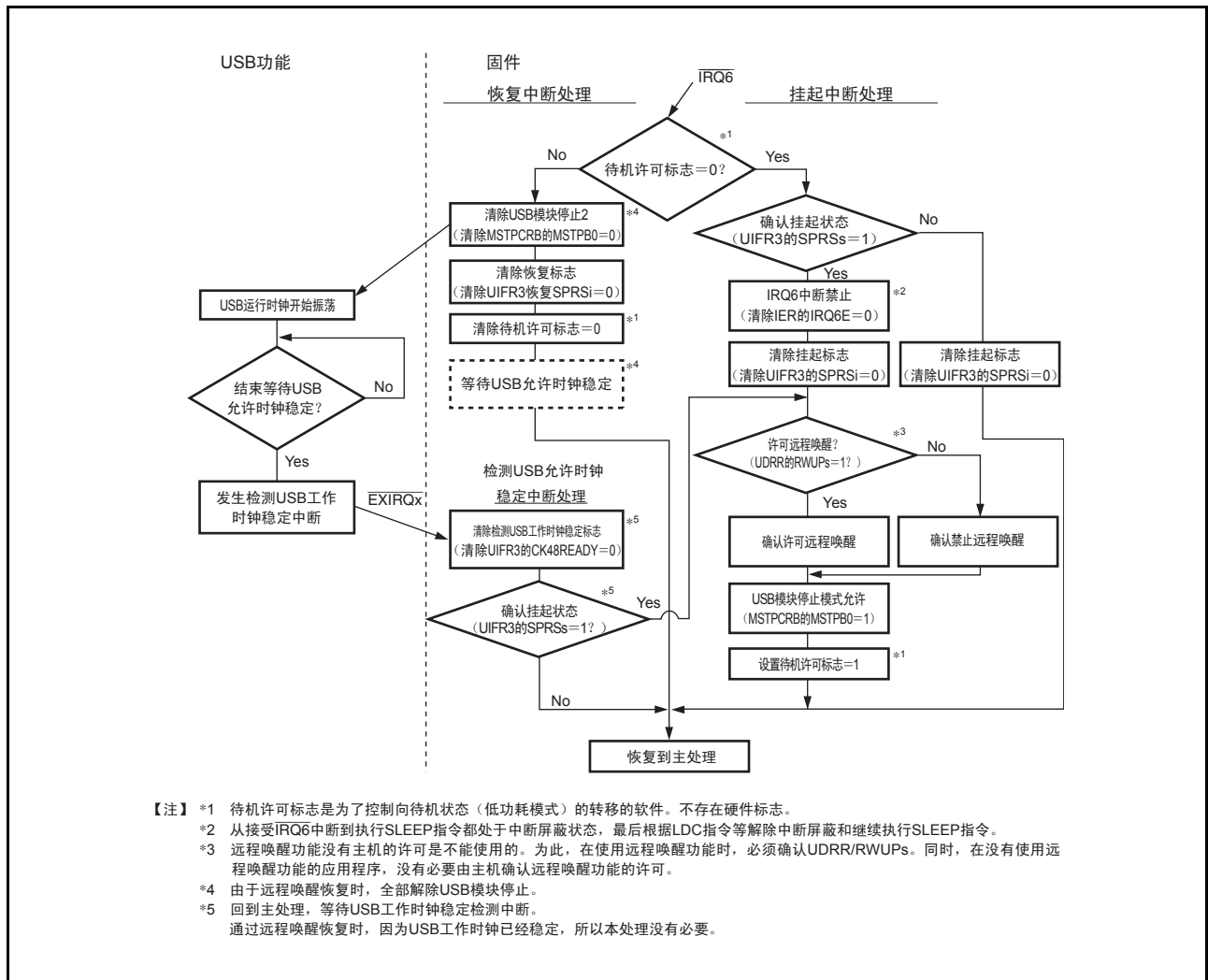


图 14.8 挂起 / 恢复时的中断处理流程图

14.5.3.3 挂起 / 远程唤醒处理

图 14.9 和图 14.10 所示的是挂起 / 远程唤醒处理流程图。通过来自本功能的远程唤醒信号，使 USB 总线从挂起状态转换至非挂起状态（恢复）时，请按照以下流程处理：

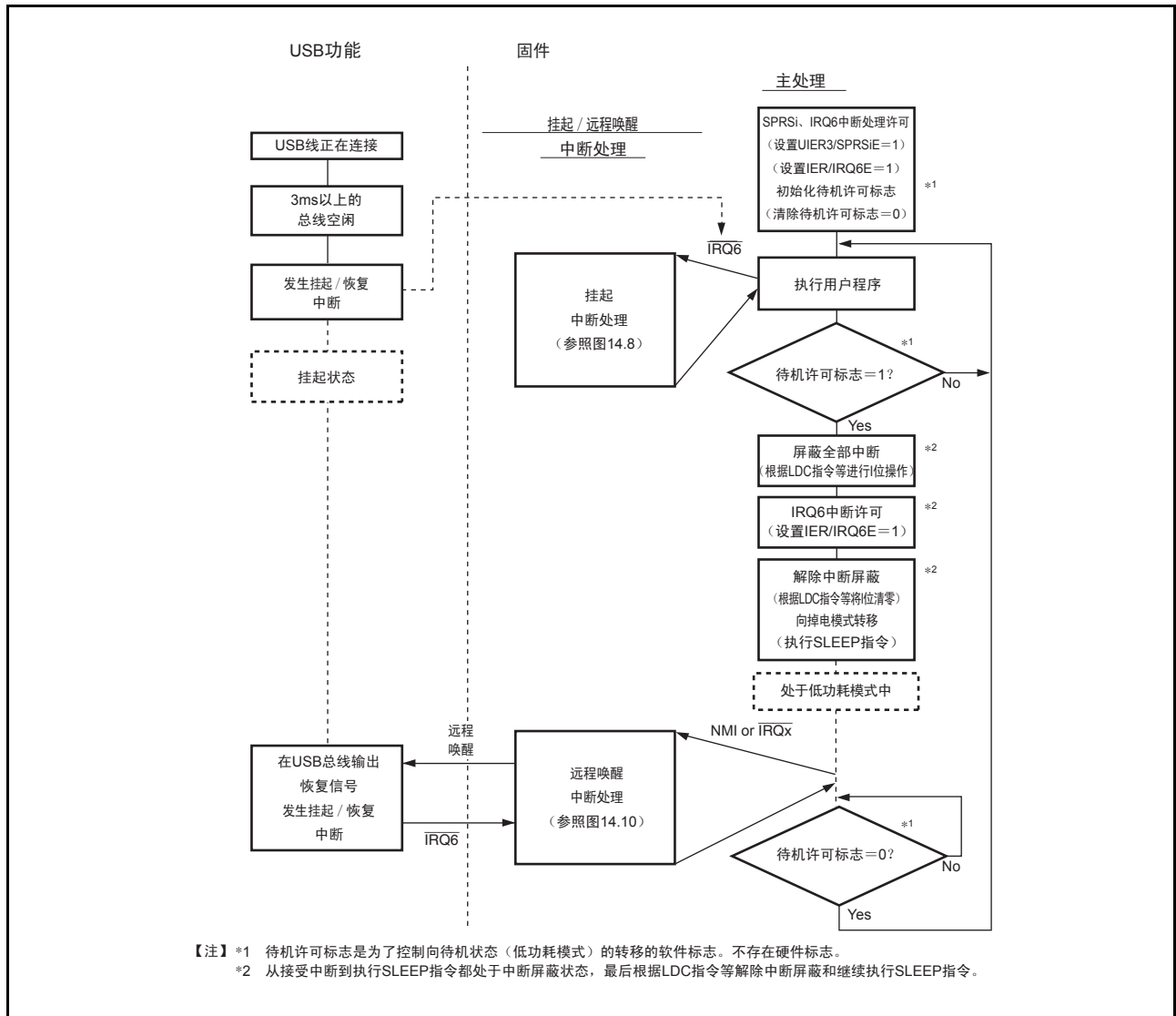


图 14.9 挂起 / 远程唤醒处理流程图

14.5.3.4 远程唤醒中断处理

图 14.10 所示的是远程唤醒中断处理流程图。

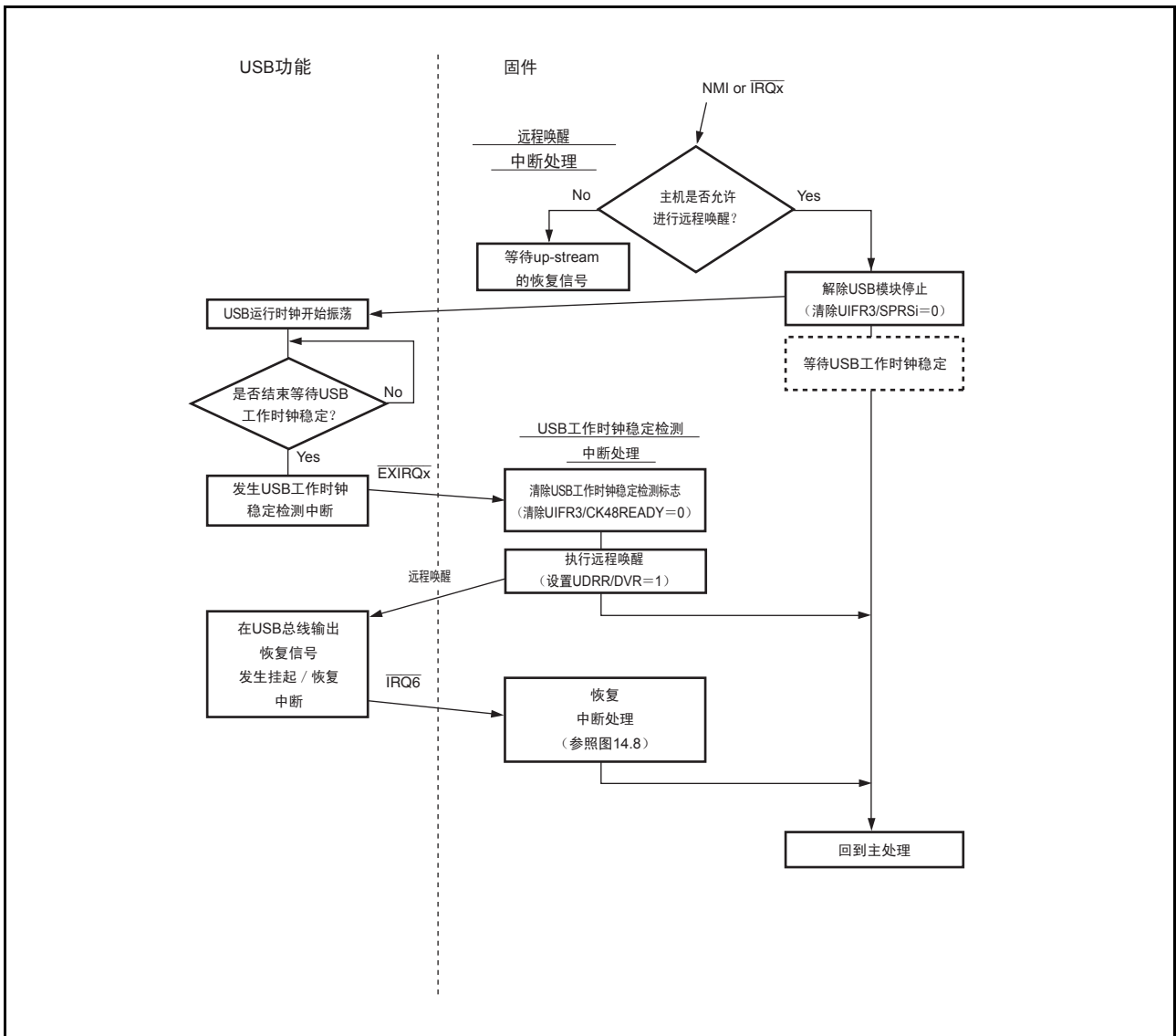


图 14.10 远程唤醒中断处理流程图

14.5.4 控制传送

控制传送由设置、数据（有时也会被省略）、状态 3 个阶段（图 14.11）构成。并且数据阶段由多个总线处理构成。以下图 14.12 ~ 图 14.16 所示的是各阶段的运行流程。

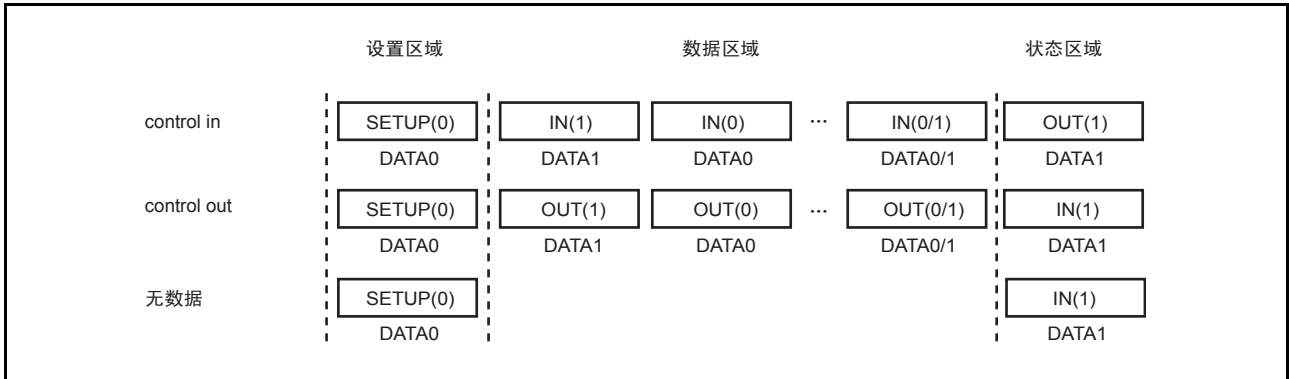


图 14.11 各传送阶段的结构

14.5.4.1 设置阶段

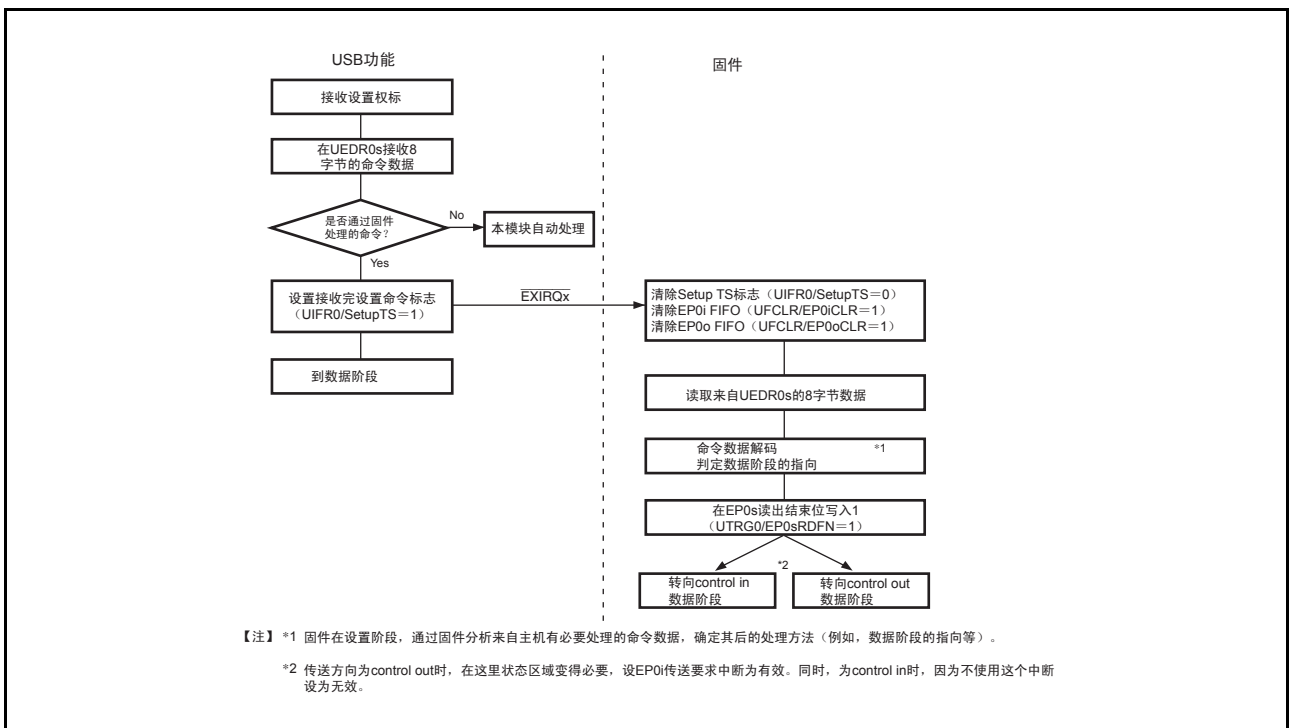


图 14.12 设置阶段的运行

14.5.4.2 数据阶段 (control in 时)

固件首先在设置阶段解析来自主机的指令数据，然后判断之后的数据阶段方向。根据指令数据的解析结果，数据阶段为 in 传送时，在 FIFO 中写入 1 信息包的想要传送到主机的数据。有想要传送的数据时，最初写入的数据传送到主机后 (UIFR0/EP0i TS = 1)，在 FIFO 中写入数据。

数据阶段的结束是依据主机发送 out 权标进入到状态阶段来判断。

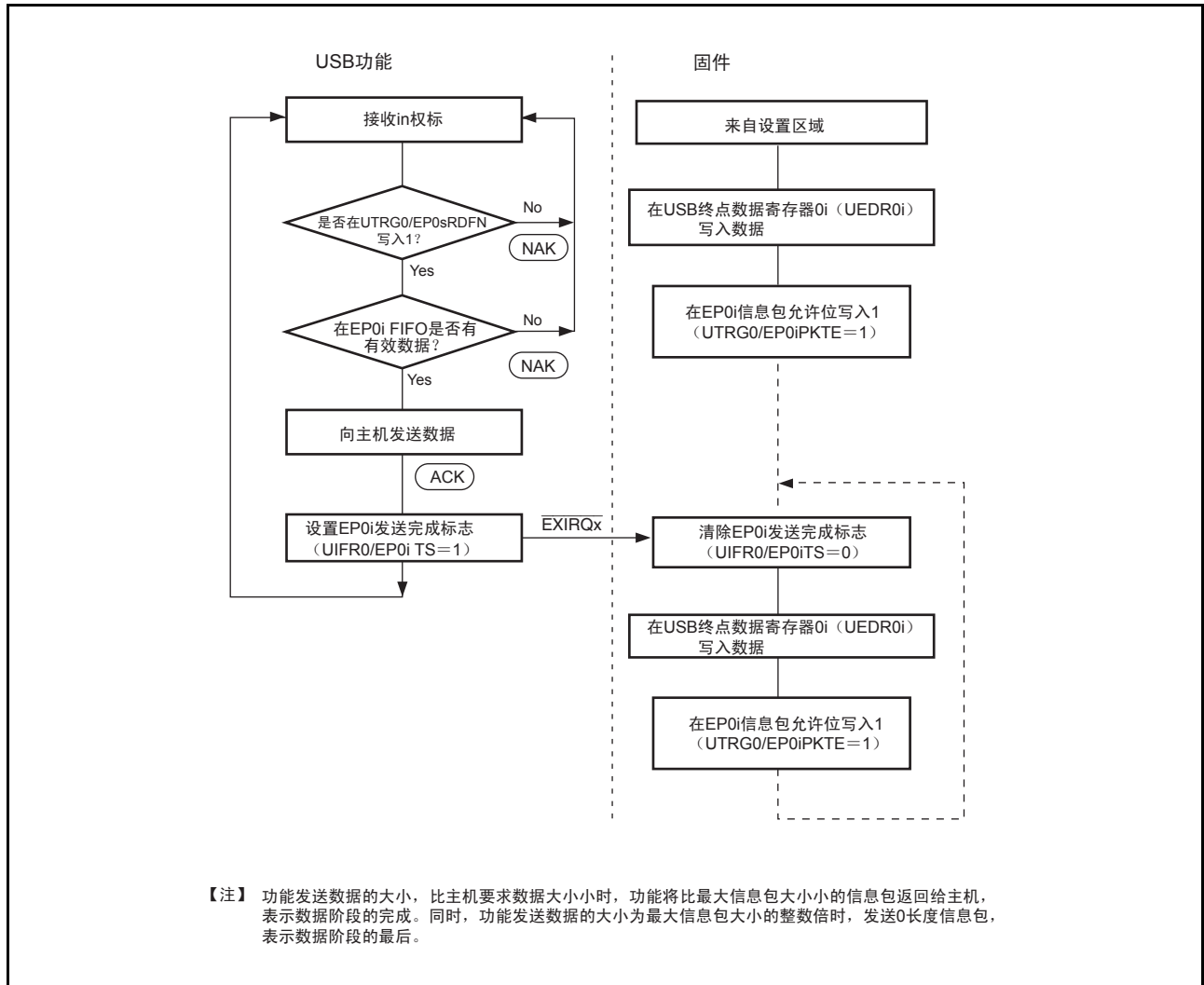


图 14.13 数据阶段的运行 (control in 时)

14.5.4.3 数据阶段 (control out 时)

固件首先在设置阶段解析来自主机的指令数据，再判断之后的数据阶段方向。根据指令数据的解析结果，数据阶段为 out 传送时，等待来自主机的数据，接收数据后 (UIFR0/EP0oTS = 1)，从 FIFO 读出数据。固件在 EP0o 读取完成位写入 1，使接收 FIFO 为空来等待下一个数据的接收。

数据阶段的结束是依据主机发送 in 权标进入到状态阶段进行判断。

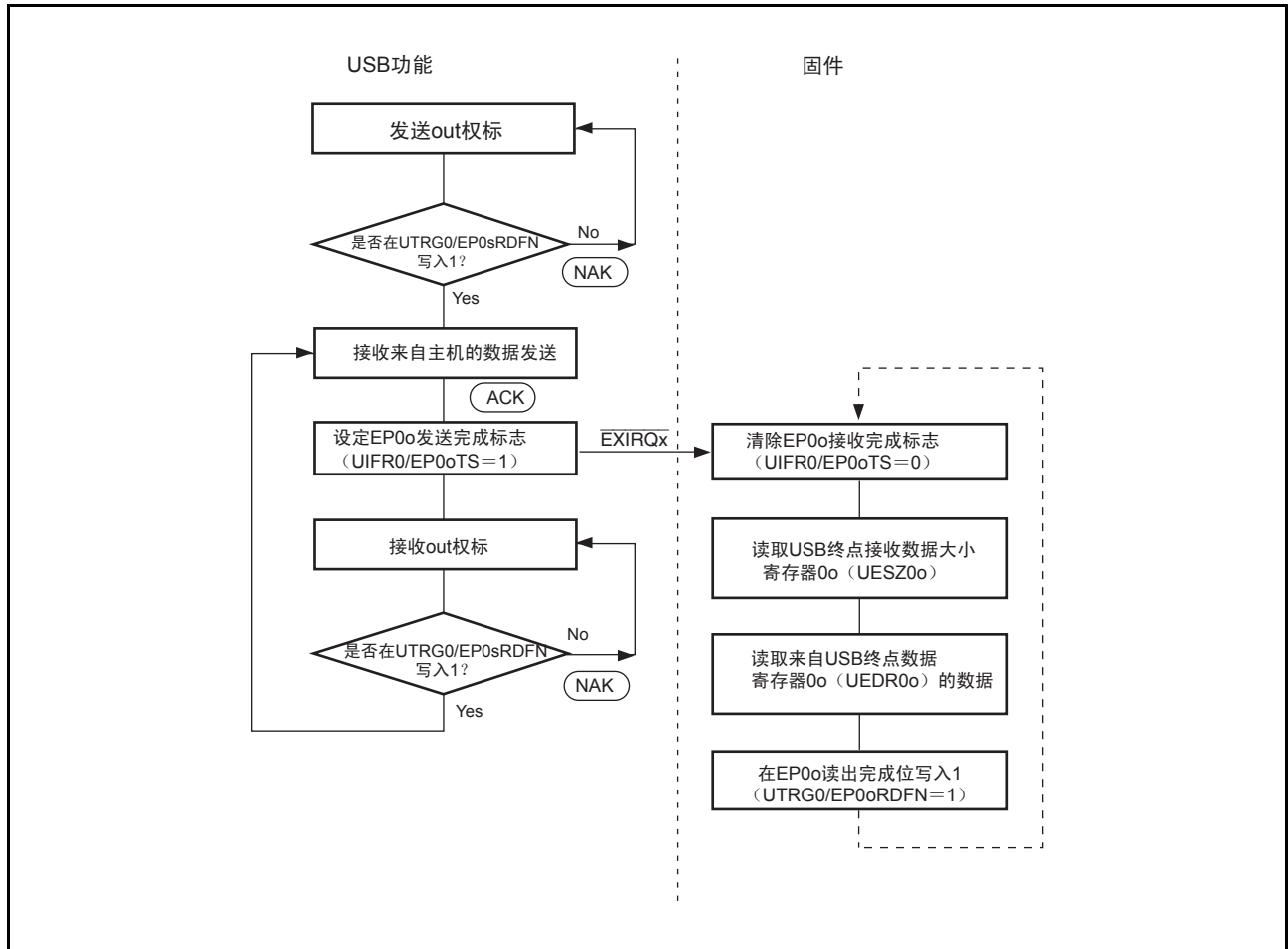


图 14.14 数据阶段 (control out 时) 的运行

14.5.4.4 状态阶段 (control in 时)

以来自主机的 out 权标开始 control in 的状态阶段。固件接收来自主机的 0 字节数据，完成控制传送。

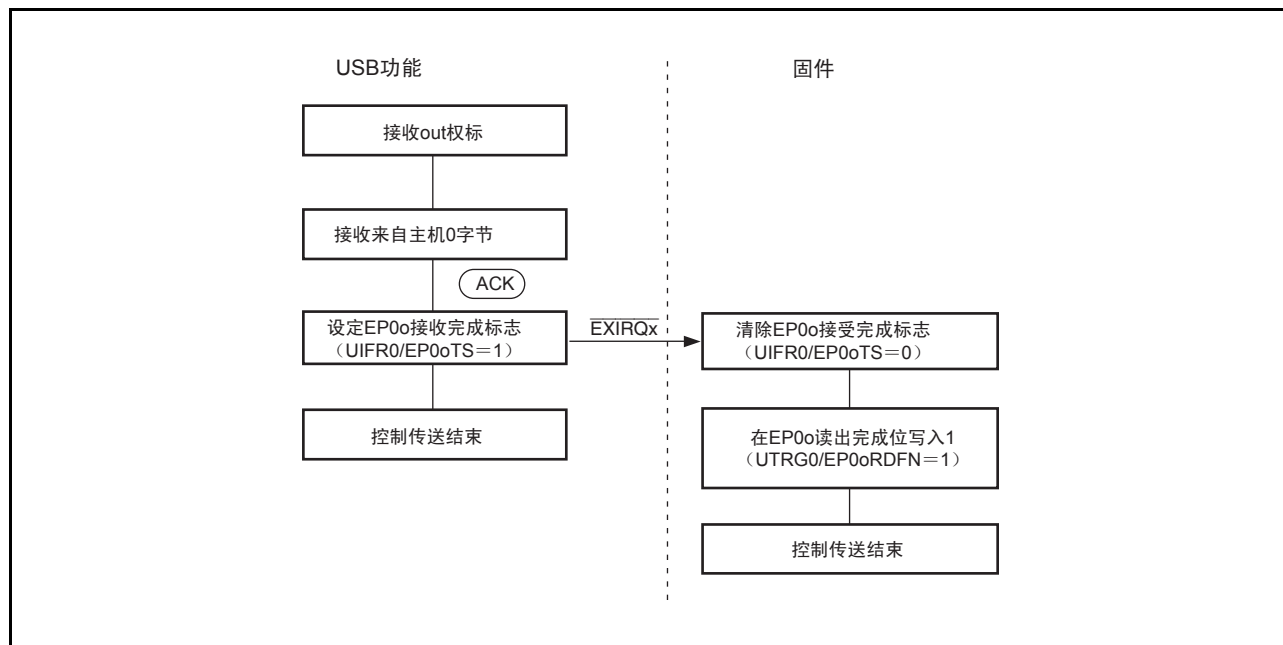


图 14.15 状态阶段 (control in 时) 的运行

14.5.4.5 状态阶段 (control out 时)

Control out 时的状态阶段以自主机的 in 权标开始。由于接收状态阶段开始的 in 权标时，数据还没有进入到 EP0i FIFO，因此会进入 EP0i 传送请求中断。固件通过此中断，识别到可开始状态阶段。接着，要向主机发送 0 字节数据，不要给 EP0i FIFO 中写入数据，而是在 EP0i 包允许位写入 1。据此，通过下一个 in 权标向主机发送 0 字节，结束控制传送。

但是，固件在全部完成数据阶段相关的处理后，请在 EP0i 包允许位写入 1。

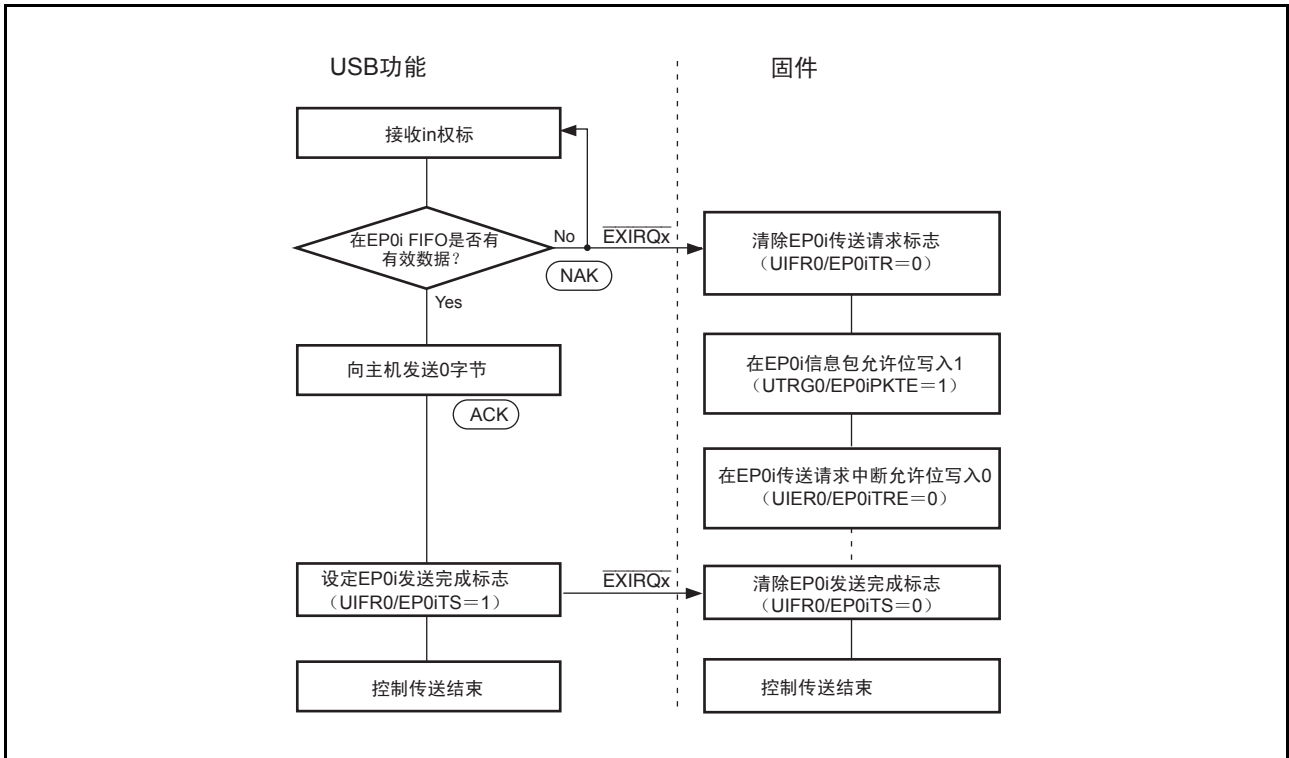


图 14.16 状态阶段 (control out 时) 的运行

14.5.5 中断 in 传送 — 终点 3 —

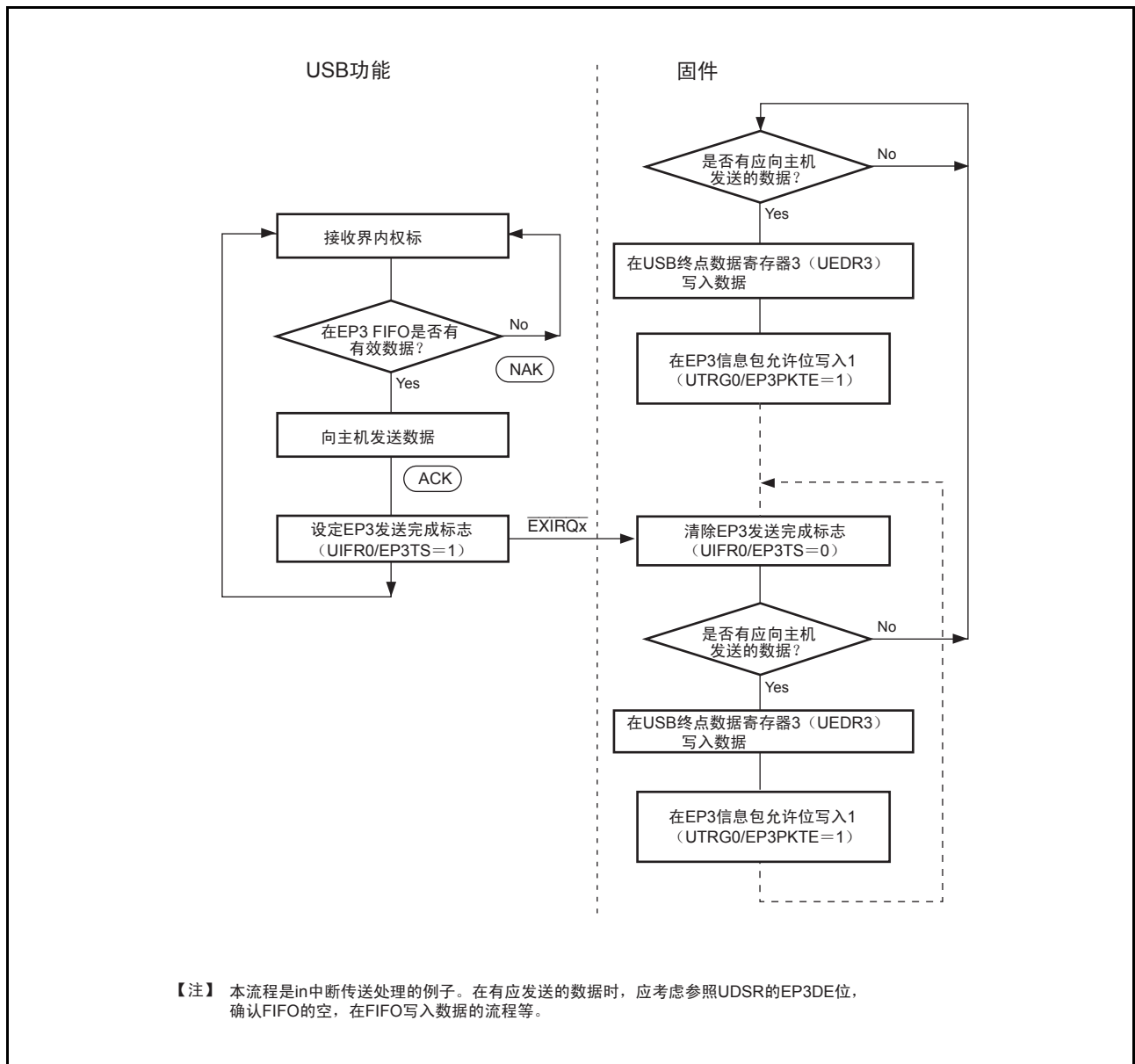


图 14.17 EP3 中断 in 传送的运行

14.5.6 Bulk in 传送 (2 面 FIFO) — 终点 1 —

EP1 配有 2 面 64 字节的 FIFO。但是，这种双 FIFO 配置对用户发送数据和写入发送数据没有影响。但是每 1 面进行 1 次写数据。例如：即使 2 面 FIFO 都为 EMPTY，连续写入 128 字节数据后，都不能进行 EP1PKTE。必须每 64 字节进行 EP1PKTE。

通过 bulk in 传送给主机发送数据时，首先在 UIER1/EP1EMPTYE 中写入 1，允许 EP1 FIFO 终点中断。由于最初 EP1 的 2 面 FIFO 都变为 EMPTY，就会立即发生 EP1 FIFO empty 中断。利用这个中断，在数据寄存器中写入发送数据。最初写入 1 面的发送数据后，另一面的 FIFO 是空的，因此可以立即向另一面写入发送数据。2 面同时 FULL 时，EP1EMPTY 变为 0。即使 1 面 EMPTY，UIFR1 / EP1EMPTY 也会被置位为 1。数据发送完成后，如果从主机返回 ACK，执行数据发送的 FIFO 变为 EMPTY。此时，另一面的 FIFO 中预备了有效的发送数据时，就可进行连续发送。

全部发送完成后，在 UIER1/EP1EMPTYE 中写入 0，请将来自 EXIRQ0 或者 EXIRQ1 管脚的中断请求置为禁止。

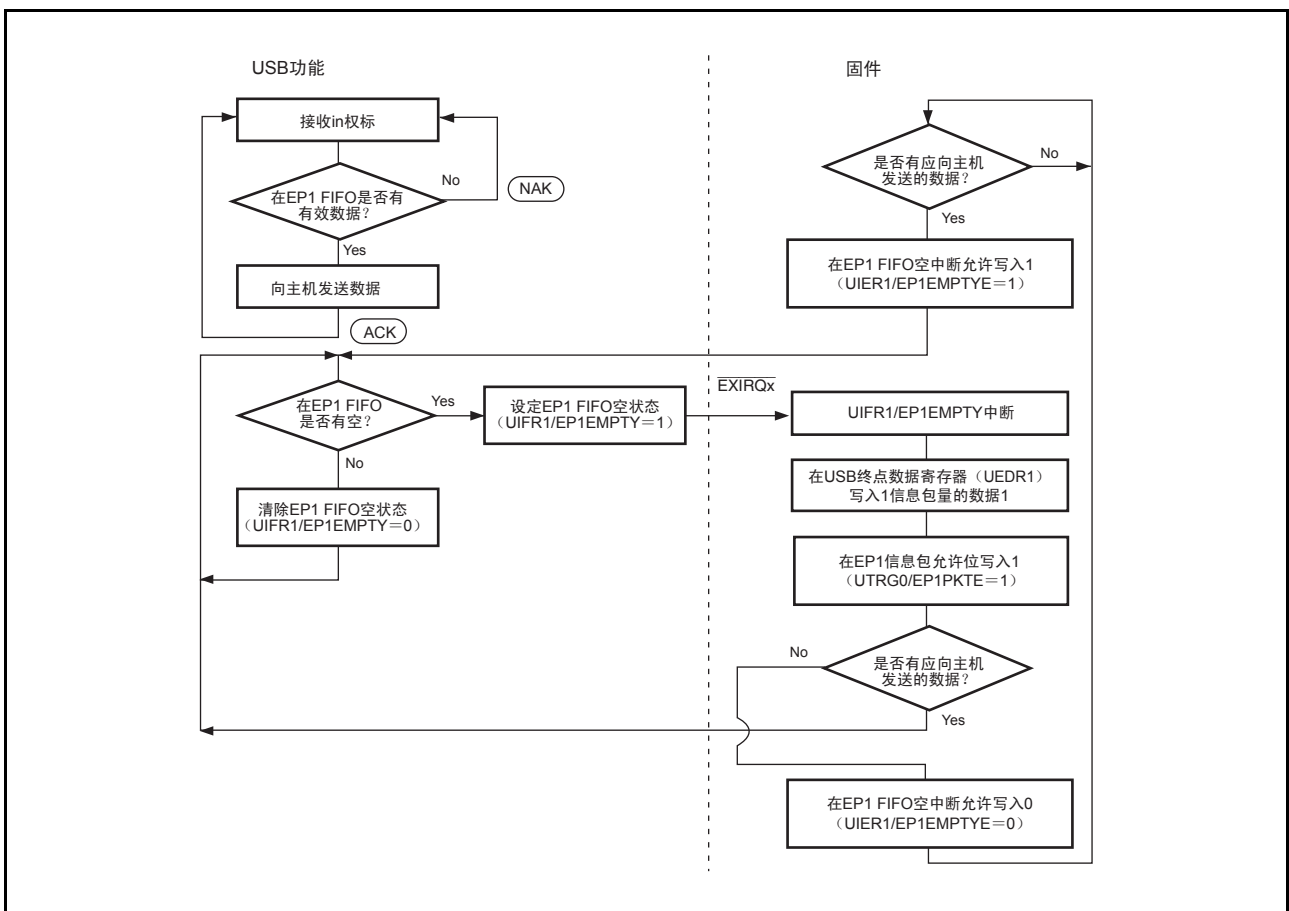


图 14.18 EP1 Bulk in 传送的运行

14.5.7 Bulk out 传送 (2 面 FIFO) — 终点 2 —

EP2 具有 2 面 64 字节的 FIFO。但是，这种双 FIFO 配置对用户发送数据和写入发送数据没有影响。

即使 1 面 FIFO 接收完成，UIFR1/EP2READY 也会被置位。FIFO 在 2 面都为 EMPTY 的状态，由于最开始的接收后，另一面 FIFO 为空，因此可以立即接收下一个信息包。2 面同时为 FULL 时，NAK 会自动返回到主机。接收数据后，完成接收数据的读取后，在 UTRG0/EP2RDFN 中写入 1。通过此操作，当前读取完成的 FIFO 变为 EMPTY，成为可接收下一个信息包的状态。

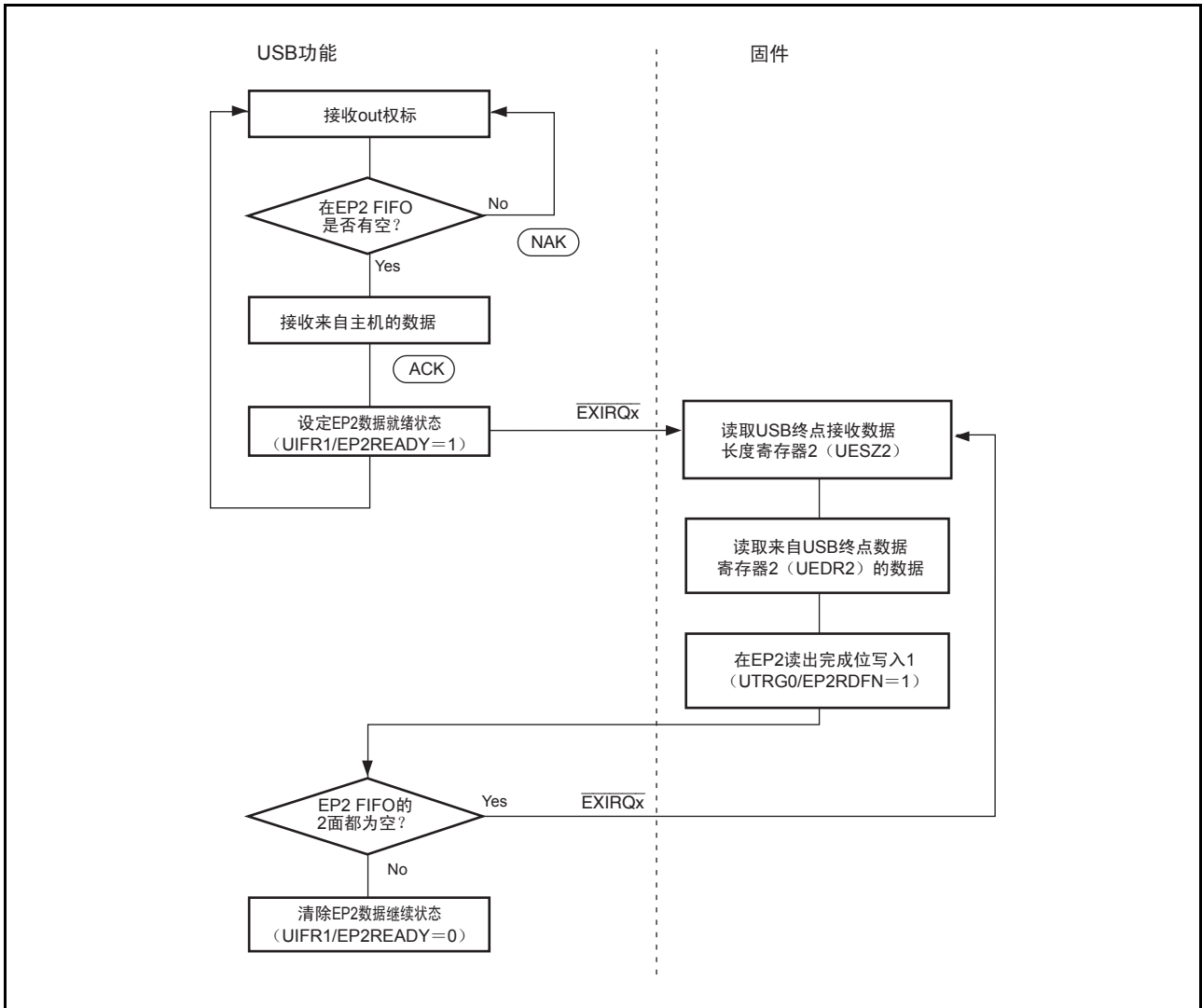


图 14.19 EP2 Bulk out 传送的运行

14.5.8 USB 标准指令和 class/vender 指令的处理

14.5.8.1 通过控制传送将发送来的指令进行处理

通过控制传送从主机发送来的指令，需要利用固件对指令进行解码和处理。以下表 14.5 所示的是固件指令解码。

表 14.5 固件的指令解码

不必用固件解码	需要用固件解码
Clear Feature	Get Descriptor
Get Configuration	Synch Frame
Get Interface	Set Descriptor
Get Status	Class/Vendor 指令
Set Address	
Set Configuration	
Set Feature	
Set Interface	

不需要用固件解码时，会自动进行指令解码、数据阶段、状态阶段处理。因此用户不用作任何操作。并且也不会发生中断。

需要用固件解码时，本模块将指令保存在 EP0s 的 FIFO。正常接收完成后，UIER0/SetupTS 标志被置位，从 EXIRQx 运行中断请求。在此中断程序中，通过 EP0s 的数据寄存器 (UEDR0s) 读取得 8 字节数据，请用固件解码。此后，根据解码结果，进行必要的的数据阶段、状态阶段的处理。

14.5.9 挂起操作

14.5.9.1 概要

本模块的挂起操作，有以下 2 种情况。

1. 固件因某些原因而强制使终点挂起时
2. 由于违反 USB 的规格，在本模块内部自动挂起时

本模块内具有保持各终点状态（是否挂起）的内部状态位。从主机发送来处理时，本模块参考此内部状态位，判断是否向主机返回挂起。

此位不能用固件解除。解除时，请使用 Clear Feature 指令，从主机清除。

但是，对于 EP0 的内部状态位只有在接收设置指令时会被自动清除。

14.5.9.2 固件需要强制挂起时

对于本模块，固件请求挂起时，使用 UESTL 寄存器。固件想要挂起特定的终点时，要设置 EPnSTL 对应的位（参考图 14.20 (1-1)）。此时内部状态位不发生变化。

接着，对于从主机设置 EPnSTL 位的终点，处理被传送来时，本模块参考内部状态位，如果没有被置位，则参考 EPnSTL 位（图 14.20 (1-2)）。在此，如果 EPnSTL 位没有被置位，内部状态位不发生变化，接受处理。如果 EPnSTL 位被置位，本模块要设置内部状态位，对主机返回挂起信号交换（图 14.20 (1-3)）。

内部状态位置位一次后，与 EPnSTL 无关，通过 Clear Feature 指令从主机清除前，都会保持内部状态位。即使通过 Clear Feature 指令将相应的位清除（图 14.20 (3-1)），在 EPnSTL 位设置期间，每次执行对相应终点的处理时，内部状态位均被置位，因此本模块会返回挂起信号交换（图 14.20 (1-2)）。要解除挂起，必须用固件清除 EPnSTL 位，甚至需要用 Clear Feature 指令清除内部状态位（图 14.20 (2-1) ~ (2-3)）。

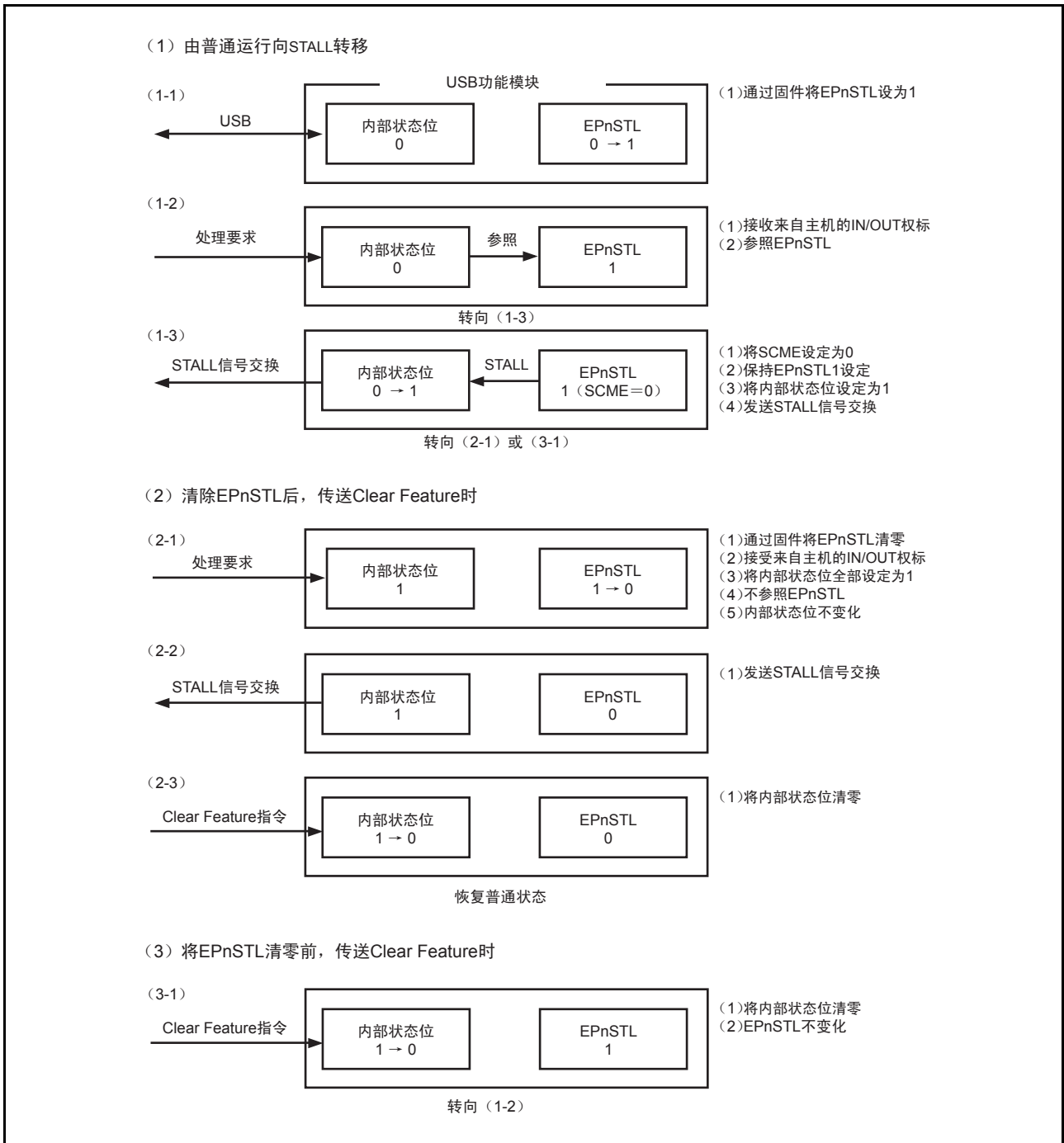


图 14.20 需要用固件强制挂起时

14.5.9.3 USB 功能模块自动被挂起时

用 Set Feature 指令设定挂起时，如果本模块的结构内容和通过 Get Descriptor 返回到主机的信息有所差异时或者违反 USB 规格时，与 EPnSTL 无关，本模块会自动设置对应终点的内部状态位，并返回挂起信号交换（图 14.21 (1-1)）。

内部状态位被置位一次后，与 EPnSTL 无关，通过 Clear Feature 指令从主机被清除前，仍保持内部状态位。用 Clear Feature 指令清除相应位后，就会参考 EPnSTL（图 14.21 (3-1)）。在内部状态位置位期间，对于相应终点即使执行处理，内部状态位也会被置位，因此本模块会返回挂起信号交换（图 14.21 (2-1)、(2-2)）。要解除挂起，必须要用 Clear Feature 指令清除内部状态位（图 14.21 (3-1)）。如果是通过固件设置 EPnSTL，也请清除 EPnSTL（图 14.21 (2-1)）。

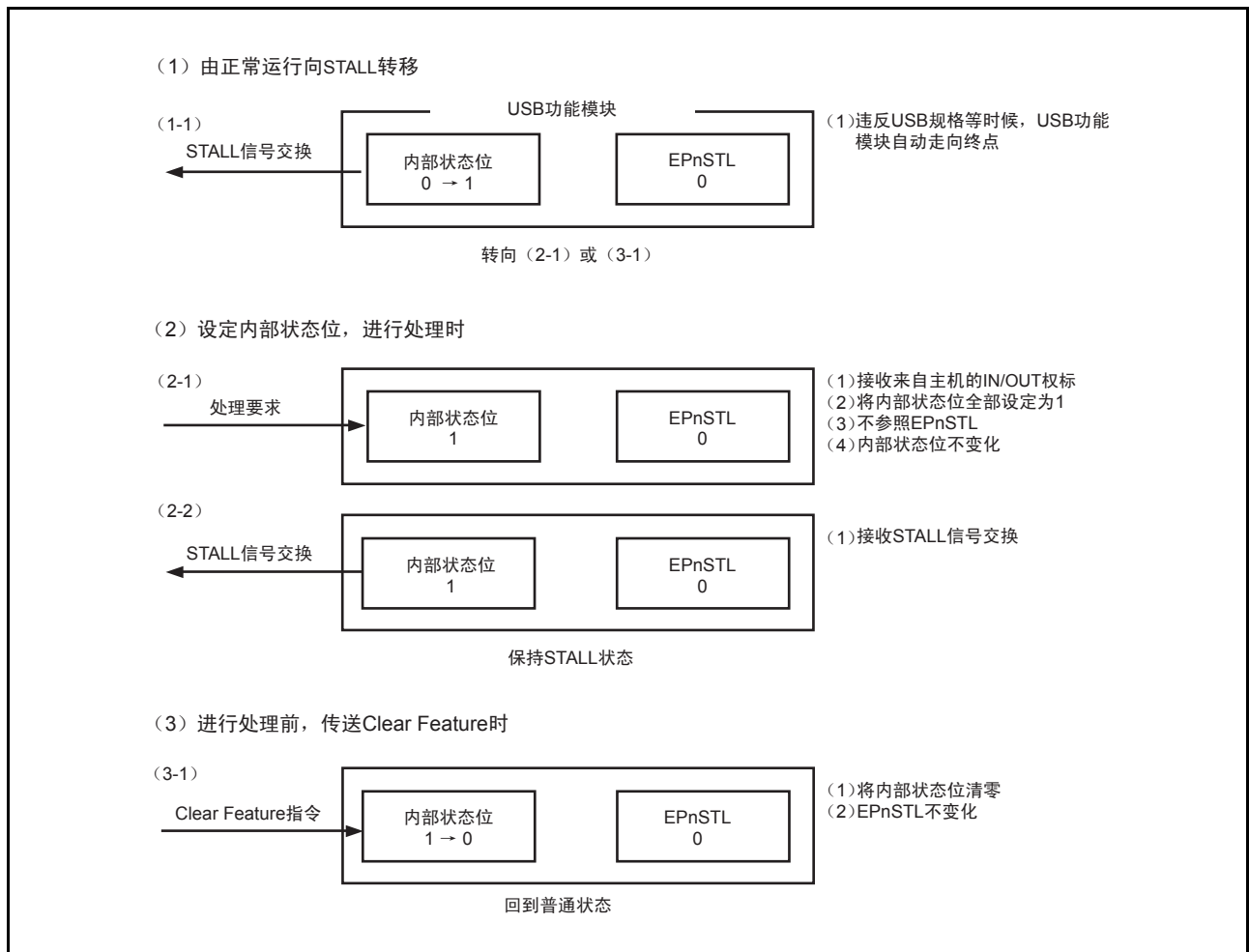


图 14.21 USB 功能模块被自动挂起时

14.6 DMA 传送说明

USB 数据的 DMA 传送有 USB 请求与自动请求 2 种。

14.6.1 通过 USB 请求的 DMA 传送

14.6.1.1 概要

进行内置 DMAC 的 USB 请求 ($\overline{\text{DREQ}}$ 电平启动) 传送时, 仅支持满地址模式的普通模式 (周期挪用模式)。可传送的终点只有 Bulk 传送的 EP1、EP2 (对应寄存器是 UEDR1、UEDR2)。并且请将本模块作为区域 6 的外部设备进行访问。不能当作外置 ACK 设备进行访问 (不能单地址传送)。

然而, 即使将 UDMAR 寄存器的 EP2T1 位置 1, 允许 DMA 传送, 对于 EP2, 忽略 0 字节数据的接收。

14.6.1.2 内置 DMAC 的设定

内置 DMAC 的设定请在 USB 请求 (使用 $\overline{\text{DREQ}}$ 信号)、低电平输入启动、字节长度、全地址模式传送、DMABCR 寄存器的 DTA 位 = 1 时使用。此时, 如果完成内置 DMAC 设定次数的传送, DMAC 就会停止。但是, 本模块仍残留有 DMA 传送请求时, 与 DMAC 的状态无关, 可继续低电平断言 $\overline{\text{DREQ}}$ 信号, 请注意。

14.6.1.3 关于 EP1 的 DMA 传送

UDMAR 寄存器的 EPIT1 位是 DMA 传送允许位, 使用 UDMAR 寄存器的 EPIT0 位指定的信号。如果在 EPIT1 中写入 1, EP1 的数据 FIFO 即使 1 面为 EMPTY, $\overline{\text{DREQ}}$ 信号也会被低电平断言。FIFO2 面都为 FULL, $\overline{\text{DREQ}}$ 信号就会变为 High。

