

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.

# SH7080 群

瑞萨 32 位 RISC 单片机

SuperH™ RISC engine 族

SH7083	R5F7083 R5M7083 R5S7083
SH7084	R5F7084 R5M7084 R5S7084
SH7085	R5F7085 R5M7085 R5S7085
SH7086	R5F7086

## 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（切除患部、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

本文对适用于微型计算机所有产品的“使用上的注意事项”进行说明。有关个别的使用上的注意事项请参照本文。此外，如果在记载上有与本说明书的本文有差异之处，以本文的记载为标准。

### 1. 未使用的引角的处理

**【注意】**将未使用的引角按照本文的“未使用引角的处理”进行处理。

CMOS产品的输入引角一般为高阻抗输入。如果将未使用的输入引角处于开放状态运行，由于感应现象，外加LSI周围的噪声，在LSI内部产生穿透电流，被误认为是输入信号，引起误动作。未使用的引角，按照本文的“未使用引角的处理”中说明的指示进行处理。

### 2. 通电时的处理

**【注意】**通电时产品处于不定状态。

通电时，LSI内部电路处于不确定状态，寄存器的设定和各引角的状态不定。通过外部复位引角对产品复位时，从通电到复位有效期间，不能保证引角的状态。

同样，使用内置加电复位功能对产品进行复位时，从通电到达到复位产生的一定电压的期间，不能保证引角的状态。

### 3. 禁止存取未定义的地址

**【注意】**禁止存取未定义的地址

在地址区域中，有被分配将来用作功能扩展的未定义地址。因为无法保证存取这些地址时的运行，所以请不要进行存取。

### 4. 关于时钟

**【注意】**复位时，时钟稳定后请解除复位。

在程序运行中切换时钟时，请在要切换成的时钟稳定之后进行。复位时，在通过使用外部阵荡器（或者外部阵荡电路）的时钟开始运行的系统中，在时钟充分稳定后，解除复位。另外，在程序运行中，切换成使用外部阵荡器（或者外部震荡电路）的时钟时，在要切换成的时钟充分稳定后进行。

### 5. 关于产品间的差异

**【注意】**在变更不同型号的产品时，请事先确认是否没问题。

即使是同一个群中的微型计算机，如果产品型号不同，由于内部存储器、版图模式等不同，有可能特性也不同。因此，在变更不同型号的产品时，请对每一个型号的产品进行系统评价测试。

# 本手册的使用方法

## 1 目的和对象

本手册是一本帮助用户理解本单片机的硬件功能和电特性的手册。它以使用本手册设计应用系统的用户为对象。在使用本手册时，需要具备电路、逻辑电路以及单片机的基础知识。

本手册由产品概要、CPU、系统控制功能、外围功能、电特性、使用上的注意事项几大部分组成。

必须在充分确认过注意事项后使用本单片机。注意事项记录在各章的正文中、各章的最后和注意事项章中。

修订记录归纳了对旧版本记载内容的更正或追加的主要位置。并不是修订内容的全部记载。详情请确认本手册的正文。

H8 群准备了以下的文献。请使用最新的文献。最新版本刊登在瑞萨科技的主页上。

文献的种类	记载内容	资料名	资料号
数据表	硬件的概要和电特性	H8 群数据表	RJJ03B0120
硬件手册	硬件的说明（引脚配置、存储器映像、外围功能的说明、电特性、时序）和工作说明 ※外围功能的使用方法必须参照应用注意事项。	H8 群硬件手册	本硬件手册
软件手册	CPU 指令设定的说明	H8 系列软件说明	RJJ09B0002
应用注意事项	外围功能的使用方法、应用例子 参考程序 应用汇编语言、C 语言编成方法	刊登在瑞萨科技的主页上	
RENESAS TECHNICAL UPDATE	相关产品说明、文献等的快速公告		

## 2. Description of Numbers and Symbols

Aspects of the notations for register names, bit names, numbers, and symbolic names in this manual are explained below.

(1) Overall notation

In descriptions involving the names of bits and bit fields within this manual, the modules and registers to which the bits belong may be clarified by giving the names in the forms "module name"."register name"."bit name" or "register name"."bit name".

(2) Register notation

The style "register name"\_"instance number" is used in cases where there is more than one instance of the same function or similar functions.

[Example] CMCSR\_0: Indicates the CMCSR register for the compare-match timer of channel 0.

(3) Number notation

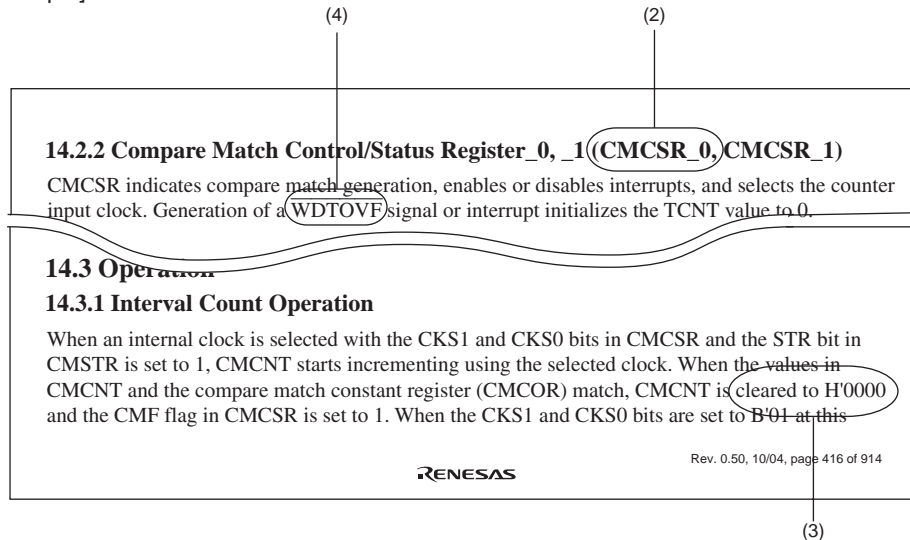
Binary numbers are given as B'nnnn (B' may be omitted if the number is obviously binary), hexadecimal numbers are given as H'nnnn or 0xnnnn, and decimal numbers are given as nnnn.

[Examples] Binary: B'11 or 11  
Hexadecimal: H'EFA0 or 0xEFA0  
Decimal: 1234

(4) Notation for active-low

An overbar on the name indicates that a signal or pin is active-low.

[Example]  $\overline{\text{WDTOVF}}$

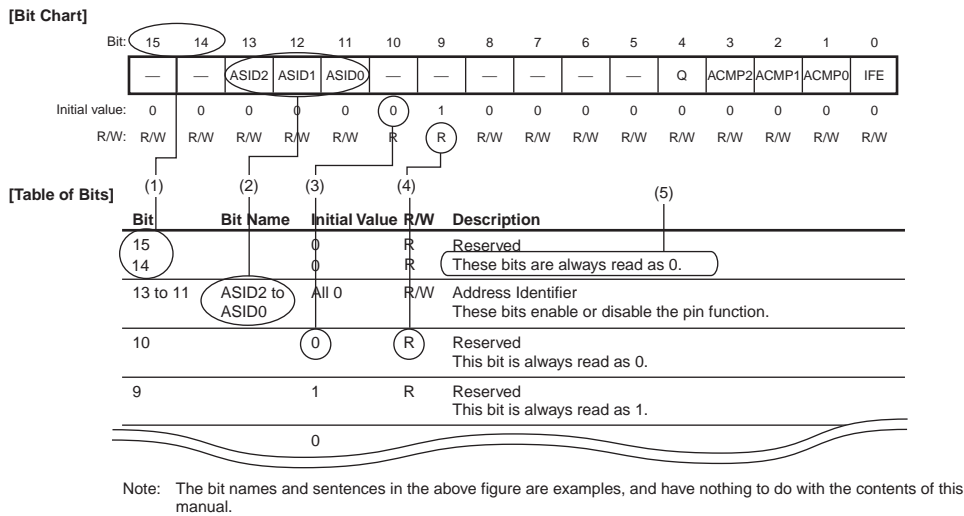


Note: The bit names and sentences in the above figure are examples and have nothing to do with the contents of this manual.



### 3. Description of Registers

Each register description includes a bit chart, illustrating the arrangement of bits, and a table of bits, describing the meanings of the bit settings. The standard format and notation for bit charts and tables are described below.



- (1) Bit  
Indicates the bit number or numbers.  
In the case of a 32-bit register, the bits are arranged in order from 31 to 0. In the case of a 16-bit register, the bits are arranged in order from 15 to 0.
- (2) Bit name  
Indicates the name of the bit or bit field.  
When the number of bits has to be clearly indicated in the field, appropriate notation is included (e.g., ASID[3:0]).  
A reserved bit is indicated by " ".  
Certain kinds of bits, such as those of timer counters, are not assigned bit names. In such cases, the entry under Bit Name is blank.
- (3) Initial value  
Indicates the value of each bit immediately after a power-on reset, i.e., the initial value.  
0: The initial value is 0  
1: The initial value is 1  
 : The initial value is undefined
- (4) R/W  
For each bit and bit field, this entry indicates whether the bit or field is readable or writable, or both writing to and reading from the bit or field are impossible.  
The notation is as follows:  
R/W: The bit or field is readable and writable.  
R/(W): The bit or field is readable and writable.  
However, writing is only performed to flag clearing.  
R: The bit or field is readable.  
"R" is indicated for all reserved bits. When writing to the register, write the value under Initial Value in the bit chart to reserved bits or fields.  
W: The bit or field is writable.
- (5) Description  
Describes the function of the bit or field and specifies the values for writing.

#### 4. 省略语及简称的说明

省略 / 简称	全称	备注
ACIA	Asynchronous Communication Interface Adapter	异步式通信适配器
bps	bits per second	传送速度单位
CRC	Cyclic Redundancy Check	周期冗余校验
DMA	Direct Memory Access	
DMAC	Direct Memory Access Controller	
GSM	Global System for Mobile Communications	
Hi-Z	High Impedance	
IEBus	Inter Equipment bus	NEC 电子公司提倡的通信方式
I/O	Input/Output	输入 / 输出
IrDA	Infrared Data Association	红外线数据协会
LSB	Least Significant Bit	最低有效位
MSB	Most Significant Bit	最高有效位
NC	Non-Connection	未连接引脚
PLL	Phase Locked Loop	锁相环路
PWM	Pulse Width Modulation	脉冲宽度调制
SFR	Special Function Registers	外围电路控制寄存器组
SIM	Subscriber Identity Module	ISO-7816 规定的 IC 卡
UART	Universal Asynchronous Receiver/Transmitter	异步串行接口
VCO	Voltage Controlled Oscillator	电压控制振荡器

## 前言

本 LSI 是以高速 H8S/2000 CPU 为核心，集成了系统构成所必需的外围功能的单片机。

它包括系统构成必须的外围功能，如 ROM、RAM、16 位定时器、看门狗定时器、串行通信接口、CAN、A/D 转换器、PWM 控制器、LCD 控制器 / 驱动器、时钟振荡电路、I/O 端口等，可作为高等级控制系统的嵌入式单片机使用。片上 ROM 带有单一电源闪存 (F-ZTAT™)\*，可随时编程，迅速且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。

**【注】** \* F-ZTAT 是瑞萨科技 ( 有限公司 ) 的商标。

**对象** 本手册以 “在应用系统设计中采用 H8S/2282 群及 H8S/2280 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 “能让用户理解 H8S/2282 群、H8S/2280 群的硬件功能和电特性” 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》。

### 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 CPU、系统控制功能、外围功能、电特性等分类依次构成。
- 希望了解 CPU 功能的详细内容时  
→ 请参考 《H8S/2600 系列、H8S/2000 系列软件手册》。
- 若知道寄存器名称，而希望了解详细功能时  
→ 在本书后面附有 “索引”。请从索引搜索页号。  
《第 27 章 寄存器一览》有关地址、位内容及初始化值，汇总在 《第 27 章 寄存器一览》。

**范例** 寄存器名称 : 多个通道带有串行通信接口等相同或类似的功能，  
将采用以下的标记方法。

XXX\_N (XXX 是基本寄存器的名称、N 是通道编号)

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、10 进制为 xxxx

信号的标记 : 低电平有效的信号带有上划线 xxx。

# 目 录

第 1 章 概要 .....	1
1.1 SH7083/84/85/86 的特点 .....	1
1.2 框图 .....	6
1.3 引脚排列图 .....	7
1.4 引脚功能 .....	12
第 2 章 CPU .....	18
2.1 特点 .....	18
2.2 寄存器结构 .....	19
2.2.1 通用寄存器 (Rn) .....	20
2.2.2 控制寄存器 .....	20
2.2.3 系统寄存器 .....	21
2.2.4 寄存器的初始值 .....	21
2.3 数据格式 .....	22
2.3.1 寄存器的数据格式 .....	22
2.3.2 存储器的数据格式 .....	22
2.3.3 立即数的数据格式 .....	22
2.4 指令特点 .....	23
2.4.1 RISC 方式 .....	23
2.4.2 寻址方式 .....	25
2.4.3 指令格式 .....	28
2.5 指令系统 .....	30
2.5.1 指令系统的分类 .....	30
2.5.2 数据传送指令 .....	33
2.5.3 算术运算指令 .....	34
2.5.4 逻辑运算指令 .....	35
2.5.5 移位指令 .....	36
2.5.6 转移指令 .....	36
2.5.7 系统控制指令 .....	37
2.6 处理状态 .....	38
第 3 章 MCU 运行模式 .....	40
3.1 运行模式的选择 .....	40
3.2 输入 / 输出引脚 .....	40
3.3 各运行模式的说明 .....	41
3.3.1 模式 0 (MCU 扩展模式 0) .....	41
3.3.2 模式 1 (MCU 扩展模式 1) .....	41
3.3.3 模式 2 (MCU 扩展模式 2) .....	41
3.3.4 模式 3 (单芯片模式) .....	41
3.4 地址映像 .....	42
3.5 本 LSI 的初始状态 .....	49
3.6 更改运行模式时的注意事项 .....	49
第 4 章 时钟振荡器 (CPG) .....	50
4.1 特点 .....	50
4.2 输入 / 输出引脚 .....	53
4.3 时钟运行模式 .....	53
4.4 寄存器说明 .....	57
4.4.1 频率控制寄存器 (FRQCR) .....	57

4.4.2	振荡停止检测控制寄存器 (OSCCR) .....	59
4.5	频率的变更方法 .....	60
4.6	振荡器 .....	60
4.6.1	连接晶体谐振器的方法 .....	60
4.6.2	输入外部时钟的方法 .....	61
4.7	振荡停止检测功能 .....	62
4.8	使用注意事项 .....	62
4.8.1	有关谐振器的注意事项 .....	62
4.8.2	电路板设计的注意事项 .....	62
<b>第 5 章</b>	<b>异常处理 .....</b>	<b>64</b>
5.1	概要 .....	64
5.1.1	异常处理的种类和优先级 .....	64
5.1.2	异常处理的运行 .....	65
5.1.3	异常处理向量表 .....	65
5.2	复位 .....	67
5.2.1	复位的种类 .....	67
5.2.2	上电复位 .....	67
5.2.3	手动复位 .....	68
5.3	地址错误 .....	69
5.3.1	地址错误的发生源 .....	69
5.3.2	地址错误的异常处理 .....	69
5.4	中断 .....	70
5.4.1	中断源 .....	70
5.4.2	中断优先级 .....	71
5.4.3	中断异常处理 .....	71
5.5	由指令引起的异常 .....	71
5.5.1	由指令引起的异常种类 .....	71
5.5.2	陷阱指令 .....	72
5.5.3	槽非法指令 .....	72
5.5.4	一般非法指令 .....	72
5.6	异常处理的接受 .....	72
5.7	异常处理后的堆栈状态 .....	73
5.8	使用注意事项 .....	73
5.8.1	堆栈指针 (SP) 的值 .....	73
5.8.2	向量基址寄存器 (VBR) 的值 .....	73
5.8.3	在地址错误异常处理的堆栈存取中发生的地址错误 .....	73
5.8.4	有关槽非法指令异常处理的注意事项 .....	74
<b>第 6 章</b>	<b>中断控制器 (INTC) .....</b>	<b>75</b>
6.1	特点 .....	75
6.2	输入 / 输出引脚 .....	76
6.3	寄存器说明 .....	76
6.3.1	中断控制寄存器 0 (ICR0) .....	77
6.3.2	IRQ 控制寄存器 (IRQCR) .....	77
6.3.3	IRQ 状态寄存器 (IRQSR) .....	79
6.3.4	中断优先级寄存器 A ~ F、H ~ M (IPRA ~ IPRF、IPRH ~ IPRM) .....	84
6.4	中断源 .....	85
6.4.1	外部中断源 .....	85
6.4.2	内部外围模块中断 .....	86
6.4.3	用户断点中断 .....	86
6.5	中断异常处理向量表 .....	86

6.6	运行说明	89
6.6.1	中断运行的流程	89
6.6.2	中断异常处理结束后的堆栈状态	91
6.7	中断响应时间	92
6.8	通过中断请求信号的数据传送	93
6.8.1	中断请求信号作为 DTC 启动源和 CPU 中断源而不作为 DMAC 启动源的情况	94
6.8.2	中断请求信号作为 DMAC 启动源而不作为 CPU 中断源和 DTC 启动源的情况	94
6.8.3	中断请求信号作为 DTC 启动源而不作为 CPU 中断源和 DMAC 启动源的情况	94
6.8.4	中断请求信号作为 CPU 中断源而不作为 DTC 启动源和 DMAC 启动源的情况	95
6.9	使用注意事项	95
<b>第 7 章</b>	<b>用户断点控制器 (UBC)</b>	<b>96</b>
7.1	特点	96
7.2	输入 / 输出引脚	98
7.3	寄存器说明	98
7.3.1	断点地址寄存器 A (BARA)	99
7.3.2	断点地址屏蔽寄存器 A (BAMRA)	99
7.3.3	断点总线周期寄存器 A (BBRA)	100
7.3.4	断点数据寄存器 A (BDRA) (只限 F-ZTAT 版)	101
7.3.5	断点数据屏蔽寄存器 A (BDMRA) (只限 F-ZTAT 版)	101
7.3.6	断点地址寄存器 B (BARB)	102
7.3.7	断点地址屏蔽寄存器 B (BAMRB)	102
7.3.8	断点数据寄存器 B (BDRB) (只限 F-ZTAT 版)	103
7.3.9	断点数据屏蔽寄存器 B (BDMRB) (只限 F-ZTAT 版)	103
7.3.10	断点总线周期寄存器 B (BBRB)	104
7.3.11	断点控制寄存器 (BRCR)	105
7.3.12	执行次数断点寄存器 (BETR) (只限 F-ZTAT 版)	107
7.3.13	转移源寄存器 (BRSR) (只限 F-ZTAT 版)	108
7.3.14	转移目标寄存器 (BRDR) (只限 F-ZTAT 版)	109
7.4	运行说明	109
7.4.1	用户断点运行的流程	109
7.4.2	取指令周期的用户断点	110
7.4.3	数据存取周期的用户断点	111
7.4.4	顺序断点	111
7.4.5	被保存的程序计数器值	112
7.4.6	PC 跟踪	112
7.4.7	使用例	113
7.5	使用注意事项	117
<b>第 8 章</b>	<b>数据传送控制器 (DTC)</b>	<b>118</b>
8.1	特点	118
8.2	寄存器说明	120
8.2.1	DTC 模式寄存器 A (MRA)	121
8.2.2	DTC 模式寄存器 B (MRB)	122
8.2.3	DTC 源地址寄存器 (SAR)	123
8.2.4	DTC 目标地址寄存器 (DAR)	123
8.2.5	DTC 传送计数寄存器 A (CRA)	123
8.2.6	DTC 传送计数寄存器 B (CRB)	124
8.2.7	DTC 允许寄存器 A ~ E (DTCERA ~ DTCERE)	124
8.2.8	DTC 控制寄存器 (DTCCR)	125
8.2.9	DTC 向量基址寄存器 (DTCVBR)	126
8.2.10	总线功能扩展寄存器 (BSCEHR)	126

8.3	启动源 .....	126
8.4	传送信息的分配和 DTC 向量表 .....	127
8.5	运行说明 .....	130
8.5.1	传送信息的跳读功能 .....	133
8.5.2	传送信息的回写省略功能 .....	134
8.5.3	正常传送模式 .....	134
8.5.4	重复传送模式 .....	135
8.5.5	块传送模式 .....	136
8.5.6	链传送 .....	137
8.5.7	运行时序 .....	137
8.5.8	DTC 的执行状态 .....	139
8.5.9	DTC 的总线权释放时序 .....	141
8.5.10	DTC 启动优先级的设定 .....	143
8.6	通过中断启动 DTC .....	144
8.7	DTC 使用例 .....	145
8.7.1	正常传送 .....	145
8.7.2	计数器 =0 的链传送 .....	145
8.8	中断源 .....	146
8.9	使用注意事项 .....	146
8.9.1	模块待机模式的设定 .....	146
8.9.2	内部 RAM .....	146
8.9.3	DTCE 位的设定 .....	146
8.9.4	链传送 .....	146
8.9.5	传送信息起始地址 / 源地址 / 目标地址 .....	146
8.9.6	通过 DTC 存取 DMAC/DTC 寄存器 .....	147
8.9.7	IRQ 中断作为 DTC 传送源时的注意事项 .....	147
8.9.8	SCI 和 SCIF 作为 DTC 启动源时的注意事项 .....	147
8.9.9	中断源标志的清除 .....	147
8.9.10	NMI 中断和 DTC 启动的竞争 .....	147
8.9.11	DTC 启动请求在中途被取消时的运行 .....	147
<b>第 9 章</b>	<b>总线状态控制器 (BSC) .....</b>	<b>148</b>
9.1	特点 .....	148
9.2	输入 / 输出引脚 .....	150
9.3	区域概要 .....	151
9.3.1	空间分割 .....	151
9.3.2	地址映像 .....	151
9.4	寄存器说明 .....	165
9.4.1	通用控制寄存器 (CMNCR) .....	166
9.4.2	CSn 空间总线控制寄存器 (CSnBCR) (n=0 ~ 8) .....	167
9.4.3	CSn 空间等待控制寄存器 (CSnWCR) (n=0 ~ 8) .....	170
9.4.4	SDRAM 控制寄存器 (SDCR) .....	184
9.4.5	刷新定时器的控制 / 状态寄存器 (RTCSR) .....	186
9.4.6	刷新定时器的计数器 (RTCNT) .....	187
9.4.7	刷新时间常数寄存器 (RTCOR) .....	188
9.4.8	总线功能扩展寄存器 (BSCEHR) .....	189
9.5	运行说明 .....	192
9.5.1	端法 / 存取长度和数据定位 .....	192
9.5.2	通常空间接口 .....	195
9.5.3	存取等待的控制 .....	200
9.5.4	CSn 有效期间的扩展 .....	202
9.5.5	MPX-I/O 接口 .....	203

9.5.6	SDRAM 接口 .....	205
9.5.7	突发 ROM (异步) 接口 .....	226
9.5.8	带字节选择的 SRAM 接口 .....	228
9.5.9	PCMCIA 接口 .....	232
9.5.10	突发 MPX-I/O 接口 .....	236
9.5.11	突发 ROM (时钟同步) 接口 .....	239
9.5.12	存取周期之间的等待 .....	240
9.5.13	总线仲裁 .....	248
9.5.14	其他 .....	252
9.5.15	从 CPU 向内部 FLASH 和内部 RAM 的存取 .....	252
9.5.16	从 CPU 向内部外围 I/O 寄存器的存取 .....	253
9.5.17	从 CPU 向外部存储器的存取 .....	254
<b>第 10 章</b>	<b>直接存储器存取控制器 (DMAC) .....</b>	<b>256</b>
10.1	特点 .....	256
10.2	输入 / 输出引脚 .....	258
10.3	寄存器说明 .....	258
10.3.1	DMA 源地址寄存器_0 ~ 3 (SAR_0 ~ 3) .....	259
10.3.2	DMA 目标地址寄存器_0 ~ 3 (DAR_0 ~ 3) .....	259
10.3.3	DMA 传送计数寄存器_0 ~ 3 (DMATCR_0 ~ 3) .....	260
10.3.4	DMA 通道控制寄存器_0 ~ 3 (CHCR_0 ~ 3) .....	260
10.3.5	DMA 操作寄存器 (DMAOR) .....	264
10.3.6	总线功能扩展寄存器 (BSCEHR) .....	265
10.4	运行说明 .....	266
10.4.1	传送流程 .....	266
10.4.2	DMA 传送请求 .....	268
10.4.3	通道优先级 .....	270
10.4.4	DMA 的传送种类 .....	274
10.4.5	总线周期的状态数和 DREQ 引脚的采样时序 .....	282
10.4.6	运行时序 .....	285
10.5	使用注意事项 .....	286
10.5.1	DACK 引脚输出的注意事项 .....	286
10.5.2	外围模块的 DMA 传送 .....	286
10.5.3	模块待机模式的设定 .....	286
10.5.4	通过 DMAC 存取 DMAC/DTC 寄存器 .....	286
10.5.5	SCI 作为 DMAC 启动源的注意事项 .....	286
10.5.6	CHCR 的设定 .....	286
10.5.7	启动多个通道时的注意事项 .....	286
10.5.8	输入传送请求时的注意事项 .....	286
10.5.9	NMI 中断和 DMAC 启动的竞争 .....	287
10.5.10	DMAC 存取内部 RAM 的周期数 .....	287
10.5.11	以 MTU2 为启动源的突发模式中的 DMAC 传送的注意事项 .....	287
10.5.12	总线功能扩展寄存器 (BSCEHR) .....	287
<b>第 11 章</b>	<b>多功能定时器脉冲单元 2 (MTU2) .....</b>	<b>288</b>
11.1	特点 .....	288
11.2	输入 / 输出引脚 .....	292
11.3	寄存器的说明 .....	293
11.3.1	定时器的控制寄存器 (TCR) .....	296
11.3.2	定时器的模式寄存器 (TMDR) .....	299
11.3.3	定时器的 I/O 控制寄存器 (TIOR) .....	301
11.3.4	定时器的比较匹配清除寄存器 (TCNTCMPCLR) .....	319



11.3.5	定时器的中断允许寄存器 (TIER)	320
11.3.6	定时器的状态寄存器 (TSR)	323
11.3.7	定时器的缓冲运行传送模式寄存器 (TBTM)	327
11.3.8	定时器的输入捕捉控制寄存器 (TICCR)	328
11.3.9	定时器的同步清除寄存器 (TSYCR)	329
11.3.10	定时器的 A/D 转换开始请求的控制寄存器 (TADCR)	330
11.3.11	定时器的 A/D 转换开始请求的周期设定寄存器 (TADCORA/B_4)	332
11.3.12	定时器的 A/D 转换开始请求的周期设定缓冲寄存器 (TADCOBRA/B_4)	332
11.3.13	定时器的计数器 (TCNT)	332
11.3.14	定时器的通用寄存器 (TGR)	333
11.3.15	定时器的开始寄存器 (TSTR)	333
11.3.16	定时器的同步寄存器 (TSYR)	335
11.3.17	定时器的计数器同步开始寄存器 (TCSYSTR)	336
11.3.18	定时器的读写允许寄存器 (TRWER)	337
11.3.19	定时器的输出主控允许寄存器 (TOER)	338
11.3.20	定时器的输出控制寄存器 1 (TOCR1)	339
11.3.21	定时器的输出控制寄存器 2 (TOCR2)	341
11.3.22	定时器的输出电平缓冲寄存器 (TOLBR)	344
11.3.23	定时器的门控寄存器 (TGCR)	345
11.3.24	定时器的子计数器 (TCNTS)	346
11.3.25	定时器的空载时间数据寄存器 (TDDR)	346
11.3.26	定时器的周期数据寄存器 (TCDR)	347
11.3.27	定时器的周期缓冲寄存器 (TCBR)	347
11.3.28	定时器的跳过中断设定寄存器 (TITCR)	347
11.3.29	定时器的跳过中断次数计数器 (TITCNT)	349
11.3.30	定时器的缓冲传送设定寄存器 (TBTER)	350
11.3.31	定时器的空载时间允许寄存器 (TDER)	351
11.3.32	定时器的波形控制寄存器 (TWCR)	351
11.3.33	和总线主控的接口	353
11.4	运行说明	353
11.4.1	基本运行	353
11.4.2	同步运行	358
11.4.3	缓冲运行	359
11.4.4	级联运行	363
11.4.5	PWM 模式	368
11.4.6	相位计数模式	373
11.4.7	复位同步 PWM 模式	379
11.4.8	互补 PWM 模式	381
11.4.9	A/D 转换开始请求的延迟功能	411
11.4.10	MTU2-MTU2S 的同步运行	414
11.4.11	外部脉宽的测量功能	418
11.4.12	空载时间的补偿功能	419
11.4.13	互补 PWM 在“波峰/波谷”的 TCNT 捕捉运行	420
11.5	中断源	421
11.5.1	中断源和优先级	421
11.5.2	DTC/DMAC 的启动	423
11.5.3	A/D 转换器的启动	423
11.6	运行时序	425
11.6.1	输入/输出时序	425
11.6.2	中断信号的时序	431
11.7	使用注意事项	436
11.7.1	模块待机模式的设定	436

11.7.2	输入时钟的限制事项 .....	436
11.7.3	周期设定的注意事项 .....	436
11.7.4	TCNT 的写操作和清除的竞争 .....	437
11.7.5	TCNT 的写操作和递增计数的竞争 .....	437
11.7.6	TGR 的写操作和比较匹配的竞争 .....	438
11.7.7	缓冲寄存器的写操作和比较匹配的竞争 .....	438
11.7.8	缓冲寄存器的写操作和 TCNT 清除的竞争 .....	439
11.7.9	TGR 的读操作和输入捕捉的竞争 .....	440
11.7.10	TGR 的写操作和输入捕捉的竞争 .....	441
11.7.11	缓冲寄存器的写操作和输入捕捉的竞争 .....	442
11.7.12	级联中 TCNT_2 的写操作和上溢 / 下溢的竞争 .....	443
11.7.13	互补 PWM 模式停止时的计数器值 .....	444
11.7.14	互补 PWM 模式中缓冲运行的设定 .....	444
11.7.15	复位同步 PWM 模式的缓冲运行和比较匹配标志 .....	445
11.7.16	复位同步 PWM 模式的上溢标志 .....	446
11.7.17	上溢 / 下溢和计数器清除的竞争 .....	446
11.7.18	TCNT 的写操作和上溢 / 下溢的竞争 .....	447
11.7.19	从通常运行或者 PWM 模式 1 转移到复位同步 PWM 模式时的注意事项 .....	447
11.7.20	互补 PWM 模式或者复位同步 PWM 模式的输出电平 .....	447
11.7.21	模块待机时的中断 .....	447
11.7.22	级联中 TCNT_1 和 TCNT_2 的同时输入捕捉 .....	448
11.8	MTU2 输出引脚的初始化方法 .....	449
11.8.1	运行模式 .....	449
11.8.2	开始复位时的运行 .....	449
11.8.3	因运行中的异常等而重新设定时的运行 .....	449
11.8.4	因运行中的异常等而初始化引脚的步骤和模式转移的概要 .....	450
<b>第 12 章</b>	<b>多功能定时器脉冲单元 2S (MTU2S) .....</b>	<b>480</b>
12.1	输入 / 输出引脚 .....	481
12.2	寄存器说明 .....	482
<b>第 13 章</b>	<b>端口输出的允许 (POE) .....</b>	<b>484</b>
13.1	特点 .....	484
13.2	输入 / 输出引脚 .....	486
13.3	寄存器说明 .....	487
13.3.1	输入电平控制 / 状态寄存器 1 (ICSR1) .....	487
13.3.2	输出电平控制 / 状态寄存器 1 (OCSR1) .....	490
13.3.3	输入电平控制 / 状态寄存器 2 (ICSR2) .....	491
13.3.4	输出电平控制 / 状态寄存器 2 (OCSR2) .....	493
13.3.5	输入电平控制 / 状态寄存器 3 (ICSR3) .....	494
13.3.6	软件端口的输出允许寄存器 (SPOER) .....	495
13.3.7	端口的输出允许控制寄存器 1 (POECR1) .....	496
13.3.8	端口的输出允许控制寄存器 2 (POECR2) .....	497
13.4	运行说明 .....	500
13.4.1	输入电平的检测 .....	501
13.4.2	输出电平的比较 .....	502
13.4.3	高阻抗状态的解除 .....	502
13.5	中断 .....	503
13.6	使用注意事项 .....	503
13.6.1	从看门狗定时器产生上电复位时的引脚状态 .....	503
<b>第 14 章</b>	<b>看门狗定时器 (WDT) .....</b>	<b>504</b>

14.1	特点	504
14.2	输入 / 输出引脚	505
14.3	寄存器说明	505
14.3.1	看门狗定时器的计数器 (WTCNT)	505
14.3.2	看门狗定时器的控制 / 状态寄存器 (WTCSR)	506
14.3.3	寄存器存取时的注意事项	507
14.4	运行说明	508
14.4.1	软件待机模式的解除步骤	508
14.4.2	看门狗定时器模式的用法	508
14.4.3	间隔定时器模式的用法	509
14.5	使用注意事项	509
14.5.1	WTCNT 的设定值	509
<b>第 15 章 串行通信接口 (SCI)</b>		<b>510</b>
15.1	特点	510
15.2	输入 / 输出引脚	511
15.3	寄存器说明	512
15.3.1	接收移位寄存器 (SCRSR)	513
15.3.2	接收数据寄存器 (SCRDR)	513
15.3.3	发送移位寄存器 (SCTSR)	513
15.3.4	发送数据寄存器 (SCTDR)	514
15.3.5	串行模式寄存器 (SCSMR)	514
15.3.6	串行控制寄存器 (SCSCR)	516
15.3.7	串行状态寄存器 (SCSSR)	518
15.3.8	串行端口寄存器 (SCSPTR)	520
15.3.9	串行方向控制寄存器 (SCSDCR)	522
15.3.10	位速率寄存器 (SCBRR)	523
15.4	运行说明	530
15.4.1	概要	530
15.4.2	异步模式中的运行	532
15.4.3	时钟同步模式中的运行	540
15.4.4	多处理器通信功能	547
15.4.5	多处理器串行数据的发送	548
15.4.6	多处理器串行数据的接收	549
15.5	SCI 的中断源和 DMAC/DTC	552
15.6	串行端口寄存器 (SCSPTR) 和 SCI 引脚的关系	553
15.7	使用注意事项	554
15.7.1	有关 SCTDR 的写操作和 TDRE 标志的关系	554
15.7.2	有关同时发生多个接收错误时的运行	554
15.7.3	有关中止的检测和处理	554
15.7.4	中止的发送	554
15.7.5	异步模式的接收数据采样时序和接收容限	555
15.7.6	DMAC/DTC 的使用注意事项	556
15.7.7	时钟同步外部时钟模式的注意事项	556
15.7.8	模块待机模式的设定	556
<b>第 16 章 带 FIFO 的串行通信接口 (SCIF)</b>		<b>557</b>
16.1	特点	557
16.2	输入 / 输出引脚	558
16.3	寄存器说明	559
16.3.1	接收移位寄存器 (SCRSR)	559
16.3.2	接收 FIFO 数据寄存器 (SCFRDR)	559

16.3.3	发送移位寄存器 (SCTSR)	560
16.3.4	发送 FIFO 数据寄存器 (SCFTDR)	560
16.3.5	串行模式寄存器 (SCSMR)	561
16.3.6	串行控制寄存器 (SCSCR)	562
16.3.7	串行状态寄存器 (SCFSR)	564
16.3.8	位速率寄存器 (SCBRR)	568
16.3.9	FIFO 控制寄存器 (SCFCR)	576
16.3.10	FIFO 数据个数寄存器 (SCFDR)	577
16.3.11	串行端口寄存器 (SCSPTR)	578
16.3.12	线路状态寄存器 (SCLSR)	580
16.4	运行说明	581
16.4.1	概要	581
16.4.2	异步模式中的运行	582
16.4.3	时钟同步模式中的运行	590
16.5	SCIF 的中断源和 DTC	597
16.6	串行端口寄存器 (SCSPTR) 和 SCIF 引脚的关系	598
16.7	使用注意事项	600
16.7.1	有关 SCFTDR 的写操作和 TDFE 标志	600
16.7.2	有关 SCFRDR 的读操作和 RDF 标志	600
16.7.3	有关中止的检测和处理	600
16.7.4	中止的发送	600
16.7.5	异步模式的接收数据采样时序和接收容限	600
16.7.6	模块待机模式的设定	601
16.7.7	DTC 使用注意事项	601
16.7.8	关于串行状态寄存器 (SCFSR) 的 FER 标志核 PER 标志	601
<b>第 17 章</b>	<b>同步串行通信单元 (SSU)</b>	<b>602</b>
17.1	特点	602
17.2	输入 / 输出引脚	604
17.3	寄存器说明	604
17.3.1	SS 控制寄存器 H (SSCRH)	605
17.3.2	SS 控制寄存器 L (SSCRL)	606
17.3.3	SS 模式寄存器 (SSMR)	607
17.3.4	SS 允许寄存器 (SSER)	608
17.3.5	SS 状态寄存器 (SSSR)	609
17.3.6	SS 控制寄存器 2 (SSCR2)	611
17.3.7	SS 发送数据寄存器 0 ~ 3 (SSTDR0 ~ SSTDR3)	612
17.3.8	SS 接收数据寄存器 0 ~ 3 (SSRDR0 ~ SSRDR3)	613
17.3.9	SS 移位寄存器 (SSTRSR)	613
17.4	运行说明	614
17.4.1	传送时钟	614
17.4.2	时钟的相位、极性和数据的关系	614
17.4.3	数据输入 / 输出引脚和移位寄存器的关系	615
17.4.4	各通信模式和引脚功能	616
17.4.5	SSU 模式	618
17.4.6	SCS 引脚控制和冲突错误	624
17.4.7	时钟同步通信模式	625
17.5	SSU 的中断源和 DTC	629
17.6	使用注意事项	629
17.6.1	模块待机模式的设定	629
17.6.2	SSTDR 和 SSRDR 寄存器的存取	629
17.6.3	SSU 从属模式中的连续发送和接收时的注意事项	629

第 18 章 I <sup>2</sup> C 总线接口 2 (I <sup>2</sup> C2)	630
18.1 特点	630
18.2 输入 / 输出引脚	632
18.3 寄存器说明	632
18.3.1 I <sup>2</sup> C 总线控制寄存器 1 (ICCR1)	633
18.3.2 I <sup>2</sup> C 总线控制寄存器 2 (ICCR2)	635
18.3.3 I <sup>2</sup> C 总线模式寄存器 (ICMR)	636
18.3.4 I <sup>2</sup> C 总线中断允许寄存器 (ICIER)	637
18.3.5 I <sup>2</sup> C 总线状态寄存器 (ICSR)	638
18.3.6 从属地址寄存器 (SAR)	640
18.3.7 I <sup>2</sup> C 总线发送数据寄存器 (ICDRT)	640
18.3.8 I <sup>2</sup> C 总线接收数据寄存器 (ICDRR)	640
18.3.9 I <sup>2</sup> C 总线移位寄存器 (ICDRS)	641
18.3.10 NF2CYC 寄存器 (NF2CYC)	641
18.4 运行说明	642
18.4.1 I <sup>2</sup> C 总线格式	642
18.4.2 主发送	643
18.4.3 主接收	644
18.4.4 从属发送	646
18.4.5 从属接收	647
18.4.6 时钟同步串行格式	649
18.4.7 噪声消除电路	651
18.4.8 使用例	652
18.5 I <sup>2</sup> C2 的中断源	656
18.6 DTC 的运行	656
18.7 位同步电路	657
18.8 使用注意事项	658
18.8.1 模块待机模式的设定	658
18.8.2 停止条件和开始条件 (重新发送) 的产生	658
18.8.3 开始条件和停止条件的连续产生	658
18.8.4 关于作为多主使用时的设定	658
18.8.5 在主接收模式下读取 ICDRR	658
第 19 章 A/D 转换器 (ADC)	659
19.1 特点	659
19.2 输入 / 输出引脚	661
19.3 寄存器说明	662
19.3.1 A/D 数据寄存器 0 ~ 15 (ADDR0 ~ ADDR15)	663
19.3.2 A/D 控制 / 状态寄存器 _0 ~ 2 (ADCSR_0 ~ 2)	663
19.3.3 A/D 控制寄存器 _0 ~ 2 (ADCR_0 ~ 2)	665
19.3.4 A/D 触发选择寄存器 _0、1 (ADTSR_0、1)	668
19.4 运行说明	672
19.4.1 单通道模式	672
19.4.2 连续扫描模式	672
19.4.3 单周期扫描模式	672
19.4.4 输入采样和 A/D 转换时间	673
19.4.5 通过 MTU2 启动 A/D 转换器	674
19.4.6 外部触发的输入时序	675
19.4.7 2 通道扫描	676
19.5 中断源和 DMAC/DTC 传送请求	676
19.6 A/D 转换精度的定义	677
19.7 使用注意事项	679

19.7.1	模块待机模式的设定 .....	679
19.7.2	关于容许信号源阻抗 .....	679
19.7.3	关于对绝对精度的影响 .....	679
19.7.4	模拟电源引脚等的设定范围 .....	679
19.7.5	电路板设计的注意事项 .....	680
19.7.6	噪声对策的注意事项 .....	680
<b>第 20 章</b>	<b>比较匹配定时器 (CMT) .....</b>	<b>681</b>
20.1	特点 .....	681
20.2	寄存器说明 .....	682
20.2.1	比较匹配定时器的开始寄存器 (CMSTR) .....	682
20.2.2	比较匹配定时器的控制/状态寄存器 (CMCSR) .....	683
20.2.3	比较匹配计数器 (CMCNT) .....	684
20.2.4	比较匹配常数寄存器 (CMCOR) .....	684
20.3	运行说明 .....	685
20.3.1	周期计数运行 .....	685
20.3.2	CMCNT 的计数时序 .....	685
20.4	中断 .....	686
20.4.1	CMT 的中断源 .....	686
20.4.2	比较匹配标志的置位时序 .....	686
20.4.3	比较匹配标志的清除时序 .....	686
20.5	使用注意事项 .....	687
20.5.1	模块待机模式的设定 .....	687
20.5.2	CMCNT 的写操作和比较匹配的竞争 .....	687
20.5.3	CMCNT 的字写和递增计数的竞争 .....	687
20.5.4	CMCNT 的字节写和递增计数的竞争 .....	688
20.5.5	CMCNT 和 CMCOR 的比较匹配 .....	688
<b>第 21 章</b>	<b>引脚功能控制器 (PFC) .....</b>	<b>689</b>
21.1	寄存器说明 .....	736
21.1.1	端口 A 的 IO 寄存器 L、H (PAIORL、PAIORH) .....	738
21.1.2	端口 A 的控制寄存器 L1 ~ L4、H1 ~ H4 (PACRL1 ~ PACRL4、PACRH1 ~ PACRH4) .....	738
21.1.3	端口 B 的 IO 寄存器 L (PBIORL) .....	763
21.1.4	端口 B 的控制寄存器 L1 ~ L3 (PBCRL1 ~ PBCRL3) .....	763
21.1.5	端口 C 的 IO 寄存器 L、H (PCIORL、PCIORH) .....	769
21.1.6	端口 C 的控制寄存器 L1 ~ L4、H1 ~ H3 (PCCRL1 ~ PCCRL4、PCCRH1 ~ PCCRH3) .....	769
21.1.7	端口 D 的 IO 寄存器 L、H (PDIORL、PDIORH) .....	781
21.1.8	端口 D 的控制寄存器 L1 ~ L4、H1 ~ H4 (PDCRL1 ~ PDCRL4、PDCRH1 ~ PDCRH4) .....	781
21.1.9	端口 E 的 IO 寄存器 L、H (PEIORL、PEIORH) .....	794
21.1.10	端口 E 的控制寄存器 L1 ~ L4、H1、H2 (PECRL1 ~ PECRL4、PECRH1、PECRH2) .....	794
21.1.11	大电流端口的控制寄存器 (HCPCR) .....	815
21.1.12	IRQOUT 功能的控制寄存器 (IFCR) .....	816
21.2	使用注意事项 .....	817
<b>第 22 章</b>	<b>I/O 端口 .....</b>	<b>818</b>
22.1	端口 A .....	819
22.1.1	寄存器说明 .....	822
22.1.2	端口 A 的数据寄存器 H、L (PADRH、PADRL) .....	822

22.1.3	端口 A 的端口寄存器 H、L (PAPRH、PAPRL)	827
22.2	端口 B	832
22.2.1	寄存器说明	833
22.2.2	端口 B 的数据寄存器 L (PBDRL)	833
22.2.3	端口 B 的端口寄存器 L (PBPRL)	835
22.3	端口 C	836
22.3.1	寄存器说明	838
22.3.2	端口 C 的数据寄存器 H、L (PCDRH、PCDRL)	838
22.3.3	端口 C 的端口寄存器 H、L (PCPRH、PCPRL)	841
22.4	端口 D	843
22.4.1	寄存器说明	845
22.4.2	端口 D 的数据寄存器 H、L (PDDRH、PDDRL)	845
22.4.3	端口 D 的端口寄存器 H、L (PDPRH、PDPRL)	848
22.5	端口 E	850
22.5.1	寄存器说明	853
22.5.2	端口 E 的数据寄存器 H、L (PEDRH、PEDRL)	853
22.5.3	端口 E 的端口寄存器 H、L (PEPRH、PEPRL)	857
22.6	端口 F	860
22.6.1	寄存器说明	861
22.6.2	端口 F 的数据寄存器 L (PFDRL)	861
<b>第 23 章</b>	<b>闪存</b>	<b>863</b>
23.1	特点	863
23.2	概要	864
23.2.1	框图	864
23.2.2	运行模式	864
23.2.3	模式比较	866
23.2.4	闪存结构	867
23.2.5	块分割	868
23.2.6	编程 / 擦除的接口	869
23.3	输入 / 输出引脚	870
23.4	寄存器说明	871
23.4.1	寄存器一览表	871
23.4.2	编程 / 擦除的接口寄存器	872
23.4.3	编程 / 擦除的接口参数	877
23.4.4	RAM 仿真寄存器 (RAMER)	887
23.5	板上编程模式	888
23.5.1	引导模式	888
23.5.2	用户编程模式	891
23.5.3	用户引导模式	899
23.6	保护	902
23.6.1	硬件保护	902
23.6.2	软件保护	902
23.6.3	错误保护	903
23.7	通过 RAM 对闪存进行仿真	904
23.8	使用注意事项	906
23.8.1	用户 MAT 和用户引导 MAT 的转换	906
23.8.2	执行编程 / 擦除过程中的中断	907
23.8.3	其他注意事项	908
23.9	附录	909
23.9.1	引导模式的标准串行通信接口规格	909
23.9.2	过程程序或者编程数据的可保存区	930

23.10 编程器模式 .....	934
<b>第 24 章 掩模型 ROM .....</b>	<b>935</b>
24.1 使用注意事项 .....	935
24.1.1 模块待机模式的设定 .....	935
<b>第 25 章 RAM .....</b>	<b>936</b>
25.1 使用注意事项 .....	936
25.1.1 模块待机模式的设定 .....	936
25.1.2 地址错误 .....	936
25.1.3 RAM 的初始值 .....	936
<b>第 26 章 低功耗模式 .....</b>	<b>937</b>
26.1 特点 .....	937
26.1.1 低功耗模式的种类 .....	937
26.2 输入 / 输出引脚 .....	938
26.3 寄存器说明 .....	938
26.3.1 待机控制寄存器 1 (STBCR1) .....	938
26.3.2 待机控制寄存器 2 (STBCR2) .....	939
26.3.3 待机控制寄存器 3 (STBCR3) .....	940
26.3.4 待机控制寄存器 4 (STBCR4) .....	941
26.3.5 待机控制寄存器 5 (STBCR5) .....	942
26.3.6 待机控制寄存器 6 (STBCR6) .....	943
26.3.7 RAM 控制寄存器 (RAMCR) .....	944
26.4 睡眠模式 .....	944
26.4.1 向睡眠模式的转移 .....	944
26.4.2 睡眠模式的解除 .....	944
26.5 软件待机模式 .....	945
26.5.1 向软件待机模式的转移 .....	945
26.5.2 软件待机模式的解除 .....	945
26.6 深度软件待机模式 .....	946
26.6.1 向深度软件待机模式的转移 .....	946
26.6.2 深度软件待机模式的解除 .....	946
26.7 模块待机功能 .....	946
26.7.1 向模块待机功能的转移 .....	946
26.7.2 模块待机功能的解除 .....	946
26.8 使用注意事项 .....	946
26.8.1 振荡稳定待机中的功耗 .....	946
26.8.2 深度软件待机模式 .....	946
26.8.3 执行睡眠指令时 .....	946
<b>第 27 章 寄存器一览 .....</b>	<b>947</b>
27.1 寄存器地址一览表 (地址顺序) .....	948
27.2 寄存器位一览 .....	961
27.3 各运行模式的寄存器状态 .....	984
<b>第 28 章 电特性 .....</b>	<b>996</b>
28.1 绝对最大额定值 .....	996
28.2 DC 特性 .....	997
28.3 AC 特性 .....	1001
28.3.1 时钟时序 .....	1002
28.3.2 控制信号的时序 .....	1004



28.3.3	AC 总线的时序规格 .....	1007
28.3.4	直接存储器存取控制器 (DMAC) 的时序 .....	1042
28.3.5	多功能定时器脉冲单元 2 (MTU2) 的时序 .....	1043
28.3.6	多功能定时器脉冲单元 2S (MTU2S) 的时序 .....	1044
28.3.7	I/O 端口的时序 .....	1044
28.3.8	看门狗定时器 (WDT) 的时序 .....	1045
28.3.9	串行通信接口 (SCI) 的时序 .....	1045
28.3.10	带 FIFO 的串行通信接口 (SCIF) 的时序 .....	1047
28.3.11	串行通信单元 (SSU) 的时序 .....	1049
28.3.12	端口输出允许 (POE) 的时序 .....	1052
28.3.13	I <sup>2</sup> C 总线接口 2 (I <sup>2</sup> C2) 的时序 .....	1053
28.3.14	UBC 的触发时序 .....	1054
28.3.15	A/D 转换器的时序 .....	1054
28.3.16	AC 特性的测量条件 .....	1055
28.4	A/D 转换器特性 .....	1055
28.5	闪存特性 .....	1056
28.6	使用注意事项 .....	1056
28.6.1	VCL 电容连接方法 .....	1056
附录	.....	1057
附录 A	引脚状态.....	1057
附录 B	总线相关信号的引脚状态.....	1080
附录 C	型号一览.....	1095
附录 D	外形尺寸图.....	1097
索引	.....	1100

## 第 1 章 概要

### 1.1 SH7083/84/85/86 的特点

本 LSI 是以瑞萨科技的 RISC 方式 CPU 为核心，集聚了系统构成上所需外围功能的 RISC 单片机。

本 LSI 的 CPU 有 RISC (Reduced Instruction Set Computer) 方式的指令系统，因为以 1 条指令 1 个状态 (1 个系统时钟周期) 执行基本指令，所以大大提高了指令的执行速度。另外，采用内部 32 位结构，强化了数据处理能力。对于以前的单片机不能实现的高速性实时控制等应用系统，使用本 LSI 的 CPU 能构成低成本、高性能和高功能的系统。

作为系统构成上需要的外围功能，本 LSI 内置大容量 ROM、RAM、直接存储器存取控制器 (DMAC)、数据传送控制器 (DTC)、定时器、串行通信接口 (SCI)、带 FIFO 的串行通信接口 (SCIF)、同步通信单元 (SSU)、A/D 转换器、中断控制器 (INTC)、I/O 端口和 I<sup>2</sup>C 总线接口 2 (I<sup>2</sup>C2) 等。

另外，通过外部存储器存取支持功能，本 LSI 能直接连接到存储器和外围 LSI。从而能降低系统的成本。

内部 ROM 有内置闪存的 F-ZTAT™ (Flexible Zero Turn Around Time) 版\* 和掩模型 ROM 版两种。闪存除了能使用支持本 LSI 的编程器进行编程以外，还能通过软件进行编程和擦除，因此用户能在 LSI 安装于电路板上的状态下改写闪存。

本 LSI 的特点如表 1.1 所示。

**【注】** \* F-ZTAT™ 是株式会社瑞萨科技的商标。

表 1.1 SH7083/84/85/86 的特点

项目	特 点
CPU	<ul style="list-style-type: none"> <li>• 32 位 RISC (Reduced Instruction Set Computer) 方式 CPU</li> <li>• 指令长度: 通过固定为 16 位提高代码效率</li> <li>• 装入 - 存储结构 (在寄存器之间进行基本运算)</li> <li>• 通用寄存器: 32 位 ×16 个</li> <li>• 流水线: 5 段流水线方式</li> <li>• 内置乘法器: 以 2 ~ 5 周期执行 32×32→64 的乘法运算</li> <li>• 基本指令: 62 种面向 C 语言的指令系统</li> </ul> <p>【注】 请注意: 槽非法指令规格和以前的 SH-2 有所不同。详细内容请参照 “5.8.4, 有关槽非法指令异常处理的注意事项”。</p>
运行模式	<ul style="list-style-type: none"> <li>• 运行模式 <ul style="list-style-type: none"> <li>单芯片模式</li> <li>扩展 ROM 有效模式</li> <li>扩展 ROM 无效模式</li> </ul> </li> <li>• 处理状态 <ul style="list-style-type: none"> <li>程序执行状态</li> <li>异常处理状态</li> <li>总线权释放状态</li> </ul> </li> <li>• 低功耗状态 <ul style="list-style-type: none"> <li>睡眠模式</li> <li>软件待机模式</li> <li>深度软件待机模式</li> <li>模块待机模式</li> </ul> </li> </ul>
用户断点控制器 (UBC)	<ul style="list-style-type: none"> <li>• 能将地址、数据值、存取类型和数据大小全部设定为断点条件</li> <li>• 支持顺序断点功能</li> <li>• 2 个断点通道</li> </ul>
内部 ROM	<ul style="list-style-type: none"> <li>• 256K 字节或者 512K 字节</li> </ul>
内部 RAM	<ul style="list-style-type: none"> <li>• 16K 字节或者 32K 字节</li> </ul>
总线状态控制器 (BSC)	<ul style="list-style-type: none"> <li>• 支持 9 个区域的地址空间: 8 个最大 64M 字节的区域 (CS0 ~ 7) 和 1 个最大 1G 字节的区域 (CS8) (SH7083 为 3 个区域, SH7084/85 为 8 个区域, SH7086 为 9 个区域)</li> <li>• 8 位外部总线</li> <li>• 16 位外部总线</li> <li>• 32 位外部总线 (只限 SH7085/86)</li> <li>• 各区域可独立设定以下功能: <ul style="list-style-type: none"> <li>总线宽 (8、16、32 位)</li> <li>存取等待周期数</li> <li>存取等待周期的设定</li> </ul> </li> <li>按区域指定连接的存储器, 支持 SRAM、带字节选择的 SRAM、突发 ROM (时钟同步 / 异步)、MPX-I/O、突发 MPX-I/O、SDRAM 和 PCMCIA</li> <li>• 对目标区域输出片选信号</li> </ul>
直接存储器存取控制器 (DMAC)	<ul style="list-style-type: none"> <li>• 4 个通道</li> <li>• 可进行外部请求</li> <li>• 突发模式和周期挪用模式</li> </ul>

项目	特点
数据传送控制器 (DTC)	<p>可通过外围 I/O 的中断请求, 进行与 CPU 独立的数据传送</p> <ul style="list-style-type: none"> <li>可按各中断源设定传送模式 (设定存储器中的传送模式)</li> <li>可对 1 个启动源进行多个数据传送</li> <li>丰富的传送模式 可选择正常模式 / 重复模式 / 块传送模式</li> <li>可将传送单位设定为字节 / 字 / 长字</li> <li>对 CPU 请求启动 DTC 的中断 在 1 次数据传送结束后, 可对 CPU 产生中断</li> <li>在指定的数据传送全部结束后, 可对 CPU 产生中断</li> </ul>
中断控制器 (INTC)	<ul style="list-style-type: none"> <li>9 个外部中断引脚 (NMI、IRQ7 ~ IRQ0)</li> <li>内部外围中断: 设定各模块的优先级</li> <li>向量地址: 各中断源固有的向量地址</li> </ul>
用户调试接口 (H-UDI) (只限 F-ZTAT 版)	<ul style="list-style-type: none"> <li>支持 E10A 仿真器</li> </ul>
高级用户调试器 (AUD) (只限支持 E10A 全功能的 F-ZTAT 版)	<ul style="list-style-type: none"> <li>支持 E10A 仿真器</li> </ul>
时钟振荡器 (CPG)	<ul style="list-style-type: none"> <li>时钟模式: 对输入时钟可选择外部输入或者晶体谐振器</li> <li>生成 5 种时钟 CPU 时钟: 80MHz (Max.) 总线时钟: 40MHz (Max.) 外围时钟: 40MHz (Max.) MTU2 专用时钟: 40MHz (Max.) MTU2S 专用时钟: 80MHz (Max.)</li> </ul>
看门狗定时器 (WDT)	<ul style="list-style-type: none"> <li>1 个通道的看门狗定时器</li> <li>能请求中断</li> </ul>

项目	特点
多功能定时器脉冲单元 2 (MTU2)	<ul style="list-style-type: none"> <li>以 6 个通道的 16 位定时器为基础，最多能输入 / 输出 16 种脉冲 (SH7083 最多为 13 种) 以及输入 3 个脉冲</li> <li>21 个输出比较匹配寄存器兼输入捕捉寄存器</li> <li>共有 21 个独立的比较匹配电路</li> <li>能选择 8 种计数器输入时钟</li> <li>输入捕捉功能</li> <li>脉冲输出模式 交替 / PWM / 互补 PWM / 复位同步 PWM</li> <li>多个计数器的同步功能</li> <li>互补 PWM 输出模式 输出用于控制 6 相 (SH7083 为 4 相) 反相器的非重叠波形 空载时间的自动设定 可在 0 ~ 100% 范围内任意设定 PWM 占空比 输出 OFF 功能 A/D 转换请求的延迟功能 空载时间的补偿功能 波峰 / 波谷的跳过中断功能</li> <li>复位同步 PWM 模式 3 相输出任意占空比的正相 / 反相 PWM 波形</li> <li>相位计数模式 能处理 2 相编码器的计数</li> </ul>
多功能定时器脉冲单元 (MTU2S)	<ul style="list-style-type: none"> <li>只限 MTU2 的通道 3、4、5 的子集版</li> <li>可运行在最大 80MHz</li> </ul>
端口输出的允许 (POE)	<ul style="list-style-type: none"> <li>对 MTU2/MTU2S 波形输出引脚进行高阻抗控制</li> </ul>
比较匹配定时器 (CMT)	<ul style="list-style-type: none"> <li>16 位计数器</li> <li>产生比较匹配中断</li> <li>2 个通道</li> </ul>
串行通信接口 (SCI)	<ul style="list-style-type: none"> <li>时钟同步 / 异步方式</li> <li>3 个通道</li> </ul>
带 FIFO 的串行通信接口 (SCIF)	<ul style="list-style-type: none"> <li>时钟同步 / 异步方式</li> <li>内置分别用于接受和发送的 16 字节的 FIFO</li> <li>1 个通道</li> </ul>
同步串行通信单元 (SSU)	<ul style="list-style-type: none"> <li>可选择主控模式和从属模式</li> <li>可选择标准模式和双向模式</li> <li>可从 8/16/32 位中选择接受和发送数据的长度</li> <li>可同时进行接受和发送 (全双工)</li> <li>可连续进行串行通信</li> <li>1 个通道</li> </ul>
I <sup>2</sup> C 总线接口 2 (I <sup>2</sup> C2)	<ul style="list-style-type: none"> <li>符合 Philips 公司提倡的 I<sup>2</sup>C 总线接口方式</li> <li>内置主控模式 / 从属模式</li> <li>可连续进行接受和发送</li> <li>可选择 I<sup>2</sup>C 总线格式 / 时钟同步串行格式</li> <li>1 个通道</li> </ul>

项目	特点
A/D 转换器 (ADC)	<ul style="list-style-type: none"> <li>• 10 位 × 8 个通道 (SH7083/84/85)</li> <li>• 10 位 × 16 个通道 (SH7086)</li> <li>• 能通过外部触发或者 MTU2/MTU2S 请求转换</li> <li>• 内置 2 个单元的采样 &amp; 保持功能 (能同时进行 2 个通道的采样) (SH7083/84/85)</li> <li>• 内置 3 个单元的采样 &amp; 保持功能 (能同时进行 3 个通道的采样) (SH7086)</li> </ul>
I/O 端口	<ul style="list-style-type: none"> <li>• 65 个通用输入 / 输出引脚和 8 个通用输入引脚 (SH7083)</li> <li>• 76 个通用输入 / 输出引脚和 8 个通用输入引脚 (SH7084)</li> <li>• 100 个通用输入 / 输出引脚和 8 个通用输入引脚 (SH7085)</li> <li>• 118 个通用输入 / 输出引脚和 16 个通用输入引脚 (SH7086)</li> <li>• 输入 / 输出兼用端口能按位进行输入 / 输出的转换</li> </ul>
封装	<ul style="list-style-type: none"> <li>• TQFP1414-100 (0.5 节距) (SH7083)</li> <li>• LQFP2020-112 (0.65 节距) (SH7084)</li> <li>• LQFP2020-144 (0.5 节距) (SH7085)</li> <li>• LQFP2424-176 (0.5 节距) (SH7086)</li> </ul>
电源电压	Vcc: 3.0 ~ 3.6V 或者 4.5 ~ 5.5V, AVcc: 4.5 ~ 5.5V

## 1.2 框图

SH7083/84/85/86 的框图如图 1.1 所示。

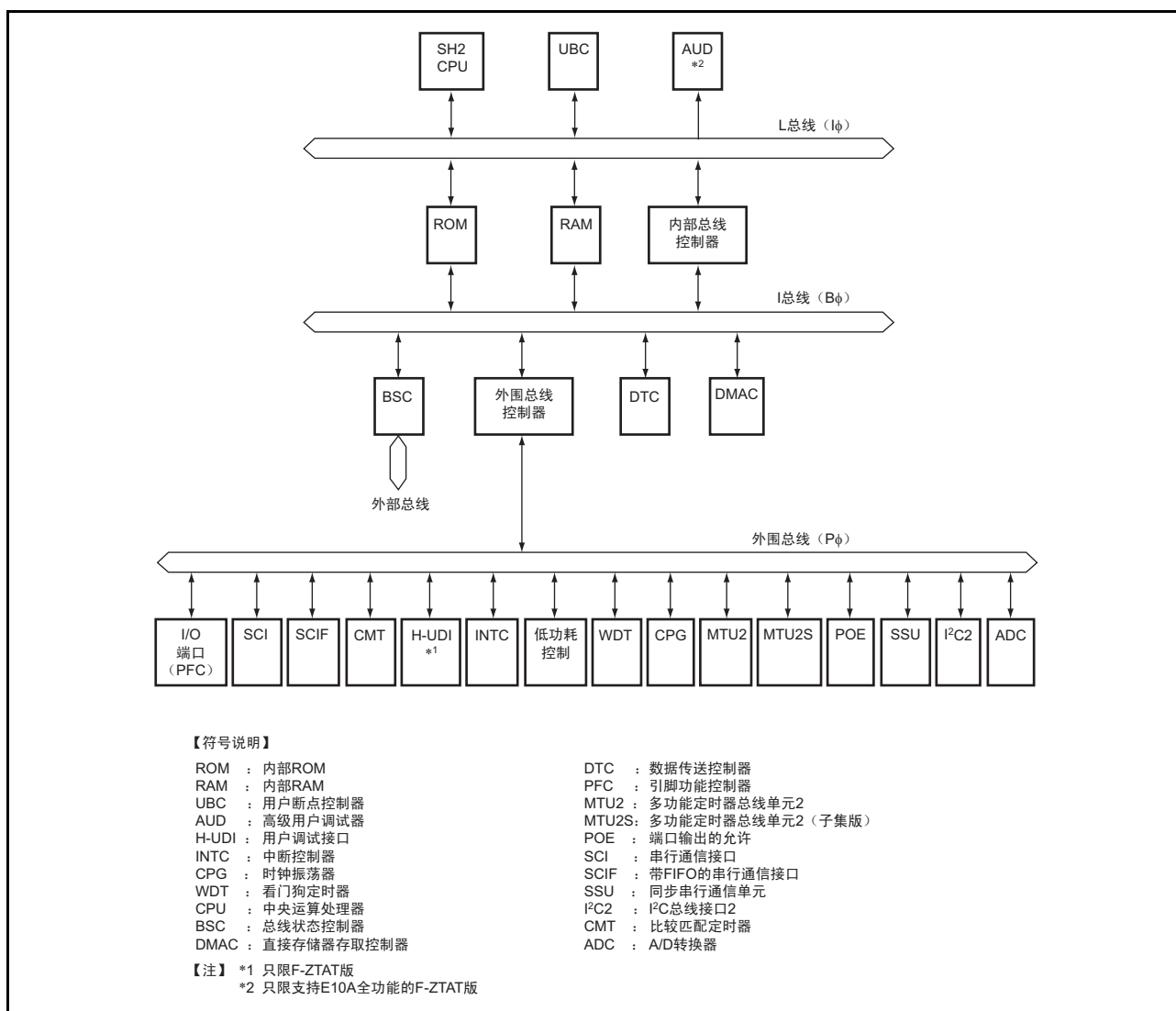


图 1.1 SH7083/84/85/86 的框图

1.3 引脚排列图

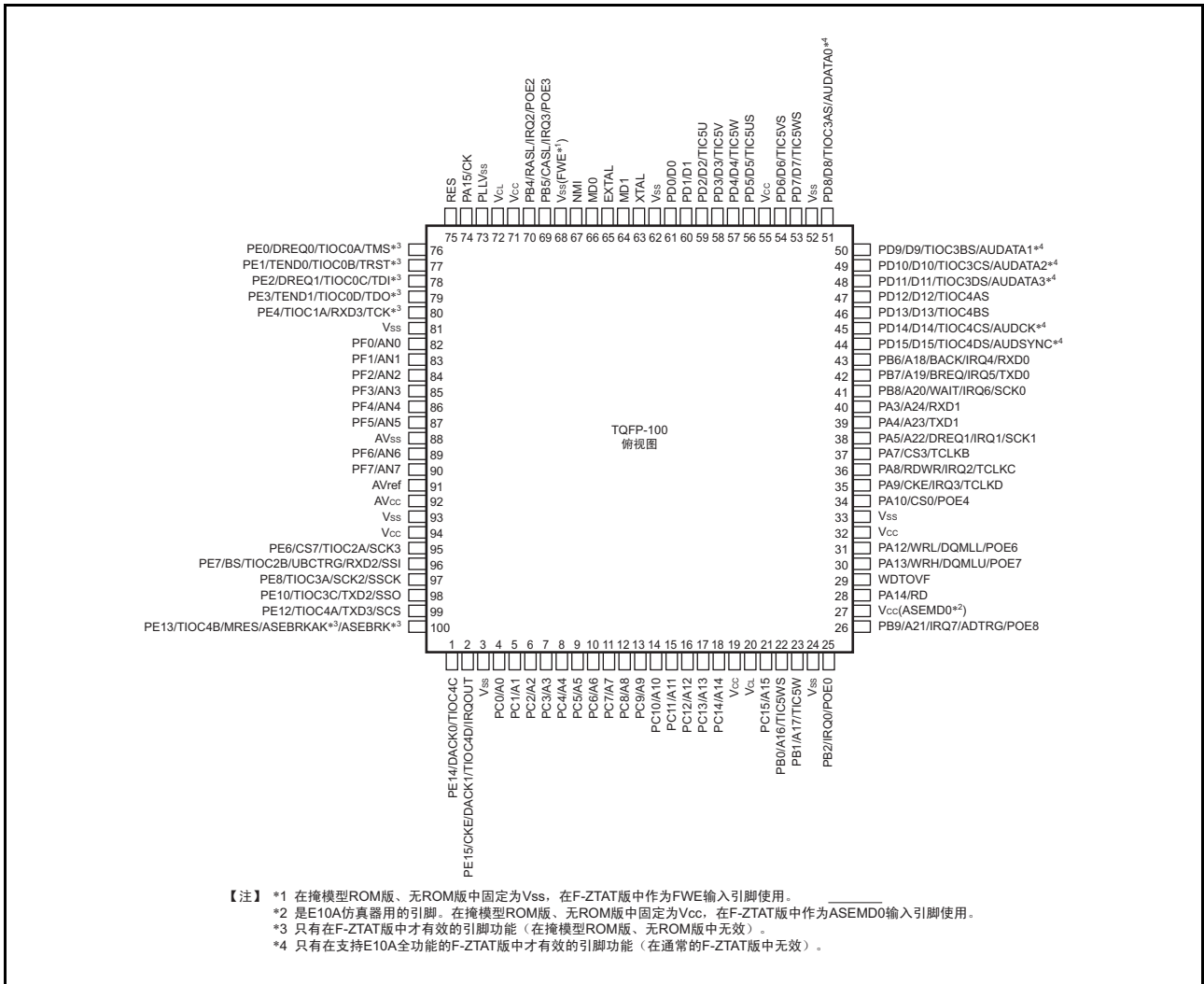


图 1.2 SH7083 的引脚排列图（TQFP1414-100）



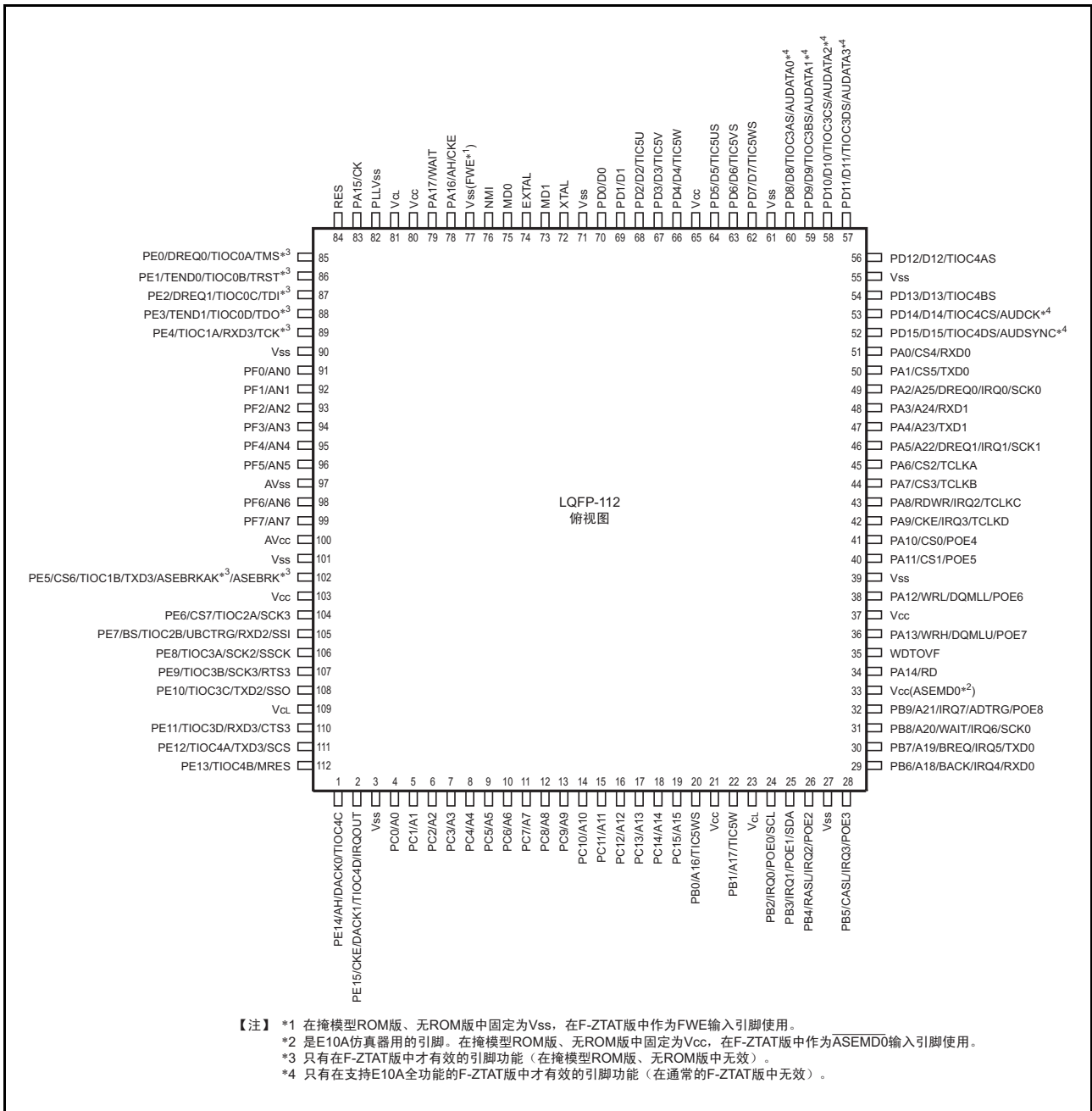


图 1.3 SH7084 的引脚排列图

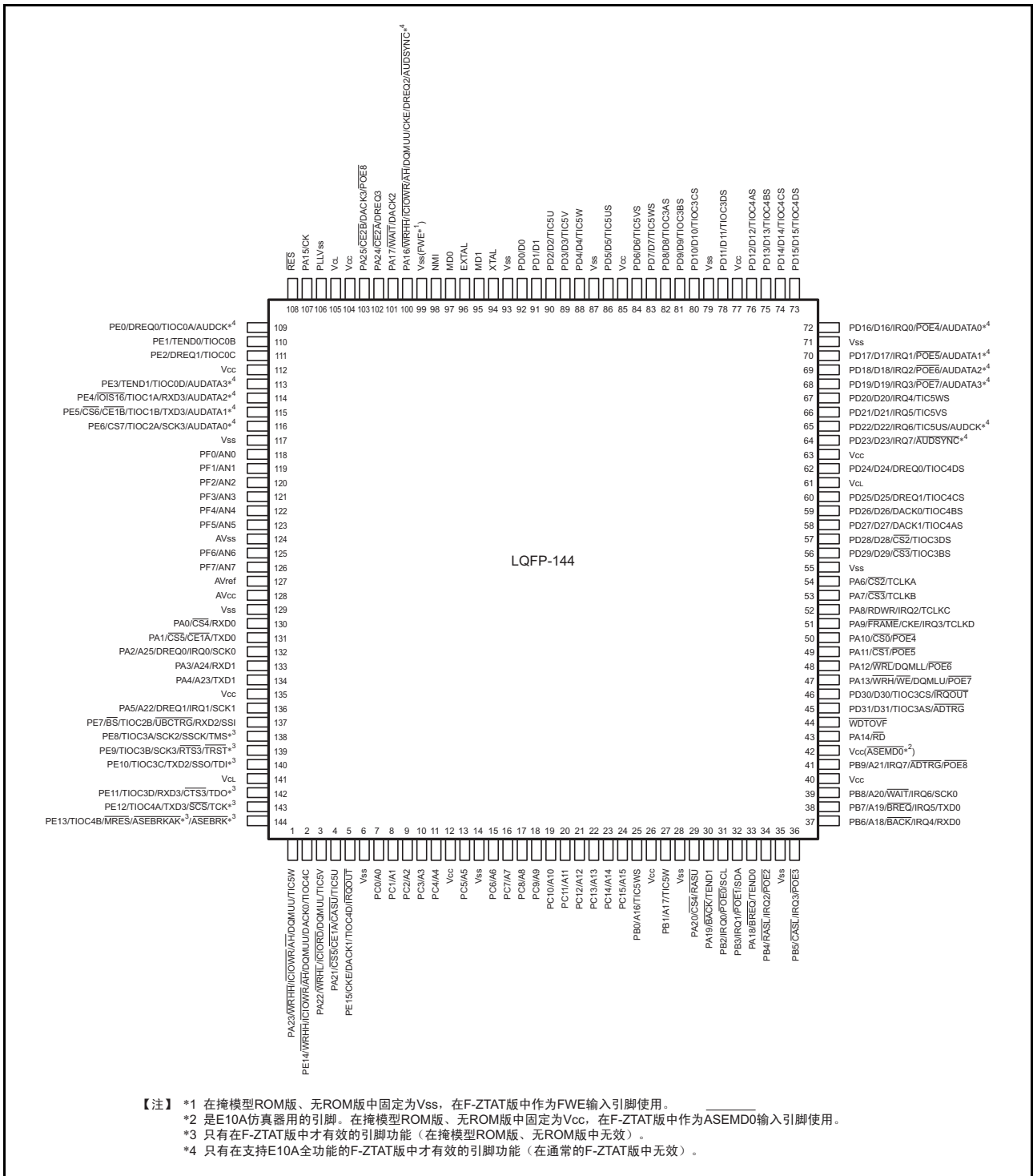


图 1.4 SH7085 的引脚排列图

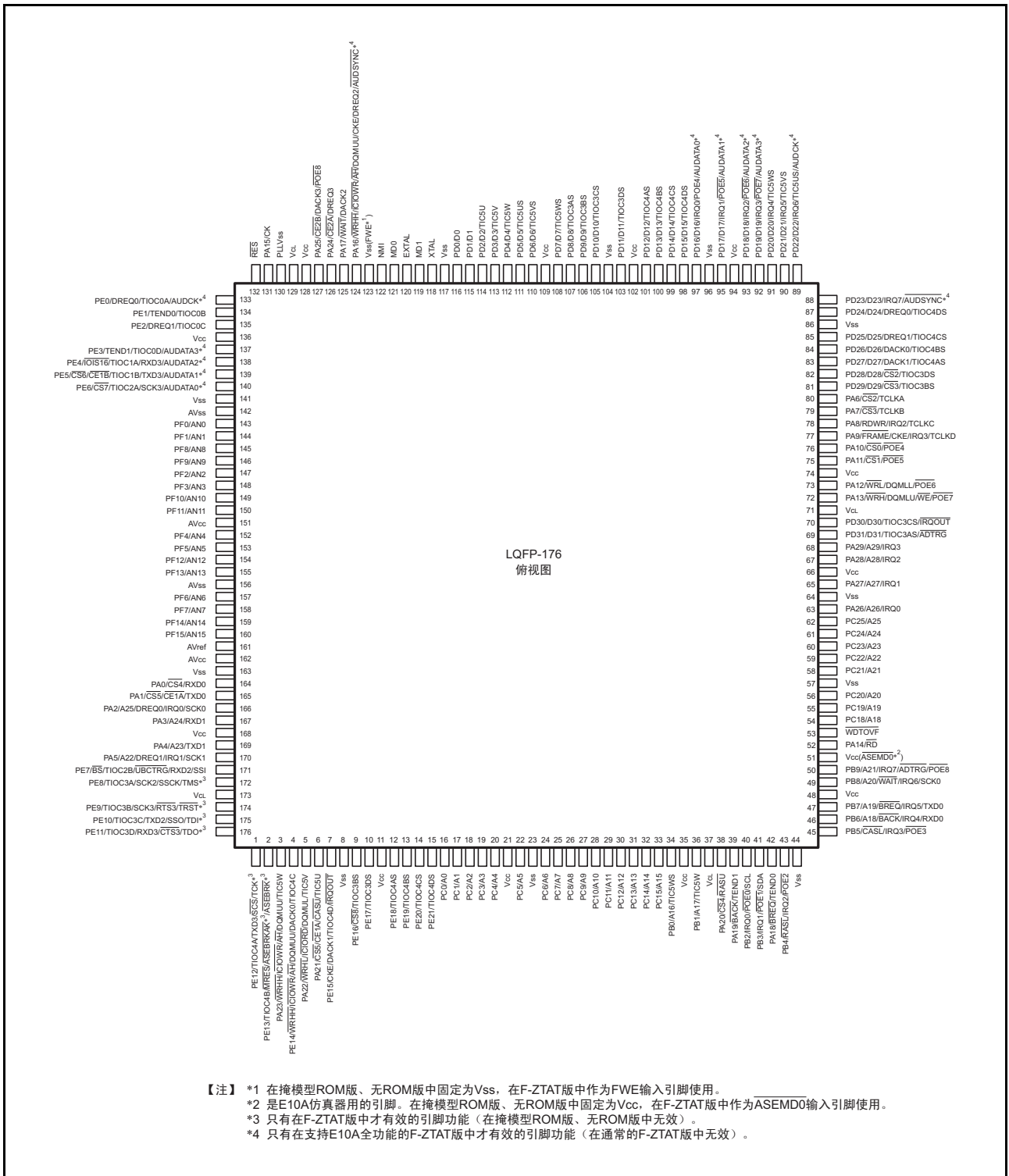


图 1.5 SH7086 的引脚排列图

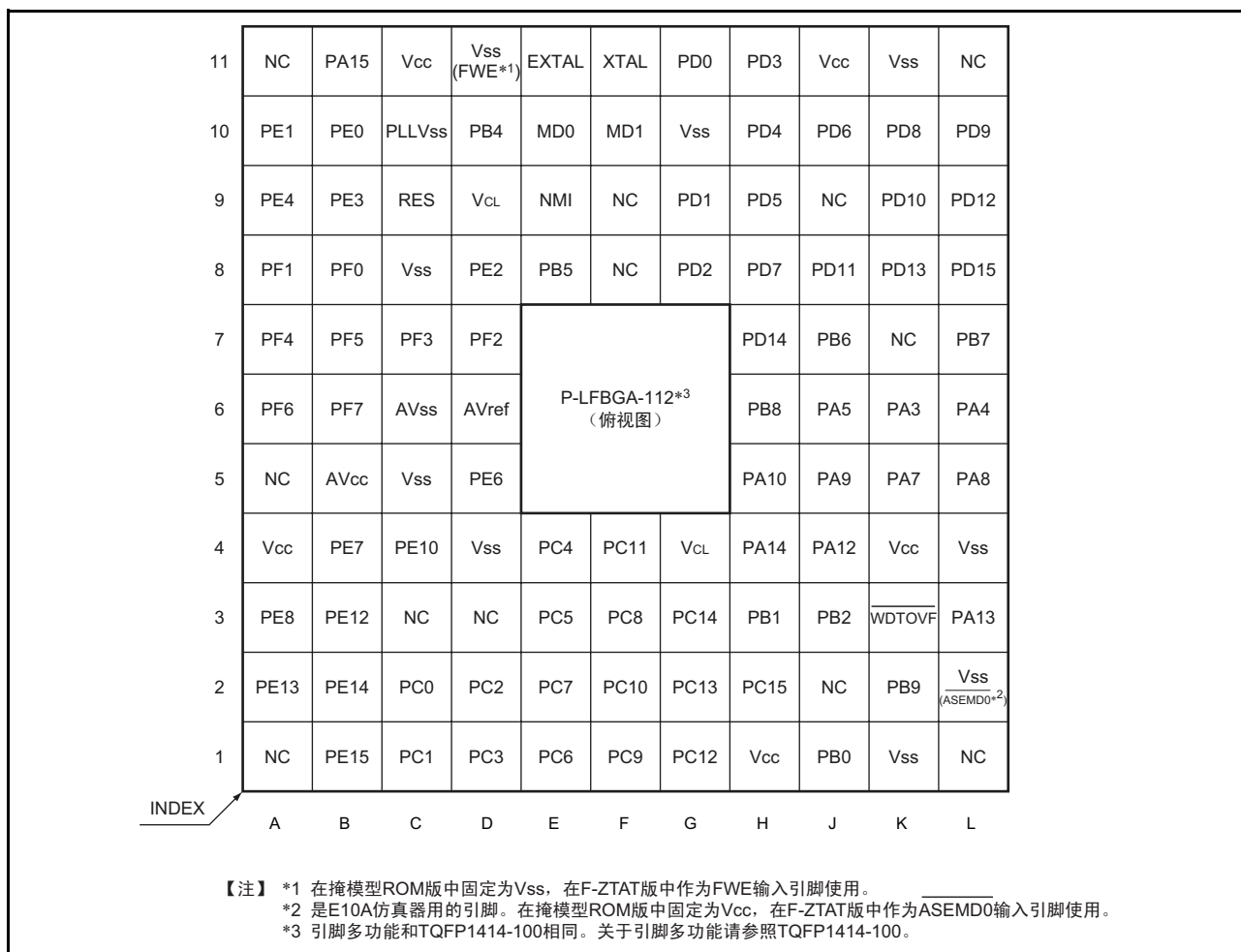


图 1.6 SH7083 引脚排列图 (P-LFBGA-112)

## 1.4 引脚功能

各引脚的功能如表 1.2 所示。

表 1.2 引脚功能

分类	引脚名	输入 / 输出	名称	功 能
电源	Vcc	输入	电源	是电源引脚。必须将全部 Vcc 引脚连接到系统电源。如果有开路引脚，就不运行。
	Vss	输入	接地	是接地引脚。必须将全部 Vss 引脚连接到系统电源 (0V)。如果有开路引脚，就不运行。
	VCL	输出	内部降压电源	是用于内部降压电源的外接电容引脚。必须通过 0.47 $\mu$ F 的电容将全部的 VCL 引脚连接到 Vss (靠近引脚)。
时钟	PLLvss	输入	用于 PLL 的接地	是用于内部 PLL 振荡器的接地引脚。
	EXTAL	输入	外部时钟	连接晶体谐振器，也能输入外部时钟。
	XTAL	输出	晶体	连接晶体谐振器。
	CK	输出	系统时钟	给外部芯片提供系统时钟。
运行模式的控制	MD1、MD0	输入	模式设定	设定运行模式。请不要在运行中改变此引脚。
	FWE	输入	允许闪存的编程	是用于闪存的引脚。能防止闪存被编程 / 擦除。
系统控制	$\overline{\text{RES}}$	输入	上电复位	当此引脚为低电平时，处于上电复位状态。
	$\overline{\text{MRES}}$	输入	手动复位	当此引脚为低电平时，处于手动复位状态。
	$\overline{\text{WDTOVF}}$	输出	看门狗定时器的上溢	是来自 WDT 的上溢输出信号。 如果需要下拉，就必须使用大于等于 1M $\Omega$ 的电阻。
	$\overline{\text{BREQ}}$	输入	总线权请求	在外部芯片请求释放总线权时，置为低电平。
	$\overline{\text{BACK}}$	输出	总线权请求的应答	表示已将总线权释放给外部芯片。能得知输出 BREQ 信号的芯片已接受 BACK 信号，获得了总线权。
中断	NMI	输入	非屏蔽中断	是非屏蔽的中断请求引脚。如果不使用，就必须固定为高电平或者低电平。
	IRQ7 ~ IRQ0	输入	中断请求 7 ~ 0	是可屏蔽的中断请求引脚。 可选择电平输入或者边沿输入。在选择边沿输入时，可选择上升沿、下降沿或者双边沿。
	$\overline{\text{IRQOUT}}$	输出	中断请求的输出	表示已发生中断源。也能在总线释放时得知中断的发生。
地址总线	A29 ~ A0	输出	地址总线	输出地址。 SH7083 为 A24 ~ A0。 SH7084/85 为 A25 ~ A0。
数据总线	D31 ~ D0	输入 / 输出	数据总线	是 32 位双向总线。 SH7083/84 为 D15 ~ D0。

分类	引脚名	输入 / 输出	名称	功能
总线控制	$\overline{CS8} \sim \overline{CS0}$	输出	片选 8 ~ 0	用于外部存储器或者芯片的片选信号。 SH7083 为 $\overline{CS7}$ 、 $\overline{CS3}$ 、 $\overline{CS0}$ 。 SH7084/85 为 $\overline{CS7} \sim \overline{CS0}$ 。
	$\overline{RD}$	输出	读	表示从外部芯片读数据。
	$\overline{RDWR}$	输出	读写	是读写信号。
	$\overline{BS}$	输出	总线开始	总线周期开始
	$\overline{AH}$	输出	地址保持	是对于使用地址 / 数据多路复用总线的芯片的地址保持时序信号。只限 SH7084/85/86。
	$\overline{FRAME}$	输出	帧信号	对于突发 MPX-I/O 接口, 在最后的总线周期前被置为低电平, 表示下一个总线周期为最后的存取。 只限 SH7085/86。
	$\overline{WRHH}$	输出	HH 字节写	表示写外部数据的 bit31 ~ 24。 只限 SH7085/86。
	$\overline{WRHL}$	输出	HL 字节写	表示写外部数据的 bit23 ~ 16。 只限 SH7085/86。
	$\overline{WRH}$	输出	高位字节写	表示写外部数据的 bit15 ~ 8。
	$\overline{WRL}$	输出	低位字节写	表示写外部数据的 bit7 ~ 0。
	$\overline{WAIT}$	输入	等待	是给存取外围空间时的总线周期插入等待周期的输入。
	$\overline{RASL}$	输出	RAS	连接 SDRAM 的 $\overline{RAS}$ 引脚。
	$\overline{RASU}$	输出	RAS	连接 SDRAM 的 $\overline{RAS}$ 引脚。 只限 SH7085/86。
	$\overline{CASL}$	输出	CAS	连接 SDRAM 的 $\overline{CAS}$ 引脚。
	$\overline{CASU}$	输出	CAS	连接 SDRAM 的 $\overline{CAS}$ 引脚。 只限 SH7085/86。
	$\overline{CKE}$	输出	时钟允许	连接 SDRAM 的 $\overline{CKE}$ 引脚。
	$\overline{DQMUU}$	输出	HH 字节选择	选择 SDRAM 数据总线的 bit31 ~ 24。 只限 SH7085/86。
	$\overline{DQMUL}$	输出	HL 字节选择	选择 SDRAM 数据总线的 bit23 ~ 16。 只限 SH7085/86。
	$\overline{DQMLU}$	输出	高位字节选择	选择 SDRAM 数据总线的 bit15 ~ 8。
	$\overline{DQMLL}$	输出	低位字节选择	选择 SDRAM 数据总线的 bit7 ~ 0。
	$\overline{CE1A}$	输出	PCMCIA 卡的低位字节选择	是用于 PCMCIA 连接到区域 5 的片选引脚。 只限 SH7085/86。
	$\overline{CE1B}$	输出	PCMCIA 卡的低位字节选择	是用于 PCMCIA 连接到区域 6 的片选引脚。 只限 SH7085/86。
	$\overline{CE2A}$	输出	PCMCIA 卡的高位字节选择	是用于 PCMCIA 连接到区域 5 的片选引脚。 只限 SH7085/86。
$\overline{CE2B}$	输出	PCMCIA 卡的高位字节选择	是用于 PCMCIA 连接到区域 6 的片选引脚。 只限 SH7085/86。	

分类	引脚名	输入 / 输出	名称	功 能
总线控制	$\overline{\text{ICIORWR}}$	输出	PCMCIA I/O 写选通	连接 PCMCIA I/O 的写选通引脚。 只限 SH7085/86。
	$\overline{\text{ICIORDR}}$	输出	PCMCIA I/O 读选通	连接 PCMCIA I/O 的读选通引脚。 只限 SH7085/86。
	$\overline{\text{WE}}$	输出	PCMCIA 存储器写选通	连接 PCMCIA 存储器的写选通引脚。 只限 SH7085/86。
	$\overline{\text{IOIS16}}$	输入	PCMCIA 动态总线宽度变更	在小端法 (little endian) 时, 表示 PCMCIA 的 16 位宽度的 I/O。本 LSI 不支持小端法, 所以必须固定为低电平。 只限 SH7085/86。
直接存储器存取控制器 (DMAC)	DREQ3 ~ DREQ0	输入	DMA 传送请求	是从外部输入 DMA 传送请求的引脚。 SH7083/84 为 DREQ1、DREQ0。
	DACK3 ~ DACK0	输出	DMA 传送选通	是对 DMA 传送请求的外部 I/O 输出门控信号的引脚。 SH7083/84 为 DACK1、DACK0。
	TEND1、TEND0	输出	DMA 传送结束	是 DMA 传送结束的输出引脚。
多功能定时器脉冲单元 2 (MTU2)	TCLKA、TCLKB、TCLKC、TCKLD	输入	MTU2 定时器的时钟输入	是定时器的外部时钟输入引脚。 SH7083 为 TCLKB、TCLKC、TCKLD。
	TIOC0A、TIOC0B、TIOC0C、TIOC0D	输入 / 输出	MTU2 的输入捕捉 / 输出比较 (通道 0)	是 TGRA_0 ~ TGRD_0 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。
	TIOC1A、TIOC1B	输入 / 输出	MTU2 的输入捕捉 / 输出比较 (通道 1)	是 TGRA_1、TGRB_1 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。
	TIOC2A、TIOC2B	输入 / 输出	MTU2 的输入捕捉 / 输出比较 (通道 2)	是 TGRA_2、TGRB_2 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。
	TIOC3A、TIOC3B、TIOC3C、TIOC3D	输入 / 输出	MTU2 的输入捕捉 / 输出比较 (通道 3)	是 TGRA_3 ~ TGRD_3 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。 SH7083 为 TIOC3A、TIOC3C。
	TIOC4A、TIOC4B、TIOC4C、TIOC4D	输入 / 输出	MTU2 的输入捕捉 / 输出比较 (通道 4)	是 TGRA_4 ~ TGRD_4 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。
	TIC5U、TIC5V、TIC5W	输入	MTU2 的输入捕捉 (通道 5)	是 TGRU_5、TGRV_5、TGRW_5 的输入捕捉的输入引脚。

分类	引脚名	输入 / 输出	名称	功 能
多功能定时器 脉冲单元 2S (MTU2S)	TIOC3AS、 TIOC3BS、 TIOC3CS、 TIOC3DS	输入 / 输出	MTU2S 的输入 捕捉 / 输出比较 (通道 3)	是 TGRA_3S ~ TGRD_3S 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。
	TIOC4AS、 TIOC4BS、 TIOC4CS、 TIOC4DS	输入 / 输出	MTU2S 的输入 捕捉 / 输出比较 (通道 4)	是 TGRA_4S ~ TGRD_4S 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。
	TIC5US、 TIC5VS、 TIC5WS	输入	MTU2S 的输入 捕捉 (通道 5)	是 TGRU_5S、TGRV_5S、TGRW_5S 的输入捕 捉的输入引脚。
端口输出的允许 (POE)	$\overline{POE8} \sim \overline{POE0}$	输入	端口输出的控制	是将 MTU2/MTU2S 波形输出引脚变为高阻抗状态 的请求信号的输入引脚。 SH7083 为 $\overline{POE8} \sim \overline{POE6}$ 、 $\overline{POE4} \sim \overline{POE2}$ 、 $\overline{POE0}$ 。
串行通信接口 (SCI)	TXD2 ~ TXD0	输出	发送数据	是用于发送数据的引脚。
	RXD2 ~ RXD0	输入	接收数据	是用于接收数据的引脚。
	SCK2 ~ SCK0	输入 / 输出	串行时钟	是时钟的输入 / 输出引脚。
带 FIFO 的串行 通信接口 (SCIF)	TXD3	输出	发送数据	是用于发送数据的引脚。
	RXD3	输入	接收数据	是用于接收数据的引脚。
	SCK3	输入 / 输出	串行时钟	是时钟的输入 / 输出引脚。
	$\overline{RTS3}$	输出	发送请求	是调制解调器的控制引脚。 SH7083 没有此引脚。
	$\overline{CTS3}$	输入	发送允许	是调制解调器的控制引脚。 SH7083 没有此引脚。
同步串行通信 单元 (SSU)	SSO	输入 / 输出	数据	是数据的输入 / 输出引脚。
	SSI	输入 / 输出	数据	是数据的输入 / 输出引脚。
	SSCK	输入 / 输出	时钟	是时钟的输入 / 输出引脚。
	$\overline{SCS}$	输入 / 输出	片选	是片选的输入 / 输出引脚。
I <sup>2</sup> C 总线接口 2 (I <sup>2</sup> C2)	SCL	输入 / 输出	I <sup>2</sup> C 时钟的 输入 / 输出	是 I <sup>2</sup> C 总线的时钟输入 / 输出引脚。 只限 SH7084/85/86。
	SDA	输入 / 输出	I <sup>2</sup> C 数据的 输入 / 输出	是 I <sup>2</sup> C 总线的数据输入 / 输出引脚。 只限 SH7084/85/86。



分类	引脚名	输入 / 输出	名称	功 能
A/D 转换器 (ADC)	AN15 ~ AN0	输入	模拟输入引脚	是模拟输入引脚。 SH7083/84/85 为 AN7 ~ AN0。
	$\overline{\text{ADTRG}}$	输入	A/D 转换的 触发输入	是用于 A/D 转换开始的外部触发输入引脚。
	AVref	输入	模拟基准电源	是模拟基准电引脚。 只限 SH7083/85/86。 (SH7084 在 LSI 内部连接 AVcc)
	AVcc	输入	模拟电源	是 A/D 转换器的电源引脚。在不使用 A/D 转换器 时, 必须连接到系统电源 (Vcc)。 必须将全部 AVcc 引脚连接到系统电源 (Vcc)。 如果有开路引脚, 就不运行。
	AVss	输入	模拟接地	是 A/D 转换器的接地引脚, 必须连接到系统电源 (0V)。 必须将全部 AVss 引脚连接到系统电源 (0V)。如 果有开路引脚, 就不运行。
I/O 端口	PA29 ~ PA0	输入 / 输出	通用端口	是 30 位通用输入 / 输出端口的引脚。 SH7083 为 PA15 ~ PA12、PA10 ~ PA7、 PA5 ~ PA3。 SH7084 为 PA17 ~ PA0。 SH7085 为 PA25 ~ PA0。
	PB9 ~ PB0	输入 / 输出	通用端口	是 10 位通用输入 / 输出端口的引脚。 SH7083 为 PB9 ~ PB4、PB2 ~ PB0。
	PC25 ~ PC18、 PC15 ~ PC0	输入 / 输出	通用端口	是 24 位通用输入 / 输出端口的引脚。 SH7083/84/85 为 PC15 ~ PC0。
	PD31 ~ PD0	输入 / 输出	通用端口	是 32 位通用输入 / 输出端口的引脚。 SH7083/84 为 PD15 ~ PD0。
	PE21 ~ PE0	输入 / 输出	通用端口	是 22 位通用输入 / 输出端口的引脚。 SH7083 为 PE15 ~ PE12、PE10、PE8 ~ PE6、 PE4 ~ PE0。 SH7084/85 为 PE15 ~ PE0。
	PF15 ~ PF0	输入	通用端口	是 16 位通用输入端口的引脚。 SH7083/84/85 为 PF7 ~ PF0。
用户断点控制器 (UBC)	$\overline{\text{UBCTR}}\overline{\text{G}}$	输出	用户断点的触发 输出	是 UBC 条件一致的触发输出引脚。
用户调试接口 (H-UDI) (只限 F-ZTAT 版)	TCK	输入	测试时钟	是测试时钟的输入引脚。
	TMS	输入	测试模式的选择	是测试模式选择信号的输入引脚。
	TDI	输入	测试数据的输入	是指令和数据的串行输入引脚。
	TDO	输出	测试数据的输出	是指令和数据的串行输出引脚。
	$\overline{\text{TRST}}$	输入	测试复位	是初始化信号的输入引脚。

分类	引脚名	输入 / 输出	名称	功 能
高级用户调试器 (AUD) (只限支持 E10A 全功能的 F- ZTAT 版)	AUDATA3 ~ AUDATA0	输出	AUD 数据	是转移地址的输出引脚。
	AUDCK	输出	AUD 时钟	是同步时钟的输出引脚。
	AUDSYNC	输出	AUD 同步信号	是数据起始位置识别信号的输出引脚。
E10A 接口 (只限 F-ZTAT 版)	ASEMD0	输入	ASE 模式	设定 ASE 模式。 当给本引脚输入低电平时，为 ASE 模式；当输入高电平时，为通常模式。在 ASE 模式中，能使用仿真器专用功能。如果引脚没有任何输入，就在内部进行上拉。
	ASEBRK	输入	断点请求	是 E10A 仿真器的断点输入。
	ASEBRKAK	输出	断点模式应答	表示 E10A 仿真器已进入断点模式。

## 【使用注意事项】

请不要下拉 WDT0VF 引脚。如果需要下拉，必须用大于等于 1MΩ 的电阻进行下拉。

## 第 2 章 CPU

### 2.1 特点

- 通用寄存器：32 位 × 16 个
- 基本指令：62 种
- 寻址方式：11 种
  - 寄存器直接 (Rn)
  - 寄存器间接 (@Rn)
  - 后增寄存器间接 (@Rn+)
  - 先减寄存器间接 (@-Rn)
  - 带位移量的寄存器间接 (@disp:4, Rn)
  - 带变址的寄存器间接 (@R0, Rn)
  - 带位移量的 GBR 间接 (@disp:8, GBR)
  - 带变址的 GBR 间接 (@R0, GBR)
  - 带位移量的 PC 相对 (@disp:8, PC)
  - PC 相对 (disp:8/disp:12/Rn)
  - 立即数 (#imm:8)

## 2.2 寄存器结构

寄存器有通用寄存器（32 位 ×16 个）、控制寄存器（32 位 ×3 个）和系统寄存器（32 位 ×4 个）共 3 种。

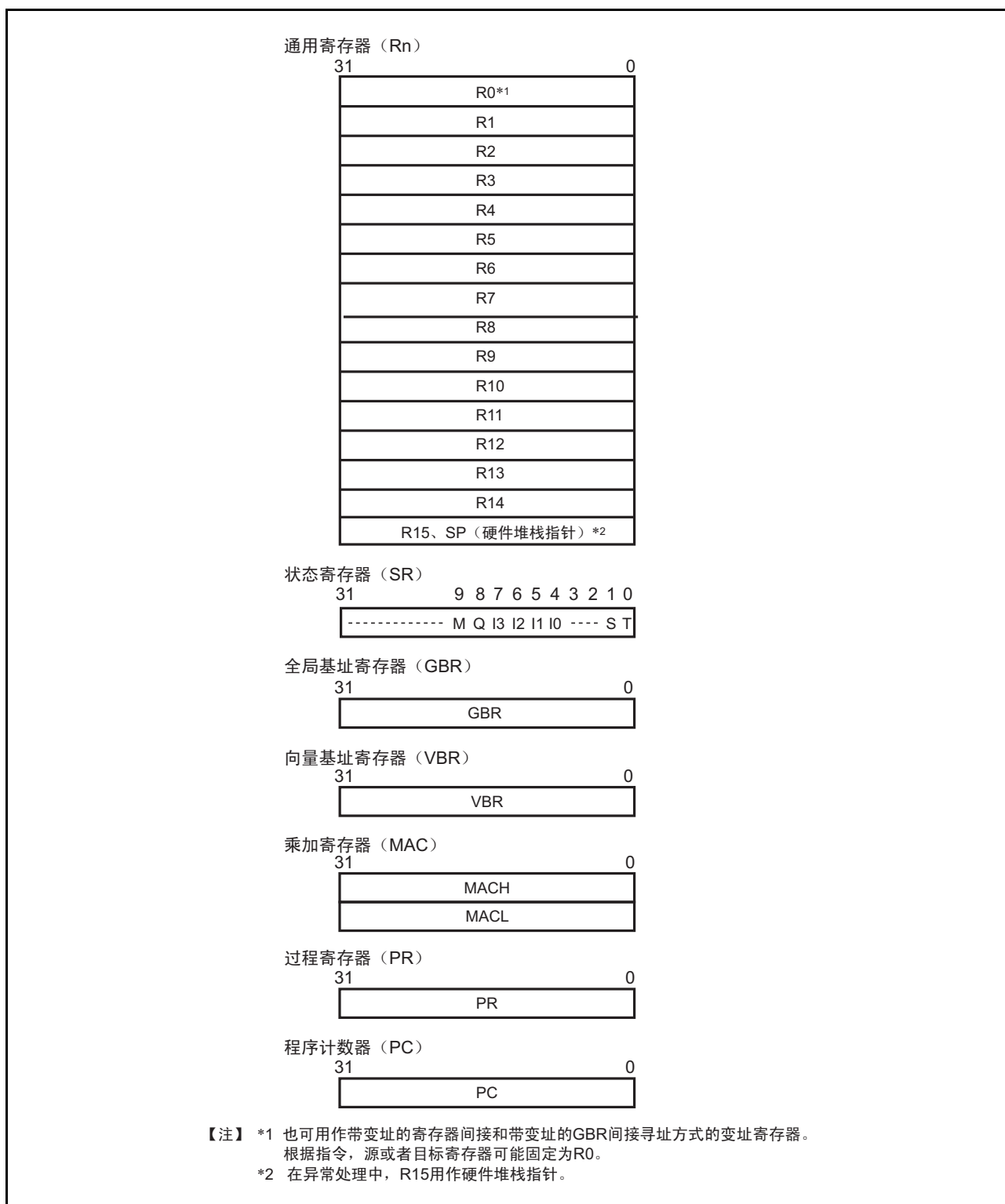


图 2.1 CPU 内部寄存器结构

## 2.2.1 通用寄存器 (Rn)

通用寄存器 (Rn) 为 32 位长, 从 R0 到 R15 共有 16 个, 用于数据处理和地址计算。R0 也用作变址寄存器, 有些指令能使用的寄存器固定为 R0。R15 用作硬件堆栈指针 (SP), 使用 R15 参照堆栈, 保存或者恢复异常处理中的状态寄存器 (SR) 和程序计数器 (PC)。

## 2.2.2 控制寄存器

控制寄存器为 32 位长, 有状态寄存器 (SR)、全局基址寄存器 (GBR) 和向量基址寄存器 (VBR) 共 3 个。SR 表示处理的状态。GBR 作为 GBR 间接寻址方式的基地址, 用于内部外围模块寄存器的数据传送等。VBR 用作含有中断的异常处理向量区的基地址。

### (1) 状态寄存器 (SR)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	M	Q	I[3:0]				-	-	S	T
初始值:	0	0	0	0	0	0	不定	不定	1	1	1	1	0	0	不定	不定
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 10	—	全 0	R	保留位 读写的值总是 0。
9	M	不定	R/W	由 DIV0U、DIV0S、DIV1 指令使用。
8	Q	不定	R/W	由 DIV0U、DIV0S、DIV1 指令使用。
7 ~ 4	I[3:0]	1111	R/W	中断屏蔽位
3、2	—	全 0	R	保留位 读写的值总是 0。
1	S	不定	R/W	S 位 由乘加指令使用。
0	T	不定	R/W	T 位 以下指令表示真 (1) 或者伪 (0): MOVT、CMP/cond、TAS、TST、BT(BT/S)、BF(BF/S)、SETT、CLRT 以下指令表示进位、借位、上溢、下溢: ADDV、ADDC、SUBV、SUBC、NEGC、DIV0U、DIV0S、DIV1、 SHAR、SHAL、SHLR、SHLL、ROTR、ROTL、ROTCR、ROTCL

### (2) 全局基址寄存器 (GBR)

表示 GBR 间接寻址方式的基地址。GBR 间接寻址方式用于内部外围模块的寄存器区等的数据传送和逻辑运算。

### (3) 向量基址寄存器 (VBR)

表示异常处理向量区的基地址。

## 2.2.3 系统寄存器

系统寄存器为 32 位长，有乘加寄存器 (MACH 和 MACL)、过程寄存器 (PR) 和程序计数器 (PC) 共 4 个。

### (1) 乘加寄存器 (MACH 和 MACL)

这是保存乘法运算和乘加运算结果的寄存器。

### (2) 过程寄存器 (PR)

这是保存子程序过程返回地址的寄存器。

### (3) 程序计数器 (PC)

PC 表示当前执行指令的 4 字节 (2 条指令) 后的地址。

## 2.2.4 寄存器的初始值

复位后的寄存器值如表 2.1 所示。

表 2.1 寄存器的初始值

分类	寄存器	初始值
通用寄存器	R0 ~ R14	不定
	R15 (SP)	向量地址表中 SP 的值
控制寄存器	SR	I3 ~ I0 为 1111 (H'F)，保留位为 0，其他位不定
	GBR	不定
	VBR	H'00000000
系统寄存器	MACH、MACL、PR	不定
	PC	向量地址表中 PC 的值

## 2.3 数据格式

### 2.3.1 寄存器的数据格式

寄存器操作数的数据长度总是长字（32 位）。在将存储器的数据装入到寄存器时，如果存储器操作数的数据长度为字节（8 位）或者字（16 位），就将数据扩展（符号扩展）为长字，保存到寄存器。

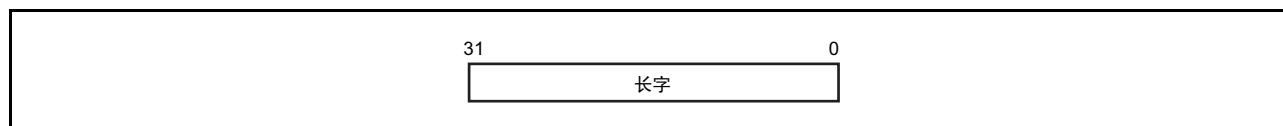


图 2.2 寄存器的数据格式

### 2.3.2 存储器的数据格式

有字节、字和长字数据格式。

字节数据可分配到任意地址，字数据必须从  $2n$  地址开始、长字数据必须从  $4n$  地址开始分配。如果不按此规定进行存取，就会发生地址错误。此时，不保证存取的结果。尤其是对硬件堆栈指针（SP、R15）指向的堆栈，用长字保存程序计数器（PC）和状态寄存器（SR），所以必须将硬件堆栈指针的值设定为  $4n$ 。

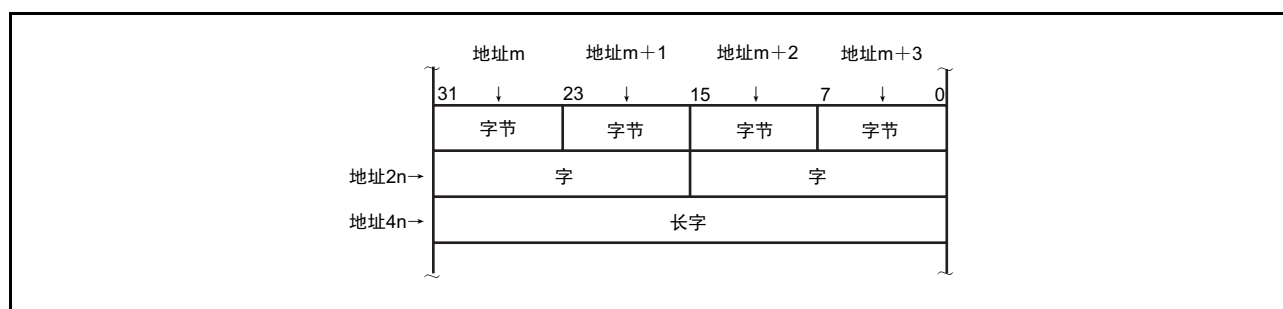


图 2.3 存储器的数据格式

### 2.3.3 立即数的数据格式

字节（8 位）的立即数分配在指令码中。

MOV、ADD、CMP/EQ 指令将立即数进行符号扩展后用长字进行运算，TST、AND、OR、XOR 指令将立即数进行零扩展后用长字进行运算。因此，如果 AND 指令使用立即数，目标寄存器的高 24 位总是被清除。

字和长字的立即数不分配在指令码中而分配在存储器的表中。通过带位移量的 PC 相对寻址方式的立即数传送指令（MOV），参照存储器的表。

## 2.4 指令特点

### 2.4.1 RISC 方式

指令为 RIS 方式，特点如下：

#### (1) 16 位固定长指令

指令长度全部是 16 位固定长，因此能提高程序码的效率。

#### (2) 1 条指令 / 1 个状态

采用流水线方式，基本指令能以 1 条指令执行 1 个状态。

#### (3) 数据长度

运算的基本数据长度是长字，存储器的存取长度能选择字节 / 字 / 长字。对于存储器的字节和字数据，在进行符号扩展后用长字进行运算；对于立即数，在算术运算时进行符号扩展而在逻辑运算时进行零扩展后，用长字进行运算。

表 2.2 字数据的符号扩展

本 LSI 的 CPU	说 明	其他 CPU 的例子
MOV.W @ (disp, PC), R1 ADD R1, R0 ..... .DATA.W H'1234	将数据扩展（符号扩展）为 32 位，R1 为 H'00001234。 然后用 ADD 指令进行运算。	ADD.W #H'1234, R0

【注】 用 @ (disp, PC) 参照立即数。

#### (4) 装入 - 存储结构

在寄存器之间进行基本运算。将数据装入寄存器后，和存储器进行运算（装入 - 存储结构）。但是，AND 等的位操作指令直接对存储器进行运算。

#### (5) 延迟转移

无条件转移指令是延迟转移指令。在延迟转移指令的情况下，先执行紧接着延迟转移指令之后的指令，然后进行转移。因此，减少了转移时的流水线混乱。条件转移指令有延迟转移指令和通常转移指令 2 种。

表 2.3 延迟转移指令

本 LSI 的 CPU	说 明	其他 CPU 的例子
BRA TRGET ADD R1, R0	在转移到 TRGET 前执行 ADD。	ADD.W R1, R0 BRA TRGET

#### (6) 乘法 / 乘加运算

以 1 ~ 2 个状态执行  $16 \times 16 \rightarrow 32$  的乘法运算，以 2 ~ 3 个状态执行  $16 \times 16 + 64 \rightarrow 64$  的乘加运算，以 2 ~ 4 个状态执行  $32 \times 32 \rightarrow 64$  的乘法运算和  $32 \times 32 + 64 \rightarrow 64$  的乘加运算。



## (7) T 位

比较结果反映在 SR 的 T 位，根据真伪进行条件转移。只用最小限度的指令改变 T 位，提高了处理速度。

表 2.4 T 位

本 LSI 的 CPU	说 明	其他 CPU 的例子
CMP/GE R1, R0 BT TRGET0 BF TRGET1	当 $R0 \geq R1$ 时，T 位被置位。 当 $R0 \geq R1$ 时，转移到 TRGET0。 当 $R0 < R1$ 时，转移到 TRGET1。	CMP.W R1, R0 BGE TRGET0 BLT TRGET1
ADD #1, R0 CMP/EQ #0, R0 BT TRGET	ADD 不改变 T 位。 当 $R0=0$ 时，T 位被置位。 当 $R0=0$ 时，进行转移。	SUB.W #1, R0 BEQ TRGET

## (8) 立即数

字节（8 位）的立即数分配在指令码中，字和长字的立即数不分配在指令码中而分配在存储器的表中。通过带位移量的 PC 相对寻址方式的立即数传送指令（MOV），参照存储器的表。

表 2.5 立即数的参照

区 分	本 LSI 的 CPU	其他 CPU 的例子
8 位立即数	MOV #H'12, R0	MOV.B #H'12, R0
16 位立即数	MOV.W @(disp, PC), R0 ..... .DATA.W H'1234	MOV.W #H'1234, R0
32 位立即数	MOV.L @(disp, PC), R0 ..... .DATA.L H'12345678	MOV.L #H'12345678, R0

【注】 用@(disp, PC) 参照立即数。

## (9) 绝对地址

在用绝对地址参照数据时，预先将绝对地址的值分配到存储器的表中。在执行指令时，使用立即数的装入方法将该值传送到寄存器，并通过寄存器间接寻址方式参照数据。

表 2.6 绝对地址的参照

区 分	本 LSI 的 CPU	其他 CPU 的例子
绝对地址	MOV.L @(disp, PC), R1 MOV.B @R1, R0 ..... .DATA.L H'12345678	MOV.B @H'12345678, R0

【注】 用@(disp, PC) 参照立即数。

## (10) 16 位 /32 位的位移量

在用 16 位或者 32 位的位移量参照数据时，预先将位移量的值分配到存储器的表中。在执行指令时，使用立即数的装入方法将该值传送到寄存器，并通过带变址寄存器间接寻址方式参照数据。

表 2.7 位移量的参照

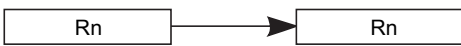
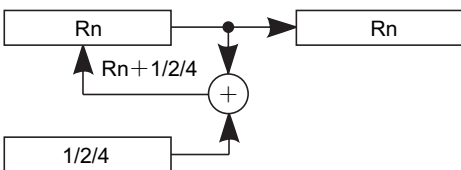
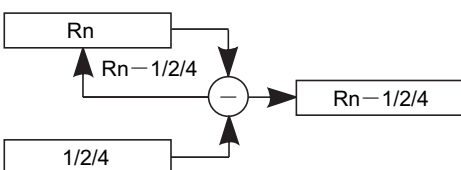
区分	本 LSI 的 CPU	其他 CPU 的例子
16 位的位移量	MOV.W @ (disp, PC), R0 MOV.W @(R0, R1), R2 ..... .DATA.W H'1234	MOV.W @(H'1234, R1), R2

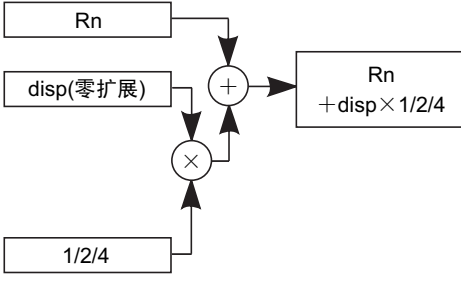
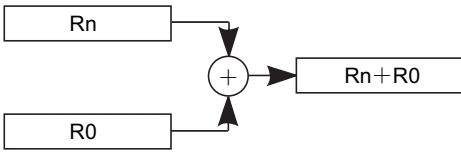
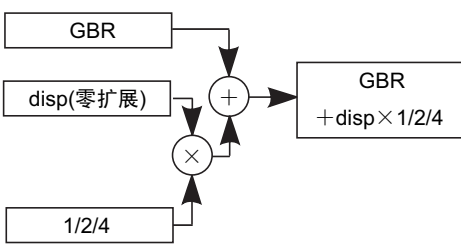
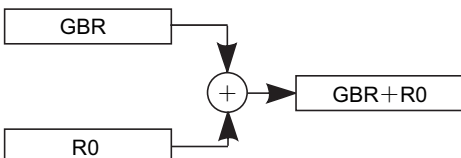
【注】 用@ (disp, PC) 参照立即数。

## 2.4.2 寻址方式

寻址方式和有效地址的计算方法如表 2.8 所示。

表 2.8 寻址方式和有效地址

寻址方式	指令格式	有效地址的计算方法	计算式
寄存器直接	Rn	有效地址为寄存器 Rn。 (操作数为寄存器 Rn 的内容)	—
寄存器间接	@Rn	有效地址为寄存器 Rn 的内容。 	Rn
后增寄存器间接	@Rn+	有效地址为寄存器 Rn 的内容。在执行指令后 Rn 加上常数。常数在操作数长度为字节时是 1，为字时是 2，为长字时是 4。 	执行指令后 字节: $Rn + 1 \rightarrow Rn$ 字: $Rn + 2 \rightarrow Rn$ 长字: $Rn + 4 \rightarrow Rn$
先减寄存器间接	@-Rn	有效地址为先减去常数后的寄存器 Rn 的内容。常数在操作数长度为字节时是 1，为字时是 2，为长字时是 4。 	字节: $Rn - 1 \rightarrow Rn$ 字: $Rn - 2 \rightarrow Rn$ 长字: $Rn - 4 \rightarrow Rn$ (通过计算后的 Rn 执行指令)

寻址方式	指令格式	有效地址的计算方法	计算式
带位移量的寄存器间接	@(disp:4, Rn)	有效地址为寄存器 Rn 加上 4 位位移量 disp 后的内容。disp 在进行零扩展后，操作数长度为字节时乘 1，为字时乘 2，为长字时乘 4。 	字节: $Rn + disp$ 字: $Rn + disp \times 2$ 长字: $Rn + disp \times 4$
带变址的寄存器间接	@(R0, Rn)	有效地址为寄存器 Rn 加上 R0 后的内容。 	$Rn + R0$
带位移量的 GBR 间接	@(disp: 8, GBR)	有效地址为寄存器 GBR 加上 8 位位移量 disp 后的内容。disp 在进行零扩展后，操作数长度为字节时乘 1，为字时乘 2，为长字时乘 4。 	字节: $GBR + disp$ 字: $GBR + disp \times 2$ 长字: $GBR + disp \times 4$
带变址的 GBR 间接	@(R0, GBR)	有效地址为寄存器 GBR 加上 R0 后的内容。 	$GBR + R0$

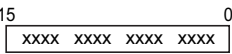
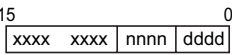
寻址方式	指令格式	有效地址的计算方法	计算式
带位移量的 PC 相对	@(disp: 8, PC)	<p>有效地址为寄存器 PC 加上 8 位位移量 disp 后的内容。disp 在进行零扩展后，操作数长度为字时乘 2，为长字时乘 4。另外，操作数长度为长字时，屏蔽 PC 的低 2 位。</p> <p style="text-align: center;">*在长字时</p>	<p>字: <math>PC + disp \times 2</math>            长字: <math>PC \&amp; H'FFFFFFFC + disp \times 4</math></p>
PC 相对	disp: 8	<p>有效地址为寄存器 PC 加上 8 位位移量 disp（符号扩展后的 disp）乘 2 后的内容。</p>	$PC + disp \times 2$
	disp: 12	<p>有效地址为寄存器 PC 加上 12 位位移量 disp（符号扩展后的 disp）乘 2 后的内容。</p>	$PC + disp \times 2$
	Rn	<p>执行地址为寄存器 PC 加上 Rn 后的内容。</p>	$PC + Rn$
立即数	#imm:8	TST、AND、OR、XOR 指令的 8 位立即数 imm 进行零扩展。	—
	#imm:8	MOV、ADD、CMP/EQ 指令的 8 位立即数 imm 进行符号扩展。	—
	#imm:8	TRAPA 指令的 8 位立即数 imm 进行零扩展后乘 4。	—

## 2.4.3 指令格式

本章节说明指令格式、以及源操作数和目标操作数。操作数的含义因指令码而不同。符号如下：

xxxx: 指令码  
 mmmm: 源寄存器  
 nnnn: 目标寄存器  
 iiiii: 立即数  
 dddd: 位移量

表 2.9 指令格式

指令格式	源操作数	目标操作数	指令例子
0 格式 	—	—	NOP
n 格式 	—	nnnn: 寄存器直接	MOVT Rn
	控制寄存器 或者系统寄存器	nnnn: 寄存器直接	STS MACH, Rn
	控制寄存器 或者系统寄存器	nnnn: 先减寄存器间接	STC.L SR, @-Rn
m 格式 	mmmm: 寄存器直接	控制寄存器 或者系统寄存器	LDC Rm, SR
	mmmm: 后增寄存器间接	控制寄存器 或者系统寄存器	LDC.L @Rm+, SR
	mmmm: 寄存器间接	—	JMP @Rm
	mmmm: 使用 Rm 的 PC 相对	—	BRAF Rm
nm 格式 	mmmm: 寄存器直接	nnnn: 寄存器直接	ADD Rm, Rn
	mmmm: 寄存器直接	nnnn: 寄存器间接	MOV.L Rm, @Rn
	mmmm: 后增寄存器间接 (乘加运算) nnnn: * 后增寄存器间接 (乘加运算)	MACH、MACL	MAC.W @Rm+, @Rn+
	mmmm: 后增寄存器间接	nnnn: 寄存器直接	MOV.L @Rm+, Rn
	mmmm: 寄存器直接	nnnn: 先减寄存器间接	MOV.L Rm, @-Rn
	mmmm: 寄存器直接	nnnn: 带变址的寄存器 间接	MOV.L Rm, @(R0, Rn)
	mmmm: 寄存器直接	nnnn: 带变址的寄存器 间接	MOV.L Rm, @(R0, Rn)
md 格式 	mmmmddd: 带位移量的寄存器间接	R0 (寄存器直接)	MOV.B @(disp, Rm), R0
nd4 格式 	R0 (寄存器直接)	nnnnddd: 带位移量的寄存器间接	MOV.B R0, @(disp, Rn)

指令格式	源操作数	目标操作数	指令例子
nmd 格式 <div style="text-align: center;"> <math>15 \qquad \qquad \qquad 0</math>  <span style="border: 1px solid black; padding: 2px;">xxxx   nnnn   mmmm   dddd</span> </div>	mmmm: 寄存器直接	nnnnddd: 带位移量的寄存器间接	MOV.L Rm, @(disp, Rn)
	mmmmddd: 带位移量的寄存器间接	nnnn: 寄存器直接	MOV.L @(disp, Rm), Rn
d 格式 <div style="text-align: center;"> <math>15 \qquad \qquad \qquad 0</math>  <span style="border: 1px solid black; padding: 2px;">xxxx   xxxx   dddd   dddd</span> </div>	ddddddd: 带位移量的 GBR 间接	R0 (寄存器直接)	MOV.L @(disp, GBR), R0
	R0 (寄存器直接)	ddddddd: 带位移量的 GBR 间接	MOV.L R0, @(disp, GBR)
	ddddddd: 带位移量的 PC 相对	R0 (寄存器直接)	MOVA @(disp, PC), R0
	—	ddddddd: PC 相对	BF label
d12 格式 <div style="text-align: center;"> <math>15 \qquad \qquad \qquad 0</math>  <span style="border: 1px solid black; padding: 2px;">xxxx   dddd   dddd   dddd</span> </div>	—	ddddddddddd: PC 相对	BRA label (label=disp+pc)
nd8 格式 <div style="text-align: center;"> <math>15 \qquad \qquad \qquad 0</math>  <span style="border: 1px solid black; padding: 2px;">xxxx   nnnn   dddd   dddd</span> </div>	ddddddd: 带位移量的 PC 相对	nnnn: 寄存器直接	MOV.L @(disp, PC), Rn
i 格式 <div style="text-align: center;"> <math>15 \qquad \qquad \qquad 0</math>  <span style="border: 1px solid black; padding: 2px;">xxxx   xxxx   iiii   iiii</span> </div>	iiiiiii: 立即数	带变址的 GBR 间接	AND.B #imm, @(R0, GBR)
	iiiiiii: 立即数	R0 (寄存器直接)	AND #imm, R0
	iiiiiii: 立即数	—	TRAPA #imm
ni 格式 <div style="text-align: center;"> <math>15 \qquad \qquad \qquad 0</math>  <span style="border: 1px solid black; padding: 2px;">xxxx   nnnn   iiii   iiii</span> </div>	iiiiiii: 立即数	nnnn: 寄存器直接	ADD #imm, Rn

【注】 \* 在乘加指令中， nnnn 为源寄存器。

## 2.5 指令系统

### 2.5.1 指令系统的分类

指令的分类如表 2.10 所示。

表 2.10 指令的分类

分类	指令的种类	操作码	功 能	指令数
数据传送指令	5	MOV	传送数据 传送立即数 传送外围模块数据 传送结构体数据	39
		MOVA	传送有效地址	
		MOV T	传送 T 位	
		SWAP	交换高位和低位	
		XTRCT	抽出连接寄存器的中间部分	
算术运算指令	21	ADD	2 进制加法	33
		ADDC	带进位的 2 进制加法	
		ADDV	带上溢的 2 进制加法	
		CMP/cond	比较	
		DIV1	除法	
		DIV0S	带符号的除法初始化	
		DIV0U	无符号的除法初始化	
		DMULS	带符号的双精度乘法	
		DMULU	无符号的双精度乘法	
		DT	递减和测试	
		EXTS	符号扩展	
		EXTU	零扩展	
		MAC	乘加运算、双精度乘加运算	
		MUL	双精度乘法	
		MULS	带符号乘法	
		MULU	无符号乘法	
		NEG	符号取反	
		NEGC	带借位的符号取反	
		SUB	2 进制减法	
SUBC	带借位的 2 进制减法			
SUBV	带下溢的 2 进制减法			

分类	指令的种类	操作码	功 能	指令数
逻辑运算指令	6	AND	逻辑与运算	14
		NOT	位取反	
		OR	逻辑或运算	
		TAS	存储器测试和置位	
		TST	逻辑与运算的 T 位置位	
		XOR	逻辑异或运算	
移位指令	10	ROTL	左周期 1 位	14
		ROTR	右周期 1 位	
		ROTCL	带 T 位的左周期 1 位	
		ROTCR	带 T 位的右周期 1 位	
		SHAL	算术左移 1 位	
		SHAR	算术右移 1 位	
		SHLL	逻辑左移 1 位	
		SHLLn	逻辑左移 n 位	
		SHLR	逻辑右移 1 位	
		SHLRn	逻辑右移 n 位	
转移指令	9	BF	条件转移、带延迟的条件转移 (当 T=0 时转移)	11
		BT	条件转移、带延迟的条件转移 (当 T=1 时转移)	
		BRA	无条件转移	
		BRAF	无条件转移	
		BSR	转移到子程序过程	
		BSRF	转移到子程序过程	
		JMP	无条件转移	
		JSR	转移到子程序过程	
RTS	从子程序过程返回			



分类	指令的种类	操作码	功能	指令数
系统控制指令	11	CLRT	清除 T 位	31
		CLRMAC	清除 MAC 寄存器	
		LDC	装入到控制寄存器	
		LDS	装入到系统寄存器	
		NOP	无操作	
		RTE	从异常处理返回	
		SETT	T 位的置位	
		SLEEP	转移到低功耗状态	
		STC	保存控制寄存器的内容	
		STS	保存系统寄存器的内容	
		TRAPA	陷阱异常处理	
	计 62			142

通过以下格式按照分类顺序说明指令的指令码、操作和执行状态：

指令	指令码	操作概略	执行状态	T 位
用助记符表示。	按照 MSB ←→ LSB 的顺序表示。	表示操作的概略。	是无等待时的值。 *1	表示执行指令后的 T 位的值。
符号说明	符号说明	符号说明		符号说明
OP.Sz SRC, DEST	m m m m: 源寄存器	→、←: 传送方向		—: 不变
OP: 操作码	n n n n: 目标寄存器	(xx): 存储器操作数		
Sz: 长度	0000: R0	M/Q/T: SR 内的标志位		
SRC: 源	0001: R1	&: 每位的逻辑与		
DEST: 目标	.....	: 每位的逻辑或		
Rm: 源寄存器	1111: R15	^: 每位的逻辑异或		
Rn: 目标寄存器	i i i i: 立即数	~: 每位的逻辑否定		
imm: 立即数	d d d d: 位移量	<<n: 左移 n 位		
disp: 位移量 *2		>>n: 右移 n 位		

【注】 \*1 有关指令的执行状态

表中所示的执行状态是最小值。实际上根据以下条件，指令的执行状态数将会增加：

- (1) 当取指令和数据存取发生竞争时
- (2) 当装入指令（存储器 → 寄存器）的目标寄存器和紧接着的指令所使用的寄存器相同时

\*2 根据指令的操作数长度等进行倍增（×1、×2、×4）。

详细内容请参照《SH-1/SH-2/SH-DSP 软件手册》。

## 2.5.2 数据传送指令

表 2.11 数据传送指令

指令	指令码	操作	执行状态	T 位
MOV #imm, Rn	1110nnnniiiiiii	#imm→符号扩展→Rn	1	—
MOV.W @(disp, PC), Rn	1001nnnnddddddd	(disp×2+PC)→符号扩展→Rn	1	—
MOV.L @(disp, PC), Rn	1101nnnnddddddd	(disp×4+PC)→Rn	1	—
MOV Rm, Rn	0110nnnnmmmm0011	Rm→Rn	1	—
MOV.B Rm, @Rn	0010nnnnmmmm0000	Rm→(Rn)	1	—
MOV.W Rm, @Rn	0010nnnnmmmm0001	Rm→(Rn)	1	—
MOV.L Rm, @Rn	0010nnnnmmmm0010	Rm→(Rn)	1	—
MOV.B @Rm, Rn	0110nnnnmmmm0000	(Rm)→符号扩展→Rn	1	—
MOV.W @Rm, Rn	0110nnnnmmmm0001	(Rm)→符号扩展→Rn	1	—
MOV.L @Rm, Rn	0110nnnnmmmm0010	(Rm)→Rn	1	—
MOV.B Rm, @-Rn	0010nnnnmmmm0100	Rn-1→Rn, Rm→(Rn)	1	—
MOV.W Rm, @-Rn	0010nnnnmmmm0101	Rn-2→Rn, Rm→(Rn)	1	—
MOV.L Rm, @-Rn	0010nnnnmmmm0110	Rn-4→Rn, Rm→(Rn)	1	—
MOV.B @Rm+, Rn	0110nnnnmmmm0100	(Rm)→符号扩展→Rn, Rm+1→Rm	1	—
MOV.W @Rm+, Rn	0110nnnnmmmm0101	(Rm)→符号扩展→Rn, Rm+2→Rm	1	—
MOV.L @Rm+, Rn	0110nnnnmmmm0110	(Rm)→Rn, Rm+4→Rm	1	—
MOV.B R0, @(disp, Rn)	10000000nnnndddd	R0→(disp+Rn)	1	—
MOV.W R0, @(disp, Rn)	10000001nnnndddd	R0→(disp×2+Rn)	1	—
MOV.L Rm, @(disp, Rn)	0001nnnnmmmmdddd	Rm→(disp×4+Rn)	1	—
MOV.B @(disp, Rm), R0	10000100mmmmdddd	(disp+Rm)→符号扩展→R0	1	—
MOV.W @(disp, Rm), R0	10000101mmmmdddd	(disp×2+Rm)→符号扩展→R0	1	—
MOV.L @(disp, Rm), Rn	0101nnnnmmmmdddd	(disp×4+Rm)→Rn	1	—
MOV.B Rm, @(R0, Rn)	0000nnnnmmmm0100	Rm→(R0+Rn)	1	—
MOV.W Rm, @(R0, Rn)	0000nnnnmmmm0101	Rm→(R0+Rn)	1	—
MOV.L Rm, @(R0, Rn)	0000nnnnmmmm0110	Rm→(R0+Rn)	1	—
MOV.B @(R0, Rm), Rn	0000nnnnmmmm1100	(R0+Rm)→符号扩展→Rn	1	—
MOV.W @(R0, Rm), Rn	0000nnnnmmmm1101	(R0+Rm)→符号扩展→Rn	1	—
MOV.L @(R0, Rm), Rn	0000nnnnmmmm1110	(R0+Rm)→Rn	1	—
MOV.B R0, @(disp, GBR)	11000000dddddddd	R0→(disp+GBR)	1	—
MOV.W R0, @(disp, GBR)	11000001dddddddd	R0→(disp×2+GBR)	1	—
MOV.L R0, @(disp, GBR)	11000010dddddddd	R0→(disp×4+GBR)	1	—
MOV.B @(disp, GBR), R0	11000100dddddddd	(disp+GBR)→符号扩展→R0	1	—
MOV.W @(disp, GBR), R0	11000101dddddddd	(disp×2+GBR)→符号扩展→R0	1	—
MOV.L @(disp, GBR), R0	11000110dddddddd	(disp×4+GBR)→R0	1	—
MOVA @(disp, PC), R0	11000111dddddddd	disp×4+PC→R0	1	—

指令	指令码	操作	执行状态	T 位
MOVT Rn	0000nnnn00101001	T→Rn	1	—
SWAP.B Rm, Rn	0110nnnnmmmm1000	Rm→ 交换低位 2 字节的高低字节 →Rn	1	—
SWAP.W Rm, Rn	0110nnnnmmmm1001	Rm→ 交换高低字 →Rn	1	—
XTRCT Rm, Rn	0010nnnnmmmm1101	Rm: Rn 中间 32 位 →Rn	1	—

### 2.5.3 算术运算指令

表 2.12 算术运算指令

指令	指令码	操作	执行状态	T 位
ADD Rm, Rn	0011nnnnmmmm1100	Rn+Rm→Rn	1	—
ADD #imm, Rn	0111nnnniiiiiii	Rn+imm→Rn	1	—
ADDC Rm, Rn	0011nnnnmmmm1110	Rn+Rm+T→Rn, 进位 →T	1	进位
ADDV Rm, Rn	0011nnnnmmmm1111	Rn+Rm→Rn, 上溢 →T	1	上溢
CMP/EQ #imm, R0	10001000iiiiiii	当 R0=imm 时, 1→T	1	比较结果
CMP/EQ Rm, Rn	0011nnnnmmmm0000	当 Rn=Rm 时, 1→T	1	比较结果
CMP/HS Rm, Rn	0011nnnnmmmm0010	当无符号 Rn ≥ Rm 时, 1→T	1	比较结果
CMP/GE Rm, Rn	0011nnnnmmmm0011	当带符号 Rn ≥ Rm 时, 1→T	1	比较结果
CMP/HI Rm, Rn	0011nnnnmmmm0110	当无符号 Rn > Rm 时, 1→T	1	比较结果
CMP/GT Rm, Rn	0011nnnnmmmm0111	当带符号 Rn > Rm 时, 1→T	1	比较结果
CMP/PL Rn	0100nnnn00010101	当 Rn > 0 时, 1→T	1	比较结果
CMP/PZ Rn	0100nnnn00010001	当 Rn ≥ 0 时, 1→T	1	比较结果
CMP/STR Rm, Rn	0010nnnnmmmm1100	当任意字节相等时, 1→T	1	比较结果
DIV1 Rm, Rn	0011nnnnmmmm0100	单步除法 (Rn÷Rm)	1	计算结果
DIV0S Rm, Rn	0010nnnnmmmm0111	Rn 的 MSB→Q, Rm 的 MSB→M, M^Q→T	1	计算结果
DIV0U	0000000000011001	0→M/Q/T	1	0
DMULS.L Rm, Rn	0011nnnnmmmm1101	带符号 Rn×Rm→MACH 和 MACL 32×32→64 位	2 ~ 5*	—
DMULU.L Rm, Rn	0011nnnnmmmm0101	无符号 Rn×Rm→MACH 和 MACL 32×32→64 位	2 ~ 5*	—
DT Rn	0100nnnn00010000	Rn-1→Rn 当 Rn 为 0 时, 1→T 当 Rn 不为 0 时, 0→T	1	比较结果
EXTS.B Rm, Rn	0110nnnnmmmm1110	将 Rm 的字节进行符号扩展 →Rn	1	—
EXTS.W Rm, Rn	0110nnnnmmmm1111	将 Rm 的字进行符号扩展 →Rn	1	—
EXTU.B Rm, Rn	0110nnnnmmmm1100	将 Rm 的字节进行零扩展 →Rn	1	—
EXTU.W Rm, Rn	0110nnnnmmmm1101	将 Rm 的字进行零扩展 →Rn	1	—
MAC.L @Rm+, @Rn+	0000nnnnmmmm1111	带符号 (Rn)×(Rm)+MAC→MAC 32×32+64→64 位	2 ~ 5*	—

指令	指令码	操作	执行状态	T 位
MAC.W @Rm+, @Rn+	0100nnnnmmmm1111	带符号 (Rn)×(Rm)+MAC→MAC 16×16+64→64 位	2 ~ 4*	—
MUL.L Rm, Rn	0000nnnnmmmm0111	Rn×Rm→MACL 32×32→32 位	2 ~ 5*	—
MULS.W Rm, Rn	0010nnnnmmmm1111	带符号 Rn×Rm→MACL 16×16→32 位	1 ~ 3*	—
MULU.W Rm, Rn	0010nnnnmmmm1110	无符号 Rn×Rm→MACL 16×16→32 位	1 ~ 3*	—
NEG Rm, Rn	0110nnnnmmmm1011	0-Rm→Rn	1	—
NEGC Rm, Rn	0110nnnnmmmm1010	0-Rm-T→Rn, 借位 →T	1	借位
SUB Rm, Rn	0011nnnnmmmm1000	Rn-Rm→Rn	1	—
SUBC Rm, Rn	0011nnnnmmmm1010	Rn-Rm-T→Rn, 借位 →T	1	借位
SUBV Rm, Rn	0011nnnnmmmm1011	Rn-Rm→Rn, 下溢 →T	1	下溢

【注】 \* 表示通常的执行状态。

## 2.5.4 逻辑运算指令

表 2.13 逻辑运算指令

指令	指令码	操作	执行状态	T 位
AND Rm, Rn	0010nnnnmmmm1001	Rn&Rm→Rn	1	—
AND #imm, R0	11001001iiiiiii	R0&imm→R0	1	—
AND.B #imm, @(R0, GBR)	11001101iiiiiii	(R0+GBR)&imm→(R0+GBR)	3	—
NOT Rm, Rn	0110nnnnmmmm0111	~Rm→Rn	1	—
OR Rm, Rn	0010nnnnmmmm1011	Rn   Rm→Rn	1	—
OR #imm, R0	11001011iiiiiii	R0   imm→R0	1	—
OR.B #imm, @(R0, GBR)	11001111iiiiiii	(R0+GBR)   imm→(R0+GBR)	3	—
TAS.B @Rn	0100nnnn00011011	当 (Rn) 为 0 时, 1→T、 1→MSB of (Rn)	4	测试结果
TST Rm, Rn	0010nnnnmmmm1000	当 Rn&Rm 的结果为 0 时, 1→T	1	测试结果
TST #imm, R0	11001000iiiiiii	当 R0&imm 的结果为 0 时, 1→T	1	测试结果
TST.B #imm, @(R0, GBR)	11001100iiiiiii	当 (R0+GBR)&imm 的结果为 0 时, 1→T	3	测试结果
XOR Rm, Rn	0010nnnnmmmm1010	Rn^Rm→Rn	1	—
XOR #imm, R0	11001010iiiiiii	R0^imm→R0	1	—
XOR.B #imm, @(R0, GBR)	11001110iiiiiii	(R0+GBR)^imm→(R0+GBR)	3	—

## 2.5.5 移位指令

表 2.14 移位指令

指令	指令码	操作	执行状态	T 位
ROTL Rn	0100nnnn00000100	$T \leftarrow Rn \leftarrow MSB$	1	MSB
ROTR Rn	0100nnnn00000101	$LSB \rightarrow Rn \rightarrow T$	1	LSB
ROTCL Rn	0100nnnn00100100	$T \leftarrow Rn \leftarrow T$	1	MSB
ROTCLR Rn	0100nnnn00100101	$T \rightarrow Rn \rightarrow T$	1	LSB
SHAL Rn	0100nnnn00100000	$T \leftarrow Rn \leftarrow 0$	1	MSB
SHAR Rn	0100nnnn00100001	$MSB \rightarrow Rn \rightarrow T$	1	LSB
SHLL Rn	0100nnnn00000000	$T \leftarrow Rn \leftarrow 0$	1	MSB
SHLR Rn	0100nnnn00000001	$0 \rightarrow Rn \rightarrow T$	1	LSB
SHLL2 Rn	0100nnnn00001000	$Rn \ll 2 \rightarrow Rn$	1	—
SHLR2 Rn	0100nnnn00001001	$Rn \gg 2 \rightarrow Rn$	1	—
SHLL8 Rn	0100nnnn00011000	$Rn \ll 8 \rightarrow Rn$	1	—
SHLR8 Rn	0100nnnn00011001	$Rn \gg 8 \rightarrow Rn$	1	—
SHLL16 Rn	0100nnnn00101000	$Rn \ll 16 \rightarrow Rn$	1	—
SHLR16 Rn	0100nnnn00101001	$Rn \gg 16 \rightarrow Rn$	1	—

## 2.5.6 转移指令

表 2.15 转移指令

指令	指令码	操作	执行状态	T 位
BF label	10001011ddddddd	当 T=0 时, $disp \times 2 + PC \rightarrow PC$ 当 T=1 时, nop	3/1*	—
BF/S label	10001111ddddddd	延迟转移 当 T=0 时, $disp \times 2 + PC \rightarrow PC$ 当 T=1 时, nop	2/1*	—
BT label	10001001ddddddd	当 T=1 时, $disp \times 2 + PC \rightarrow PC$ 当 T=0 时, nop	3/1*	—
BT/S label	10001101ddddddd	延迟转移 当 T=1 时, $disp \times 2 + PC \rightarrow PC$ 当 T=0 时, nop	2/1*	—
BRA label	1010ddddddddddd	延迟转移 $disp \times 2 + PC \rightarrow PC$	2	—
BRAF Rm	0000mmmm00100011	延迟转移 $Rm + PC \rightarrow PC$	2	—
BSR label	1011ddddddddddd	延迟转移 $PC \rightarrow PR, disp \times 2 + PC \rightarrow PC$	2	—
BSRF Rm	0000mmmm00000011	延迟转移 $PC \rightarrow PR, Rm + PC \rightarrow PC$	2	—

指令	指令码	操作	执行状态	T 位
JMP @Rm	0100mmmm00101011	延迟转移 Rm→PC	2	—
JSR @Rm	0100mmmm00001011	延迟转移 PC→PR, Rm→PC	2	—
RTS	0000000000001011	延迟转移 PR→PC	2	—

【注】 \* 在不转移时，为 1 个状态。

## 2.5.7 系统控制指令

表 2.16 系统控制指令

指令	指令码	操作	执行状态	T 位
CLRT	0000000000001000	0→T	1	0
CLRMACH	0000000000101000	0→MACH 和 MACL	1	—
LDC Rm, SR	0100mmmm00001110	Rm→SR	6	LSB
LDC Rm, GBR	0100mmmm00011110	Rm→GBR	4	—
LDC Rm, VBR	0100mmmm00101110	Rm→VBR	4	—
LDC.L @Rm+, SR	0100mmmm00000111	(Rm)→SR, Rm+4→Rm	8	LSB
LDC.L @Rm+, GBR	0100mmmm00010111	(Rm)→GBR, Rm+4→Rm	4	—
LDC.L @Rm+, VBR	0100mmmm00100111	(Rm)→VBR, Rm+4→Rm	4	—
LDS Rm, MACH	0100mmmm00001010	Rm→MACH	1	—
LDS Rm, MACL	0100mmmm00011010	Rm→MACL	1	—
LDS Rm, PR	0100mmmm00101010	Rm→PR	1	—
LDS.L @Rm+, MACH	0100mmmm00000110	(Rm)→MACH, Rm+4→Rm	1	—
LDS.L @Rm+, MACL	0100mmmm00010110	(Rm)→MACL, Rm+4→Rm	1	—
LDS.L @Rm+, PR	0100mmmm00100110	(Rm)→PR, Rm+4→Rm	1	—
NOP	0000000000001001	无操作	1	—
RTE	0000000000101011	延迟转移 堆栈区 →PC/SR	5	—
SETT	000000000011000	1→T	1	1
SLEEP	000000000011011	睡眠	4*	—
STC SR, Rn	0000nnnn00000010	SR→Rn	1	—
STC GBR, Rn	0000nnnn00010010	GBR→Rn	1	—
STC VBR, Rn	0000nnnn00100010	VBR→Rn	1	—
STC.L SR, @- Rn	0100nnnn00000011	Rn-4→Rn, SR→(Rn)	1	—
STC.L GBR, @-Rn	0100nnnn00010011	Rn-4→Rn, GBR→(Rn)	1	—
STC.L VBR, @-Rn	0100nnnn00100011	Rn-4→Rn, VBR→(Rn)	1	—
STS MACH, Rn	0000nnnn00001010	MACH→Rn	1	—
STS MACL, Rn	0000nnnn00011010	MACL→Rn	1	—

指令	指令码	操作	执行状态	T 位	
STS	PR, Rn	0000nmmn00101010	PR→Rn	1	—
STS.L	MACH, @-Rn	0100nmmn00000010	Rn-4→Rn, MACH→(Rn)	1	—
STS.L	MACL, @-Rn	0100nmmn00010010	Rn-4→Rn, MACL→(Rn)	1	—
STS.L	PR, @-Rn	0100nmmn00100010	Rn-4→Rn, PR→(Rn)	1	—
TRAPA	#imm	11000011iiiiiiii	PC/SR→堆栈区, (imm×4+VBR)→PC	8	—

【注】 \* 是转移到睡眠状态前的状态数。

有关指令的执行状态

表中所示的执行状态为最小值。实际上根据以下条件，指令的执行状态将会增加：

- (1) 当发生取指令和数据存取的竞争时
- (2) 当装入指令（存储器→寄存器）的目标寄存器和紧接着的指令所使用的寄存器相同时

## 2.6 处理状态

CPU 的处理状态有复位状态、异常处理状态、总线权释放状态、程序执行状态和低功耗状态共 5 种。状态之间的转移如图 2.4 所示。

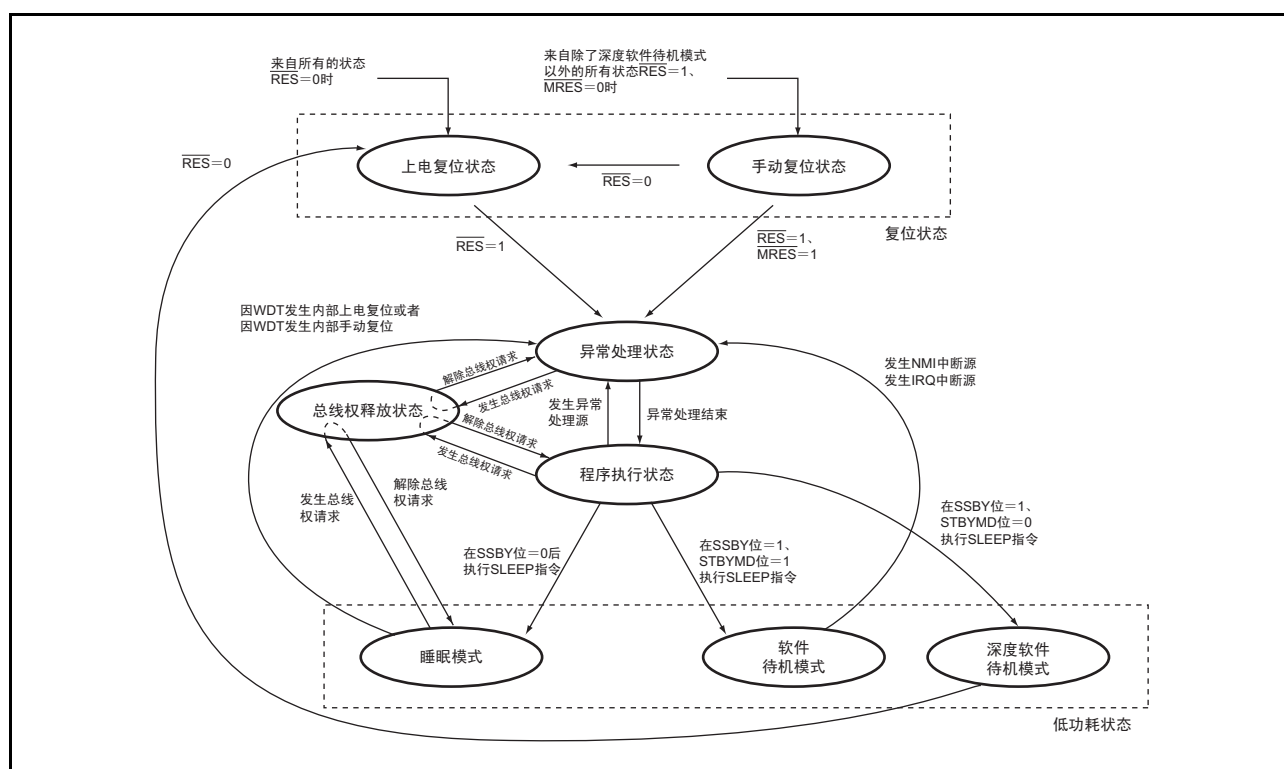


图 2.4 处理状态的转移图

### (1) 复位状态

这是 CPU 被复位的状态。当  $\overline{\text{RES}}$  引脚为低电平时，进入上电复位状态；当  $\overline{\text{RES}}$  引脚为高电平并且  $\overline{\text{MRES}}$  引脚为低电平时，进入手动复位状态。

### (2) 异常处理状态

异常处理状态是 CPU 根据复位和中断等异常处理源改变处理状态流程时的过渡状态。

在复位时，从异常处理向量表中分别取出并分别保存作为程序计数器（PC）初始值的执行起始地址和堆栈指针（SP）的初始值，转移到起始地址后开始执行程序。

在发生中断时，参照 SP 后将 PC 和状态寄存器（SR）保存到堆栈区。从异常处理向量表中取出异常服务程序的起始地址，转移到该地址后开始执行程序。

以后的处理状态为程序执行状态。

### (3) 程序执行状态

这是 CPU 依次执行程序的状态。

### (4) 低功耗状态

这是 CPU 停止运行并处于低功耗的状态。通过睡眠指令进入睡眠模式、软件待机模式或者深度软件待机模式。

### (5) 总线权释放状态

这是对请求总线权的芯片，CPU 释放总线的状态。



## 第 3 章 MCU 运行模式

### 3.1 运行模式的选择

本 LSI 有 4 种 MCU 运行模式和 3 种用于内部闪存编程的模式。

通过组合 FWE 引脚、MD1 引脚和 MD0 引脚，设定运行模式。

本 LSI 能设定的运行模式和这些引脚的组合如表 3.1 所示，请不要设定此表中没有的组合。

另外，在接通系统电源时，必须进行上电复位处理。

MCU 运行模式有 MCU 扩展模式 0 ~ 2 和单芯片模式。

用于内部闪存编程的模式有引导模式（单板编程模式）、用户引导模式和用户编程模式。

表 3.1 运行模式的选择

MCU 运行模式	引脚设定			模式	内部 ROM	CS0 空间的总线宽度			
	FWE	MD1	MD0			SH7083	SH7084	SH7085	SH7086
模式 0	0	0	0	MCU 扩展模式 0	无效	8	8	16	16
模式 1	0	0	1	MCU 扩展模式 1	无效	16	16	32	32
模式 2	0	1	0	MCU 扩展模式 2	有效	通过 BSC 的 CS0BCR 设定			
模式 3	0	1	1	单芯片模式	有效	—			
模式 4*	1	0	0	引导模式	有效	—			
模式 5*	1	0	1	用户引导模式	有效	通过 BSC 的 CS0BCR 设定			
模式 6*	1	1	0	用户编程模式	有效	通过 BSC 的 CS0BCR 设定			
模式 7*	1	1	1		有效	—			

【注】 \* 闪存的编程模式。

### 3.2 输入 / 输出引脚

运行模式相关的引脚结构如表 3.2 所示。

表 3.2 引脚结构

名称	输入 / 输出	功 能
MD0	输入	指定运行模式
MD1	输入	指定运行模式
FWE	输入	用于内部闪存编程 / 擦除的硬件允许引脚

### 3.3 各运行模式的说明

#### 3.3.1 模式 0（MCU 扩展模式 0）

在模式 0 中，CS0 空间的总线宽度为外部存储器空间的 8 位（SH7083/84）或者 16 位（SH7085/86）总线。

#### 3.3.2 模式 1（MCU 扩展模式 1）

在模式 1 中，CS0 空间的总线宽度为外部存储器空间的 16 位（SH7083/84）或者 32 位（SH7085/86）总线。

#### 3.3.3 模式 2（MCU 扩展模式 2）

在模式 2 中，内部 ROM 有效，能使用 CS0 空间。

#### 3.3.4 模式 3（单芯片模式）

在单芯片模式中能使用全部端口，但是不能使用外部地址。

### 3.4 地址映像

各运行模式的地址映像如图 3.1 ~ 图 3.7 所示。

	模式0、1 【内部ROM无效】		模式2 【内部ROM有效】		模式3 【单芯片】
H'00000000	CS0空间	H'00000000	内部ROM (256KB)	H'00000000	内部ROM (256KB)
H'01FFFFFF H'02000000		H'0003FFFF H'00040000		H'0003FFFF H'00040000	
	保留	H'01FFFFFF H'02000000	保留		保留
H'0BFFFFFF H'0C000000	CS3空间	H'03FFFFFF H'04000000	CS0空间		
H'0DFFFFFF H'0E000000	保留	H'0BFFFFFF H'0C000000	保留		
		H'0DFFFFFF H'0E000000	CS3空间		
H'1BFFFFFF H'1C000000	CS7空间	H'1BFFFFFF H'1C000000	保留		
H'1DFFFFFF H'1E000000	保留	H'1DFFFFFF H'1E000000	CS7空间		
			保留		
H'FFF7FFFF H'FFF80000	SDRAM模式的设定	H'FFF7FFFF H'FFF80000	保留		
H'FFF9FFFF H'FFFA0000	保留	H'FFF9FFFF H'FFFA0000	SDRAM模式的设定		
H'FFF7FFF H'FFF8000	内部RAM (16KB)	H'FFF7FFF H'FFF8000	保留	H'FFF7FFF H'FFF8000	
H'FFFBFFF H'FFFC000	保留	H'FFFBFFF H'FFFC000	内部RAM (16KB)	H'FFFBFFF H'FFFC000	
H'FFFFFFF	外围I/O	H'FFFFFFF	外围I/O	H'FFFFFFF	

图 3.1 SH7083 (闪存 256KB 版) 的各运行模式的地址映像

模式0、1 【内部ROM无效】		模式2 【内部ROM有效】		模式3 【单芯片】	
H'00000000	CS0空间	H'00000000	内部ROM (512KB)	H'00000000	内部ROM (512KB)
H'01FFFFFF H'02000000	保留	H'0007FFFF H'00080000	保留	H'0007FFFF H'00080000	保留
H'0BFFFFFF H'0C000000	CS3空间	H'01FFFFFF H'02000000	CS0空间		
H'0BFFFFFF H'0E000000	保留	H'03FFFFFF H'04000000	保留		
H'1BFFFFFF H'1C000000	CS7空间	H'0BFFFFFF H'0C000000	CS3空间		
H'1DFFFFFF H'1E000000	保留	H'0FFFFFFF H'10000000	保留		
H'FFF7FFFF H'FFF80000	SDRAM模式的设定	H'1BFFFFFF H'1C000000	CS7空间		
H'FFF9FFFF H'FFFA0000	保留	H1FFFFFF H'20000000	保留		
H'FFF3FFF H'FFF4000	内部RAM (32KB)	H'FFF7FFFF H'FFF80000	SDRAM模式的设定		
H'FFFBFFF H'FFFC000	外围I/O	H'FFF9FFFF H'FFFA0000	保留		
H'FFFFFFF		H'FFF3FFF H'FFF4000	内部RAM (32KB)	H'FFF3FFF H'FFF4000	
		H'FFFBFFF H'FFFC000	外围I/O	H'FFFBFFF H'FFFC000	外围I/O
		H'FFFFFFF		H'FFFFFFF	

图 3.2 SH7083（闪存 512KB 版）的各运行模式的地址映像

模式0、1 【内部ROM无效】		模式2 【内部ROM有效】		模式3 【单芯片】	
H'00000000	CS0空间	H'00000000	内部ROM (256KB)	H'00000000	内部ROM (256KB)
		H'0003FFFF H'00040000	保留	H'0003FFFF H'00040000	
		H'01FFFFFF H'02000000	CS0空间		
H'03FFFFFF H'04000000	CS1空间	H'03FFFFFF H'04000000	CS1空间		
H'07FFFFFF H'08000000	CS2空间	H'07FFFFFF H'08000000	CS2空间		
H'0BFFFFFF H'0C000000	CS3空间	H'0BFFFFFF H'0C000000	CS3空间		
H'0FFFFFFF H'10000000	CS4空间	H'0FFFFFFF H'10000000	CS4空间		
H'13FFFFFF H'14000000	CS5空间	H'13FFFFFF H'14000000	CS5空间		
H'17FFFFFF H'18000000	CS6空间	H'17FFFFFF H'18000000	CS6空间		保留
H'1BFFFFFF H'1C000000	CS7空间	H'1BFFFFFF H'1C000000	CS7空间		
H'1FFFFFFF H'20000000	保留	H'1FFFFFFF H'20000000	保留		
H'FFF7FFFF H'FFF80000	SDRAM模式的设定	H'FFF7FFFF H'FFF80000	SDRAM模式的设定		
H'FFF9FFFF H'FFFA0000	保留	H'FFF9FFFF H'FFFA0000	保留		
H'FFF7FFF H'FFF8000	内部RAM (16KB)	H'FFF7FFF H'FFF8000	内部RAM (16KB)	H'FFF7FFF H'FFF8000	内部RAM (16KB)
H'FFFBFFF H'FFFC000	外围I/O	H'FFFBFFF H'FFFC000	外围I/O	H'FFFBFFF H'FFFC000	外围I/O
H'FFFFFFF		H'FFFFFFF		H'FFFFFFF	

图 3.3 SH7084 (闪存 256KB 版) 的各模式的地址映像

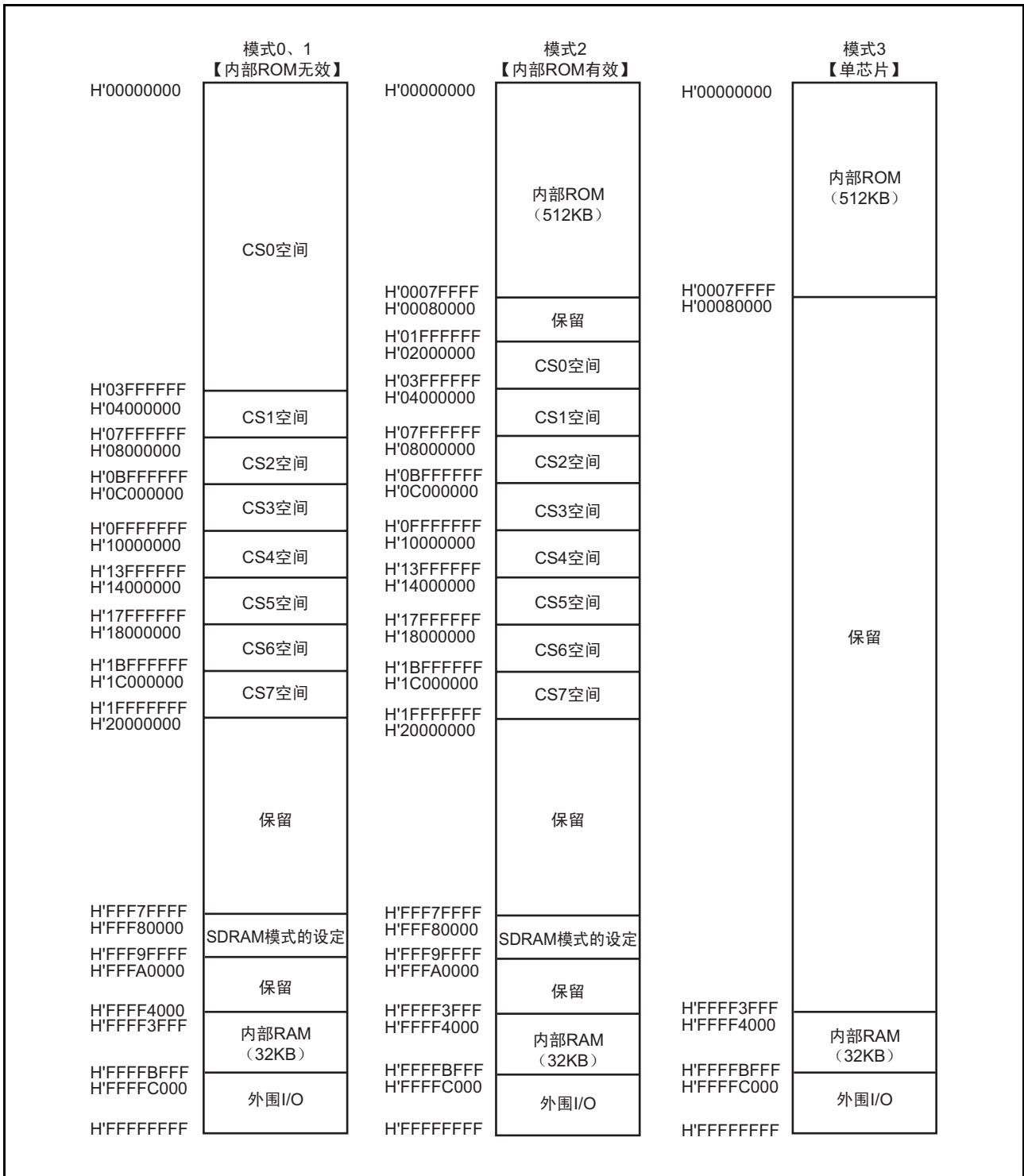


图 3.4 SH7084（闪存 512KB 版）的各模式的地址映像

模式0、1 【内部ROM无效】		模式2 【内部ROM有效】		模式3 【单芯片】	
H'00000000	CS0空间	H'00000000	内部ROM (256KB)	H'00000000	内部ROM (256KB)
H'03FFFFFF H'04000000		H'0003FFFF H'00040000		保留	
H'07FFFFFF H'08000000		CS1空间	H'01FFFFFF H'02000000	CS0空间	保留
H'0BFFFFFF H'0C000000		CS2空间	H'03FFFFFF H'04000000	CS1空间	
H'0FFFFFFF H'10000000		CS3空间	H'07FFFFFF H'08000000	CS2空间	
H'13FFFFFF H'14000000		CS4空间	H'0BFFFFFF H'0C000000	CS3空间	
H'17FFFFFF H'18000000		CS5空间	H'0FFFFFFF H'10000000	CS4空间	
H'1BFFFFFF H'1C000000	CS6空间	H'13FFFFFF H'14000000	CS5空间		
H1FFFFFFF H'20000000	CS7空间	H'17FFFFFF H'18000000	CS6空间		
	保留	H'1BFFFFFF H'1C000000	CS7空间		
		H1FFFFFFF H'20000000	保留		
H'FFF7FFF H'FFF8000	SDRAM模式的设定	H'FFF7FFF H'FFF8000	SDRAM模式的设定		
H'FFF9FFF H'FFFA000	保留	H'FFF9FFF H'FFFA000	保留		
H'FFFF7FFF H'FFFF8000	内部RAM (16KB)	H'FFFF7FFF H'FFFF8000	内部RAM (16KB)	H'FFFF7FFF H'FFFF8000	内部RAM (16KB)
H'FFFFBFFF H'FFFFC000	外围I/O	H'FFFFBFFF H'FFFFC000	外围I/O	H'FFFFBFFF H'FFFFC000	外围I/O
H'FFFFFFF		H'FFFFFFF		H'FFFFFFF	

图 3.5 SH7085（闪存 256KB 版）的各模式的地址映像

模式0、1 【内部ROM无效】		模式2 【内部ROM有效】		模式3 【单芯片】	
H'00000000	CS0空间	H'00000000	内部ROM (512KB)	H'00000000	内部ROM (512KB)
		H'0007FFFF H'00080000	保留	H'0007FFFF H'00080000	
H'03FFFFFF H'04000000	CS1空间	H'01FFFFFF H'02000000	CS0空间		
H'07FFFFFF H'08000000	CS2空间	H'03FFFFFF H'04000000	CS1空间		
H'0BFFFFFF H'0C000000	CS3空间	H'07FFFFFF H'08000000	CS2空间		
H'0FFFFFFF H'10000000	CS4空间	H'0BFFFFFF H'0C000000	CS3空间		
H'13FFFFFF H'14000000	CS5空间	H'0FFFFFFF H'10000000	CS4空间		
H'17FFFFFF H'18000000	CS6空间	H'13FFFFFF H'14000000	CS5空间		
H'1BFFFFFF H'1C000000	CS7空间	H'17FFFFFF H'18000000	CS6空间		保留
H'1FFFFFFF H'20000000	保留	H'1BFFFFFF H'1C000000	CS7空间		
		H'1FFFFFFF H'20000000	保留		
H'FFF7FFFF H'FFF80000	SDRAM模式的设定	H'FFF7FFFF H'FFF80000	SDRAM模式的设定		
H'FFF9FFFF H'FFFA0000	保留	H'FFF9FFFF H'FFFA0000	保留		
H'FFF3FFF H'FFF4000	内部RAM (32KB)	H'FFF3FFF H'FFF4000	内部RAM (32KB)	H'FFF3FFF H'FFF4000	内部RAM (32KB)
H'FFFBFFF H'FFFC000	外围I/O	H'FFFBFFF H'FFFC000	外围I/O	H'FFFBFFF H'FFFC000	外围I/O
H'FFFFFFF		H'FFFFFFF		H'FFFFFFF	

图 3.6 SH7085 (闪存 512KB 版) 的各模式的地址映像



模式0、1 【内部ROM无效】		模式2 【内部ROM有效】		模式3 【单芯片】		
H'00000000	CS0空间	H'00000000	内部ROM (512KB)	H'00000000	内部ROM (512KB)	
			H'0007FFFF H'00080000	保留	H'0007FFFF H'00080000	保留
			H'01FFFFFF H'02000000	CS0空间		
H'03FFFFFF H'04000000		CS1空间	H'03FFFFFF H'04000000	CS1空间		
H'07FFFFFF H'08000000		CS2空间	H'07FFFFFF H'08000000	CS2空间		
H'0BFFFFFF H'0C000000		CS3空间	H'0BFFFFFF H'0C000000	CS3空间		
H'0FFFFFFF H'10000000		CS4空间	H'0FFFFFFF H'10000000	CS4空间		
H'13FFFFFF H'14000000		CS5空间	H'13FFFFFF H'14000000	CS5空间		
H'17FFFFFF H'18000000		CS6空间	H'17FFFFFF H'18000000	CS6空间		
H'1BFFFFFF H'1C000000		CS7空间	H'1BFFFFFF H'1C000000	CS7空间		
H'1FFFFFFF H'20000000		保留	H'1FFFFFFF H'20000000	保留		
H'3FFFFFFF H'40000000		CS8空间	H'3FFFFFFF H'40000000	CS8空间		
H'7FFFFFFF H'80000000		保留	H'7FFFFFFF H'80000000	保留		
H'FFF7FFFF H'FFF80000		SDRAM模式的设定	H'FFF7FFFF H'FFF80000	SDRAM模式的设定		
H'FFF9FFFF H'FFFA0000		保留	H'FFF9FFFF H'FFFA0000	保留		
H'FFF3FFF H'FFF4000		内部RAM (32KB)	H'FFF3FFF H'FFF4000	内部RAM (32KB)	H'FFF3FFF H'FFF4000	
H'FFFBFFF H'FFFC000	外围I/O	H'FFFBFFF H'FFFC000	外围I/O	H'FFFBFFF H'FFFC000	外围I/O	
H'FFFFFFF		H'FFFFFFF		H'FFFFFFF		

图 3.7 SH7086 的各模式的地址映像

### 3.5 本 LSI 的初始状态

为了降低功耗，本 LSI 在初始状态下将一部分内部模块设定为模块待机状态。因此，要使这些模块运行时，必须解除模块待机状态。详细内容请参照“第 26 章 低功耗模式”。

### 3.6 更改运行模式时的注意事项

如果在给本 LSI 外加电源时要更改运行模式，就必须在上电复位状态（给  $\overline{\text{RES}}$  引脚外加低电平）下进行。

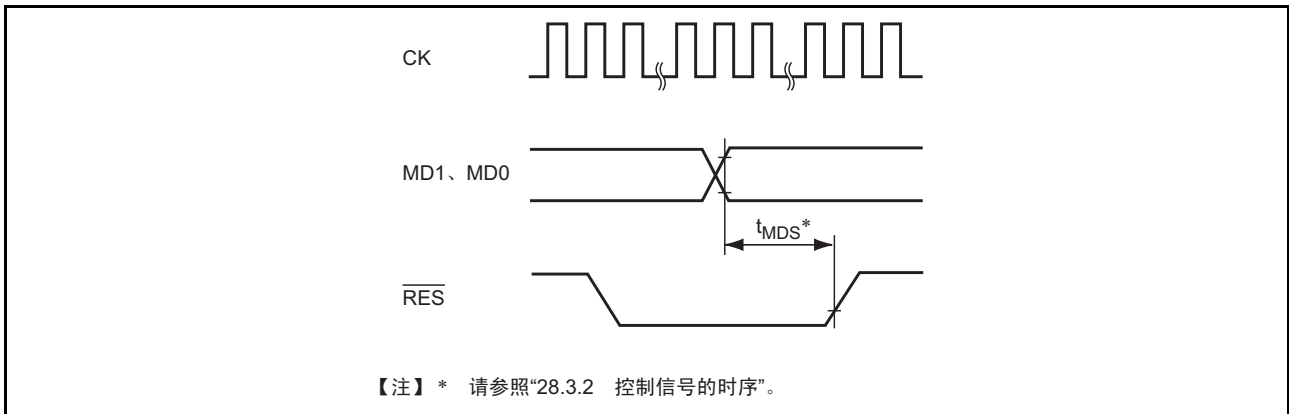


图 3.8 更改运行模式时的复位输入时序

## 第 4 章 时钟振荡器 (CPG)

本 LSI 内置时钟振荡器 (CPG)。

CPG 生成内部时钟 (I $\phi$ )、总线时钟 (B $\phi$ )、外围时钟 (P $\phi$ ) 以及用于 MTU2S 和 MTU2 模块的时钟 (MI $\phi$  和 MP $\phi$ ) 并进行低功耗模式的控制。

### 4.1 特点

- 能独立生成 5 种时钟  
能独立生成用于 CPU 的内部时钟 (I $\phi$ )、用于外围模块的外围时钟 (P $\phi$ )、用于内部总线的总线时钟 (B $\phi$ =CK)、用于内部 MTU2S 模块的 MTU2S 时钟 (MI $\phi$ ) 以及用于内部 MTU2 模块的 MTU2 时钟 (MP $\phi$ )。
- 频率的变更功能  
能通过 CPG 的内部分频电路独立更改内部时钟 (I $\phi$ )、总线时钟 (B $\phi$ )、外围时钟 (P $\phi$ )、MTU2S 时钟 (MI $\phi$ ) 和 MTU2 时钟 (MP $\phi$ ) 的频率。根据频率控制寄存器 (FRQCR) 的设定, 通过软件更改频率。
- 低功耗模式的控制  
能停止睡眠模式和待机模式中的时钟以及模块待机功能中的特定模块。
- 振荡停止的检测功能  
如果因某种理由而停止从时钟输入引脚提供时钟, 定时器引脚就能自动进入高阻抗状态。

时钟振荡器的框图如图 4.1 所示。

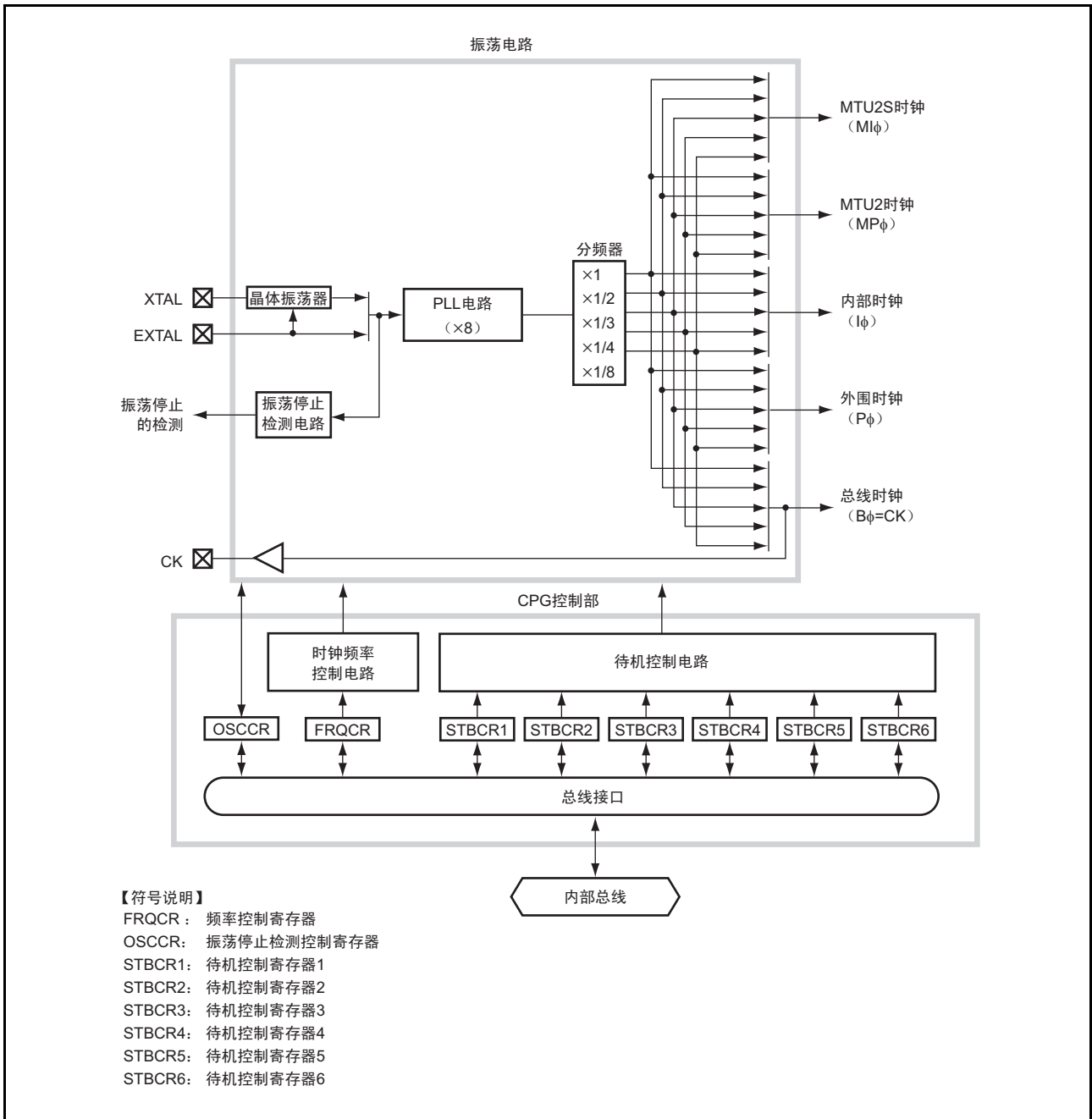


图 4.1 时钟振荡器的框图

时钟振荡器的各模块的功能如下：

(1) PLL 电路

PLL 电路有将晶体振荡器或者 EXTAL 引脚输入的时钟频率递增 8 倍的功能。倍增率总是固定为 8 倍。

(2) 晶体振荡器

这是将晶体谐振器连接到 XTAL、EXTAL 引脚时的振荡电路。

(3) 分频器

分频器生成内部时钟 ( $I\phi$ )、总线时钟 ( $B\phi$ )、外围时钟 ( $P\phi$ ) 以及 MTU2S 时钟 ( $MI\phi$ ) 和 MTU2 时钟 ( $MP\phi$ ) 使用的运行频率时钟。

运行频率可选择 PLL 电路的输出频率的 1 倍、1/2 倍、1/3 倍、1/4 倍或者 1/8 倍。

通过频率控制寄存器 (FRQCR) 设定分频比。

(4) 振荡停止的检测电路

这是检测晶体谐振器异常的电路。

(5) 时钟频率的控制电路

时钟频率的控制电路通过设定频率控制寄存器 (FRQCR) 来控制时钟的频率。

(6) 待机控制电路

待机控制电路控制睡眠 / 待机模式中的内部振荡电路和其他模块的状态。

(7) 频率控制寄存器 (FRQCR)

频率控制寄存器 (FRQCR) 中分配了内部时钟 ( $I\phi$ )、总线时钟 ( $B\phi$ )、外围时钟 ( $P\phi$ ) 以及 MTU2S 时钟 ( $MI\phi$ ) 和 MTU2 时钟 ( $MP\phi$ ) 频率的各分频比控制位。

(8) 振荡停止检测控制寄存器 (OSCCR)

振荡停止检测控制寄存器 (OSCCR) 中分配了振荡停止检测标志位和外部引脚的标志输出选择位。

(9) 待机控制寄存器 1 ~ 6 (STBCR1 ~ STBCR6)

待机控制寄存器 (STBCR) 中分配了低功耗模式的各控制位。有关待机控制寄存器请参照“第 26 章 低功耗模式”。

各模块的运行时钟如表 4.1 所示。

表 4.1 各模块的运行时钟

运行时钟	对应模块	运行时钟	对应模块	
内部时钟 (I $\phi$ )	CPU	外围时钟 (P $\phi$ )	POE	
	UBC		SCI	
ROM	SCIF			
RAM	SSU			
	I <sup>2</sup> C2			
	A/D			
	CMT			
	WDT			
总线时钟 (B $\phi$ )	BSC		MTU2 时钟 (MP $\phi$ )	MTU2
	DMAC		MTU2S 时钟 (MI $\phi$ )	MTU2S
	DTC			

## 4.2 输入 / 输出引脚

CPG 的引脚结构和功能如表 4.2 所示。

表 4.2 时钟振荡器的引脚结构和功能

名称	引脚名	输入 / 输出	功 能
晶体谐振器输入 / 输出引脚 (时钟输入引脚)	XTAL	输出	连接晶体谐振器。
	EXTAL	输入	连接晶体谐振器, 或者用作外部时钟的输入引脚。
时钟输出引脚	CK	输出	用作外部时钟的输出引脚。

【注】 在使用时钟输出引脚 (CK) 时, 需要根据引脚功能控制器 (PFC) 进行引脚的设定。有关 PFC 的设定, 请参照 “第 21 章 引脚功能控制器 (PFC)”。

## 4.3 时钟运行模式

本 LSI 的时钟运行模式如表 4.3 所示。

表 4.3 时钟运行模式

模式	时钟输入 / 输出		PLL 电路	分频器的输入
	供给源	输出		
1	EXTAL 输入 晶体谐振器	CK*	ON (×8)	×8

【注】 \* 从 CK 引脚输出时钟时, 需要设定 PFC。有关 PFC 的设定, 请参照 “第 21 章 引脚功能控制器 (PFC)”。

• 模式 1

因为从 EXTAL 引脚输入外部时钟，通过 PLL 电路将频率递增 8 倍后提供给本 LSI 内部，所以降低了外部生成的时钟频率。能使用的输入时钟频率为 5MHz 到 12.5MHz，内部时钟 (I $\phi$ ) 的频率范围为 10MHz 到 80MHz。

最大运行频率：I $\phi$ =80MHz、B $\phi$ =40MHz、P $\phi$ =40MHz、MI $\phi$ =80MHz、MP $\phi$ =40MHz

能通过 FRQCR 设定的分频比如表 4.4 所示。

表 4.4 由 FRQCR 设定的分频比

PLL 倍增率	FRQCR 的分频比设定					时钟比					时钟频率 (MHz) *					
	I $\phi$	B $\phi$	P $\phi$	MI $\phi$	MP $\phi$	I $\phi$	B $\phi$	P $\phi$	MI $\phi$	MP $\phi$	输入时钟	I $\phi$	B $\phi$	P $\phi$	MI $\phi$	MP $\phi$
×8	1/8	1/8	1/8	1/8	1/8	1	1	1	1	1	10	10	10	10	10	10
	1/4	1/8	1/8	1/8	1/8	2	1	1	1	1		20	10	10	10	10
	1/4	1/8	1/8	1/4	1/8	2	1	1	2	1		20	10	10	20	10
	1/4	1/4	1/8	1/8	1/8	2	2	1	1	1		20	20	10	10	10
	1/4	1/4	1/8	1/4	1/8	2	2	1	2	1		20	20	10	20	10
	1/4	1/4	1/8	1/4	1/4	2	2	1	2	2		20	20	10	20	20
	1/4	1/4	1/4	1/4	1/4	2	2	2	2	2		20	20	20	20	20
	1/3	1/3	1/3	1/3	1/3	8/3	8/3	8/3	8/3	8/3		26	26	26	26	26
	1/2	1/8	1/8	1/8	1/8	4	1	1	1	1		40	10	10	10	10
	1/2	1/8	1/8	1/4	1/8	4	1	1	2	1		40	10	10	20	10
	1/2	1/8	1/8	1/2	1/8	4	1	1	4	1		40	10	10	40	10
	1/2	1/4	1/8	1/8	1/8	4	2	1	1	1		40	20	10	10	10
	1/2	1/4	1/8	1/4	1/8	4	2	1	2	1		40	20	10	20	10
	1/2	1/4	1/8	1/4	1/4	4	2	1	2	2		40	20	10	20	20
	1/2	1/4	1/8	1/2	1/8	4	2	1	4	1		40	20	10	40	10
	1/2	1/4	1/8	1/2	1/4	4	2	1	4	2		40	20	10	40	20
	1/2	1/4	1/4	1/4	1/4	4	2	2	2	2		40	20	20	20	20
	1/2	1/4	1/4	1/2	1/4	4	2	2	4	2		40	20	20	40	20
	1/2	1/2	1/8	1/8	1/8	4	4	1	1	1		40	40	10	10	10
	1/2	1/2	1/8	1/4	1/8	4	4	1	2	1		40	40	10	20	10
	1/2	1/2	1/8	1/4	1/4	4	4	1	2	2		40	40	10	20	20
	1/2	1/2	1/8	1/2	1/8	4	4	1	4	1		40	40	10	40	10
	1/2	1/2	1/8	1/2	1/4	4	4	1	4	2		40	40	10	40	20
	1/2	1/2	1/8	1/2	1/2	4	4	1	4	4		40	40	10	40	40
1/2	1/2	1/4	1/4	1/4	4	4	2	2	2	40	40	20	20	20		
1/2	1/2	1/4	1/2	1/4	4	4	2	4	2	40	40	20	40	20		
1/2	1/2	1/4	1/2	1/2	4	4	2	4	4	40	40	20	40	40		
1/1	1/8	1/8	1/8	1/8	8	1	1	1	1	80	10	10	10	10		

PLL 倍增率	FRQCR 的分频比设定					时钟比					时钟频率 (MHz) *					
	l $\phi$	B $\phi$	P $\phi$	Ml $\phi$	MP $\phi$	l $\phi$	B $\phi$	P $\phi$	Ml $\phi$	MP $\phi$	输入时钟	l $\phi$	B $\phi$	P $\phi$	Ml $\phi$	MP $\phi$
×8	1/1	1/8	1/8	1/4	1/8	8	1	1	2	1	10	80	10	10	20	10
	1/1	1/8	1/8	1/2	1/8	8	1	1	4	1		80	10	10	40	10
	1/1	1/8	1/8	1/1	1/8	8	1	1	8	1		80	10	10	80	10
	1/1	1/4	1/8	1/8	1/8	8	2	1	1	1		80	20	10	10	10
	1/1	1/4	1/8	1/4	1/8	8	2	1	2	1		80	20	10	20	10
	1/1	1/4	1/8	1/4	1/4	8	2	1	2	2		80	20	10	20	20
	1/1	1/4	1/8	1/2	1/8	8	2	1	4	1		80	20	10	40	10
	1/1	1/4	1/8	1/2	1/4	8	2	1	4	2		80	20	10	40	20
	1/1	1/4	1/8	1/1	1/8	8	2	1	8	1		80	20	10	80	10
	1/1	1/4	1/8	1/1	1/4	8	2	1	8	2		80	20	10	80	20
	1/1	1/4	1/4	1/4	1/4	8	2	2	2	2		80	20	20	20	20
	1/1	1/4	1/4	1/2	1/4	8	2	2	4	2		80	20	20	40	20
	1/1	1/4	1/4	1/1	1/4	8	2	2	8	2		80	20	20	80	20
	1/1	1/3	1/3	1/3	1/3	8	8/3	8/3	8/3	8/3		80	26	26	26	26
	1/1	1/3	1/3	1/1	1/3	8	8/3	8/3	8	8/3		80	26	26	80	26
	1/1	1/2	1/8	1/8	1/8	8	4	1	1	1		80	40	10	10	10
	1/1	1/2	1/8	1/4	1/8	8	4	1	2	1		80	40	10	20	10
	1/1	1/2	1/8	1/4	1/4	8	4	1	2	2		80	40	10	20	20
	1/1	1/2	1/8	1/2	1/8	8	4	1	4	1		80	40	10	40	10
	1/1	1/2	1/8	1/2	1/4	8	4	1	4	2		80	40	10	40	20
	1/1	1/2	1/8	1/2	1/2	8	4	1	4	4		80	40	10	40	40
	1/1	1/2	1/8	1/1	1/8	8	4	1	8	1		80	40	10	80	10
	1/1	1/2	1/8	1/1	1/4	8	4	1	8	2		80	40	10	80	20
	1/1	1/2	1/8	1/1	1/2	8	4	1	8	4		80	40	10	80	40
	1/1	1/2	1/4	1/4	1/4	8	4	2	2	2		80	40	20	20	20
	1/1	1/2	1/4	1/2	1/4	8	4	2	4	2		80	40	20	40	20
	1/1	1/2	1/4	1/2	1/2	8	4	2	4	4		80	40	20	40	40
	1/1	1/2	1/4	1/1	1/4	8	4	2	8	2		80	40	20	80	20
1/1	1/2	1/4	1/1	1/2	8	4	2	8	4	80	40	20	80	40		
1/1	1/2	1/2	1/2	1/2	8	4	4	4	4	80	40	40	40	40		
1/1	1/2	1/2	1/1	1/2	8	4	4	8	4	80	40	40	80	40		



PLL 倍增率	FRQCR 的分频比设定					时钟比					时钟频率 (MHz) *					
	I $\phi$	B $\phi$	P $\phi$	MI $\phi$	MP $\phi$	I $\phi$	B $\phi$	P $\phi$	MI $\phi$	MP $\phi$	输入时钟	I $\phi$	B $\phi$	P $\phi$	MI $\phi$	MP $\phi$
×8	1/1	1/1	1/4	1/4	1/4	8	8	2	2	2	5	40	40	10	10	10
	1/1	1/1	1/4	1/2	1/4	8	8	2	4	2		40	40	10	20	10
	1/1	1/1	1/4	1/2	1/2	8	8	2	4	4		40	40	10	20	20
	1/1	1/1	1/4	1/1	1/4	8	8	2	8	2		40	40	10	40	10
	1/1	1/1	1/4	1/1	1/2	8	8	2	8	4		40	40	10	40	20
	1/1	1/1	1/4	1/1	1/1	8	8	2	8	8		40	40	10	40	40
	1/1	1/1	1/3	1/3	1/3	8	8	8/3	8/3	8/3		40	40	13	13	13
	1/1	1/1	1/3	1/1	1/3	8	8	8/3	8	8/3		40	40	13	40	13
	1/1	1/1	1/3	1/1	1/1	8	8	8/3	8	8		40	40	13	40	40
	1/1	1/1	1/2	1/2	1/2	8	8	4	4	4		40	40	20	20	20
	1/1	1/1	1/2	1/1	1/2	8	8	4	8	4		40	40	20	40	20
	1/1	1/1	1/2	1/1	1/1	8	8	4	8	8		40	40	20	40	40
1/1	1/1	1/1	1/1	1/1	8	8	8	8	8	40	40	40	40	40		

【注】 \* 时钟频率是假定了输入时钟频率时的值。

- PLL 电路的倍增率只为 ×8。分频器的分频比可以选择 ×1、×1/2、×1/3、×1/4 或者 ×1/8。  
通过频率控制寄存器，给要设定的各时钟设定分频比。
- PLL 电路的输出频率为晶体谐振器或者 EXTAL 引脚输入的时钟频率乘以 PLL 电路的倍增率（8 倍）。
- 分频器的输入总是 PLL 电路的输出。
- 内部时钟（I $\phi$ ）的频率为晶体谐振器或者 EXTAL 引脚输入的时钟频率乘以 PLL 电路的倍增率（8 倍）和分频器的分频比。  
内部时钟（I $\phi$ ）的频率不能超过最大运行频率（80MHz）。
- 总线时钟（B $\phi$ ）的频率为晶体谐振器或者 EXTAL 引脚输入的时钟频率乘以 PLL 电路的倍增率（8 倍）和分频器的分频比。  
总线时钟（B $\phi$ ）的频率不能超过 40MHz 和内部时钟（I $\phi$ ）的频率。
- 外围时钟（P $\phi$ ）的频率为晶体谐振器或者 EXTAL 引脚输入的时钟频率乘以 PLL 电路的倍增率（8 倍）和分频器的分频比。  
外围时钟（P $\phi$ ）的频率不能超过 40MHz 和总线时钟（B $\phi$ ）的频率。
- 在使用 MTU2S 和 MTU2 时，MTU2S 时钟（MI $\phi$ ）的频率不能超过内部时钟（I $\phi$ ）的频率并且不能低于 MTU2 时钟（MP $\phi$ ）的频率，MTU2 时钟（MP $\phi$ ）的频率不能超过 MTU2S 时钟（MI $\phi$ ）和总线时钟（B $\phi$ ）的频率并且不能低于外围时钟（P $\phi$ ）的频率。  
MTU2S 时钟（MI $\phi$ ）和 MTU2 时钟（MP $\phi$ ）的频率为晶体谐振器或者 EXTAL 引脚输入的时钟频率乘以 PLL 电路的倍增率（8 倍）和分频器的分频比。
- CK 引脚的频率总是和总线时钟（B $\phi$ ）的频率相等。

## 4.4 寄存器说明

时钟振荡器有以下寄存器。有关寄存器的地址和各处理模式的寄存器状态，请参照“第 27 章 寄存器一览”。

表 4.5 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
频率控制寄存器	FRQCR	R/W	H'36DB	H'FFFE800	16
振荡停止检测控制寄存器	OSCCR	R/W	H'00	H'FFFE814	8

### 4.4.1 频率控制寄存器 (FRQCR)

FRQCR 是 16 位可读写寄存器，能指定内部时钟 (I $\phi$ )、总线时钟 (B $\phi$ )、外围时钟 (P $\phi$ )、MTU2S 时钟 (MI $\phi$ ) 和 MTU2 时钟 (MP $\phi$ ) 的分频比。FRQCR 只能进行字存取。

只能在上电复位时 (WDT 上溢的上电复位除外) 对此寄存器进行初始化，初始值为 H'36DB。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	IFC[2:0]			BFC[2:0]			PFC[2:0]			MIFC[2:0]			MPFC[2:0]		
初始值:	0	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写的值总是 0。
14 ~ 12	IFC[2:0]	011	R/W	内部时钟 (I $\phi$ ) 的分频比 对于 PLL 电路的输出频率，指定内部时钟 (I $\phi$ ) 的分频比。设定禁止设定的值时，不保证之后的运行。 000: $\times 1$ 倍 001: $\times 1/2$ 倍 010: $\times 1/3$ 倍 011: $\times 1/4$ 倍 100: $\times 1/8$ 倍 其他: 禁止设定
11 ~ 9	BFC[2:0]	011	R/W	总线时钟 (B $\phi$ ) 的分频比 对于 PLL 电路的输出频率，指定总线时钟 (B $\phi$ ) 的分频比。设定禁止设定的值时，不保证之后的运行。 000: $\times 1$ 倍 001: $\times 1/2$ 倍 010: $\times 1/3$ 倍 011: $\times 1/4$ 倍 100: $\times 1/8$ 倍 其他: 禁止设定

位	位名	初始值	R/W	说明
8 ~ 6	PFC[2:0]	011	R/W	外围时钟 (P $\phi$ ) 的分频比 对于 PLL 电路的输出频率, 指定外围时钟 (P $\phi$ ) 的分频比。设定禁止设定的值时, 不保证之后的运行。 000: $\times 1$ 倍 001: $\times 1/2$ 倍 010: $\times 1/3$ 倍 011: $\times 1/4$ 倍 100: $\times 1/8$ 倍 其他: 禁止设定
5 ~ 3	—	011	R/W	MTU2S 时钟 (Ml $\phi$ ) 的分频比 对于 PLL 电路的输出频率, 指定 MTU2S 时钟 (Ml $\phi$ ) 的分频比。设定禁止设定的值时, 不保证之后的运行。 000: $\times 1$ 倍 001: $\times 1/2$ 倍 010: $\times 1/3$ 倍 011: $\times 1/4$ 倍 100: $\times 1/8$ 倍 其他: 禁止设定
2 ~ 0	MPFC[2:0]	011	R/W	MTU2 时钟 (MP $\phi$ ) 的分频比 对于 PLL 电路的输出频率, 指定 MTU2 时钟 (MP $\phi$ ) 的分频比。设定禁止设定的值时, 不保证之后的运行。 000: $\times 1$ 倍 001: $\times 1/2$ 倍 010: $\times 1/3$ 倍 011: $\times 1/4$ 倍 100: $\times 1/8$ 倍 其他: 禁止设定

## 4.4.2 振荡停止检测控制寄存器 (OSCCR)

OSCCR 是 8 位可读写寄存器，选择振荡停止检测标志和外部引脚的标志输出。OSCCR 只能进行字节存取。

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	OSC STOP	-	OSC ERS
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

位	位名	初始值	R/W	说明
7 ~ 3	—	全 0	R	保留位 读写的值总是 0。
2	OSCSTOP	0	R	振荡停止检测标志 [置位条件] <ul style="list-style-type: none"> <li>• 在通常运行中检测到时钟输入已停止时</li> <li>• 已转移到软件待机模式时</li> </ul> [清除条件] <ul style="list-style-type: none"> <li>• 从 RES 引脚进行上电复位时</li> <li>• 从软件待机模式返回时</li> </ul>
1	—	0	R	保留位 读写的值总是 0。
0	OSCERS	0	R/W	选择振荡停止检测标志的输出 选择是否从 $\overline{\text{WDTOVF}}$ 引脚输出振荡停止检测标志信号。 0: 只从 $\overline{\text{WDTOVF}}$ 引脚输出 WDT 上溢信号。 1: 从 $\overline{\text{WDTOVF}}$ 引脚输出 WDT 上溢信号和振荡停止检测标志信号。

## 4.5 频率的变更方法

能通过改变分频器的分频比，更改内部时钟、总线时钟、外围时钟、MTU2S 时钟和 MTU2 时钟的频率。由软件通过频率控制寄存器 (FRQCR) 控制这些频率，方法如下：

1. 在初始状态下， $IFC2 \sim IFC0 = H'011$  ( $\times 1/4$  倍)、 $BFC2 \sim BFC0 = H'011$  ( $\times 1/4$  倍)、 $PFC2 \sim PFC0 = H'011$  ( $\times 1/4$  倍)、 $MIFC2 \sim MIFC0 = H'011$  ( $\times 1/4$  倍)、 $MPFC2 \sim MPFC0 = H'011$  ( $\times 1/4$  倍)。
2. 停止除 CPU、内部 ROM 和内部 RAM 以外的模块。
3. 将  $IFC2 \sim IFC0$ 、 $BFC2 \sim BFC0$ 、 $PFC2 \sim PFC0$ 、 $MIFC2 \sim MIFC0$ 、 $MPFC2 \sim MPFC0$  位设定为目标值。由于 PLL 电路的频率倍增率固定为  $\times 8$  倍，所以只能通过设定分频比决定频率。此时的频率必须设定为内部时钟 ( $I\phi$ )  $\geq$  总线时钟 ( $B\phi$ )  $\geq$  外围时钟 ( $P\phi$ )。另外，在使用 MTU2S 时钟和 MTU2 时钟的情况下，必须设定内部时钟 ( $I\phi$ )  $\geq$  MTU2S 时钟 ( $MI\phi$ )  $\geq$  MTU2 时钟 ( $MP\phi$ )  $\geq$  外围时钟 ( $P\phi$ ) 且总线时钟 ( $B\phi$ )  $\geq$  MTU2 时钟 ( $MP\phi$ )。  
再者，FRQCR 的改写处理必须在内部 ROM 或内部 RAM 上进行程序的执行。
4. 产生 FRQCR 改写指令且经过  $(1 \sim 24n)cyc + 11B\phi + 7P\phi$  时间后进行切换。  
n: 在 FRQCR 的 BFC 位设定的分频率 (1、1/2、1/3、1/4、1/8)  
cyc: 通过 PLL 使 EXTAL 倍增 8 倍的时钟

【注】 (1 ~ 24n) 根据内部状态而改变。

## 4.6 振荡器

提供时钟的方法有连接晶体谐振器和输入外部时钟的方法。

### 4.6.1 连接晶体谐振器的方法

晶体谐振器的连接例子如图 4.2 所示，必须使用表 4.6 所示的阻尼电阻  $R_d$  和频率为 5 ~ 12.5MHz 的晶体谐振器。

有关晶体谐振器和 LSI 的匹配，请与晶体谐振器厂家洽谈。

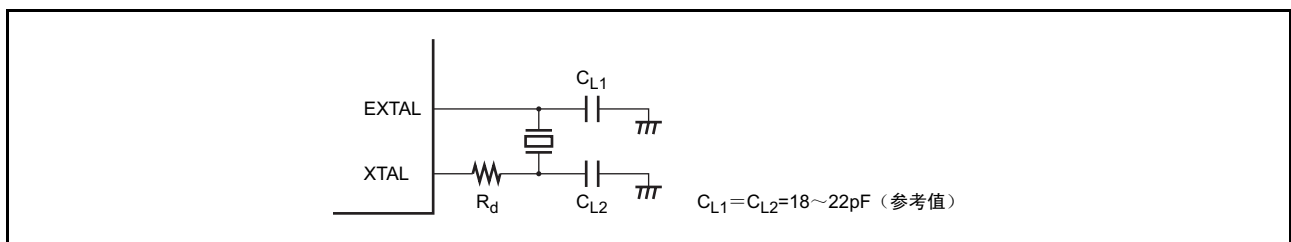


图 4.2 晶体谐振器的连接例子

表 4.6 阻尼电阻值 (参考值)

频率 (MHz)	5	8	10	12.5
$R_d$ ( $\Omega$ ) (参考值)	500	200	0	0

晶体谐振器的等效电路如图 4.3 所示，必须使用如表 4.7 所示特性的晶体谐振器。

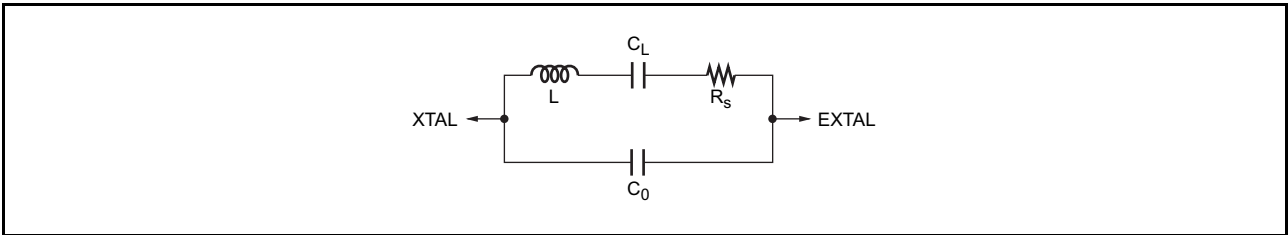


图 4.3 晶体谐振器的等效电路

表 4.7 晶体谐振器的特性

频率 (MHz)	5	8	10	12.5
R <sub>s</sub> Max. (Ω) (参考值)	120	80	60	50
C <sub>0</sub> Max. (pF) (参考值)	7			

#### 4.6.2 输入外部时钟的方法

输入外部时钟的连接例子如图 4.4 所示。如果在软件待机模式中停止外部时钟，就必须输入高电平。在运行时，外部输入时钟的频率必须在 5 ~ 12.5MHz 之间。XTAL 引脚的寄生电容不能大于 10pF。

即使在输入外部时钟的情况下，为了确保 PLL 的稳定时间，在接通电源或者解除软件待机时，等待时间必须至少为振荡稳定时间。

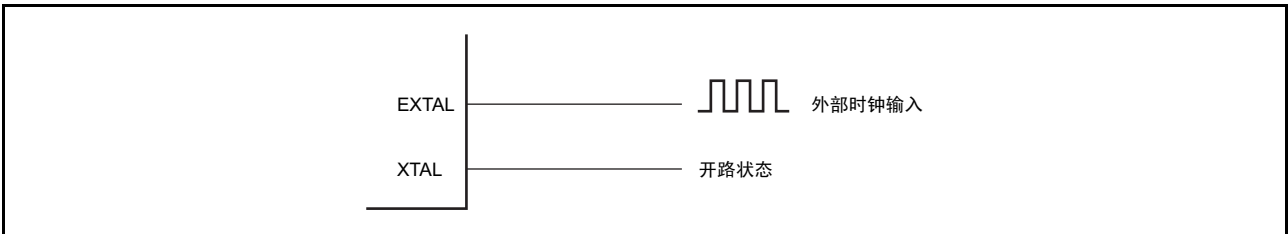


图 4.4 外部时钟的连接例子

## 4.7 振荡停止检测功能

为了防备因某种系统异常而引起振荡器停止的情况，CPG 具有检测时钟停止的功能。

如果检测到 EXTAL 输入在一定的期间内不发生变化，就将 OSCCR 寄存器的 OSCSTOP 位设定为 1，并且将此状态保持到 RES 引脚的上电复位或者解除软件待机模式为止。此时，一旦将 OSCERS 位设定为 1，就从 WDTOVF 引脚输出振荡停止检测标志信号。另外，和 PFC 的设定无关，能将大电流端口（MTU2 的 TIOC3B、TIOC3D、TIOC4A ~ TIOC4D 以及 MTU2S 的 TIOC3BS、TIOC3DS、TIOC4AS ~ TIOC4DS 为多路复用引脚）置为高阻抗。详细内容请参照“21.1.11 大电流端口的控制寄存器（HCPCR）”和“附录 A. 引脚状态”。

即使在软件待机状态下，也能将上述引脚置为高阻抗。详细内容请参照“21.1.11 大电流端口的控制寄存器（HCPCR）”和“附录 A. 引脚状态”。在解除软件待机状态后，变为通常运行状态。另外，在非软件待机状态下振荡异常停止时，LSI 的其他运行不稳定。此时，即使重新开始振荡，包括上述引脚的 LSI 运行也不稳定。

即使在 EXTAL 输入不发生变化的情况下，本 LSI 的 PLL 电路仍在 100kHz ~ 10MHz（因温度和工作电压而发生变动）范围内继续振荡。

## 4.8 使用注意事项

### 4.8.1 有关谐振器的注意事项

因为谐振器的各种特性和用户的电路板设计密切相关，所以请用户参考本章介绍的谐振器的连接例子，在给予充分的评价后使用。因为振荡电路的电路常数因谐振器、安装电路的寄生电容而不同，所以必须和谐振器厂家商谈后作决定。另外，外加在振荡引脚的电压不能超过最大额定值。

### 4.8.2 电路板设计的注意事项

本 LSI 虽然实施了防止辐射噪声的对策，但是如果需要进一步降低辐射噪声，建议使用多层电路板并使用系统接地专用层。

在使用晶体谐振器时，必须尽量将谐振器和负载电容排列在 XTAL 和 EXTAL 引脚附近。如图 4.5 所示，振荡电路附近不能通信号线，否则有可能因电感而不能正常振荡。

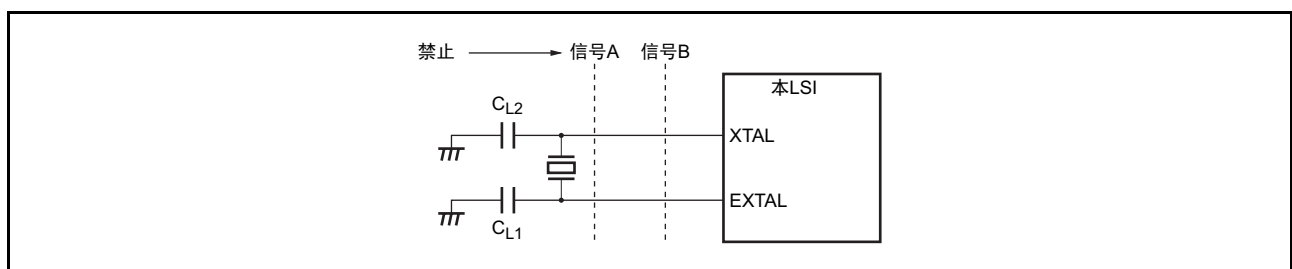


图 4.5 有关振荡电路电路板设计的注意事项

PLL 电路的附接推荐电路如图 4.6 所示，必须将 PLL Vss 和 Vcc、Vss 远离电路板的电源供给源，并且在 Vcc 和 Vss 引脚的附近插入旁路电容 CB 和 CPB。

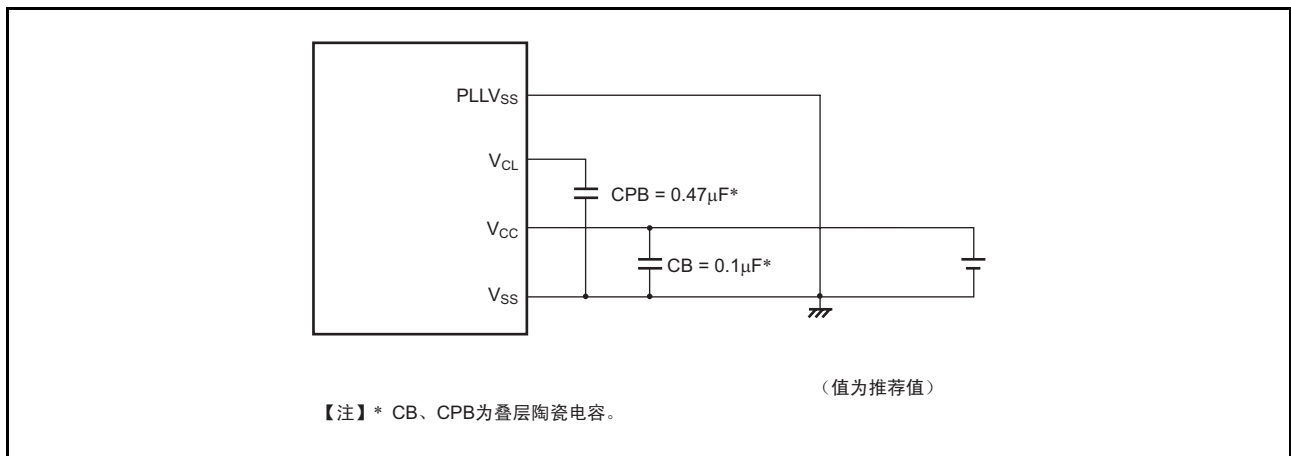


图 4.6 PLL 电路的外接推荐电路



## 第 5 章 异常处理

### 5.1 概要

#### 5.1.1 异常处理的种类和优先级

如表 5.1 所示，根据复位、地址错误、中断和指令的各异常源启动异常处理，异常源设有优先级，如果同时发生多个异常源，就按照此优先级接受处理。

表 5.1 异常源的种类和优先级

	异常处理	优先级
复位	上电复位	高  低
	手动复位	
中断	用户断点（执行指令前的断点）	
地址错误	CPU 地址错误（取指令）	
指令	一般非法指令（未定义的代码）	
	槽非法指令（配置在紧接着延迟转移指令 *1 之后的未定义代码或者 PC 改写指令 *2）	
	陷阱指令（TRAPA 指令）	
地址错误	CPU 地址错误（数据存取）	
中断	用户断点（执行指令后的断点或者操作数断点）	
地址错误	DMAC/DTC 地址错误（数据存取）	
中断	NMI	
	IRQ	
	内部外围模块	

【注】 \*1 延迟转移指令：JMP、JSR、BRA、BSR、RTS、RTE、BF/S、BT/S、BSRF、BRAFL

\*2 PC 改写指令：JMP、JSR、BRA、BSR、RTS、RTE、BT、BF、TRAPA、BF/S、BT/S、BSRF、BRAFL、LDC Rm, SR、LDC.L @Rm+, SR

### 5.1.2 异常处理的运行

在如表 5.2 所示的时序中检测到各异常源后开始处理。

表 5.2 异常源检测和异常处理的开始时序

异常处理		异常源检测和异常处理的开始时序
复位	上电复位	在 $\overline{\text{RES}}$ 引脚从低电平变为高电平或者 WDT 上溢时开始处理
	手动复位	在 $\overline{\text{MRES}}$ 引脚从低电平变为高电平或者 WDT 上溢时开始处理
地址错误		在指令的译码阶段被检测，并在执行中的指令结束后开始处理
中断		
指令	陷阱指令	通过执行 TRAPA 指令开始处理
	一般非法指令	当延迟转移指令（延迟槽）以外的未定义代码被译码时开始处理
	槽非法指令	当配置在紧接着延迟转移指令（延迟槽）之后的未定义代码或者 PC 改写指令被译码时开始处理

当启动异常处理时，CPU 的运行如下：

#### (1) 由复位引起的异常处理

从异常处理向量表中（PC 和 SP 在上电复位时分别为地址 H'00000000 和 H'00000004；在手动复位时分别为地址 H'00000008 和 H'0000000C）取出程序计数器（PC）和堆栈指针（SP）的初始值。有关异常处理向量表，请参照“5.1.3 异常处理向量表”。然后，设定向量基准寄存器（VBR）为 H'00000000，状态寄存器（SR）的中断屏蔽位（I3 ~ I0）为 H'F（B'1111），并从异常处理向量表中取出的 PC 地址开始执行程序。

#### (2) 由地址错误、中断和指令引起的异常处理

将 SR 和 PC 保存到 R15 指向的堆栈。在中断异常处理时，将中断优先级写到 SR 的中断屏蔽位（I3 ~ I0）；在由地址错误或者指令引起的异常处理时，I3 ~ I0 位不受影响。然后，从异常处理向量表中取出起始地址，并从该地址开始执行程序。

### 5.1.3 异常处理向量表

执行异常处理前，需要预先将异常处理向量表设定到存储器，并将异常服务程序的起始地址保存到异常处理向量表中（将 PC 和 SP 的初始值保存到复位异常处理表中）。

分别给各异常源分配了不同的向量号和向量表地址偏移量，从对应的向量号和向量表地址偏移量算出向量表地址。在异常处理中，从此向量表地址指向的异常处理向量表中取出异常服务程序的起始地址。

向量号和向量表地址偏移量、向量表地址的计算方法分别如表 5.3 和表 5.4 所示。

表 5.3 异常处理向量表

异常源		向量号	向量表地址偏移量
上电复位	PC	0	H'00000000 ~ H'00000003
	SP	1	H'00000004 ~ H'00000007
手动复位	PC	2	H'00000008 ~ H'0000000B
	SP	3	H'0000000C ~ H'0000000F
一般非法指令		4	H'00000010 ~ H'00000013
(系统保留)		5	H'00000014 ~ H'00000017
槽非法指令		6	H'00000018 ~ H'0000001B
(系统保留)		7	H'0000001C ~ H'0000001F
		8	H'00000020 ~ H'00000023
CPU 地址错误		9	H'00000024 ~ H'00000027
DMAC/DTC 地址错误		10	H'00000028 ~ H'0000002B
中断	NMI	11	H'0000002C ~ H'0000002F
	用户断点	12	H'00000030 ~ H'00000033
(系统保留)		13	H'00000034 ~ H'00000037
		⋮	⋮
		31	H'0000007C ~ H'0000007F
陷阱指令 (用户向量)		32	H'00000080 ~ H'00000083
		⋮	⋮
		63	H'000000FC ~ H'000000FF
中断	IRQ0	64	H'00000100 ~ H'00000103
	IRQ1	65	H'00000104 ~ H'00000107
	IRQ2	66	H'00000108 ~ H'0000010B
	IRQ3	67	H'0000010C ~ H'0000010F
	IRQ4	68	H'00000110 ~ H'00000113
	IRQ5	69	H'00000114 ~ H'00000117
	IRQ6	70	H'00000118 ~ H'0000011B
	IRQ7	71	H'0000011C ~ H'0000011F
内部外围模块 *		72	H'00000120 ~ H'00000123
		⋮	⋮
		255	H'000003FC ~ H'000003FF

【注】 \* 有关各内部外围模块的中断向量号和向量表偏移量，请参照“第 6 章 中断控制器 (INTC)”的表 6.3。

表 5.4 异常处理向量表地址的计算方法

异常源	向量表地址的计算方法
复位	向量表地址 = (向量表地址偏移量) = (向量号) × 4
地址错误、中断、指令	向量表地址 = VBR + (向量表地址偏移量) = VBR + (向量号) × 4

【注】 VBR: 向量基址寄存器  
向量表地址偏移量: 参照表 5.3  
向量号: 参照表 5.3

## 5.2 复位

### 5.2.1 复位的种类

复位是优先级最高的异常处理源，有上电复位和手动复位 2 种。如表 5.5 所示，在上电复位或者手动复位时，CPU 状态都被初始化。另外，内部外围模块的寄存器在上电复位时被初始化，而在手动复位时不被初始化。

表 5.5 复位状态

种类	转移到复位状态的条件			内部状态		
	$\overline{\text{RES}}$	WDT 上溢	$\overline{\text{MRES}}$	CPU/INTC	内部外围模块	POE、PFC、I/O 端口
上电复位	低电平	—	—	初始化	初始化	初始化
	高电平	上溢	高电平	初始化	初始化	初始化
手动复位	高电平	没有上溢	低电平	初始化	不初始化	不初始化

### 5.2.2 上电复位

#### (1) 由 $\overline{\text{RES}}$ 引脚引起的上电复位

如果将  $\overline{\text{RES}}$  引脚置为低电平，本 LSI 就进入上电复位状态。为了确保本 LSI 的复位， $\overline{\text{RES}}$  引脚的低电平在接通电源或者待机时（在时钟停止时），至少必须保持振荡稳定时间；在时钟运行时，至少保持 20tcyc。在上电复位的状态下，CPU 的内部状态和内部外围模块的寄存器全部被初始化。有关上电复位状态下的各引脚状态，请参照“附录 A. 引脚状态”。

在上电复位的状态下，如果将  $\overline{\text{RES}}$  引脚保持一定期间的低电平后变为高电平，就开始上电复位的异常处理。此时 CPU 的运行如下：

1. 从异常处理向量表中取出程序计数器（PC）的初始值（执行起始地址）。
2. 从异常处理向量表中取出堆栈指针（SP）的初始值。
3. 将向量基址寄存器（VBR）清除为 H'00000000，状态寄存器（SR）的中断屏蔽位（I3 ~ I0）置为 H'F（B'1111）。
4. 分别给 PC 和 SP 设定从异常处理向量表中取出的值，然后开始执行程序。

另外，必须在接通系统电源时进行上电复位处理。

## (2) 由 WDT 引起的上电复位

如果设定为在 WDT 看门狗定时器模式中发生上电复位，就在 WDT 的 WTCNT 发生上溢时进入上电复位状态。

此时，由 WDT 引起的复位不初始化时钟振荡器（CPG）的频率控制寄存器（FRQCR）和看门狗定时器（WDT）的寄存器（这些寄存器只在由  $\overline{\text{RES}}$  引脚引起的上电复位时被初始化）。

另外，如果同时发生由  $\overline{\text{RES}}$  引脚输入信号引起的复位和由 WDT 上溢引起的复位，就优先进行由  $\overline{\text{RES}}$  引脚引起的复位，并将 WTCSR 的 WO VF 位清 0。当开始由 WDT 引起的上电复位处理时，CPU 的运行如下：

1. 从异常处理向量表中取出程序计数器（PC）的初始值（执行起始地址）。
2. 从异常处理向量表中取出堆栈指针（SP）的初始值。
3. 将向量基准寄存器（VBR）清除为 H'00000000，状态寄存器（SR）的中断屏蔽位（I3 ~ I0）置为 HF（B'1111）。
4. 分别给 PC 和 SP 设定从异常处理向量表中取出的值，然后开始执行程序。

### 5.2.3 手动复位

如果在  $\overline{\text{RES}}$  引脚为高电平时将  $\overline{\text{MRES}}$  引脚置为低电平，本 LSI 就进入手动复位状态。为了确保本 LSI 的复位， $\overline{\text{MRES}}$  引脚的低电平在软件待机时（在时钟停止时），至少必须保持 WDT 设定的振荡稳定时间；在时钟运行时，至少保持 20tcyc。在手动复位的状态下，CPU 的内部状态被初始化，而内部外围模块的各寄存器不被初始化。如果在总线周期中进入手动复位的状态，就在等待总线周期结束后开始手动复位的异常处理。因此，总线周期不会因手动复位而中途停止。但是，如果将  $\overline{\text{MRES}}$  置为低电平，就必须将低电平保持到总线周期结束并进入手动复位状态为止（低电平必须保持大于最长的总线周期）。有关手动复位状态下的各引脚状态，请参照“附录 A. 引脚状态”。在手动复位的状态下，如果将  $\overline{\text{MRES}}$  引脚保持一定期间的低电平后变为高电平，就开始手动复位的异常处理。此时，CPU 按照和上电复位异常处理相同的步骤运行。

## 5.3 地址错误

### 5.3.1 地址错误的发生源

如表 5.6 所示，在取指令或者读写数据时发生地址错误。

表 5.6 总线周期和地址错误

总线周期		总线周期的内容	地址错误的发生
种类	总线主控器		
取指令	CPU	从偶数地址开始取指令	无（正常）
		从奇数地址开始取指令	发生地址错误
		从非内部外围模块空间取指令	无（正常）
		从内部外围模块空间取指令	发生地址错误
		在单芯片模式中，从外部存储器空间取指令	发生地址错误
读写数据	CPU 或者 DMAC 或者 DTC	从偶数地址开始存取字数据	无（正常）
		从奇数地址开始存取字数据	发生地址错误
		从长字边界开始存取长字数据	无（正常）
		从非长字边界开始存取长字数据	发生地址错误
		在内部外围模块空间内存取字数据和字节数据	无（正常）
		在 16 位内部外围模块空间内存取长字数据	无（正常）
		在 8 位内部外围模块空间内存取长字数据	无（正常）
		在单芯片模式中，存取外部存储空间	发生地址错误

### 5.3.2 地址错误的异常处理

当发生地址错误时，发生地址错误的总线周期就结束，并在执行中的指令结束后开始地址错误的异常处理。此时 CPU 的运行如下：

1. 将状态寄存器（SR）保存到堆栈。
2. 将程序计数器（PC）保存到堆栈。保存的 PC 值是发生本异常处理的指令地址。但是，如果发生本异常处理的指令已被分配到延迟槽，就保存前一条延迟转移指令的地址。
3. 从对应发生地址错误的异常处理向量表中取出异常服务程序的起始地址，并从该地址开始执行程序。此时的转移不是延迟转移。

## 5.4 中断

### 5.4.1 中断源

如表 5.7 所示，启动异常处理的中断源有 NMI、用户断点、IRQ 和内部外围模块。

表 5.7 中断源

种类	请求源	源数
NMI	NMI 引脚（来自外部的输入）	1
用户断点	用户断点控制器（UBC）	1
IRQ	IRQ0 ~ IRQ7 引脚（来自外部的输入）	8
内部外围模块	直接存储器存取控制器（DMAC）	8
	多功能定时器脉冲单元 2（MTU2）	28
	多功能定时器脉冲单元 2S（MTU2S）	13
	看门狗定时器（WDT）	1
	数据传送控制器（DTC）	1
	总线状态控制器（BSC）	1
	A/D 转换器（A/D_0、A/D_1、A/D_2）	3
	比较匹配定时器（CMT_0、CMT_1）	2
	串行通信接口（SCI_0、SCI_1、SCI_2）	12
	带 FIFO 的串行通信接口（SCI_3）	4
	同步串行通信单元（SSU）	3
	端口输出的允许（POE）	3
I <sup>2</sup> C 总线接口 2（I <sup>2</sup> C2）	5	

分别给各中断源分配了不同的向量号和向量表偏移量。有关向量号和向量表地址偏移量，请参照“第 6 章 中断控制器（INTC）”的表 6.3。

## 5.4.2 中断优先级

中断源设有优先级，如果同时发生多个中断（多重中断），就根据中断控制器（INTC）判断优先级，并按照该判断结果启动异常处理。

用优先级 0 ~ 16 表示中断源的优先级，优先级 0 为最低、优先级 16 为最高。NMI 中断是优先级为 16 并且是不能屏蔽的最高级中断，随时被接受。用户断点中断的优先级为 15。能通过 INTC 的中断优先级寄存器 A ~ F、H ~ M（IPRA ~ IPRF、IPRH ~ IPRM）自由设定 IRQ 中断和内部外围模块中断的优先级（表 5.8），能设定的优先级为 0 ~ 15，不能设定优先级 16。有关 IPRA ~ IPRF、IPRH ~ IPRM，请参照“6.3.4 中断优先级寄存器 A ~ F、H ~ M（IPRA ~ IPRF、IPRH ~ IPRM）”。

表 5.8 中断优先级

种类	优先级	备注
NMI	16	固定优先级，不能屏蔽
用户断点	15	固定优先级
IRQ	0 ~ 15	通过中断优先级设定寄存器 A ~ F、H ~ M（IPRA ~ IPRF、IPRH ~ IPRM）来设定
内部外围模块		

## 5.4.3 中断异常处理

如果发生中断，就根据中断控制器（INTC）判断优先级。NMI 随时被接受，但是其他中断只在其优先级高于设定在状态寄存器（SR）的中断屏蔽位（I3 ~ I0）的优先级时才被接受。

如果接受中断，就开始中断异常处理。在中断异常处理中，CPU 将 SR 和程序计数器（PC）保存到堆栈，并将接受的中断优先级的值写到 SR 的 I3 ~ I0 位。但是在 NMI 的情况下，优先级为 16，而设定到 I3 ~ I0 位的值为 HF（级 15）。然后，从对应被接受中断的异常处理向量表中取出异常服务程序的起始地址，转移到该地址后开始执行程序。有关中断异常处理请参照“6.6 运行说明”。

## 5.5 由指令引起的异常

### 5.5.1 由指令引起的异常种类

如表 5.9 所示，启动异常处理的指令有陷阱指令、槽非法指令和一般非法指令。

表 5.9 由指令引起的异常种类

种类	源指令	备注
陷阱指令	TRAPA	
槽非法指令 *	配置在紧接着延迟转移指令（延迟槽）之后的未定义代码或者 PC 改写指令	延迟转移指令: JMP、JSR、BRA、BSR、RTS、RTE、BF/S、BT/S、BSRF、BRA PC 改写指令: JMP、JSR、BRA、BSR、RTS、RTE、BT、BF、TRAPA、BF/S、BT/S、BSRF、BRA、LDC Rm, SR、LDC.L @Rm+, SR
一般非法指令 *	在延迟槽以外的未定义代码	

【注】\* 如果对 H'F000 ~ H'FFFF 以外的未定义指令进行译码，就不保证运行。



### 5.5.2 陷阱指令

如果执行 TRAPA 指令，就开始陷阱指令的异常处理。此时 CPU 的运行如下：

1. 将状态寄存器（SR）保存到堆栈。
2. 将程序计数器（PC）保存到堆栈。保存的 PC 值是 TRAPA 指令的下一个指令起始地址。
3. 从对应 TRAPA 指令指定向量号的异常处理向量表中取出异常服务程序起始地址，转移到该地址后开始执行程序。此时不是延迟转移。

### 5.5.3 槽非法指令

将配置在紧接着延迟转移指令之后的指令称为“配置给延迟槽的指令”。在配置给延迟槽的指令为未定义代码时，如果对此未定义的代码进行译码，就开始槽非法指令的异常处理；在配置给延迟槽的指令为 PC 改写指令时，如果对 PC 改写指令进行译码，就开始槽非法指令的异常处理。在进行槽非法指令的异常处理时，CPU 的运行如下：

1. 将 SR 保存到堆栈。
2. 将 PC 保存到堆栈。保存的 PC 值为未定义代码或者 PC 改写指令前一条延迟转移指令的地址。
3. 从对应发生异常的异常处理向量表中取出异常服务程序起始地址，转移到该地址后开始执行程序。此时的转移不是延迟转移。

### 5.5.4 一般非法指令

如果对配置在除紧接着延迟转移指令（延迟槽）以外的未定义代码进行译码，就开始一般非法指令的异常处理。此时，CPU 按照和槽非法指令的异常处理相同的步骤运行。但是，和槽非法指令的异常处理不同，保存的 PC 值为此未定义代码的起始地址。

## 5.6 异常处理的接受

如表 5.10 所示，如果在紧接着延迟槽或者中断禁止指令之后发生除复位以外的全部异常源，就有可能不立即被接受而被保留。这种情况下，在接受的指令被译码后才能接受以上的异常源。

表 5.10 发生在紧接着延迟槽和中断禁止指令之后的异常源

发生时间	异常源				
	地址错误	一般非法指令	槽非法指令	陷阱指令	中断
延迟槽	× *2	—	× *2	—	× *3
紧接在中断禁止指令之后 *1	○	○	○	○	× *4

#### 【符号说明】

- ：接受。
- ×：不接受。
- ：不可能的情况。

【注】 \*1 中断禁止指令：LDC、LDC.L、STC、STC.L、LDS、LDS.L、STS、STS.L

\*2 在执行延迟转移指令前被接受。但是，如果在 RTE 指令的延迟槽中出现地址错误或者槽非法指令，就不保证运行。

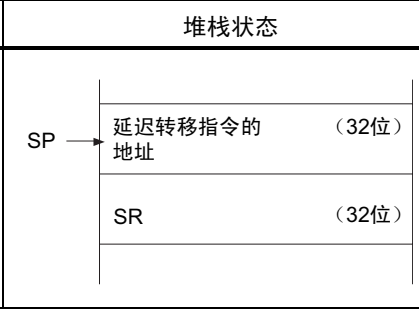
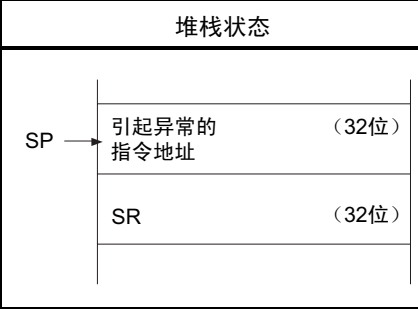
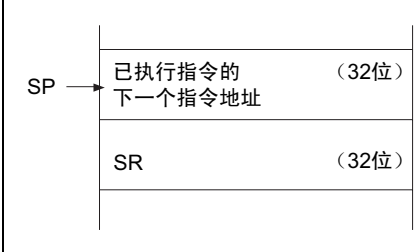
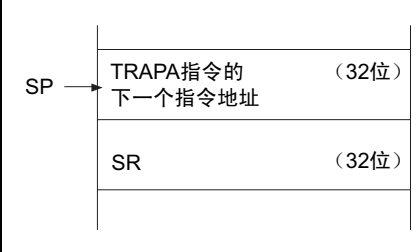

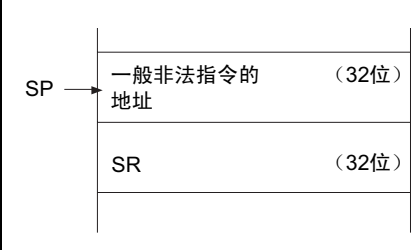
\*3 在延迟转移后（在延迟槽指令和延迟转移目标指令之间）被接受。

\*4 在执行紧接着中断禁止指令之后的指令后（在中断禁止指令后的指令和下一条指令之间）被接受。

## 5.7 异常处理后的堆栈状态

异常处理结束后的堆栈状态如表 5.11 所示。

表 5.11 异常处理结束后的堆栈状态

种类	堆栈状态	种类	堆栈状态
地址错误 (引起异常发生的指令在延迟槽时)		地址错误 (引起异常发生的指令不在延迟槽时)	
中断		陷阱指令	
槽非法指令		一般非法指令	

## 5.8 使用注意事项

### 5.8.1 堆栈指针 (SP) 的值

SP 的值必须是 4 的倍数。否则，如果通过异常处理存取堆栈，就会发生地址错误。

### 5.8.2 向量基址寄存器 (VBR) 的值

VBR 的值必须是 4 的倍数。否则，如果通过异常处理存取堆栈，就会发生地址错误。

### 5.8.3 在地址错误异常处理的堆栈存取中发生的地址错误

如果 SP 不是 4 的倍数，就会在异常处理（中断等）的堆栈存取中发生地址错误，在该异常处理后转移到地址错误的异常处理。即使在地址错误异常处理的堆栈存取中也会发生地址错误。但是，为了不使地址错误异常处理的堆栈存取无限继续，不接受此时的地址错误。因此，能将程序的控制转移到地址错误的异常服务程序并进行错误处理。

如果在异常处理的堆栈存取中发生地址错误，就执行堆栈存取的总线周期（写）。在 SR 和 PC 的堆栈存取中，因为 SP 和 PC 分别减 4，所以即使在堆栈存取结束后，SP 的值也不是 4 的倍数。另外，堆栈存取时输出的地址值是对 SP 值的低 2 位进行 0 方向舍入后的值。此时，堆栈存取的写数据不定。

#### 5.8.4 有关槽非法指令异常处理的注意事项

本 LSI 的槽非法指令异常处理的规格和以前的 SH-2 有以下不同点：

- 以前的 SH-2: LDC Rm, SR、LDC.L @Rm+, SR 不为槽非法指令的对象。
- 本 LSI: LDC Rm, SR、LDC.L @Rm+, SR 为槽非法指令的对象。

与此有关的本公司软件产品的对应情况如下：

##### (1) 编译程序

V.4 以后版本的编译程序不将该指令配置到延迟槽。

##### (2) $\mu$ ITRON 规格实时 OS

###### (a) HI7000/4、HI-SH7

在 OS 内，延迟槽中没有该指令。

###### (b) HI7000

在 OS 内，延迟槽中有该指令，所以本 LSI 发生槽非法指令。

###### (c) 其他

在通过汇编程序进行记述或者导入中间件时，本 LSI 有可能发生槽非法指令。

另外，摘自该指令串的检查程序登载于本公司网站的产品信息页上，如果需要，请从主页下载后进行确认。

## 第 6 章 中断控制器 (INTC)

中断控制器 (INTC) 判断中断源的优先级以及控制向 CPU 的中断请求。

### 6.1 特点

- 能将中断优先级设定为 16 级
- NMI 的噪声消除功能
- 能将发生的中断输出到外部 (IRQOUT 引脚)

INTC 的框图如图 6.1 所示。

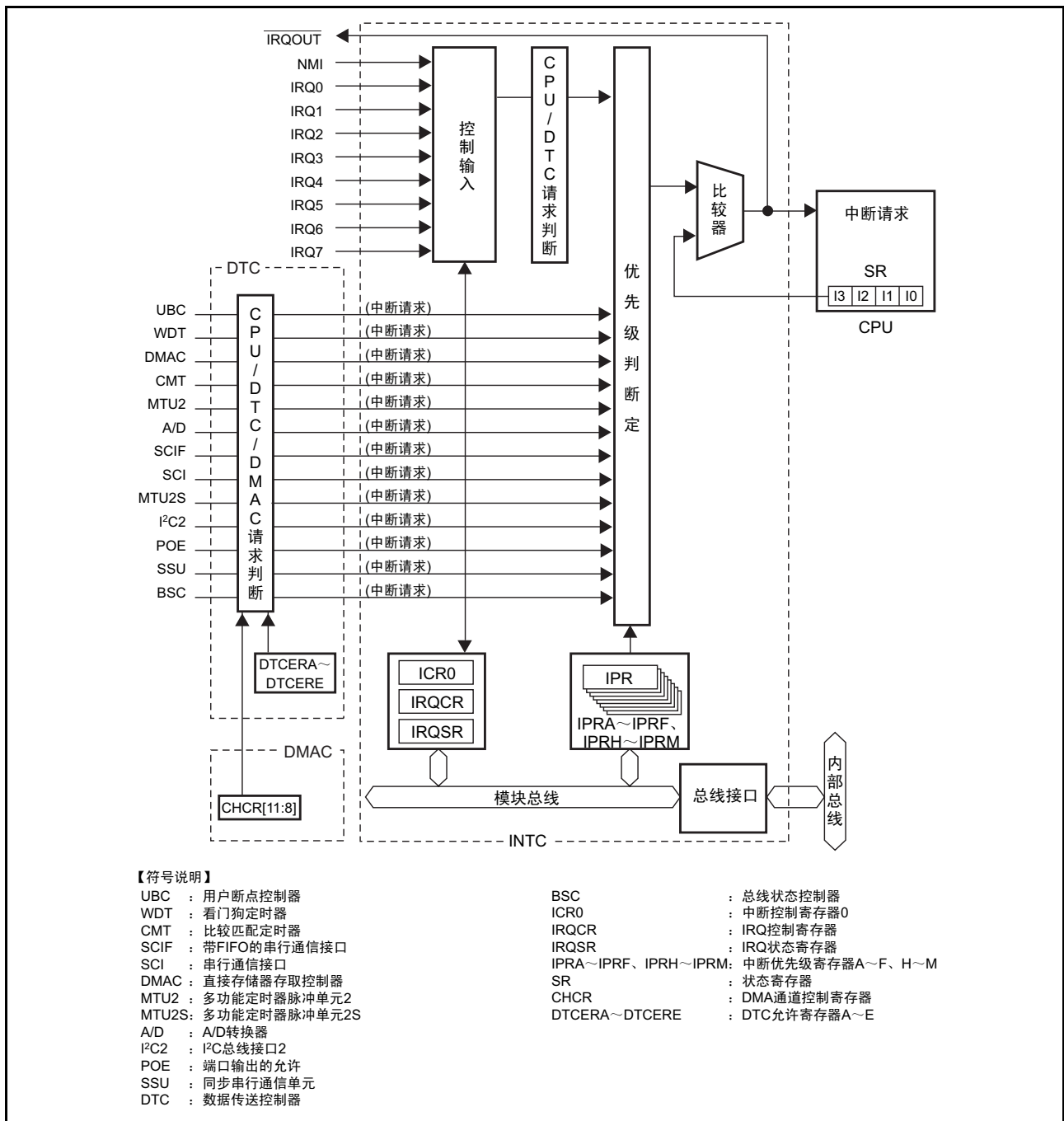


图 6.1 INTC 的框图

## 6.2 输入 / 输出引脚

INTC 的引脚如表 6.1 所示。

表 6.1 引脚结构

名称	引脚名	输入 / 输出	功 能
非屏蔽中断输入引脚	NMI	输入	输入不可屏蔽的中断请求信号
中断请求输入引脚	IRQ0 ~ IRQ7	输入	输入可屏蔽的中断请求信号
中断请求输出引脚	$\overline{\text{IRQOUT}}$	输出	输出通知发生中断源的信号

## 6.3 寄存器说明

INTC 有以下寄存器。有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 27 章 寄存器一览”。

表 6.2 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
中断控制寄存器 0	ICR0	R/W	H'x000	H'FFFFE900	8、16
IRQ 控制寄存器	IRQCR	R/W	H'0000	H'FFFFE902	8、16
IRQ 状态寄存器	IRQSR	R/W	H'xx00	H'FFFFE904	8、16
中断优先级寄存器 A	IPRA	R/W	H'0000	H'FFFFE906	8、16
中断优先级寄存器 B	IPRB	R/W	H'0000	H'FFFFE908	8、16
中断优先级寄存器 C	IPRC	R/W	H'0000	H'FFFFE980	16
中断优先级寄存器 D	IPRD	R/W	H'0000	H'FFFFE982	16
中断优先级寄存器 E	IPRE	R/W	H'0000	H'FFFFE984	16
中断优先级寄存器 F	IPRF	R/W	H'0000	H'FFFFE986	16
中断优先级寄存器 H	IPRH	R/W	H'0000	H'FFFFE98A	16
中断优先级寄存器 I	IPRI	R/W	H'0000	H'FFFFE98C	16
中断优先级寄存器 J	IPRJ	R/W	H'0000	H'FFFFE98E	16
中断优先级寄存器 K	IPRK	R/W	H'0000	H'FFFFE990	16
中断优先级寄存器 L	IPRL	R/W	H'0000	H'FFFFE992	16
中断优先级寄存器 M	IPRM	R/W	H'0000	H'FFFFE994	16

### 6.3.1 中断控制寄存器 0 (ICR0)

ICR0 是 16 位寄存器，设定外部中断输入引脚 NMI 的输入信号检测模式以及表示 NMI 引脚的输入电平。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NMIL	-	-	-	-	-	-	NMIE	-	-	-	-	-	-	-	-
初始值:	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R

【注】\* NMI 引脚为高电平时此位是 1，为低电平时是 0。

位	位名	初始值	R/W	说明
15	NMIL	*	R	NMI 的输入电平 设定 NMI 引脚输入信号的电平。能通过读此位得知 NMI 引脚的电平。写操作无效。 0: NMI 引脚为低电平 1: NMI 引脚为高电平
14 ~ 9	—	全 0	R	保留位 读写的值总是 0。
8	NMIE	0	R/W	NMI 的边沿选择 0: 在 NMI 输入的下降沿检测中断请求 1: 在 NMI 输入的上升沿检测中断请求
7 ~ 0	—	全 0	R	保留位 读写的值总是 0。

### 6.3.2 IRQ 控制寄存器 (IRQCR)

IRQCR 是 16 位寄存器，设定外部中断输入引脚 IRQ0 ~ IRQ7 的输入信号检测模式。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ71S	IRQ70S	IRQ61S	IRQ60S	IRQ51S	IRQ50S	IRQ41S	IRQ40S	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15	IRQ71S	0	R/W	IRQ7 的检测选择 选择 IRQ7 中断输入引脚的输入信号检测模式。 00: 在 IRQ7 输入引脚的低电平检测中断请求 01: 在 IRQ7 输入引脚的下降沿检测中断请求 10: 在 IRQ7 输入引脚的上升沿检测中断请求 11: 在 IRQ7 输入引脚的双边沿检测中断请求
14	IRQ70S	0	R/W	
13	IRQ61S	0	R/W	IRQ6 的检测选择 选择 IRQ6 中断输入引脚的输入信号检测模式。 00: 在 IRQ6 输入引脚的低电平检测中断请求 01: 在 IRQ6 输入引脚的下降沿检测中断请求 10: 在 IRQ6 输入引脚的上升沿检测中断请求 11: 在 IRQ6 输入引脚的双边沿检测中断请求
12	IRQ60S	0	R/W	

位	位名	初始值	R/W	说 明
11 10	IRQ51S IRQ50S	0 0	R/W R/W	IRQ5 的检测选择 选择 IRQ5 中断输入引脚的输入信号检测模式。 00: 在 IRQ5 输入引脚的低电平检测中断请求 01: 在 IRQ5 输入引脚的下降沿检测中断请求 10: 在 IRQ5 输入引脚的上升沿检测中断请求 11: 在 IRQ5 输入引脚的双边沿检测中断请求
9 8	IRQ41S IRQ40S	0 0	R/W R/W	IRQ4 的检测选择 选择 IRQ4 中断输入引脚的输入信号检测模式。 00: 在 IRQ4 输入引脚的低电平检测中断请求 01: 在 IRQ4 输入引脚的下降沿检测中断请求 10: 在 IRQ4 输入引脚的上升沿检测中断请求 11: 在 IRQ4 输入引脚的双边沿检测中断请求
7 6	IRQ31S IRQ30S	0 0	R/W R/W	IRQ3 的检测选择 选择 IRQ3 中断输入引脚的输入信号检测模式。 00: 在 IRQ3 输入引脚的低电平检测中断请求 01: 在 IRQ3 输入引脚的下降沿检测中断请求 10: 在 IRQ3 输入引脚的上升沿检测中断请求 11: 在 IRQ3 输入引脚的双边沿检测中断请求
5 4	IRQ21S IRQ20S	0 0	R/W R/W	IRQ2 的检测选择 选择 IRQ2 中断输入引脚的输入信号检测模式。 00: 在 IRQ2 输入引脚的低电平检测中断请求 01: 在 IRQ2 输入引脚的下降沿检测中断请求 10: 在 IRQ2 输入引脚的上升沿检测中断请求 11: 在 IRQ2 输入引脚的双边沿检测中断请求
3 2	IRQ11S IRQ10S	0 0	R/W R/W	IRQ1 的检测选择 选择 IRQ1 中断输入引脚的输入信号检测模式。 00: 在 IRQ1 输入引脚的低电平检测中断请求 01: 在 IRQ1 输入引脚的下降沿检测中断请求 10: 在 IRQ1 输入引脚的上升沿检测中断请求 11: 在 IRQ1 输入引脚的双边沿检测中断请求
1 0	IRQ01S IRQ00S	0 0	R/W R/W	IRQ0 的检测选择 选择 IRQ0 中断输入引脚的输入信号检测模式。 00: 在 IRQ0 输入引脚的低电平检测中断请求 01: 在 IRQ0 输入引脚的下降沿检测中断请求 10: 在 IRQ0 输入引脚的上升沿检测中断请求 11: 在 IRQ0 输入引脚的双边沿检测中断请求

### 6.3.3 IRQ 状态寄存器 (IRQSR)

IRQSR 是 16 位寄存器，表示外部中断输入引脚 IRQ0 ~ IRQ7 的状态和中断请求状态。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ7L	IRQ6L	IRQ5L	IRQ4L	IRQ3L	IRQ2L	IRQ1L	IRQ0L	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
初始值:	1	1	1	1	*	*	*	*	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

【注】\* 对应的IRQ引脚为高电平时此位是1，为低电平时是0。

位	位名	初始值	R/W	说明
15	IRQ7L	*	R	表示 IRQ7 的引脚状态。 0: IRQ7 引脚为低电平 1: IRQ7 引脚为高电平
14	IRQ6L	*	R	表示 IRQ6 的引脚状态。 0: IRQ6 引脚为低电平 1: IRQ6 引脚为高电平
13	IRQ5L	*	R	表示 IRQ5 的引脚状态。 0: IRQ5 引脚为低电平 1: IRQ5 引脚为高电平
12	IRQ4L	*	R	表示 IRQ4 的引脚状态。 0: IRQ4 引脚为低电平 1: IRQ4 引脚为高电平
11	IRQ3L	*	R	表示 IRQ3 的引脚状态。 0: IRQ3 引脚为低电平 1: IRQ3 引脚为高电平
10	IRQ2L	*	R	表示 IRQ2 的引脚状态。 0: IRQ2 引脚为低电平 1: IRQ2 引脚为高电平
9	IRQ1L	*	R	表示 IRQ1 的引脚状态。 0: IRQ1 引脚为低电平 1: IRQ1 引脚为高电平
8	IRQ0L	*	R	表示 IRQ0 的引脚状态。 0: IRQ0 引脚为低电平 1: IRQ0 引脚为高电平



位	位名	初始值	R/W	说 明
7	IRQ7F	0	R/W	<p>表示 IRQ7 中断请求的状态。</p> <p>在已设定电平检测时</p> <p>0: 没有 IRQ7 中断请求 [清除条件]</p> <ul style="list-style-type: none"> <li>给 IRQ7 引脚输入高电平</li> </ul> <p>1: 有 IRQ7 中断请求 [置位条件]</p> <ul style="list-style-type: none"> <li>给 IRQ7 引脚输入低电平</li> </ul> <p>在已设定边沿检测时,</p> <p>0: 未检测到 IRQ7 中断请求 [清除条件]</p> <ul style="list-style-type: none"> <li>在读 IRQ7F=1 的状态后, 写 0</li> <li>接受 IRQ7 中断</li> </ul> <p>1: 检测到 IRQ7 中断请求 [置位条件]</p> <ul style="list-style-type: none"> <li>检测到 IRQ7 引脚的指定边沿</li> </ul>
6	IRQ6F	0	R/W	<p>表示 IRQ6 中断请求的状态。</p> <p>在已设定电平检测时</p> <p>0: 没有 IRQ6 中断请求 [清除条件]</p> <ul style="list-style-type: none"> <li>给 IRQ6 引脚输入高电平</li> </ul> <p>1: 有 IRQ6 中断请求 [置位条件]</p> <ul style="list-style-type: none"> <li>给 IRQ6 引脚输入低电平</li> </ul> <p>在已设定边沿检测时,</p> <p>0: 未检测到 IRQ6 中断请求 [清除条件]</p> <ul style="list-style-type: none"> <li>在读 IRQ6F=1 的状态后, 写 0</li> <li>接受 IRQ6 中断</li> </ul> <p>1: 检测到 IRQ6 中断请求 [置位条件]</p> <ul style="list-style-type: none"> <li>检测到 IRQ6 引脚的指定边沿</li> </ul>

位	位名	初始值	R/W	说 明
5	IRQ5F	0	R/W	<p>表示 IRQ5 中断请求的状态。</p> <p>在已设定电平检测时</p> <p>0: 没有 IRQ5 中断请求 [清除条件]</p> <ul style="list-style-type: none"> <li>给 IRQ5 引脚输入高电平</li> </ul> <p>1: 有 IRQ5 中断请求 [置位条件]</p> <ul style="list-style-type: none"> <li>给 IRQ5 引脚输入低电平</li> </ul> <p>在已设定边沿检测时,</p> <p>0: 未检测到 IRQ5 中断请求 [清除条件]</p> <ul style="list-style-type: none"> <li>在读 IRQ5F=1 的状态后, 写 0</li> <li>接受 IRQ5 中断</li> </ul> <p>1: 检测到 IRQ5 中断请求 [置位条件]</p> <ul style="list-style-type: none"> <li>检测到 IRQ5 引脚的指定边沿</li> </ul>
4	IRQ4F	0	R/W	<p>表示 IRQ4 中断请求的状态。</p> <p>在已设定电平检测时</p> <p>0: 没有 IRQ4 中断请求 [清除条件]</p> <ul style="list-style-type: none"> <li>给 IRQ4 引脚输入高电平</li> </ul> <p>1: 有 IRQ4 中断请求 [置位条件]</p> <ul style="list-style-type: none"> <li>给 IRQ4 引脚输入低电平</li> </ul> <p>在已设定边沿检测时,</p> <p>0: 未检测到 IRQ4 中断请求 [清除条件]</p> <ul style="list-style-type: none"> <li>在读 IRQ4F=1 的状态后, 写 0</li> <li>接受 IRQ4 中断</li> </ul> <p>1: 检测到 IRQ4 中断请求 [置位条件]</p> <ul style="list-style-type: none"> <li>检测到 IRQ4 引脚的指定边沿</li> </ul>

位	位名	初始值	R/W	说 明
3	IRQ3F	0	R/W	<p>表示 IRQ3 中断请求的状态。</p> <p>在已设定电平检测时</p> <p>0: 没有 IRQ3 中断请求 [清除条件]</p> <ul style="list-style-type: none"> <li>给 IRQ3 引脚输入高电平</li> </ul> <p>1: 有 IRQ3 中断请求 [置位条件]</p> <ul style="list-style-type: none"> <li>给 IRQ3 引脚输入低电平</li> </ul> <p>在已设定边沿检测时,</p> <p>0: 未检测到 IRQ3 中断请求 [清除条件]</p> <ul style="list-style-type: none"> <li>在读 IRQ3F=1 的状态后, 写 0</li> <li>接受 IRQ3 中断</li> </ul> <p>1: 检测到 IRQ3 中断请求 [置位条件]</p> <ul style="list-style-type: none"> <li>检测到 IRQ3 引脚的指定边沿</li> </ul>
2	IRQ2F	0	R/W	<p>表示 IRQ2 中断请求的状态。</p> <p>在已设定电平检测时</p> <p>0: 没有 IRQ2 中断请求 [清除条件]</p> <ul style="list-style-type: none"> <li>给 IRQ2 引脚输入高电平</li> </ul> <p>1: 有 IRQ2 中断请求 [置位条件]</p> <ul style="list-style-type: none"> <li>给 IRQ2 引脚输入低电平</li> </ul> <p>在已设定边沿检测时</p> <p>0: 未检测到 IRQ2 中断请求 [清除条件]</p> <ul style="list-style-type: none"> <li>在读 IRQ2F=1 的状态后, 写 0</li> <li>接受 IRQ2 中断</li> </ul> <p>1: 检测到 IRQ2 中断请求 [置位条件]</p> <ul style="list-style-type: none"> <li>检测到 IRQ2 引脚的指定边沿</li> </ul>

位	位名	初始值	R/W	说 明
1	IRQ1F	0	R/W	<p>表示 IRQ1 中断请求的状态。</p> <p>在已设定电平检测时</p> <p>0: 没有 IRQ1 中断请求 [清除条件]</p> <ul style="list-style-type: none"> <li>给 IRQ1 引脚输入高电平</li> </ul> <p>1: 有 IRQ1 中断请求 [置位条件]</p> <ul style="list-style-type: none"> <li>给 IRQ1 引脚输入低电平</li> </ul> <p>在已设定边沿检测时</p> <p>0: 未检测到 IRQ1 中断请求 [清除条件]</p> <ul style="list-style-type: none"> <li>在读 IRQ1F=1 的状态, 后写 0</li> <li>接受 IRQ1 中断</li> </ul> <p>1: 检测到 IRQ1 中断请求 [置位条件]</p> <ul style="list-style-type: none"> <li>检测到 IRQ1 引脚的指定边沿</li> </ul>
0	IRQ0F	0	R/W	<p>表示 IRQ0 中断请求的状态。</p> <p>在已设定电平检测时</p> <p>0: 没有 IRQ0 中断请求 [清除条件]</p> <ul style="list-style-type: none"> <li>给 IRQ0 引脚输入高电平</li> </ul> <p>1: 有 IRQ0 中断请求 [置位条件]</p> <ul style="list-style-type: none"> <li>给 IRQ0 引脚输入低电平</li> </ul> <p>在已设定边沿检测时</p> <p>0: 未检测到 IRQ0 中断请求 [清除条件]</p> <ul style="list-style-type: none"> <li>在读 IRQ0F=1 的状态后, 写 0</li> <li>接受 IRQ0 中断</li> </ul> <p>1: 检测到 IRQ 中断请求 [置位条件]</p> <ul style="list-style-type: none"> <li>检测到 IRQ0 引脚的指定边沿</li> </ul>

【注】 \* 对应的 IRQ 引脚为高电平时此位是 1, 为低电平时是 0。

## 6.3.4 中断优先级寄存器 A ~ F、H ~ M (IPRA ~ IPRF、IPRH ~ IPRM)

IPR 是 13 个 16 位可读 / 写寄存器，设定除 NMI 以外的中断源优先级（级 15 ~ 0）。有关各中断源和 IPR 的对应请参照表 6.3。通过给 bit15 ~ 12、bit11 ~ 8、bit7 ~ 4、bit3 ~ 0 的各 4 位设定 H'0 到 H'F 范围内的值，决定对应中断请求的优先级。对于没有分配的保留位，必须设定 H'0 (B'0000)。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IPR[15:12]				IPR[11:8]				IPR[7:4]				IPR[3:0]			
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 12	IPR[15:12]	0000	R/W	设定对应中断源的优先级 0000: 优先级 0 (最低)      1000: 优先级 8 0001: 优先级 1              1001: 优先级 9 0010: 优先级 2              1010: 优先级 10 0011: 优先级 3              1011: 优先级 11 0100: 优先级 4              1100: 优先级 12 0101: 优先级 5              1101: 优先级 13 0110: 优先级 6              1110: 优先级 14 0111: 优先级 7              1111: 优先级 15 (最高)
11 ~ 8	IPR[11:8]	0000	R/W	设定对应中断源的优先级 0000: 优先级 0 (最低)      1000: 优先级 8 0001: 优先级 1              1001: 优先级 9 0010: 优先级 2              1010: 优先级 10 0011: 优先级 3              1011: 优先级 11 0100: 优先级 4              1100: 优先级 12 0101: 优先级 5              1101: 优先级 13 0110: 优先级 6              1110: 优先级 14 0111: 优先级 7              1111: 优先级 15 (最高)
7 ~ 4	IPR[7:4]	0000	R/W	设定对应中断源的优先级 0000: 优先级 0 (最低)      1000: 优先级 8 0001: 优先级 1              1001: 优先级 9 0010: 优先级 2              1010: 优先级 10 0011: 优先级 3              1011: 优先级 11 0100: 优先级 4              1100: 优先级 12 0101: 优先级 5              1101: 优先级 13 0110: 优先级 6              1110: 优先级 14 0111: 优先级 7              1111: 优先级 15 (最高)
3 ~ 0	IPR[3:0]	0000	R/W	设定对应中断源的优先级 0000: 优先级 0 (最低)      1000: 优先级 8 0001: 优先级 1              1001: 优先级 9 0010: 优先级 2              1010: 优先级 10 0011: 优先级 3              1011: 优先级 11 0100: 优先级 4              1100: 优先级 12 0101: 优先级 5              1101: 优先级 13 0110: 优先级 6              1110: 优先级 14 0111: 优先级 7              1111: 优先级 15 (最高)

【注】 在此表中位名用一般名称表示，而在寄存器一览表中用模块名表示。

## 6.4 中断源

### 6.4.1 外部中断源

中断源分为用户断点、NMI、IRQ 和内部外围模块 4 类。各中断的优先级用优先级的值 (0 ~ 16) 表示, 优先级 0 最低, 优先级 16 最高。如果设定为优先级 0, 该中断就被屏蔽。

#### (1) NMI 中断

NMI 中断是优先级 16 的中断, 并且总被接受。检测 NMI 引脚的输入边沿, 检测边沿能通过设定中断控制寄存器 0 (ICR0) 的 NMI 边沿选择位 (NMIE), 选择上升沿检测或者下降沿。

通过 NMI 中断异常处理, 将状态寄存器 (SR) 的中断屏蔽位 (I3 ~ I0) 设定为 15。

#### (2) IRQ7 ~ IRQ0 中断

IRQ 中断是由 IRQ7 ~ IRQ0 引脚输入引起的中断。能通过设定 IRQ 控制寄存器 (IRQCR) 的 IRQ 检测选择位 (IRQ71S、IRQ70S ~ IRQ01S、IRQ00S), 给每个引脚选择低电平检测、下降沿检测、上升沿检测或者双边沿检测。另外, 能通过中断优先级寄存器 A、B (IPRA 和 IPRB), 在 0 ~ 15 范围内给每个引脚设定优先级。如果将 IRQ 中断设定为低电平检测, 就在 IRQ 引脚为低电平时将中断请求信号送给 INTC, 当 IRQ 引脚变为高电平时, 就停止中断请求信号的传送。能通过读 IRQ 状态寄存器 (IRQSR) 的 IRQ 标志 (IRQ7F ~ IRQ0F) 确认有无中断请求。

如果将 IRQ 中断设定为下降沿 (上升沿或者双边沿) 检测, 就在 IRQ 引脚从高电平变为低电平 (从低电平变为高电平或者从高电平变为低电平) 时检测到中断请求, 将中断请求信号送给 INTC。IRQ 中断请求的检测结果一直保持到该中断请求被接受为止。另外, 能通过读 IRQ 状态寄存器 (IRQSR) 的 IRQ 标志 (IRQ7F ~ IRQ0F) 确认是否检测到 IRQ 中断请求, 能通过读 1 后写 0 来取消 IRQ 中断请求的检测结果。

通过 IRQ 中断异常处理, 将状态寄存器 (SR) 的中断屏蔽位 (I3 ~ I0) 设定为已接受的 IRQ 中断优先级的值。

IRQ7 ~ IRQ0 中断的框图如图 6.2 所示。

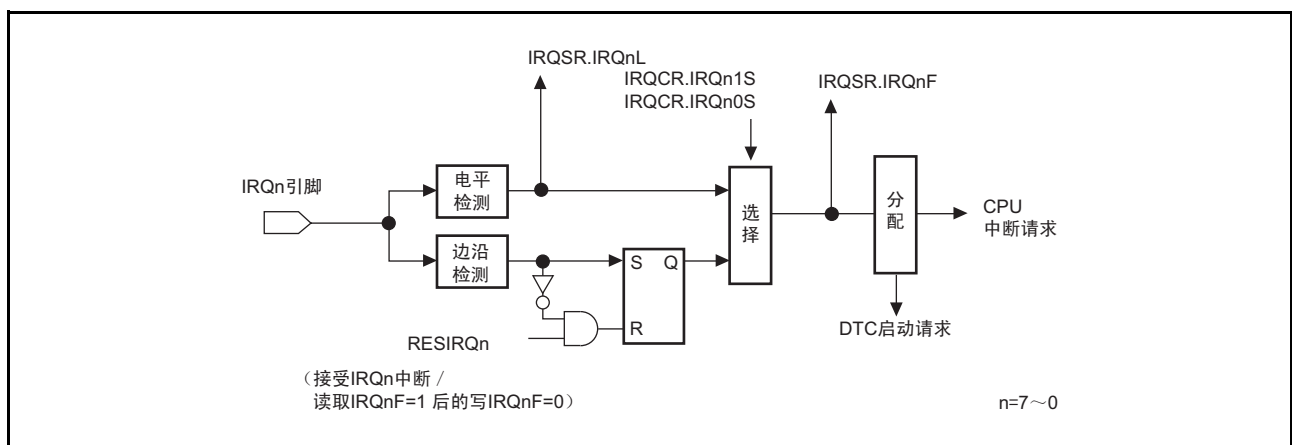


图 6.2 IRQ0 ~ IRQ7 中断控制

### 6.4.2 内部外围模块中断

内部外围模块中断是由以下所示的内部外围模块发生的中断。

因为每个中断源分配了不同的中断向量，所以不需要在异常服务程序中判断中断源。能通过中断优先级寄存器 C ~ F、H ~ M (IPRC ~ IPRF、IPRH ~ IPRM)，在 0 ~ 15 的范围内给每个模块设定优先级。通过内部外围模块中断的异常处理，将状态寄存器 (SR) 的中断屏蔽位 (I3 ~ I0) 设定为已接受的内部外围模块中断优先级的值。

### 6.4.3 用户断点中断

用户断点中断是在用户断点控制器 (UBC) 设定的中断条件成立时发生的中断，优先级为 15。在边沿检测到用户断点中断请求，并且该中断请求保持到被接受为止。通过用户断点异常处理，将状态寄存器 (SR) 的中断屏蔽位 (I3 ~ I0) 设定为 15。有关用户断点的详细内容，请参照“第 7 章 用户断点控制器 (UBC)”。

## 6.5 中断异常处理向量表

中断源、向量号、向量地址偏移量和中断优先级如表 6.3 所示。

各中断源分别分配了不同的向量号和向量地址偏移量，向量地址由向量号和向量地址偏移量算出。通过中断异常处理从向量地址指向的向量表中取出异常服务程序的起始地址。有关向量地址的算法，请参照“第 5 章 异常处理”的表 5.4。

能通过中断优先级寄存器 A ~ F、H ~ M (IPRA ~ IPRF、IPRH ~ IPRM)，在 0 ~ 15 的范围内给每个引脚或者模块任意设定 IRQ 中断和内部外围模块中断的优先级。但是，因为小向量号的中断源分配了高优先级，所以不能更改同一 IPR 指定的多个中断源的优先级。通过上电复位，将 IRQ 中断和内部外围模块中断的优先级设定为 0。在将多个中断源设定为相同优先级的情况下，如果同时发生这些中断，就根据如表 6.3 所示的默认优先级进行处理。

表 6.3 中断源、向量地址和中断优先级一览表

中断发生源	名称	向量号	向量表起始地址	IPR	默认优先级
用户断点		12	H'00000030	—	高  低
外部引脚	NMI	11	H'0000002C	—	
	IRQ0	64	H'00000100	IPRA15 ~ IPRA12	
	IRQ1	65	H'00000104	IPRA11 ~ IPRA8	
	IRQ2	66	H'00000108	IPRA7 ~ IPRA4	
	IRQ3	67	H'0000010C	IPRA3 ~ IPRA0	
	IRQ4	68	H'00000110	IPRB15 ~ IPRB12	
	IRQ5	69	H'00000114	IPRB11 ~ IPRB8	
	IRQ6	70	H'00000118	IPRB7 ~ IPRB4	
	IRQ7	71	H'0000011C	IPRB3 ~ IPRB0	
DMAC_0	DEI0	72	H'00000120	IPRC15 ~ IPRC12	
DMAC_1	DEI1	76	H'00000130	IPRC11 ~ IPRC8	
DMAC_2	DEI2	80	H'00000140	IPRC7 ~ IPRC4	
DMAC_3	DEI3	84	H'00000150	IPRC3 ~ IPRC0	

中断发生源	名称	向量号	向量表起始地址	IPR	默认优先级		
MTU2_0	TGIA_0	88	H'00000160	IPRD15 ~ IPRD12	高 ↑		
	TGIB_0	89	H'00000164				
	TGIC_0	90	H'00000168				
	TGID_0	91	H'0000016C				
	TCIV_0	92	H'00000170	IPRD11 ~ IPRD8			
	TGIE_0	93	H'00000174				
	TGIF_0	94	H'00000178				
MTU2_1	TGIA_1	96	H'00000180	IPRD7 ~ IPRD4	↑		
	TGIB_1	97	H'00000184				
	TCIV_1	100	H'00000190	IPRD3 ~ IPRD0			
	TCIU_1	101	H'00000194				
MTU2_2	TGIA_2	104	H'000001A0	IPRE15 ~ IPRE12		↑	
	TGIB_2	105	H'000001A4				
	TCIV_2	108	H'000001B0	IPRE11 ~ IPRE8			
	TCIU_2	109	H'000001B4				
MTU2_3	TGIA_3	112	H'000001C0	IPRE7 ~ IPRE4	↑		
	TGIB_3	113	H'000001C4				
	TGIC_3	114	H'000001C8				
	TGID_3	115	H'000001CC				
	TCIV_3	116	H'000001D0	IPRE3 ~ IPRE0			
MTU2_4	TGIA_4	120	H'000001E0	IPRF15 ~ IPRF12		↑	
	TGIB_4	121	H'000001E4				
	TGIC_4	122	H'000001E8				
	TGID_4	123	H'000001EC				
	TCIV_4	124	H'000001F0	IPRF11 ~ IPRF8			
MTU2_5	TGIU_5	128	H'00000200	IPRF7 ~ IPRF4	↑		
	TGIV_5	129	H'00000204				
	TGIW_5	130	H'00000208				
POE (MTU2)	OEI1	132	H'00000210	IPRF3 ~ IPRF0		↑	
	OEI3	133	H'00000214				
I <sup>2</sup> C2*	IINAKI	156	H'00000270	IPRH11 ~ IPRH8			↑
MTU2S_3	TGIA_3S	160	H'00000280	IPRH7 ~ IPRH4			
	TGIB_3S	161	H'00000284				
	TGIC_3S	162	H'00000288				
	TGID_3S	163	H'0000028C				
	TCIV_3S	164	H'00000290	IPRH3 ~ IPRH0			
MTU2S_4	TGIA_4S	168	H'000002A0	IPRI15 ~ IPRI12	↑		
	TGIB_4S	169	H'000002A4				
	TGIC_4S	170	H'000002A8				
	TGID_4S	171	H'000002AC				
	TCIV_4S	172	H'000002B0	IPRI11 ~ IPRI8			
						低 ↓	





## 6.6 运行说明

### 6.6.1 中断运行的流程

以下说明发生中断时的运行流程，运行流程如图 6.3 所示。

1. 各中断请求源将中断请求信号送给中断控制器。
2. 中断控制器根据中断优先级寄存器 A ~ F、H ~ M (IPRA ~ IPRF、IPRH ~ IPRM)，从送来的中断请求中选择优先级最高的中断，而忽视 \* 优先级低的中断。此时，如果发生多个同优先级或者同模块内的中断，就根据表 6.3 中所示的默认优先级选择优先级最高的中断。
3. 中断控制器选择的中断优先级和 CPU 状态寄存器 (SR) 的中断屏蔽位 (I3 ~ I0) 比较。忽视相同或者低于设定在 I3 ~ I0 位的优先级的中断，而只接受高于 I3 ~ I0 位的优先级的中断，向 CPU 发送中断请求信号。
4. 当中断控制器接受中断时，从  $\overline{\text{IRQOUT}}$  引脚输出低电平。
5. 在对 CPU 要执行的指令进行译码时，检测到从中断控制器送来的中断请求，并将要执行的该指令替换为中断异常处理。
6. 将 SR 和程序计数器 (PC) 保存到堆栈。
7. 将接受的中断优先级写到 SR 的 I3 ~ I0 位。
8. 如果被接受的中断是电平检测或者内部外围模块的中断，就从  $\overline{\text{IRQOUT}}$  引脚输出高电平。如果被接受的中断是边沿检测，就在将流程 5. 中 CPU 要执行的指令替换为中断异常处理时，从  $\overline{\text{IRQOUT}}$  引脚输出高电平。但是，在中断控制器接受高于处理中的中断优先级的其他中断时， $\overline{\text{IRQOUT}}$  引脚仍为低电平。
9. 从对应被接受的中断异常处理向量表中取出异常服务程序起始地址，转移到该地址后开始执行程序。此时的转移不是延迟转移。

**【注】** 必须在中断处理程序中清除中断源标志。为了不错误地再次接受应该被清除的中断源，必须在清除后读中断源标志，并在确认中断源标志已被清除后执行 RTE 指令。

\* 设定为边沿检测的中断请求保留到被接受为止。如果是 IRQ 中断，就能通过存取 IRQ 状态寄存器 (IRQSR) 取消。另外，通过上电复位或者手动复位清除由边沿检测保留的中断。

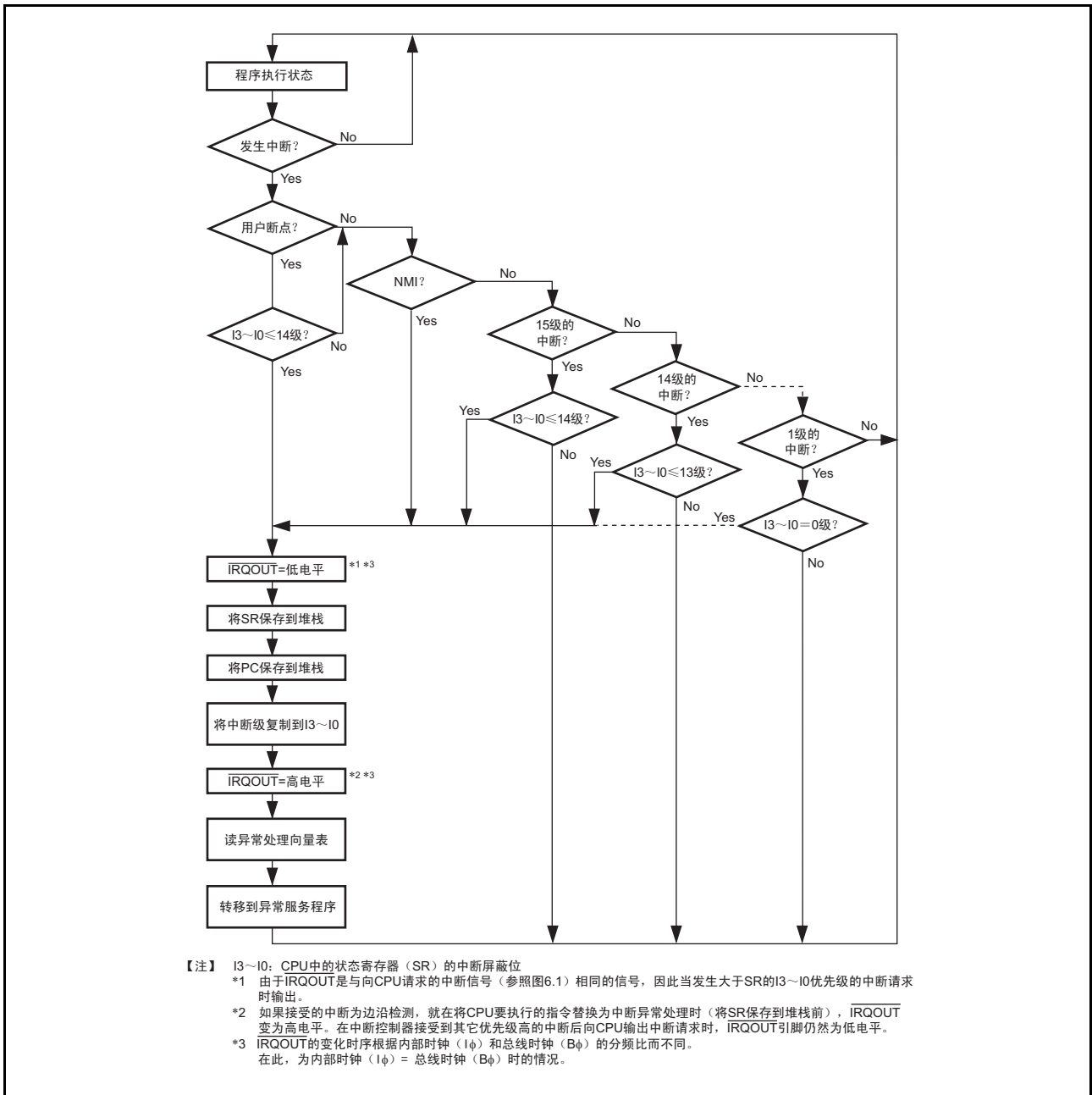


图 6.3 中断运行流程

## 6.6.2 中断异常处理结束后的堆栈状态

中断异常处理结束后的堆栈状态如图 6.4 所示。

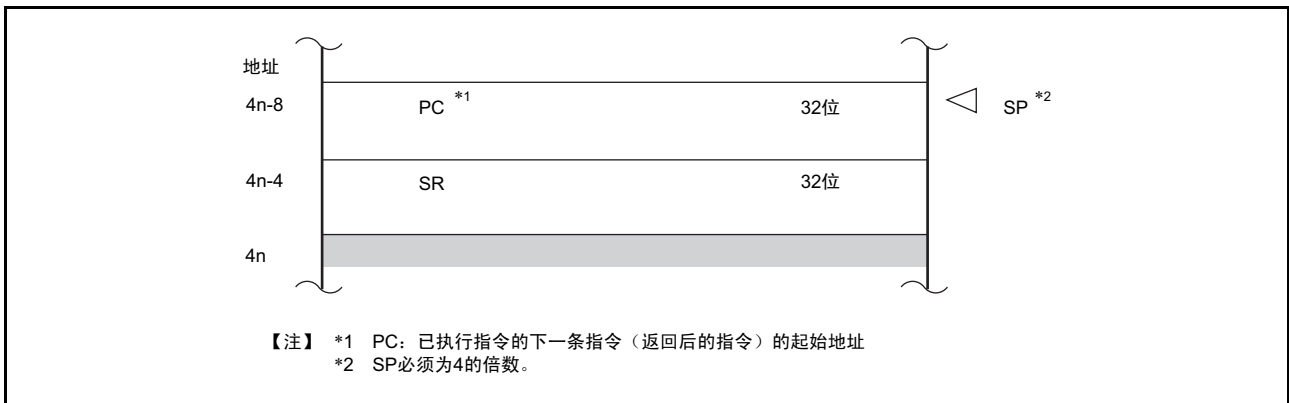


图 6.4 中断异常处理结束后的堆栈状态

## 6.7 中断响应时间

从发生中断请求到进行中断异常处理、开始取异常服务程序的起始指令为止的时间（中断响应时间）如表 6.4 所示。

表 6.4 中断响应时间

项 目		状态数			备 注
		NMI	IRQ	外围模块	
DMAC/DTC 的启动判断		—	$2 \times Bcyc$	$1 \times Pcyc$	
优先级的判断以及与 SR 屏蔽位的比较时间		$1 \times lcyc + 2 \times Pcyc$	$1 \times lcyc + 1 \times Pcyc$	$1 \times lcyc + 2 \times Pcyc$	
到 CPU 执行中的顺序结束为止的等待时间		$X (\geq 0)$			最长的顺序是中断异常处理和地址错误异常处理，为 $X = 7 \times lcyc + m1 + m2 + m3 + m4$ 。但是，在后续指令为屏蔽中断指令时，可能更长。
从开始中断异常处理到开始取异常服务程序起始指令的时间		$8 \times lcyc + m1 + m2 + m3$			保存 SR、PC 以及取向量地址。
响应时间	合计	$9 \times lcyc + 2 \times Pcyc + m1 + m2 + m3 + X$	$9 \times lcyc + 1 \times Pcyc + 2 \times Bcyc + m1 + m2 + m3 + X$	$9 \times lcyc + 3 \times Pcyc + m1 + m2 + m3 + X$	
	最小时 *	$12 \times lcyc + 2 \times Pcyc$	$12 \times lcyc + 1 \times Pcyc + 2 \times Bcyc$	$12 \times lcyc + 3 \times Pcyc$	SR、PC 和向量表全部在内部 RAM 时。
	最大时	$16 \times lcyc + 2 \times Pcyc + 2(m1 + m2 + m3) + m4$	$16 \times lcyc + 1 \times Pcyc + 2 \times Bcyc + 2(m1 + m2 + m3) + m4$	$16 \times lcyc + 3 \times Pcyc + 2(m1 + m2 + m3) + m4$	

【注】 m1 ~ m4 为下述存储器存取时需要的状态数：

m1: 保存 SR (写长字)

m2: 保存 PC (写长字)

m3: 读向量地址 (读长字)

m4: 取中断服务程序的起始指令

\* 当  $m1 = m2 = m3 = m4 = 1 \times lcyc$  时

## 6.8 通过中断请求信号的数据传送

能通过中断请求信号进行以下的数据传送。

- 只启动 DMAC，不产生 CPU 中断
- 只启动 DTC，CPU 中断取决于 DTC 的设置

中断源中被指定为 DMAC 启动源的中断源被屏蔽，不输入到 INTC。屏蔽条件如下：

屏蔽条件 = 源选择 (CH0)+ 源选择 (CH1)+ 源选择 (CH2)+ 源选择 (CH3)

当对应的 DTCE 位是 1 时，INTC 屏蔽 CPU 中断。DTCE 清除条件和中断源标志清除条件如下：

DTCE 清除条件 = DTC 传送结束 · DTCECLR

中断源标志清除条件 = DTC 传送结束 · DTCECLR+DMAC 传送结束

其中，DTCECLR = DISEL+ 计数器 0

控制框图如图 6.5 和图 6.6 所示。

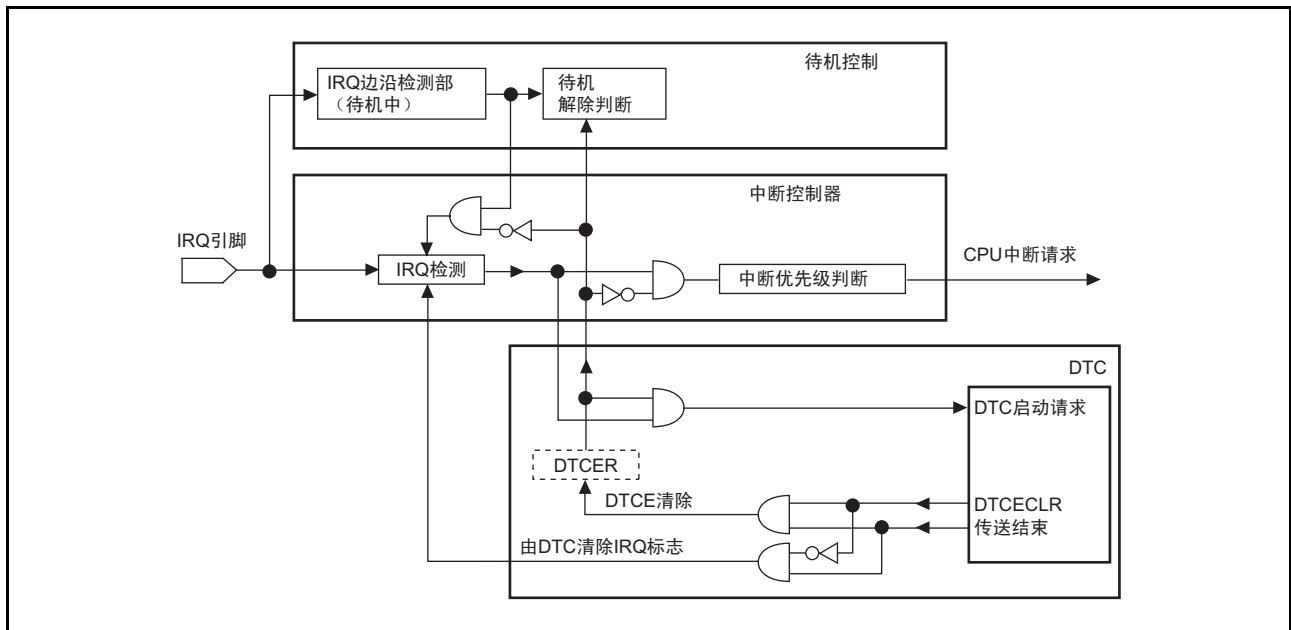


图 6.5 IRQ 中断控制框图

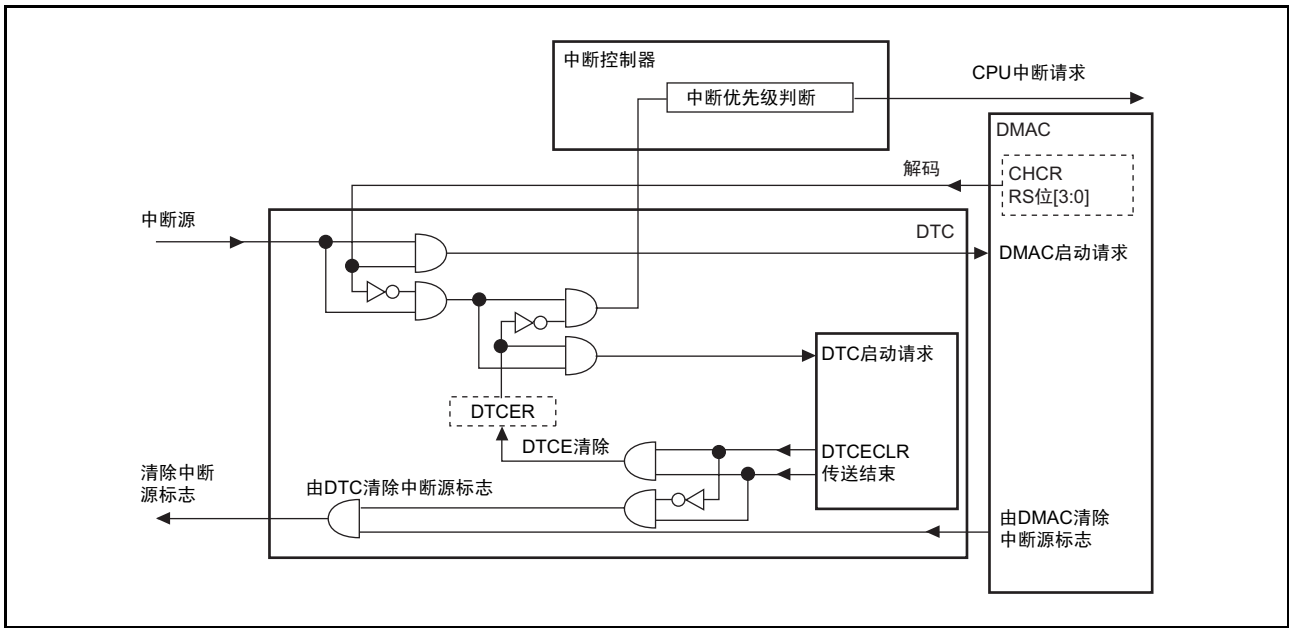


图 6.6 外围模块的中断控制框图

### 6.8.1 中断请求信号作为 DTC 启动源和 CPU 中断源而不作为 DMAC 启动源的情况

1. 不能选择 DMAC 的启动源。
2. 将对应 DTC 的 DTCE 位和 DIESEL 位置 1。
3. 当发生中断时，将启动请求发送到 DTC。
4. 如果进行数据传送，DTC 就将 DTCE 位清 0，向 CPU 请求中断，但是不清除启动源。
5. CPU 通过中断处理程序清除中断源，然后确认传送计数器的值。在传送计数器的值  $\neq 0$  时，将 DTCE 位置 1，允许下一次的数据传送；如果传送计数器的值  $= 0$ ，就通过中断处理程序进行所需的结束处理。

### 6.8.2 中断请求信号作为 DMAC 启动源而不作为 CPU 中断源和 DTC 启动源的情况

1. 选择 DMAC 的启动源。与中断优先级寄存器的设定和 DTC 寄存器的设定无关，屏蔽 CPU 中断源和 DTC 启动源。
2. 当发生中断时，将启动源送给 DMAC。
3. 在传送时，DMAC 清除启动源。

### 6.8.3 中断请求信号作为 DTC 启动源而不作为 CPU 中断源和 DMAC 启动源的情况

1. 不能选择 DMAC 的启动源。
2. 将对应 DTC 的 DTCE 位置 1，将 DIESEL 位清 0。
3. 当发生中断时，将启动源送给 DTC。
4. 如果进行数据传送，DTC 就清除启动源。由于 DTCE 位保持为 1，所以不向 CPU 请求中断。
5. 但是，在传送计数器  $= 0$  时，将 DTCE 位清 0，向 CPU 请求中断。
6. CPU 通过中断处理程序进行所需的结束处理。

#### 6.8.4 中断请求信号作为 CPU 中断源而不作为 DTC 启动源和 DMAC 启动源的情况

1. 不能选择 DMAC 的启动源。
2. 将对应 DTC 的 DTCE 位清 0。
3. 当发生中断时，向 CPU 请求中断。
4. CPU 通过中断处理程序清除中断源，进行所需的处理。

#### 6.9 使用注意事项

必须在中断处理程序中清除中断源标志。为了不错误地再次接受应该被清除的中断源，必须在清除后读中断源标志，并在确认中断源标志已被清除后执行 RTE 指令。



## 第 7 章 用户断点控制器 (UBC)

用户断点控制器 (UBC) 提供程序的调试功能。能通过使用此功能容易地建立自监视调试程序, 即使不使用内部电路仿真器, 也能通过本 LSI 简单地调试程序。能给 UBC 设定的断点条件有取指令或者数据的读写、数据长度、数据内容、地址值以及取指令时的停止时序。

另外, 掩模型 ROM 只有 L 总线取指令地址断点 (2 个通道)。

### 7.1 特点

1. 能设定如下的断点比较条件:

断点通道数: 2 个通道 (通道 A 和 B)

能将用户断点设定为通道 A、B 独立或者连续 (顺序) 的条件。

(顺序断点的设定: 在通道 A 的断点条件一致后, 通道 B 的断点条件一致, 并且两者不发生在同一总线周期时。)

- 地址  
能按位屏蔽 32 位 (地址) 的比较。能选择 L 地址总线 (LAB) 或者 I 地址总线 (IAB)。
  - 数据  
能屏蔽 32 位。  
能选择 L 数据总线 (LDB) 或者 I 数据总线 (IDB)。
  - 总线周期  
取指令或者存取数据。
  - 读或者写
  - 操作数长度  
支持字节、字和长字。
2. 能执行用户指定的用户断点中断的异常处理程序。
  3. 在取指令周期中, 能将用户断点设定在指令执行前或在指令执行后。
  4. 作为断点条件 (只对通道 B), 能指定的重复次数最多为  $2^{12}-1$ 。
  5. 支持 4 组转移源缓冲器 / 转移目标缓冲器 (支持 E10A 全功能的 F-ZTAT 版为 8 组)。

UBC 的框图如图 7.1 所示。

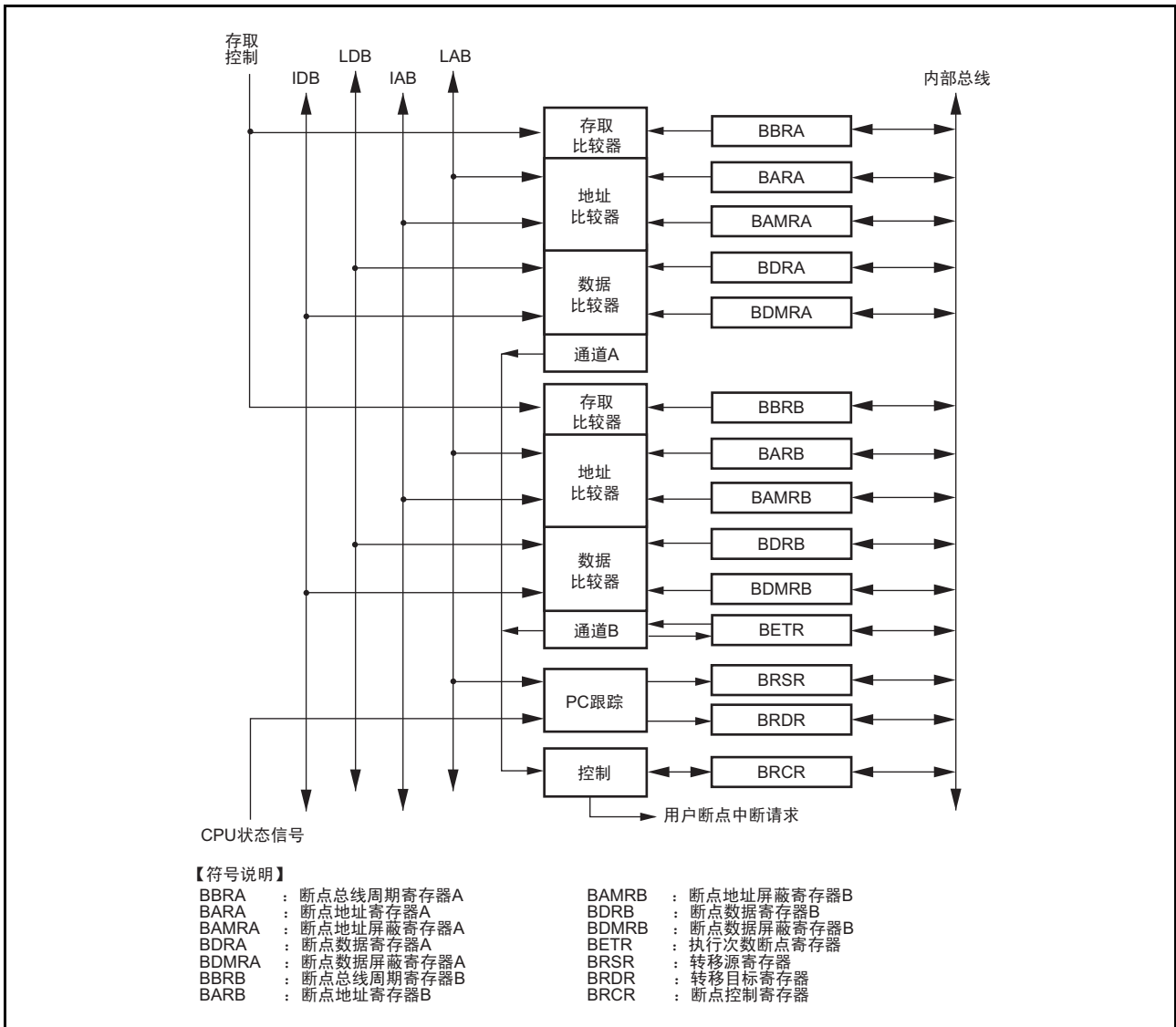


图 7.1 UBC 的框图

## 7.2 输入 / 输出引脚

UBC 的引脚如表 7.1 所示。

表 7.1 引脚结构

名称	引脚名	输入 / 输出	功能
用户断点触发输出	UBCTRG	输出	是 UBC 条件一致的触发输出引脚。

## 7.3 寄存器说明

UBC 有以下寄存器。有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 27 章 寄存器一览”。

表 7.2 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
断点地址寄存器 A	BARA	R/W	H'00000000	H'FFFFFF300	32
断点地址屏蔽寄存器 A	BAMRA	R/W	H'00000000	H'FFFFFF304	32
断点总线周期寄存器 A	BBRA	R/W	H'0000	H'FFFFFF308	16
断点数据寄存器 A	BDRA*	R/W	H'00000000	H'FFFFFF310	32
断点数据屏蔽寄存器 A	BDMRA*	R/W	H'00000000	H'FFFFFF314	32
断点地址寄存器 B	BARB	R/W	H'00000000	H'FFFFFF320	32
断点地址屏蔽寄存器 B	BAMRB	R/W	H'00000000	H'FFFFFF324	32
断点总线周期寄存器 B	BBRB	R/W	H'0000	H'FFFFFF328	16
断点数据寄存器 B	BDRB*	R/W	H'00000000	H'FFFFFF330	32
断点数据屏蔽寄存器 B	BDMRB*	R/W	H'00000000	H'FFFFFF334	32
断点控制寄存器	BRCR	R/W	H'00000000	H'FFFFFF3C0	32
转移源寄存器	BRSR*	R	H'0xxxxxxx	H'FFFFFF3D0	32
转移目标寄存器	BRDR*	R	H'0xxxxxxx	H'FFFFFF3D4	32
执行次数断点寄存器	BETR*	R/W	H'0000	H'FFFFFF3DC	16

【注】 \* 只限 F-ZTAT 版。

### 7.3.1 断点地址寄存器 A (BARA)

BARA 是 32 位可读写寄存器，指定作为通道 A 断点条件的地址。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAA31	BAA30	BAA29	BAA28	BAA27	BAA26	BAA25	BAA24	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
31 ~ 0	BAA31 ~ BAA0	全 0	R/W	断点地址 A 保存指定通道 A 断点条件的 LAB 或者 IAB 的地址。

### 7.3.2 断点地址屏蔽寄存器 A (BAMRA)

BAMRA 是 32 位可读写寄存器，在 BARA 指定的断点地址位中指定要屏蔽的位。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAMA31	BAMA30	BAMA29	BAMA28	BAMA27	BAMA26	BAMA25	BAMA24	BAMA23	BAMA22	BAMA21	BAMA20	BAMA19	BAMA18	BAMA17	BAMA16
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAMA15	BAMA14	BAMA13	BAMA12	BAMA11	BAMA10	BAMA9	BAMA8	BAMA7	BAMA6	BAMA5	BAMA4	BAMA3	BAMA2	BAMA1	BAMA0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
31 ~ 0	BAMA31 ~ BAMA0	全 0	R/W	断点地址屏蔽 A 在 BARA (BAA31 ~ BAA0) 指定的通道 A 的断点地址位中，指定要屏蔽的位。 0: 断点地址位 BAA <sub>n</sub> 包含在断点条件中 1: 断点地址位 BAA <sub>n</sub> 被屏蔽，不包含在断点条件中 【注】 n=31 ~ 0

## 7.3.3 断点总线周期寄存器 A (BBRA)

BBRA 是 16 位可读写寄存器，作为通道 A 的断点条件，指定 (1) I 总线周期的总线主控 (2) L 总线周期或者 I 总线周期 (3) 取指令或者数据存取 (4) 读或者写 (5) 操作数长度。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	CPA2*	CPA1*	CPA0*	CDA1*	CDA0	IDA1*	IDA0	RWA1*	RWA0	SZA1*	SZA0*
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

【注】\* 在掩模型 ROM 版、无 ROM 版中为保留位。读写的值总是 0。

位	位名	初始值	R/W	说 明
15 ~ 11	—	全 0	R	保留位 读写的值总是 0。
10 9 8	CPA2* CPA1* CPA0*	0 0 0	R/W R/W R/W	I 总线的总线主控选择 A 当选择 I 总线作为通道 A 断点条件的总线周期时，选择总线主控；当选择 L 总线作为总线周期时，本位无效。 000：不进行条件比较 xx1：断点条件包含 CPU 周期 x1x：断点条件包含 DMAC 周期 1xx：断点条件包含 DTC 周期
7 6	CDA1* CDA0	0 0	R/W R/W	L 总线周期 / I 总线周期的选择 A 将 L 总线周期或者 I 总线周期选择为通道 A 断点条件的总线周期。 00：不进行条件比较 01：断点条件为 L 总线周期 10：断点条件为 I 总线周期 11：断点条件为 L 总线周期
5 4	IDA1* IDA0	0 0	R/W R/W	取指令 / 数据存取的选择 A 将取指令周期或者数据存取周期选择为通道 A 断点条件的总线周期。 00：不进行条件比较 01：断点条件为取指令周期 10：断点条件为数据存取周期 11：断点条件为取指令周期或者数据存取周期
3 2	RWA1* RWA0	0 0	R/W R/W	读 / 写的选择 A 将读周期或者写周期选择为通道 A 断点条件的总线周期。 00：不进行条件比较 01：断点条件为读周期 10：断点条件为写周期 11：断点条件为读或者写周期
1 0	SZA1* SZA0*	0 0	R/W R/W	操作数长度的选择 A 选择总线周期的操作数长度作为通道 A 的断点条件。 00：断点条件不包含操作数长度 01：断点条件为字节存取 10：断点条件为字存取 11：断点条件为长字存取 【注】在指定操作数长度时，地址边界和操作数长度必须一致。

【符号说明】x：Don't care

【注】\* 在掩模型 ROM 版、无 ROM 版中为保留位。读写的值总是 0。

### 7.3.4 断点数据寄存器 A (BDRA) (只限 F-ZTAT 版)

BDRA 是 32 位可读写寄存器。作为断点条件 A 对象的数据总线有 2 种，通过断点总线周期寄存器 A (BBRA) 的控制位 CDA1 和 CDA0 来选择。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BDA31	BDA30	BDA29	BDA28	BDA27	BDA26	BDA25	BDA24	BDA23	BDA22	BDA21	BDA20	BDA19	BDA18	BDA17	BDA16
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BDA15	BDA14	BDA13	BDA12	BDA11	BDA10	BDA9	BDA8	BDA7	BDA6	BDA5	BDA4	BDA3	BDA2	BDA1	BDA0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 0	BDA31 ~ BDA0	全 0	R/W	断点数据位 A 保存指定通道 A 断点条件的数据。 当通过 BBRA 选择 I 总线时，就给 BDA31 ~ BDA0 指定 IDB 的断点数据；当选择 L 总线时，就给 BDA31 ~ BDA0 指定 LDB 的断点数据。

- 【注】
1. 如果断点条件包含数据总线的值，就必须指定操作数长度。
  2. 如果将字节长度指定为断点条件，作为 BDRA 中的断点数据，就必须给 bit15 ~ 8 和 bit7 ~ 0 设定相同的字节数据。

### 7.3.5 断点数据屏蔽寄存器 A (BDMRA) (只限 F-ZTAT 版)

BDMRA 是 32 位可读写寄存器，在 BDRA 指定的断点数据位中指定要屏蔽的位。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BDMA31	BDMA30	BDMA29	BDMA28	BDMA27	BDMA26	BDMA25	BDMA24	BDMA23	BDMA22	BDMA21	BDMA20	BDMA19	BDMA18	BDMA17	BDMA16
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BDMA15	BDMA14	BDMA13	BDMA12	BDMA11	BDMA10	BDMA9	BDMA8	BDMA7	BDMA6	BDMA5	BDMA4	BDMA3	BDMA2	BDMA1	BDMA0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 0	BDMA31 ~ BDMA0	全 0	R/W	断点数据屏蔽 A 在 BDRA (BDA31 ~ BDA0) 指定的通道 A 的断点数据位中，指定要屏蔽的位。 0: 断点数据位 BDA <sub>n</sub> 包含在断点条件中 1: 断点数据位 BDA <sub>n</sub> 被屏蔽，不包含在断点条件中 【注】 n=31 ~ 0

- 【注】
1. 如果断点条件包含数据总线的值，就必须指定操作数长度。
  2. 如果将字节长度指定为断点条件，作为 BDMRA 中的断点屏蔽数据，就必须给 bit15 ~ 8 和 bit7 ~ 0 设定相同的字节数据。

### 7.3.6 断点地址寄存器 B (BARB)

BARB 是 32 位可读写寄存器，指定作为通道 B 断点条件的地址。作为断点条件 B 对象的地址总线有 2 种，通过断点总线周期寄存器 B (BBRB) 的控制位 CDB1 和 CDB0 来选择。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAB31	BAB30	BAB29	BAB28	BAB27	BAB26	BAB25	BAB24	BAB23	BAB22	BAB21	BAB20	BAB19	BAB18	BAB17	BAB16
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAB15	BAB14	BAB13	BAB12	BAB11	BAB10	BAB9	BAB8	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1	BAB0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
31 ~ 0	BAB31 ~ BAB0	全 0	R/W	断点地址 B 指定作为通道 B 断点条件的地址。 当通过 BBRB 选择 I 总线或者 L 总线时，就给 BAB31 ~ BAB0 指定 IAB 或者 LAB 的地址。

### 7.3.7 断点地址屏蔽寄存器 B (BAMRB)

BAMRB 是 32 位可读写寄存器，在 BARB 指定的断点地址位中指定要屏蔽的位。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAMB31	BAMB30	BAMB29	BAMB28	BAMB27	BAMB26	BAMB25	BAMB24	BAMB23	BAMB22	BAMB21	BAMB20	BAMB19	BAMB18	BAMB17	BAMB16
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAMB15	BAMB14	BAMB13	BAMB12	BAMB11	BAMB10	BAMB9	BAMB8	BAMB7	BAMB6	BAMB5	BAMB4	BAMB3	BAMB2	BAMB1	BAMB0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
31 ~ 0	BAMB31 ~ BAMB0	全 0	R/W	断点地址屏蔽 B 在 BARB (BAB31 ~ BAB0) 指定的通道 B 的断点地址位中，指定要屏蔽的位 0: 断点地址位 BABn 包含在断点条件中 1: 断点地址位 BABn 被屏蔽，不包含在断点条件中 【注】 n=31 ~ 0

### 7.3.8 断点数据寄存器 B (BDRB) (只限 F-ZTAT 版)

BDRB 是 32 位可读写寄存器。作为断点条件 B 对象的数据总线有 2 种，通过断点总线周期寄存器 B (BBRB) 的控制位 CDB1 和 CDB0 来选择。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BDB31	BDB30	BDB29	BDB28	BDB27	BDB26	BDB25	BDB24	BDB23	BDB22	BDB21	BDB20	BDB19	BDB18	BDB17	BDB16
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BDB15	BDB14	BDB13	BDB12	BDB11	BDB10	BDB9	BDB8	BDB7	BDB6	BDB5	BDB4	BDB3	BDB2	BDB1	BDB0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 0	BDB31 ~ BDB0	全 0	R/W	断点数据位 B 保存指定通道 B 断点条件的数据。 当通过 BBRB 选择 I 总线时，就给 BDB31 ~ BDB0 指定 IDB 的断点数据；当选择 L 总线，就给 BDB31 ~ BDB0 指定 LDB 的断点数据。

- 【注】
1. 如果断点条件包含数据总线的值，就指定操作数长度。
  2. 如果将字节长度指定为断点条件，作为 BDRB 的断点数据，就必须给 bit15 ~ 8 和 bit7 ~ 0 设定相同的字节数据。

### 7.3.9 断点数据屏蔽寄存器 B (BDMRB) (只限 F-ZTAT 版)

BDMRB 是 32 位可读写寄存器，在 BDRB 指定的断点数据位中指定要屏蔽的位。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BDMB31	BDMB30	BDMB29	BDMB28	BDMB27	BDMB26	BDMB25	BDMB24	BDMB23	BDMB22	BDMB21	BDMB20	BDMB19	BDMB18	BDMB17	BDMB16
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BDMB15	BDMB14	BDMB13	BDMB12	BDMB11	BDMB10	BDMB9	BDMB8	BDMB7	BDMB6	BDMB5	BDMB4	BDMB3	BDMB2	BDMB1	BDMB0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 0	BDMB31 ~ BDMB0	全 0	R/W	断点数据屏蔽 B 在 BDRB (BDB31 ~ BDB0) 指定的通道 B 的断点数据位中，指定要屏蔽的位。 0: 断点数据位 BDBn 包含在断点条件中 1: 断点数据位 BDBn 被屏蔽，不包含在断点条件中 【注】 n=31 ~ 0

- 【注】
1. 如果断点条件包含数据总线的值，就必须指定操作数长度。
  2. 如果将字节长度指定为断点条件，作为 BDMRB 中的断点屏蔽数据，就必须给 bit15 ~ 8 和 bit7 ~ 0 设定相同的字节数据。



## 7.3.10 断点总线周期寄存器 B (BBRB)

BBRB 是 16 位可读写寄存器，作为通道 B 的断点条件，指定 (1) I 总线周期的总线主控 (2) L 总线周期或者 I 总线周期 (3) 取指令或者数据存取 (4) 读或者写 (5) 操作数长度。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	CPB2*	CPB1*	CPB0*	CDB1*	CDB0	IDB1*	IDB0	RWB1*	RWB0	SZB1*	SZAB*
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

【注】\* 在掩模型 ROM 版、无 ROM 版中为保留位。读写的值总是 0。

位	位名	初始值	R/W	说 明
15 ~ 11	—	全 0	R	保留位 读写的值总是 0。
10 9 8	CPB2* CPB1* CPB0*	0 0 0	R/W R/W R/W	I 总线的总线主控选择 B 当选择 I 总线作为通道 B 断点条件的总线周期时，选择总线主控；当选择 L 总线作为总线周期时，本位无效。 000: 不进行条件比较 xx1: 断点条件中包含 CPU 周期 x1x: 断点条件中包含 DMAC 周期 1xx: 断点条件中包含 DTC 周期
7 6	CDB1* CDB0	0 0	R/W R/W	L 总线周期 / I 总线周期的选择 B 将 L 总线周期或者 I 总线周期选择为通道 B 断点条件的总线周期。 00: 不进行条件比较 01: 断点条件为 L 总线周期 10: 断点条件为 I 总线周期 11: 断点条件为 L 总线周期
5 4	IDB1* IDB0	0 0	R/W R/W	取指令 / 数据存取的选择 B 将取指令周期或者数据存取周期选择为通道 B 断点条件的总线周期。 00: 不进行条件比较 01: 断点条件为取指令周期 10: 断点条件为数据存取周期 11: 断点条件为取指令周期或者数据存取周期
3 2	RWB1* RWB0	0 0	R/W R/W	读 / 写的选择 B 将读周期或者写周期选择为通道 B 断点条件的总线周期。 00: 不进行条件比较 01: 断点条件为读周期 10: 断点条件为写周期 11: 断点条件为读或者写周期
1 0	SZB1* SZB0*	0 0	R/W R/W	操作数长度的选择 B 选择总线周期的操作数长度作为通道 B 的断点条件。 00: 断点条件不包含操作数长度 01: 断点条件为字节存取 10: 断点条件为字存取 11: 断点条件为长字存取 【注】在指定操作数长度时，地址边界和操作数长度必须一致。

【符号说明】x: Don't care

【注】\* 在掩模型 ROM 版、无 ROM 版中为保留位。读写的值总是 0。

## 7.3.11 断点控制寄存器 (BRCR)

BRCR 设定以下条件:

1. 将通道 A、B 指定为 2 个独立的通道条件或者 1 个连续的条件。
2. 将用户断点设定在指令执行前或者在指令执行后。
3. 指定通道 B 的比较条件是否包含执行次数。
4. 指定通道 A、B 的比较条件是否包含数据总线的值。
5. 允许 PC 跟踪。
6. 选择  $\overline{\text{UBCTRG}}$  输出的脉宽
7. 指定是否在通道 A、B 的比较条件一致时请求用户断点中断。

BRCR 是 32 位可读写寄存器, 有断点条件一致的标志和设定各种断点条件的位。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	UTRGW[1:0]	UBIDB	-	UBIDA	-	
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCM FCA	SCM FCB	SCM FDA	SCM FDB	PCTE	PCBA	-	-	DBEA	PCBB	DBEB	-	SEQ	-	-	ETBE
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R	R/W	R	R	R/W

位	位名	初始值	R/W	说 明
31 ~ 22	—	全 0	R	保留位 读写的值总是 0。
21 ~ 20	UTRGW [1:0]	00	R/W	$\overline{\text{UBCTRG}}$ 输出的脉冲宽度选择 选择断点条件一致时的 $\overline{\text{UBCTRG}}$ 输出的脉宽。 00: 禁止设定 01: 将 $\overline{\text{UBCTRG}}$ 输出的脉宽设定为 $3 \sim 4t_{\text{Bcyc}}$ 10: 将 $\overline{\text{UBCTRG}}$ 输出的脉宽设定为 $7 \sim 8t_{\text{Bcyc}}$ 11: 将 $\overline{\text{UBCTRG}}$ 输出的脉宽设定为 $15 \sim 16t_{\text{Bcyc}}$ 【注】 $t_{\text{Bcyc}}$ 表示外部时钟 ( $B\phi = \text{CK}$ ) 的频率。
19	UBIDB	0	R/W	用户断点的禁止 B 选择在满足通道 B 的断点条件时, 是否禁止用户断点中断请求。 0: 在满足断点条件时, 允许用户断点中断请求 1: 在满足断点条件时, 禁止用户断点中断请求
18	—	0	R	保留位 读写的值总是 0。
17	UBIDA	0	R/W	用户断点的禁止 A 选择在满足通道 A 的断点条件时, 是否禁止用户断点中断请求。 0: 在满足断点条件时, 允许用户断点中断请求 1: 在满足断点条件时, 禁止用户断点中断请求
16	—	0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说 明
15	SCMFCA	0	R/W	L 总线周期条件一致的标志 A 当满足给通道 A 设定的断点条件的 L 总线周期条件时, 就将此标志置 1。 通过给此位写 0 清除此标志。 0: 通道 A 的 L 总线周期条件不一致 1: 通道 A 的 L 总线周期条件一致
14	SCMFCB	0	R/W	L 总线周期条件一致的标志 B 当满足给通道 B 设定的断点条件的 L 总线周期条件时, 就将此标志置 1。 通过给此位写 0 清除此标志。。 0: 通道 B 的 L 总线周期条件不一致 1: 通道 B 的 L 总线周期条件一致
13	SCMFDA	0	R/W	I 总线周期条件一致的标志 A 当满足给通道 A 设定的断点条件的 I 总线周期条件时, 就将此标志置 1。 通过给此位写 0 清除此标志。 0: 通道 A 的 I 总线周期条件不一致 1: 通道 A 的 I 总线周期条件一致
12	SCMFDB	0	R/W	I 总线周期条件一致的标志 B 当满足给通道 B 设定的断点条件的 I 总线周期条件时, 就将此标志置 1。 通过给此位写 0 清除此标志。 0: 通道 B 的 I 总线周期条件不一致 1: 通道 B 的 I 总线周期条件一致
11	PCTE	0	R/W	PC 跟踪的允许 0: 禁止 PC 跟踪 1: 允许 PC 跟踪
10	PCBA	0	R/W	PC 断点选择 A 选择通道 A 取指令周期的断点时序是在指令执行前还是在指令执行后。 0: 将通道 A 的 PC 断点设定在指令执行前 1: 将通道 A 的 PC 断点设定在指令执行后
9、8	—	全 0	R	保留位 读写的值总是 0。
7	DBEA	0	R/W	数据中断允许 A 选择通道 A 的断点条件是否包含数据总线条件。 0: 通道 A 的断点条件不包含数据总线条件 1: 通道 A 的断点条件包含数据总线条件
6	PCBB	0	R/W	PC 中断选择 B 选择通道 B 取指令周期的断点时序是在指令执行前还是在指令执行后。 0: 将通道 B 的 PC 断点设定在指令执行前 1: 将通道 B 的 PC 断点设定在指令执行后
5	DBEB	0	R/W	数据中断允许 B 选择通道 B 的断点条件是否包含数据总线条件。 0: 通道 B 的断点条件不包含数据总线条件 1: 通道 B 的断点条件包含数据总线条件
4	—	0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说 明
3	SEQ	0	R/W	顺序条件选择 选择通道 A 和 B 是 2 个独立条件还是连续的 1 个条件。 0: 在独立的条件下比较通道 A 和通道 B 1: 在连续的条件下比较通道 A 和通道 B (先是通道 A, 其次是通道 B)
2、1	—	全 0	R	保留位 读写的值总是 0。
0	ETBE	0	R/W	执行次数中断的允许 只对通道 B 将执行次数断点条件置为有效。如果此位是 1, 就在断点条件的发生次数和 BETR 寄存器指定的执行次数相等时, 请求用户断点中断。 0: 将通道 B 的执行次数断点条件置为无效 1: 将通道 B 的执行次数断点条件置为有效

### 7.3.12 执行次数断点寄存器 (BETR) (只限 F-ZTAT 版)

BETR 是 16 位可读写寄存器。如果将通道 B 的执行次数断点条件置为有效, 此寄存器就指定暂停的次数。最大值为  $2^{12}-1$ 。每满足一次断点条件, BETR 就减 1。如果 BETR 为 H'0001 并且满足断点条件, 就请求用户断点中断。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	BET[11:0]											
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 12	—	全 0	R	保留位 读写的值总是 0。
11 ~ 0	BET[11:0]	全 0	R/W	执行次数

## 7.3.13 转移源寄存器 (BRSR) (只限 F-ZTAT 版)

BRSR 是 32 位只读寄存器，保存转移源指令地址的 bit27 ~ 0。BRSR 有 1 个在发生转移时被置 1 的标志位，此标志位在读 BRSR、或者将 PC 跟踪禁止状态设定为允许、上电复位或者手动复位时被清 0，而其他位在复位时不被初始化。4 个（支持 E10A 全功能的 F-ZTAT 版是 8 个）BRSR 寄存器具有队列结构，被保存的寄存器每转移一次就移位一次。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SVF	-	-	-	BSA27	BSA26	BSA25	BSA24	BSA23	BSA22	BSA21	BSA20	BSA19	BSA18	BSA17	BSA16
初始值:	0	0	0	0	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BSA15	BSA14	BSA13	BSA12	BSA11	BSA10	BSA9	BSA8	BSA7	BSA6	BSA5	BSA4	BSA3	BSA2	BSA1	BSA0
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
31	SVF	0	R	BRSR 有效标志 表示转移源的地址是否已被保存。此标志在发生转移时被置 1；在读 BRSR、或者将 PC 跟踪禁止状态设定为允许或者上电复位的条件下被清 0。 0: BRSR 寄存器的值无效 1: BRSR 寄存器的值有效
30 ~ 28	—	全 0	R	保留位 读写的值总是 0。
27 ~ 0	BSA27 ~ BSA0	不定	R	转移源地址 这些位保存转移源地址的 bit27 ~ 0。

### 7.3.14 转移目标寄存器 (BRDR) (只限 F-ZTAT 版)

BRDR 是 32 位只读寄存器，保存转移目标指令地址的 bit27 ~ 0。BRDR 有 1 个在发生转移时被置 1 的标志位，此标志位在读 BRDR、或者将 PC 跟踪禁止状态设定为允许、上电复位或者手动复位时被清 0，而其他位在复位时不被初始化。4 个（支持 E10A 全功能的 F-ZTAT 版是 8 个）BRDR 寄存器具有队列结构，被保存的寄存器每转移一次就移位一次。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DVF	-	-	-	BDA27	BDA26	BDA25	BDA24	BDA23	BDA22	BDA21	BDA20	BDA19	BDA18	BDA17	BDA16
初始值:	0	0	0	0	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BDA15	BDA14	BDA13	BDA12	BDA11	BDA10	BDA9	BDA8	BDA7	BDA6	BDA5	BDA4	BDA3	BDA2	BDA1	BDA0
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
31	DVF	0	R	BRDR 有效标志 表示转移目标地址是否已被保存。此标志在发生转移时被置 1；在读 BRDR、或者将 PC 跟踪禁止状态设定为允许或者上电复位的条件下被清 0。 0: BRDR 寄存器的值无效 1: BRDR 寄存器的值有效
30 ~ 28	—	全 0	R	保留位 读写的值总是 0。
27 ~ 0	BDA27 ~ BDA0	不定	R	转移目标地址 这些位保存转移目标地址的 bit27 ~ 0。

## 7.4 运行说明

### 7.4.1 用户断点运行的流程

从设定断点条件到用户断点异常处理的运行流程如下：

1. 将断点地址设定在断点地址寄存器 (BARA 或者 BARB)、要屏蔽的地址设定在断点地址屏蔽寄存器 (BAMRA 或者 BAMRB)、断点数据设定在断点数据寄存器 (BDRA 或者 BDRB)、要屏蔽的数据设定在断点数据屏蔽寄存器 (BDMRA 或者 BDMRB)、总线断点条件设定在断点总线周期寄存器 (BBRA 或者 BBRB)。BBRA 或者 BBRB 的 3 对控制位 (即 L 总线周期 / I 总线周期的选择位、取指令 / 数据存取的选择位和读 / 写的选择位) 中的任意 1 对是 B'00，就不发生用户断点的暂停。将中断控制设定在 BRCCR 的位。必须在设定完其他全部的中断相关寄存器后，设定 BBRA 或者 BBRB。
2. 当满足断点条件时，UBC 在将用户断点中断请求送给 CPU 的同时，将各通道的 L 总线条件一致标志 (SCMFCA 或者 SCMFCA) 和 I 总线条件一致标志 (SCMFDA 或者 SCMFDB) 置位。
3. 能使用对应的条件一致标志 (SCMFCA、SCMFDA、SCMFCA、SCMFDB) 检查设定条件是否一致。根据条件一致将标志置位，而不能复位。为了能再次使用标志，必须先写 0，然后将标志复位。
4. 有时几乎同时发生通道 A 和通道 B 设定的断点条件一致，此时即使送给 CPU 的用户断点中断请求只有 1 个，也有可能设定 2 个条件一致标志。

5. 当选择 I 总线作为断点条件时，必须注意以下事项：
  - CPU 连接 DMAC 和 DTC。UBC 监视通过 BBRA 寄存器的 CPA2 ~ CPA0 位和 BBRB 寄存器的 CPB2 ~ CPB0 位选择的总线主控所生成的总线周期，并进行条件一致比较。
  - 将由 CPU 的 L 总线上的取指令引起的 I 总线周期（包含读周期）定义为 I 总线上的取指令周期，其他定义为数据存取周期。
  - DMAC/DTC 产生的 I 总线周期只为数据存取周期。
  - 当给 I 总线设定断点条件时，即使在由 CPU 执行指令引起的 I 总线周期中条件一致，也不能决定是在哪条指令接受用户断点。

#### 7.4.2 取指令周期的用户断点

1. 如果将“不包含 L 总线 / 取指令 / 读 / 字、长字或者操作数长度”设定到断点总线周期寄存器（BBRA 或者 BBRB），断点条件就为 L 总线的取指令周期。能通过对对应通道的中断控制寄存器（BRCR）的 PCBA 或者 PCBB 位，选择是在指令执行前还是在指令执行后发生用户断点的暂停。在将取指令周期设定为断点条件时，必须将断点地址寄存器（BARA 或者 BARB）的 LSB 清零。如果此位被置 1，就不发生用户断点的暂停。
2. 如果在将取指令的用户断点设定在执行该指令前的状态下断点条件一致，就取出指令，然后在要执行指令前发生用户断点的暂停。因此，此功能不能用于因预取（在转移或者中断转移中取出但不执行的指令）而取出的指令。如果对延迟转移指令的延迟槽设定此类断点条件，就在执行延迟转移指令前发生用户断点的暂停。

**【注】** 如果延迟转移指令不发生转移，就不将后续的指令视为延迟槽。

3. 如果在断点条件中将用户断点设定在指令执行后，就在执行和断点条件一致的指令后并在执行下一条指令前发生用户断点的暂停。和执行前发生用户断点暂停的情况一样，此功能不能用于因预取而取出的指令。如果对延迟转移指令及其延迟槽设定这种断点条件，就不会在转移地址的第一条指令前发生用户断点的暂停。
4. 如果设定取指令周期，就忽视断点数据寄存器（BDRA 或者 BDRB）。因此，不能给取指令周期的用户断点条件设定断点数据。
5. 如果给取指令周期的用户断点条件设定 I 总线，就不对 I 总线上的取指令周期进行条件判断。详细内容请参照“7.4.1 用户断点运行的流程”的第 5 项。