- 关于 UTRG0 寄存器的 EP1PKTE

用 DMA 传送来传送 EP1 的发送数据时, 如果 1 面的数据 FIFO (64 字节) 变为 FULL, 在模块内部进行与在 EP1PKTE 中自动写 1 相同的处理。因此, 传送数据是 64 字节的倍数时, 用户不必在 EP1PKTE 中写入 1。其它情况下, 只有对未满 64 字节的最后的数据, 用户需要在 EP1PKTE 中写入 1。此处理必须通过 DMAC 的 DMA 传送完成中断进行。如果是未满 64 字节的数据传送以外的情况, 用户在 EP1PKTE 中写入 1, 就成为多余的处理, 而不能正常运行。

例如, 表示用 EP1 向主机发送 150 字节的数据时。此时在内部进行与图 14.22 的 2 个地方自动向 EP1PKTE 写入 1 相同的处理。由于此处理是在当前选择的数据 FIFO 的数据为 FULL 时进行的, 因此只有在传送 64 字节的数据时自动处理。发送不满 64 字节的数据时, 不自动进行。

- EP1 的 DMA 传送顺序例

- UDMAR/EPIT1、EPIT0 位设定
- DMAC 各种设定 (DMAC 中设定 150 字节的传送次数)
- DMAC 启动
- DMA 传送
- 由于 DMA 传送完成中断, 在 UTRG0/EP1PKTE 位写入 1

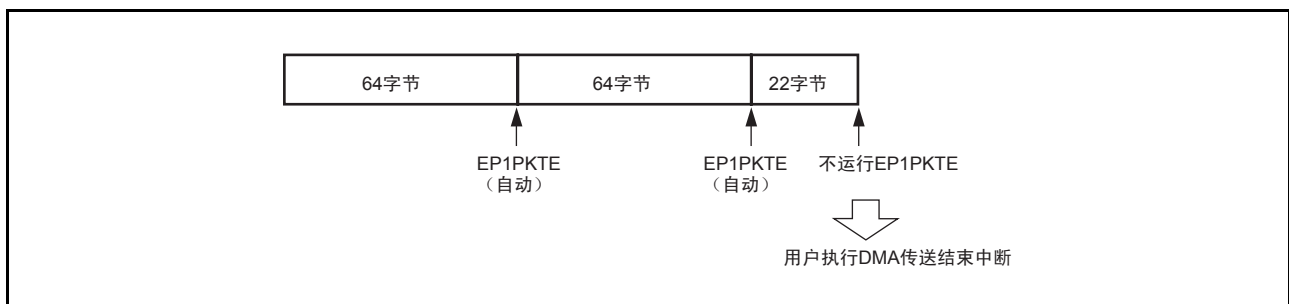


图 14.22 UTRG0 寄存器的 EP1PKTE 运行

14.6.1.4 关于 EP2 的 DMA 传送

UDMAR 寄存器的 EP2T1 位是 DMA 传送允许位，要使用 UDMAR 寄存器的 EP2T0 位指定的 $\overline{\text{DREQ}}$ 信号。如果在 EP2T1 中写入 1，EP2 的数据 FIFO 1 面以上为 FULL (READY 状态) 时， $\overline{\text{DREQ}}$ 信号会被断言为 Low，即有效的接收数据存在于 FIFO 内时 (READY 状态)， $\overline{\text{DREQ}}$ 信号都会被断言为 Low。读取所有的接收数据，2 面 FIFO 都变为 EMPTY， $\overline{\text{DREQ}}$ 信号就会变为 High。

- UTRG0 寄存器的 EP2RDFN 的运行

用 DMA 传送来发送 EP2 的接收数据时，读取 1 面数据 FIFO 后，请不要在 EP2RDFN 位写入 1。不使用 DMA 传送时，读取 1 面数据 FIFO 后，如果不在 EP2RDFN 位写 1，就不能读取下一个数据，但是使用 DMA 传送时，当前选择面的数据 FIFO 为 EMPTY，就会在模块内部执行与自动在 EP2RDFN 中写 1 的相同处理，因此用户不必在 EP2RDFN 中写 1。如果用户在 EP2RDFN 中写 1，则成为进行多余的处理，而不能正常运行，请注意。

例如：表示用 EP2 从主机接收 150 字节的数据时的情况。此时，在内部执行与图 14.23 的 3 个地方自动在 EP2RDFN 中写 1 相同的处理。由于此处理是在当前选择的数据 FIFO 的数据为 EMPTY 时进行的，无论传送 64 字节的数据，还是传送 64 字节以下的数据，都会同样地被自动处理。

- EP2 的 DMA 传送顺序例

DMAC 的传送单位以 1 信息包为单位。因此，EP2READY 标志置位后，确认通过主机接收的数据长度，将接收的大小设定为 DMAC 的传送次数。

- A. UDMAR/EP2T1、EP2T0 位设定

- B. UIFR1/EP2READY 标志置位等待

- C. DMAC 各种设定

读取 UESZ2 的值，设定接收数据长度 (64 字节以下) 的传送次数

- D. DMAC 启动

- E. DMA 传送 (传送 64 字节以下)

- F. DMA 传送完成等待

- G. 反复以上 2~6

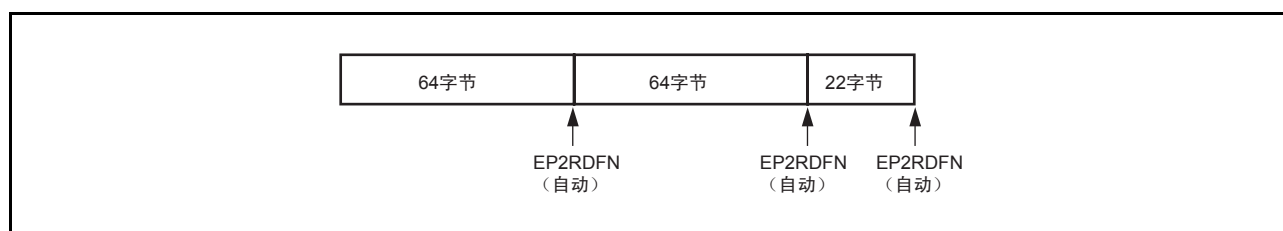


图 14.23 UTRG0 寄存器的 EP2RDFN 运行

14.6.2 通过自动请求的 DMA 传送

14.6.2.1 概要

传送内部 DMAC 的自动请求时，可选择突发模式传送或周期挪用传送。可传送的终点为所有的数据寄存器 (UEDR0s、UEDR0i、UEDR0o、UEDR1、UEDR2、UEDR3)。确认对应各数据寄存器的标志或中断后，再启动 DMA。另外，在自动请求模式中，因为不使用 UDMAR 寄存器，所以，请设定 UDMAR=H'00。

14.6.2.2 内部 DMAC 的设定

请在自动请求、字节大小、满地址模式传送及数据寄存器的最大数据包大小，使用内部 DMAC 的设定。此时，如果内部 DMAC 在特定时间里的传送结束，则停止 DMAC。

14.6.2.3 EP0i、EP1、EP3 的 DMA 传送

- UTRG0 寄存器的 EPnPKTE (n=0i、1、3)

传送自动请求时，因不能自动进行给 EPnPKTE 写入 1 的处理，此时，请注意。必须通过 CPU 给 EPnPKTE 写入 1。

通过 EP1 给主机传送 150 字节的数据时的情况如例所示。此时，有必要在图 14.26 的 3 个地方，给 EPnPKTE 写入 1。

- EP1 的 DMA 传送顺序例

DMAC 的传送以 1 数据为单位进行。因此，传送次数请设定在各终点最大数据包大小以下。

- 确认 UIFR1/EP1EMPTY 标志 = 1
- EP1 数据传送用的 DMAC 的各种设定 (自动请求、地址设定等)
- 给 DMAC 设定 64 字节 (最大数据包大小以下) 的传送次数
- DMAC 启动 (读取 DTE = 0 后、写入 DTE = 1)
- DMA 传送
- DMA 传送结束后，UTRG0/EP1PKTE 位写入 1
- 重复上述 A ~ F
- 确认 UIFR1/EP1EMPTY 标志 = 1
- 给 DMAC 设定 22 字节的传送次数
- DMAC 启动 (读取 DTE = 0 后、写入 DTE = 1)
- DMA 传送
- DMA 传送结束后、UTRG0/EP1PKTE 位写入 1

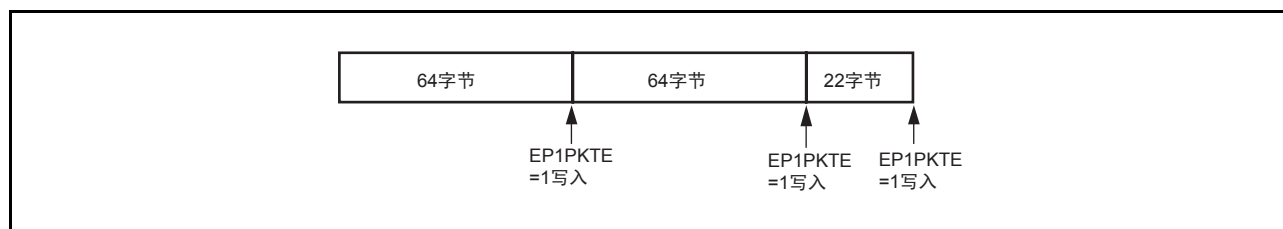


图 14.24 UTRG0 寄存器的 EP1PKTE 运行 (自动请求)

14.6.2.4 EP0o、EP2 的 DMA 传送

- UTRG0 寄存器的 EPnRDFN (n=0o、2)

传送自动请求时，因不能自动进行给 EPnRDFN 写入 1 的处理，此时，请注意。必须通过 CPU 给 EPnRDFN 写入 1。

通过 EP2 从主机接收 150 字节的数据时的情况如例所示。此时，有必要在图 14.27 的 3 个地址，给 EP2RDFN 写入 1。

- EP2 的 DMA 传送顺序例

DMAC 的传送以 1 数据包为单位进行。因此，传送次数请设定在各终点最大数据包大小以下。

- 拥有 UIFR1/EP2READY 标志位
- EP2 数据传送用的 DMAC 的各种设定（自动请求、地址设定等）读取 UESZ2 的值，并设定接收数据大小（64 字节以下）的传送次数
- DMAC 启动（读取 DTE = 0 后、写入 DTE = 1）
- DMA 传送（传送 64 字节以下）
- DMA 传送结束后，UTRG0/EP2RDFN 位写入 1
- 重复上述 A ~ E

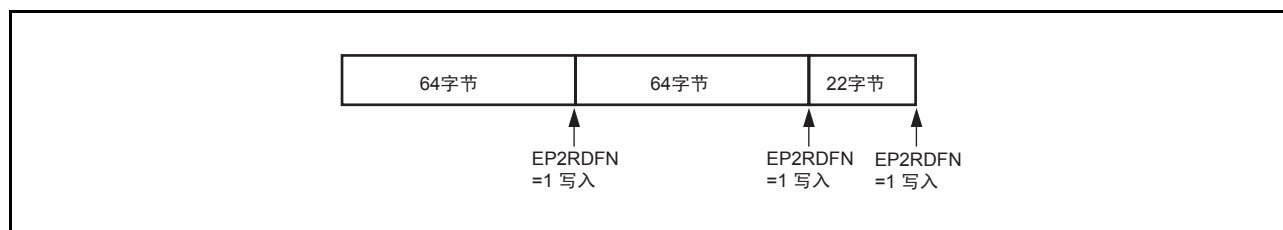


图 14.25 UTRG0 寄存器的 EP2RDFN 运行（自动请求）

14.7 USB 外部电路

本 LSI 的 USB 外部电路例如图 14.26 和图 14.27 所示。

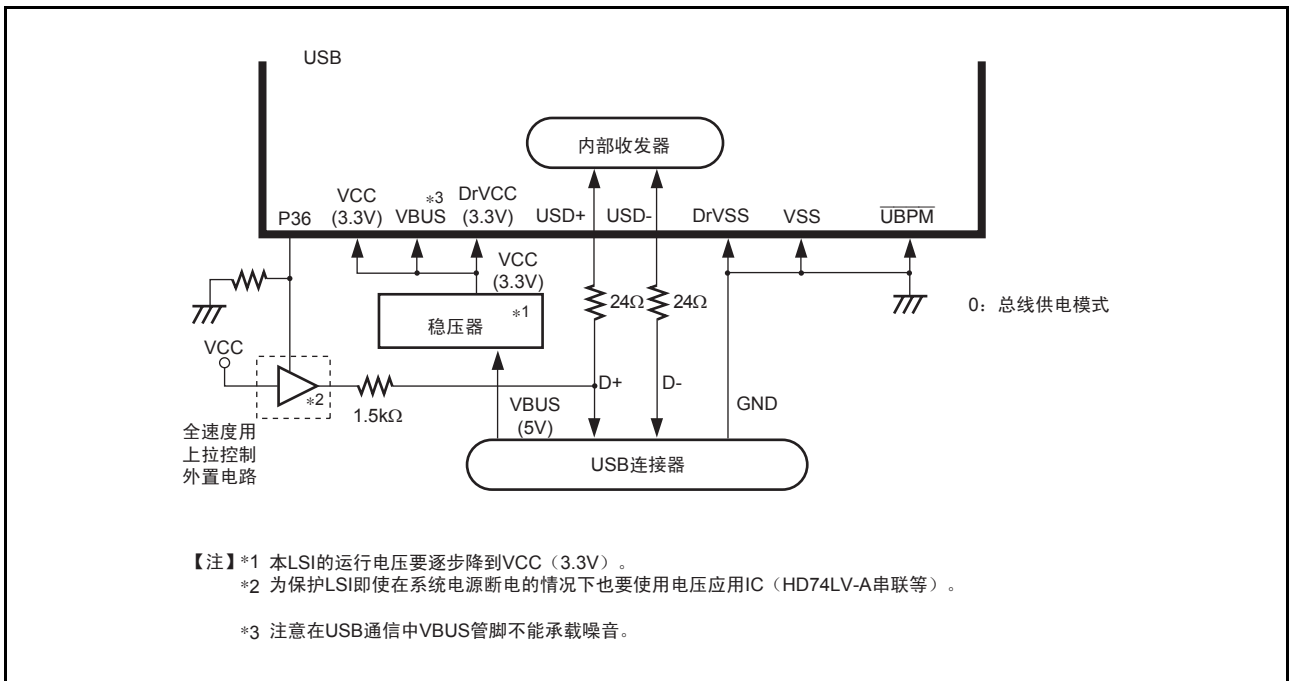


图 14.26 总线电源模式时的电路例

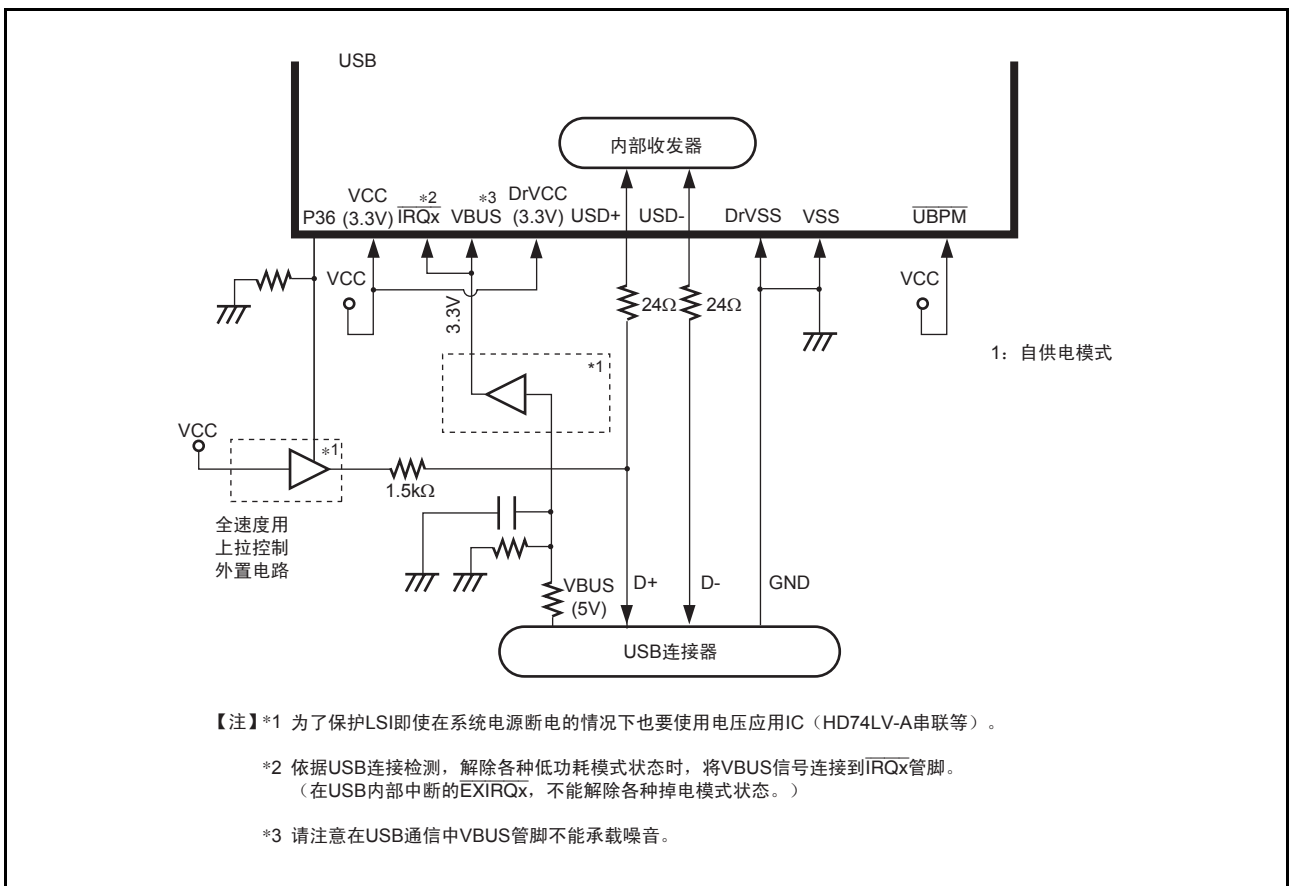


图 14.27 自身电源模式时的电路例

14.8 使用注意事项

14.8.1 使用仿真器注意事项

1. 如果将UEDR0_o和UEDR2用I/O寄存器窗口功能表示，EP0_oFIFO和EP2_oFIFO的读取指针会误操作，UEDR0_o、UESZ0_o、UEDR2、UESZ2就不能正确读取，因此请不要显示UEDR0_o和UEDR2。
2. 在E6000，USB模块被实装在外扩展电路板上，由于在外模块处理需要访问，就会有以下制约。
E10A和产品芯片没有以下制约。
 - H8S/2218群的模式7（单芯片模式）中不支持USB运行。
 - 在H8S/2218群的模式6（内置ROM有效模式）和H8S/2212群的模式7（单芯片模式）下使用时，在初始状态 $\overline{CS6}$ 和A9~A0变为输入管脚，因此访问本模块前，设定为P72DDR = 1、AE3~AE0 = B'0010、PC7DDR~PC0DDR = H'FF，将 $\overline{CS6}$ 和A9~A0置为输出管脚。
 - 在H8S/2218群的模式4、5（内置ROM无效模式）下使用时，设定为P72DDR = 1、AE3~AE0 = B'0010，请将 $\overline{CS6}$ 和A9~A8置为输出管脚。

14.8.2 关于总线接口

本模块的总线接口按照外部区域6的总线规格。因此，访问本模块前，在总线控制器的寄存器务必将区域6设定为总线宽8位、3状态访问。并且，地址H'C00100~H'DFFFFFF是USB保留区域，是禁止访问区域。

14.8.3 关于运行频率

单片机的主时钟请使用24MHz或者16MHz。

将主时钟作为基本时钟，在内置PLL将24MHz倍频或者将16MHz3倍频，生成48MHz的USB运行时钟。并且，本模块不支持中速模式、睡眠模式、监视模式、子激活模式、子睡眠模式，因此请务必在高速模式下使用。

14.8.4 接收设置数据

接收8字节设置数据的EP0_sFIFO，需要注意以下几点：

在USB中，必须要接收设置指令，因此UDC的写入比单片机的读取优先级更高。接收完成后，在单片机进行数据读取中，开始接收下一个设置指令时，为了使写入优先，在单片机的读取必须强制无效。因此，开始接收后，读取的值为不定值。

由于它不能保证FIFO中使用的DP-RAM写入读取都访问同一个地址时的数据，所以才会强制使读取无效。

14.8.5 FIFO的清除

通信中途将USB线拔下时，有时在FIFO内会残留以前发送接收中的数据。因此，将线连接上后，请迅速清除FIFO。同时，总线复位时也要清除FIFO。

然而，从主机正在接收数据或者正在向主机发送数据，请不要清除FIFO。

14.8.6 $\overline{IRQ6}$ 中断

$\overline{IRQ6}$ 的挂起/恢复中断请求必须被设定为下降沿。

14.8.7 数据寄存器的通读 / 盖写

在 CPU 中读取 / 写入数据寄存器时，请注意以下几点：

- 发送用数据寄存器 (UEDR0i、UEDR3、UEDR1)
发送用数据寄存器的写入请不要大于最大包的大小。具有 2 面 FIFO 的 EP1，一次写入务必在最大包大小以内。写入数据后，如果在 UTRG0 寄存器的 PKTE 中写入 1，就会在本模块内切换面，并可以在另一面写入数据。因此，请不要 2 面连续进行数据写入。
- 接收用数据寄存器 (UEDR0o、UEDR2)
接收用数据寄存器请不要读取大于有效接收数据数。即不要读取的字节数大于在接收数据大小寄存器中显示的字节。具有 2 面 FIFO 的 EP2，1 次可读取的最大数据数为最大信息包大小。完成当前有效面的数据读取后，请务必在 UTRG0 寄存器的 RDFN 中写入 1。
通过此操作，就会切换到另一面，新的字节数也会反映到接收数据大小，就可以读取下一个数据。
并且，没有接收数据时，请不要读取数据寄存器。如果读取，控制模块内 FIFO 的指针会发生变化，不能正常运行。

14.8.8 关于复位

- USB 通信运行中的手动复位，如果保持 USD+、USD- 的管脚状态，本 LSI 会停止，因此请不要使用。本模块寄存器使用了一部分同步复位，因此在模块内部时钟稳定后，需要解除复位。因此在初始设定时的复位解除，请按照以下顺序进行：
 - A. USB 模块停止 1 解除：清除 EXMDLSTP 寄存器的 USBSTOP1 = 0
 - B. USB 运行时钟选择：向 UCTLR 寄存器的 UCKS3 ~ 0 写入
 - C. USB 模块停止 2 解除：清除 MSTPCRB 寄存器的 MSTPB0 = 0
 - D. 等待 USB 运行时钟稳定：等待 UIFR3 寄存器的 CK48READY = 1 设置
 - E. USB 接口复位解除：清除 UCTLR 寄存器的 UIFRST = 0
 - F. 解除 UDC 内核复位：清除 UCTLR 寄存器的 UDCRST = 0
 详细请参考 14.5.1 初始设定及 14.5.2 USB 线的连接 / 断开的流程。
- 由于在通过看门狗定时器（以下 WDT）进行加电复位时，USB 的寄存器不被初始化，CPU 失控时，通过 WDT 进行加电复位后 USB 有可能不能正常运行（从管脚输入加电复位信号时，USB 的寄存器会被初始化，因此不存在什么问题）。
因此，在复位后的初始化程序，向以下 3 个寄存器写入初始值，确认 USB 的所有寄存器进行初始化。
UCTLR = H'03、UIER3 = H'80、UIFR3 = H'00

14.8.9 EP0 相关中断源的分配

分配到本模块的 UIFR0 的 EP0 相关中断源（位 3 ~ 位 0），必须要通过 UISR0 分配到相同中断信号（EXIRQ_x）。对其它中断源没有特别制约。

14.8.10 关于 VBUS、 $\overline{\text{IRQ}}_x$ 管脚的电平转移

本模块的 VBUS、 $\overline{\text{IRQ}}_x$ 管脚必须通过电平转移连接在 USB 连接器的 VBUS 管脚上。因为它是在本模块内部检测 USB 线的连接 / 断开的运行电路。

即使在本模块上搭载的产品电源为 OFF 状态，USB 线连接在产品上时，会给 USB 连接器的 VBUS 管脚输入约 5V 的电压。在此情况下，为了不破坏搭载本模块的 LSI，在电源 OFF 时，请使用可向管脚输入电压的电平转移器（HD74LV-A 系列等）。

14.8.11 USB 终点数据寄存器的读取 / 写入

在 CPU 通过字传送命令或长字传送命令向发送方的 USB 终点数据寄存器 (UEDR0i、UEDR1、UEDR3) 写入数据时, 请不要写入多于应该发送的数据大小。

向主机发送 7 字节的数据时, 用长字传送指令写入 2 次或者用字传送指令写入 4 次, 就会向主机传送 8 字节的数据。因此为了正确写入 7 字节的数据, 用长字传送指令写入 1 次, 用字传送指令写入 1 次, 用字节传送指令写入 1 次或者用字传送指令写入 3 次, 用字节传送指令写入 1 次。

并且, 读取接收方的 USB 终点数据寄存器 (UEDR0o、UEDR2) 的数据时, 请仅正确读取 USB 终点接收大小寄存器 (UESZ0o、UESZ2) 中所示的字节数。

通过内置 DMAC 将 USB 终点数据寄存器的数据进行 DMA 传送时, 务必设定为字节长度传送。由于字长度传送, 不支持奇数字节的 DMA 传送, 所以请不要设定。

14.8.12 向各低功耗模式转换和解除各低功耗模式时的限制事项

向各低功耗模式转换前，请设定为 USB 模块停止 2 状态。并且请不要复位 UDC 内核。

解除各低功耗模式，要再次访问本模块时，解除 USB 模块停止 2 状态后，请等待 USB 运行时钟 (48MHz) 的运行稳定时间。

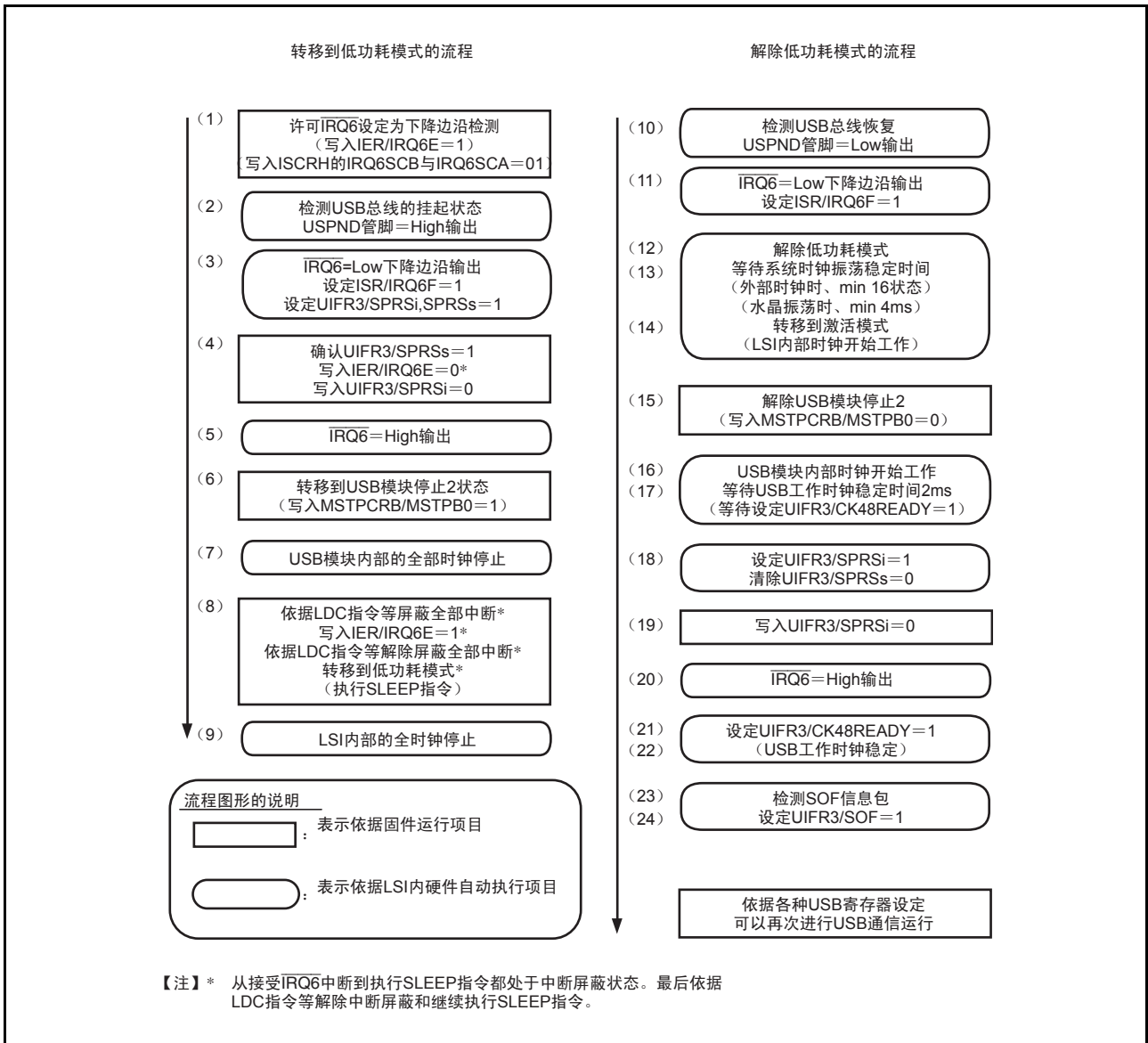


图 14.28 流程图

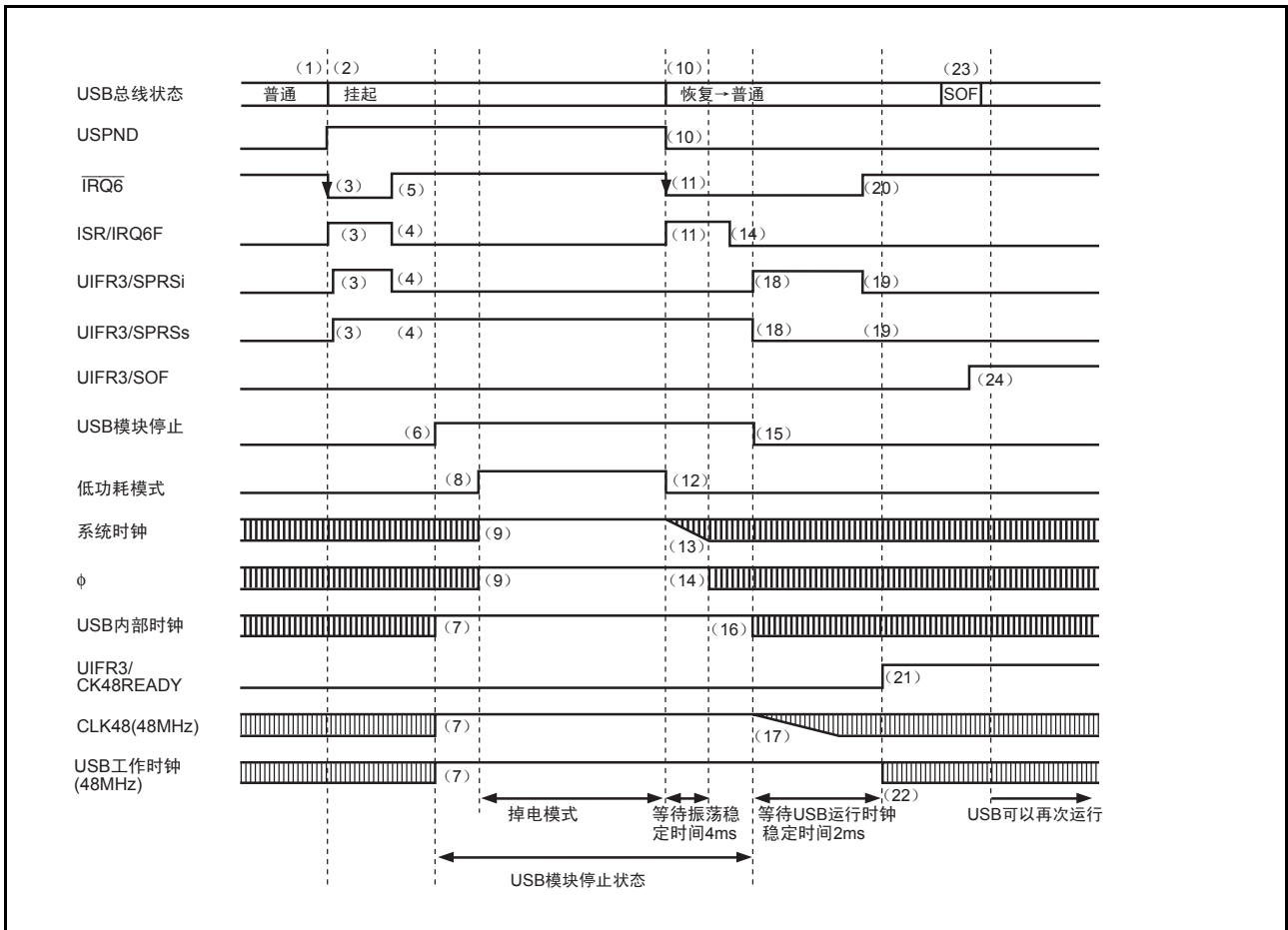


图 14.29 时序图

14.8.13 关于 USB 外部电路

本 USB 外部电路例只是参考例，在充分确认是否没有系统问题后，再进行电路板设计。

本电路例不能保证正确运行。并且，来自外部的电涌以及系统需要 ESD 噪声对策时，请通过保护二极管和噪声取消电路作为对策。

14.8.14 未使用 USB 时的管脚处理

请如下进行处理：

$DrV_{cc} = V_{cc}$ 、 $DrV_{ss} = 0V$ 、 $USD+ = USD- = USPND =$ 开路状态、 $\overline{VBUS} = \overline{UBPM} = 0V$

14.8.15 使用 TR 中断的注意事项

在 EP0i/EP1/EP3 的 in 传送中有传送请求中断 (TR 中断)，使用本中断时请注意以下几点：

从 USB 主机传来 IN 权标时，相应终点的 FIFO 没有数据时 TR 中断标志也会被置位。但是，图 14.30 所示的时序会连续发生 TR 中断。即使在这种情况下，也请不要发生误操作。

【注】 本模块接收 IN 权标时，相应 EP 的 FIFO 中没有数据，则进行 NAK 判定，TR 中断标志在 NAK 信号交换发送后被置位。因此，UTRG0/PKTE 的写入因下一个 IN 权标而推迟时，TR 中断标志会被再次置位。

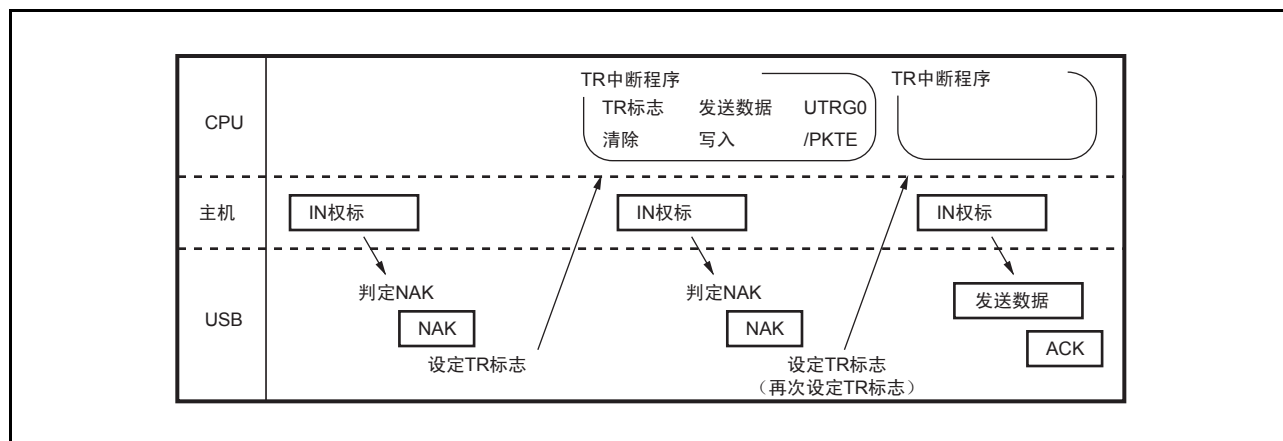


图 14.30 TR 中断标志的设置时序

15. A/D 转换器

本 LSI 内置逐次逼近型 10 位 A/D 转换器，包括 6 个可选模拟输入通道。A/D 转换器的框图如图 15.1 所示。

15.1 特点

- 分辨率：10 位
- 输入通道：6 通道
- 转换时间：平均 1 通道 8.4 μ s（16MHz 运行时）、11.08 μ s（24MHz 运行时）
- 运行模式：2 种
 - 单通道模式：1 个通道的 A/D 转换
 - 扫描模式：1~4 通道的连续 A/D 转换
- 数据寄存器：4 个
 - 每个通道的 A/D 转换结果保存于各通道对应的 16 位数据寄存器
- 采样保持功能
- A/D 转换开始方法：3 种
 - 软件
 - 定时器（TPU）的转换开始触发器
 - 外部触发器信号（ $\overline{\text{ADTRG}}$ ）
- 中断源
 - 能产生 A/D 转换完成中断请求（ADI）
- 可设定模块待机模式
- 可设定模拟转换电压范围
 - 将参考电压管脚（Vref）作为模拟基准电压，设定模拟转换电压范围

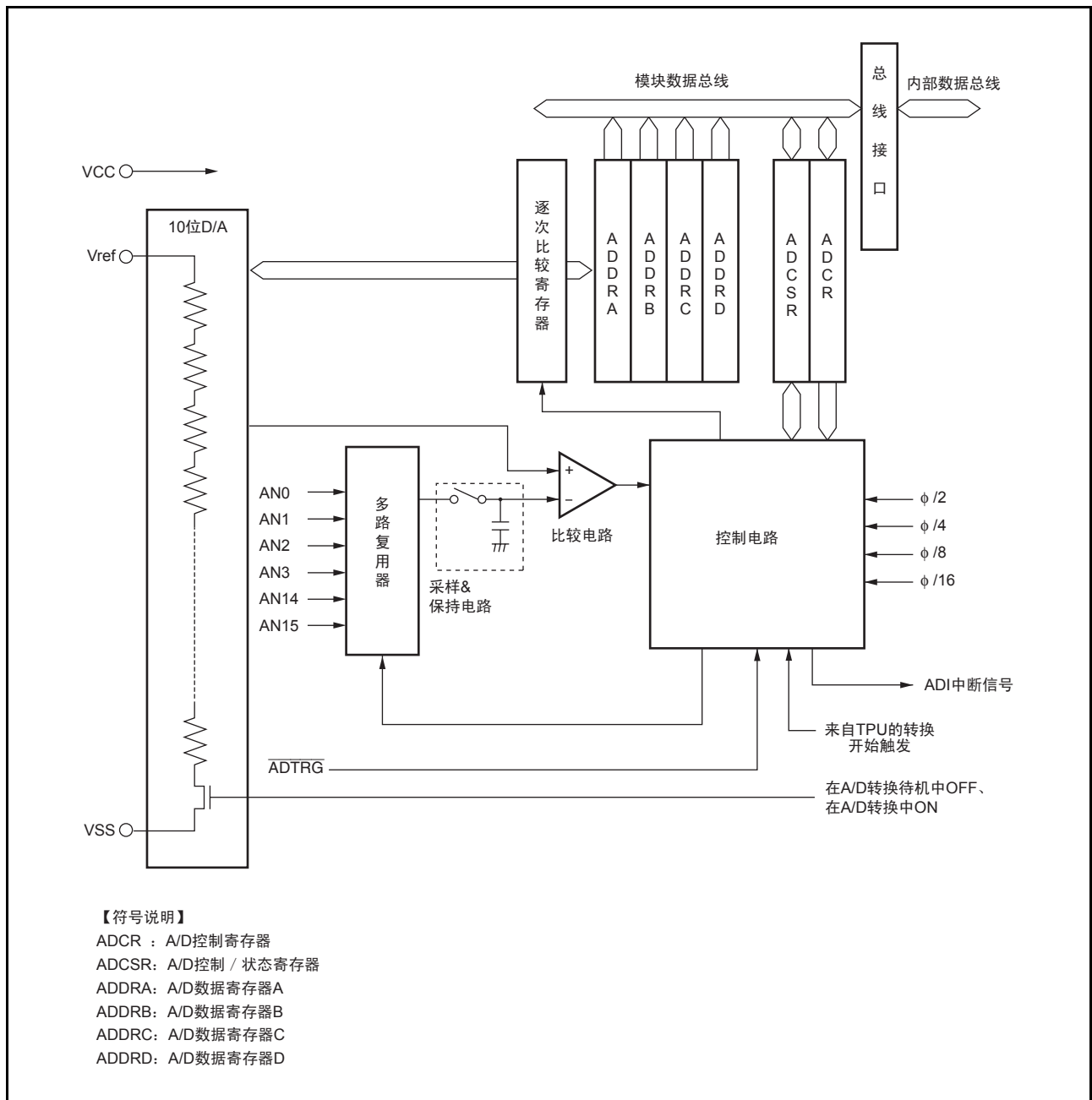


图 15.1 A/D 转换器框图

15.2 输入 / 输出管脚

A/D 转换器中使用的管脚如表 15.1 所示。AN0 ~ AN3、AN14、AN15 是模拟输入管脚。VCC、VSS 管脚是 A/D 转换器内部模拟部分的电源。Vref 是 A/D 转换基准电压管脚。

表 15.1 管脚构成

管脚名称	符号	输入输出	功能
电源管脚	VCC	输入	模拟部分的电源管脚及基准电压（与数字电路部分共用）
接地管脚	VSS	输入	模拟部分的接地及基准电压（与数字电路部分共用）
参考电压管脚	Vref	输入	A/D 转换的基准电压
模拟输入管脚 0	AN0	输入	模拟输入管脚
模拟输入管脚 1	AN1	输入	
模拟输入管脚 2	AN2	输入	
模拟输入管脚 3	AN3	输入	
模拟输入管脚 14	AN14	输入	
模拟输入管脚 15	AN15	输入	
A/D 外部触发器输入管脚	$\overline{\text{ADTRG}}$	输入	

15.3 寄存器说明

A/D 转换器中有以下寄存器。

- A/D 数据寄存器 A (ADDRA)
- A/D 数据寄存器 B (ADDRB)
- A/D 数据寄存器 C (ADDRC)
- A/D 数据寄存器 D (ADDRD)
- A/D 控制 / 状态寄存器 (ADCSR)
- A/D 控制寄存器 (ADCR)

15.3.1 A/D 数据寄存器 A ~ D (ADDRA ~ ADDRD)

ADDR 是保存 A/D 转换结果的 16 位只读寄存器，有 ADDRA ~ ADDRD 4 个。保存各模拟输入管脚转换结果的 ADDR 如表 15.2 所示。

10 位转换数据被保存在 ADDR 的位 15 到位 6。位 5 ~ 位 0 的读取值总为 0。

A/D 转换器与 CPU 间的数据总线为 8 位宽。可从 CPU 直接读取高位字节，而低位字节是通过临时寄存器读取的，读取高位字节时，低位字节的数据传送到临时寄存器。读取 ADDR 时，必须以字为单位读取或只读高位字节。

ADDR 的初始值是 H'0000。

表 15.2 模拟输入通道与 ADDR 的对照

模拟输入通道	保存转换结果的 A/D 数据寄存器
AN0	ADDRA
AN1	ADDRB
AN2、AN14	ADDRC
AN3、AN15	ADDRD

15.3.2 A/D 控制 / 状态寄存器 (ADCSR)

ADCSR 控制 A/D 转换运行。

位	位名	初始值	R/W	说明
7	ADF	0	R/(W)*	A/D 结束标志 表示 A/D 转换结束的状态标志。 [置位条件] • 在单通道模式, A/D 转换结束时 • 在扫描模式, 可选择的所有通道的 A/D 转换结束时 [清除条件] • 读取到 1 的状态后, 写入 0 时 • 通过 ADI 中断启动 DMAC, 读取 ADDR 时
6	ADIE	0	R/W	A/D 中断允许 如果该位置 1, 就允许 ADF 产生 ADI 中断请求。
5	ADST	0	R/W	A/D 开始 该位清零后, 中止 A/D 转换, 变为待机状态。该位置 1, 开始 A/D 转换。 可通过软件、定时器的转换开始触发器、A/D 外部触发器 ($\overline{\text{ADTRG}}$) 置位为 1。 在单通道模式下, 所选择通道的 A/D 转化结束后, 该位会自动清零。 扫描模式下, 该位在被软件、复位、软件待机或模块停止模式清除前, 依次将被选择的通道进行连续转换。
4	SCAN	0	R/W	扫描模式 选择 A/D 转换的运行模式。在 A/D 转换停止时 (ADST = 0), 设定位。 0: 单通道模式 1: 扫描模式
3 2 1 0	CH3 CH2 CH1 CH0	0 0 0 0	R/W R/W R/W R/W	通道选择 3 ~ 0 选择模拟输入通道。 SCAN = 0 时 0000: AN0 0001: AN1 0010: AN2 0011: AN3 01xx: 禁止设定 10xx: 禁止设定 110x: 禁止设定 1110: AN14 1111: AN15 SCAN = 1 时 0000: AN0 0001: AN0、AN1 0010: AN0 ~ AN2 0011: AN0 ~ AN3 01xx: 禁止设定 1xxx: 禁止设定 【符号说明】x: Don't care

【注】 * 为清除标志只能写入 0。

15.3.3 A/D 控制寄存器 (ADCR)

ADCR 允许通过外部触发器开始 A/D 转换。

位	位名	初始值	R/W	说明
7 6	TRGS1 TRGS0	0 0	R/W R/W	定时触发选择 1、0 通过外部触发信号，允许开始 A/D 转换。在 A/D 转换停止时 (ADST=0) 置位。 00: 通过软件，开始 A/D 转换 01: 通过来自 TPU 的转换触发开始 A/D 转换 10: 禁止设定 11: 通过外部触发器管脚 (ADTRG) 开始 A/D 转换
5、4	—	全为 1	—	保留位 读取时读为 1。写入定义为无效。
3 2	CKS1 CKS0	0 0	R/W R/W	时钟选择 1、0 设定 A/D 转换时间。在 A/D 转换停止时 (ADST=0)，切换 A/D 转换时间。 00: 转换时间 = 530 状态 (max) 01: 转换时间 = 266 状态 (max) 10: 转换时间 = 134 状态 (max) 11: 转换时间 = 68 状态 (max) 请将转换时间设定为大于 22.6 A/D 转换特性中所示的转换时间。
1、0	—	全为 1	—	保留位 读取时常读出 1。写入时也请写入 1。

15.4 与总线主控器的接口

ADDRA ~ ADDR4 是 16 位寄存器，与总线主控器之间的数据总线为 8 位宽。因此，从总线主控器可直接存取高位字节，但是低位字节的存取要通过 8 位的临时寄存器（TEMP）来进行。

按照如下读取来自 ADDR 的数据：通过读取高位字节，将高位字节值传送到 CPU，低位字节的值传送到 TEMP。接着，通过读取低位字节，TEMP 的内容就被传送到 CPU。

读取 ADDR 时，请务必按照高位字节、低位字节的顺序进行读取。并且，仅能读取高位字节，如果仅读取低位字节，不能保证读取内容，请注意。

图 15.2 所示的是 ADDR 存取时的数据流。

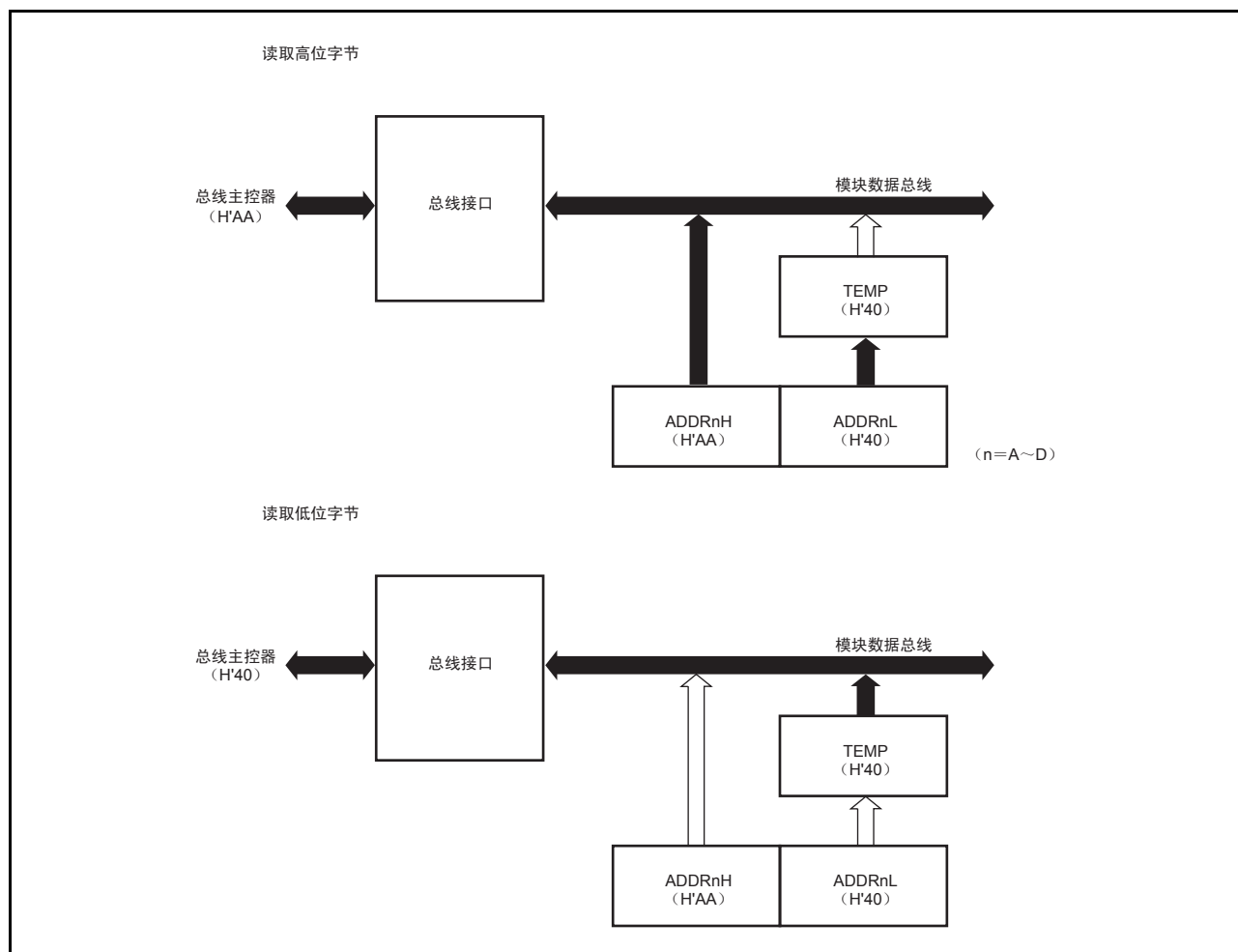


图 15.2 ADDR 的存取运行（读取 H'AA40 时）

15.5 运行说明

A/D 转换器采用逐次逼近方式，其分辨率是 10 位。其运行模式有单通道模式和扫描模式。切换运行模式或模拟输入通道时，为了避免发生误操作，先将 ADCSR 的 ADST 位清零；运行模式或模拟输入通道的变更和 ADST 位的置位可同时进行。

15.5.1 单通道模式

单通道模式是将指定的 1 个通道的模拟输入进行 1 次如下的 A/D 转换。

1. 如果通过软件、定时器或外部触发器输入，将 ADCSR 的 ADST 位置 1，被选择的通道开始 A/D 转换。
2. A/D 转换结束后，A/D 转换结果被传送到该通道对应的 A/D 数据寄存器。
3. A/D 转换结束后，ADCSR 的 ADF 位被置为 1。此时，如果 ADIE 位被置为 1，则会产生 ADI 中断请求。
4. ADST 位在 A/D 转换过程中保持为 1，并且在转换结束后自动被清除，A/D 转换器变为待机状态。

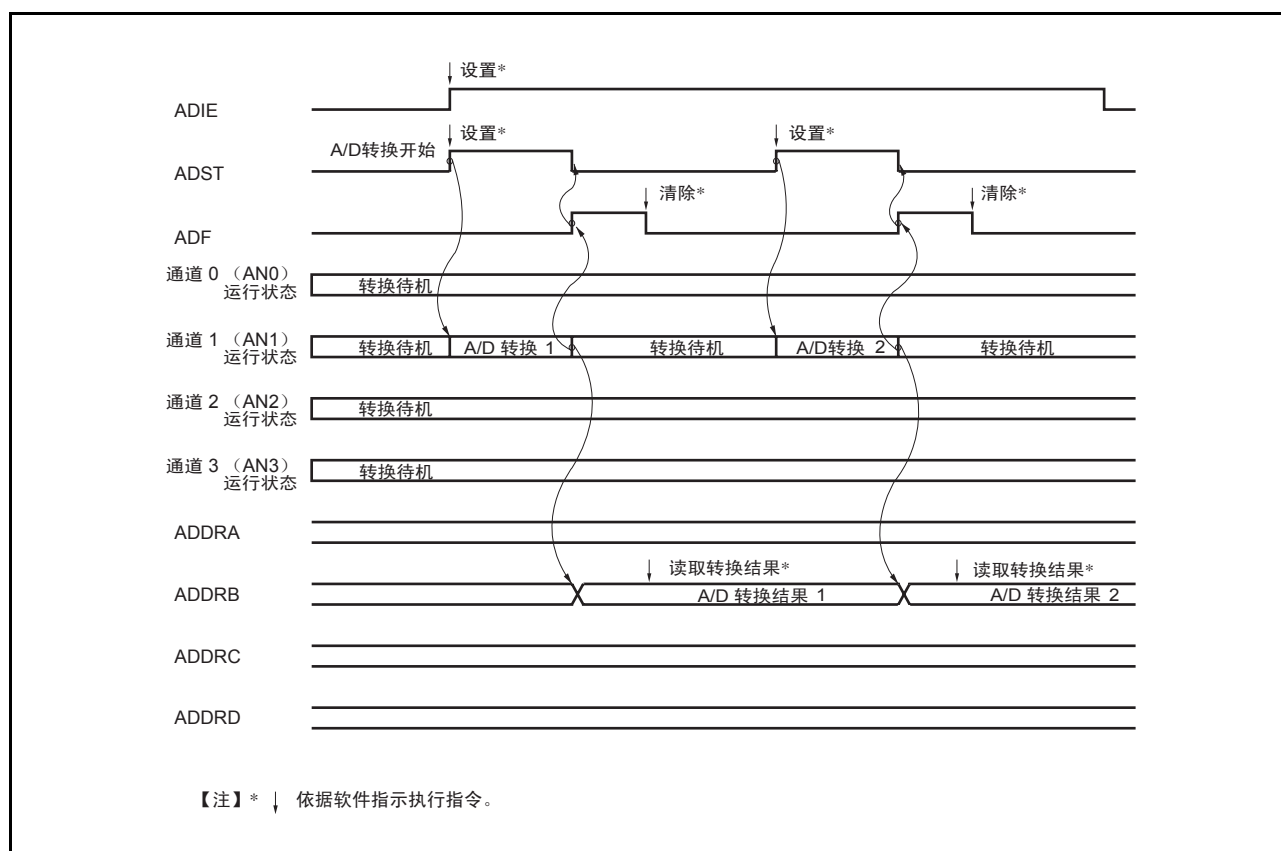


图 15.3 A/D 转换器的运行（单通道模式 选择通道 1 时）

15.5.2 扫描模式

扫描模式是将指定的最多 4 个通道的模拟输入按照如下操作连续进行 A/D 转换。

1. 如果通过软件、定时器或外部触发器输入，将 ADCSR 的 ADST 位置 1，就从组里的第 1 个通道 (AN0) 开始 A/D 转换。
2. 各通道的 A/D 转换结束后，A/D 转换结果被依次传送到每个通道对应的 A/D 寄存器。
3. 如果选择的所有通道的 A/D 转换均结束，ADCSR 的 ADF 标志就被置 1。这时如果 ADIE 位被置为 1，就会产生 ADI 中断请求。A/D 转换器会再次从组里的第一通道开始 A/D 转换。
4. ADST 位不被清除，保持为 1 时，A/D 转换器重复 2.~3. 的操作。如果将 ADST 位清零，将会停止 A/D 转换。

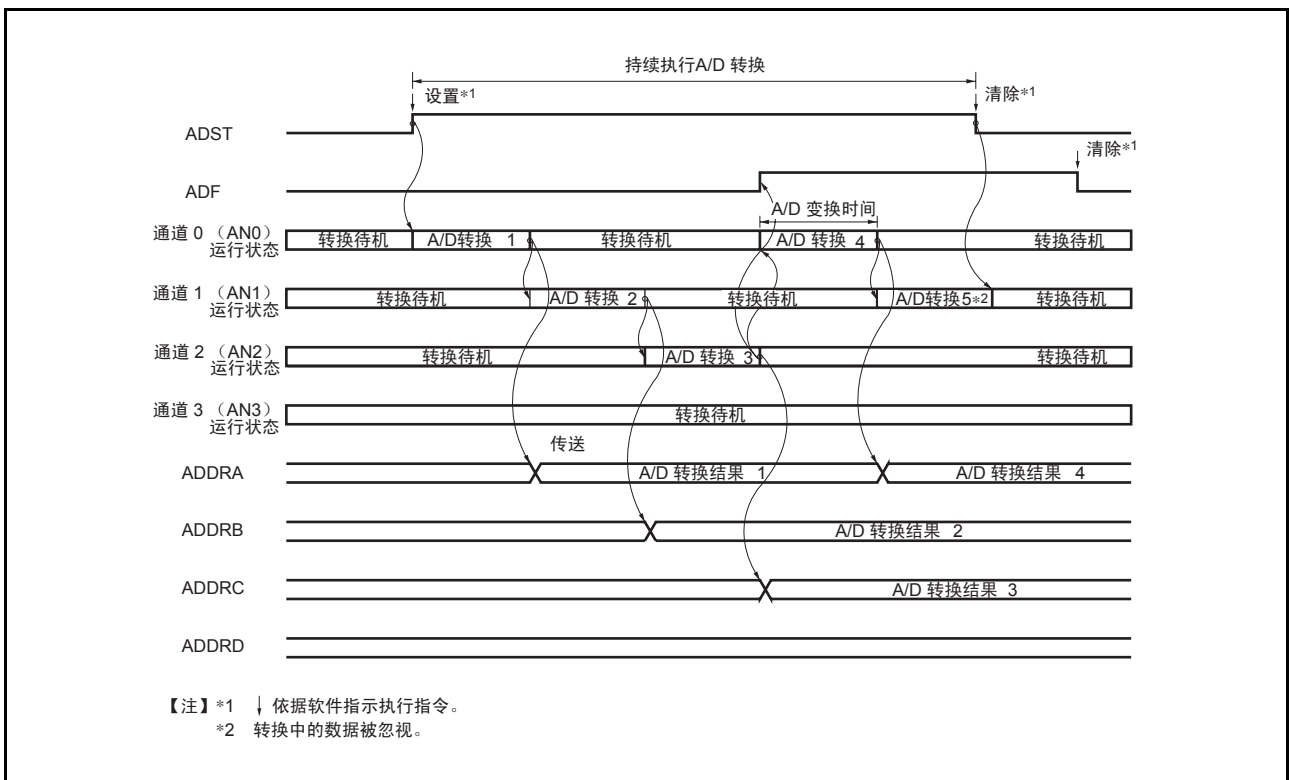


图 15.4 A/D 转换器的运行（扫描模式 选择 AN0 ~ AN2 的 3 通道时）

15.5.3 输入采样和 A/D 转换时间

A/D 转换器中内置采样保持电路。将 ADCSR 的 ADST 位置为 1 后，经过 A/D 转换开始延迟时间 (t_D)，A/D 转换器对输入进行采样后才开始转换。A/D 转换的时序如图 15.5 所示，A/D 转换时间如表 15.3 和表 15.4 所示。

如图 15.5 所示，A/D 转换时间 (t_{CONV}) 包含 t_D 和输入采样时间 (t_{SPL})。这里的 t_D 由对 ADCSR 的写入时序决定，不是固定值。因此转换时间在如表 15.4 所示的范围内变化。

扫描模式的转换时间是表 15.3 所示的值为第 1 次的转换时间，第 2 次以后的转换时间则为表 15.4 所示的值。

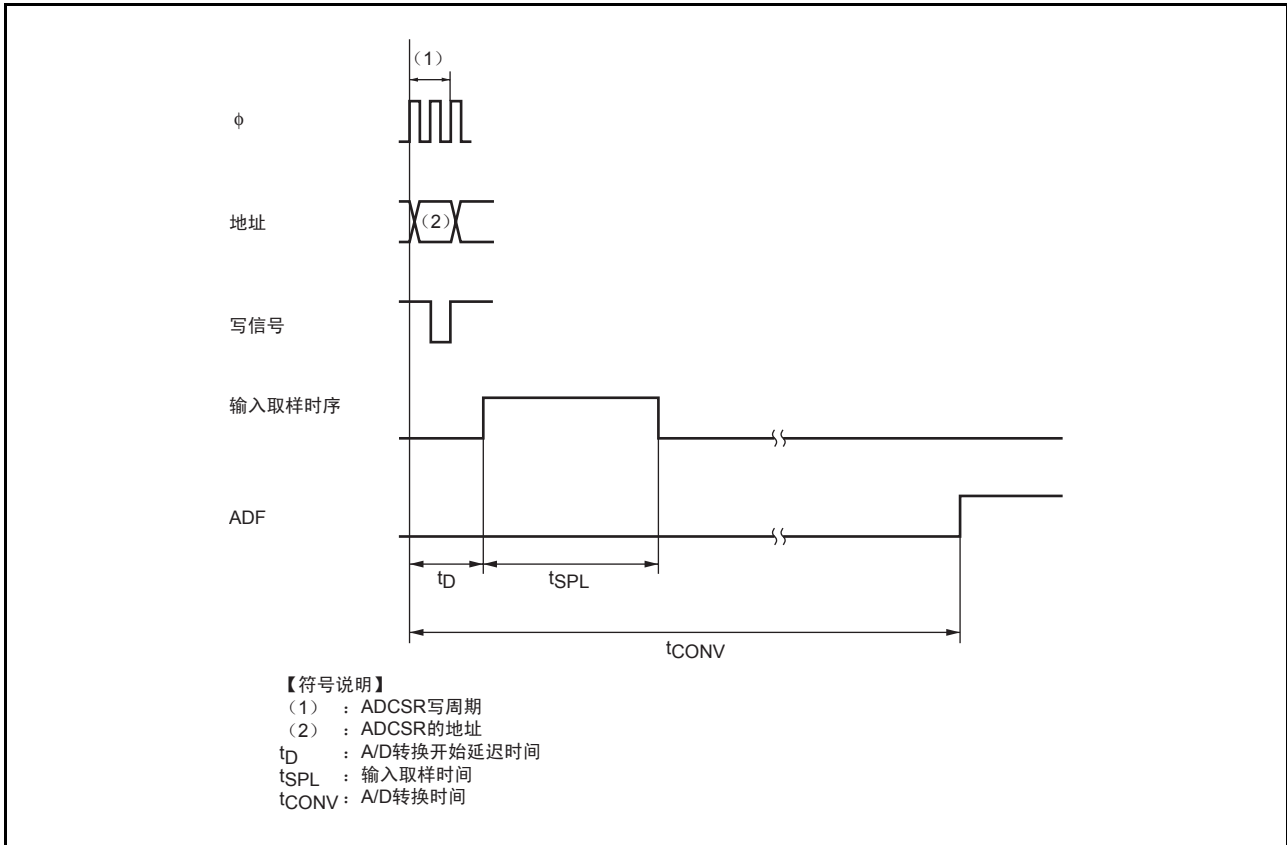


图 15.5 A/D 转换时序

表 15.3 A/D 转换时间（单通道模式）

项 目	符 号	CKS1 = 0						CKS1 = 1					
		CKS0 = 0			CKS0 = 1			CKS0 = 0			CKS0 = 1		
		min	typ	max	min	typ	max	min	typ	max	min	typ	max
A/D 转换开始延迟时间	t_D	18	—	33	10	—	17	6	—	9	4	—	5
输入采样时间	t_{SPL}	—	127	—	—	63	—	—	31	—	—	15	—
A/D 转换时间	t_{CONV}	515	—	530	259	—	266	131	—	134	67	—	68

【注】 表中数值单位是 state。

表 15.4 A/D 转换时间（扫描模式）

CKS1	CKS0	转换时间 (state)
0	0	512 (固定)
	1	256 (固定)
1	0	128 (固定)
	1	64 (固定)

15.5.4 外部触发器输入时序

也可通过外部触发器的输入，开始 A/D 转换。ADCR 的 TRGS1、TRGS0 位被置为 11 时，会从 $\overline{\text{ADTRG}}$ 管脚引入外部触发信号。在 $\overline{\text{ADTRG}}$ 下降沿时，ADCSR 的 ADST 位置为 1，开始 A/D 转换。无论是在单通道模式还是扫描模式，其他的运行都与通过软件将 ADST 位置 1 的运行情况相同。此时序如图 15.6 所示。

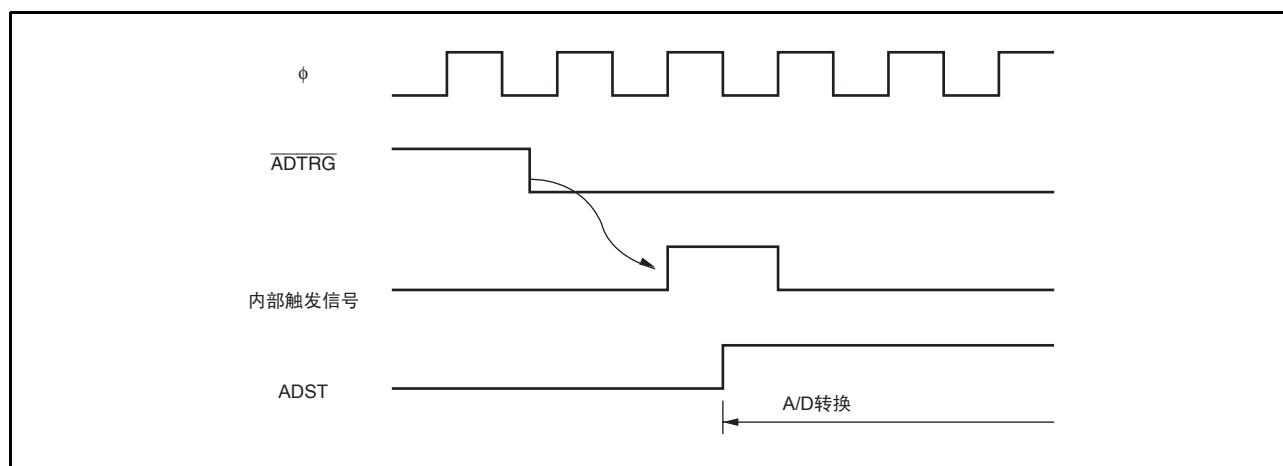


图 15.6 外部触发器输入时序

15.6 中断源

在 A/D 转换结束后，A/D 转换器会产生 A/D 转换结束中断（ADI）。在 A/D 转换结束后，ADCSR 的 ADF 会被置 1，这时如果 ADIE 置为 1，ADI 中断请求就会被允许。ADI 中断可以激活 DMAC。

表 15.5 A/D 转换器的中断源

名称	中断源	中断标志	DMAC 启动
ADI	A/D 转换结束	ADF	可

15.7 A/D 转换精度的定义

本 LSI 的 A/D 转换精度定义如下。

- 分辨率
A/D 转换器的数字输出码位数
- 量化误差
它是 A/D 转换器的固有偏差，在 1/2 LSB 中的偏差（图 15.7）
- 偏移误差
数字输出从最小电压值 B'000000000 (H'000) 变为 B'000000001 (H'001) 时，模拟输入电压值与理想 A/D 转换特性的偏差（图 15.8）
- 满刻度误差
数字输出从 B'111111110 (H'3FE) 变为 B'111111111 (H'3FF) 时，模拟输入电压值与理想 A/D 转换特性的偏差（图 15.8）
- 非线性误差
从零电压到满刻度电压之间的理想 A/D 转换特性的误差。不包含偏移误差、满刻度误差、量化误差（图 15.8）。
- 绝对精度
数字值和模拟输入值之间的误差。包含偏移误差、满刻度误差、量化误差及非线性误差。

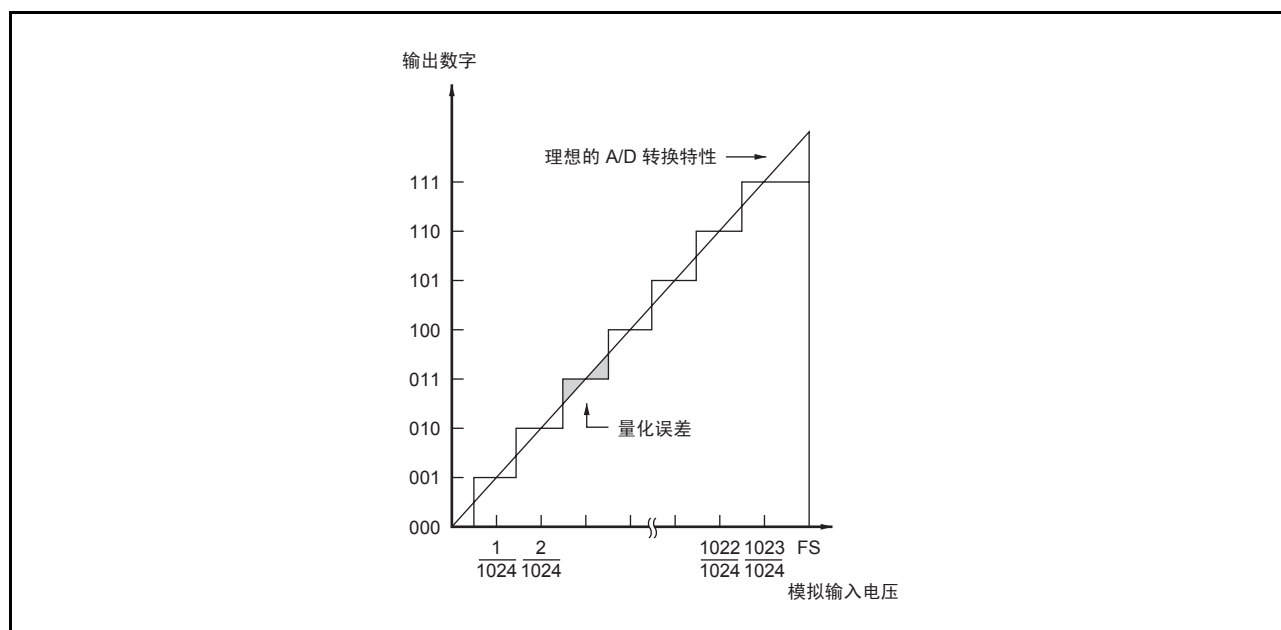


图 15.7 A/D 转换精度的定义 (1)

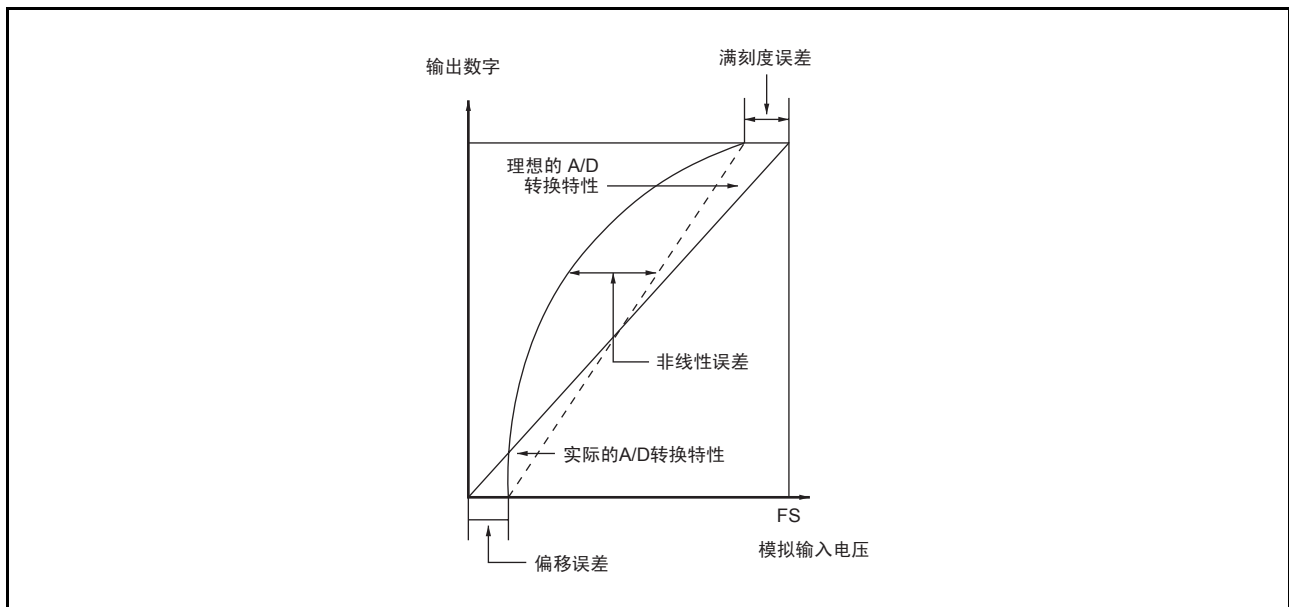


图 15.8 A/D 转换精度的定义 (2)

15.8 使用时的注意事项

15.8.1 模块停止模式的设定

通过模块停止控制寄存器，可设定 A/D 转换器运行的禁止 / 允许。为初始值时，A/D 转换器的运行停止。通过解除模块停止模式，可进行寄存器的存取。详细内容请参考 20. 低功耗状态。

15.8.2 关于容许信号源阻抗

针对信号源阻抗为 $5k\ \Omega$ 以下的输入信号，本 LSI 的模拟输入端设计可保证其转换精度。这是在采样时间内，对 A/D 转换器的采样保持电路的输入电容进行充电所制定的规格，传感器的输出阻抗超过 $5k\ \Omega$ 时，会出现充电不足，而不能保证 A/D 转换精度的情况。在单通道模式进行转换时，如果外部配置一个大容量电容，输入负载实质上只是 $10k\ \Omega$ 的内部输入电阻，因此可忽略信号源的阻抗，但是，在此情况下，会形成低通滤波器，有时会无法跟踪微分系数大的模拟输入信号（例如电压变动率大于 $5mV/\mu s$ ）（图 15.9）。在转换高速模拟信号或在扫描模式进行转换时，必须插入一个低阻抗的缓冲器。

15.8.3 对绝对精度的影响

由于附加电容，会与 GND 耦合，因此如果 GND 中有噪声，就可能会降低绝对精度，必须与 AVSS 等电位稳定的 GND 连接。

并且，要注意，在实装电路板上，滤波器电路不要干扰数字信号也不要充当天线。

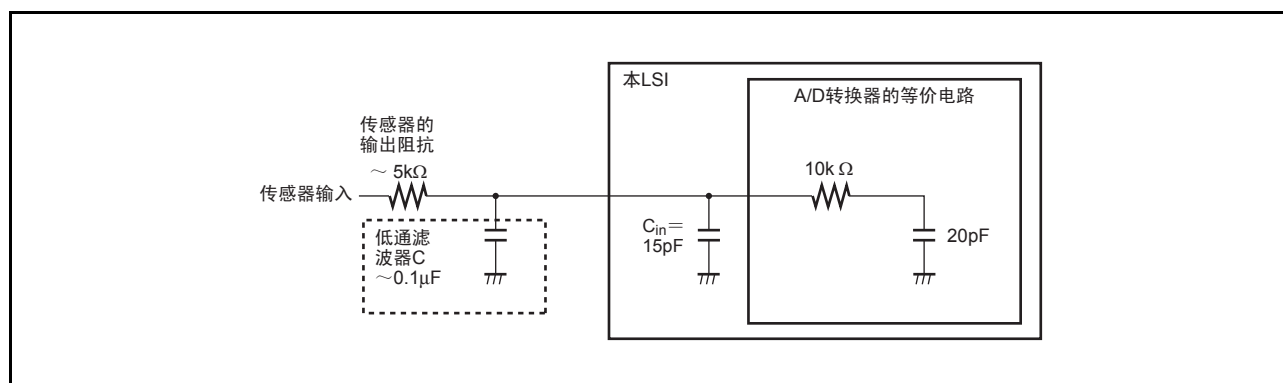


图 15.9 模拟输入电路

15.8.4 模拟电源管脚其他的设定范围

超过以下所示电压设定范围使用时，有时会给 LSI 的可靠性带来不良影响。

- 模拟输入电压的设定范围
A/D 转换中，向模拟输入管脚 ANn 输入的电压范围为 $VSS \leq ANn \leq Vref$ 。
- Vref 的设定范围
在 Vref 管脚的参考电压值的设定范围为 $Vref \leq VCC$ 。

15.8.5 电路板设计时的注意事项

为了确定噪声对策或防止过大电涌等异常电压而引起的模拟输入管脚（ANn）、模拟基准电压（Vref）的破坏，请充分讨论电路板的设计后再决定。

表 15.6 模拟管脚的规格

项目	min	max	单位
模拟输入容量	—	20	pF
容许信号源阻抗	—	5*	k Ω

【注】 * $V_{CC} = 2.7 \sim 3.6V$

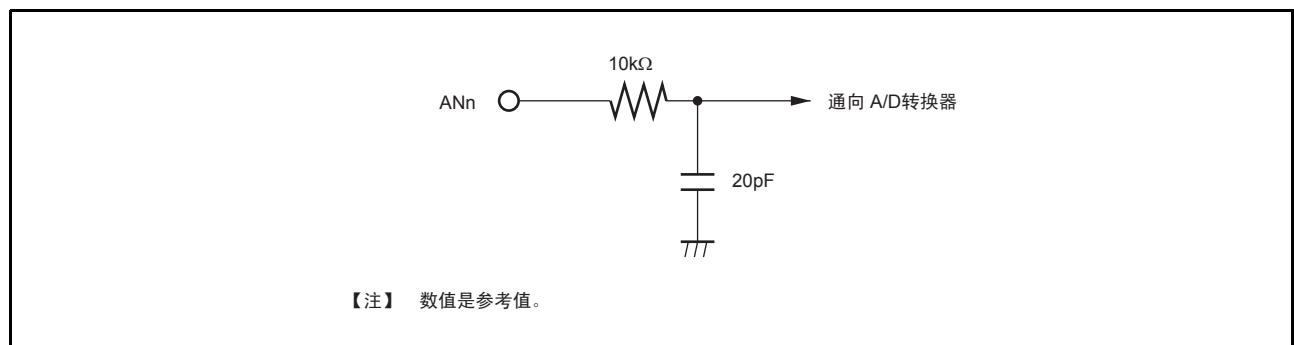


图 15.10 模拟输入管脚等价电路

16. ROM

H8S/2218 与 H8S/2212 内置 12K 字节、H8S/2217 与 H8S/2211 内置 8K 字节、H8S/2210 内置 4K 字节的高速静态 RAM。RAM 以 16 位宽度的数据总线与 CPU 相连接，无论字节数据、字数据，均可以 1 态存取。

可通过系统控制寄存器（SYSCR）的 RAME 位，控制 RAM 有效或无效。关于 SYSCR，请参考 3.2.2 系统控制寄存器（SYSCR）。

产品分类		ROM 类型	RAM 容量	RAM 地址
H8S/2218 群	HD64F2218 HD64F2218U	闪存版	12K 字节	H'FFC000 ~ H'FFEFBF H'FFFC0 ~ H'FFFFFF
	HD6432217	掩膜 ROM 版	8K 字节	H'FFD000 ~ H'FFEFBF H'FFFC0 ~ H'FFFFFF
H8S/2212 群	HD64F2212 HD64F2212U	闪存版	12K 字节	H'FFC000 ~ H'FFEFBF H'FFFC0 ~ H'FFFFFF
	HD64F2211 HD64F2211U		8K 字节	H'FFD000 ~ H'FFEFBF H'FFFC0 ~ H'FFFFFF
	HD6432211	掩膜 ROM 版	8K 字节	H'FFD000 ~ H'FFEFBF H'FFFC0 ~ H'FFFFFF
	HD6432210		4K 字节	H'FFE000 ~ H'FFEFBF H'FFFC0 ~ H'FFFFFF

17. 闪存（F-ZTAT 版）

闪存的框图如图 17.1 所示。闪存寄存器版内置闪存特点如下所示：

17.1 特点

- 容量：

产品分类		ROM 容量	ROM 地址
H8S/2218 群	HD64F2218、HD64F2218U	128K 字节	H'000000 ~ H'01FFFF（模式 6、7）
H8S/2212 群	HD64F2212、HD64F2212U		H'000000 ~ H'01FFFF（模式 7）
	HD64F2211、HD64F2211U	64K 字节	H'000000 ~ H'00FFFF（模式 7）

- 写入 / 擦除方式
写入为 128 字节单位同时写入方式。擦除以块为单位进行。闪存由 32K 字节×2 块、28K 字节×1 块、16K 字节×8 块、8K 字节×1 块、1K 字节×4 块所构成。进行全部擦除时，务请以块逐个擦除。
- 改写次数
可改写最少 100 次。
- 单板上编程模式：2 种
引导模式
SCI 引导模式：HD64F2218，HD64F2212，HD64F2211
USB 引导模式：HD64F2218U，HD64F2212U，HD64F2211U
用户程序模式
引导模式下，启动内部引导程序，通过进行全部擦除或者写入的引导模式，就可以在单板上写入/擦除。除此之外，用户程序模式下，也可以对任意块进行擦除和写入。
- 比特率自动匹配
SCI 引导模式下，传送数据时，主机的传送比特率与本 LSI 的比特率自动匹配。
- 写入/擦除保护
可给闪存的写入/擦除设置硬件保护、软件保护或错误保护。
- 编程器模式
除单板上编程外，还有使用 PROM 编程器进行编程/擦除的编程器模式。
- 内部 RAM 中的仿真功能
通过将闪存和内部 RAM 的一部分重叠，可实时仿真闪存的编程。

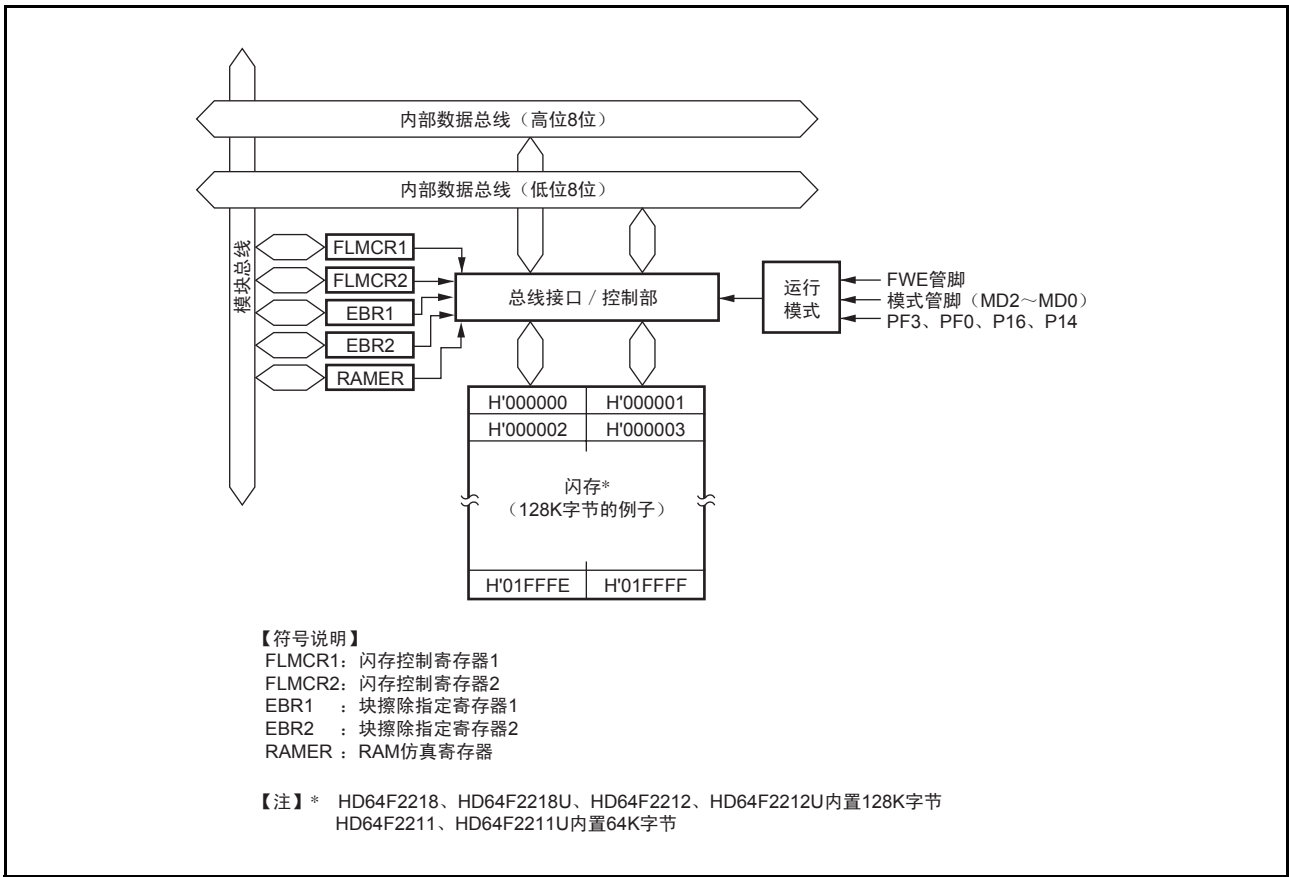


图 17.1 闪存的框图

17.2 模式转移图

在复位状态下，如果设置模式管脚和 FWE 管脚，并开始复位，则本 LSI 转移到如图 17.2 所示的工作模式。在用户模式下，虽然可以读取闪存，但不可以对闪存进行编程 / 擦除。引导模式、用户程序模式、编程器模式都可以对闪存进行写入 / 擦除。

引导模式与用户程序模式的不同点如表 17.1 所示。引导模式、用户程序模式分别如图 17.3、图 17.4 所示。

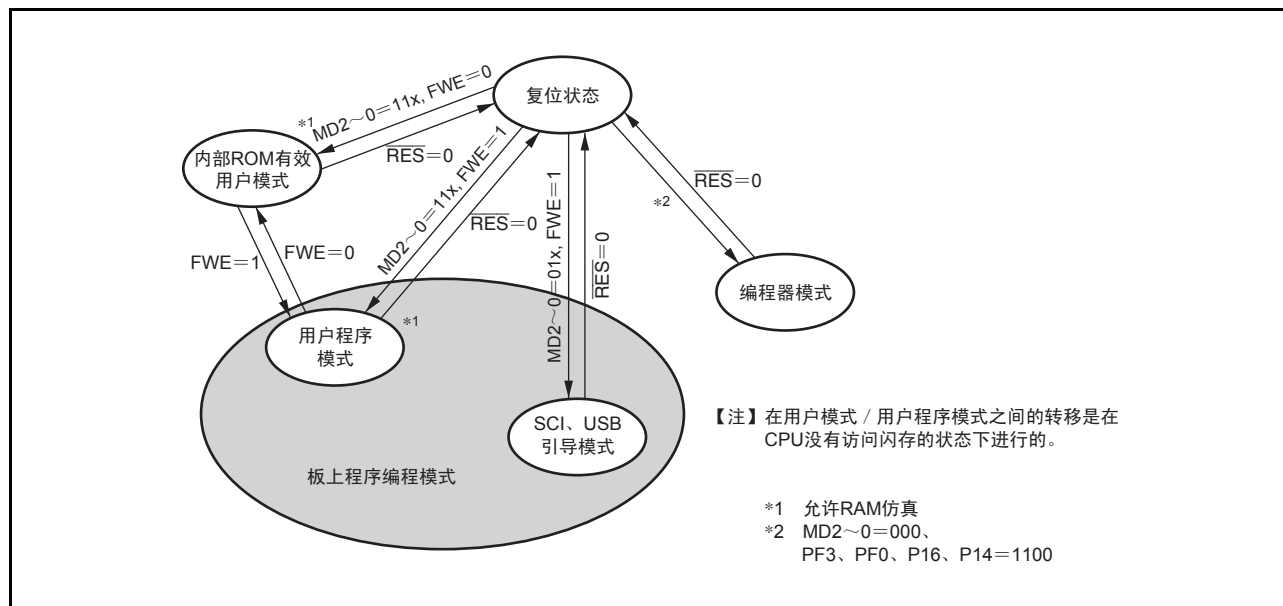


图 17.2 闪存的状态转移

表 17.1 引导模式与用户程序模式的不同点

	SCI、USB 引导模式	用户程序模式	用户模式
全部擦除	○	○	×
分块擦除	×	○	×
改写控制程序*	编程 / 编程验证	擦除 / 擦除验证 编程 / 编程验证 仿真	—

【注】* 用户必须按照推荐的算法准备。

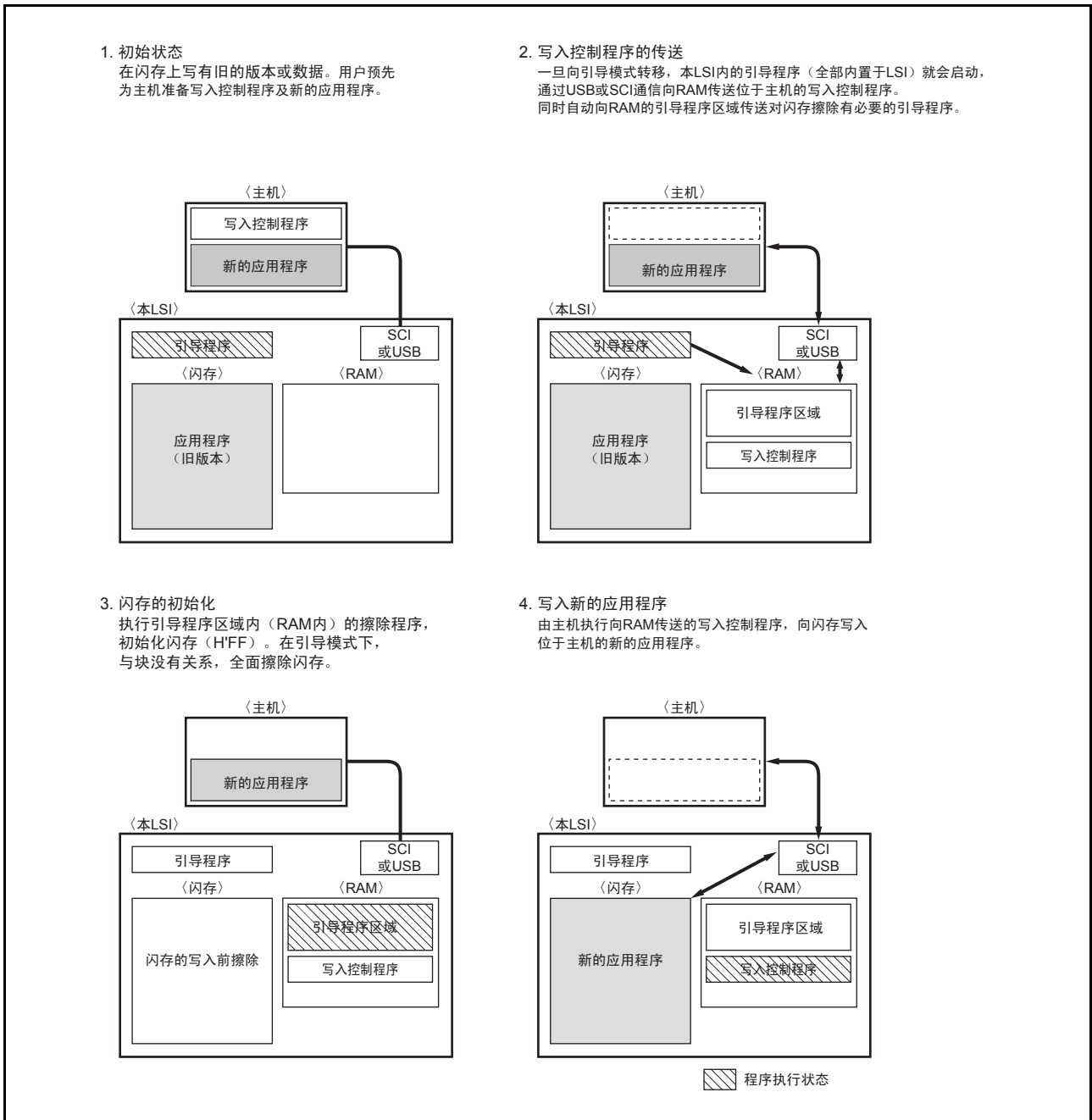


图 17.3 引导模式 (例)

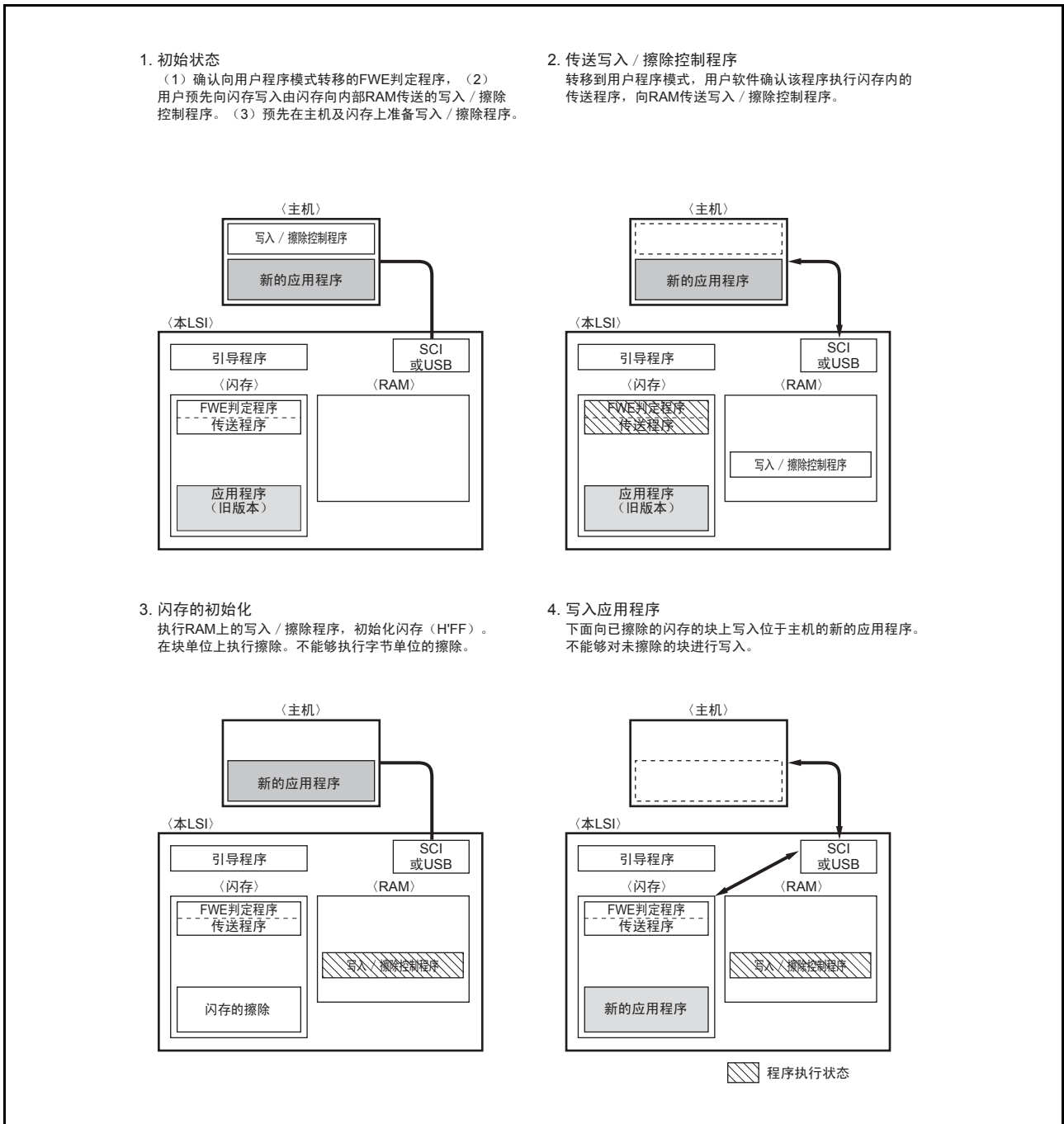


图 17.4 用户编程模式 (例)

17.3 块结构

HD64F2218、HD64F2218U、HD64F2212、HD64F2212U 的 128K 字节闪存框图如图 17.5 所示。粗线框表示擦除块。细线框表示写入单位，框内的数值表示地址。闪存被分成 1K 字节 (4 块)，28K 字节 (1 块)，16K 字节 (1 块)，8K 字节 (2 块)，32K 字节 (2 块)，以这些单位进行擦除。从低位地址为 H'00 或 H'80 的地址以 128 字节为单位开始写入。

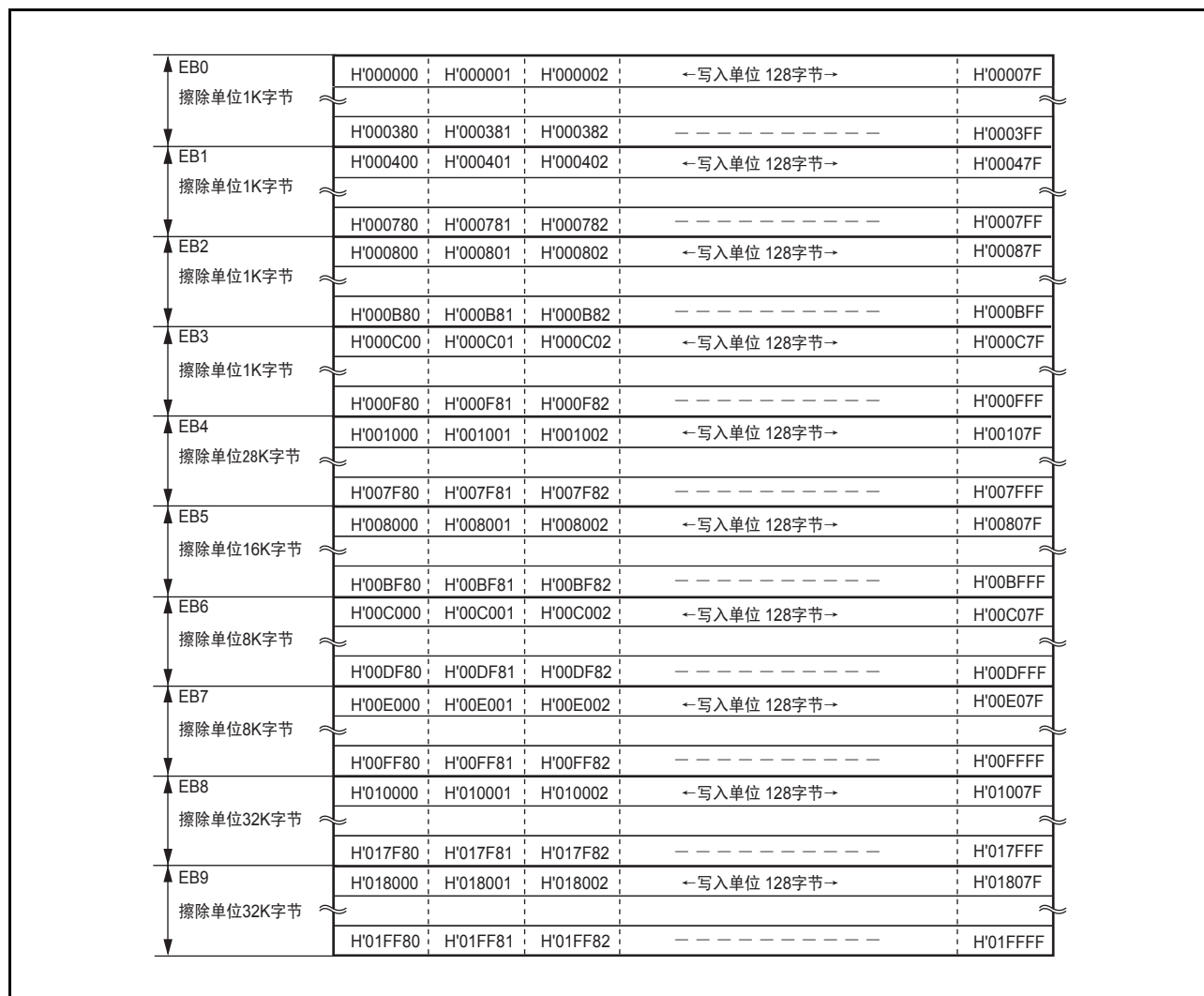


图 17.5 闪存的块结构

HD64F2211、HD64F2211U 的 64K 字节闪存的块结构如图 17.6 所示。粗线框表示擦除块。细线框表示写入单位。框内数值表示地址。闪存被分成 1K 字节 (4 块)、28K 字节 (1 块)、16K 字节 (1 块)、8K 字节 (2 块)，以这些单位进行擦除。从低位地址为 H'00 或 H'80 的地址以 128 字节为单位开始写入。

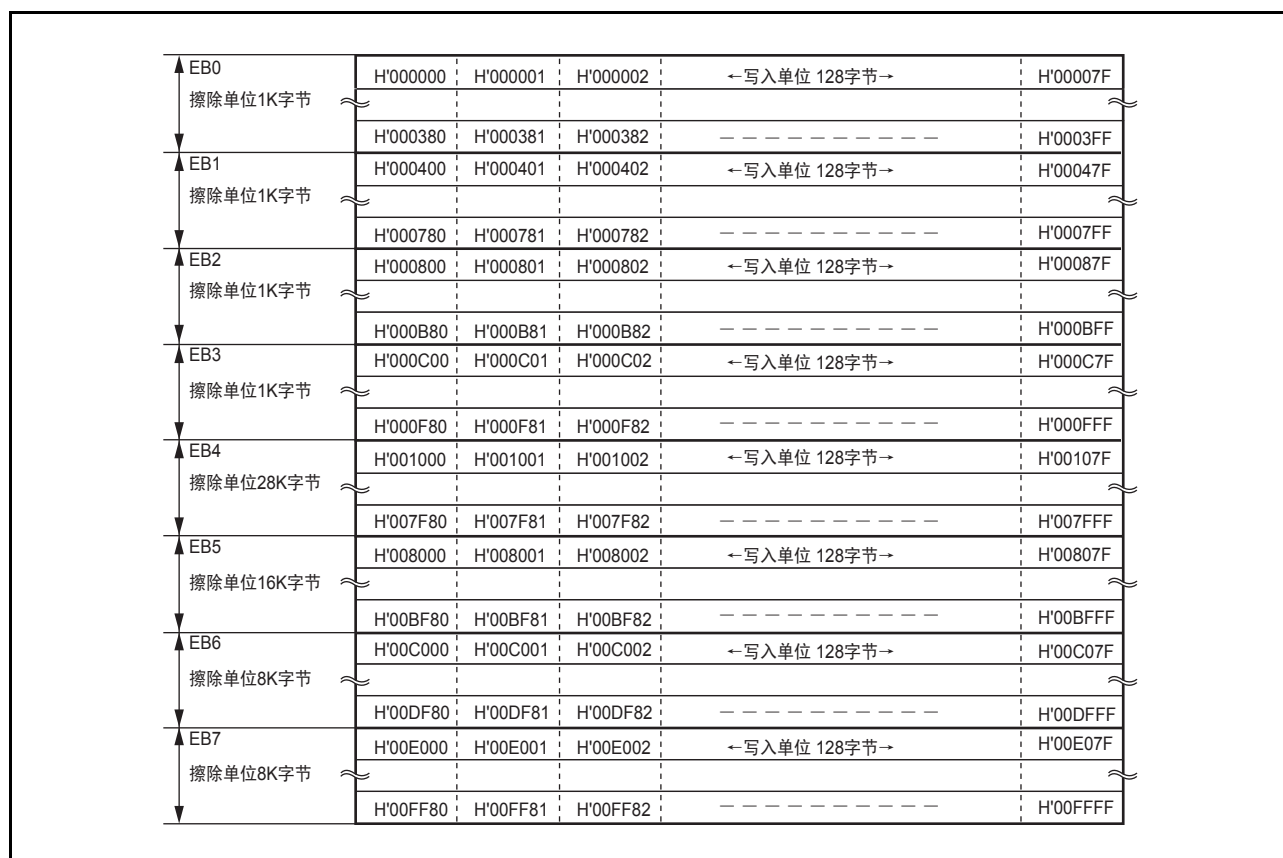


图 17.6 闪存的块结构

17.4 输入 / 输出管脚

通过表 17.2 所示的管脚控制闪存。

表 17.2 管脚结构

管脚名称	输入 / 输出	功能	
$\overline{\text{RES}}$	输入	复位	共通
FWE	输入	对闪存的写入 / 擦除进行硬件保护	
MD2、MD1、MD0	输入	设置工作模式	
PF3、PF0、P16、P14	输入	设置编程器模式的工作模式	
TxD2	输出	串行发送数据输出	HD64F2218 HD64F2212 HD64F2211
RxD2	输入	串行接收数据输入	
USD+、USD-	输入 / 输出	USB 数据输入 / 输出	HD64F2218U HD64F2212U HD64F2211U
VBUS	输入	USB 数据线的连接 / 切断检测	
$\overline{\text{UBPM}}$	输入	USB 总线电源模式 / 自身电源模式设置	
USPND	输出	USB 延缓输出	
P36 (PUPD+)	输出	D+ 上拉控制	

17.5 寄存器的说明

闪存有以下寄存器。

- 闪存控制寄存器1 (FLMCR1)
- 闪存控制寄存器2 (FLMCR2)
- 擦除块指定寄存器1 (EBR1)
- 擦除块指定寄存器2 (EBR2)
- RAM仿真寄存器 (RAMER)
- 串行控制寄存器X (SCRX)

掩模型 ROM 版中没有上述寄存器，因此，读取时，读出不定值。写入定义为无效。

17.5.1 闪存控制寄存器 1 (FLMCR1)

FLMCR1 使闪存转移到编程模式、编程验证模式、擦除模式、擦除验证模式。具体设置方法，可参考 17.8 闪存的编程 / 擦除。

位	位名	初始值	R/W	说明
7	FWE	—*	R	闪存编程允许 反映 FWE 管脚的输入电平。FWE 管脚为低电平时，则为 0。为高电平时，则为 1。为 0 时，进入硬件保护状态。
6	SWE1	0	R/W	软件编程允许 此位为 1 时，可进行闪存的编程 / 擦除。为 0 时，该寄存器的 5 ~ 0 位及 EBR1, EBR2 的各位不可置位。 [置位条件] FWE = 1 时
5	ESU1	0	R/W	擦除设置 如果置 1，则进入擦除设置状态。如果清除，则解除设置状态。必须在 E1 位置 1 前置位。 [置位条件] FWE = 1, SWE1 = 1 时
4	PSU1	0	R/W	编程设置 如果置 1，则进入编程设置状态，如果清除，则解除设置状态。必须在 P1 位置 1 前置位。 [置位条件] FWE = 1, SWE1 = 1 时
3	EV1	0	R/W	擦除验证 如果置 1，则转移到擦除验证模式，如果清除，则解除擦除验证模式。 [置位条件] FWE = 1, SWE1 = 1 时
2	PV1	0	R/W	编程验证 如果置 1，则转移到编程验证模式，如果清除，则解除编程验证模式。 [置位条件] FWE = 1, SWE1 = 1 时
1	E1	0	R/W	擦除 SWE1 = 1, ESU1 = 1 的状态下，如果此位置 1，则转移到擦除模式，如果清除，则解除擦除模式。 [置位条件] FWE = 1, SWE1 = 1, ESU1 = 1 时
0	P1	0	R/W	编程 SWE1 = 1, PSU1 = 1 的状态下，如果此位置 1，则转移到编程模式，如果清除，则解除编程模式。 [置位条件] FWE = 1, SWE1 = 1, PSU1 = 1 时

【注】 * 通过 FWE 管脚的状态设置。

17.5.2 闪存控制寄存器 2（FLMCR2）

FLMCR2 表示闪存的写入 / 擦除状态。FLMCR2 为读取专用寄存器，不可进行写入。

位	位名称	初始值	R/W	说明
7	FLER	0	R	在对闪存进行写入 / 擦除的过程中，检测出错误，并进入错误保护状态时，将此位置位。 详细情况请参考 17.9.3 错误保护。
6~0	—	全为 0	—	保留位 读取时，总是读出 0。

17.5.3 擦除块指定寄存器 1（EBR1）

EBR1 是指定闪存的擦除块的寄存器。FLMCR 的 SWE1 位为 0 时，将 EBR1 初始化为 H'00。该寄存器不可将包括 EBR2 在内的 2 个以上的位同时设置为 1。如果设置，则将 EBR1 和 EBR2 全部清 0。

位	位名	初始值	R/W	说明
7	EB7	0	R/W	此位为 1 时，EB7（H'00E000 ~ H'00FFFF）的 8K 字节变为擦除对象。
6	EB6	0	R/W	此位为 1 时，EB6（H'00C000 ~ H'00DFFF）的 8K 字节变为擦除对象。
5	EB5	0	R/W	此位为 1 时，EB5（H'008000 ~ H'00BFFF）的 16K 字节变为擦除对象。
4	EB4	0	R/W	此位为 1 时，EB4（H'001000 ~ H'007FFF）的 28 字节变为擦除对象。
3	EB3	0	R/W	此位为 1 时，EB3（H'000C00 ~ H'000FFF）的 1K 字节变为擦除对象。
2	EB2	0	R/W	此位为 1 时，EB2（H'000800 ~ H'000BFF）的 1K 字节变为擦除对象。
1	EB1	0	R/W	此位为 1 时，EB1（H'000400 ~ H'0007FF）的 1K 字节变为擦除对象。
0	EB0	0	R/W	此位为 1 时，EB0（H'000000 ~ H'00003FF）的 1K 字节变为擦除对象。

17.5.4 擦除块指定寄存器 2（EBR2）

EBR2 是指定闪存的擦除块寄存器。FLMCR1 的 SWE1 位为 0 时，将 EBR2 初始化为 H'00，该寄存器不可将包括 EBR1 在内的 2 个以上的位同时设置为 1。如果设置，则将 EBR1 及 EBR2 全部清除。

【注】 这些寄存器为 HD64F2211, HD64F2211U 所用。写入时，必须写入 H'00。

位	位名	初始值	R/W	说明
7~2	—	全为 0	R/W	保留位 写入时，必须写入 0。
1	EB9	0	R/W	此位为 1 时，EB 9（H'018000 ~ H'01FFFF）的 32K 字节变为擦除对象。
0	EB8	0	R/W	此位为 1 时，EB 8（H'010000 ~ H'017FFF）的 32K 字节变为擦除对象。

17.5.5 RAM 仿真寄存器 (RAMER)

在仿真闪存的实时改写时，RAMER 为设置闪存区域与 RAM 一部分重叠的寄存器。RAMER 的设置必须在用户模式或用户编程模式下进行。为了切实执行仿真功能，在本寄存器改写后，不可直接存取 RAM 仿真的对象 ROM。如果直接存取，则不能保证进行正常存取。具体设置方法请参照 17.7 RAM 中的闪存仿真。

位	位名	初始值	R/W	说明
7~4	—	全为 0	R/W	保留位 写入时，必须写入 0。
3	RAMS	0	R/W	RAM 选择 为 RAM 中的闪存仿真选择位。此位为 1 时，RAM 的一部分与闪存重叠，闪存进入全部块的编程 / 擦除保护状态。
2 1 0	RAM2 RAM1 RAM0	0 0 0	R/W R/W R/W	闪存区域选择 RAMS 为 1 时，选择与 RAM 区域重叠的那部分闪存区域。这些区与 1K 字节的擦除块相对应。 000: H'000000 ~ H'0003FF (EB0) 001: H'000400 ~ H'0007FF (EB1) 010: H'000800 ~ H'000BFF (EB2) 011: H'000C00 ~ H'000FFF (EB3) 100: 禁止设置 101: 禁止设置 110: 禁止设置 111: 禁止设置

17.5.6 串行控制寄存器 X (SCRX)

SCRX 执行寄存器存取的控制。

位	位名	初始值	R/W	说明
7~4	—	全为 0	R/W	保留位 写入时必须写入 0。
3	FLSHE	0	R/W	闪存控制寄存器允许 控制闪存的控制寄存器 (FLMCR1, FLMCR2, EBR1, EBR2) 的 CPU 存取。如果 FLSHE 位置 1，则可对闪存控制寄存器进行读取 / 写入。清 0 时，闪存的控制寄存器取消选定。这时，保持闪存控制寄存器的内容。 0: 地址 H'FFFFA8 ~ H'FFFFAC 的区域不可选择闪存控制寄存器 1: 地址 H'FFFFA8 ~ H'FFFFAC 的区域选择闪存控制寄存器
2~0	—	全为 0	R/W	保留位 写入时必须写入 0。

17.6 单板上编程模式

如果设置为单板上编程模式，则可对内部闪存进行写入、擦除、验证。本模式有 2 种工作模式（引导模式，用户编程模式）。各模式的设置方法请参照表 17.3。另外，向闪存的相关模式转移的状态转移图请参照图 17.2。

表 17.3 单板编程模式的设置方法

模式设置		EMLE	FWE	MD2	MD1	MD0
SCI 引导模式 (HD64F2218, HD64F2212, HD64F2211)	高级·单片模式	0	1	0	1	×
USB 引导模式 (HD64F2218U, HD64F2212U, HD64F2211U)	高级·单片模式 系统时钟 24MHz 时	0	1	0	1	0
	高级·单片模式 系统时钟 16MHz 时	0	1	0	1	1
用户编程模式	高级·内部 ROM 有效扩展模式 (MCU 工作模式 6)	0	1	1	1	0
	高级·单片模式 (MCU 工作模式 7)	0	1	1	1	1

17.6.1 SCI 引导模式 (HD64F2218、HD64F2212、HD64F2211)

如果将本 LSI 的管脚设置为引导模式后开始复位，则启动事先装在单片机内的引导程序，并使用 SCI 将主机准备的编程控制程序依次发送到本 LSI。本 LSI 将通过 SCI 接收到的编程控制程序写入内部 RAM 的编程控制程序区。传送结束后，转移到编程控制程序区的起始地址，并进入编程控制程序执行状态。（执行闪存的编程）。

引导模式时的系统结构图如图 17.7 所示。

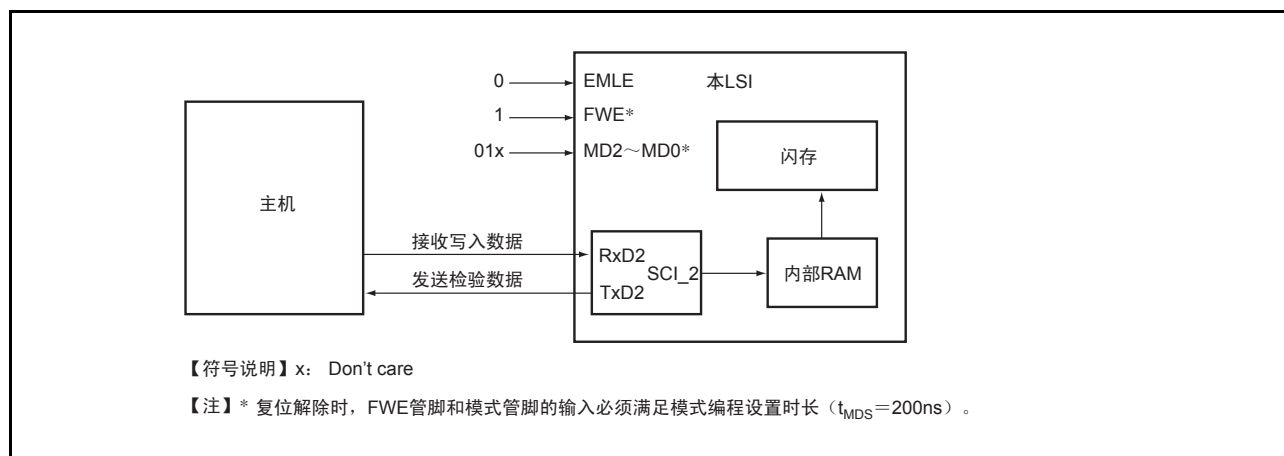


图 17.7 SCI 引导模式时的系统结构图

引导模式下从复位解除到转移到编程控制程序为止的运行如表 17.4 所示。

1. 在引导模式下，主机需要事先准备闪存的编程控制程序。要按照表 17.8 闪存的运行状态准备编程控制程序。
引导模式下，如果有数据写入到闪存（全部数据均不为1）时，擦除闪存的全部块。引导模式用于不能执行用户编程模式时的强制返回。例如：单板状态下的初始编程或在用户编程模式中启动的程序被意外擦除时。
2. SCI_2 设置为异步模式，接收发送信号的格式为（8位数据，1停止位，无奇偶性）。
3. 如果启动引导程序，则测定从主机连续发送的异步串行通信数据 H'00 的低电平期间，计算比特率，并将 SCI_1 的比特率与主机的比特率匹配。复位解除要在 RxD 管脚为高电平的状态下进行。根据需要，必须在电路板上，将 RxD 管脚及 TxD 管脚上拉。从复位解除到能测定低电平期间为止，大约需要 100 个状态。
4. 在比特率的匹配结束后，发送 1 个字节的 H'00 作为调整结束信号，因此，如果主机正常接收调整结束信号，则必须发送 1 个字节的 H'55。如果不能正常接收，则必须通过复位，再次启动引导模式。由于主机的比特率与本 LSI 的系统时钟频率的组合，会产生容许范围内比特率不匹配的情况。因此，主机的传送比特率与本 LSI 的系统时钟频率必须设定在表 17.5 的范围内。
5. 在引导模式下，内部 RAM 的 4K 字节（H'FFE000～H'FFEFBF）被引导模式使用。可存放主机发送的编程控制程序的区域 HD64F2218、HD64F2218U、HD64F2212、HD64F2212U 为 8K 字节（H'FFC000～H'FFDFFF）、HD64F2211、HD64F2211U 为 4K 字节（H'FFD000～H'FFDFFF）。在程序运行转移到编程控制程序之前，不可使用引导程序区。另外，转移到编程控制程序后，仍保持 RAM 内的引导程序，这时，必须注意。
6. 虽然在转移到编程控制程序时，SCI_2 已结束接收发送操作（SCR 的 RE = 0，TE = 0），但因为在 BRR 中匹配的比特率值被保持，因此，编程控制程序能继续用它接收发送与主机间的编程数据和验证数据。TxD 管脚进入高电平输出状态。转移到编程控制程序后，CPU 的通用寄存器不定。特别是堆栈指针被隐含地使用在子程序调用等，因此，必须在编程控制程序的开头初始化。
7. 通过复位解除引导模式。使复位管脚保持低电平，在最少经过 20 个状态后，设置 FWE 管脚和模式管脚，并解除 * 复位。另外，发生 WDT 的溢出复位时也能解除引导模式。
8. 在引导模式过程中，不可改变模式管脚的输入电平。在复位中，如果改变模式管脚的输入电平（低电平→高电平），则因为工作模式切换，地址输出兼用端口及总线控制输出信号（ \overline{AD} 、 \overline{RD} 、 \overline{WR} ）的状态也发生改变。因此，必须注意避免在复位过程中将这些管脚设置成输出信号，及避免与外部信号发生冲突
9. 在对闪存进行编程或擦除的过程中，不可使用中断。