### 7.4.3 数据存取周期的用户断点

1. 对数据存取的断点，如果将 L 总线指定为断点条件，就对由执行指令存取的地址（和数据）进行条件比较并发生用户断点的暂停；如果将 I 总线指定为断点条件，就对包含 I 总线上 CPU 的全部总线主控所产生的数据存取周期的地址（和数据）进行条件比较并发生用户断点的暂停。有关产生到 I 总线上的 CPU 总线周期，请参照“7.4.1 用户断点运行的流程”的第 5 项。
2. 有关数据存取周期地址和各操作数长度的比较条件关系如表 7.3 所示。

表 7.3 数据存取周期地址和操作数长度的比较条件

存取长度	比较地址
长字	断点地址寄存器的 bit31 ~ 2 和地址总线的 bit31 ~ 2 比较
字	断点地址寄存器的 bit31 ~ 1 和地址总线的 bit31 ~ 1 比较
字节	断点地址寄存器的 bit31 ~ 0 和地址总线的 bit31 ~ 0 比较

例如，当给断点地址寄存器（BARA/BARB）设定地址 H'00001003 时，满足断点条件的总线周期（假设满足其他全部条件）包含以下含义：

H'00001000 中的长字存取  
 H'00001002 中的字存取  
 H'00001003 中的字节存取

3. 当断点条件包含数据值时  
 当断点条件包含数据值时，给断点总线周期寄存器（BBRA/BBRB）指定长字、字或者字节作为操作数长度，并且在地址条件和数据条件一致时发生用户断点的暂停。此时，为了指定字节数据，必须给断点数据寄存器（BDRA/BDRB）和断点数据屏蔽寄存器（BDMRA/BDMRB）的 bit15 ~ 8、bit7 ~ 0 这 2 个字节设定相同的数据。如果选择字或者字节的操作数长度，就忽视 BDRA 或者 BDRB、BDMRA 或者 BDMRB 的 bit31 ~ 16。
4. 当选择 L 总线时，就结束条件一致指令的执行，并在执行下一条指令前发生用户断点的暂停。但是，如果条件中包含数据值，就有可能在条件一致指令的下一条指令执行结束后暂停。当选择 I 总线时，就不能特定要发生用户断点暂停的指令。如果在延迟转移指令或者该延迟槽中发生此类用户断点，就在转移地址的第一条指令前不发生用户断点的暂停。

### 7.4.4 顺序断点

1. 如果将 BRCCR 的 SEQ 位设定为 1，就在通道 A 断点条件一致后并且通道 B 断点条件一致时发生顺序断点。如果在通道 A 断点条件一致前通道 B 断点条件一致，就不发生用户断点的暂停。另外，如果通道 A 和通道 B 的断点条件同时一致，也不发生顺序断点。在指定顺序断点时，如果想在通道 A 条件一致但通道 B 条件不一致时清除通道 A 一致标志，必须给 BRCCR 寄存器的 SEQ 位写 0 并将通道 A 的条件一致标志清 0。
2. 在指定顺序断点时，能选择 L 总线或者 I 总线，也能指定执行次数的断点条件。例如：当指定执行次数的断点条件时，如果在通道 A 断点条件一致后并且通道 B 断点条件 BETR=H'0001 时一致，就满足断点条件。



### 7.4.5 被保存的程序计数器值

在发生用户断点的暂停时，将继续执行的指令地址保存到堆栈，然后进入异常处理状态。如果将 L 总线指定为断点条件，就能决定发生用户断点暂停的指令（断点条件包含数据的情况除外）；如果将 I 总线指定为断点条件，就不能决定发生用户断点暂停的指令。

1. 当将取指令（执行指令前）指定为断点条件时  
将断点条件一致指令的地址保存到堆栈。不执行条件一致的指令，而在执行前发生用户断点的暂停。但是，如果在延迟槽指令发生条件一致，就将延迟转移指令的地址保存到堆栈。
2. 当将取指令（执行指令后）指定为断点条件时  
将断点条件一致指令的下一条指令地址保存到堆栈。执行条件一致的指令，并在执行下一条指令前发生用户断点的暂停。如果在延迟转移指令或者延迟槽发生条件一致，就执行这些指令，并将转移目标地址保存到堆栈。
3. 当将数据存取（只限地址）指定为断点条件时  
将紧接着断点条件一致指令之后的地址保存到堆栈。执行条件一致的指令，并在发生下一条指令前发生用户断点的暂停。但是，如果在延迟槽发生条件一致，就将转移目标地址保存到堆栈。
4. 当将数据存取（地址 + 数据）指定为断点条件时  
如果给断点条件追加数据值，就将断点条件一致指令的下一条指令地址或者下下一条指令地址保存到堆栈。发生用户断点暂停的位置无法正确决定。  
如果在延迟槽指令发生条件一致，就将转移地址保存到堆栈。另外，如果条件一致指令的下一条指令为转移指令，就可能在执行转移指令或者延迟槽指令结束后发生用户断点的暂停。此时，也将转移目标地址保存到堆栈。

### 7.4.6 PC 跟踪

1. 通过将 BRCR 的 PCTE 设定为 1，允许 PC 跟踪。如果发生转移（转移指令和中断异常），就将转移源地址和转移目标地址保存到 BRSR 和 BRDR。
2. 按照转移的种类，被保存到 BRSR 和 BRDR 的值分别如下：
  - 当因转移指令而发生转移时，转移指令的地址保存到 BRSR、转移目标指令的地址保存到 BRDR。
  - 当因中断或者一般异常而发生转移时，因发生异常而被保存的堆栈值保存到 BRSR、异常处理程序的起始地址保存到 BRDR。
3. BRSR 和 BRDR 由 4 组（支持 E10A 全功能的 F-ZTAT 版是 8 组）队列结构构成。当读被保存在 PC 跟踪寄存器的地址时，先读队列的第一项。BRSR 和 BRDR 共享读指针，必须按照 BRSR、BRDR 的顺序读取，队列只在读 BRDR 后进行移位。如果将 BRCR 的 PCTE 位从 OFF 切换到 ON，队列的值就无效。
4. 4 组（支持 E10A 全功能的 F-ZTAT 版是 8 组）队列和 AUD 共享。必须在将 STBCR5 的 MSTP25 位设定为 0 并且将 STBCR6 的 AUDSRST 位设定为 1 后，将 BRCR 的 PCTE 位设定为 1。AUD 功能只限支持 E10A 全功能的 F-ZTAT 版，但是对于通常的 F-ZTAT 版，也需要进行同样的设定。

### 7.4.7 使用例

#### (1) 指定为 L 总线取指令周期的断点条件

(例 1-1)

- 寄存器指定

BARA=H'00000404、BAMRA=H'00000000、BBRA=H'0054、BDRA=H'00000000、BDMRA=H'00000000、  
BARB=H'00008010、BAMRB=H'00000006、BBRB=H'0054、BDRB=H'00000000、BDMRB=H'00000000、  
BRCCR=H'00000400

指定条件：通道 A/ 通道 B 独立模式

<通道 A >

地址：H'00000404、地址屏蔽：H'00000000

数据：H'00000000、数据屏蔽：H'00000000

总线周期：L 总线 / 取指令（执行指令后） / 读（断点条件不包含操作数长度）

<通道 B >

地址：H'00008010、地址屏蔽：H'00000006

数据：H'00000000、数据屏蔽：H'00000000

总线周期：L 总线 / 指令取（执行指令前） / 读（断点条件不包含操作数长度）

在执行地址 H'00000404 的指令后或者在执行地址 H'00008010 ~ H'00008016 的指令前发生用户断点的暂停。

(例 1-2)

- 寄存器指定

BARA=H'00037226、BAMRA=H'00000000、BBRA=H'0056、BDRA=H'00000000、BDMRA=H'00000000、  
BARB=H'0003722E、BAMRB=H'00000000、BBRB=H'0056、BDRB=H'00000000、BDMRB=H'00000000、  
BRCCR=H'00000008

指定条件：通道 A/ 通道 B 顺序模式

<通道 A >

地址：H'00037226、地址屏蔽：H'00000000

数据：H'00000000、数据屏蔽：H'00000000

总线周期：L 总线 / 取指令（执行指令前） / 读 / 字

<通道 B >

地址：H'0003722E、地址屏蔽：H'00000000

数据：H'00000000、数据屏蔽：H'00000000

总线周期：L 总线 / 取指令（执行指令前） / 读 / 字

在执行地址 H'00037226 的指令后并在执行地址 H'0003722E 的指令前发生用户断点的暂停。

(例 1-3)

## • 寄存器指定

BARA=H'00027128、BAMRA=H'00000000、BBRA=H'005A、BDRA=H'00000000、BDMRA=H'00000000、  
 BARB=H'00031415、BAMRB=H'00000000、BBRB=H'0054、BDRB=H'00000000、BDMRB=H'00000000、  
 BRRCR=H'00000000

指定条件：通道 A/ 通道 B 独立模式

<通道 A >

地址：H'00027128、地址屏蔽：H'00000000

数据：H'00000000、数据屏蔽：H'00000000

总线周期：L 总线 / 取指令（执行指令前） / 写 / 字

<通道 B >

地址：H'00031415、地址屏蔽：H'00000000

数据：H'00000000、数据屏蔽：H'00000000

总线周期：L 总线 / 取指令（执行指令前） / 读（断点条件不包含操作数长度）

在通道 A 中，因为取指令不是写周期，所以不发生用户断点的暂停；在通道 B 中，因为不对偶数地址取指令，所以也不发生用户断点的暂停。

(例 1-4)

## • 寄存器指定

BARA=H'00037226、BAMRA=H'00000000、BBRA=H'005A、BDRA=H'00000000、BDMRA=H'00000000、  
 BARB=H'0003722E、BAMRB=H'00000000、BBRB=H'0056、BDRB=H'00000000、BDMRB=H'00000000、  
 BRRCR=H'00000008

指定条件：通道 A/ 通道 B 顺序模式

<通道 A >

地址：H'00037226、地址屏蔽：H'00000000

数据：H'00000000、数据屏蔽：H'00000000

总线周期：L 总线 / 取指令（执行指令前） / 写 / 字

<通道 B >

地址：H'0003722E、地址屏蔽：H'00000000

数据：H'00000000、数据屏蔽：H'00000000

总线周期：L 总线 / 取指令（执行指令前） / 写 / 字

在通道 A 中，因为取指令不是写周期，所以不发生顺序条件一致，也不发生用户断点的暂停。

## (例 1-5)

## • 寄存器指定

BARA=H'00000500、BAMRA=H'00000000、BBRA=H'0057、BDRA=H'00000000、BDMRA=H'00000000、  
 BARB=H'00001000、BAMRB=H'00000000、BBRB=H'0057、BDRB=H'00000000、BDMRB=H'00000000、  
 BRRCR=H'00000001、BETR=H'0005

指定条件：通道 A/ 通道 B 独立模式

<通道 A >

地址：H'00000500、地址屏蔽：H'00000000

数据：H'00000000、数据屏蔽：H'00000000

总线周期：L 总线 / 取指令（执行指令前） / 读 / 长字

允许执行次数中断（5 次）

<通道 B >

地址：H'00001000、地址屏蔽：H'00000000

数据：H'00000000、数据屏蔽：H'00000000

总线周期：L 总线 / 取指令（执行指令前） / 读 / 长字

在通道 A 中，在执行 4 次地址 H'00000500 的指令后并在执行第 5 次指令前发生用户断点的暂停；在通道 B 中，在执行地址 H'00001000 的指令前发生用户断点的暂停。

## (例 1-6)

## • 寄存器指定

BARA=H'00008404、BAMRA=H'00000FFF、BBRA=H'0054、BDRA=H'00000000、BDMRA=H'00000000、  
 BARB=H'00008010、BAMRB=H'00000006、BBRB=H'0054、BDRB=H'00000000、BDMRB=H'00000000、  
 BRRCR=H'00000400

指定条件：通道 A/ 通道 B 独立模式

<通道 A >

地址：H'00008404、地址屏蔽：H'00000FFF

数据：H'00000000、数据屏蔽：H'00000000

总线周期：L 总线 / 指令取（执行指令后） / 读（断点条件不包含操作数长度）

<通道 B >

地址：H'00008010、地址屏蔽：H'00000006

数据：H'00000000、数据屏蔽：H'00000000

总线周期：L 总线 / 取指令（执行指令前） / 读（断点条件不包含操作数长度）

在执行地址 H'00008000 ~ H'00008FFE 的指令后或者在执行地址 H'00008010 ~ H'00008016 的指令前发生用户断点的暂停。

## (2) 指定为 L 总线数据存取周期的断点条件

(例 2-1)

## • 寄存器指定

BARA=H'00123456、BAMRA=H'00000000、BBRA=H'0064、BDRA=H'12345678、BDMRA=H'FFFFFFFF、  
BARB=H'000ABCDE、BAMRB=H'000000FF、BBRB=H'006A、BDRB=H'0000A512、BDMRB=H'00000000、  
BRRCR=H'00000080

指定条件：通道 A/ 通道 B 独立模式

<通道 A >

地址：H'00123456、地址屏蔽：H'00000000

数据：H'12345678 数据屏蔽：H'FFFFFFFF

总线周期：L 总线 / 数据存取 / 读 (断点条件不包含操作数长度)

<通道 B >

地址：H'000ABCDE、地址屏蔽：H'000000FF

数据：H'0000A512、数据屏蔽：H'00000000

总线周期：L 总线 / 数据存取 / 写 / 字

在通道 A 中，在读地址 H'00123454 的长字、或者地址 H'00123456 的字、或者地址 H'00123456 的字节时发生用户断点的暂停；在通道 B 中，在给 H'000ABC00 ~ H'000ABCFE 写字数据 H'A512 时发生用户断点的暂停。

## (3) 给 I 总线数据存取周期指定的断点条件

(例 3-1)

## • 寄存器指定

BARA=H'00314154、BAMRA=H'00000000、BBRA=H'0194、BDRA=H'12345678、BDMRA=H'FFFFFFFF、  
BARB=H'00055555、BAMRB=H'00000000、BBRB=H'01A9、BDRB=H'00007878、BDMRB=H'0000F0F0、  
BRRCR=H'00000080

指定条件：通道 A/ 通道 B 独立模式

<通道 A >

地址：H'00314154、地址屏蔽：H'00000000

数据：H'12345678、数据屏蔽：H'FFFFFFFF

总线周期：I 总线 (CPU 周期) / 指令取 / 读 (断点条件不包含操作数长度)

<通道 B >

地址：H'00055555、地址屏蔽：H'00000000

数据：H'00000078、数据屏蔽：H'0000000F

总线周期：I 总线 (CPU 周期) / 数据存取 / 写 / 字节

在通道 A 中，在外部存储空间的地址 H'00314156 取指令时发生用户断点的暂停；在通道 B 中，在 CPU 将字节数据 H'7x 写到外部存储空间的地址 H'00055555 时发生用户断点的暂停。

## 7.5 使用注意事项

1. 通过 I 总线进行 UBC 寄存器的读写。因此，在执行改写 UBC 寄存器的指令后到实际反映该值的期间，有可能不发生想要的用户断点。为得知更改 UBC 寄存器的时序，必须在最后读已写的寄存器。此后的指令对已写的寄存器值有效。
2. UBC 不能用相同的通道监视 L 总线周期和 I 总线周期。
3. 有关顺序中断指定的注意事项如下：  
如果设定顺序中断，就在发生 A 通道一致后并在发生 B 通道一致时，发生条件一致。因此，即使设定了通道 A 一致和通道 B 一致同时发生的总线周期，也不进行用户断点的暂停。
4. 如果在同一条指令中发生用户断点和其他异常，就按照“第 5 章 异常处理”的表 5.1 中规定的优先级进行判断。如果发生更高优先级的异常，就不发生用户断点的暂停。
  - 比其他任何异常优先接受指令执行前的断点。
  - 如果指令执行后的断点或者数据存取的断点与更高优先级的再执行型异常（包含指令执行前的断点）同时发生，就接受再执行型异常处理，并且不将条件一致标志置位（但是，有第 5 项的例外事项）。通过异常处理解除再执行型异常源，在结束重新执行的同一指令时，再次进行用户断点的暂停，并将标志置位。
  - 如果指令执行后的断点或者数据存取的断点与更高优先级的完成型异常（TRAPA）同时发生，就不进行用户断点的暂停，但将条件一致标志置位。
5. 作为第 4 项的例外，有以下注意事项：  
对于因数据存取而发生 CPU 地址错误的指令，如果指令执行后的断点或者数据存取的断点成立，就优先用户断点中断而发生 CPU 地址错误，并将 UBC 的条件一致标志置位。
6. 在延迟槽中进行用户断点的暂停时，有以下注意事项：  
如果给 RTE 指令的延迟槽指令设定指令执行前的断点，就在执行 RTE 指令的转移目标前不进行用户断点的暂停。
7. 在 UBC 模块待机时，不能使用用户断点功能，并且不能读写 UBC 寄存器，否则，就不保证其值。
8. 对 SLEEP 指令和延迟槽为 SLEEP 指令的转移指令，不能设定指令执行后的断点。另外，对 SLEEP 指令和 SLEEP 指令的前 1 ~ 2 条指令，不能设定数据存取的断点。
9. 当 UBC、DTC 或者 DMAC 在运行中时，在 I 总线上不能正确判断通过 CPU 存取的外部空间。如果在上述条件下在 I 总线上判断存取外部空间时，必须选择全总线主控。这时就无法用特定的条件判断总线主控。但是，从数据的值推定总线主控时，通过包含在判定条件的数据就可以推定总线主控。

## 第 8 章 数据传送控制器 (DTC)

本 LSI 内置数据传送控制器 (DTC)，能通过中断请求启动 DTC 进行数据传送。

### 8.1 特点

- 能进行任意通道数的传送
- 能进行链传送 (对于一个启动源进行多个数据的传送)  
只能在指定次数的数据传送后进行链传送 (当计数器 =0 时)
- 传送模式: 3 种  
可选择正常传送模式、重复传送模式或者块传送模式  
可选择传送源地址和传送目标地址的增量、减量或者固定
- 能用 32 位指定传送源和传送目标的地址, 并且可直接指定 4G 字节的地址空间
- 能将传送数据的数据长度设定为字节、字或者长字
- 能向 CPU 请求启动 DTC 的中断  
能在一数据传送结束后向 CPU 请求中断  
能在指定的数据传送结束后向 CPU 请求中断
- 能指定传送信息的跳读
- 对选择固定的传送源地址或者传送目标地址, 执行回写省略功能
- 能设定模块停止模式
- 能设定短地址模式
- 可从 5 种时序中选择总线权的释放时序
- 可从 2 种优先级中选择 DTC 启动时的优先级

DTC 的框图如图 8.1 所示, 可将 DTC 传送信息分配到数据区\*。

【注】 \* 如果将传送信息分配到内部 RAM, 就必须将 RAMCR 的 RAME 位设定为 1。

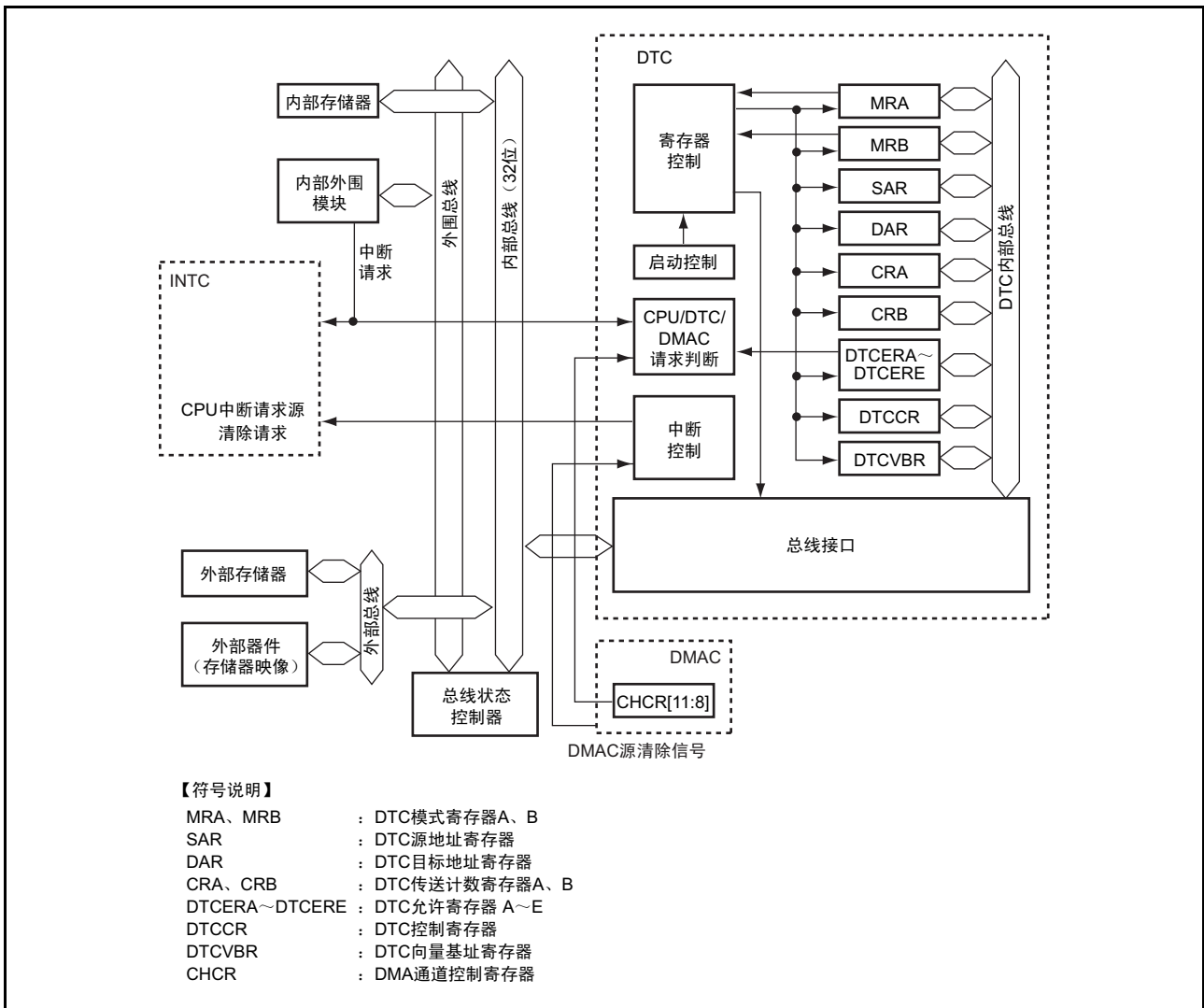


图 8.1 DTC 的框图



## 8.2 寄存器说明

DTC 有以下寄存器。有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 27 章 寄存器一览”。

CPU 不能直接存取 MRA、MRB、SAR、DAR、CRA 和 CRB 寄存器，这些寄存器的内容作为传送信息被分配到数据区。如果发生 DTC 启动源，就根据各启动源决定的向量地址读取传送信息的起始地址，将任意的传送信息传送到 DTC 内，进行数据传送。当数据传送结束时，回写这些寄存器的内容。

但是，CPU 能直接存取 DTCERA ~ DTCERE、DTCCR 和 DTCVBR。

表 8.1 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
DTC 允许寄存器 A	DTCERA	R/W	H'0000	H'FFFCC80	8、16
DTC 允许寄存器 B	DTCERB	R/W	H'0000	H'FFFCC82	8、16
DTC 允许寄存器 C	DTCERC	R/W	H'0000	H'FFFCC84	8、16
DTC 允许寄存器 D	DTCERD	R/W	H'0000	H'FFFCC86	8、16
DTC 允许寄存器 E	DTCERE	R/W	H'0000	H'FFFCC88	8、16
DTC 控制寄存器	DTCCR	R/W	H'00	H'FFFCC90	8
DTC 向量基址寄存器	DTCVBR	R/W	H'00000000	H'FFFCC94	8、16、32
总线功能扩展寄存器	BSCEHR	R/W	H'0000	H'FFF89A	8、16

## 8.2.1 DTC 模式寄存器 A (MRA)

MRA 选择 DTC 的运行模式。CPU 不能直接存取 MRA。

位:	7	6	5	4	3	2	1	0
	MD[1:0]		Sz[1:0]		SM[1:0]		-	-
初始值:	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	-	-	-	-	-	-	-	-

位	位名	初始值	R/W	说 明
7、6	MD[1:0]	不定	—	DTC 模式 1、0 指定 DTC 的传送模式。 00: 正常传送模式 01: 重复传送模式 10: 块传送模式 11: 禁止设定
5、4	Sz[1:0]	不定	—	DTC 数据的传送长度 1、0 指定传送数据的长度。 00: 字节传送 01: 字传送 10: 长字传送 11: 禁止设定
3、2	SM[1:0]	不定	—	源地址模式 1、0 指定数据传送后的 SAR 运行。 0x: SAR 固定 (不进行 SAR 的回写。) 10: 传送后 SAR 增量 (Sz1、Sz0 为 B'00 时 SAR+1, 为 B'01 时 SAR+2, 为 B'10 时 SAR+4) 11: 传送后 SAR 减量 (Sz1、Sz0 为 B'00 时 SAR-1, 为 B'01 时 SAR-2, 为 B'10 时 SAR-4)
1、0	—	不定	—	保留位 写的值总是 0。

【注】 x: Don't care

## 8.2.2 DTC 模式寄存器 B (MRB)

MRB 选择 DTC 的运行模式。CPU 不能直接存取 MRB。

位:	7	6	5	4	3	2	1	0
	CHNE	CHNS	DISEL	DTS	DM[1:0]	-	-	-
初始值:	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	-	-	-	-	-	-	-	-

位	位名	初始值	R/W	说 明
7	CHNE	不定	—	DTC 链传送的允许 指定链传送。链传送的详细内容请参照“8.5.6 链传送”。由 CHNS 位选择链传送的条件。 0: 禁止链传送 1: 允许链传送
6	CHNS	不定	—	DTC 链传送的选择 选择链传送的条件。如果下一次传送是链传送，就不进行指定传送次数的结束判断，并且不清除启动源标志或者 DTCER。 0: 连续进行链传送 1: 只在传送计数器 =0 时进行链传送
5	DISEL	不定	—	DTC 中断的选择 当此位是 1 时，每进行 1 次数据传送或者 1 次块数据传送，就向 CPU 请求中断。当此位是 0 时，只在指定次数的数据传送结束时，向 CPU 请求中断。 【注】如果将 I <sup>2</sup> C2 作为启动源，就必须将此位设定为 0。
4	DTS	不定	—	DTC 传送模式的选择 在重复传送模式或者块传送模式中，指定源或者目标为重复区或者块区域。 0: 目标为重复区或者块区域 1: 源为重复区域或者块区域
3、2	DM[1:0]	不定	—	目标地址模式 1、0 指定数据传送后的 DAR 运行。 0x: DAR 固定 (不进行 DAR 的回写) 10: 传送后 DAR 增量 (Sz1、Sz0 为 B'00 时 DAR+1, 为 B'01 时 DAR+2, 为 B'10 时 DAR+4) 11: 传送后 DAR 减量 (Sz1、Sz0 为 B'00 时 DAR-1, 为 B'01 时 DAR-2, 为 B'10 时 DAR-4)
1、0	—	不定	—	保留位 写的值总是 0。

【注】 x: Don't care

### 8.2.3 DTC 源地址寄存器 (SAR)

SAR 是 32 位寄存器，指定 DTC 传送数据的传送源地址。

CPU 不能直接存取 SAR。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

### 8.2.4 DTC 目标地址寄存器 (DAR)

DAR 是 32 位寄存器，指定 DTC 传送数据的传送目标地址。

CPU 不能直接存取 DAR。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

### 8.2.5 DTC 传送计数寄存器 A (CRA)

CRA 是 16 位寄存器，指定 DTC 传送数据的传送次数。

在正常传送模式中，CRA 用作 16 位传送计数器（1 ~ 65536）。每进行 1 次数据传送，CRA 减 1，当计数器值为 H'0000 时，就在清除与启动源对应的 DTCE<sub>n</sub> 位（n=15 ~ 0）后向 CPU 请求中断。当设定值为 H'0001 时，传送次数为 1 次；当设定值为 H'FFFF 时，传送次数为 65535 次；当设定值为 H'0000 时，传送次数为 65536 次。

在重复传送模式中，CRA 被分为高 8 位的 CRAH 和低 8 位的 CRAL。CRAH 保存传送次数，CRAL 用作 8 位传送计数器（1 ~ 256）。每进行 1 次数据传送，CRAL 减 1，当计数器值为 H'00 时，就将 CRAH 的内容传送到 CRAL。当设定值 CRAH=CRAL=H'01 时，传送次数为 1 次；当设定值为 H'FF 时，传送次数为 255 次；当设定值为 H'00 时，传送次数为 256 次。

在块传送模式中，CRA 也被分为高 8 位的 CRAH 和低 8 位的 CRAL。CRAH 保存块大小，CRAL 用作 8 位块大小的计数器（1 ~ 256 个字节、1 ~ 256 个字或者 1 ~ 256 个长字）。每进行 1 个字节（1 个字或者 1 个长字）的数据传送，CRAL 减 1，当计数器值为 H'00 时，就将 CRAH 的内容传送到 CRAL。当设定值 CRAH=CRAL=H'01 时，块为 1 个字节（1 个字或者 1 个长字）；当设定值为 H'FF 时，块为 255 个字节（255 字或者 255 个长字）；当设定值为 H'00 时，块为 256 个字节（256 个字或者 256 个长字）。

CPU 不能直接存取 CRA。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

### 8.2.6 DTC 传送计数寄存器 B (CRB)

CRB 是 16 位寄存器，在块传送模式中，指定 DTC 块数据传送的传送次数。CRB 用作 16 位传送次数的计数器（1 ~ 65536），每进行 1 次块数据传送，CRB 减 1，当计数器值为 H'0000 时，在清除与启动源对应的 DTCE<sub>n</sub> 位（n=15 ~ 0）后向 CPU 请求中断。当设定值为 H'0001 时，传送次数为 1 次；当设定值为 H'FFFF 时，传送次数为 65535 次；当设定值为 H'0000 时，传送次数为 65536 次。

在正常传送模式和重复传送模式中不使用 CRB。CPU 不能直接存取 CRB。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

### 8.2.7 DTC 允许寄存器 A ~ E (DTCERA ~ DTCERE)

DTCE<sub>R</sub> 是选择启动 DTC 的中断源寄存器，有 DTCERA ~ DTCERE。有关各中断源和 DTCE 位的对应请参照表 8.2。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTCE15	DTCE14	DTCE13	DTCE12	DTCE11	DTCE10	DTCE9	DTCE8	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0	
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	DTCE15	0	R/W	DTC 启动的允许 15 ~ 0
14	DTCE14	0	R/W	如果将相应位设定为 1，就选择对应的中断源作为 DTC 启动源。
13	DTCE13	0	R/W	[清除条件]
12	DTCE12	0	R/W	• 当读到 1 状态（要清除的位）后写 0 时
11	DTCE11	0	R/W	• 当 MRB 的 DISEL 位是 1 并且 1 次数据传送结束时
10	DTCE10	0	R/W	• 当指定次数的传送结束时
9	DTCE9	0	R/W	在 DISEL 位是 0 并且指定次数的传送没有结束时，不清除此位。
8	DTCE8	0	R/W	[置位条件]
7	DTCE7	0	R/W	• 当读到 0 的状态（要被置位的位）后写 1 时
6	DTCE6	0	R/W	
5	DTCE5	0	R/W	
4	DTCE4	0	R/W	
3	DTCE3	0	R/W	
2	DTCE2	0	R/W	
1	DTCE1	0	R/W	
0	DTCE0	0	R/W	

## 8.2.8 DTC 控制寄存器 (DTCCR)

DTCCR 设定传送信息的跳读。

位:	7	6	5	4	3	2	1	0
	-	-	-	RRS	RCHNE	-	-	ERR
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R	R	R/(W)*

【注】\* 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说 明
7 ~ 5	—	全 0	R	保留位 读写的值总是 0。
4	RRS	0	R/W	允许 DTC 传送信息的跳读 控制向量地址和传送信息的读操作。DTC 向量号随时和上次启动的向量号进行比较，如果向量号的值一致并且此位是 1，就不读向量地址和传送信息而进行 DTC 的数据传送。如果上次的启动是链传送，就必须读取向量地址和传送信息。 但是，如果将总线功能扩展寄存器 (BSCEHR) 的 DTPR 位设定为 1，就与本位的设定无关，不跳读传送信息。 0: 不跳读传送信息 1: 当向量号的值一致时跳读传送信息
3	RCHNE	0	R/W	允许 DTC 重复传送后的链传送 在进行重复传送时，允许或者禁止链传送（当传送计数器 =0 时）。如果在进行重复传送时传送计数器 (CRAL) 为 0，因为 CRAH 指定的值被回写到 CRAL，所以在传送计数器 =0 时不发生链传送。通过将此位设定为 1，在回写传送计数器时允许链传送。 0: 禁止重复传送后的链传送 1: 允许重复传送后的链传送
2、1	—	全 0	R	保留位 能读但不能写这些位。
0	ERR	0	R/(W)*	传送停止标志 表示发生 DTC 地址错误或者 NMI 中断请求。如果在启动 DTC 时发生 DTC 地址错误或者 NMI 中断请求，就在释放 DTC 的总线权后执行 DTC 地址错误或者 NMI 中断的处理。在数据传送后，DTC 以传送信息的写状态停止。 0: 没有中断请求 1: 发生中断请求 [清除条件] • 当读到 1 的状态后写 0 时

### 8.2.9 DTC 向量基址寄存器 (DTCVBR)

DTCVBR 是 32 位寄存器，设定用于计算向量表地址的基址。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
31 ~ 12		全 0	R/W	对于 bit11 ~ 0, 读写的值总是 0。
11 ~ 0	—	全 0	R	

### 8.2.10 总线功能扩展寄存器 (BSCEHR)

BSCEHR 是 16 位寄存器，设定 DTC 总线权的释放时序等。对于优先进行 DTC 传送以及减少 DTC 启动周期数，BSCEHR 能设定有效的功能。详细内容请参照“9.4.8 总线功能扩展寄存器 (BSCEHR)”。

## 8.3 启动源

通过中断请求启动 DTC。由 DTCER 选择要启动 DTC 的中断源，当 DTCER 的对应位被置 1 时，对应的中断源就为 DTC 的启动源；当对应位被清零时，对应的中断源就为 CPU 的中断源。在 1 次数据传送（在链传送时，连续的最后传送）结束时，清除启动源中断标志或者 DTCER 的对应位。

### 8.4 传送信息的分配和 DTC 向量表

将传送信息分配到数据区。传送信息的起始地址必须为地址  $4n$ ，否则就忽视低 2 位进行存取 ([1:0]=B'00)。数据区中的传送信息的分配如图 8.2 所示。只有在全部分 DTC 传送的传送源 / 传送目标都为内部 RAM 和内部外围模块时，才能通过将“9.4.8 总线功能扩展寄存器 (BSCEHR)”中的 DTSA 位设定为 1 来选择短地址模式。

通常，传送信息需要 4 个长字，但是通过选择短地址模式，能将传送信息的长度减少为 3 个长字，缩短 DTC 的启动时间。

DTC 按照启动源从向量表读取传送信息的起始地址，然后从该起始地址开始读传送信息。DTC 向量表和传送信息的对应如图 8.3 所示。

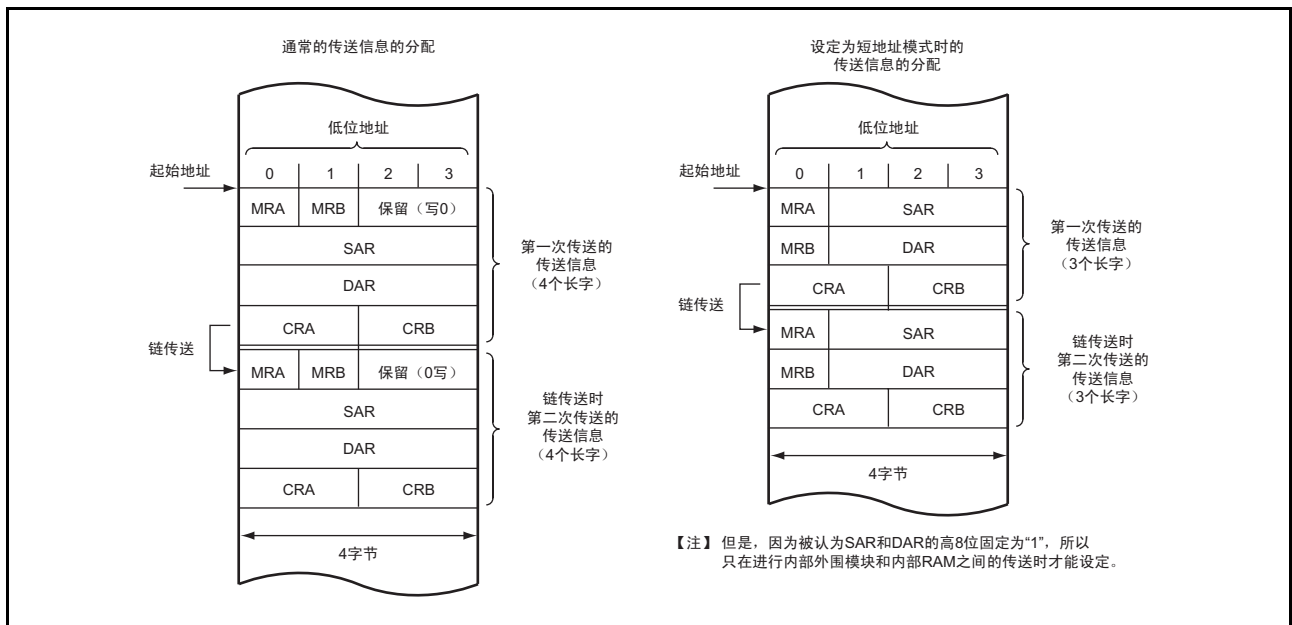


图 8.2 数据区中的传送信息的分配

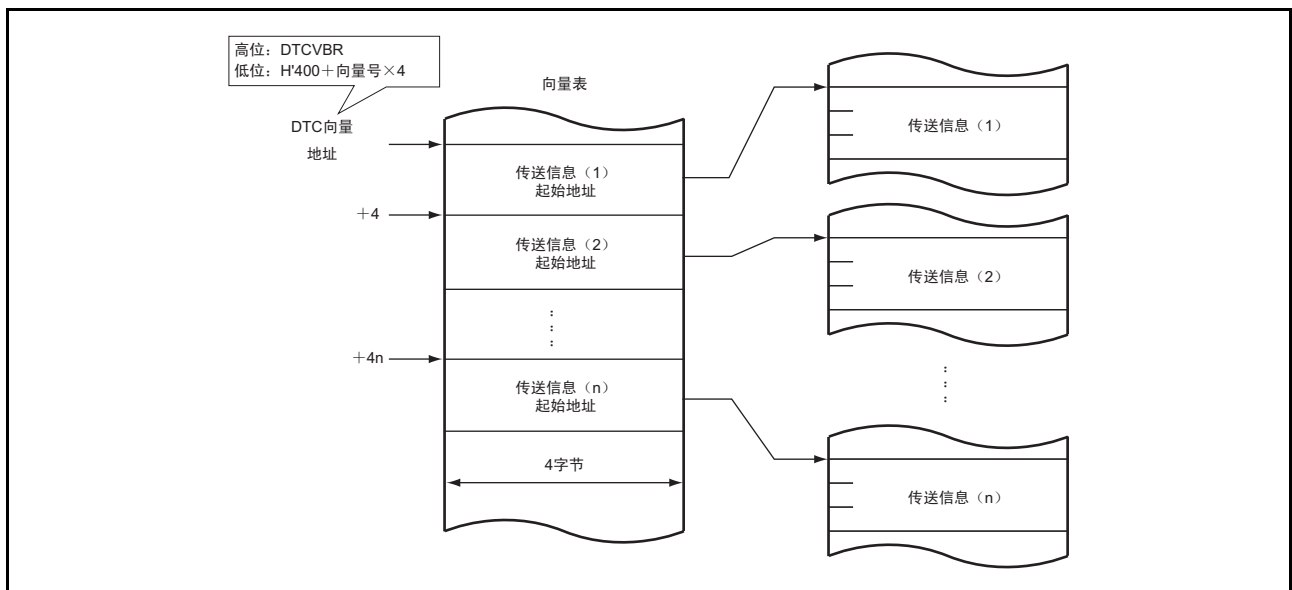


图 8.3 DTC 向量表和传送信息的对应



DTC 的启动源和向量地址的对应如表 8.2 所示。

表 8.2 中断源和 DTC 向量地址以及对应的 DTCE

启动源的发生源	启动源	向量号	DTC 向量地址的偏移量	DTCE*1	传送源	传送目标	优先级
外部引脚	IRQ0	64	H'500	DTCERA15	任意 *2	任意 *2	高 ↑
	IRQ1	65	H'504	DTCERA14	任意 *2	任意 *2	
	IRQ2	66	H'508	DTCERA13	任意 *2	任意 *2	
	IRQ3	67	H'50C	DTCERA12	任意 *2	任意 *2	
	IRQ4	68	H'510	DTCERA11	任意 *2	任意 *2	
	IRQ5	69	H'514	DTCERA10	任意 *2	任意 *2	
	IRQ6	70	H'518	DTCERA9	任意 *2	任意 *2	
	IRQ7	71	H'51C	DTCERA8	任意 *2	任意 *2	
MTU2_0	TGIA_0	88	H'560	DTCERB15	任意 *2	任意 *2	
	TGIB_0	89	H'564	DTCERB14	任意 *2	任意 *2	
	TGIC_0	90	H'568	DTCERB13	任意 *2	任意 *2	
	TGID_0	91	H'56C	DTCERB12	任意 *2	任意 *2	
MTU2_1	TGIA_1	96	H'580	DTCERB11	任意 *2	任意 *2	
	TGIB_1	97	H'584	DTCERB10	任意 *2	任意 *2	
MTU2_2	TGIA_2	104	H'5A0	DTCERB9	任意 *2	任意 *2	
	TGIB_2	105	H'5A4	DTCERB8	任意 *2	任意 *2	
MTU2_3	TGIA_3	112	H'5C0	DTCERB7	任意 *2	任意 *2	
	TGIB_3	113	H'5C4	DTCERB6	任意 *2	任意 *2	
	TGIC_3	114	H'5C8	DTCERB5	任意 *2	任意 *2	
	TGID_3	115	H'5CC	DTCERB4	任意 *2	任意 *2	
MTU2_4	TGIA_4	120	H'5E0	DTCERB3	任意 *2	任意 *2	
	TGIB_4	121	H'5E4	DTCERB2	任意 *2	任意 *2	
	TGIC_4	122	H'5E8	DTCERB1	任意 *2	任意 *2	
	TGID_4	123	H'5EC	DTCERB0	任意 *2	任意 *2	
	TCIV_4	124	H'5F0	DTCERC15	任意 *2	任意 *2	
MTU2_5	TGIU_5	128	H'600	DTCERC14	任意 *2	任意 *2	
	TGIV_5	129	H'604	DTCERC13	任意 *2	任意 *2	
	TGIW_5	130	H'608	DTCERC12	任意 *2	任意 *2	
MTU2S_3	TGIA_3S	160	H'680	DTCERC3	任意 *2	任意 *2	低 ↓
	TGIB_3S	161	H'684	DTCERC2	任意 *2	任意 *2	
	TGIC_3S	162	H'688	DTCERC1	任意 *2	任意 *2	
	TGID_3S	163	H'68C	DTCERC0	任意 *2	任意 *2	



## 8.5 运行说明

传送模式有正常传送模式、重复传送模式和块传送模式。能通过将传送信息保存到数据区，进行任意通道数的数据传送。当启动 DTC 时，在从数据区读取传送信息后进行数据传送，并且在数据传送后回写传送信息。

DTC 用 SAR 指定传送源地址，用 DAR 指定传送目标地址。在传送后，SAR 和 DAR 分别被增量、减量或者固定。

DTC 的传送模式如表 8.3 所示。

表 8.3 DTC 的传送模式

传送模式	1 次传送请求 能传送的数据长度	存储器地址的增减	传送次数
正常传送模式	1 个字节 / 字 / 长字	增减 1、2、4 或者 · 固定	1 ~ 65536 次
重复传送模式 *1	1 个字节 / 字 / 长字	增减 1、2、4 或者 · 固定	1 ~ 256 次 *3
块传送模式 *2	CRAH 指定的块大小 (1 ~ 256 个字节 / 字 / 长字)	增减 1、2、4 或者 · 固定	1 ~ 65536 次 *4

【注】 \*1 将源或者目标设定为重复区域

\*2 将源或者目标设定为块区域

\*3 在进行指定次数的传送后，恢复初始状态继续运行

\*4 1 次表示 1 块

另外，能通过预先将 MRB 的 CHNE 位设定为 1，用一个启动源进行多次传送（链传送），也能通过设定 MRB 的 CHNS 位，在传送计数器 =0 时进行链传送的设定。

DTC 的运行流程图如图 8.4 所示，DTC 的传送条件（包括链传送）如表 8.4 所示（省略了从第 2 次传送到第 3 次传送的组合）。

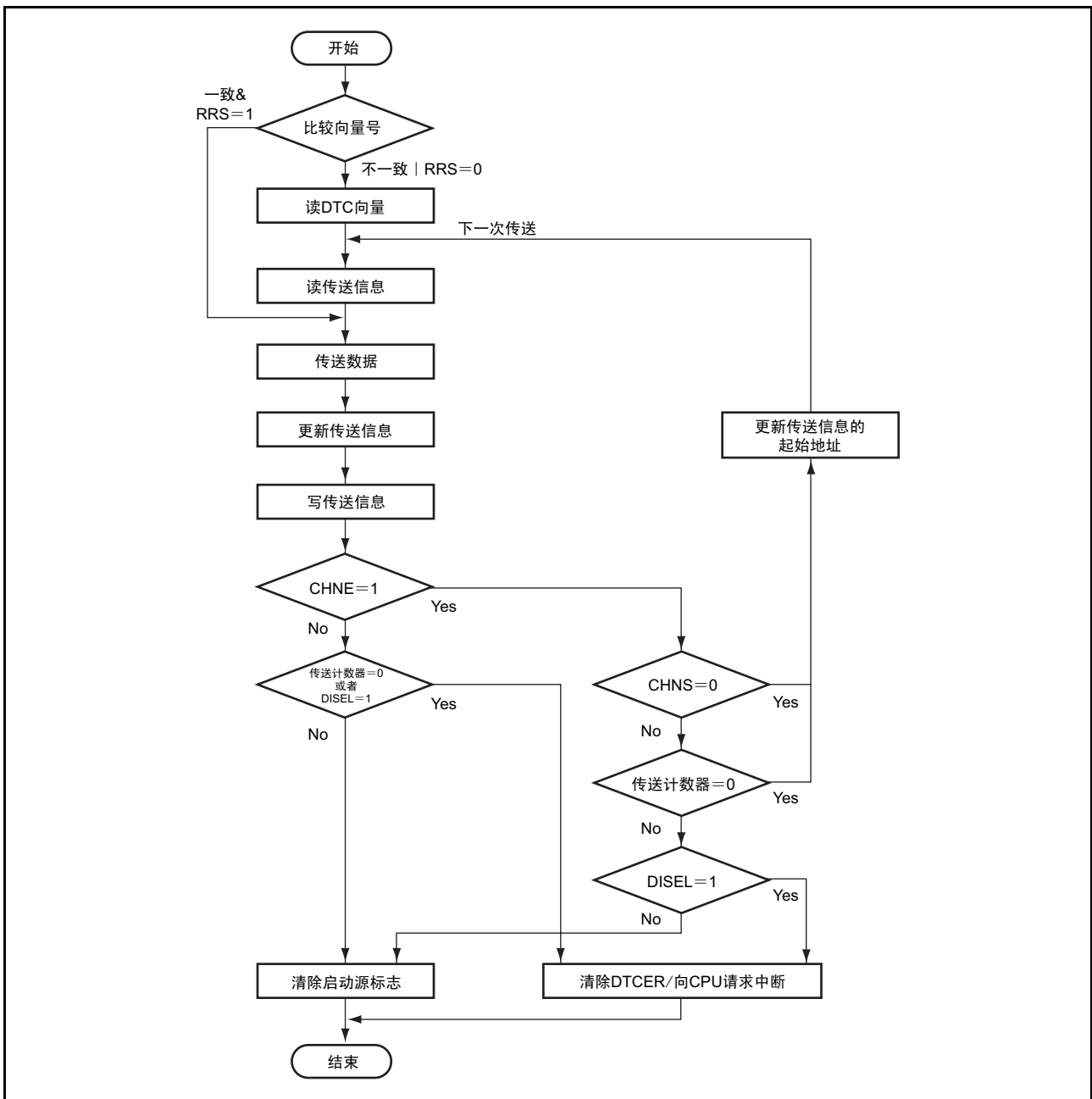


图 8.4 DTC 的运行流程图

表 8.4 DTC 的传送条件 (包括链传送)

传送模式	第 1 次传送					第 2 次传送					DTC 传送	
	CHNE	CHNS	RCHNE	DISEL	传送计数器 *1	CHNE	CHNS	RCHNE	DISEL	传送计数器 *1		
正常	0	—	—	0	非 0	—	—	—	—	—	在第 1 次传送结束	
	0	—	—	0	0	—	—	—	—	—	在第 1 次传送结束 向 CPU 请求中断	
	0	—	—	1	—	—	—	—	—	—		
	1	0	—	—	—	—	0	—	—	0	非 0	在第 2 次传送结束
						—	0	—	—	0	0	在第 2 次传送结束 向 CPU 请求中断
						—	0	—	—	1	—	
	1	1	—	0	非 0	—	—	—	—	—	在第 1 次传送结束	
	1	1	—	1	非 0	—	—	—	—	—	在第 1 次传送结束 向 CPU 请求中断	
	1	1	—	—	0	—	0	—	—	0	非 0	在第 2 次传送结束
						—	0	—	—	0	0	在第 2 次传送结束 向 CPU 请求中断
—						0	—	—	1	—		
重复	0	—	—	0	—	—	—	—	—	—	在第 1 次传送结束	
	0	—	—	1	—	—	—	—	—	—	在第 1 次传送结束 向 CPU 请求中断	
	1	0	—	—	—	—	0	—	—	0	—	在第 2 次传送结束
						—	0	—	—	1	—	在第 2 次传送结束 向 CPU 请求中断
	1	1	—	0	非 0	—	—	—	—	—	在第 1 次传送结束	
	1	1	—	1	非 0	—	—	—	—	—	在第 1 次传送结束 向 CPU 请求中断	
	1	1	0	0	0*2	—	—	—	—	—	在第 1 次传送结束	
	1	1	0	1	0*2	—	—	—	—	—	在第 1 次传送结束 向 CPU 请求中断	
	1	1	1	—	0*2	—	0	—	—	0	—	在第 2 次传送结束
						—	0	—	—	1	—	在第 2 次传送结束 向 CPU 请求中断

传送模式	第 1 次传送					第 2 次传送					DTC 传送
	CHNE	CHNS	RCHNE	DISEL	传送计数器 *1	CHNE	CHNS	RCHNE	DISEL	传送计数器 *1	
块	0	—	—	0	非 0	—	—	—	—	—	在第 1 次传送结束
	0	—	—	0	0	—	—	—	—	—	在第 1 次传送结束 向 CPU 请求中断
	0	—	—	1	—	—	—	—	—	—	
	1	0	—	—	—	0	—	—	0	非 0	在第 2 次传送结束
						0	—	—	0	0	在第 2 次传送结束 向 CPU 请求中断
	0	—	—	1	—	—	—	—	—	—	在第 1 次传送结束 向 CPU 请求中断
	1	1	—	0	—	—	—	—	—	—	在第 1 次传送结束 向 CPU 请求中断
	1	1	—	1	非 0	—	—	—	—	—	在第 1 次传送结束 向 CPU 请求中断
	1	1	—	1	0	0	—	—	0	非 0	在第 2 次传送结束
0						—	—	0	0	在第 2 次传送结束 向 CPU 请求中断	
0						—	—	1	—	在第 2 次传送结束 向 CPU 请求中断	

【注】 \*1 正常传送模式：CRA；重复传送模式：CRAL；块传送模式：CRB

\*2 表示 CRAL 的内容被改写为 CRAH 的内容时。

### 8.5.1 传送信息的跳读功能

能通过设定 DTCCR 的 RRS 位，跳读向量地址和传送信息。DTC 向量号随时和上次启动的向量号进行比较，如果比较结果一致并且 RRS=1，就不读向量地址和传送信息而进行 DTC 的数据传送。如果上次的启动是链传送，就必须读向量地址和传送信息。传送信息的跳读时序图如图 8.5 所示。

要更新向量表和传送信息时，必须先设定 RRS=0，然后更新向量表和传送信息，最后设定 RRS 位（置 1）。如果设定 RRS=0，所保持的向量号就被取消，并在下次启动时读取已更新的向量表和传送信息。

但是，在总线功能扩展寄存器（BSCEHR）的 DTPR 位是 1 的情况下，本功能总是无效。

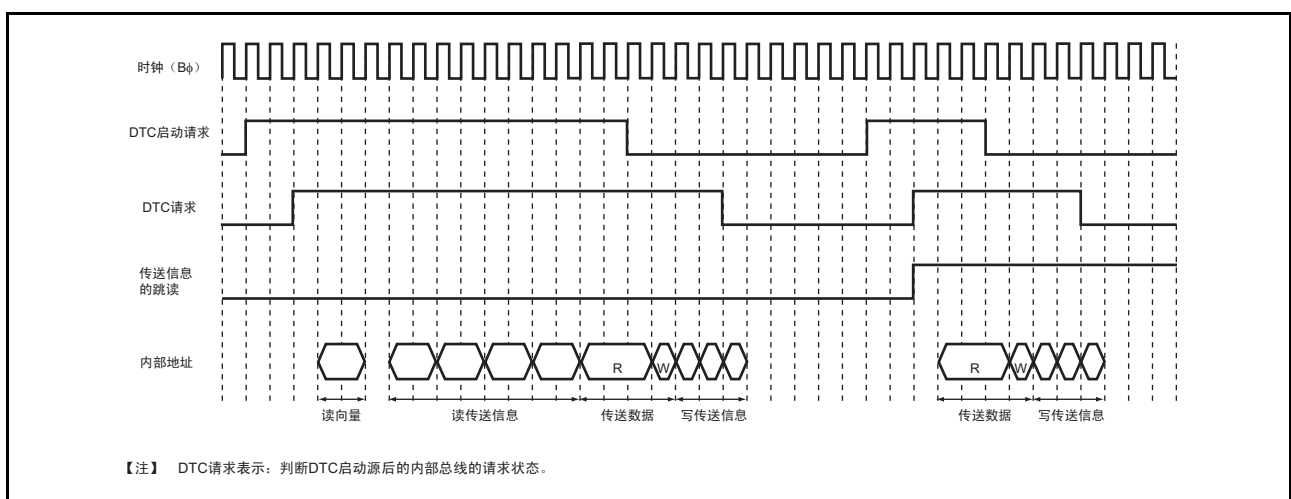


图 8.5 传送信息的跳读时序图

（从内部外围模块启动，I<sub>φ</sub>:B<sub>φ</sub>:P<sub>φ</sub>=1:1/2:1/2，从内部外围模块将数据传送到内部 RAM，传送信息的写操作为 3 个状态）

### 8.5.2 传送信息的回写省略功能

如果将 MRA 的 SM1 位和 MRB 的 DM1 位设定为固定地址，就不回写一部分传送信息。传送信息的回写省略条件和省略回写的寄存器如表 8.5 所示。总是回写 CRA 和 CRB，但是不回写 MRA 和 MRB。

表 8.5 传送信息的回写省略条件和省略回写的寄存器

SM1	DM1	SAR	DAR
0	0	省略	省略
0	1	省略	回写
1	0	回写	省略
1	1	回写	回写

### 8.5.3 正常传送模式

用一个启动源进行 1 个字节、1 个字或者 1 个长字的数据传送，传送次数为 1 ~ 65536。能分别将传送源地址和传送目标地址设定为增量、减量或者固定。当指定次数的传送结束时，就能向 CPU 请求中断。

正常传送模式的寄存器功能和存储器映像分别如表 8.6 所示和图 8.6 所示。

表 8.6 正常传送模式的寄存器功能

寄存器	功能	回写的值
SAR	传送源的地址	增量 / 减量 / 固定 *
DAR	传送目标的地址	增量 / 减量 / 固定 *
CRA	传送计数 A	CRA-1
CRB	传送计数 B	不更新

【注】 \* 不进行传送信息的回写。

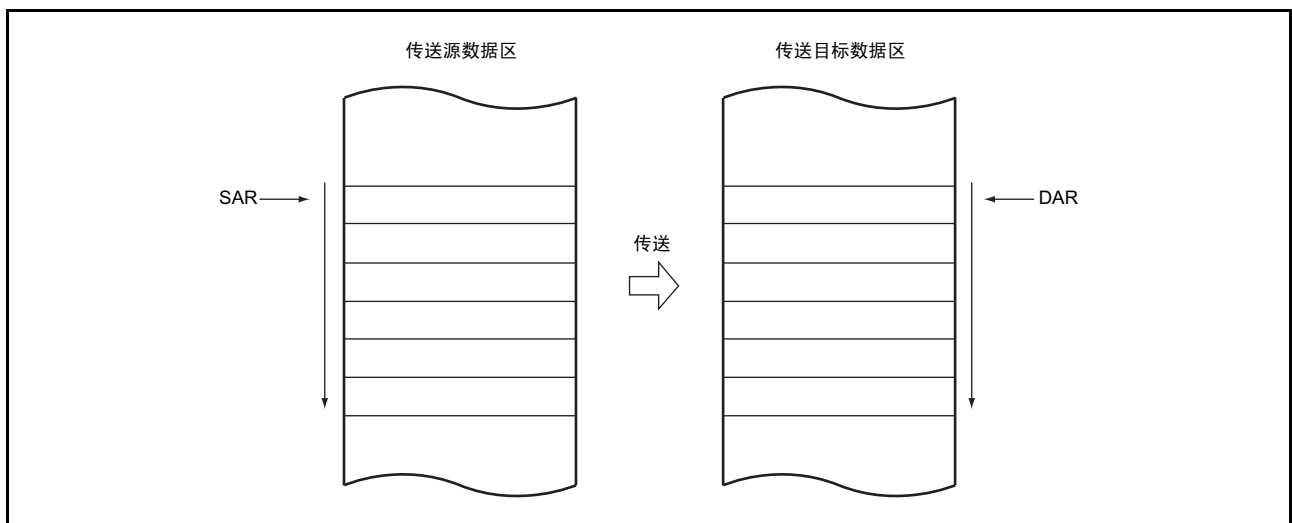


图 8.6 正常传送模式的存储器映像

### 8.5.4 重复传送模式

用一个启动源进行 1 个字节、1 个字或者 1 个长字的数据传送。通过 MRB 的 DTS 位将传送源或者传送目标指定为重复区域，传送次数为 1 ~ 256。当指定次数的传送结束时，传送计数器和被指定为重复区域的地址寄存器恢复到初始状态，进行重复传送。其他地址寄存器被连续增量、减量或者固定。在重复传送模式中，当传送计数器 (CRAL) 为 H'00 时，CRAL 就被更新为 CRAH 的值，因此传送计数器不会变为 H'00，所以在 DISEL=0 时不向 CPU 请求中断。

重复传送模式的寄存器功能和存储器映像分别如表 8.7 和图 8.7 所示。

表 8.7 重复传送模式中的寄存器功能

寄存器	功能	回写的值	
		CRAL≠1	CRAL=1
SAR	传送源地址	增量 / 减量 / 固定 *	(DTS=0) 增量 / 减量 / 固定 * (DTS=1) SAR 的初始值
DAR	传送目标地址	增量 / 减量 / 固定 *	(DTS=0) DAR 的初始值 (DTS=1) 增量 / 减量 / 固定 *
CRAH	保存传送计数	CRAH	CRAH
CRAL	传送计数 A	CRAL-1	CRAH
CRB	传送计数 B	不更新	不更新

【注】 \* 不进行传送信息的回写。

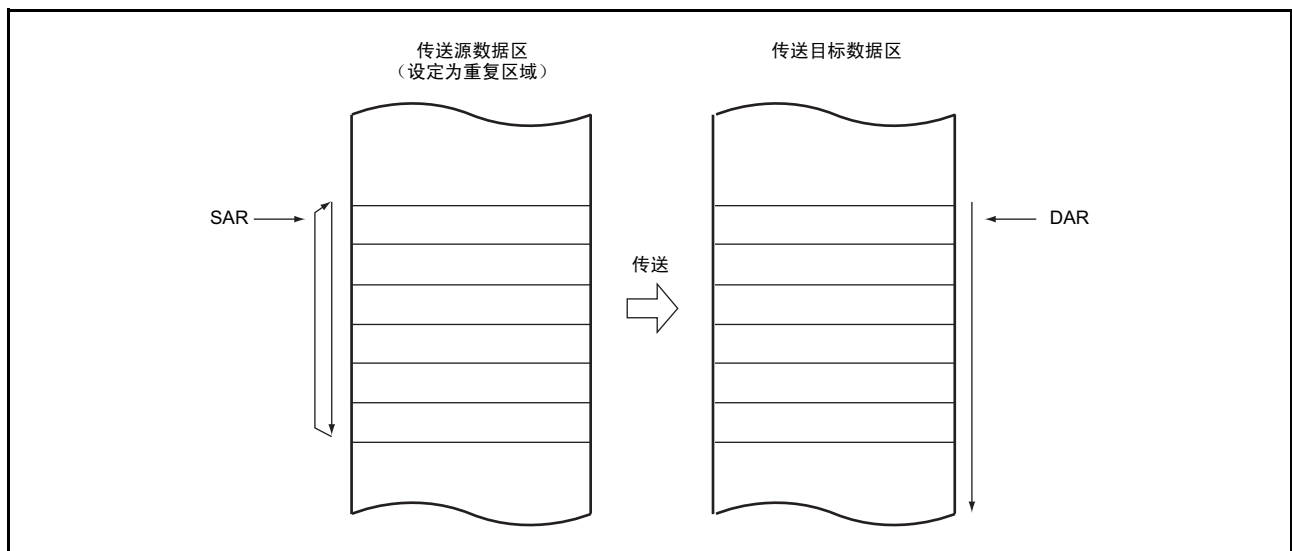


图 8.7 重复传送模式的存储器映像 (将传送源指定为重复区域时)



### 8.5.5 块传送模式

用一个启动源进行 1 块的块数据传送。通过 MRB 的 DTS 位将传送源或者传送目标指定为块区域，块大小为 1 ~ 256 个字节（1 ~ 256 个字或者 1 ~ 256 长字）。当 1 块的块数据传送结束时，块大小的计数器（CRAL）和被指定为块区域的地址寄存器（在 DTS=1 时为 SAR，在 DTS=0 时为 DAR）恢复到初始状态。其他地址寄存器被连续增量、减量或者固定。传送次数为 1 ~ 65536，当指定次数的块传送结束时，能向 CPU 请求中断。

块传送模式的寄存器功能和存储器映像分别如表 8.8 和图 8.8 所示。

表 8.8 块传送模式的寄存器功能

寄存器	功能	回写的值
SAR	传送源地址	(DTS=0) 增量 / 减量 / 固定 * (DTS=1) SAR 的初始值
DAR	传送目标地址	(DTS=0) DAR 的初始值 (DTS=1) 增量 / 减量 / 固定 *
CRAH	保存块大小	CRAH
CRAL	块大小的计数器	CRAH
CRB	块传送次数的计数器	CRB-1

【注】 \* 不进行传送信息的回写。

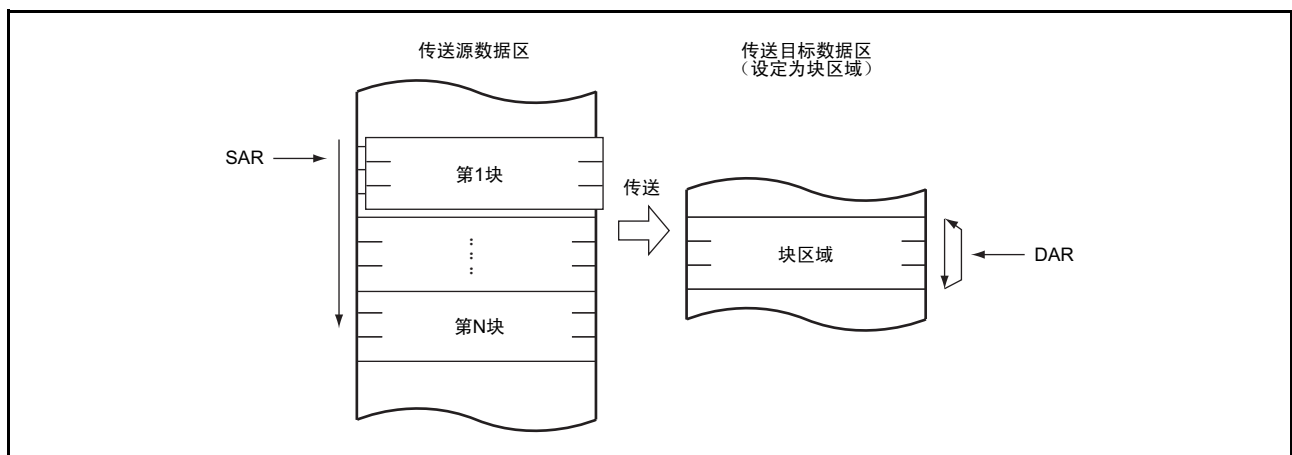


图 8.8 块传送模式的存储器映像（将传送目标指定为块区域时）

### 8.5.6 链传送

如果将 MRB 的 CHNE 位设定为 1，就能用一个启动源连续进行多个数据的传送；如果将 MRB 的 CHNE 位和 CHNS 位分别置 1，就只能在传送计数器 =0 时进行链传送。能分别对定义数据传送的 SAR、DAR、CRA、CRB、MRA 和 MRB 进行独立设定。链传送的运行如图 8.9 所示。

在设定 CHNE=1 时的数据传送中，不会在指定次数的传送结束或者因 DISEL=1 时向 CPU 请求中断。另外，在 CHNE=1 时的传送中，不影响 DTCCER 和作为启动源的中断源标志。

在重复传送模式中，如果分别将 DTCCR 的 RCHNE 位、MRB 的 CHNE 和 CHNS 位设定为 1，就能在计数器 =1 时的传送后进行链传送。

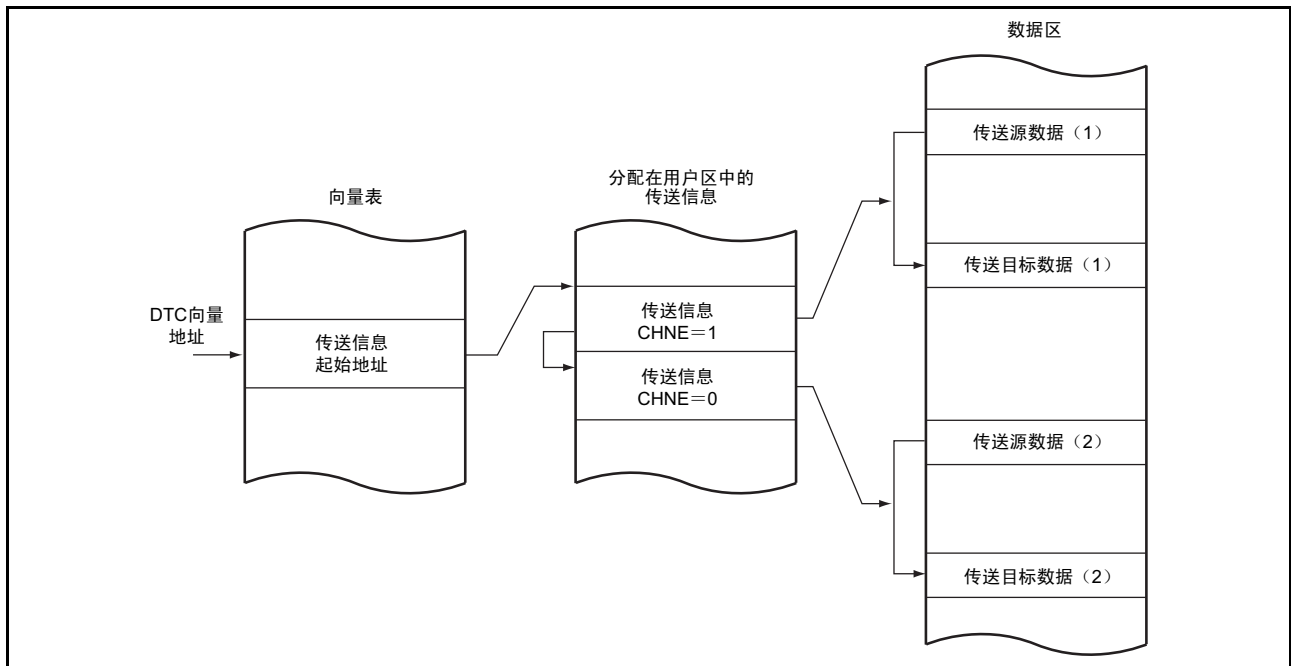


图 8.9 链传送的运行

### 8.5.7 运行时序

DTC 的运行时序如图 8.10 ~图 8.15 所示。

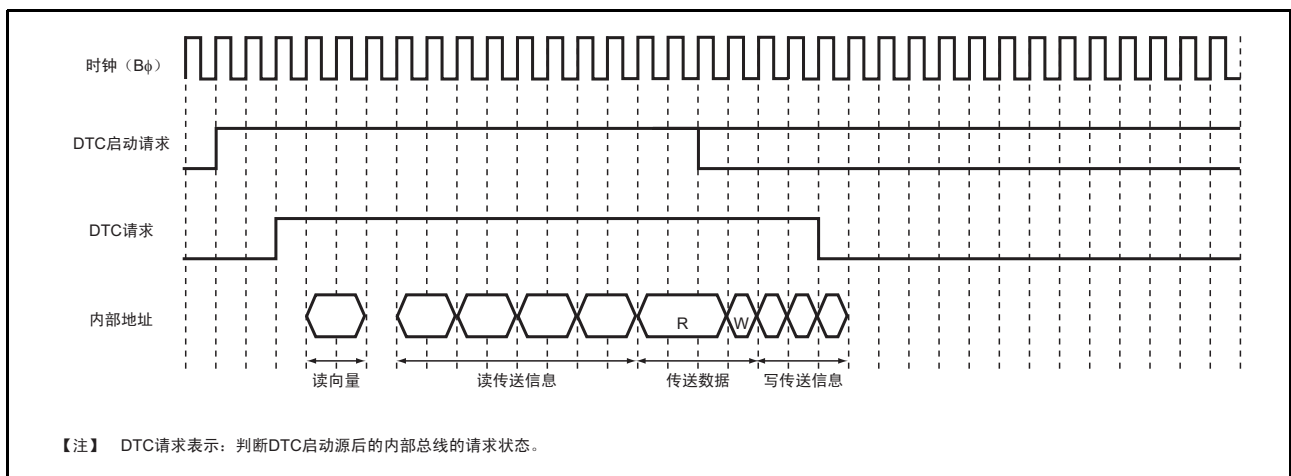


图 8.10 DTC 的运行时序例【正常传送或者重复传送】  
 (从内部外围模块启动, Iφ:Bφ:Pφ=1:1/2:1/2,  
 从内部外围模块将数据传送到内部 RAM, 传送信息的写操作为 3 个状态)

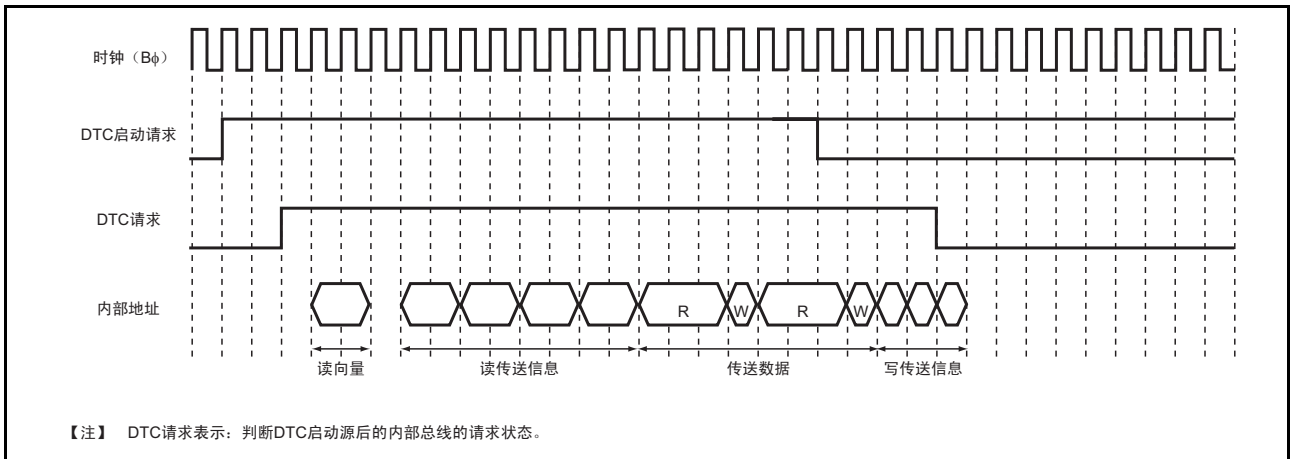


图 8.11 DTC 的运行时序例【正常传送，块大小 =2】  
 (从内部外围模块启动，I<sub>φ</sub>:B<sub>φ</sub>:P<sub>φ</sub>=1:1/2:1/2，  
 从内部外围模块将数据传送到内部 RAM，传送信息的写操作为 3 个状态)

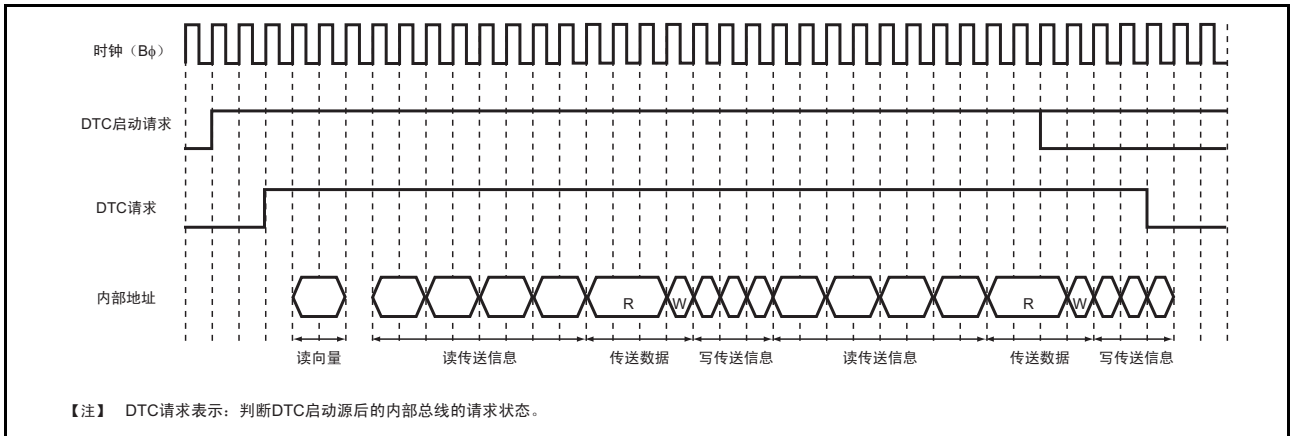


图 8.12 DTC 的运行时序例【链传送】  
 (从内部外围模块启动，I<sub>φ</sub>:B<sub>φ</sub>:P<sub>φ</sub>=1:1/2:1/2，  
 从内部外围模块将数据传送到内部 RAM，传送信息的写操作为 3 个状态)

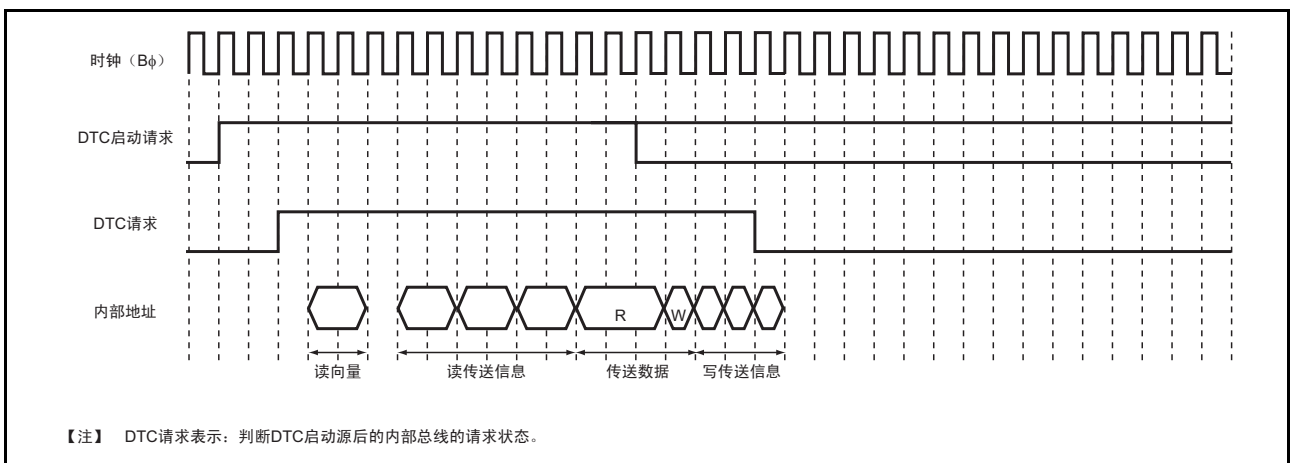


图 8.13 DTC 的运行时序例【短地址模式：正常传送或者重复传送】  
 (从内部外围模块启动，I<sub>φ</sub>:B<sub>φ</sub>:P<sub>φ</sub>=1:1/2:1/2，  
 从内部外围模块将数据传送到内部 RAM，传送信息的写操作为 3 个状态)

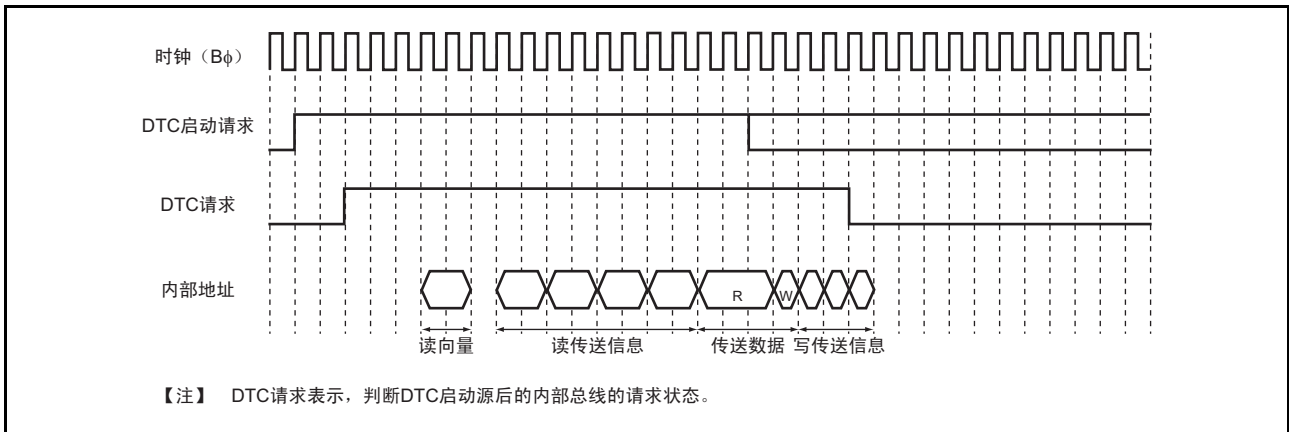


图 8.14 DTC 的运行时序例【正常传送或者重复传送，DTPR=1】  
 (从内部外围模块启动，Iφ:Bφ:Pφ=1:1/2:1/2，  
 从内部外围模块将数据传送到内部 RAM，传送信息的写操作为 3 个状态)

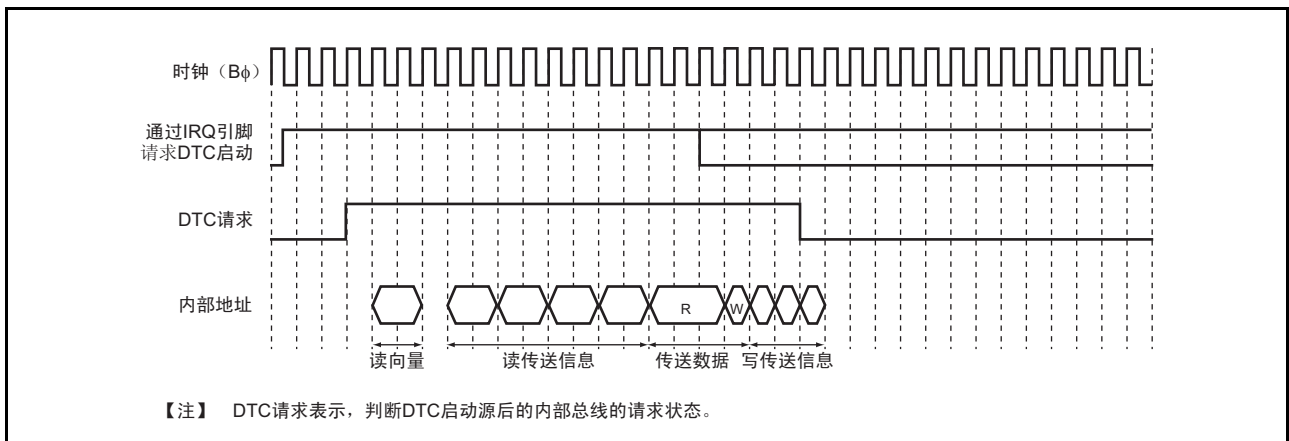


图 8.15 DTC 的运行时序例【正常传送模式或者重复传送】  
 (从 IRQ 启动，Iφ:Bφ:Pφ=1:1/2:1/2，  
 从内部外围模块将数据传送到内部 RAM，传送信息的写操作为 3 个状态)

### 8.5.8 DTC 的执行状态

DTC 的 1 次数据传送的执行状态以及执行状态所需的状态数分别如表 8.9 和表 8.10 所示。

表 8.9 DTC 的执行状态

模式	读向量 I		读传送信息 J			写传送信息 K			读数据 L	写数据 M	内部运行 N	
正常	1	0*1	4	3*4	0*1	3	2*2	1*3	1	1	1	0*1
重复	1	0*1	4	3*4	0*1	3	2*2	1*3	1	1	1	0*1
块	1	0*1	4	3*4	0*1	3	2*2	1*3	1 · P	1 · P	1	0*1

- 【注】 \*1 在跳读传送信息时  
 \*2 在 SAR 或者 DAR 为固定模式时  
 \*3 在 SAR 和 DAR 都是固定模式时  
 \*4 在短地址模式时  
 P: 块大小 (CRAH和CRAL的初始设定)

表 8.10 执行状态所需的状态数

存取对象		内部 RAM*1/ROM*2	内部 I/O 寄存器		外部器件 *5		
总线宽度		32 位	8 位 *4	16 位	8 位	16 位	32 位
存取状态		1B $\phi$ ~ 3B $\phi$ *1*2	2P $\phi$	2P $\phi$	2B $\phi$	2B $\phi$	2B $\phi$
执行 状态	读向量 S <sub>I</sub>	1B $\phi$ ~ 3B $\phi$ *1*2	—	—	9B $\phi$	5B $\phi$	3B $\phi$
	读传送信息 S <sub>J</sub>	1B $\phi$ ~ 3B $\phi$ *1	—	—	9B $\phi$	5B $\phi$	3B $\phi$
	写传送信息 S <sub>K</sub>	1B $\phi$ ~ 3B $\phi$ *1	—	—	2B $\phi$ *6	2B $\phi$ *6	2B $\phi$ *6
	读字节数据 S <sub>L</sub>	1B $\phi$ ~ 3B $\phi$ *1	1B $\phi$ +2P $\phi$ *3	1B $\phi$ +2P $\phi$ *3	3B $\phi$	3B $\phi$	3B $\phi$
	读字数据 S <sub>L</sub>	1B $\phi$ ~ 3B $\phi$ *1	—	1B $\phi$ +2P $\phi$ *3	5B $\phi$	3B $\phi$	3B $\phi$
	读长字数据 S <sub>L</sub>	1B $\phi$ ~ 3B $\phi$ *1	—	1B $\phi$ +4P $\phi$ *3	9B $\phi$	5B $\phi$	3B $\phi$
	写字节数据 S <sub>M</sub>	1B $\phi$ ~ 3B $\phi$ *1	1B $\phi$ +2P $\phi$ *3	1B $\phi$ +2P $\phi$ *3	2B $\phi$ *6	2B $\phi$ *6	2B $\phi$ *6
	写字数据 S <sub>M</sub>	1B $\phi$ ~ 3B $\phi$ *1	—	1B $\phi$ +2P $\phi$ *3	2B $\phi$ *6	2B $\phi$ *6	2B $\phi$ *6
	写长字数据 S <sub>M</sub>	1B $\phi$ ~ 3B $\phi$ *1	—	1B $\phi$ +4P $\phi$ *3	2B $\phi$ *6	2B $\phi$ *6	2B $\phi$ *6
	内部运行 S <sub>N</sub>		1				

【注】 \*1 存取对象是内部 RAM。周期数因 I $\phi$ :B $\phi$  比而不同。

	读	写
当 I $\phi$ :B $\phi$ =1:1 时	3B $\phi$	3B $\phi$
当 I $\phi$ :B $\phi$ =1:1/2 时	2B $\phi$	1B $\phi$
当 I $\phi$ :B $\phi$ =1:1/3 时	2B $\phi$	1B $\phi$
当 I $\phi$ :B $\phi$ =1:1/4 (小于等于 1/4) 时	1B $\phi$	1B $\phi$

\*2 存取对象是内部 ROM。周期数因 I $\phi$ :B $\phi$  比而不同，和内部 RAM 相等。只能读向量。

\*3 表中是最快的情况。在因内部总线的状态而变慢的情况下，1B $\phi$  为 1P $\phi$ 。

\*4 应该为 I<sup>2</sup>C<sup>2</sup>。

\*5 因 BSC 寄存器的设定而不同。表中是等待 =0 并且 CSnWCR 的 WM 位 =1 时的例子。

\*6 因总线的状态而不同。

如果正在使用外部总线或者经常发生如块传送写操作时的外部等待时间长，就不能有效利用写缓冲区而增加周期数。

有关写缓冲区的详细内容请参照“9.5.14(2) 从 LSI 内部总线主控器的存取”。

通过以下计算式求执行状态数：

另外，表示用 1 个启动源进行传送的次数（CHNE 位被置 1 后的数 +1）的总和。

$$\text{执行状态数} = I \cdot S_I + \sum (J \cdot S_J + K \cdot S_K + L \cdot S_L + M \cdot S_M) + N \cdot S_N$$

### 8.5.9 DTC 的总线权释放时序

当发生启动请求时，DTC 对总线仲裁器请求总线权。在读向量后、产生读向量后的 NOP 时、读传送信息后、1 次数据传送后或者回写传送信息后，DTC 释放总线权，而正在读传送信息、正在进行 1 次数据传送或者正在回写传送信息时不释放总线权。

能通过总线功能扩展寄存器 (BSCEHR) 设定总线权的释放时序，详细内容请参照“9.4.8 总线功能扩展寄存器 (BSCEHR)”。根据设定，不同的总线权释放时序如表 8.11 所示。但是，不能进行设定 1 ~ 设定 5 以外的设定，并且不能在 DTC 启动时更改设定。

时序图如图 8.16 所示。

表 8.11 DTC 的总线权释放时序

设定	总线功能扩展寄存器 (BSCEHR) 的设定					总线权的释放时序 (○: 释放总线权, ×: 不释放总线权)					
	DTLOCK	CSSTP1	CSSTP2	CSSTP3	DTBST	读向量后	产生 NOP 时 *1	读传送信息后	1 次传送后	写传送信息后	
										通常	连续传送时
设定 1*4	1	0	*3	1	0	○	○	○	○	○	○
设定 2*3	0	0	0	*3	0	×	○	×	×	○	○
设定 3	0	1	*3	*3	0	×	×	×	×	○	○
设定 4*2	0	1	*3	*3	1	×	×	×	×	○	×
设定 5	1	1	*3	1	0	○	×	○	○	○	○

【注】 \*1 只对读向量后的 CPU 外部存取请求，释放总线权。

\*2 设定 4 有以下限制：

- 必须通过频率控制寄存器 (FRQCR) 将时钟设定为  $l\phi:B\phi:P\phi:M\phi:MP\phi=8:4:4:4:4$ 、 $4:2:2:2:2$  或者  $2:1:1:1:1$ 。
- 必须将向量信息分配到内部 ROM 或者内部 RAM。
- 必须将传送信息分配到内部 RAM。
- 传送源和传送目标必须在内部 RAM 和内部外围模块之间或者外部存储器和内部外围模块之间。

\*3 Don't care

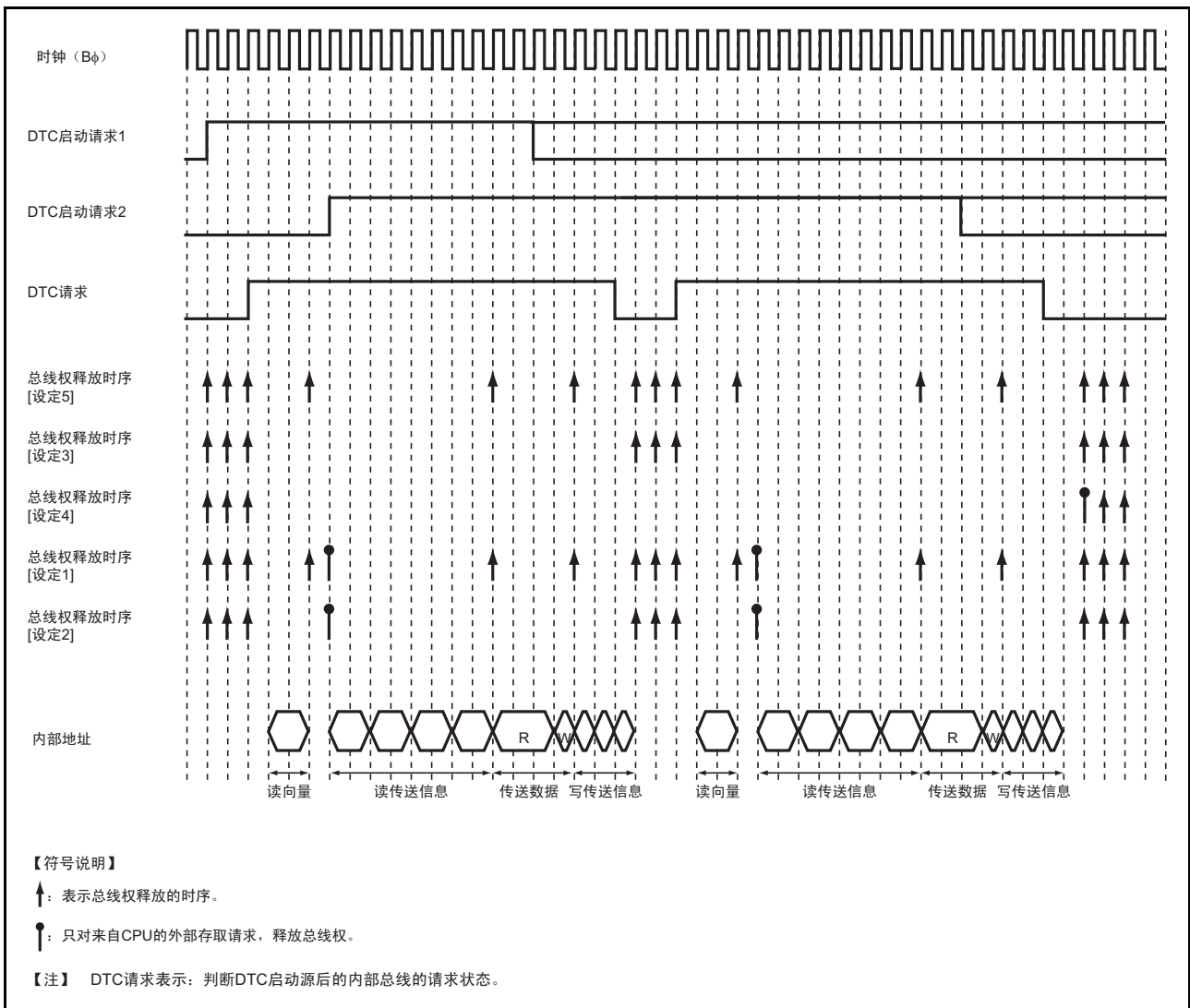


图 8.16 DTC 的运行时序例【正常传送中 2 个源竞争时】  
 (从内部外围模块启动,  $I\phi:B\phi:P\phi=1:1/2:1/2$ ,  
 从内部外围模块将数据传送到内部 RAM, 传送信息的写操作为 3 个状态)

## 8.5.10 DTC 启动优先级的设定

通过设定总线功能扩展寄存器 (BSCEHR) 的 DTPR 位, 就能在启动 DTC 前发生多个 DTC 启动请求时, 选择是从最初发生的启动请求开始传送还是根据 DTC 启动优先级开始传送。但是, 如果在启动 DTC 时发生多个 DTC 启动请求, 就根据 DTC 启动优先级进行下一次传送。DTC 启动优先级的运行例子如图 8.17 所示。

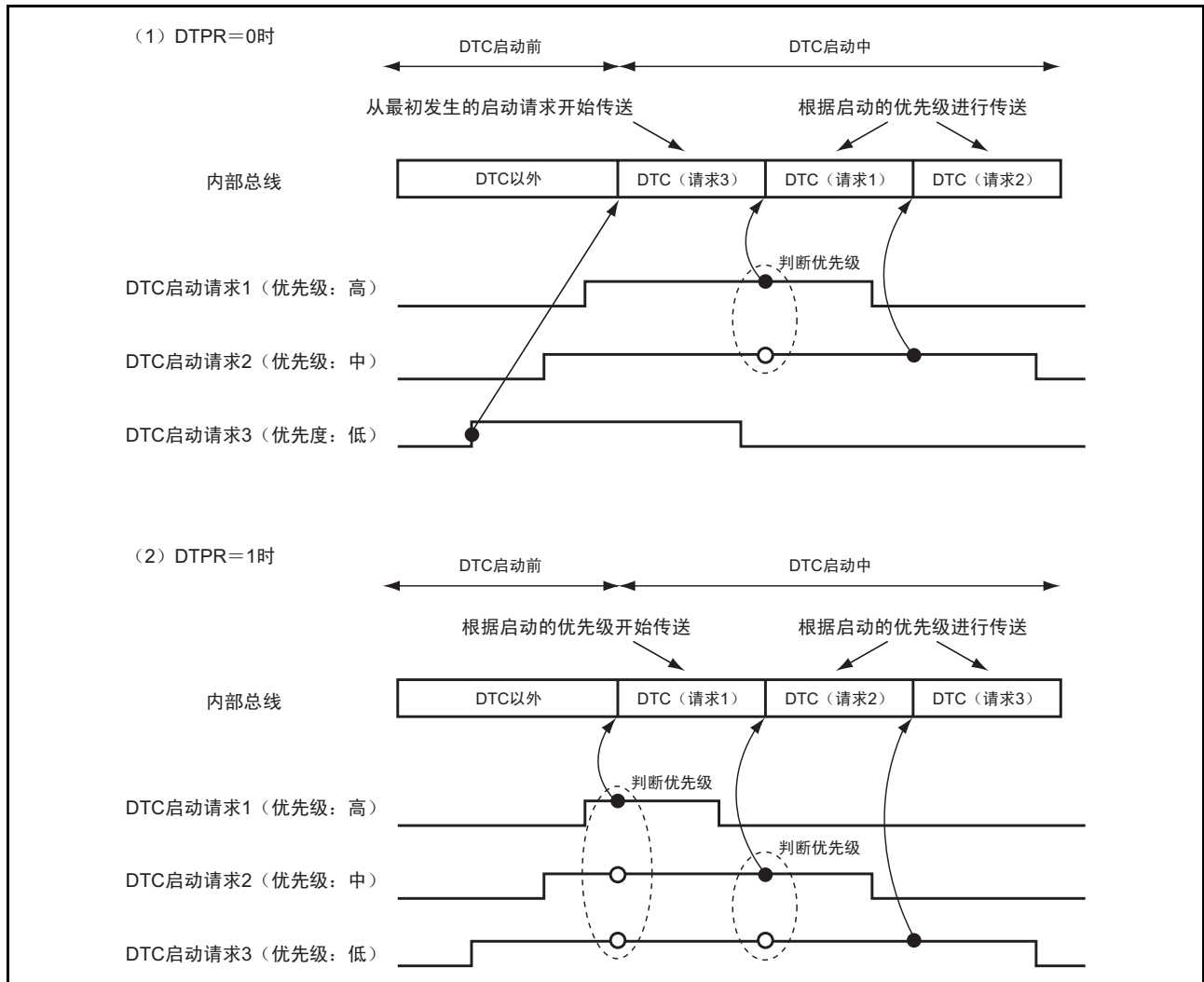


图 8.17 DTC 启动优先级的运行例子



### 8.6 通过中断启动 DTC

通过中断启动 DTC 的使用步骤如图 8.18 所示。

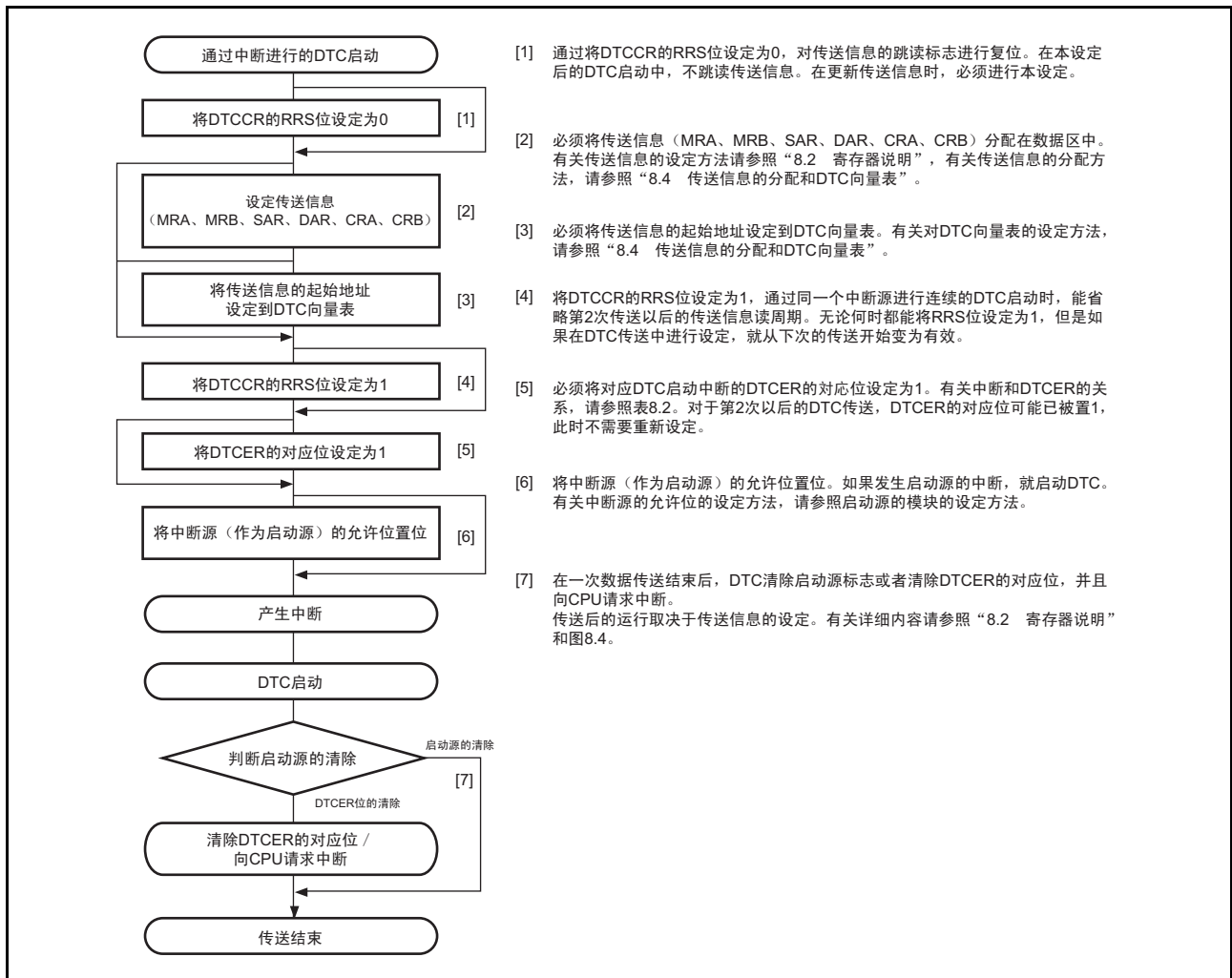


图 8.18 通过中断启动 DTC 的方法

## 8.7 DTC 使用例

### 8.7.1 正常传送

以通过 SCI 进行 128 个字节的数据接收为例说明 DTC 的使用例：

1. MRA 设定为源地址固定 (SM1=SM0=0)、目标地址增量 (DM1=1、DM0=0)、正常传送模式 (MD1=MD0=0) 和字节长度 (Sz1=Sz0=0)，DTS 位可为任意值。MRB 设定为通过 1 次中断进行 1 次数据传送 (CHNE=0、DISEL=0)。SAR 设定为 SCI 的 RDR 地址，DAR 设定为保存数据的 RAM 起始地址，CRA 设定为 128 (H'0080)，CRB 可为任意值。
2. 将用于 RXI 中断的传送信息的起始地址设定到 DTC 向量表。
3. 将 DTCER 的对应位设定为 1。
4. 将 SCI 设定为规定的接收模式。将 SCR 的 RIE 位设定为 1，允许接受结束 (RXI) 中断。另外，如果在 SCI 接收运行时发生接收错误，就不进行以后的接收，所以必须设定为 CPU 能接受接收错误中断。
5. 每当 SCI 的 1 个字节的数据接收结束时，SSR 的 RDRF 标志就被置 1，产生 RXI 中断，启动 DTC。通过 DTC 将接收数据从 RDR 传送到 RAM，进行 DAR 的增量和 CRA 的减量。RDRF 标志自动清零。
6. 如果在 128 次的数据传送结束后 CRA 变为 0，RDRF 标志就保持 1，DTCE 位被清零，并向 CPU 请求 RXI 中断。必须通过中断处理程序结束处理。

### 8.7.2 计数器 =0 的链传送

通过只在计数器变为 0 时进行第 2 数据传送并且重新设定第 1 数据传送，就能进行传送次数  $\geq 256$  次的重复传送。

这是构成 128K 字节输入缓冲区的例子。在此，输入缓冲区被设定为从低位地址 H'0000 开始。计数器 =0 的链传送如图 8.19 所示。

1. 作为第 1 数据传送，设定用于输入数据的正常传送模式。传送源地址必须固定、CRA=H'0000 (65,536 次)、CHNE=1、CHNS=1、DISEL=0。
2. 必须在其他区域 (ROM 等) 准备第 1 数据传送的传送目标起始地址 (65,536 次传送单位) 的高 8 位地址。例如，在输入缓冲区为 H'200000 ~ H'21FFFF 时准备 H'21 和 H'20。
3. 作为第 2 数据传送，设定用于重新设定第 1 数据传送的传送目标地址的重复传送模式 (源为重复区域)。传送目标作为第 1 传送信息区的 DAR 的高 8 位。此时，必须设定 CHNE=DISEL=0。如果将上述输入缓冲区为 H'200000 ~ H'21FFFF，就必须设定传送计数器 =2。
4. 通过中断执行 65,536 次第 1 数据传送。当第 1 数据传送的传送计数器变为 0 时，启动第 2 数据传送。将第 1 数据传送的传送源地址的高 8 位设定为 H'21，第 1 数据传送的传送目标地址的低 16 位的传送计数器为 H'0000。
5. 接着，通过中断执行由第 1 传送指定的 65,536 次第 1 数据传送。当第 1 数据传送的传送计数器变为 0 时，启动第 2 数据传送。将第 1 数据传送的传送源地址的高 8 位设定为 H'20，第 1 数据传送的传送目标地址低 16 位的传送计数器为 H'0000。
6. 无限重复上述的项 4 和项 5。因为第 2 数据传送是重复传送模式，所以不向 CPU 请求中断。

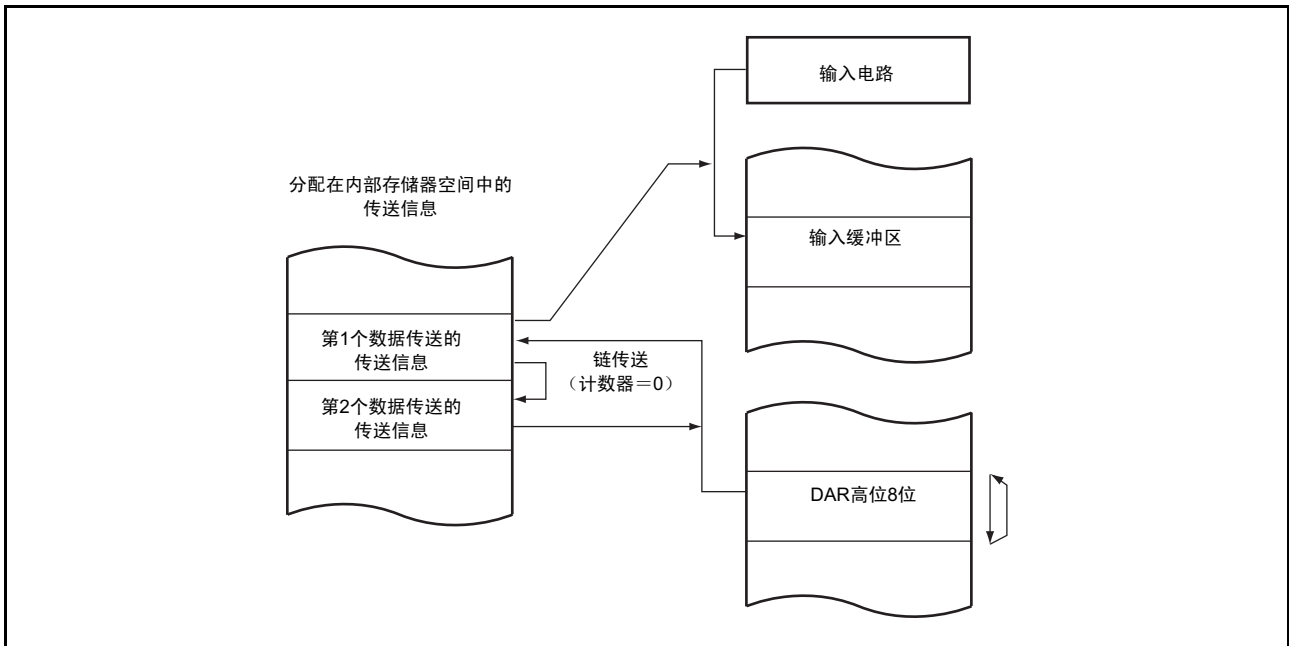


图 8.19 计数器 =0 的链传送

## 8.8 中断源

在 DTC 结束指定次数的数据传送时、或者在 **DISEL** 位被置 1 的 1 次数据传送或者 1 次块数据传送结束时，向 CPU 请求中断。在中断启动时，产生设定为启动源的中断。这些对 CPU 的中断受的屏蔽级或者中断控制器的优先级控制。详细内容请参照“6.8 通过中断请求信号的数据传送”。

## 8.9 使用注意事项

### 8.9.1 模块待机模式的设定

能通过待机控制寄存器，禁止或者允许 DTC 的运行，初始值为禁止 DTC 运行。在设定为模块待机模式时，禁止 DTC 的运行，但是不禁止寄存器的存取。不能在启动 DTC 时设定为模块待机模式。如果要转换到软件待机模式或者模块待机模式，就必须清除全部 **DTCER** 寄存器。详细内容请参照“第 26 章 低功耗模式”。

### 8.9.2 内部 RAM

能将传送信息分配到内部 RAM。此时，必须将 **RAMCR** 的 **RAME** 位清 0。

### 8.9.3 DTCE 位的设定

要设定 **DTCE** 位时，先禁止中断，然后在读到相应寄存器的 0 后写 1。另外，不能在传送 DTC 时更改 **DTCE** 位。

### 8.9.4 链传送

在执行链传送时，在最后连接数据的传送时清除启动源或者 **DTCER**。另一方面，在读写规定的寄存器时，清除 **SCI**、**SCIF**、**SSU**、**I<sup>2</sup>C2** 和 A/D 转换器的中断 / 启动源。

### 8.9.5 传送信息起始地址 / 源地址 / 目标地址

必须将指向向量表的传送信息的起始地址指定为地址  $4n$ 。

另外，必须将传送信息分配到内部 RAM 或者外部存储器空间。

### 8.9.6 通过 DTC 存取 DMAC/DTC 寄存器

不能使用 DTC 存取 DMAC/DTC 的寄存器，也不能使用 DMAC 存取 DTC 的寄存器。

### 8.9.7 IRQ 中断作为 DTC 传送源时的注意事项

- 不能通过该 IRQ 中断解除软件待机模式。
- 不能在软件待机时产生的 IRQ 边沿进行 DTC 传送。
- 在检测到 IRQ 的低电平时，为了在 DTC 传送结束后向 CPU 请求中断（传送计数器 =0 或者 DISEL=1），必须在 CPU 接受中断前保持 IRQ 引脚的低电平。

### 8.9.8 SCI 和 SCIF 作为 DTC 启动源时的注意事项

- 当通过 SCI 的 TXI 中断启动 DTC 时，不能将 SCI 的 TEND 标志用作传送结束标志。
- 当通过 SCIF 的 TXIF 中断启动 DTC 时，不能将 SCIF 的 TEND 标志用作传送结束标志。

### 8.9.9 中断源标志的清除

对于 DTC 传送结束后产生的中断，和通常的中断一样，必须在中断处理程序中清除中断源标志。详细内容请参照“6.9 使用注意事项”。

### 8.9.10 NMI 中断和 DTC 启动的竞争

在 NMI 中断和 DTC 启动发生竞争时优先 NMI 中断，所以 ERR 位被置 1，并且不启动 DTC。

另外，NMI 产生的 DTC 停止判断需要 1 个 Bcyc+3 个 Pcyc 周期；IRQ 产生的 DTC 启动判断需要 2 个 Bcyc 周期；外围模块产生的 DTC 启动判断需要 1 个 Pcyc 周期。

### 8.9.11 DTC 启动请求在中途被取消时的运行

DTC 从接受请求后起，直到回写完成的一连串 DTC 处理结束为止，不接受下一个启动请求。

## 第 9 章 总线状态控制器 (BSC)

总线状态控制器 (BSC) 对连接外部地址空间的各种存储器和外部器件输出控制信号, 能直接连接 SRAM、SDRAM 等各种存储器和外部器件。

### 9.1 特点

#### 1. 外部地址空间

- CS0 ~ CS7 各空间最大支持 64MB, CS8 空间最大支持 1GB
- 每个空间都能指定通常空间接口、带字节选择的 SRAM 接口、突发 ROM (时钟同步或者异步)、MPX-I/O、突发 MPX-I/O、SDRAM 和 PCMCIA 的存储器种类
- 每个空间都可选择数据总线宽度 (8 位、16 位或者 32 位)
- 每个空间都能控制等待状态的插入
- 每次读写存取都能控制等待状态的插入
- 在连续存取为读 - 写 (同一空间或者不同空间)、读 - 读 (同一空间或者不同空间) 或者起始周期为写周期的 5 种情况下, 能独立设定空闲周期

#### 2. 通常空间接口

- 支持能直接连接 SRAM 的接口

#### 3. 突发 ROM (异步) 接口

能高速存取具有页模式功能的 ROM

#### 4. MPX-I/O 接口

能直接连接地址 / 数据多路复用所需的外围 LSI

#### 5. SDRAM 接口

- 最多能在 2 个区域设定 SDRAM
- 支持行地址或者列地址的多路复用输出
- 能通过单次读或者单次写进行高效存取
- 能通过存储体激活模式进行高速存取
- 支持自动刷新和自刷新

#### 6. 带字节选择的 SRAM 接口

- 支持能直接连接带字节选择的 SRAM 接口

#### 7. 直接连接 PCMCIA 的接口

- 支持 JEIDA 规格的 Ver4.2 (PCMCIA2.1 Rev2.1) 规定的 IC 存储卡和 I/O 卡接口
- 能通过程序控制等待状态的插入

#### 8. 突发 MPX-I/O 接口

- 能直接连接地址 / 数据多路复用所需的外围 LSI
- 支持突发传送

#### 9. 突发 ROM (时钟同步) 接口

- 能直接连接时钟同步型的突发 ROM

#### 10. 刷新功能

- 支持自动刷新和自刷新
- 能通过刷新计数器或者时钟选择来设定刷新间隔
- 能通过设定刷新次数 (1、2、4、6 或者 8) 集中刷新

#### 11. 将刷新计数器用作间隔定时器

- 能通过比较匹配产生中断请求

BSC 的框图如图 9.1 所示。

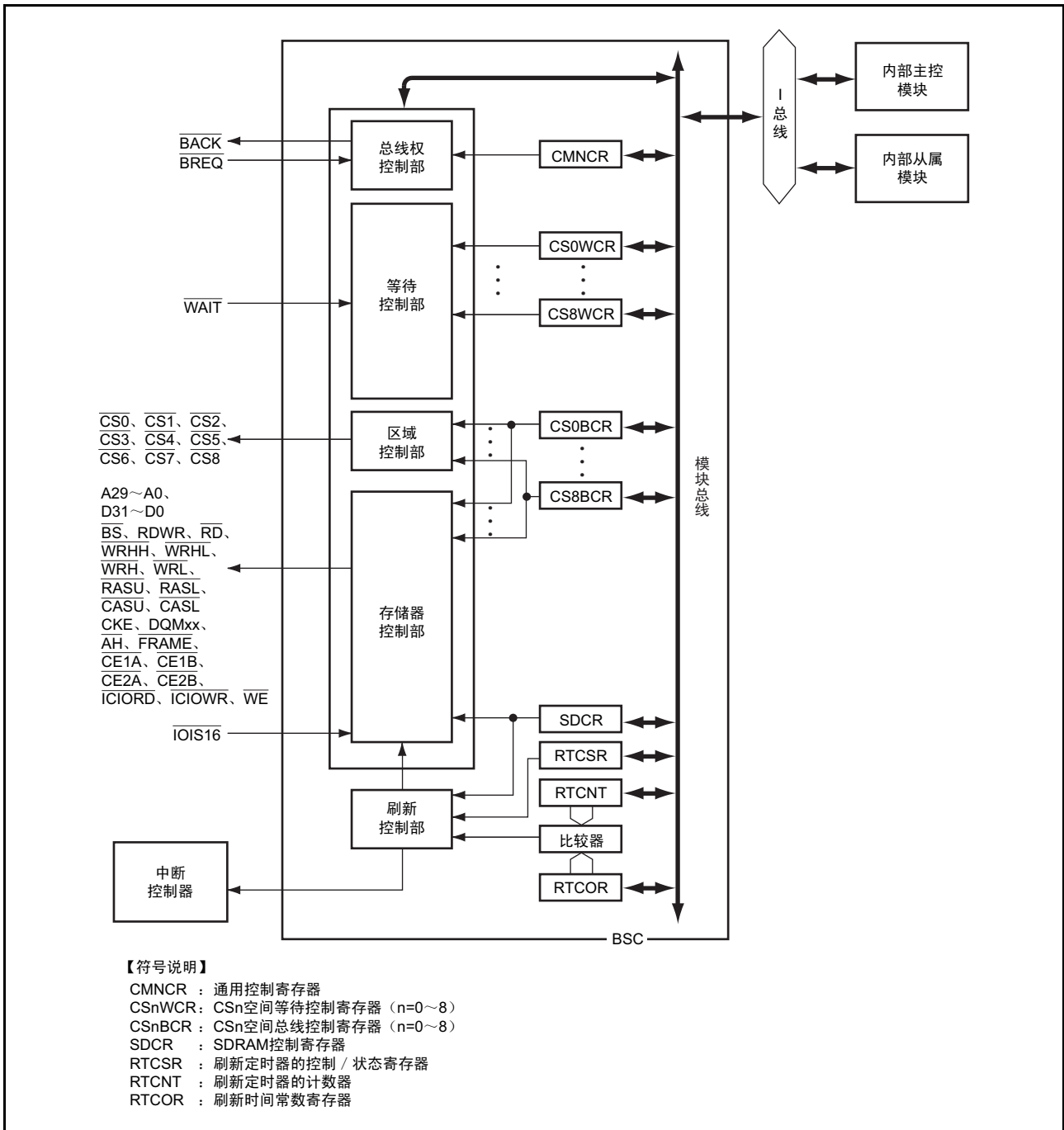


图 9.1 BSC 的框图

## 9.2 输入 / 输出引脚

BSC 的引脚结构如表 9.1 所示。

表 9.1 引脚结构

引脚名	输入 / 输出	功 能
A29 ~ A0	输出	地址总线
D31 ~ D0	输入 / 输出	数据总线
$\overline{BS}$	输出	总线周期的开始信号 在存取通常空间、突发 ROM (时钟同步 / 异步)、MPX-I/O、突发 MPX-I/O 以及 PCMCIA 时有效。在存取 SDRAM 时, 此引脚和 $\overline{CAS}$ 同时有效。
$\overline{CS0} \sim \overline{CS8}$	输出	片选
$\overline{CE1A}$	输出	允许连接区域 5 的 PCMCIA 的片选
$\overline{CE2A}$	输出	允许连接区域 5 的 PCMCIA 的片选
$\overline{CE1B}$	输出	允许连接区域 6 的 PCMCIA 的片选
$\overline{CE2B}$	输出	允许连接区域 6 的 PCMCIA 的片选
RDWR	输出	读写信号 在连接 SDRAM 或者带字节选择的 SRAM 时, 将此引脚连接到 WE 引脚
$\overline{RD}$	输出	读脉冲信号 (读数据输出的允许信号) 在使用 PCMCIA 时, 此引脚表示存储器读周期的门控信号
$\overline{WRHH}$	输出	对应 D31 ~ D24 的字节写指示 在连接带字节选择的 SRAM 时, 将此引脚连接到字节选择引脚
$\overline{WRHL}$	输出	对应 D23 ~ D16 的字节写指示 在连接带字节选择的 SRAM 时, 将此引脚连接到字节选择引脚
$\overline{WRH}$	输出	对应 D15 ~ D8 的字节写指示 在连接带字节选择的 SRAM 时, 将此引脚连接到字节选择引脚
$\overline{WRL}$	输出	对应 D7 ~ D0 的字节写指示 在连接带字节选择的 SRAM 时, 将此引脚连接到字节选择引脚
$\overline{RASU}$ 、 $\overline{RASL}$	输出	在连接 SDRAM 时, 将此引脚连接到 $\overline{RAS}$ 引脚
$\overline{CASU}$ 、 $\overline{CASL}$	输出	在连接 SDRAM 时, 将此引脚连接到 $\overline{CAS}$ 引脚
CKE	输出	在连接 SDRAM 时, 将此引脚连接到 CKE 引脚
$\overline{IOIS16}$	输入	PCMCIA 的 16 位 I/O 信号 因为本 LSI 不支持小端法, 所以必须将此引脚置为低电平。
DQMUU DQMUL DQMLU DQMLL	输出	在连接 SDRAM 时, 将此引脚连接到 DQMxx 引脚 DQMUU: 对应 D31 ~ D24 的选择信号 DQMUL: 对应 23 ~ D16 的选择信号 DQMLU: 对应 D15 ~ D8 的选择信号 DQMLL: 对应 D7 ~ D0 的选择信号
$\overline{AH}$	输出	使用 MPX-I/O 时的地址保持信号
$\overline{FRAME}$	输出	使用突发 MPX-I/O 接口时的 FRAME 信号
$\overline{WAIT}$	输入	外部等待的输入
$\overline{BREQ}$	输入	总线权请求的输入

引脚名	输入 / 输出	功 能
$\overline{\text{BACK}}$	输出	总线使用的允许输出
$\overline{\text{ICIORW}}$	输出	在使用 PCMCIA 时, 此引脚表示 I/O 写操作的门控信号
$\overline{\text{ICIODR}}$	输出	在使用 PCMCIA 时, 此引脚表示 I/O 读操作的门控信号
$\overline{\text{WE}}$	输出	在使用 PCMCIA 时, 此引脚表示存储器写周期的门控信号

### 9.3 区域概要

#### 9.3.1 空间分割

本 LSI 是 32 位地址空间的体系结构。

如表 9.2 ~ 表 9.15 所示, 本 LSI 能分别将各种存储器连接到 9 个空间, 输出对应各存储器的片选信号 ( $\overline{\text{CS0}} \sim \overline{\text{CS8}}$ )。CS0 在存取区域 0 时有效。在区域 2、3 上连接同步 DRAM 时,  $\overline{\text{RASx}}$ 、 $\overline{\text{CASx}}$ 、 $\overline{\text{RD}}/\overline{\text{WR}}$ 、 $\overline{\text{DQMxx}}$  等信号也有效。另外在区域 5 和 6 中, 选择 PCMCIA 接口时, 在相应的存取位上加  $\overline{\text{CS5}}$ 、 $\overline{\text{CS6}}$ 、 $\overline{\text{CE1A}}$ 、 $\overline{\text{CE1B}}$ 、 $\overline{\text{CE2A}}$ 、 $\overline{\text{CE2B}}$  有效。

#### 9.3.2 地址映像

外部地址空间共有 1.5G 字节, 被分割为 9 个空间使用。按空间指定被连接的存储器种类和数据总线宽度。全空间的地址映像如表 9.2 ~ 表 9.15 所示。

表 9.2 地址映像 (SH7083 (闪存 256KB 版)、内部 ROM 有效模式)

地址	空间	存储器种类	容量	总线宽度
H'00000000 ~ H'0003FFFF	内部 ROM		256K 字节	32 位
H'00040000 ~ H'01FFFFFF	保留			
H'02000000 ~ H'03FFFFFF	CS0 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步) 突发 ROM (同步)	32M 字节	8/16 位 *
H'04000000 ~ H'0BFFFFFF	保留			
H'0C000000 ~ H'0DFFFFFF	CS3 空间	通常空间 带字节选择的 SRAM SDRAM	32M 字节	8/16 位 *
H'0E000000 ~ H'1BFFFFFF	保留			
H'1C000000 ~ H'1DFFFFFF	CS7 空间	通常空间 带字节选择的 SRAM	32M 字节	8/16 位 *
H'1E000000 ~ H'FFF7FFFF	保留			
H'FFF80000 ~ H'FFF9FFFF	用于设定 SDRAM 模式			
H'FFFA0000 ~ H'FFFF7FFF	保留			
H'FFFF8000 ~ H'FFFFBFFF	内部 RAM		16K 字节	32 位
H'FFFFC000 ~ H'FFFFFFF	内部外围模块		16K 字节	8/16 位

【注】 不能存取保留空间, 否则就不能保证运行。在单芯片模式中, 不能使用内部 ROM、内部 RAM 和内部外围模块以外的空间。

\* 通过寄存器进行选择



表 9.3 地址映像 (SH7083 (闪存 256KB 版)、内部 ROM 无效模式)

地址	空间	存储器种类	容量	总线宽度
H'00000000 ~ H'01FFFFFF	CS0 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步) 突发 ROM (同步)	32M 字节	8/16 位 *1
H'02000000 ~ H'0BFFFFFF	保留			
H'0C000000 ~ H'0DFFFFFF	CS3 空间	通常空间 带字节选择的 SRAM SDRAM	32M 字节	8/16 位 *2
H'0E000000 ~ H'1BFFFFFF	保留			
H'1C000000 ~ H'1DFFFFFF	CS7 空间	通常空间 带字节选择的 SRAM	32M 字节	8/16 位 *2
H'1E000000 ~ H'FFF7FFFF	保留			
H'FFF80000 ~ H'FFF9FFFF	用于设定 SDRAM 模式			
H'FFFA0000 ~ H'FFFF7FFF	保留			
H'FFFF8000 ~ H'FFFFBFFF	内部 RAM		16K 字节	32 位
H'FFFC0000 ~ H'FFFFFFF	内部外围模块		16K 字节	8/16 位

【注】 不能存取保留空间，否则就不能保证运行。

\*1 通过模式引脚进行选择

\*2 通过寄存器进行选择

表 9.4 地址映像 (SH7083 (闪存 512KB 版)、内部 ROM 有效模式)

地址	空间	存储器种类	容量	总线宽度
H'00000000 ~ H'0007FFFF	内部 ROM		512K 字节	32 位
H'00080000 ~ H'01FFFFFF	保留			
H'02000000 ~ H'03FFFFFF	CS0 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步) 突发 ROM (同步)	32M 字节	8/16 位 *
H'04000000 ~ H'0BFFFFFF	保留			
H'0C000000 ~ H'0DFFFFFF	CS3 空间	通常空间 带字节选择的 SRAM SDRAM	32M 字节	8/16 位 *
H'0E000000 ~ H'1BFFFFFF	保留			
H'1C000000 ~ H'1DFFFFFF	CS7 空间	通常空间 带字节选择的 SRAM	32M 字节	8/16 位 *
H'1E000000 ~ H'FFF7FFFF	保留			
H'FFF80000 ~ H'FFF9FFFF	用于设定 SDRAM 模式			
H'FFFA0000 ~ H'FFFF3FFF	保留			
H'FFFF4000 ~ H'FFFFBFFF	内部 RAM		32K 字节	32 位
H'FFFC0000 ~ H'FFFFFFF	内部外围模块		16K 字节	8/16 位

【注】 不能存取保留空间，否则就不能保证运行。在单芯片模式中，不能使用内部 ROM、内部 RAM 和内部外围模块以外的空间。

\* 通过寄存器进行选择

表 9.5 地址映像 (SH7083 (闪存 512KB 版)、内部 ROM 无效模式)

地址	空间	存储器种类	容量	总线宽度
H'00000000 ~ H'01FFFFFF	CS0 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步) 突发 ROM (同步)	32M 字节	8/16 位 *1
H'02000000 ~ H'0BFFFFFF	保留			
H'0C000000 ~ H'0DFFFFFF	CS3 空间	通常空间 带字节选择的 SRAM SDRAM	32M 字节	8/16 位 *2
H'0E000000 ~ H'1BFFFFFF	保留			
H'1C000000 ~ H'1DFFFFFF	CS7 空间	通常空间 带字节选择的 SRAM	32M 字节	8/16 位 *2
H'1E000000 ~ H'FFF7FFFF	保留			
H'FFF80000 ~ H'FFF9FFFF	用于设定 SDRAM 模式			
H'FFFA0000 ~ H'FFFF3FFF	保留			
H'FFFF4000 ~ H'FFFFBFFF	内部 RAM		32K 字节	32 位
H'FFFC0000 ~ H'FFFFFFF	内部外围模块		16K 字节	8/16 位

【注】 不能存取保留空间，否则就不能保证运行。

\*1 通过模式引脚进行选择

\*2 通过寄存器进行选择

表 9.6 地址映像 (SH7084 (闪存 256KB 版)、内部 ROM 有效模式)

地址	空间	存储器种类	容量	总线宽度
H'00000000 ~ H'0003FFFF	内部 ROM		256K 字节	32 位
H'00040000 ~ H'01FFFFFF	保留			
H'02000000 ~ H'03FFFFFF	CS0 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步) 突发 ROM (同步)	32M 字节	8/16 位 *
H'04000000 ~ H'07FFFFFF	CS1 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16 位 *
H'08000000 ~ H'0BFFFFFF	CS2 空间	通常空间 带字节选择的 SRAM SDRAM	64M 字节	8/16 位 *
H'0C000000 ~ H'0FFFFFFF	CS3 空间	通常空间 带字节选择的 SRAM SDRAM	64M 字节	8/16 位 *
H'10000000 ~ H'13FFFFFF	CS4 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步)	64M 字节	8/16 位 *
H'14000000 ~ H'17FFFFFF	CS5 空间	通常空间 带字节选择的 SRAM MPX-I/O	64M 字节	8/16 位 *
H'18000000 ~ H'1BFFFFFF	CS6 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16 位 *
H'1C000000 ~ H'1FFFFFFF	CS7 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16 位 *
H'20000000 ~ H'FFF7FFFF	保留			
H'FFF80000 ~ H'FFF9FFFF	用于设定 SDRAM 模式			
H'FFFA0000 ~ H'FFFF7FFF	保留			
H'FFFF8000 ~ H'FFFFBFFF	内部 RAM		16K 字节	32 位
H'FFFFC000 ~ H'FFFFFFF	内部外围模块		16K 字节	8/16 位

【注】 不能存取保留空间，否则就不能保证运行。在单芯片模式中，不能使用内部 ROM、内部 RAM 和内部外围模块以外的空间。

\* 通过寄存器进行选择

表 9.7 地址映像 (SH7084 (闪存 256KB 版)、内部 ROM 无效模式)

地址	空间	存储器种类	容量	总线宽度
H'00000000 ~ H'03FFFFFF	CS0 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步) 突发 ROM (同步)	64M 字节	8/16 位 *1
H'04000000 ~ H'07FFFFFF	CS1 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16 位 *2
H'08000000 ~ H'0BFFFFFF	CS2 空间	通常空间 带字节选择的 SRAM SDRAM	64M 字节	8/16 位 *2
H'0C000000 ~ H'0FFFFFFF	CS3 空间	通常空间 带字节选择的 SRAM SDRAM	64M 字节	8/16 位 *2
H'10000000 ~ H'13FFFFFF	CS4 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步)	64M 字节	8/16 位 *2
H'14000000 ~ H'17FFFFFF	CS5 空间	通常空间 带字节选择的 SRAM MPX-I/O	64M 字节	8/16 位 *2
H'18000000 ~ H'1BFFFFFF	CS6 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16 位 *2
H'1C000000 ~ H'1FFFFFFF	CS7 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16 位 *2
H'20000000 ~ H'FFF7FFFF	保留			
H'FFF80000 ~ H'FFF9FFFF	用于设定 SDRAM 模式			
H'FFFA0000 ~ H'FFF7FFFF	保留			
H'FFFF8000 ~ H'FFFFBFFF	内部 RAM		16K 字节	32 位
H'FFFFC000 ~ H'FFFFFFF	内部外围模块		16K 字节	8/16 位

【注】 不能存取保留空间，否则就不能保证运行。

\*1 通过模式引脚进行选择

\*2 通过寄存器进行选择

表 9.8 地址映像 (SH7084 (闪存 512KB 版)、内部 ROM 有效模式)

地址	空间	存储器种类	容量	总线宽度
H'00000000 ~ H'0007FFFF	内部 ROM		512K 字节	32 位
H'00080000 ~ H'01FFFFFF	保留			
H'02000000 ~ H'03FFFFFF	CS0 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步) 突发 ROM (同步)	32M 字节	8/16 位 *
H'04000000 ~ H'07FFFFFF	CS1 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16 位 *
H'08000000 ~ H'0BFFFFFF	CS2 空间	通常空间 带字节选择的 SRAM SDRAM	64M 字节	8/16 位 *
H'0C000000 ~ H'0FFFFFFF	CS3 空间	通常空间 带字节选择的 SRAM SDRAM	64M 字节	8/16 位 *
H'10000000 ~ H'13FFFFFF	CS4 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步)	64M 字节	8/16 位 *
H'14000000 ~ H'17FFFFFF	CS5 空间	通常空间 带字节选择的 SRAM MPX-I/O	64M 字节	8/16 位 *
H'18000000 ~ H'1BFFFFFF	CS6 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16 位 *
H'1C000000 ~ H'1FFFFFFF	CS7 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16 位 *
H'20000000 ~ H'FFF7FFFF	保留			
H'FFF80000 ~ H'FFF9FFFF	用于设定 SDRAM 模式			
H'FFFA0000 ~ H'FFFF3FFF	保留			
H'FFFF4000 ~ H'FFFFBFFF	内部 RAM		32K 字节	32 位
H'FFFFC000 ~ H'FFFFFFF	内部外围模块		16K 字节	8/16 位

【注】 不能存取保留空间，否则就不能保证运行。在单芯片模式中，不能使用内部 ROM、内部 RAM 和内部外围模块以外的空间。

\* 通过寄存器进行选择

表 9.9 地址映像 (SH7084 (闪存 512KB 版)、内部 ROM 无效模式)

地址	空间	存储器种类	容量	总线宽度
H'00000000 ~ H'03FFFFFF	CS0 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步) 突发 ROM (同步)	64M 字节	8/16 位 *1
H'04000000 ~ H'07FFFFFF	CS1 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16 位 *2
H'08000000 ~ H'0BFFFFFF	CS2 空间	通常空间 带字节选择的 SRAM SDRAM	64M 字节	8/16 位 *2
H'0C000000 ~ H'0FFFFFFF	CS3 空间	通常空间 带字节选择的 SRAM SDRAM	64M 字节	8/16 位 *2
H'10000000 ~ H'13FFFFFF	CS4 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步)	64M 字节	8/16 位 *2
H'14000000 ~ H'17FFFFFF	CS5 空间	通常空间 带字节选择的 SRAM MPX-I/O	64M 字节	8/16 位 *2
H'18000000 ~ H'1BFFFFFF	CS6 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16 位 *2
H'1C000000 ~ H'1FFFFFFF	CS7 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16 位 *2
H'20000000 ~ H'FFF7FFFF	保留			
H'FFF80000 ~ H'FFF9FFFF	用于设定 SDRAM 模式			
H'FFFA0000 ~ H'FFF3FFF	保留			
H'FFFF4000 ~ H'FFFFBFFF	内部 RAM		32K 字节	32 位
H'FFFFC000 ~ H'FFFFFFF	内部外围模块		16K 字节	8/16 位

【注】 不能存取保留空间，否则就不能保证运行。

\*1 通过模式引脚进行选择

\*2 通过寄存器进行选择

表 9.10 地址映像 (SH7085 (闪存 256KB 版)、内部 ROM 有效模式)

地址	空间	存储器种类	容量	总线宽度
H'00000000 ~ H'0003FFFF	内部 ROM		256K 字节	32 位
H'00040000 ~ H'01FFFFFF	保留			
H'02000000 ~ H'03FFFFFF	CS0 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步) 突发 ROM (同步)	32M 字节	8/16/32 位 *
H'04000000 ~ H'07FFFFFF	CS1 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16/32 位 *
H'08000000 ~ H'0BFFFFFF	CS2 空间	通常空间 带字节选择的 SRAM SDRAM	64M 字节	8/16/32 位 *
H'0C000000 ~ H'0FFFFFFF	CS3 空间	通常空间 带字节选择的 SRAM SDRAM	64M 字节	8/16/32 位 *
H'10000000 ~ H'13FFFFFF	CS4 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步)	64M 字节	8/16/32 位 *
H'14000000 ~ H'17FFFFFF	CS5 空间	通常空间 带字节选择的 SRAM PCMCIA MPX-I/O	64M 字节	8/16/32 位 *
H'18000000 ~ H'1BFFFFFF	CS6 空间	通常空间 带字节选择的 SRAM PCMCIA 突发 MPX-I/O	64M 字节	8/16/32 位 *
H'1C000000 ~ H'1FFFFFFF	CS7 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16/32 位 *
H'20000000 ~ H'FFF7FFFF	保留			
H'FFF80000 ~ H'FFF9FFFF	用于设定 SDRAM 模式			
H'FFFA0000 ~ H'FFFF7FFF	保留			
H'FFFF8000 ~ H'FFFFBFFF	内部 RAM		16K 字节	32 位
H'FFFFC000 ~ H'FFFFFFF	内部外围模块		16K 字节	8/16 位

【注】 不能存取保留空间，否则就不能保证运行。在单芯片模式中，不能使用内部 ROM、内部 RAM 和内部外围模块以外的空间。

\* 通过寄存器进行选择



表 9.11 地址映像 (SH7085 (闪存 256KB 版)、内部 ROM 无效模式)

地址	空间	存储器种类	容量	总线宽度
H'00000000 ~ H'03FFFFFF	CS0 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步) 突发 ROM (同步)	64M 字节	16/32 位 *1
H'04000000 ~ H'07FFFFFF	CS1 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16/32 位 *2
H'08000000 ~ H'0BFFFFFF	CS2 空间	通常空间 带字节选择的 SRAM SDRAM	64M 字节	8/16/32 位 *2
H'0C000000 ~ H'0FFFFFFF	CS3 空间	通常空间 带字节选择的 SRAM SDRAM	64M 字节	8/16/32 位 *2
H'10000000 ~ H'13FFFFFF	CS4 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步)	64M 字节	8/16/32 位 *2
H'14000000 ~ H'17FFFFFF	CS5 空间	通常空间 带字节选择的 SRAM PCMCIA MPX-I/O	64M 字节	8/16/32 位 *2
H'18000000 ~ H'1BFFFFFF	CS6 空间	通常空间 带字节选择的 SRAM PCMCIA 突发 MPX-I/O	64M 字节	8/16/32 位 *2
H'1C000000 ~ H'1FFFFFFF	CS7 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16/32 位 *2
H'20000000 ~ H'FFF7FFFF	保留			
H'FFF80000 ~ H'FFF9FFFF	用于 SDRAM 模式设定			
H'FFFA0000 ~ H'FFFF7FFF	保留			
H'FFFF8000 ~ H'FFFFBFFF	内部 RAM		16K 字节	32 位
H'FFFFC000 ~ H'FFFFFFF	内部外围模块		16K 字节	8/16 位

【注】 不能存取保留空间，否则就不能保证运行。

\*1 通过模式引脚进行选择

\*2 通过寄存器进行选择

表 9.12 地址映像 (SH7085 (闪存 512KB 版)、内部 ROM 有效模式)

地址	空间	存储器种类	容量	总线宽度
H'00000000 ~ H'0007FFFF	内部 ROM		512K 字节	32 位
H'00080000 ~ H'01FFFFFF	保留			
H'02000000 ~ H'03FFFFFF	CS0 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步) 突发 ROM (同步)	32M 字节	8/16/32 位 *
H'04000000 ~ H'07FFFFFF	CS1 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16/32 位 *
H'08000000 ~ H'0BFFFFFF	CS2 空间	通常空间 带字节选择的 SRAM SDRAM	64M 字节	8/16/32 位 *
H'0C000000 ~ H'0FFFFFFF	CS3 空间	通常空间 带字节选择的 SRAM SDRAM	64M 字节	8/16/32 位 *
H'10000000 ~ H'13FFFFFF	CS4 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步)	64M 字节	8/16/32 位 *
H'14000000 ~ H'17FFFFFF	CS5 空间	通常空间 带字节选择的 SRAM PCMCIA MPX-I/O	64M 字节	8/16/32 位 *
H'18000000 ~ H'1BFFFFFF	CS6 空间	通常空间 带字节选择的 SRAM PCMCIA 突发 MPX-I/O	64M 字节	8/16/32 位 *
H'1C000000 ~ H'1FFFFFFF	CS7 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16/32 位 *
H'20000000 ~ H'FFF7FFFF	保留			
H'FFF80000 ~ H'FFF9FFFF	用于 SDRAM 模式设定			
H'FFFA0000 ~ H'FFFF3FFF	保留			
H'FFFF4000 ~ H'FFFFBFFF	内部 RAM		32K 字节	32 位
H'FFFFC000 ~ H'FFFFFFF	内部外围模块		16K 字节	8/16 位

【注】 不能存取保留空间，否则就不能保证运行。在单芯片模式中，不能使用内部 ROM、内部 RAM 和内部外围模块以外的空间。

\* 通过寄存器进行选择

表 9.13 地址映像 (SH7085 (闪存 512KB 版)、内部 ROM 无效模式)

地址	空间	存储器种类	容量	总线宽度
H'00000000 ~ H'03FFFFFF	CS0 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步) 突发 ROM (同步)	64M 字节	16/32 位 *1
H'04000000 ~ H'07FFFFFF	CS1 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16/32 位 *2
H'08000000 ~ H'0BFFFFFF	CS2 空间	通常空间 带字节选择的 SRAM SDRAM	64M 字节	8/16/32 位 *2
H'0C000000 ~ H'0FFFFFFF	CS3 空间	通常空间 带字节选择的 SRAM SDRAM	64M 字节	8/16/32 位 *2
H'10000000 ~ H'13FFFFFF	CS4 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步)	64M 字节	8/16/32 位 *2
H'14000000 ~ H'17FFFFFF	CS5 空间	通常空间 带字节选择的 SRAM PCMCIA MPX-I/O	64M 字节	8/16/32 位 *2
H'18000000 ~ H'1BFFFFFF	CS6 空间	通常空间 带字节选择的 SRAM PCMCIA 突发 MPX-I/O	64M 字节	8/16/32 位 *2
H'1C000000 ~ H'1FFFFFFF	CS7 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16/32 位 *2
H'20000000 ~ H'FFF7FFFF	保留			
H'FFF80000 ~ H'FFF9FFFF	用于 SDRAM 模式设定			
H'FFFA0000 ~ H'FFFF3FFF	保留			
H'FFFF4000 ~ H'FFFFBFFF	内部 RAM		32K 字节	32 位
H'FFFC0000 ~ H'FFFFFFF	内部外围模块		16K 字节	8/16 位

【注】 不能存取保留空间，否则就不能保证运行。

\*1 通过模式引脚进行选择

\*2 通过寄存器进行选择

表 9.14 地址映像 (SH7086、内部 ROM 有效模式)

地址	空间	存储器种类	容量	总线宽度
H'00000000 ~ H'0007FFFF	内部 ROM		512K 字节	32 位
H'00080000 ~ H'01FFFFFF	保留			
H'02000000 ~ H'03FFFFFF	CS0 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步) 突发 ROM (同步)	32M 字节	8/16/32 位 *
H'04000000 ~ H'07FFFFFF	CS1 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16/32 位 *
H'08000000 ~ H'0BFFFFFF	CS2 空间	通常空间 带字节选择的 SRAM SDRAM	64M 字节	8/16/32 位 *
H'0C000000 ~ H'0FFFFFFF	CS3 空间	通常空间 带字节选择的 SRAM SDRAM	64M 字节	8/16/32 位 *
H'10000000 ~ H'13FFFFFF	CS4 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步)	64M 字节	8/16/32 位 *
H'14000000 ~ H'17FFFFFF	CS5 空间	通常空间 带字节选择的 SRAM PCMCIA MPX-I/O	64M 字节	8/16/32 位 *
H'18000000 ~ H'1BFFFFFF	CS6 空间	通常空间 带字节选择的 SRAM PCMCIA 突发 MPX-I/O	64M 字节	8/16/32 位 *
H'1C000000 ~ H'1FFFFFFF	CS7 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16/32 位 *
H'20000000 ~ H'3FFFFFFF	保留			
H'40000000 ~ H'7FFFFFFF	CS8 空间	通常空间 带字节选择的 SRAM	1G 字节	8/16/32 位 *
H'80000000 ~ H'FFF7FFFF	保留			
H'FFF80000 ~ H'FFF9FFFF	用于 SDRAM 模式设定			
H'FFFA0000 ~ H'FFF3FFFF	保留			
H'FFF40000 ~ H'FFFBFFFF	内部 RAM		32K 字节	32 位
H'FFFC0000 ~ H'FFFFFFF	内部外围模块		16K 字节	8/16 位

【注】 不能存取保留空间，否则就不能保证运行。在单芯片模式中，不能使用内部 ROM、内部 RAM 和内部外围模块以外的空间。

\* 通过寄存器进行选择

表 9.15 地址映像 (SH7086、内部 ROM 无效模式)

地址	空间	存储器种类	容量	总线宽度
H'00000000 ~ H'03FFFFFF	CS0 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步) 突发 ROM (同步)	64M 字节	16/32 位 *1
H'04000000 ~ H'07FFFFFF	CS1 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16/32 位 *2
H'08000000 ~ H'0BFFFFFF	CS2 空间	通常空间 带字节选择的 SRAM SDRAM	64M 字节	8/16/32 位 *2
H'0C000000 ~ H'0FFFFFFF	CS3 空间	通常空间 带字节选择的 SRAM SDRAM	64M 字节	8/16/32 位 *2
H'10000000 ~ H'13FFFFFF	CS4 空间	通常空间 带字节选择的 SRAM 突发 ROM (异步)	64M 字节	8/16/32 位 *2
H'14000000 ~ H'17FFFFFF	CS5 空间	通常空间 带字节选择的 SRAM PCMCIA MPX-I/O	64M 字节	8/16/32 位 *2
H'18000000 ~ H'1BFFFFFF	CS6 空间	通常空间 带字节选择的 SRAM PCMCIA 突发 MPX-I/O	64M 字节	8/16/32 位 *2
H'1C000000 ~ H'1FFFFFFF	CS7 空间	通常空间 带字节选择的 SRAM	64M 字节	8/16/32 位 *2
H'20000000 ~ H'3FFFFFFF	保留			
H'40000000 ~ H'7FFFFFFF	CS8 空间	通常空间 带字节选择的 SRAM	1G 字节	8/16/32 位 *2
H'80000000 ~ H'FFF7FFFF	保留			
H'FFF80000 ~ H'FFF9FFFF	用于设定 SDRAM 模式			
H'FFFA0000 ~ H'FFFF3FFF	保留			
H'FFFF4000 ~ H'FFFFBFFF	内部 RAM		32K 字节	32 位
H'FFFC0000 ~ H'FFFFFFF	内部外围模块		16K 字节	8/16 位

【注】 不能存取保留空间，否则就不能保证运行。

\*1 通过模式引脚进行选择

\*2 通过寄存器进行选择

## 9.4 寄存器说明

BSC 有以下寄存器。有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 27 章 寄存器一览”。在结束存储器的接口设定前，不能存取 CS0 以外的空间。

表 9.16 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
通用控制寄存器	CMNCR	R/W	H'00001010	H'FFFFFF00	32
CS0 空间总线控制寄存器	CS0BCR	R/W	H'36DB0600	H'FFFFFF04	32
CS1 空间总线控制寄存器	CS1BCR	R/W	H'36DB0600	H'FFFFFF08	32
CS2 空间总线控制寄存器	CS2BCR	R/W	H'36DB0600	H'FFFFFF0C	32
CS3 空间总线控制寄存器	CS3BCR	R/W	H'36DB0600	H'FFFFFF10	32
CS4 空间总线控制寄存器	CS4BCR	R/W	H'36DB0600	H'FFFFFF14	32
CS5 空间总线控制寄存器	CS5BCR	R/W	H'36DB0600	H'FFFFFF18	32
CS6 空间总线控制寄存器	CS6BCR	R/W	H'36DB0600	H'FFFFFF1C	32
CS7 空间总线控制寄存器	CS7BCR	R/W	H'36DB0600	H'FFFFFF20	32
CS8 空间总线控制寄存器	CS8BCR	R/W	H'36DB0600	H'FFFFFF24	32
CS0 空间等待控制寄存器	CS0WCR	R/W	H'00000500	H'FFFFFF28	32
CS1 空间等待控制寄存器	CS1WCR	R/W	H'00000500	H'FFFFFF2C	32
CS2 空间等待控制寄存器	CS2WCR	R/W	H'00000500	H'FFFFFF30	32
CS3 空间等待控制寄存器	CS3WCR	R/W	H'00000500	H'FFFFFF34	32
CS4 空间等待控制寄存器	CS4WCR	R/W	H'00000500	H'FFFFFF38	32
CS5 空间等待控制寄存器	CS5WCR	R/W	H'00000500	H'FFFFFF3C	32
CS6 空间等待控制寄存器	CS6WCR	R/W	H'00000500	H'FFFFFF40	32
CS7 空间等待控制寄存器	CS7WCR	R/W	H'00000500	H'FFFFFF44	32
CS8 空间等待控制寄存器	CS8WCR	R/W	H'00000500	H'FFFFFF48	32
SDRAM 控制寄存器	SDCR	R/W	H'00000000	H'FFFFFF4C	32
刷新定时器的控制 / 状态寄存器	RTCSR	R/W	H'00000000	H'FFFFFF50	32
刷新定时器的计数器	RTCNT	R/W	H'00000000	H'FFFFFF54	32
刷新时间常数寄存器	RTCOR	R/W	H'00000000	H'FFFFFF58	32
总线功能扩展寄存器	BSCEHR	R/W	H'0000	H'FFFFE89A	8、16

## 9.4.1 通用控制寄存器 (CMNCR)

CMNCR 是共同控制各区域的 32 位寄存器。

在结束寄存器的初始设定前，不能存取区域 0 以外的外部存储器。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	DMAIW[1:0]	DMAIWA	-	-	-	HIZMEM	HIZCNT	
初始值:	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
31 ~ 13	—	全 0	R	保留位 读写的值总是 0。
12	—	1	R	保留位 读写的值总是 1。
11 ~ 8	—	全 0	R	保留位 读写的值总是 0。
7、6	DMAIW[1:0]	00	R/W	指定 DMA 单地址传送时的存取周期之间的等待 在 DMA 单地址传送时，此位指定从带 DACK 的外部器件输出数据后要插入的空闲周期数。空闲周期的插入方法因 DMAIWA 位的指定而不同。 00: 不插入空闲周期 01: 插入 1 个空闲周期 10: 插入 2 个空闲周期 11: 插入 4 个空闲周期
5	DMAIWA	0	R/W	指定 DMA 单地址传送时的存取周期之间的等待插入方法 此位指定空闲周期（由 DMAIW1 位和 DMAIW0 位指定）的插入方法。 当此位是 0 时，在带 DACK 的外部器件驱动数据总线后，并且在其他器件（包括本 LSI）驱动数据总线时插入空闲周期。如果带 DACK 的外部器件连续驱动数据总线，就不插入空闲周期。当此位为 1 时，即使连续存取带 DACK 的外部器件，也总是在 1 次存取结束后插入空闲周期。 0: 在带 DACK 外部器件驱动数据总线后，并且在其他器件驱动数据总线时插入空闲周期 1: 在存取带 DACK 的外部器件后，总是插入空闲周期
4	—	1	R	保留位 读写的值总是 1。
3、2	—	全 0	R	保留位 读写的值总是 0。
1	HIZMEM	0	R/W	High-Z 存储器控制此位指定软件待机模式时的 A29 ~ A0、 $\overline{BS}$ 、 $\overline{CSn}$ 、 $\overline{RDWR}$ 、 $\overline{WRxx}$ 、 $\overline{RD}$ 、 $\overline{AH}$ 、 $\overline{FRAME}$ 、 $\overline{ICIORD}$ 、 $\overline{ICIORW}$ 、 $\overline{WE}$ 、 $\overline{CE1A}$ 、 $\overline{CE1B}$ 、 $\overline{CE2A}$ 和 $\overline{CE2B}$ 引脚状态。在释放总线时，与此位无关，这些引脚为高阻抗状态。 0: 在软件待机模式时，引脚为高阻抗状态 1: 在软件待机模式时，引脚为驱动状态

位	位名	初始值	R/W	说 明
0	HIZCNT	0	R/W	High-Z 控制此位指定软件待机模式以及释放总线权时的 CKE、 $\overline{\text{RASU}}$ 、 $\overline{\text{RASL}}$ 、 $\overline{\text{CASU}}$ 、 $\overline{\text{CASL}}$ 引脚状态。 0: 在软件待机模式和释放总线权时, CKE、 $\overline{\text{RASU}}$ 、 $\overline{\text{RASL}}$ 、 $\overline{\text{CASU}}$ 、 $\overline{\text{CASL}}$ 引脚为高阻抗状态 1: 在软件待机模式和释放总线权时, CKE、 $\overline{\text{RASU}}$ 、 $\overline{\text{RASL}}$ 、 $\overline{\text{CASU}}$ 、 $\overline{\text{CASL}}$ 引脚为驱动状态

#### 9.4.2 CSn 空间总线控制寄存器 (CSnBCR) (n=0 ~ 8)

CSnBCR 设定连接各空间的存储器种类、空间的数据总线宽度以及存取周期之间的等待数。

在结束寄存器的初始设定前, 不能存取区域 0 以外的外部存储器。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	IWW[1:0]	-	IWRWD[1:0]	-	IWRWS[1:0]	-	IWRRD[1:0]	-	IWRRS[1:0]	-	-	-	-	-
初始值:	0	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1
R/W:	R	R	R/W	R/W	R	R/W	R/W	R	R/W	R/W	R	R/W	R/W	R	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	TYPE[2:0]	-	BSZ[1:0]	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	1*	1*	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R	R	R	R	R	R	R	R	R

【注】\* 如果内部ROM无效, 就在上电复位时, CS0BCR采样指定总线宽度的外部引脚 (MD0和MD1) 的值。

位	位名	初始值	R/W	说 明
31、30	—	全 0	R	保留位 读写的值总是 0。
29、28	IWW[1:0]	11	R/W	指定写 - 读 / 写 - 写周期之间的空闲 此位指定在存取已连接空间的存储器后要插入的空闲周期数。 写 - 读周期和写 - 写周期为对象周期。 00: 不插入空闲周期 01: 插入 1 个空闲周期 10: 插入 2 个空闲周期 11: 插入 4 个空闲周期
27	—	0	R	保留位 读写的值总是 0。
26、25	IWRWD[1:0]	11	R/W	指定不同空间的读 - 写周期之间的空闲 此位指定在存取已连接空间的存储器后要插入的空闲周期数。对象周期是在不同空间的连续读 - 写周期。 00: 不插入空闲周期 01: 插入 1 个空闲周期 10: 插入 2 个空闲周期 11: 插入 4 个空闲周期
24	—	0	R	保留位 读写的值总是 0。



位	位名	初始值	R/W	说 明
23、22	IWRWS[1:0]	11	R/W	指定同一空间的读 - 写周期之间的空闲 此位指定在存取已连接空间的存储器后要插入的空闲周期数。对象周期是在同一空间的连续读 - 写周期。 00: 没不插入空闲周期 01: 插入 1 个空闲周期 10: 插入 2 个空闲周期 11: 插入 4 个空闲周期
21	—	0	R	保留位 读写的值总是 0。
20、19	IWRRD[1:0]	11	R/W	指定不同空间的读 - 读周期之间的空闲 此位指定在存取已连接空间的存储器后要插入的空闲周期数。对象周期是在不同空间的连续读 - 读周期。 00: 不插入空闲周期 01: 插入 1 个空闲周期 10: 插入 2 个空闲周期 11: 插入 4 个空闲周期
18	—	0	R	保留位 读写的值总是 0。
17、16	IWRRS[1:0]	11	R/W	指定同一空间的读 - 读周期之间的空闲 此位指定在存取已连接空间的存储器后要插入的空闲周期数。对象周期是在同一空间的连续读 - 读周期。 00: 不插入空闲周期 01: 插入 1 个空闲周期 10: 插入 2 个空闲周期 11: 插入 4 个空闲周期
15	—	0	R	保留位 读写的值总是 0。
14 ~ 12	TYPE[2:0]	000	R/W	存储器种类的指定 此位设定要连接空间的存储器种类。 000: 通常空间 001: 突发 ROM (异步) 010: MPX-I/O 011: 带字节选择的 SRAM 100: SDRAM 101: PCMCIA 110: 突发 MPX-I/O 111: 突发 ROM (时钟同步) 各区域的存储器种类请参照表 9.2 ~ 表 9.15。 <b>【注】</b> 1. 如果将区域 6 设定为突发 MPX-I/O, 就不能将区域 2 和区域 3 设定为 SDRAM 空间。 2. 只有在区域 2 和区域 3 才能设定 SDRAM。只在一个区域上连接 SDRAM 时, 必须让区域 3 作为可设定 SDRAM 的区域。这时, 区域 2 必须作为通常空间或带字节选择的 SRAM。
11	—	0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说 明
10、9	BSZ[1:0]	11*	R/W	<p>数据总线宽度的指定 此位指定空间的数据总线宽度。 00: 保留 (不能设定) 01: 8 位 10: 16 位 11: 32 位 在 MPX-I/O 的情况下, 通过地址选择总线宽度。</p> <p><b>【注】</b> 1. 当区域 5 设定为 MPX-I/O 时, 如果将此位设定为 11, 就根据 CS5WCR 的 SZSEL 位指定的地址选择总线宽度 (8 位或者 16 位)。 2. 当内部 ROM 无效时, 通过外部输入引脚设定区域 0 的数据总线宽度。忽视 CS0BCR 的 BSZ1、0 位的设定。 3. 当区域 6 设定为突发 MPX-I/O 时, 只能将总线宽度设定为 32 位。 4. 当区域 5 或者区域 6 设定为 PCMCIA 空间时, 能将总线宽度设定为 8 或者 16 位。 5. 当区域 2 或者区域 3 设定为 SDRAM 空间时, 能将总线宽度设定为 16 或者 32 位。</p>
8 ~ 0	—	全 0	R	保留位 读写的值总是 0。

**【注】** \* 在内部 ROM 无效的情况下, CS0BCR 在上电复位时对指定总线宽度的外部引脚 (MD0 和 MD1) 的值进行采样。

### 9.4.3 CSn 空间等待控制寄存器 (CSnWCR) (n=0 ~ 8)

CSnWCR 设定有关存储器存取的各种等待周期。此寄存器的位结构因 CSn 空间总线控制寄存器 (CSnBCR) 设定的存储器种类 (TYPE2、1、0) 而发生以下的变化。必须在存取对象区域前并且在设定 CSnBCR 寄存器后设定 CSnWCR 寄存器。

#### (1) 通常空间、带字节选择的 SRAM

- CS0WCR、CS1WCR、CS2WCR、CS3WCR、CS4WCR、CS5WCR、CS6WCR、CS7WCR、CS8WCR

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	BAS	-	WW[2:0]		
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SW[1:0]		WR[3:0]			WM	-	-	-	-	HW[1:0]		
初始值:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
31 ~ 21	—	全 0	R	保留位 读写的值总是 0。
20	BAS	0	R/W	带字节选择的 SRAM 时的字节存取选择 此位设定带字节选择的 SRAM 接口时的 $\overline{WRxx}$ 和 RDWR 信号时序。 0: $\overline{WRxx}$ 在读写时有效, RDWR 在写存取周期中有效 1: $\overline{WRxx}$ 在读写存取周期中有效, RDWR 在写时有效
19	—	0	R	保留位 读写的值总是 0。
18 ~ 16	WW[2:0]	000	R/W	写存取的等待周期数 此位指定写存取所需的周期数。 000: 与 WR3 ~ WR0 设定 (读存取等待) 的周期相同 001: 0 个周期 010: 1 个周期 011: 2 个周期 100: 3 个周期 101: 4 个周期 110: 5 个周期 111: 6 个周期
15 ~ 13	—	全 0	R	保留位 读写的值总是 0。
12、11	SW[1:0]	00	R/W	地址和 $\overline{CSn}$ 有效 $\rightarrow \overline{RD}$ 和 $\overline{WRxx}$ 有效的延迟周期数 此位指定从地址和 $\overline{CSn}$ 有效到 $\overline{RD}$ 和 $\overline{WRxx}$ 有效的延迟周期数。 00: 0.5 个周期 01: 1.5 个周期 10: 2.5 个周期 11: 3.5 个周期

位	位名	初始值	R/W	说 明
10 ~ 7	WR[3:0]	1010	R/W	<p>读存储的等待周期数 此位指定读存储所需的等待周期数。</p> <p>0000: 0 个周期 0001: 1 个周期 0010: 2 个周期 0011: 3 个周期 0100: 4 个周期 0101: 5 个周期 0110: 6 个周期 0111: 8 个周期 1000: 10 个周期 1001: 12 个周期 1010: 14 个周期 1011: 18 个周期 1100: 24 个周期 1101: 保留 (不能设定) 1110: 保留 (不能设定) 1111: 保留 (不能设定)</p>
6	WM	0	R/W	<p>外部等待屏蔽的指定 此位指定是将外部等待输入设定为有效还是忽视外部等待输入。即使存取等待周期数是 0，此位的设定也有效</p> <p>0: 外部等待输入有效 1: 忽视外部等待输入</p>
5 ~ 2	—	全 0	R	<p>保留位 读写的值总是 0。</p>
1、0	HW[1:0]	00	R/W	<p><math>\overline{RD}</math> 和 <math>\overline{WRxx}</math> 无效 → 地址和 <math>\overline{CSn}</math> 无效的延迟周期数 此位指定从 <math>\overline{RD}</math> 和 <math>\overline{WRxx}</math> 无效到地址和 <math>\overline{CSn}</math> 无效的延迟周期数。</p> <p>00: 0.5 个周期 01: 1.5 个周期 10: 2.5 个周期 11: 3.5 个周期</p>

(2) MPX-I/O

• CS5WCR

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	SZSEL	MPXW	-	WW[2:0]		
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SW[1:0]		WR[3:0]				WM	-	-	-	-	HW[1:0]	
初始值:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

位	位名	初始值	R/W	说 明																				
31 ~ 22	—	全 0	R	保留位 读写的值总是 0。																				
21	SZSEL	0	R/W	MPX-I/O 接口总线宽度的指定位 此位指定要选择总线宽度（将 CS5BCR 的 BSZ1 和 BSZ0 设定为 11 时）的地址。此设定只在将区域 5 设定为 MPX-I/O 时有效。 0: 由地址 A14 选择总线宽度 1: 由地址 A21 选择总线宽度 通过 SZSEL 位和 A14、A21 选择总线宽度的关系如下所示: <table border="1" style="margin: 10px auto;"> <thead> <tr> <th>SZSEL</th> <th>A14</th> <th>A21</th> <th>说 明</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>没有影响</td> <td>8 位</td> </tr> <tr> <td>0</td> <td>1</td> <td>没有影响</td> <td>16 位</td> </tr> <tr> <td>1</td> <td>没有影响</td> <td>0</td> <td>8 位</td> </tr> <tr> <td>1</td> <td>没有影响</td> <td>1</td> <td>16 位</td> </tr> </tbody> </table>	SZSEL	A14	A21	说 明	0	0	没有影响	8 位	0	1	没有影响	16 位	1	没有影响	0	8 位	1	没有影响	1	16 位
SZSEL	A14	A21	说 明																					
0	0	没有影响	8 位																					
0	1	没有影响	16 位																					
1	没有影响	0	8 位																					
1	没有影响	1	16 位																					
20	MPXW	0	R/W	MPX-I/O 接口地址的等待 此位设定是否插入 MPX-I/O 接口地址周期的等待。本设定只在将区域 5 设定为 MPX-I/O 时有效。 0: 不插入等待 1: 插入 1 个周期等待																				
19	—	0	R	保留位 读写的值总是 0。																				
18 ~ 16	WW[2:0]	000	R/W	写存取等待周期数 此位指定写存取所需的周期数。 000: 与 WR3 ~ WR0 设定的（读存取等待）周期相同 001: 0 个周期 010: 1 个周期 011: 2 个周期 100: 3 个周期 101: 4 个周期 110: 5 个周期 111: 6 个周期																				

位	位名	初始值	R/W	说明
15 ~ 13	—	全 0	R	保留位 读写的值总是 0。
12、11	SW[1:0]	00	R/W	地址和 $\overline{CSn}$ 有效 $\rightarrow \overline{RD}$ 和 $\overline{WRxx}$ 有效的延迟周期数 此位指定从地址和 $\overline{CSn}$ 有效到 $\overline{RD}$ 和 $\overline{WRxx}$ 有效的延迟周期数。 00: 0.5 个周期 01: 1.5 个周期 10: 2.5 个周期 11: 3.5 个周期
10 ~ 7	WR[3:0]	1010	R/W	读存取的等待周期数 此位指定读存取所需的等待周期数。 0000: 0 个周期 0001: 1 个周期 0010: 2 个周期 0011: 3 个周期 0100: 4 个周期 0101: 5 个周期 0110: 6 个周期 0111: 8 个周期 1000: 10 个周期 1001: 12 个周期 1010: 14 个周期 1011: 18 个周期 1100: 24 个周期 1101: 保留 (不能设定) 1110: 保留 (不能设定) 1111: 保留 (不能设定)
6	WM	0	R/W	外部等待屏蔽的指定 此位指定是将外部等待输入设定为有效还是忽视外部等待输入。即使存取等待周期数是 0, 此位的设定也有效 0: 外部等待输入有效 1: 忽视外部等待输入
5 ~ 2	—	全 0	R	保留位 读写的值总是 0。
1、0	HW[1:0]	00	R/W	$\overline{RD}$ 和 $\overline{WRxx}$ 无效 $\rightarrow$ 地址和 $\overline{CSn}$ 无效的延迟周期数 此位指定从 $\overline{RD}$ 和 $\overline{WRxx}$ 无效到地址和 $\overline{CSn}$ 无效的延迟周期数。 00: 0.5 个周期 01: 1.5 个周期 10: 2.5 个周期 11: 3.5 个周期

(3) 突发 ROM (异步)

• CS0WCR、CS4WCR

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	BEN	-	-	BW[1:0]	
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SW[1:0]		W[3:0]				WM	-	-	-	-	HW[1:0]	
初始值:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
31 ~ 21	—	全 0	R	保留位 读写的值总是 0。
20	BEN	0	R/W	突发允许的指定 此位指定是否允许在发生 16 字节存取时, 对 16 位总线进行 8 突发存取以及对 8 位总线进行 16 突发存取。如果将此位设定为 0, 就对 16 位总线进行 4 次 2 突发存取或者对 8 位总线进行 4 次 4 突发存取。 如果使用不支持 8 突发和 16 突发存取的器件, 就必须将此位设定为 1。 0: 允许对 16 位总线进行 8 突发存取 以及对 8 位总线进行 16 突发存取 1: 禁止对 16 位总线进行 8 突发存取 以及 8 位总线进行 16 突发存取
19、18	—	全 0	R	保留位 读写的值总是 0。
17、16	BW[1:0]	00	R/W	突发的等待周期数 此位指定要插入到突发存取时的第 2 次以后存取周期的等待周期数。 00: 0 个周期 01: 1 个周期 10: 2 个周期 11: 3 个周期
15 ~ 13	—	全 0	R	保留位 读写的值总是 0。
12、11	SW[1:0]	00	R/W	地址和 $\overline{CSn}$ 有效 $\rightarrow \overline{RD}$ 和 $\overline{WRxx}$ 有效的延迟周期数 此位指定从地址和 $\overline{CSn}$ 有效到 $\overline{RD}$ 和 $\overline{WRxx}$ 有效的延迟周期数。 00: 0.5 个周期 01: 1.5 个周期 10: 2.5 个周期 11: 3.5 个周期

位	位名	初始值	R/W	说 明
10 ~ 7	W[3:0]	1010	R/W	<p>存取等待周期数</p> <p>此位指定要插入到第 1 次存取周期的等待周期数。</p> <p>0000: 0 个周期  0001: 1 个周期  0010: 2 个周期  0011: 3 个周期  0100: 4 个周期  0101: 5 个周期  0110: 6 个周期  0111: 8 个周期  1000: 10 个周期  1001: 12 个周期  1010: 14 个周期  1011: 18 个周期  1100: 24 个周期  1101: 保留 (不能设定)  1110: 保留 (不能设定)  1111: 保留 (不能设定)</p>
6	WM	0	R/W	<p>外部等待屏蔽的指定</p> <p>此位指定是将外部等待输入设定为有效还是忽视外部等待输入。即使存取等待周期数是 0, 此位设定也有效。</p> <p>0: 外部等待输入有效  1: 忽视外部等待输入</p>
5 ~ 2	—	全 0	R	<p>保留位</p> <p>读写的值总是 0。</p>
1、0	HW[1:0]	00	R/W	<p><math>\overline{RD}</math> 和 <math>\overline{WRxx}</math> 无效 → 地址和 <math>\overline{CSn}</math> 无效的延迟周期数</p> <p>此位指定从 <math>\overline{RD}</math> 和 <math>\overline{WRxx}</math> 无效到地址和 <math>\overline{CSn}</math> 无效的延迟周期数。</p> <p>00: 0.5 个周期  01: 1.5 个周期  10: 2.5 个周期  11: 3.5 个周期</p>



## (4) SDRAM

当区域 2 和区域 3 都被设定为 SDRAM 时，WTRP1/0 位、WTRCD0/1 位、TRWL1/0 位和 WTRC1/0 位的设定是通用的。当只有 1 个区域连接 SDRAM 时，必须将区域 3 设定为 SDRAM，而将区域 2 设定为通常空间或者带字节选择的 SRAM。

## • CS2WCR

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	A2CL[1:0]	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
31 ~ 11	—	全 0	R	保留位 读写的值总是 0。
10	—	1	R	保留位 读写的值总是 1。
9	—	0	R	保留位 读写的值总是 0。
8、7	A2CL[1:0]	10	R/W	区域 2 的 CAS 潜伏时间 此位指定区域 2 的 CAS 潜伏时间。 00: 1 个周期 01: 2 个周期 10: 3 个周期 11: 4 个周期
6 ~ 0	—	全 0	R	保留位 读写的值总是 0。

• CS3WCR

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	WTRP[1:0]	-	WTRCD[1:0]	-	A3CL[1:0]	-	-	-	TRWL[1:0]	-	WTRC[1:0]	-	-	-	-
初始值:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R	R/W	R/W	R	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W

位	位名	初始值	R/W	说 明
31 ~ 15	—	全 0	R	保留位 读写的值总是 0。
14、13	WTRP[1:0]	00	R/W	预充电结束的等待周期数 在以下情况下，此位指定为了等待预充电结束而要插入的最小等待周期数： <ul style="list-style-type: none"> <li>• 从启动自动预充电到对同一存储体产生 ACTV 命令</li> <li>• 从产生 PRE/PALL 命令到对同一存储体产生 ACTV 命令</li> <li>• 从产生自动刷新时的 PALL 命令到产生 REF 命令</li> <li>• 从产生自刷新时的 PALL 命令到产生 SELF 命令</li> </ul> 区域 2 和区域 3 的设定是通用的。 00: 0 个周期（不插入等待周期） 01: 1 个周期 10: 2 个周期 11: 3 个周期
12	—	0	R	保留位 读写的值总是 0。
11、10	WTRCD[1:0]	01	R/W	ACTV 命令 → READ(A)/WRIT(A) 命令之间的等待周期数 此位指定从产生 ACTV 命令后到产生 READ(A)/WRIT(A) 命令的最小等待周期数。 区域 2 和区域 3 的设定是通用的。 00: 0 个周期（不插入等待周期） 01: 1 个周期 10: 2 个周期 11: 3 个周期
9	—	0	R	保留位 读写的值总是 0。
8、7	A3CL[1:0]	10	R/W	区域 3 的 CAS 潜伏时间 此位指定区域 3 的 CAS 潜伏时间。 00: 1 个周期 01: 2 个周期 10: 3 个周期 11: 4 个周期
6、5	—	全 0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说 明
4、3	TRWL[1:0]	00	R/W	<p>启动预充电的等待周期数</p> <p>在以下情况下，此位指定为了等待预充电启动而要插入的最小等待周期数。</p> <ul style="list-style-type: none"> <li>从本 LSI 产生 WRITA 命令到在 SDRAM 内启动自动预充电在非存储体激活模式中产生 WRITA 命令后，对同一存储体产生 ACTV 命令时。</li> <li>另外，必须通过各 SDRAM 的数据表确认在 SDRAM 内接受 WRITA 命令后启动自动预充电需要几个周期。该周期数不能超过此位指定的周期数。</li> <li>从产生 WRIT 命令到产生 PRE 命令在存储体激活模式中，存取同一存储体内的不同行地址时。</li> </ul> <p>区域 2 和区域 3 的设定是通用的。</p> <p>00: 0 个周期 (不插入等待周期)</p> <p>01: 1 个周期</p> <p>10: 2 个周期</p> <p>11: 3 个周期</p>
2	—	0	R	<p>保留位</p> <p>读写的值总是 0。</p>
1、0	WTRC[1:0]	00	R/W	<p>REF 命令 / 自刷新解除 →ACTV/REF/MRS 命令之间的空闲周期数</p> <p>此位指定以下情况下的命令之间的最小空闲周期数。</p> <ul style="list-style-type: none"> <li>从产生 REF 命令到产生 ACTV/REF/MRS 命令</li> <li>从解除自刷新到产生 ACTV/REF/MRS 命令</li> </ul> <p>区域 2 和区域 3 的设定是通用的。</p> <p>00: 2 个周期</p> <p>01: 3 个周期</p> <p>10: 5 个周期</p> <p>11: 8 个周期</p>

(5) PCMCIA

• CS5WCR、CS6WCR

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	SA[1:0]		-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	TED[3:0]				PCW[3:0]				WM	-	-	TEH[3:0]			
初始值:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
31 ~ 22	—	全 0	R	保留位 读写的值总是 0。
21、20	SA[1:0]	00	R/W	空间属性的指定 在设定 PCMCIA 接口时，此位指定存储卡接口或者 I/O 卡接口。 SA1 0: 将 A25=1 的空间指定为存储卡接口 1: 将 A25=1 的空间指定为 I/O 卡接口 SA0 0: 将 A25=0 的空间指定为存储卡接口 1: 将 A25=0 的空间指定为 I/O 卡接口
19 ~ 15	—	全 0	R	保留位 读写的值总是 0。
14 ~ 11	TED[3:0]	0000	R/W	从地址输出到 $\overline{RD}$ 、 $\overline{WE}$ 有效的延迟时间 此位设定从 PCMCIA 接口的地址输出到 $\overline{RD}$ 和 $\overline{WE}$ 有效的延迟时间。 0000: 0.5 个周期 0001: 1.5 个周期 0010: 2.5 个周期 0011: 3.5 个周期 0100: 4.5 个周期 0101: 5.5 个周期 0110: 6.5 个周期 0111: 7.5 个周期 1000: 8.5 个周期 1001: 9.5 个周期 1010: 10.5 个周期 1011: 11.5 个周期 1100: 12.5 个周期 1101: 13.5 个周期 1110: 14.5 个周期 1111: 15.5 个周期

位	位名	初始值	R/W	说 明
10 ~ 7	PCW[3:0]	1010	R/W	<p>存取的等待周期数 此位设定要插入的等待周期数。</p> <p>0000: 3 个周期 0001: 6 个周期 0010: 9 个周期 0011: 12 个周期 0100: 15 个周期 0101: 18 个周期 0110: 22 个周期 0111: 26 个周期 1000: 30 个周期 1001: 33 个周期 1010: 36 个周期 1011: 38 个周期 1100: 52 个周期 1101: 60 个周期 1110: 64 个周期 1111: 80 个周期</p>
6	WM	0	R/W	<p>外部等待屏蔽的指定 此位指定是将外部等待输入设定为有效还是忽视外部等待输入。即使存取等待周期数是 0，此位的设定也有效。</p> <p>0: 外部等待输入有效 1: 忽视外部等待输入</p>
5、4	—	全 0	R	<p>保留位 读写的值总是 0。</p>
3 ~ 0	TEH[3:0]	0000	R/W	<p>从 <math>\overline{RD}</math>、<math>\overline{WE}</math> 无效到地址输出的延迟保持时间 此位设定 PCMCIA 接口的 <math>\overline{RD}</math> 和 <math>\overline{WE}</math> 无效后的地址保持时间。</p> <p>0000: 0.5 个周期 0001: 1.5 个周期 0010: 2.5 个周期 0011: 3.5 个周期 0100: 4.5 个周期 0101: 5.5 个周期 0110: 6.5 个周期 0111: 7.5 个周期 1000: 8.5 个周期 1001: 9.5 个周期 1010: 10.5 个周期 1011: 11.5 个周期 1100: 12.5 个周期 1101: 13.5 个周期 1110: 14.5 个周期 1111: 15.5 个周期</p>

(6) 突发 MPX-I/O

• CS6WCR

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	MPXAW[1:0]	MPXMD	-	-	BW[1:0]	
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	W[3:0]			WM	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

位	位名	初始值	R/W	说 明																																																								
31 ~ 22	—	全 0	R	保留位 读写的值总是 0。																																																								
21、20	MPXAW[1:0]	00	R/W	地址周期的插入等待数 此位设定要插入到地址周期的等待数。 00: 0 个周期 01: 1 个周期 10: 2 个周期 11: 3 个周期																																																								
19	MPXMD	0	R/W	突发 MPX-I/O 接口的模式指定 此位指定 16 字节存取时的存取模式。 0: 对长度为 16 字节的传送进行 4 突发存取 1: 对长度为 8 字节的传送进行 2 突发存取 MPXMD=0 时的传送长度: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>D31</th> <th>D30</th> <th>D29</th> <th>传送长度</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>字节 (1 字节)</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>字 (2 字节)</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>长字 (4 字节)</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>保留</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>16 字节</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>保留</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>保留</td></tr> </tbody> </table> MPXMD=1 时的传送长度: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>D31</th> <th>D30</th> <th>D29</th> <th>传送长度</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>字节 (1 字节)</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>字 (2 字节)</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>长字 (4 字节)</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>4 倍字 (8 字节)</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>保留</td></tr> </tbody> </table>	D31	D30	D29	传送长度	0	0	0	字节 (1 字节)	0	0	1	字 (2 字节)	0	1	0	长字 (4 字节)	0	1	1	保留	1	0	0	16 字节	1	0	1	保留	1	1	0	保留	D31	D30	D29	传送长度	0	0	0	字节 (1 字节)	0	0	1	字 (2 字节)	0	1	0	长字 (4 字节)	0	1	1	4 倍字 (8 字节)	1	0	0	保留
D31	D30	D29	传送长度																																																									
0	0	0	字节 (1 字节)																																																									
0	0	1	字 (2 字节)																																																									
0	1	0	长字 (4 字节)																																																									
0	1	1	保留																																																									
1	0	0	16 字节																																																									
1	0	1	保留																																																									
1	1	0	保留																																																									
D31	D30	D29	传送长度																																																									
0	0	0	字节 (1 字节)																																																									
0	0	1	字 (2 字节)																																																									
0	1	0	长字 (4 字节)																																																									
0	1	1	4 倍字 (8 字节)																																																									
1	0	0	保留																																																									

位	位名	初始值	R/W	说 明
18	—	0	R	保留位 读写的值总是 0。
17、16	BW[1:0]	00	R/W	突发的等待周期数 此位指定要插入到突发存取时的第 2 次以后存取周期的等待周期数。 00: 0 个周期 01: 1 个周期 10: 2 个周期 11: 3 个周期
15 ~ 11	—	全 0	R	保留位 读写的值总是 0。
10 ~ 7	W[3:0]	1010	R/W	存取等待周期数 此位指定要插入到第 1 次突发存取周期或者单地址存取的等待周期数。 0000: 0 个周期 0001: 1 个周期 0010: 2 个周期 0011: 3 个周期 0100: 4 个周期 0101: 5 个周期 0110: 6 个周期 0111: 8 个周期 1000: 10 个周期 1001: 12 个周期 1010: 14 个周期 1011: 18 个周期 1100: 24 个周期 1101: 保留 (不能设定) 1110: 保留 (不能设定) 1111: 保留 (不能设定)
6	WM	0	R/W	外部等待屏蔽的指定 此位指定是将外部等待输入设定为有效还是忽视外部等待输入。即使存取等待周期数是 0, 此位的设定也有效。 0: 外部等待输入有效 1: 忽视外部等待输入
5 ~ 0	—	全 0	R	保留位 读写的值总是 0。

(7) 突发 ROM (时钟同步)

• CS0WCR

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	BW[1:0]
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	W[3:0]			WM	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
31 ~ 18	—	全 0	R	保留位 读写的值总是 0。
17、16	BW[1:0]	00	R/W	突发的等待周期数 此位指定要插入到突发存取时的第 2 次以后存取周期的等待周期数。 00: 0 个周期 01: 1 个周期 10: 2 个周期 11: 3 个周期
15 ~ 11	—	全 0	R	保留位 读写的值总是 0。
10 ~ 7	W[3:0]	1010	R/W	存取的等待周期数 此位指定要插入到第 1 次存取周期的等待周期数。 0000: 0 个周期 0001: 1 个周期 0010: 2 个周期 0011: 3 个周期 0100: 4 个周期 0101: 5 个周期 0110: 6 个周期 0111: 8 个周期 1000: 10 个周期 1001: 12 个周期 1010: 14 个周期 1011: 18 个周期 1100: 24 个周期 1101: 保留 (不能设定) 1110: 保留 (不能设定) 1111: 保留 (不能设定)
6	WM	0	R/W	外部等待屏蔽的指定 此位指定是将外部等待输入设定为有效还是忽视外部等待输入。即使存取等待周期数是 0, 此位的设定也有效。 0: 外部等待输入有效 1: 忽视外部等待输入
5 ~ 0	—	全 0	R	保留位 读写的值总是 0。



## 9.4.4 SDRAM 控制寄存器 (SDCR)

SDCR 指定 SDRAM 的刷新方法、存取方法以及要连接的 SDRAM 种类。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	A2ROW[1:0]	-	-	A2COL[1:0]	
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	RFSH	RMODE	-	BACTV	-	-	-	A3ROW[1:0]	-	-	A3COL[1:0]	
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R	R/W	R	R	R	R/W	R/W	R	R/W	R/W

位	位名	初始值	R/W	说 明
31 ~ 21	—	全 0	R	保留位 读写的值总是 0。
20、19	A2ROW[1:0]	00	R/W	区域 2 行地址的位数 此位指定区域 2 行地址的位数。 00: 11 位 01: 12 位 10: 13 位 11: 保留 (不能设定)
18	—	0	R	保留位 读写的值总是 0。
17、16	A2COL[1:0]	00	R/W	区域 2 列地址的位数 此位指定区域 2 列地址的位数。 00: 8 位 01: 9 位 10: 10 位 11: 保留 (不能设定)
15 ~ 12	—	全 0	R	保留位 读写的值总是 0。
11	RFSH	0	R/W	刷新控制 此位指定是否刷新 SDRAM。 0: 不刷新 1: 刷新
10	RMODE	0	R/W	刷新控制 此位指定当 RFSH 位是 1 时, 是进行自动刷新还是进行自刷新。当 RFSH 位和此位都设定为 1 时, 设定后立即进入自刷新模式; 当 RFSH 位设定为 1 而此位设定为 0 时, 就根据 RTCSR、RTCNT 和 RTCOR 寄存器设定的内容进行自动刷新。 0: 自动刷新 1: 自刷新
9	—	0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说 明
8	BACTV	0	R/W	<p>存储体激活模式</p> <p>此位指定是通过自动预充电模式（使用 READA 命令和 WRITA 命令）还是通过存储体激活模式（使用 READ 命令和 WRIT 命令）进行存取。</p> <p>0: 自动预充电模式（使用 READA 命令和 WRITA 命令）</p> <p>1: 存储体激活模式（使用 READ 命令和 WRIT 命令）</p> <p><b>【注】</b> 只有区域 3 才能设定存储体激活模式。此时，总线宽度可设定为 16 位或者 32 位。另外，如果将区域 2 和区域 3 都设定为 SDRAM，就必须设定为自动预充电模式。</p>
7 ~ 5	—	全 0	R	<p>保留位</p> <p>读写的值总是 0。</p>
4、3	A3ROW[1:0]	00	R/W	<p>区域 3 行地址的位数</p> <p>此位指定区域 3 行地址的位数。</p> <p>00: 11 位</p> <p>01: 12 位</p> <p>10: 13 位</p> <p>11: 保留（不能设定）</p>
2	—	0	R	<p>保留位</p> <p>读写的值总是 0。</p>
1、0	A3COL[1:0]	00	R/W	<p>区域 3 列地址的位数</p> <p>此位指定区域 3 列地址的位数。</p> <p>00: 8 位</p> <p>01: 9 位</p> <p>10: 10 位</p> <p>11: 保留（不能设定）</p>

## 9.4.5 刷新定时器的控制 / 状态寄存器 (RTCSR)

RTCSR 进行有关刷新 SDRAM 的各种设定。

在写 RTCSR 时，必须将写数据的高 16 位设定为 H'A55A 来解除写保护。

必须注意：因为对刷新定时器的计数器 (RTCNT) 进行递增计数的时钟只在上电复位时进行相位的匹配，所以从 CKS[2:0] 被设定为 B'000 以外的值开始到使定时器运行后最初设定比较匹配标志的期间有误差。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	CMF	CMIE	CKS[2:0]			RRC[2:0]		
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 16	—	全 0	R/W	写保护的解除位 为了在写 RTCSR 时解除写保护，必须将这些位设定为 H'A55A。读的值总是 0。
15 ~ 8	—	全 0	R	保留位 读写的值总是 0。
7	CMF	0	R/W	比较匹配标志 此位是表示刷新定时器的计数器 (RTCNT) 和刷新时间常数寄存器 (RTCOR) 的值一致的状态标志。在以下条件下进行置位 / 清除： 0: 清除条件：在 CMF=1 的状态下，在读 RTCSR 后给 CMF 写 0 时 1: 置位条件：当 RTCNT=RTCOR 时
6	CMIE	0	R/W	比较匹配的中断允许 此位设定在 RTCSR 的 CMF 位被置 1 时是否允许由 CMF 产生的中断请求。 0: 禁止由 CMF 产生的中断请求 1: 允许由 CMF 产生的中断请求
5 ~ 3	CKS[2:0]	000	R/W	时钟选择 此位选择刷新定时器的计数器 (RTCNT) 进行递增计数的时钟。 000: 停止递增计数 001: B $\phi$ /4 010: B $\phi$ /16 011: B $\phi$ /64 100: B $\phi$ /256 101: B $\phi$ /1024 110: B $\phi$ /2048 111: B $\phi$ /4096

位	位名	初始值	R/W	说 明
2 ~ 0	RRC[2:0]	000	R/W	刷新次数 此位指定在刷新定时器的计数器 (RTCNT) 和刷新时间常数寄存器 (RTCOR) 的值一致后发生刷新请求时的连续刷新次数。通过本功能可延长发生刷新的周期。 000: 1 次 001: 2 次 010: 4 次 011: 6 次 100: 8 次 101: 保留 (不能设定) 110: 保留 (不能设定) 111: 保留 (不能设定)

#### 9.4.6 刷新定时器的计数器 (RTCNT)

RTCNT 是 8 位计数器，通过 RTCSR 的 CKS2、CKS1 和 CKS0 位选择的时钟进行递增计数。如果 RTCNT 和 RTCOR 的值一致，RTCNT 就被清 0。当递增计数到 255 时，就返回 0。在写 RTCNT 时，必须将写数据的高 16 位设定为 H'A55A 来解除写保护。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-								
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
31 ~ 16	—	全 0	R/W	写保护的解除位 为了在写 RTCNT 时解除写保护，必须将这些位设定为 H'A55A。读的值总是 0。
15 ~ 8	—	全 0	R	保留位 读写的值总是 0。
7 ~ 0		全 0	R/W	8 位计数器

### 9.4.7 刷新时间常数寄存器 (RTCOR)

RTCOR 是 8 位寄存器。当 RTCOR 和 RTCNT 的值一致时，RTCSR 的 CMF 位被置 1，RTCNT 被清 0。

如果 SDCR 的 RFSH 位已被置 1，就通过此一致信号产生刷新请求，刷新请求被保持到实际刷新为止。如果在下一个一致信号前还没有处理刷新请求，前一个请求就无效。

如果 RTCSR 的 CMIE 位已被置 1，就通过此一致信号产生中断请求，在 RTCSR 的 CMF 位被清 0 前连续输出中断请求。CMF 位的清除只对中断产生影响，不会因此而清除刷新请求。因此，能设定为边进行刷新边使用中断对刷新请求的次数进行计数等，也能同时设定刷新和间隔定时器的中断。在写 RTCOR 时，必须将写数据的高 16 位设定为 H'A55A 来解除写保护。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-								
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
31 ~ 16	—	全 0	R/W	写保护的解除位 为了在写 RTCOR 时解除写保护，必须将此位设定为 H'A55A。读的值总是 0。
15 ~ 8	—	全 0	R	保留位 读写的值总是 0。
7 ~ 0		全 0	R/W	8 位计数器

### 9.4.8 总线功能扩展寄存器 (BSCEHR)

BSCEHR 是 16 位寄存器，设定 DTC 和 DMAC 的总线权释放时序等。对于优先进行 DTC 或者 DMAC 传送以及减少 DTC 启动周期数，BSCEHR 能设定有效的功能。

有关 DTLOCK、CSSTP1 和 DTBST 位的组合引起的 DTC 运行的不同点，请参照“8.5.9 DTC 的总线权释放时序”。

通过设定 CSSTP2 位，能提高突发模式中的 DMAC 传送性能和 DTC 传送 (DTLOCK 位是 0) 性能。还能通过设定 CSSTP3 位，选择周期挪用模式中的 DMAC 传送 /DTC 传送和 CPU 存取外部空间的优先级。

通过设定 DTSA 位，能实现 DTC 的短地址模式。有关短地址模式的详细内容，请参照“8.4 传送信息的分配和 DTC 向量表”。

当在启动 DTC 前发生多个 DTC 启动源时，能通过 DTPR 位设定 DTC 启动的优先级。

对于 DMMTU4 ~ 0 位，需要设定和 MTU2 突发模式中的 DMAC 传送时对应的位。

另外，不能在启动 DMAC 或者 DTC 时更改本寄存器。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTLOCK	CSSTP1	-	CSSTP2	DTBST	DTSA	CSSTP3	DTPR	-	-	-	DMMTU4	DMMTU3	DMMTU2	DMMTU1	DMMTU0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	DTLOCK	0	R/W	DTC 锁定的允许 此位选择 DTC 的总线权释放时序。 0: 在产生读向量后的 NOP 时以及在回写传送信息后，释放总线权。 1: 在读向量后、产生读向量后的 NOP、读传送信息后、传送 1 次数据后以及回写传送信息后，释放总线权。
14	CSSTP1	0	R/W	产生 DTC 的 NOP 时的总线权释放选择 此位选择是否在启动 DTC 期间产生读向量后的 NOP 时，对 CPU 存取外部空间的请求释放总线权。 但是，当 CSSTP2 位是 1 时，与此位的设定无关，始终不释放总线权。 0: 在产生 DTC 的 NOP 时释放总线权 1: 在产生 DTC 的 NOP 时不释放总线权
13	—	0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说 明
12	CSSTP2	0	R/W	<p>突发模式中的 DMAC/DTC 传送的总线权释放选择</p> <p>在以外部请求或者 MTU2 为启动源的通道固定模式中，当进行突发模式中的 DMAC 传送时或者进行 DTC 传送（DTLOCK 位是 0）时，对于 CPU 存取外部空间的请求，此位选择是否在每传送 1 次数据后释放总线权。</p> <ul style="list-style-type: none"> <li>• DMAC 传送时 <ul style="list-style-type: none"> <li>0: 在每传送 1 次数据后释放总线权</li> <li>1: 在全部数据传送结束后释放总线权</li> </ul> </li> <li>• DTC 传送时 <ul style="list-style-type: none"> <li>0: 当 DTLOCK 位和 CSSTP1 位都是 0 时，在产生读向量后的 NOP 时释放总线权</li> <li>当 DTLOCK 位是 0 而 CSSTP1 位是 1 时，在每传送 1 次数据后释放总线权</li> <li>1: 在全部数据传送结束后释放总线权</li> </ul> </li> </ul> <p>【注】 在周期模式中，与此位的设定无关，在全部数据传送结束后释放总线权。</p>
11	DTBST	0	R/W	<p>DTC 突发的允许</p> <p>当对 DTC 产生多个启动请求时，不释放总线权而连续启动 DTC。</p> <ul style="list-style-type: none"> <li>0: 每当 1 个 DTC 启动源结束时，释放总线权。</li> <li>1: 在全部 DTC 启动源结束前，不释放总线权而连续启动 DTC。</li> </ul> <p>【注】 当将此位设定为 1 时，有以下限制：</p> <ol style="list-style-type: none"> <li>1. 必须通过频率控制寄存器（FRQCR）将时钟设定为 <math>l\phi:B\phi:P\phi:MI\phi:MP\phi=8:4:4:4:4、4:2:2:2:2</math> 或者 <math>2:1:1:1:1</math></li> <li>2. 必须将向量信息分配到内部 ROM 或者内部 RAM</li> <li>3. 必须将传送信息分配到内部 RAM</li> <li>4. 必须将传送源和传送目标分配到内部 RAM 和内部外围模块之间、或者外部存储器和内部外围模块之间</li> </ol>
10	DTSA	0	R/W	<p>DTC 短地址模式</p> <p>是用 3 个长字读 DTC 传送信息的模式。</p> <ul style="list-style-type: none"> <li>0: 用 4 个长字读传送信息。传送信息的分配为图 8.2 通常模式时的分配。</li> <li>1: 用 3 个长字读传送信息。传送信息的分配为图 8.2 短模式时的分配。</li> </ul> <p>【注】 因为将 SAR 和 DAR 的高 8 位视为 1，所以只能在内部外围模块和内部 RAM 之间的传送时使用。</p>
9	CSSTP3	0	R/W	<p>CPU 存取外部存储器的优先选择</p> <p>此位选择在周期挪用模式中的 DMAC 传送和 DTC 传送时，是否优先 CPU 存取外部空间。</p> <ul style="list-style-type: none"> <li>0: 优先 DMAC 传送和 DTC 传送</li> <li>1: 优先 CPU 存取外部空间</li> </ul> <p>【注】 本位为 0 时，在 CPU 存取内部 I/O 和之后产生的 CPU 存取外部空间之间，插入 <math>1B\phi</math> 的 NOP。</p>

位	位名	初始值	R/W	说 明
8	DTPR	0	R/W	<p>DTC 的启动优先级的指定</p> <p>此位选择在启动 DTC 前发生多个 DTC 启动请求时, 是从最初发生的启动请求还是根据 DTC 的启动优先级开始传送。</p> <p>但是, 如果在启动 DTC 时发生多个 DTC 启动请求, 就根据 DTC 的启动优先级进行下一次传送。</p> <p>0: 从最初发生的 DTC 启动请求开始传送</p> <p>1: 根据 DTC 的启动优先级开始传送</p> <p><b>【注】</b> 当在将此位设定为 1 时, 有以下限制:</p> <ol style="list-style-type: none"> <li>1. 必须将向量信息分配到内部 ROM 或者内部 RAM</li> <li>2. 必须将传送信息分配到内部 RAM</li> <li>3. 传送信息的跳读功能总是无效</li> </ol>
7 ~ 5	—	全 0	R	<p>保留位</p> <p>读写的值总是 0。</p>
4	DMMTU4	0	R/W	<p>允许以 TGIA_4 为启动源的突发模式中的 DMAC 传送</p> <p>在以 MTU2 的 TGIA_4 中断为启动源的突发模式中进行 DMAC 传送时, 必须将此位设定为 1。</p> <p>0: 禁止以 TGIA_4 为启动源的突发模式中的 DMAC 传送</p> <p>1: 允许以 TGIA_4 为启动源的突发模式中的 DMAC 传送</p> <p><b>【注】</b> 在进行周期挪用模式中的 DMAC 传送时, 必须将此位设定为 0。</p>
3	DMMTU3	0	R/W	<p>允许以 TGIA_3 为启动源的突发模式中的 DMAC 传送</p> <p>在以 MTU2 的 TGIA_3 中断为启动源的突发模式中进行 DMAC 传送时, 必须将此位设定为 1。</p> <p>0: 禁止以 TGIA_3 为启动源的突发模式中的 DMAC 传送</p> <p>1: 允许以 TGIA_3 为启动源的突发模式中的 DMAC 传送</p> <p><b>【注】</b> 在进行周期挪用模式中的 DMAC 传送时, 必须将此位设定为 0。</p>
2	DMMTU2	0	R/W	<p>允许以 TGIA_2 为启动源的突发模式中的 DMAC 传送</p> <p>在以 MTU2 的 TGIA_2 中断为启动源的突发模式中进行 DMAC 传送时, 必须将此位设定为 1。</p> <p>0: 禁止以 TGIA_2 为启动源的突发模式中的 DMAC 传送</p> <p>1: 允许以 TGIA_2 为启动源的突发模式中的 DMAC 传送</p> <p><b>【注】</b> 在进行周期挪用模式中的 DMAC 传送时, 必须将此位设定为 0。</p>
1	DMMTU1	0	R/W	<p>允许以 TGIA_1 为启动源的突发模式中的 DMAC 传送</p> <p>在以 MTU2 的 TGIA_1 中断为启动源的突发模式中进行 DMAC 传送时, 必须将此位设定为 1。</p> <p>0: 禁止以 TGIA_1 为启动源的突发模式中的 DMAC 传送</p> <p>1: 允许以 TGIA_1 为启动源的突发模式中的 DMAC 传送</p> <p><b>【注】</b> 在进行周期挪用模式中的 DMAC 传送时, 必须将此位设定为 0。</p>
0	DMMTU0	0	R/W	<p>允许以 TGIA_0 为启动源的突发模式中的 DMAC 传送</p> <p>在以 MTU2 的 TGIA_0 中断为启动源的突发模式中进行 DMAC 传送时, 必须将此位设定为 1。</p> <p>0: 禁止以 TGIA_0 为启动源的突发模式中的 DMAC 传送</p> <p>1: 允许以 TGIA_0 为启动源的突发模式中的 DMAC 传送</p> <p><b>【注】</b> 当进行周期挪用模式中的 DMAC 传送时, 必须将此位设定为 0。</p>



## 9.5 运行说明

### 9.5.1 端法 / 存取长度和数据定位

本 LSI 支持高位字节 (MSB) 为地址 0 的大端法字节数据的排列方法。

通常存储器和带字节选择的 SRAM 可选择 8 位、16 位或者 32 位数据总线；SDRAM 可选择 16 位或者 32 位数据总线；PCMCIA 接口可选择 8 位或者 16 位数据总线；MPX-I/O 为 8 位或者 16 位固定宽度的数据总线，或者根据存取地址为 8 位或者 16 位可变宽度的数据总线；突发 MPX-I/O 固定为 32 位数据总线。根据各器件的数据总线宽度进行数据的定位，如果要从 8 位宽的器件中读取长字数据，就必须读 4 次。在本 LSI 的各接口之间自动进行数据的定位和数据长度的转换。

器件的数据长度和存取单位的关系如表 9.17 ~ 表 9.19 所示。

表 9.17 32 位外部器件的存取和数据定位

操作	数据总线				门控信号			
	D31 ~ D24	D23 ~ D16	D15 ~ D8	D7 ~ D0	$\overline{\text{WRHH}}$ 、 $\overline{\text{DQMUU}}$	$\overline{\text{WRHL}}$ 、 $\overline{\text{DQMUL}}$	$\overline{\text{WRH}}$ 、 $\overline{\text{DQMLU}}$	$\overline{\text{WRL}}$ 、 $\overline{\text{DQMLL}}$
地址 0 的字节存取	数据 7 ~ 0	—	—	—	有效	—	—	—
地址 1 的字节存取	—	数据 7 ~ 0	—	—	—	有效	—	—
地址 2 的字节存取	—	—	数据 7 ~ 0	—	—	—	有效	—
地址 3 的字节存取	—	—	—	数据 7 ~ 0	—	—	—	有效
地址 0 的字存取	数据 15 ~ 8	数据 7 ~ 0	—	—	有效	有效	—	—
地址 2 的字存取	—	—	数据 15 ~ 8	数据 7 ~ 0	—	—	有效	有效
地址 0 的长字存取	数据 31 ~ 24	数据 23 ~ 16	数据 15 ~ 8	数据 7 ~ 0	有效	有效	有效	有效

表 9.18 16 位外部器件的存取和数据定位

操作		数据总线				门控信号			
		D31 ~ D24	D23 ~ D16	D15 ~ D8	D7 ~ D0	$\overline{\text{WRHH}}$ 、 $\overline{\text{DQMUU}}$	$\overline{\text{WRHL}}$ 、 $\overline{\text{DQMUL}}$	$\overline{\text{WRH}}$ 、 $\overline{\text{DQMLU}}$	$\overline{\text{WRL}}$ 、 $\overline{\text{DQMLL}}$
地址 0 的字节存取		—	—	数据 7 ~ 0	—	—	有效	—	
地址 1 的字节存取		—	—	—	数据 7 ~ 0	—	—	有效	
地址 2 的字节存取		—	—	数据 7 ~ 0	—	—	有效	—	
地址 3 的字节存取		—	—	—	数据 7 ~ 0	—	—	有效	
地址 0 的字存取		—	—	数据 15 ~ 8	数据 7 ~ 0	—	—	有效	有效
地址 2 的字存取		—	—	数据 15 ~ 8	数据 7 ~ 0	—	—	有效	有效
地址 0 的 长字存取	第 1 次 (地址 0)	—	—	数据 31 ~ 24	数据 23 ~ 16	—	—	有效	有效
	第 2 次 (地址 2)	—	—	数据 15 ~ 8	数据 7 ~ 0	—	—	有效	有效

表 9.19 8 位外部器件的存取和数据定位

操作		数据总线				门控信号			
		D31 ~ D24	D23 ~ D16	D15 ~ D8	D7 ~ D0	$\overline{\text{WRHH}}$ 、 DQMUU	$\overline{\text{WRHL}}$ 、 DQMUL	$\overline{\text{WRH}}$ 、 DQMLU	$\overline{\text{WRL}}$ 、 DQMLL
地址 0 的字节存取		—	—	—	数据 7 ~ 0	—	—	—	有效
地址 1 的字节存取		—	—	—	数据 7 ~ 0	—	—	—	有效
地址 2 的字节存取		—	—	—	数据 7 ~ 0	—	—	—	有效
地址 3 的字节存取		—	—	—	数据 7 ~ 0	—	—	—	有效
地址 0 的字存取	第 1 次 (地址 0)	—	—	—	数据 15 ~ 8	—	—	—	有效
	第 2 次 (地址 1)	—	—	—	数据 7 ~ 0	—	—	—	有效
地址 2 的字存取	第 1 次 (地址 2)	—	—	—	数据 15 ~ 8	—	—	—	有效
	第 2 次 (地址 3)	—	—	—	数据 7 ~ 0	—	—	—	有效
地址 0 的 长字存取	第 1 次 (地址 0)	—	—	—	数据 31 ~ 24	—	—	—	有效
	第 2 次 (地址 1)	—	—	—	数据 23 ~ 16	—	—	—	有效
	第 3 次 (地址 2)	—	—	—	数据 15 ~ 8	—	—	—	有效
	第 4 次 (地址 3)	—	—	—	数据 7 ~ 0	—	—	—	有效

## 9.5.2 通常空间接口

### (1) 基本时序

通常空间存取主要针对直接连接没有字节选择引脚的 SRAM 而输出门控信号。在使用带字节选择引脚的 SRAM 时，请参照“9.5.8 带字节选择的 SRAM 接口”。通常空间存取的基本时序如图 9.2 所示。没有等待的通常存取以 2 个周期结束， $\overline{BS}$  信号表示总线周期的开始而且为 1 个周期有效。

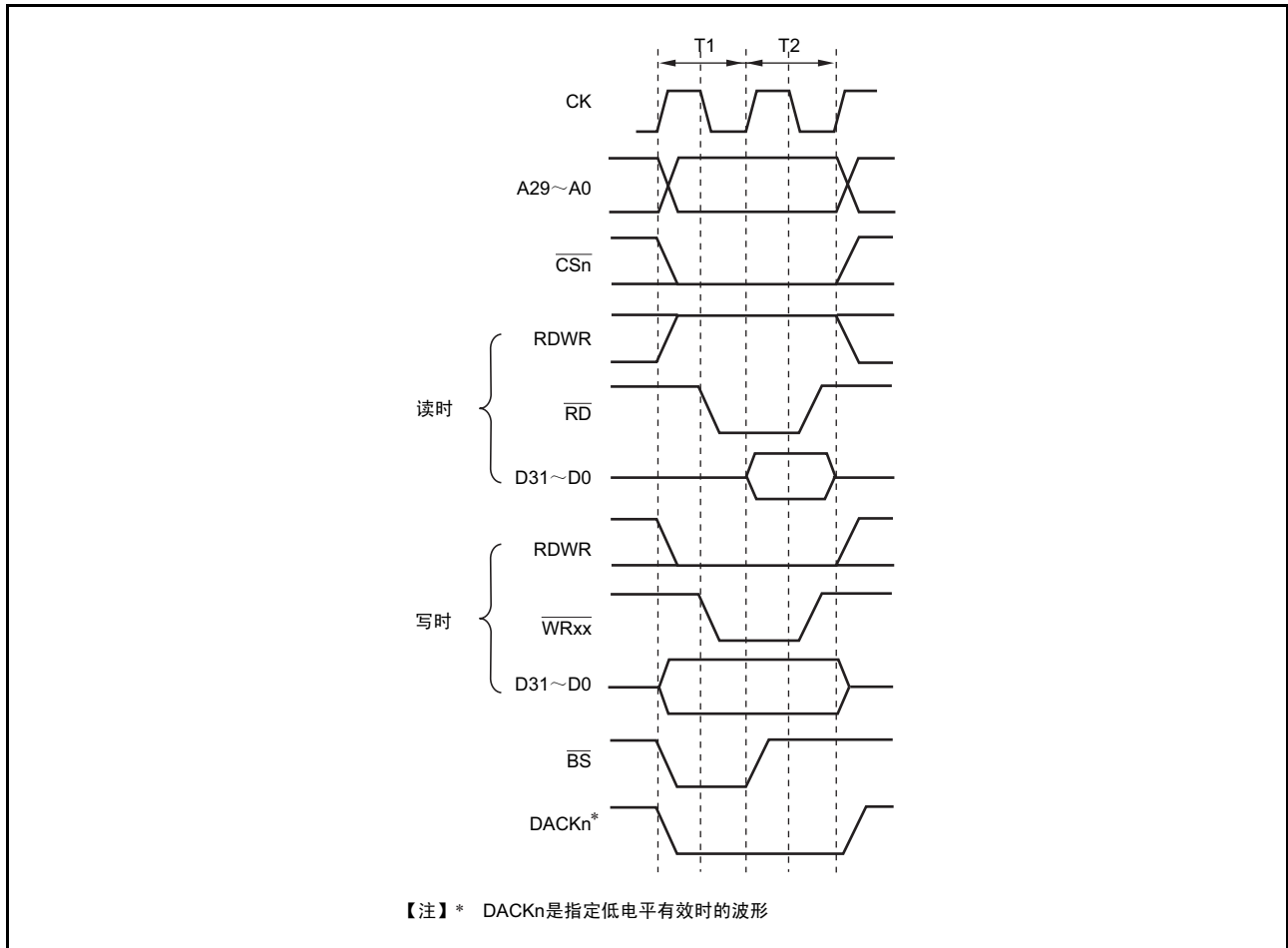


图 9.2 通常空间的基本存取（存取等待 = 0）

在进行读操作时，没有给外部总线指定存取长度。因为正确的存取起始地址输出到地址的最低位却没有指定存取长度，所以对于 32 位器件总是读 32 位数据，而对于 16 位器件总是读 16 位数据。在进行写操作时，只有写字节的  $\overline{WRxx}$  信号有效。

在给数据总线设置缓冲器时，需要使用  $\overline{RD}$  进行读数据的输出控制。必须注意：因为在没有存取时  $\overline{RDWR}$  信号为读状态（高电平输出），如果使用  $\overline{RDWR}$  信号进行外接数据缓冲器的控制，就有可能发生输出冲突。

通常空间的连续存取例子如图 9.3 和图 9.4 所示。当将 CSnWCR.WM 位设定为 0 时，为了评价外部等待而插入 1 个 Tnop 周期（图 9.3）；当将 CSnWCR.WM 位设定为 1 时，就忽视外部等待并能抑止 Tnop 周期的插入（图 9.4）。

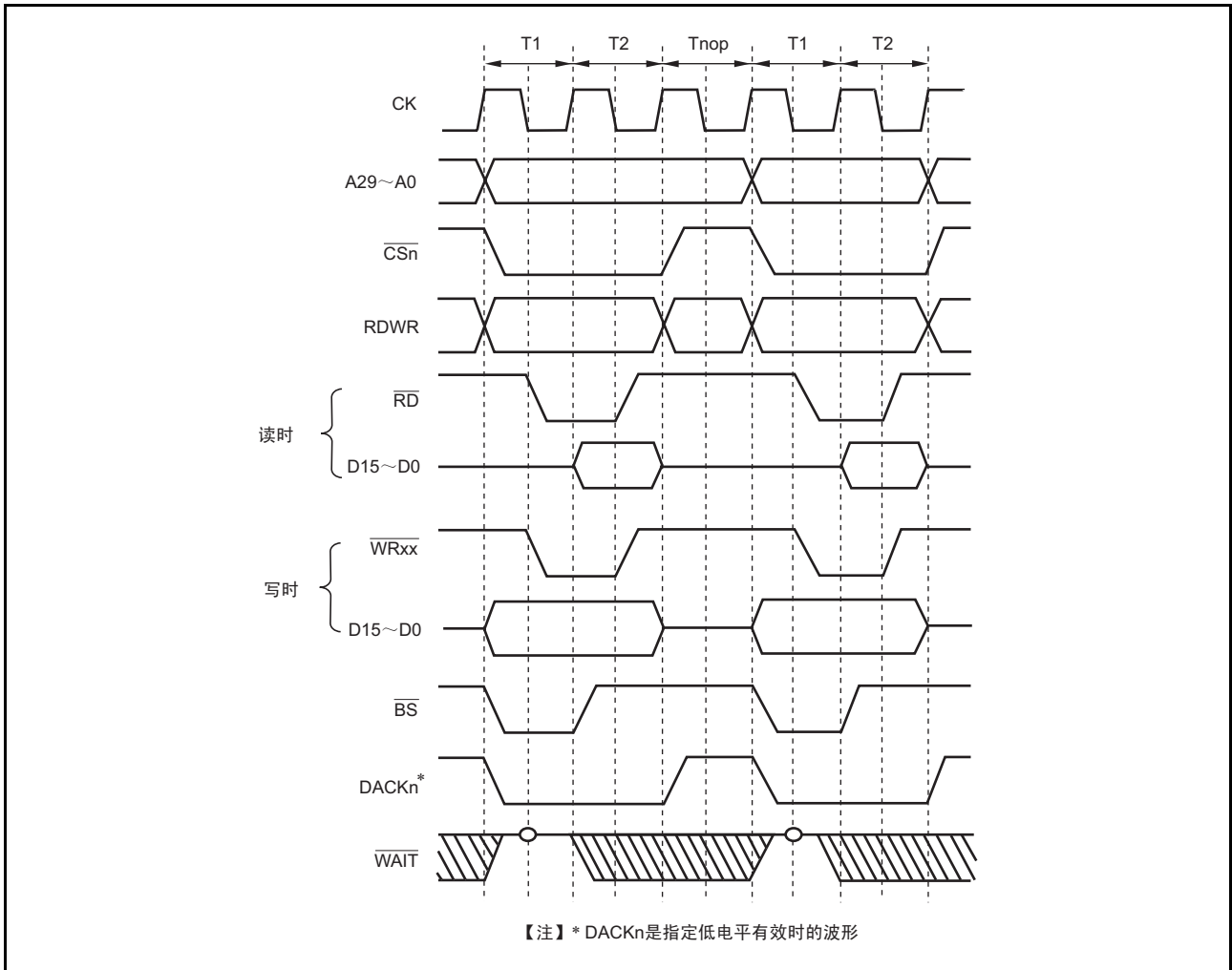


图 9.3 通常空间的连续存取例 1  
16 位总线、长字存取、CSnWCR.WM 位 = 0（存取等待 = 0、周期之间的等待 = 0）

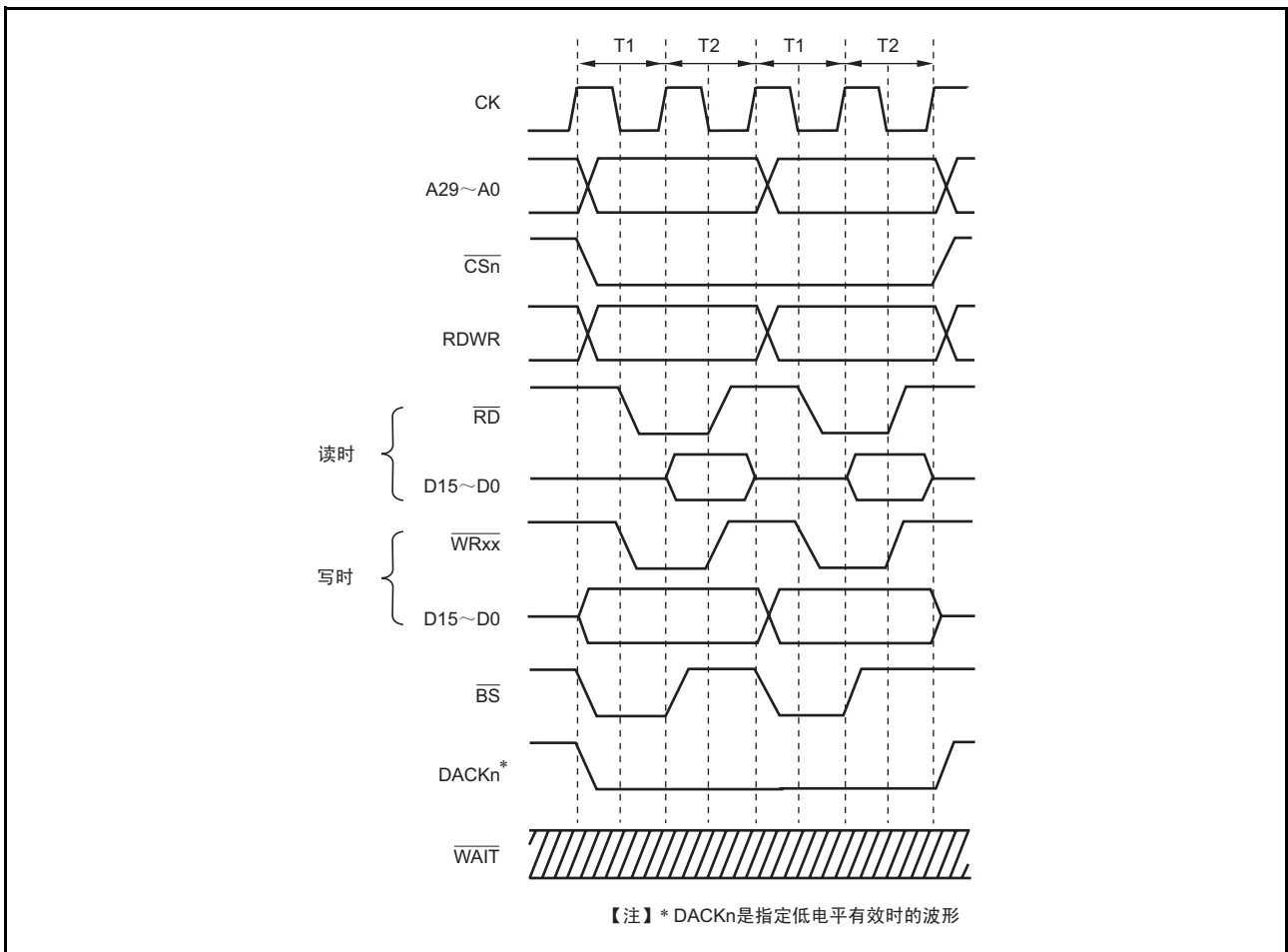


图 9.4 通常空间的连续存取例 2  
 16 位总线、长字存取、CSnWCR.WM 位 =1 (存取等待 =0、周期之间的等待 =0)

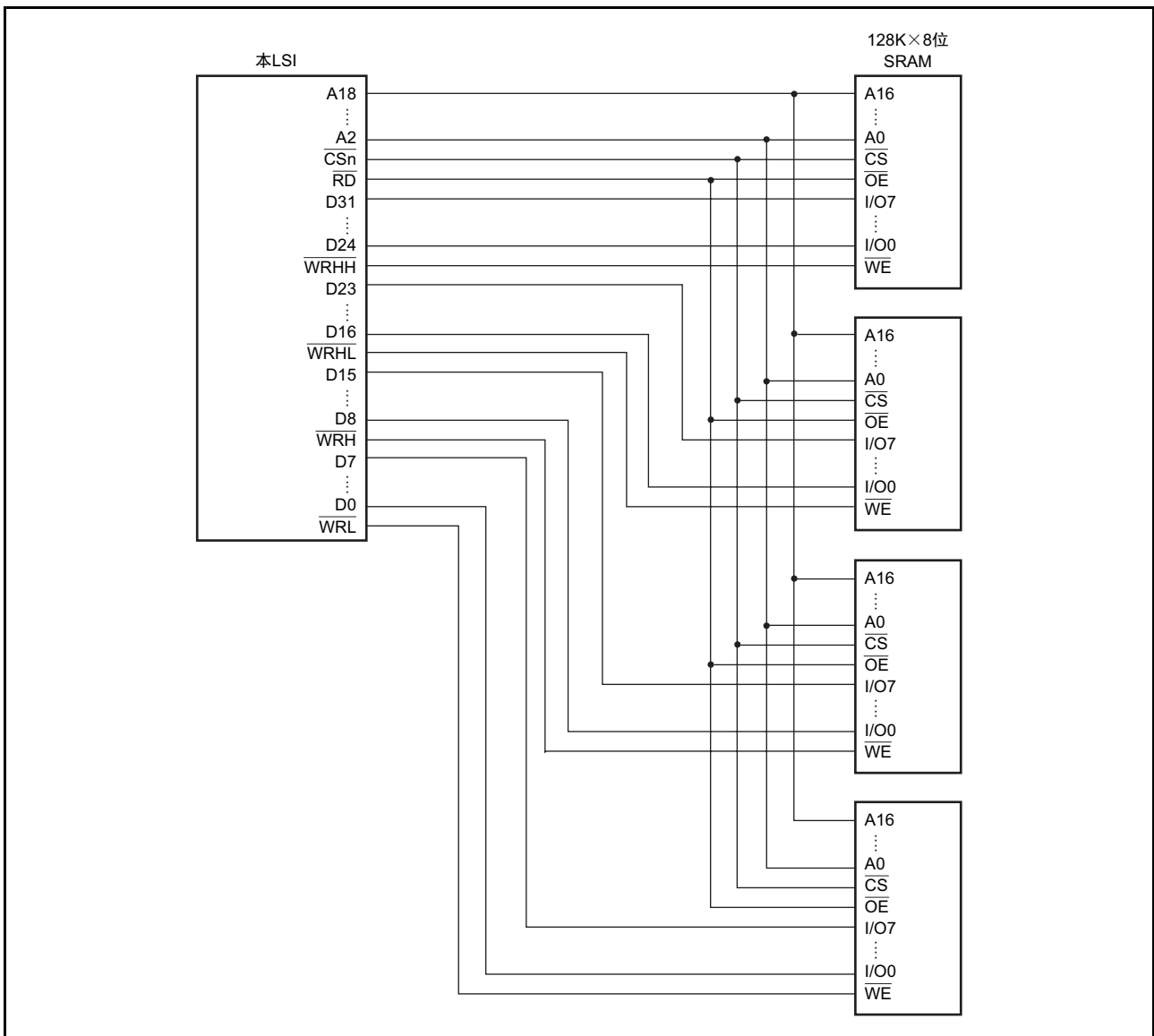


图 9.5 32 位数据的 SRAM 连接例子

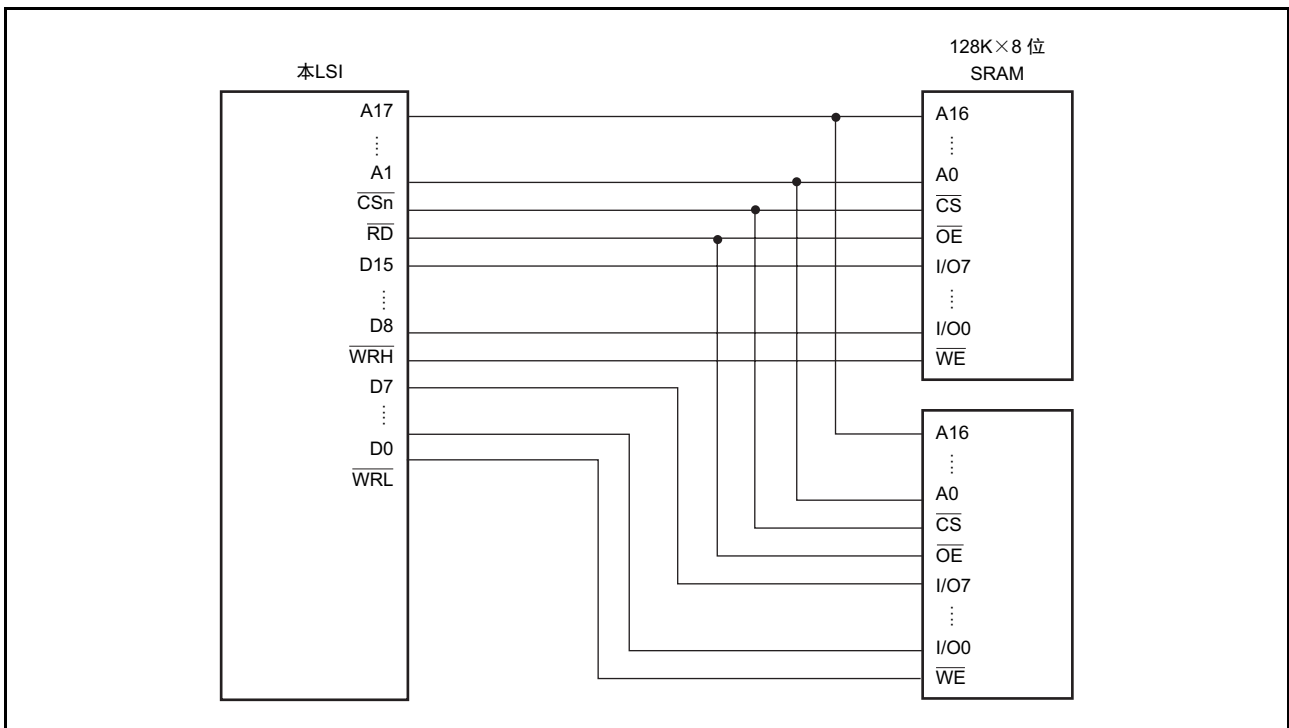


图 9.6 16 位数据的 SRAM 连接例子

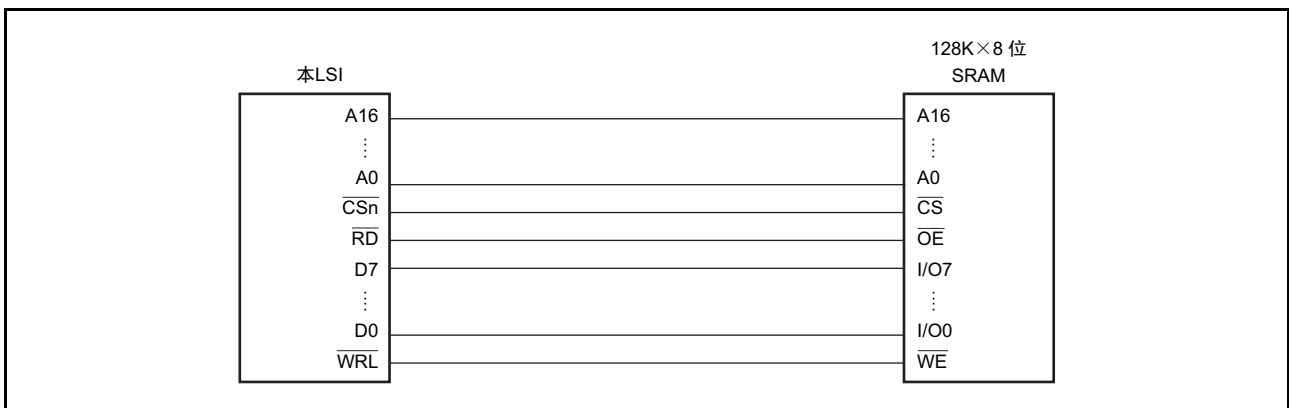


图 9.7 8 位数据的 SRAM 连接例子



### 9.5.3 存取等待的控制

能通过设定  $CS_nWCR$  的  $WR3$  位、 $WR2$  位、 $WR1$  位和  $WR0$  位，控制通常空间存取的等待周期的插入，还能在读写存取中独立插入等待周期。如图 9.8 所示，在通常空间存取中只插入被指定为等待周期的  $T_w$  周期数。

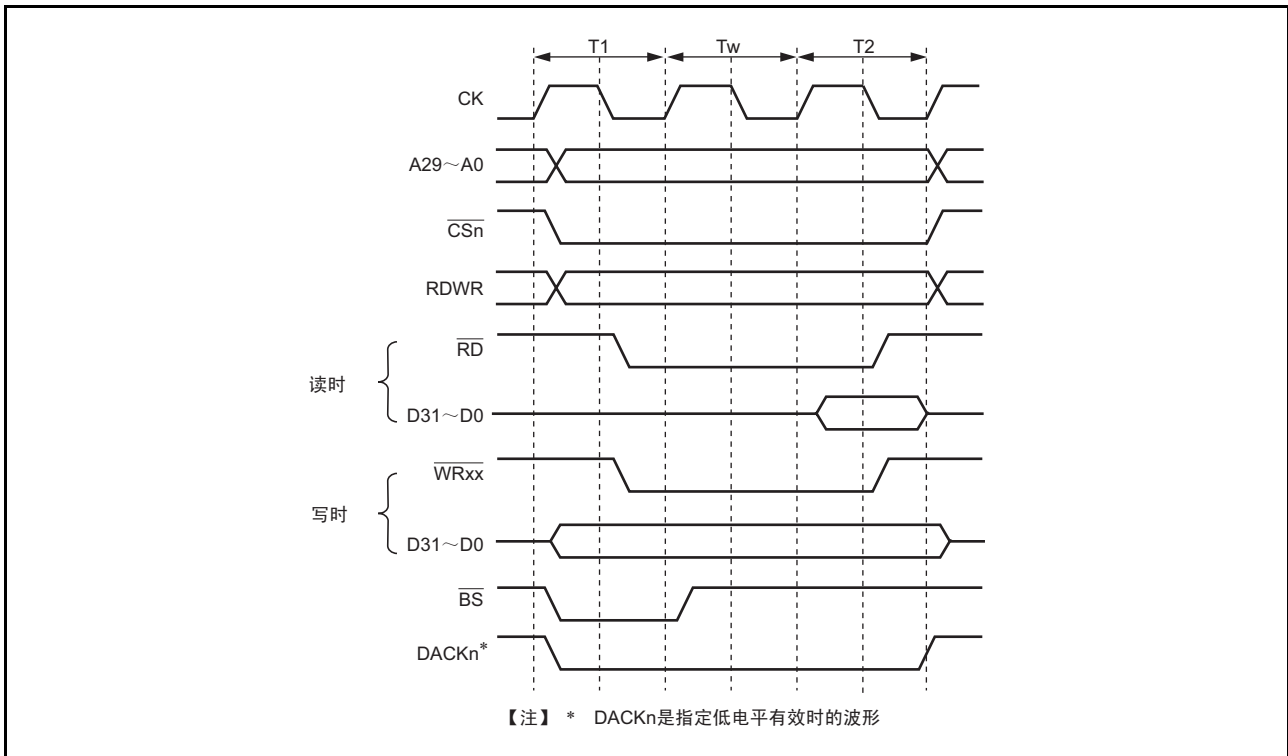


图 9.8 通常空间存取的等待时序（只限软件等待）

在将 CSnWCR 的 WM 位设定为 0 时，也对来自外部的等待输入  $\overline{\text{WAIT}}$  信号进行采样， $\overline{\text{WAIT}}$  信号的采样如图 9.9 所示。将 2 个周期的等待指定为软件等待，在从 T1 周期或者 Tw 周期转移到 T2 周期时，在 CK 的下降沿对  $\overline{\text{WAIT}}$  信号进行采样。

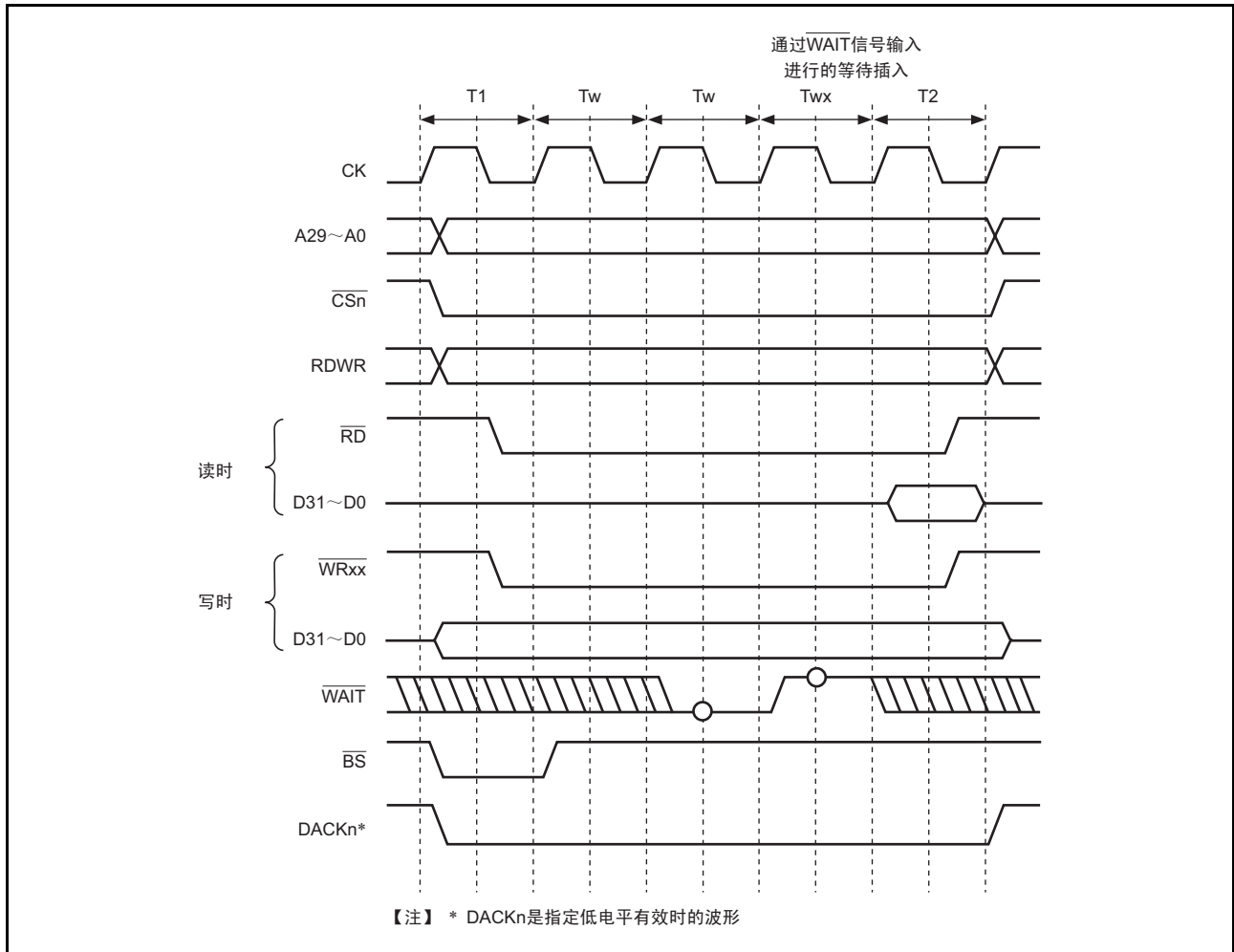


图 9.9 通常空间存取的等待时序（通过  $\overline{\text{WAIT}}$  信号进行的等待插入）

### 9.5.4 $\overline{CSn}$ 有效期间的扩展

能通过设定  $CSnWCR$  的  $SW1$  位和  $SW0$  位, 指定从  $\overline{CSn}$  有效到  $\overline{RD}$  和  $\overline{WRxx}$  有效的周期数, 还能通过设定  $HW1$  位和  $HW0$  位, 指定从  $\overline{RD}$  和  $\overline{WRxx}$  无效到  $\overline{CSn}$  无效的周期数。因此, 能对应各种外部器件的接口。在如图 9.10 所示的例子中, 将  $T_h$  周期和  $T_f$  周期分别附加在通常周期的前后, 在这些周期中, 除了  $\overline{RD}$  和  $\overline{WRxx}$  无效以外, 其他信号都有效。另外, 因为数据被延长输出到  $T_f$  周期为止, 所以对写操作较慢的器件有效。

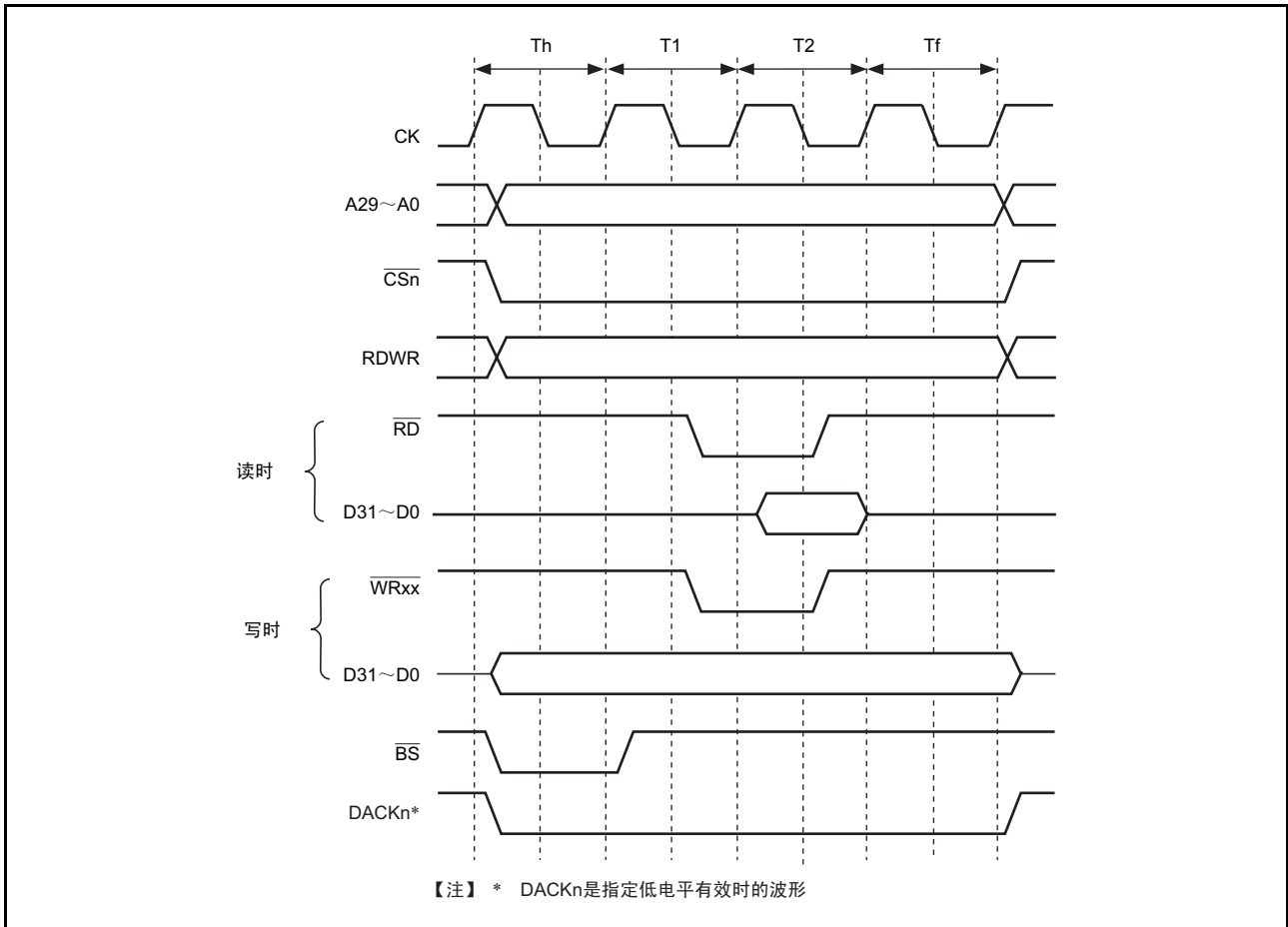


图 9.10  $\overline{CSn}$  有效期间的扩展

## 9.5.5 MPX-I/O 接口

MPX 空间的存取时序如下所示:

在 MPX 空间, 通过  $\overline{CSn}$ 、 $\overline{AH}$ 、 $\overline{RD}$  和  $\overline{WRxx}$  信号控制存取。MPX 空间的基本存取在进行 2 个周期的地址输出后继续进行通常空间的存取。地址输出周期和数据输入 / 输出周期的总线宽度固定为 8 位或者 16 位、或者根据存取地址为 8 位或者 16 位可变宽度。

在  $Ta2$  周期到  $Ta3$  周期之间从  $D15 \sim D0$  或者  $D7 \sim D0$  输出地址,  $Ta1$  周期为高阻抗状态, 即使在连续存取时不插入空闲周期, 也能防止地址和数据的冲突。另外, 通过将  $CS5WCR$  寄存器的  $MPXW$  位设定为 1, 地址输出变为 3 个周期。

$\overline{RDWR}$  信号和  $\overline{CSn}$  信号同时输出, 在读周期中输出高电平, 而在写周期中输出低电平。

数据周期和通常空间的存取周期相同。

时序图如图 9.11 ~ 图 9.13 所示。

必须注意: SH7080 群的 MPX-I/O 接口的操作时序和 SH7040 群不同。尤其是  $\overline{AH}$  信号, 在不存取 MPX-I/O 空间时, SH7080 群的  $\overline{AH}$  信号为无效 (高电平) 状态, 而 SH7040 群的  $\overline{AH}$  信号为有效 (低电平) 状态。

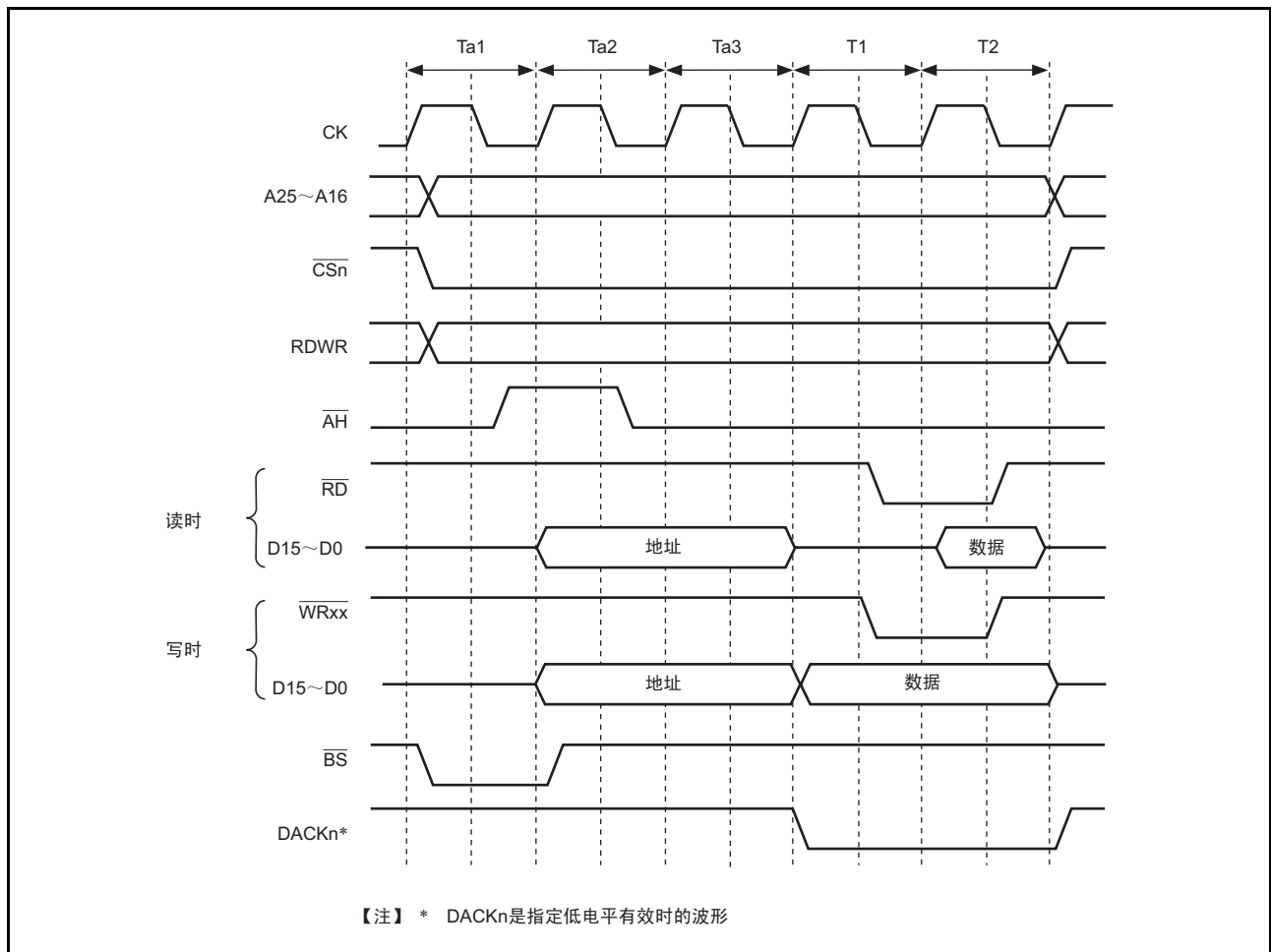


图 9.11 MPX 空间的存取时序  
(地址周期无等待、数据周期无等待)

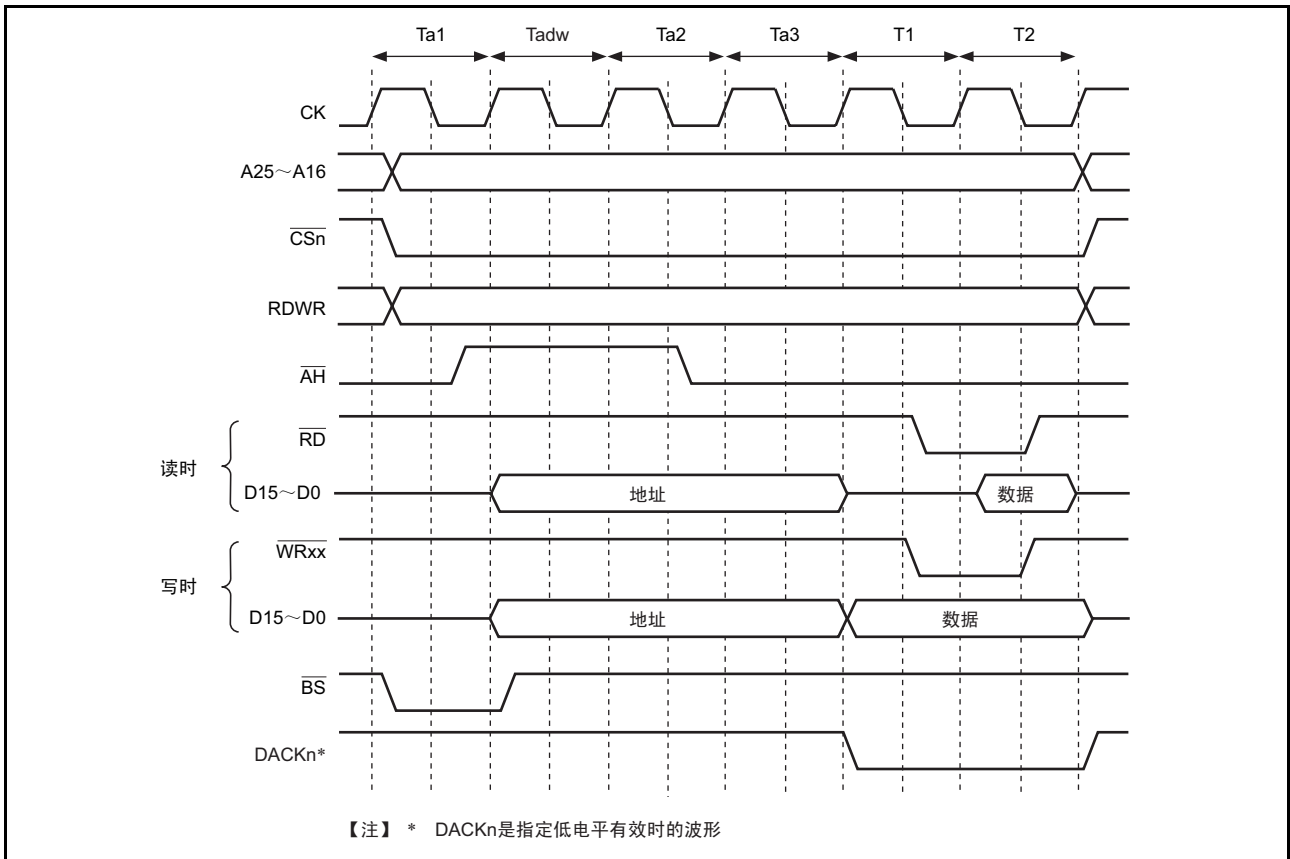


图 9.12 MPX 空间的存取时序  
(地址周期等待 =1、数据周期无等待)

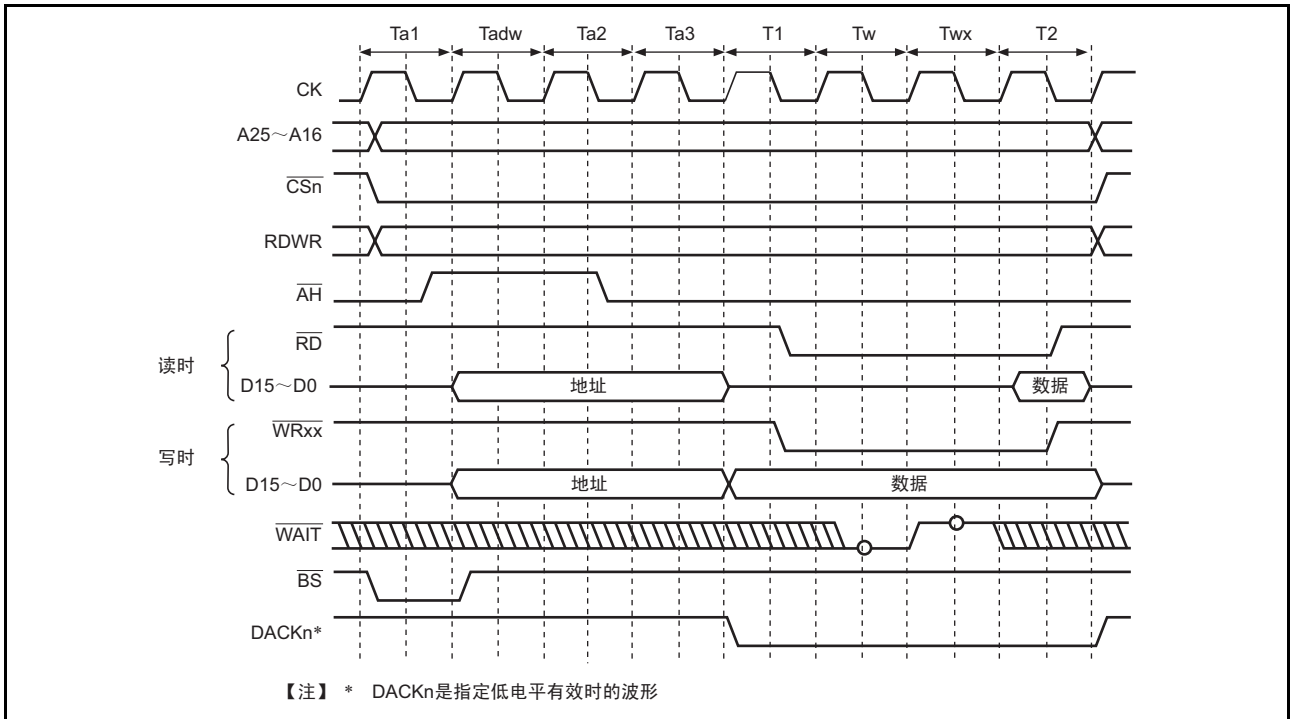


图 9.13 MPX 空间的存取时序  
(地址周期的存取等待 =1、数据周期等待 =1、外部等待 =1)

## 9.5.6 SDRAM 接口

### (1) 直接连接 SDRAM 的接口

能连接本 LSI 的 SDRAM 产品为：11/12/13 位的行地址、8/9/10 位的列地址、存储体数最多为 4 个，并且在读写命令周期中使用 A10 引脚设定预充电模式。

直接连接 SDRAM 的控制信号有  $\overline{\text{RASU}}$ 、 $\overline{\text{RASL}}$ 、 $\overline{\text{CASU}}$ 、 $\overline{\text{CASL}}$ 、RDWR、DQM<sub>UU</sub>、DQM<sub>UL</sub>、DQM<sub>LU</sub>、DQM<sub>LL</sub>、CKE、CS<sub>2</sub> 和 CS<sub>3</sub>。除 CS<sub>2</sub> 和 CS<sub>3</sub> 以外的信号对各空间是通用的，并且只在 CS<sub>2</sub> 和 CS<sub>3</sub> 有效时 CKE 以外的信号才有效。最多能将 SDRAM 连接到 2 个空间，可将连接 SDRAM 空间的数据总线宽度设定为 32 位或者 16 位。

SDRAM 的运行模式支持突发读 / 单次写（突发长度为 1）和突发读 / 突发写（突发长度为 1）。

通过  $\overline{\text{RASU}}$ 、 $\overline{\text{RASL}}$ 、 $\overline{\text{CASU}}$ 、 $\overline{\text{CASL}}$ 、RDWR 以及特定的地址信号指定 SDRAM 的命令。命令有 NOP、自动刷新 (REF)、自刷新 (SELF)、全部存储体的预充电 (PALL)、指定存储体的预充电 (PRE)、存储体激活 (ACTV)、读 (READ)、带预充电的读 (READA)、写 (WRIT)、带预充电的写 (WRITA) 以及模式寄存器 (MRS) 的写。通过 DQM<sub>UU</sub>、DQM<sub>UL</sub>、DQM<sub>LU</sub> 和 DQM<sub>LL</sub> 指定要存取的字节，读写相应的 DQM<sub>xx</sub> 为低电平的字节。DQM<sub>xx</sub> 和要存取字节的关系请参照“9.5.1 端法 / 存取长度和数据定位”。

本 LSI 和 SDRAM 的连接例子如图 9.14 ~ 图 9.16 所示。

如图 9.16 所示，通过在同一 CS 空间内使用  $\overline{\text{RASU}}$ 、 $\overline{\text{RASL}}$ 、 $\overline{\text{CASU}}$  和  $\overline{\text{CASL}}$ ，能使本 LSI 连接 2 个最大 32M 字节的 SDRAM。此时，在同一 CS 空间内有共 8 个存储体，用  $\overline{\text{RASL}}$  和  $\overline{\text{CASL}}$  指定的 4 个 SDRAM 存储体、用  $\overline{\text{RASU}}$  和  $\overline{\text{CASU}}$  指定的 4 个 SDRAM 存储体。在 A25=0 的地址存取时， $\overline{\text{RASL}}$  和  $\overline{\text{CASL}}$  有效；在 A25=1 的地址存取时， $\overline{\text{RASU}}$  和  $\overline{\text{CASU}}$  有效。

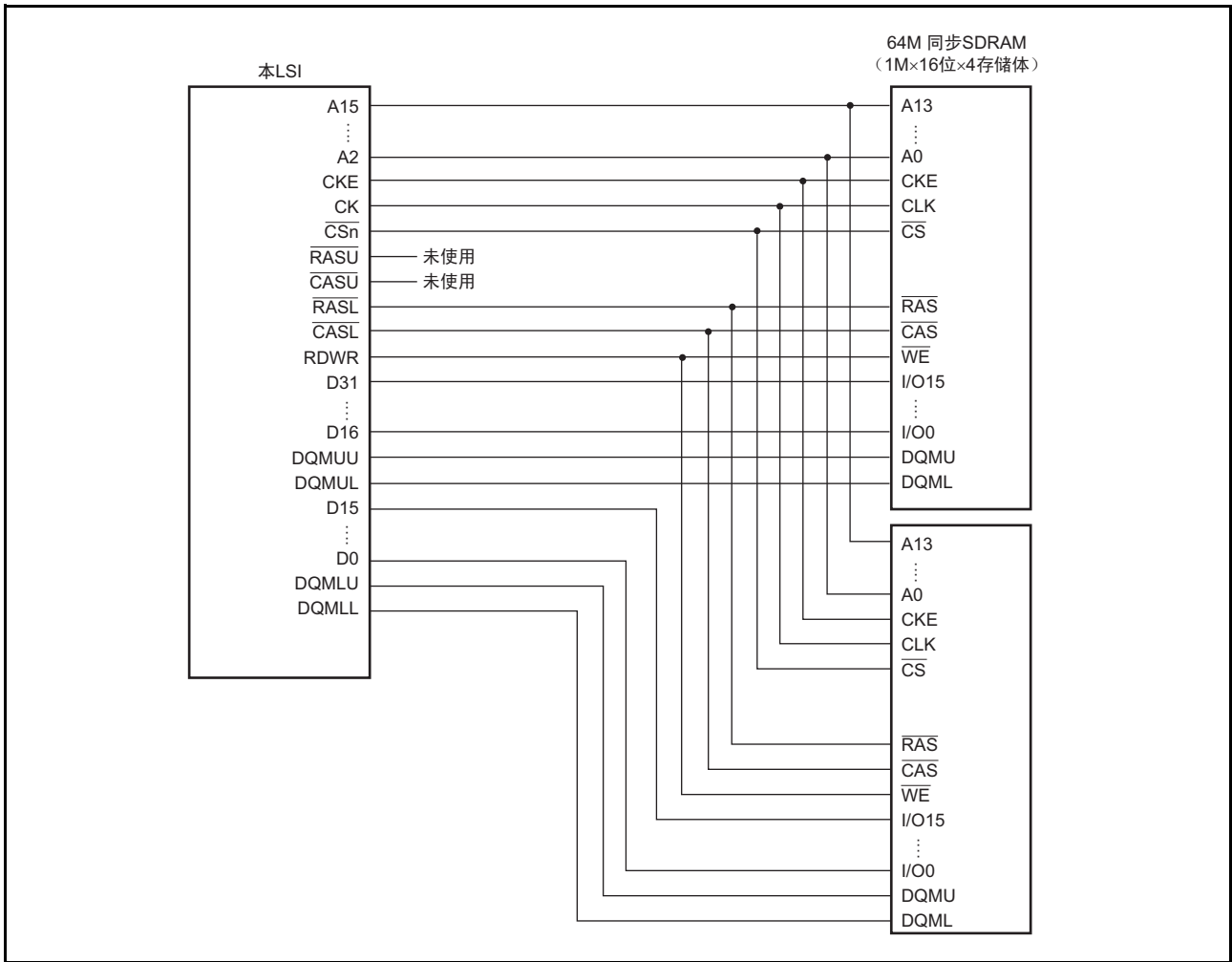


图 9.14 32 位数据的 SDRAM 连接例子 (未使用  $\overline{\text{RASU}}$  和  $\overline{\text{CASU}}$ )

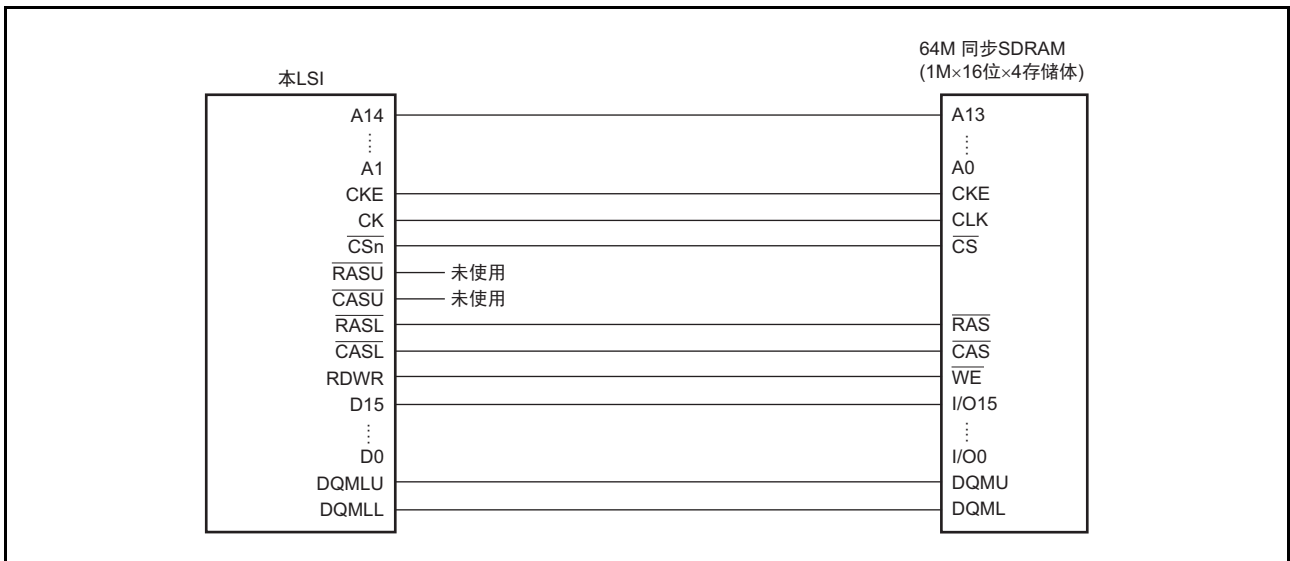
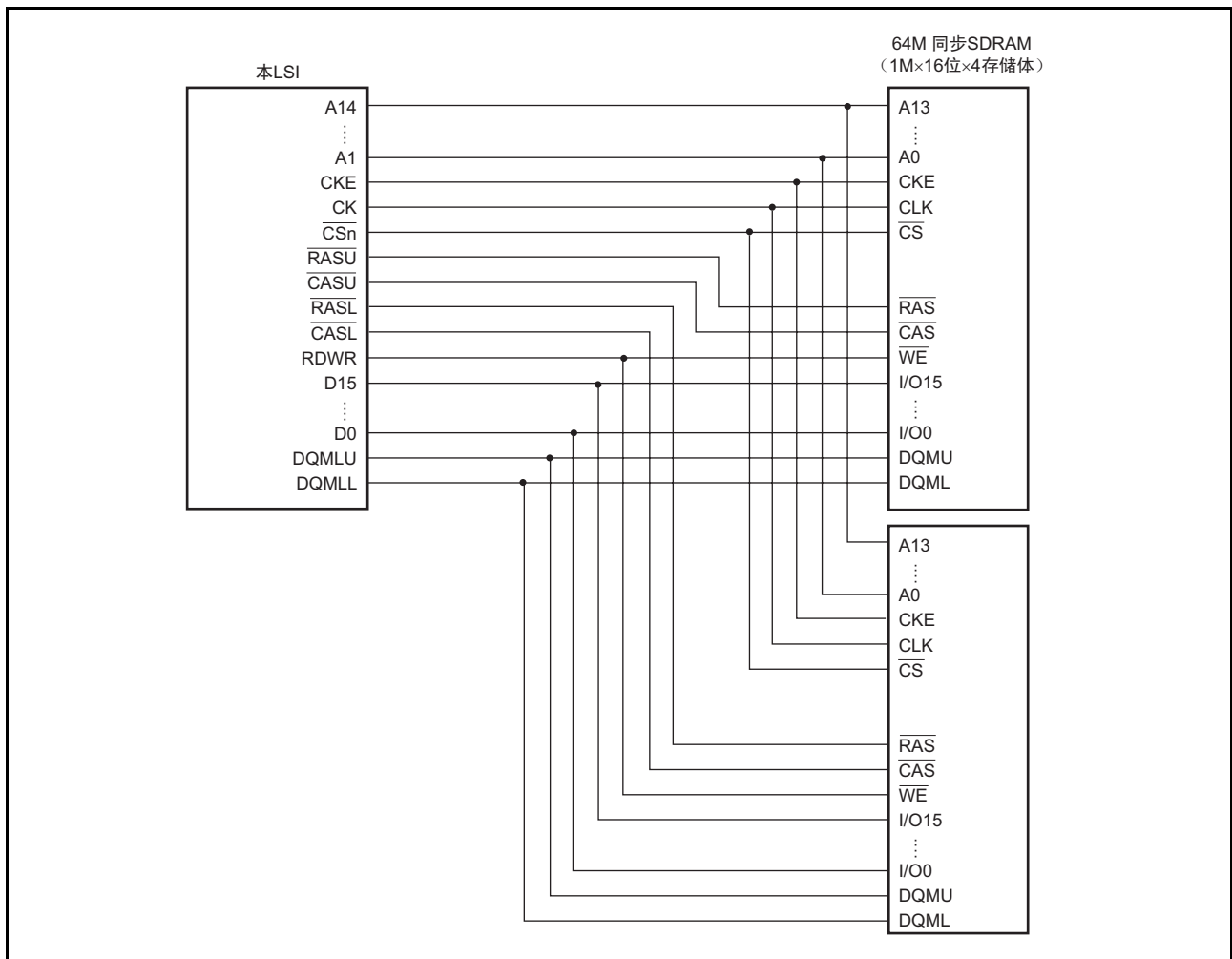


图 9.15 16 位数据的 SDRAM 连接例子 (未使用  $\overline{\text{RASU}}$  和  $\overline{\text{CASU}}$ )

图 9.16 16 位数据的 SDRAM 连接例子 (使用  $\overline{\text{RASU}}$  和  $\overline{\text{CASU}}$ )

## (2) 地址的多路复用

根据  $\text{CSnBCR}$  的  $\text{BSZ}[1:0]$  位、 $\text{SDCR}$  的  $\text{AxROW}[1:0]$  位和  $\text{AxCOL}[1:0]$  位的设定进行地址的多路复用，此时不需要外接地址多路复用的电路就能连接 SDRAM。 $\text{BSZ}[1:0]$ 、 $\text{AxROW}[1:0]$ 、 $\text{AxCOL}[1:0]$  的设定和输出到地址引脚的位的关系如表 9.20 ~ 表 9.25 所示。不能进行此表以外的设定，否则就不保证运行。 $\text{A25} \sim \text{A18}$  不进行多路复用，总是输出原地址。

当数据总线宽度为 16 位 ( $\text{BSZ}[1:0]=\text{B}'10$ ) 时，SDRAM 的 A0 引脚指定字地址，因此必须将 SDRAM 的 A0 引脚连接到本 LSI 的 A1 引脚，SDRAM 的 A1 引脚连接到本 LSI 的 A2 引脚，后面的引脚连接以此类推。当数据总线宽度为 32 位 ( $\text{BSZ}[1:0]=\text{B}'11$ ) 时，SDRAM 的 A0 引脚指定长字地址，因此必须将 SDRAM 的 A0 引脚连接到本 LSI 的 A2 引脚，SDRAM 的 A1 引脚连接到本 LSI 的 A3 引脚，后面的引脚连接以此类推。



表 9.20 BSZ[1:0]、A2/3ROW[1:0]、A2/3COL[1:0] 和地址多路复用的关系 (1)

设定				
BSZ[1:0]	A2/ 3ROW[1:0]	A2/ 3COL[1:0]		
11 (32位)	00 (11位)	00 (8位)		
本 LSI 的 输出引脚	行地址的 输出周期	列地址的 输出周期	同步 DRAM 的 引脚	功能
A17	A25	A17		未使用
A16	A24	A16		
A15	A23	A15		
A14	A22*2	A22*2	A12(BA1)	指定 存储体
A13	A21*2	A21*2	A11(BA0)	
A12	A20	L/H*1	A10/AP	指定地址 / 预充电
A11	A19	A11	A9	地址 同步 DRAM 的 引脚
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	
A1	A9	A1		未使用
A0	A8	A0		
连接存储器的例子				
64M 位产品 (512K 字 × 32 位 × 4 个存储体、列 8 位产品) 1 个 16M 位产品 (512K 字 × 16 位 × 2 个存储体、列 8 位产品) 2 个				

设定				
BSZ[1:0]	A2/ 3ROW[1:0]	A2/ 3COL[1:0]		
11 (32位)	01 (12位)	00 (8位)		
本 LSI 的 输出引脚	行地址的 输出周期	列地址的 输出周期	同步 DRAM 的 引脚	功能
A17	A24	A17		未使用
A16	A23	A16		
A15	A23*2	A23*2		
A14	A22*2	A22*2	A12(BA0)	
A13	A21	A13	A11	地址
A12	A20	L/H*1	A10/AP	指定地址 / 预充电
A11	A19	A11	A9	地址 同步 DRAM 的 引脚
A10	A18	A10	A8	
A9	A17	A9	A7	
A8	A16	A8	A6	
A7	A15	A7	A5	
A6	A14	A6	A4	
A5	A13	A5	A3	
A4	A12	A4	A2	
A3	A11	A3	A1	
A2	A10	A2	A0	
A1	A9	A1		未使用
A0	A8	A0		
连接存储器的例子				
128M 位产品 (1M 字 × 32 位 × 4 个存储体、列 8 位产品) 1 个 64M 位产品 (1M 字 × 16 位 × 4 个存储体、列 8 位产品) 2 个				

【注】 \*1 L/H 是用于指定命令的位，根据存取模式固定为低电平或者高电平。

\*2 指定存储体的地址。

表 9.21 BSZ[1:0]、A2/3ROW[1:0]、A2/3COL[1:0] 和地址多路复用的关系 (2)

设定				
BSZ[1:0]	A2/ 3ROW[1:0]	A2/ 3COL[1:0]		
11 (32 位)	01 (12 位)	01 (9 位)		
本 LSI 的 输出引脚	行地址的 输出周期	列地址的 输出周期	同步 DRAM 的 引脚	功能
A17	A26	A17		未使用
A16	A25	A16		
A15	A24*2	A24*2	A13(BA1)	指定 存储体
A14	A23*2	A23*2	A12(BA0)	
A13	A22	A13	A11	地址
A12	A21	L/H*1	A10/AP	指定地址 / 预充电
A11	A20	A11	A9	地址
A10	A19	A10	A8	
A9	A18	A9	A7	
A8	A17	A8	A6	
A7	A16	A7	A5	
A6	A15	A6	A4	
A5	A14	A5	A3	
A4	A13	A4	A2	
A3	A12	A3	A1	
A2	A11	A2	A0	
A1	A10	A1		未使用
A0	A9	A0		
连接存储器的例子				
256M 位产品 (2M 字 × 32 位 × 4 个存储体、列 9 位产品) 1 个 128M 位产品 (2M 字 × 16 位 × 4 个存储体、列 9 位产品) 2 个				

设定				
BSZ[1:0]	A2/ 3ROW[1:0]	A2/ 3COL[1:0]		
11 (32 位)	01 (12 位)	10 (10 位)		
本 LSI 的 输出引脚	行地址的 输出周期	列地址的 输出周期	同步 DRAM 的 引脚	功能
A17	A27	A17		未使用
A16	A26	A16		
A15	A25*2*3	A25*2*3	A13(BA1)	指定 存储体
A14	A24*2	A24*2	A12(BA0)	
A13	A23	A13	A11	地址
A12	A22	L/H*1	A10/AP	指定地址 / 预充电
A11	A21	A11	A9	地址
A10	A20	A10	A8	
A9	A19	A9	A7	
A8	A18	A8	A6	
A7	A17	A7	A5	
A6	A16	A6	A4	
A5	A15	A5	A3	
A4	A14	A4	A2	
A3	A13	A3	A1	
A2	A12	A2	A0	
A1	A11	A1		未使用
A0	A10	A0		
连接存储器的例子				
512M 位产品 (4M 字 × 32 位 × 4 个存储体、列 10 位产品) 1 个 256M 位产品 (4M 字 × 16 位 × 4 个存储体、列 10 位产品) 2 个				

【注】 \*1 L/H 是用于指定命令的的位，根据存取模式固定为低电平或者高电平。  
\*2 指定存储体的地址。  
\*3 因为 A25 指定存储体的地址，所以只有  $\overline{\text{RASL}}$  有效，而  $\overline{\text{RASU}}$  无效。

表 9.22 BSZ[1:0]、A2/3ROW[1:0]、A2/3COL[1:0] 和地址多路复用的关系 (3)

设定					
BSZ[1:0]	A2/3ROW[1:0]	A2/3COL[1:0]			
11 (32 位)	10 (13 位)	01 (9 位)			
输出引脚	输出的 行地址	输出的 列地址	同步 DRAM 的引脚	功能	
A17	A26	A17		未使用	
A16	A25*2*3	A25*2*3	A14(BA1)	指定存储体	
A15	A24*2	A24*2	A13(BA0)		
A14	A23	A14	A12	地址	
A13	A22	A13	A11		
A12	A21	L/H*1	A10/AP	指定地址 / 预充电	
A11	A20	A11	A9	地址	
A10	A19	A10	A8		
A9	A18	A9	A7		
A8	A17	A8	A6		
A7	A16	A7	A5		
A6	A15	A6	A4		
A5	A14	A5	A3		
A4	A13	A4	A2		
A3	A12	A3	A1		
A2	A11	A2	A0		
A1	A10	A1			未使用
A0	A9	A0			
连接存储器的例子					
512M 位产品 (4M 字 × 32 位 × 4 存储体、列 9 位产品) 1 个 256M 位产品 (4M 字 × 16 位 × 4 存储体、列 9 位产品) 2 个					

- 【注】 \*1 L/H 是用于指定命令的位，根据存取模式固定为低电平或者高电平。  
\*2 指定存储体的地址。  
\*3 因为 A25 指定存储体的地址，所以只有  $\overline{\text{RASL}}$  有效，而  $\overline{\text{RASU}}$  无效。

表 9.23 BSZ[1:0]、A2/3ROW[1:0]、A2/3COL[1:0] 和地址多路复用的关系 (4)

设定				
BSZ[1:0]	A2/ 3ROW[1:0]	A2/ 3COL[1:0]		
10 (16 位)	00 (11 位)	00 (8 位)		
本 LSI 的 输出引脚	行地址的 输出周期	列地址的 输出周期	同步 DRAM 的 引脚	功能
A17	A25	A17		未使用
A16	A24	A16		
A15	A23	A15		
A14	A22	A14		
A13	A21*2	A21*2	A12(BA1)	指定存储 体
A12	A20*2	A20*2	A11(BA0)	
A11	A19	L/H*2	A10/AP	指定地址 / 预充电
A10	A18	A10	A9	地址
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		
连接存储器的例子				
16M 位产品 (512K 字 × 16 位 × 2 个存储体、列 8 位产品) 1 个				

设定				
BSZ[1:0]	A2/ 3ROW[1:0]	A2/ 3COL[1:0]		
10 (16 位)	01 (12 位)	00 (8 位)		
本 LSI 的 输出引脚	行地址的 输出周期	列地址的 输出周期	同步 DRAM 的 引脚	功能
A17	A25	A17		未使用
A16	A24	A16		
A15	A23	A15		
A14	A22*2	A22*2		
A13	A21*2	A21*2	A12(BA0)	
A12	A20	A12	A11	地址
A11	A19	L/H*1	A10/AP	指定地址 / 预充电
A10	A18	A10	A9	地址
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		
连接存储器的例子				
64M 位产品 (1M 字 × 16 位 × 4 个存储体、列 8 位产品) 1 个				

【注】 \*1 L/H 是用于指定命令的位，根据存取模式固定为低电平或者高电平。  
\*2 指定存储体的地址。

表 9.24 BSZ[1:0]、A2/3ROW[1:0]、A2/3COL[1:0] 和地址多路复用的关系 (5)

设定				
BSZ[1:0]	A2/ 3ROW[1:0]	A2/ 3COL[1:0]		
10 (16 位)	01 (12 位)	01 (9 位)		
本 LSI 的 输出引脚	行地址的 输出周期	列地址的 输出周期	同步 DRAM 的 引脚	功能
A17	A26	A17		未使用
A16	A25	A16		
A15	A24	A15		
A14	A23*2	A23*2	A13(BA1)	指定 存储体
A13	A22*2	A22*2	A12(BA0)	
A12	A21	A12	A11	地址
A11	A20	L/H*1	A10/AP	指定地址 / 预充电
A10	A19	A10	A9	地址
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		未使用
连接存储器的例子				
128M 位产品 (2M 字 × 16 位 × 4 个存储体、列 9 位产品) 1 个				

设定				
BSZ[1:0]	A2/ 3ROW[1:0]	A2/ 3COL[1:0]		
10 (16 位)	01 (12 位)	10 (10 位)		
本 LSI 的 输出引脚	行地址的 输出周期	列地址的 输出周期	同步 DRAM 的 引脚	功能
A17	A27	A17		未使用
A16	A26	A16		
A15	A25	A15		
A14	A24*2	A24*2	A13(BA1)	指定存储 体
A13	A23*2	A23*2	A12(BA0)	
A12	A22	A12	A11	地址
A11	A21	L/H*1	A10/AP	指定地址 / 预充电
A10	A20	A10	A9	地址
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		未使用
连接存储器的例子				
256M 位产品 (4M 字 × 16 位 × 4 个存储体、列 10 位产品) 1 个				

【注】 \*1 L/H 是用于指定命令的位，根据存取模式固定为低电平或者高电平。

\*2 指定存储体的地址。

表 9.25 BSZ[1:0]、A2/3ROW[1:0]、A2/3COL[1:0] 和地址多路复用的关系 (6)

设定				
BSZ[1:0]	A2/ 3ROW[1:0]	A2/ 3COL[1:0]		
10 (16 位)	10 (13 位)	01 (9 位)		
本 LSI 的 输出引脚	行地址的 输出周期	列地址的 输出周期	同步 DRAM 的 引脚	功能
A17	A26	A17		未使用
A16	A25	A16		
A15	A24*2	A24*2	A14(BA1)	指定存储 体
A14	A23*2	A23*2	A13(BA0)	
A13	A22	A13	A12	地址
A12	A21	A12	A11	
A11	A20	L/H*1	A10/AP	指定地址 / 预充电
A10	A19	A10	A9	地址
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		未使用
连接存储器的例子				
256M 位产品 (4M 字 × 16 位 × 4 个存储体、列 9 位产品) 1 个				

设定				
BSZ[1:0]	A2/ 3ROW[1:0]	A2/ 3COL[1:0]		
10 (16 位)	10 (13 位)	10 (10 位)		
本 LSI 的 输出引脚	行地址的 输出周期	列地址的 输出周期	同步 DRAM 的 引脚	功能
A17	A27	A17		未使用
A16	A26	A16		
A15	A25*2*3	A25*2*3	A14(BA1)	指定存储 体
A14	A24*2	A24*2	A13(BA0)	
A13	A23	A13	A12	地址
A12	A22	A12	A11	
A11	A21	L/H*1	A10/AP	指定地址 / 预充电
A10	A20	A10	A9	地址
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		未使用
连接存储器的例子				
512M 位产品 (8M 字 × 16 位 × 4 个存储体、列 10 位产品) 1 个				

【注】 \*1 L/H 是用于指定命令的位，根据存取模式固定为低电平或者高电平。  
 \*2 指定存储体的地址。  
 \*3 因为 A25 指定存储体的地址，所以只有  $\overline{\text{RASL}}$  有效，而  $\overline{\text{RASU}}$  无效。

## (3) 突发读

本 LSI 产生突发读的条件如下：

1. 当读的存取长度大于数据总线宽度时
2. 当通过 DMAC 进行 16 字节传送时

本 LSI 总是以突发长度为 1 存取 SDRAM。例如，当从连接 32 位数据总线的 SDRAM 连续读取 16 字节的数据时，连续进行 4 次突发长度为 1 的读操作，此时的存取称为突发数 4。存取长度和突发数的关系如表 9.26 所示。

表 9.26 存取长度和突发数的关系

总线宽度	存取长度	突发数
16 位	8 位	1
	16 位	1
	32 位	2
	16 字节	8
32 位	8 位	1
	16 位	1
	32 位	1
	16 字节	4

突发读时的时序图如图 9.17 和图 9.18 所示。在突发读时序中，在  $T_r$  周期输出 ACTV 命令、在  $T_{c1}$ 、 $T_{c2}$ 、 $T_{c3}$  周期产生 READ 命令、在  $T_{c4}$  周期产生 READA 命令，并且在从  $T_{d1}$  到  $T_{d4}$  的周期中的外部时钟 (CK) 的上升沿接收读数据。 $T_{ap}$  周期是等待 SDRAM 内部自动预充电结束 (由 READA 命令引起的) 的周期，在此期间不对同一存储体产生新命令，但是可存取不同 CS 空间或者同一 SDRAM 的不同存储体。通过指定 CS3WCR 的 WTRP1 和 WTRP0 决定  $T_{ap}$  的周期数。

本 LSI 为了用各种频率连接 SDRAM，能通过设定 CSnWCR 的各个位插入等待周期。各种等待的设定例子如图 9.18 所示。能通过 CS3WCR 的 WTRCD1 和 WTRCD0 位指定从 ACTV 命令输出周期  $T_r$  到 READA 命令输出周期  $T_{c1}$  之间的周期数。如果将 WTRCD1 和 WTRCD0 至少设定为 1 个周期，就在  $T_r$  周期和  $T_{c1}$  周期之间插入 NOP 命令产生周期的  $T_{rw}$  周期。能通过 CS2WCR 的 A2CL1 和 A2CL0 位以及 CS3WCR 的 A3CL1 位和 A3CL0 位，在 CS2 和 CS3 空间内分别独立指定从 READA 命令输出周期  $T_{c1}$  到读数据取周期  $T_{d1}$  之间的周期数。此周期数相当于同步 DRAM 的 CAS 潜伏时间。同步 DRAM 的 CAS 潜伏时间最多为 3 个周期，但是本 LSI 能设定 1 ~ 4 个周期，这是为了在本 LSI 和同步 DRAM 之间连接含锁存器的电路。

$T_{de}$  周期是将读数据传送到本 LSI 内部时所需的空闲周期，在进行突发读或者单次读时，总是产生 1 个周期。

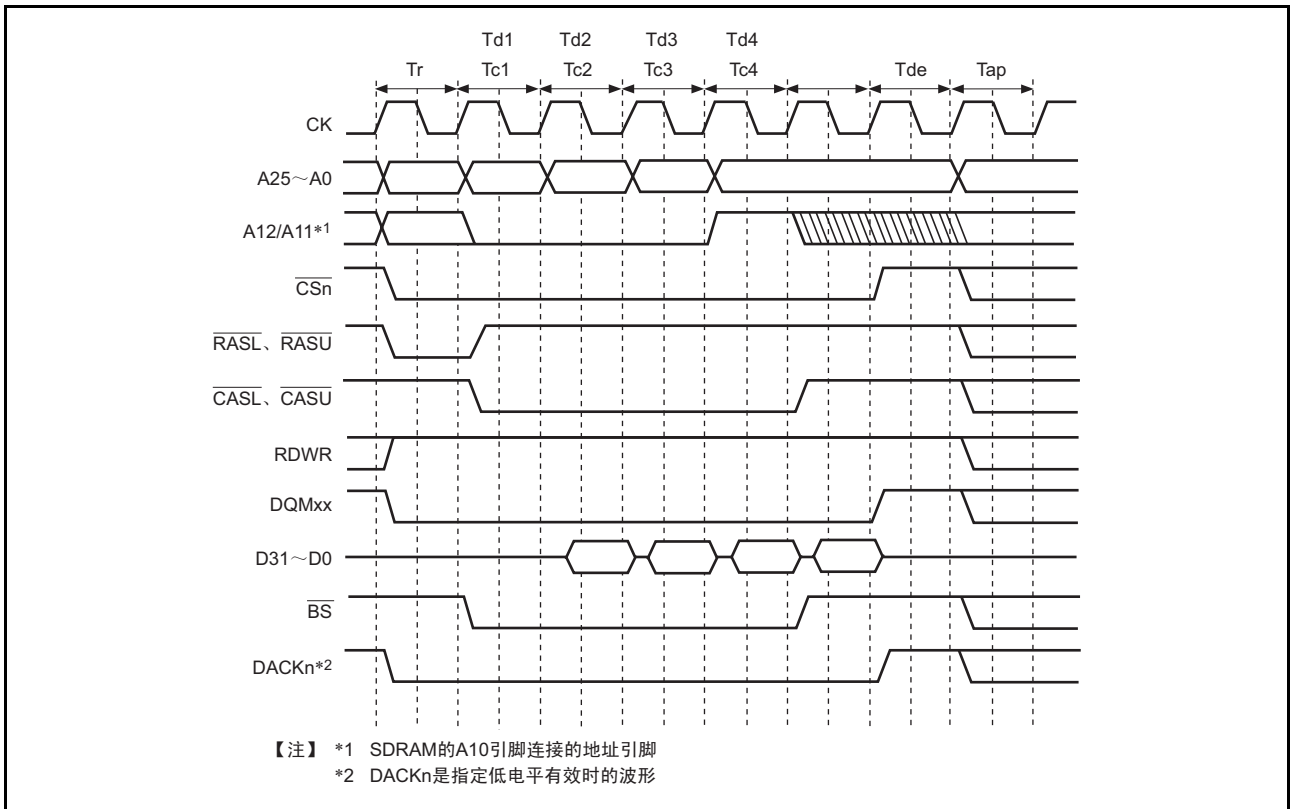


图 9.17 突发读的基本时序 (自动预充电)

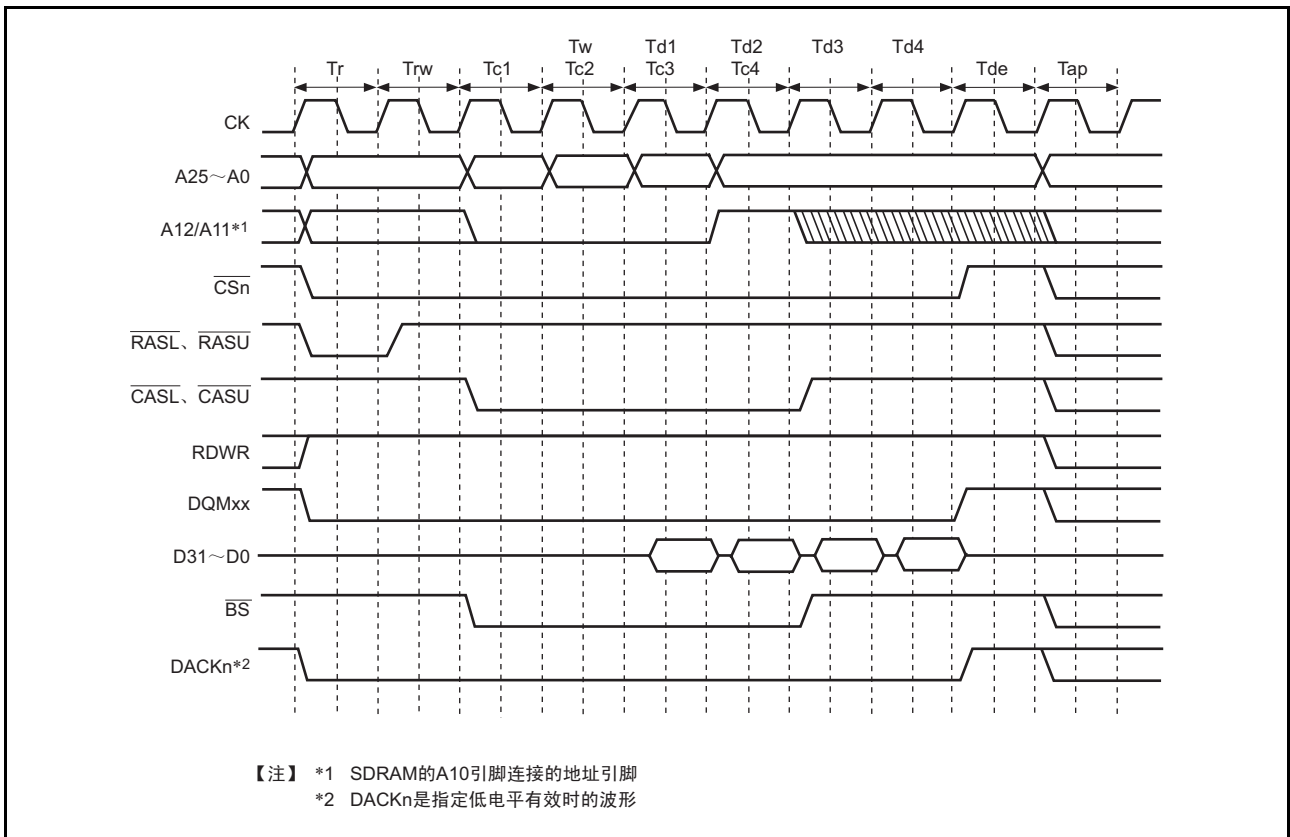


图 9.18 突发读的等待指定时序 (自动预充电)



(4) 单次读

当数据总线宽度不小于存取长度时，进行 1 次读存取后就结束，称之为单次读。因为在突发读 / 单次写模式中突发长度设定为 1，所以 SDRAM 只输出需要的数据而不产生多余的总线周期。

单次读的基本时序图如图 9.19 所示。

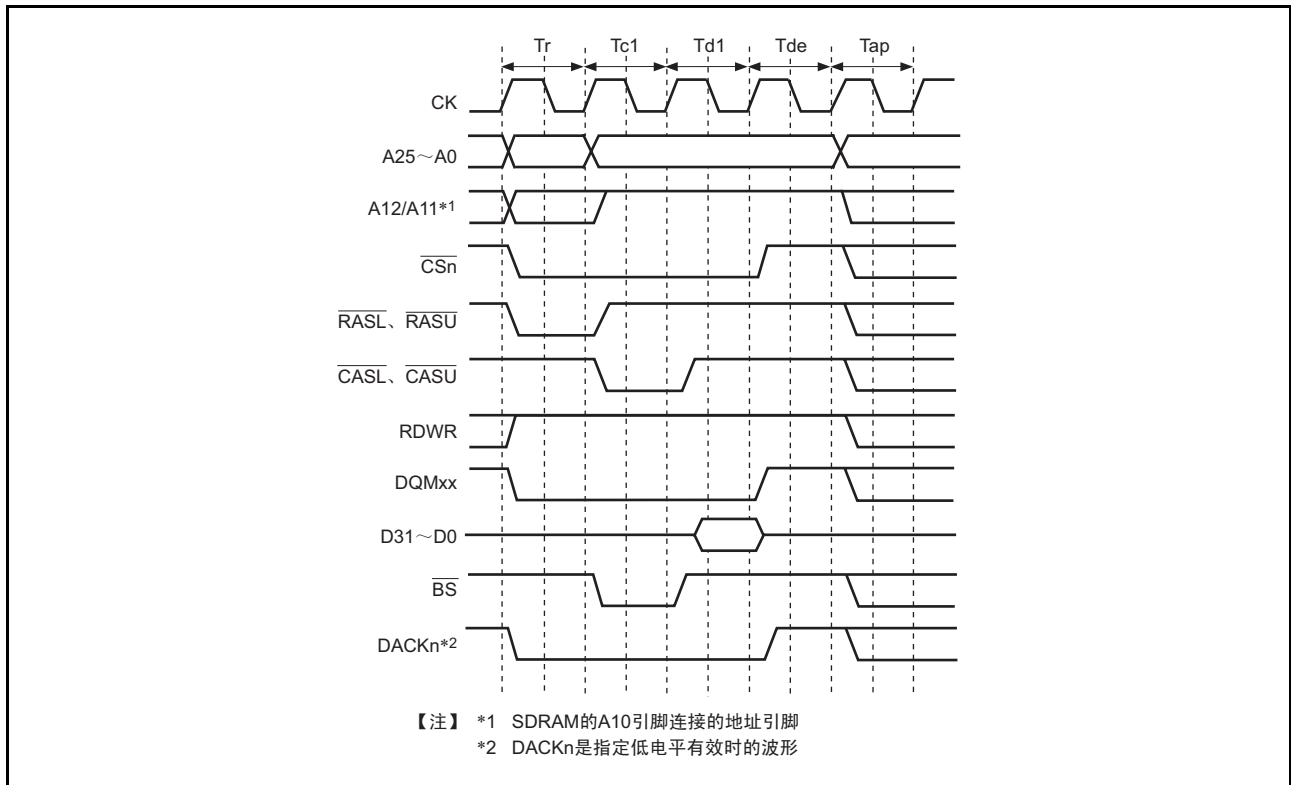


图 9.19 单次读的基本时序 (自动预充电)

## (5) 突发写

LSI 产生突发写的条件如下：

1. 当写的存取长度大于数据总线宽度时
2. 当通过 DMAC 进行 16 字节传送时

本 LSI 总是以突发长度为 1 存取 SDRAM。例如，当从连接 32 位数据总线的 SDRAM 连续写 16 字节的数据时，连续进行 4 次突发长度为 1 的写操作。存取长度和突发数的关系如表 9.26、突发写的时序图如图 9.20 所示。在突发写时序中，在  $T_r$  周期输出  $\overline{\text{ACTV}}$  命令、在  $T_{c1}$ 、 $T_{c2}$ 、 $T_{c3}$  周期产生  $\overline{\text{WRIT}}$  命令、在  $T_{c4}$  周期产生进行自动预充电的  $\overline{\text{WRITA}}$  命令。在写周期中，同时输出写数据和写命令，在输出带自动预充电的写命令后，连续存在  $T_{rw1}$  周期（等待自动预充电的启动）和  $T_{ap}$  周期（等待自动预充电结束）。 $T_{ap}$  周期是等待 SDRAM 内部自动预充电结束（由  $\overline{\text{WRITA}}$  命令引起的）的周期。在  $T_{rw1}$  周期和  $T_{ap}$  周期之间不对同一存储体产生新命令，但是可存取不同 CS 空间或者同一 SDRAM 的不同存储体。通过指定 CS3WCR 的  $\text{TRWL1}$  位和  $\text{TRWL0}$  位决定  $T_{rw1}$  周期，通过指定 CS3WCR 的  $\text{WTRP1}$  位和  $\text{WTRP0}$  位决定  $T_{ap}$  周期。

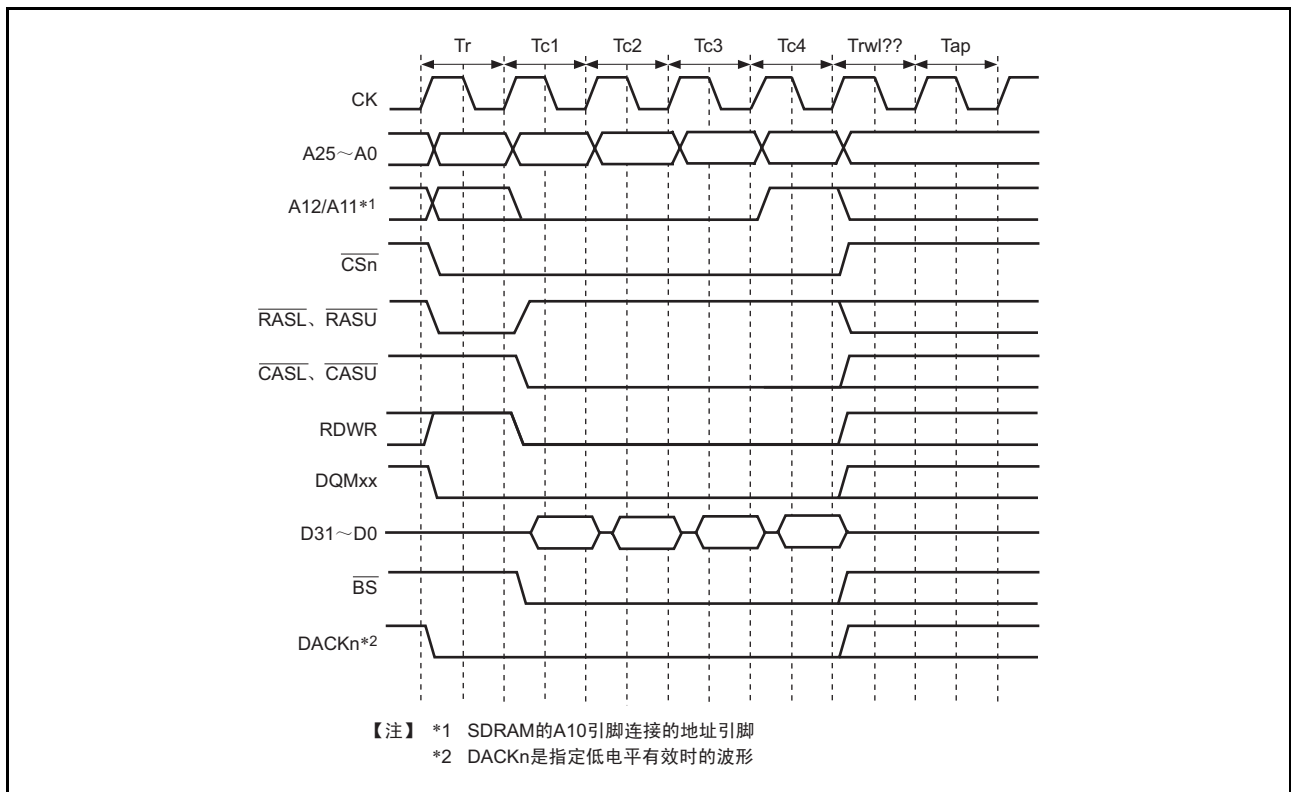


图 9.20 突发写的基本时序（自动预充电）

(6) 单次写

当数据总线宽度不小于存取长度时, 进行 1 次写存取后就结束, 称之为单次写。单次写的基本时序图如图 9.21 所示。

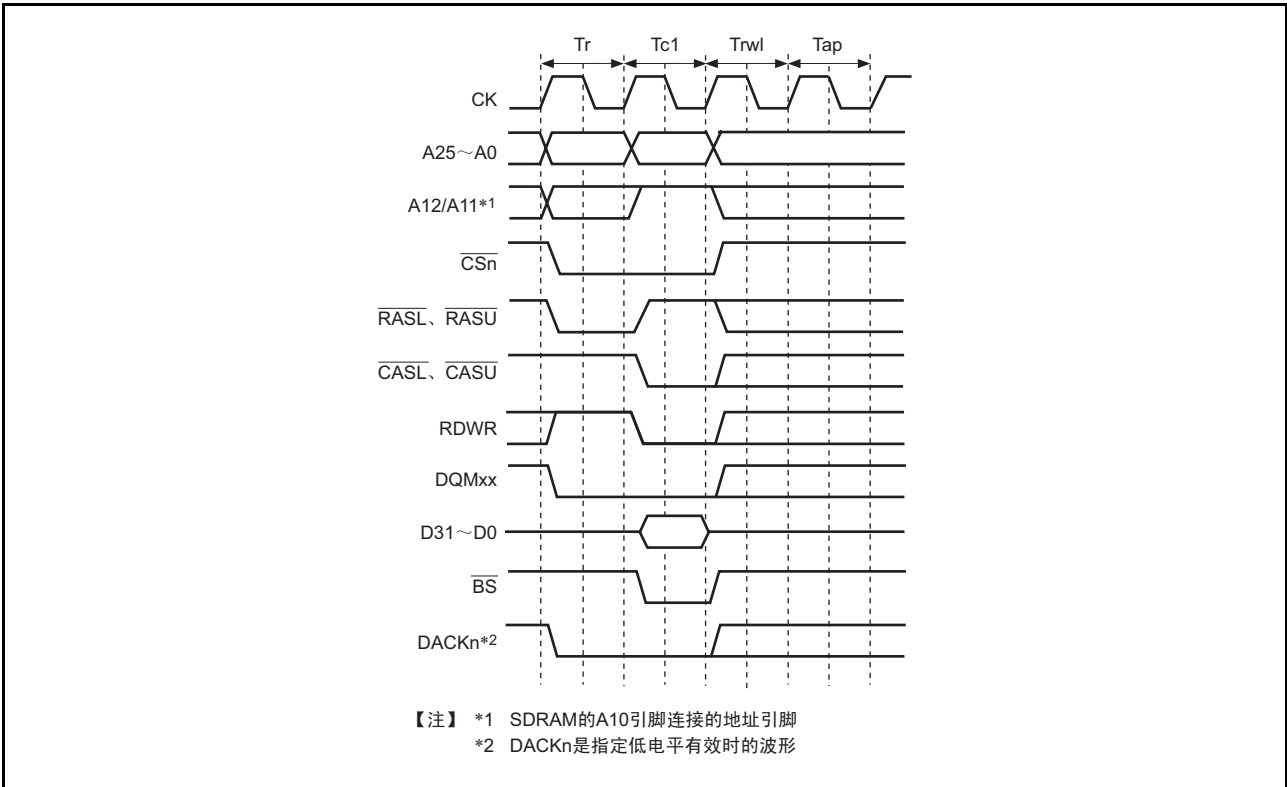


图 9.21 单次写的基本时序 (自动预充电)

### (7) 存储体激活

能使用 SDRAM 的存储体功能对同一行地址执行高速存取。当 SDCR 的 BACTV 位是 1 时，使用不自动预充电的命令 (READ 或者 WRIT) 进行存取，称之为存储体激活功能。但是，存储体激活功能只对区域 3 有效。当区域 3 设定为存储体激活模式时，必须将区域 2 空间设定为通常空间或者带字节选择的 SRAM；当区域 2 和区域 3 都设定为 SDRAM 时，必须设定为自动预充电模式。

当使用存储体激活功能时，即使存取结束也不进行预充电。如果存取同一存储体的同一行地址，就能不产生 ACTV 命令而直接产生 READ 命令或者 WRIT 命令。因为 SDRAM 内部被分成了多个存储体，所以能预先将各存储体的每个行地址设定为激活状态。如果下一次存取的行地址是不同的行地址，就先产生 PRE 命令进行相应存储体的预充电，在预充电结束后按顺序产生 ACTV 命令、READ 命令或者 WRIT 命令。当继续存取不同的行地址时，有可能因为在发生存取请求后进行预充电，反而延长了存取时间。通过 CS3WCR 的 WTRP[1:0] 位指定从产生 PRE 命令到产生 ACTV 命令的周期数。

如果在写操作时进行自动预充电，就不能在产生 WRITA 命令后的  $Trwl+Tap$  周期内对同一存储体产生命令。当使用存储体激活模式时，如果是同一行地址就能连续产生 READ 命令或者 WRIT 命令，因此每写 1 次就能缩短  $Trwl+Tap$  周期的周期数。

各存储体处于激活状态的时间 ( $tRAS$ ) 有限制。在程序执行中，如果不能保证在维持此限制的周期中存取不同的行地址，就不能将刷新周期设定为大于  $tRAS$ 。

不自动预充电的突发读周期、同一行地址的突发读周期和不同行地址的突发读周期分别如图 9.22、图 9.23 和图 9.24 所示，不自动预充电的单次写周期、同一行地址的单次写周期和不同行地址的单次写周期分别如图 9.25、图 9.26 和图 9.27 所示。

在图 9.23 中，在产生 READ 命令的  $Tc$  周期前插入不进行任何操作的  $Tnop$  周期。这是为了在从 SDRAM 读数据时，对于指定读字节的  $DQMxx$  信号保持 2 个周期的潜伏时间。在 CAS 潜伏时间不小于 2 个周期的情况下，因为已保持 2 个周期的潜伏时间，所以即使在  $Tc$  周期以后使  $DQMxx$  信号有效也不插入  $Tnop$  周期。

只从存取被设定为存储体激活功能的空间的各存储体来看，只要连续存取同一行地址，就以图 9.22 或者图 9.25 开始，重复图 9.23 或者图 9.26。在此之间，即使存取不同空间或者不同存储体也不影响。如果在存储体激活模式中存取不同的行地址，替代图 9.23 或者图 9.26 而执行图 9.24 或者图 9.27 的总线周期。即使在存储体激活模式中，也在刷新周期或者通过总线仲裁释放总线后，全部存储体都为非激活状态。

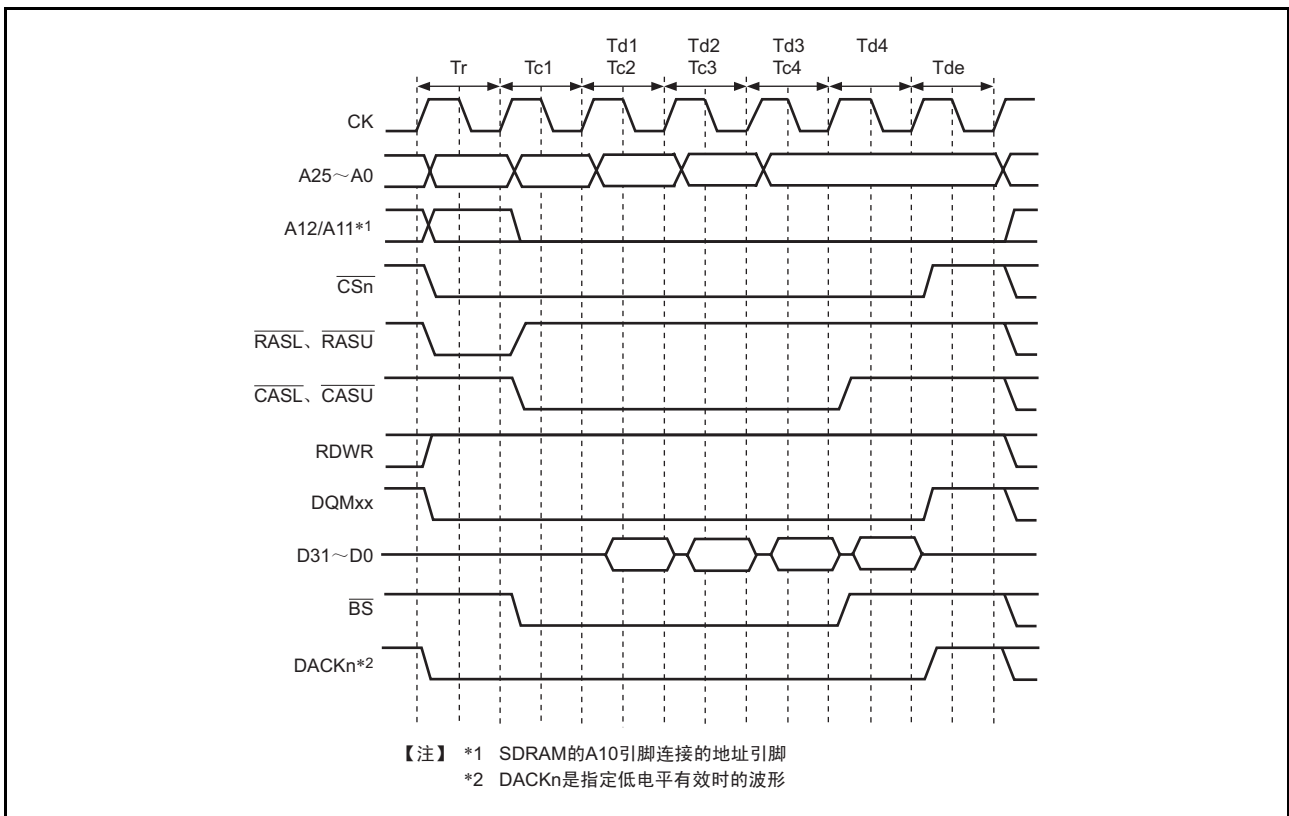


图 9.22 突发读的时序（不自动预充电）

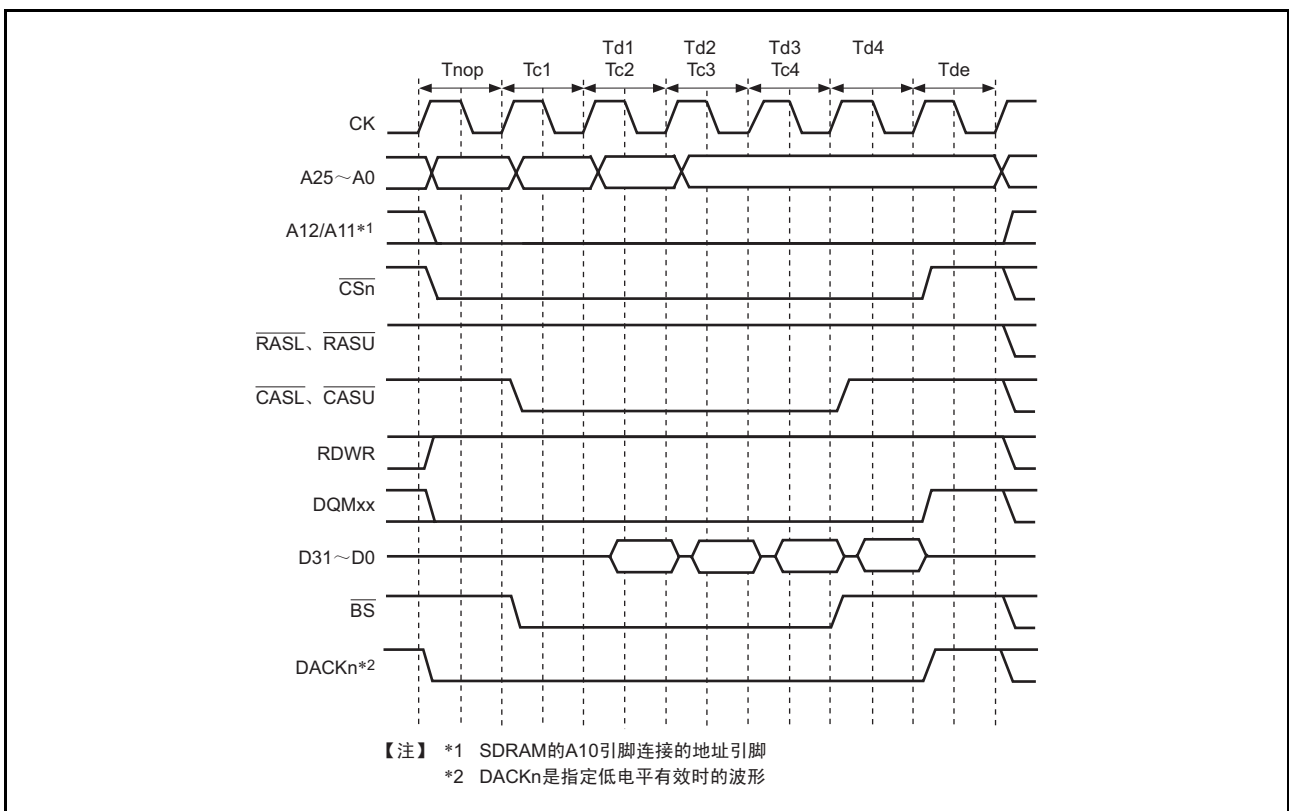


图 9.23 突发读的时序（存储体激活、同一行地址）

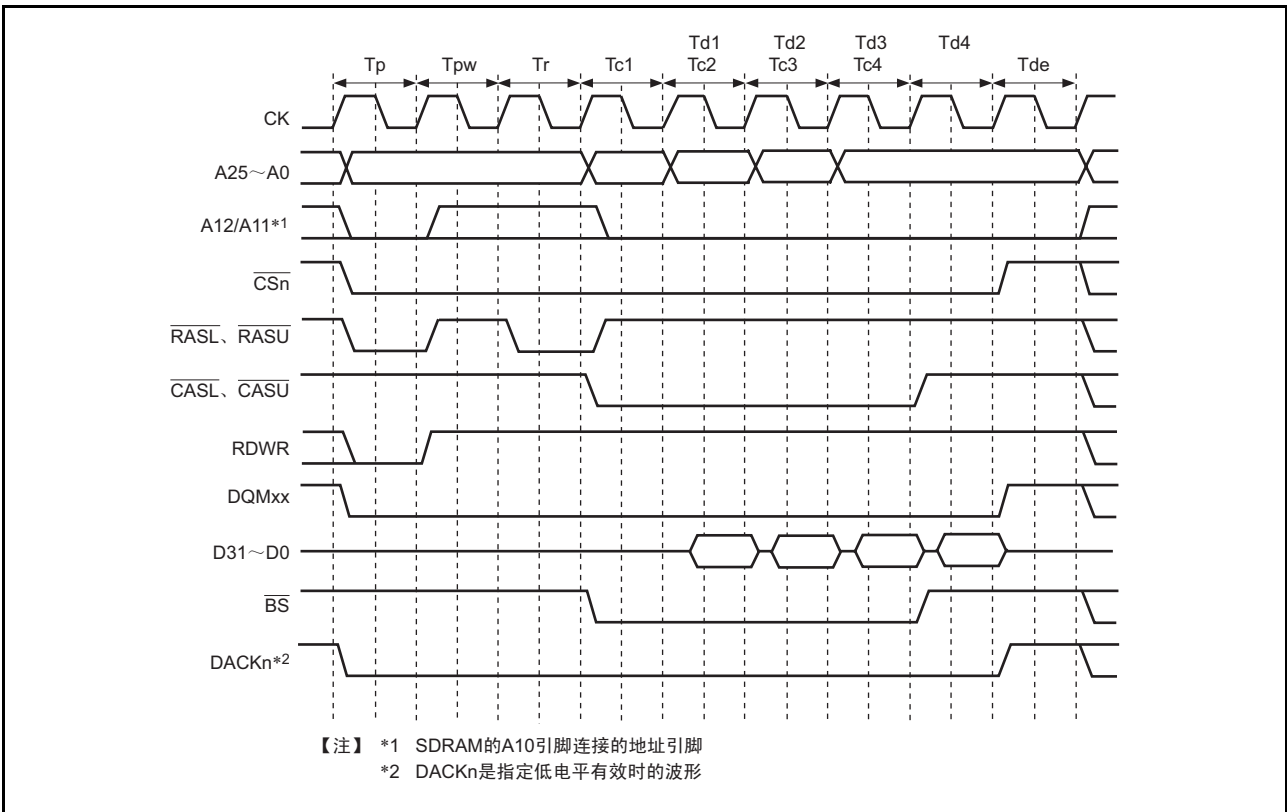


图 9.24 突发读的时序 (存储体激活、不同行地址)

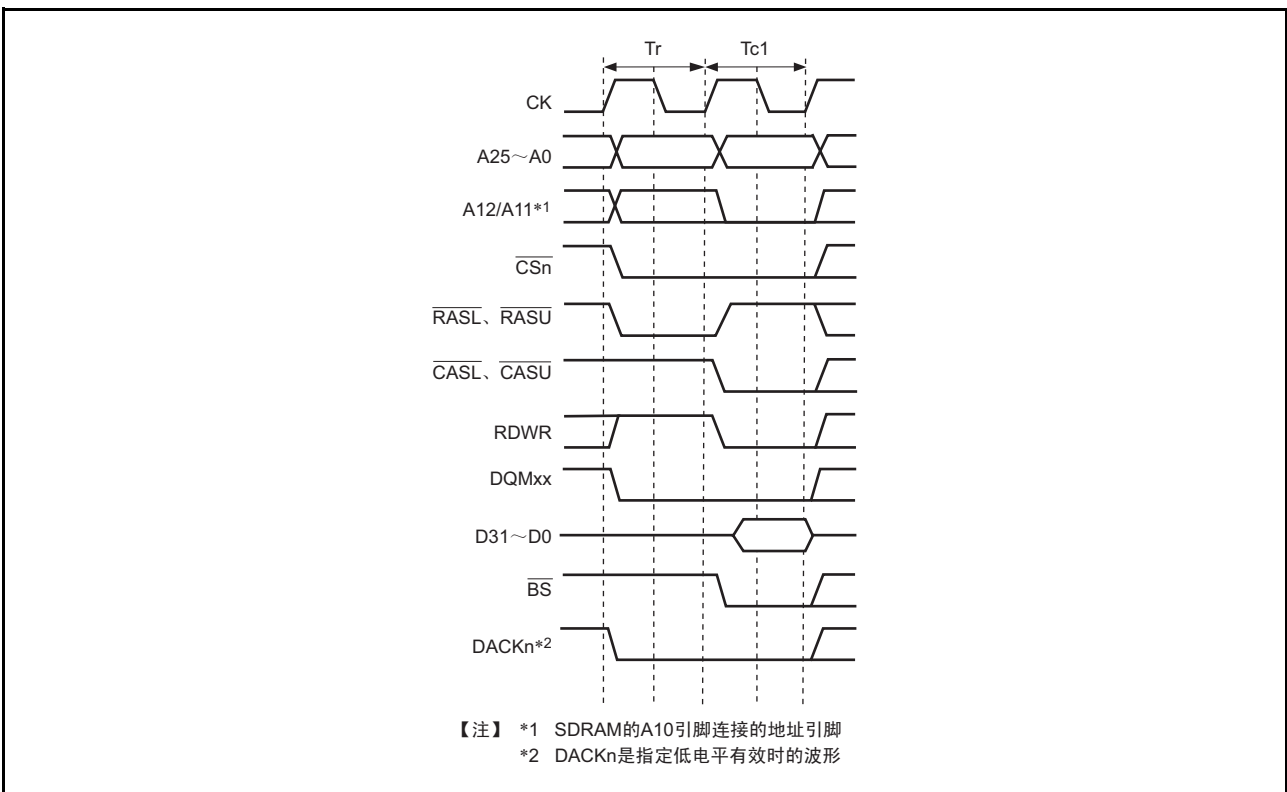


图 9.25 单次写的时序 (不自动预充电)

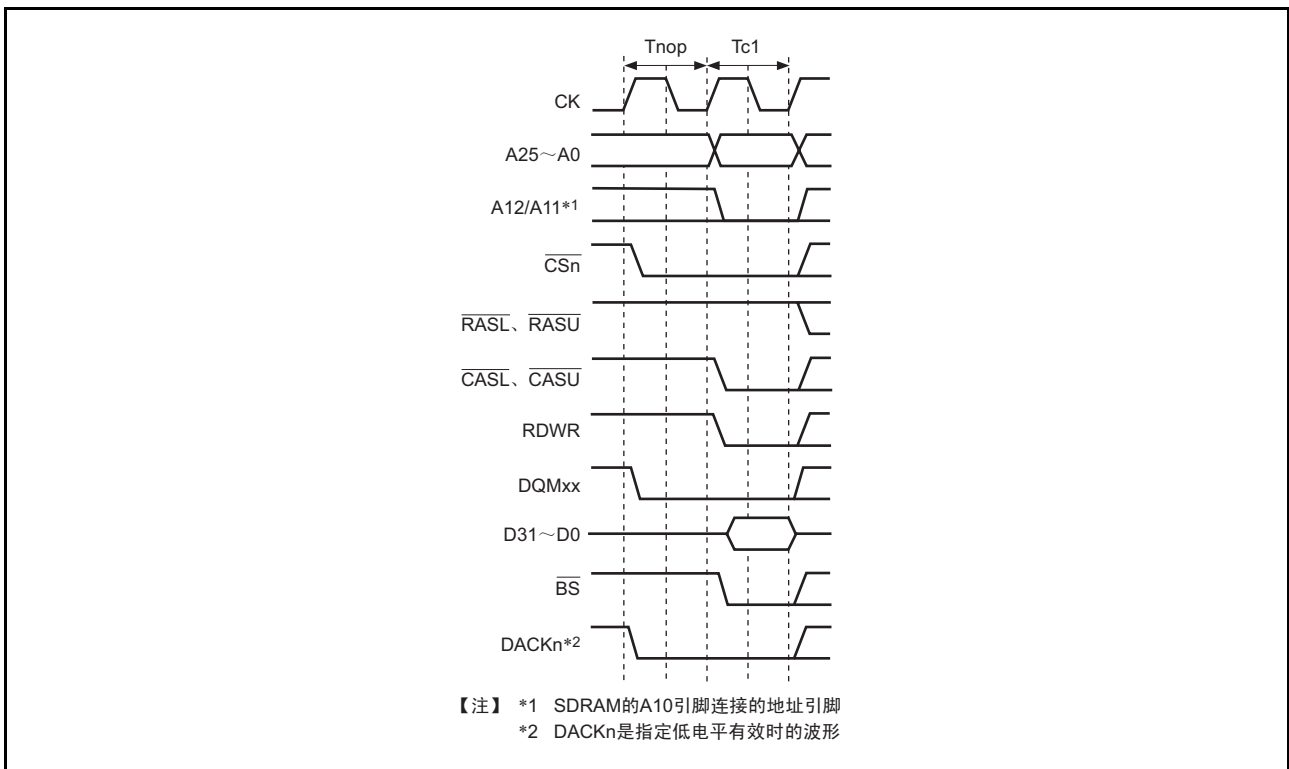


图 9.26 单次写的时序（存储体激活、同一行地址）

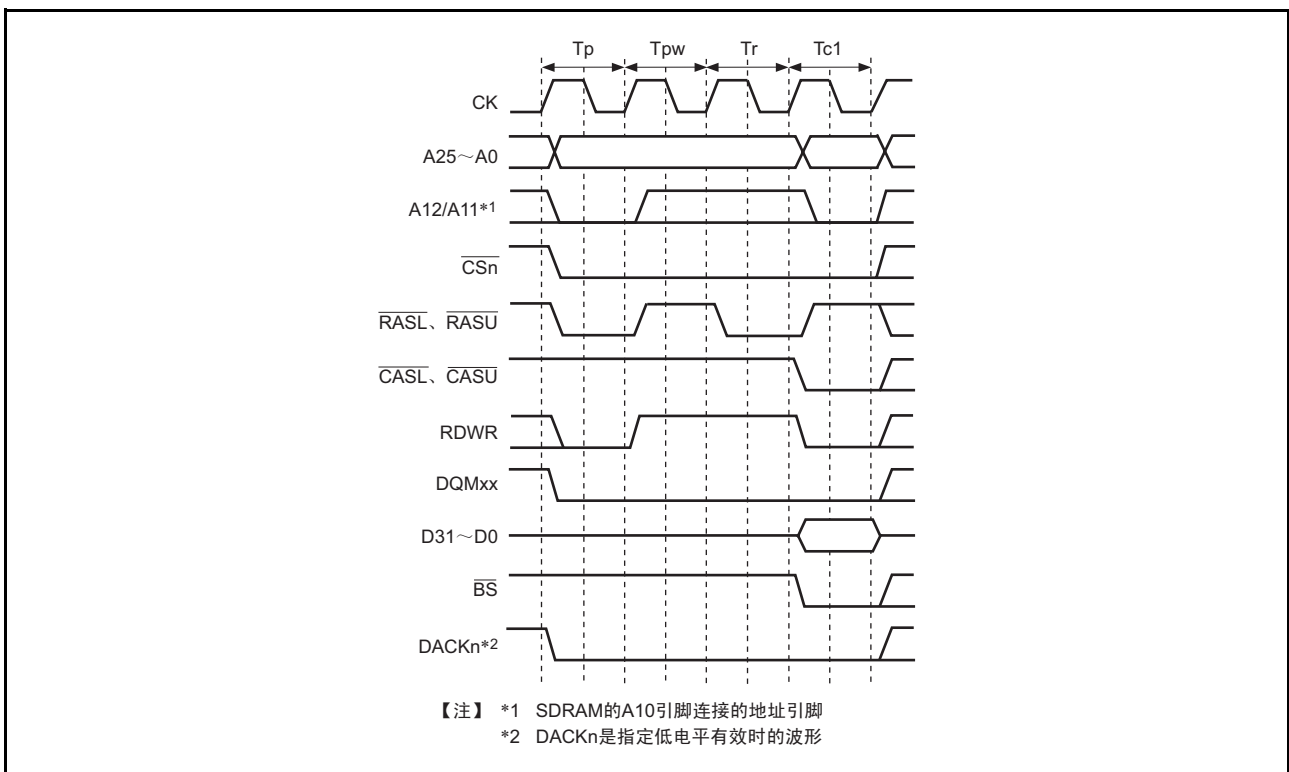


图 9.27 单次写的时序（存储体激活、不同行地址）

## (8) 刷新

BSC 具有控制 SDRAM 刷新的功能。能通过将 SDCR 的 RFSH 位和 RMODE 位分别设定为 1 和 0 进行自动刷新，还能通过设定 RTCSR 的 RRC[2:0] 位进行连续刷新。另外，在长时间不存取 SDRAM 时，能通过将 RFSH 位和 RMODE 位都设定为 1，启动功耗少的自刷新。