【注】 * 在复位解除时，FWE 管脚和模式管脚的输入，需要满足模式编程设置时间（ $t_{MDS} = 200ns$ ）。

表 17.4 引导模式的运行

项目	主机的运行	本 LSI 的运行
		复位开始后转移到引导程序
比特率调整 ↓	以指定的比特率连续发送 H'00 如果正常接收 H'00，则发送 1 字节 H'55	测定接收数据 H'00 的低电平期间 计算比特率，设置 SCI_2 的 BRR 作为比特率调整结束后的信号，发送 1 字节 H'00 如果接收 H'55，则向主机发送 1 字节的 H'AA
传送编程控制程序的字节数 (N) ↓	以高位字节，低位字节的顺序发送 2 个字节的传送编程控制程序的字节数 (N)	将接收到的 2 个字节数据作为验证数据回送到主机
编程控制程序的传送 (重复 N 次) ↓	发送 1 字节的编程控制程序	将接收到的数据回送到主机的同时传送到 RAM
存储器擦除 ↓		检查闪存的数据，写入时，擦除全部块，并向主机发送 H'AA 如果不可擦除时，则发送 H'FF，停止运行
编程控制程序的执行		转移到传送至内部 RAM 的编程控制程序，并开始执行

表 17.5 可进行比特率自动匹配的系统时钟频率

主机的比特率	本 LSI 的系统时钟频率范围
19200bps	8 ~ 24MHz
9600bps	6 ~ 24MHz
4800bps	6 ~ 24MHz

17.6.2 USB 引导模式（HD64F2218U、HD64F2212U、HD64F2211U）

17.6.2.1 特点

- 可选择总线电源和自身电源模式
- 通过系统时钟 16MHz，PLL3 倍频（FWE = 1，MD2 ~ MD0 = 011）或系统时钟 24MHz PLL2 倍频（FWE = 1，MD2 ~ MD0 = 010）生成支持 USB 运行时钟
- D+ 上拉控制连接仅支持 P36 管脚
- 列举情况请参照表 17.6

表 17.6 列举情况

USB 规格	Ver.1.1	
传送模式	Control (in、out)，Bulk (in、out)	
最大功率模式	自身电源模式（UBPM 管脚 = 1）	100mA
	总线电源模式（UBPM 管脚 = 0）	500mA
终点结构	<pre> graph TD C1[Configuration 1] --> I0[Interface Number 0] I0 --> AS0[Alternate Setting 0] AS0 --> EP0[EP0 Control (in,out) 64Bytes] AS0 --> EP1[EP1 Bulk (in) 64Bytes] AS0 --> EP2[EP2 Bulk (out) 64Bytes] </pre>	

17.6.2.2 USB 引导模式运行注意事项

- 16MHz 或 24MHz 的系统时钟必须与 FWE、MD2 ~ MD0 管脚设置正确匹配。
- D+ 上拉控制连接必须使用 P36 管脚。
- 为了在对闪存的写入/擦除过程中提供稳定电压，不能与经过总线电源 HUB 的数据线相连。
- 如果在对闪存的写入/擦除过程中拔掉 USB 数据线，最坏时，可能会对 LSI 造成永久破坏，因此，必须特别注意。
- 为总线电源模式时，即使 USB 总线进入挂起模式，本 LSI 也不向各种低功耗模式转移。

17.6.2.3 概要

如果将本 LSI 的管脚设置为引导模式后开始复位，则启动事先装入单片机内的引导程序，并使用 USB 将主机准备的编程控制程序依次发送至本 LSI。本 LSI 将通过 USB 接收到的编程控制程序写入内部 RAM 的编程控制程序区。传送结束后，转移到编程控制程序区的起始地址，并进入编程控制程序运行状态（执行闪存的编程）。USB 引导模式时的系统结构图如图 17.8 所示。

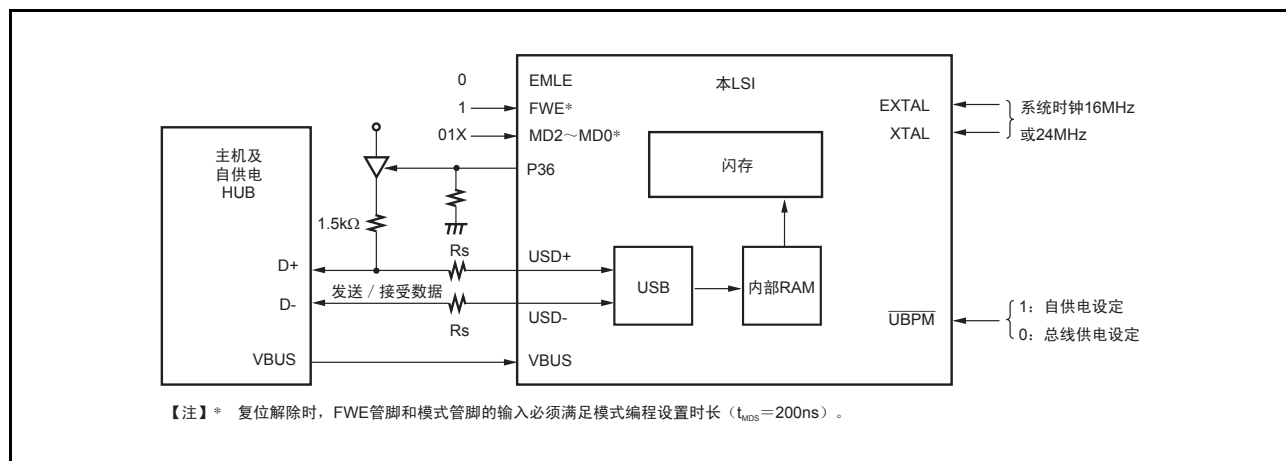


图 17.8 USB 引导模式时的系统结构图

USB 引导模式下从复位解除到转移到编程控制程序为止的运行如表 17.7 所示。

1. 在引导模式下，主机需预先准备闪存的编程控制程序。必须按照 17.8 闪存的编程/擦除准备编程控制程序。
引导模式下，如果有数据写入闪存（全部数据均不为1）时，擦除闪存的全部块。引导模式用于不能执行用户编程模式时的强制返回。例如：单板状态下的初始编程或在用户编程模式中启动的程序被意外擦除时。
2. 启动引导模式时，执行与主机有关的仿真。仿真情况如表 17.6 所示。
如果完成仿真，主机必须发送1字节的H'55。如果不能正常接收，则通过复位重新启动引导模式。
3. 主机发出的频率必须设置为MHz×100单位的数值。
(例：16.00MHz→H'0640，24.00MHz→H'0960)
4. 在引导模式下，内部RAM的4K字节（H'FFE000～H'FFEFBF）被引导程序使用。可存放主机发送的编程控制程序的区域如果是HD64F2218、HD64F2218U、HD64F2212、HD64F2212U为8K字节（H'FFC000～H'FFDFFF），如果是HD64F2211、HD64F2211U为4K字节（H'FFD000～H'FFDFFF）。在程序运行转移到编程控制程序之前不可使用引导程序区。另外，转移到编程控制程序后，仍保持RAM内的引导程序，这时，必须注意。
5. 转移到编程控制程序时，USB保持连接状态，因此，编程控制程序能继续使用它接收发送与主机间的编程数据和验证数据。转移到编程控制程序后，CPU的通用寄存器不定。特别是堆栈指针被隐含地使用在子程序调用等，因此，必须在编程控制程序的开头初始化。
6. 通过复位解除引导模式。使复位管脚保持低电平，在最少经过20个状态后，设置FWE管脚和模式管脚，并解除*复位。另外，发生WDT的溢出复位时也能解除引导模式。
7. 在引导模式过程中，不可改变模式管脚的输入电平。在复位中，如果改变模式管脚的输入电平（低电平→高电平），则因为工作模式切换，地址输出兼用端口及总线控制输出信号（ \overline{AS} 、 \overline{RD} 、 \overline{WR} ）的状态也会发生改变。因此，必须注意避免在复位过程中将这些管脚设置成输出信号，及避免与外部信号发生冲突。
8. 在对闪存进行编程或擦除的过程中，不可使用中断。

【注】 * 解除复位时，FWE管脚及模式管脚的输入必须满足模式编程设置时间（ $t_{MDS} = 200ns$ ）。

表 17.7 USB 引导模式的运行

项目	主机的运行	本 LSI 的运行
		复位开始后转移到引导程序
USB 引导模式开始 ↓	USB 仿真结束后, 发送 1 个字节的 H'55	如果接收 H'55, 则向主机发送 1 个字节的 H'AA
传送时钟情况 ↓	发送频率 (2 字节), 倍频种类数 (1 字节), 倍频率 (1 字节) 16MHz 时, 发送 H'0640, H'01, H'01 24MHz 时, 发送 H'0960, H'01, H'01	接收到的数据在各范围内时, 向主机发送 H'AA 任意一个接收到的数据在范围外时, 向主机发送 H'FF, 并停止运行
传送编程控制程序的字节数 (N) ↓	发送 2 个字节的传送编程控制程序的字节数 (N)	接收到的字节数在范围内时, 向主机发送 H'AA 接收到的字节数在范围外时, 向主机发送 H'FF, 并停止运行
传送编程控制程序及总值	将编程控制程序分成 N 个字节发送 发送总值 (编程控制程序的总和的补数 (1 字节))	将接收到的数据传送至内部 RAM 算出接收到的总值及传送至内部 RAM 的编程控制程序的 1 个字节单位的总和 和为 0 时, 向主机发送 H'AA 和不为 0 时, 向主机发送 H'FF, 并停止运行
存储器擦除 ↓	发送全部擦除状态指令 (H'3A) 接收 H'11 时, 再次发送全部擦除状态指令 (H'3A)	闪存全面擦除开始 如果接收全面擦除状态指令时, 正在进行全面擦除处理, 则向主机发送 H'11 如果接收全面擦除状态指令时, 已经结束全部块的擦除, 则向主机发送 H'06 如果接收全面擦除状态指令时, 不可进行擦除, 则向主机发送 H'EE, 并停止运行
编程控制程序的执行		转移到传送至内部 RAM 的编程控制程序, 并开始执行

17.6.3 用户编程模式

在用户模式下，也可以通过转移到用户准备的编程 / 擦除程序，在单板上擦除改写任意块。用户需要准备电路板上 FWE 的控制，提供单板上的改写数据，还有程序转移的条件设置。另外，根据需要，有必要给闪存的一部分写入编程 / 擦除程序，或者写入为了从外部提供的编程 / 擦除程序的程序。由于在编程 / 擦除过程中，不能读出闪存，因此必须和引导模式一样，把编程 / 擦除程序传送到内部 RAM 中，再运行。用户模式下的编程 / 擦除步骤如图 17.9 所示。请根据 17.8 闪存的编程 / 擦除准备编程 / 擦除程序。

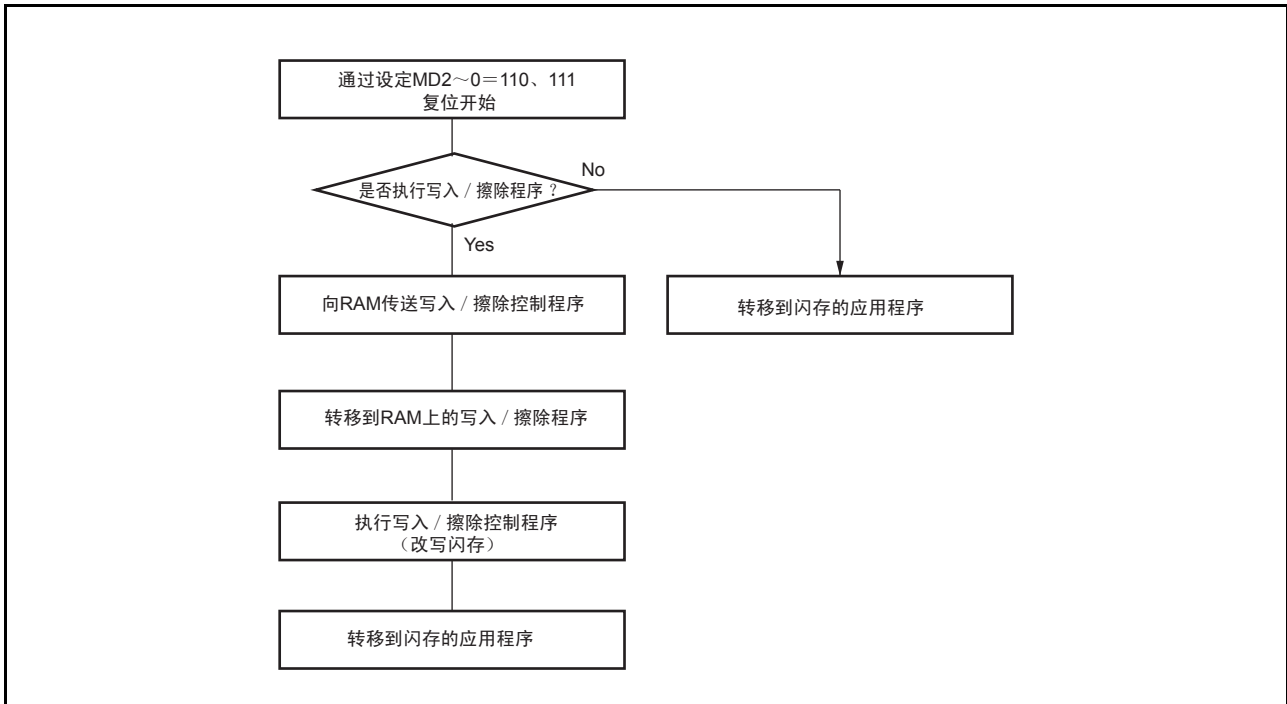


图 17.9 用户模式下的编程 / 擦除

17.7 RAM 中的闪存仿真

通过 RAM 仿真寄存器（RAMER）将闪存的一部分块与 RAM 重叠使用，可在内部 RAM 中对写入闪存的改写数据进行实时仿真，可仿真模式有用户模式和用户编程模式。对闪存的改写进行实时仿真的例如图 17.10 所示。

1. 设置RAMER，将需要实时改写的区域与RAM重叠。
2. 使用被重叠的RAM进行仿真。
3. 确定改写数据后，清除RAMS位，并解除RAM的重叠。
4. 将写入到重叠RAM的数据写入闪存空间。

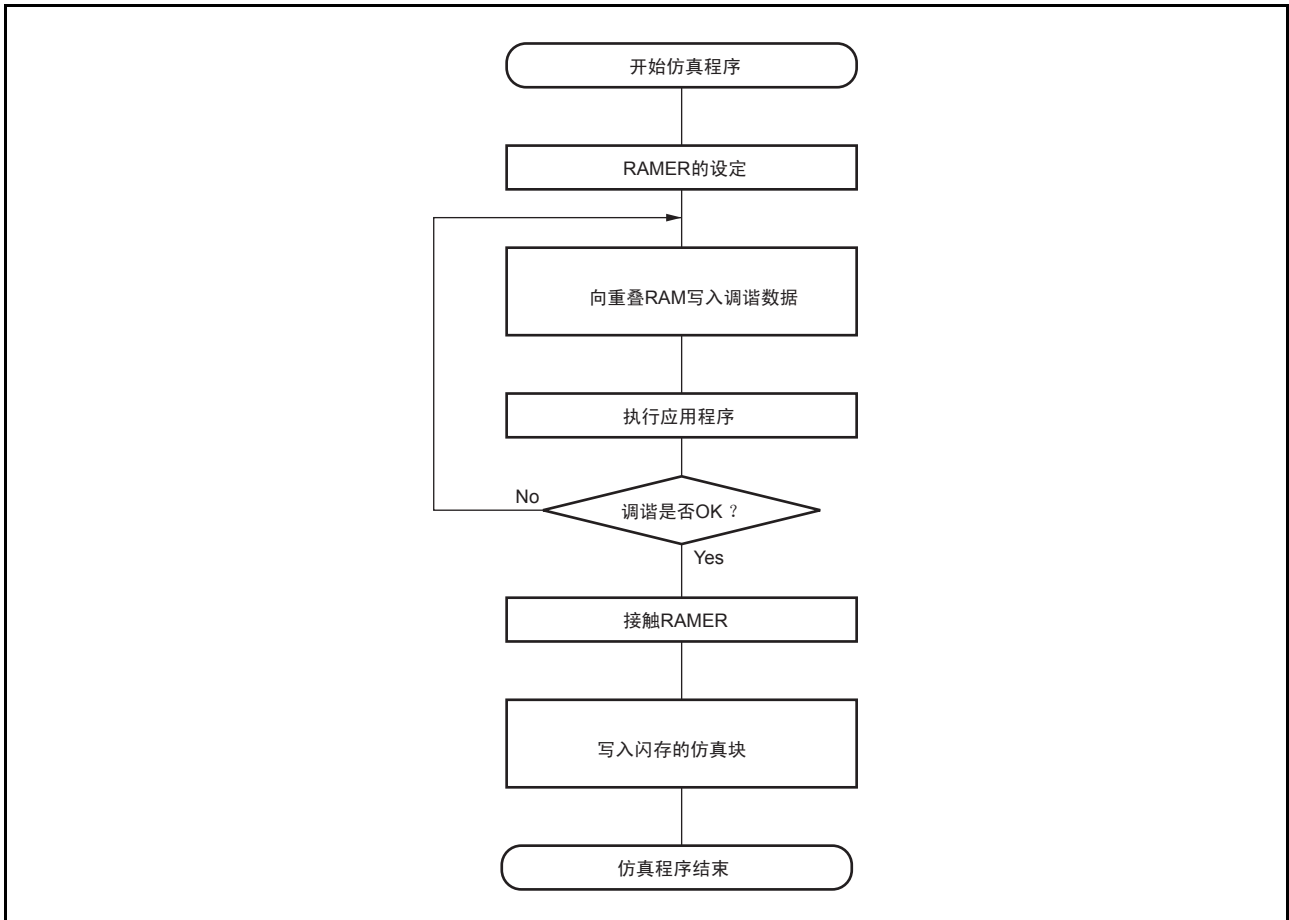


图 17.10 RAM 中的仿真流程图

将闪存的块 EB1 重叠的例子如图 17.11 所示。

1. 将被重叠的 RAM 区域固定为 H'FFD000 ~ H'FFD3FF 的 1K 字节。
2. 可重叠的闪存区域，可通过 RAMER 选择 1K 字节的 EB0 ~ EB3 之中的 1 块。
3. 被重叠的 RAM 区可以从闪存内的地址及原来的 RAM 地址双方进行存取。
4. RAMER 的 RAMS 位置 1 时，闪存的全部块进入编程/擦除保护（仿真保护），即使 FLMCR1 的 P1 位或 E1 位置位，也不向编程模式及擦除模式转移。
5. 即使按照擦除算法运行程序，也不擦除 RAM 区。
6. 块 EB0 包括向量表。进行 RAM 仿真时，重叠 RAM 中也需要向量表。

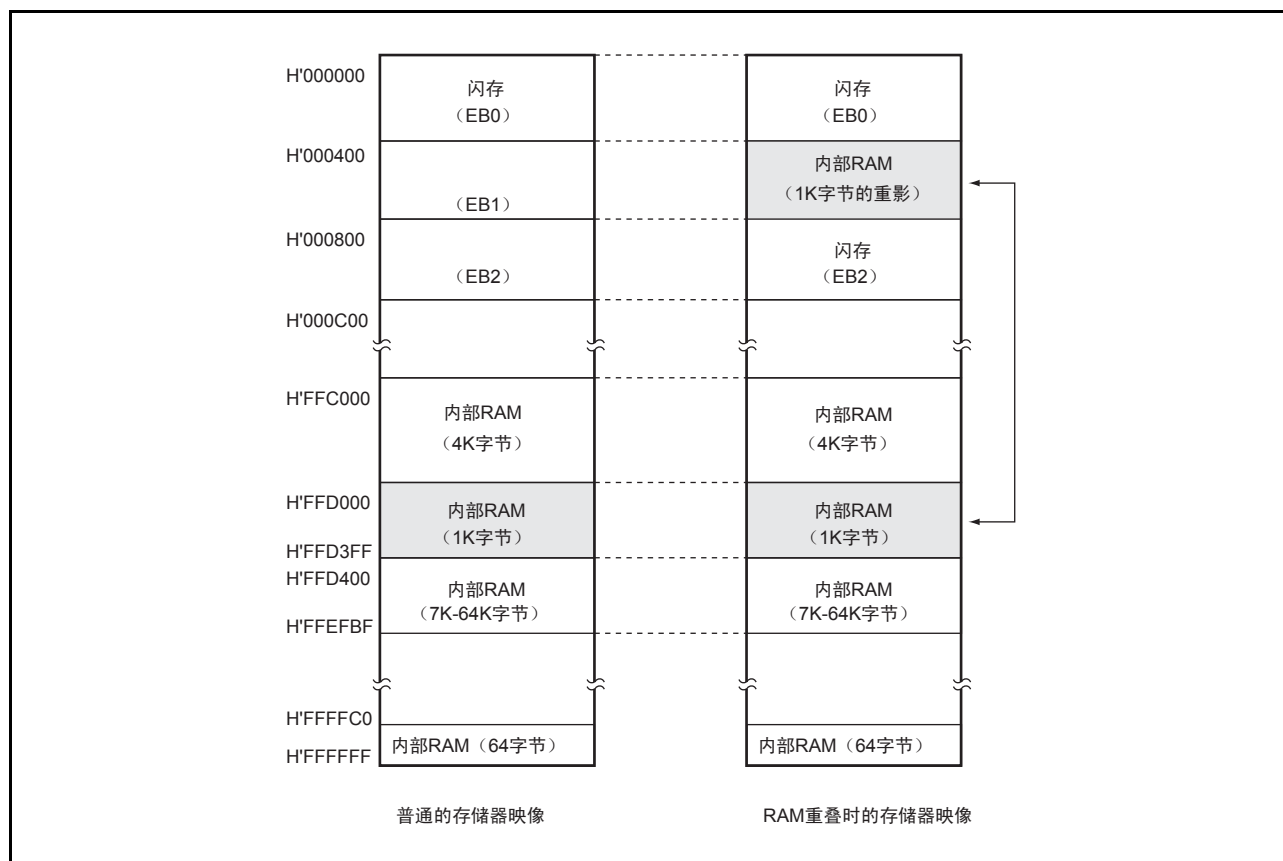


图 17.11 RAM 的重叠例

17.8 闪存的编程 / 擦除

采用软件方式，用 CPU 对单板上的闪存进行编程 / 擦除。闪存根据 FLMCR1 的设置，转移到编程模式，编程验证模式，擦除模式和擦除验证模式。组合引导模式下的编程控制程序 and 用户模式下的编程 / 擦除程序，进行编程 / 擦除。闪存的编程，请按照 17.8.1 编程 / 编程验证进行，闪存的擦除请按照 17.8.2 擦除 / 擦除验证进行。

17.8.1 编程 / 编程验证

请按照图 17.12 所示的编程 / 编程验证流程图对闪存进行编程。如果按照这个流程图进行编程操作，可以把数据或程序写入闪存而不使芯片遭受电压应力或牺牲数据的可靠性。

1. 在擦除状态下进行编程，对已经编程的地址不可进行再次编程。
2. 一次编程的单位为 128 字节。即使写入的数据不满 128 字节，也必须向闪存传送 128 字节的数据。没必要写入的地址的数据请写入 H'FF。
3. 必须在 RAM 上保证 128 字节的编程数据区，128 字节的再次编程数据区和 128 字节的追加编程数据区。再次编程数据的运算，追加编程数据的运算如图 17.12 进行。
4. 必须以字节为单位，从再次编程数据区或者追加编程数据区给闪存连续传送 128 字节。程序地址及 128 字节的数据被锁存在闪存内。必须将传送地的闪存的起始地址低 8 位设定为 H'00 或 H'80。
5. P1 位置 1 的时间为编程时间。请按照图 17.12 设置编程时间。
6. 为了避免因程序失控等造成的重复编程，而设置看门狗定时器。溢出周期必须比 $(y+z1 + \alpha + \beta)$ μs 大。
7. 对验证地址进行虚写，必须给低 1 位为 B'0 的地址写入 1 字节的 H'FF。从已进行虚写的地址开始，以字为单位读取验证数据。
8. 对同一位重复进行编程 / 编程验证，不要超过 $(N1 + N2)$ 次。

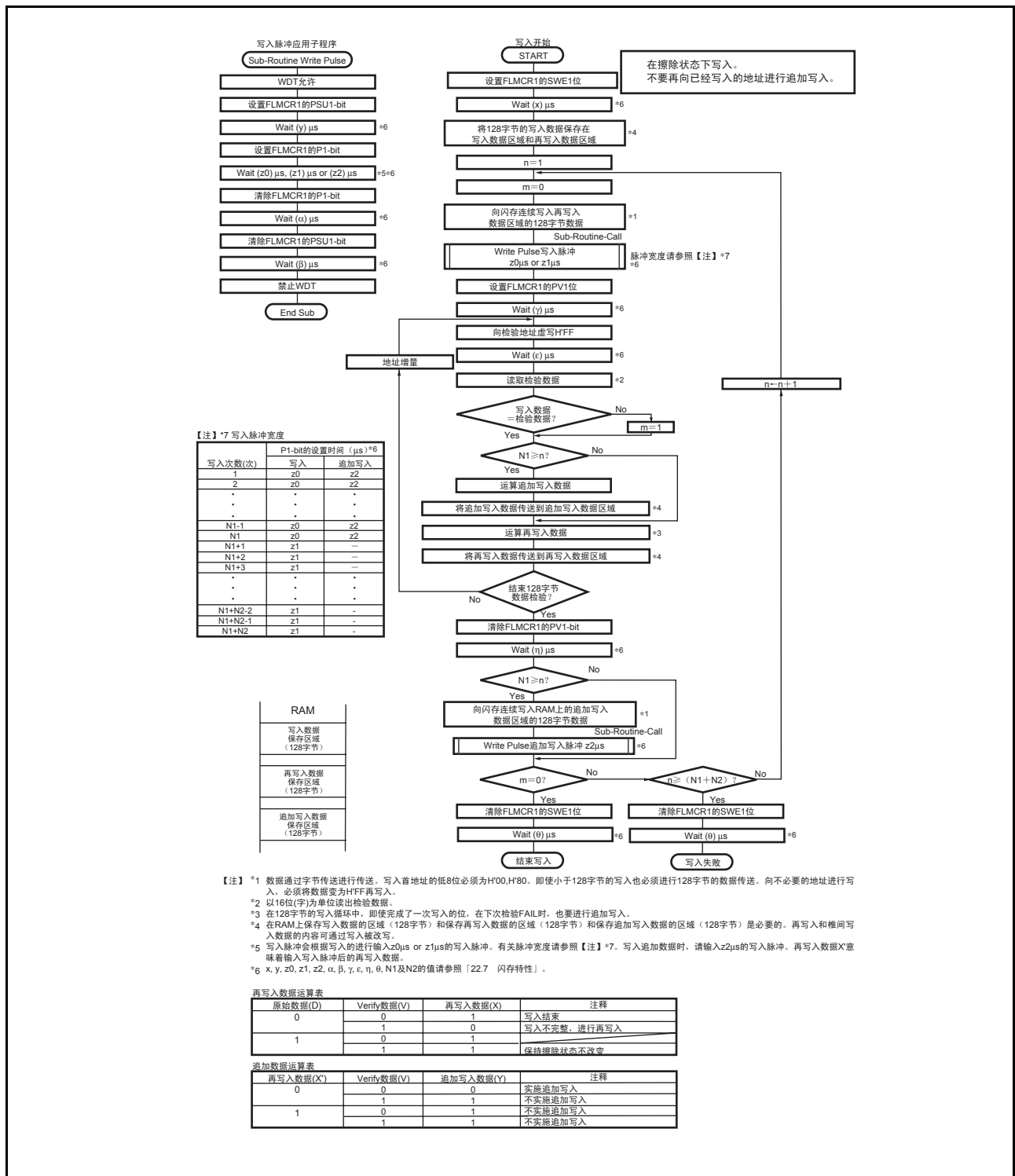


图 17.12 编程 / 编程验证流程图

17.8.2 擦除 / 擦除验证

按照图 17.13 所示的擦除 / 擦除验证流程图进行擦除。

1. 在擦除前不必进行预写入（把要擦除的存储器的所有数据全部置为0）。
2. 擦除以块为单位进行。通过擦除块指定寄存器1, 2（EBR1、EBR2）选择1块要擦除的块。擦除多块时，也必须按块逐个擦除。
3. E1位置1的时间为擦除时间。
4. 为了避免因程序失控造成的重复编程，而设置看门狗定时器。溢出周期必须比 $(y+z+ \alpha + \beta)$ ms大。
5. 对验证地址进行虚写，必须给低1位为B'0的地址写入1字节的H'FF。从已进行虚写的地址开始，以长字为单位读取验证数据。
6. 读出的数据在未擦除时，再次设置为擦除模式，同样重复擦除/擦除验证顺序。但是，重复次数不要超过（N）次。

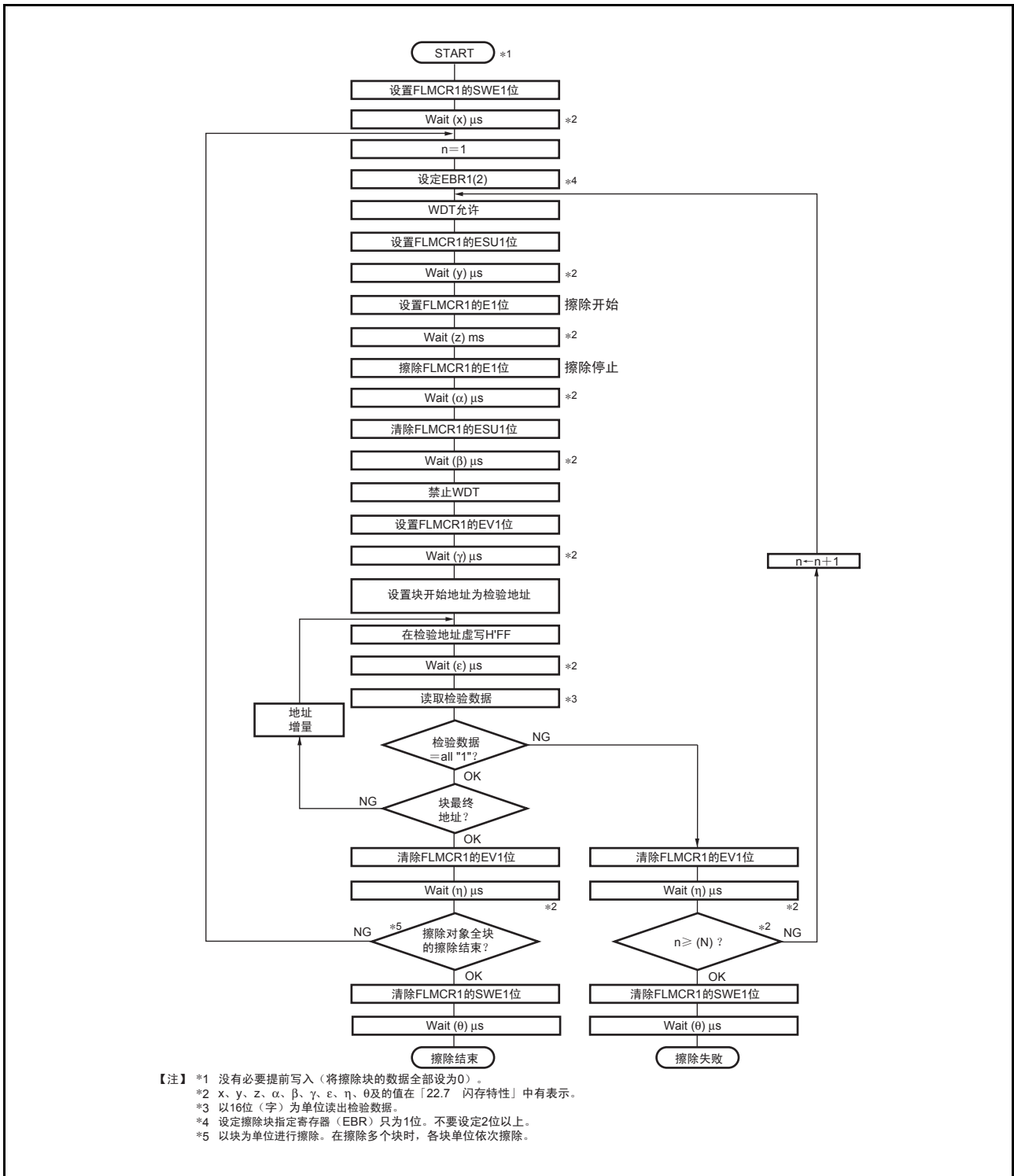


图 17.13 擦除 / 擦除验证流程图

17.9 编程 / 擦除保护

闪存的编程 / 擦除保护状态有硬件保护, 软件保护及错误保护 3 种。

17.9.1 硬件保护

硬件保护是通过向复位 (也包括通过 WDT 的加电复位) 或者待机模式的状态转移及 FWE = 低电平, 来强制禁止和中断闪存的编程 / 擦除的状态。闪存控制寄存器 1 (FLMCR1)、闪存控制寄存器 2 (FLMCR2) 块指定寄存器 1 (EBR1) 和块指定寄存器 2 (EBR2) 被初始化。通过 RES 管脚复位时, 如果从加入电源到震荡稳定为止, 不保持 RES 管脚为低电平, 就不能进入复位状态。另外, 运行中的复位, 在 AC 特性规定的 RES 脉冲宽度之间, 必须保持 RES 脚为低电平。

17.9.2 软件保护

通过软件清除 FLMCR1 的 SWE1 位, 使全部块的编程 / 擦除进入保护状态。在此状态下, 即使将 FLMCR1 的 P1 位或 E1 位置位, 也不向编程模式或擦除模式转移。另外, 通过设置块指定寄存器 1 (EBR1) 块指定寄存器 2 (EBR2) 可以按块进行擦除保护。如果将 EBR1, EBR2 都设置为 H'00, 则全部块均进入擦除保护状态。通过将 RAMER 的 RAMS 位置位, 也可进入全部块的编程 / 擦除保护状态。

17.9.3 错误保护

错误保护是在对闪存进行编程 / 擦除过程中, 检测出 CPU 失控或不按照编程 / 擦除算法的运行, 强制中断编程 / 擦除运行的状态。通过中断编程 / 擦除运行, 来防止因改写 / 重复擦除给闪存造成的损坏。

如果在对闪存进行编程 / 擦除的过程中, 检测出以下错误, 则 FLMCR2 的 FLER 位被置 1, 进入错误保护状态。

FLER 位置位条件 (错误保护)

- 读出处于编程/擦除过程中的闪存 (包括读向量及取向量)
- 在编程/擦除过程中, 开始除复位以外的异常处理
- 在编程/擦除过程中, 执行 SLEEP 指令
- 在编程/擦除过程中, CPU 将总线权释放给 DMAC

此时, 虽然保持 FLMCR1, FLMCR2, EBR1 和 EBR2 的内容, 但在检测出错误时, 编程模式或擦除模式被强制中断。即使将 P1 位和 E1 位置位, 也不向编程模式或擦除模式转移。但是, PV1 位和 EV1 位被保持, 并且能转移到验证模式。只能通过加电复位或硬件待机, 才能解除错误保护状态。

17.10 编程 / 擦除闪存时的中断

在对闪存进行编程 / 擦除的过程中 (FLMCR1 的 P1 位或 E1 位置位), 或者在引导程序执行过程中 *1, 为了使编程 / 擦除操作最优先, 必须禁止包括 NMI 输入在内的所有中断。这样可以避免以下的运行状态。

1. 如果在编程/擦除过程中发生中断, 则不能保证按照编程/擦除算法进行正常运行。
2. 编程/擦除过程中的中断异常处理, 不能正常地读向量 *2, 且 CPU 失控。
3. 如果在引导程序运行过程中发生中断, 则不能正常地执行引导模式的顺序。

【注】 *1 到编程控制程序完成编程为止, 需要禁止 CPU 内部及外部的中断请求。

*2 由于以下 2 点, 不能正常读取向量。

- 在编程/擦除过程中 (FLMCR1 的 P1 位或 E1 位置位), 即使执行闪存的读取, 也不能读出正确值 (值不定)。
- 如果中断向量表中尚未写入值, 则不能正常执行中断异常处理。

17.11 编程器模式

在编程器模式与单个闪存相同，可通过插座适配器，用 PROM 编程器进行编程 / 擦除。请使用支持瑞萨 128K 字节或 64K 字节闪存内部单片机器件类型的 PROM 编程器。

编程器模式时的存储器映射如图 17.14 所示。

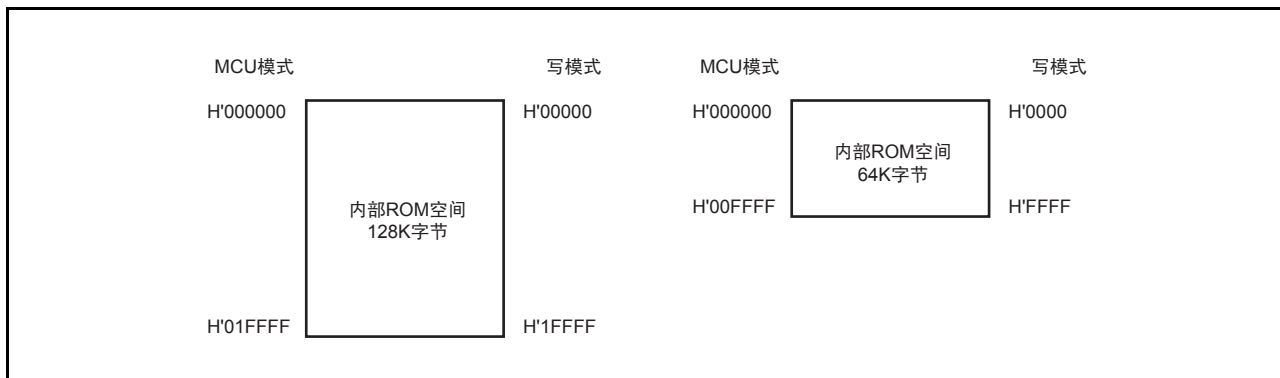


图 17.14 编程器模式时的存储器映射

17.12 闪存的低功耗状态注意事项

在用户模式，闪存进入以下的任意状态：

- 通常工作状态
可高速读出闪存。
- 待机状态
停止闪存的所有电路。
- 低功耗状态
停止电源电路的一部分，LSI 为子时钟操作时，可读出闪存。

本 LSI 的工作模式与闪存状态的关系如表 17.8 所示。闪存从低功耗工作状态或待机状态恢复到通常工作状态时，需要已停止的电源电路的稳定时间。包括使用外部时钟的情况，为了使恢复到通常工作模式时的待机时间保持在 100 μ s 以上，在从监视模式或待机模式恢复时，必须设定 SBYCR 的 STS2 ~ STS0，从闪存模块停止模式恢复时，必须用软件等待。

表 17.8 闪存的运行状态

本 LSI 的工作模式	闪存的状态
激活模式 睡眠模式	通常工作状态
监视模式 待机模式 闪存模块停止模式	待机状态 (恢复到通常工作状态时，需要 100 μ s 的待机时间)
子激活模式 子睡眠模式	低功耗状态（只读）

17.13 编程 / 擦除闪存时的注意事项

使用单板上编程模式 RAM 仿真功能及编程器模式时的注意事项。

1. 必须以规定的电压时序进行编程/擦除。
外加大于额定值的电压时，可能会对产品造成永久性损坏。
请使用支持瑞萨闪存内部单片机器件类型（FZTAT128V3A，FZTAT64V3A）的 PROM 编程器。不可将编程器设置为 HN27C4096。另外，不可使用规定以外的插座适配器。误使用时，可能会造成损坏。
2. 加电/断电时的注意事项
必须在 Vcc 确定以后，再给 FWE 管脚外加高电平。另外，必须在切断 Vcc 之前将 FWE 管脚置为低电平。外加/切断 Vcc 电源时，必须将 FWE 管脚固定为低电平，并将闪存置入硬件保护状态。加电/断电时序，必须满足由于停电引起的电源切断和再次加电。
3. 外加/解除 FWE 的注意事项
FWE 的外加必须在确定单片机运行的状态下进行。如果不满足单片机运行稳定状态，则必须将 FWE 管脚固定为低电平，并将其置入保护状态。外加/解除 FWE 时，为了防止对闪存的误编程，误擦除，必须注意以下几点。
 - 请在 VCC 电压在额定电压范围内处于稳定的状态下，外加 FWE。
 - 引导模式下，请在复位过程中，外加/解除 FWE。
 - 用户编程模式下，不论复位的状态如何，都可进行 FWE = 高电平/低电平的切换。另外，即使在闪存的程序执行过程中，也可进行 FWE 输入的切换。
 - 请在程序没有失控的状态下，外加 FWE。
 - 请在清除 FLMCR1 的 SWE1，ESU1，PSU1，EV1，PV1，P1，E1 位的状态下，解除 FWE。外加/解除 FWE 时，不可错误地将 SWE1，ESU1，PSU1，EV1，PV1，P1，E1 位置位。
4. 不可给 FWE 管脚外加恒定的高电平。
只有在对闪存进行编程/擦除时，才给 FWE 管脚外加高电平。因此，请避免给 FWE 管脚外加恒定高电平的系统结构。另外，即使在外加高电平时，为了避免重复编程，重复擦除，也必须启动看门狗定时器，以防止程序失控等的发生。
5. 请按照推荐的算法，对闪存进行编程/擦除。
推荐算法，可以在不使芯片遭受电压应力或牺牲程序数据的可靠性的情况下，进行编程/擦除。另外，将 FLMCR1 的 P1 位，E1 位置位时，为防止程序失控必须预先设置看门狗定时器。
6. 在闪存的程序执行过程中，不可置位/清除 SWE1 位。
请在清除 SWE1 位，并等待 (0) * μ s 以上的时间后，进行闪存的程序执行及数据读出。
如果将 SWE1 位置位，虽然能改写闪存的数据，但只有为验证（编程/擦除中的验证）时，才可对闪存进行存取。另外，在编程/擦除/验证过程中，不可清除 SWE1 位。如果在给 FWE 管脚输入高电平的状态下，即使使用 RAM 中的仿真功能，也必须清除 SWE1 位以后，再进行闪存上的程序执行和数据读出。
但是，对于闪存空间及重叠的 RAM 区，不论置位/清除 SWE1 位，均可读取/写入。
【注】* 请参照 22.7 闪存特性。
7. 在闪存的编程/擦除过程中，不可使用中断。
在外加 FWE 状态下，为了使编程/擦除操作更加优先，因此，必须禁止包括 NMI 在内的所有中断。

8. 不可进行追加编程。改写必须在擦除后进行。
单板上编程时，对 128 字节的编程单位只能进行 1 次编程。编程器模式时，也只能对 128 字节的编程单位进行 1 次编程。
在这些编程单位块被全部擦除的状态下进行编程。
9. 在编程前，必须确认 PROM 编程器安装是否正确。
如果 PROM 编程器的插座，插座适配器以及器件的索引不一致时，会产生过电流损坏产品的情况。
10. 在编程过程中，不可用手触碰插座适配器和器件。
可能会因为接触不良，而产生不能很好地编程的情况。
11. 加电后，必须进入复位状态。
必须在震荡稳定期间 100 μ s 以上进入复位。
12. 在运行过程中进入复位时，必须在 SWE1 为低电平时进入复位。
必须在清除 SWE1 位，并等待 (0) * μ s 以上时间后，才进入复位。
【注】* 请参照 22.7 闪存特性。

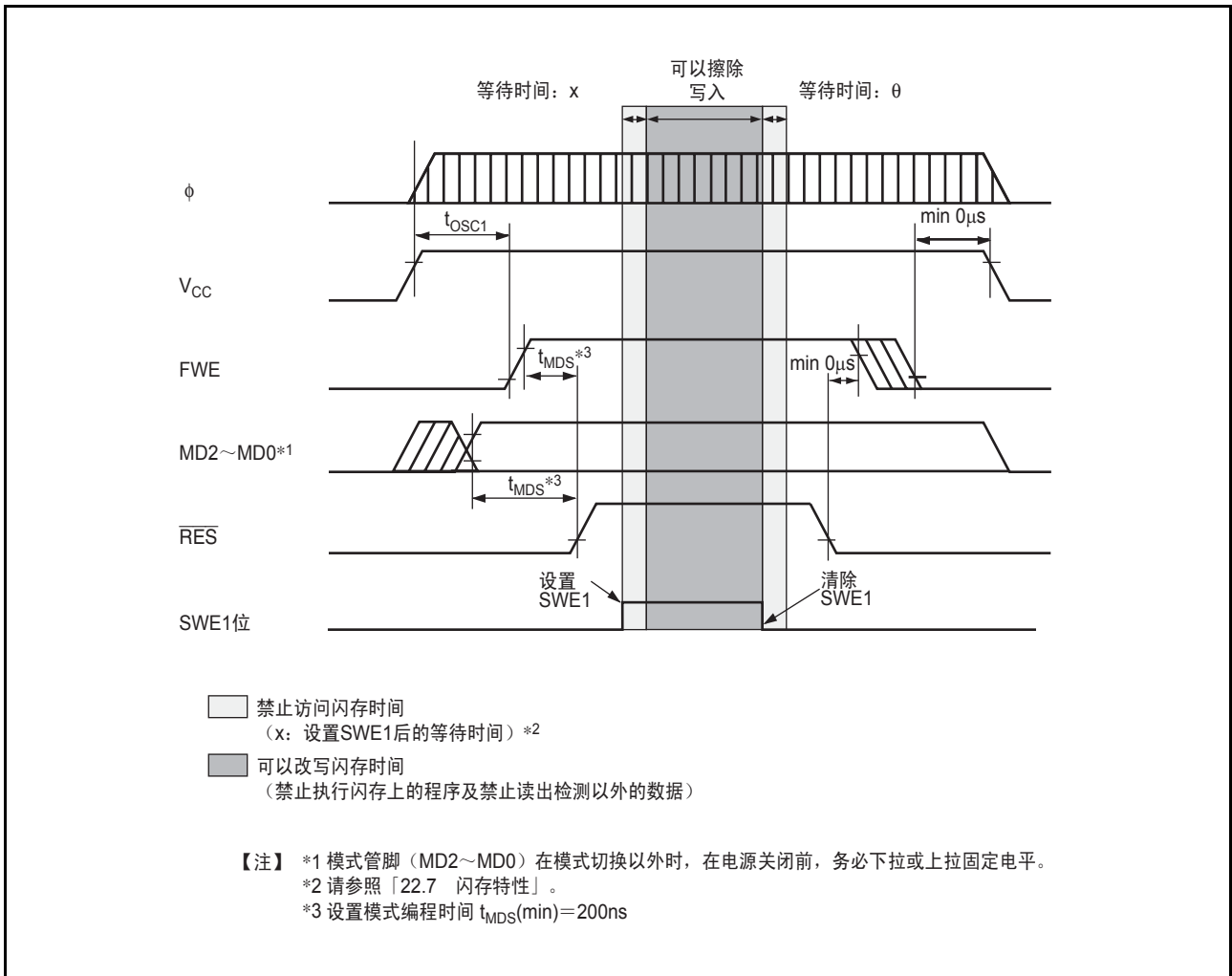


图 17.15 加电 / 断电时序 (引导模式)

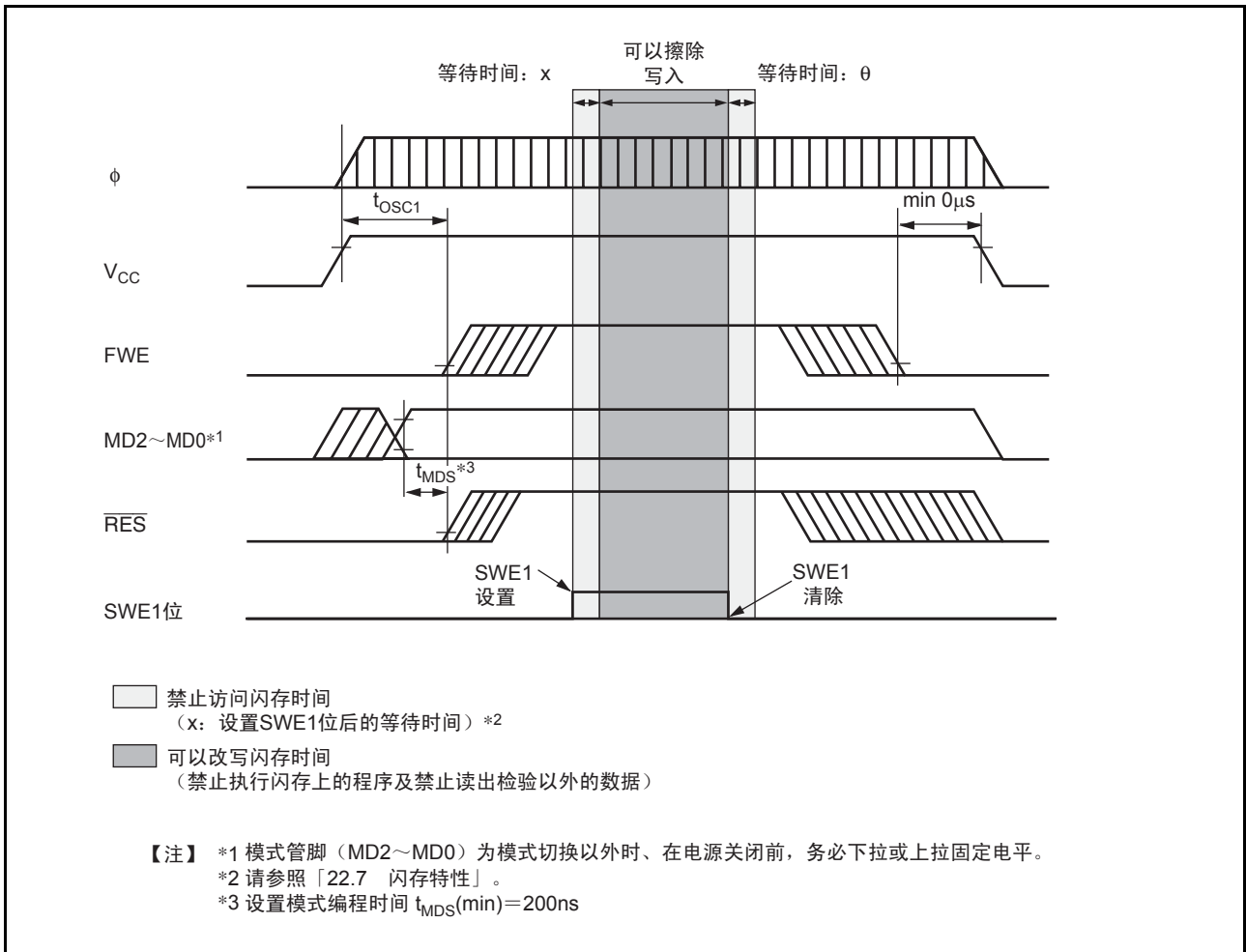


图 17.16 加电 / 断电时序 (用户编程模式)

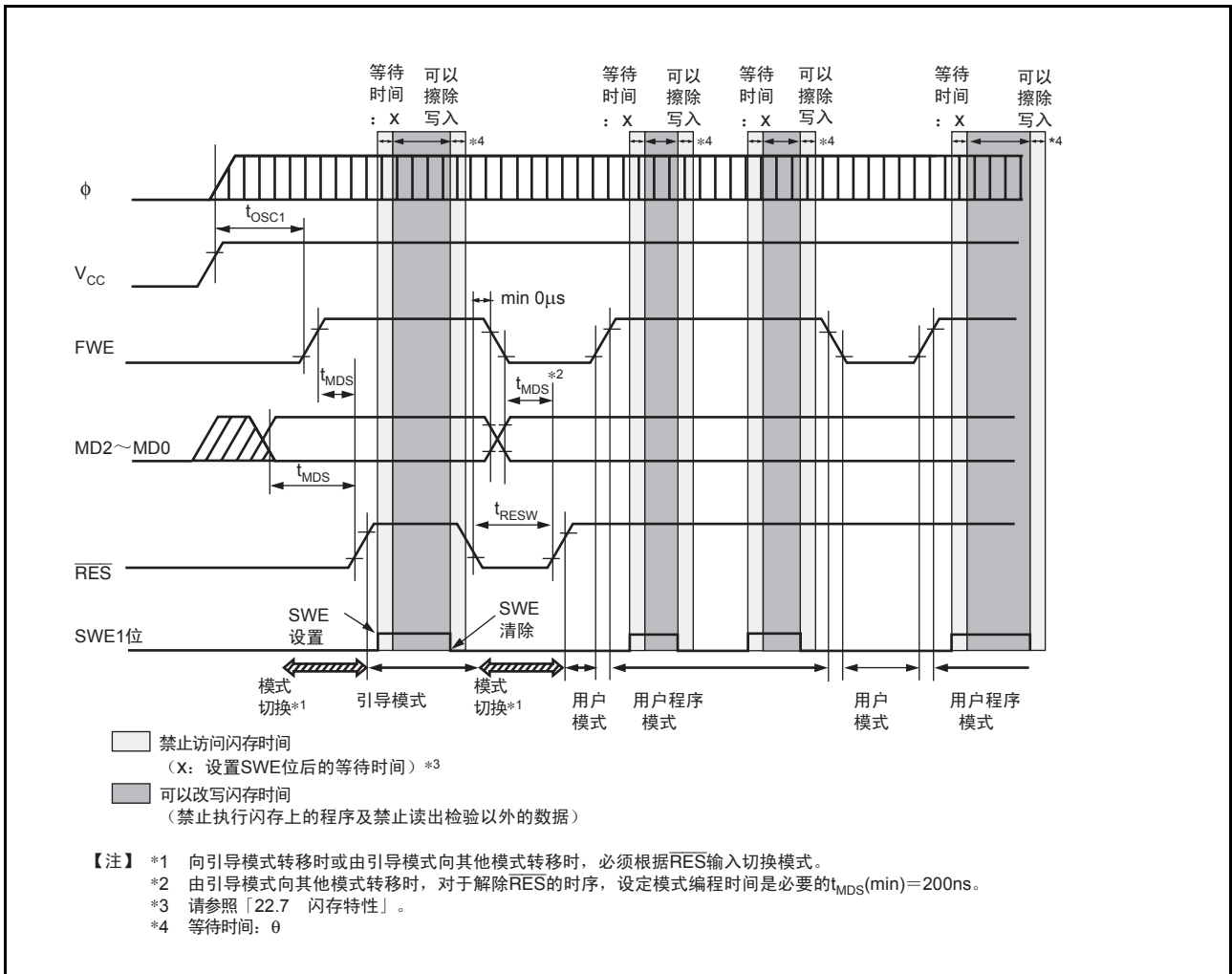


图 17.17 模式转移时序 (例: 引导模式→用户模式←→用户编程模式)

17.14 F-ZTAT 单片机的掩模型 ROM 化时的注意事项

掩模型 ROM 版中没有存在于 F-ZTAT 版中的用于控制闪存的内部寄存器，表 17.9 所示的寄存器存在于 F-ZTAT 版中，但掩模版中没有。如果在掩模版中读取表 17.9 所示的寄存器时，就会读出不定值。因此，将在 F-ZTAT 版中开发的应用软件改变为掩模型 ROM 版时，必须修改应用软件使其不受表 17.9 所示寄存器的影响。

表 17.9 ROM 版中没有存在于 F-ZTAT 版的寄存器

寄存器名称	略称	地址
闪存控制寄存器 1	FLMCR1	H'FFA8
闪存控制寄存器 2	FLMCR2	H'FFA9
擦除块指定寄存器 1	EBR1	H'FFAA
擦除块指定寄存器 2	EBR2	H'FFAB
RAM 仿真寄存器	RAMER	H'FEDB
串行控制寄存器 X	SCRX	H'FDB4

18. 掩模型 ROM

掩模型 ROM 有如下特点：

18.1 特点

- 容量：

产品分类		ROM 容量	ROM 地址（模式 6, 7）
H8S/2218 群	HD6432217	64K 字节	H'000000 ~ H'00FFFF
H8S/2212 群	HD6432211	64K 字节	H'000000 ~ H'00FFFF
	HD6432210	32K 字节	H'000000 ~ H'007FFF