## (a) 自动刷新

以 RTCSR 的 CKS[2:0] 位选择的输入时钟和 RTCOR 的设定值所定的间隔，进行 RTCSR 的 RRC[2:0] 设定次数的刷新。为了符合所使用的 SDRAM 的刷新闻隔规定，必须设定各寄存器，顺序为先设定 RTCOR、RTCNT、SDCR 的 RFSH 位和 RMODE 位，然后设定 RTCSR 的 CKS[2:0] 位和 RRC[2:0] 位。如果通过 CKS[2:0] 选择输入时钟，RTCNT 就从当时的值开始递增计数。RTCNT 的值随时和 RTCOR 的值进行比较，如果两者的值一致，就产生刷新请求并执行 RRC[2:0] 设定次数的自动刷新，同时将 RTCNT 清 0，重新开始递增计数。

自动刷新周期的时序如图 9.28 所示。如果在启动自动刷新时存在正在预充电中的存储体，就等待其结束，然后在  $T_p$  周期产生 PALL 命令，使全部的存储体从激活状态进入预充电状态。接着，插入 CS3WCR 的 WTRP[1:0] 位所设个数的空闲周期，然后在  $T_{rr}$  周期中产生 REF 命令。在  $T_{rr}$  周期之后和 CS3WCR 的 WTRC[1:0] 位指定的周期数之间不产生新命令。为了符合 SDRAM 刷新周期时间的规定 ( $t_{RC}$ )，需要设定 WTRC[1:0]。如果 CS3WCR 的 WTRP[1:0] 位的设定值至少是 1 个周期，就在  $T_p$  周期和  $T_{rr}$  周期之间插入空闲周期。

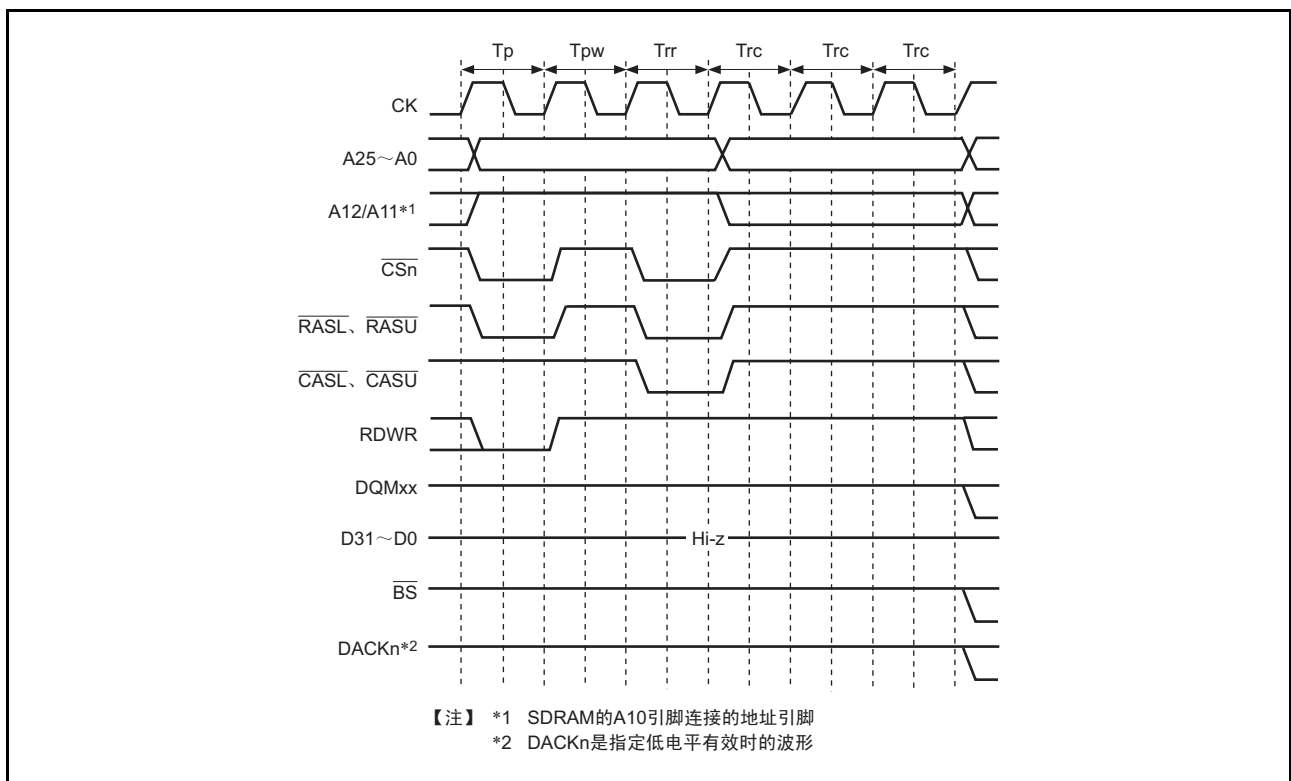


图 9.28 自动刷新的时序



## (b) 自刷新

自刷新是一种在 SDRAM 内部生成刷新时序和刷新地址的待机模式，通过将 SDCR 的 RFSH 位和 RMODE 位都设定为 1 进行启动。如果在启动自刷新时存在正在预充电中的存储体，就等待自刷新结束，然后在  $T_p$  周期中产生 PALL 命令。接着，在插入 CS3WCR 的 WTRP[1:0] 位设定的空闲周期后产生 SELF 命令。不能在自刷新状态期间存取 SDRAM。通过将 RMODE 位设定为 0 来解除自刷新，如果解除自刷新，就在 CS3WCR 的 WTRC[1:0] 位指定的周期数之间不产生命令。

自刷新的时序如图 9.29 所示。在解除自刷新后，必须立即进行能以正确的间隔执行自动刷新的设定。当从自动刷新状态进入自刷新状态时，如果在解除自刷新时 RFSH=1、RMODE=0，就重新开始自动刷新。在从解除自刷新到开始自动刷新需要时间的情况下，能通过将 (RTCOR 的值 -1) 设定到 RTCNT，立即开始自动刷新。

在设定为自刷新后，本 LSI 在进入待机状态后仍继续保持自刷新状态，并且通过中断从待机状态返回后也保持自刷新状态。但是，即使在待机状态下，也需要将 CMNCR 寄存器的 HIZCNT 位设定为 1，驱动 CKE 及其他引脚。

不能通过手动复位解除自刷新状态。

在进行上电复位时，BSC 的寄存器被初始化，所以解除自刷新状态。

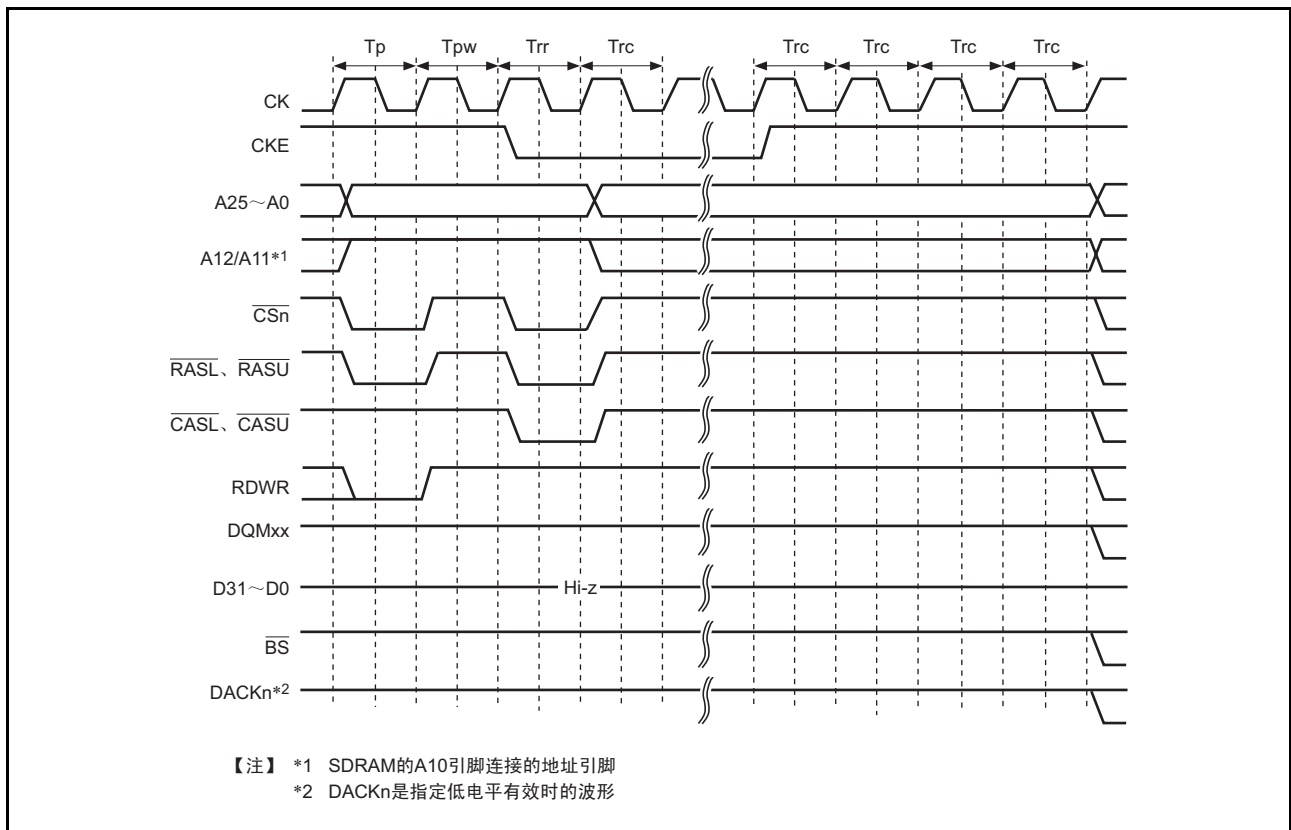


图 9.29 自刷新的时序

## (9) 刷新请求和总线周期的关系

如果在执行总线周期中发生刷新请求，就在等待到总线周期结束后执行刷新；如果在通过总线仲裁功能释放总线时发生刷新请求，就在等待到获得总线权后执行刷新。本 LSI 设有作为在刷新等待执行期间的总线权请求信号的 IRQOUT 引脚，在获得总线权前，IRQOUT 一直有效（低电平）。

如果在刷新等待执行的状态下发生新的刷新请求，就取消前一个刷新请求。为了正确地进行刷新，需要比刷新间隔长的总线周期或者不占有总线权。

即使在自刷新时发生总线权请求，也不在解除自刷新前释放总线。

## (10) 上电顺序

在使用 SDRAM 时，需要在上电后对 SDRAM 进行模式设定。为了正确地初始化 SDRAM，先设定 BSC 的寄存器，然后通过存取 SDMR2 或者 SDMR3 寄存器，写 SDRAM 的模式寄存器。在设定 SDRAM 的模式寄存器时，通过  $\overline{CSn}$ 、 $\overline{RASU}$ 、 $\overline{RASL}$ 、 $\overline{CASU}$ 、 $\overline{CASL}$  和 RDWR 的组合，将当时的地址信号值取到 SDRAM。假设要设定的值为 X，通过对地址 X+（区域 2：H'FFF84000，区域 3：H'FFF85000）进行字写，就能将值 X 写到 SDRAM 的模式寄存器，此时忽视写数据。在设定本 LSI 支持的突发读 / 单次写（突发长度为 1）或者突发读 / 突发写（突发长度为 1）、CAS 潜伏时间为 2 ~ 3、Wrap 型 = 顺序以及突发长度为 1 时，将任意数据写到表 9.27 所示的存取地址，此时给外部地址引脚 A12 开始的位输出 0。

表 9.27 写 SDRAM 模式寄存器时的存取地址

- 区域 2 的设定 (SDMR2)  
突发读 / 单次写（突发长度为 1）的情况

数据总线宽度	CAS 潜伏时间	存取地址	外部地址引脚
16 位	2	H'FFF84440	H'0000440
	3	H'FFF84460	H'0000460
32 位	2	H'FFF84880	H'0000880
	3	H'FFF848C0	H'00008C0

突发读 / 突发写（突发长度为 1）的情况

数据总线宽度	CAS 潜伏时间	存取地址	外部地址引脚
16 位	2	H'FFF84040	H'0000040
	3	H'FFF84060	H'0000060
32 位	2	H'FFF84080	H'0000080
	3	H'FFF840C0	H'00000C0

- 区域 3 的设定 (SDMR3)  
突发读 / 单次写（突发长度为 1）的情况

数据总线宽度	CAS 潜伏时间	存取地址	外部地址引脚
16 位	2	H'FFF85440	H'0000440
	3	H'FFF85460	H'0000460
32 位	2	H'FFF85880	H'0000880
	3	H'FFF858C0	H'00008C0

突发读 / 突发写（突发长度为 1）的情况

数据总线宽度	CAS 潜伏时间	存取地址	外部地址引脚
16 位	2	H'FFF85040	H'0000040
	3	H'FFF85060	H'0000060
32 位	2	H'FFF85080	H'0000080
	3	H'FFF850C0	H'00000C0

模式寄存器的设定时序如图 9.30 所示。首先产生全部存储体的预充电命令 (PALL)，其次产生 8 次自动刷新命令 (REF)，最后产生模式寄存器的写命令 (MRS)。在 PALL 和第 1 次 REF 之间插入 CS3WCR 的 WTRP[1:0] 位所设个数的空闲周期；在 REF 和 REF 之间以及第 8 次 REF 和 MRS 之间插入 CS3WCR 的 WTRC[1:0] 位所设个数的空闲周期；在 MRS 和下一次产生的命令之间至少插入 1 个空闲周期。

必须在接通电源后并且在进行全部存储体的预充电 (PALL) 前，对 SDRAM 确保一定的空闲时间。所需的空闲时间请参照 SDRAM 的使用手册。如果复位信号的脉冲宽度长于此空闲时间，立即设定模式寄存器也没关系，但是如果短于空闲时间就需要注意。

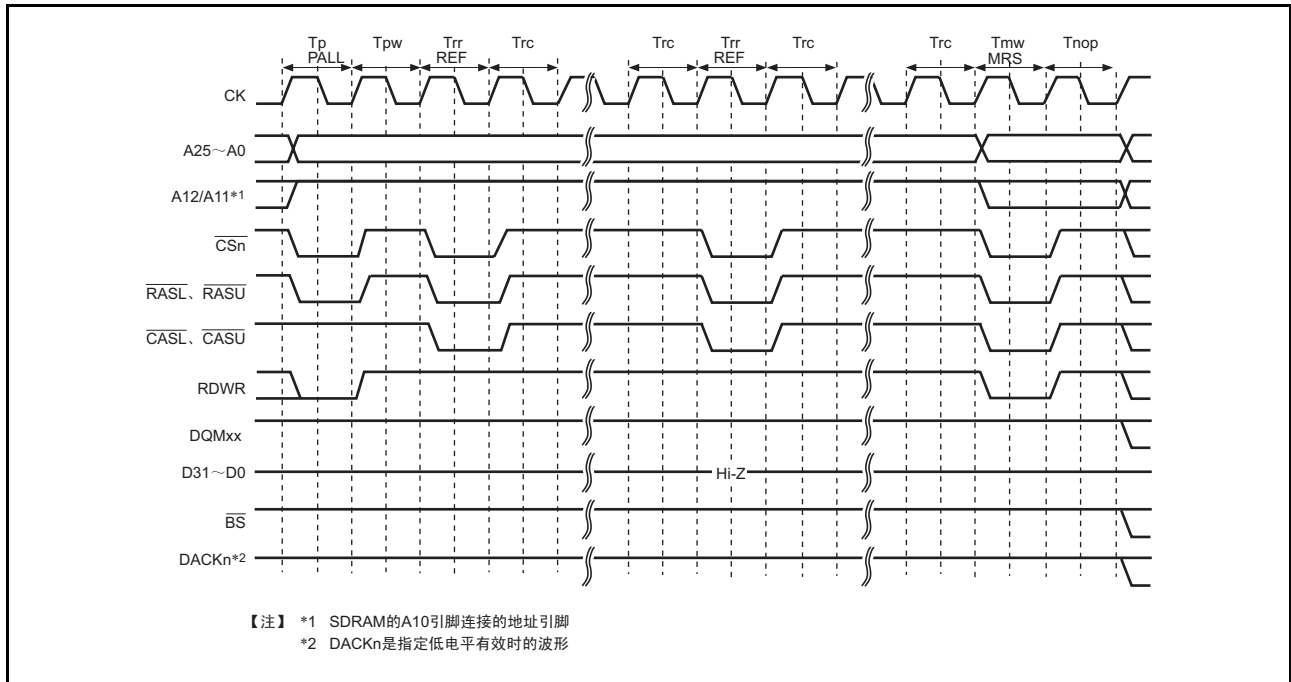


图 9.30 SDRAM 模式寄存器的写时序 (以 JEDEC 为基准)

### 9.5.7 突发 ROM (异步) 接口

突发 ROM (异步) 接口用于通过称为突发模式或者页模式等的地址转换，存取具有高速读取功能的存储器。基本上进行和通常空间相同的存取，但是在最初的周期结束时，不将 RD 信号置为有效而只转换地址，进行第 2 次以后的存取。第 2 次以后的存取在 CK 的下降沿发生地址变化。

在最初的存取周期中插入 CSnWCR 的 W[3:0] 位所设个数的等待周期，在第 2 次以后的存取周期中插入 CSnWCR 的 BW[1:0] 位所设个数的等待周期。

当存取突发 ROM (异步) 时，BS 信号只对最初的存取周期有效，外部等待输入也只对最初的存取周期有效。

当通过突发 ROM (异步) 接口进行单次存取 (而不进行突发运行) 时，存取时序和通常空间相同。

总线宽度、存取长度和突发数的关系如表 9.28、时序图如图 9.31 所示。

表 9.28 总线宽度、存取长度和突发数的关系

总线宽度	BEN 位	存取长度	突发数	存取次数
8 位	没有影响	8 位	1	1
	没有影响	16 位	2	1
	没有影响	32 位	4	1
	0	16 字节	16	1
	1		4	4
16 位	没有影响	8 位	1	1
	没有影响	16 位	1	1
	没有影响	32 位	2	1
	0	16 字节	8	1
	1		2	4
32 位	没有影响	8 位	1	1
	没有影响	16 位	1	1
	没有影响	32 位	1	1
	没有影响	16 字节	4	1

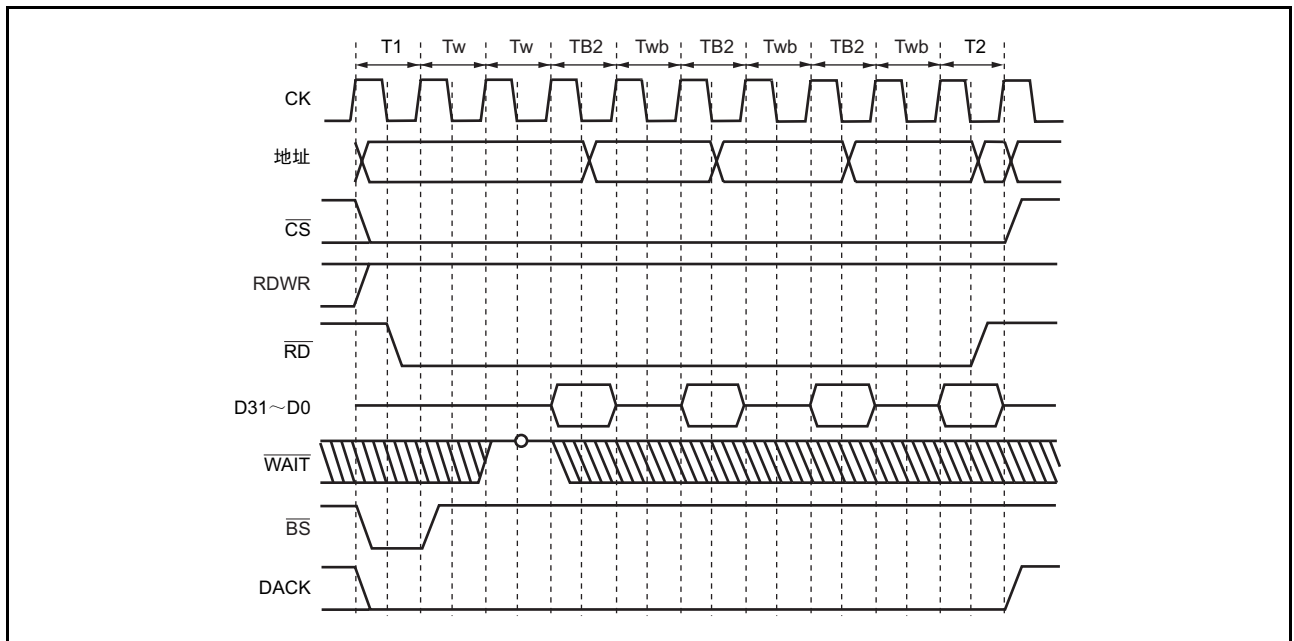


图 9.31 突发 ROM (异步) 的存取  
(32 位总线的 16 字节传送 (突发数=4)、初次存取等待=2、第 2 次以后的存取等待=1)

### 9.5.8 带字节选择的 SRAM 接口

带字节选择的 SRAM 接口是在读或者写总线周期中也输出字节选择引脚信号 ( $\overline{WR_{xx}}$ ) 的存储器接口。此接口有 16 位数据引脚, 用于存取有如 UB 或者 LB 的高位字节选择引脚和低位字节选择引脚的 SRAM。

当 CSnWCR 寄存器的 BAS 位是 0 (初始值) 时, 带字节选择的 SRAM 接口的写存取时序和通常空间接口相同。然而在进行读操作时,  $\overline{WR_{xx}}$  引脚的时序和通常空间接口不同, 从  $\overline{WR_{xx}}$  引脚输出字节选择信号。基本存取时序如图 9.32 所示。尤其在进行写操作时, 通过字节选择引脚 ( $\overline{WR_{xx}}$ ) 的时序写存储器, 所以必须确认所使用的存储器数据表。

当 CSnWCR 寄存器的 BAS 位是 1 时,  $\overline{WR_{xx}}$  引脚和 RDWR 引脚的时序发生变化。基本存取时序如图 9.33 所示。尤其在进行写操作时, 通过写允许引脚 (RDWR) 时序写存储器。必须通过设定 CSnWCR 的 HW[1:0] 位, 确保从 RDWR 无效到写数据的保持时间。软件等待设定时的存取时序如图 9.34 所示。

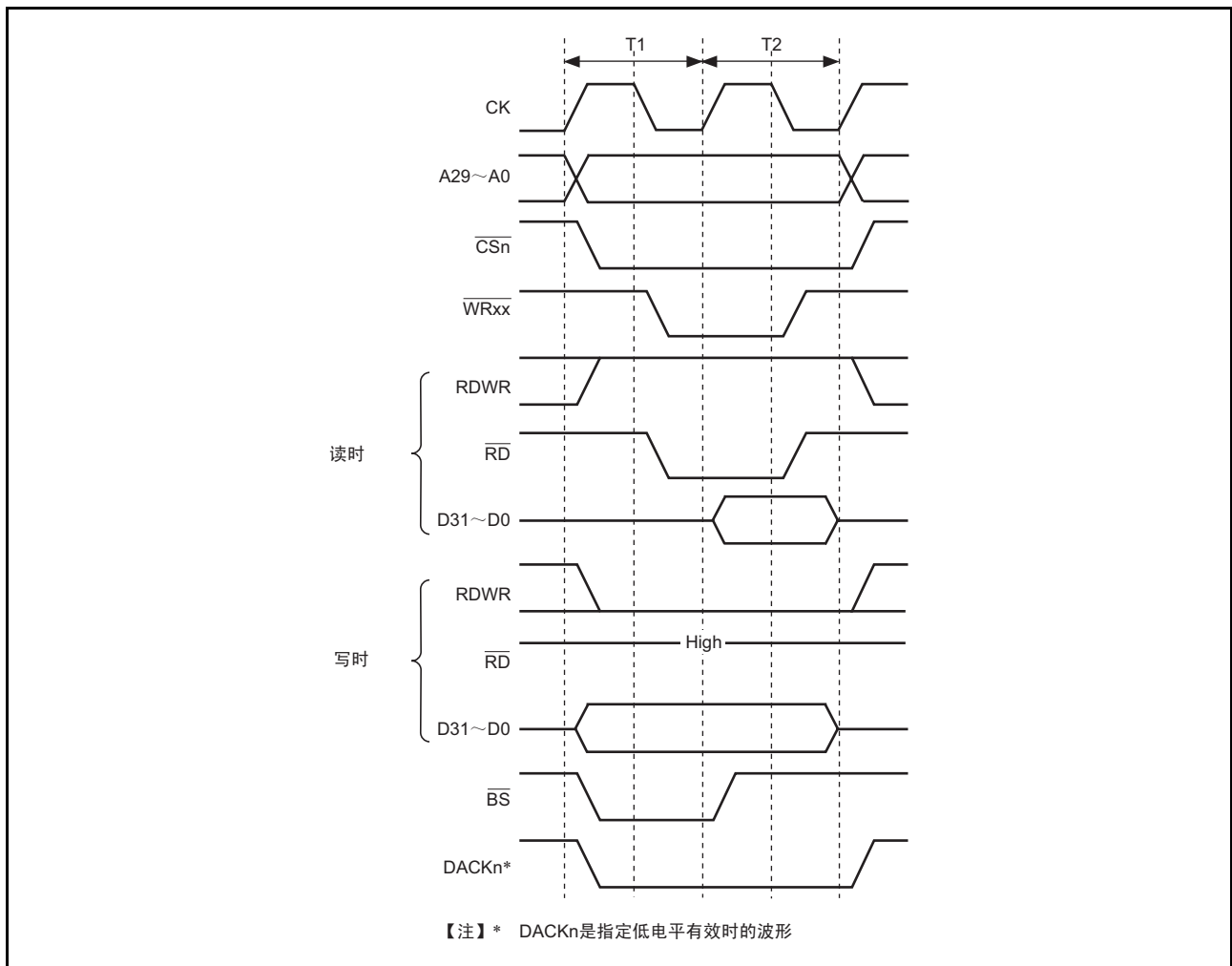


图 9.32 带字节选择的 SRAM 的基本存取时序 (BAS=0)

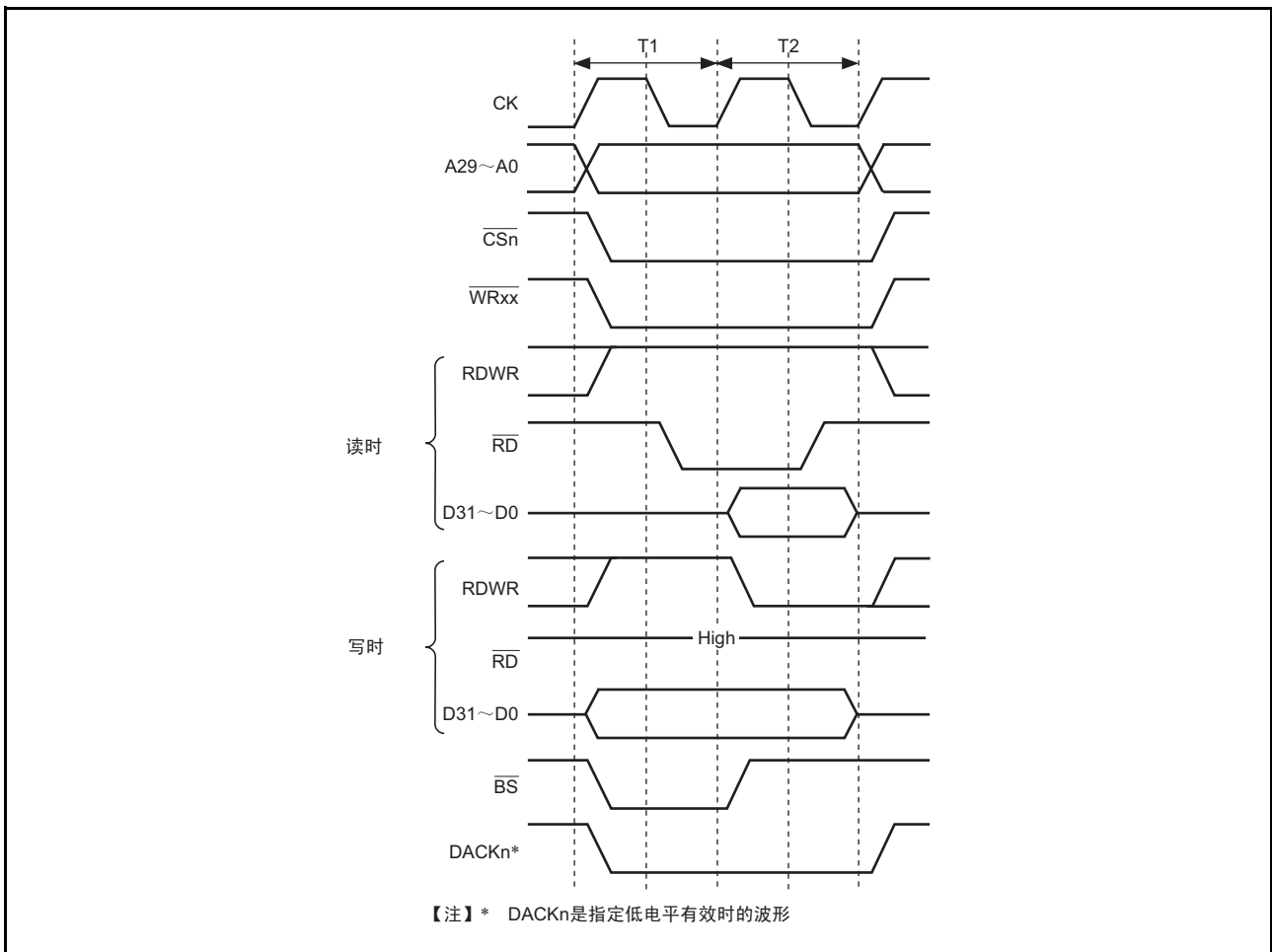


图 9.33 带字节选择的 SRAM 的基本存取时序 (BAS=1)

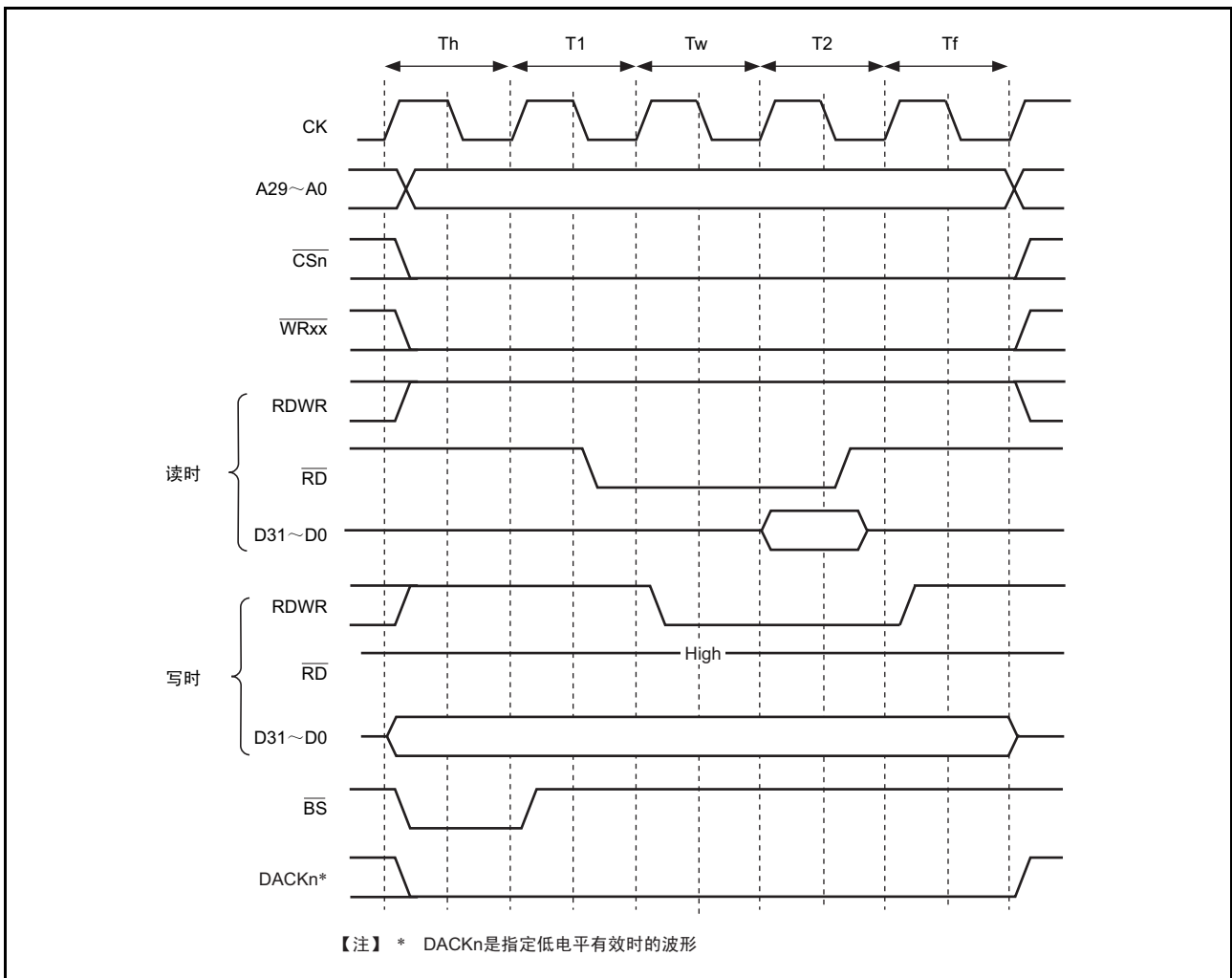


图 9.34 带字节选择的 SRAM 的等待时序  
(BAS=1、SW[1:0]=01、WR[3:0]=0001、HW[1:0]=01)

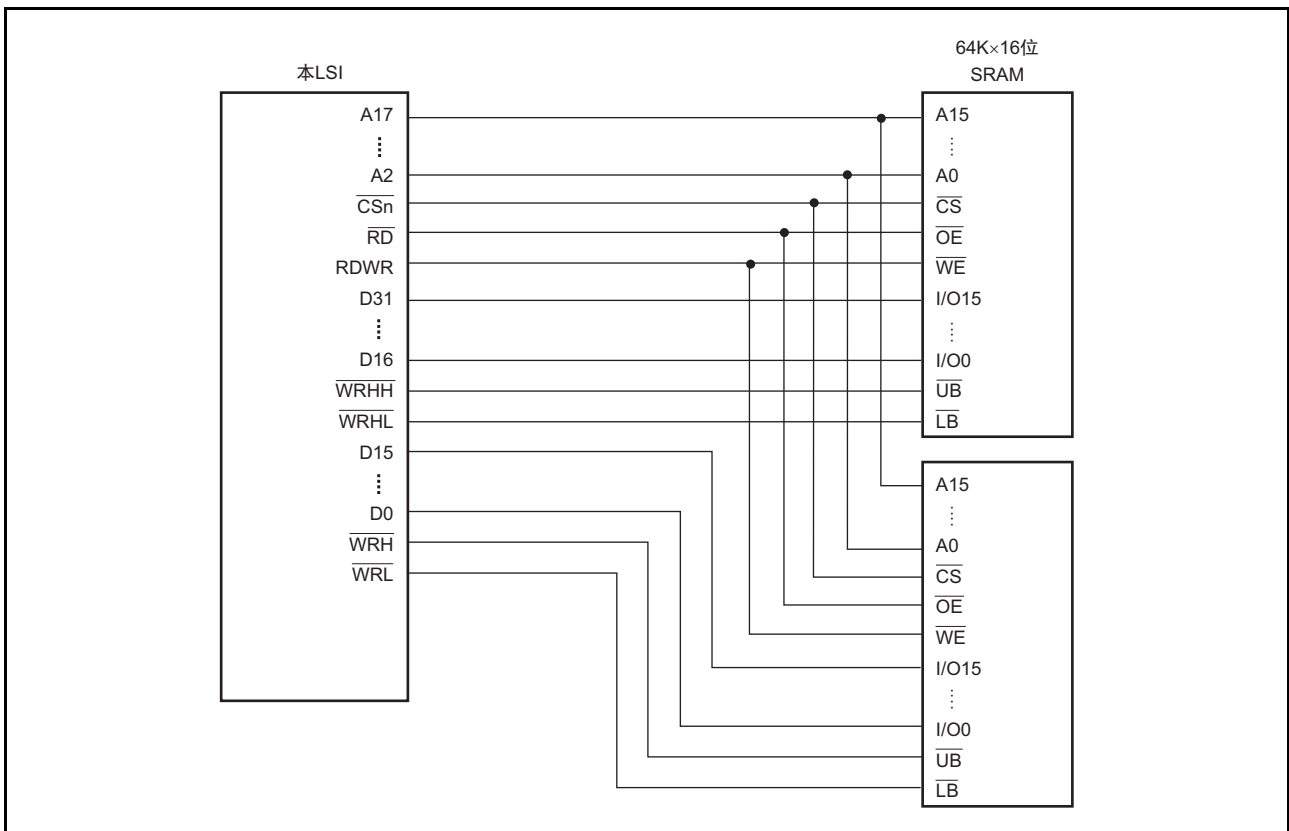


图 9.35 带字节选择的 SRAM 连接例 (32 位数据)

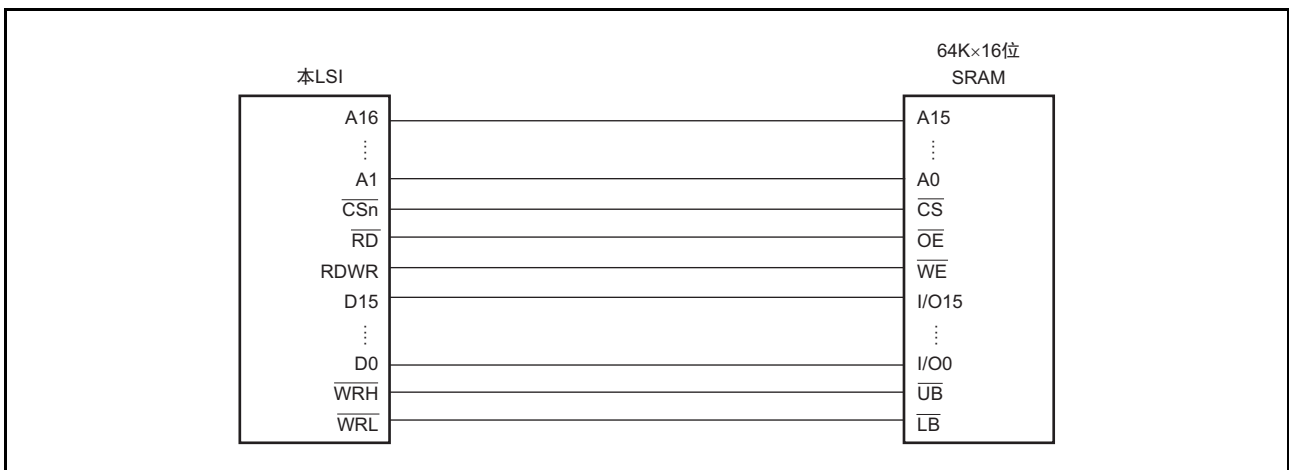


图 9.36 带字节选择的 SRAM 连接例 (16 位数据)



### 9.5.9 PCMCIA 接口

本 LSI 能在区域 5 和区域 6 中设定 PCMCIA 接口。通过将 CSnBCR (n=5、6) 的 TYPE[2:0] 位设定为 B'101, 将域 5 和区域 6 设定为符合 JEIDA 规格 Ver4.2 (PCMCIA2.1 Rev2.1) 的 IC 存储卡和 I/O 卡接口, 还能通过 CSnWCR (n=5、6) 的 SA[1:0] 位, 将各区域的前 32MB 和后 32MB 设定为 IC 存储卡或者 I/O 卡接口。例如, 当将 CS5WCR 的 SA1 位和 SA0 位分别设定为 1 和 0 时, 前 32MB 是 IC 存储卡接口, 而后 32MB 是 I/O 卡接口。

在使用 PCMCIA 接口时, 必须通过 CS5BCR 的 BSZ[1:0] 位或者 CS6BCR 的 BSZ[1:0] 位, 将总线宽度设定为 8 位或者 16 位。

本 LSI 和 PCMCIA 卡的连接例子如图 9.37 所示。为了能进行 PCMCIA 卡的热插拔 (在提供系统电源时插拔 PCMCIA 卡), 需要在本 LSI 的总线接口和 PCMCIA 卡之间连接 3-State 缓冲器。

因为在 JEIDA 和 PCMCIA 的规格中没有明确规定大端法模式的运行规格, 所以本 LSI 大端法模式的 PCMCIA 接口为独自规定的接口规格。

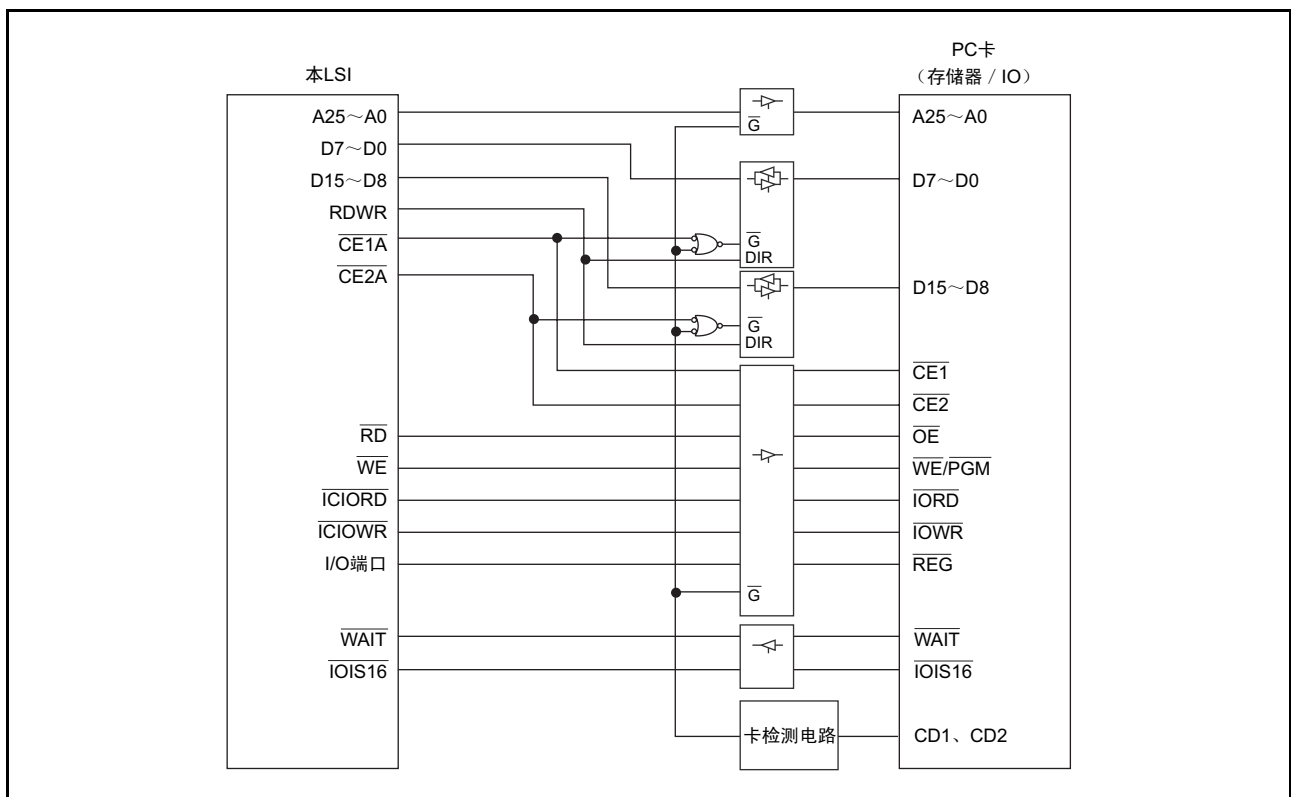


图 9.37 PCMCIA 接口的连接例

(1) 存储卡接口的基本时序

PCMCIA 的 IC 存储卡接口的基本时序如图 9.38 所示。在将区域 5 和区域 6 设定为 PCMCIA 接口的情况下，如果存取各区域的通用存储空间，就作为 IC 存储卡接口自动进行总线的存取。如果外部总线频率 (CK) 变高，对于  $\overline{RD}$  和  $\overline{WE}$ ，地址 (A25 ~ A0)、卡的允许信号 ( $\overline{CE1A}$ 、 $\overline{CE1B}$ 、 $\overline{CE2A}$ 、 $\overline{CE2B}$ ) 以及写周期中的写数据 (D15 ~ D0) 的准备时间和保持时间就会不足。因此，本 LSI 能通过 CS5WCR 或者 CS6WCR 寄存器，分别给物理空间的区域 5 和区域 6 设定准备时间和保持时间。另外，和通常空间接口一样，能进行软件等待以及通过 WAIT 引脚进行硬件等待。PCMCIA 存储卡的总线等待时序如图 9.39 所示。

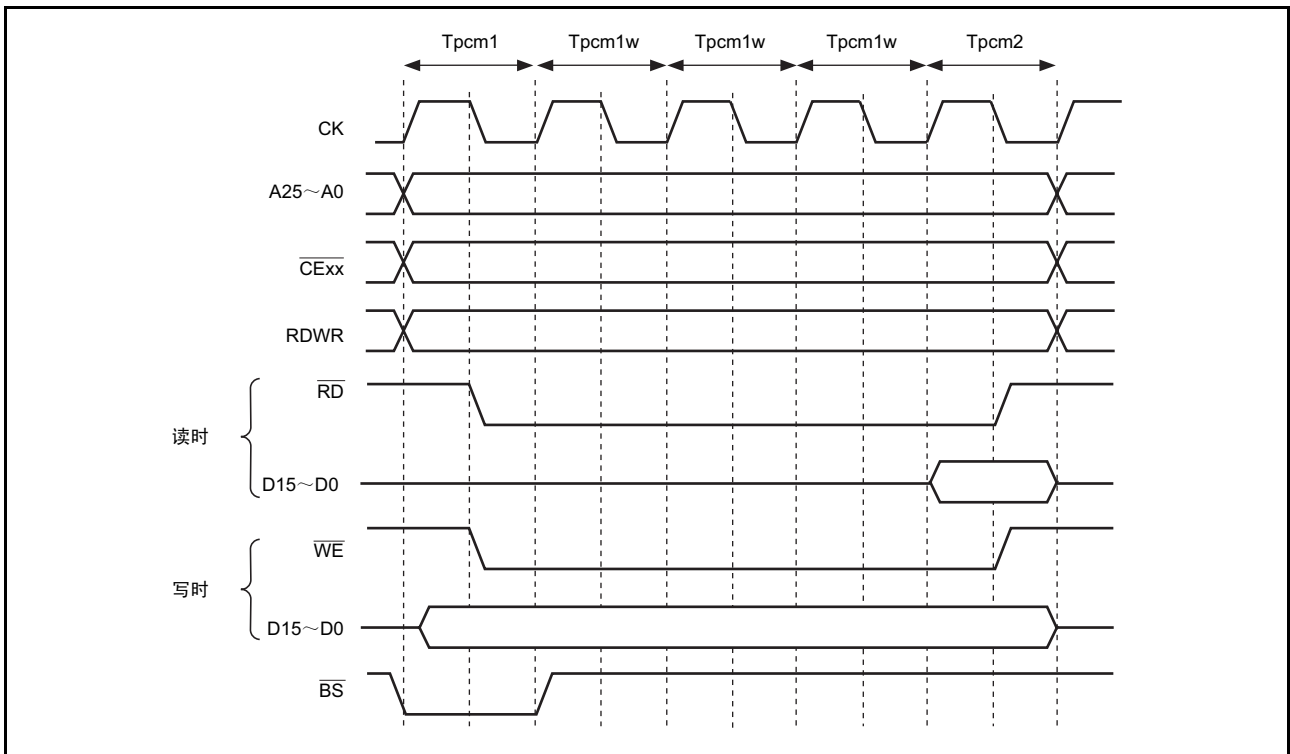


图 9.38 PCMCIA 存储卡接口的基本时序

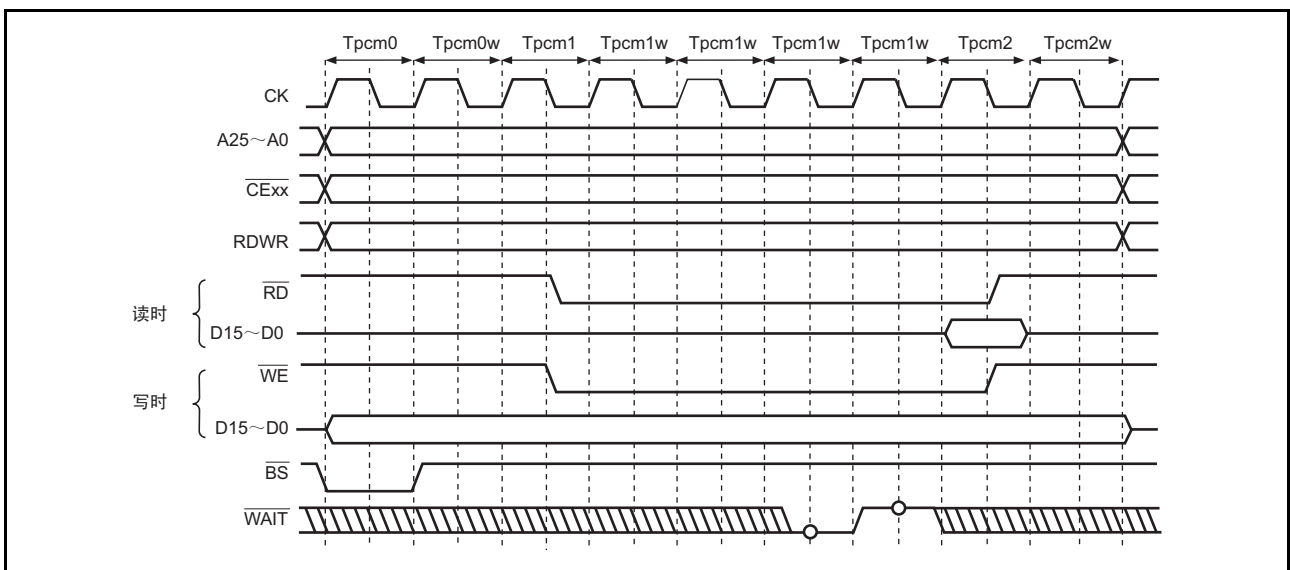


图 9.39 PCMCIA 存储卡接口的等待时序  
(TED[3:0]=B'0010、TEH[3:0]=B'0001、硬件等待 =1)

当将 32M 字节存储空间全部用作 IC 存储卡接口时，利用端口等生成通用存储器和属性存储器的转换信号  $\overline{\text{REG}}$ 。另外，在存储空间不超过 16M 字节并且够用的情况下，通过将存储空间用作 16M 字节的通用存储空间和 16M 字节的属性存储空间，A24 引脚就能用作  $\overline{\text{REG}}$  信号。

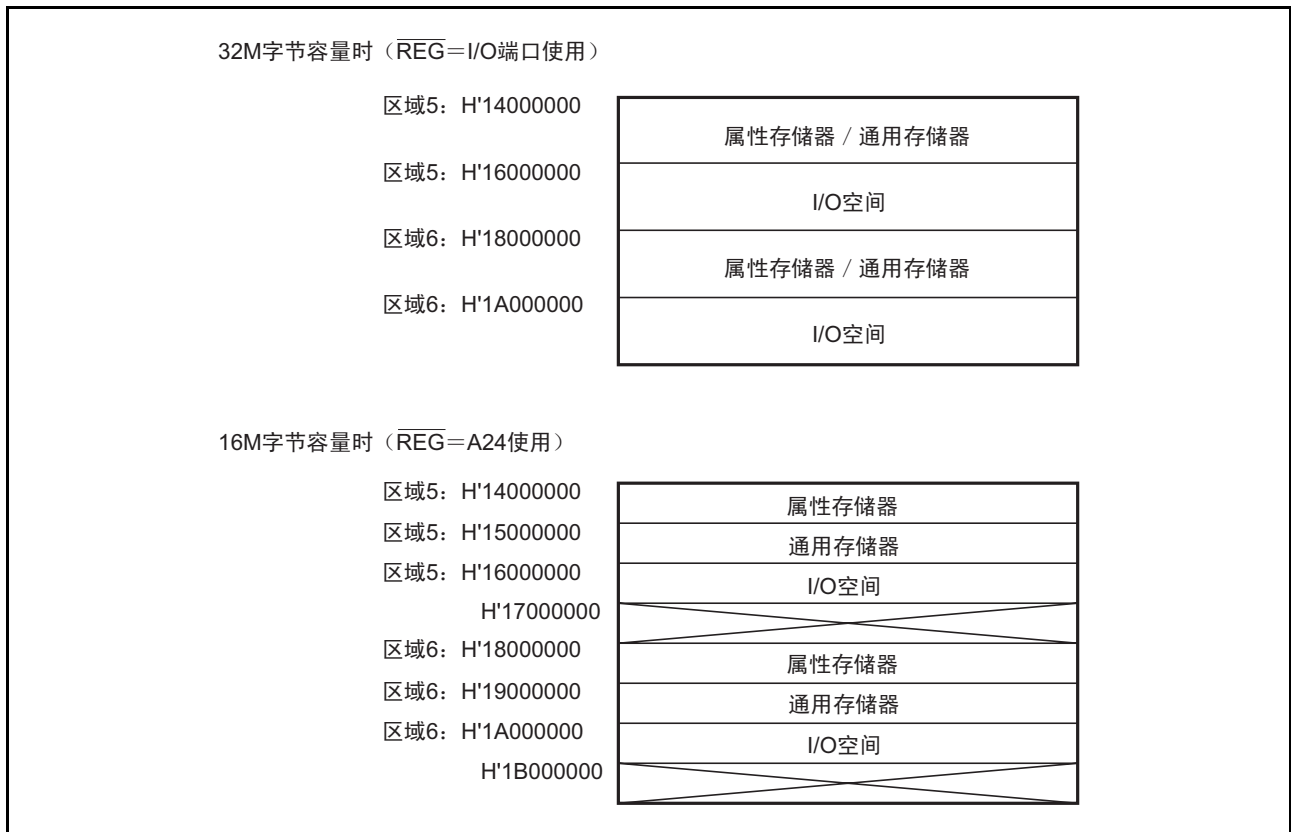


图 9.40 PCMCIA 空间的分配例  
(CS5BWCR.SA[1:0]=B'10、CS6BWCR.SA[1:0]=B'10)

(2) I/O 卡接口的基本时序

PCMCIA 的 I/O 卡接口时序如图 9.41 和图 9.42 所示。

通过要存取的地址转换 I/O 卡接口和 IC 存储卡接口。如果将区域 5 设定为 PCMCIA 并将 CS5WCR 的 SA1、SA0 位设定为 1，就分别将地址 H'16000000 ~ H'17FFFFFF、H'14000000 ~ H'15FFFFFF 分配为 I/O 卡区域。如果将区域 6 设定为 PCMCIA 并将 CS6WCR 的 SA1、SA0 位设定为 1，就分别将地址 H'1A000000 ~ H'1BFFFFFF、H'18000000 ~ H'19FFFFFF 分配为 I/O 卡区域。

因为本 LSI 不支持小端法，所以必须将 IOIS16 信号固定为低电平。

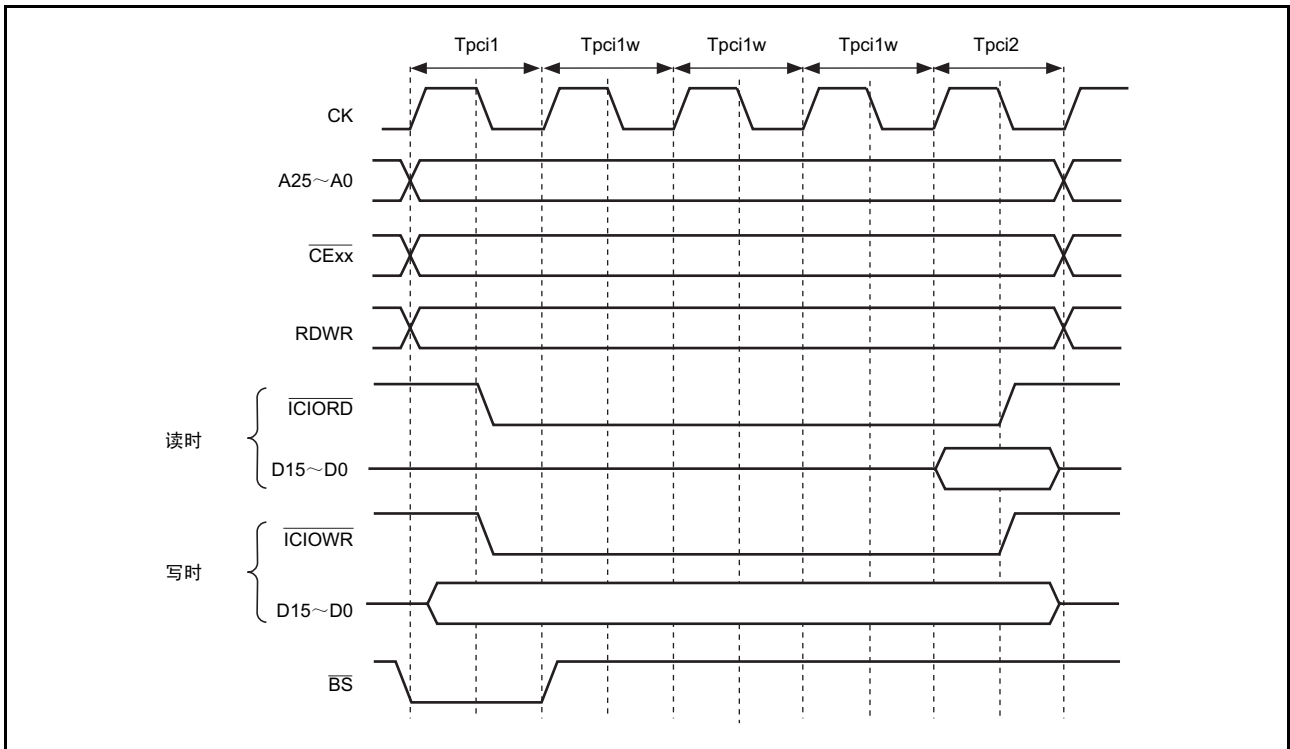


图 9.41 PCMCIA I/O 卡接口的基本时序

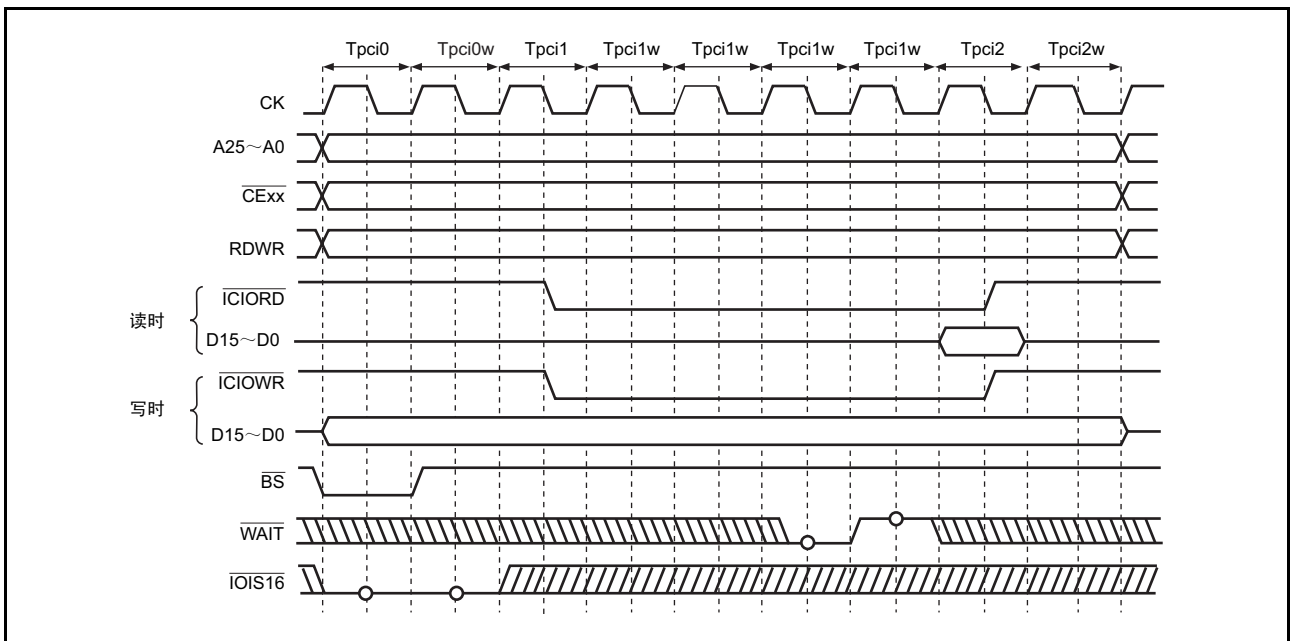


图 9.42 PCMCIA I/O 卡接口的等待时序  
(TED[3:0]=B'0010、TEH[3:0]=B'0001、硬件等待 =1)

### 9.5.10 突发 MPX-I/O 接口

突发 MPX 器件的连接例和突发 MPX 空间的存取时序分别如图 9.43 和图 9.44 ~ 图 9.47 所示。

通过设定 CS6BCR 寄存器的 TYPE[2:0] 位，能将区域 6 设定为突发地址 / 数据多路复用 I/O 接口。通过本接口能容易地连接到使用地址 / 数据多路复用的 32 位单总线的外部存储器控制器芯片。将地址输出到 D25 ~ D0，在地址周期中将存取长度输出到 D31 ~ D29。D31 ~ D29 的输出和存取长度的对应请参照 CS6WCR 寄存器说明。

将通常的地址输出到地址引脚 A25 ~ A0。

在使用突发 MPX 接口时，总线宽度固定为 32 位，所以必须将 CS6BCR 的 BSZ[1:0] 位设定为 32 位。另外，能插入软件等待以及 WAIT 引脚产生的硬件等待。

在进行读操作时，即使软件等待已设定为 0，也能在地址输出后自动插入 1 个周期的等待。

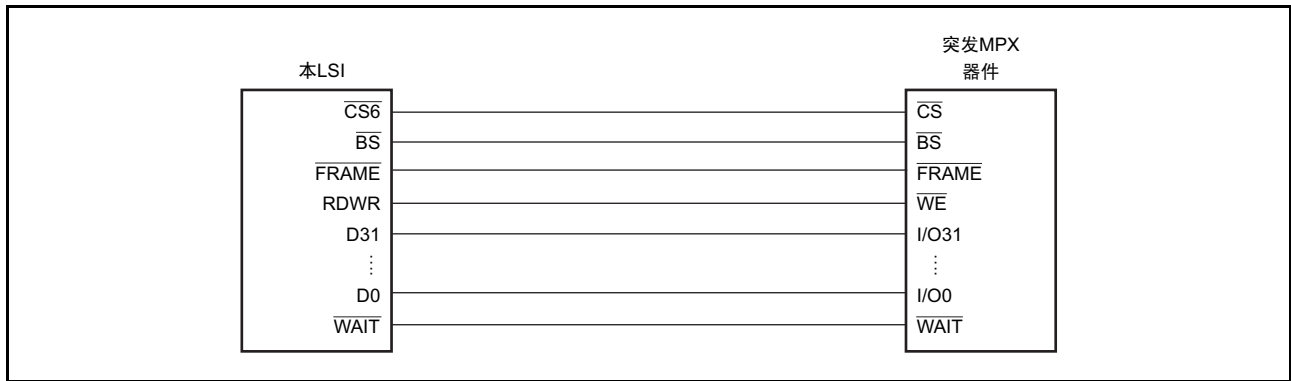


图 9.43 突发 MPX 器件的连接例

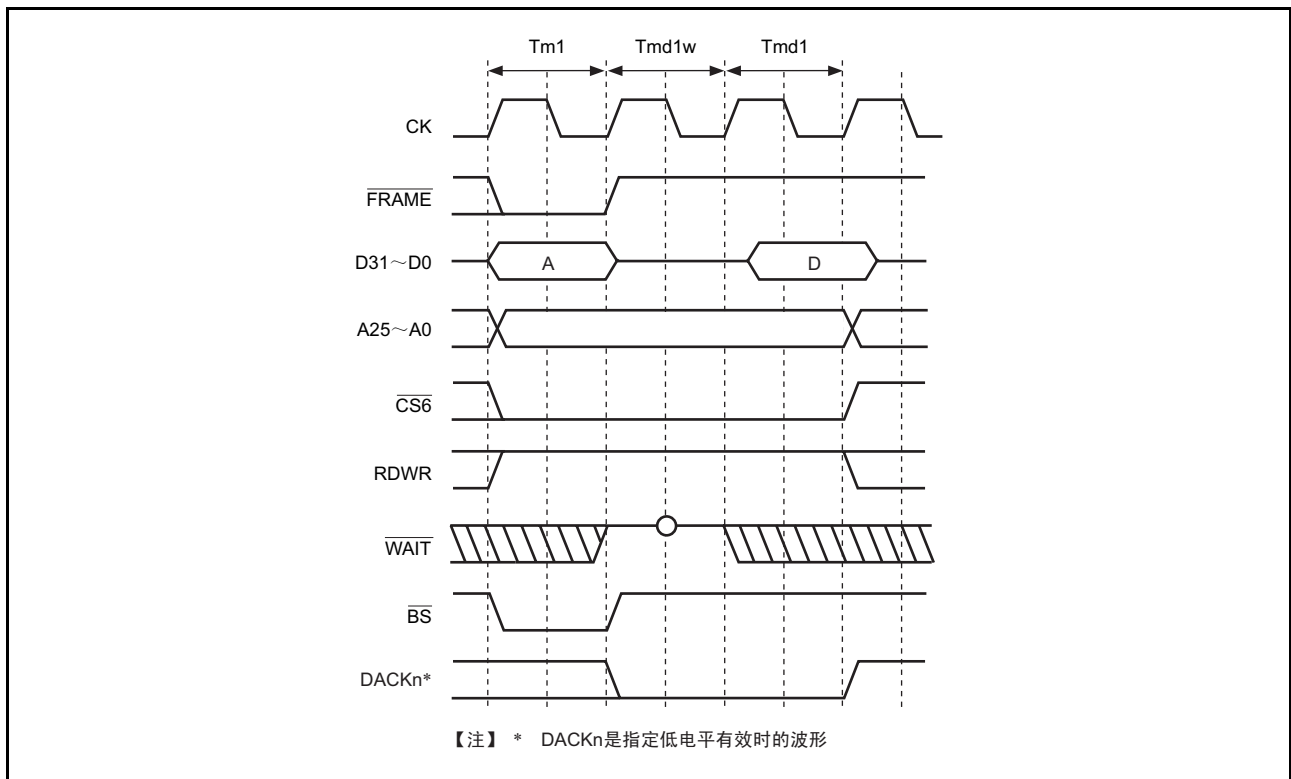


图 9.44 突发 MPX 空间的存取时序  
(单次读、无等待或者软件等待 = 1)

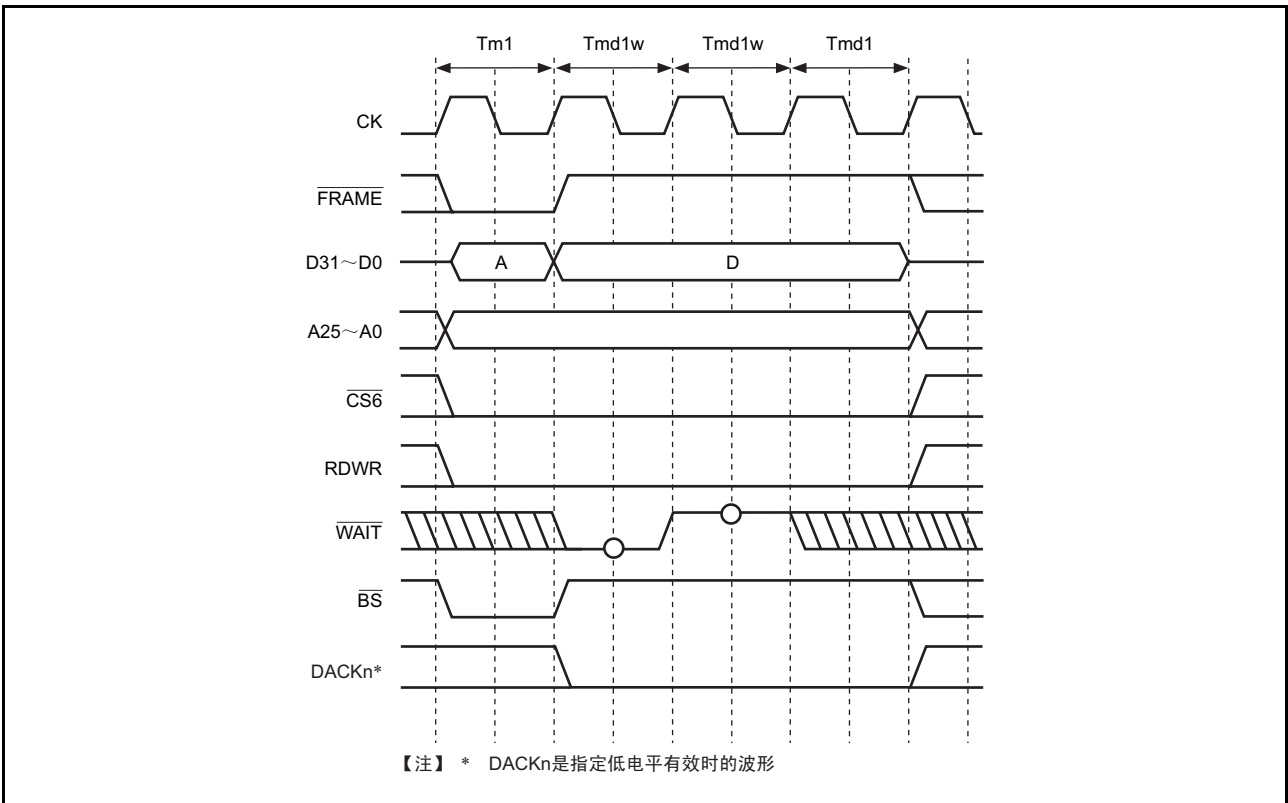


图 9.45 突发 MPX 空间的存取时序  
(单次写、软件等待 =1、硬件等待 =1)

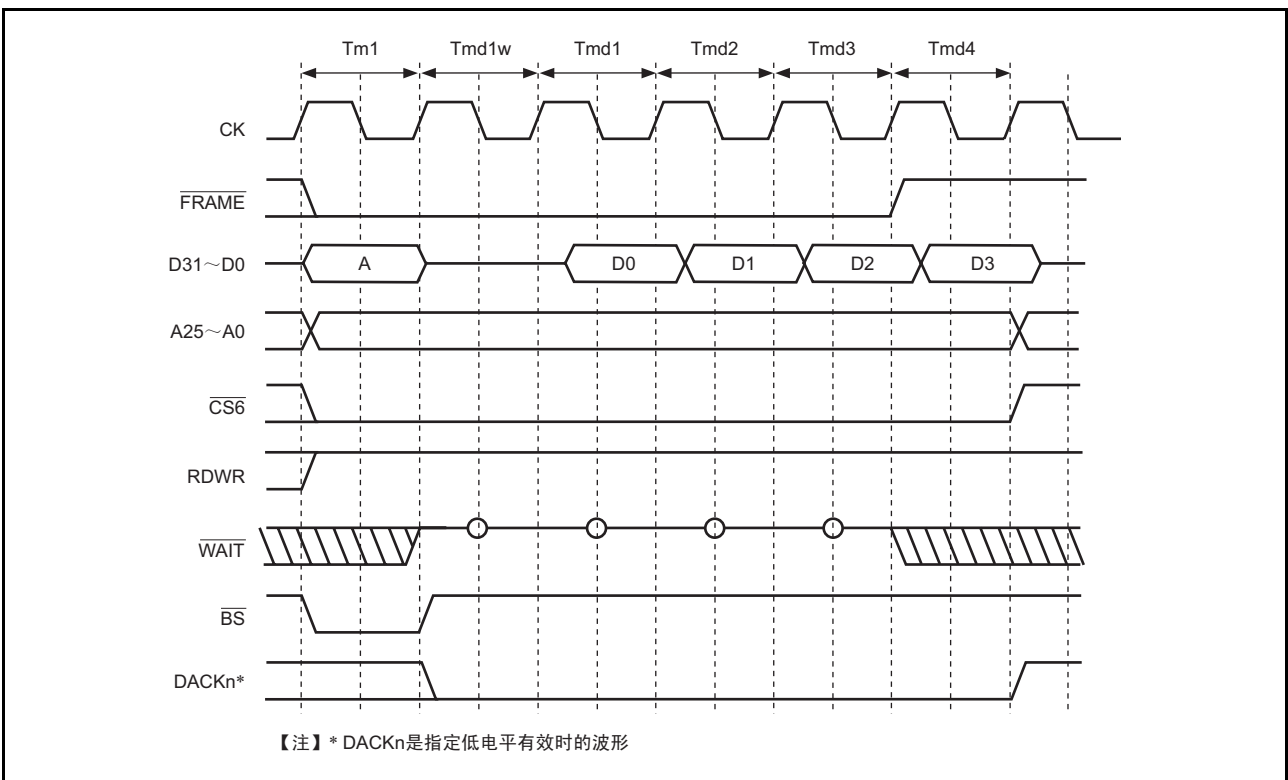


图 9.46 突发 MPX 空间的存取时序  
(突发读、无等待或者软件等待 =1)

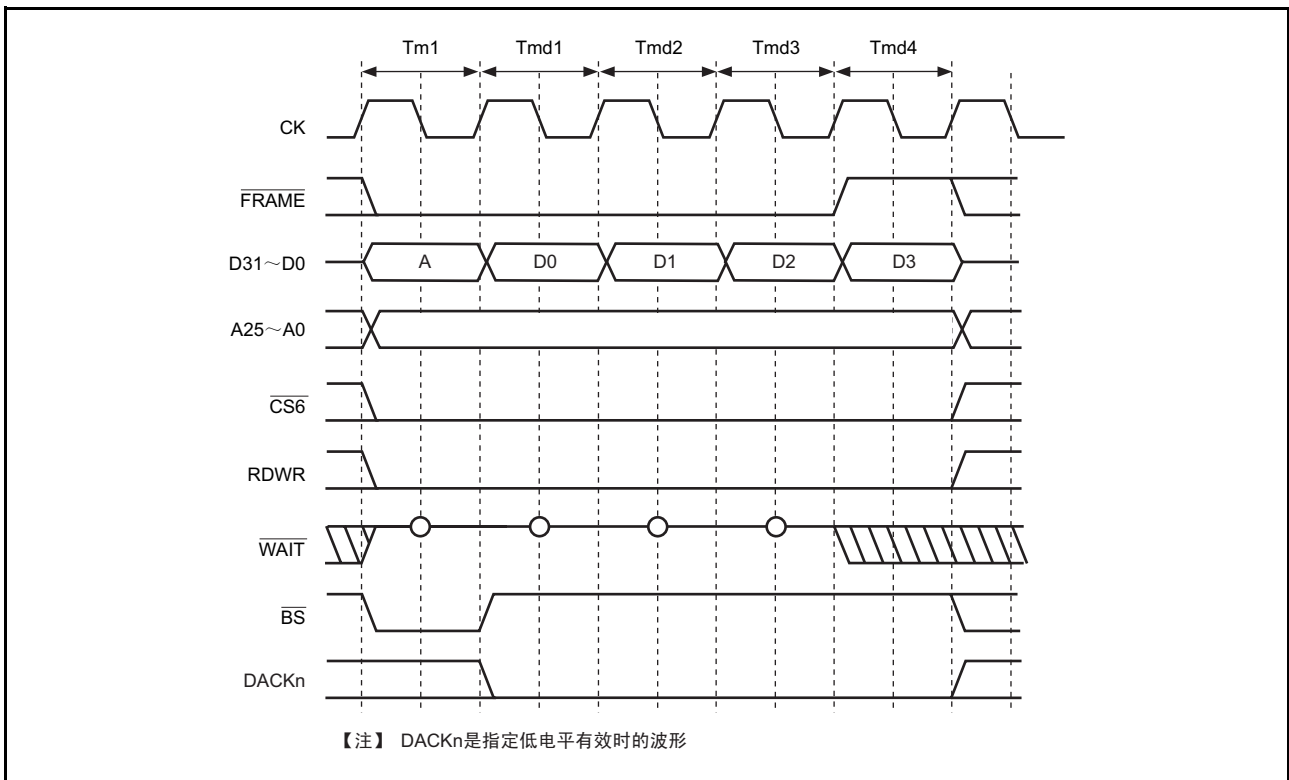


图 9.47 突发 MPX 空间的存取时序 (突发写、无等待)

### 9.5.11 突发 ROM (时钟同步) 接口

突发 ROM (时钟同步) 接口能高速存取具有同步突发功能的 ROM。基本上进行和通常空间相同的存取, 但是本接口只能设定在区域 0。

在最初的存取周期中插入 CS0WCR 的 W[3:0] 位所设个数的等待周期, 在第 2 次以后的存取周期中插入 CS0WCR 的 BW[1:0] 位所设个数的等待周期。

当存取突发 ROM (时钟同步) 时,  $\overline{BS}$  信号只对最初的存取周期有效, 外部等待输入也只对最初的存取周期有效。

当总线宽度为 16 位时, 必须将突发长度设定为 8; 当总线宽度为 32 位时, 必须将突发长度设定为 4。本接口不支持 8 位总线, 并且对全部读存取进行突发运行。例如, 以 16 位总线进行长字存取时, 在读需要的 2 个数据后, 空读剩余的 6 个数据。

因为这样的空读周期会增加存储器存取时间, 降低程序的执行速度和 DMA 的传送速度, 所以必须有效地利用 DMA 的 16 字节读操作。

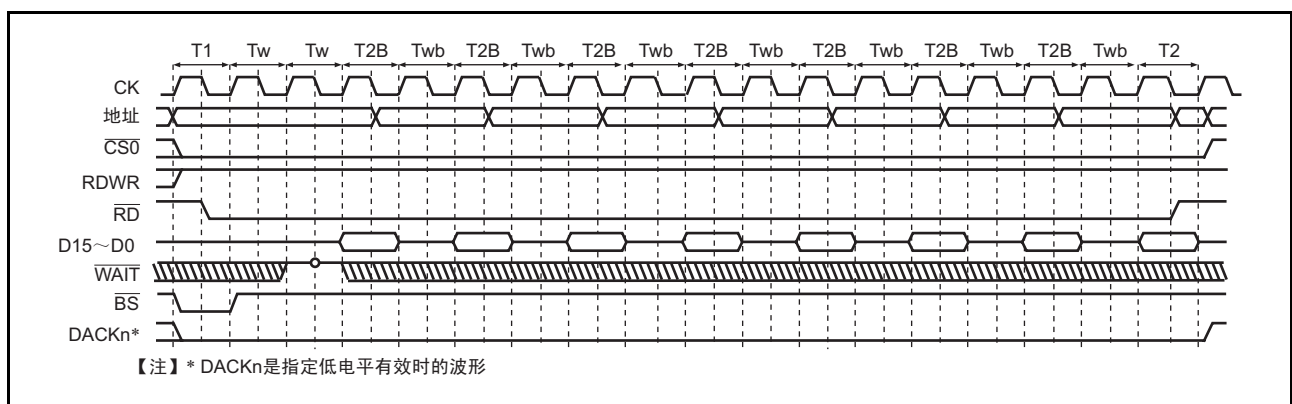


图 9.48 突发 ROM (时钟同步) 的存取  
(突发长度 =8、初次存取等待 =2、第 2 次以后的存取等待 =1)



### 9.5.12 存取周期之间的等待

由于 LSI 工作频率的提高, 有可能在低速器件的数据输出结束时来不及关闭数据缓冲器, 导致和下一次器件的数据输出发生冲突, 从而降低器件的可靠性或者引起误动作。为了防止发生这种情况, 在连续的存取周期之间插入等待, 以避免数据的冲突。

通过 CSnBCR 的 IWW[1:0] 位、IWRWD[1:0] 位、IWRWS[1:0] 位、IWRRD[1:0] 位和 IWRRS[1:0] 以及 CMNCR 的 DMAIW[1:0] 位和 DMAIWA 位指定存取周期之间的等待周期数。可在存取周期之间插入等待 (空闲周期) 的条件如下:

1. 连续存取为写 - 读或者写 - 写
2. 连续存取为不同空间的读 - 写
3. 连续存取为同一空间的读 - 写
4. 连续存取为不同空间的读 - 读
5. 连续存取为同一空间的读 - 读
6. 在 DMA 单地址传送中的外部器件的数据输出周期后, 从其他器件 (包括本 LSI) 输出数据时 (DMAIWA=0)
7. 在 DMA 单地址传送中的外部器件的数据输出周期后, 发生存取时 (DMAIWA=1)

除了上述所示的存取周期之间的等待 (空闲周期) 以外, 为了确保连接内部总线的接口或者多路复用引脚 ( $\overline{WRxx}$ ) 的最小脉冲宽度, 有时需要插入空闲周期。

8. 与内部总线连接的接口相关的外部总线的空闲周期
  - (a) 在外部总线空闲周期或者读周期后的写存取前, 插入 1 个空闲周期。
  - (b) 为了在外部总线的读周期结束时将读数据传送到内部总线, 插入 1 个空闲周期。  
在紧接着读周期后的写周期中, 包含上述 (a) 插入 2 ~ 3 个周期的空闲周期。
9. 不同存储器之间存取时的外部总线的空闲周期  
进行不同存储器之间的存取时, 在以下情况下插入空闲周期。此时, 因为  $\overline{WRxx}$  的变化时序相同, 所以将 BAS 位 =1 的带字节选择的 SRAM 接口作为 SDRAM 接口处理。
  - (a) 在通过 SDRAM 接口进行写存取后, 存取 SDRAM 以外的接口时, 插入 1 个空闲周期。
  - (b) 在存取外部等待无效的通常空间接口以及 BAS 位 =0 的带字节选择的 SRAM 接口后, 存取 SDRAM 接口时, 插入 1 个空闲周期。
  - (c) 在存取 MPX-I/O 接口后, 存取 SDRAM 接口时, 插入 1 个空闲周期。
  - (d) 在外部总线空闲的状态下存取 MPX-I/O 接口时, 插入 2 个空闲周期。
  - (e) 在通常空间接口、BAS 位 =0 的带字节选择的 SRAM 接口和 SDRAM 接口的读周期后, 存取 MPX-I/O 接口时, 插入 1 个空闲周期。
  - (f) 在 SDRAM 接口的写周期后, 存取 MPX-I/O 接口时, 插入 2 个空闲周期。

在通常空间接口和 SDRAM 接口时插入的最小空闲周期数如表 9.29 ~ 表 9.34 所示。表中的 “CSnBCR 的空闲设定” 表示由 IWW、IWRWD、IWRWS、IWRRD 或者 IWRRS 设定的空闲周期数。

表 9.29 通常空间接口的 CPU 存取之间的最小空闲周期数

BSC 寄存器的设定		不超过总线宽度的存取				超过总线宽度的存取					
CSnWCR.WM 的设定	CSnBCR 的空闲设定	读 → 读	写 → 写	读 → 写	写 → 读	连续读 *1	连续写 *1	读 → 读 *2	写 → 写 *2	读 → 写 *2	写 → 读 *2
1	0	1,1,1,1	0,0,0,0	3,3,3,4	0,0,0,0	0,0,0,0	0,0,0,0	1,1,1,1	0,0,0,0	3,3,3,4	0,0,0,0
0	0	1,1,1,1	1,1,1,1	3,3,3,4	1,1,1,1	1,1,1,1	1,1,1,1	1,1,1,1	1,1,1,1	3,3,3,4	1,1,1,1
1	1	1,1,1,1	1,1,1,1	3,3,3,4	1,1,1,1	1,1,1,1	1,1,1,1	1,1,1,1	1,1,1,1	3,3,3,4	1,1,1,1
0	1	1,1,1,1	1,1,1,1	3,3,3,4	1,1,1,1	1,1,1,1	1,1,1,1	1,1,1,1	1,1,1,1	3,3,3,4	1,1,1,1
1	2	2,2,2,2	2,2,2,2	3,3,3,4	2,2,2,2	2,2,2,2	2,2,2,2	2,2,2,2	2,2,2,2	3,3,3,4	2,2,2,2
0	2	2,2,2,2	2,2,2,2	3,3,3,4	2,2,2,2	2,2,2,2	2,2,2,2	2,2,2,2	2,2,2,2	3,3,3,4	2,2,2,2
1	4	4,4,4,4	4,4,4,4	4,4,4,4	4,4,4,4	4,4,4,4	4,4,4,4	4,4,4,4	4,4,4,4	4,4,4,4	4,4,4,4
0	4	4,4,4,4	4,4,4,4	4,4,4,4	4,4,4,4	4,4,4,4	4,4,4,4	4,4,4,4	4,4,4,4	4,4,4,4	4,4,4,4

【注】 表中的最小空闲周期数按顺序表示  $t_{\phi:B\phi}=4:1$ 、 $3:1$ 、 $2:1$ 、 $1:1$  的情况。

\*1 表示在 16 位总线的 32 位存取中，地址 0 的字存取和地址 2 的字存取之间的最小空闲周期数；在 8 位总线的 16 位存取中，地址 0 的字节存取和地址 1 的字节存取之间的最小空闲周期数；在 8 位总线的 32 位存取中，地址 0 的字节存取、地址 1 的字节存取、地址 2 的字节存取和地址 3 的字节存取之间的最小空闲周期数；在 16 字节传送中的连续存取之间的最小空闲周期数

\*2 上述以外

表 9.30 通常空间接口的 DMAC 双地址模式和 DTC 传送中的存取之间的最小空闲周期数

BSC 寄存器的设定		不超过总线宽度的存取		超过总线宽度的存取			
CSnWCR.WM 的设定	CSnBCR 的空闲设定	读 → 写	写 → 读	连续读 *1	读 → 写 *2	连续写 *1	写 → 读 *2
1	0	2	0	0	2	0	0
0	0	2	1	1	2	1	1
1	1	2	1	1	2	1	1
0	1	2	1	1	2	1	1
1	2	2	2	2	2	2	2
0	2	2	2	2	2	2	2
1	4	4	4	4	4	4	4
0	4	4	4	4	4	4	4

【注】 因为 DMAC 和 DTC 是通过  $B\phi$  运行，所以和时钟比没有关系。

\*1 表示在 16 位总线的 32 位存取中，地址 0 的字存取和地址 2 的字存取之间的最小空闲周期数；在 8 位总线的 16 位存取中，地址 0 的字节存取和地址 1 的字节存取之间的最小空闲周期数；在 8 位总线的 32 位存取中，地址 0 的字节存取、地址 1 的字节存取、地址 2 的字节存取和地址 3 的字节存取之间的最小空闲周期数；在 16 字节传送中的连续存取之间的最小空闲周期数

\*2 上述以外

表 9.31 从带 DACK 的外部器件到  
通常空间接口的 DMAC 单地址模式传送中的最小空闲周期数

• 从带 DACK 的外部器件到通常空间接口的传送

BSC 寄存器的设定 *3			最小空闲周期数	
CSnWCR.WM 的设定	CMNCR.DMAIWA 的设定	CMNCR.DMAIW 的空闲设定	超过总线宽度的存取 *1	不超过总线宽度的存取 *2
1	0	—	0	1*5
0	0	—	1	1
1	1	0	0	1*5
0	1	0	1	1
1	1	1	1	1
0	1	1	1	1
1	1	2	2	2
0	1	2	2	2
1	1	4	4	4
0	1	4	4	4

• 从通常空间接口到带 DACK 的外部器件的传送

BSC 寄存器的设定 *4		最小空闲周期数	
CSnWCR.WM 的设定	CSnBCR 的空闲设定	超过总线宽度的存取 *1	不超过总线宽度的存取 *2
1	0	0	2
0	0	1	3
1	1	1	2
0	1	1	3
1	2	2	2
0	2	2	3
1	4	4	4
0	4	4	4

【注】 因为 DMAC 是通过 B $\phi$  运行，所以和时钟比没有关系。

\*1 表示在 16 位总线的 32 位存取中，地址 0 的字存取和地址 2 的字存取之间的最小空闲周期数；在 8 位总线的 16 位存取中，地址 0 的字节存取和地址 1 的字节存取之间的最小空闲周期数；在 8 位总线的 32 位存取中，地址 0 的字节存取、地址 1 的字节存取、地址 2 的字节存取和地址 3 的字节存取之间的最小空闲周期数；在 16 字节传送中的连续存取之间的最小空闲周期数

\*2 上述以外

\*3 在从带 DACK 的外部器件到通常空间接口的单次传送中，不受 CSnBCR 的 IWW 位、IWRWD 位、IWRWS 位、IWRRD 位和 IWRRS 位的影响。

\*4 在从通常空间接口到带 DACK 的外部器件的单次传送中，不受 CMNCR 的 DMAIWA 位和 DMAIW 位的影响。

\*5 如果至少将 CSnWCR 的 HW[1:0] 位设定为 2.5 个周期，空闲周期数就为 0。

表 9.32 SDRAM 接口的 CPU、DMAC 双地址模式和 DTC 中的存取之间的最小空闲周期数

BSC 寄存器的设定			CPU 存取				DMAC 存取和 DTC 存取	
CSnBCR 的空闲设定	CS3WCR.WTRP 的设定	CS3WCR.TRWL 的设定	读 → 读	写 → 写	读 → 写	写 → 读	读 → 写	写 → 读
0	1	0	1,1,1,1	0,0,0,0	3,3,3,4	0,0,0,0	2	0
0	1	1	1,1,1,1	1,1,1,1	3,3,3,4	1,1,1,1	2	1
0	1	2	1,1,1,1	2,2,2,2	3,3,3,4	2,2,2,2	2	2
0	1	3	1,1,1,1	3,3,3,3	3,3,3,4	3,3,3,3	2	3
0	2	0	2,2,2,2	1,1,1,1	3,3,3,4	1,1,1,1	2	1
0	2	1	2,2,2,2	2,2,2,2	3,3,3,4	2,2,2,2	2	2
0	2	2	2,2,2,2	3,3,3,3	3,3,3,4	3,3,3,3	2	3
0	2	3	2,2,2,2	4,4,4,4	3,3,3,4	4,4,4,4	2	4
0	3	0	3,3,3,3	2,2,2,2	3,3,3,4	2,2,2,2	3	2
0	3	1	3,3,3,3	3,3,3,3	3,3,3,4	3,3,3,3	3	3
0	3	2	3,3,3,3	4,4,4,4	3,3,3,4	4,4,4,4	3	4
0	3	3	3,3,3,3	5,5,5,5	3,3,3,4	5,5,5,5	3	5
0	4	0	4,4,4,4	3,3,3,3	4,4,4,4	3,3,3,3	4	3
0	4	1	4,4,4,4	4,4,4,4	4,4,4,4	4,4,4,4	4	4
0	4	2	4,4,4,4	5,5,5,5	4,4,4,4	5,5,5,5	4	5
0	4	3	4,4,4,4	6,6,6,6	4,4,4,4	6,6,6,6	4	6
1	1	0	2,2,2,2	1,1,1,1	3,3,3,4	1,1,1,1	2	1
1	1	1	2,2,2,2	1,1,1,1	3,3,3,4	1,1,1,1	2	1
1	1	2	2,2,2,2	2,2,2,2	3,3,3,4	2,2,2,2	2	2
1	1	3	2,2,2,2	3,3,3,3	3,3,3,4	3,3,3,3	2	3
1	2	0	2,2,2,2	1,1,1,1	3,3,3,4	1,1,1,1	2	1
1	2	1	2,2,2,2	2,2,2,2	3,3,3,4	2,2,2,2	2	2
1	2	2	2,2,2,2	3,3,3,3	3,3,3,4	3,3,3,3	2	3
1	2	3	2,2,2,2	4,4,4,4	3,3,3,4	4,4,4,4	2	4
1	3	0	3,3,3,3	2,2,2,2	3,3,3,4	2,2,2,2	3	2
1	3	1	3,3,3,3	3,3,3,3	3,3,3,4	3,3,3,3	3	3
1	3	2	3,3,3,3	4,4,4,4	3,3,3,4	4,4,4,4	3	4
1	3	3	3,3,3,3	5,5,5,5	3,3,3,4	5,5,5,5	3	5
1	4	0	4,4,4,4	3,3,3,3	4,4,4,4	3,3,3,3	4	3
1	4	1	4,4,4,4	4,4,4,4	4,4,4,4	4,4,4,4	4	4
1	4	2	4,4,4,4	5,5,5,5	4,4,4,4	5,5,5,5	4	5
1	4	3	4,4,4,4	6,6,6,6	4,4,4,4	6,6,6,6	4	6

BSC 寄存器的设定			CPU 存取				DMAC 存取和 DTC 存取	
CSnBCR 的空闲设定	CS3WCR.WTRP 的设定	CS3WCR.TRWL 的设定	读 → 读	写 → 写	读 → 写	写 → 读	读 → 写	写 → 读
2	1	0	3,3,3,3	2,2,2,2	3,3,3,4	2,2,2,2	3	2
2	1	1	3,3,3,3	2,2,2,2	3,3,3,4	2,2,2,2	3	2
2	1	2	3,3,3,3	2,2,2,2	3,3,3,4	2,2,2,2	3	2
2	1	3	3,3,3,3	3,3,3,3	3,3,3,4	3,3,3,3	3	3
2	2	0	3,3,3,3	2,2,2,2	3,3,3,4	2,2,2,2	3	2
2	2	1	3,3,3,3	2,2,2,2	3,3,3,4	2,2,2,2	3	2
2	2	2	3,3,3,3	3,3,3,3	3,3,3,4	3,3,3,3	3	3
2	2	3	3,3,3,3	4,4,4,4	3,3,3,4	4,4,4,4	3	4
2	3	0	3,3,3,3	2,2,2,2	3,3,3,4	2,2,2,2	3	2
2	3	1	3,3,3,3	3,3,3,3	3,3,3,4	3,3,3,3	3	3
2	3	2	3,3,3,3	4,4,4,4	3,3,3,4	4,4,4,4	3	4
2	3	3	3,3,3,3	5,5,5,5	3,3,3,4	5,5,5,5	3	5
2	4	0	4,4,4,4	3,3,3,3	4,4,4,4	3,3,3,3	4	3
2	4	1	4,4,4,4	4,4,4,4	4,4,4,4	4,4,4,4	4	4
2	4	2	4,4,4,4	5,5,5,5	4,4,4,4	5,5,5,5	4	5
2	4	3	4,4,4,4	6,6,6,6	4,4,4,4	6,6,6,6	4	6
4	1	0	5,5,5,5	4,4,4,4	5,5,5,5	4,4,4,4	5	4
4	1	1	5,5,5,5	4,4,4,4	5,5,5,5	4,4,4,4	5	4
4	1	2	5,5,5,5	4,4,4,4	5,5,5,5	4,4,4,4	5	4
4	1	3	5,5,5,5	4,4,4,4	5,5,5,5	4,4,4,4	5	4
4	2	0	5,5,5,5	4,4,4,4	5,5,5,5	4,4,4,4	5	4
4	2	1	5,5,5,5	4,4,4,4	5,5,5,5	4,4,4,4	5	4
4	2	2	5,5,5,5	4,4,4,4	5,5,5,5	4,4,4,4	5	4
4	2	3	5,5,5,5	4,4,4,4	5,5,5,5	4,4,4,4	5	4
4	3	0	5,5,5,5	4,4,4,4	5,5,5,5	4,4,4,4	5	4
4	3	1	5,5,5,5	4,4,4,4	5,5,5,5	4,4,4,4	5	4
4	3	2	5,5,5,5	4,4,4,4	5,5,5,5	4,4,4,4	5	4
4	3	3	5,5,5,5	5,5,5,5	5,5,5,5	5,5,5,5	5	5
4	4	0	5,5,5,5	4,4,4,4	5,5,5,5	4,4,4,4	5	4
4	4	1	5,5,5,5	4,4,4,4	5,5,5,5	4,4,4,4	5	4
4	4	2	5,5,5,5	5,5,5,5	5,5,5,5	5,5,5,5	5	5
4	4	3	5,5,5,5	6,6,6,6	5,5,5,5	6,6,6,6	5	6

【注】表中的 CPU 存取部分按顺序表示  $t_{\phi} : B_{\phi} = 4:1$ 、 $3:1$ 、 $2:1$ 、 $1:1$  的情况。

因为 DMAC 和 DTC 是通过  $B_{\phi}$  运行，因此和时钟比无关。

表 9.33 SDRAM 接口的 DMAC 单地址模式中的  
存取之间的最小空闲周期数 (1)

- 从带 DACK 的外部器件到 SDRAM 接口的传送

BSC 寄存器的设定 *2			最小空闲数
CMNCR.DMAIW 的设定	CS3WCR.WTRP 的设定	CS3WCR.TRWL 的设定	
0	1	0	1*3
0	1	1	1
0	1	2	2
0	1	3	3
0	2	0	1
0	2	1	2
0	2	2	3
0	2	3	4
0	3	0	2
0	3	1	3
0	3	2	4
0	3	3	5
0	4	0	3
0	4	1	4
0	4	2	5
0	4	3	6
1	1	0	1
1	1	1	1
1	1	2	2
1	1	3	3
1	2	0	1
1	2	1	2
1	2	2	3
1	2	3	4
1	3	0	2
1	3	1	3
1	3	2	4
1	3	3	5
1	4	0	3
1	4	1	4
1	4	2	5
1	4	3	6
2	1	0	2

BSC 寄存器的设定 *2			最小空闲数
CMNCR.DMAIW 的设定	CS3WCR.WTRP 的设定	CS3WCR.TRWL 的设定	
2	1	1	2
2	1	2	2
2	1	3	3
2	2	0	2
2	2	1	2
2	2	2	3
2	2	3	4
2	3	0	2
2	3	1	3
2	3	2	4
2	3	3	5
2	4	0	3
2	4	1	4
2	4	2	5
2	4	3	6
4	1	0	4
4	1	1	4
4	1	2	4
4	1	3	4
4	2	0	4
4	2	1	4
4	2	2	4
4	2	3	4
4	3	0	4
4	3	1	4
4	3	2	4
4	3	3	5
4	4	0	4
4	4	1	4
4	4	2	5
4	4	3	6

表 9.34 SDRAM 接口的 DMAC 单地址模式中的  
存取之间的最小空闲周期数 (2)

- 从 SDRAM 接口到带 DACK 外部器件的传送

BSC 寄存器的设定 *2		最小空闲数
CS3BCR 的空闲设定	CS3WCR.WTRP 的设定	
0	1	3
0	2	3
0	3	3
0	4	4
1	1	3
1	2	3
1	3	3
1	4	4
2	1	3
2	2	3
2	3	3
2	4	4
4	1	5
4	2	5
4	3	5
4	4	5

【注】 因为 DMAC 是通过 B $\phi$  运行，因此和时钟比无关。

\*1 在从带 DACK 的外部器件到 SDRAM 接口的单次传送中，不受 CSnBCR 的 IWW 位、IWRWD 位、IWRWS 位、IWRRD 位、IWRRS 位的影响。  
当 CMNCR.DMAIWA=0 时，和表 9.33 中 CMNCR.DMAIW[1:0]=0 的设定相同。

\*2 上述以外

\*3 最多将 WTRCD 设定为 1 个周期。



### 9.5.13 总线仲裁

本 LSI 在通常状态下占有总线权，在接受外部器件的总线权请求后释放总线。另外，本 LSI 内部有 CPU、DMAC、DTC 共 3 个总线主控，释放给这些总线主控的总线权的优先级如下：

外部器件的总线权请求 ( $\overline{\text{BREQ}}$ ) > CPU > DTC > DMAC > CPU

但是在 DTC 和 DMAC 请求总线权的状态下，CPU 不能连续获得总线权。

有关 CPU 存取外部空间的请求，有以下 2 种情况：

1. 当总线功能扩展寄存器 (BSCEHR) 的 CSSTP2 位是 1 时，CPU 存取外部空间的请求优先级低于 DMAC 突发送请求和 DTC 发送请求 (总线功能扩展寄存器 (BSCEHR) 的 DTLOCK 位是 0 时)。
2. 在 CPU 存取外部空间时，如果按照 DMAC、DTC 的顺序产生启动请求，就优先执行 DMAC 传送。在 CPU 存取外部空间期间，DTC/DMAC 发生竞争时的总线仲裁如图 9.49 所示。

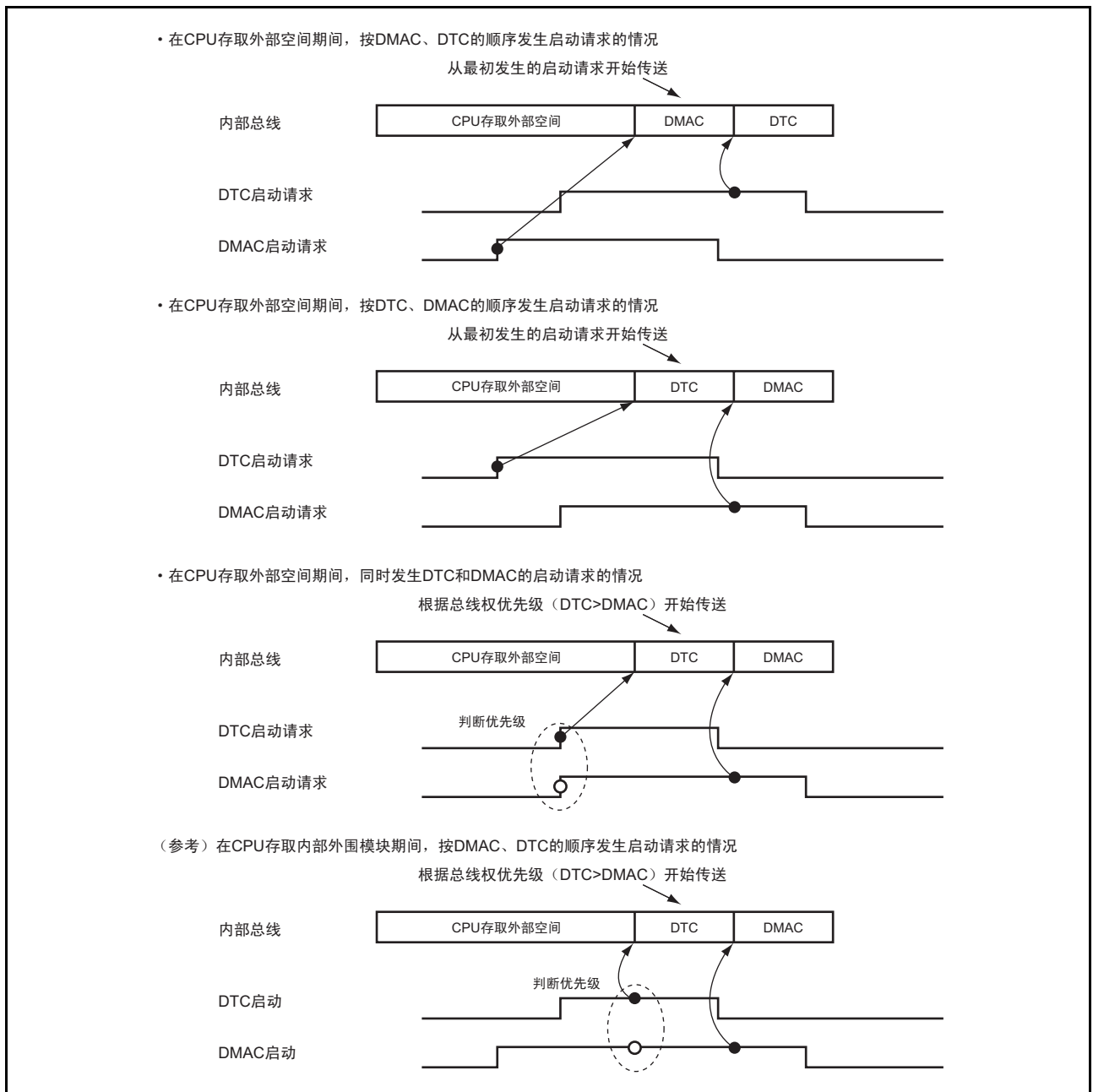


图 9.49 CPU 存取外部空间期间 DTC/DMAC 发生竞争时的总线仲裁

另外，对于“9.5.14(2) 从 LSI 内部总线主控器的存取”中记述的写缓冲器操作，在 CPU 读和写外部空间时，CPU 和 DTC/DMAC 的仲裁不同。在 CPU 存取外部空间期间，发生 DTC/DMAC 启动请求时的总线仲裁如图 9.50 所示。

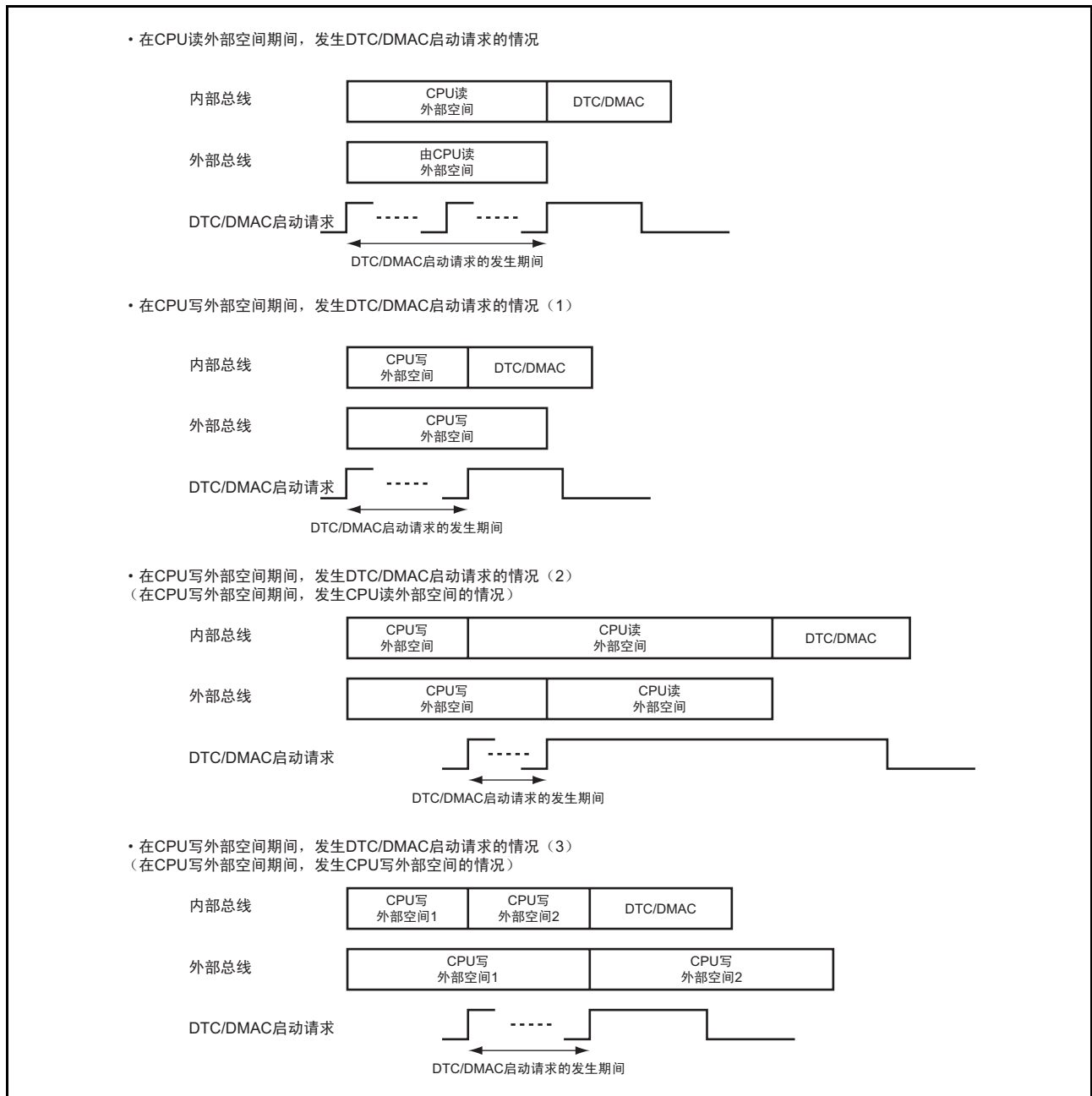


图 9.50 CPU 存取外部空间期间发生 DTC/DMAC 启动请求时的总线仲裁

不进行总线仲裁的状态如下所示：

1. 在 TAS 命令的读周期和写周期之间
2. 因数据总线宽度小于存取长度而产生的多个总线周期（例如，对 8 位数据总线的存储器进行长字存取时的总线周期之间）
3. 通过 DMAC 进行 16 字节传送时

在将总线权移交给外部器件时，为了防止外部器件的误动作，需要在释放总线前将全部总线控制信号置于无效状态。在接受总线权时，总线控制信号从无效状态开始驱动。为了防止在总线控制信号的高阻抗状态下外来噪声引起的误动作，这些控制信号需要上拉电阻。

在总线周期的边界处与外部器件进行总线权的交接。在有总线权请求时，如果总线周期不在执行，就立即释放总线权；如果总线周期正在执行中，就要等到总线周期结束后才释放总线权。从 LSI 外部来看，即使总线周期不在执行，也有可能由于在存取周期之间插入等待而在内部总线周期已经开始执行，所以不能从  $\overline{CSn}$  信号及其他总线控制信号来判断是否立即释放总线。

在通过  $\overline{BREQ}$  和  $\overline{BACK}$  的信号交换进行外部总线的释放时，需要不小的开销。在从属模块的任务较多的情况下，必须将系统设定为通过获得一次总线权来执行多次总线周期。通过减少交接总线权所需的周期，使系统设计变得简单。

只要不接受外部器件的总线权请求就保有总线权。当接受外部总线权请求  $\overline{BREQ}$  的有效电平（低电平）时，在执行中的总线周期结束后释放总线，并将总线使用的允许  $\overline{BACK}$  置为有效（低电平）。在接受表示外部器件已释放总线的  $\overline{BREQ}$  的无效电平（高电平）后，将  $\overline{BACK}$  置为无效（高电平），重新开始使用总线。

在使用 SDRAM 接口时，如果有激活的存储体，就产生全部存储体的预充电命令（PALL），在命令结束后释放总线。

另外，本 LSI 即使在向外部释放总线权中，只要不产生向外部器件的存取，处理就会继续。之后，当向外部器件的存取产生时，就变为等待总线权恢复的状态。

在释放总线期间，不能转移到睡眠模式、软件待机模式和深度软件待机模式。

具体的总线释放顺序如下：

首先与 CK 的上升沿同步，将地址总线 and 数据总线设定为高阻抗，在此 0.5 个周期后与 CK 的下降沿同步，将总线使用允许信号置为有效，然后在 CK 的下一个上升沿，将  $\overline{CSn}$  等总线控制信号设定为高阻抗。这些总线控制信号至少在变为高阻抗的 1 个周期前被设定为高电平。在 CK 的下降沿对总线权请求信号进行采样。另外，通过设定 CMNCR 的 HIZCNT 位，即使在释放总线时，也能通过释放前的值连续驱动  $\overline{CKE}$ 、 $\overline{RASU}$ 、 $\overline{RASL}$ 、 $\overline{CASU}$ 、 $\overline{CASL}$ 。

从外部器件重新获得总线权的顺序如下：

如果在 CK 的下降沿检测到  $\overline{BREQ}$  无效，就在 1.5 个周期后用高电平开始驱动总线控制信号，接着在 CK 的下一个下降沿，将总线使用的允许信号设定为无效，然后在 CK 的上升沿，开始驱动地址总线 and 数据总线。在总线控制信号置为有效后到实际开始总线周期，最快的情况是在 CK 上升沿（与驱动地址和数据信号相同）。总线仲裁时序如图 9.51 所示。

为了减少由仲裁产生的开销，根据用户各自设计的外部器件，有可能连续发生多次总线存取。在这种情况下，为了保证 SDRAM 的刷新，需要进行在外部器件的总线占有时间不超过刷新周期的情况下释放总线权的设计。在此，本 LSI 的  $\overline{IRQOUT}$  引脚用作在等待执行刷新的期间请求总线权的信号，而且在获得总线权前（等待执行刷新的期间） $\overline{IRQOUT}$  保持有效（低电平）。如果在外部器件接受了总线权请求后释放总线权，总线权就能还给本 LSI 并执行刷新。

如果发生了总线权请求（ $\overline{BREQ}$  为低电平有效），就必须在允许使用总线（ $\overline{BACK}$  为低电平有效）后，释放总线权（ $\overline{BREQ}$  为高电平无效）。如果在  $\overline{BACK}$  有效前将  $\overline{BREQ}$  置为无效，根据  $\overline{BREQ}$  无效时序， $\overline{BACK}$  只有 1 个周期的有效，就有可能在外部器件和本 LSI 之间产生总线冲突。

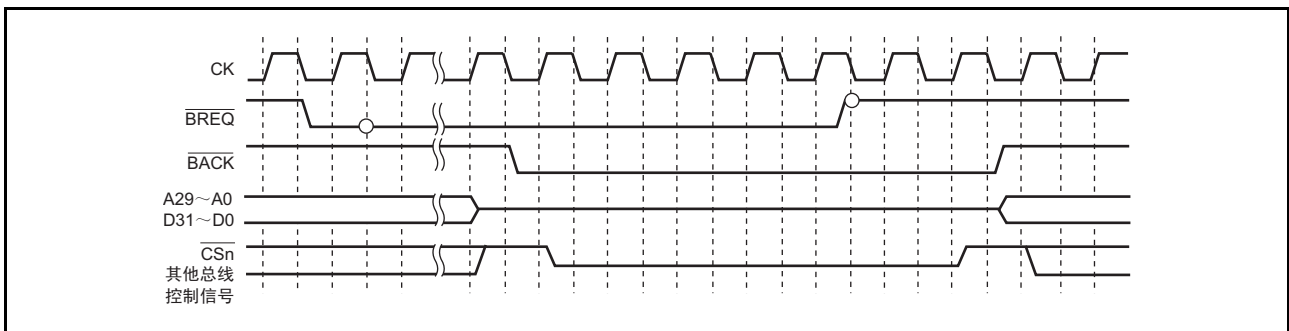


图 9.51 总线仲裁

在向 DMAC 执行总线仲裁时，因为在接收 DMAC 的总线权处理时需要  $IB\phi$  时间，所以在 I 总线上就会插入  $IB\phi$  时间的 NOP。

在向 TDC 执行总线仲裁时，因为在接收 DTC 的总线权处理时不需要插入 NOP 等，所以持续进行总线存取的处理。

### 9.5.14 其他

#### (1) 复位

只在上电复位时才对总线状态控制器进行完全的初始化。在上电复位时，与总线周期是否在执行中无关，立即将全部信号置为无效，关闭输出缓冲器，并且对全部控制寄存器进行初始化。在待机模式、睡眠模式和手动复位时，不对总线状态控制器的控制寄存器进行初始化。如果进行手动复位，就只在当前正在执行中的总线周期结束前执行该总线周期，然后进入存取等待状态。在执行 DMAC 的 16 字节传送时，为了取消手动复位引起的总线主控的存取请求，以长字为单位中止存取。在手动复位信号有效时，RTCNT 也能递增计数，所以能产生刷新并启动刷新周期，但是不接受 BREQ 产生的仲裁请求。

#### (2) 从 LSI 内部总线控制器的存取

本 LSI 内部分为 L 总线、I 总线和外围总线共 3 种总线。CPU 连接 L 总线，DMAC、DTC 和总线状态控制器连接 I 总线，低速外围模块连接外围总线。另外，内部存储器连接 L 总线和 I 总线。

在存取外部空间和内部外围模块时，通过 I 总线启动存取周期。因此，在 CPU 存取内部存储器时，能不发生和 CPU 的总线仲裁而启动 DMAC/DTC。

因为总线状态控制器有一段写缓冲器，所以即使在写周期芯片外部的总线周期没有结束，I 总线也能用于其他的存取。如果在写芯片外部的低速存储器后读写内部外围模块，有可能不等待低速存储器的写操作结束就存取内部外围模块。

因为 CPU 总是等到操作结束才开始读，所以要在确认数据已写到实际器件后继续处理时，只需连续对相同地址进行虚读，就能确认写操作的结束。

对于 DMAC 或者 DTC 的存取，总线状态控制器的写缓冲器也同样起作用。因此，在进行双地址的 DMA 传送时，不等待写周期结束就启动下一次读周期。但是，当 DMA 的源地址和目标地址都在外部存储器空间时，就等到前一次写周期结束后才开始下一次读周期。

如果在写缓冲器时更改 BSC 的寄存器就不能正确地存取，所以不能在写存取后立即更改 BSC 的寄存器。如果需要在写存取后立即更改 BSC 的寄存器，就必须进行写数据的虚拟读，并在确认写存取结束后更改 BSC 的寄存器。

### 9.5.15 从 CPU 向内部 FLASH 和内部 RAM 的存取

内部 FLASH 的读出和  $I_{\phi}$  时钟同步，通过 1 个时钟进行存取。关于写编程、擦除请参照“第 23 章 闪存”。内部 RAM 的读 / 写和  $I_{\phi}$  时钟同步，通过 1 个时钟进行存取。详细内容请参照“第 25 章 RAM”。

### 9.5.16 从 CPU 向内部外围 I/O 寄存器的存取

从 CPU 向内部外围 I/O 寄存器存取时的存取周期数如表 9.35 所示。

表 9.35 向内部外围 I/O 寄存器的存取周期数

	存取周期数
写	$(3+n) \times I\phi + (1+m) \times B\phi + 2 \times P\phi$
读	$(3+n) \times I\phi + (1+m) \times B\phi + 2 \times P\phi + 2 \times I\phi$

- 【注】**
- $I\phi:B\phi=8:1$  时,  $n=0 \sim 7$   
 $I\phi:B\phi=4:1$  时,  $n=0 \sim 3$   
 $B\phi:P\phi=8:1$  时,  $m=0 \sim 7$   
 $I\phi:B\phi=3:1$  时,  $n=0 \sim 2$   
 $B\phi:P\phi=3:1$  时,  $m=0 \sim 2$   
 $I\phi:B\phi=2:1$  时,  $n=0 \sim 1$   
 $B\phi:P\phi=2:1$  时,  $m=0 \sim 1$   
 $I\phi:B\phi=1:1$  时,  $n=0$   
 $B\phi:P\phi=1:1$  时,  $m=0$   
 $n、m$  取决于内部的执行状态。
  - $MI\phi、MP\phi$  的时钟比不影响存取周期数。

本产品采用同步式伦理，总线由分层的总线结构构成。向各个总线输入或输出数据时，L 总线和  $I\phi$  时钟的上升同步进行，I 总线和  $B\phi$  时钟的上升同步进行，外围总线和  $P\phi$  时钟的上升同步进行。当  $I\phi:B\phi:P\phi=4:2:2$ ，且在外围总线的连接宽度为 16 位、 $P\phi$  的 2 个周期的存取寄存器上对字节长度的数据进行写存取的时序例子如图 9.52 所示。CPU 存取内部外围 I/O 寄存器时，向 L 总线输出数据后，向 I 总线传送数据时需要  $I\phi$  的 3 个周期的准备时间。在这 3 个周期后，就可以和  $B\phi$  的上升沿同步向 I 总线传送数据。但是在  $I\phi:B\phi=4:2$  时， $B\phi$  的 1 时钟之间  $I\phi$  有 2 个时钟。因此  $I\phi:B\phi=4:2$  时，在  $(3+n) \times I\phi、n=0 \sim 1$  期间内从 L 总线传送到 I 总线（例中为  $3 \times I\phi$ ）。数据出现在 L 总线上的时序和  $B\phi$  上升沿的时序间的关系取决于程序的执行状态。在例中因为  $n=0、m=0$ ，所以存取的时间为  $3 \times I\phi + 1 \times B\phi + 2 \times P\phi$ 。

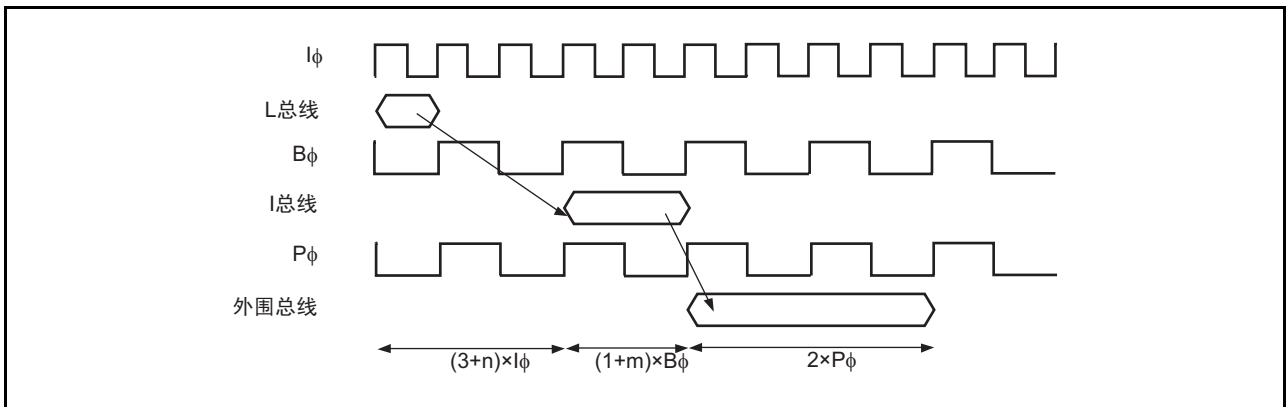


图 9.52  $I\phi:B\phi:P\phi=4:2:2$  时向内部外围 I/O 寄存器写存取的时序

当  $I\phi:B\phi:P\phi=4:2:1$  时向外围总线读存取的时序例子如图 9.53 所示。从 L 总线到外围总线的传送和写时相同，但是读时需要向 CPU 传送在外围总线上读出的值。从外围总线到 I 总线、从 I 总线到 L 总线的传送也和各个总线时钟的上升同步进行。但是因为  $I\phi \geq B\phi \geq P\phi$ ，所以实际上需要  $2 \times I\phi$  的时间。在例中由于  $n=0、m=1$ ，所以存取时间为  $3 \times I\phi + 2 \times B\phi + 2 \times P\phi + 2 \times I\phi$ 。

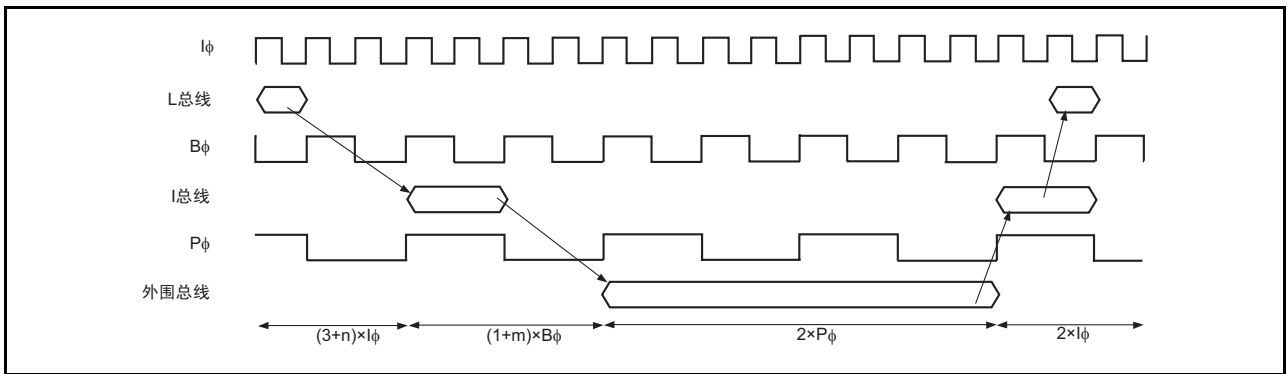


图 9.53  $l\phi:B\phi:P\phi=4:2:1$  时向内部外围 I/O 寄存器读存取的时序

### 9.5.17 从 CPU 向外部存储器的存取

CPU 外部存取时的存取周期数如表 9.36 所示。外部存取周期不仅根据表 9.36 里显示出的时钟比、存取长度、LSI 的外部总线宽度，而且根据等待插入等设定而变化。等待插入等设定的详细内容请参照“9.4 寄存器说明”。

表 9.36 外部存取周期数

外部总线宽度	存取长度	写 / 读	存取周期数
8 位	字节	写	$(1+n)\times l\phi+(3+m)\times B\phi$
		读	$(1+n)\times l\phi+(3+m)\times B\phi+1\times l\phi$
	字	写	$(1+n)\times l\phi+(3+m)\times B\phi+1\times(2+o)\times B\phi$
		读	$(1+n)\times l\phi+(3+m)\times B\phi+1\times(2+o)\times B\phi+1\times l\phi$
	长字	写	$(1+n)\times l\phi+(3+m)\times B\phi+3\times(2+o)\times B\phi$
		读	$(1+n)\times l\phi+(3+m)\times B\phi+3\times(2+o)\times B\phi+1\times l\phi$
16 位	字节 / 字	写	$(1+n)\times l\phi+(3+m)\times B\phi$
		读	$(1+n)\times l\phi+(3+m)\times B\phi+1\times l\phi$
	长字	写	$(1+n)\times l\phi+(3+m)\times B\phi+1\times(2+o)\times B\phi$
		读	$(1+n)\times l\phi+(3+m)\times B\phi+1\times(2+o)\times B\phi+1\times l\phi$
32 位	字节 / 字 / 长字	写	$(1+n)\times l\phi+(3+m)\times B\phi$
		读	$(1+n)\times l\phi+(3+m)\times B\phi+1\times l\phi$

【注】 n:  $l\phi:B\phi=8:1$  时,  $n=0\sim 7$   
 $l\phi:B\phi=4:1$  时,  $n=0\sim 3$   
 $l\phi:B\phi=3:1$  时,  $n=0\sim 2$   
 $l\phi:B\phi=2:1$  时,  $n=0\sim 1$   
 $l\phi:B\phi=1:1$  时,  $n=0$

m、o: m: 等待设定, o: 等待设定 + 空闲设定等。  
 详细内容请参照“9.4 寄存器说明”。

本产品采用同步式伦理，总线由分层的总线结构构成。向各个总线输入或输出数据时，L 总线和 I $\phi$  时钟的上升同步进行，I 总线、外部总线和 B $\phi$  时钟的上升同步进行。当 I $\phi$ :B $\phi$ =2:1，且用宽度为 8 位的外部总线对字数据进行写存取时的时序例子如图 9.54 所示。CPU 向 L 总线输出数据后，和 B $\phi$  的上升沿同步向 I 总线传送数据。I $\phi$ :B $\phi$ =2:1 时，B $\phi$  的 1 个时钟之间 I $\phi$  有 2 个时钟。因此 I $\phi$ :B $\phi$ =2:1 时，在 (1+n)  $\times$  I $\phi$ 、n=0 ~ 1 期间内，数据从 L 总线传送到 I 总线（例中为 2  $\times$  I $\phi$ ）。数据出现在 L 总线上的时序和 B $\phi$  上升沿的时序之间的关系取决于程序的执行状态。出现在 I 总线上的数据，在 B $\phi$  的 1 个周期后传送到外部总线。进行 1 回外部存取最快需要 2 个周期，但是通过在 BSC 寄存器上的设定可以延长此时间（存取周期计算式中的 m、o）。在图 9.54 中，对总线宽度为 8 位的总线进行写字数据，所以需要 2 回的外部存取。又因为例中在 n=0、m=0，所以存取的时间为 2  $\times$  I $\phi$ +3  $\times$  B $\phi$ +2  $\times$  B $\phi$ 。

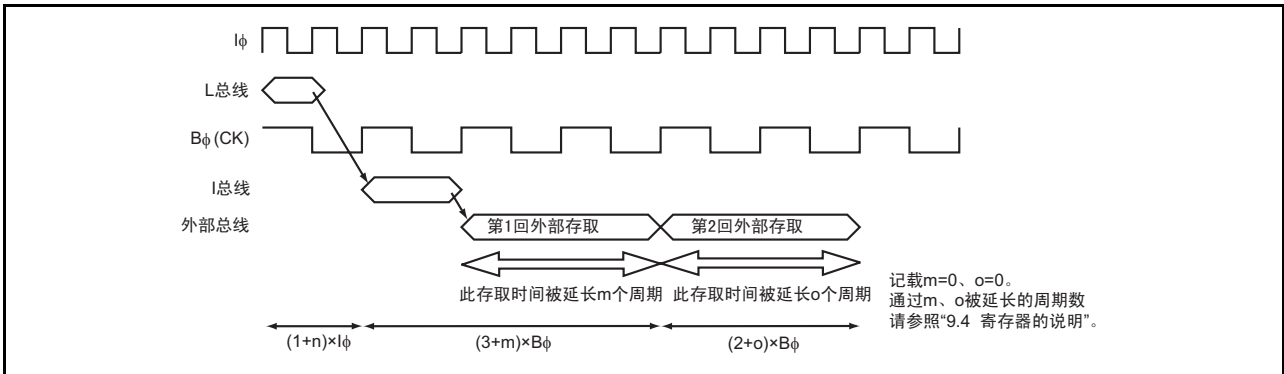


图 9.54 I $\phi$ :B $\phi$ =2:1、且外部总线宽度为 8 位时，将字数据写存取到外部存储器的时序

在 I $\phi$ :B $\phi$ =4:1、且外部总线宽度  $\geq$  数据宽度的条件下读存取时的时序例子如图 9.55 所示。从 L 总线到外部总线的传送和写时相同，但是读时需要向 CPU 传送在外部总线上读出的值。从外部总线到 I 总线、从 I 总线到 L 总线的传送也和各个总线时钟的上升同步进行。但是在实际的运行中，从外部总线到 L 总线的传送需要 I $\phi$  的时间。在例中由于 n=2、m=0、o=0，所以存取时间为 3  $\times$  I $\phi$ +3  $\times$  B $\phi$ +1  $\times$  I $\phi$ 。

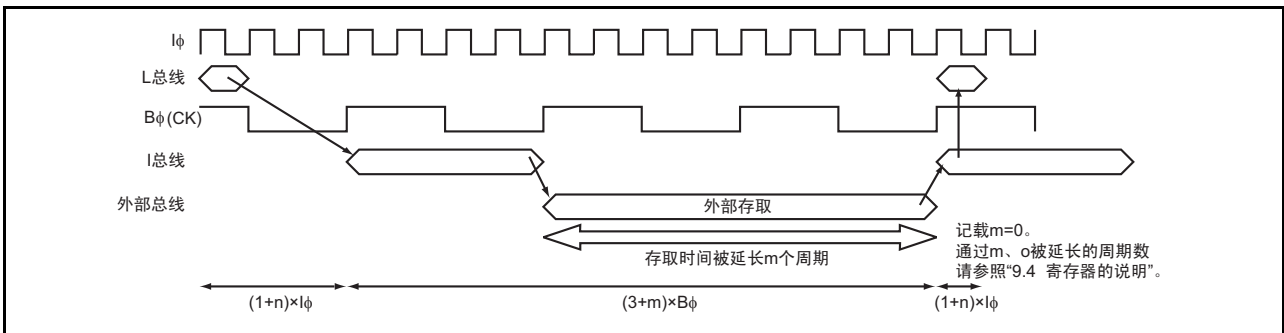


图 9.55 在 I $\phi$ :B $\phi$ =4:1、且外部总线宽度  $\geq$  数据宽度的条件下读存取时的时序

从 CPU 的存取周期数中删除 L 总线存取时所需要的 I $\phi$  量就是通过 DMAC、DTC 进行存取的周期数。



## 第 10 章 直接存储器存取控制器 (DMAC)

本 LSI 内置直接存储器存取控制器 (DMAC)。DMAC 能代替 CPU 在带 DACK (传送请求的应答信号) 的外部器件、外部存储器、内部存储器、被映像为存储器的外部器件以及内部外围模块之间进行高速数据传送。

### 10.1 特点

- 通道数: 4 个通道 (能接受外部请求)
- 地址空间: 在体系结构上为 4G 字节
- 传送数据长: 字节、字 (2 字节)、长字 (4 字节)、16 字节 (长字 ×4)
- 最大传送次数: 16,777,216 次
- 地址模式: 可选择单地址模式或者双地址模式
- 传送请求:  
可选择外部请求、内部外围模块请求或者自动请求
- 总线模式:  
可选择周期挪用模式 (通常模式和间歇模式) 或者突发模式
- 优先级: 可选择通道优先级的固定模式或者周期模式
- 中断请求: 能在数据传送结束时向 CPU 请求中断
- 外部请求的检测:  
可选择 DREQ 输入的低 / 高电平检测或者上升沿 / 下降沿检测
- 传送请求的接受信号: DACK 能独立设定有效电平

DMAC 的框图如图 10.1 所示。

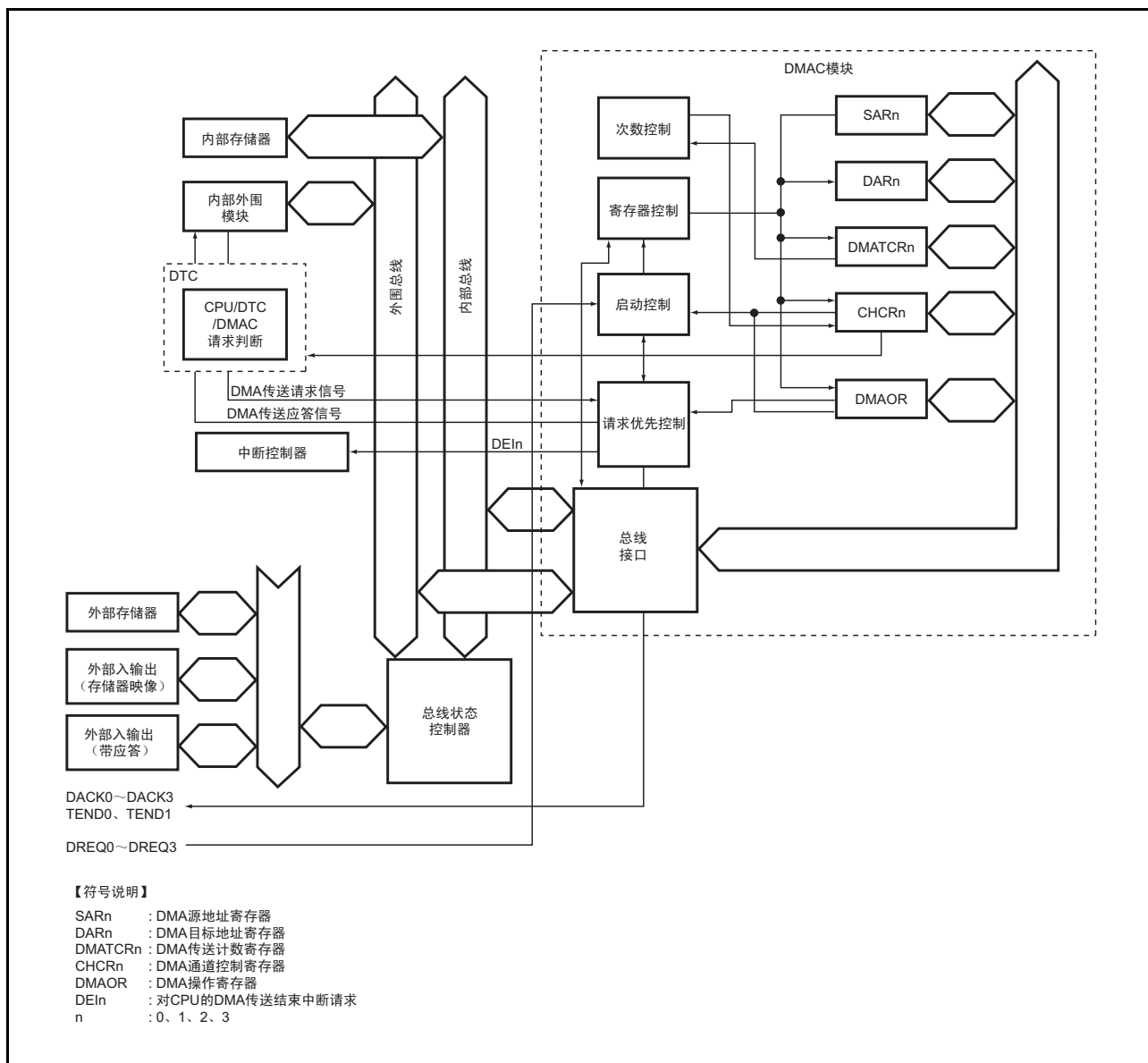


图 10.1 DMAC 的框图

## 10.2 输入 / 输出引脚

和 DMAC 相关的外部引脚如下所示。

连接外部总线的引脚结构如表 10.1 所示。DMAC 有 4 个通道的引脚用于外部总线。

表 10.1 外部总线的引脚结构

通道	名称	引脚名	输入 / 输出	功 能
0	DMA 传送请求	DREQ0	输入	将 DMA 传送请求从外部器件输入到通道 0
	DMA 传送请求的接受	DACK0	输出	将 DMA 传送请求的接受从 DMAC 通道 0 输出到外部器件
	DMA 传送结束	TEND0	输出	输出 DMAC 通道 0 的 DMA 传送结束
1	DMA 传送请求	DREQ1	输入	将 DMA 传送请求从外部器件输入到通道 1
	DMA 传送请求的接受	DACK1	输出	将 DMA 传送请求的接受从 DMAC 通道 1 输出到外部器件
	DMA 传送结束	TEND1	输出	输出 DMAC 通道 1 的 DMA 传送结束
2	DMA 传送请求	DREQ2	输入	将 DMA 传送请求从外部器件输入到通道 2
	DMA 传送请求的接受	DACK2	输出	将 DMA 传送请求的接受从 DMAC 通道 2 输出到外部器件
3	DMA 传送请求	DREQ3	输入	将 DMA 传送请求从外部器件输入到通道 3
	DMA 传送请求的接受	DACK3	输出	将 DMA 传送请求的接受从 DMAC 通道 3 输出到外部器件

## 10.3 寄存器说明

DMAC 有以下寄存器。有关这些寄存器地址和各处理模式的寄存器状态，请参照“第 27 章 寄存器一览”。如同通道 0 的 SAR 记为 SAR\_0 一样，记载各通道的寄存器。

表 10.2 寄存器结构

通道	寄存器名	略称	R/W	初始值	地址	存取长度
0	DMA 源地址寄存器_0	SAR_0	R/W	H'00000000	H'FFFFEB20	16、32
	DMA 目标地址寄存器_0	DAR_0	R/W	H'00000000	H'FFFFEB24	16、32
	DMA 传送计数寄存器_0	DMATCR_0	R/W	H'00000000	H'FFFFEB28	16、32
	DMA 通道控制寄存器_0	CHCR_0	R/W	H'00000000	H'FFFFEB2C	8、16、32
1	DMA 源地址寄存器_1	SAR_1	R/W	H'00000000	H'FFFFEB30	16、32
	DMA 目标地址寄存器_1	DAR_1	R/W	H'00000000	H'FFFFEB34	16、32
	DMA 传送计数寄存器_1	DMATCR_1	R/W	H'00000000	H'FFFFEB38	16、32
	DMA 通道控制寄存器_1	CHCR_1	R/W	H'00000000	H'FFFFEB3C	8、16、32
2	DMA 源地址寄存器_2	SAR_2	R/W	H'00000000	H'FFFFEB40	16、32
	DMA 目标地址寄存器_2	DAR_2	R/W	H'00000000	H'FFFFEB44	16、32
	DMA 传送计数寄存器_2	DMATCR_2	R/W	H'00000000	H'FFFFEB48	16、32
	DMA 通道控制寄存器_2	CHCR_2	R/W	H'00000000	H'FFFFEB4C	8、16、32

通道	寄存器名	略称	R/W	初始值	地址	存取长度
0	DMA 源地址寄存器_3	SAR_3	R/W	H'00000000	H'FFFFEB50	16、32
	DMA 目标地址寄存器_3	DAR_3	R/W	H'00000000	H'FFFFEB54	16、32
	DMA 传送计数寄存器_3	DMATCR_3	R/W	H'00000000	H'FFFFEB58	16、32
	DMA 通道控制寄存器_3	CHCR_3	R/W	H'00000000	H'FFFFEB5C	8、16、32
通用	DMA 操作寄存器	DMAOR	R/W	H'0000	H'FFFFEB60	8、16
	总线功能扩展寄存器	BSCEHR	R/W	H'0000	H'FFFFE89A	8、16

### 10.3.1 DMA 源地址寄存器\_0 ~ 3 (SAR\_0 ~ 3)

SAR 是 32 位可读写寄存器，指定传送源的地址，在 DMA 运行中表示下一个传送源地址。在单地址模式中，当传送源是带 DACK 的外部器件时，忽视 SAR。

在进行 16 位或者 32 位的数据传送时，必须分别指定 16 位和 32 位的边界地址。在以 16 字节为单位进行传送时，必须给 16 字节的边界设定值。

SAR 的初始值不定。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

### 10.3.2 DMA 目标地址寄存器\_0 ~ 3 (DAR\_0 ~ 3)

DAR 是 32 位可读写寄存器，指定传送目标地址，在 DMAC 运行中表示下一个传送目标地址。在单地址模式中，当传送目标是带 DACK 的外部器件时，忽视 DAR。

在进行 16 位或者 32 位的数据传送时，必须分别指定 16 位和 32 位的边界地址。在以 16 字节为单位进行传送时，必须给 16 字节的边界设定值。

DAR 的初始值不定。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

### 10.3.3 DMA 传送计数寄存器\_0 ~ 3 (DMATCR\_0 ~ 3)

DMATCR 是 32 位可读写寄存器，指定 DMA 的传送次数。当设定值是 H'00000001 时，传送次数是 1 次；当设定值是 H'00FFFFFF 时，传送次数是 16,777,215 次；当设定值是 H'00000000 时，传送次数是 16,777,216 次（最大传送次数）。在 DMA 传送中，DMATCR 表示剩余的传送次数。

DMATCR 高 8 位的读写值总是 0。

在进行 16 字节传送时，传送 1 次 16 字节（128 位）就进行 1 次计数。

DMATCR 的初始值不定。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

### 10.3.4 DMA 通道控制寄存器\_0 ~ 3 (CHCR\_0 ~ 3)

CHCR 是 32 位可读写寄存器，控制 DMA 的传送模式。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	DO	TL	-	-	-	-	AM	AL
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DM[1:0]		SM[1:0]		RS[3:0]			DL	DS	TB	TS[1:0]		IE	TE	DE	
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/(W)*	R/W

【注】\* 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说 明
31 ~ 24	—	全 0	R	保留位 读写的值总是 0。
23	DO	0	R/W	DMA 溢出 选择是通过超出 0 还是通过超出 1 检测 DREQ。 0: 通过超出 0 检测 DREQ 1: 通过超出 1 检测 DREQ
22	TL	0	R/W	传送结束电平 指定 TEND 信号是高电平有效还是低电平有效。 0: TEND 为低电平输出 1: TEND 为高电平输出
21、20	—	全 0	R	保留位 读写的值总是 0。
19	—	0	R	保留位 当启动 DMAC 时，值不定。

位	位名	初始值	R/W	说 明
18	—	0	R	保留位 读写的值总是 0。
17	AM	0	R/W	应答模式 选择在双地址模式中，是在数据读周期还是在数据写周期输出 DACK。 在单地址模式中，和此位的指定无关，随时输出 DACK。 0: 在读周期中输出 DACK (双地址模式) 1: 在写周期中输出 DACK (双地址模式)
16	AL	0	R/W	应答电平 指定 DACK 信号是高电平有效还是低电平有效。 0: DACK 为低电平输出 1: DACK 为高电平输出
15、14	DM[1:0]	00	R/W	目标地址模式 1、0 指定 DMA 传送目标地址的增减 (在单地址模式中，当数据传送到带 DACK 的外部器件时，忽视 DM1 位和 DM0 位)。 00: 目标地址固定 (以 16 字节为单位传送时，禁止设定) 01: 目标地址增加 (以字节为单位传送时 DAR+1、以字为单位传送时 DAR+2、以长字为单位传送时 DAR+4、以 16 字节为单位传送时 DAR+16) 10: 目标地址减少 (以字节为单位传送时 DAR-1、以字为单位传送时 DAR-2、以长字为单位传送时 DAR-4、以 16 字节为单位传送时禁止设定) 11: 禁止设定
13、12	SM[1:0]	00	R/W	源地址模式 1、0 指定 DMA 传送源地址的增减 (在单地址模式中，当从带 DACK 的外部器件传送数据时，忽视 SM1 位和 SM0 位)。 00: 源地址固定 (当进行 16 字节传送时，禁止设定) 01: 源地址增加 (以字节为单位传送时 SAR+1，以字为单位传送时 SAR+2，以长字为单位传送时 SAR+4，以 16 字节传送时 SAR+16) 10: 源地址减少 (以字节为单位传送时 SAR-1，以字为单位传送时 SAR-2，以长字为单位传送时 SAR-4，以 16 字节传送时禁止设定) 11: 禁止设定

位	位名	初始值	R/W	说 明
11 ~ 8	RS[3:0]	0000	R/W	<p>资源选择 3 ~ 0 指定传送请求源。必须在 DMA 允许位 (DE) 是 0 的状态下更改传送请求源。</p> <p>0000: 外部请求、双地址模式 0001: 禁止设定 0010: 外部请求、单地址模式 外部地址空间 → 带 DACK 的外部器件 0011: 外部请求、单地址模式 带 DACK 的外部器件 → 外部地址空间 0100: 自动请求 0101: 禁止设定 0110: MTU2 (TGIA_0) 0111: MTU2 (TGIA_1) 1000: MTU2 (TGIA_2) 1001: MTU2 (TGIA_3) 1010: MTU2 (TGIA_4) 1011: A/D_1 (ADI_1) 1100: SCI_0 (TXI_0) 1101: SCI_0 (RXI_0) 1110: SCI_1 (TXI_1) 1111: SCI_1 (RXI_1)</p>
7 6	DL DS	0 0	R/W R/W	<p>DREQ 电平、DREQ 边沿选择 选择 DREQ 输入的检测方法和检测电平。 如果将传送请求源指定为内部外围模块或者自动请求，此位无效。</p> <p>00: 低电平检测 01: 下降沿检测 10: 高电平检测 11: 上升沿检测</p>
5	TB	0	R/W	<p>传送总线模式 选择 DMA 传送的总线模式。</p> <p>0: 周期挪用模式 1: 突发模式</p> <p><b>【注】</b> 在以 MTU2 为启动请求源的突发模式中进行 DMAC 传送时，请设定“9.4.8 总线功能扩展寄存器 (BSCEHR)”的相应的 DMMTU4 ~ 0 位。</p>
4、3	TS[1:0]	00	R/W	<p>传送长度 1、0 选择 DMA 的传送单位。当传送源或者传送目标是已指定传送长度的内部外围模块的寄存器时，必须选择该传送长度。</p> <p>00: 以字节为单位 01: 以字 (2 字节) 为单位 10: 以长字 (4 字节) 为单位 11: 以 16 字节为单位 (4 次长字传送)</p>

位	位名	初始值	R/W	说 明
2	IE	0	R/W	<p>中断允许</p> <p>指定在 DMA 传送结束时是否向 CPU 请求中断。如果将 IE 位设定为 1 并且 TE 位被置位, 就向 CPU 请求中断 (DEI)。</p> <p>0: 禁止中断请求 1: 允许中断请求</p>
1	TE	0	R/(W)*	<p>传送结束标志</p> <p>当 DMA 传送计数寄存器 (DMATCR) 的值是 0 并且 DMA 传送结束时, TE 位就被置 1。当 DMATCR 的值不是 0 时, 如果因 NMI 中断或者 DMA 地址错误引起传送结束以及因清除 DE 位和 DMA 操作寄存器 (DMAOR) 的 DME 位而结束传送, TE 位就不被置位。为了清除 TE 位, 必须在读 TE 位的 1 后写 0。</p> <p>当 TE 位已被置位时, 即使将 DE 位设定为 1, 也不允许传送。</p> <p>0: DMA 传送中或者 DMA 传送中断 [清除条件]</p> <ul style="list-style-type: none"> <li>在读 TE 位的 1 后写 0</li> </ul> <p>1: DMA 传送结束 (因为 DMATCR=0)</p>
0	DE	0	R/W	<p>DMA 允许</p> <p>允许或者禁止 DMA 传送。在自动请求模式中, 如果将 DE 位和 DMAOR 的 DME 位设定为 1, 就开始传送。但是, TE 位、DMAOR 的 NMIF 位和 AE 位必须全部为 0。在外部请求模式和外围模块请求模式中, 如果在 DE 位和 DME 位被置 1 后还有相应器件或者相应外围模块的 DMA 传送请求, 就开始传送。但是, 此时和自动请求模式一样, TE 位、NMIF 位和 AE 位必须全部为 0。当将 DE 位清 0 时, 能中断传送。</p> <p>0: 禁止 DMA 传送 1: 允许 DMA 传送</p>

【注】 \* 为了清除标志, 只能在读 1 后写 0。



## 10.3.5 DMA 操作寄存器 (DMAOR)

DMAOR 是 16 位可读写寄存器，指定 DMA 传送时的通道优先级，也表示 DMA 的传送状态。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	CMS[1:0]		-	-	PR[1:0]		-	-	-	-	-	AE	NMIF	DME
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R	R	R	R/(W)*	R/(W)*	R/W

【注】\* 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说 明
15、14	—	全 0	R	保留位 读写的值总是 0。
13、12	CMS[1:0]	00	R/W	周期挪用模式的选择 1、0 选择周期挪用模式中的通常模式或者间歇模式。 为了将间歇模式置为有效，全部通道的总线模式必须都是周期挪用模式。 00: 通常模式 01: 禁止设定 10: 间歇模式 16 外部总线时钟的每 16 个时钟执行 1 次 DMA 传送 11: 间歇模式 64 外部总线时钟的每 64 个时钟执行 1 次 DMA 传送
11、10	—	全 0	R	保留位 读写的值总是 0。
9、8	PR[1:0]	00	R/W	优先级模式 1、0 如果多个通道同时发生传送请求，此位决定要执行的通道优先级。 00: CH0 > CH1 > CH2 > CH3 01: CH0 > CH2 > CH3 > CH1 10: 禁止设定 11: 周期模式
7 ~ 3	—	全 0	R	保留位 读写的值总是 0。
2	AE	0	R/(W)*	地址错误标志 表示已发生由 DMAC 引起的地址错误。如果 AE 位被置位，即使将 CHCR 的 DE 位和 DMAOR 的 DME 位设定为 1，也不允许 DMA 传送。 为了清除 AE 位，必须在读 AE 位的 1 后写 0。 0: 没有 DMAC 引起的地址错误 [清除条件] • 在读 AE 位的 1 后写 0 1: 发生 DMAC 引起的地址错误

位	位名	初始值	R/W	说 明
1	NMIF	0	R/(W)*	<p>NMI 标志</p> <p>表示已发生 NMI 中断。如果 NMIF 位被置位，即使将 CHCR 的 DE 位和 DMAOR 的 DME 位设定为 1，也不允许 DMA 传送。为了清除 NMIF 位，必须在读 NMIF 位的 1 写 0。</p> <p>当 NMI 输入时，对于正在执行中的 DMA 传送，只能进行 1 个传送单位的传送。如果 DMAC 不在运行，即使输入 NMI 中断，也将 NMIF 位设定为 1。</p> <p>0: 没发生 NMI 中断</p> <p>[清除条件]</p> <ul style="list-style-type: none"> <li>在读 NMIF 位的 1 后写 0</li> <li>1: 发生 NMI 中断</li> </ul>
0	DME	0	R/W	<p>DMA 主控的允许</p> <p>允许或者禁止全部通道的 DMA 传送。如果将 DME 位和 CHCR 的 DE 位设定为 1，就允许 DMA 传送。但是，传送通道 CHCR 的 TE 位、DMAOR 的 NMIF 位和 AE 位必须全部为 0。如果清除 DME 位，就中断全部通道的 DMA 传送。</p> <p>0: 禁止全部通道的 DMA 传送</p> <p>1: 允许全部通道的 DMA 传送</p>

【注】 \* 为了清除标志，只能在读 1 后写 0。

### 10.3.6 总线功能扩展寄存器 (BSCEHR)

BSCEHR 是 16 位可读写寄存器，设定 DMAC 的总线权释放时序等。对于优先 DMAC 传送，BSCEHR 能设定有效的功能。详细内容请参照“9.4.8 总线功能扩展寄存器 (BSCEHR)”。

## 10.4 运行说明

当发生 DMA 传送请求时，DMAC 按照已决定的通道优先级开始传送；当满足传送结束条件时，DMAC 就结束传送。传送请求有自动请求、外部请求和内部外围模块请求 3 种模式。可选择突发模式或者周期挪用模式的总线模式。

### 10.4.1 传送流程

在给 DMA 源地址寄存器 (SAR)、DMA 目标地址寄存器 (DAR)、DMA 传送计数寄存器 (DMATCR)、DMA 通道控制寄存器 (CHCR) 和 DMA 操作寄存器 (DMAOR) 设定目标传送条件后，DMAC 按照以下步骤进行数据传送：

1. 检查是否为传送允许状态 (DE=1、DME=1、TE=0、AE=0、NMIF=0)。
2. 如果在传送允许状态下发生传送请求，就传送 1 个传送单位的数据 (取决于 TS0 和 TS1 位的设定)。在自动请求模式中，如果 DE 位和 DME 位被置 1，就自动开始传送。每进行 1 次传送，DMATCR 的值就递 1。具体的传送流程因地址模式和总线模式而不同。
3. 如果指定次数的传送结束 (DMATCR 的值为 0)，就正常结束传送。此时，如果 CHCR 的 IE 位已经置 1，就向 CPU 请求 DEI 中断。
4. 如果发生 DMAC 引起的地址错误或者 NMI 中断，就中断传送。即使 CR 的 DE 位或者 DMAOR 的 DME 位已被清 0，也中断传送。

**【注】** DMAC 传送中断时的数据传送状态和寄存器的更新状态

- 当发生 DMAC 地址错误时，不进行数据传送，但是更新 SAR、DAR 和 DMATCR。
- 当发生 NMI 中断时，在进行 1 个传送单位的数据传送后中断传送，并且正确更新 SAR、DAR 和 DMATCR。
- 当清除 CHCR 的 DE 位和 DMAOR 的 DME 位时，在进行 1 个传送单位的数据传送后中断传送，并且正确更新 SAR、DAR 和 DMATCR。

上述的流程图如图 10.2 所示。

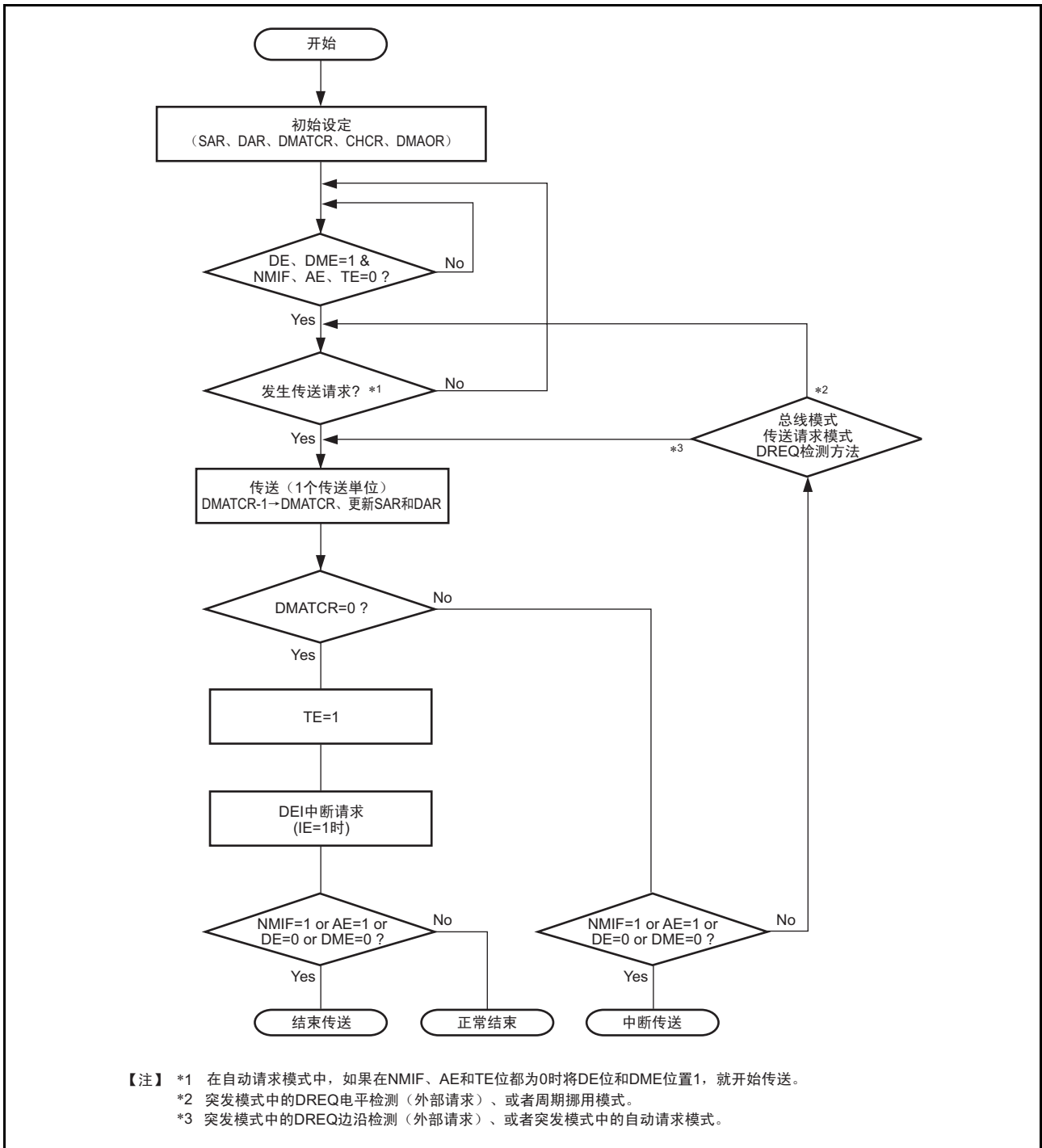


图 10.2 DMA 传送流程图

### 10.4.2 DMA 传送请求

基本的使用方法是让数据的传送源或者传送目标产生 DMA 传送请求，但是也有让既不是传送源又不是传送目标的外部器件或者内部外围模块产生 DMA 传送请求的使用方法。

传送请求有自动请求、外部请求和内部外围模块请求 3 种。根据各 DMA 通道 CHCR 的 RS3 ~ RS0 位选择传送请求。

#### (1) 自动请求模式

如同存储器之间的传送或者不能产生传送请求的内部外围模块和存储器之间的传送，在没有外部传送请求信号时，自动请求模式在 DMAC 内部自动产生传送请求信号。如果各 DMA 通道 CHCR 的 DE 位和 DMAOR 的 DME 位被置 1，就开始传送。但是，CHCR 的 TE 位、DMAOR 的 AE 位和 NMIF 位必须全部为 0。

#### (2) 外部请求模式

外部请求模式是通过 LSI 外部器件的传送请求信号 (DREQ0 ~ DREQ3) 开始传送的模式。根据系统，从表 10.3 所示的模式中选择 1 个使用。如果在允许 DMA 传送 (DE=1、DME=1、TE=0、AE=0、NMIF=0) 时输入 REQ，就开始 DMA 传送。

表 10.3 通过 RS 位选择的外部请求模式

RS3	RS2	RS1	RS0	地址模式	传送源	传送目标
0	0	0	0	双地址模式	任意	任意
		1	0	单地址模式	外部存储器或者被映像为存储器的外部器件	带 DACK 的外部器件
			1		带 DACK 的外部器件	外部存储器或者被映像为存储器的外部器件

如表 10.4 所示，通过 CHCR\_0 ~ CHCR\_3 的 DREQ 电平 (DL) 位和 DREQ 边沿选择 (DS) 位，选择是通过边沿还是通过电平来检测 DREQ。传送请求源不一定需要是数据的传送源或者传送目标。

表 10.4 通过 DL 或者 DS 位选择的外部请求检测

CHCR_0 ~ CHCR_3		外部请求的检测方法
DL	DS	
0	0	低电平检测
	1	下降沿检测
1	0	高电平检测
	1	上升沿检测

【注】 在设定 CHCR\_0 ~ CHCR\_3 前，必须预先通过引脚功能控制器 (PFC) 选择 DREQ 引脚功能。

当接受 DREQ 时，DREQ 引脚就处于不能接受请求的状态。对于已接受的 DREQ，在输出应答 DACK 后，DREQ 引脚重新返回到能接受请求的状态。

在通过电平检测 DREQ 时，根据从输出 DACK 后到检测下一个 DREQ 的时序，有在执行和请求相同次数的传送后中断传送 (超出 0) 的情况以及在执行比请求多 1 次的传送后中断传送 (超出 1) 的情况。通过 CHCR\_0 ~ CHCR\_3 的 DO 位选择超出 0 或者超出 1。

表 10.5 通过 DO 位选择外部请求检测

CHCR_0 ~ CHCR_3	外部请求
DO	
0	超出 0
1	超出 1

## (3) 内部外围模块请求模式

在内部外围模块请求模式中，通过内部外围模块的 DMA 传送请求信号进行传送。DMA 传送请求信号有 5 来自种多功能定时器脉冲单元 2 (MTU2) 的比较匹配中断和输入捕捉中断，10 个来自 2 个串行通信接口 (SCI) 的接收数据满中断 (RXI)、发送数据空中断 (TXI) 和 A/D 转换器的 A/D 转换结束中断 (ADI)。

如果在选择内部外围模块请求模式时处于 DMA 传送允许状态 (DE=1、DME=1、TE=0、AE=0、NMIF=0)，就通过传送请求信号进行传送。

传送请求源不一定需要是数据的传送源或者传送目标。但是，如果将传送请求设定为 SCI 发送数据空的传送请求 (TXI)，传送目标就必须是该 SCI 的传送数据寄存器 (TDR)；同样地，如果将传送请求设定为 SCI 接收数据满的传送请求 (RXI)，传送源就必须是该 SCI 的接收数据寄存器 (RDR)；如果将传送请求设定为 A/D 转换器的 A/D 转换结束的传送请求 (ADI)，传送源就必须是该 A/D 转换器的寄存器。

表 10.6 通过 RS 位选择内部外围模块请求模式

RS3	RS2	RS1	RS0	DMAC 传送 请求源	DMAC 传送 请求信号	传送源	传送目标	总线模式
0	1	1	0	MTU2	TGIA_0	任意 *	任意 *	突发 / 周期挪用模式
			1	MTU2	TGIA_1	任意 *	任意 *	突发 / 周期挪用模式
1	0	0	0	MTU2	TGIA_2	任意 *	任意 *	突发 / 周期挪用模式
			1	MTU2	TGIA_3	任意 *	任意 *	突发 / 周期挪用模式
		1	0	MTU2	TGIA_4	任意 *	任意 *	突发 / 周期挪用模式
			1	A/D_1	ADI1	ADDR4 ~ ADDR7	任意 *	周期挪用模式
	1	0	0	SCI_0 发送部	TXI_0	任意 *	SCTDR_0	周期挪用模式
			1	SCI_0 接收部	RXI_0	SCRDR_0	任意 *	周期挪用模式
		1	0	SCI_1 发送部	TXI_1	任意 *	SCTDR_1	周期挪用模式
			1	SCI_1 接收部	RXI_1	SCRDR_1	任意 *	周期挪用模式

【注】 MTU2: 多功能定时器脉冲单元 2

SCI\_0、SCI\_1: 串行通信接口的通道 0、1

ADDR4 ~ ADDR7: A/D 转换器通道 1 的 A/D 数据寄存器

SCTDR\_0、SCTDR\_1: SCI\_0、SCI\_1 的发送数据寄存器

SCRDR\_0、SCRDR\_1: SCI\_0、SCI\_1 的接收数据寄存器

\* 外部存储器、被映像为存储器的外部器件、内部存储器、内部外围模块 (DMAC、DTC、BSC、UBC 除外)

为了从内部外围模块输出传送请求，必须在设定各模块对应的中断允许位后输出中断信号。

如果将内部外围模块的中断请求信号用作 DMA 传送请求信号，就不向 CPU 请求中断。详细内容请参照“6.8 通过中断请求信号的数据传送”。

当进行对应的 DMA 传送时，就在周期挪用模式的 1 次传送时或者在发生突发模式的最后传送时，自动取消表 10.7 中的传送请求信号。

### 10.4.3 通道优先级

如果同时对多个通道产生传送请求，DMAC 按照规定的优先级进行传送。通道优先级可通过 DMA 操作寄存器 (DMAOR) 的 PR1 位和 PR0 位从固定和周期 2 种模式中选择。

#### (1) 固定模式

在固定模式中，通道优先级不变。

固定模式的通道优先级有以下 2 种：

- CH0 > CH1 > CH2 > CH3
- CH0 > CH2 > CH3 > CH1

通过 DMA 操作寄存器 (DMAOR) 的 PR1 位和 PR0 位选择这些模式。

#### (2) 周期模式

在周期模式中，每当一个通道的 1 个传送单位（字节、字、长字或者 16 字节为单位）的传送结束时，就更改优先级，使该通道优先级变为最低，如图 10.3 所示。另外，刚复位的周期模式优先级为 CH0 > CH1 > CH2 > CH3。

当指定周期模式时，在多个通道的总线模式中不能同时存在周期挪用模式和突发模式。

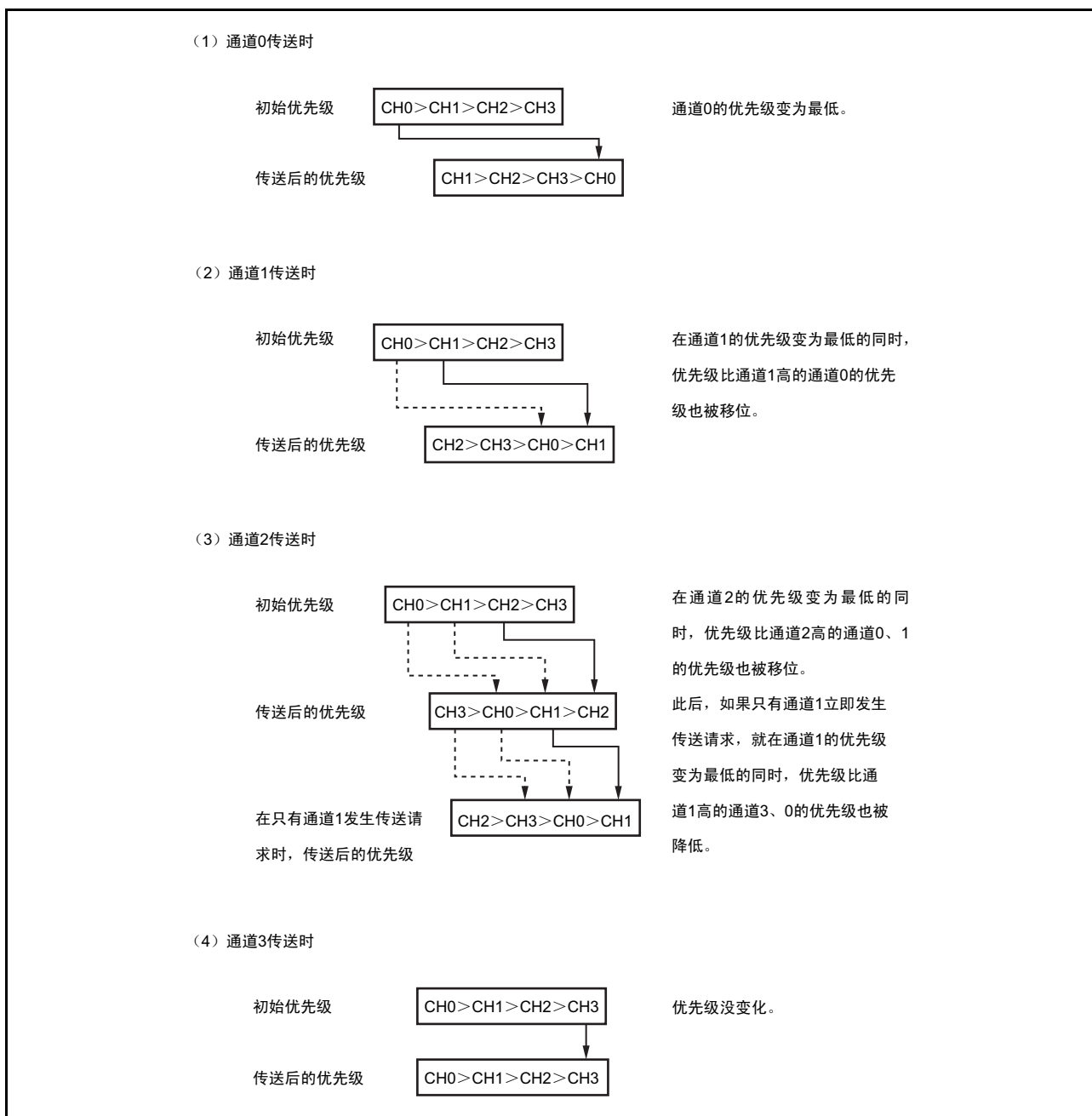


图 10.3 周期模式



当通道 0 和通道 3 同时发生传送请求并且在通道 0 传送时发生通道 1 的传送请求时，通道优先级的变化如图 10.4 所示。此时，DMAC 的运行如下：

1. 通道 0 和通道 3 同时发生传送请求。
2. 因为通道 0 的优先级高于通道 3，所以开始通道 0 的传送（通道 3 等待传送）。
3. 在通道 0 传送时通道 1 发生传送请求（通道 1 和通道 3 都等待传送）。
4. 当通道 0 的传送结束时，通道 0 的优先级变为最低。
5. 因为此时通道 1 的优先级高于通道 3，所以开始通道 1 的传送（通道 3 等待传送）。
6. 当通道 1 的传送结束时，通道 1 的优先级变为最低。
7. 开始通道 3 的传送。
8. 当通道 3 的传送结束时，通道 3 的优先级变为最低，同时通道 2 的优先级也变低。

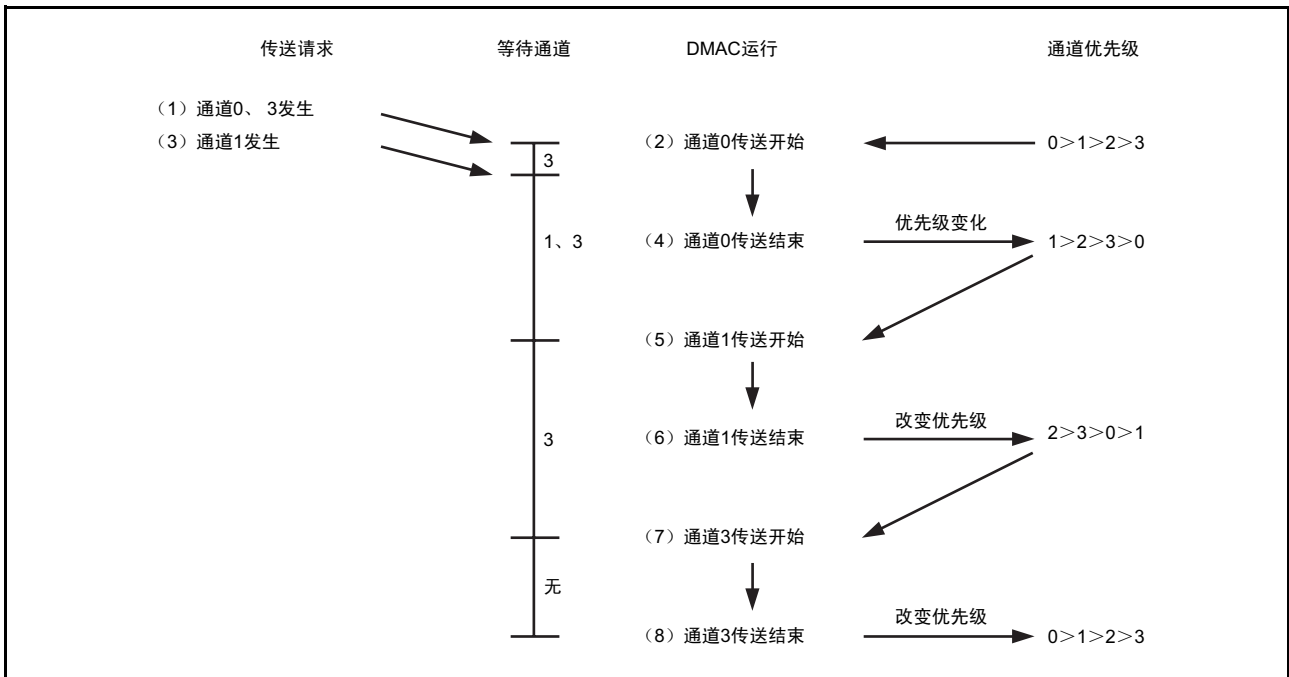


图 10.4 周期模式中的通道优先级

(3) 发生多个 DMAC 启动请求时的启动优先级

当发生多个 DMAC 启动请求时，就根据启动优先级进行传送。但是，如果在 DMAC 不为总线主控时发生多个 DMAC 启动请求，就从最初发生的启动请求开始传送。另外，在 CPU 存取外部空间时，从最初和第 2 个发生的启动请求开始传送。DMAC 的启动优先级的运行例子如图 10.5 所示。

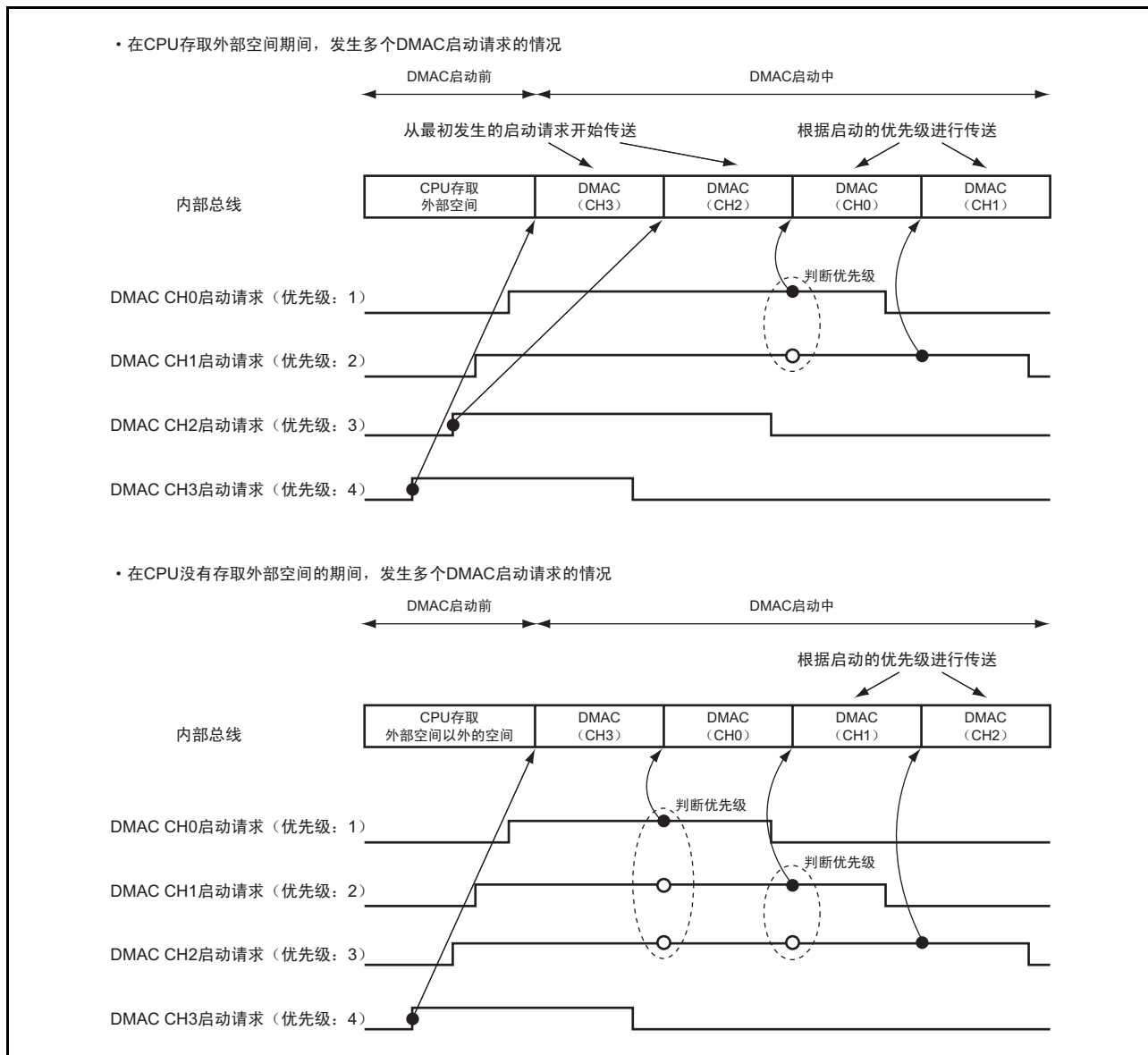


图 10.5 DMAC 的启动优先级的运行例 (优先级固定模式为 CH0 > CH1 > CH2 > CH3 时)

#### 10.4.4 DMA 的传送种类

根据存取传送源和传送目标的总线周期次数，将 DMA 传送分为单地址模式传送和双地址模式传送。具体的传送时序因总线模式而不同，总线模式有周期挪用模式和突发模式。DMAC 能支持的传送如表 10.7 所示。

表 10.7 能支持的 DMA 传送

传送源	传送目标				
	带 DACK 的外部器件	外部存储器	被映像为存储器的外部器件	内部外围模块	内部存储器
带 DACK 的外部器件	不能	双、单	双、单	不能	不能
外部存储器	双、单	双	双	双	双
被映像为存储器的外部器件	双、单	双	双	双	双
内部外围模块	不能	双	双	双	双
内部存储器	不能	双	双	双	双

- 【注】
1. 双：双地址模式
  2. 单：单地址模式
  3. 内部外围模块只能对允许长字存取的寄存器进行 16 字节传送。

## (1) 地址模式

## (a) 双地址模式

双地址模式是用于通过地址存取传送源和传送目标的模式。传送源和传送目标可以是外部模块也可以是内部器件。在此模式中，DMAC 在读周期中存取传送源，在写周期中存取传送目标，通过 2 个总线周期进行传送。此时，传送数据暂时被保存在 DMAC。例如，对于图 10.6 所示的外部存储器之间的传送，在读周期中从一个外部存储器将数据读到 DMAC，然后在写周期中将该数据写到另一个外部存储器。

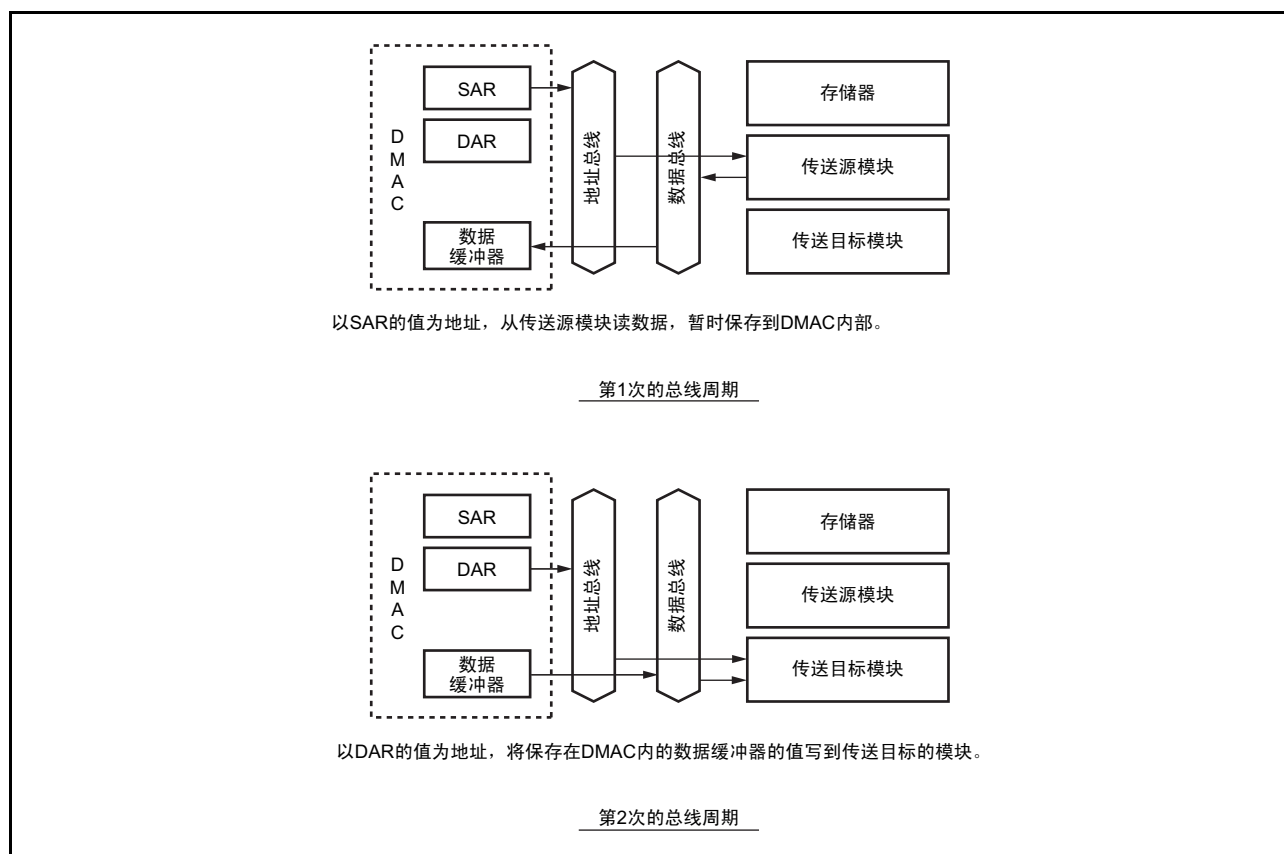


图 10.6 双地址模式的数据流程

传送请求可以是自动请求、外部请求或者内部外围模块请求。在双地址模式中，能在读周期或者写周期中输出 DACK，DACK 是在读周期中还是在写周期中输出，能通过通道控制寄存器 (CHCR) 的 AM 位设定。双地址模式中的 DMA 传送时序例子如图 10.7 所示。

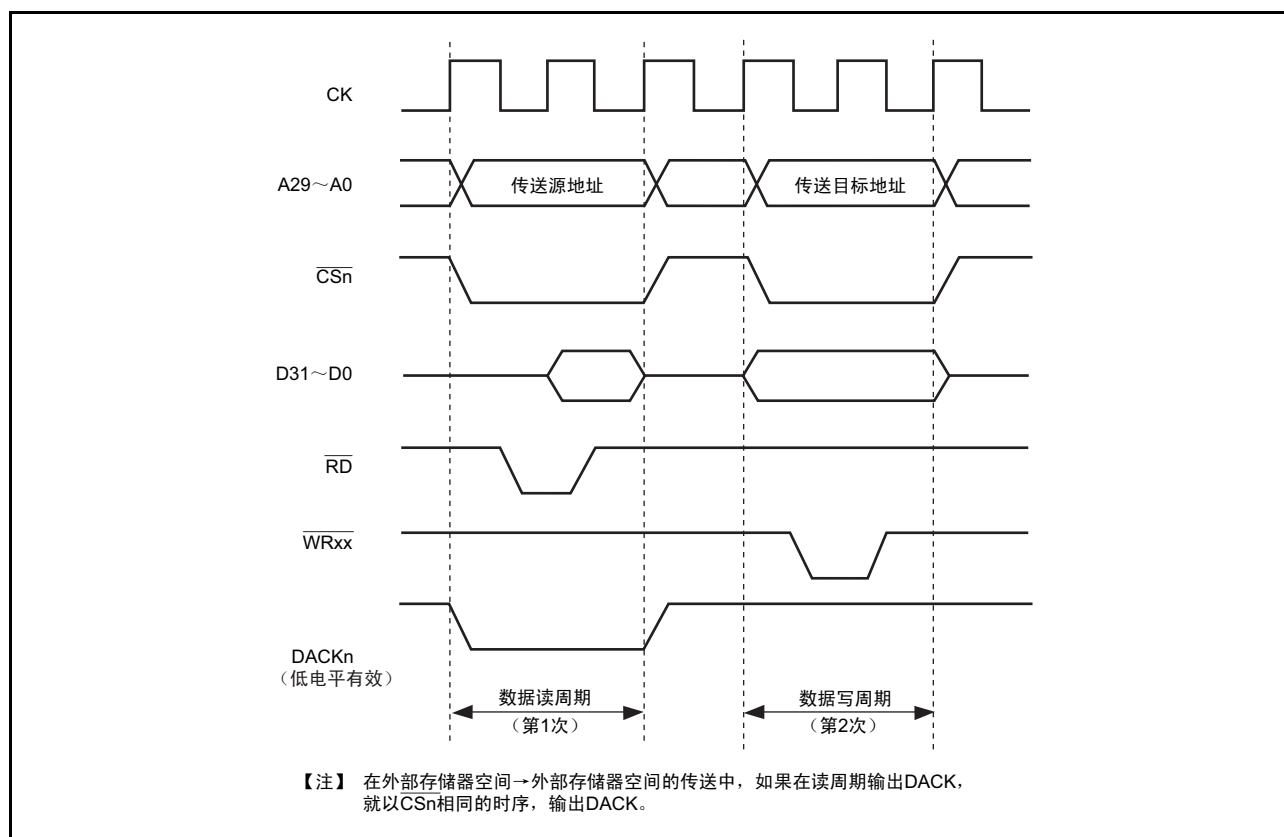


图 10.7 双地址模式中的 DMA 传送时序例  
(传送源：通常存储器，传送目标：通常存储器)

(b) 单地址模式

单地址模式是用于在传送源和传送目标都是外部器件时通过 DACK 信号存取 (选择) 其中一个模块而通过地址存取另一个模块的模式。在此模式中, DMAC 在将传送请求的接收信号 DACK 输出到一个外部器件进行存取的同时, 将地址输出到另一个器件, 通过 1 个总线周期进行 DMA 传送。例如, 对于图 10.8 所示的外部存储器和带 DACK 的外部器件之间的传送, 在和外部器件将数据输出到数据总线的相同总线周期中, 将该数据写到外部存储器。

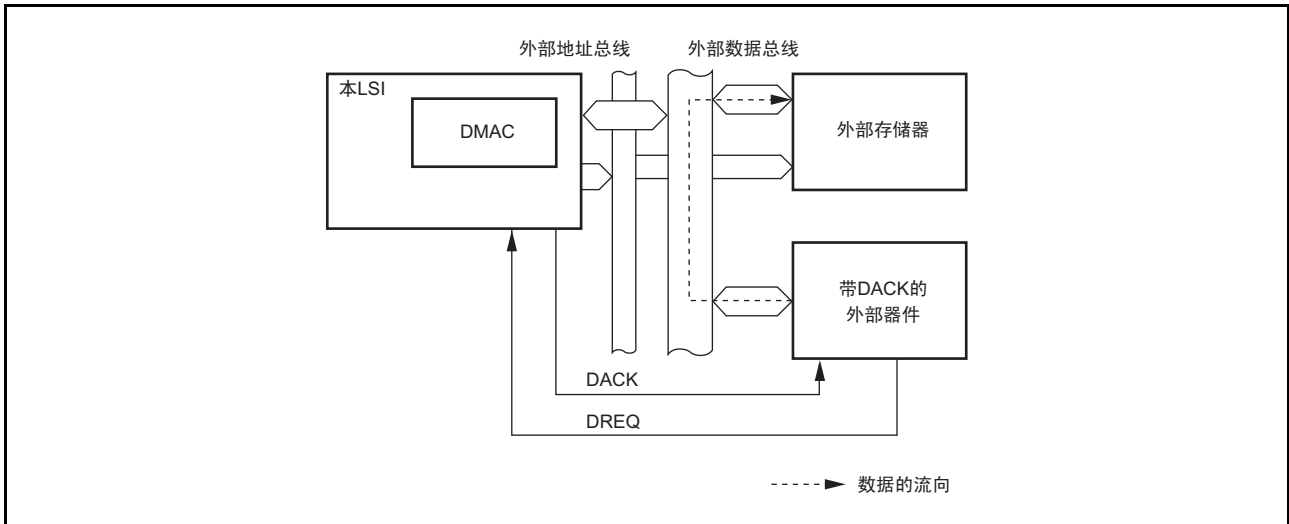


图 10.8 单地址模式的数据流程

能在单地址模式中进行带 DACK 的外部器件和被映像为存储器的外部器件之间的传送、以及带 DACK 的外部器件和外部存储器之间的传送。无论那种情况, 传送请求都只是外部请求 (DREQ)。

单地址模式中的 DMA 传送时序例子如图 10.9 所示。

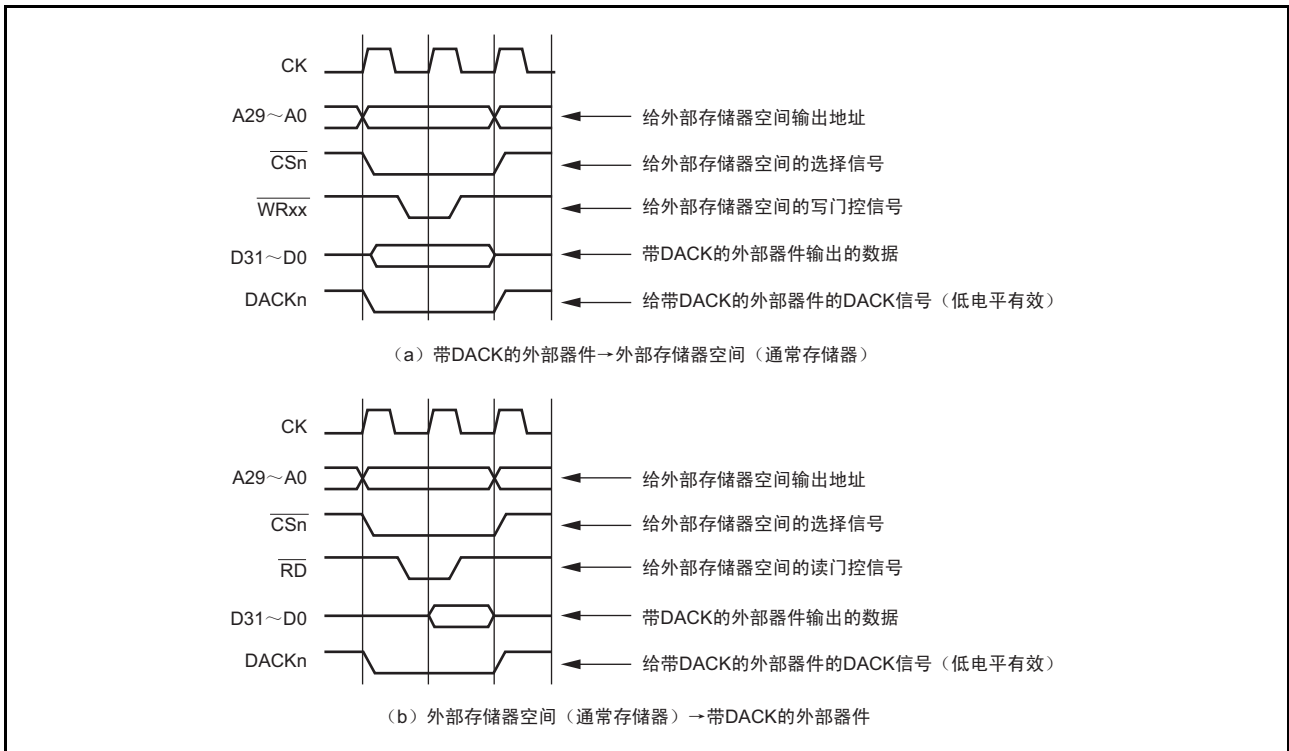


图 10.9 单地址模式中的 DMA 传送时序例

(2) 总线模式

总线模式有周期挪用模式和突发模式，通过通道控制寄存器 (CHCR) 的 TB 位进行选择。

(a) 周期挪用模式

● 通常模式

在周期挪用的通常模式中，每当 1 个传送单位 (字节、字、长字或者 16 字节为单位) 的传送结束时，DMAC 就将总线权交给其他总线主控。如果此后发生传送请求，就从其他总线主控取回总线权，重新进行 1 个传送单位的传送，当该传送结束时，又将总线权交给其他总线主控。重复此操作直到满足传送结束条件为止。

周期挪用的通常模式与传送请求源、传送源和传送目标无关，可在全部传送区间内使用。

周期挪用通常模式中的 DMA 传送时序例子如图 10.10 所示，图中的传送条件如下：

- 双地址模式
- DREQ 低电平检测

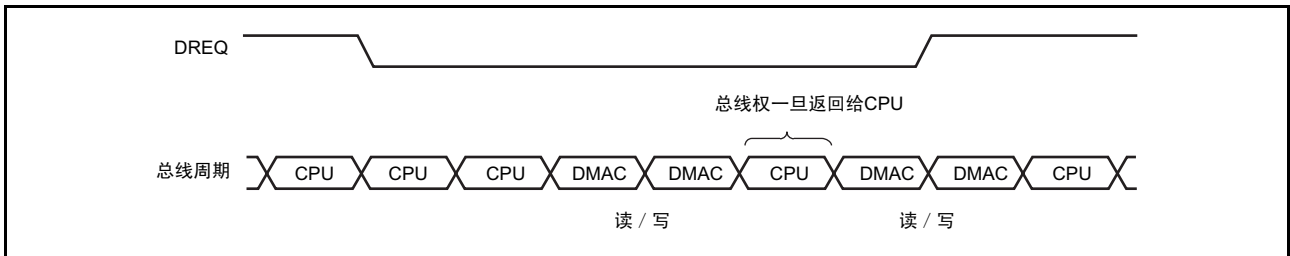


图 10.10 周期挪用通常模式中的 DMA 传送例 (双地址、DREQ 低电平检测)

● 间歇模式 16 和间歇模式 64

在周期挪用的间歇模式中，每当 1 个传送单位 (字节、字、长字或者 16 字节为单位) 的传送结束时，DMAC 就将总线权交给其他总线主控。如果此后发生传送请求，就通过 Bφ 计数等待 16 个时钟或者 64 个时钟，然后从其他总线主控取回总线权，重新进行 1 个传送单位的传送，当该传送结束时，又将总线权交给其他总线主控。重复此操作直到满足结束条件为止。因此，和周期挪用的通常模式相比，能降低 DMA 传送的总线占有率。

间歇模式与传送请求源、传送源和传送目标无关，可在全部传送区间内使用，但是全部通道的总线模式必须都是周期挪用模式。

周期挪用间歇模式中的 DMA 传送时序例子如图 10.11 所示，图中的传送条件如下：

- 双地址模式
- DREQ 低电平检测

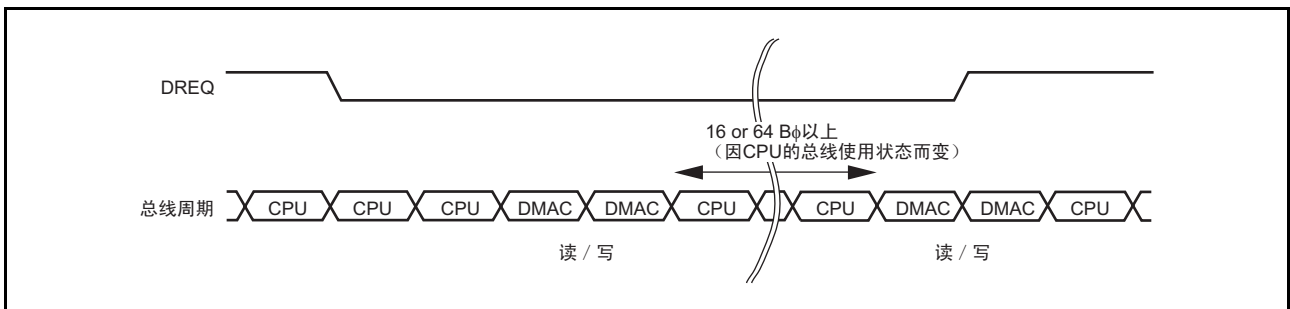


图 10.11 周期挪用间歇模式中的 DMA 传送例 (双地址、DREQ 低电平检测)

## (b) 突发模式

在突发模式中，DMAC 一旦取得总线权，就在满足传送结束条件前不释放总线权而连续进行传送。但是，在外部请求模式中通过电平检测 DREQ 的情况下，如果 DREQ 变为无效电平，即使尚未满足传送结束条件，也在结束已接受请求的 DMA 传送请求后将总线权交给其他总线主控。

突发模式中的 DMA 传送时序如图 10.12 所示。

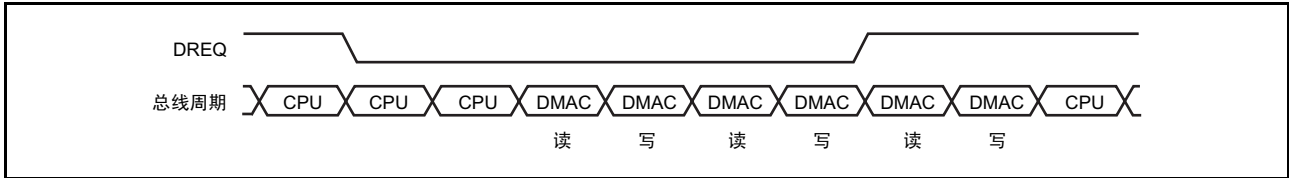


图 10.12 突发模式中的 DMA 传送例（双地址、DREQ 低电平检测）



## (3) DMA 传送区间和请求模式、总线模式的关系

DMA 传送区间和请求模式、总线模式等相关事项如表 10.8 所示。

表 10.8 DMA 传送区间和请求模式、总线模式等相关事项一览表

地址模式	传送区间	请求模式	总线模式	传送长度 (位)	可使用的 通道
双地址模式	带 DACK 的外部器件和外部存储器	外部	B/C	8/16/32/128	0 ~ 3
	带 DACK 的外部器件和被映像为存储器的外部器件	外部	B/C	8/16/32/128	0 ~ 3
	外部存储器和外部存储器	均可 *1	B/C	8/16/32/128	0 ~ 3
	外部存储器和被映像为存储器的外部器件	均可 *1	B/C	8/16/32/128	0 ~ 3
	被映像为存储器的外部器件和被映像为存储器的外部器件	均可 *1	B/C	8/16/32/128	0 ~ 3
	外部存储器和内部外围模块	均可 *2	B/C*3	8/16/32/128*4	0 ~ 3
	被映像为存储器的外部器件和内部外围模块	均可 *2	B/C*3	8/16/32/128*4	0 ~ 3
	内部外围模块和内部外围模块	均可 *2	B/C*3	8/16/32/128*4	0 ~ 3
	内部存储器和内部存储器	均可 *1	B/C	8/16/32/128	0 ~ 3
	内部存储器和被映像为存储器的外部器件	均可 *1	B/C	8/16/32/128	0 ~ 3
	内部存储器和内部外围模块	均可 *2	B/C*3	8/16/32/128*4	0 ~ 3
	内部存储器和外部存储器	均可 *1	B/C	8/16/32/128	0 ~ 3
单地址模式	带 DACK 的外部器件和外部存储器	外部	B/C	8/16/32	0 ~ 3
	带 DACK 的外部器件和被映像为存储器的外部器件	外部	B/C	8/16/32	0 ~ 3

## 【符号说明】

B: 突发模式

C: 周期挪用模式

【注】 \*1 外部请求、自动请求和内部外围模块请求均可。但是，在内部外围模块请求的情况下，不能将 SCI 和 A/D 转换器指定为传送请求源。

\*2 外部请求、自动请求和内部外围模块请求均可。但是，在传送请求源为 SCI 或者 A/D 转换器的情况下，传送源或者传送目标必须是 SCI 或者 A/D 转换器。

\*3 在传送请求源为 SCI 或者 A/D 转换器的情况下，只限于周期挪用模式。

\*4 作为传送源或者传送目标的内部外围模块寄存器所允许的存取长度。

(4) 总线模式和通道优先级

在优先级固定模式 (CH0 > CH1) 中, 如果优先级更高的通道 0 发生传送请求, 即使通道 1 正以突发模式进行传送, 也立即开始通道 0 的传送。

此时, 如果通道 0 也是突发模式, 在优先级高的通道 0 的传送全部结束后, 继续通道 1 的传送。

如果通道 0 是周期挪用模式, 首先在优先级高的通道 0 进行 1 个传送单位的传送后, 不释放总线权而连续进行通道 1 的传送。然后按照通道 0→通道 1→通道 0→通道 1 进行交替传送, 即总线状态是在周期挪用模式传送结束后的 CPU 周期被替换为突发模式传送的状态 (以下, 称为突发模式的优先执行)。

此例如图 10.13 所示。如果发生竞争的突发模式有多个通道, 就优先执行优先级最高的通道。

如果在多个通道进行 DMA 传送, 在发生竞争的突发传送全部结束前, 总线权不释放给总线主控。

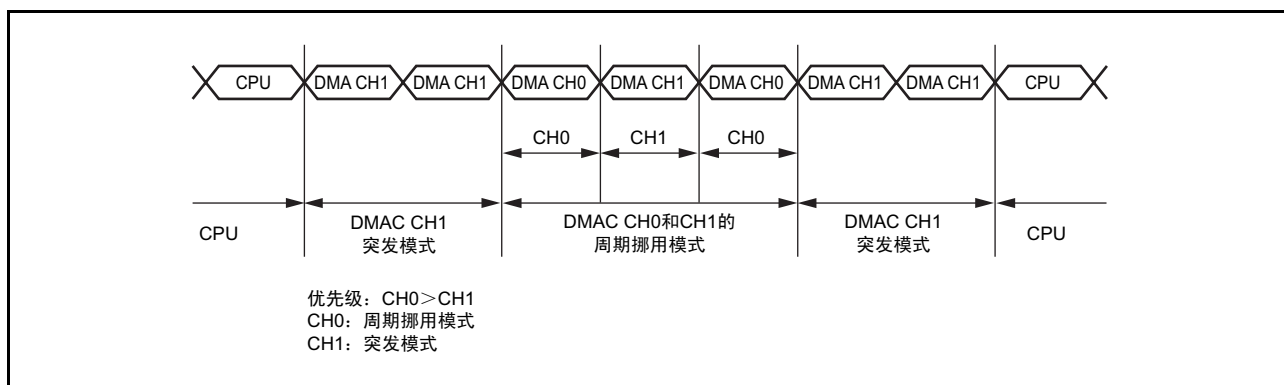


图 10.13 多个通道运行时的总线状态

在周期模式中, 按照图 10.13 所示的规则, 优先级发生变化。但是, 在总线模式中, 不能同时存在周期挪用模式的通道和突发模式的通道。

## 10.4.5 总线周期的状态数和 DREQ 引脚的采样时序

## (1) 总线周期的状态数

DMAC 为总线主控时和 CPU 为总线主控时一样，由总线状态控制器 (BSC) 控制总线周期的状态数，详细内容请参照“第 9 章 总线状态控制器 (BSC)”。

## (2) DREQ 引脚的采样时序

各总线模式中的 DREQ 输入的采样时序如图 10.14 ~ 图 10.17 所示。

通过 DREQ 判断 DMAC 的启动时，需要 3 个 Bcyc 周期 (Bcyc 表示外部时钟 ( $B\phi=CK$ ) 的周期)。第 1 次接受 DREQ 时的 DACK 输出因内部总线的状态、CHCR 的 AM 位的设定以及传送源 / 传送目标区的 BSC 设定而不同，但是最快也需要 6 个 Bcyc 周期。

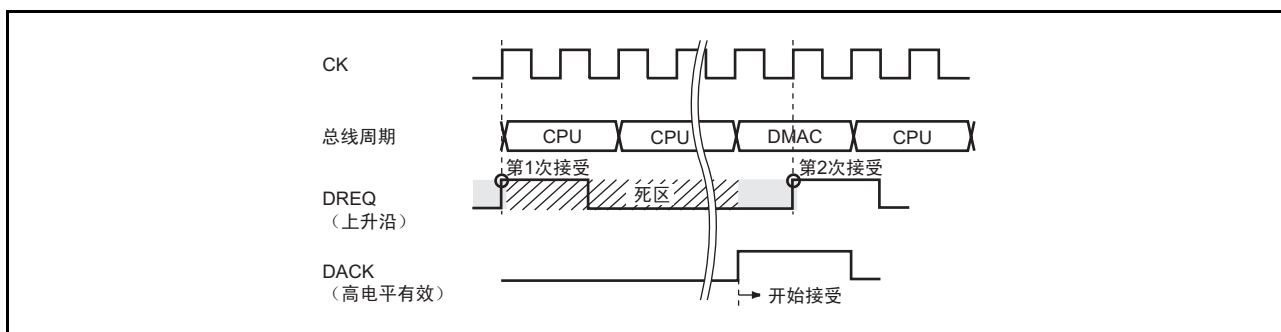


图 10.14 周期挪用模式、边沿检测时的 DREQ 输入检测的时序例子

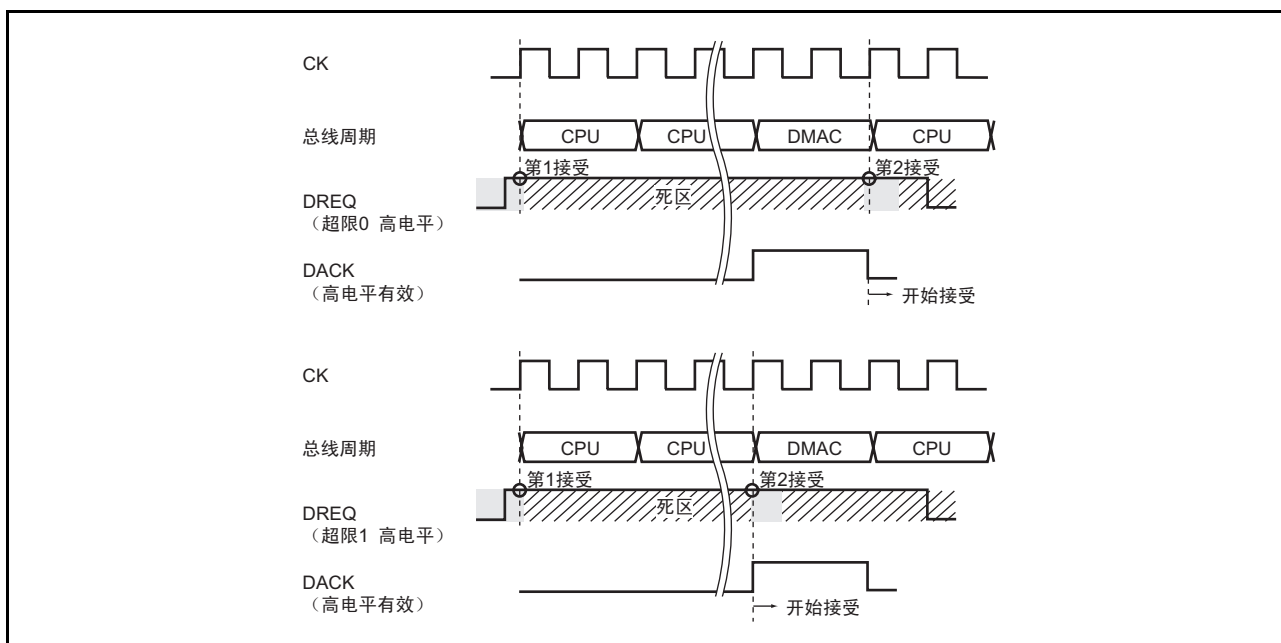


图 10.15 周期挪用模式、电平检测时的 DREQ 输入检测的时序例子

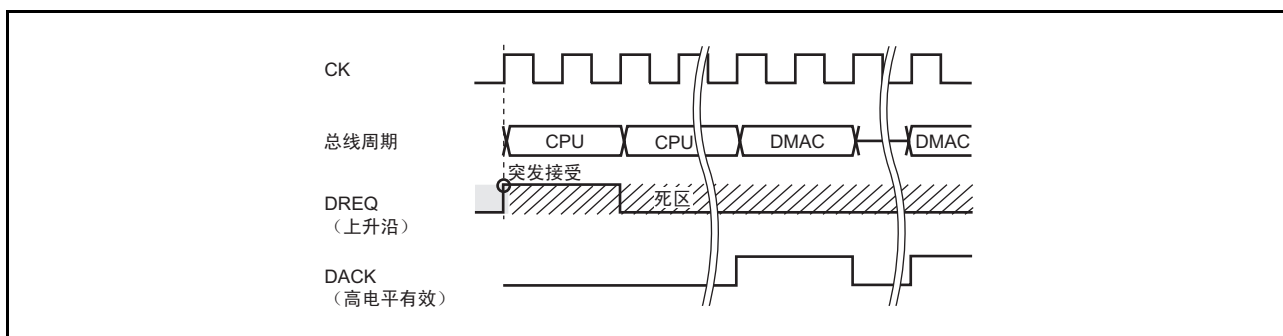


图 10.16 突发模式、边沿检测时的 DREQ 输入检测的时序例子

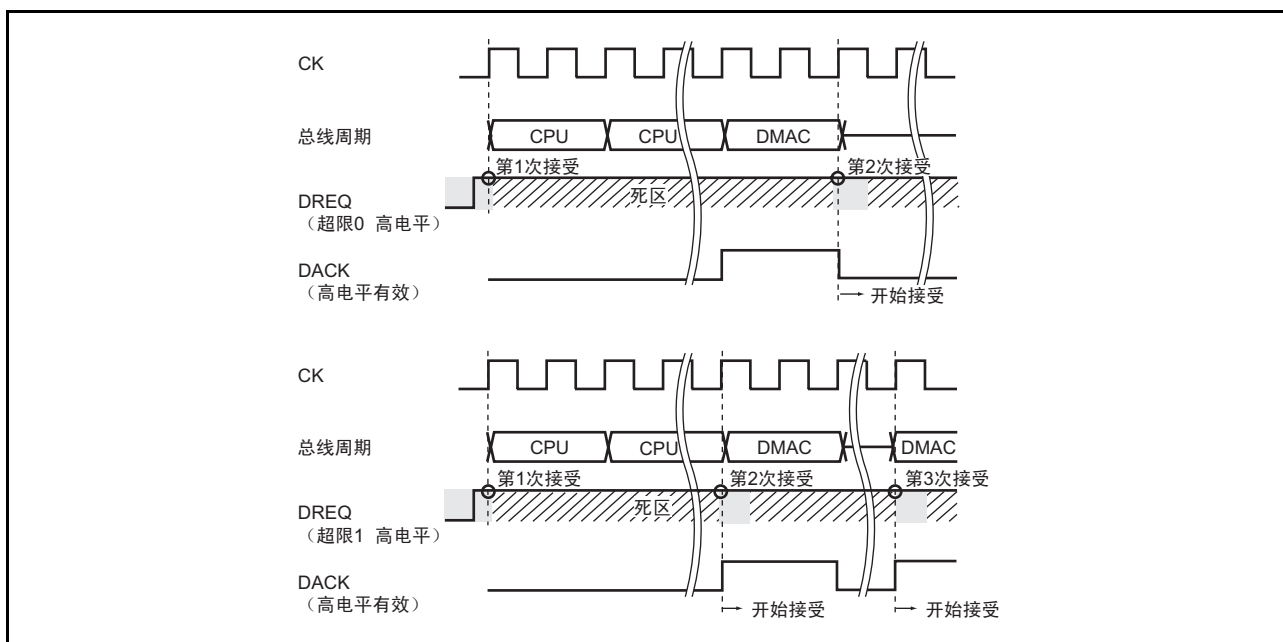


图 10.17 突发模式、电平检测时的 DREQ 输入检测的时序例子

TEND 的输出时序如图 10.18 所示。

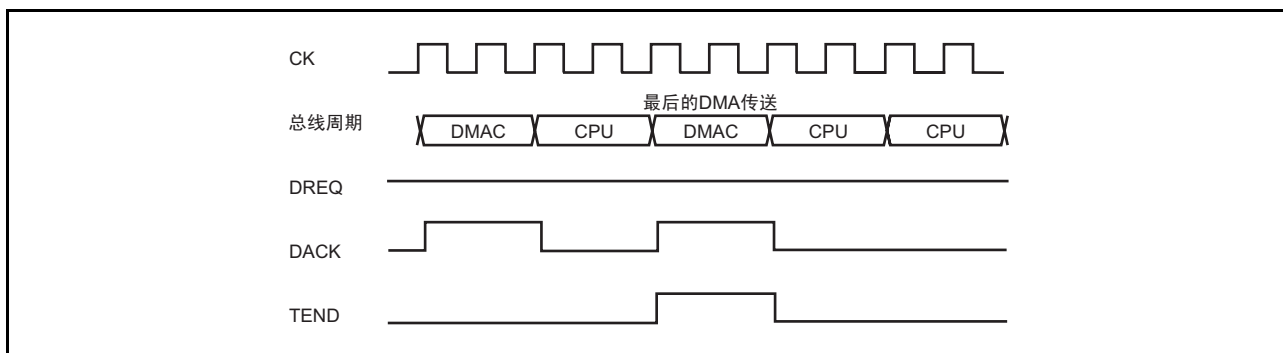


图 10.18 DMA 传送结束时序 (周期挪用、电平检测)

必须注意：在对外部器件进行 16 字节传送、或者对 8 位、16 位外部器件进行长字存取、或者对 8 位外部器件进行字存取时，DMA 传送单位被分割为多个总线周期。在总线周期之间将  $\overline{CS}$  设定为无效时，为了定位数据，和  $\overline{CS}$  一样 DACK 输出和 TEND 输出被分割。此例如图 10.19 所示。

对于被分割的 DACK，不能正确地检测到 DREQ 的采样，而且有可能发生最多 1 次的超出。无论是在使用不分割 DACK 的设定时还是在分割 DACK 时，都必须指定不大于外部器件总线宽度的传送长度。

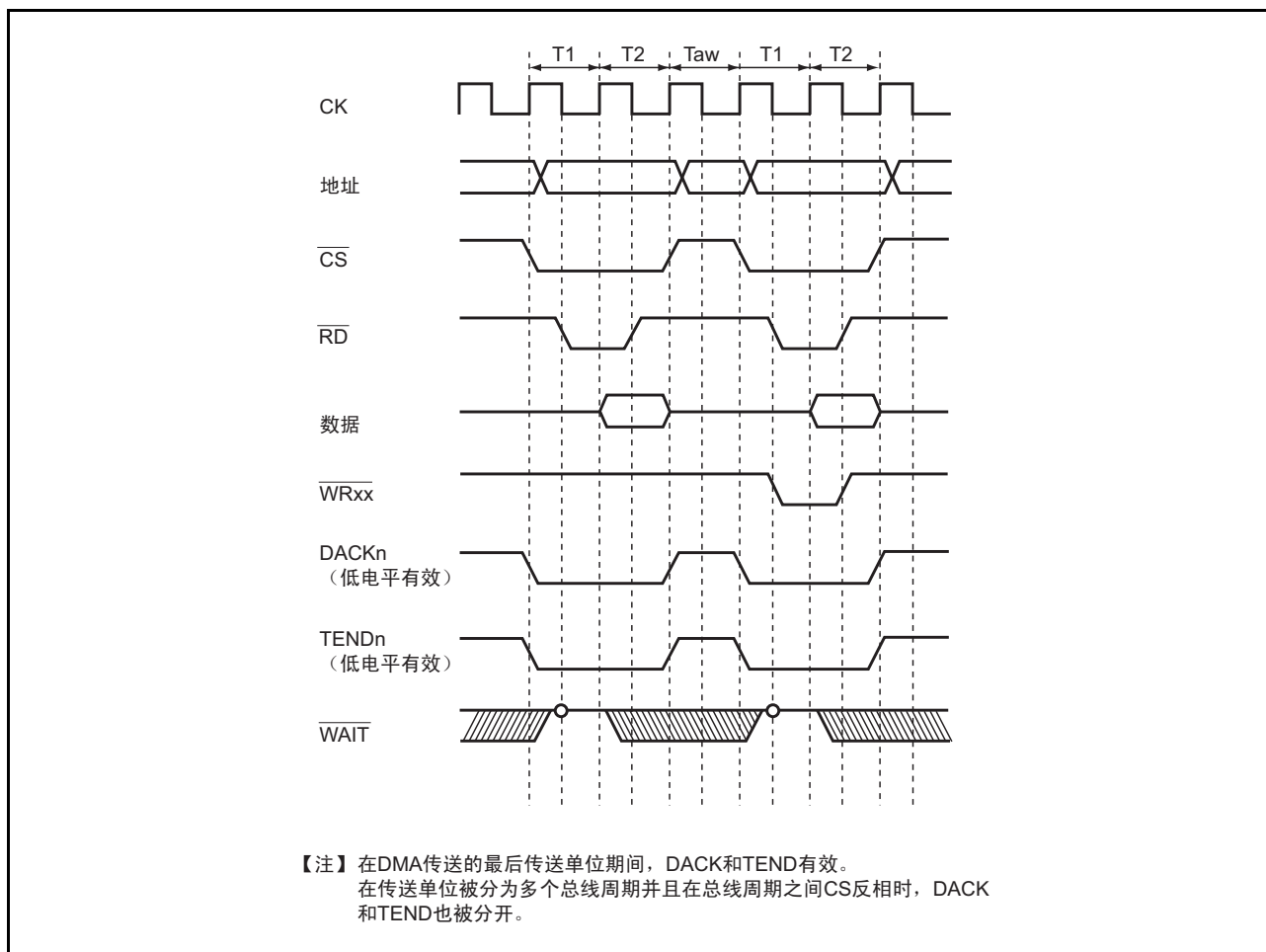


图 10.19 BSC 通常存储器存取  
(无等待、空闲周期 = 1、16 位器件的长字存取)

### 10.4.6 运行时序

DMAC 的运行时序如图 10.20、10.21 所示。

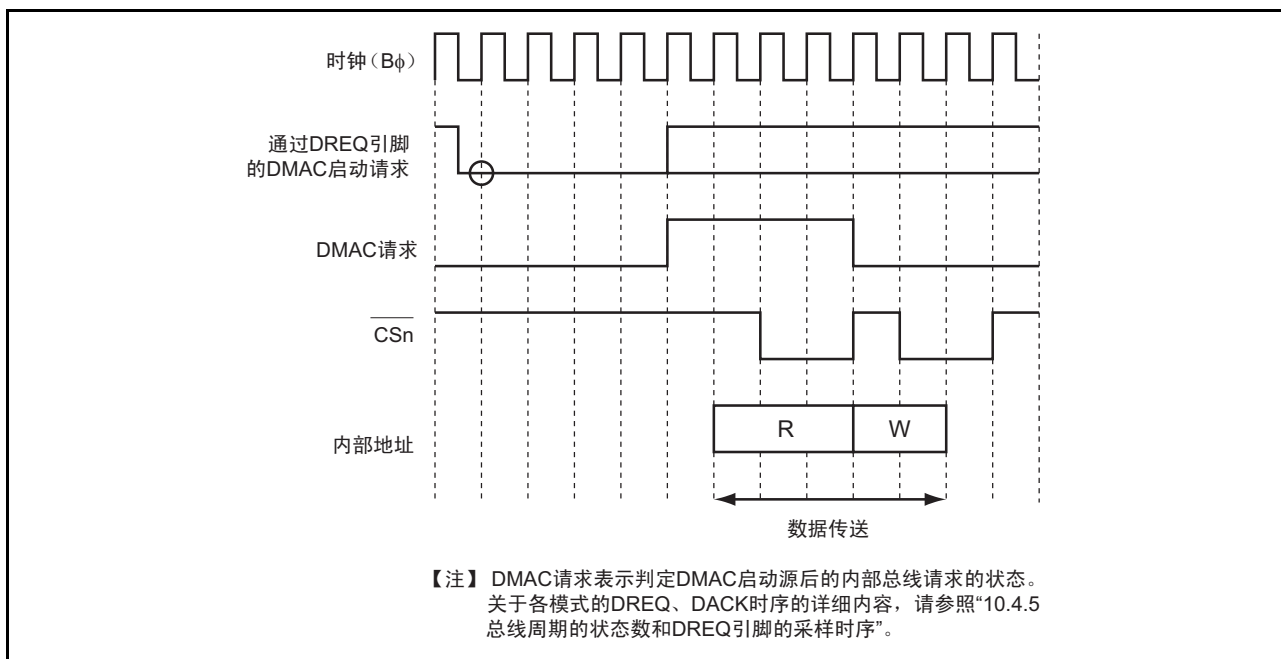


图 10.20 DMAC 的运行时序例子【从 DREQ 启动】  
(在周期挪用传送、双地址模式、低电平检测、 $I\phi:B\phi:P\phi=1:1/2:1/2$ 、  
从外部存储器到外部存储器的数据传送、空闲 / 等待 =0 时)

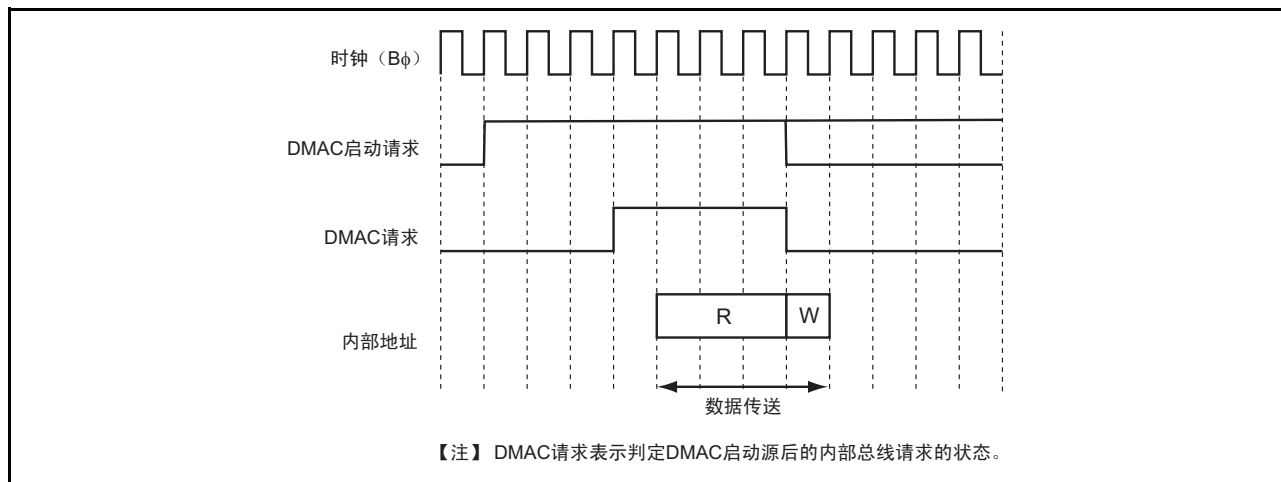


图 10.21 DMAC 的运行时序例子【内部外围模块】  
(在周期挪用传送、双地址模式、低电平检测、 $I\phi:B\phi:P\phi=1:1/2:1/2$ 、  
从内部外围模块到内部 RAM 的数据传送时)

## 10.5 使用注意事项

在使用 DMAC 时，必须注意以下事项。

### 10.5.1 DACK 引脚输出的注意事项

如果在多个通道中同时设定突发模式和周期挪用模式，就有可能在突发传送的最后不必要地将 DACK 置为有效。

在满足以下条件时会发生这种现象：

1. 当突发模式和周期挪用模式两种模式的多个通道同时进行 DMA 传送时
2. 当将突发模式中使用的通道设定为双地址模式并且在写周期中输出 DACK 时
3. 在突发传送结束后，虽然已接受周期挪用的传送请求，但是 DMAC 不能连续取得总线权时

能通过以下 3 种方法之一避免以上现象：

1. 对策 1：必须在确认突发传送结束后（TE 位 =1），执行其他周期挪用模式的 DMA 传送。
2. 对策 2：使用突发模式的通道不能在写周期中设定 DACK 的输出。
3. 对策 3：在多个通道同时进行 DMA 传送时，必须将全部通道都设定为突发模式或者周期挪用模式。

### 10.5.2 外围模块的 DMA 传送

在通过外围模块进行 DMA 传送时，不能将总线时钟（B $\phi$ ）：外围时钟（P $\phi$ ）、总线时钟（B $\phi$ ）：MTU2 时钟（MP $\phi$ ）和总线时钟（B $\phi$ ）：MTU2S 时钟（MI $\phi$ ）的时钟比设定为 1:1/3 或者 1:1/4。

### 10.5.3 模块待机模式的设定

能通过待机控制寄存器禁止或者允许 DMAC 的运行。初始值为 DMAC 运行禁止状态。通过解除模块待机模式，允许寄存器的存取。

但是，不能在 DMAC 运行中转移到软件待机模式或者模块待机模式。如果要转移到软件待机模式和模块待机模式，就必须在清除全部通道的 DE 位后进行转移，详细内容请参照“第 26 章 低功耗模式”。

### 10.5.4 通过 DMAC 存取 DMAC/DTC 寄存器

不能使用 DMAC 存取 DMAC/DTC 寄存器，也不能使用 DTC 存取 DMAC 寄存器。

### 10.5.5 SCI 作为 DMAC 启动源的注意事项

在通过 SCI 的 TXI 中断启动 DMAC 时，不能将 SCI 的 TEND 标志用作传送结束标志。

### 10.5.6 CHCR 的设定

要更改 CHCR 的设定时，必须在清除该通道的 DE 位后进行更改。

### 10.5.7 启动多个通道时的注意事项

不能给多个通道设定相同的内部请求。

### 10.5.8 输入传送请求时的注意事项

必须在结束 DMAC 的设定后输入传送请求。

### 10.5.9 NMI 中断和 DMAC 启动的竞争

因为在 NMI 中断和 DMAC 启动发生竞争时优先 NMI 中断，所以 NMIF 位被置 1，不启动 DMAC。

另外，通过 NMI 判断 DMAC 的停止时需要 1 个 Bcyc+3 个 Pcyc 的周期，通过 DREQ 判断 DMAC 的启动时需要 3 个 Bcyc 周期，通过外围模块判断 DMAC 的启动时需要 1 个 Pcyc 周期（Bcyc 表示外部总线时钟的周期，Pcyc 表示外围时钟的周期）。

### 10.5.10 DMAC 存取内部 RAM 的周期数

DMAC 存取内部 RAM 的周期数取决于读 / 写、I $\phi$ （内部时钟）和 B $\phi$ （外部总线时钟）的时钟比，周期数如表 10.9 所示。

表 10.9 DMAC 存取内部 RAM 的周期数

I $\phi$ :B $\phi$ 的设定	读	写
1:1	3×Bcyc	3×Bcyc
1:1/2	2×Bcyc	1×Bcyc
1:1/3	2×Bcyc	1×Bcyc
不大于 1:1/4	1×Bcyc	1×Bcyc

- 【注】
1. Bcyc 表示外部总线时钟的周期。
  2. 如“9.5.16 从 CPU 向内部外围 I/O 寄存器的存取”和“9.5.17 从 CPU 向外部存储器的存取”所示，从 CPU 的存取周期数中删除 L 总线存取时所必需的 I $\phi$  量就是向内部外围 I/O 以及外部器件的存取周期数。

### 10.5.11 以 MTU2 为启动源的突发模式中的 DMAC 传送的注意事项

在以 MTU2 为启动源的突发模式中进行 DMAC 传送时，必须将总线功能扩展寄存器（BSCEHR）的 DMMTU4 ~ 0 位的对应位置位，详细内容请参照“9.4.8 总线功能扩展寄存器（BSCEHR）”。

### 10.5.12 总线功能扩展寄存器（BSCEHR）

对于优先 DMAC 传送，总线功能扩展寄存器（BSCEHR）能设定有效的功能，详细内容请参照“9.4.8 总线功能扩展寄存器（BSCEHR）”。



## 第 11 章 多功能定时器脉冲单元 2 (MTU2)

本 LSI 内置由 6 个通道的 16 位定时器构成的多功能定时器脉冲单元 2 (MTU2)。

### 11.1 特点

- 能进行最多 16 个脉冲输入 / 输出和 3 个脉冲输入
- 各通道可选择 8 种计数器输入时钟 (通道 5 可选择 4 种)
- 通道 0 ~ 4 能设定以下的运行: 通过比较匹配进行的波形输出、输入捕捉功能、计数器的清除运行、多个定时器的计数器 (TCNT) 的同时写、通过比较匹配 / 输入捕捉进行的同时清除、通过计数器的同步运行进行的各寄存器的同步输入 / 输出、通过和同步运行组合进行的最多 12 相 PWM 输出
- 通道 0、3、4 能设定缓冲运行
- 通道 1、2 能各自独立设定相位计数模式
- 级联运行
- 进行内部 16 位总线的高速存取
- 28 种中断源
- 能进行寄存器数据的自动传送
- 能生成 A/D 转换器的转换开始触发
- 能设定模块待机模式
- 能通过 CH3、4 的联动运行设定互补 PWM 或者复位 PWM 的正负 3 相共 6 相的波形输出
- 能通过 CH0、3、4 联动运行设定使用互补 PWM 或者复位 PWM 的 AC 同步马达 (无刷 DC 马达) 驱动模式, 并可选择 2 种 (斩波和电平) 波形输出
- CH5 有用于空载时间补偿的计数器功能
- 在互补 PWM 模式中, 能跳过在计数器波峰 / 波谷的中断和 A/D 转换器的转换开始触发

表 11.1 MTU2 的功能一览表

项 目	通道 0	通道 1	通道 2	通道 3	通道 4	通道 5
计数时钟	MP $\phi$ /1 MP $\phi$ /4 MP $\phi$ /16 MP $\phi$ /64 TCLKA TCLKB TCLKC TCLKD	MP $\phi$ /1 MP $\phi$ /4 MP $\phi$ /16 MP $\phi$ /64 MP $\phi$ /256 TCLKA TCLKB	MP $\phi$ /1 MP $\phi$ /4 MP $\phi$ /16 MP $\phi$ /64 MP $\phi$ /1024 TCLKA TCLKB TCLKC	MP $\phi$ /1 MP $\phi$ /4 MP $\phi$ /16 MP $\phi$ /64 MP $\phi$ /256 MP $\phi$ /1024 TCLKA TCLKB	MP $\phi$ /1 MP $\phi$ /4 MP $\phi$ /16 MP $\phi$ /64 MP $\phi$ /256 MP $\phi$ /1024 TCLKA TCLKB	MP $\phi$ /1 MP $\phi$ /4 MP $\phi$ /16 MP $\phi$ /64
通用寄存器 (TGR)	TGRA_0 TGRB_0 TGRE_0	TGRA_1 TGRB_1	TGRA_2 TGRB_2	TGRA_3 TGRB_3	TGRA_4 TGRB_4	TGRU_5 TGRV_5 TGRW_5
通用寄存器 / 缓冲寄存器	TGRC_0 TGRD_0 TGRF_0	—	—	TGRC_3 TGRD_3	TGRC_4 TGRD_4	—
输入 / 输出引脚	TIOC0A TIOC0B TIOC0C TIOC0D	TIOC1A TIOC1B	TIOC2A TIOC2B	TIOC3A TIOC3B TIOC3C TIOC3D	TIOC4A TIOC4B TIOC4C TIOC4D	输入引脚 TIC5U TIC5V TIC5W
计数器的清除功能	TGR 的比较匹 配或者输入捕捉	TGR 的比较匹 配或者输入捕捉	TGR 的比较匹 配或者输入捕捉	TGR 的比较匹 配或者输入捕捉	TGR 的比较匹 配或者输入捕捉	TGR 的比较匹 配或者输入捕捉
比较匹配 输出	0 输出	○	○	○	○	—
	1 输出	○	○	○	○	—
	交替输出	○	○	○	○	—
输入捕捉功能	○	○	○	○	○	○
同步运行	○	○	○	○	○	—
PWM 模式 1	○	○	○	○	○	—
PWM 模式 2	○	○	○	—	—	—
互补 PWM 模式	—	—	—	○	○	—
复位 PWM 模式	—	—	—	○	○	—
AC 同步马达驱动模 式	○	—	—	○	○	—
相位计数模式	—	○	○	—	—	—
缓冲运行	○	—	—	○	○	—
用于空载时间补偿 的计数器功能	—	—	—	—	—	○
DMAC 的启动	TGRA_0 的 比较匹配或者 输入捕捉	TGRA_1 的 比较匹配或者 输入捕捉	TGRA_2 的 比较匹配或者 输入捕捉	TGRA_3 的 比较匹配或者 输入捕捉	TGRA_4 的 比较匹配或者 输入捕捉	—

项 目	通道 0	通道 1	通道 2	通道 3	通道 4	通道 5
DTC 的启动	TGR 的 比较匹配或者 输入捕捉	TGR 的 比较匹配或者 输入捕捉	TGR 的 比较匹配或者 输入捕捉	TGR 的 比较匹配或者 输入捕捉	TGR 的 比较匹配或者 输入捕捉 和 TCNT 的 上溢 / 下溢	TGR 的 比较匹配或者 输入捕捉
A/D 转换的开始触 发	TGRA_0 的 比较匹配或者 输入捕捉 TGRE_0 的比 较匹配	TGRA_1 的 比较匹配或者 输入捕捉	TGRA_2 的 比较匹配或者 输入捕捉	TGRA_3 的 比较匹配或者 输入捕捉	TGRA_4 的 比较匹配或者 输入捕捉 在互补 PWM 模 式中 TCNT_4 的下溢 (波谷)	—
中断源	7 个源 • 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 • 比较匹配 0E • 比较匹配 0F • 上溢	4 个源 • 比较匹配 / 输入捕捉 1A • 比较匹配 / 输入捕捉 1B • 上溢 • 下溢	4 个源 • 比较匹配 / 输入捕捉 2A • 比较匹配 / 输入捕捉 2B • 上溢 • 下溢	5 个源 • 比较匹配 / 输入捕捉 3A • 比较匹配 / 输入捕捉 3B • 比较匹配 / 输入捕捉 3C • 比较匹配 / 输入捕捉 3D • 上溢	5 个源 • 比较匹配 / 输入捕捉 4A • 比较匹配 / 输入捕捉 4B • 比较匹配 / 输入捕捉 4C • 比较匹配 / 输入捕捉 4D • 上溢 / 下溢	3 个源 • 比较匹配 / 输入捕捉 5U • 比较匹配 / 输入捕捉 5V • 比较匹配 / 输入捕捉 5W
A/D 转换的开始请 求延迟功能	—	—	—	—	• TADCORA_4 和 TCNT_4 一致时, 请求 A/D 转换的开 始 • TADCORB_4 和 TCNT_4 一致时, 请求 A/D 转换的开 始	—
跳过中断功能	—	—	—	• 跳过 TGRA_3 的 比较匹配中断	• 跳过 TCIV_4 的 中断	—

## 【符号说明】

○: 能

—: 不能

MTU2 的框图如图 11.1 所示。

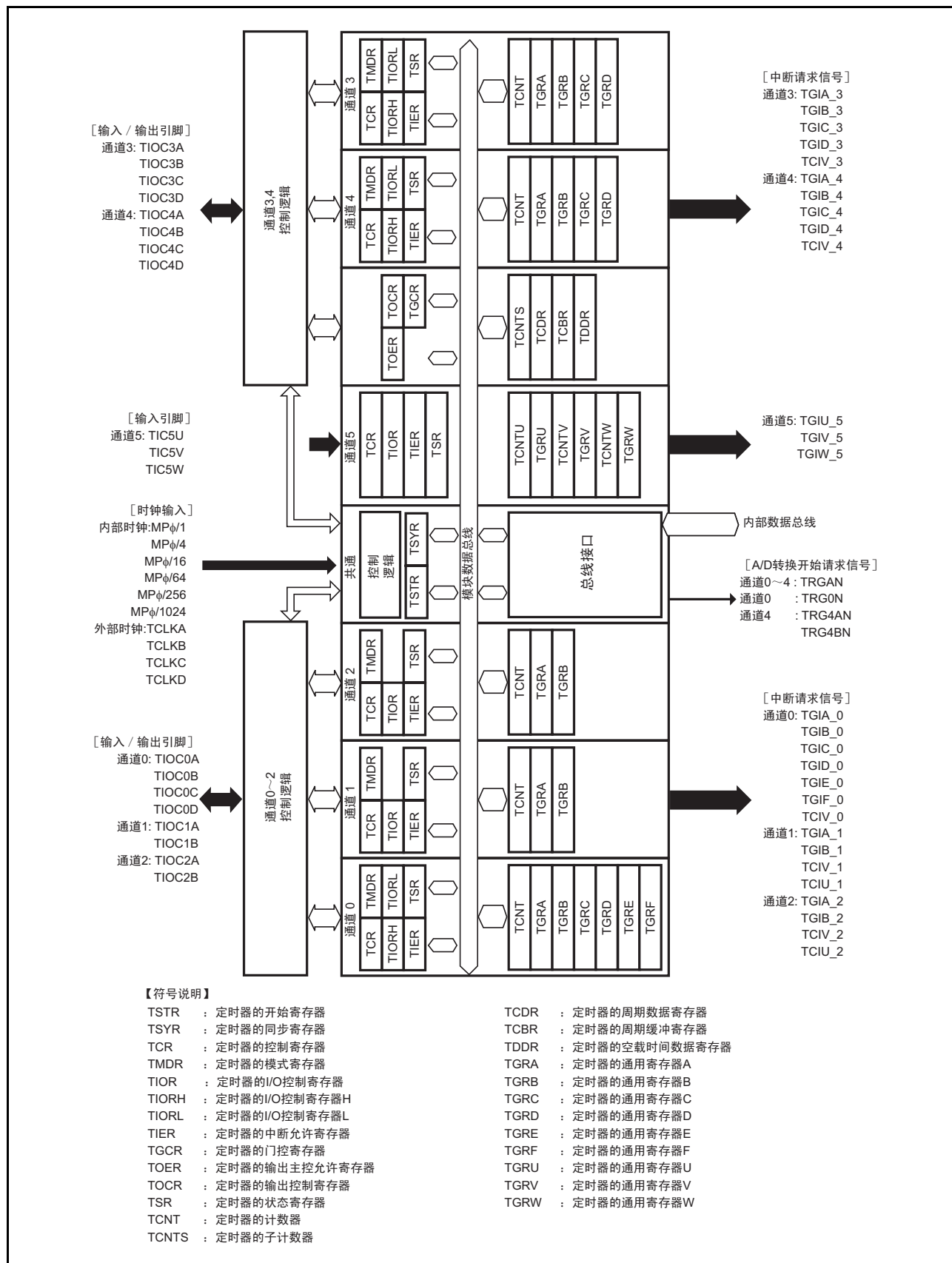


图 11.1 MTU2 的框图

## 11.2 输入 / 输出引脚

表 11.2 引脚结构

通道	引脚名	输入 / 输出	功 能
通用	TCLKA	输入	外部时钟 A 的输入引脚 (通道 1 的相位计数模式的 A 相输入)
	TCLKB	输入	外部时钟 B 的输入引脚 (通道 1 的相位计数模式的 B 相输入)
	TCLKC	输入	外部时钟 C 的输入引脚 (通道 2 的相位计数模式的 A 相输入)
	TCLKD	输入	外部时钟 D 的输入引脚 (通道 2 的相位计数模式的 B 相输入)
0	TIOC0A	输入 / 输出	TGRA_0 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	TIOC0B	输入 / 输出	TGRB_0 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	TIOC0C	输入 / 输出	TGRC_0 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	TIOC0D	输入 / 输出	TGRD_0 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
1	TIOC1A	输入 / 输出	TGRA_1 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	TIOC1B	输入 / 输出	TGRB_1 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
2	TIOC2A	输入 / 输出	TGRA_2 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	TIOC2B	输入 / 输出	TGRB_2 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
3	TIOC3A	输入 / 输出	TGRA_3 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	TIOC3B	输入 / 输出	TGRB_3 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	TIOC3C	输入 / 输出	TGRC_3 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	TIOC3D	输入 / 输出	TGRD_3 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
4	TIOC4A	输入 / 输出	TGRA_4 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	TIOC4B	输入 / 输出	TGRB_4 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	TIOC4C	输入 / 输出	TGRC_4 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	TIOC4D	输入 / 输出	TGRD_4 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
5	TIC5U	输入	TGRU_5 的输入捕捉的输入引脚 / 外部脉冲的输入引脚
	TIC5V	输入	TGRV_5 的输入捕捉的输入引脚 / 外部脉冲的输入引脚
	TIC5W	输入	TGRW_5 的输入捕捉的输入引脚 / 外部脉冲的输入引脚

### 11.3 寄存器的说明

在 MTU2 中各通道有以下寄存器。有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 27 章 寄存器一览”。为了区分各通道的寄存器名，将通道 0 的 TCR 记载为 TCR\_0。

表 11.3 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
定时器的控制寄存器_3	TCR_3	R/W	H'00	H'FFFC200	8、16、32
定时器的控制寄存器_4	TCR_4	R/W	H'00	H'FFFC201	8
定时器的模式寄存器_3	TMDR_3	R/W	H'00	H'FFFC202	8、16
定时器的模式寄存器_4	TMDR_4	R/W	H'00	H'FFFC203	8
定时器的 I/O 控制寄存器 H_3	TIORH_3	R/W	H'00	H'FFFC204	8、16、32
定时器的 I/O 控制寄存器 L_3	TIORL_3	R/W	H'00	H'FFFC205	8
定时器的 I/O 控制寄存器 H_4	TIORH_4	R/W	H'00	H'FFFC206	8、16
定时器的 I/O 控制寄存器 L_4	TIORL_4	R/W	H'00	H'FFFC207	8
定时器的中断允许寄存器_3	TIER_3	R/W	H'00	H'FFFC208	8、16
定时器的中断允许寄存器_4	TIER_4	R/W	H'00	H'FFFC209	8
定时器的输出主控允许寄存器	TOER	R/W	H'C0	H'FFFC20A	8
定时器的门控寄存器	TGCR	R/W	H'80	H'FFFC20D	8
定时器的输出控制寄存器 1	TOCR1	R/W	H'00	H'FFFC20E	8、16
定时器的输出控制寄存器 2	TOCR2	R/W	H'00	H'FFFC20F	8
定时器的计数器_3	TCNT_3	R/W	H'0000	H'FFFC210	16、32
定时器的计数器_4	TCNT_4	R/W	H'0000	H'FFFC212	16
定时器的周期数据寄存器	TCDR	R/W	H'FFFF	H'FFFC214	16、32
定时器的空载时间数据寄存器	TDDR	R/W	H'FFFF	H'FFFC216	16
定时器的通用寄存器 A_3	TGRA_3	R/W	H'FFFF	H'FFFC218	16、32
定时器的通用寄存器 B_3	TGRB_3	R/W	H'FFFF	H'FFFC21A	16
定时器的通用寄存器 A_4	TGRA_4	R/W	H'FFFF	H'FFFC21C	16、32
定时器的通用寄存器 B_4	TGRB_4	R/W	H'FFFF	H'FFFC21E	16
定时器的子计数器	TCNTS	R	H'0000	H'FFFC220	16、32
定时器的周期缓冲寄存器	TCBR	R/W	H'FFFF	H'FFFC222	16
定时器的通用寄存器 C_3	TGRC_3	R/W	H'FFFF	H'FFFC224	16、32
定时器的通用寄存器 D_3	TGRD_3	R/W	H'FFFF	H'FFFC226	16
定时器的通用寄存器 C_4	TGRC_4	R/W	H'FFFF	H'FFFC228	16、32
定时器的通用寄存器 D_4	TGRD_4	R/W	H'FFFF	H'FFFC22A	16
定时器的状态寄存器_3	TSR_3	R/W	H'C0	H'FFFC22C	8、16
定时器的状态寄存器_4	TSR_4	R/W	H'C0	H'FFFC22D	8
定时器的跳过中断设定寄存器	TITCR	R/W	H'00	H'FFFC230	8、16
定时器的跳过中断次数计数器	TITCNT	R	H'00	H'FFFC231	8
定时器的缓冲传送设定寄存器	TBTER	R/W	H'00	H'FFFC232	8

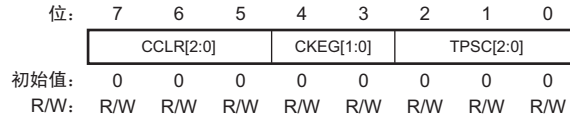
寄存器名	略称	R/W	初始值	地址	存取长度
定时器的空载时间允许寄存器	TDER	R/W	H'01	H'FFFFC234	8
定时器的输出电平缓冲寄存器	TOLBR	R/W	H'00	H'FFFFC236	8
定时器的缓冲运行传送模式寄存器_3	TBTM_3	R/W	H'00	H'FFFFC238	8、16
定时器的缓冲运行传送模式寄存器_4	TBTM_4	R/W	H'00	H'FFFFC239	8
定时器的 A/D 转换开始请求的控制寄存器	TADCR	R/W	H'0000	H'FFFFC240	16
定时器的 A/D 转换开始请求的周期设定寄存器 A_4	TADCORA_4	R/W	H'FFFF	H'FFFFC244	16、32
定时器的 A/D 转换开始请求的周期设定寄存器 B_4	TADCORB_4	R/W	H'FFFF	H'FFFFC246	16
定时器的 A/D 转换开始请求的周期设定缓冲寄存器 A_4	TADCOBRA_4	R/W	H'FFFF	H'FFFFC248	16、32
定时器的 A/D 转换开始请求的周期设定缓冲寄存器 B_4	TADCOBRB_4	R/W	H'FFFF	H'FFFFC24A	16
定时器的波形控制寄存器	TWCR	R/W	H'00	H'FFFFC260	8
定时器的开始寄存器	TSTR	R/W	H'00	H'FFFFC280	8、16
定时器的同步寄存器	TSYR	R/W	H'00	H'FFFFC281	8
定时器的计数器同步开始寄存器	TCSYSTR	R/W	H'00	H'FFFFC282	8
定时器的读写允许寄存器	TRWER	R/W	H'01	H'FFFFC284	8
定时器的控制寄存器_0	TCR_0	R/W	H'00	H'FFFFC300	8、16、32
定时器的模式寄存器_0	TMDR_0	R/W	H'00	H'FFFFC301	8
定时器的 I/O 控制寄存器 H_0	TIORH_0	R/W	H'00	H'FFFFC302	8、16
定时器的 I/O 控制寄存器 L_0	TIORL_0	R/W	H'00	H'FFFFC303	8
定时器的中断允许寄存器_0	TIER_0	R/W	H'00	H'FFFFC304	8、16、32
定时器的状态寄存器_0	TSR_0	R/W	H'C0	H'FFFFC305	8
定时器的计数器_0	TCNT_0	R/W	H'0000	H'FFFFC306	16
定时器的通用寄存器 A_0	TGRA_0	R/W	H'FFFF	H'FFFFC308	16、32
定时器的通用寄存器 B_0	TGRB_0	R/W	H'FFFF	H'FFFFC30A	16
定时器的通用寄存器 C_0	TGRC_0	R/W	H'FFFF	H'FFFFC30C	16、32
定时器的通用寄存器 D_0	TGRD_0	R/W	H'FFFF	H'FFFFC30E	16
定时器的通用寄存器 E_0	TGRE_0	R/W	H'FFFF	H'FFFFC320	16、32
定时器的通用寄存器 F_0	TGRF_0	R/W	H'FFFF	H'FFFFC322	16
定时器的中断允许寄存器 2_0	TIER2_0	R/W	H'00	H'FFFFC324	8、16
定时器的状态寄存器 2_0	TSR2_0	R/W	H'C0	H'FFFFC325	8
定时器的缓冲运行传送模式寄存器_0	TBTM_0	R/W	H'00	H'FFFFC326	8
定时器的控制寄存器_1	TCR_1	R/W	H'00	H'FFFFC380	8、16
定时器的模式寄存器_1	TMDR_1	R/W	H'00	H'FFFFC381	8
定时器的 I/O 控制寄存器_1	TIOR_1	R/W	H'00	H'FFFFC382	8

寄存器名	略称	R/W	初始值	地址	存取长度
定时器的中断允许寄存器_1	TIER_1	R/W	H'00	H'FFFFC384	8、16、32
定时器的状态寄存器_1	TSR_1	R/W	H'C0	H'FFFFC385	8
定时器的计数器_1	TCNT_1	R/W	H'0000	H'FFFFC386	16
定时器的通用寄存器 A_1	TGRA_1	R/W	H'FFFF	H'FFFFC388	16、32
定时器的通用寄存器 B_1	TGRB_1	R/W	H'FFFF	H'FFFFC38A	16
定时器的输入捕捉控制寄存器	TICCR	R/W	H'00	H'FFFFC390	8
定时器的控制寄存器_2	TCR_2	R/W	H'00	H'FFFFC400	8、16
定时器的模式寄存器_2	TMDR_2	R/W	H'00	H'FFFFC401	8
定时器的 I/O 控制寄存器_2	TIOR_2	R/W	H'00	H'FFFFC402	8
定时器的中断允许寄存器_2	TIER_2	R/W	H'00	H'FFFFC404	8、16、32
定时器的状态寄存器_2	TSR_2	R/W	H'C0	H'FFFFC405	8
定时器的计数器_2	TCNT_2	R/W	H'0000	H'FFFFC406	16
定时器的通用寄存器 A_2	TGRA_2	R/W	H'FFFF	H'FFFFC408	16、32
定时器的通用寄存器 B_2	TGRB_2	R/W	H'FFFF	H'FFFFC40A	16
定时器的计数器 U_5	TCNTU_5	R/W	H'0000	H'FFFFC480	16、32
定时器的通用寄存器 U_5	TGRU_5	R/W	H'FFFF	H'FFFFC482	16
定时器的控制寄存器 U_5	TCRU_5	R/W	H'00	H'FFFFC484	8
定时器的 I/O 控制寄存器 U_5	TIORU_5	R/W	H'00	H'FFFFC486	8
定时器的计数器 V_5	TCNTV_5	R/W	H'0000	H'FFFFC490	16、32
定时器的通用寄存器 V_5	TGRV_5	R/W	H'FFFF	H'FFFFC492	16
定时器的控制寄存器 V_5	TCRV_5	R/W	H'00	H'FFFFC494	8
定时器的 I/O 控制寄存器 V_5	TIORV_5	R/W	H'00	H'FFFFC496	8
定时器的计数器 W_5	TCNTW_5	R/W	H'0000	H'FFFFC4A0	16、32
定时器的通用寄存器 W_5	TGRW_5	R/W	H'FFFF	H'FFFFC4A2	16
定时器的控制寄存器 W_5	TCRW_5	R/W	H'00	H'FFFFC4A4	8
定时器的 I/O 控制寄存器 W_5	TIORW_5	R/W	H'00	H'FFFFC4A6	8
定时器的状态寄存器_5	TSR_5	R/W	H'00	H'FFFFC4B0	8
定时器的中断允许寄存器_5	TIER_5	R/W	H'00	H'FFFFC4B2	8
定时器的开始寄存器_5	TSTR_5	R/W	H'00	H'FFFFC4B4	8
定时器的比较匹配清除寄存器	TCNTCMPCLR	R/W	H'00	H'FFFFC4B6	8



### 11.3.1 定时器的控制寄存器 (TCR)

TCR 是控制各通道 TCNT 的 8 位可读写寄存器。在 MTU2 中，通道 0 ~ 4 各有 1 个，通道 5 有 3 个 TCRU/V/W\_5，共计 8 个 TCR。必须在停止 TCNT 运行的状态下进行 TCR 的设定。



位	位名	初始值	R/W	说 明
7 ~ 5	CCLR[2:0]	000	R/W	计数器的清除 2、1、0 选择 TCNT 的计数器清除源。详细内容请参照表 11.4 和表 11.5。
4、3	CKEG[1:0]	00	R/W	时钟边沿 1、0 选择输入时钟的边沿。如果在双边沿对内部时钟进行计数，输入时钟的周期就为 1/2（例如：MP $\phi$ /4 的双边沿 = MP $\phi$ /2 的上升沿）。在通道 1、2 使用相位计数模式时，忽视本设定，优先设定相位计数模式。内部时钟的边沿选择在输入时钟为 MP $\phi$ /4 或者慢于 MP $\phi$ /4 时有效。如果选择 MP $\phi$ /1 或者其他通道的上溢 / 下溢为输入时钟，虽然能写值，但是在运行时为初始值。 00: 在上升沿进行计数 01: 在下降沿进行计数 1x: 在双边沿进行计数
2 ~ 0	TPSC[2:0]	000	R/W	定时器的预定标器 2、1、0 选择 TCNT 的计数器时钟，各通道能独立选择时钟源。详细内容请参照表 11.6 ~ 表 11.10。

【符号说明】 x: Don't care

表 11.4 CCLR2 ~ CCLR0 (通道 0、3、4)

通道	bit7	bit6	bit5	说 明
	CCLR2	CCLR1	CCLR0	
0、3、4	0	0	0	禁止 TCNT 的清除
	0	0	1	通过 TGRA 的比较匹配 / 输入捕捉进行 TCNT 的清除
	0	1	0	通过 TGRB 的比较匹配 / 输入捕捉进行 TCNT 的清除
	0	1	1	通过同步清除 / 同步运行的其他通道的计数器清除进行 TCNT 的清除 *1
	1	0	0	禁止 TCNT 的清除
	1	0	1	通过 TGRC 的比较匹配 / 输入捕捉进行 TCNT 的清除 *2
	1	1	0	通过 TGRD 的比较匹配 / 输入捕捉进行 TCNT 的清除 *2
	1	1	1	通过同步清除 / 同步运行的其他通道的计数器清除进行 TCNT 的清除 *1

【注】 \*1 通过将 TSYR 的 SYNC 位设定为 1 进行同步运行的设定。

\*2 在将 TGRC 或者 TGRD 用作缓冲寄存器时，优先设定缓冲寄存器。此时，因为不发生比较匹配 / 输入捕捉，所以不清除 TCNT。

表 11.5 CCLR2 ~ CCLR0 (通道 1、2)

通道	bit7	bit6	bit5	说 明
	保留 *2	CCLR1	CCLR0	
1、2	0	0	0	禁止 TCNT 的清除
	0	0	1	通过 TGRA 的比较匹配 / 输入捕捉进行 TCNT 的清除
	0	1	0	通过 TGRB 的比较匹配 / 输入捕捉进行 TCNT 的清除
	0	1	1	通过同步清除 / 同步运行的其他通道的计数器清除进行 TCNT 的清除 *1

【注】 \*1 通过将 TSYR 的 SYNC 位设定为 1 进行同步运行的设定。

\*2 在通道 1、2 中，bit7 为保留位。读的值总是 0，写操作无效。

表 11.6 TPSC2 ~ TPSC0 (通道 0)

通道	bit2	bit1	bit0	说 明
	TPSC2	TPSC1	TPSC0	
0	0	0	0	内部时钟：通过 MP $\phi$ /1 进行计数
	0	0	1	内部时钟：通过 MP $\phi$ /4 进行计数
	0	1	0	内部时钟：通过 MP $\phi$ /16 进行计数
	0	1	1	内部时钟：通过 MP $\phi$ /64 进行计数
	1	0	0	外部时钟：通过 TCLKA 引脚的输入进行计数
	1	0	1	外部时钟：通过 TCLKB 引脚的输入进行计数
	1	1	0	外部时钟：通过 TCLKC 引脚的输入进行计数
	1	1	1	外部时钟：通过 TCLKD 引脚的输入进行计数

表 11.7 TPSC2 ~ TPSC0 (通道 1)

通道	bit2	bit1	bit0	说 明
	TPSC2	TPSC1	TPSC0	
1	0	0	0	内部时钟：通过 MP $\phi$ /1 进行计数
	0	0	1	内部时钟：通过 MP $\phi$ /4 进行计数
	0	1	0	内部时钟：通过 MP $\phi$ /16 进行计数
	0	1	1	内部时钟：通过 MP $\phi$ /64 进行计数
	1	0	0	外部时钟：通过 TCLKA 引脚的输入进行计数
	1	0	1	外部时钟：通过 TCLKB 引脚的输入进行计数
	1	1	0	内部时钟：通过 MP $\phi$ /256 进行计数
	1	1	1	通过 TCNT_2 的上溢 / 下溢进行计数

【注】 在通道 1 为相位计数模式时，此设定无效。

表 11.8 TPSC2 ~ TPSC0 (通道 2)

通道	bit2	bit1	bit0	说 明
	TPSC2	TPSC1	TPSC0	
2	0	0	0	内部时钟: 通过 MP $\phi$ /1 进行计数
	0	0	1	内部时钟: 通过 MP $\phi$ /4 进行计数
	0	1	0	内部时钟: 通过 MP $\phi$ /16 进行计数
	0	1	1	内部时钟: 通过 MP $\phi$ /64 进行计数
	1	0	0	外部时钟: 通过 TCLKA 引脚的输入进行计数
	1	0	1	外部时钟: 通过 TCLKB 引脚的输入进行计数
	1	1	0	外部时钟: 通过 TCLKC 引脚的输入进行计数
	1	1	1	内部时钟: 通过 MP $\phi$ /1024 进行计数

【注】 在通道 2 为相位计数模式时, 此设定无效。

表 11.9 TPSC2 ~ TPSC0 (通道 3、4)

通道	bit2	bit1	bit0	说 明
	TPSC2	TPSC1	TPSC0	
3、4	0	0	0	内部时钟: 通过 MP $\phi$ /1 进行计数
	0	0	1	内部时钟: 通过 MP $\phi$ /4 进行计数
	0	1	0	内部时钟: 通过 MP $\phi$ /16 进行计数
	0	1	1	内部时钟: 通过 MP $\phi$ /64 进行计数
	1	0	0	内部时钟: 通过 MP $\phi$ /256 进行计数
	1	0	1	内部时钟: 通过 MP $\phi$ /1024 进行计数
	1	1	0	外部时钟: 通过 TCLKA 引脚的输入进行计数
	1	1	1	外部时钟: 通过 TCLKB 引脚的输入进行计数

表 11.10 TPSC1、TPSC0 (通道 5)

通道	bit1	bit0	说 明
	TPSC1	TPSC0	
5	0	0	内部时钟: 通过 MP $\phi$ /1 进行计数
	0	1	内部时钟: 通过 MP $\phi$ /4 进行计数
	1	0	内部时钟: 通过 MP $\phi$ /16 进行计数
	1	1	内部时钟: 通过 MP $\phi$ /64 进行计数

【注】 在通道 5 中, bit7 ~ 2 为保留位。读写的值总是 0。

## 11.3.2 定时器的模式寄存器 (TMDR)

TMDR 是 8 位可读写寄存器，设定各通道的运行模式。在 MTU2 中，通道 0 ~ 4 各有 1 个，共计 5 个 TMDR。必须在停止 TCNT 运行的状态下进行 TMDR 的设定。

位:	7	6	5	4	3	2	1	0
	-	BFE	BFB	BFA	MD[3:0]			
初始值:	0	0	0	0	0	0	0	0
R/W:	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7	—	0	—	保留位 读写的值总是 0。
6	BFE	0	R/W	缓冲运行 E 选择使 TGRE_0 和 TGRF_0 进行通常运行还是进行缓冲运行。 即使将 TGRF 用作缓冲寄存器，也会发生 TGRF 的比较匹配。 在通道 1、2、3、4 中，此位为保留位，读写的值总是 0。 0: TGRE_0 和 TGRF_0 进行通常运行 1: TGRE_0 和 TGRF_0 进行缓冲运行
5	BFB	0	R/W	缓冲运行 B 设定使 TGRB 进行通常运行还是组合 TGRB 和 TGRD 进行缓冲运行。如果将 TGRD 用作缓冲寄存器，除了互补 PWM 模式以外，不发生 TGRD 的输入捕捉 / 输出比较。但是，在互补 PWM 模式中发生 TGRD 的比较匹配。另外，如果在互补 PWM 模式的 Tb 区间发生比较匹配，TGFD 就被置位，所以必须将定时器的中断允许寄存器_3/4 (TIER_3/4) 的 TGIED 位设定为 0。 在没有 TGRD 的通道 1、2 中，此位为保留位，读写的值总是 0。 0: TGRB 和 TGRD 进行通常运行 1: TGRB 和 TGRD 进行缓冲运行
4	BFA	0	R/W	缓冲运行 A 设定使 TGRA 进行通常运行还是组合 TGRA 和 TGRC 进行缓冲运行。如果将 TGRC 用作缓冲寄存器，除了互补 PWM 模式以外，不发生 TGRC 的输入捕捉 / 输出比较。但是，在互补 PWM 模式中发生 TGRC 的比较匹配。另外，如果在互补 PWM 模式的 Tb 区间发生通道 4 的比较匹配，TGFC 就被置位，所以必须将定时器的中断允许寄存器_4 (TIER_4) 的 TGIEC 位设定为 0。 在没有 TGRC 的通道 1、2 中，此位为保留位，读写的值总是 0。 0: TGRA 和 TGRC 进行通常运行 1: TGRA 和 TGRC 进行缓冲运行
3 ~ 0	MD[3:0]	0000	R/W	模式 3 ~ 0 MD3 ~ MD0 设定定时器的运行模式。 详细内容请参照表 11.11。

表 11.11 设定 MD3 ~ MD0 位的运行模式

bit3	bit2	bit1	bit0	说 明
MD3	MD2	MD1	MD0	
0	0	0	0	通常运行
0	0	0	1	禁止设定
0	0	1	0	PWM 模式 1
0	0	1	1	PWM 模式 2*1
0	1	0	0	相位计数模式 1*2
0	1	0	1	相位计数模式 2*2
0	1	1	0	相位计数模式 3*2
0	1	1	1	相位计数模式 4*2
1	0	0	0	复位同步 PWM 模式 *3
1	0	0	1	禁止设定
1	0	1	x	禁止设定
1	1	0	0	禁止设定
1	1	0	1	互补 PWM 模式 1 (在波峰进行传送) *3
1	1	1	0	互补 PWM 模式 2 (在波谷进行传送) *3
1	1	1	1	互补 PWM 模式 3 (在波峰和波谷进行传送) *3

【符号说明】 x: Don't care

【注】 \*1 不能对通道 3、4 设定 PWM 模式 2。

\*2 不能对通道 0、3、4 设定相位计数模式。

\*3 只有通道 3 能设定复位同步 PWM 模式或者互补 PWM 模式。

如果对通道 3 设定复位同步 PWM 模式或者互补 PWM 模式，通道 4 的设定无效，自动服从通道 3 的设定。但是，不能对通道 4 设定复位同步 PWM 模式或者互补 PWM 模式。

不能对通道 0、1、2 设定复位同步 PWM 模式或者互补 PWM 模式。

### 11.3.3 定时器的 I/O 控制寄存器 (TIOR)

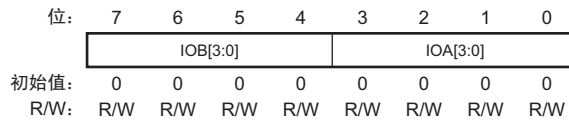
TIOR 是控制 TGR 的 8 位可读写寄存器。在 MTU2 中，通道 0、3、4 各有 2 个，通道 1、2 各有 1 个，通道 5 有 3 个 TIORU/V/W\_5，共计 11 个 TIOR。

在 TMDR 的设定为通常运行、PWM 模式、相位计数模式时，设定 TIOR。

TIOR 指定的初始输出在计数器停止（将 TSTR 的 CST 位清 0）的状态下有效。另外，在 PWM 模式 2 的情况下，指定计数器清 0 时的输出。

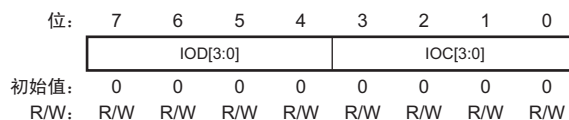
如果将 TGRC 或者 TGRD 设定为缓冲运行，本设定无效，作为缓冲寄存器运行。

- TIORH\_0、TIOR\_1、TIOR\_2、TIORH\_3、TIORH\_4



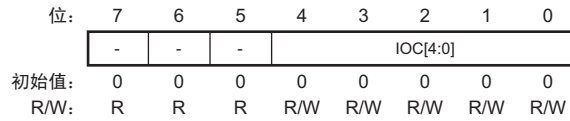
位	位名	初始值	R/W	说 明
7 ~ 4	IOB[3:0]	0000	R/W	I/O 控制 B3 ~ B0 IOB3 ~ IOB0 位设定 TGRB 的功能。 请参照下述表： TIORH_0: 表 11.12    TIOR_1: 表 11.14    TIOR_2: 表 11.15 TIORH_3: 表 11.16    TIORH_4: 表 11.18
3 ~ 0	IOA[3:0]	0000	R/W	I/O 控制 A3 ~ A0 IOA3 ~ IOA0 设定 TGRA 的功能。 请参照下述表： TIORH_0: 表 11.20    TIOR_1: 表 11.22    TIOR_2: 表 11.23 TIORH_3: 表 11.24    TIORH_4: 表 11.26

- TIORL\_0、TIORL\_3、TIORL\_4



位	位名	初始值	R/W	说 明
7 ~ 4	IOD[3:0]	0000	R/W	I/O 控制 D3 ~ D0 IOD3 ~ IOD0 位设定 TGRD 的功能。 请参照下述表： TIORL_0: 表 11.13    TIORL_3: 表 11.17    TIORL_4: 表 11.19
3 ~ 0	IOC[3:0]	0000	R/W	I/O 控制 C3 ~ C0 IOC3 ~ IOC0 位设定 TGRC 的功能。 请参照下述表： TIORL_0: 表 11.21    TIORL_3: 表 11.25    TIORL_4: 表 11.27

- TIORU\_5、TIORV\_5、TIORW\_5



位	位名	初始值	R/W	说 明
7 ~ 5	—	全 0	R	保留位 读写的值总是 0。
4 ~ 0	IOC[4:0]	00000	R/W	I/O 控制 C4 ~ C0 IOC4 ~ IOC0 位设定 TGRU/V/W_5 的功能。 详细内容请参照表 11.28。

表 11.12 TIORH\_0 (通道 0)

bit7	bit6	bit5	bit4	说 明	
IOB3	IOB2	IOB1	IOB0	TGRB_0 的功能	TIOC0B 引脚的功能
0	0	0	0	输出比较寄存器	保持输出 *
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	0	0	0	输入捕捉寄存器	在上升沿进行输入捕捉
1	0	0	1		在下降沿进行输入捕捉
1	0	1	x		在双边沿进行输入捕捉
1	1	x	x		捕捉输入源是通道 1/ 计数时钟, 通过 TCNT_1 的递增计数 / 递减计数进行输入捕捉

【符号说明】 x: Don't care

【注】 \* 从上电复位后到设定 TIOR 为止, 输出 0。

表 11.13 TIORL\_0 (通道 0)

bit7	bit6	bit5	bit4	说 明	
IOD3	IOD2	IOD1	IOD0	TGRD_0 的功能	TIOC0D 引脚的功能
0	0	0	0	输出比较寄存器 *2	保持输出 *1
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配进行 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	0	0	0	输入捕捉寄存器 *2	在上升沿进行输入捕捉
1	0	0	1		在下降沿进行输入捕捉
1	0	1	x		在双边沿进行输入捕捉
1	1	x	x		捕捉输入源是通道 1/ 计数时钟, 通过 TCNT_1 的递增计数 / 递减计数进行输入捕捉

【符号说明】 x: Don't care

【注】 \*1 从上电复位后到设定 TIOR 为止, 输出 0。

\*2 如果将 TMDR\_0 的 BFB 位设定为 1, TGRD\_0 用作缓冲寄存器, 本设定就无效并且不发生输入捕捉 / 输出比较。



表 11.14 TIOR\_1 (通道 1)

bit7	bit6	bit5	bit4	说 明	
IOB3	IOB2	IOB1	IOB0	TGRB_1 的功能	TIOC1B 引脚的功能
0	0	0	0	TGRB_1 为输出比较寄存器	保持输出 *
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	0	0	0	输入捕捉寄存器	在上升沿进行输入捕捉
1	0	0	1		在下降沿进行输入捕捉
1	0	1	x		在双边沿进行输入捕捉
1	1	x	x		通过发生 TGRC_0 的比较匹配 / 输入捕捉进行输入捕捉

【符号说明】 x: Don't care

【注】 \* 从上电复位后到设定 TIOR 为止, 输出 0。

表 11.15 TIOR\_2 (通道 2)

bit7	bit6	bit5	bit4	说 明	
IOB3	IOB2	IOB1	IOB0	TGRB_2 的功能	TIOC2B 引脚的功能
0	0	0	0	TGRB_2 为输出比较寄存器	保持输出 *
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	x	0	0	输入捕捉寄存器	在上升沿进行输入捕捉
1	x	0	1		在下降沿进行输入捕捉
1	x	1	x		在双边沿进行输入捕捉

【符号说明】 x: Don't care

【注】 \* 从上电复位后到设定 TIOR 为止, 输出 0。

表 11.16 TIORH\_3 (通道 3)

bit7	bit6	bit5	bit4	说 明	
IOB3	IOB2	IOB1	IOB0	TGRB_3 的功能	TIOC3B 引脚的功能
0	0	0	0	输出比较寄存器	保持输出 *
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出、 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	x	0	0	输入捕捉寄存器	在上升沿进行输入捕捉
1	x	0	1		在下降沿进行输入捕捉
1	x	1	x		在双边沿进行输入捕捉

【符号说明】 x: Don't care

【注】 \* 从上电复位后到设定 TIOR 为止, 输出 0。

表 11.17 TIORL\_3 (通道 3)

bit7	bit6	bit5	bit4	说 明	
IOD3	IOD2	IOD1	IOD0	TGRD_3 的功能	TIOC3D 引脚的功能
0	0	0	0	输出比较寄存器 *2	保持输出 *1
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	x	0	0	输入捕捉寄存器 *2	在上升沿进行输入捕捉
1	x	0	1		在下降沿进行输入捕捉
1	x	1	x		在双边沿进行输入捕捉

【符号说明】 x: Don't care

【注】 \*1 从上电复位后到设定 TIOR 为止, 输出 0。

\*2 如果将 TMDR\_3 的 BFB 位设定为 1, TGRD\_3 用作缓冲寄存器, 本设定就无效并且不发生输入捕捉 / 输出比较。

表 11.18 TIORH\_4 (通道 4)

bit7	bit6	bit5	bit4	说 明	
IOB3	IOB2	IOB1	IOB0	TGRB_4 的功能	TIOC4B 引脚的功能
0	0	0	0	TGRB_4 为输出比较寄存器	保持输出 *
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	x	0	0	输入捕捉寄存器	在上升沿进行输入捕捉
1	x	0	1		在下降沿进行输入捕捉
1	x	1	x		在双边沿进行输入捕捉

【符号说明】 x: Don't care

【注】 \* 从上电复位后到设定 TIOR 为止, 输出 0。

表 11.19 TIORL\_4 (通道 4)

bit7	bit6	bit5	bit4	说 明	
IOD3	IOD2	IOD1	IOD0	TGRD_4 的功能	TIOC4D 引脚的功能
0	0	0	0	输出比较寄存器 *2	保持输出 *1
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	x	0	0		输入捕捉寄存器 *2
1	x	0	1	在下降沿进行输入捕捉	
1	x	1	x	在双边沿进行输入捕捉	

【符号说明】 x: Don't care

【注】 \*1 从上电复位后到设定 TIOR 为止，输出 0。

\*2 如果将 TMDR\_4 的 BFB 位设定为 1，TGRD\_4 用作缓冲寄存器，本设定就无效并且不发生输入捕捉 / 输出比较。

表 11.20 TIORH\_0 (通道 0)

bit3	bit2	bit1	bit0	说 明	
IOA3	IOA2	IOA1	IOA0	TGRA_0 的功能	TIOC0A 引脚的功能
0	0	0	0	输出比较寄存器	保持输出 *
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	0	0	0	输入捕捉寄存器	在上升沿进行输入捕捉
1	0	0	1		在下降沿进行输入捕捉
1	0	1	x		在双边沿进行输入捕捉
1	1	x	x		捕捉输入源是通道 1/ 计数时钟, 通过 TCNT_1 的递增计数 / 递减计数进行输入捕捉

【符号说明】 x: Don't care

【注】 \* 从上电复位后到设定 TIOR 为止, 输出 0。

表 11.21 TIORL\_0 (通道 0)

bit3	bit2	bit1	bit0	说 明	
IOC3	IOC2	IOC1	IOC0	TGRC_0 的功能	TIOC0C 引脚的功能
0	0	0	0	输出比较寄存器 *2	保持输出 *1
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	0	0	0	输入捕捉寄存器 *2	在上升沿进行输入捕捉
1	0	0	1		在下降沿进行输入捕捉
1	0	1	x		在双边沿进行输入捕捉
1	1	x	x		捕捉输入源是通道 1/ 计数时钟, 通过 TCNT_1 的递增计数 / 递减计数进行输入捕捉

【符号说明】 x: Don't care

【注】 \*1 从上电复位后到设定 TIOR 为止, 输出 0。

\*2 如果将 TMDR\_0 的 BFA 位设定为 1, TGRC\_0 用作缓冲寄存器, 本设定就无效并且不发生输入捕捉 / 输出比较。



表 11.22 TIOR\_1 (通道 1)

bit3	bit2	bit1	bit0	说 明	
IOA3	IOA2	IOA1	IOA0	TGRA_1 的功能	TIOC1A 引脚的功能
0	0	0	0	输出比较寄存器	保持输出 *
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	0	0	0		输入捕捉寄存器
1	0	0	1	在下降沿进行输入捕捉	
1	0	1	x	在双边沿进行输入捕捉	
1	1	x	x	通过发生 TGRA_0 的比较匹配 / 输入捕捉进行输入捕捉	

【符号说明】 x: Don't care

【注】 \* 从上电复位后到设定 TIOR 为止, 输出 0。

表 11.23 TIOR\_2 (通道 2)

bit3	bit2	bit1	bit0	说 明	
IOA3	IOA2	IOA1	IOA0	TGRA_2 的功能	TIOC2A 引脚的功能
0	0	0	0	输出比较寄存器	保持输出 *
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	x	0	0	输入捕捉寄存器	在上升沿进行输入捕捉
1	x	0	1		在下降沿进行输入捕捉
1	x	1	x		在双边沿进行输入捕捉

【符号说明】 x: Don't care

【注】 \* 从上电复位后到设定 TIOR 为止, 输出 0。

表 11.24 TIORH\_3 (通道 3)

bit3	bit2	bit1	bit0	说 明	
IOA3	IOA2	IOA1	IOA0	TGRA_3 的功能	TIOC3A 引脚的功能
0	0	0	0	输出比较寄存器	保持输出 *
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	x	0	0	输入捕捉寄存器	在上升沿进行输入捕捉
1	x	0	1		在下降沿进行输入捕捉
1	x	1	x		在双边沿进行输入捕捉

【符号说明】 x: Don't care

【注】 \* 从上电复位后到设定 TIOR 为止, 输出 0。

表 11.25 TIORL\_3 (通道 3)

bit3	bit2	bit1	bit0	说 明	
IOC3	IOC2	IOC1	IOC0	TGRC_3 的引脚	TIOC3C 引脚的功能
0	0	0	0	输出比较寄存器 *2	保持输出 *1
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	x	0	0		输入捕捉寄存器 *2
1	x	0	1	在下降沿进行输入捕捉	
1	x	1	x	在双边沿进行输入捕捉	

【符号说明】 x: Don't care

【注】 \*1 从上电复位后到设定 TIOR 为止, 输出 0。

\*2 如果将 TMDR\_3 的 BFA 位设定为 1, TGRC\_3 用作缓冲寄存器, 本设定就无效并且不发生输入捕捉 / 输出比较。

表 11.26 TIORH\_4 (通道 4)

bit3	bit2	bit1	bit0	说 明	
IOA3	IOA2	IOA1	IOA0	TGRA_4 的功能	TIOC4A 引脚的功能
0	0	0	0	输出比较寄存器	保持输出 *
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	x	0	0	输入捕捉寄存器	在上升沿进行输入捕捉
1	x	0	1		在下降沿进行输入捕捉
1	x	1	x		在双边沿进行输入捕捉

【符号说明】 x: Don't care

【注】 \* 从上电复位后到设定 TIOR 为止, 输出 0。

表 11.27 TIORL\_4 (通道 4)

bit3	bit2	bit1	bit0	说 明	
IOC3	IOC2	IOC1	IOC0	TGRC_4 的功能	TIOC4C 引脚的功能
0	0	0	0	输出比较寄存器 *2	保持输出 *1
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	x	0	0		输入捕捉寄存器 *2
1	x	0	1	在下降沿进行输入捕捉	
1	x	1	x	在双边沿进行输入捕捉	

【符号说明】 x: Don't care

【注】 \*1 从上电复位后到设定 TIOR 为止, 输出 0。

\*2 如果将 TMDR\_4 的 BFA 位设定为 1, TGRC\_4 用作缓冲寄存器, 本设定就无效并且不发生输入捕捉 / 输出比较。

表 11.28 TIORU\_5、TIORV\_5、TIORW\_5 (通道 5)

bit4	bit3	bit2	bit1	bit0	说 明	
IOC4	IOC3	IOC2	IOC1	IOC0	TGRU_5、TGRV_5、 TGRW_5 的功能	TIC5U、TIC5V、TIC5W 引脚的功能
0	0	0	0	0	比较匹配寄存器	比较匹配
0	0	0	0	1		禁止设定
0	0	0	1	x		禁止设定
0	0	1	x	x		禁止设定
0	1	x	x	x		禁止设定
1	0	0	0	0	输入捕捉寄存器	禁止设定
1	0	0	0	1		在上升沿进行输入捕捉
1	0	0	1	0		在下降沿进行输入捕捉
1	0	0	1	1		在双边沿进行输入捕捉
1	0	1	x	x		禁止设定
1	1	0	0	0		禁止设定
1	1	0	0	1		用于测量外部输入信号 Low 电平脉宽 在互补 PWM 模式的波谷进行捕捉
1	1	0	1	0		用于测量外部输入信号 Low 电平脉宽 在互补 PWM 模式的波峰进行捕捉
1	1	0	1	1		用于测量外部输入信号 Low 电平脉宽 在互补 PWM 模式的波峰和波谷进行捕捉
1	1	1	0	0		禁止设定
1	1	1	0	1		用于测量外部输入信号 High 电平脉宽 在互补 PWM 模式的波谷进行捕捉
1	1	1	1	0		用于测量外部输入信号 High 电平脉宽 在互补 PWM 模式的波峰进行捕捉
1	1	1	1	1		用于测量外部输入信号 High 电平脉宽 在互补 PWM 模式的波峰和波谷进行捕捉

【符号说明】 x: Don't care

## 11.3.4 定时器的比较匹配清除寄存器 (TCNTCMPCLR)

TCNTCMPCLR 是 8 位可读写寄存器，能设定 TCNTU\_5、TCNTV\_5 和 TCNTW\_5 的清除请求。在 MTU2 中，通道 5 有 1 个 TCNTCMPCLR。

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	CMP CLR5U	CMP CLR5V	CMP CLR5W
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7 ~ 3	—	全 0	R	保留位 读写的值总是 0。
2	CMPCLR5U	0	R/W	TCNT 比较清除 5U 允许或者禁止因 TGRU_5 的比较匹配 / 输入捕捉而产生的 TCNTU_5 的清除请求。 0: 禁止通过 TCNTU_5 和 TGRU_5 的比较匹配 / 输入捕捉将 TCNTU_5 清 H'0000 1: 允许通过 TCNTU_5 和 TGRU_5 的比较匹配 / 输入捕捉将 TCNTU_5 清 H'0000
1	CMPCLR5V	0	R/W	TCNT 比较清除 5V 允许或者禁止因 TGRV_5 的比较匹配 / 输入捕捉而产生的 TCNTV_5 的清除请求。 0: 禁止通过 TCNTV_5 和 TGRV_5 的比较匹配 / 输入捕捉将 TCNTV_5 清 H'0000 1: 允许通过 TCNTV_5 和 TGRV_5 的比较匹配 / 输入捕捉将 TCNTV_5 清 H'0000
0	CMPCLR5W	0	R/W	TCNT 比较清除 5W 允许或者禁止因 TCNTW_5 和 TGRW_5 的比较匹配 / 输入捕捉而产生的 TCNTW_5 的清除请求。 0: 禁止通过 TCNTW_5 和 TGRW_5 的比较匹配 / 输入捕捉将 TCNTW_5 清 H'0000 1: 允许通过 TCNTW_5 和 TGRW_5 的比较匹配 / 输入捕捉将 TCNTW_5 的清 H'0000



### 11.3.5 定时器的中断允许寄存器 (TIER)

TIER 是 8 位可读写寄存器，允许或者禁止各通道的中断请求。在 MTU2 中，通道 0 有 2 个，通道 1 ~ 5 各有 1 个，共计 7 个 TIER。

- TIER\_0、TIER\_1、TIER\_2、TIER\_3、TIER\_4

位:	7	6	5	4	3	2	1	0
	TTGE	TTGE2	TCIEU	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7	TTGE	0	R/W	A/D 转换开始请求的允许 允许或者禁止因 TGRA 的输入捕捉 / 比较匹配产生的 A/D 转换器开始请求。 0: 禁止 A/D 转换开始请求的发生 1: 允许 A/D 转换开始请求的发生
6	TTGE2	0	R/W	A/D 转换开始请求的允许 2 在互补 PWM 模式中，允许或者禁止因 TCNT_4 的下溢（波谷）而产生的 A/D 转换请求。 在通道 0 ~ 3 中，此位为保留位，读写的值总是 0。 0: 禁止因 TCNT_4 的下溢（波谷）产生的 A/D 转换请求 1: 允许因 TCNT_4 的下溢（波谷）产生的 A/D 转换请求
5	TCIEU	0	R/W	下溢的中断允许 在通道 1、2 中，当 TSR 的 TCFU 标志被置 1 时，允许或者禁止 TCFU 标志的中断请求（TCIU）。 在通道 0、3、4 中，此位为保留位，读写的值总是 0。 0: 禁止 TCFU 的中断请求（TCIU） 1: 允许 TCFU 的中断请求（TCIU）
4	TCIEV	0	R/W	上溢的中断允许 当 TSR 的 TCFV 标志被置 1 时，允许或者禁止 TCFV 标志的中断请求（TCIV）。 0: 禁止 TCFV 的中断请求（TCIV） 1: 允许 TCFV 的中断请求（TCIV）
3	TGIED	0	R/W	TGR 中断允许 D 在通道 0、3、4 中，当 TSR 的 TGFD 位被置 1 时，允许或者禁止 TGFD 位的中断请求（TGID）。 在通道 1、2 中，此位为保留位，读写的值总是 0。 0: 禁止 TGFD 位的中断请求（TGID） 1: 允许 TGFD 位的中断请求（TGID）
2	TGIEC	0	R/W	TGR 中断允许 C 在通道 0、3、4 中，当 TSR 的 TGFC 位被置 1 时，允许或者禁止 TGFC 位的中断请求（TGIC）。 在通道 1、2 中，此位为保留位，读写的值总是 0。 0: 禁止 TGFC 位的中断请求（TGIC） 1: 允许 TGFC 位的中断请求（TGIC）

位	位名	初始值	R/W	说 明
1	TGIEB	0	R/W	TGR 中断允许 B 当 TSR 的 TGFB 位被置 1 时，允许或者禁止 TGFB 位的中断请求 (TGIB)。 0: 禁止 TGFB 位的中断请求 (TGIB) 1: 允许 TGFB 位的中断请求 (TGIB)
0	TGIEA	0	R/W	TGR 中断允许 A 当 TSR 的 TGFA 位被置 1 时，允许或者禁止 TGFA 位的中断请求 (TGIA)。 0: 禁止 TGFA 位的中断请求 (TGIA) 1: 允许 TGFA 位的中断请求 (TGIA)

• TIER2\_0

位:	7	6	5	4	3	2	1	0
	TTGE2	-	-	-	-	-	TGIEF	TGIEE
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
7	TTGE2	0	R/W	A/D 转换开始请求的允许 2 允许或者禁止因 TCNT_0 和 TGRE_0 的比较匹配而产生的 A/D 转换开始请求。 0: 禁止因 TCNT_0 和 TGRE_0 的比较匹配而产生的 A/D 转换开始请求 1: 允许因 TCNT_0 和 TGRE_0 的比较匹配而产生的 A/D 转换开始请求
6 ~ 2	—	全 0	R	保留位 读写的值总是 0。
1	TGIEF	0	R/W	TGR 中断允许 F 允许或者禁止因 TCNT_0 和 TGRF_0 的比较匹配而产生的中断请求。 0: 禁止 TGFE 位的中断请求 (TGIF) 1: 允许 TGFE 位的中断请求 (TGIF)
0	TGIEE	0	R/W	TGR 中断允许 E 允许或者禁止因 TCNT_0 和 TGRE_0 的比较匹配而产生的中断请求。 0: 禁止 TGEE 位的中断请求 (TGIE) 1: 允许 TGEE 位的中断请求 (TGIE)

## • TIER\_5

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	TGIE5U	TGIE5V	TGIE5W
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7 ~ 3	—	全 0	R	保留位 读写的值总是 0。
2	TGIE5U	0	R/W	TGR 中断允许 5U TSR_5 的 CMFU5 位被置 1 时，允许或禁止由 CMFU5 位引起的中断请求 (TGIU_5)。 0: 禁止 TGIU_5 的中断请求 1: 允许 TGIU_5 的中断请求
1	TGIE5V	0	R/W	TGR 中断允许 5V TSR_5 的 CMFV5 位被置 1 时，允许或禁止由 CMFV5 位引起的中断请求 (TGIV_5)。 0: 禁止 TGIV_5 的中断请求 1: 允许 TGIV_5 的中断请求
0	TGIE5W	0	R/W	TGR 中断允许 5W TSR_5 的 CMFW5 位置被 1 时，允许或禁止由 CMFW5 位引起的中断请求 (TGIW_5)。 0: 禁止 TGIW_5 的中断请求 1: 允许 TGIW_5 的中断请求

### 11.3.6 定时器的状态寄存器 (TSR)

TSR 是 8 位可读写寄存器，表示各通道的状态。在 MTU2 中，通道 0 有 2 个，通道 1 ~ 5 各有 1 个，共计 7 个 TSR。

- TSR\_0、TSR\_1、TSR\_2、TSR\_3、TSR\_4

位:	7	6	5	4	3	2	1	0
	TCFD	-	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA
初始值:	1	1	0	0	0	0	0	0
R/W:	R	R	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1

【注】\*1 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说 明
7	TCFD	1	R	计数方向标志 是表示通道 1 ~ 4 的 TCNT 计数方向的状态标志。 在通道 0 中，此位为保留位，读写的值总是 1。 0: TCNT 进行递减计数 1: TCNT 进行递增计数
6	—	1	R	保留位 读写的值总是 1。
5	TCFU	0	R/(W)*1	下溢标志 是表示在通道 1、2 为相位计数模式时 TCNT 下溢的状态标志。为了清除标志，只能写 0。 在通道 0、3、4 中，此位为保留位，读写的值总是 0。 [置位条件] • 当 TCNT 的值发生下溢 (H'0000→H'FFFF) 时 [清除条件] • 在 TCFU=1 的状态下，当读 TCFU 后，给 TCFU 写 0 时 *2
4	TCFV	0	R/(W)*1	上溢标志 是表示 TCNT 上溢的状态标志。为了清除标志，只能写 0。 [置位条件] • 当 TCNT 的值发生上溢 (H'FFFF→H'0000) 时 对于通道 4，在互补 PWM 模式中，当 TCNT_4 的值发生下溢 (H'0001→H'0000) 时，本标志也被置位。 [清除条件] • 在 TCFV=1 的状态下，当读 TCFV 后，给 TCFV 写 0 时 *2 对于通道 4，在通过 TCIV 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位为 0 时，本标志清 0。

位	位名	初始值	R/W	说 明
3	TGFD	0	R/(W)*1	<p>输入捕捉 / 输出比较标志 D</p> <p>是表示通道 0、3、4 的 TGRD 输入捕捉或者比较匹配的状态标志。为了清除标志，只能写 0。在通道 1、2 中，此位为保留位，读写的值总是 0。</p> <p>[置位条件]</p> <ul style="list-style-type: none"> <li>在 TGRD 用作输出比较寄存器的情况下，当 TCNT=TGRD 时</li> <li>在 TGRD 用作输入捕捉的情况下，当通过输入捕捉信号将 TCNT 的值传送到 TGRD 时</li> </ul> <p>[清除条件]</p> <ul style="list-style-type: none"> <li>在通过 TGID 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位为 0 时</li> <li>在 TGFD=1 的状态下，当读 TGFD 后，给 TGFD 写 0 时 *2</li> </ul>
2	TGFC	0	R/(W)*1	<p>输入捕捉 / 输出比较标志 C</p> <p>是表示通道 0、3、4 的 TGRC 输入捕捉或者比较匹配的状态标志。为了清除标志，只能写 0。在通道 1、2 中，此位为保留位，读写的值总是 0。</p> <p>[置位条件]</p> <ul style="list-style-type: none"> <li>在 TGRC 用作输出比较寄存器的情况下，当 TCNT=TGRC 时</li> <li>在 TGRC 用作输入捕捉的情况下，当通过输入捕捉信号将 TCNT 的值传送到 TGRC 时</li> </ul> <p>[清除条件]</p> <ul style="list-style-type: none"> <li>在通过 TGIC 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位为 0 时</li> <li>在 TGFC=1 的状态下，当读 TGFC 后，给 TGFC 写 0 时 *2</li> </ul>
1	TGFB	0	R/(W)*1	<p>输入捕捉 / 输出比较标志 B</p> <p>是表示 TGRB 的输入捕捉或者比较匹配的状态标志。为了清除标志，只能写 0。</p> <p>[置位条件]</p> <ul style="list-style-type: none"> <li>在 TGRB 用作输出比较寄存器的情况下，当 TCNT=TGRB 时</li> <li>在 TGRB 用作输入捕捉的情况下，当通过输入捕捉信号将 TCNT 的值传送到 TGRB 时</li> </ul> <p>[清除条件]</p> <ul style="list-style-type: none"> <li>在通过 TGIB 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位为 0 时</li> <li>在 TGFB=1 的状态下，当读 TGFB 后，给 TGFB 写 0 时 *2</li> </ul>
0	TGFA	0	R/(W)*1	<p>输入捕捉 / 输出比较标志 A</p> <p>是表示 TGRA 的输入捕捉或者比较匹配的状态标志。为了清除标志，只能写 0。</p> <p>[置位条件]</p> <ul style="list-style-type: none"> <li>在 TGRA 用作输出比较寄存器的情况下，当 TCNT=TGRA 时</li> <li>在 TGRA 用作输入捕捉的情况下，当通过输入捕捉信号将 TCNT 的值传送到 TGRA</li> </ul> <p>[清除条件]</p> <ul style="list-style-type: none"> <li>在通过 TGIA 中断启动 DMAC 时</li> <li>在通过 TGIA 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位为 0 时</li> <li>在 TGFA=1 的状态下，当读 TGFA 后，给 TGFA 写 0 时 *2</li> </ul>

【注】 \*1 为了清除标志，只能在读 1 后写 0。

\*2 如果在读 1 后写 0 前发生下一个标志的置位，即使写 0 也不清除标志，所以必须重新读 1 再写 0。

## • TSR2\_0

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	TGFF	TGFE
初始值:	1	1	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/(W)*1R/(W)*1	

【注】\*1 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说 明
7、6	—	全 1	R	保留位 读写的值总是 1。
5 ~ 2	—	全 0	R	保留位 读写的值总是 0。
1	TGFF	0	R/(W)*1	比较匹配标志 F 是表示 TCNT_0 和 TGRF_0 比较匹配的状态标志。 [置位条件] • 在将 TGRF_0 用作比较寄存器的情况下，当 TCNT_0=TGRF_0 时 [清除条件] • 在 TGFF=1 的状态下，当读 TGFF 后，给 TGFF 写 0 时 *2
0	TGFE	0	R/(W)*1	比较匹配标志 E 是表示 TCNT_0 和 TGRE_0 比较匹配的状态标志。 [置位条件] • 在将 TGRE_0 用作比较寄存器的情况下，当 TCNT_0=TGFE_0 时 [清除条件] • 在 TGFE=1 的状态下，当读 TGFE 后，给 TGFE 写 0 时 *2

【注】\*1 为了清除标志，只能在读 1 后写 0。

\*2 如果在读 1 后写 0 前发生下一个比较匹配标志的置位，即使写 0 也不清除标志，所以必须重新读 1 再写 0。

• TSR\_5

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	CMFU5	CMFV5	CMFW5
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/(W)*1	R/(W)*1	R/(W)*1

【注】\*1 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说明
7 ~ 3	—	全 0	R	保留位 读写的值总是 0。
2	CMFU5	0	R/(W)*1	比较匹配 / 输入捕捉标志 U5 是表示 TGRU_5 的输入捕捉或者比较匹配的状态标志。为了清除标志，只能写 0。 [置位条件] <ul style="list-style-type: none"> <li>在 TGRU_5 用作比较匹配寄存器的情况下，当 TCNTU_5=TGRU_5 时</li> <li>在 TGRU_5 用作输入捕捉的情况下，当通过输入捕捉信号将 TCNTU_5 的值传送到 TGRU_5 时</li> <li>在 TGRU_5 用作测量外部输入信号脉宽的情况下，当将 TCNTU_5 的值传送到 TGRU_5 时 *2</li> </ul> [清除条件] <ul style="list-style-type: none"> <li>在通过 TGIU_5 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位为 0 时</li> <li>在 CMFU5=1 的状态下，当读 CMFU5 后，给 CMFU5 写 0 时</li> </ul>
1	CMFV5	0	R/(W)*1	比较匹配 / 输入捕捉标志 V5 是表示 TGRV_5 的输入捕捉或者比较匹配的状态标志。为了清除标志，只能写 0。 [置位条件] <ul style="list-style-type: none"> <li>在 TGRV_5 用作比较匹配寄存器的情况下，当 TCNTV_5=TGRV_5 时</li> <li>在 TGRV_5 用作输入捕捉的情况下，当通过输入捕捉信号将 TCNTV_5 的值传送到 TGRV_5 时</li> <li>在 TGRV_5 用作测量外部输入信号脉宽的情况下，当将 TCNTV_5 的值传送到 TGRV_5 时 *2</li> </ul> [清除条件] <ul style="list-style-type: none"> <li>在通过 TGIV_5 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位为 0 时</li> <li>在 CMFV5=1 的状态下，当读 CMFV5 后，给 CMFV5 写 0 时</li> </ul>
0	CMFW5	0	R/(W)*1	比较匹配 / 输入捕捉标志 W5 是表示 TGRW_5 的输入捕捉或者比较匹配的状态标志。 [置位条件] <ul style="list-style-type: none"> <li>在 TGRW_5 用作比较匹配寄存器的情况下，当 TCNTW_5=TGRW_5 时</li> <li>在 TGRW_5 用作输入捕捉的情况下，当通过输入捕捉信号将 TCNTW_5 的值传送到 TGRW_5 时</li> <li>在 TGRW_5 用作测量外部输入信号脉宽的情况下，将 TCNTW_5 的值传送到 TGRW_5 时 *2</li> </ul> [清除条件] <ul style="list-style-type: none"> <li>在通过 TGIW_5 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位为 0 时</li> <li>在 CMFW5=1 的状态下，当读 CMFW5 后，给 CMFW5 写 0 时</li> </ul>

【注】\*1 为了清除标志，只能在读 1 后写 0。

\*2 通过定时器的 I/O 控制寄存器 U\_5/V\_5/W\_5 (TIORU\_5/V\_5/W\_5) 的 IOC 位设定要传送的时序。

## 11.3.7 定时器的缓冲运行传送模式寄存器 (TBTM)

TBTM 是 8 位可读写寄存器，设定在 PWM 模式中缓冲寄存器到定时器的通用寄存器的传送时序。在 MTU2 中，通道 0、3、4 各有 1 个，共计 3 个 TBTM。

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	TTSE	TTSB	TTSA
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7 ~ 3	—	全 0	R	保留位 读写的值总是 0。
2	TTSE	0	R/W	时序的选择 E 设定在缓冲运行时 TGRF_0 到 TGRE_0 的传送时序。 在通道 3、4 中，此位为保留位，读写的值总是 0。另外，在 PWM 以外的模式中使用通道 0 时，不能将本位设定为 1。 0: 当发生通道 0 的比较匹配 E 时 1: 当清除 TCNT_0 时
1	TTSB	0	R/W	时序的选择 B 设定在各通道的缓冲运行时 TGRD 到 TGRB 的传送时序。另外，在 PWM 以外的模式使用的通道，不能将本位设定为 1。 0: 当各通道发生比较匹配 B 时 1: 当清除各通道的 TCNT 时
0	TTSA	0	R/W	时序的选择 A 设定在各通道的缓冲运行时 TGRC 到 TGRA 的传送时序。另外，在 PWM 以外的模式使用的通道，不能将本位设定为 1。 0: 当各通道发生比较匹配 A 时 1: 当清除各通道的 TCNT 时



## 11.3.8 定时器的输入捕捉控制寄存器 (TICCR)

TICCR 是 8 位可读写寄存器，控制 TCNT\_1 和 TCNT\_2 级联时的输入捕捉条件。在 MTU2 中，通道 1 有 1 个 TICCR。

位:	7	6	5	4	3	2	1	0
	-	-	-	-	I2BE	I2AE	I1BE	I1AE
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7 ~ 4	—	全 0	R	保留位 读写的值总是 0。
3	I2BE	0	R/W	输入捕捉的允许 选择是否将 TIOC2B 引脚追加到 TGRB_1 的输入捕捉条件。 0: 不将 TIOC2B 引脚追加到 TGRB_1 的输入捕捉条件 1: 将 TIOC2B 引脚追加到 TGRB_1 的输入捕捉条件
2	I2AE	0	R/W	输入捕捉的允许 选择是否将 TIOC2A 引脚追加到 TGRA_1 的输入捕捉条件。 0: 不将 TIOC2A 引脚追加到 TGRA_1 的输入捕捉条件 1: 将 TIOC2A 引脚追加到 TGRA_1 的输入捕捉条件
1	I1BE	0	R/W	输入捕捉的允许 选择是否将 TIOC1B 引脚追加到 TGRB_2 的输入捕捉条件。 0: 不将 TIOC1B 引脚追加到 TGRB_2 的输入捕捉条件 1: 将 TIOC1B 引脚追加到 TGRB_2 的输入捕捉条件
0	I1AE	0	R/W	输入捕捉的允许 选择是否将 TIOC1A 引脚追加到 TGRA_2 的输入捕捉条件。 0: 不将 TIOC1A 引脚追加到 TGRA_2 的输入捕捉条件 1: 将 TIOC1A 引脚追加到 TGRA_2 的输入捕捉条件

## 11.3.9 定时器的同步清除寄存器 (TSYCR)

TSYCR 是 8 位可读写寄存器，设定从 MTU2 进行 MTU2S 的 TCNT\_3 和 TCNT\_4 的同步清除条件。MTU2S 的通道 3 有 1 个 TSYCR，但是 MTU2 没有 TSYCR。

位:	7	6	5	4	3	2	1	0
	CE0A	CE0B	CE0C	CE0D	CE1A	CE1B	CE2A	CE2B
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7	CE0A	0	R/W	清除的允许 0A 设定是禁止还是允许由 MTU2 的 TSR_0 的 TGFA 标志置位进行的清除。 0: 禁止由 TSR_0 的 TGFA 标志置位进行的清除 1: 允许由 TSR_0 的 TGFA 标志置位进行的清除
6	CE0B	0	R/W	清除的允许 0B 设定是禁止还是允许由 MTU2 的 TSR_0 的 TGFB 标志置位进行的清除。 0: 禁止由 TSR_0 的 TGFB 标志置位进行的清除 1: 允许由 TSR_0 的 TGFB 标志置位进行的清除
5	CE0C	0	R/W	清除的允许 0C 设定是禁止还是允许由 MTU2 的 TSR_0 的 TGFC 标志置位进行的清除。 0: 禁止由 TSR_0 的 TGFC 标志置位进行的清除 1: 允许由 TSR_0 的 TGFC 标志置位进行的清除
4	CE0D	0	R/W	清除的允许 0D 设定是禁止还是允许由 MTU2 的 TSR_0 的 TGFD 标志置位进行的清除。 0: 禁止由 TSR_0 的 TGFD 标志置位进行的清除 1: 允许由 TSR_0 的 TGFD 标志置位进行的清除
3	CE1A	0	R/W	清除的允许 1A 设定是禁止还是允许由 MTU2 的 TSR_1 的 TGFA 标志置位进行的清除。 0: 禁止由 TSR_1 的 TGFA 标志置位进行的清除 1: 允许由 TSR_1 的 TGFA 标志置位进行的清除
2	CE1B	0	R/W	清除的允许 1B 设定是禁止还是允许由 MTU2 的 TSR_1 的 TGFB 标志置位进行的清除。 0: 禁止由 TSR_1 的 TGFB 标志置位进行的清除 1: 允许由 TSR_1 的 TGFB 标志置位进行的清除
1	CE2A	0	R/W	清除的允许 2A 设定是禁止还是允许由 MTU2 的 TSR_2 的 TGFA 标志置位进行的清除。 0: 禁止由 TSR_2 的 TGFA 标志置位进行的清除 1: 允许由 TSR_2 的 TGFA 标志置位进行的清除
0	CE2B	0	R/W	清除的允许 2B 设定是禁止还是允许由 MTU2 的 TSR_2 的 TGFB 标志置位进行的清除。 0: 禁止由 TSR_2 的 TGFB 标志置位进行的清除 1: 允许由 TSR_2 的 TGFB 标志置位进行的清除

## 11.3.10 定时器的 A/D 转换开始请求的控制寄存器 (TADCR)

TADCR 是 16 位可读写寄存器，允许或者禁止 A/D 转换开始请求以及设定 A/D 转换开始请求是否与跳过中断功能联动。在 MTU2 中，通道 4 有 1 个 TADCR。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BF[1:0]		-	-	-	-	-	-	UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
初始值:	0	0	0	0	0	0	0	0	0	0*	0	0*	0*	0*	0*	0*
R/W:	R/W	R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

【注】\* 除了互补PWM模式以外，不能置1。

位	位名	初始值	R/W	说明
15、14	BF[1:0]	00	R/W	TADCOBRA/B_4 传送时序的选择 选择 TADCOBRA/B_4 到 TADCORA/B_4 的传送时序。 详细内容请参照表 11.29。
13 ~ 8	—	全 0	R	保留位 读写的值总是 0。
7	UT4AE	0	R/W	递增计数 TRG4AN 的允许 在进行 TCNT_4 的递增计数时，允许或者禁止 A/D 转换的开始请求 (TRG4AN)。 0: 在进行 TCNT_4 的递增计数时，禁止 A/D 转换的开始请求 (TRG4AN) 1: 在进行 TCNT_4 的递增计数时，允许 A/D 转换的开始请求 (TRG4AN)
6	DT4AE	0*	R/W	递减计数 TRG4AN 的允许 在进行 TCNT_4 的递减计数时，允许或者禁止 A/D 转换的开始请求 (TRG4AN)。 0: 在进行 TCNT_4 的递减计数时，禁止 A/D 转换的开始请求 (TRG4AN) 1: 在进行 TCNT_4 的递减计数时，允许 A/D 转换的开始请求 (TRG4AN)
5	UT4BE	0	R/W	递增计数 TRG4BN 的允许 在进行 TCNT_4 的递增计数时，允许或者禁止 A/D 转换的开始请求 (TRG4BN)。 0: 在进行 TCNT_4 的递增计数时，禁止 A/D 转换的开始请求 (TRG4BN) 1: 在进行 TCNT_4 的递增计数时，允许 A/D 转换的开始请求 (TRG4BN)
4	DT4BE	0*	R/W	递减计数 TRG4BN 的允许 在进行 TCNT_4 的递减计数时，允许或者禁止 A/D 转换的开始请求 (TRG4BN)。 0: 在进行 TCNT_4 的递减计数时，禁止 A/D 转换的开始请求 (TRG4BN) 1: 在进行 TCNT_4 的递减计数时，允许 A/D 转换的开始请求 (TRG4BN)

位	位名	初始值	R/W	说 明
3	ITA3AE	0*	R/W	允许 TGIA_3 跳过中断功能的联动 选择 A/D 转换的开始请求 (TRG4AN) 是否与 TGIA_3 跳过中断功能联动。 0: 不与 TGIA_3 跳过中断功能联动 1: 与 TGIA_3 跳过中断功能联动
2	ITA4VE	0*	R/W	允许 TCIV_4 跳过中断功能的联动 选择 A/D 转换的开始请求 (TRG4AN) 是否与 TCIV_4 跳过中断功能联动。 0: 不与 TCIV_4 跳过中断功能联动 1: 与 TCIV_4 跳过中断功能联动
1	ITB3AE	0*	R/W	允许 TGIA_3 跳过中断功能的联动 选择 A/D 转换的开始请求 (TRG4BN) 是否与 TGIA_3 跳过中断功能联动。 0: 不与 TGIA_3 跳过中断功能联动 1: 与 TGIA_3 跳过中断功能联动
0	ITB4VE	0*	R/W	允许 TCIV_4 跳过中断功能的联动 选择 A/D 转换的开始请求 (TRG4BN) 是否与 TCIV_4 跳过中断功能联动。 0: 不与 TCIV_4 跳过中断功能联动 1: 与 TCIV_4 跳过中断功能联动

- 【注】 1. 禁止以 8 位为单位存取 TADCR，必须以 16 位为单位进行存取。
2. 在禁止跳过中断功能时（将定时器的跳过中断设定寄存器 (TITCR) 的 T3AEN、T4VEN 位或者 TITCR 的跳过次数设定位 (3ACOR 和 4VCOR) 设定为 0 时），必须设定为不与跳过中断功能联动（将定时器的 A/D 转换开始请求控制寄存器 (TADCR) 的 ITA3AE、ITA4VE、ITB3AE、ITB4VE 位设定为 0）。
3. 在禁止跳过中断功能时，如果设定为与跳过中断功能联动，就不进行 A/D 转换的开始请求。
- \* 除了互补 PWM 模式以外，不能置 1。

表 11.29 通过 BF1、BF0 位设定的传送时序

bit7	bit6	说 明
BF1	BF0	
0	0	不将周期设定缓冲寄存器的值传送到周期设定寄存器
0	1	在 TCNT_4 的波峰，将周期设定缓冲寄存器的值传送到周期设定寄存器 *1
1	0	在 TCNT_4 的波谷，将周期设定缓冲寄存器的值传送到周期设定寄存器 *2
1	1	在 TCNT_4 的波峰和波谷，将周期设定缓冲寄存器的值传送到周期设定寄存器 *2

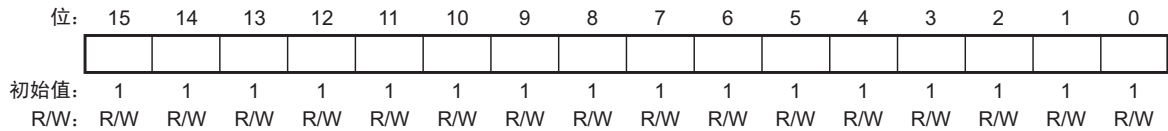
- 【注】 \*1 在互补 PWM 模式中 TCNT\_4 处于波谷时，或者在复位同步 PWM 模式中 TCNT\_3 和 TGRA\_3 比较匹配时，或者在 PWM 模式 1/ 通常运行模式中 TCNT\_4 和 TGRA\_4 比较匹配时，将周期设定缓冲寄存器的值传送到周期设定寄存器。

\*2 除了互补 PWM 模式以外，禁止设定。

### 11.3.11 定时器的 A/D 转换开始请求的周期设定寄存器 (TADCORA/B\_4)

TADCORA/B\_4 是 16 位可读写寄存器。当 TADCORA/B\_4 和 TCNT\_4 一致时, 就发生对应的 A/D 转换开始请求。

TADCORA/B\_4 的初始值为 H'FFFF。

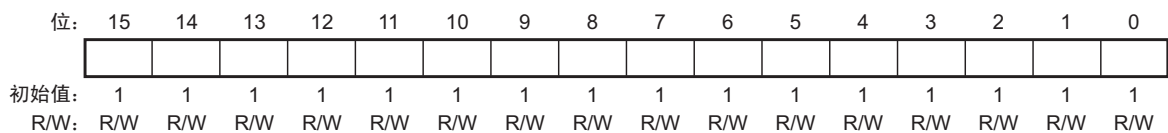


【注】 禁止以8位为单位存取TADCORA/B\_4, 必须以16位为单位进行存取。

### 11.3.12 定时器的 A/D 转换开始请求的周期设定缓冲寄存器 (TADCOBRA/B\_4)

TADCOBRA/B\_4 是 16 位可读写寄存器。在波峰或者波谷将 TADCORA/B\_4 的缓冲寄存器的值传送到 TADCORA/B\_4。

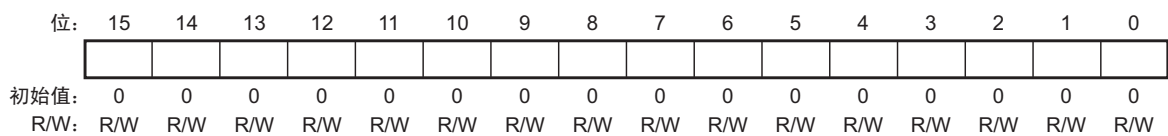
TADCOBRA/B\_4 的初始值为 H'FFFF。



【注】 禁止以8位为单位存取TADCOBRA/B\_4, 必须以16位为单位进行存取。

### 11.3.13 定时器的计数器 (TCNT)

TCNT 是 16 位可读写计数器。通道 0 ~ 4 各有 1 个, 通道 5 有 3 个 TCNTU/V/W\_5, 共计 8 个 TCNT。TCNT 在复位时被初始化为 H'0000。



【注】 禁止以8位为单位存取TCNT, 必须以16位为单位进行存取。

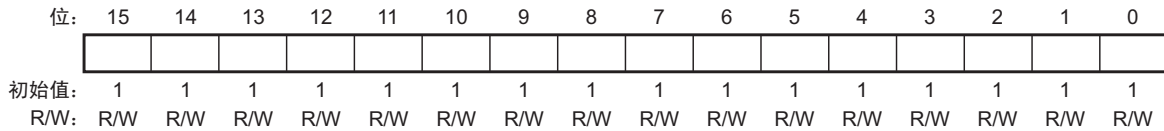
### 11.3.14 定时器的通用寄存器 (TGR)

TGR 是 16 位可读写寄存器。通道 0 有 6 个，通道 1、2 各有 2 个，通道 3、4 各有 4 个，通道 5 有 3 个，共计 21 个通用寄存器。

TGRA、TGRB、TGRC 和 TGRD 是输出比较 / 输入捕捉兼用的寄存器。能将通道 0、3、4 的 TGRC 和 TGRD 作为缓冲寄存器。TGR 和缓冲寄存器的组合为 TGRA-TGRC 和 TGRB-TGRD。

TGRE\_0 和 TGRF\_0 用作比较寄存器，当 TCNT\_0 和 TGRE\_0 一致时，就会发生 A/D 转换开始请求。能将 TGRF 作为缓冲寄存器。TGR 和缓冲寄存器的组合为 TGRE-TGRF。

TGRU\_5、TGRV\_5 和 TGRW\_5 是比较匹配 / 输入捕捉 / 测量外部脉宽兼用的寄存器。



【注】 禁止以8位为单位存取TGR，必须以16位为单位进行存取。TGR的初始值是H'FFFF。

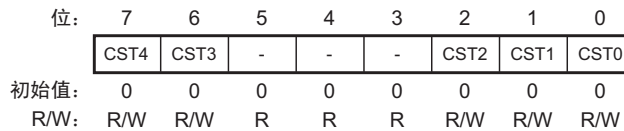
### 11.3.15 定时器的开始寄存器 (TSTR)

TSTR 是 8 位可读写寄存器，选择通道 0 ~ 4 的 TCNT 的运行 / 停止。

TSTR\_5 是 8 位可读写寄存器，选择通道 5 的 TCNTU/V/W\_5 的运行 / 停止。

在对 TMDR 设定运行模式时或者对 TCR 设定 TCNT 的计数时钟时，必须停止 TCNT 的计数器运行。

- TSTR



位	位名	初始值	R/W	说 明
7	CST4	0	R/W	计数器的开始 4、3
6	CST3	0	R/W	选择 TCNT 的运行或者停止。 在 TIOC 引脚为输出的状态下，如果在运行中给 CST 位写 0，就停止计数器并且保持 TIOC 引脚的输出比较的输出电平。如果在 CST 位为 0 的状态下写 TIOR，引脚的输出电平就被更新为已设定的初始输出值。 0: 停止 TCNT_4、TCNT_3 的计数 1: TCNT_4、TCNT_3 计数
5 ~ 3	—	全 0	R	保留位 读写的值总是 0。
2	CST2	0	R/W	计数器的开始 2 ~ 0
1	CST1	0	R/W	选择 TCNT 的运行或者停止。
0	CST0	0	R/W	在 TIOC 引脚为输出的状态下，如果在运行中给 CST 位写 0，就停止计数器并且保持 TIOC 引脚的输出比较的输出电平。如果在 CST 位为 0 的状态下写 TIOR，引脚的输出电平就被更新为已设定的初始输出值。 0: 停止 TCNT_2 ~ TCNT_0 的计数 1: TCNT_2 ~ TCNT_0 计数

## • TSTR\_5

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	CSTU5	CSTV5	CSTW5
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7 ~ 3	—	全 0	R	保留位 读写的值总是 0。
2	CSTU5	0	R/W	计数器的开始 U5 选择 TCNTU_5 的运行或者停止。 0: 停止 TCNTU_5 的计数 1: TCNTU_5 计数
1	CSTV5	0	R/W	计数器的开始 V5 选择 TCNTV_5 的运行或者停止。 0: 停止 TCNTV_5 的计数 1: TCNTV_5 计数
0	CSTW5	0	R/W	计数器的开始 W5 选择 TCNTW_5 的运行或者停止。 0: 停止 TCNTW_5 的计数 1: TCNTW_5 计数

## 11.3.16 定时器的同步寄存器 (TSYR)

TSYR 是 8 位可读写寄存器，选择通道 0 ~ 4 的 TCNT 独立运行或者同步运行。对应位设定为 1 的通道进行同步运行。

位:	7	6	5	4	3	2	1	0
	SYNC4	SYNC3	-	-	-	SYNC2	SYNC1	SYNC0
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7	SYNC4	0	R/W	定时器的同步 4、3
6	SYNC3	0	R/W	选择和其他通道的独立运行或者同步运行。 当选择同步运行时，能对多个 TCNT 进行同步预置，并且能通过其他通道的计数器清除进行同步清除。 要设定同步运行时，至少需要将 2 个通道的 SYNC 位设定为 1。要设定同步清除时，除了 SYNC 位以外，还需要通过 TCR 的 CCLR2 ~ CCLR0 位设定 TCNT 的清除源。 0: TCNT_4、TCNT_3 进行独立运行 (TCNT 的预置 / 清除与其他通道无关) 1: TCNT_4、TCNT_3 进行同步运行能进行 TCNT 的同步预置 / 同步清除
5 ~ 3	—	全 0	R	保留位 读写的值总是 0。
2	SYNC2	0	R/W	定时器的同步 2 ~ 0
1	SYNC1	0	R/W	选择和其他通道的独立运行或者同步运行。
0	SYNC0	0	R/W	当选择同步运行时，能对多个 TCNT 进行同步预置，并且能通过其他通道的计数器清除进行同步清除。 要设定同步运行时，至少需要将 2 个通道的 SYNC 位设定为 1。要设定同步清除时，除了 SYNC 位以外，还需要通过 TCR 的 CCLR2 ~ CCLR0 位设定 TCNT 的清除源。 0: TCNT_2 ~ TCNT_0 进行独立运行 (TCNT 的预置 / 清除与其他通道无关) 1: TCNT_2 ~ TCNT_0 进行同步运行能进行 TCNT 的同步预置 / 同步清除



## 11.3.17 定时器的计数器同步开始寄存器 (TCSYSTR)

TCSYSTR 是 8 位可读写寄存器，进行 MTU2 和 MTU2S 计数器的同步开始，但是 MTU2S 没有 TCSYSTR。

位:	7	6	5	4	3	2	1	0
	SCH0	SCH1	SCH2	SCH3	SCH4	-	SCH3S	SCH4S
初始值:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R/(W)*	R/(W)*

【注】\* 为了对寄存器置位，只能写1。

位	位名	初始值	R/W	说 明
7	SCH0	0	R/(W)*	同步开始 控制 MTU2 的 TCNT_0 的同步开始。 0: 不同步开始 MTU2 的 TCNT_0 1: 同步开始 MTU2 的 TCNT_0 [清除条件] • SCH0=1 的状态下，当给 MTU2 的 TSTR 的 CST0 位设定 1 时
6	SCH1	0	R/(W)*	同步开始 控制 MTU2 的 TCNT_1 的同步开始。 0: 不同步开始 MTU2 的 TCNT_1 1: 同步开始 MTU2 的 TCNT_1 [清除条件] • 在 SCH1=1 的状态下，当给 MTU2 的 TSTR 的 CST1 位设定 1 时
5	SCH2	0	R/(W)*	同步开始 控制 MTU2 的 TCNT_2 的同步开始。 0: 不同步开始 MTU2 的 TCNT_2 1: 同步开始 MTU2 的 TCNT_2 [清除条件] • SCH2=1 的状态下，当给 MTU2 的 TSTR 的 CST2 位设定 1 时
4	SCH3	0	R/(W)*	同步开始 控制 MTU2 的 TCNT_3 的同步开始。 0: 不同步开始 MTU2 的 TCNT_3 1: 同步开始 MTU2 的 TCNT_3 [清除条件] • SCH3=1 的状态下，当给 MTU2 的 TSTR 的 CST3 位设定 1 时
3	SCH4	0	R/(W)*	同步开始 控制 MTU2 的 TCNT_4 的同步开始。 0: 不同步开始 MTU2 的 TCNT_4 1: 同步开始 MTU2 的 TCNT_4 [清除条件] • 在 SCH4=1 的状态下，当给 MTU2 的 TSTR 的 CST4 位设定 1 时
2	—	0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说 明
1	SCH3S	0	R/(W)*	同步开始 控制 MTU2S 的 TCNT_3S 的同步开始。 0: 不同步开始 MTU2S 的 TCNT_3S 1: 同步开始 MTU2S 的 TCNT_3S [清除条件] • 在 SCH3S=1 的状态下, 当给 MTU2S 的 TSTRS 的 CST3 位设定 1 时
0	SCH4S	0	R/(W)*	同步开始 控制 MTU2S 的 TCNT_4S 的同步开始。 0: 不同步开始 MTU2S 的 TCNT_4S 1: 同步开始 MTU2S 的 TCNT_4S [清除条件] • 在 SCH4S=1 的状态下, 当给 MTU2S 的 TSTRS 的 CST4 位设定 1 时

【注】 \* 为了将寄存器置位, 只能写 1。

### 11.3.18 定时器的读写允许寄存器 (TRWER)

TRWER 是 8 位可读写寄存器, 允许或者禁止通道 3、4 的误写防止对象寄存器 / 计数器的存取。

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	RWE
初始值:	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R/W

位	位名	初始值	R/W	说 明
7 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	RWE	1	R/W	允许读写 允许 / 禁止误写防止对象寄存器的读写。 0: 禁止寄存器的读写 1: 允许寄存器的读写 [清除条件] • RWE=1 的状态下, 当读 RWE 后, 给 RWE 写 0 时

- 误写防止的对象寄存器 / 计数器

TCR\_3、4、TMDR\_3、4、TIORH\_3、4、TIORL\_3、4、TIER\_3、4、TGRA\_3、4、TGRB\_3、4、TOER、TOCR1、TOCR2、TGCR、TCDR、TDDR 和 TCNT\_3、4, 共计 22 寄存器。

## 11.3.19 定时器的输出主控允许寄存器 (TOER)

TOER 是 8 位可读写寄存器，允许或者禁止输出引脚的 TIOC4D、TIOC4C、TIOC3D、TIOC4B、TIOC4A、TIOC3B 的输出设定。如果不设定 TOER 的各位，就不能正确地输出这些引脚。对于通道 3、4，必须在设定通道 3、4 的 TIOR 前设定 TOER 的值。

位:	7	6	5	4	3	2	1	0
	-	-	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
初始值:	1	1	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7、6	—	全 1	R	保留位 读写的值总是 1。
5	OE4D	0	R/W	主控的允许 TIOC4D 允许或者禁止 TIOC4D 引脚的 MTU2 输出。 0: 禁止 MTU2 输出 (无效电平) * 1: 允许 MTU2 输出
4	OE4C	0	R/W	主控的允许 TIOC4C 允许或者禁止 TIOC4C 引脚的 MTU2 输出。 0: 禁止 MTU2 输出 (无效电平) * 1: 允许 MTU2 输出
3	OE3D	0	R/W	主控的允许 TIOC3D 允许或者禁止 TIOC3D 引脚的 MTU2 输出。 0: 禁止 MTU2 输出 (无效电平) * 1: 允许 MTU2 输出
2	OE4B	0	R/W	主控的允许 TIOC4B 允许或者禁止 TIOC4B 引脚的 MTU2 输出。 0: 禁止 MTU2 输出 (无效电平) * 1: 允许 MTU2 输出
1	OE4A	0	R/W	主控的允许 TIOC4A 允许或者禁止 TIOC4A 引脚的 MTU2 输出。 0: 禁止 MTU2 输出 (无效电平) * 1: 允许 MTU2 输出
0	OE3B	0	R/W	主控的允许 TIOC3B 允许或者禁止 TIOC3B 引脚的 MTU2 输出。 0: 禁止 MTU2 输出 (无效电平) * 1: 允许 MTU2 输出

**【注】** \* 无效电平取决于定时器的输出控制寄存器 1/2 (TOCR1/2) 的设定。详细内容请参照“11.3.20 定时器的输出控制寄存器 1 (TOCR1)”和“11.3.21 定时器的输出控制寄存器 2 (TOCR2)”。另外，除了互补 PWM 模式 / 复位同步 PWM 模式以外，在进行 MTU2 输出时必须置 1，如果置 0 就输出低电平。

## 11.3.20 定时器的输出控制寄存器 1 (TOCR1)

TOCR1 是 8 位可读写寄存器，允许或者禁止与互补 PWM 模式 / 复位同步 PWM 模式的 PWM 周期同步的交替输出，进行 PWM 输出电平的反相控制。

位:	7	6	5	4	3	2	1	0
	-	PSYE	-	-	TOCL	TOCS	OLSN	OLSP
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R/(W)*	R/W	R/W	R/W

【注】\* 在上电复位后，只能写1次的1。写1后不能写0。

位	位名	初始值	R/W	说 明
7	—	0	R	保留位 读写的值总是 0。
6	PSYE	0	R/W	PWM 同步输出的允许 允许或者禁止与 PWM 周期同步的交替输出。 0: 禁止交替输出 1: 允许交替输出
5、4	—	全 0	R	保留位 读写的值总是 0。
3	TOCL	0	R/(W)*	TOC 寄存器的写禁止位 *1 禁止或者允许写 TOCR1 寄存器的 TOCS 位、OLSN 位、OLSP 位。 0: 允许写 TOCS 位、OLSN 位、OLSP 位 1: 禁止写 TOCS 位、OLSN 位、OLSP 位
2	TOCS	0	R/W	TOC 的选择位 对于互补 PWM 模式 / 复位同步 PWM 模式的输出电平的设定，是选择 TOCR1 的设定有效还是选择 TOCR2 的设定有效。 0: TOCR1 的设定有效 1: TOCR2 的设定有效
1	OLSN	0	R/W	输出电平的选择 N*2 在复位同步 PWM 模式 / 互补 PWM 模式中，选择反相的输出电平。请参照表 11.30。
0	OLSP	0	R/W	输出电平的选择 P*2 在复位同步 PWM 模式 / 互补 PWM 模式中，选择正相的输出电平。请参照表 11.31。

【注】 \*1 能通过将 TOCL 位设定为 1，防止 CPU 失控时的误写。

\*2 通过将 TOCS 位设定为 0，本设定有效。

表 11.30 输出电平的选择功能

bit1	功 能			
OLSN	初始输出	有效电平	比较匹配输出	
			递增计数	递减计数
0	高电平	低电平	高电平	低电平
1	低电平	高电平	低电平	高电平

【注】 反相波形的初始输出值在开始计数并经过空载时间后变为有效电平。

表 11.31 输出电平的选择功能

bit0	功 能			
OLSP	初始输出	有效电平	比较匹配输出	
			递增计数	递减计数
0	高电平	低电平	低电平	高电平
1	低电平	高电平	高电平	低电平

OLSN=1、OLSP=1 时的互补 PWM 模式的输出例 (1 相) 如图 11.2 所示。

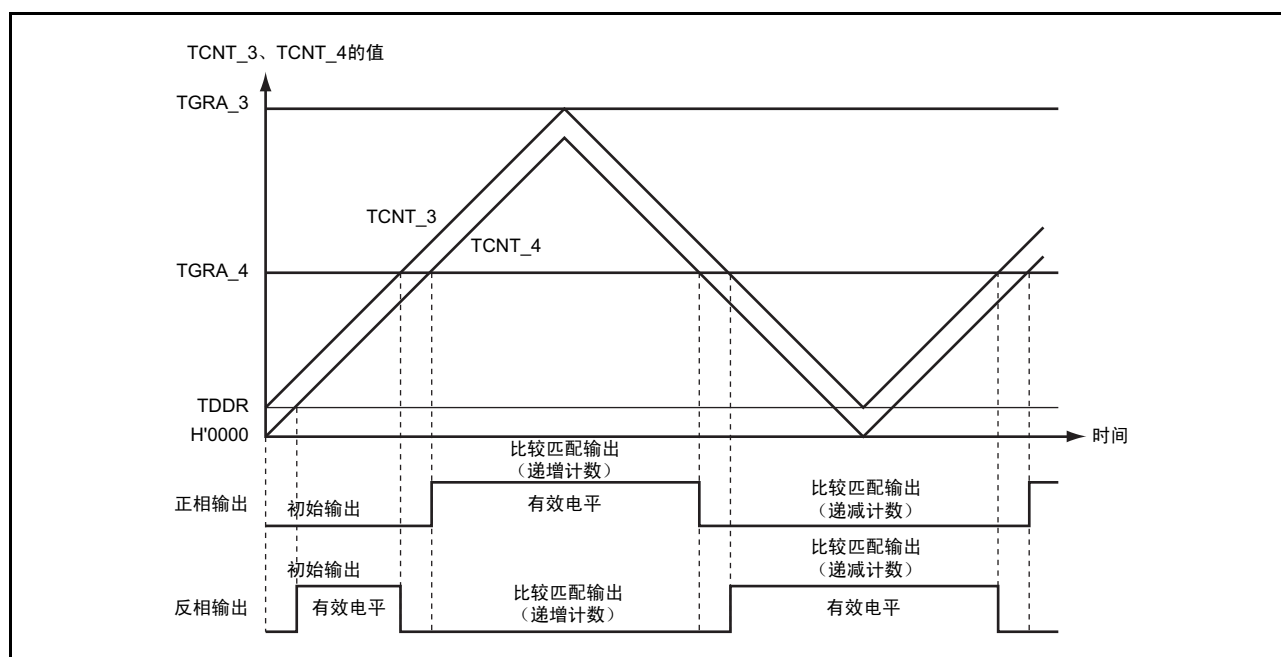


图 11.2 互补 PWM 模式的输出电平的例子

## 11.3.21 定时器的输出控制寄存器 2 (TOCR2)

TOCR2 是 8 位可读写寄存器，在互补 PWM 模式 / 复位同步 PWM 模式中进行 PWM 输出电平的反相控制。

位:	7	6	5	4	3	2	1	0
	BF[1:0]	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P	
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7、6	BF[1:0]	00	R/W	TOLBR 缓冲传送时序的选择 选择 TOLBR 到 TOCR2 的缓冲传送时序。 详细内容请参照表 11.32。
5	OLS3N	0	R/W	输出电平的选择 3N* 在复位同步 PWM 模式 / 互补 PWM 模式中，选择 TIOC4D 的输出电平， 请参照表 11.33。
4	OLS3P	0	R/W	输出电平的选择 3P* 在复位同步 PWM 模式 / 互补 PWM 模式中，选择 TIOC4B 的输出电平。 请参照表 11.34。
3	OLS2N	0	R/W	输出电平的选择 2N* 在复位同步 PWM 模式 / 互补 PWM 模式中，选择 TIOC4C 的输出电平。 请参照表 11.35。
2	OLS2P	0	R/W	输出电平的选择 2P* 在复位同步 PWM 模式 / 互补 PWM 模式中，选择 TIOC4A 的输出电平。 请参照表 11.36。
1	OLS1N	0	R/W	输出电平的选择 1N* 在复位同步 PWM 模式 / 互补 PWM 模式中，选择 TIOC3D 的输出电平。 请参照表 11.37。
0	OLS1P	0	R/W	输出电平的选择 1P* 在复位同步 PWM 模式 / 互补 PWM 模式中，选择 TIOC3B 的输出电平。 请参照表 11.38。

【注】 \* 通过将 TOCR1 的 TOCS 位设定为 1，本设定有效。

表 11.32 BF1、BF0 位的设定

bit7	bit6	说 明	
BF1	BF0	互补 PWM 模式	复位 PWM 模式
0	0	不将缓冲寄存器 (TOLBR) 的值传送到 TOCR2	不将缓冲寄存器 (TOLBR) 的值传送到 TOCR2
0	1	在 TCNT_4 的波峰, 将缓冲寄存器 (TOLBR) 的值传送到 TOCR2	在清除 TCNT_3/4 计数器时, 将缓冲寄存器 (TOLBR) 的值传送到 TOCR2
1	0	在 TCNT_4 的波谷, 将缓冲寄存器 (TOLBR) 的值传送到 TOCR2	禁止设定
1	1	在 TCNT_4 的波峰和波谷, 将缓冲寄存器 (TOLBR) 的值传送到 TOCR2	禁止设定

表 11.33 TIOC4D 输出电平的选择功能

bit5	功 能			
OLS3N	初始输出	有效电平	比较匹配输出	
			递增计数	递减计数
0	高电平	低电平	高电平	低电平
1	低电平	高电平	低电平	高电平

【注】 反相波形的初始输出值在开始计数并经过空载时间后变为有效电平。

表 11.34 TIOC4B 输出电平的选择功能

bit4	功 能			
OLS3P	初始输出	有效电平	比较匹配输出	
			递增计数	递减计数
0	高电平	低电平	低电平	高电平
1	低电平	高电平	高电平	低电平

表 11.35 TIOC4C 输出电平的选择功能

bit3	功 能			
OLS2N	初始输出	有效电平	比较匹配输出	
			递增计数	递减计数
0	高电平	低电平	高电平	低电平
1	低电平	高电平	低电平	高电平

【注】 反相波形的初始输出值在开始计数并经过空载时间后变为有效电平。

表 11.36 TIOC4A 输出电平的选择功能

bit2	功 能			
OLS2P	初始输出	有效电平	比较匹配输出	
			递增计数	递减计数
0	高电平	低电平	低电平	高电平
1	低电平	高电平	高电平	低电平

表 11.37 TIOC3D 输出电平的选择功能

bit1	功 能			
OLS1N	初始输出	有效电平	比较匹配输出	
			递增计数	递减计数
0	高电平	低电平	高电平	低电平
1	低电平	高电平	低电平	高电平

【注】 反相波形的初始输出值在开始计数并经过空载时间后变为有效电平。

表 11.38 TIOC4B 输出电平的选择功能

bit0	功 能			
OLS1P	初始输出	有效电平	比较匹配输出	
			递增计数	递减计数
0	高电平	低电平	低电平	高电平
1	低电平	高电平	高电平	低电平



### 11.3.22 定时器的输出电平缓冲寄存器 (TOLBR)

TOLBR 是 TOCR2 的缓冲寄存器，设定互补 PWM 模式 / 复位同步 PWM 模式中的 PWM 输出电平，是 8 位可读写寄存器。

位:	7	6	5	4	3	2	1	0
	-	-	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7、6	—	全 0	R	保留位 读写的值总是 0。
5	OLS3N	0	R/W	必须给 TOCR2 的 OLS3N 位设定缓冲传送的值。
4	OLS3P	0	R/W	必须给 TOCR2 的 OLS3P 位设定缓冲传送的值。
3	OLS2N	0	R/W	必须给 TOCR2 的 OLS2N 位设定缓冲传送的值。
2	OLS2P	0	R/W	必须给 TOCR2 的 OLS2P 位设定缓冲传送的值。
1	OLS1N	0	R/W	必须给 TOCR2 的 OLS1N 位设定缓冲传送的值。
0	OLS1P	0	R/W	必须给 TOCR2 的 OLS1P 位设定缓冲传送的值。

在缓冲运行中设定 PWM 输出电平时的设定步骤例子如图 11.3 所示。

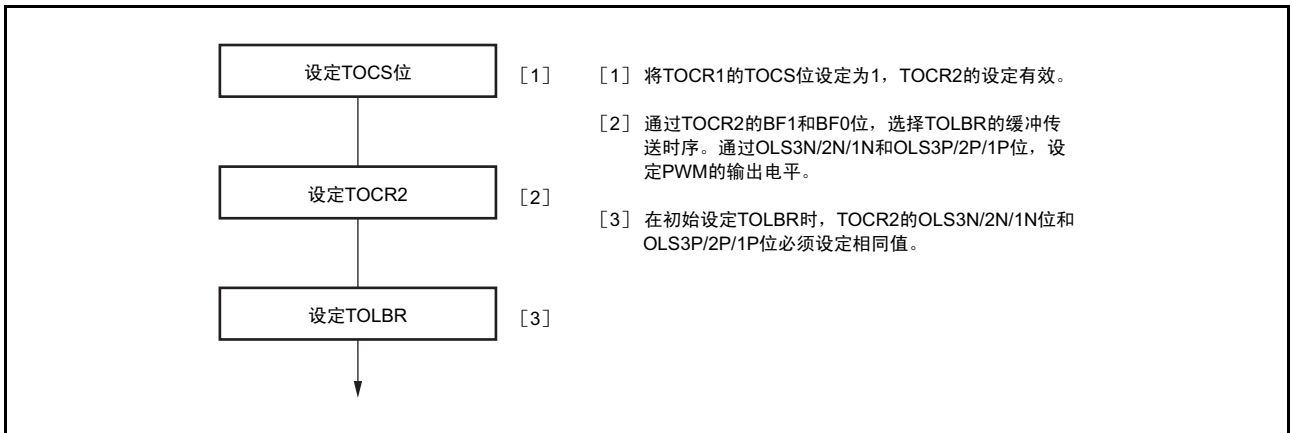


图 11.3 在缓冲运行中设定 PWM 输出电平时的设定步骤例子

## 11.3.23 定时器的门控寄存器 (TGCR)

TGCR 是 8 位可读写寄存器，在复位同步 PWM 模式 / 互补 PWM 模式中进行无刷 DC 马达控制所需要的波形输出控制。除了互补 PWM 模式 / 复位同步 PWM 模式以外，本寄存器的设定无效。

位:	7	6	5	4	3	2	1	0
	-	BDC	N	P	FB	WF	VF	UF
初始值:	1	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7	—	1	R	保留位 读写的值总是 1。
6	BDC	0	R/W	无刷 DC 马达 选择是将本寄存器的功能设定为有效还是无效。 0: 通常输出 1: 本寄存器的功能有效
5	N	0	R/W	反相输出 (N) 控制 当反相引脚 (TIOC3D 引脚、TIOC4C 引脚、TIOC4D 引脚) 为输出引脚时，选择电平输出或者复位同步 PWM/ 互补 PWM 输出。 0: 电平输出 1: 复位同步 PWM/ 互补 PWM 输出
4	P	0	R/W	正相输出 (P) 控制 当正相引脚 (TIOC3B 引脚、TIOC4A 引脚、TIOC4B 引脚) 为输出引脚时，选择电平输出或者复位同步 PWM/ 互补 PWM 输出。 0: 电平输出 1: 复位同步 PWM/ 互补 PWM 输出
3	FB*	0	R/W	外部反馈信号的允许 选择是通过 MTU2/ 通道 0 的 TGRA、TGRB、TGRC 的输入捕捉信号还是通过给 TGCR 的 bit2 ~ 0 写 0 或者 1 自动进行正相 / 反相的输出转换。 0: 通过外部输入进行输出的转换 (输入源为通道 0 的 TGRA、TGRB、TGRC 的输入捕捉信号) 1: 通过软件进行输出的转换 (TGCR 的 UF、VF、WF 的设定值)
2	WF	0	R/W	输出相的转换 2 ~ 0 设定正相 / 反相的输出相 ON/OFF。这些位的设定只在本寄存器的 FB 位为 1 时有效。此时，bit2 ~ 0 的设定取代外部输入。请参照表 11.39。
1	VF	0	R/W	
0	UF	0	R/W	

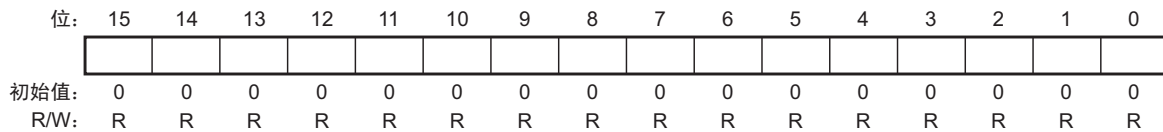
【注】 \* 在 MTU2S 上将 BDC 位置 1 时，FB 位一定不能设定为 0。

表 11.39 输出电平的选择功能

bit2	bit1	bit0	功 能					
			TIOC3B	TIOC4A	TIOC4B	TIOC3D	TIOC4C	TIOC4D
WF	VF	UF	U相	V相	W相	U相	V相	W相
0	0	0	OFF	OFF	OFF	OFF	OFF	OFF
0	0	1	ON	OFF	OFF	OFF	OFF	ON
0	1	0	OFF	ON	OFF	ON	OFF	OFF
0	1	1	OFF	ON	OFF	OFF	OFF	ON
1	0	0	OFF	OFF	ON	OFF	ON	OFF
1	0	1	ON	OFF	OFF	OFF	ON	OFF
1	1	0	OFF	OFF	ON	ON	OFF	OFF
1	1	1	OFF	OFF	OFF	OFF	OFF	OFF

### 11.3.24 定时器的子计数器 (TCNTS)

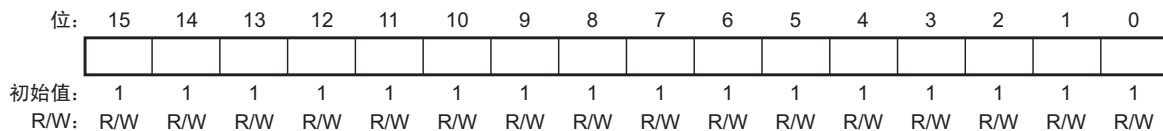
TCNTS 是只在互补 PWM 模式中使用的 16 位只读计数器，初始值为 H'0000。



【注】 禁止以8位为单位存取TCNTS，必须以16位为单位进行存取。

### 11.3.25 定时器的空载时间数据寄存器 (TDDR)

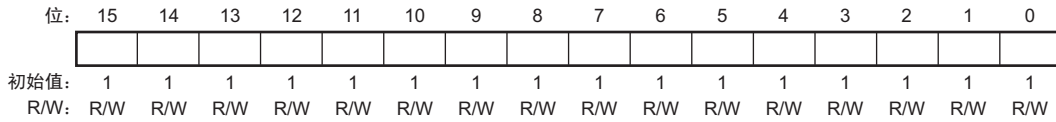
TDDR 是只在互补 PWM 模式中使用的 16 位寄存器，设定 TCNT\_3 和 TCNT\_4 计数器的偏移值。在互补 PWM 模式中清除 TCNT\_3、TCNT\_4 计数器后重新开始计数时，将 TDDR 寄存器的值装入到 TCNT\_3 计数器并开始计数。TDDR 的初始值为 H'FFFF。



【注】 禁止以8位为单位存取TDDR，必须以16位为单位进行存取。

### 11.3.26 定时器的周期数据寄存器 (TCDR)

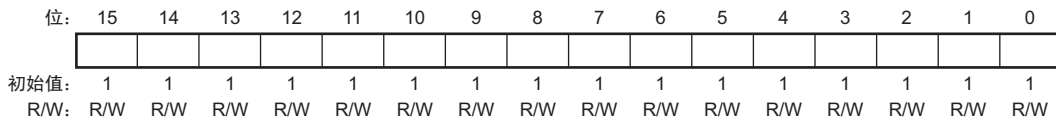
TCDR 是只在互补 PWM 模式中使用的 16 位寄存器。TCDR 寄存器的值必须设定为 PWM 载波周期的 1/2。本寄存器在互补 PWM 模式中随时和 TCNTS 计数器比较，如果一致，TCNTS 计数器就转换计数方向（递减计数 → 递增计数）。TCDR 的初始值为 H'FFFF。



【注】 禁止以8位为单位存取TCDR，必须以16位为单位进行存取。

### 11.3.27 定时器的周期缓冲寄存器 (TCBR)

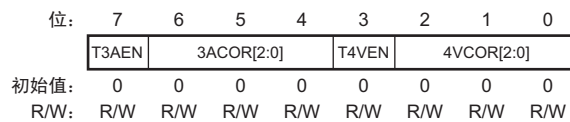
TCBR 是只在互补 PWM 模式中使用的 16 位寄存器，用作 TCDR 寄存器的缓冲寄存器。在 TMDR 寄存器设定的传送时序中将 TCBR 寄存器的值传送到 TCDR 寄存器。TCBR 的初始值为 H'FFFF。



【注】 禁止以8位为单位存取TCBR，必须以16位为单位进行存取。

### 11.3.28 定时器的跳过中断设定寄存器 (TITCR)

TITCR 是 8 位可读写寄存器，禁止或者允许跳过中断功能以及设定跳过中断次数。MTU2 有 1 个 TITCR。



位	位名	初始值	R/W	说 明
7	T3AEN	0	R/W	T3AEN 禁止或者允许 TGIA_3 的跳过中断功能。 0: 禁止 TGIA_3 的跳过中断功能 1: 允许 TGIA_3 的跳过中断功能
6 ~ 4	3ACOR[2:0]	000	R/W	以 0 ~ 7 次设定 TGIA_3 的跳过中断次数。* 详细内容请参照表 11.40。
3	T4VEN	0	R/W	T4VEN 禁止或者允许 TCIV_4 的跳过中断功能。 0: 禁止 TCIV_4 的跳过中断功能 1: 允许 TCIV_4 的跳过中断功能
2 ~ 0	4VCOR[2:0]	000	R/W	以 0 ~ 7 次设定 TCIV_4 的跳过中断次数。* 详细内容请参照表 11.41。

【注】 \* 如果将跳过中断次数设定为 0，就不跳过中断。  
另外，在更改跳过中断次数前，必须在将 T3AEN、T4VEN 位设定为 0 后清除跳过中断次数计数器 (TITCNT)。

表 11.40 通过 3ACOR2 ~ 3ACOR0 位设定跳过中断次数

bit6	bit5	bit4	说 明
3ACOR2	3ACOR1	3ACOR0	
0	0	0	不跳过 TGIA_3 的中断
0	0	1	将 TGIA_3 的跳过中断次数设定为 1 次
0	1	0	将 TGIA_3 的跳过中断次数设定为 2 次
0	1	1	将 TGIA_3 的跳过中断次数设定为 3 次
1	0	0	将 TGIA_3 的跳过中断次数设定为 4 次
1	0	1	将 TGIA_3 的跳过中断次数设定为 5 次
1	1	0	将 TGIA_3 的跳过中断次数设定为 6 次
1	1	1	将 TGIA_3 的跳过中断次数设定为 7 次

表 11.41 通过 4VCOR2 ~ 4VCOR0 位设定跳过中断次数

bit2	bit1	bit0	说 明
4VCOR2	4VCOR1	4VCOR0	
0	0	0	不跳过 TCIV_4 的中断
0	0	1	将 TCIV_4 的跳过中断次数设定为 1 次
0	1	0	将 TCIV_4 的跳过中断次数设定为 2 次
0	1	1	将 TCIV_4 的跳过中断次数设定为 3 次
1	0	0	将 TCIV_4 的跳过中断次数设定为 4 次
1	0	1	将 TCIV_4 的跳过中断次数设定为 5 次
1	1	0	将 TCIV_4 的跳过中断次数设定为 6 次
1	1	1	将 TCIV_4 的跳过中断次数设定为 7 次

## 11.3.29 定时器的跳过中断次数计数器 (TITCNT)

TITCNT 是 8 位可读计数器。MTU2 有 1 个 TITCNT，在 TCNT\_3 和 TCNT\_4 停止计数后，保持 TITCNT 的值。

位:	7	6	5	4	3	2	1	0
	-	3ACNT[2:0]			-	4VCNT[2:0]		
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
7	—	0	R	保留位 总是读 0。
6 ~ 4	3ACNT[2:0]	000	R	TGIA_3 中断计数器 如果将 TITCR 的 T3AEN 位设定为 1，就在出现 TGIA_3 中断源时进行递增计数（加 1）。 [清除条件] <ul style="list-style-type: none"> <li>• TITCR 的 3ACOR2 ~ 3ACOR0 和 TITCNT 的 3ACNT2 ~ 3ACNT0 一致时</li> <li>• TITCR 的 T3AEN 位为 0 时</li> <li>• 当 TITCR 的 3ACOR2 ~ 3ACOR0 为 0 时</li> </ul>
3	—	0	R	保留位 总是读 0。
2 ~ 0	4VCNT[2:0]	000	R	TCIV_4 中断计数器 如果将 TITCR 的 T4VEN 位设定为 1，就在出现 TCIV_4 中断源时进行递增计数（加 1）。 [清除条件] <ul style="list-style-type: none"> <li>• 当 TITCR 的 4VCOR2 ~ 4VCOR0 和 TITCNT 的 4VCNT2 ~ 4VCNT0 一致时</li> <li>• 当 TITCR 的 T4VEN 位为 0 时</li> <li>• 当 TITCR 的 4VCOR2 ~ 4VCOR0 为 0 时</li> </ul>

**【注】** 在要清除 TITCNT 的值时，必须将 TITCR 的 T3AEN 位和 T4VEN 位清 0。

## 11.3.30 定时器的缓冲传送设定寄存器 (TBTER)

TBTER 是 8 位可读写寄存器，设定是否抑制缓冲寄存器\*（在互补 PWM 模式中使用的缓冲寄存器）到暂存器的传送，或者设定是否与跳过中断功能联动。MTU2 有 1 个 TBTER。

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	BTE[1:0]	
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

位	位名	初始值	R/W	说明
7~2	—	全 0	R	保留位 读写的值总是 0。
1、0	BTE[1:0]	00	R/W	设定是否抑制缓冲寄存器*（互补 PWM 模式中使用的缓冲寄存器）到暂存器的传送，或者设定是否与跳过中断功能联动。详细内容请参照表 11.42。

【注】 \* 对象缓冲寄存器  
TGRC\_3、TGRD\_3、TGRC\_4、TGRD\_4、TCBR

表 11.42 BTE1 和 BTE0 的位设定

bit1	bit0	说明
BTE1	BTE0	
0	0	不抑制缓冲寄存器到暂存器的传送*1 并且不与跳过中断功能联动
0	1	抑制缓冲寄存器到暂存器的传送
1	0	将缓冲寄存器到暂存器的传送*2与跳过中断功能联动*2
1	1	禁止设定

【注】 \*1 按照 TMDR 的 MD3 ~ MD0 的设定进行传送。详细内容请参照“11.4.8 互补 PWM 模式”。

\*2 在禁止跳过中断功能时（在将定时器的跳过中断设定寄存器（TITCR）的 T3AEN、T4VEN 位或者 TITCR 的跳过间隔次数设定（3ACOR 和 4VCOR）置 0 时），必须设定为缓冲传送不与跳过中断功能联动（将定时器的缓冲传送寄存器（TBTER）的 BTE1 置 0）。如果设定为缓冲传送与跳过中断功能联动，就不进行缓冲传送。

### 11.3.31 定时器的空载时间允许寄存器 (TDER)

TDER 是 8 位可读写寄存器。通道 3 有 1 个 TDER，能控制互补 PWM 模式的空载时间的生成。MTU2 有 1 个 TDER。必须在 TCNT 停止运行的状态下进行 TDER 的设定。

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	TDER
初始值:	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R/(W)

位	位名	初始值	R/W	说 明
7 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	TDER	1	R/(W)	空载时间允许寄存器 设定是否生成空载时间。 0: 不生成空载时间 1: 生成空载时间 * [清除条件] • TDER=1 的状态下, 当读 TDER 后, 给 TDER 写 0 时

【注】 \* 必须设定为  $TDDR \geq 1$ 。

### 11.3.32 定时器的波形控制寄存器 (TWCR)

TWCR 是 8 位可读写寄存器，设定在互补 PWM 模式中发生 TNCT\_3、TNCT\_4 的同步计数器清除时的输出波形的控制以及设定是否进行由 TGRA\_3 比较匹配引起的计数器清除。必须在停止 TCNT 运行的状态下进行 TWCR 的 CCE 位和 WRE 位的设定。

位:	7	6	5	4	3	2	1	0
	CCE	-	-	-	-	-	SCC	WRE
初始值:	0*	0	0	0	0	0	0	0
R/W:	R/(W)	R	R	R	R	R	R/(W)	R/(W)

【注】 \* 除了互补 PWM 模式 1 以外, 不能置 1。



位	位名	初始值	R/W	说 明
7	CCE	0*	R/(W)	<p>比较匹配清除的允许</p> <p>设定是否在互补 PWM 模式中进行由 TGRA_3 比较匹配引起的计数器清除。</p> <p>0: 不进行由 TGRA_3 比较匹配引起的计数器清除</p> <p>1: 进行由 TGRA_3 比较匹配引起的计数器清除</p> <p>[置位条件]</p> <ul style="list-style-type: none"> <li>• CCE=0 的状态下, 当读 CCE 后, 给 CCE 写 1 时</li> </ul>
6 ~ 2	—	全 0	R	<p>保留位</p> <p>读写的值总是 0。</p>
1	SCC	0	R/(W)	<p>同步清除控制</p> <p>设定在互补 PWM 模式中发生 MTU2-MTU2S 计数器同步清除时是否清除 MTU2S 的 TCNT_3 和 TCNT_4。</p> <p>在使用本功能时, 必须将 MTU2S 设定为互补 PWM 模式。</p> <p>另外, 在计数器运行中改写 SCC 位时, 不能更改 CCE 位和 WRE 位的值。只有在波谷的 Tb 区间以外的区间发生同步清除时, 才能通过设定 SCC 位使由 MTU2 进行的同步清除变为无效。如果在包含 TCNT_3 和 TCNT_4 刚开始计数后的波谷的 Tb 区间发生同步清除, 就清除 MTU2S 的 TCNT_3 和 TCNT_4。</p> <p>有关互补 PWM 模式的波谷 Tb 区间, 请参照图 11.40。</p> <p>对于 MTU2, 此位是保留位, 读写的值总是 0。</p> <p>0: 由 MTU2-MTU2S 同步清除功能进行的 MTU2S 的 TCNT_3 和 TCNT_4 清除有效</p> <p>1: 由 MTU2-MTU2S 同步清除功能进行的 MTU2S 的 TCNT_3 和 TCNT_4 清除无效</p> <p>[置位条件]</p> <ul style="list-style-type: none"> <li>• SCC=0 的状态下, 当读 SCC 后, 给 SCC 写 1 时</li> </ul>
0	WRE	0	R/(W)	<p>波形保持的允许</p> <p>选择在互补 PWM 模式中发生同步计数器清除时的输出波形。本功能只在互补 PWM 模式的波谷 Tb 区间发生同步清除时保持波形。如果在其他区间发生同步清除, 就与 WRE 位的设定无关, 输出 TOCR 寄存器设定的初始值。另外, 如果在 TCNT_3、TCNT_4 刚开始的波谷 Tb 区间发生同步清除, 也输出 TOCR 寄存器设定的初始值。</p> <p>有关互补 PWM 模式的波谷 Tb 区间, 请参照图 11.40。</p> <p>0: 输出 TOCR 寄存器设定的初始输出值</p> <p>1: 保持同步清除前的波形</p> <p>[置位条件]</p> <ul style="list-style-type: none"> <li>• WRE=0 的状态下, 当读 WRE 后, 给 WRE 写 1 时</li> </ul>

【注】 \* 当不是互补 PWM 模式 1 时, 不能设定为 1。

### 11.3.33 和总线主控的接口

定时器的计数器 (TCNT)、通用寄存器 (TGR)、定时器的子计数器 (TCNTS)、定时器的周期缓冲寄存器 (TCBR)、定时器的空载时间数据寄存器 (TDDR)、定时器的周期数据寄存器 (TCDR)、定时器的 A/D 转换开始请求的控制寄存器 (TADCR)、定时器的 A/D 转换开始请求的周期设定寄存器 (TADCOR) 以及定时器的 A/D 转换开始请求的周期设定缓冲寄存器 (TADCOBR) 是 16 位寄存器。因为和总线主控之间的数据总线为 16 位宽, 所以能以 16 位为单位进行读写, 而不能以 8 位为单位进行读写。必须以 16 位为单位进行存取。

上述以外的寄存器是 8 位寄存器, 因为和 CPU 之间的数据总线为 16 位宽, 所以能以 16 位为单位进行读写, 也能以 8 位为单位进行读写。

## 11.4 运行说明

### 11.4.1 基本运行

各通道有 TCNT 和 TGR。TCNT 能进行递增计数运行、自由运行、周期计数器运行和外部事件计数运行。TGR 能分别用作输入捕捉寄存器和输出比较寄存器。必须通过引脚功能控制器 (PFC) 设定 MTU2 的外部引脚功能。

#### (1) 计数器的运行

如果将 TSTR 的 CST0 ~ CST4 位、TSTR\_5 的 CSTU5、CSTV5、CSTW5 位设定为 1, 对应通道的 TCNT 就开始计数运行, 能进行自由运行计数器的运行和周期计数器运行等。

#### (a) 计数运行的设定步骤例子

计数运行的设定步骤例子如图 11.4 所示。

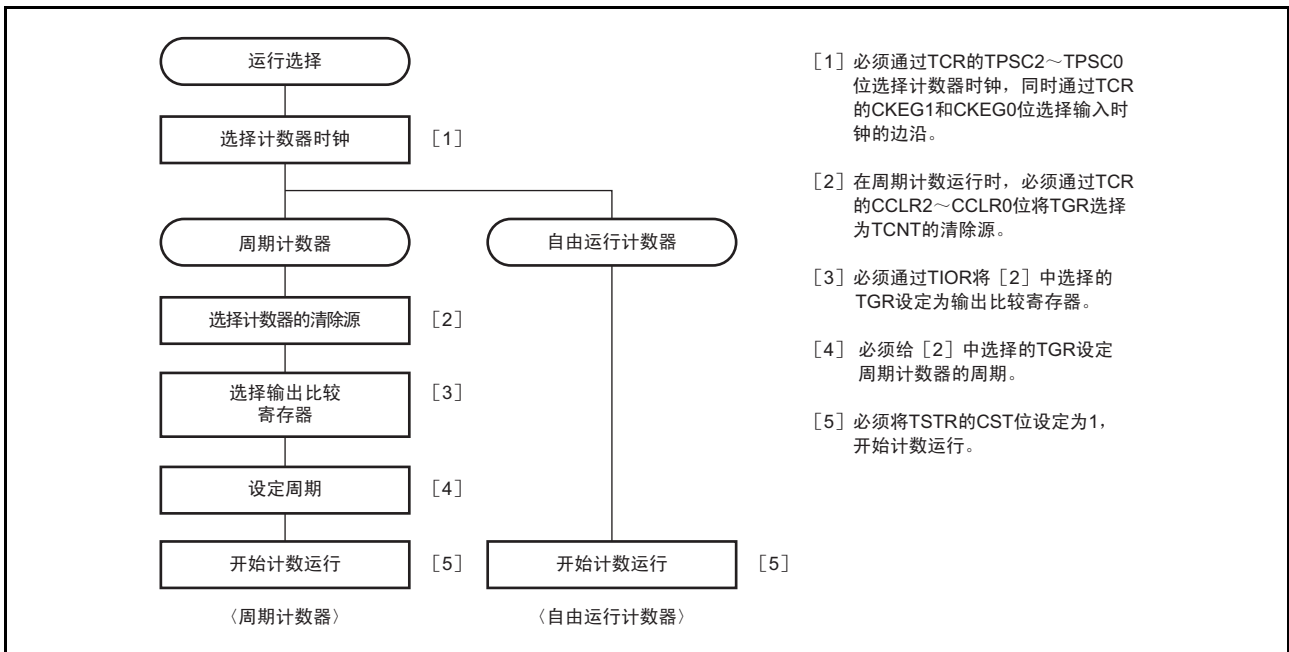


图 11.4 计数器运行的设定步骤例子

## (b) 自由运行计数的运行和周期计数运行

MTU2 的 TCNT 在刚复位后全部为自由运行计数器的设定，如果将 TSTR 的对应位设定为 1，就作为自由运行计数器开始递增计数。当 TCNT 发生上溢 (H'FFFF→H'0000) 时，TSR 的 TCFV 位就被置 1。此时，如果对应 TIER 的 TCIEV 位是 1，MTU2 就请求中断。在 TCNT 发生上溢后，从 H'0000 开始继续递增计数。

自由运行计数器的运行如图 11.5 所示。

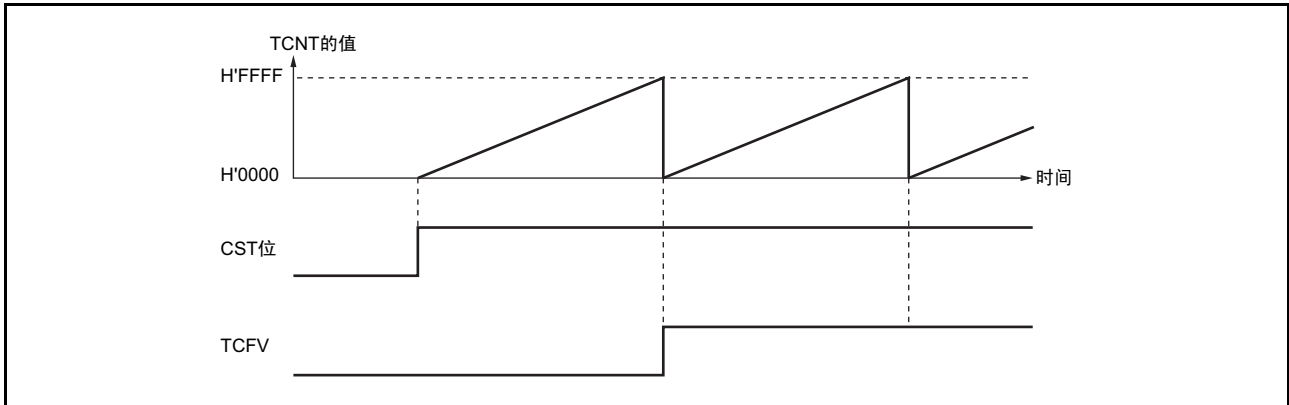


图 11.5 自由运行计数器的运行

当选择比较匹配为 TCNT 的清除源时，对应通道的 TCNT 就进行周期计数运行。将用于周期设定的 TGR 设定为输出比较寄存器，并通过 TCR 的 CCLR2 ~ CCLR0 位选择比较匹配的计数器清除。在设定后，如果将 TSTR 的对应位设定为 1，就作为周期计数器开始递增计数运行。当计数值和 TGR 的值一致时，TSR 的 TGF 位就被置 1，TCNT 被清 H'0000。

此时，如果对应 TIER 的 TGIE 位是 1，MTU2 就请求中断。TCNT 在进行比较匹配后，从 H'0000 开始继续递增计数。

周期计数器的运行如图 11.6 所示。

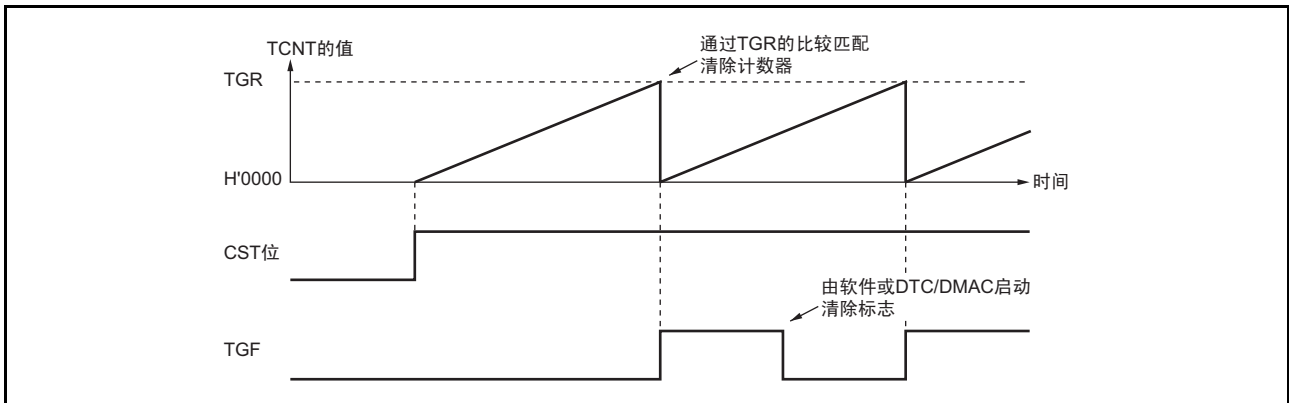


图 11.6 周期计数器的运行

(2) 比较匹配的波形输出功能

MTU2 能通过比较匹配从对应的输出引脚进行 0 输出 / 1 输出 / 交替输出。

(a) 通过比较匹配进行波形输出运行的设定步骤例子

通过比较匹配进行波形输出运行的设定步骤例子如图 11.7 所示。

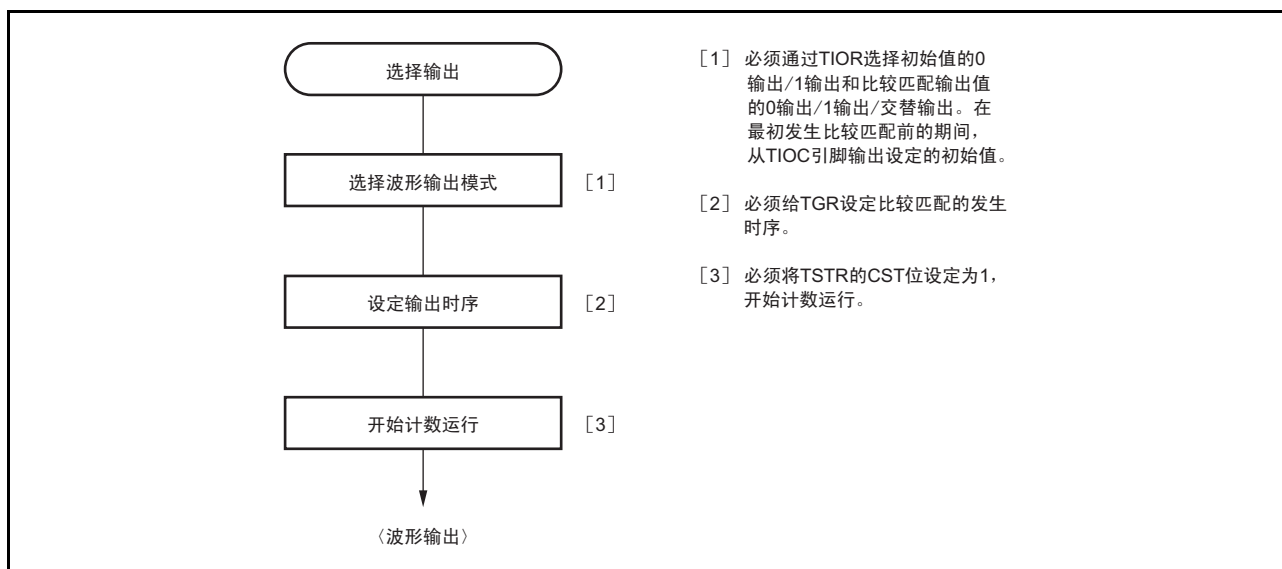


图 11.7 通过比较匹配进行波形输出的运行例子

(b) 波形输出运行的例子

0 输出 / 1 输出的例子如图 11.8 所示。

在此例中，将 TCNT 作为自由运行的计数运行，并设定为通过比较匹配 A 进行 1 输出、通过比较匹配 B 进行 0 输出。当设定的电平和引脚的电平一致时，引脚的电平不变。

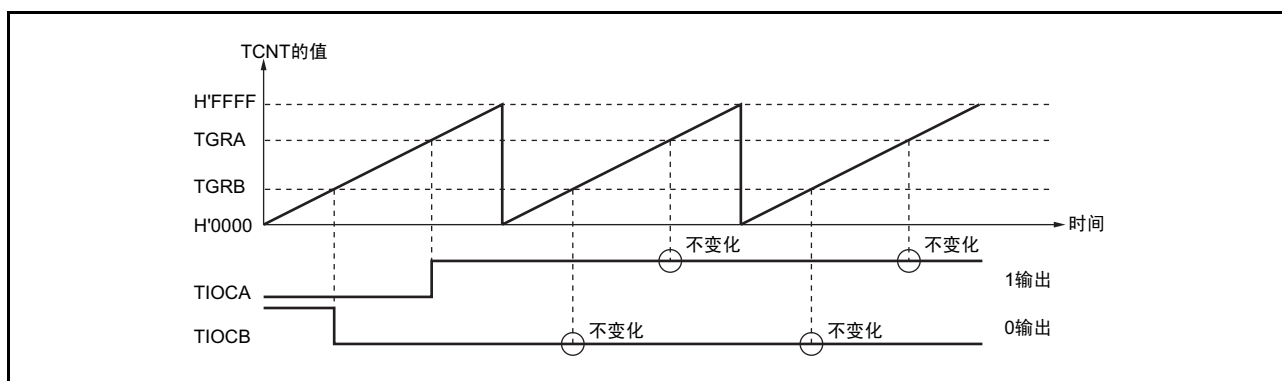


图 11.8 0 输出 / 1 输出的运行例

交替输出的例子如图 11.9 所示。

在此例中，将 TCNT 作为周期计数运行（通过比较匹配 B 进行计数器清除），并将比较匹配 A 和 B 都设定为交替输出。

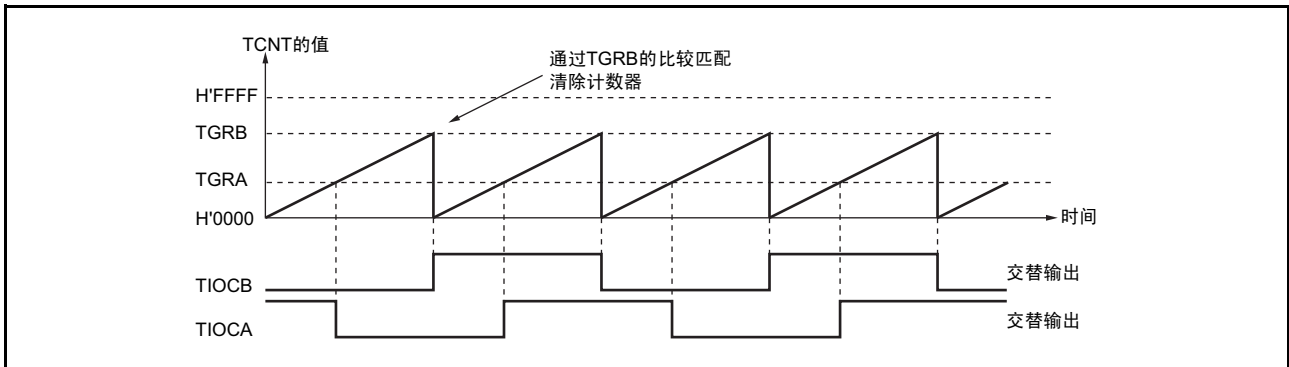


图 11.9 交替输出的运行例

### (3) 输入捕捉功能

能检测 TIOC 引脚的输入边沿，将 TCNT 的值传送到 TGR。

能从上升沿 / 下降沿 / 双边沿中选择检测边沿。通道 0、1 也能将其他通道的计数器输入时钟或者比较匹配信号作为输入捕捉源。

**【注】** 如果在通道 0、1 中将其他通道的计数器输入时钟作为输入捕捉的输入，就不能选择 MPφ/1 作为输入捕捉的计数器输入时钟。如果选择 MPφ/1，就不发生输入捕捉。

#### (a) 输入捕捉运行的设定步骤例子

输入捕捉运行的设定步骤例子如图 11.10 所示。

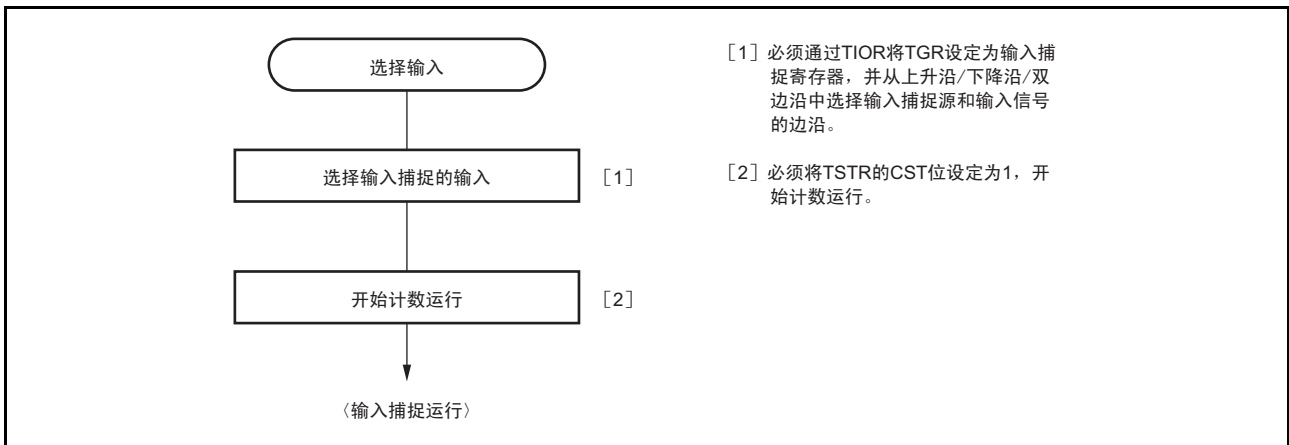


图 11.10 输入捕捉运行的设定例

## (b) 输入捕捉运行的例子

输入捕捉的运行例子如图 11.11 所示。

在此例中，选择上升/下降的双边沿为 TIOCA 引脚输入捕捉的输入边沿，选择下降沿为 TIOCB 引脚输入捕捉的输入边沿，并且将 TCNT 设定为通过 TGRB 的输入捕捉进行计数器清除。

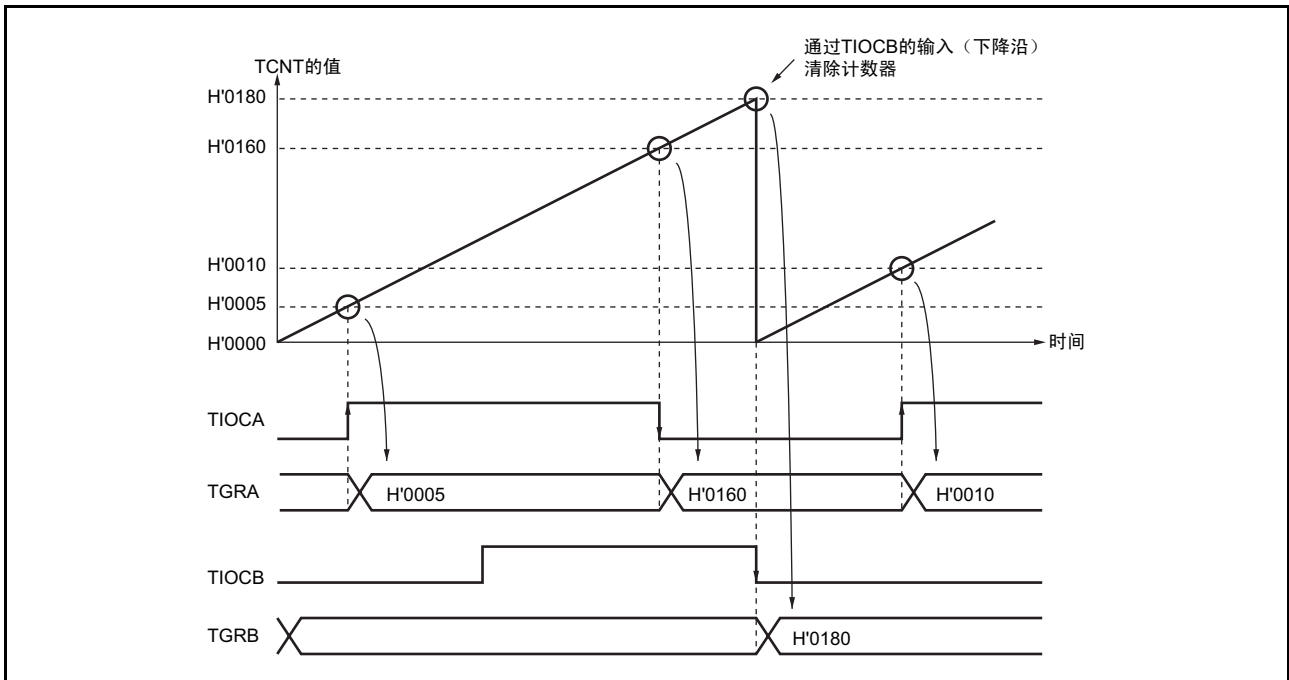


图 11.11 输入捕捉的运行例

### 11.4.2 同步运行

同步运行能同时改写多个 TCNT 的值（同步预置），还能通过设定 TCR 同时清除多个 TCNT（同步清除）。

通过同步运行，能增加对于 1 个时基要运行的 TGR 个数。

通道 0 ~ 4 都能设定为同步运行。

通道 5 不能进行同步运行。

#### (1) 同步运行的设定步骤例子

同步运行的设定步骤例子如图 11.12 所示。

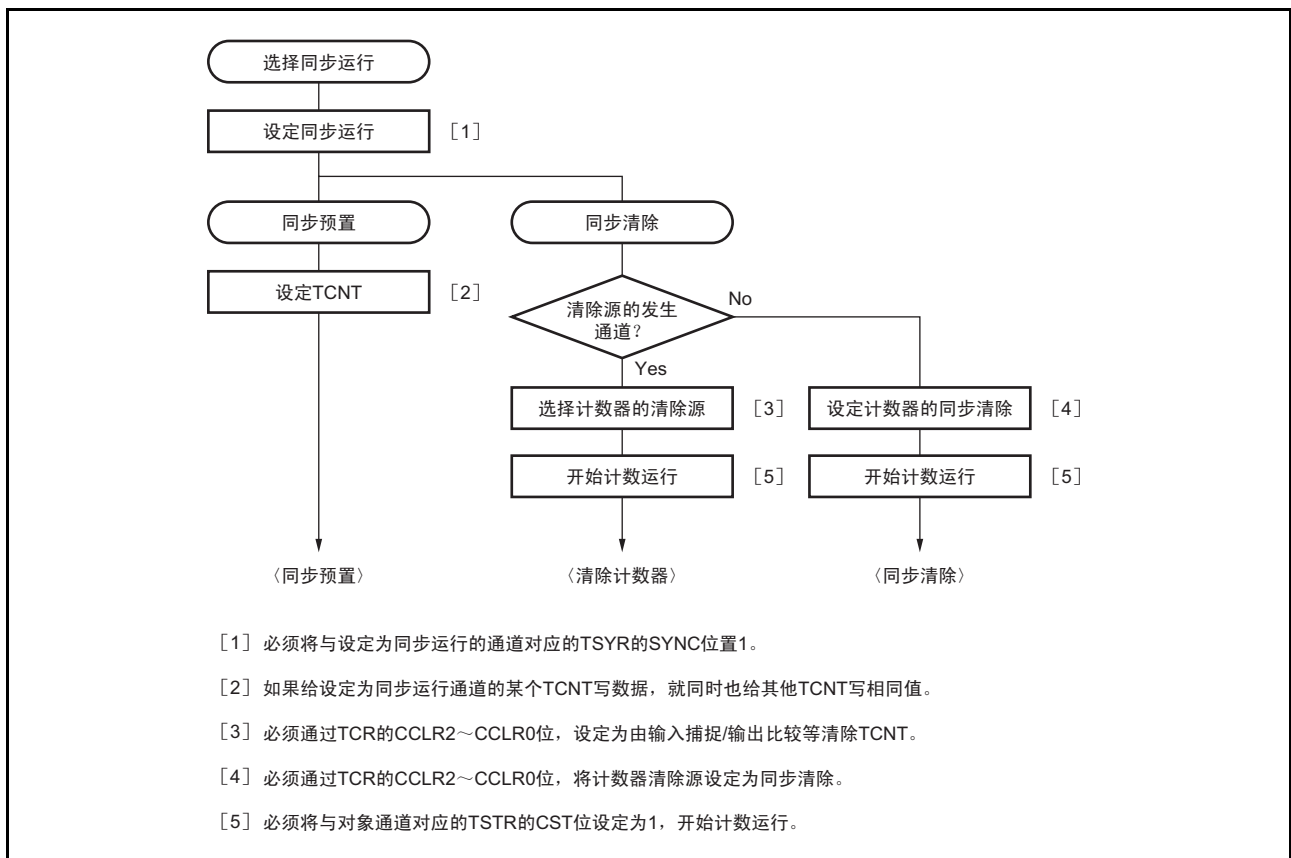


图 11.12 同步运行的设定步骤例子

(2) 同步运行的例子

同步运行的例子如图 11.13 所示。

在此例中，设定通道 0 ~ 2 为同步运行和 PWM 模式 1，通道 0 的计数器清除源为 TGRB\_0 的比较匹配以及通道 1、2 的计数器清除源为同步清除。

从 TIOC0A、TIOC1A、TIOC2A 引脚输出 3 相的 PWM 波形。此时，通道 0 ~ 2 的 TCNT 进行同步预置或者由 TGRB\_0 比较匹配引起的同步清除，并且设定在 TGRB\_0 的数据为 PWM 周期。

有关 PWM 模式请参照“11.4.5 PWM 模式”。

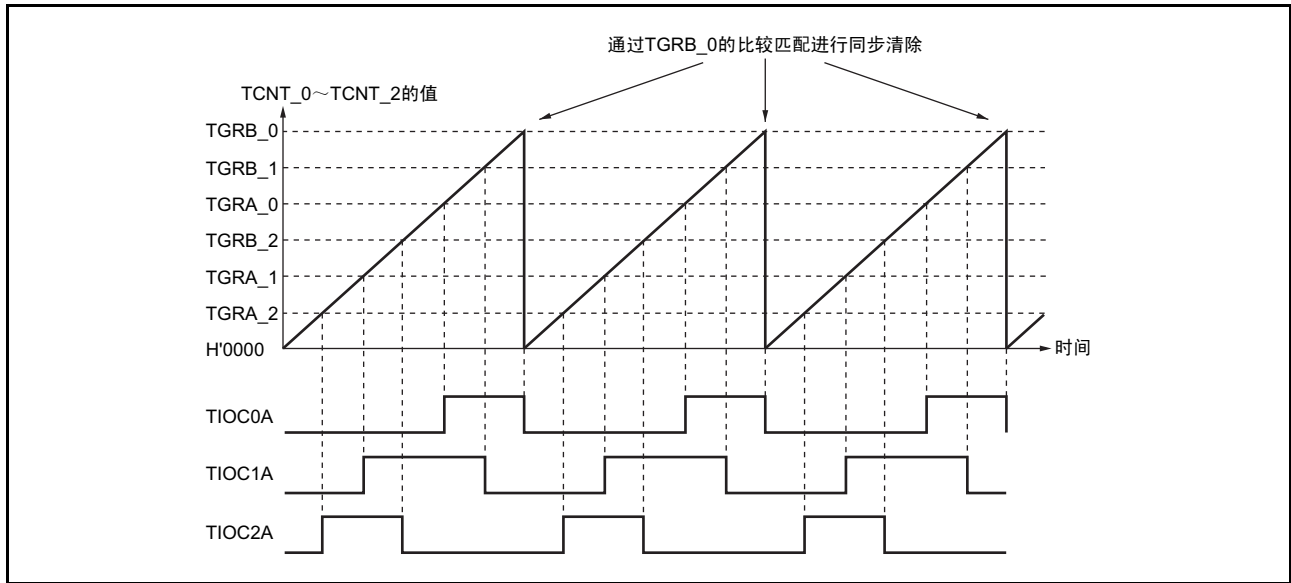


图 11.13 同步运行的例子

11.4.3 缓冲运行

缓冲运行是通道 0、3、4 具有的功能。TGRC 和 TGRD 能用作缓冲寄存器，并且通道 0 的 TGRF 也能用作缓冲寄存器。

缓冲运行因 TGR 是设定为输入捕捉寄存器还是设定为比较匹配寄存器而运行的内容不同。

【注】 TGRE\_0 不能设定为输入捕捉寄存器，只作为比较匹配寄存器运行。

缓冲运行时的寄存器组合如表 11.43 所示

表 11.43 寄存器的组合

通道	定时器的通用寄存器	缓冲寄存器
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0
	TGRE_0	TGRF_0
3	TGRA_3	TGRC_3
	TGRB_3	TGRD_3
4	TGRA_4	TGRC_4
	TGRB_4	TGRD_4



• TGR 为输出比较寄存器的情况

如果发生比较匹配，就将对应通道的缓冲寄存器的值传送到定时器的通用寄存器。  
此运行如图 11.14 所示。

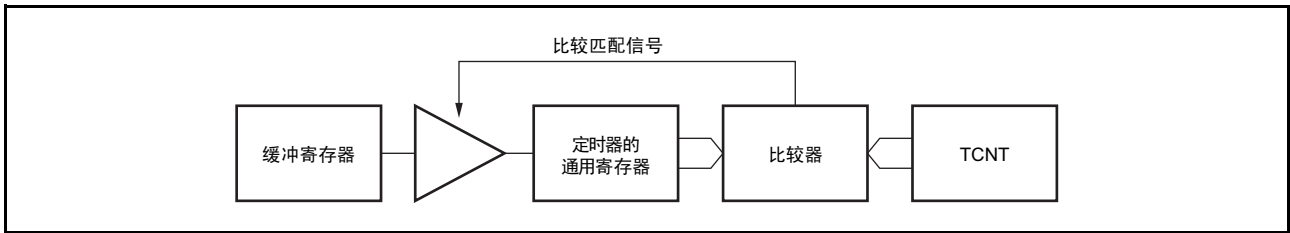


图 11.14 比较匹配的缓冲运行

• TGR 为输入捕捉寄存器的情况

如果发生输入捕捉，就在将 TCNT 的值传送到 TGR 的同时，将以前保存在 TGR 的值传送到缓冲寄存器。  
此运行如图 11.15 所示。

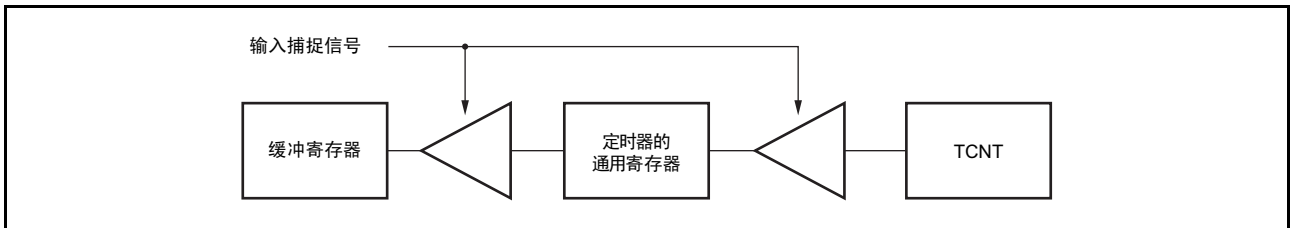


图 11.15 输入捕捉的缓冲运行

(1) 缓冲运行的设定步骤例子

缓冲运行的设定步骤例子如图 11.16 所示。

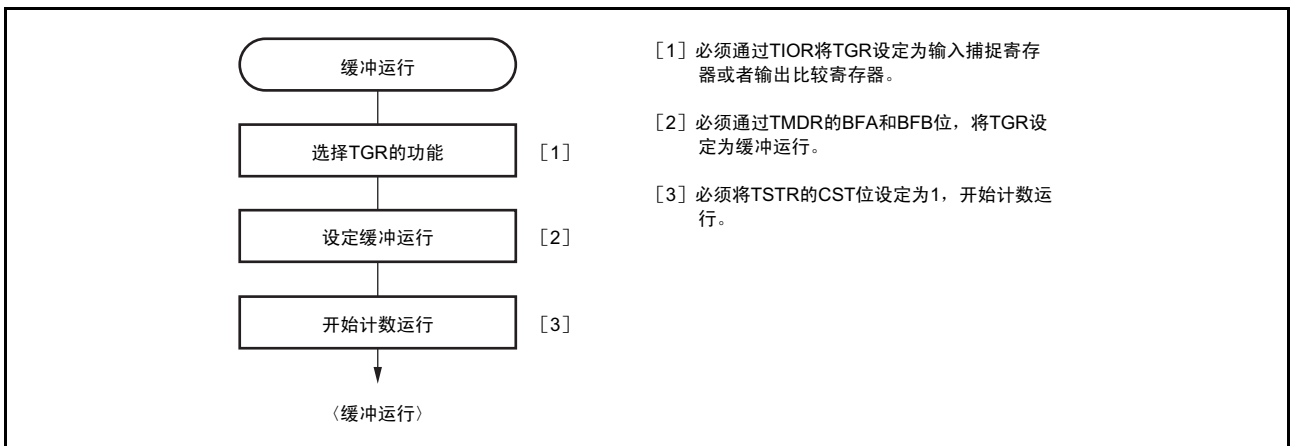


图 11.16 缓冲运行的设定步骤例子

## (2) 缓冲运行的例子

## (a) TGR 为输出比较寄存器的情况

设定通道 0 为 PWM 模式 1 以及 TGRA 和 TGRC 为缓冲运行时的运行例子如图 11.17 所示。在此例中，通过比较匹配 B 清除 TCNT、通过比较匹配 A 进行 1 输出以及通过比较匹配 B 进行 0 输出，并将 TBTM 的 TTSA 位设定为 0。

因为设定了缓冲运行，所以当发生比较匹配 A 时，就在使输出发生变化的同时，将缓冲寄存器 TGRC 的值传送到定时器的通用寄存器 TGRA。每当发生比较匹配 A 时，重复此运行。

有关 PWM 模式请参照“11.4.5 PWM 模式”。

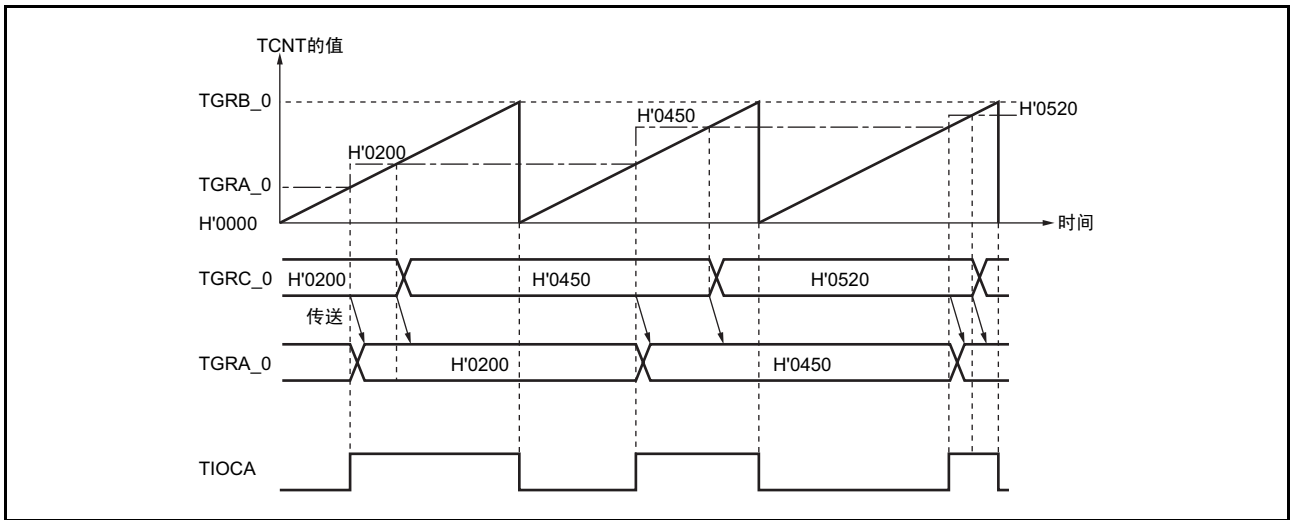


图 11.17 缓冲运行的例子 (1)

## (b) TGR 为输入捕捉寄存器的情况

设定 TGRA 为输入捕捉寄存器以及 TGRA 和 TGRC 为缓冲运行时的运行例子如图 11.18 所示。

TCNT 通过 TGRA 的输入捕捉进行计数器清除，选择上升 / 下降的双边沿为 TIOCA 引脚输入捕捉的输入边沿。

因为设定了缓冲运行，所以在通过输入捕捉 A 将 TCNT 的值保存到 TGRA 的同时，将以前保存在 TGRA 的值传送到 TGRC。

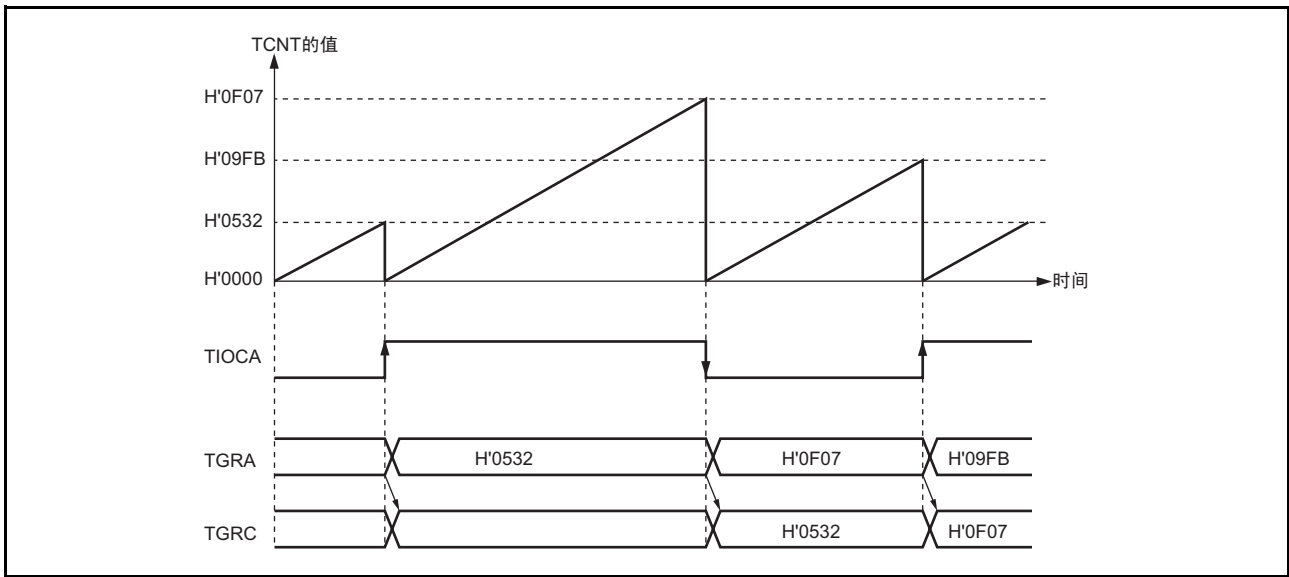


图 11.18 缓冲运行的例子 (2)

### (3) 选择缓冲运行时的缓冲寄存器到定时器的通用寄存器的传送时序

能通过设定缓冲运行的传送模式寄存器 (TBTM\_0、TBTM\_3、TBTM\_4)，选择缓冲寄存器 (对于通道 0 为 PWM 模式 1、2 中的缓冲寄存器；对于通道 3、4 为 PWM 模式 1 中的缓冲寄存器) 到定时器的通用寄存器的传送时序。可选择的缓冲传送时序是在发生比较匹配时 (初始值) 或是在清除 TCNT 时的时序。在此，所谓清除 TCNT 时是指以下的任意条件成立时：

- 当 TCNT 发生上溢 (H'FFFF→H'0000) 时
- 在计数器运行中，当给 TCNT 写 H'0000 时
- 当通过 TCR 的 CCLR2 ~ CCLR0 位设定的清除源将 TCNT 清除 H'0000 时

**【注】** 必须在停止 TCNT 运行的状态下进行 TBTM 的设定。

设定通道 0 为 PWM 模式 1 以及 TGRA\_0 和 TGRC\_0 为缓冲运行时的运行例子如图 11.19 所示。在此例中，通过比较匹配 B 清除 TCNT\_0、通过比较匹配 A 进行 1 输出以及通过比较匹配 B 进行 0 输出，并将 TBTM\_0 的 TTSA 位设定为 1。

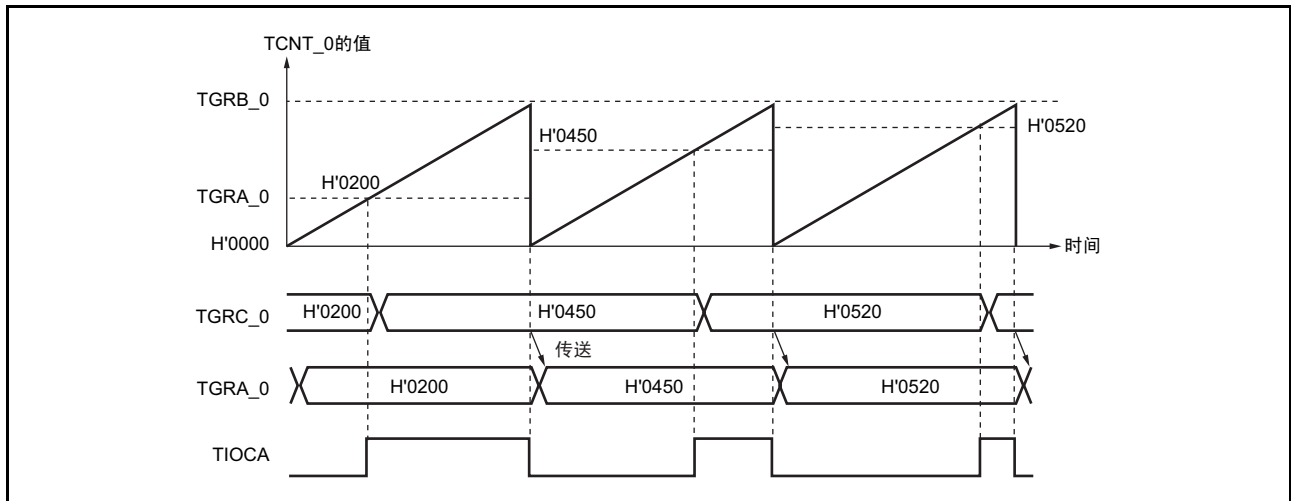


图 11.19 将 TGRC\_0 到 TGRA\_0 的缓冲传送时序选择为清除 TCNT\_0 时的运行例

#### 11.4.4 级联运行

级联运行是将 2 个通道的 16 位计数器连接为 32 位计数器的功能。

通过 TCR 的 TPSC2 ~ TPSC0 位将通道 1 的计数器时钟设定为“通过 TCNT\_2 的上溢 / 下溢进行计数”，运行此功能。

只在低 16 位的 TCNT 为相位计数模式时发生下溢。

级联的组合如表 11.44 所示。

**【注】** 如果将通道 1 设定为相位计数模式，计数器时钟的设定无效，并在相位计数模式中独立运行。

表 11.44 级联的组合

组合	高 16 位	低 16 位
通道 1 和通道 2	TCNT_1	TCNT_2

如果在级联运行时进行 TCNT\_1 和 TCNT\_2 的同时输入捕捉，就能通过输入捕捉控制寄存器 (TICCR) 的设定，将输入引脚追加到输入捕捉条件。有关级联时的输入捕捉，请参照“11.7.22 级联中 TCNT\_1 和 TCNT\_2 的同时输入捕捉”。

TICCR 设定值和输入捕捉的输入引脚的对应如表 11.45 所示。

表 11.45 TICCR 设定值和输入捕捉的输入引脚的对应

对象输入捕捉	TICCR 设定值	输入捕捉的输入引脚
从 TCNT_1 到 TGRA_1 的输入捕捉	I2AE 位 =0 (初始值)	TIOC1A
	I2AE 位 =1	TIOC1A、TIOC2A
从 TCNT_1 到 TGRB_1 的输入捕捉	I2BE 位 =0 (初始值)	TIOC1B
	I2BE 位 =1	TIOC1B、TIOC2B
从 TCNT_2 到 TGRA_2 的输入捕捉	I1AE 位 =0 (初始值)	TIOC2A
	I1AE 位 =1	TIOC2A、TIOC1A
从 TCNT_2 到 TGRB_2 的输入捕捉	I1BE 位 =0 (初始值)	TIOC2B
	I1BE 位 =1	TIOC2B、TIOC1B

(1) 级联运行的设定步骤例子

级联运行的设定步骤例子如图 11.20 所示。

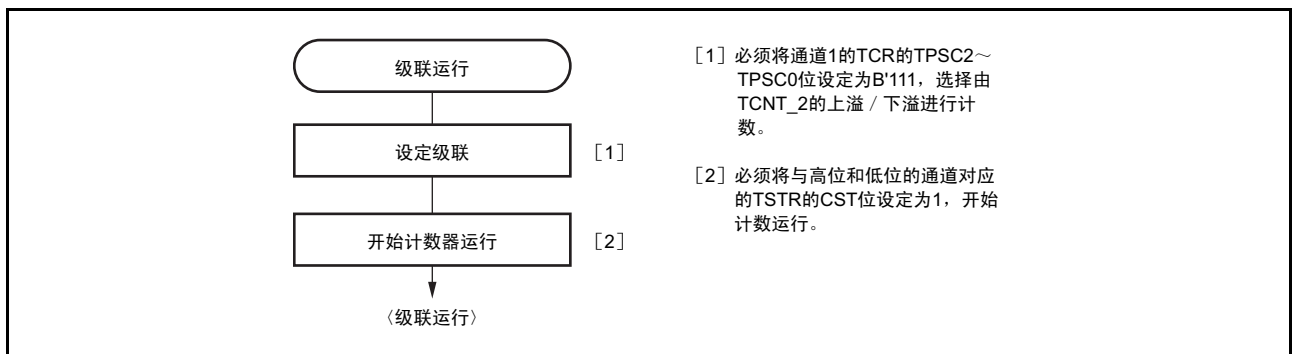


图 11.20 级联运行的设定步骤

(2) 级联运行的例子 (a)

TCNT\_1 通过 TCNT\_2 的上溢 / 下溢进行计数并将通道 2 设定为相位计数模式时的运行如图 11.21 所示。TCNT\_1 通过 TCNT\_2 的上溢进行递增计数并通过 TCNT\_2 的下溢进行递减计数。

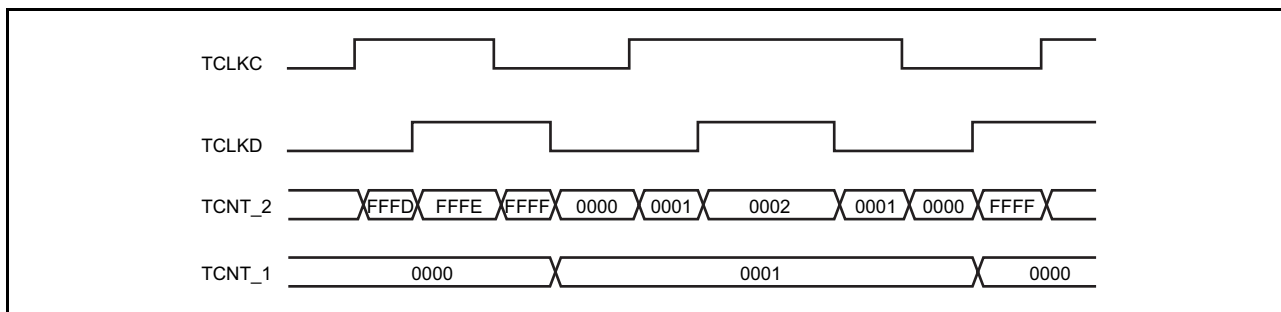


图 11.21 级联的运行例 (a)

(3) 级联运行的例子 (b)

将 TCNT\_1 和 TCNT\_2 进行级联并给 TICCR 的 I2AE 位设定 1 (将 TIOC2A 引脚追加到 TGRA\_1 的输入捕捉条件) 时的运行如图 11.22 所示。在此例中, 将 TIOR\_1 的 IOA0 ~ IOA3 设定为在 TIOC1A 的上升沿进行输入捕捉, 将 TIOR\_2 的 IOA0 ~ IOA3 设定为在 TIOC2A 的上升沿进行输入捕捉。

此时, 将 TIOC1A 和 TIOC2A 的上升沿设定为 TGRA\_1 的输入捕捉条件。另外, TGRA\_2 的输入捕捉条件为 TIOC2A 的上升沿。

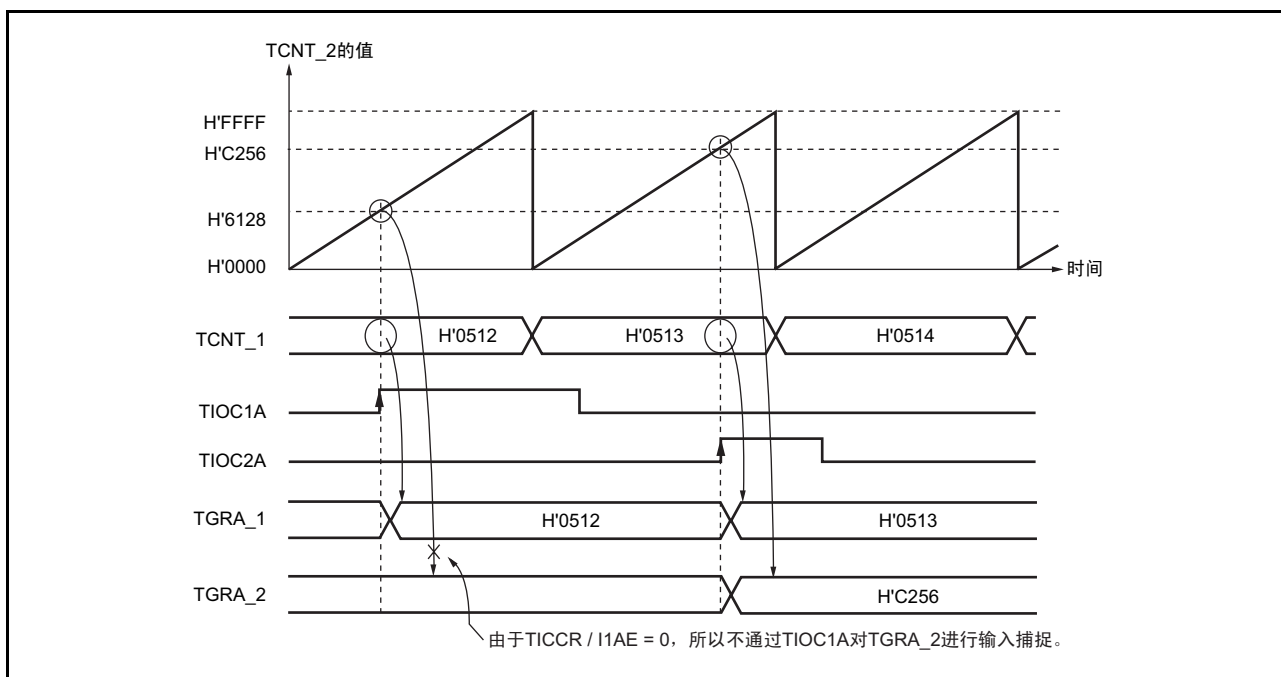


图 11.22 级联的运行例 (b)

(4) 级联运行的例子 (c)

将 TCNT\_1 和 TCNT\_2 进行级联并给 TICCR 的 I2AE 位和 I1AE 位设定 1 (将 TIOC2A 引脚追加到 TGRA\_1 的输入捕捉条件并将 TIOC1A 引脚追加到 TGRA\_2 的输入捕捉条件) 时的运行如图 11.23 所示。在此例中, 将 TIOR\_1 和 TIOR\_2 的 IOA0 ~ IOA3 都设定为在 TIOC1A 和 TIOC2A 的双边沿进行输入捕捉。此时, TIOC1A 和 TIOC2A 输入的 OR 为 TGRA\_1 和 TGRA\_2 的输入捕捉条件。

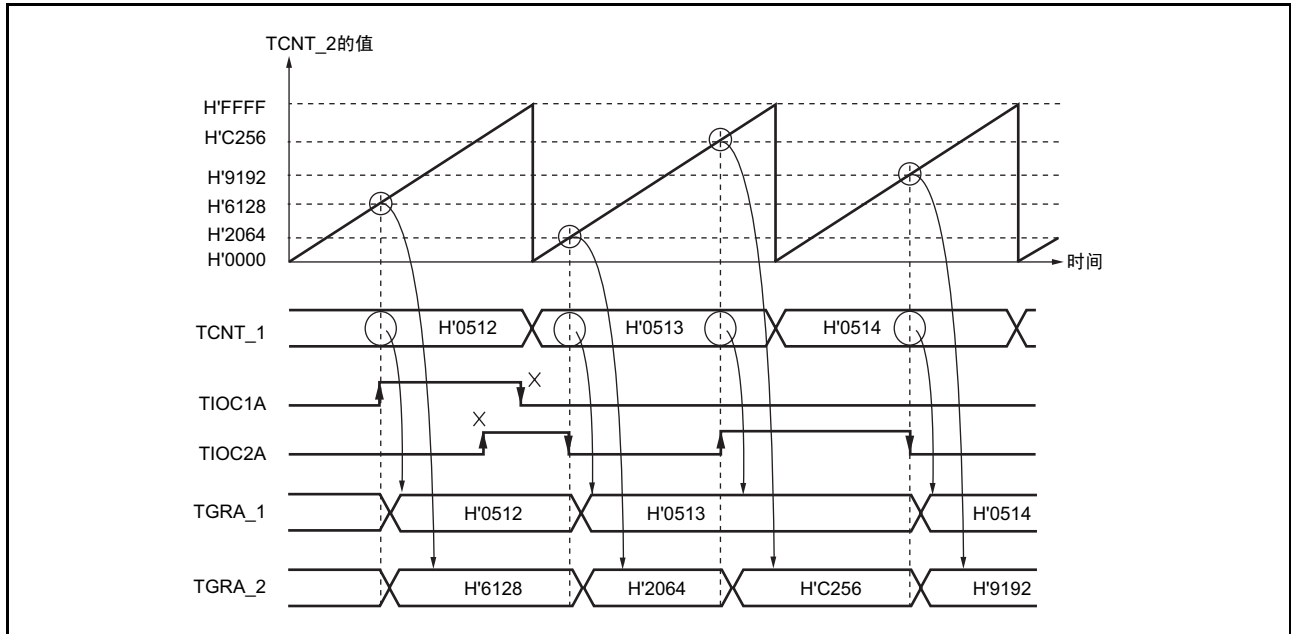


图 11.23 级联的运行例 (c)

## (5) 级联运行的例子 (d)

将 TCNT\_1 和 TCNT\_2 进行级联并给 TICCR 的 I2AE 位设定 1 (将 TIOC2A 引脚追加到 TGRA\_1 的输入捕捉条件) 时的运行如图 11.24 所示。在此例中, 将 TIOR\_1 的 IOA0 ~ IOA3 设定为通过发生 TGRA\_0 的比较匹配进行输入捕捉, 将 TIOR\_2 的 IOA0 ~ IOA3 设定为在 TIOC2A 的上升沿进行输入捕捉。

此时, 因为 TIOR\_1 设定为通过发生 TGRA\_0 的比较匹配 / 输入捕捉进行输入捕捉, 所以即使给 TICCR 的 I2AE 位设定 1, TIOC2A 的边沿也不为 TGRA\_1 的输入捕捉条件。

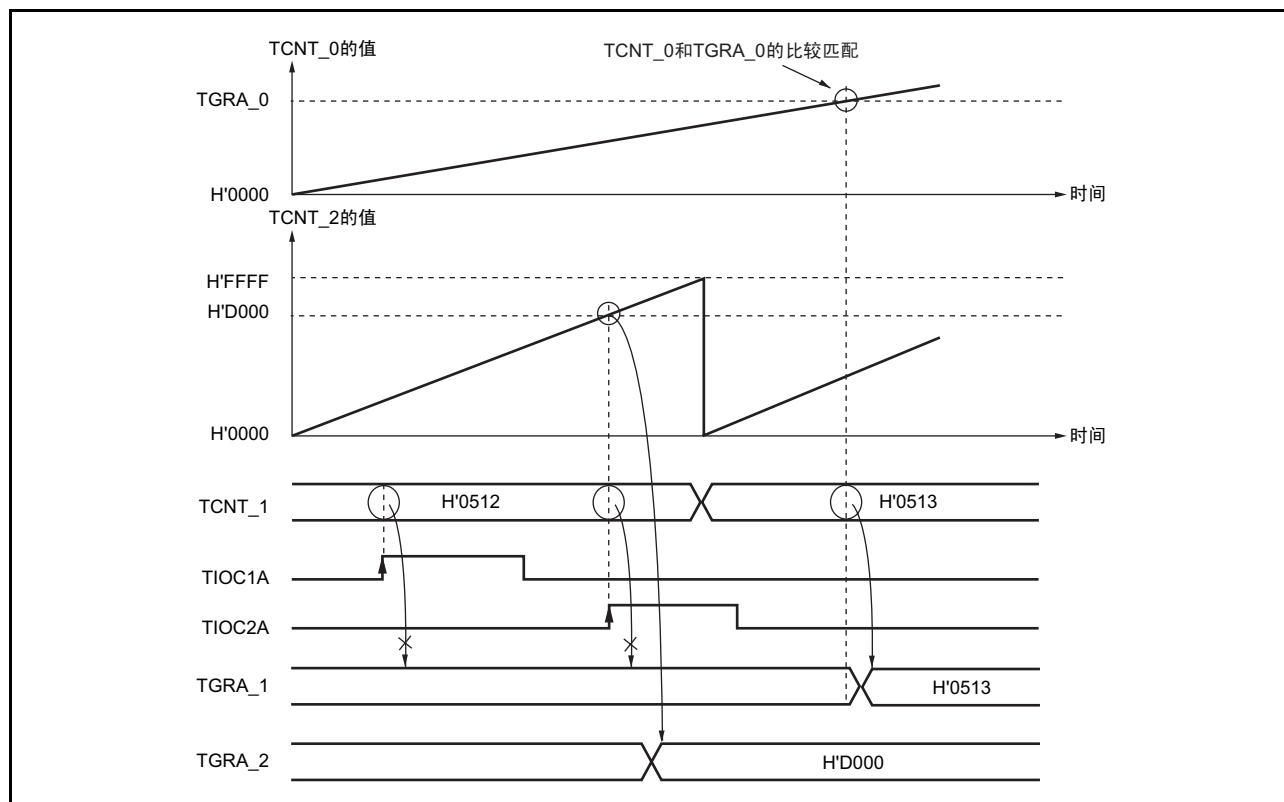


图 11.24 级联的运行例 (d)



### 11.4.5 PWM 模式

PWM 模式是从输出引脚输出各种 PWM 波形的模式。可从 0 输出 / 1 输出 / 交替输出中选择各 TGR 比较匹配的输出现平。

能通过设定各 TGR，输出占空比为 0 ~ 100% 的 PWM 波形。

能通过将 TGR 的比较匹配设定为计数器清除源，给该寄存器设定周期。全部通道能独立设定为 PWM 模式，也能进行同步运行。

PWM 模式有以下 2 种：

#### (a) PWM 模式 1

配对使用 TGRA 和 TGRB、TGRC 和 TGRD，从 TIOCA、TIOCC 引脚生成 PWM 输出。通过比较匹配 A、C 从 TIOCA、TIOCC 引脚进行 TIOR 的 IOA3 ~ IOA0 和 IOC3 ~ IOC0 位指定的输出，并且通过比较匹配 B、D 从 TIOCA、TIOCC 引脚进行 TIOR 的 IOB3 ~ IOB0 和 IOD3 ~ IOD0 位指定的输出。设定在 TGRA、TGRC 的值为初始输出值。如果配对使用的 TGR 设定值相同，即使发生比较匹配，输出值也不变。

在 PWM 模式 1 中能进行最多 8 相的 PWM 输出。

#### (b) PWM 模式 2

将 1 个 TGR 用于周期寄存器，其他 TGR 用于占空比寄存器，生成 PWM 输出。通过比较匹配进行 TIOR 指定的输出。另外，在通过同步寄存器的比较匹配进行计数器清除后，各引脚的输出值为 TIOR 设定的初始值。如果周期寄存器和占空比寄存器的设定值相同，即使发生比较匹配，输出值也不变。

在 PWM 模式 2 中，能通过同步运行的并用来进行最多 8 相的 PWM 输出。

PWM 输出引脚和寄存器的对应如表 11.46 所示。

表 11.46 各 PWM 输出的寄存器和输出引脚

通道	寄存器	输出引脚	
		PWM 模式 1	PWM 模式 2
0	TGRA_0	TIOC0A	TIOC0A
	TGRB_0		TIOC0B
	TGRC_0	TIOC0C	TIOC0C
	TGRD_0		TIOC0D
1	TGRA_1	TIOC1A	TIOC1A
	TGRB_1		TIOC1B
2	TGRA_2	TIOC2A	TIOC2A
	TGRB_2		TIOC2B
3	TGRA_3	TIOC3A	不能设定
	TGRB_3		
	TGRC_3	TIOC3C	
	TGRD_3		
4	TGRA_4	TIOC4A	不能设定
	TGRB_4		
	TGRC_4	TIOC4C	
	TGRD_4		

【注】 在 PWM 模式 2 中，不能对已设定周期的 TGR 进行 PWM 输出。

(1) PWM 模式的设定步骤例子

PWM 模式的设定步骤例子如图 11.25 所示。

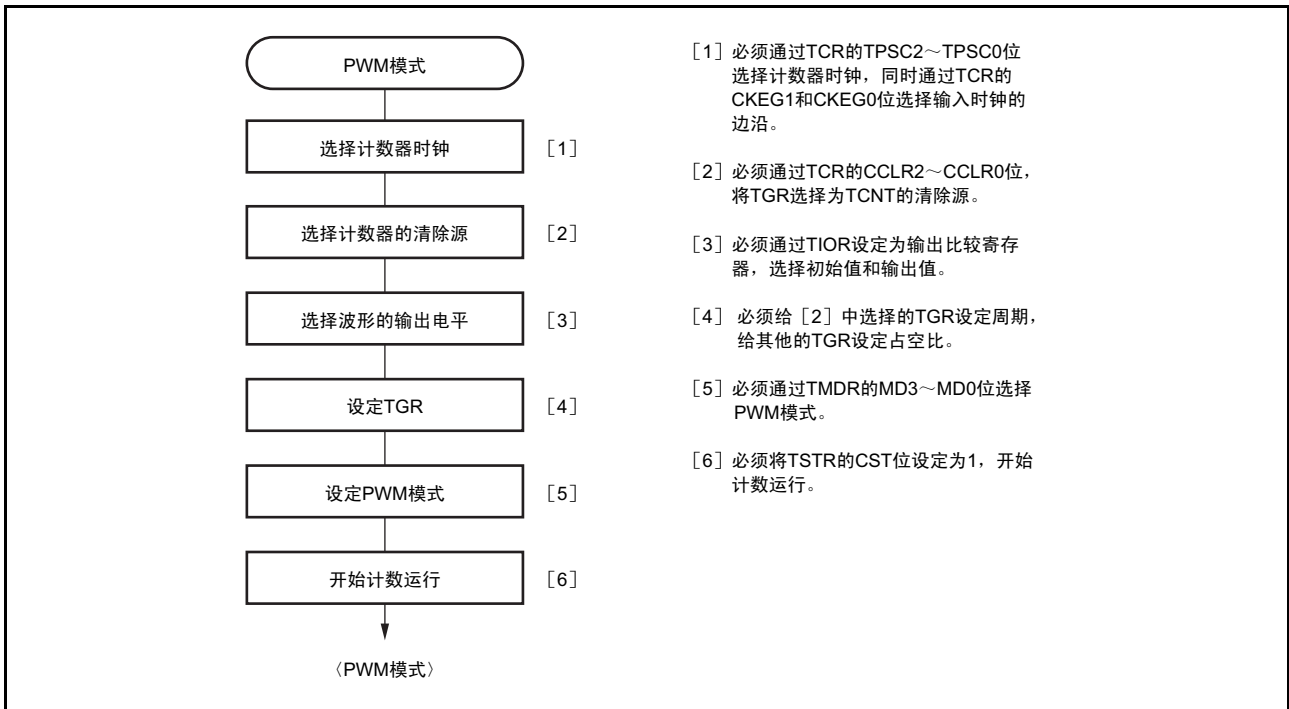


图 11.25 PWM 模式的设定步骤例子

(2) PWM 模式的运行例

PWM1 模式的运行例子如图 11.26 所示。

在此例中，TCNT 的清除源作为 TGRA 的比较匹配，并将 TGRA 的初始输出值和输出值置 0、TGRB 的输出值置 1。

此时，设定在 TGRA 的值为周期，设定在 TGRB 的值为占空比。

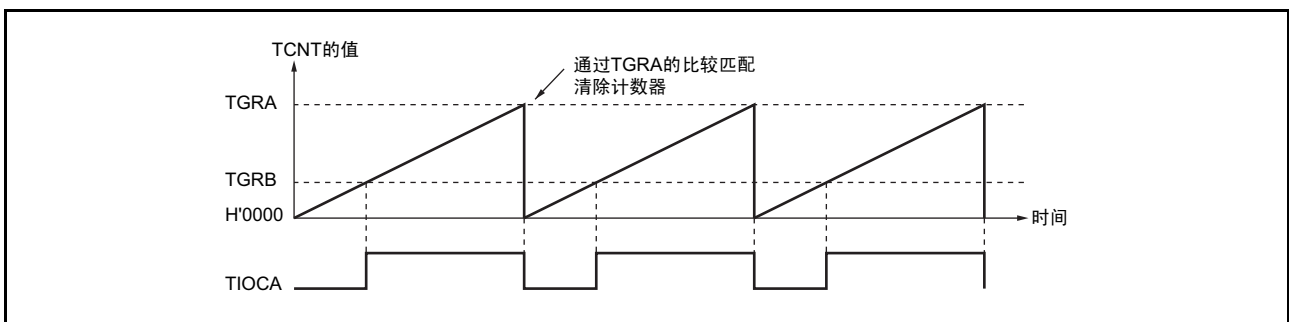


图 11.26 PWM 模式的运行例

PWM 模式 2 的运行例子如图 11.27 所示。

在此例中，通道 0 和 1 同步运行，TCNT 的清除源作为 TGRB\_1 的比较匹配，并将其他的 TGR (TGRA\_0 ~ TGRD\_0、TGRA\_1) 的初始输出值置 0、输出值置 1，输出 5 相的 PWM 波形。

此时，设定在 TGR1B 的值为周期，设定在其他 TGR 的值为占空比。

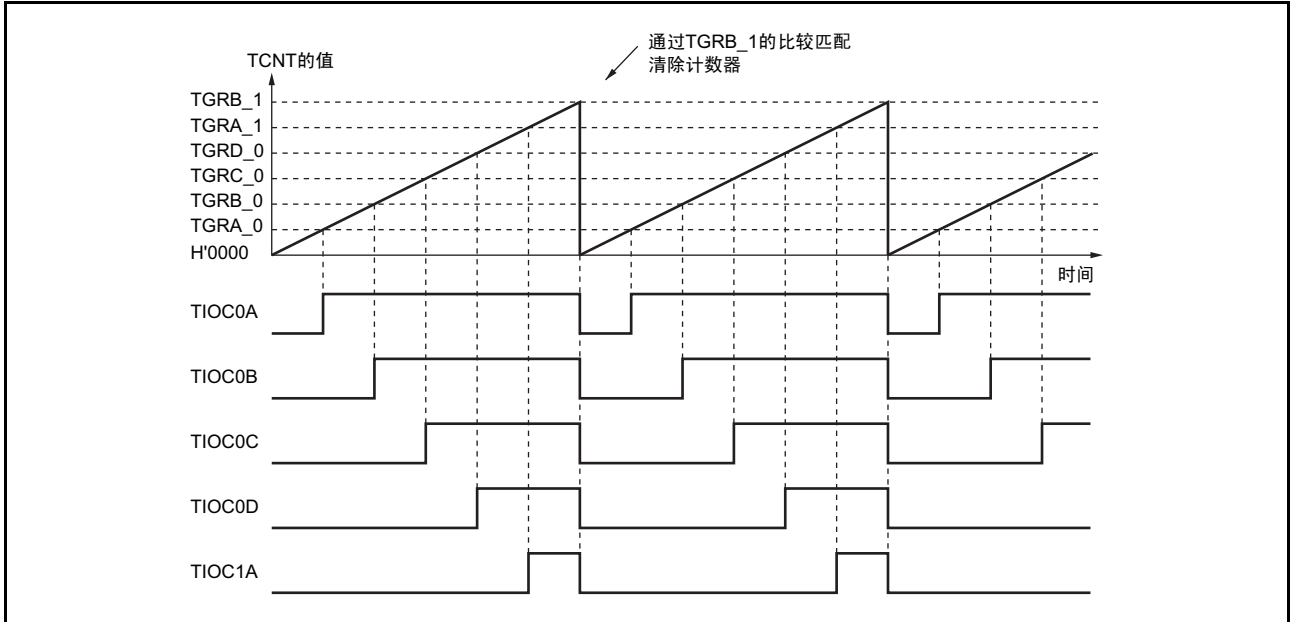


图 11.27 PWM 模式的运行例

在 PWM 模式中，输出占空比为 0% 和占空比为 100% 的 PWM 波形的例子如图 11.28 所示。

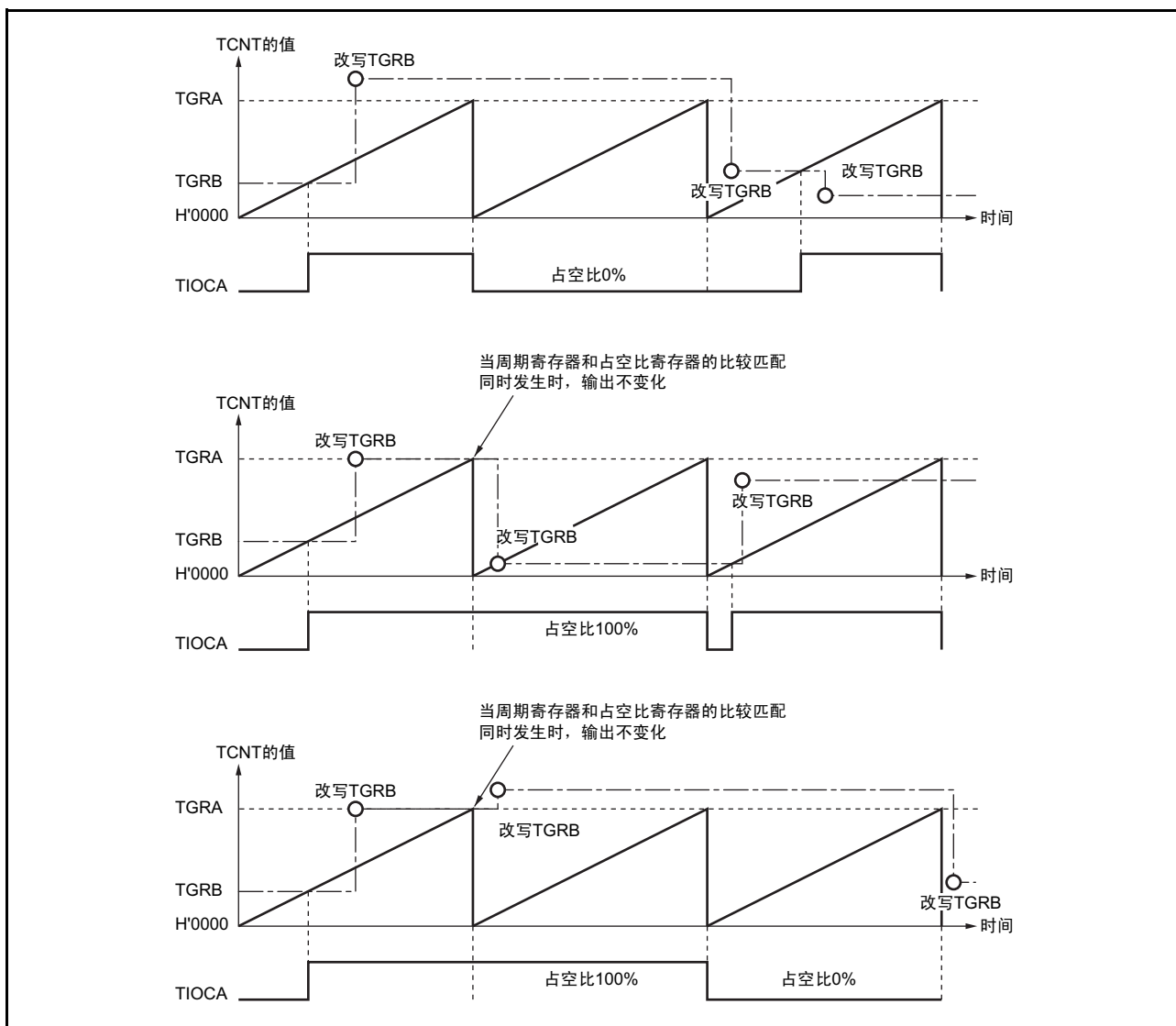


图 11.28 PWM 模式的运行例

### 11.4.6 相位计数模式

相位计数模式通过设定通道 1、2 检测 2 个外部时钟输入的相位差，并将 TCNT 进行递增计数 / 递减计数。

当设定为相位计数模式时，就与 TCR 的 TPSC2 ~ TPSC0 位、CKEG1、CKEG0 位的设定无关，选择外部时钟为计数器输入时钟，TCNT 作为递增 / 递减计数器运行。但是，因为 TCR 的 CCLR1、CCLR0 位、TIOR、TIER、TGR 的功能有效，所以能使用输入捕捉 / 比较匹配功能和中断功能。

能用作 2 相编码器脉冲的输入。

在 TCNT 进行递增计数时，如果发生上溢，TSR 的 TCFV 标志就被置位。在 TCNT 进行递减计数时，如果发生下溢，TCFU 标志就被置位。

TSR 的 TCFD 位是计数方向标志。能通过读 TCFD 标志，确认 TCNT 是在进行递增计数还是在进行递减计数。

外部时钟引脚和通道的对应如表 11.47 所示。

表 11.47 相位计数模式时钟的输入引脚

通道	外部时钟引脚	
	A 相	B 相
将通道 1 设定为相位计数模式	TCLKA	TCLKB
将通道 2 设定为相位计数模式	TCLKC	TCLKD

#### (1) 相位计数模式的设定步骤例子

相位计数模式的设定步骤例子如图 11.29 所示。

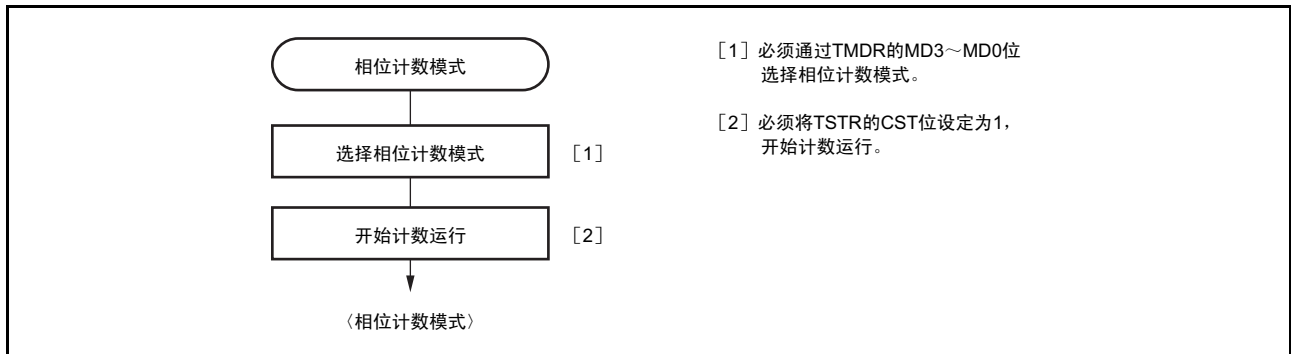


图 11.29 相位计数模式的设定步骤例子

(2) 相位计数模式的运行例

在相位计数模式中，TCNT 通过 2 个外部时钟的相位差进行递增计数 / 递减计数。根据计数条件，有 4 种模式。

(a) 相位计数模式 1

相位计数模式 1 的运行例子如图 11.30 所示，TCNT 递增计数 / 递减计数的条件如表 11.48 所示。

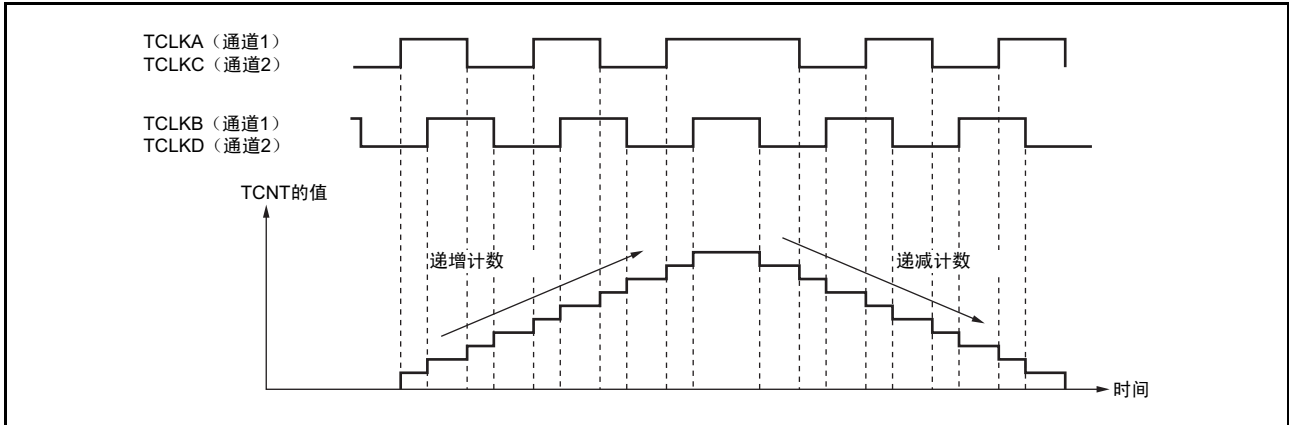


图 11.30 相位计数模式 1 的运行例

表 11.48 相位计数模式 1 的递增计数 / 递减计数的条件

TCLKA (通道 1) TCLKC (通道 2)	TCLKB (通道 1) TCLKD (通道 2)	运行内容
High 电平	上升沿	递增计数
Low 电平	下降沿	
上升沿	Low 电平	
下降沿	High 电平	
High 电平	下降沿	递减计数
Low 电平	上升沿	
上升沿	High 电平	
下降沿	Low 电平	

【符号说明】

上升沿

下降沿

(b) 相位计数模式 2

相位计数模式 2 的运行例子如图 11.31 所示，TCNT 递增计数 / 递减计数的条件如表 11.49 所示。

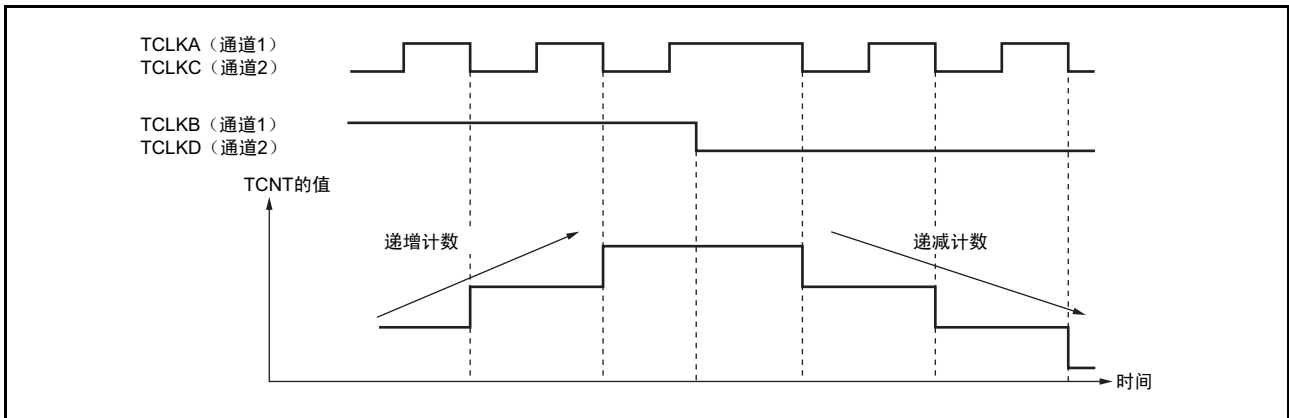


图 11.31 相位计数模式 2 的运行例

表 11.49 相位计数模式 2 的递增计数 / 递减计数的条件

TCLKA (通道 1) TCLKC (通道 2)	TCLKB (通道 1) TCLKD (通道 2)	运行内容
High 电平		不计数 (Don't care)
Low 电平		不计数 (Don't care)
	Low 电平	不计数 (Don't care)
	High 电平	递增计数
High 电平		不计数 (Don't care)
Low 电平		不计数 (Don't care)
	High 电平	不计数 (Don't care)
	Low 电平	递减计数

【符号说明】

: 上升沿

: 下降沿



(c) 相位计数模式 3

相位计数模式 3 的运行例子如图 11.32 所示，TCNT 递增计数 / 递减计数的条件如表 11.50 所示。

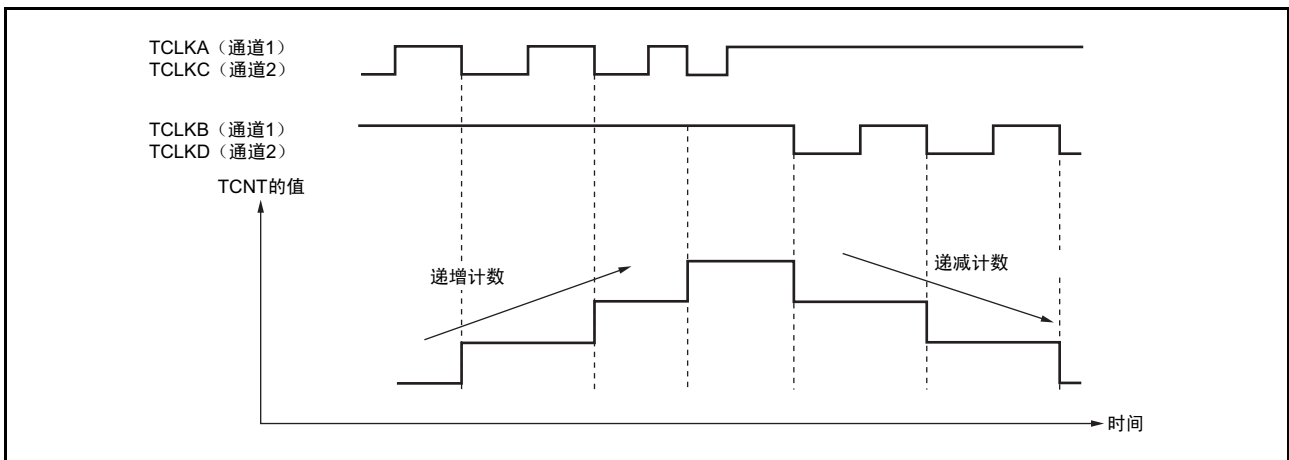


图 11.32 相位计数模式 3 的运行例

表 11.50 相位计数模式 3 的递增计数 / 递减计数的条件

TCLKA (通道 1) TCLKC (通道 2)	TCLKB (通道 1) TCLKD (通道 2)	运行内容
High 电平	上升沿	不计数 (Don't care)
Low 电平	下降沿	不计数 (Don't care)
上升沿	Low 电平	不计数 (Don't care)
下降沿	High 电平	递增计数
High 电平	下降沿	递减计数
Low 电平	上升沿	不计数 (Don't care)
上升沿	High 电平	不计数 (Don't care)
下降沿	Low 电平	不计数 (Don't care)

【符号说明】

上升沿

下降沿

(d) 相位计数模式 4

相位计数模式 4 的运行例如图 11.33 所示，TCNT 递增计数 / 递减计数的条件如表 11.51 所示。

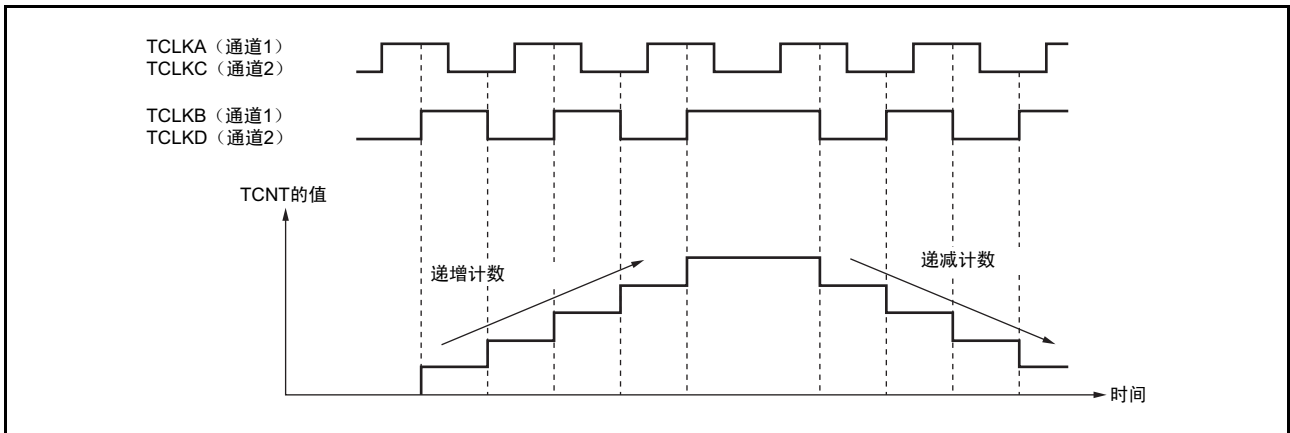


图 11.33 相位计数模式 4 的运行例

表 11.51 相位计数模式 4 的递增计数 / 递减计数的条件

TCLKA (通道 1) TCLKC (通道 2)	TCLKB (通道 1) TCLKD (通道 2)	运行内容
High 电平	上升沿	递增计数
Low 电平	下降沿	
上升沿	Low 电平	不计数 (Don't care)
下降沿	High 电平	
High 电平	下降沿	递减计数
Low 电平	上升沿	
上升沿	High 电平	不计数 (Don't care)
下降沿	Low 电平	

【符号说明】

上升沿

下降沿

## (3) 相位计数模式的应用例

将通道 1 设定为相位计数模式，并与通道 0 配合输入伺服马达的 2 相编码器脉冲后检测位置或者速度的例子如图 11.34 所示。

将通道 1 设定为相位计数模式 1，给 TCLKA 和 TCLKB 输入编码器脉冲的 A 相和 B 相。

通道 0 通过 TGRC\_0 的比较匹配清除 TCNT 计数器， TGRA\_0 和 TGRC\_0 用于比较匹配功能，并设定速度控制周期和位置控制周期。 TGRB\_0 用于输入捕捉功能，并使 TGRB\_0 和 TGRD\_0 进行缓冲运行。将 TGRB\_0 的输入捕捉源作为通道 1 的计数器输入时钟，检测 2 相译码器的 4 倍增脉冲的脉宽。

将通道 1 的 TGRA\_1 和 TGRB\_1 设定为输入捕捉功能，输入捕捉源选择通道 0 的 TGRA\_0 和 TGRC\_0 的比较匹配，并保存各控制周期时的递增 / 递减计数器的值。

用此方法能检测正确的位置 / 速度。

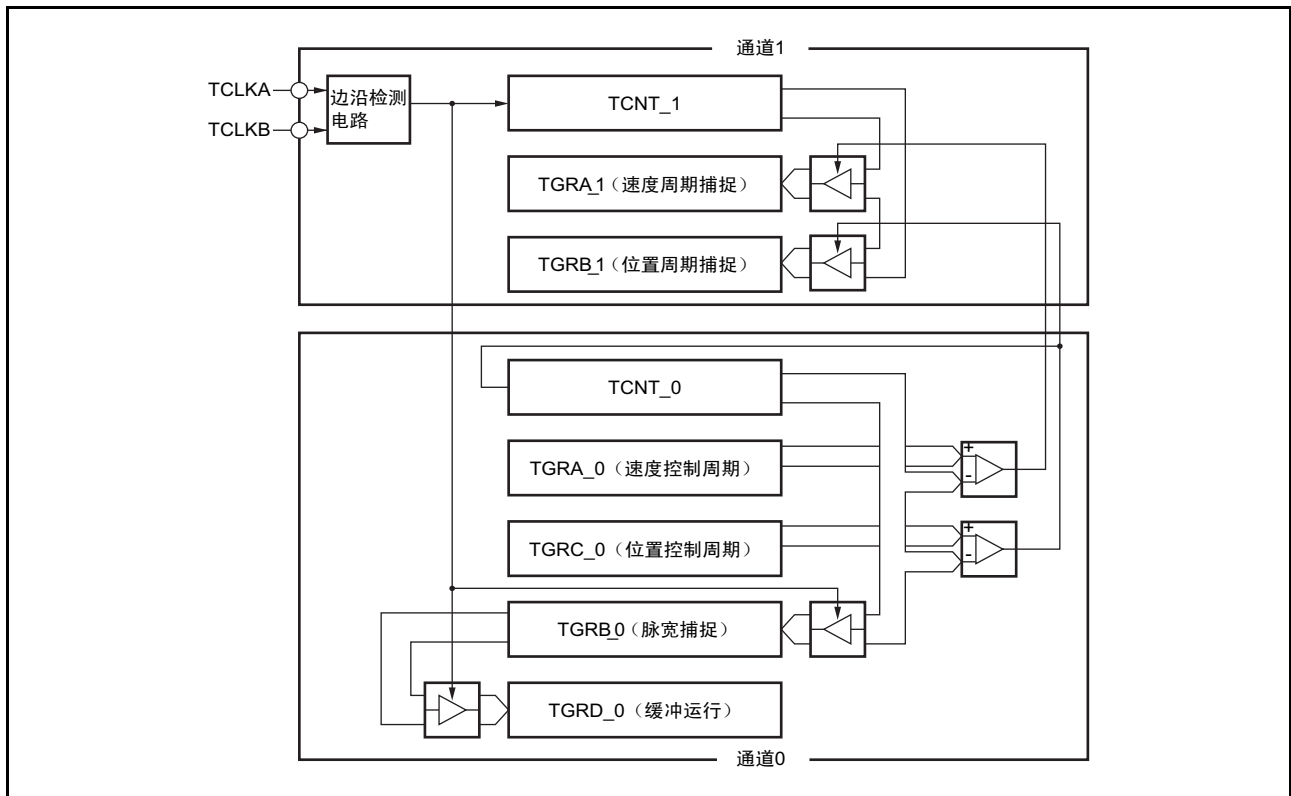


图 11.34 相位计数模式的应用例

### 11.4.7 复位同步 PWM 模式

在复位同步 PWM 模式中，通过组合通道 3、4 将共享同一波形转换点的 PWM 波形（正相和反相）进行 3 相输出。

当设定为复位同步 PWM 模式时，TIOC3B、TIOC3D、TIOC4A、TIOC4C、TIOC4B 和 TIOC4D 引脚为 PWM 输出引脚，定时器的计数器 3 (TCNT\_3) 用作递增计数器。

使用的 PWM 输出引脚和寄存器设定分别如表 11.52 和表 11.53 所示。

表 11.52 复位同步 PWM 模式中的输出引脚

通道	输出引脚	说明
3	TIOC3B	PWM 输出引脚 1
	TIOC3D	PWM 输出引脚 1' (PWM 输出 1 的反相波形)
4	TIOC4A	PWM 输出引脚 2
	TIOC4C	PWM 输出引脚 2' (PWM 输出 2 的反相波形)
	TIOC4B	PWM 输出引脚 3
	TIOC4D	PWM 输出引脚 3' (PWM 输出 3 的反相波形)

表 11.53 复位同步 PWM 模式中的寄存器设定

寄存器	设定内容
TCNT_3	初始设定 H'0000
TCNT_4	初始设定 H'0000
TGRA_3	设定 TCNT_3 的计数周期
TGRB_3	设定从 TIOC3B、TIOC3D 引脚输出的 PWM 波形的变化点
TGRA_4	设定从 TIOC4A、TIOC4C 引脚输出的 PWM 波形的变化点
TGRB_4	设定从 TIOC4B、TIOC4D 引脚输出的 PWM 波形的变化点

(1) 复位同步 PWM 模式的设定步骤例子

复位同步 PWM 模式的设定步骤例子如图 11.35 所示。

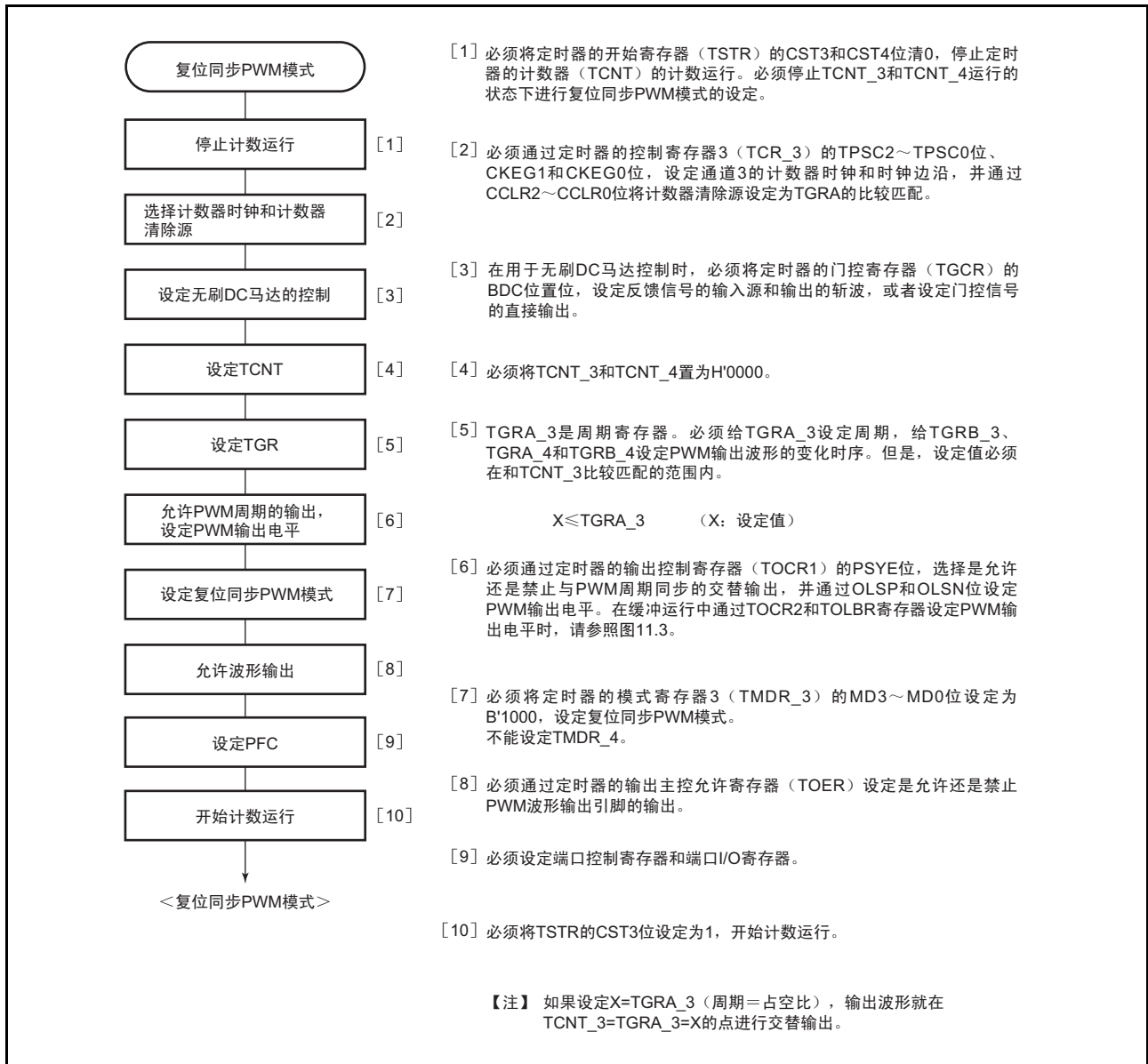


图 11.35 复位同步 PWM 模式的设定步骤例子

## (2) 复位同步 PWM 模式的运行例

复位同步 PWM 模式的运行例子如图 11.36 所示。

在复位同步 PWM 模式中，TCNT\_3 和 TCNT\_4 作为递增计数器运行。如果 TCNT\_3 和 TGRA\_3 比较匹配，就清除计数器，并从 H'0000 重新开始递增计数。每当发生各 TGRB\_3、TGRA\_4、TGRB\_4 的比较匹配和计数器清除时，PWM 输出引脚就进行交替输出。

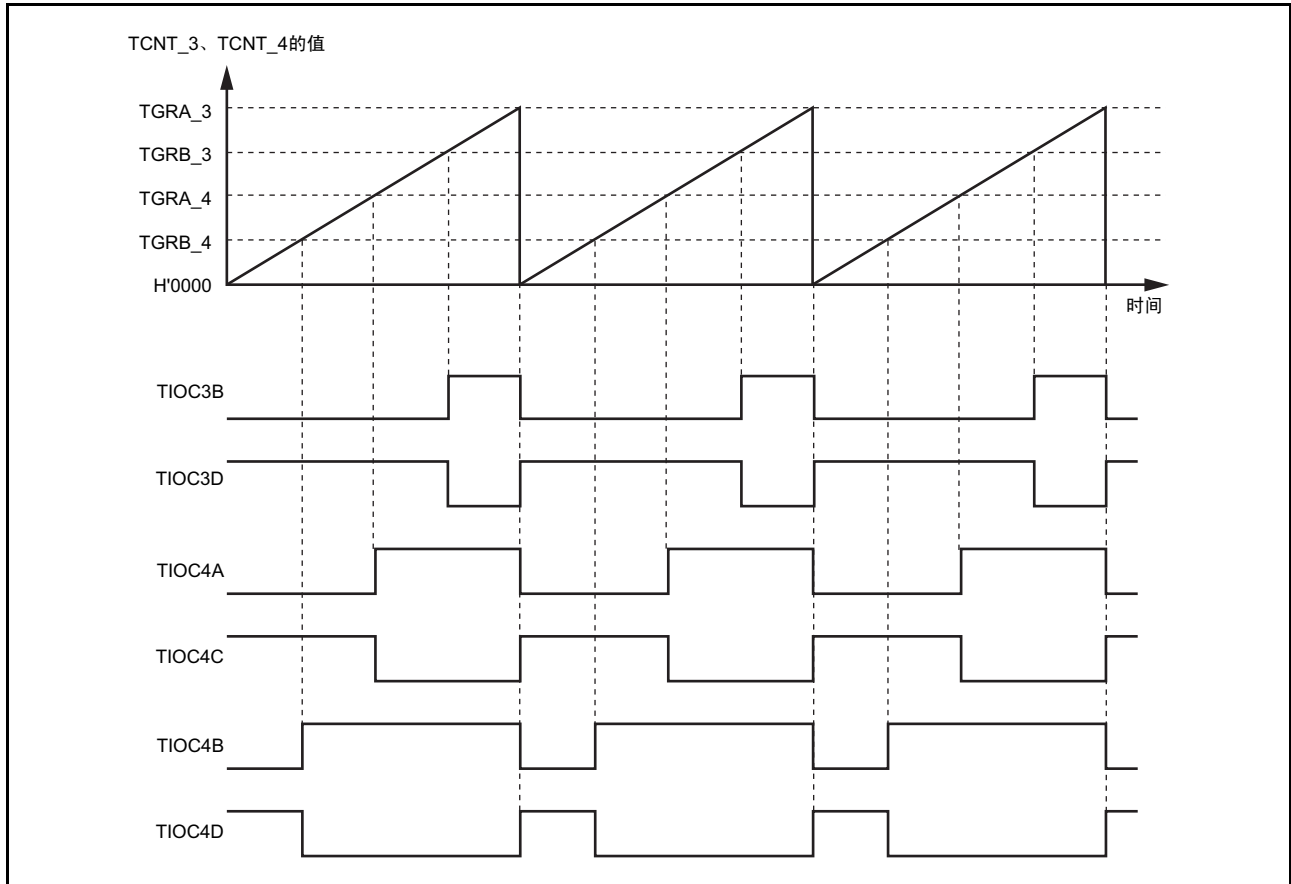


图 11.36 复位同步 PWM 模式的运行例（当设定 TOCR 的 OLSN=1、OLSP=1 时）

### 11.4.8 互补 PWM 模式

在互补 PWM 模式中，通过组合通道 3、4 将正相和反相为非重叠关系的 PWM 波形进行 3 相输出，也能设定没有非重叠时间。

当设定为互补 PWM 模式时，TIOC3B、TIOC3D、TIOC4A、TIOC4B、TIOC4C、TIOC4D 引脚为 PWM 输出引脚，能将 TIOC3A 引脚设定为与 PWM 周期同步的交替输出。

另外，TCNT\_3 和 TCNT\_4 用作递增 / 递减计数器。

使用的 PWM 输出引脚和寄存器设定分别如表 11.54 和表 11.55 所示。

作为端口功能，支持通过外部信号将 PWM 输出进行直接 OFF 的功能。

表 11.54 互补 PWM 模式中的输出引脚

通道	输出引脚	说明
3	TIOC3A	与 PWM 周期同步的交替输出 (或者输入 / 输出端口)
	TIOC3B	PWM 输出引脚 1
	TIOC3C	输入 / 输出端口 *
	TIOC3D	PWM 输出引脚 1' (PWM 输出 1 的非重叠关系的反相波形。也能设定没有非重叠时间)
4	TIOC4A	PWM 输出引脚 2
	TIOC4C	PWM 输出引脚 2' (PWM 输出 2 的非重叠关系的反相波形。也能设定没有非重叠时间)
	TIOC4B	PWM 输出引脚 3
	TIOC4D	PWM 输出引脚 3' (PWM 输出 3 的非重叠关系的反相波形。也能设定没有非重叠时间)

【注】 \* 在互补 PWM 模式中，不能将 TIOC3C 引脚设定为定时器的输入 / 输出引脚。

表 11.55 互补 PWM 模式中的寄存器设定

通道	计数器 / 寄存器	说明	由 CPU 进行读写
3	TCNT_3	从空载时间寄存器的设定值开始递增计数	能通过设定 *TRWER 进行屏蔽
	TGRA_3	设定 TCNT_3 的上限值 (1/2 的载波周期 + 空载时间)	能通过设定 *TRWER 进行屏蔽
	TGRB_3	PWM 输出 1 的比较寄存器	能通过设定 *TRWER 进行屏蔽
	TGRC_3	TGRA_3 的缓冲寄存器	总是能进行读写
	TGRD_3	PWM 输出 1/TGRB_3 的缓冲寄存器	总是能进行读写
4	TCNT_4	初始设定 H'0000 并开始递增计数	能通过设定 *TRWER 进行屏蔽
	TGRA_4	PWM 输出 2 的比较寄存器	能通过设定 *TRWER 进行屏蔽
	TGRB_4	PWM 输出 3 的比较寄存器	能通过设定 *TRWER 进行屏蔽
	TGRC_4	PWM 输出 2/TGRA_4 的缓冲寄存器	总是能进行读写
	TGRD_4	PWM 输出 3/TGRB_4 的缓冲寄存器	总是能进行读写
定时器的空载时间数据寄存器 (TDDR)		设定 TCNT_4 和 TCNT_3 的偏移值 (空载时间的值)	能通过设定 *TRWER 进行屏蔽
定时器的周期数据寄存器 (TCDR)		设定 TCNT_4 的上限值 (1/2 的载波周期)	能通过设定 *TRWER 进行屏蔽
定时器的周期缓冲寄存器 (TCBR)		TCDR 的缓冲寄存器	总是能进行读写
子计数器 (TCNTS)		生成空载时间的子计数器	只能读
暂存器 1 (TEMP1)		PWM 输出 1/TGRB_3 的暂存器	不能进行读写
暂存器 2 (TEMP2)		PWM 输出 2/TGRA_4 的暂存器	不能进行读写
暂存器 3 (TEMP3)		PWM 输出 3/TGRB_4 的暂存器	不能进行读写

【注】 \* 能通过设定 TRWER (定时器的读写允许寄存器)，允许或者禁止存取。





(1) 互补 PWM 模式的设定步骤例子

互补 PWM 模式的设定步骤例子如图 11.38 所示。



图 11.38 互补 PWM 模式的设定步骤例子

## (2) 互补 PWM 模式的运行概要

在互补 PWM 模式中能进行 6 相的 PWM 输出。互补 PWM 模式的计数器运行和运行例分别如图 11.39 和图 11.40 所示。

## (a) 计数器运行

在互补 PWM 模式中，TCNT\_3、TCNT\_4 和 TCNTS（3 个计数器）进行递增计数。

当 TCNT\_3 被设定为互补 PWM 模式并且 TSTR 的 CST 位是 0 时，设定在 TDDR 的值就自动置为初始值。

当 CST 位被置 1 时，TCNT\_3 就进行递增计数（计数到 TGRA\_3 的设定值为止）。当 TCNT\_3 和 TGRA\_3 一致时，TCNT\_3 就转换为递减计数；当 TCNT\_3 和 TDDR 一致时，TCNTS 又转换为递增计数，重复此运行。

另外，将 TCNT\_4 的初始值设定为 H'0000。

当 CST 位被置 1 时，TCNT\_4 就与 TCNT\_3 同步进行递增计数。当 TCNT\_4 和 TCDR 一致时，TCNT\_4 就转换为递减计数；当 TCNT\_4 计数到 H'0000 时，TCNT\_4 又转换为递增计数，重复此运行。

TCNTS 是只读计数器，不需要设定初始值。

在 TCNT\_3、4 进行递增计数时，如果 TCNT\_3 和 TCDR 一致，TCNTS 就开始递减计数。当 TCNTS 和 TCDR 一致时，TCNTS 就转换为递增计数；当 TCNTS 和 TGRA\_3 一致时，TCNTS 就被清除为 H'0000。

在 TCNT\_3、TCNT\_4 进行递减计数时，如果 TCNT\_4 和 TDDR 一致，TCNTS 就开始递增计数。当 TCNTS 和 TDDR 一致时，TCNTS 就转换为递减计数；当 TCNTS 计数到 H'0000 时，TCNTS 就被设定为 TGRA\_3 的值。

TCNTS 只在计数运行的期间与设定 PWM 占空比的比较寄存器和暂存器比较。

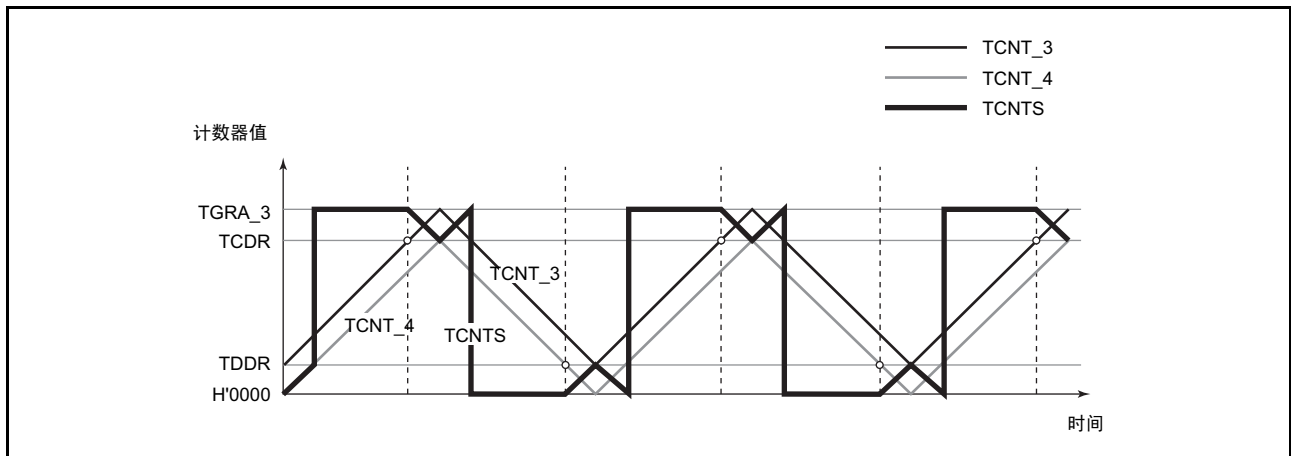


图 11.39 互补 PWM 模式的计数器运行

## (b) 寄存器运行

在互补 PWM 模式中使用比较寄存器、缓冲寄存器和暂存器等 9 个寄存器。互补 PWM 模式的运行例子如图 11.40 所示。

为了进行 PWM 输出，寄存器 TGRB\_3、TGRA\_4、TGRB\_4 总是和计数器比较。当这些寄存器和计数器一致时，定时器的输出控制寄存器 (TOCR) 的 OLSN、OLSP 位设定的值就被输出。

这些比较寄存器的缓冲寄存器为 TGRD\_3、TGRC\_4、TGRD\_4。

另外，在缓冲寄存器和比较寄存器之间有暂存器，但是不能从 CPU 存取暂存器。

要更改比较寄存器的数据时，必须给对应的缓冲寄存器写要更改的数据。缓冲寄存器总是能读写。

在 Ta 区间总是将写到缓冲寄存器的数据传送到暂存器，而在 Tb 区间不传送到暂存器。在 Tb 区间结束后，将在此区间已写到缓冲寄存器的数据传送到暂存器。

如果 Tb 区间结束的 TCNTS 在进行递增计数时和 TGRA\_3 一致或者在进行递减计数时计数到 H'0000，就将已传送到暂存器的值传送到比较寄存器。能通过定时器的模式寄存器 (TMDR) 的 MD3 ~ MD0 位选择此暂存器到比较寄存器的传送时序。图 11.40 是选择在波谷更改的模式例子。

在不向暂存器传送数据的 Tb (图 11.40 中的 Tb1) 区间，暂存器具有和比较寄存器相同的功能，和计数器比较。在此区间，对于 1 相的输出有 2 个比较匹配寄存器，比较寄存器存有更改前的数据，暂存器存有新更改后的数据。而且在此区间，TCNT\_3、4 和 TCNTS (3 个计数器) 与比较寄存器和暂存器 (2 个寄存器) 比较，控制 PWM 输出。

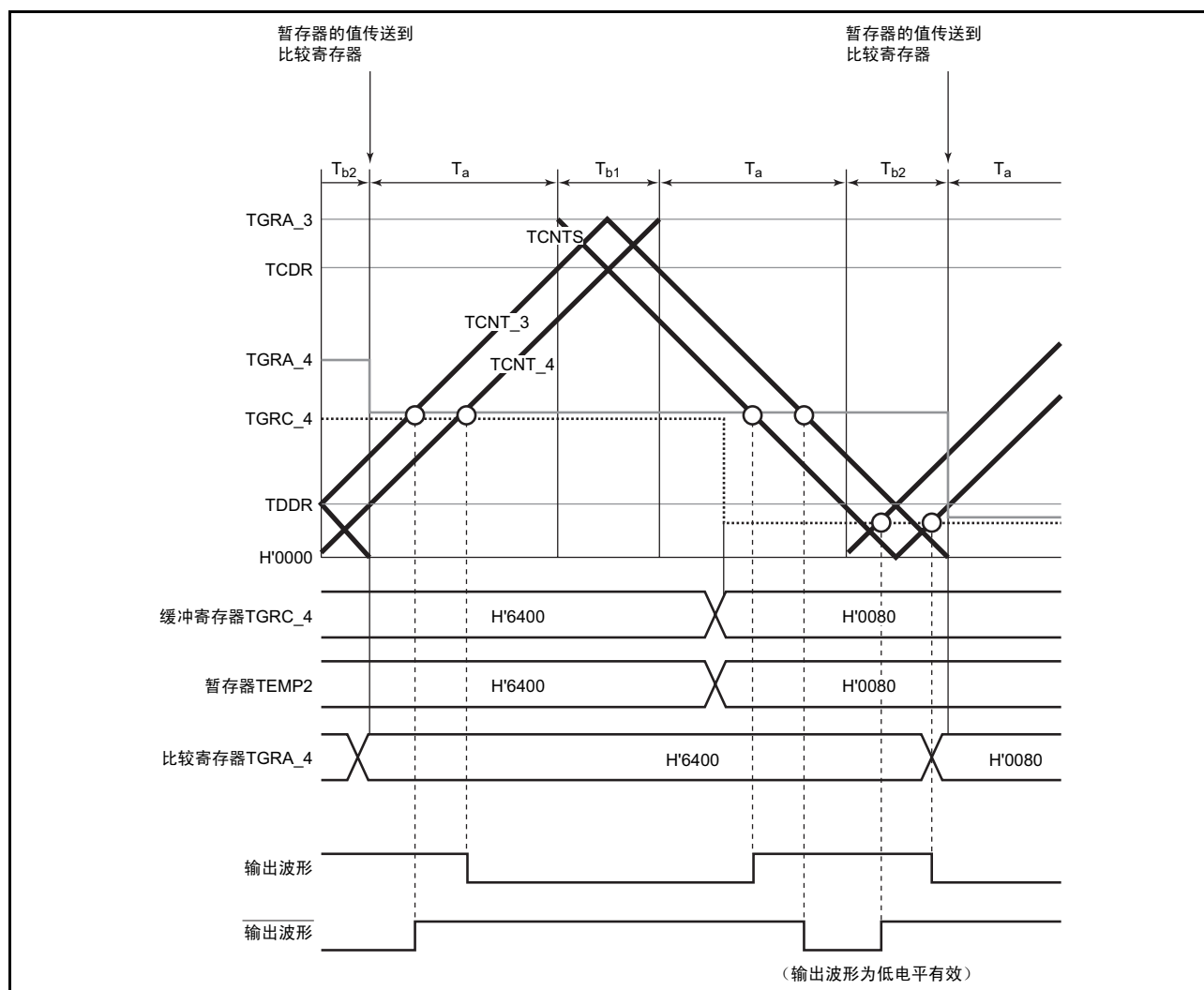


图 11.40 互补 PWM 模式的运行例

**(c) 初始设定**

在互补 PWM 模式中，有 6 个需要初始设定的寄存器，还有 1 个设定是否生成空载时间的寄存器（只能在不生成空载时间的情况下进行设定）。

在通过定时器的模式寄存器（TMDR）的 MD3 ~ MD0 位设定为互补 PWM 模式前，必须设定以下寄存器的初始值：

TGRC\_3 用作 TGRA\_3 的缓冲寄存器，设定 1/2 的 PWM 载波周期 + 空载时间 Td。定时器的周期缓冲寄存器（TCBR）用作定时器的周期数据寄存器（TCDR）的缓冲寄存器，设定 1/2 的 PWM 载波周期，还给定定时器的空载时间寄存器（TDDR）设定空载时间 Td。

如果不生成空载时间，就将定时器的空载时间允许寄存器（TDER）的 TDER 位设定为 0，给 TGRC\_3 和 TGRA\_3 设定 1/2 的 PWM 载波周期 +1，并且给 TDDR 置 1。

给缓冲寄存器 TGRD\_3、TGRC\_4 和 TGRD\_4（3 个寄存器）分别设定 PWM 占空比的初始值。

在设定为互补 PWM 模式的同时，将 5 个缓冲寄存器（TDDR 除外）的设定值分别传送到对应的比较寄存器。

另外，必须在设定为互补 PWM 模式前将 TCNT\_4 设定为 H'0000。

表 11.56 需要初始设定的寄存器和计数器

寄存器 / 计数器	设定值
TGRC_3	1/2 的 PWM 载波周期 + 空载时间 Td (如果通过 TDER 设定为不生成空载时间，就为 1/2 的 PWM 载波周期 +1)
TDDR	空载时间 Td (如果通过 TDER 设定为不生成空载时间，就为 1)
TCBR	1/2 的 PWM 载波周期
TGRD_3、TGRC_4、TGRD_4	各相 PWM 占空比的初始值
TCNT_4	H'0000

**【注】** TGRC\_3 的设定值必须为设定在 TCBR 的 PWM 载波周期的 1/2 值和设定在 TDDR 的空载时间 Td 的值的和。但是，如果通过 TDER 设定为不生成空载时间，就为 1/2 的 PWM 载波周期 +1。

**(d) PWM 输出电平的设定**

在互补 PWM 模式中，通过定时器的输出控制寄存器 1（TOCR1）的 OLSN、OLSP 位或者定时器的输出控制寄存器 2（TOCR2）的 OLS1P ~ OLS3P、OLS1N ~ OLS3N 位，设定 PWM 脉冲的输出电平。

能按 3 相正相和 3 相反相（6 相输出）设定输出电平。

另外，必须在解除互补 PWM 模式的状态下进行输出电平的设定或者变更。

**(e) 空载时间的设定**

在互补 PWM 模式中，输出正相和反相为非重叠关系的 PWM 脉冲。此非重叠时间称为空载时间。

将非重叠时间设定到定时器的空载时间数据寄存器（TDDR）。设定在 TDDR 的值为 TCNT\_3 计数器的初始值，生成 TCNT\_3 和 TCNT\_4 的非重叠关系。必须在解除互补 PWM 模式的状态下更改 TDDR 的内容。

**(f) 不生成空载时间的设定**

通过将定时器的空载时间允许寄存器（TDER）的 TDER 位设定为 0，进行不生成空载时间的设定。只能在 TDER=1 的状态下读 TDER 后给 TDER 写 0 时，将 TDER 置 0。

给 TGRA\_3、TGRC\_3 设定 1/2 的 PWM 载波周期 +1，并且给定定时器的空载时间数据寄存器（TDDR）置 1。

如果设定为不生成空载时间，就能输出无空载时间的 PWM 波形。不生成空载时间的运行例子如图 11.41 所示。

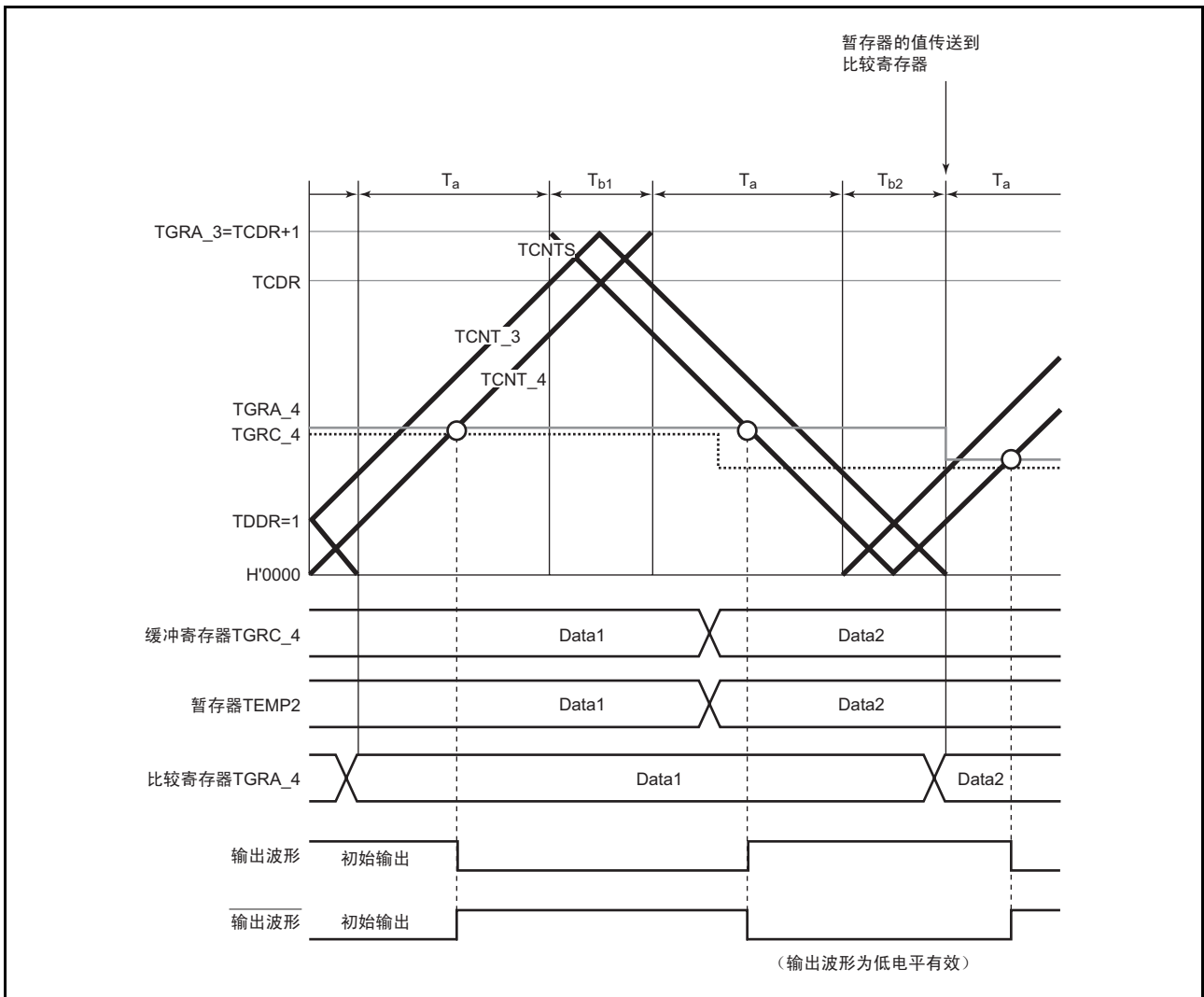


图 11.41 不生成空载时间的运行例

(g) PWM 周期的设定

在互补 PWM 模式中，将 PWM 脉冲的周期设定到 TGRA\_3（设定 TCNT3 的上限值）和 TCDR（设定 TCNT\_4 的上限值）这 2 个寄存器。必须将这 2 个寄存器设定为以下的关系：

生成空载时间：TGRA\_3 的设定值 = TCDR 的设定值 + TDDR 的设定值

不生成空载时间：TGRA\_3 的设定值 = TCDR 的设定值 + 1

另外，必须通过给缓冲寄存器的 TGRC\_3、TCBR 设定值进行 TGRA\_3、TCDR 的设定。在定时器的模式寄存器（TMDR）的 MD3 ~ MD0 选择的传送时序中，将设定在 TGRC\_3 和 TCBR 的值同时传送到 TGRA\_3 和 TCDR。

如果在波峰更新数据，就从下一个周期反映更改的 PWM 周期；如果在波谷更新数据，就从该周期反映更改的 PWM 周期。在波峰更改 PWM 周期时的运行例子如图 11.42 所示。

另外，有关各缓冲寄存器数据的更新方法，请参照以下的“(h) 寄存器数据的更新”。

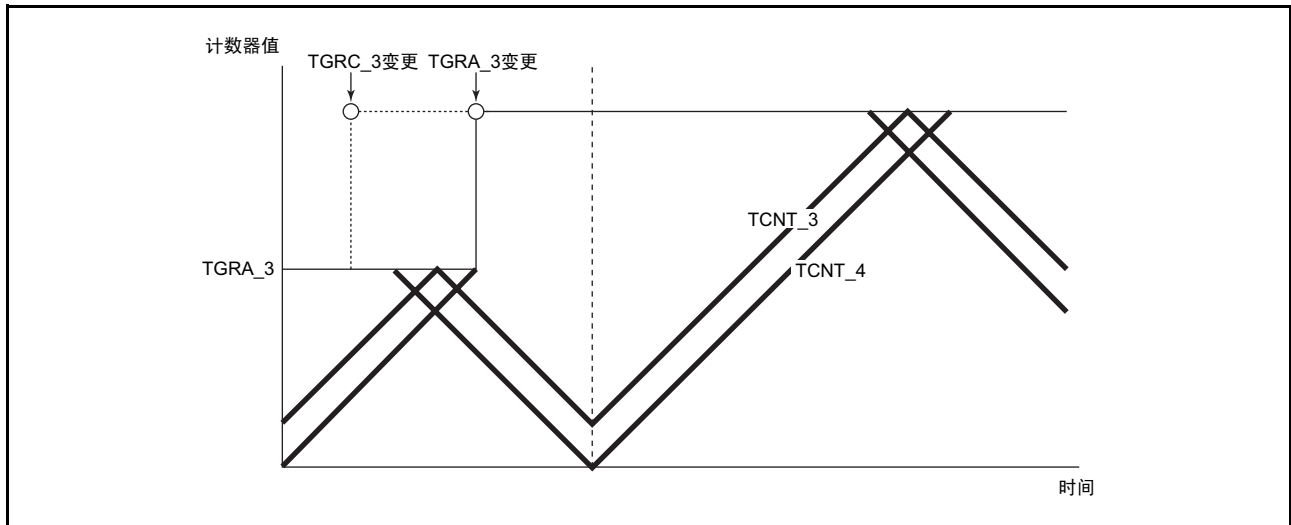


图 11.42 PWM 周期的变更例子

#### (h) 寄存器数据的更新

在互补 PWM 模式中更新比较寄存器的数据时，使用缓冲寄存器。更新的数据随时能写到缓冲寄存器。另外，能在有缓冲寄存器的运行中更改的寄存器为 5 个用于 PWM 占空比和载波周期的寄存器。

在这些寄存器和缓冲寄存器之间有各自的暂存器。在子计数器 TCNTS 不进行计数的期间，如果更新寄存器的数据，暂存器的值也会被改写。TCNTS 在进行计数中，不将缓冲寄存器的值传送到暂存器，而在 TCNTS 停止计数后将写在缓冲寄存器的值传送到暂存器。

在定时器的模式寄存器 (TMDR) 的 MD3 ~ MD0 位设定的数据更新时序中，将暂存器的值传送到比较寄存器。互补 PWM 模式中的数据更新例子如图 11.43 所示，此图是在计数器的波峰和波谷更新数据的模式。

在改写缓冲寄存器的数据时，最后必须写 TGRD\_4。在写 TGRD\_4 后，5 个寄存器全部同时将缓冲寄存器的数据传送到暂存器。

如果不全部更新 5 个寄存器或者不更新 TGRD\_4 的数据，就必须在写要更新的寄存器数据后写 TGRD\_4。此时，写到 TGRD\_4 的数据必须和写之前的数据相同。

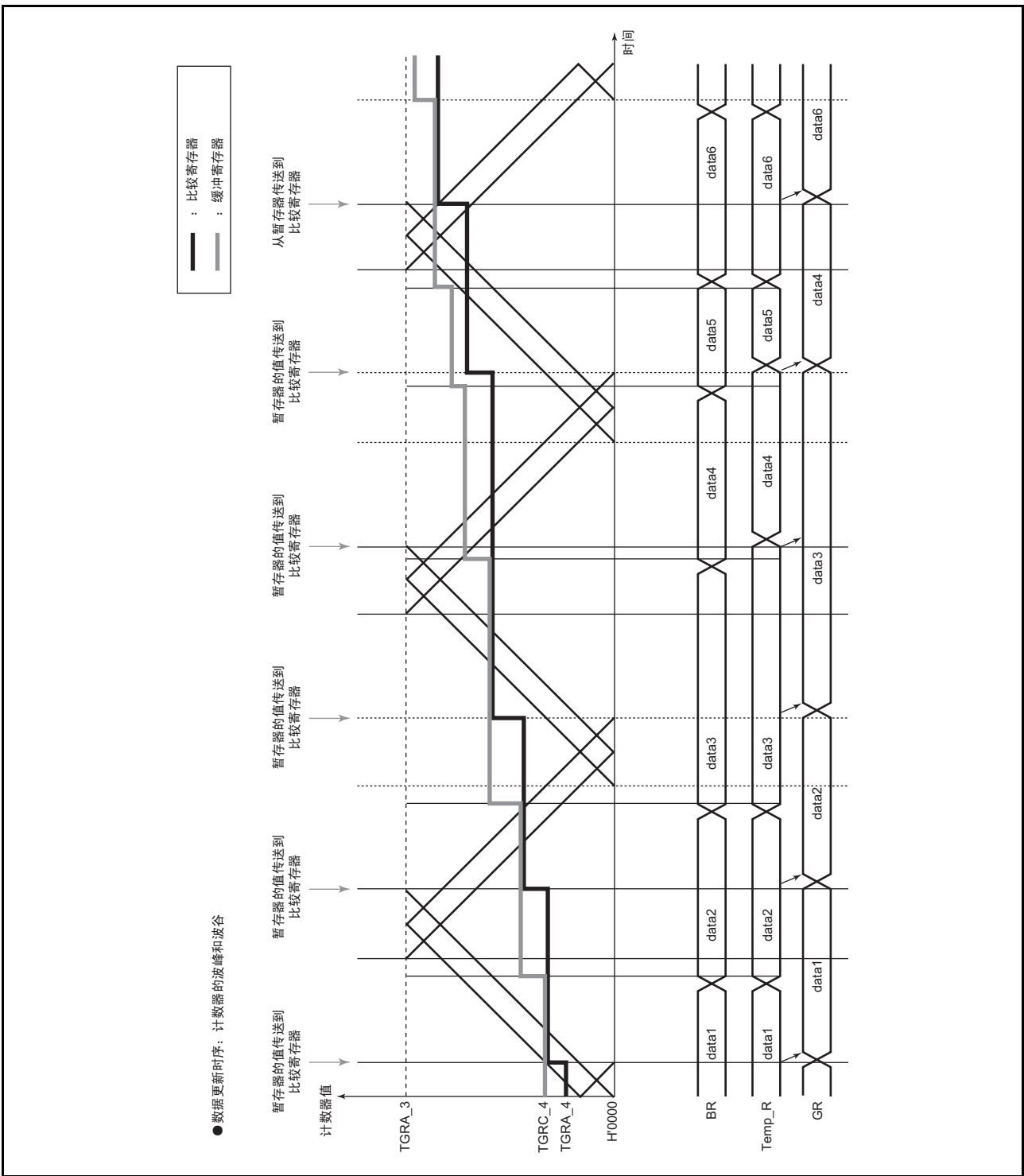


图 11.43 互补 PWM 模式的数据更新例

(i) 互补 PWM 模式的初始输出

在互补 PWM 模式中，通过设定定时器的输出控制寄存器 1 (TOCR1) 的 OLSN、OLSP 位或者定时器的输出控制寄存器 2 (TOCR2) 的 OLS1N ~ OLS3N、OLS1P ~ OLS3P 位，决定初始输出。

此初始输出为 PWM 脉冲的无效电平，从通过定时器的模式寄存器 (TMDR) 设定互补 PWM 模式到 TCNT\_4 大于空载时间寄存器 (TDDR) 的设定值为止，输出此初始输出。互补 PWM 模式的初始输出例子如图 11.44 所示。

另外，PWM 占空比的初始值小于 TDDR 值的波形例子如图 11.45 所示。

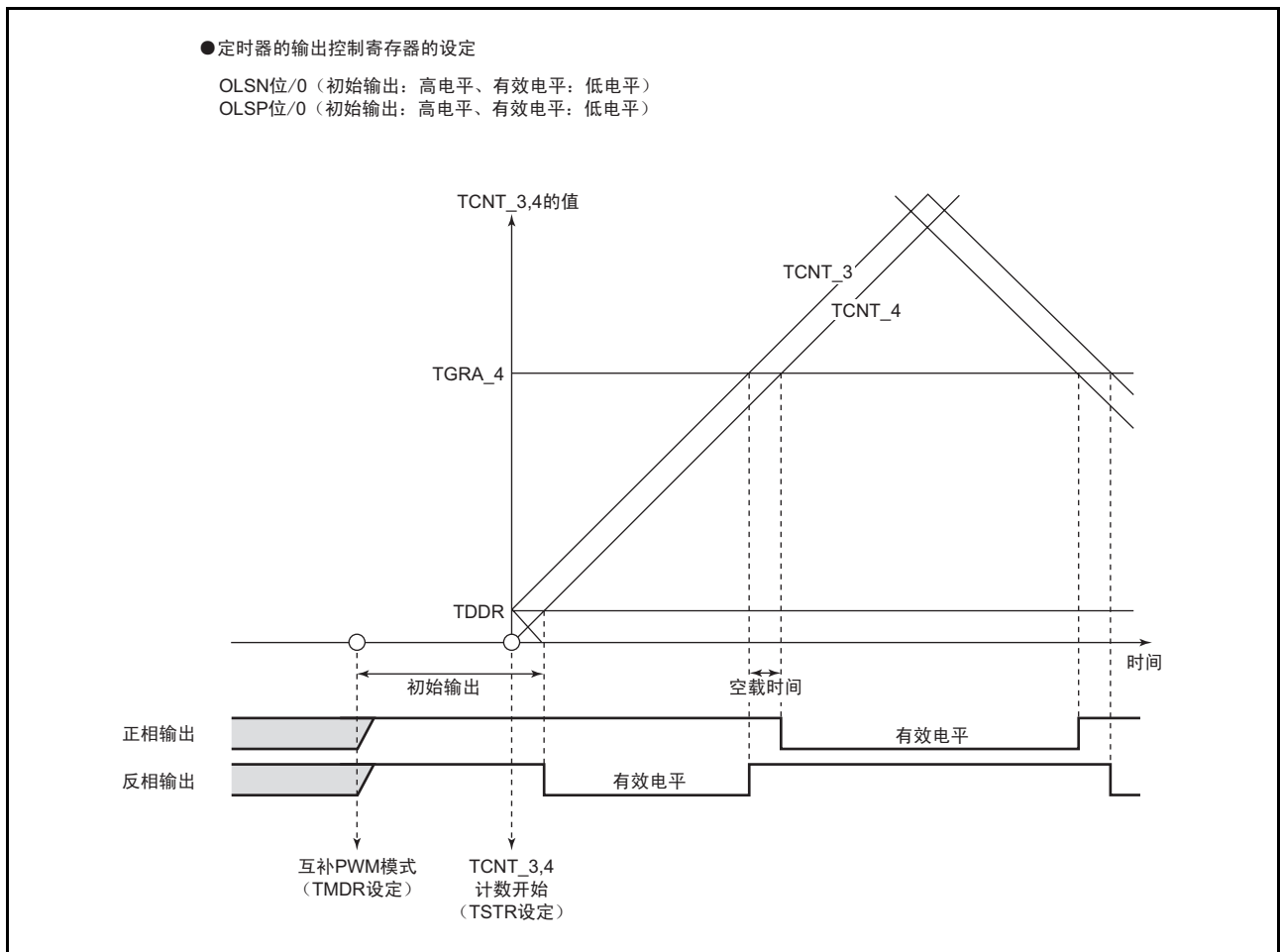


图 11.44 互补 PWM 模式的初始输出例 (1)



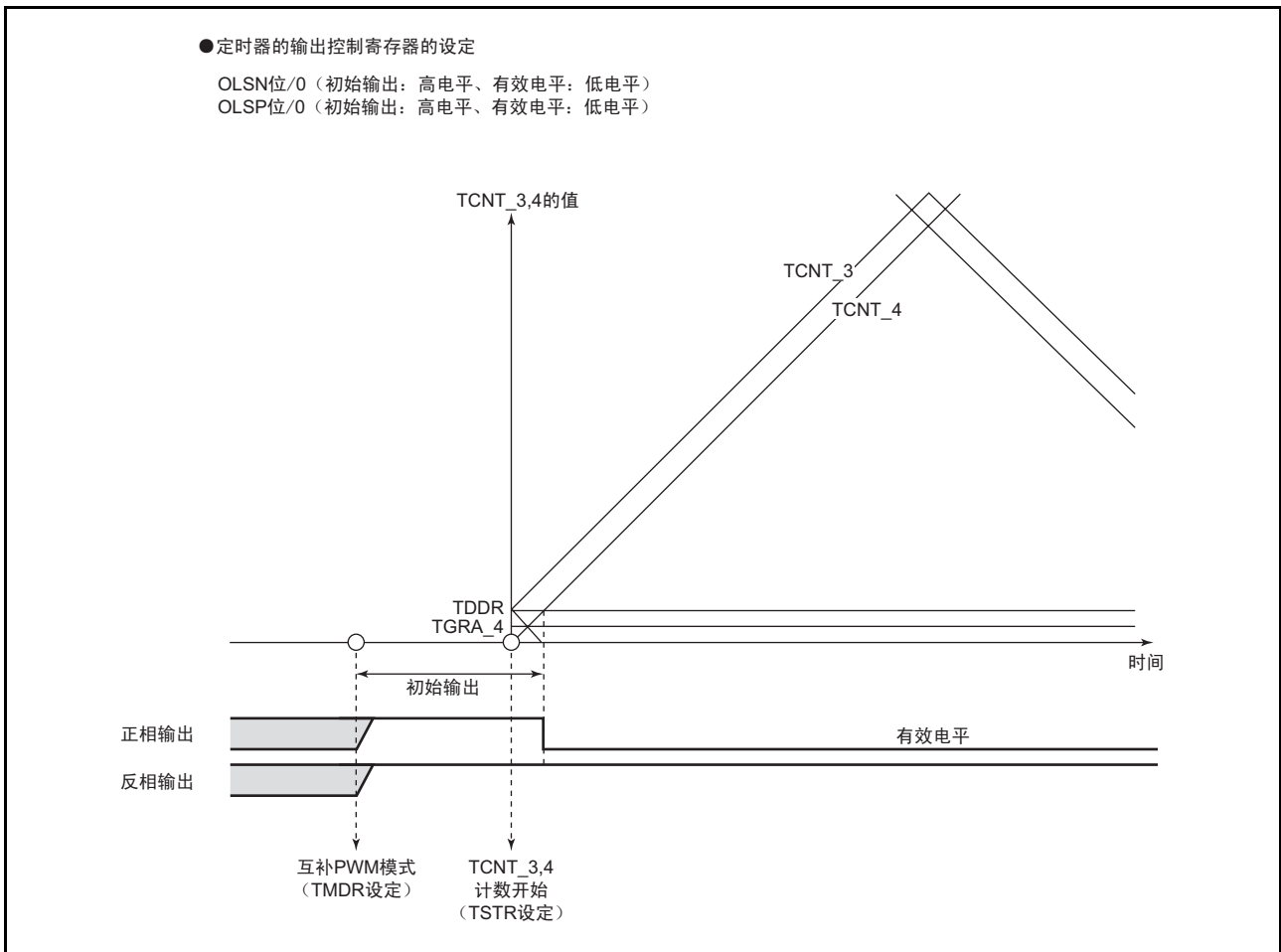


图 11.45 互补 PWM 模式的初始输出例 (2)

#### (j) 互补 PWM 模式的 PWM 输出生成方法

在互补 PWM 模式中，将正相和反相有非重叠时间的 PWM 波形进行 3 相输出。此非重叠时间称为空载时间。

在发生计数器和数据寄存器的比较匹配时，通过输出定时器的输出控制寄存器选择的输出电平生成 PWM 波形。另外，在 TCNTS 进行计数的期间，因为产生 0 ~ 100% 连续的 PWM 脉冲，所以同时比较数据寄存器和暂存器的值。此时，与 ON、OFF 比较匹配产生的时序都会改变，为了确保空载时间并且不使正相 / 反相的 ON 时间重叠，必须优先 OFF 各相的比较匹配。互补 PWM 模式的波形生成例子如图 11.46 ~ 图 11.48 所示。

通过和实线计数器的比较匹配生成正相 / 反相的 OFF 时序，通过和点线计数器（比实线计数器迟空载时间后的计数器运行）的比较匹配生成 ON 时序。在 T1 期间，最优先使反相 OFF 的 a 的比较匹配，忽视比 a 先产生的比较匹配。另外，在 T2 期间，最优先使正相 OFF 的 c 的比较匹配，忽视比 c 先产生的比较匹配。

如图 11.46 所示，通常按照 a→b→c→d（或者 c→d→a'→b'）的顺序产生比较匹配。

当比较匹配偏离 a→b→c→d 的顺序时，因为使反相 OFF 的时间短于空载时间的 2 倍，所以表示正相不为 ON；当比较匹配偏离 c→d→a'→b' 的顺序时，因为使正相 OFF 的时间短于空载时间的 2 倍，所以表示反相不为 ON。

如图 11.47 所示，如果在 a 的比较匹配之后先产生 c 的比较匹配，就忽视 b 的比较匹配，而通过 d 的比较匹配使反相 OFF。这是为了通过比 b 的比较匹配（正相 ON 时序）先产生正相 OFF 的 c 的比较匹配，优先使正相 OFF（在 OFF 状态时即使“OFF”命令信号，波形也不发生变化）。

同样地，在图 11.48 所示的例子中，比 c 的比较匹配先产生和暂存器的新数据比较匹配的 a'，但是在产生使正相 OFF 的 c 前忽视其他比较匹配，因此反相不为 ON。

如此，在互补 PWM 模式中，优先 OFF 时序的比较匹配，即使 ON 时序的比较匹配比 OFF 先产生也被忽视。

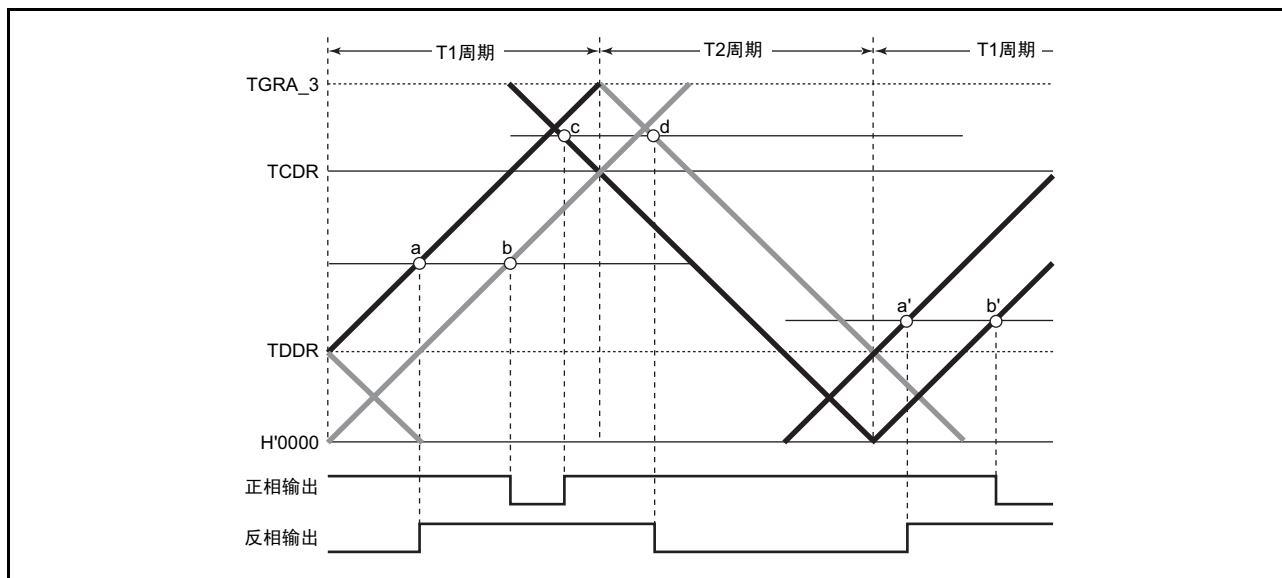


图 11.46 互补 PWM 模式的波形输出例 (1)

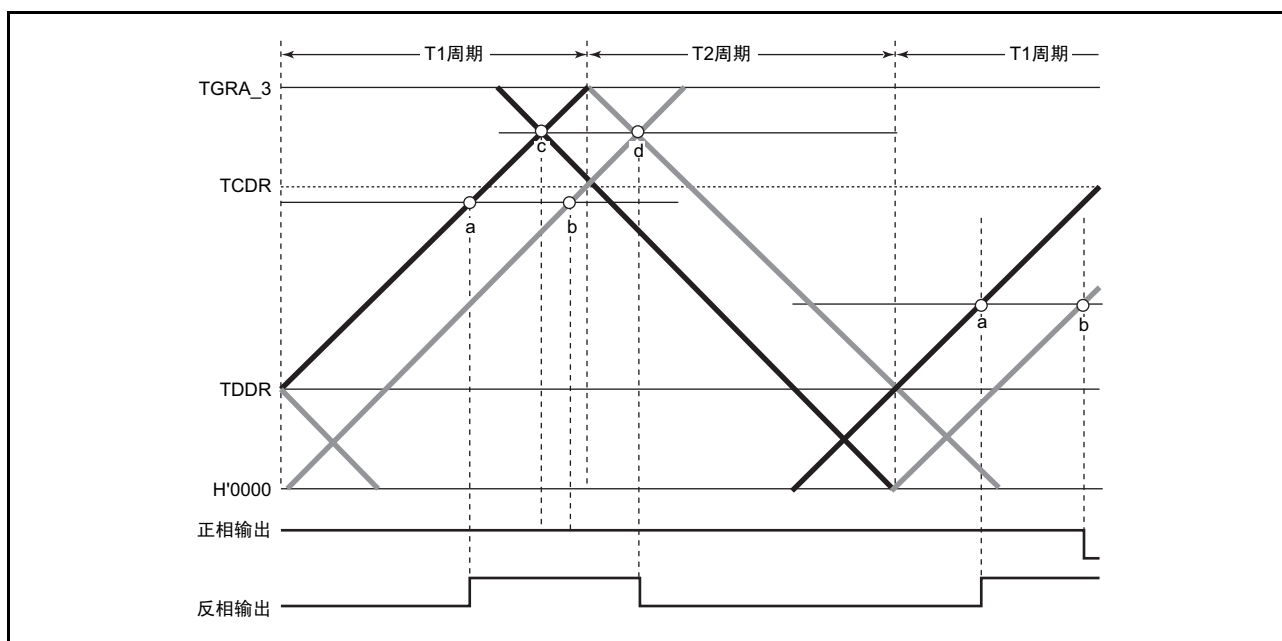


图 11.47 互补 PWM 模式的波形输出例 (2)

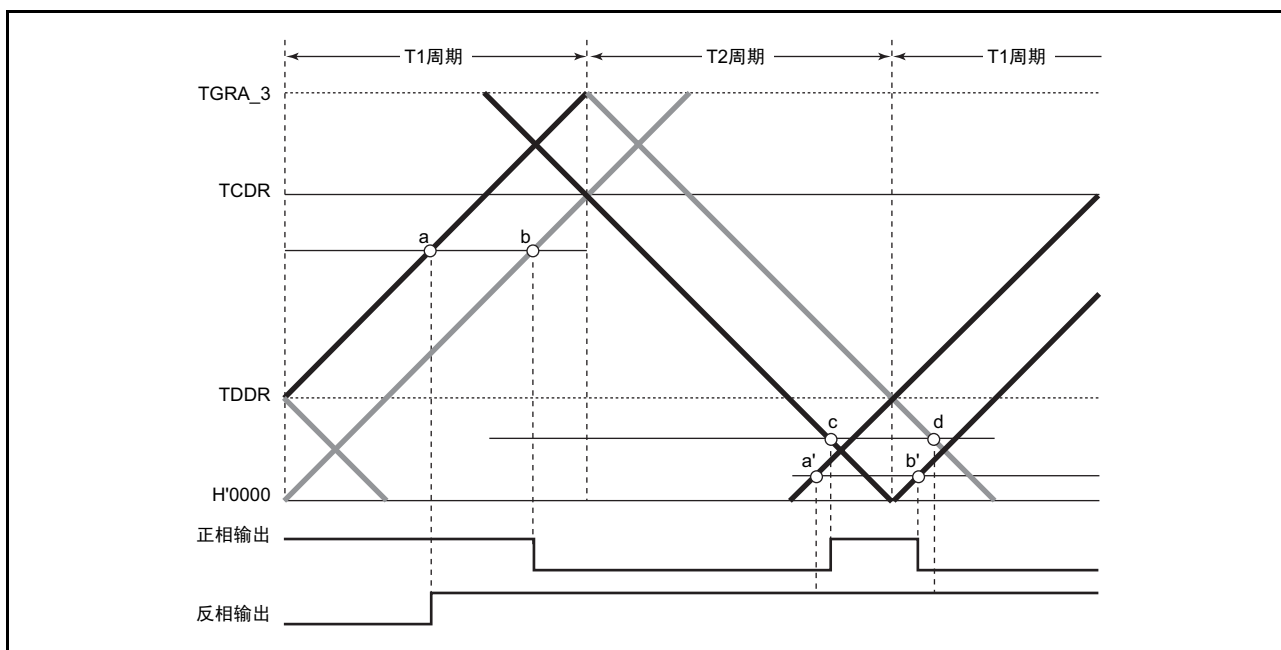


图 11.48 互补 PWM 模式的波形输出例 (3)

(k) 互补 PWM 模式的 0%、100% 占空比输出

在互补 PWM 模式中，能任意输出 0%、100% 的占空比，输出例子如图 11.49 ~ 图 11.53 所示。

如果将数据寄存器的值设定为 H'0000，就输出 100% 的占空比，此时的波形是正相为 100%ON 状态的波形。另外，如果将数据寄存器的值设定为 TGRA\_3 的相同值，就输出 0% 的占空比，此时的波形是正相为 100%OFF 状态的波形。

此时，同时产生 ON 和 OFF 的比较匹配，但是，如果同相 ON 的比较匹配和 OFF 的比较匹配同时产生，双方的比较匹配都被忽视而波形不变。

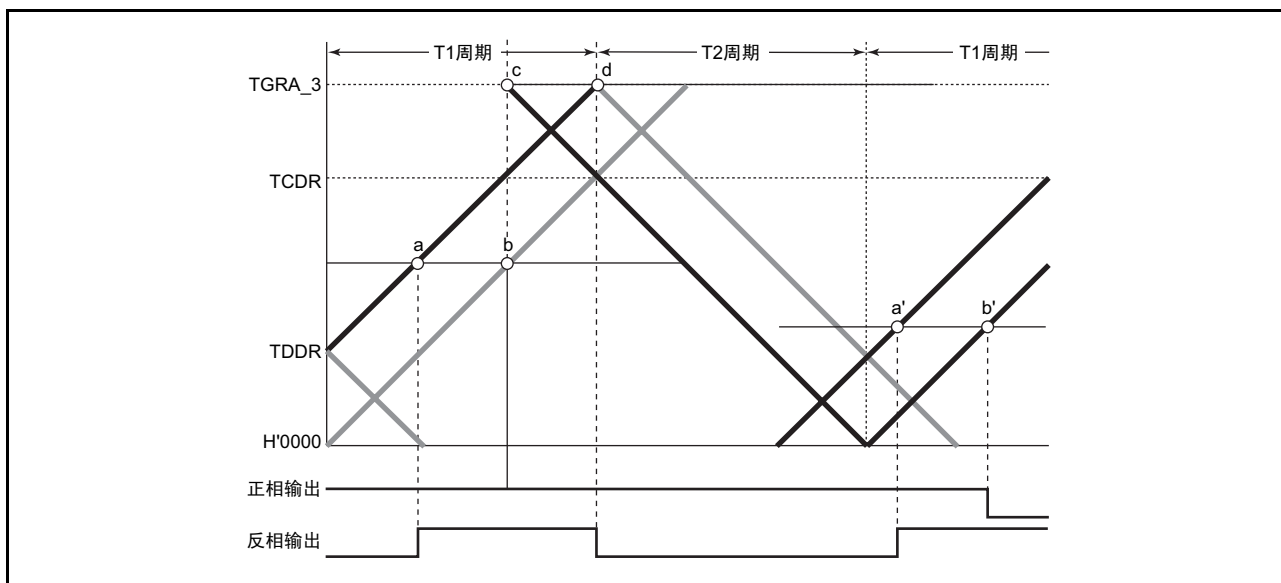


图 11.49 互补 PWM 模式的 0%、100% 波形输出例 (1)

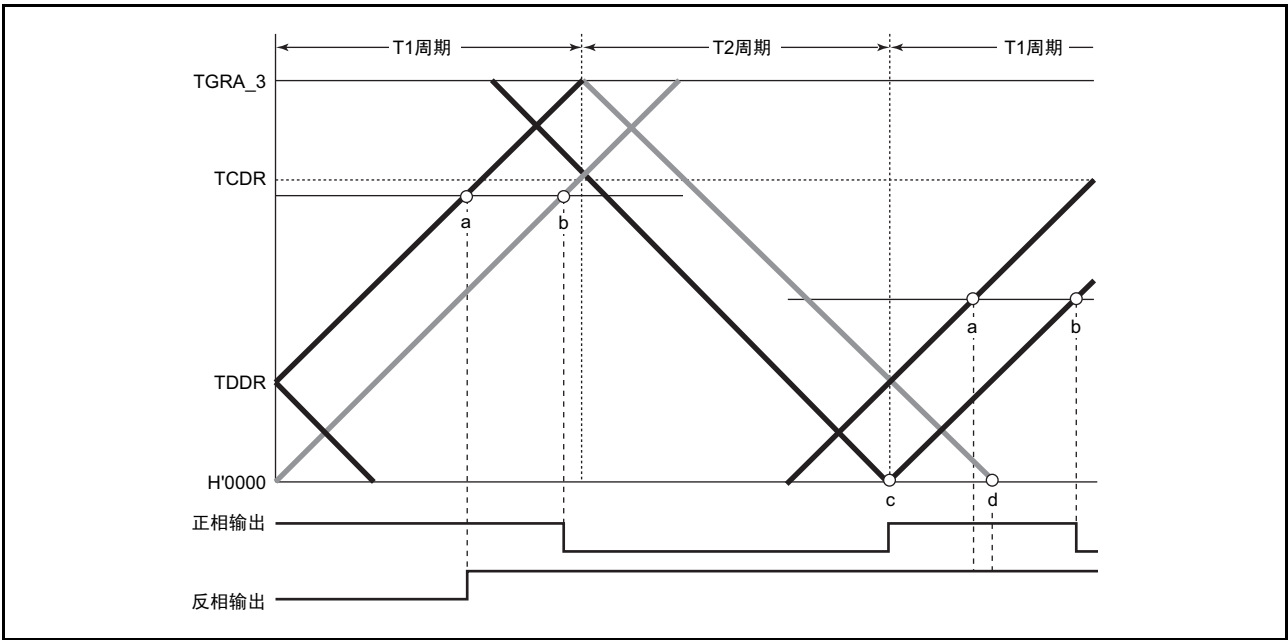


图 11.50 互补 PWM 模式的 0%、100% 波形输出例 (2)

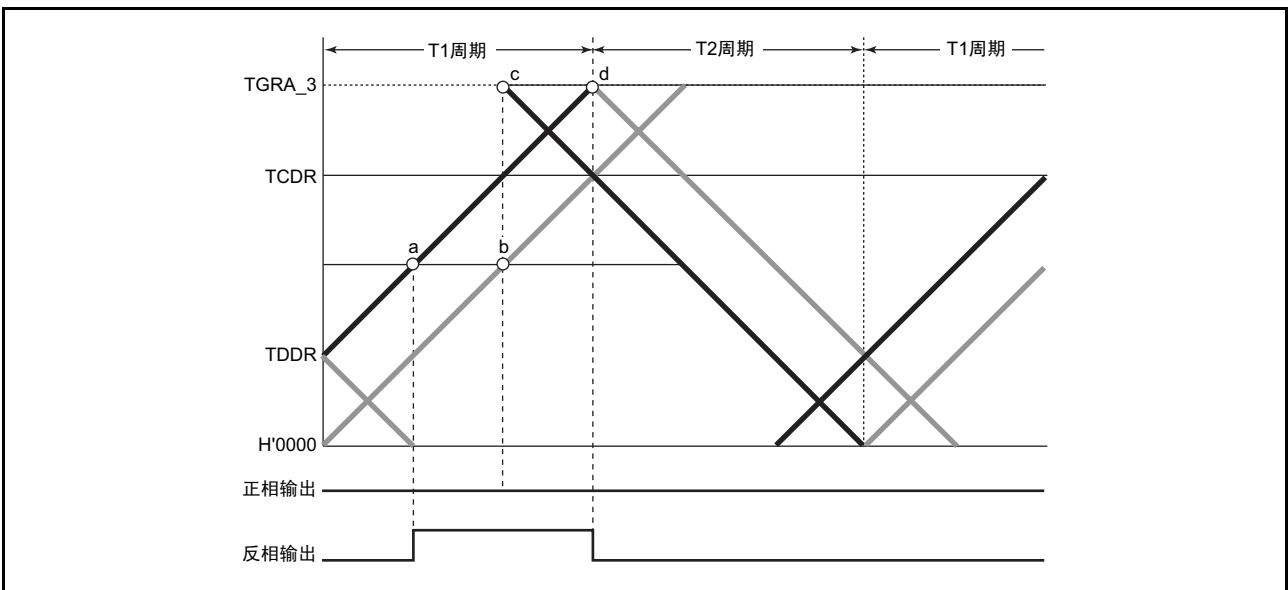


图 11.51 互补 PWM 模式的 0%、100% 波形输出例 (3)

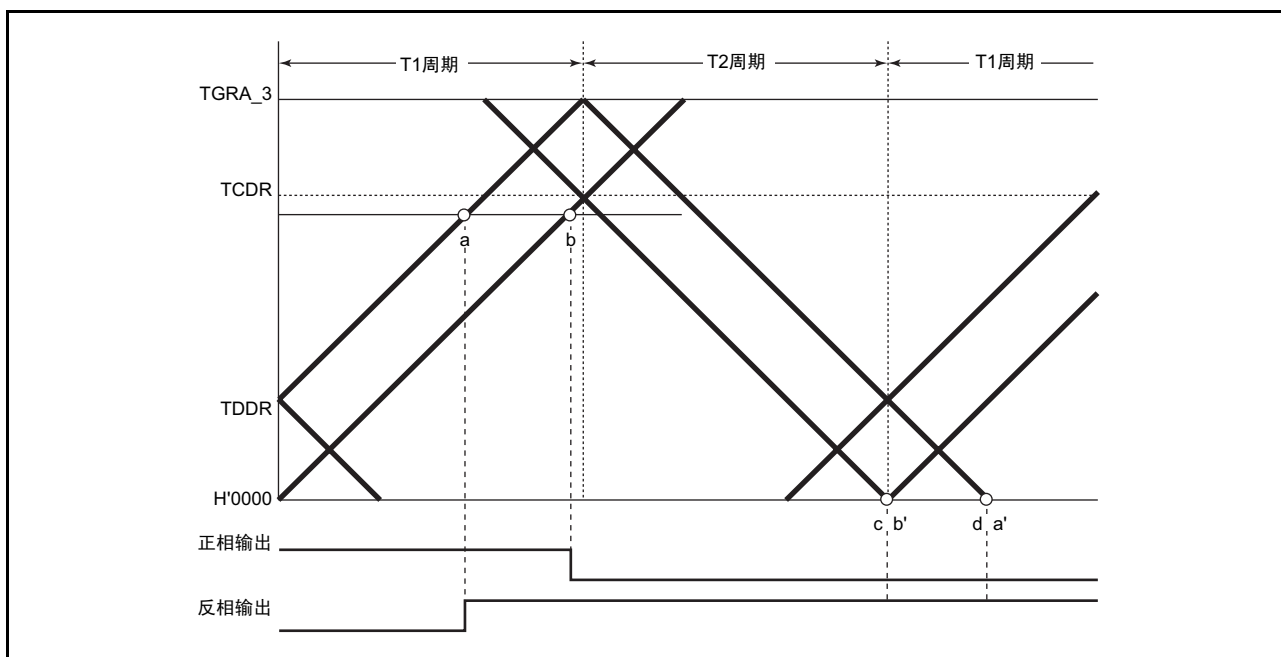


图 11.52 互补 PWM 模式的 0%、100% 波形输出例 (4)

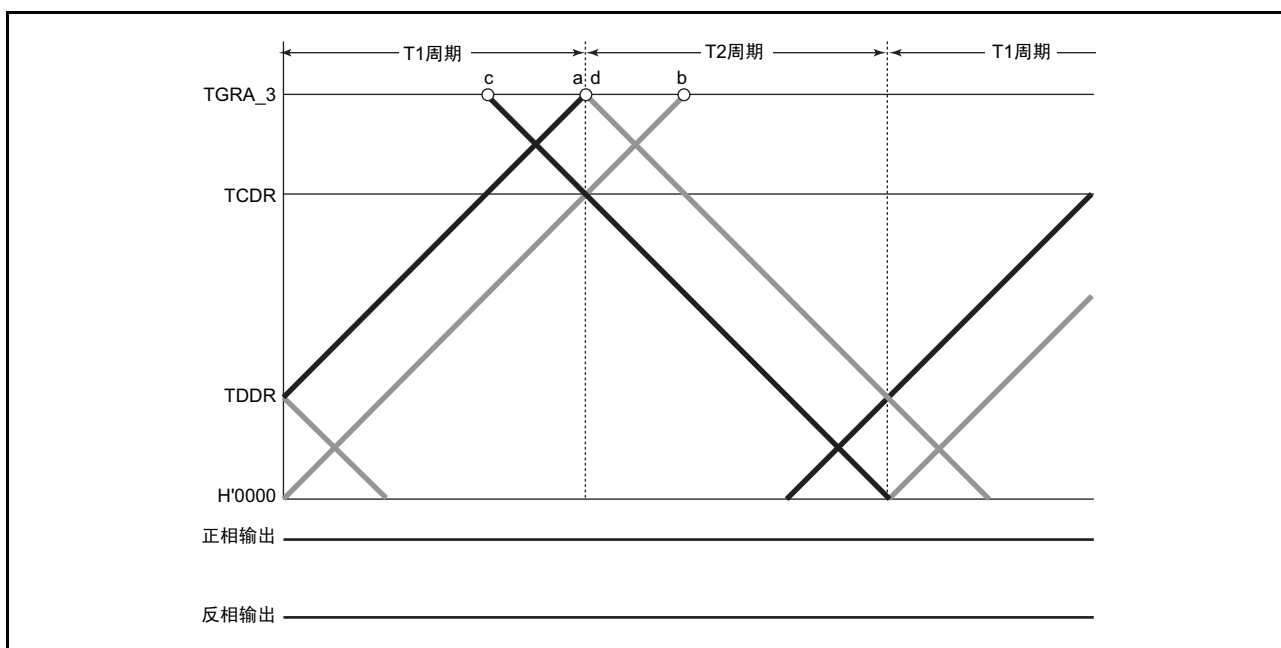


图 11.53 互补 PWM 模式的 0%、100% 波形输出例 (5)

(I) 与 PWM 周期同步的交替输出

在互补 PWM 模式中，能通过将定时器的输出控制寄存器 (TOCR) 的 PSYE 位设定为 1 进行与 PWM 载波周期同步的交替输出。交替输出的波形例子如图 11.54 所示。

通过 TCNT\_3 和 TGRA\_3 的比较匹配以及 TCNT4 和 H'0000 的比较匹配进行交替输出。

此交替输出的输出引脚是 TIOC3A 引脚，初始输出为 1。

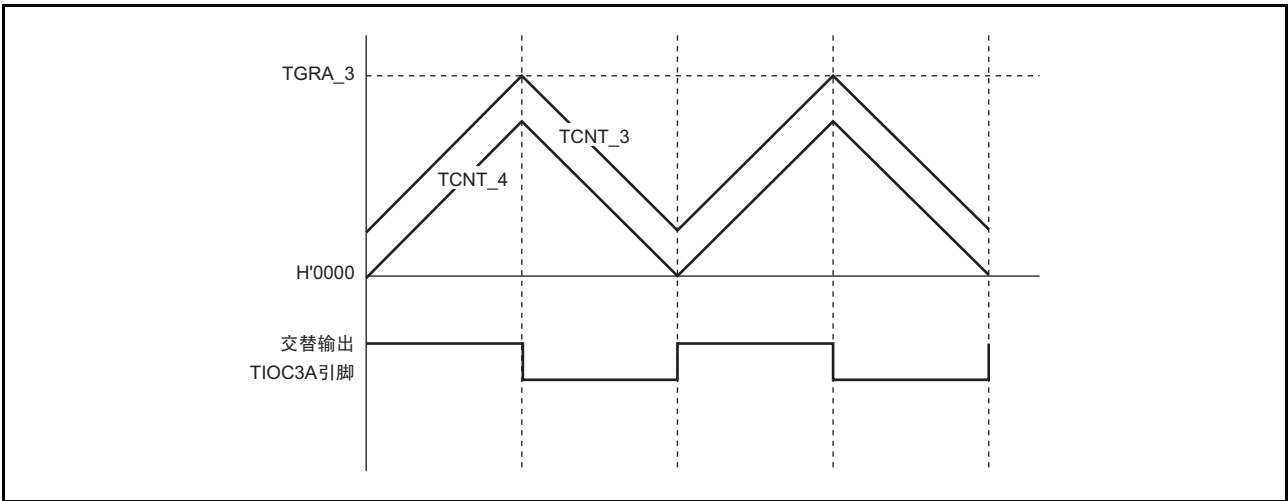


图 11.54 与 PWM 输出同步的交替输出的波形例子

(m) 其他通道的计数器清除

在互补 PWM 模式中，通过定时器的同步寄存器 (TSYR) 设定为与其他通道的同步模式，并且通过定时器的控制寄存器 (TCR) 的 CCLR2 ~ CCLR0 选择同步清除，就能由其他通道进行 TCNT\_3、TCNT\_4 和 TCNTS 的清除。

运行例子如图 11.55 所示。

使用此功能，能通过外部信号进行计数器的清除和重新开始。

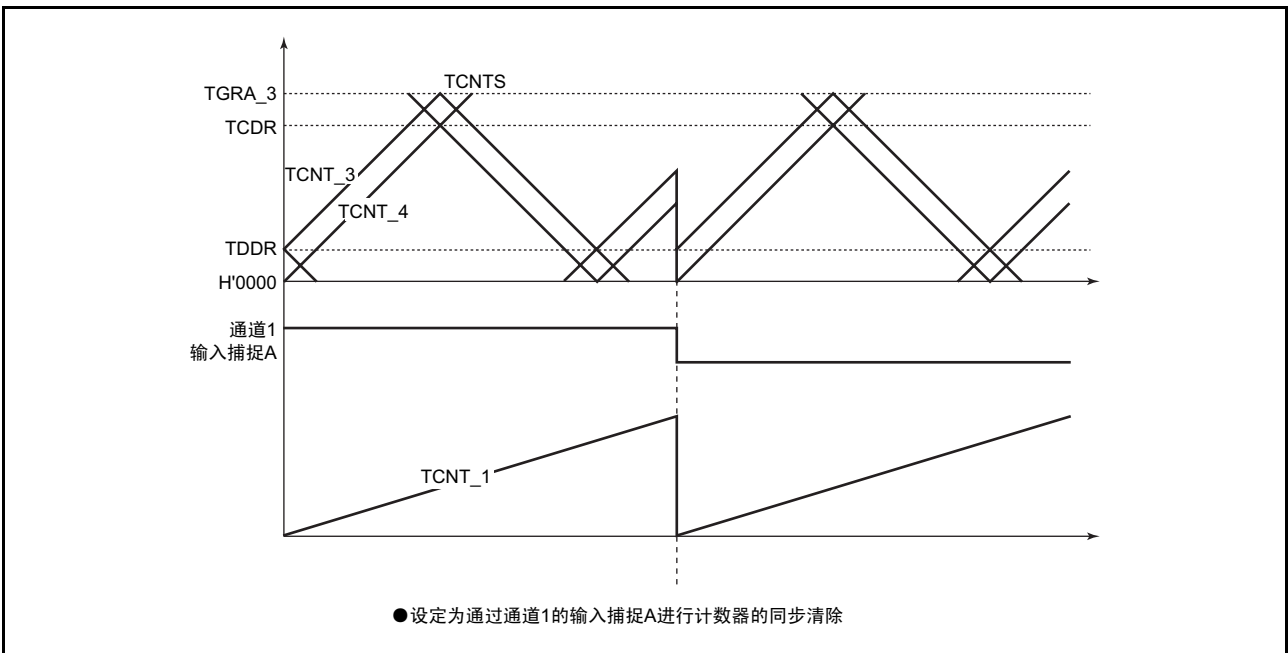


图 11.55 与其他通道同步的计数器清除

## (n) 互补 PWM 模式中的同步计数器清除时的输出波形控制

能通过将 TWCR 寄存器的 WRE 位设定为 1，抑制在互补 PWM 模式的波谷 Tb 区间发生同步计数器清除时的初始输出。因此，能抑制同步计数器清除时占空比的急剧变化。

只在如图 11.56 的(00)、(01)的波谷 Tb 区间进行同步清除时，才能抑制初始输出（通过将 WRE 位设定为 1）。如果在其他时序中发生同步清除，就输出 TOCR 寄存器的 OLS 位设定的初始值。另外，即使在波谷的 Tb 区间，如果在图 11.56 的(1)所示的计数器刚开始的初始输出期间发生同步清除，也不抑制初始输出。

本功能可用于 MTU2 和 MTU2S，MTU2 和 MTU2S 计数器的清除源分别是 MTU2 的通道 0 ~ 2 的同步清除和 MTU2 的通道 0 ~ 2 的标志置位（比较匹配 / 输入捕捉）。

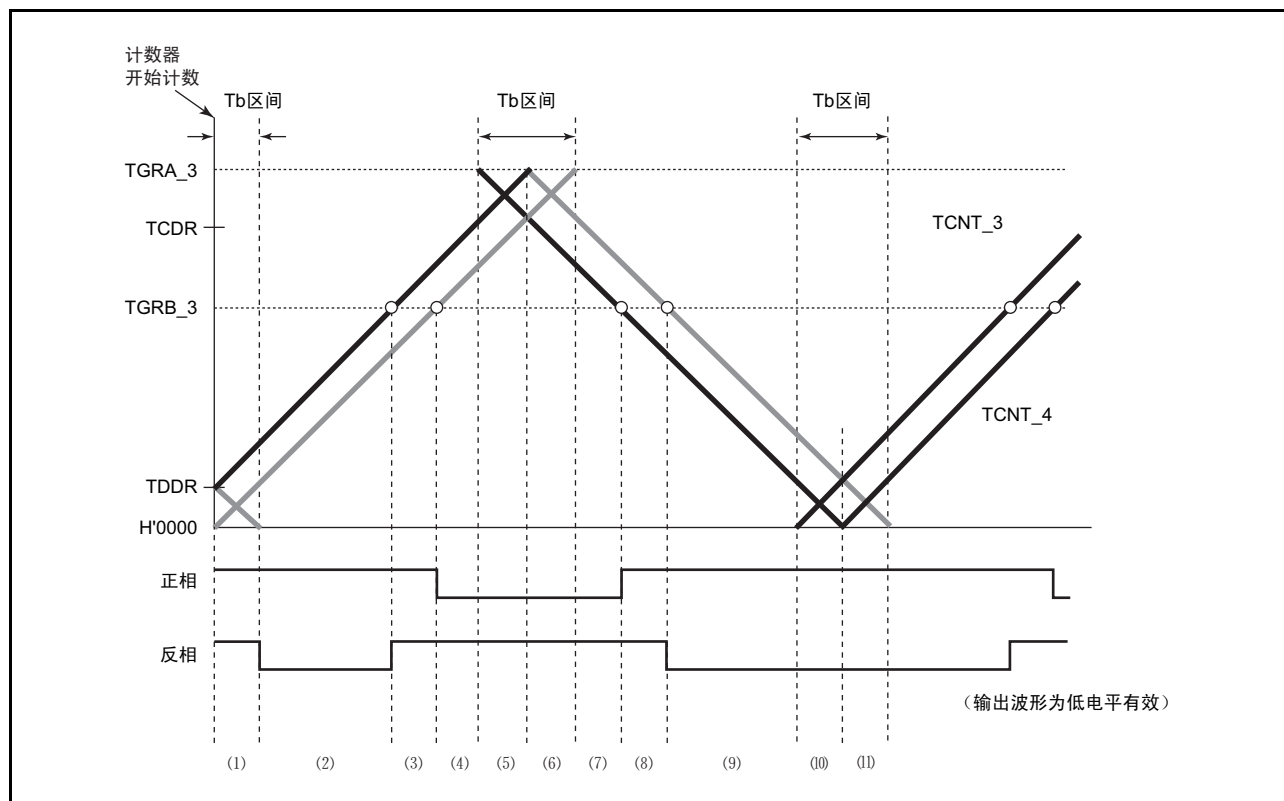


图 11.56 同步计数器清除的时序

• 互补 PWM 模式中的同步计数器清除时的输出波形控制的设定步骤例子

互补 PWM 模式中的同步计数器清除时的输出波形控制的设定步骤例子如图 11.57 所示。

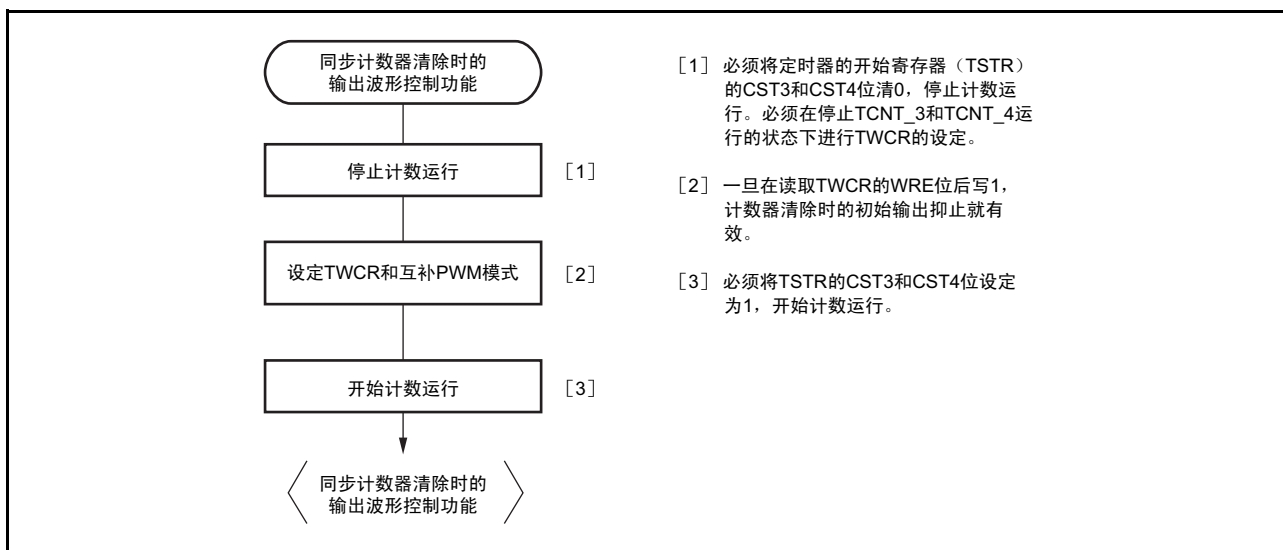


图 11.57 互补 PWM 模式中的同步计数器清除时的输出波形控制的设定步骤例子

• 互补 PWM 模式中的同步计数器清除时的输出波形控制的运行例

在将 TWCR 的 WRE 位设定为 1 的状态下使 MTU2 进行互补 PWM 运行，并进行同步计数器清除时的运行例子如图 11.58 ~ 图 11.61 所示。在此，图 11.58 ~ 图 11.61 的同步计数器清除的时序分别是图 11.56 的 (3)、(6)、(8)、(11) 所示的时序。

在此例中，对于 MTU2S，在将 TWCR 的 SCC 位设定为 0 以及将 WRE 位设定为 1 的状态下，使 MTU2S 进行互补 PWM 运行，相当于计数器同步清除的情况。

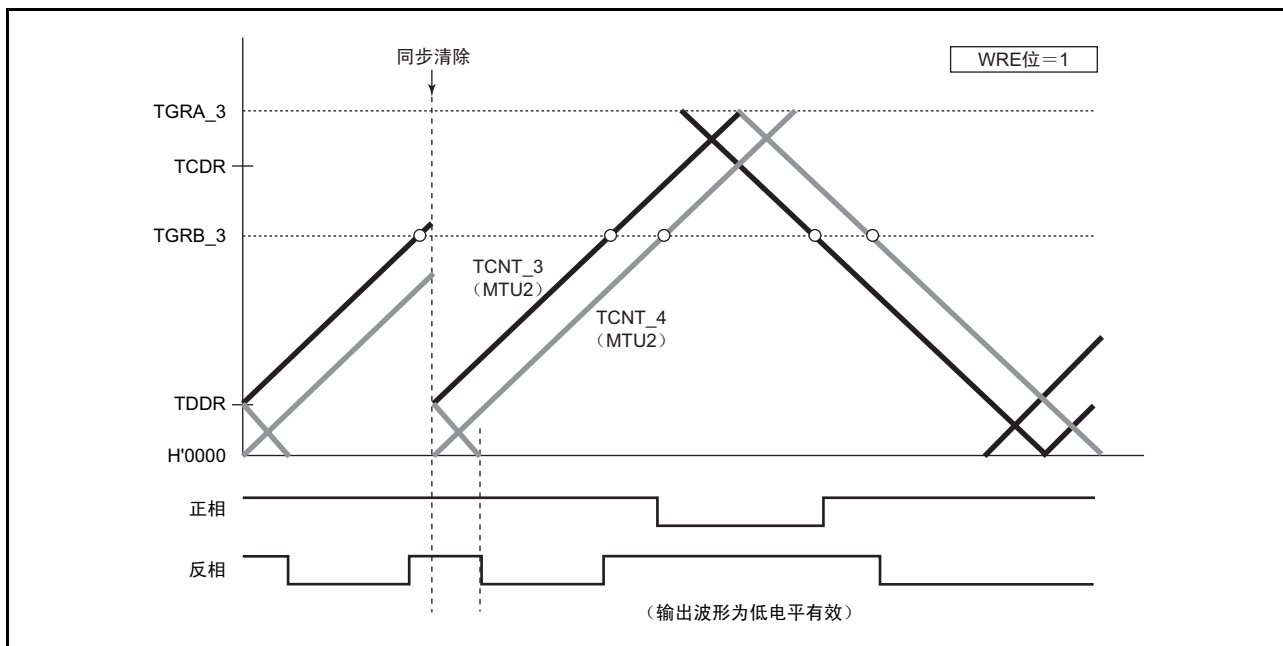


图 11.58 在递增计数中的空载时间发生同步清除  
(图 11.56 的时序(3)、MTU2 的 TWCR 寄存器的 WRE 位 = 1)



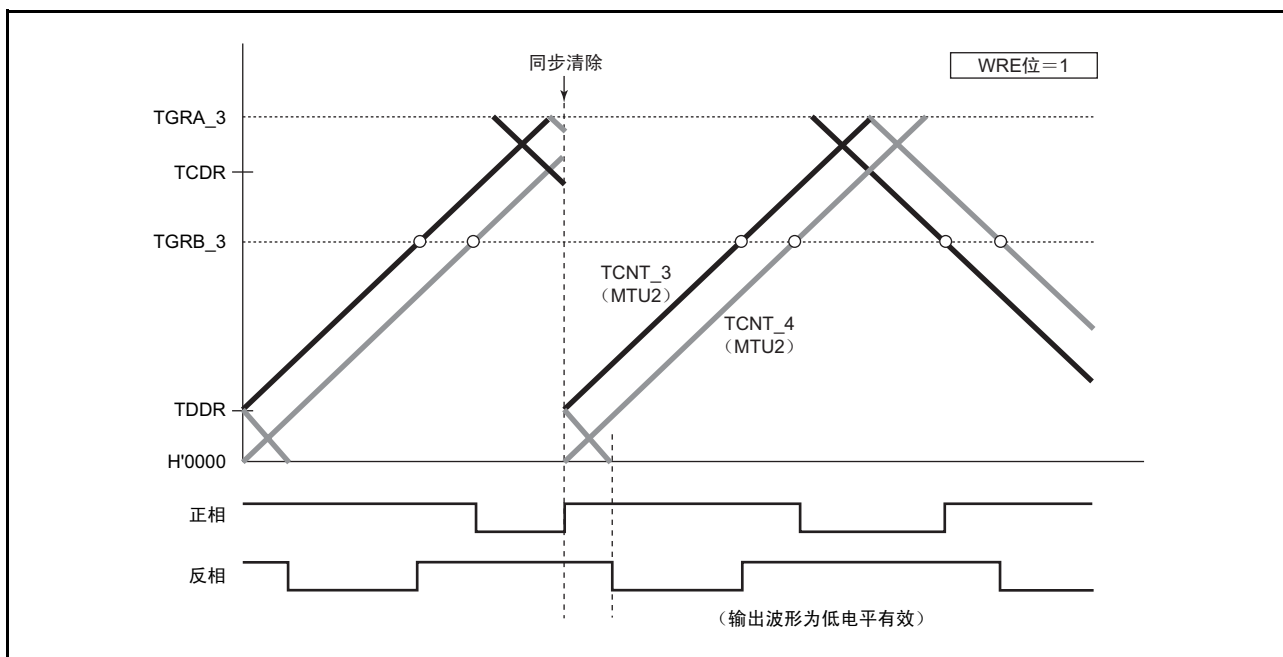


图 11.59 在波峰的 Tb 区间发生同步清除  
(图 11.56 的时序(6)、 MTU2 的 TWCR 寄存器的 WRE 位 =1)

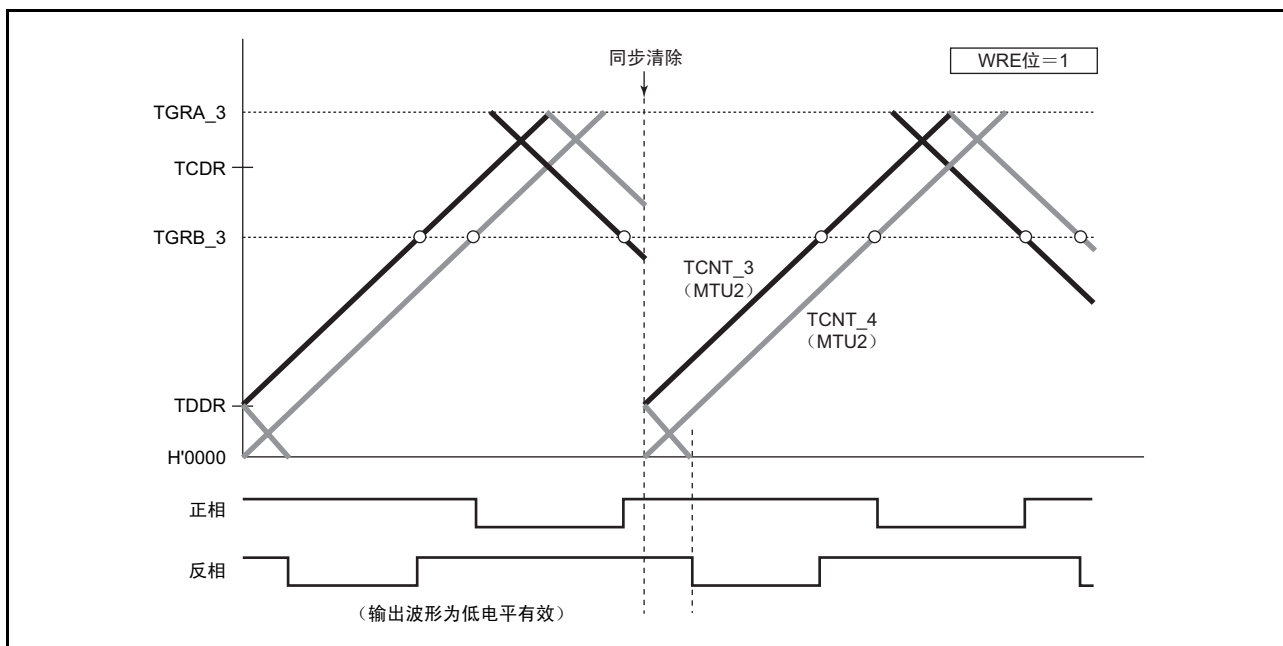


图 11.60 在递减计数中的空载时间发生同步清除  
(图 11.56 的时序(8)、 TWCR 寄存器的 WRE 位 =1)

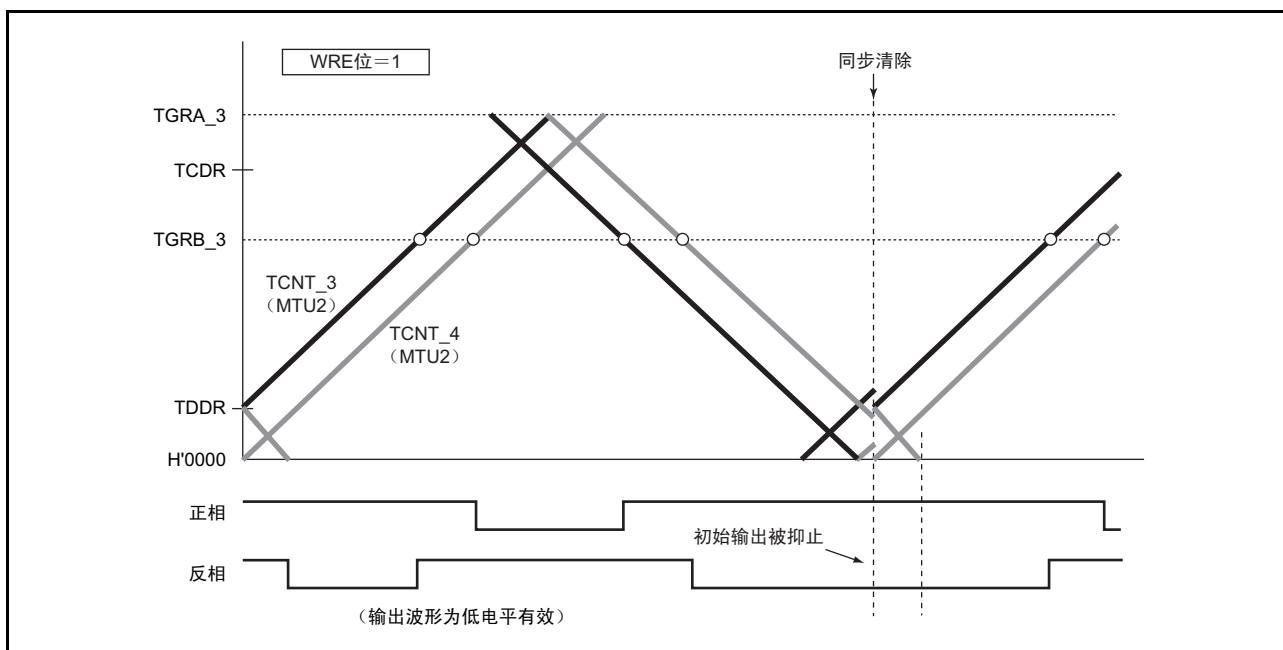


图 11.61 在波谷的 Tb 区间发生同步清除  
(图 11.56 的时序(I)、TWCR 寄存器的 WRE 位 =1)

(o) MTU2-MTU2S 计数器同步清除的抑止功能

对于 MTU2S，能通过将 TWCR 寄存器的 SCC 位设定为 1，抑止由 MTU2 进行的同步清除。

能通过 SCC 位的设定抑止由 MTU2 进行同步清除的区间如图 11.62 所示。

在使用本功能时，必须将 MTU2S 设定为互补 PWM 模式。

有关由 MTU2 进行同步清除的详细内容，请参照“11.4.10(2) 利用 MTU2 标志的置位源进行 MTU2S 计数器的清除 (MTU2-MTU2S 计数器同步清除)”。

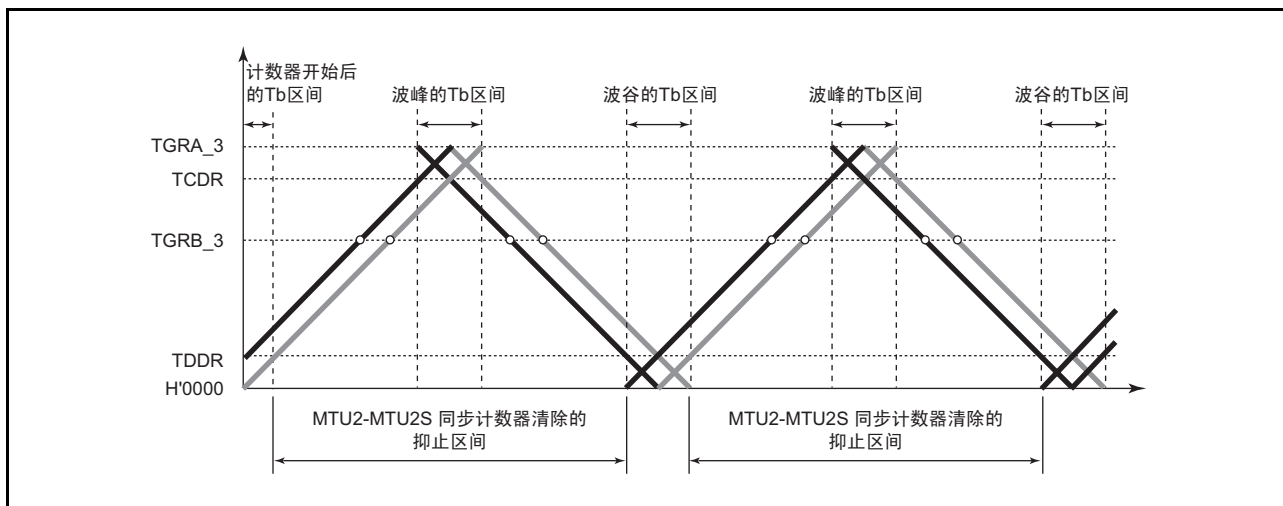


图 11.62 通过 TWCR 的 SCC 位的置位，进行 MTU2-MTU2S 同步清除的抑止区间

• MTU2-MTU2S 计数器同步清除抑止功能的设定步骤例子

MTU2-MTU2S 计数器同步清除抑止功能的设定步骤例子如图 11.63 所示。

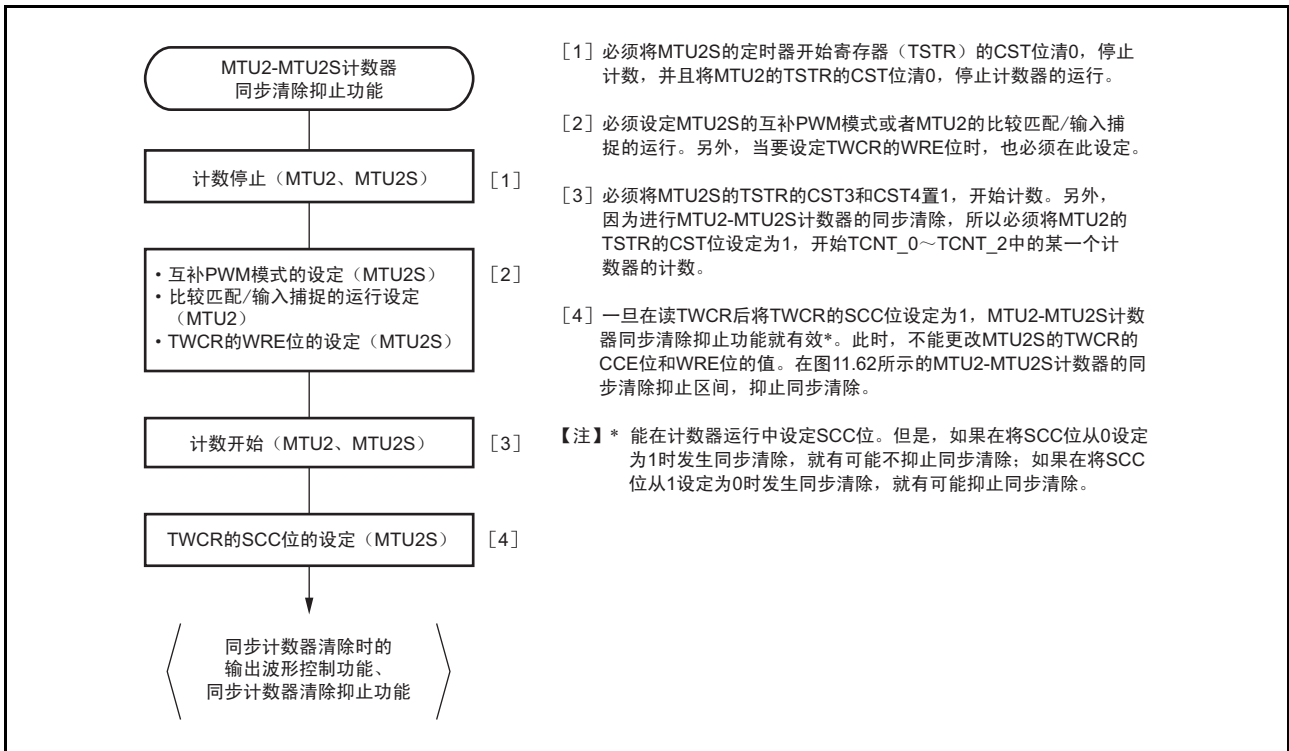


图 11.63 MTU2-MTU2S 计数器同步清除抑止功能的设定步骤例子

• MTU2-MTU2S 计数器同步清除抑止功能的运行例

如图 11.64 ~图 11.67 所示, 将 MTU2S 的 TWCR 的 SCC 位设定为 1, 使 MTU2S 进行互补 PWM 运行, 并且使 MTU2-MTU2S 计数器同步清除抑止功能有效。在此, 图 11.64 ~图 11.67 的计数器同步清除时序分别为图 11.56 的(3)、(6)、(8)、(11)所示的时序, 并且在此例中将 MTU2S 的 TWCR 的 WRE 位设定为 1。

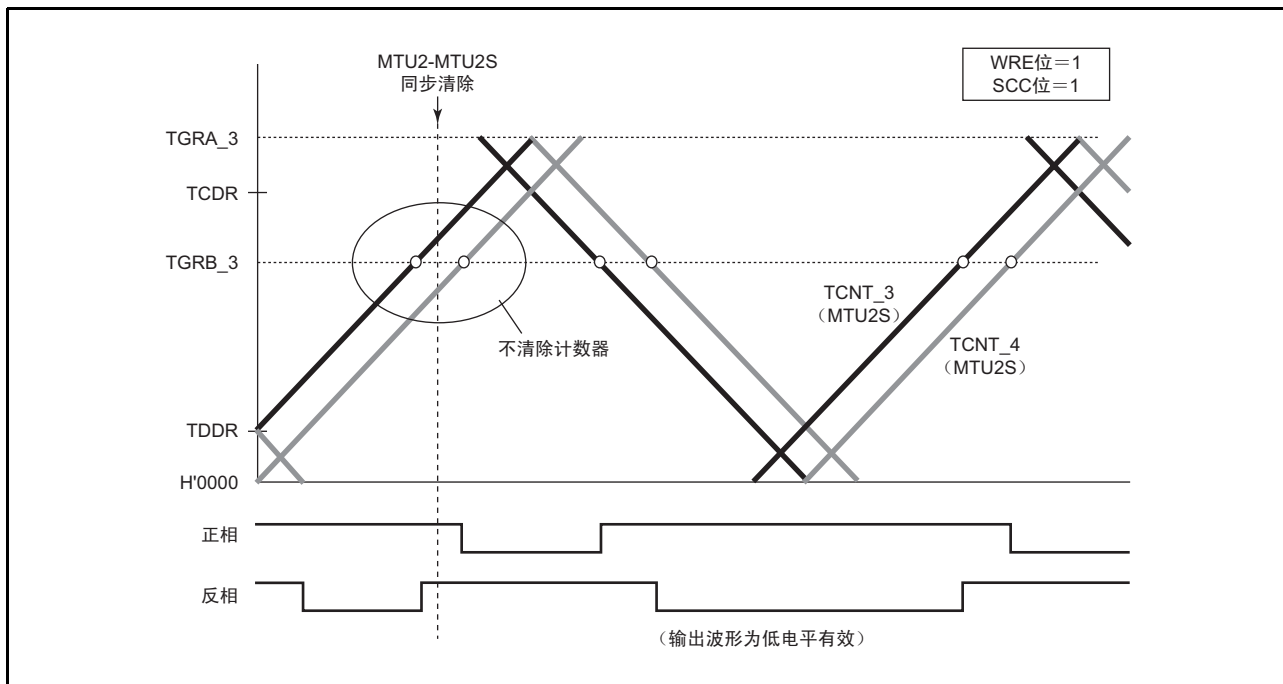


图 11.64 在递增计数中的空载时间发生同步清除  
(图 11.56 的时序(3)、MTU2S 的 TWCR 寄存器的 WRE 位 =1、SCC 位 =1)

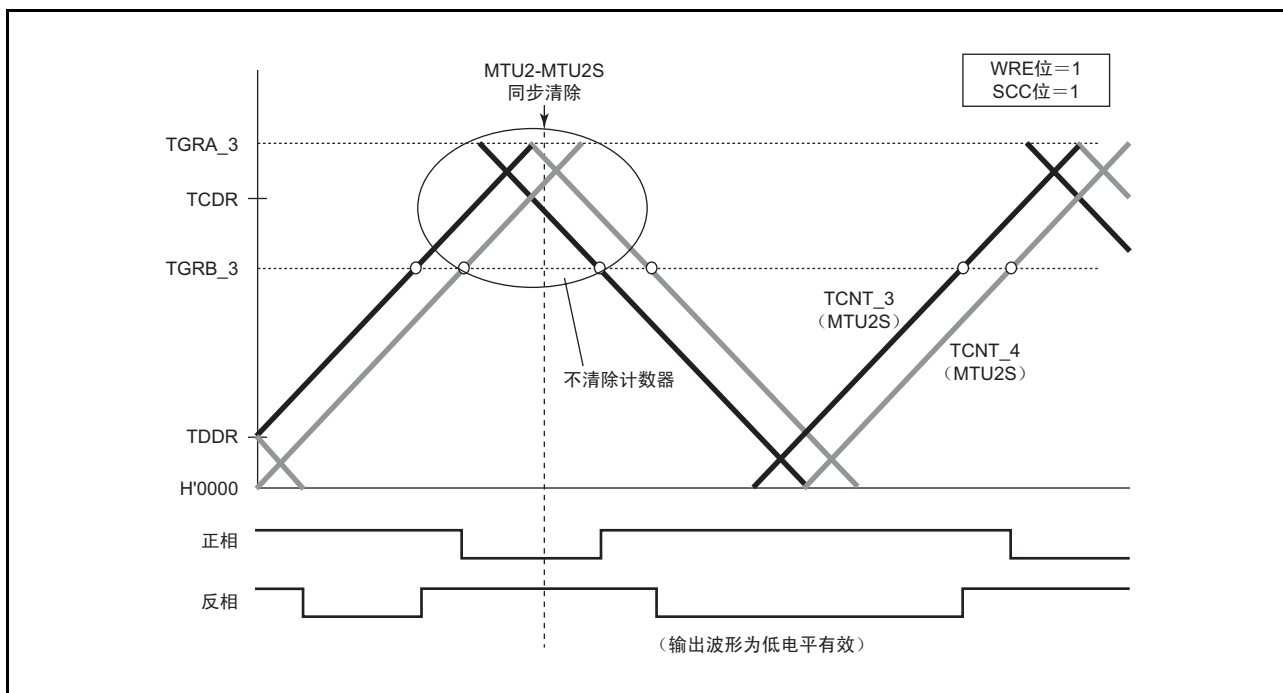


图 11.65 在波峰的 Tb 区间发生同步清除  
(图 11.56 的时序(6)、MTU2S 的 TWCR 寄存器的 WRE 位 =1、SCC 位 =1)

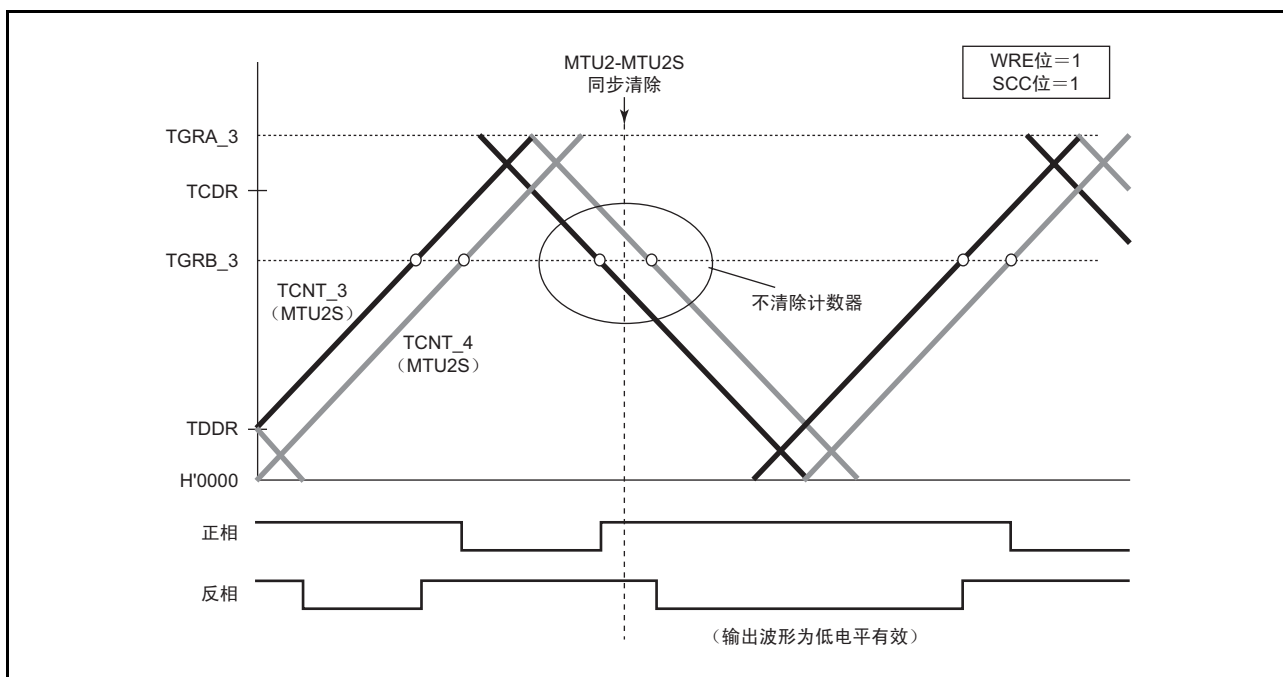


图 11.66 在递减计数中的空载时间发生同步清除  
(图 11.56 的时序(8)、MTU2S 的 TWCR 寄存器的 WRE 位 =1、SCC 位 =1)

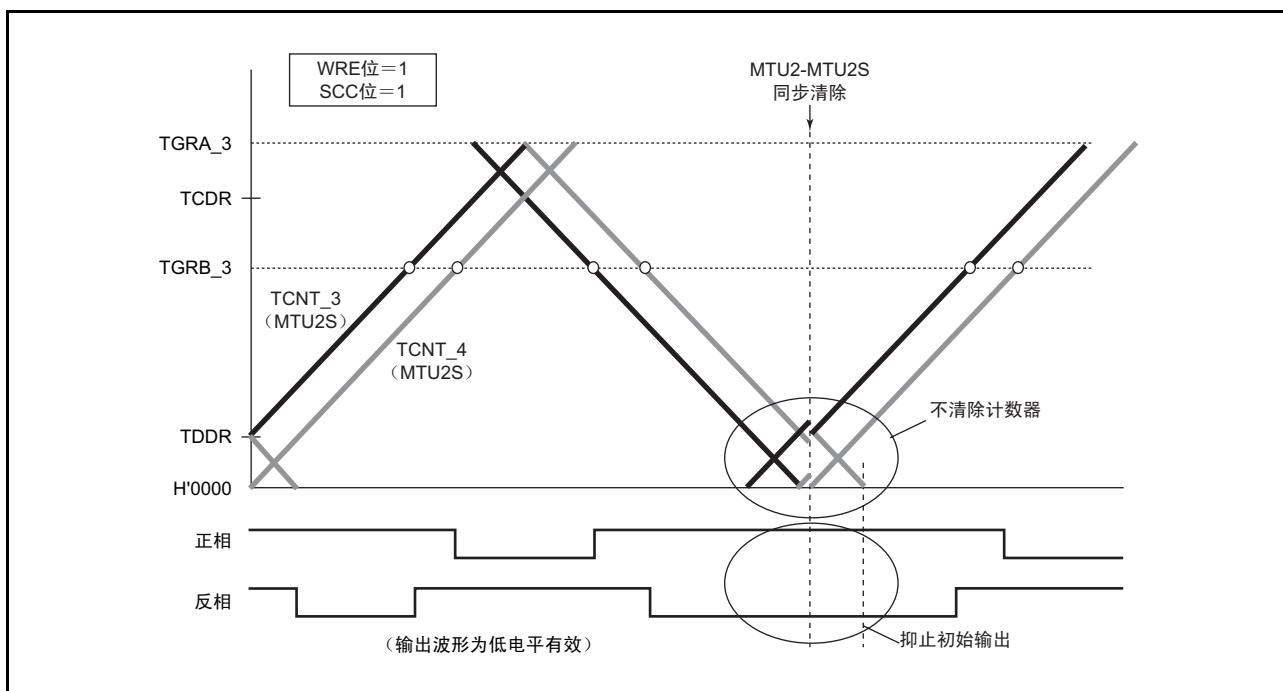


图 11.67 在波谷的 Tb 区间发生同步清除  
(图 11.56 的时序(11)、MTU2S 的 TWCR 寄存器的 WRE 位 =1、SCC 位 =1)

## (p) 通过 TGRA\_3 的比较匹配清除计数器

在互补 PWM 模式中，能通过设定定时器的波形控制寄存器 (TWCR) 的 CCE 位，在 TGRA\_3 的比较匹配清除 TCNT\_3、TCNT\_4 和 TCNTS。

运行例子如图 11.68 所示。

- 【注】
1. 只能在互补 PWM 模式 1 (在波峰进行传送) 中使用。
  2. 不能设定为和其他通道同步的清除功能 (不能将定时器的同步寄存器 (TSYR) 的 SYNC0 ~ SYNC4 位和定时器的同步清除寄存器 (TSYCR) 的 CE0A/B/C/D、CE1A/B/C/D 位设定为 1)。
  3. PWM 占空比不能置 H'0000。
  4. 不能将定时器的输出控制寄存器 1 (TOCR1) 的 PSYE 位设定为 1。

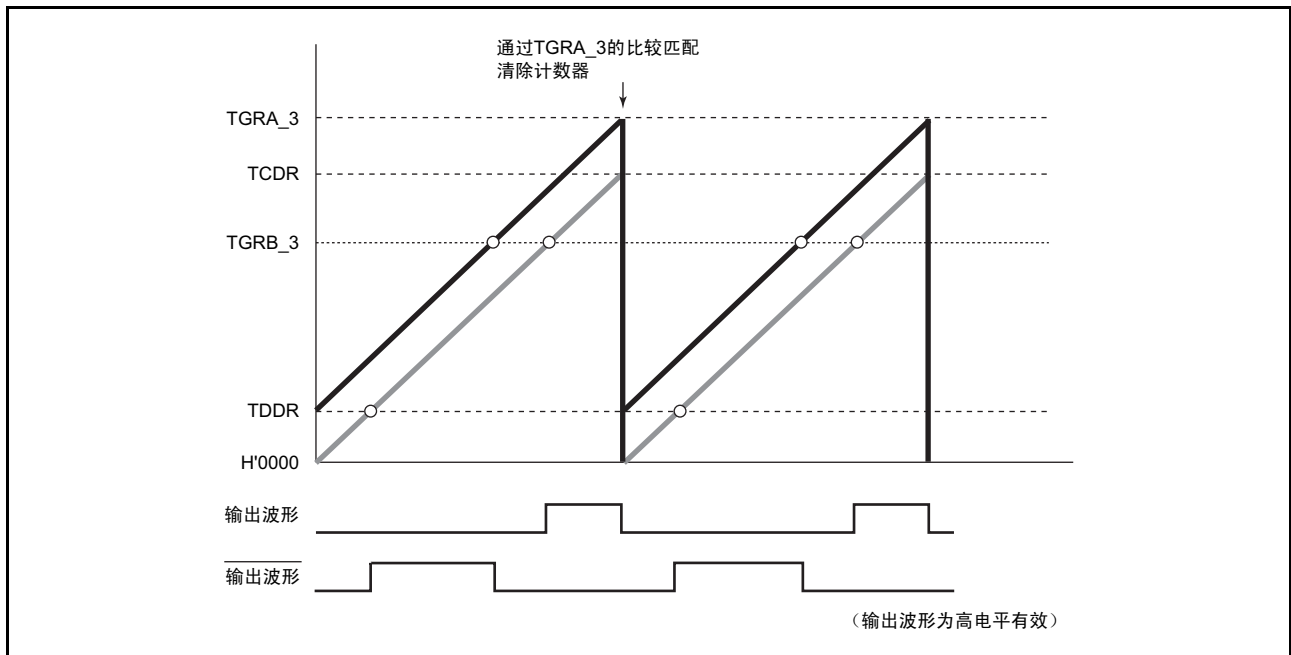


图 11.68 通过 TGRA\_3 的比较匹配清除计数器的运行例

## (q) AC 同步马达 (无刷 DC 马达) 的驱动波形输出例

在互补 PWM 模式中，能使用定时器的门控寄存器 (TGCR)，简单地控制无刷 DC 马达。使用 TGCR 的无刷 DC 马达的驱动波形例子如图 11.69 ~ 图 11.72 所示。

为了转换 3 相无刷 DC 马达的输出相，在使用霍尔元件等检测外部信号时，将 TGCR 的 FB 位设定为 0。此时，将表示磁极位置的外部信号输入到通道 0 的定时器的输入引脚 TIOC0A、TIOC0B 和 TIOC0C 引脚 (必须通过 PFC 进行设定)。如果 TIOC0A、TIOC0B 和 TIOC0C 这 3 个引脚检测到边沿，输出的 ON/OFF 就自动转换。

当 FB 位是 1 时，如果给 TGCR 的 UF、VF、WF 各位设定 0 或者 1，就会转换输出的 ON/OFF。

从互补 PWM 模式的 6 相输出引脚进行驱动波形的输出。

对于此 6 相输出，能通过将 N 位或者 P 位设定为 1，在 ON 输出时，使用互补 PWM 模式的输出，进行载波输出。在 N 位或者 P 位是 0 时，为电平输出。

另外，6 相输出的有效电平 (ON 输出时的电平) 与 N 位和 P 位的设定无关，能通过定时器的输出控制寄存器 (TOCR) 的 OLSN 位和 OLSP 位进行设定。

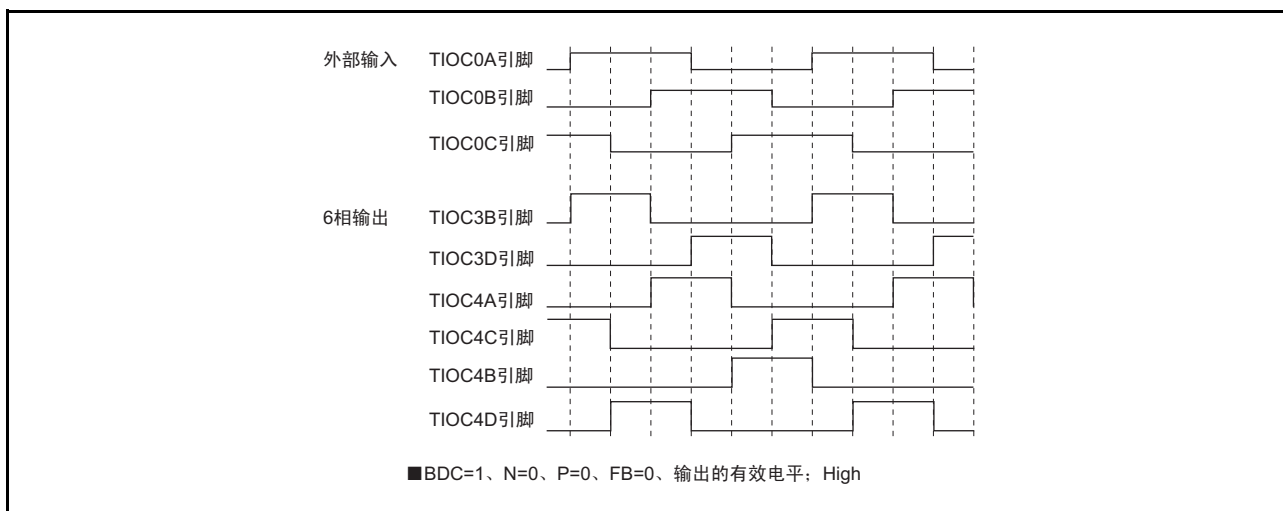


图 11.69 通过外部输入进行输出相转换的运行例 (1)

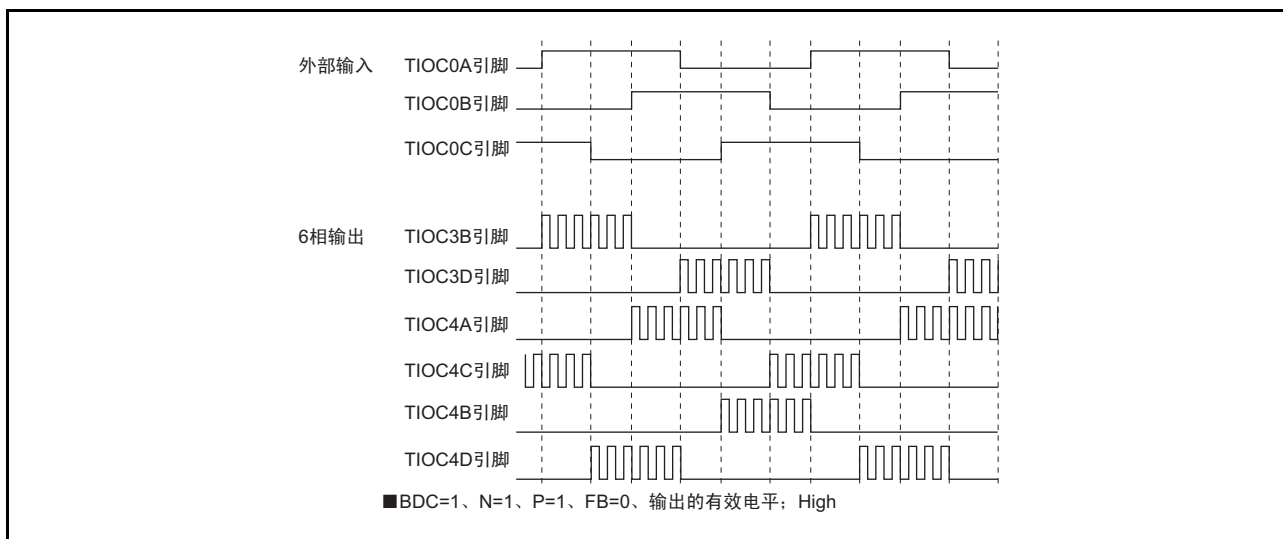


图 11.70 通过外部输入进行输出相转换的运行例 (2)

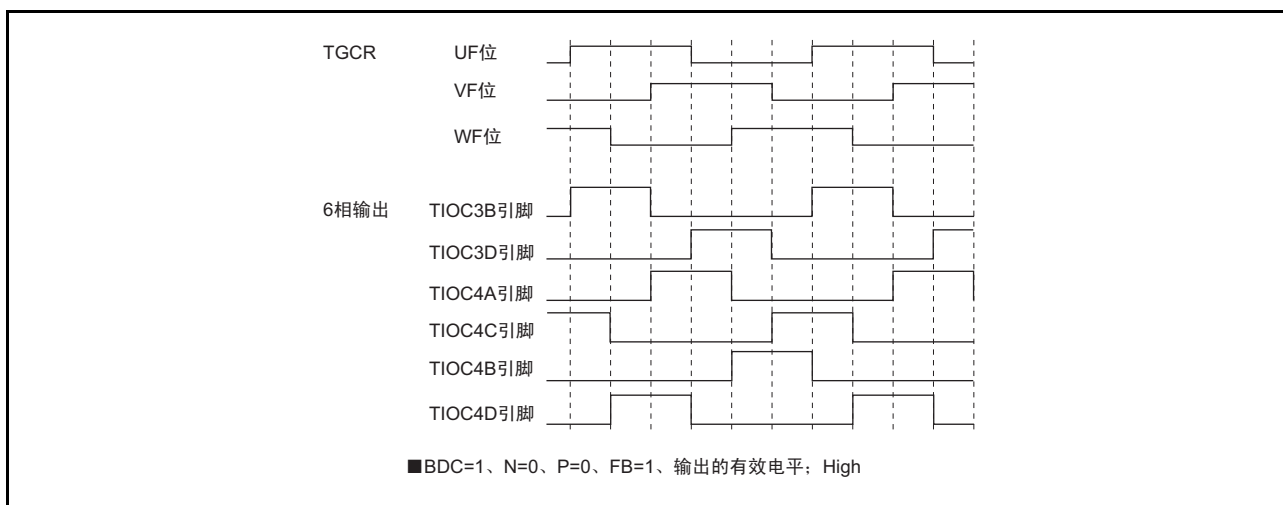


图 11.71 通过设定 UF、VF、WF 位进行输出相转换的运行例 (1)

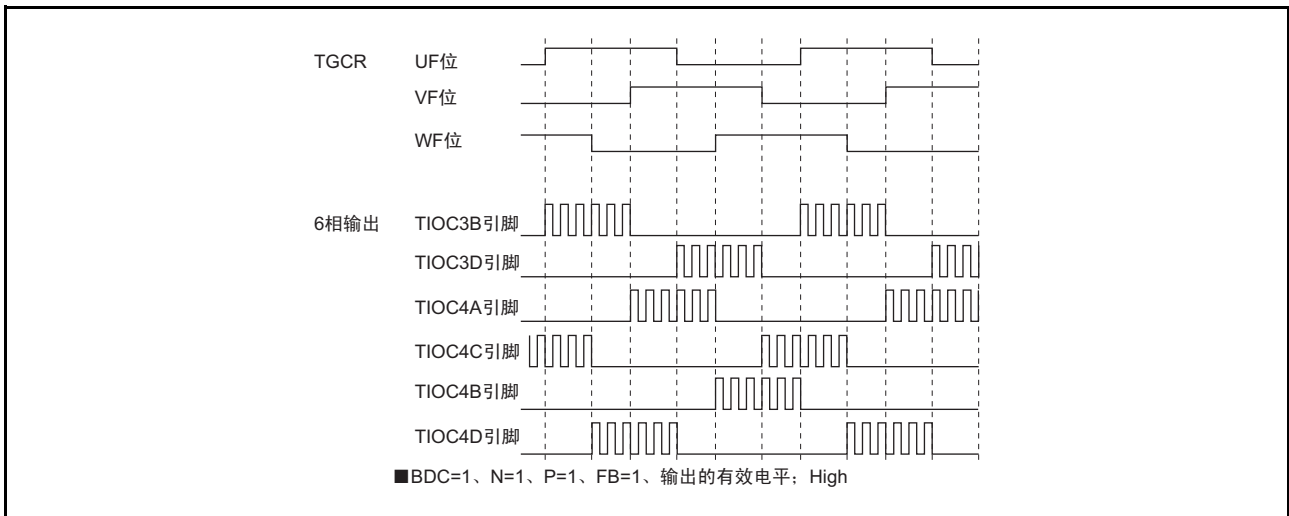


图 11.72 通过设定 UF、VF、WF 位进行输出相转换的运行例 (2)

### (r) A/D 转换开始请求的设定

在互补 PWM 模式中，能使用 TGRA\_3 的比较匹配、TCNT\_4 的下溢（波谷）和通道 3、4 以外通道的比较匹配进行 A/D 转换的开始请求。

如果使用 TGRA\_3 的比较匹配设定开始请求，就能在 TCNT\_3 的波峰开始 A/D 转换。

能通过将定时器的中断允许寄存器（TIER）的 TTGE 位设定为 1，设定 A/D 转换的开始请求，并能通过将 TIER\_4 的 TTGE2 位设定为 1，设定 TCNT\_4 的下溢（波谷）的 A/D 转换的开始请求。

### (3) 互补 PWM 模式的跳过中断功能

通过设定定时器的跳过中断设定寄存器（TITCR），最多能使通道 3 和通道 4 的 TGIA\_3（波峰的中断）和 TCIV\_4（波谷的中断）跳过 7 次中断。

通过设定定时器的缓冲传送寄存器（TBTER），能与缓冲寄存器到暂存器 / 比较寄存器的数据传送联动，跳过中断。有关和缓冲寄存器的联动，请参照“(c) 与跳过中断功能联动的缓冲传送控制”。

通过设定定时器的 A/D 转换请求控制寄存器（TADCR），能与 A/D 转换开始请求延迟功能联动的 A/D 转换开始请求，跳过中断。有关和 A/D 转换开始请求延迟功能的联动，请参照“11.4.9 A/D 转换开始请求的延迟功能”。

必须通过设定 TIER\_3、TIER\_4 寄存器，在将 TGIA\_3 和 TCIV\_4 中断请求设定为禁止状态后，并且在不会产生由比较匹配引起的 TGFA\_3 和 TCFV\_4 标志置位的状态下，设定定时器的跳过中断设定寄存器（TITCR）。另外，在更改跳过次数前，必须将 T3AEN 和 T4VEN 位设定为 0，清除跳过计数器。

### (a) 跳过中断功能的设定步骤例子

跳过中断功能的设定步骤例子和跳过中断次数的可更改期间分别如图 11.73 和图 11.74 所示。



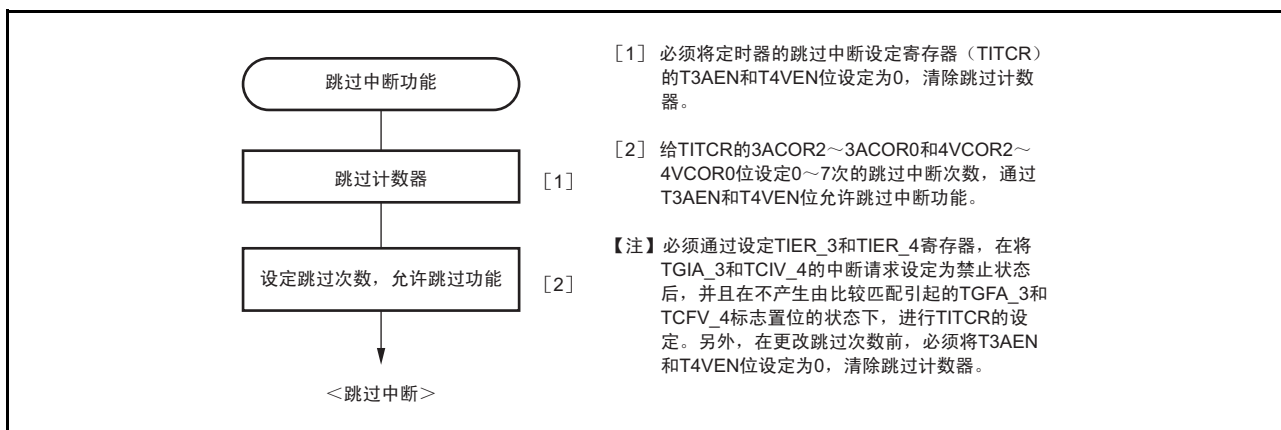


图 11.73 跳过中断功能的设定步骤例子

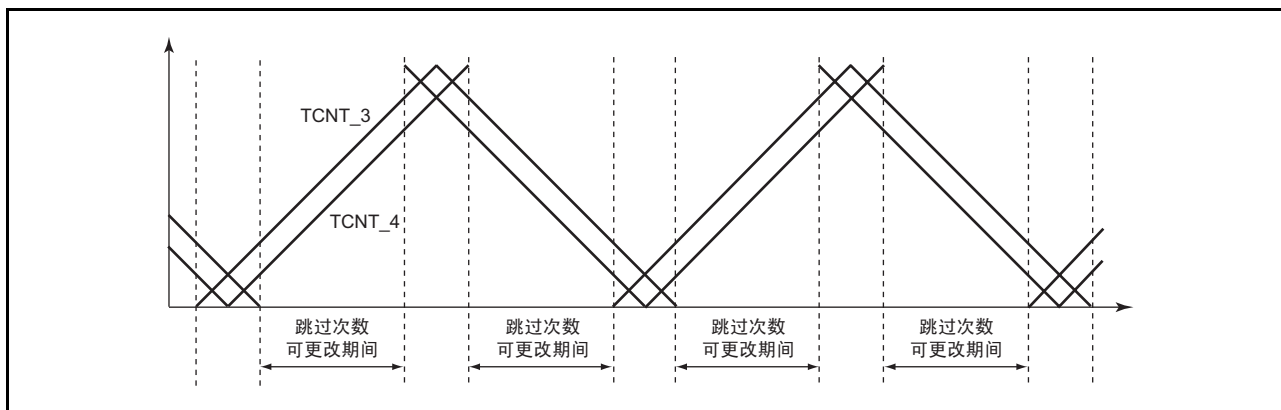


图 11.74 跳过中断次数的可更改期间

(b) 跳过中断功能的运行例

在通过定时器的跳过中断设定寄存器 (TITCR) 的 3ACOR 位将跳过中断次数设定为 3 次并将 T3AEN 位设定为 1 时, TGIA\_3 跳过中断功能的运行例如图 11.75 所示。

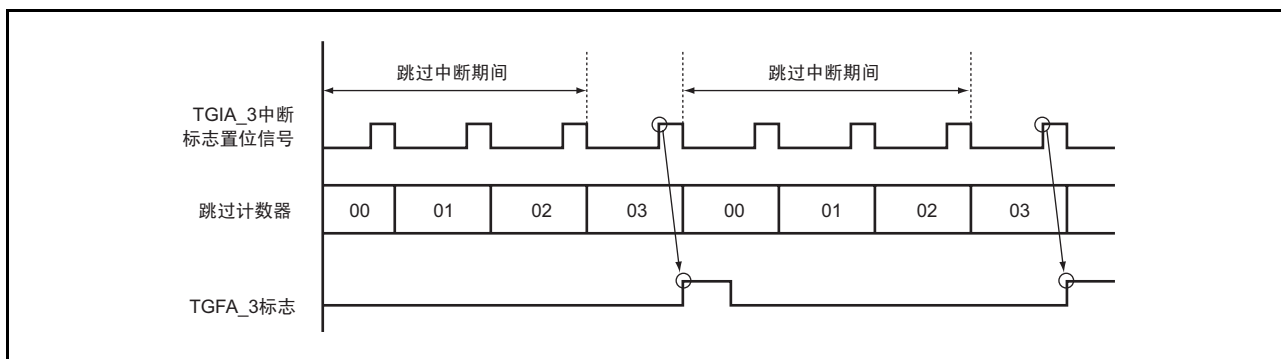


图 11.75 跳过中断功能的运行例

(c) 与跳过中断功能联动的缓冲传送控制

通过设定定时器的缓冲传送设定寄存器 (TBTER) 的 BTE1、BTE0 位, 可选择在互补 PWM 模式中是否进行缓冲寄存器到暂存器的缓冲传送, 或者设定是否与跳过中断功能联动。

设定为抑制缓冲传送 (BTE1=0、BTE0=1) 时的运行例子如图 11.76 所示。在此设定期间, 不将缓冲寄存器的值传送到暂存器。

设定为缓冲传送与跳过中断功能 (BTE1=1、BTE0=0) 联动时的运行例子如图 11.77 所示。此设定在缓冲传送允许期间以外, 不将缓冲寄存器的值传送到暂存器。

另外, 如果将定时器的跳过中断设定寄存器 (TITCR) 的 T3AEN 位或者 T4VEN 位或者 T3AEN/T4VEN 位设定为 1, 各缓冲传送允许期间就不同。TITCR 的 T3AEN、T4VEN 位的设定和缓冲传送允许期间的关系如图 11.78 所示。

**【注】** 本功能必须和跳过中断功能配合使用。

在禁止跳过中断功能时 (在将定时器的跳过中断设定寄存器 (TITCR) 的 T3AEN、T4VEN 位或者 TITCR 的跳过次数设定位 (3ACOR 和 4VCOR) 置 0 时), 必须设定为缓冲传送不与跳过中断功能联动 (将定时器的缓冲传送设定寄存器 (TBTER) 的 BTE1 置 0), 否则就不进行缓冲传送。

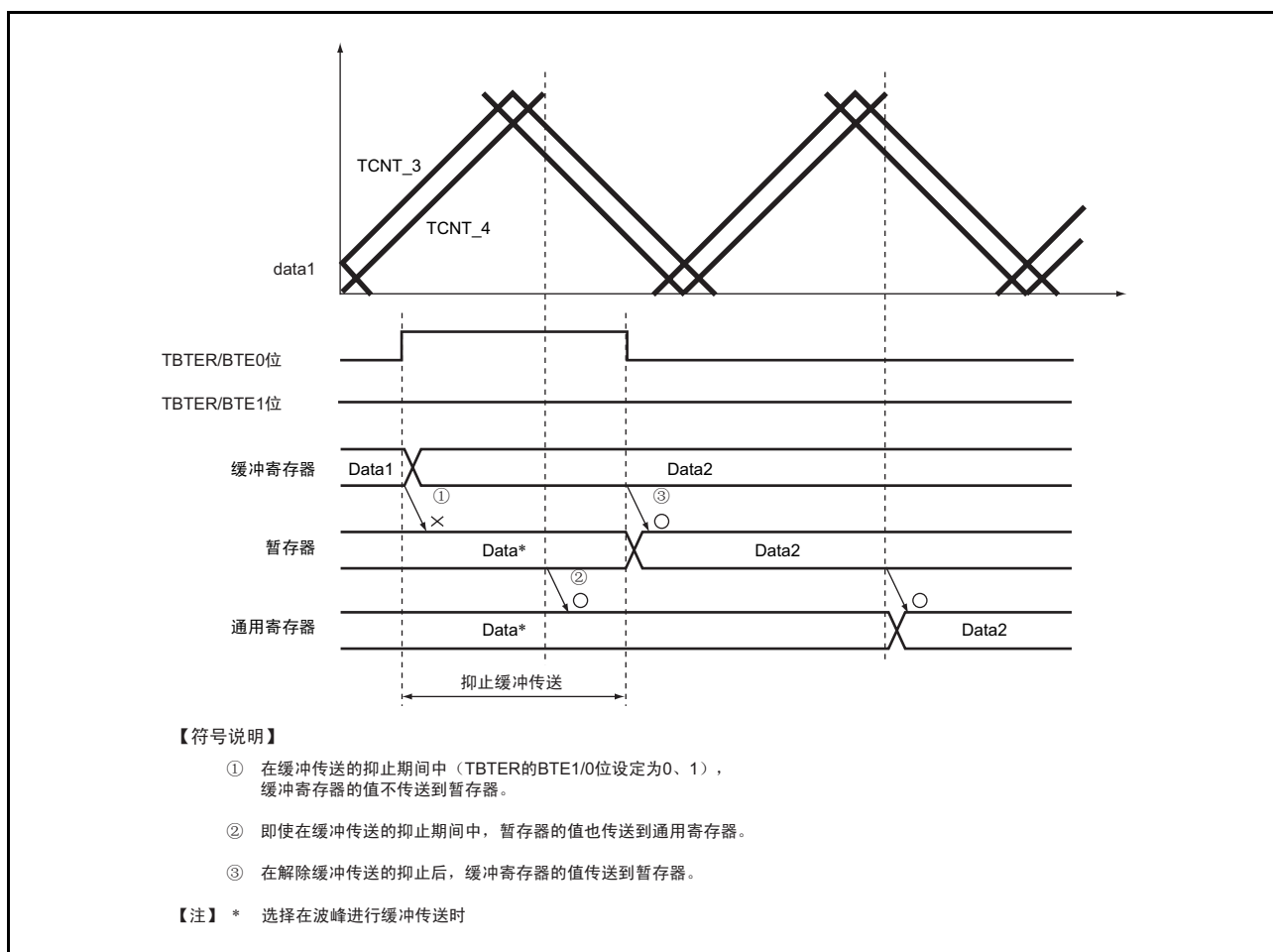


图 11.76 设定为抑制缓冲传送 (BTE1=0、BTE0=1) 时的运行例

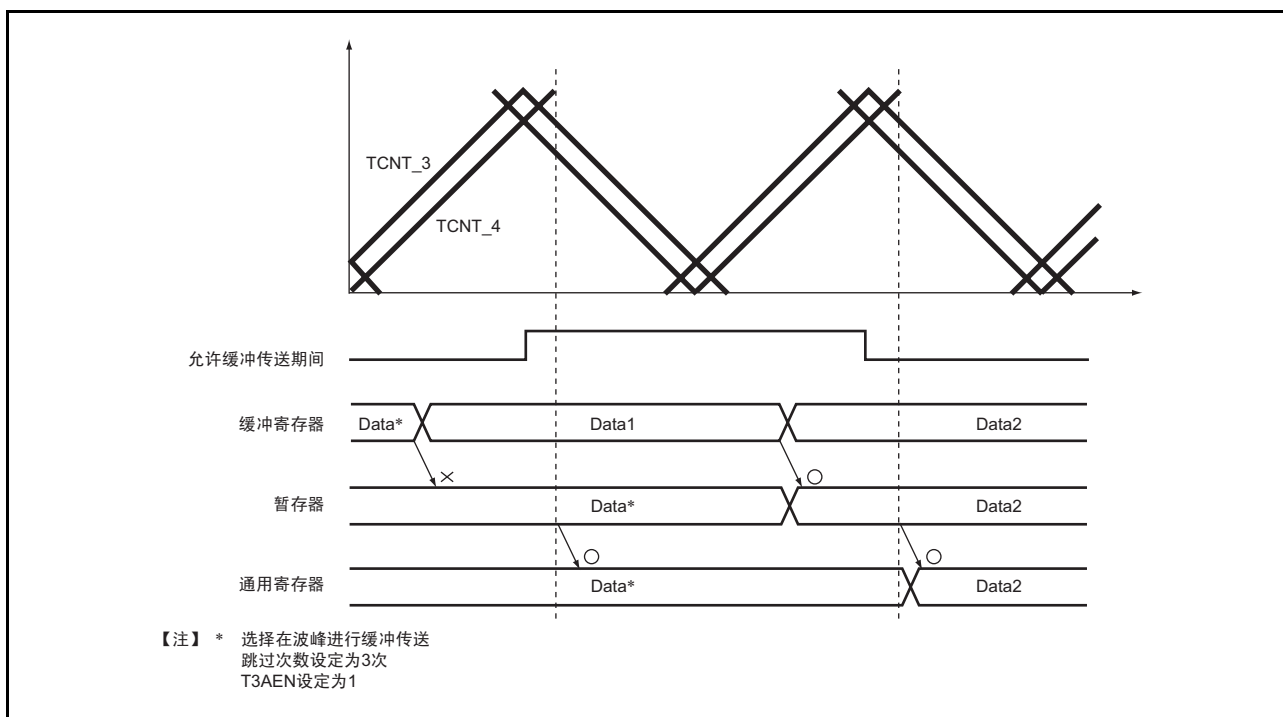


图 11.77 设定为缓冲传送与跳过中断功能联动 (BTE1=1、BTE0=0) 时的运行例

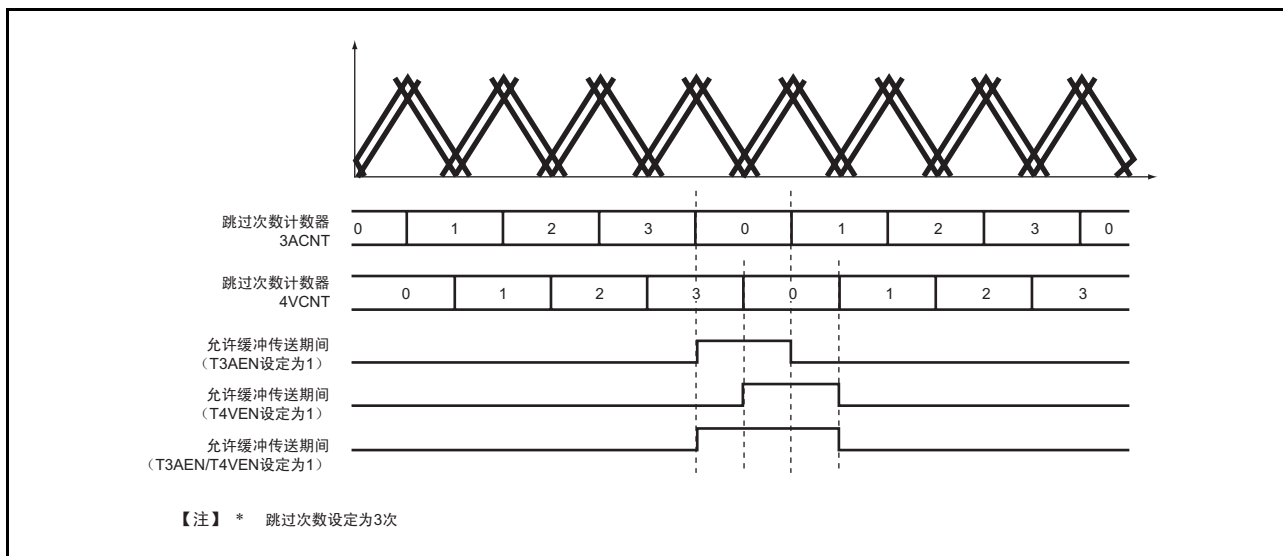


图 11.78 定时器的跳过中断设定寄存器 (TITCR) 的 T3AEN、T4VEN 位的设定和缓冲传送允许期间的关系

**(4) 互补 PWM 模式的输出保护功能**

互补 PWM 模式的输出具有以下保护功能:

**(a) 寄存器和计数器的误写防止功能**

在互补 PWM 模式使用的寄存器和计数器中,除了随时能改写的缓冲寄存器以外,可通过设定定时器的读写允许寄存器 (TRWER) 的 RWE 位,允许或者禁止由 CPU 存取模式寄存器、控制寄存器、比较寄存器和计数器。对象寄存器为通道 3 和通道 4 的一部分寄存器,如下所示:

TCR\_3 和 TCR\_4、TMDR\_3 和 TMDR\_4、TIORH\_3 和 TIORH\_4、TIORL\_3 和 TIORL\_4、TIER\_3 和 TIER\_4、TCNT\_3 和 TCNT\_4、TGRA\_3 和 TGRA\_4、TGRB\_3 和 TGRB\_4、TOER、TOCR、TGCR、TCDR、TDDR 共计 21 个寄存器

此功能通过将模式寄存器、控制寄存器和计数器等设定为禁止由 CPU 进行存取,能防止因 CPU 失控而产生的误写。在禁止存取的状态下读对象寄存器时,读的值不定并且写操作无效。

**(b) 由外部信号停止 PWM 输出的功能**

能通过输入所指定的外部信号,将 6 相 PWM 输出引脚自动置为高阻抗状态。

详细内容请参照“第 13 章 端口输出的允许 (POE)”。

**(c) 振荡停止时的 PWM 输出停止功能**

当检测出输入到本 LSI 的时钟已停止时,6 相 PWM 输出引脚就自动变为高阻抗状态。但是,如果时钟又开始振荡,就不保证引脚的状态。

详细内容请参照“4.7 振荡停止检测功能”。

**11.4.9 A/D 转换开始请求的延迟功能**

能通过设定通道 4 定时器的 A/D 转换开始请求的控制寄存器 (TADCR)、定时器的 A/D 转换开始请求的周期寄存器 (TADCORA\_4 和 TADCORB\_4) 和定时器的 A/D 转换开始请求的周期缓冲寄存器 (TADCOBRA\_4 和 TADCOBRB\_4),进行 A/D 转换的开始请求。

TCNT\_4 和 TADCORA\_4、TADCORB\_4 比较,当 TCNT\_4 和 TADCORA\_4、TADCORB\_4 一致时,A/D 转换开始请求的延迟功能就进行各自的 A/D 转换开始请求 (TRG4AN 和 TRG4BN)。

另外,通过设定 TADCR 的 ITA3AE、ITA4VE、ITB3AE 和 ITB4VE 位,能与跳过中断功能联动来跳过 A/D 转换的开始请求 (TRG4AN 和 TRG4BN)。

(a) A/D 转换开始请求的延迟功能的设定步骤例子

A/D 转换开始请求的延迟功能的设定步骤例子如图 11.79 所示。

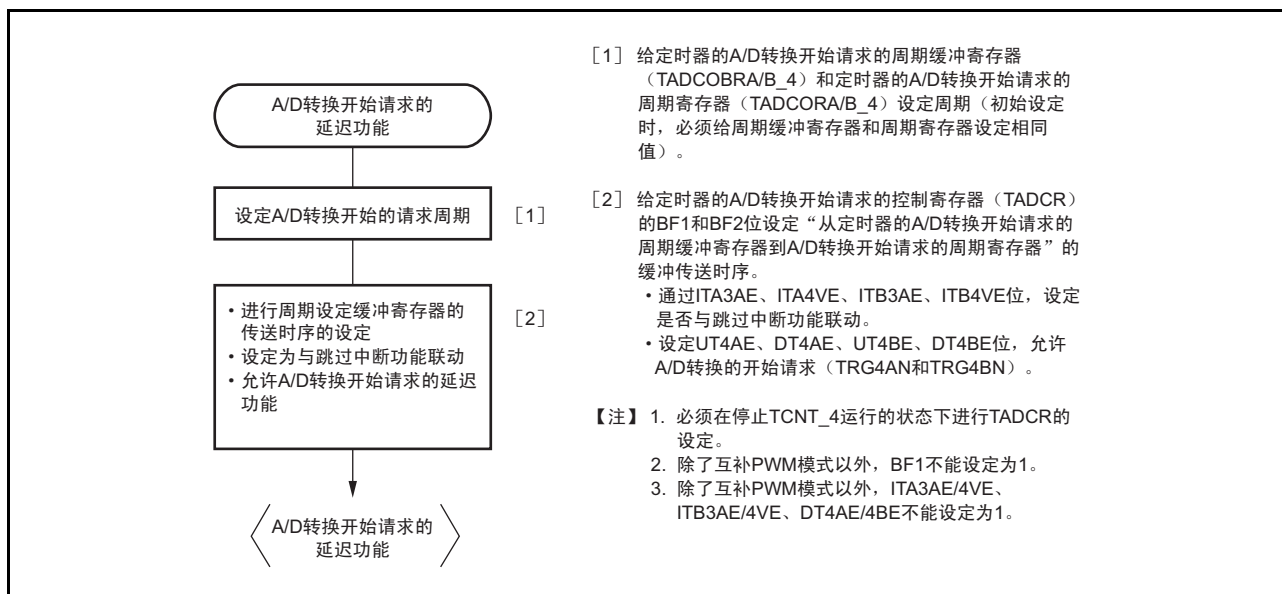


图 11.79 A/D 转换开始请求的延迟功能的设定步骤例子

(b) A/D 转换开始请求的延迟功能的基本运行例

将缓冲传送时序设定为 TCNT\_4 的波谷, 并且在 TCNT\_4 进行递减计数时输出 A/D 转换的开始请求信号 (TRG4AN) 的情况下, A/D 转换的开始请求信号 (TRG4AN) 的基本运行例子如图 11.80 所示。

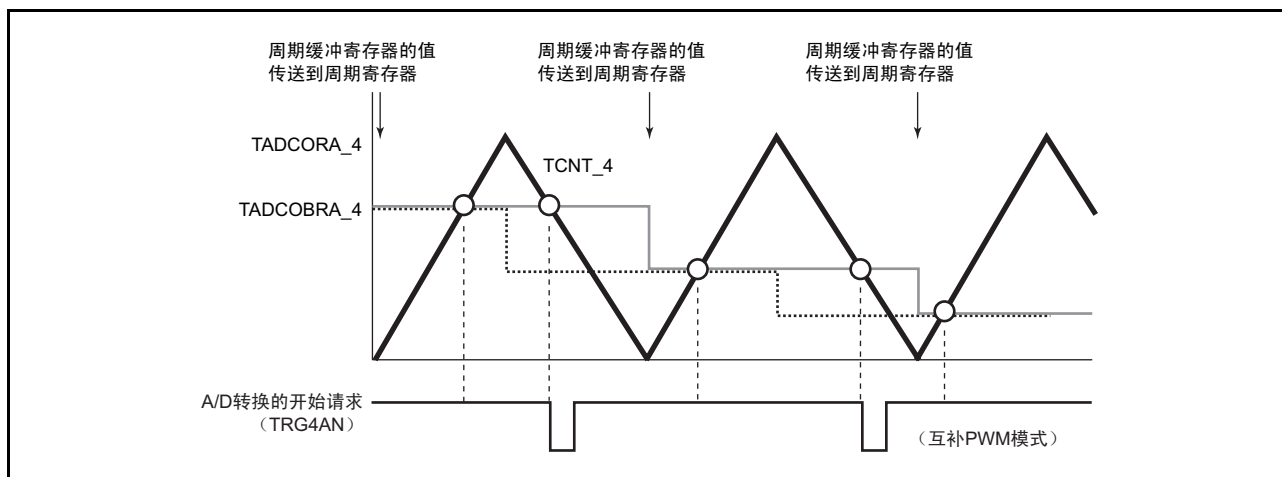


图 11.80 A/D 转换的开始请求信号 (TRG4AN) 的基本运行例

(c) 缓冲传送

通过将数据写到定时器的 A/D 转换开始请求的周期设定缓冲寄存器 (TADCOBRA/B\_4)，更新定时器的 A/D 转换开始请求的周期设定寄存器 (TADCORA/B\_4) 的数据。能通过设定定时器的 A/D 转换开始请求的控制寄存器 (TADCR\_4) 的 BF1、BF0 位，选择从定时器的 A/D 转换开始请求的周期设定缓冲寄存器到定时器的 A/D 转换开始请求的周期设定寄存器的传送时序。

(d) 与跳过中断功能联动的 A/D 转换开始请求的延迟功能

通过设定定时器的 A/D 转换开始请求的控制寄存器 (TADCR) 的 ITA3AE、ITA4VE、ITB3AE 和 ITB4VE 位，能与跳过中断功能联动进行 A/D 转换的开始请求 (TRG4AN 和 TRG4BN)。

在 TCNT\_4 进行递增计数或者递减计数时允许 TRG4AN 输出，并且在与跳过中断功能联动的情况下，A/D 转换的开始请求信号 (TRG4AN) 的运行例子如图 11.81 所示。

另外，在 TCNT\_4 进行递增计数时允许 TRG4AN 输出，并且在与跳过中断功能联动的情况下，A/D 转换的开始请求信号 (TRG4AN) 的运行例子如图 11.82 所示。

**【注】** 本功能必须和跳过中断功能配合使用。

在禁止跳过中断功能时 (将定时器的跳过中断设定寄存器 (TITCR) 的 T3AEN、T4VEN 位或者 TITCR 跳过次数的设定值 (3ACOR 和 4VCOR) 置 0 时)，必须设定为不与跳过中断功能联动 (将定时器的 A/D 转换开始请求的控制寄存器 (TADCR) 的 ITA3AE、ITA4VE、ITB3AE 和 ITB4VE 位设定为 0)。

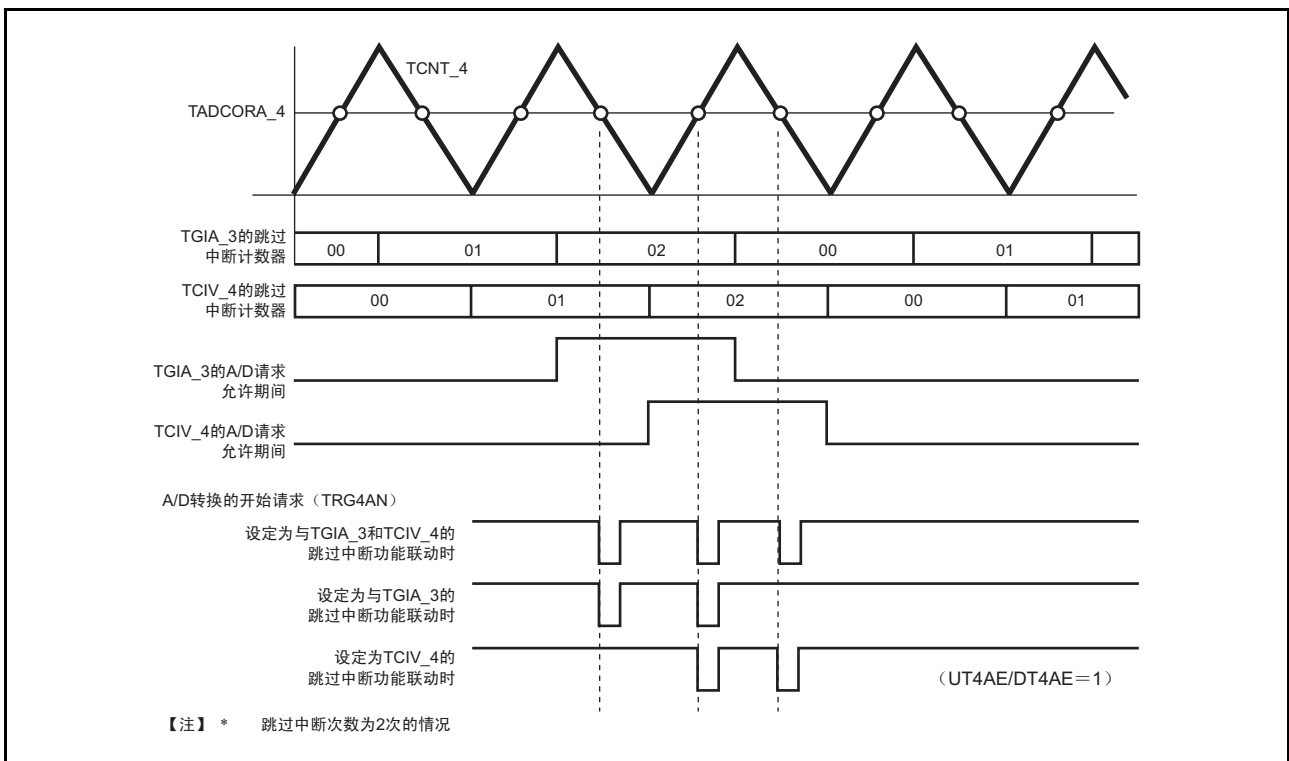


图 11.81 与跳过中断功能联动时 A/D 转换的开始请求信号 (TRG4AN) 的运行例

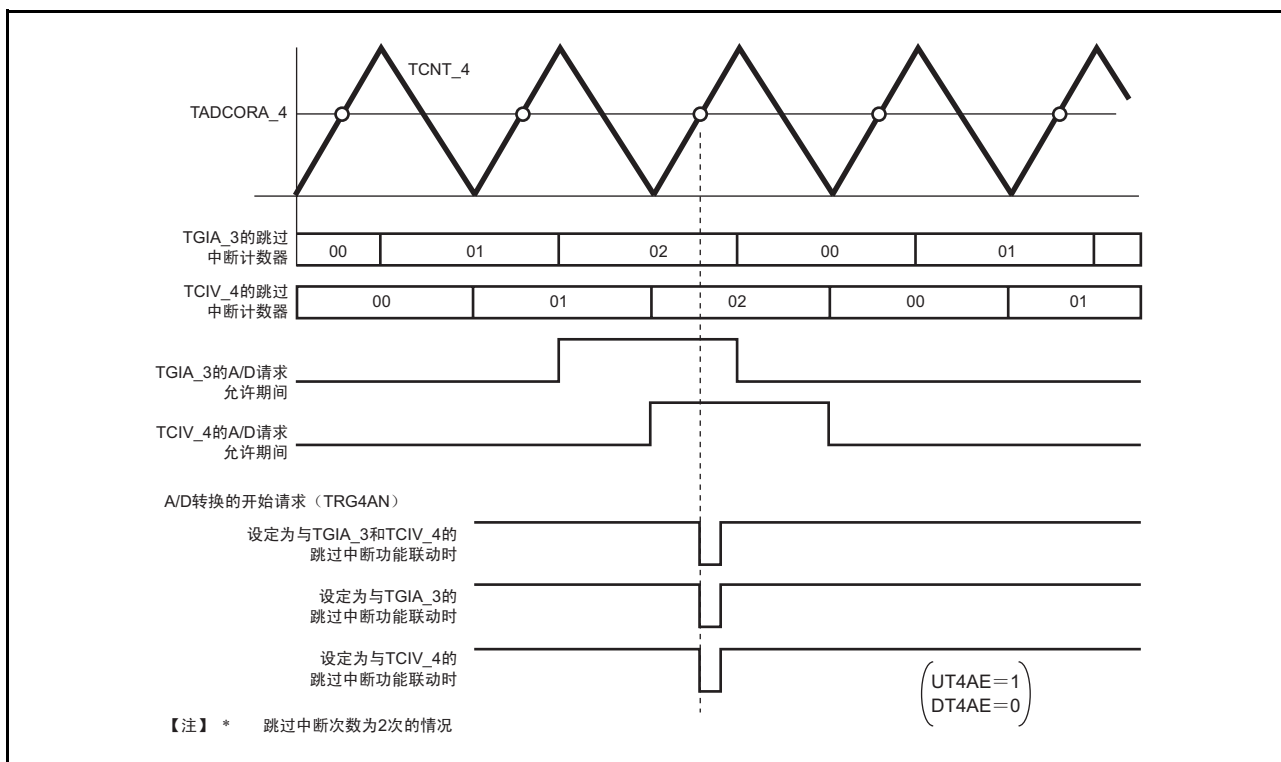


图 11.82 与跳过中断功能联动时 A/D 转换的开始请求信号 (TRG4AN) 的运行例

### 11.4.10 MTU2-MTU2S 的同步运行

#### (1) MTU2-MTU2S 计数器的同步开始

通过设定 MTU2 的 TCSYSTR 寄存器，能同步开始以不同时钟运行的 MTU2 和 MTU2S 的计数器。

#### (a) MTU2-MTU2S 计数器同步开始的设定步骤例子

计数器同步开始的设定步骤例子如图 11.83 所示。

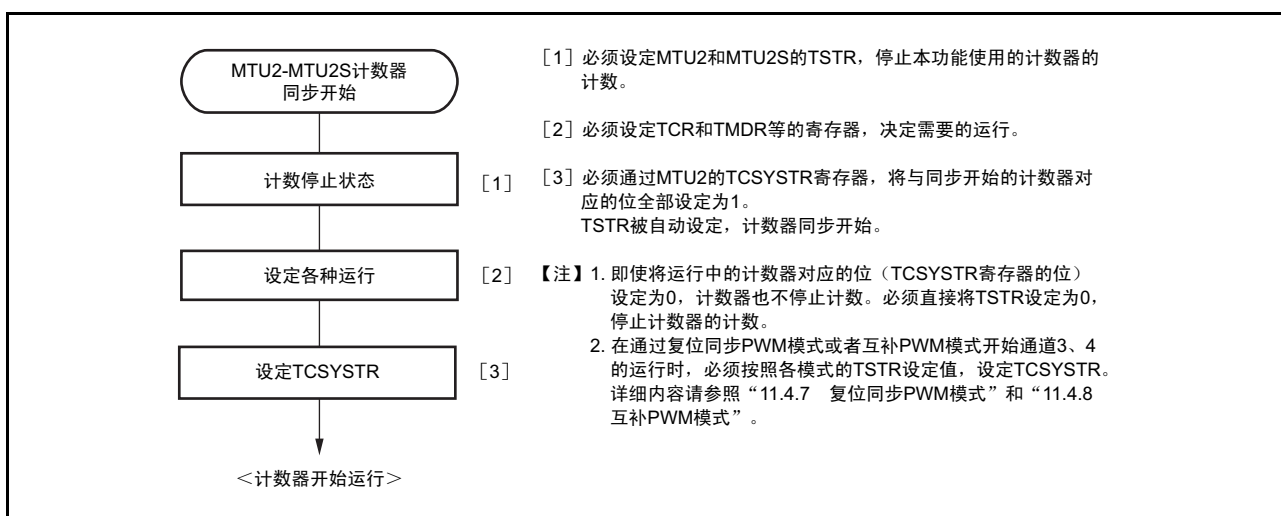


图 11.83 计数器同步开始的设定步骤例子

(b) 计数器同步开始的运行例

MTU2 和 MTU2S 的时钟频率比为 1:1、1:2、1:3、1:4 时的计数器同步开始的运行例子分别如图 11.84、图 11.85、图 11.86 和图 11.87 所示。

在这些例子中，计数时钟设定为 MPφ/1。

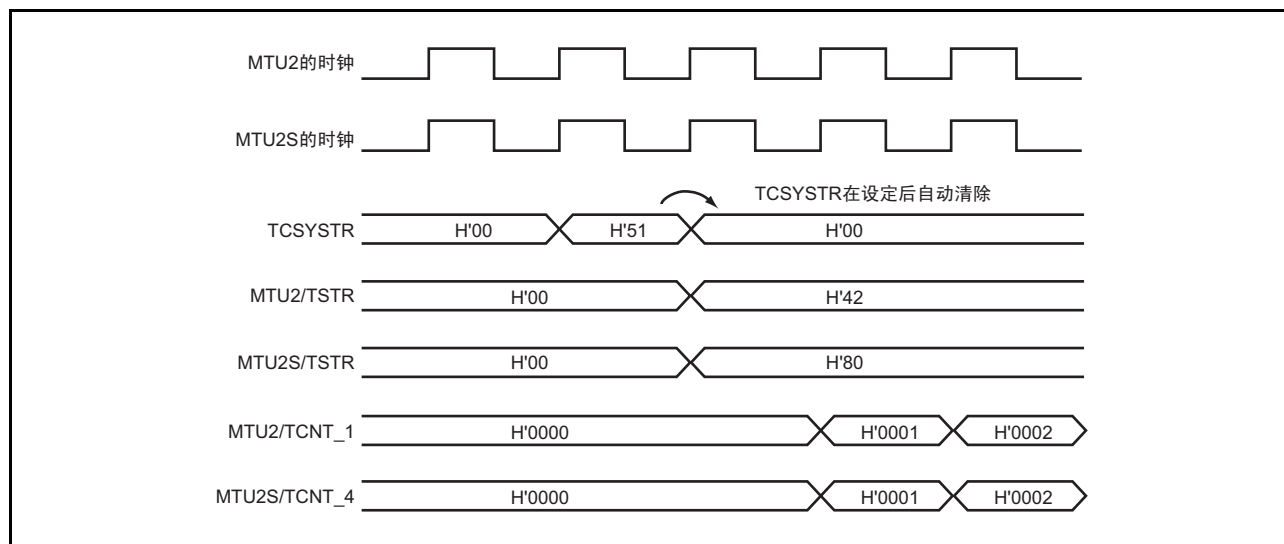


图 11.84 计数器同步开始的运行例 (MTU2 和 MTU2S 的时钟频率比为 1:1)

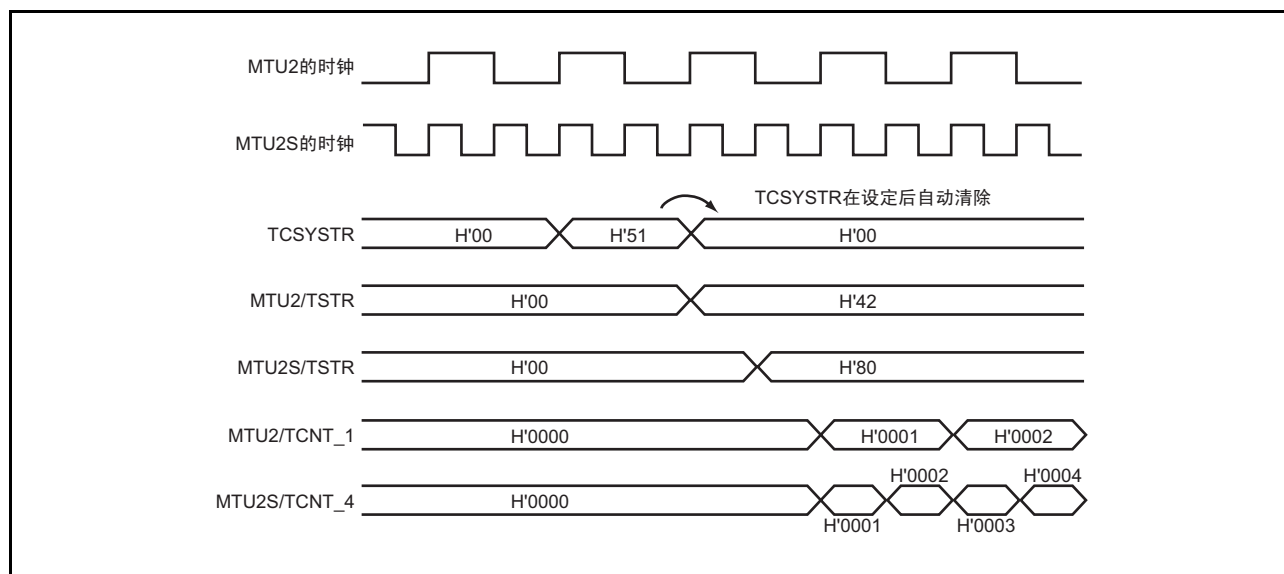


图 11.85 计数器同步开始的运行例 (MTU2 和 MTU2S 的时钟频率比为 1:2)



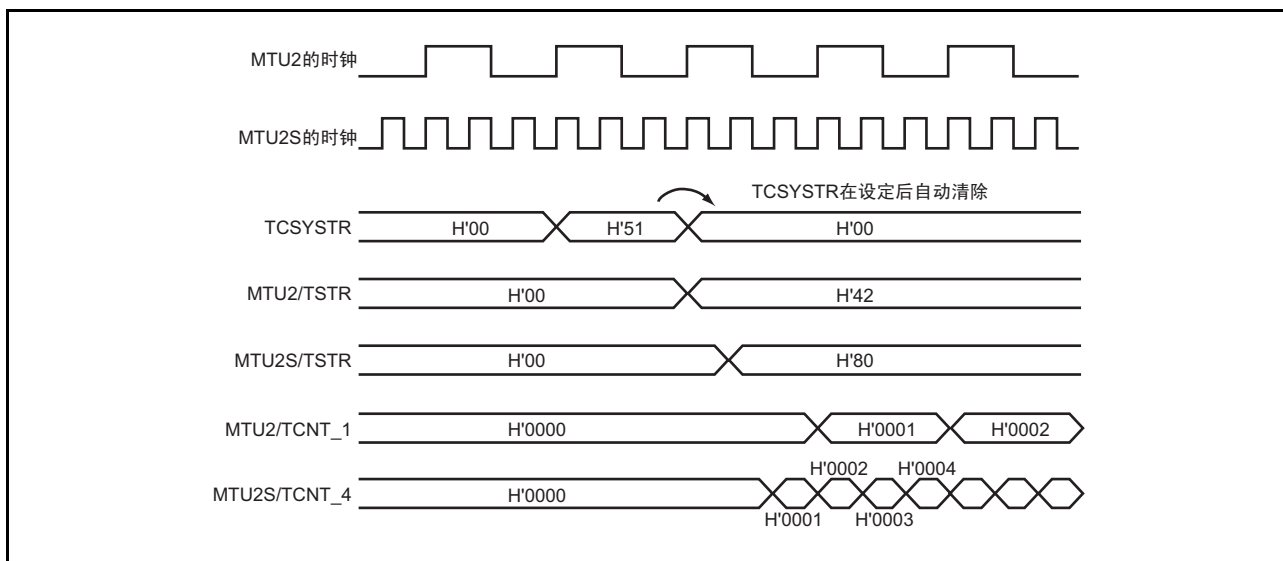


图 11.86 计数器同步开始的运行例 (MTU2 和 MTU2S 的时钟频率比为 1:3)

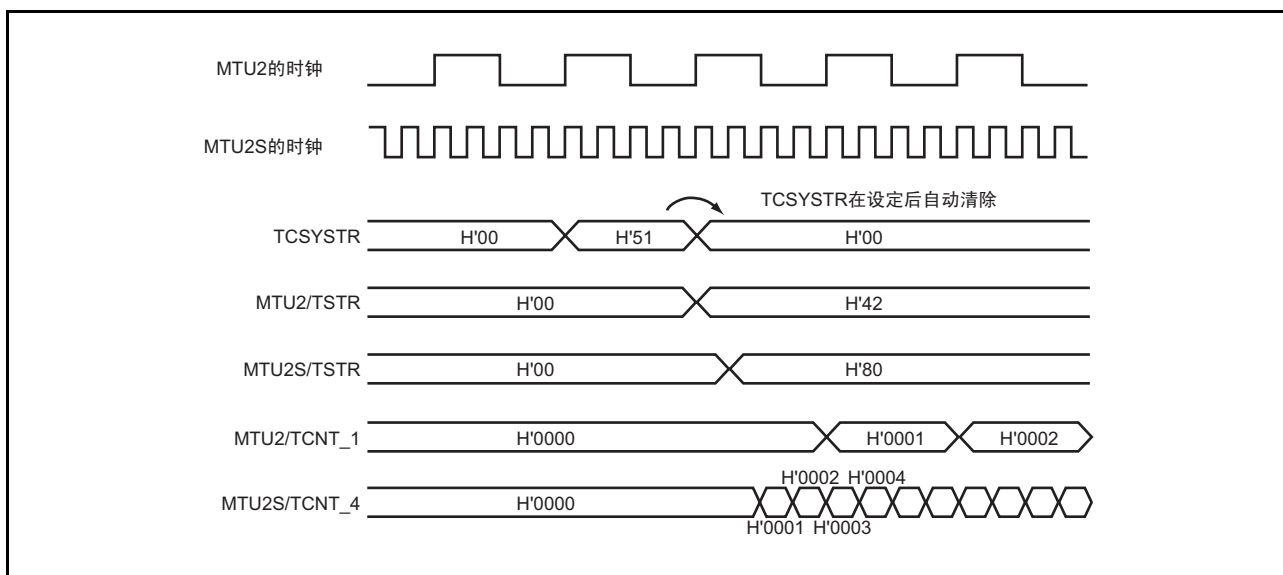


图 11.87 计数器同步开始的运行例 (MTU2 和 MTU2S 的时钟频率比为 1:4)

(2) 利用 MTU2 标志的置位源进行 MTU2S 计数器的清除 (MTU2-MTU2S 计数器同步清除)

通过设定 TSYCR\_3 寄存器, MTU2S 能利用 MTU2 的 TSR\_0 ~ TSR\_2 标志的置位源, 进行计数器的清除。

(a) 利用 MTU2 标志的置位源进行 MTU2S 计数器清除的设定步骤例子

利用 MTU2 标志的置位源进行 MTU2S 计数器清除的设定步骤例子如图 11.88 所示。

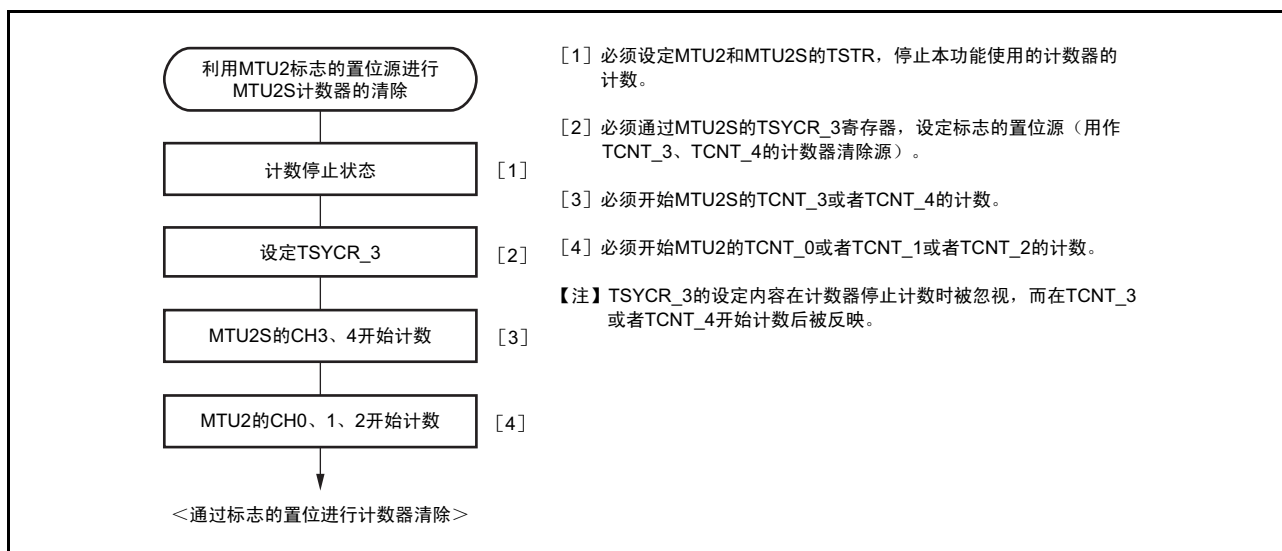


图 11.88 利用 MTU2 标志的置位源进行 MTU2S 计数器清除的设定步骤例子

(b) 利用 MTU2 标志的置位源进行 MTU2S 计数器清除的运行例

利用 MTU2 标志的置位源进行 MTU2S 计数器清除的运行例子如图 11.89 和图 11.90 所示。

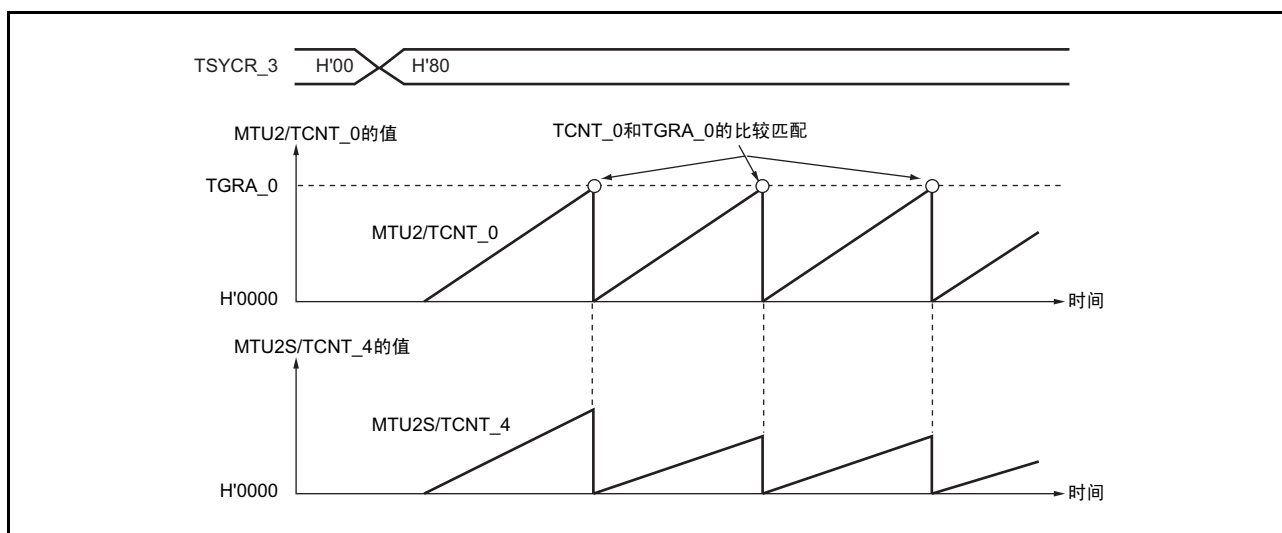


图 11.89 利用 MTU2 标志的置位源进行 MTU2S 计数器清除的运行例 (1)

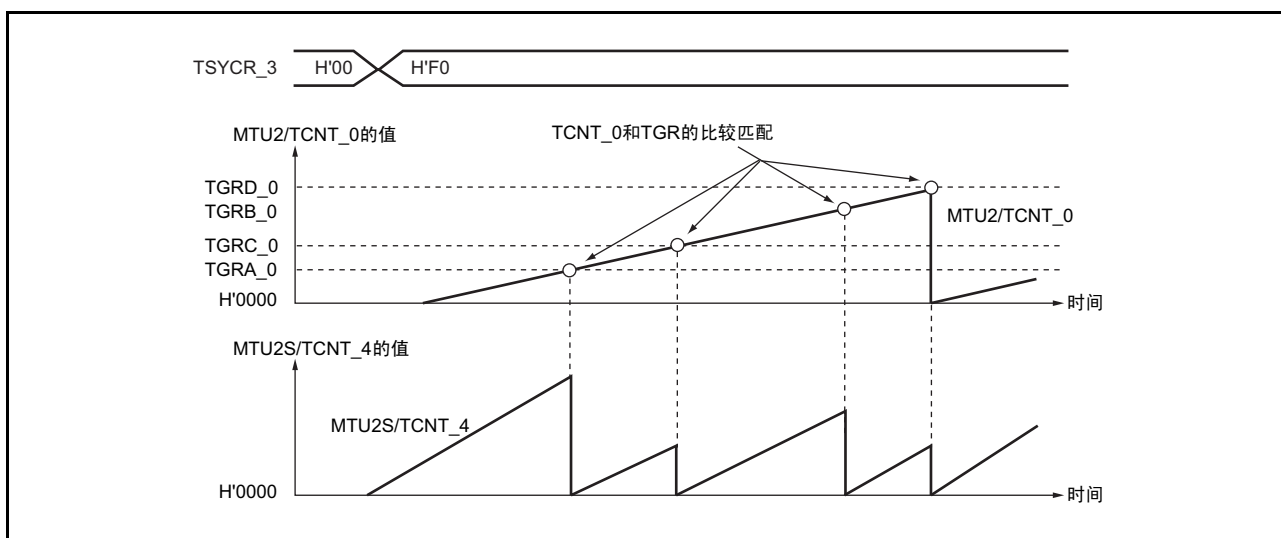


图 11.90 利用 MTU2 标志的置位源进行 MTU2S 计数器清除的运行例 (2)

### 11.4.11 外部脉宽的测量功能

通道 5 最多能测量 3 个外部脉宽。

#### (1) 测量外部脉宽的设定步骤例子

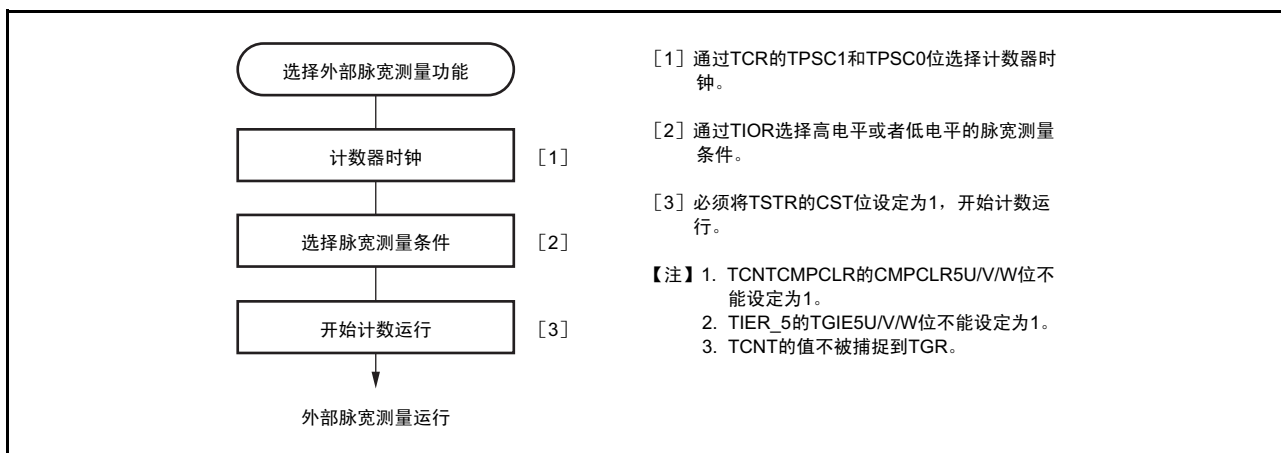


图 11.91 测量外部脉宽的设定步骤例子

#### (2) 测量外部脉宽的运行例

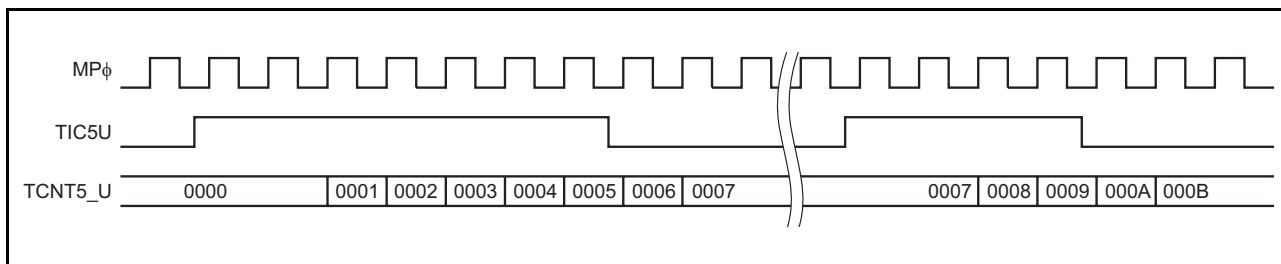


图 11.92 测量外部脉宽的运行例 (测量高电平的脉宽)