- 以 16 位宽度的数据总线与总线主控器连接
可 1 态存取字节数据/字数据

掩模型 ROM 的框图如图 18.1 所示。

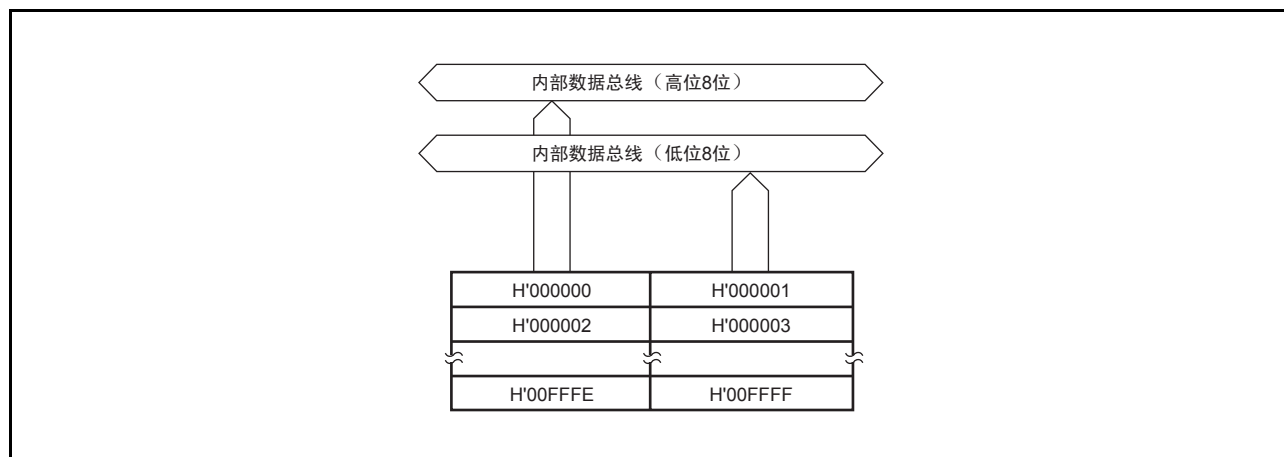


图 18.1 掩模型 ROM 的框图（64K 字节的例）

19. 时钟振荡器

本 LSI 内置一个时钟振荡器，并生成系统时钟 (ϕ)、总线主控器时钟及内部时钟。时钟振荡器由主时钟振荡器、占空比调整电路、时钟振荡器、中速时钟分频器、总线主控器时钟选择电路、子时钟振荡器、波形形成电路、PLL (Phase Locked Loop) 电路、USB 运行时钟选择电路构成。时钟振荡器的框图如图 19.1 所示。

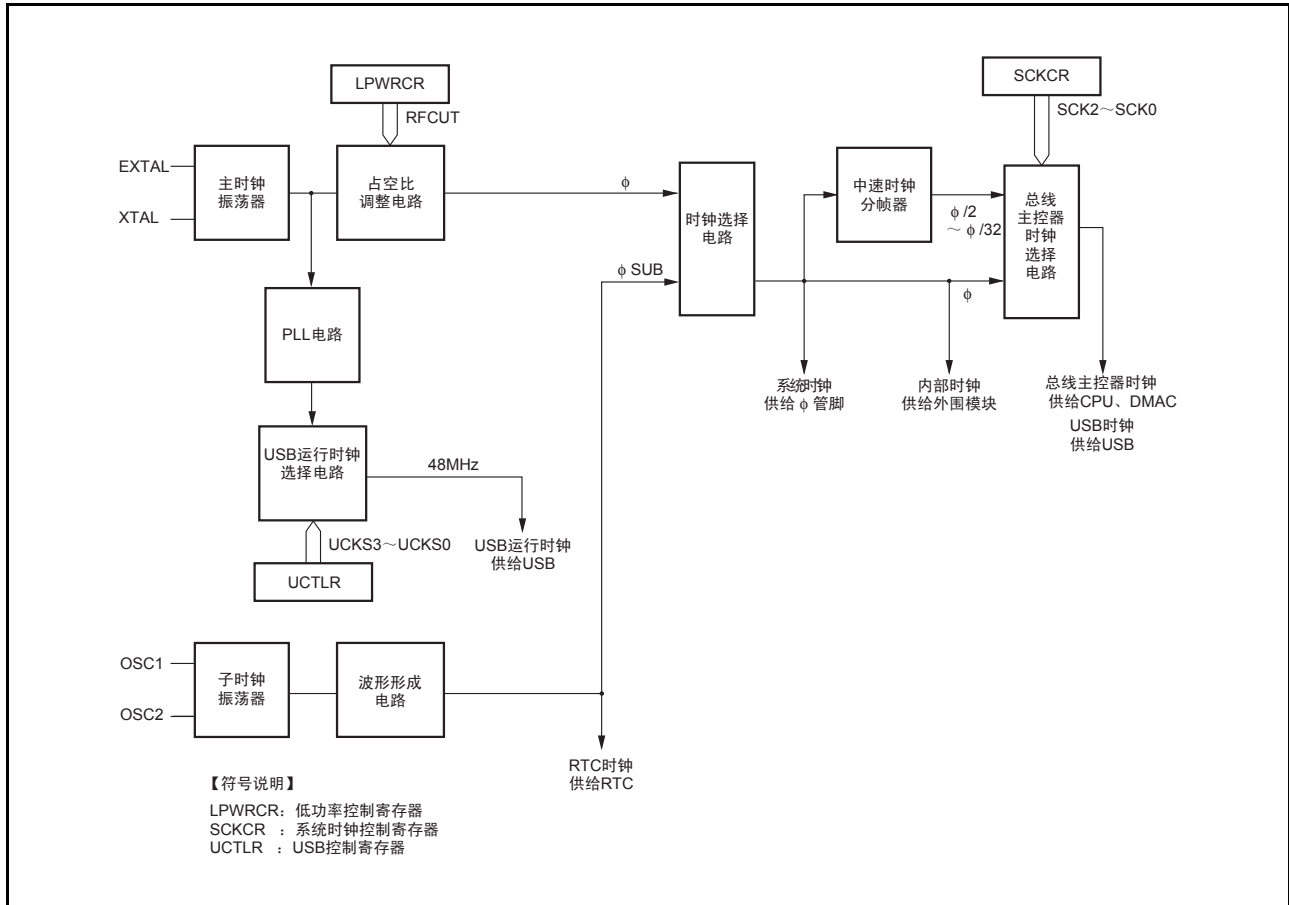


图 19.1 时钟振荡器框图

主时钟振荡器发出的频率通过软件设置低功率控制寄存器 (LPWRCR) 和系统时钟控制寄存器 (SCKCR) 来改变。

通过软件设置 USB 控制寄存器 (UCTLR)，来选择 PLL 电路的 48MHz 时钟。详情请参照 14. 通用串行总线 (USB)。

19.1 寄存器的说明

时钟振荡器有以下寄存器：

- 系统时钟控制寄存器（SCKCR）
- 低功率控制寄存器（LPWRCR）

19.1.1 系统时钟控制寄存器（SCKCR）

SCKCR 执行 ϕ 输出，中速模式控制。

位	位名	初始值	R/W	说明
7	PSTOP	0	R/W	ϕ 时钟输出禁止 控制 ϕ 输出。工作模式不同，运行也有所不同。详细情况请参照 20.11 ϕ 时钟输出控制。 0: ϕ 输出，固定为高电平或高阻抗 1: 固定为高电平或高阻抗
6	—	0	R/W	保留位 虽然可读取 / 写入，但写入时，总是作为 0。
5、4	—	全为 0	—	保留位 读取时，总是读出 0。
3	—	0	R/W	保留位 虽可读取 / 写入，但写入时，总是作为 0。
2 1 0	SCK2 SCK1 SCK0	0 0 0	R/W R/W R/W	系统时钟选择 2 ~ 0 选择总线主控器时钟。另外，转移到子激活模式，监视模式运行时，须将 SCK2 ~ SCK0 全设置为 0。 000: 高速模式 001: 中速时钟 $\phi/2$ 010: 中速时钟 $\phi/4$ 011: 中速时钟 $\phi/8$ 100: 中速时钟 $\phi/16$ 101: 中速时钟 $\phi/32$ 11x: 禁止设置

【符号说明】

x: Don't care

19.1.2 低功耗控制寄存器 (LPWRCCR)

LPWRCCR 执行低功耗模式的控制, 噪音消除采样频率的选择、子时钟振荡器的控制及是否使用系统时钟振荡器的内置反馈电阻和占空比调整电路的选择。

位	位名	初始值	R/W	说明
7	DTON	0	R/W	直接传送标志 0: 在高速模式或中速模式下, 执行 SLEEP 指令时, 转移到睡眠模式软件待机模式或监视模式*; 在子激活模式下, 执行 SLEEP 指令时, 转移到子睡眠模式或监视模式。 1: 在高级模式或中速模式下, 执行 SLEEP 指令时, 直接转移到子激活模式* 或转移到睡眠模式, 软件待机模式在子激活模式下, 执行 SLEEP 指令时, 直接转移到高速模式, 或转移到子睡眠模式。
6	LSON	0	R/W	低速标志 0: 在高速模式或中速模式下, 执行 SLEEP 指令时, 转移到睡眠模式, 软件待机模式或监视模式* 在子激活模式下, 执行 SLEEP 指令时, 转移到监视模式* 或直接转移到高速模式。解除监视模式后, 转移到高速模式 1: 在高速模式下, 执行 SLEEP 指令时, 转移到监视模式* 或转移到子激活模式。* 在子激活模式下, 执行 SLEEP 指令时, 转移到子睡眠模式或监视模式。解除监视模式后转移到子激活模式
5	NESEL	0	R/W	选择噪音消除采样频率 此位选择子时钟振荡器生成的子时钟 (ϕ SUB) 为采样频率。 0: 用 ϕ 的 32 分频时钟采样 1: 用 ϕ 的 4 分频时钟采样
4	SUBSTP	0	R/W	控制子时钟振荡器 控制子时钟振荡器的运行 / 停止。不使用时钟时, 必须将其置为 1。 0: 运行子时钟振荡器 1: 停止子时钟振荡器
3	RFCUT	0	R/W	控制内置反馈电阻 输入外部时钟时, 选择是否使用主时钟振荡器的内置反馈电阻和占空比调整电路。 使用晶体振荡器时, 不要进行存取。 在输入外部时钟状态下, 设置此位后, 必须转移到软件待机模式。转移到软件待机模式时, 切换是否使用主时钟振荡器的内置反馈电阻及占空比调整电路。 0: 使用主时钟振荡器的内置反馈电阻及占空比调整电路 1: 不使用主时钟振荡器的内置反馈电阻及占空比调整电路
2	—	0	R/W	保留位 虽然可读取 / 写入, 但写入时, 总是作为 0。
1	STC1	0	R/W	设定倍频
0	STC0	0	R/W	设置内置于评估芯片的 PLL 电路倍频。 在转移到软件待机模式后, 指定的倍频变得有效。 在本 LSI 中, 必须使用 STC1 = STC0 = 1 的设置。因为复位后, 变为 STC1 = STC0 = 0, 因此, 复位后必须设置 STC1 = STC0 = 1。 00: x1 01: x2 (禁止设置) 10: x4 (禁止设置) 11: PLL 为旁路

【注】* 在向监视模式、子激活模式转移时, 必须设置为高速模式。

19.2 系统时钟振荡器

提供系统时钟的方法有，连接晶体振荡器和输入外部时钟两种。

19.2.1 连接晶体振荡器的方法

连接晶体振荡器的连接例如图 19.2 所示。请使用表 19.1 所列举的阻尼电阻 R_d 。另外，晶体振荡器必须使用 AT 截止并联谐振晶体。

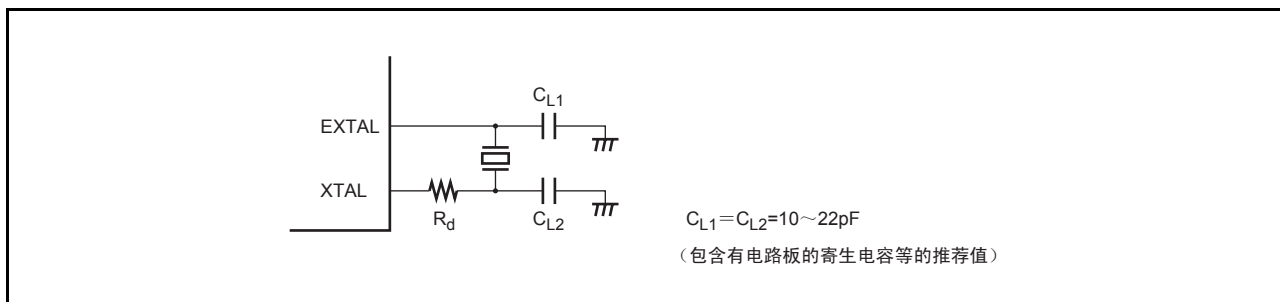


图 19.2 水晶振荡器连接例

表 19.1 阻尼电阻值

频率 (MHz)	6	8	10	13	16	20	24
R_d (Ω)	300	200	100	0	0	0	0

晶体振荡器的等效电路如图 19.3 所示。必须使用如表 19.2 所示特性的谐振器。

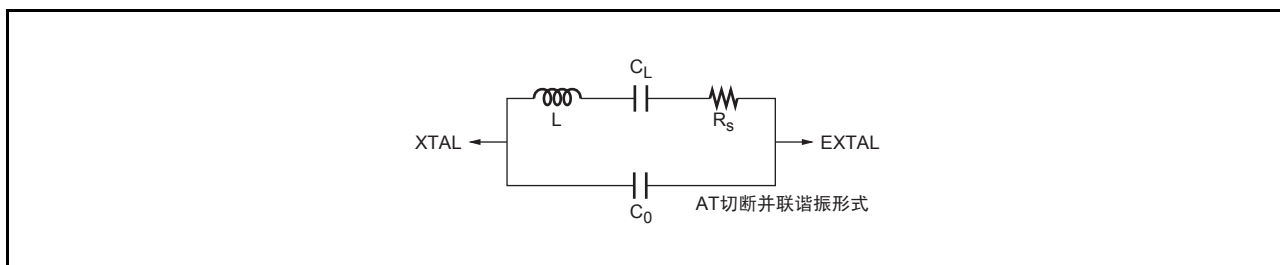


图 19.3 晶体振荡器的等效电路

表 19.2 晶体振荡器的特性

频率 (MHz)	6	8	10	13	16	20	24
R_{smax} (Ω)	100	80	60	60	50	40	40
C_{0max} (pF)	7						

19.2.2 输入外部时钟的方法

输入外部时钟的连接例如图 19.4 所示。在 XTAL 管脚为开路状态时，必须确保旁置电容在 10pF 以下。在待机模式、子激活模式、子睡眠模式及监视模式下，如果给 XTAL 管脚输入反相时钟，必须将外部时钟设置为高电平。

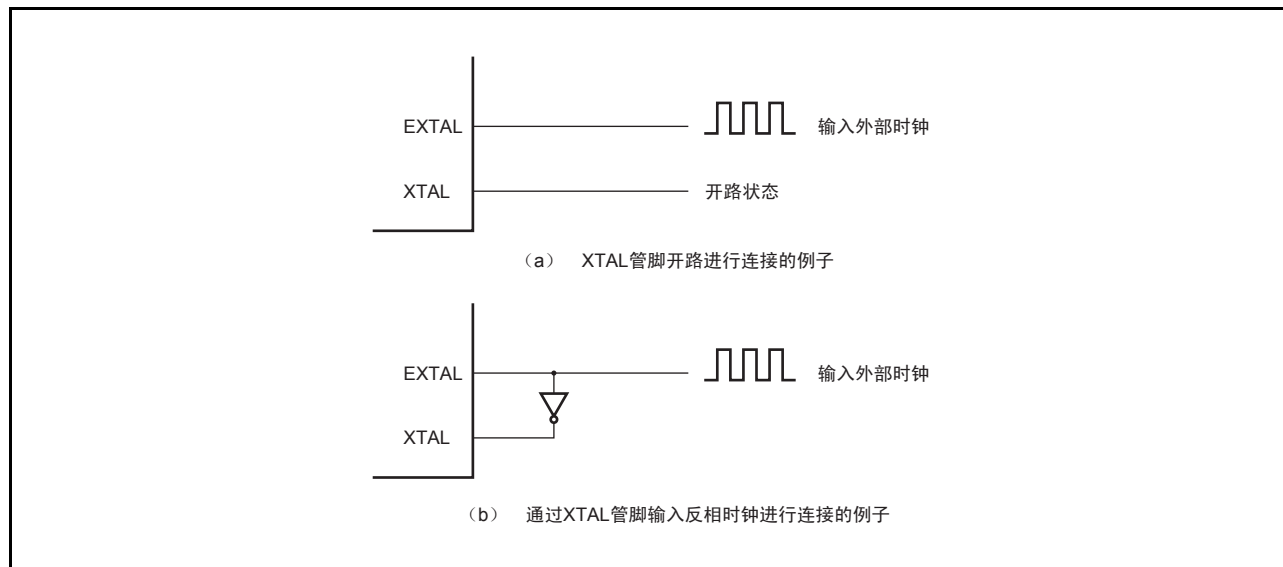


图 19.4 外部时钟的连接例

外部时钟的输入条件如表 19.3 所示。

表 19.3 外部时钟的输入条件

项目	符号	VCC = 2.4 ~ 3.6V		VCC = 2.7 ~ 3.6V		VCC = 3.0 ~ 3.6V		单位	测定条件
		min	max	min	max	min	max		
外部时钟输入低电平脉冲宽度	t_{EXL}	65	—	25	—	15.5	—	ns	图 19.5
外部时钟输入高电平脉冲宽度	t_{EXH}	65	—	25	—	15.5	—	ns	
外部时钟上升时间	t_{EXr}	—	15	—	6.25	—	5.25	ns	
外部时钟下降时间	t_{EXf}	—	15	—	6.25	—	5.25	ns	
时钟低电平脉冲宽度	t_{CL}	0.35	0.65	0.4	0.6	0.4	0.6	tcyc	图 22.3
时钟高电平脉冲宽度	t_{CH}	0.35	0.65	0.4	0.6	0.4	0.6	tcyc	

不使用占空比调整电路时的外部时钟输入条件如表 19.4 所示。

另外，不使用占空比调整电路时，根据输入波形，最高运行频率变低，这时必须注意。例如， $t_{EXL} = t_{EXH} = 20.8\text{ns}$ ， $t_{EXr} = t_{EXf} = 5.25\text{ns}$ 时，根据时钟周期时间 = 52.1ns，变为最高运行频率 = 19.2MHz。

表 19.4 外部时钟输入条件（未使用占空比调整电路）

项目	符号	VCC = 2.4 ~ 3.6V		VCC = 2.7 ~ 3.6V		VCC = 3.0 ~ 3.6V		单位	测定条件
		min	max	min	max	min	max		
外部时钟输入低电平脉冲宽度	t_{EXL}	80	—	31.25	—	20.8	—	ns	图 19.5
外部时钟输入高电平脉冲宽度	t_{EXH}	80	—	31.25	—	20.8	—	ns	
外部时钟上升时间	t_{EXr}	—	15	—	6.25	—	5.25	ns	
外部时钟下降时间	t_{EXf}	—	15	—	6.25	—	5.25	ns	

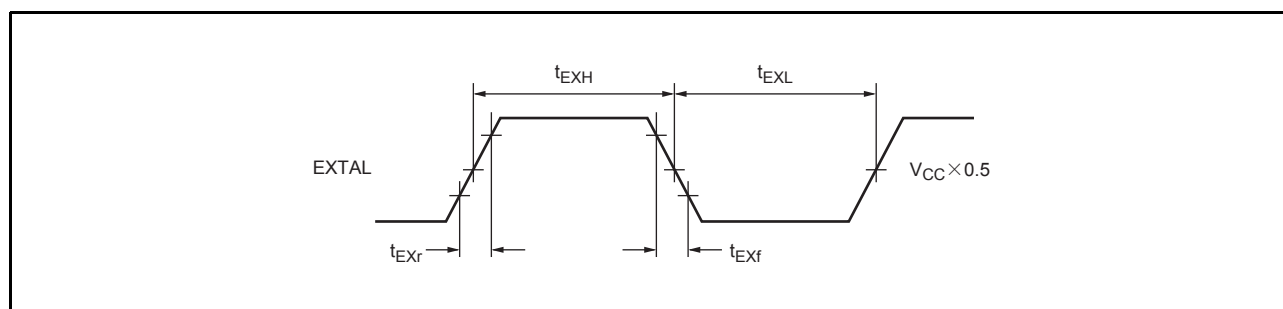


图 19.5 外部时钟输入时序

19.3 占空比调整电路

振荡频率在 5MHz 或以上时，占空比调整电路有效，调整系统时钟振荡器的输出时钟占空比，并生成系统时钟 (ϕ)。

19.4 中速时钟分频器

中速时钟分频器将系统时钟分频，并生成 $\phi/2$ ， $\phi/4$ ， $\phi/8$ ， $\phi/16$ ， $\phi/32$ 。

19.5 总线主控器时钟发生电路

总线主控器时钟选择电路，通过设置 SCKCR 的 SCK2 ~ SCK0 位，选择给总线主控器及 USB 提供的时钟。它们可从高速模式或中速时钟 ($\phi/2$ ， $\phi/4$ ， $\phi/8$ ， $\phi/16$ ， $\phi/32$) 中选择。

19.6 子时钟振荡器

19.6.1 连接 32.768KHz 连接晶体振荡器的方法

给子时钟分频器提供时钟时，连接 32.768KHz 晶体振荡器如图 19.6 所示。32.768KHz 晶体振荡器的等效电路如图 19.7 所示。

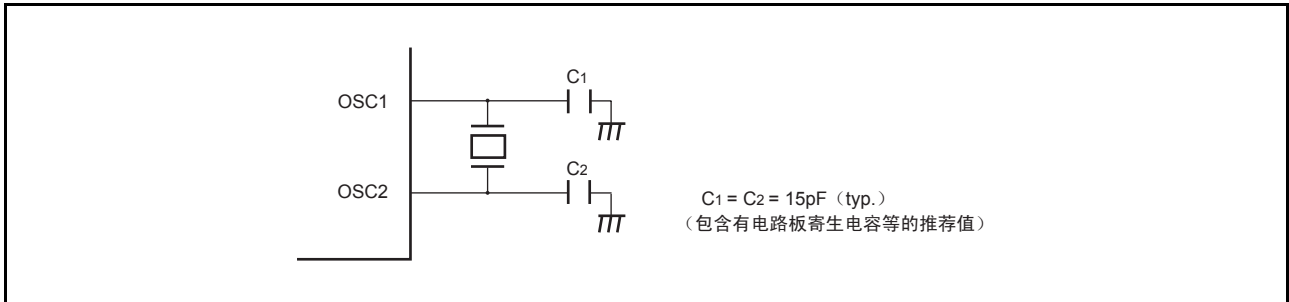


图 19.6 32.768KHz 晶体振荡器的连接例

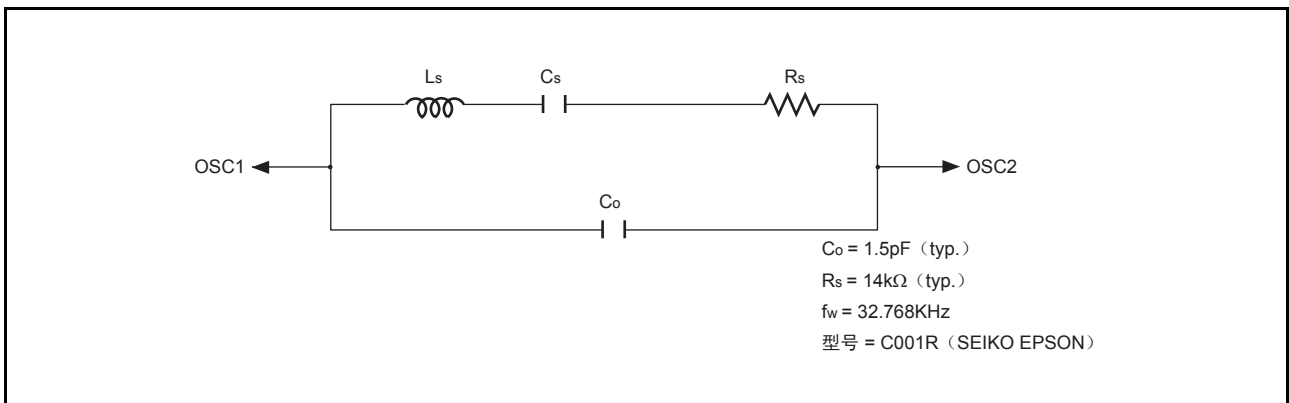


图 19.7 32.768KHz 晶体振荡器的等效电路

19.6.2 不使用子时钟的管脚处理

如图 19.8 所示，不使用子时钟时，必须将 OSC1 管脚连接到 Vss 上，使 OSC2 管脚开路，将 LPWRCR 的 SUBSTP 位置 1。不置 1 时，向低功耗模式的转换就有可能无法顺利进行。

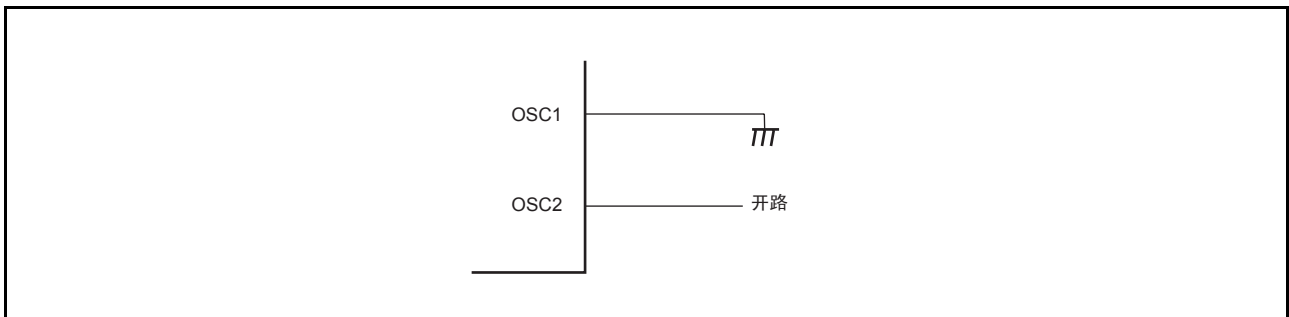


图 19.8 不使用子时钟时的管脚处理

19.7 子时钟波形成形电路

为了除去从 OSC1 管脚输入的子时钟噪音，用 ϕ 时钟的分频时钟进行采样。通过 LPWRCR 的 NESEL 位来设置采样频率。详细情况请参照 19.1.2 低功率控制寄存器（LPWRCR）。

在子激活模式、子睡眠模式及监视模式下，不可进行采样。

19.8 USB 专用 PLL 电路

PLL 电路使主时钟振荡器的 16MHz 或 24MHz 时钟实现 2 倍或 3 倍频，并生成 USB 运行用的 48MHz 时钟。

与是否使用 PLL 电路无关，必须将 PLLVCC 管脚、PLLVSS 管脚分别连接到 Vcc 接地（Vss）上。

使用 PLL 电路时，必须将 UCTLR 寄存器的 UCKS3 ~ UCKS0 位置位。详细情况请参照 14. 通用串行总线（USB）。

19.9 使用注意事项

19.9.1 振荡器注意事项

因为谐振振荡器的诸多特性密切关系到用户的电路板设计，因此希望用户参照本章介绍的振荡器的连接例，充分研究后再使用。振荡器的电路额定值取决于谐振器和安装电路的寄生电容等因素，因此必须和振荡器制造厂家磋商后再做决定。振荡器的外加电压不能超过最大额定值。

19.9.2 电路板设计注意事项

使用晶体振荡器时，须尽量将振荡器及负载电容设置在 XTAL，EXTAL 管脚附近。如图 19.9 所示振荡电路附近不要让信号线通过。否则，可能发生因感应而不能正确振荡的情况。OSC1、OSC2 管脚也要注意这些事项。

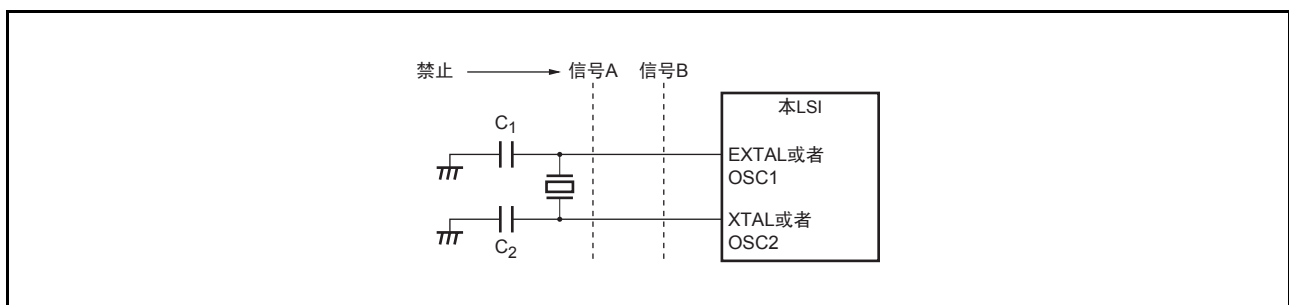


图 19.9 振荡电路的电路板设计注意事项

19.9.3 切换外部时钟注意事项

当 2 种以上的外部时钟（例：16MHz 和 13MHz）作为系统时钟使用，并切换输入时钟时，必须在软件待机模式下进行。

外部时钟切换电路例、外部时钟切换时序例分别如图 19.10，图 19.11 所示。

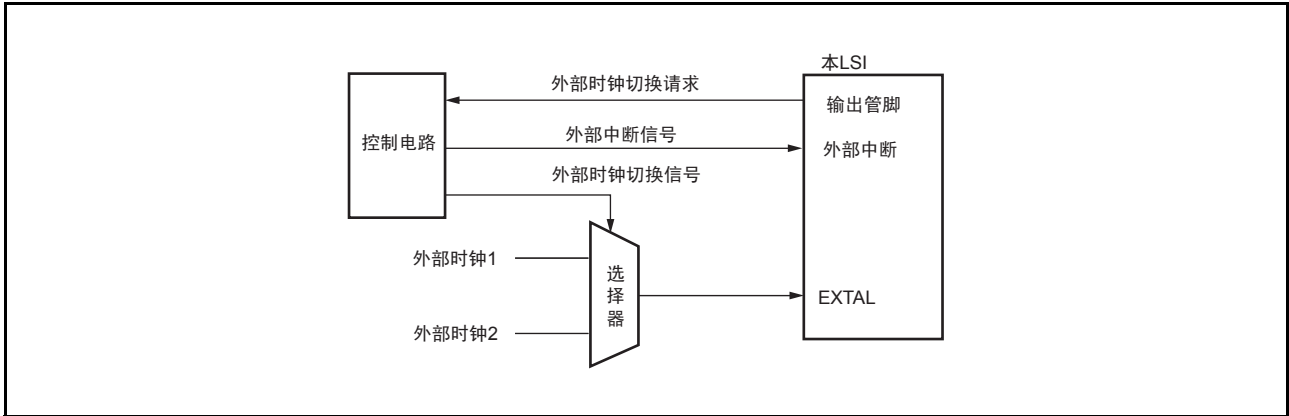


图 19.10 外部时钟切换电路例

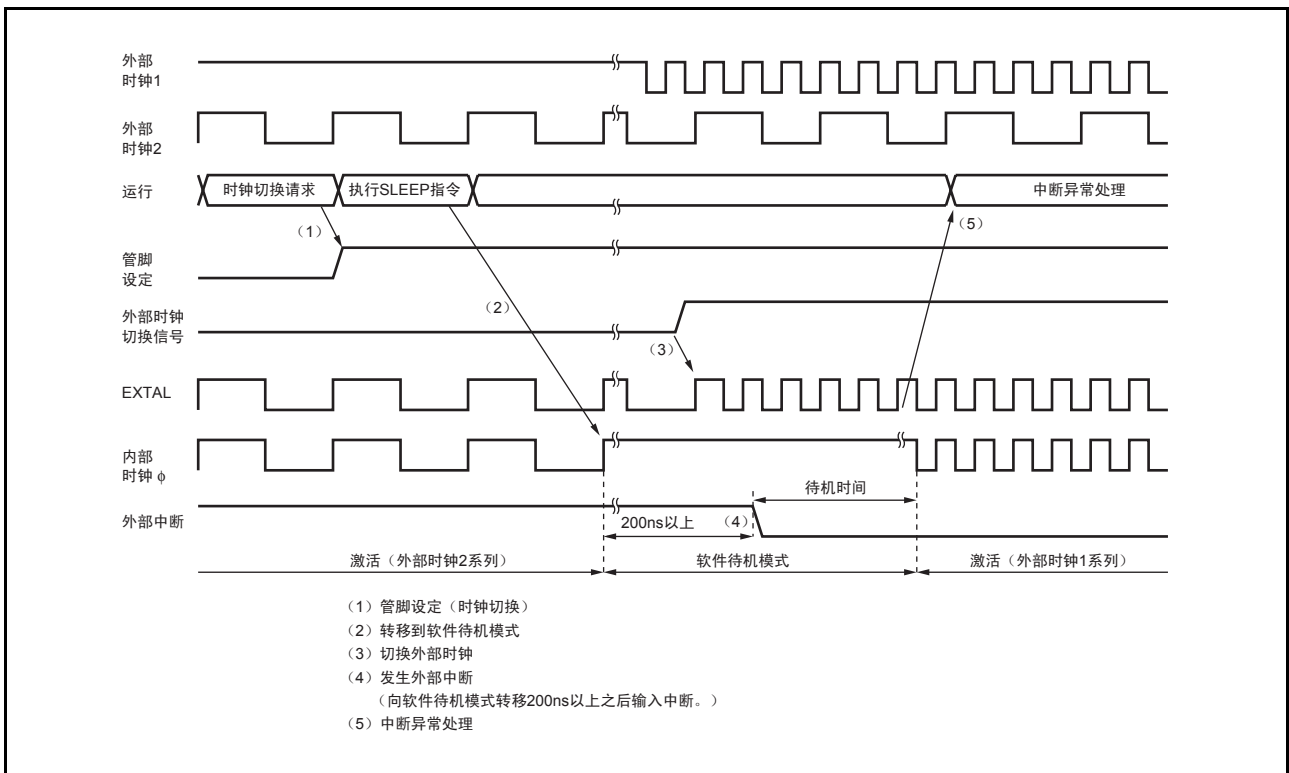


图 19.11 外部时钟切换时序例

20. 低功耗状态

本 LSI，除通常的程序运行状态外，还有停止 CPU 及振荡器的运行，降低功耗的低功耗状态。通过对 CPU，内部外围功能等的分别控制，可实现低功耗化。

本 LSI 的运行模式，除高速模式外还有：

- 中速模式
- 子激活模式
- 睡眠模式
- 子睡眠模式
- 监视模式
- 模块停止模式
- 软件待机模式
- 硬件待机模式

等低功耗状态。睡眠模式为 CPU 的状态，中速模式为 CPU 及总线主控器的状态，子激活模式为 CPU，总线主控器及内部外围功能的状态，模块停止状态为内部外围功能（包括除 CPU 以外的总线主控器）的状态。这些状态可进行组合设置。

复位后，进入高速模式，及除闪存和 DMAC 以外的模块停止模式。

各模式下的 LSI 的内部状态及低功耗模式转移条件分别如表 20.1，表 20.2 所示。另外，模式转移图如图 20.1 所示。

表 20.1 在各模式下本 LSI 的内部状态

功能		高速	中速	睡眠	模块停止	监视	子激活	子睡眠	软件待机	硬件待机	
系统时钟振荡器		运行	运行	运行	运行	停止	停止	停止	停止	停止	
子时钟振荡器		运行 / 停止	运行 / 停止	运行 / 停止	运行 / 停止	运行	运行	运行	运行 / 停止	停止	
CPU 运行	指令	运行	中速	停止	运行	停止	子时钟 运行	停止	停止	停止	
	寄存器		运行	保持		保持		保持			不定
RAM		运行	运行	运行	运行	保持	运行	保持	保持	保持	
I/O		运行	运行	运行	运行	保持	运行	运行	保持	高阻抗	
外部 中断	NMI	运行	运行	运行	运行	运行	运行	运行	运行	停止	
	IRQ0 ~ 4、7										
外围 功能	DMAC		运行	中速 运行	运行 (保持)	停止 (保持)	停止 (保持)	停止 (保持)	停止 (保持)	停止 (复位)	
	WDT		运行	运行	运行	运行	停止 (保持)	子时钟 运行	子时钟 运行	停止 (保持)	停止 (复位)
	RTC	时钟运行	子时钟 运行	子时钟 运行	子时钟 运行	停止 (保持)	子时钟 运行	子时钟 运行	子时钟 运行	子时钟运 行 / 停止	停止 (复位)
		自由运行定 时器运行	运行	运行	运行	停止 (保持)	停止 (保持)	停止 (保持)	停止 (保持)	停止 (保持)	停止 (复位)
	TPU		运行	运行	运行	停止 (保持)	停止 (保持)	停止 (保持)	停止 (保持)	停止 (保持)	停止 (复位)
	SCI					停止 (保持)	停止 (保持)	停止 (保持)	停止 (保持)	停止 (保持)	停止 (复位)
	A/D		运行	运行	运行	停止 (复位)	停止 (复位)	停止 (复位)	停止 (复位)	停止 (复位)	停止 (复位)
	USB		运行	不保证 运行	运行	停止 (保持)	不保证运行 必须设置为模块停止状态。			停止 (保持)	停止 (复位)
PLL 电路		停止				停止					

【注】 停止 (保持) 时, 保持内部寄存器的值。内部状态运行中断。

停止 (复位) 时, 初始化内部寄存器的值及内部状态。

为模块停止模式时, 只停止已进行停止设置的模块。(复位或保持)。

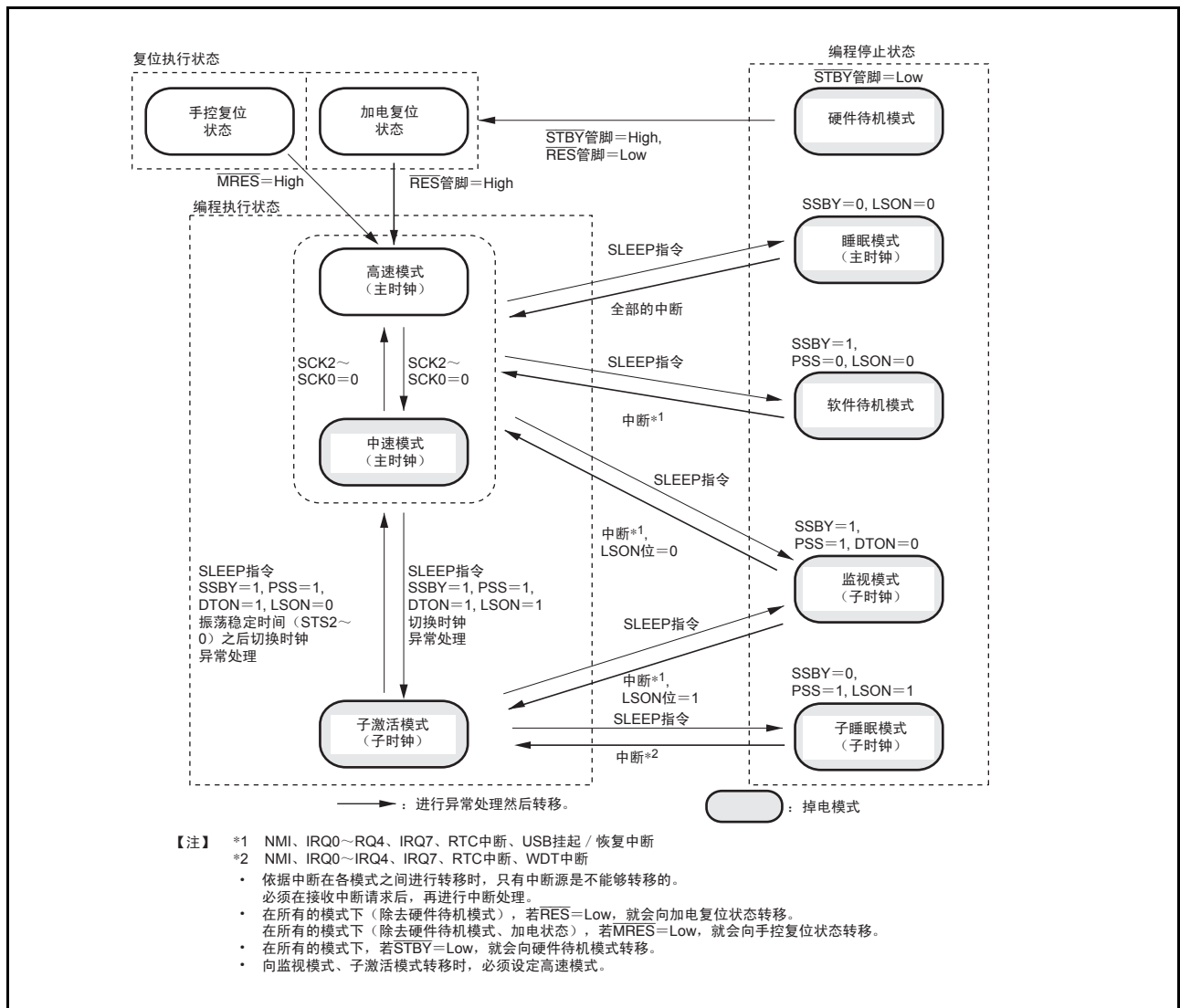


图 20.1 模式转移图

表 20.2 低功耗模式转移条件

转移前的状态	转移时的控制位状态				通过 SLEEP 指令转移后的状态	通过中断恢复后的状态
	SSBY	PSS	LSON	DTON		
高速 / 中速	0	×	0	×	睡眠	高速 / 中速
	0	×	1	×	—	—
	1	0	0	×	软件待机	高速 / 中速
	1	0	1	×	—	—
	1	1	0	0	监视	高速
	1	1	1	0	监视	子激活
	1	1	0	1	—	—
	1	1	1	1	子激活	—
子激活	0	0	×	×	—	—
	0	1	0	×	—	—
	0	1	1	×	子睡眠	子激活
	1	0	×	×	—	—
	1	1	0	0	监视	高速
	1	1	1	0	监视	子激活
	1	1	0	1	高速	—
	1	1	1	1	—	—

【符号说明】

×: Don't care

—: 不可设置

20.1 寄存器说明

与低功耗模式有关的寄存器有以下几种。关于低功率控制寄存器（LPWRCR），请参照 19.1.2 低功率控制寄存器（LPWRCR）。关于系统时钟控制寄存器（SCKCR），请参照 19.1.1 系统时钟控制寄存器（SCKCR）。

- 待机控制寄存器（SBYCR）
- 系统时钟控制寄存器（SCKCR）
- 低功率控制寄存器（LPWRCR）
- 定时器控制 / 状态寄存器（TCSR_1）
- 模块停止控制寄存器 A（MSTPCRA）
- 模块停止控制寄存器 B（MSTPCRB）
- 模块停止控制寄存器 C（MSTPCRC）
- 扩展模块停止寄存器（EXMDLSTP）

20.1.1 待机控制寄存器 (SBYCR)

SBYCR 执行软件待机模式的控制。

位	位名称	初始值	R/W	说明
7	SSBY	0	R/W	<p>软件待机 指定执行 SLEEP 指令后的转移模式。</p> <p>0: 在高速模式, 中速模式下, 执行 SLEEP 指令后, 转移睡眠模式; 在子激活模式下, 执行 SLEEP 指令后, 转移到子睡眠模式</p> <p>1: 在高速模式, 中速模式下, 执行 SLEEP 指令后, 转移到软件待机模式, 子激活模式或监视模式; 在子激活模式下, 执行 SLEEP 指令后, 转移到监视模式或高速模式另外, 通过外部中断解除软件待机模式, 并转移到通常运行时, 该位保持不变。清除时, 必须写入 0。</p>
6 5 4	STS2 STS1 STS0	0 0 0	R/W R/W R/W	<p>待机定时器选择 2 ~ 0</p> <p>通过外部中断解除软件待机模式, 监视模式, 子激活模式时, 选择到时钟稳定为止的 MCU 待机时间。为晶体振荡器时, 请参照表 20.3 和表 22.4, 根据运行频率, 待机时间需在 $t_{osc2}ms$ (振荡稳定时间) 以上。为外部时钟时, 可任意选择待机时间。但在 F-ZTAT 版中, 为外部时钟时, 不可使用 16 个状态的待机时间。需选在 100μs 以上。</p> <p>000: 待机时间 = 8192 个状态 001: 待机时间 = 16384 个状态 010: 待机时间 = 32768 个状态 011: 待机时间 = 65536 个状态 100: 待机时间 = 131072 个状态 101: 待机时间 = 262144 个状态 110: 待机时间 = 2048 个状态 111: 待机时间 = 16 个状态</p>
3	OPE	1	R/W	<p>输出端口允许</p> <p>在软件待机模式、监视模式及直接转移时, 选择是高阻抗还是保持地址总线及总线控制信号 ($\overline{CS0} \sim \overline{CS7}$、$\overline{AS}$、$\overline{RD}$、$\overline{HWR}$、$\overline{LWR}$) 的输出。</p> <p>0: 高阻抗 1: 保持输出状态</p>
2~0	—	全为 0	—	<p>保留位</p> <p>读取时, 总是读出 0。写入定义为无效。</p>

20.1.2 定时器控制 / 状态寄存器 (TCSR_1)

TCSR_1 控制低功耗模式转移时的运行。

位	位名称	初始值	R/W	说明
7~5	—	全为 0	—	保留位 写入时，总是写入 0。
4	PSS	0	R/W	<p>预定标器选择</p> <p>0: 在高速模式或中速模式下，执行 SLEEP 指令时，转移到睡眠模式、软件待机模式。</p> <p>1: 在高速模式或中速模式下，执行 SLEEP 指令时，转移到睡眠模式、监视模式、子激活模式。</p> <ul style="list-style-type: none"> • 在子激活模式下，执行 SLEEP 指令时，转移到子睡眠模式、监视模式、高速模式。 <p>另外，TCSR_1 不容易改写，写入方法与一般的寄存器不同。请按照以下方法进行读取 / 写入。</p> <p>a. 写入 必须使用字传送指令，将高位字节作为 H'A5，低位字节作为写入数据进行传送。（为 PSS 位写入 1 时为 H'A510）</p> <p>b. 读取 可用一般寄存器的读取方法进行读取。</p>
3~0	—	全为 0	—	保留位 写入时，总是写入 0。

20.1.3 模块停止控制寄存器 A ~ C (MSTPCRA ~ MSTPCRC)

MSTPCR 包含 3 个 8 位可读取 / 写入的寄存器，它来控制模块停止模式。为 1 时，对应的寄存器变为模块停止模式，清除时，解除模块停止模式。MSTPCR 只能在 CPU 中进行写入。

- MSTPCRA

位	位名称	初始值	R/W	对象模块
7	MSTPA7	0	R/W	DMA 控制器 (DMAC)
6	MSTPA6*	0	R/W	—
5	MSTPA5	1	R/W	16 位定时器脉冲单位 (TPU)
4	MSTPA4*	1	R/W	—
3	MSTPA3*	1	R/W	—
2	MSTPA2*	1	R/W	—
1	MSTPA1	1	R/W	A/D 转换器
0	MSTPA0*	1	R/W	—

- MSTPCRB

位	位名称	初始值	R/W	对象模块
7	MSTPB7	1	R/W	串行通信接口 _0 (SCI_0)
6	MSTPB6*	1	R/W	—
5	MSTPB5	1	R/W	串行通信接口 _2 (SCI_2)
4	MSTPB4*	1	R/W	—
3	MSTPB3*	1	R/W	—
2	MSTPB2*	1	R/W	—
1	MSTPB1*	1	R/W	—
0	MSTPB0	1	R/W	USB

• MSTPCRC

位	位名称	初始值	R/W	对象模块
7	MSTPC7*	1	R/W	—
6	MSTPC6*	1	R/W	—
5	MSTPC5	1	R/W	—
4	MSTPC4*	1	R/W	—
3	MSTPC3*	1	R/W	—
2	MSTPC2*	1	R/W	—
1	MSTPC1	0	R/W	闪存（掩模型 ROM 版中为保留位，设置定义为无效） 【注】 必须在内部 RAM 和外部存储器的程序执行过程中，进行闪存的模块停止设置。如果通过闪存上的程序，停止闪存，则停止以后的程序，并进入死锁状态。 闪存模块停止模式的使用例如图 20.2 所示。
0	MSTPC0*	1	R/W	—

【注】 * MSTPA6 可读取 / 写入，初始值为 0。写入时通常为 1。
MSTPA4 ~ MSTPA2, MSTPA0, MSTPB6, MSTPB4 ~ MSTPB1, MSTPC7 ~ MSTPC2, MSTPC0 可读取 / 写入，初始值为 1。写入时通常为 1。

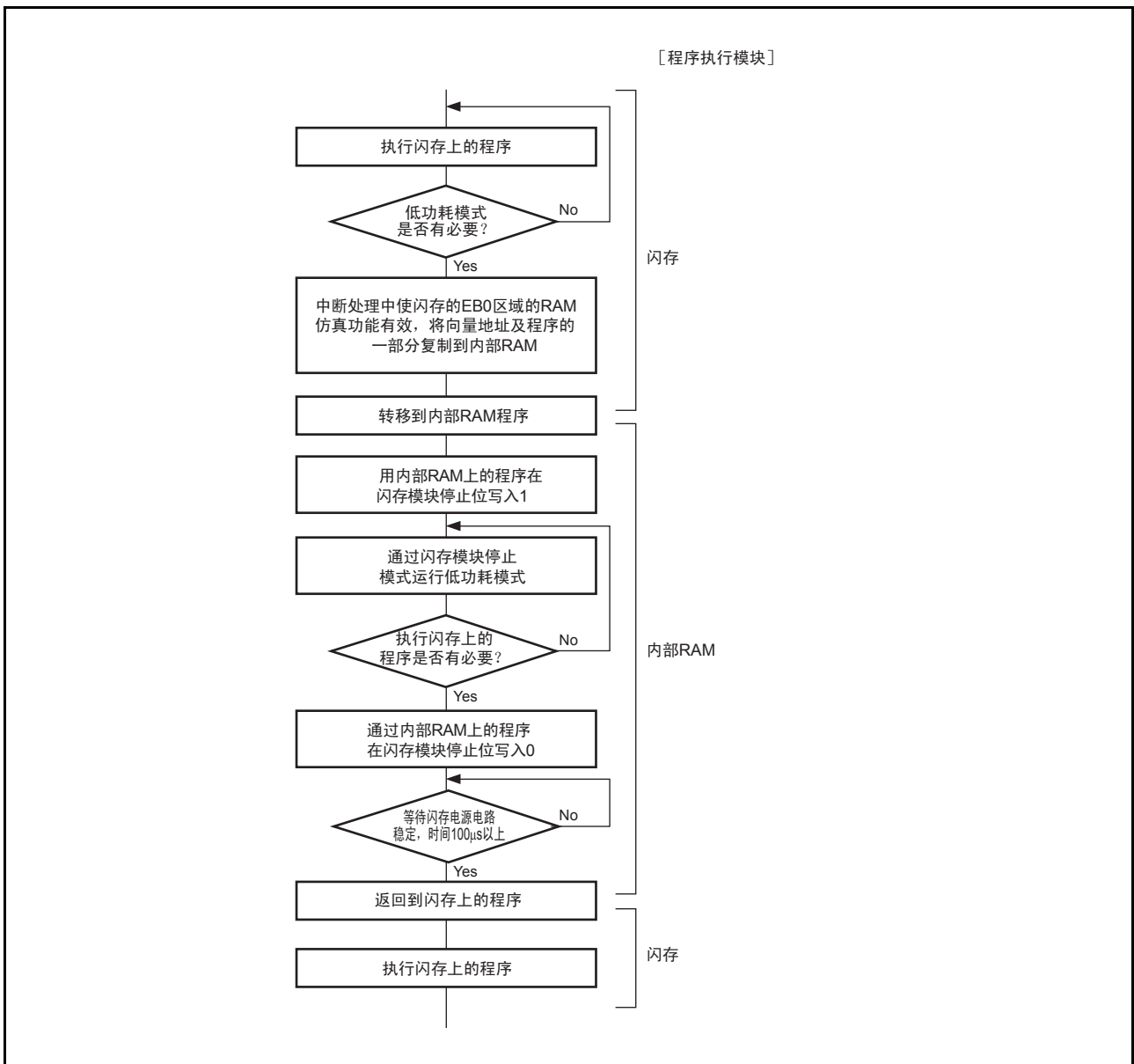


图 20.2 闪存模块停止模式的使用例

20.1.4 扩展模块停止寄存器 (EXMDLSTP)

EXMDLSTP 控制 RTC 及 USB 的时钟提供。为 1 时，对应的模块进入模块停止模式，清除时，解除模块停止模式。EXMDLSTP 只能在 CPU 中进行写入。

位	位名称	初始值	R/W	对象模块
7~2	—	不定	—	保留位 读取时，读出不定值。写入定义为无效。
1	RTCSTOP	0	R/W	RTC
0	USBSTOP1	0	R/W	USB

20.2 中速模式

如果将 SCKCR 的 SCK2 ~ SCK0 位置 1，则在当前总线周期结束时变为中速模式。中速模式下，CPU 以通过 SCK2 ~ SCK0 位指定的运行时钟为基准 ($\phi/2$, $\phi/4$, $\phi/8$, $\phi/16$, $\phi/32$) 运行。CPU 以外的总线主控器 (DMAC) 也在中速模式下运行。

总线主控器以外的内部外围功能通常以高速时钟 (ϕ) 为基准运行。

对于中速模式下的数据总线运行时钟，以指定的数据状态进行总线存取。例如，作为运行时钟选择 $\phi/4$ 时，内部存储器变为 4 态存取，内部 I/O 寄存器变为 8 态存取。

通过全部清除 SCK2 ~ SCK0 位，来解除中速模式，在中速模式总线周期结束时，转移到高速模式，并解除中速模式。

如果在 SBYCR 的 SSBY 位及 LPWRCR 的 LSON 位全部清 0 的状态下，执行 SLEEP 指令，则转移到睡眠模式。通过中断解除睡眠模式时，恢复到中速模式。

另外，如果在 SSBY 位置 1，LSON 位和 TCSR_1 的 PSS 位清 0 的状态下，执行 SLEEP 指令，则转移到软件待机模式。通过外部中断解除软件待机模式时，恢复到中速模式。

$\overline{\text{RES}}$ 管脚、 $\overline{\text{MRES}}$ 管脚 * 置为低电平时，转移到复位状态，并解除中速模式。同样通过看门狗定时器溢出产生的复位也可解除中速模式。

如果将 $\overline{\text{STBY}}$ 管脚置为低电平，则转移到硬件待机模式。

向中速模式转移，解除的时序如图 20.3 所示。

【注】 * 仅限于 H8S/2218 群使用。

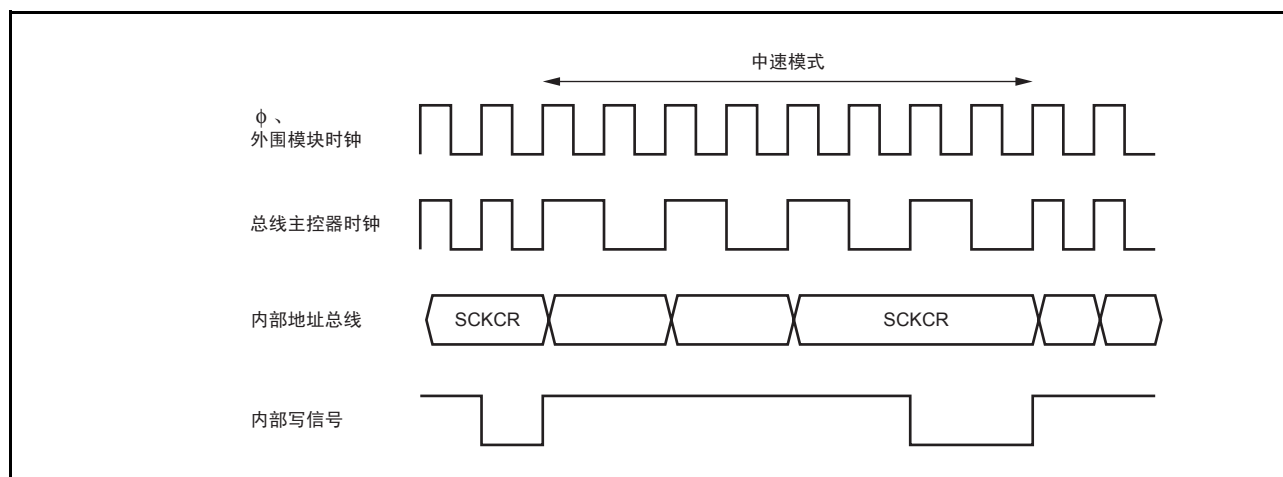


图 20.3 中速模式的转移解除时序

20.3 睡眠模式

20.3.1 向睡眠模式的转移

如果在清除 SBYCR 的 SSBY 位和 LPWRCR 的 LSON 位的状态下，执行 SLEEP 指令，则 CPU 进入睡眠模式。为睡眠模式时，虽然停止 CPU 的运行，但保持 CPU 的内部寄存器的内容。CPU 以外的外围功能继续运行。

20.3.2 睡眠模式的解除

通过所有中断 $\overline{\text{RES}}$ 、 $\overline{\text{MRES}}$ * 或 $\overline{\text{STBY}}$ 管脚，解除睡眠模式。

- 通过中断解除

如果发生中断，则解除睡眠模式，并开始中断异常处理。禁止中断时，或者除 NMI 以外的中断被 CPU 屏蔽时，不解除睡眠模式。

- 通过 $\overline{\text{RES}}$ 管脚、 $\overline{\text{MRES}}$ 管脚 * 解除

如果将 $\overline{\text{RES}}$ 管脚、 $\overline{\text{MRES}}$ 管脚 * 置为低电平，则进入复位状态。如果在规定的复位输入时间以后，将 $\overline{\text{RES}}$ 管脚， $\overline{\text{MRES}}$ 管脚 * 置为高电平，则 CPU 开始复位异常处理。

- 通过 $\overline{\text{STBY}}$ 管脚解除

如果将 $\overline{\text{STBY}}$ 管脚置为低电平，则转移到硬件待机模式。

【注】 * 仅限于 H8S/2218 群使用。

20.4 软件待机模式

20.4.1 向软件待机模式转移

如果在 SBYCR 的 SSBY 位置 1，LPWRCR 的 LSON 位和 TCSR_1 的 PSS 位清 0 的状态下，执行 SLEEP 指令，则进入软件待机模式。在该模式下，停止 CPU、内部外围功能及振荡器的全部功能。但 CPU 的内部寄存器的内容、内部 RAM 的数据、内部外围功能（除 A/D 转换器）及 I/O 端口的状态保持不变。本模式下，由于振荡器停止，因此功耗明显降低。