### 11.4.12 空载时间的补偿功能

通过测量输出波形的延迟并将此延迟反映到占空比，能将外部脉宽的测量功能用作互补 PWM 运行时 PWM 输出波形的空载时间补偿功能。

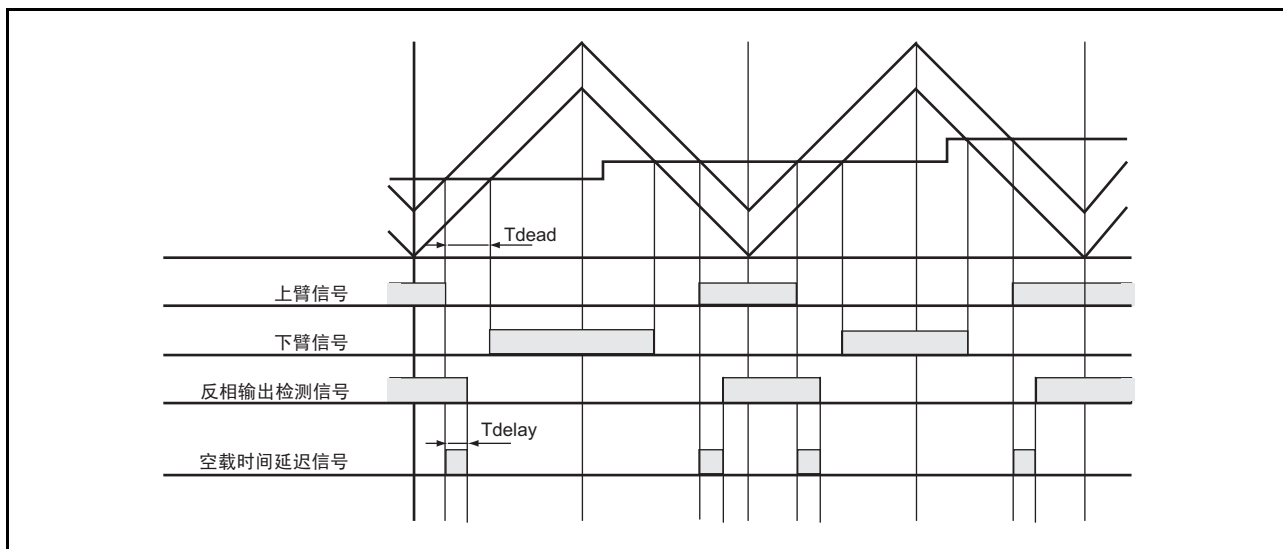


图 11.93 互补 PWM 运行时的空载时间延迟

#### (1) 空载时间补偿功能的设定步骤例子

使用通道 5 的 3 个计数器的空载时间补偿功能的设定步骤例子如图 11.94 所示。

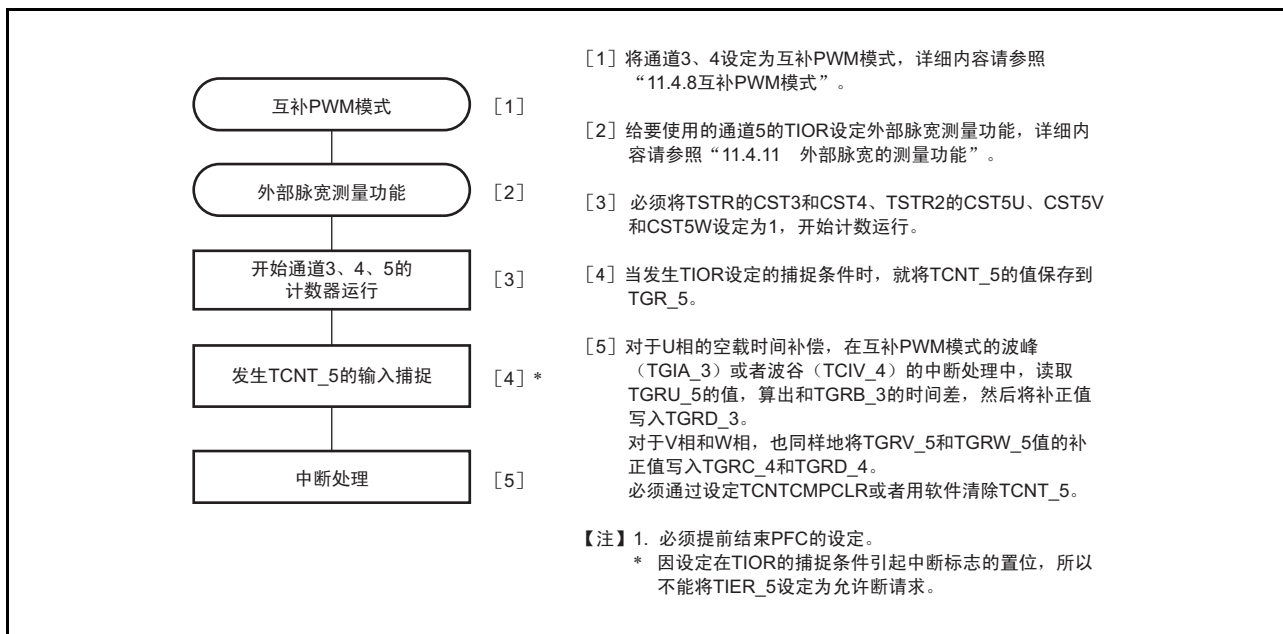


图 11.94 空载时间补偿功能的设定步骤例子

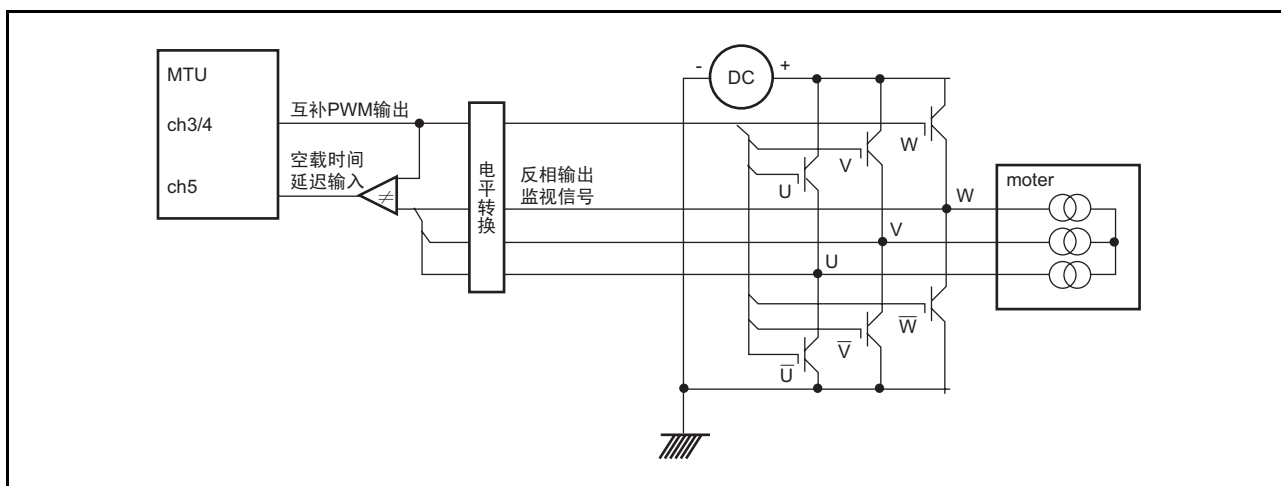


图 11.95 构成马达控制电路的例子

### 11.4.13 互补 PWM 在“波峰 / 波谷”的 TCNT 捕捉运行

当互补 PWM 运行时，在“波峰、波谷、波峰和波谷”将 TCNT 的值保存到 TGR。通过 TIOR 选择要保存到 TGR 的时序转换。

TCNT 用作自由运行计数器（不被清除）并在设定的“波峰 / 波谷”对 TGR 进行捕捉的运行例子如图 11.96 所示。

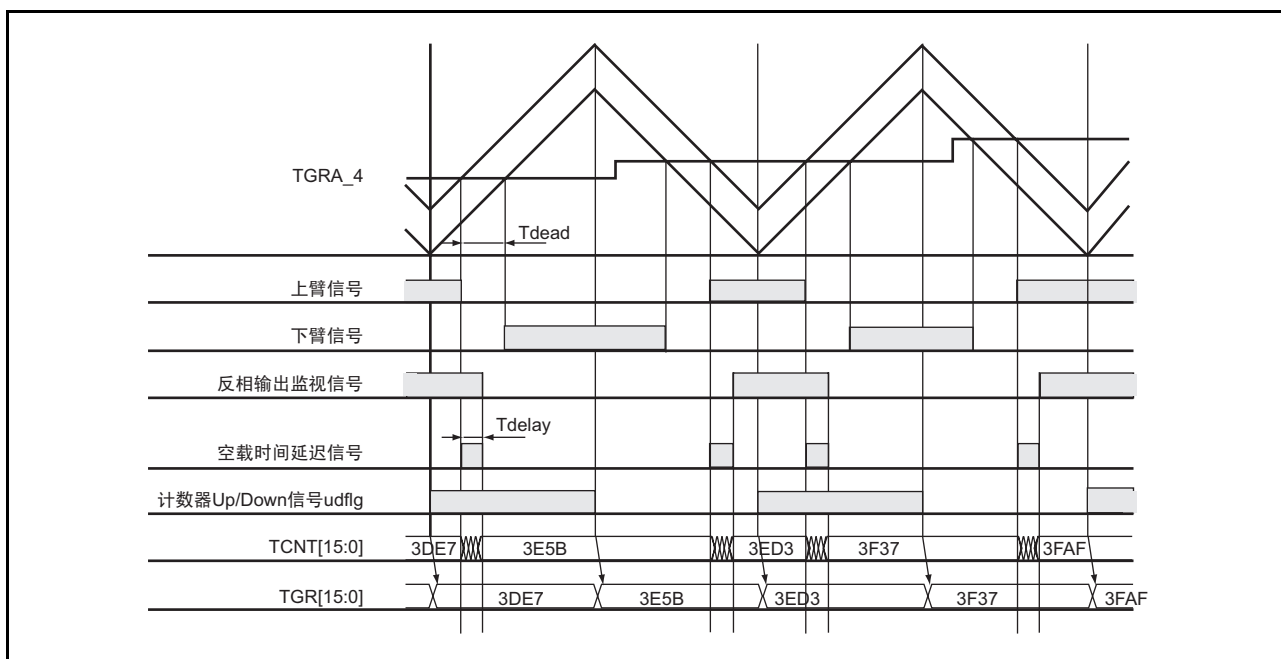


图 11.96 互补 PWM 在“波峰 / 波谷”的 TCNT 捕捉运行

## 11.5 中断源

### 11.5.1 中断源和优先级

MTU2 的中断源有 TGR 的输入捕捉 / 比较匹配、TCNT 的上溢和 TCNT 的下溢 3 种。因为各中断源有各自专用的状态标志和允许 / 禁止位，所以能独立允许或者禁止中断请求信号的产生。

当发生中断源时，TSR 对应的状态标志就被置 1。此时，如果 TIER 对应的允许 / 禁止位已被置 1，就请求中断。通过将状态标志清 0 解除中断请求。

能通过中断控制器更改通道之间的优先级，并且通道内的优先级是固定的。详细内容请参照“第 6 章 中断控制器 (INTC)”。

MTU2 的中断源一览表如表 11.57 所示。

表 11.57 MTU2 中断源

通道	名称	中断源	中断标志	DMAC 的启动	DTC 的启动	优先级
0	TGIA_0	TGRA_0 的输入捕捉 / 比较匹配	TGFA_0	可	可	高 ↑
	TGIB_0	TGRB_0 的输入捕捉 / 比较匹配	TGFB_0	不可	可	
	TGIC_0	TGRC_0 的输入捕捉 / 比较匹配	TGFC_0	不可	可	
	TGID_0	TGRD_0 的输入捕捉 / 比较匹配	TGFD_0	不可	可	
	TCIV_0	TCNT_0 的上溢	TCFV_0	不可	不可	
	TGIE_0	TGRE_0 的比较匹配	TGFE_0	不可	不可	
	TGIF_0	TGRF_0 的比较匹配	TGFF_0	不可	不可	
1	TGIA_1	TGRA_1 的输入捕捉 / 比较匹配	TGFA_1	可	可	↓
	TGIB_1	TGRB_1 的输入捕捉 / 比较匹配	TGFB_1	不可	可	
	TCIV_1	TCNT_1 的上溢	TCFV_1	不可	不可	
	TCIU_1	TCNT_1 的下溢	TCFU_1	不可	不可	
2	TGIA_2	TGRA_2 的输入捕捉 / 比较匹配	TGFA_2	可	可	
	TGIB_2	TGRB_2 的输入捕捉 / 比较匹配	TGFB_2	不可	可	
	TCIV_2	TCNT_2 的上溢	TCFV_2	不可	不可	
	TCIU_2	TCNT_2 的下溢	TCFU_2	不可	不可	
3	TGIA_3	TGRA_3 的输入捕捉 / 比较匹配	TGFA_3	可	可	
	TGIB_3	TGRB_3 的输入捕捉 / 比较匹配	TGFB_3	不可	可	
	TGIC_3	TGRC_3 的输入捕捉 / 比较匹配	TGFC_3	不可	可	
	TGID_3	TGRD_3 的输入捕捉 / 比较匹配	TGFD_3	不可	可	
	TCIV_3	TCNT_3 的上溢	TCFV_3	不可	不可	
4	TGIA_4	TGRA_4 的输入捕捉 / 比较匹配	TGFA_4	可	可	
	TGIB_4	TGRB_4 的输入捕捉 / 比较匹配	TGFB_4	不可	可	
	TGIC_4	TGRC_4 的输入捕捉 / 比较匹配	TGFC_4	不可	可	
	TGID_4	TGRD_4 的输入捕捉 / 比较匹配	TGFD_4	不可	可	
	TCIV_4	TCNT_4 的上溢 / 下溢	TCFV_4	不可	可	
5	TGIU_5	TGRU_5 的输入捕捉 / 比较匹配	TGFU_5	不可	可	
	TGIV_5	TGRV_5 的输入捕捉 / 比较匹配	TGFV_5	不可	可	
	TGIW_5	TGRW_5 的输入捕捉 / 比较匹配	TGFW_5	不可	可	

【注】 表示有关复位后的初始状态。能通过中断控制器更改通道之间的优先级。

### (1) 输入捕捉 / 比较匹配的中断

在通过各通道的 TGR 输入捕捉 / 比较匹配将 TSR 的 TGF 标志置 1 时，如果 TIER 的 TGIE 位已被置 1，就请求中断。通过将 TGF 标志清 0 解除中断请求。在 MTU2 中，通道 0 有 6 个，通道 3、4 各有 4 个，通道 1、2 各有 2 个，通道 5 各有 3 个，共 21 个输入捕捉 / 比较匹配的中断。通道 0 的 TGFE\_0 和 TGFF\_0 标志在输入捕捉时不被置位。

### (2) 上溢中断

在通过各通道的 TCNT 上溢将 TSR 的 TCFV 标志置 1 时，如果 TIER 的 TCIEV 位已被置 1，就请求中断。通过将 TCFV 标志清 0 解除中断请求。在 MTU2 中，各通道有 1 个，共 5 个上溢中断。

### (3) 下溢中断

在通过各通道的 TCNT 下溢将 TSR 的 TCFU 标志置 1 时，如果 TIER 的 TCIEU 位已被置 1，就请求中断。通过将 TCFU 标志清 0 解除中断请求。在 MTU2 中，通道 1、2 各有 1 个，共 2 个下溢中断。

## 11.5.2 DTC/DMAC 的启动

### (1) DTC 的启动

能通过各通道的 TGR 的输入捕捉 / 比较匹配中断或者通道 4 的上溢中断，启动 DTC。详细内容请参照“第 8 章 数据传送控制器 (DTC)”。

MTU2 有 20 个输入捕捉 / 比较匹配中断和上溢中断能作为 DTC 的启动源。其中，通道 0、3 各有 4 个、通道 1、2 各有 2 个、通道 4 有 5 个、通道 5 有 3 个。

### (2) DMAC 的启动

能通过各通道的 TGRA 的输入捕捉 / 比较匹配中断，启动 DMAC。详细内容请参照“第 10 章 直接存储器存取控制器 (DMAC)”。

MTU2 有 5 个 TGRA 寄存器的输入捕捉 / 比较匹配中断能作为 DMAC 的启动源。其中，通道 0 ~ 4 各有 1 个。

如果通过 MTU2 启动 DMAC，就在 DMAC 请求内部总线权时清除启动源。所以根据内部总线的状态，即使清除了启动源，也会发生 DMAC 传送进入开始等待状态的期间。另外，在通过 MTU2 进行 DMAC 突发传送时，需要设定总线功能扩展寄存器 (BSCEHR)。详细内容请参照“9.4.8 总线功能扩展寄存器 (BSCEHR)”。

## 11.5.3 A/D 转换器的启动

MTU2 能通过以下 3 种方法启动 A/D 转换器。

各中断源和 A/D 转换开始请求的对应如表 11.58 所示。

### (1) TGRA 的输入捕捉 / 比较匹配和在互补 PWM 模式中的 TCNT\_4 波谷启动 A/D 转换器

能通过各通道的 TGRA 输入捕捉 / 比较匹配启动 A/D 转换器。另外，在给 TIER\_4 的 TTGE2 位设定 1 的状态下，如果互补 PWM 运行，即使在 TCNT\_4 为波谷 (TCNT\_4=H'0000) 时也能启动 A/D 转换器。

在以下所示的条件中，对 A/D 转换器产生 A/D 转换的开始请求 TRGAN。

- 在通过各通道的 TGRA 输入捕捉 / 比较匹配将 TSR 的 TGFA 标志置 1 时，TIER 的 TTGE 位已被置 1
- 在给 TIER\_4 的 TTGE2 位设定 1 的状态下，当互补 PWM 运行并且 TCNT\_4 为波谷 (TCNT\_4=H'0000) 时

此时，如果已在 A/D 转换器侧选择 MTU2 的转换开始触发 TRGAN，就开始进行 A/D 转换。



## (2) 通过 TCNT\_0 和 TGRE\_0 的比较匹配启动 A/D 转换器

通过通道 0 的 TCNT\_0 和 TGRE\_0 的比较匹配，能产生 A/D 转换的开始请求 TRG0N，启动 A/D 转换器。

在通过通道 0 的 TCNT\_0 和 TGRE\_0 的比较匹配将 TSR2\_0 的 TGFE 标志置 1 时，如果 TIER2\_0 的 TTGE2 位已被置 1，就对 A/D 转换器产生 A/D 转换的开始请求 TRG0N。此时，如果已在 A/D 转换器侧选择 MTU2 的转换开始触发 TRG0N，就开始进行 A/D 转换。

## (3) 通过 A/D 转换开始请求的延迟功能启动 A/D 转换器

如果给 A/D 转换开始请求的控制寄存器 (TADCRC) 的 TAD4AE、TAD4BE 位设定 1，能通过 TADCORA、TADCORB 和 TCNT\_4 的一致产生 TRG4AN 和 TRG4BN，启动 A/D 转换器。详细内容请参照“11.4.9 A/D 转换开始请求的延迟功能”。

当发生 TRG4AN 时，如果已在 A/D 转换器侧选择 MTU2 的转换开始触发 TRG4AN，就开始进行 A/D 转换；当发生 TRG4BN 时，如果已在 A/D 转换器侧选择 MTU2 的转换开始触发 TRG4BN，就开始进行 A/D 转换。

表 11.58 各中断源和 A/D 转换开始请求的对应

对象	中断源	A/D 转换的开始请求
TGRA_0 和 TCNT_0	输入捕捉 / 比较匹配	TRGAN
TGRA_1 和 TCNT_1		
TGRA_2 和 TCNT_2		
TGRA_3 和 TCNT_3		
TGRA_4 和 TCNT_4		
TCNT_4	互补 PWM 模式中 TCNT_4 的波谷	
TGRE_0 和 TCNT_0	比较匹配	TRG0N
TADCORA 和 TCNT_4		TRG4AN
TADCORB 和 TCNT_4		TRG4BN

## 11.6 运行时序

### 11.6.1 输入 / 输出时序

#### (1) TCNT 的计数时序

内部时钟运行时的 TCNT 计数时序如图 11.97 和图 11.98、外部时钟运行（正常模式和相位计数模式）时的 TCNT 计数时序如图 11.99 和图 11.100 所示。

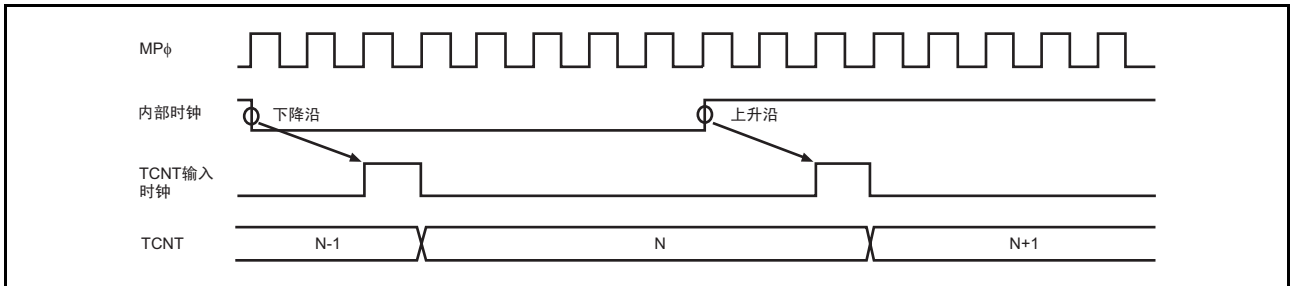


图 11.97 内部时钟运行时的计数时序（通道 0 ~ 4）

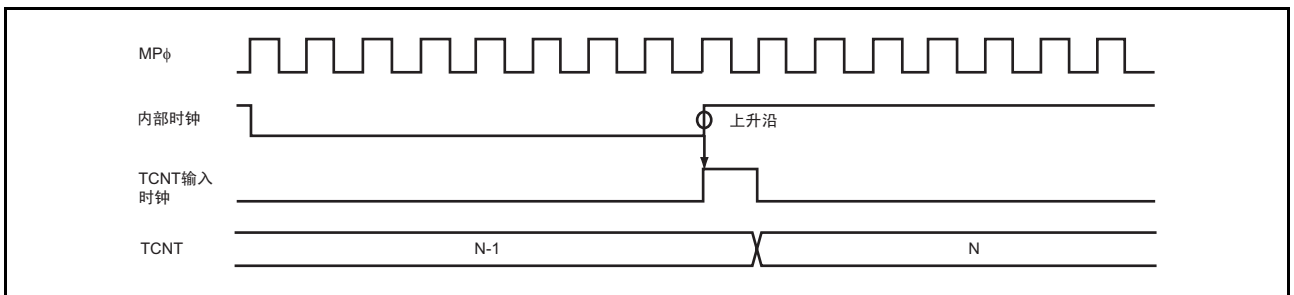


图 11.98 内部时钟运行时的计数时序（通道 5）

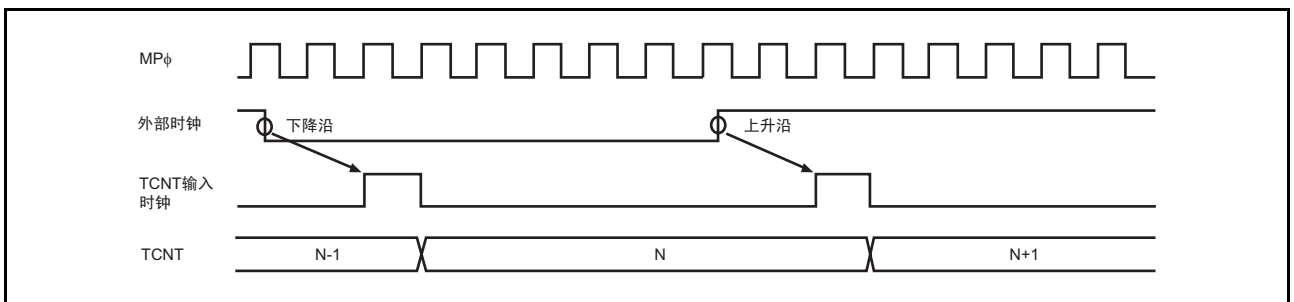


图 11.99 外部时钟运行时的计数时序（通道 0 ~ 4）

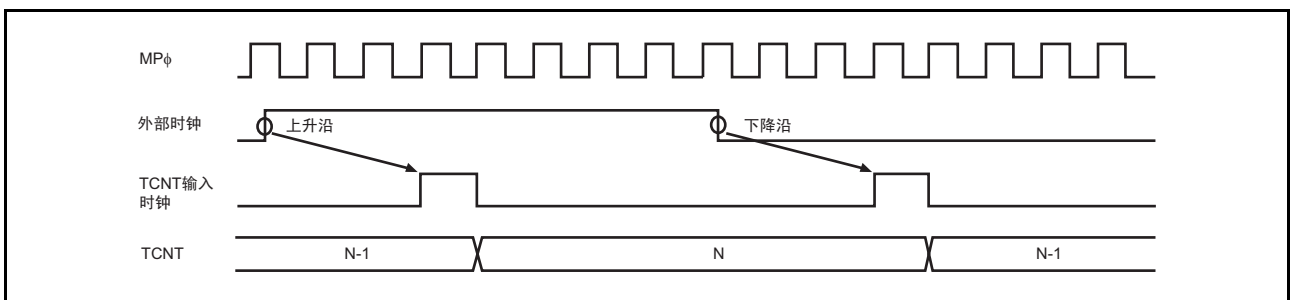


图 11.100 外部时钟运行时的计数时序（相位计数模式）

(2) 输出比较的输出时序

在 TCNT 和 TGR 一致的最后状态（更新 TCNT 一致后的计数值时）产生比较匹配信号。在产生比较匹配信号时，将 TIOR 设定的输出值输出到输出比较的输出引脚（TIOC 引脚）。在 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。

输出比较的输出时序（正常模式和 PWM 模式）如图 11.101、输出比较的输出时序（互补 PWM 模式和复位同期 PWM 模式）如图 11.102 所示。

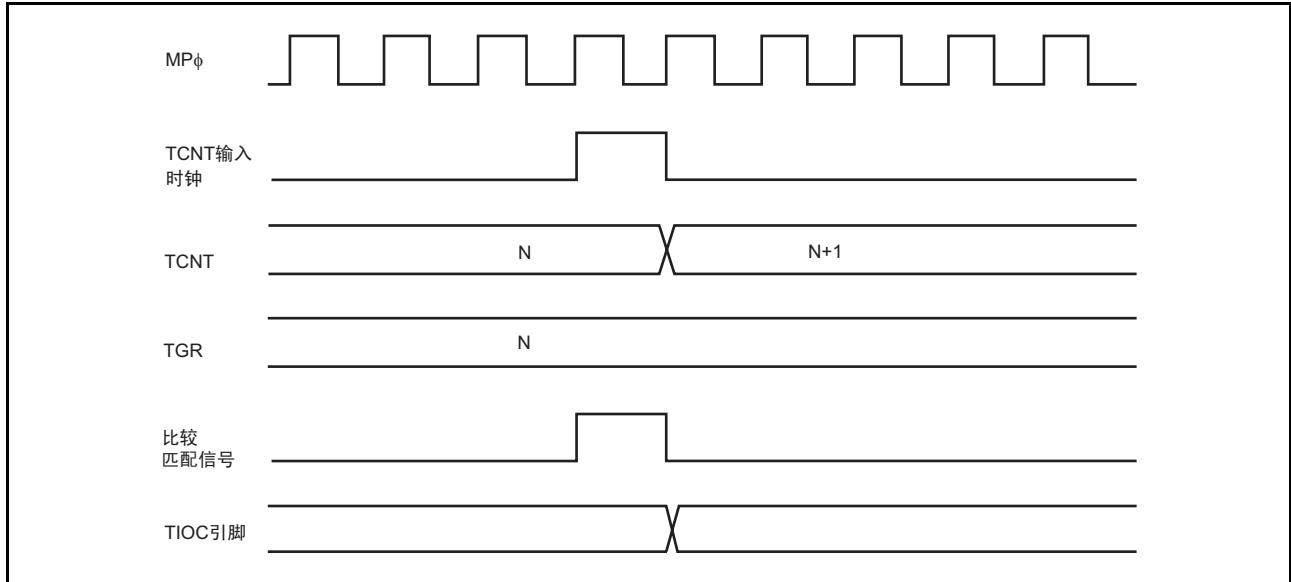


图 11.101 输出比较的输出时序（正常模式和 PWM 模式）

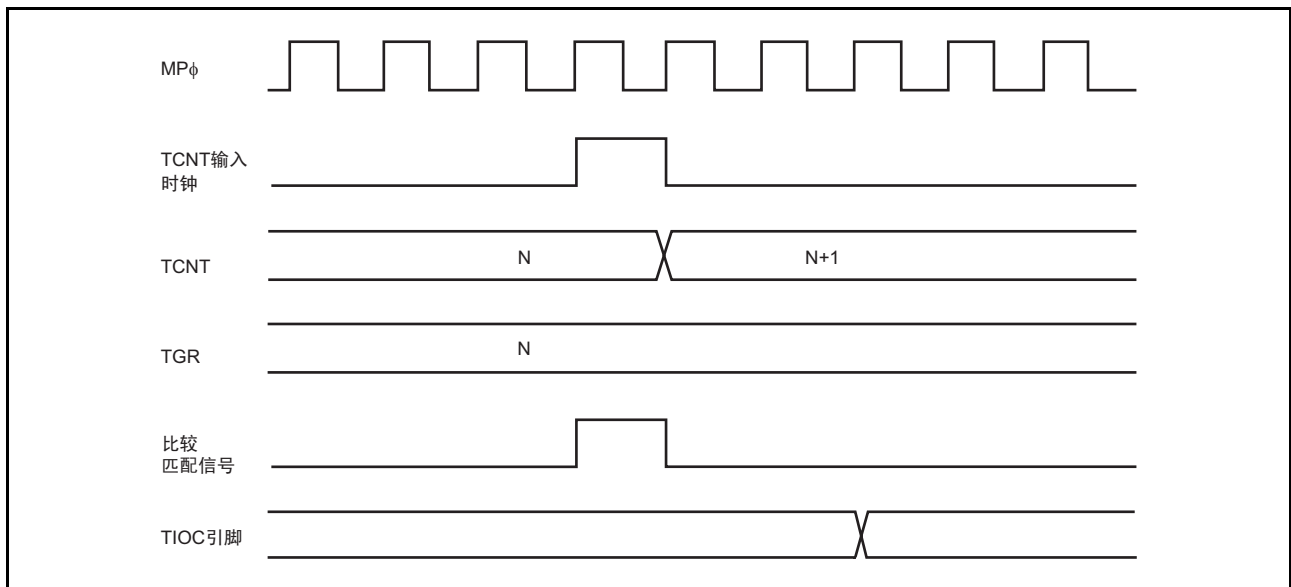


图 11.102 输出比较的输出时序（互补 PWM 模式和复位同期 PWM 模式）

(3) 输入捕捉的输入信号时序

输入捕捉的输入信号时序如图 11.103 所示。

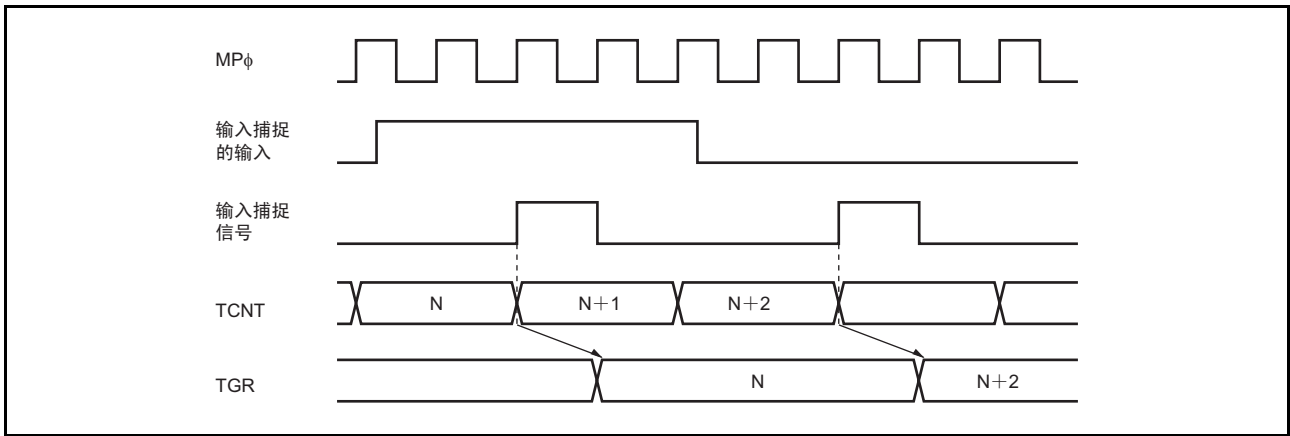


图 11.103 输入捕捉的输入信号时序

(4) 通过比较匹配 / 输入捕捉进行计数器清除的时序

指定通过比较匹配进行计数器清除时的时序如图 11.104 和图 11.105 所示。

指定通过输入捕捉进行计数器清除时的时序如图 11.106 所示。

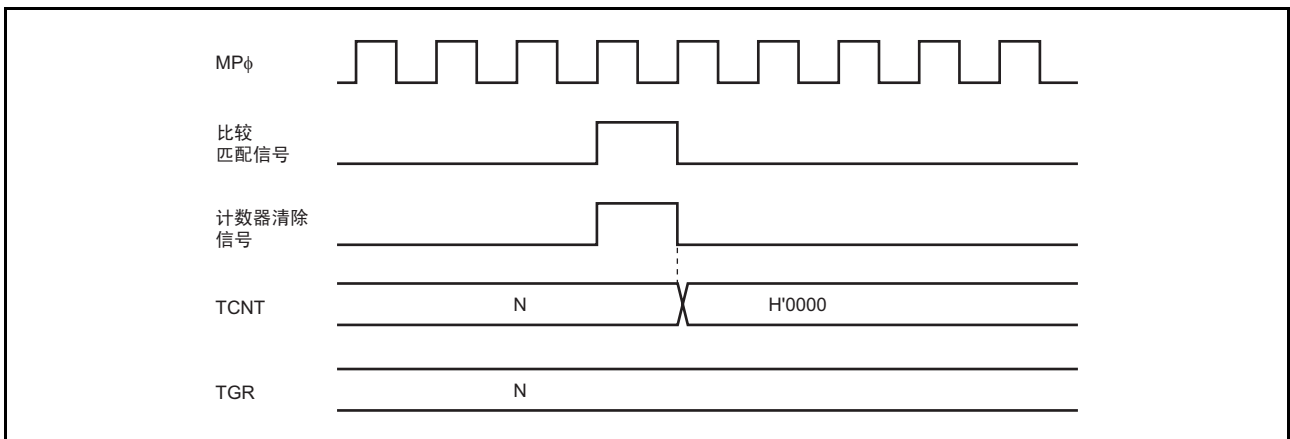


图 11.104 计数器清除的时序 (比较匹配) (通道 0 ~ 4)

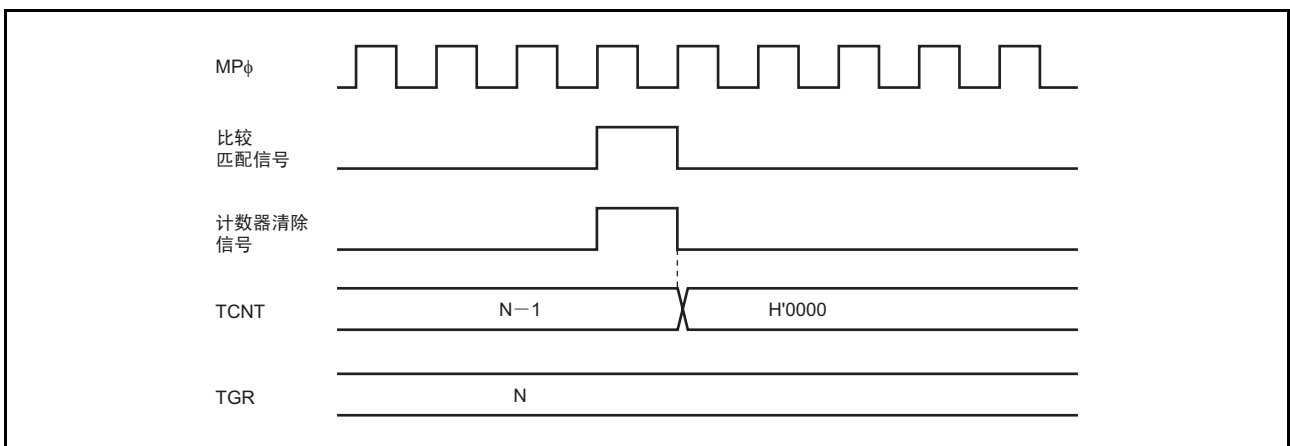


图 11.105 计数器清除的时序 (比较匹配) (通道 5)

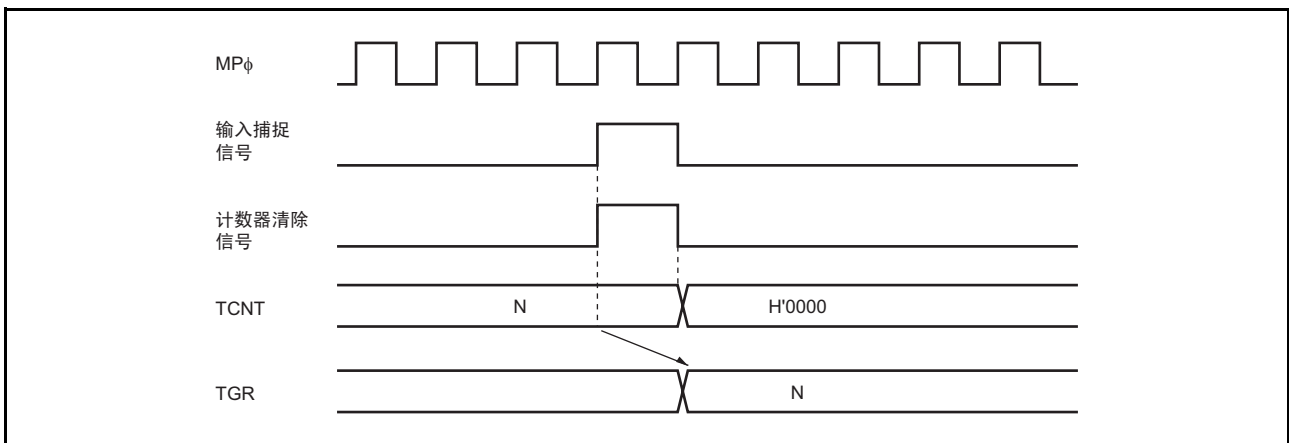


图 11.106 计数器清除的时序 (输入捕捉)

(5) 缓冲运行的时序

缓冲运行的时序如图 11.107 ~ 图 11.109 所示。

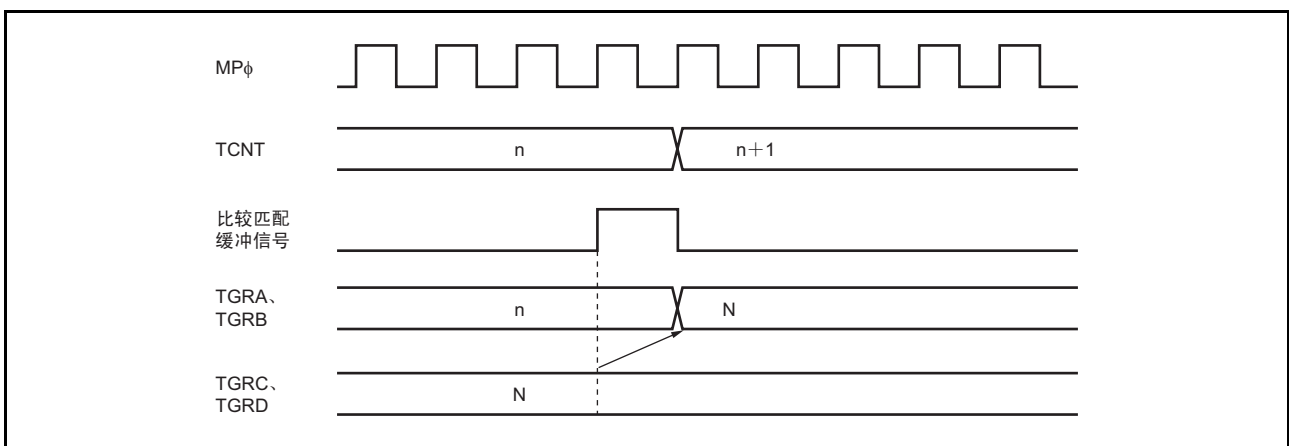


图 11.107 缓冲运行的时序 (比较匹配)

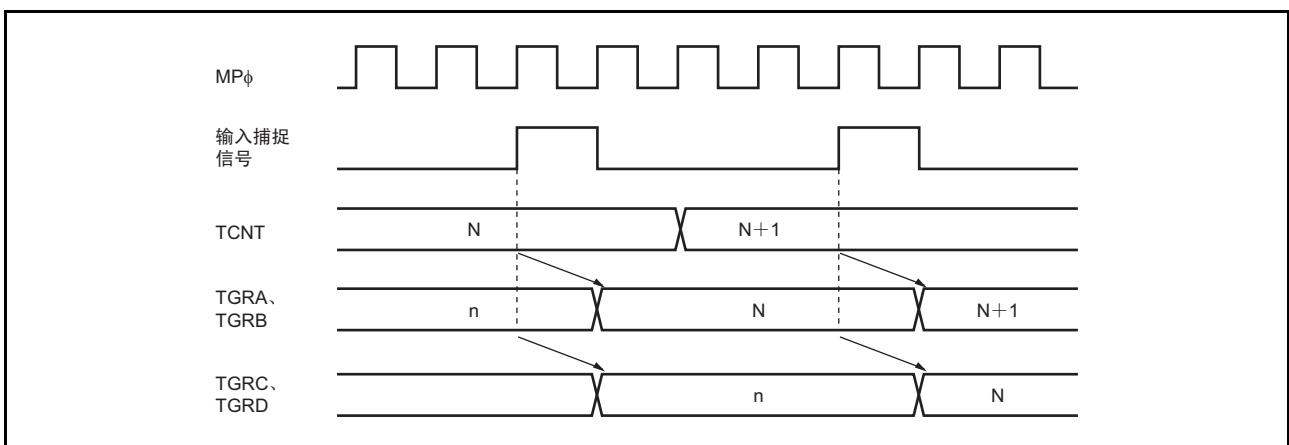


图 11.108 缓冲运行的时序 (输入捕捉)

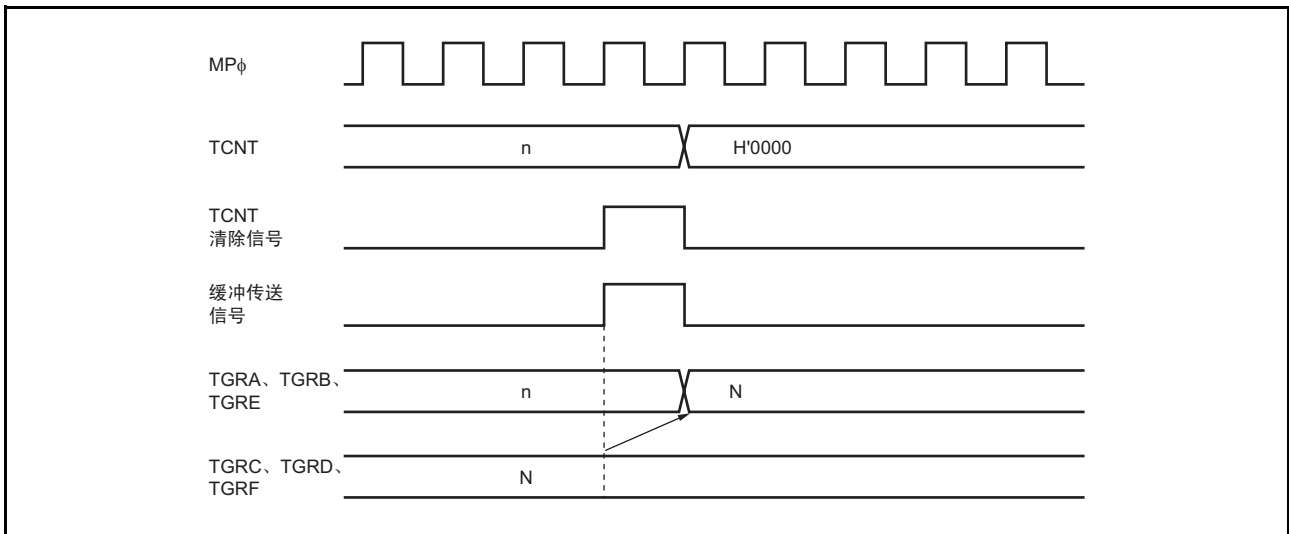


图 11.109 缓冲运行的时序 (清除 TCNT)

(6) 缓冲传送的时序 (互补 PWM 模式)

互补 PWM 模式中缓冲传送的时序如图 11.110 ~ 图 11.112 所示。

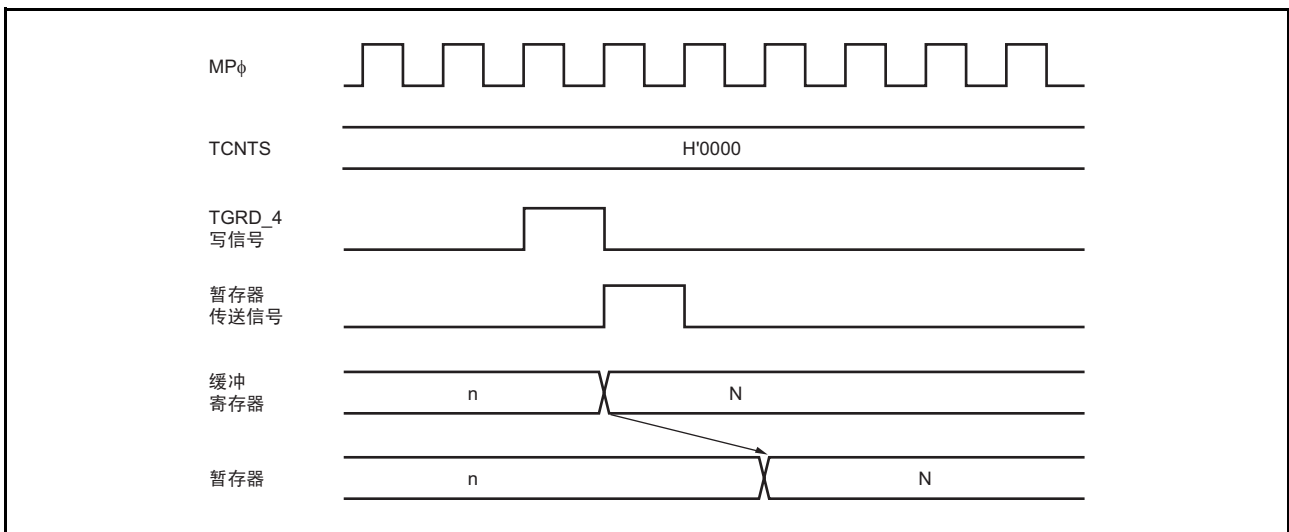


图 11.110 缓冲寄存器到暂存器的传送时序 (TCNTS 停止中)

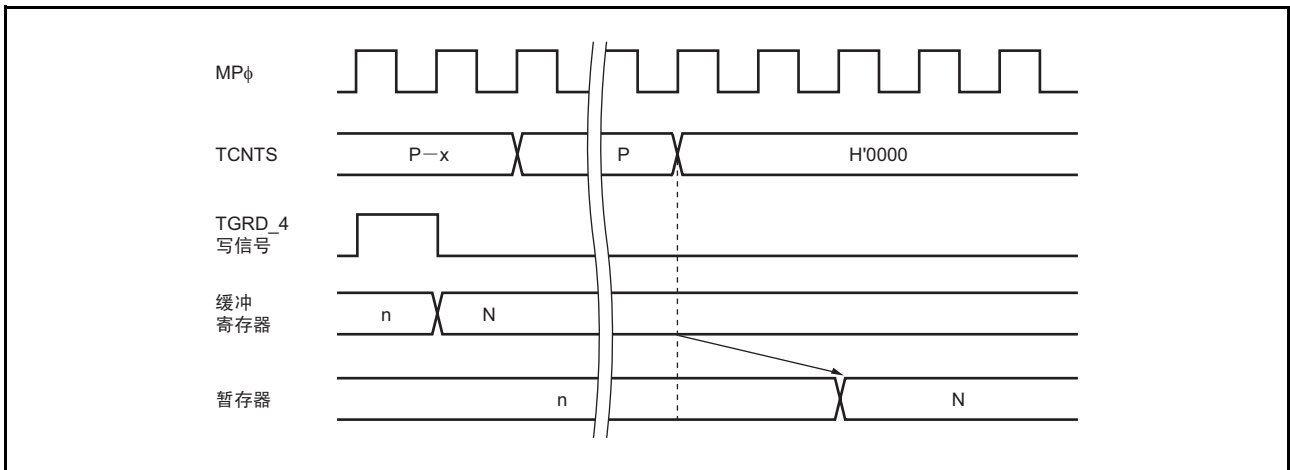


图 11.111 缓冲寄存器到暂存器的传送时序 (TCNTS 运行中)

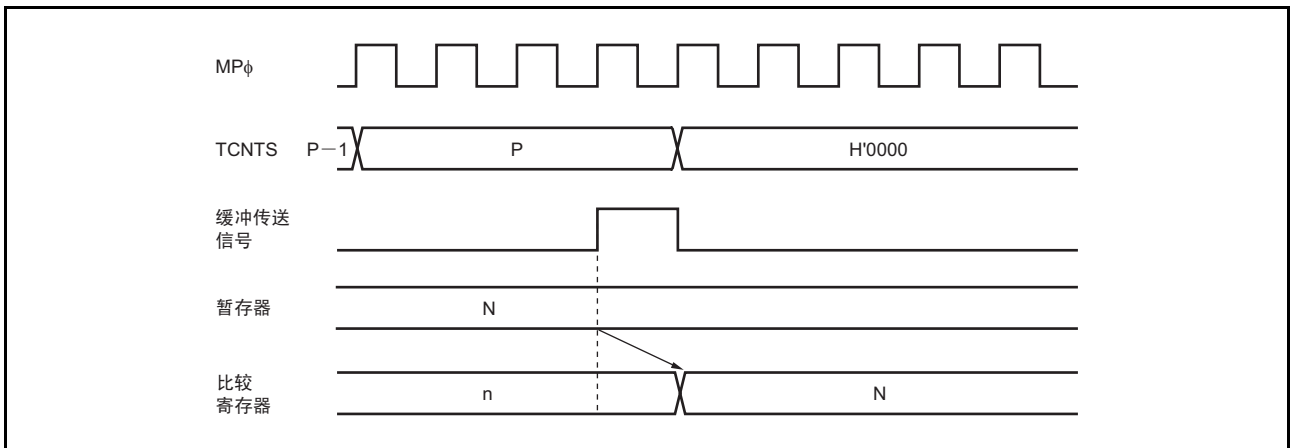


图 11.112 暂存器到比较寄存器的传送时序

## 11.6.2 中断信号的时序

## (1) 比较匹配时的 TGF 标志的置位时序

由比较匹配产生的 TSR 的 TGF 标志置位时序和 TGI 中断请求信号的时序如图 11.113 和图 11.114 所示。

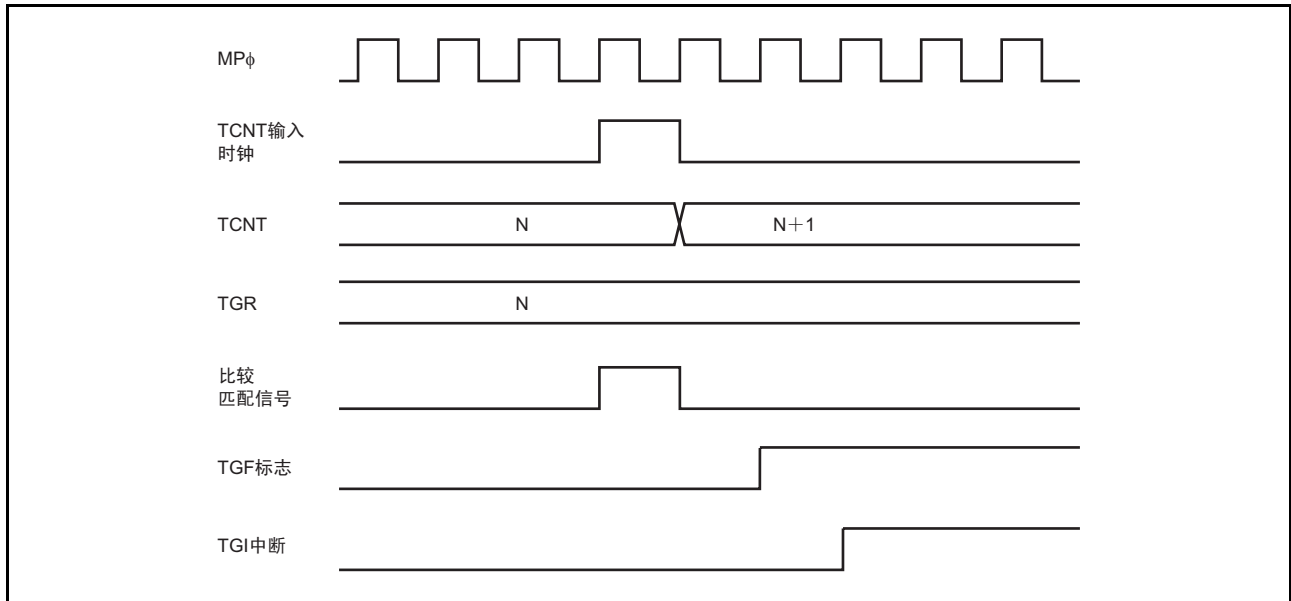


图 11.113 TGI 中断时序 (比较匹配) (通道 0 ~ 4)

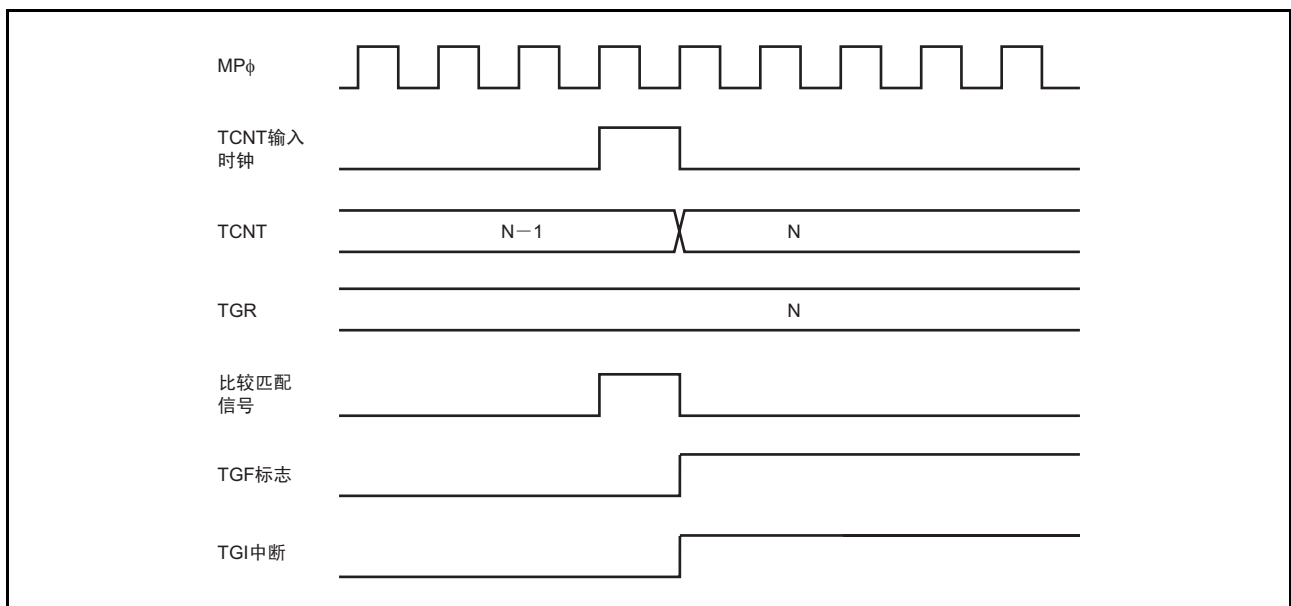


图 11.114 TGI 中断时序 (比较匹配) (通道 5)



(2) 输入捕捉时的 TGF 标志的置位时序

由输入捕捉产生 TSR 的 TGF 标志的置位时序和 TGI 中断请求信号的时序如图 11.115 和图 11.116 所示。

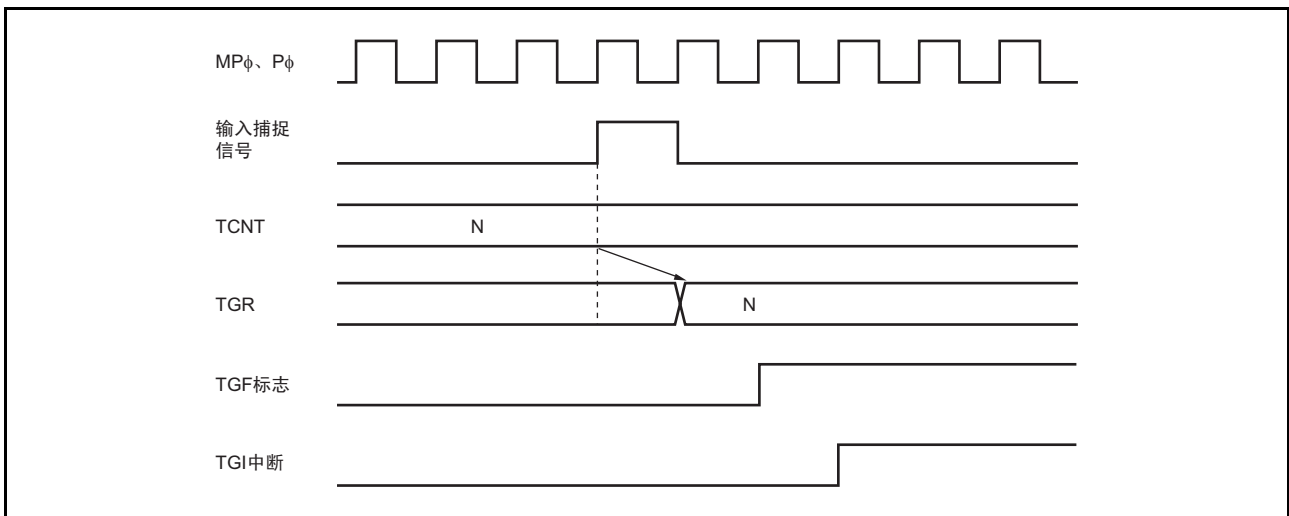


图 11.115 TGI 中断时序 (输入捕捉) (通道 0 ~ 4)

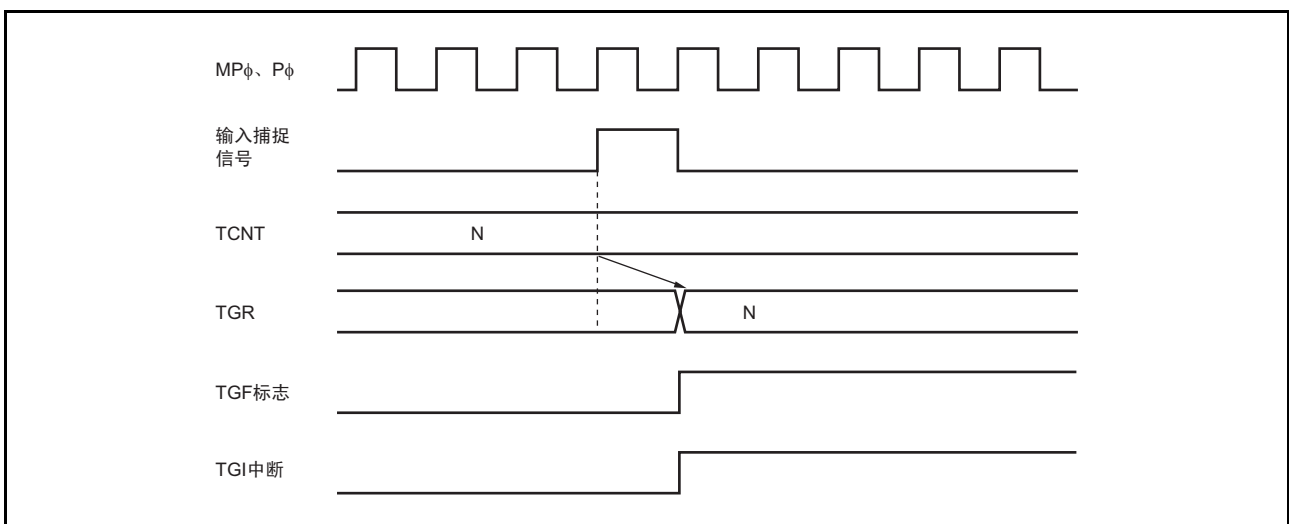


图 11.116 TGI 中断时序 (输入捕捉) (通道 5)

(3) TCFV 标志 /TCFU 标志的置位时序

由上溢产生的 TSR 的 TCFV 标志的置位时序和 TCIV 中断请求信号的时序如图 11.117 所示。  
 由下溢产生的 TSR 的 TCFU 标志的置位时序和 TCIU 中断请求信号的时序如图 11.118 所示。

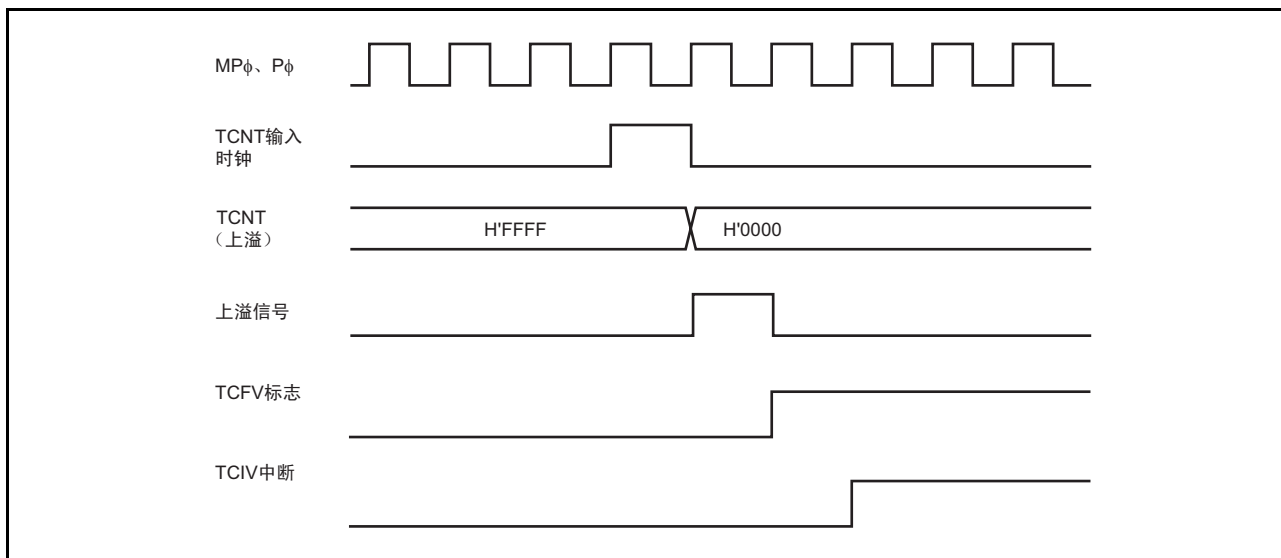


图 11.117 TCIV 中断的置位时序

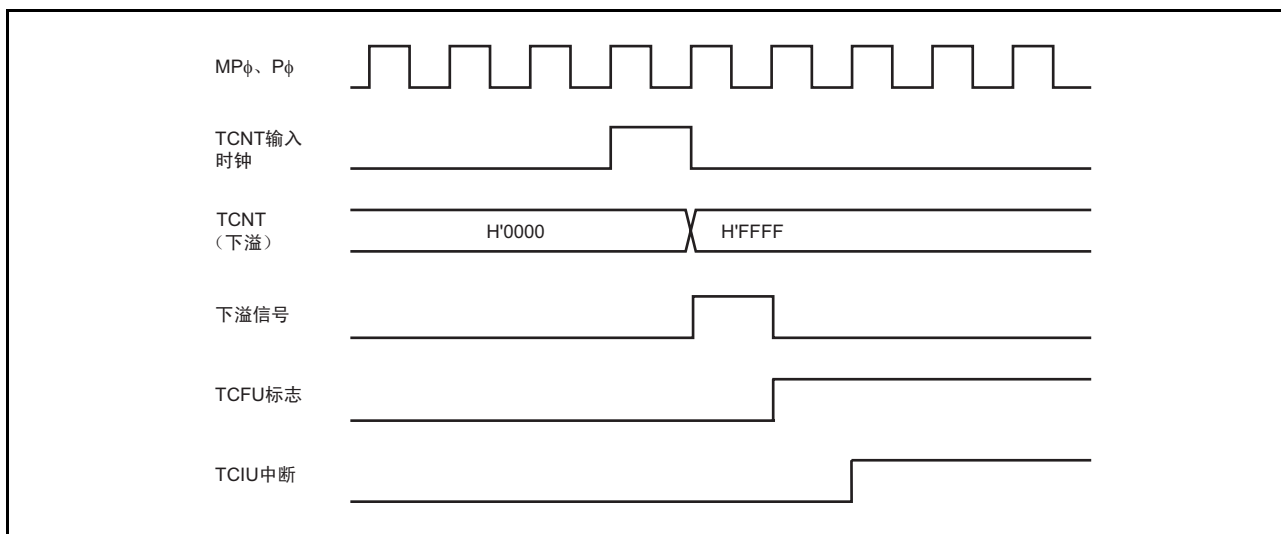


图 11.118 TCIU 中断的置位时序

(4) 状态标志的清除时序

如果 CPU 在读 1 的状态后写 0，就清除状态标志。在启动 DTC/DMAC 时，能自动清除此状态标志。通过 CPU 清除状态标志的时序如图 11.119 和图 11.120、通过 DTC/DMAC 清除状态标志的时序如图 11.121 ~ 图 11.123 所示。

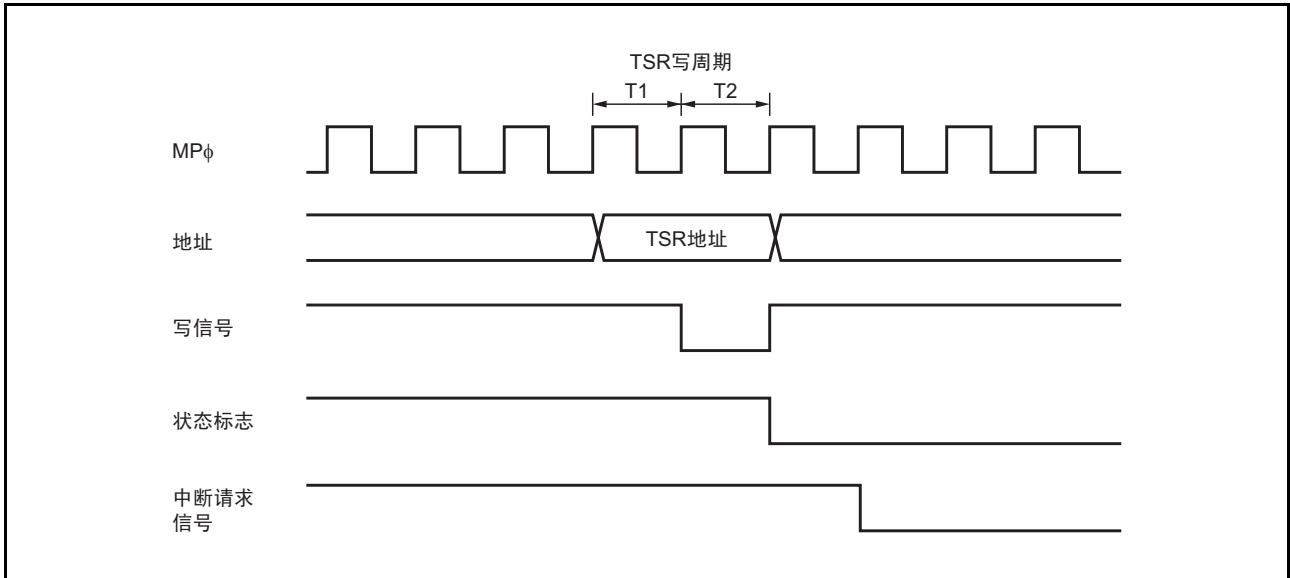


图 11.119 通过 CPU 清除状态标志的时序 (通道 0 ~ 4)

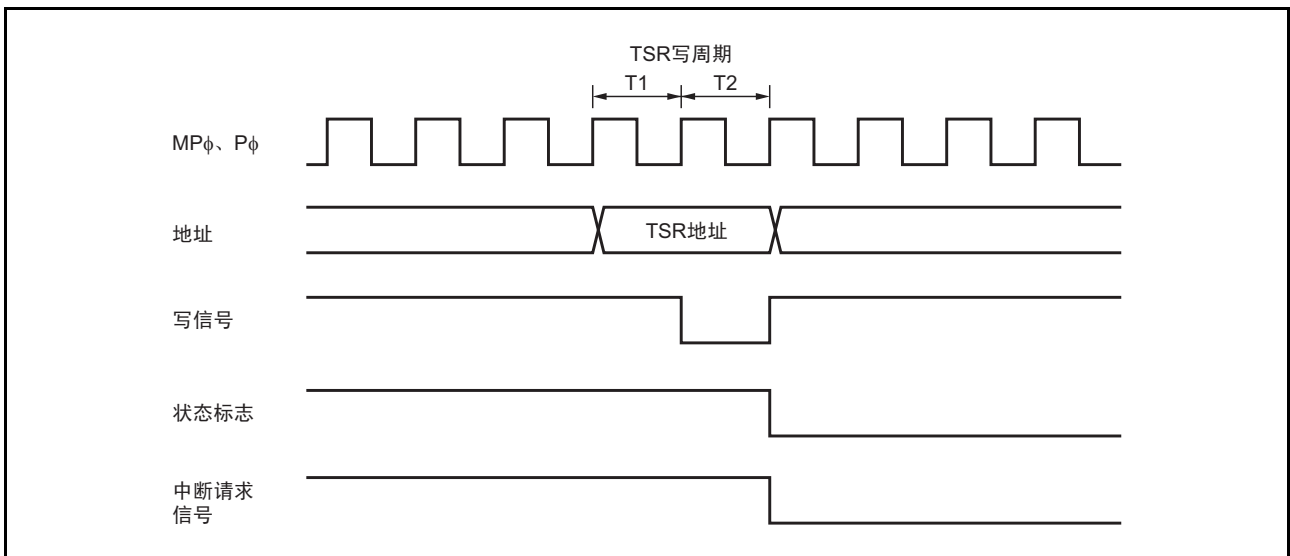


图 11.120 通过 CPU 清除状态标志的时序 (通道 5)

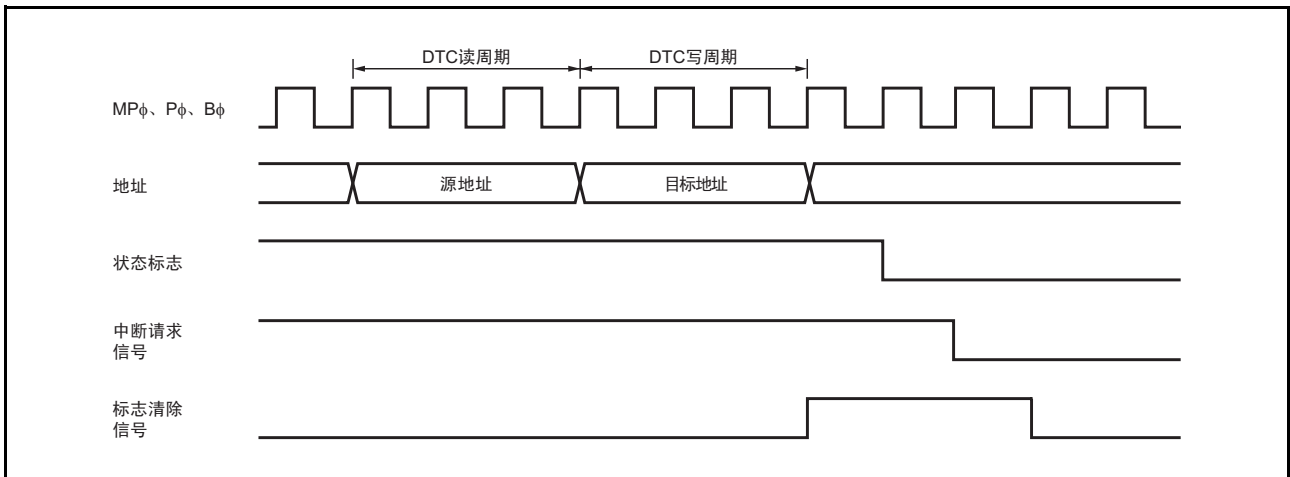


图 11.121 通过 DTC 的启动清除状态标志的时序 (通道 0 ~ 4)

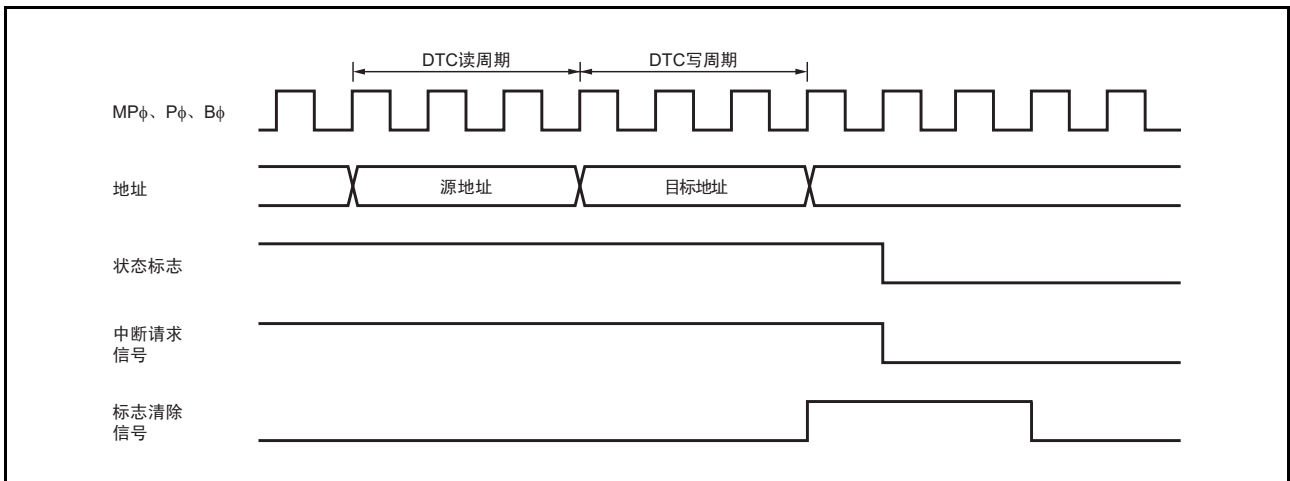


图 11.122 通过 DTC 的启动清除状态标志的时序 (通道 5)

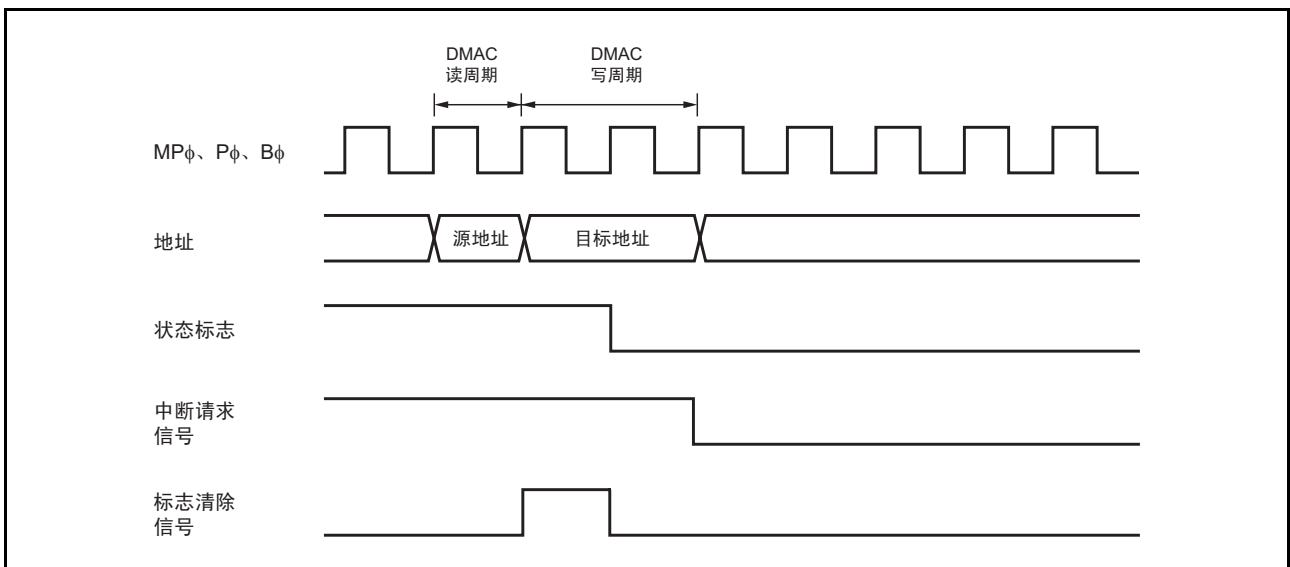


图 11.123 通过 DMAC 的启动清除状态标志的时序

## 11.7 使用注意事项

### 11.7.1 模块待机模式的设定

MTU2 能通过待机控制寄存器允许或者禁止本模块的运行，初始值为停止 MTU2 的运行。另外，通过解除模块待机模式，允许寄存器的存取。详细内容请参照“第 26 章 低功耗模式”。

### 11.7.2 输入时钟的限制事项

在单边沿的情况下，输入时钟的脉宽需要在 1.5 个状态以上；在双边沿的情况下，输入时钟的脉宽需要在 2.5 个状态以上。必须注意：如果小于上述脉宽就不能正常运行。

在相位计数模式的情况下，2 个输入时钟的相位差和重叠都需要在 1.5 个状态以上，并且脉宽需要在 2.5 个状态以上。相位计数模式的输入时钟条件如图 11.124 所示。

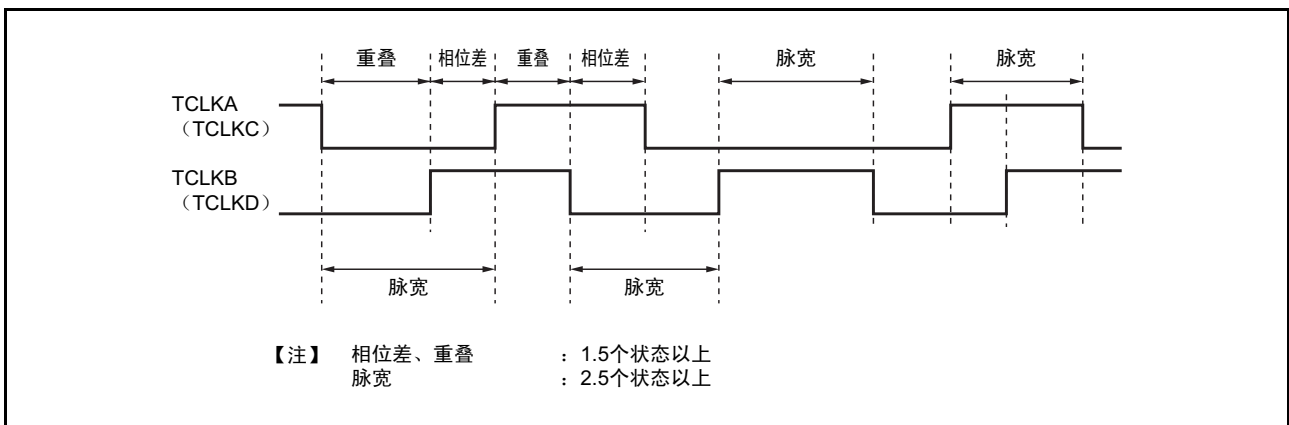


图 11.124 相位计数模式中的相位差、重叠和脉宽

### 11.7.3 周期设定的注意事项

如果设定为通过比较匹配清除计数器，就在 TCNT 和 TGR 的值一致的最后状态（更新 TCNT 一致后的计数值时）清除 TCNT。因此，实际的计数器频率为以下的表达式：

- 通道 0 ~ 4

$$f = \frac{MP\phi}{(N+1)}$$

- 通道 5

$$f = \frac{MP\phi}{N}$$

- f : 计数器频率  
MP $\phi$  : MTU2 的时钟工作频率  
N : TGR 的设定值

### 11.7.4 TCNT 的写操作和清除的竞争

如果在 TCNT 写周期中的 T2 状态产生计数器清除信号，就不写 TCNT 而优先清除 TCNT。此时序如图 11.125 所示。

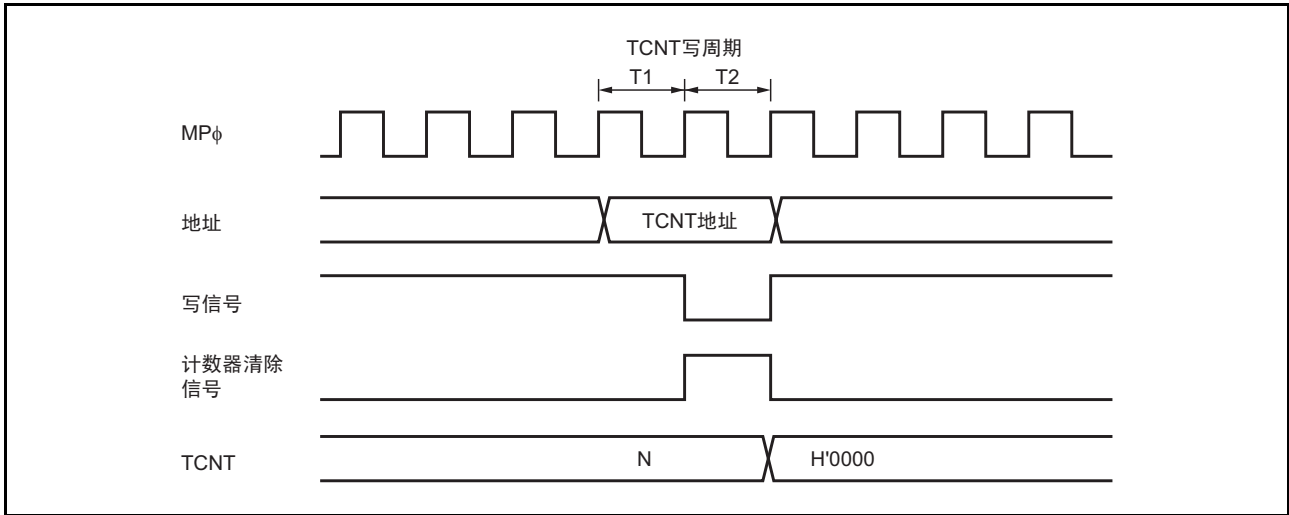


图 11.125 TCNT 的写操作和清除的竞争

### 11.7.5 TCNT 的写操作和递增计数的竞争

即使在 TCNT 写周期中的 T2 状态发生递增计数信号，也不进行递增计数而优先写 TCNT。此时序如图 11.126 所示。

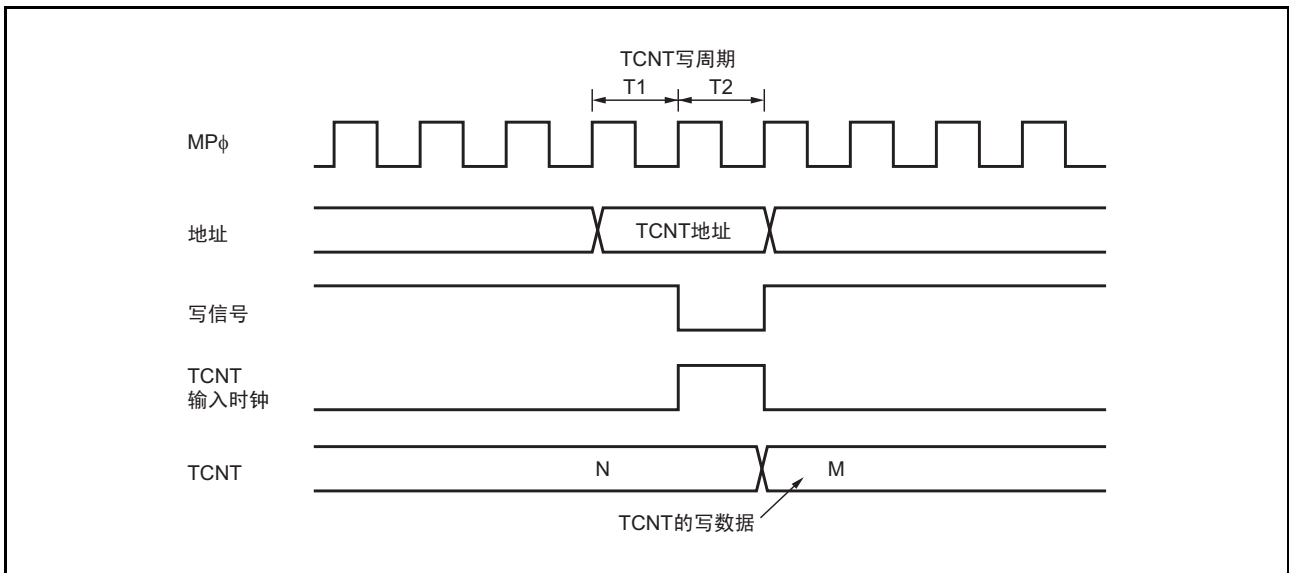


图 11.126 TCNT 的写操作和递增计数的竞争

### 11.7.6 TGR 的写操作和比较匹配的竞争

如果在 TGR 写周期中的 T2 状态产生比较匹配信号，就写 TGR 并且产生比较匹配信号。此时序如图 11.127 所示。

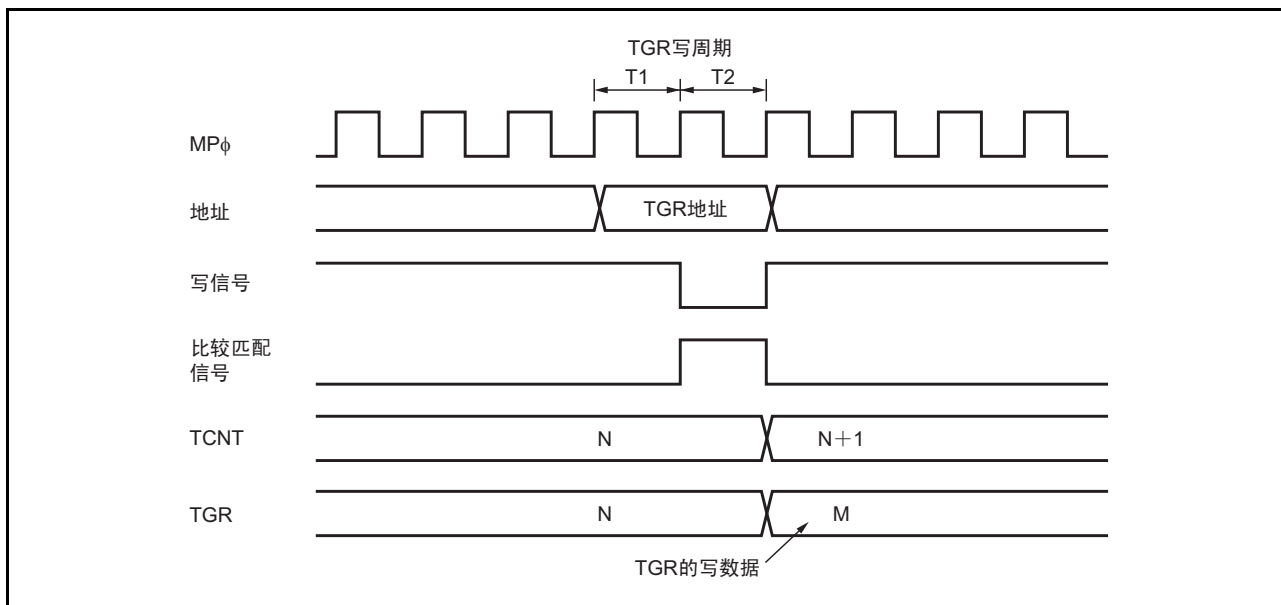


图 11.127 TGR 的写操作和比较匹配的竞争

### 11.7.7 缓冲寄存器的写操作和比较匹配的竞争

如果在 TGR 写周期中的 T2 状态产生比较匹配信号，由缓冲运行传送到 TGR 的数据就为写操作前的数据。此时序如图 11.128 所示。

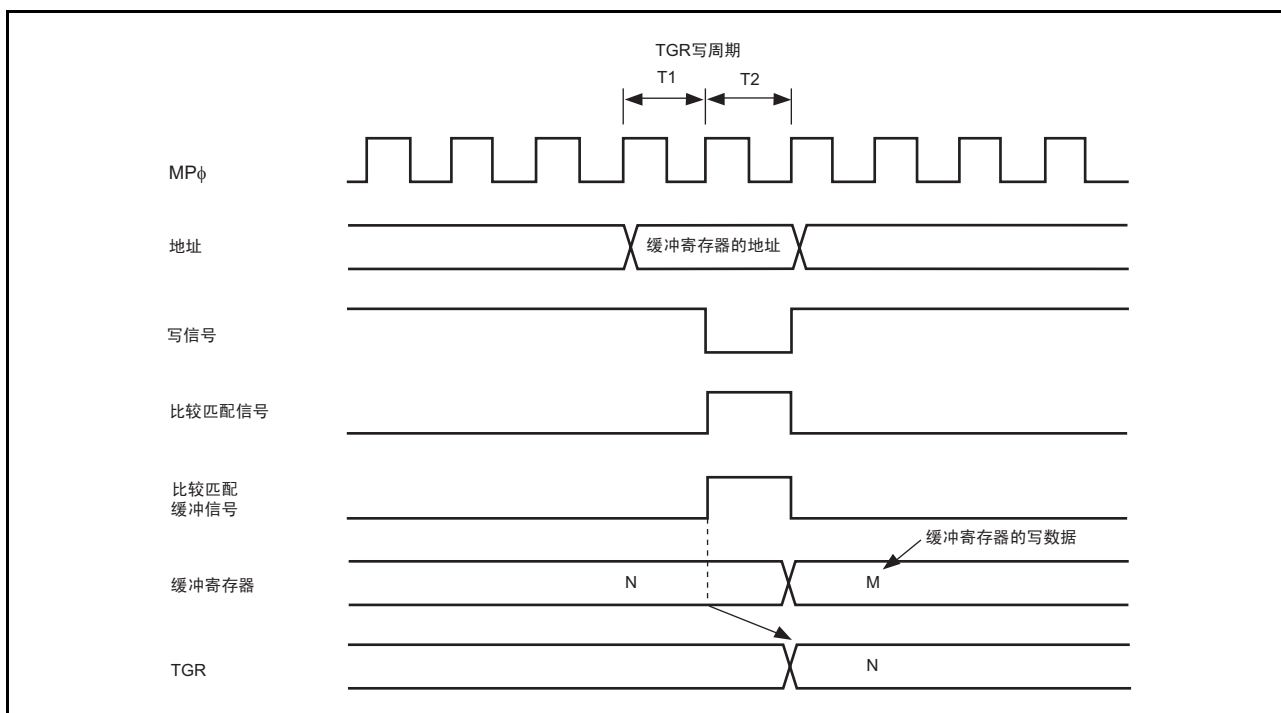


图 11.128 缓冲寄存器的写操作和比较匹配的竞争

### 11.7.8 缓冲寄存器的写操作和 TCNT 清除的竞争

通过缓冲传送模式寄存器 (TBTM) 将缓冲传送时序设定为清除 TCNT 时, 如果在 TGR 写周期中的 T2 状态产生 TCNT 的清除信号, 由缓冲运行传送到 TGR 的数据就为写操作前的数据。

此时序如图 11.129 所示。

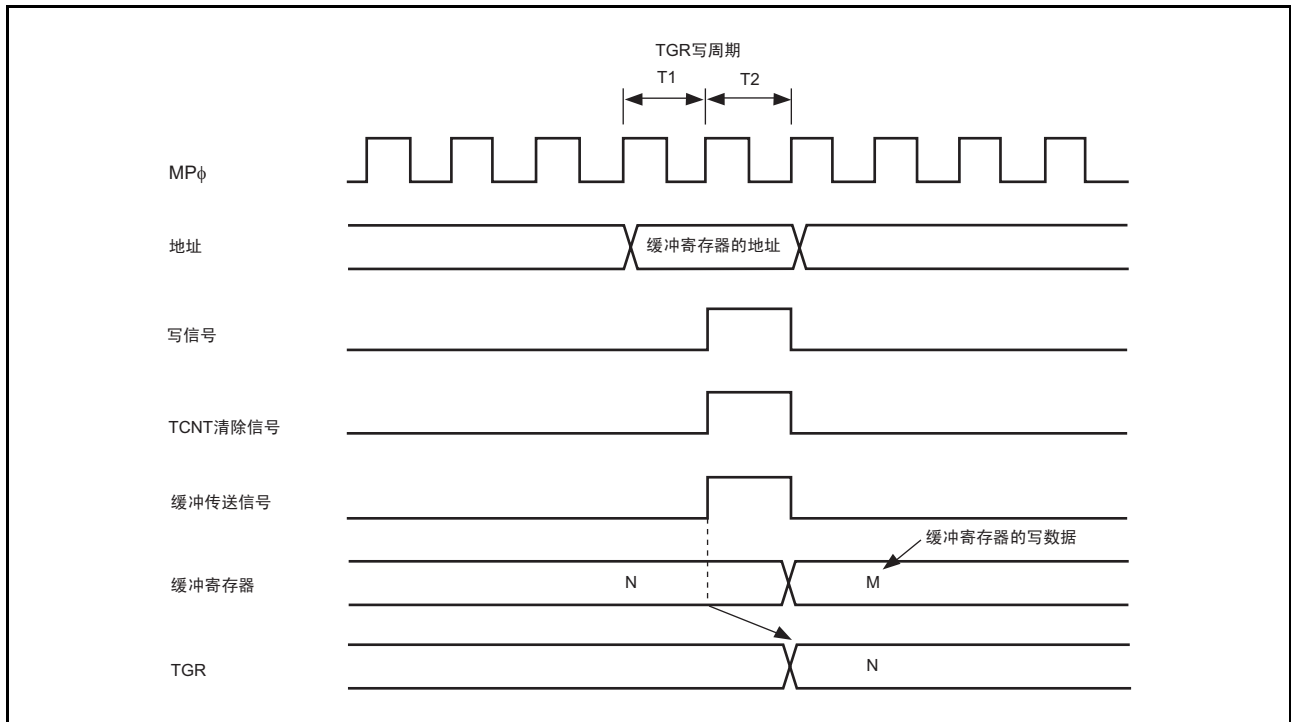


图 11.129 缓冲寄存器的写操作和 TCNT 清除的竞争



### 11.7.9 TGR 的读操作和输入捕捉的竞争

如果在 TGR 读周期中的 T1 状态产生输入捕捉信号，读出的数据在通道 0 ~ 4 中为输入捕捉传送前的数据，而在通道 5 中为输入捕捉传送后的数据。

此时序如图 11.130 和图 11.131 所示。

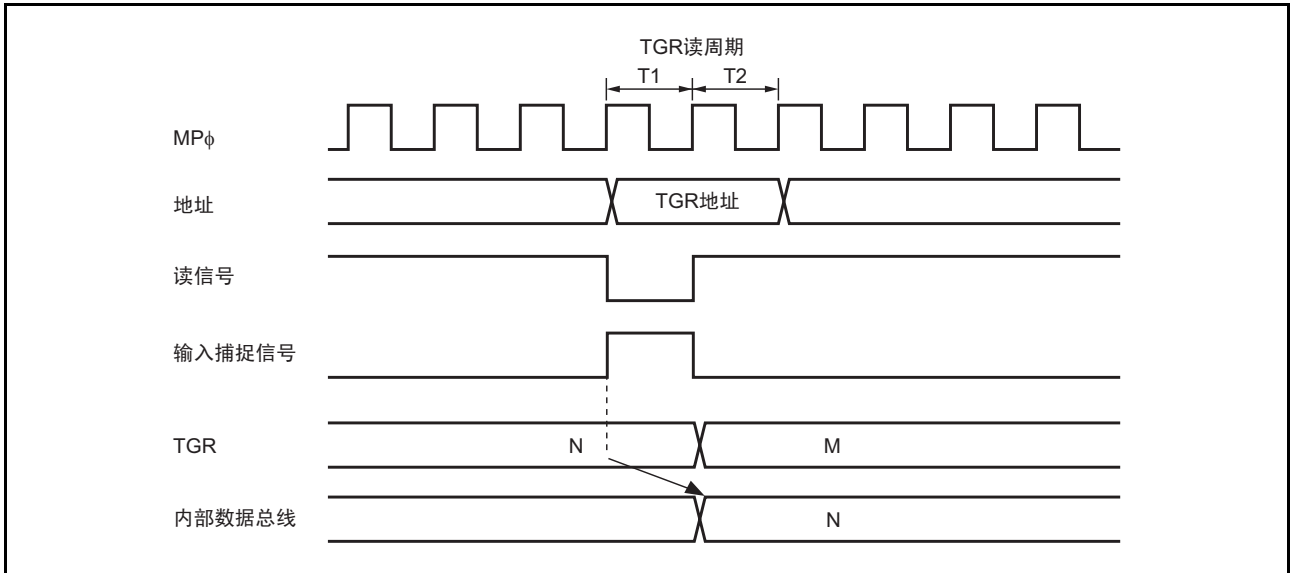


图 11.130 TGR 的读操作和输入捕捉的竞争 (通道 0 ~ 4)

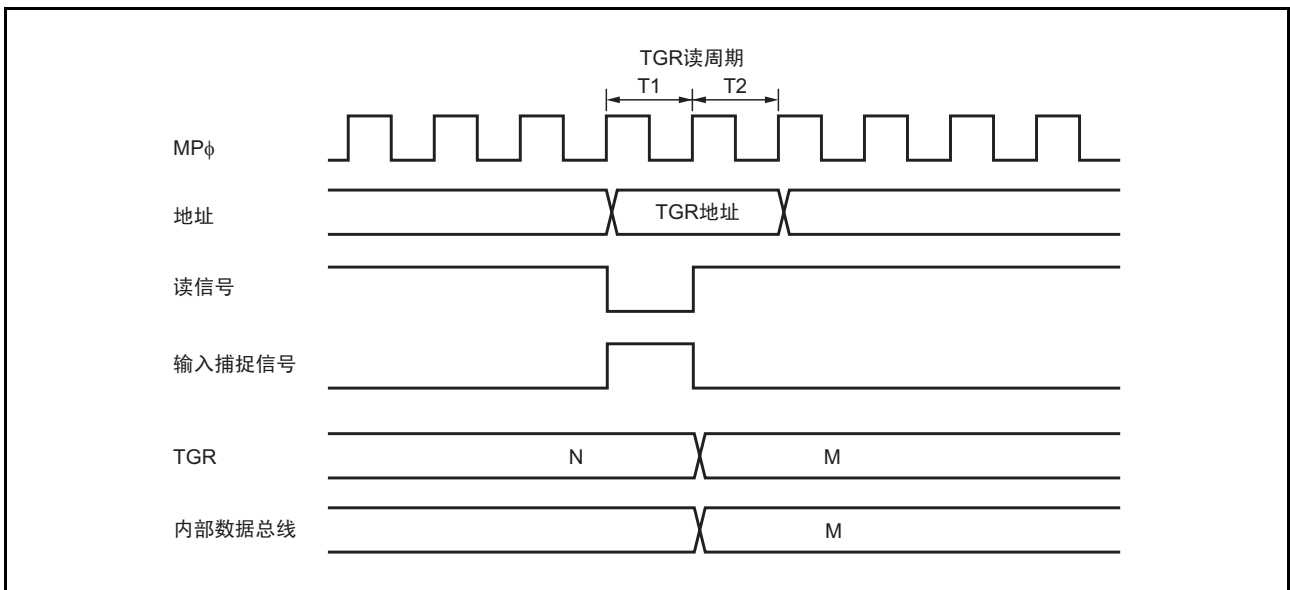


图 11.131 TGR 的读操作和输入捕捉的竞争 (通道 5)

### 11.7.10 TGR 的写操作和输入捕捉的竞争

如果在 TGR 写周期中的 T2 状态产生输入捕捉信号，在通道 0 ~ 4 中不写 TGR 而优先输入捕捉，但是在通道 5 中写 TGR 并且产生输入捕捉信号。

此时序如图 11.132 和图 11.133 所示。

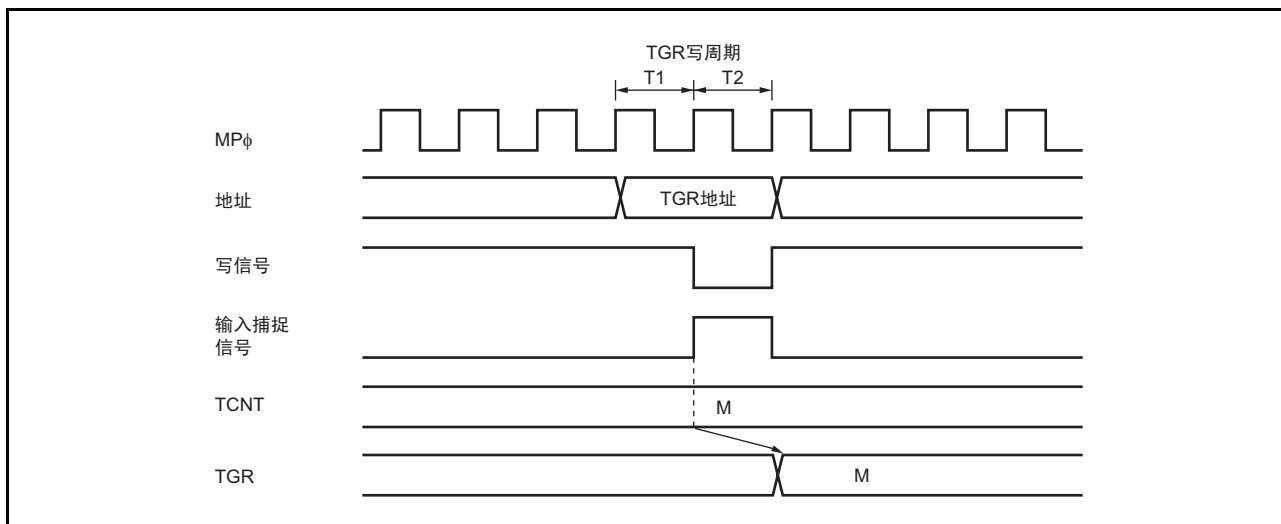


图 11.132 TGR 的写操作和输入捕捉的竞争 (通道 0 ~ 4)

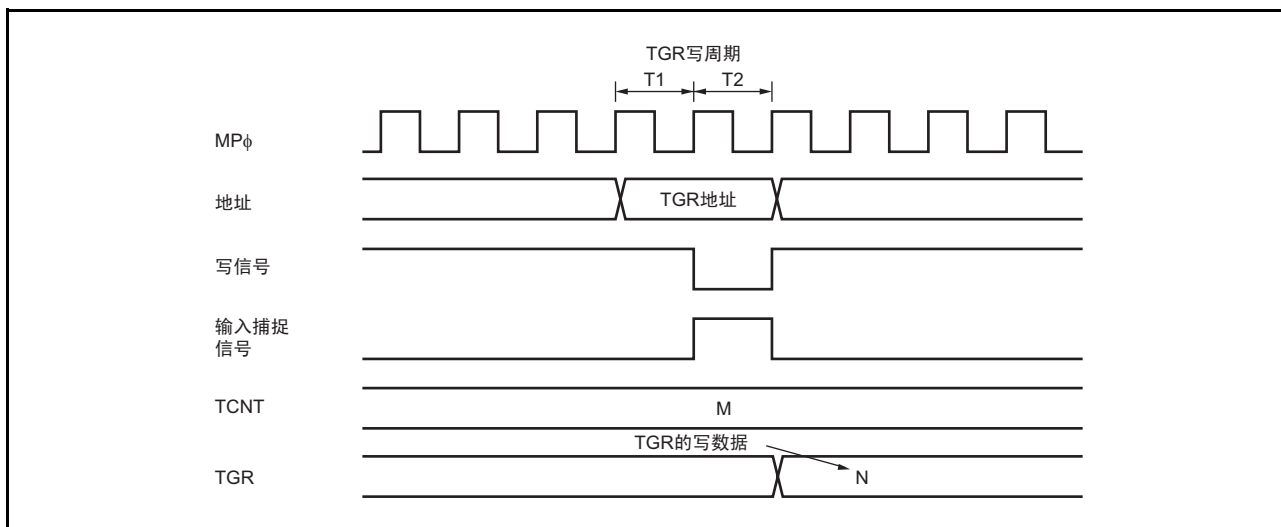


图 11.133 TGR 的写操作和输入捕捉的竞争 (通道 5)

## 11.7.11 缓冲寄存器的写操作和输入捕捉的竞争

如果在缓冲器写周期中的 T2 状态产生输入捕捉信号，就不写缓冲寄存器而优先缓冲运行。  
此时序如图 11.134 所示。

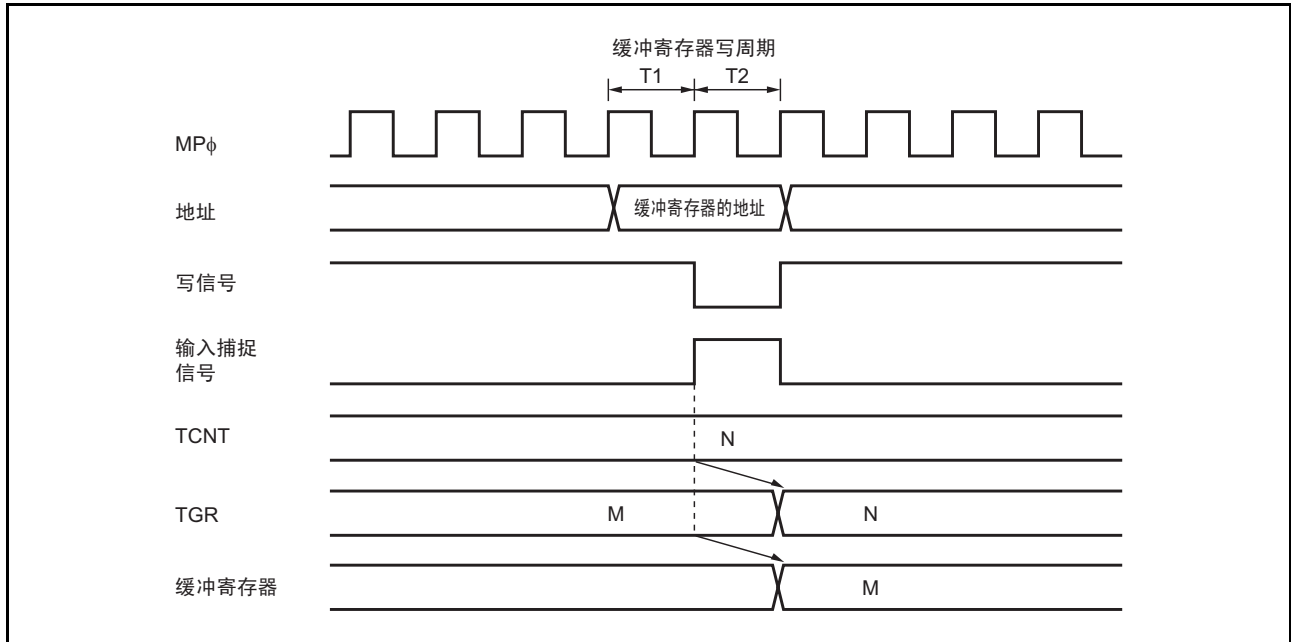


图 11.134 缓冲寄存器的写操作和输入捕捉的竞争

### 11.7.12 级联中 TCNT\_2 的写操作和上溢 / 下溢的竞争

将定时器的计数器 (TCNT\_1 和 TCNT\_2) 进行级联, 如果 TCNT\_1 在进行计数的瞬间 (TCNT\_2 发生上溢 / 下溢的瞬间) 和 TCNT\_2 写周期中的 T2 状态发生竞争, 就写 TCNT\_2 来禁止 TCNT\_1 的计数信号。此时, TGRA\_1 用作比较匹配寄存器, 当 TGRA\_1 和 TCNT\_1 的值一致时, 就产生比较匹配信号。

如果选择 TCNT\_1 计数时钟为通道 0 的输入捕捉源, TGRA\_0 ~ D\_0 就进行输入捕捉; 如果选择 TGRC\_0 的比较匹配 / 输入捕捉为 TGRB\_1 的输入捕捉源, TGRB\_1 就进行输入捕捉。

此时序如图 11.135 所示。

另外, 在级联运行中设定 TCNT 的清除时, 必须进行通道 1 和通道 2 的同步设定。

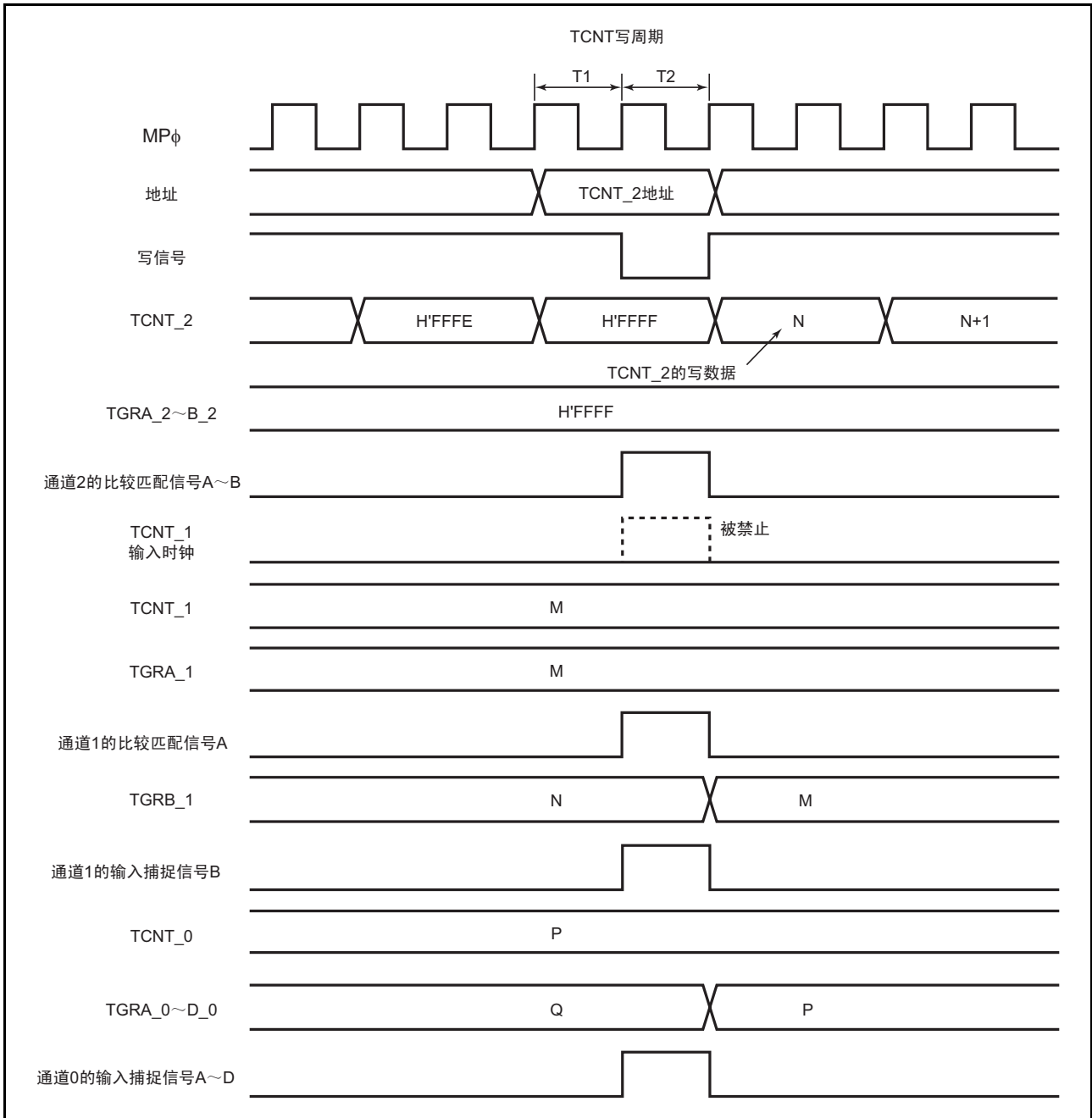


图 11.135 级联中 TCNT\_2 的写操作和上溢 / 下溢的竞争

### 11.7.13 互补 PWM 模式停止时的计数器值

在互补 PWM 模式中，如果 TCNT\_3 和 TCNT\_4 在运行中停止计数，TCNT\_3 就为定时器的空载时间寄存器 (TDDR) 的值，TCNT\_4 为 H'0000。

当重新开始互补 PWM 模式时，就自动从初始状态开始计数。

此说明如图 11.136 所示。

另外，在其他运行模式中开始计数时，必须给 TCNT\_3 和 TCNT\_4 设定计数的初始值。

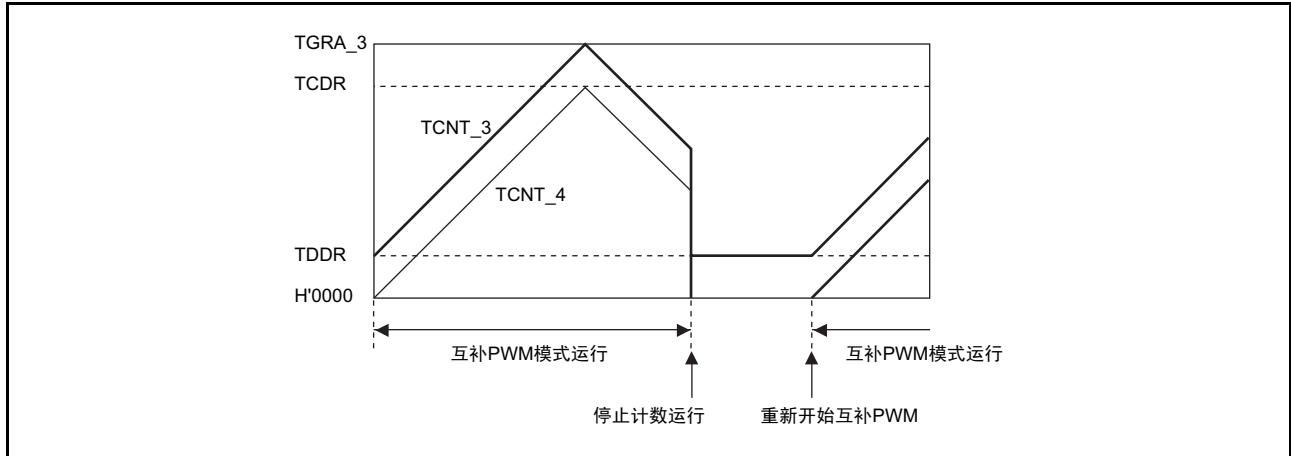


图 11.136 互补 PWM 模式停止时的计数器值

### 11.7.14 互补 PWM 模式中缓冲运行的设定

在互补 PWM 模式中，必须在缓冲运行中改写 PWM 周期设定寄存器 (TGRA\_3)、定时器的周期数据寄存器 (TCDR) 和占空比设定寄存器 (TGRB\_3、TGRA\_4、TGRB\_4)。

按照 TMDR\_3 的 BFA 和 BFB 位的设定，进行互补 PWM 模式中的通道 3 和通道 4 的缓冲运行。如果将 TMDR\_3 的 BFA 位设定为 1，TGRC\_3 就用作 TGRA\_3 的缓冲寄存器，TGRC\_4 用作 TGRA\_4 的缓冲寄存器，TCBR 用作 TCDR 的缓冲寄存器。

### 11.7.15 复位同步 PWM 模式的缓冲运行和比较匹配标志

要在复位同步 PWM 模式中设定缓冲运行时，必须将 TMDR\_4 的 BFA 和 BFB 位设定为 0。如果将 TMDR\_4 的 BFA 位设定为 1，TIOC4C 引脚就不能输出波形。

按照 TMDR\_3 的 BFA 和 BFB 位的设定，进行复位同步 PWM 模式中的通道 3 和通道 4 的缓冲运行。例如，如果将 TMDR\_3 的 BFA 位设定为 1，TGRC\_3 就用作 TGRA\_3 的缓冲寄存器，TGRC\_4 用作 TGRA\_4 的缓冲寄存器。

当 TGRC\_3 和 TGRD\_3 用作缓冲寄存器时，TSR\_3、TSR\_4 的 TGFC 位和 TGFD 位不被置位。

在将 TMDR\_3 的 BFA 和 BFB 位设定为 1 并且 TMDR\_4 的 BFA 和 BFB 位设定为 0 时，TGR\_3、TGR\_4、TIOC3 和 TIOC4 的运行例子如图 11.137 所示。

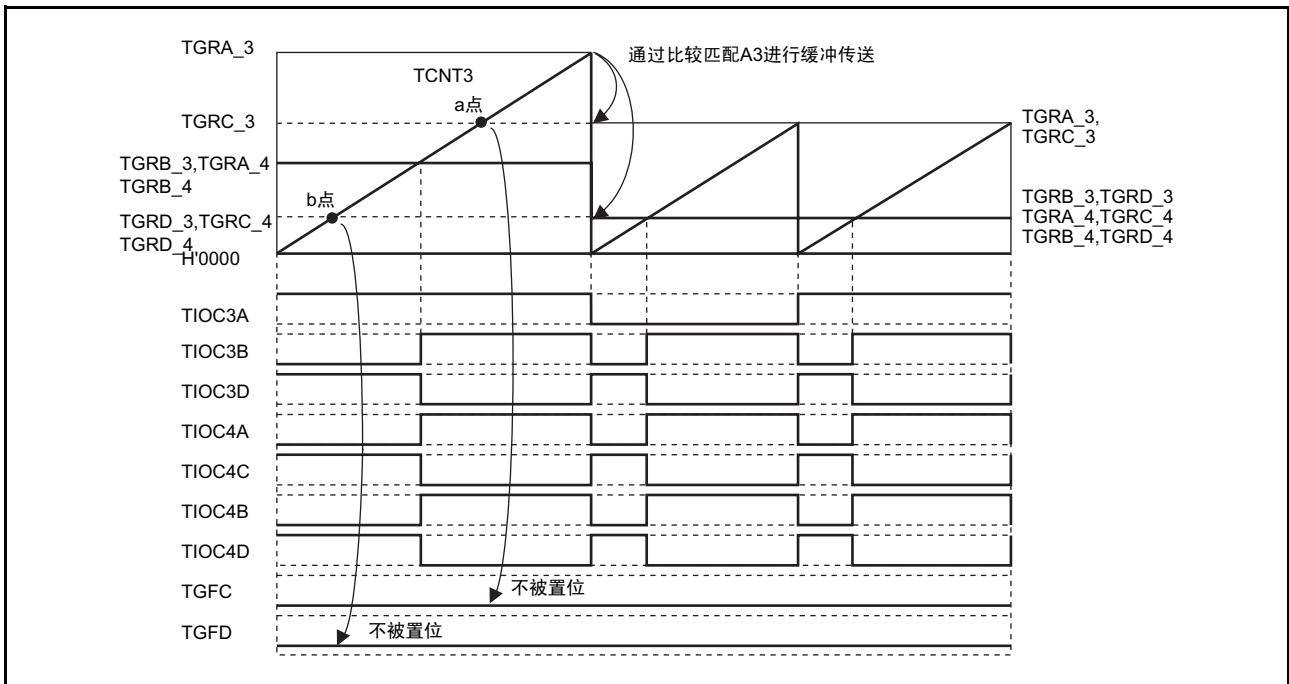


图 11.137 复位同步 PWM 模式的缓冲运行和比较匹配标志

### 11.7.16 复位同步 PWM 模式的上溢标志

如果设定复位同步 PWM 模式并将 TSTR 的 CST3 位设定为 1，就开始 TCNT\_3 和 TCNT\_4 的计数运行。此时，TCNT\_4 的计数时钟源和计数边沿服从 TCR\_3 的设定。

在复位同步 PWM 模式中，当周期寄存器 TGRA\_3 的设定值为 H'FFFF 并给计数器清除源指定 TGRA\_3 的比较匹配时，如果 TCNT\_3 和 TCNT\_4 递增计数到 H'FFFF，就产生和 TGRA\_3 的比较匹配，并且同时清除 TCNT\_3 和 TCNT\_4。此时，TSR 的上溢标志 TCFV 位不被置位。

在复位同步 PWM 模式中，当周期寄存器 TGRA\_3 的设定值为 H'FFFF 并给计数器清除源指定 TGRA\_3 的比较匹配而没有进行同步设定时，TCFV 位的运行例子如图 11.138 所示。

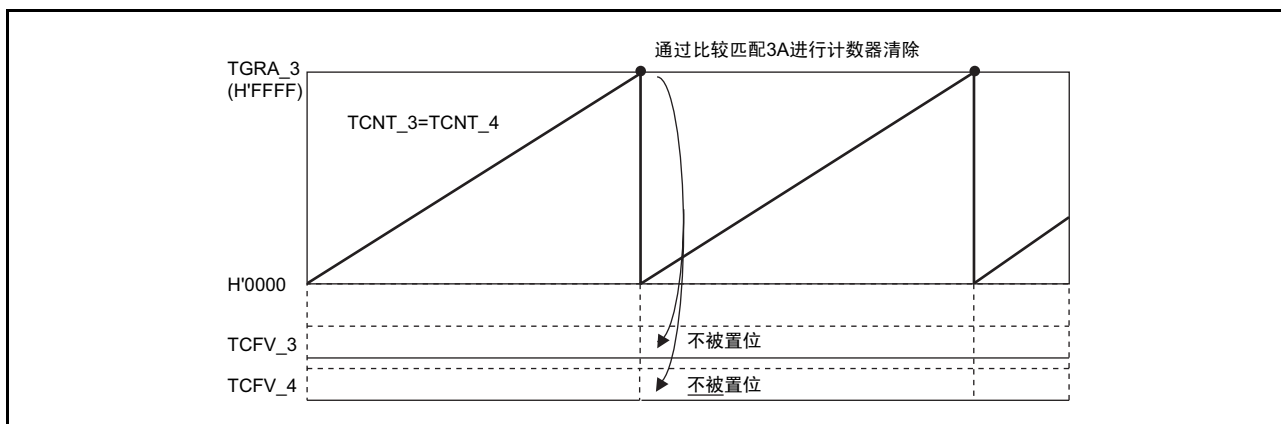


图 11.138 复位同步 PWM 模式的上溢标志

### 11.7.17 上溢 / 下溢和计数器清除的竞争

如果同时发生上溢 / 下溢和计数器清除，就不将 TSR 的 TCFV/TCFU 标志置位而优先清除 TCNT。将 TGR 的比较匹配作为清除源并给 TGR 设定 H'FFFF 时的运行时序如图 11.139 所示。

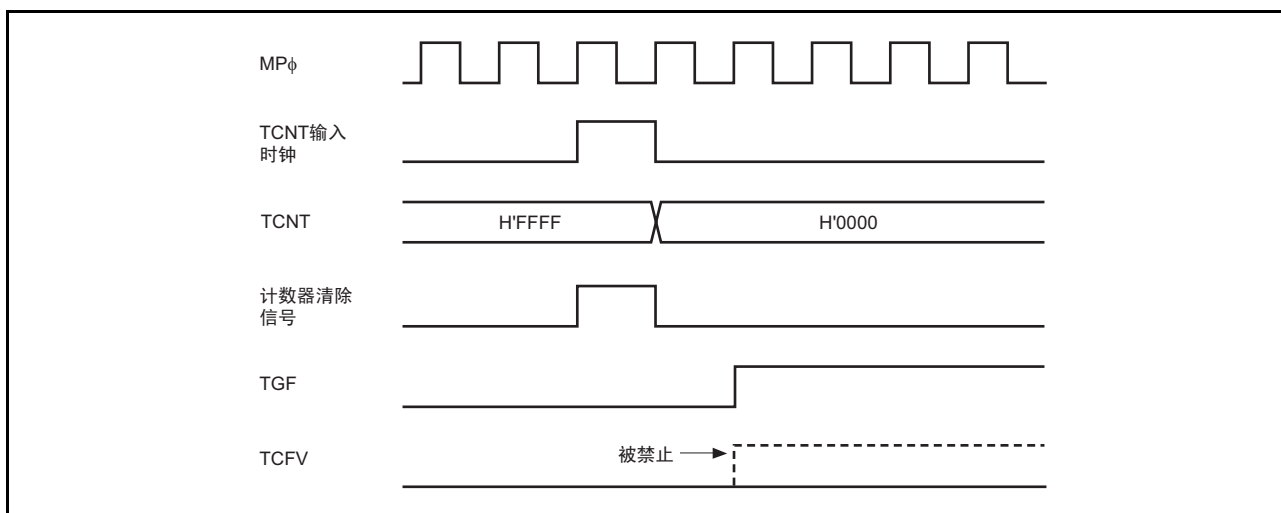


图 11.139 上溢和计数器清除的竞争

### 11.7.18 TCNT 的写操作和上溢 / 下溢的竞争

即使在 TCNT 写周期中的 T2 状态发生递增计数 / 递减计数和上溢 / 下溢，也优先写 TCNT 而不将 TSR 的 TCFV/TCFU 标志置位。

TCNT 的写操作和上溢发生竞争时的运行时序如图 11.140 所示。

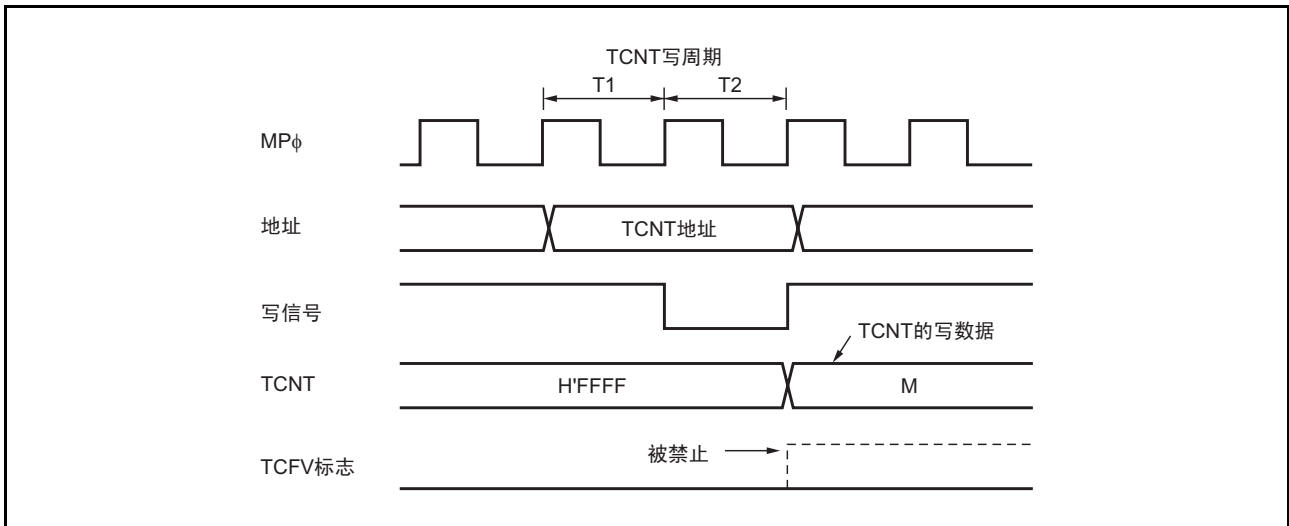


图 11.140 TCNT 的写操作和上溢的竞争

### 11.7.19 从通常运行或者 PWM 模式 1 转移到复位同步 PWM 模式时的注意事项

从通道 3、4 的通常运行或者 PWM 模式转移到复位同步 PWM 模式时，在将输出引脚（TIOC3B、TIOC3D、TIOC4A、TIOC4C、TIOC4B 和 TIOC4D）置为高电平的状态下停止计数器。必须注意：如果在转移到同步 PWM 模式后运行，就不能正确地进行引脚的初始输出。

要从通常运行转移到同步 PWM 模式时，必须在将 H'11 写到 TIORH\_3、TIORL\_3、TIORH\_4 和 TIORL\_4 寄存器并将输出引脚初始化为低电平后，先设定寄存器的初始值 H'00 再进行模式转移。

要从 PWM 模式 1 转移到复位同步 PWM 模式时，必须暂且先转移到通常运行，在将输出引脚初始化为低电平后，先设定寄存器的初始值 H'00 再转移到复位同步 PWM 模式。

### 11.7.20 互补 PWM 模式或者复位同步 PWM 模式的输出电平

在通道 3、4 为互补 PWM 模式或者复位同步 PWM 模式时，通过定时器的输出控制寄存器（TOCR）的 OLSP 和 OLSN 位设定 PWM 波形的输出电平，并且 TIOR 必须为 H'00。

### 11.7.21 模块待机时的中断

如果在请求中断的状态下变为模块待机，就不能清除 CPU 的中断源或者 DTC/DMAC 的启动源。必须预先禁止中断再设定为模块待机模式。



### 11.7.22 级联中 TCNT\_1 和 TCNT\_2 的同时输入捕捉

在将定时器的计数器 1、2 (TCNT\_1 和 TCNT\_2) 进行级联后用作 32 位计数器时, 即使 TIOC1A 和 TIOC2A 或者 TIOC1B 和 TIOC2B 同时进行输入捕捉的输入, 也会在与内部时钟同步将输入到 TCNT\_1 和 TCNT\_2 的外部输入捕捉信号取到内部时, 有可能因 TIOC1A、TIOC2A 或者 TIOC1B、TIOC2B 的取时序产生偏差而不能正确地捕捉级联计数器的值。

例如, TCNT\_1 (高 16 位的计数器) 应该捕捉由 TCNT\_2 (低 16 位的计数器) 的上溢产生的递增计数值, 却会捕捉递增计数前的计数值。此时, 应该将 TCNT\_1=H'FFF1 和 TCNT\_2=H'0000 的值传送到 TGRA\_1 和 TGRA\_2 或者 TGRB\_1 和 TGRB\_2, 却会误传送 TCNT\_1=H'FFF0 和 TCNT\_2=H'0000 的值。

在 MTU2 中追加输入 1 个输入捕捉就可以同时捕捉 TCNT\_1 和 TCNT\_2 的功能。如果使用本功能, 就可以在 TCNT\_1 和 TCNT\_2 的捕捉时序无偏差的状态下, 取得 32 位计数。详细内容请参照“11.3.8 定时器的输入捕捉控制寄存器 (TICCR)”。

## 11.8 MTU2 输出引脚的初始化方法

### 11.8.1 运行模式

MTU2 有以下 6 种运行模式，能在任意的模式中进行波形输出。

- 正常模式 (通道 0 ~ 4)
- PWM 模式 1 (通道 0 ~ 4)
- PWM 模式 2 (通道 0 ~ 2)
- 相位计数模式 1 ~ 4 (通道 1、2)
- 互补 PWM 模式 (通道 3、4)
- 复位同步 PWM 模式 (通道 3、4)

在此说明各模式中的 MTU2 输出引脚的初始化方法。

### 11.8.2 开始复位时的运行

MTU2 的输出引脚 (TIOC\*) 在复位或者待机模式时被初始化为低电平。因为通过引脚功能控制器 (PFC) 选择 MTU2 的引脚功能，所以在设定 PFC 时将当时的 MTU2 的引脚状态输出到端口。如果在复位后立即通过 PFC 选择 MTU2 的输出，就将 MTU2 输出的初始状态 (低电平) 输出到端口。在有效电平为低电平的情况下，因为系统马上就运行，所以必须在结束 MTU2 的输出引脚的初始设定后进行 PFC 的设定。

【注】 \* 通道号 + 端口符号。

### 11.8.3 因运行中的异常等而重新设定时的运行

如果在 MTU2 运行中发生异常，就必须通过系统截止 MTU2 的输出。即，通过 PFC 将引脚的输出转换为端口输出，并且输出有效电平的反相信号。另外，有关大电流引脚，也能使用端口输出的允许 (POE)，通过硬件截止输出。因运行中的异常等而重新设定时的引脚的初始化步骤以及重新设定后在其他运行模式中重新开始时的步骤如下所示。

如上所述，因为 MTU2 有 6 种运行模式，所以有 36 种模式转移的组合，但是有通道和模式的组合中不存在的转移，模式转移的组合一览表如表 11.59 所示。

表中使用下述的符号表示：

Normal: 正常模式

PWM1: PWM 模式 1

PWM2: PWM 模式 2

PCM: 相位计数模式 1 ~ 4

CPWM: 互补 PWM 模式

RPWM: 复位同步 PWM 模式

表 11.59 模式转移的组合

	Normal	PWM1	PWM2	PCM	CPWM	RPWM
Normal	(1)	(2)	(3)	(4)	(5)	(6)
PWM1	(7)	(8)	(9)	(10)	(11)	(12)
PWM2	(13)	(14)	(15)	(16)	无	无
PCM	(17)	(18)	(19)	(20)	无	无
CPWM	(21)	(22)	无	无	(23) (24)	(25)
RPWM	(26)	(27)	无	无	(28)	(29)

#### 11.8.4 因运行中的异常等而初始化引脚的步骤和模式转移的概要

- 在转移到定时器的 I/O 控制寄存器 (TIOR) 选择的引脚输出电平模式 (Normal、PWM1、PWM2、PCM) 时, 必须通过设定 TIOR 初始化引脚。
- 因为在 PWM 模式 1 中不将波形输出到 TIOC\*B (TIOC\*D) 引脚, 所以即使设定 TIOR 也不初始化引脚。要进行初始化时, 必须在正常模式中进行初始化后转移到 PWM 模式 1。
- 因为在 PWM 模式 2 中不将波形输出到周期寄存器的引脚, 所以即使设定 TIOR 也不初始化引脚。要进行初始化时, 必须在正常模式中进行初始化后转移到 PWM 模式 2。
- 在正常模式或者 PWM 模式 2 中, 如果 TGRC 和 TGRD 都用作缓冲寄存器, 即使设定 TIOR 也不初始化缓冲寄存器的引脚。要进行初始化时, 必须在解除缓冲模式进行初始化后重新设定缓冲模式。
- 在 PWM 模式 1 中, 如果 TGRC 或者 TGRD 用作缓冲寄存器, 即使设定 TIOR 也不初始化 TGRC 的引脚。要初始化 TGRC 的引脚时, 必须在解除缓冲模式进行初始化后重新设定缓冲模式。
- 在转移到定时器的输出控制寄存器 (TOCR) 选择的引脚输出电平模式 (CPWM、RPWM) 时, 必须转移到正常模式并且通过 TIOR 进行初始化, 在将 TIOR 返回到初始值后, 通过定时器的输出器件允许寄存器 (TOER) 暂时禁止通道 3、4 的输出, 然后按照模式的设定步骤 (TOCR 的设定、TMDR 的设定和 TOER 的设定) 运行。

【注】 本项记述中的 \* 为通道号。

按照表 11.59 的组合 No. 初始化引脚的步骤如下所示。另外，有效电平为低电平。

(1) 在正常模式的运行中发生异常并在正常模式中重新开始时的运行

在正常模式中发生异常，重新设定后在正常模式中重新开始的说明如图 11.141 所示。

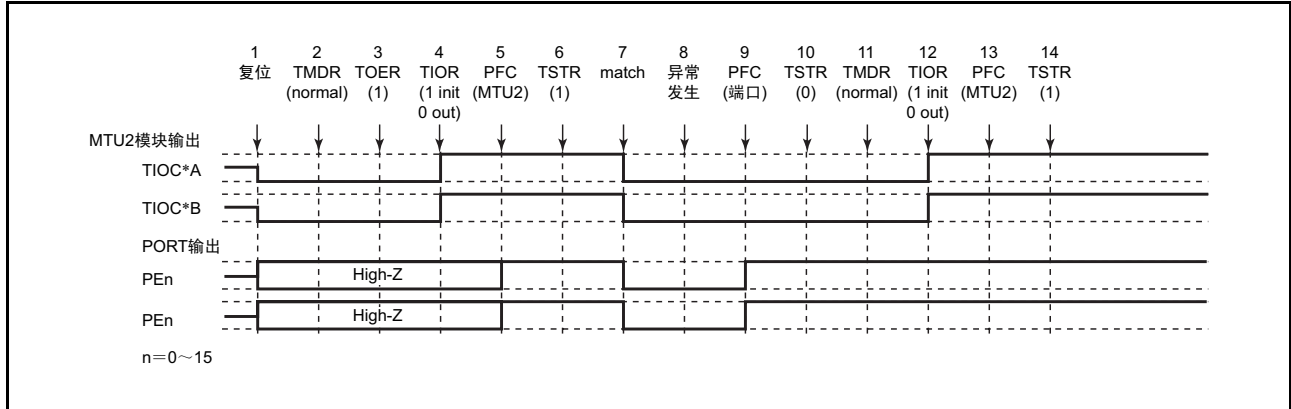


图 11.141 在正常模式中发生异常并在正常模式中恢复

1. 在复位后，MTU2 的输出为低电平，端口为高阻抗状态。
2. 在复位后，TMDR 为正常模式。
3. 在通道 3、4 中通过 TIOR 初始化引脚前，必须通过 TOER 允许输出。
4. 必须通过 TIOR 初始化引脚（例中，初始输出为高电平，比较匹配为低电平输出）。
5. 必须通过 PFC 设定为 MTU2 的输出。
6. 通过 TSTR 开始计数。
7. 通过产生比较匹配，输出低电平。
8. 发生了异常。
9. 必须通过 PFC 设定为端口输出，输出有效电平的反相信号。
10. 通过 TSTR 停止计数。
11. 在正常模式中重新开始时不需要此步骤。
12. 必须通过 TIOR 初始化引脚。
13. 必须通过 PFC 设定为 MTU2 的输出。
14. 通过 TSTR 重新开始。

(2) 在正常模式的运行中发生异常并在 PWM 模式 1 中重新开始时的运行

在正常模式中发生异常，重新设定后在 PWM 模式 1 中重新开始时的说明如图 11.142 所示。

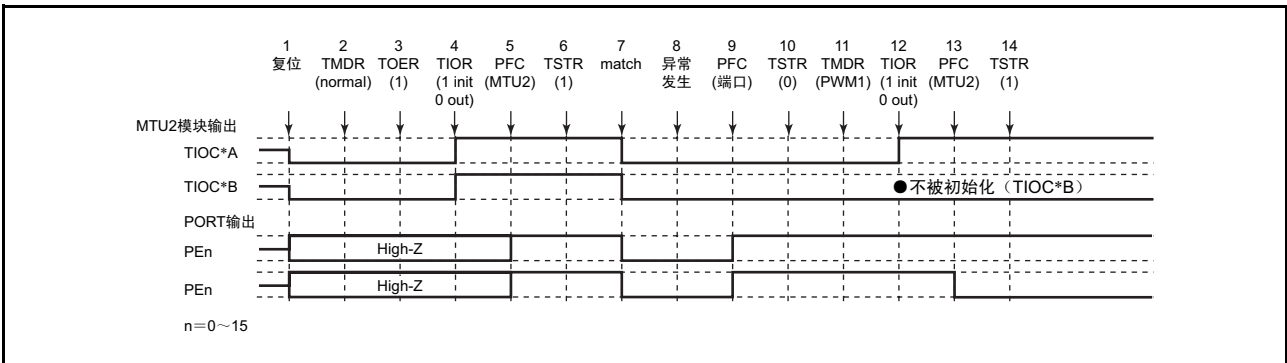


图 11.142 在正常模式中发生异常并在 PWM 模式 1 中恢复

1 ~ 10 和图 11.141 通用。

11. 设定 PWM 模式 1。

12. 必须通过 TIOR 初始化引脚（在 PWM 模式 1 中不初始化 TIOC\*B。要进行初始化时，必须在正常模式中进行初始化后转移到 PWM 模式 1）。

13. 必须通过 PFC 设定为 MTU2 的输出。

14. 通过 TSTR 重新开始。

(3) 在正常模式的运行中发生异常并在 PWM 模式 2 中重新开始时的运行

在正常模式中发生异常，重新设定后在 PWM 模式 2 中重新开始时的说明如图 11.143 所示。

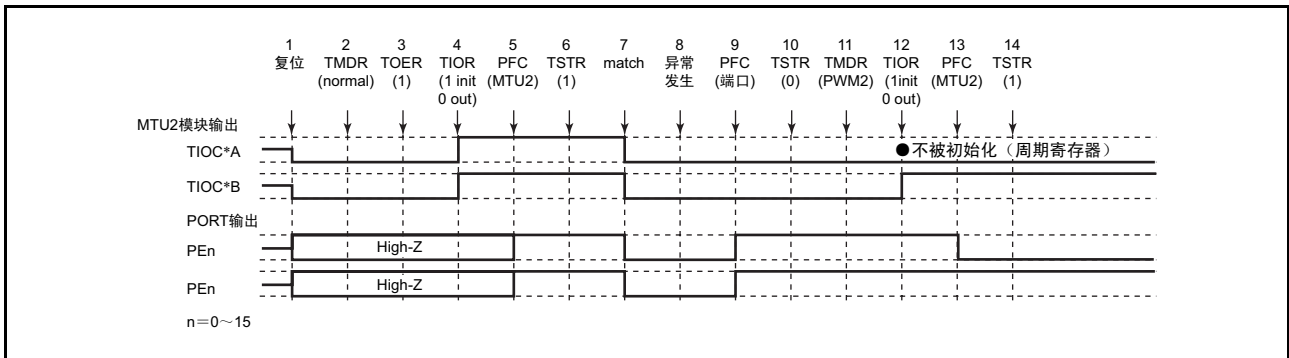


图 11.143 在正常模式中发生异常并在 PWM 模式 2 中恢复

1 ~ 10 和图 11.141 通用。

11. 设定 PWM 模式 2。

12. 必须通过 TIOR 初始化引脚（在 PWM 模式 2 中不初始化周期寄存器的引脚。要进行初始化时，必须在正常模式中进行初始化后转移到 PWM 模式 2）。

13. 通过 PFC 设定为 MTU2 的输出。

14. 通过 TSTR 重新开始。

**【注】** 只能在通道 0 ~ 2 中设定 PWM 模式 2，因此不需要设定 TOER。

(4) 在正常模式的运行中发生异常并在相位计数模式中重新开始时的运行

在正常模式中发生异常，重新设定后在相位计数模式中重新开始的说明如图 11.144 所示。

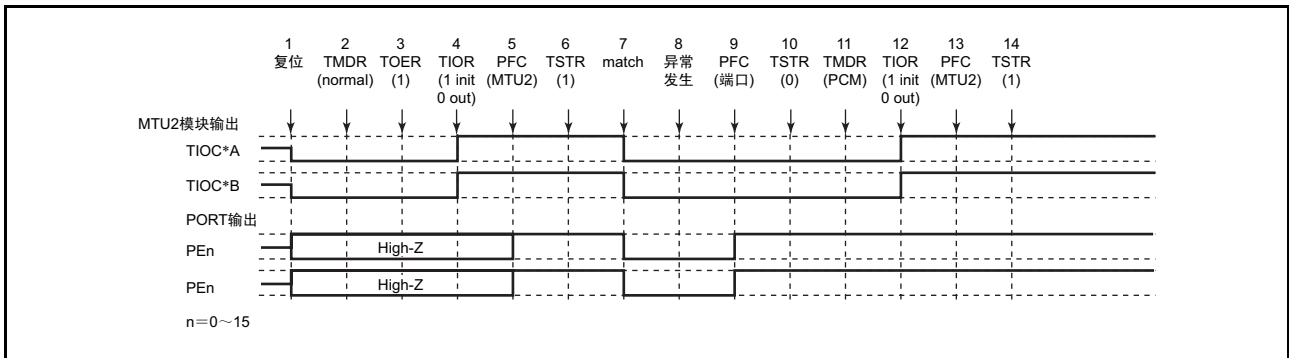


图 11.144 在正常模式中发生异常并在相位计数模式中恢复

1 ~ 10 和图 11.141 通用。

11. 设定相位计数模式。
12. 必须通过 TIOR 初始化引脚。
13. 必须通过 PFC 设定为 MTU2 的输出。
14. 通过 TSTR 重新开始。

**【注】** 只能在通道 1、2 中设定相位计数模式，因此不需要设定 TOER。

(5) 在正常模式的运行中发生异常并在互补 PWM 模式中重新开始时的运行

在正常模式中发生异常，重新设定后在互补 PWM 模式中重新开始的说明如图 11.145 所示。

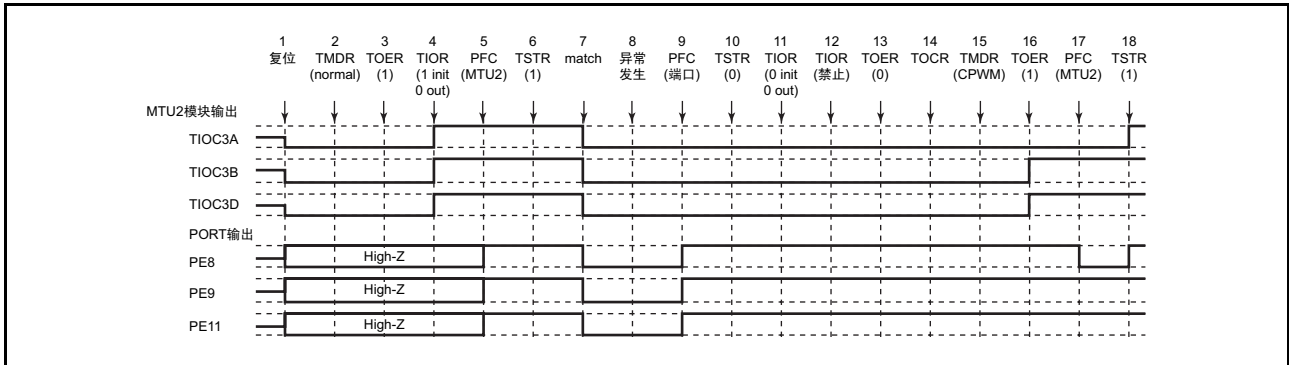


图 11.145 在正常模式中发生异常并在互补 PWM 模式中恢复

1 ~ 10 和图 11.141 通用。

11. 必须通过 TIOR 初始化正常模式的波形生成部。
12. 必须通过 TIOR 禁止正常模式的波形生成部的运行。
13. 必须通过 TOER 禁止通道 3、4 的输出。
14. 必须通过 TOCR 选择互补 PWM 的输出电平并允许或者禁止周期输出。
15. 设定互补 PWM。
16. 必须通过 TOER 允许通道 3、4 的输出。
17. 必须通过 PFC 设定为 MTU2 的输出。
18. 通过 TSTR 重新开始。



(6) 在正常模式的运行中发生异常并在复位同步 PWM 模式中重新开始时的运行

在正常模式中发生异常，重新设定后在复位同步 PWM 模式中重新开始的说明如图 11.146 所示。

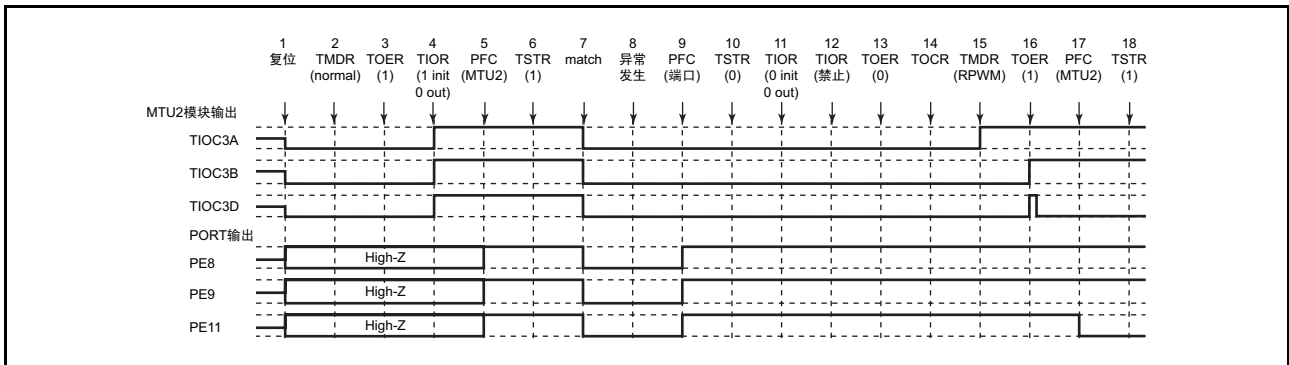


图 11.146 在正常模式中发生异常并在复位同步 PWM 模式中恢复

1 ~ 13 和图 11.141 通用。

14. 必须通过 TOCR 选择复位同步 PWM 的输出电平并允许或者禁止周期输出。

15. 设定复位同步 PWM。

16. 必须通过 TOER 允许通道 3、4 的输出。

17. 必须通过 PFC 设定为 MTU2 的输出。

18. 通过 TSTR 重新开始。

## (7) 在 PWM 模式 1 的运行中发生异常并在正常模式中重新开始时的运行

在 PWM 模式 1 中发生异常，重新设定后在正常模式中重新开始时的说明如图 11.147 所示。

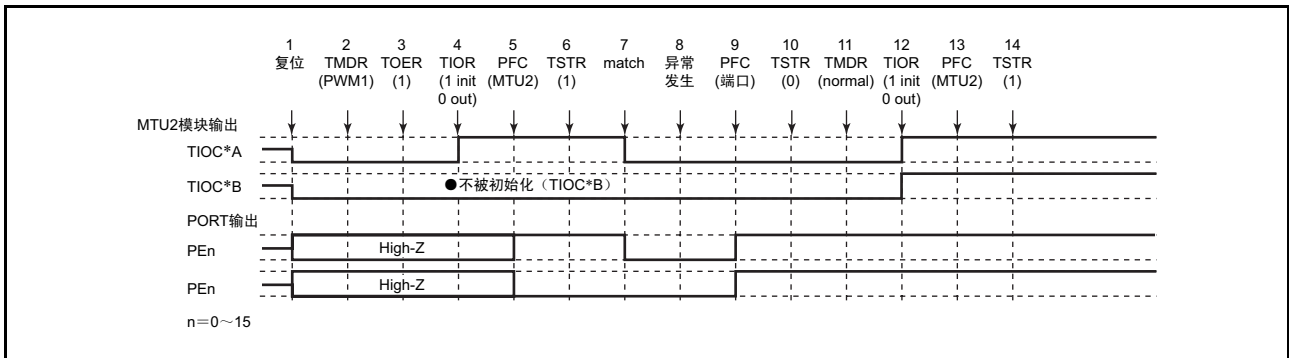


图 11.147 在 PWM 模式 1 中发生异常并在正常模式中恢复

1. 在复位后，MTU2 的输出为低电平，端口为高阻抗状态。
2. 必须设定 PWM 模式 1。
3. 在通道 3、4 中通过 TIOR 初始化引脚前，必须通过 TOER 允许输出。
4. 必须通过 TIOR 初始化引脚（例中，初始输出为高电平，比较匹配为低电平输出，在 PWM 模式 1 中不初始化 TIOC\*B）。
5. 通过 PFC 设定为 MTU2 的输出。
6. 通过 TSTR 开始计数。
7. 通过产生比较匹配，输出低电平。
8. 发生了异常。
9. 必须通过 PFC 设定为端口输出，输出有效电平的反相信号。
10. 通过 TSTR 停止计数。
11. 必须设定正常模式。
12. 必须通过 TIOR 初始化引脚。
13. 必须通过 PFC 设定为 MTU2 的输出。
14. 通过 TSTR 重新开始。

(8) 在 PWM 模式 1 的运行中发生异常并在 PWM 模式 1 中重新开始时的运行

在 PWM 模式 1 中发生异常，重新设定后在 PWM 模式 1 中重新开始时的说明如图 11.148 所示。

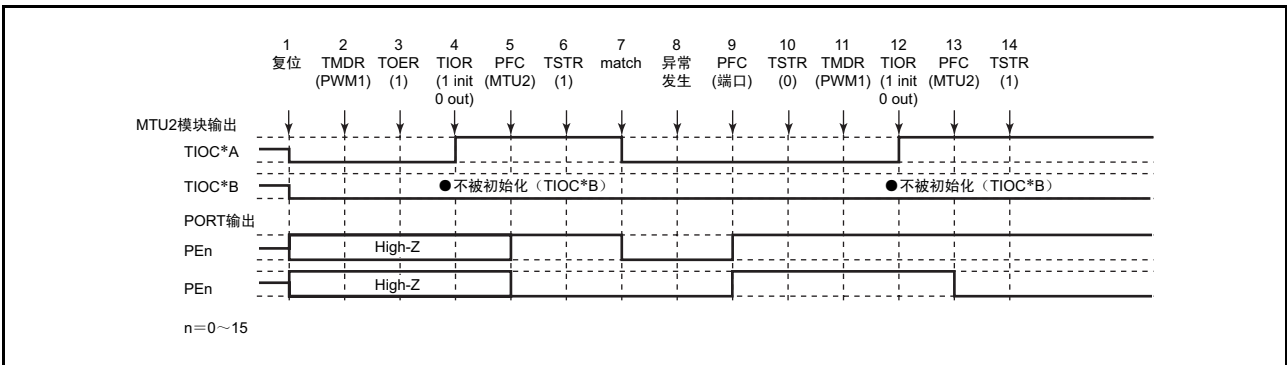


图 11.148 在 PWM 模式 1 中发生异常并在 PWM 模式 1 中恢复

1 ~ 10 和图 11.147 通用。

11. 在 PWM 模式 1 中重新开始时不需要此步骤。

12. 必须通过 TIOR 初始化引脚（在 PWM 模式 1 中不初始化 TIOC\*B）。

13. 必须通过 PFC 设定为 MTU2 的输出。

14. 通过 TSTR 重新开始。

(9) 在 PWM 模式 1 的运行中发生异常并在 PWM 模式 2 中重新开始时的运行

在 PWM 模式 1 中发生异常，重新设定后在 PWM 模式 2 中重新开始的说明如图 11.149 所示。

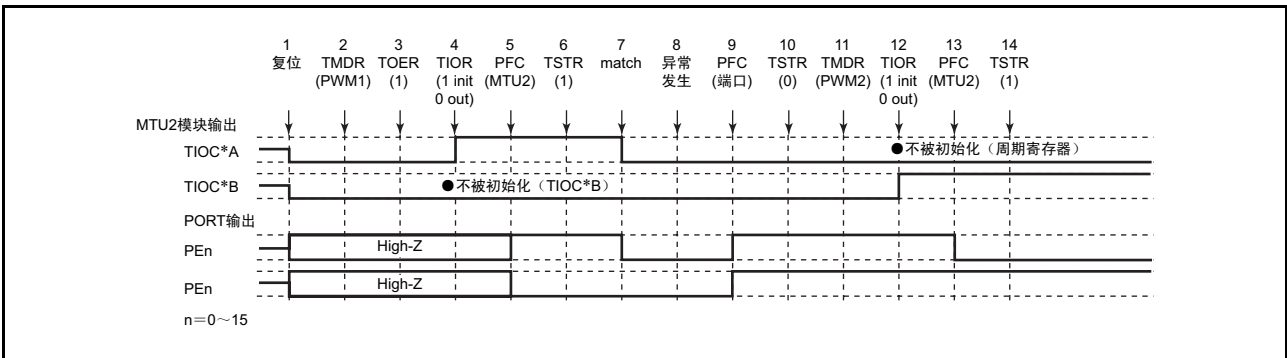


图 11.149 在 PWM 模式 1 中发生异常并在 PWM 模式 2 中恢复

1 ~ 10 和图 11.147 通用。

11. 设定 PWM 模式 2。

12. 必须通过 TIOR 初始化引脚（在 PWM 模式 2 中不初始化周期寄存器的引脚）。

13. 必须通过 PFC 设定为 MTU2 的输出。

14. 通过 TSTR 重新开始。

**【注】** 只能在通道 0 ~ 2 中设定 PWM 模式 2，因此不需要设定 TOER。

(10) 在 PWM 模式 1 的运行中发生异常并在相位计数模式中重新开始时的运行

在 PWM 模式 1 中发生异常，重新设定后在相位计数模式中重新开始时的说明如图 11.150 所示。

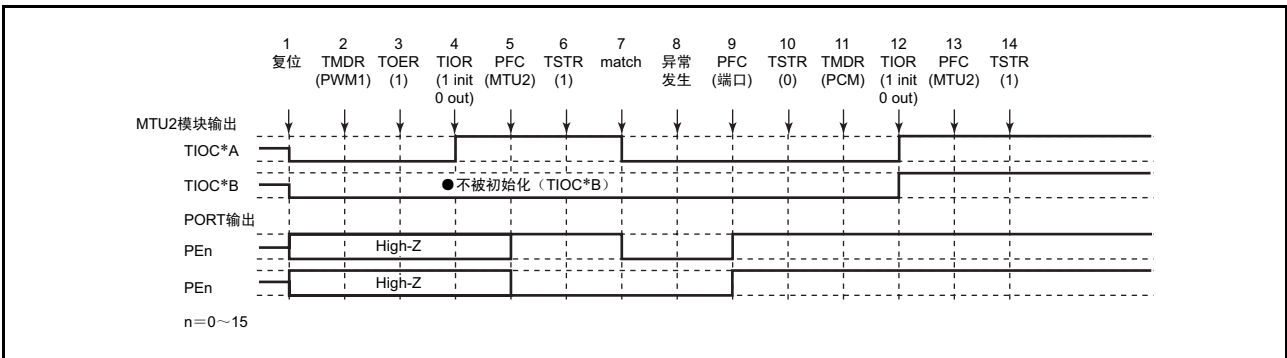


图 11.150 在 PWM 模式 1 中发生异常并在相位计数模式中恢复

1 ~ 10 和图 11.147 通用。

11. 设定相位计数模式。
12. 必须通过 TIOR 初始化引脚。
13. 必须通过 PFC 设定为 MTU2 的输出。
14. 通过 TSTR 重新开始。

**【注】** 只能在通道 1、2 中设定相位计数模式，因此不需要设定 TOER。

## (11) 在 PWM 模式 1 的运行中发生异常并在互补 PWM 模式中重新开始时的运行

在 PWM 模式 1 中发生异常，重新设定后在互补 PWM 模式中重新开始的说明如图 11.151 所示。

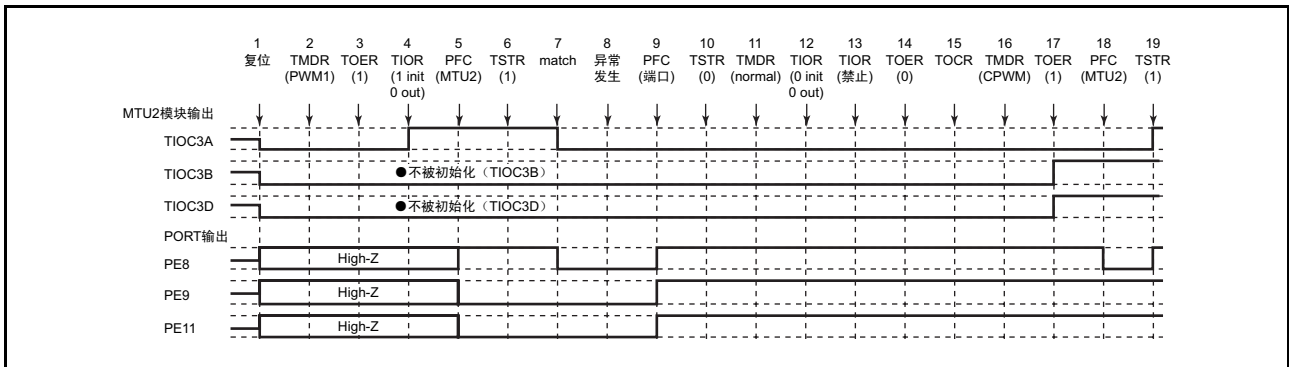


图 11.151 在 PWM 模式 1 中发生异常并在互补 PWM 模式中恢复

1 ~ 10 和图 11.147 通用。

11. 为了初始化波形生成部，必须设定正常模式。
12. 必须通过 TIOR 初始化 PWM 模式 1 的波形生成部。
13. 必须通过 TIOR 禁止 PWM 模式 1 的波形生成部的运行
14. 必须通过 TOER 禁止通道 3、4 的输出。
15. 必须通过 TOCR 选择 PWM 的输出电平并允许或者禁止周期输出。
16. 设定互补 PWM。
17. 必须通过 TOER 允许通道 3、4 的输出。
18. 必须通过 PFC 设定为 MTU2 的输出。
19. 通过 TSTR 重新开始。

(12) 在 PWM 模式 1 的运行中发生异常并在复位同步 PWM 模式中重新开始时的运行

在 PWM 模式 1 中发生异常，重新设定后在复位同步 PWM 模式中重新开始的说明如图 11.152 所示。

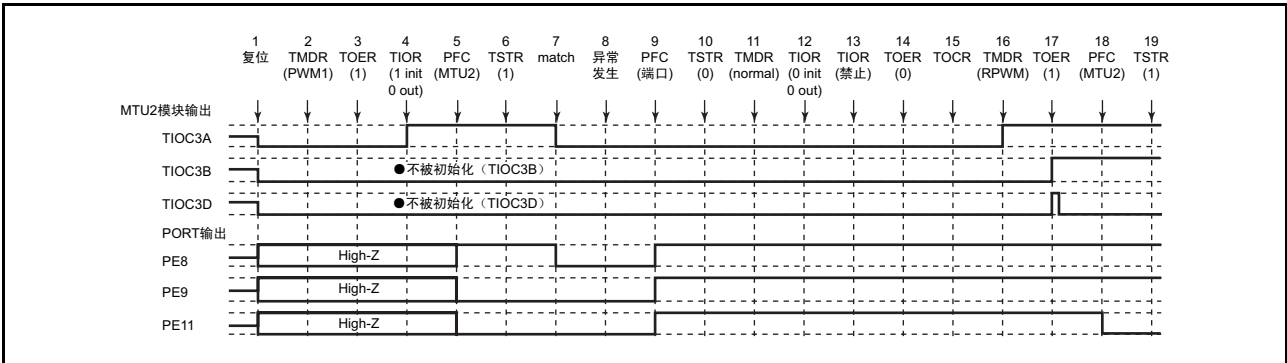


图 11.152 在 PWM 模式 1 中发生异常并在复位同步 PWM 模式中恢复

1 ~ 14 和图 11.151 通用。

15. 必须通过 TOCR 选择同步 PWM 的输出电平并允许或者禁止周期输出。

16. 设定复位同步 PWM。

17. 必须通过 TOER 允许通道 3、4 的输出。

18. 通过 PFC 设定为 MTU2 的输出。

19. 通过 TSTR 重新开始。

## (13) 在 PWM 模式 2 的运行中发生异常并在正常模式中重新开始时的运行

在 PWM 模式 2 中发生异常，重新设定后在正常模式中重新开始时的说明如图 11.153 所示。

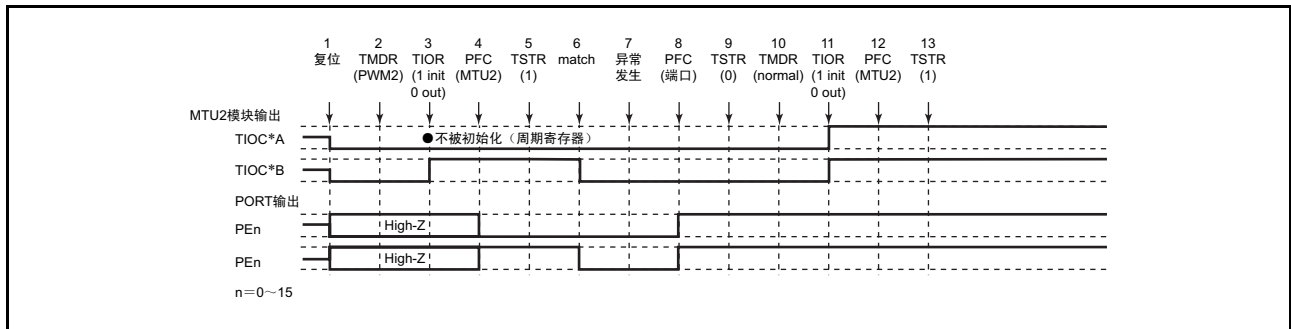


图 11.153 在 PWM 模式 2 中发生异常并在正常模式中恢复

1. 在复位后，MTU2 的输出为低电平，端口为高阻抗状态。
2. 必须设定 PWM 模式 2。
3. 必须通过 TIOR 初始化引脚（例中，初始输出为高电平，比较匹配为低电平输出，在 PWM 模式 2 中不初始化周期寄存器的引脚，TIOC\*A 为周期寄存器）。
4. 必须通过 PFC 设定为 MTU2 的输出。
5. 通过 TSTR 开始计数。
6. 通过产生比较匹配，输出低电平。
7. 发生了异常。
8. 必须通过 PFC 设定端口输出，输出有效电平的反相信号。
9. 通过 TSTR 停止计数。
10. 必须设定正常模式。
11. 必须通过 TIOR 初始化引脚。
12. 必须通过 PFC 设定为 MTU2 的输出。
13. 通过 TSTR 重新开始。



(14) 在 PWM 模式 2 的运行中发生异常并在 PWM 模式 1 中重新开始时的运行

在 PWM 模式 2 中发生异常，重新设定后在 PWM 模式 1 中重新开始的说明如图 11.154 所示。

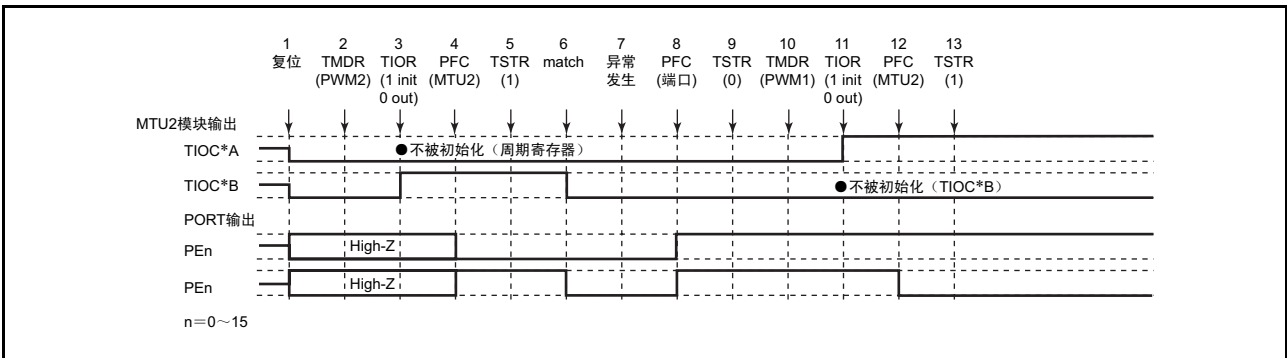


图 11.154 在 PWM 模式 2 中发生异常并在 PWM 模式 1 中恢复

1 ~ 9 和图 11.153 通用。

10. 设定 PWM 模式 1。

11. 必须通过 TIOR 初始化引脚（在 PWM 模式 1 中不初始化 TIOC\*B）。

12. 必须通过 PFC 设定为 MTU2 的输出。

13. 通过 TSTR 重新开始。

(15) 在 PWM 模式 2 的运行中发生异常并在 PWM 模式 2 中重新开始时的运行

在 PWM 模式 2 中发生异常，重新设定后在 PWM 模式 2 中重新开始时的说明如图 11.155 所示。

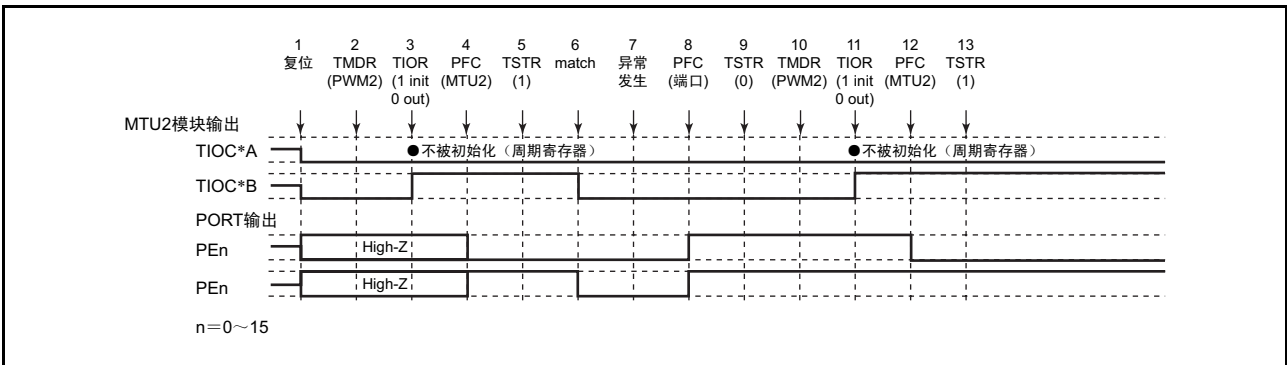


图 11.155 PWM 模式 2 中发生异常并在 PWM 模式 2 中恢复

1 ~ 9 和图 11.153 通用。

10. 在 PWM 模式 2 中重新开始时不需要此步骤。

11. 必须通过 TIOR 初始化引脚 (在 PWM 模式 2 中不初始化周期寄存器的引脚)。

12. 必须通过 PFC 设定为 MTU2 的输出。

13. 通过 TSTR 重新开始。

(16) 在 PWM 模式 2 的运行中发生异常并在相位计数模式中重新开始时的运行

在 PWM 模式 2 中发生异常，重新再设定后在位相计数模式中重新开始的说明如图 11.156 所示。

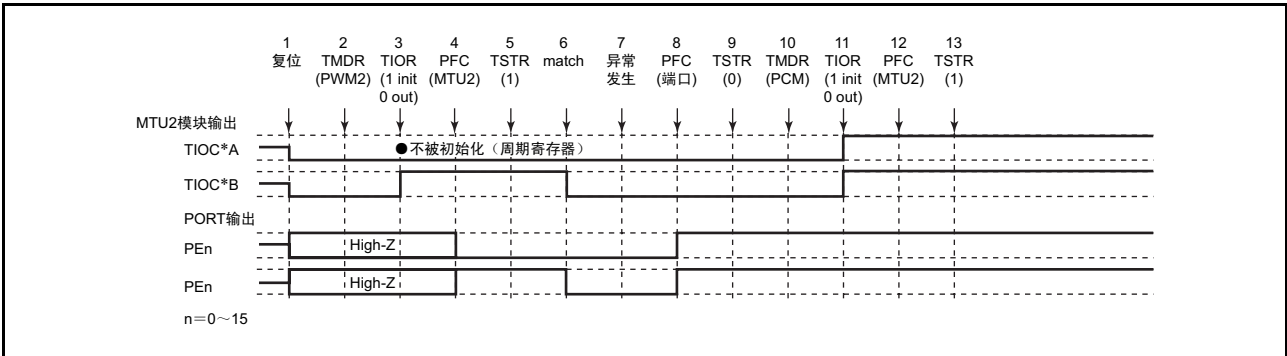


图 11.156 在 PWM 模式 2 中发生异常并在相位计数模式中恢复

1 ~ 9 和图 11.153 通用。

10. 设定相位计数模式。

11. 必须通过 TIOR 初始化引脚。

12. 必须通过 PFC 设定为 MTU2 的输出。

13. 通过 TSTR 重新开始。

## (17) 在相位计数模式的运行中发生异常并在正常模式中重新开始时的运行

在相位计数模式中发生异常，重新设定后在正常模式中重新开始的说明如图 11.157 所示。

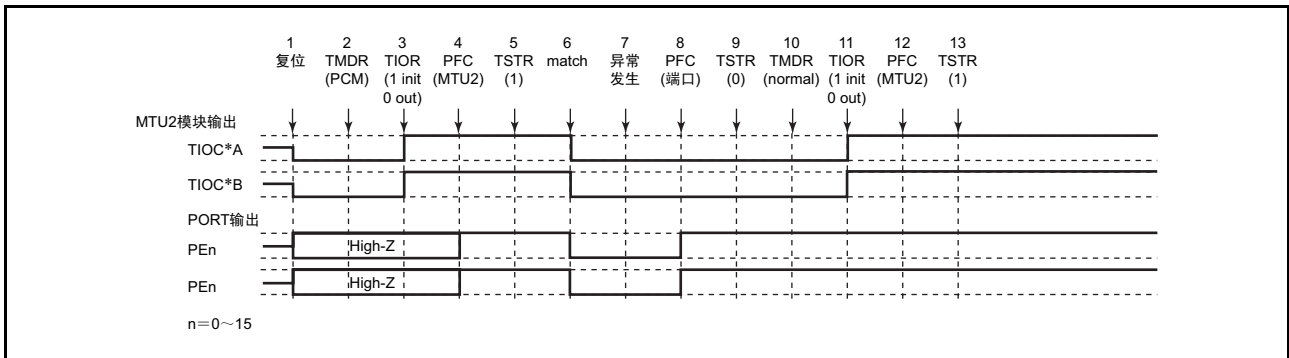


图 11.157 在相位计数模式中发生异常并在正常模式中恢复

1. 在复位后，MTU2 的输出为低电平，端口为高阻抗状态。
2. 必须设定相位计数模式。
3. 必须通过 TIOR 初始化引脚（例中，初始输出为高电平，比较输出为低电平输出）。
4. 必须通过 PFC 设定为 MTU2 的输出。
5. 通过 TSTR 开始计数。
6. 通过产生比较匹配，输出低电平。
7. 发生了异常。
8. 必须通过 PFC 设定端口输出，输出有效电平的反相信号。
9. 通过 TSTR 停止计数。
10. 必须在正常模式中进行设定。
11. 必须通过 TIOR 初始化引脚。
12. 必须通过 PFC 设定为 MTU2 的输出。
13. 通过 TSTR 重新开始。

## (18) 在相位计数模式的运行中发生异常并在 PWM 模式 1 中重新开始时的运行

在相位计数模式中发生异常，重新设定后在 PWM 模式 1 中重新开始时的说明如图 11.158 所示。

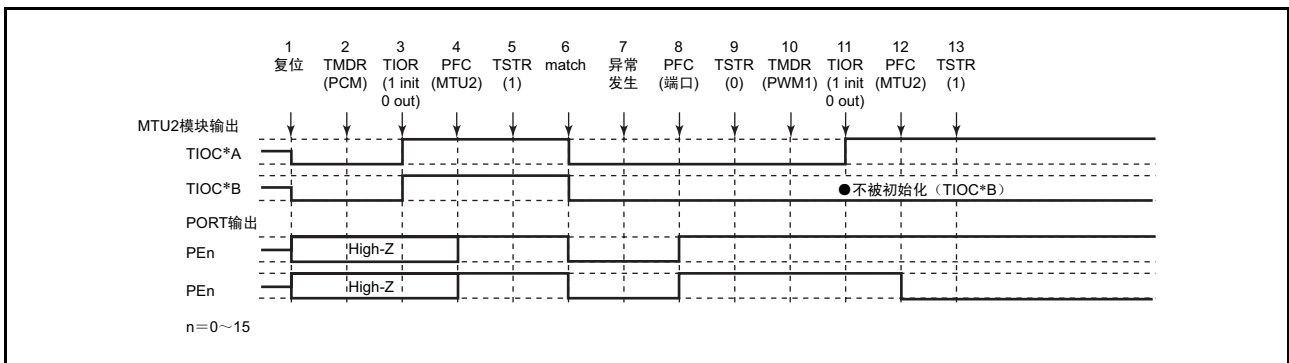


图 11.158 在相位计数模式中发生异常并在 PWM 模式 1 中恢复

1 ~ 9 和图 11.157 通用。

10. 设定 PWM 模式 1。

11. 必须通过 TIOR 初始化引脚（在 PWM 模式 1 中不初始化 TIOC\*B）。

12. 必须通过 PFC 设定为 MTU2 的输出。

13. 通过 TSTR 重新开始。

(19) 在相位计数模式的运行中发生异常并在 PWM 模式 2 中重新开始时的运行

在相位计数模式中发生异常，重新设定后在 PWM 模式 2 中重新开始时的说明如图 11.159 所示。

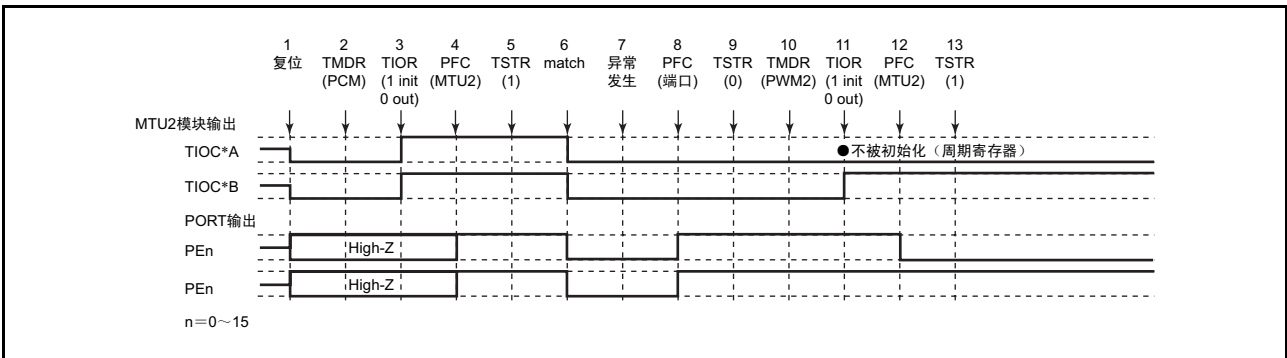


图 11.159 在相位计数模式中发生异常并在 PWM 模式 2 中恢复

1 ~ 9 和图 11.157 通用。

10. 设定 PWM 模式 2。

11. 必须通过 TIOR 初始化引脚（在 PWM 模式 2 中不初始化周期寄存器的引脚）。

12. 必须通过 PFC 设定为 MTU2 的输出。

13. 通过 TSTR 重新开始。

(20) 在相位计数模式的运行中发生异常并在相位计数模式中重新开始时的运行

在相位计数模式中发生异常，重新设定后在相位计数模式中重新开始时的说明如图 11.160 所示。

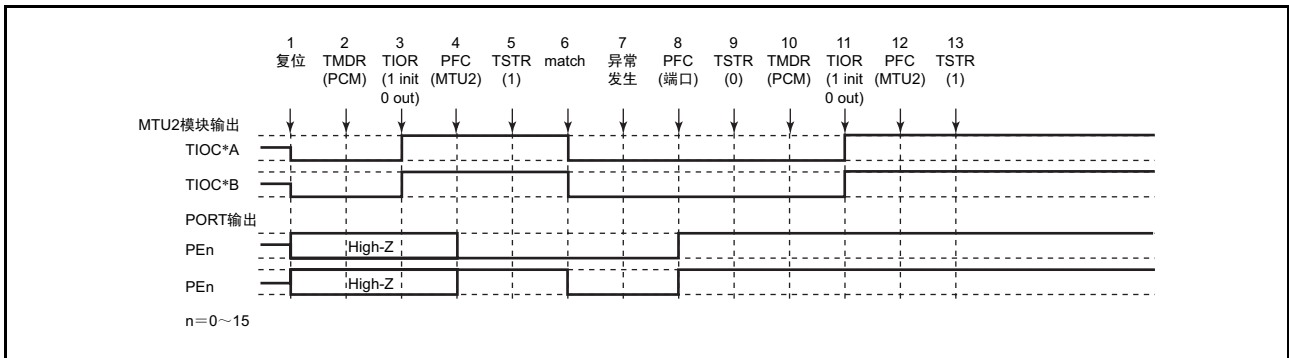


图 11.160 在相位计数模式中发生异常并在相位计数模式中恢复

1 ~ 9 和图 11.157 通用。

10. 在相位计数模式中重新开始时不需要此步骤。

11. 必须通过 TIOR 初始化引脚。

12. 必须通过 PFC 设定为 MTU2 的输出。

13. 通过 TSTR 重新开始。

## (21) 在互补 PWM 模式的运行中发生异常并在正常模式中重新开始时的运行

在互补 PWM 模式中发生异常，重新设定后在正常模式中重新开始时的说明如图 11.161 所示。

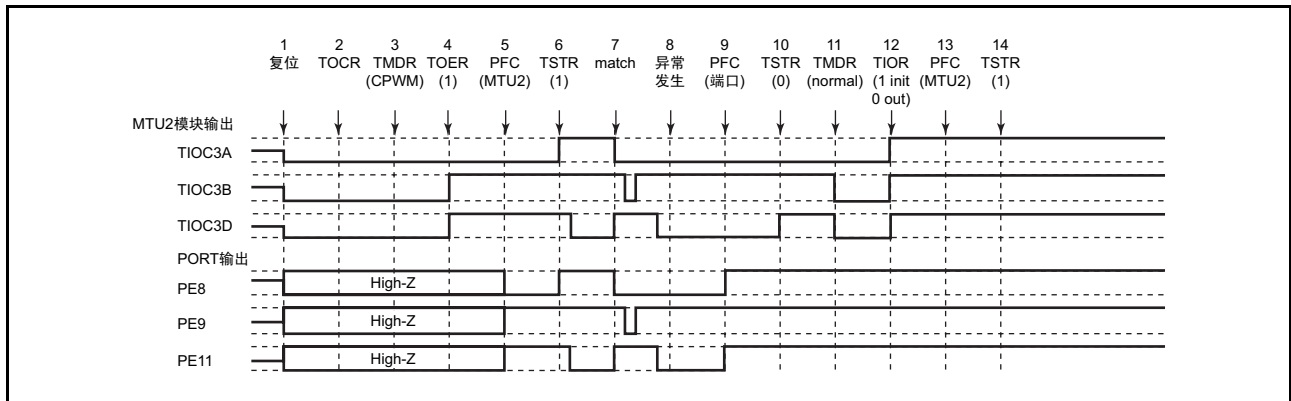


图 11.161 在互补 PWM 模式中发生异常并在正常模式中恢复

1. 在复位后，MTU2 的输出为低电平，端口为高阻抗状态。
2. 必须通过 TOCR 选择互补 PWM 的输出电平并允许或者禁止周期输出。
3. 设定互补 PWM。
4. 必须通过 TOER 允许通道 3、4 的输出。
5. 必须通过 PFC 设定为 MTU2 的输出。
6. 通过 TSTR 开始计数。
7. 通过产生比较匹配，输出互补 PWM 波形。
8. 发生了异常。
9. 必须通过 PFC 设定为端口输出，输出有效电平的反相信号。
10. 通过 TSTR 停止计数 (MTU2 的输出为互补 PWM 输出的初始值)。
11. 必须设定正常模式 (MTU2 的输出为低电平)。
12. 必须通过 TIOR 初始化引脚。
13. 必须通过 PFC 设定为 MTU2 的输出。
14. 通过 TSTR 重新开始。



(22) 在互补 PWM 模式的运行中发生异常并在 PWM 模式 1 中重新开始时的运行

在互补 PWM 模式中发生异常，重新设定后在 PWM 模式 1 中重新开始时的说明如图 11.162 所示。

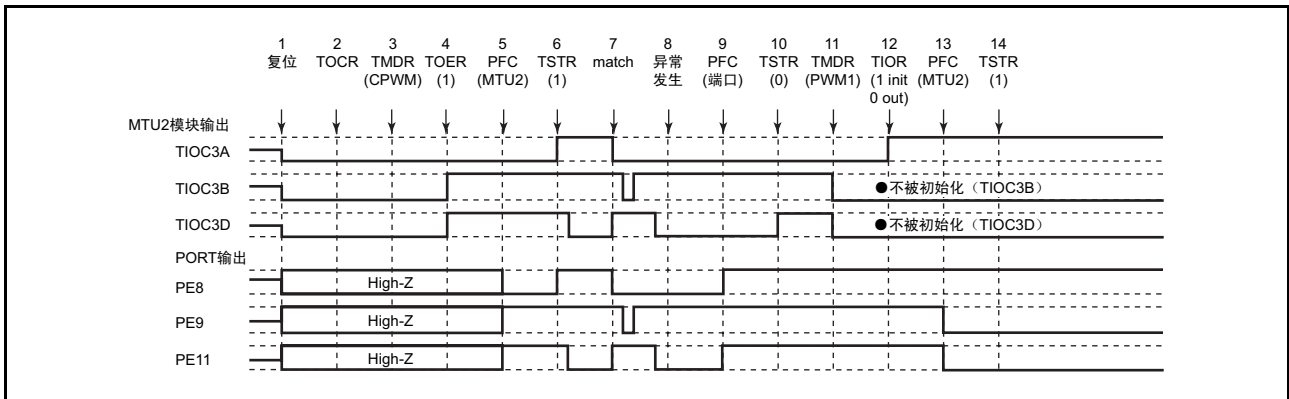


图 11.162 在互补 PWM 模式中发生异常并在 PWM 模式 1 中恢复

1 ~ 10 和图 11.161 通用。

- 11. 必须设定 PWM 模式 1 (MTU2 的输出为低电平)。
- 12. 必须通过 TIOR 初始化引脚 (在 PWM 模式 1 中不初始化 TIOC\*B)。
- 13. 必须通过 PFC 设定为 MTU2 的输出。
- 14. 通过 TSTR 重新开始。

(23) 在互补 PWM 模式的运行中发生异常并在互补 PWM 模式中重新开始时的运行

在互补 PWM 模式中发生异常，重新设定后在互补 PWM 模式中重新开始（从停止计数器时的周期和占空比的设定值重新开始）时的说明如图 11.163 所示。

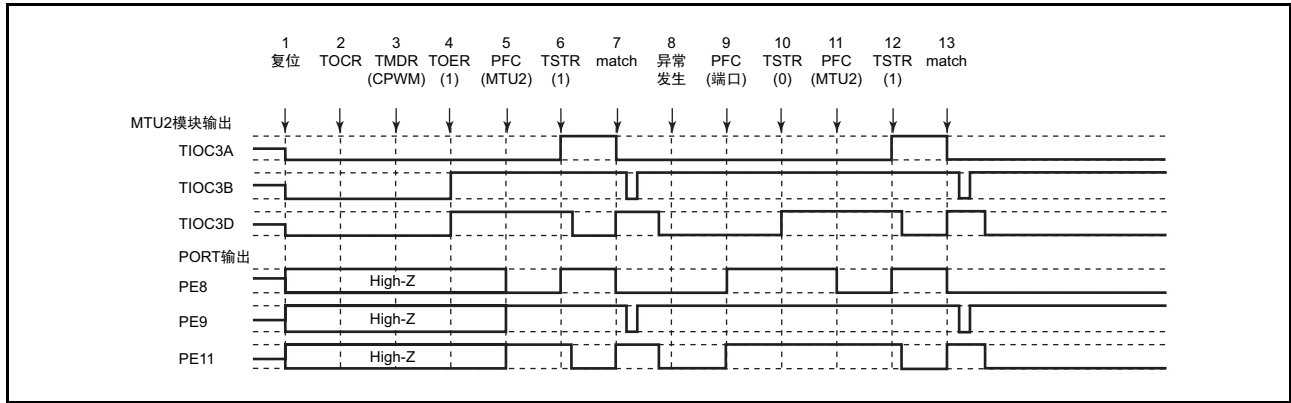


图 11.163 在互补 PWM 模式中发生异常并在互补 PWM 模式中恢复

1 ~ 10 和图 11.161 通用。

11. 必须通过 PFC 设定为 MTU2 的输出。

12. 通过 TSTR 重新开始。

13. 通过产生比较匹配，输出互补 PWM 波形。

(24) 在互补 PWM 模式的运行中发生异常并在互补 PWM 模式中重新开始时的运行

在互补 PWM 模式中发生异常，重新设定后在互补 PWM 模式中重新开始（从周期和占空比的新设定值重新开始）时的说明如图 11.164 所示。

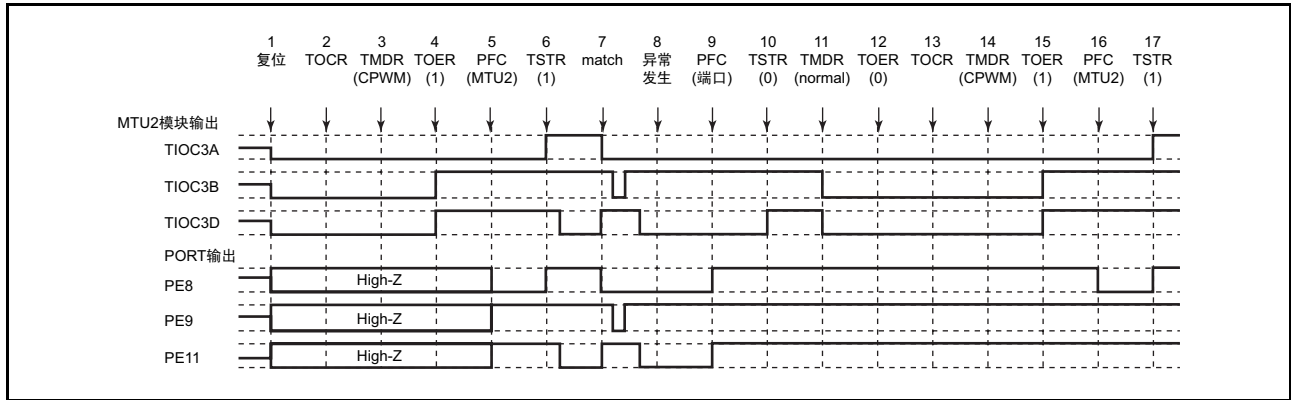


图 11.164 在互补 PWM 模式中发生异常并在互补 PWM 模式中恢复

1 ~ 10 和图 11.161 通用。

11. 必须设定正常模式和新的设定值（MTU2 的输出为低电平）。

12. 必须通过 TOER 禁止通道 3、4 的输出。

13. 必须通过 TOCR 选择互补 PWM 模式的输出电平并允许或者禁止周期输出。

14. 设定互补 PWM。

15. 必须通过 TOER 允许通道 3、4 的输出。

16. 必须通过 PFC 设定为 MTU2 的输出。

17. 通过 TSTR 重新开始。

## (25) 在互补 PWM 模式的运行中发生异常并在复位同步 PWM 模式中重新开始时的运行

在互补 PWM 模式中发生异常，重新设定后在复位同步 PWM 模式中重新开始时的说明如图 11.165 所示。

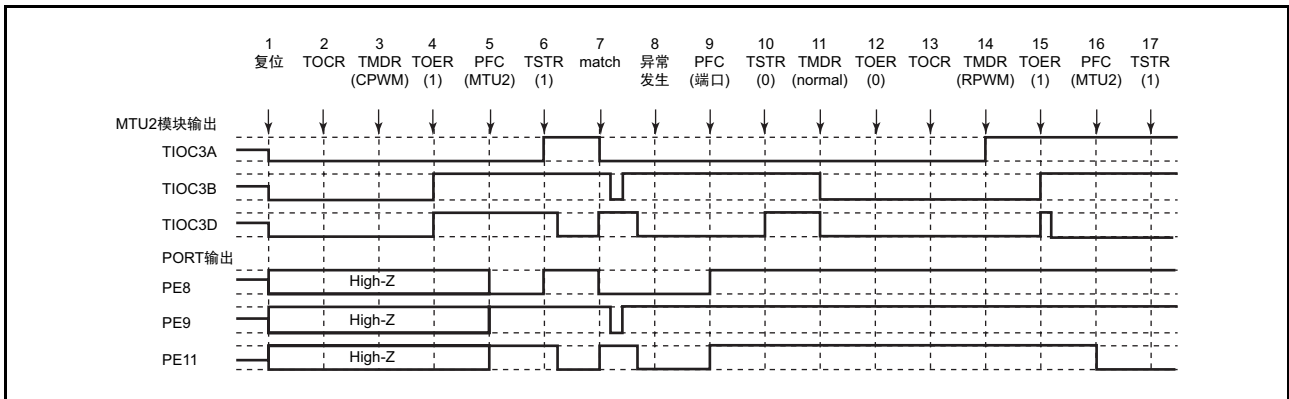


图 11.165 在互补 PWM 模式中发生异常并在复位同步 PWM 模式中恢复

1 ~ 10 和图 11.161 通用。

11. 必须设定正常模式 (MTU2 的输出为低电平)。
12. 必须通过 TOER 禁止通道 3、4 的输出。
13. 必须通过 TOCR 选择复位同步 PWM 模式的输出电平并允许或者禁止周期输出。
14. 设定复位同步 PWM。
15. 必须通过 TOER 允许通道 3、4 的输出。
16. 必须通过 PFC 设定为 MTU2 的输出。
17. 通过 TSTR 重新开始。

## (26) 在复位同步 PWM 模式的运行中发生异常并在正常模式中重新开始时的运行

在复位同步 PWM 模式中发生异常，重新设定后在正常模式中重新开始的说明如图 11.166 所示。

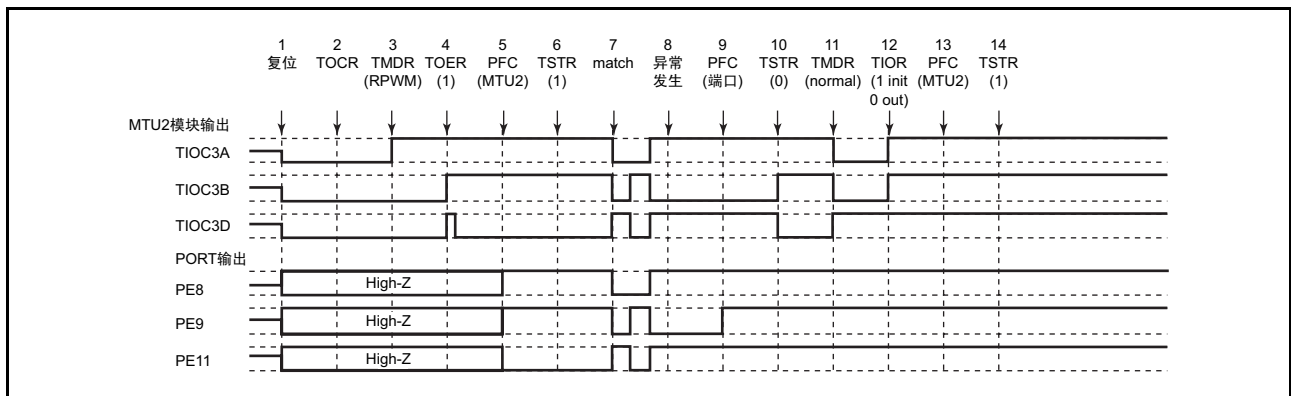


图 11.166 在复位同步 PWM 模式中发生异常并在正常模式中恢复

1. 在复位后，MTU2 的输出为低电平，端口为高阻抗状态。
2. 必须通过 TOCR 选择复位同步 PWM 的输出电平并允许或者禁止周期输出。
3. 设定复位同步 PWM。
4. 必须通过 TOER 允许通道 3、4 的输出。
5. 必须通过 PFC 设定为 MTU2 的输出。
6. 通过 TSTR 开始计数。
7. 通过产生比较匹配，输出复位同步 PWM 波形。
8. 发生了异常。
9. 必须通过 PFC 设定为端口输出，输出有效电平的反相信号。
10. 通过 TSTR 停止计数（MTU2 的输出为复位同步 PWM 输出的初始值）。
11. 必须设定正常模式（MTU2 输出的正相为低电平，反相为高电平）。
12. 必须通过 TIOR 初始化引脚。
13. 必须通过 PFC 设定为 MTU2 的输出。
14. 通过 TSTR 重新开始。

(27) 在复位同步 PWM 模式的运行中发生异常并在 PWM 模式 1 中重新开始时的运行

在复位同步 PWM 模式中发生异常，重新设定后在 PWM 模式 1 中重新开始的说明如图 11.167 所示。

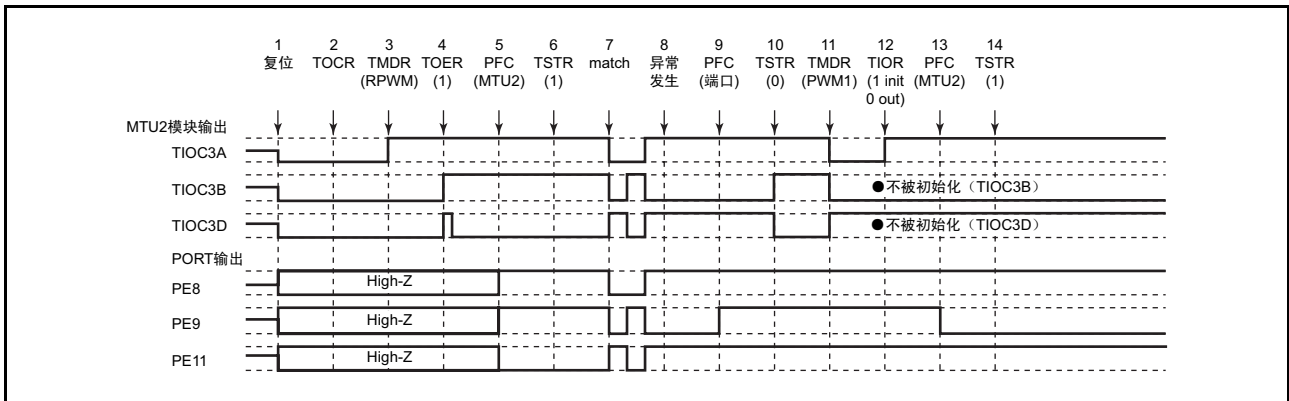


图 11.167 在复位同步 PWM 模式中发生异常并在 PWM 模式 1 中恢复

1 ~ 10 和图 11.166 通用。

11. 必须设定 PWM 模式 1 (MTU2 输出的正相为低电平, 反相为高电平)。

12. 通过 TIOR 不初始化引脚 (在 PWM 模式 1 中不初始化 TIOC\*B)。

13. 必须通过 PFC 设定为 MTU2 的输出。

14. 通过 TSTR 重新开始。

## (28) 在复位同时 PWM 模式的运行发生异常并在互补 PWM 模式中重新开始时的运行

在复位同步 PWM 模式中发生异常，重新设定后在互补 PWM 模式中重新开始时的说明如图 11.168 所示。

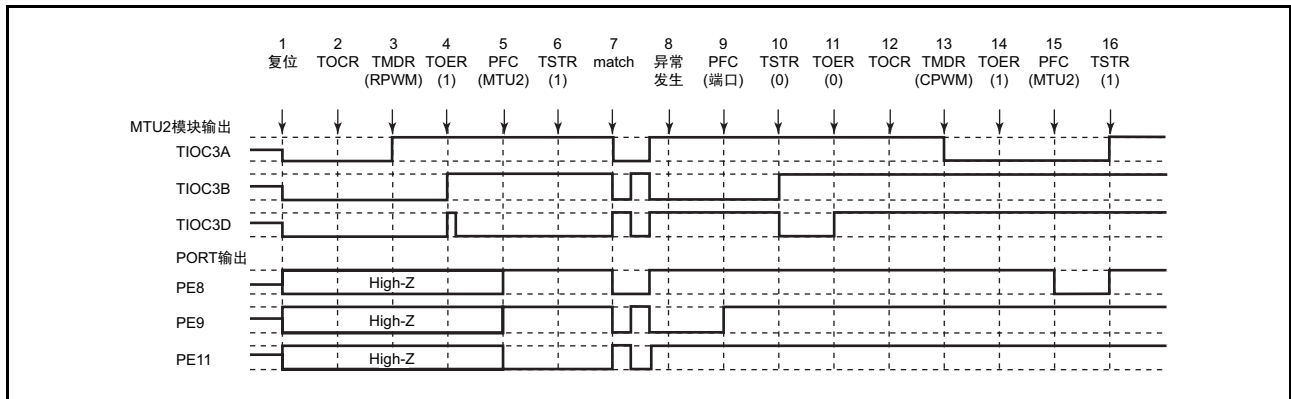


图 11.168 在复位同步 PWM 模式中发生异常并在互补 PWM 模式中恢复

1 ~ 10 和图 11.166 通用。

11. 必须通过 TOER 禁止通道 3、4 的输出。
12. 必须通过 TOCR 选择互补 PWM 的输出电平并允许或者禁周期输出。
13. 设定互补 PWM (MTU2 的周期输出引脚为低电平)。
14. 必须通过 TOER 允许通道 3、4 的输出。
15. 必须通过 PFC 设定为 MTU2 的输出。
16. 通过 TSTR 重新开始。

(29) 在复位同步 PWM 模式的运行中发生异常并在复位同步 PWM 模式中重新开始时的运行

在复位同步 PWM 模式中发生异常，重新设定后在复位同步 PWM 模式中重新开始时的说明如图 11.169 所示。

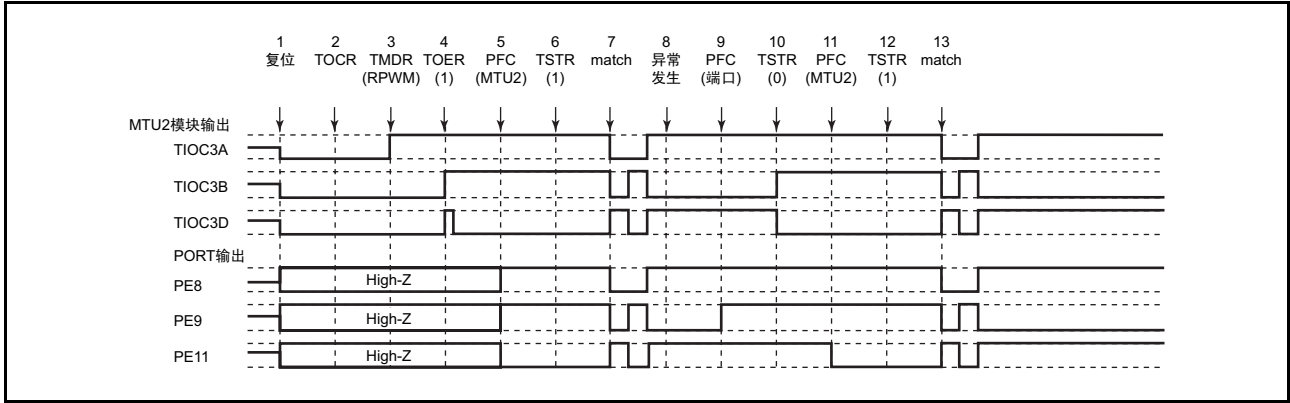


图 11.169 在复位同步 PWM 模式中发生异常并在复位同步 PWM 模式中恢复

1 ~ 10 和图 11.166 通用。

11. 必须通过 PFC 设定为 MTU2 的输出。

12. 通过 TSTR 重新开始。

13. 通过产生比较匹配，输出复位同步 PWM 波形。



## 第 12 章 多功能定时器脉冲单元 2S (MTU2S)

本 LSI 内置由 3 个通道 16 位定时器构成的多功能定时器脉冲单元 2S (MTU2S)，MTU2S 是内置 MTU2 的通道 3、4、5 的模块，详细内容请参照“第 11 章 多功能定时器脉冲单元 2 (MTU2)”。为了区别 MTU2，在输入/输出引脚名和寄存器名的末尾追加“S”，例如，TIOC3A 记为 TIOC3AS，TGRA\_3 记为 TGRA\_3S 等。

MTU2S 只在互补 PWM 模式的输出时才能以最大 80MHz 的工作频率运行，否则只能以最大 40MHz 的工作频率运行。

表 12.1 MTU2S 的功能一览表

项 目	通道 3	通道 4	通道 5
计数时钟	MIφ/1 MIφ/4 MIφ/16 MIφ/64 MIφ/256 MIφ/1024	MIφ/1 MIφ/4 MIφ/16 MIφ/64 MIφ/256 MIφ/1024	MIφ/1 MIφ/4 MIφ/16 MIφ/64
通用寄存器 (TGR)	TGRA_3S TGRB_3S	TGRA_4S TGRB_4S	TGRU_5S TGRV_5S TGRW_5S
通用寄存器 / 缓冲寄存器	TGRC_3S TGRD_3S	TGRC_4S TGRD_4S	—
输入 / 输出引脚	TIOC3AS TIOC3BS TIOC3CS TIOC3DS	TIOC4AS TIOC4BS TIOC4CS TIOC4DS	输入引脚 TIC5US TIC5VS TIC5WS
计数器的清除功能	TGR 的比较匹配 或者输入捕捉	TGR 的比较匹配 或者输入捕捉	TGR 的比较匹配 或者输入捕捉
比较匹配 的输出	0 输出	○	—
	1 输出	○	—
	交替输出	○	—
输入捕捉功能	○	○	○
同步运行	○	○	—
PWM 模式 1	○	○	—
PWM 模式 2	—	—	—
互补 PWM 模式	○	○	—
复位 PWM 模式	○	○	—
AC 同步马达驱动模式	—	—	—
相位计数模式	—	—	—
缓冲运行	○	○	—
用于空载时间补偿的计数器功能	—	—	○

项 目	通道 3	通道 4	通道 5
DTC 启动	TGR 的比较匹配 或者输入捕捉	TGR 的比较匹配或者输入捕 捉 TCNT 的上溢 / 下溢	TGR 的比较匹配 或者输入捕捉
A/D 转换的开始触发	TGRA_3S 的比较匹配或者输 入捕捉	TGRA_4S 的比较匹配或者输 入捕捉 在互补 PWM 模式中 TCNT_4S 的下溢 (波谷)	—
中断源	5 个源 • 比较匹配 / 输入捕捉 3AS • 比较匹配 / 输入捕捉 3BS • 比较匹配 / 输入捕捉 3CS • 比较匹配 / 输入捕捉 3DS • 上溢	5 个源 • 比较匹配 / 输入捕捉 4AS • 比较匹配 / 输入捕捉 4BS • 比较匹配 / 输入捕捉 4CS • 比较匹配 / 输入捕捉 4DS • 上溢 / 下溢	3 个源 • 比较匹配 / 输入捕捉 5US • 比较匹配 / 输入捕捉 5VS • 比较匹配 / 输入捕捉 5WS
A/D 转换开始请求 的延迟功能	—	• 在 TADCORA_4S 和 TCNT_4S 一致时, 进行 A/D 转换开始请求 • 在 TADCORB_4S 和 TCNT_4S 一致时, 进行 A/D 转换开始请求	—
跳过中断功能	• 跳过 TGRA_3S 的比较匹 配中断	• 跳过 TCIV_4S 中断	—

## 【符号说明】

○: 能

—: 不能

## 12.1 输入 / 输出引脚

表 12.2 引脚结构

通道	引脚名	输入 / 输出	功 能
3	TIOC3AS	输入 / 输出	TGRA_3S 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	TIOC3BS	输入 / 输出	TGRB_3S 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	TIOC3CS	输入 / 输出	TGRC_3S 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	TIOC3DS	输入 / 输出	TGRD_3S 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
4	TIOC4AS	输入 / 输出	TGRA_4S 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	TIOC4BS	输入 / 输出	TGRB_4S 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	TIOC4CS	输入 / 输出	TGRC_4S 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	TIOC4DS	输入 / 输出	TGRD_4S 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
5	TIC5US	输入	TGRU_5S 的输入捕捉的输入引脚 / 外部脉冲的输入引脚
	TIC5VS	输入	TGRV_5S 的输入捕捉的输入引脚 / 外部脉冲的输入引脚
	TIC5WS	输入	TGRW_5S 的输入捕捉的输入引脚 / 外部脉冲的输入引脚

## 12.2 寄存器说明

MTU2S 的各通道有以下寄存器，有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 27 章 寄存器一览”。如同通道 3 的 TCR 记为 TCR\_3S 一样，记载各通道的寄存器。

表 12.3 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
定时器的控制寄存器_3S	TCR_3S	R/W	H'00	H'FFFC600	8、16、32
定时器的控制寄存器_4S	TCR_4S	R/W	H'00	H'FFFC601	8
定时器的模式寄存器_3S	TMDR_3S	R/W	H'00	H'FFFC602	8、16
定时器的模式寄存器_4S	TMDR_4S	R/W	H'00	H'FFFC603	8
定时器的 I/O 控制寄存器 H_3S	TIORH_3S	R/W	H'00	H'FFFC604	8、16、32
定时器的 I/O 控制寄存器 L_3S	TIORL_3S	R/W	H'00	H'FFFC605	8
定时器的 I/O 控制寄存器 H_4S	TIORH_4S	R/W	H'00	H'FFFC606	8、16
定时器的 I/O 控制寄存器 L_4S	TIORL_4S	R/W	H'00	H'FFFC607	8
定时器的中断允许寄存器_3S	TIER_3S	R/W	H'00	H'FFFC608	8、16
定时器的中断允许寄存器_4S	TIER_4S	R/W	H'00	H'FFFC609	8
定时器的输出主控允许寄存器 S	TOERS	R/W	H'C0	H'FFFC60A	8
定时器的门控寄存器 S	TGCRS	R/W	H'80	H'FFFC60D	8
定时器的输出控制寄存器 1S	TOCR1S	R/W	H'00	H'FFFC60E	8、16
定时器的输出控制寄存器 2S	TOCR2S	R/W	H'00	H'FFFC60F	8
定时器的计数器_3S	TCNT_3S	R/W	H'0000	H'FFFC610	16、32
定时器的计数器_4S	TCNT_4S	R/W	H'0000	H'FFFC612	16
定时器的周期数据寄存器 S	TCDRS	R/W	H'FFFF	H'FFFC614	16、32
定时器的空载时间数据寄存器 S	TDDRS	R/W	H'FFFF	H'FFFC616	16
定时器的通用寄存器 A_3S	TGRA_3S	R/W	H'FFFF	H'FFFC618	16、32
定时器的通用寄存器 B_3S	TGRB_3S	R/W	H'FFFF	H'FFFC61A	16
定时器的通用寄存器 A_4S	TGRA_4S	R/W	H'FFFF	H'FFFC61C	16、32
定时器的通用寄存器 B_4S	TGRB_4S	R/W	H'FFFF	H'FFFC61E	16
定时器的子计数器 S	TCNTSS	R	H'0000	H'FFFC620	16、32
定时器的周期缓冲寄存器 S	TCBRS	R/W	H'FFFF	H'FFFC622	16
定时器的通用寄存器 C_3S	TGRC_3S	R/W	H'FFFF	H'FFFC624	16、32
定时器的通用寄存器 D_3S	TGRD_3S	R/W	H'FFFF	H'FFFC626	16
定时器的通用寄存器 C_4S	TGRC_4S	R/W	H'FFFF	H'FFFC628	16、32
定时器的通用寄存器 D_4S	TGRD_4S	R/W	H'FFFF	H'FFFC62A	16
定时器的状态寄存器_3S	TSR_3S	R/W	H'C0	H'FFFC62C	8、16
定时器的状态寄存器_4S	TSR_4S	R/W	H'C0	H'FFFC62D	8
定时器的跳过中断设定寄存器 S	TITCRS	R/W	H'00	H'FFFC630	8、16
定时器的跳过中断次数计数器 S	TITCNTS	R	H'00	H'FFFC631	8

寄存器名	略称	R/W	初始值	地址	存取长度
定时器的缓冲传送设定寄存器 S	TBTERS	R/W	H'00	H'FFFC632	8
定时器的空载时间允许寄存器 S	TDERS	R/W	H'01	H'FFFC634	8
定时器的输出电平缓冲寄存器 S	TOLBRS	R/W	H'00	H'FFFC636	8
定时器的缓冲运行传送模式寄存器 _3S	TBTM_3S	R/W	H'00	H'FFFC638	8、16
定时器的缓冲运行传送模式寄存器 _4S	TBTM_4S	R/W	H'00	H'FFFC639	8
定时器的 A/D 转换开始请求的控制寄存器 S	TADCRS	R/W	H'0000	H'FFFC640	16
定时器的 A/D 转换开始请求的周期设定寄存器 A_4S	TADCORA_4S	R/W	H'FFFF	H'FFFC644	16、32
定时器的 A/D 转换开始请求的周期设定寄存器 B_4S	TADCORB_4S	R/W	H'FFFF	H'FFFC646	16
定时器的 A/D 转换开始请求的周期设定缓冲寄存器 A_4S	TADCOBRA_4S	R/W	H'FFFF	H'FFFC648	16、32
定时器的 A/D 转换开始请求的周期设定缓冲寄存器 B_4S	TADCOBRB_4S	R/W	H'FFFF	H'FFFC64A	16
定时器的同步清除寄存器 S	TSYCRS	R/W	H'00	H'FFFC650	8
定时器的波形控制寄存器 S	TWCRS	R/W	H'00	H'FFFC660	8
定时器的开始寄存器 S	TSTRS	R/W	H'00	H'FFFC680	8、16
定时器的同步寄存器 S	TSYRS	R/W	H'00	H'FFFC681	8
定时器的读写允许寄存器 S	TRWERS	R/W	H'01	H'FFFC684	8
定时器的计数器 U_5S	TCNTU_5S	R/W	H'0000	H'FFFC880	16、32
定时器的通用寄存器 U_5S	TGRU_5S	R/W	H'FFFF	H'FFFC882	16
定时器的控制寄存器 U_5S	TCRU_5S	R/W	H'00	H'FFFC884	8
定时器的 I/O 控制寄存器 U_5S	TIORU_5S	R/W	H'00	H'FFFC886	8
定时器的计数器 V_5S	TCNTV_5S	R/W	H'0000	H'FFFC890	16、32
定时器的通用寄存器 V_5S	TGRV_5S	R/W	H'FFFF	H'FFFC892	16
定时器的控制寄存器 V_5S	TCRV_5S	R/W	H'00	H'FFFC894	8
定时器的 I/O 控制寄存器 V_5S	TIORV_5S	R/W	H'00	H'FFFC896	8
定时器的计数器 W_5S	TCNTW_5S	R/W	H'0000	H'FFFC8A0	16、32
定时器的通用寄存器 W_5S	TGRW_5S	R/W	H'FFFF	H'FFFC8A2	16
定时器的控制寄存器 W_5S	TCRW_5S	R/W	H'00	H'FFFC8A4	8
定时器的 I/O 控制寄存器 W_5S	TIORW_5S	R/W	H'00	H'FFFC8A6	8
定时器的状态寄存器 _5S	TSR_5S	R/W	H'00	H'FFFC8B0	8
定时器的中断允许寄存器 _5S	TIER_5S	R/W	H'00	H'FFFC8B2	8
定时器的开始寄存器 _5S	TSTR_5S	R/W	H'00	H'FFFC8B4	8
定时器的比较匹配清除寄存器 S	TCNTCMPCLRS	R/W	H'00	H'FFFC8B6	8

## 第 13 章 端口输出的允许 (POE)

根据  $\overline{POE0} \sim \overline{POE8}$  引脚的输入变化、大电流引脚 (MTU2 的 TIOC3B、TIOC3D、TIOC4A、TIOC4B、TIOC4C、TIOC4D 以及 MTU2S 的 TIOC3BS、TIOC3DS、TIOC4AS、TIOC4BS、TIOC4CS、TIOC4DS 为多路复用引脚) 的输出状态或者寄存器的设定, 端口输出的允许 (POE) 能将大电流引脚和 MTU2 的 CH0 引脚 (TIOC0A、TIOC0B、TIOC0C、TIOC0D 为多路复用引脚) 置为高阻抗状态, 同时能产生中断请求。

### 13.1 特点

- 能给  $\overline{POE0} \sim \overline{POE8}$  的各输入引脚设定下降沿、 $P\phi/8 \times 16$  次、 $P\phi/16 \times 16$  次或者  $P\phi/128 \times 16$  次的低电平采样。
- 能通过  $\overline{POE0} \sim \overline{POE8}$  引脚的下降沿或者低电平采样, 将大电流引脚和 MTU2 的 CH0 引脚置为高阻抗状态。
- 如果在和大电流引脚的输出电平进行比较的同时, 有效电平持续输出了 1 个周期以上, 就能将大电流引脚置为高阻抗状态。
- 能通过写 POE 的寄存器将大电流引脚和 MTU2 的 CH0 引脚置为高阻抗状态。
- 可分别通过输入电平的采样或者输出电平的比较结果产生中断。

如图 13.1 的框图所示, POE 由输入电平的检测电路、输出电平的比较电路和高阻抗请求 / 中断请求的生成电路构成。

不同的是: 即使在振荡器停止或者软件待机状态下, 也能将大电流引脚置为高阻抗状态。详细内容请参照“21.1.11 大电流端口的控制寄存器 (HCPCR)”和“附录 A. 引脚状态”。

POE 的框图如图 13.1 所示。

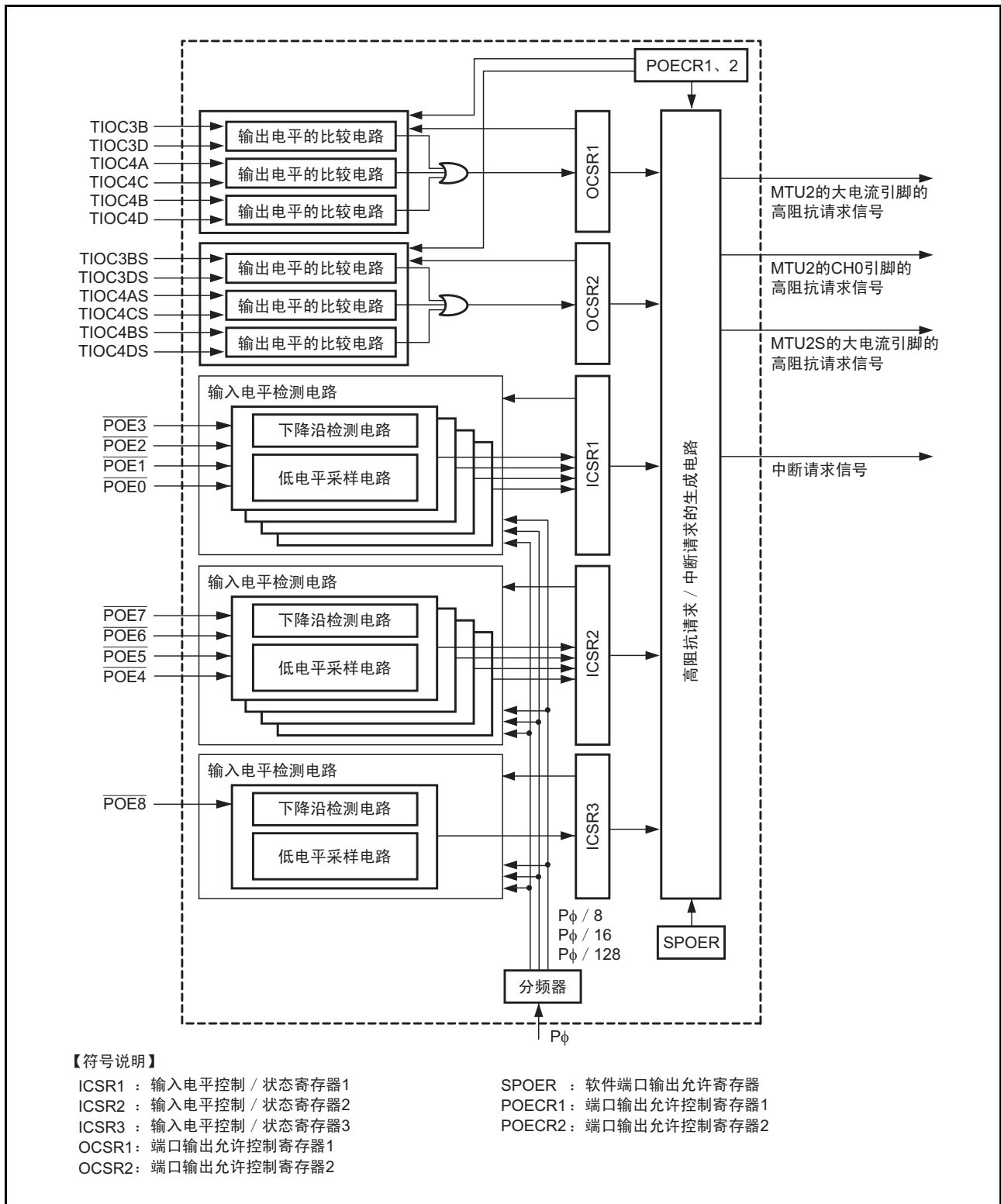


图 13.1 POE 的框图

## 13.2 输入 / 输出引脚

表 13.1 引脚结构

名称	引脚名	输入 / 输出	功 能
端口输出允许的 输入引脚 0 ~ 3	$\overline{\text{POE0}} \sim \overline{\text{POE3}}$	输入	输入将 MTU2 的大电流引脚置为高阻抗状态的请求信号
端口输出允许的 输入引脚 4 ~ 7	$\overline{\text{POE4}} \sim \overline{\text{POE7}}$	输入	输入将 MTU2S 的大电流引脚置为高阻抗状态的请求信号
端口输出允许的 输入引脚 8	$\overline{\text{POE8}}$	输入	输入将 MTU2 的 CH0 引脚置为高阻抗状态的请求信号

通过表 13.2 所示的引脚组合进行输出电平的比较。

表 13.2 引脚组合

引脚组合	输入 / 输出	功 能
PE9/TIOC3B 和 PE11/TIOC3D	输出	当 2 个引脚的有效电平 (MTU2 定时器的输出控制寄存器 (TOCR) 的输出电平选择 P 位 (OLSP) 是 0 时为低电平输出, 是 1 时为高电平输出) 持续输出了 1 个外围时钟 ( $P\phi$ ) 周期以上时, 将 MTU2 的大电流引脚置为高阻抗状态。当通过设定引脚功能控制器选择 MTU2 的输出功能或者通用输出功能时, 进行上述有效电平的比较, 否则不进行比较。 能通过 POE 的寄存器设定对哪个组合进行输出比较以及是否进行高阻抗控制。
PE12/TIOC4A 和 PE14/TIOC4C	输出	
PE13/TIOC4B 和 PE15/TIOC4D	输出	
PD9/TIOC3BS 和 PD11/TIOC3DS	输出	当 2 个引脚的有效电平 (MTU2S 定时器的输出控制寄存器 (TOCR) 的输出电平选择 P 位 (OLSP) 是 0 时为低电平输出, 是 1 时为高电平输出) 持续输出了 1 个外围时钟 ( $P\phi$ ) 周期以上时, 将 MTU2S 的大电流引脚置为高阻抗状态。当通过设定引脚功能控制器选择 MTU2S 的输出功能或者通用输出功能时, 进行上述有效电平的比较, 否则不进行比较。 能通过 POE 的寄存器设定对哪个组合进行输出比较以及是否进行高阻抗控制。
PD12/TIOC4AS 和 PD14/TIOC4CS	输出	
PD13/TIOC4BS 和 PD15/TIOC4DS	输出	
PD29/TIOC3BS 和 PD28/TIOC3DS	输出	
PD27/TIOC4AS 和 PD25/TIOC4CS	输出	
PD26/TIOC4BS 和 PD24/TIOC4DS	输出	
PE16/TIOC3BS 和 PE17/TIOC3DS	输出	
PE18/TIOC4AS 和 PE20/TIOC4CS	输出	
PE19/TIOC4BS 和 PE21/TIOC4DS	输出	

### 13.3 寄存器说明

POE 有以下寄存器。有关这些寄存器的地址和各处理模式中的寄存器状态，请参照“第 27 章 寄存器一览”。

表 13.3 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
输入电平控制 / 状态寄存器 1	ICSR1	R/W	H'0000	H'FFFFD000	8、16、32
输出电平控制 / 状态寄存器 1	OCSR1	R/W	H'0000	H'FFFFD002	8、16
输入电平控制 / 状态寄存器 2	ICSR2	R/W	H'0000	H'FFFFD004	8、16、32
输出电平控制 / 状态寄存器 2	OCSR2	R/W	H'0000	H'FFFFD006	8、16
输入电平控制 / 状态寄存器 3	ICSR3	R/W	H'0000	H'FFFFD008	8、16
软件端口输出允许寄存器	SPOER	R/W	H'00	H'FFFFD00A	8
端口输出允许控制寄存器 1	POECR1	R/W	H'00	H'FFFFD00B	8
端口输出允许控制寄存器 2	POECR2	R/W	H'7700	H'FFFFD00C	8、16

#### 13.3.1 输入电平控制 / 状态寄存器 1 (ICSR1)

ICSR1 是 16 位可读写寄存器，选择  $\overline{\text{POE0}} \sim \overline{\text{POE3}}$  引脚的输入模式、控制中断的允许 / 禁止以及表示各状态。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	POE3F	POE2F	POE1F	POE0F	-	-	-	PIE1	POE3M[1:0]	POE2M[1:0]	POE1M[1:0]	POE0M[1:0]				
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R	R	R	R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

【注】\*1 为了清除标志，只能在读1后写0。

\*2 只能在上电复位后写1次。

位	位名	初始值	R/W	说明
15	POE3F	0	R/(W)*1	<p>POE3 标志位</p> <p>表示给 POE3 引脚输入了高阻抗请求。</p> <p>[清除条件]</p> <ul style="list-style-type: none"> <li>当读 POE3F=1 的状态后，给 POE3F 写 0 时（当通过 ICSR1 的 bit7、6 设定为下降沿时）</li> <li>在通过 P<math>\phi</math>/8、16、128 时钟对 POE3 输入引脚进行高电平采样时，当读 POE3F=1 的状态后，给 POE3F 写 0 时（当通过 ICSR1 的 bit7、6 设定为低电平采样时）</li> </ul> <p>[置位条件]</p> <ul style="list-style-type: none"> <li>当 POE3 引脚有 ICSR1 的 bit7、6 设定的输入时</li> </ul>



位	位名	初始值	R/W	说 明
14	POE2F	0	R/(W)*1	<p>POE2 标志位 表示给 <math>\overline{\text{POE2}}</math> 引脚输入了高阻抗请求。</p> <p>[清除条件]</p> <ul style="list-style-type: none"> <li>当读 POE2F=1 的状态后, 给 POE2F 写 0 时 (当通过 ICSR1 的 bit5、4 设定为下降沿时)</li> <li>在通过 P<math>\phi</math>/8、16、128 时钟对 POE2 输入引脚进行高电平采样时, 当读 POE2F=1 的状态后, 给 POE2F 写 0 时 (当通过 ICSR1 的 bit5、4 设定为低电平采样时)</li> </ul> <p>[置位条件]</p> <ul style="list-style-type: none"> <li>当 <math>\overline{\text{POE2}}</math> 引脚有 ICSR1 的 bit5、4 设定的输入时</li> </ul>
13	POE1F	0	R/(W)*1	<p>POE1 标志位 表示给 <math>\overline{\text{POE1}}</math> 引脚输入了高阻抗请求。</p> <p>[清除条件]</p> <ul style="list-style-type: none"> <li>当读 POE1F=1 的状态后, 给 POE1F 写 0 时 (当通过 ICSR1 的 bit3、2 设定为下降沿时)</li> <li>在通过 P<math>\phi</math>/8、16、128 时钟对 POE1 输入引脚进行高电平采样时, 当读 POE1F=1 的状态后, 给 POE1F 写 0 时 (当通过 ICSR1 的 bit3、2 设定为低电平采样时)</li> </ul> <p>[置位条件]</p> <ul style="list-style-type: none"> <li>当 <math>\overline{\text{POE1}}</math> 引脚有 ICSR1 的 bit3、2 设定的输入时</li> </ul>
12	POE0F	0	R/(W)*1	<p>POE0 标志位 表示给 <math>\overline{\text{POE0}}</math> 引脚输入了高阻抗请求。</p> <p>[清除条件]</p> <ul style="list-style-type: none"> <li>当读 POE0F=1 的状态后, 给 POE0F 写 0 时 (当通过 ICSR1 的 bit1、0 设定为下降沿时)</li> <li>在通过 P<math>\phi</math>/8、16、128 时钟对 POE0 输入引脚进行高电平采样时, 当读 POE0F=1 的状态后, 给 POE0F 写 0 时 (当通过 ICSR1 的 bit1、0 设定为低电平采样时)</li> </ul> <p>[置位条件]</p> <ul style="list-style-type: none"> <li>当 <math>\overline{\text{POE0}}</math> 引脚有 ICSR1 的 bit1、0 设定的输入时</li> </ul>
11 ~ 9	—	全 0	R	保留位 读写的值总是 0。
8	PIE1	0	R/W	<p>端口中断允许位 1 指定在 ICSR1 的 POE0F ~ POE3F 位中有 1 位被置 1 时是否请求中断。</p> <p>0: 禁止中断请求 1: 允许中断请求</p>
7、6	POE3M[1:0]	00	R/W*2	<p>POE3 模式位 1、0 选择 <math>\overline{\text{POE3}}</math> 引脚的输入模式。</p> <p>00: 在 POE3 输入引脚的下降沿接受请求。 01: 通过 P<math>\phi</math>/8 时钟对 POE3 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。 10: 通过 P<math>\phi</math>/16 时钟对 POE3 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。 11: 通过 P<math>\phi</math>/128 时钟对 POE3 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。</p>

位	位名	初始值	R/W	说 明
5、4	POE2M[1:0]	00	R/W*2	<p>POE2 模式位 1、0 选择 POE2 引脚的输入模式。</p> <p>00: 在 POE2 输入引脚的下降沿接受请求。</p> <p>01: 通过 <math>P\phi/8</math> 时钟对 POE2 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。</p> <p>10: 通过 <math>P\phi/16</math> 时钟对 POE2 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。</p> <p>11: 通过 <math>P\phi/128</math> 时钟对 POE2 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。</p>
3、2	POE1M[1:0]	00	R/W*2	<p>POE1 模式位 1、0 选择 POE1 引脚的输入模式。</p> <p>00: 在 POE1 输入引脚的下降沿接受请求。</p> <p>01: 通过 <math>P\phi/8</math> 时钟对 POE1 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。</p> <p>10: 通过 <math>P\phi/16</math> 时钟对 POE1 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。</p> <p>11: 通过 <math>P\phi/128</math> 时钟对 POE1 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。</p>
1、0	POE0M[1:0]	00	R/W*2	<p>POE0 模式位 1、0 选择 POE0 引脚的输入模式。</p> <p>00: 在 POE0 输入引脚的下降沿接受请求。</p> <p>01: 通过 <math>P\phi/8</math> 时钟对 POE0 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。</p> <p>10: 通过 <math>P\phi/16</math> 时钟对 POE0 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。</p> <p>11: 通过 <math>P\phi/128</math> 时钟对 POE0 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。</p>

【注】 \*1 为了清除标志, 只能在读 1 后写 0

\*2 在上电复位后, 只能写 1 次。

## 13.3.2 输出电平控制 / 状态寄存器 1 (OCSR1)

OCSR1 是 16 位可读写寄存器，允许或者禁止输出电平的比较、控制中断的允许 / 禁止以及表示各状态。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSF1	-	-	-	-	-	OCE1	OIE1	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R	R	R	R	R	R/W*2	R/W	R	R	R	R	R	R	R	R

【注】\*1 为了清除标志，只能在读1后写0。

\*2 只能在上电复位后写1次。

位	位名	初始值	R/W	说 明
15	OSF1	0	R/(W)*1	输出短路的标志位 1 表示在 MTU2 要比较的 3 组 2 相输出中至少有 1 组同时为有效电平。 [清除条件] • 当读 OSF1=1 的状态后，给 OSF1 写 0 时 [置位条件] • 当 3 组 2 相输出中至少有 1 组同时为有效电平时
14 ~ 10	—	全 0	R	保留位 读写的值总是 0。
9	OCE1	0	R/W*2	输出短路的高阻抗允许位 1 设定在 OCSR1 的 OSF1 位被置位时是否将引脚置为高阻抗状态。 0: 不将引脚置为高阻抗状态。 1: 将引脚置为高阻抗状态。
8	OIE1	0	R/W	输出短路的中断允许位 1 指定在 OCSR1 的 OSF1 位被置位时是否请求中断。 0: 禁止中断请求 1: 允许中断请求
7 ~ 0	—	全 0	R	保留位 读写的值总是 0。

## 13.3.3 输入电平控制 / 状态寄存器 2 (ICSR2)

ICSR2 是 16 位可读写寄存器，选择  $\overline{\text{POE4}} \sim \overline{\text{POE7}}$  引脚的输入模式、控制中断的允许 / 禁止以及表示各状态。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	POE7F	POE6F	POE5F	POE4F	-	-	-	PIE2	POE7M[1:0]	POE6M[1:0]	POE5M[1:0]	POE4M[1:0]				
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R	R	R	R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

【注】\*1 为了清除标志，只能在读1后写0。

\*2 只能在上电复位后写1次。

位	位名	初始值	R/W	说 明
15	POE7F	0	R/(W)*1	<p>POE7 标志位</p> <p>表示给 <math>\overline{\text{POE7}}</math> 引脚输入了高阻抗请求。</p> <p>[清除条件]</p> <ul style="list-style-type: none"> <li>当读 POE7F=1 的状态后，给 POE7F 写 0 时（当通过 ICSR2 的 bit7、6 设定为下降沿时）</li> <li>在通过 P<math>\phi</math>/8、16、128 时钟对 POE7 输入引脚进行高电平采样时，当读 POE7F=1 的状态后，给 POE7F 写 0 时（当通过 ICSR2 的 bit7、6 设定为低电平采样时）</li> </ul> <p>[置位条件]</p> <ul style="list-style-type: none"> <li>当 <math>\overline{\text{POE7}}</math> 引脚有 ICSR2 的 bit7、6 设定的输入时</li> </ul>
14	POE6F	0	R/(W)*1	<p>POE6 标志位</p> <p>表示给 <math>\overline{\text{POE6}}</math> 引脚输入了高阻抗请求。</p> <p>[清除条件]</p> <ul style="list-style-type: none"> <li>当读 POE6F=1 的状态后，给 POE6F 写 0 时（当通过 ICSR2 的 bit5、4 设定为下降沿时）</li> <li>在通过 P<math>\phi</math>/8、16、128 时钟对 POE6 输入引脚进行高电平采样时，当读 POE6F=1 的状态后，给 POE6F 写 0 时（当通过 ICSR2 的 bit5、4 设定为低电平采样时）</li> </ul> <p>[置位条件]</p> <ul style="list-style-type: none"> <li>当 <math>\overline{\text{POE6}}</math> 引脚有 ICSR2 的 bit5、4 设定的输入时</li> </ul>
13	POE5F	0	R/(W)*1	<p>POE5 标志位</p> <p>表示给 <math>\overline{\text{POE5}}</math> 引脚输入了高阻抗请求。</p> <p>[清除条件]</p> <ul style="list-style-type: none"> <li>当读 POE5F=1 的状态后，给 POE5F 写 0 时（当通过 ICSR2 的 bit3、2 设定为下降沿时）</li> <li>在通过 P<math>\phi</math>/8、16、128 时钟对 POE5 输入引脚进行高电平采样时，当读 POE5F=1 的状态后，给 POE5F 写 0 时（当通过 ICSR2 的 bit3、2 设定为低电平采样时）</li> </ul> <p>[置位条件]</p> <ul style="list-style-type: none"> <li>当 <math>\overline{\text{POE5}}</math> 引脚有 ICSR2 的 bit3、2 设定的输入时</li> </ul>

位	位名	初始值	R/W	说 明
12	POE4F	0	R/(W)*1	<p>POE4 标志位 表示给 <math>\overline{\text{POE4}}</math> 引脚输入了高阻抗请求。</p> <p>[清除条件]</p> <ul style="list-style-type: none"> <li>当读 POE4F=1 的状态后, 给 POE4F 写 0 时 (当通过 ICSR2 的 bit1、0 设定为下降沿时)</li> <li>在通过 P<math>\phi</math>/8、16、128 时钟对 POE4 输入引脚进行高电平采样时, 当读 POE4F=1 的状态后, 给 POE4F 写 0 时 (当通过 ICSR2 的 bit1、0 设定为低电平采样时)</li> </ul> <p>[置位条件]</p> <ul style="list-style-type: none"> <li>当 <math>\overline{\text{POE4}}</math> 引脚有 ICSR2 的 bit1、0 设定的输入时</li> </ul>
11 ~ 9	—	全 0	R	<p>保留位 读写的值总是 0。</p>
8	PIE1	0	R/W	<p>端口中断允许位 1 指定在 ICSR2 的 POE4F ~ POE7F 位中有 1 位被置 1 时是否请求中断。</p> <p>0: 禁止中断请求 1: 允许中断请求</p>
7、6	POE7M[1:0]	00	R/W*2	<p>POE7 模式位 1、0 选择 <math>\overline{\text{POE7}}</math> 引脚的输入模式。</p> <p>00: 在 POE7 输入引脚的下降沿接受请求。 01: 通过 P<math>\phi</math>/8 时钟对 POE7 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。 10: 通过 P<math>\phi</math>/16 时钟对 POE7 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。 11: 通过 P<math>\phi</math>/128 时钟对 POE7 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。</p>
5、4	POE6M[1:0]	00	R/W*2	<p>POE6 模式位 1、0 选择 <math>\overline{\text{POE6}}</math> 引脚的输入模式。</p> <p>00: 在 POE6 输入引脚的下降沿接受请求。 01: 通过 P<math>\phi</math>/8 时钟对 POE6 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。 10: 通过 P<math>\phi</math>/16 时钟对 POE6 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。 11: 通过 P<math>\phi</math>/128 时钟对 POE6 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。</p>
3、2	POE5M[1:0]	00	R/W*2	<p>POE5 模式位 1、0 选择 <math>\overline{\text{POE5}}</math> 引脚的输入模式。</p> <p>00: 在 POE5 输入引脚的下降沿接受请求。 01: 通过 P<math>\phi</math>/8 时钟对 POE5 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。 10: 通过 P<math>\phi</math>/16 时钟对 POE5 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。 11: 通过 P<math>\phi</math>/128 时钟对 POE5 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。</p>

位	位名	初始值	R/W	说 明
1、0	POE4M[1:0]	00	R/W*2	POE4 模式位 1、0 选择 POE4 引脚的输入模式。 00: 在 POE4 输入引脚的下降沿接受请求。 01: 通过 P $\phi$ /8 时钟对 POE4 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。 10: 通过 P $\phi$ /16 时钟对 POE4 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。 11: 通过 P $\phi$ /128 时钟对 POE4 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。

【注】 \*1 为了清除标志, 只能在读 1 后写 0

\*2 在上电复位后, 只能写 1 次。

### 13.3.4 输出电平控制 / 状态寄存器 2 (OCSR2)

OCSR2 是 16 位可读写寄存器, 允许或者禁止输出电平的比较、控制中断的允许 / 禁止以及表示各状态。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSF2	-	-	-	-	-	OCE2	OIE2	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R	R	R	R	R	R/W*2	R/W	R	R	R	R	R	R	R	R

【注】 \*1 为了清除标志, 只能在读 1 后写 0。

\*2 只能在上电复位后写 1 次。

位	位名	初始值	R/W	说 明
15	OSF2	0	R/(W)*1	输出短路的标志位 2 表示在 MTU2S 要比较的 3 组 2 相输出中至少有 1 组同时为有效电平。 [清除条件] • 当读 OSF2=1 的状态后, 给 OSF2 写 0 时 [置位条件] • 当 3 组 2 相输出中至少有 1 组同时为有效电平时
14 ~ 10	—	全 0	R	保留位 读写的值总是 0。
9	OCE2	0	R/W*2	输出短路的高阻抗允许位 2 设定在 OCSR2 的 OSF2 位被置位时是否将引脚置为高阻抗状态。 0: 不将引脚置为高阻抗状态。 1: 将引脚置为高阻抗状态。
8	OIE2	0	R/W	输出短路的中断允许位 2 指定在 OCSR2 的 OSF2 位被置位时是否请求中断。 0: 禁止中断请求 1: 允许中断请求
7 ~ 0	—	全 0	R	保留位 读写的值总是 0。

## 13.3.5 输入电平控制 / 状态寄存器 3 (ICSR3)

ICSR3 是 16 位可读写寄存器，选择 POE8 引脚的输入模式、控制中断的允许 / 禁止以及表示各状态。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	POE8F	-	-	POE8E	PIE3	-	-	-	-	-	-	-	POE8M[1:0]
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/(W)*1	R	R	R/W*2	R/W	R	R	R	R	R	R	R	R/W*2

【注】\*1 为了清除标志，只能在读1后写0。

\*2 只能在上电复位后写1次。

位	位名	初始值	R/W	说明
15 ~ 13	—	全 0	R	保留位 读写的值总是 0。
12	POE8F	0	R/(W)*1	POE8 的标志位 表示给 POE8 引脚输入了高阻抗请求。 [清除条件] • 当读 POE8F=1 的状态后，给 POE8F 写 0 时（当通过 ICSR3 的 bit1、0 设定为下降沿时） • 在通过 P $\phi$ /8、16、128 时钟对 POE8 输入引脚进行高电平采样时，当读 POE8F=1 的状态后，给 POE8F 写 0 时（当通过 ICSR3 的 bit1、0 设定为低电平采样时） [置位条件] • 当 POE8 引脚有 ICSR3 的 bit1、0 设定的输入时
11、10	—	全 0	R	保留位 读写的值总是为 0。
9	POE8E	0	R/W*2	POE8 的高阻抗允许位 设定在 ICSR3 的 POE8F 位被置位时是否将引脚置为高阻抗状态。 0: 不将引脚置为高阻抗状态。 1: 将引脚置为高阻抗状态。
8	PIE3	0	R/W	端口的中断允许位 3 指定在 ICSR3 的 POE8 位被置 1 时是否请求中断。 0: 禁止中断请求 1: 允许中断请求
7 ~ 2	—	全 0	R	保留位 读写的值总是 0。
1、0	POE8M[1:0]	00	R/W*2	POE8 模式位 1、0 选择 POE8 引脚的输入模式。 00: 在 POE8 输入引脚的下降沿接受请求。 01: 通过 P $\phi$ /8 时钟对 POE8 输入引脚进行 16 次低电平采样，当全部为低电平时，接受请求。 10: 通过 P $\phi$ /16 时钟对 POE8 输入引脚进行 16 次低电平采样，当全部为低电平时，接受请求。 11: 通过 P $\phi$ /128 时钟对 POE8 输入引脚进行 16 次低电平采样，当全部为低电平时，接受请求。

【注】\*1 为了清除标志，只能在读 1 后写 0。

\*2 在上电复位后，只能写 1 次。

## 13.3.6 软件端口的输出允许寄存器 (SPOER)

SPOER 是 8 位可读写寄存器，对引脚进行高阻抗控制。

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	MTU2S HIZ	MTU2 CH0HIZ	MTU2 CH34HIZ
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7 ~ 3	—	全 0	R	保留位 读写的值总是 0。
2	MTU2SHIZ	0	R/W	MTU2S 的高阻抗输出 对 MTU2S 的大电流引脚进行高阻抗控制。 0: 不置为高阻抗状态。 [清除条件] • 上电复位 • 当读 MTU2SHIZ=1 的状态后, 给 MTU2SHIZ 写 0 时 1: 置为高阻抗状态。 [置位条件] • 当给 MTU2SHIZ 写 1 时
1	MTU2CH0HIZ	0	R/W	MTU2 的 CH0 高阻抗输出 对 MTU2 的 CH0 引脚进行高阻抗控制。 0: 不置为高阻抗状态。 [清除条件] • 上电复位 • 当读 MTU2CH0HIZ=1 的状态后, 给 MTU2CH0HIZ 写 0 时 1: 置为高阻抗状态。 [置位条件] • 当给 MTU2CH0HIZ 写 1 时
0	MTU2CH34HIZ	0	R/W	MTU2 的 CH3、4 高阻抗输出 对 MTU2 的大电流引脚进行高阻抗控制。 0: 不置为高阻抗状态 [清除条件] • 上电复位 • 当读 MTU2CH34HIZ=1 的状态后, 给 MTU2CH34HIZ 写 0 时 1: 置为高阻抗状态。 [置位条件] • 当给 MTU2CH34HIZ 写 1 时



## 13.3.7 端口的输出允许控制寄存器 1 (POECR1)

POECR1 是 8 位可读写寄存器，对引脚进行高阻抗控制。

位:	7	6	5	4	3	2	1	0
	-	-	-	-	MTU2 PE3ZE	MTU2 PE2ZE	MTU2 PE1ZE	MTU2 PE0ZE
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W*	R/W*	R/W*	R/W*

【注】\* 只能在上电复位后写1次。

位	位名	初始值	R/W	说 明
7 ~ 4	—	全 0	R	保留位 读写的值总是 0。
3	MTU2PE3ZE	0	R/W*	MTU2 PE3 的高阻抗允许位 设定在 POE8F 和 MTU2CH0HIZ 位中有 1 位被置位时是否将 MTU2 的 CH0 引脚 PE3/TIOC0D 置为高阻抗状态。 0: 不置为非高阻抗状态。 1: 置为高阻抗状态。
2	MTU2PE2ZE	0	R/W*	MTU2 PE2 的高阻抗允许位 设定在 POE8F 和 MTU2CH0HIZ 位中有 1 位被置位时是否将 MTU2 的 CH0 引脚 PE2/TIOC0C 置为高阻抗状态。 0: 不置为非高阻抗状态。 1: 置为高阻抗状态。
1	MTU2PE1ZE	0	R/W*	MTU2 PE1 的高阻抗允许位 设定在 POE8F 和 MTU2CH0HIZ 位中有 1 位被置位时是否将 MTU2 的 CH0 引脚 PE1/TIOC0B 置为高阻抗状态。 0: 不置为非高阻抗状态。 1: 置为高阻抗状态。
0	MTU2PE0ZE	0	R/W*	MTU2 PE0 的高阻抗允许位 设定在 POE8F 和 MTU2CH0HIZ 位中有 1 位被置位时是否将 MTU2 的 CH0 引脚 PE0/TIOC0A 置为高阻抗状态。 0: 不置为非高阻抗状态。 1: 置为高阻抗状态。

【注】\* 在上电复位后，只能写 1 次。

## 13.3.8 端口的输出允许控制寄存器 2 (POECR2)

POECR2 是 16 位可读写寄存器，对引脚进行高阻抗控制。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	MTU2 P1CZE	MTU2 P2CZE	MTU2 P3CZE	-	MTU2S P1CZE	MTU2S P2CZE	MTU2S P3CZE	-	MTU2S P4CZE	MTU2S P5CZE	MTU2S P5CZE	-	MTU2S P7CZE	MTU2S P8CZE	MTU2S P9CZE
初始值:	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0
R/W:	R	R/W*	R/W*	R/W*	R	R/W*	R/W*	R/W*	R	R/W*	R/W*	R/W*	R	R/W*	R/W*	R/W*

【注】\* 只能在上电复位后写1次。

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	MTU2P1CZE	1	R/W*	MTU2 端口 1 的输出比较 / 高阻抗允许位 设定是否对 MTU2 的大电流引脚 PE9/TIOC3B 和 PE11/TIOC3D 进行输出电平比较，并且设定在 OCE1 位是 1 时 OSF1 位被置位的情况下或者在 POE0F、POE1F、POE2F、POE3F 和 TU2CH34HIZ 位中有 1 位被置位的情况下，是否将这些引脚置为高阻抗状态。 0: 不进行输出电平的比较并且不置为高阻抗状态。 1: 进行输出电平的比较并且置为高阻抗状态。
13	MTU2P2CZE	1	R/W*	MTU2 端口 2 的输出比较 / 高阻抗允许位 设定是否对 MTU2 的大电流引脚 PE12/TIOC4A 和 PE14/TIOC4C 进行输出电平比较，并且设定在 OCE1 位是 1 时 OSF1 位被置位的情况下或者在 POE0F、POE1F、POE2F、POE3F 和 MTU2CH34HIZ 位中有 1 位被置位的情况下，是否将这些引脚置为高阻抗状态。 0: 不进行输出电平的比较并且不置为高阻抗状态。 1: 进行输出电平的比较并且置为高阻抗状态。
12	MTU2P3CZE	1	R/W*	MTU2 端口 3 的输出比较 / 高阻抗允许位 设定是否对 MTU2 的大电流引脚 PE13/TIOC4B 和 PE15/TIOC4D 进行输出电平比较，并且设定在 OCE1 位是 1 时 OSF1 位被置位的情况下或者在 POE0F、POE1F、POE2F、POE3F 和 MTU2CH34HIZ 位中有 1 位被置位的情况下，是否将这些引脚置为高阻抗状态。 0: 不进行输出电平的比较并且不置为高阻抗状态。 1: 进行输出电平的比较并且置为高阻抗状态。
11	—	0	R	保留位 读写的值总是 0。
10	MTU2SP1CZE	1	R/W*	MTU2S 端口 1 的输出比较 / 高阻抗允许位 设定是否对 MTU2S 的大电流引脚 PE16/TIOC3BS 和 PE17/TIOC3DS 进行输出电平比较，并且设定在 OCE2 位是 1 时 OSF2 位被置位的情况下或者在 POE4F、POE5F、POE6F、POE7F 和 MTU2SHIZ 位中有 1 位被置位的情况下，是否将这些引脚置为高阻抗状态。 0: 不进行输出电平的比较并且不置为高阻抗状态。 1: 进行输出电平的比较并且置为高阻抗状态。

位	位名	初始值	R/W	说 明
9	MTU2SP2CZE	1	R/W*	MTU2S 端口 2 的输出比较 / 高阻抗允许位 设定是否对 MTU2S 的大电流引脚 PE18/TIOC4AS 和 PE20/TIOC4CS 进行输出电平比较, 并且设定在 OCE2 位是 1 时 OSF2 位被置位的情况下或者在 POE4F、POE5F、POE6F、POE7F 和 MTU2SHIZ 位中有 1 位被置位的情况下, 是否将这些引脚置为高阻抗状态。 0: 不进行输出电平的比较并且不置为高阻抗状态。 1: 进行输出电平的比较并且置为高阻抗状态。
8	MTU2SP3CZE	1	R/W*	MTU2S 端口 3 的输出比较 / 高阻抗允许位 设定是否对 MTU2S 的大电流引脚 PE19/TIOC4BS 和 PE21/TIOC4DS 进行输出电平比较, 并且设定在 OCE2 位是 1 时 OSF2 位被置位的情况下或者在 POE4F、POE5F、POE6F、POE7F 和 MTU2SHIZ 位中有 1 位被置位的情况下, 是否将这些引脚置为高阻抗状态。 0: 不进行输出电平的比较并且不置为高阻抗状态。 1: 进行输出电平的比较并且置为高阻抗状态。
7	—	0	R	保留位 读写的值总是 1。
6	MTU2SP4CZE	0	R/W*	MTU2S 端口 4 的输出比较 / 高阻抗允许位 设定是否对 MTU2S 的大电流引脚 PD9/TIOC3BS 和 PD11/TIOC3DS 进行输出电平比较, 并且设定在 OCE2 位是 1 时 OSF2 位被置位的情况下或者在 POE4F、POE5F、POE6F、POE7F 和 MTU2SHIZ 位中有 1 位被置位的情况下, 是否将这些引脚置为高阻抗状态。 0: 不进行输出电平的比较并且不置为高阻抗状态。 1: 进行输出电平的比较并且置为高阻抗状态。
5	MTU2SP5CZE	0	R/W*	MTU2S 端口 5 的输出比较 / 高阻抗允许位 设定是否对 MTU2S 的大电流引脚 PD12/TIOC4AS 和 PD14/TIOC4CS 进行输出电平比较, 并且设定在 OCE2 位是 1 时 OSF2 位被置位的情况下或者在 POE4F、POE5F、POE6F、POE7F 和 MTU2SHIZ 位中有 1 位被置位的情况下, 是否将这些引脚置为高阻抗状态。 0: 不进行输出电平的比较并且不置为高阻抗状态。 1: 进行输出电平的比较并且置为高阻抗状态。
4	MTU2SP6CZE	0	R/W*	MTU2S 端口 6 的输出比较 / 高阻抗允许位 设定是否对 MTU2S 的大电流引脚 PD13/TIOC4BS 和 PD15/TIOC4DS 进行输出电平比较, 并且设定在 OCE2 位是 1 时 OSF2 位被置位的情况下或者在 POE4F、POE5F、POE6F、POE7F 和 MTU2SHIZ 位中有 1 位被置位的情况下, 是否将这些引脚置为高阻抗状态。 0: 不进行输出电平的比较并且不置为高阻抗状态。 1: 进行输出电平的比较并且置为高阻抗状态。
3	—	0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说 明
2	MTU2SP7CZE	0	R/W*	<p>MTU2S 端口 7 的输出比较 / 高阻抗允许位</p> <p>设定是否对 MTU2S 的大电流引脚 PD29/TIOC3BS 和 PD28/TIOC3DS 进行输出电平比较, 并且设定在 OCE2 位是 1 时 OSF2 位被置位的情况下或者在 POE4F、POE5F、POE6F、POE7F 和 MTU2SHIZ 位中有 1 位被置位的情况下, 是否将这些引脚置为高阻抗状态。</p> <p>0: 不进行输出电平的比较并且不置为高阻抗状态。 1: 进行输出电平的比较并且置为高阻抗状态。</p>
1	MTU2SP8CZE	0	R/W*	<p>MTU2S 端口 8 的输出比较 / 高阻抗允许位</p> <p>设定是否对 MTU2S 的大电流引脚 PD27/TIOC4AS 和 PD25/TIOC4CS 进行输出电平比较, 并且设定在 OCE2 位是 1 时 OSF2 位被置位的情况下或者在 POE4F、POE5F、POE6F、POE7F 和 MTU2SHIZ 位中有 1 位被置位的情况下, 是否将这些引脚置为高阻抗状态。</p> <p>0: 不进行输出电平的比较并且不置为高阻抗状态。 1: 进行输出电平的比较并且置为高阻抗状态。</p>
0	MTU2SP9CZE	0	R/W*	<p>MTU2S 端口 9 的输出比较 / 高阻抗允许位</p> <p>设定是否对 MTU2S 的大电流引脚 PD26/TIOC4BS 和 PD24/TIOC4DS 进行输出电平比较, 并且设定在 OCE2 位是 1 时 OSF2 位被置位的情况下或者在 POE4F、POE5F、POE6F、POE7F 和 MTU2SHIZ 位中有 1 位被置位的情况下, 是否将这些引脚置为高阻抗状态。</p> <p>0: 不进行输出电平的比较并且不置为高阻抗状态。 1: 进行输出电平的比较并且置为高阻抗状态。</p>

【注】 \* 在上电复位后, 只能写 1 次。

## 13.4 运行说明

为高阻抗控制对象的引脚及其条件如表 13.4 所示。

表 13.4 高阻抗控制对象的引脚及其条件

引脚	条件	详细内容
MTU2 的大电流引脚 (PE9/TIOC3B 和 PE11/TIOC3D)	进行输入电平的检测、 输出电平的比较或者 SPOER 寄存器的设定	MTU2P1CZE • ((POE3F+POE2F+POE1F+POE0F)+ (OSF1 • OCE1)+(MTU2CH34HIZ))
MTU2 的大电流引脚 (PE12/TIOC4A 和 PE14/TIOC4C)	进行输入电平的检测、 输出电平的比较或者 SPOER 寄存器的设定	MTU2P2CZE • ((POE3F+POE2F+POE1F+POE0F)+ (OSF1 • OCE1)+(MTU2CH34HIZ))
MTU2 的大电流引脚 (PE13/TIOC4B 和 PE15/TIOC4D)	进行输入电平的检测、 输出电平的比较或者 SPOER 寄存器的设定	MTU2P3CZE • ((POE3F+POE2F+POE1F+POE0F)+ (OSF1 • OCE1)+(MTU2CH34HIZ))
MTU2S 的大电流引脚 (PE16/TIOC3BS 和 PE17/TIOC3DS)	进行输入电平的检测、 输出电平的比较或者 SPOER 寄存器的设定	MTU2SP1CZE • ((POE4F+POE5F+POE6F+POE7F)+ (OSF2 • OCE2)+(MTU2SHIZ))
MTU2S 的大电流引脚 (PE18/TIOC4AS 和 PE20/TIOC4CS)	进行输入电平的检测、 输出电平的比较或者 SPOER 寄存器的设定	MTU2SP2CZE • ((POE4F+POE5F+POE6F+POE7F)+ (OSF2 • OCE2)+(MTU2SHIZ))
MTU2S 的大电流引脚 (PE19/TIOC4BS 和 PE21/TIOC4DS)	进行输入电平的检测、 输出电平的比较或者 SPOER 寄存器的设定	MTU2SP3CZE • ((POE4F+POE5F+POE6F+POE7F)+ (OSF2 • OCE2)+(MTU2SHIZ))
MTU2S 的大电流引脚 (PD9/TIOC3BS 和 PD11/TIOC3DS)	进行输入电平的检测、 输出电平的比较或者 SPOER 寄存器的设定	MTU2SP4CZE • ((POE4F+POE5F+POE6F+POE7F)+ (OSF2 • OCE2)+(MTU2SHIZ))
MTU2S 的大电流引脚 (PD12/TIOC4AS 和 PD14/TIOC4CS)	进行输入电平的检测、 输出电平的比较或者 SPOER 寄存器的设定	MTU2SP5CZE • ((POE4F+POE5F+POE6F+POE7F)+ (OSF2 • OCE2)+(MTU2SHIZ))
MTU2S 的大电流引脚 (PD13/TIOC4BS 和 PD15/TIOC4DS)	进行输入电平的检测、 输出电平的比较或者 SPOER 寄存器的设定	MTU2SP6CZE • ((POE4F+POE5F+POE6F+POE7F)+ (OSF2 • OCE2)+(MTU2SHIZ))
MTU2S 的大电流引脚 (PD29/TIOC3BS 和 PD28/TIOC3DS)	进行输入电平的检测、 输出电平的比较或者 SPOER 寄存器的设定	MTU2SP7CZE • ((POE4F+POE5F+POE6F+POE7F)+ (OSF2 • OCE2)+(MTU2SHIZ))
MTU2S 的大电流引脚 (PD27/TIOC4AS 和 PD25/TIOC4CS)	进行输入电平的检测、 输出电平的比较或者 SPOER 寄存器的设定	MTU2SP8CZE • ((POE4F+POE5F+POE6F+POE7F)+ (OSF2 • OCE2)+(MTU2SHIZ))
MTU2S 的大电流引脚 (PD26/TIOC4BS 和 PD24/TIOC4DS)	进行输入电平的检测、 输出电平的比较或者 SPOER 寄存器的设定	MTU2SP9CZE • ((POE4F+POE5F+POE6F+POE7F)+ (OSF2 • OCE2)+(MTU2SHIZ))
MTU2 的 CH0 引脚 (PE0/TIOC0A)	进行输入电平的检测或者 SPOER 寄存器的设定	MTU2PE0ZE ((POE8F • POE8E)+(MTU2CH0HIZ))
MTU2 的 CH0 引脚 (PE1/TIOC0B)	进行输入电平的检测或者 SPOER 寄存器的设定	MTU2PE1ZE ((POE8F • POE8E)+(MTU2CH0HIZ))

引脚	条件	详细内容
MTU2 的 CH0 引脚 (PE2/TIOC0C)	进行输入电平的检测或者 SPOER 寄存器的设定	MTU2PE2ZE ((POE8F · POE8E)+(MTU2CH0HIZ))
MTU2 的 CH0 引脚 (PE3/TIOC0D)	进行输入电平的检测或者 SPOER 寄存器的设定	MTU2PE3ZE ((POE8F · POE8E)+(MTU2CH0HIZ))

### 13.4.1 输入电平的检测

当  $\overline{POE0} \sim \overline{POE8}$  引脚发生 ICSR1 ~ ICSR3 设定的输入条件时，将大电流引脚和 MTU2 的 CH0 引脚置为高阻抗状态。但是，只在选择了大电流引脚和 MTU2 的 CH0 引脚的通用输入 / 输出功能或者 MTU2、MTU2S 功能时才置为高阻抗状态。

#### (1) 下降沿检测

当  $\overline{POE0} \sim \overline{POE8}$  引脚从高电平变为低电平时，将大电流引脚和 MTU2 的 CH0 引脚置为高阻抗状态。从  $\overline{POE0} \sim \overline{POE8}$  的引脚输入到将引脚置为高阻抗状态的时序例子如图 13.2 所示。

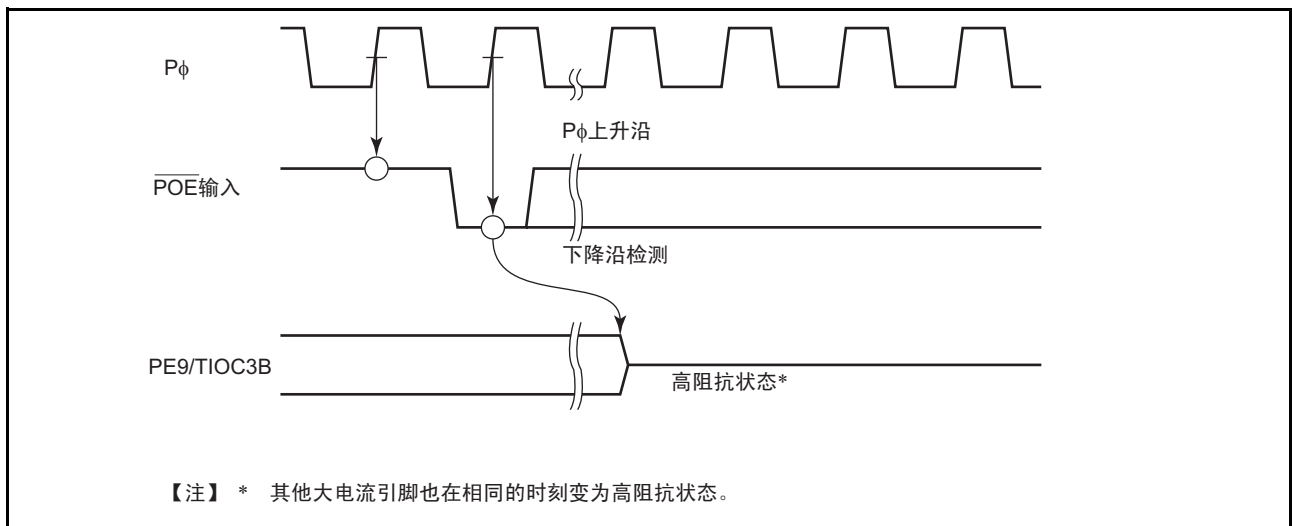


图 13.2 下降沿检测

(2) 低电平检测

低电平检测如图 13.3 所示。通过 ICSR1 ~ ICSR3 设定的采样时钟连续进行 16 次低电平采样。此时，只要有 1 次检测到高电平，就不接受请求。

另外，从采样时钟到大电流引脚变为高阻抗状态的时序与下降沿检测的情况相同。

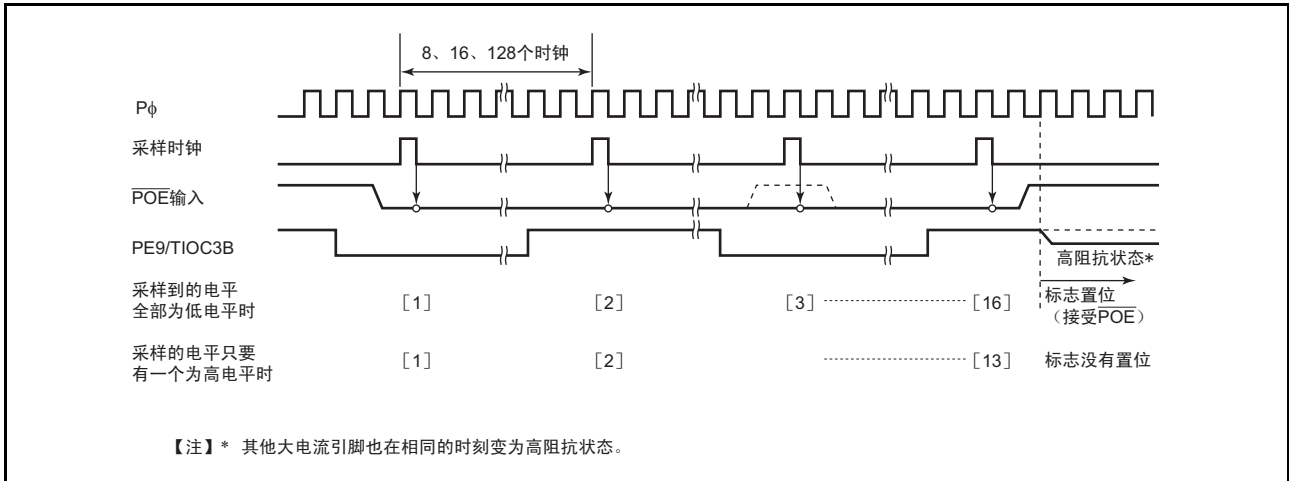


图 13.3 低电平检测

13.4.2 输出电平的比较

以 TIOC3B 和 TIOC3D 的组合为例，输出电平比较的运行如图 13.4 所示。其他引脚的组合也一样。

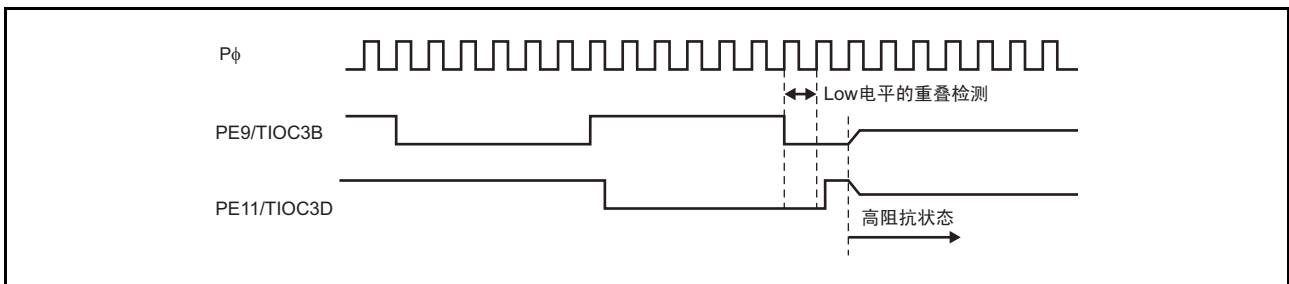


图 13.4 输出电平比较

13.4.3 高阻抗状态的解除

因输入电平检测而变为高阻抗状态的大电流引脚通过上电复位恢复到初始状态，或者通过清除 ICSR1 ~ ICSR3 的 POE0F ~ POE8F 标志 (bit12 ~ 15) 解除大电流引脚的高阻抗状态。但是，当通过 ICSR1 ~ ICSR3 的 bit0 ~ 7 设定为低电平采样时，如果不是在 POE 引脚输入高电平并进行高电平采样后，即使给标志位写 0 也无效并且不能清除标志。

解除因输出电平检测而变为高阻抗状态的大电流引脚通过上电复位恢复到初始状态，或者通过清除 OCSR1、OCSR2 的 OSF1、OSF2 标志 (bit15) 解除大电流引脚的高阻抗状态。但是，如果不是在从大电流引脚输出无效电平后，即使对标志位写 0 也无效并且不能清除标志。能通过设定 MTU2、MTU2S 内的寄存器进行无效电平的输出。

### 13.5 中断

如果在输入电平检测或者输出电平比较时条件一致，POE 就能在发出中断请求后产生中断。中断的种类以及产生中断请求的条件如表 13.5 所示。

表 13.5 中断请求的种类和条件

名称	中断源	中断标志位	条件
OEI1	输出允许中断 1	POE3F、POE2F、POE1F、POE0F、OSF1	$PIE1 \cdot (POE3F+POE2F+POE1F+POE0F)+OIE1 \cdot OSF1$
OEI3	输出允许中断 3	POE8F	$PIE3 \cdot POE8F$
OEI2	输出允许中断 2	POE4F、POE5F、POE6F、POE7F、OSF2	$PIE2 \cdot (POE4F+POE5F+POE6F+POE7F)+OIE2 \cdot OSF2$

### 13.6 使用注意事项

#### 13.6.1 从看门狗定时器产生上电复位时的引脚状态

如果从看门狗定时器 (WDT) 产生上电复位，引脚功能控制器 (PFC) 就被初始化，因此 I/O 端口变为是初始值的通用输入。但是，通过端口输出允许 (POE) 使引脚在高阻抗处理中从 WDT 产生上电复位时，在切换到通用输入为止要经过外围时钟 (Pφ) 1 个周期的时间，且该时间内引脚的状态为输出。

通过 MTU2、MTU2S 的短路检测，在高阻抗状态处理中从 WDT 产生上电复位时也是同样的状态。

通过选择定时器输出时的 POE 输入，在高阻抗处理中，从 WDT 产生上电复位时的状态如图 13.5 所示。

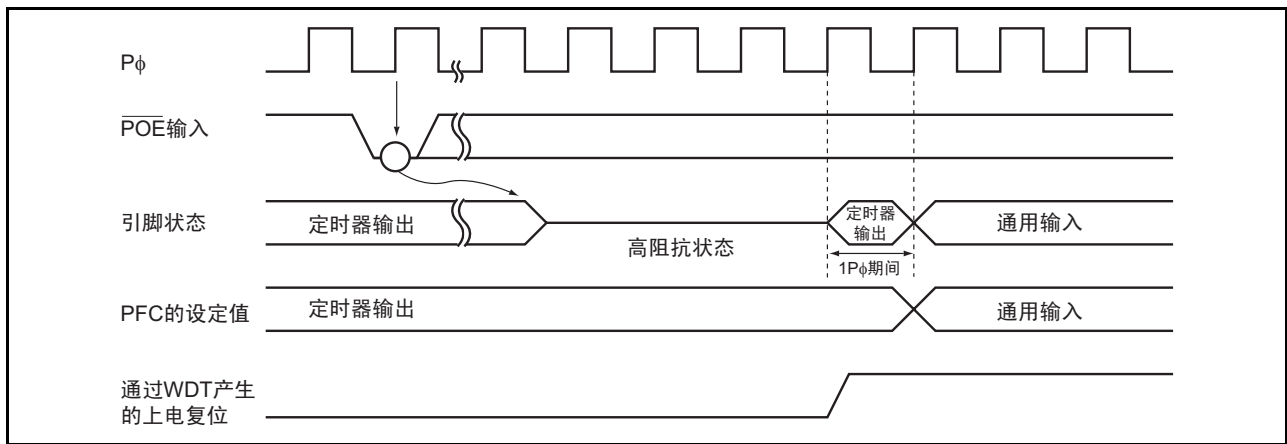


图 13.5 从看门狗定时器产生上电复位时的引脚状态



## 第 14 章 看门狗定时器 (WDT)

本 LSI 内置看门狗定时器 (WDT)，当因系统失控等原因 CPU 不能改写计数器的值而发生上溢时，能对本 LSI 内部进行复位。

WDT 是 1 个通道的定时器，能以外围时钟为输入时钟，在解除软件待机模式时用作时钟稳定时间的计数，并且还能用作间隔定时器。

### 14.1 特点

- 能用于确保时钟的稳定时间  
在解除软件待机模式时使用。
- 可在看门狗定时器模式和间隔定时器模式之间进行转换
- 在看门狗定时器模式中产生内部复位  
因计数器上溢而产生内部复位。
- 在间隔定时器模式中产生中断  
因计数器上溢而产生间隔定时器中断。
- 可选择 8 种计数器的输入时钟  
可从 8 种分频时钟 ( $\times 1 \sim \times 1/4096$ ) 中选择外围时钟。
- 可从上电复位和手动复位中选择复位的种类。

WDT 的框图如图 14.1 所示。

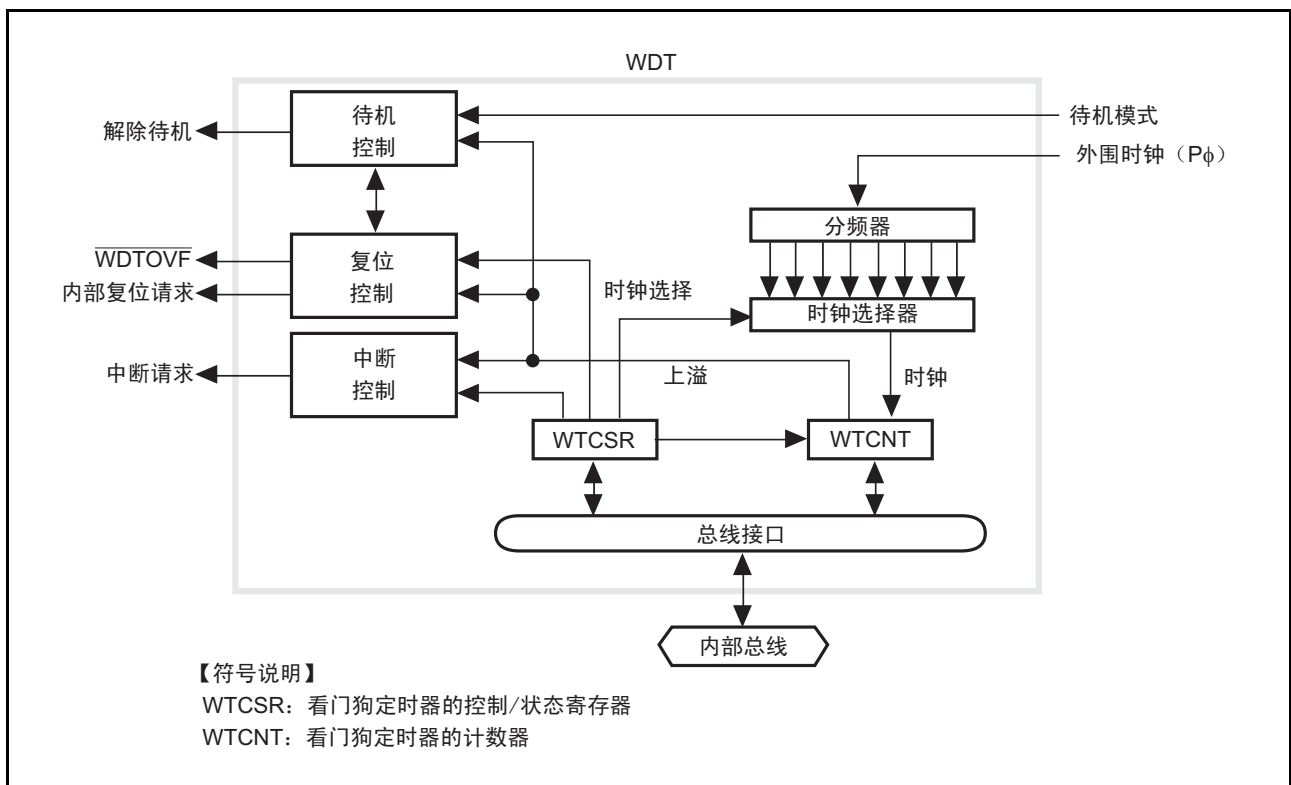


图 14.1 WDT 的框图

## 14.2 输入 / 输出引脚

WDT 的引脚功能如表 14.1 所示。

表 14.1 WDT 的引脚结构

名称	引脚名	输入 / 输出	功 能
看门狗定时器的上溢	WDTOVF	输出	因在看门狗定时器模式中发生上溢而产生内部复位，在 WTCSR 的 CKS2 ~ CKS0 位设定的 1 个时钟周期内，本引脚输出低电平。

## 14.3 寄存器说明

WDT 有以下寄存器。有关这些寄存器的地址以及各处理模式中的寄存器状态，请参照“第 27 章 寄存器一览”。

表 14.2 寄存器结构

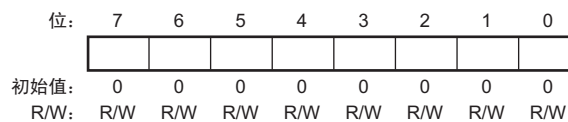
寄存器名	略称	R/W	初始值	地址	存取长度
看门狗定时器的计数器	WTCNT	R/W	H'00	H'FFFFE810	8、16
看门狗定时器的控制 / 状态寄存器	WTCSR	R/W	H'00	H'FFFFE812	8、16

### 14.3.1 看门狗定时器的计数器 (WTCNT)

WTCNT 是 8 位可读写寄存器，也是通过所选时钟进行累加计数的计数器。如果发生上溢，就会在看门狗定时器模式中产生复位，而在间隔定时器模式中产生中断。在因 WDT 的上溢而产生内部复位时，对 WTCNT 计数器进行初始化。但是，只在通过 RES 引脚进行上电复位时，WTCNT 计数器才被初始化为 H'00。

必须在将高位字节置为 H'5A 后以字单位写 WTCNT 计数器，而以字节单位读 WTCNT。

**【注】** 为了本寄存器不被误改写，写数据的方法和一般寄存器不同，详细内容请参照“14.3.3 寄存器存取时的注意事项”。



### 14.3.2 看门狗定时器的控制 / 状态寄存器 (WTCSR)

WTCSR 是 8 位可读写寄存器，由选择计数时钟的位、上溢标志位和允许位构成。

在因 WDT 的上溢而产生内部复位时，保持 WTCSR 寄存器的值。只在通过 RES 引脚进行上电复位时，WTCSR 寄存器才被初期化为 H'00。如果将 WTCSR 计数器用于软件待机解除时的时钟稳定时间的计数，就在计数器发生上溢后仍保持该值。

必须在将高位字节置为 H'A5 后以字单位写 WTCSR 寄存器，而以字节单位读 WTCSR 寄存器。

**【注】** 为了本寄存器不被误改写，写数据的方法和一般寄存器不同。详细内容请参照“14.3.3 寄存器存取时的注意事项”。

位:	7	6	5	4	3	2	1	0
	TME	WT/IT	RSTS	WOVF	IOVF	CKS[2:0]		
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7	TME	0	R/W	定时器的允许 开始或者停止定时器的运行。如果 WDT 用于解除软件待机模式，就必须将此位设定为 0。 0: 禁止定时器的运行。停止累加计数，保持 WTCNT 的值 1: 允许定时器的运行
6	WT/IT	0	R/W	定时器模式的选择 指定是用作看门狗定时器还是用作间隔定时器。 0: 间隔定时器模式 1: 看门狗定时器模式 <b>【注】</b> 如果在 WDT 运行时改写 WT/IT，就有可能无法正确地进行累加计数。
5	RSTS	0	R/W	复位选择 指定在看门狗定时器模式中 WTCNT 发生上溢后产生的复位种类，而在间隔定时器模式中忽视此设定值。 0: 上电复位 1: 手动复位
4	WOVF	0	R/W	看门狗定时器的上溢 表示在看门狗定时器模式中 WTCNT 已发生上溢，而在间隔定时器模式中此位不被置位。 0: 没有发生上溢 1: 在看门狗定时器模式中 WTCNT 已发生上溢
3	IOVF	0	R/W	间隔定时器的上溢 表示在间隔定时器模式中 WTCNT 已发生上溢，而在看门狗定时器模式中此位不被置位。 0: 没有发生上溢 1: 在间隔定时器模式中 WTCNT 已发生上溢

位	位名	初始值	R/W	说 明
2 ~ 0	CKS[2:0]	000	R/W	<p>时钟选择 2 ~ 0 从分频外围时钟 (P<math>\phi</math>) 得到的 8 种时钟中选择用于 WTCNT 计数的时钟。括弧内为外围时钟 P<math>\phi</math>=40MHz 时的上溢周期值。</p> <p>000: P<math>\phi</math> (6.4<math>\mu</math>s) 001: P<math>\phi</math>/4 (25.6<math>\mu</math>s) 010: P<math>\phi</math>/16 (102.4<math>\mu</math>s) 011: P<math>\phi</math>/32 (204.8<math>\mu</math>s) 100: P<math>\phi</math>/64 (409.6<math>\mu</math>s) 101: P<math>\phi</math>/256 (1.64ms) 110: P<math>\phi</math>/1024 (6.55ms) 111: P<math>\phi</math>/4096 (26.21ms)</p> <p>【注】 如果在 WDT 运行时改写 CKS2 ~ CKS0 位, 就有可能无法正确地进行累加计数, 所以在改写 CKS2 ~ CKS0 位时, 必须使 WDT 停止。</p>

### 14.3.3 寄存器存取时的注意事项

为了看门狗定时器的计数器 (WTCNT) 和看门狗定时器的控制 / 状态寄存器不易被改写, 写数据的方法和一般寄存器不同。必须按照以下方法进行写操作:

- 在写 WTCNT 和 WTCSR 时, 必须使用字传送指令, 而字节传送和长字传送指令不能用于写操作。

如图 14.2 所示, 在写 WTCNT 时, 必须将高位字节置为 H'5A、低位字节置为写数据后进行传送; 在写 WTCSR 时, 必须将高位字节置为 H'A5、低位字节置为写数据后进行传送。如此进行传送, 就能将低位字节的数据写到 WTCNT 或者 WTCSR。

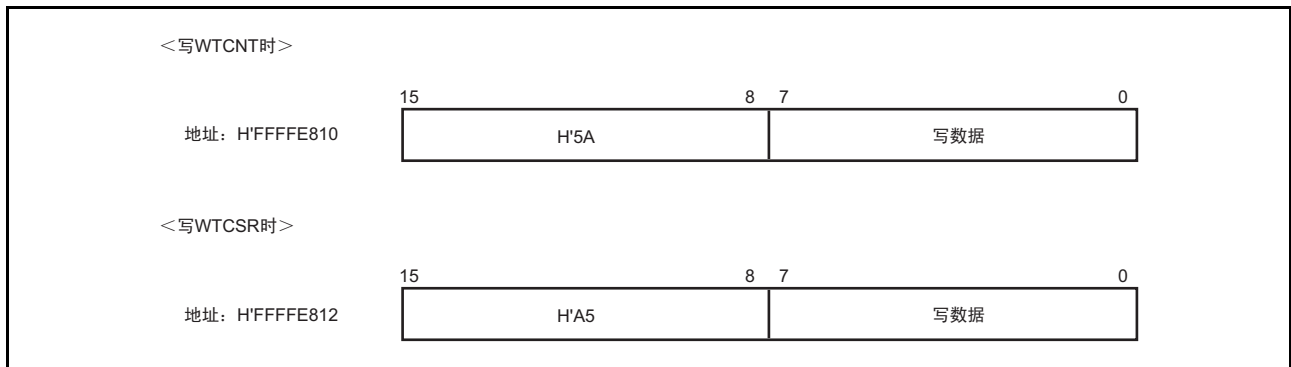


图 14.2 WTCNT 和 WTCSR 的写操作

## 14.4 运行说明

### 14.4.1 软件待机模式的解除步骤

在通过 NMI 中断或者外部中断 (IRQ) 解除软件待机模式时使用 WDT，步骤如下所示 (因为在通过复位进行解除时 WDT 不运行，所以在时钟稳定前，RES 引脚必须保持低电平)。

1. 在进入软件待机模式前，必须将 WTCSR 寄存器的 TME 位设定为 0。如果将 TME 位设定为 1，就有可能在计数上溢时产生不正常的复位或者间隔定时器中断。
2. 预先将要使用的计数时钟的种类和计数器的初始值分别设定到 WTCSR 寄存器 CKS2 ~ CKS0 位和 WTCNT 计数器。在设定这些值时，计数上溢前的时间必须长于时钟振荡稳定时间。
3. 通过执行 SLEEP 指令进入软件待机模式，停止时钟的运行。
4. 如果 NMI 引脚或者 IRQ 引脚的输入电平发生变化，就通过边沿检测开始 WDT 的计数。
5. 如果 WDT 发生计数上溢，CPG 就开始提供时钟，本 LSI 重新开始运行。此时，WTCSR 寄存器的 WOVSF 标志不被置位。

### 14.4.2 看门狗定时器模式的用法

在看门狗定时器模式的运行中，每当计数器发生上溢，就产生由 WTCSR 寄存器的 RSTS 位选择的内部复位，并且 WDTOVF 引脚输出有效信号。

1. 将 WTCSR 寄存器的 WT/IT 位设定为 1，并将复位的种类设定到 RSTS 位、计数时钟的种类设定到 CKS2 ~ CKS0 位、计数器的初始值设定到 WTCNT 计数器。
2. 如果 WTCSR 寄存器的 TME 位被置 1，就在看门狗定时器模式中开始计数。
3. 在看门狗定时器模式的运行中，如果要使计数器不发生上溢，就必须定期改写计数器。
4. 如果计数器发生上溢，WDT 就将 WTCSR 寄存器的 WOVSF 标志置 1，在 CKS2 ~ CKS0 位设定的计数器时钟的 1 个周期内 WDTOVF 引脚输出有效信号，并且产生由 RSTS 位指定种类的复位，然后计数器继续计数。

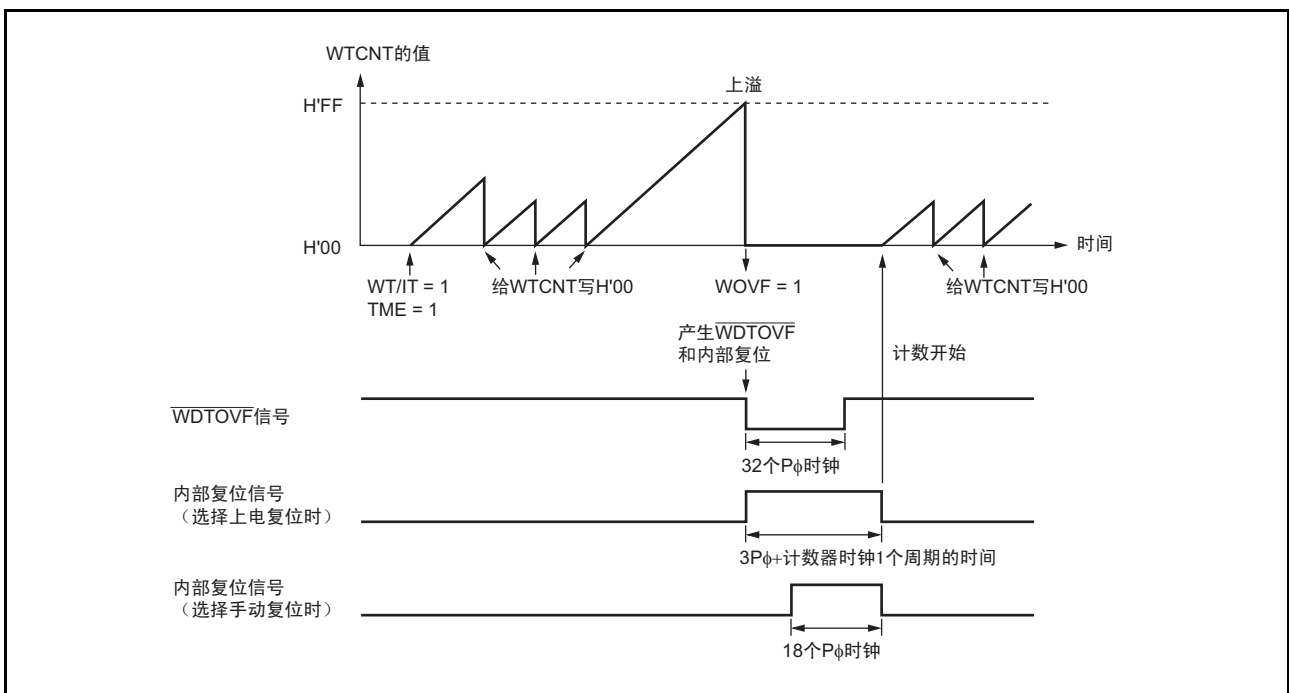


图 14.3 看门狗定时器模式中的运行  
(通过 CKS0 ~ CKS2 位将 WTCNT 的计数器时钟设定为  $P\phi/32$ )

### 14.4.3 间隔定时器模式的使用法

在间隔定时器模式的运行中，每当计数器发生上溢时，就产生间隔定时器中断，因此能每隔一定的时间产生 1 次中断。

1. 将 WTCSR 寄存器的 WT/IT 位清 0，并将计数时钟的种类设定到 CKS2 ~ CKS0 位、计数器的初始值设定到 WTCNT 计数器。
2. 如果 WTCSR 寄存器的 TME 位被置 1，就在间隔定时器模式中开始计数。
3. 当计数器发生上溢时，WDT 将 WTCSR 寄存器的 IOVF 标志置 1，并将间隔定时器的中断请求传送到 INTC，然后计数器继续计数。

## 14.5 使用注意事项

### 14.5.1 WTCNT 的设定值

如果在间隔定时器模式中给 WTCNT 设定 H'FF，就在 1 个计数时钟后的 H'FF→H'00 时不发生上溢，而在 257 个计数时钟后的 H'FF→H'00 时发生上溢。

如果在看门狗定时器模式中给 WTCNT 设定 H'FF，就在 1 个计数时钟后的 H'FF→H'00 时发生上溢。

## 第 15 章 串行通信接口 (SCI)

本 LSI 具有 3 个独立通道的串行通信接口 (SCI: Serial Communication Interface)。SCI 能以异步通信和时钟同步通信的 2 种方式进行串行通信, 在异步模式中, 能与 Universal Asynchronous Receiver/Transmitter (UART) 或者 Asynchronous Communication Interface Adapter (ACIA) 等标准异步通信 LSI 进行串行数据的通信。另外, 在异步模式中具有多个处理器之间的串行通信功能 (多处理器通信功能)。

### 15.1 特点

- 可从异步模式或者时钟同步模式中选择串行通信模式
- 异步模式
  - 通过以字符为单位取得同步的异步方式进行串行数据的通信, 能与 Universal Asynchronous Receiver/Transmitter (UART) 和 Asynchronous Communication Interface Adapter (ACIA) 等标准异步通信 LSI 进行串行数据的通信。
  - 能从 12 种格式中选择串行数据的通信格式。
  - 数据长: 7 位或者 8 位
  - 停止位长: 1 位或者 2 位
  - 奇偶校验: 偶校验、奇校验或者无奇偶校验
  - 可进行多处理器间的通信
  - 接收错误的检测: 检测奇偶校验错误、溢出错误和帧错误
  - 中止的检测: 当发生帧错误时, 能通过直接读 RXD 引脚的电平检测中止。
- 时钟同步模式
  - 与时钟同步进行串行数据通信, 能与具有时钟同步通信功能的其他 LSI 进行串行数据的通信。
  - 串行数据的通信格式为 1 种。
  - 数据长: 8 位
  - 接收错误的检测: 检测溢出错误
- 能进行全双工通信
  - 因为具有独立的发送部和接收部, 所以能同时发送和接收。另外, 因为发送部和接收部都为双缓冲结构, 所以能进行串行数据的连续发送和连续接收。
- 能通过内部波特率发生器选择任意的位速率
- 发送和接收的时钟源可选择波特率发生器的内部时钟或者 SCK 引脚的外部时钟
- 可选择 LSB 先或者 MSB 先 (异步 7 位数据除外)
- 4 种中断源
  - 中断源有发送数据空、发送结束、接收数据满和接收错误共 4 种, 能分别独立请求中断。另外, 能通过发送数据空请求和接收数据满请求, 启动数据传送控制器 (DTC) 或者 DMA 控制器 (DMAC) 进行数据的传送。
- 能设定模块待机模式

SCI 的框图如图 15.1 所示。

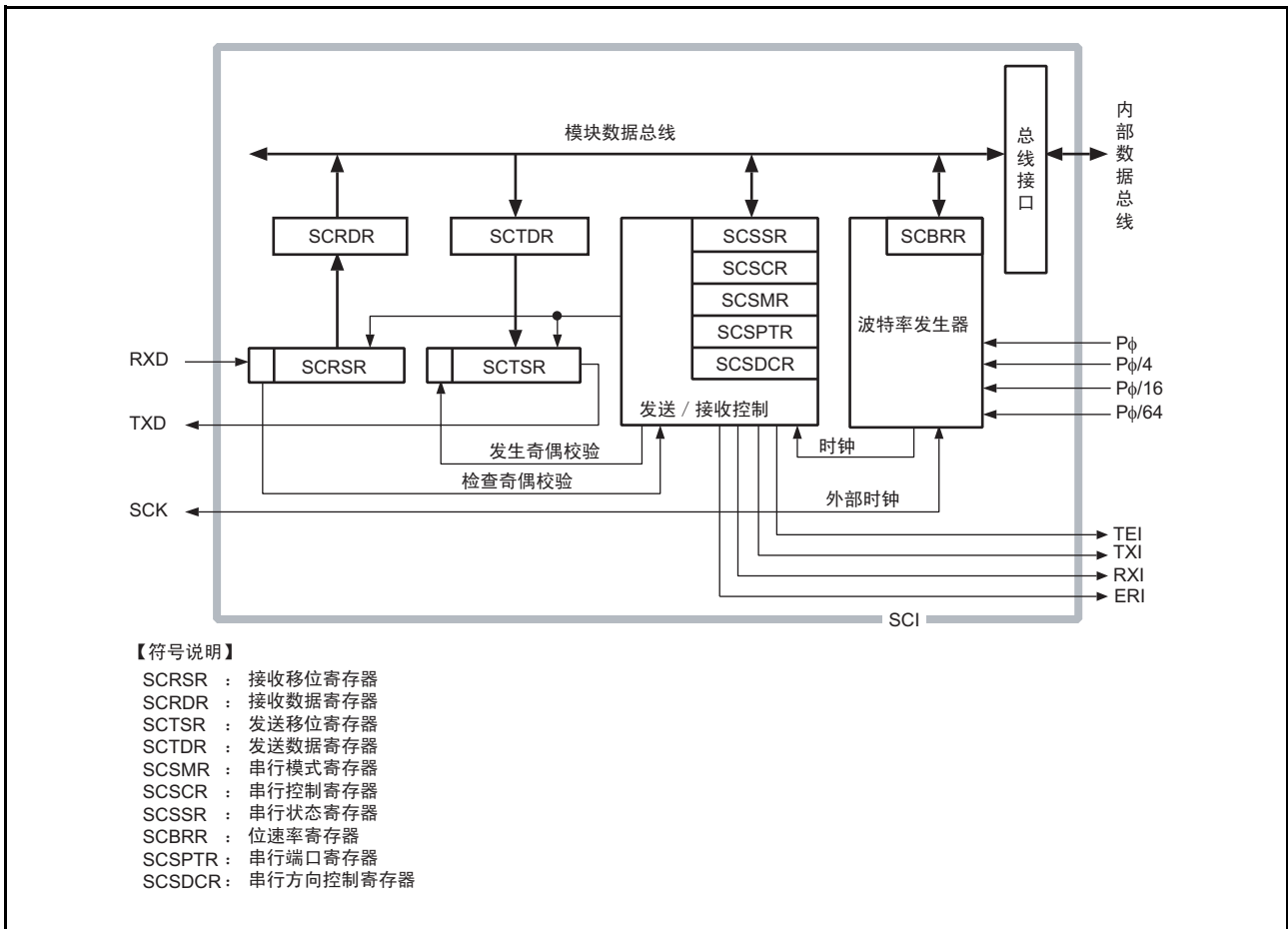


图 15.1 SCI 的框图

### 15.2 输入 / 输出引脚

SCI 有表 15.1 所示的输入 / 输出引脚。

表 15.1 引脚结构

通道	引脚名 *	输入 / 输出	功能
0	SCK0	输入 / 输出	通道 0 的时钟输入 / 输出引脚
	RXD0	输入	通道 0 的接收数据的输入引脚
	TXD0	输出	通道 0 的发送数据的输出引脚
1	SCK1	输入 / 输出	通道 1 的时钟输入 / 输出引脚
	RXD1	输入	通道 1 的接收数据的输入引脚
	TXD1	输出	通道 1 的发送数据的输出引脚
2	SCK2	输入 / 输出	通道 2 的时钟输入 / 输出引脚
	RXD2	输入	通道 2 的接收数据的输入引脚
	TXD2	输出	通道 2 的发送数据的输出引脚

【注】 \* 在本文中省略通道，分别略称为 SCK、RXD、TXD。



## 15.3 寄存器说明

SCI 有以下各通道的寄存器。有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 27 章 寄存器一览”。

表 15.2 寄存器结构

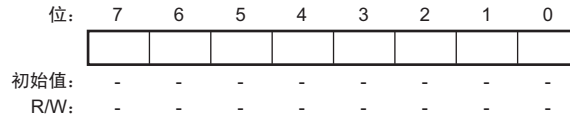
通道	寄存器名	略称	R/W	初始值	地址	存取长度
0	串行模式寄存器_0	SCSMR_0	R/W	H'00	H'FFFFFFC000	8
	位速率寄存器_0	SCBRR_0	R/W	H'FF	H'FFFFFFC002	8
	串行控制寄存器_0	SCSCR_0	R/W	H'00	H'FFFFFFC004	8
	发送数据寄存器_0	SCTDR_0	—	—	H'FFFFFFC006	8
	串行状态寄存器_0	SCSSR_0	R/W	H'84	H'FFFFFFC008	8
	接收数据寄存器_0	SCRDR_0	—	—	H'FFFFFFC00A	8
	串行方向控制寄存器_0	SCSDCR_0	R/W	H'F2	H'FFFFFFC00C	8
	串行端口寄存器_0	SCSPTR_0	R/W	H'0x	H'FFFFFFC00E	8
1	串行模式寄存器_1	SCSMR_1	R/W	H'00	H'FFFFFFC080	8
	位速率寄存器_1	SCBRR_1	R/W	H'FF	H'FFFFFFC082	8
	串行控制寄存器_1	SCSCR_1	R/W	H'00	H'FFFFFFC084	8
	发送数据寄存器_1	SCTDR_1	—	—	H'FFFFFFC086	8
	串行状态寄存器_1	SCSSR_1	R/W	H'84	H'FFFFFFC088	8
	接收数据寄存器_1	SCRDR_1	—	—	H'FFFFFFC08A	8
	串行方向控制寄存器_1	SCSDCR_1	R/W	H'F2	H'FFFFFFC08C	8
	串行端口寄存器_1	SCSPTR_1	R/W	H'0x	H'FFFFFFC08E	8
2	串行模式寄存器_2	SCSMR_2	R/W	H'00	H'FFFFFFC100	8
	位速率寄存器_2	SCBRR_2	R/W	H'FF	H'FFFFFFC102	8
	串行控制寄存器_2	SCSCR_2	R/W	H'00	H'FFFFFFC104	8
	发送数据寄存器_2	SCTDR_2	—	—	H'FFFFFFC106	8
	串行状态寄存器_2	SCSSR_2	R/W	H'84	H'FFFFFFC108	8
	接收数据寄存器_2	SCRDR_2	—	—	H'FFFFFFC10A	8
	串行方向控制寄存器_2	SCSDCR_2	R/W	H'F2	H'FFFFFFC10C	8
	串行端口寄存器_2	SCSPTR_2	R/W	H'0x	H'FFFFFFC10E	8

### 15.3.1 接收移位寄存器 (SCRSR)

SCRSR 是接收串行数据的寄存器。

SCI 按照从 LSB (bit0) 开始接收的顺序, 将 RXD 引脚输入的串行数据保存到 SCRSR, 并转换为并行数据。当结束 1 字节数据的接收时, 数据就自动传送到 SCRDR。

CPU 不能直接读写 SCRSR。



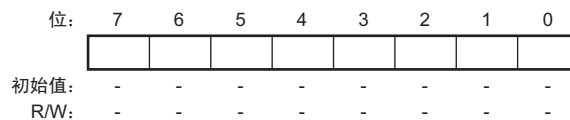
### 15.3.2 接收数据寄存器 (SCRDR)

SCRDR 是保存已接收串行数据的寄存器。

当结束 1 字节串行数据的接收时, SCI 就将接收的串行数据从接收移位寄存器 (SCRSR) 传送并保存到 SCRDR, 然后结束接收运行, SCRSR 为可接收状态。

因为 SCRSR 和 SCRDR 为双缓冲, 所以能连续接收。

SCRDR 是只读寄存器, 所以 CPU 不能写此寄存器。



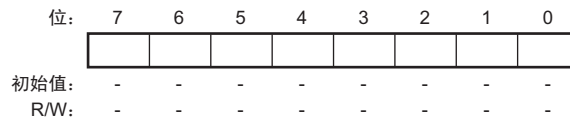
### 15.3.3 发送移位寄存器 (SCTSR)

SCTSR 是发送串行数据的寄存器。

一旦将发送数据从发送数据寄存器 (SCTDR) 传送到 SCTSR, SCI 就从 LSB (bit0) 按顺序将数据送到 TXD 引脚, 进行串行数据的发送。

当结束 1 字节数据的发送时, 就自动将下一个发送数据从 SCTDR 传送到 SCTSR, 开始发送。但是, 在串行状态寄存器 (SCSSR) 的 TDRE 标志被置 1 时, 不将数据从 SCTDR 传送到 SCTSR。

CPU 不能直接读写 SCTSR。



### 15.3.4 发送数据寄存器 (SCTDR)

SCTDR 是保存串行发送数据的 8 位寄存器。

如果检测到发送移位寄存器 (SCTSR) 为空, SCI 就将写到 SCTDR 的发送数据传送到 SCTSR, 开始串行发送。如果在 SCTSR 的串行数据发送中将下一个发送数据写到 SCTDR, 就能连续串行发送。

CPU 随时能读写 SCTDR。

位:	7	6	5	4	3	2	1	0
初始值:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

### 15.3.5 串行模式寄存器 (SCSMR)

SCSMR 是设定 SCI 的串行通信格式和选择波特率发生器时钟源的 8 位寄存器。

CPU 随时能读 SCSMR。

位:	7	6	5	4	3	2	1	0
	C/ $\bar{A}$	CHR	PE	O/ $\bar{E}$	STOP	MP	CKS[1:0]	
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7	C/ $\bar{A}$	0	R/W	通信模式 SCI 的运行模式可选择异步模式或者时钟同步模式。 0: 以异步模式运行 1: 以时钟同步模式运行
6	CHR	0	R/W	字符长 异步模式的数据长可选择 7 位或者 8 位数据。在时钟同步模式中与 CHR 的设定无关, 数据长固定为 8 位数据。如果选择 7 位数据, 就不发送 SCTDR (发送数据寄存器) 的 MSB (bit7)。 0: 8 位数据 1: 7 位数据
5	PE	0	R/W	奇偶校验的允许 在异步模式中, 选择是否在发送时附加奇偶校验位以及在接收时检查奇偶校验位。在时钟同步模式中, 与 PE 位的设定无关, 不附加和检查奇偶校验位。 0: 禁止奇偶校验位的附加和检查 1: 允许奇偶校验位的附加和检查 * <b>【注】</b> * 如果将 PE 位设定为 1, 就在发送时将 O/ $\bar{E}$ 位指定的偶校验或者奇校验附加到发送数据后发送; 而在接收时检查接收的奇偶校验位是否为 O/ $\bar{E}$ 位指定的偶校验或者奇校验。

位	位名	初始值	R/W	说 明
4	O $\bar{E}$	0	R/W	<p>奇偶校验模式</p> <p>通过偶校验或者奇校验选择奇偶校验的附加和检查。O<math>\bar{E}</math> 位的设定只在异步模式中将 PE 位设定为 1 (允许奇偶校验位的附加和检查) 时有效。在时钟同步模式或者异步模式中禁止附加和检查奇偶校验为时, O<math>\bar{E}</math> 位的指定无效。</p> <p>0: 偶校验 1: 奇校验</p> <p>如果设定为偶校验, 就在发送时附加奇偶校验位, 使奇偶校验位和发送字符中的 1 的个数为偶数, 然后发送。在接收时检查奇偶校验位和接收字符中的 1 的个数是否为偶数。</p> <p>如果设定为奇校验, 就在发送时附加奇偶校验位, 使奇偶校验位和发送字符中的 1 的个数为奇数, 然后发送。在接收时检查奇偶校验位和接收字符中的 1 的个数是否为奇数。</p>
3	STOP	0	R/W	<p>停止位长</p> <p>从 1 位或者 2 位中选择异步模式中的停止位长。STOP 位的设定只在异步模式中有效。在设定为时钟同步模式时, 因为不附加停止位, 所以此位的设定无效。</p> <p>0: 1 个停止位 *1 1: 2 个停止位 *2</p> <p>另外, 在接收时, 与 STOP 位的设定无关, 只检查接收的停止位的第 1 位。当停止位的第 2 位是 1 时, 作为停止位处理, 而当停止位的第 2 位是 0 时, 作为下一个发送字符的起始位处理。</p> <p><b>【注】</b> *1 发送时, 在发送字符的末尾附加 1 位的 1 (停止位) 后发送。 *2 发送时, 在发送字符的末尾附加 2 位的 1 (停止位) 后发送。</p>
2	MP	0	R/W	<p>多处理器模式 (只在异步模式中有效)</p> <p>允许或者禁止多处理器功能。在多处理器模式中, PE、O<math>\bar{E}</math> 位的设定无效。</p> <p>0: 禁止多处理器 1: 允许多处理器</p>
1、0	CKS[1:0]	00	R/W	<p>时钟的选择 1、0</p> <p>选择内部波特率发生器的时钟源。能通过设定 CKS1、CKS0 位, 从 P<math>\phi</math>、P<math>\phi</math>/4、P<math>\phi</math>/16、P<math>\phi</math>/64 中选择时钟源。</p> <p>有关时钟源、位速率寄存器的设定值和波特率的关系, 请参照 “15.3.10 位速率寄存器 (SCBRR)”。</p> <p>00: P<math>\phi</math> 钟源 01: P<math>\phi</math>/4 钟源 10: P<math>\phi</math>/16 钟源 11: P<math>\phi</math>/64 钟源</p> <p><b>【注】</b> P<math>\phi</math>: 外围钟源</p>

## 15.3.6 串行控制寄存器 (SCSCR)

SCSCR 是进行 SCI 的发送和接收、输出异步模式中的串行时钟、允许或者禁止中断请求，以及选择发送和接收时钟源的寄存器。

CPU 随时能读写 SCSCR。

位:	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7	TIE	0	R/W	发送的中断允许 在串行状态寄存器 (SCSSR) 的 TDRE 标志被置 1 时 (串行发送数据从发送数据寄存器 (SCTDR) 传送到发送移位寄存器 (SCTSR) 后), 允许或者禁止发送数据空的中断 (TXI) 请求。 能通过读取 TDRE 标志的 1 后将其清 0 或者通过将 TIE 位清 0, 解除 TXI 中断请求。 0: 禁止发送数据空的中断 (TXI) 请求 1: 允许发送数据空的中断 (TXI) 请求
6	RIE	0	R/W	接收的中断允许 在 SCSSR 的 RDRF 标志被置 1 时 (串行接收数据从接收移位寄存器 (SCRSR) 传送到接收数据寄存器 (SCRDR) 后), 允许或者禁止发生接收数据满的中断 (RXI) 请求和接收错误的中断 (ERI) 请求。 能通过读取 RDRF 标志或者 FER、PER、ORER 标志的 1 后将其清 0、或者通过将 RIE 位清 0, 解除 RXI 和 ERI 中断请求。 0: 禁止接收数据满的中断 (RXI) 请求和接收错误的中断 (ERI) 请求 1: 允许接收数据满的中断 (RXI) 请求和接收错误的中断 (ERI) 请求
5	TE	0	R/W	发送允许 允许或者禁止 SCI 的串行发送。 0: 禁止发送 *1 1: 允许发送 *2 <b>【注】</b> *1 SCSSR 的 TDRE 标志固定为 1。 *2 在此状态下, 如果在发送数据写到 SCTDR 后将 SCSSR 的 TDRE 标志清 0, 就开始串行发送。 另外, 必须在将 TE 位设定为 1 前设定串行模式寄存器 (SCSMR), 决定发送格式。
4	RE	0	R/W	接收的允许 允许或者禁止 SCI 的串行接收。 0: 禁止接收 *1 1: 允许接收 *2 <b>【注】</b> *1 必须注意: 即使将 RE 位清 0, RDRF、FER、PER、ORER 的各标志也不受影响而保持状态。 *2 在此状态下, 如果在异步模式中检测到起始位或者在时钟同步模式中检测到同步时钟输入, 就开始串行接收。 另外, 必须在将 RE 位设定为 1 前设定 SCSMR, 决定接收格式。

位	位名	初始值	R/W	说 明
3	MPIE	0	R/W	<p>多处理器的中断允许（在异步模式中 SCSMR 的 MP=1 时有效）</p> <p>如果将此位设定为 1，就跳读多处理器位是 0 的数据，禁止设定 SCSSR 的 RDRF、FER、ORER 的各状态标志。如果接收到多处理器位是 1 的数据，就自动清除此位，返回到通常的接收运行。详细内容请参照“15.4.4 多处理器通信功能”。</p>
2	TEIE	0	R/W	<p>发送结束的中断允许</p> <p>在发送 MSB 数据并且 SCTDR 中无有效的发送数据时，允许或者禁止发送结束的中断（TEI）请求。</p> <p>能通过在读取 SCSSR 的 TDRE 标志的 1 后将此标志和 TEND 标志清 0、或者通过将 TEIE 位清 0，解除 TEI 中断请求。</p> <p>0：禁止发送结束的中断（TEI）请求 1：允许发送结束的中断（TEI）请求</p>
1、0	CKE[1:0]	00	R/W	<p>时钟的允许 1、0</p> <p>选择 SCI 的时钟源，允许或者禁止 SCK 引脚的时钟输出。通过 CKE1 位和 CKE0 位的组合，将 SCK 引脚设定为串行时钟的输出引脚或者串行时钟的输入引脚。</p> <p>在时钟同步模式中，如果设定为同步时钟输出，就必须在将 SCSMR 的 C/A 位设定为 1 后，设定 CKE1、CKE0 位。有关 SCI 的时钟源选择，请参照“15.4 运行说明”的表 15.14。</p> <p>异步模式</p> <p>00：内部时钟 /SCK 引脚是输入引脚（忽视输入信号） 01：内部时钟 /SCK 引脚是时钟的输出引脚 *1 10：外部时钟 /SCK 引脚是时钟的输入引脚 *2 11：外部时钟 /SCK 引脚是时钟的输入引脚 *2</p> <p>时钟同步模式</p> <p>00：内部时钟 /SCK 引脚是同步时钟的输出引脚 01：内部时钟 /SCK 引脚是同步时钟的输出引脚 10：外部时钟 /SCK 引脚是同步时钟的输入引脚 11：外部时钟 /SCK 引脚是同步时钟的输入引脚</p> <p>【注】 *1 输出时钟的频率是位速率的 16 倍 *2 输入时钟的频率是位速率的 16 倍</p>

## 15.3.7 串行状态寄存器 (SCSSR)

SCSSR 是表示 SCI 运行状态的状态标志的 8 位寄存器。

CPU 随时能读写 SSR。但是，不能给 TDRE、RDRF、ORER、PER、FER 的各标志写 1。另外，为了将这些标志清 0，需要预先读取 1。而且，TEND 标志为只读，不能写。

位:	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
初始值:	1	0	0	0	0	1	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

【注】\* 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说 明
7	TDRE	1	R/(W)*	发送数据寄存器空 表示数据已从发送数据寄存器 (SCTDR) 传送到发送移位寄存器 (SCTSR)，并且为能将下一个串行发送数据写到 SCTDR 的状态。 0: 表示已将有效的发送数据写到 SCTDR [清除条件] <ul style="list-style-type: none"> <li>• 当读 TDRE=1 的状态后写 0 时</li> <li>• 通过 TXI 中断，用 DMAC 给 SCTDR 写数据时</li> <li>• 通过 TXI 中断启动 DTC 并且在 DTC 的 MRB 的 DISEL 位为 0 时给 SCTDR 传送发送数据时</li> </ul> 1: 表示 SCTDR 中无有效的发送数据 [置位条件] <ul style="list-style-type: none"> <li>• 当上电复位或者硬件待机模式时</li> <li>• 当 SCSCR 的 TE 位为 0 时</li> <li>• 当数据已从 SCTDR 传送到 SCTSR 并且为能将数据写到 SCTDR 的状态时</li> </ul>
6	RDRF	0	R/(W)*	接收数据寄存器满 表示接收的数据被保存到接收数据寄存器 (SCRDR)。 0: 表示有效的接收数据没被保存到 SCRDR [清除条件] <ul style="list-style-type: none"> <li>• 当上电复位或者硬件待机模式时</li> <li>• 当读 RDRF=1 后写 0 时</li> <li>• 通过 RXI 中断，用 DMAC 读 SCRDR 的数据时</li> <li>• 通过 RXI 中断启动 DTC 并且在 DTC 的 MRB 的 DISEL 位为 0 时从 SCRDR 传送数据时</li> </ul> 1: 表示有效的接收数据已被保存到 SCRDR [置位条件] <ul style="list-style-type: none"> <li>• 当串行接收正常结束且将接收数据从 SCRSR 传送到 SCRDR 时</li> </ul> 【注】在接收时检测到错误或者将串行控制寄存器 (SCSCR) 的 RE 位清 0 时，SCRDR 和 RDRF 标志不受影响而保持以前的状态。必须注意：如果在 RDRF 标志被置 1 的状态下结束下一个数据的接收，就发生溢出错误，丢失接收的数据。

位	位名	初始值	R/W	说 明
5	ORER	0	R/(W)*	<p>溢出错误 表示接收时发生溢出错误并异常结束。</p> <p>0: 表示正在接收或者接收正常结束 *1 [清除条件]</p> <ul style="list-style-type: none"> <li>• 在上电复位或者硬件待机模式时</li> <li>• 当读 ORER=1 的状态后写 0 时</li> </ul> <p>1: 表示在接收时发生溢出错误 *2 [置位条件]</p> <ul style="list-style-type: none"> <li>• 在 RDRF=1 的状态下结束到下一个串行接收时</li> </ul> <p>【注】 *1 在将 SCSCR 的 RE 位清 0 时, ORER 标志不受影响而保持以前的状态。 *2 SCRDR 保持溢出错误发生前的接收数据, 后接收的数据会丢失。另外, 在 ORER=1 的状态下, 不能继续以后的串行接收。</p>
4	FER	0	R/(W)*	<p>帧错误 在异步模式中, 表示在接收时发生帧错误并异常结束。</p> <p>0: 表示正在接收或者接收正常结束 *1 [清除条件]</p> <ul style="list-style-type: none"> <li>• 在上电复位或者硬件待机模式时</li> <li>• 当读 FER=1 的状态后写 0 时</li> </ul> <p>1: 表示在接收时发生帧错误 [置位条件]</p> <ul style="list-style-type: none"> <li>• 当 SCI 在结束接收时检查接收数据末尾的停止位是否为 1、停止位是否为 0 时 *2</li> </ul> <p>【注】 *1 在将 SCSCR 的 RE 位清 0 时, FER 标志不受影响而保持以前的状态。 *2 在 2 个停止位模式中, 只判断第 1 位的停止位是否为 1, 不检查第 2 位的停止位。另外, 将发生帧错误时的接收数据传送到 SCRDR, 但是, RDRF 标志不被置位。另外, 在 FER 标志被置 1 的状态下, 不能继续以后的串行接收。</p>
3	PER	0	R/(W)*	<p>奇偶校验错误 在异步模式中, 表示在附加奇偶校验的接收时发生奇偶校验错误并异常结束。</p> <p>0: 表示正在接收或者接收正常结束 *1 [清除条件]</p> <ul style="list-style-type: none"> <li>• 在上电复位或者硬件待机模式时</li> <li>• 当读 PER=1 的状态后写 0 时</li> </ul> <p>1: 表示在接收时发生奇偶校验错误 *2 [置位条件]</p> <ul style="list-style-type: none"> <li>• 当接收时的接收数据和奇偶校验位的 1 的个数与串行模式寄存器 (SCSMR) 的 O/E 位指定的偶校验或奇校验的设定不一致时</li> </ul> <p>【注】 *1 在将 SCSCR 的 RE 位清 0 时, PER 标志不受影响而保持以前的状态。 *2 将发生奇偶校验错误时的接收数据传送到 SCRDR, 但是 RDRF 标志不被置位。另外, 在 PER 标志被置 1 的状态下, 不能继续以后的串行接收。</p>



位	位名	初始值	R/W	说 明
2	TEND	1		发送的结束 表示在进行发送字符的最后位时 SCTDR 中无有效数据并结束发送。 TEND 标志为只读，不能写。 0: 表示正在发送 [清除条件] • 当读 TDRE=1 的状态后给 TDRE 标志写 0 时 1: 表示发送已结束 [置位条件] • 在上电复位或者硬件待机模式时 • 当 SCSCR 的 TE 位为 0 时 • 在发送 1 字节的串行发送字符的最后位时 TDRE=1 <b>【注】</b> 在通过 TXI 中断启动 DMAC 或者 DTC 并且将数据写到 SCTDR 时，TEND 标志不定，所以 TEND 标志不能用作发送结束标志。
1	MPB	0		多处理器位 保存接收帧中的多处理器位的值。当 SCSCR 的 RE 为 0 时，此位不变化。
0	MPBT	0		多处理器位的传送 设定给发送帧附加的多处理器位的值。

**【注】** \* 为了清除标志，只能在读 1 后写 0。

### 15.3.8 串行端口寄存器 (SCSPTR)

SCSPTR 对串行通信接口 (SCI) 引脚多路复用的端口进行输入 / 输出和数据的控制。能通过写 TXD 引脚的输出数据，控制串行发送和接收的中止，并且能用 bit3 和 bit2 读取 SCK 引脚的数据以及写 SCK 引脚的输出数据，bit7 控制 RXI 中断的允许和禁止。SCSPTR 为 8 位寄存器，能随时由 CPU 读写。另外，读 SCI 引脚的值时，必须使用端口寄存器。详细内容请参照“第 22 章 I/O 端口”。

位:	7	6	5	4	3	2	1	0
	EIO	-	-	-	SPB1IO	SPB1DT	SPB0IO	SPB0DT
初始值:	0	0	0	0	0	不定	0	不定
R/W:	R/W	-	-	-	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7	EIO	0	R/W	仅错误中断 当 EIO 位是 1 时，即使将 RIE 位置 1，也不向 CPU 请求 RXI 中断。如果此设定使用 DMAC，CPU 就只处理 ERI 中断，DMAC 将读取的数据传送到存储器或者其他外围模块。此位设定是允许还是禁止 RXI 中断。 0: RIE 位允许或者禁止 RXI 和 ERI 中断 当 RIE 位是 1 时，将 RXI 和 ERI 中断送给 INTC 1: 当 RIE 位是 1 时，只将 ERI 中断送给 INTC
6 ~ 4	—	全 0	—	保留位 读写的值总是 0。

位	位名	初始值	R/W	说 明																				
3	SPB1IO	0	R/W	<p>串行端口的时钟端口输入 / 输出</p> <p>指定串行端口的 SCK 引脚的输入 / 输出。实际上在 SCK 引脚作为端口输出引脚并输出由 SPB1DT 位设定的值时，必须将 SCSSMR 的 <math>C/\bar{A}</math> 位和 SCSCR 的 CKE1、CKE0 位设定为 0。</p> <p>0: 不将 SPB1DT 位的值输出到 SCK 引脚 1: 将 SPB1DT 位的值输出到 SCK 引脚</p>																				
2	SPB1DT	不定	R/W	<p>串行端口的时钟端口数据</p> <p>指定串行端口的 SCK 引脚的输出数据。通过 SPB1IO 位指定输出是否有效（详细内容请参照 SPB1IO 位的说明）。输出时，SPB1DT 位的值被输出到 SCK 引脚。</p> <p>0: 输出数据为低电平 1: 输出数据为高电平</p>																				
1	SPB0IO	0	R/W	<p>串行端口的中止输出</p> <p>通过组合 SPB0DT 位和 SCSCR 的 TE 位，控制 TXD 引脚。</p>																				
0	SPB0DT	不定	R/W	<p>串行端口的中止数据</p> <p>通过组合 SPB0IO 位和 SCSCR 位的 TE 位，控制 TXD 引脚。 但是，必须通过引脚功能控制器（PFC）选择 TXD 引脚功能。</p> <table border="1" data-bbox="596 949 1425 1317"> <thead> <tr> <th>SCSCR 的 TE 位设定值</th> <th>SPB0IO 位设定值</th> <th>SPB0DT 位设定值</th> <th>TXD 引脚状态</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>*</td> <td>SPB0DT 输出无效状态（初始状态）</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>低电平输出</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>高电平输出</td> </tr> <tr> <td>1</td> <td>*</td> <td>*</td> <td>按照串行内核逻辑输出发送数据</td> </tr> </tbody> </table> <p>【注】 * Don't care</p>	SCSCR 的 TE 位设定值	SPB0IO 位设定值	SPB0DT 位设定值	TXD 引脚状态	0	0	*	SPB0DT 输出无效状态（初始状态）	0	1	0	低电平输出	0	1	1	高电平输出	1	*	*	按照串行内核逻辑输出发送数据
SCSCR 的 TE 位设定值	SPB0IO 位设定值	SPB0DT 位设定值	TXD 引脚状态																					
0	0	*	SPB0DT 输出无效状态（初始状态）																					
0	1	0	低电平输出																					
0	1	1	高电平输出																					
1	*	*	按照串行内核逻辑输出发送数据																					

## 15.3.9 串行方向控制寄存器 (SCSDCR)

SCSDCR 通过 DIR 位选择 LSB 先或者 MSB 先。在 8 位长的情况下, 不论那种串行通信模式都能选择 LSB 先或者 MSB 先。

位:	7	6	5	4	3	2	1	0
	-	-	-	-	DIR	-	-	-
初始值:	1	1	1	1	0	0	1	0
R/W:	R	R	R	R	R/W	R	R	R

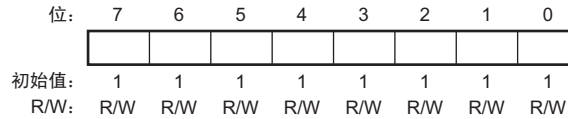
位	位名	初始值	R/W	说 明
7 ~ 4	—	全 1	R	保留位 读写的值总是 1。
3	DIR	0	R/W	数据的传送方向 选择串行或者并行转换格式。发送 / 接收格式在 8 位时有效。 0: 以 LSB 先发送 SCTDR 的内容 以 LSB 先将接收数据保存到 SCRDR 1: 以 MSB 先发送 SCTDR 的内容 以 MSB 先将接收数据保存到 SCRDR
2	—	0	R	保留位 读写的值总是 0。
1	—	1	R	保留位 读写的值总是 1。
0	—	0	R	保留位 读写的值总是 0。