20.4.2 软件待机模式的解除

通过外部中断（NMI 管脚， $\overline{\text{IRQ0}} \sim \overline{\text{IRQ4}}$ 管脚、 $\overline{\text{IRQ7}}$ 管脚），RTC 中断（ $\overline{\text{IRQ5}}$ 信号），USB 挂起 / 恢复中断（ $\overline{\text{IRQ6}}$ 信号）、 $\overline{\text{RES}}$ 管脚， $\overline{\text{MRES}}$ 管脚 * 或 $\overline{\text{STBY}}$ 管脚，解除软件待机模式。

- 通过中断解除

如果输入 NMI， $\overline{\text{IRQ0}} \sim \overline{\text{IRQ7}}$ 中断请求，则开始时钟振荡，经过 SBYCR 的 STS2 ~ STS0 位设定的时间后，给整个 LSI 提供稳定时钟，解除软件待机模式，并开始中断异常处理。

通过 $\overline{\text{IRQ0}} \sim \overline{\text{IRQ7}}$ 中断，解除软件待机模式时，将对应的允许位置 1，且设置不可发生比 $\overline{\text{IRQ0}} \sim \overline{\text{IRQ7}}$ 中断优先级高的中断。在 CPU 屏蔽中断时，不可解除软件待机模式。

- 通过 $\overline{\text{RES}}$ 管脚、 $\overline{\text{MRES}}$ 管脚 * 解除

如果将 $\overline{\text{RES}}$ 管脚、 $\overline{\text{MRES}}$ 管脚 * 置为低电平，则开始时钟振荡。时钟开始振荡的同时，给整个 LSI 提供时钟。这时，到时钟振荡稳定为止， $\overline{\text{RES}}$ 管脚、 $\overline{\text{MRES}}$ 管脚 * 必须保持低电平。如果将 $\overline{\text{RES}}$ 管脚、 $\overline{\text{MRES}}$ 管脚 * 置为高电平，则 CPU 开始复位异常处理。

- 通过管脚解除

如果将管脚置为低电平，则转移到硬件待机模式。

【注】 * 仅限于 H8S/2218 群使用。

20.4.3 软件待机模式解除后的振荡稳定时间的设置

SBYCR 的 STS2 ~ STS0 位的设置如下所示。

- 晶体振荡器时
为了使待机时间在 t_{osc2} ms（振荡稳定时间）以上，需设置 STS2 ~ STS0 位。
运行频率及设置 STS2 ~ STS0 位的待机时间如表 20.3 所示。
- 外部时钟时
可设置任意值。通常情况下，最好使用最小时间。但在 F-ZTAT 版中，需将待机时间设置在 $100\mu\text{s}$ 或以上（闪存的电源稳定时间）。

表 20.3 振荡稳定时间的设置

STS2	STS1	STS0	待机时间	24 MHz	20 MHz	16 MHz	13 MHz	10 MHz	8 MHz	6 MHz	4 MHz	2 MHz	单位
0	0	0	8192 状态	0.34	0.41	0.51	0.63	0.82	1.0	1.4	2.0	4.1	ms
		1	16384 状态	0.68	0.82	1.0	1.3	1.6	2.0	2.7	4.1	8.2	
	1	0	32768 状态	1.4	1.6	2.0	2.5	3.3	4.1	5.5	8.2	16.4	
		1	65536 状态	2.7	3.3	4.1	5.0	6.6	8.2	10.9	16.4	32.8	
1	0	0	131072 状态	5.5	6.6	8.2	10.1	13.1	16.4	21.8	32.8	65.5	μs
		1	262144 状态	10.9	13.1	16.4	20.2	26.2	32.8	43.7	65.5	131.1	
	1	0	2048 状态	0.09	0.10	0.13	0.16	0.20	0.26	0.34	0.51	1.0	
		1	16 状态	0.67	0.80	1.0	1.2	1.6	2.0	2.7	4.0	8.0	

【注】 ：推荐设置时间（条件请参照表 22.4 时钟时序 t_{osc2} 。）

20.4.4 软件待机模式的应用例

如图 20.4 所示，在 NMI 管脚的下沿，转移到软件待机模式，在 NMI 管脚的上升沿，解除软件待机模式。

例子中，在 SYSCR 的 NMIEG 位清 0（指定下降沿）的状态下，接受 NMI 中断后，将 NMIEG 位置 1（指定上升沿），SSBY 位置 1 后、执行 SLEEP 指令，并转移到软件待机模式。

此后，在 NMI 管脚的上升沿，解除软件待机模式。

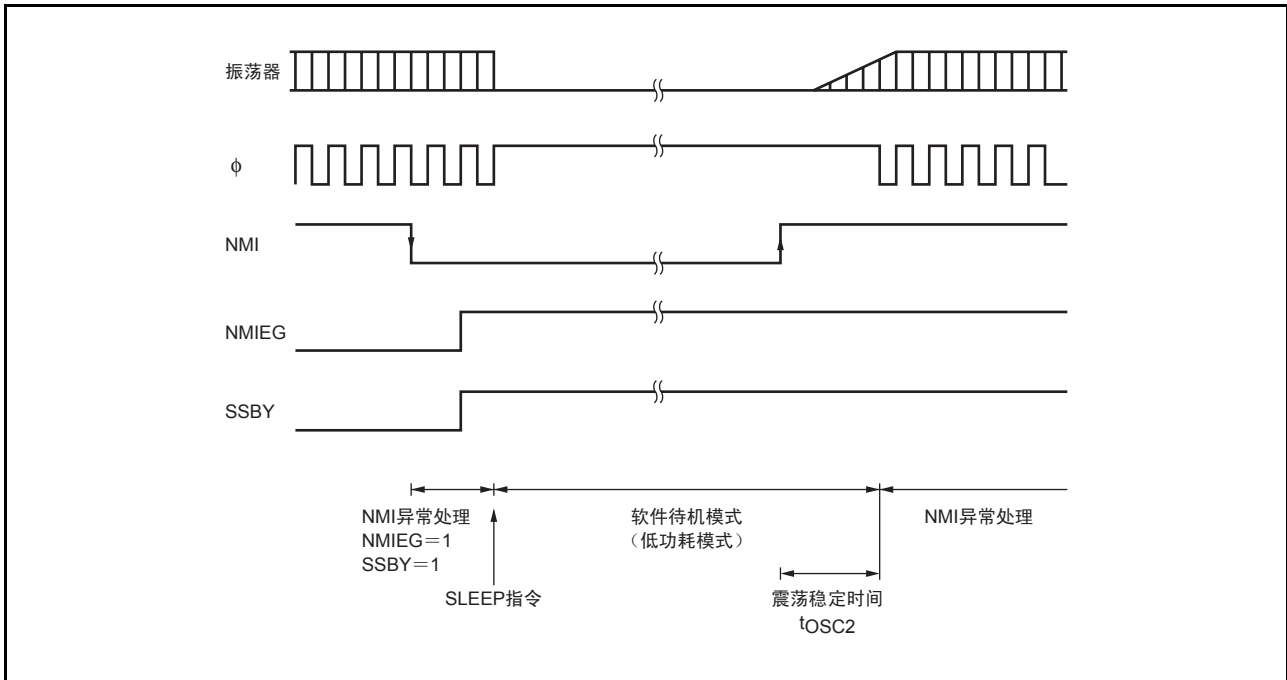


图 20.4 软件待机模式的应用例

20.5 硬件待机模式

20.5.1 向硬件待机模式转移

如果将 $\overline{\text{STBY}}$ 管脚置为低电平，任何状态都会变为硬件待机模式。

在硬件待机模式下，由于所有功能均进入复位状态，且运行停止，因此功耗明显降低。这时，只要在规定的电压的范围内，就保持内部 RAM 的数据。I/O 端口进入高阻抗状态。

为了保持内部 RAM 的数据，必须在 $\overline{\text{STBY}}$ 管脚置为低电平前，将 SYSCR 的 RAME 位清 0。另外，在硬件待机模式中，不要改变模式管脚（MD2 ~ MD0）的状态。

20.5.2 硬件待机模式的解除

通过 $\overline{\text{STBY}}$ 管脚和 $\overline{\text{RES}}$ 管脚，解除硬件待机模式。在 $\overline{\text{RES}}$ 管脚为低电平的状态下，如果将 $\overline{\text{STBY}}$ 管脚置为高电平，则进入复位状态，开始时钟振荡。此时，到时钟振荡稳定为止（晶体振荡时，震荡稳定时间在 t_{OSC1} 以上） $\overline{\text{RES}}$ 管脚必须保持低电平。此后，如果 $\overline{\text{RES}}$ 管脚置为高电平，则经过复位异常处理状态转移到程序执行状态。

20.5.3 硬件待机模式的时序

硬件待机模式的时序如图 20.4 所示。在 $\overline{\text{RES}}$ 管脚置为低电平以后，如果将 $\overline{\text{STBY}}$ 管脚置为低电平，则转移到硬件待机模式。将 $\overline{\text{STBY}}$ 管脚置为高电平，并经过时钟振荡稳定时间后，通过将 $\overline{\text{RES}}$ 管脚从低电平置为高电平，解除硬件待机模式。

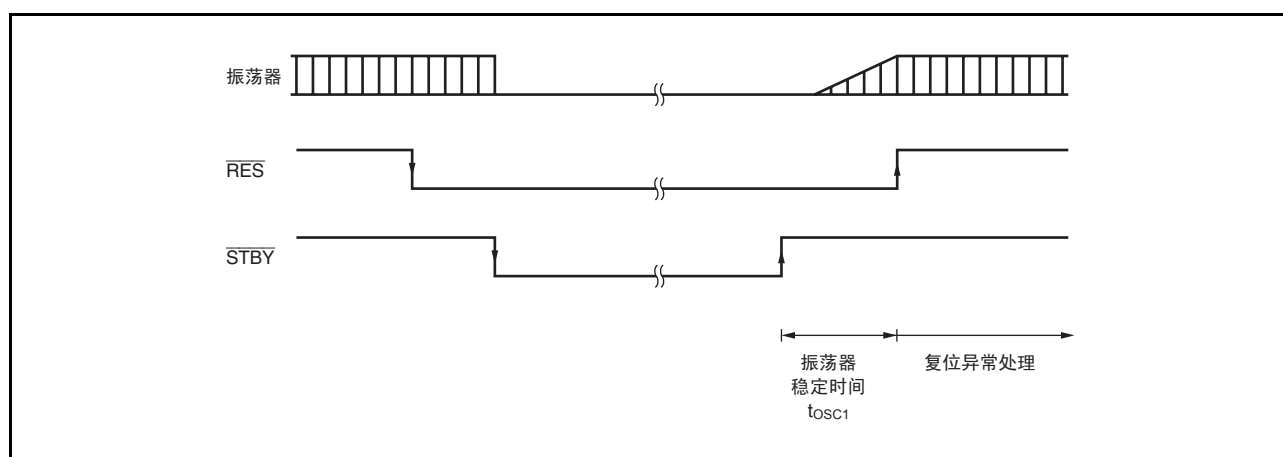


图 20.5 硬件待机模式的时序

20.5.4 有关硬件待机模式的转移时序

20.5.4.1 硬件待机模式的转移时序

(1) 在 SYSCR 的 RAME 位置 1 的状态下，保持 RAM 的内容时

如图 20.6 所示， $\overline{\text{STBY}}$ 信号变为低电平之前， $\overline{\text{RES}}$ 信号必须保持至少 10 个状态的低电平。另外，在 $\overline{\text{RES}}$ 信号变为低电平之后， $\overline{\text{STBY}}$ 信号必须等待至少 0ns，才能变为高电平。

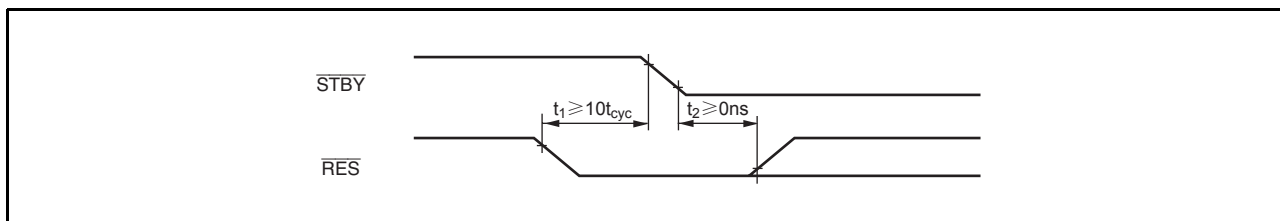


图 20.6 硬件待机模式的转移时序

(2) 在 SYSCR 的 RAME 位清 0 的状态下，或不保持 RAM 的内容时

不必像 (1) 那样将 $\overline{\text{RES}}$ 信号置为低电平。

20.5.4.2 从硬件待机模式恢复的时序

在 $\overline{\text{STBY}}$ 信号变为高电平执行加电复位之前，要将 $\overline{\text{RES}}$ 信号置为低电平需等待 100ns 以上。

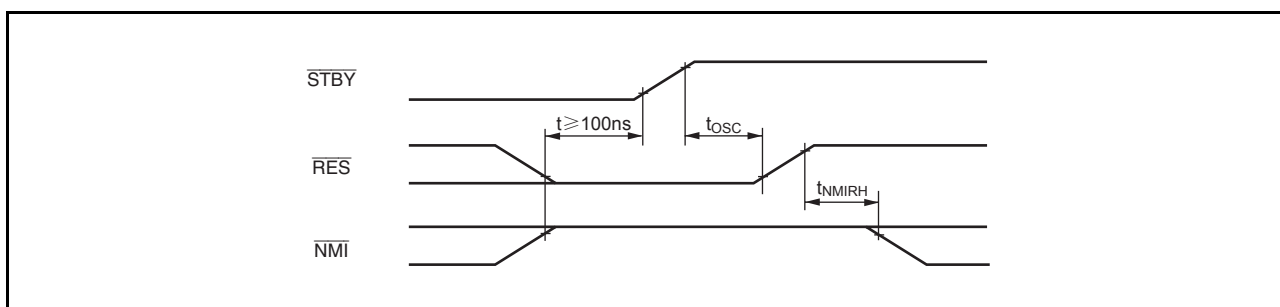


图 20.7 从硬件待机模式恢复的时序

20.6 模块停止模式

可以以内部外围功能的模块为单位设置模块停止模式。

如果将 MSTPCR 对应的 MSTP 位置 1，则在总线周期结束时停止模块运行，并转移到模块停止模式。此时 CPU 独立继续运行。

通过将对应的 MSTP 位清 0，解除模块停止模式，并在总线周期结束时，重新开始运行。在模块停止模式下，保持除 A/D 转换器以外的所有模块的内部状态。

复位解除后，除 DMAC 和闪存以外所有模块均变为模块停止状态。

如果该寄存器的模块设置为模块停止模式，则不可读取 / 写入。

另外，因为在全部模块停止的状态下，如果转移到睡眠模式，则停止总线控制器及 I/O 端口的运行，因此，可更明显的降低功耗。

20.7 监视模式

20.7.1 向监视模式转移

在高速或子激活模式下，如果 SBYCR 的 SSBY 位 = 1，LPWRCR 的 DTON 位 = 0，TCSR_1 的 PSS 位 = 1 时，执行 SLEEP 指令，则 CPU 转移到监视模式。

为监视模式时，CPU 停止运行。另外，RTC 以外的外围功能也停止运行。CPU 的内部寄存器的内容，内部 RAM 的数据，内部外围功能（A/D 转换器除外）和 I/O 端口的状态被保持。转移到监视模式时，必须将 SCKCR 的 SCK2 ~ SCK0 各位清 0。

20.7.2 监视模式的解除

通过中断（WOVI 中断，NMI 管脚， $\overline{\text{IRQ0}} \sim \overline{\text{IRQ7}}$ ） $\overline{\text{RES}}$ 管脚、 $\overline{\text{MRES}}$ 管脚*、或 $\overline{\text{STBY}}$ 管脚，解除监视模式。

- 通过中断解除

如果发生中断，则解除监视模式，LPWRCR 的 LSON 位 = 0 时，转移到高速或中速模式；LSON 位 = 1 时，转移到子激活模式。转移到高速模式时，经过 SBYCR 的 STS2 ~ STS0 设置的时间后，给整个 LSI 提供稳定时钟，并开始中断异常处理。此外，IRQ0 ~ IRQ7 中断相对应的允许位 / 管脚功能切换位被清 0 时，通过中断允许寄存器，禁止接受内部外围功能产生的中断时，或通过 CPU 屏蔽中断时，不能解除监视模式。

从监视模式向高速模式转移时，时钟振荡稳定时间的设置，请参照 20.4.3 软件待机模式解除后的振荡稳定时间的设置。

- 通过 $\overline{\text{RES}}$ 管脚、 $\overline{\text{MRES}}$ 管脚*解除

关于通过 $\overline{\text{RES}}$ 管脚、 $\overline{\text{MRES}}$ 管脚*解除的情况，请参照 20.4.2 软件待机模式的解除。

【注】* 仅限于 H8S/2218 群使用。

- 通过 $\overline{\text{STBY}}$ 管脚解除

如果将 $\overline{\text{STBY}}$ 管脚置为低电平，则转移到硬件待机模式。

20.8 子睡眠模式

20.8.1 向子睡眠模式的转移

子激活模式下，如果在 SBYCR 的 SSBY 位 = 0，LPWRCR 的 LSON 位 = 1，TCSR_1 的 PSS 位 = 1 的状态下，执行 SLEEP 指令，则 CPU 转移到子睡眠模式。

为子睡眠模式时，CPU 停止运行。另外，除 RTC 和 WDT 以外的外围功能停止运行。保持 CPU 的内部寄存器的内容，RAM 的数据，内部外围功能（除 A/D 转换器）及 I/O 端口的状态。

20.8.2 子睡眠模式的解除

通过中断（内部外围功能的中断、NMI 管脚、 $\overline{\text{IRQ0}} \sim \overline{\text{IRQ7}}$ ） $\overline{\text{RES}}$ 管脚、 $\overline{\text{MRES}}$ 管脚*、或 $\overline{\text{STBY}}$ 管脚解除子睡眠模式。

- 通过中断解除

如果发生中断，则解除子睡眠模式，开始中断异常处理。

此外， $\overline{\text{IRQ0}} \sim \overline{\text{IRQ7}}$ 中断相对应的允许位/管脚功能切换位被清 0 时，通过中断允许寄存器，禁止接受内部外围功能产生的中断时，或通过 CPU 屏蔽时，不能解除子睡眠模式。

- 通过 $\overline{\text{RES}}$ 管脚、 $\overline{\text{MRES}}$ 管脚*解除

有关通过 $\overline{\text{RES}}$ 管脚、 $\overline{\text{MRES}}$ 管脚*解除的情况，请参照 20.4.2 软件待机模式的解除。

- 通过 $\overline{\text{STBY}}$ 管脚解除

如果将 $\overline{\text{STBY}}$ 管脚置为低电平，则转移到硬件待机模式。

【注】 * 仅限于 H8S/2218 群使用。

20.9 子激活模式

20.9.1 向子激活模式的转移

为高速模式时，如果在 SBYCR 的 SSBY 位 = 1，LPWRCR 的 DTON 位 = 1，LSON 位 = 1，TCSR_1 的 PSS 位 = 1 的状态下，执行 SLEEP 指令，则 CPU 转移到子激活模式。另外，在监视模式下发生中断时，如果 LPWRCR 的 LSON 位 = 1，则转移到子激活模式。

为子激活模式时，CPU 通过子时钟低速运转，逐步执行程序。为子激活模式时，停止除 RTC 和 WDT 以外的外围功能的运行。

另外，在子激活模式下运行时，SCKCR 的 SCK2 ~ SCK0 的各位必须作为 0。

20.9.2 子激活模式的解除

通过 SLEEP 指令， $\overline{\text{RES}}$ 管脚， $\overline{\text{MRES}}$ 管脚 * 或 $\overline{\text{STBY}}$ 管脚，解除子激活模式。

- 通过 SLEEP 指令解除

如果在 SBYCR 的 SSBY 位 = 1，LPWRCR 的 DTON 位 = 0，TCSR_1 的 PSS 位 = 1 的状态下，执行 SLEEP 指令，则转移到监视模式。另外，如果在 SBYCR 的 SSBY 位 = 0，LPWRCR 的 LSON 位 = 1，TCSR_1 的 PSS 位 = 1 的状态下，执行 SLEEP 指令，则转移到子睡眠模式。如果在 SBYCR 的 SSBY 位 = 1，LPWRCR 的 DTON 位 = 1，LSON 位 = 0，TCSR_1 的 PSS 位 = 1 的状态下，执行 SLEEP 指令，则直接转移到高速模式（SCK0 ~ SCK2 全为 0）。

- 通过 $\overline{\text{RES}}$ 管脚， $\overline{\text{MRES}}$ 管脚 * 解除

有关通过 $\overline{\text{RES}}$ 管脚， $\overline{\text{MRES}}$ 管脚 * 解除的情况，请参照 20.4.2 软件待机模式的解除。

【注】 * 仅限于 H8S/2218 群使用。

- 通过 $\overline{\text{STBY}}$ 管脚解除

如果将 $\overline{\text{STBY}}$ 管脚置为低电平，则转移到硬件待机模式。

20.10 直接转移

CPU 执行程序的运行模式有：高速模式，中速模式，子激活模式三种。在高速模式与子激活模式之间，程序不间断地进行转移称为直接转移。通过将 LPWRCR 的 DTON 位置 1，执行 SLEEP 指令，可进行直接转移。转移后，开始直接转移中断异常处理。

20.10.1 从高速模式到子激活模式的直接转移

为高速模式时，如果在 SBYCR 的 SSBY 位 = 1，LPWRCR 的 LSON 位 = 1，DTON 位 = 1，TCSR_1 的 PSS 位 = 1 的状态下，执行 SLEEP 指令，则转移到子激活模式。

20.10.2 从子激活模式到高速模式的直接转移

为子激活模式时，如果在 SBYCR 的 SSBY 位 = 1，LPWRCR 的 LSON 位 = 0，DTON 位 = 1，TCSR_1 的 PSS 位 = 1 的状态下，执行 SLEEP 指令，则在经过 SBYCR 的 STS2 ~ STS0 设置的时间后，直接转移到高速模式。

20.11 ϕ 时钟输出控制

可通过 SCKCR 的 PSTOP 位，对应端口的 DDR，控制 ϕ 时钟的输出。如果 PSTOP 位置 1，则在总线周期结束时，停止 ϕ 时钟， ϕ 输出变为高电平。在 PSTOP 清 0 的状态下，允许 ϕ 时钟输出。另外，如果将对应端口的 DDR 清 0，则禁止 ϕ 时钟输出，变为输入端口。各处理状态下的 ϕ 管脚状态如表 20.4 所示。

表 20.4 各处理状态下的 ϕ 管脚状态

寄存器的设置值		高速模式，中速模式 子激活模式	睡眠模式 子睡眠模式	软件待机模式 监视模式，直接转移	硬件 待机模式
DDR	PSTOP				
0	X	高阻抗	高阻抗	高阻抗	高阻抗
1	0	ϕ 输出	ϕ 输出	High 固定	高阻抗
1	1	High 固定	High 固定	High 固定	高阻抗

20.12 使用时的注意事项

20.12.1 I/O 端口的状态

在软件待机模式及监视模式下，保持 I/O 端口的状态。另外，将 OPE 位置 1 时，地址总线，总线控制信号的输出被保持。因此，输出高电平时，输出电流的消耗电流不减少。

20.12.2 振荡稳定待机中的消耗电流

在振荡稳定待机中消耗电流增加。

20.12.3 闪存的模块停止

在内部 RAM 和外部存储器的程序运行过程中，执行闪存的模块停止设置。详情请参照 20.1.3 模块停止控制寄存器 A ~ C (MSTPCRA ~ MSTPCRC)。

20.12.4 DMAC 的模块停止

根据 DMAC 的运行状态，有 MSTPA7 位不置 1 的情况。DMAC 的模块停止模式的设置必须在 DMAC 未激活的状态下进行。详情请参照 7. DMA 控制器 (DMAC)。

20.12.5 内部外围模块的中断

- 模块停止模式

在模块停止模式下，不可运行此中断。因此，如果在请求中断的状态下，模块停止，则不可清除 CPU 的中断源或 DMAC 的启动源。

必须在设置模块停止模式之前禁止中断。

- 子激活模式/监视模式

子激活模式下停止运行的内部外围模块 (DMAC、TPU)，在子激活模式下，不可清除相关中断。因此，在请求中断的状态下，如果转换至子激活模式，则不可清除 CPU 的中断源。

请事先将各模块的中断设定为不可使用后，再执行 SLEEP 指令，并转换至子激活模式/监视模式。

20.12.6 子激活模式 / 监视模式转移和 DMAC 的模块停止

转移到子激活模式，监视模式时，将 DMAC 设置为模块停止（给 MSTPA7 位写入 1），并作为 1 读取 MSTPA7 位后，进行模式转移。从子激活模式转移到激活模式后，模块停止解除。

另外，在子激活模式中，发生 DMAC 启动源时，转移到激活模式后，模块停止解除时，启动 DMAC。

20.12.7 MSTPCR 的写入

MSTPCR 只能通过 CPU 写入。

21. 寄存器一览表

在寄存器一览表中，汇总了有关内部寄存器的地址、位组合及在各种工作模式下的寄存器状态信息。表述方法如下：

1. 寄存器地址一览表（按地址顺序）
 - 从分配的低地址寄存器开始按顺序记载。
 - 按模块名称分类。
 - 表示存取长度。

2. 寄存器位一览表
 - 按照**寄存器地址一览表（按地址顺序）**的顺序，记载寄存器的位组合。
 - 保留位在位名称部分用"－"表示。
 - 位名部分为位编号时，表示它的所有寄存器都被分配为计数器或保存数据。
 - 为16位或24位寄存器时，从MSB的位开始记载。

3. 各工作模式下的寄存器状态
 - 按**寄存器地址一览表（按地址顺序）**的顺序，记载寄存器的状态。
 - 表示在基本工作模式中的寄存器状态，为内部模块固有复位等情况时，请参照内部模块这一章。

21.1 寄存器地址一览表（按地址顺序）

数据总线宽度表示寄存器存取时的位数。

存取状态数表示指定标准时钟的状态数。

寄存器名称	简称	位数	地址	数据总线宽度	存取状态数	模块
USB 保留区	—	—	H'C00000 ~ H'C0007F	—	—	USB
USB 控制寄存器	UCTLR	8	H'C00080	8	3	
USB 测试寄存器 A	UTSTRA	8	H'C00081	8	3	
USB DMAC 传送请求寄存器	UDMAR	8	H'C00082	8	3	
USB 设备恢复寄存器	UDRR	8	H'C00083	8	3	
USB 触发寄存器 0	UTRG0	8	H'C00084	8	3	
USB FIFO 清除寄存器 0	UFCLR0	8	H'C00086	8	3	
USB 终点延迟寄存器 0	UESTL0	8	H'C00088	8	3	
USB 终点延迟寄存器 1	UESTL1	8	H'C00089	8	3	
USB 终点数据寄存器 0s	UEDR0s	8	H'C00090 ~ H'C00093	8	3	
USB 终点数据寄存器 0i	UEDR0i	8	H'C00094 ~ H'C00097	8	3	
USB 终点数据寄存器 0o	UEDR0o	8	H'C00098 ~ H'C0009B	8	3	
USB 终点数据寄存器 3	UEDR3	8	H'C0009C ~ H'C0009F	8	3	
USB 终点数据寄存器 1	UEDR1	8	H'C000A0 ~ H'C000A3	8	3	
USB 终点数据寄存器 2	UEDR2	8	H'C000A4 ~ H'C000A7	8	3	
USB 终点接收数据长度寄存器 0o	UESZ0o	8	H'C000BC	8	3	
USB 终点接收数据长度寄存器 2	UESZ2	8	H'C000BD	8	3	
USB 中断标志寄存器 0	UIFR0	8	H'C000C0	8	3	
USB 中断标志寄存器 1	UIFR1	8	H'C000C1	8	3	
USB 中断标志寄存器 3	UIFR3	8	H'C000C3	8	3	
USB 中断允许寄存器 0	UIER0	8	H'C000C4	8	3	
USB 中断允许寄存器 1	UIER1	8	H'C000C5	8	3	
USB 中断允许寄存器 3	UIER3	8	H'C000C7	8	3	
USB 中断选择寄存器 0	UISR0	8	H'C000C8	8	3	
USB 中断选择寄存器 1	UISR1	8	H'C000C9	—	—	
USB 中断选择寄存器 3	UISR3	8	H'C000CB	8	3	
USB 数据状态寄存器	UDSR	8	H'C000CC	8	3	
USB 配置值寄存器	UCVR	8	H'C000CF	8	3	
USB 测试寄存器 0	UTSTR0	8	H'C000F0	8	3	
USB 测试寄存器 1	UTSTR1	8	H'C000F1	8	3	
USB 测试寄存器 2	UTSTR2	8	H'C000F2	8	3	
USB 测试寄存器 B	UTSTRB	8	H'C000FB	8	3	
USB 测试寄存器 C	UTSTRC	8	H'C000FC	8	3	

寄存器名称	简称	位数	地址	数据总线宽度	存取状态数	模块
USB 测试寄存器 D	UTSTRD	8	H'C000FD	8	3	USB
USB 测试寄存器 E	UTSTRE	8	H'C000FE	8	3	
USB 测试寄存器 F	UTSTRF	8	H'C000FF	8	3	
USB 保留区	—	—	H'C00100 ~ H'DFFFFFF	—	—	
串行控制寄存器 X	SCRX	8	H'FDB4	8	2	FLASH
待机控制寄存器	SBYCR	8	H'FDE4	8	2	SYSTEM
系统控制寄存器	SYSCR	8	H'FDE5	8	2	
系统时钟控制寄存器	SCKCR	8	H'FDE6	8	2	
模式控制寄存器	MDCR	8	H'FDE7	8	2	
模块停止控制寄存器 A	MSTPCRA	8	H'FDE8	8	2	
模块停止控制寄存器 B	MSTPCRB	8	H'FDE9	8	2	
模块停止控制寄存器 C	MSTPCRC	8	H'FDEA	8	2	BSC
管脚功能控制寄存器	PFCR	8	H'FDEB	8	2	SYSTEM
低功率控制寄存器	LPWRCR	8	H'FDEC	8	2	BSC
时钟输出控制寄存器	OUTCR	8	H'FDEF	8	2	PORT
串行扩展模式寄存器 A_0	SEMRA_0	8	H'FDF8	8	2	SCI_0
串行扩展模式寄存器 B_0	SEMRB_0	8	H'FDF9	8	2	
IRQ 感知控制寄存器 H	ISCRH	8	H'FE12	8	2	INT
IRQ 感知控制寄存器 L	ISCR L	8	H'FE13	8	2	
IRQ 允许寄存器	IER	8	H'FE14	8	2	
IRQ 状态寄存器	ISR	8	H'FE15	8	2	
端口 1 数据方向寄存器	P1DDR	8	H'FE30	8	2	PORT
端口 3 数据方向寄存器	P3DDR	8	H'FE32	8	2	
端口 7 数据方向寄存器	P7DDR	8	H'FE36	8	2	
端口 A 数据方向寄存器	PADDR	8	H'FE39	8	2	
端口 B 数据方向寄存器	PBDDR	8	H'FE3A	8	2	
端口 C 数据方向寄存器	PCDDR	8	H'FE3B	8	2	
端口 D 数据方向寄存器	PDDDR	8	H'FE3C	8	2	
端口 E 数据方向寄存器	PEDDR	8	H'FE3D	8	2	
端口 F 数据方向寄存器	PFDDR	8	H'FE3E	8	2	
端口 G 数据方向寄存器	PGDDR	8	H'FE3F	8	2	
端口 A 上拉 MOS 控制寄存器	PAPCR	8	H'FE40	8	2	
端口 B 上拉 MOS 控制寄存器	PBPCR	8	H'FE41	8	2	
端口 C 上拉 MOS 控制寄存器	PCPCR	8	H'FE42	8	2	
端口 D 上拉 MOS 控制寄存器	PDPCR	8	H'FE43	8	2	
端口 E 上拉 MOS 控制寄存器	PEPCR	8	H'FE44	8	2	

寄存器名称	简称	位数	地址	数据总线宽度	存取状态数	模块
端口 3 漏极开路控制寄存器	P3ODR	8	H'FE46	8	2	PORT
端口 A 漏极开路控制寄存器	PAODR	8	H'FE47	8	2	
定时器开始寄存器	TSTR	8	H'FEB0	16	2	TPU
定时器同步寄存器	TSYR	8	H'FEB1	16	2	
中断优先级寄存器 A	IPRA	8	H'FEC0	8	2	INT
中断优先级寄存器 B	IPRB	8	H'FEC1	8	2	
中断优先级寄存器 C	IPRC	8	H'FEC2	8	2	
中断优先级寄存器 D	IPRD	8	H'FEC3	8	2	
中断优先级寄存器 E	IPRE	8	H'FEC4	8	2	
中断优先级寄存器 F	IPRF	8	H'FEC5	8	2	
中断优先级寄存器 G	IPRG	8	H'FEC6	8	2	
中断优先级寄存器 J	IPRJ	8	H'FEC9	8	2	
中断优先级寄存器 K	IPRK	8	H'FECA	8	2	
中断优先级寄存器 M	IPRM	8	H'FECC	8	2	
总线宽度控制寄存器	ABWCR	8	H'FED0	8	2	BSC
存取状态控制寄存器	ASTCR	8	H'FED1	8	2	
等待控制寄存器 H	WCRH	8	H'FED2	8	2	
等待控制寄存器 L	WCRL	8	H'FED3	8	2	
总线控制寄存器 H	BCRH	8	H'FED4	8	2	
总线控制寄存器 L	BCRL	8	H'FED5	8	2	
RAM 仿真寄存器	RAMER	8	H'FEDB	8	2	FLASH
存储器地址寄存器 0AH	MAR0AH	16	H'FEE0	16	2	DMAC
存储器地址寄存器 0AL	MAR0AL	16	H'FEE2	16	2	
I/O 地址寄存器 0A	IOAR0A	16	H'FEE4	16	2	
传送计数寄存器 0A	ETCR0A	16	H'FEE6	16	2	
存储器地址寄存器 0BH	MAR0BH	16	H'FEE8	16	2	
存储器地址寄存器 0BL	MAR0BL	16	H'FEEA	16	2	
I/O 地址寄存器 0B	IOAR0B	16	H'FEEC	16	2	
传送计数寄存器 0B	ETCR0B	16	H'FEEE	16	2	
存储器地址寄存器 1AH	MAR1AH	16	H'FEF0	16	2	
存储器地址寄存器 1AL	MAR1AL	16	H'FEF2	16	2	
I/O 地址寄存器 1A	IOAR1A	16	H'FEF4	16	2	
传送计数寄存器 1A	ETCR1A	16	H'FEF6	16	2	
存储器地址寄存器 1BH	MAR1BL	16	H'FEF8	16	2	
存储器地址寄存器 1BL	IOAR1B	16	H'FEFA	16	2	
I/O 地址寄存器 1B	ETCR1B	16	H'FEFC	16	2	

寄存器名称	简称	位数	地址	数据总线宽度	存取状态数	模块	
传送计数寄存器 1B	ETCR1B	16	H'FEFE	16	2	DMAC	
端口 1 数据寄存器	P1DR	8	H'FF00	8	2	PORT	
端口 3 数据寄存器	P3DR	8	H'FF02	8	2		
端口 7 数据寄存器	P7DR	8	H'FF06	8	2		
端口 A 数据寄存器	PADR	8	H'FF09	8	2		
端口 B 数据寄存器	PBDR	8	H'FF0A	8	2		
端口 C 数据寄存器	PCDR	8	H'FF0B	8	2		
端口 D 数据寄存器	PDDR	8	H'FF0C	8	2		
端口 E 数据寄存器	PEDR	8	H'FF0D	8	2		
端口 F 数据寄存器	PFDR	8	H'FF0E	8	2		
端口 G 数据寄存器	PGDR	8	H'FF0F	8	2		
定时器控制寄存器_0	TCR_0	8	H'FF10	16	2		TPU_0
定时器模式寄存器_0	TMDR_0	8	H'FF11	16	2		
定时器 I/O 控制寄存器 H_0	TIORH_0	8	H'FF12	16	2		
定时器 I/O 控制寄存器 L_0	TIORL_0	8	H'FF13	16	2		
定时器中断允许寄存器_0	TIER_0	8	H'FF14	16	2		
定时器状态寄存器_0	TSR_0	8	H'FF15	16	2		
定时器计数器_0	TCNT_0	16	H'FF16	16	2		
定时器通用寄存器 A_0	TGRA_0	16	H'FF18	16	2		
定时器通用寄存器 B_0	TGRB_0	16	H'FF1A	16	2		
定时器通用寄存器 C_0	TGRC_0	16	H'FF1C	16	2		
定时器通用寄存器 D_0	TGRD_0	16	H'FF1E	16	2		
定时器控制寄存器_1	TCR_1	8	H'FF20	16	2	TPU_1	
定时器模式寄存器_1	TMDR_1	8	H'FF21	16	2		
定时器 I/O 控制寄存器_1	TIOR_1	8	H'FF22	16	2		
定时器中断允许寄存器_1	TIER_1	8	H'FF24	16	2		
定时器状态寄存器_1	TSR_1	8	H'FF25	16	2		
定时器计数器_1	TCNT_1	16	H'FF26	16	2		
定时器通用寄存器 A_1	TGRA_1	16	H'FF28	16	2		
定时器通用寄存器 B_1	TGRB_1	16	H'FF2A	16	2		
定时器控制寄存器_2	TCR_2	8	H'FF30	16	2		TPU_2
定时器模式寄存器_2	TMDR_2	8	H'FF31	16	2		
定时器 I/O 控制寄存器 2	TIOR_2	8	H'FF32	16	2		
定时器中断允许寄存器 2	TIER_2	8	H'FF34	16	2		
定时器状态寄存器_2	TSR_2	8	H'FF35	16	2		
定时器计数器_2	TCNT_2	16	H'FF36	16	2		

寄存器名称	简称	位数	地址	数据总线宽度	存取状态数	模块
定时器通用寄存器 A_2	TGRA_2	16	H'FF38	16	2	TPU_2
定时器通用寄存器 B_2	TGRB_2	16	H'FF3A	16	2	
扩展模块停止寄存器	EXMDLSTP	8	H'FF40	8	2	SYSTEM
秒数据寄存器 / 自由运行计数器	RSECDR	8	H'FF48	8	2	RTC
分数据寄存器	RMINDR	8	H'FF49	8	2	
时数据寄存器	RHRDR	8	H'FF4A	8	2	
星期数据寄存器	RWKDR	8	H'FF4B	8	2	
RTC 控制寄存器 1	RTCCR1	8	H'FF4C	8	2	
RTC 控制寄存器 2	RTCCR2	8	H'FF4D	8	2	
时钟源选择寄存器	RTCCSR	8	H'FF4F	8	2	
DMA 控制寄存器 0A	DMACR0A	8	H'FF62	16	2	DMAC
DMA 控制寄存器 0B	DMACR0B	8	H'FF63	16	2	
DMA 控制寄存器 1A	DMACR1A	8	H'FF64	16	2	
定时器控制 / 状态寄存器	TCSR	8	H'FF74	16	2	WDT
定时器计数器	TCNT	8	H'FF74 (写入时)	16	2	
定时器计数器	TCNT	8	H'FF75 (读取时)	16	2	
复位控制 / 状态寄存器	RSTCSR	8	H'FF76 (写入时)	16	2	
复位控制 / 状态寄存器	RSTCSR	8	H'FF77 (读取时)	16	2	
串行模式寄存器_0	SMR_0	8	H'FF78	8	2	SCI_0
比特率寄存器_0	BRR_0	8	H'FF79	8	2	
串行控制寄存器_0	SCR_0	8	H'FF7A	8	2	
传输数据寄存器_0	TDR_0	8	H'FF7B	8	2	
串行状态寄存器_0	SSR_0	8	H'FF7C	8	2	
接收数据寄存器_0	RDR_0	8	H'FF7D	8	2	
智能卡模式寄存器_0	SCMR_0	8	H'FF7E	8	2	
串行模式寄存器_2	SMR_2	8	H'FF88	8	2	SCI_2
比特率寄存器_2	BRR_2	8	H'FF89	8	2	
串行控制寄存器_2	SCR_2	8	H'FF8A	8	2	
传输数据寄存器_2	TDR_2	8	H'FF8B	8	2	
串行状态寄存器_2	SSR_2	8	H'FF8C	8	2	
接收数据寄存器_2	RDR_2	8	H'FF8D	8	2	
智能卡模式寄存器_2	SCMR_2	8	H'FF8E	8	2	
A/D 数据寄存器 AH	ADDRAH	8	H'FF90	8	2	A/D
A/D 数据寄存器 AL	ADDRAL	8	H'FF91	8	2	
A/D 数据寄存器 BH	ADDRBH	8	H'FF92	8	2	
A/D 数据寄存器 BL	ADDRBL	8	H'FF93	8	2	

寄存器名称	简称	位数	地址	数据总线宽度	存取状态数	模块
A/D 数据寄存器 CH	ADDRCH	8	H'FF94	8	2	A/D
A/D 数据寄存器 CL	ADDRCL	8	H'FF95	8	2	
A/D 数据寄存器 DH	ADDRDH	8	H'FF96	8	2	
A/D 数据寄存器 DL	ADDRDL	8	H'FF97	8	2	
A/D 控制 / 状态寄存器	ADCSR	8	H'FF98	8	2	
A/D 控制寄存器	ADCR	8	H'FF99	8	2	
定时器控制 / 状态寄存器	TCSR_1	8	H'FFA2	16	2	SYSTEM
闪存控制寄存器 1	FLMCR1	8	H'FFA8	8	2	FLASH
闪存控制寄存器 2	FLMCR2	8	H'FFA9	8	2	
擦除块指定寄存器 1	EBR1	8	H'FFAA	8	2	
擦除块指定寄存器 2	EBR2	8	H'FFAB	8	2	
端口 1 寄存器	PORT1	8	H'FFB0	8	2	PORT
端口 3 寄存器	PORT3	8	H'FFB2	8	2	
端口 4 寄存器	PORT4	8	H'FFB3	8	2	
端口 7 寄存器	PORT7	8	H'FFB6	8	2	
端口 9 寄存器	PORT9	8	H'FFB8	8	2	
端口 A 寄存器	PORTA	8	H'FFB9	8	2	
端口 B 寄存器	PORTB	8	H'FFBA	8	2	
端口 C 寄存器	PORTC	8	H'FFBB	8	2	
端口 D 寄存器	PORTD	8	H'FFBC	8	2	
端口 E 寄存器	PORTE	8	H'FFBD	8	2	
端口 F 寄存器	PORTF	8	H'FFBE	8	2	
端口 G 寄存器	PORTG	8	H'FFBF	8	2	

21.2 寄存器位一览表

内部外围模块的寄存器位名如下所示：

16 位、32 位寄存器每 8 位分 2 段或 4 段表示。

寄存器简称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	模块
UCTLR	—	USPNDE	UCKS3	UCKS2	UCKS1	UCKS0	UIFRST	UDCRST	USB
UTSTRA	—	—	—	—	—	—	—	—	
UDMAR	—	—	—	—	EP2T1	EP2T0	EP1T1	EP1T0	
UDRR	—	—	—	—	—	—	RWUPs	DVR	
UTRG0	—	—	EP2RDFN	EP1PKTE	EP3PKTE	EP0oRDFN	EP0iPKTE	EP0sRDFN	
UFCLR0	—	—	EP2CLR	EP1CLR	EP3CLR	EP0oCLR	EP0iCLR	—	
UESTL0	—	—	EP2STL	EP1STL	EP3STL	—	—	EP0STL	
UESTL1	SCME	—	—	—	—	—	—	—	
UEDR0s	D7	D6	D5	D4	D3	D2	D1	D0	
UEDR0i	D7	D6	D5	D4	D3	D2	D1	D0	
UEDR0o	D7	D6	D5	D4	D3	D2	D1	D0	
UEDR3	D7	D6	D5	D4	D3	D2	D1	D0	
UEDR1	D7	D6	D5	D4	D3	D2	D1	D0	
UEDR2	D7	D6	D5	D4	D3	D2	D1	D0	
UESZ0o	—	D6	D5	D4	D3	D2	D1	D0	
UESZ2	—	D6	D5	D4	D3	D2	D1	D0	
UIFR0	BRST	—	EP3TR	EP3TS	EP0oTS	EP0iTR	EP0iTS	SetupTS	
UIFR1	—	—	—	—	EP1ALL EMPTYs	EP2 READY	EP1TR	EP1 EMPTY	
UIFR3	CK48 READY	SOF	SETC	—	SPRSs	SPRSi	VBUSs	VBUSi	
UIER0	BRSTE	—	EP3TRE	EP3TSE	EP0oTSE	EP0iTRE	EP0iTSE	SetupTSE	
UIER1	—	—	—	—	—	EP2 READYE	EP1TRE	EP1 EMPTYE	
UIER3	CK48 READYE	SOFE	SETCE	—	—	SPRSiE	—	VBUSiE	
UISR0	BRSTS	—	EP3TRS	EP3TSS	EP0oTSS	EP0iTRS	EP0iTSS	SetupTSS	
UISR1	—	—	—	—	—	EP2 READYS	EP1TRS	EP1 EMPTYYS	
UISR3	CK48 READYS	SOFS	SETCS	—	—	—	—	VBUSiS	
UDSR	—	—	—	—	—	EP1DE	EP3DE	EP0iDE	
UCVR	—	—	CNFV0	—	—	—	—	—	
UTSTR0	PTSTE	—	—	—	SUSPEN	\overline{OE}	FSE0	VPO	

寄存器简称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	模块
UTSTR1	VBUS	UBPM	—	—	—	RCV	VP	VM	USB
UTSTR2	—	—	—	—	—	—	—	—	
UTSTRB	—	—	—	—	—	—	—	—	
UTSTRC	—	—	—	—	—	—	—	—	
UTSTRD	—	—	—	—	—	—	—	—	
UTSTRE	—	—	—	—	—	—	—	—	
UTSTRF	—	—	—	—	—	—	—	—	
SCRX	—	—	—	—	FLSHE	—	—	—	FLASH
SBYCR	SSBY	STS2	STS1	STS0	OPE	—	—	—	SYSTEM
SYSCR	—	—	INTM1	INTM0	NMIEG	MRESE	—	RAME	
SCKCR	PSTOP	—	—	—	—	SCK2	SCK1	SCK0	
MDCR	—	—	—	—	FWE	MDS2	MDS1	MDS0	
MSTPCRA	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0	
MSTPCRB	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0	
MSTPCRC	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0	
PFCR	—	—	—	—	AE3	AE2	AE1	AE0	BSC
LPWRCR	DTON	LSON	NESEL	SUBSTP	RFCUT	—	STC1	SCT0	SYSTEM
OUTCR	—	—	—	—	—	PF7OUT2	PF7OUT1	PF7OUT0	PORT
SEMRA_0	SSE	TCS2	TCS1	TCS0	ABCS	ACS2	ACS1	ACS0	SCI_0
SEMRB_0	ACS3	—	—	—	TIOCA2E	TIOCA1E	TIOCC0E	TIOCA0E	
ISCRH	IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA	INT
ISURL	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA	
IER	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E	
ISR	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F	
P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	PORT
P3DDR	—	P36DDR	—	—	—	P32DDR	P31DDR	P30DDR	
P7DDR	P77DDR	P76DDR	P75DDR	P74DDR	—	—	P71DDR	P70DDR	
PADDR	—	—	—	—	PA3DDR	PA2DDR	PA1DDR	PA0DDR	
PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR	
PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR	
PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR	
PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR	
PFDDR	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR	
PGDDR	—	—	—	PG4DDR	PG3DDR	PG2DDR	PG1DDR	PG0DDR	
PAPCR	—	—	—	—	PA3PCR	PA2PCR	PA1PCR	PA0PCR	
PBPCR	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR	

寄存器简称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	模块
PCPCR	PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR	PORT
PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR	
PEPCR	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR	
P3ODR	—	P36ODR	—	—	—	P32ODR	P31ODR	P30ODR	
PAODR	—	—	—	—	PA3ODR	PA2ODR	PA1ODR	PA0ODR	
TSTR	—	—	—	—	—	CST2	CST1	CST0	TPU
TSYR	—	—	—	—	—	SYNC2	SYNC1	SYNC0	
IPRA	—	IPRA6	IPRA5	IPRA4	—	IPRA2	IPRA1	IPRA0	INT
IPRB	—	IPRB6	IPRB5	IPRB4	—	IPRB2	IPRB1	IPRB0	
IPRC	—	IPRC6	IPRC5	IPRC4	—	—	—	—	
IPRD	—	IPRD6	IPRD5	IPRD4	—	—	—	—	
IPRE	—	—	—	—	—	IPRE2	IPRE1	IPRE0	
IPRF	—	IPRF6	IPRF5	IPRF4	—	IPRF2	IPRF1	IPRF0	
IPRG	—	IPRG6	IPRG5	IPRG4	—	—	—	—	
IPRJ	—	IPRJ6	IPRJ5	IPRJ4	—	IPRJ2	IPRJ1	IPRJ0	
IPRK	—	—	—	—	—	IPRK2	IPRK1	IPRK0	
IPRM	—	IPRM6	IPRM5	IPRM4	—	—	—	—	
ABWCR	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0	BSC
ASTCR	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0	
WCRH	W71	W70	W61	W60	W51	W50	W41	W40	
WCRL	W31	W30	W21	W20	W11	W10	W01	W00	
BCRH	ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	RMTS2	RMTS1	RMTS0	
BCRL	BRLE	—	—	—	—	—	—	WAITE	
RAMER	—	—	—	—	RAMS	—	RAM1	RAM0	FLASH
MAR0A	—	—	—	—	—	—	—	—	DMAC
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16	
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
IOAR0A	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
ETCR0A	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MAR0B	—	—	—	—	—	—	—	—	
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16	
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

寄存器简称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	模块	
IOAR0B	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	DMAC	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0		
ETCR0B	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0		
MAR1A	—	—	—	—	—	—	—	—		
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16		
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0		
IOAR1A	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0		
ETCR1A	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0		
MAR1B	—	—	—	—	—	—	—	—		
	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16		
	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0		
IOAR1B	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0		
ETCR1B	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0		
P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR	PORT	
P3DR	—	P36DR	—	—	—	P32DR	P31DR	P30DR		
P7DR	P77DR	P76DR	P75DR	P74DR	—	—	P71DR	P70DR		
PADR	—	—	—	—	PA3DR	PA2DR	PA1DR	PA0DR		
PBDR	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR		
PCDR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR		
PDDR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR		
PEDR	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR		
PFDR	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR		
PGDR	—	—	—	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR		
TCR_0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0		TPU_0
TMDR_0	—	—	BFB	BFA	MD3	MD2	MD1	MD0		
TIORH_0	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0		
TIORL_0	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0		
TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA		
TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA		

寄存器简称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	模块
TCNT_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	TPU_0
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRC_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRD_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_1
TMDR_1	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_2
TMDR_2	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
EXMDLSTP	—	—	—	—	—	—	RTCSTOP	USBSTOP1	SYSTEM
RSECDR	BSY	SC12	SC11	SC10	SC03	SC02	SC01	SC00	RTC
RMINDR	BSY	MN12	MN11	MN10	MN03	MN02	MN01	MN00	
RHRDR	BSY	—	HR11	HR10	HR03	HR02	HR01	HR00	

寄存器简称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	模块
RWKDR	BSY	—	—	—	—	WK2	WK1	KWK0	RTC
RTCCR1	RUN	12/24	PM	RST	—	—	—	—	
RTCCR2	—	—	FOIE	WKIE	DYIE	HRIE	MNIE	SEIE	
RTCCSR	—	RCS6	RCS5	—	RCS3	RCS2	RCS1	RCS0	
DMACR0A*1	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	DMAC
DMACR0A*2	DTSZ	SAID	SAIDE	BLKDIR	BLKE	—	—	—	
DMACR0B*1	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	
DMACR0B*2	—	DAID	DAIDE	—	DTF3	DTF2	DTF1	DTF0	
DMACR1A*1	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	
DMACR1A*2	DTSZ	SAID	SAIDE	BLKDIR	BLKE	—	—	—	
DMACR1B*1	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	
DMACR1B*2	—	DAID	DAIDE	—	DTF3	DTF2	DTF1	DTF0	
DMABCR*1	FAE1	FAE0	—	—	DTA1B	DTA1A	DTA0B	DTA0A	
	DTE1B	DTE1A	DTE0B	DTE0A	DTIE1B	DTIE1A	DTIE0B	DTIE0A	
DMABCR*2	FAE1	FAE0	—	—	DTA1	—	DTA0	—	
	DTME1	DTE1	DTME0	DTE0	DTIE1B	DTIE1A	DTIE0B	DTIE0A	
TCSR	OVF	WT/IT	TME	—	—	CKS2	CKS1	CKS0	WDT
TCNT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
RSTCSR	WOVF	RSTE	RSTS	—	—	—	—	—	
SMR_0	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0	SCI_0
SMR_0*3	GM	BLK	PE	O/E	BCP1	BCP0	CKS1	CKS0	
BRR_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCR_0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SSR_0	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
SSR_0*3	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT	
RDR_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_0	—	—	—	—	SDIR	SINV	—	SMIF	
SMR_2	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0	SCI_2
SMR_2*3	GM	BLK	PE	O/E	BCP1	BCP0	CKS1	CKS0	
BRR_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCR_2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SSR_2	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
SSR_2*3	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT	
RDR_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

寄存器简称	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	模块
SCMR_2	—	—	—	—	SDIR	SINV	—	SMIF	SCI_2
ADDRAH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
ADDRAL	AD1	AD0	—	—	—	—	—	—	
ADDRBH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRCL	AD1	AD0	—	—	—	—	—	—	
ADDRDH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
ADDRDL	AD1	AD0	—	—	—	—	—	—	
ADCSR	ADF	ADIE	ADST	SCAN	—	CH2	CH1	CH0	
ADCR	TRGS1	TRGS0	—	—	CKS1	CKS0	—	—	
TCSR_1	—	—	—	PSS	—	—	—	—	SYSTEM
FLMCR1	FWE	SWE1	ESU1	PSU1	EV1	PV1	E1	P1	FLASH
FLMCR2	FLER	—	—	—	—	—	—	—	
EBR1	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0	
EBR2	—	—	—	—	—	—	EB9	EB8	
PORT1	P17	P16	P15	P14	P13	P12	P11	P10	PORT
PORT3	—	P36	—	—	—	P32	P31	P30	
PORT4	—	—	—	—	P43	P42	P41	P40	
PORT7	P77	P76	P75	P74	—	—	P71	P70	
PORT9	P97	P96	—	—	—	—	—	—	
PORTA	—	—	—	—	PA3	PA2	PA1	PA0	
PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	
PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	
PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	
PORTE	PE7	PE6	PE5	PE4	PR3	PE2	PE1	PE0	
PORTF	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	
PORTG	—	—	—	PG4	PG3	PG2	PG1	PG0	

【注】 *1 短地址模式

*2 满地址模式

*3 智能卡接口

21.3 各工作模式下的寄存器状态

寄存器简称	加电复位	手动复位	高速	中速	睡眠	模块停止	监视	子激活	子睡眠	软件待机	硬件待机	模块
UCTLR	初始化*	—	—	—	—	—	—	—	—	—	初始化	USB
UTSTRA	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UDMAR	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UDRR	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UTRG0	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UFCLR0	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UESTL0	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UESTL1	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UEDR0s	—	—	—	—	—	—	—	—	—	—	—	
UEDR0i	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UEDR0o	—	—	—	—	—	—	—	—	—	—	—	
UEDR3	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UEDR1	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UEDR2	—	—	—	—	—	—	—	—	—	—	—	
UESZ0o	—	—	—	—	—	—	—	—	—	—	—	
UESZ2	—	—	—	—	—	—	—	—	—	—	—	
UIFR0	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UIFR1	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UIFR3	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UIER0	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UIER1	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UIER3	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UISR0	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UISR1	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UISR3	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UDSR	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UCVR	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UTSTR0	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UTSTR1	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UTSTR2	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UTSTRB	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UTSTRC	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UTSTRD	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UTSTRE	初始化*	—	—	—	—	—	—	—	—	—	初始化	
UTSTRF	初始化*	—	—	—	—	—	—	—	—	—	初始化	
SCRX	初始化	初始化	—	—	—	—	—	—	—	—	初始化	FLASH

寄存器简称	加电复位	手动复位	高速	中速	睡眠	模块停止	监视	子激活	子睡眠	软件待机	硬件待机	模块
SBYCR	初始化	初始化	—	—	—	—	—	—	—	—	初始化	SYSTEM
SYSCR	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
SCKCR	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
MDCR	初始化	—	—	—	—	—	—	—	—	—	初始化	
MSTPCRA	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
MSTPCRB	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
MSTPCRC	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
PFCR	初始化	—	—	—	—	—	—	—	—	—	初始化	PSC
LPWRCR	初始化	—	—	—	—	—	—	—	—	—	初始化	SYSTEM
OUTCR	初始化	初始化	—	—	—	—	—	—	—	—	初始化	PORT
SEMRA_0	初始化	初始化	—	—	—	—	—	—	—	—	初始化	SCI_0
SEMRB_0	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
ISCRH	初始化	初始化	—	—	—	—	—	—	—	—	初始化	INT
ISCR_L	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
IER	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
ISR	初始化	初始化	—	—	—	—	—	—	—	—	初始化	PORT
P1DDR	初始化	—	—	—	—	—	—	—	—	—	初始化	
P3DDR	初始化	—	—	—	—	—	—	—	—	—	初始化	
P7DDR	初始化	—	—	—	—	—	—	—	—	—	初始化	
PADDR	初始化	—	—	—	—	—	—	—	—	—	初始化	
PBDDR	初始化	—	—	—	—	—	—	—	—	—	初始化	
PCDDR	初始化	—	—	—	—	—	—	—	—	—	初始化	
PDDDR	初始化	—	—	—	—	—	—	—	—	—	初始化	
PEDDR	初始化	—	—	—	—	—	—	—	—	—	初始化	
PFDDR	初始化	—	—	—	—	—	—	—	—	—	初始化	
PGDDR	初始化	—	—	—	—	—	—	—	—	—	初始化	
PAPCR	初始化	—	—	—	—	—	—	—	—	—	初始化	
PBPCR	初始化	—	—	—	—	—	—	—	—	—	初始化	
PCPCR	初始化	—	—	—	—	—	—	—	—	—	初始化	
PDPCR	初始化	—	—	—	—	—	—	—	—	—	初始化	
PEPCR	初始化	—	—	—	—	—	—	—	—	—	初始化	
P3ODR	初始化	—	—	—	—	—	—	—	—	—	初始化	
PAODR	初始化	—	—	—	—	—	—	—	—	—	初始化	
TSTR	初始化	初始化	—	—	—	—	—	—	—	—	初始化	TPU
TSYR	初始化	初始化	—	—	—	—	—	—	—	—	初始化	

寄存器简称	加电复位	手动复位	高速	中速	睡眠	模块停止	监视	子激活	子睡眠	软件待机	硬件待机	模块
IPRA	初始化	初始化	—	—	—	—	—	—	—	—	初始化	INT
IPRB	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
IPRC	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
IPRD	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
IPRE	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
IPRF	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
IPRG	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
IPRJ	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
IPRK	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
IPRM	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
ABWCR	初始化	—	—	—	—	—	—	—	—	—	初始化	PSC
ASTCR	初始化	—	—	—	—	—	—	—	—	—	初始化	
WCRH	初始化	—	—	—	—	—	—	—	—	—	初始化	
WCRL	初始化	—	—	—	—	—	—	—	—	—	初始化	
BCRH	初始化	—	—	—	—	—	—	—	—	—	初始化	
BCRL	初始化	—	—	—	—	—	—	—	—	—	初始化	
RAMER	初始化	—	—	—	—	—	—	—	—	—	初始化	FLASH
MAR0A	—	—	—	—	—	—	—	—	—	—	—	DMAC
IOAR0A	—	—	—	—	—	—	—	—	—	—	—	
ETCR0A	—	—	—	—	—	—	—	—	—	—	—	
MAR0B	—	—	—	—	—	—	—	—	—	—	—	
IOAR0B	—	—	—	—	—	—	—	—	—	—	—	
ETCR0B	—	—	—	—	—	—	—	—	—	—	—	
MAR1A	—	—	—	—	—	—	—	—	—	—	—	
IOAR1A	—	—	—	—	—	—	—	—	—	—	—	
ETCR1A	—	—	—	—	—	—	—	—	—	—	—	
MAR1B	—	—	—	—	—	—	—	—	—	—	—	
IOAR1B	—	—	—	—	—	—	—	—	—	—	—	
ETCR1B	—	—	—	—	—	—	—	—	—	—	—	

寄存器简称	加电复位	手动复位	高速	中速	睡眠	模块停止	监视	子激活	子睡眠	软件待机	硬件待机	模块
P1DR	初始化	—	—	—	—	—	—	—	—	—	初始化	PORT
P3DR	初始化	—	—	—	—	—	—	—	—	—	初始化	
P7DR	初始化	—	—	—	—	—	—	—	—	—	初始化	
PADR	初始化	—	—	—	—	—	—	—	—	—	初始化	
PBDR	初始化	—	—	—	—	—	—	—	—	—	初始化	
PCDR	初始化	—	—	—	—	—	—	—	—	—	初始化	
PDDR	初始化	—	—	—	—	—	—	—	—	—	初始化	
PEDR	初始化	—	—	—	—	—	—	—	—	—	初始化	
PFDR	初始化	—	—	—	—	—	—	—	—	—	初始化	
PGDR	初始化	—	—	—	—	—	—	—	—	—	初始化	
TCR_0	初始化	初始化	—	—	—	—	—	—	—	—	初始化	TPU_0
TMDR_0	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TIORH_0	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TIORL_0	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TIER_0	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TSR_0	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TCNT_0	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TGRA_0	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TGRB_0	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TGRC_0	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TGRD_0	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TCR_1	初始化	初始化	—	—	—	—	—	—	—	—	初始化	TPU_1
TMDR_1	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TIOR_1	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TIER_1	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TSR_1	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TCNT_1	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TGRA_1	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TGRB_1	初始化	初始化	—	—	—	—	—	—	—	—	初始化	

寄存器简称	加电复位	手动复位	高速	中速	睡眠	模块停止	监视	子激活	子睡眠	软件待机	硬件待机	模块
TCR_2	初始化	初始化	—	—	—	—	—	—	—	—	初始化	TPU_2
TMDR_2	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TIOR_2	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TIER_2	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TSR_2	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TCNT_2	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TGRA_2	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TGRB_2	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
EXMDLSTP	初始化	—	—	—	—	—	—	—	—	—	初始化	SYSTEM
RSECDR	—	—	—	—	—	—	—	—	—	—	初始化	RTC
RMINDR	—	—	—	—	—	—	—	—	—	—	初始化	
RHRDR	—	—	—	—	—	—	—	—	—	—	初始化	
RWKDR	—	—	—	—	—	—	—	—	—	—	初始化	
RTCCR1	—	—	—	—	—	—	—	—	—	—	初始化	
RTCCR2	—	—	—	—	—	—	—	—	—	—	初始化	
RTCCSR	初始化	—	—	—	—	—	—	—	—	—	初始化	
DMACR0A	初始化	初始化	—	—	—	—	—	—	—	—	初始化	DMAC
DMACR0B	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
DMACR1A	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
DMACR1B	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
DMABCR	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TCSR	初始化	初始化	—	—	—	—	—	—	—	—	初始化	WDT
TCNT	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
RSTCSR	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
SMR_0	初始化	初始化	—	—	—	—	—	—	—	—	初始化	SCI_0
BRR_0	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
SCR_0	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TDR_0	初始化	初始化	—	—	—	—	初始化	初始化	初始化	初始化	初始化	
SSR_0	初始化	初始化	—	—	—	—	初始化	初始化	初始化	初始化	初始化	
RDR_0	初始化	初始化	—	—	—	—	初始化	初始化	初始化	初始化	初始化	
SCMR_0	初始化	初始化	—	—	—	—	—	—	—	—	初始化	

寄存器简称	加电复位	手动复位	高速	中速	睡眠	模块停止	监视	子激活	子睡眠	软件待机	硬件待机	模块
SMR_2	初始化	初始化	—	—	—	—	—	—	—	—	初始化	SCI_1
BRR_2	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
SCR_2	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
TDR_2	初始化	初始化	—	—	—	—	初始化	初始化	初始化	初始化	初始化	
SSR_2	初始化	初始化	—	—	—	—	初始化	初始化	初始化	初始化	初始化	
RDR_2	初始化	初始化	—	—	—	—	初始化	初始化	初始化	初始化	初始化	
SCMR_2	初始化	初始化	—	—	—	—	—	—	—	—	初始化	
ADDRAH	初始化	初始化	—	—	—	—	初始化	初始化	初始化	初始化	初始化	A/D
ADDRAL	初始化	初始化	—	—	—	—	初始化	初始化	初始化	初始化	初始化	
ADDRBH	初始化	初始化	—	—	—	—	初始化	初始化	初始化	初始化	初始化	
ADDRBL	初始化	初始化	—	—	—	—	初始化	初始化	初始化	初始化	初始化	
ADDRCH	初始化	初始化	—	—	—	—	初始化	初始化	初始化	初始化	初始化	
ADDRCL	初始化	初始化	—	—	—	—	初始化	初始化	初始化	初始化	初始化	
ADDRDH	初始化	初始化	—	—	—	—	初始化	初始化	初始化	初始化	初始化	
ADDRDL	初始化	初始化	—	—	—	—	初始化	初始化	初始化	初始化	初始化	
ADCSR	初始化	初始化	—	—	—	—	初始化	初始化	初始化	初始化	初始化	
ADCR	初始化	初始化	—	—	—	—	初始化	初始化	初始化	初始化	初始化	
TCSR_1	初始化	初始化	—	—	—	—	—	—	—	初始化	初始化	SYSTEM
FLMCR1	初始化	—	—	—	—	—	—	—	—	初始化	初始化	FLASH
FLMCR2	初始化	—	—	—	—	—	—	—	—	初始化	初始化	
EBR1	初始化	—	—	—	—	—	—	—	—	初始化	初始化	
EBR2	初始化	—	—	—	—	—	—	—	—	初始化	初始化	
PORT1	—	—	—	—	—	—	—	—	—	—	—	PORT
PORT3	—	—	—	—	—	—	—	—	—	—	—	
PORT4	—	—	—	—	—	—	—	—	—	—	—	
PORT7	—	—	—	—	—	—	—	—	—	—	—	
PORT9	—	—	—	—	—	—	—	—	—	—	—	
PORTA	—	—	—	—	—	—	—	—	—	—	—	
PORTB	—	—	—	—	—	—	—	—	—	—	—	
PORTC	—	—	—	—	—	—	—	—	—	—	—	
PORTD	—	—	—	—	—	—	—	—	—	—	—	
PORTE	—	—	—	—	—	—	—	—	—	—	—	
PORTF	—	—	—	—	—	—	—	—	—	—	—	
PORTG	—	—	—	—	—	—	—	—	—	—	—	

【注】— 没有初始化。

* USB 的寄存器在通过 WDT 加电复位时，不被初始化。

22. 电特性

22.1 绝对最大额定值

绝对最大额定值如表 22.1 所示。

表 22.1 绝对最大额定值

项目	符号	额定值	单位
电源电压	V_{CC} , $PLL V_{CC}$, DrV_{CC}	-0.3 ~ +4.3	V
输入电压	V_{in}	-0.3 ~ $V_{CC}+0.3$	V
参考电源电压	V_{ref}	-0.3 ~ $V_{CC}+0.3$	V
模拟输入电压	V_{AN}	-0.3 ~ $V_{CC}+0.3$	V
运行温度	T_{opr}	普通规格产品: -20 ~ +75*	°C
		宽温度范围规格产品: -40~+85*	
保存温度	T_{stg}	-55 ~ +125	°C

【使用注意事项】

超过绝对最大额定值使用 LSI 时，有时会对 LSI 造成永久性破坏。

【注】 * 闪存写入 / 擦除时的运行温度范围为 $T_a = -20 \sim 75^\circ\text{C}$ 。

22.2 电源电压与运行频率范围

电源电压与运行范围（阴影部分）如图 22.1 所示。

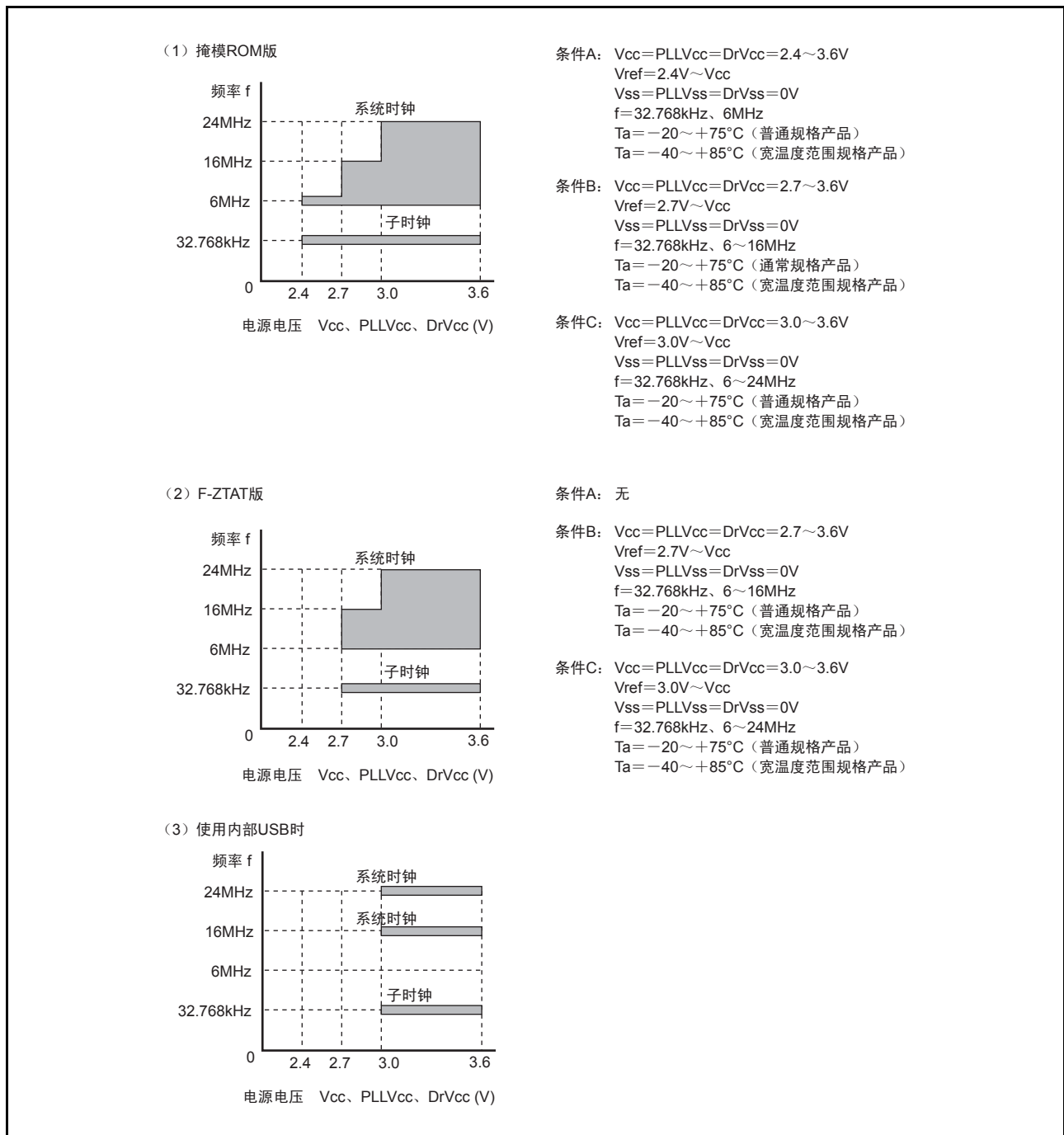


图 22.1 电源电压与运行范围

22.3 DC 特性

DC 特性如表 22.2 所示。输出容许电流如表 22.3 所示。

表 22.2 DC 特性

条件 A: $V_{CC} = PLLV_{CC} = DrV_{CC} = 2.4 \sim 3.6V$, $V_{ref} = 2.4V \sim V_{CC}$, $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$, $f = 32.768kHz, 6MHz$, $T_a = -20 \sim +75^{\circ}C$ (普通规格产品), $T_a = -40 \sim +85^{\circ}C$ (宽温度范围规格产品)

条件 B: $V_{CC} = PLLV_{CC} = DrV_{CC} = 2.7 \sim 3.6V$, $V_{ref} = 2.7V \sim V_{CC}$, $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$, $f = 32.768kHz, 6 \sim 16MHz$, $T_a = -20 \sim +75^{\circ}C$ (普通规格产品), $T_a = -40 \sim +85^{\circ}C$ (宽温度范围规格产品)

条件 C: $V_{CC} = PLLV_{CC} = DV_{CC} = 3.0 \sim 3.6V$, $V_{ref} = 3.0V \sim V_{CC}$, $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$, $f = 32.768kHz, 6 \sim 24MHz$, $T_a = -20 \sim +75^{\circ}C$ (普通规格产品), $T_a = -40 \sim +85^{\circ}C$ (宽温度范围规格产品)

项目	符号	min.	typ.	max.	单位	测定条件
施密特触发器输入电压	$\overline{IRQ0} \sim \overline{IRQ4}, \overline{IRQ7}$	$V_{CC} \times 0.2$	—	—	V	
	VT^+	—	—	$V_{CC} \times 0.8$	V	
	$VT^+ - VT^-$	$V_{CC} \times 0.05$	—	—	V	
输入高电平电压	$\overline{RES}, \overline{STBY}, \overline{NMI}, MD2 \sim MD0, \overline{TRST}, TCK, TMS, TDI, EMLE, VBUS, \overline{UBPM}, FWE^*4$	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	EXTAL 端口 1、3、4、7、9、A ~ G	$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$	V	
输入低电平电压	$\overline{RES}, \overline{STBY}, MD2 \sim MD0, \overline{TRST}, TCK, TMS, TDI, EMLE, VBUS, \overline{UBPM}, FWE^*4$	-0.3	—	$V_{CC} \times 0.1$	V	
	EXTAL、NMI、 端口 1、3、4、7、9、A ~ G	-0.3	—	$V_{CC} \times 0.2$	V	
输出高电平电压	全部输出管脚	V_{OH}	$V_{CC} - 0.5$	—	V	$I_{OH} = -200\mu A$
		$V_{CC} - 1.0$	—	—	V	$I_{OH} = -1mA$
输出低电平电压	全部输出管脚	V_{OL}	—	0.4	V	$I_{OL} = 0.8mA$
输入漏泄电流	$\overline{RES}, VBUS, \overline{UBPM}, \overline{STBY}, \overline{NMI}, EMLE, MD2 \sim MD0, FWE^*4$ 端口 4、9	$ I_{in} $	—	1.0	μA	$V_{in} = 0.5 \sim V_{CC} - 0.5V$
三态漏泄电流 (off state)	端口 1、3、7、 端口 A ~ G	$ I_{TSI} $	—	1.0	μA	$V_{in} = 0.5 \sim V_{CC} - 0.5V$
输入上拉 MOS 电流	端口 A ~ E $\overline{TDI}, TCK, TMS, \overline{TRST}$	$-I_P$	10	—	μA	$V_{in} = 0V$

项目	符号	min.	typ.	max.	单位	测定条件		
输入容量	RES, NMI	C _{in}	—	—	30	pF	V _{in} = 0V f = 1MHz T _a = 25°C	
	RES, NMI 以外的全部输入管脚		—	—	15	pF		
消耗电流*1	正常运行 (USB 停止) 时	I _{CC} *2	—	22	35	mA	f = 16MHz V _{CC} = 3.3V V _{CC} = 3.6V	
			—	31	50	mA		
	正常运行 (USB 运行) 时		—	30	45	mA	f = 16MHz (PLL3 倍频)	
			—	41	60	mA	f = 24MHz (PLL2 倍频)	
	睡眠时		—	16	30	mA	f = 16MHz (USB, PLL 停止时)	
			—	22	45	mA	f = 24MHz (USB, PLL 停止时)	
	除闪存外的全部 模块停止时		—	16	—	mA	f = 16MHz (参考值)	
			—	24	—	mA	f = 24MHz (参考值)	
	子激活模式时		—	45	180	μA	使用 32.768KHz 晶体振荡器 时, V _{CC} =3.3V EMLE = 0	
			—	30*5	—	—		
	子睡眠模式时		—	35	100	μA		
			—	20*5	—	—		
	监视模式时		—	5	40	μA		
	待机时*3		—	1.0	10	μA	T _a ≤ 50°C	32.768kHz 与 RTC 停止 EMLE = 0
—		—	50	μA	50°C < T _a			
参考 电源电流	A/D 转换中	A _I CC	—	1.3	2.5	mA	V _{ref} = 3.3V	
	A/D 转换待机时		—	0.01	5.0	μA		
RAM 待机电压	V _{RAM}	2.0	—	—	v			

【注】 即使不使用 A/D 转换器时，也请不要将 V_{ref} 管脚设为开路状态，将 V_{ref} 管脚连接至 V_{CC} 上。

*1 消耗电流值为 V_{IH} min = V_{CC} - 0.2V, V_{IL} max = 0.2V 条件下，将所有输出管脚置于无负载状态，并将内部上拉 MOS 置于断电状态时的值。

*2 依赖于 V_{CC} 与 f 的 I_{CC} 公式如下：（参考）

$$I_{CCmax.} = 5 \text{ (mA)} + 0.52 \text{ (mA/(MHz} \times \text{V))} \times V_{CC} \times f \text{ (正常运行, USB 停止时)}$$

$$I_{CCmax.} = 9 \text{ (mA)} + 0.60 \text{ (mA/(MHz} \times \text{V))} \times V_{CC} \times f \text{ (正常运行, USB 运行时)}$$

$$I_{CCmax.} = 1 \text{ (mA)} + 0.51 \text{ (mA/(MHz} \times \text{V))} \times V_{CC} \times f \text{ (睡眠时)}$$

*3 为 V_{RAM} ≤ V_{CC} < 2.7V 时、V_{IH} min = V_{CC} × 0.9, V_{IL} max = 0.3V 时的值。

*4 FWE 管脚仅适用 F-ZTAT 版。

*5 为内部 RAM 程序执行过程中，将闪存设定为模块停止状态时的参考值。仅适用 F-ZTAT 版。

表 22.3 输出容许电流

条件 A: $V_{CC} = PLLV_{CC} = DrV_{CC} = 2.4 \sim 3.6V$, $V_{ref} = 2.4V \sim V_{CC}$, $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$, $f = 32.768kHz, 6MHz$, $T_a = -20 \sim +75^{\circ}C$ (普通规格产品), $T_a = -40 \sim +85^{\circ}C$ (宽温度范围规格产品)

条件 B: $V_{CC} = PLLV_{CC} = DrV_{CC} = 2.7 \sim 3.6V$, $V_{ref} = 2.7V \sim V_{CC}$, $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$, $f = 32.768kHz, 6 \sim 16MHz$, $T_a = -20 \sim +75^{\circ}C$ (普通规格产品), $T_a = -40 \sim +85^{\circ}C$ (宽温度范围规格产品)

条件 C: $V_{CC} = PLLV_{CC} = DrV_{CC} = 3.0 \sim 3.6V$, $V_{ref} = 3.0V \sim V_{CC}$, $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$, $f = 32.768kHz, 6 \sim 24MHz$, $T_a = -20 \sim +75^{\circ}C$ (普通规格产品), $T_a = -40 \sim +85^{\circ}C$ (宽温度范围规格产品)

项目		符号	min.	typ.	max.	单位
输出低电平容许电流 (每个管脚)	全部输出管脚	I_{OL}	—	—	1.0	mA
输出低电平容许电流 (总和)	全部输出管脚的总和	ΣI_{OL}	—	—	60	mA
输出高电平容许电流 (每个管脚)	全部输出管脚	$-I_{OH}$	—	—	1.0	mA
输出高电平容许电流 (总和)	全部输出管脚的总和	$\Sigma -I_{OH}$	—	—	30	mA

【注】 为确保 LSI 的可靠性, 输出电流值不可超过表 22.3 所示的值。

22.4 AC 特性

AC 测定条件如图 22.2 所示。

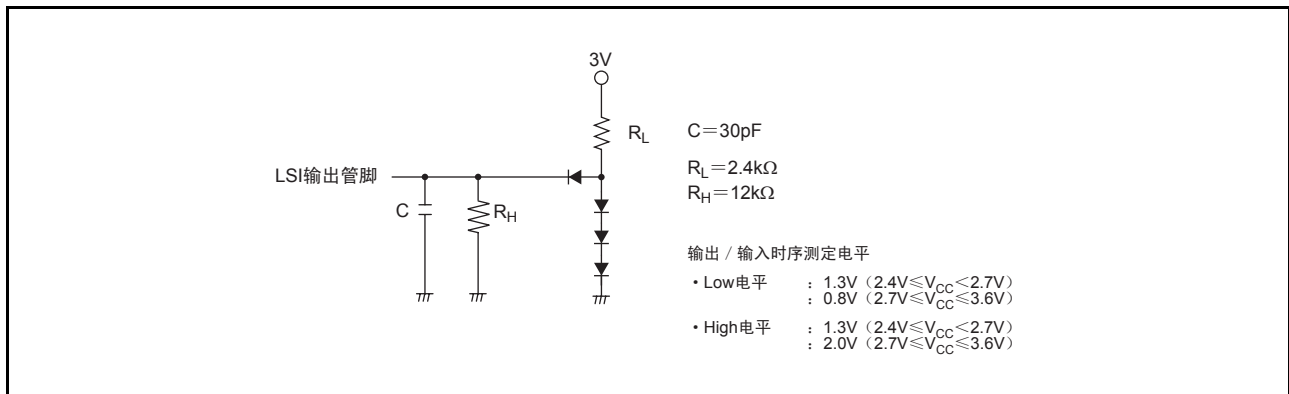


图 22.2 输出负载电路

22.4.1 时钟时序

时钟时序如表 22.4 所示。

表 22.4 时钟时序

条件 A: $V_{CC} = PLLV_{CC} = DrV_{CC} = 2.4 \sim 3.6V$, $V_{ref} = 2.4V \sim V_{CC}$, $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$, $f = 32.768kHz, 6MHz$, $T_a = -20 \sim +75^{\circ}C$ (普通规格产品), $T_a = -40 \sim +85^{\circ}C$ (宽温度范围规格产品)

条件 B: $V_{CC} = PLLV_{CC} = DrV_{CC} = 2.7 \sim 3.6V$, $V_{ref} = 2.7V \sim V_{CC}$, $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$, $f = 32.768kHz, 6 \sim 16MHz$, $T_a = -20 \sim +75^{\circ}C$ (普通规格产品), $T_a = -40 \sim +85^{\circ}C$ (宽温度范围规格产品)

条件 C: $V_{CC} = PLLV_{CC} = DrV_{CC} = 3.0 \sim 3.6V$, $V_{ref} = 3.0V \sim V_{CC}$, $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$, $f = 32.768kHz, 6 \sim 24MHz$, $T_a = -20 \sim +75^{\circ}C$ (普通规格产品), $T_a = -40 \sim +85^{\circ}C$ (宽温度范围规格产品)

项目	符号	条件 A		条件 B		条件 C		单位	测定条件
		min.	max.	min.	max.	min.	max.		
时钟周期时间	t_{cyc}	166.6		62.5	166.6	41.6	166.6	ns	图 22.3
时钟高电平脉冲宽度	t_{CH}	50	—	20	—	13	—	ns	
时钟低电平脉冲宽度	t_{CL}	50	—	20	—	13	—	ns	
时钟上升时间	t_{Cr}	—	25	—	10	—	7	ns	
时钟下降时间	t_{Cf}	—	25	—	10	—	7	ns	
复位振荡稳定时间 (晶体)	t_{OSC1}	40	—	20	—	20	—	ms	图 22.4
软件待机振荡稳定时间 (晶体)	t_{OSC2}	16	—	8	—	8	—	ms	图 22.4, 图 19.2 $C_{L1} = C_{L2} = 10 \sim 22pF$
		16	—	8	—	4	—	ms	图 20.4, 图 19.2 $C_{L1} = C_{L2} = 10 \sim 15pF$
外部时钟输出稳定延迟时间	t_{DEXT}	1000	—	500	—	500	—	μs	图 22.4
子时钟振荡稳定时间	t_{OSC3}	—	4	—	2	—	2	s	
子时钟振荡器振荡频率	f_{SUB}	32.768		32.768		32.768		kHz	
子时钟 (ϕ_{SUB}) 周期时间	t_{SUB}	30.5		30.5		30.5		μs	

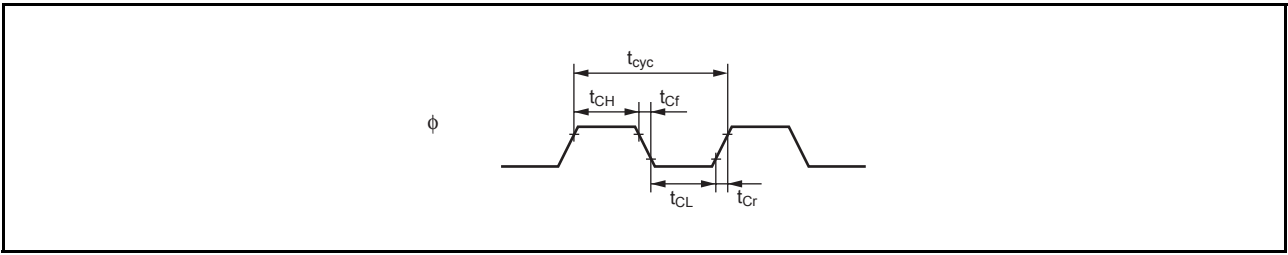


图 22.3 系统时钟时序

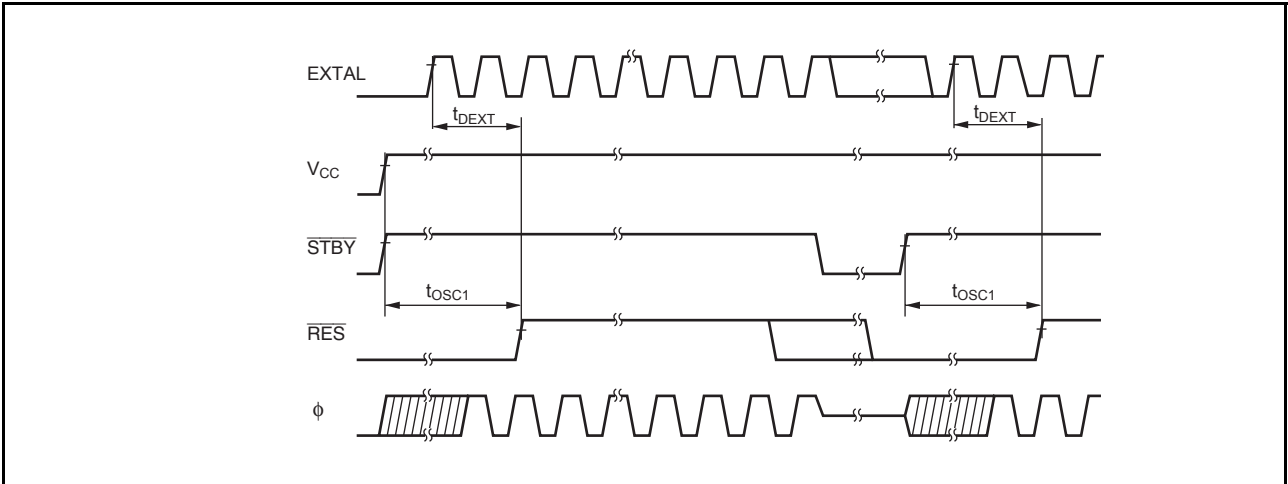


图 22.4 振荡稳定时间时序

22.4.2 控制信号时序

控制信号时序如表 22.5 所示。

表 22.5 控制信号时序

条件 A: $V_{CC} = PLLV_{CC} = DrV_{CC} = 2.4 \sim 3.6V$, $V_{ref} = 2.4V \sim V_{CC}$, $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$, $f = 32.768kHz, 6MHz$, $T_a = -20 \sim +75^{\circ}C$ (普通规格产品), $T_a = -40 \sim +85^{\circ}C$ (宽温度范围规格产品)

条件 B: $V_{CC} = PLLV_{CC} = DrV_{CC} = 2.7 \sim 3.6V$, $V_{ref} = 2.7V \sim V_{CC}$, $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$, $f = 32.768kHz, 6 \sim 16MHz$, $T_a = -20 \sim +75^{\circ}C$ (普通规格产品), $T_a = -40 \sim +85^{\circ}C$ (宽温度范围规格产品)

条件 C: $V_{CC} = PLLV_{CC} = DrV_{CC} = 3.0 \sim 3.6V$, $V_{ref} = 3.0V \sim V_{CC}$, $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$, $f = 32.768kHz, 6 \sim 24MHz$, $T_a = -20 \sim +75^{\circ}C$ (普通规格产品), $T_a = -40 \sim +85^{\circ}C$ (宽温度范围规格产品)

项目	符号	条件 A		条件 B, C		单位	测定条件
		min.	max.	min.	max.		
\overline{RES} 设置时间	t_{RESS}	350	—	250	—	ns	图 22.5
\overline{RES} 脉冲宽度	t_{RESW}	20	—	20	—	t_{cyc}	
\overline{MRES} 设置时间	t_{MRESS}	350	—	250	—	ns	
\overline{MRES} 脉冲宽度	t_{MRESW}	20	—	20	—	t_{cyc}	
NMI 设置时间	t_{NMIS}	350	—	250	—	ns	图 22.6
NMI 保持时间	t_{NMIH}	10	—	10	—	ns	
NMI 脉冲宽度 (从软件待机模式返回时)	t_{NMIW}	300	—	200	—	ns	
\overline{IRQ} 设置时间	t_{IRQS}	350	—	250	—	ns	
\overline{IRQ} 保持时间	t_{IRQH}	10	—	10	—	ns	
\overline{IRQ} 脉冲时间 (从软件待机模式返回时)	t_{IRQW}	300	—	200	—	ns	

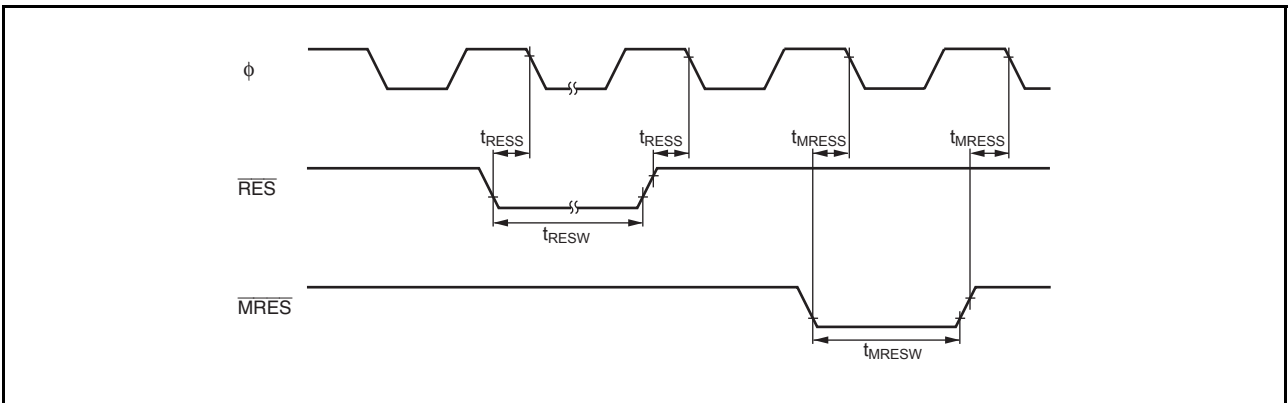


图 22.5 复位输入时序

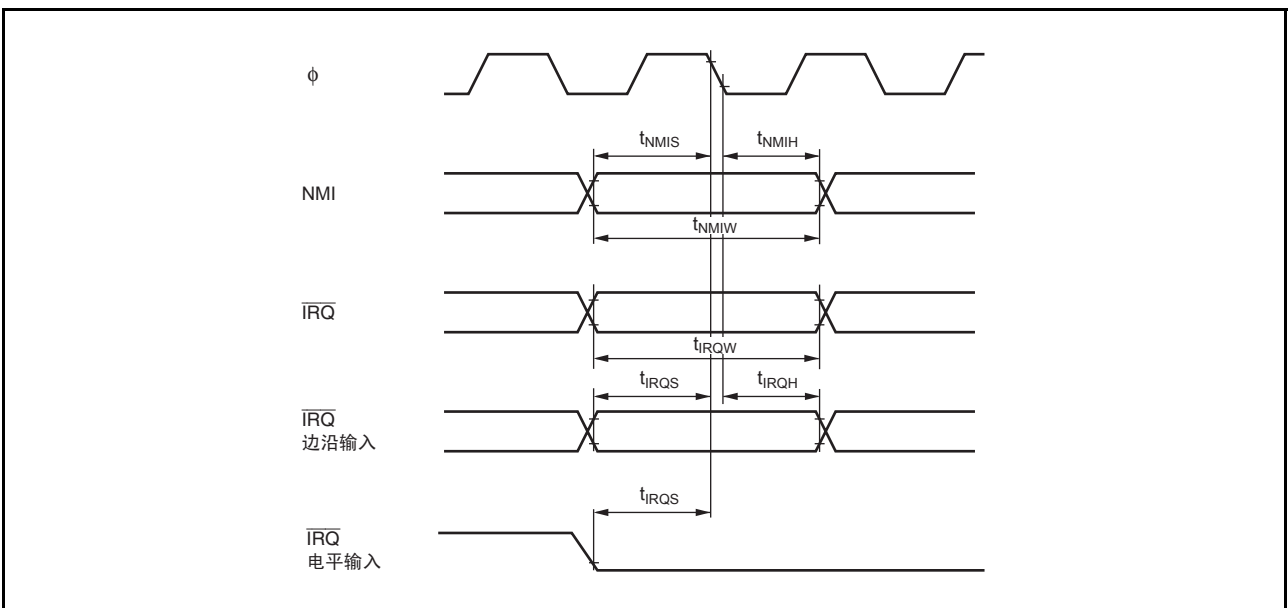


图 22.6 中断输入时序

22.4.3 总线时序

总线时序如表 22.6 所示。

表 22.6 总线时序

条件 A: $V_{CC} = PLLV_{CC} = DrV_{CC} = 2.4 \sim 3.6V$, $V_{ref} = 2.4V \sim V_{CC}$, $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$, $f = 32.768kHz, 6MHz$, $T_a = -20 \sim +75^{\circ}C$ (普通规格产品), $T_a = -40 \sim +85^{\circ}C$ (宽温度范围规格产品)

条件 B: $V_{CC} = PLLV_{CC} = DrV_{CC} = 2.7 \sim 3.6V$, $V_{ref} = 2.7V \sim V_{CC}$, $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$, $f = 32.768kHz, 6 \sim 16MHz$, $T_a = -20 \sim +75^{\circ}C$ (普通规格产品), $T_a = -40 \sim +85^{\circ}C$ (宽温度范围规格产品)

条件 C: $V_{CC} = PLLV_{CC} = DrV_{CC} = 3.0 \sim 3.6V$, $V_{ref} = 3.0V \sim V_{CC}$, $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$, $f = 32.768kHz, 6 \sim 24MHz$, $T_a = -20 \sim +75^{\circ}C$ (普通规格产品), $T_a = -40 \sim +85^{\circ}C$ (宽温度范围规格产品)

项目	符号	条件 A		条件 B		条件 C		单位	测定条件
		min.	max.	min.	max.	min.	max.		
地址延迟时间	t_{AD}	—	90	—	50	—	30	ns	图 22.7、 图 22.8、 图 22.10
地址准备时间	t_{AS}	$0.5 \times t_{cyc}$ - 60	—	$0.5 \times t_{cyc}$ -30	—	$0.5 \times t_{cyc}$ - 20	—	ns	
地址保持时间	t_{AH}	$0.5 \times t_{cyc}$ - 30	—	$0.5 \times t_{cyc}$ -15	—	$0.5 \times t_{cyc}$ - 8	—	ns	
CS 延迟时间	t_{CSD}	—	90	—	50	—	30	ns	图 22.7、 图 22.8
AS 延迟时间	t_{ASD}	—	90	—	50	—	25	ns	图 22.7、 图 22.8、 图 22.10
RD 延迟时间 1	t_{RSD1}	—	90	—	50	—	25	ns	图 22.7、 图 22.8
RD 延迟时间 2	t_{RSD2}	—	90	—	50	—	25	ns	图 22.7、 图 22.8、 图 22.10
读取数据设置时间	t_{RDS}	50	—	30	—	20	—	ns	
读取数据保持时间	t_{RDH}	0	—	0	—	0	—	ns	
读取数据存取时间 2	t_{ACC2}	—	$1.5 \times t_{cyc}$ - 90	—	$1.5 \times t_{cyc}$ - 65	—	$1.5 \times t_{cyc}$ - 35	ns	图 22.7
读取数据存取时间 3	t_{ACC3}	—	$2.0 \times t_{cyc}$ - 90	—	$2.0 \times t_{cyc}$ - 65	—	$2.0 \times t_{cyc}$ - 40	ns	图 22.7、 图 22.10
读取数据存取时间 4	t_{ACC4}	—	$2.5 \times t_{cyc}$ - 90	—	$2.5 \times t_{cyc}$ - 65	—	$2.5 \times t_{cyc}$ - 35	ns	
读取数据存取时间 5	t_{ACC5}	—	$3.0 \times t_{cyc}$ - 90	—	$3.0 \times t_{cyc}$ -65	—	$3.0 \times t_{cyc}$ - 40	ns	
WR 延迟时间 1	t_{WRD1}	—	90	—	50	—	20	ns	图 22.7、 图 22.8
WR 延迟时间 2	t_{WRD2}	—	90	—	50	—	25	ns	
WR 脉冲宽度 1	t_{WSW1}	$1.0 \times t_{cyc}$ - 60	—	$1.0 \times t_{cyc}$ -30	—	$1.0 \times t_{cyc}$ - 20	—	ns	
WR 脉冲宽度 2	t_{WSW2}	$1.5 \times t_{cyc}$ - 60	—	$1.5 \times t_{cyc}$ -30	—	$1.5 \times t_{cyc}$ - 20	—	ns	图 22.8

项目	符号	条件 A		条件 B		条件 C		单位	测定条件
		min.	max.	min.	max.	min.	max.		
写入数据延迟时间	t_{WDD}	—	100	—	50	—	30	ns	图 22.7、 图 22.8
写入数据设置时间	t_{WDS}	$0.5 \times t_{cyc}$ - 80	—	$0.5 \times t_{cyc}$ - 30	—	$0.5 \times t_{cyc}$ - 20	—	ns	图 22.8
写入数据保持时间	t_{WDH}	$0.5 \times t_{cyc}$ - 60	—	$0.5 \times t_{cyc}$ - 15	—	$0.5 \times t_{cyc}$ - 10	—	ns	图 22.7、 图 22.8
WAIT 设置时间	t_{WTS}	90	—	50	—	25	—	ns	图 22.9
WAIT 保持时间	t_{WTH}	10	—	10	—	5	—	ns	
BREQ 设置时间	t_{BRQS}	90	—	50	—	25	—	ns	图 22.11
BACK 延迟时间	t_{BACD}	—	90	—	50	—	35	ns	
总线浮动时间	t_{BZD}	—	160	—	80	—	50	ns	

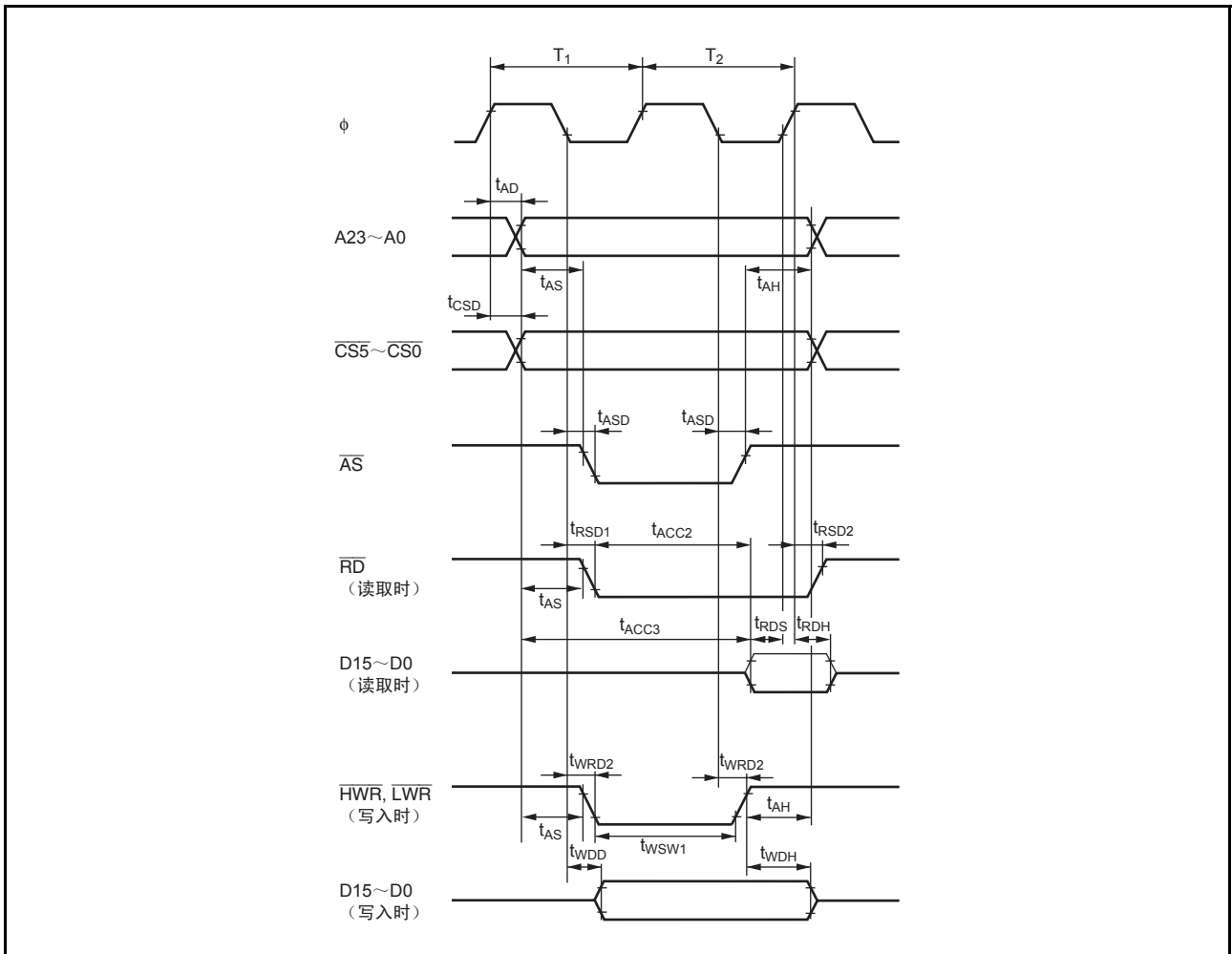


图 22.7 基本总线时序 / 2 态存取

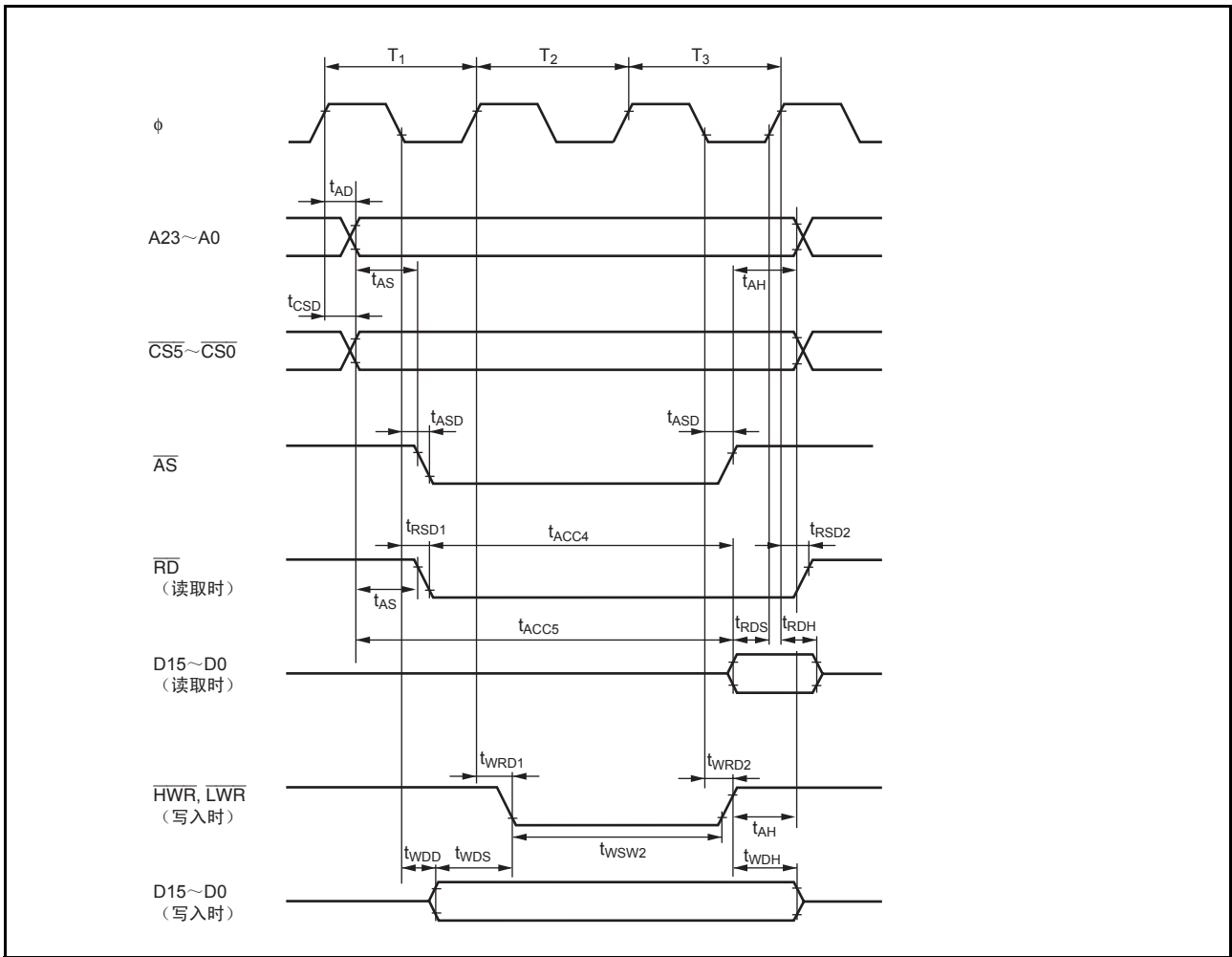


图 22.8 基本总线时序 /3 态存取

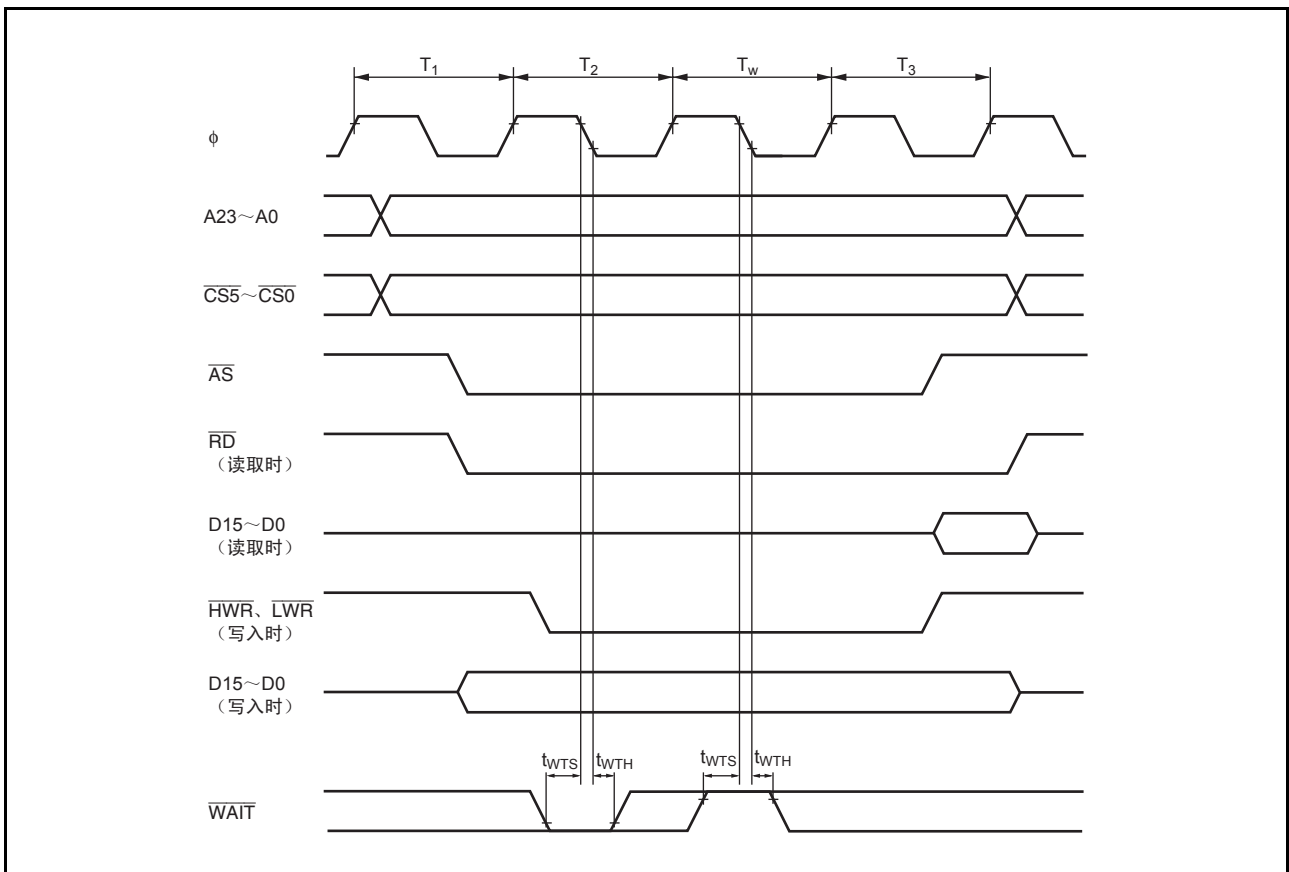


图 22.9 基本总线时序 /3 态存取 1 等待

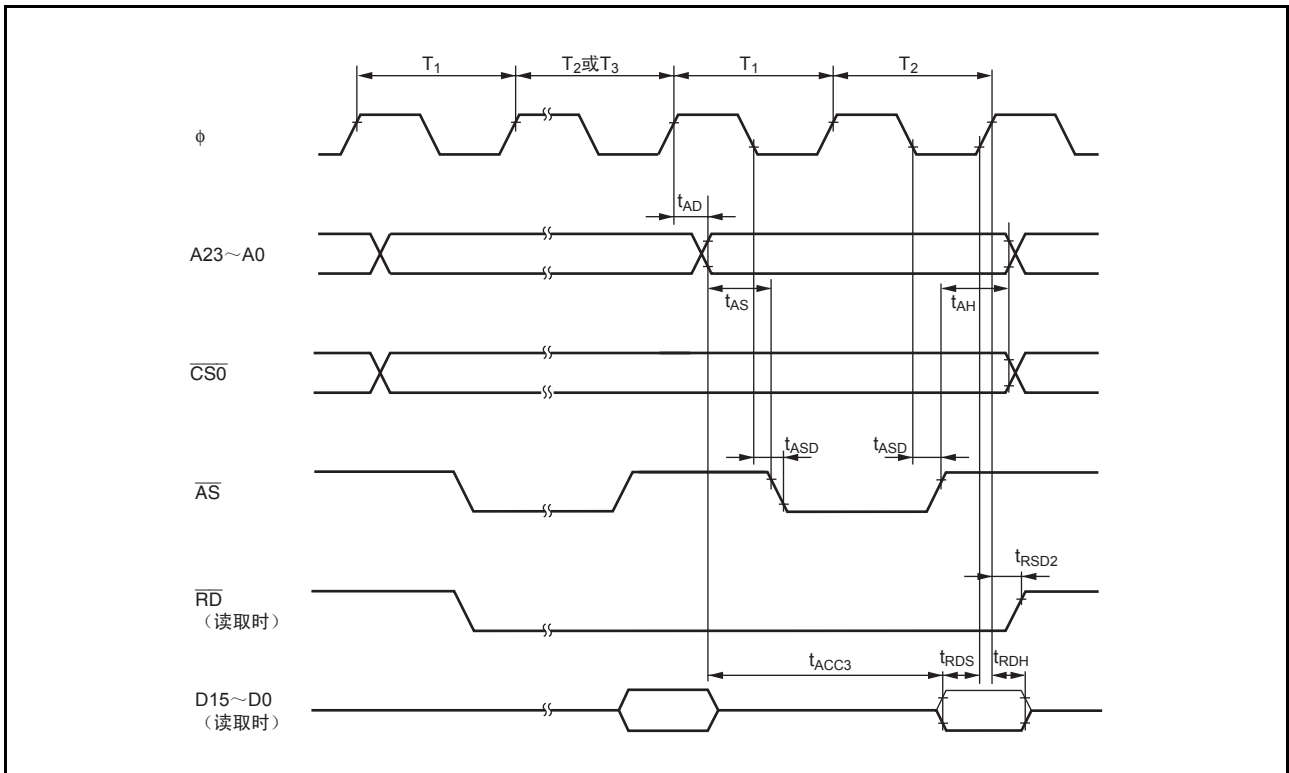


图 22.10 突发式 ROM 存取时序 /2 态存取

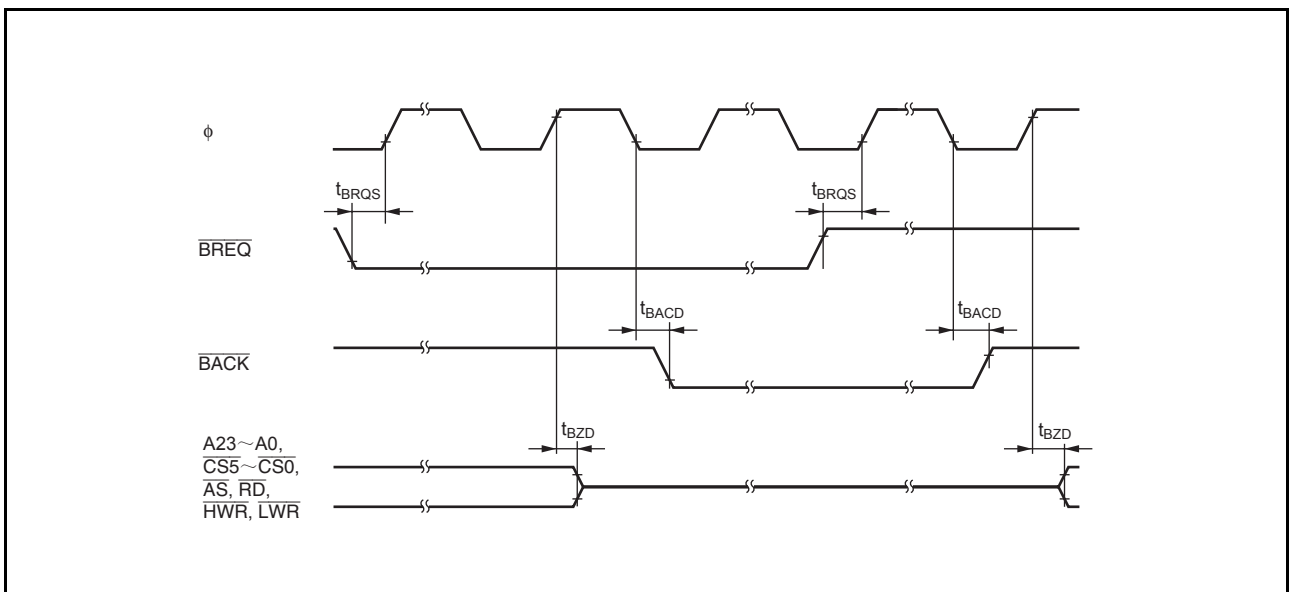


图 22.11 外部总线仲裁时序

22.4.4 内部外围模块时序

内部外围时序如表 22.7 所示。

表 22.7 内部外围时序

条件 A: $V_{CC} = PLLV_{CC} = DrV_{CC} = 2.4 \sim 3.6V$, $V_{ref} = 2.4V \sim V_{CC}$, $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$, $f = 32.768kHz, 6MHz$, $T_a = -20 \sim +75^{\circ}C$ (普通规格产品), $T_a = -40 \sim +85^{\circ}C$ (宽温度范围规格产品)

条件 B: $V_{CC} = PLLV_{CC} = DrV_{CC} = 2.7 \sim 3.6V$, $V_{ref} = 2.7V \sim V_{CC}$, $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$, $f = 32.768kHz, 6 \sim 16MHz$, $T_a = -20 \sim +75^{\circ}C$ (普通规格产品), $T_a = -40 \sim +85^{\circ}C$ (宽温度范围规格产品)

条件 C: $V_{CC} = PLLV_{CC} = DrV_{CC} = 3.0 \sim 3.6V$, $V_{ref} = 3.0V \sim V_{CC}$, $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$, $f = 32.768kHz, 6 \sim 24MHz$, $T_a = -20 \sim +75^{\circ}C$ (普通规格产品), $T_a = -40 \sim +85^{\circ}C$ (宽温度范围规格产品)