### 15.3.10 位速率寄存器 (SCBRR)

SCBRR 是 8 位寄存器，根据串行模式寄存器 (SCSMR) 的 CKS1、CKS0 位选择的波特率发生器的运行时钟，设定串行发送和接收的位速率。

CPU 随时能读 SCBRR。

通过以下计算式求 SCBRR 的设定值：



(异步模式)

$$N = \frac{P_{\phi}}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

(时钟同步模式)

$$N = \frac{P_{\phi}}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: 位速率 (bit/s)

N: 波特率发生器的 SCBRR 设定值 ( $0 \leq N \leq 255$ )

(必须设定满足电特性的值)

$P_{\phi}$ : 外围模块的运行频率 (MHz)

n: 波特率发生器输入时钟 (n=0、1、2、3)

(n 和时钟的关系请参照表 15.3)。

表 15.3 SCSMR 的设定值

n	时钟	SCSMR 的设定值	
		CKS1	CKS0
0	$P_{\phi}$	0	0
1	$P_{\phi}/4$	0	1
2	$P_{\phi}/16$	1	0
3	$P_{\phi}/64$	1	1

通过以下计算式求异步模式的位速率误差：

$$\text{误差 (\%)} = \left\{ \frac{P_{\phi} \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

异步模式的 SCBRR 的设定例子如表 15.4 ~ 表 15.6 所示，时钟同步模式的 SCBRR 的设定例子如表 15.7 ~ 表 15.9 所示。

表 15.4 位速率的 SCBRR 的设定例子 (异步模式) (1)

位速率 (bit/s)	P <sub>φ</sub> (MHz)																	
	10			12			14			16			18			20		
	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)
110	2	177	-0.25	2	212	0.03	2	248	-0.17	3	70	0.03	3	79	-0.12	3	88	-0.25
150	2	129	0.16	2	155	0.16	2	181	0.16	2	207	0.16	2	233	0.16	3	64	0.16
300	2	64	0.16	2	77	0.16	2	90	0.16	2	103	0.16	2	116	0.16	2	129	0.16
600	1	129	0.16	1	155	0.16	1	181	0.16	1	207	0.16	1	233	0.16	2	64	0.16
1200	1	64	0.16	1	77	0.16	1	90	0.16	1	103	0.16	1	116	0.16	1	129	0.16
2400	0	129	0.16	0	155	0.16	0	181	0.16	0	207	0.16	0	233	0.16	1	64	0.16
4800	0	64	0.16	0	77	0.16	0	90	0.16	0	103	0.16	0	116	0.16	0	129	0.16
9600	0	32	-1.36	0	38	0.16	0	45	-0.93	0	51	0.16	0	58	-0.69	0	64	0.16
14400	0	21	-1.36	0	25	0.16	0	29	1.27	0	34	-0.79	0	38	0.16	0	42	0.94
19200	0	15	1.73	0	19	-2.34	0	22	-0.93	0	25	0.16	0	28	1.02	0	32	-1.36
28800	0	10	-1.36	0	12	0.16	0	14	1.27	0	16	2.12	0	19	-2.34	0	21	-1.36
31250	0	9	0.00	0	11	0.00	0	13	0.00	0	15	0.00	0	17	0.00	0	19	0.00
38400	0	7	1.73	0	9	-2.34	0	10	3.57	0	12	0.16	0	14	-2.34	0	15	1.73

表 15.5 位速率的 SCBRR 的设定例子 (异步模式) (2)

位速率 (bit/s)	P <sub>φ</sub> (MHz)																	
	22			24			26			28			30			32		
	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)
110	3	97	-0.35	3	106	-0.44	3	114	0.36	3	123	0.23	3	132	0.13	3	141	0.03
150	3	71	-0.54	3	77	0.16	3	84	-0.43	3	90	0.16	3	97	-0.35	3	103	0.16
300	2	142	0.16	2	155	0.16	2	168	0.16	2	181	0.16	2	194	0.16	2	207	0.16
600	2	71	-0.54	2	77	0.16	2	84	-0.43	2	90	0.16	2	97	-0.35	2	103	0.16
1200	1	142	0.16	1	155	0.16	1	168	0.16	1	181	0.16	1	194	0.16	1	207	0.16
2400	1	71	-0.54	1	77	0.16	1	84	-0.43	1	90	0.16	1	97	-0.35	1	103	0.16
4800	0	142	0.16	0	155	0.16	0	168	0.16	0	181	0.16	0	194	0.16	0	207	0.16
9600	0	71	-0.54	0	77	0.16	0	84	-0.43	0	90	0.16	0	97	-0.35	0	103	0.16
14400	0	47	-0.54	0	51	0.16	0	55	0.76	0	60	-0.39	0	64	0.16	0	68	0.64
19200	0	35	-0.54	0	38	0.16	0	41	0.76	0	45	-0.93	0	48	-0.35	0	51	0.16
28800	0	23	-0.54	0	25	0.16	0	27	0.76	0	29	1.27	0	32	-1.36	0	34	-0.79
31250	0	21	0.00	0	23	0.00	0	25	0.00	0	27	0.00	0	29	0.00	0	31	0.00
38400	0	17	-0.54	0	19	-2.34	0	20	0.76	0	22	-0.93	0	23	1.73	0	25	0.16

表 15.6 位速率的 SCBRR 的设定例子 (异步模式) (3)

位速率 (bit/s)	P $\phi$ (MHz)											
	34			36			38			40		
	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)
110	3	150	-0.05	3	159	-0.12	3	168	-0.19	3	177	-0.25
150	3	110	-0.29	3	116	0.16	3	123	-0.24	3	129	0.16
300	2	220	0.16	2	233	0.16	2	246	0.16	3	64	0.16
600	2	110	-0.29	2	116	0.16	2	123	-0.24	2	129	0.16
1200	1	220	0.16	1	233	0.16	1	246	0.16	2	64	0.16
2400	1	110	-0.29	1	116	0.16	1	123	-0.24	1	129	0.16
4800	0	220	0.16	0	233	0.16	0	246	0.16	1	64	0.16
9600	0	110	-0.29	0	116	0.16	0	123	-0.24	0	129	0.16
14400	0	73	-0.29	0	77	0.16	0	81	0.57	0	86	-0.22
19200	0	54	0.62	0	58	-0.69	0	61	-0.24	0	64	0.16
28800	0	36	-0.29	0	38	0.16	0	40	0.57	0	42	0.94
31250	0	33	0.00	0	35	0.00	0	37	0.00	0	39	0.00
38400	0	27	-1.18	0	28	1.02	0	30	-0.24	0	32	-1.36

表 15.7 位速率的 SCBRR 的设定例子 (时钟同步模式) (1)

位速率 (bit/s)	P $\phi$ (MHz)											
	10		12		14		16		18		20	
	n	N	n	N	n	N	n	N	n	N	n	N
250	3	155	3	187	3	218	3	249				
500	3	77	3	93	3	108	3	124	3	140	3	155
1000	2	155	2	187	2	218	2	249	3	69	3	77
2500	1	249	2	74	2	87	2	99	2	112	2	124
5000	1	124	1	149	1	174	1	199	1	224	1	249
10000	0	249	1	74	1	87	1	99	1	112	1	124
25000	0	99	0	119	0	139	0	159	0	179	0	199
50000	0	49	0	59	0	69	0	79	0	89	0	99
100000	0	24	0	29	0	34	0	39	0	44	0	49
250000	0	9	0	11	0	13	0	15	0	17	0	19
500000	0	4	0	5	0	6	0	7	0	8	0	9
1000000	—	—	0	2	—	—	0	3	—	—	0	4
2500000	0	0*	—	—	—	—	—	—	—	—	0	1
5000000			—	—	—	—	—	—	—	—	0	0*

表 15.8 位速率的 SCBRR 的设定例子 (时钟同步模式) (2)

位速率 (bit/s)	P $\phi$ (MHz)											
	22		24		26		28		30		32	
	n	N	n	N	n	N	n	N	n	N	n	N
250												
500	3	171	3	187	3	202	3	218	3	233	3	249
1000	3	85	3	93	3	101	3	108	3	116	3	124
2500	2	137	2	149	2	162	2	174	2	187	2	199
5000	2	68	2	74	2	80	2	87	2	93	2	99
10000	1	137	1	149	1	162	1	174	1	187	1	199
25000	0	219	0	239	1	64	1	69	1	74	1	79
50000	0	109	0	119	0	129	0	139	0	149	0	159
100000	0	54	0	59	0	64	0	69	0	74	0	79
250000	0	21	0	23	0	25	0	27	0	29	0	31
500000	0	10	0	11	0	12	0	13	0	14	0	15
1000000	—	—	0	5	—	—	0	6	—	—	0	7
2500000	—	—	—	—	—	—	—	—	0	2	—	—
5000000	—	—	—	—	—	—	—	—	—	—	—	—

表 15.9 位速率的 SCBRR 的设定例子 (时钟同步模式) (3)

位速率 (bit/s)	P $\phi$ (MHz)							
	34		36		38		40	
	n	N	n	N	n	N	n	N
250								
500								
1000	3	132	3	140	3	147	3	155
2500	2	212	2	224	2	237	2	249
5000	2	105	2	112	2	118	2	124
10000	1	212	1	224	1	237	1	249
25000	1	84	1	89	1	94	1	99
50000	0	169	0	179	0	189	0	199
100000	0	84	0	89	0	94	0	99
250000	0	33	0	35	0	37	0	39
500000	0	16	0	17	0	18	0	19
1000000	—	—	0	8	—	—	0	9
2500000	—	—	—	—	—	—	0	3
5000000	—	—	—	—	—	—	0	1

【注】 必须尽量将误差设定在 1% 以内。

【符号说明】

空白栏: 不能设定。

— : 能设定, 但是会出现误差。

\* : 不能连续发送和接收。



使用波特率发生器时异步模式的各频率的最大位速率如表 15.10 所示。另外，输入外部时钟时的最大位速率如表 15.11 和表 15.12 所示。

表 15.10 使用波特率发生器时的各频率的最大位速率（异步模式）

P $\phi$ (MHz)	最大位速率 (bit/s)	设定值	
		n	N
10	312500	0	0
12	375000	0	0
14	437500	0	0
16	500000	0	0
18	562500	0	0
20	625000	0	0
22	687500	0	0
24	750000	0	0
26	812500	0	0
28	875000	0	0
30	937500	0	0
32	1000000	0	0
34	1062500	0	0
36	1125000	0	0
38	1187500	0	0
40	1250000	0	0

表 15.11 输入外部时钟时的最大位速率 (异步模式)

P $\phi$ (MHz)	外部输入时钟 (MHz)	最大位速率 (bit/s)
10	2.5000	156250
12	3.0000	187500
14	3.5000	218750
16	4.0000	250000
18	4.5000	281250
20	5.0000	312500
22	5.5000	343750
24	6.0000	375000
26	6.5000	406250
28	7.0000	437500
30	7.5000	468750
32	8.0000	500000
34	8.5000	531250
36	9.0000	562500
38	9.5000	593750
40	10.0000	625000

表 15.12 输入外部时钟时的最大位速率 (时钟同步模式)

P $\phi$ (MHz)	外部输入时钟 (MHz)	最大位速率 (bit/s)
10	1.6667	1666666.7
12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0
20	3.3333	3333333.3
22	3.6667	3666666.7
24	4.0000	4000000.0
26	4.3333	4333333.3
28	4.6667	4666666.7
30	5.0000	5000000.0
32	5.3333	5333333.3
34	5.6667	5666666.7
36	6.0000	6000000.0
38	6.3333	6333333.3
40	6.6667	6666666.7

## 15.4 运行说明

### 15.4.1 概要

SCI 能通过异步模式 (以字符为单位边同步边通信) 和时钟同步模式 (通过时钟脉冲边同步边通信) 的 2 种方式进行串行通信。

如表 15.13 所示, 通过串行模式寄存器 (SCSMR) 选择异步模式、时钟同步模式和发送格式。另外, 如表 15.14 所示, SCI 的时钟源由 SCSMR 的 C/A 位和串行控制寄存器 (SCSCR) 的 CKE1、CKE0 位的组合决定。

#### (1) 异步模式

- 数据长: 可从 7 位或者 8 位中选择
- 可选择奇偶校验的附加、1 位或 2 位的停止位的附加 (根据这些组合, 决定发送 / 接收格式和字符长)
- 在接收时, 能检测帧错误、奇偶校验错误、溢出错误和中止
- SCI 的时钟源: 可从内部时钟或者外部时钟中选择  
 选择内部时钟: 能通过波特率发生器的时钟运行 SCI, 输出频率为 16 倍位速率的时钟  
 选择外部时钟: 需要输入频率为 16 倍位速率的时钟 (不使用内部波特率发生器)

## (2) 时钟同步模式

- 发送 / 接收格式: 固定为 8 位数据
- 在接收时, 能检测溢出错误
- SCI 的时钟源: 可从内部时钟或者外部时钟中选择  
 选择内部时钟: 通过波特率发生器的时钟运行 SCI, 将同步时钟输出到外部  
 选择外部时钟: 不使用内部波特率发生器而通过输入的同步时钟运行

表 15.13 SCSMR 的设定值和串行发送 / 接收格式

SCSMR 的设定值				模式	SCI 的发送 / 接收格式		
bit7	bit6	bit5	bit3		数据长	奇偶校验位	停止位长
C/ $\bar{A}$	CHR	PE	STOP				
0	0	0	0	异步模式	8 位数据	无	1 位
			1				2 位
		1	0			有	1 位
			1				2 位
	1	0	0		7 位数据	无	1 位
			1				2 位
		1	0			有	1 位
			1				2 位
1	x	x	x	时钟同步模式	8 位数据	无	无

【符号说明】 x : Don't care

表 15.14 SCSMR、SCSCR 的设定和 SCI 时钟源的选择

SCSMR	SCSCR 的设定值		模式	时钟源	SCK 引脚的功能
bit7	bit1	bit0			
C/ $\bar{A}$	CKE1	CKE0			
0	0	0	异步模式	内部	SCI 不使用 SCK 引脚
		1			输出位速率的 16 倍频率时钟
	1	0		外部	输入位速率的 16 倍频率时钟
		1			
1	0	0	时钟同步模式	内部	输出同步时钟
		1			
	1	0		外部	输入同步时钟
		1			

### 15.4.2 异步模式中的运行

异步模式是以字符为单位边同步边进行串行通信的模式，将表示开始通信的起始位和表示结束通信的停止位附加到数据后进行字符的发送和接收。

在 SCI 内部发送部和接收部独立，所以能进行全双工通信。另外，发送部和接收部都为双缓冲结构，所以能在发送和接收中读写数据，进行连续的发送和接收。

异步串行通信的一般格式如图 15.2 所示。

在异步串行通信中，通信线路通常保持为标记状态（高电平）。SCI 监视通信线路，将空闲状态（低电平）视为起始位，开始串行通信。

以起始位（低电平）开始，数据（LSB 先：从最低位开始）、奇偶校验位（高/低电平），最后为停止位（高电平）的顺序构成串行通信的 1 个字符。

在异步模式中，SCI 在接收时通过起始位的下降沿取得同步。另外，因为 SCI 在 1 个为 16 倍位速率频率的第 8 个时钟采样数据，所以在各位的中央取得通信数据。

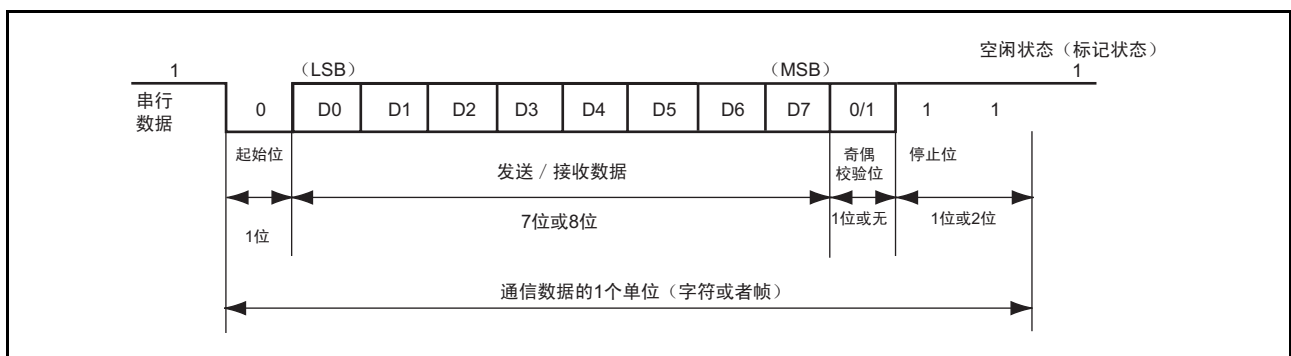


图 15.2 异步通信的数据格式  
(8 位数据 / 有奇偶校验 / 2 个停止位的例子)

(1) 发送 / 接收的格式

在异步模式中，能设定的发送 / 接收格式如表 15.15 所示。

发送 / 接收格式有 12 种，能通过设定串行模式寄存器 (SCSMR) 进行选择。

表 15.15 串行发送 / 接收格式 (异步模式)

SCSMR 的设定				串行发送 / 接收格式和帧长												
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	x	1	0	S	8位数据								MPB	STOP		
0	x	1	1	S	8位数据								MPB	STOP	STOP	
1	x	1	0	S	7位数据							MPB	STOP			
1	x	1	1	S	7位数据							MPB	STOP	STOP		

【符号说明】

- S : 起始位
- STOP : 停止位
- P : 奇偶校验位
- MPB : 多处理器位
- x : Don't care

(2) 时钟

能通过设定 SCSSMR 的 C/A 位和串行控制寄存器 (SCSCR) 的 CKE1、CKE0 位, 从内部波特率发生器生成的内部时钟或者 SCK 引脚输入的外部时钟中选择 SCI 的发送和接收时钟。有关 SCI 时钟源的选择请参照表 15.14。

在将外部时钟输入到 SCK 引脚时, 输入频率是 16 倍位速率的时钟。

在通过内部时钟运行时, 能从 SCK 引脚输出时钟。此时, 输出的时钟频率为位速率的 16 倍。

(3) 数据的发送和接收运行

• SCI 的初始化 (异步模式)

在发送和接收数据前, 必须首先将 SCSCR 的 TE 位和 RE 位清 0, 然后按照以下顺序初始化 SCI。

在更改运行模式和通信格式等时, 也必须在将 TE 位和 RE 位清 0 后按照以下步骤进行。如果将 TE 位清 0, TDRE 标志就被置 1, 发送移位寄存器 (SCTSR) 被初始化。必须注意: 即使将 RE 位清 0, 也保持 RDRF、PER、FER、ORER 的各标志和接收数据寄存器 (SCRDR) 的内容。在使用外部时钟时, 不能在含有初始化的运行中停止时钟, 否则运行不稳定。

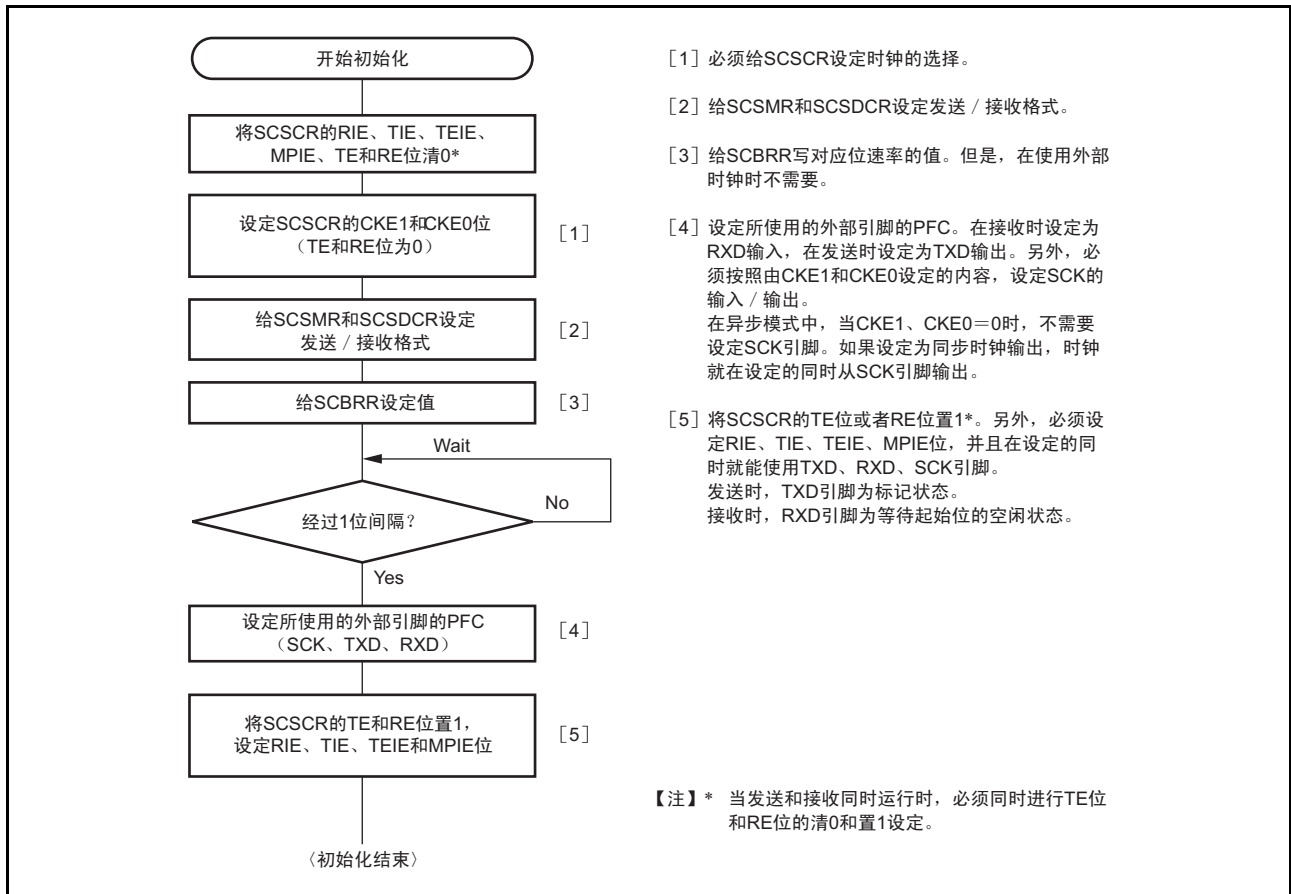
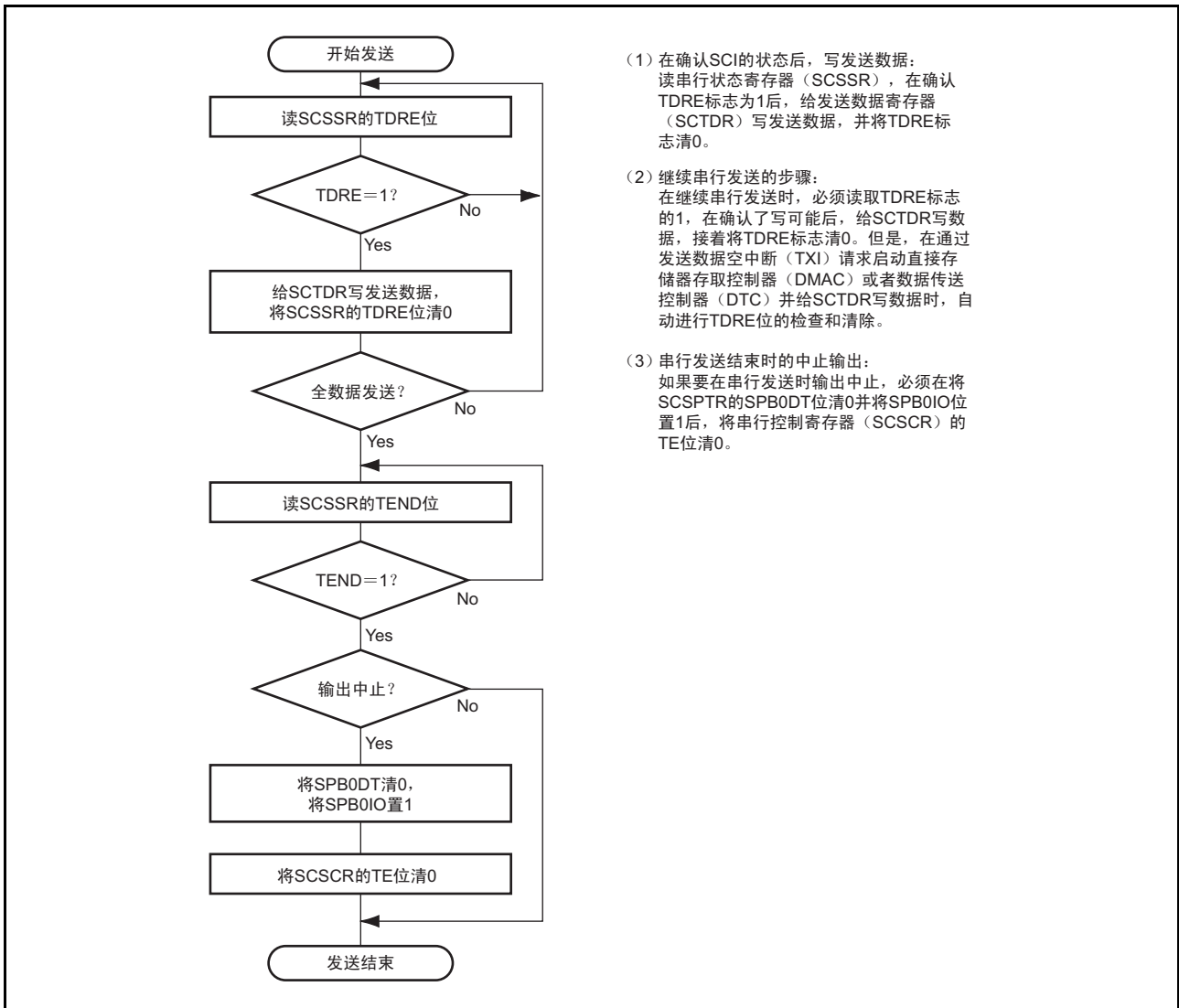


图 15.3 SCI 的初始化流程图例子

• 串行数据的发送 (异步模式)

串行发送的流程图例子如图 15.4 所示。

在将 SCI 设定为可发送的运行状态后, 必须按照以下步骤发送串行数据。



- (1) 在确认SCI的状态后, 写发送数据:  
读串行状态寄存器 (SCSSR), 在确认 TDRE 标志为 1 后, 给发送数据寄存器 (SCTDR) 写发送数据, 并将 TDRE 标志清 0。
- (2) 继续串行发送的步骤:  
在继续串行发送时, 必须读取 TDRE 标志的 1, 在确认了写可能后, 给 SCTDR 写数据, 接着将 TDRE 标志清 0。但是, 在通过发送数据空中断 (TXI) 请求启动直接存储器存取控制器 (DMAC) 或者数据传送控制器 (DTC) 并给 SCTDR 写数据时, 自动进行 TDRE 位的检查和清除。
- (3) 串行发送结束时的中止输出:  
如果要在串行发送时输出中止, 必须在将 SCSPTR 的 SPB0DT 位清 0 并将 SPB0IO 位置 1 后, 将串行控制寄存器 (SCSCR) 的 TE 位清 0。

图 15.4 串行发送的流程图例子



串行发送时的 SCI 运行如下：

1. SCI 监视串行状态寄存器 (SCSSR) 的 TDRE 标志，如果该位为 0，就认为已将数据写到发送数据寄存器 (SCTDR)，将数据从 SCTDR 传送到发送移位寄存器 (SCTSR)。
2. 在数据从 SCTDR 传送到 SCTSR 后，将 TDRE 标志设定为 1，开始发送。  
此时，如果串行控制寄存器 (SCSCR) 的 TIE 位已被置 1，就产生发送数据空的中断 (TXI) 请求。

按照以下顺序从 TXD 引脚送出串行发送数据：

- (a) 起始位：输出 1 位的 0。
  - (b) 发送数据：从 LSB 按顺序输出 8 位或者 7 位的数据。
  - (c) 奇偶校验位或者多处理器位：输出 1 位的奇偶校验位（偶校验或者奇校验）或者 1 位的多处理器位。  
另外，也能选择不输出奇偶校验位或者多处理器位的格式。
  - (d) 停止位：输出 1 位或者 2 位的 1（停止位）。
  - (e) 标记状态：在送出开始下一次发送的起始位前连续输出 1。
3. SCI 在送出停止位时检查 TDRE 标志。

如果 TDRE 标志为 0，就将数据从 SCTDR 传送到 SCTSR，在送出停止位后，开始下一帧的串行发送。  
如果 TDRE 标志为 1，就将串行状态寄存器 (SCSSR) 的 TEND 位设定为 1，在送出停止位后变为输出 1 的标记状态。此时，如果 SCSCR 的 TEIE 标志已被置 1，就产生 TEI 中断请求。

异步模式发送时的运行例子如图 15.5 所示。

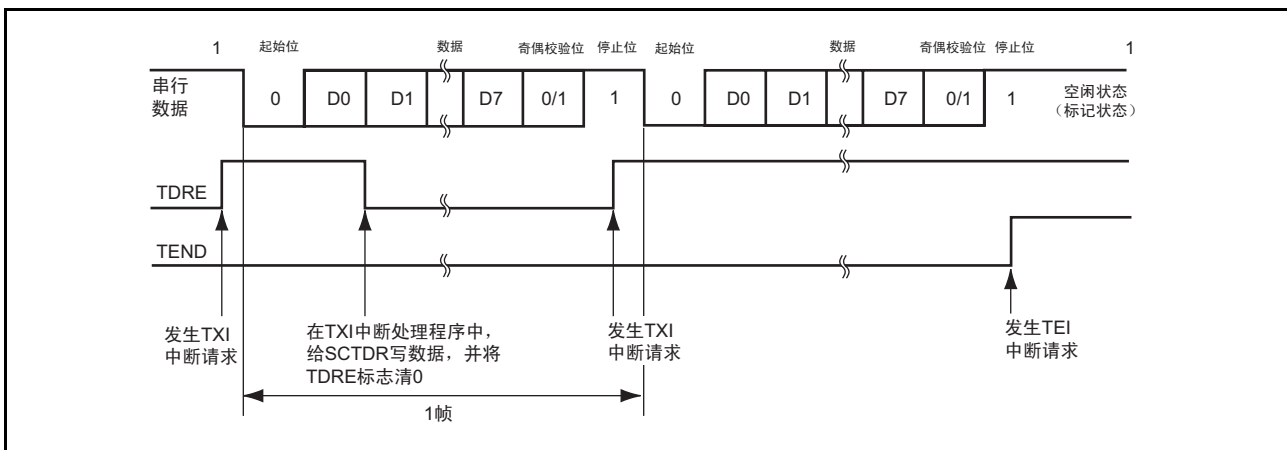
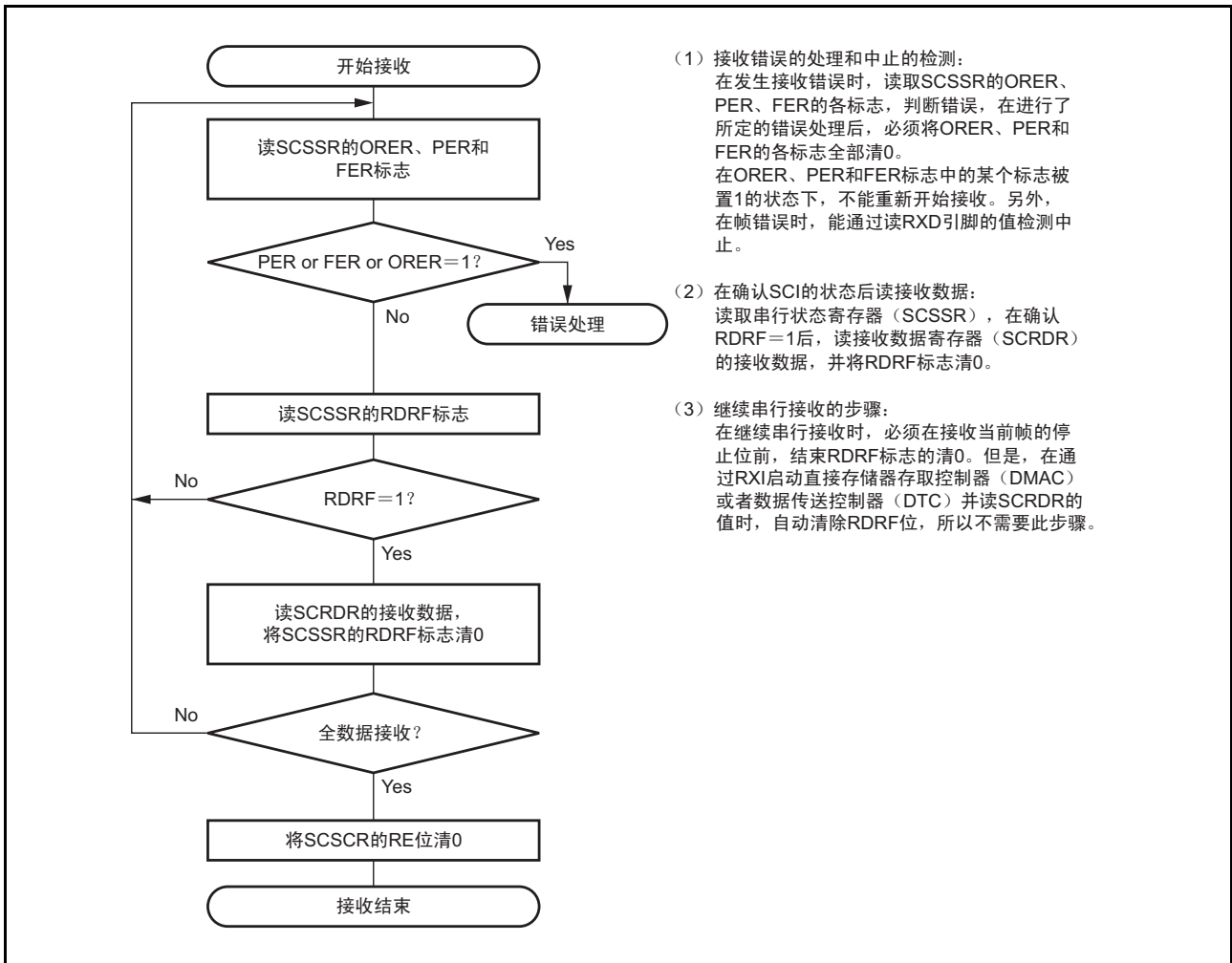


图 15.5 异步模式发送时的运行例子  
(8 位数据 / 有奇偶校验 / 1 个停止位的例子)

• 串行数据的接收 (异步模式)

串行接收的流程图例子如图 15.6 和图 15.7 所示。

在将 SCI 设定为可接受的运行状态后, 必须按照以下步骤接收串行数据。



- (1) 接收错误的处理和中止的检测:  
在发生接收错误时, 读取SCSSR的ORER、PER、FER的各标志, 判断错误, 在进行了所定的错误处理后, 必须将ORER、PER和FER的各标志全部清0。  
在ORER、PER和FER标志中的某个标志被置1的状态下, 不能重新开始接收。另外, 在帧错误时, 能通过读RXD引脚的值检测中止。
- (2) 在确认SCI的状态后读接收数据:  
读取串行状态寄存器 (SCSSR), 在确认RDRF=1后, 读接收数据寄存器 (SCRDR) 的接收数据, 并将RDRF标志清0。
- (3) 继续串行接收的步骤:  
在继续串行接收时, 必须在接收当前帧的停止位前, 结束RDRF标志的清0。但是, 在通过RXI启动直接存储器存取控制器 (DMAC) 或者数据传送控制器 (DTC) 并读SCRDR的值时, 自动清除RDRF位, 所以不需要此步骤。

图 15.6 串行接收的流程图例子 (1)

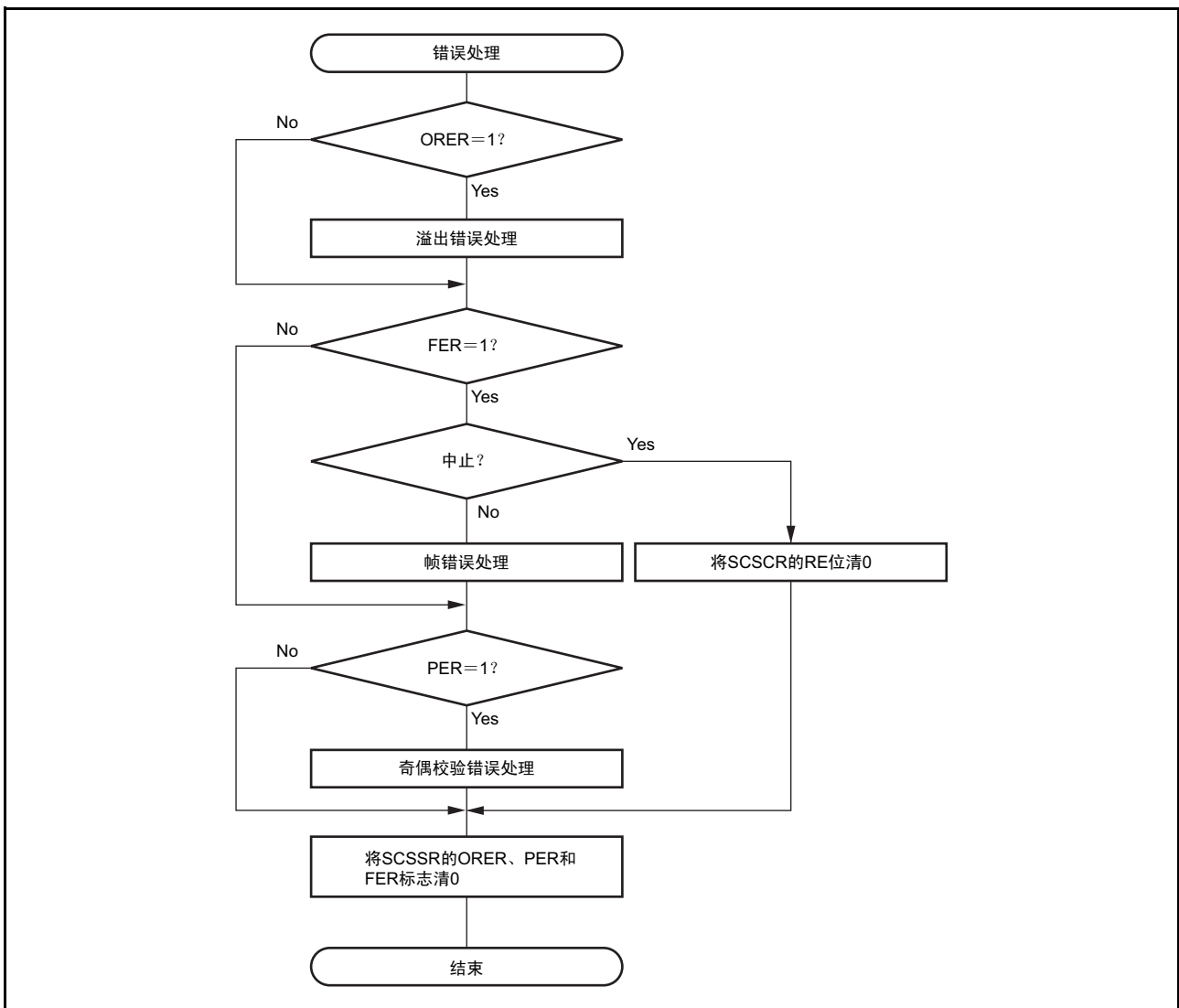


图 15.7 串行接收的流程图例子 (2)

接收时的 SCI 运行如下：

1. SCI 监视通信线路，如果检测到起始位的 0，就与内部同步开始接收。
2. 按照从 SCRSR 的 LSB 到 MSB 的顺序保存已接收的数据。
3. 接收奇偶校验位和停止位。

在接收后，SCI 进行以下检查：

- (a) 奇偶校验的检查：检查接收数据的 1 的个数是否为串行模式寄存器 (SCSMR) 的  $\overline{O/E}$  位设定的偶校验或奇校验。
- (b) 停止位的检查：检查停止位是否为 1。  
但是，在 2 个停止位的情况下，只检查第 1 位的停止位。
- (c) 状态检查：检查 RDRF 标志是否为 0，即是否为能将接收数据从接收移位寄存器 (SCRSR) 传送到 SCRDR 的状态。

在以上检查全部通过后，将 RDRF 标志设定为 1，并且将接收数据保存到 SCRDR。

如果在错误检查时发生接收错误，就进行如表 15.16 所示的运行。

**【注】** 在发生接收错误的状态下，不能进行以后的接收。另外，因为在接收时 RDRF 标志不被置 1，所以必将错误标志清 0。

4. 在 RDRF 标志为 1 时，如果 SCSCR 的 RIE 位已被置 1，就产生接收数据满的中断 (RXI) 请求。  
另外，在 ORER、PER 和 FER 标志中的某个标志为 1 时，如果 SCSCR 的 RIE 位已被置 1，就产生接收错误的中断 (ERI) 请求。

表 15.16 接收错误和发生条件

接收错误	略称	发生条件	数据传送
溢出错误	ORER	在 SCSSR 的 RDRF 标志被置 1 的状态下结束下一个数据接收时	不将接收数据从 SCRSR 传送到 SCRDR。
帧错误	FER	当停止位为 0 时	将接收数据从 SCRSR 传送到 SCRDR。
奇偶校验错误	PER	当 SCSMR 设定的偶校验 / 奇校验和接收的数据不同时	将接收数据从 SCRSR 传送到 SCRDR。

异步模式接收时的运行例子如图 15.8 所示。

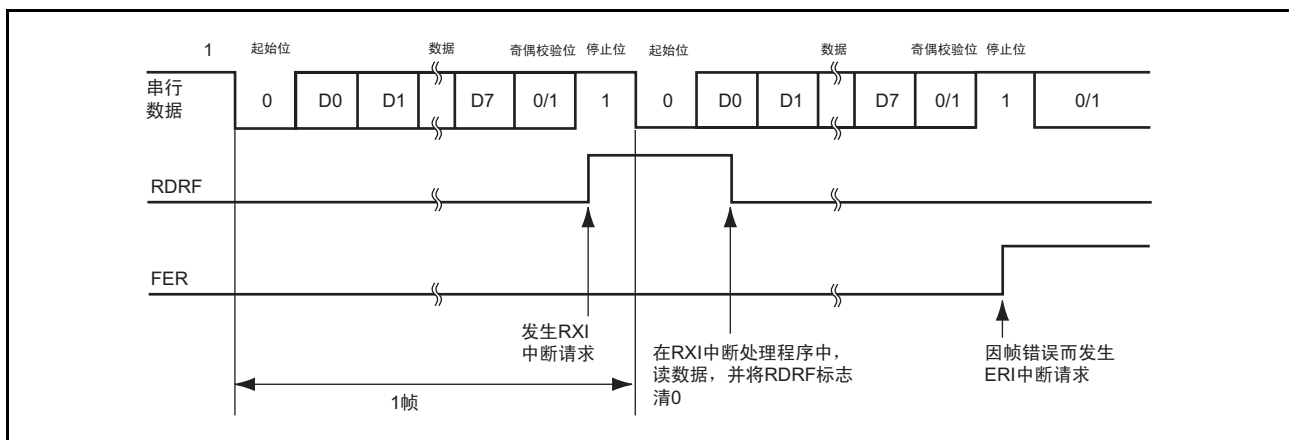


图 15.8 SCI 接收时的运行例子 (8 位数据 / 有奇偶校验 / 1 个停止位的例子)

### 15.4.3 时钟同步模式中的运行

时钟同步模式是与时钟脉冲同步进行数据发送和接收的模式，适用于高速串行通信。

在 SCI 内部发送部和接收部独立，所以能通过共享时钟进行全双工通信。

另外，发送部和接收部都为双缓冲结构，所以能在发送和接收中读写数据，进行连续的发送和接收。

时钟同步串行通信的一般格式如图 15.9 所示。

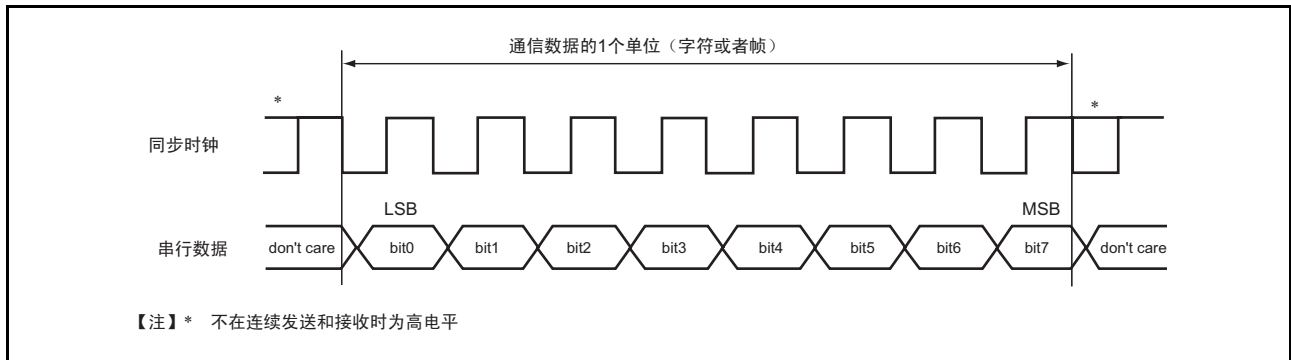


图 15.9 时钟同步通信的数据格式

在时钟同步串行通信中，在同步时钟的下降沿和下一个下降沿之间将数据输出到通信线路，保证在同步时钟的上升沿数据的有效。

以数据的 LSB（最初）到 MSB（最后）的顺序输出串行通信的 1 个字符，输出 MSB 后的通信线路状态保持 MSB 的状态。

在时钟同步模式中，SCI 与同步时钟的上升沿同步接收数据。

#### (1) 发送 / 接收格式

发送 / 接收格式固定为 8 位数据。

不能附加奇偶校验位。

#### (2) 时钟

能通过设定 SCSMR 的  $\overline{C/A}$  位和 SCSCR 的 CKE1、CKE0 位，选择内部波特率发生器生成的内部时钟或者 SCK 引脚输入的外部同步时钟。有关 SCI 时钟源的选择请参照表 15.14。

在通过内部时钟运行时，从 SCK 引脚输出同步时钟。

同步时钟在发送和接收 1 个字符时输出 8 个脉冲，而在不进行发送和接收时固定为高电平。但是，只进行接收时，就会发生溢出错误或输出同步时钟直到 RE 位清 0 为止。如果进行 n 个字符数的接收时，必须将时钟源作为外部时钟。在使用内部时钟时，必须按照先设定 RE=1 且 TE=1，再在发送 n 个字符数的虚拟数据的同时接收 n 个字符数的顺序。

#### (3) 数据的发送和接收运行

##### • SCI 的初始化（时钟同步模式）

在发送和接收数据前，必须在将串行控制寄存器（SCSCR）的 TE 和 RE 位清 0 后，按照以下步骤初始化 SCI。

在更改模式和通信格式等时，必须在将 TE 和 RE 位清 0 后按照以下步骤进行。如果将 TE 位清 0，TDRE 标志就被置 1，发送移位寄存器（SCTSR）被初始化。

必须注意：即使将 RE 位清 0，也保持 RDRF、PER、FER、ORER 的各标志和接收数据寄存器（SCRDR）的内容。

SCI 的初始化流程图例子如图 15.10 所示。

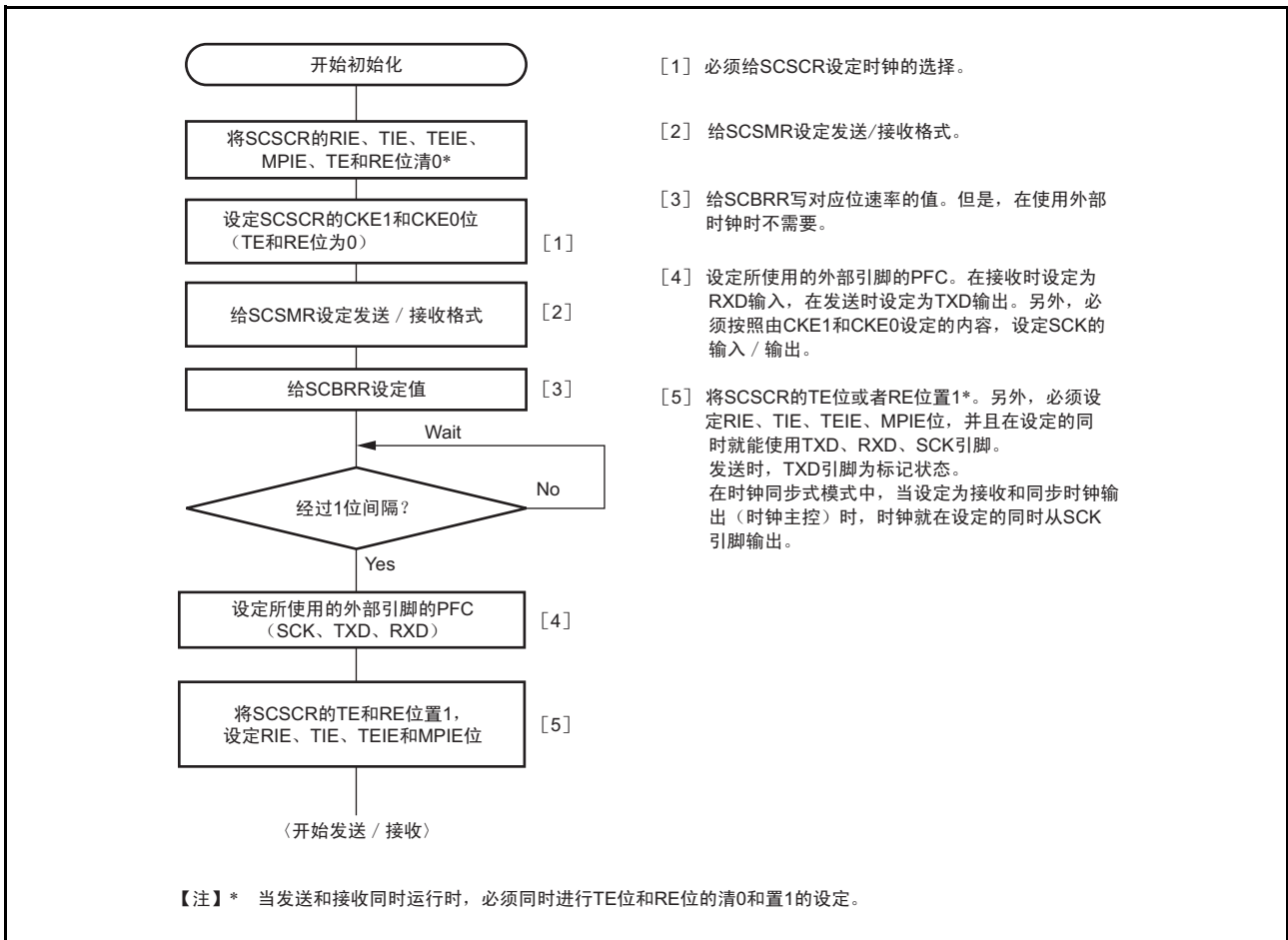


图 15.10 SCI 的初始化流程图例子

### • 串行数据的发送 (时钟同步模式)

串行发送的流程图例子如图 15.11 所示。

在将 SCI 设定为可发送的运行状态后, 必须按照以下步骤发送串行数据。

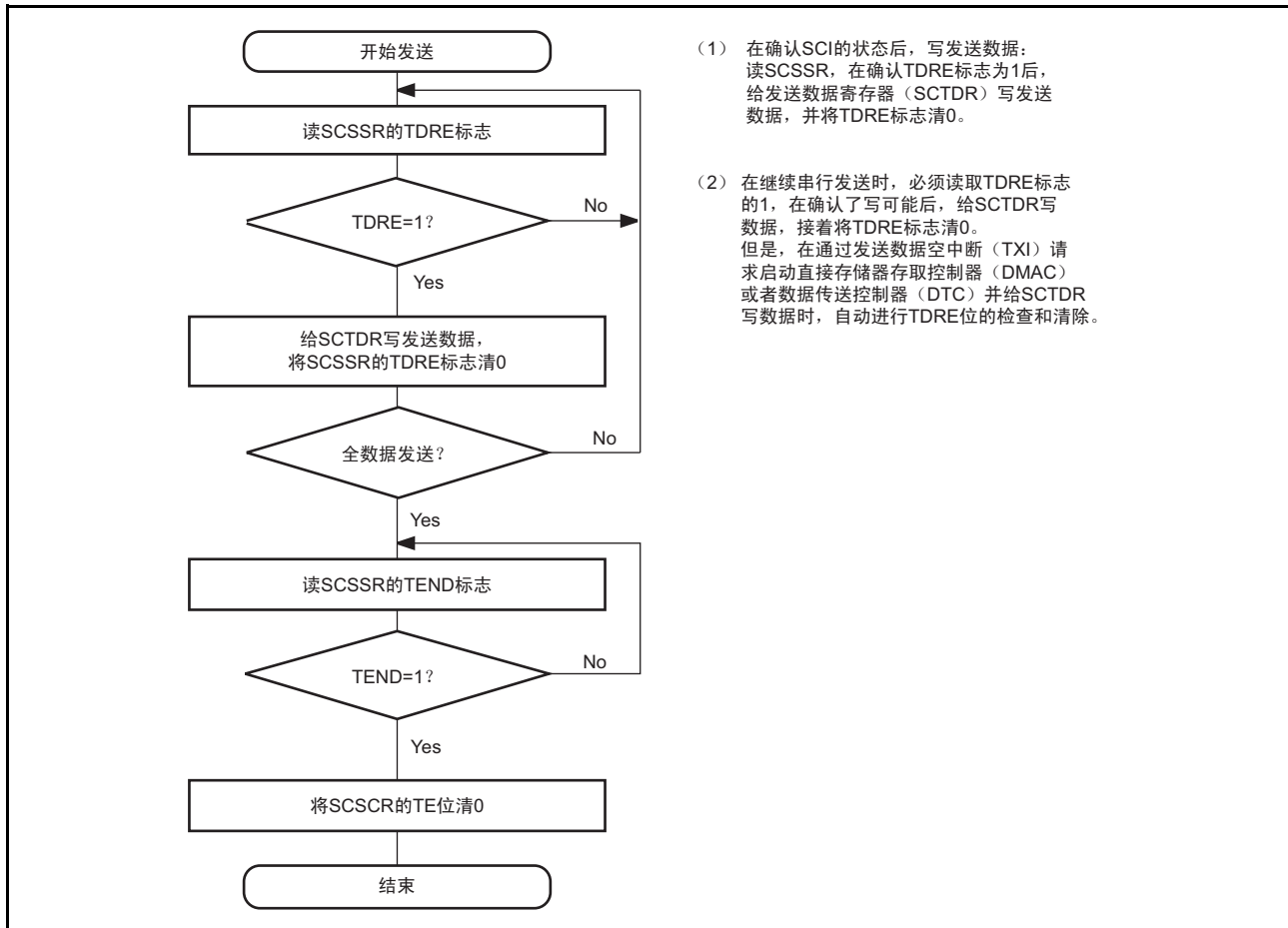


图 15.11 串行发送的流程图例子

串行发送时的 SCI 运行如下:

1. SCI 监视串行状态寄存器 (SCSSR) 的 TDRE 标志, 如果为 0, 就认为已将数据写到发送数据寄存器 (SCTDR), 将数据从 SCTDR 传送到发送移位寄存器 (SCTSR)。
2. 在数据从 SCTDR 传送到 SCTSR 后, 将 TDRE 标志设定为 1, 开始发送。  
此时, 如果串行控制寄存器 (SCSCR) 的发送数据空的中断允许位 (TIE) 已被置 1, 就产生发送数据空的中断 (TXI) 请求。  
在设定为时钟输出模式时, SCI 输出同步时钟的 8 个脉冲。  
在设定为外部时钟时, 与输入时钟同步输出数据。  
按照 LSB (bit0) ~ MSB (bit7) 的顺序从 TXD 引脚送出串行发送数据。
3. SCI 在送出 MSB (bit7) 时检查 TDRE 标志。  
如果 TDRE 标志为 0, 就将数据从 SCTDR 传送到 SCTSR, 开始下一帧的串行发送。  
如果 TDRE 标志为 1, 就将串行状态寄存器 (SCSSR) 的 TEND 标志设定为 1, 在送出 MSB (bit7) 后发送数据的引脚 (TXD 引脚) 保持状态。  
此时, 如果 SCSCR 的发送结束的中断允许位 (TEIE) 已被置 1, 就产生发送结束的中断请求 (TEI)。
4. 结束串行发送后, SCK 引脚固定为高电平。

SCI 发送时的运行例子如图 15.12 所示。

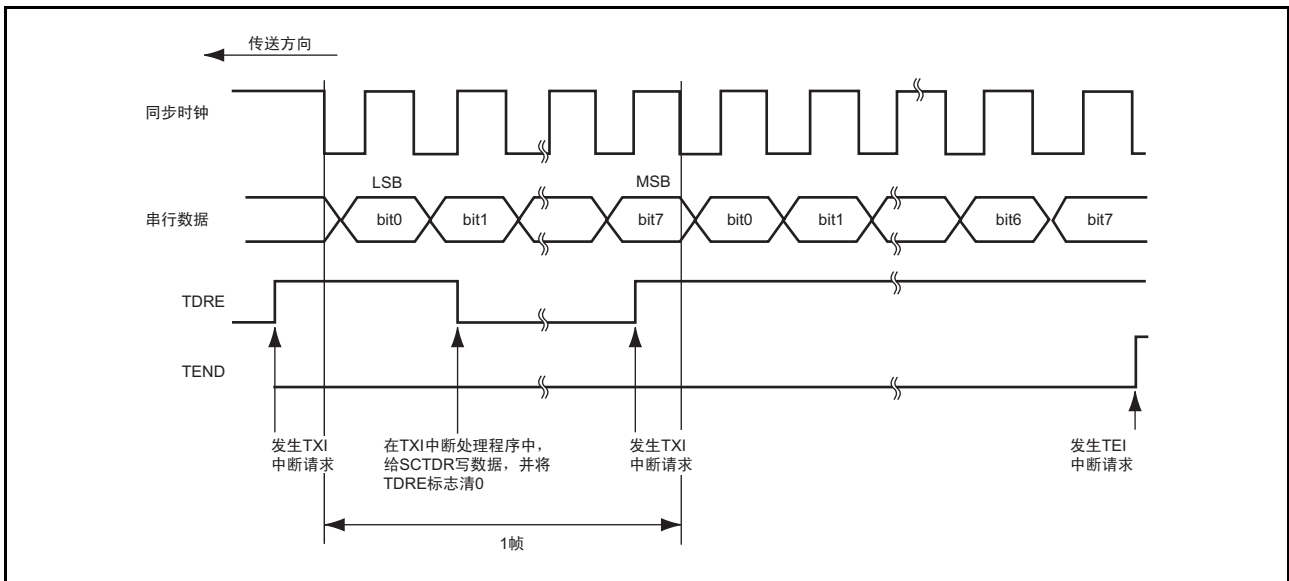


图 15.12 SCI 发送时的运行例子



• 串行数据的接收 (时钟同步模式)

串行接收的流程图例子如图 15.13 和图 15.14 所示。

在将 SCI 设定为可接收的运行状态后, 必须按照以下步骤接收串行数据。

在将运行模式从异步模式转换为时钟同步模式时, 必须确认 ORER、PER、FER 的各标志是否已被清 0。如果 FER、PER 标志已被置 1, RDRF 标志就不能置位, 并且不能发送和接收。

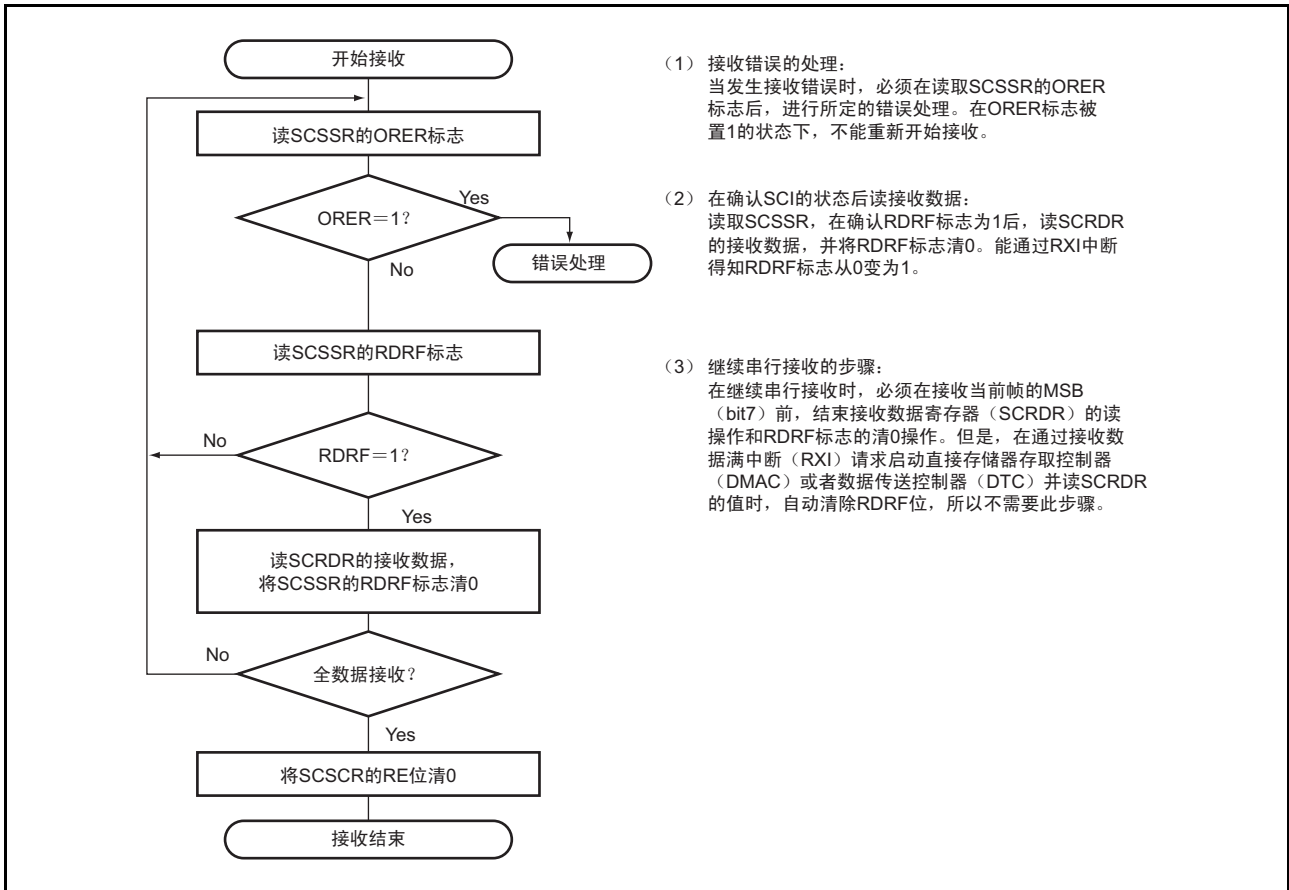


图 15.13 串行数据的接收流程图例子 (1)

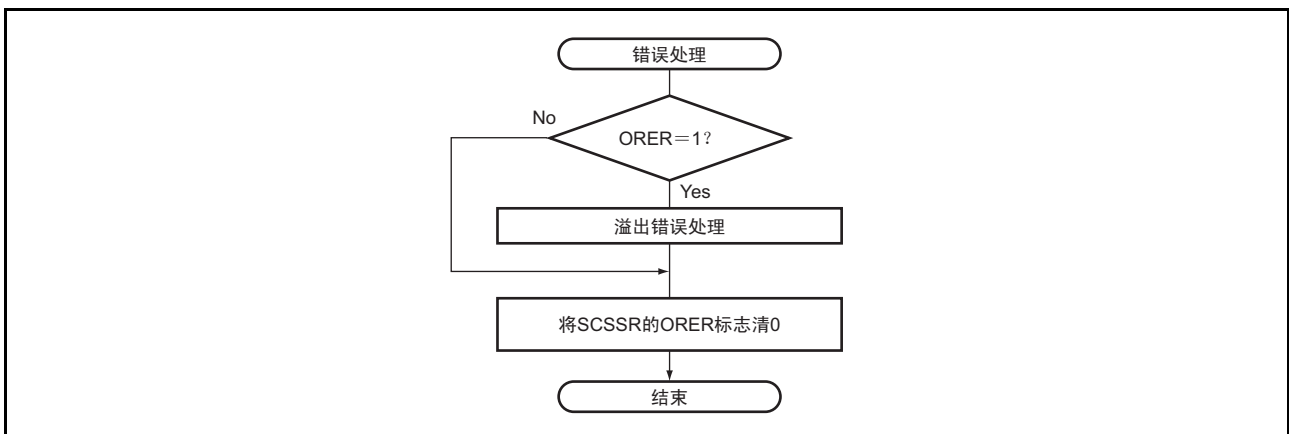


图 15.14 串行数据接收的流程图例子 (2)

接收时的 SCI 运行如下：

1. SCI 与同步时钟的输入或输出同步，开始接收。
2. 按照接收移位寄存器 (SCRSR) 的 LSB 到 MSB 的顺序保存接收的数据。  
在接收后，SCI 检查 RDRF 标志是否为 0，即是否为能将接收数据从 SCRSR 传送到接收数据寄存器 (SCRDR) 的状态。  
在此检查通过后，将 RDRF 标志设定为 1，并且将接收数据保存到 SCRDR。  
如果在错误检查时发生接收错误，就进行如表 15.16 所示的运行，在此状态下不能进行以后的发送和接收。  
另外，因为在接收时 RDRF 标志不被置 1，所以必须将此标志清 0。
3. 在 RDRF 标志为 1 时，如果串行控制寄存器 (SCSCR) 的 RIE 位已被置 1，就产生接收数据满的中断 (RXI) 请求。  
另外，在 ORER 标志为 1 时，如果 SCSCR 的 RIE 位已被置 1，就产生接收错误的中断 (ERI) 请求。

SCI 接收时的运行例子如图 15.15 所示。

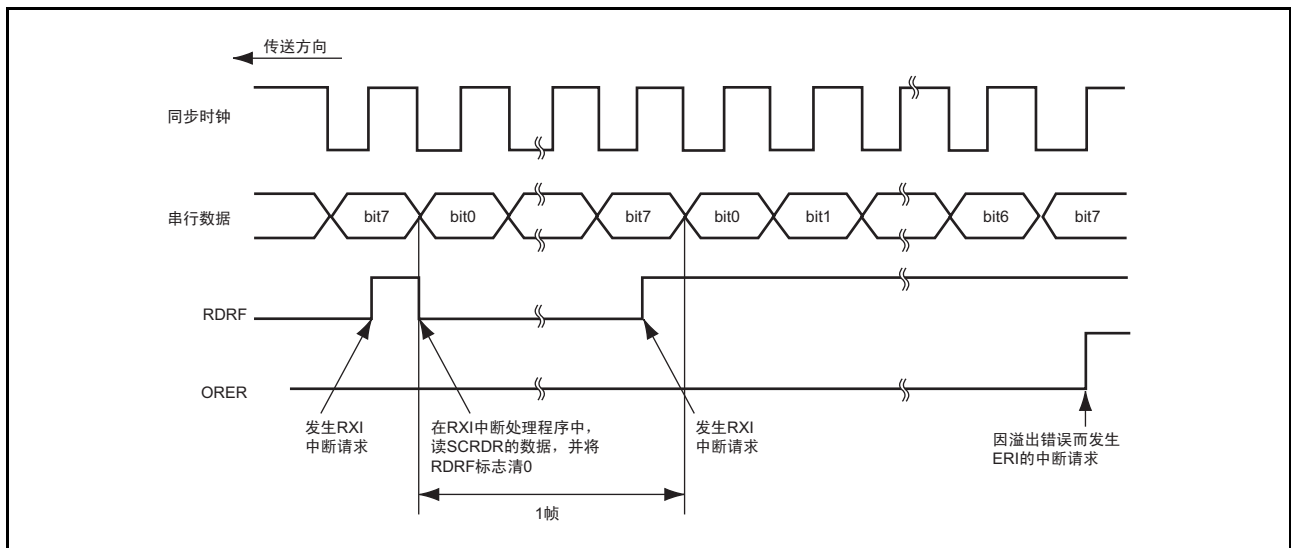


图 15.15 SCI 接收时的运行例子

• 串行数据的同时发送和接收 (时钟同步模式)

串行发送和接收的同时运行的流程图例子如图 15.16 所示。

在将 SCI 设定为可接收和发送的运行状态后，必须按照以下步骤同时发送和接收串行数据。

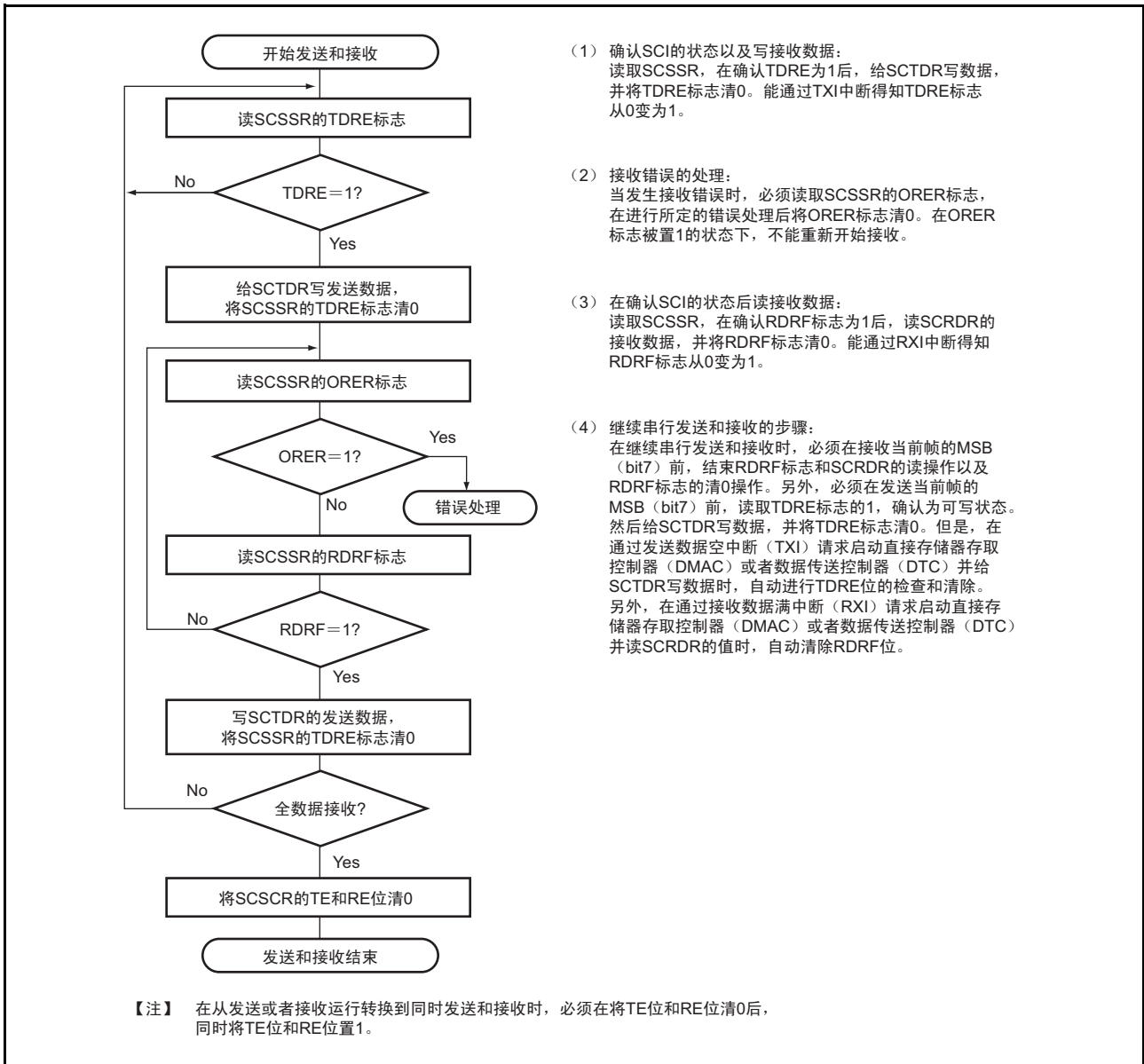


图 15.16 串行数据发送和接收的流程图例子

#### 15.4.4 多处理器通信功能

如果使用多处理器通信功能，就能通过附加多处理器位的异步串行通信，在多个处理器之间共享通信线路进行数据的发送和接收。在多处理器通信中，给接收站分配各自特有的 ID 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的数据发送周期构成，用多处理器位区分 ID 发送周期和数据发送周期。当多处理器位是 1 时，为 ID 发送周期；当多处理器位是 0 时，为数据发送周期。使用多处理器格式的处理器之间的通信例子如图 15.17 所示。发送站首先发送多处理器位为 1 的接收站 ID 码，接着发送多处理器位为 0 的发送数据。如果接收站接收到多处理器位为 1 的通信数据，就将接收数据与本站的 ID 比较，如果一致，就继续接收被发送的通信数据；如果不一致，就在接收到下一个多处理器位为 1 的通信数据之前，跳读通信数据。

SCI 为了支持此功能，在 SCSCR 中设有 MPIE 位。如果将 MPIE 置 1，就在接收到多处理器位为 1 的数据之前，禁止将接收数据从 SCRSR 传送到 SCRDR，并禁止接收错误的检测以及禁止 SCSSR 的 RDRF、FER、OER 各状态标志的置位。如果接收到多处理器位为 1 的字符，就在将 SCSSR 的 MPBR 置 1 的同时自动清除 MPIE，然后返回到通常的接收运行状态。此时，如果 SCSCR 的 RIE 已被置位，就产生 RXI 中断。

在指定多处理器格式时，奇偶校验位的指定无效。除此以外与通常的异步模式相同，多处理器通信时的时钟也和通常的异步模式相同。

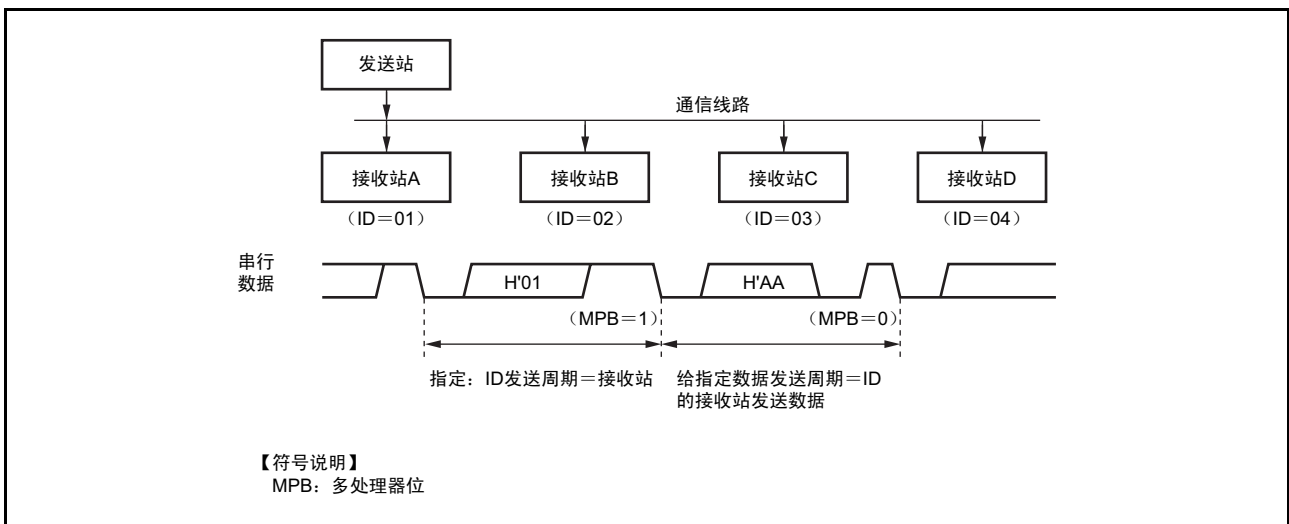


图 15.17 使用多处理器格式的通信的例子（将数据 H'AA 发送到接收站 A 的例子）

## 15.4.5 多处理器串行数据的发送

多处理器数据处理的流程图例子如图 15.18 所示。必须在 ID 发送周期将 SCSSR 的 MPBT 置 1 后发送 ID 码，在数据发送周期将 SCSSR 的 MPBT 清 0 后发送数据。其他运行和异步模式的运行相同。

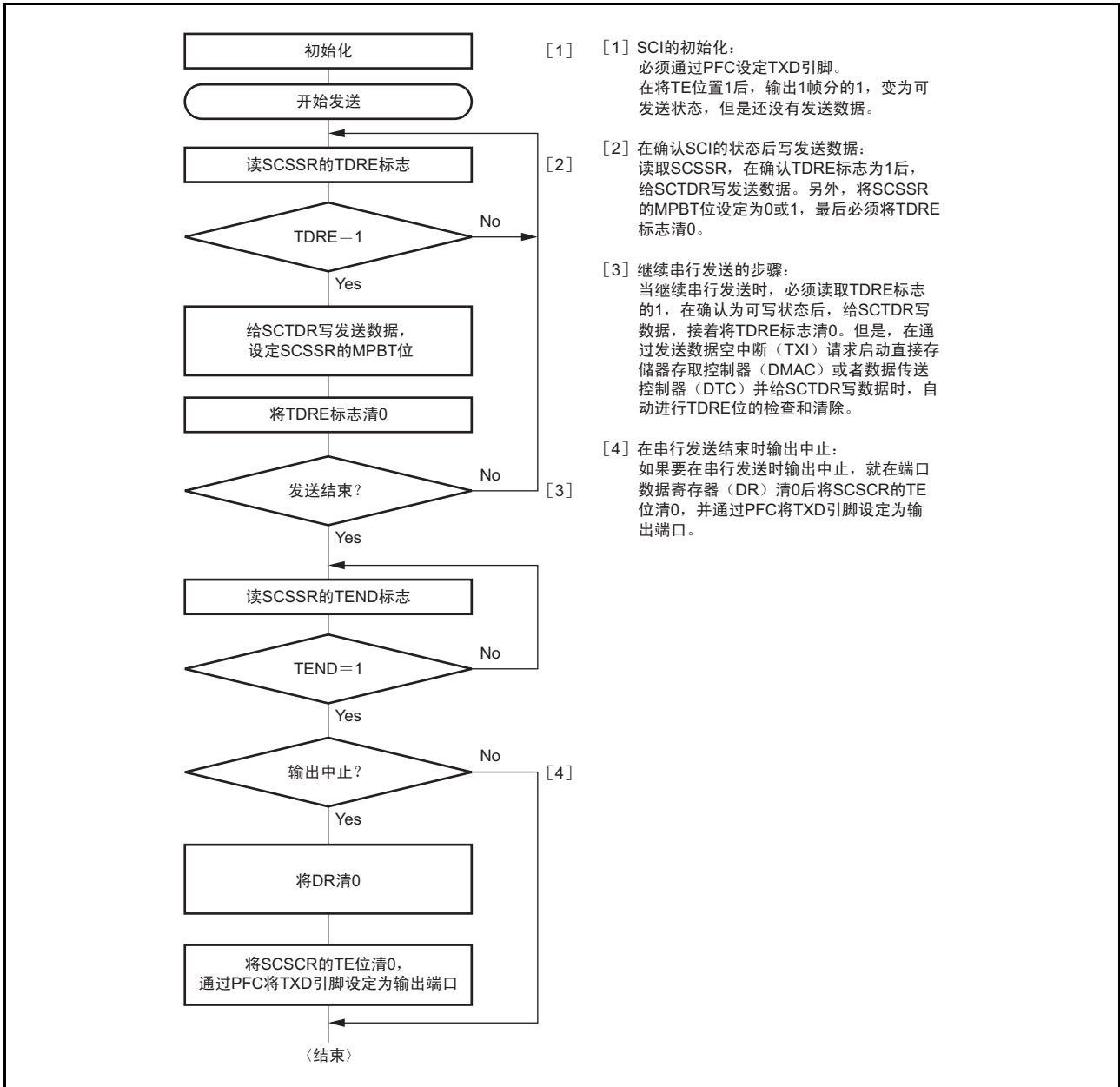


图 15.18 多处理器串行发送的流程图例子

### 15.4.6 多处理器串行数据的接收

多处理器数据接收的流程图例子如图 15.20 和图 15.21 所示。如果将 SCSCR 的 MPIE 置 1，就在接收到多处理器位为 1 的通信数据之前，跳读通信数据。如果接收到多处理器位为 1 的通信数据，就将接收数据传送到 SCRDR，此时产生 RXI 中断请求。其他运行和异步模式的运行相同。接收时的运行例子如图 15.19 所示。

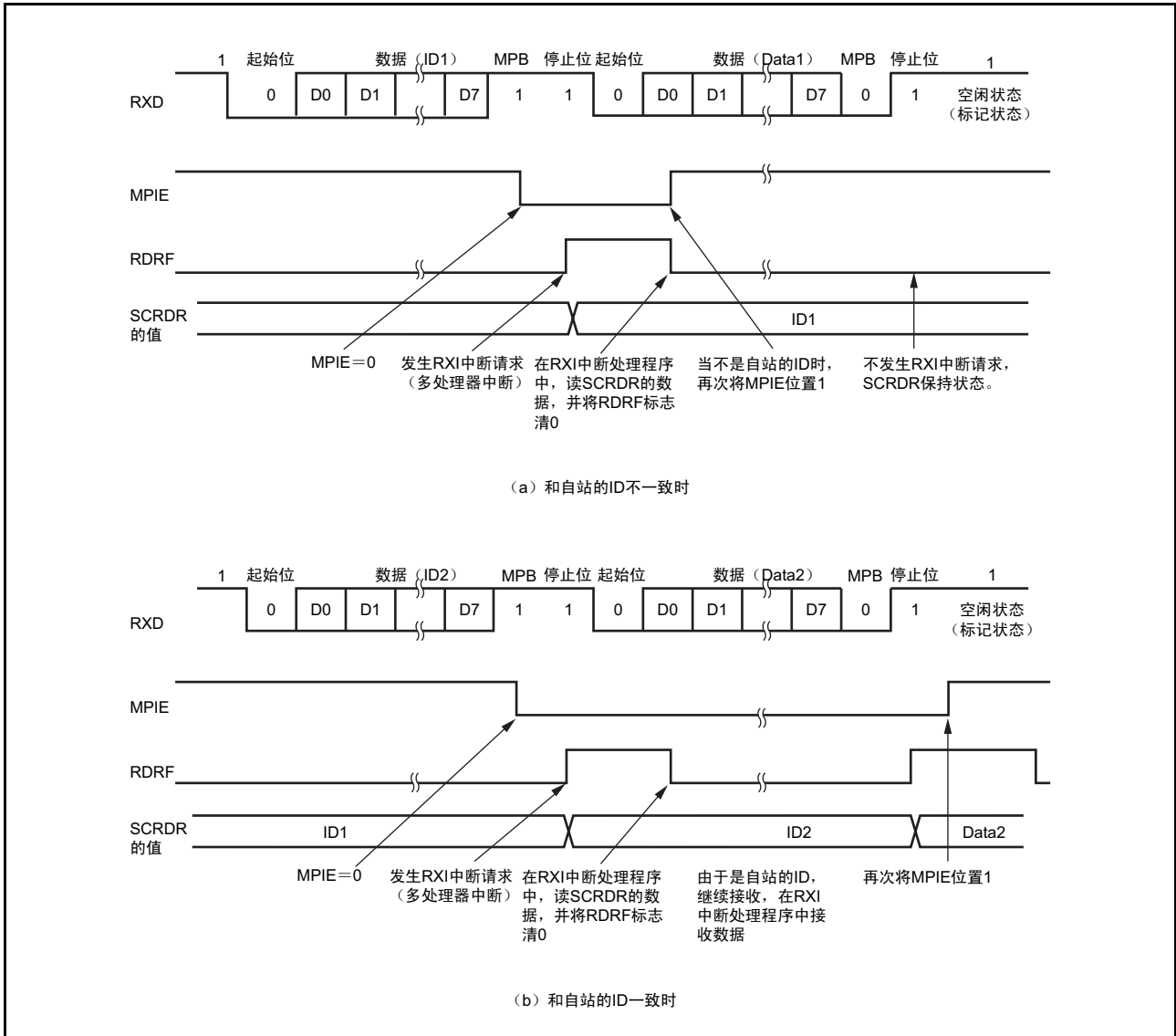


图 15.19 SCI 接收时的运行例子 (8 位数据 / 有多处理器位 / 1 个停止位的例子)

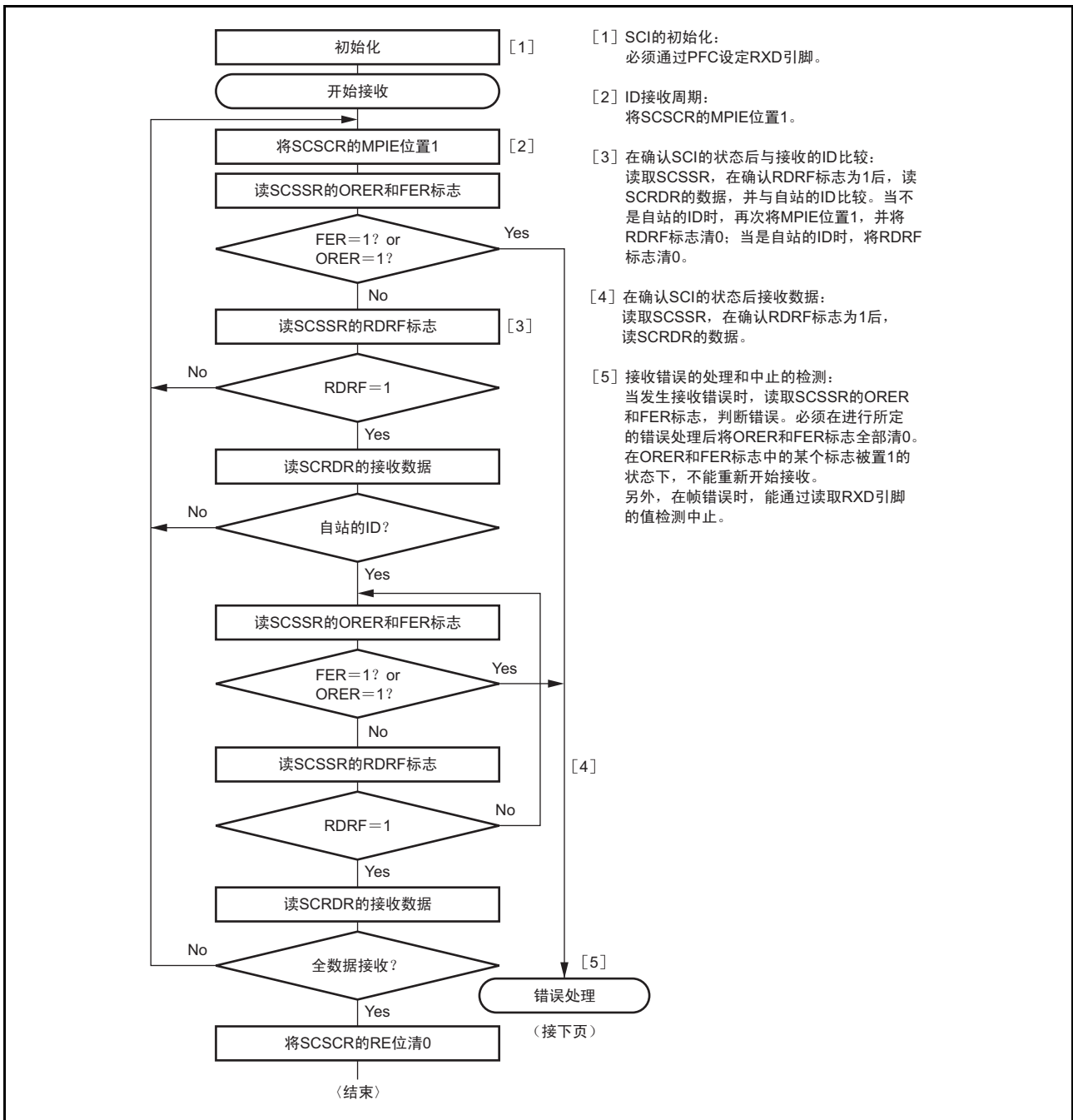


图 15.20 多处理器串行接收的流程图例子 (1)

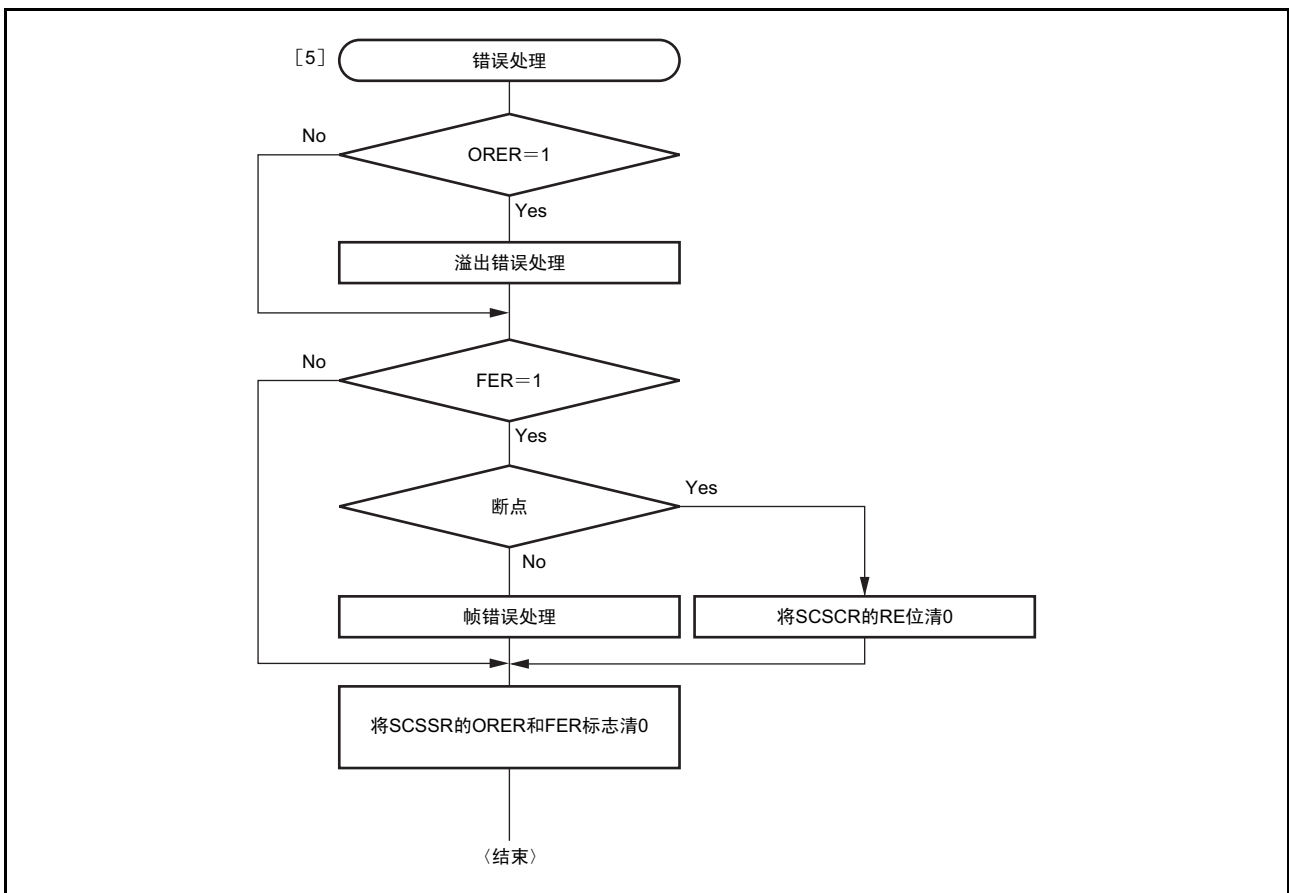


图 15.21 多处理器串行接收的流程图例子 (2)



## 15.5 SCI 的中断源和 DMAC/DTC

SCI 有发送结束的中断 (TEI) 请求、接收错误的中断 (ERI) 请求、接收数据满的中断 (RXI) 请求和发送数据空的中断 (TXI) 请求共 4 种。

各中断源和优先级如表 15.17 所示。能通过 SCSCR 的 TIE、RIE、TEIE 位以及 SCSPTTR 的 EIO 位，允许或者禁止各中断源。另外，分别将各中断请求独立传送到中断控制器。

如果串行状态寄存器 (SCSSR) 的 TDRE 标志被置 1，就产生 TDR 空中断请求。能通过 TDR 空的中断请求启动直接存储器存取控制器 (DMAC) 或者数据传送控制器 (DTC)，进行数据的传送。当通过 DMAC/DTC 给发送数据寄存器 (SCTDR) 写数据时，就自动清除 TDRE 标志。

如果 SCSSR 的 RDRF 标志被置 1，就产生 RDR 满中断请求。能通过 RDR 满的中断请求启动 DMAC/DTC，进行数据的传送。当通过 DMAC/DTC 读接收数据寄存器 (SCRDR) 时，就自动清除 RDRF 标志。

如果 SCSSR 的 ORER、FER 标志或者 PER 被置 1，就产生 ERI 中断请求，不能通过此 ERI 中断请求启动 DMAC/DTC。当通过 DMAC/DTC 进行接收数据处理并通过对 CPU 的中断进行接收错误处理时，必须将 RIE 位和 SCSPTTR 的 EIO 位同时置 1，使中断错误只发生在接收错误时。如果将 EIO 位设定为 0，即使在正常接收数据时，也对 CPU 产生中断。

如果 SCSSR 的 TEND 标志被置 1，就产生 TEI 中断请求，不能通过此 TEI 中断请求启动 DMAC/DTC。

另外，TXI 中断表示可写发送数据，TEI 中断表示发送结束。

表 15.17 SCI 中断源

中断源	内 容	DMAC/DTC 的启动
ERI	接收错误 (ORER、FER、PER) 的中断	不可
RXI	接收数据满 (RDRF) 的中断	可
TXI	发送数据空 (TDRE) 的中断	可
TEI	发送结束 (TEND) 的中断	不可

### 15.6 串行端口寄存器 (SCSPTR) 和 SCI 引脚的关系

SCSPTR 和 SCI 引脚的关系如图 15.22 和图 15.23 所示。

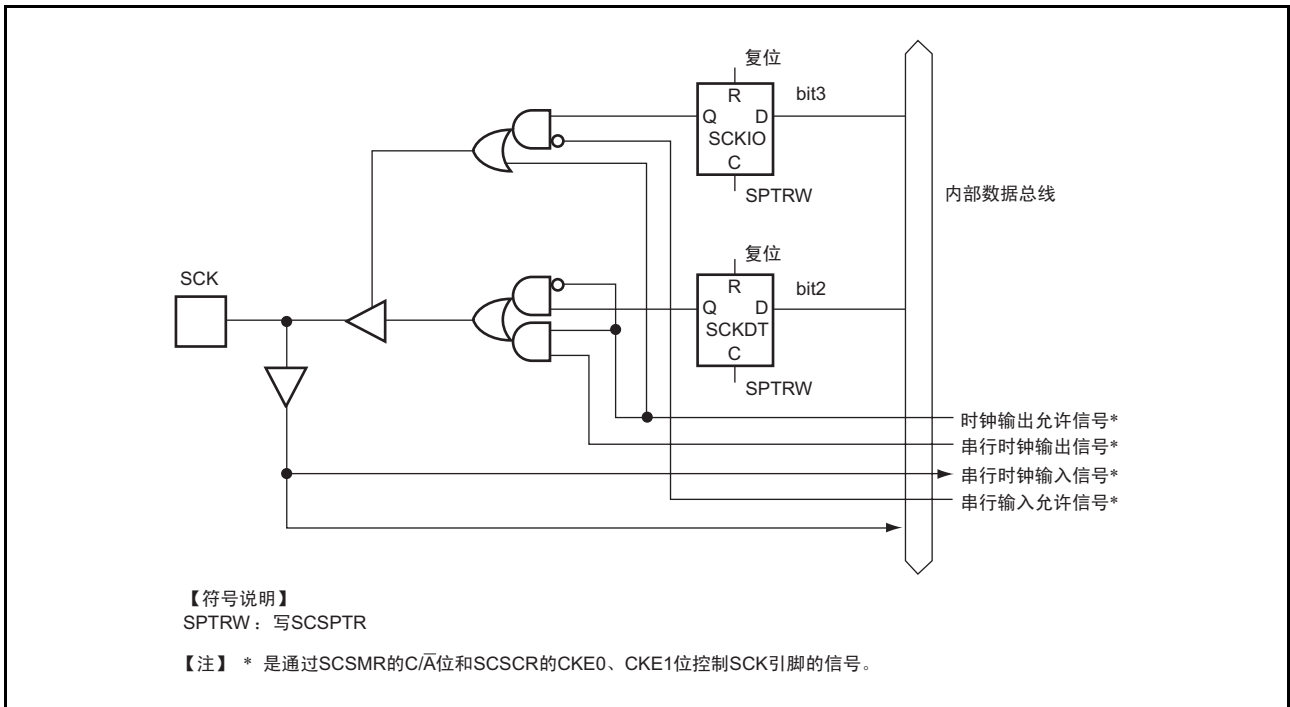


图 15.22 SCKIO 位、SPBDT 位和 SCK 引脚的关系

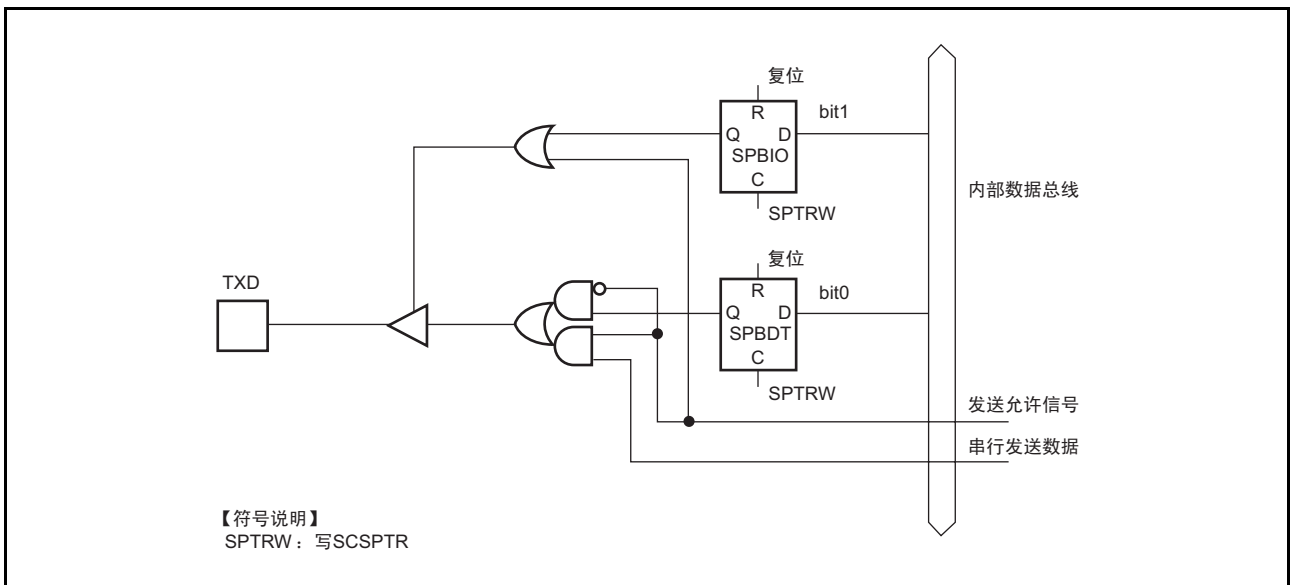


图 15.23 SPBIO 位、SPBDT 位和 TXD 引脚的关系

## 15.7 使用注意事项

### 15.7.1 有关 SCTDR 的写操作和 TDRE 标志的关系

串行状态寄存器 (SCSSR) 的 TDRE 标志是表示已将发送数据从发送数据寄存器 (SCTDR) 传送到发送移位寄存器 (SCTSR) 的状态标志。如果 SCI 将数据从 SCTDR 传送到 SCTSR, TDRE 位就被置 1。

与 TDRE 标志无关, 能将数据写到 SCTDR。但是, 如果在 TDRE 标志为 0 的状态下将新数据写到 SCTDR, 因为还没有传送到 SCTSR, 所以被保存到 SCTDR 的数据就会丢失。因此, 必须在确认 TDRE 标志已被置 1 后给 SCTDR 写发送数据。

### 15.7.2 有关同时发生多个接收错误时的运行

当同时发生多个接收错误时, SCSSR 各状态标志的状态如表 15.18 所示。另外, 在发生溢出错误时, 因为不将数据从接收移位寄存器 (SCRSR) 传送到接收数据寄存器 (SCRDR), 所以接收数据就会丢失。

表 15.18 SCSSR 状态标志的状态和接收数据的传送

接收错误的状态	SCSSR 的状态标志				接收数据的传送 SCRSR→SCRDR
	RDRF	ORER	FER	PER	
溢出错误	1	1	0	0	×
帧错误	0	0	1	0	○
奇偶校验错误	0	0	0	1	○
溢出错误 + 帧错误	1	1	1	0	×
溢出错误 + 奇偶校验错误	1	1	0	1	×
帧错误 + 奇偶校验错误	0	0	1	1	○
溢出错误 + 帧错误 + 奇偶校验错误	1	1	1	1	×

#### 【符号说明】

○: 将接收数据从 SCRSR 传送到 SCRDR。

×: 不将接收数据从 SCRSR 传送到 SCRDR。

### 15.7.3 有关中止的检测和处理

在检测帧错误 (FER) 时, 能通过直接读 RXD 引脚的值检测中止。在中止时, RXD 引脚的输入始终为 0, 所以 FER 标志被置位, 并且奇偶校验错误 (PER) 也可能被置位。

必须注意: SCI 在接收中止后还继续接收运行, 但是接收的数据不传送到 SCRDR。

### 15.7.4 中止的发送

能通过串行端口寄存器 (SCSPTR) 的 SPB0IO 和 SPB0DT 位决定 TXD 引脚的输入 / 输出的条件和电平, 所以利用这一功能可以发送中止。

从串行发送的初始化到 TE 位被置 1, TXD 引脚不工作。在此期间, 标记状态被 SPB0DT 位的值代替。因此, 先要将 SPB0IO 和 SPB0DT 位设定为 1 (输出高电平)。

当要在串行发送期间时发送中止时, 就必须在将 SPB0DT 位清 0 (低电平) 后, 再将 TE 位清 0 (停止发送)。如果将 TE 位清 0, 就与当前的发送状态无关, 初始化发送部并从 TXD 引脚输出 0。

### 15.7.5 异步模式的接收数据采样时序和接收容限

在异步模式中，SCI 通过传送速率的 16 倍频率的基本时钟运行。

在接收时，SCI 通过基本时钟采样起始位的下降沿，取得内部同步，而在基本时钟的第 8 个时钟脉冲的上升沿将接收数据取到内部。

如图 15.24 所示。

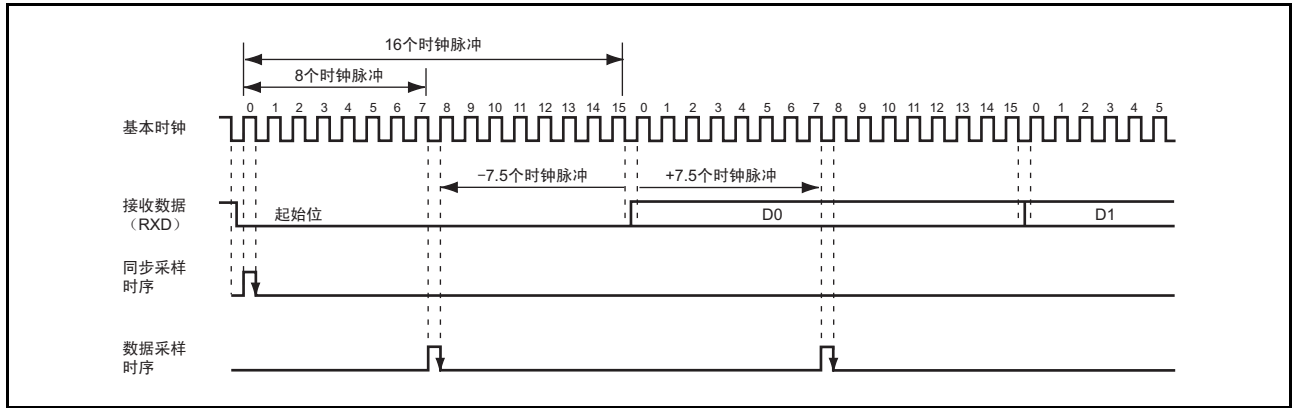


图 15.24 异步模式的接收数据采样时序

因此，能用表达式 (1) 表示异步模式的接收容限。

$$M = \left| \left( 0.5 - \frac{1}{2N} \right) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100[\%] \quad \dots \text{表达式 (1)}$$

M : 接收容限 (%)

N : 对应时钟的位速率比 (N=16)

D : 时钟占空比 (D=0 ~ 1.0)

L : 帧长 (L=9 ~ 12)

F : 时钟频率的偏差绝对值

当表达式 (1) 的中 F=0、D=0.5 时，根据表达式 (2) 接收容限就为 46.875%。

当 D=0.5、F=0 时

$$M = (0.5 - 1/(2 \times 16)) \times 100\%$$

$$= 46.875\%$$

…表达式 (2)

此值为计算值，所以在系统设计时必须留有 20 ~ 30% 的余地。

### 15.7.6 DMAC/DTC 的使用注意事项

1. 当将外部时钟源用于同步时钟时，在通过 DMAC/DTC 更新 SCTDR 后，必须通过外围运行时钟至少经过 5 个周期后输入外部时钟。如果在更新 SCTDR 后的 4 个周期以内输入发送时钟，就可能发生误动作（参照图 15.25）。

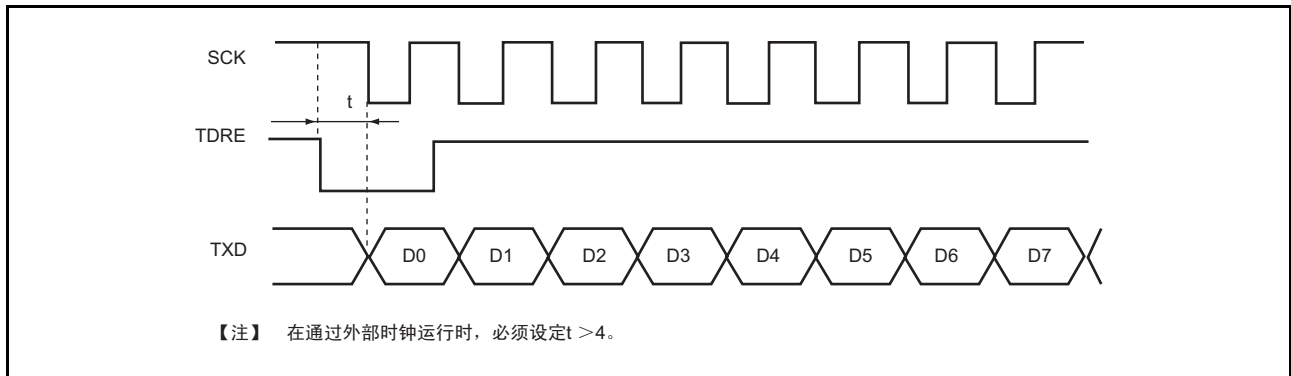


图 15.25 通过 DMAC/DTC 进行同步时钟传送的例子

2. 通过 TXI 中断启动 DMAC/DTC，给 SCTDR 写数据时，TEND 标志不定，所以 TEND 标志不能用作发送结束标志。

### 15.7.7 时钟同步外部时钟模式的注意事项

必须在将外部时钟 SCK 从 0 变为 1 后经过 4 个以上的外围运行时钟周期后，然后设定为 TE=1、RE=1。必须在外部时钟 SCK 为 1 时设定为 TE=RE=1。

### 15.7.8 模块待机模式的设定

SCI 能通过待机控制寄存器，禁止或者允许本模块的运行。初始值为停止 SCI 的运行。通过解除模块待机模式，允许寄存器的存取。详细内容请参照“第 26 章 低功耗模式”。

## 第 16 章 带 FIFO 的串行通信接口 (SCIF)

本 LSI 内置 1 个通道的带 FIFO 的串行通信接口 (SCIF: Serial Communication Interface with FIFO)。SCIF 能以异步通信和时钟同步通信 2 种方式进行串行通信。

本 LSI 内置了分别用于接收和发送的 16 段 FIFO 寄存器，能连续进行高效率的高速通信。

### 16.1 特点

- 异步串行通信
 

通过以字符为单位取得同步的异步方式进行串行数据通信，能与 Universal Asynchronous Receiver/Transmitter (UART) 或者 Asynchronous Communication Interface Adapter (ACIA) 等标准异步通信的 LSI 进行串行数据通信，能从 8 种格式中选择串行数据的通信格式。

数据长 : 7 位或者 8 位

停止位长 : 1 位或者 2 位

奇偶校验 : 偶校验、奇校验或者无奇偶校验

接收错误的检测 : 检测奇偶校验错误、帧错误和溢出错误

中止的检测 : 当发生帧错误并且继续有至少 1 帧长的空间 0 (低电平) 时，就检测到中止；当发生帧错误时，能通过直接从串行端口寄存器读取 RXD 引脚的电平来检测到中止。
- 时钟同步串行通信
 

与时钟同步进行串行数据通信，能和具有时钟同步通信功能的其他 LSI 进行串行数据通信。串行数据的通信格式只有 1 种。

数据长 : 8 位

接收错误的检测 : 检测溢出错误
- 能进行全双工通信
 

具有独立的发送部和接收部，所以能同时发送和接收，而且发送部和接收部都是 16 段 FIFO 缓冲结构，所以能连续发送和连续接收串行数据。
- 可通过内部波特率发生器选择任意的位速率
- 内部或者外部的发送和接收时钟源
 

可选择波特率发生器 (内部时钟) 或者 SCK 引脚 (外部时钟)
- 4 种中断源
 

中断源有发送 FIFO 数据空、中止、接收 FIFO 数据满、接收错误的 4 种中断，能分别独立请求中断。
- 能在发送 FIFO 数据空或者接收 FIFO 数据满时，启动 DTC 进行数据传送。
- 能设定模块待机模式。
- 在异步模式中，有调制解调器功能 ( $\overline{\text{RTS}}$ 、 $\overline{\text{CTS}}$ )。
- 能得知发送和接收 FIFO 寄存器的数据个数以及接收 FIFO 寄存器的接收数据的接收错误数。
- 在异步模式接收时，能检测到超时错误 (DR)。

SCIF 的框图如图 16.1 所示。

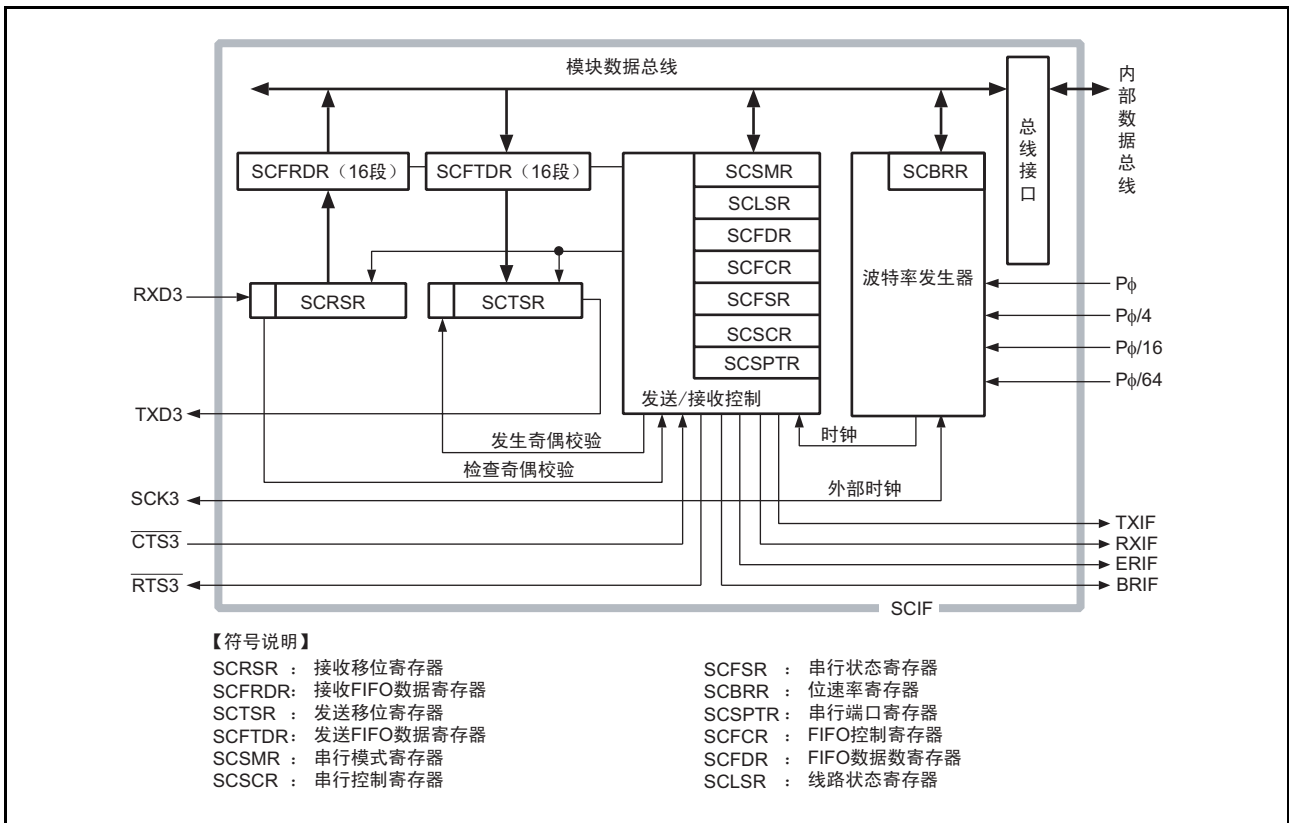


图 16.1 SCIF 的框图

### 16.2 输入 / 输出引脚

SCIF 的输入 / 输出引脚如表 16.1 所示。

表 16.1 引脚结构

通道	名称	引脚名 *	输入 / 输出	功 能
3	串行时钟引脚	SCK3	输入 / 输出	时钟的输入 / 输出
	接收数据引脚	RXD3	输入	接收数据的输入
	发送数据引脚	TXD3	输出	发送数据的输出
	请求发送引脚	$\overline{\text{RTS3}}$	输出	请求发送
	清除发送引脚	$\overline{\text{CTS3}}$	输入	清除发送

【注】 \* 本文省略了通道，分别简称为 SCK、RXD、TXD、 $\overline{\text{RTS}}$ 、 $\overline{\text{CTS}}$ 。

### 16.3 寄存器说明

SCIF 的寄存器如下所示，这些寄存器指定数据格式和位速率以及控制发送部和接收部。

表 16.2 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
串行模式寄存器 _3	SCSMR_3	R/W	H'0000	H'FFFC180	16
位速率寄存器 _3	SCBRR_3	R/W	H'FF	H'FFFC182	8
串行控制寄存器 _3	SCSCR_3	R/W	H'0000	H'FFFC184	16
发送 FIFO 数据寄存器 _3	SCFTDR_3	W	H'xx	H'FFFC186	8
串行状态寄存器 _3	SCFSR_3	R/W	H'0060	H'FFFC188	16
接收 FIFO 数据寄存器 _3	SCFRDR_3	R	H'xx	H'FFFC18A	8
FIFO 控制寄存器 _3	SCFCR_3	R/W	H'0000	H'FFFC18C	16
FIFO 数据个数寄存器 _3	SCFDR_3	R	H'0000	H'FFFC18E	16
串行端口寄存器 _3	SCSPTR_3	R/W	H'00xx	H'FFFC190	16
线路状态寄存器 _3	SCLSR_3	R/W	H'0000	H'FFFC192	16

#### 16.3.1 接收移位寄存器 (SCRSR)

SCRSR 是接收串行数据的寄存器。

SCIF 按照从 LSB (bit0) 开始接收的顺序，将 RXD 引脚输入的串行数据保存到 SCRSR，并转换为并行数据。当 1 字节的数据接收结束时，数据就自动传送到接收 FIFO 数据寄存器 (SCFRDR)。

CPU 不能直接读写 SCRSR。

位:	7	6	5	4	3	2	1	0
初始值:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

#### 16.3.2 接收 FIFO 数据寄存器 (SCFRDR)

SCFRDR 是保存已接收串行数据的 8 位 16 段 FIFO 寄存器。

当 1 字节的串行数据接收结束时，SCIF 就将已接收的串行数据从接收移位寄存器 (SCRSR) 传送并保存到 SCFRDR，然后结束接收运行。在结束 16 字节的保存前可连续接收，CPU 能读但不能写 SCFRDR。如果在接收 FIFO 数据寄存器中没有接收数据的状态下读 SCFRDR，读的值不定；如果此寄存器的接收数据满，会丢失以后接收的串行数据。

位:	7	6	5	4	3	2	1	0
初始值:	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
7 ~ 0		不定	R	用于串行接收数据的 FIFO



### 16.3.3 发送移位寄存器 (SCTSR)

SCTSR 是发送串行数据的寄存器。

一旦将发送数据从发送 FIFO 数据寄存器 (SCFTDR) 传送到 SCTSR, SCIF 就从 LSB (bit0) 按顺序将数据发送到 TXD 引脚, 进行串行数据的发送。当 1 字节的数据发送结束时, 下一个发送数据就自动从 SCFTDR 传送到 SCTSR 开始发送。

CPU 不能直接读写 SCTSR。

位:	7	6	5	4	3	2	1	0
初始值:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

### 16.3.4 发送 FIFO 数据寄存器 (SCFTDR)

SCFTDR 是保存串行发送数据的 8 位 16 段 FIFO 寄存器。

如果检测到发送移位寄存器 (SCTSR) 为空, SCIF 就将写到 SCFTDR 的发送数据传送到 SCTSR, 开始串行发送。可连续进行串行发送直到 SCFTDR 为发送数据空。CPU 随时能写 SCFTDR。

如果 SCFTDR 为发送数据满 (16 字节), 就不能写下一个数据。即使写, 数据也被忽视。

位:	7	6	5	4	3	2	1	0
初始值:	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W

位	位名	初始值	R/W	说 明
7 ~ 0		不定	W	用于串行发送数据的 FIFO

## 16.3.5 串行模式寄存器 (SCSMR)

SCSMR 是设定 SCIF 的串行通信格式和选择波特率发生器时钟源的 16 位寄存器。CPU 随时能读写 SCSMR。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	C/A	CHR	PE	O/E	STOP	-	-	CKS[1:0]
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 8	—	全 0	R	保留位 读写的值总是 0。
7	C/A	0	R/W	通信模式 SCIF 的运行模式可选择异步模式或者时钟同步模式。 0: 异步模式 1: 时钟同步模式
6	CHR	0	R/W	字符长度 异步模式的数据长可选择 7 位或者 8 位数据。在时钟同步模式中, 与 CHR 的设定无关, 数据长固定为 8 位数据。 0: 8 位数据 1: 7 位数据 * <b>【注】</b> * 如果选择 7 位数据, 就不发送发送 FIFO 数据寄存器的 MSB (7 位)。
5	PE	0	R/W	奇偶校验的允许 在异步模式中, 选择是否在发送时附加奇偶校验位以及在接收时检查奇偶校验位。在时钟同步模式中, 与 PE 位的设定无关, 不附加也不检查奇偶校验位。 0: 禁止附加和检查奇偶校验位 1: 允许附加和检查奇偶校验位 * <b>【注】</b> * 如果 PE 位被置 1, 就在发送时给发送数据附加 O/E 位指定的偶校验位或者奇校验位后发送, 而在接收时检查已接收的奇偶校验位是否为 O/E 位指定的偶校验位或者奇校验位。
4	O/E	0	R/W	奇偶校验模式 选择是通过偶校验还是通过奇校验附加和检查奇偶校验位。只在异步模式中将 PE 位设定为 1 并且允许附加和检查奇偶校验位时, O/E 位的设定才有效; 在时钟同步模式或者异步模式中禁止附加和检查奇偶校验位时, O/E 位的指定无效。 0: 偶校验 *1 1: 奇校验 *2 <b>【注】</b> *1 如果设定为偶校验, 就在发送时附加奇偶校验位, 使奇偶校验位和发送字符中的 1 的个数为偶数, 然后发送; 在接收时检查奇偶校验位和接收字符中 1 的个数是否为偶数。 *2 如果设定为奇校验, 就在发送时附加奇偶校验位, 使奇偶校验位和发送字符中的 1 的个数为奇数, 然后发送; 在接收时检查奇偶校验位和接收字符中 1 的个数是否为奇数。

位	位名	初始值	R/W	说 明
3	STOP	0	R/W	停止位长 异步模式中的停止位长可选择 1 位或者 2 位。STOP 位的设定只在异步模式中有效。因为在设定为时钟同步模式时不附加停止位，所以此位的设定无效。另外，在接收时，与 STOP 位的设定无关，只检查已接收的第 1 个停止位。当第 2 个停止位是 1 时，作为停止位处理，而当第 2 个停止位是 0 时，作为下一个发送字符的起始位处理。 0: 1 个停止位 发送时，在发送字符的末尾附加 1 位的 1（停止位）后发送。 1: 2 个停止位 发送时，在发送字符的末尾附加 2 位的 1（停止位）后发送。
2	—	0	R	保留位 读写的值总是 0。
1、0	CKS[1:0]	00	R/W	时钟选择 1、0 选择内部波特率发生器的时钟源。 有关时钟源、位速率寄存器的设定值和波特率的关系，请参照“16.3.8 位速率寄存器 (SCBRR)”。 00: P $\phi$ 时钟 01: P $\phi$ /4 时钟 10: P $\phi$ /16 时钟 11: P $\phi$ /64 时钟 <b>【注】</b> P $\phi$ : 外围时钟

### 16.3.6 串行控制寄存器 (SCSCR)

SCSCR 是进行 SCIF 的发送和接收、允许或者禁止中断请求以及选择发送 / 接收时钟源的 16 位寄存器。CPU 随时能读写 SCSCR。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	TIE	RIE	TE	RE	REIE	-	CKE[1:0]	
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 8	—	全 0	R	保留位 读写的值总是 0。
7	TIE	0	R/W	发送的中断允许 在串行状态寄存器 (SCFSR) 的 TDFE 标志被置 1 时 (串行发送数据从发送 FIFO 数据寄存器 (SCFTDR) 传送到发送移位寄存器 (SCTSR) 后，并且发送 FIFO 寄存器的数据个数少于指定的发送触发个数)，允许或者禁止发送 FIFO 数据空的中断 (TXIF) 请求。 0: 禁止发送 FIFO 数据空的中断 (TXIF) 请求 * 1: 允许发送 FIFO 数据空的中断 (TXIF) 请求 <b>【注】</b> * 能通过写发送数据 (数据量大于 SCFTDR 指定的发送触发个数)、在读 TDFE 标志的 1 后将该标志清 0 或者通过将 TIE 清 0 来解除 TXIF。

位	位名	初始值	R/W	说 明
6	RIE	0	R/W	<p>接收的中断允许 允许或者禁止以下的中断请求：</p> <ol style="list-style-type: none"> <li>1. SCFSR 的 RDF 标志或者 DR 标志被置 1 时的接收数据满的中断 (RXIF) 请求</li> <li>2. SCFSR 的 ER 标志被置 1 时的接收错误中断 (ERIF) 请求</li> <li>3. SCFSR 的 BRK 标志或者 SCLSR 的 ORER 标志被置 1 时的中止中断 (BRIF) 请求。</li> </ol> <p>0: 禁止接收数据满中断 (RXIF) 请求、接收错误中断 (ERIF) 请求和中止中断 (BRIF) 请求 *</p> <p>1: 允许接收数据满中断 (RXIF) 请求、接收错误中断 (ERIF) 请求和中止中断 (BRIF) 请求</p> <p><b>【注】</b> * 通过在读 RDF 标志或者 DR 标志的 1 后将该位清 0、或者通过将 RIE 位清 0 来解除 RXIF 中断。通过在读取 ER 标志、BRK 标志或者 ORER 标志的 1 后将这些标志清 0、或者通过将 RIE、REIE 位清 0 来解除 ERIF、BRIF 中断。</p>
5	TE	0	R/W	<p>发送的允许 允许或者禁止串行发送。</p> <p>0: 禁止发送 1: 允许发送 *</p> <p><b>【注】</b> * 在此状态下, 如果将发送数据写到 SCFTDR, 就开始串行发送。必须在将 TE 位设定为 1 前设定 SCSMR 和 SCFCR, 决定发送格式以及对发送 FIFO 进行复位。</p>
4	RE	0	R/W	<p>接收的允许 允许或者禁止串行接收。</p> <p>0: 禁止接收 *1 1: 允许接收 *2</p> <p><b>【注】</b> *1 必须注意: 即使将 RE 位清 0, DR、ER、BRK、RDF、FER、PER 和 ORER 的各位也不受影响而保持状态。 *2 在此状态下, 如果在异步模式中检测到起始位以及在时钟同步模式中检测到同步时钟输入, 就开始串行接收。必须在将 RE 位设定为 1 前设定串行模式寄存器 (SCSMR) 和 FIFO 控制寄存器 (SCFCR), 决定接收格式以及对接收 FIFO 进行复位。</p>
3	REIE	0	R/W	<p>接收错误中断的允许 允许或者禁止接收错误的中断 (ERIF09) 请求和中止中断 (BRIF) 请求。但是 REIE 位的设定只在 RIE 位是 0 时有效。</p> <p>0: 禁止接收错误的中断 (ERIF) 请求和中止中断 (BRIF) 请求 *</p> <p>1: 允许接收错误的中断 (ERIF) 请求和中止中断 (BRIF) 请求</p> <p><b>【注】</b> * 通过在读 ER 标志、BRK 标志或者 ORER 标志的 1 后将该标志清 0、或者通过将 RIE 位、REIE 位清 0 来解除 ERIF、BRIF 中断。如果 REIE 位被置 1, 即使将 RIE 位设定为 0, 也发生 ERIF 中断或者 BRIF 中断。</p>
2	—	0	R	<p>保留位 读写的值总是 0。</p>

位	位名	初始值	R/W	说 明
1、0	CKE[1:0]	00	R/W	<p>时钟允许 1、0</p> <p>选择 SCIF 时钟源以及禁止或者允许 SCK 引脚输出时钟。通过 CKE1 位和 CKE0 位的组合，将 SCK 引脚设定为串行时钟的输出引脚或者串行时钟的输入引脚。但是 CKE0 位的设定只在内部时钟运行 (CKE1=0) 时有效。当外部时钟运行 (CKE1=1) 时，CKE0 位的设定无效。如果用在时钟同步模式中，就必须由 SCSMR 决定 SCIF 的运行模式，然后设定 CKE1 和 CKE0 位。</p> <ul style="list-style-type: none"> <li>• 异步模式                     <ul style="list-style-type: none"> <li>00: 内部时钟 /SCK 引脚是输入引脚 (忽视输入信号)。SCK 的引脚状态取决于 SCSPTR 的 SCKIO 位和 SCKDT 位。</li> <li>01: 内部时钟 /SCK 引脚是时钟的输出引脚 (输出时钟的频率是位速率的 16 倍)</li> <li>10: 外部时钟 /SCK 引脚是时钟的输入引脚 (输入时钟的频率是位速率的 16 倍)</li> <li>11: 禁止设定</li> </ul> </li> <li>• 时钟同步模式                     <ul style="list-style-type: none"> <li>00: 内部时钟 /SCK 引脚是同步时钟的输出引脚</li> <li>01: 内部时钟 /SCK 引脚是同步时钟的输出引脚</li> <li>10: 外部时钟 /SCK 引脚是同步时钟的输入引脚</li> <li>11: 禁止设定</li> </ul> </li> </ul>

### 16.3.7 串行状态寄存器 (SCFSR)

SCFSR 是 16 位寄存器，高 8 位表示接收 FIFO 数据寄存器数据的接收错误数，低 8 位表示 SCIF 的运行状态标志。

CPU 随时能读写 SCFSR，但是不能给 ER、TEND、TDFE、BRK、RDF 和 DR 的各状态标志写 1。另外，为了将这些标志清 0，必须先读取 1。FER 标志和 PER 标志为只读标志，不能写。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PER[3:0]				FER[3:0]				ER	TEND	TDFE	BRK	FER	PER	RDF	DR
初始值:	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/(W)*	R/(W)*

【注】\* 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说 明
15 ~ 12	PER[3:0]	0000	R	<p>奇偶校验错误数</p> <p>表示在接收 FIFO 数据寄存器 (SCFRDR) 保存的接收数据中发生奇偶校验错误的的数据个数。</p> <p>在 SCFSR 的 ER 位被置位后，bit15 ~ 12 的值表示发生奇偶校验错误的的数据个数。如果 SCFRDR 的 16 字节接收数据都发生奇偶校验错误，PER3 ~ PER0 就为 0。</p>

位	位名	初始值	R/W	说 明
11 ~ 8	FER[3:0]	0000	R	<p>帧错误数</p> <p>表示在接收 FIFO 数据寄存器 (SCFRDR) 保存的接收数据中发生帧错误的的数据个数。</p> <p>在 SCFSR 的 ER 位被置位后, bit11 ~ 8 的值表示发生帧错误的的数据个数。如果 SCFRDR 的 16 字节接收数据都发生帧错误, FER3 ~ FER0 就为 0。</p>
7	ER	0	R/(W)*	<p>接收错误</p> <p>表示发生了帧错误或者在接收含有奇偶校验位的数据时发生了奇偶校验错误。*1</p> <p>0: 表示正在接收或者接收已正常结束</p> <p>[清除条件]</p> <ul style="list-style-type: none"> <li>• 上电复位</li> <li>• 在读 ER=1 的状态后写 0 时</li> </ul> <p>1: 表示在接收时发生帧错误或者奇偶校验错误</p> <p>[置位条件]</p> <ul style="list-style-type: none"> <li>• 在 1 次数据接收结束时, 检查出接收数据的最后停止位不是 1 而是 0 时 *2</li> <li>• 在接收时的接收数据和奇偶校验位的 1 的个数与串行模式寄存器 (SCSMR) 的 O/E 位指定的偶数 / 奇数校验的设定不一致时</li> </ul> <p>【注】 *1 在 SCSCR 的 RE 位被清 0 时, ER 标志不受影响而保持原来的状态。即使发生接收错误, 也将接收数据传送到 SCFRDR 并继续接收运行。能通过 SCFSR 的 FER 或者 PER 位来判断从 SCFRDR 读取的数据是否有接收错误。</p> <p>*2 在 2 个停止位模式中, 只检查第 1 个停止位而不检查第 2 个停止位。</p>
6	TEND	1	R/(W)*	<p>发送结束</p> <p>表示在发送字符的最后位时, 因 SCFTDR 中没有有效数据而结束发送。</p> <p>0: 表示正在发送</p> <p>[清除条件]</p> <ul style="list-style-type: none"> <li>• 当给 SCFTDR 写发送数据, 并在读 TEND=1 的状态后给 TEND 标志写 0 时</li> </ul> <p>1: 表示发送已结束</p> <p>[置位条件]</p> <ul style="list-style-type: none"> <li>• 上电复位</li> <li>• 当 SCSCR 的 TE 位是 0 时</li> <li>• 当发送 1 字节串行发送字符的最后位时, SCFTDR 中没有发送数据时</li> </ul> <p>【注】 因为在通过 TXIF 中断启动 DTC 后将数据写到 SCFTDR 时, TEND 标志不定, 所以不能将 TEND 标志用作发送结束标志。</p>

位	位名	初始值	R/W	说 明
5	TDFE	1	R/(W)*	<p>发送 FIFO 数据空</p> <p>表示数据从发送 FIFO 数据寄存器 (SCFTDR) 传送到发送移位寄存器 (SCTSR)、SCFTDR 的数据个数小于 FIFO 控制寄存器 (SCFCR) 的 TTRG1 位和 TTRG0 位指定的发送触发数据个数、以及能将发送数据写到 SCFTDR。</p> <p>0: 表示写到 SCFTDR 的发送数据个数大于指定的发送触发个数 [清除条件]</p> <ul style="list-style-type: none"> <li>• 当在读 TDFE=1 的状态后, 将大于指定发送触发个数的发送数据个数写到 SCFTDR 并给 TDFE 写 0 时</li> <li>• 当通过 DTC 将大于指定发送触发个数的数据写到 SCFTDR 时</li> </ul> <p>1: 表示写到 SCFTDR 的发送数据个数小于等于指定发送触发个数 [置位条件]</p> <ul style="list-style-type: none"> <li>• 上电复位</li> <li>• 当写到发送结果 SCFTDR 的发送数据个数小于等于指定发送触发个数时</li> </ul> <p><b>【注】</b> * 因为 SCFTDR 是 16 字节 FIFO 寄存器, 所以在 TDFE=1 的状态下, 能写的最大数据个数为 16 减去指定发送触发个数。即使写更多的数据, 此数据也被忽视。SCFDR 的高 8 位表示 SCFTDR 的数据个数。</p>
4	BRK	0	R/(W)*	<p>中止的检测</p> <p>表示已检测到接收数据的中止信号。</p> <p>0: 没有中止信号 [清除条件]</p> <ul style="list-style-type: none"> <li>• 上电复位</li> <li>• 当在读 BRK=1 的状态后写 0 时</li> </ul> <p>1: 接收到中止信号 * [置位条件]</p> <ul style="list-style-type: none"> <li>• 在接收含有帧错误的数据后, 继续有至少 1 帧长的空间 0 (低电平) 时</li> </ul> <p><b>【注】</b> * 当检测到中止时, 在检测后停止将接收数据 (H'00) 传送到 SCFRDR。当中止结束并且接收信号标记变为 1 时, 重新开始传送接收数据。</p>
3	FER	0	R	<p>帧错误表示</p> <p>表示在异步模式中, 从接收 FIFO 数据寄存器 (SCFRDR) 读取的数据是否发生帧错误。</p> <p>0: 表示下一次从 SCFRDR 读取的接收数据没有发生帧错误 [清除条件]</p> <ul style="list-style-type: none"> <li>• 上电复位</li> <li>• 下一次从 SCFRDR 读取的数据没有帧错误</li> </ul> <p>1: 表示下一次从 SCFRDR 读取的接收数据发生了帧错误 [置位条件]</p> <ul style="list-style-type: none"> <li>• 下一个 SCFRDR 读数据发生了帧错误</li> </ul>

位	位名	初始值	R/W	说 明
2	PER	0	R	<p>奇偶校验错误</p> <p>表示在异步模式中，从接收 FIFO 数据寄存器 (SCFRDR) 读取的数据是否发生奇偶校验错误。</p> <p>0: 表示下一次从 SCFRDR 读取的接收数据没有发生奇偶校验错误 [清除条件]</p> <ul style="list-style-type: none"> <li>• 上电复位</li> <li>• 下一个 SCFRDR 读数据没有奇偶校验错误</li> </ul> <p>1: 表示下一次从 SCFRDR 读取的接收数据发生了奇偶校验错误 [置位条件]</p> <ul style="list-style-type: none"> <li>• 下一个 SCFRDR 读数据发生了奇偶校验错误</li> </ul>
1	RDF	0	R/(W)*	<p>接收 FIFO 数据满</p> <p>表示接收数据被传送到接收 FIFO 数据寄存器 (SCFRDR) 并且 SCFRDR 的接收数据个数大于 FIFO 控制寄存器 (SCFCR) 的 RTRG1、RTRG0 位指定的接收触发个数。</p> <p>0: 表示写到 SCFRDR 的发送数据个数小于指定的接收触发个数 [清除条件]</p> <ul style="list-style-type: none"> <li>• 上电复位</li> <li>• 在读 RDF=1 后，读 SCFRDR，使 SCFRDR 的接收数据个数小于指定接收触发个数并且给 RDF 写 0 时</li> <li>• 在通过 DTC 读 SCFRDR，使 SCFRDR 的接收数据个数小于指定触发个数时</li> </ul> <p>1: 表示 SCFRDR 的接收数据个数不小于指定的接收触发个数 [置位条件]</p> <ul style="list-style-type: none"> <li>• 当不小于指定接收触发个数的接收数据个数被保存在 SCFRDR 时 *</li> </ul> <p><b>【注】</b> * 因为 SCFRDR 是 16 字节的 FIFO 寄存器，所以当 RDF 是 1 时，能读的最大数据个数为指定的接收触发个数。在读取 SCFRDR 的全部数据后，如果继续读数据，读的值就不定。SCFDR 的高 8 位表示 SCFRDR 的接收数据个数。</p>
0	DR	0	R/(W)*	<p>接收数据就绪</p> <p>表示在异步模式中，将小于指定接收触发个数的数据保存到接收 FIFO 数据寄存器 (SCFRDR) 并且从最后的停止位经过 15ETU 的时间后还没有接收到下一个数据。在时钟同步模式中，此位不被置位。</p> <p>0: 表示正在接收或者在接收正常结束后接收数据没有留在 SCFRDR 中 [清除条件]</p> <ul style="list-style-type: none"> <li>• 上电复位</li> <li>• 在读出 DR=1 的状态后，读出 SCFRDR 内的全部接收数据并且写 0 时</li> <li>• 通过 DTC 读出 SCFRDR 内的全部接收数据时</li> </ul> <p>1: 表示没有接收到下一个接收数据 [置位条件]</p> <ul style="list-style-type: none"> <li>• 当将小于指定接收触发个数的数据保存到 SCFRDR 并且从最后的停止位经过 15ETU 的时间 * 后还没有接收到下一个数据时</li> </ul> <p><b>【注】</b> * 相当于 8 位和 1 个停止位格式的 1.5 帧。ETU (Element Time Unit: 单元时间单位)</p>

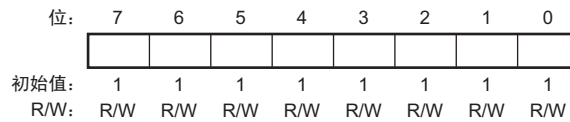
**【注】** \* 为了清除标志，只能在读 1 后写 0。



### 16.3.8 位速率寄存器 (SCBRR)

SCBRR 是 8 位寄存器，根据串行模式寄存器 (SCSMR) 的 CKS1 位或者 CKS0 位所选择的波特率发生器的运行时钟，设定串行发送和接收位速率。

CPU 随时能读写 SCBRR。在上电复位时，SCBRR 被初始化为 H'FF。



通过以下计算式求 SCBRR 的设定值:

异步模式

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

时钟同步模式

$$N = \frac{P\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

**B** : 位速率 (bit/s)

**N** : 波特率发生器的 SCBRR 设定值 ( $0 \leq N \leq 255$ )

(必须设定满足电特性的值。)

**Pφ** : 外围模块的工作频率 (MHz)

**n** : 波特率发生器的输入时钟 (n=0、1、2、3)

(n 和时钟的关系请参照表 16.3)

表 16.3 SCSMR 的设定

n	时钟	SCSMR 的设定值	
		CKS1	CKS0
0	Pφ	0	0
1	Pφ/4	0	1
2	Pφ/16	1	0
3	Pφ/64	1	1

通过以下计算式求异步模式的位速率误差:

$$\text{误差 (\%)} = \left\{ \frac{P\phi \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

异步模式的 SCBRR 设定例子如表 16.4 ~ 表 16.6、时钟同步模式的 SCBRR 设定例子如表 16.7 ~ 表 16.9 所示，使用波特率发生器时异步模式的各频率的最大位速率如表 16.10、输入外部时钟时的最大位速率如表 16.11 和表 16.12 所示。

表 16.4 位速率的 SCBRR 设定例子 (异步模式) (1)

位速率 (bit/s)	P <sub>φ</sub> (MHz)																	
	10			12			14			16			18			20		
	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)
110	2	177	-0.25	2	212	0.03	2	248	-0.17	3	70	0.03	3	79	-0.12	3	88	-0.25
150	2	129	0.16	2	155	0.16	2	181	0.16	2	207	0.16	2	233	0.16	3	64	0.16
300	2	64	0.16	2	77	0.16	2	90	0.16	2	103	0.16	2	116	0.16	2	129	0.16
600	1	129	0.16	1	155	0.16	1	181	0.16	1	207	0.16	1	233	0.16	2	64	0.16
1200	1	64	0.16	1	77	0.16	1	90	0.16	1	103	0.16	1	116	0.16	1	129	0.16
2400	0	129	0.16	0	155	0.16	0	181	0.16	0	207	0.16	0	233	0.16	1	64	0.16
4800	0	64	0.16	0	77	0.16	0	90	0.16	0	103	0.16	0	116	0.16	0	129	0.16
9600	0	32	-1.36	0	38	0.16	0	45	-0.93	0	51	0.16	0	58	-0.69	0	64	0.16
14400	0	21	-1.36	0	25	0.16	0	29	1.27	0	34	-0.79	0	38	0.16	0	42	0.94
19200	0	15	1.73	0	19	-2.34	0	22	-0.93	0	25	0.16	0	28	1.02	0	32	-1.36
28800	0	10	-1.36	0	12	0.16	0	14	1.27	0	16	2.12	0	19	-2.34	0	21	-1.36
31250	0	9	0.00	0	11	0.00	0	13	0.00	0	15	0.00	0	17	0.00	0	19	0.00
38400	0	7	1.73	0	9	-2.34	0	10	3.57	0	12	0.16	0	14	-2.34	0	15	1.73

表 16.5 位速率的 SCBRR 设定例子 (异步模式) (2)

位速率 (bit/s)	P <sub>φ</sub> (MHz)																	
	22			24			26			28			30			32		
	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)
110	3	97	-0.35	3	106	-0.44	3	114	0.36	3	123	0.23	3	132	0.13	3	141	0.03
150	3	71	-0.54	3	77	0.16	3	84	-0.43	3	90	0.16	3	97	-0.35	3	103	0.16
300	2	142	0.16	2	155	0.16	2	168	0.16	2	181	0.16	2	194	0.16	2	207	0.16
600	2	71	-0.54	2	77	0.16	2	84	-0.43	2	90	0.16	2	97	-0.35	2	103	0.16
1200	1	142	0.16	1	155	0.16	1	168	0.16	1	181	0.16	1	194	0.16	1	207	0.16
2400	1	71	-0.54	1	77	0.16	1	84	-0.43	1	90	0.16	1	97	-0.35	1	103	0.16
4800	0	142	0.16	0	155	0.16	0	168	0.16	0	181	0.16	0	194	0.16	0	207	0.16
9600	0	71	-0.54	0	77	0.16	0	84	-0.43	0	90	0.16	0	97	-0.35	0	103	0.16
14400	0	47	-0.54	0	51	0.16	0	55	0.76	0	60	-0.39	0	64	0.16	0	68	0.64
19200	0	35	-0.54	0	38	0.16	0	41	0.76	0	45	-0.93	0	48	-0.35	0	51	0.16
28800	0	23	-0.54	0	25	0.16	0	27	0.76	0	29	1.27	0	32	-1.36	0	34	-0.79
31250	0	21	0.00	0	23	0.00	0	25	0.00	0	27	0.00	0	29	0.00	0	31	0.00
38400	0	17	-0.54	0	19	-2.34	0	20	0.76	0	22	-0.93	0	23	1.73	0	25	0.16

表 16.6 位速率的 SCBRR 设定例子 (异步模式) (3)

位速率 (bit/s)	P $\phi$ (MHz)											
	34			36			38			40		
	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)
110	3	150	-0.05	3	159	-0.12	3	168	-0.19	3	177	-0.25
150	3	110	-0.29	3	116	0.16	3	123	-0.24	3	129	0.16
300	2	220	0.16	2	233	0.16	2	246	0.16	3	64	0.16
600	2	110	-0.29	2	116	0.16	2	123	-0.24	2	129	0.16
1200	1	220	0.16	1	233	0.16	1	246	0.16	2	64	0.16
2400	1	110	-0.29	1	116	0.16	1	123	-0.24	1	129	0.16
4800	0	220	0.16	0	233	0.16	0	246	0.16	1	64	0.16
9600	0	110	-0.29	0	116	0.16	0	123	-0.24	0	129	0.16
14400	0	73	-0.29	0	77	0.16	0	81	0.57	0	86	-0.22
19200	0	54	0.62	0	58	-0.69	0	61	-0.24	0	64	0.16
28800	0	36	-0.29	0	38	0.16	0	40	0.57	0	42	0.94
31250	0	33	0.00	0	35	0.00	0	37	0.00	0	39	0.00
38400	0	27	-1.18	0	28	1.02	0	30	-0.24	0	32	-1.36

表 16.7 位速率的 SCBRR 设定例子 (时钟同步模式) (1)

位速率 (bit/s)	P $\phi$ (MHz)											
	10		12		14		16		18		20	
	n	N	n	N	n	N	n	N	n	N	n	N
250	3	155	3	187	3	218	3	249				
500	3	77	3	93	3	108	3	124	3	140	3	155
1000	2	155	2	187	2	218	2	249	3	69	3	77
2500	1	249	2	74	2	87	2	99	2	112	2	124
5000	1	124	1	149	1	174	1	199	1	224	1	249
10000	0	249	1	74	1	87	1	99	1	112	1	124
25000	0	99	0	119	0	139	0	159	0	179	0	199
50000	0	49	0	59	0	69	0	79	0	89	0	99
100000	0	24	0	29	0	34	0	39	0	44	0	49
250000	0	9	0	11	0	13	0	15	0	17	0	19
500000	0	4	0	5	0	6	0	7	0	8	0	9
1000000	—	—	0	2	—	—	0	3	—	—	0	4
2500000	0	0*	—	—	—	—	—	—	—	—	0	1
5000000			—	—	—	—	—	—	—	—	0	0*

表 16.8 位速率的 SCBRR 设定例子 (时钟同步模式) (2)

位速率 (bit/s)	P $\phi$ (MHz)											
	22		24		26		28		30		32	
	n	N	n	N	n	N	n	N	n	N	n	N
250												
500	3	171	3	187	3	202	3	218	3	233	3	249
1000	3	85	3	93	3	101	3	108	3	116	3	124
2500	2	137	2	149	2	162	2	174	2	187	2	199
5000	2	68	2	74	2	80	2	87	2	93	2	99
10000	1	137	1	149	1	162	1	174	1	187	1	199
25000	0	219	0	239	1	64	1	69	1	74	1	79
50000	0	109	0	119	0	129	0	139	0	149	0	159
100000	0	54	0	59	0	64	0	69	0	74	0	79
250000	0	21	0	23	0	25	0	27	0	29	0	31
500000	0	10	0	11	0	12	0	13	0	14	0	15
1000000	—	—	0	5	—	—	0	6	—	—	0	7
2500000	—	—	—	—	—	—	—	—	0	2	—	—
5000000	—	—	—	—	—	—	—	—	—	—	—	—

表 16.9 位速率的 SCBRR 设定例子 (时钟同步模式) (3)

位速率 (bit/s)	P <sub>φ</sub> (MHz)							
	34		36		38		40	
	n	N	n	N	n	N	n	N
250								
500								
1000	3	132	3	140	3	147	3	155
2500	2	212	2	224	2	237	2	249
5000	2	105	2	112	2	118	2	124
10000	1	212	1	224	1	237	1	249
25000	1	84	1	89	1	94	1	99
50000	0	169	0	179	0	189	0	199
100000	0	84	0	89	0	94	0	99
250000	0	33	0	35	0	37	0	39
500000	0	16	0	17	0	18	0	19
1000000	—	—	0	8	—	—	0	9
2500000	—	—	—	—	—	—	0	3
5000000	—	—	—	—	—	—	0	1

【注】 必须尽量将误差设定在 1% 以内。

【符号说明】

空白栏：不能设定。

— ：能设定，但是会出现误差。

\* ：不能连续发送和接收。

表 16.10 使用波特率发生器时各频率的最大位速率 (异步模式)

P $\phi$ (MHz)	最大位速率 (bit/s)	设定值	
		n	N
10	312500	0	0
12	375000	0	0
14	437500	0	0
16	500000	0	0
18	562500	0	0
20	625000	0	0
22	687500	0	0
24	750000	0	0
26	812500	0	0
28	875000	0	0
30	937500	0	0
32	1000000	0	0
34	1062500	0	0
36	1125000	0	0
38	1187500	0	0
40	1250000	0	0

表 16.11 输入外部时钟时的最大位速率 (异步模式)

P $\phi$ (MHz)	外部输入时钟 (MHz)	最大位速率 (bit/s)
10	2.5000	156250
12	3.0000	187500
14	3.5000	218750
16	4.0000	250000
18	4.5000	281250
20	5.0000	312500
22	5.5000	343750
24	6.0000	375000
26	6.5000	406250
28	7.0000	437500
30	7.5000	468750
32	8.0000	500000
34	8.5000	531250
36	9.0000	562500
38	9.5000	593750
40	10.0000	625000

表 16.12 输入外部时钟时的最大位速率 (时钟同步模式)

P $\phi$ (MHz)	外部输入时钟 (MHz)	最大位速率 (bit/s)
10	1.6667	1666666.7
12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0
20	3.3333	3333333.3
22	3.6667	3666666.7
24	4.0000	4000000.0
26	4.3333	4333333.3
28	4.6667	4666666.7
30	5.0000	5000000.0
32	5.3333	5333333.3
34	5.6667	5666666.7
36	6.0000	6000000.0
38	6.3333	6333333.3
40	6.6667	6666666.7



## 16.3.9 FIFO 控制寄存器 (SCFCR)

SCFCR 是 16 位寄存器，对发送 FIFO 数据寄存器和接收 FIFO 数据寄存器的数据个数进行复位以及设定触发数据个数，SCFCR 还含有回送测试的允许位。

CPU 总能读写 SCFCR。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	RSTRG[2:0]			RTRG[1:0]		TTRG[1:0]		MCE	TFRST	RFRST	LOOP
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 11	—	全 0	R	保留位 读写的值总是 0。
10 ~ 8	RSTRG[2:0]	000	R/W	RTS 输出的有效触发 1、0 当保存在接收 FIFO 数据寄存器 (SCFRDR) 的接收数据个数不小于以下所示的触发设定个数时，RTS 信号为高电平。 此位只在异步模式中允许调制解调器信号时有效。 000: 15 001: 1 010: 4 011: 6 100: 8 101: 10 110: 12 111: 14
7、6	RTRG[1:0]	00	R/W	接收 FIFO 数据个数的触发 1、0 设定接收数据个数 (指定接收触发个数)，作为串行状态寄存器 (SCFSR) 的 RDF 标志置位的基准。 如果保存在接收 FIFO 数据寄存器 (SCFRDR) 的接收数据个数不小于以下所示的设定触发个数，就将 RDF 标志置位。 • 异步模式                      • 时钟同步模式 00: 1                              00: 1 01: 4                              01: 2 10: 8                              10: 8 11: 14                             11: 14
5、4	TTRG[1:0]	00	R/W	发送 FIFO 数据个数的触发 1、0 设定发送数据个数 (指定发送触发个数)，作为串行状态寄存器 (SCFSR) 的 TDFE 标志的置位基准。 如果保存到发送 FIFO 数据寄存器 (SCFTDR) 的发送数据个数不大于以下所示的设定触发个数，就将 TDFE 标志置位。 00: 8 (8) * 01: 4 (12) * 10: 2 (14) * 11: 0 (16) * 【注】* ( ) 内的值表示 TDFE 标志被置位时的 SCFTDR 寄存器的空字节数。

位	位名	初始值	R/W	说 明
3	MCE	0	R/W	调制解调器的控制允许 允许或者禁止调制解调器的控制信号 $\overline{\text{CTS}}$ 和 $\overline{\text{RTS}}$ 。 在时钟同期模式中，MCE 必须总是为 0。 0: 禁止调制解调器信号 * 1: 允许调制解调器信号 【注】* 与输入值无关， $\overline{\text{CTS}}$ 和 $\overline{\text{RTS}}$ 都被固定为 0 (低电平)。
2	TFRST	0	R/W	发送 FIFO 数据寄存器的复位 将发送 FIFO 数据寄存器复位成空状态 (使寄存器内的发送数据无效)。 0: 禁止复位 * 1: 允许复位 【注】* 在上电复位时进行复位。
1	RFRST	0	R/W	接收 FIFO 数据寄存器的复位 将接收 FIFO 数据寄存器复位成空状态 (使寄存器内的接收数据无效)。 0: 禁止复位 * 1: 允许复位 【注】* 在上电复位时，进行复位。
0	LOOP	0	R/W	回送测试 (LOOP) 能在内部将发送输出引脚 (TXD) 和接收输入引脚 (RXD)、 $\overline{\text{RTS}}$ 引脚和 $\overline{\text{CTS}}$ 引脚连接，进行回送测试。 0: 禁止回送测试 1: 允许回送测试

### 16.3.10 FIFO 数据个数寄存器 (SCFDR)

SCFDR 是 16 位寄存器，表示保存在发送 FIFO 数据寄存器 (SCFTDR) 和接收 FIFO 数据寄存器 (SCFRDR) 的数据个数。

高 8 位表示 SCFTDR 发送数据的个数，低 8 位表示 SCFRDR 的接收数据的个数。CPU 随时能读 SCFDR。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	T[4:0]				-	-	-	R[4:0]					
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 13	—	全 0	R	保留位 读写的值总是 0。
12 ~ 8	T[4:0]	00000	R	表示保存在 SCFTDR 的未发送数据的个数。 H'00 表示 SCFTDR 中没有发送数据，H'10 表示全部发送数据保存在 SCFTDR。
7 ~ 5	—	全 0	R	保留位 读写的值总是 0。
4 ~ 0	R[4:0]	00000	R	表示保存在 SCFRDR 的接收数据的个数。 H'00 表示 SCFRDR 中没有接收数据，H'10 表示全部接收数据保存在 SCFRDR。

### 16.3.11 串行端口寄存器 (SCSPTR)

SCSPTR 是 16 位寄存器，控制被多路复用引脚 SCIF 的端口输入 / 输出引脚和数据。能通过 bit7 和 bit6、bit5 和 bit4、bit3 和 bit2 分别控制 RTS 引脚、CTS 引脚和 SCK 引脚，通过 bit1 或者 bit0 将输出数据写到 TXD 引脚，并能控制串行发送和接收的中止。

请配合以下对各位的说明，参照“16.6 串行端口寄存器 (SCSPTR) 和 SCIF 引脚的关系”。

CPU 随时能读写 SCSPTR。如果要读 SCIF 引脚的值，就必须使用端口寄存器，详细内容请参照“第 22 章 I/O 端口”。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPBIO	SPBDT
初始值:	0	0	0	0	0	0	0	0	0	不定	0	不定	0	不定	0	不定
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明																				
15 ~ 8	—	全 0	R	保留位 读写的值总是 0。																				
7	RTSIO	0	R/W	RTS 端口的输出指定 与 RTSDT 位和 SCFCR 的 MCE 位一起控制 $\overline{\text{RTS}}$ 引脚。																				
6	RTSDT	不定	R/W	RTS 端口数据 与 RTSIO 位和 SCFCR 的 MCE 位一起控制 $\overline{\text{RTS}}$ 引脚。 但是需要通过 PFC (引脚功能控制器) 选择 $\overline{\text{RTS}}$ 引脚功能。 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>SCFCR 的 MCE 位的设定值</th><th>RTSIO 位的设定值</th><th>RTSDT 位的设定值</th><th><math>\overline{\text{RST}}</math> 引脚状态</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>*</td><td>禁止设定 (初始状态)</td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>输出低电平</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>输出高电平</td></tr> <tr> <td>1</td><td>*</td><td>*</td><td>根据调制解调器控制逻辑, 进行顺序输出</td></tr> </tbody> </table> <p>【注】* Don't care</p>	SCFCR 的 MCE 位的设定值	RTSIO 位的设定值	RTSDT 位的设定值	$\overline{\text{RST}}$ 引脚状态	0	0	*	禁止设定 (初始状态)	0	1	0	输出低电平	0	1	1	输出高电平	1	*	*	根据调制解调器控制逻辑, 进行顺序输出
SCFCR 的 MCE 位的设定值	RTSIO 位的设定值	RTSDT 位的设定值	$\overline{\text{RST}}$ 引脚状态																					
0	0	*	禁止设定 (初始状态)																					
0	1	0	输出低电平																					
0	1	1	输出高电平																					
1	*	*	根据调制解调器控制逻辑, 进行顺序输出																					
5	CTSIO	0	R/W	CTS 端口的输入 / 输出指定 与 CTSDT 位和 SCFCR 的 MCE 位一起控制 $\overline{\text{CTS}}$ 引脚。																				
4	CTSDT	不定	R/W	CTS 端口数据 与 CTSIO 位和 SCFCR 的 MCE 位一起控制 $\overline{\text{CTS}}$ 引脚。 但是需要通过 PFC (引脚功能控制器) 选择 $\overline{\text{CTS}}$ 引脚功能。 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>SCFCR 的 MCE 位的设定值</th><th>CTSIO 位的设定值</th><th>CTSDT 位的设定值</th><th><math>\overline{\text{CTS}}</math> 引脚状态</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>*</td><td>禁止设定 (初始状态)</td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>输出低电平</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>输出高电平</td></tr> <tr> <td>1</td><td>*</td><td>*</td><td>输入到调制解调器控制逻辑</td></tr> </tbody> </table> <p>【注】* Don't care</p>	SCFCR 的 MCE 位的设定值	CTSIO 位的设定值	CTSDT 位的设定值	$\overline{\text{CTS}}$ 引脚状态	0	0	*	禁止设定 (初始状态)	0	1	0	输出低电平	0	1	1	输出高电平	1	*	*	输入到调制解调器控制逻辑
SCFCR 的 MCE 位的设定值	CTSIO 位的设定值	CTSDT 位的设定值	$\overline{\text{CTS}}$ 引脚状态																					
0	0	*	禁止设定 (初始状态)																					
0	1	0	输出低电平																					
0	1	1	输出高电平																					
1	*	*	输入到调制解调器控制逻辑																					

位	位名	初始值	R/W	说 明																																																							
3	SCKIO	0	R/W	SCK 端口的输入 / 输出指定 与 SCKDT 位、SCSMR 的 C/A 位、SCSCR 的 CK1 位和 CK0 位一起控制 SCK 引脚。																																																							
2	SCKDT	不定	R/W	<p>SCK 端口数据 与 SCKIO 位、SCSMR 的 C/A 位、SCSCR 的 CK1 位和 CK0 位一起控制 SCK 引脚。 但是需要通过 PFC (引脚功能控制器) 选择 SCK 引脚功能。</p> <table border="1"> <thead> <tr> <th>SCSMR 的 C/A 位的设定值</th> <th>SCSCR 的 CK1 和 CK0 位的设定值</th> <th>SCKIO 位的设定值</th> <th>SCKDT 位的设定值</th> <th>SCK 引脚状态</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>00</td> <td>0</td> <td>*</td> <td>禁止设定 (初始状态)</td> </tr> <tr> <td>0</td> <td>00</td> <td>1</td> <td>0</td> <td>输出低电平</td> </tr> <tr> <td>0</td> <td>00</td> <td>1</td> <td>1</td> <td>输出高电平</td> </tr> <tr> <td>0</td> <td>01</td> <td>*</td> <td>*</td> <td>根据串行核心逻辑, 输出内部时钟</td> </tr> <tr> <td>0</td> <td>10</td> <td>*</td> <td>*</td> <td>将外部时钟输入到串行核心逻辑</td> </tr> <tr> <td>0</td> <td>11</td> <td>*</td> <td>*</td> <td>禁止设定</td> </tr> <tr> <td>1</td> <td>00</td> <td>*</td> <td>*</td> <td>根据串行核心逻辑, 输出内部时钟</td> </tr> <tr> <td>1</td> <td>01</td> <td>*</td> <td>*</td> <td>根据串行核心逻辑, 输出内部时钟</td> </tr> <tr> <td>1</td> <td>10</td> <td>*</td> <td>*</td> <td>将外部时钟输入到串行核心逻辑</td> </tr> <tr> <td>1</td> <td>11</td> <td>*</td> <td>*</td> <td>禁止设定</td> </tr> </tbody> </table> <p>【注】* Don't care</p>	SCSMR 的 C/A 位的设定值	SCSCR 的 CK1 和 CK0 位的设定值	SCKIO 位的设定值	SCKDT 位的设定值	SCK 引脚状态	0	00	0	*	禁止设定 (初始状态)	0	00	1	0	输出低电平	0	00	1	1	输出高电平	0	01	*	*	根据串行核心逻辑, 输出内部时钟	0	10	*	*	将外部时钟输入到串行核心逻辑	0	11	*	*	禁止设定	1	00	*	*	根据串行核心逻辑, 输出内部时钟	1	01	*	*	根据串行核心逻辑, 输出内部时钟	1	10	*	*	将外部时钟输入到串行核心逻辑	1	11	*	*	禁止设定
SCSMR 的 C/A 位的设定值	SCSCR 的 CK1 和 CK0 位的设定值	SCKIO 位的设定值	SCKDT 位的设定值	SCK 引脚状态																																																							
0	00	0	*	禁止设定 (初始状态)																																																							
0	00	1	0	输出低电平																																																							
0	00	1	1	输出高电平																																																							
0	01	*	*	根据串行核心逻辑, 输出内部时钟																																																							
0	10	*	*	将外部时钟输入到串行核心逻辑																																																							
0	11	*	*	禁止设定																																																							
1	00	*	*	根据串行核心逻辑, 输出内部时钟																																																							
1	01	*	*	根据串行核心逻辑, 输出内部时钟																																																							
1	10	*	*	将外部时钟输入到串行核心逻辑																																																							
1	11	*	*	禁止设定																																																							
1	SPBIO	0	R/W	串行端口中止的输出指定 与 SPBDT 位和 SCSCR 的 TE 位一起控制 TXD 引脚。																																																							
0	SPBDT	不定	R/W	<p>串行端口的中止数据 与 SPBIO 位和 SCSCR 的 TE 位一起控制 TXD 引脚。 但是需要通过 PFC (引脚功能控制器) 选择 TXD 引脚功能。</p> <table border="1"> <thead> <tr> <th>SCSCR 的 TE 位的设定值</th> <th>SPBIO 位的设定值</th> <th>SPBDT 位的设定值</th> <th>TXD 引脚状态</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>*</td> <td>禁止设定 (初始状态)</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>输出低电平</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>输出高电平</td> </tr> <tr> <td>1</td> <td>*</td> <td>*</td> <td>根据串行核心逻辑, 输出发送数据</td> </tr> </tbody> </table> <p>【注】* Don't care</p>	SCSCR 的 TE 位的设定值	SPBIO 位的设定值	SPBDT 位的设定值	TXD 引脚状态	0	0	*	禁止设定 (初始状态)	0	1	0	输出低电平	0	1	1	输出高电平	1	*	*	根据串行核心逻辑, 输出发送数据																																			
SCSCR 的 TE 位的设定值	SPBIO 位的设定值	SPBDT 位的设定值	TXD 引脚状态																																																								
0	0	*	禁止设定 (初始状态)																																																								
0	1	0	输出低电平																																																								
0	1	1	输出高电平																																																								
1	*	*	根据串行核心逻辑, 输出发送数据																																																								

## 16.3.12 线路状态寄存器 (SCLSR)

SCLSR 是随时能由 CPU 读写的 16 位寄存器。但是不能给 ORER 的状态标志写 1。为了清 0，必须先读 1。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	ORER
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/(W)*

【注】\* 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说 明
15 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	ORER	0	R/(W)*	溢出错误 表示在接收时发生了溢出错误而异常结束。 0: 表示正在接收或者接收已正常结束 *1 [清除条件] • 上电复位 • 在读 ORER=1 的状态后写 0 时 1: 表示在接收时发生了溢出错误 *2 [置位条件] • 在接收 FIFO 满的状态下，结束下一次串行接收时 【注】 *1 在将串行控制寄存器 (SCSCR) 的 RE 位清 0 时，ORER 标志不受影响而保持原来的状态。 *2 接收 FIFO 数据寄存器 (SCFRDR) 保持发生溢出错误前的接收数据，丢失后来接收的数据。另外，在设定 ORER=1 的状态下，不能继续以后的串行接收。

## 16.4 运行说明

### 16.4.1 概要

SCIF 能以异步模式（以字符为单位边同步边通信）和时钟同步模式（通过时钟脉冲边同步边通信）2 种方式进行串行通信。

SCIF 内置了分别用于接收和发送的 16 段 FIFO 寄存器，能减少 CPU 的开销以及能连续进行高速通信。另外，还备有用作调制解调器控制信号的  $\overline{\text{RTS}}$  和  $\overline{\text{CTS}}$  信号。

通过串行模式寄存器（SCSMR）选择发送和接收格式，如表 16.13 所示。SCIF 时钟源取决于串行控制寄存器（SCSCR）的 CKE1 位和 CKE0 位的组合，如表 16.14 所示。

#### (1) 异步模式

- 数据长：可选择 7 位或者 8 位  
可选择附加奇偶校验和附加 1 位 / 2 位的停止位（根据这些组合，决定发送 / 接收格式和字符长）
- 在接收时，能检测帧错误、奇偶校验错误、接收 FIFO 数据满、溢出错误、接收数据就绪和中止
- 表示各发送和接收 FIFO 寄存器所保存的数据个数
- SCIF 的时钟源：可选择内部时钟或者外部时钟  
当选择内部时钟时，通过波特率发生器时钟运行 SCIF  
当选择外部时钟时，需要输入频率为 16 倍位速率的时钟（不使用内部波特率发生器）

#### (2) 时钟同步模式

- 发送 / 接收格式：固定为 8 位数据
- 在接收时，能检测溢出错误
- SCIF 的时钟源：可选择内部时钟或者外部时钟  
当选择内部时钟时，通过波特率发生器的时钟运行 SCIF，并且将同步时钟输出到外部  
当选择外部时钟时，不使用内部波特率发生器而使用输入的同步时钟运行 SCIF

表 16.13 SCSMR 的设定值和 SCIF 的发送 / 接收格式

SCSMR 的设定值				模式	SCIF 的发送 / 接收格式		
bit7	bit6	bit5	bit3		数据长	奇偶校验位	停止位长
$\overline{\text{C/A}}$	CHR	PE	STOP				
0	0	0	0	异步模式	8 位	无	1 位
			1				2 位
		1	0			有	1 位
			1				2 位
	1	0	0		7 位	无	1 位
			1				2 位
		1	0			有	1 位
			1				2 位
1	x	x	x	时钟同步模式	8 位	无	无

【符号说明】 x: Don't care

表 16.14 SCSMR、SCSCR 的设定和 SCIF 时钟源的选择

SCSMR	SCSCR 的设定值		模式	时钟源	SCK 引脚的功能
	bit7	bit1			
C/A	CKE1	CKE0			
0	0	0	异步模式	内部	SCIF 不使用 SCK 引脚。SCK 引脚状态取决于 SCSPTR 的 SCKIO 位和 SCKDT 位。
		1			输出位速率的 16 倍频率时钟
	1	0		外部	输入位速率的 16 倍频率时钟
		1		—	禁止设定
1	0	x	时钟同步模式	内部	输出同步时钟
		1		外部	输入同步时钟
	1	—		禁止设定	

【符号说明】 x: Don't care

### 16.4.2 异步模式中的运行

异步模式是以字符为单位边同步边进行串行通信的模式，将表示通信开始的起始位和表示通信结束的停止位附加到数据后进行字符的发送和接收。

在 SCIF 内部发送部和接收部独立，所以能进行全双工通信，而且发送部和接收部都是 16 段 FIFO 缓冲结构，所以能在发送和接收中读写数据，进行连续的发送和接收。异步模式串行通信的一般格式如图 16.2 所示。

在异步串行通信中，通信线路通常保持为标记状态（高电平）。SCIF 监视通信线路，将空状态（低电平）视为起始位，开始串行通信。

以起始位（低电平）开始，数据（LSB 先：从最低位开始）、奇偶校验位（高 / 低电平），最后为停止位（高电平）的顺序构成串行通信的 1 个字符。

在异步模式中，SCIF 在接收时通过起始位的下降沿取得同步。另外，因为 SCIF 在 1 个为 16 倍位速率频率的第 8 个时钟采样数据，所以在各位的中央取得通信数据。

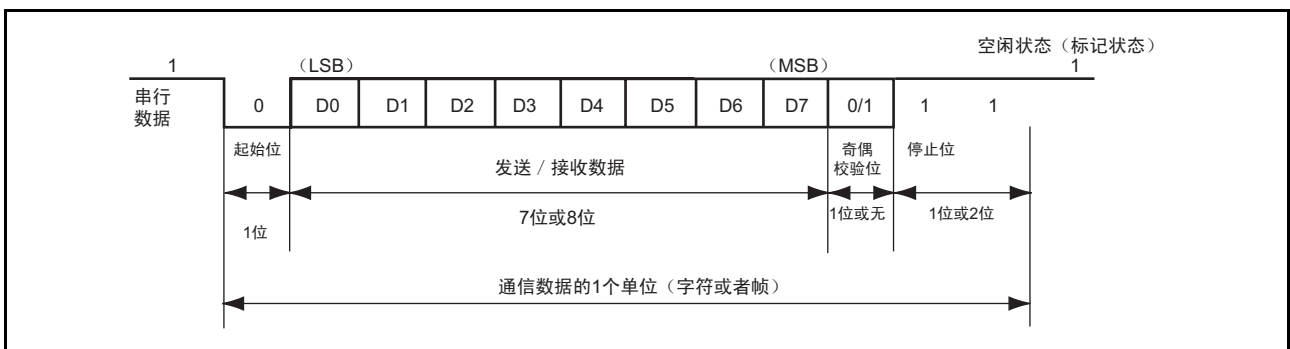


图 16.2 异步通信的数据格式  
(8 位数据 / 有奇偶校验 / 2 个停止位的例子)

## (1) 发送 / 接收格式

能设定的发送 / 接收格式如表 16.15 所示。

发送 / 接收格式有 8 种，能通过串行模式寄存器 (SCSMR) 的设定进行选择。

表 16.15 串行发送 / 接收格式 (异步模式)

SCSMR的设定			串行发送 / 接收格式和帧长													
CHR	PE	STOP	1	2	3	4	5	6	7	8	9	10	11	12		
0	0	0	START	8位数据								STOP				
0	0	1	START	8位数据								STOP	STOP			
0	1	0	START	8位数据								P	STOP			
0	1	1	START	8位数据								P	STOP	STOP		
1	0	0	START	7位数据							STOP					
1	0	1	START	7位数据							STOP	STOP				
1	1	0	START	7位数据							P	STOP				
1	1	1	START	7位数据							P	STOP	STOP			

## 【符号说明】

START: 起始位

STOP : 停止位

P : 奇偶校验位

## (2) 时钟

能通过设定 SCSMR 的  $\overline{C/A}$  位和串行控制寄存器 (SCSCR) 的 CKE1 位、CKE0 位，从内部波特率发生器生成的内部时钟或者 SCK 引脚输入的外部时钟中选择 SCIF 的发送和接收时钟。有关 SCIF 时钟源的选择请参照表 16.14。

在给 SCK 引脚输入外部时钟时，必须输入频率是 16 倍位速率的时钟。

在通过内部时钟运行时，能从 SCK 引脚输出时钟。此时，输出的时钟频率是位速率的 16 倍。



## (3) 数据的发送和接收运行

## • SCIF 的初始化 (异步模式)

在发送和接收数据前, 必须首先将串行控制寄存器 (SCSCR) 的 TE 位和 RE 位清 0, 然后按照以下顺序初始化 SCIF。

在更改运行模式和通信格式等时, 也必须在将 TE 位和 RE 位清 0 后, 按照以下步骤进行。如果将 TE 位清 0, 发送移位寄存器 (SCTSR) 就被初始化。必须注意: 即使将 TE 和 RE 位清 0, 串行状态寄存器 (SCFSR)、发送 FIFO 数据寄存器 (SCFTDR) 和接收 FIFO 数据寄存器 (SCFRDR) 也不初始化而保持内容。必须在发送全部的发送数据并且 SCFSR 的 TEND 标志被置位后, 将 TE 位清 0。即使在发送中也能将 TE 位清 0, 但在清 0 后发送数据为标记状态。另外, 在将 TE 位重新设定为 1 开始发送前, 必须在将 SCFCR 的 TFRST 位设定为 1 后, 再将 SCFTDR 复位。

在使用外部时钟时, 不能在含有初始化的运行中停止时钟, 否则运行就不稳定。SCIF 的初始化流程图例子如图 16.3 所示。

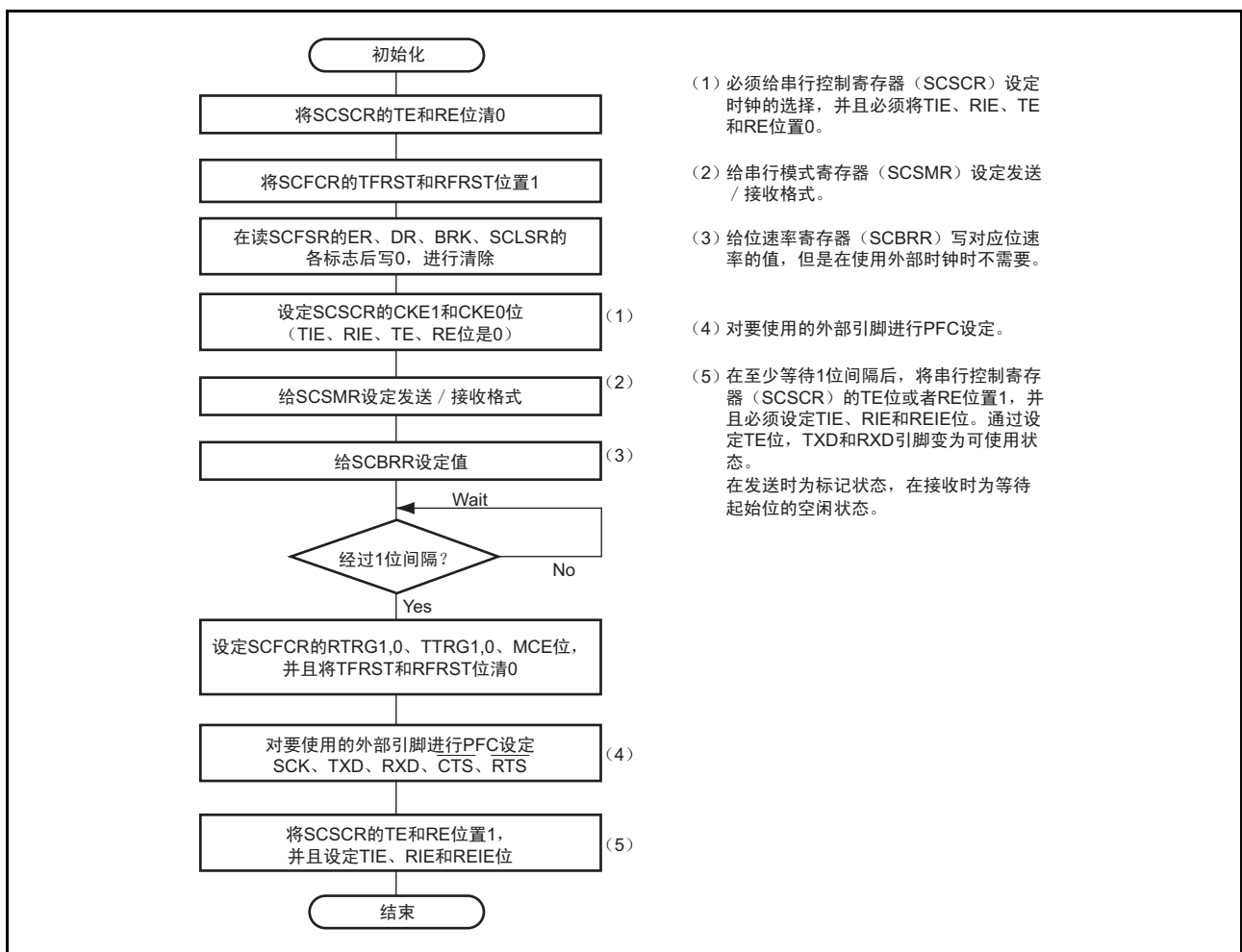


图 16.3 SCIF 的初始化流程图例子

• 串行数据发送 (异步模式)

串行发送的流程图例子如图 16.4 所示。

在将 SCIF 设定为可发送的运行状态后, 必须按照以下步骤发送串行数据。

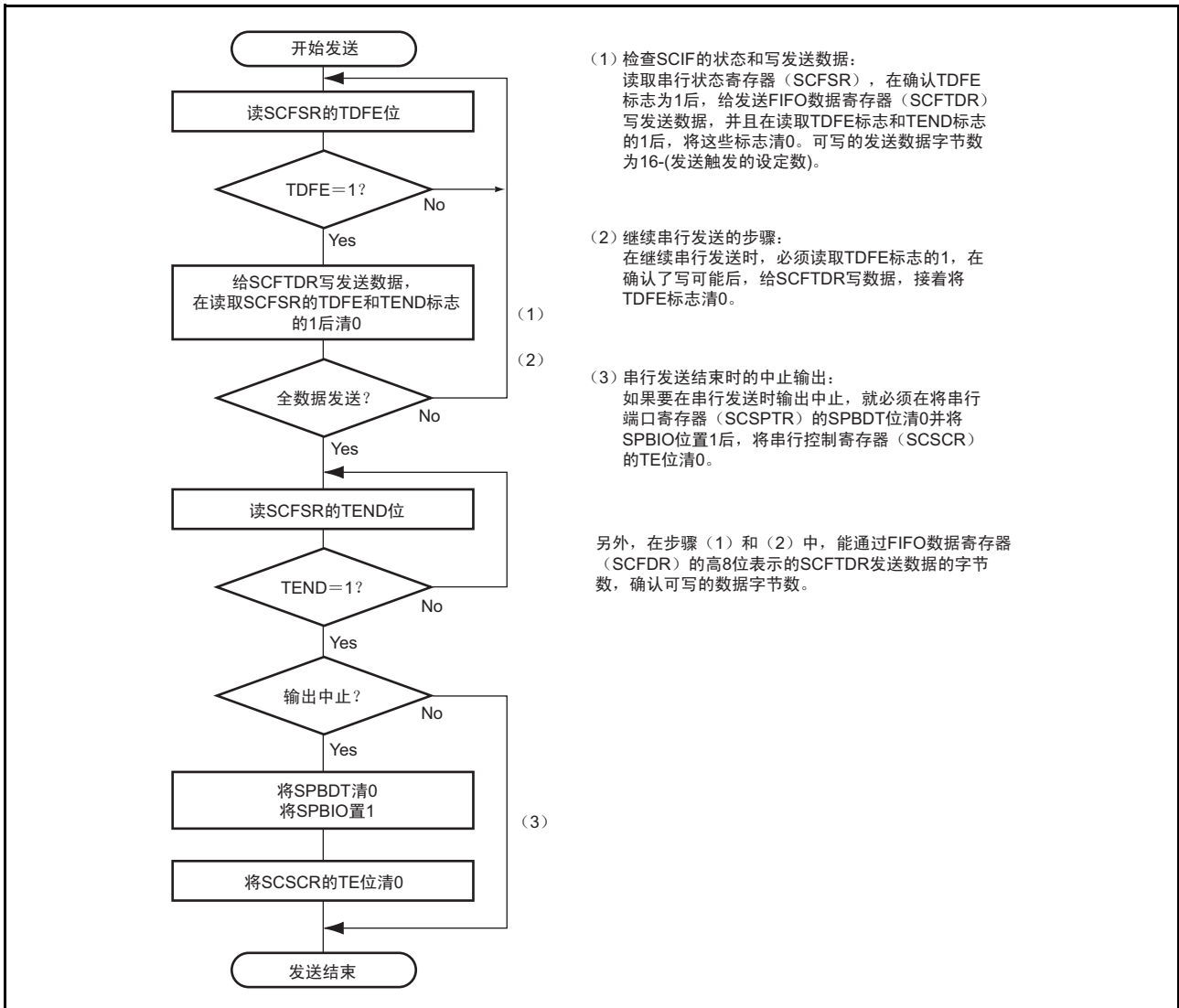


图 16.4 串行发送的流程图例子

串行发送时的 SCIF 运行如下：

1. 如果将数据写到发送 FIFO 数据寄存器 (SCFTDR)，SCIF 就将数据从 SCFTDR 传送到发送移位寄存器 (SCTSR)。在将发送数据写到 SCFTDR 前，必须确认串行状态寄存器 (SCFSR) 的 TDFE 标志是 1。能写的发送数据字节数是 (16 - 发送触发设定数)。
2. 如果将数据从 SCFTDR 传送到 SCTSR 后开始发送，就进行连续发送，直到 SCFTDR 中没有发送数据为止。如果 SCFTDR 的发送数据字节数不大于 FIFO 控制寄存器 (SCFCR) 设定的发送触发个数，就将 TDFE 标志置位。此时，如果串行控制寄存器 (SCSCR) 的 TIE 位已被置 1，就产生发送 FIFO 数据空的中断 (TXIF) 请求。

按照以下顺序，从 TXD 引脚发送串行发送数据。

- (a) 起始位：输出 1 位的 0。
  - (b) 发送数据：从 LSB 开始，按顺序输出 8 位或者 7 位数据。
  - (c) 输出奇偶校验位 (偶校验或者奇校验)。  
另外，也能选择不输出奇偶校验位的格式。
  - (d) 停止位：输出 1 位或者 2 位的 1 (停止位)。
  - (e) 标记状态：在送出开始下一次发送的起始位前连续输出 1。
3. SCIF 在送出停止位时检查 SCFTDR 的发送数据。如果有数据，就将数据从 SCFTDR 传送到 SCTSR，在送出停止位后，开始下一帧的串行发送；如果没有发送数据，就将 SCFSR 的 TEND 标志置 1，在送出停止位后变为连续输出 1 的标记状态。

异步模式发送时的运行例子如图 16.5 所示。

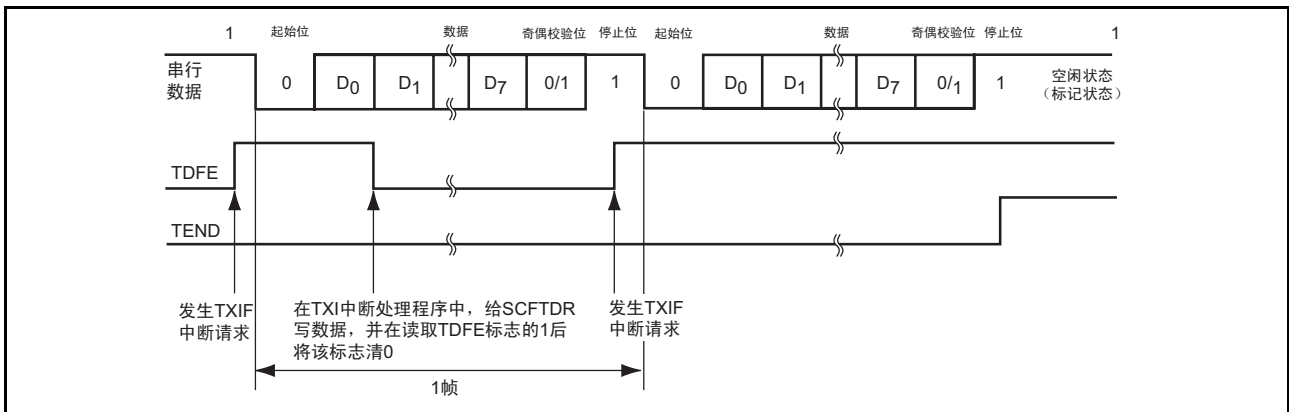


图 16.5 发送时的运行例 (8 位数据 / 有奇偶校验 / 1 个停止位)

4. 如果允许调制解调器控制，就能通过  $\overline{\text{CTS}}$  输入值停止或者重新开始发送。当  $\overline{\text{CTS}}$  被置 1 时，如果正在发送，就在 1 帧发送结束后进入标记状态；当  $\overline{\text{CTS}}$  被置 0 时，下一个发送数据就从起始位开始输出。使用调制解调器控制的运行例子如图 16.6 所示。

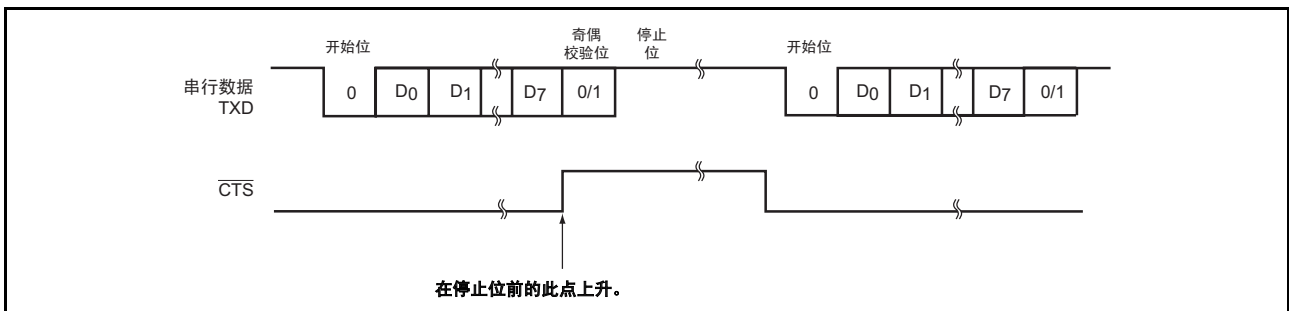


图 16.6 使用调制解调器控制的运行例 ( $\overline{\text{CTS}}$ )

• 串行数据的接收 (异步模式)

串行接收的流程图例子如图 16.7 和图 16.8 所示。

在将 SCIF 设定为可接收的运行状态后, 必须按照以下步骤接收串行数据。

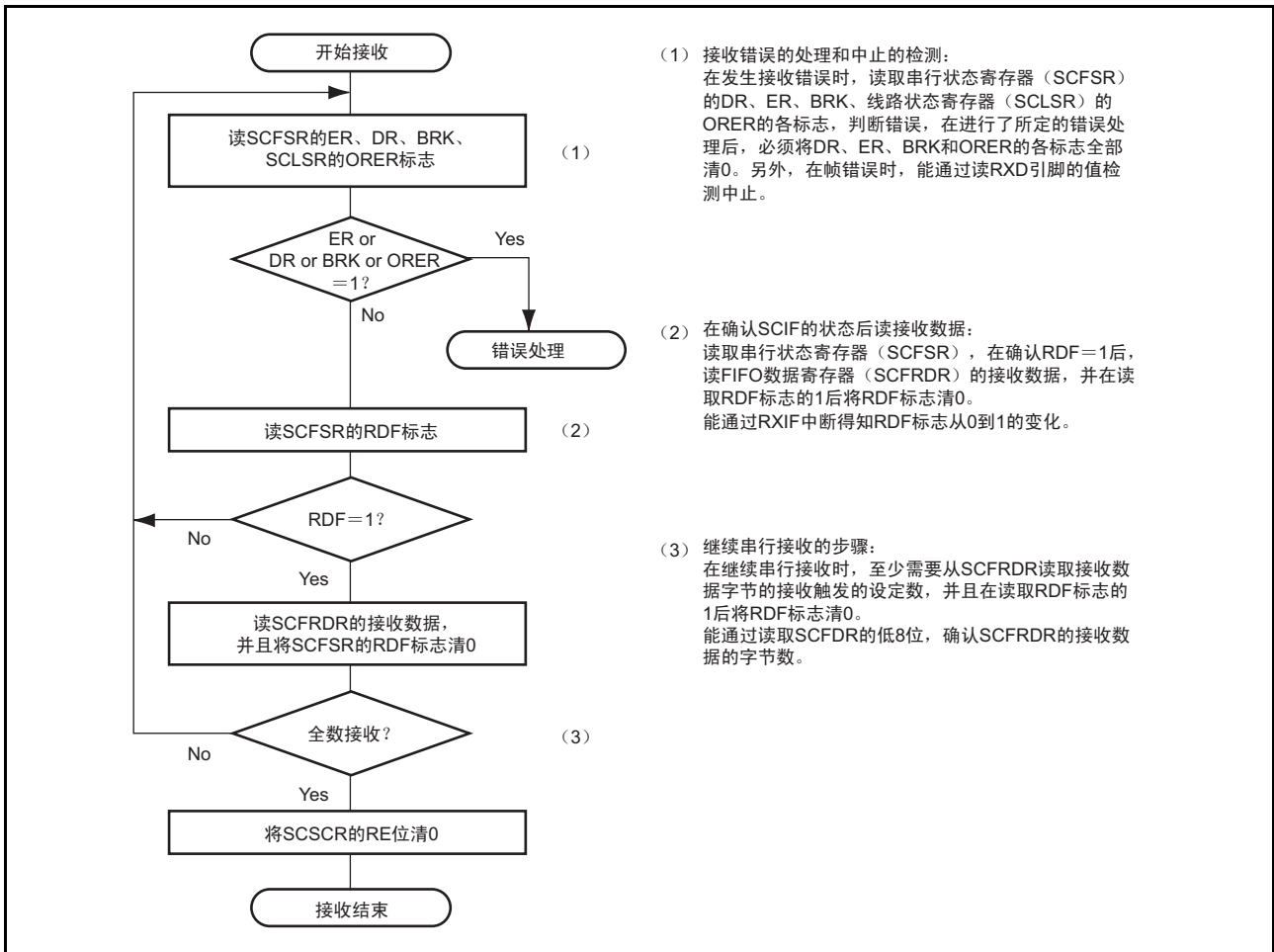
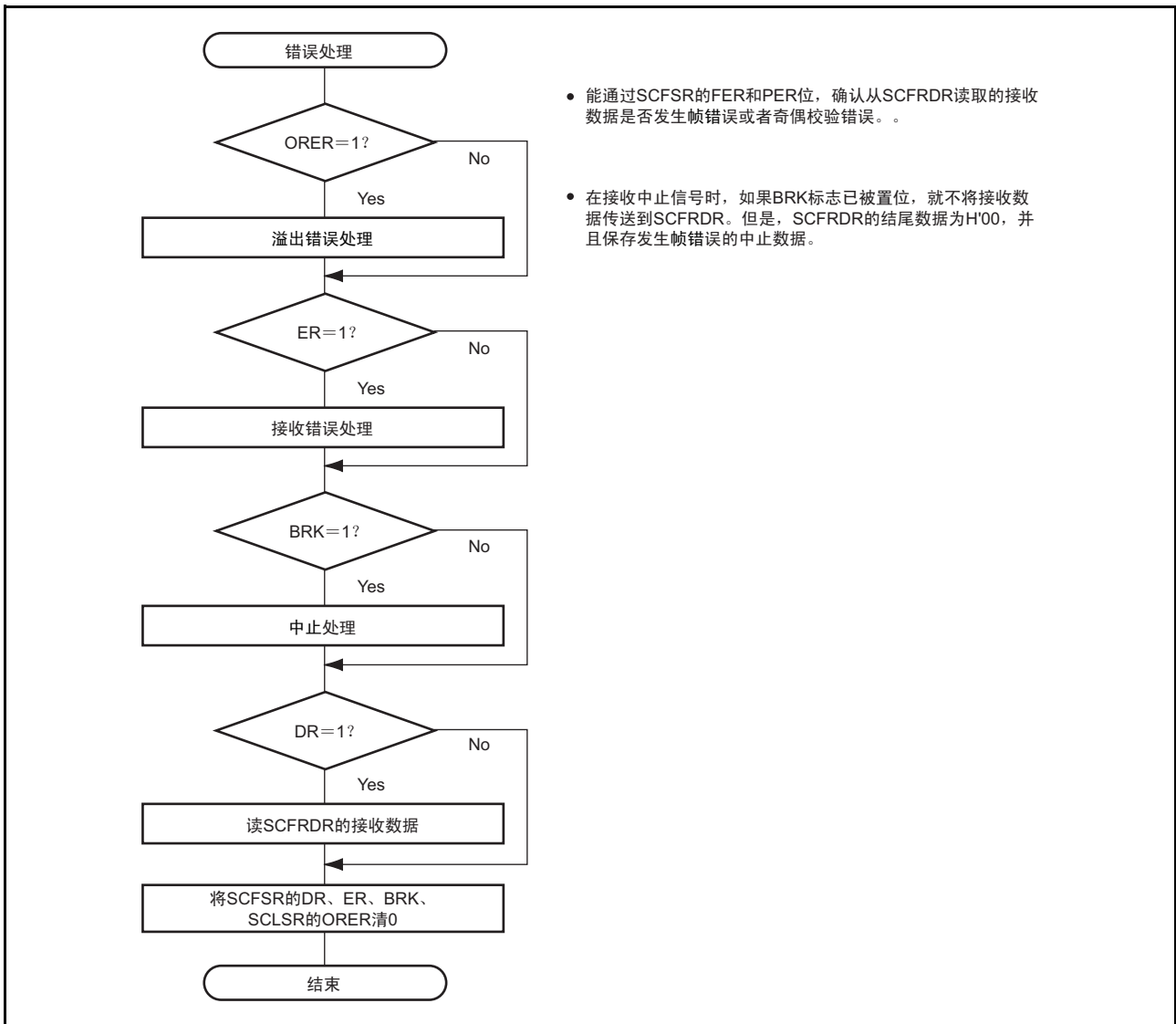


图 16.7 串行接收的流程图例子 (1)



- 能通过SCFSR的FER和PER位，确认从SCFRDR读取的接收数据是否发生帧错误或者奇偶校验错误。。
- 在接收中止信号时，如果BRK标志已被置位，就不将接收数据传送到SCFRDR。但是，SCFRDR的结尾数据为H'00，并且保存发生帧错误的中止数据。

图 16.8 串行接收的流程图例子 (2)

接收时的 SCIF 运行如下:

1. SCIF 监视通信线路, 如果检测到起始位的 0, 就与内部同步开始接收。
2. 按照从 RSR 的 LSB 到 MSB 的顺序保存已接收的数据。
3. 接收奇偶校验位和停止位。

在接收后, SCI 进行以下检查:

- (a) 停止位的检查: 检查停止位是否为 1。  
但是在 2 个停止位的情况下, 只检查第 1 个停止位。
- (b) 检查是否为能将接收数据从接收移位寄存器 (SCRSR) 传送到 SCFRDR 的状态。
- (c) 溢出错误的检查: 检查 OREER 标志是否为 0 (表示没有发生溢出错误)。
- (d) 中止的检查: 检查 BRK 标志是否为 0 (表示没有设定中止状态)。

在以上检查全部通过后, 将接收数据保存到 SCFRDR。

**【注】** 即使发生奇偶校验错误或者帧错误, 也能继续接收。

4. 在 RDF 标志或者 DR 标志是 1 时, 如果 SCSCR 的 RIE 位已被置 1, 就产生接收 FIFO 数据满的中断 (RXIF) 请求; 在 ER 标志是 1 时, 如果 SCSCR 的 RIE 位或者 REIE 位已被置 1, 就产生接收错误的中断 (ERIF) 请求; 在 BRK 标志或者 OREER 标志是 1 时, 如果 SCSCR 的 RIE 位或者 REIE 位已被置 1, 就产生中止接收的中断 (BRIF) 请求。

异步模式接收时的运行例子如图 16.9 所示。

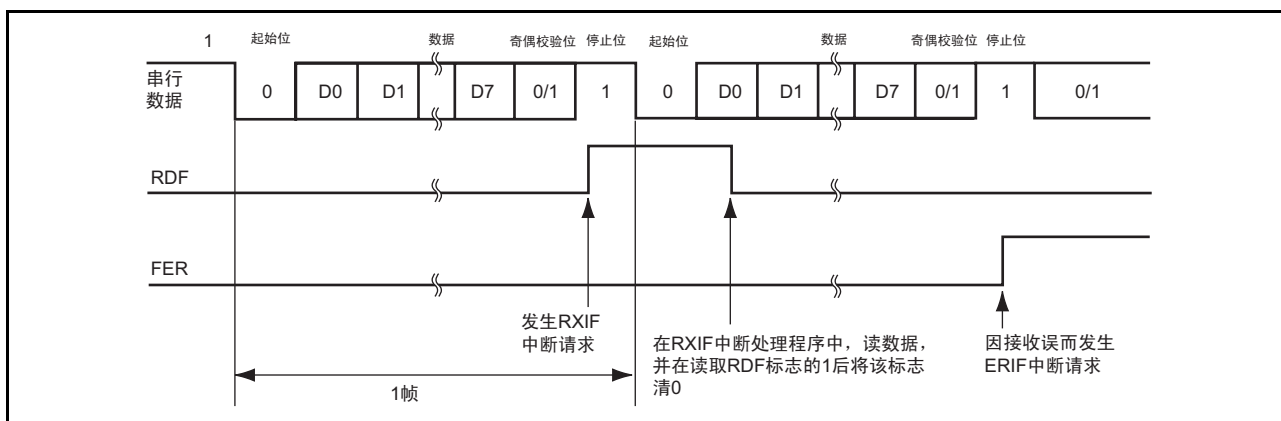


图 16.9 SCIF 接收时的运行例 (8 位数据 / 有奇偶校验 / 1 个停止位)

5. 当调制解调器控制有效时, 就在 SCFRDR 为空时输出  $\overline{\text{RTS}}$  信号。 $\overline{\text{RTS}}$  为 0 时为可接收状态,  $\overline{\text{RTS}}$  为 1 时表示 SCFRDR 的数据已满, 不能再接收。

使用调制解调器控制时的运行例子如图 16.10 所示。

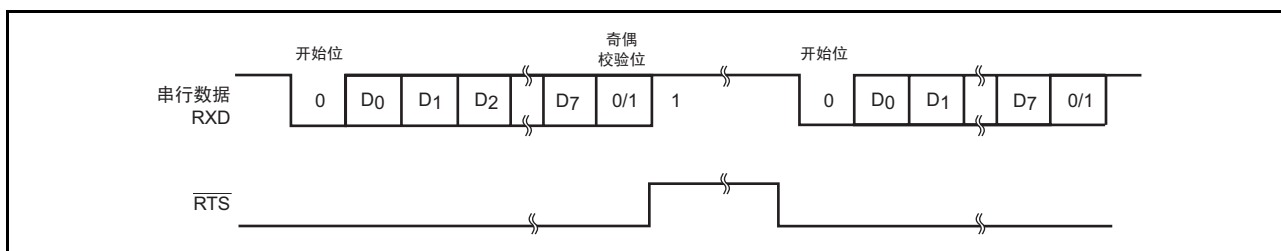


图 16.10 使用调制解调器控制时的运行例 ( $\overline{\text{RTS}}$ )

### 16.4.3 时钟同步模式中的运行

时钟同步模式是与时钟脉冲同步进行数据发送和接收的模式，适用于高速串行通信。

在 SCIF 内部发送部和接收部独立，所以能通过共享时钟进行全双工通信，而且发送部和接收部都是 16 段 FIFO 缓冲结构，所以能在发送和接收中读写数据，进行连续的发送和接收。

时钟同步串行通信的一般格式如图 16.11 所示。

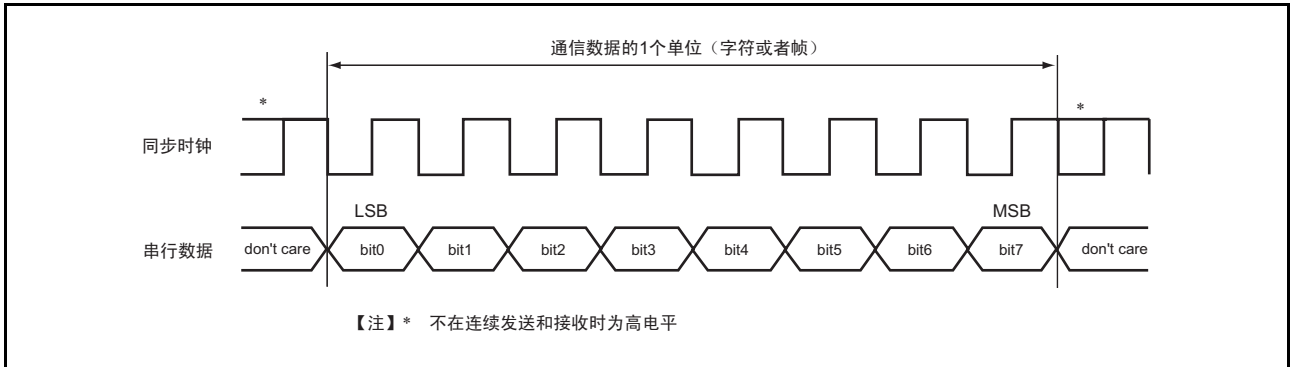


图 16.11 时钟同步通信的数据格式

在时钟同步串行通信中，在同步时钟的下降沿和下一个下降沿之间将数据输出到通信线路，保证在同步时钟的上升沿数据有效。

以数据的 LSB（最初）到 MSB（最后）的顺序输出串行通信的 1 个字符。输出 MSB 后的通信线路状态保持 MSB 的状态。

在时钟同步模式中，SCIF 与同步时钟的上升沿同步接收数据。

#### (1) 发送 / 接收格式

固定为 8 位数据。

不能附加奇偶校验位。

#### (2) 时钟

能通过设定 SCSMR 的  $\overline{C/A}$  位和 SCSCR 的 CKE1 位、CKE0 位，选择内部波特率发生器生成的内部时钟或者 SCK 引脚输入的外部同步时钟。

在通过内部时钟运行时，SCK 引脚输出同步时钟。同步时钟在发送和接收 1 个字符时输出 8 个脉冲，而在不进行发送和接收时固定为高电平。如果只在接收时选择内部时钟，就在 SCSCR 的 RE 位是 1 的期间并且在接收 FIFO 内的数据个数达到接收触发设定个数前，输出时钟脉冲。此时，输出  $8 \times (16+1) = 136$  个脉冲的同步时钟。如果进行 n 个字符数的接收时，必须将时外部时钟作为时钟预源。在使用内部时钟时，必须按照先设定 RE=1 且 TE=1，再在发送 n 个字符数的虚拟数据的同时接收 n 个字符数的顺序。

#### (3) 数据的发送和接收运行

##### • SCIF 的初始化（时钟同步模式）

在发送和接收数据前，必须在将串行控制寄存器（SCSCR）的 TE 和 RE 位清 0 后，按照以下步骤初始化 SCIF。

在更改模式和通信格式等时，必须在将 TE 和 RE 位清 0 后，按照以下步骤进行更改。如果将 TE 位清 0，就初始化发送移位寄存器（SCTSR）。必须注意：即使将 RE 位清 0，也保持 RDF、PER、FER、ORER 各标志和接收数据寄存器（SCRDR）的内容。

SCIF 的初始化流程图例子如图 16.12 所示。

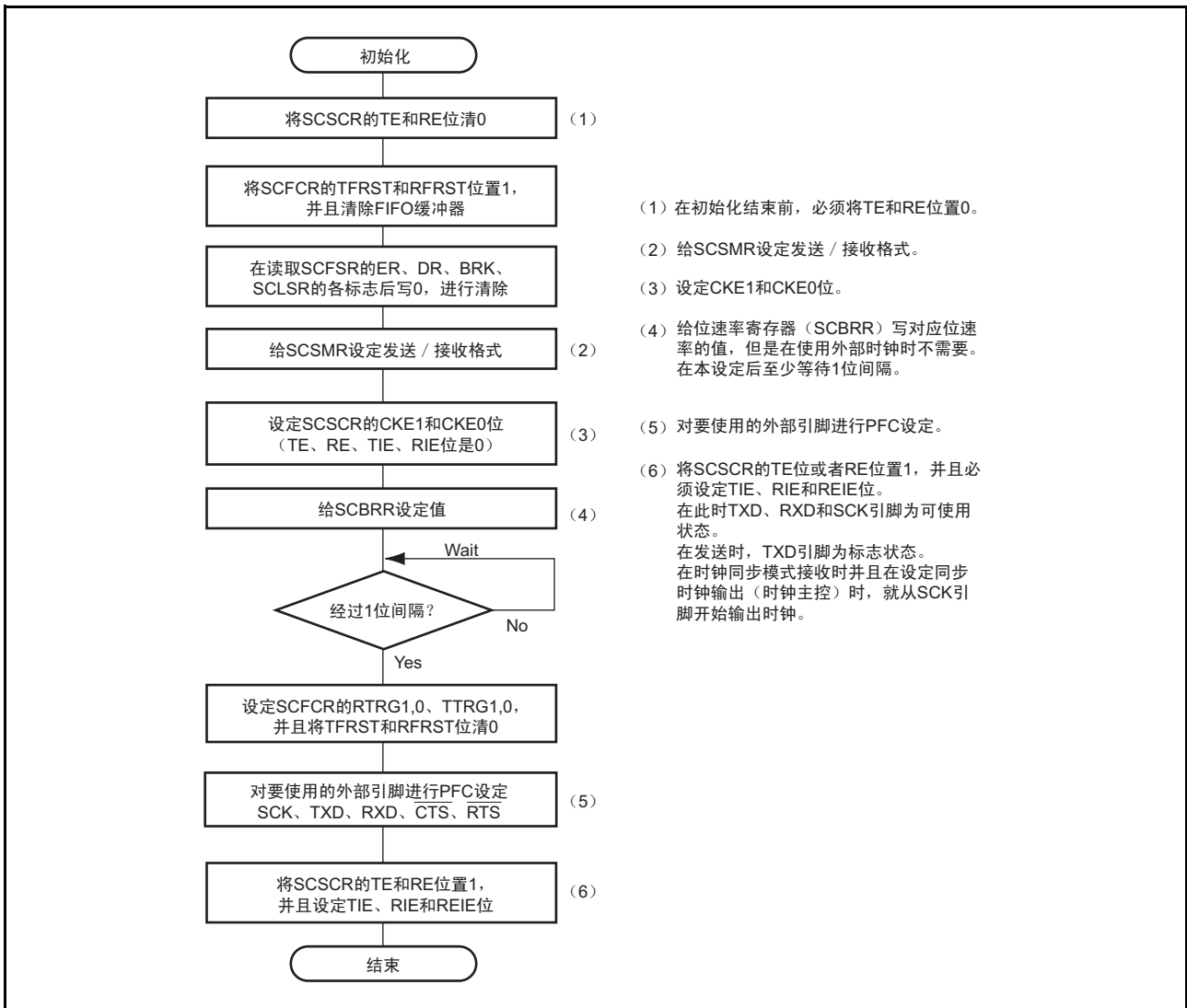


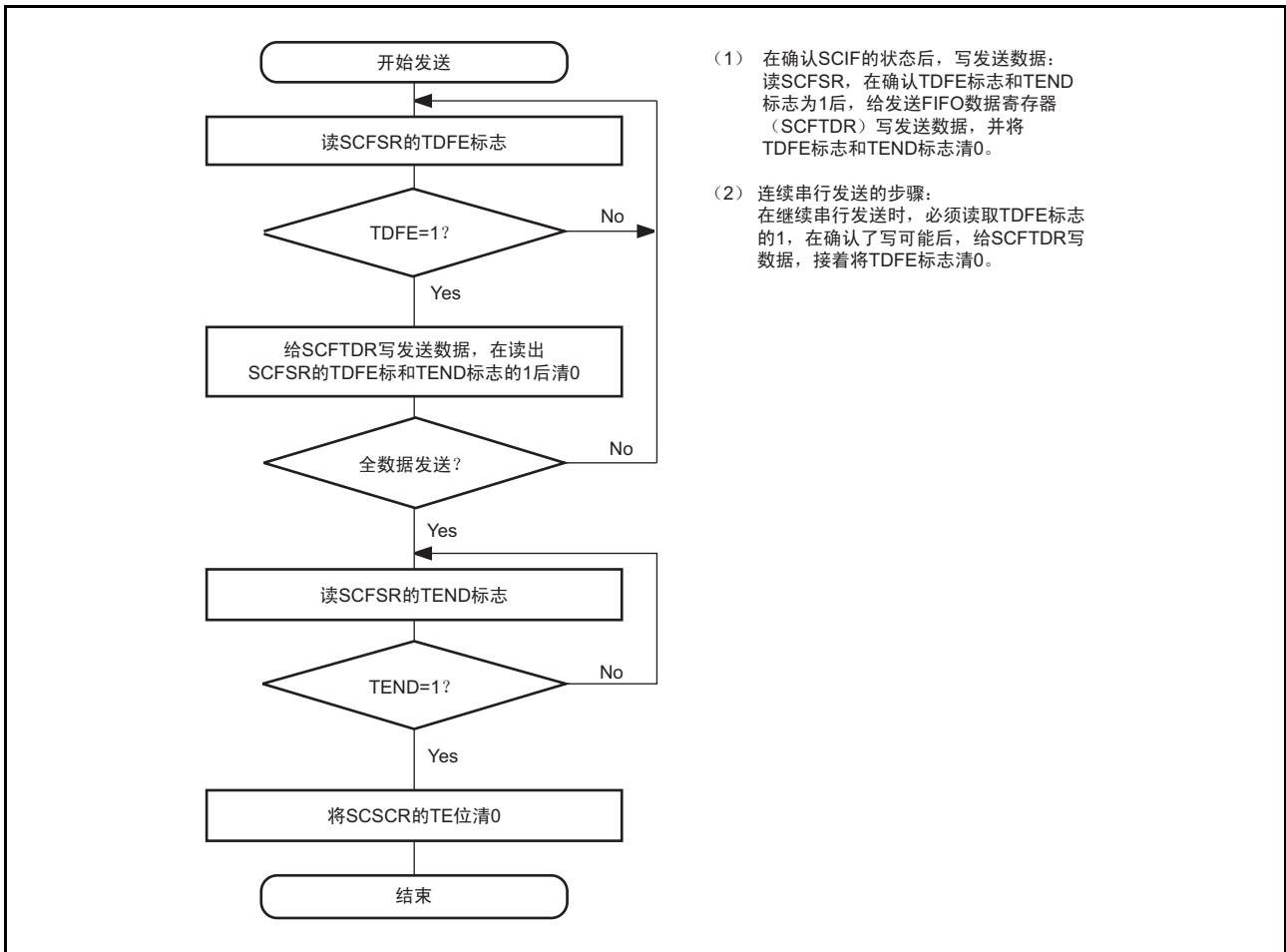
图 16.12 SCIF 的初始化流程图例子



• 串行数据的发送 (时钟同步模式)

串行发送的流程图例子如图 16.13 所示。

必须按照步骤, 在将 SCIF 设定为可发送的运行状态后发送串行数据。



(1) 在确认SCIF的状态后, 写发送数据: 读SCFSR, 在确认TDFE标志和TEND标志为1后, 给发送FIFO数据寄存器 (SCFTDR) 写发送数据, 并将TDFE标志和TEND标志清0。

(2) 连续串行发送的步骤: 在继续串行发送时, 必须读取TDFE标志的1, 在确认了写可能后, 给SCFTDR写数据, 接着将TDFE标志清0。

图 16.13 串行发送的流程图例子

串行发送时的 SCIF 运行如下：

1. 如果将数据写到发送 FIFO 数据寄存器 (SCFTDR)，SCIF 就将数据从 SCFTDR 传送到发送移位寄存器 (SCTSR)。在将发送数据写到 SCFTDR 前，必须确认串行状态寄存器 (SCFSR) 的 TDFE 标志是否为 1。能写的发送数据字节数为 (16- 发送触发设定数)。
2. 如果将数据从 SCFTDR 传送到 SCTSR 后开始发送，就进行连续发送，直到 SCFTDR 中没有发送数据为止。当 SCFTDR 的发送数据字节数小于 FIFO 控制寄存器 (SCFCR) 设定的发送触发个数时，TDFE 标志被置位。此时，如果串行控制寄存器 (SCSCR) 的 TIE 位已被置 1，就产生发送 FIFO 数据空的中断 (TXIF) 请求。

当设定为时钟输出模式时，SCIF 输出同步时钟的 8 个脉冲。当设定为外部时钟时，与输入时钟同步输出数据。按照 LSB 到 MSB 的顺序从 TXD 引脚送出串行发送数据。

3. SCIF 在送出最后位时检查 SCFTDR 的发送数据。如果有发送数据，就将数据从 SCFTDR 传送到 SCTSR，开始下一帧的串行发送；如果没有发送数据，就将串行状态寄存器 (SCFSR) 的 TEND 标志设定为 1，在送出最后位后，TXD 引脚保持状态。
4. 在结束串行发送后，SCK 引脚固定为高电平。

SCIF 发送时的运行例子如图 16.14 所示。

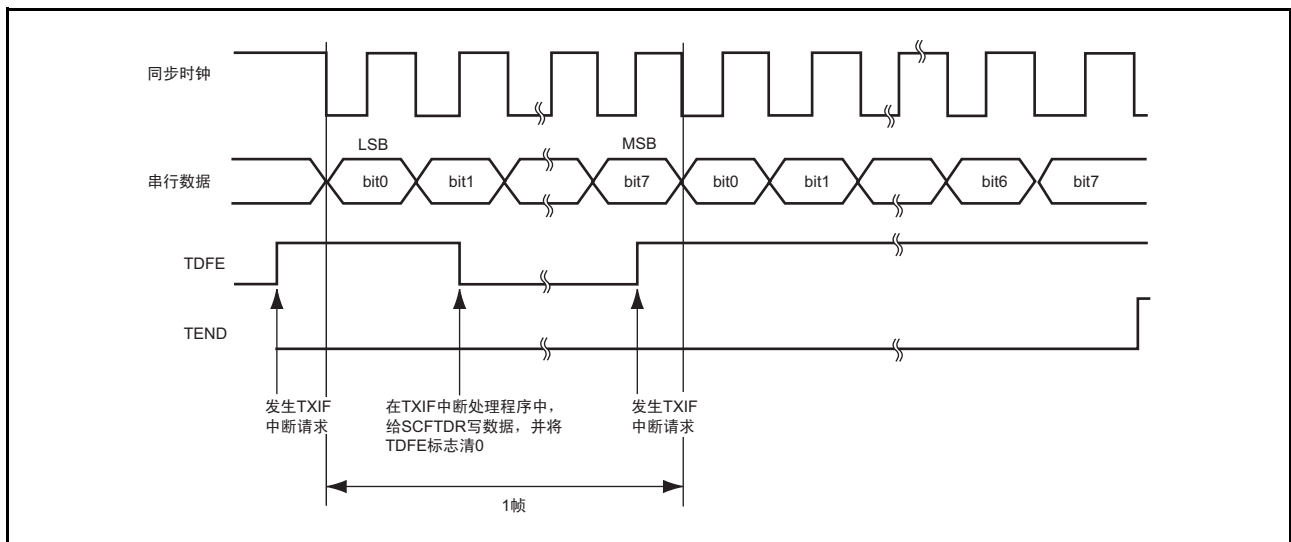


图 16.14 SCIF 发送时的运行例

• 串行数据的接收 (时钟同步模式)

串行接收的流程图例子如图 16.15 和图 16.16 所示。

在将 SCIF 设定为可接收状态后, 必须按照以下步骤接收串行数据。

在不初始化 SCIF 并将运行从异步模式转换为时钟同步模式时, 必须确认 ORER、PER、FER 各标志是否已被清 0。

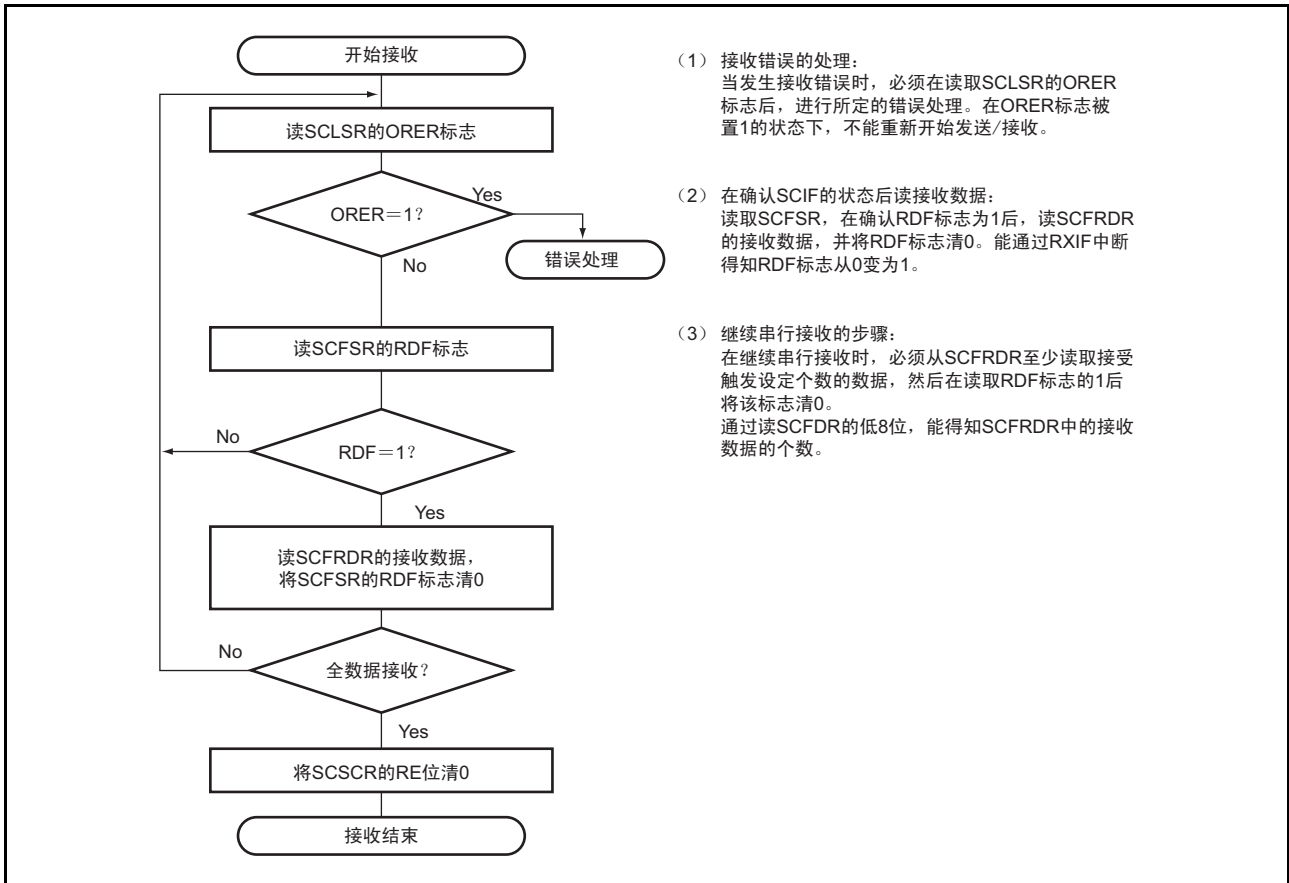


图 16.15 串行接收的流程图例子 (1)

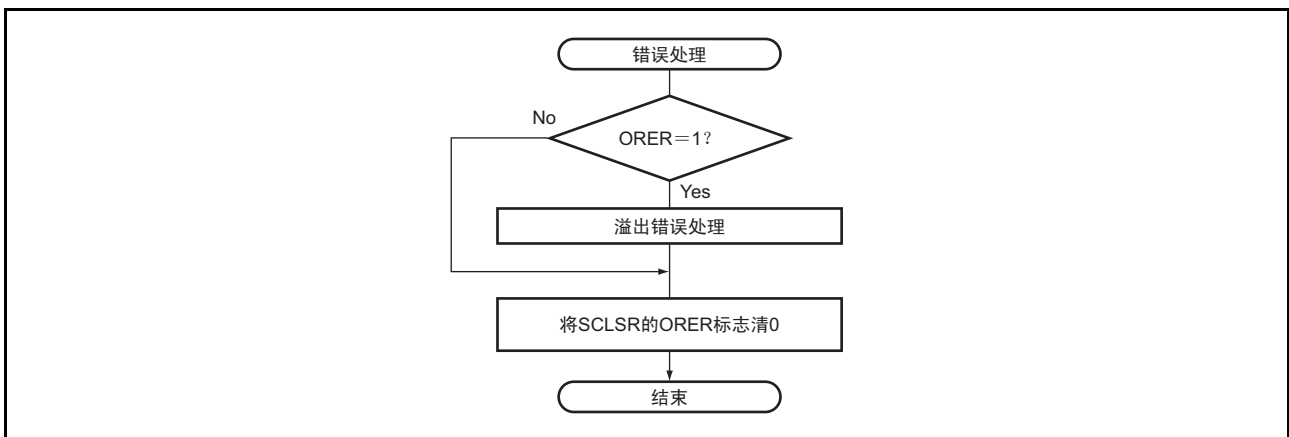


图 16.16 串行接收的流程图例子 (2)

串行接收时的 SCIF 运行如下：

1. SCIF 与同步时钟的输入或输出同步，开始接收。
2. 按照接收移位寄存器 (SCRSR) 的 LSB 到 MSB 的顺序保存接收数据。  
在接收后，SCIF 检查是否为能将接收数据从 SCRSR 传送到 SCFRDR 的状态。在此检查通过后，将接收数据保存到 SCFRDR。  
如果在错误检查时检测到溢出错误，就不能进行以后的接收。
3. 在 RDF 标志是 1 时，如果串行控制寄存器 (SCSCR) 的 RIE 位已被置 1，就产生接收数据满的中断 (RXIF) 请求；在 ORER 标志是 1 时，如果 SCSCR 的 RIE 位或者 REIE 位已被置 1，就产生中止的中断 (BRIF) 请求。

SCIF 接收时的运行例子如图 16.17。

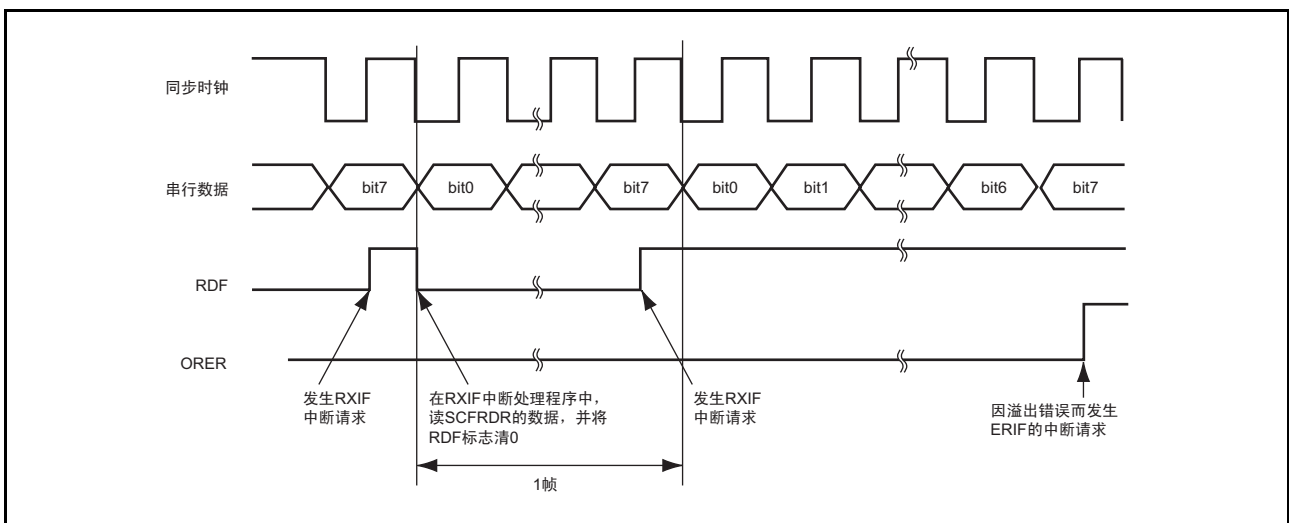


图 16.17 SCIF 接收时的运行例

• 串行数据的同时发送和接收 (时钟同步模式)

串行数据的同时发送和接收的流程图例子如图 16.18 所示。

在将 SCIF 设定为可发送和可接收的运行状态后, 必须按照以下步骤同时发送和接收串行数据。

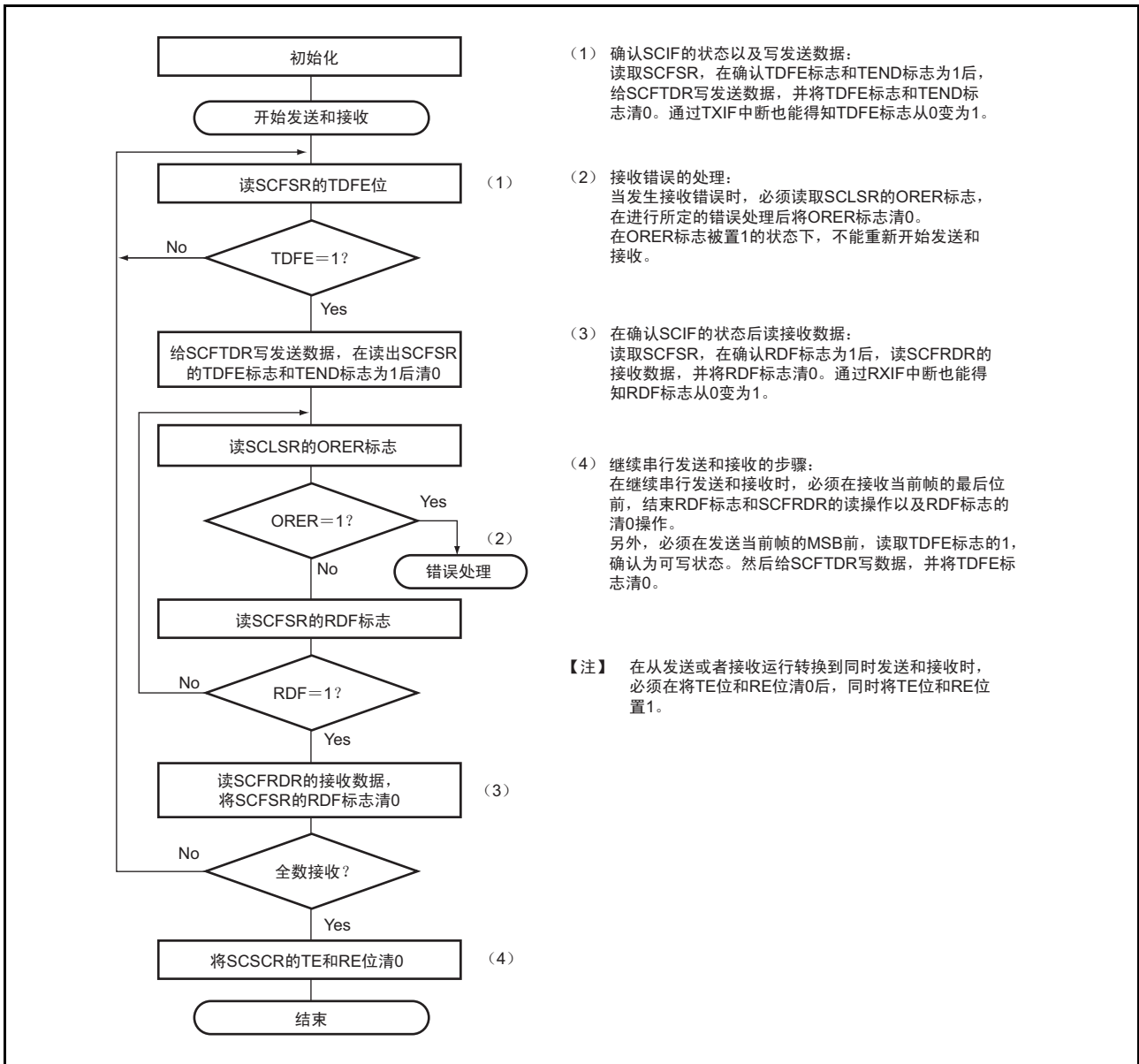


图 16.18 串行发送和接收的流程图例子

## 16.5 SCIF 的中断源和 DTC

SCIF 有发送 FIFO 数据空的中断 (TXIF) 请求、接收错误的中断 (ERIF) 请求、接收数据满的中断 (RXIF) 请求和中止的中断 (BRIF) 请求共 4 种。

各中断源和优先级如表 16.16 所示。能通过 SCSCR 的 TIE 位、RIE 位、REIE 位允许或者禁止各中断源。另外，分别将各中断请求独立传送到中断控制器。

在 TIE 位允许 TXIF 中断时，如果将串行状态寄存器 (SCFSR) 的 TDFE 标志设定为 1，就产生 TXIF 中断。

在 RIE 位允许 RXIF 中断时，如果将 SCFSR 的 RDF 标志或者 DR 标志设定为 1，就产生 RXIF 中断。但是只在异步模式时，才能通过将 DR 标志置 1 产生 RXIF 中断。

在 RIE 位或者 REIE 位允许 BRIF 中断时，如果将 SCFSR 的 BRK 标志或者 SCLSR 的 ORER 标志设定为 1，就产生 BRIF 中断。

在 RIE 位或者 REIE 位允许 ERIF 中断时，如果将 SCFCR 的 ER 标志设定为 1，就产生 ERIF 中断。

如果将 SCSCR 的 RIE 位设定为 0，并且将 REIE 位设定为 1，就不能产生 RXIF 中断而只能产生 ERIF 和 BRIF 中断。

另外，TXIF 中断表示能写发送数据，RXIF 中断表示 SCFRDR 中有接收数据。

表 16.16 SCIF 中断源

中断源	内 容	中断允许位	启动 DTC
ERIF	接收错误 (ER) 的中断	RIE 或者 REIE	×
RXIF	接收 FIFO 数据满 (RDF) 或者数据就绪 (DR) 的中断	RIE	○
BRIF	中止 (BRK) 或者溢出错误 (ORER) 的中断	RIE 或者 REIE	×
TXIF	发送 FIFO 数据空 (TDFE) 中断	TIE	○

### 16.6 串行端口寄存器 (SCSPTR) 和 SCIF 引脚的关系

SCSPTR 和 SCIF 引脚的关系如图 16.19 ~ 图 16.22 所示。

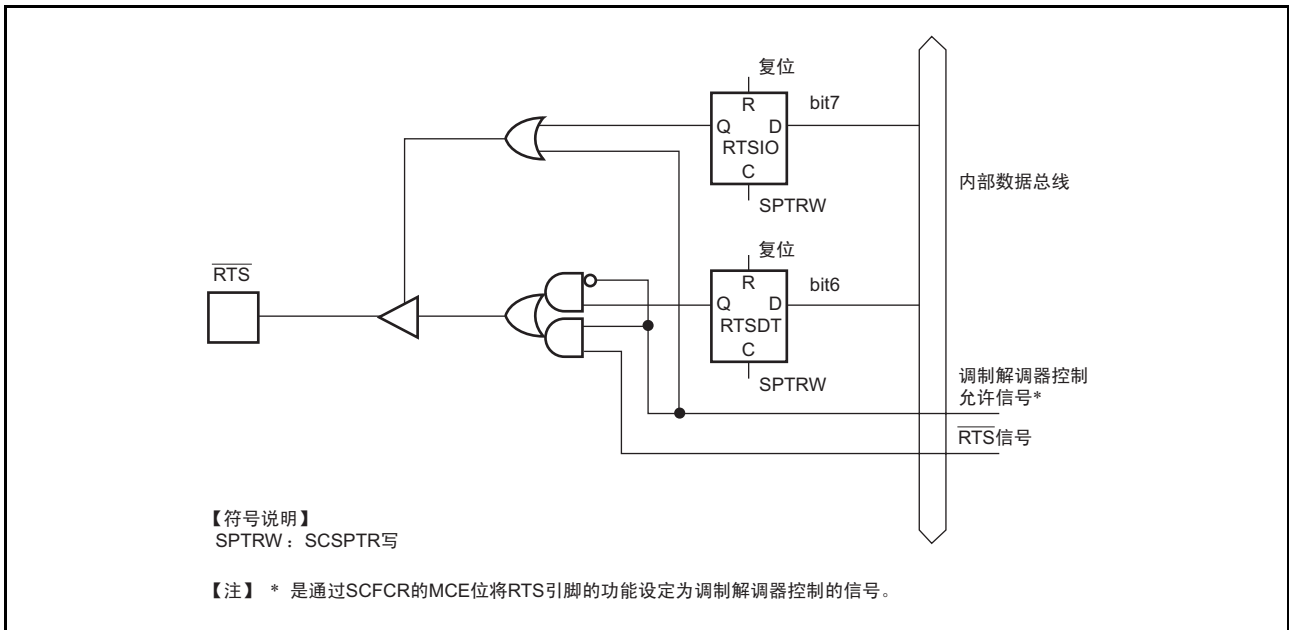


图 16.19 RTSIO 位、RTSDT 位和  $\overline{\text{RTS}}$  引脚的关系

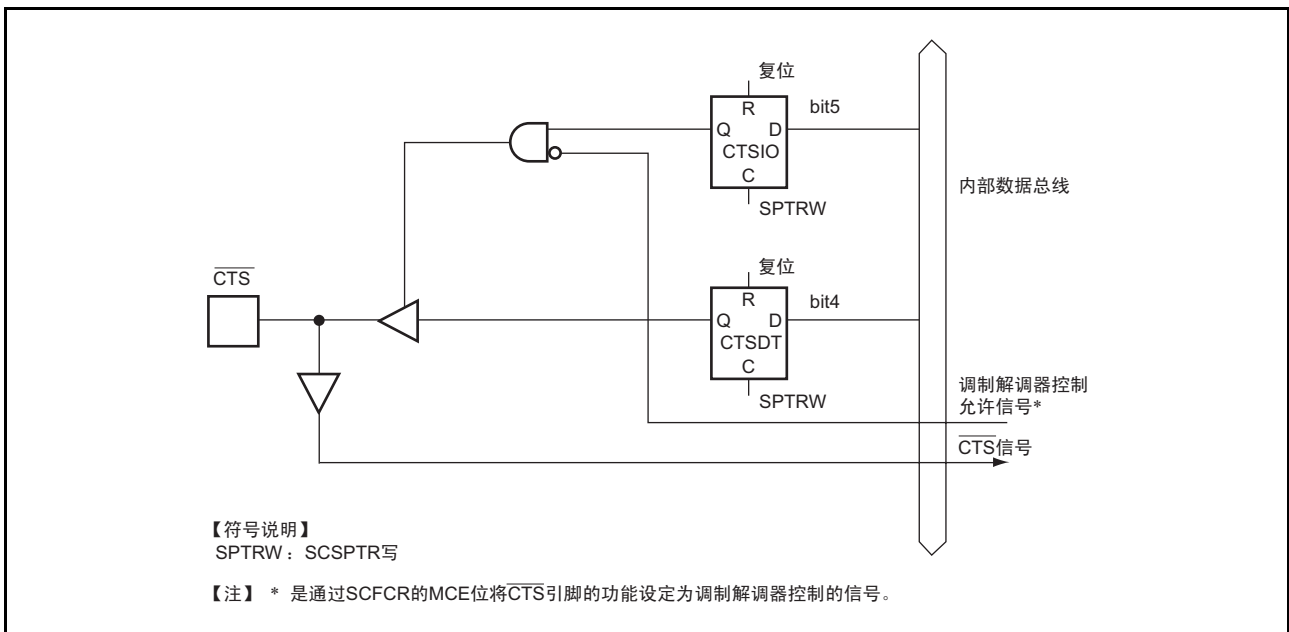


图 16.20 CTSIO 位、CTSDT 位和  $\overline{\text{CTS}}$  引脚的关系

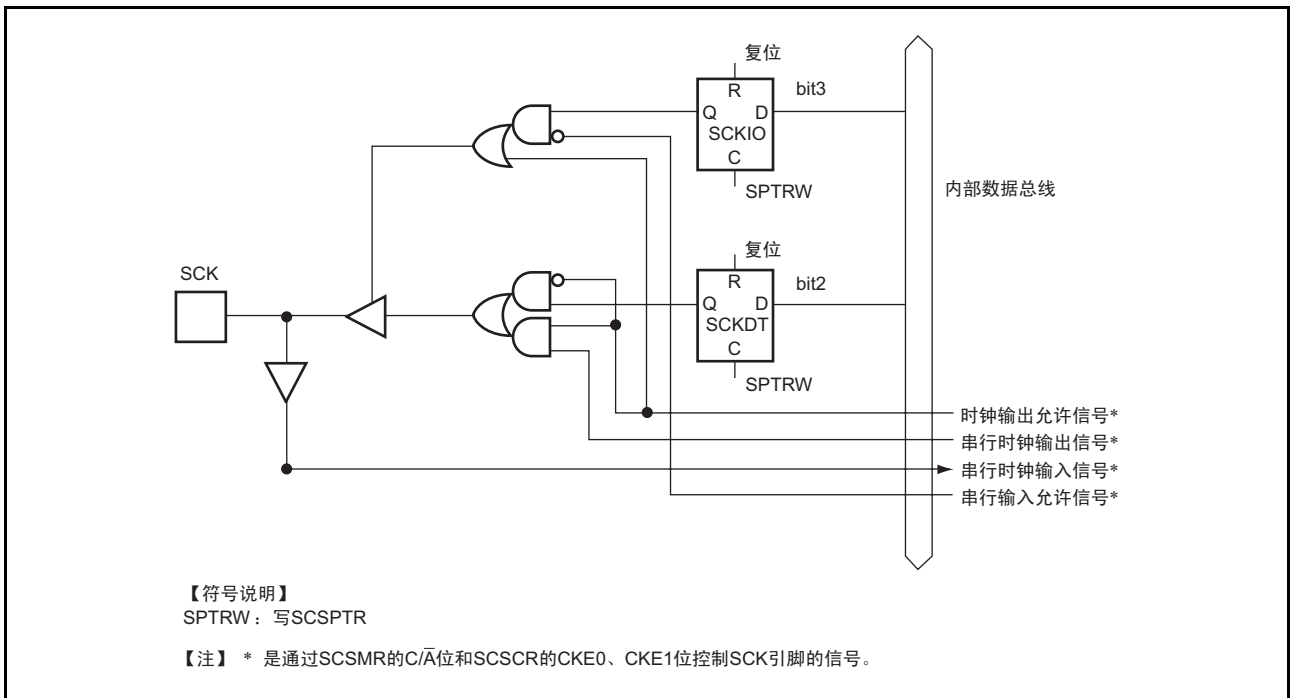


图 16.21 SCKIO 位、SCKDT 位和 SCK 引脚的关系

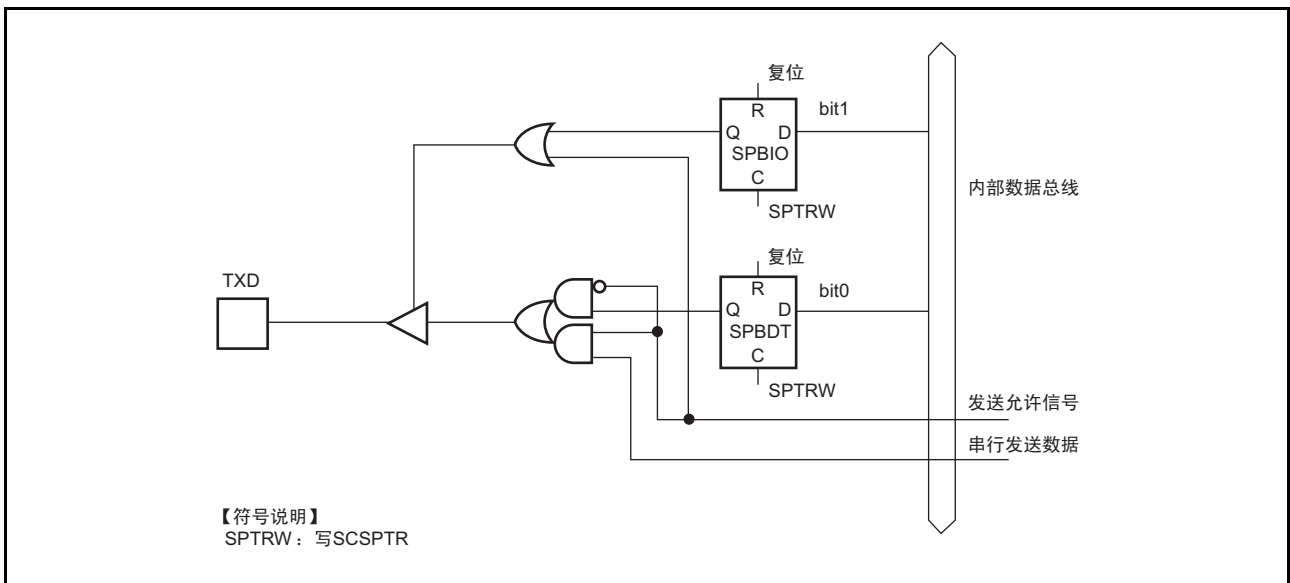


图 16.22 SPBIO 位、SPBDT 位和 TXD 引脚的关系



## 16.7 使用注意事项

在使用 SCIF 时，必须注意以下几点：

### 16.7.1 有关 SCFTDR 的写操作和 TDFE 标志

如果写到发送 FIFO 数据寄存器 (SCFTDR) 的发送数据字节数小于 FIFO 控制寄存器 (SCFCR) 的 TTRG1 位和 TTRG0 位设定的发送触发个数时，就将串行状态寄存器 (SCFSR) 的 TDFE 标志置位。在 TDFE 标志被置位后，能在 SCFTDR 变空之前写发送数据，进行高效率的连续发送。

如果写到 SCFTDR 的数据字节数小于发送触发个数，即使在读 1 后将 TDFE 标志清 0，TDFE 也将被再次置 1。如果写 SCFTDR 的数据字节数超过发送触发个数，必须在读 1 后将 TDFE 清 0。

能通过 FIFO 数据个数寄存器 (SCFDR) 的高 8 位得知 SCFTDR 的发送数据字节数。

### 16.7.2 有关 SCFRDR 的读操作和 RDF 标志

如果接收 FIFO 数据寄存器 (SCFRDR) 的接收数据字节数大于 FIFO 控制寄存器 (SCFCR) 的 RTRG1 位和 RTRG0 位设定的接收触发个数，就将串行状态寄存器 (SCFSR) 的 RDF 标志置位。在 RDF 被置位后，能从 SCFRDR 读取触发个数的接收数据，进行高效率的连续接收。

如果 SCFRDR 的数据字节数大于接收触发个数，即使在读 1 后将 RDF 标志清 0，RDF 标志也将被再次置 1。因此，为了使接收 FIFO 数据寄存器 (SCFRDR) 内的数据个数小于触发个数，必须在读接收数据后，读 RDF 标志的 1，然后清除 RDF。

能从 FIFO 数据个数寄存器 (SCFDR) 高 8 位得知 SCFRDR 的接收数据字节数。

### 16.7.3 有关中止的检测和处理

在检测帧错误 (FER) 时，能通过直接读 RXD 引脚的值检测中止。在中止时，RXD 引脚的输入始终为 0，所以 FER 标志被置位，并且奇偶校验错误标志也可能 (PER) 被置位。

如果检测到中止，就停止向 SCFRDR 传送接收数据，但是 SCIF 继续接收。

### 16.7.4 中止的发送

能通过串行端口寄存器 (SCSPTR) 的 SPBIO 位和 SPBDT 位决定 TXD 引脚的输入 / 输出的条件和电平，所以能让这一功能发送中止。

从串行发送的初始化到 TE 位被置 1 (能发送)，TXD 引脚不工作。在此期间，标记状态被 SPBDT 位的值替代。因此，先要将 SPBIO 和 SPBDT 位设定为 1 (高电平输出)。

当要在串行发送时发送中止，就必须在将 SPBDT 位清 0 (指定低电平) 后，再将 TE 位清 0 (停止发送)。如果将 TE 位清 0，就和当前的发送状态无关，初始化发送部并从 TXD 引脚输出 0。

### 16.7.5 异步模式的接收数据采样时序和接收容限

SCIF 通过传送速率的 16 倍频率的基本时钟运行。在接收时，SCIF 通过基本时钟采样起始位的下降沿，取得内部的同步，而在基本时钟的第 8 个时钟脉冲的上升沿将接收数据取到内部。此时序如图 16.23 所示。

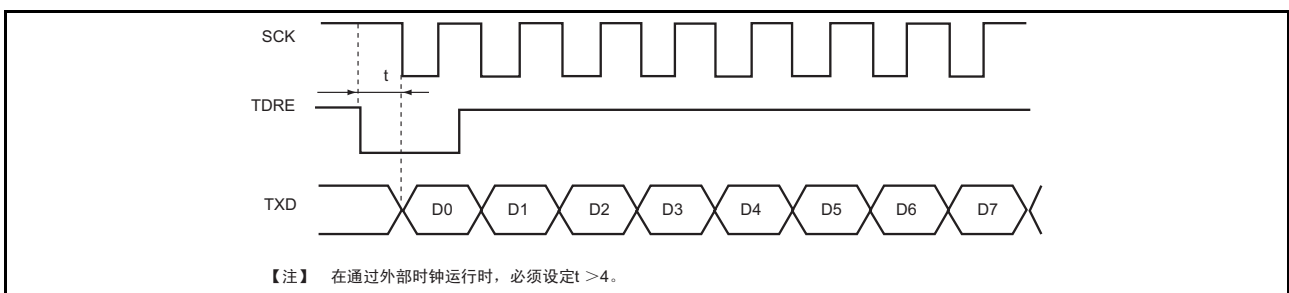


图 16.23 接收数据的采样时序

因此，能用表达式 (1) 表示异步模式的接收容限。

$$M = \left\{ \left( 0.5 - \frac{1}{2N} \right) - (L-0.5)F - \frac{|D-0.5|}{N} (1+F) \right\} \times 100[\%] \quad \cdots \text{表达式 (1)}$$

- M : 接收容限 (%)  
 N : 时钟的位速率比 (N=16)  
 D : 时钟占空比 (D=0 ~ 1.0)  
 L : 帧长 (L=9 ~ 12)  
 F : 时钟频率的偏差绝对值

如果表达式 (1) 中 F=0 并且 D=0.5，根据表达式 (2)，接收容限为 46.875%。

当 D=0.5 并且 F=0 时

$$\begin{aligned} M &= (0.5 - 1/(2 \times 16)) \times 100 \% \\ &= 46.875 \% \end{aligned} \quad \cdots \text{表达式 (2)}$$

此值为计算值，所以在系统设计时必须留有 20 ~ 30% 的余地。

### 16.7.6 模块待机模式的设定

SCIF 能通过待机控制寄存器，禁止或者允许本模块的运行。初始值为停止 SCIF 的运行。通过解除模块待机模式，允许寄存器的存取。详细内容请参照“第 26 章 低功耗模式”。

### 16.7.7 DTC 使用注意事项

在通过 TXIF 中断启动 DTC 后将数据写到 SCFTDR 时，TEND 标志不定，所以不能将 TEND 标志用作发送结束标志。

### 16.7.8 关于串行状态寄存器 (SCFSR) 的 FER 标志核 PER 标志

串行状态寄存器 (SCFSR) 的 FER 标志和 PER 标志是下一个读出的接收 FIFO 数据寄存器 (SCFROD) 的状态标志。通过 CPU 或 DTC 读出接收 FIFO 数据寄存器时，接收数据的帧错误及奇偶校验错误就会消失。

确认接收数据的帧错误及奇偶校验错误的状态时，必须在读出串行状态寄存器后，再读接收 FIFO 寄存器。

## 第 17 章 同步串行通信单元 (SSU)

本 LSI 具有 1 个通道的同步串行通信单元 (SSU: Synchronous Serial communication Unit)。SSU 有主模式 (以本 LSI 为主器件, 将时钟输出到外部进行同步串行通信) 和从属模式 (从外部器件输入时钟进行同步串行通信), 能在不同时钟极性和不同时钟相位的器件之间进行同步串行通信。

### 17.1 特点

- 可选择 SSU 模式和时钟同步通信模式
- 可选择主模式和从属模式
- 可选择标准模式和双向模式
- 能与不同时钟相位和不同时钟极性的其他器件进行同步串行通信
- 发送 / 接收数据长度可选择 8 位、16 位或者 32 位
- 能进行全双工通信  
装载了能同时进行发送和接收的移位寄存器
- 能进行连续串行通信
- 可选择 LSB 先方式或者 MSB 先方式
- 可选择 7 种内部时钟 (P $\phi$ /4、P $\phi$ /8、P $\phi$ /16、P $\phi$ /32、P $\phi$ /64、P $\phi$ /128、P $\phi$ /256) 和外部时钟作为时钟源
- 中断源: 5 种  
发送结束、发送数据空、接收数据满、溢出错误和冲突错误共 5 种中断源  
能通过发送数据空请求或者接收数据满请求, 启动数据传送控制器 (DTC) 进行数据传送。
- 能设定模块待机模式

SSU 的框图如图 17.1 所示。

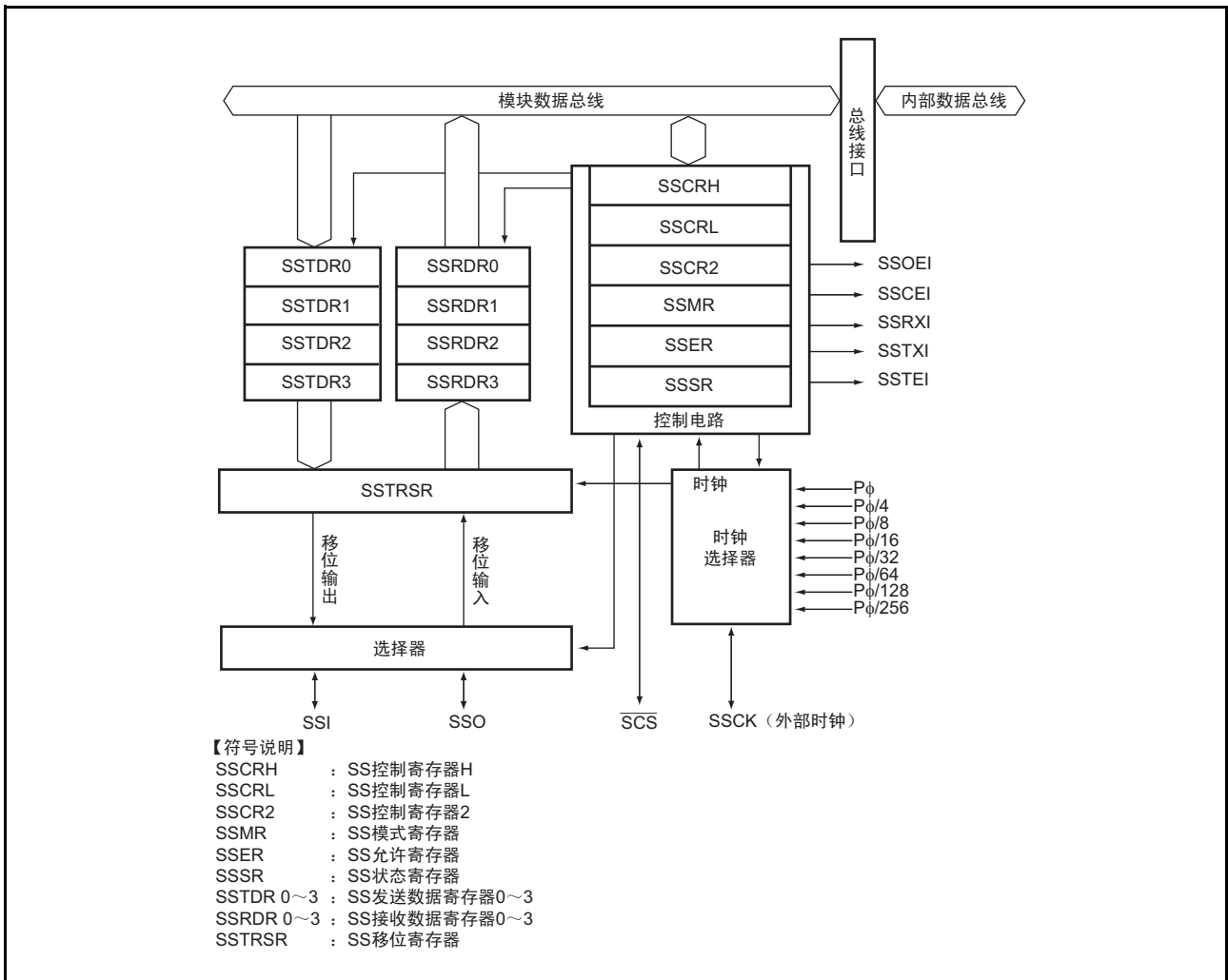


图 17.1 SSU 的框图

## 17.2 输入 / 输出引脚

SSU 有表 17.1 中的输入 / 输出引脚。

表 17.1 引脚结构

引脚名	输入 / 输出	功 能
SSCK	输入 / 输出	SSU 的时钟输入 / 输出引脚
SSI	输入 / 输出	SSU 的数据输入 / 输出引脚
SSO	输入 / 输出	SSU 的数据输入 / 输出引脚
$\overline{\text{SCS}}$	输入 / 输出	SSU 的片选输入 / 输出引脚

## 17.3 寄存器说明

SSU 有以下寄存器，有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 27 章 寄存器一览”。

表 17.2 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
SS 控制寄存器 H	SSCRH	R/W	H'0D	H'FFFCD00	8、16
SS 控制寄存器 L	SSCRL	R/W	H'00	H'FFFCD01	8
SS 模式寄存器	SSMR	R/W	H'00	H'FFFCD02	8、16
SS 允许寄存器	SSER	R/W	H'00	H'FFFCD03	8
SS 状态寄存器	SSSR	R/W	H'04	H'FFFCD04	8、16
SS 控制寄存器 2	SSCR2	R/W	H'00	H'FFFCD05	8
SS 发送数据寄存器 0	SSTDR0	R/W	H'00	H'FFFCD06	8、16
SS 发送数据寄存器 1	SSTDR1	R/W	H'00	H'FFFCD07	8
SS 发送数据寄存器 2	SSTDR2	R/W	H'00	H'FFFCD08	8、16
SS 发送数据寄存器 3	SSTDR3	R/W	H'00	H'FFFCD09	8
SS 接收数据寄存器 0	SSRDR0	R	H'00	H'FFFCD0A	8、16
SS 接收数据寄存器 1	SSRDR1	R	H'00	H'FFFCD0B	8
SS 接收数据寄存器 2	SSRDR2	R	H'00	H'FFFCD0C	8、16
SS 接收数据寄存器 3	SSRDR3	R	H'00	H'FFFCD0D	8

## 17.3.1 SS 控制寄存器 H (SSCRH)

SSCRH 进行主器件 / 从属器件的选择、双向模式的允许、SSO 引脚输出值的选择、SSCK 引脚的选择以及 SCS 引脚的选择。

位:	7	6	5	4	3	2	1	0
	MSS	BIDE	-	SOL	SOLP	-	CSS[1:0]	
初始值:	0	0	0	0	1	1	0	1
R/W:	R/W	R/W	R	R/W	R/W	R	R/W	R/W

位	位名	初始值	R/W	说 明
7	MSS	0	R/W	主器件 / 从属器件的选择 选择是将 SSU 用作主模式还是用作从属模式。当用作主模式时，就从 SSCK 引脚输出传送时钟。如果 SSSR 的 CE 位被置位，此位就自动被清除。 0: 选择从属模式 1: 选择主模式
6	BIDE	0	R/W	双向模式的允许 选择是使用串行数据的输入引脚和输出引脚的 2 个引脚还是只使用 1 个引脚。如果选择双向模式，就不能同时进行发送和接收。详细内容请参照“17.4.3 数据输入 / 输出引脚和移位寄存器的关系”。 0: 标准模式（使用数据输入引脚和数据输出引脚的 2 个引脚进行通信） 1: 双向模式（通信时，数据输入和数据输出只使用 1 个引脚）
5	—	0	R	保留位 读写的值总是 0。
4	SOL	0	R/W	串行数据输出值的选择 在发送结束后，串行数据的输出保持发送数据的最后位的值的电平，但是能在发送前或者发送后更改串行数据的输出电平。当要更改输出电平时，必须用 MOV 指令将 SOLP 位设定为 0。另外，如果在数据传送中写此位，就会产生误动作，所以不能在发送中更改此位。 0: 将串行数据的输出电平更改为 Low 电平 1: 将串行数据的输出电平更改为 High 电平
3	SOLP	1	R/W	SOL 位的写保护 当要更改串行数据的输出电平时，必须用 MOV 指令设定 SOL=1 并且 SOLP=0、或者 SOL=0 并且 SOLP=0。 0: 能通过 SOL 的值更改输出电平 1: 不能通过 SOL 的值更改输出电平 读的值总是 1。
2	—	1	R	保留位 读写的值总是 1。

位	位名	初始值	R/W	说 明
1、0	CSS[1:0]	01	R/W	SCS 引脚的选择 选择是将 SCS 引脚用作 SCS 输入还是用作 SCS 输出。 00: 禁止设定 01: 用作 SCS 输入 10: 用作 SCS 自动输入 / 输出 (在传送前后为 SCS 输入, 在传送中为 Low 电平输出) 11: 用作 SCS 自动输出 (在传送前后为 High 电平输出, 在传送中为 Low 电平输出)

### 17.3.2 SS 控制寄存器 L (SSCRL)

SSCRL 选择运行模式、软件复位以及发送 / 接收数据的长度。

位:	7	6	5	4	3	2	1	0
	FCLRM	SSUMS	SRES	-	-	-	DATS[1:0]	
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
7	FCLRM	0	R/W	标志清除模式 选择是在写 SSTDR 或者读 SSRDR 时还是在 DTC 传送结束时清除 SSRXI 和 SSTXI 的中断标志。在使用 DTC 时, 必须将此位设定为 0。 0: 在 DTC 传送结束时 1: 在存取寄存器时
6	SSUMS	0	R/W	SSU 模式和时钟同步通信模式的选择。 0: SSU 模式 1: 时钟同步通信模式
5	SRES	0	R/W	软件复位 如果将此位设定为 1, 就对 SSU 内部定序器进行强制复位, 然后自动清除此位, 并且对 SSSR 的 ORER 位、TEND 位、TDRE 位、RDRF 位、CE 位和 SSER 的 TE 位、RE 位进行初始化, 其他 SSU 内部寄存器的值保持不变。 要在中途中断传送时, 必须给此位写 1, 对内部定序器进行复位。
4 ~ 2	—	全 0	R	保留位 读写的值总是 0。
1、0	DATS[1:0]	00	R/W	发送 / 接收数据长度的选择 选择串行数据的长度。 00: 8 位 01: 16 位 10: 32 位 11: 设定无效

## 17.3.3 SS 模式寄存器 (SSMR)

SSMR 选择 MSB 先 /LSB 先、时钟极性、时钟相位和传送时钟速率。

位:	7	6	5	4	3	2	1	0
	MLS	CPOS	CPHS	-	-	CKS[2:0]		
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7	MLS	0	R/W	MSB 先 /LSB 先的选择 选择是以 MSB 先还是以 LSB 先进行串行数据的传送。 0: LSB 先 1: MSB 先
6	CPOS	0	R/W	时钟极性的选择 选择 SSCK 时钟的极性。 0: 空闲时为 High 电平输出, 激活时为 Low 电平输出 1: 空闲时为 Low 电平输出, 激活时为 High 电平输出
5	CPHS	0	R/W	时钟相位的选择 (只在 SSU 模式中有效) 选择 SSCK 时钟的相位。 0: 在最初的边沿, 数据发生变化 1: 在最初的边沿, 数据被锁存
4、3	—	全 0	R	保留位 读写的值总是 0。
2 ~ 0	CKS[2:0]	000	R/W	传送时钟速率的选择 在选择内部时钟时, 选择传送时钟速率 (预定标器的分频比)。 000: 保留 001: P $\phi$ /4 010: P $\phi$ /8 011: P $\phi$ /16 100: P $\phi$ /32 101: P $\phi$ /64 110: P $\phi$ /128 111: P $\phi$ /256



## 17.3.4 SS 允许寄存器 (SSER)

SSER 设定发送的允许、接收的允许以及中断请求的允许。

位:	7	6	5	4	3	2	1	0
	TE	RE	-	-	TEIE	TIE	RIE	CEIE
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7	TE	0	R/W	发送的允许 当此位被置 1 时，能进行发送。
6	RE	0	R/W	接收的允许 当此位被置 1 时，能进行接收。
5、4	—	全 0	R	保留位 读写的值总是 0。
3	TEIE	0	R/W	发送结束中断的允许 当此位被置 1 时，允许 SSTEI 中断请求。
2	TIE	0	R/W	发送中断的允许 当此位被置 1 时，允许 SSTXI 中断请求。
1	RIE	0	R/W	接收中断的允许 当此位被置 1 时，允许 SSRXI 中断请求和 SSOEI 中断请求。
0	CEIE	0	R/W	冲突错误中断的允许 当此位被置 1 时，允许 SSCEI 中断请求。

## 17.3.5 SS 状态寄存器 (SSSR)

SSSR 是各种中断的状态标志寄存器。

位:	7	6	5	4	3	2	1	0
	-	ORER	-	-	TEND	TDRE	RDRF	CE
初始值:	0	0	0	0	0	1	0	0
R/W:	R	R/W	R	R	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7	—	0	R	保留位 读写的值总是 0。
6	ORER	0	R/W	溢出错误 此位表示在 RDRF=1 的状态下接收下一个数据时, 因发生溢出错误而异常结束。SSRDR 保持发生溢出错误前的 1 帧接收数据, 而丢失后来接收的数据。另外, 不能在 ORER=1 的状态下继续进行串行接收和串行发送。 [置位条件] • 在 RDRF=1 的状态下, 当下一个串行接收的 1 字节数据结束时 [清除条件] • 当读到 ORER=1 的状态后, 给 ORER 标志写 0 时
5、4	—	全 0	R	保留位 读写的值总是 0。
3	TEND	0	R/W	发送结束 [置位条件] • 当 SSCR2 的 TENDSTS 是 0 并且在 TDRE=1 的状态下发送数据的最后位时 • 当 SSCR2 的 TENDSTS 是 1 并且在 TDRE=1 的状态下发送数据的最后位后 [清除条件] • 当读到 TEND=1 的状态后, 给 TEND 标志写 0 时 • 当将数据写到 SSTDR 时
2	TDRE	1	R/W	发送数据空 显示 SSTDR 内有无数据。 [置位条件] • 当 SSER 的 TE 是 0 时 • 当将数据从 SSTDR 传送到 SSTRSR 并且能将数据写到 SSTDR 时 [清除条件] • 当读到 TDRE=1 的状态后, 给 TDRE 标志写 0 时 • 当 TE=1 并且将数据写到 SSTDR 时 • 当通过 SSTXI 中断启动 DTC 并且在 DTC 的 MRB 的 DISEL 位是 0 的状态下将传送数据写到 SSTDR 时

位	位名	初始值	R/W	说 明
1	RDRF	0	R/W	<p>接收数据寄存器满 显示 SSRDR 内有无数据。</p> <p>[置位条件]</p> <ul style="list-style-type: none"> <li>当串行接收正常结束并且将接收数据从 SSTRSR 传送到 SSRDR 时</li> </ul> <p>[清除条件]</p> <ul style="list-style-type: none"> <li>当读到 RDRF=1 的状态后, 给 RDRF 标志写 0 时</li> <li>当从 SSRDR 读取接收数据时</li> <li>当通过 SSRXI 中断启动 DTC 并且在 DTC 的 MRB 的 DISEL 位是 0 的状态下从 SSRDR 读取接收数据时</li> </ul>
0	CE	0	R/W	<p>冲突错误 / 不完全错误 此位表示在 SSUMS=0 (SSU 模式) 并且 MSS=1 (主器件) 的状态下, 因从外部给 <math>\overline{SCS}</math> 输入 0 而发生冲突错误; 在 SSUMS=0 (SSU 模式) 并且 MSS=0 (从属器件) 的状态下, 因在 <math>\overline{SCS}</math> 引脚变为 1 时判断为主器件已停止传送而发生不完全错误。另外, 在 CE=1 的状态下, 不能继续进行串行接收和串行发送。必须在重新开始传送前, 将 SSCRL 的 SRES 置 1, 对内部定序器进行复位。</p> <p>[置位条件]</p> <ul style="list-style-type: none"> <li>当是主器件 (SSCRH 的 MSS=1) 并且给 <math>\overline{SCS}</math> 引脚输入 Low 电平时</li> <li>当是从属器件 (SSCRH 的 MSS=0) 并且在传送中 <math>\overline{SCS}</math> 引脚变为 1 时</li> </ul> <p>[清除条件]</p> <ul style="list-style-type: none"> <li>当读到 1 的状态后写 0 时</li> </ul>

## 17.3.6 SS 控制寄存器 2 (SSCR2)

SSCR2 是用来设定  $\overline{\text{SCS}}$  引脚的有效时序、SSO 引脚的数据输出时序和 TEND 位的置位时序的寄存器。

位:	7	6	5	4	3	2	1	0
	SDOS	SSCKOS	SCSOS	TENDSTS	SCSATS	SSODTS	-	-
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R

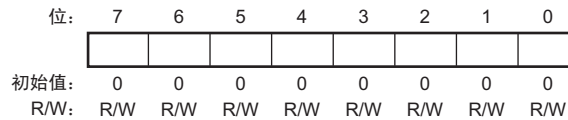
位	位名	初始值	R/W	说 明
7 ~ 5	—	全 0	R	保留位 读写的值总是 0。
4	TENDSTS	0	R/W	选择 TEND 位的置位时序 (只在设定 SSU 模式和主模式时有效) 0: 在发送最后位时, 置 TEND 位 1: 在发送最后位后, 置 TEND 位
3	SCSATS	0	R/W	选择 $\overline{\text{SCS}}$ 引脚的有效时序 (只在设定 SSU 模式和主模式时有效) 0: $t_{\text{LEAD}}$ 和 $t_{\text{LAG}}$ 的输出期间 Min. 为 $1/2 \times t_{\text{SUcyc}}$ 1: $t_{\text{LEAD}}$ 和 $t_{\text{LAG}}$ 的输出期间 Min. 为 $3/2 \times t_{\text{SUcyc}}$
2	SSODTS	0	R/W	选择 SSO 引脚的数据输出时序 (只在设定 SSU 模式和主模式时有效) 0: 在 BIDE=0、MSS=1、TE=1 或者 BIDE=1、TE=1、RE=0 时, SSO 引脚输出数据 1: 在 BIDE=0、MSS=1、TE=1 或者 BIDE=1、TE=1、RE=0 时并且在 $\overline{\text{SCS}}$ 引脚为 Low 电平期间, SSO 引脚输出数据
1、0	—	全 0	R	保留位 读写的值总是 0。

### 17.3.7 SS 发送数据寄存器 0 ~ 3 (SSTDR0 ~ SSTDR3)

SSTDR 是保存发送数据的 8 位寄存器。当通过设定 SSCRL 的 DATS1 位和 DATS0 位来选择 8 位数据长时，SSTDR0 有效；当选择 16 位数据长时，SSTDR0 和 SSTDR1 有效；当选择 32 位数据长时，SSTDR0、SSTDR1、SSTDR2 和 SSTDR3 有效。不能存取无效的 SSTDR。

如果检测到 SSTRSR 为空，SSU 就将写在 SSTDR 的发送数据传送到 SSTRSR，开始串行发送。如果在发送 SSTRSR 串行数据时，预先将下一个数据写到 SSTDR，就能进行连续串行发送。

CPU 和 DTC 随时能读写 SSTDR，但是为了可靠地进行串行通信，必须在确认 SSSR 的 TDRE 已被置 1 后才能写 SSTDR。



位	位名	初始值	R/W	说 明
7 ~ 0		全 0	R/W	串行发送数据

表 17.3 SSCRL 的 DATS 位设定和 SSTDR 的对应表

	DATS[1:0] 的设定			
	00	01	10	11 (设定无效)
SSTDR0	有效	有效	有效	无效
SSTDR1	无效	有效	有效	无效
SSTDR2	无效	无效	有效	无效
SSTDR3	无效	无效	有效	无效

### 17.3.8 SS 接收数据寄存器 0 ~ 3 (SSRDR0 ~ SSRDR3)

SSRDR 是保存接收数据的 8 位寄存器。当通过设定 SSCRL 的 DATS1 位和 DATS0 位来选择 8 位数据长时, SSRDR0 有效; 当选择 16 位数据长时, SSRDR0 和 SSRDR1 有效; 当选择 32 位数据长时, SSRDR0、SSRDR1、SSRDR2 和 SSRDR3 有效。不能存取无效的 SSRDR。

当结束 1 字节的数据接收时, SSU 将接收的串行数据从 SSTRSR 传送并保存到 SSRDR, 然后 SSTRSR 为可接收状态。因为 SSTRSR 和 SSRDR 为双缓冲结构, 所以能进行连续接收。

必须在确认 SSSR 寄存器的 RDRF 位已被置 1 后才能读 SSRDR。

SSRDR 是只读寄存器, 所以 CPU 不能写此寄存器。

位:	7	6	5	4	3	2	1	0
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
7 ~ 0		全 0	R	串行接收数据

表 17.4 SSCRL 的 DATS 位设定和 SSRDR 的对应表

	DATS[1:0] 的设定			
	00	01	10	11 (设定无效)
SSRDR0	有效	有效	有效	无效
SSRDR1	无效	有效	有效	无效
SSRDR2	无效	无效	有效	无效
SSRDR3	无效	无效	有效	无效

### 17.3.9 SS 移位寄存器 (SSTRSR)

SSTRSR 是发送和接收串行数据的移位寄存器。

在将发送数据从 SSTDR 传送到 SSTRSR 时, 如果 SSMR 的 MLS=0 时, 就将 SSTDR 的 bit0 (LSB 先通信) 传送到 SSTRSR 的 bit0; 如果 MLS=1, 就将 SSTDR 的 bit7 (MSB 先通信) 传送到 SSTRSR 的 bit0。然后从 SSTRSR 的 LSB (bit0) 开始按顺序将数据传送到 SSO 引脚, 进行串行数据的发送。

在接收时, 按照从 LSB (bit0) 开始接收的顺序, 将 SSI 引脚输入的串行数据设定到 SSTRSR。当结束 1 字节的数据接收时, SSTRSR 的数据就自动传送到 SSRDR。CPU 不能直接存取 SSTRSR。

位:	7	6	5	4	3	2	1	0
初始值:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

## 17.4 运行说明

### 17.4.1 传送时钟

可从 7 种内部时钟和外部时钟中选择传送时钟。在使用本模块时，需要通过 PFC 预先将 SSCK 引脚设定为有效。在 SSCRH 的 MSS=1 时，选择内部时钟并且 SSCK 引脚为输出引脚，当开始传送时，就从 SSCK 引脚输出 SSMR 的 CKS2 ~ CKS0 所设传送率的时钟；在 MSS=0 时，选择外部时钟并且 SSCK 引脚为输入引脚。

### 17.4.2 时钟的相位、极性和数据的关系

当 SSRL 的 SSUMS=0 时，通过组合 SSMR 的 CPOS 和 CPHS 来改变时钟的相位、极性和传送数据的关系，这些关系如图 17.2 所示。当 SSUMS=1 时，CPOS 的设定有效，但是 CPHS 的设定无效。

能通过设定 SSMR 的 MLS，选择是以 MSB 先还是以 LSB 先进行传送；当 MLS=0 时，按照从 LSB 到 MSB 的顺序进行传送；当 MLS=1 时，按照从 MSB 到 LSB 的顺序进行传送。

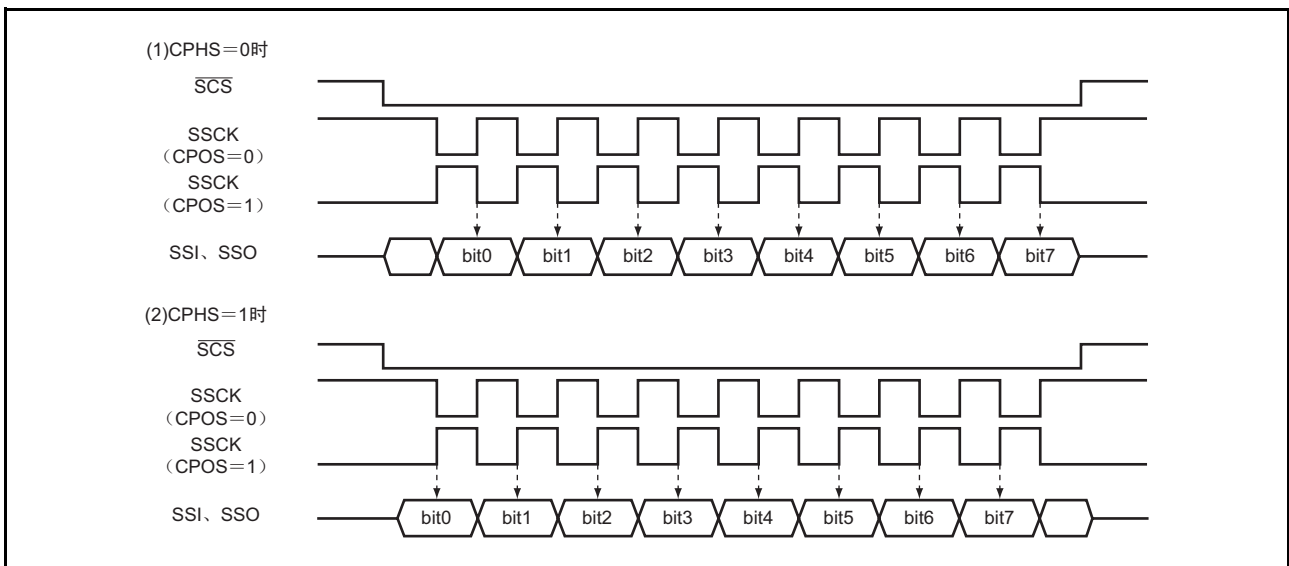


图 17.2 时钟的相位、极性和数据的关系

## 17.4.3 数据输入 / 输出引脚和移位寄存器的关系

通过组合 SSCRH 的 MSS、BIDE 和 SSCRL 的 SSUMS，改变数据输入 / 输出引脚和 SS 移位寄存器 (SSTRSR) 的连接关系，这些连接关系如图 17.3 所示。

在 BIDE=0 并且 MSS=1 (标准、主模式) 的状态下运行时，SSU 就从 SSO 引脚发送串行数据，从 SSI 引脚接收串行数据 (图 17.3(1))；在 BIDE=0 并且 MSS=0 (标准、从属模式) 的状态下运行时，SSU 就从 SSI 引脚发送串行数据，从 SSO 引脚接收串行数据 (图 17.3(2))。

在 BIDE=1 (双向模式) 的状态下，与主模式和从属模式无关，SSU 从 SSO 引脚发送或者接收串行数据 (图 17.3(3)、图 17.3(4))。但是不能同时将 TE 和 RE 置 1 进行同时发送和接收，只能选择 TE 和 RE 中的一个。

在 SSUMS=1 的状态下运行时，SSU 就从 SSO 引脚发送串行数据，从 SSI 引脚接收串行数据。当 MSS=1 时，SSU 就从 SSCK 引脚输出内部时钟；当 MSS=0 时，SSCK 引脚为输入引脚 (图 17.3(5)、图 17.3(6))。

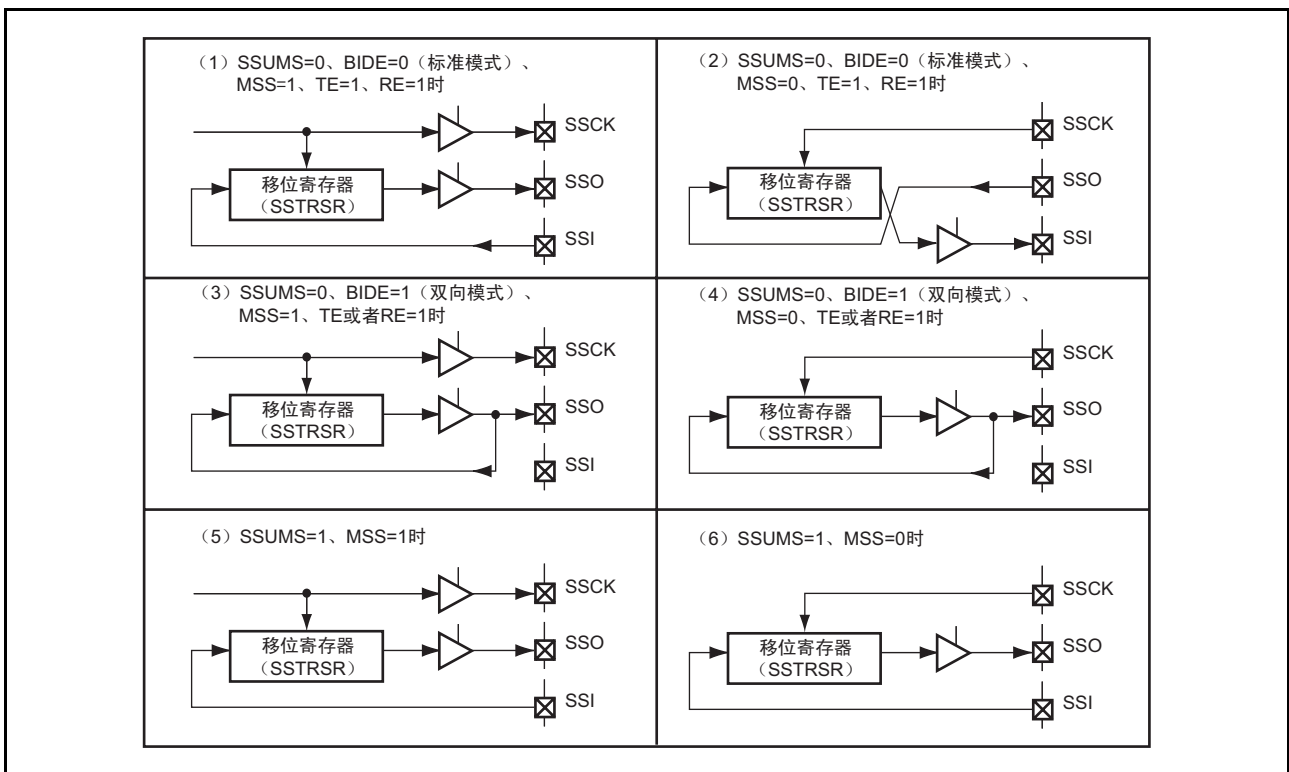


图 17.3 数据输入 / 输出引脚和移位寄存器的关系



## 17.4.4 各通信模式和引脚功能

SSU 通过设定各通信模式和寄存器来转换输入 / 输出引脚 (SSI、SSO、SSCK、 $\overline{\text{SCS}}$ ) 的功能。必须用端口 I/O 寄存器设定引脚的输入 / 输出转换。各通信模式和输入 / 输出引脚的关系如表 17.5 ~ 表 17.7 所示。

表 17.5 各通信模式和 SSI、SSO 引脚的状态

通信模式	寄存器状态					引脚状态	
	SSUMS	BIDE	MSS	TE	RE	SSI	SSO
SSU 通信模式	0	0	0	0	1	—	输入
				1	0	输出	—
			1	0	1	输入	—
				1	0	—	输出
				1	1	输出	输入
				1	0	—	输出
SSU (双向) 通信模式	0	1	0	0	1	—	输入
				1	0	—	输出
			1	0	1	—	输入
				1	0	—	输出
时钟同步 通信模式	1	0	0	0	1	输入	—
				1	0	—	输出
			1	0	1	输入	—
				1	0	—	输出
				0	1	输入	—
				1	0	—	输出
				1	输入	输出	

【符号说明】 —: 不用作 SSU 引脚

表 17.6 各通信模式和 SSCK 引脚的状态

通信模式	寄存器状态		引脚状态
	SSUMS	MSS	SSCK
SSU 通信模式	0	0	输入
		1	输出
时钟同步 通信模式	1	0	输入
		1	输出

表 17.7 各通信模式的  $\overline{\text{SCS}}$  引脚的状态

通信模式	寄存器状态				引脚状态	
	SSUMS	MSS	CSS1	CSS0	$\overline{\text{SCS}}$	
SSU 通信模式	0	0	x	x	输入	
		1	0	0	0	—
			0	1	1	输入
			1	0	0	自动输入 / 输出
			1	1	1	输出
时钟同步 通信模式	1	x	x	x	—	

【符号说明】 x: Don't care

—: 不用作 SSU 引脚

### 17.4.5 SSU 模式

在 SSU 模式中，使用时钟线 (SSCK)、数据输入线 (SSI 或者 SSO)、数据输出线 (SSI 或者 SSO) 和片选 ( $\overline{\text{SCS}}$ ) 进行数据传送。

也支持 1 个引脚用作数据输入线和数据输出线的双向模式。

#### (1) SSU 模式的初始设定

SSU 模式的初始设定例子如图 17.4 所示。在发送或者接收数据前，必须在将 SSER 的 TE 和 RE 清 0 后进行初始设定。

**【注】** 如果要更改运行模式和通信格式，就必须在将 TE 和 RE 清 0 后进行更改。必须注意：如果将 TE 清 0，TDRE 就被置 1。但是，即使将 RE 清 0，也保持 RDRF 和 ORER 的各标志以及 SSRDR 的内容。

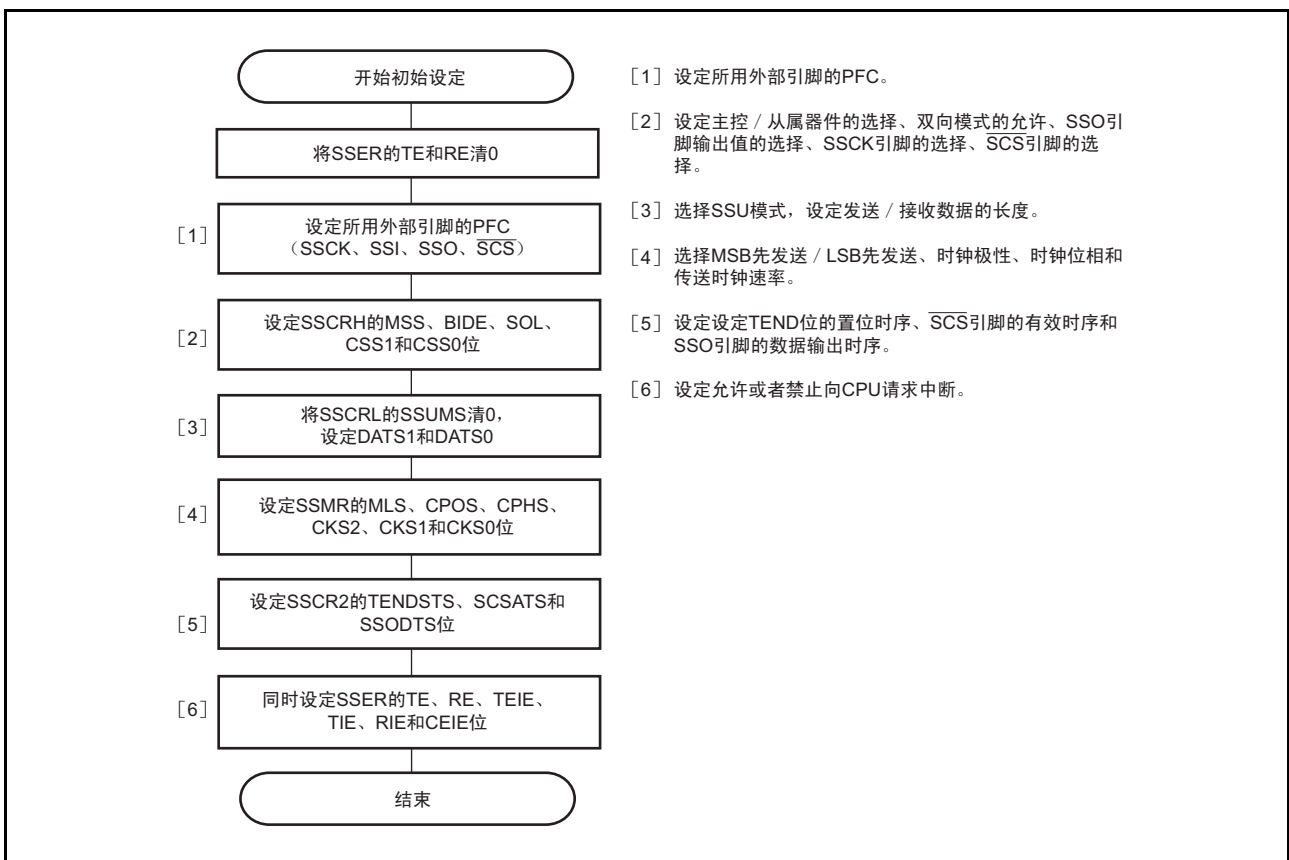


图 17.4 SSU 模式的初始设定例

#### (2) 数据的发送

发送时的运行例子和数据发送的流程图例子分别如图 17.5 和图 17.6 所示。

在发送数据时，SSU 的运行如下：

当 SSU 设定为主器件时，就输出传送时钟和数据；当设定为从属器件时，如果给  $\overline{\text{SCS}}$  引脚输入 Low 电平并且从 SSCK 引脚输入传送时钟，SSU 就和此传送时钟同步输出数据。

在 SSER 的 TE 被置 1 后，如果将发送数据写到 SSTDR，SSU 就将 SSSR 的 TDRE 自动清 0 并将数据从 SSTDR 传送到 SSTRSR，然后将 TDRE 置 1 开始发送。此时，如果 SSER 的 TIE 已被置 1，就产生 TXI 中断请求。

如果在 TDRE=0 的状态下 1 帧数据的传送结束, 就将数据从 SSTDR 传送到 SSTRSR, 开始下一帧的发送。  
 如果在 TDRE=1 的状态下发送第 8 位, SSSR 的 TEND 就被置 1 并保持此状态。此时, 如果 SSER 的 TEIE 已被置 1, 就产生 TEI 中断。在发送结束后, 当 SSMR 的 CPOS=0 时, SSCK 引脚固定为 High 电平; 当 CPOS=1 时, SSCK 引脚固定为 Low 电平。

不能在 SSSR 的 ORER 被置 1 的状态下发送数据, 必须在发送前确认 ORER 是否已被清 0。

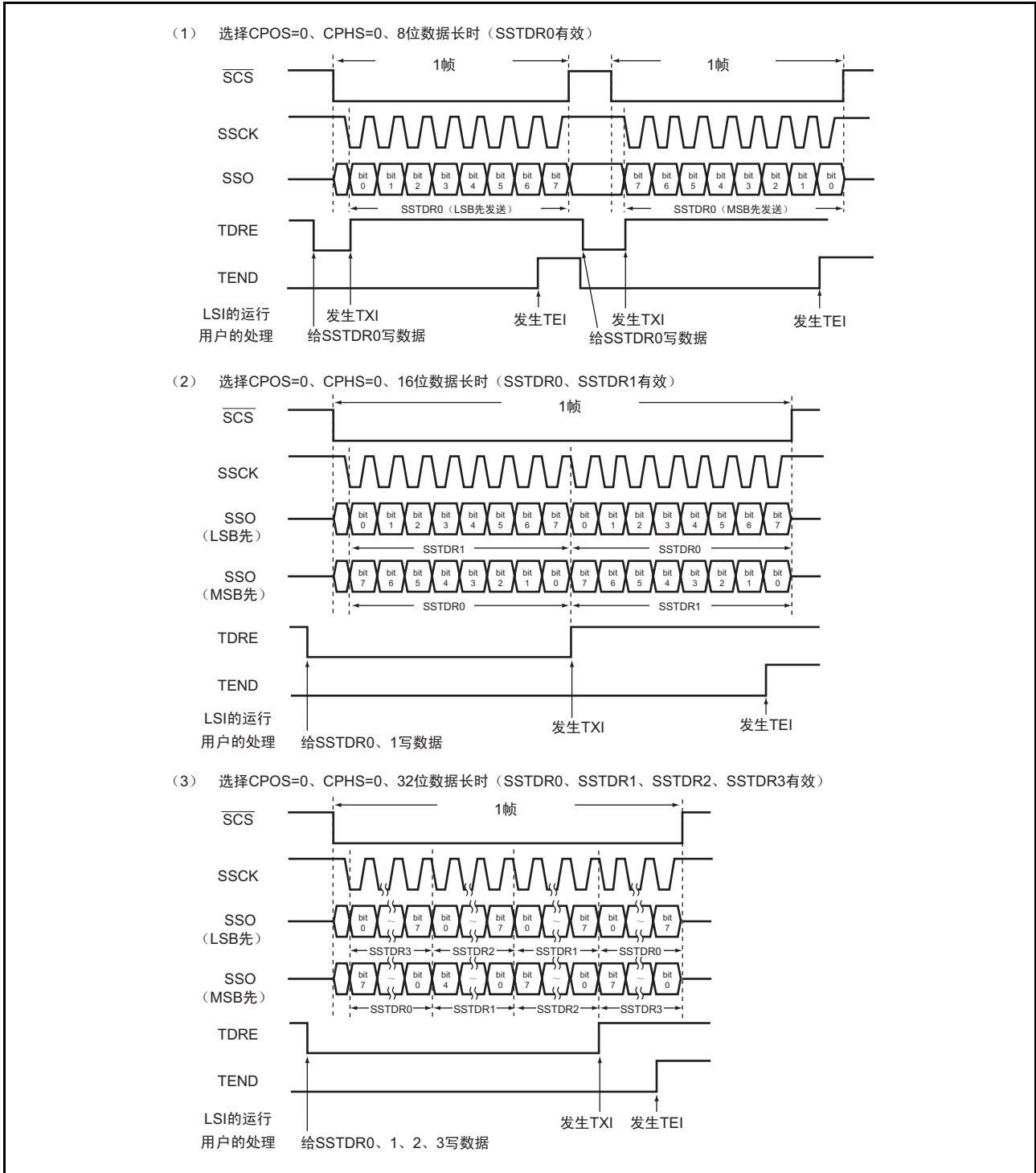


图 17.5 发送时的运行例 (SSU 模式)

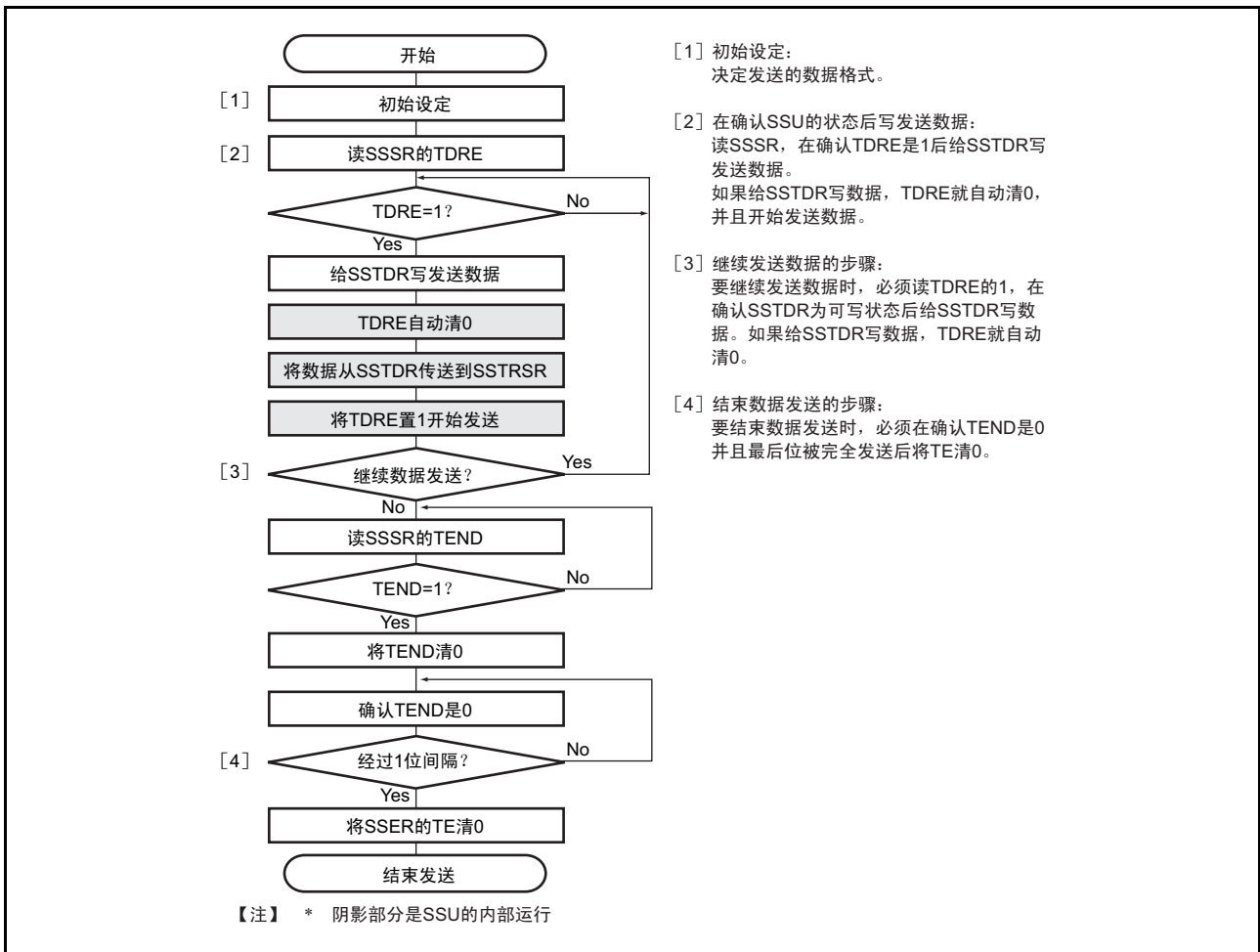


图 17.6 数据发送的流程图例子 (SSU 模式)

(3) 数据的接收

接收时的运行例子和数据接收的流程图例子分别如图 17.7 和图 17.8 所示。

在接收数据时，SSU 的运行如下：

当 SSER 的 RE 置 1 并虚读 SSRDR 时，SSU 开始接收运行。

当 SSU 设定为主器件时，输出传送时钟并且输入接收数据；当设定为从属器件时，如果给  $\overline{SCS}$  引脚输入 Low 电平并且从 SSCK 引脚输入传送时钟，SSU 就与此传送时钟同步输入接收数据。

在接收 1 帧的数据后，将 SSSR 的 RDRF 置 1，并且将接收数据保存到 SSRDR。此时，如果 SSER 的 RIE 已被置 1，就产生 RXI 中断请求。当读 SSRDR 时，RDRF 自动清 0。

如果在 RDRF=1 的状态下第 8 个时钟上升，SSSR 的 ORER 就被置 1，并且因发生溢出错误 (OEI) 而停止接收。因为在 ORER=1 的状态下不能进行接收，所以在重新开始接收时，必须将 ORER 清 0。

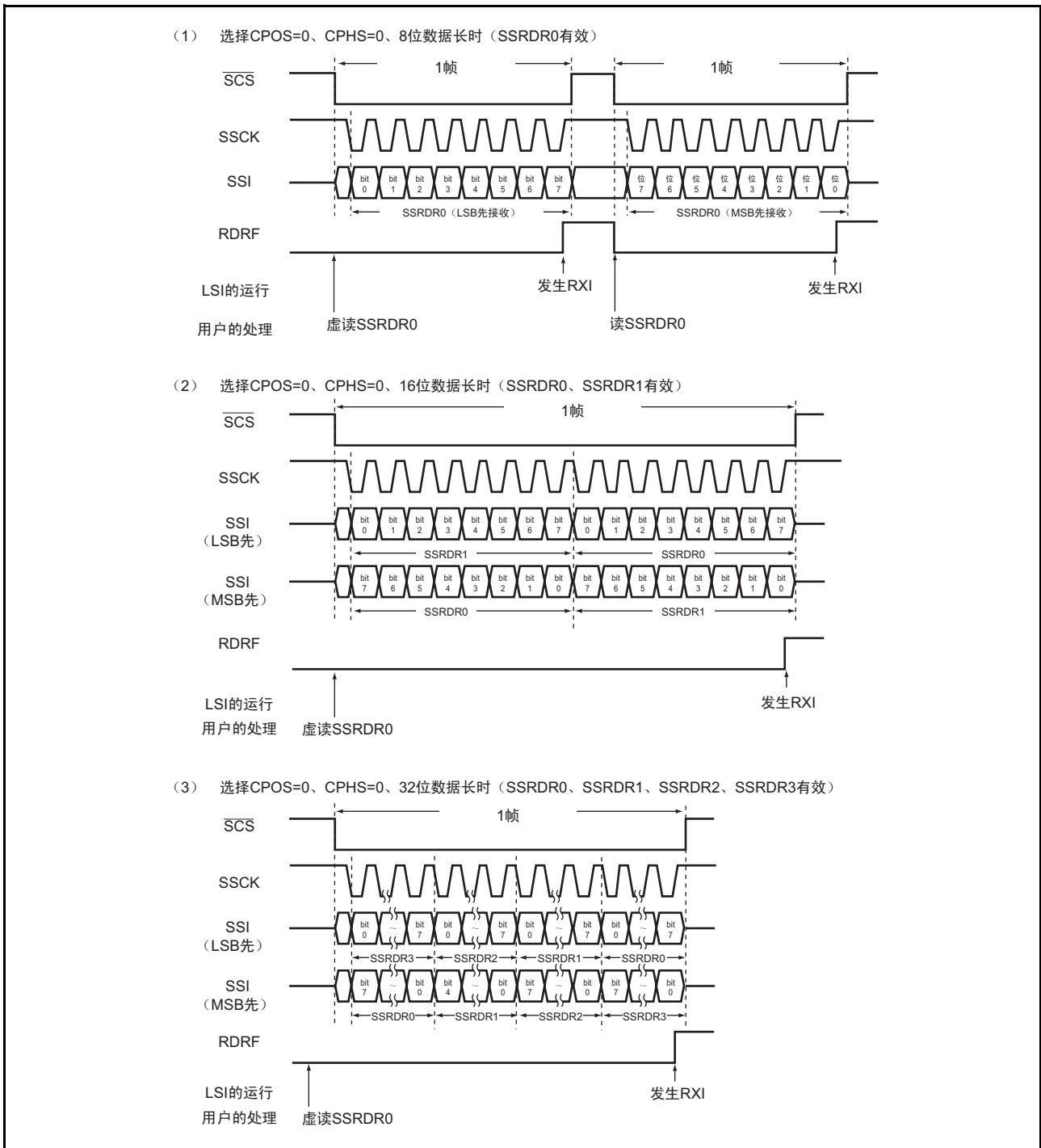
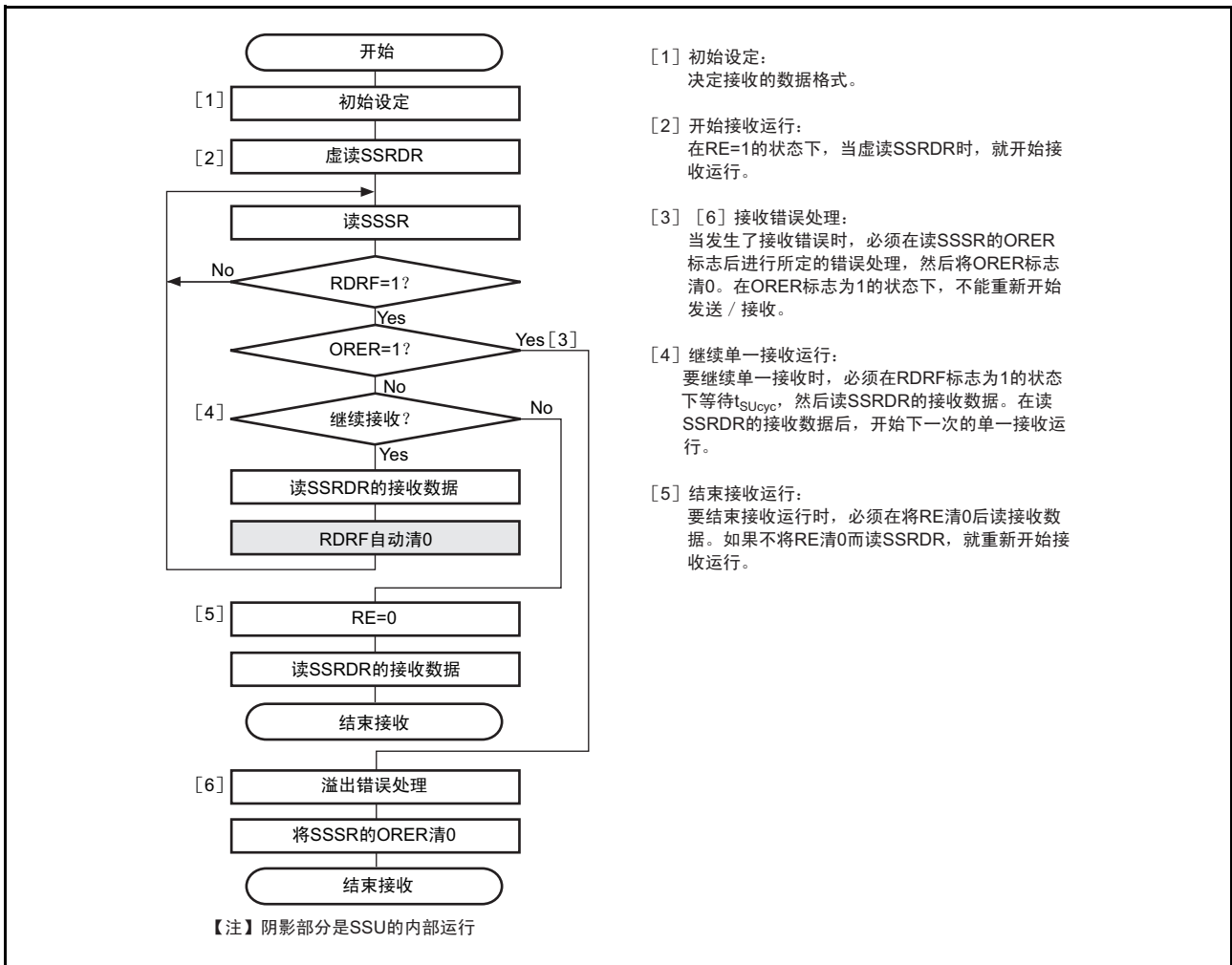


图 17.7 接收时的运行例 (SSU 模式)



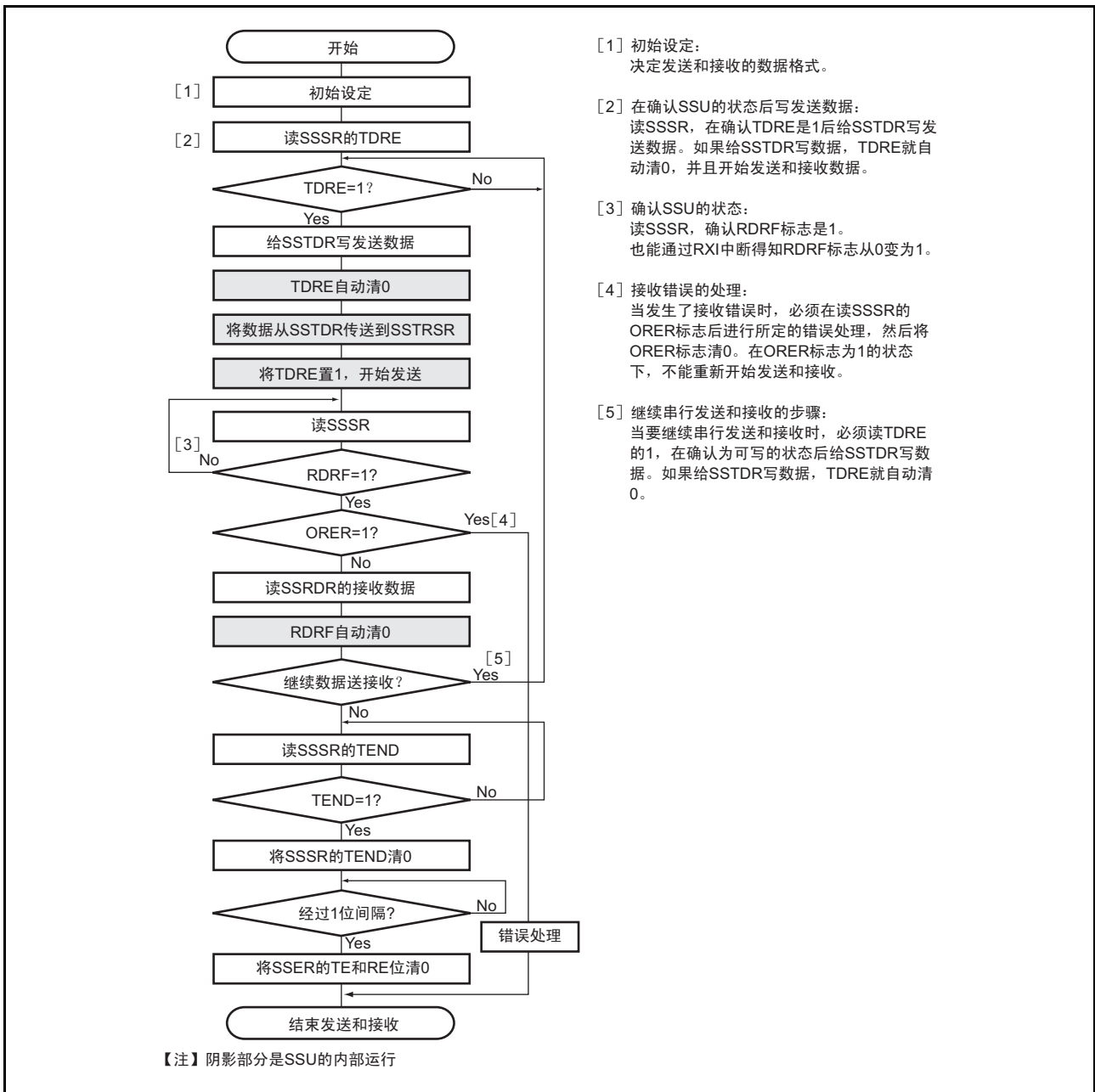
- [1] 初始设定:  
决定接收的数据格式。
- [2] 开始接收运行:  
在RE=1的状态下, 当虚读SSRDR时, 就开始接收运行。
- [3] [6] 接收错误处理:  
当发生了接收错误时, 必须在读SSSR的ORER标志后进行所定的错误处理, 然后将ORER标志清0。在ORER标志为1的状态下, 不能重新开始发送 / 接收。
- [4] 继续单一接收运行:  
要继续单一接收时, 必须在RDRF标志为1的状态下等待 $t_{SUcyc}$ , 然后读SSRDR的接收数据。在读SSRDR的接收数据后, 开始下一次的单一接收运行。
- [5] 结束接收运行:  
要结束接收运行时, 必须在将RE清0后读接收数据。如果不将RE清0而读SSRDR, 就重新开始接收运行。

图 17.8 数据接收的流程图例子 (SSU 模式)

#### (4) 数据的发送和接收

同时发送和接收的流程图例子如图 17.9 所示, 数据的发送和接收是发送数据和接收数据的复合运行。在  $TE=RE=1$  的状态下, 通过将发送数据写到 SSTDR, 开始发送和接收数据。

在从发送模式 ( $TE=1$ ) 或者接收模式 ( $RE=1$ ) 转换为发送和接收模式 ( $TE=RE=1$ ) 时, 必须在将 TE 和 RE 清 0 后进行转换。另外, 必须在确认 TEND、RDRF 和 ORER 已被清 0 后将 TE 和 RE 置 1。



- [1] 初始设定:  
决定发送和接收的数据格式。
- [2] 在确认SSU的状态后写发送数据:  
读SSSR, 在确认TDRE是1后给SSTD写发送数据。如果给SSTD写数据, TDRE就自动清0, 并且开始发送和接收数据。
- [3] 确认SSU的状态:  
读SSSR, 确认RDRF标志是1。  
也能通过RXI中断得知RDRF标志从0变为1。
- [4] 接收错误的处理:  
当发生了接收错误时, 必须在读SSSR的ORER标志后进行所定的错误处理, 然后将ORER标志清0。在ORER标志为1的状态下, 不能重新开始发送和接收。
- [5] 继续串行发送和接收的步骤:  
当要继续串行发送和接收时, 必须读TDRE的1, 在确认为可写的状态后给SSTD写数据。如果给SSTD写数据, TDRE就自动清0。

图 17.9 同时发送和接收的流程图例子 (SSU 模式)



### 17.4.6 SCS 引脚控制和冲突错误

如果设定 SSCRH 的 CSS1、CSS0=10 和 SSCRL 的 SSUMS=0 并且将 SSCRH 的 MSS 置 1，就在串行传送前和传送结束后，SCS 引脚为输入引脚 (Hi-Z)，并且检测冲突错误。在此期间，如果从 SCS 引脚输入 Low 电平，就发生冲突错误，SSSR 的 CE 被置位，并且 MSS 被清除。

**【注】** 在冲突错误被置位的状态下，不能进行以后的发送或者接收。在开始发送或者接收前，必须将 CE 清 0。

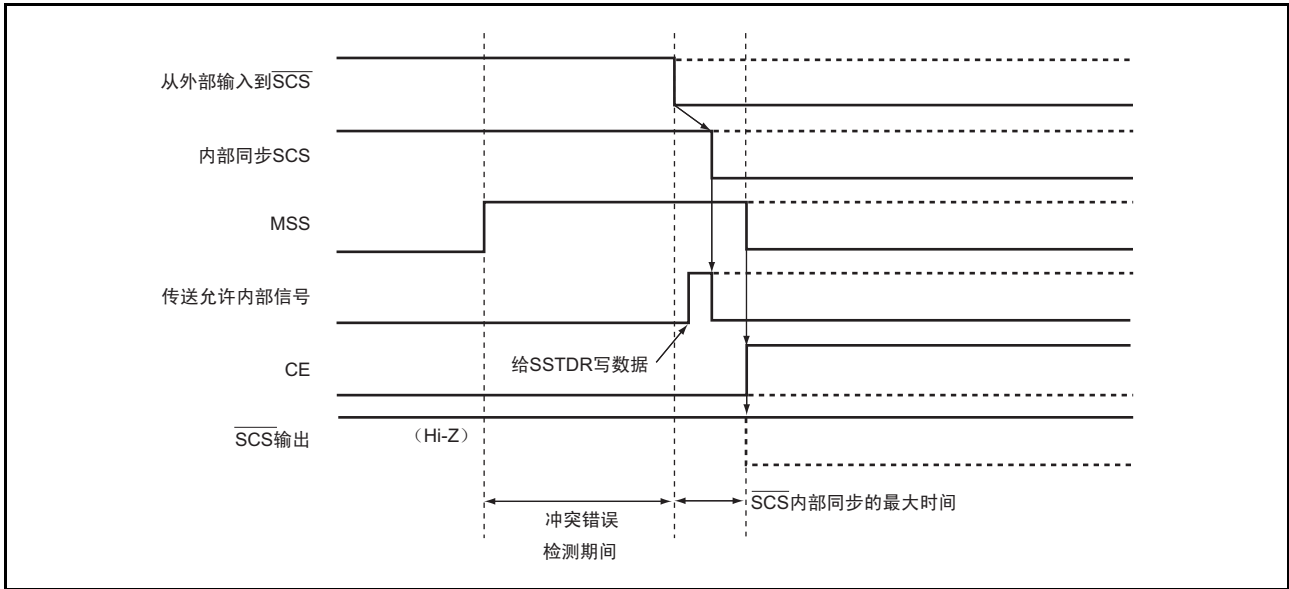


图 17.10 冲突错误的检测时序 (传送前)

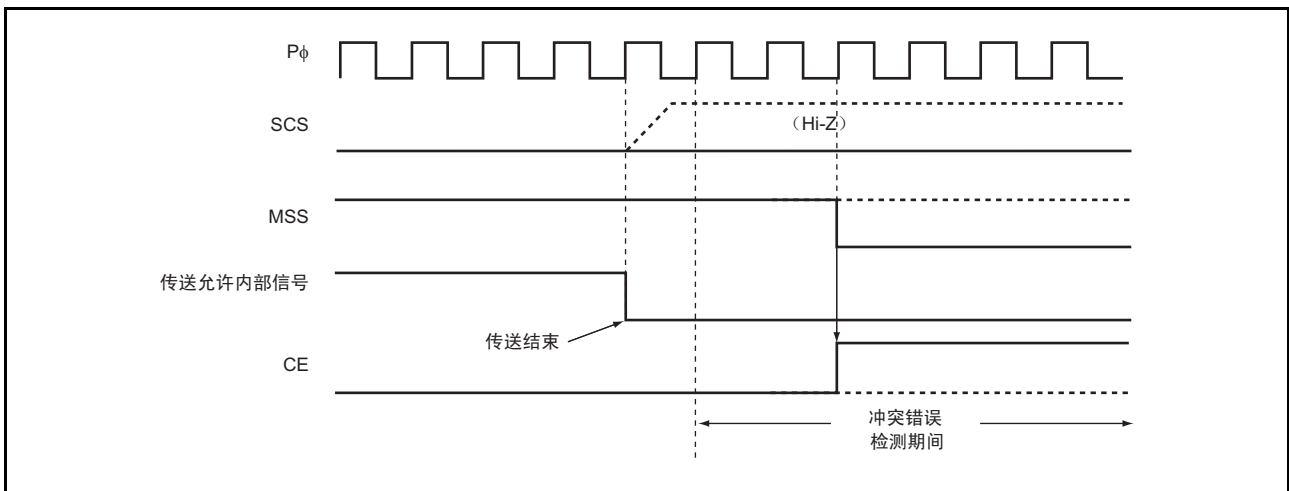


图 17.11 冲突错误的检测时序 (传送结束后)

### 17.4.7 时钟同步通信模式

在时钟同步通信模式中，使用时钟线 (SSCK)、数据输入线 (SSI) 和数据输出线 (SSO) 进行数据传送。

#### (1) 时钟同步通信模式的初始设定

时钟同步通信模式的初始设定例子如图 17.12 所示，在发送或者接收数据前，必须将 SSERTE 和 RE 清 0 进行初始设定。

**【注】** 如果要更改运行模式和通信格式，就必须在将 TE 和 RE 清 0 后进行更改。必须注意：如果将 TE 清 0，TDRE 就被置 1。但是，即使将 RE 清 0，也保持 RDRF 和 ORER 的各标志以及 SSRDR 的内容。



图 17.12 时钟同步通信模式的初始设定例

#### (2) 数据的发送

发送时的运行例子和发送数据的流程图例子分别如图 17.13 和图 17.14 所示。

在时钟同步通信模式中发送数据时，SSU 的运行如下：

当 SSU 设定为主器件时，输出传送时钟和数据；当 SSU 设定为从属器件时，如果从 SSCK 引脚输入传送时钟，SSU 就和此传送时钟同步输出数据。

在 SSER 的 TE 被置 1 后，如果将发送数据写到 SSTDR，SSU 就将 SSSR 的 TDRE 自动清 0 并将数据从 SSTDR 传送到 SSTRSR，然后将 TDRE 置 1 开始发送。此时，如果 SSER 的 TIE 已被置 1，就产生 TXI 中断请求。

如果在 TDRE=0 的状态下 1 帧的数据传送结束，就将数据从 SSTDR 传送到 SSTRSR，开始下一帧的发送。如果在 TDRE=1 的状态下送出第 8 位，SSSR 的 TEND 就被置 1 并保持此状态。此时，如果 SSER 的 TEIE 已被置 1，就产生 TEI 中断请求。

不能在 SSSR 的 ORER 被置 1 的状态下发送数据，必须在发送前确认 ORER 是否已被清 0。

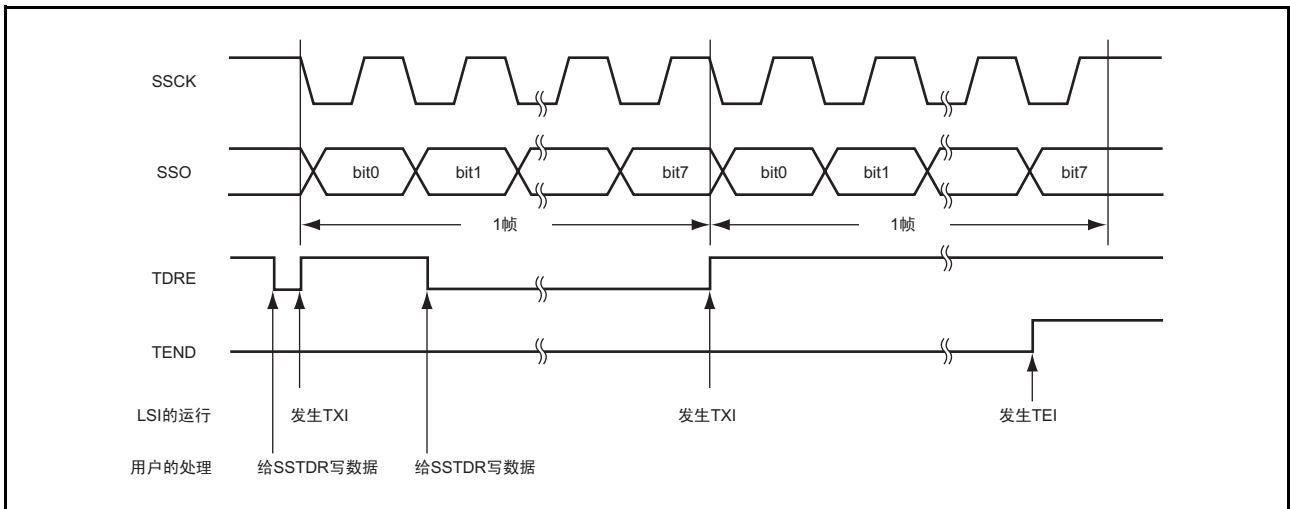


图 17.13 发送时的运行例 (时钟同步通信模式)