I/O 端口	项目	符号	条件 A		条件 B		条件 C		单位	测定条件
			min.	max.	min.	max.	min.	max.		
I/O 端口	输出数据延迟时间	t_{PWD}	—	150	—	60	—	40	ns	图 22.12
	输入数据设置时间	t_{PRS}	80	—	50	—	30	—		
	输入数据保持时间	t_{PRH}	50	—	50	—	30	—		
TPU	定时器输出延迟时间	t_{TOCD}	—	150	—	60	—	40	ns	图 22.13
	定时器输入设置时间	t_{TICS}	60	—	40	—	30	—		
	定时器时钟输入设置时间	t_{TCKS}	1.5	—	40	—	30	—	ns	图 22.14
	定时器时钟脉冲宽度	单边沿指定	t_{TCKWH}	60	—	1.5	—	1.5		
双边沿指定		t_{TCKWL}	2.5	—	2.5	—	2.5	—		
SCI	输入时钟周期	异步	t_{Scyc}	4	—	4	—	4	tcyc	图 22.15
		时钟同步		6	—	6	—	6		
	输入时钟脉冲宽度	t_{SCKW}	0.4	0.6	0.4	0.6	0.4	0.6	tcyc	
	输入时钟上升时间	t_{SCKr}	—	1.5	—	1.5	—	1.5		
	输入时钟下降时间	t_{SCKf}	—	1.5	—	1.5	—	1.5		
	发送数据延迟时间	t_{TXD}	—	150	—	60	—	40	ns	图 22.16
	接收数据设置时间 (时钟同步)	t_{RXS}	150	—	60	—	40	—		
接收数据保持时间 (时钟同步)	t_{RXH}	150	—	60	—	40	—			
A/D 转换器	触发输入设置时间	t_{TRGS}	60	—	40	—	30	—	ns	图 22.17
边界扫描	TCK 周期时间	t_{Tcyc}	166.6	—	62.5	—	41.6	—	ns	图 22.18
	TCK 高电平脉冲宽度	t_{TCKH}	0.4	0.6	0.4	0.6	0.4	0.6		
	TCK 低电平脉冲宽度	t_{TCKL}	0.4	0.6	0.4	0.6	0.4	0.6		
边界扫描	TRST 脉冲宽度	t_{TRSW}	20	—	20	—	20	—	tcyc	图 22.19
	TRST 设置时间	t_{TRSS}	350	—	250	—	250	—		
边界扫描	TDI 设置时间	t_{TDIS}	80	—	30	—	20	—	ns	图 22.20
	TDI 保持时间	t_{TDIH}	10	—	10	—	10	—		
	TMS 设置时间	t_{TMSS}	80	—	30	—	20	—		
	TMS 保持时间	t_{TMSH}	10	—	10	—	10	—		
	TDO 延迟时间	t_{TDOD}	—	100	—	40	—	35		

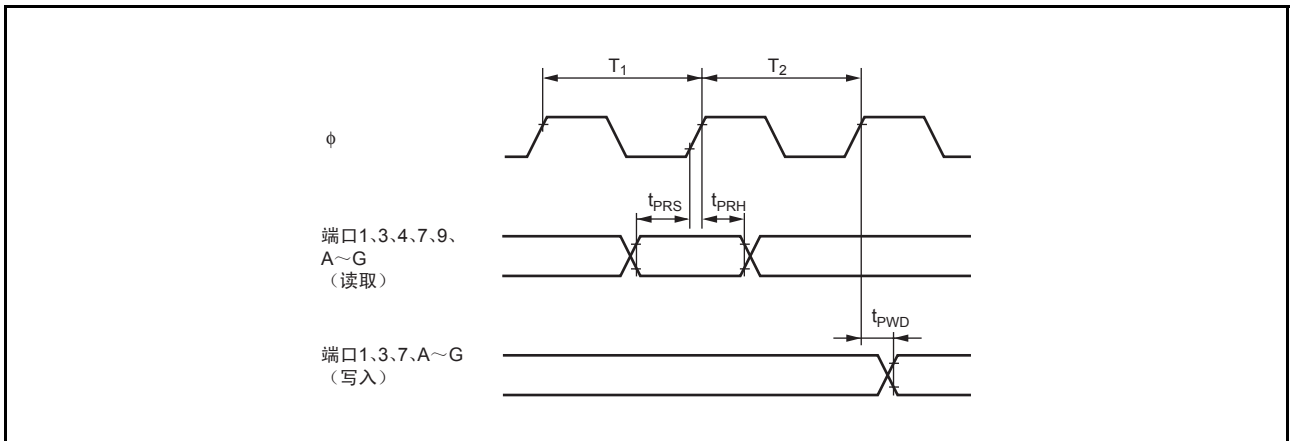


图 22.12 I/O 端口输入 / 输出时序

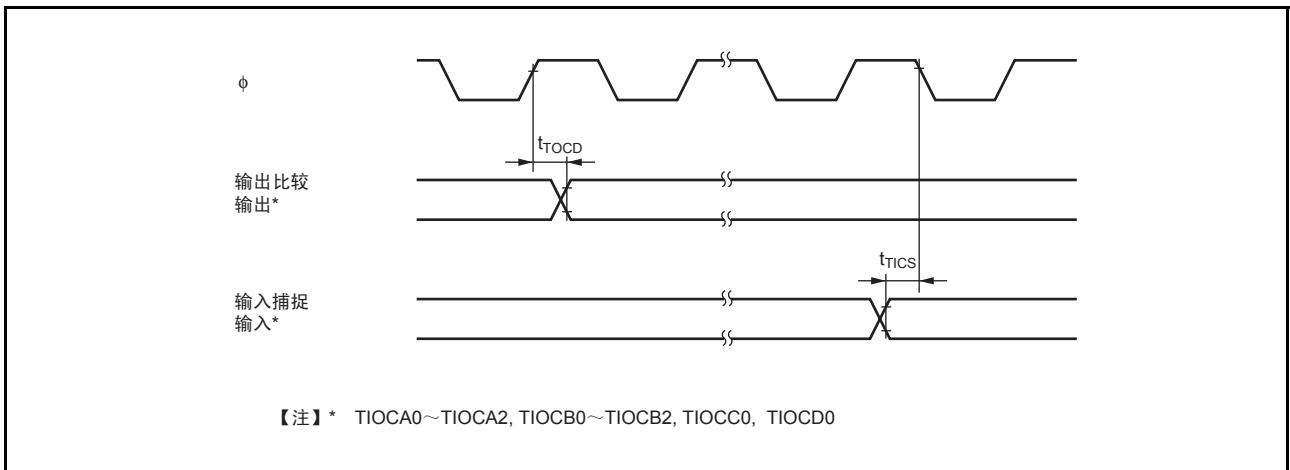


图 22.13 TPU 输入 / 输出时序

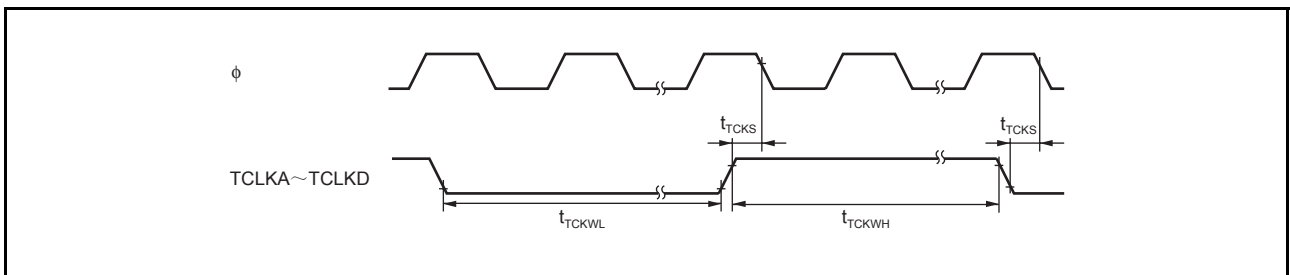


图 22.14 TPU 时钟输入时序

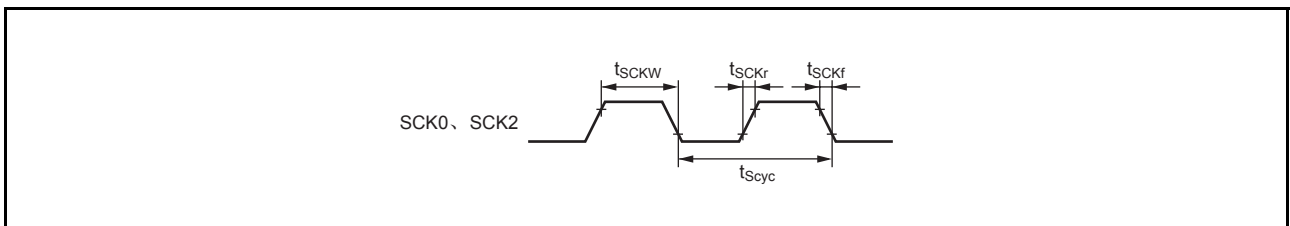


图 22.15 SCK 时钟输入时序

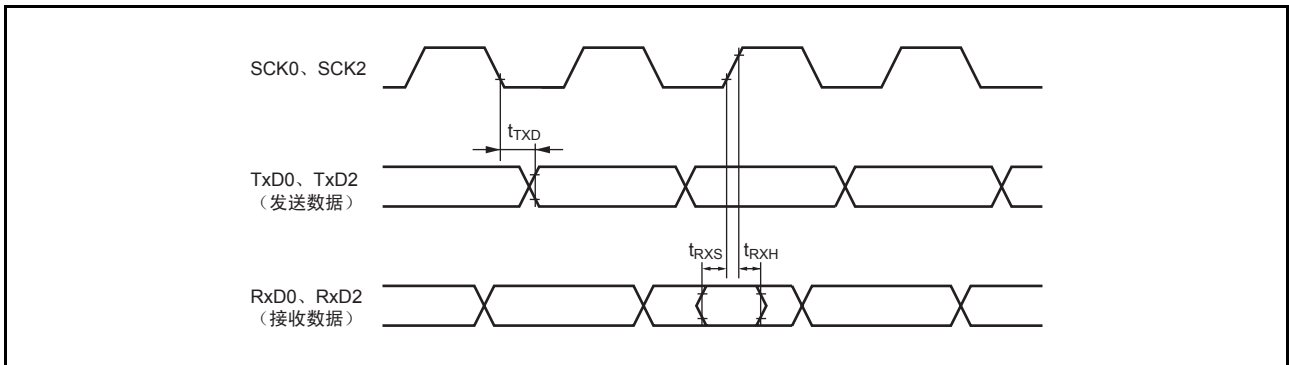


图 22.16 SCI 输入输出时序 / 时钟同步模式

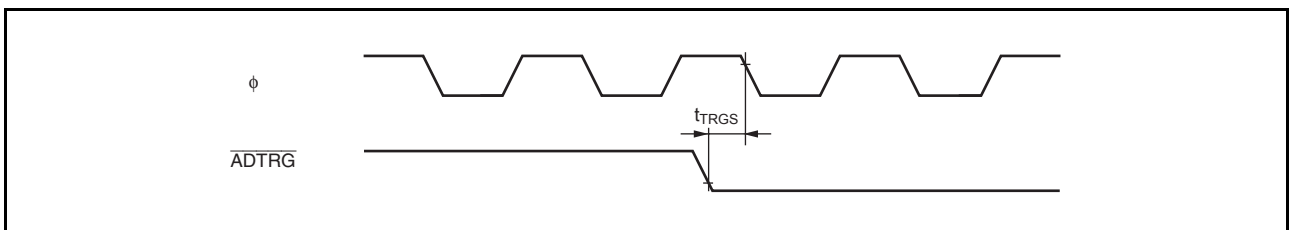


图 22.17 A/D 转换器外部触发输入时序

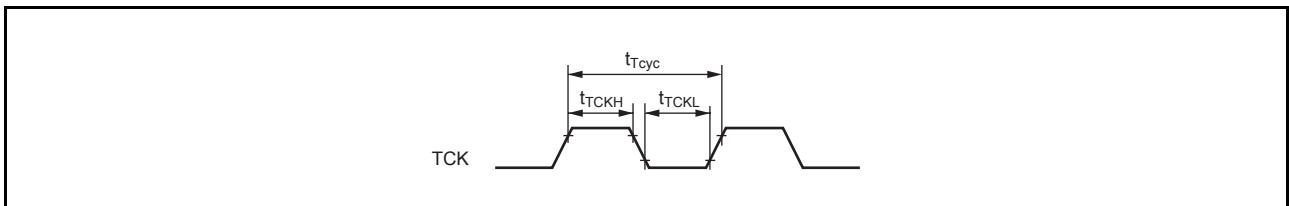


图 22.18 边界扫描 TCK 输入时序

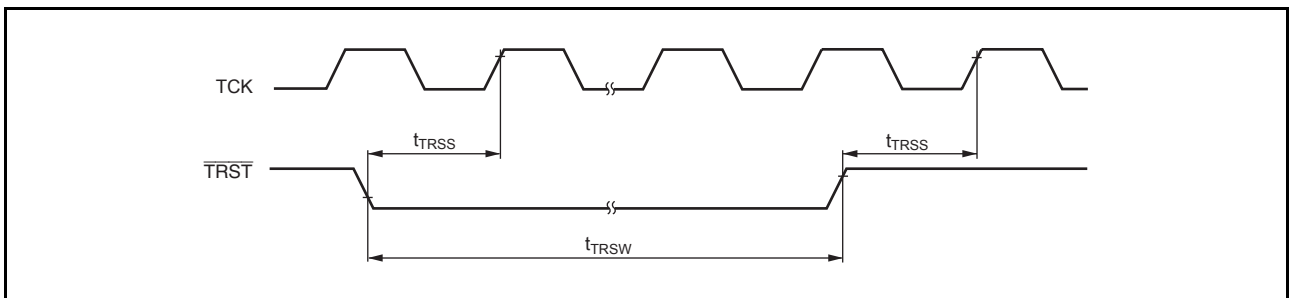


图 22.19 边界扫描 TRST 输入时序 (复位保持时)

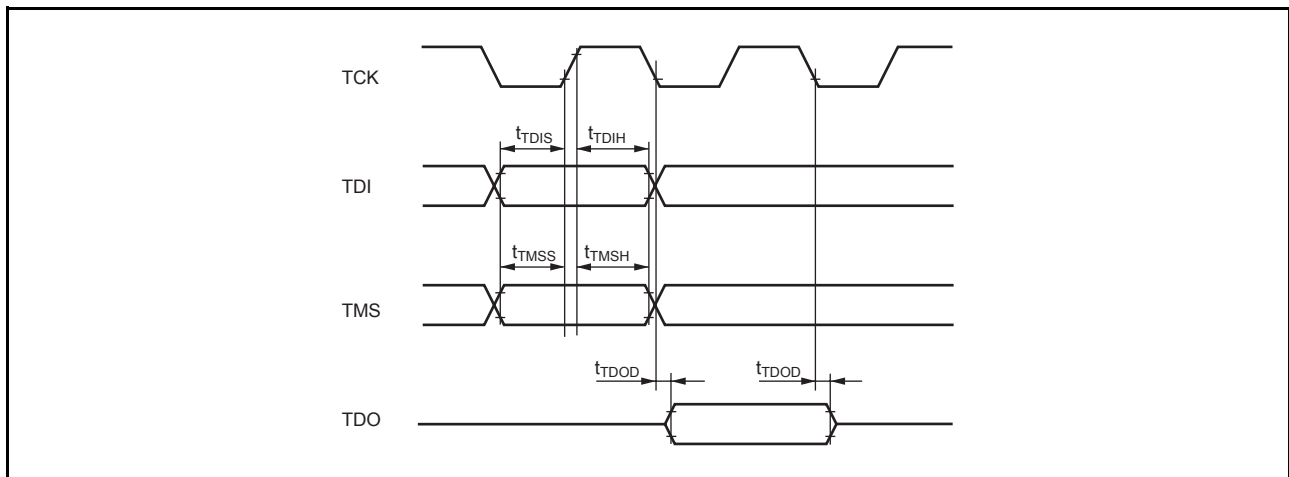


图 22.20 边界扫描传送时序

22.5 USB 特性

内部 USB 收发器使用时的 USB 特性（USD+，USD- 管脚特性）如表 22.8 所示。

表 22.8 使用内部 USB 收发器时的 USB 特性（USD+、USD- 管脚特性）

条件： $V_{CC} = PLLV_{CC} = DrV_{CC} = 3.0 \sim 3.6V$ ， $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$ ， $f = 16MHz, 24MHz$ ， $T_a = -20 \sim +75^{\circ}C$ （普通规格产品）， $T_a = -40 \sim +85^{\circ}C$ （宽温度范围规格产品）

项目	符号	条件		单位	测定条件		
		min.	max.				
输入特性	输入高电平电压	V_{IH}	2.0	—	V	图 22.21、 图 22.22	
	输入低电平电压	V_{IL}	—	0.8	V		
	差动输入敏感度	V_{DI}	0.2	—	V		$ (D+) - (D-) $
	差动共模列	V_{CM}	0.8	2.5	V		
输出特性	输出高电平电压	V_{OH}	2.8	—	V	$I_{OH} = -200\mu A$	
	输出低电平电压	V_{OL}	—	0.3	V	$I_{OL} = 2mA$	
	交叉电压	V_{CRS}	1.3	2.0	V		
	上升时间	t_R	4	20	ns		
	下降时间	t_F	4	20	ns		
	上升 / 下降时间匹配	t_{RFM}	90	111.11	%	(T_R/T_F)	
	输出电阻	Z_{DRV}	28	44	Ω	包含 $R_s = 24 \Omega$	

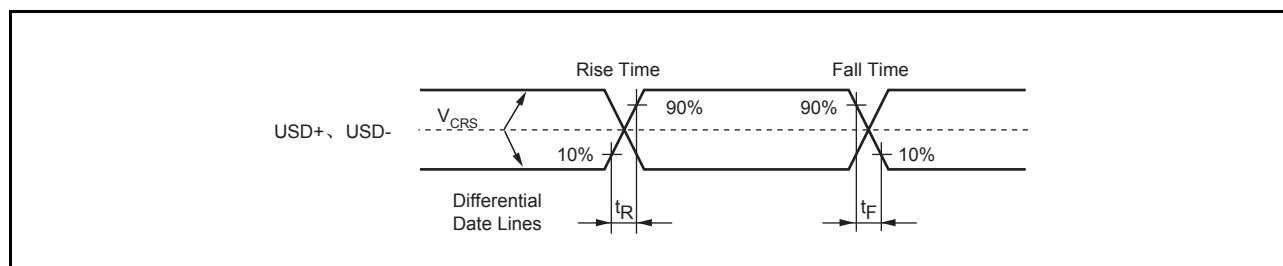


图 22.21 数据信号时序

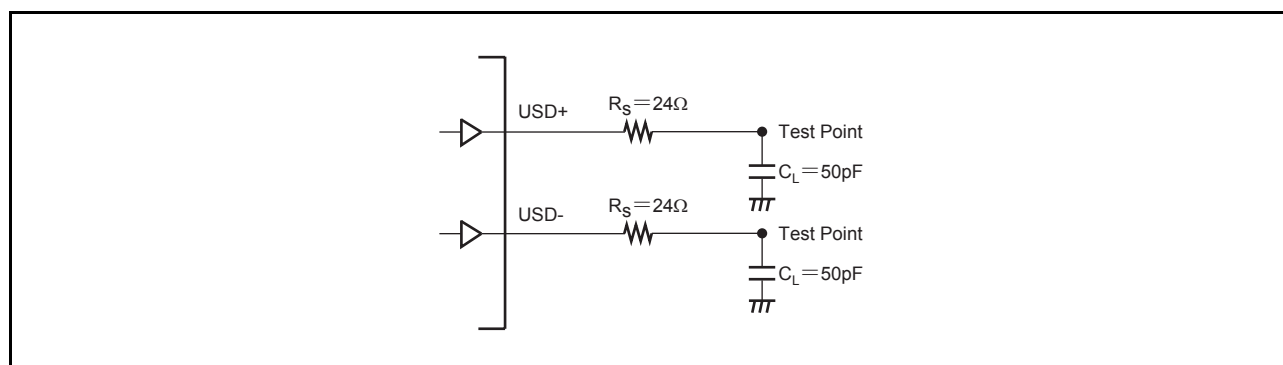


图 22.22 负载条件

22.6 A/D 转换特性

A/D 转换特性如表 22.9 所示。

表 22.9 A/D 转换特性

条件 A: $V_{CC} = PLLV_{CC} = DrV_{CC} = 2.4 \sim 3.6V$, $V_{ref} = 2.4V \sim V_{CC}$, $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$, $f = 32.768kHz, 6MHz$, $T_a = -20 \sim +75^{\circ}C$ (普通规格产品), $T_a = -40 \sim +85^{\circ}C$ (宽温度范围规格产品)

条件 B: $V_{CC} = PLLV_{CC} = DrV_{CC} = 2.7 \sim 3.6V$, $V_{ref} = 2.7V \sim V_{CC}$, $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$, $f = 32.768kHz, 6 \sim 16MHz$, $T_a = -20 \sim +75^{\circ}C$ (普通规格产品), $T_a = -40 \sim +85^{\circ}C$ (宽温度范围规格产品)

条件 C: $V_{CC} = PLLV_{CC} = DrV_{CC} = 3.0 \sim 3.6V$, $V_{ref} = 3.0V \sim V_{CC}$, $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$, $f = 32.768kHz, 6 \sim 24MHz$, $T_a = -20 \sim +75^{\circ}C$ (普通规格产品), $T_a = -40 \sim +85^{\circ}C$ (宽温度范围规格产品)

项目	条件 A			条件 B			条件 C			单位
	min.	typ.	max.	min.	typ.	max.	min.	typ.	max.	
分辨率	10	10	10	10	10	10	10	10	10	位
转换时间	21.8	—	—	8.1	—	—	10.6	—	—	μs
模拟输入容量	—	—	20	—	—	20	—	—	20	pF
容许信号源接口	—	—	5	—	—	5	—	—	5	k Ω
非线性误差	—	—	± 6.0	—	—	± 6.0	—	—	± 6.0	LSB
偏移误差	—	—	± 4.0	—	—	± 4.0	—	—	± 4.0	LSB
满刻度误差	—	—	± 4.0	—	—	± 4.0	—	—	± 4.0	LSB
量化误差	—	—	± 0.5	—	—	± 0.5	—	—	± 0.5	LSB
绝对精度	—	—	± 8.0	—	—	± 6.0	—	—	± 6.0	LSB

22.7 闪存特性

闪存特性如表 22.10 所示。

表 22.10 闪存特性

条件: $V_{CC} = PLLV_{CC} = DrV_{CC} = 3.0 \sim 3.6V$ 、 $V_{ref} = 2.7V \sim V_{CC}$ 、 $V_{SS} = PLLV_{SS} = DrV_{SS} = 0V$ 、 $T_a = -20 \sim +75^{\circ}C$ (写入/擦除时的运行温度范围)

项目		符号	min.	typ.	max.	单位
写入时间 *1*2*4		t_P	—	10	200	ms/128 字节
擦除时间 *1*3*5		t_E	—	50	1000	ms/ 块
改写次数		N_{WEC}	100*6	10000*7	—	次
数据保持时间 *8		t_{DRP}	10	—	—	年
写入时	PSU1 置位后的等待时间 *1	y	50	50	—	μs
	P1 位置位后的等待时间 *1*4	$z0$	28	30	32	μs
		$z1$	198	200	202	μs
		$z2$	8	10	12	μs
	清除 P1 位后的等待时间 *1	α	5	5	—	μs
	清除 PSU1 位后的等待时间 *1	β	5	4	—	μs
	PV1 位置位后的等待时间 *1	γ	4	2	—	μs
	H'FF 虚写后的等待时间 *1	ϵ	2	2	—	μs
	清除 PV1 位后的等待时间 *1	η	2	—	—	μs
	最大写入次数 *1*4	$N1$	—	—	6*4	次
$N2$		—	—	994*4	次	
共用	置 SWE1 位后的等待 *1	x	1	1	—	μs
	清除 SWE1 位后的等待时间 *1	θ	100	100	—	μs
擦除时	ESU1 位置位后的等待时间 *1	y	100	100	—	μs
	E1 位置位后的等待时间 *1*5	z	10	10	100	μs
	清除 E1 位后的等待时间 *1	α	10	10	—	μs
	清除 ESU1 位后的等待时间 *1	β	10	10	—	μs
	EV1 位置位后的等待时间 *1	γ	20	20	—	μs
	H'FF 虚写后的等待时间 *1	ϵ	2	2	—	μs
	清除 EV1 位后的等待时间 *1	η	4	4	—	μs
	最大擦除次数 *1*5	N	—	—	100	次

【注】 *1 按照写入 / 擦除的算法设定各时间。

*2 每 128 字节的写入时间（表示将闪存控制寄存器 1（FLMCR1）的 P 位置位的总时间。不含写入验证时间。）

*3 块擦除的时间（表示将 FLMCR1 的 E1 位置位的总时间。不含擦除验证时间。）

*4 写入时间的最大值

$$\begin{aligned} t_p(\max) &= P1 \text{ 置位后的等待时间 (z)} \times \text{最大写入次数 (N1 + N2)} \\ &= (z0 + z2) \times 6 + z1 \times 994 \end{aligned}$$

*5 擦除时间最大值

$$t_E(\max) = E1 \text{ 位置位后的等待时间 (z)} \times \text{最大擦除次数 (N)}$$

*6 保证改写后的所有特性的 min 次数（保证在 1 ~ min 值的范围内）。

*7 25°C 时的参考值（通常改写至此值后发挥功能）

*8 在包含 min 值的规格范围内改写的数据保持特性。

22.8 使用注意事项

22.8.1 设计印刷电路板注意事项

实际设计希望充分考虑 LSI 开关过渡电流所产生辐射噪音的相对对策后，再进行使用。具体措施例如下所示。

1. 使用有电源层及 GND 层的多层印刷电路板。
2. LSI 的 Vcc - GND (Vss) 与 PLLVcc - PLLGND 之间安装旁路电容 (0.1μF 左右)。

22.8.2 F-ZTAT 版与掩模型 ROM 版的特性

F-ZTAT 版与掩模型 ROM 版虽然均满足本手册记述的电特性，但因制作工艺、内部 ROM 及版图模式等的不同，有时电特性的实际值和运行容限、噪音容限等也有所不同。

在使用 F-ZTAT 版进行系统评估试验的情况下，在切换至掩模型 ROM 版时，对掩模型 ROM 版也必须进行同样的评估试验。

附录

附录 1. 各管脚状态中的 I/O 端口状态

端口名管脚名	MCU 运行模式	加电复位	手动复位	硬件待机 模式	软件待机模式、监 视模式	总线权释放 状态	程序运行状态 睡眠模式	
P17 ~ P14	4 ~ 7	T	keep	T	keep	keep	输入输出端口	
P13/A23 P12/A22 P11/A21	7	T	keep	T	keep	keep	输入输出端口	
	根据 AEn 位选择地 址输出	4~6	T	keep	T	[OPE = 0] T [OPE = 1] keep	T	地址输出
	端口选择	4~6	T	keep	T	keep	keep	输入输出端口
P10/A20	7	T	keep	T	keep	keep	输入输出端口	
	根据 AEn 位选择地 址输出	4、5	L	keep	T	[OPE = 0] T [OPE = 1] keep	T	地址输出
		6	T					
	端口选择	4~6	T*1	keep	T	keep	keep	输入输出端口
端口 3	4~7	T	keep	T	keep	keep	输入输出端口	
端口 4	4~7	T	T	T	T	T	输入端口	
P77 ~ P75*3	7	T	keep	T	keep	keep	输入输出端口	
P74*2	4~7	T	keep	T	keep	keep	输入输出端口	
P71/ $\overline{\text{CS5}}$ *2 P70/ $\overline{\text{CS4}}$ *2	7	T	keep	T	keep	keep	输入输出端口	
	4~6	T	keep	T	[DDR • OPE = 0] T [DDR • OPE = 1] H	T	[DDR = 0] 输入端口 [DDR = 1] $\overline{\text{CS5}} \sim \overline{\text{CS4}}$	
端口 9	4~7	T	T	T	[DAOEn = 1] keep [DAOEn = 0] T	keep	输入端口	
端口 A	7	T	keep	T	keep	keep	输入输出端口	
	根据 AEn 位选择地 址输出	4、5	L	keep	T	[OPE = 0] T [OPE = 1] keep	T	地址输出
		6	T					
	端口选择	4~6	T*1	keep	T	keep	keep	输入输出端口

端口名管脚名		MCU 运行模式	加电复位	手动复位	硬件待机 模式	软件待机模式、监 视模式	总线权释放 状态	程序运行状态 睡眠模式
端口 B* ²		7	T	keep	T	keep	keep	输入输出端口
	根据 AEn 位 选择地址 输出	4、5	L	keep	T	[OPE = 0] T [OPE = 1] keep	T	地址输出
		6	T					
	端口选择	4~6	T* ¹	keep	T	keep	keep	输入输出端口
端口 C* ²		4、5	L	keep	T	[OPE = 0] T [OPE = 1] keep	T	地址输出
		6	T	keep	T	[DDR • OPE = 0] T [DDR • OPE = 1] keep	T	[DDR = 0] 输入端口 [DDR = 1] 地址输出
		7	T	keep	T	keep	keep	输入输出端口
端口 D* ²		4~6	T	T	T	T	T	数据总线
		7	T	keep	T	keep	keep	输入输出端口
端口 E	8 位总线	4~6	T	keep	T	keep	keep	输入输出端口
	16 位 总线	4~6	T	T	T	T	T	数据总线
		7	T	keep	T	keep	keep	输入输出端口
PF7/φ		4~6	时钟输出	[DDR = 0] 输入端口 [DDR = 1] 时钟输出	T	[DDR = 0] 输入端口 [DDR = 1] H	[DDR = 0] 输入端口 [DDR = 1] 时钟输出	[DDR = 0] 输入端口 [DDR = 1] 时钟输出
		7	T	keep	T	[DDR = 0] 输入端口 [DDR = 1] H	[DDR = 0] 输入端口 [DDR = 1] 时钟输出	[DDR = 0] 输入端口 [DDR = 1] 时钟输出
PF6/ \overline{AS} * ² PF5/ \overline{RD} * ² PF4/ \overline{HWR} * ²		4~6	H	H	T	[OPE = 0] T [OPE = 1] H	T	\overline{AS} 、 \overline{RD} 、 \overline{HWR}
		7	T	keep	T	keep	keep	输入输出端口
PF3/ \overline{LWR}		7	T	keep	T	keep	keep	输入输出端口
	8 位总线	4~6	(模式 4) H	keep	T	keep	keep	输入输出端口
	16 位 总线	4~6	(模式 5、6) T	H	T	[OPE = 0] T [OPE = 1] H	T	\overline{LWR}

端口名管脚名	MCU 运行模式	加电复位	手动复位	硬件待机 模式	软件待机模式、监 视模式	总线权释放 状态	程序运行状态 睡眠模式
PF2/ $\overline{\text{WAIT}}^{*2}$	4~6	T	keep	T	[WAITE = 0] keep [WAITE = 1] T	[WAITE = 0] keep [WAITE = 1] T	[WAITE = 0] 输入输出端口 [WAITE = 1] $\overline{\text{WAIT}}$
	7	T	keep	T	keep	keep	输入输出端口
PF1/ $\overline{\text{BACK}}^{*2}$	4~6	T	keep	T	[BRLE = 0] keep [BRLE = 1] H	L	[BRLE = 0] 输入输出端口 [BRLE = 1] $\overline{\text{BACK}}$
	7	T	keep	T	keep	keep	输入输出端口
PF0/ $\overline{\text{BREQ}}$	4~6	T	keep	T	[BRLE = 0] keep [BRLE = 1] T	T	[BRLE = 0] 输入输出端口 [BRLE = 1] $\overline{\text{BREQ}}$
	7	T	keep	T	keep	keep	输入输出端口
PG4/ $\overline{\text{CS0}}^{*2}$	4、5	H	keep	T	[DDR • OPE = 0] T	T	[DDR = 0] 输入端口
	6	T			[DDR • OPE = 1] H		[DDR = 1] $\overline{\text{CS0}}$ (睡眠模式) 時 H
	7	T	keep	T	keep	keep	输入输出端口
PG3/ $\overline{\text{CS1}}^{*2}$ PG2/ $\overline{\text{CS2}}^{*2}$ PG1/ $\overline{\text{CS3}}$	4~6	T	keep	T	[DDR • OPE = 0] T [DDR • OPE = 1] H	T	[DDR = 0] 输入端口 [DDR = 1] $\overline{\text{CS1}} \sim \overline{\text{CS3}}$
	7	T	keep	T	keep	keep	输入输出端口
PG0 ^{*3}	4~7	T	keep	T	keep	keep	输入输出端口

【符号说明】

H : 高电平

L : 低电平

T : 高阻抗

keep : 输入端口高阻抗、保持输入端口。

DDR : 数据方向寄存器

OPE : 输出端口允许

WAITE: 等待输入允许

BRLE : 总线释放允许

【注】 *1 模式 4、5 中为 L (地址输入)

*2 仅针对 H8S/2218 群

*3 仅针对 H8S/2212 群

附录 2. 型号一览

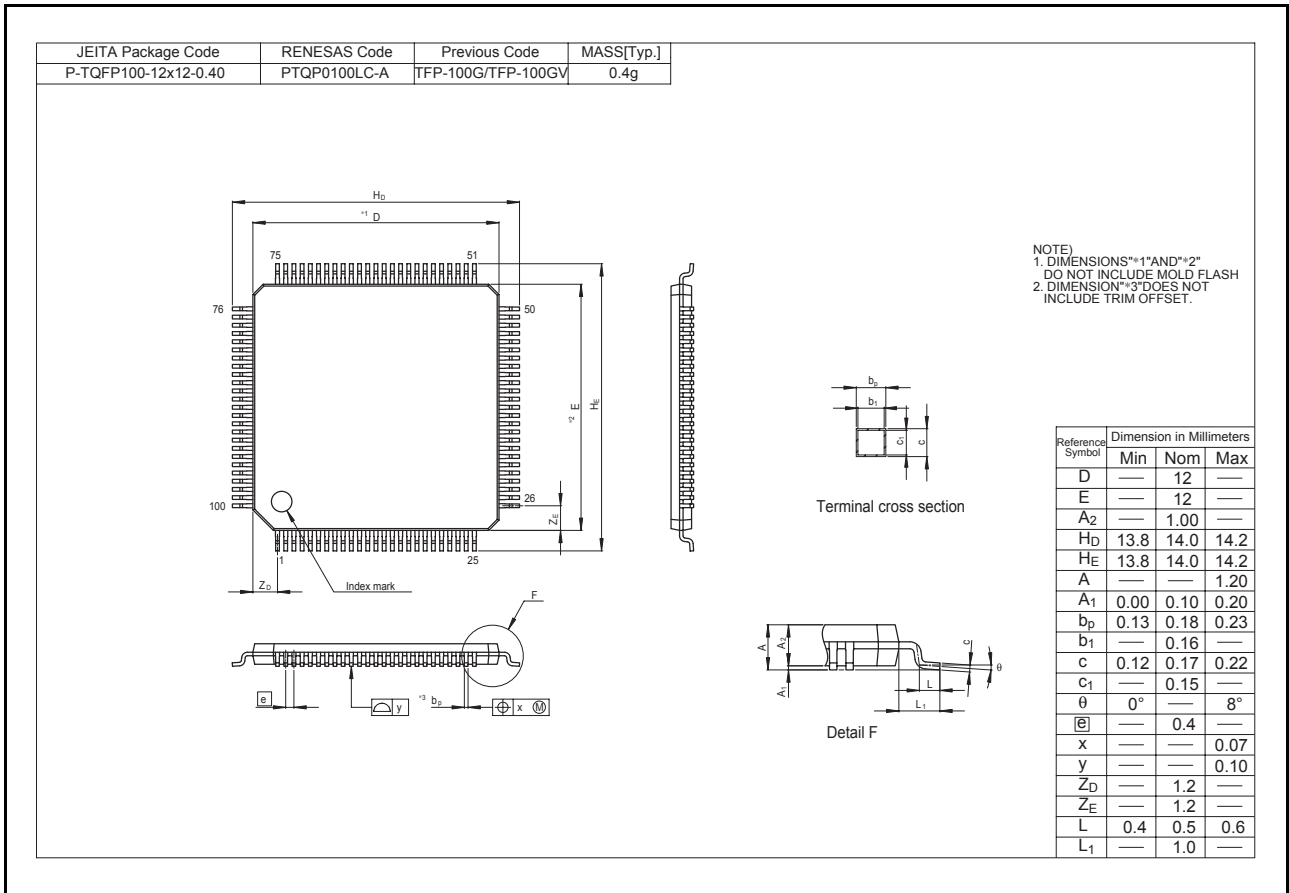
产品分类		产品型号	目录型号	标记型号	封装 (代码)
H8S/2218 群	闪存版	HD64F2218	HD64F2218TF24	F2218TF24	100 管脚 TQFP (TFP-100G、TFP-100GV)
			HD64F2218BR24	64F2218BR24	112 管脚 P-LFBGA (BP-112、BP-112V)
		HD64F2218U	HD64F2218UTF24	F2218UTF24	100 管脚 TQFP (TFP-100G、TFP-100GV)
			HD64F2218UBR24	64F2218UBR24	112 管脚 P-LFBGA (BP-112、BP-112V)
	掩模 ROM 版	HD6432217	HD6432217(***)TF	2217(***)TF	100 管脚 TQFP (TFP-100G、TFP-100GV)
			HD6432217(***)BR	2217(***)BR	112 管脚 P-LFBGA (BP-112、BP-112V)
H8S/2212 群	闪存版	HD64F2212	HD64F2212FP24	2212FP24	64 管脚 LQFP (FP-64E、FP-64EV)
			HD64F2212NP24	F2212NP24	64 管脚 VQFN (TNP-64B、TNP-64BV)
		HD64F2212U	HD64F2212UFP24	2212UFP24	64 管脚 LQFP (FP-64E、FP-64EV)
			HD64F2212UNP24	F2212UNP24	64 管脚 VQFN (TNP-64B、TNP-64BV)
		HD64F2211	HD64F2211FP24	2211FP24	64 管脚 LQFP (FP-64E、FP-64EV)
			HD64F2211NP24	F2211NP24	64 管脚 VQFN (TNP-64B、TNP-64BV)
		HD64F2211U	HD64F2211UFP24	2211UFP24	64 管脚 LQFP (FP-64E、FP-64EV)
			HD64F2211UNP24	F2211UNP24	64 管脚 VQFN (TNP-64B、TNP-64BV)
	掩模 ROM 版	HD6432211	HD6432211(***)FP	2211(***)FP	64 管脚 LQFP (FP-64E、FP-64EV)
			HD6432211(***)NP*1	2211(***)NP	64 管脚 VQFN (TNP-64B、TNP-64BV)
		HD6432210	HD6432210(***)FP	2210(***)FP	64 管脚 LQFP (FP-64E、FP-64EV)
			HD6432210(***)NP*1	2210(***)NP	64 管脚 VQFN (TNP-64B、TNP-64BV)

【符号说明】

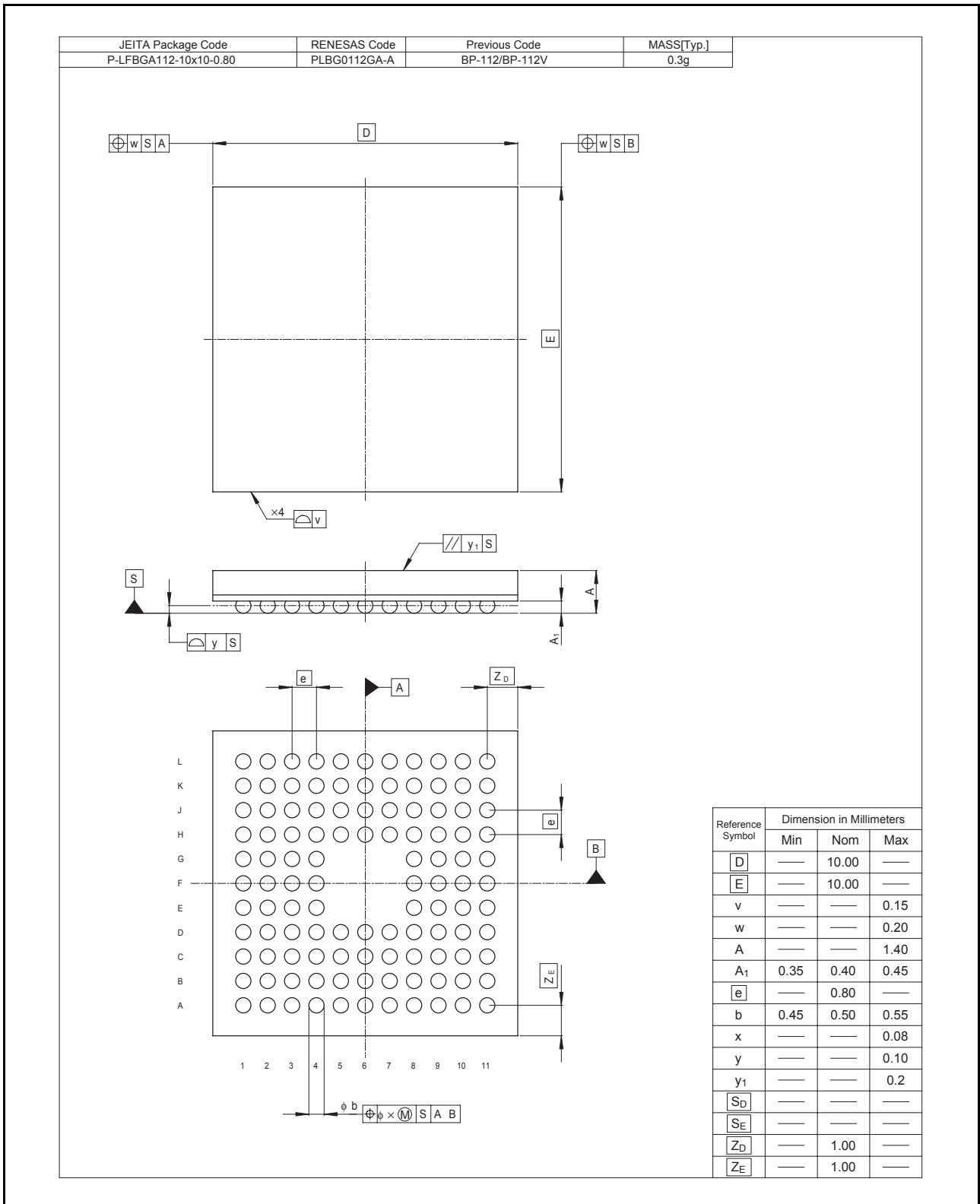
(***) 为 ROM 代码。

【注】 *1 开发中。

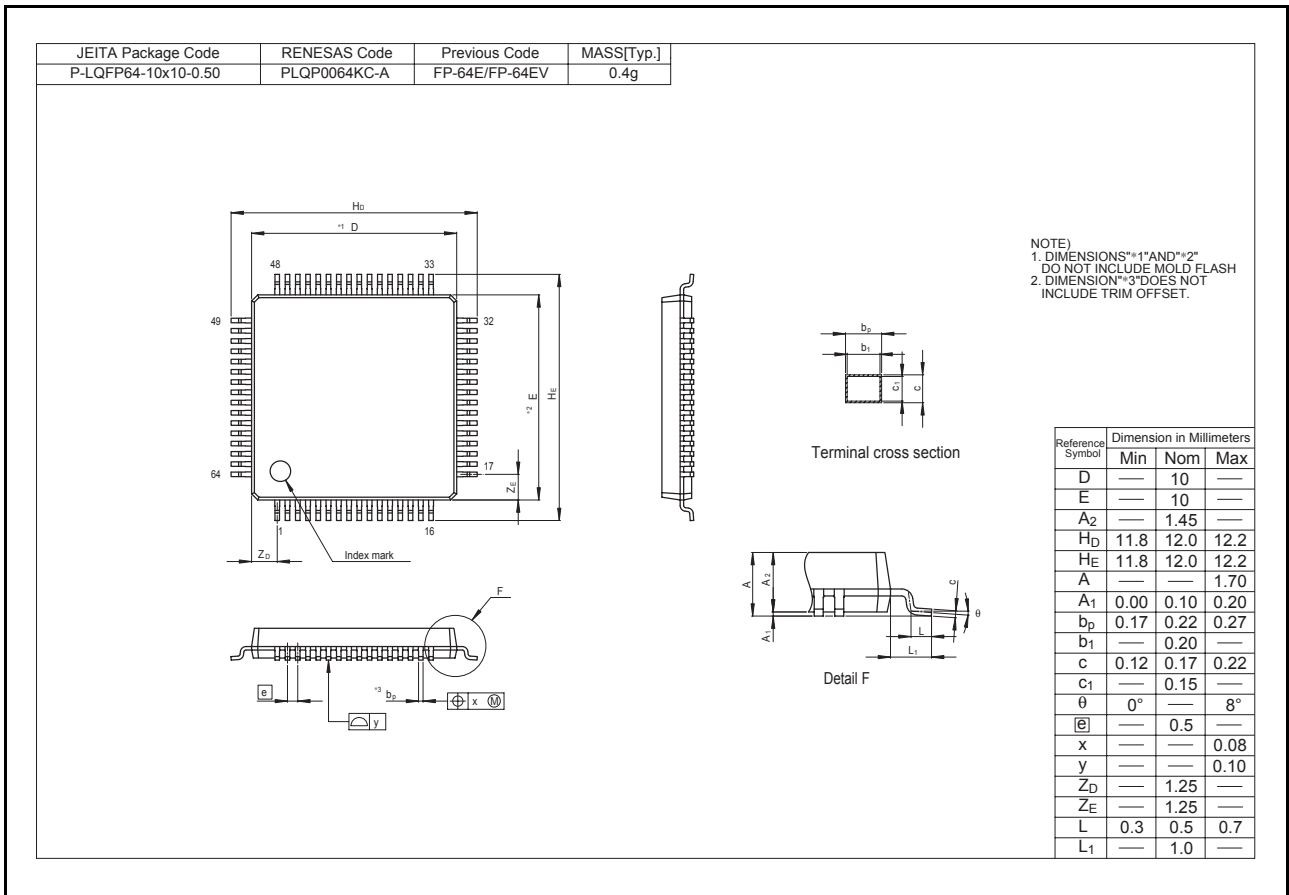
附录 3. 外形尺寸图



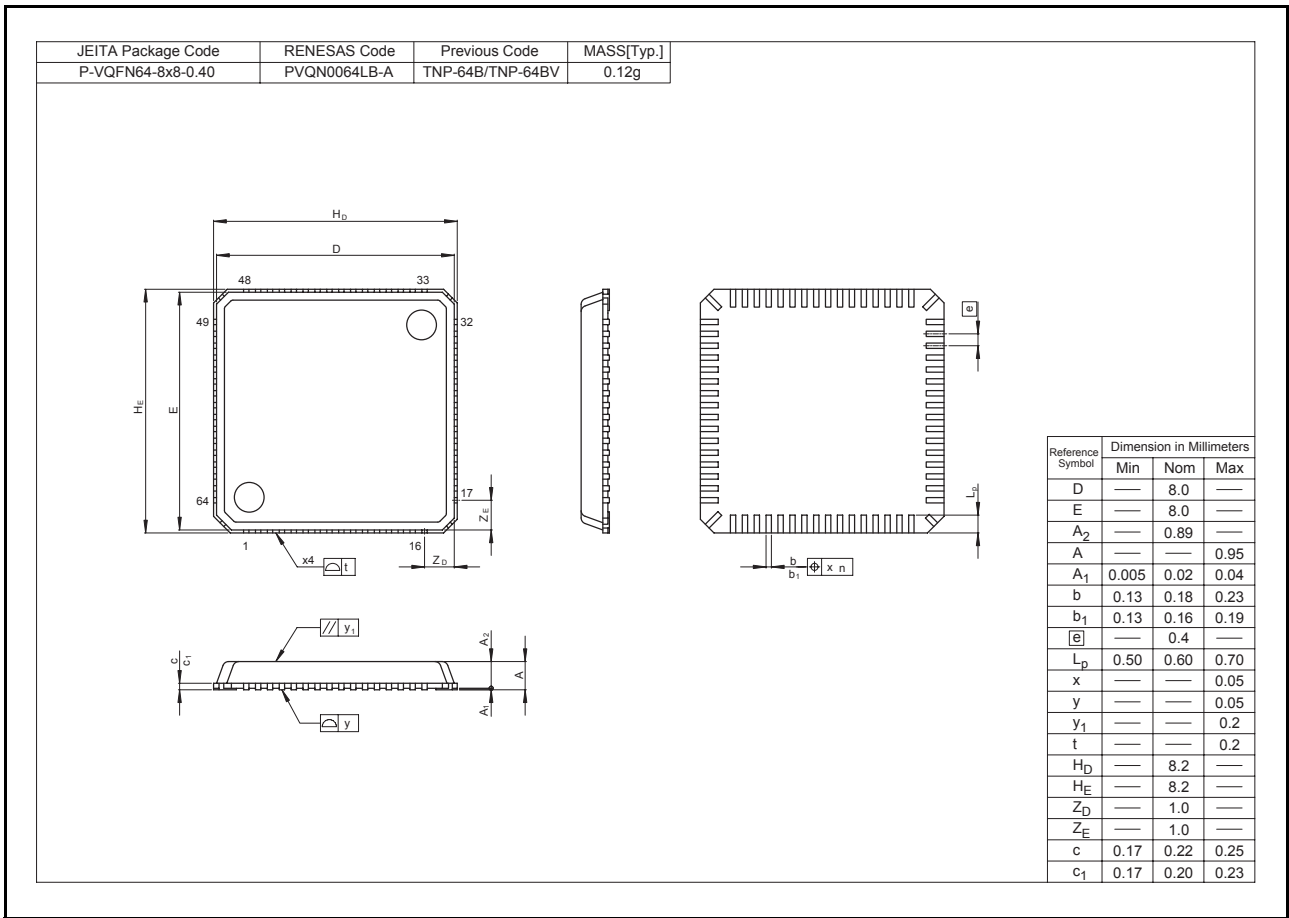
附图 3.1 外形尺寸图 (TFP-100G、TFP-100GV)



附图 3.2 外形尺寸图 (BP-112、BP-112V)



附图 3.3 外形尺寸图 (FP-64E、FP-64EV)



附图 3.4 外形尺寸图 (TNP-64B、TNP-64BV)

索引

Numerics

16 位定时器脉冲单元..... 224

A

A/D 转换器..... 445
A/D 转换器的启动..... 267
A/D 转换时间..... 454
ADI..... 455

B

Bcc..... 41,47
Bulk in 传送..... 428
Bulk out 传送..... 429
比特率..... 326
编程 / 编程验证..... 482
编程器模式..... 487
边界扫描..... 374

C

擦除 / 擦除验证..... 484
擦除块..... 470
CPU 运行模式..... 29
操作字段..... 49
程序计数器..... 36
程序计数器相对寻址..... 51
串行通信接口..... 301
存储器间接寻址..... 52
存储器周期..... 107
错误保护..... 486

D

DMA 传送说明..... 434
带位移量的寄存器间接寻址..... 50
单板上编程..... 472
单通道模式..... 452
地址空间..... 33
地址映射..... 65
读取时刻顺序..... 298
堆栈指针 (SP)..... 35
堆栈状态..... 75

E

EA 扩展字段..... 49

F

仿真..... 480
复位..... 70

复位异常处理..... 71

G

高级模式..... 31
跟踪位..... 36
跟踪异常处理..... 73
工作模式的选择..... 60
挂起操作..... 430
挂起 / 恢复..... 418

H

后增寄存器间接寻址..... 50
缓冲运行..... 253

J

寄存器

ABWCR..... 98,529,535,542
ADCR..... 450,532,539,545
ADCSR..... 449,532,539,545
ADDR..... 448,531,539,545
ASTCR..... 98,529,535,542
BCRH..... 101,529,535,542
BCRL..... 102,529,535,542
BRR..... 326,531,538,545
BSCANR..... 378
BYPASS..... 378
DMACR..... 136,531,538,544
EBR1..... 470,532,539,545
EBR2..... 470,532,539,545
ETCR..... 135,529,535,542
FLMCR1..... 469,532,539,545
FLMCR2..... 470,532,539,545
IDCODE..... 377
IER..... 80,528,534,541
INSTR..... 376
IOAR..... 134,529,535,542
IPR..... 79,529,535,542
ISCR..... 81,528,534,541
ISR..... 82,528,534,541
LPWRCR..... 497,528,534,541
MAR..... 134,529,535,542
MDCR..... 61,528,534,541
MSTPCR..... 511,528,534,541
P1DDR..... 178,528,534,541
P1DR..... 178,530,536,543
P3DDR..... 184,528,534,541
P3DR..... 184,530,536,543
P3ODR..... 185,529,535,541
P7DDR..... 188,528,534,541
P7DR..... 189,530,536,543
PADDR..... 193,528,534,541

PADR	193,530,536,543	TIER	241,530,537,543
PAODR	194,529,535,541	TIOR	232,530,537,543
PAPCR	194,528,534,541	TMDR	231,530,537,543
PBDDR	198,528,534,541	TSR	242,530,537,543
PBDR	198,530,536,543	TSTR	244,529,535,541
PBPCR	199,528,534,541	TSYR	244,529,535,541
PCDDR	202,528,534,541	UCTLR	393,527,533,540
PCDR	202,530,536,543	UCVR	407,527,533,540
PCPCR	203,528,535,541	UDMAR	394,527,533,540
PDDDR	206,528,534,541	UDRR	395,527,533,540
PDDR	206,530,536,543	UDSR	407,527,533,540
PDPCR	207,528,535,541	UEDR0i	399,527,533,540
PEDDR	210,528,534,541	UEDR0o	399,527,533,540
PEDR	210,530,536,543	UEDR0s	399,527,533,540
PEPCR	211,528,535,541	UESTL0	398,527,533,540
PFCR	103,528,534,541	UESTL1	398,527,533,540
PFDDR	215,528,534,541	UESZ0o	400,527,533,540
PFDR	215,530,536,543	UESZ2	400,527,533,540
PGDDR	220,528,534,541	UFCLR0	397,527,533,540
PGDR	221,530,536,543	UIER0	404,527,533,540
PORT1	179,532,539,545	UIER1	404,527,533,540
PORT3	185,532,539,545	UIER3	405,527,533,540
PORT4	187,532,539,545	UIFR0	401,527,533,540
PORT7	190,532,539,545	UIFR1	402,527,533,540
PORT9	192,532,539,545	UIFR3	403,527,533,540
PORTA	194,532,539,545	UISR0	405,527,533,540
PORTB	199,532,539,545	UISR1	406,527,533,540
PORTC	203,532,539,545	UISR3	406,527,533,540
PORTD	207,532,539,545	UTRG0	396,527,533,540
PORTE	211,532,539,545	UTSTR0	408,527,533,540
PORTF	216,532,539,545	UTSTR1	409,527,534,540
PORTG	221,532,539,545	WCRH	99,529,535,542
RAMER	471,529,535,542	WCRL	99,529,535,542
RDR	304,531,538,544	寄存器间接寻址	50
RHRDR	292	寄存器直接寻址	50
RMINDR	292,531,537,544	寄存器字段	49
RSECDR	531,537,544	奇偶校验错误	340
RSTCSR	284,531,538,544	间隔定时器模式	286
RTCCR1	294,531,538,544	交替输出	250
RTCCR2	295,531,538,544	绝对地址	51
RTCCSR	296,531,538,544		
RWKDR	293,531,538,544		
SBYCR	508,528,534,541	K	
SCKCR	496,528,534,541	看门狗定时器	281
SCMR	315,531,538,544	控制传送	422
SCR	308,531,538,544	块传送指令	48
SCRX	471,528,534,540	扩展控制寄存器 (EXR)	36
SEMRA_0	316,528,534,541		
SEMRB_0	318,528,534,541		
SMR	305,531,538,544	L	
SSR	311,531,538,544	立即数寻址	51
SYSCR	62,528,534,541	逻辑运算	45
TCNT	244,282,530,531,537,538,543,544		
TCR	228,530,536,543		
TCSR	283,531,538,544	M	
TDR	304,531,538,544	MARK 状态	369
TGR	244,530,543		

N		外部触发.....	455
NMI 中断.....	82	位操作指令.....	46
内部总线主控器.....	95		
P		X	
PLL 电路.....	502	系统控制指令.....	48
PWM 模式.....	257	先减寄存器间接寻址.....	50
普通模式.....	29	陷阱指令异常处理.....	74
		相位计数模式.....	261
		写入单位.....	466,467
		寻址模式.....	50
R		Y	
软件保护.....	486	掩模型 ROM.....	494
		异步模式.....	333
S		异常处理.....	68
扫描模式.....	453	溢出错误.....	340
闪存.....	567	移位指令.....	45
上溢.....	287	引导模式.....	472
实时时钟 (RTC).....	290	硬件保护.....	486
时钟振荡器.....	495	用比较匹配输出波形功能.....	249
数据传送指令.....	43	用户编程模式.....	479
输入捕捉功能.....	250	有效地址.....	50,53
算数运算指令.....	44		
		Z	
T		帧错误.....	340
TCI0V.....	266	指令系统.....	41
TCI1U.....	266	中断 in 传送.....	427
TCI1V.....	266	中断控制模式.....	85
TCI2U.....	266	中断控制器.....	77
TCI2V.....	266	中断屏蔽位.....	37
TGI0A.....	266	中断请求屏蔽级别.....	36
TGI0B.....	266	中断异常处理.....	73
TGI0C.....	266	中断异常处理向量表.....	84
TGI0D.....	266	中断优先级寄存器.....	77
TGI1A.....	266	中止状态.....	369
TGI1B.....	266	周期计数操作.....	248
TGI2A.....	266	转移指令.....	47
TGI2B.....	266	自由运行计数操作.....	248
TRAPA 指令.....	51,74	总线仲裁.....	128
条件码寄存器.....	37	总线周期.....	108
条件字段.....	49		
同步工作.....	252		
通用串行总线.....	389		
通用寄存器.....	35		
U			
USB 标准指令和 class/vender 指令的处理.....	430		
USB 线的连接 / 断开.....	414		
W			
WOVI.....	287		

修订记录	H8S/2218 群、 H8S/2212 群 硬件手册
------	-----------------------------

Rev.	发行日	修订内容	
		页	修订处
1.00	2007.09.27	一	初版发行

**瑞萨 16 位单片机
硬件手册
H8S/2218 群、H8S/2212 群**

Publication Date: 1st Edition, Sep. 27, 2007
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan



RENESAS SALES OFFICES

<http://www.renesas.com>

Refer to "<http://www.renesas.com/en/network>" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510



H8S/2218群、H8S/2212群



瑞萨电子株式会社