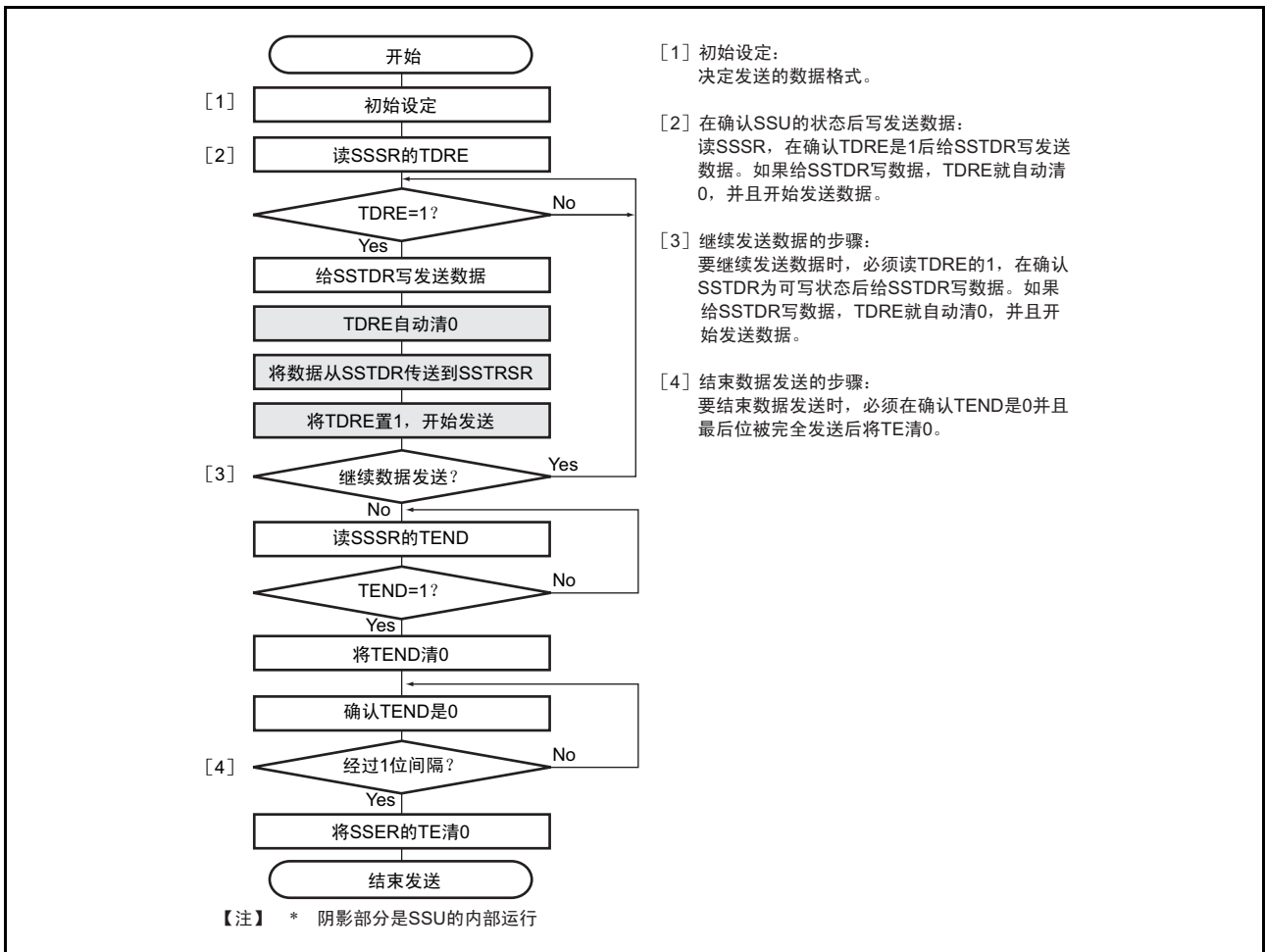


图 17.14 发送数据的流程图例子 (时钟同步通信模式)

(3) 数据的接收

接收时的运行例子和接收数据的流程图例子分别如图 17.15 和图 17.16 所示。

在接收数据时, SSU 的运行如下:

当 SSER 的 RE 置 1 时，SSU 开始接收运行。

当 SSU 设定为主器件时，输出传送时钟并且输入接收数据；当设定为从属器件时，如果从 SSCK 引脚输入传送时钟，SSU 就和此传送时钟同步输入接收数据。

在接收 1 帧的数据后，将 SSSR 的 RDRF 置 1，并且将接收数据保存到 SSRDR。此时，如果 SSER 的 RIE 已被置 1，就产生 RXI 中断请求。当读 SSRDR 时，RDRF 自动清 0。

如果在 RDRF=1 的状态下第 8 个时钟上升，SSSR 的 ORER 就被置 1，并且因发生溢出错误 (OEI) 而停止接收。因为在 ORER=1 的状态下不能进行接收，所以在重新开始接收时，必须将 ORER 清 0。

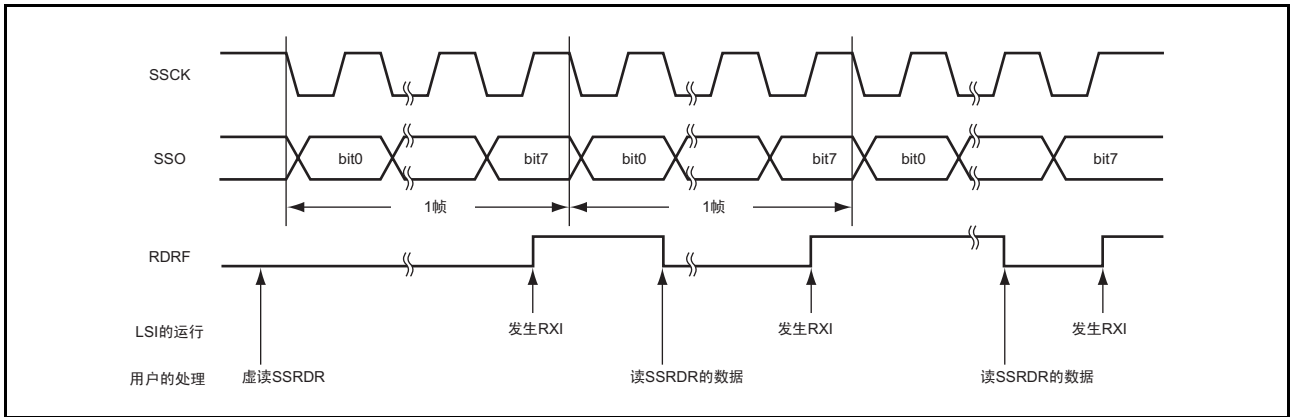


图 17.15 接收时的运行例 (时钟同步通信模式)

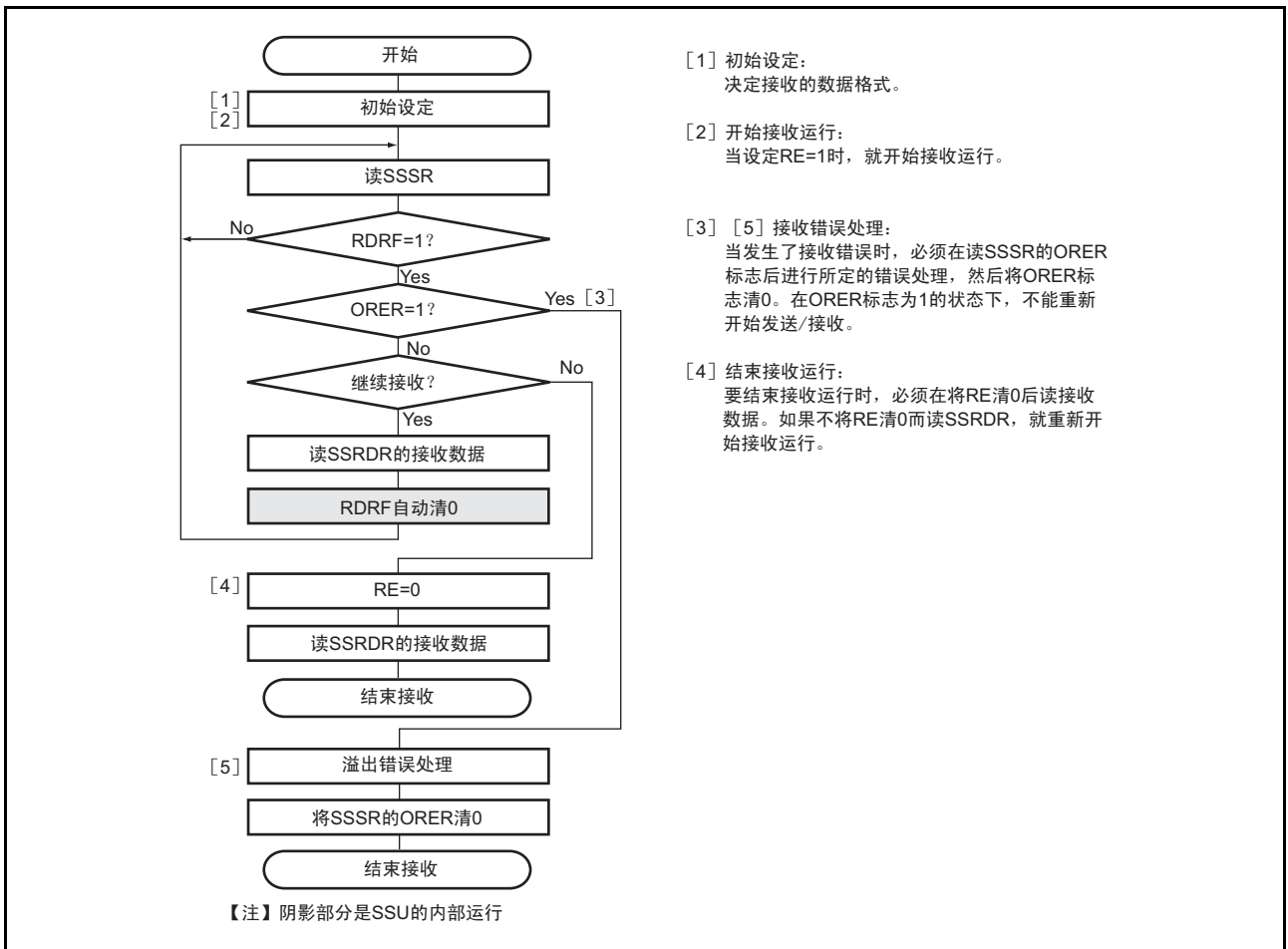


图 17.16 接收数据的流程图例子 (时钟同步通信模式)

(4) 数据的发送和接收

同时发送和接收的流程图例子如图 17.17 所示，数据的发送和接收是发送数据和接收数据的复合运行。在 TE=RE=1 的状态下，通过将发送数据写到 SSTDR，开始发送和接收数据。

在从发送模式 (TE=1) 或者接收模式 (RE=1) 转换为发送和接收模式 (TE=RE=1) 时，必须在将 TE 和 RE 清 0 后进行转换。另外，必须在确认 TEND、RDRF 和 ORER 已被清 0 后将 TE 和 RE 置 1。

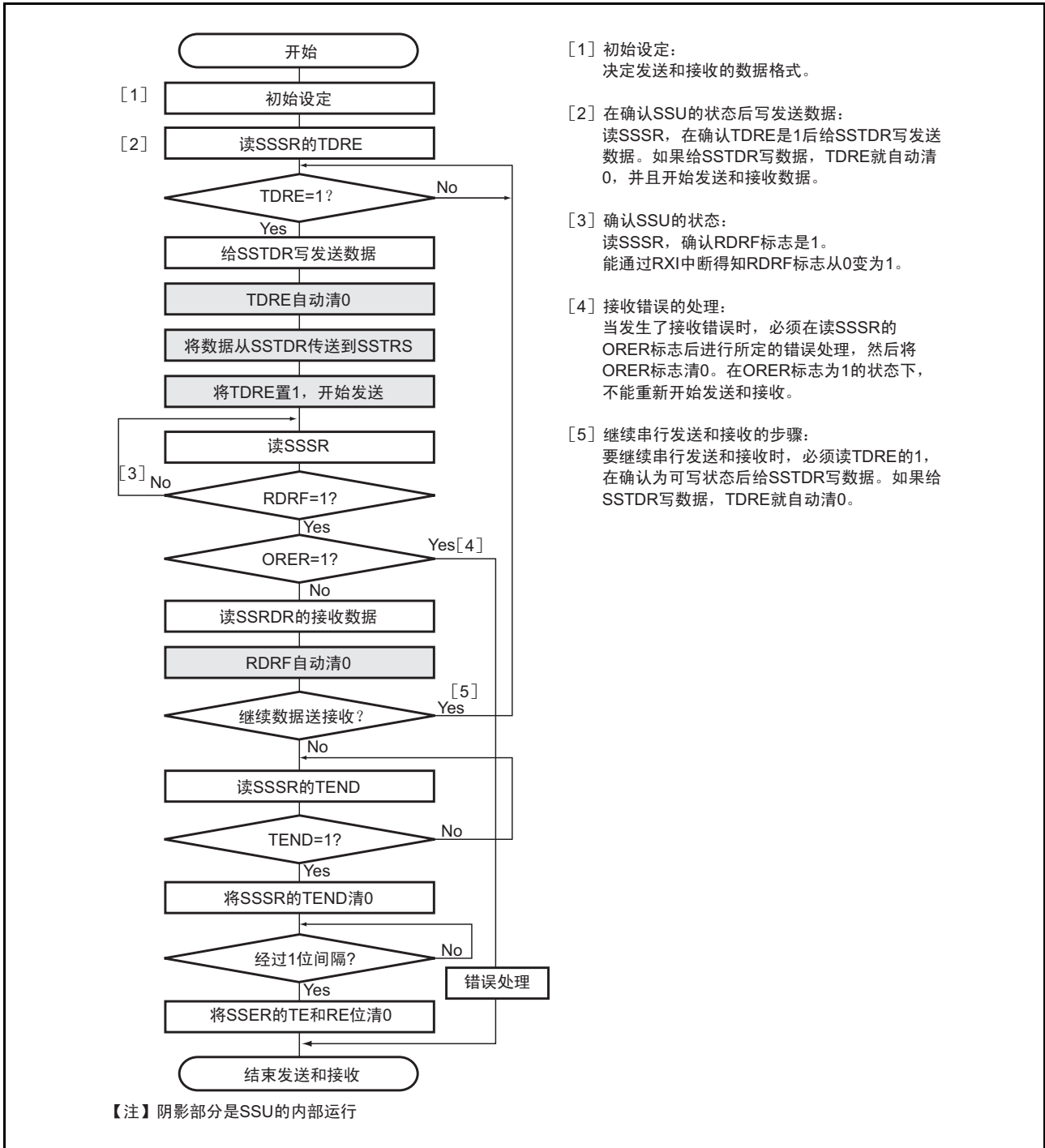


图 17.17 同时发送和接收的流程图例子 (时钟同步通信模式)

## 17.5 SSU 的中断源和 DTC

SSU 的中断请求有溢出错误、冲突错误、接收数据满、发送数据空和发送结束中断，还能通过接收数据满、发送数据空的中断请求启动 DTC 进行数据传送。

由于溢出错误、冲突错误的中断请求被分配到 SSERI 的向量地址，发送数据空、发送结束的中断请求被分配到 SSTXI 的向量地址，所以需要根据标志判断中断源，中断源如表 17.8 所示。

当表 17.8 的中断条件成立时，就产生中断请求。必须通过 CPU 或者 DTC 的数据传送来清除中断源。

表 17.8 SSU 中断源

名称	中断源	略称	中断条件	DTC 的启动
SSERI	溢出错误	SSOEI	(RIE=1) · (ORER=1)	—
	冲突错误	SSCEI	(CEIE=1) · (CE=1)	—
SSRXI	接收数据满	SSRXI	(RIE=1) · (RDRF=1)	○
SSTXI	发送数据空	SSTXI	(TIE=1) · (TDRE=1)	○
	发送结束	SSTEI	(TEIE=1) · (TEND=1)	—

## 17.6 使用注意事项

### 17.6.1 模块待机模式的设定

SSU 能通过待机控制寄存器，禁止或者允许本模块的运行，初始值为停止 SSU 的运行。能通过解除模块待机模式进行寄存器的存取，详细内容请参照“第 26 章 低功耗模式”。

### 17.6.2 SSTDR 和 SSRDR 寄存器的存取

不能存取由 SSCRL 寄存器的 DATS 位设定为无效的 SSTDR 和 SSRDR 寄存器，否则有可能无法正常进行以后的发送和接收。

### 17.6.3 SSU 从属模式中的连续发送和接收时的注意事项

在 SSU 从属模式中进行连续发送和接收时，必须按帧将  $\overline{\text{SCS}}$  引脚设定为无效（高电平）。如果  $\overline{\text{SCS}}$  引脚有效（低电平）状态超过 1 帧，就不能正常地发送和接收。

## 第 18 章 I<sup>2</sup>C 总线接口 2 (I<sup>2</sup>C2)

I<sup>2</sup>C 总线接口 2 基于菲利普斯公司提倡的 I<sup>2</sup>C 总线 (Inter IC Bus) 接口方式, 具有子集功能。必须注意: 控制 I<sup>2</sup>C 总线的一部分寄存器结构和菲利普斯公司不同。

I<sup>2</sup>C 总线接口 2 的框图和输入 / 输出引脚的外部电路连接例分别如图 18.1 和图 18.2 所示。

### 18.1 特点

- 可选择 I<sup>2</sup>C 总线格式或者时钟同步串行格式
- 能连续发送和连续接收  
因为移位寄存器、发送数据寄存器和接收数据寄存器各自独立, 所以能进行连续发送和连续接收
- 能设定模块待机模式

#### I<sup>2</sup>C 总线格式

- 在主模式中, 自动生成开始条件和停止条件
- 在接收时, 可选择应答的输出电平
- 在发送时, 自动装入应答位
- 内置了位同步功能  
在主模式中, 按位监视 SCL 状态, 自动取得同步。在尚未做好传送准备时, 将 SCL 置为 Low 电平使之处于等待状态。
- 中断源: 6 种  
发送数据空 (包括从属地址一致时)、发送结束、接收数据满 (包括从属地址一致时)、仲裁失败、NACK 检测、停止条件的检测  
能通过发送数据空请求或者接收数据满请求, 启动数据传送控制器 (DTC) 进行数据传送。
- 能直接驱动总线  
在选择总线驱动功能时, SCL 引脚和 SDA 引脚为 NMOS 漏极开路输出。

#### 时钟同步串行格式

- 中断源: 4 种  
发送数据空、发送结束、接收数据满、溢出错误  
能通过发送数据空请求或者接收数据满请求, 启动数据传送控制器 (DTC) 进行数据传送。

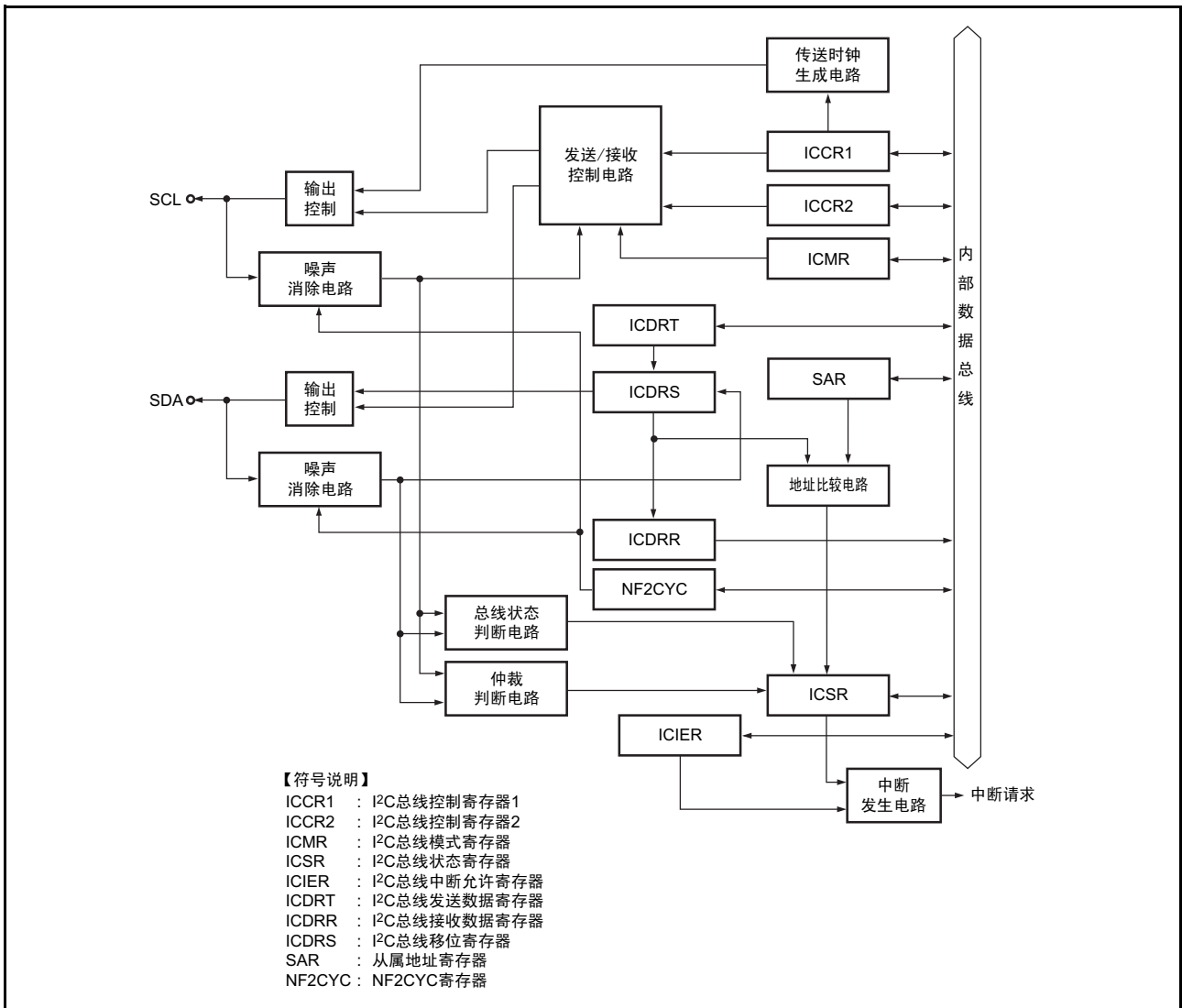


图 18.1 I<sup>2</sup>C 总线接口 2 的框图

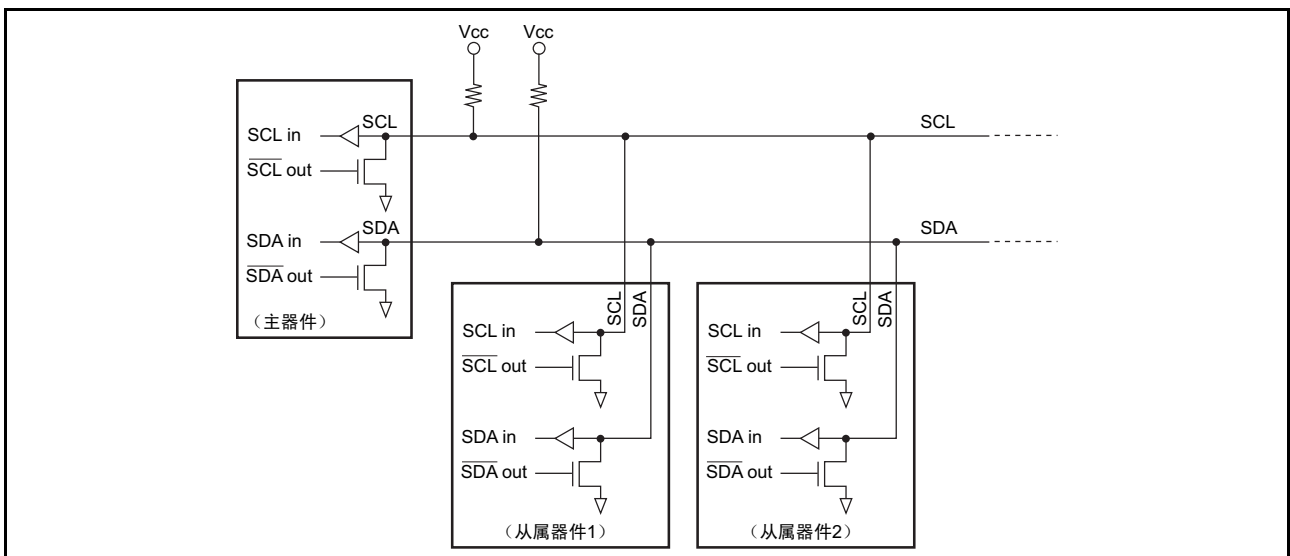


图 18.2 输入 / 输出引脚的外部电路连接例



## 18.2 输入 / 输出引脚

I<sup>2</sup>C 总线接口 2 使用的引脚结构如表 18.1 所示。

表 18.1 引脚结构

名称	引脚名	输入 / 输出	功 能
串行时钟引脚	SCL	输入 / 输出	I <sup>2</sup> C 串行时钟的输入 / 输出引脚
串行数据引脚	SDA	输入 / 输出	I <sup>2</sup> C 串行数据的输入 / 输出引脚

## 18.3 寄存器说明

I<sup>2</sup>C 总线接口 2 有以下寄存器，有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 27 章 寄存器一览”。

表 18.2 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
I <sup>2</sup> C 总线控制寄存器 1	ICCR1	R/W	H'00	H'FFFFCD80	8
I <sup>2</sup> C 总线控制寄存器 2	ICCR2	R/W	H'7D	H'FFFFCD81	8
I <sup>2</sup> C 总线模式寄存器	ICMR	R/W	H'38	H'FFFFCD82	8
I <sup>2</sup> C 总线中断允许寄存器	ICIER	R/W	H'00	H'FFFFCD83	8
I <sup>2</sup> C 总线状态寄存器	ICSR	R/W	H'00	H'FFFFCD84	8
从属地址寄存器	SAR	R/W	H'00	H'FFFFCD85	8
I <sup>2</sup> C 总线发送数据寄存器	ICDRT	R/W	H'FF	H'FFFFCD86	8
I <sup>2</sup> C 总线接收数据寄存器	ICDRR	R/W	H'FF	H'FFFFCD87	8
NF2CYC 寄存器	NF2CYC	R/W	H'00	H'FFFFCD88	8

18.3.1 I<sup>2</sup>C 总线控制寄存器 1 (ICCR1)

ICCR1 是 8 位可读写寄存器，进行 I<sup>2</sup>C 总线接口 2 的运行 / 停止和发送 / 接收的控制以及主模式 / 从属模式、发送模式 / 接收模式和主模式传送时钟频率的选择。

位:	7	6	5	4	3	2	1	0
	ICE	RCVD	MST	TRS	CKS[3:0]			
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7	ICE	0	R/W	I <sup>2</sup> C 总线接口 2 的允许 0: 本模块处于功能停止状态 1: 本模块处于可传送状态 (SCL/SDA 处于总线驱动状态)
6	RCVD	0	R/W	禁止接收 设定允许或者禁止在 TRS=0 的状态下不读 ICDRR 的连续接收运行。在主接收模式下，如果在 SCL 的第 8 个时钟的上升沿为止还不能读 ICDRR 时，必须设定 RCVD=1，并且按每个字节进行接收。 0: 允许连续接收运行 1: 禁止连续接收运行
5 4	MST TRS	0 0	R/W R/W	主模式和从属模式的选择 发送模式和接收模式的选择 如果在 I <sup>2</sup> C 总线格式的主模式中总线竞争失败，就通过硬件对 MST 和 TRS 进行复位，变为从属接收模式。另外，必须在传送帧期间更改 TRS。在从属接收模式中，如果产生开始条件后的 7 位与 SAR 设定的从属地址一致并且第 8 位是 1，TRS 就自动置 1。如果在时钟同步串行格式的主接收模式中发生溢出错误，MST 就被清 0，变为从属接收模式。 根据 MST 和 TRS 的组合，设定以下运行模式。如果选择时钟同步串行格式并且 MST=1，就为时钟输出。 00: 从属接收模式 01: 从属发送模式 10: 主接收模式 11: 主发送模式
3 ~ 0	CKS[3:0]	0000	R/W	传送时钟的选择 3 ~ 0 在主模式中，必须根据需要的传送率 (参照表 18.3) 进行设定。在从属模式中，用于确保发送模式中的数据准备时间。当 CKS3=0 时，数据准备时间为 10 个 tpcyc；当 CKS3=1 时，为 20 个 tpcyc (tpcyc 表示 P <sub>φ</sub> 周期)。

表 18.3 传送率

bit3	bit2	bit1	bit0	时钟	传送率					
CKS3	CKS2	CKS1	CKS0		P $\phi$ =10MHz	P $\phi$ =16MHz	P $\phi$ =20MHz	P $\phi$ =25MHz	P $\phi$ =33MHz	P $\phi$ =40MHz
0	0	0	0	P $\phi$ /28	357kHz	571kHz	714kHz	893kHz	1.18MHz	1.43MHz
			1	P $\phi$ /40	250kHz	400kHz	500kHz	625kHz	825kHz	1.00MHz
		1	0	P $\phi$ /48	208kHz	333kHz	417kHz	521kHz	688kHz	833kHz
			1	P $\phi$ /64	156kHz	250kHz	313kHz	391kHz	516kHz	625kHz
	1	0	0	P $\phi$ /80	125kHz	200kHz	250kHz	313kHz	413kHz	500kHz
			1	P $\phi$ /100	100kHz	160kHz	200kHz	250kHz	330kHz	400kHz
		1	0	P $\phi$ /112	89.3kHz	143kHz	179kHz	223kHz	295kHz	357kHz
			1	P $\phi$ /128	78.1kHz	125kHz	156kHz	195kHz	258kHz	313kHz
1	0	0	0	P $\phi$ /112	89.3kHz	143kHz	179kHz	223kHz	295kHz	357kHz
			1	P $\phi$ /160	62.5kHz	100kHz	125kHz	156kHz	206kHz	250kHz
		1	0	P $\phi$ /192	52.1kHz	83.3kHz	104kHz	130kHz	172kHz	208kHz
			1	P $\phi$ /256	39.1kHz	62.5kHz	78.1kHz	97.7kHz	129kHz	156kHz
	1	0	0	P $\phi$ /320	31.3kHz	50.0kHz	62.5kHz	78.1kHz	103kHz	125kHz
			1	P $\phi$ /400	25.0kHz	40.0kHz	50.0kHz	62.5kHz	82.5kHz	100kHz
		1	0	P $\phi$ /448	22.3kHz	35.7kHz	44.6kHz	55.8kHz	73.7kHz	89.3kHz
			1	P $\phi$ /512	19.5kHz	31.3kHz	39.1kHz	48.8kHz	64.5kHz	78.1kHz

18.3.2 I<sup>2</sup>C 总线控制寄存器 2 (ICCR2)

ICCR2 是 8 位可读写寄存器，控制开始 / 停止条件的产生、SDA 引脚的操作、SCL 引脚的监视以及 I<sup>2</sup>C 控制部的复位。

位:	7	6	5	4	3	2	1	0
	BBSY	SCP	SDAO	SDAOP	SCLO	-	IICRST	-
初始值:	0	1	1	1	1	1	0	1
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R

位	位名	初始值	R/W	说 明
7	BBSY	0	R/W	总线忙 有表示 I <sup>2</sup> C 总线占有 / 释放状态的标志功能和主模式开始 / 停止条件的产生功能。在时钟同步串行格式时，此位的读取值总是 0。在 I <sup>2</sup> C 总线格式时并且在 SCL=High 电平的状态下，如果 SDA 从 High 电平变为 Low 电平，就认为已产生开始条件，此位被置 1；如果 SDA 从 Low 电平变为 High 电平，就认为已产生停止条件，此位被清 0。在产生开始条件时，给 BBSY 写 1 并给 SCP 写 0，在重新产生开始条件时也同样；在产生停止条件时，给 BBSY 写 0 并给 SCP 写 0。
6	SCP	1	R/W	开始 / 停止条件的产生禁止位 在主模式中，SCP 位控制开始 / 停止条件的产生。在产生开始条件时，给 BBSY 写 1 并给 SCP 写 0，在重新发送开始条件时也同样；在产生停止条件时，给 BBSY 写 0 并给 SCP 写 0。此位的读取值总是 1，即使写 1 也不保存数据。
5	SDAO	1	R/W	SDA 输出值的控制 要更改 SDA 的输出电平时，SDAO 和 SDAOP (bit4) 的组合使用。另外，不能在传送中进行此位的操作。 0: 在读操作时，SDA 引脚的输出为 Low 电平 在写操作时，将 SDA 引脚的输出更改为 Low 电平 1: 在读操作时，SDA 引脚的输出为 High 电平 在写操作时，将 SDA 引脚的输出更改为 Hi-Z (通过外部上拉电阻输出 High 电平)
4	SDAOP	1	R/W	SDAO 的写保护 通过改写 SDAOP 位来更改 SDA 引脚的输出电平。要更改输出电平时，给 SDAOP 和 SDAOP 都写 0 或者给 SDAOP 写 1 而给 SDAOP 写 0。此位的读取值总是 1。
3	SCLO	1	R	SCLO 监视 SCL 的输出电平。在读操作时，如果 SCLO 是 1，SCL 引脚的输出就为 High 电平；如果 SCLO 是 0，SCL 引脚的输出就为 Low 电平。
2	—	1	R	保留位 读写的值总是 1。
1	IICRST	0	R/W	IIC 控制部的复位 IICRST 对 I <sup>2</sup> C 寄存器以外的控制部进行复位。在 I <sup>2</sup> C 运行中因通信不正常等引起中止时，如果将 IICRST 位设定为 1，就能对 I <sup>2</sup> C 的一部分寄存器和控制部进行复位。
0	—	1	R	保留位 读写的值总是 1。

18.3.3 I<sup>2</sup>C 总线模式寄存器 (ICMR)

ICMR 是 8 位可读写寄存器，选择 MSB 先 /LSB 先和传送位数。

位:	7	6	5	4	3	2	1	0
	MLS	-	-	-	BCWP	BC[2:0]		
初始值:	0	0	1	1	1	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明																		
7	MLS	0	R/W	MSB 先 /LSB 先的选择 0: MSB 先 1: LSB 先 当用于 I <sup>2</sup> C 总线格式时，必须将此位设定为 0。																		
6	—	0	R	保留位 读写的值总是 0。																		
5、4	—	全 1	R	保留位 读写的值总是 1。																		
3	BCWP	1	R/W	BC 的写保护 控制 BC2 ~ BC0 的写操作。在改写 BC2 ~ BC0 时，必须将此位设定为 0。另外，不能在时钟同步串行格式时改写 BC。 0: 在写操作时，设定 BC2 ~ BC0 的值 1: 在写操作时，总是读 1 在写操作时，BC2 ~ BC0 的设定值无效																		
2 ~ 0	BC[2:0]	000	R/W	<p>位计数器 2 ~ 0 指定下一次要传送的数据位数。如果读这些位，就能得知剩余的传送位数。在 I<sup>2</sup>C 总线格式时，数据加上 1 位应答位进行传送。必须在传送帧期间设定这些位。如果要设定非 000 值，就必须在 SCL 为 Low 电平状态下进行设定。在包含应答位的数据传送结束后，这些位自动返回 000。另外，检测出停止条件后，就会自动变为 111。通过上电复位、待机模式或者将 ICCR2 的 IICRST 置 1 来清除这些位。如果选择时钟同步串行格式，就不能改写此位。</p> <table style="width: 100%; border: none;"> <tr> <td style="width: 50%; vertical-align: top;">I<sup>2</sup>C 总线格式</td> <td style="width: 50%; vertical-align: top;">时钟同步串行格式</td> </tr> <tr> <td>000: 9 位</td> <td>000: 8 位</td> </tr> <tr> <td>001: 2 位</td> <td>001: 1 位</td> </tr> <tr> <td>010: 3 位</td> <td>010: 2 位</td> </tr> <tr> <td>011: 4 位</td> <td>011: 3 位</td> </tr> <tr> <td>100: 5 位</td> <td>100: 4 位</td> </tr> <tr> <td>101: 6 位</td> <td>101: 5 位</td> </tr> <tr> <td>110: 7 位</td> <td>110: 6 位</td> </tr> <tr> <td>111: 8 位</td> <td>111: 7 位</td> </tr> </table>	I <sup>2</sup> C 总线格式	时钟同步串行格式	000: 9 位	000: 8 位	001: 2 位	001: 1 位	010: 3 位	010: 2 位	011: 4 位	011: 3 位	100: 5 位	100: 4 位	101: 6 位	101: 5 位	110: 7 位	110: 6 位	111: 8 位	111: 7 位
I <sup>2</sup> C 总线格式	时钟同步串行格式																					
000: 9 位	000: 8 位																					
001: 2 位	001: 1 位																					
010: 3 位	010: 2 位																					
011: 4 位	011: 3 位																					
100: 5 位	100: 4 位																					
101: 6 位	101: 5 位																					
110: 7 位	110: 6 位																					
111: 8 位	111: 7 位																					

18.3.4 I<sup>2</sup>C 总线中断允许寄存器 (ICIER)

ICIER 是 8 位可读写寄存器，允许各种中断源、选择应答的有效 / 无效、设定发送应答以及确认接收应答。

位:	7	6	5	4	3	2	1	0
	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

位	位名	初始值	R/W	说 明
7	TIE	0	R/W	发送中断的允许 在 ICSR 的 TDRE 被置位时，允许或者禁止发送数据空的中断 (IITXI)。 0: 禁止发送数据空的中断请求 (IITXI) 1: 允许发送数据空的中断请求 (IITXI)
6	TEIE	0	R/W	发送结束中断的允许 在 ICSR 的 TDRE 是 1 的状态下第 9 个时钟上升时，TEIE 允许或者禁止发送结束中断 (IITEI)。能通过将 TEND 或者 TEIE 清 0 来解除 IITEI。 0: 禁止发送结束的中断请求 (IITEI) 1: 允许发送结束的中断请求 (IITEI)
5	RIE	0	R/W	接收中断的允许 在将接收数据从 ICDRS 传送到 ICDRR 并且 ICSR 的 RDRF 被置 1 时，RIE 允许或者禁止接收数据满的中断请求 (IIRXI) 以及时钟同步格式时的溢出错误的中断请求 (IIERI)。能通过将 RDRF 或者 RIE 清 0 来清除 IIRXI。 0: 禁止接收数据满的中断请求 (IIRXI) 1: 允许接收数据满的中断请求 (IIRXI)
4	NAKIE	0	R/W	NACK 接收中断的允许 在 ICSR 的 NACKF 或者 AL/OVE 被置位时，NAKIE 允许或者禁止 NACK 检测的中断请求 (IINAKI) 以及时钟同步格式时的溢出错误 (ICSR 的 OVE 置位) 的中断请求 (IIERI)。能通过将 NACKF、AL/OVE 或者 NAKIE 清 0 来清除 IINAKI。 0: 禁止 NACK 接收的中断请求 (IINAKI) 1: 允许 NACK 接收的中断请求 (IINAKI)
3	STIE	0	R/W	停止条件检测中断的允许 在 ICSR 的 STOP 被置位时，STIE 允许或者禁止停止条件检测的中断请求 (IISTPI)。 0: 禁止停止条件检测的中断请求 (IISTPI) 1: 允许停止条件检测的中断请求 (IISTPI)
2	ACKE	0	R/W	应答位的判断选择 0: 忽视接收应答的内容，进行连续传送。 1: 当接收应答是 1 时，中断传送。
1	ACKBR	0	R	接收应答 在发送模式中，保存从接收器件接收到的应答位的内容，写操作无效。通过将 ICCR2 的 BBSY 置 1 来清除此位。 0: 接收应答 =0 1: 接收应答 =1
0	ACKBT	0	R/W	发送应答 在接收模式中，设定在应答时要发送的位的内容。 0: 在应答时序中发送 0 1: 在应答时序中发送 1

18.3.5 I<sup>2</sup>C 总线状态寄存器 (ICSR)

ICSR 是 8 位可读写寄存器，确认各种中断请求标志及其状态。

位:	7	6	5	4	3	2	1	0
	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7	TDRE	0	R/W	发送数据空 [置位条件] <ul style="list-style-type: none"> <li>• 当发送数据从 ICDRT 传送到 ICDRS 并且 ICDRT 为空时</li> <li>• 当 TRS 被置位时</li> <li>• 当产生开始条件 (包含重新产生) 时</li> <li>• 在从属模式中, 从接收模式变为发送模式时</li> </ul> [清除条件] <ul style="list-style-type: none"> <li>• 当读到 1 的状态后写 0 时</li> <li>• 当数据写到 ICDRT 时</li> <li>• 当通过 IITXI 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位是 0 时</li> </ul>
6	TEND	0	R/W	发送结束 [置位条件] <ul style="list-style-type: none"> <li>• I<sup>2</sup>C 总线格式: 在 TDRE 是 1 的状态下 SCL 的第 9 个时钟上升时</li> <li>• 时钟同步串行格式: 当送出发送帧的最后位时</li> </ul> [清除条件] <ul style="list-style-type: none"> <li>• 当读到 1 的状态后写 0 时</li> <li>• 当数据写到 ICDRT 时</li> <li>• 当通过 IITXI 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位是 0 时</li> </ul>
5	RDRF	0	R/W	接收数据寄存器满 [置位条件] <ul style="list-style-type: none"> <li>• 当接收数据从 ICDRS 传送到 ICDRR 时</li> </ul> [清除条件] <ul style="list-style-type: none"> <li>• 当读到 1 的状态后写 0 时</li> <li>• 当读 ICDRR 时</li> <li>• 当通过 IIRXI 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位是 0 时</li> </ul>
4	NACKF	0	R/W	无应答的检测标志 * [置位条件] <ul style="list-style-type: none"> <li>• 在 ICIER 的 ACKE=1 的状态下, 发送时接收器件没有应答</li> </ul> [清除条件] <ul style="list-style-type: none"> <li>• 当读到 1 的状态后写 0 时</li> </ul>
3	STOP	0	R/W	停止条件的检测标志 [置位条件] <ul style="list-style-type: none"> <li>• 主模式: 在帧的传送结束后, 检测到停止条件时</li> <li>• 从属模式: 在检测到开始条件后的第 1 字节的从属地址和 SAR 设定的地址一致后, 检测到停止条件时</li> </ul> [清除条件] <ul style="list-style-type: none"> <li>• 当读到 1 的状态后写 0 时</li> </ul>

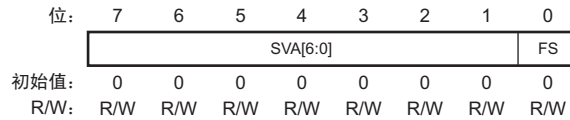
位	位名	初始值	R/W	说 明
2	AL/OVE	0	R/W	<p>仲裁失败标志 / 溢出错误标志</p> <p>在 I<sup>2</sup>C 总线格式时, AL/OVE 表示在主模式中总线竞争失败; 在时钟同步串行格式时, 表示在 RDRF=1 的状态下接收到最后的位。</p> <p>在多个主器件几乎同时要占有总线时, 如果 I<sup>2</sup>C 总线接口 2 监视到的 SDA 数据和自己输出的数据不同, 就将 AL 标志置 1, 表示由其他主器件占有总线。</p> <p>[置位条件]</p> <ul style="list-style-type: none"> <li>主发送模式: 在 SCL 的上升沿, 内部 SDA 和 SDA 引脚的电平不一致时</li> <li>主模式: 在检测到开始条件时, SDA 引脚为 High 电平时</li> <li>时钟同步串行格式: 在 RDRF=1 的状态下接收到最后的位时</li> </ul> <p>[清除条件]</p> <ul style="list-style-type: none"> <li>当读到 1 的状态后写 0 时</li> </ul>
1	AAS	0	R/W	<p>从属地址的识别标志</p> <p>在从属接收模式中, 如果紧接在开始条件后的第 1 帧和 SAR 的 SVA6 ~ SVA0 一致, 此位就被置位。</p> <p>[置位条件]</p> <ul style="list-style-type: none"> <li>在从属接收模式中检测到从属地址时</li> <li>在从属接收模式中检测到一般调用地址时</li> </ul> <p>[清除条件]</p> <ul style="list-style-type: none"> <li>当读到 1 的状态后写 0 时</li> </ul>
0	ADZ	0	R/W	<p>一般调用地址的识别标志</p> <p>此位在 I<sup>2</sup>C 总线格式的从属接收模式中有效</p> <p>[置位条件]</p> <ul style="list-style-type: none"> <li>在从属接收模式中检测到一般调用地址时</li> </ul> <p>[清除条件]</p> <ul style="list-style-type: none"> <li>当读到 1 的状态后写 0 时</li> </ul>

【注】\* 如果检测到 NACKF=1, 就必须通过传送结束的处理来清除 NACKF, 而且在 NACKF 被清除前不能进行以后的发送或者接收。



### 18.3.6 从属地址寄存器 (SAR)

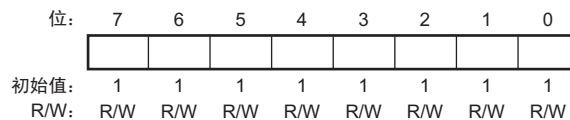
SAR 是 8 位可读写寄存器，选择格式以及设定从属地址。在 I<sup>2</sup>C 总线格式的从属模式中，如果在开始条件后送来的第 1 帧的高 7 位和 SAR 的高 7 位一致，SAR 就作为从属器件运行。



位	位名	初始值	R/W	说 明
7 ~ 1	SVA[6:0]	全 0	R/W	从属地址 6 ~ 0 设定与连接 I <sup>2</sup> C 总线的其他从属器件地址不同的唯一地址。
0	FS	0	R/W	格式的选择 0: 选择 I <sup>2</sup> C 总线格式 1: 选择时钟同步串行格式

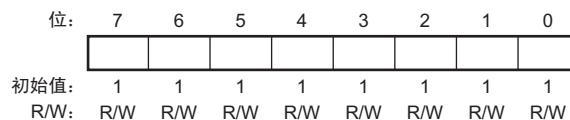
### 18.3.7 I<sup>2</sup>C 总线发送数据寄存器 (ICDRT)

ICDRT 是保存发送数据的 8 位可读写寄存器，当检测到移位寄存器 (ICDRS) 为空时，就将写到 ICDRT 的发送数据传送到 ICDRS，开始数据发送。在发送 ICDRS 的数据时，如果预先将下一个发送数据写到 ICDRT，就能进行连续发送。如果将 ICMR 的 MLS 位设定为 1 并且在将发送数据写到 ICDRT 后读取 ICDRT 时，就读到 MSB/LSB 顺序相反的数据。ICDRT 的初始值是 H'FF。



### 18.3.8 I<sup>2</sup>C 总线接收数据寄存器 (ICDRR)

ICDRR 是保存接收数据的 8 位寄存器。当 1 字节的数据接收结束时，就将接收数据从 ICDRS 传送到 ICDRR，并能接收下一个数据。因为 ICDRR 是接收专用寄存器，所以 CPU 不能写 ICDRR。ICDRR 的初始值是 H'FF。



### 18.3.9 I<sup>2</sup>C 总线移位寄存器 (ICDRS)

ICDRS 是用于发送或者接收数据的寄存器。在发送时，将发送数据从 ICDRT 传送到 ICDRS，然后从 SDA 引脚发送数据。在接收时，当 1 字节的数据接收结束时，就将数据从 ICDRS 传送到 ICDRR。CPU 不能直接读此寄存器。

位:	7	6	5	4	3	2	1	0
初始值:	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-

### 18.3.10 NF2CYC 寄存器 (NF2CYC)

NF2CYC 是 8 位可读写寄存器，能选择 SCL 引脚和 SDA 引脚的噪声消除范围。有关运行的详细内容请参照“18.4.7 噪声消除电路”。

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	NF2CYC
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

位	位名	初始值	R/W	说 明
7 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	NF2CYC	0	R/W	噪声消除范围的选择 0: 能通过外围时钟消除 1 个周期以内的噪声 1: 能通过外围时钟消除 2 个周期以内的噪声

## 18.4 运行说明

I<sup>2</sup>C 总线接口 2 能通过设定 SAR 的 FS，以 I<sup>2</sup>C 总线格式或者时钟同步串行格式进行通信。

### 18.4.1 I<sup>2</sup>C 总线格式

I<sup>2</sup>C 总线格式和 I<sup>2</sup>C 总线时序分别如图 18.3 和图 18.4 所示。开始条件后的第 1 帧一定是由 8 位构成。

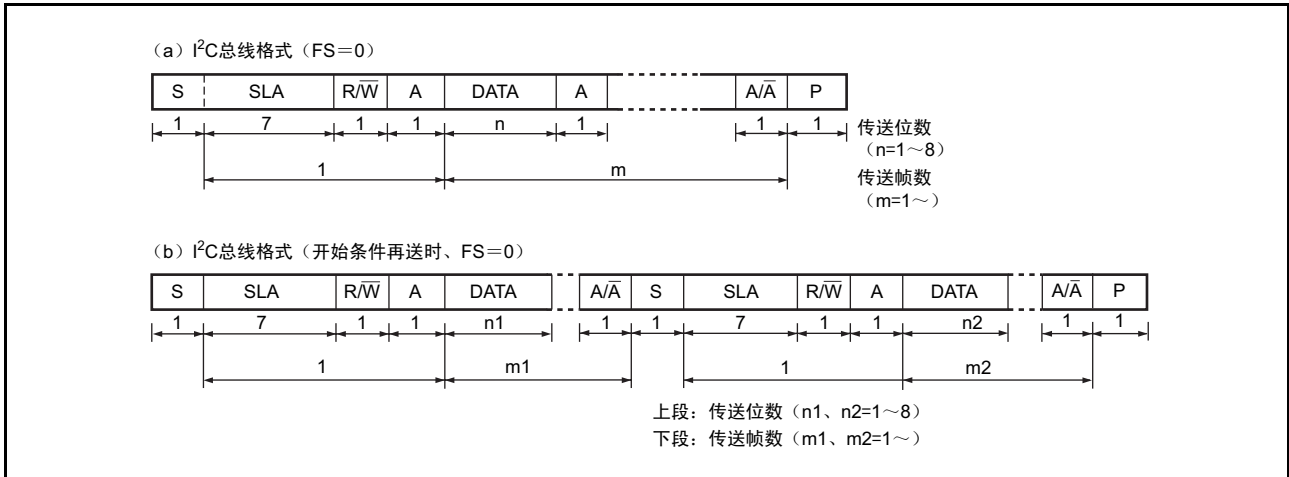


图 18.3 I<sup>2</sup>C 总线格式

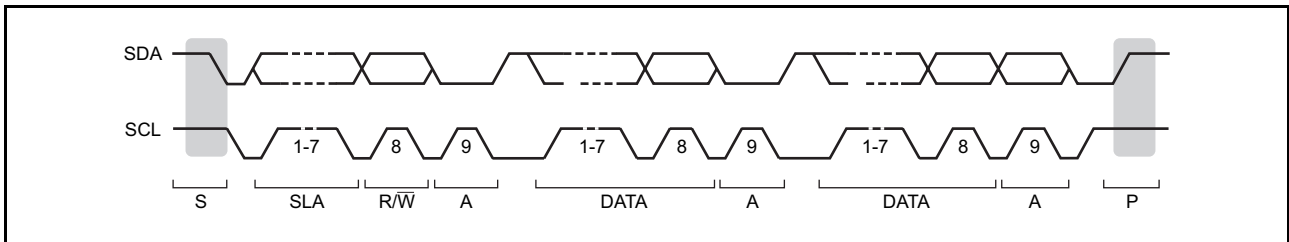


图 18.4 I<sup>2</sup>C 总线时序

#### 【符号说明】

- S : 开始条件。在 SCL=High 电平状态下，主器件将 SDA 从 High 电平变为 Low 电平。
- SLA : 从属地址
- $\overline{R/W}$  : 表示发送和接收的方向。当  $\overline{R/W}$  是 1 时，将数据从从属器件发送到主器件；当  $\overline{R/W}$  是 0 时，将数据从主器件发送到从属器件。
- A : 应答。接收器件将 SDA 置为 Low 电平。
- DATA : 发送和接收的数据
- P : 停止条件。在 SCL=High 电平状态下，主器件将 SDA 从 Low 电平变为 High 电平。

### 18.4.2 主发送

在主发送模式中，主器件输出发送时钟和发送数据，从属器件返回应答。有关主发送模式的运行时序，请参照图 18.5 和图 18.6。

主发送模式的发送步骤和运行如下所示：

1. 将 ICCR1 的 ICE 位设定为 1，然后设定 ICMR 的 MLS 和 ICCR1 的 CKS3 ~ CKS0 等（初始设定）。
2. 读 ICCR2 的 BBSY 标志，在确认总线处于释放状态后将 ICCR1 的 MST 和 TRS 设定为主发送模式，然后写 BBSY=1 和 SCP=0（产生开始条件），生成开始条件。
3. 在确认 ICSR 的 TDRE 被置位后将发送数据写到 ICDRT（第 1 个字节表示从属地址和  $\overline{R/W}$  的数据）。此时，TDRE 自动清 0，在将数据从 ICDRT 传送到 ICDRS 后重新将 TDRE 置位。
4. 在 TDRE 被置位的状态下结束 1 字节的数据发送，并在发送时钟的第 9 个时钟的上升沿将 ICSR 的 TEND 置位。读 ICIER 的 ACKBR，在确认已选择从属器件后将第 2 个字节的数据写到 ICDRT。当 ACKBR 是 1 时，从属器件未被识别，所以产生停止条件。通过写 BBSY=0 和 SCP=0 产生停止条件。另外，在数据准备完成前或者在产生停止条件前，将 SCL 固定为 Low 电平。
5. 每当 TDRE 被置位时，将第 2 个字节以后的发送数据写到 ICDRT。
6. 如果将要发送的字节数写到 ICDRT，就在 TDRE 被置位的状态下等待 TEND 的置位（最后字节的发送结束），或者在 ICIER 的 ACKE 被置位的状态下等待接收器件的 NACK（ICSR 的 NACKF=1）。然后产生停止条件，清除 TEND 或者 NACKF。
7. 当 ICSR 的 STOP 被置位时，返回到从属接收模式。

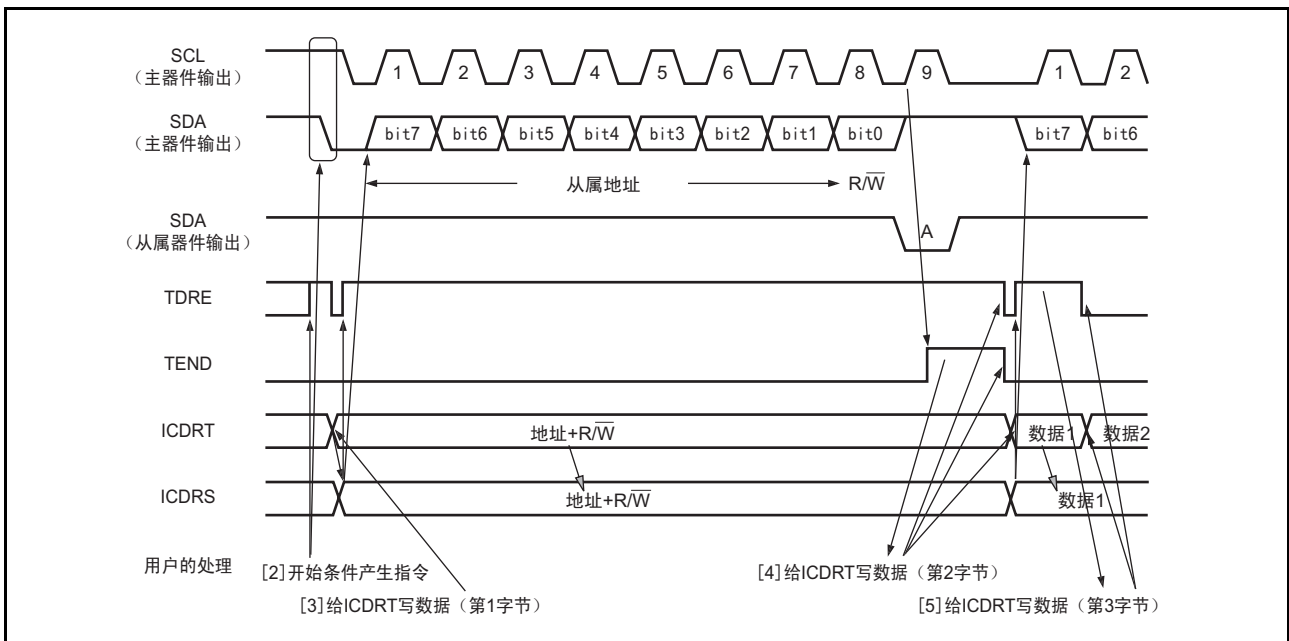


图 18.5 主发送模式的运行时序 (1)

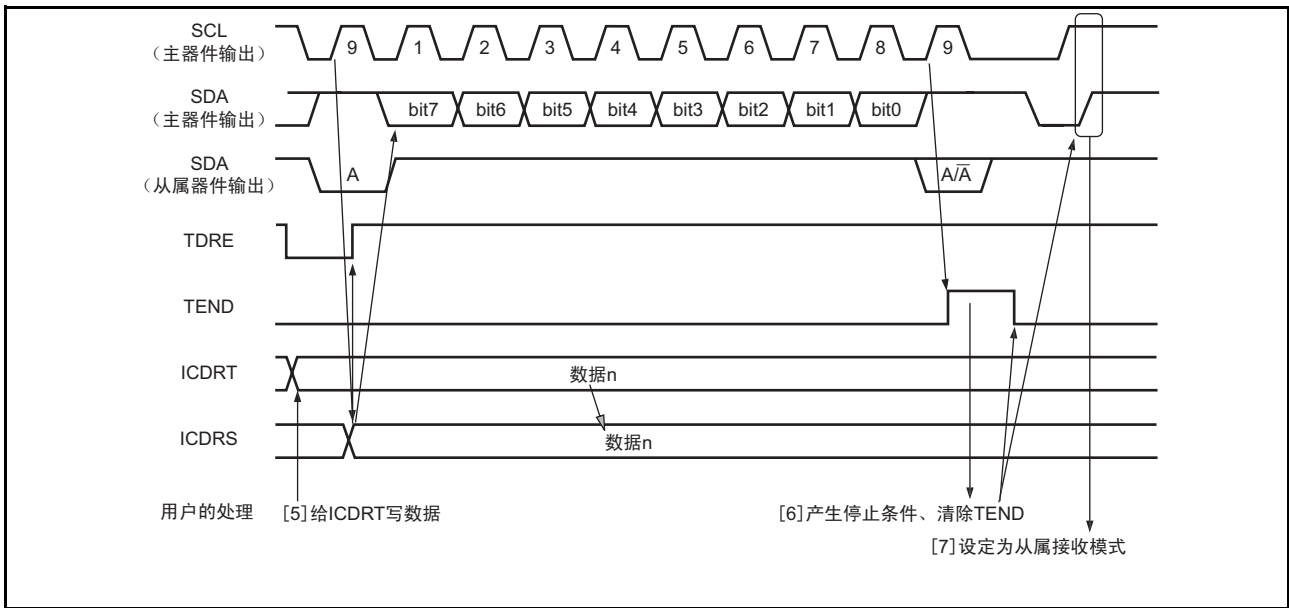


图 18.6 主发送模式的运行时序 (2)

### 18.4.3 主接收

在主接收模式中，主器件输出接收时钟，从属器件接收数据并返回应答。有关主接收模式的运行时序，请参照图 18.7 和图 18.8。

主接收模式的接收步骤和运行如下所示：

1. 在清除 ICSR 的 TEND 后，清除 ICCR1 的 TRS，从主发送模式转换为主接收模式，然后清除 TDRE。
2. 当虚读 ICDRR 时，开始接收 \*，与内部时钟同步输出接收时钟并接收数据。在接收时钟的第 9 个时钟，主器件将 ICIER 的 ACKBT 设定的电平输出到 SDA。
3. 结束 1 帧数据的接收，并在接收时钟的第 9 个时钟的上升沿将 ICSR 的 RDRF 置位。此时，能通过读 ICDRR 来读取所接收的数据，同时清除 RDRF。
4. 如果在每次 RDRF 被置位时读 ICDRR，就能进行连续接收。如果在 RDRF 置位的状态下因其他处理而推迟 ICDRR 的读操作并且第 8 个时钟下降，就在读 ICDRR 前 SCL 固定为 Low 电平。
5. 在下一次接收为最后 1 帧的情况下，就要在读 ICDRR 前将 ICCR1 的 RCVD 置位，从而能在下一次接收后处于可产生停止条件的状态。
6. 如果在接收时钟的第 9 个时钟的上升沿将 RDRF 置位，就产生停止条件。
7. 如果 ICSR 的 STOP 被置位，就读 ICDRR，然后清除 RCVD。
8. 返回到从属接收模式。

**【注】** \* 如果只要接收 1 个字节，就必须在 ICCR1 的 RCVD 被置位后，虚读 ICDRR。

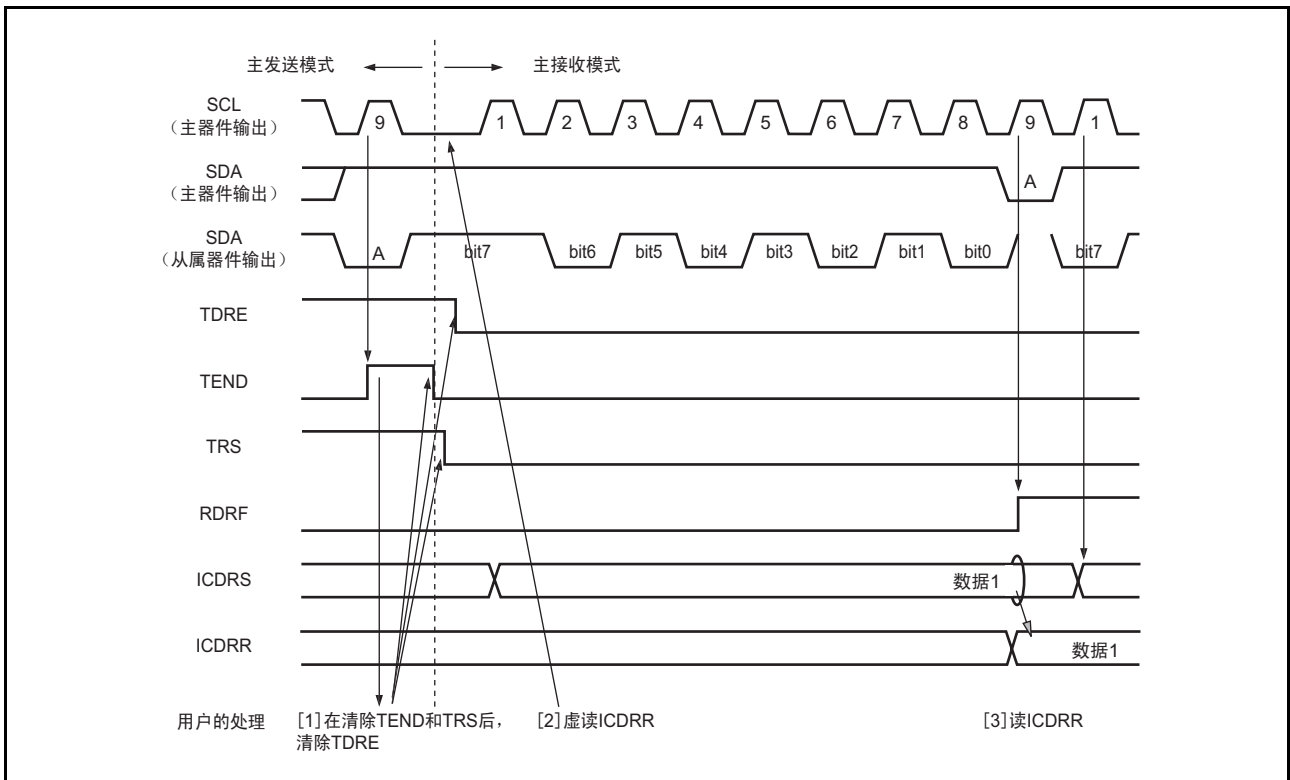


图 18.7 主接收模式的运行时序 (1)

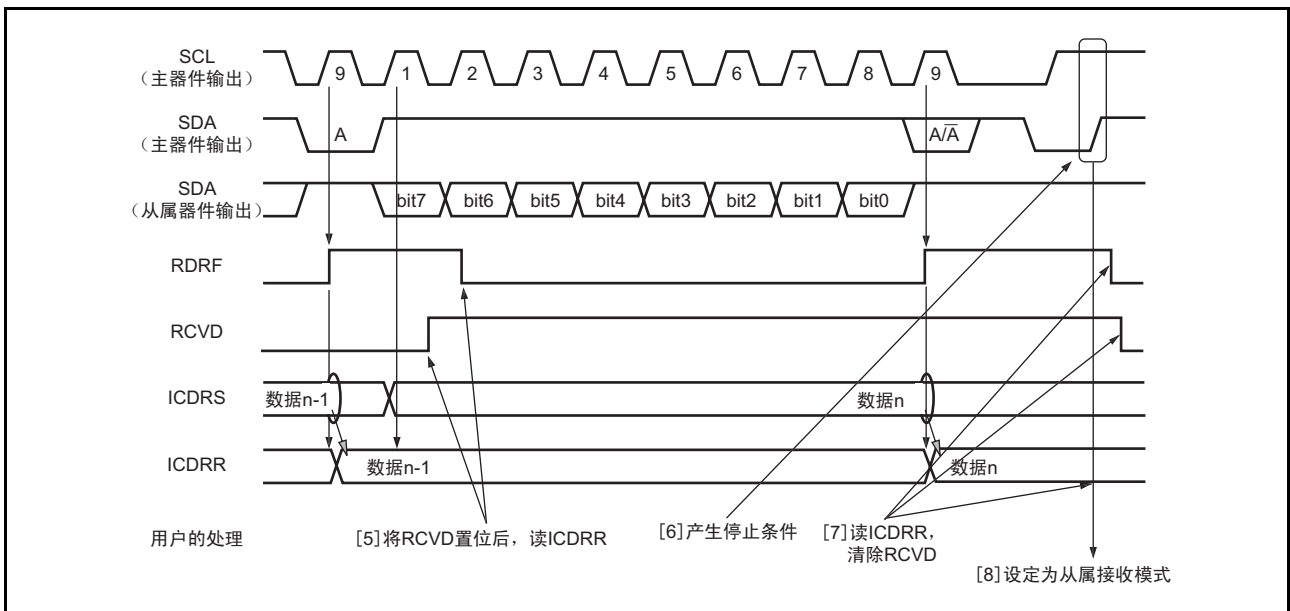


图 18.8 主接收模式的运行时序 (2)

### 18.4.4 从属发送

在从属发送模式中，从属器件输出发送数据，主器件输出接收时钟并返回应答。有关从属发送模式的运行时序，请参照图 18.9 和图 18.10。

从属发送模式的发送步骤和运行如下所示：

1. 将 ICCR1 的 ICE 位设定为 1，然后设定 ICMR 的 MLS 和 ICCR1 的 CKS3 ~ CKS0 等（初始设定）。将 ICCR1 的 MST 和 TRS 设定为从属接收模式，等待从属地址的一致。
2. 如果在检测到开始条件后第 1 帧的从属地址一致，从属器件就在第 9 个时钟的上升沿将 ICIER 的 ACKBT 设定的电平输出到 SDA。此时，当第 8 位数据（R/W）是 1 时，ICCR1 的 TRS 和 ICSR 的 TDRE 就被置位，自动转换为从属发送模式。如果在每次 TDRE 被置位时将发送数据写到 ICDRT，就能进行连续发送。
3. 如果在最后的发送数据写到 ICDRT 后将 TDRE 置位，就在 TDRE=1 的状态下等待 ICSR 的 TEND 的置位。如果 TEND 被置位，就清除 TEND。
4. 为了结束处理，清除 TRS 并虚读 ICDRR，从而能释放 SCL。
5. 清除 TDRE。

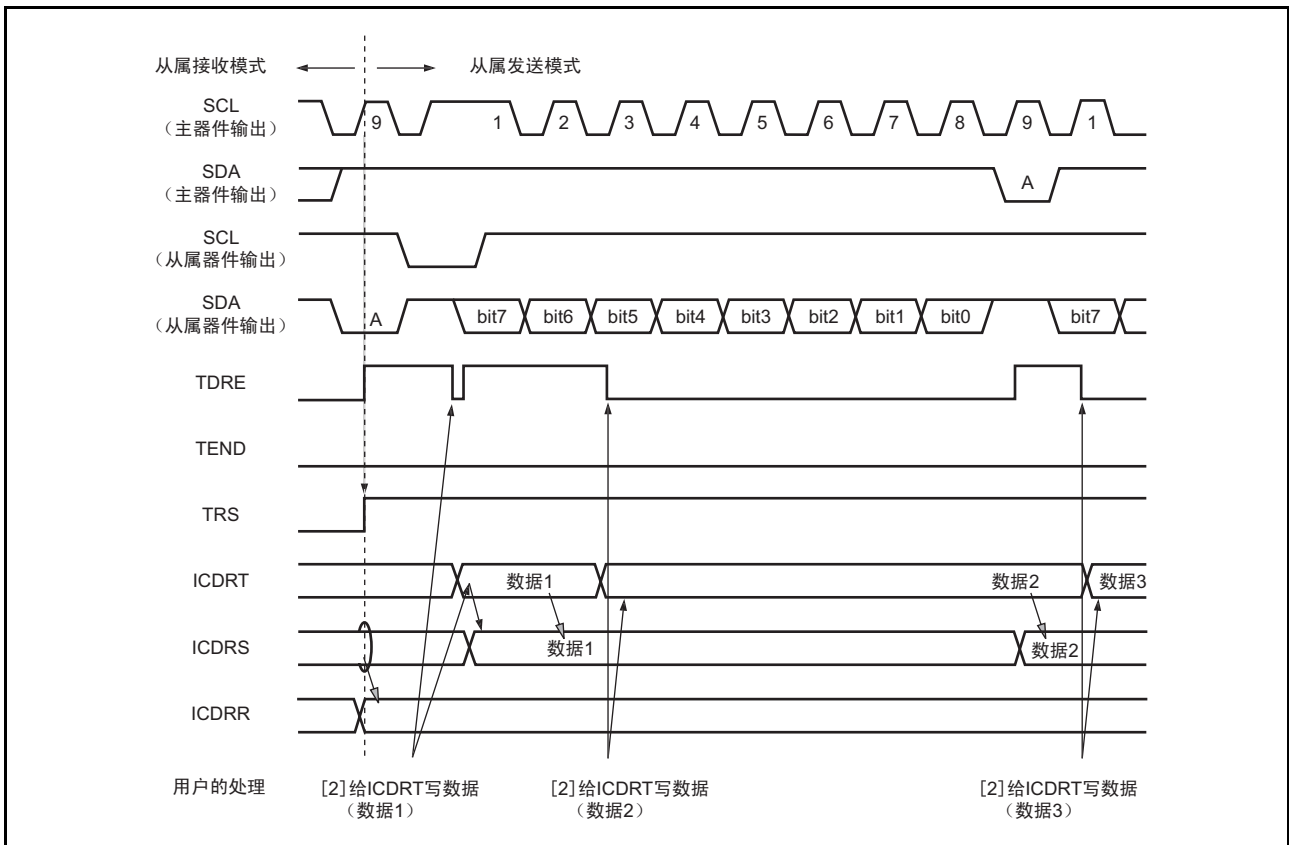


图 18.9 从属发送模式的运行时序 (1)

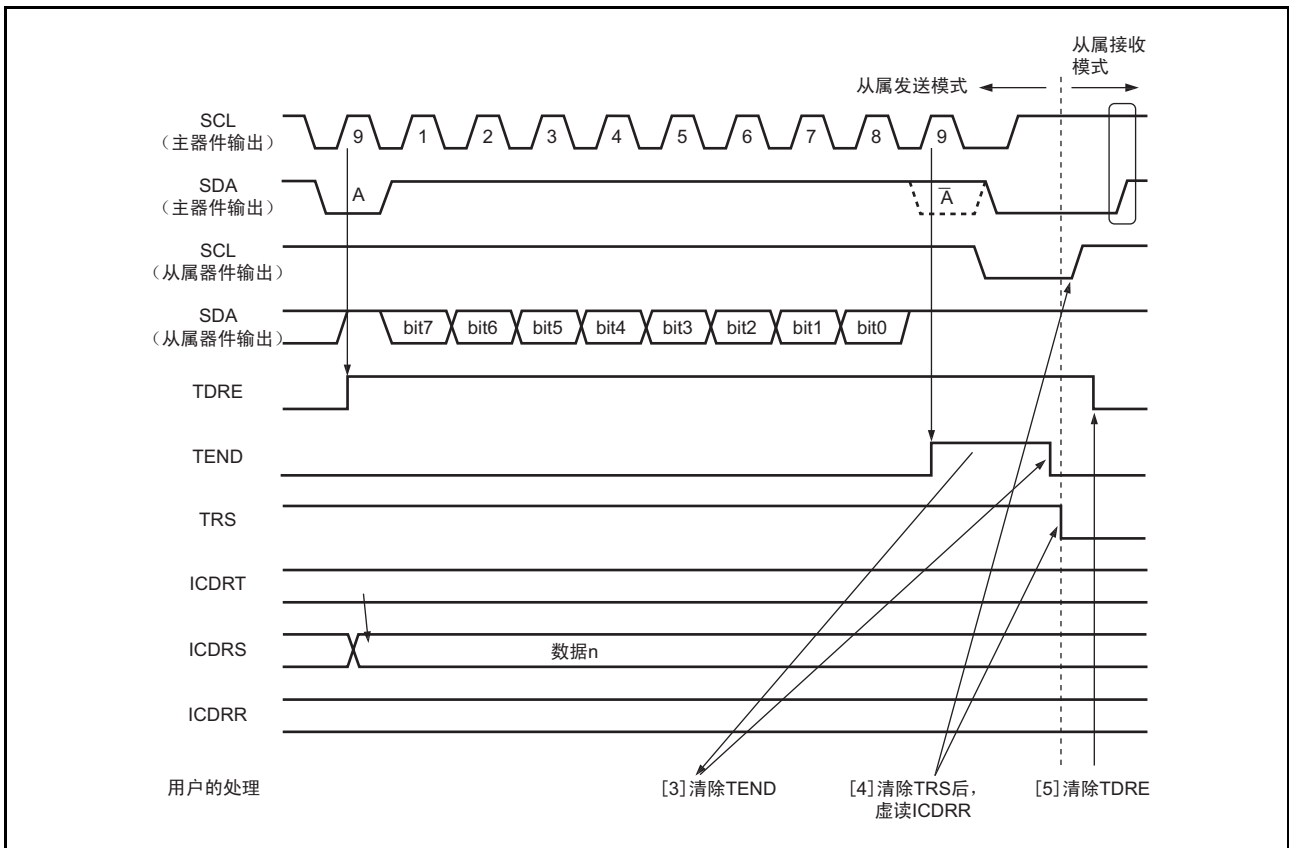


图 18.10 从属发送模式的运行时序 (2)

### 18.4.5 从属接收

在从属接收模式中，主器件输出发送时钟和发送数据，从属器件返回应答。有关从属接收模式的运行时序，请参照图 18.11 和图 18.12。

从属接收模式的接收步骤和运行如下所示：

1. 将 ICCR1 的 ICE 位设定为 1，然后设定 ICMR 的 MLS 和 ICCR1 的 CKS3 ~ CKS0 等（初始设定）。将 ICCR1 的 MST 和 TRS 设定为从属接收模式，等待从属地址的一致。
2. 如果在检测到开始条件后第 1 帧的从属地址一致，从属器件就在第 9 个时钟的上升沿将 ICIER 的 ACKBT 设定的电平输出到 SDA。因为 ICSR 的 RDRF 同时被置位，所以虚读 ICDRR（读出的数据表示从属地址 +R/W，所以不要）。
3. 每当 RDRF 被置位时，虚读 ICDRR。如果在 RDRF 被置位的状态下第 8 个时钟下降，就在读 ICDRR 前 SCL 固定为 Low 电平。读 ICDRR 前所更改的应答设定（返回给主器件）反映在下一个传送帧。
4. 同样，通过读 ICDRR 来读取最后的字节。



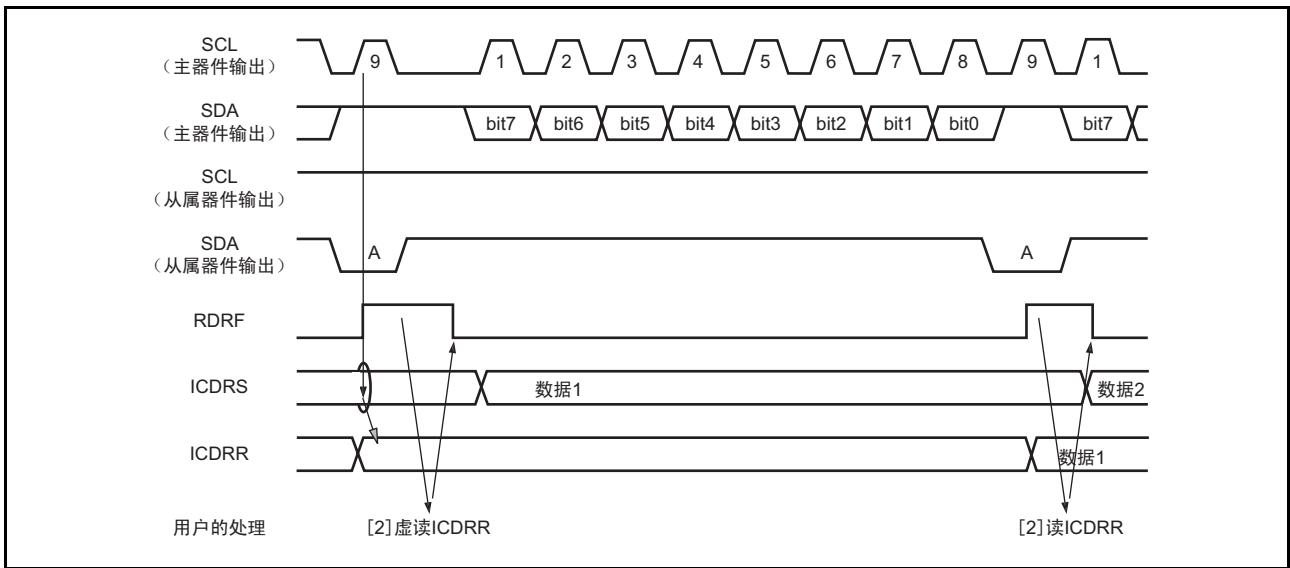


图 18.11 从属接收模式的运行时序 (1)

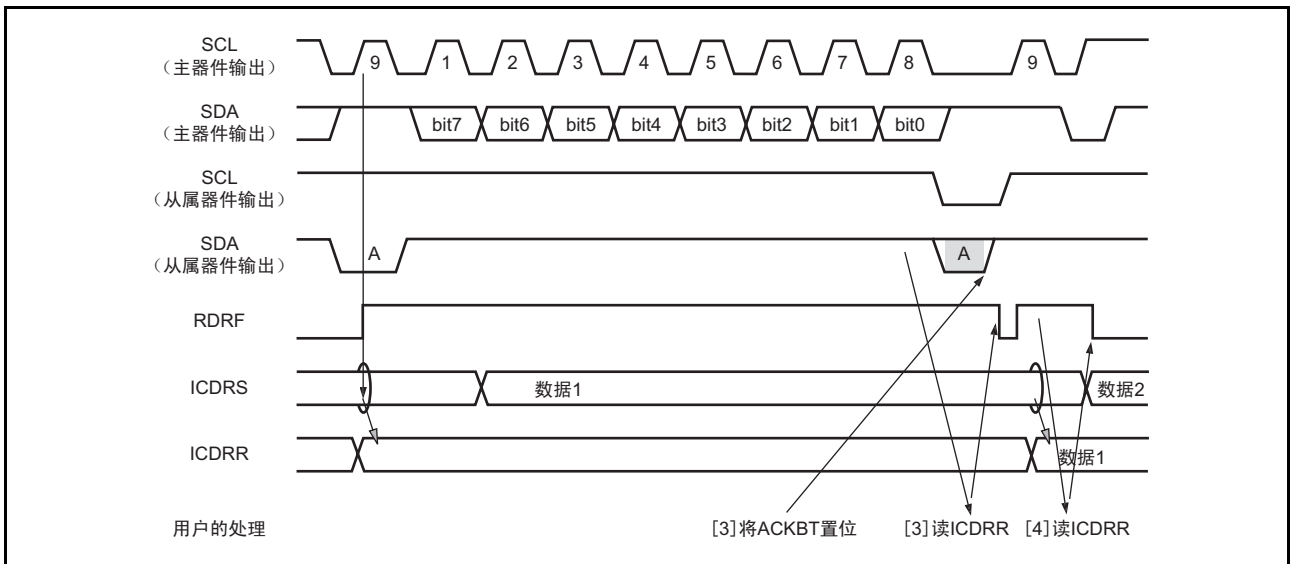


图 18.12 从属接收模式的运行时序 (2)

### 18.4.6 时钟同步串行格式

此模块能通过将 SAR 的 FS 设定为 1，以时钟同步串行格式进行通信。当 ICCR1 的 MST=1 时，从 SCL 输出传送时钟；当 MST=0 时，输入外部时钟。

#### (1) 数据的传送格式

时钟同步串行格式的传送格式如图 18.13 所示。

在 SCL 时钟的下降沿和下一个上升沿之间输出传送数据，并确保 SCL 时钟上升沿的数据。能通过 ICMR 的 MLS 选择 MSB 先或者 LSB 先的数据传送顺序，还能通过 ICCR2 的 SDAO，在传送待机时更改 SDA 的输出电平。

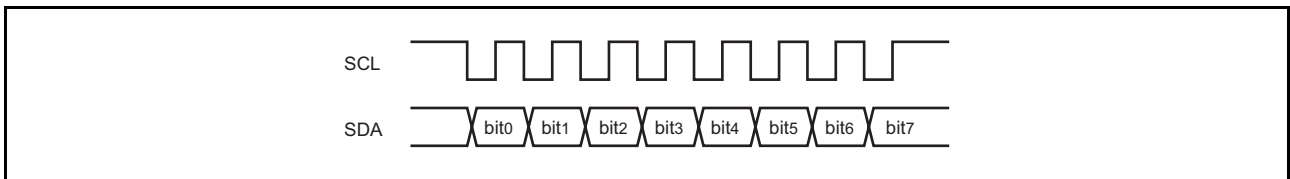


图 18.13 时钟同步串行格式的传送格式

#### (2) 发送

在发送模式中，与传送时钟的下降沿同步从 SDA 输出发送数据。当 ICCR1 的 MST=1 时，输出传送时钟；当 MST=0 时，输入传送时钟。有关发送模式的运行时序请参照图 18.14。

发送模式的步骤和运行如下所示：

1. 将 ICCR1 的 ICE 位设定为 1，然后设定 ICCR1 的 MST 和 CKS3 ~ CKS0 等（初始设定）。
2. 设定 ICCR1 的 TRS，转换为发送模式，ICSR 的 TDRE 被置位。
3. 当确认 TDRE 被置位时，将发送数据写到 ICDRT，数据从 ICDRT 传送到 ICDRS 并且 TDRE 自动置位。如果在每次 TDRE 被置位时将数据写到 ICDRT，就能进行连续发送。要从发送模式转换为接收模式时，必须在 TDRE 被置位的状态下清除 TRS。

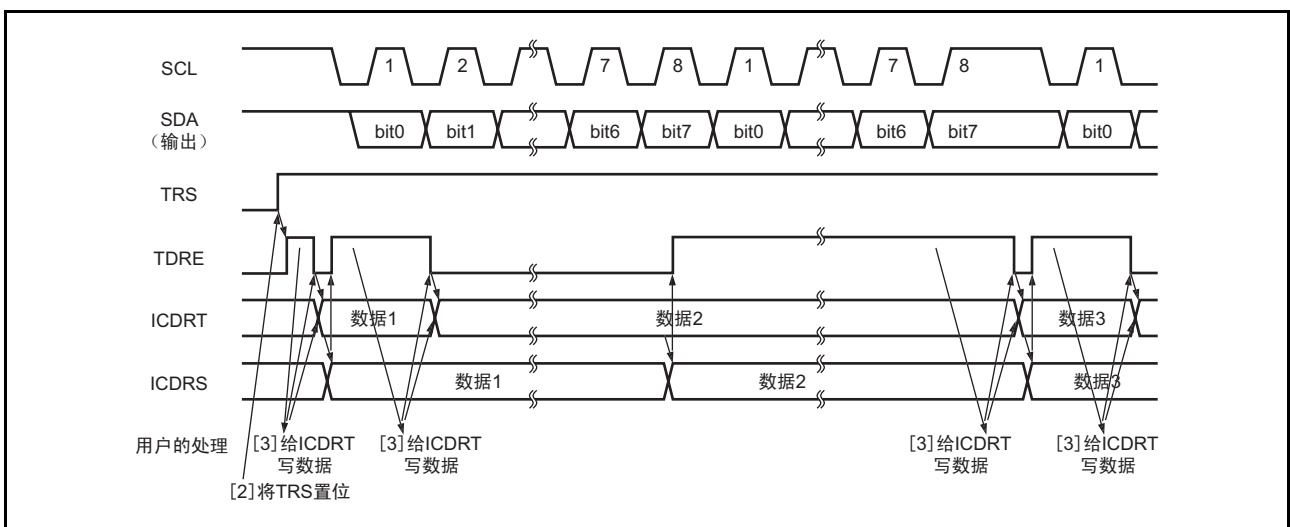


图 18.14 发送模式的运行时序

## (3) 接收

在接收模式中，在传送时钟的上升沿锁存数据。当 ICCR1 的 MST=1 时，输出传送时钟；当 MST=0 时，输入传送时钟。有关接收模式的运行时序请参照图 18.15。

接收模式的步骤和运行如下所示：

1. 将 ICCR1 的 ICE 位设定为 1，然后设定 ICCR1 的 MST 和 CKS3 ~ CKS0 等（初始设定）。
2. 在输出传送时钟时，设定 MST=1，开始输出接收时钟。
3. 当接收结束时，数据从 ICDRS 传送到 ICDRR，并且 ICSR 的 RDRF 被置位。当 MST=1 时，因为为可接收下一个字节的状态，所以能连续输出时钟。如果在每次 RDRF 被置位时读 ICDRR，就能进行连续接收。如果在 RDRF 被置位的状态下第 8 个时钟上升，就检测到溢出，将 ICSR 的 AL/OVE 置位。此时 ICDRR 保存前一个接收数据。
4. 当 MST=1 时，为了停止接收，先将 ICCR1 的 RCVD 置位，然后读 ICDRR。于是在下一个字节接收结束后 SCL 固定为 High 电平。

**【注】** 当 MST=1 并且只要接收 1 个字节时，必须按照以下步骤进行，运行时序请参照图 18.16。

1. 将 ICCR1 的 ICE 位设定为 1，然后设定 ICCR1 的 CKS3 ~ CKS0 等（初始设定）。
2. 在 ICCR1 的 RCVD 位是 0 的状态下，设定 MST=1，开始输出接收时钟。
3. 必须在确认 ICMR 的 BC2 位已被置 1 后设定 ICCR1 的 RCVD=1。于是在输出 1 字节的接收时钟后 SCL 固定为 High 电平。

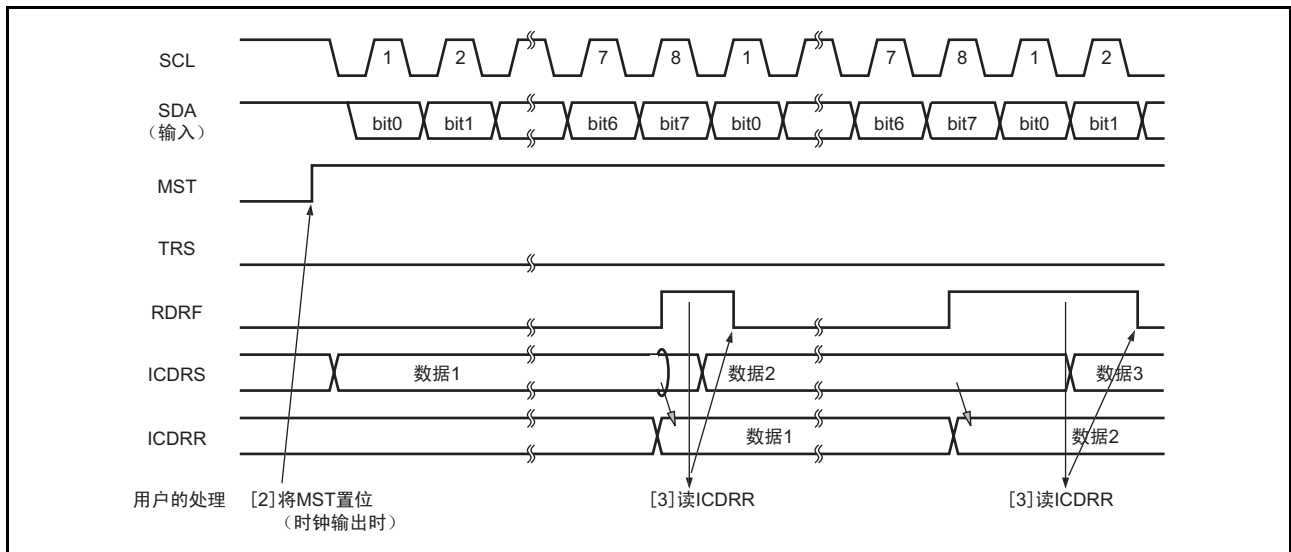


图 18.15 接收模式的运行时序

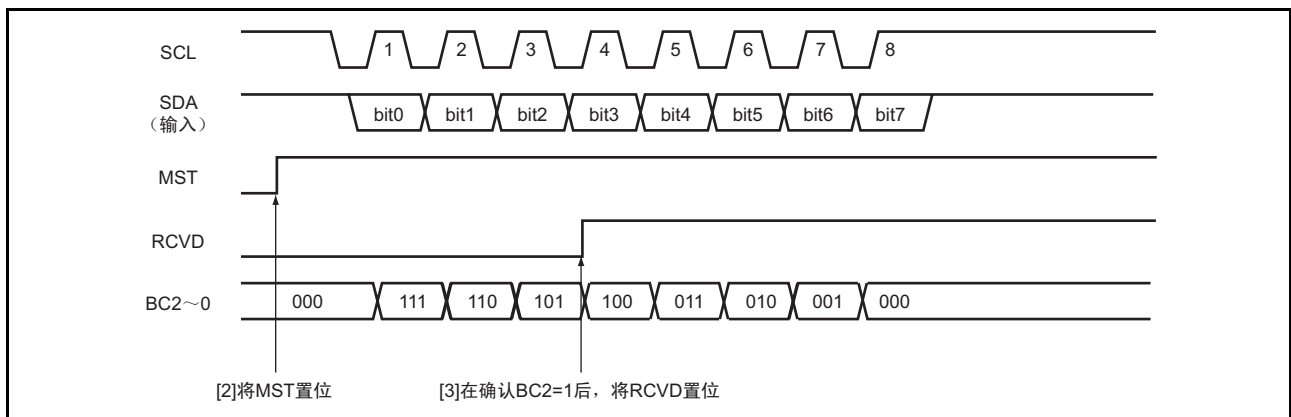


图 18.16 接收 1 字节数据的运行时序

### 18.4.7 噪声消除电路

SCL 引脚状态和 SDA 引脚状态经过噪声消除电路被取到内部噪声消除电路的框图如图 18.17 所示。

噪声消除电路由 3 段串联的锁存电路和一致检测电路构成。通过外围时钟对 SCL 引脚输入信号（或者 SDA 引脚输入信号）进行采样，当 NF2CYC 寄存器的值是“0”并且 2 个锁存器的输出信号一致时或者当 NF2CYC 寄存器的值是“1”并且 3 个锁存器的输出信号一致时，就将电平传送到后面的段。如果锁存器的输出信号不一致，就保持以前的值。

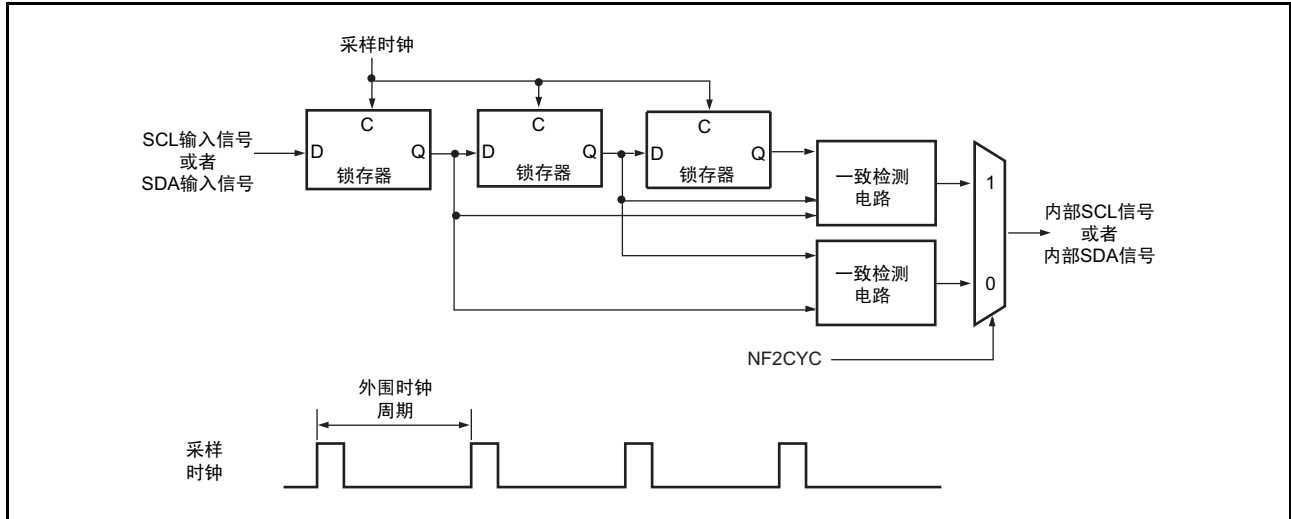


图 18.17 噪声消除电路的框图

18.4.8 使用例

在使用 I<sup>2</sup>C 总线接口 2 时，各模式的流程图例子如图 18.18 ~ 图 18.21 所示。

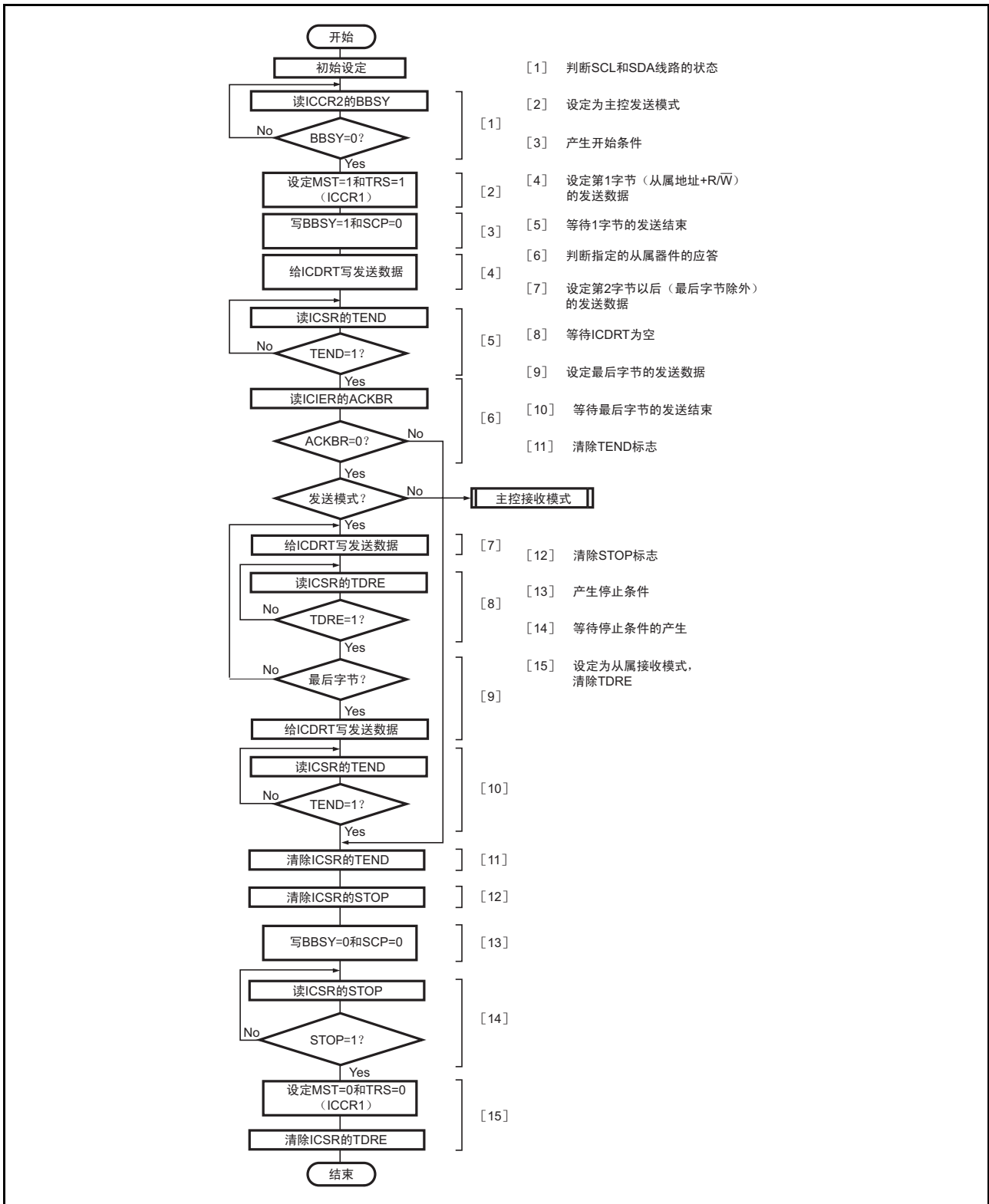


图 18.18 主发送模式的流程图例子

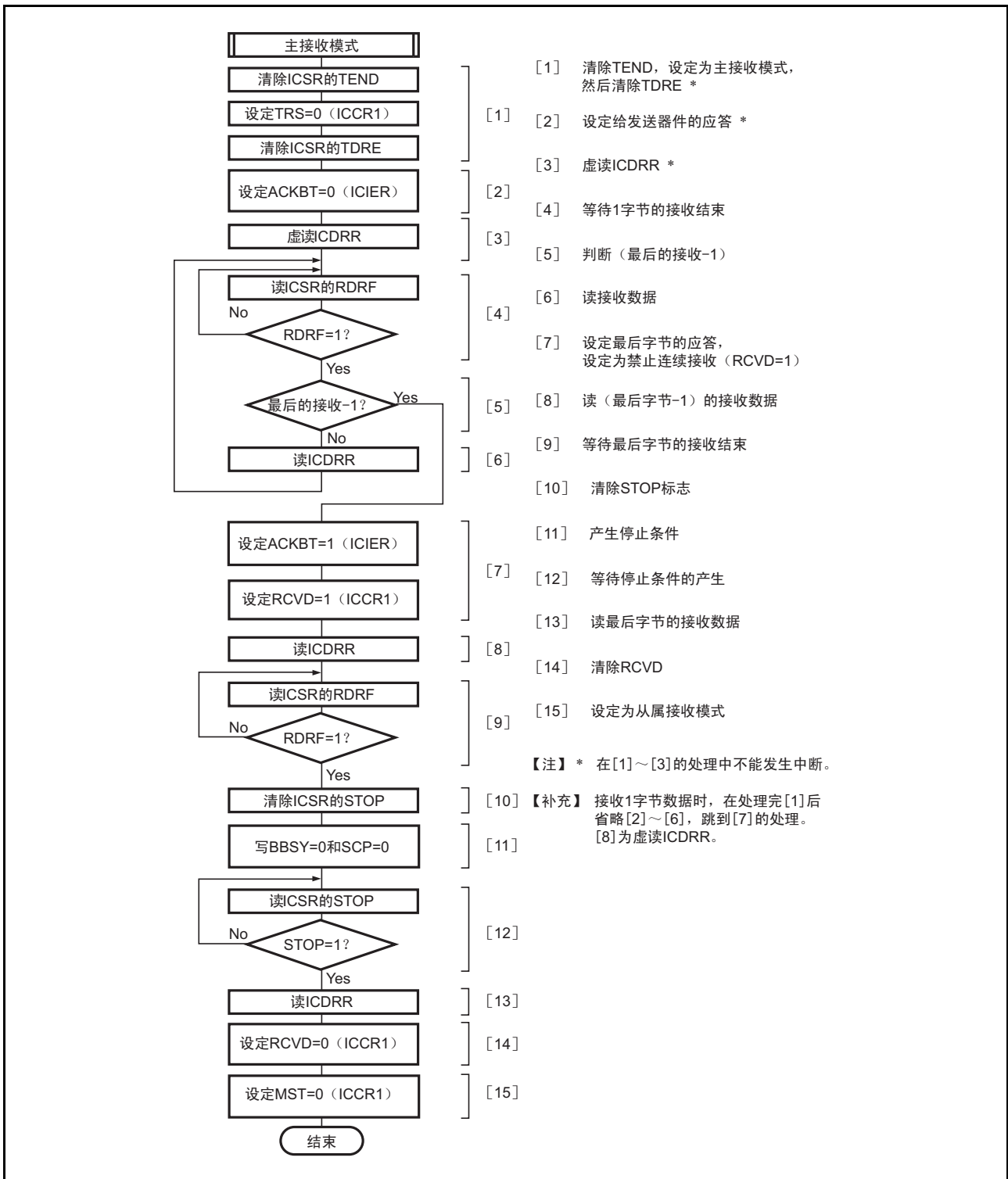


图 18.19 主接收模式的流程图例子

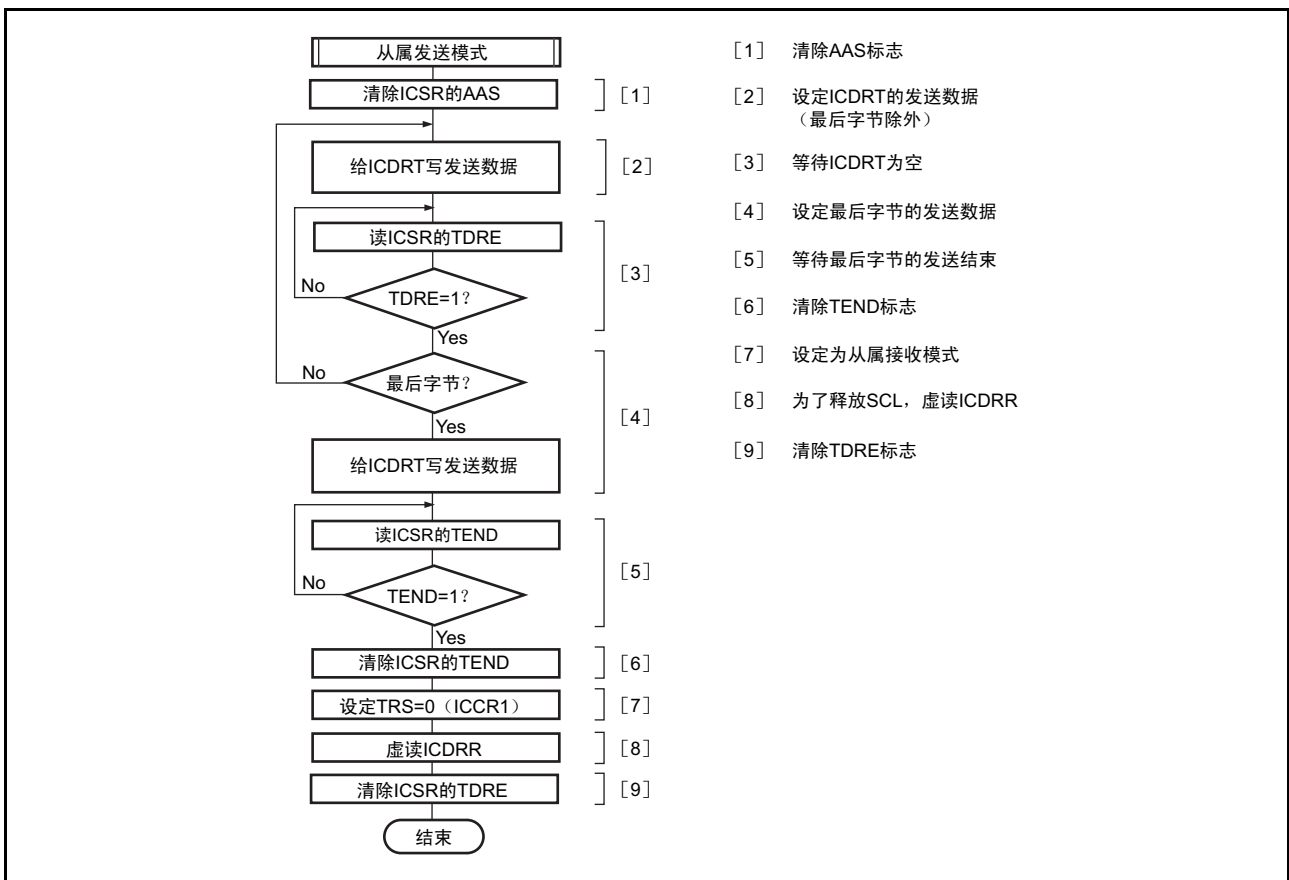


图 18.20 从属发送模式的流程图例子

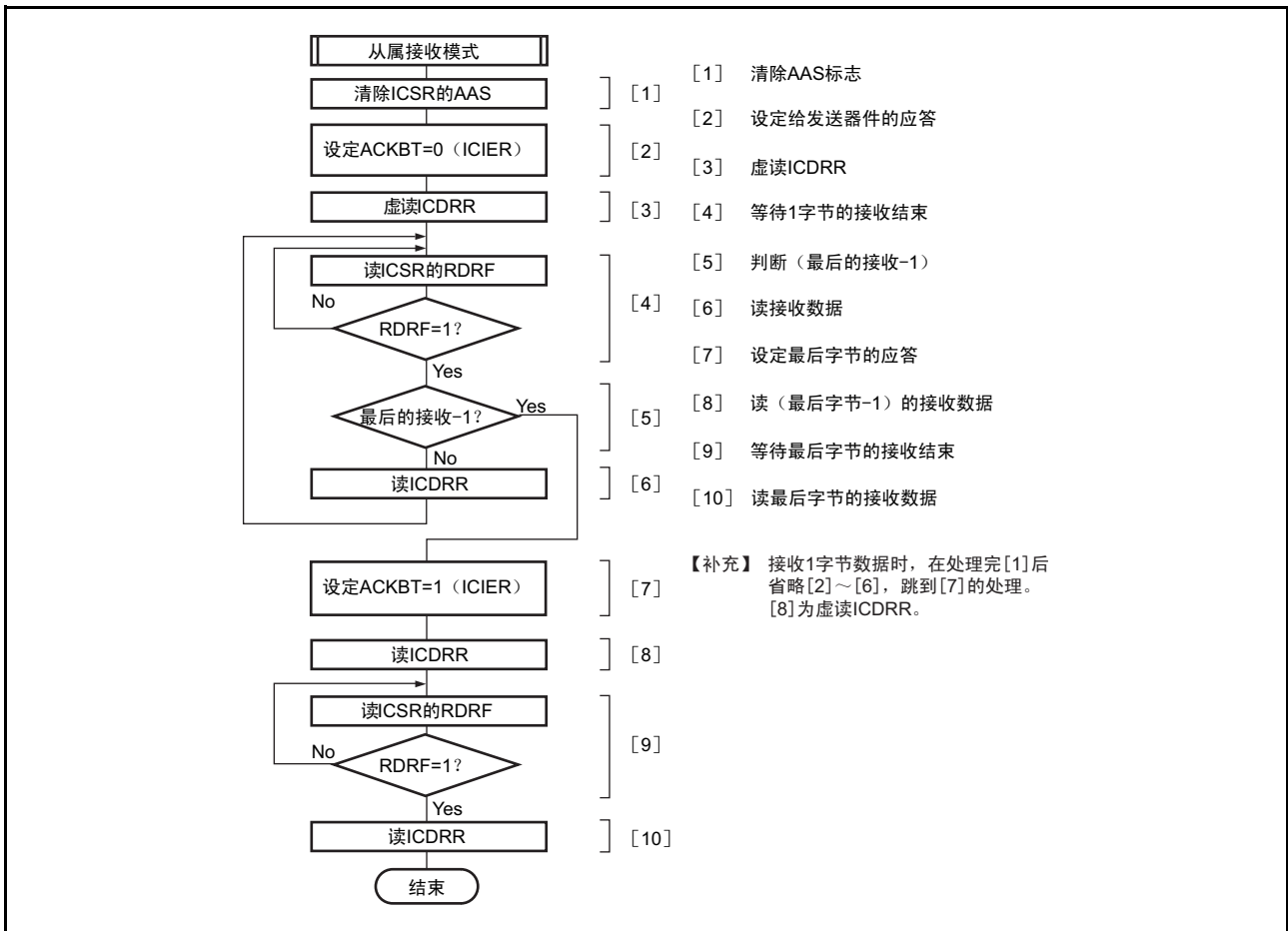


图 18.21 从属接收模式的流程图例子



## 18.5 I<sup>2</sup>C2 的中断源

此模块的中断请求有发送数据空、发送结束、接收数据满、NACK 的检测、停止条件的检测和仲裁失败 / 溢出错误中断，各中断请求的内容如表 18.4 所示。

表 18.4 中断请求一览表

中断请求	略称	中断条件	I <sup>2</sup> C 总线格式	时钟同步串行格式	DTC 的启动
发送数据空	IITXI	(TDRE=1) · (TIE=1)	○	○	○
发送结束	IITEI	(TEND=1) · (TEIE=1)	○	○	×
接收数据满	IIRXI	(RDRF=1) · (RIE=1)	○	○	○
停止条件的检测	IISTPI	(STOP=1) · (STIE=1)	○	×	×
NACK 的检测	IINAKI	{(NACKF=1)+(AL=1)} · (NAKIE=1)	○	×	×
仲裁失败 / 溢出错误			○	○	×

当表 18.4 的中断条件是 1 时，CPU 就执行中断异常处理。在异常处理中，必须清除各自的中断源。但是必须注意：通过将发送数据写到 ICDRT，自动清除 TDRE 和 TEND；通过读 ICDRT，自动清除 TDRF。尤其在将发送数据写到 ICDRT 的同时 TDRE 再次被置位，此时如果清除 TDRE，就有可能多发送 1 个字节。另外，在通过 DTC 进行指定次数的传送时，TDRE、TEND 和 RDRF 自动清除，但是在传送结束时，TDRE、TEND 和 RDRF 不自动清除。

## 18.6 DTC 的运行

在 I<sup>2</sup>C 总线格式中，通过从属地址和  $\overline{R/W}$  位选择从属器件和传送方向，通过应答位进行接收的确认和最后帧的显示，所以使用 DTC 进行的数据连续传送需要通过中断结合 CPU 处理进行。

使用 DTC 的处理例子如表 18.5 所示，假设在从属模式中知道传送数据的个数。

表 18.5 使用 DTC 的运行例

项目	主发送模式	主接收模式	从属发送模式	从属接收模式
从属地址 + $\overline{R/W}$ 位的发送 / 接收	通过 DTC 发送 (写 ICDR)	通过 CPU 发送 (写 ICDR)	通过 CPU 接收 (读 ICDR)	通过 CPU 接收 (读 ICDR)
读虚拟数据	—	通过 CPU 处理 (读 ICDR)	—	—
数据的发送 / 接收	通过 DTC 发送 (写 ICDR)	通过 DTC 接收 (读 ICDR)	通过 DTC 发送 (写 ICDR)	通过 DTC 接收 (读 ICDR)
最后帧的处理	不需要	通过 CPU 接收 (读 ICDR)	不需要	通过 CPU 接收 (读 ICDR)
设定 DTC 传送数据的帧数	发送：实际的数据个数 +1 (+1 是指从属地址 + $\overline{R/W}$ 位数)	接收：实际的数据个数	发送：实际的数据个数	接收：实际的数据个数

### 18.7 位同步电路

在设定主模式时，此模块在以下 2 种状态下，有可能缩短 High 电平期间，所以需要监视 SCL，按位取得同步进行通信。

- 当 SCL 被从属器件拉到 Low 电平时
- 因 SCL 线的负载（负载电容或者上拉电阻）而使 SCL 的上升变得缓慢时

位同步电路的时序如图 18.22 所示，SCL 从 Low 电平输出变为 Hi-Z 到监视 SCL 为止的时间如表 18.6 所示。

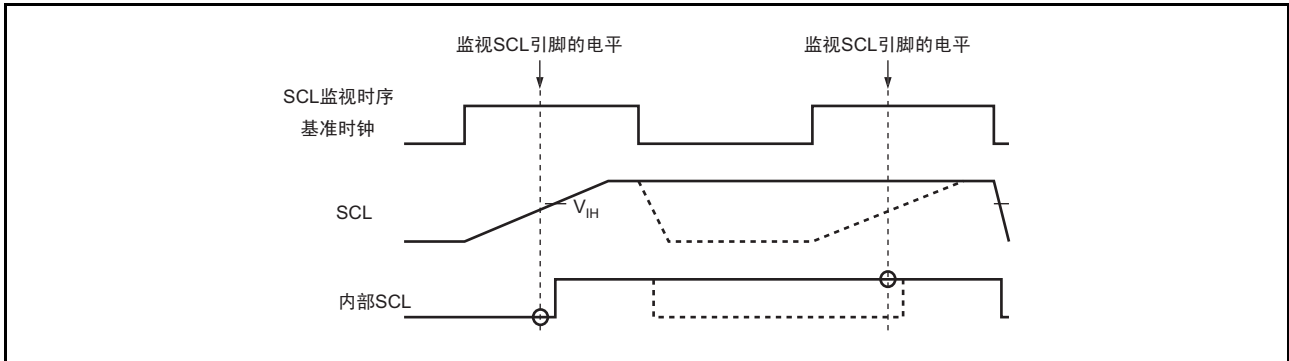


图 18.22 位同步电路的时序

表 18.6 SCL 的监视时间

CKS3	CKS2	NF2CYC	SCL 的监视时间 *1
0	0	0	6.5 t <sub>pcyc</sub> *2
		1	5.5 t <sub>pcyc</sub> *2
	1	0	18.5 t <sub>pcyc</sub> *2
		1	17.5 t <sub>pcyc</sub> *2
1	0	0	16.5 t <sub>pcyc</sub> *2
		1	15.5 t <sub>pcyc</sub> *2
	1	0	40.5 t <sub>pcyc</sub> *2
		1	39.5 t <sub>pcyc</sub> *2

【注】 \*1 监视从“SCL 监视时序的基准时钟”的上升沿到“监视 SCL 的时间”后的 SCL 引脚电平。

\*2 t<sub>pcyc</sub> 表示外围时钟的周期

## 18.8 使用注意事项

### 18.8.1 模块待机模式的设定

能通过待机控制寄存器，禁止或者允许 I<sup>2</sup>C2 模块的运行，初始值为停止 I<sup>2</sup>C2 的运行。通过解除模块待机模式，允许寄存器的存取，详细内容请参照“第 26 章 低功耗模式”。

### 18.8.2 停止条件和开始条件（重新发送）的产生

必须在识别第 9 个时钟的下降沿后产生停止条件和开始条件（重新发送）。能通过检查 I<sup>2</sup>C 总线控制寄存器 2 (ICCR2) 的 SCLO 位识别第 9 个时钟的下降沿。如果在下述 1. 或者 2. 的条件下并且在特定的时序产生停止条件和开始条件（重新发送），就有可能无法正常地输出停止条件和开始条件（重新发送），否则就没有问题。

1. 因 SCL 的总线负载（负载电容或者上拉电阻）而使 SCL 的上升时间长于“18.7 位同步电路”中规定的时间
2. 因从属器件拉长了第 8 个时钟和第 9 个时钟的 Low 电平时间而使位同步电路起作用时

### 18.8.3 开始条件和停止条件的连续产生

请不要让开始条件和停止条件连续产生。如果想连续产生开始条件和停止条件时，必须在发送从属地址后再让停止条件产生。

### 18.8.4 关于作为多主使用时的设定

1. 关于传送速率的设定值  
作为多主使用时，必须设定大于等于其他主器件的最快传送速率的 1/1.8。例如，其他主器件的最快传送速率为 400kbps 时，该 LSI 的 I<sup>2</sup>C 的传送速率必须设定为大于等于 223kbps (=400/1.8)。
2. ICCR1 的 MST 位、TRS 位  
作为多主使用时，必须用 MOV 指令设定 ICCR1 的 MST 位和 TRS 位。
3. 发生仲裁失败时  
在发生仲裁失败时，必须确认 ICCR1 的 MST 位和 TRS 位是否清 0。如果 ICCR1 的 MST 位和 TRS 位是 0 以外时，必须清 0。

### 18.8.5 在主接收模式下读取 ICDRR

在主接收模式下，必须在 SCL 的第 8 个时钟的上升沿为止进行 ICDRR 的读取。如果在第 8 个时钟的上升时还来不及读取 ICDRR，在 ICSR 的 RDRF 位为 1 的状态下接收下一个数据时，第 8 个时钟被固定为 L 电平，第 9 个时钟被输出。

在 SCL 的第 8 个时钟的上升沿时还来不及进行 ICDRR 的读取时，将 ICRR1 的 RCVD 位置 1，按照每个字节进行通信。

## 第 19 章 A/D 转换器 (ADC)

本 LSI 内置逐次逼近方式的 10 位 A/D 转换器。

### 19.1 特点

- 分辨率：10 位
- 输入通道：  
SH7083/84/85 有 8 个通道（内置 2 个独立的 A/D 转换模块）  
SH7086 有 16 个通道（内置 3 个独立的 A/D 转换模块）
- 转换时间：平均 1 个通道 2.0 $\mu$ s（在 P $\phi$ =25MHz 运行时）
- 运行模式：3 种  
单通道模式：1 个通道的 A/D 转换  
连续扫描模式：SH7083/84/85 进行最多 4 个通道的重复 A/D 转换，SH7086 进行最多 8 个通道的重复 A/D 转换  
单周期扫描模式：SH7083/84/85 进行最多 4 个通道的连续 A/D 转换，SH7086 进行最多 8 个通道的连续 A/D 转换
- 数据寄存器：将 A/D 转换结果保存到对应各输入通道的 16 位数据寄存器
- 采样和保持功能
- A/D 转换的开始方法：3 种  
软件  
可选择多功能定时器脉冲单元 2（MTU2）或者多功能定时器脉冲单元 2S（MTU2S）的转换开始触发外部触发信号
- 中断源：产生 A/D 转换结束的中断请求（ADI）
- 能设定模块待机模式

A/D 转换器的框图如图 19.1 所示。

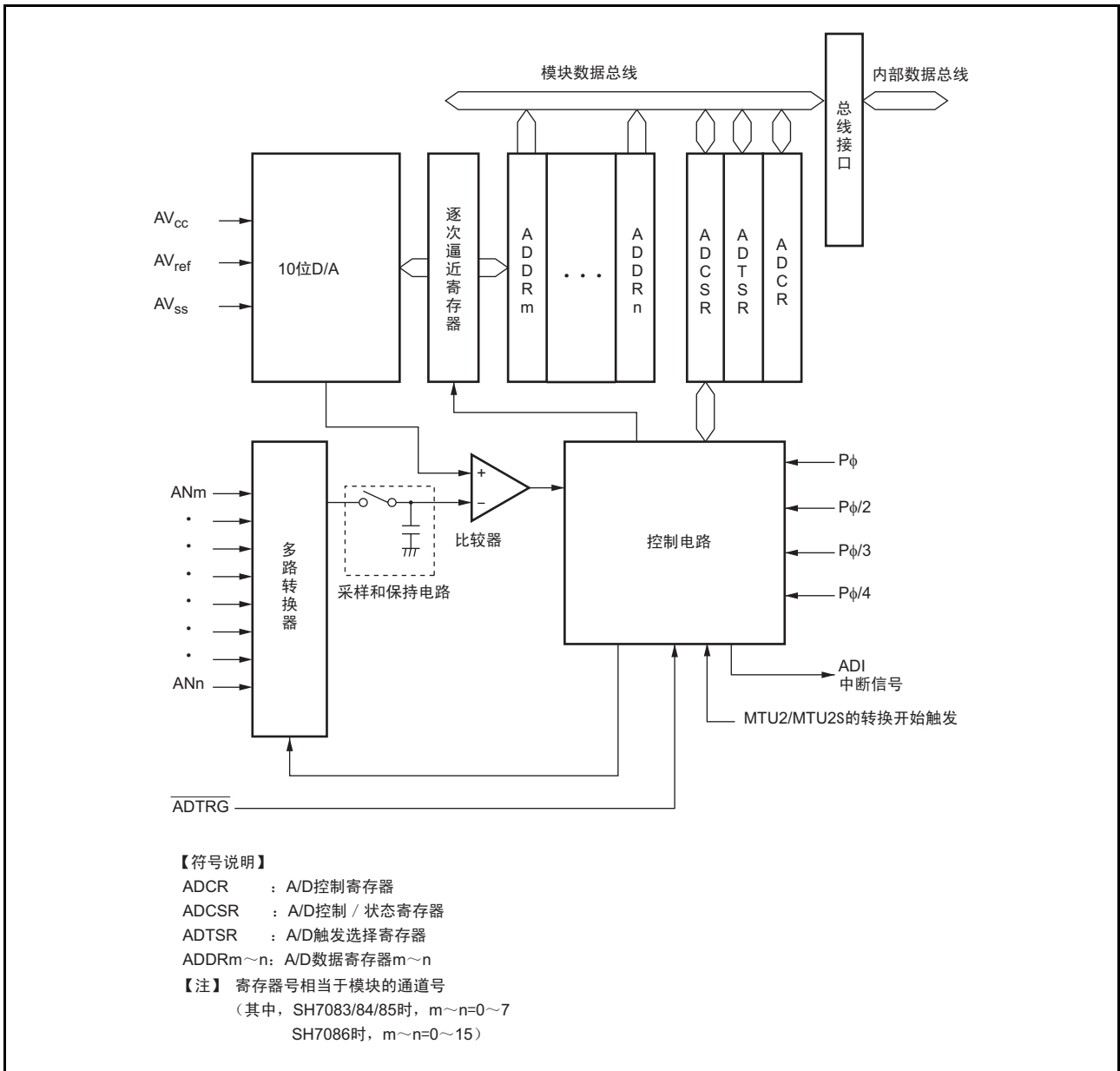


图 19.1 A/D 转换器的框图 (1 个模块)

## 19.2 输入 / 输出引脚

A/D 转换器使用的引脚如表 19.1 所示。SH7083/84/85 的 A/D 转换器由 2 个 A/D 转换模块构成，SH7086 的 A/D 转换器由 3 个 A/D 转换模块构成，A/D 模块 0、1 的输入通道分为 2 个通道组。

表 19.1 引脚结构

模块区分	引脚名	输入 / 输出	功 能		产品区分			
					SH7083	SH7084	SH7085	SH7086
通用	AVcc	输入	模拟部的电源引脚和基准电压		○	○	○	○
	AVref	输入	A/D 转换的基准电压		○	—	○	○
	AVss	输入	模拟部的接地和基准电压		○	○	○	○
	$\overline{\text{ADTRG}}$	输入	A/D 外部触发的输入引脚		○	○	○	○
A/D 模块 0 (A/D_0)	AN0	输入	模拟输入引脚 0	组 0	○	○	○	○
	AN1	输入	模拟输入引脚 1		○	○	○	○
	AN2	输入	模拟输入引脚 2	组 1	○	○	○	○
	AN3	输入	模拟输入引脚 3		○	○	○	○
A/D 模块 1 (A/D_1)	AN4	输入	模拟输入引脚 4	组 0	○	○	○	○
	AN5	输入	模拟输入引脚 5		○	○	○	○
	AN6	输入	模拟输入引脚 6	组 1	○	○	○	○
	AN7	输入	模拟输入引脚 7		○	○	○	○
A/D 模块 2 (A/D_2)	AN8	输入	模拟输入引脚 8		—	—	—	○
	AN9	输入	模拟输入引脚 9		—	—	—	○
	AN10	输入	模拟输入引脚 10		—	—	—	○
	AN11	输入	模拟输入引脚 11		—	—	—	○
	AN12	输入	模拟输入引脚 12		—	—	—	○
	AN13	输入	模拟输入引脚 13		—	—	—	○
	AN14	输入	模拟输入引脚 14		—	—	—	○
	AN15	输入	模拟输入引脚 15		—	—	—	○

【注】 连接各引脚的 A/D 模块不同。因为各模块都有控制寄存器，所以必须分别进行设定。

### 19.3 寄存器说明

A/D 转换器有以下寄存器。有关这些寄存器的地址和各处理模式中的寄存器状态，请参照“第 27 章 寄存器一览”。

表 19.2 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
A/D 数据寄存器 0	ADDR0	R	H'0000	H'FFFC900	16
A/D 数据寄存器 1	ADDR1	R	H'0000	H'FFFC902	16
A/D 数据寄存器 2	ADDR2	R	H'0000	H'FFFC904	16
A/D 数据寄存器 3	ADDR3	R	H'0000	H'FFFC906	16
A/D 控制 / 状态寄存器 _0	ADCSR_0	R/W	H'0000	H'FFFC910	16
A/D 控制寄存器 _0	ADCR_0	R/W	H'0000	H'FFFC912	16
A/D 数据寄存器 4	ADDR4	R	H'0000	H'FFFC980	16
A/D 数据寄存器 5	ADDR5	R	H'0000	H'FFFC982	16
A/D 数据寄存器 6	ADDR6	R	H'0000	H'FFFC984	16
A/D 数据寄存器 7	ADDR7	R	H'0000	H'FFFC986	16
A/D 控制 / 状态寄存器 _1	ADCSR_1	R/W	H'0000	H'FFFC990	16
A/D 控制寄存器 _1	ADCR_1	R/W	H'0000	H'FFFC992	16
A/D 数据寄存器 8	ADDR8	R	H'0000	H'FFFC A00	16
A/D 数据寄存器 9	ADDR9	R	H'0000	H'FFFC A02	16
A/D 数据寄存器 10	ADDR10	R	H'0000	H'FFFC A04	16
A/D 数据寄存器 11	ADDR11	R	H'0000	H'FFFC A06	16
A/D 数据寄存器 12	ADDR12	R	H'0000	H'FFFC A08	16
A/D 数据寄存器 13	ADDR13	R	H'0000	H'FFFC A0A	16
A/D 数据寄存器 14	ADDR14	R	H'0000	H'FFFC A0C	16
A/D 数据寄存器 15	ADDR15	R	H'0000	H'FFFC A0E	16
A/D 控制 / 状态寄存器 _2	ADCSR_2	R/W	H'0000	H'FFFC A10	16
A/D 控制寄存器 _2	ADCR_2	R/W	H'0000	H'FFFC A12	16
A/D 触发选择寄存器 _0	ADTSR_0	R/W	H'0000	H'FFFE890	8、16
A/D 触发选择寄存器 _1	ADTSR_1	R/W	H'0000	H'FFFE892	8、16

### 19.3.1 A/D 数据寄存器 0 ~ 15 (ADDR0 ~ ADDR15)

ADDR 是用于保存 A/D 转换结果的 16 位只读寄存器，各模拟输入通道的转换结果保存到对应序号的 ADDR（例如，AN4 的转换结果保存到 A/D 数据寄存器 ADDR4）。

10 位转换数据保存在 ADDR 的 bit15 ~ bit6，低 6 位的读出值总是 0。

ADDR 的初始值为 H'0000。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15 ~ 6		全 0	R	位的数据 (10 位)
5 ~ 0	—	全 0	R	保留位 读写的值总是 0。

### 19.3.2 A/D 控制 / 状态寄存器 \_0 ~ 2 (ADCSR\_0 ~ 2)

ADCSR 控制各模块的 A/D 转换运行。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADF	ADIE	-	-	TRGE	-	CONADF	STC	CKSL[1:0]	ADM[1:0]	ADCS	CH[2:0]				
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/W	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

【注】\* 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说明
15	ADF	0	R/(W)*	A/D 结束标志 表示 A/D 转换结束的状态。 [置位条件] • 在单通道模式中，当 A/D 转换结束时 • 在扫描模式中，当所选通道的 A/D 转换全部结束时 [清除条件] • 当读 1 的状态后写 0 时 • 通过 ADI 中断启动 DMAC/DTC 并读 ADDR 时
14	ADIE	0	R/W	A/D 中断 (ADI) 的允许 如果置 1，就允许由 ADF 产生的 ADI 中断。 必须在 ADST 为 0 的状态下转换运行模式。
13、12	—	全 0	R	保留位 读写的值总是 0。



位	位名	初始值	R/W	说 明
13、12	—	全 0	R	保留位 读写的值总是 0。
11	TRGE	0	R/W	触发的允许 设定由 ADTRG 或者 MTU2 触发或者 MTU2S 触发产生的 A/D 转换开始。 0: 由触发产生的 A/D 转换开始无效 1: 由触发产生的 A/D 转换开始有效 必须在 ADST 为 0 的状态下转换运行模式。
10	—	0	R	保留位 读写的值总是 0。
9	CONADF	0	R/W	ADF 控制 控制 2 通道扫描模式中的 ADF 运行。此位只在 2 通道扫描模式并且在设定为由触发产生的 A/D 转换开始 (TRGE=1) 时有效, 而在单通道模式、4 通道扫描模式或者 8 通道扫描模式时被忽视。 0: 在组 0 触发或者组 1 触发的转换结束时, ADF 被置位。 1: 在组 0 触发和组 1 触发的转换都结束时, ADF 被置位, 并且不影响触发的顺序。 必须在 ADST 为 0 的状态下转换运行模式。
8	STC	0	R/W	状态控制 此位和 CKSL1、CKSL0 组合, 进行 A/D 转换时间的设定。 0: 50 个状态 1: 64 个状态 必须在 ADST 为 0 的状态下进行 A/D 转换时间的转换。
7、6	CKSL[1:0]	00	R/W	时钟的选择 1、0 设定 A/D 转换时间。 00: $P\phi/4$ 01: $P\phi/3$ 10: $P\phi/2$ 11: $P\phi$ 必须在 ADST 为 0 的状态下进行 A/D 转换时间的转换。 在 $P\phi \leq 25[\text{MHz}]$ 时, 可设定 CKSL[1:0]=B'11。
5、4	ADM[1:0]	00	R/W	A/D 模式 1、0 选择 A/D 转换的运行模式。2 通道扫描模式能用于 A/D 模块 0 和 A/D 模块 1, 不能对 A/D 模块 2 设定 2 通道扫描模式。 00: 单通道模式 01: 4 通道扫描模式 10: 8 通道扫描模式 11: 2 通道扫描模式 必须在 ADST 为 0 的状态下转换运行模式。

3	ADCS	0	R/W	A/D 连续扫描 选择扫描模式中的单周期扫描模式或者连续扫描模式。只在扫描模式时有效。 0: 单周期扫描 1: 连续扫描 必须在 ADST 为 0 的状态下转换运行模式。
2 ~ 0	CH[2:0]	000	R/W	通道的选择 2 ~ 0 选择 A/D 转换的模拟输入通道 (参照表 19.3)。 必须在 ADST 为 0 的状态下转换运行模式。

【注】 \* 为了清除标志, 只能在读 1 后写 0。

### 19.3.3 A/D 控制寄存器\_0 ~ 2 (ADCR\_0 ~ 2)

ADCR 控制各模块的 A/D 转换运行。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	ADST	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15、14	—	全 0	R	保留位 读写的值总是 0。
13	ADST	0	R/W	A/D 的开始 如果清 0, 就中止 A/D 转换, 进入待机状态; 如果置 1, 就开始 A/D 转换。在单通道模式和单周期扫描模式中, 当所选通道的 A/D 转换结束时, 此位就自动清除。在连续扫描模式中, 在通过软件、复位、软件待机模式或者模块待机模式清除此位前, 依次连续转换所选的通道。
12 ~ 0	—	全 0	R	保留位 读写的值总是 0。

表 19.3 通道选择一览表

## • 单模式

bit2	bit1	bit0	模拟输入通道		
CH2	CH1	CH0	单模式		
			A/D_0	A/D_1	A/D_2
0	0	0	AN0	AN4	AN8
		1	AN1	AN5	AN9
	1	0	AN2	AN6	AN10
		1	AN3	AN7	AN11
1	0	0	禁止设定	禁止设定	AN12
		1			AN13
	1	0	禁止设定	禁止设定	AN14
		1			AN15

## • 2 通道扫描模式

bit2	bit1	bit0	模拟输入通道							
CH2	CH1	CH0	启动软件时			软件启动以外				
			A/D_0	A/D_1	A/D_2	A/D_0		A/D_1		A/D_2
						Group0	Group1	Group0	Group1	
0	0	0	AN0	AN4	禁止设定	AN0	AN2	AN4	AN6	禁止设定
		1	AN0、AN1	AN4、AN5		AN0、AN1	AN2、AN3	AN4、AN5	AN6、AN7	
	1	0	AN2	AN6		禁止设定	禁止设定	禁止设定	禁止设定	
		1	AN2、AN3	AN6、AN7						
1	0	0	禁止设定	禁止设定	禁止设定	禁止设定	禁止设定	禁止设定	禁止设定	
		1								
	1	0	禁止设定	禁止设定						
		1								

【注】 即使设定 2、4、8 通道扫描模式，也只有在 CH[2:0] 上选择的通道才进行运行。例如，即使在连续扫描模式下设定 8 通道扫描模式，如果设定 CH[2:0]=000 时，AN8 的转换就连续执行。

## • 4 通道扫描模式

bit2	bit1	bit0	模拟输入通道		
CH2	CH1	CH0	4 通道扫描模式 *		
			A/D_0	A/D_1	A/D_2
0	0	0	AN0	AN4	AN8
		1	AN0、AN1	AN4、AN5	AN8、AN9
	1	0	AN0 ~ AN2	AN4 ~ AN6	AN8 ~ AN10
		1	AN0 ~ AN3	AN4 ~ AN7	AN8 ~ AN11
1	0	0	禁止设定	禁止设定	AN12
		1			AN12、AN13
	1	0			AN12 ~ AN14
		1			AN12 ~ AN15

【注】 \* 通过 ADCS 位可以设定连续扫描或者 1 个周期扫描。

即使设定 2、4、8 通道扫描模式，也只有在 CH[2:0] 上选择的通道才进行运行。例如，即使在连续扫描模式下设定 8 通道扫描模式，如果设定 CH[2:0]=000 时，AN8 的转换就连续执行。

## • 8 通道扫描模式

bit2	bit1	bit0	模拟输入通道
CH2	CH1	CH0	8 通道扫描模式 *
			A/D_2
0	0	0	AN8
		1	AN8、AN9
	1	0	AN8 ~ AN10
		1	AN8 ~ AN11
1	0	0	AN8 ~ AN12
		1	AN8 ~ AN13
	1	0	AN8 ~ AN14
		1	AN8 ~ AN15

【注】 \* 通过 ADCS 位可以设定连续扫描或者 1 个周期扫描。

即使设定 2、4、8 通道扫描模式，也只有在 CH[2:0] 上选择的通道才进行运行。例如，即使在连续扫描模式下设定 8 通道扫描模式，如果设定 CH[2:0]=000 时，AN8 的转换就连续执行。

## 19.3.4 A/D 触发选择寄存器\_0、1 (ADTSR\_0、1)

ADTSR 允许由外部触发产生的 A/D 转换开始。

尤其是在 2 通道扫描模式中，A/D 模块 0 以及 A/D 模块 1 内的 4 个通道分为组 0 和组 1 共 2 组，并能指定各自独立的 A/D 触发。

## • ADTSR\_0

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRG11S[3:0]				TRG01S[3:0]				TRG1S[3:0]				TRG0S[3:0]			
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 12	TRG11S[3:0]	0000	R/W	<p>A/D 触发 1 的组 1 选择 3、2、1、0 对 A/D 模块 1 的 2 通道扫描模式中的组 1，选择外部触发或者 MTU2 或者 MTU2S 的 A/D 转换开始触发。</p> <p>0000: 外部触发引脚 (ADTRG) 的输入 0001: MTU2 各通道的 TGRA 输入捕捉 / 比较匹配，或者在互补 PWM 模式中 TCNT_4 的波谷 (TRGAN) 0010: MTU2 的 CH0 比较匹配 (TRG0N) 0011: MTU2 的 A/D 转换开始请求延迟 (TRG4AN) 0100: MTU2 的 A/D 转换开始请求延迟 (TRG4BN) 0101: MTU2S 各通道的 TGRA 输入捕捉 / 比较匹配，或者在互补 PWM 模式中 TCNT_4 的波谷 (TRGAN) 0110: 禁止设定 0111: MTU2S 的 A/D 转换开始请求延迟 (TRG4AN) 1000: MTU2S 的 A/D 转换开始请求延迟 (TRG4BN) 1001: 禁止设定 101x: 禁止设定 11xx: 禁止设定</p> <p>必须在 A/D 控制寄存器 (ADCR) 的 ADST 为 0 的状态下进行选择器的转换。 在 2 通道扫描模式中，为了不同时产生组 0 和组 1 的转换请求，必须给组 0 和组 1 的转换请求分别指定不同的源。</p>

位	位名	初始值	R/W	说 明
11 ~ 8	TRG01S[3:0]	0000	R/W	<p>A/D 触发 0 的组 1 选择 3、2、1、0</p> <p>对 A/D 模块 0 的 2 通道扫描模式中的组 1，选择外部触发或者 MTU2 或者 MTU2S 的 A/D 转换开始触发。</p> <p>0000: 外部触发引脚 (ADTRG) 的输入</p> <p>0001: 在 MTU2 各通道的 TGRA 输入捕捉 / 比较匹配，或者在互补 PWM 模式中 TCNT_4 的波谷 (TRGAN)</p> <p>0010: MTU2 的 CH0 比较匹配 (TRG0N)</p> <p>0011: MTU2 的 A/D 转换开始请求延迟 (TRG4AN)</p> <p>0100: MTU2 的 A/D 转换开始请求延迟 (TRG4BN)</p> <p>0101: 在 MTU2S 各通道的 TGRA 输入捕捉 / 比较匹配，或者在互补 PWM 模式中 TCNT_4 的波谷 (TRGAN)</p> <p>0110: 禁止设定</p> <p>0111: MTU2S 的 A/D 转换开始请求延迟 (TRG4AN)</p> <p>1000: MTU2S 的 A/D 转换开始请求延迟 (TRG4BN)</p> <p>1001: 禁止设定</p> <p>101x: 禁止设定</p> <p>11xx: 禁止设定</p> <p>必须在 A/D 控制寄存器 (ADCR) 的 ADST 为 0 的状态下进行选择器的转换。</p> <p>在 2 通道扫描模式中，为了不同时产生组 0 和组 1 的转换请求，必须给组 0 和组 1 的转换请求分别指定不同的源。</p>
7 ~ 4	TRG1S[3:0]	0000	R/W	<p>A/D 触发 1 的选择 3、2、1、0</p> <p>选择 A/D 模块 1 的外部触发或者 MTU2 或者 MTU2S 的 A/D 转换开触发。</p> <p>0000: 外部触发引脚 (ADTRG) 的输入</p> <p>0001: 在 MTU2 各通道的 TGRA 输入捕捉 / 比较匹配，或者在互补 PWM 模式中 TCNT_4 的波谷 (TRGAN)</p> <p>0010: MTU2 的 CH0 比较匹配 (TRG0N)</p> <p>0011: MTU2 的 A/D 转换开始请求延迟 (TRG4AN)</p> <p>0100: MTU2 的 A/D 转换开始请求延迟 (TRG4BN)</p> <p>0101: 在 MTU2S 各通道的 TGRA 输入捕捉 / 比较匹配，或者在互补 PWM 模式中 TCNT_4 的波谷 (TRGAN)</p> <p>0110: 禁止设定</p> <p>0111: MTU2S 的 A/D 转换开始请求延迟 (TRG4AN)</p> <p>1000: MTU2S 的 A/D 转换开始请求延迟 (TRG4BN)</p> <p>1001: 禁止设定</p> <p>101x: 禁止设定</p> <p>11xx: 禁止设定</p> <p>必须在 A/D 控制寄存器 (ADCR) 的 ADST 为 0 的状态下进行选择器的转换。</p> <p>在 2 通道扫描模式中，为了不同时产生组 0 和组 1 的转换请求，必须给组 0 和组 1 的转换请求分别指定不同的源。</p>

位	位名	初始值	R/W	说 明
3 ~ 0	TRG0S[3:0]	0000	R/W	<p>A/D 触发 0 的选择 3、2、1、0 选择 A/D 模块 0 的外部触发或者 MTU2 或者 MTU2S 的 A/D 转换开始触发。在 2 通道扫描模式中，对组 0 选择外部触发或者 MTU2 或者 MTU2S 的 A/D 转换开始触发。</p> <p>0000: 外部触发引脚 (ADTRG) 的输入 0001: 在 MTU2 各通道的 TGRA 输入捕捉 / 比较匹配，或者在互补 PWM 模式中 TCNT_4 的波谷 (TRGAN) 0010: MTU2 的 ch0 比较匹配 (TRG0N) 0011: MTU2 的 A/D 转换开始请求延迟 (TRG4AN) 0100: MTU2 的 A/D 转换开始请求延迟 (TRG4BN) 0101: 在 MTU2S 各通道的 TGRA 输入捕捉 / 比较匹配，或者在互补 PWM 模式中 TCNT_4 的波谷 (TRGAN) 0110: 禁止设定 0111: MTU2S 的 A/D 转换开始请求延迟 (TRG4AN) 1000: MTU2S 的 A/D 转换开始请求延迟 (TRG4BN) 1001: 禁止设定 101x: 禁止设定 11xx: 禁止设定</p> <p>必须在 A/D 控制寄存器 (ADCR) 的 ADST 为 0 的状态下进行选择器的转换。 在 2 通道扫描模式中，为了不同时产生组 0 和组 1 的转换请求，必须给组 0 和组 1 的转换请求分别指定不同的源。</p>

【符号说明】 x: Don't care

• ADTSR\_1

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRG2S[3:0]				-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 12	TRG2S[3:0]	0000	R/W	<p>A/D 触发 2 的选择 3、2、1、0 选择 A/D 模块 2 的外部触发或者 MTU2 或者 MTU2S 的 A/D 转换开触发。</p> <p>0000: 外部触发引脚 (<math>\overline{\text{ADTRG}}</math>) 的输入 0001: 在 MTU2 各通道的 TGRA 输入捕捉 / 比较匹配, 或者在互补 PWM 模式中 TCNT_4 的波谷 (TRGAN) 0010: MTU2 的 CH0 比较匹配 (TRG0N) 0011: MTU2 的 A/D 转换开始请求延迟 (TRG4AN) 0100: MTU2 的 A/D 转换开始请求延迟 (TRG4BN) 0101: 在 MTU2S 各通道的 TGRA 输入捕捉 / 比较匹配, 或者在互补 PWM 模式中 TCNT_4 的波谷 (TRGAN) 0110: 禁止设定 0111: MTU2S 的 A/D 转换开始请求延迟 (TRG4AN) 1000: MTU2S 的 A/D 转换开始请求延迟 (TRG4BN) 1001: 禁止设定 101x: 禁止设定 11xx: 禁止设定</p> <p>必须在 A/D 控制寄存器 (ADCR) 的 ADST 为 0 的状态下进行选择器的转换。</p>
11 ~ 0	—	全 0	R	<p>保留位 读写的值总是 0。</p>

【符号说明】 x: Don't care



## 19.4 运行说明

A/D 转换器采用逐次逼近方式，分辨率为 10 位。运行模式有单通道模式和扫描模式。扫描模式有连续扫描模式和单周期扫描模式。为了避免误动作，必须在 ADCR 的 ADST 位为 0 的状态下转换运行模式或者模拟输入通道。

### 19.4.1 单通道模式

单通道模式将指定的 1 个通道的模拟输入进行如下的 1 次 A/D 转换：

1. 如果通过软件、MTU2、MTU2S 或者外部触发输入将 ADCR 的 ADST 位设定为 1，就开始对所选通道进行 A/D 转换。
2. 如果 A/D 转换结束，就将 A/D 转换结果传送到对应该通道的 A/D 数据寄存器。
3. 在 A/D 转换结束后，ADCSR 的 ADF 位被置 1。此时，如果 ADIE 位已被置 1，就产生 ADI 中断请求。
4. ADST 位在 A/D 转换中保持 1，当转换结束时，就自动清除并且 A/D 转换器变为待机状态。如果在 A/D 转换中 ADST 位被清 0，就中止转换并且 A/D 转换器变为待机状态。

### 19.4.2 连续扫描模式

连续扫描模式将指定通道（SH7083/84/85 最多为 4 个通道，SH7086 最多为 8 个通道）的模拟输入依次进行如下的 A/D 转换：

1. 如果通过软件、MTU2、MTU2S 或者外部触发输入将 ADCR 的 ADST 位设定为 1，就按照模拟输入通道号从小到大的顺序（例如，AN0、AN1…AN7）进行 A/D 转换。
2. 如果各通道的 A/D 转换结束，就将 A/D 转换结果依次传送到对应该通道的 A/D 数据寄存器。
3. 如果所选通道的 A/D 转换全部结束，ADCSR 的 ADF 位就被置 1。此时，如果 ADIE 位已被置 1，就产生 ADI 中断请求。A/D 转换器再次从组中的第 1 个通道开始 A/D 转换。
4. ADST 位不自动清除，在置为 1 的期间重复执行 2. ~ 3.。如果 ADST 位被清 0，就中止 A/D 转换并且 A/D 转换器变为待机状态。

### 19.4.3 单周期扫描模式

单周期扫描模式将指定的通道（SH7083/84/85 最多为 4 个通道，SH7086 最多为 8 个通道）的模拟输入进行如下的 1 次 A/D 转换。

1. 如果通过软件、MTU2、MTU2S 或者外部触发输入将 ADCR 的 ADST 位设定为 1，就按照模拟输入通道号从小到大的顺序（例如，AN0、AN1…AN7）进行 A/D 转换。
2. 如果各通道的 A/D 转换结束，就将 A/D 转换结果依次传送到对应该通道的 A/D 数据寄存器。
3. 如果所选通道的 A/D 转换全部结束，ADCSR 的 ADF 位就被置 1。此时，如果 ADIE 位已被置 1，就产生 ADI 中断请求。
4. 如果转换结束，ADST 位就自动清除并且 A/D 转换器变为待机状态。如果在 A/D 转换中 ADST 位被清 0，就中止转换并且 A/D 转换器变为待机状态。

#### 19.4.4 输入采样和 A/D 转换时间

A/D 转换器内置各模块的采样和保持电路。如果在将 ADCR 的 ADST 位置 1 后经过 A/D 转换开始的延迟时间 ( $t_D$ )，A/D 转换器就对输入进行采样，然后开始转换。A/D 转换的时序如图 19.2、A/D 转换时间如表 19.4 所示。

如图 19.2 所示，A/D 转换时间 ( $t_{CONV}$ ) 包含  $t_D$  和输入采样时间 ( $t_{SPL}$ )。在此， $t_D$  由 ADCR 的写时序决定，而不是定值。因此，转换时间在表 19.4 所示的范围内发生变化。

关于扫描模式的转换时间，表 19.4 所示的值为第 1 次转换时间，表 19.5 所示的值为第 2 次以后的转换时间。

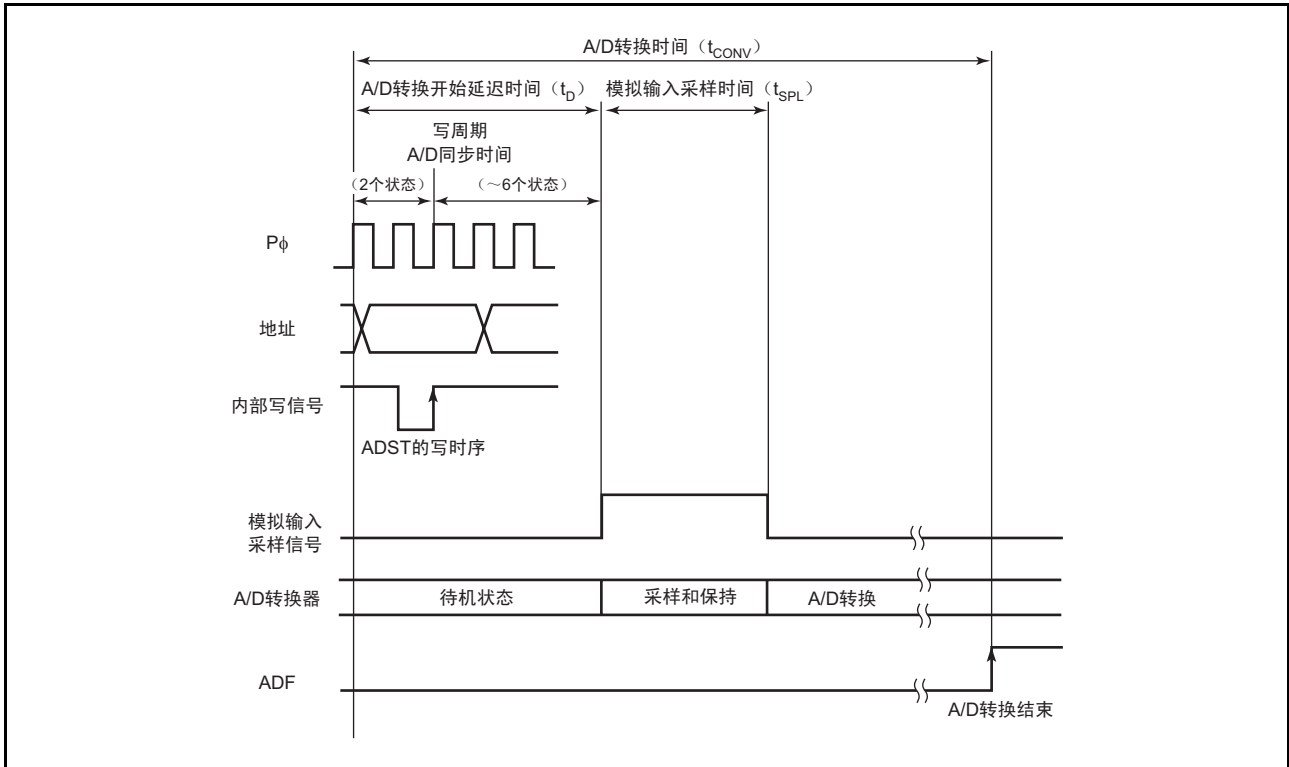


图 19.2 A/D 转换时序

表 19.4 A/D 转换时间 (单通道模式)

项 目	符号	STC=0											
		CKSL1=0						CKSL1=1					
		CKSL0=0			CKSL0=1			CKSL0=0			CKSL0=1		
		Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.
A/D 转换开始的延迟时间	$t_D$	2	—	6	2	—	5	2	—	4	2	—	3
输入采样时间	$t_{SPL}$	—	24	—	—	18	—	—	12	—	—	6	—
A/D 转换时间	$t_{CONV}$	202	—	206	152	—	155	102	—	104	52	—	53

项 目	符号	STC=1											
		CKSL1=0						CKSL1=1					
		CKSL0=0			CKSL0=1			CKSL0=0			CKSL0=1		
		Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.
A/D 转换开始的延迟时间	$t_D$	2	—	6	2	—	5	2	—	4	2	—	3
输入采样时间	$t_{SPL}$	—	36	—	—	27	—	—	18	—	—	9	—
A/D 转换时间	$t_{CONV}$	258	—	262	194	—	197	130	—	132	66	—	67

【注】 表中的数值单位是相对于  $P\phi$  的状态。

表 19.5 A/D 转换时间 (扫描模式)

STC	CKSL1	CKSL0	转换时间 (状态)	转换时间的计算例	
				在 $P\phi=25\text{MHz}$ 时	在 $P\phi=40\text{MHz}$ 时
0	0	0	200 (固定)	$8\mu\text{s}$	$5\mu\text{s}$
		1	150 (固定)	$6\mu\text{s}$	$3.8\mu\text{s}$
	1	0	100 (固定)	$4\mu\text{s}$	$2.5\mu\text{s}$
		1	50 (固定)	$2\mu\text{s}$	禁止设定
1	0	0	256 (固定)	$10.2\mu\text{s}$	$6.4\mu\text{s}$
		1	192 (固定)	$7.7\mu\text{s}$	$4.8\mu\text{s}$
	1	0	128 (固定)	$5.1\mu\text{s}$	$3.2\mu\text{s}$
		1	64 (固定)	$2.6\mu\text{s}$	禁止设定

### 19.4.5 通过 MTU2 启动 A/D 转换器

能通过 MTU2、MTU2S 间隔定时器的 A/D 转换请求，独立启动 A/D 转换器。

要通过 MTU2、MTU2S 启动 A/D 转换器时，将 A/D 控制 / 状态寄存器 (ADCSR) 的 TRGE 位设定为 1，然后设定 A/D 触发选择寄存器 (ADTSR)。在此状态下，如果发生 MTU2、MTU2S 间隔定时器的 A/D 转换请求，就将 ADCSR 的 ADST 位设定为 1。从 ADST 位被置 1 后到 A/D 转换开始前的时序与通过软件给 ADST 位写 1 时的时序相同。

### 19.4.6 外部触发的输入时序

也能通过外部触发输入来开始 A/D 转换。在 A/D 控制 / 状态寄存器 (ADCSR) 的  $\overline{\text{TRGE}}$  位设定为 1 后, 将 A/D 触发选择寄存器\_0、1 (ADTSR\_0、ADTSR\_1) 设定为外部触发引脚输入时, 从  $\overline{\text{ADTRG}}$  引脚输入外部触发。在  $\overline{\text{ADTRG}}$  的下降沿, ADCR 的 ADST 位被置 1, 开始 A/D 转换。与单通道模式和扫描模式无关, 其他运行都和通过软件将 ADST 位设定为 1 的情况相同。此时序如图 19.3 所示。

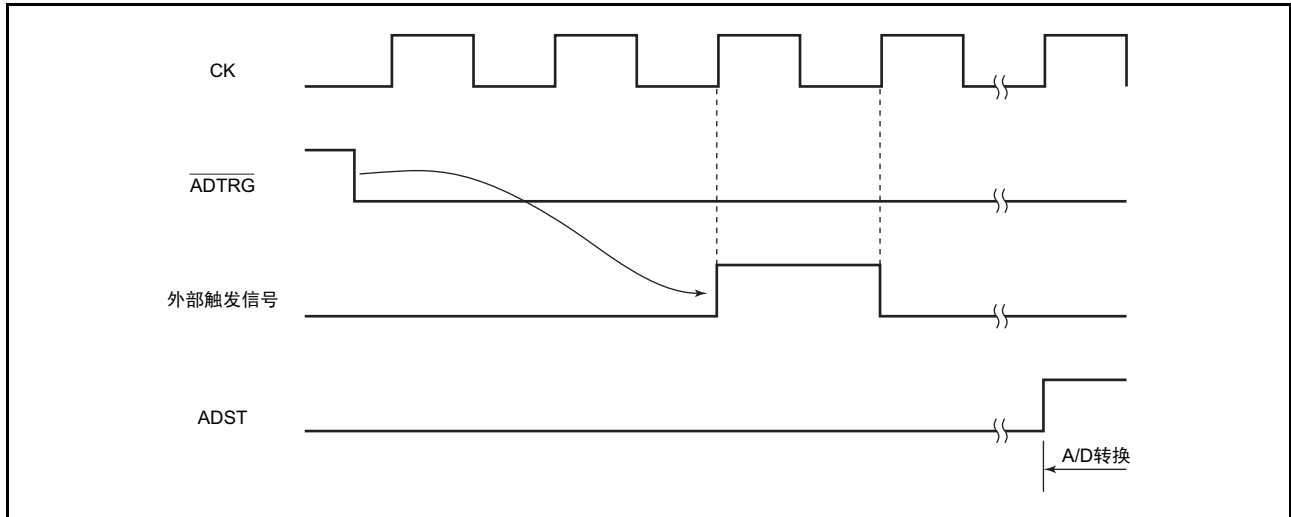


图 19.3 外部触发输入的时序

### 19.4.7 2 通道扫描

2 通道扫描模式是将 4 通道的模拟输入分为组 0 和组 1，通过个别的触发可以给组 0 和组 1 选择启动源。2 通道扫描模式的转换结束中断可以选择组 0 或者组 1 的结束，也可选择组 1 和组 0 的结束。通过触发使转换开始时，必须给 ADTSR 的组 0 和组 1 设定不同的启动源。另外，如果在组 0 的转换中发生组 1 的转换请求时，将忽视组 1 的转换请求。给组 0 的 A/D 转换开始请求设定 MTU2 的 TRG4AN、组 1 的 A/D 转换开始请求设定 MTU2 的 TRG4BN 时的运行例子如图 19.4 所示。

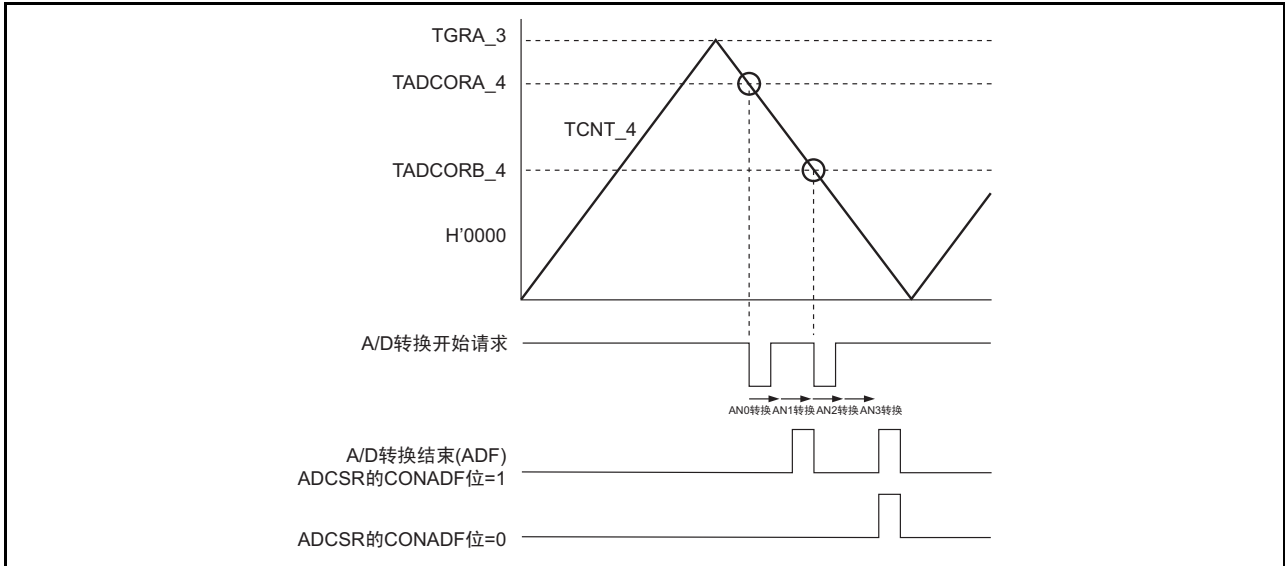


图 19.4 2 通道扫描的运行例子

### 19.5 中断源和 DMAC/DTC 传送请求

A/D 转换器能产生 A/D 转换结束的中断请求 (ADI)。如果将 A/D 控制 / 状态寄存器 (ADCSR) 的 ADIE 位设定为 1 就能允许 ADI；如果清 0 就能禁止 ADI。

另外，能在发生 ADI 时启动 DMAC/DTC，此时不向 CPU 产生中断。

如果通过 ADI 启动 DMAC/DTC，就在通过 DMAC/DTC 进行数据传送时自动清除 ADCSR 的 ADF 位。如果通过 DMAC/DTC 读取由 ADI 中断转换的数据，就能实现不给软件产生任何负担的连续转换。

表 19.6 A/D 转换器的中断源

名称	中断源	中断标志	DTC 的启动	DMAC 的启动
ADI0	A/D_0 转换结束	ADCSR_0 的 ADF	可	不可
ADI1	A/D_1 转换结束	ADCSR_1 的 ADF	可	可
ADI2	A/D_2 转换结束	ADCSR_2 的 ADF	可	不可

## 19.6 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的位数

- 量化误差

A/D 转换固有的偏差，为  $1/2\text{LSB}$ （图 19.5）。偏移误差

在数字输出从最小电压值 B'000000000 (H'00) 变为 B'000000001 (H'01) 时，与模拟输入电压值的理想 A/D 转换特性的偏差（图 19.6）

- 满刻度误差

在数字输出从 B'111111110 (H'3FE) 变为 B'111111111 (H'3FF) 时，与模拟输入电压值的理想 A/D 转换特性引起的偏差（图 19.6）

- 非线性误差

在零电压和满刻度电压之间，与理想 A/D 转换特性的误差。但是，不包括偏移误差、满刻度误差和量化误差（图 19.6）。

- 绝对精度

数字值与模拟输入值的偏差，包括偏移误差、满刻度误差、量化误差和非线性误差。

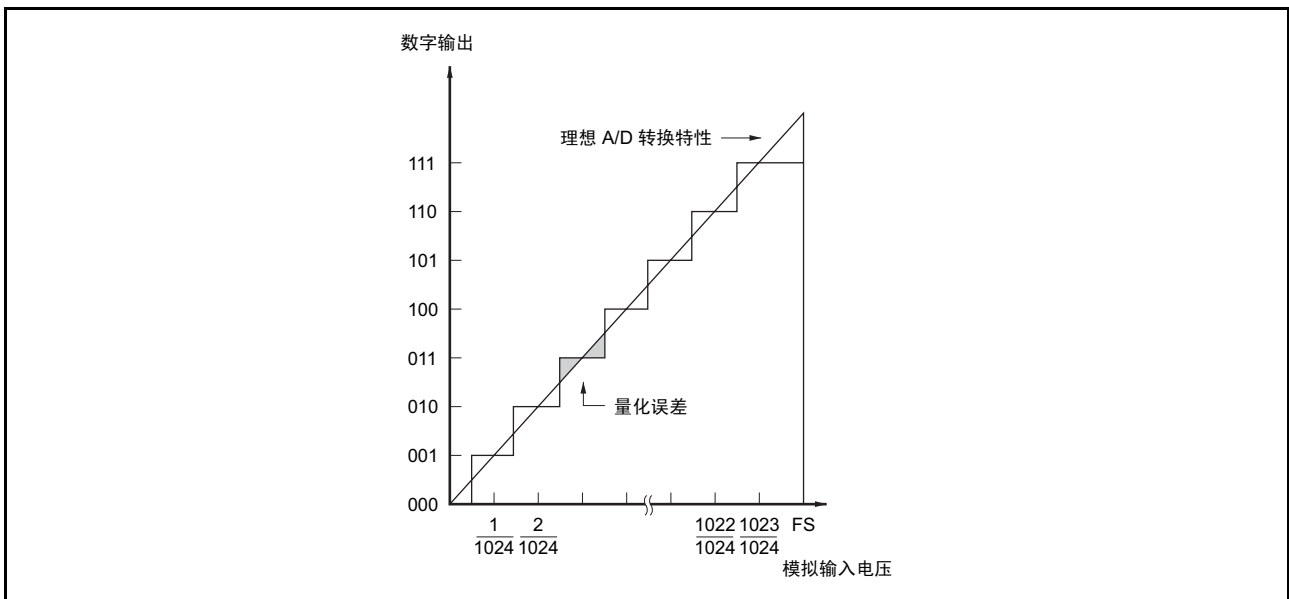


图 19.5 A/D 转换精度的定义

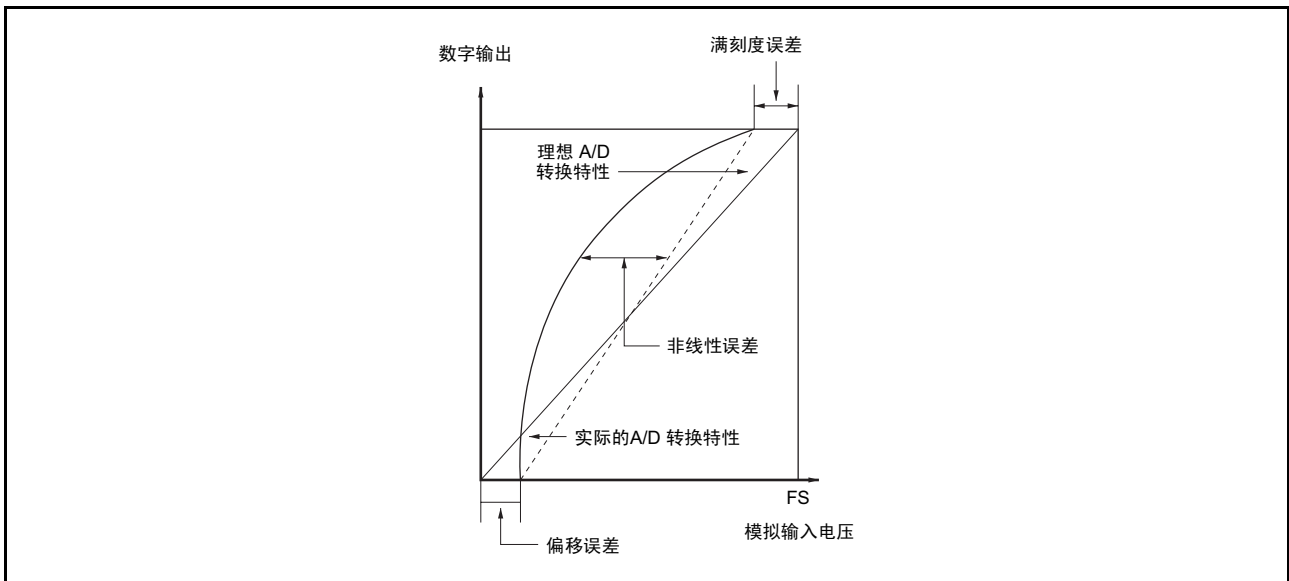


图 19.6 A/D 转换精度的定义

## 19.7 使用注意事项

### 19.7.1 模块待机模式的设定

A/D 转换器能通过待机控制寄存器允许或者禁止本模块的运行，初始值为停止 A/D 转换器的运行。通过解除模块待机模式，允许寄存器的存取。详细内容请参照“第 26 章 低功耗模式”。

### 19.7.2 关于容许信号源阻抗

对于信号源阻抗不超过  $1\text{k}\Omega$  的输入信号，本 LSI 的模拟输入能保证转换精度，这是为了在采样时间内，对 A/D 转换器的采样和保持电路的输入电容进行充电而制定的规格。在传感器的输出阻抗超过  $1\text{k}\Omega$  时，有可能发生充电不足并且不能保证 A/D 转换精度的情况。在单通道模式中进行转换并且在外接大电容的情况下，因为输入负载实际上只有  $10\text{k}\Omega$  的内部输入电阻，所以信号源阻抗可忽略不计。但是，由于形成低通滤波器，所以有可能无法跟踪大微分系数的模拟信号（例如，电压的变动率在  $5\text{mV}/\mu\text{s}$  以上）（图 19.7）。在转换高速模拟信号或者在扫描模式中进行转换时，必须插入一个低阻抗的缓冲器。

### 19.7.3 关于对绝对精度的影响

由于附加电容会导致与 GND 的耦合，因此，如果在 GND 中有噪声，就有可能降低绝对精度，所以必须与 AVSS 等电稳定的 GND 连接。

另外，必须注意：在安装电路板上滤波器电路不要干涉数字信号也不要充当天线。

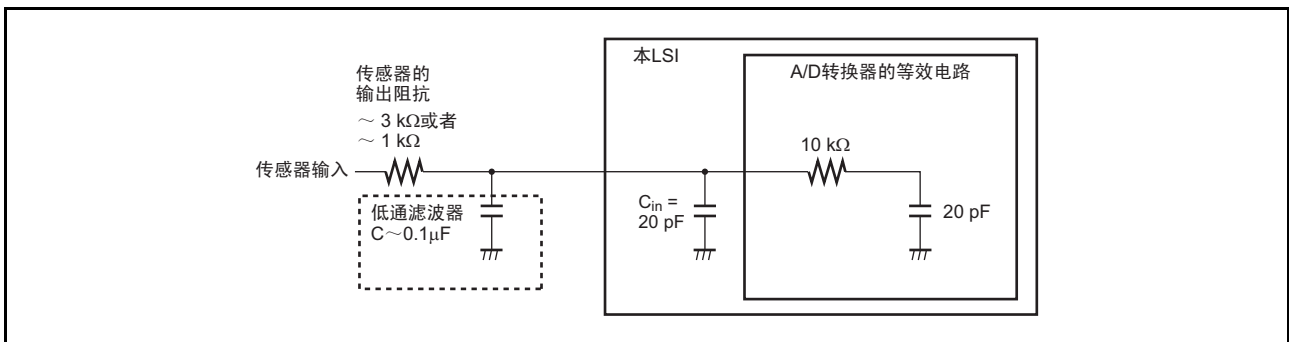


图 19.7 模拟输入电路的例子

### 19.7.4 模拟电源引脚等的设定范围

如果超出以下电压的设定范围使用 LSI，就会给 LSI 的可靠性带来不良影响。

- 模拟输入电压的设定范围  
在 A/D 转换中，必须将模拟输入引脚 ANn 的外加电压设定在  $AVSS \leq VAN \leq AVref$  的范围内。
- AVcc、AVss 和 Vcc、Vss 的关系  
AVcc、AVss 和 Vcc、Vss 的关系为  $AVSS=VSS$ ，并且在不使用 A/D 转换器时，AVcc 和 AVss 引脚不能开路。
- AVref 输入电压的设定范围  
AVref 引脚的输入电压必须  $AVref \leq AVcc$ 。  
当不使用 A/D 转换器时，必须  $AVref=AVcc$ 。



### 19.7.5 电路板设计的注意事项

在设定电路板时，必须尽量将数字电路和模拟电路分开，不能使数字电路的信号线和模拟电路的信号布线交叉或者靠近。因电感等引起模拟电路的误动作并给 A/D 转换值带来不良影响。模拟输入引脚 (AN0 ~ AN15) 和模拟电源电压 (AVcc) 必须通过模拟接地 (AVss) 和数字电路分开，并且模拟接地 (AVss) 必须单点连接到电路板上稳定的接地 (Vss)。

### 19.7.6 噪声对策的注意事项

为了防止因过大电涌等异常电压引起对模拟输入引脚 (AN0 ~ AN15) 的破坏，如图 19.8 所示，必须将保护电路连接在 AVcc—AVss 之间，并将连接 AVcc 的旁路电容以及连接 AN0 ~ AN15 的滤波器电容连接到 AVss。

另外，如果连接用于滤波器的电容，AN0 ~ AN15 的输入电流就被平均，所以有可能产生误差。在扫描模式等模式中频繁地进行 A/D 转换时，如果对 A/D 转换器内部的采样和保持电路的电容进行充放电的电流大于从输入阻抗 (Rin) 输入的电流，模拟输入引脚的电压就会产生误差。因此，在决定电路常数时，请进行充分的探讨。

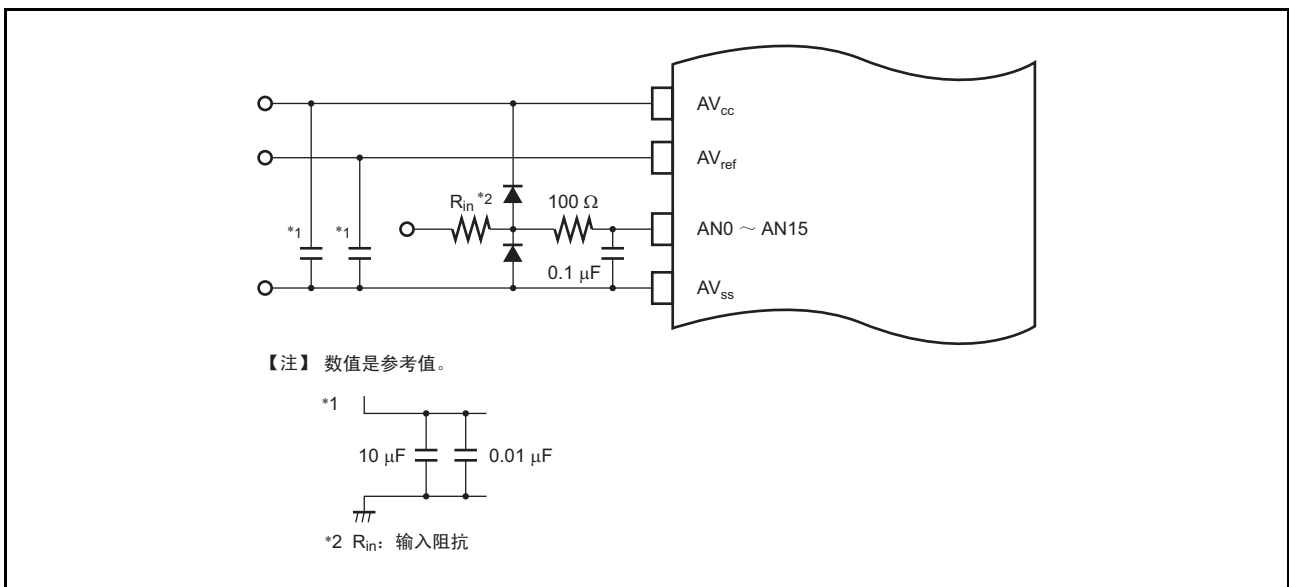


图 19.8 模拟输入保护电路的例子

表 19.7 模拟引脚的规格

项目	Min.	Max.	单位	条件
模拟输入电容	—	20	pF	—
容许信号源阻抗	—	3	kΩ	$P\phi \leq 20\text{MHz}$
		1		$P\phi > 20\text{MHz}$

## 第 20 章 比较匹配定时器 (CMT)

本 LSI 内置由 2 个通道的 16 位定时器构成的比较匹配定时器 (CMT)。CMT 有 16 位的计数器，能按各设定的周期产生中断。

### 20.1 特点

- 2 个通道可独立选择 4 种计数器输入时钟  
可选择 4 种内部时钟 (Pφ/8、Pφ/32、Pφ/128、Pφ/512)
- 能在比较匹配时请求中断
- 能设定模块待机模式

CMT 的框图如图 20.1 所示。

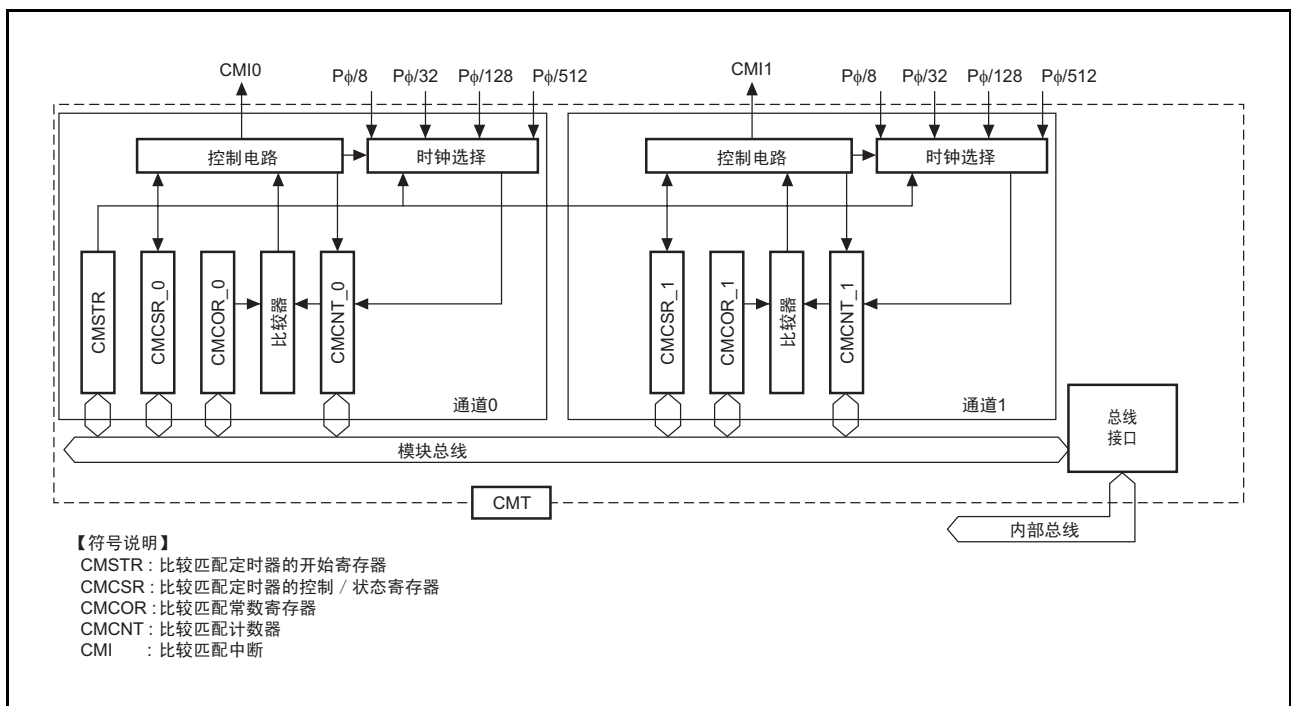


图 20.1 CMT 的框图

## 20.2 寄存器说明

CMT 有以下寄存器。有关这些寄存器的地址和各处理模式中的寄存器状态，请参照“第 27 章 寄存器一览”。另外，本章中省略了通道数。

表 20.1 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
比较匹配定时器的开始寄存器	CMSTR	R/W	H'0000	H'FFFFCE00	8、16、32
比较匹配定时器的控制 / 状态寄存器_0	CMCSR_0	R/W	H'0000	H'FFFFCE02	8、16
比较匹配计数器_0	CMCNT_0	R/W	H'0000	H'FFFFCE04	8、16、32
比较匹配常数寄存器_0	CMCOR_0	R/W	H'FFFF	H'FFFFCE06	8、16
比较匹配定时器的控制 / 状态寄存器_1	CMCSR_1	R/W	H'0000	H'FFFFCE08	8、16、32
比较匹配计数器_1	CMCNT_1	R/W	H'0000	H'FFFFCE0A	8、16
比较匹配常数寄存器_1	CMCOR_1	R/W	H'FFFF	H'FFFFCE0C	8、16、32

### 20.2.1 比较匹配定时器的开始寄存器 (CMSTR)

CMSTR 是 16 位寄存器，选择比较匹配计数器 (CMCNT) 的运行 / 停止。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	STR1	STR0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 2	—	全 0	R	保留位 读写的值总是 0。
1	STR1	0	R/W	计数开始 1 选择比较匹配计数器_1 的运行 / 停止。 0: CMCNT_1 停止计数 1: CMCNT_1 开始计数
0	STR0	0	R/W	计数开始 0 选择比较匹配计数器_0 的运行 / 停止。 0: CMCNT_0 停止计数 1: CMCNT_0 开始计数

## 20.2.2 比较匹配定时器的控制 / 状态寄存器 (CMCSR)

CMCSR 是 16 位寄存器，表示比较匹配的产生以及设定中断和计数器输入时钟。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	CMF	CMIE	-	-	-	-	-	CKS[1:0]
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	(R/W)*1	R/W	R	R	R	R	R/W	R/W

【注】\*1 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说 明
15 ~ 8	—	全 0	R	保留位 读写的值总是 0。
7	CMF	0	(R/W)*1	比较匹配标志 表示 CMCNT 和 CMCOR 的值是否一致。 0: CMCNT 和 CMCOR 的值不一致 [清除条件] • 当读 CMF=1 后，写 0 时 *2 • 通过 CMI 中断启动 DTC 并在 DTC 的 MRB 的 DISEL 位为 0 时存取 CMT 寄存器时 [置位条件] 1: CMCNT 和 CMCOR 的值一致
6	CMIE	0	R/W	比较匹配的中断允许 选择在 CMCNT 和 CMCOR 值一致时 (CMF=1) 是允许还是禁止比较匹配中断 (CMI) 的产生。 0: 禁止比较匹配中断 (CMI) 1: 允许比较匹配中断 (CMI)
5 ~ 2	—	全 0	R	保留位 读写的值总是 0。
1、0	CKS[1:0]	00	R/W	时钟选择 1、0 从分频外围运行时钟 (P $\phi$ ) 后的 4 种内部时钟中，选择 CMCNT 的输入时钟。当 CMSTR 的 STR 位被置 1 时，通过 CKS1 和 CKS0 位选择的时钟，CMCNT 开始计数。 00: P $\phi$ /8 01: P $\phi$ /32 10: P $\phi$ /128 11: P $\phi$ /512

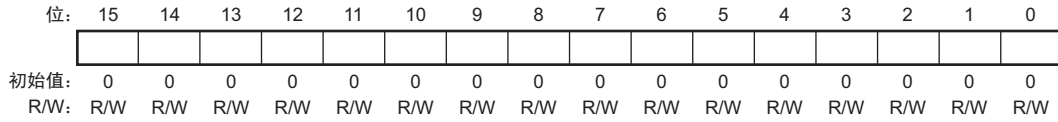
【注】\*1 为了清除标志，只能在读 1 后写 0。

\*2 如果在读 1 后写 0 前发生由下一个比较匹配引起的标志置位，即使写 0 也不清除标志，所以必须重新读 1 后再写 0。

### 20.2.3 比较匹配计数器 (CMCNT)

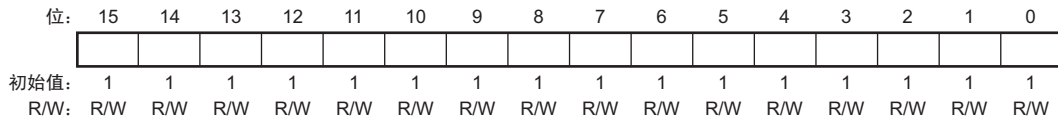
CMCNT 是 16 位寄存器，用作递增计数器。如果通过 CMCSR 的 CKS1、CKS0 位选择计数器输入时钟，并且将 CMST 的 STR 位设定为 1，CMCN 就通过所选的时钟开始计数。当 CMCNT 的值和比较匹配常数寄存器 (CMCOR) 的值一致时，CMCNT 被清除为 H'0000，并且 CMCSR 的 CMF 标志被置 1。

CMCNT 的初始值是 H'0000。



### 20.2.4 比较匹配常数寄存器 (CMCOR)

CMCOR 是 16 位寄存器，设定和 CMCNT 比较匹配前的时间，初始值是 H'FFFF。



## 20.3 运行说明

### 20.3.1 周期计数运行

如果通过 CMCSR 的 CKS1、CKS0 位选择内部时钟，并且将 CMSTR 的 STR 位设定为 1，CMCNT 就通过所选的时钟开始递增计数。当 CMCNT 的值和 CMCOR 的值一致时，CMCNT 被清除为 H'0000，并且 CMCSR 的 CMF 标志被置 1。此时，如果 CMCSR 寄存器的 CMIE 位已被置 1，就请求比较匹配中断 (CMI)，CMCNT 从 H'0000 重新开始递增计数。

比较匹配计数器的运行如图 20.2 所示。

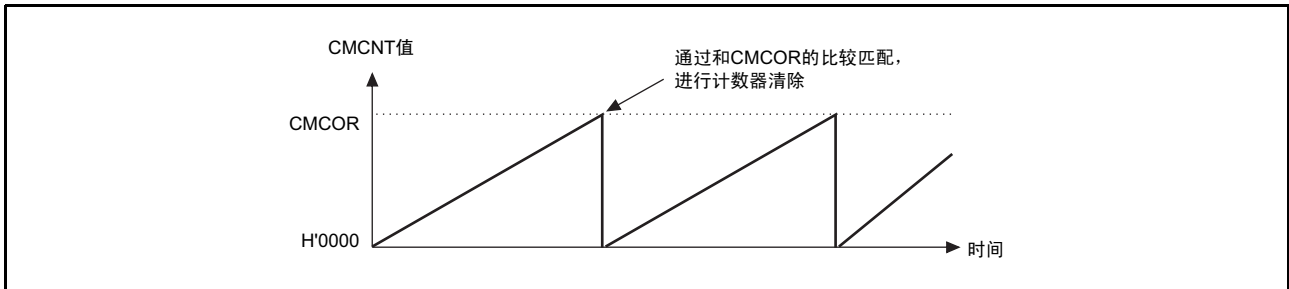


图 20.2 计数器的运行

### 20.3.2 CMCNT 的计数时序

能通过 CMCSR 的 CKS1、CKS0 位，从分频时钟 ( $P\phi$ ) 后的 4 种时钟 ( $P\phi/8$ 、 $P\phi/32$ 、 $P\phi/128$ 、 $P\phi/512$ ) 中选择一中时钟个。该时序如图 20.3 所示。

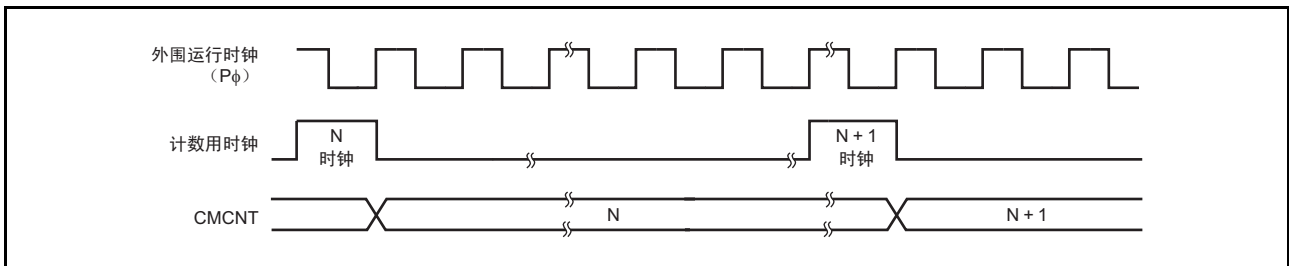


图 20.3 计数时序

## 20.4 中断

### 20.4.1 CMT 的中断源

CMT 有各通道的比较匹配中断，分别分配了独立的向量地址。当中断请求标志 (CMF) 和中断允许位 (CMIE) 都被置 1 时，就输出该中断请求。在通过中断请求启动 CPU 中断的情况下，能通过设定中断控制器更改通道之间的优先级。详细内容请参照“第 6 章 中断控制器 (INTC)”。

另外，还能将中断请求作为数据传送控制器 (DTC) 的启动源。此时，通道间的优先顺序固定。详细内容请参照“第 8 章 数据传送控制器 (DTC)”。

### 20.4.2 比较匹配标志的置位时序

如果 CMCOR 和 CMCNT 一致，就产生比较匹配信号，并且 CMCSR 的 CMF 位被置 1。在一致的最后状态 (将 CMCNT 的值更新为 H'0000 时) 产生比较匹配信号。即，如果在 CMCOR 和 CMCNT 一致后不输入用于 CMCNT 计数器的时钟，就不产生比较匹配信号。CMF 位的置位时序如图 20.4 所示。

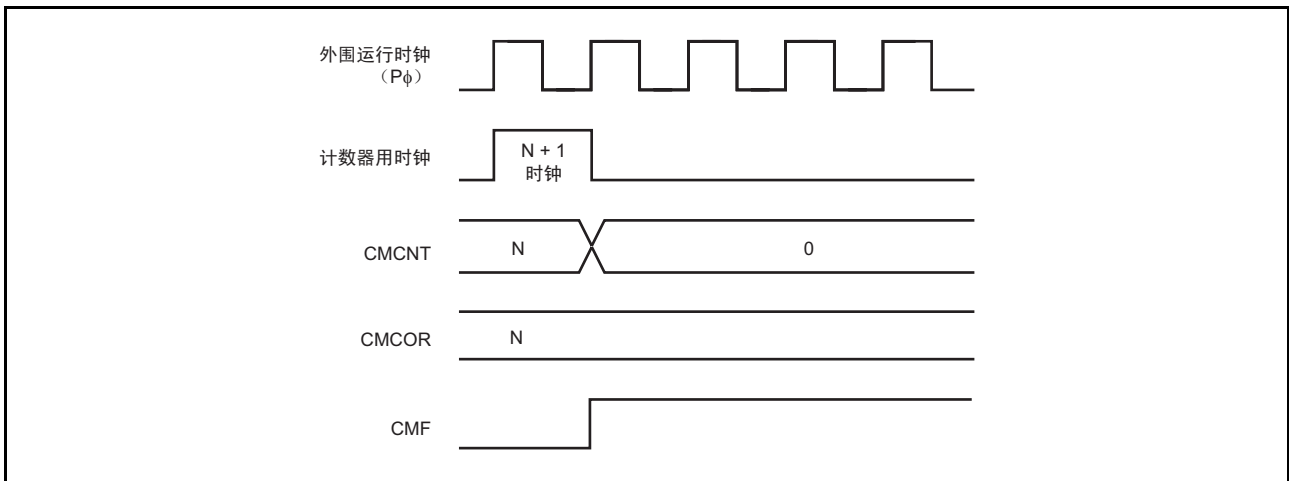


图 20.4 CMF 位的置位时序

### 20.4.3 比较匹配标志的清除时序

通过在读 CMF=1 后写 0，清除 CMCSR 的 CMF 位。

## 20.5 使用注意事项

### 20.5.1 模块待机模式的设定

CMT 能通过待机控制寄存器允许或者禁止本模块的运行，初始值为停止 CMT 的运行。通过解除模块待机模式，允许寄存器的存取。详细内容请参照“第 26 章 低功耗模式”。

### 20.5.2 CMCNT 的写操作和比较匹配的竞争

如果在 CMCNT 计数器的写周期中的 T2 状态产生比较匹配信号，就不写 CMCNT 计数器而优先清除 CMCNT 计数器。此时序如图 20.5 所示。

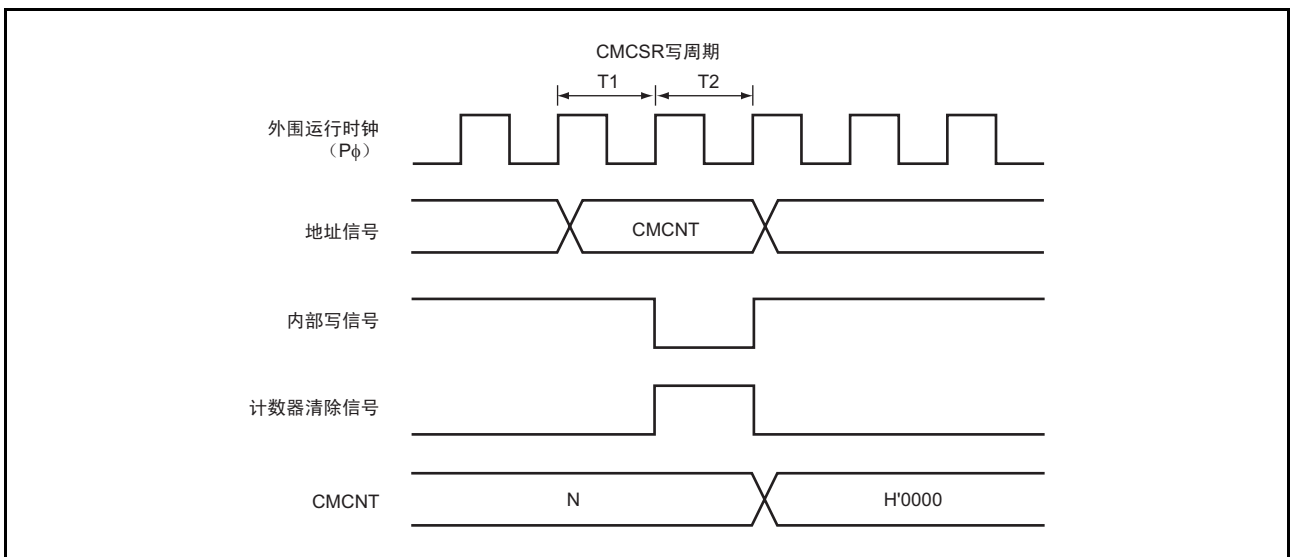


图 20.5 CMCNT 的写操作和比较匹配的竞争

### 20.5.3 CMCNT 的字写和递增计数的竞争

即使在 CMCNT 计数器的字写周期中的 T2 状态发生递增计数，也不进行递增计数而优先写计数器。此时序如图 20.6 所示。

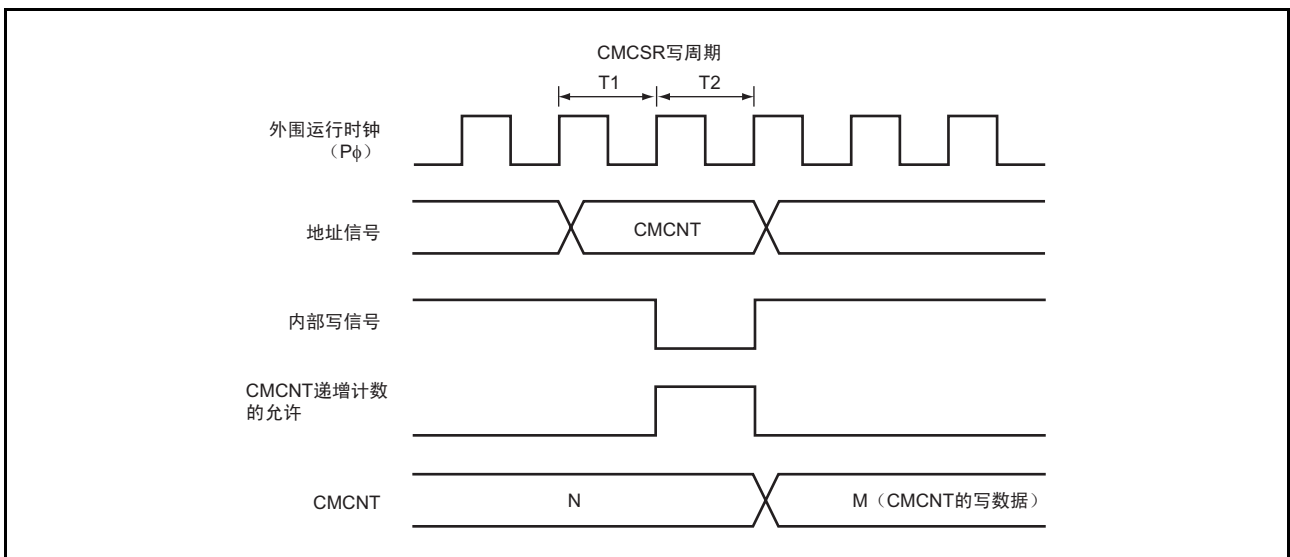


图 20.6 CMCNT 的字写和递增计数的竞争



### 20.5.4 CMCNT 的字节写和递增计数的竞争

即使在 CMCNT 的字节写周期中的 T2 状态发生递增计数，为字节写对象的计数器也不进行递增计数而优先写计数器。并且，不为字节写对象的计数器也不进行递增计数而为写操作之前的内容。

在 CMCNTH 写周期中的 T2 状态发生递增计数时的时序如图 20.7 所示。

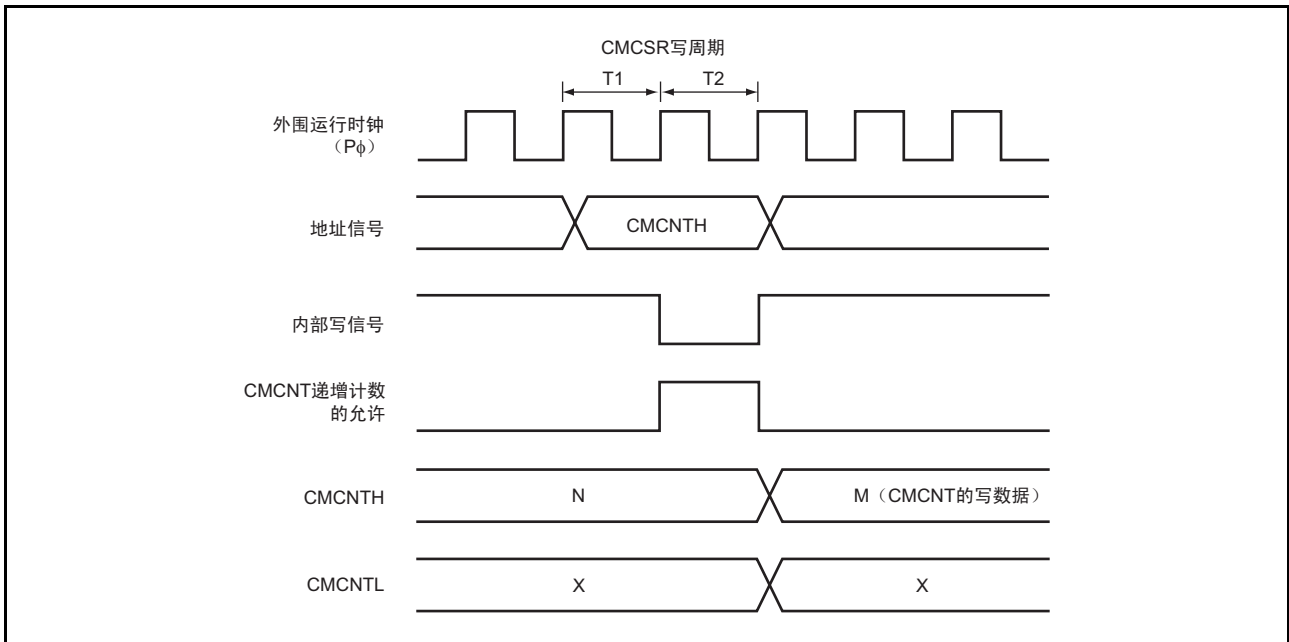


图 20.7 CMCNT 的字节写和递增计数的竞争

### 20.5.5 CMCNT 和 CMCOR 的比较匹配

在停止 CMCNT 计数运行的状态下，不能给 CMCNT 和 CMCOR 设定相同的值。否则，CMCSR 的 CMF 位就被置 1 并且 CMCNT 被清除为 H'0000。

## 第 21 章 引脚功能控制器 (PFC)

引脚功能控制器 (PFC) 由选择多路复用引脚功能及其输入 / 输出方向的寄存器构成。本 LSI 的多路复用引脚如表 21.1 ~ 表 21.16 所示。

按运行模式分类的引脚功能一览表如表 21.17 ~ 表 21.20 所示。

表 21.1 多路复用引脚一览表 (SH7083、端口 A)

端口	功能 1 (相关模块)	功能 2 (相关模块)	功能 3 (相关模块)	功能 4 (相关模块)	功能 5 (相关模块)
A	PA3 输入 / 输出 (端口)	A24 输出 (BSC)	RXD1 输入 (SCI)	—	—
	PA4 输入 / 输出 (端口)	A23 输出 (BSC)	TXD1 输出 (SCI)	—	—
	PA5 输入 / 输出 (端口)	A22 输出 (BSC)	DREQ1 输入 (DMAC)	IRQ1 输入 (INTC)	SCK1 输入 / 输出 (SCI)
	PA7 输入 / 输出 (端口)	$\overline{\text{CS}}3$ 输出 (BSC)	TCLKB 输入 (MTU2)	—	—
	PA8 输入 / 输出 (端口)	RDWR 输出 (BSC)	IRQ2 输入 (INTC)	TCLKC 输入 (MTU2)	—
	PA9 输入 / 输出 (端口)	CKE 输出 (BSC)	IRQ3 输入 (INTC)	TCLKD 输入 (MTU2)	—
	PA10 输入 / 输出 (端口)	$\overline{\text{CS}}0$ 输出 (BSC)	$\overline{\text{POE}}4$ 输入 (POE)	—	—
	PA12 输入 / 输出 (端口)	$\overline{\text{WRL}}/\overline{\text{DQMLL}}$ 输出 (BSC)	$\overline{\text{POE}}6$ 输入 (POE)	—	—
	PA13 输入 / 输出 (端口)	$\overline{\text{WRH}}/\overline{\text{DQMLU}}$ 输出 (BSC)	$\overline{\text{POE}}7$ 输入 (POE)	—	—
	PA14 输入 / 输出 (端口)	$\overline{\text{RD}}$ 输出 (BSC)	—	—	—
	PA15 输入 / 输出 (端口)	CK 输出 (CPG)	—	—	—

表 21.2 多路复用引脚一览表 (SH7084、端口 A)

端口	功能 1 (相关模块)	功能 2 (相关模块)	功能 3 (相关模块)	功能 4 (相关模块)	功能 5 (相关模块)
A	PA0 输入 / 输出 (端口)	$\overline{\text{CS4}}$ 输出 (BSC)	RXD0 输入 (SCI)	—	—
	PA1 输入 / 输出 (端口)	$\overline{\text{CS5}}$ 输出 (BSC)	TXD0 输出 (SCI)	—	—
	PA2 输入 / 输出 (端口)	A25 输出 (BSC)	DREQ0 输入 (DMAC)	IRQ0 输入 (INTC)	SCK0 输入 / 输出 (SCI)
	PA3 输入 / 输出 (端口)	A24 输出 (BSC)	RXD1 输入 (SCI)	—	—
	PA4 输入 / 输出 (端口)	A23 输出 (BSC)	TXD1 输出 (SCI)	—	—
	PA5 输入 / 输出 (端口)	A22 输出 (BSC)	DREQ1 输入 (DMAC)	IRQ1 输入 (INTC)	SCK1 输入 / 输出 (SCI)
	PA6 输入 / 输出 (端口)	$\overline{\text{CS6}}$ 输出 (BSC)	TCLKA 输入 (MTU2)	—	—
	PA7 输入 / 输出 (端口)	$\overline{\text{CS3}}$ 输出 (BSC)	TCLKB 输入 (MTU2)	—	—
	PA8 输入 / 输出 (端口)	RDWR 输出 (BSC)	IRQ2 输入 (INTC)	TCLKC 输入 (MTU2)	—
	PA9 输入 / 输出 (端口)	CKE 输出 (BSC)	IRQ3 输入 (INTC)	TCLKD 输入 (MTU2)	—
	PA10 输入 / 输出 (端口)	$\overline{\text{CS0}}$ 输出 (BSC)	$\overline{\text{POE4}}$ 输入 (POE)	—	—
	PA11 输入 / 输出 (端口)	$\overline{\text{CS1}}$ 输出 (BSC)	$\overline{\text{POE5}}$ 输入 (POE)	—	—
	PA12 输入 / 输出 (端口)	$\overline{\text{WRL/DQMLL}}$ 输出 (BSC)	$\overline{\text{POE6}}$ 输入 (POE)	—	—
	PA13 输入 / 输出 (端口)	$\overline{\text{WRH/DQMLU}}$ 输出 (BSC)	$\overline{\text{POE7}}$ 输入 (POE)	—	—
	PA14 输入 / 输出 (端口)	$\overline{\text{RD}}$ 输出 (BSC)	—	—	—
	PA15 输入 / 输出 (端口)	CK 输出 (CPG)	—	—	—
	PA16 输入 / 输出 (端口)	$\overline{\text{AH}}$ 输出 (BSC)	CKE 输出 (BSC)	—	—
PA17 输入 / 输出 (端口)	$\overline{\text{WAIT}}$ 输入 (BSC)	—	—	—	

表 21.3 多路复用引脚一览表 (SH7085、端口 A)

端口	功能 1 (相关模块)	功能 2 (相关模块)	功能 3 (相关模块)	功能 4 (相关模块)	功能 5 (相关模块)
A	PA0 输入 / 输出 (端口)	$\overline{\text{CS4}}$ 输出 (BSC)	RXD0 输入 (SCI)	—	—
	PA1 输入 / 输出 (端口)	$\overline{\text{CS5/CE1A}}$ 输出 (BSC)	TXD0 输出 (SCI)	—	—
	PA2 输入 / 输出 (端口)	A25 输出 (BSC)	DREQ0 输入 (DMAC)	IRQ0 输入 (INTC)	SCK0 输入 / 输出 (SCI)
	PA3 输入 / 输出 (端口)	A24 输出 (BSC)	RXD1 输入 (SCI)	—	—
	PA4 输入 / 输出 (端口)	A23 输出 (BSC)	TXD1 输出 (SCI)	—	—
	PA5 输入 / 输出 (端口)	A22 输出 (BSC)	DREQ1 输入 (DMAC)	IRQ1 输入 (INTC)	SCK1 输入 / 输出 (SCI)
	PA6 输入 / 输出 (端口)	$\overline{\text{CS2}}$ 输出 (BSC)	TCLKA 输入 (MTU2)	—	—
	PA7 输入 / 输出 (端口)	$\overline{\text{CS3}}$ 输出 (BSC)	TCLKB 输入 (MTU2)	—	—
	PA8 输入 / 输出 (端口)	RDWR 输出 (BSC)	IRQ2 输入 (INTC)	TCLKC 输入 (MTU2)	—
	PA9 输入 / 输出 (端口)	$\overline{\text{FRAME}}$ 输出 (BSC)	CKE 输出 (BSC)	IRQ3 输入 (INTC)	TCLKD 输入 (MTU2)
	PA10 输入 / 输出 (端口)	$\overline{\text{CS0}}$ 输出 (BSC)	$\overline{\text{POE4}}$ 输入 (POE)	—	—
	PA11 输入 / 输出 (端口)	$\overline{\text{CS1}}$ 输出 (BSC)	$\overline{\text{POE5}}$ 输入 (POE)	—	—
	PA12 输入 / 输出 (端口)	$\overline{\text{WRL/DQMLL}}$ 输出 (BSC)	$\overline{\text{POE6}}$ 输入 (POE)	—	—
	PA13 输入 / 输出 (端口)	$\overline{\text{WRH/WE/DQMLU}}$ 输出 (BSC)	$\overline{\text{POE7}}$ 输入 (POE)	—	—
	PA14 输入 / 输出 (端口)	$\overline{\text{RD}}$ 输出 (BSC)	—	—	—
	PA15 输入 / 输出 (端口)	CK 输出 (CPG)	—	—	—
	PA16 输入 / 输出 (端口)	$\overline{\text{WRHH/ICLOWR/AH/}}\overline{\text{DQMUU}}$ 输出 (BSC)	CKE 输出 (BSC)	DREQ2 输入 (DMAC)	$\overline{\text{AUDSYNC}}$ 输出 (AUD) *
	PA17 输入 / 输出 (端口)	$\overline{\text{WAIT}}$ 输入 (BSC)	DACK2 输出 (DMAC)	—	—
	PA18 输入 / 输出 (端口)	$\overline{\text{BREQ}}$ 输入 (BSC)	TEND0 输出 (DMAC)	—	—
PA19 输入 / 输出 (端口)	$\overline{\text{BACK}}$ 输出 (BSC)	TEND1 输出 (DMAC)	—	—	

端口	功能 1 (相关模块)	功能 2 (相关模块)	功能 3 (相关模块)	功能 4 (相关模块)	功能 5 (相关模块)
A	PA20 输入 / 输出 (端口)	$\overline{\text{CS4}}$ 输出 (BSC)	$\overline{\text{RASU}}$ 输出 (BSC)	—	—
	PA21 输入 / 输出 (端口)	$\overline{\text{CS5/CE1A}}$ 输出 (BSC)	$\overline{\text{CASU}}$ 输出 (BSC)	TIC5U 输入 (MTU2)	—
	PA22 输入 / 输出 (端口)	$\overline{\text{WRHL/ICIORD/}}/$ DQMUL 输出 (BSC)	TIC5V 输入 (MTU2)	—	—
	PA23 输入 / 输出 (端口)	$\overline{\text{WRHH/ICIOWR/AH/}}/$ DQMUU 输出 (BSC)	TIC5W 输入 (MTU2)	—	—
	PA24 输入 / 输出 (端口)	$\overline{\text{CE2A}}$ 输出 (BSC)	DREQ3 输入 (DMAC)	—	—
	PA25 输入 / 输出 (端口)	$\overline{\text{CE2B}}$ 输出 (BSC)	DACK3 输出 (DMAC)	$\overline{\text{POE8}}$ 输入 (POE)	—

【注】 \* 只限支持 E10A 全功能的 F-ZTAT 版本。

表 21.4 多路复用引脚一览表 (SH7086、端口 A)

端口	功能 1 (相关模块)	功能 2 (相关模块)	功能 3 (相关模块)	功能 4 (相关模块)	功能 5 (相关模块)
A	PA0 输入 / 输出 (端口)	$\overline{\text{CS4}}$ 输出 (BSC)	RXD0 输入 (SCI)	—	—
	PA1 输入 / 输出 (端口)	$\overline{\text{CS5/CE1A}}$ 输出 (BSC)	TXD0 输出 (SCI)	—	—
	PA2 输入 / 输出 (端口)	A25 输出 (BSC)	DREQ0 输入 (DMAC)	IRQ0 输入 (INTC)	SCK0 输入 / 输出 (SCI)
	PA3 输入 / 输出 (端口)	A24 输出 (BSC)	RXD1 输入 (SCI)	—	—
	PA4 输入 / 输出 (端口)	A23 输出 (BSC)	TXD1 输出 (SCI)	—	—
	PA5 输入 / 输出 (端口)	A22 输出 (BSC)	DREQ1 输入 (DMAC)	IRQ1 输入 (INTC)	SCK1 输入 / 输出 (SCI)
	PA6 输入 / 输出 (端口)	$\overline{\text{CS2}}$ 输出 (BSC)	TCLKA 输入 (MTU2)	—	—
	PA7 输入 / 输出 (端口)	$\overline{\text{CS3}}$ 输出 (BSC)	TCLKB 输入 (MTU2)	—	—
	PA8 输入 / 输出 (端口)	RDWR 输出 (BSC)	IRQ2 输入 (INTC)	TCLKC 输入 (MTU2)	—
	PA9 输入 / 输出 (端口)	$\overline{\text{FRAME}}$ 输出 (BSC)	CKE 输出 (BSC)	IRQ3 输入 (INTC)	TCLKD 输入 (MTU2)
	PA10 输入 / 输出 (端口)	$\overline{\text{CS0}}$ 输出 (BSC)	$\overline{\text{POE4}}$ 输入 (POE)	—	—
	PA11 输入 / 输出 (端口)	$\overline{\text{CS1}}$ 输出 (BSC)	$\overline{\text{POE5}}$ 输入 (POE)	—	—

端口	功能 1 (相关模块)	功能 2 (相关模块)	功能 3 (相关模块)	功能 4 (相关模块)	功能 5 (相关模块)
A	PA12 输入 / 输出 (端口)	$\overline{\text{WRL/DQMLL}}$ 输出 (BSC)	$\overline{\text{POE6}}$ 输入 (POE)	—	—
	PA13 输入 / 输出 (端口)	$\overline{\text{WRH/DQMLU/WE}}$ 输出 (BSC)	$\overline{\text{POE7}}$ 输入 (POE)	—	—
	PA14 输入 / 输出 (端口)	$\overline{\text{RD}}$ 输出 (BSC)	—	—	—
	PA15 输入 / 输出 (端口)	CK 输出 (CPG)	—	—	—
	PA16 输入 / 输出 (端口)	$\overline{\text{WRHH/ICIOWR/AH/}}/$ $\overline{\text{DQMUU}}$ 输出 (BSC)	CKE 输出 (BSC)	DREQ2 输入 (DMAC)	$\overline{\text{AUDSYNC}}$ 输出 (AUD) *
	PA17 输入 / 输出 (端口)	$\overline{\text{WAIT}}$ 输入 (BSC)	DACK2 输出 (DMAC)	—	—
	PA18 输入 / 输出 (端口)	$\overline{\text{BREQ}}$ 输入 (BSC)	TEND0 输出 (DMAC)	—	—
	PA19 输入 / 输出 (端口)	$\overline{\text{BACK}}$ 输出 (BSC)	TEND1 输出 (DMAC)	—	—
	PA20 输入 / 输出 (端口)	$\overline{\text{CS4}}$ 输出 (BSC)	$\overline{\text{RASU}}$ 输出 (BSC)	—	—
	PA21 输入 / 输出 (端口)	$\overline{\text{CS5/CE1A}}$ 输出 (BSC)	$\overline{\text{CASU}}$ 输出 (BSC)	TIC5U 输入 (MTU2)	—
	PA22 输入 / 输出 (端口)	$\overline{\text{WRHL/ICIORD/}}/$ $\overline{\text{DQMUL}}$ 输出 (BSC)	TIC5V 输入 (MTU2)	—	—
	PA23 输入 / 输出 (端口)	$\overline{\text{WRHH/ICIOWR/AH/}}/$ $\overline{\text{DQMUU}}$ 输出 (BSC)	TIC5W 输入 (MTU2)	—	—
	PA24 输入 / 输出 (端口)	$\overline{\text{CE2A}}$ 输出 (BSC)	DREQ3 输入 (DMAC)	—	—
	PA25 输入 / 输出 (端口)	$\overline{\text{CE2B}}$ 输出 (BSC)	DACK3 输出 (DMAC)	$\overline{\text{POE8}}$ 输入 (POE)	—
	PA26 输入 / 输出 (端口)	A26 输出 (BSC)	IRQ0 输入 (INTC)	—	—
	PA27 输入 / 输出 (端口)	A27 输出 (BSC)	IRQ1 输入 (INTC)	—	—
	PA28 输入 / 输出 (端口)	A28 输出 (BSC)	IRQ2 输入 (INTC)	—	—
PA29 输入 / 输出 (端口)	A29 输出 (BSC)	IRQ3 输入 (INTC)	—	—	

【注】 \* 只限支持 E10A 全功能的 F-ZTAT 版本。

表 21.5 多路复用引脚一览表 (SH7083、端口 B)

端口	功能 1 (相关模块)	功能 2 (相关模块)	功能 3 (相关模块)	功能 4 (相关模块)	功能 5 (相关模块)
B	PB0 输入 / 输出 (端口)	A16 输出 (BSC)	TIC5WS 输入 (MTU2S)	—	—
	PB1 输入 / 输出 (端口)	A17 输出 (BSC)	TIC5W 输入 (MTU2)	—	—
	PB2 输入 / 输出 (端口)	IRQ0 输入 (INTC)	$\overline{\text{POE0}}$ 输入 (POE)	—	—
	PB4 输入 / 输出 (端口)	$\overline{\text{RASL}}$ 输出 (BSC)	IRQ2 输入 (INTC)	$\overline{\text{POE2}}$ 输入 (POE)	—
	PB5 输入 / 输出 (端口)	$\overline{\text{CASL}}$ 输出 (BSC)	IRQ3 输入 (INTC)	$\overline{\text{POE3}}$ 输入 (POE)	—
	PB6 输入 / 输出 (端口)	A18 输出 (BSC)	$\overline{\text{BACK}}$ 输出 (BSC)	IRQ4 输入 (INTC)	RXD0 输入 (SCI)
	PB7 输入 / 输出 (端口)	A19 输出 (BSC)	$\overline{\text{BREQ}}$ 输入 (BSC)	IRQ5 输入 (INTC)	TXD0 输出 (SCI)
	PB8 输入 / 输出 (端口)	A20 输出 (BSC)	$\overline{\text{WAIT}}$ 输入 (BSC)	IRQ6 输入 (INTC)	SCK0 输入 / 输出 (SCI)
	PB9 输入 / 输出 (端口)	A21 输出 (BSC)	IRQ7 输入 (INTC)	$\overline{\text{ADTRG}}$ 输入 (A/D)	$\overline{\text{POE8}}$ 输入 (POE)

表 21.6 多路复用引脚一览表 (SH7084/85/86、端口 B)

端口	功能 1 (相关模块)	功能 2 (相关模块)	功能 3 (相关模块)	功能 4 (相关模块)	功能 5 (相关模块)
B	PB0 输入 / 输出 (端口)	A16 输出 (BSC)	TIC5WS 输入 (MTU2S)	—	—
	PB1 输入 / 输出 (端口)	A17 输出 (BSC)	TIC5W 输入 (MTU2)	—	—
	PB2 输入 / 输出 (端口)	IRQ0 输入 (INTC)	$\overline{\text{POE0}}$ 输入 (POE)	SCL 输入 / 输出 (I <sup>2</sup> C2)	—
	PB3 输入 / 输出 (端口)	IRQ1 输入 (INTC)	$\overline{\text{POE1}}$ 输入 (POE)	SDA 输入 / 输出 (I <sup>2</sup> C2)	—
	PB4 输入 / 输出 (端口)	$\overline{\text{RASL}}$ 输出 (BSC)	IRQ2 输入 (INTC)	$\overline{\text{POE2}}$ 输入 (POE)	—
	PB5 输入 / 输出 (端口)	$\overline{\text{CASL}}$ 输出 (BSC)	IRQ3 输入 (INTC)	$\overline{\text{POE3}}$ 输入 (POE)	—
	PB6 输入 / 输出 (端口)	A18 输出 (BSC)	$\overline{\text{BACK}}$ 输出 (BSC)	IRQ4 输入 (INTC)	RXD0 输入 (SCI)
	PB7 输入 / 输出 (端口)	A19 输出 (BSC)	$\overline{\text{BREQ}}$ 输入 (BSC)	IRQ5 输入 (INTC)	TXD0 输出 (SCI)
	PB8 输入 / 输出 (端口)	A20 输出 (BSC)	$\overline{\text{WAIT}}$ 输入 (BSC)	IRQ6 输入 (INTC)	SCK0 输入 / 输出 (SCI)
	PB9 输入 / 输出 (端口)	A21 输出 (BSC)	IRQ7 输入 (INTC)	$\overline{\text{ADTRG}}$ 输入 (A/D)	$\overline{\text{POE8}}$ 输入 (POE)

表 21.7 多路复用引脚一览表 (SH7083/84/85、端口 C)

端口	功能 1 (相关模块)	功能 2 (相关模块)
C	PC0 输入 / 输出 (端口)	A0 输出 (BSC)
	PC1 输入 / 输出 (端口)	A1 输出 (BSC)
	PC2 输入 / 输出 (端口)	A2 输出 (BSC)
	PC3 输入 / 输出 (端口)	A3 输出 (BSC)
	PC4 输入 / 输出 (端口)	A4 输出 (BSC)
	PC5 输入 / 输出 (端口)	A5 输出 (BSC)
	PC6 输入 / 输出 (端口)	A6 输出 (BSC)
	PC7 输入 / 输出 (端口)	A7 输出 (BSC)
	PC8 输入 / 输出 (端口)	A8 输出 (BSC)
	PC9 输入 / 输出 (端口)	A9 输出 (BSC)
	PC10 输入 / 输出 (端口)	A10 输出 (BSC)
	PC11 输入 / 输出 (端口)	A11 输出 (BSC)
	PC12 输入 / 输出 (端口)	A12 输出 (BSC)
	PC13 输入 / 输出 (端口)	A13 输出 (BSC)
	PC14 输入 / 输出 (端口)	A14 输出 (BSC)
PC15 输入 / 输出 (端口)	A15 输出 (BSC)	



表 21.8 多路复用引脚一览表 (SH7086、端口 C)

端口	功能 1 (相关模块)	功能 2 (相关模块)
C	PC0 输入 / 输出 (端口)	A0 输出 (BSC)
	PC1 输入 / 输出 (端口)	A1 输出 (BSC)
	PC2 输入 / 输出 (端口)	A2 输出 (BSC)
	PC3 输入 / 输出 (端口)	A3 输出 (BSC)
	PC4 输入 / 输出 (端口)	A4 输出 (BSC)
	PC5 输入 / 输出 (端口)	A5 输出 (BSC)
	PC6 输入 / 输出 (端口)	A6 输出 (BSC)
	PC7 输入 / 输出 (端口)	A7 输出 (BSC)
	PC8 输入 / 输出 (端口)	A8 输出 (BSC)
	PC9 输入 / 输出 (端口)	A9 输出 (BSC)
	PC10 输入 / 输出 (端口)	A10 输出 (BSC)
	PC11 输入 / 输出 (端口)	A11 输出 (BSC)
	PC12 输入 / 输出 (端口)	A12 输出 (BSC)
	PC13 输入 / 输出 (端口)	A13 输出 (BSC)
	PC14 输入 / 输出 (端口)	A14 输出 (BSC)
	PC15 输入 / 输出 (端口)	A15 输出 (BSC)
	PC18 输入 / 输出 (端口)	A18 输出 (BSC)
	PC19 输入 / 输出 (端口)	A19 输出 (BSC)
	PC20 输入 / 输出 (端口)	A20 输出 (BSC)
	PC21 输入 / 输出 (端口)	A21 输出 (BSC)
	PC22 输入 / 输出 (端口)	A22 输出 (BSC)
	PC23 输入 / 输出 (端口)	A23 输出 (BSC)
	PC24 输入 / 输出 (端口)	A24 输出 (BSC)
	PC25 输入 / 输出 (端口)	A25 输出 (BSC)

表 21.9 多路复用引脚一览表 (SH7083/84、端口 D)

端口	功能 1 (相关模块)	功能 2 (相关模块)	功能 3 (相关模块)	功能 4 (相关模块)
D	PD0 输入 / 输出 (端口)	D0 输入 / 输出 (BSC)	—	—
	PD1 输入 / 输出 (端口)	D1 输入 / 输出 (BSC)	—	—
	PD2 输入 / 输出 (端口)	D2 输入 / 输出 (BSC)	TIC5U 输入 (MTU2)	—
	PD3 输入 / 输出 (端口)	D3 输入 / 输出 (BSC)	TIC5V 输入 (MTU2)	—
	PD4 输入 / 输出 (端口)	D4 输入 / 输出 (BSC)	TIC5W 输入 (MTU2)	—
	PD5 输入 / 输出 (端口)	D5 输入 / 输出 (BSC)	TIC5US 输入 (MTU2S)	—
	PD6 输入 / 输出 (端口)	D6 输入 / 输出 (BSC)	TIC5VS 输入 (MTU2S)	—
	PD7 输入 / 输出 (端口)	D7 输入 / 输出 (BSC)	TIC5WS 输入 (MTU2S)	—
	PD8 输入 / 输出 (端口)	D8 输入 / 输出 (BSC)	TIOC3AS 输入 / 输出 (MTU2S)	AUDATA0 输出 (AUD) *
	PD9 输入 / 输出 (端口)	D9 输入 / 输出 (BSC)	TIOC3BS 输入 / 输出 (MTU2S)	AUDATA1 输出 (AUD) *
	PD10 输入 / 输出 (端口)	D10 输入 / 输出 (BSC)	TIOC3CS 输入 / 输出 (MTU2S)	AUDATA2 输出 (AUD) *
	PD11 输入 / 输出 (端口)	D11 输入 / 输出 (BSC)	TIOC3DS 输入 / 输出 (MTU2S)	AUDATA3 输出 (AUD) *
	PD12 输入 / 输出 (端口)	D12 输入 / 输出 (BSC)	TIOC4AS 输入 / 输出 (MTU2S)	—
	PD13 输入 / 输出 (端口)	D13 输入 / 输出 (BSC)	TIOC4BS 输入 / 输出 (MTU2S)	—
	PD14 输入 / 输出 (端口)	D14 输入 / 输出 (BSC)	TIOC4CS 输入 / 输出 (MTU2S)	AUDCK 输出 (AUD) *
PD15 输入 / 输出 (端口)	D15 输入 / 输出 (BSC)	TIOC4DS 输入 / 输出 (MTU2S)	AUDSYN $\bar{C}$ 输出 (AUD) *	

【注】 \* 只限支持 E10A 全功能的 F-ZTAT 版本。

表 21.10 多路复用引脚一览表 (SH7085/86、端口 D)

端口	功能 1 (相关模块)	功能 2 (相关模块)	功能 3 (相关模块)	功能 4 (相关模块)	功能 5 (相关模块)
D	PD0 输入 / 输出 (端口)	D0 输入 / 输出 (BSC)	—	—	—
	PD1 输入 / 输出 (端口)	D1 输入 / 输出 (BSC)	—	—	—
	PD2 输入 / 输出 (端口)	D2 输入 / 输出 (BSC)	TIC5U 输入 (MTU2)	—	—
	PD3 输入 / 输出 (端口)	D3 输入 / 输出 (BSC)	TIC5V 输入 (MTU2)	—	—
	PD4 输入 / 输出 (端口)	D4 输入 / 输出 (BSC)	TIC5W 输入 (MTU2)	—	—
	PD5 输入 / 输出 (端口)	D5 输入 / 输出 (BSC)	TIC5US 输入 (MTU2S)	—	—
	PD6 输入 / 输出 (端口)	D6 输入 / 输出 (BSC)	TIC5VS 输入 (MTU2S)	—	—
	PD7 输入 / 输出 (端口)	D7 输入 / 输出 (BSC)	TIC5WS 输入 (MTU2S)	—	—
	PD8 输入 / 输出 (端口)	D8 输入 / 输出 (BSC)	TIOC3AS 输入 / 输出 (MTU2S)	—	—
	PD9 输入 / 输出 (端口)	D9 输入 / 输出 (BSC)	TIOC3BS 输入 / 输出 (MTU2S)	—	—
	PD10 输入 / 输出 (端口)	D10 输入 / 输出 (BSC)	TIOC3CS 输入 / 输出 (MTU2S)	—	—
	PD11 输入 / 输出 (端口)	D11 输入 / 输出 (BSC)	TIOC3DS 输入 / 输出 (MTU2S)	—	—
	PD12 输入 / 输出 (端口)	D12 输入 / 输出 (BSC)	TIOC4AS 输入 / 输出 (MTU2S)	—	—
	PD13 输入 / 输出 (端口)	D13 输入 / 输出 (BSC)	TIOC4BS 输入 / 输出 (MTU2S)	—	—
	PD14 输入 / 输出 (端口)	D14 输入 / 输出 (BSC)	TIOC4CS 输入 / 输出 (MTU2S)	—	—
	PD15 输入 / 输出 (端口)	D15 输入 / 输出 (BSC)	TIOC4DS 输入 / 输出 (MTU2S)	—	—
	PD16 输入 / 输出 (端口)	D16 输入 / 输出 (BSC)	IRQ0 输入 (INTC)	$\overline{\text{POE}}4$ 输入 (POE)	AUDATA0 输出 (AUD) *
	PD17 输入 / 输出 (端口)	D17 输入 / 输出 (BSC)	IRQ1 输入 (INTC)	$\overline{\text{POE}}5$ 输入 (POE)	AUDATA1 输出 (AUD) *
	PD18 输入 / 输出 (端口)	D18 输入 / 输出 (BSC)	IRQ2 输入 (INTC)	$\overline{\text{POE}}6$ 输入 (POE)	AUDATA2 输出 (AUD) *
PD19 输入 / 输出 (端口)	D19 输入 / 输出 (BSC)	IRQ3 输入 (INTC)	$\overline{\text{POE}}7$ 输入 (POE)	AUDATA3 输出 (AUD) *	

端口	功能 1 (相关模块)	功能 2 (相关模块)	功能 3 (相关模块)	功能 4 (相关模块)	功能 5 (相关模块)
D	PD20 输入 / 输出 (端口)	D20 输入 / 输出 (BSC)	IRQ4 输入 (INTC)	TIC5WS 输入 (MTU2S)	—
	PD21 输入 / 输出 (端口)	D21 输入 / 输出 (BSC)	IRQ5 输入 (INTC)	TIC5VS 输入 (MTU2S)	—
	PD22 输入 / 输出 (端口)	D22 输入 / 输出 (BSC)	IRQ6 输入 (INTC)	TIC5US 输入 (MTU2S)	AUDCK 输出 (AUD) *
	PD23 输入 / 输出 (端口)	D23 输入 / 输出 (BSC)	IRQ7 输入 (INTC)	$\overline{\text{AUDSYNC}}$ 输出 (AUD) *	—
	PD24 输入 / 输出 (端口)	D24 输入 / 输出 (BSC)	DREQ0 输入 (DMAC)	TIOC4DS 输入 / 输出 (MTU2S)	—
	PD25 输入 / 输出 (端口)	D25 输入 / 输出 (BSC)	DREQ1 输入 (DMAC)	TIOC4CS 输入 / 输出 (MTU2S)	—
	PD26 输入 / 输出 (端口)	D26 输入 / 输出 (BSC)	DACK0 输出 (DMAC)	TIOC4BS 输入 / 输出 (MTU2S)	—
	PD27 输入 / 输出 (端口)	D27 输入 / 输出 (BSC)	DACK1 输出 (DMAC)	TIOC4AS 输入 / 输出 (MTU2S)	—
	PD28 输入 / 输出 (端口)	D28 输入 / 输出 (BSC)	$\overline{\text{CS2}}$ 输出 (BSC)	TIOC3DS 输入 / 输出 (MTU2S)	—
	PD29 输入 / 输出 (端口)	D29 输入 / 输出 (BSC)	$\overline{\text{CS3}}$ 输出 (BSC)	TIOC3BS 输入 / 输出 (MTU2S)	—
	PD30 输入 / 输出 (端口)	D30 输入 / 输出 (BSC)	TIOC3CS 输入 / 输出 (MTU2S)	$\overline{\text{IRQOUT}}$ 输出 (INTC)	—
	PD31 输入 / 输出 (端口)	D31 输入 / 输出 (BSC)	TIOC3AS 输入 / 输出 (MTU2S)	$\overline{\text{ADTRG}}$ 输入 (A/D)	—

【注】 \* 只限支持 E10A 全功能的 F-ZTAT 版本。

表 21.11 多路复用引脚一览表 (SH7083、端口 E)

端口	功能 1 (相关模块)	功能 2 (相关模块)	功能 3 (相关模块)	功能 4 (相关模块)	功能 5 (相关模块)	功能 6 (相关模块)
E	PE0 输入 / 输出 (端口)	DREQ0 输入 (DMAC)	TIOC0A 输入 / 输出 (MTU2)	TMS 输入 (H-UDI) *	—	—
	PE1 输入 / 输出 (端口)	TEND0 输出 (DMAC)	TIOC0B 输入 / 输出 (MTU2)	$\overline{\text{TRST}}$ 输入 (H-UDI) *	—	—
	PE2 输入 / 输出 (端口)	DREQ1 输入 (DMAC)	TIOC0C 输入 / 输出 (MTU2)	TDI 输入 (H-UDI) *	—	—
	PE3 输入 / 输出 (端口)	TEND1 输出 (DMAC)	TIOC0D 输入 / 输出 (MTU2)	TDO 输出 (H-UDI) *	—	—
	PE4 输入 / 输出 (端口)	TIOC1A 输入 / 输出 (MTU2)	RXD3 输入 (SCIF)	TCK 输入 (H-UDI) *	—	—
	PE6 输入 / 输出 (端口)	$\overline{\text{CS7}}$ 输出 (BSC)	TIOC2A 输入 / 输出 (MTU2)	SCK3 输入 / 输出 (SCIF)	—	—
	PE7 输入 / 输出 (端口)	$\overline{\text{BS}}$ 输出 (BSC)	TIOC2B 输入 / 输出 (MTU2)	$\overline{\text{UBCTRG}}$ 输出 (UBC)	RXD2 输入 (SCI)	SSI 输入 / 输出 (SSU)
	PE8 输入 / 输出 (端口)	TIOC3A 输入 / 输出 (MTU2)	SCK2 输入 / 输出 (SCI)	SSCK 输入 / 输出 (SSU)	—	—
	PE10 输入 / 输出 (端口)	TIOC3C 输入 / 输出 (MTU2)	TXD2 输出 (SCI)	SSO 输入 / 输出 (SSU)	—	—
	PE12 输入 / 输出 (端口)	TIOC4A 输入 / 输出 (MTU2)	TXD3 输出 (SCIF)	$\overline{\text{SCS}}$ 输入 / 输出 (SSU)	—	—
	PE13 输入 / 输出 (端口)	TIOC4B 输入 / 输出 (MTU2)	$\overline{\text{MRES}}$ 输入 (INTC)	$\overline{\text{ASEBRKAK}}$ 输出 (E10A) *	$\overline{\text{ASEBRK}}$ 输入 (E10A) *	—
	PE14 输入 / 输出 (端口)	DACK0 输出 (DMAC)	TIOC4C 输入 / 输出 (MTU2)	—	—	—
	PE15 输入 / 输出 (端口)	CKE 输出 (BSC)	DACK1 输出 (DMAC)	TIOC4D 输入 / 输出 (MTU2)	$\overline{\text{IRQOUT}}$ 输出 (INTC)	—

【注】 \* 只限 F-ZTAT 版。

表 21.12 多路复用引脚一览表 (SH7084、端口 E)

端口	功能 1 (相关模块)	功能 2 (相关模块)	功能 3 (相关模块)	功能 4 (相关模块)	功能 5 (相关模块)	功能 6 (相关模块)
E	PE0 输入 / 输出 (端口)	DREQ0 输入 (DMAC)	TIOC0A 输入 / 输出 (MTU2)	TMS 输入 (H-UDI) *	—	—
	PE1 输入 / 输出 (端口)	TEND0 输出 (DMAC)	TIOC0B 输入 / 输出 (MTU2)	$\overline{\text{TRST}}$ 输入 (H-UDI) *	—	—
	PE2 输入 / 输出 (端口)	DREQ1 输入 (DMAC)	TIOC0C 输入 / 输出 (MTU2)	TDI 输入 (H-UDI) *	—	—
	PE3 输入 / 输出 (端口)	TEND1 输出 (DMAC)	TIOC0D 输入 / 输出 (MTU2)	TDO 输出 (H-UDI) *	—	—
	PE4 输入 / 输出 (端口)	TIOC1A 输入 / 输出 (MTU2)	RXD3 输入 (SCIF)	TCK 输入 (H-UDI) *	—	—
	PE5 输入 / 输出 (端口)	$\overline{\text{CS6}}$ 输出 (BSC)	TIOC1B 输入 / 输出 (MTU2)	TXD3 输出 (SCIF)	$\overline{\text{ASEBRKAK}}$ 输出 (E10A) *	$\overline{\text{ASEBRK}}$ 输入 (E10A) *
	PE6 输入 / 输出 (端口)	$\overline{\text{CS7}}$ 输出 (BSC)	TIOC2A 输入 / 输出 (MTU2)	SCK3 输入 / 输出 (SCIF)	—	—
	PE7 输入 / 输出 (端口)	$\overline{\text{BS}}$ 输出 (BSC)	TIOC2B 输入 / 输出 (MTU2)	$\overline{\text{UBCTRG}}$ 输出 (UBC)	RXD2 输入 (SCI)	SSI 输入 / 输出 (SSU)
	PE8 输入 / 输出 (端口)	TIOC3A 输入 / 输出 (MTU2)	SCK2 输入 / 输出 (SCI)	SSCK 输入 / 输出 (SSU)	—	—
	PE9 输入 / 输出 (端口)	TIOC3B 输入 / 输出 (MTU2)	SCK3 输入 / 输出 (SCIF)	$\overline{\text{RTS3}}$ 输出 (SCIF)	—	—
	PE10 输入 / 输出 (端口)	TIOC3C 输入 / 输出 (MTU2)	TXD2 输出 (SCI)	SSO 输入 / 输出 (SSU)	—	—
	PE11 输入 / 输出 (端口)	TIOC3D 输入 / 输出 (MTU2)	RXD3 输入 (SCIF)	$\overline{\text{CTS3}}$ 输入 (SCIF)	—	—
	PE12 输入 / 输出 (端口)	TIOC4A 输入 / 输出 (MTU2)	TXD3 输出 (SCIF)	$\overline{\text{SCS}}$ 输入 / 输出 (SSU)	—	—
	PE13 输入 / 输出 (端口)	TIOC4B 输入 / 输出 (MTU2)	$\overline{\text{MRES}}$ 输入 (INTC)	—	—	—
	PE14 输入 / 输出 (端口)	$\overline{\text{AH}}$ 输出 (BSC)	DACK0 输出 (DMAC)	TIOC4C 输入 / 输出 (MTU2)	—	—
	PE15 输入 / 输出 (端口)	CKE 输出 (BSC)	DACK1 输出 (DMAC)	TIOC4D 输入 / 输出 (MTU2)	$\overline{\text{IRQOUT}}$ 输出 (INTC)	—

【注】\* 只限 F-ZTAT 版。

表 21.13 多路复用引脚一览表 (SH7085、端口 E)

端口	功能 1 (相关模块)	功能 2 (相关模块)	功能 3 (相关模块)	功能 4 (相关模块)	功能 5 (相关模块)	功能 6 (相关模块)
E	PE0 输入 / 输出 (端口)	DREQ0 输入 (DMAC)	TIOC0A 输入 / 输出 (MTU2)	AUDCK 输出 (AUD) *1	—	—
	PE1 输入 / 输出 (端口)	TEND0 输出 (DMAC)	TIOC0B 输入 / 输出 (MTU2)	—	—	—
	PE2 输入 / 输出 (端口)	DREQ1 输入 (DMAC)	TIOC0C 输入 / 输出 (MTU2)	—	—	—
	PE3 输入 / 输出 (端口)	TEND1 输出 (DMAC)	TIOC0D 输入 / 输出 (MTU2)	AUDATA3 输出 (AUD) *1	—	—
	PE4 输入 / 输出 (端口)	IOIS16 输入 (BSC)	TIOC1A 输入 / 输出 (MTU2)	RXD3 输入 (SCIF)	AUDATA2 输出 (AUD) *1	—
	PE5 输入 / 输出 (端口)	CS6/CE1B 输出 (BSC)	TIOC1B 输入 / 输出 (MTU2)	TXD3 输出 (SCIF)	AUDATA1 输出 (AUD) *1	—
	PE6 输入 / 输出 (端口)	CS7 输出 (BSC)	TIOC2A 输入 / 输出 (MTU2)	SCK3 输入 / 输出 (SCIF)	AUDATA0 输出 (AUD) *1	—
	PE7 输入 / 输出 (端口)	BS 输出 (BSC)	TIOC2B 输入 / 输出 (MTU2)	UBCTRG 输出 (UBC)	RXD2 输入 (SCI)	SSI 输入 / 输出 (SSU)
	PE8 输入 / 输出 (端口)	TIOC3A 输入 / 输出 (MTU2)	SCK2 输入 / 输出 (SCI)	SSCK 输入 / 输出 (SSU)	TMS 输入 (H-UDI) *2	—
	PE9 输入 / 输出 (端口)	TIOC3B 输入 / 输出 (MTU2)	SCK3 输入 / 输出 (SCIF)	RTS3 输出 (SCIF)	TRST 输入 (H-UDI) *2	—
	PE10 输入 / 输出 (端口)	TIOC3C 输入 / 输出 (MTU2)	TXD2 输出 (SCI)	SSO 输入 / 输出 (SSU)	TDI 输入 (H-UDI) *2	—
	PE11 输入 / 输出 (端口)	TIOC3D 输入 / 输出 (MTU2)	RXD3 输入 (SCIF)	CTS3 输入 (SCIF)	TDO 输出 (H-UDI) *2	—
	PE12 输入 / 输出 (端口)	TIOC4A 输入 / 输出 (MTU2)	TXD3 输出 (SCIF)	SCS 输入 / 输出 (SSU)	TCK 输入 (H-UDI) *2	—
	PE13 输入 / 输出 (端口)	TIOC4B 输入 / 输出 (MTU2)	MRES 输入 (INTC)	ASEBRKAK 输出 (E10A) *2	ASEBRK 输入 (E10A) *2	—
	PE14 输入 / 输出 (端口)	WRHH/ICIOWR/ AH/DQMUU 输出 (BSC)	DACK0 输出 (DMAC)	TIOC4C 输入 / 输出 (MTU2)	—	—
PE15 输入 / 输出 (端口)	CKE 输出 (BSC)	DACK1 输出 (DMAC)	TIOC4D 输入 / 输出 (MTU2)	IRQOUT 输出 (INTC)	—	

【注】 \*1 只限支持 E10A 全功能的 F-ZTAT 版本。

\*2 只限 F-ZTAT 版。

表 21.14 多路复用引脚一览表 (SH7086、端口 E)

端口	功能 1 (相关模块)	功能 2 (相关模块)	功能 3 (相关模块)	功能 4 (相关模块)	功能 5 (相关模块)	功能 6 (相关模块)
E	PE0 输入 / 输出 (端口)	DREQ0 输入 (DMAC)	TIOC0A 输入 / 输出 (MTU2)	AUDCK 输出 (AUD) *1	—	—
	PE1 输入 / 输出 (端口)	TEND0 输出 (DMAC)	TIOC0B 输入 / 输出 (MTU2)	—	—	—
	PE2 输入 / 输出 (端口)	DREQ1 输入 (DMAC)	TIOC0C 输入 / 输出 (MTU2)	—	—	—
	PE3 输入 / 输出 (端口)	TEND1 输出 (DMAC)	TIOC0D 输入 / 输出 (MTU2)	AUDATA3 输出 (AUD) *1	—	—
	PE4 输入 / 输出 (端口)	$\overline{\text{IOIS16}}$ 输入 (BSC)	TIOC1A 输入 / 输出 (MTU2)	RXD3 输入 (SCIF)	AUDATA2 输出 (AUD) *1	—
	PE5 输入 / 输出 (端口)	$\overline{\text{CS6/CE1B}}$ 输出 (BSC)	TIOC1B 输入 / 输出 (MTU2)	TXD3 输出 (SCIF)	AUDATA1 输出 (AUD) *1	—
	PE6 输入 / 输出 (端口)	$\overline{\text{CS7}}$ 输出 (BSC)	TIOC2A 输入 / 输出 (MTU2)	SCK3 输入 / 输出 (SCIF)	AUDATA0 输出 (AUD) *1	—
	PE7 输入 / 输出 (端口)	$\overline{\text{BS}}$ 输出 (BSC)	TIOC2B 输入 / 输出 (MTU2)	$\overline{\text{UBCTR}}\overline{\text{G}}$ 输出 (UBC)	RXD2 输入 (SCI)	SSI 输入 / 输出 (SSU)
	PE8 输入 / 输出 (端口)	TIOC3A 输入 / 输出 (MTU2)	SCK2 输入 / 输出 (SCI)	SSCK 输入 / 输出 (SSU)	TMS 输入 (H-UDI) *2	—
	PE9 输入 / 输出 (端口)	TIOC3B 输入 / 输出 (MTU2)	SCK3 输入 / 输出 (SCIF)	$\overline{\text{RTS3}}$ 输出 (SCIF)	$\overline{\text{TRST}}$ 输入 (H-UDI) *2	—
	PE10 输入 / 输出 (端口)	TIOC3C 输入 / 输出 (MTU2)	TXD2 输出 (SCI)	SSO 输入 / 输出 (SSU)	TDI 输入 (H-UDI) *2	—
	PE11 输入 / 输出 (端口)	TIOC3D 输入 / 输出 (MTU2)	RXD3 输入 (SCIF)	$\overline{\text{CTS3}}$ 输入 (SCIF)	TDO 输出 (H-UDI) *2	—
	PE12 输入 / 输出 (端口)	TIOC4A 输入 / 输出 (MTU2)	TXD3 输出 (SCIF)	$\overline{\text{SCS}}$ 输入 / 输出 (SSU)	TCK 输入 (H-UDI) *2	—
	PE13 输入 / 输出 (端口)	TIOC4B 输入 / 输出 (MTU2)	$\overline{\text{MRES}}$ 输入 (INTC)	$\overline{\text{ASEBRKAK}}$ 输出 (E10A) *2	$\overline{\text{ASEBRK}}$ 输入 (E10A) *2	—
	PE14 输入 / 输出 (端口)	$\overline{\text{WRHH/ICIOWR/}}$ $\overline{\text{AH/DQMUJ}}$ 输出 (BSC)	DACK0 输出 (DMAC)	TIOC4C 输入 / 输出 (MTU2)	—	—
	PE15 输入 / 输出 (端口)	CKE 输出 (BSC)	DACK1 输出 (DMAC)	TIOC4D 输入 / 输出 (MTU2)	$\overline{\text{IRQOUT}}$ 输出 (INTC)	—
	PE16 输入 / 输出 (端口)	$\overline{\text{CS8}}$ 输出 (BSC)	TIOC3BS 输入 / 输出 (MTU2S)	—	—	—
	PE17 输入 / 输出 (端口)	TIOC3DS 输入 / 输出 (MTU2S)	—	—	—	—
	PE18 输入 / 输出 (端口)	TIOC4AS 输入 / 输出 (MTU2S)	—	—	—	—



端口	功能 1 (相关模块)	功能 2 (相关模块)	功能 3 (相关模块)	功能 4 (相关模块)	功能 5 (相关模块)	功能 6 (相关模块)
E	PE19 输入 / 输出 (端口)	TIOC4BS 输入 / 输出 (MTU2S)	—	—	—	—
	PE20 输入 / 输出 (端口)	TIOC4CS 输入 / 输出 (MTU2S)	—	—	—	—
	PE21 输入 / 输出 (端口)	TIOC4DS 输入 / 输出 (MTU2S)	—	—	—	—

【注】 \*1 只限支持 E10A 全功能的 F-ZTAT 版本。

\*2 只限 F-ZTAT 版。

表 21.15 多路复用引脚一览表 (SH7083/84/85、端口 F)

端口	功能 1 (相关模块)	功能 2 (相关模块)
F	PF0 输入 (端口)	AN0 输入 (A/D)
	PF1 输入 (端口)	AN1 输入 (A/D)
	PF2 输入 (端口)	AN2 输入 (A/D)
	PF3 输入 (端口)	AN3 输入 (A/D)
	PF4 输入 (端口)	AN4 输入 (A/D)
	PF5 输入 (端口)	AN5 输入 (A/D)
	PF6 输入 (端口)	AN6 输入 (A/D)
	PF7 输入 (端口)	AN7 输入 (A/D)

【注】 在 A/D 转换中 AN 输入功能有效。

表 21.16 多路复用引脚一览表 (SH7086、端口 F)

端口	功能 1 (相关模块)	功能 2 (相关模块)
F	PF0 输入 (端口)	AN0 输入 (A/D)
	PF1 输入 (端口)	AN1 输入 (A/D)
	PF2 输入 (端口)	AN2 输入 (A/D)
	PF3 输入 (端口)	AN3 输入 (A/D)
	PF4 输入 (端口)	AN4 输入 (A/D)
	PF5 输入 (端口)	AN5 输入 (A/D)
	PF6 输入 (端口)	AN6 输入 (A/D)
	PF7 输入 (端口)	AN7 输入 (A/D)
	PF8 输入 (端口)	AN8 输入 (A/D)
	PF9 输入 (端口)	AN9 输入 (A/D)
	PF10 输入 (端口)	AN10 输入 (A/D)
	PF11 输入 (端口)	AN11 输入 (A/D)
	PF12 输入 (端口)	AN12 输入 (A/D)
	PF13 输入 (端口)	AN13 输入 (A/D)
	PF14 输入 (端口)	AN14 输入 (A/D)
	PF15 输入 (端口)	AN15 输入 (A/D)

【注】 在 A/D 转换中 AN 输入功能有效。

表 21.17 按运行模式分类的引脚功能一览表 (SH7083 (1))

引脚号	引脚名			
	内部 ROM 无效 (MCU 模式 0)		内部 ROM 无效 (MCU 模式 1)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
19,32,55,71,94	Vcc	Vcc	Vcc	Vcc
3,24,33,52,62,81,93	Vss	Vss	Vss	Vss
20,72	VCL	VCL	VCL	VCL
92	AVcc	AVcc	AVcc	AVcc
88	AVss	AVss	AVss	AVss
91	AVref	AVref	AVref	AVref
73	PLLvss	PLLvss	PLLvss	PLLvss
65	EXTAL	EXTAL	EXTAL	EXTAL
63	XTAL	XTAL	XTAL	XTAL
66	MD0	MD0	MD0	MD0
64	MD1	MD1	MD1	MD1
68	FWE	FWE	FWE	FWE

引脚号	引脚名			
	内部 ROM 无效 (MCU 模式 0)		内部 ROM 无效 (MCU 模式 1)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
75	RES	RES	RES	RES
29	WDTOVF	WDTOVF	WDTOVF	WDTOVF
67	NMI	NMI	NMI	NMI
27	ASEMD0	ASEMD0	ASEMD0	ASEMD0
40	PA3	PA3/A24/RXD1	PA3	PA3/A24/RXD1
39	PA4	PA4/A23/TXD1	PA4	PA4/A23/TXD1
38	PA5	PA5/A22/DREQ1/IRQ1/SCK1	PA5	PA5/A22/DREQ1/IRQ1/SCK1
37	PA7	PA7/CS3/TCLKB	PA7	PA7/CS3/TCLKB
36	PA8	PA8/RDWR/IRQ2/TCLKC	PA8	PA8/RDWR/IRQ2/TCLKC
35	PA9	PA9/CKE/IRQ3/TCLKD	PA9	PA9/CKE/IRQ3/TCLKD
34	CS0	PA10/CS0/POE4	CS0	PA10/CS0/POE4
31	WRL	PA12/WRL/DQMLL/POE6	WRL	PA12/WRL/DQMLL/POE6
30	WRH	PA13/WRH/DQMLU/POE7	WRH	PA13/WRH/DQMLU/POE7
28	RD	PA14/RD	RD	PA14/RD
74	CK	PA15/CK	CK	PA15/CK
22	A16	PB0/A16/TIC5WS	A16	PB0/A16/TIC5WS
23	A17	PB1/A17/TIC5W	A17	PB1/A17/TIC5W
25	PB2	PB2/IRQ0/POE0	PB2	PB2/IRQ0/POE0
70	PB4	PB4/RASL/IRQ2/POE2	PB4	PB4/RASL/IRQ2/POE2
69	PB5	PB5/CASL/IRQ3/POE3	PB5	PB5/CASL/IRQ3/POE3
43	PB6	PB6/A18/BACK/IRQ4/RXD0	PB6	PB6/A18/BACK/IRQ4/RXD0
42	PB7	PB7/A19/BREQ/IRQ5/TXD0	PB7	PB7/A19/BREQ/IRQ5/TXD0
41	PB8	PB8/A20/WAIT/IRQ6/SCK0	PB8	PB8/A20/WAIT/IRQ6/SCK0
26	PB9	PB9/A21/IRQ7/ADTRG/POE8	PB9	PB9/A21/IRQ7/ADTRG/POE8
4	A0	PC0/A0	A0	PC0/A0
5	A1	PC1/A1	A1	PC1/A1
6	A2	PC2/A2	A2	PC2/A2
7	A3	PC3/A3	A3	PC3/A3
8	A4	PC4/A4	A4	PC4/A4
9	A5	PC5/A5	A5	PC5/A5
10	A6	PC6/A6	A6	PC6/A6
11	A7	PC7/A7	A7	PC7/A7
12	A8	PC8/A8	A8	PC8/A8
13	A9	PC9/A9	A9	PC9/A9
14	A10	PC10/A10	A10	PC10/A10

引脚号	引脚名			
	内部 ROM 无效 (MCU 模式 0)		内部 ROM 无效 (MCU 模式 1)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
15	A11	PC11/A11	A11	PC11/A11
16	A12	PC12/A12	A12	PC12/A12
17	A13	PC13/A13	A13	PC13/A13
18	A14	PC14/A14	A14	PC14/A14
21	A15	PC15/A15	A15	PC15/A15
61	D0	PD0/D0	D0	PD0/D0
60	D1	PD1/D1	D1	PD1/D1
59	D2	PD2/D2/TIC5U	D2	PD2/D2/TIC5U
58	D3	PD3/D3/TIC5V	D3	PD3/D3/TIC5V
57	D4	PD4/D4/TIC5W	D4	PD4/D4/TIC5W
56	D5	PD5/D5/TIC5US	D5	PD5/D5/TIC5US
54	D6	PD6/D6/TIC5VS	D6	PD6/D6/TIC5VS
53	D7	PD7/D7/TIC5WS	D7	PD7/D7/TIC5WS
51	PD8/(AUDATA0*2)	PD8/D8/TIOC3AS	D8/(AUDATA0*2)	PD8/D8/TIOC3AS
50	PD9/(AUDATA1*2)	PD9/D9/TIOC3BS	D9/(AUDATA1*2)	PD9/D9/TIOC3BS
49	PD10/(AUDATA2*2)	PD10/D10/TIOC3CS	D10/(AUDATA2*2)	PD10/D10/TIOC3CS
48	PD11/(AUDATA3*2)	PD11/D11/TIOC3DS	D11/(AUDATA3*2)	PD11/D11/TIOC3DS
47	PD12	PD12/D12/TIOC4AS	D12	PD12/D12/TIOC4AS
46	PD13	PD13/D13/TIOC4BS	D13	PD13/D13/TIOC4BS
45	PD14/(AUDCK*2)	PD14/D14/TIOC4CS	D14/(AUDCK*2)	PD14/D14/TIOC4CS
44	PD15/(AUDSYNC*2)	PD15/D15/TIOC4DS	D15/(AUDSYNC*2)	PD15/D15/TIOC4DS
76	PE0/(TMS*1)	PE0/DREQ0/TIOC0A	PE0/(TMS*1)	PE0/DREQ0/TIOC0A
77	PE1/(TRST*1)	PE1/TEND0/TIOC0B	PE1/(TRST*1)	PE1/TEND0/TIOC0B
78	PE2/(TDI*1)	PE2/DREQ1/TIOC0C	PE2/(TDI*1)	PE2/DREQ1/TIOC0C
79	PE3/(TDO*1)	PE3/TEND1/TIOC0D	PE3/(TDO*1)	PE3/TEND1/TIOC0D
80	PE4/(TCK*1)	PE4/TIOC1A/RXD3	PE4/(TCK*1)	PE4/TIOC1A/RXD3
95	PE6	PE6/ $\overline{CS7}$ /TIOC2A/SCK3	PE6	PE6/ $\overline{CS7}$ /TIOC2A/SCK3
96	PE7	PE7/ $\overline{BS}$ /TIOC2B/ $\overline{UBCTR\overline{G}}$ / RXD2/SSI	PE7	PE7/ $\overline{BS}$ /TIOC2B/ $\overline{UBCTR\overline{G}}$ / RXD2/SSI
97	PE8	PE8/TIOC3A/SCK2/SSCK	PE8	PE8/TIOC3A/SCK2/SSCK
98	PE10	PE10/TIOC3C/TXD2/SSO	PE10	PE10/TIOC3C/TXD2/SSO
99	PE12	PE12/TIOC4A/TXD3/ $\overline{SCS}$	PE12	PE12/TIOC4A/TXD3/ $\overline{SCS}$
100	PE13/( $\overline{ASEBRKAK}$ / $\overline{ASEBRK}$ *1)	PE13/TIOC4B/ $\overline{MRES}$	PE13/( $\overline{ASEBRKAK}$ / $\overline{ASEBRK}$ *1)	PE13/TIOC4B/ $\overline{MRES}$
1	PE14	PE14/DACK0/TIOC4C	PE14	PE14/DACK0/TIOC4C

引脚号	引脚名			
	内部 ROM 无效 (MCU 模式 0)		内部 ROM 无效 (MCU 模式 1)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
2	PE15	PE15/CKE/DACK1/TIOC4D /IRQOUT	PE15	PE15/CKE/DACK1/TIOC4D /IRQOUT
82	PF0/AN0	PF0/AN0	PF0/AN0	PF0/AN0
83	PF1/AN1	PF1/AN1	PF1/AN1	PF1/AN1
84	PF2/AN2	PF2/AN2	PF2/AN2	PF2/AN2
85	PF3/AN3	PF3/AN3	PF3/AN3	PF3/AN3
86	PF4/AN4	PF4/AN4	PF4/AN4	PF4/AN4
87	PF5/AN5	PF5/AN5	PF5/AN5	PF5/AN5
89	PF6/AN6	PF6/AN6	PF6/AN6	PF6/AN6
90	PF7/AN7	PF7/AN7	PF7/AN7	PF7/AN7

【注】 \*1 当使用 E10A (ASEMD0=Low 电平) 时, 这些引脚被固定为 TMS、TRST、TDI、TDO、TCK、ASEBRKAK/ASEBRK。

\*2 只限支持 E10A 全功能的 F-ZTAT 版本。当使用 E10A 的 AUD 功能时, 这些引脚被固定为 AUD 功能。

表 21.17 按运行模式分类的引脚功能一览表 (SH7083 (2))

引脚号	引脚名			
	内部 ROM 有效 (MCU 模式 2)		单芯片模式 (MCU 模式 3)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
19,32,55, 71,94	Vcc	Vcc	Vcc	Vcc
3,24,33,52, 62,81,93	Vss	Vss	Vss	Vss
20,72	Vcl	Vcl	Vcl	Vcl
92	AVcc	AVcc	AVcc	AVcc
88	AVss	AVss	AVss	AVss
91	AVref	AVref	AVref	AVref
73	PLLvss	PLLvss	PLLvss	PLLvss
65	EXTAL	EXTAL	EXTAL	EXTAL
63	XTAL	XTAL	XTAL	XTAL
66	MD0	MD0	MD0	MD0
64	MD1	MD1	MD1	MD1
68	FWE	FWE	FWE	FWE
75	RES	RES	RES	RES
29	WDTOVF	WDTOVF	WDTOVF	WDTOVF
67	NMI	NMI	NMI	NMI
27	ASEMD0	ASEMD0	ASEMD0	ASEMD0

引脚号	引脚名			
	内部 ROM 有效 (MCU 模式 2)		单芯片模式 (MCU 模式 3)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
40	PA3	PA3/A24/RXD1	PA3	PA3/RXD1
39	PA4	PA4/A23/TXD1	PA4	PA4/TXD1
38	PA5	PA5/A22/DREQ1/IRQ1/SCK1	PA5	PA5/DREQ1/IRQ1/SCK1
37	PA7	PA7/ $\overline{\text{CS3}}$ /TCLKB	PA7	PA7/TCLKB
36	PA8	PA8/RDWR/IRQ2/TCLKC	PA8	PA8/IRQ2/TCLKC
35	PA9	PA9/CKE/IRQ3/TCLKD	PA9	PA9/IRQ3/TCLKD
34	PA10	PA10/ $\overline{\text{CS0}}$ /POE4	PA10	PA10/POE4
31	PA12	PA12/ $\overline{\text{WRL}}$ /DQMLL/POE6	PA12	PA12/POE6
30	PA13	PA13/ $\overline{\text{WRH}}$ /DQMLU/POE7	PA13	PA13/POE7
28	PA14	PA14/ $\overline{\text{RD}}$	PA14	PA14
74	CK	PA15/CK	PA15	PA15
22	PB0	PB0/A16/TIC5WS	PB0	PB0/TIC5WS
23	PB1	PB1/A17/TIC5W	PB1	PB1/TIC5W
25	PB2	PB2/IRQ0/ $\overline{\text{POE0}}$	PB2	PB2/IRQ0/ $\overline{\text{POE0}}$
70	PB4	PB4/ $\overline{\text{RASL}}$ /IRQ2/ $\overline{\text{POE2}}$	PB4	PB4/IRQ2/ $\overline{\text{POE2}}$
69	PB5	PB5/ $\overline{\text{CASL}}$ /IRQ3/ $\overline{\text{POE3}}$	PB5	PB5/IRQ3/ $\overline{\text{POE3}}$
43	PB6	PB6/A18/ $\overline{\text{BACK}}$ /IRQ4/RXD0	PB6	PB6/IRQ4/RXD0
42	PB7	PB7/A19/ $\overline{\text{BREQ}}$ /IRQ5/TXD0	PB7	PB7/IRQ5/TXD0
41	PB8	PB8/A20/ $\overline{\text{WAIT}}$ /IRQ6/SCK0	PB8	PB8/IRQ6/SCK0
26	PB9	PB9/A21/IRQ7/ $\overline{\text{ADTRG}}$ / $\overline{\text{POE8}}$	PB9	PB9/IRQ7/ $\overline{\text{ADTRG}}$ / $\overline{\text{POE8}}$
4	PC0	PC0/A0	PC0	PC0
5	PC1	PC1/A1	PC1	PC1
6	PC2	PC2/A2	PC2	PC2
7	PC3	PC3/A3	PC3	PC3
8	PC4	PC4/A4	PC4	PC4
9	PC5	PC5/A5	PC5	PC5
10	PC6	PC6/A6	PC6	PC6
11	PC7	PC7/A7	PC7	PC7
12	PC8	PC8/A8	PC8	PC8
13	PC9	PC9/A9	PC9	PC9
14	PC10	PC10/A10	PC10	PC10/
15	PC11	PC11/A11	PC11	PC11
16	PC12	PC12/A12	PC12	PC12
17	PC13	PC13/A13	PC13	PC13
18	PC14	PC14/A14	PC14	PC14

引脚号	引脚名			
	内部 ROM 有效 (MCU 模式 2)		单芯片模式 (MCU 模式 3)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
21	PC15	PC15/A15	PC15	PC15
61	PD0	PD0/D0	PD0	PD0
60	PD1	PD1/D1	PD1	PD1
59	PD2	PD2/D2/TIC5U	PD2	PD2/TIC5U
58	PD3	PD3/D3/TIC5V	PD3	PD3/TIC5V
57	PD4	PD4/D4/TIC5W	PD4	PD4/TIC5W
56	PD5	PD5/D5/TIC5US	PD5	PD5/TIC5US
54	PD6	PD6/D6/TIC5VS	PD6	PD6/TIC5VS
53	PD7	PD7/D7/TIC5WS	PD7	PD7/TIC5WS
51	PD8/(AUDATA0*2)	PD8/D8/TIOC3AS	PD8/(AUDATA0*2)	PD8/TIOC3AS
50	PD9/(AUDATA1*2)	PD9/D9/TIOC3BS	PD9/(AUDATA1*2)	PD9/TIOC3BS
49	PD10/(AUDATA2*2)	PD10/D10/TIOC3CS	PD10/(AUDATA2*2)	PD10/TIOC3CS
48	PD11/(AUDATA3*2)	PD11/D11/TIOC3DS	PD11/(AUDATA3*2)	PD11/TIOC3DS
47	PD12	PD12/D12/TIOC4AS	PD12	PD12/TIOC4AS
46	PD13	PD13/D13/TIOC4BS	PD13	PD13/TIOC4BS
45	PD14/(AUDCK*2)	PD14/D14/TIOC4CS	PD14/(AUDCK*2)	PD14/TIOC4CS
44	PD15/(AUDSYNC*2)	PD15/D15/TIOC4DS	PD15/(AUDSYNC*2)	PD15/TIOC4DS
76	PE0/(TMS*1)	PE0/DREQ0/TIOC0A	PE0/(TMS*1)	PE0/DREQ0/TIOC0A
77	PE1/(TRST*1)	PE1/TEND0/TIOC0B	PE1/(TRST*1)	PE1/TIOC0B
78	PE2/(TDI*1)	PE2/DREQ1/TIOC0C	PE2/(TDI*1)	PE2/DREQ1/TIOC0C
79	PE3/(TDO*1)	PE3/TEND1/TIOC0D	PE3/(TDO*1)	PE3/TIOC0D
80	PE4/(TCK*1)	PE4/TIOC1A/RXD3	PE4/(TCK*1)	PE4/TIOC1A/RXD3
95	PE6	PE6/ $\overline{\text{CS7}}$ /TIOC2A/SCK3	PE6	PE6/TIOC2A/SCK3
96	PE7	PE7/ $\overline{\text{BS}}$ /TIOC2B/ $\overline{\text{UBCTRG}}$ / RXD2/SSI	PE7	PE7/TIOC2B/ $\overline{\text{UBCTRG}}$ / RXD2/SSI
97	PE8	PE8/TIOC3A/SCK2/SSCK	PE8	PE8/TIOC3A/SCK2/SSCK
98	PE10	PE10/TIOC3C/TXD2/SSO	PE10	PE10/TIOC3C/TXD2/SSO
99	PE12	PE12/TIOC4A/TXD3/ $\overline{\text{SCS}}$	PE12	PE12/TIOC4A/TXD3/ $\overline{\text{SCS}}$
100	PE13/( $\overline{\text{ASEBRKAK}}$ / $\overline{\text{ASEBRK}}$ *1)	PE13/TIOC4B/ $\overline{\text{MRES}}$	PE13/( $\overline{\text{ASEBRKAK}}$ / $\overline{\text{ASEBRK}}$ *1)	PE13/TIOC4B/ $\overline{\text{MRES}}$
1	PE14	PE14/DACK0/TIOC4C	PE14	PE14/TIOC4C
2	PE15	PE15/CKE/DACK1/TIOC4D $\overline{\text{IRQOUT}}$	PE15	PE15/TIOC4D $\overline{\text{IRQOUT}}$
82	PF0/AN0	PF0/AN0	PF0/AN0	PF0/AN0
83	PF1/AN1	PF1/AN1	PF1/AN1	PF1/AN1

引脚号	引脚名			
	内部 ROM 有效 (MCU 模式 2)		单芯片模式 (MCU 模式 3)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
84	PF2/AN2	PF2/AN2	PF2/AN2	PF2/AN2
85	PF3/AN3	PF3/AN3	PF3/AN3	PF3/AN3
86	PF4/AN4	PF4/AN4	PF4/AN4	PF4/AN4
87	PF5/AN5	PF5/AN5	PF5/AN5	PF5/AN5
89	PF6/AN6	PF6/AN6	PF6/AN6	PF6/AN6
90	PF7/AN7	PF7/AN7	PF7/AN7	PF7/AN7

【注】 \*1 当使用 E10A (ASEMD0=Low 电平) 时, 这些引脚被固定为 TMS、TRST、TDI、TDO、TCK、ASEBRKAK/ASEBRK。

\*2 只限支持 E10A 全功能的 F-ZTAT 版本。当使用 E10A 的 AUD 功能时, 这些引脚被固定为 AUD 功能。

表 21.18 按运行模式分类的引脚功能一览表 (SH7084 (1))

引脚号	引脚名			
	内部 ROM 无效 (MCU 模式 0)		内部 ROM 无效 (MCU 模式 1)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
21,37,65,80,103	Vcc	Vcc	Vcc	Vcc
3,27,39,55,61,71,90,101	Vss	Vss	Vss	Vss
23,81,109	Vcl	Vcl	Vcl	Vcl
100	AVcc	AVcc	AVcc	AVcc
97	AVss	AVss	AVss	AVss
82	PLLvss	PLLvss	PLLvss	PLLvss
74	EXTAL	EXTAL	EXTAL	EXTAL
72	XTAL	XTAL	XTAL	XTAL
75	MD0	MD0	MD0	MD0
73	MD1	MD1	MD1	MD1
77	FWE	FWE	FWE	FWE
84	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$
35	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$
76	NMI	NMI	NMI	NMI
33	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$
51	PA0	PA0/ $\overline{\text{CS4}}$ /RXD0	PA0	PA0/ $\overline{\text{CS4}}$ /RXD0
50	PA1	PA1/ $\overline{\text{CS5}}$ /TXD0	PA1	PA1/ $\overline{\text{CS5}}$ /TXD0
49	PA2	PA2/A25/DREQ0/IRQ0/SCK0	PA2	PA2/A25/DREQ0/IRQ0/SCK0
48	PA3	PA3/A24/RXD1	PA3	PA3/A24/RXD1
47	PA4	PA4/A23/TXD1	PA4	PA4/A23/TXD1



引脚号	引脚名			
	内部 ROM 无效 (MCU 模式 0)		内部 ROM 无效 (MCU 模式 1)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
46	PA5	PA5/A22/DREQ1/IRQ1/SCK1	PA5	PA5/A22/DREQ1/IRQ1/SCK1
45	PA6	PA6/CS2/TCLKA	PA6	PA6/CS2/TCLKA
44	PA7	PA7/CS3/TCLKB	PA7	PA7/CS3/TCLKB
43	PA8	PA8/RDWR/IRQ2/TCLKC	PA8	PA8/RDWR/IRQ2/TCLKC
42	PA9	PA9/CKE/IRQ3/TCLKD	PA9	PA9/CKE/IRQ3/TCLKD
41	CS0	PA10/CS0/POE4	CS0	PA10/CS0/POE4
40	CS1	PA11/CS1/POE5	CS1	PA11/CS1/POE5
38	WRL	PA12/WRL/DQMLL/POE6	WRL	PA12/WRL/DQMLL/POE6
36	WRH	PA13/WRH/DQMLU/POE7	WRH	PA13/WRH/DQMLU/POE7
34	RD	PA14/RD	RD	PA14/RD
83	CK	PA15/CK	CK	PA15/CK
78	PA16	PA16/AH/CKE	PA16	PA16/AH/CKE
79	PA17	PA17/WAIT	PA17	PA17/WAIT
20	A16	PB0/A16/TIC5WS	A16	PB0/A16/TIC5WS
22	A17	PB1/A17/TIC5W	A17	PB1/A17/TIC5W
24	PB2	PB2/IRQ0/POE0/SCL	PB2	PB2/IRQ0/POE0/SCL
25	PB3	PB3/IRQ1/POE1/SDA	PB3	PB3/IRQ1/POE1/SDA
26	PB4	PB4/RASL/IRQ2/POE2	PB4	PB4/RASL/IRQ2/POE2
28	PB5	PB5/CASL/IRQ3/POE3	PB5	PB5/CASL/IRQ3/POE3
29	PB6	PB6/A18/BACK/IRQ4/RXD0	PB6	PB6/A18/BACK/IRQ4/RXD0
30	PB7	PB7/A19/BREQ/IRQ5/TXD0	PB7	PB7/A19/BREQ/IRQ5/TXD0
31	PB8	PB8/A20/WAIT/IRQ6/SCK0	PB8	PB8/A20/WAIT/IRQ6/SCK0
32	PB9	PB9/A21/IRQ7/ADTRG/POE8	PB9	PB9/A21/IRQ7/ADTRG/POE8
4	A0	PC0/A0	A0	PC0/A0
5	A1	PC1/A1	A1	PC1/A1
6	A2	PC2/A2	A2	PC2/A2
7	A3	PC3/A3	A3	PC3/A3
8	A4	PC4/A4	A4	PC4/A4
9	A5	PC5/A5	A5	PC5/A5
10	A6	PC6/A6	A6	PC6/A6
11	A7	PC7/A7	A7	PC7/A7
12	A8	PC8/A8	A8	PC8/A8
13	A9	PC9/A9	A9	PC9/A9
14	A10	PC10/A10	A10	PC10/A10
15	A11	PC11/A11	A11	PC11/A11

引脚号	引脚名			
	内部 ROM 无效 (MCU 模式 0)		内部 ROM 无效 (MCU 模式 1)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
16	A12	PC12/A12	A12	PC12/A12
17	A13	PC13/A13	A13	PC13/A13
18	A14	PC14/A14	A14	PC14/A14
19	A15	PC15/A15	A15	PC15/A15
70	D0	PD0/D0	D0	PD0/D0
69	D1	PD1/D1	D1	PD1/D1
68	D2	PD2/D2/TIC5U	D2	PD2/D2/TIC5U
67	D3	PD3/D3/TIC5V	D3	PD3/D3/TIC5V
66	D4	PD4/D4/TIC5W	D4	PD4/D4/TIC5W
64	D5	PD5/D5/TIC5US	D5	PD5/D5/TIC5US
63	D6	PD6/D6/TIC5VS	D6	PD6/D6/TIC5VS
62	D7	PD7/D7/TIC5WS	D7	PD7/D7/TIC5WS
60	PD8/(AUDATA0*2)	PD8/D8/TIOC3AS	D8/(AUDATA0*2)	PD8/D8/TIOC3AS
59	PD9/(AUDATA1*2)	PD9/D9/TIOC3BS	D9/(AUDATA1*2)	PD9/D9/TIOC3BS
58	PD10/(AUDATA2*2)	PD10/D10/TIOC3CS	D10/(AUDATA2*2)	PD10/D10/TIOC3CS
57	PD11/(AUDATA3*2)	PD11/D11/TIOC3DS	D11/(AUDATA3*2)	PD11/D11/TIOC3DS
56	PD12	PD12/D12/TIOC4AS	D12	PD12/D12/TIOC4AS
54	PD13	PD13/D13/TIOC4BS	D13	PD13/D13/TIOC4BS
53	PD14/(AUDCK*2)	PD14/D14/TIOC4CS	D14/(AUDCK*2)	PD14/D14/TIOC4CS
52	PD15/(AUDSYN*2)	PD15/D15/TIOC4DS	D15/(AUDSYN*2)	PD15/D15/TIOC4DS
85	PE0/(TMS*1)	PE0/DREQ0/TIOC0A	PE0/(TMS*1)	PE0/DREQ0/TIOC0A
86	PE1/(TRST*1)	PE1/TEND0/TIOC0B	PE1/(TRST*1)	PE1/TEND0/TIOC0B
87	PE2/(TDI*1)	PE2/DREQ1/TIOC0C	PE2/(TDI*1)	PE2/DREQ1/TIOC0C
88	PE3/(TDO*1)	PE3/TEND1/TIOC0D	PE3/(TDO*1)	PE3/TEND1/TIOC0D
89	PE4/(TCK*1)	PE4/TIOC1A/RXD3	PE4/(TCK*1)	PE4/TIOC1A/RXD3
102	PE5/(ASEBRKAK /ASEBRK*1)	PE5/CS6/TIOC1B/TXD3	PE5/(ASEBRKAK /ASEBRK*1)	PE5/CS6/TIOC1B/TXD3
104	PE6	PE6/CS7/TIOC2A/SCK3	PE6	PE6/CS7/TIOC2A/SCK3
105	PE7	PE7/BS/TIOC2B/UBCTRG/ RXD2/SSI	PE7	PE7/BS/TIOC2B/UBCTRG/ RXD2/SSI
106	PE8	PE8/TIOC3A/SCK2/SSCK	PE8	PE8/TIOC3A/SCK2/SSCK
107	PE9	PE9/TIOC3B/SCK3/RTS3	PE9	PE9/TIOC3B/SCK3/RTS3
108	PE10	PE10/TIOC3C/TXD2/SSO	PE10	PE10/TIOC3C/TXD2/SSO
110	PE11	PE11/TIOC3D/RXD3/CTS3	PE11	PE11/TIOC3D/RXD3/CTS3
111	PE12	PE12/TIOC4A/TXD3/SCS	PE12	PE12/TIOC4A/TXD3/SCS

引脚号	引脚名			
	内部 ROM 无效 (MCU 模式 0)		内部 ROM 无效 (MCU 模式 1)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
112	PE13	PE13/TIOC4B/MRES	PE13	PE13/TIOC4B/MRES
1	PE14	PE14/AH/DACK0/TIOC4C	PE14	PE14/AH/DACK0/TIOC4C
2	PE15	PE15/CKE/DACK1/TIOC4D /IRQOUT	PE15	PE15/CKE/DACK1/TIOC4D /IRQOUT
91	PF0/AN0	PF0/AN0	PF0/AN0	PF0/AN0
92	PF1/AN1	PF1/AN1	PF1/AN1	PF1/AN1
93	PF2/AN2	PF2/AN2	PF2/AN2	PF2/AN2
94	PF3/AN3	PF3/AN3	PF3/AN3	PF3/AN3
95	PF4/AN4	PF4/AN4	PF4/AN4	PF4/AN4
96	PF5/AN5	PF5/AN5	PF5/AN5	PF5/AN5
98	PF6/AN6	PF6/AN6	PF6/AN6	PF6/AN6
99	PF7/AN7	PF7/AN7	PF7/AN7	PF7/AN7

【注】 \*1 当使用 E10A (ASEMD0=Low 电平) 时, 这些引脚被固定为 TMS、TRST、TDI、TDO、TCK、ASEBRKAK/ASEBRK。

\*2 只限支持 E10A 全功能的 F-ZTAT 版本。当使用 E10A 的 AUD 功能时, 这些引脚被固定为 AUD 功能。

表 21.18 按运行模式分类的引脚功能一览表 (SH7084 (2))

引脚号	引脚名			
	内部 ROM 有效 (MCU 模式 2)		单芯片模式 (MCU 模式 3)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
21,37,65, 80,103	Vcc	Vcc	Vcc	Vcc
3,27,39,55, 61,71,90,101	Vss	Vss	Vss	Vss
23,81,109	Vcl	Vcl	Vcl	Vcl
100	AVcc	AVcc	AVcc	AVcc
97	AVss	AVss	AVss	AVss
82	PLLvss	PLLvss	PLLvss	PLLvss
74	EXTAL	EXTAL	EXTAL	EXTAL
72	XTAL	XTAL	XTAL	XTAL
75	MD0	MD0	MD0	MD0
73	MD1	MD1	MD1	MD1
77	FWE	FWE	FWE	FWE
84	RES	RES	RES	RES
35	WDTOVF	WDTOVF	WDTOVF	WDTOVF
76	NMI	NMI	NMI	NMI

引脚号	引脚名			
	内部 ROM 有效 (MCU 模式 2)		单芯片模式 (MCU 模式 3)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
33	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$
51	PA0	PA0/ $\overline{\text{CS4/RXD0}}$	PA0	PA0/RXD0
50	PA1	PA1/ $\overline{\text{CS5/TXD0}}$	PA1	PA1/TXD0
49	PA2	PA2/A25/DREQ0/IRQ0/SCK0	PA2	PA2/DREQ0/IRQ0/SCK0
48	PA3	PA3/A24/RXD1	PA3	PA3/RXD1
47	PA4	PA4/A23/TXD1	PA4	PA4/TXD1
46	PA5	PA5/A22/DREQ1/IRQ1/SCK1	PA5	PA5/IRQ1/SCK1
45	PA6	PA6/ $\overline{\text{CS2/TCLKA}}$	PA6	PA6/TCLKA
44	PA7	PA7/ $\overline{\text{CS3/TCLKB}}$	PA7	PA7/TCLKB
43	PA8	PA8/RDWR/IRQ2/TCLKC	PA8	PA8/IRQ2/TCLKC
42	PA9	PA9/CKE/IRQ3/TCLKD	PA9	PA9/IRQ3/TCLKD
41	PA10	PA10/ $\overline{\text{CS0/POE4}}$	PA10	PA10/ $\overline{\text{POE4}}$
40	PA11	PA11/ $\overline{\text{CS1/POE5}}$	PA11	PA11/ $\overline{\text{POE5}}$
38	PA12	PA12/ $\overline{\text{WRL/DQMLL/POE6}}$	PA12	PA12/ $\overline{\text{POE6}}$
36	PA13	PA13/ $\overline{\text{WRH/DQMLU/POE7}}$	PA13	PA13/ $\overline{\text{POE7}}$
34	PA14	PA14/ $\overline{\text{RD}}$	PA14	PA14
83	CK	PA15/CK	PA15	PA15
78	PA16	PA16/ $\overline{\text{AH/CKE}}$	PA16	PA16
79	PA17	PA17/ $\overline{\text{WAIT}}$	PA17	PA17
20	PB0	PB0/A16/TIC5WS	PB0	PB0/TIC5WS
22	PB1	PB1/A17/TIC5W	PB1	PB1/TIC5W
24	PB2	PB2/IRQ0/ $\overline{\text{POE0/SCL}}$	PB2	PB2/IRQ0/ $\overline{\text{POE0/SCL}}$
25	PB3	PB3/IRQ1/ $\overline{\text{POE1/SDA}}$	PB3	PB3/IRQ1/ $\overline{\text{POE1/SDA}}$
26	PB4	PB4/ $\overline{\text{RASL/IRQ2/POE2}}$	PB4	PB4/IRQ2/ $\overline{\text{POE2}}$
28	PB5	PB5/ $\overline{\text{CASL/IRQ3/POE3}}$	PB5	PB5/IRQ3/ $\overline{\text{POE3}}$
29	PB6	PB6/A18/ $\overline{\text{BACK/IRQ4/RXD0}}$	PB6	PB6/IRQ4/RXD0
30	PB7	PB7/A19/ $\overline{\text{BREQ/IRQ5/TXD0}}$	PB7	PB7/IRQ5/TXD0
31	PB8	PB8/A20/ $\overline{\text{WAIT/IRQ6/SCK0}}$	PB8	PB8/IRQ6/SCK0
32	PB9	PB9/A21/IRQ7/ $\overline{\text{ADTRG/POE8}}$	PB9	PB9/IRQ7/ $\overline{\text{ADTRG/POE8}}$
4	PC0	PC0/A0	PC0	PC0
5	PC1	PC1/A1	PC1	PC1
6	PC2	PC2/A2	PC2	PC2
7	PC3	PC3/A3	PC3	PC3
8	PC4	PC4/A4	PC4	PC4
9	PC5	PC5/A5	PC5	PC5

引脚号	引脚名			
	内部 ROM 有效 (MCU 模式 2)		单芯片模式 (MCU 模式 3)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
10	PC6	PC6/A6	PC6	PC6
11	PC7	PC7/A7	PC7	PC7
12	PC8	PC8/A8	PC8	PC8
13	PC9	PC9/A9	PC9	PC9
14	PC10	PC10/A10	PC10	PC10
15	PC11	PC11/A11	PC11	PC11
16	PC12	PC12/A12	PC12	PC12
17	PC13	PC13/A13	PC13	PC13
18	PC14	PC14/A14	PC14	PC14
19	PC15	PC15/A15	PC15	PC15
70	PD0	PD0/D0	PD0	PD0
69	PD1	PD1/D1	PD1	PD1
68	PD2	PD2/D2/TIC5U	PD2	PD2/TIC5U
67	PD3	PD3/D3/TIC5V	PD3	PD3/TIC5V
66	PD4	PD4/D4/TIC5W	PD4	PD4/TIC5W
64	PD5	PD5/D5/TIC5US	PD5	PD5/TIC5US
63	PD6	PD6/D6/TIC5VS	PD6	PD6/TIC5VS
62	PD7	PD7/D7/TIC5WS	PD7	PD7/TIC5WS
60	PD8/(AUDATA0*2)	PD8/D8/TIOC3AS	PD8/(AUDATA0*2)	PD8/TIOC3AS
59	PD9/(AUDATA1*2)	PD9/D9/TIOC3BS	PD9/(AUDATA1*2)	PD9/TIOC3BS
58	PD10/(AUDATA2*2)	PD10/D10/TIOC3CS	PD10/(AUDATA2*2)	PD10/TIOC3CS
57	PD11/(AUDATA3*2)	PD11/D11/TIOC3DS	PD11/(AUDATA3*2)	PD11/TIOC3DS
56	PD12	PD12/D12/TIOC4AS	PD12	PD12/TIOC4AS
54	PD13	PD13/D13/TIOC4BS	PD13	PD13/TIOC4BS
53	PD14/(AUDCK*2)	PD14/D14/TIOC4CS	PD14/(AUDCK*2)	PD14/TIOC4CS
52	PD15/(AUDSYNC*2)	PD15/D15/TIOC4DS	PD15/(AUDSYNC*2)	PD15/TIOC4DS
85	PE0/(TMS*1)	PE0/DREQ0/TIOC0A	PE0/(TMS*1)	PE0/DREQ0/TIOC0A
86	PE1/(TRST*1)	PE1/TEND0/TIOC0B	PE1/(TRST*1)	PE1/TIOC0B
87	PE2/(TDI*1)	PE2/DREQ1/TIOC0C	PE2/(TDI*1)	PE2/DREQ1/TIOC0C
88	PE3/(TDO*1)	PE3/TEND1/TIOC0D	PE3/(TDO*1)	PE3/TIOC0D
89	PE4/(TCK*1)	PE4/TIOC1A/RXD3	PE4/(TCK*1)	PE4/TIOC1A/RXD3
102	PE5/(ASEBRKAK /ASEBRK*1)	PE5/CS6/TIOC1B/TXD3	PE5/(ASEBRKAK /ASEBRK*1)	PE5/TIOC1B/TXD3
104	PE6	PE6/CS7/TIOC2A/SCK3	PE6	PE6/TIOC2A/SCK3

引脚号	引脚名			
	内部 ROM 有效 (MCU 模式 2)		单芯片模式 (MCU 模式 3)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
105	PE7	PE7/ $\overline{\text{BS}}$ /TIOC2B/ $\overline{\text{UBCTRG}}$ / $\overline{\text{RXD2}}$ /SSI	PE7	PE7/TIOC2B/ $\overline{\text{UBCTRG}}$ / $\overline{\text{RXD2}}$ /SSI
106	PE8	PE8/TIOC3A/SCK2/SSCK	PE8	PE8/TIOC3A/SCK2/SSCK
107	PE9	PE9/TIOC3B/SCK3/ $\overline{\text{RTS3}}$	PE9	PE9/TIOC3B/SCK3/ $\overline{\text{RTS3}}$
108	PE10	PE10/TIOC3C/ $\overline{\text{TXD2}}$ /SSO	PE10	PE10/TIOC3C/ $\overline{\text{TXD2}}$ /SSO
110	PE11	PE11/TIOC3D/ $\overline{\text{RXD3}}$ / $\overline{\text{CTS3}}$	PE11	PE11/TIOC3D/ $\overline{\text{RXD3}}$ / $\overline{\text{CTS3}}$
111	PE12	PE12/TIOC4A/ $\overline{\text{TXD3}}$ / $\overline{\text{SCS}}$	PE12	PE12/TIOC4A/ $\overline{\text{TXD3}}$ / $\overline{\text{SCS}}$
112	PE13	PE13/TIOC4B/ $\overline{\text{MRES}}$	PE13	PE13/TIOC4B/ $\overline{\text{MRES}}$
1	PE14	PE14/ $\overline{\text{AH}}$ /DACK0/TIOC4C	PE14	PE14/TIOC4C
2	PE15	PE15/ $\overline{\text{CKE}}$ /DACK1/TIOC4D/ $\overline{\text{IRQOUT}}$	PE15	PE15/TIOC4D/ $\overline{\text{IRQOUT}}$
91	PF0/AN0	PF0/AN0	PF0/AN0	PF0/AN0
92	PF1/AN1	PF1/AN1	PF1/AN1	PF1/AN1
93	PF2/AN2	PF2/AN2	PF2/AN2	PF2/AN2
94	PF3/AN3	PF3/AN3	PF3/AN3	PF3/AN3
95	PF4/AN4	PF4/AN4	PF4/AN4	PF4/AN4
96	PF5/AN5	PF5/AN5	PF5/AN5	PF5/AN5
98	PF6/AN6	PF6/AN6	PF6/AN6	PF6/AN6
99	PF7/AN7	PF7/AN7	PF7/AN7	PF7/AN7

【注】 \*1 当使用 E10A ( $\overline{\text{ASEMD0}}$ =Low 电平) 时, 这些引脚被固定为 TMS、 $\overline{\text{TRST}}$ 、TDI、TDO、TCK、 $\overline{\text{ASEBRKAK}}$ / $\overline{\text{ASEBRK}}$ 。

\*2 只限支持 E10A 全功能的 F-ZTAT 版本。当使用 E10A 的 AUD 功能时, 这些引脚被固定为 AUD 功能。

表 21.19 按运行模式分类的引脚功能一览表 (SH7085 (1))

引脚号	引脚名			
	内部 ROM 无效 (MCU 模式 0)		内部 ROM 无效 (MCU 模式 1)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
12,26,40,63, 77,85,104, 112,135	Vcc	Vcc	Vcc	Vcc
6,14,28,35, 55,71,79,87, 93,117,129	Vss	Vss	Vss	Vss
61,105,141	Vcl	Vcl	Vcl	Vcl
128	AVcc	AVcc	AVcc	AVcc
124	AVss	AVss	AVss	AVss
127	AVref	AVref	AVref	AVref
106	PLLvss	PLLvss	PLLvss	PLLvss
96	EXTAL	EXTAL	EXTAL	EXTAL
94	XTAL	XTAL	XTAL	XTAL
97	MD0	MD0	MD0	MD0
95	MD1	MD1	MD1	MD1
99	FWE	FWE	FWE	FWE
108	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$
44	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$
98	NMI	NMI	NMI	NMI
42	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$
130	PA0	PA0/ $\overline{\text{CS4}}$ /RXD0	PA0	PA0/ $\overline{\text{CS4}}$ /RXD0
131	PA1	PA1/ $\overline{\text{CS5}}$ / $\overline{\text{CE1A}}$ /TXD0	PA1	PA1/ $\overline{\text{CS5}}$ / $\overline{\text{CE1A}}$ /TXD0
132	PA2	PA2/A25/DREQ0/IRQ0/SCK0	PA2	PA2/A25/DREQ0/IRQ0/SCK0
133	PA3	PA3/A24/RXD1	PA3	PA3/A24/RXD1
134	PA4	PA4/A23/TXD1	PA4	PA4/A23/TXD1
136	PA5	PA5/A22/DREQ1/IRQ1/SCK1	PA5	PA5/A22/DREQ1/IRQ1/SCK1
54	PA6	PA6/ $\overline{\text{CS2}}$ /TCLKA	PA6	PA6/ $\overline{\text{CS2}}$ /TCLKA
53	PA7	PA7/ $\overline{\text{CS3}}$ /TCLKB	PA7	PA7/ $\overline{\text{CS3}}$ /TCLKB
52	PA8	PA8/RDWR/IRQ2/TCLKC	PA8	PA8/RDWR/IRQ2/TCLKC
51	PA9	PA9/ $\overline{\text{FRAME}}$ /CKE/IRQ3/TCLKD	PA9	PA9/ $\overline{\text{FRAME}}$ /CKE/IRQ3/TCLKD
50	$\overline{\text{CS0}}$	PA10/ $\overline{\text{CS0}}$ /POE4	$\overline{\text{CS0}}$	PA10/ $\overline{\text{CS0}}$ /POE4
49	$\overline{\text{CS1}}$	PA11/ $\overline{\text{CS1}}$ /POE5	$\overline{\text{CS1}}$	PA11/ $\overline{\text{CS1}}$ /POE5
48	$\overline{\text{WRL}}$	PA12/ $\overline{\text{WRL}}$ /DQMLL/POE6	$\overline{\text{WRL}}$	PA12/ $\overline{\text{WRL}}$ /DQMLL/POE6
47	$\overline{\text{WRH}}$	PA13/ $\overline{\text{WRH}}$ / $\overline{\text{WE}}$ /DQMLU/POE7	$\overline{\text{WRH}}$	PA13/ $\overline{\text{WRH}}$ / $\overline{\text{WE}}$ /DQMLU/POE7
43	$\overline{\text{RD}}$	PA14/ $\overline{\text{RD}}$	$\overline{\text{RD}}$	PA14/ $\overline{\text{RD}}$

引脚号	引脚名			
	内部 ROM 无效 (MCU 模式 0)		内部 ROM 无效 (MCU 模式 1)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
107	CK	PA15/CK	CK	PA15/CK
100	PA16/(AUDSYNC*2)	PA16/WRHH/ICIOR/AH/ DQMUU/CKE/DREQ2	PA16/(AUDSYNC*2)	PA16/WRHH/ICIOR/AH/ DQMUU/CKE/DREQ2
101	PA17	PA17/WAIT/DACK2	PA17	PA17/WAIT/DACK2
33	PA18	PA18/BREQ/TEND0	PA18	PA18/BREQ/TEND0
30	PA19	PA19/BACK/TEND1	PA19	PA19/BACK/TEND1
29	PA20	PA20/CS4/RASU	PA20	PA20/CS4/RASU
4	PA21	PA21/CS5/CE1A/CASU/TIC5U	PA21	PA21/CS5/CE1A/CASU/TIC5U
3	PA22	PA22/WRHL/ICIOR/DQMUL/ TIC5V	WRHL	PA22/WRHL/ICIOR/DQMUL/ TIC5V
1	PA23	PA23/WRHH/ICIOR/AH/ DQMUU/TIC5W	WRHH	PA23/WRHH/ICIOR/AH/ DQMUU/TIC5W
102	PA24	PA24/CE2A/DREQ3	PA24	PA24/CE2A/DREQ3
103	PA25	PA25/CE2B/DACK3/POE8	PA25	PA25/CE2B/DACK3/POE8
25	A16	PB0/A16/TIC5WS	A16	PB0/A16/TIC5WS
27	A17	PB1/A17/TIC5W	A17	PB1/A17/TIC5W
31	PB2	PB2/IRQ0/POE0/SCL	PB2	PB2/IRQ0/POE0/SCL
32	PB3	PB3/IRQ1/POE1/SDA	PB3	PB3/IRQ1/POE1/SDA
34	PB4	PB4/RASL/IRQ2/POE2	PB4	PB4/RASL/IRQ2/POE2
36	PB5	PB5/CASL/IRQ3/POE3	PB5	PB5/CASL/IRQ3/POE3
37	PB6	PB6/A18/BACK/IRQ4/RXD0	PB6	PB6/A18/BACK/IRQ4/RXD0
38	PB7	PB7/A19/BREQ/IRQ5/TXD0	PB7	PB7/A19/BREQ/IRQ5/TXD0
39	PB8	PB8/A20/WAIT/IRQ6/SCK0	PB8	PB8/A20/WAIT/IRQ6/SCK0
41	PB9	PB9/A21/IRQ7/ADTRG/POE8	PB9	PB9/A21/IRQ7/ADTRG/POE8
7	A0	PC0/A0	A0	PC0/A0
8	A1	PC1/A1	A1	PC1/A1
9	A2	PC2/A2	A2	PC2/A2
10	A3	PC3/A3	A3	PC3/A3
11	A4	PC4/A4	A4	PC4/A4
13	A5	PC5/A5	A5	PC5/A5
15	A6	PC6/A6	A6	PC6/A6
16	A7	PC7/A7	A7	PC7/A7
17	A8	PC8/A8	A8	PC8/A8
18	A9	PC9/A9	A9	PC9/A9
19	A10	PC10/A10	A10	PC10/A10



引脚号	引脚名			
	内部 ROM 无效 (MCU 模式 0)		内部 ROM 无效 (MCU 模式 1)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
20	A11	PC11/A11	A11	PC11/A11
21	A12	PC12/A12	A12	PC12/A12
22	A13	PC13/A13	A13	PC13/A13
23	A14	PC14/A14	A14	PC14/A14
24	A15	PC15/A15	A15	PC15/A15
92	D0	PD0/D0	D0	PD0/D0
91	D1	PD1/D1	D1	PD1/D1
90	D2	PD2/D2/TIC5U	D2	PD2/D2/TIC5U
89	D3	PD3/D3/TIC5V	D3	PD3/D3/TIC5V
88	D4	PD4/D4/TIC5W	D4	PD4/D4/TIC5W
86	D5	PD5/D5/TIC5US	D5	PD5/D5/TIC5US
84	D6	PD6/D6/TIC5VS	D6	PD6/D6/TIC5VS
83	D7	PD7/D7/TIC5WS	D7	PD7/D7/TIC5WS
82	D8	PD8/D8/TIOC3AS	D8	PD8/D8/TIOC3AS
81	D9	PD9/D9/TIOC3BS	D9	PD9/D9/TIOC3BS
80	D10	PD10/D10/TIOC3CS	D10	PD10/D10/TIOC3CS
78	D11	PD11/D11/TIOC3DS	D11	PD11/D11/TIOC3DS
76	D12	PD12/D12/TIOC4AS	D12	PD12/D12/TIOC4AS
75	D13	PD13/D13/TIOC4BS	D13	PD13/D13/TIOC4BS
74	D14	PD14/D14/TIOC4CS	D14	PD14/D14/TIOC4CS
73	D15	PD15/D15/TIOC4DS	D15	PD15/D15/TIOC4DS
72	PD16/(AUDATA0*2)	PD16/D16/IRQ0/ $\overline{\text{POE4}}$	D16/(AUDATA0*2)	PD16/D16/IRQ0/ $\overline{\text{POE4}}$
70	PD17/(AUDATA1*2)	PD17/D17/IRQ1/ $\overline{\text{POE5}}$	D17/(AUDATA1*2)	PD17/D17/IRQ1/ $\overline{\text{POE5}}$
69	PD18/(AUDATA2*2)	PD18/D18/IRQ2/ $\overline{\text{POE6}}$	D18/(AUDATA2*2)	PD18/D18/IRQ2/ $\overline{\text{POE6}}$
68	PD19/(AUDATA3*2)	PD19/D19/IRQ3/ $\overline{\text{POE7}}$	D19/(AUDATA3*2)	PD19/D19/IRQ3/ $\overline{\text{POE7}}$
67	PD20	PD20/D20/IRQ4/TIC5WS	D20	PD20/D20/IRQ4/TIC5WS
66	PD21	PD21/D21/IRQ5/TIC5VS	D21	PD21/D21/IRQ5/TIC5VS
65	PD22/(AUDCK*2)	PD22/D22/IRQ6/TIC5US	D22/(AUDCK*2)	PD22/D22/IRQ6/TIC5US
64	PD23/(AUDSYNC*2)	PD23/D23/IRQ7	D23/(AUDSYNC*2)	PD23/D23/IRQ7
62	PD24	PD24/D24/DREQ0/TIOC4DS	D24	PD24/D24/DREQ0/TIOC4DS
60	PD25	PD25/D25/DREQ1/TIOC4CS	D25	PD25/D25/DREQ1/TIOC4CS
59	PD26	PD26/D26/DACK0/TIOC4BS	D26	PD26/D26/DACK0/TIOC4BS
58	PD27	PD27/D27/DACK1/TIOC4AS	D27	PD27/D27/DACK1/TIOC4AS
57	PD28	PD28/D28/ $\overline{\text{CS2}}$ /TIOC3DS	D28	PD28/D28/ $\overline{\text{CS2}}$ /TIOC3DS
56	PD29	PD29/D29/ $\overline{\text{CS3}}$ /TIOC3BS	D29	PD29/D29/ $\overline{\text{CS3}}$ /TIOC3BS

引脚号	引脚名			
	内部 ROM 无效 (MCU 模式 0)		内部 ROM 无效 (MCU 模式 1)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
46	PD30	PD30/D30/TIOC3CS/IRQOUT	D30	PD30/D30/TIOC3CS/IRQOUT
45	PD31	PD31/D31/TIOC3AS/ADTRG	D31	PD31/D31/TIOC3AS/ADTRG
109	PE0/(AUDCK*2)	PE0/DREQ0/TIOC0A	PE0/(AUDCK*2)	PE0/DREQ0/TIOC0A
110	PE1	PE1/TEND0/TIOC0B	PE1	PE1/TEND0/TIOC0B
111	PE2	PE2/DREQ1/TIOC0C	PE2	PE2/DREQ1/TIOC0C
113	PE3/(AUDATA3*2)	PE3/TEND1/TIOC0D	PE3/(AUDATA3*2)	PE3/TEND1/TIOC0D
114	PE4/(AUDATA2*2)	PE4/IOIS16/TIOC1A/RXD3	PE4/(AUDATA2*2)	PE4/IOIS16/TIOC1A/RXD3
115	PE5/(AUDATA1*2)	PE5/CS6/CE1B/TIOC1B/TXD3	PE5/(AUDATA1*2)	PE5/CS6/CE1B/TIOC1B/TXD3
116	PE6/(AUDATA0*2)	PE6/CS7/TIOC2A/SCK3	PE6/(AUDATA0*2)	PE6/CS7/TIOC2A/SCK3
137	PE7	PE7/BS/TIOC2B/UBCTRG/ RXD2/SSI	PE7	PE7/BS/TIOC2B/UBCTRG/ RXD2/SSI
138	PE8/(TMS*1)	PE8/TIOC3A/SCK2/SSCK	PE8/(TMS*1)	PE8/TIOC3A/SCK2/SSCK
139	PE9/(TRST*1)	PE9/TIOC3B/SCK3/RTS3	PE9/(TRST*1)	PE9/TIOC3B/SCK3/RTS3
140	PE10/(TDI*1)	PE10/TIOC3C/TXD2/SSO	PE10/(TDI*1)	PE10/TIOC3C/TXD2/SSO
142	PE11/(TDO*1)	PE11/TIOC3D/RXD3/CTS3	PE11/(TDO*1)	PE11/TIOC3D/RXD3/CTS3
143	PE12/(TCK*1)	PE12/TIOC4A/TXD3/SCS	PE12/(TCK*1)	PE12/TIOC4A/TXD3/SCS
144	PE13/(ASEBRKAK /ASEBRK*1)	PE13/TIOC4B/MRES	PE13/(ASEBRKAK /ASEBRK*1)	PE13/TIOC4B/MRES
2	PE14	PE14/WRHH/ICIOWR/AH/ DQMUU/DACK0/TIOC4C	PE14	PE14/WRHH/ICIOWR/AH/ DQMUU/DACK0/TIOC4C
5	PE15	PE15/CKE/DACK1/TIOC4D /IRQOUT	PE15	PE15/CKE/DACK1/TIOC4D /IRQOUT
118	PF0/AN0	PF0/AN0	PF0/AN0	PF0/AN0
119	PF1/AN1	PF1/AN1	PF1/AN1	PF1/AN1
120	PF2/AN2	PF2/AN2	PF2/AN2	PF2/AN2
121	PF3/AN3	PF3/AN3	PF3/AN3	PF3/AN3
122	PF4/AN4	PF4/AN4	PF4/AN4	PF4/AN4
123	PF5/AN5	PF5/AN5	PF5/AN5	PF5/AN5
125	PF6/AN6	PF6/AN6	PF6/AN6	PF6/AN6
126	PF7/AN7	PF7/AN7	PF7/AN7	PF7/AN7

【注】 \*1 当使用 E10A (ASEMD0=Low 电平) 时, 这些引脚被固定为 TMS、TRST、TDI、TDO、TCK、ASEBRKAK/ASEBRK。

\*2 只限支持 E10A 全功能的 F-ZTAT 版本。当使用 E10A 的 AUD 功能时, 这些引脚被固定为 AUD 功能。

表 21.19 按运行模式分类的引脚功能一览表 (SH7085 (2))

引脚号	引脚名			
	内部 ROM 有效 (MCU 模式 2)		单芯片模式 (MCU 模式 3)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
12,26,40,63, 77,85,104, 112,135	Vcc	Vcc	Vcc	Vcc
6,14,28,35, 55,71,79,87, 93,117,129	Vss	Vss	Vss	Vss
61,105,141	Vcl	Vcl	Vcl	Vcl
128	AVcc	AVcc	AVcc	AVcc
124	AVss	AVss	AVss	AVss
127	AVref	AVref	AVref	AVref
106	PLLvss	PLLvss	PLLvss	PLLvss
96	EXTAL	EXTAL	EXTAL	EXTAL
94	XTAL	XTAL	XTAL	XTAL
97	MD0	MD0	MD0	MD0
95	MD1	MD1	MD1	MD1
99	FWE	FWE	FWE	FWE
108	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$
44	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$
98	NMI	NMI	NMI	NMI
42	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$
130	PA0	PA0/ $\overline{\text{CS4}}$ /RXD0	PA0	PA0/RXD0
131	PA1	PA1/ $\overline{\text{CS5}}$ / $\overline{\text{CE1A}}$ /TXD0	PA1	PA1/TXD0
132	PA2	PA2/A25/DREQ0/IRQ0/SCK0	PA2	PA2/DREQ0/IRQ0/SCK0
133	PA3	PA3/A24/RXD1	PA3	PA3/RXD1
134	PA4	PA4/A23/TXD1	PA4	PA4/TXD1
136	PA5	PA5/A22/DREQ1/IRQ1/SCK1	PA5	PA5/DREQ1/IRQ1/SCK1
54	PA6	PA6/ $\overline{\text{CS2}}$ /TCLKA	PA6	PA6/TCLKA
53	PA7	PA7/ $\overline{\text{CS3}}$ /TCLKB	PA7	PA7/TCLKB
52	PA8	PA8/RDWR/IRQ2/TCLKC	PA8	PA8/IRQ2/TCLKC
51	PA9	PA9/ $\overline{\text{FRAME}}$ /CKE/IRQ3/TCLKD	PA9	PA9/IRQ3/TCLKD
50	PA10	PA10/ $\overline{\text{CS0}}$ / $\overline{\text{POE4}}$	PA10	PA10/ $\overline{\text{POE4}}$
49	PA11	PA11/ $\overline{\text{CS1}}$ / $\overline{\text{POE5}}$	PA11	PA11/ $\overline{\text{POE5}}$
48	PA12	PA12/ $\overline{\text{WRL}}$ / $\overline{\text{DQMLL}}$ / $\overline{\text{POE6}}$	PA12	PA12/ $\overline{\text{POE6}}$
47	PA13	PA13/ $\overline{\text{WRH}}$ / $\overline{\text{WE}}$ / $\overline{\text{DQMLU}}$ / $\overline{\text{POE7}}$	PA13	PA13/ $\overline{\text{POE7}}$
43	PA14	PA14/ $\overline{\text{RD}}$	PA14	PA14

引脚号	引脚名			
	内部 ROM 有效 (MCU 模式 2)		单芯片模式 (MCU 模式 3)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
107	CK	PA15/CK	PA15	PA15
100	PA16/(AUDSYNC*2)	PA16/WRHH/ICIOWR/AH/ DQMUU/CKE/DREQ2	PA16/(AUDSYNC*2)	PA16/DREQ2
101	PA17	PA17/WAIT/DACK2	PA17	PA17
33	PA18	PA18/BREQ/TEND0	PA18	PA18
30	PA19	PA19/BACK/TEND1	PA19	PA191
29	PA20	PA20/CS4/RASU	PA20	PA20
4	PA21	PA21/CS5/CE1A/CASU/TIC5U	PA21	PA21/TIC5U
3	PA22	PA22/WRHL/ICIORD/DQMUL/ TIC5V	PA22	PA22/TIC5V
1	PA23	PA23/WRHH/ICIOWR/AH/ DQMUU/TIC5W	PA23	PA23/TIC5W
102	PA24	PA24/CE2A/DREQ3	PA24	PA24/DREQ3
103	PA25	PA25/CE2B/DACK3/POE8	PA25	PA25/POE8
25	PB0	PB0/A16/TIC5WS	PB0	PB0/TIC5WS
27	PB1	PB1/A17/TIC5W	PB1	PB1/TIC5W
31	PB2	PB2/IRQ0/POE0/SCL	PB2	PB2/IRQ0/POE0/SCL
32	PB3	PB3/IRQ1/POE1/SDA	PB3	PB3/IRQ1/POE1/SDA
34	PB4	PB4/RASL/IRQ2/POE2	PB4	PB4/IRQ2/POE2
36	PB5	PB5/CASL/IRQ3/POE3	PB5	PB5/IRQ3/POE3
37	PB6	PB6/A18/BACK/IRQ4/RXD0	PB6	PB6/IRQ4/RXD0
38	PB7	PB7/A19/BREQ/IRQ5/TXD0	PB7	PB7/IRQ5/TXD0
39	PB8	PB8/A20/WAIT/IRQ6/SCK0	PB8	PB8/IRQ6/SCK0
41	PB9	PB9/A21/IRQ7/ADTRG/POE8	PB9	PB9/IRQ7/ADTRG/POE8
7	PC0	PC0/A0	PC0	PC0
8	PC1	PC1/A1	PC1	PC1
9	PC2	PC2/A2	PC2	PC2
10	PC3	PC3/A3	PC3	PC3
11	PC4	PC4/A4	PC4	PC4
13	PC5	PC5/A5	PC5	PC5
15	PC6	PC6/A6	PC6	PC6
16	PC7	PC7/A7	PC7	PC7
17	PC8	PC8/A8	PC8	PC8
18	PC9	PC9/A9	PC9	PC9
19	PC10	PC10/A10	PC10	PC10

引脚号	引脚名			
	内部 ROM 有效 (MCU 模式 2)		单芯片模式 (MCU 模式 3)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
20	PC11	PC11/A11	PC11	PC11
21	PC12	PC12/A12	PC12	PC12
22	PC13	PC13/A13	PC13	PC13
23	PC14	PC14/A14	PC14	PC14
24	PC15	PC15/A15	PC15	PC15
92	PD0	PD0/D0	PD0	PD0
91	PD1	PD1/D1	PD1	PD1
90	PD2	PD2/D2/TIC5U	PD2	PD2/TIC5U
89	PD3	PD3/D3/TIC5V	PD3	PD3/TIC5V
88	PD4	PD4/D4/TIC5W	PD4	PD4/TIC5W
86	PD5	PD5/D5/TIC5US	PD5	PD5/TIC5US
84	PD6	PD6/D6/TIC5VS	PD6	PD6/TIC5VS
83	PD7	PD7/D7/TIC5WS	PD7	PD7/TIC5WS
82	PD8	PD8/D8/TIOC3AS	PD8	PD8/TIOC3AS
81	PD9	PD9/D9/TIOC3BS	PD9	PD9/TIOC3BS
80	PD10	PD10/D10/TIOC3CS	PD10	PD10/TIOC3CS
78	PD11	PD11/D11/TIOC3DS	PD11	PD11/TIOC3DS
76	PD12	PD12/D12/TIOC4AS	PD12	PD12/TIOC4AS
75	PD13	PD13/D13/TIOC4BS	PD13	PD13/TIOC4BS
74	PD14	PD14/D14/TIOC4CS	PD14	PD14/TIOC4CS
73	PD15	PD15/D15/TIOC4DS	PD15	PD15/TIOC4DS
72	PD16/(AUDATA0*2)	PD16/D16/IRQ0/ $\overline{\text{POE4}}$	PD16/(AUDATA0*2)	PD16/IRQ0/ $\overline{\text{POE4}}$
70	PD17/(AUDATA1*2)	PD17/D17/IRQ1/ $\overline{\text{POE5}}$	PD17/(AUDATA1*2)	PD17/IRQ1/ $\overline{\text{POE5}}$
69	PD18/(AUDATA2*2)	PD18/D18/IRQ2/ $\overline{\text{POE6}}$	PD18/(AUDATA2*2)	PD18/IRQ2/ $\overline{\text{POE6}}$
68	PD19/(AUDATA3*2)	PD19/D19/IRQ3/ $\overline{\text{POE7}}$	PD19/(AUDATA3*2)	PD19/IRQ3/ $\overline{\text{POE7}}$
67	PD20	PD20/D20/IRQ4/TIC5WS	PD20	PD20/IRQ4/TIC5WS
66	PD21	PD21/D21/IRQ5/TIC5VS	PD21	PD21/IRQ5/TIC5VS
65	PD22/(AUDCK*2)	PD22/D22/IRQ6/TIC5US	PD22/(AUDCK*2)	PD22/IRQ6/TIC5US
64	PD23/(AUDSYNC*2)	PD23/D23/IRQ7	PD23/(AUDSYNC*2)	PD23/IRQ7
62	PD24	PD24/D24/DREQ0/TIOC4DS	PD24	PD24/DREQ0/TIOC4DS
60	PD25	PD25/D25/DREQ1/TIOC4CS	PD25	PD25/DREQ1/TIOC4CS
59	PD26	PD26/D26/DACK0/TIOC4BS	PD26	PD26/TIOC4BS
58	PD27	PD27/D27/DACK1/TIOC4AS	PD27	PD27/TIOC4AS
57	PD28	PD28/D28/ $\overline{\text{CS2}}$ /TIOC3DS	PD28	PD28/TIOC3DS
56	PD29	PD29/D29/ $\overline{\text{CS3}}$ /TIOC3BS	PD29	PD29/TIOC3BS

引脚号	引脚名			
	内部 ROM 有效 (MCU 模式 2)		单芯片模式 (MCU 模式 3)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
46	PD30	PD30/D30/TIOC3CS/ $\overline{\text{IRQOUT}}$	PD30	PD30/TIOC3CS/ $\overline{\text{IRQOUT}}$
45	PD31	PD31/D31/TIOC3AS/ $\overline{\text{ADTRG}}$	PD31	PD31/TIOC3AS/ $\overline{\text{ADTRG}}$
109	PE0/(AUDCK*2)	PE0/DREQ0/TIOC0A	PE0/(AUDCK*2)	PE0/DREQ0/TIOC0A
110	PE1	PE1/TEND0/TIOC0B	PE1	PE1/TIOC0B
111	PE2	PE2/DREQ1/TIOC0C	PE2	PE2/DREQ1/TIOC0C
113	PE3/(AUDATA3*2)	PE3/TEND1/TIOC0D	PE3/(AUDATA3*2)	PE3/TIOC0D
114	PE4/(AUDATA2*2)	PE4/ $\overline{\text{IOIS16}}$ /TIOC1A/RXD3	PE4/(AUDATA2*2)	PE4/TIOC1A/RXD3
115	PE5/(AUDATA1*2)	PE5/ $\overline{\text{CS6}}$ / $\overline{\text{CE1B}}$ /TIOC1B/TXD3	PE5/(AUDATA1*2)	PE5/TIOC1B/TXD3
116	PE6/(AUDATA0*2)	PE6/ $\overline{\text{CS7}}$ /TIOC2A/SCK3	PE6/(AUDATA0*2)	PE6/TIOC2A/SCK3
137	PE7	PE7/ $\overline{\text{BS}}$ /TIOC2B/ $\overline{\text{UBCTRG}}$ / RXD2/SSI	PE7	PE7/TIOC2B/ $\overline{\text{UBCTRG}}$ / RXD2/SSI
138	PE8/(TMS*1)	PE8/TIOC3A/SCK2/SSCK	PE8(TMS*1)	PE8/TIOC3A/SCK2/SSCK
139	PE9/( $\overline{\text{TRST}}$ *1)	PE9/TIOC3B/SCK3/ $\overline{\text{RTS3}}$	PE9/( $\overline{\text{TRST}}$ *1)	PE9/TIOC3B/SCK3/ $\overline{\text{RTS3}}$
140	PE10/(TDI*1)	PE10/TIOC3C/TXD2/SSO	PE10/(TDI*1)	PE10/TIOC3C/TXD2/SSO
142	PE11/(TDO*1)	PE11/TIOC3D/RXD3/ $\overline{\text{CTS3}}$	PE11/(TDO*1)	PE11/TIOC3D/RXD3/ $\overline{\text{CTS3}}$
143	PE12/(TCK*1)	PE12/TIOC4A/TXD3/ $\overline{\text{SCS}}$	PE12/(TCK*1)	PE12/TIOC4A/TXD3/ $\overline{\text{SCS}}$
144	PE13/( $\overline{\text{ASEBRKAK}}$ / ASEBRK*1)	PE13/TIOC4B/ $\overline{\text{MRES}}$	PE13/( $\overline{\text{ASEBRKAK}}$ / ASEBRK*1)	PE13/TIOC4B/ $\overline{\text{MRES}}$
2	PE14	PE14/ $\overline{\text{WRHH}}$ / $\overline{\text{ICIOWR}}$ / $\overline{\text{AH}}$ / DQMUU/DACK0/TIOC4C	PE14	PE14/TIOC4C
5	PE15	PE15/CKE/DACK1/TIOC4D $\overline{\text{IRQOUT}}$	PE15	PE15/TIOC4D $\overline{\text{IRQOUT}}$
118	PF0/AN0	PF0/AN0	PF0/AN0	PF0/AN0
119	PF1/AN1	PF1/AN1	PF1/AN1	PF1/AN1
120	PF2/AN2	PF2/AN2	PF2/AN2	PF2/AN2
121	PF3/AN3	PF3/AN3	PF3/AN3	PF3/AN3
122	PF4/AN4	PF4/AN4	PF4/AN4	PF4/AN4
123	PF5/AN5	PF5/AN5	PF5/AN5	PF5/AN5
125	PF6/AN6	PF6/AN6	PF6/AN6	PF6/AN6
126	PF7/AN7	PF7/AN7	PF7/AN7	PF7/AN7

【注】 \*1 当使用 E10A (ASEMD0=Low 电平) 时, 这些引脚被固定 TMS、 $\overline{\text{TRST}}$ 、TDI、TDO、TCK、 $\overline{\text{ASEBRKAK}}$ / $\overline{\text{ASEBRK}}$ 。

\*2 只限支持 E10A 全功能的 F-ZTAT 版本。当使用 E10A 的 AUD 功能时, 这些引脚被固定为 AUD 功能。

表 21.20 按运行模式分类的引脚功能一览表 (SH7086 (1))

引脚号	引脚名			
	内部 ROM 无效 (MCU 模式 0)		内部 ROM 无效 (MCU 模式 1)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
11,21,35,48, 66,74,94, 102,109,128, 136,168	Vcc	Vcc	Vcc	Vcc
8,23,44,57, 64,86,96, 104,117, 141,163	Vss	Vss	Vss	Vss
37,71, 129,173	Vcl	Vcl	Vcl	Vcl
151,162	AVcc	AVcc	AVcc	AVcc
142,156	AVss	AVss	AVss	AVss
161	AVref	AVref	AVref	AVref
130	PLLvss	PLLvss	PLLvss	PLLvss
120	EXTAL	EXTAL	EXTAL	EXTAL
118	XTAL	XTAL	XTAL	XTAL
121	MD0	MD0	MD0	MD0
119	MD1	MD1	MD1	MD1
123	FWE	FWE	FWE	FWE
132	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$
53	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$
122	NMI	NMI	NMI	NMI
51	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$
164	PA0	PA0/ $\overline{\text{CS4}}$ /RXD0	PA0	PA0/ $\overline{\text{CS4}}$ /RXD0
165	PA1	PA1/ $\overline{\text{CS5}}$ / $\overline{\text{CE1A}}$ /TXD0	PA1	PA1/ $\overline{\text{CS5}}$ / $\overline{\text{CE1A}}$ /TXD0
166	PA2	PA2/A25/DREQ0/IRQ0/SCK0	PA2	PA2/A25/DREQ0/IRQ0/SCK0
167	PA3	PA3/A24/RXD1	PA3	PA3/A24/RXD1
169	PA4	PA4/A23/TXD1	PA4	PA4/A23/TXD1
170	PA5	PA5/A22/DREQ1/IRQ1/SCK1	PA5	PA5/A22/DREQ1/IRQ1/SCK1
80	PA6	PA6/ $\overline{\text{CS2}}$ /TCLKA	PA6	PA6/ $\overline{\text{CS2}}$ /TCLKA
79	PA7	PA7/ $\overline{\text{CS3}}$ /TCLKB	PA7	PA7/ $\overline{\text{CS3}}$ /TCLKB
78	PA8	PA8/RDWR/IRQ2/TCLKC	PA8	PA8/RDWR/IRQ2/TCLKC
77	PA9	PA9/ $\overline{\text{FRAME}}$ /CKE/IRQ3/TCLKD	PA9	PA9/ $\overline{\text{FRAME}}$ /CKE/IRQ3/TCLKD
76	$\overline{\text{CS0}}$	PA10/ $\overline{\text{CS0}}$ /POE4	$\overline{\text{CS0}}$	PA10/ $\overline{\text{CS0}}$ /POE4
75	$\overline{\text{CS1}}$	PA11/ $\overline{\text{CS1}}$ /POE5	$\overline{\text{CS1}}$	PA11/ $\overline{\text{CS1}}$ /POE5

引脚号	引脚名			
	内部 ROM 无效 (MCU 模式 0)		内部 ROM 无效 (MCU 模式 1)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
73	$\overline{\text{WRL}}$	PA12/ $\overline{\text{WRL}}$ / $\overline{\text{DQMLL}}$ / $\overline{\text{POE6}}$	$\overline{\text{WRL}}$	PA12/ $\overline{\text{WRL}}$ / $\overline{\text{DQMLL}}$ / $\overline{\text{POE6}}$
72	$\overline{\text{WRH}}$	PA13/ $\overline{\text{WRH}}$ / $\overline{\text{DQMLU}}$ / $\overline{\text{WE}}$ / $\overline{\text{POE7}}$	$\overline{\text{WRH}}$	PA13/ $\overline{\text{WRH}}$ / $\overline{\text{DQMLU}}$ / $\overline{\text{WE}}$ / $\overline{\text{POE7}}$
52	$\overline{\text{RD}}$	PA14/ $\overline{\text{RD}}$	$\overline{\text{RD}}$	PA14/ $\overline{\text{RD}}$
131	CK	PA15/CK	CK	PA15/CK
124	PA16/( $\overline{\text{AUDSYNC}}^*2$ )	PA16/ $\overline{\text{WRHH}}$ / $\overline{\text{ICIOWR}}$ / $\overline{\text{AH}}$ / $\overline{\text{DQMUU}}$ / $\overline{\text{CKE}}$ / $\overline{\text{DREQ2}}$	PA16/( $\overline{\text{AUDSYNC}}^*2$ )	PA16/ $\overline{\text{WRHH}}$ / $\overline{\text{ICIOWR}}$ / $\overline{\text{AH}}$ / $\overline{\text{DQMUU}}$ / $\overline{\text{CKE}}$ / $\overline{\text{DREQ2}}$
125	PA17	PA17/ $\overline{\text{WAIT}}$ / $\overline{\text{DACK2}}$	PA17	PA17/ $\overline{\text{WAIT}}$ / $\overline{\text{DACK2}}$
42	PA18	PA18/ $\overline{\text{BREQ}}$ / $\overline{\text{TEND0}}$	PA18	PA18/ $\overline{\text{BREQ}}$ / $\overline{\text{TEND0}}$
39	PA19	PA19/ $\overline{\text{BACK}}$ / $\overline{\text{TEND1}}$	PA19	PA19/ $\overline{\text{BACK}}$ / $\overline{\text{TEND1}}$
38	PA20	PA20/ $\overline{\text{CS4}}$ / $\overline{\text{RASU}}$	PA20	PA20/ $\overline{\text{CS4}}$ / $\overline{\text{RASU}}$
6	PA21	PA21/ $\overline{\text{CS5}}$ / $\overline{\text{CE1A}}$ / $\overline{\text{CASU}}$ / $\overline{\text{TIC5U}}$	PA21	PA21/ $\overline{\text{CS5}}$ / $\overline{\text{CE1A}}$ / $\overline{\text{CASU}}$ / $\overline{\text{TIC5U}}$
5	PA22	PA22/ $\overline{\text{WRHL}}$ / $\overline{\text{ICIORD}}$ / $\overline{\text{DQMUL}}$ / $\overline{\text{TIC5V}}$	$\overline{\text{WRHL}}$	PA22/ $\overline{\text{WRHL}}$ / $\overline{\text{ICIORD}}$ / $\overline{\text{DQMUL}}$ / $\overline{\text{TIC5V}}$
3	PA23	PA23/ $\overline{\text{WRHH}}$ / $\overline{\text{ICIOWR}}$ / $\overline{\text{AH}}$ / $\overline{\text{DQMUU}}$ / $\overline{\text{TIC5W}}$	$\overline{\text{WRHH}}$	PA23/ $\overline{\text{WRHH}}$ / $\overline{\text{ICIOWR}}$ / $\overline{\text{AH}}$ / $\overline{\text{DQMUU}}$ / $\overline{\text{TIC5W}}$
126	PA24	PA24/ $\overline{\text{CE2A}}$ / $\overline{\text{DREQ3}}$	PA24	PA24/ $\overline{\text{CE2A}}$ / $\overline{\text{DREQ3}}$
127	PA25	PA25/ $\overline{\text{CE2B}}$ / $\overline{\text{DACK3}}$ / $\overline{\text{POE8}}$	PA25	PA25/ $\overline{\text{CE2B}}$ / $\overline{\text{DACK3}}$ / $\overline{\text{POE8}}$
63	PA26	PA26/A26/IRQ0	PA26	PA26/A26/IRQ0
65	PA27	PA27/A27/IRQ1	PA27	PA27/A27/IRQ1
67	PA28	PA28/A28/IRQ2	PA28	PA28/A28/IRQ2
68	PA29	PA29/A29/IRQ3	PA29	PA29/A29/IRQ3
34	A16	PB0/A16/TIC5WS	A16	PB0/A16/TIC5WS
36	A17	PB1/A17/TIC5W	A17	PB1/A17/TIC5W
40	PB2	PB2/IRQ0/ $\overline{\text{POE0}}$ / $\overline{\text{SCL}}$	PB2	PB2/IRQ0/ $\overline{\text{POE0}}$ / $\overline{\text{SCL}}$
41	PB3	PB3/IRQ1/ $\overline{\text{POE1}}$ / $\overline{\text{SDA}}$	PB3	PB3/IRQ1/ $\overline{\text{POE1}}$ / $\overline{\text{SDA}}$
43	PB4	PB4/ $\overline{\text{RASL}}$ / $\overline{\text{IRQ2}}$ / $\overline{\text{POE2}}$	PB4	PB4/ $\overline{\text{RASL}}$ / $\overline{\text{IRQ2}}$ / $\overline{\text{POE2}}$
45	PB5	PB5/ $\overline{\text{CASL}}$ / $\overline{\text{IRQ3}}$ / $\overline{\text{POE3}}$	PB5	PB5/ $\overline{\text{CASL}}$ / $\overline{\text{IRQ3}}$ / $\overline{\text{POE3}}$
46	PB6	PB6/A18/ $\overline{\text{BACK}}$ / $\overline{\text{IRQ4}}$ / $\overline{\text{RXD0}}$	PB6	PB6/A18/ $\overline{\text{BACK}}$ / $\overline{\text{IRQ4}}$ / $\overline{\text{RXD0}}$
47	PB7	PB7/A19/ $\overline{\text{BREQ}}$ / $\overline{\text{IRQ5}}$ / $\overline{\text{TXD0}}$	PB7	PB7/A19/ $\overline{\text{BREQ}}$ / $\overline{\text{IRQ5}}$ / $\overline{\text{TXD0}}$
49	PB8	PB8/A20/ $\overline{\text{WAIT}}$ / $\overline{\text{IRQ6}}$ / $\overline{\text{SCK0}}$	PB8	PB8/A20/ $\overline{\text{WAIT}}$ / $\overline{\text{IRQ6}}$ / $\overline{\text{SCK0}}$
50	PB9	PB9/A21/ $\overline{\text{IRQ7}}$ / $\overline{\text{ADTRG}}$ / $\overline{\text{POE8}}$	PB9	PB9/A21/ $\overline{\text{IRQ7}}$ / $\overline{\text{ADTRG}}$ / $\overline{\text{POE8}}$
16	A0	PC0/A0	A0	PC0/A0
17	A1	PC1/A1	A1	PC1/A1
18	A2	PC2/A2	A2	PC2/A2
19	A3	PC3/A3	A3	PC3/A3



引脚号	引脚名			
	内部 ROM 无效 (MCU 模式 0)		内部 ROM 无效 (MCU 模式 1)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
20	A4	PC4/A4	A4	PC4/A4
22	A5	PC5/A5	A5	PC5/A5
24	A6	PC6/A6	A6	PC6/A6
25	A7	PC7/A7	A7	PC7/A7
26	A8	PC8/A8	A8	PC8/A8
27	A9	PC9/A9	A9	PC9/A9
28	A10	PC10/A10	A10	PC10/A10
29	A11	PC11/A11	A11	PC11/A11
30	A12	PC12/A12	A12	PC12/A12
31	A13	PC13/A13	A13	PC13/A13
32	A14	PC14/A14	A14	PC14/A14
33	A15	PC15/A15	A15	PC15/A15
54	A18	PC18/A18	A18	PC18/A18
55	A19	PC19/A19	A19	PC19/A19
56	A20	PC20/A20	A20	PC20/A20
58	A21	PC21/A21	A21	PC21/A21
59	A22	PC22/A22	A22	PC22/A22
60	A23	PC23/A23	A23	PC23/A23
61	A24	PC24/A24	A24	PC24/A24
62	A25	PC25/A25	A25	PC25/A25
116	D0	PD0/D0	D0	PD0/D0
115	D1	PD1/D1	D1	PD1/D1
114	D2	PD2/D2/TIC5U	D2	PD2/D2/TIC5U
113	D3	PD3/D3/TIC5V	D3	PD3/D3/TIC5V
112	D4	PD4/D4/TIC5W	D4	PD4/D4/TIC5W
111	D5	PD5/D5/TIC5US	D5	PD5/D5/TIC5US
110	D6	PD6/D6/TIC5VS	D6	PD6/D6/TIC5VS
108	D7	PD7/D7/TIC5WS	D7	PD7/D7/TIC5WS
107	D8	PD8/D8/TIOC3AS	D8	PD8/D8/TIOC3AS
106	D9	PD9/D9/TIOC3BS	D9	PD9/D9/TIOC3BS
105	D10	PD10/D10/TIOC3CS	D10	PD10/D10/TIOC3CS
103	D11	PD11/D11/TIOC3DS	D11	PD11/D11/TIOC3DS
101	D12	PD12/D12/TIOC4AS	D12	PD12/D12/TIOC4AS
100	D13	PD13/D13/TIOC4BS	D13	PD13/D13/TIOC4BS
99	D14	PD14/D14/TIOC4CS	D14	PD14/D14/TIOC4CS

引脚号	引脚名			
	内部 ROM 无效 (MCU 模式 0)		内部 ROM 无效 (MCU 模式 1)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
98	D15	PD15/D15/TIOC4DS	D15	PD15/D15/TIOC4DS
97	PD16/(AUDATA0*2)	PD16/D16/IRQ0/POE4	PD16/(AUDATA0*2)	PD16/D16/IRQ0/POE4
95	PD17/(AUDATA1*2)	PD17/D17/IRQ1/POE5	PD17/(AUDATA1*2)	PD17/D17/IRQ1/POE5
93	PD18/(AUDATA2*2)	PD18/D18/IRQ2/POE6	PD18/(AUDATA2*2)	PD18/D18/IRQ2/POE6
92	PD19/(AUDATA3*2)	PD19/D19/IRQ3/POE7	D19/(AUDATA3*2)	PD19/D19/IRQ3/POE7
91	PD20	PD20/D20/IRQ4/TIC5WS	D20	PD20/D20/IRQ4/TIC5WS
90	PD21	PD21/D21/IRQ5/TIC5VS	D21)	PD21/D21/IRQ5/TIC5VS
89	PD22/(AUDCK*2)	PD22/D22/IRQ6/TIC5US	D22/(AUDCK*2)	PD22/D22/IRQ6/TIC5US
88	PD23/(AUDSYNC*2)	PD23/D23/IRQ7	D23/(AUDSYNC*2)	PD23/D23/IRQ7
87	PD24	PD24/D24/DREQ0/TIOC4DS	D24	PD24/D24/DREQ0/TIOC4DS
85	PD25	PD25/D25/DREQ1/TIOC4CS	D25	PD25/D25/DREQ1/TIOC4CS
84	PD26	PD26/D26/DACK0/TIOC4BS	D26	PD26/D26/DACK0/TIOC4BS
83	PD27	PD27/D27/DACK1/TIOC4AS	D27	PD27/D27/DACK1/TIOC4AS
82	PD28	PD28/D28/ $\overline{\text{CS2}}$ /TIOC3DS	D28	PD28/D28/ $\overline{\text{CS2}}$ /TIOC3DS
81	PD29	PD29/D29/ $\overline{\text{CS3}}$ /TIOC3BS	D29	PD29/D29/ $\overline{\text{CS3}}$ /TIOC3BS
70	PD30	PD30/D30/TIOC3CS/ $\overline{\text{IRQOUT}}$	D30	PD30/D30/TIOC3CS/ $\overline{\text{IRQOUT}}$
69	PD31	PD31/D31/TIOC3AS/ $\overline{\text{ADTRG}}$	D31	PD31/D31/TIOC3AS/ $\overline{\text{ADTRG}}$
133	PE0/(AUDCK*2)	PE0/DREQ0/TIOC0A	PE0/(AUDCK*2)	PE0/DREQ0/TIOC0A
134	PE1	PE1/TEND0/TIOC0B	PE1	PE1/TEND0/TIOC0B
135	PE2	PE2/DREQ1/TIOC0C	PE2	PE2/DREQ1/TIOC0C
137	PE3/(AUDATA3*2)	PE3/TEND1/TIOC0D	PE3/(AUDATA3*2)	PE3/TEND1/TIOC0D
138	PE4/(AUDATA2*2)	PE4/ $\overline{\text{IOIS16}}$ /TIOC1A/RXD3	PE4/(AUDATA2*2)	PE4/ $\overline{\text{IOIS16}}$ /TIOC1A/RXD3
139	PE5/(AUDATA1*2)	PE5/ $\overline{\text{CS6}}$ / $\overline{\text{CE1B}}$ /TIOC1B/TXD3	PE5/(AUDATA1*2)	PE5/ $\overline{\text{CS6}}$ / $\overline{\text{CE1B}}$ /TIOC1B/TXD3
140	PE6/(AUDATA0*2)	PE6/ $\overline{\text{CS7}}$ /TIOC2A/SCK3	PE6/(AUDATA0*2)	PE6/ $\overline{\text{CS7}}$ /TIOC2A/SCK3
171	PE7	PE7/ $\overline{\text{BS}}$ /TIOC2B/ $\overline{\text{UBCTRG}}$ / RXD2/SSI	PE7	PE7/ $\overline{\text{BS}}$ /TIOC2B/ $\overline{\text{UBCTRG}}$ / RXD2/SSI
172	PE8/(TMS*1)	PE8/TIOC3A/SCK2/SSCK	PE8(TMS*1)	PE8/TIOC3A/SCK2/SSCK
174	PE9/(TRST*1)	PE9/TIOC3B/SCK3/RTS3	PE9/(TRST*1)	PE9/TIOC3B/SCK3/RTS3
175	PE10/(TDI*1)	PE10/TIOC3C/TXD2/SSO	PE10/(TDI*1)	PE10/TIOC3C/TXD2/SSO
176	PE11/(TDO*1)	PE11/TIOC3D/RXD3/CTS3	PE11/(TDO*1)	PE11/TIOC3D/RXD3/CTS3
1	PE12/(TCK*1)	PE12/TIOC4A/TXD3/ $\overline{\text{SCS}}$	PE12/(TCK*1)	PE12/TIOC4A/TXD3/ $\overline{\text{SCS}}$
2	PE13/(ASEBRKAK $\overline{\text{ASEBRK}}$ *1)	PE13/TIOC4B/MRES	PE13/(ASEBRKAK $\overline{\text{ASEBRK}}$ *1)	PE13/TIOC4B/MRES
4	PE14	PE14/ $\overline{\text{WRHH}}$ / $\overline{\text{CIOWR}}$ / $\overline{\text{AH}}$ / DQMUU/DACK0/TIOC4C	PE14	PE14/ $\overline{\text{WRHH}}$ / $\overline{\text{CIOWR}}$ / $\overline{\text{AH}}$ / DQMUU/DACK0/TIOC4C

引脚号	引脚名			
	内部 ROM 无效 (MCU 模式 0)		内部 ROM 无效 (MCU 模式 1)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
7	PE15	PE15/CKE/DACK1/TIOC4D /IRQOUT	PE15	PE15/CKE/DACK1/TIOC4D /IRQOUT
9	PE16	PE16/CS8/TIOC3BS	PE16	PE16/CS8/TIOC3BS
10	PE17	PE17/TIOC3DS	PE17	PE17/TIOC3DS
12	PE18	PE18/TIOC4AS	PE18	PE18/TIOC4AS
13	PE19	PE19/TIOC4BS	PE19	PE19/TIOC4BS
14	PE20	PE20/TIOC4CS	PE20	PE20/TIOC4CS
15	PE21	PE21/TIOC4DS	PE21	PE21/TIOC4DS
143	PF0/AN0	PF0/AN0	PF0/AN0	PF0/AN0
144	PF1/AN1	PF1/AN1	PF1/AN1	PF1/AN1
147	PF2/AN2	PF2/AN2	PF2/AN2	PF2/AN2
148	PF3/AN3	PF3/AN3	PF3/AN3	PF3/AN3
152	PF4/AN4	PF4/AN4	PF4/AN4	PF4/AN4
153	PF5/AN5	PF5/AN5	PF5/AN5	PF5/AN5
157	PF6/AN6	PF6/AN6	PF6/AN6	PF6/AN6
158	PF7/AN7	PF7/AN7	PF7/AN7	PF7/AN7
145	PF8/AN8	PF8/AN8	PF8/AN8	PF8/AN8
146	PF9/AN9	PF9/AN9	PF9/AN9	PF9/AN9
149	PF10/AN10	PF10/AN10	PF10/AN10	PF10/AN10
150	PF11/AN11	PF11/AN11	PF11/AN11	PF11/AN11
154	PF12/AN12	PF12/AN12	PF12/AN12	PF12/AN12
155	PF13/AN13	PF13/AN13	PF13/AN13	PF13/AN13
159	PF14/AN14	PF14/AN14	PF14/AN14	PF14/AN14
160	PF15/AN15	PF15/AN15	PF15/AN15	PF15/AN15

【注】 \*1 当使用 E10A (ASEMD0=Low 电平) 时, 这些引脚被固定为 TMS、TRST、TDI、TDO、TCK、ASEBRKAK/ASEBRK。

\*2 只限支持 E10A 全功能的 F-ZTAT 版本。当使用 E10A 的 AUD 功能时, 这些引脚被固定为 AUD 功能。

表 21.20 按运行模式分类的引脚功能一览表 (SH7086 (2))

引脚号	引脚名			
	内部 ROM 有效 (MCU 模式 2)		单芯片模式 (MCU 模式 3)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
11,21,35,48, 66,74,94, 102,109,128, 136,168	Vcc	Vcc	Vcc	Vcc
8,23,44,57, 64,86,96, 104,117, 141,163	Vss	Vss	Vss	Vss
37,71, 129,173	Vcl	Vcl	Vcl	Vcl
151,162	AVcc	AVcc	AVcc	AVcc
142,156	AVss	AVss	AVss	AVss
161	AVref	AVref	AVref	AVref
130	PLLvss	PLLvss	PLLvss	PLLvss
120	EXTAL	EXTAL	EXTAL	EXTAL
118	XTAL	XTAL	XTAL	XTAL
121	MD0	MD0	MD0	MD0
119	MD1	MD1	MD1	MD1
123	FWE	FWE	FWE	FWE
132	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$
53	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$
122	NMI	NMI	NMI	NMI
51	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$
164	PA0	PA0/ $\overline{\text{CS4}}$ /RXD0	PA0	PA0/RXD0
165	PA1	PA1/ $\overline{\text{CS5}}$ / $\overline{\text{CE1A}}$ /TXD0	PA1	PA1/TXD0
166	PA2	PA2/A25/DREQ0/IRQ0/SCK0	PA2	PA2/DREQ0/IRQ0/SCK0
167	PA3	PA3/A24/RXD1	PA3	PA3/RXD1
169	PA4	PA4/A23/TXD1	PA4	PA4/TXD1
170	PA5	PA5/A22/DREQ1/IRQ1/SCK1	PA5	PA5/IRQ1/SCK1
80	PA6	PA6/ $\overline{\text{CS2}}$ /TCLKA	PA6	PA6/TCLKA
79	PA7	PA7/ $\overline{\text{CS3}}$ /TCLKB	PA7	PA7/TCLKB
78	PA8	PA8/RDWR/IRQ2/TCLKC	PA8	PA8/IRQ2/TCLKC
77	PA9	PA9/ $\overline{\text{FRAME}}$ /CKE/IRQ3/TCLKD	PA9	PA9/RQ3/TCLKD
76	PA10	PA10/ $\overline{\text{CS0}}$ / $\overline{\text{POE4}}$	PA10	PA10/ $\overline{\text{POE4}}$
75	PA11	PA11/ $\overline{\text{CS1}}$ / $\overline{\text{POE5}}$	PA11	PA11/ $\overline{\text{POE5}}$

引脚号	引脚名			
	内部 ROM 有效 (MCU 模式 2)		单芯片模式 (MCU 模式 3)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
73	PA12	PA12/WRL/DQMLL/POE6	PA12	PA12/POE6
72	PA13	PA13/WRH/DQMLU/WE/POE7	PA13	PA13/POE7
52	PA14	PA14/RD	PA14	PA14
131	CK	PA15/CK	PA15	PA15
124	PA16/(AUDSYNC*2)	PA16/WRHH/ICIOWR/AH/ DQMUU/CKE/DREQ2	PA16/(AUDSYNC*2)	PA16/DREQ2
125	PA17	PA17/WAIT/DACK2	PA17	PA17
42	PA18	PA18/BREQ/TEND0	PA18	PA18
39	PA19	PA19/BACK/TEND1	PA19	PA19
38	PA20	PA20/CS4/RASU	PA20	PA20
6	PA21	PA21/CS5/CE1A/CASU/TIC5U	PA21	PA21/TIC5U
5	PA22	PA22/WRHL/ICIORD/DQMUL/ TIC5V	PA22	PA22/TIC5V
3	PA23	PA23/WRHH/ICIOWR/AH/ DQMUU/TIC5W	PA23	PA23/TIC5W
126	PA24	PA24/CE2A/DREQ3	PA24	PA24/DREQ3
127	PA25	PA25/CE2B/DACK3/POE8	PA25	PA25/POE8
63	PA26	PA26/A26/IRQ0	PA26	PA26/IRQ0
65	PA27	PA27/A27/IRQ1	PA27	PA27/IRQ1
67	PA28	PA28/A28/IRQ2	PA28	PA28/IRQ2
68	PA29	PA29/A29/IRQ3	PA29	PA29/IRQ3
34	PB0	PB0/A16/TIC5WS	PB0	PB0/TIC5WS
36	PB1	PB1/A17/TIC5W	PB1	PB1/TIC5W
40	PB2	PB2/IRQ0/POE0/SCL	PB2	PB2/IRQ0/POE0/SCL
41	PB3	PB3/IRQ1/POE1/SDA	PB3	PB3/IRQ1/POE1/SDA
43	PB4	PB4/RASL/IRQ2/POE2	PB4	PB4/IRQ2/POE2
45	PB5	PB5/CASL/IRQ3/POE3	PB5	PB5/IRQ3/POE3
46	PB6	PB6/A18/BACK/IRQ4/RXD0	PB6	PB6/IRQ4/RXD0
47	PB7	PB7/A19/BREQ/IRQ5/TXD0	PB7	PB7/IRQ5/TXD0
49	PB8	PB8/A20/WAIT/IRQ6/SCK0	PB8	PB8/IRQ6/SCK0
50	PB9	PB9/A21/IRQ7/ADTRG/POE8	PB9	PB9/IRQ7/ADTRG/POE8
16	PC0	PC0/A0	PC0	PC0
17	PC1	PC1/A1	PC1	PC1
18	PC2	PC2/A2	PC2	PC2
19	PC3	PC3/A3	PC3	PC3

引脚号	引脚名			
	内部 ROM 有效 (MCU 模式 2)		单芯片模式 (MCU 模式 3)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
20	PC4	PC4/A4	PC4	PC4
22	PC5	PC5/A5	PC5	PC5
24	PC6	PC6/A6	PC6	PC6
25	PC7	PC7/A7	PC7	PC7
26	PC8	PC8/A8	PC8	PC8
27	PC9	PC9/A9	PC9	PC9
28	PC10	PC10/A10	PC10	PC10
29	PC11	PC11/A11	PC11	PC11
30	PC12	PC12/A12	PC12	PC12
31	PC13	PC13/A13	PC13	PC13
32	PC14	PC14/A14	PC14	PC14
33	PC15	PC15/A15	PC15	PC15
54	PC18	PC18/A18	PC18	PC18
55	PC19	PC19/A19	PC19	PC19
56	PC20	PC20/A20	PC20	PC20
58	PC21	PC21/A21	PC21	PC21
59	PC22	PC22/A22	PC22	PC22
60	PC23	PC23/A23	PC23	PC23
61	PC24	PC24/A24	PC24	PC24
62	PC25	PC25/A25	PC25	PC25
116	PD0	PD0/D0	PD0	PD0
115	PD1	PD1/D1	PD1	PD1
114	PD2	PD2/D2/TIC5U	PD2	PD2/TIC5U
113	PD3	PD3/D3/TIC5V	PD3	PD3/TIC5V
112	PD4	PD4/D4/TIC5W	PD4	PD4/TIC5W
111	PD5	PD5/D5/TIC5US	PD5	PD5/TIC5US
110	PD6	PD6/D6/TIC5VS	PD6	PD6/TIC5VS
108	PD7	PD7/D7/TIC5WS	PD7	PD7/TIC5WS
107	PD8	PD8/D8/TIOC3AS	PD8	PD8/TIOC3AS
106	PD9	PD9/D9/TIOC3BS	PD9	PD9/TIOC3BS
105	PD10	PD10/D10/TIOC3CS	PD10	PD10/TIOC3CS
103	PD11	PD11/D11/TIOC3DS	PD11	PD11/TIOC3DS
101	PD12	PD12/D12/TIOC4AS	PD12	PD12/TIOC4AS
100	PD13	PD13/D13/TIOC4BS	PD13	PD13/TIOC4BS
99	PD14	PD14/D14/TIOC4CS	PD14	PD14/TIOC4CS

引脚号	引脚名			
	内部 ROM 有效 (MCU 模式 2)		单芯片模式 (MCU 模式 3)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
98	PD15	PD15/D15/TIOC4DS	PD15	PD15/TIOC4DS
97	PD16/(AUDATA0*2)	PD16/D16/IRQ0/POE4	PD16/(AUDATA0*2)	PD16/IRQ0/POE4
95	PD17/(AUDATA1*2)	PD17/D17/IRQ1/POE5	PD17/(AUDATA1*2)	PD17/IRQ1/POE5
93	PD18/(AUDATA2*2)	PD18/D18/IRQ2/POE6	PD18/(AUDATA2*2)	PD18/IRQ2/POE6
92	PD19/(AUDATA3*2)	PD19/D19/IRQ3/POE7	PD19/(AUDATA3*2)	PD19/IRQ3/POE7
91	PD20	PD20/D20/IRQ4/TIC5WS	PD20	PD20/IRQ4/TIC5WS
90	PD21	PD21/D21/IRQ5/TIC5VS	PD21	PD21/IRQ5/TIC5VS
89	PD22/(AUDCK*2)	PD22/D22/IRQ6/TIC5US	PD22/(AUDCK*2)	PD22/IRQ6/TIC5US
88	PD23/(AUDSYNC*2)	PD23/D23/IRQ7	PD23/(AUDSYNC*2)	PD23/IRQ7
87	PD24	PD24/D24/DREQ0/TIOC4DS	PD24	PD24/DREQ0/TIOC4DS
85	PD25	PD25/D25/DREQ1/TIOC4CS	PD25	PD25/DREQ1/TIOC4CS
84	PD26	PD26/D26/DACK0/TIOC4BS	PD26	PD26/TIOC4BS
83	PD27	PD27/D27/DACK1/TIOC4AS	PD27	PD27/TIOC4AS
82	PD28	PD28/D28/CS2/TIOC3DS	PD28	PD28/TIOC3DS
81	PD29	PD29/D29/CS3/TIOC3BS	PD29	PD29/TIOC3BS
70	PD30	PD30/D30/TIOC3CS/IRQOUT	PD30	PD30/TIOC3CS/IRQOUT
69	PD31	PD31/D31/TIOC3AS/ADTRG	PD31	PD31/TIOC3AS/ADTRG
133	PE0/(AUDCK*2)	PE0/DREQ0/TIOC0A	PE0/(AUDCK*2)	PE0/DREQ0/TIOC0A
134	PE1	PE1/TEND0/TIOC0B	PE1	PE1/TIOC0B
135	PE2	PE2/DREQ1/TIOC0C	PE2	PE2/DREQ1/TIOC0C
137	PE3/(AUDATA3*2)	PE3/TEND1/TIOC0D	PE3/(AUDATA3*2)	PE3/TIOC0D
138	PE4/(AUDATA2*2)	PE4/IOIS16/TIOC1A/RXD3	PE4/(AUDATA2*2)	PE4/TIOC1A/RXD3
139	PE5/(AUDATA1*2)	PE5/CS6/CE1B/TIOC1B/TXD3	PE5/(AUDATA1*2)	PE5/TIOC1B/TXD3
140	PE6/(AUDATA0*2)	PE6/CS7/TIOC2A/SCK3	PE6/(AUDATA0*2)	PE6/TIOC2A/SCK3
171	PE7	PE7/BS/TIOC2B/UBCTRG/ RXD2/SSI	PE7	PE7/TIOC2B/UBCTRG/ RXD2/SSI
172	PE8/(TMS*1)	PE8/TIOC3A/SCK2/SSCK	PE8(TMS*1)	PE8/TIOC3A/SCK2/SSCK
174	PE9/(TRST*1)	PE9/TIOC3B/SCK3/RTS3	PE9/(TRST*1)	PE9/TIOC3B/SCK3/RTS3
175	PE10/(TDI*1)	PE10/TIOC3C/TXD2/SSO	PE10/(TDI*1)	PE10/TIOC3C/TXD2/SSO
176	PE11/(TDO*1)	PE11/TIOC3D/RXD3/CTS3	PE11/(TDO*1)	PE11/TIOC3D/RXD3/CTS3
1	PE12/(TCK*1)	PE12/TIOC4A/TXD3/SCS	PE12/(TCK*1)	PE12/TIOC4A/TXD3/SCS
2	PE13/(ASEBRKAK /ASEBRK*1)	PE13/TIOC4B/MRES	PE13/(ASEBRKAK /ASEBRK*1)	PE13/TIOC4B/MRES
4	PE14	PE14/WRHH/ICIOWR/AH/ DQMUU/DACK0/TIOC4C	PE14	PE14/TIOC4C

引脚号	引脚名			
	内部 ROM 有效 (MCU 模式 2)		单芯片模式 (MCU 模式 3)	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
7	PE15	PE15/CKE/DACK1/TIOC4D /IRQOUT	PE15	PE15/TIOC4D /IRQOUT
9	PE16	PE16/CS8/TIOC3BS	PE16	PE16/TIOC3BS
10	PE17	PE17/TIOC3DS	PE17	PE17/TIOC3DS
12	PE18	PE18/TIOC4AS	PE18	PE18/TIOC4AS
13	PE19	PE19/TIOC4BS	PE19	PE19/TIOC4BS
14	PE20	PE20/TIOC4CS	PE20	PE20/TIOC4CS
15	PE21	PE21/TIOC4DS	PE21	PE21/TIOC4DS
143	PF0/AN0	PF0/AN0	PF0/AN0	PF0/AN0
144	PF1/AN1	PF1/AN1	PF1/AN1	PF1/AN1
147	PF2/AN2	PF2/AN2	PF2/AN2	PF2/AN2
148	PF3/AN3	PF3/AN3	PF3/AN3	PF3/AN3
152	PF4/AN4	PF4/AN4	PF4/AN4	PF4/AN4
153	PF5/AN5	PF5/AN5	PF5/AN5	PF5/AN5
157	PF6/AN6	PF6/AN6	PF6/AN6	PF6/AN6
158	PF7/AN7	PF7/AN7	PF7/AN7	PF7/AN7
145	PF8/AN8	PF8/AN8	PF8/AN8	PF8/AN8
146	PF9/AN9	PF9/AN9	PF9/AN9	PF9/AN9
149	PF10/AN10	PF10/AN10	PF10/AN10	PF10/AN10
150	PF11/AN11	PF11/AN11	PF11/AN11	PF11/AN11
154	PF12/AN12	PF12/AN12	PF12/AN12	PF12/AN12
155	PF13/AN13	PF13/AN13	PF13/AN13	PF13/AN13
159	PF14/AN14	PF14/AN14	PF14/AN14	PF14/AN14
160	PF15/AN15	PF15/AN15	PF15/AN15	PF15/AN15

【注】 \*1 当使用 E10A (ASEMD0=Low 电平) 时, 这些引脚被固定为 TMS、TRST、TDI、TDO、TCK、ASEBRKAK/ASEBRK。

\*2 只限支持 E10A 全功能的 F-ZTAT 版本。当使用 E10A 的 AUD 功能时, 这些引脚被固定为 AUD 功能。



## 21.1 寄存器说明

PFC 有以下寄存器，有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 27 章 寄存器一览”。

表 21.21 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
端口 A 的 IO 寄存器 H	PAIORH	R/W	H'0000	H'FFFFD104	8、16、32
端口 A 的 IO 寄存器 L	PAIORL	R/W	H'0000	H'FFFFD106	8、16
端口 A 的控制寄存器 H4	PACRH4	R/W	H'0000	H'FFFFD108	8、16、32
端口 A 的控制寄存器 H3	PACRH3	R/W	H'0000	H'FFFFD10A	8、16
端口 A 的控制寄存器 H2	PACRH2	R/W	H'0000*	H'FFFFD10C	8、16、32
端口 A 的控制寄存器 H1	PACRH1	R/W	H'0000	H'FFFFD10E	8、16
端口 A 的控制寄存器 L4	PACRL4	R/W	H'0000*	H'FFFFD110	8、16、32
端口 A 的控制寄存器 L3	PACRL3	R/W	H'0000*	H'FFFFD112	8、16
端口 A 的控制寄存器 L2	PACRL2	R/W	H'0000	H'FFFFD114	8、16、32
端口 A 的控制寄存器 L1	PACRL1	R/W	H'0000	H'FFFFD116	8、16
端口 B 的 IO 寄存器 L	PBIORL	R/W	H'0000	H'FFFFD186	8、16
端口 B 的控制寄存器 L3	PBCRL3	R/W	H'0000	H'FFFFD192	8、16
端口 B 的控制寄存器 L2	PBCRL2	R/W	H'0000	H'FFFFD194	8、16、32
端口 B 的控制寄存器 L1	PBCRL1	R/W	H'0000*	H'FFFFD196	8、16
端口 C 的 IO 寄存器 H	PCIORH	R/W	H'0000	H'FFFFD204	8、16、32
端口 C 的 IO 寄存器 L	PCIORL	R/W	H'0000	H'FFFFD206	8、16
端口 C 的控制寄存器 H3	PCCR3H3	R/W	H'0000*	H'FFFFD20A	8、16
端口 C 的控制寄存器 H2	PCCR3H2	R/W	H'0000*	H'FFFFD20C	8、16、32
端口 C 的控制寄存器 H1	PCCR3H1	R/W	H'0000*	H'FFFFD20E	8、16
端口 C 的控制寄存器 L4	PCCR3L4	R/W	H'0000*	H'FFFFD210	8、16、32
端口 C 的控制寄存器 L3	PCCR3L3	R/W	H'0000*	H'FFFFD212	8、16
端口 C 的控制寄存器 L2	PCCR3L2	R/W	H'0000*	H'FFFFD214	8、16、32
端口 C 的控制寄存器 L1	PCCR3L1	R/W	H'0000*	H'FFFFD216	8、16
端口 D 的 IO 寄存器 H	PDIORH	R/W	H'0000	H'FFFFD284	8、16、32
端口 D 的 IO 寄存器 L	PDIORL	R/W	H'0000	H'FFFFD286	8、16
端口 D 的控制寄存器 H4	PDCRH4	R/W	H'0000*	H'FFFFD288	8、16、32
端口 D 的控制寄存器 H3	PDCRH3	R/W	H'0000*	H'FFFFD28A	8、16
端口 D 的控制寄存器 H2	PDCRH2	R/W	H'0000*	H'FFFFD28C	8、16、32
端口 D 的控制寄存器 H1	PDCRH1	R/W	H'0000*	H'FFFFD28E	8、16
端口 D 的控制寄存器 L4	PDCRL4	R/W	H'0000*	H'FFFFD290	8、16、32
端口 D 的控制寄存器 L3	PDCRL3	R/W	H'0000*	H'FFFFD292	8、16
端口 D 的控制寄存器 L2	PDCRL2	R/W	H'0000*	H'FFFFD294	8、16、32
端口 D 的控制寄存器 L1	PDCRL1	R/W	H'0000*	H'FFFFD296	8、16

寄存器名	略称	R/W	初始值	地址	存取长度
端口 E 的 IO 寄存器 H	PEIORH	R/W	H'0000	H'FFFFD304	8、16、32
端口 E 的 IO 寄存器 L	PEIORL	R/W	H'0000	H'FFFFD306	8、16
端口 E 的控制寄存器 H2	PECRH2	R/W	H'0000	H'FFFFD30C	8、16、32
端口 E 的控制寄存器 H1	PECRH1	R/W	H'0000	H'FFFFD30E	8、16
端口 E 的控制寄存器 L4	PECRL4	R/W	H'0000	H'FFFFD310	8、16、32
端口 E 的控制寄存器 L3	PECRL3	R/W	H'0000	H'FFFFD312	8、16
端口 E 的控制寄存器 L2	PECRL2	R/W	H'0000	H'FFFFD314	8、16、32
端口 E 的控制寄存器 L1	PECRL1	R/W	H'0000	H'FFFFD316	8、16
大电流端口的控制寄存器	HCPCR	R/W	H'000F	H'FFFFD320	8、16、32
IRQOUT 功能的控制寄存器	IFCR	R/W	H'0000	H'FFFFD322	8、16

【注】 \* 寄存器的初始值因运行模式的设定和产品而不同，有关详细内容请参照本章的各寄存器说明。

### 21.1.1 端口 A 的 IO 寄存器 L、H (PAIORL、PAIORH)

PAIORL 和 PAIORH 是 16 位可读写寄存器，选择端口 A 的引脚输入 / 输出方向。PA29IOR ~ PA0IOR 位分别对应 PA29 ~ PA0 引脚（省略引脚名中端口以外的多路复用引脚名）。PAIORL 和 PAIORH 在端口 A 的引脚功能为通用输入 / 输出（PA15 ~ PA0 和 PA29 ~ PA16）时有效，否则无效。

如果将 PAIORL 和 PAIORH 的位设定为 1，对应的引脚就为输出引脚；如果置 0 就为输入引脚。但是在 SH7083 中，PAIORH 的 bit13 ~ 0 以及 PAIORL 的 bit11、bit6 和 bit2 ~ 0 无效；在 SH7084 中，PAIORH 的 bit13 ~ 2 无效；在 SH7085 中，PAIORH 的 bit13 ~ 10 无效。PAIORH 的 bit15 和 bit14 是保留位，读写的值总是 0。PAIORL 和 PAIORH 的初始值都是 H'0000。

#### (1) 端口 A 的 IO 寄存器 H (PAIORH)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PA29 IOR	PA28 IOR	PA27 IOR	PA26 IOR	PA25 IOR	PA24 IOR	PA23 IOR	PA22 IOR	PA21 IOR	PA20 IOR	PA19 IOR	PA18 IOR	PA17 IOR	PA16 IOR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

#### (2) 端口 A 的 IO 寄存器 L (PAIORL)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15 IOR	PA14 IOR	PA13 IOR	PA12 IOR	PA11 IOR	PA10 IOR	PA9 IOR	PA8 IOR	PA7 IOR	PA6 IOR	PA5 IOR	PA4 IOR	PA3 IOR	PA2 IOR	PA1 IOR	PA0 IOR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

### 21.1.2 端口 A 的控制寄存器 L1 ~ L4、H1 ~ H4 (PACRL1 ~ PACRL4、PACRH1 ~ PACRH4)

PACRL1 ~ PACRL4 和 PACRH1 ~ PACRH4 是 16 位可读写寄存器，选择端口 A 的多路复用引脚功能。

#### (1) SH7083

- 端口 A 的控制寄存器 H4 ~ H1 (PACRH4 ~ PACRH1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 0	—	全 0	R	保留位 读写的值总是 0。

• 端口 A 的控制寄存器 L4 (PACRL4)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA15 MD2	PA15 MD1	PA15 MD0	-	PA14 MD2	PA14 MD1	PA14 MD0	-	PA13 MD2	PA13 MD1	PA13 MD0	-	PA12 MD2	PA12 MD1	PA12 MD0
初始值:	0	0	0	0*1	0	0	0	0*2	0	0	0	0*2	0	0	0	0*2
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PA15MD2	0	R/W	PA15 模式位 选择 PA15/CK 引脚的功能。 000: PA15 输入 / 输出 (端口) 001: CK 输出 (CPG) *3 其他: 禁止设定
13	PA15MD1	0	R/W	
12	PA15MD0	0*1	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PA14MD2	0	R/W	PA14 模式位 选择 PA14/RD 引脚的功能。 000: PA14 输入 / 输出 (端口) 001: RD 输出 (BSC) *3 其他: 禁止设定
9	PA14MD1	0	R/W	
8	PA14MD0	0*2	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PA13MD2	0	R/W	PA13 模式位 选择 PA13/WRH/DQMLU/POE7 引脚的功能。 000: PA13 输入 / 输出 (端口) 001: WRH/DQMLU 输出 (BSC) *3 011: POE7 输入 (POE) 其他: 禁止设定
5	PA13MD1	0	R/W	
4	PA13MD0	0*2	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PA12MD2	0	R/W	PA12 模式位 选择 PA12/WRL/DQMLL/POE6 引脚的功能。 000: PA12 输入 / 输出 (端口) 001: WRL/DQMLL 输出 (BSC) *3 011: POE6 输入 (POE) 其他: 禁止设定
1	PA12MD1	0	R/W	
0	PA12MD0	0*2	R/W	

【注】 \*1 在内部 ROM 有效或者无效的外部扩展模式中，初始值是 1。

\*2 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*3 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 A 的控制寄存器 L3 (PACRL3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	PA10 MD2	PA10 MD1	PA10 MD0	-	PA9 MD2	PA9 MD1	PA9 MD0	-	PA8 MD2	PA8 MD1	PA8 MD0
初始值:	0	0	0	0	0	0	0	0*1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 11	—	全 0	R	保留位 读写的值总是 0。
10	PA10MD2	0	R/W	PA10 模式位 选择 PA10/CS0/POE4 引脚的功能。 000: PA10 输入 / 输出 (端口) 001: CS0 输出 (BSC) *2 011: POE4 输入 (POE) 其他: 禁止设定
9	PA10MD1	0	R/W	
8	PA10MD0	0*1	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PA9MD2	0	R/W	PA9 模式位 选择 PA9/CKE/IRQ3/TCLKD 引脚的功能。 000: PA9 输入 / 输出 (端口) 001: TCLKD 输入 (MTU2) 010: IRQ3 输入 (INTC) 101: CKE 输出 (BSC) *2 其他: 禁止设定
5	PA9MD1	0	R/W	
4	PA9MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PA8MD2	0	R/W	PA8 模式位 选择 PA8/RDWR/IRQ2/TCLKC 引脚的功能。 000: PA8 输入 / 输出 (端口) 001: TCLKC 输入 (MTU2) 010: IRQ2 输入 (INTC) 101: RDWR 输出 (BSC) *2 其他: 禁止设定
1	PA8MD1	0	R/W	
0	PA8MD0	0	R/W	

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 A 的控制寄存器 L2 (PACRL2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA7 MD2	PA7 MD1	PA7 MD0	-	-	-	-	-	PA5 MD2	PA5 MD1	PA5 MD0	-	PA4 MD2	PA4 MD1	PA4 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PA7MD2	0	R/W	PA7 模式位 选择 PA7/CS3/TCLKB 引脚的功能。 000: PA7 输入 / 输出 (端口) 001: TCLKB 输入 (MTU2) 010: CS3 输出 (BSC) * 其他: 禁止设定
13	PA7MD1	0	R/W	
12	PA7MD0	0	R/W	
11 ~ 7	—	全 0	R	保留位 读写的值总是 0。
6	PA5MD2	0	R/W	PA5 模式位 选择 PA5/A22/DREQ1/IRQ1/SCK1 引脚的功能。 000: PA5 输入 / 输出 (端口) 001: SCK1 输入 / 输出 (SCI) 010: DREQ1 输入 (DMAC) 011: IRQ1 输入 (INTC) 101: A22 输出 (BSC) * 其他: 禁止设定
5	PA5MD1	0	R/W	
4	PA5MD0	0	R/W	
3	—	0	R	
2	PA4MD2	0	R/W	PA4 模式位 选择 PA4/A23/TXD1 引脚的功能。 000: PA4 输入 / 输出 (端口) 001: TXD1 输出 (SCI) 101: A23 输出 (BSC) * 其他: 禁止设定
1	PA4MD1	0	R/W	
0	PA4MD0	0	R/W	

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 A 的控制寄存器 L1 (PACRL1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA3 MD2	PA3 MD1	PA3 MD0	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PA3MD2	0	R/W	PA3 模式位 选择 PA3/A24/RXD1 引脚的功能。 000: PA3 输入 / 输出 (端口) 001: RXD1 输出 (SCI) 101: A24 输出 (BSC) * 其他: 禁止设定
13	PA3MD1	0	R/W	
12	PA3MD0	0	R/W	
11 ~ 0	—	全 0	R	保留位 读写的值总是 0。

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

(2) SH7084

• 端口 A 的控制寄存器 H4 ~ H2 (PACRH4 ~ PACRH2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 0	—	全 0	R	保留位 读写的值总是 0。

• 端口 A 的控制寄存器 H1 (PACRH1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PA17 MD1	PA17 MD0	-	PA16 MD2	PA16 MD1	PA16 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 6	—	全 0	R	保留位 读写的值总是 0。
5 4	PA17MD1 PA17MD0	0 0	R/W R/W	PA17 模式位 选择 PA17/WAIT 引脚的功能。 00: PA17 输入 / 输出 (端口) 01: WAIT 输入 (BSC) * 其他: 禁止设定
3	—	0	R	保留位 读写的值总是 0。
2 1 0	PA16MD2 PA16MD1 PA16MD0	0 0 0	R/W R/W R/W	PA16 模式位 选择 PA16/AH/CKE 引脚的功能。 000: PA16 输入 / 输出 (端口) 001: AH 输出 (BSC) * 101: CKE 输出 (BSC) * 其他: 禁止设定

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。



• 端口 A 的控制寄存器 L4 (PACRL4)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA15 MD2	PA15 MD1	PA15 MD0	-	PA14 MD2	PA14 MD1	PA14 MD0	-	PA13 MD2	PA13 MD1	PA13 MD0	-	PA12 MD2	PA12 MD1	PA12 MD0
初始值:	0	0	0	0*1	0	0	0	0*2	0	0	0	0*2	0	0	0	0*2
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PA15MD2	0	R/W	PA15 模式位 选择 PA15/CK 引脚的功能。 000: PA15 输入 / 输出 (端口) 001: CK 输出 (CPG) *3 其他: 禁止设定
13	PA15MD1	0	R/W	
12	PA15MD0	0*1	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PA14MD2	0	R/W	PA14 模式位 选择 PA14/RD 引脚的功能。 000: PA14 输入 / 输出 (端口) 001: RD 输出 (BSC) *3 其他: 禁止设定
9	PA14MD1	0	R/W	
8	PA14MD0	0*2	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PA13MD2	0	R/W	PA13 模式位 选择 PA13/WRH/DQMLU/POE7 引脚的功能。 000: PA13 输入 / 输出 (端口) 001: WRH/DQMLU 输出 (BSC) *3 011: POE7 输入 (POE) 其他: 禁止设定
5	PA13MD1	0	R/W	
4	PA13MD0	0*2	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PA12MD2	0	R/W	PA12 模式位 选择 PA12/WRL/DQMLL/POE6 引脚的功能。 000: PA12 输入 / 输出 (端口) 001: WRL/DQMLL 输出 (BSC) *3 011: POE6 输入 (POE) 其他: 禁止设定
1	PA12MD1	0	R/W	
0	PA12MD0	0*2	R/W	

【注】 \*1 在内部 ROM 有效或者无效的外部扩展模式中，初始值是 1。

\*2 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*3 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

## • 端口 A 的控制寄存器 L3 (PACRL3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA11 MD2	PA11 MD1	PA11 MD0	-	PA10 MD2	PA10 MD1	PA10 MD0	-	PA9 MD2	PA9 MD1	PA9 MD0	-	PA8 MD2	PA8 MD1	PA8 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PA11MD2	0	R/W	PA11 模式位 选择 PA11/CS1/POE5 引脚的功能。 000: PA11 输入 / 输出 (端口) 001: CS1 输出 (BSC) *3 011: POE5 输入 (POE) *2 其他: 禁止设定
13	PA11MD1	0	R/W	
12	PA11MD0	0*1	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PA10MD2	0	R/W	PA10 模式位 选择 PA10/CS0/POE4 引脚的功能。 000: PA10 输入 / 输出 (端口) 001: CS0 输出 (BSC) *3 011: POE4 输入 (POE) 其他: 禁止设定
9	PA10MD1	0	R/W	
8	PA10MD0	0*1	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PA9MD2	0	R/W	PA9 模式位 选择 PA9/CKE/IRQ3/TCLKD 引脚的功能。 000: PA9 输入 / 输出 (端口) 001: TCLKD 输入 (MTU2) 010: IRQ3 输入 (INTC) 101: CKE 输出 (BSC) *3 其他: 禁止设定
5	PA9MD1	0	R/W	
4	PA9MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PA8MD2	0	R/W	PA8 模式位 选择 PA8/RDWR/IRQ2/TCLKC 引脚的功能。 000: PA8 输入 / 输出 (端口) 001: TCLKC 输入 (MTU2) 010: IRQ2 输入 (INTC) 101: RDWR 输出 (BSC) *3 其他: 禁止设定
1	PA8MD1	0	R/W	
0	PA8MD0	0	R/W	

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 如果设定为  $\overline{POE5}$  输入，以后就不能更改此设定。

\*3 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 A 的控制寄存器 L2 (PACRL2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA7 MD2	PA7 MD1	PA7 MD0	-	PA6 MD2	PA6 MD1	PA6 MD0	-	PA5 MD2	PA5 MD1	PA5 MD0	-	PA4 MD2	PA4 MD1	PA4 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PA7MD2	0	R/W	PA7 模式位 选择 PA7/ $\overline{CS3}$ /TCLKB 引脚的功能。 000: PA7 输入 / 输出 (端口) 001: TCLKB 输入 (MTU2) 010: $\overline{CS3}$ 输出 (BSC) * 其他: 禁止设定
13	PA7MD1	0	R/W	
12	PA7MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PA6MD2	0	R/W	PA6 模式位 选择 PA6/ $\overline{CS2}$ /TCLKA 引脚的功能。 000: PA6 输入 / 输出 (端口) 001: TCLKA 输入 (MTU2) 010: $\overline{CS2}$ 输出 (BSC) * 其他: 禁止设定
9	PA6MD1	0	R/W	
8	PA6MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PA5MD2	0	R/W	PA5 模式位 选择 PA5/A22/DREQ1/IRQ1/SCK1 引脚的功能。 000: PA5 输入 / 输出 (端口) 001: SCK1 输入 / 输出 (SCI) 010: DREQ1 输入 (DMAC) 011: IRQ1 输入 (INTC) 101: A22 输出 (BSC) * 其他: 禁止设定
5	PA5MD1	0	R/W	
4	PA5MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PA4MD2	0	R/W	PA4 模式位 选择 PA4/A23/TXD1 引脚的功能。 000: PA4 输入 / 输出 (端口) 001: TXD1 输出 (SCI) 101: A23 输出 (BSC) * 其他: 禁止设定
1	PA4MD1	0	R/W	
0	PA4MD0	0	R/W	

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 A 的控制寄存器 L1 (PACRL1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA3 MD2	PA3 MD1	PA3 MD0	-	PA2 MD2	PA2 MD1	PA2 MD0	-	PA1 MD2	PA1 MD1	PA1 MD0	-	PA0 MD2	PA0 MD1	PA0 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PA3MD2	0	R/W	PA3 模式位 选择 PA3/A24/RXD1 引脚的功能。 000: PA3 输入 / 输出 (端口) 001: RXD1 输入 (SCI) 101: A24 输出 (BSC) * 其他: 禁止设定
13	PA3MD1	0	R/W	
12	PA3MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PA2MD2	0	R/W	PA2 模式位 选择 PA2/A25/DREQ0/IRQ0/SCK0 引脚的功能。 000: PA2 输入 / 输出 (端口) 001: SCK0 输入 / 输出 (SCI) 010: DREQ0 输入 (DMAC) 011: IRQ0 输入 (INTC) 101: A25 输出 (BSC) * 其他: 禁止设定
9	PA2MD1	0	R/W	
8	PA2MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PA1MD2	0	R/W	PA1 模式位 选择 PA1/CS5/TXD0 引脚的功能。 000: PA1 输入 / 输出 (端口) 001: TXD0 输出 (SCI) 101: CS5 输出 (BSC) * 其他: 禁止设定
5	PA1MD1	0	R/W	
4	PA1MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PA0MD2	0	R/W	PA0 模式位 选择 PA0/CS4/RXD0 引脚的功能。 000: PA0 输入 / 输出 (端口) 001: RXD0 输入 (SCI) 101: CS4 输出 (BSC) * 其他: 禁止设定
1	PA0MD1	0	R/W	
0	PA0MD0	0	R/W	

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

(3) SH7085

• 端口 A 的控制寄存器 H4 (PACRH4)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 0	—	全 0	R	保留位 读写的值总是 0。

• 端口 A 的控制寄存器 H3 (PACRH3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PA25 MD1	PA25 MD0	-	-	PA24 MD1	PA24 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 6	—	全 0	R	保留位 读写的值总是 0。
5 4	PA25MD1 PA25MD0	0 0	R/W R/W	PA25 模式位 选择 PA25/ $\overline{\text{CE2B}}$ /DACK3/ $\overline{\text{POE8}}$ 引脚的功能。 00: PA25 输入 / 输出 (端口) 01: $\overline{\text{CE2B}}$ 输出 (BSC) * 10: DACK3 输出 (DMAC) * 11: $\overline{\text{POE8}}$ 输入 (POE)
3、2	—	全 0	R	保留位 读写的值总是 0。
1 0	PA24MD1 PA24MD0	0 0	R/W R/W	PA24 模式位 选择 PA24/ $\overline{\text{CE2A}}$ /DREQ3 引脚的功能。 00: PA24 输入 / 输出 (端口) 01: $\overline{\text{CE2A}}$ 输出 (BSC) * 10: DREQ3 输入 (DMAC) 其他: 禁止设定

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 A 的控制寄存器 H2 (PACRH2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PA23 MD1	PA23 MD0	-	-	PA22 MD1	PA22 MD0	-	-	PA21 MD1	PA21 MD0	-	-	PA20 MD1	PA20 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
15、14	—	全 0	R	保留位 读写的值总是 0。
13 12	PA23MD1 PA23MD0	0 0*1	R/W R/W	PA23 模式位 选择 PA23/WRHH/ICIOWR/AH/DQMUU/TIC5W 引脚的功能。 00: PA23 输入 / 输出 (端口) 01: WRHH/ICIOWR/AH/DQMUU 输出 (BSC) *2 11: TIC5W 输入 (MTU2) 其他: 禁止设定
11、10	—	全 0	R	保留位 读写的值总是 0。
9 8	PA22MD1 PA22MD0	0 0*1	R/W R/W	PA22 模式位 选择 PA22/WRHL/ICIORD/DQMUL/TIC5V 引脚的功能。 00: PA22 输入 / 输出 (端口) 01: WRHL/ICIORD/DQMUL 输出 (BSC) *2 11: TIC5V 输入 (MTU2) 其他: 禁止设定
7、6	—	全 0	R	保留位 读写的值总是 0。
5 4	PA21MD1 PA21MD0	0 0	R/W R/W	PA21 模式位 选择 PA21/CS5/CE1A/CASU/TIC5U 引脚的功能。 00: PA21 输入 / 输出 (端口) 01: CS5/CE1A 输出 (BSC) *2 10: CASU 输出 (BSC) *2 11: TIC5U 输入 (MTU2)
3、2	—	全 0	R	保留位 读写的值总是 0。
1 0	PA20MD1 PA20MD0	0 0	R/W R/W	PA20 模式位 选择 PA20/CS4/RASU 引脚的功能。 00: PA20 输入 / 输出 (端口) 01: CS4 输出 (BSC) *2 10: RASU 输出 (BSC) *2 其他: 禁止设定

【注】 \*1 在内部 ROM 无效的 32 位外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 A 的控制寄存器 H1 (PACRH1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PA19 MD1	PA19 MD0	-	-	PA18 MD1	PA18 MD0	-	-	PA17 MD1	PA17 MD0	-	PA16 MD2	PA16 MD1	PA16 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15、14	—	全 0	R	保留位 读写的值总是 0。
13 12	PA19MD1 PA19MD0	0 0	R/W R/W	PA19 模式位 选择 PA19/BACK/TEND1 引脚的功能。 00: PA19 输入 / 输出 (端口) 01: $\overline{\text{BACK}}$ 输出 (BSC) * 10: TEND1 输出 (DMAC) * 其他: 禁止设定
11、10	—	全 0	R	保留位 读写的值总是 0。
9 8	PA18MD1 PA18MD0	0 0	R/W R/W	PA18 模式位 选择 PA18/BREQ/TEND0 引脚的功能。 00: PA18 输入 / 输出 (端口) 01: $\overline{\text{BREQ}}$ 输入 (BSC) * 10: TEND0 输出 (DMAC) * 其他: 禁止设定
7、6	—	全 0	R	保留位 读写的值总是 0。
5 4	PA17MD1 PA17MD0	0 0	R/W R/W	PA17 模式位 选择 PA17/WAIT/DACK2 引脚的功能。 00: PA17 输入 / 输出 (端口) 01: $\overline{\text{WAIT}}$ 输入 (BSC) * 10: DACK2 输出 (DMAC) * 其他: 禁止设定
3	—	0	R	保留位 读写的值总是 0。
2 1 0	PA16MD2 PA16MD1 PA16MD0	0 0 0	R/W R/W R/W	PA16 模式位 选择 PA16/ $\overline{\text{WRHH}}/\overline{\text{ICIOWR}}/\overline{\text{AH}}/\overline{\text{DQMUU}}/\overline{\text{CKE}}/\overline{\text{DREQ2}}/\overline{\text{AUDSYNC}}$ 引脚的功能。当使用 E10A 的 AUD 功能时, 此引脚被固定为 $\overline{\text{AUDSYNC}}$ 输出。 000: PA16 输入 / 输出 (端口) 001: $\overline{\text{WRHH}}/\overline{\text{ICIOWR}}/\overline{\text{AH}}/\overline{\text{DQMUU}}$ 输出 (BSC) * 010: DREQ2 输入 (DMAC) 101: CKE 输出 (BSC) * 其他: 禁止设定

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 A 的控制寄存器 L4 (PACRL4)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA15 MD2	PA15 MD1	PA15 MD0	-	PA14 MD2	PA14 MD1	PA14 MD0	-	PA13 MD2	PA13 MD1	PA13 MD0	-	PA12 MD2	PA12 MD1	PA12 MD0
初始值:	0	0	0	0*1	0	0	0	0*2	0	0	0	0*2	0	0	0	0*2
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PA15MD2	0	R/W	PA15 模式位 选择 PA15/CK 引脚的功能。 000: PA15 输入 / 输出 (端口) 001: CK 输出 (CPG) *3 其他: 禁止设定
13	PA15MD1	0	R/W	
12	PA15MD0	0*1	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PA14MD2	0	R/W	PA14 模式位 选择 PA14/ $\overline{RD}$ 引脚的功能。 000: PA14 输入 / 输出 (端口) 001: $\overline{RD}$ 输出 (BSC) *3 其他: 禁止设定
9	PA14MD1	0	R/W	
8	PA14MD0	0*2	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PA13MD2	0	R/W	PA13 模式位 选择 PA13/ $\overline{WRH/WE/DQMLU/POE7}$ 引脚的功能。 000: PA13 输入 / 输出 (端口) 001: $\overline{WRH/WE/DQMLU}$ 输出 (BSC) *3 011: $\overline{POE7}$ 输入 (POE) 其他: 禁止设定
5	PA13MD1	0	R/W	
4	PA13MD0	0*2	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PA12MD2	0	R/W	PA12 模式位 选择 PA12/ $\overline{WRL/DQMLL/POE6}$ 引脚的功能。 000: PA12 输入 / 输出 (端口) 001: $\overline{WRL/DQMLL}$ 输出 (BSC) *3 011: $\overline{POE6}$ 输入 (POE) 其他: 禁止设定
1	PA12MD1	0	R/W	
0	PA12MD0	0*2	R/W	

【注】 \*1 在内部 ROM 有效或者无效的外部扩展模式中，初始值是 1。

\*2 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*3 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。



• 端口 A 的控制寄存器 L3 (PACRL3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA11 MD2	PA11 MD1	PA11 MD0	-	PA10 MD2	PA10 MD1	PA10 MD0	-	PA9 MD2	PA9 MD1	PA9 MD0	-	PA8 MD2	PA8 MD1	PA8 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PA11MD2	0	R/W	PA11 模式位 选择 PA11/CS1/POE5 引脚的功能。 000: PA11 输入 / 输出 (端口) 001: CS1 输出 (BSC) *3 011: POE5 输入 (POE) *2 其他: 禁止设定
13	PA11MD1	0	R/W	
12	PA11MD0	0*1	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PA10MD2	0	R/W	PA10 模式位 选择 PA10/CS0/POE4 引脚的功能。 000: PA10 输入 / 输出 (端口) 001: CS0 输出 (BSC) *3 011: POE4 输入 (POE) 其他: 禁止设定
9	PA10MD1	0	R/W	
8	PA10MD0	0*1	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PA9MD2	0	R/W	PA9 模式位 选择 PA9/FRAME/CKE/IRQ3/TCLKD 引脚的功能 000: PA9 输入 / 输出 (端口) 001: TCLKD 输入 (MTU2) 010: IRQ3 输入 (INTC) 011: FRAME 输出 (BSC) *3 101: CKE 输出 (BSC) *3 其他: 禁止设定
5	PA9MD1	0	R/W	
4	PA9MD0	0	R/W	
3	—	0	R	
2	PA8MD2	0	R/W	PA8 模式位 选择 PA8/RDWR/IRQ2/TCLKC 引脚的功能。 000: PA8 输入 / 输出 (端口) 001: TCLKC 输入 (MTU2) 010: IRQ2 输入 (INTC) 101: RDWR 输出 (BSC) *3 其他: 禁止设定
1	PA8MD1	0	R/W	
0	PA8MD0	0	R/W	

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 如果设定为 POE5 输入，以后就不能更改此设定。

\*3 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 A 的控制寄存器 L2 (PACRL2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA7 MD2	PA7 MD1	PA7 MD0	-	PA6 MD2	PA6 MD1	PA6 MD0	-	PA5 MD2	PA5 MD1	PA5 MD0	-	PA4 MD2	PA4 MD1	PA4 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PA7MD2	0	R/W	PA7 模式位 选择 PA7/ $\overline{CS3}$ /TCLKB 引脚的功能。 000: PA7 输入 / 输出 (端口) 001: TCLKB 输入 (MTU2) 010: $\overline{CS3}$ 输出 (BSC) * 其他: 禁止设定
13	PA7MD1	0	R/W	
12	PA7MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PA6MD2	0	R/W	PA6 模式位 选择 PA6/ $\overline{CS2}$ /TCLKA1 引脚的功能。 000: PA6 输入 / 输出 (端口) 001: TCLKA 输入 (MTU2) 010: $\overline{CS2}$ 输出 (BSC) * 其他: 禁止设定
9	PA6MD1	0	R/W	
8	PA6MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PA5MD2	0	R/W	PA5 模式位 选择 PA5/A22/DREQ1/IRQ1/SCK1 引脚的功能。 000: PA5 输入 / 输出 (端口) 001: SCK1 输入 / 输出 (SCI) 010: DREQ1 输入 (DMAC) 011: IRQ1 输入 (INTC) 101: A22 输出 (BSC) * 其他: 禁止设定
5	PA5MD1	0	R/W	
4	PA5MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PA4MD2	0	R/W	PA4 模式位 选择 PA4/A23/TXD1 引脚的功能。 000: PA4 输入 / 输出 (端口) 001: TXD1 输出 (SCI) 101: A23 输出 (BSC) * 其他: 禁止设定
1	PA4MD1	0	R/W	
0	PA4MD0	0	R/W	

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 A 的控制寄存器 L1 (PACRL1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA3 MD2	PA3 MD1	PA3 MD0	-	PA2 MD2	PA2 MD1	PA2 MD0	-	PA1 MD2	PA1 MD1	PA1 MD0	-	PA0 MD2	PA0 MD1	PA0 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PA3MD2	0	R/W	PA3 模式位 选择 PA3/A24/RXD1 引脚的功能。 000: PA3 输入 / 输出 (端口) 001: RXD1 输入 (SCI) 101: A24 输出 (BSC) * 其他: 禁止设定
13	PA3MD1	0	R/W	
12	PA3MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PA2MD2	0	R/W	PA2 模式位 选择 PA2/A25/DREQ0/IRQ0/SCK0 引脚的功能。 000: PA2 输入 / 输出 (端口) 001: SCK0 输入 / 输出 (SCI) 010: DREQ0 输入 (DMAC) 011: IRQ0 输入 (INTC) 101: A25 输出 (BSC) * 其他: 禁止设定
9	PA2MD1	0	R/W	
8	PA2MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PA1MD2	0	R/W	PA1 模式位 选择 PA1/ <u>CS5</u> / <u>CE1A</u> /TXD0 引脚的功能。 000: PA1 输入 / 输出 (端口) 001: TXD0 输出 (SCI) 101: <u>CS5</u> / <u>CE1A</u> 输出 (BSC) * 其他: 禁止设定
5	PA1MD1	0	R/W	
4	PA1MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PA0MD2	0	R/W	PA0 模式位 选择 PA0/ <u>CS4</u> /RXD0 引脚的功能。 000: PA0 输入 / 输出 (端口) 001: RXD0 输入 (SCI) 101: <u>CS4</u> 输出 (BSC) * 其他: 禁止设定
1	PA0MD1	0	R/W	
0	PA0MD0	0	R/W	

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

(4) SH7086

• 端口 A 的控制寄存器 H4 (PACRH4)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PA29 MD1	PA29 MD0	-	-	PA28 MD1	PA28 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 6	—	全 0	R	保留位 读写的值总是 0。
5 4	PA29MD1 PA29MD0	0 0	R/W R/W	PA29 模式位 选择 PA29/A29/IRQ3 引脚的功能。 00: PA29 输入 / 输出 (端口) 01: A29 输出 (BSC) * 11: IRQ3 输入 (INTC) 其他: 禁止设定
3、2	—	全 0	R	保留位 读写的值总是 0。
1 0	PA28MD1 PA28MD0	0 0	R/W R/W	PA28 模式位 选择 PA28/A28/IRQ2 引脚的功能。 00: PA28 输入 / 输出 (端口) 01: A28 输出 (BSC) * 11: IRQ2 输入 (INTC) 其他: 禁止设定

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 A 的控制寄存器 H3 (PACRH3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PA27 MD1	PA27 MD0	-	-	PA26 MD1	PA26 MD0	-	-	PA25 MD1	PA25 MD0	-	-	PA24 MD1	PA24 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
15、14	—	全 0	R	保留位 读写的值总是 0。
13 12	PA27MD1 PA27MD0	0 0	R/W R/W	PA27 模式位 选择 PA27/A27/IRQ1 引脚的功能。 00: PA27 输入 / 输出 (端口) 01: A27 输出 (BSC) * 11: IRQ1 输入 (INTC) 其他: 禁止设定
11、10	—	全 0	R	保留位 读写的值总是 0。
9 8	PA26MD1 PA26MD0	0 0	R/W R/W	PA26 模式位 选择 PA26/A26/IRQ0 引脚的功能。 00: PA26 输入 / 输出 (端口) 01: A26 输出 (BSC) * 11: IRQ0 输入 (INTC) 其他: 禁止设定
7、6	—	全 0	R	保留位 读写的值总是 0。
5 4	PA25MD1 PA25MD0	0 0	R/W R/W	PA25 模式位 选择 PA25/CE2B/DACK3/POE8 引脚的功能。 00: PA25 输入 / 输出 (端口) 01: CE2B 输出 (BSC) * 10: DACK3 输出 (DMAC) * 11: POE8 输入 (POE)
3、2	—	全 0	R	保留位 读写的值总是 0。
1 0	PA24MD1 PA24MD0	0 0	R/W R/W	PA24 模式位 选择 PA24/CE2A/DREQ3 引脚的功能。 00: PA24 输入 / 输出 (端口) 01: CE2A 输出 (BSC) * 10: DREQ3 输入 (DMAC) 其他: 禁止设定

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 A 的控制寄存器 H2 (PACRH2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PA23 MD1	PA23 MD0	-	-	PA22 MD1	PA22 MD0	-	-	PA21 MD1	PA21 MD0	-	-	PA20 MD1	PA20 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
15、14	—	全 0	R	保留位 读写的值总是 0。
13 12	PA23MD1 PA23MD0	0 0*1	R/W R/W	PA23 模式位 选择 PA23/WRHH/ICIOWR/AH/DQMUU/TIC5W 引脚的功能。 00: PA23 输入 / 输出 (端口) 01: WRHH/ICIOWR/AH/DQMUU 输出 (BSC) *2 11: TIC5W 输入 (MTU2) 其他: 禁止设定
11、10	—	全 0	R	保留位 读写的值总是 0。
9 8	PA22MD1 PA22MD0	0 0*1	R/W R/W	PA22 模式位 选择 PA22/WRHL/ICIORD/DQMUL/TIC5V 引脚的功能。 00: PA22 输入 / 输出 (端口) 01: WRHL/ICIORD/DQMUL 输出 (BSC) *2 11: TIC5V 输入 (MTU2) 其他: 禁止设定
7、6	—	全 0	R	保留位 读写的值总是 0。
5 4	PA21MD1 PA21MD0	0 0	R/W R/W	PA21 模式位 选择 PA21/CS5/CE1A/CASU/TIC5U 引脚的功能。 00: PA21 输入 / 输出 (端口) 01: CS5/CE1A 输出 (BSC) *2 10: CASU 输出 (BSC) *2 11: TIC5U 输入 (MTU2)
3、2	—	全 0	R	保留位 读写的值总是 0。
1 0	PA20MD1 PA20MD0	0 0	R/W R/W	PA20 模式位 选择 PA20/CS4/RASU 引脚的功能。 00: PA20 输入 / 输出 (端口) 01: CS4 输出 (BSC) *2 10: RASU 输出 (BSC) *2 其他: 禁止设定

【注】 \*1 在内部 ROM 无效的 32 位外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 A 的控制寄存器 H1 (PACRH1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PA19 MD1	PA19 MD0	-	-	PA18 MD1	PA18 MD0	-	-	PA17 MD1	PA17 MD0	-	PA16 MD2	PA16 MD1	PA16 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15、14	—	全 0	R	保留位 读写的值总是 0。
13 12	PA19MD1 PA19MD0	0 0	R/W R/W	PA19 模式位 选择 PA19/BACK/TEND1 引脚的功能。 00: PA19 输入 / 输出 (端口) 01: $\overline{\text{BACK}}$ 输出 (BSC) * 10: TEND1 输出 (DMAC) * 其他: 禁止设定
11、10	—	全 0	R	保留位 读写的值总是 0。
9 8	PA18MD1 PA18MD0	0 0	R/W R/W	PA18 模式位 选择 PA18/BREQ/TEND0 引脚的功能。 00: PA18 输入 / 输出 (端口) 01: $\overline{\text{BREQ}}$ 输入 (BSC) * 10: TEND0 输出 (DMAC) * 其他: 禁止设定
7、6	—	全 0	R	保留位 读写的值总是 0。
5 4	PA17MD1 PA17MD0	0 0	R/W R/W	PA17 模式位 选择 PA17/WAIT/DACK2 引脚的功能。 00: PA17 输入 / 输出 (端口) 01: $\overline{\text{WAIT}}$ 输入 (BSC) * 10: DACK2 输出 (DMAC) * 其他: 禁止设定
3	—	0	R	保留位 读写的值总是 0。
2 1 0	PA16MD2 PA16MD1 PA16MD0	0 0 0	R/W R/W R/W	PA16 模式位 选择 PA16/ $\overline{\text{WRHH}}/\overline{\text{ICIOWR}}/\overline{\text{AH}}/\overline{\text{DQMUU}}/\overline{\text{CKE}}/\overline{\text{DREQ2}}/\overline{\text{AUDSYNC}}$ 引脚的功能。当使用 E10A 的 AUD 功能时, 此引脚被固定为 $\overline{\text{AUDSYNC}}$ 输出。 000: PA16 输入 / 输出 (端口) 001: $\overline{\text{WRHH}}/\overline{\text{ICIOWR}}/\overline{\text{AH}}/\overline{\text{DQMUU}}$ 输出 (BSC) * 010: DREQ2 输入 (DMAC) 101: CKE 输出 (BSC) * 其他: 禁止设定

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 A 的控制寄存器 L4 (PACRL4)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA15 MD2	PA15 MD1	PA15 MD0	-	PA14 MD2	PA14 MD1	PA14 MD0	-	PA13 MD2	PA13 MD1	PA13 MD0	-	PA12 MD2	PA12 MD1	PA12 MD0
初始值:	0	0	0	0*1	0	0	0	0*2	0	0	0	0*2	0	0	0	0*2
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PA15MD2	0	R/W	PA15 模式位 选择 PA15/CK 引脚的功能。 000: PA15 输入 / 输出 (端口) 001: CK 输出 (CPG) *3 其他: 禁止设定
13	PA15MD1	0	R/W	
12	PA15MD0	0*1	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PA14MD2	0	R/W	PA14 模式位 选择 PA14/ $\overline{RD}$ 引脚的功能。 000: PA14 输入 / 输出 (端口) 001: $\overline{RD}$ 输出 (BSC) *3 其他: 禁止设定
9	PA14MD1	0	R/W	
8	PA14MD0	0*2	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PA13MD2	0	R/W	PA13 模式位 选择 PA13/ $\overline{WRH/DQMLU/WE/POE7}$ 引脚的功能。 000: PA13 输入 / 输出 (端口) 001: $\overline{WRH/DQMLU/WE}$ 输出 (BSC) *3 011: $\overline{POE7}$ 输入 (POE) 其他: 禁止设定
5	PA13MD1	0	R/W	
4	PA13MD0	0*2	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PA12MD2	0	R/W	PA12 模式位 选择 PA12/ $\overline{WRL/DQMLL/POE6}$ 引脚的功能。 000: PA12 输入 / 输出 (端口) 001: $\overline{WRL/DQMLL}$ 输出 (BSC) *3 011: $\overline{POE6}$ 输入 (POE) 其他: 禁止设定
1	PA12MD1	0	R/W	
0	PA12MD0	0*2	R/W	

【注】 \*1 在内部 ROM 有效或者无效的外部扩展模式中，初始值是 1。

\*2 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*3 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。



## • 端口 A 的控制寄存器 L3 (PACRL3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA11 MD2	PA11 MD1	PA11 MD0	-	PA10 MD2	PA10 MD1	PA10 MD0	-	PA9 MD2	PA9 MD1	PA9 MD0	-	PA8 MD2	PA8 MD1	PA8 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PA11MD2	0	R/W	PA11 模式位 选择 PA11/ $\overline{\text{CS1}}$ / $\overline{\text{POE5}}$ 引脚的功能。 000: PA11 输入 / 输出 (端口) 001: $\overline{\text{CS1}}$ 输出 (BSC) *3 011: $\overline{\text{POE5}}$ 输入 (POE) *2 其他: 禁止设定
13	PA11MD1	0	R/W	
12	PA11MD0	0*1	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PA10MD2	0	R/W	PA10 模式位 选择 PA10/ $\overline{\text{CS0}}$ / $\overline{\text{POE4}}$ 引脚的功能。 000: PA10 输入 / 输出 (端口) 001: $\overline{\text{CS0}}$ 输出 (BSC) *3 011: $\overline{\text{POE4}}$ 输入 (POE) 其他: 禁止设定
9	PA10MD1	0	R/W	
8	PA10MD0	0*1	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PA9MD2	0	R/W	PA9 模式位 选择 PA9/ $\overline{\text{FRAME}}$ / $\overline{\text{CKE}}$ / $\overline{\text{IRQ3}}$ / $\overline{\text{TCLKD}}$ 引脚的功能。 000: PA9 输入 / 输出 (端口) 001: $\overline{\text{TCLKD}}$ 输入 (MTU2) 010: $\overline{\text{IRQ3}}$ 输入 (INTC) 011: $\overline{\text{FRAME}}$ 输出 (BSC) *3 101: $\overline{\text{CKE}}$ 输出 (BSC) *3 其他: 禁止设定
5	PA9MD1	0	R/W	
4	PA9MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PA8MD2	0	R/W	PA8 模式位 选择 PA8/ $\overline{\text{RDWR}}$ / $\overline{\text{IRQ2}}$ / $\overline{\text{TCLKC}}$ 引脚的功能。 000: PA8 输入 / 输出 (端口) 001: $\overline{\text{TCLKC}}$ 输入 (MTU2) 010: $\overline{\text{IRQ2}}$ 输入 (INTC) 101: $\overline{\text{RDWR}}$ 输出 (BSC) *3 其他: 禁止设定
1	PA8MD1	0	R/W	
0	PA8MD0	0	R/W	

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 如果设定为  $\overline{\text{POE5}}$  输入，以后就不能更改此设定。

\*3 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 A 的控制寄存器 L2 (PACRL2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA7 MD2	PA7 MD1	PA7 MD0	-	PA6 MD2	PA6 MD1	PA6 MD0	-	PA5 MD2	PA5 MD1	PA5 MD0	-	PA4 MD2	PA4 MD1	PA4 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PA7MD2	0	R/W	PA7 模式位 选择 PA7/ $\overline{CS3}$ /TCLKB 引脚的功能。 000: PA7 输入 / 输出 (端口) 001: TCLKB 输入 (MTU2) 010: $\overline{CS3}$ 输出 (BSC) * 其他: 禁止设定
13	PA7MD1	0	R/W	
12	PA7MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PA6MD2	0	R/W	PA6 模式位 选择 PA6/ $\overline{CS2}$ /TCLKA 引脚的功能。 000: PA6 输入 / 输出 (端口) 001: TCLKA 输入 (MTU2) 010: $\overline{CS2}$ 输出 (BSC) * 其他: 禁止设定
9	PA6MD1	0	R/W	
8	PA6MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PA5MD2	0	R/W	PA5 模式位 选择 PA5/A22/DREQ1/IRQ1/SCK1 引脚的功能。 000: PA5 输入 / 输出 (端口) 001: SCK1 输入 / 输出 (SCI) 010: DREQ1 输入 (DMAC) 011: IRQ1 输入 (INTC) 101: A22 输出 (BSC) * 其他: 禁止设定
5	PA5MD1	0	R/W	
4	PA5MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PA4MD2	0	R/W	PA4 模式位 选择 PA4/A23/TXD1 引脚的功能。 000: PA4 输入 / 输出 (端口) 001: TXD1 输出 (SCI) 101: A23 输出 (BSC) * 其他: 禁止设定
1	PA4MD1	0	R/W	
0	PA4MD0	0	R/W	

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 A 的控制寄存器 L1 (PACRL1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA3 MD2	PA3 MD1	PA3 MD0	-	PA2 MD2	PA2 MD1	PA2 MD0	-	PA1 MD2	PA1 MD1	PA1 MD0	-	PA0 MD2	PA0 MD1	PA0 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PA3MD2	0	R/W	PA3 模式位 选择 PA3/A24/RXD1 引脚的功能。 000: PA3 输入 / 输出 (端口) 001: RXD1 输入 (SCI) 101: A24 输出 (BSC) * 其他: 禁止设定
13	PA3MD1	0	R/W	
12	PA3MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PA2MD2	0	R/W	PA2 模式位 选择 PA2/A25/DREQ0/IRQ0/SCK0 引脚的功能。 000: PA2 输入 / 输出 (端口) 001: SCK0 输入 / 输出 (SCI) 010: DREQ0 输入 (DMAC) 011: IRQ0 输入 (INTC) 101: A25 输出 (BSC) * 其他: 禁止设定
9	PA2MD1	0	R/W	
8	PA2MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PA1MD2	0	R/W	PA1 模式位 选择 PA1/ <u>CS5</u> / <u>CE1A</u> /TXD0 引脚的功能。 000: PA1 输入 / 输出 (端口) 001: <u>TXD0</u> 输出 (SCI) 101: <u>CS5</u> / <u>CE1A</u> 输出 (BSC) * 其他: 禁止设定
5	PA1MD1	0	R/W	
4	PA1MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PA0MD2	0	R/W	PA0 模式位 选择 PA0/ <u>CS4</u> /RXD0 引脚的功能。 000: PA0 输入 / 输出 (端口) 001: <u>RXD0</u> 输入 (SCI) 101: <u>CS4</u> 输出 (BSC) * 其他: 禁止设定
1	PA0MD1	0	R/W	
0	PA0MD0	0	R/W	

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

### 21.1.3 端口 B 的 IO 寄存器 L (PBIORL)

PBIORL 是 16 位可读写寄存器，选择端口 B 的引脚输入 / 输出方向。PB9IOR ~ PB0IOR 位分别对应 PB9 引脚 ~ PB0 引脚（省略引脚名中端口以外的多路复用引脚名）。PBIORL 在端口 B 的引脚功能为通用输入 / 输出（PB9 ~ PB0）时有效，否则无效。

如果将 PBIORL 的位设定为 1，对应的引脚就为输出引脚；如果置 0 就为输入引脚。

但是在 SH7083 中，PBIORL 的 bit3 无效。

PBIORL 的 bit15 ~ 10 为保留位，读写的值总是 0。

PBIORL 的初始值是 H'0000。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PB9 IOR	PB8 IOR	PB7 IOR	PB6 IOR	PB5 IOR	PB4 IOR	PB3 IOR	PB2 IOR	PB1 IOR	PB0 IOR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

### 21.1.4 端口 B 的控制寄存器 L1 ~ L3 (PBCRL1 ~ PBCRL3)

PBCRL1 ~ PBCRL3 是 16 位可读写寄存器，选择端口 B 的多路复用引脚功能。

#### (1) SH7083

- 端口 B 的控制寄存器 L3 (PBCRL3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	PB9 MD2	PB9 MD1	PB9 MD0	-	PB8 MD2	PB8 MD1	PB8 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 7	—	全 0	R	保留位 读写的值总是 0。
6 5 4	PB9MD2 PB9MD1 PB9MD0	0 0 0	R/W R/W R/W	PB9 模式位 选择 PB9/A21/IRQ7/ADTRG/POE8 引脚的功能。 000: PB9 输入 / 输出（端口） 001: IRQ7 输入（INTC） 010: A21 输出（BSC）* 011: ADTRG 输入（A/D） 110: POE8 输入（POE） 其他: 禁止设定
3	—	0	R	保留位 读写的值总是 0。
2 1 0	PB8MD2 PB8MD1 PB8MD0	0 0 0	R/W R/W R/W	PB8 模式位 选择 PB8/A20/WAIT/IRQ6/SCK0 引脚的功能。 000: PB8 输入 / 输出（端口） 001: IRQ6 输入（INTC） 010: A20 输出（BSC）* 011: WAIT 输入（BSC）* 100: SCK0 输入 / 输出（SCI） 其他: 禁止设定

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 B 的控制寄存器 L2 (PBCRL2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PB7 MD2	PB7 MD1	PB7 MD0	-	PB6 MD2	PB6 MD1	PB6 MD0	-	PB5 MD2	PB5 MD1	PB5 MD0	-	PB4 MD2	PB4 MD1	PB4 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PB7MD2	0	R/W	PB7 模式位 选择 PB7/A19/BREQ/IRQ5/TXD0 引脚的功能。 000: PB7 输入 / 输出 (端口) 001: IRQ5 输入 (INTC) 010: A19 输出 (BSC) *1 011: BREQ 输入 (BSC) *1 100: TXD0 输出 (SCI) 其他: 禁止设定
13	PB7MD1	0	R/W	
12	PB7MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PB6MD2	0	R/W	PB6 模式位 选择 PB6/A18/BACK/IRQ4/RXD0 引脚的功能。 000: PB6 输入 / 输出 (端口) 001: IRQ4 输入 (INTC) 010: A18 输出 (BSC) *1 011: BACK 输出 (BSC) *1 100: RXD0 输入 (SCI) 其他: 禁止设定
9	PB6MD1	0	R/W	
8	PB6MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PB5MD2	0	R/W	PB5 模式位 选择 PB5/CASL/IRQ3/POE3 引脚的功能。 000: PB5 输入 / 输出 (端口) 001: IRQ3 输入 (INTC) 010: POE3 输入 (POE) *2 100: CASL 输出 (BSC) *1 其他: 禁止设定
5	PB5MD1	0	R/W	
4	PB5MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PB4MD2	0	R/W	PB4 模式位 选择 PB4/RASL/IRQ2/POE2 引脚的功能。 000: PB4 输入 / 输出 (端口) 001: IRQ2 输入 (INTC) 010: POE2 输入 (POE) 100: RASL 输出 (BSC) *1 其他: 禁止设定
1	PB4MD1	0	R/W	
0	PB4MD0	0	R/W	

【注】 \*1 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

\*2 如果设定为 POE3 输入，以后就不能更改此设定。

• 端口 B 的控制寄存器 L1 (PBCRL1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	PB2 MD2	PB2 MD1	PB2 MD0	-	PB1 MD2	PB1 MD1	PB1 MD0	-	PB0 MD2	PB0 MD1	PB0 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0*1	0	0	0	0*1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 11	—	全 0	R	保留位 读写的值总是 0。
10	PB2MD2	0	R/W	PB2 模式位 选择 PB2/IRQ0/POE0 引脚的功能。 000: PB2 输入 / 输出 (端口) 001: IRQ0 输入 (INTC) 010: POE0 输入 (POE) 其他: 禁止设定
9	PB2MD1	0	R/W	
8	PB2MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PB1MD2	0	R/W	PB1 模式位 选择 PB1/A17/TIC5W 引脚的功能。 000: PB1 输入 / 输出 (端口) 001: A17 输出 (BSC) *2 011: TIC5W 输入 (MTU2) 其他: 禁止设定
5	PB1MD1	0	R/W	
4	PB1MD0	0*1	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PB0MD2	0	R/W	PB0 模式位 选择 PB0/A16/TIC5WS 引脚的功能。 000: PB0 输入 / 输出 (端口) 001: A16 输出 (BSC) *2 011: TIC5WS 输入 (MTU2S) 其他: 禁止设定
1	PB0MD1	0	R/W	
0	PB0MD0	0*1	R/W	

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

(2) SH7084/85/86

• 端口 B 的控制寄存器 L3 (PBCRL3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	PB9 MD2	PB9 MD1	PB9 MD0	-	PB8 MD2	PB8 MD1	PB8 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 7	—	全 0	R	保留位 读写的值总是 0。
6	PB9MD2	0	R/W	PB9 模式位 选择 PB9/A21/IRQ7/ADTRG/POE8 引脚的功能。 000: PB9 输入 / 输出 (端口) 001: IRQ7 输入 (INTC) 010: A21 输出 (BSC) * 011: ADTRG 输入 (A/D) 110: POE8 输入 (POE) 其他: 禁止设定
5	PB9MD1	0	R/W	
4	PB9MD0	0	R/W	
3	—	0	R	
2	PB8MD2	0	R/W	PB8 模式位 选择 PB8/A20/WAIT/IRQ6/SCK0 引脚的功能。 000: PB8 输入 / 输出 (端口) 001: IRQ6 输入 (INTC) 010: A20 输出 (BSC) * 011: WAIT 输入 (BSC) * 100: SCK0 输入 / 输出 (SCI) 其他: 禁止设定
1	PB8MD1	0	R/W	
0	PB8MD0	0	R/W	

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 B 的控制寄存器 L2 (PBCRL2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PB7 MD2	PB7 MD1	PB7 MD0	-	PB6 MD2	PB6 MD1	PB6 MD0	-	PB5 MD2	PB5 MD1	PB5 MD0	-	PB4 MD2	PB4 MD1	PB4 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PB7MD2	0	R/W	PB7 模式位 选择 PB7/A19/BREQ/IRQ5/TXD0 引脚的功能。 000: PB7 输入 / 输出 (端口) 001: IRQ5 输入 (INTC) 010: A19 输出 (BSC) *2 011: BREQ 输入 (BSC) *2 100: TXD0 输出 (SCI) 其他: 禁止设定
13	PB7MD1	0	R/W	
12	PB7MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PB6MD2	0	R/W	PB6 模式位 选择 PB6/A18/BACK/IRQ4/RXD0 引脚的功能。 000: PB6 输入 / 输出 (端口) 001: IRQ4 输入 (INTC) 010: A18 输出 (BSC) *2 011: BACK 输出 (BSC) *2 100: RXD0 输入 (SCI) 其他: 禁止设定
9	PB6MD1	0	R/W	
8	PB6MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PB5MD2	0	R/W	PB5 模式位 选择 PB5/CASL/IRQ3/POE3 引脚的功能。 000: PB5 输入 / 输出 (端口) 001: IRQ3 输入 (INTC) 010: POE3 输入 (POE) *1 100: CASL 输出 (BSC) *2 其他: 禁止设定
5	PB5MD1	0	R/W	
4	PB5MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PB4MD2	0	R/W	PB4 模式位 选择 PB4/RASL/IRQ2/POE2 引脚的功能。 000: PB4 输入 / 输出 (端口) 001: IRQ2 输入 (INTC) 010: POE2 输入 (POE) 100: RASL 输出 (BSC) *2 其他: 禁止设定
1	PB4MD1	0	R/W	
0	PB4MD0	0	R/W	

【注】 \*1 如果设定为  $\overline{POE3}$  输入，以后就不能更改此设定。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。



• 端口 B 的控制寄存器 L1 (PBCRL1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PB3 MD2	PB3 MD1	PB3 MD0	-	PB2 MD2	PB2 MD1	PB2 MD0	-	PB1 MD2	PB1 MD1	PB1 MD0	-	PB0 MD2	PB0 MD1	PB0 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0*1	0	0	0	0*1
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PB3MD2	0	R/W	PB3 模式位 选择 PB3/IRQ1/POE1/SDA 引脚的功能。 000: PB3 输入 / 输出 (端口) 001: IRQ1 输入 (INTC) 010: POE1 输入 (POE) 100: SDA 输入 / 输出 (I <sup>2</sup> C2) 其他: 禁止设定
13	PB3MD1	0	R/W	
12	PB3MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PB2MD2	0	R/W	PB2 模式位 选择 PB2/IRQ0/POE0/SCL 引脚的功能。 000: PB2 输入 / 输出 (端口) 001: IRQ0 输入 (INTC) 010: POE0 输入 (POE) 100: SCL 输入 / 输出 (I <sup>2</sup> C2) 其他: 禁止设定
9	PB2MD1	0	R/W	
8	PB2MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PB1MD2	0	R/W	PB1 模式位 选择 PB1/A17/TIC5W 引脚的功能。 000: PB1 输入 / 输出 (端口) 001: A17 输出 (BSC) *2 011: TIC5W 输入 (MTU2) 其他: 禁止设定
5	PB1MD1	0	R/W	
4	PB1MD0	0*1	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PB0MD2	0	R/W	PB0 模式位 选择 PB0/A16/TIC5WS 引脚的功能。 000: PB0 输入 / 输出 (端口) 001: A16 输出 (BSC) *2 011: TIC5WS 输入 (MTU2S) 其他: 禁止设定
1	PB0MD1	0	R/W	
0	PB0MD0	0*1	R/W	

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

### 21.1.5 端口 C 的 IO 寄存器 L、H (PCIORL、PCIORH)

PCIORL 和 PCIORH 是 16 位可读写寄存器，选择端口 C 的引脚输入 / 输出方向。PC25IOR ~ PC18IOR 和 PC15IOR ~ PC0IOR 位分别对应 PC25 ~ PC18 和 PC15 ~ PC0 引脚（省略引脚名中端口以外的多路复用引脚名）。PCIORL 和 PCIORH 在端口 C 的引脚功能为通用输入 / 输出（PC15 ~ PC0 和 PC25 ~ PC18）时有效，否则无效。

如果将 PCIORL 和 PCIORH 的位设定为 1，对应的引脚就为输出引脚；如果置 0 就为输入引脚。但是在 SH7083/84/85 中，PCIORH 的 bit9 ~ 2 无效。

PCIORH 的 bit15 ~ 10、1、0 为保留位，读写的值总是 0。

PCIORL 和 PCIORH 的初始值都是 H'0000。

#### (1) 端口 C 的 IO 寄存器 H (PCIORH)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PC25 IOR	PC24 IOR	PC23 IOR	PC22 IOR	PC21 IOR	PC20 IOR	PC19 IOR	PC18 IOR	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

#### (2) 端口 C 的 IO 寄存器 L (PCIORL)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC15 IOR	PC14 IOR	PC13 IOR	PC12 IOR	PC11 IOR	PC10 IOR	PC9 IOR	PC8 IOR	PC7 IOR	PC6 IOR	PC5 IOR	PC4 IOR	PC3 IOR	PC2 IOR	PC1 IOR	PC0 IOR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

### 21.1.6 端口 C 的控制寄存器 L1 ~ L4、H1 ~ H3 (PCCRL1 ~ PCCRL4、PCCRH1 ~ PCCRH3)

PCCRL1 ~ PCCRL4 和 PCCRH1 ~ PCCRH3 是 16 位可读写的寄存器，选择端口 C 的多路复用引脚功能。

#### (1) SH7083/84/85

- 端口 C 的控制寄存器 H3 ~ H1 (PCCRH3 ~ PCCRH1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 0	—	全 0	R	保留位 读写的值总是 0。

• 端口 C 的控制寄存器 L4 (PCCR4)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PC15 MD0	-	-	-	PC14 MD0	-	-	-	PC13 MD0	-	-	-	PC12 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

位	位名	初始值	R/W	说 明
15 ~ 13	—	全 0	R	保留位 读写的值总是 0。
12	PC15MD0	0*1	R/W	PC15 模式位 选择 PC15/A15 引脚的功能。 0: PC15 输入 / 输出 (端口) 1: A15 输出 (BSC) *2
11 ~ 9	—	全 0	R	保留位 读写的值总是 0。
8	PC14MD0	0*1	R/W	PC14 模式位 选择 PC14/A14 引脚的功能。 0: PC14 输入 / 输出 (端口) 1: A14 输出 (BSC) *2
7 ~ 5	—	全 0	R	保留位 读写的值总是 0。
4	PC13MD0	0*1	R/W	PC13 模式位 选择 PC13/A13 引脚的功能。 0: PC13 输入 / 输出 (端口) 1: A13 输出 (BSC) *2
3 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	PC12MD0	0*1	R/W	PC12 模式位 选择 PC12/A12 引脚的功能。 0: PC12 输入 / 输出 (端口) 1: A12 输出 (BSC) *2

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 C 的控制寄存器 L3 (PCCL3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PC11 MD0	-	-	-	PC10 MD0	-	-	-	PC9 MD0	-	-	-	PC8 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

位	位名	初始值	R/W	说 明
15 ~ 13	—	全 0	R	保留位 读写的值总是 0。
12	PC11MD0	0*1	R/W	PC11 模式位 选择 PC11/A11 引脚的功能。 0: PC11 输入 / 输出 (端口) 1: A11 输出 (BSC) *2
11 ~ 9	—	全 0	R	保留位 读写的值总是 0。
8	PC10MD0	0*1	R/W	PC10 模式位 选择 PC10/A10 引脚的功能。 0: PC10 输入 / 输出 (端口) 1: A10 输出 (BSC) *2
7 ~ 5	—	全 0	R	保留位 读写的值总是 0。
4	PC9MD0	0*1	R/W	PC9 模式位 选择 PC9/A9 引脚的功能。 0: PC9 输入 / 输出 (端口) 1: A9 输出 (BSC) *2
3 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	PC8MD0	0*1	R/W	PC8 模式位 选择 PC8/A8 引脚的功能。 0: PC8 输入 / 输出 (端口) 1: A8 输出 (BSC) *2

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 C 的控制寄存器 L2 (PCCR2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PC7 MD0	-	-	-	PC6 MD0	-	-	-	PC5 MD0	-	-	-	PC4 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

位	位名	初始值	R/W	说 明
15 ~ 13	—	全 0	R	保留位 读写的值总是 0。
12	PC7MD0	0*1	R/W	PC7 模式位 选择 PC7/A7 引脚的功能。 0: PC7 输入 / 输出 (端口) 1: A7 输出 (BSC) *2
11 ~ 9	—	全 0	R	保留位 读写的值总是 0。
8	PC6MD0	0*1	R/W	PC6 模式位 选择 PC6/A6 引脚的功能。 0: PC6 输入 / 输出 (端口) 1: A6 输出 (BSC) *2
7 ~ 5	—	全 0	R	保留位 读写的值总是 0。
4	PC5MD0	0*1	R/W	PC5 模式位 选择 PC5/A5 引脚的功能。 0: PC5 输入 / 输出 (端口) 1: A5 输出 (BSC) *2
3 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	PC4MD0	0*1	R/W	PC4 模式位 选择 PC4/A4 引脚的功能。 0: PC4 输入 / 输出 (端口) 1: A4 输出 (BSC) *2

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

## • 端口 C 的控制寄存器 L1 (PCCR1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PC3 MD0	-	-	-	PC2 MD0	-	-	-	PC1 MD0	-	-	-	PC0 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

位	位名	初始值	R/W	说 明
15 ~ 13	—	全 0	R	保留位 读写的值总是 0。
12	PC3MD0	0*1	R/W	PC3 模式位 选择 PC3/A3 引脚的功能。 0: PC3 输入 / 输出 (端口) 1: A3 输出 (BSC) *2
11 ~ 9	—	全 0	R	保留位 读写的值总是 0。
8	PC2MD0	0*1	R/W	PC2 模式位 选择 PC2/A2 引脚的功能。 0: PC2 输入 / 输出 (端口) 1: A2 输出 (BSC) *2
7 ~ 5	—	全 0	R	保留位 读写的值总是 0。
4	PC1MD0	0*1	R/W	PC1 模式位 选择 PC1/A1 引脚的功能。 0: PC1 输入 / 输出 (端口) 1: A1 输出 (BSC) *2
3 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	PC0MD0	0*1	R/W	PC0 模式位 选择 PC0/A0 引脚的功能。 0: PC0 输入 / 输出 (端口) 1: A0 输出 (BSC) *2

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

(2) SH7086

• 端口 C 的控制寄存器 H3 (PCCR3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	PC25 MD0	-	-	-	PC24 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0*1	0	0	0	0*1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

位	位名	初始值	R/W	说 明
15 ~ 5	—	全 0	R	保留位 读写的值总是 0。
4	PC25MD0	0*1	R/W	PC25 模式位 选择 PC25/A25 引脚的功能。 0: PC25 输入 / 输出 (端口) 1: A25 输出 (BSC) *2
3 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	PC24MD0	0*1	R/W	PC4 模式位 选择 PC24/A24 引脚的功能。 0: PC24 输入 / 输出 (端口) 1: A24 输出 (BSC) *2

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 C 的控制寄存器 H2 (PCCR H2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PC23 MD0	-	-	-	PC22 MD0	-	-	-	PC21 MD0	-	-	-	PC20 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

位	位名	初始值	R/W	说 明
15 ~ 13	—	全 0	R	保留位 读写的值总是 0。
12	PC23MD0	0*1	R/W	PC23 模式位 选择 PC23/A23 引脚的功能。 0: PC23 输入 / 输出 (端口) 1: A23 输出 (BSC) *2
11 ~ 9	—	全 0	R	保留位 读写的值总是 0。
8	PC22MD0	0*1	R/W	PC22 模式位 选择 PC22/A22 引脚的功能。 0: PC22 输入 / 输出 (端口) 1: A22 输出 (BSC) *2
7 ~ 5	—	全 0	R	保留位 读写的值总是 0。
4	PC21MD0	0*1	R/W	PC21 模式位 选择 PC21/A21 引脚的功能。 0: PC21 输入 / 输出 (端口) 1: A21 输出 (BSC) *2
3 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	PC20MD	0*1	R/W	PC20 模式位 选择 PC20/A20 引脚的功能。 0: PC20 输入 / 输出 (端口) 1: A20 输出 (BSC) *2

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。



• 端口 C 的控制寄存器 H1 (PCCR H1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PC19 MD0	-	-	-	PC18 MD0	-	-	-	-	-	-	-	-
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 13	—	全 0	R	保留位 读写的值总是 0。
12	PC19MD0	0*1	R/W	PC19 模式位 选择 PC19/A19 引脚的功能。 0: PC19 输入 / 输出 (端口) 1: A19 输出 (BSC) *2
11 ~ 9	—	全 0	R	保留位 读写的值总是 0。
8	PC18MD0	0*1	R/W	PC18 模式位 选择 PC18/A18 引脚的功能。 0: PC18 输入 / 输出 (端口) 1: A18 输出 (BSC) *2
7 ~ 0	—	全 0	R	保留位 读写的值总是 0。

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 C 的控制寄存器 L4 (PCCR4)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PC15 MD0	-	-	-	PC14 MD0	-	-	-	PC13 MD0	-	-	-	PC12 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

位	位名	初始值	R/W	说 明
15 ~ 13	—	全 0	R	保留位 读写的值总是 0。
12	PC15MD0	0*1	R/W	PC15 模式位 选择 PC15/A15 引脚的功能。 0: PC15 输入 / 输出 (端口) 1: A15 输出 (BSC) *2
11 ~ 9	—	全 0	R	保留位 读写的值总是 0。
8	PC14MD0	0*1	R/W	PC14 模式位 选择 PC14/A14 引脚的功能。 0: PC14 输入 / 输出 (端口) 1: A14 输出 (BSC) *2
7 ~ 5	—	全 0	R	保留位 读写的值总是 0。
4	PC13MD0	0*1	R/W	PC13 模式位 选择 PC13/A13 引脚的功能。 0: PC13 输入 / 输出 (端口) 1: A13 输出 (BSC) *2
3 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	PC12MD0	0*1	R/W	PC12 模式位 选择 PC12/A12 引脚的功能。 0: PC12 输入 / 输出 (端口) 1: A12 输出 (BSC) *2

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 C 的控制寄存器 L3 (PCRL3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PC11 MD0	-	-	-	PC10 MD0	-	-	-	PC9 MD0	-	-	-	PC8 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

位	位名	初始值	R/W	说 明
15 ~ 13	—	全 0	R	保留位 读写的值总是 0。
12	PC11MD0	0*1	R/W	PC11 模式位 选择 PC11/A11 引脚的功能。 0: PC11 输入 / 输出 (端口) 1: A11 输出 (BSC) *2
11 ~ 9	—	全 0	R	保留位 读写的值总是 0。
8	PC10MD0	0*1	R/W	PC10 模式位 选择 PC10/A10 引脚的功能。 0: PC10 输入 / 输出 (端口) 1: A10 输出 (BSC) *2
7 ~ 5	—	全 0	R	保留位 读写的值总是 0。
4	PC9MD0	0*1	R/W	PC9 模式位 选择 PC9/A9 引脚的功能。 0: PC9 输入 / 输出 (端口) 1: A9 输出 (BSC) *2
3 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	PC8MD0	0*1	R/W	PC8 模式位 选择 PC8/A8 引脚的功能。 0: PC8 输入 / 输出 (端口) 1: A8 输出 (BSC) *2

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 C 的控制寄存器 L2 (PCCL2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PC7 MD0	-	-	-	PC6 MD0	-	-	-	PC5 MD0	-	-	-	PC4 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

位	位名	初始值	R/W	说 明
15 ~ 13	—	全 0	R	保留位 读写的值总是 0。
12	PC7MD0	0*1	R/W	PC7 模式位 选择 PC7/A7 引脚的功能。 0: PC7 输入 / 输出 (端口) 1: A7 输出 (BSC) *2
11 ~ 9	—	全 0	R	保留位 读写的值总是 0。
8	PC6MD0	0*1	R/W	PC6 模式位 选择 PC6/A6 引脚的功能。 0: PC6 输入 / 输出 (端口) 1: A6 输出 (BSC) *2
7 ~ 5	—	全 0	R	保留位 读写的值总是 0。
4	PC5MD0	0*1	R/W	PC5 模式位 选择 PC5/A5 引脚的功能。 0: PC5 输入 / 输出 (端口) 1: A5 输出 (BSC) *2
3 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	PC4MD0	0*1	R/W	PC4 模式位 选择 PC4/A4 引脚的功能。 0: PC4 输入 / 输出 (端口) 1: A4 输出 (BSC) *2

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 C 的控制寄存器 L1 (PCCR1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PC3 MD0	-	-	-	PC2 MD0	-	-	-	PC1 MD0	-	-	-	PC0 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

位	位名	初始值	R/W	说 明
15 ~ 13	—	全 0	R	保留位 读写的值总是 0。
12	PC3MD0	0*1	R/W	PC3 模式位 选择 PC3/A3 引脚的功能。 0: PC3 输入 / 输出 (端口) 1: A3 输出 (BSC) *2
11 ~ 9	—	全 0	R	保留位 读写的值总是 0。
8	PC2MD0	0*1	R/W	PC2 模式位 选择 PC2/A2 引脚的功能。 0: PC2 输入 / 输出 (端口) 1: A2 输出 (BSC) *2
7 ~ 5	—	全 0	R	保留位 读写的值总是 0。
4	PC1MD0	0*1	R/W	PC1 模式位 选择 PC1/A1 引脚的功能。 0: PC1 输入 / 输出 (端口) 1: A1 输出 (BSC) *2
3 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	PC0MD0	0*1	R/W	PC0 模式位 选择 PC0/A0 引脚的功能。 0: PC0 输入 / 输出 (端口) 1: A0 输出 (BSC) *2

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

### 21.1.7 端口 D 的 IO 寄存器 L、H (PDIORL、PDIORH)

PDIORL 和 PDIORH 是 16 位可读写寄存器，选择端口 D 的引脚输入 / 输出方向。PD31IOR ~ PD0IOR 位分别对应 PD31 ~ PD0 引脚（省略引脚名中端口以外的多路复用引脚名）。PDIORL 和 PDIORH 在端口 D 的引脚功能为通用输入 / 输出（PD15 ~ PD0 和 PD31 ~ PD16）以及 MTU2S 的 TI0C 输入 / 输出时有效，否则无效。

如果将 PDIORL 和 PDIORH 的位设定为 1，对应的引脚就为输出引脚；如果置 0 就为输入引脚。但是在 SH7083/84 中，PDIORH 无效。

PDIORL 和 PDIORH 的初始值都是 H'0000。

#### (1) 端口 D 的 IO 寄存器 H (PDIORH)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD31 IOR	PD30 IOR	PD29 IOR	PD28 IOR	PD27 IOR	PD26 IOR	PD25 IOR	PD24 IOR	PD23 IOR	PD22 IOR	PD21 IOR	PD20 IOR	PD19 IOR	PD18 IOR	PD17 IOR	PD16 IOR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

#### (2) 端口 D 的 IO 寄存器 L (PDIORL)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD15 IOR	PD14 IOR	PD13 IOR	PD12 IOR	PD11 IOR	PD10 IOR	PD9 IOR	PD8 IOR	PD7 IOR	PD6 IOR	PD5 IOR	PD4 IOR	PD3 IOR	PD2 IOR	PD1 IOR	PD0 IOR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

### 21.1.8 端口 D 的控制寄存器 L1 ~ L4、H1 ~ H4 (PDCRL1 ~ PDCRL4、PDCRH1 ~ PDCRH4)

PDCRL1 ~ PDCRL4 和 PDCRH1 ~ PDCRH4 是 16 位可读写寄存器，选择端口 D 的多路复用引脚功能。

#### (1) SH7083/84

- 端口 D 的控制寄存器 H4 ~ H1 (PDCRH4 ~ PDCRH1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 0	—	全 0	R	保留位 读写的值总是 0。

• 端口 D 的控制寄存器 L4 (PDCRL4)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PD15 MD1	PD15 MD0	-	-	PD14 MD1	PD14 MD0	-	-	PD13 MD1	PD13 MD0	-	-	PD12 MD1	PD12 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
15、14	—	全 0	R	保留位 读写的值总是 0。
13 12	PD15MD1 PD15MD0	0 0*1	R/W R/W	PD15 模式位 选择 PD15/D15/TIOC4DS/AUDSYNC 引脚的功能。当使用 E10A 的 AUD 功能时，此引脚被固定为 AUDSYNC 输出。 00: PD15 输入 / 输出 (端口) 01: D15 输入 / 输出 (BSC) *2 11: TIOC4DS 输入 / 输出 (MTU2S) 其他: 禁止设定
11、10	—	全 0	R	保留位 读写的值总是 0。
9 8	PD14MD1 PD14MD0	0 0*1	R/W R/W	PD14 模式位 选择 PD14/D14/TIOC4CS/AUDCK 引脚的功能。当使用 E10A 的 AUD 功能时，此引脚被固定为 AUDCK 输出。 00: PD14 输入 / 输出 (端口) 01: D14 输入 / 输出 (BSC) *2 11: TIOC4CS 输入 / 输出 (MTU2S) 其他: 禁止设定
7、6	—	全 0	R	保留位 读写的值总是 0。
5 4	PD13MD1 PD13MD0	0 0*1	R/W R/W	PD13 模式位 选择 PD13/D13/TIOC4BS 引脚的功能。 00: PD13 输入 / 输出 (端口) 01: D13 输入 / 输出 (BSC) *2 11: TIOC4BS 输入 / 输出 (MTU2S) 其他: 禁止设定
3、2	—	全 0	R	保留位 读写的值总是 0。
1 0	PD12MD1 PD12MD0	0 0*1	R/W R/W	PD12 模式位 选择 PD12/D12/TIOC4AS 引脚的功能。 00: PD12 输入 / 输出 (端口) 01: D12 输入 / 输出 (BSC) *2 11: TIOC4AS 输入 / 输出 (MTU2S) 其他: 禁止设定

【注】 \*1 在内部 ROM 无效的 16 位外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 D 的控制寄存器 L3 (PDCRL3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PD11 MD1	PD11 MD0	-	PD10 MD2	PD10 MD1	PD10 MD0	-	PD9 MD2	PD9 MD1	PD9 MD0	-	PD8 MD2	PD8 MD1	PD8 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15、14	—	全 0	R	保留位 读写的值总是 0。
13 12	PD11MD1 PD11MD0	0 0*1	R/W R/W	PD11 模式位 选择 PD11/D11/TIOC3DS/AUDATA3 引脚的功能。当使用 E10A 的 AUD 功能时，此引脚被固定为 AUDATA3 输出。 00: PD11 输入 / 输出 (端口) 01: D11 输入 / 输出 (BSC) *2 11: TIOC3DS 输入 / 输出 (MTU2S) 其他: 禁止设定
11	—	0	R	保留位 读写的值总是 0。
10 9 8	PD10MD2 PD10MD1 PD10MD0	0 0 0*1	R/W R/W R/W	PD10 模式位 选择 PD10/D10/TIOC3CS/AUDATA2 引脚的功能。当使用 E10A 的 AUD 功能时，此引脚被固定为 AUDATA2 输出。 000: PD10 输入 / 输出 (端口) 001: D10 输入 / 输出 (BSC) *2 011: TIOC3CS 输入 / 输出 (MTU2S) 其他: 禁止设定
7	—	0	R	保留位 读写的值总是 0。
6 5 4	PD9MD2 PD9MD1 PD9MD0	0 0 0*1	R/W R/W R/W	PD9 模式位 选择 PD9/D9/TIOC3BS/AUDATA1 引脚的功能。当使用 E10A 的 AUD 功能时，此引脚被固定为 AUDATA1 输出。 000: PD9 输入 / 输出 (端口) 001: D9 输入 / 输出 (BSC) *2 011: TIOC3BS 输入 / 输出 (MTU2S) 其他: 禁止设定
3	—	0	R	保留位 读写的值总是 0。
2 1 0	PD8MD2 PD8MD1 PD8MD0	0 0 0*1	R/W R/W R/W	PD8 模式位 选择 PD8/D8/TIOC3AS/AUDATA0 引脚的功能。当使用 E10A 的 AUD 功能时，此引脚被固定为 AUDATA0 输出。 000: PD8 输入 / 输出 (端口) 001: D8 输入 / 输出 (BSC) *2 011: TIOC3AS 输入 / 输出 (MTU2S) 其他: 禁止设定

【注】 \*1 在内部 ROM 无效的 16 位外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。



• 端口 D 的控制寄存器 L2 (PDCRL2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PD7 MD2	PD7 MD1	PD7 MD0	-	PD6 MD2	PD6 MD1	PD6 MD0	-	PD5 MD2	PD5 MD1	PD5 MD0	-	PD4 MD2	PD4 MD1	PD4 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PD7MD2	0	R/W	PD7 模式位 选择 PD7/D7/TIC5WS 引脚的功能。 000: PD7 输入 / 输出 (端口) 001: D7 输入 / 输出 (BSC) *2 010: TIC5WS 输入 (MTU2S) 其他: 禁止设定
13	PD7MD1	0	R/W	
12	PD7MD0	0*1	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PD6MD2	0	R/W	PD6 模式位 选择 PD6/D6/TIC5VS 引脚的功能。 000: PD6 输入 / 输出 (端口) 001: D6 输入 / 输出 (BSC) *2 010: TIC5VS 输入 (MTU2S) 其他: 禁止设定
9	PD6MD1	0	R/W	
8	PD6MD0	0*1	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PD5MD2	0	R/W	PD5 模式位 选择 PD5/D5/TIC5US 引脚的功能。 000: PD5 输入 / 输出 (端口) 001: D5 输入 / 输出 (BSC) *2 010: TIC5US 输入 (MTU2S) 其他: 禁止设定
5	PD5MD1	0	R/W	
4	PD5MD0	0*1	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PD4MD2	0	R/W	PD4 模式位 选择 PD4/D4/TIC5W 引脚的功能。 000: PD4 输入 / 输出 (端口) 001: D4 输入 / 输出 (BSC) *2 010: TIC5W 输入 (MTU2) 其他: 禁止设定
1	PD4MD1	0	R/W	
0	PD4MD0	0*1	R/W	

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 D 的控制寄存器 L1 (PDCRL1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PD3 MD2	PD3 MD1	PD3 MD0	-	PD2 MD2	PD2 MD1	PD2 MD0	-	PD1 MD2	PD1 MD1	PD1 MD0	-	PD0 MD2	PD0 MD1	PD0 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PD3MD2	0	R/W	PD3 模式位 选择 PD3/D3/TIC5V 引脚的功能。 000: PD3 输入 / 输出 (端口) 001: D3 输入 / 输出 (BSC) *2 010: TIC5V 输入 (MTU2) 其他: 禁止设定
13	PD3MD1	0	R/W	
12	PD3MD0	0*1	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PD2MD2	0	R/W	PD2 模式位 选择 PD2/D2/TIC5U 引脚的功能。 000: PD2 输入 / 输出 (端口) 001: D2 输入 / 输出 (BSC) *2 010: TIC5U 输入 (MTU2) 其他: 禁止设定
9	PD2MD1	0	R/W	
8	PD2MD0	0*1	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PD1MD2	0	R/W	PD1 模式位 选择 PD1/D1 引脚的功能。 000: PD1 输入 / 输出 (端口) 001: D1 输入 / 输出 (BSC) *2 其他: 禁止设定
5	PD1MD1	0	R/W	
4	PD1MD0	0*1	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PD0MD2	0	R/W	PD0 模式位 选择 PD0/D0 引脚的功能。 000: PD0 输入 / 输出 (端口) 001: D0 输入 / 输出 (BSC) *2 其他: 禁止设定
1	PD0MD1	0	R/W	
0	PD0MD0	0*1	R/W	

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

(2) SH7085/86

• 端口 D 的控制寄存器 H4 (PDCRH4)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PD31 MD1	PD31 MD0	-	-	PD30 MD1	PD30 MD0	-	-	PD29 MD1	PD29 MD0	-	-	PD28 MD1	PD28 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
15、14	—	全 0	R	保留位 读写的值总是 0。
13 12	PD31MD1 PD31MD0	0 0*1	R/W R/W	PD31 模式位 选择 PD31/D31/TIOC3AS/ADTRG 引脚的功能。 00: PD31 输入 / 输出 (端口) 01: D31 输入 / 输出 (BSC) *2 10: ADTRG 输入 (A/D) 11: TIOC3AS 输入 / 输出 (MTU2S)
11、10	—	全 0	R	保留位 读写的值总是 0。
9 8	PD30MD1 PD30MD0	0 0*1	R/W R/W	PD30 模式位 选择 PD30/D30/TIOC3CS/IRQOUT 引脚的功能。 00: PD30 输入 / 输出 (端口) 01: D30 输入 / 输出 (BSC) *2 10: IRQOUT 输出 (INTC) 11: TIOC3CS 输入 / 输出 (MTU2S)
7、6	—	全 0	R	保留位 读写的值总是 0。
5 4	PD29MD1 PD29MD0	0 0*1	R/W R/W	PD29 模式位 选择 PD29/D29/CS3/TIOC3BS 引脚的功能。 00: PD29 输入 / 输出 (端口) 01: D29 输入 / 输出 (BSC) *2 10: CS3 输出 (BSC) *2 11: TIOC3BS 输入 / 输出 (MTU2S)
3、2	—	全 0	R	保留位 读写的值总是 0。
1 0	PD28MD1 PD28MD0	0 0*1	R/W R/W	PD28 模式位 选择 PD28/D28/CS2/TIOC3DS 引脚的功能。 00: PD28 输入 / 输出 (端口) 01: D28 输入 / 输出 (BSC) *2 10: CS2 输出 (BSC) *2 11: TIOC3DS 输入 / 输出 (MTU2S)

【注】 \*1 在内部 ROM 无效的 32 位外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 D 的控制寄存器 H3 (PDCRH3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PD27 MD1	PD27 MD0	-	-	PD26 MD1	PD26 MD0	-	-	PD25 MD1	PD25 MD0	-	-	PD24 MD1	PD24 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
15、14	—	全 0	R	保留位 读写的值总是 0。
13 12	PD27MD1 PD27MD0	0 0*1	R/W R/W	PD27 模式位 选择 PD27/D27/DACK1/TIOC4AS 引脚的功能。 00: PD27 输入 / 输出 (端口) 01: D27 输入 / 输出 (BSC) *2 10: DACK1 输出 (DMAC) *2 11: TIOC4AS 输入 / 输出 (MTU2S)
11、10	—	全 0	R	保留位 读写的值总是 0。
9 8	PD26MD1 PD26MD0	0 0*1	R/W R/W	PD26 模式位 选择 PD26/D26/DACK0/TIOC4BS 引脚的功能。 00: PD26 输入 / 输出 (端口) 01: D26 输入 / 输出 (BSC) *2 10: DACK0 输出 (DMAC) *2 11: TIOC4BS 输入 / 输出 (MTU2S)
7、6	—	全 0	R	保留位 读写的值总是 0。
5 4	PD25MD1 PD25MD0	0 0*1	R/W R/W	PD25 模式位 选择 PD25/D25/DREQ1/TIOC4CS 引脚的功能。 00: PD25 输入 / 输出 (端口) 01: D25 输入 / 输出 (BSC) *2 10: DREQ1 输入 (DMAC) 11: TIOC4CS 输入 / 输出 (MTU2S)
3、2	—	全 0	R	保留位 读写的值总是 0。
1 0	PD24MD1 PD24MD0	0 0*1	R/W R/W	PD24 模式位 选择 PD24/D24/DREQ0/TIOC4DS 引脚的功能。 00: PD24 输入 / 输出 (端口) 01: D24 输入 / 输出 (BSC) 10: DREQ0 输出 (DMAC) 11: TIOC4DS 输入 / 输出 (MTU2S)

【注】 \*1 在内部 ROM 无效的 32 位外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 D 的控制寄存器 H2 (PDCRH2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PD23 MD1	PD23 MD0	-	PD22 MD2	PD22 MD1	PD22 MD0	-	PD21 MD2	PD21 MD1	PD21 MD0	-	PD20 MD2	PD20 MD1	PD20 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15、14	—	全 0	R	保留位 读写的值总是 0。
13 12	PD23MD1 PD23MD0	0 0*1	R/W R/W	PD23 模式位 选择 PD23/D23/IRQ7/AUDSYNC 引脚的功能。当使用 E10A 的 AUD 功能时，此引脚被固定为 AUDSYNC 输出。 00: PD23 输入 / 输出 (端口) 01: D23 输入 / 输出 (BSC) *2 10: IRQ7 输入 (INTC) 其他: 禁止设定
11	—	0	R	保留位 读写的值总是 0。
10 9 8	PD22MD2 PD22MD1 PD22MD0	0 0 0*1	R/W R/W R/W	PD22 模式位 选择 PD22/D22/IRQ6/TIC5US/AUDCK 引脚的功能。当使用 E10A 的 AUD 功能时，此引脚被固定为 AUDCK 输出。 000: PD22 输入 / 输出 (端口) 001: D22 输入 / 输出 (BSC) *2 010: IRQ6 输入 (INTC) 100: TIC5US 输入 (MTU2S) 其他: 禁止设定
7	—	0	R	保留位 读写的值总是 0。
6 5 4	PD21MD2 PD21MD1 PD21MD0	0 0 0*1	R/W R/W R/W	PD21 模式位 选择 PD21/D21/IRQ5/TIC5VS 引脚的功能。 000: PD21 输入 / 输出 (端口) 001: D21 输入 / 输出 (BSC) *2 010: IRQ5 输入 (INTC) 100: TIC5VS 输入 (MTU2S) 其他: 禁止设定
3	—	0	R	保留位 读写的值总是 0。
2 1 0	PD20MD2 PD20MD1 PD20MD0	0 0 0*1	R/W R/W R/W	PD20 模式位 选择 PD20/D20/IRQ4/TIC5WS 引脚的功能。 000: PD20 输入 / 输出 (端口) 001: D20 输入 / 输出 (BSC) *2 010: IRQ4 输入 (INTC) 100: TIC5WS 输入 (MTU2S) 其他: 禁止设定

【注】 \*1 在内部 ROM 无效的 32 位外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 D 的控制寄存器 H1 (PDCRH1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PD19 MD2	PD19 MD1	PD19 MD0	-	PD18 MD2	PD18 MD1	PD18 MD0	-	PD17 MD2	PD17 MD1	PD17 MD0	-	PD16 MD2	PD16 MD1	PD16 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PD19MD2	0	R/W	PD19 模式位 选择 PD19/D19/IRQ3/POE7/AUDATA3 引脚的功能。当使用 E10A 的 AUD 功能时，此引脚被固定为 AUDATA3 输出。 000: PD19 输入 / 输出 (端口) 001: D19 输入 / 输出 (BSC) *2 010: IRQ3 输入 (INTC) 100: POE7 输入 (POE) 其他: 禁止设定
13	PD19MD1	0	R/W	
12	PD19MD0	0*1	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PD18MD2	0	R/W	PD18 模式位 选择 PD18/D18/IRQ2/POE6/AUDATA2 引脚的功能。当使用 E10A 的 AUD 功能时，此引脚被固定为 AUDATA2 输出。 000: PD18 输入 / 输出 (端口) 001: D18 输入 / 输出 (BSC) *2 010: IRQ2 输入 (INTC) 100: POE6 输入 (POE) 其他: 禁止设定
9	PD18MD1	0	R/W	
8	PD18MD0	0*1	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PD17MD2	0	R/W	PD17 模式位 选择 PD17/D17/IRQ1/POE5/AUDATA1 引脚的功能。当使用 E10A 的 AUD 功能时，此引脚被固定为 AUDATA1 输出。 000: PD17 输入 / 输出 (端口) 001: D17 输入 / 输出 (BSC) *2 010: IRQ1 输入 (INTC) 100: POE5 输入 (POE) 其他: 禁止设定
5	PD17MD1	0	R/W	
4	PD17MD0	0*1	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PD16MD2	0	R/W	PD16 模式位 选择 PD16/D16/IRQ0/POE4/AUDATA0 引脚的功能。当使用 E10A 的 AUD 功能时，此引脚被固定为 AUDATA0 输出。 000: PD16 输入 / 输出 (端口) 001: D16 输入 / 输出 (BSC) *2 010: IRQ0 输入 (INTC) 100: POE4 输入 (POE) 其他: 禁止设定
1	PD16MD1	0	R/W	
0	PD16MD0	0*1	R/W	

【注】 \*1 在内部 ROM 无效的 32 位外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 D 的控制寄存器 L4 (PDCRL4)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PD15 MD1	PD15 MD0	-	-	PD14 MD1	PD14 MD0	-	-	PD13 MD1	PD13 MD0	-	-	PD12 MD1	PD12 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
15、14	—	全 0	R	保留位 读写的值总是 0。
13 12	PD15MD1 PD15MD0	0 0*1	R/W R/W	PD15 模式位 选择 PD15/D15/TIOC4DS 引脚的功能。 00: PD15 输入 / 输出 (端口) 01: D15 输入 / 输出 (BSC) *2 11: TIOC4DS 输入 / 输出 (MTU2S) 其他: 禁止设定
11、10	—	全 0	R	保留位 读写的值总是 0。
9 8	PD14MD1 PD14MD0	0 0*1	R/W R/W	PD14 模式位 选择 PD14/D14/TIOC4CS 引脚的功能。 00: PD14 输入 / 输出 (端口) 01: D14 输入 / 输出 (BSC) *2 11: TIOC4CS 输入 / 输出 (MTU2S) 其他: 禁止设定
7、6	—	全 0	R	保留位 读写的值总是 0。
5 4	PD13MD1 PD13MD0	0 0*1	R/W R/W	PD13 模式位 选择 PD13/D13/TIOC4BS 引脚的功能。 00: PD13 输入 / 输出 (端口) 01: D13 输入 / 输出 (BSC) *2 11: TIOC4BS 输入 / 输出 (MTU2S) 其他: 禁止设定
3、2	—	全 0	R	保留位 读写的值总是 0。
1 0	PD12MD1 PD12MD0	0 0*1	R/W R/W	PD12 模式位 选择 PD12/D12/TIOC4AS 引脚的功能。 00: PD12 输入 / 输出 (端口) 01: D12 输入 / 输出 (BSC) *2 11: TIOC4AS 输入 / 输出 (MTU2S) 其他: 禁止设定

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 D 的控制寄存器 L3 (PDCRL3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PD11 MD1	PD11 MD0	-	PD10 MD2	PD10 MD1	PD10 MD0	-	PD9 MD2	PD9 MD1	PD9 MD0	-	PD8 MD2	PD8 MD1	PD8 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15、14	—	全 0	R	保留位 读写的值总是 0。
13 12	PD11MD1 PD11MD0	0 0*1	R/W R/W	PD11 模式位 选择 PD11/D11/TIOC3DS 引脚的功能。 00: PD11 输入 / 输出 (端口) 01: D11 输入 / 输出 (BSC) *2 11: TIOC3DS 输入 / 输出 (MTU2S) 其他: 禁止设定
11	—	0	R	保留位 读写的值总是 0。
10 9 8	PD10MD2 PD10MD1 PD10MD0	0 0 0*1	R/W R/W R/W	PD10 模式位 选择 PD10/D10/TIOC3CS 引脚的功能。 000: PD10 输入 / 输出 (端口) 001: D10 输入 / 输出 (BSC) *2 011: TIOC3CS 输入 / 输出 (MTU2S) 其他: 禁止设定
7	—	0	R	保留位 读写的值总是 0。
6 5 4	PD9MD2 PD9MD1 PD9MD0	0 0 0*1	R/W R/W R/W	PD9 模式位 选择 PD9/D9/TIOC3BS 引脚的功能。 000: PD9 输入 / 输出 (端口) 001: D9 输入 / 输出 (BSC) *2 011: TIOC3BS 输入 / 输出 (MTU2S) 其他: 禁止设定
3	—	0	R	保留位 读写的值总是 0。
2 1 0	PD8MD2 PD8MD1 PD8MD0	0 0 0*1	R/W R/W R/W	PD8 模式位 选择 PD8/D8/TIOC3AS 引脚的功能。 000: PD8 输入 / 输出 (端口) 001: D8 输入 / 输出 (BSC) *2 011: TIOC3AS 输入 / 输出 (MTU2S) 其他: 禁止设定

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。



• 端口 D 的控制寄存器 L2 (PDCRL2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PD7 MD2	PD7 MD1	PD7 MD0	-	PD6 MD2	PD6 MD1	PD6 MD0	-	PD5 MD2	PD5 MD1	PD5 MD0	-	PD4 MD2	PD4 MD1	PD4 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PD7MD2	0	R/W	PD7 模式位 选择 PD7/D7/TIC5WS 引脚的功能。 000: PD7 输入 / 输出 (端口) 001: D7 输入 / 输出 (BSC) *2 010: TIC5WS 输入 (MTU2S) 其他: 禁止设定
13	PD7MD1	0	R/W	
12	PD7MD0	0*1	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PD6MD2	0	R/W	PD6 模式位 选择 PD6/D6/TIC5VS 引脚的功能。 000: PD6 输入 / 输出 (端口) 001: D6 输入 / 输出 (BSC) *2 010: TIC5VS 输入 (MTU2S) 其他: 禁止设定
9	PD6MD1	0	R/W	
8	PD6MD0	0*1	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PD5MD2	0	R/W	PD5 模式位 选择 PD5/D5/TIC5US 引脚的功能。 000: PD5 输入 / 输出 (端口) 001: D5 输入 / 输出 (BSC) *2 010: TIC5US 输入 (MTU2S) 其他: 禁止设定
5	PD5MD1	0	R/W	
4	PD5MD0	0*1	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PD4MD2	0	R/W	PD4 模式位 选择 PD4/D4/TIC5W 引脚的功能。 000: PD4 输入 / 输出 (端口) 001: D4 输入 / 输出 (BSC) *2 010: TIC5W 输入 (MTU2) 其他: 禁止设定
1	PD4MD1	0	R/W	
0	PD4MD0	0*1	R/W	

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 D 的控制寄存器 L1 (PDCRL1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PD3 MD2	PD3 MD1	PD3 MD0	-	PD2 MD2	PD2 MD1	PD2 MD0	-	PD1 MD2	PD1 MD1	PD1 MD0	-	PD0 MD2	PD0 MD1	PD0 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PD3MD2	0	R/W	PD3 模式位 选择 PD3/D3/TIC5V 引脚的功能。 000: PD3 输入 / 输出 (端口) 001: D3 输入 / 输出 (BSC) *2 010: TIC5V 输入 (MTU2) 其他: 禁止设定
13	PD3MD1	0	R/W	
12	PD3MD0	0*1	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PD2MD2	0	R/W	PD2 模式位 选择 PD2/D2/TIC5U 引脚的功能。 000: PD2 输入 / 输出 (端口) 001: D2 输入 / 输出 (BSC) *2 010: TIC5U 输入 (MTU2) 其他: 禁止设定
9	PD2MD1	0	R/W	
8	PD2MD0	0*1	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PD1MD2	0	R/W	PD1 模式位 选择 PD1/D1 引脚的功能。 000: PD1 输入 / 输出 (端口) 001: D1 输入 / 输出 (BSC) *2 其他: 禁止设定
5	PD1MD1	0	R/W	
4	PD1MD0	0*1	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PD0MD2	0	R/W	PD0 模式位 选择 PD0/D0 引脚的功能。 000: PD0 输入 / 输出 (端口) 001: D0 输入 / 输出 (BSC) *2 其他: 禁止设定
1	PD0MD1	0	R/W	
0	PD0MD0	0*1	R/W	

【注】 \*1 在内部 ROM 无效的外部扩展模式中，初始值是 1。

\*2 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

### 21.1.9 端口 E 的 IO 寄存器 L、H (PEIORL、PEIORH)

PEIORL 和 PEIORH 是 16 位可读写寄存器，选择端口 E 的引脚输入 / 输出方向。PE21IOR ~ PE0IOR 位分别对应 PE21 引脚 ~ PE0 引脚（省略引脚名中端口以外的多路复用引脚名）。PEIORL 在端口 E 的引脚功能为通用输入 / 输出（PE15 ~ PE0）以及 MTU2 的 TIOC 输入 / 输出时有效，否则无效；PEIORH 在端口 E 的引脚功能为通用输入 / 输出（PE21 ~ PE16）以及 MTU2S 的 TIOC 输入 / 输出时有效，否则无效。

如果将 PEIORL 和 PEIORH 的位设定为 1，对应的引脚就为输出引脚；如果置 0 就为输入引脚。

但是在 SH7083 中，PEIORH 和 PEIORL 的 bit11、bit9、bit5 无效；在 SH7084/85 中，PEIORH 无效。

PEIORH 的 bit15 ~ 6 为保留位，读写的值总是 0。

PEIORL 和 PEIORH 的初始值都是 H'0000。

#### (1) 端口 E 的 IO 寄存器 H (PEIORH)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PE21 IOR	PE20 IOR	PE19 IOR	PE18 IOR	PE17 IOR	PE16 IOR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

#### (2) 端口 E 的 IO 寄存器 L (PEIORL)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15 IOR	PE14 IOR	PE13 IOR	PE12 IOR	PE11 IOR	PE10 IOR	PE9 IOR	PE8 IOR	PE7 IOR	PE6 IOR	PE5 IOR	PE4 IOR	PE3 IOR	PE2 IOR	PE1 IOR	PE0 IOR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

### 21.1.10 端口 E 的控制寄存器 L1 ~ L4、H1、H2 (PECRL1 ~ PECRL4、PECRH1、PECRH2)

PECRL1 ~ PECRL4、PECRH1 和 PECRH2 是 16 位可读写寄存器，选择端口 E 的多路复用引脚功能。

#### (1) SH7083

- 端口 E 的控制寄存器 H2、H1 (PECRH2、PECRH1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 0	—	全 0	R	保留位 读写的值总是 0。

• 端口 E 的控制寄存器 L4 (PECRL4)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE15 MD2	PE15 MD1	PE15 MD0	-	PE14 MD2	PE14 MD1	PE14 MD0	-	-	PE13 MD1	PE13 MD0	-	PE12 MD2	PE12 MD1	PE12 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PE15MD2	0	R/W	PE15 模式位 选择 PE15/CKE/DACK1/TIOC4D/IRQOUT 引脚的功能。 000: PE15 输入 / 输出 (端口) 001: TIOC4D 输入 / 输出 (MTU2) 010: DACK1 输出 (DMAC) * 011: IRQOUT 输出 (INTC) 101: CKE 输出 (BSC) * 其他: 禁止设定
13	PE15MD1	0	R/W	
12	PE15MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PE14MD2	0	R/W	PE14 模式位 选择 PE14/DACK0/TIOC4C 引脚的功能。 000: PE14 输入 / 输出 (端口) 001: TIOC4C 输入 / 输出 (MTU2) 010: DACK0 输出 (DMAC) * 其他: 禁止设定
9	PE14MD1	0	R/W	
8	PE14MD0	0	R/W	
7、6	—	全 0	R	保留位 读写的值总是 0。
5	PE13MD1	0	R/W	PE13 模式位 选择 PE13/TIOC4B/MRES/ASEBRKAK/ASEBRK 引脚的功能。当使用 E10A (ASEMD0=Low 电平) 时, 此引脚被固定为 ASEBRKAK 输出或者 ASEBRK 输入。 00: PE13 输入 / 输出 (端口) 01: TIOC4B 输入 / 输出 (MTU2) 10: MRES 输入 (INTC) 其他: 禁止设定
4	PE13MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PE12MD2	0	R/W	PE12 模式位 选择 PE12/TIOC4A/TXD3/SCS 引脚的功能。 000: PE12 输入 / 输出 (端口) 001: TIOC4A 输入 / 输出 (MTU2) 011: TXD3 输出 (SCIF) 101: SCS 输入 / 输出 (SSU) 其他: 禁止设定
1	PE12MD1	0	R/W	
0	PE12MD0	0	R/W	

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 E 的控制寄存器 L3 (PECRL3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	PE10 MD2	PE10 MD1	PE10 MD0	-	-	-	-	-	PE8 MD2	PE8 MD1	PE8 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 11	—	全 0	R	保留位 读写的值总是 0。
10	PE10MD2	0	R/W	PE10 模式位 选择 PE10/TIOC3C/TXD2/SSO 引脚的功能。 000: PE10 输入 / 输出 (端口) 001: TIOC3C 输入 / 输出 (MTU2) 010: TXD2 输出 (SCI) 101: SSO 输入 / 输出 (SSU) 其他: 禁止设定
9	PE10MD1	0	R/W	
8	PE10MD0	0	R/W	
7 ~ 3	—	全 0	R	保留位 读写的值总是 0。
2	PE8MD2	0	R/W	PE8 模式位 选择 PE8/TIOC3A/SCK2/SSCK 引脚的功能。 000: PE8 输入 / 输出 (端口) 001: TIOC3A 输入 / 输出 (MTU2) 010: SCK2 输入 / 输出 (SCI) 101: SSCK 输入 / 输出 (SSU) 其他: 禁止设定
1	PE8MD1	0	R/W	
0	PE8MD0	0	R/W	

• 端口 E 的控制寄存器 L2 (PECRL2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE7 MD2	PE7 MD1	PE7 MD0	-	PE6 MD2	PE6 MD1	PE6 MD0	-	-	-	-	-	PE4 MD2	PE4 MD1	PE4 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PE7MD2	0	R/W	PE7 模式位 选择 PE7/ $\overline{\text{BS}}$ /TIOC2B/ $\overline{\text{UBCTRG}}$ /RXD2/SSI 引脚的功能。 000: PE7 输入 / 输出 (端口) 001: TIOC2B 输入 / 输出 (MTU2) 010: RXD2 输入 (SCI) 011: $\overline{\text{BS}}$ 输出 (BSC) * 101: SSI 输入 / 输出 (SSU) 111: $\overline{\text{UBCTRG}}$ 输出 (UBC) 其他: 禁止设定
13	PE7MD1	0	R/W	
12	PE7MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PE6MD2	0	R/W	PE6 模式位 选择 PE6/ $\overline{\text{CS7}}$ /TIOC2A/SCK3 引脚的功能。 000: PE6 输入 / 输出 (端口) 001: TIOC2A 输入 / 输出 (MTU2) 010: SCK3 输入 / 输出 (SCIF) 101: $\overline{\text{CS7}}$ 输出 (BSC) * 其他: 禁止设定
9	PE6MD1	0	R/W	
8	PE6MD0	0	R/W	
7 ~ 3	—	全 0	R	保留位 读写的值总是 0。
2	PE4MD2	0	R/W	PE4 模式位 选择 PE4/TIOC1A/RXD3/TCK 引脚的功能。当使用 E10A (ASEMD0=Low 电平) 时, 此引脚被固定为 TCK 输入。 000: PE4 输入 / 输出 (端口) 001: TIOC1A 输入 / 输出 (MTU2) 010: RXD3 输入 (SCIF) 其他: 禁止设定
1	PE4MD1	0	R/W	
0	PE4MD0	0	R/W	

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 E 的控制寄存器 L1 (PECRL1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE3 MD2	PE3 MD1	PE3 MD0	-	PE2 MD2	PE2 MD1	PE2 MD0	-	PE1 MD2	PE1 MD1	PE1 MD0	-	-	PE0 MD1	PE0 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PE3MD2	0	R/W	PE3 模式位 选择 PE3/TEND1/TIOC0D/TDO 引脚的功能。当使用 E10A (ASEMD0=Low 电平) 时, 此引脚被固定为 TDO 输出。 000: PE3 输入 / 输出 (端口) 001: TIOC0D 输入 / 输出 (MTU2) 010: TEND1 输出 (DMAC) * 其他: 禁止设定
13	PE3MD1	0	R/W	
12	PE3MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PE2MD2	0	R/W	PE2 模式位 选择 PE2/DREQ1/TIOC0C/TDI 引脚的功能。当使用 E10A (ASEMD0=Low 电平) 时, 此引脚被固定为 TDI 输入。 000: PE2 输入 / 输出 (端口) 001: TIOC0C 输入 / 输出 (MTU2) 010: DREQ1 输入 (DMAC) 其他: 禁止设定
9	PE2MD1	0	R/W	
8	PE2MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PE1MD2	0	R/W	PE1 模式位 选择 PE1/TEND0/TIOC0B/TRST 引脚的功能。当使用 E10A (ASEMD0=Low 电平) 时, 此引脚被固定为 TRST 输入。 000: PE1 输入 / 输出 (端口) 001: TIOC0B 输入 / 输出 (MTU2) 010: TEND0 输出 (DMAC) * 其他: 禁止设定
5	PE1MD1	0	R/W	
4	PE1MD0	0	R/W	
3、2	—	全 0	R	保留位 读写的值总是 0。
1	PE0MD1	0	R/W	PE0 模式位 选择 PE0/DREQ0/TIOC0A/TMS 引脚的功能。当使用 E10A (ASEMD0=Low 电平) 时, 此引脚被固定为 TMS 输入。 00: PE0 输入 / 输出 (端口) 01: TIOC0A 输入 / 输出 (MTU2) 10: DREQ0 输入 (DMAC) 其他: 禁止设定
0	PE0MD0	0	R/W	

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

(2) SH7084

- 端口 E 的控制寄存器 H2、H1 (PECRH2、PECRH1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 0	—	全 0	R	保留位 读写的值总是 0。

- 端口 E 的控制寄存器 L4 (PECRL4)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE15 MD2	PE15 MD1	PE15 MD0	-	PE14 MD2	PE14 MD1	PE14 MD0	-	-	PE13 MD1	PE13 MD0	-	PE12 MD2	PE12 MD1	PE12 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PE15MD2	0	R/W	PE15 模式位 选择 PE15/CKE/DACK1/TIOC4D/ $\overline{\text{IRQOUT}}$ 引脚的功能。 000: PE15 输入 / 输出 (端口) 001: TIOC4D 输入 / 输出 (MTU2) 010: DACK1 输出 (DMAC) * 011: $\overline{\text{IRQOUT}}$ 输出 (INTC) 101: CKE 输出 (BSC) * 其他: 禁止设定
13	PE15MD1	0	R/W	
12	PE15MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PE14MD2	0	R/W	PE14 模式位 选择 PE14/AH/DACK0/TIOC4C 引脚的功能。 000: PE14 输入 / 输出 (端口) 001: TIOC4C 输入 / 输出 (MTU2) 010: DACK0 输出 (DMAC) * 101: AH 输出 (BSC) * 其他: 禁止设定
9	PE14MD1	0	R/W	
8	PE14MD0	0	R/W	
7、6	—	全 0	R	保留位 读写的值总是 0。



位	位名	初始值	R/W	说 明
5	PE13MD1	0	R/W	PE13 模式位 选择 PE13/TIOC4B/MRES 引脚的功能。 00: PE13 输入 / 输出 (端口) 01: TIOC4B 输入 / 输出 (MTU2) 10: MRES 输入 (INTC) 其他: 禁止设定
4	PE13MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PE12MD2	0	R/W	PE12 模式位 选择 PE12/TIOC4A/TXD3/SCS 引脚的功能。 000: PE12 输入 / 输出 (端口) 001: TIOC4A 输入 / 输出 (MTU2) 011: TXD3 输出 (SCIF) 101: SCS 输入 / 输出 (SSU) 其他: 禁止设定
1	PE12MD1	0	R/W	
0	PE12MD0	0	R/W	

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 E 的控制寄存器 L3 (PECRL3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE11 MD2	PE11 MD1	PE11 MD0	-	PE10 MD2	PE10 MD1	PE10 MD0	-	PE9 MD2	PE9 MD1	PE9 MD0	-	PE8 MD2	PE8 MD1	PE8 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PE11MD2	0	R/W	PE11 模式位 选择 PE11/TIOC3D/RXD3/CTS3 引脚的功能。 000: PE11 输入 / 输出 (端口) 001: TIOC3D 输入 / 输出 (MTU2) 011: RXD3 输入 (SCIF) 100: CTS3 输入 (SCIF) 其他: 禁止设定
13	PE11MD1	0	R/W	
12	PE11MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PE10MD2	0	R/W	PE10 模式位 选择 PE10/TIOC3C/TXD2/SSO 引脚的功能。 000: PE10 输入 / 输出 (端口) 001: TIOC3C 输入 / 输出 (MTU2) 010: TXD2 输出 (SCI) 101: SSO 输入 / 输出 (SSU) 其他: 禁止设定
9	PE10MD1	0	R/W	
8	PE10MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说 明
6	PE9MD2	0	R/W	PE9 模式位 选择 PE9/TIOC3B/SCK3/RTS3 引脚的功能。 000: PE9 输入 / 输出 (端口) 001: TIOC3B 输入 / 输出 (MTU2) 011: SCK3 输入 / 输出 (SCIF) 100: RTS3 输出 (SCIF) 其他: 禁止设定
5	PE9MD1	0	R/W	
4	PE9MD0	0	R/W	
3	—	0	R	
2	PE8MD2	0	R/W	PE8 模式位 选择 PE8/TIOC3A/SCK2/SSCK 引脚的功能。 000: PE8 输入 / 输出 (端口) 001: TIOC3A 输入 / 输出 (MTU2) 010: SCK2 输入 / 输出 (SCI) 101: SSCK 输入 / 输出 (SSU) 其他: 禁止设定
1	PE8MD1	0	R/W	
0	PE8MD0	0	R/W	

• 端口 E 的控制寄存器 L2 (PECRL2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE7 MD2	PE7 MD1	PE7 MD0	-	PE6 MD2	PE6 MD1	PE6 MD0	-	PE5 MD2	PE5 MD1	PE5 MD0	-	PE4 MD2	PE4 MD1	PE4 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PE7MD2	0	R/W	PE7 模式位 选择 PE7/BS/TIOC2B/UBCTRG/RXD2/SSI 引脚的功能。 000: PE7 输入 / 输出 (端口) 001: TIOC2B 输入 / 输出 (MTU2) 010: RXD2 输入 (SCI) 011: BS 输出 (BSC) * 101: SSI 输入 / 输出 (SSU) 111: UBCTRG 输出 (UBC) 其他: 禁止设定
13	PE7MD1	0	R/W	
12	PE7MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PE6MD2	0	R/W	PE6 模式位 选择 PE6/CS7/TIOC2A/SCK3 引脚的功能。 000: PE6 输入 / 输出 (端口) 001: TIOC2A 输入 / 输出 (MTU2) 010: SCK3 输入 / 输出 (SCIF) 101: CS7 输出 (BSC) * 其他: 禁止设定
9	PE6MD1	0	R/W	
8	PE6MD0	0	R/W	

位	位名	初始值	R/W	说 明
7	—	0	R	保留位 读写的值总是 0。
6 5 4	PE5MD2 PE5MD1 PE5MD0	0 0 0	R/W R/W R/W	PE5 模式位 选择 PE5/ <u>CS6</u> /TIOC1B/TXD3/ <u>ASEBRKAK</u> / <u>ASEBRK</u> 引脚的功能。当使用 E10A ( <u>ASEMD0</u> =Low 电平) 时, 此引脚被固定为 <u>ASEBRKAK</u> 输出/ <u>ASEBRK</u> 输入。 000: PE5 输入 / 输出 (端口) 001: TIOC1B 输入 / 输出 (MTU2) 010: <u>TXD3</u> 输出 (SCIF) 101: <u>CS6</u> 输出 (BSC) * 其他: 禁止设定
3	—	0	R	保留位 读写的值总是 0。
2 1 0	PE4MD2 PE4MD1 PE4MD0	0 0 0	R/W R/W R/W	PE4 模式位 选择 PE4/ <u>TIOC1A</u> / <u>RXD3</u> / <u>TCK</u> 引脚的功能。当使用 E10A ( <u>ASEMD0</u> =Low 电平) 时, 此引脚被固定为 <u>TCK</u> 输入。 000: PE4 输入 / 输出 (端口) 001: TIOC1A 输入 / 输出 (MTU2) 010: <u>RXD3</u> 输入 (SCIF) 其他: 禁止设定

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 E 的控制寄存器 L1 (PECRL1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE3 MD2	PE3 MD1	PE3 MD0	-	PE2 MD2	PE2 MD1	PE2 MD0	-	PE1 MD2	PE1 MD1	PE1 MD0	-	-	PE0 MD1	PE0 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14 13 12	PE3MD2 PE3MD1 PE3MD0	0 0 0	R/W R/W R/W	PE3 模式位 选择 PE3/ <u>TEND1</u> /TIOC0D/ <u>TDO</u> 引脚的功能。当使用 E10A ( <u>ASEMD0</u> =Low 电平) 时, 此引脚被固定为 <u>TDO</u> 输出。 000: PE3 输入 / 输出 (端口) 001: TIOC0D 输入 / 输出 (MTU2) 010: <u>TEND1</u> 输出 (DMAC) * 其他: 禁止设定
11	—	0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说 明
10 9 8	PE2MD2 PE2MD1 PE2MD0	0 0 0	R/W R/W R/W	PE2 模式位 选择 PE2/DREQ1/TIOC0C/TDI 引脚的功能。当使用 E10A (ASEMD0=Low 电平) 时, 此引脚被固定为 TDI 输入。 000: PE2 输入 / 输出 (端口) 001: TIOC0C 输入 / 输出 (MTU2) 010: DREQ1 输入 (DMAC) 其他: 禁止设定
7	—	0	R	保留位 读写的值总是 0。
6 5 4	PE1MD2 PE1MD1 PE1MD0	0 0 0	R/W R/W R/W	PE1 模式位 选择 PE1/TEND0/TIOC0B/TRST 引脚功能。当使用 E10A (ASEMD0=Low 电平) 时, 此引脚被固定为 TRST 输入。 000: PE1 输入 / 输出 (端口) 001: TIOC0B 输入 / 输出 (MTU2) 010: TEND0 输出 (DMAC) * 其他: 禁止设定
3、2	—	全 0	R	保留位 读写的值总是 0。
1 0	PE0MD1 PE0MD0	0 0	R/W R/W	PE0 模式位 选择 PE0/DREQ0/TIOC0A/TMS 引脚的功能。当使用 E10A (ASEMD0=Low 电平) 时, 此引脚被固定为 TMS 输入。 00: PE0 输入 / 输出 (端口) 01: TIOC0A 输入 / 输出 (MTU2) 10: DREQ0 输入 (DMAC) 其他: 禁止设定

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

(3) SH7085

- 端口 E 的控制寄存器 H2、H1 (PECRH2、PECRH1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 0	—	全 0	R	保留位 读写的值总是 0。

• 端口 E 的控制寄存器 L4 (PECRL4)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE15 MD2	PE15 MD1	PE15 MD0	-	PE14 MD2	PE14 MD1	PE14 MD0	-	-	PE13 MD1	PE13 MD0	-	PE12 MD2	PE12 MD1	PE12 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PE15MD2	0	R/W	PE15 模式位 选择 PE15/ <u>CKE</u> / <u>DACK1</u> / <u>TIOC4D</u> / <u>IRQOUT</u> 引脚的功能。 000: PE15 输入 / 输出 (端口) 001: <u>TIOC4D</u> 输入 / 输出 (MTU2) 010: <u>DACK1</u> 输出 (DMAC) * 011: <u>IRQOUT</u> 输出 (INTC) 101: <u>CKE</u> 输出 (BSC) * 其他: 禁止设定
13	PE15MD1	0	R/W	
12	PE15MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PE14MD2	0	R/W	PE14 模式位 选择 PE14/ <u>WRHH</u> / <u>ICIOWR</u> / <u>AH</u> / <u>DQMUU</u> / <u>DACK0</u> / <u>TIOC4C</u> 引脚的功能。 000: PE14 输入 / 输出 (端口) 001: <u>TIOC4C</u> 输入 / 输出 (MTU2) 010: <u>DACK0</u> 输出 (DMAC) * 101: <u>WRHH</u> / <u>ICIOWR</u> / <u>AH</u> / <u>DQMUU</u> 输出 (BSC) * 其他: 禁止设定
9	PE14MD1	0	R/W	
8	PE14MD0	0	R/W	
7、6	—	全 0	R	保留位 读写的值总是 0。
5	PE13MD1	0	R/W	PE13 模式位 选择 PE13/ <u>TIOC4B</u> / <u>MRES</u> / <u>ASEBRKAK</u> / <u>ASEBRK</u> 引脚的功能。当使用 E10A ( <u>ASEMD0</u> =Low 电平) 时, 此引脚被固定为 <u>ASEBRKAK</u> 输出 / <u>ASEBRK</u> 输入。 00: PE13 输入 / 输出 (端口) 01: <u>TIOC4B</u> 输入 / 输出 (MTU2) 10: <u>MRES</u> 输入 (INTC) 其他: 禁止设定
4	PE13MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说 明
2	PE12MD2	0	R/W	PE12 模式位 选择 PE12/TIOC4A/TXD3/ $\overline{\text{SCS}}$ /TCK 引脚的功能。当使用 E10A ( $\overline{\text{ASEMD0}}$ =Low 电平) 时, 此引脚被固定为 TCK 输入。 000: PE12 输入 / 输出 (端口) 001: TIOC4A 输入 / 输出 (MTU2) 011: TXD3 输出 (SCIF) 101: $\overline{\text{SCS}}$ 输入 / 输出 (SSU) 其他: 禁止设定
1	PE12MD1	0	R/W	
0	PE12MD0	0	R/W	

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 E 的控制寄存器 L3 (PECRL3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE11 MD2	PE11 MD1	PE11 MD0	-	PE10 MD2	PE10 MD1	PE10 MD0	-	PE9 MD2	PE9 MD1	PE9 MD0	-	PE8 MD2	PE8 MD1	PE8 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PE11MD2	0	R/W	PE11 模式位 选择 PE11/TIOC3D/RXD3/ $\overline{\text{CTS3}}$ /TDO 引脚的功能。当使用 E10A ( $\overline{\text{ASEMD0}}$ =Low 电平) 时, 此引脚被固定为 TDO 输出。 000: PE11 输入 / 输出 (端口) 001: TIOC3D 输入 / 输出 (MTU2) 011: RXD3 输入 (SCIF) 100: $\overline{\text{CTS3}}$ 输入 (SCIF) 其他: 禁止设定
13	PE11MD1	0	R/W	
12	PE11MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PE10MD2	0	R/W	PE10 模式位 选择 PE10/TIOC3C/TXD2/SSO/TDI 引脚的功能。当使用 E10A ( $\overline{\text{ASEMD0}}$ =Low 电平) 时, 此引脚被固定为 TDI 输入。 000: PE10 输入 / 输出 (端口) 001: TIOC3C 输入 / 输出 (MTU2) 010: TXD2 输出 (SCI) 101: SSO 输入 / 输出 (SSU) 其他: 禁止设定
9	PE10MD1	0	R/W	
8	PE10MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说 明
6	PE9MD2	0	R/W	PE9 模式位 选择 PE9/TIOC3B/SCK3/RTS3/TRST 引脚的功能。当使用 E10A (ASEMD0=Low 电平) 时, 此引脚被固定为 TRST 输入。 000: PE9 输入 / 输出 (端口) 001: TIOC3B 输入 / 输出 (MTU2) 011: SCK3 输入 / 输出 (SCIF) 100: RTS3 输出 (SCIF) 其他: 禁止设定
5	PE9MD1	0	R/W	
4	PE9MD0	0	R/W	
3	—	0	R	
2	PE8MD2	0	R/W	PE8 模式位 选择 PE8/TIOC3A/SCK2/SSCK/TMS 引脚的功能。当使用 E10A (ASEMD0=Low 电平) 时, 此引脚被固定为 TMS 输入。 000: PE8 输入 / 输出 (端口) 001: TIOC3A 输入 / 输出 (MTU2) 010: SCK2 输入 / 输出 (SCI) 101: SSCK 输入 / 输出 (SSU) 其他: 禁止设定
1	PE8MD1	0	R/W	
0	PE8MD0	0	R/W	

• 端口 E 的控制寄存器 L2 (PECRL2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE7 MD2	PE7 MD1	PE7 MD0	-	PE6 MD2	PE6 MD1	PE6 MD0	-	PE5 MD2	PE5 MD1	PE5 MD0	-	PE4 MD2	PE4 MD1	PE4 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PE7MD2	0	R/W	PE7 模式位 选择 PE7/BS/TIOC2B/UBCTRG/RXD2/SSI 引脚的功能。 000: PE7 输入 / 输出 (端口) 001: TIOC2B 输入 / 输出 (MTU2) 010: RXD2 输入 (SCI) 011: BS 输出 (BSC) * 101: SSI 输入 / 输出 (SSU) 111: UBCTRG 输出 (UBC) 其他: 禁止设定
13	PE7MD1	0	R/W	
12	PE7MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说 明
10 9 8	PE6MD2 PE6MD1 PE6MD0	0 0 0	R/W R/W R/W	PE6 模式位 选择 PE6/ $\overline{\text{CS7}}$ /TIOC2A/SCK3/AUDATA0 引脚的功能。当使用 E10A 的 AUD 功能时，此引脚被固定为 AUDATA0 输出。 000: PE6 输入 / 输出 (端口) 001: TIOC2A 输入 / 输出 (MTU2) 010: SCK3 输入 / 输出 (SCIF) 101: CS7 输出 (BSC) * 其他: 禁止设定
7	—	0	R	保留位 读写的值总是 0。
6 5 4	PE5MD2 PE5MD1 PE5MD0	0 0 0	R/W R/W R/W	PE5 模式位 选择 PE5/ $\overline{\text{CS6}}$ / $\overline{\text{CE1B}}$ /TIOC1B/TXD3/AUDATA1 引脚的功能。当使用 E10A 的 AUD 功能时，此引脚被固定为 AUDATA1 输出。 000: PE5 输入 / 输出 (端口) 001: TIOC1B 输入 / 输出 (MTU2) 010: TXD3 输出 (SCIF) 101: $\overline{\text{CS6}}$ / $\overline{\text{CE1B}}$ 输出 (BSC) * 其他: 禁止设定
3	—	0	R	保留位 读写的值总是 0。
2 1 0	PE4MD2 PE4MD1 PE4MD0	0 0 0	R/W R/W R/W	PE4 模式位 选择 PE4/ $\overline{\text{IOIS16}}$ /TIOC1A/RXD3/AUDATA2 引脚的功能。当使用 E10A 的 AUD 功能时，此引脚被固定为 AUDATA2 输出。 000: PE4 输入 / 输出 (端口) 001: TIOC1A 输入 / 输出 (MTU2) 010: RXD3 输入 (SCIF) 101: $\overline{\text{IOIS16}}$ 输入 (BSC) * 其他: 禁止设定

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。



• 端口 E 的控制寄存器 L1 (PECRL1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE3 MD2	PE3 MD1	PE3 MD0	-	PE2 MD2	PE2 MD1	PE2 MD0	-	PE1 MD2	PE1 MD1	PE1 MD0	-	-	PE0 MD1	PE0 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PE3MD2	0	R/W	PE3 模式位 选择 PE3/TEND1/TIOC0D/AUDATA3 引脚的功能。当使用 E10A 的 AUD 功能时，此引脚被固定为 AUDATA3 输出。 000: PE3 输入 / 输出 (端口) 001: TIOC0D 输入 / 输出 (MTU2) 010: TEND1 输出 (DMAC) * 其他: 禁止设定
13	PE3MD1	0	R/W	
12	PE3MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PE2MD2	0	R/W	PE2 模式位 选择 PE2/DREQ1/TIOC0C 引脚的功能。 000: PE2 输入 / 输出 (端口) 001: TIOC0C 输入 / 输出 (MTU2) 010: DREQ1 输入 (DMAC) 其他: 禁止设定
9	PE2MD1	0	R/W	
8	PE2MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PE1MD2	0	R/W	PE1 模式位 选择 PE1/TEND0/TIOC0B 引脚的功能。 000: PE1 输入 / 输出 (端口) 001: TIOC0B 输入 / 输出 (MTU2) 010: TEND0 输出 (DMAC) * 其他: 禁止设定
5	PE1MD1	0	R/W	
4	PE1MD0	0	R/W	
3、2	—	全 0	R	保留位 读写的值总是 0。
1	PE0MD1	0	R/W	PE0 模式位 P 选择 E0/DREQ0/TIOC0A/AUDCK 引脚的功能。当使用 E10A 的 AUD 功能时，此引脚被固定为 AUDCK 输出。 00: PE0 输入 / 输出 (端口) 01: TIOC0A 输入 / 输出 (MTU2) 10: DREQ0 输入 (DMAC) 其他: 禁止设定
0	PE0MD0	0	R/W	

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

(4) SH7086

- 端口 E 的控制寄存器 H2 (PECRH2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PE21 MD1	PE21 MD0	-	-	PE20 MD1	PE20 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 6	—	全 0	R	保留位 读写的值总是 0。
5 4	PE21MD1 PE21MD0	0 0	R/W R/W	PE21 模式位 选择 PE21/TIOC4DS 引脚的功能。 00: PE21 输入 / 输出 (端口) 01: TIOC4DS 输入 / 输出 (MTU2S) 其他: 禁止设定
3、2	—	全 0	R	保留位 读写的值总是 0。
1 0	PE20MD1 PE20MD0	0 0	R/W R/W	PE20 模式位 选择 PE20/TIOC4CS 引脚的功能。 00: PE20 输入 / 输出 (端口) 01: TIOC4CS 输入 / 输出 (MTU2S) 其他: 禁止设定

• 端口 E 的控制寄存器 H1 (PECRH1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PE19 MD1	PE19 MD0	-	-	PE18 MD1	PE18 MD0	-	-	PE17 MD1	PE17 MD0	-	PE16 MD2	PE16 MD1	PE16 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15、14	—	全 0	R	保留位 读写的值总是 0。
13 12	PE19MD1 PE19MD0	0 0	R/W R/W	PE19 模式位 选择 PE19/TIOC4BS 引脚的功能。 00: PE19 输入 / 输出 (端口) 01: TIOC4BS 输入 / 输出 (MTU2S) 其他: 禁止设定
11、10	—	全 0	R	保留位 读写的值总是 0。
9 8	PE18MD1 PE18MD0	0 0	R/W R/W	PE18 模式位 选择 PE18/TIOC4AS 引脚的功能。 00: PE18 输入 / 输出 (端口) 01: TIOC4AS 输入 / 输出 (MTU2S) 其他: 禁止设定
7、6	—	全 0	R	保留位 读写的值总是 0。
5 4	PE17MD1 PE17MD0	0 0	R/W R/W	PE17 模式位 选择 PE17/TIOC3DS 引脚的功能。 00: PE17 输入 / 输出 (端口) 01: TIOC3DS 输入 / 输出 (MTU2S) 其他: 禁止设定
3	—	0	R	保留位 读写的值总是 0。
2 1 0	PE16MD2 PE16MD1 PE16MD0	0 0 0	R/W R/W R/W	PE16 模式位 选择 PE16/CS8/TIOC3BS 引脚的功能。 000: PE16 输入 / 输出 (端口) 001: TIOC3BS 输入 / 输出 (MTU2S) 101: CS8 输出 (BSC) * 其他: 禁止设定

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 E 的控制寄存器 L4 (PECRL4)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE15 MD2	PE15 MD1	PE15 MD0	-	PE14 MD2	PE14 MD1	PE14 MD0	-	-	PE13 MD1	PE13 MD0	-	PE12 MD2	PE12 MD1	PE12 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写的值总是 0。
14	PE15MD2	0	R/W	PE15 模式位 选择 PE15/CKE/DACK1/TIOC4D/ $\overline{\text{IRQOUT}}$ 引脚的功能。 000: PE15 输入 / 输出 (端口) 001: TIOC4D 输入 / 输出 (MTU2) 010: DACK1 输出 (DMAC) * 011: $\overline{\text{IRQOUT}}$ 输出 (INTC) 101: CKE 输出 (BSC) * 其他: 禁止设定
13	PE15MD1	0	R/W	
12	PE15MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PE14MD2	0	R/W	PE14 模式位 选择 PE14/WRHH/ICIOWR/AH/DQMUU/DACK0/TIOC4C 引脚的功能。 000: PE14 输入 / 输出 (端口) 001: TIOC4C 输入 / 输出 (MTU2) 010: DACK0 输出 (DMAC) * 101: WRHH/ICIOWR/AH/DQMUU 输出 (BSC) * 其他: 禁止设定
9	PE14MD1	0	R/W	
8	PE14MD0	0	R/W	
7、6	—	全 0	R	保留位 读写的值总是 0。
5	PE13MD1	0	R/W	PE13 模式位 选择 PE13/TIOC4B/MRES/ASEBRKAK/ASEBRK 引脚的功能。当使用 E10A (ASEMD0=Low 电平) 时, 此引脚被固定为 ASEBRKAK 输出 / ASEBRK 输入。 00: PE13 输入 / 输出 (端口) 01: TIOC4B 输入 / 输出 (MTU2) 10: MRES 输入 (INTC) 其他: 禁止设定
4	PE13MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PE12MD2	0	R/W	PE12 模式位 选择 PE12/TIOC4A/TXD3/SCS/TCK 引脚的功能。当使用 E10A (ASEMD0=Low 电平) 时, 此引脚被固定为 TCK 输入。 000: PE12 输入 / 输出 (端口) 001: TIOC4A 输入 / 输出 (MTU2) 011: TXD3 输出 (SCIF) 101: SCS 输入 / 输出 (SSU) 其他: 禁止设定
1	PE12MD1	0	R/W	
0	PE12MD0	0	R/W	

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 E 的控制寄存器 L3 (PECRL3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE11 MD2	PE11 MD1	PE11 MD0	-	PE10 MD2	PE10 MD1	PE10 MD0	-	PE9 MD2	PE9 MD1	PE9 MD0	-	PE8 MD2	PE8 MD1	PE8 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PE11MD2	0	R/W	PE11 模式位 选择 PE11/TIOC3D/RXD3/ $\overline{\text{CTS3}}$ /TDO 引脚的功能。当使用 E10A (ASEMD0=Low 电平) 时, 此引脚被固定为 TDO 输出。 000: PE11 输入 / 输出 (端口) 001: TIOC3D 输入 / 输出 (MTU2) 011: RXD3 输入 (SCIF) 100: $\overline{\text{CTS3}}$ 输入 (SCIF) 其他: 禁止设定
13	PE11MD1	0	R/W	
12	PE11MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PE10MD2	0	R/W	PE10 模式位 选择 PE10/TIOC3C/TXD2/SSO/TDI 引脚的功能。当使用 E10A (ASEMD0=Low 电平) 时, 此引脚被固定为 TDI 输入。 000: PE10 输入 / 输出 (端口) 001: TIOC3C 输入 / 输出 (MTU2) 010: TXD2 输出 (SCI) 101: SSO 输入 / 输出 (SSU) 其他: 禁止设定
9	PE10MD1	0	R/W	
8	PE10MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PE9MD2	0	R/W	PE9 模式位 选择 PE9/TIOC3B/SCK3/RTS3/TRST 引脚的功能。当使用 E10A (ASEMD0=Low 电平) 时, 此引脚被固定为 TRST 输入。 000: PE9 输入 / 输出 (端口) 001: TIOC3B 输入 / 输出 (MTU2) 011: SCK3 输入 / 输出 (SCIF) 100: RTS3 输出 (SCIF) 其他: 禁止设定
5	PE9MD1	0	R/W	
4	PE9MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PE8MD2	0	R/W	PE8 模式位 选择 PE8/TIOC3A/SCK2/SSCK/TMS 引脚的功能。当使用 E10A (ASEMD0=Low 电平) 时, 此引脚被固定为 TMS 输入。 000: PE8 输入 / 输出 (端口) 001: TIOC3A 输入 / 输出 (MTU2) 010: SCK2 输入 / 输出 (SCI) 101: SSCK 输入 / 输出 (SSU) 其他: 禁止设定
1	PE8MD1	0	R/W	
0	PE8MD0	0	R/W	

• 端口 E 的控制寄存器 L2 (PECRL2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE7 MD2	PE7 MD1	PE7 MD0	-	PE6 MD2	PE6 MD1	PE6 MD0	-	PE5 MD2	PE5 MD1	PE5 MD0	-	PE4 MD2	PE4 MD1	PE4 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写的值总是 0。
14	PE7MD2	0	R/W	PE7 模式位 选择 PE7/BS/TIOC2B/UBCTRG/RXD2/SSI 引脚的功能。 000: PE7 输入 / 输出 (端口) 001: TIOC2B 输入 / 输出 (MTU2) 010: RXD2 输入 (SCI) 011: BS 输出 (BSC) * 101: SSI 输入 / 输出 (SSU) 111: UBCTRG 输出 (UBC) 其他: 禁止设定
13	PE7MD1	0	R/W	
12	PE7MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PE6MD2	0	R/W	PE6 模式位 选择 PE6/CS7/TIOC2A/SCK3/AUDATA0 引脚的功能。当使用 E10A 的 AUD 功能时, 此引脚被固定为 AUDATA0 输出。 000: PE6 输入 / 输出 (端口) 001: TIOC2A 输入 / 输出 (MTU2) 010: SCK3 输入 / 输出 (SCIF) 101: CS7 输出 (BSC) * 其他: 禁止设定
9	PE6MD1	0	R/W	
8	PE6MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PE5MD2	0	R/W	PE5 模式位 选择 PE5/CS6/CE1B/TIOC1B/TXD3/AUDATA1 引脚的功能。当使用 E10A 的 AUD 功能时, 此引脚被固定为 AUDATA1 输出。 000: PE5 输入 / 输出 (端口) 001: TIOC1B 输入 / 输出 (MTU2) 010: TXD3 输出 (SCIF) 101: CS6/CE1B 输出 (BSC) * 其他: 禁止设定
5	PE5MD1	0	R/W	
4	PE5MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PE4MD2	0	R/W	PE4 模式位 选择 PE4/IOIS16/TIOC1A/RXD3/AUDATA2 引脚的功能。当使用 E10A 的 AUD 功能时, 此引脚被固定为 AUDATA2 输出。 000: PE4 输入 / 输出 (端口) 001: TIOC1A 输入 / 输出 (MTU2) 010: RXD3 输入 (SCIF) 101: IOIS16 输入 (BSC) * 其他: 禁止设定
1	PE4MD1	0	R/W	
0	PE4MD0	0	R/W	

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

• 端口 E 的控制寄存器 L1 (PECRL1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE3 MD2	PE3 MD1	PE3 MD0	-	PE2 MD2	PE2 MD1	PE2 MD0	-	PE1 MD2	PE1 MD1	PE1 MD0	-	-	PE0 MD1	PE0 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PE3MD2	0	R/W	PE3 模式位 选择 PE3/TEND1/TIOC0D/AUDATA3 引脚的功能。当使用 E10A 的 AUD 功能时，此引脚被固定为 AUDATA3 输出。 000: PE3 输入 / 输出 (端口) 001: TIOC0D 输入 / 输出 (MTU2) 010: TEND1 输出 (DMAC) * 其他: 禁止设定
13	PE3MD1	0	R/W	
12	PE3MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PE2MD2	0	R/W	PE2 模式位 选择 PE2/DREQ1/TIOC0C 引脚功能。 000: PE2 输入 / 输出 (端口) 001: TIOC0C 输入 / 输出 (MTU2) 010: DREQ1 输入 (DMAC) 其他: 禁止设定
9	PE2MD1	0	R/W	
8	PE2MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PE1MD2	0	R/W	PE1 模式位 选择 PE1/TEND0/TIOC0B 引脚的功能。 000: PE1 输入 / 输出 (端口) 001: TIOC0B 输入 / 输出 (MTU2) 010: TEND0 输出 (DMAC) * 其他: 禁止设定
5	PE1MD1	0	R/W	
4	PE1MD0	0	R/W	
3、2	—	全 0	R	保留位 读写的值总是 0。
1	PE0MD1	0	R/W	PE0 模式位 选择 PE0/DREQ0/TIOC0A/AUDCK 引脚的功能。当使用 E10A 的 AUD 功能时，此引脚被固定为 AUDCK 输出。 00: PE0 输入 / 输出 (端口) 01: TIOC0A 输入 / 输出 (MTU2) 10: DREQ0 输入 (DMAC) 其他: 禁止设定
0	PE0MD0	0	R/W	

【注】 \* 只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定。

## 21.1.11 大电流端口的控制寄存器 (HCPCR)

HCPCR 是 16 位可读写寄存器，控制大电流端口（SH7083 中的 PD9、PD11 ~ PD15、PE12 ~ PE15 共 10 个引脚，SH7084 中的 PD9、PD11 ~ PD15、PE9、PE11 ~ PE15 共 12 个引脚，SH7085 中的 PD9、PD11 ~ PD15、PD24 ~ PD29、PE9、PE11 ~ PE15 共 18 个引脚，SH7086 中的 PD9、PD11 ~ PD15、PD24 ~ PD29、PE9、PE11 ~ PE21 共 24 个引脚）。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	MZI ZDH	MZI ZDL	MZI ZEH	MZI ZEL
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 4	—	全 0	R	保留位 读写的值总是 0。
3	MZIZDH	1	R/W	端口 D 的大电流端口的高阻抗 H 在检测到振荡停止时或者在软件待机模式中，与 PFC 的设定无关，选择是否将 PD24 ~ PD29 的大电流端口置为高阻抗状态。 0: 高阻抗 1: 非高阻抗 如果将此位设定为 1，就在检测到振荡停止时保持引脚状态。有关软件待机模式中的引脚状态，请参照“附录 A. 引脚状态”。
2	MZIZDL	1	R/W	端口 D 的大电流端口的高阻抗 L 在检测到振荡停止时或者在软件待机模式中，与 PFC 的设定无关，选择是否将 PD9、PD11 ~ PD15 的大电流端口置为高阻抗状态。 0: 高阻抗 1: 非高阻抗 如果将此位设定为 1，就在检测到振荡停止时保持引脚状态。有关软件待机模式中的引脚状态，请参照“附录 A. 引脚状态”。
1	MZIZEH	1	R/W	端口 E 的大电流端口的高阻抗 H 在检测到振荡停止时或者在软件待机模式中，与 PFC 的设定无关，选择是否将 PE16 ~ PE21 的大电流端口置为高阻抗状态。 0: 高阻抗 1: 非高阻抗 如果将此位设定为 1，就在检测到振荡停止时保持引脚状态。有关软件待机模式中的引脚状态，请参照“附录 A. 引脚状态”。
0	MZIZEL	1	R/W	端口 E 的大电流端口的高阻抗 L 在检测到振荡停止时或者在软件待机模式中，与 PFC 的设定无关，选择是否将 PE9、PE11 ~ PE15 的大电流端口置为高阻抗状态。 0: 高阻抗 1: 非高阻抗 如果将此位设定为 1，就在检测到振荡停止时保持引脚状态。有关软件待机模式中的引脚状态，请参照“附录 A. 引脚状态”。



### 21.1.12 IRQOUT 功能的控制寄存器 (IFCR)

IFCR 是 16 位可读写寄存器，在通过端口 D 的控制寄存器 H4 (PDCRH4) 和端口 E 的控制寄存器 L4 (PECRL4) 将多路复用功能设定为 IRQOUT 输出时，用于控制 IRQOUT 引脚的输出。当 PDCRH4 或者 PECRL4 被设定为其他功能时，IFCR 的设定不影响引脚功能。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	IRQ MD3	IRQ MD2	IRQ MD1	IRQ MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 4	—	全 0	R	保留位 读写的值总是 0。
3 2	IRQMD3 IRQMD2	0 0	R/W R/W	端口 D 的 $\overline{\text{IRQOUT}}$ 引脚功能的选择 当 PDCRH4 的 bit9、bit8 (PD30MD1、PD30MD0) 被置为 (1、0) 时，选择 $\overline{\text{IRQOUT}}$ 引脚的功能。 00: 输出中断请求的接受信号 01: 输出刷新信号 10: 输出中断请求的接受信号或者刷新信号 (取决于此时的运行状态) 11: 总是输出高电平
1 0	IRQMD1 IRQMD0	0 0	R/W R/W	端口 E 的 $\overline{\text{IRQOUT}}$ 引脚功能的选择 当 PECRL4 的 bit14、bit13、bit12 (PE15MD2、PE15MD1、PE15MD0) 被置为 (0、1、1) 时，选择 $\overline{\text{IRQOUT}}$ 引脚的功能。 00: 输出中断请求的接受信号 01: 输出刷新信号 10: 输出中断请求的接受信号或者刷新信号 (取决于此时的运行状态) 11: 总是输出高电平

## 21.2 使用注意事项

1. 在本 LSI 中，同一功能作为多路复用功能被分配到多个引脚。这是为了提高引脚功能的选择自由度的同时方便电路板的设计，但是在 2 个以上的引脚使用 1 个功能时，必须注意以下几点：
  - 引脚功能为输入功能的情况  
通过 OR 或者 AND 逻辑将多个引脚的输入信号组合为 1 个信号，传送到 LSI 内部。因此，根据其同一功能的引脚输入状态，有可能和输入信号不同的信号传送到 LSI 内部。分配到多个引脚的输入功能的传送格式如表 21.22 所示，在 2 个以上的引脚使用以下任意功能时，必须考虑传送格式以及注意信号的极性。

表 21.22 分配到多个引脚的输入功能的传送格式

OR 型	AND 型
SCK0、SCK3、RXD0、RXD3、 TIOC3AS ~ TIOC3DS、TIOC4AS ~ TIOC4DS、 TIC5U、TIC5V、TIC5W、TIC5US、TIC5VS、TIC5WS	IRQ0 ~ IRQ7、DREQ0、DREQ1、 $\overline{\text{BREQ}}$ 、 $\overline{\text{WAIT}}$ 、 $\overline{\text{ADTRG}}$ 、 $\overline{\text{POE4}}$ ~ $\overline{\text{POE8}}$

OR 型：通过 OR 逻辑将多个引脚的输入信号组合为 1 个信号，传送到 LSI 内部。

AND 型：通过 AND 逻辑将多个引脚的输入信号组合为 1 个信号，传送到 LSI 内部。

- 引脚功能为输出功能的情况  
能从所选的全部引脚输出同一功能。
2. 输入 / 输出端口和 DREQ 或者 IRQ 是多路复用引脚，如果端口的输入从低电平状态转换为 DREQ 或者 IRQ 边沿检测，就检测该边沿。
  3. 只能设定表 21.17 ~ 表 21.20 中 PFC 可设定的功能，否则就不能保证运行。
  4. 关于在单芯片模式 (MCU 运行模式 3) 中的 PFC 设定  
在单芯片模式中，不要通过 PFC 选择地址总线、数据总线、总线控制信号、 $\overline{\text{BREQ}}$ 、 $\overline{\text{BACK}}$ 、CK、DACK、TEND。选择后，地址总线为高电平或低电平输出、地址总线为高阻抗输出、其他输出信号为高电平输出。因为  $\overline{\text{BREQ}}$  和  $\overline{\text{WAIT}}$  为输入所以请不要设定为开路。但是，总线权请求输入和外部等待无效。

## 第 22 章 I/O 端口

SH7083 的端口由 A、B、C、D、E 和 F 共 6 个端口构成。端口 A、B、C、D、E 分别是 11 位、9 位、16 位、16 位、13 位的输入/输出端口，端口 F 是 8 位输入专用端口。

SH7084 的端口由 A、B、C、D、E 和 F 共 6 个端口构成。端口 A、B、C、D、E 分别是 18 位、10 位、16 位、16 位、16 位的输入/输出端口，端口 F 是 8 位输入专用端口。

SH7085 的端口由 A、B、C、D、E 和 F 共 6 个端口构成。端口 A、B、C、D、E 分别是 26 位、10 位、16 位、32 位、16 位的输入/输出端口，端口 F 是 8 位输入专用端口。

SH7086 的端口由 A、B、C、D、E 和 F 共 6 个端口构成。端口 A、B、C、D、E 分别是 30 位、10 位、24 位、32 位、22 位的输入/输出端口，端口 F 是 16 位输入专用端口。

各端口的引脚都是和其他功能兼用的多路复用引脚，通过引脚功能控制器（PFC）选择多路复用引脚的功能。

各端口有保存引脚数据的数据寄存器。

## 22.1 端口 A

如图 22.1 所示，SH7083 的端口 A 是有 11 个引脚的输入/输出端口。



图 22.1 端口 A (SH7083)

如图 22.2 所示，SH7084 的端口 A 是有 18 个引脚的输入/输出端口。

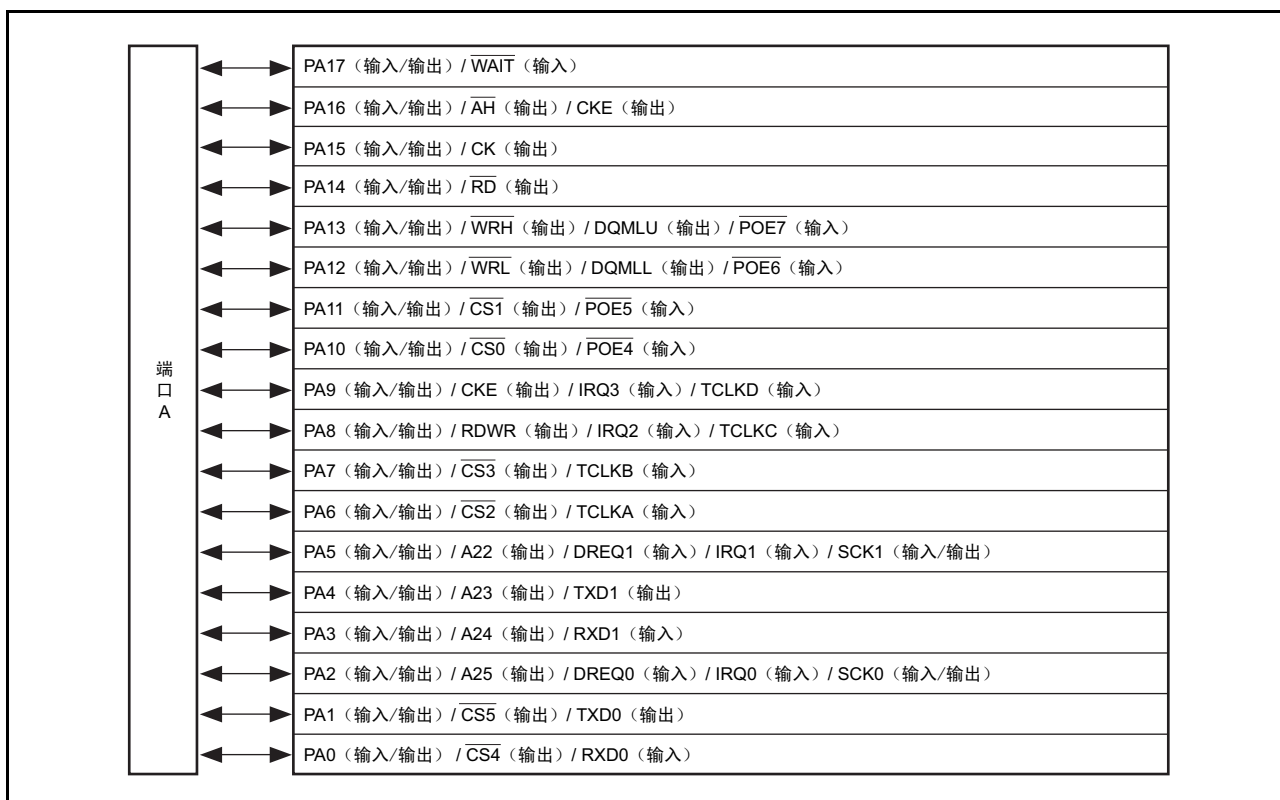


图 22.2 端口 A (SH7084)

如图 22.3 所示，SH7085 的端口 A 是有 26 个引脚的输入 / 输出端口。

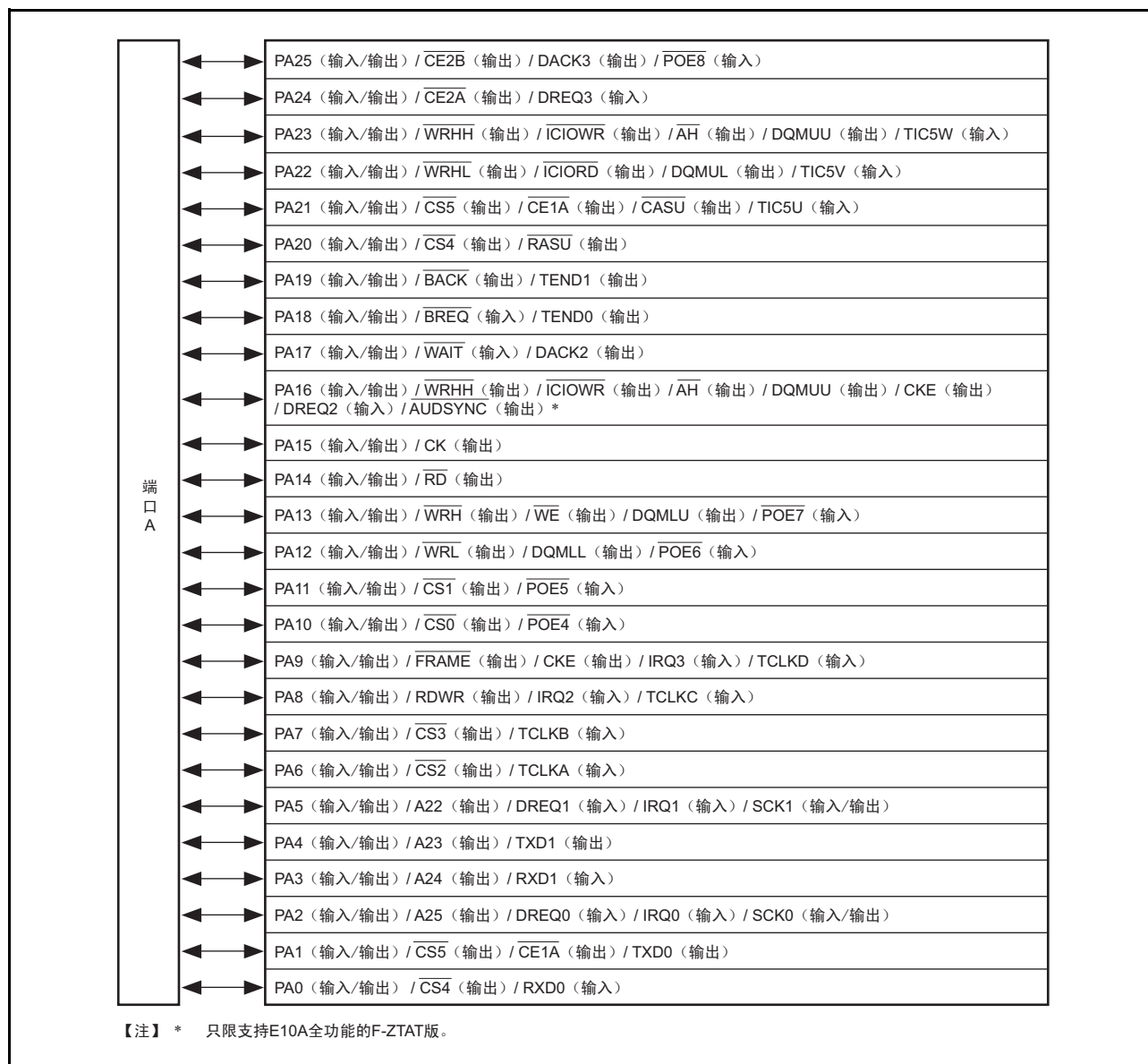


图 22.3 端口 A (SH7085)

如图 22.4 所示，SH7086 的端口 A 是有 30 个引脚的输入 / 输出端口。

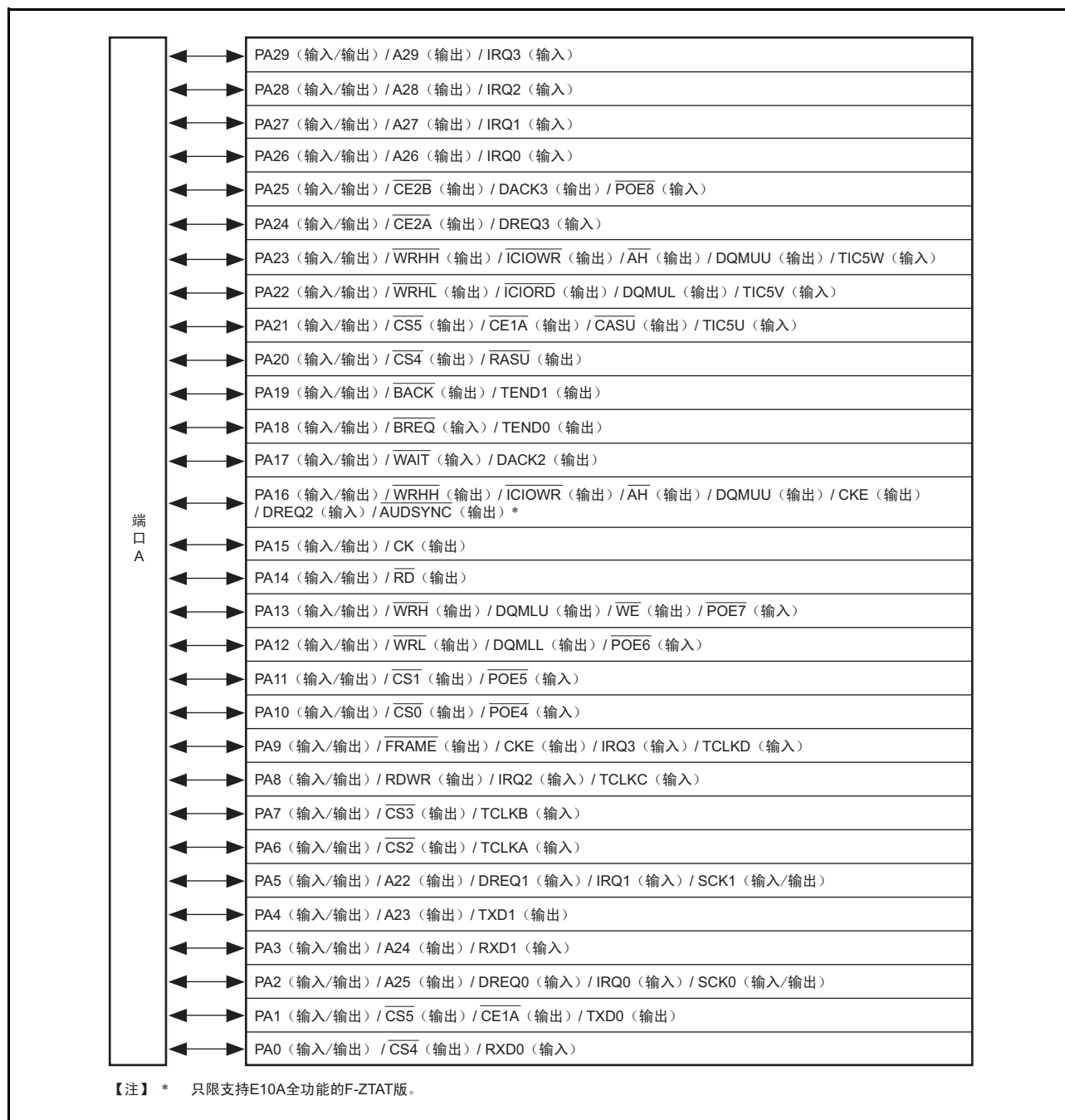


图 22.4 端口 A (SH7086)

### 22.1.1 寄存器说明

SH7083 的端口 A 是 11 位输入 / 输出端口，SH7084 的端口 A 是 18 位输入 / 输出端口，SH7085 的端口 A 是 26 位输入 / 输出端口，SH7086 的端口 A 是 30 位输入 / 输出端口。端口 A 有以下寄存器，有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 27 章 寄存器一览”。

表 22.1 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
端口 A 的数据寄存器 H	PADRH	R/W	H'0000	H'FFFFD100	8、16、32
端口 A 的数据寄存器 L	PADRL	R/W	H'0000	H'FFFFD102	8、16
端口 A 的端口寄存器 H	PAPRH	R	—	H'FFFFD11C	8、16、32
端口 A 的端口寄存器 L	PAPRL	R	—	H'FFFFD11E	8、16

### 22.1.2 端口 A 的数据寄存器 H、L (PADRH、PADRL)

PADRH 和 PADRL 是 16 位可读写寄存器，保存端口 A 的数据。SH7083 的 PA15DR ~ PA12DR 位、PA10DR ~ PA7DR 位、PA5DR ~ PA3DR 位分别对应 PA15 ~ PA12 引脚、PA10 ~ PA7 引脚、PA5 ~ PA3 引脚（省略有关兼用功能的记述）；SH7084 的 PA17DR ~ PA0DR 位分别对应 PA17 ~ PA0 引脚（省略有关兼用功能的记述）；SH7085 的 PA25DR ~ PA0DR 位分别对应 PA25 ~ PA0 引脚（省略有关兼用功能的记述）；SH7086 的 PA29DR ~ PA0DR 位分别对应 PA29 ~ PA0 引脚（省略有关兼用功能的记述）。

在引脚功能为通用输出时，如果给 PADRH 或者 PADRL 写值，就从引脚输出该值；如果读 PADRH 或者 PADRL，就直接读寄存器的值，与引脚的状态无关。

在引脚功能为通用输入时，如果读 PADRH 或者 PADRL，就直接读引脚的状态而不读寄存器的值；如果给 PADRH 或者 PADRL 写值，就能将值写到 PADRH 或者 PADRL，但是不影响引脚的状态。端口 A 的数据寄存器的读写操作如表 22.2 所示。

#### • PADRH (SH7083)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 0	—	全 0	R	保留位 读写的值总是 0。

## • PADRH (SH7084)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PA17 DR	PA16 DR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 2	—	全 0	R	保留位 读写的值总是 0。
1	PA17DR	0	R/W	参照表 22.2
0	PA16DR	0	R/W	

## • PADRH (SH7085)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PA25 DR	PA24 DR	PA23 DR	PA22 DR	PA21 DR	PA20 DR	PA19 DR	PA18 DR	PA17 DR	PA16 DR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 10	—	全 0	R	保留位 读写的值总是 0。
9	PA25DR	0	R/W	参照表 22.2
8	PA24DR	0	R/W	
7	PA23DR	0	R/W	
6	PA22DR	0	R/W	
5	PA21DR	0	R/W	
4	PA20DR	0	R/W	
3	PA19DR	0	R/W	
2	PA18DR	0	R/W	
1	PA17DR	0	R/W	
0	PA16DR	0	R/W	



## • PADRH (SH7086)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PA29 DR	PA28 DR	PA27 DR	PA26 DR	PA25 DR	PA24 DR	PA23 DR	PA22 DR	PA21 DR	PA20 DR	PA19 DR	PA18 DR	PA17 DR	PA16 DR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15、14	—	全 0	R	保留位 读写的值总是 0。
13	PA29DR	0	R/W	参照表 22.2
12	PA28DR	0	R/W	
11	PA27DR	0	R/W	
10	PA26DR	0	R/W	
9	PA25DR	0	R/W	
8	PA24DR	0	R/W	
7	PA23DR	0	R/W	
6	PA22DR	0	R/W	
5	PA21DR	0	R/W	
4	PA20DR	0	R/W	
3	PA19DR	0	R/W	
2	PA18DR	0	R/W	
1	PA17DR	0	R/W	
0	PA16DR	0	R/W	

## • PADRL (SH7083)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15 DR	PA14 DR	PA13 DR	PA12 DR	-	PA10 DR	PA9 DR	PA8 DR	PA7 DR	-	PA5 DR	PA4 DR	PA3 DR	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R

位	位名	初始值	R/W	说 明
15	PA15DR	0	R/W	参照表 22.2
14	PA14DR	0	R/W	
13	PA13DR	0	R/W	
12	PA12DR	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PA10DR	0	R/W	参照表 22.2
9	PA9DR	0	R/W	
8	PA8DR	0	R/W	
7	PA7DR	0	R/W	
6	—	0	R	保留位 读写的值总是 0。
5	PA5DR	0	R/W	参照表 22.2
4	PA4DR	0	R/W	
3	PA3DR	0	R/W	
2 ~ 0	—	全 0	R	保留位 读写的值总是 0。

- PADRL (SH7084/85/86)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15 DR	PA14 DR	PA13 DR	PA12 DR	PA11 DR	PA10 DR	PA9 DR	PA8 DR	PA7 DR	PA6 DR	PA5 DR	PA4 DR	PA3 DR	PA2 DR	PA1 DR	PA0 DR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	PA15DR	0	R/W	参照表 22.2
14	PA14DR	0	R/W	
13	PA13DR	0	R/W	
12	PA12DR	0	R/W	
11	PA11DR	0	R/W	
10	PA10DR	0	R/W	
9	PA9DR	0	R/W	
8	PA8DR	0	R/W	
7	PA7DR	0	R/W	
6	PA6DR	0	R/W	
5	PA5DR	0	R/W	
4	PA4DR	0	R/W	
3	PA3DR	0	R/W	
2	PA2DR	0	R/W	
1	PA1DR	0	R/W	
0	PA0DR	0	R/W	

表 22.2 端口 A 的数据寄存器 (PADR) 的读写操作

- PADRH 的 bit13 ~ 0 和 PADRL 的 bit15 ~ 0

PAIOR	引脚功能	读	写
0	通用输入	引脚的状态	能写 PADRH、L, 但是不影响引脚的状态
	非通用输入	引脚的状态	能写 PADRH、L, 但是不影响引脚的状态
1	通用输出	PADRH、L 的值	从引脚输出所写的值
	非通用输出	PADRH、L 的值	能写 PADRH、L, 但是不影响引脚的状态

### 22.1.3 端口 A 的端口寄存器 H、L (PAPRH、PAPRL)

PAPRH 和 PAPRL 是 16 位只读寄存器，能随时读引脚的状态，与 PFC 的设定无关。SH7083 的 PA15PR ~ PA12PR 位、PA10PR ~ PA7PR 位、PA5PR ~ PA3PR 位分别对应 PA15 ~ PA12 引脚、PA10 ~ PA7 引脚、PA5 ~ PA3 引脚（省略有关兼用功能的记述）；SH7084 的 PA17PR ~ PA0PR 位分别对应 PA17 ~ PA0 引脚（省略有关兼用功能的记述）；SH7085 的 PA25PR ~ PA0PR 位分别对应 PA25 ~ PA0 引脚（省略有关兼用功能的记述）；SH7086 的 PA29PR ~ PA0PR 位分别对应 PA29 ~ PA0 引脚（省略有关兼用功能的记述）。

#### • PAPRH (SH7083)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 0	—	全 0	R	保留位 读写的值总是 0。

#### • PAPRH (SH7084)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PA17 PR	PA16 PR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 2	—	全 0	R	保留位 读写的值总是 0。
1	PA17PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
0	PA16PR	引脚的状态	R	

## • PAPRH (SH7085)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PA25 PR	PA24 PR	PA23 PR	PA22 PR	PA21 PR	PA20 PR	PA19 PR	PA18 PR	PA17 PR	PA16 PR
初始值:	0	0	0	0	0	0	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 10	—	全 0	R	保留位 读写的值总是 0。
9	PA25PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
8	PA24PR	引脚的状态	R	
7	PA23PR	引脚的状态	R	
6	PA22PR	引脚的状态	R	
5	PA21PR	引脚的状态	R	
4	PA20PR	引脚的状态	R	
3	PA19PR	引脚的状态	R	
2	PA18PR	引脚的状态	R	
1	PA17PR	引脚的状态	R	
0	PA16PR	引脚的状态	R	

## • PAPRH (SH7086)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PA29 PR	PA28 PR	PA27 PR	PA26 PR	PA25 PR	PA24 PR	PA23 PR	PA22 PR	PA21 PR	PA20 PR	PA19 PR	PA18 PR	PA17 PR	PA16 PR
初始值:	0	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15、14	—	全 0	R	保留位 读写的值总是 0。
13	PA29PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
12	PA28PR	引脚的状态	R	
11	PA27PR	引脚的状态	R	
10	PA26PR	引脚的状态	R	
9	PA25PR	引脚的状态	R	
8	PA24PR	引脚的状态	R	
7	PA23PR	引脚的状态	R	
6	PA22PR	引脚的状态	R	
5	PA21PR	引脚的状态	R	
4	PA20PR	引脚的状态	R	
3	PA19PR	引脚的状态	R	
2	PA18PR	引脚的状态	R	
1	PA17PR	引脚的状态	R	
0	PA16PR	引脚的状态	R	

## • PAPRL (SH7083)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15 PR	PA14 PR	PA13 PR	PA12 PR	-	PA10 PR	PA9 PR	PA8 PR	PA7 PR	-	PA5 PR	PA4 PR	PA3 PR	-	-	-
初始值:	*	*	*	*	0	*	*	*	*	0	*	*	*	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15	PA15PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
14	PA14PR	引脚的状态	R	
13	PA13PR	引脚的状态	R	
12	PA12PR	引脚的状态	R	
11	—	0	R	保留位 读写的值总是 0。
10	PA10PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
9	PA9PR	引脚的状态	R	
8	PA8PR	引脚的状态	R	
7	PA7PR	引脚的状态	R	
6	—	0	R	保留位 读写的值总是 0。
5	PA5PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
4	PA4PR	引脚的状态	R	
3	PA3PR	引脚的状态	R	
2~0	—	全 0	R	保留位 读写的值总是 0。

- PAPRL (SH7084/85/86)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15 PR	PA14 PR	PA13 PR	PA12 PR	PA11 PR	PA10 PR	PA9 PR	PA8 PR	PA7 PR	PA6 PR	PA5 PR	PA4 PR	PA3 PR	PA2 PR	PA1 PR	PA0 PR
初始值:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15	PA15PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
14	PA14PR	引脚的状态	R	
13	PA13PR	引脚的状态	R	
12	PA12PR	引脚的状态	R	
11	PA11PR	引脚的状态	R	
10	PA10PR	引脚的状态	R	
9	PA9PR	引脚的状态	R	
8	PA8PR	引脚的状态	R	
7	PA7PR	引脚的状态	R	
6	PA6PR	引脚的状态	R	
5	PA5PR	引脚的状态	R	
4	PA4PR	引脚的状态	R	
3	PA3PR	引脚的状态	R	
2	PA2PR	引脚的状态	R	
1	PA1PR	引脚的状态	R	
0	PA0PR	引脚的状态	R	



## 22.2 端口 B

如图 22.5 所示，SH7083 的端口 B 是有 9 个引脚的输入/输出端口。

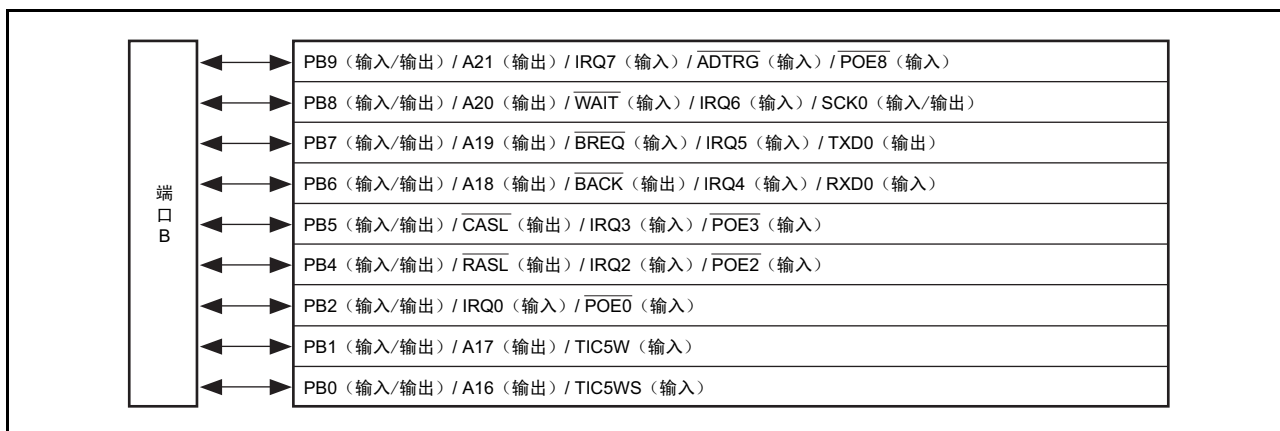


图 22.5 端口 B (SH7083)

如图 22.6 所示，SH7084/85/86 的端口 B 是有 10 个引脚的输入/输出端口。

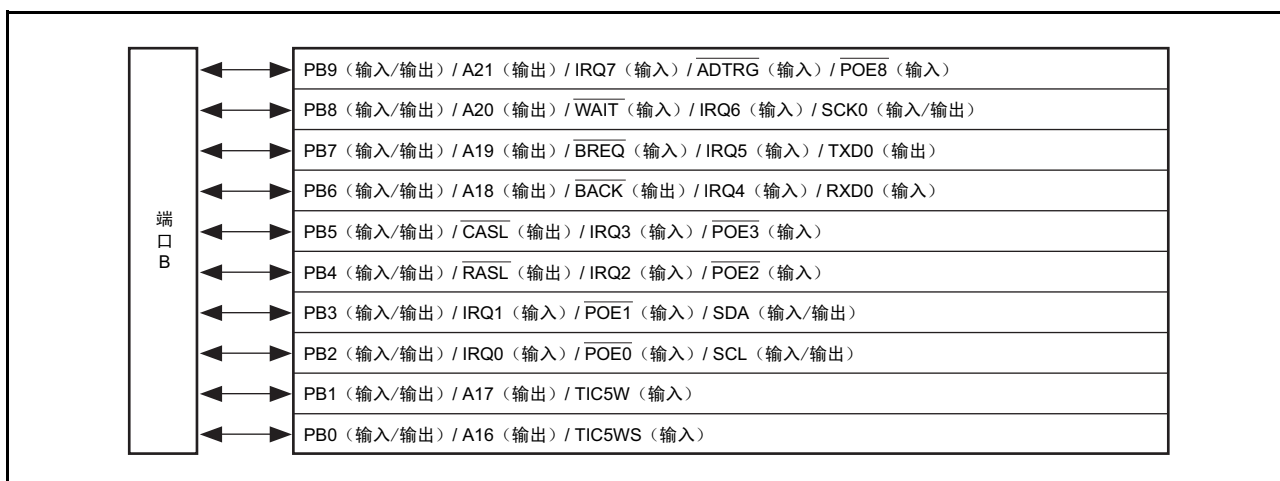


图 22.6 端口 B (SH7084/85/86)

## 22.2.1 寄存器说明

SH7083 的端口 B 是 9 位输入 / 输出端口，SH7084/85/86 的端口 B 是 10 位输入 / 输出端口。端口 B 有以下寄存器，有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 27 章 寄存器一览”。

表 22.3 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
端口 B 的数据寄存器 L	PBDRL	R/W	H'0000	H'FFFFD182	8、16
端口 B 的端口寄存器 L	PBPRL	R	H'0xxx	H'FFFFD19E	8、16

## 22.2.2 端口 B 的数据寄存器 L (PBDRL)

PBDRL 是 16 位可读写寄存器，保存端口 B 的数据。SH7083 的 PB9DR ~ PB4DR 位、PB2DR ~ PB0DR 位分别对应 PB9 ~ PB4 引脚、PB2 ~ PB0 引脚（省略有关兼用功能的记述）；SH7084/85/86 的 PB9DR ~ PB0DR 位分别对应 PB9 ~ PB0 引脚（省略有关兼用功能的记述）。

在引脚功能为通用输出时，如果给 PBDRL 写值，就从引脚输出该值；如果读 PBDRL，就直接读寄存器的值，与引脚的状态无关。

在引脚功能为通用输入能时，如果读 PBDRL，就直接读引脚的状态而不读寄存器的值；如果给 PBDRL 写值，就能将值写到 PBDRL，但是不影响引脚的状态。端口 B 的数据寄存器的读写操作如表 22.4 所示。

### • PBDRL (SH7083)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PB9 DR	PB8 DR	PB7 DR	PB6 DR	PB5 DR	PB4 DR	-	PB2 DR	PB1 DR	PB0 DR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 10	—	全 0	R	保留位 读写的值总是 0。
9	PB9DR	0	R/W	参照表 22.4
8	PB8DR	0	R/W	
7	PB7DR	0	R/W	
6	PB6DR	0	R/W	
5	PB5DR	0	R/W	
4	PB4DR	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PB2DR	0	R/W	参照表 22.4
1	PB1DR	0	R/W	
0	PB0DR	0	R/W	

- PBDRL (SH7084/85/86)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PB9 DR	PB8 DR	PB7 DR	PB6 DR	PB5 DR	PB4 DR	PB3 DR	PB2 DR	PB1 DR	PB0 DR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 10	—	全 0	R	保留位 读写的值总是 0。
9	PB9DR	0	R/W	参照表 22.4
8	PB8DR	0	R/W	
7	PB7DR	0	R/W	
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	

表 22.4 端口 B 的数据寄存器 L (PBDRL) 的读写操作

- PBDRL 的 bit9 ~ 0

PBIOR	引脚功能	读	写
0	通用输入	引脚的状态	能写 PBDRL, 但是不影响引脚的状态
	非通用输入	引脚的状态	能写 PBDRL, 但是不影响引脚的状态
1	通用输出	PBDRL 的值	从引脚输出所写的值
	非通用输出	PBDRL 的值	能写 PBDRL, 但是不影响引脚的状态

### 22.2.3 端口 B 的端口寄存器 L (PBPR)

PBPR 是 16 位只读寄存器，能随时读引脚的状态，与 PFC 的设定无关。SH7083 的 PB9PR ~ PB4PR 位、PB2PR ~ PB0PR 位分别对应 PB9 ~ PB4 引脚、PB2 ~ PB0 引脚（省略有关兼用功能的记述）；SH7084/85/86 的 PB9PR ~ PB0PR 位分别对应 PB9 ~ PB0 引脚（省略有关兼用功能的记述）。

- PBPR (SH7083)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PB9 PR	PB8 PR	PB7 PR	PB6 PR	PB5 PR	PB4 PR	-	PB2 PR	PB1 PR	PB0 PR
初始值:	0	0	0	0	0	0	*	*	*	*	*	*	0	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 10	—	全 0	R	保留位 读写的值总是 0。
9	PB9PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
8	PB8PR	引脚的状态	R	
7	PB7PR	引脚的状态	R	
6	PB6PR	引脚的状态	R	
5	PB5PR	引脚的状态	R	
4	PB4PR	引脚的状态	R	
3	—	0	R	保留位 读写的值总是 0。
2	PB2PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
1	PB1PR	引脚的状态	R	
0	PB0PR	引脚的状态	R	

## • PBPRL (SH7084/85/86)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PB9 PR	PB8 PR	PB7 PR	PB6 PR	PB5 PR	PB4 PR	PB3 PR	PB2 PR	PB1 PR	PB0 PR
初始值:	0	0	0	0	0	0	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 10	—	全 0	R	保留位 读写的值总是 0。
9	PB9PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
8	PB8PR	引脚的状态	R	
7	PB7PR	引脚的状态	R	
6	PB6PR	引脚的状态	R	
5	PB5PR	引脚的状态	R	
4	PB4PR	引脚的状态	R	
3	PB3PR	引脚的状态	R	
2	PB2PR	引脚的状态	R	
1	PB1PR	引脚的状态	R	
0	PB0PR	引脚的状态	R	

## 22.3 端口 C

如图 22.7 所示，SH7083/84/85 的端口 C 是有 16 个引脚的输入 / 输出端口。

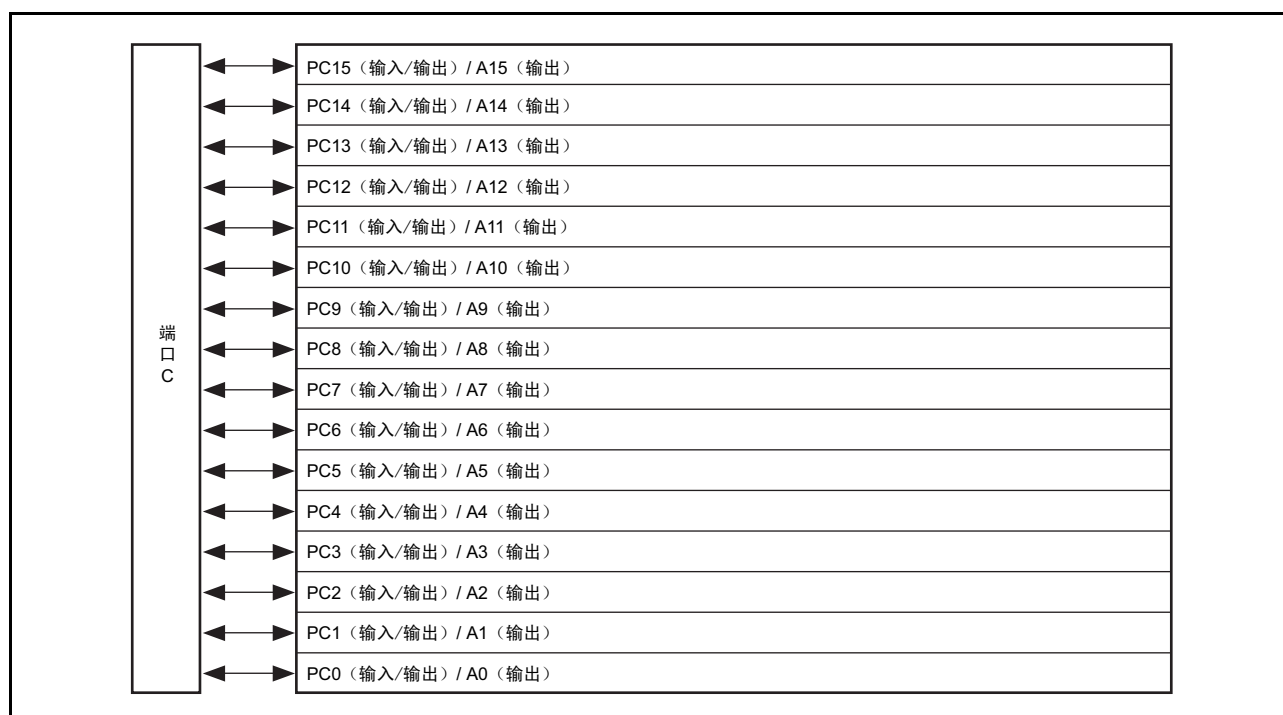


图 22.7 端口 C (SH7083/84/85)

如图 22.8 所示，SH7086 的端口 C 是有 24 个引脚的输入 / 输出端口。

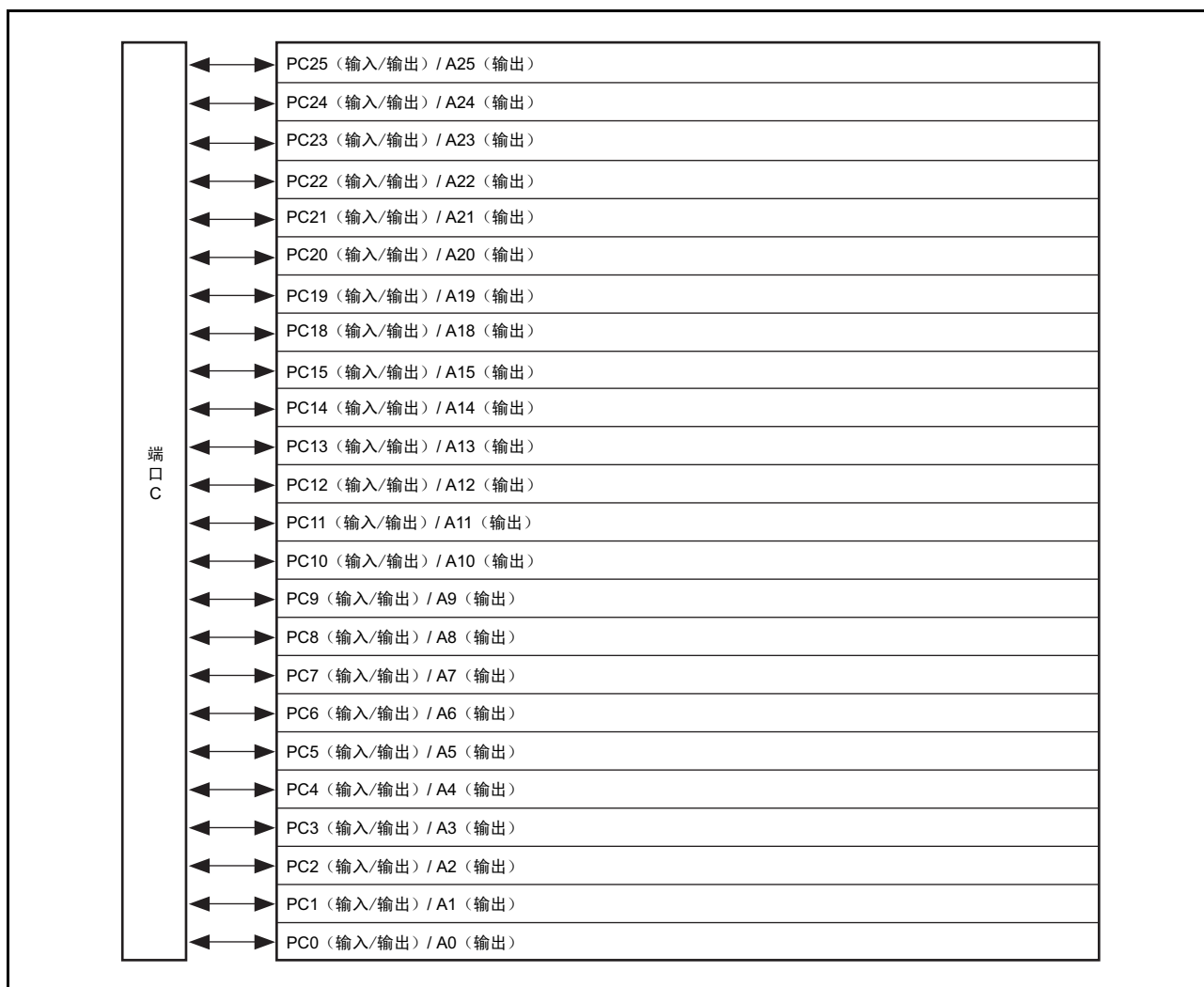


图 22.8 端口 C (SH7086)

### 22.3.1 寄存器说明

SH7083/84/85 的端口 C 是 16 位输入 / 输出端口，SH7086 的端口 C 是 24 位输入 / 输出端口。端口 C 有以下寄存器，有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 27 章 寄存器一览”。

表 22.5 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
端口 C 的数据寄存器 H	PCDRH	R/W	H'0000	H'FFFFD200	8、16、32
端口 C 的数据寄存器 L	PCDRL	R/W	H'0000	H'FFFFD202	8、16
端口 C 的端口寄存器 H	PCPRH	R	H'xxxx	H'FFFFD21C	8、16、32
端口 C 的端口寄存器 L	PCPRL	R	H'xxxx	H'FFFFD21E	8、16

### 22.3.2 端口 C 的数据寄存器 H、L (PCDRH、PCDRL)

PCDRH 和 PCDRL 是 16 位可读写寄存器，保存端口 C 的数据。SH7083/84/85 的 PC15DR ~ PC0DR 位分别对应 PC15 ~ PC0 引脚（省略有关兼用功能的记述）；SH7086 的 PC25DR ~ PC18DR 位、PC15DR ~ PC0DR 位分别对应 PC25 ~ PC18、PC15 ~ PC0 引脚（省略有关兼用功能的记述）。

在引脚功能为通用输出时，如果给 PCDRH 或者 PCDRL 写值，就从引脚输出该值；如果读 PCDRH 或者 PCDRL，就直接读寄存器的值，与引脚的状态无关。

在引脚功能为通用输入时，如果读 PCDRH 或者 PCDRL，就直接读引脚的状态而不读寄存器的值；如果给 PCDRH 或者 PCDRL 写值，就能将值写到 PCDRH 或者 PCDRL，但是不影响引脚的状态。端口 C 的数据寄存器的读写操作如表 22.6 所示。

#### • PCDRH (SH7083/84/85)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 0	—	全 0	R	保留位 读写的值总是 0。

## • PCDRH (SH7086)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PC25 DR	PC24 DR	PC23 DR	PC22 DR	PC21 DR	PC20 DR	PC19 DR	PC18 DR	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

位	位名	初始值	R/W	说 明
15 ~ 10	—	全 0	R	保留位 读写的值总是 0。
9	PC25DR	0	R/W	参照表 22.6
8	PC24DR	0	R/W	
7	PC23DR	0	R/W	
6	PC22DR	0	R/W	
5	PC21DR	0	R/W	
4	PC20DR	0	R/W	
3	PC19DR	0	R/W	
2	PC18DR	0	R/W	
1、0	—	全 0	R	保留位 读写的值总是 0。



## • PCDRL

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC15 DR	PC14 DR	PC13 DR	PC12 DR	PC11 DR	PC10 DR	PC9 DR	PC8 DR	PC7 DR	PC6 DR	PC5 DR	PC4 DR	PC3 DR	PC2 DR	PC1 DR	PC0 DR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	PC15DR	0	R/W	参照表 22.6
14	PC14DR	0	R/W	
13	PC13DR	0	R/W	
12	PC12DR	0	R/W	
11	PC11DR	0	R/W	
10	PC10DR	0	R/W	
9	PC9DR	0	R/W	
8	PC8DR	0	R/W	
7	PC7DR	0	R/W	
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	

表 22.6 端口 C 的数据寄存器 (PCDR) 的读写操作

- PCDRH 的 bit9 ~ 2 和 PCDRL 的 bit15 ~ 0

PCIOR	引脚功能	读	写
0	通用输入	引脚的状态	能写 PCDRH、L，但是不影响引脚的状态
	非通用输入	引脚的状态	能写 PCDRH、L，但是不影响引脚的状态
1	通用输出	PCDRH、L 的值	从引脚输出所写的值
	非通用输出	PCDRH、L 的值	能写 PCDRH、L，但是不影响引脚的状态

### 22.3.3 端口 C 的端口寄存器 H、L (PCPRH、PCPRL)

PCPRH 和 PCPRL 是 16 位只读寄存器，能随时读引脚的状态，与 PFC 的设定无关。SH7083/84/85 的 PC15PR ~ PC0PR 位分别对应 PC15 ~ PC0 引脚（省略引脚名端口以外的多路复用引脚名）；SH7086 的 PC25PR ~ PC18PR 位、PC15PR ~ PC0PR 位分别对应 PC25 ~ PC18、PC15 ~ PC0 引脚（省略引脚名端口以外的多路复用引脚名）。

#### • PCPRH (SH7083/84/85)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15 ~ 0	—	全 0	R	保留位 读写的值总是 0。

#### • PCPRH (SH7086)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PC25 PR	PC24 PR	PC23 PR	PC22 PR	PC21 PR	PC20 PR	PC19 PR	PC18 PR	-	-
初始值:	0	0	0	0	0	0	*	*	*	*	*	*	*	*	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15 ~ 10	—	全 0	R	保留位 读写的值总是 0。
9	PC25PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
8	PC24PR	引脚的状态	R	
7	PC23PR	引脚的状态	R	
6	PC22PR	引脚的状态	R	
5	PC21PR	引脚的状态	R	
4	PC20PR	引脚的状态	R	
3	PC19PR	引脚的状态	R	
2	PC18PR	引脚的状态	R	
1、0	—	全 0	R	保留位 读写的值总是 0。

## • PCPRL

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PC15 PR	PC14 PR	PC13 PR	PC12 PR	PC11 PR	PC10 PR	PC9 PR	PC8 PR	PC7 PR	PC6 PR	PC5 PR	PC4 PR	PC3 PR	PC2 PR	PC1 PR	PC0 PR
初始值:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15	PC15PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
14	PC14PR	引脚的状态	R	
13	PC13PR	引脚的状态	R	
12	PC12PR	引脚的状态	R	
11	PC11PR	引脚的状态	R	
10	PC10PR	引脚的状态	R	
9	PC9PR	引脚的状态	R	
8	PC8PR	引脚的状态	R	
7	PC7PR	引脚的状态	R	
6	PC6PR	引脚的状态	R	
5	PC5PR	引脚的状态	R	
4	PC4PR	引脚的状态	R	
3	PC3PR	引脚的状态	R	
2	PC2PR	引脚的状态	R	
1	PC1PR	引脚的状态	R	
0	PC0PR	引脚的状态	R	

## 22.4 端口 D

如图 22.9 所示，SH7083/84 的端口 D 是有 16 个引脚的输入 / 输出端口。

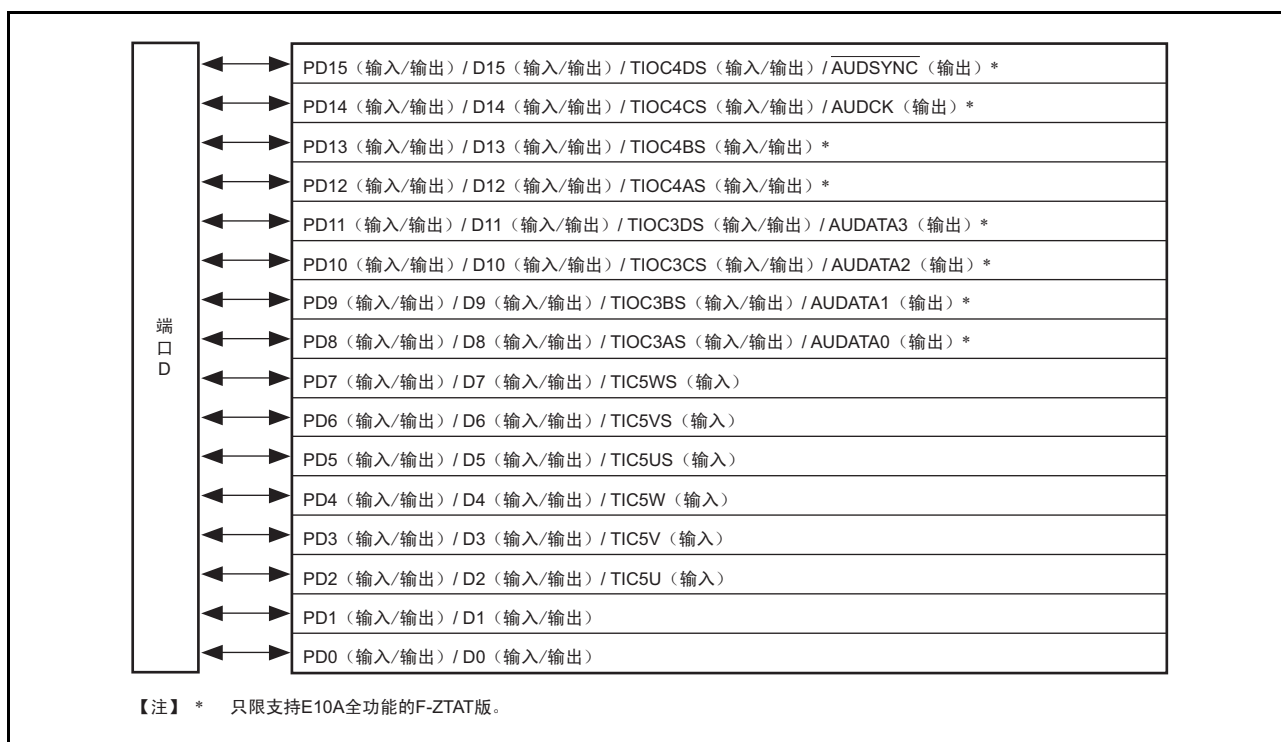


图 22.9 端口 D (SH7083/84)

如图 22.10 所示，SH7085/86 的端口 D 是有 32 个引脚的输入 / 输出端口。

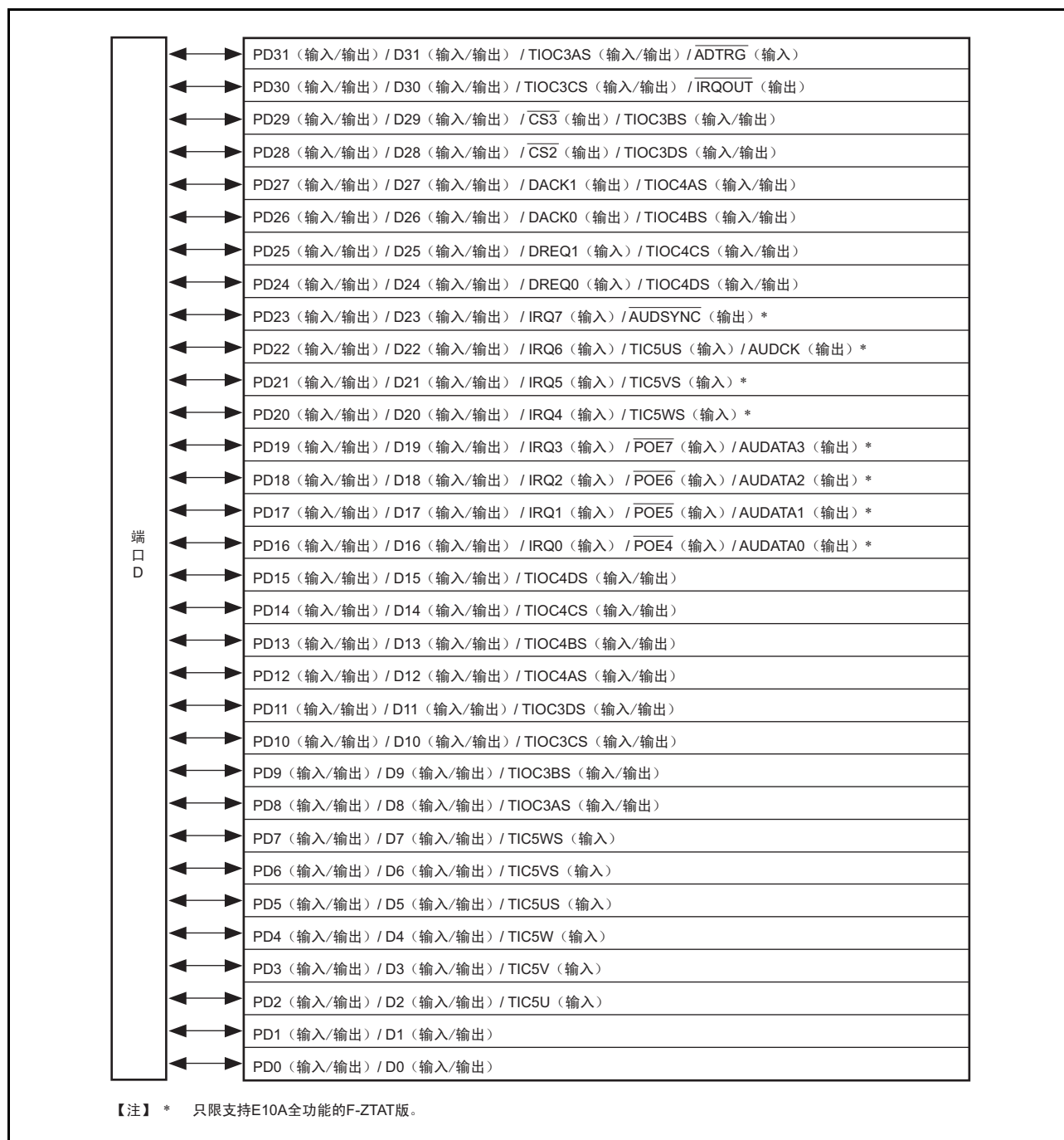


图 22.10 端口 D (SH7085/86)

### 22.4.1 寄存器说明

SH7083/84 的端口 D 是 16 位输入 / 输出端口，SH7085/86 的端口 D 是 32 位输入 / 输出端口。端口 D 有以下寄存器，有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 27 章 寄存器一览”。

表 22.7 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
端口 D 的数据寄存器 H	PDDRH	R/W	H'0000	H'FFFFD280	8、16、32
端口 D 的数据寄存器 L	PDDRL	R/W	H'0000	H'FFFFD282	8、16
端口 D 的端口寄存器 H	PDPRH	R	H'xxxx	H'FFFFD29C	8、16、32
端口 D 的端口寄存器 L	PDPRL	R	H'xxxx	H'FFFFD29E	8、16

### 22.4.2 端口 D 的数据寄存器 H、L (PDDRH、PDDRL)

PDDRH 和 PDDRL 是 16 位可读写寄存器，保存端口 D 的数据。SH7083/84 的 PD15DR ~ PD0DR 位分别对应 PD15 ~ PD0 引脚（省略有关兼用功能的记述）；SH7085/86 的 PD31DR ~ PD0DR 位分别对应 PD31 ~ PD0 引脚（省略有关兼用功能的记述）。

在引脚功能为通用输出时，如果给 PDDRH 或者 PDDRL 写值，就从引脚输出该值；如果读 PDDRH 或者 PDDRL，就直接读寄存器的值，与引脚的状态无关。

在引脚功能为通用输入时，如果读 PDDRH 或者 PDDRL，就直接读引脚的状态而不读寄存器的值；如果给 PDDRH 或者 PDDRL 写值，就能将值写到 PDDRH 或者 PDDRL，但是不影响引脚的状态。端口 D 的数据寄存器 L 的读写操作如表 22.8 所示。

#### • PDDRH (SH7083/84)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 0	—	全 0	R	保留位 读写的值总是 0。

## • PDDRH (SH7085/86)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD31 DR	PD30 DR	PD29 DR	PD28 DR	PD27 DR	PD26 DR	PD25 DR	PD24 DR	PD23 DR	PD22 DR	PD21 DR	PD20 DR	PD19 DR	PD18 DR	PD17 DR	PD16 DR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	PD31DR	0	R/W	参照表 22.8
14	PD30DR	0	R/W	
13	PD29DR	0	R/W	
12	PD28DR	0	R/W	
11	PD27DR	0	R/W	
10	PD26DR	0	R/W	
9	PD25DR	0	R/W	
8	PD24DR	0	R/W	
7	PD23DR	0	R/W	
6	PD22DR	0	R/W	
5	PD21DR	0	R/W	
4	PD20DR	0	R/W	
3	PD19DR	0	R/W	
2	PD18DR	0	R/W	
1	PD17DR	0	R/W	
0	PD16DR	0	R/W	

## • PDDRL

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD15 DR	PD14 DR	PD13 DR	PD12 DR	PD11 DR	PD10 DR	PD9 DR	PD8 DR	PD7 DR	PD6 DR	PD5 DR	PD4 DR	PD3 DR	PD2 DR	PD1 DR	PD0 DR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	PD15DR	0	R/W	参照表 22.8
14	PD14DR	0	R/W	
13	PD13DR	0	R/W	
12	PD12DR	0	R/W	
11	PD11DR	0	R/W	
10	PD10DR	0	R/W	
9	PD9DR	0	R/W	
8	PD8DR	0	R/W	
7	PD7DR	0	R/W	
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	

表 22.8 端口 D 的数据寄存器 (PDDR) 的读写操作

## • PDDRH 和 bit15 ~ 0 和 PDDRL 的 bit15 ~ 0

PDIOR	引脚功能	读	写
0	通用输入	引脚的状态	能写 PDDRH、L，但是不影响引脚的状态
	非通用输入	引脚的状态	能读 PDDRH、L，但是不影响引脚的状态
1	通用输出	PDDRH、L 的值	从引脚输出所写的值
	非通用输出	PDDRH、L 的值	能 PDDRH、L，但是不影响引脚的状态



### 22.4.3 端口 D 的端口寄存器 H、L (PDPRH、PDPRL)

PDPRH 和 PDPRL 是 16 位只读寄存器，能随时读引脚的状态，与 PFC 的设定无关。SH7083/84 的 PD15PR ~ PD0PR 位分别对应 PD15 ~ PD0 引脚（省略有关兼用功能的记述）；SH7085/86 的 PD31PR ~ PD0PR 位分别对应 PD31 ~ PD0 引脚（省略有关兼用功能的记述）。

#### • PDPRH (SH7083/84)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15 ~ 0	—	全 0	R	保留位 读写的值总是 0。

#### • PDPRH (SH7085/86)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD31 PR	PD30 PR	PD29 PR	PD28 PR	PD27 PR	PD26 PR	PD25 PR	PD24 PR	PD23 PR	PD22 PR	PD21 PR	PD20 PR	PD19 PR	PD18 PR	PD17 PR	PD16 PR
初始值:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15	PD31PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
14	PD30PR	引脚的状态	R	
13	PD29PR	引脚的状态	R	
12	PD28PR	引脚的状态	R	
11	PD27PR	引脚的状态	R	
10	PD26PR	引脚的状态	R	
9	PD25PR	引脚的状态	R	
8	PD24PR	引脚的状态	R	
7	PD23PR	引脚的状态	R	
6	PD22PR	引脚的状态	R	
5	PD21PR	引脚的状态	R	
4	PD20PR	引脚的状态	R	
3	PD19PR	引脚的状态	R	
2	PD18PR	引脚的状态	R	
1	PD17PR	引脚的状态	R	
0	PD16PR	引脚的状态	R	

## • PDPRL

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD15 PR	PD14 PR	PD13 PR	PD12 PR	PD11 PR	PD10 PR	PD9 PR	PD8 PR	PD7 PR	PD6 PR	PD5 PR	PD4 PR	PD3 PR	PD2 PR	PD1 PR	PD0 PR
初始值:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15	PD15PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
14	PD14PR	引脚的状态	R	
13	PD13PR	引脚的状态	R	
12	PD12PR	引脚的状态	R	
11	PD11PR	引脚的状态	R	
10	PD10PR	引脚的状态	R	
9	PD9PR	引脚的状态	R	
8	PD8PR	引脚的状态	R	
7	PD7PR	引脚的状态	R	
6	PD6PR	引脚的状态	R	
5	PD5PR	引脚的状态	R	
4	PD4PR	引脚的状态	R	
3	PD3PR	引脚的状态	R	
2	PD2PR	引脚的状态	R	
1	PD1PR	引脚的状态	R	
0	PD0PR	引脚的状态	R	

## 22.5 端口 E

如图 22.11 所示，SH7083 的端口 E 是有 13 个引脚的输入 / 输出端口。

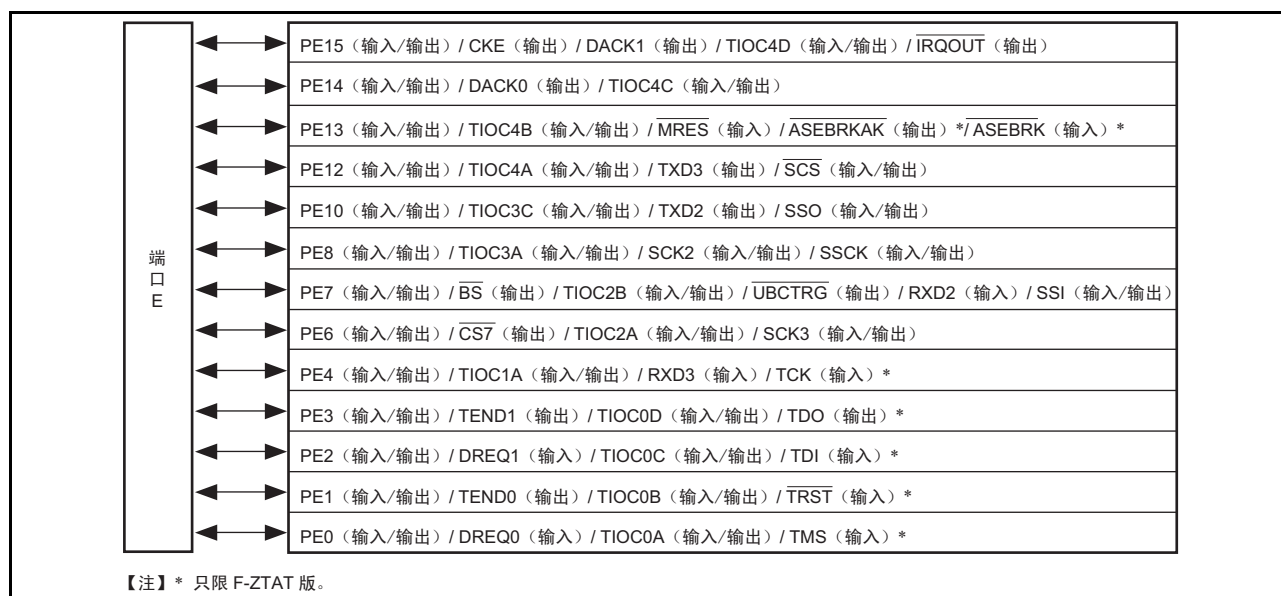


图 22.11 端口 E (SH7083)

如图 22.12 所示，SH7084 的端口 E 是有 16 个引脚的输入 / 输出端口。

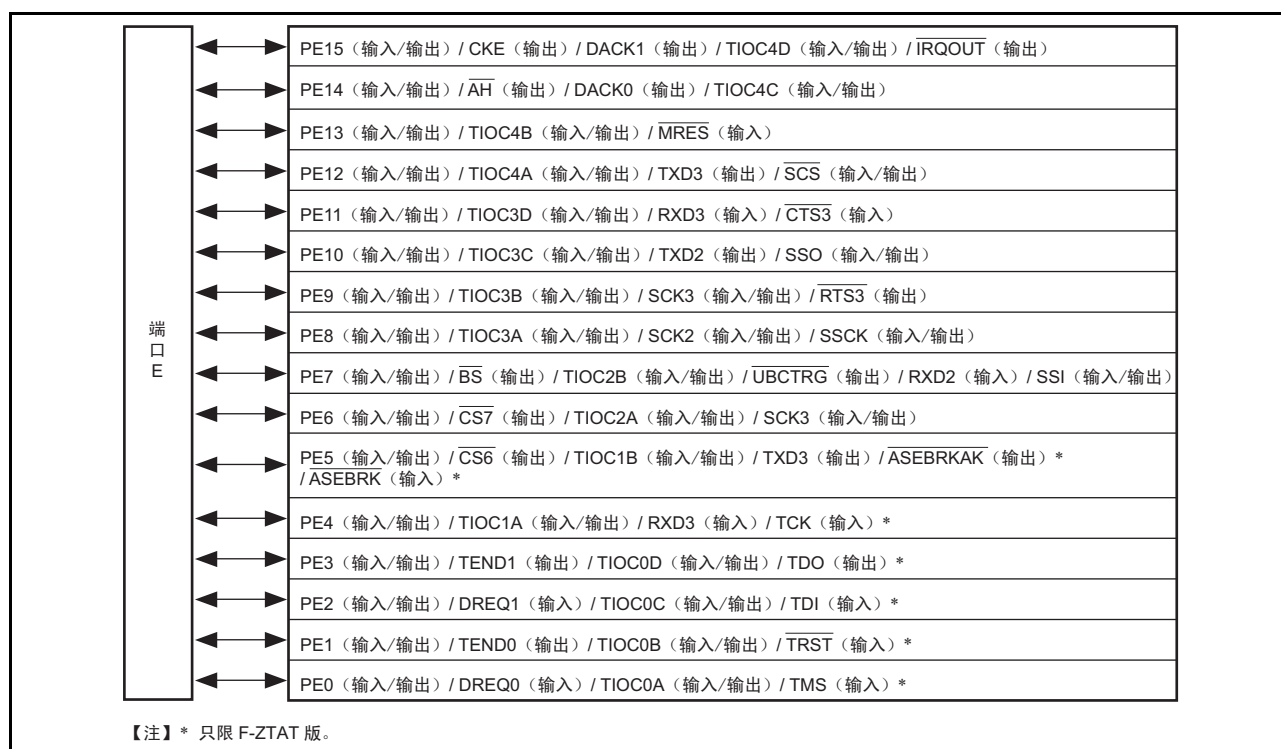


图 22.12 端口 E (SH7084)

如图 22.13 所示，SH7085 的端口 E 是有 16 个引脚的输入 / 输出端口。

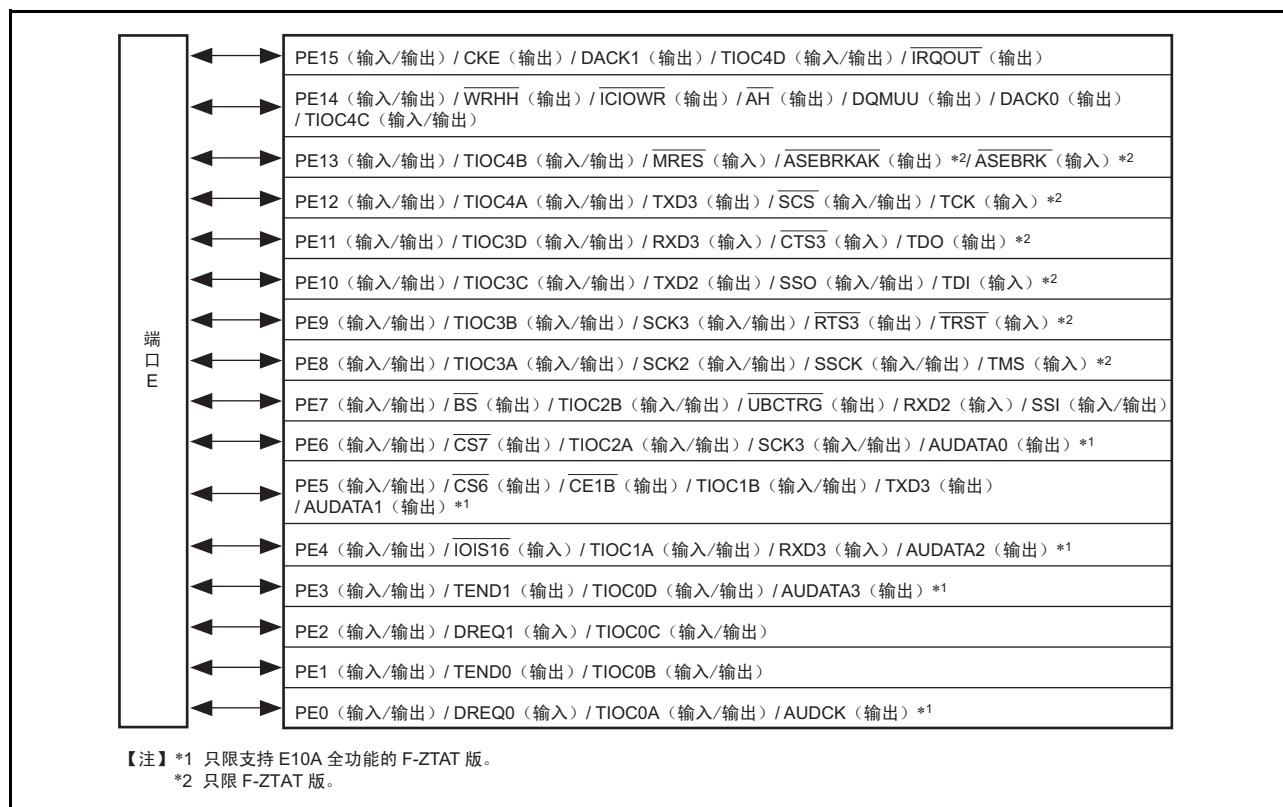


图 22.13 端口 E (SH7085)

如图 22.14 所示，SH7086 的端口 E 是有 22 个引脚的输入 / 输出端口。

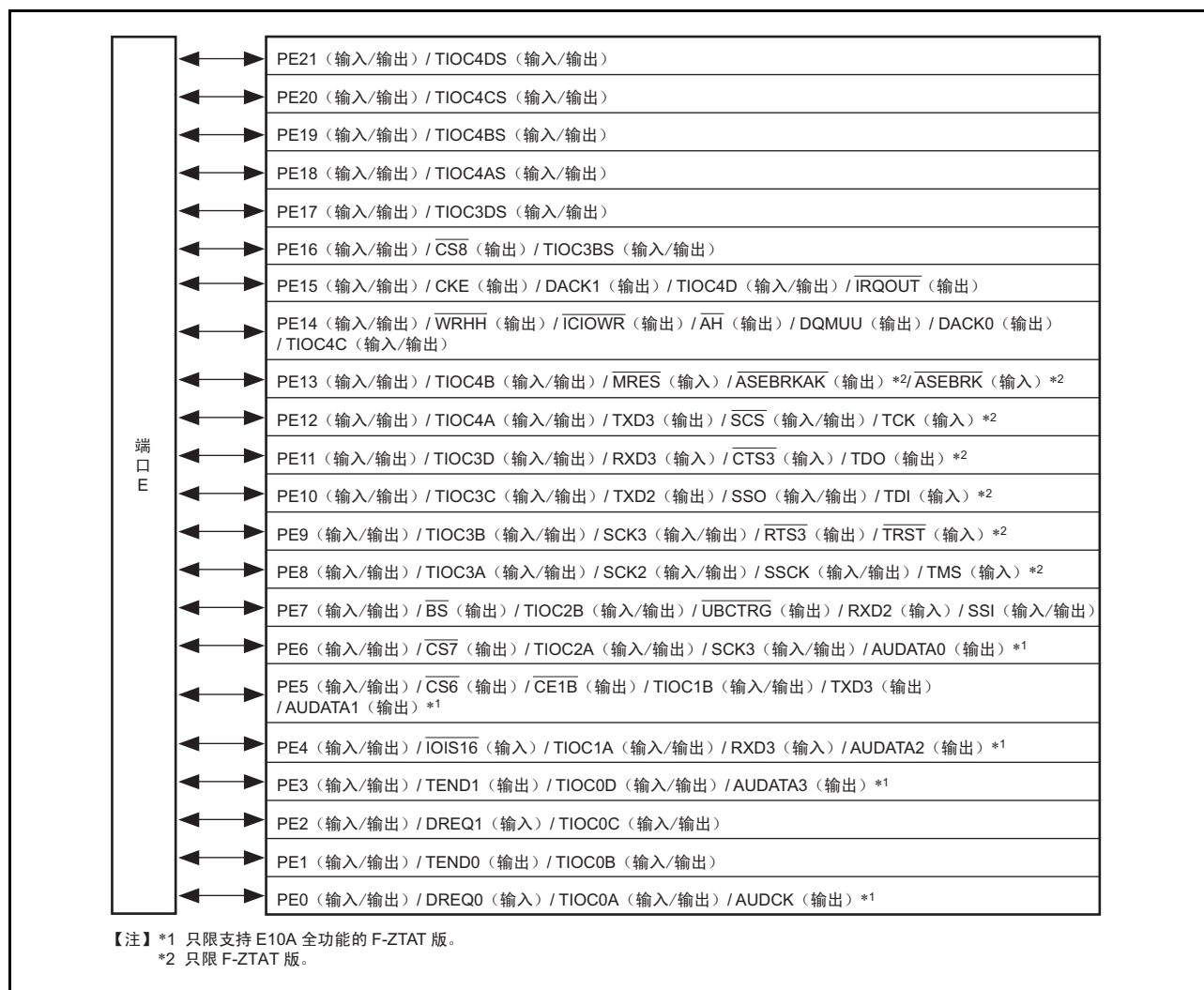


图 22.14 端口 E (SH7086)

### 22.5.1 寄存器说明

SH7083 的端口 E 是 13 位输入 / 输出端口，SH7084/85 的端口 E 是 16 位输入 / 输出端口，SH7086 的端口 E 是 22 位输入 / 输出端口。端口 E 有以下寄存器，有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 27 章 寄存器一览”。

表 22.9 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
端口 E 的数据寄存器 H	PEDRH	R/W	H'0000	H'FFFFD300	8、16、32
端口 E 的数据寄存器 L	PEDRL	R/W	H'0000	H'FFFFD302	8、16
端口 E 的端口寄存器 H	PEPRH	R	H'xxxx	H'FFFFD31C	8、16、32
端口 E 的端口寄存器 L	PEPRL	R	H'xxxx	H'FFFFD31E	8、16

### 22.5.2 端口 E 的数据寄存器 H、L (PEDRH、PEDRL)

PEDRH 和 PEDRL 是 16 位可读写寄存器，保存端口 E 的数据。SH7083 的 PE15DR ~ PE12DR 位、PE10DR 位、PE8DR ~ PE6DR 位、PE4DR ~ PE0DR 位分别对应 PE15 ~ PE12 引脚、PE10 引脚、PE8 ~ PE6 引脚、PE4 ~ PE0 引脚（省略有关兼用功能的记述）；SH7084/85 的 PE15DR ~ PE0DR 位分别对应 PE15 ~ PE0 引脚（省略有关兼用功能的记述）；SH7086 的 PE21DR ~ PE0DR 位分别对应 PE21 ~ PE0 引脚（省略有关兼用功能的记述）。

在引脚功能为通用输出时，如果给 PEDRH 或者 PEDRL 写值，就从引脚输出该值；如果读 PEDRH 或者 PEDRL，就直接读寄存器的值，与引脚的状态无关。

在引脚功能为通用输入时，如果读 PEDRH 或者 PEDRL，就直接读引脚的状态而不读寄存器的值；如果给 PEDRH 或者 PEDRL 写值，就能将值写到 PEDRH 或者 PEDRL，但是不影响引脚的状态。端口 E 的数据寄存器的读写操作如表 22.10 所示。

- PEDRH (SH7083/84/85)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 0	—	全 0	R	保留位 读写的值总是 0。

## • PEDRH (SH7086)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PE21 DR	PE20 DR	PE19 DR	PE18 DR	PE17 DR	PE16 DR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 6	—	全 0	R	保留位 读写的值总是 0。
5	PE21DR	0	R/W	参照表 22.10
4	PE20DR	0	R/W	
3	PE19DR	0	R/W	
2	PE18DR	0	R/W	
1	PE17DR	0	R/W	
0	PE16DR	0	R/W	

## • PEDRL (SH7083)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15 DR	PE14 DR	PE13 DR	PE12 DR	-	PE10 DR	-	PE8 DR	PE7 DR	PE6 DR	-	PE4 DR	PE3 DR	PE2 DR	PE1 DR	PE0 DR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	PE15DR	0	R/W	参照表 22.10
14	PE14DR	0	R/W	
13	PE13DR	0	R/W	
12	PE12DR	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PE10DR	0	R/W	参照表 22.10
9	—	0	R	保留位 读写的值总是 0。
8	PE8DR	0	R/W	参照表 22.10
7	PE7DR	0	R/W	
6	PE6DR	0	R/W	
5	—	0	R	保留位 读写的值总是 0。
4	PE4DR	0	R/W	参照表 22.10
3	PE3DR	0	R/W	
2	PE2DR	0	R/W	
1	PE1DR	0	R/W	
0	PE0DR	0	R/W	



- PEDRL (SH7084/85/86)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15 DR	PE14 DR	PE13 DR	PE12 DR	PE11 DR	PE10 DR	PE9 DR	PE8 DR	PE7 DR	PE6 DR	PE5 DR	PE4 DR	PE3 DR	PE2 DR	PE1 DR	PE0 DR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	PE15DR	0	R/W	参照表 22.10
14	PE14DR	0	R/W	
13	PE13DR	0	R/W	
12	PE12DR	0	R/W	
11	PE11DR	0	R/W	
10	PE10DR	0	R/W	
9	PE9DR	0	R/W	
8	PE8DR	0	R/W	
7	PE7DR	0	R/W	
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	

表 22.10 端口 E 的数据寄存器 (PEDR) 的读写操作

- PEDRH 的 bit5 ~ 0 和 PEDRL 的 bit15 ~ 0

PEIOR	引脚功能	读	写
0	通用输入	引脚的状态	能写 PEDRH、L，但是不影响引脚的状态
	非通用输入	引脚的状态	能写 PEDRH、L，但是不影响引脚的状态
1	通用输出	PEDRH、L 的值	从引脚输出所写的值
	非通用输出	PEDRH、L 的值	能写 PEDRH、L，但是不影响引脚的状态

### 22.5.3 端口 E 的端口寄存器 H、L (PEPRH、PEPRL)

PEPRH 和 PEPRL 是 16 位只读寄存器，能随时读引脚的状态，与 PFC 的设定无关。SH7083 的 PE15PR ~ PE12PR 位、PE10PR 位、PE8PR ~ PE6PR 位、PE4PR ~ PE0PR 位分别对应 PE15 ~ PE12 引脚、PE10 引脚、PE8 ~ PE6 引脚、PE4 ~ PE0 引脚（省略有关兼用功能的记述）；SH7084/85 的 PE15PR ~ PE0PR 位分别对应 PE15 ~ PE0 引脚（省略有关兼用功能的记述）；SH7086 的 PE21PR ~ PE0PR 位分别对应 PE21 ~ PE0 引脚（省略有关兼用功能的记述）。

#### • PEPRH (SH7083/84/85)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 0	—	全 0	R	保留位 读写的值总是 0。

#### • PEPRH (SH7086)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PE21 PR	PE20 PR	PE19 PR	PE18 PR	PE17 PR	PE16 PR
初始值:	0	0	0	0	0	0	0	0	0	0	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 6	—	全 0	R	保留位 读写的值总是 0。
5	PE21PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
4	PE20PR	引脚的状态	R	
3	PE19PR	引脚的状态	R	
2	PE18PR	引脚的状态	R	
1	PE17PR	引脚的状态	R	
0	PE16PR	引脚的状态	R	

## • PEPRL (SH7083)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15 PR	PE14 PR	PE13 PR	PE12 PR	-	PE10 PR	-	PE8 PR	PE7 PR	PE6 PR	-	PE4 PR	PE3 PR	PE2 PR	PE1 PR	PE0 PR
初始值:	*	*	*	*	0	*	0	*	*	*	0	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15	PE15PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
14	PE14PR	引脚的状态	R	
13	PE13PR	引脚的状态	R	
12	PE12PR	引脚的状态	R	
11	—	0	R	保留位 读写的值总是 0。
10	PE10PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
9	—	0	R	保留位 读写的值总是 0。
8	PE8PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
7	PE7PR	引脚的状态	R	
6	PE6PR	引脚的状态	R	
5	—	0	R	保留位 读写的值总是 0。
4	PE4PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
3	PE3PR	引脚的状态	R	
2	PE2PR	引脚的状态	R	
1	PE1PR	引脚的状态	R	
0	PE0PR	引脚的状态	R	

## • PEPRL (SH7084/85/86)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15 PR	PE14 PR	PE13 PR	PE12 PR	PE11 PR	PE10 PR	PE9 PR	PE8 PR	PE7 PR	PE6 PR	PE5 PR	PE4 PR	PE3 PR	PE2 PR	PE1 PR	PE0 PR
初始值:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15	PE15PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
14	PE14PR	引脚的状态	R	
13	PE13PR	引脚的状态	R	
12	PE12PR	引脚的状态	R	
11	PE11PR	引脚的状态	R	
10	PE10PR	引脚的状态	R	
9	PE9PR	引脚的状态	R	
8	PE8PR	引脚的状态	R	
7	PE7PR	引脚的状态	R	
6	PE6PR	引脚的状态	R	
5	PE5PR	引脚的状态	R	
4	PE4PR	引脚的状态	R	
3	PE3PR	引脚的状态	R	
2	PE2PR	引脚的状态	R	
1	PE1PR	引脚的状态	R	
0	PE0PR	引脚的状态	R	

## 22.6 端口 F

如图 22.15 所示，SH7083/84/85 的端口 F 是有 8 个引脚的输入专用端口。

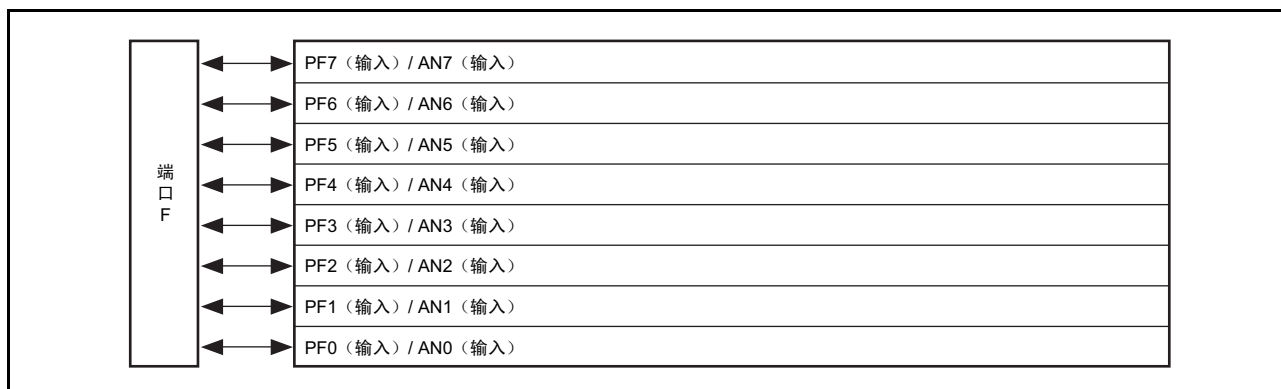


图 22.15 端口 F (SH7083/84/85)

如图 22.16 所示，SH7086 的端口 F 是有 16 个引脚的输入专用端口。

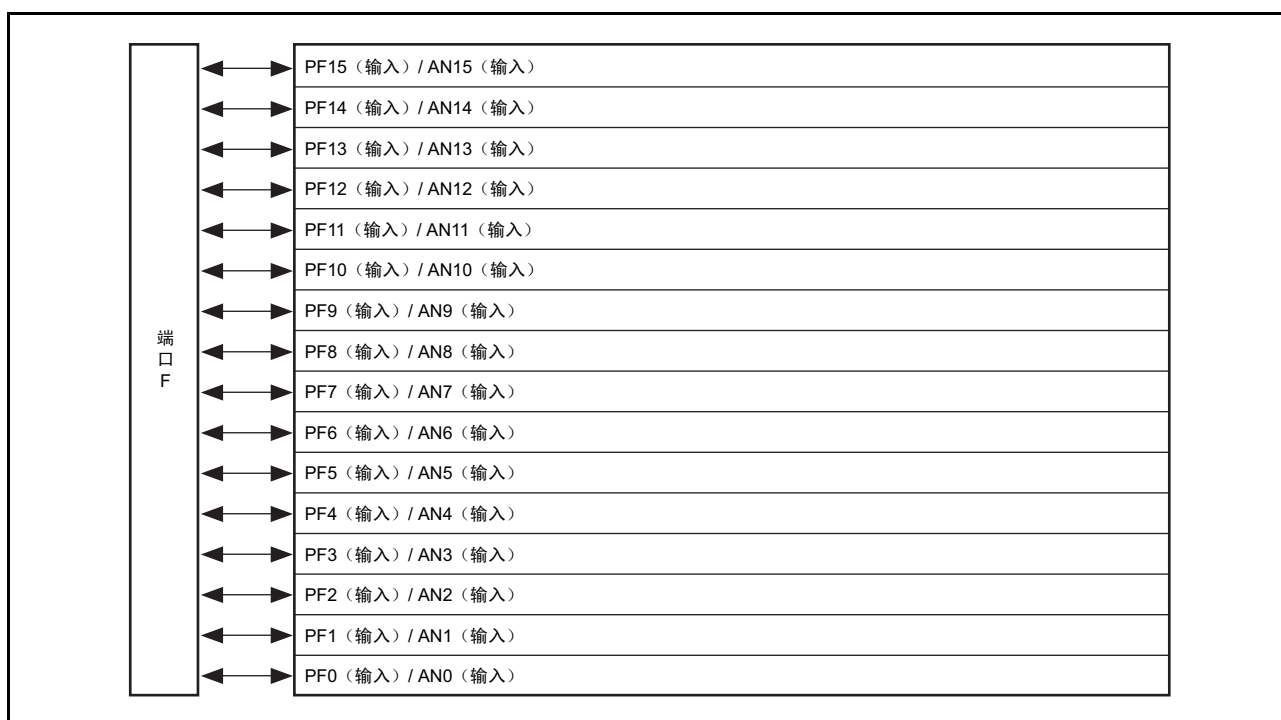


图 22.16 端口 F (SH7086)

## 22.6.1 寄存器说明

SH7083/84/85 的端口 F 是 8 位输入端口，SH7086 的端口 F 是 16 位输入端口。端口 F 有以下寄存器，有关此寄存器的地址和各处理模式的寄存器状态，请参照“第 27 章 寄存器一览”。

表 22.11 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
端口 F 的数据寄存器 L	PFDR L	R	H'xxxx	H'FFFFD382	8、16

## 22.6.2 端口 F 的数据寄存器 L (PFDR L)

PFDR L 是 16 位只读寄存器，保存端口 F 的数据。SH7083/84/85 的 PF7DR ~ PF0DR 位分别对应 PF7 ~ PF0 引脚（省略有关兼用功能的记述）；SH7086 的 PF15DR ~ PF0DR 位分别对应 PF15 ~ PF0 引脚（省略有关兼用功能的记述）。

即使给这些位写值也被忽视，不影响引脚的状态。如果读这些位，就直接读引脚的状态而不读这些位的值。但是在对 A/D 转换器的模拟输入进行采样的期间，读的值是 1。端口 F 的数据寄存器的读写操作如表 22.12 所示。

### • PFDR L (SH7083/84/85)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PF7 DR	PF6 DR	PF5 DR	PF4 DR	PF3 DR	PF2 DR	PF1 DR	PF0 DR
初始值:	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 8	—	全 0	R	保留位 读写的值总是 0。
7	PF7DR	引脚的状态	R	参照表 22.12
6	PF6DR	引脚的状态	R	
5	PF5DR	引脚的状态	R	
4	PF4DR	引脚的状态	R	
3	PF3DR	引脚的状态	R	
2	PF2DR	引脚的状态	R	
1	PF1DR	引脚的状态	R	
0	PF0DR	引脚的状态	R	

- PFDRL (SH7086)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF15 DR	PF14 DR	PF13 DR	PF12 DR	PF11 DR	PF10 DR	PF9 DR	PF8 DR	PF7 DR	PF6 DR	PF5 DR	PF4 DR	PF3 DR	PF2 DR	PF1 DR	PF0 DR
初始值:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15	PF15DR	引脚的状态	R	参照表 22.12
14	PF14DR	引脚的状态	R	
13	PF13DR	引脚的状态	R	
12	PF12DR	引脚的状态	R	
11	PF11DR	引脚的状态	R	
10	PF10DR	引脚的状态	R	
9	PF9DR	引脚的状态	R	
8	PF8DR	引脚的状态	R	
7	PF7DR	引脚的状态	R	
6	PF6DR	引脚的状态	R	
5	PF5DR	引脚的状态	R	
4	PF4DR	引脚的状态	R	
3	PF3DR	引脚的状态	R	
2	PF2DR	引脚的状态	R	
1	PF1DR	引脚的状态	R	
0	PF0DR	引脚的状态	R	

表 22.12 端口 F 的数据寄存器 L (PFDRL) 的读写操作

- PFDRL 的 bit15 ~ 0

引脚功能	读	写
通用输入	读引脚的状态	被忽视 (不影响引脚的状态)
ANn 输入	读的值是 1	被忽视 (不影响引脚的状态)

## 第 23 章 闪存

本 LSI 内置 512KB 或者 256KB 的闪存，闪存的特点如下所示：

### 23.1 特点

- 根据 LSI 启动模式设定的 2 种闪存 MAT  
内置的闪存有分配在同一个地址空间的 2 种存储空间（以下称为存储器 MAT），能根据启动时的模式设定，选择从哪一个存储 MAT 启动。在启动后，还能通过存储体转换方式转换 MAT。  
在用户模式中，上电复位时启动的用户 MAT：512KB 或者 256KB  
在用户引导模式中，上电复位时启动的用户 MAT：12KB
- 3 种板上编程模式和 1 种板外编程模式  
板上编程模式  
引导模式：能通过使用内部 SCI 接口的编程模式，改写用户 MAT 和用户引导 MAT。本模式能自动调整主机和本 LSI 之间的位速率。  
用户编程模式：能通过任意的接口改写用户 MAT。  
用户引导模式：能建立任意接口的用户引导程序，并且可改写用户 MAT。  
  
板外编程模式  
编程器模式：能通过使用 PROM 编程器的编程器模式，改写用户 MAT 和用户引导 MAT。
- 通过下载内部程序，建立编程 / 擦除的接口  
本 LSI 内置专用的编程 / 擦除的程序。在将此程序下载到内部 RAM 后，只要设定自变量参数就能进行编程 / 擦除，并且支持用户分支。  
  
用户分支：以 128 字节为单位进行编程处理，编程处理由外加写脉冲和读验证等几个步骤构成；以 1 个分割块为单位进行擦除处理，擦除处理由几个处理步骤构成。能在各步骤之间设定用户处理程序的执行，此设定称为用户分支。
- 通过内部 RAM 对闪存进行仿真的功能  
将闪存和一部分内部 RAM 重叠在一起，就能对闪存的改写进行实时仿真。
- 保护模式  
保护模式有通过设定寄存器进行的软件保护和通过 FWE 引脚进行的硬件保护共 2 种，能设定闪存的编程 / 擦除的保护状态。  
另外，如果在编程 / 擦除中检测到失控等异常情况，就转移到错误保护状态，并中断编程 / 擦除的处理。
- 编程 / 擦除的时间  
闪存的编程时间在 128 字节同时编程时为  $t_{\text{PMS}}$  (Typ.)，平均每 1 字节  $t_{\text{P}}/128\text{ms}$ ；擦除时间平均每块为  $t_{\text{ES}}$  (Typ.)。
- 改写次数  
闪存能改写到  $N_{\text{WEC}}$  次。
- 编程 / 擦除时的工作频率  
编程 / 擦除时的工作频率最大为 40MHz (Pφ)。



## 23.2 概要

### 23.2.1 框图

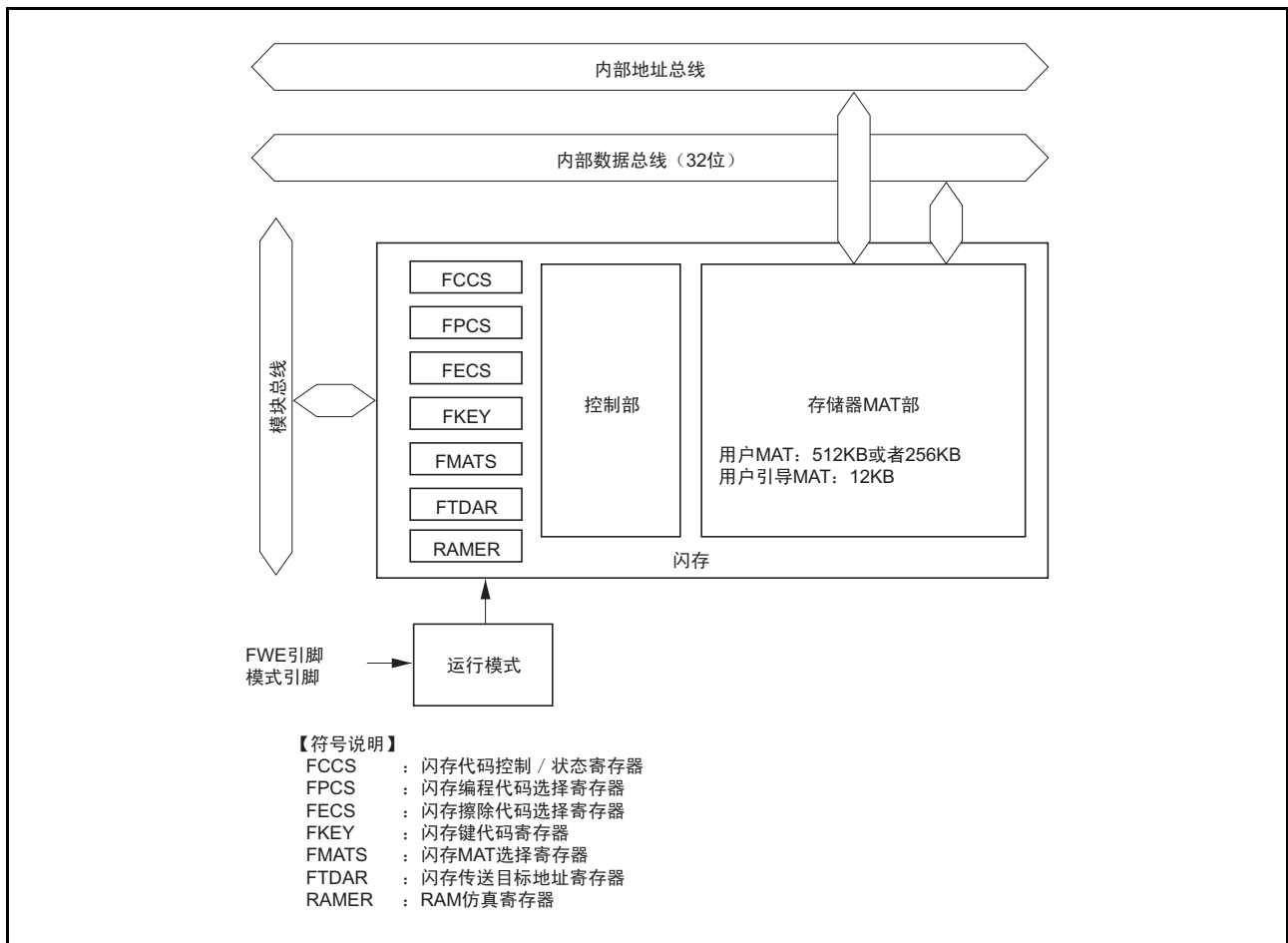


图 23.1 闪存的框图

### 23.2.2 运行模式

如果在复位状态下设定各模式引脚和 FWE 引脚，并进行复位解除，单片机就转移到如图 23.2 所示的各运行模式。各模式引脚和 FWE 引脚的设定请参照表 23.1。

1. 在 ROM 无效模式中，不能读 / 编程 / 擦除闪存，并且不能进行编程 / 擦除接口寄存器的写操作，读的值总是 H'00。
2. 在用户模式中能读闪存，但是不能编程 / 擦除。
3. 在用户编程模式、用户引导模式和引导模式中，能对闪存进行板上读 / 编程 / 擦除。
4. 在编程器模式中，使用 PROM 编程器对闪存进行读 / 编程 / 擦除操作。

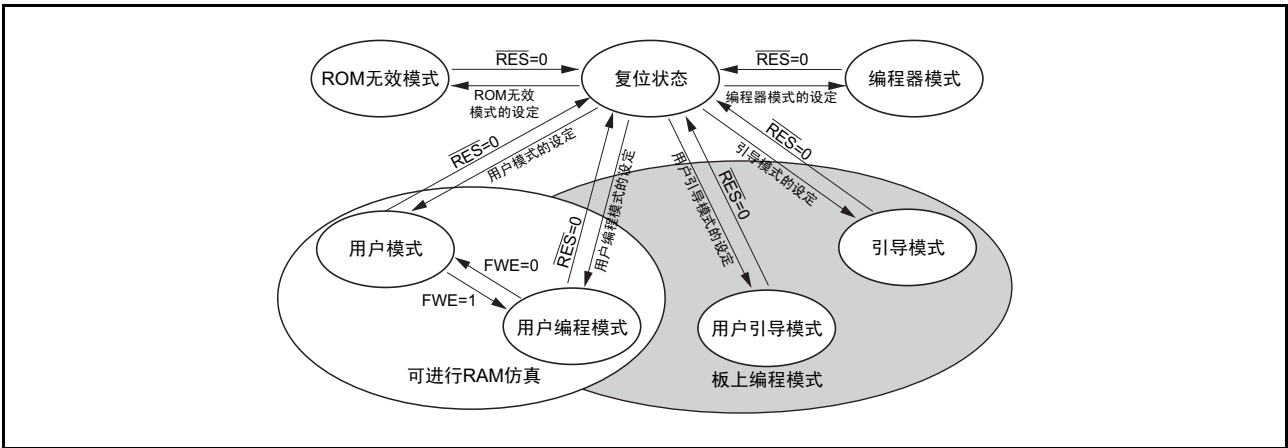


图 23.2 与闪存有关的模式转移图

表 23.1 FWE 引脚、MD 引脚的设定和运行模式 (1) (SH7083/84 的情况)

引脚	复位状态	ROM 无效模式	用户模式	用户编程模式	用户引导模式	引导模式	编程器模式
$\overline{\text{RES}}$	0	1	1	1	1	1	与专用的 PROM 编程器的条件有关
FWE	0/1	0	0	1	1	1	
MD0	0/1	0/1*1	0/1*2	0/1*2	1	0	
MD1	0/1	0	1	1	0	0	

【注】 \*1 MD0=0: 8 位外部总线, MD0=1: 16 位外部总线

\*2 MD0=0: 可使用外部总线, MD0=1: 单芯片模式 (不可使用外部总线)

表 23.1 FWE 引脚、MD 引脚的设定和运行模式 (2) (SH7085/86 的情况)

引脚	复位状态	ROM 无效模式	用户模式	用户编程模式	用户引导模式	引导模式	编程器模式
$\overline{\text{RES}}$	0	1	1	1	1	1	与专用的 PROM 编程器的条件有关
FWE	0/1	0	0	1	1	1	
MD0	0/1	0/1*1	0/1*2	0/1*2	1	0	
MD1	0/1	0	1	1	0	0	

【注】 \*1 MD0=0: 16 位外部总线, MD0=1: 32 位外部总线

\*2 MD0=0: 可使用外部总线, MD0=1: 单芯片模式 (不可使用外部总线)

## 23.2.3 模式比较

有关引导模式、用户编程模式、用户引导模式和编程器模式的编程 / 擦除的比较如表 23.2 所示。

表 23.2 编程模式的比较

	引导模式	用户编程模式	用户引导模式	编程器模式
编程 / 擦除环境	板上编程			板外编程
可编程 / 擦除的 MAT	用户 MAT 用户引导 MAT	用户 MAT	用户 MAT	用户 MAT 用户引导 MAT
编程 / 擦除控制	命令方式	编程 / 擦除接口	编程 / 擦除接口	—
全面擦除	○ (自动)	○	○	○ (自动)
块分割擦除	○ *1	○	○	×
编程数据的传送	从主机经由 SCI	从任意器件经由 RAM	从任意器件经由 RAM	经由编程器
用户分支功能	×	○	○	×
RAM 仿真	×	○	×	×
复位开始时的启动 MAT	嵌入式程序保存 MAT	用户 MAT	用户引导 MAT*2	嵌入式程序保存 MAT
向用户模式的转移	更改模式设定和复位	更改 FWE 设定	更改模式设定和复位	—

【注】 \*1 首先进行全面擦除，然后能进行特定块的擦除。

\*2 首先从嵌入式程序保存 MAT 启动，在检查了闪存相关寄存器后从用户引导 MAT 的复位向量启动。

- 只在引导模式和编程器模式中，才能对用户引导 MAT 进行编程 / 擦除。
- 在引导模式中，首先全面擦除用户 MAT 和用户引导 MAT，然后能通过命令方式对用户 MAT 或者用户引导 MAT 进行编程，但是在进入此状态之前不能读取 MAT 的内容。
- 只进行用户引导 MAT 的编程，而在用户引导模式中进行用户 MAT 的改写。或者因不使用用户引导模式而只进行用户 MAT 的改写。
- 在用户引导模式中，能通过和用户编程模式不同的模式引脚设定，实现任意接口的引导操作。

### 23.2.4 闪存结构

本 LSI 的闪存由 512KB 或者 256KB 的用户 MAT 和 12KB 的用户引导 MAT 构成。

因为用户 MAT 和用户引导 MAT 的起始地址分配了相同的地址，所以在程序的执行或者数据的存取跨越 2 个 MAT 时，需要通过 FMATS 寄存器进行 MAT 的转换。

只要在 ROM 有效模式，无论哪个模式都可读取用户 MAT/ 用户引导 MAT，但是用户引导 MAT 的改写只能在引导模式和编程器模式中进行。

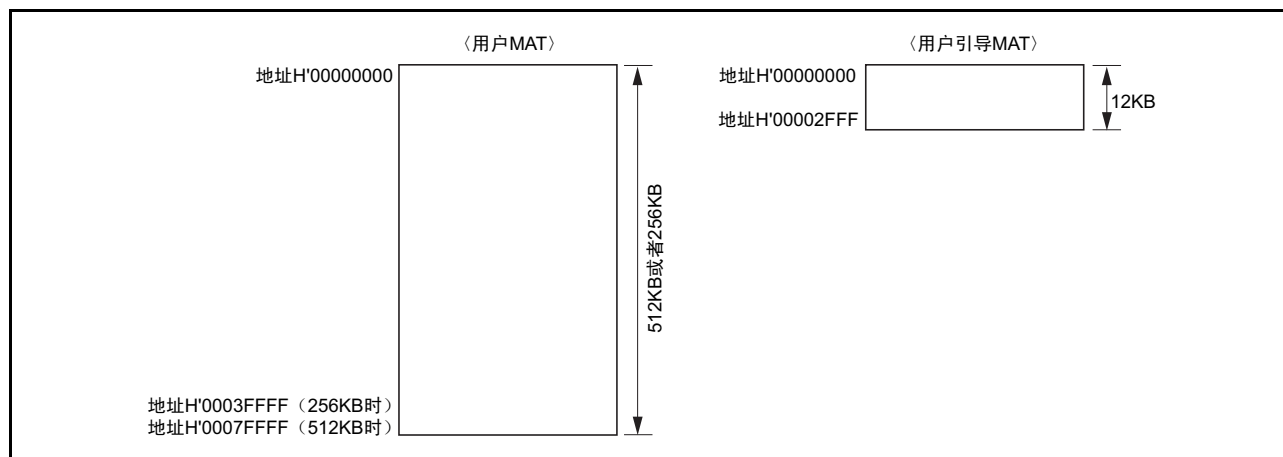


图 23.3 闪存结构图

用户 MAT 和用户引导 MAT 的存储器容量不同，不能存取超过 12KB 空间的用户引导 MAT，如果读超过 12KB 的用户引导 MAT，就读取不定的值。

### 23.2.5 块分割

如图 23.4 所示，用户 MAT 被分割为 64KB（512KB 的产品为 7 块、256KB 的产品为 3 块）、32KB（1 块）和 4KB（8 块）。能以这种块分割为单位进行擦除，在擦除时用 EB0 ~ EB15 指定擦除的块号。

4KB 分割的 8 个块是可进行 RAM 仿真的区域。

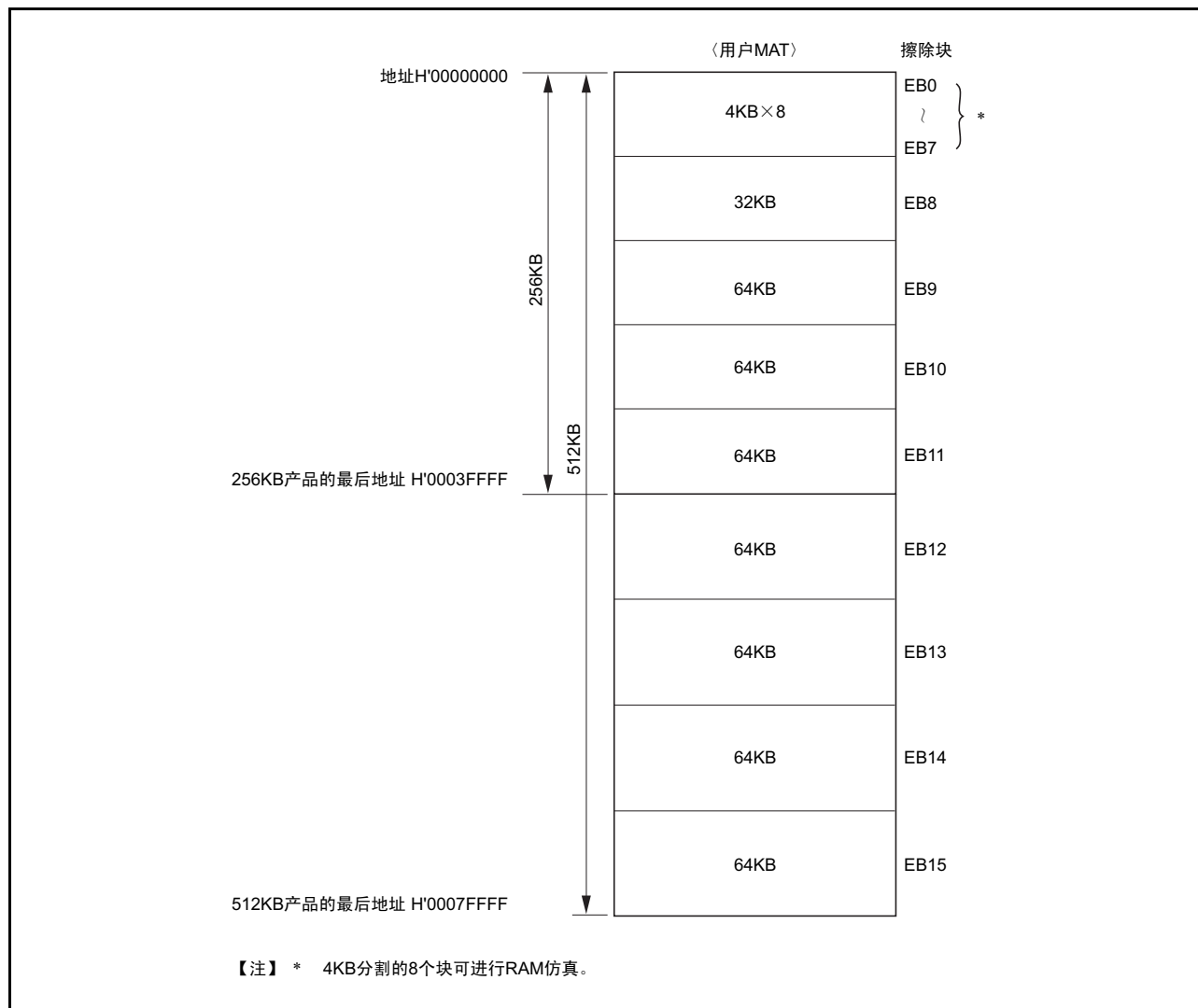


图 23.4 用户 MAT 的块分割

### 23.2.6 编程 / 擦除的接口

先将内部程序下载到内部 RAM，再通过接口寄存器 / 参数，指定编程地址 / 数据以及擦除块等进行编程 / 擦除。

在用户编程模式 / 用户引导模式中，这一连串的过程程序由用户建立，步骤的概要如下所示。详细内容在“23.5.2 用户编程模式”中说明。

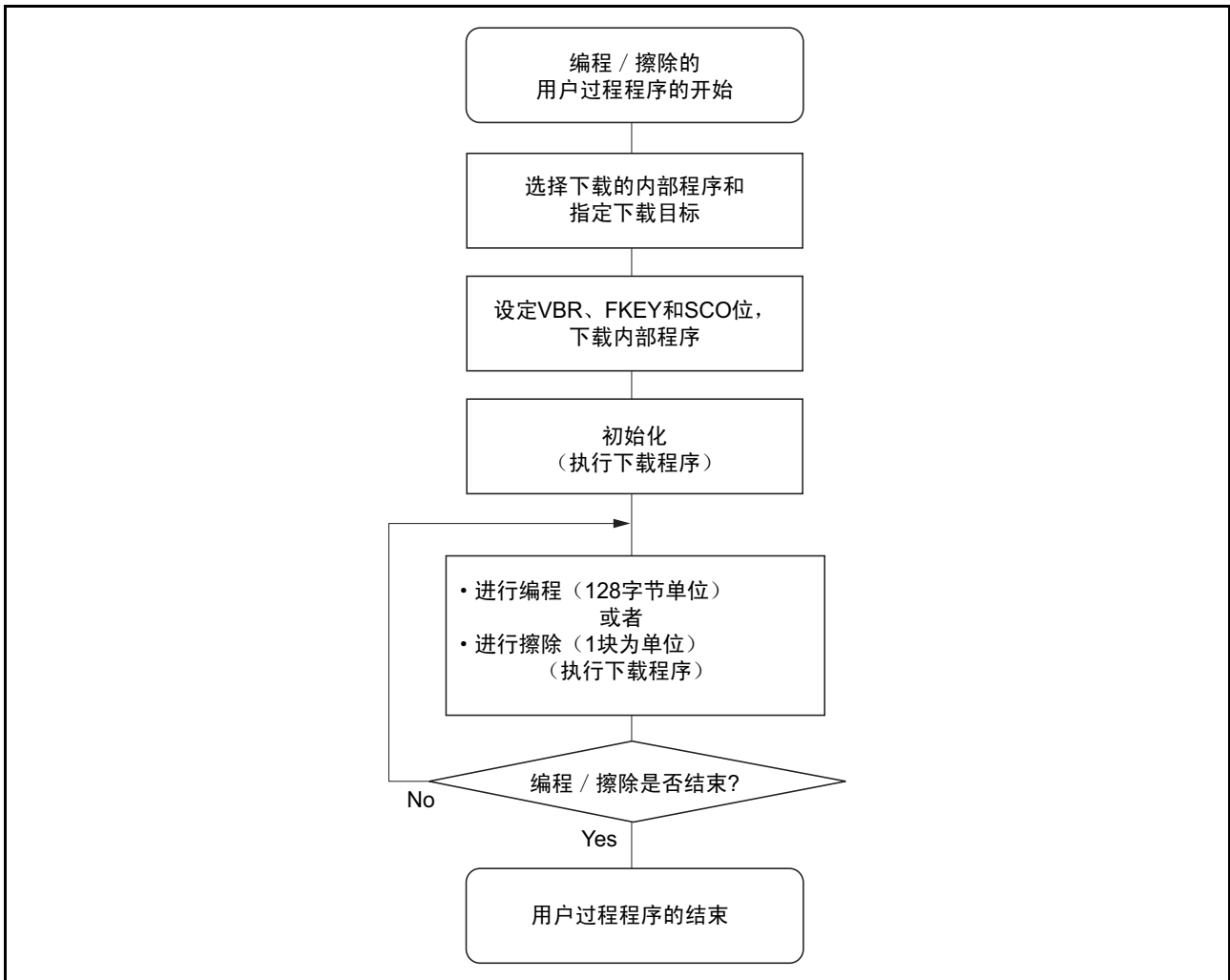


图 23.5 用户过程程序的概要

### (1) 下载的内部程序的选择和下载目标的指定

本 LSI 内置编程 / 擦除的相关程序，能将此程序下载到内部 RAM。通过设定编程 / 擦除的接口寄存器的对应位，选择要下载的内部程序，并能通过 FTDAR 寄存器，指定下载目标的地址。

### (2) 内部程序的下载

在将 CPU 的 VBR 寄存器设定为 H'84000000 后，通过设定编程 / 擦除的接口寄存器的闪存键代码寄存器 FKEY 和闪存代码控制 / 状态寄存器 FCCS 的 SCO 位，自动下载内部程序。

在下载中，闪存 MAT 和嵌入式程序保存区调换。另外，因为不能在编程 / 擦除时读闪存，所以从编程 / 擦除的下载到完成的一连串过程程序必须在内存以外的存储器（内部 RAM 等）执行。

因为下载的结果能返回给编程 / 擦除的接口参数，所以能确认是否正常下载。

另外，在下载结束后能更改 VBR。

### (3) 编程 / 擦除的初始化

在进行编程 / 擦除前设定工作频率和用户分支，必须将用户分支目标设定在非内部闪存区和非下载内部程序的区域。通过编程 / 擦除的接口参数进行这些设定。

### (4) 编程 / 擦除

为了编程 / 擦除，需要将 FWE 引脚设定为高电平，并且将模式设定为用户编程模式。

在编程时，以 128 字节为单位指定编程数据 / 编程目标地址。

在擦除时，以 1 个擦除块为单位指定擦除块。

通过编程 / 擦除的接口参数，设定这些指定并启动内部程序。通过 JSR 指令或者 BSR 指令调用（子程序调用）内部 RAM 中的特定地址，执行内部程序，执行结果返回给编程 / 擦除的接口参数。

对于闪存的编程，需要预先擦除对象区。

编程 / 擦除处理中的中断处理有限制和注意事项，详细内容请参照“23.8.2 执行编程 / 擦除过程中的中断”。

### (5) 连续进行编程 / 擦除的情况

如果 128 字节的编程或者 1 块的擦除不能结束处理，就需要更新编程地址 / 数据或者擦除块号，连续进行编程 / 擦除。

因为在处理结束后被下载的内部程序仍留在内部 RAM 中，所以在连续进行相同的处理时不需要下载和初始化。

## 23.3 输入 / 输出引脚

闪存由表 23.3 所示的引脚控制。

表 23.3 引脚结构

名称	引脚名	输入 / 输出	功能
上电复位	RES	输入	复位
闪存编程的允许	FWE	输入	闪存改写的硬件保护
模式 1	MD1	输入	设定本 LSI 的运行模式
模式 0	MD0	输入	设定本 LSI 的运行模式
发送数据	TXD1 (PA4)	输出	输出串行发送数据（在引导模式中使用）
接收数据	RXD1 (PA3)	输入	输入串行接收数据（在引导模式中使用）

## 23.4 寄存器说明

### 23.4.1 寄存器一览表

内部闪存有效时的闪存控制寄存器 / 参数如表 23.4 所示。

在存取闪存时，有读模式和写模式等几种运行模式。另外，存储器 MAT 有用户 MAT 和用户引导 MAT，对于各种运行模式和 MAT 选择，分配了专用的寄存器 / 参数。运行模式和使用的寄存器 / 参数的对应如表 23.5 所示。

表 23.4 (1) 寄存器结构

寄存器名称	略称 *4	R/W	初始值	地址	存取长度
闪存代码控制 / 状态寄存器	FCCS	R、W*1	H'00*2 H'80*2	H'FFFFCC00	8
闪存编程代码选择寄存器	FPCS	R/W	H'00	H'FFFFCC01	8
闪存擦除代码选择寄存器	FECS	R/W	H'00	H'FFFFCC02	8
闪存键代码寄存器	FKEY	R/W	H'00	H'FFFFCC04	8
闪存 MAT 选择寄存器	FMATS	R/W	H'00*3 H'AA*3	H'FFFFCC05	8
闪存传送目标地址寄存器	FTDAR	R/W	H'00	H'FFFFCC06	8
RAM 仿真寄存器	RAMER	R/W	H'0000	H'FFFFF108	16

【注】 \*1 SCO 位为只写（读取的值总是 0），而其他位为只读。

\*2 给 FWE 引脚输入低电平时的 FEW 位的初始值为 0。  
给 FWE 引脚输入高电平时的 FEW 位的初始值为 1。

\*3 以用户模式或者用户编程模式启动时的初始值是 H'00。  
以用户引导模式启动时的初始值是 H'AA。

\*4 RAMER 寄存器以外的各寄存器只在字节存取时有效。  
RAMER 寄存器可进行字节和字存取。

表 23.4 (2) 参数结构

参数名称	略称	R/W	初始值	分配	存取长度
下载的正常 / 失败结果	DPFR	R/W	不定	内部 RAM*	8、16、32
闪存的正常 / 失败结果	FPFR	R/W	不定	CPU 的 R0	8、16、32
闪存多用途地址区	FMPAR	R/W	不定	CPU 的 R5	8、16、32
闪存多用途数据目标区	FMPDR	R/W	不定	CPU 的 R4	8、16、32
闪存的擦除块选择	FEBS	R/W	不定	CPU 的 R4	8、16、32
闪存编程 / 擦除的频率控制	FPEFEQ	R/W	不定	CPU 的 R4	8、16、32
闪存用户分支地址的设定	FUBRA	R/W	不定	CPU 的 R5	8、16、32

【注】 \* FTDAR 寄存器指定的内部 RAM 区起始地址的 1 字节有效。



表 23.5 使用的寄存器 / 参数和对象模式

		下载	初始化	编程	擦除	读	RAM 仿真
编程 / 擦除的接口寄存器	FCCS	○	—	—	—	—	—
	FPCS	○	—	—	—	—	—
	PECS	○	—	—	—	—	—
	FKEY	○	—	○	○	—	—
	FMATS	—	—	○ (*1)	○ (*1)	○ (*2)	—
	FTDAR	○	—	—	—	—	—
编程 / 擦除的接口参数	DPFR	○	—	—	—	—	—
	FPFR	—	○	○	○	—	—
	FPEFEQ	—	○	—	—	—	—
	FUBRA	—	○	—	—	—	—
	FMPAR	—	—	○	—	—	—
	FMPDR	—	—	○	—	—	—
	FEBS	—	—	—	○	—	—
RAM 仿真	RAMER	—	—	—	—	—	○

【注】 \*1 在用户引导模式中，当对用户 MAT 进行编程 / 擦除时需要设定。

\*2 根据启动模式和读对象 MAT 的组合，有可能需要设定。

### 23.4.2 编程 / 擦除的接口寄存器

编程 / 擦除的接口寄存器都是 8 位寄存器，并只能以字节存取。

#### (1) 闪存代码控制 / 状态寄存器 (FCCS)

FCCS 由 FWE 引脚状态的监视位、闪存编程 / 擦除中的错误监视位以及下载内部程序的请求位构成。

位:	7	6	5	4	3	2	1	0
	FWE	MAT	-	FLER	-	-	-	SCO
初始值:	1/0	1/0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	(R)/W

位	位名	初始值	R/W	说 明
7	FWE	1/0	R	闪存编程的允许位 FWE 位监视 FWE 引脚（对闪存编程 / 擦除进行硬件保护）的输入电平。 根据 FWE 引脚状态，初始值为 0 或者 1。 0: 给 FWE 引脚输入低电平时（硬件保护状态） 1: 给 FWE 引脚输入高电平时
6	MAT	1/0	R	MAT 位 表示是选择了用户 MAT 还是选择了用户引导 MAT 的位。 0: 选择了用户 MAT 1: 选择了用户引导 MAT

位	位名	初始值	R/W	说 明
5	—	0	R	保留位 读写的值总是 0。
4	FLER	0	R	闪存错误 此位表示在闪存的编程 / 擦除中发生错误。如果 FLER=1, 闪存就转移到错误保护状态。 另外, 在 FLER=1 时, 由于闪存内部加有高电压, 为了减少闪存的损伤, 必须在长于通常的 100 $\mu$ s 复位输入期间后解除复位。 0: 闪存正常运行。闪存的编程 / 擦除保护 (错误保护) 无效 [清除条件] 当上电复位时 1: 表示在闪存的编程 / 擦除中发生错误。闪存的编程 / 擦除保护 (错误保护) 有效 [置位条件] 请参照 “23.6.3 错误保护”。
3 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	SCO	0	(R)/W	源程序的复制操作 此位请求将内部的编程 / 擦除程序下载到内部 RAM。如果给本位写 1, FPCS/FECS 寄存器选择的内部程序就自动下载到 FTDAR 寄存器指定的内部 RAM 区。为了给本位写 1, 需要解除 RAM 仿真状态, 给 FKEY 寄存器写 H'A5, 并且在内部 RAM 中执行。在给本位写 1 后, 必须立刻执行 4 条 NOP 指令。 有关下载中的中断和下载时间请分别参照 “23.8.2 执行编程 / 擦除过程中的中断” 和 “23.8.3 其他注意事项”。另外, 因为在下载结束时此位被清 0, 所以读不到此位为 1 的状态。 因为在通过 SCO 位进行下载时, 伴有内部程序保存区的存储体转换的特殊中断处理, 所以必须在下载请求 (SCO=1) 前将 VBR 的值设定为 H'84000000, 如果不设定 VBR 就会失控。当确认下载已结束, 能更改 VBR。另外, 在使用 SCO 功能时, 必须使用 FWE 引脚为高电平的模式。 0: 不将内置的编程 / 擦除程序下载到内部 RAM。 [清除条件] 当下载结束时 1: 发生将内置的编程 / 擦除程序下载到内部 RAM 的请求。 [置位条件] 在满足以下全部条件的状态下写 1 <ul style="list-style-type: none"> <li>• 给 FKEY 寄存器写 H'A5</li> <li>• 在内部 RAM 中执行</li> <li>• 非 RAM 仿真模式 (RAMER 的 RAMS=0)</li> </ul>

## (2) 闪存编程代码选择寄存器 (FPCS)

FPCS 选择与编程有关的内部程序的下载。

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PPVS
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

位	位名	初始值	R/W	说 明
7 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	PPVS	0	R/W	编程脉冲 Single 选择编程的程序。 0: 不选择内部编程程序。 [清除条件] 当传送结束时被清除。 1: 选择内部编程程序。

## (3) 闪存擦除代码选择寄存器 (FECS)

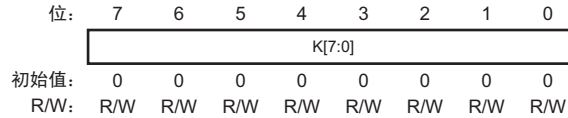
FECS 选择与擦除有关的内部程序的下载。

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	EPVB
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

位	位名	初始值	R/W	说 明
7 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	EPVB	0	R/W	擦除脉冲验证块 选择内部擦除程序。 0: 不选择内部擦除程序。 [清除条件] 当传送结束时 1: 选择内部擦除程序。

## (4) 闪存键代码寄存器 (FKEY)

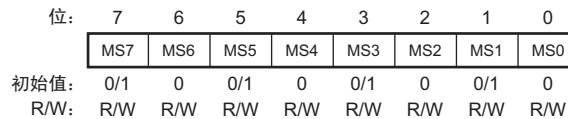
FKEY 允许内部程序的下载和闪存编程 / 擦除的软件保护。在给用于下载内部程序的 SCO 位写 1 前或者在执行已下载的编程 / 擦除程序前，如果未写键代码，就不能执行各自的处理。



位	位名	初始值	R/W	说 明
7 ~ 0	K[7:0]	全 0	R/W	<p>键代码</p> <p>只在给 FKEY 寄存器写 H'A5 后，SCO 位的写操作才有效。如果给 FKEY 寄存器写 H'A5 以外的值，就无法给 SCO 位写 1，导致内部程序无法下载到内部 RAM。</p> <p>只在给 FKEY 寄存器写 H'5A 后，才能进行闪存的编程 / 擦除。如果给 FKEY 寄存器写 H'5A 以外的值，即使执行内部编程 / 擦除程序也不能进行闪存的编程 / 擦除。</p> <p>H'A5: 允许写 SCO 位（如果值不为 H'A5，就无法设定 SCO 位）。</p> <p>H'5A: 允许编程 / 擦除（如果值不为 H'5A，就为软件保护状态）。</p> <p>H'00: 初始值</p>

## (5) 闪存 MAT 选择寄存器 (FMATS)

FMATS 指定是选择用户 MAT 还是选择用户引导 MAT。

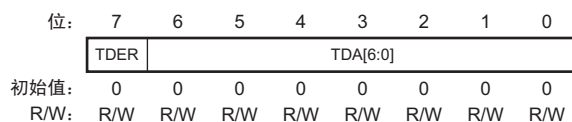


位	位名	初始值	R/W	说 明
7	MS7	0/1	R/W	<p>MAT 选择</p> <p>当值不是 H'AA 时为用户 MAT 选择状态，当值是 H'AA 时为用户引导 MAT 选择状态。如果通过内部 RAM 中的指令给 FMATS 写值，就进行 MAT 的转换。</p> <p>必须按照“23.8.1 用户 MAT 和用户引导 MAT 的转换”进行 MAT 的转换（即使通过 FMATS 选择用户引导 MAT，也不能在用户编程模式中改写用户引导 MAT。用户引导 MAT 的改写必须在引导模式或者编程器模式中进行）。</p> <p>H'AA: 选择用户引导 MAT（非 H'AA 时为用户 MAT 选择状态），是以用户引导模式启动时的初始值。</p> <p>H'00: 是以用户引导模式以外的模式启动时的初始值。</p> <p>【注】[可编程的条件]是在内部 RAM 中的执行状态。</p>
6	MS6	0	R/W	
5	MS5	0/1	R/W	
4	MS4	0	R/W	
3	MS3	0/1	R/W	
2	MS2	0	R/W	
1	MS1	0/1	R/W	
0	MS0	0	R/W	

## (6) 闪存传送目标地址寄存器 (FTDAR)

FTDAR 指定内部程序下载目标的内部 RAM 地址。

必须在给 FCCS 寄存器的 SCO 位写 1 前设定此寄存器。初始值 H'00 表示内部 RAM 的起始地址 (H'FFFF9000)。



位	位名	初始值	R/W	说 明
7	TDER	0	R/W	传送目标地址的设定错误 如果 bit6 ~ 0 (TDA6 ~ TDA0) 指定的下载起始地址有错误, 就被置 1。判断地址指定错误的方法: 将 FCCS 寄存器的 SCO 位设定为 1, 在下载处理执行结束时, 判断 TDA6 ~ TDA0 的值是否在 H'00 ~ H'04 的范围内。必须在将 SCO 位设定为 1 前, 将 FTDAR 寄存器设定在 H'00 ~ H'04 的范围内, 并且将此位的值置 0。 0: TDA6 ~ TDA0 的设定为正常值 1: TDER、TDA6 ~ TDA0 的设定值为 H'05 ~ H'FF, 表示下载中断。
6 ~ 0	TDA[6:0]	全 0	R/W	传送目标地址 指定下载的起始地址。可设定的值为 H'00 ~ H'04, 并能以 2KB 单位指定内部 RAM 的下载起始地址。 不能设定 H'05 ~ H'7F 的值, 否则就在下载处理中将此寄存器的 bit7 (TDER) 置 1 并且不进行内部程序的下载。 H'00: 将下载的起始地址设定为 H'FFFF9000 H'01: 将下载的起始地址设定为 H'FFFF9800 H'02: 将下载的起始地址设定为 H'FFFFA000 H'03: 将下载的起始地址设定为 H'FFFFA800 H'04: 将下载的起始地址设定为 H'FFFFB000 H'05 ~ H'7F: 不能设定, 否则就在下载中 bit7 (TDER) 变为 1 并且中断下载处理。

### 23.4.3 编程 / 擦除的接口参数

编程 / 擦除的接口参数对已下载的内部程序指定工作频率、用户分支目标地址、编程数据的保存位置、编程的目标地址、擦除块等以及进行处理结果的交换，此参数使用 CPU 的通用寄存器（R4、R5 和 R0）或者内部 RAM 区。初始值不定。

在下载时，保存 CPU 的全部寄存器；在初始化和执行内部程序时，保存除 R0 以外的 CPU 寄存器。R0 保存处理结果的返回值。因为堆栈区用作寄存器的保存和工作区，所以在开始处理时必须确保堆栈区（使用的堆栈区的容量最大为 128 字节）。

编程 / 擦除的接口参数被用于以下 4 个项目：

1. 下载控制
2. 编程 / 擦除前的初始化
3. 进行编程
4. 进行擦除

各项使用的参数都不相同，其对应表如表 23.6 所示。

在此，FPFR 参数返回初始化、编程和擦除的处理结果，但是根据处理内容，位的含意不同。请参照各项处理的 FPFR 说明部分。

表 23.6 使用的参数和对象模式

参数名	略称	下载	初始化	编程	擦除	R/W	初始值	分配
下载的正常 / 失败结果	DPFR	○	—	—	—	R/W	不定	内部 RAM*
闪存的正常 / 失败结果	FPFR	—	○	○	○	R/W	不定	CPU 的 R0
闪存编程 / 擦除的频率控制	FPEFEQ	—	○	—	—	R/W	不定	CPU 的 R4
闪存用户分支地址的设定	FUBRA	—	○	—	—	R/W	不定	CPU 的 R5
闪存多用途地址区	FMPAR	—	—	○	—	R/W	不定	CPU 的 R5
闪存多用途数据目标区	FMPDR	—	—	○	—	R/W	不定	CPU 的 R4
闪存擦除块的选择	FEBS	—	—	—	○	R/W	不定	CPU 的 R4

【注】 \* FTDAR 寄存器指定的下载目标起始地址的 1 字节

## (1) 下载控制

通过将 SCO 位设定为 1，自动下载内部程序。被下载的内部 RAM 区是从 FTDAR 寄存器指定的起始地址开始的 3KB 容量。有关内部 RAM 的地址映像请参照图 23.10。

下载控制由前面所述的编程 / 擦除的接口寄存器设定，返回值由 DPFR 参数传递。

- 下载的正常 / 失败结果参数（DPFR: FTDAR 寄存器指定的内部 RAM 起始地址的 1 字节）

这是下载结果的返回值，必须通过此参数的值判断下载是否已执行完毕。由于不能确认是否已将 SCO 位设定为 1，所以必须在开始下载前（将 SCO 位设定为 1 之前），将 FTDAR 寄存器指定的内部 RAM 起始地址的 1 字节设定为下载返回值以外的值（H'FF 等），然后进行准确的判断。有关下载结果的检查方法请参照“23.5.2（2.5）”项。

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	SS	FK	SF
初始值:	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7 ~ 3	—	不定	R/W	未使用位 返回 0。
2	SS	不定	R/W	源选择的错误检测位 1 次操作只能指定 1 种可下载的内部程序。如果选择 2 种以上或者不选择或者选择未被映像，就会发生错误。 0: 正常选择了下载程序 1: 发生下载错误（多重选择或者选择了未被映像）
1	FK	不定	R/W	闪存键寄存器的错误检测位 此位返回检查 FKEY 寄存器的值是否为 H'A5 的结果。 0: FKEY 寄存器的设定值正常（FKEY=H'A5） 1: FKEY 寄存器的设定值错误（FKEY 不为 H'A5）
0	SF	不定	R/W	成功 / 失败位 此位返回下载是否已正常结束。 0: 下载正常结束（无错误） 1: 下载异常结束（发生错误）

(2) 编程 / 擦除的初始化

在被下载的编程 / 擦除内部程序中包含初始化程序。

在编程 / 擦除中，由 CPU 指令构成了所定时间宽的等待周期，因此需要设定 CPU 的工作频率。另外，因为支持用户分支功能，所以还需要设定用户分支目标地址。

将这些设定作为下载的编程 / 擦除程序参数，在初始化程序中进行设定。

- 闪存编程 / 擦除的频率控制参数 (FPEFEQ: CPU 的通用寄存器 R4)  
这是设定 CPU 工作频率的参数。  
本 LSI 的工作频率范围请参照 “28.3.1 时钟时序”。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	F15	F14	F13	F12	F11	F10	F9	F8	F7	F6	F5	F4	F3	F2	F1	F0
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
31 ~ 16	—	不定	R/W	未使用位 必须置 0。
15 ~ 0	F15 ~ F0	不定	R/W	频率的设定位 设定 CPU 的工作频率，必须按以下方法算出设定值： 1. 将以 MHz 单位表示的工作频率取到小数点后 2 位，小数点后的第 3 位四舍五入。 2. 将扩大 100 倍的值转换为 2 进制数，并写到 FPEFEQ 参数（通用寄存器 R4）。作为具体例子，在 CPU 的工作频率为 28.882MHz 时，设定值如下： 1. 将 28.882 的小数点后的第 3 位四舍五入，得到 28.88。 2. 将 28.88×100=2888 转换为 2 进制数，得到的 b'0000,1011,0100,1000 (H'0B48) 设定到 R4。



- 闪存用户分支地址的设定参数（FUBRA：CPU 的通用寄存器 R5）  
这是设定用户分支目标地址的参数。能按编程 / 擦除时决定的处理单位执行已设定的用户程序。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	UA31	UA30	UA29	UA28	UA27	UA26	UA25	UA24	UA23	UA22	UA21	UA20	UA19	UA18	UA17	UA16
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UA15	UA14	UA13	UA12	UA11	UA10	UA9	UA8	UA7	UA6	UA5	UA4	UA3	UA2	UA1	UA0
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
31 ~ 0	UA31 ~ UA0	不定	R/W	<p>用户分支目标地址</p> <p>当不需要用户分支时，必须设定 H'00000000。必须将用户分支目标地址设定在非内部闪存区或者设定在非传送内部程序的 RAM 区或者设定在外部总线空间。必须注意：不能转移到没有执行码的区域，以免失控，并且不能破坏内部程序的下载区和堆栈区，否则就不能保证闪存的值。在用户分支目标的处理中，不能启动内部程序的下载程序、初始化程序和编程 / 擦除程序。不能保证从用户分支目标返回时的编程 / 擦除。另外，不能改写已准备好的编程数据。</p> <p>必须保存通用寄存器 R8 ~ R15。通用寄存器 R0 ~ R7 不需保存而能被直接使用。</p> <p>另外，不能在用户分支目标的处理中改写编程 / 擦除的接口寄存器或者转移到 RAM 仿真模式。</p> <p>在用户分支处理结束后，必须使用 RTS 指令返回到编程 / 擦除程序。</p> <p>有关用户分支处理的执行间隔，请参照“23.8.3 (2) 用户分支处理的间隔”。</p>

- 闪存的正常 / 失败结果参数 (FPFR: CPU 的通用寄存器 R0)

在此说明作为初始化处理结果返回值的 FPFR。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	BR	FQ	SF
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
31 ~ 3	—	不定	R/W	未使用位 返回 0。
2	BR	不定	R/W	用户分支的错误检测位 检查被指定的用户分支目标地址是否在非下载编程 / 擦除相关程序的保存区, 并返回其检查结果。 0: 用户分支地址的设定值正常 1: 用户分支地址的设定值异常
1	FQ	不定	R/W	频率的错误检测位 检查被指定的 CPU 工作频率是否在支持工作频率范围内, 并返回其检查结果。 0: 工作频率的设定值正常 1: 工作频率的设定值异常
0	SF	不定	R/W	成功 / 失败位 返回初始化是否正常结束。 0: 初始化正常结束 (无错误) 1: 初始化异常结束 (发生错误)

## (3) 编程的执行

在进行闪存的编程中，需要将用户 MAT 中的编程目标地址和编程数据传递给已下载的编程程序。

1. 必须将用户 MAT 中的编程目标起始地址设定到通用寄存器 R5。此参数称为 FMPAR（闪存多用途地址区参数）。

因为编程数据总是以 128 字节为单位，所以必须将用户 MAT 中的编程起始地址的边界地址的低 8 位（A7 ~ A0）置 H'00 或者 H'80。

2. 必须将用户 MAT 的编程数据准备到连续的区域。编程数据必须在能由 CPU 的 MOV.B 指令存取的空间并且在内部闪存空间以外。

如果要写的的数据未满 128 字节，必须填写虚码（H'FF），准备 128 字节的编程数据。

必须将保存准备好的编程数据的区域起始地址设定到通用寄存器 R4。此参数称为 FMPDR（闪存多用途数据目标区参数）。

有关编程处理过程的详细内容在“23.5.2 用户编程模式”中说明。

- 闪存多用途地址区参数（FMPAR：CPU 的通用寄存器 R5）

设定用户 MAT 中的编程目标起始地址。

如果设定为闪存空间以外的地址就发生错误。

另外，编程目标起始地址必须以 128 字节为边界，否则会发生错误。这些错误反映到 FPFR 参数的 bit1（WA 位）。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MOA31	MOA30	MOA29	MOA28	MOA27	MOA26	MOA25	MOA24	MOA23	MOA22	MOA21	MOA20	MOA19	MOA18	MOA17	MOA16
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MOA15	MOA14	MOA13	MOA12	MOA11	MOA10	MOA9	MOA8	MOA7	MOA6	MOA5	MOA4	MOA3	MOA2	MOA1	MOA0
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 0	MOA31 ~ MOA0	不定	R/W	MOA31 ~ MOA0 保存用户 MAT 中的编程目标起始地址，从在此指定的用户 MAT 起始地址开始连续写 128 字节，所以指定的编程目标起始地址以 128 字节为边界，并且 MOA6 ~ MOA0 总是 0。

- 闪存多用途数据目标区参数（FMPDR：CPU 的通用寄存器 R4）  
 设定用户 MAT 的编程数据保存区的起始地址。如果编程数据的保存目标在闪存内就发生错误，此错误反映到 FPFR 参数的 bit2（WD 位）。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MOD31	MOD30	MOD29	MOD28	MOD27	MOD26	MOD25	MOD24	MOD23	MOD22	MOD21	MOD20	MOD19	MOD18	MOD17	MOD16
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MOD15	MOD14	MOD13	MOD12	MOD11	MOD10	MOD9	MOD8	MOD7	MOD6	MOD5	MOD4	MOD3	MOD2	MOD1	MOD0
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 0	MOD31 ~ MOD0	不定	R/W	MOD31 ~ MOD0 保存用户 MAT 的编程数据保存区的起始地址，从在此指定的起始地址开始将连续 128 字节的数据写到用户 MAT。

- 闪存的正常 / 失败结果参数（FPFR：CPU 的通用寄存器 R0）  
 在此说明作为编程处理结果返回值的 FPFR。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	MD	EE	FK	-	WD	WA	SF
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 7	—	不定	R/W	未使用位 返回 0。
6	MD	不定	R/W	编程模式相关设定的错误检测位 返回 FWE 引脚的输入值是高电平并且不是错误保护状态的检查结果。 在 FWE 引脚为低电平时或者在错误保护状态时写 1，并能通过 FCCS 寄存器的 bit7（FWE）和 bit4（FLER）确认这些状态。有关错误保护状态的转移条件，请参照“23.6.3 错误保护”。 0：FWE、FLER 状态正常（FWE=1、FLER=0） 1：FWE=0 或者 FLER=1，处于不能编程的状态

位	位名	初始值	R/W	说 明
5	EE	不定	R/W	<p>编程时的错误检测位</p> <p>如果因用户 MAT 未被擦除而引起不能对指定的数据进行编程、或者从用户分支处理返回时一部分闪存相关寄存器被改写，此位就返回 1。如果因这些原因而此位为 1，用户 MAT 就很有可能在途中被改写，所以必须在解除错误原因后从擦除开始重新执行。另外，即使在 FMATS 寄存器的值为 H'AA（用户引导 MAT 选择状态）时进行编程，也在编程执行时出现错误。此时，用户 MAT 和用户引导 MAT 都没被改写。</p> <p>用户引导 MAT 的编程必须在引导模式或者编程器模式中进行。</p> <p>0：编程处理正常结束 1：编程处理异常结束（不能保证编程结果）</p>
4	FK	不定	R/W	<p>闪存键寄存器的错误检测位</p> <p>返回开始编程处理前的 FKEY 寄存器值的检查结果。</p> <p>0：FKEY 寄存器的设定值正常（FKEY=H'5A） 1：FKEY 寄存器的设定值错误（FKEY 不为 H'5A）</p>
3	—	不定	R/W	<p>未使用位</p> <p>返回 0。</p>
2	WD	不定	R/W	<p>编程数据地址的错误检测位</p> <p>如果将闪存区的地址指定为编程数据保存目标的起始地址，就会发生错误。</p> <p>0：编程数据地址的设定值正常 1：编程数据地址的设定值异常</p>
1	WA	不定	R/W	<p>编程地址的错误检测位</p> <p>如果将以下地址指定为编程目标的起始地址就会发生错误：</p> <ul style="list-style-type: none"> <li>• 将非闪存区指定为编程目标地址</li> <li>• 指定的地址不是以 128 字节为边界（A6 ~ A0 不为 0）</li> </ul> <p>0：编程目标地址的设定值正常 1：编程目标地址的设定值异常</p>
0	SF	不定	R/W	<p>成功 / 失败位</p> <p>此位返回编程处理是否已正常结束。</p> <p>0：正常结束（无错误） 1：异常结束（发生错误）</p>

## (4) 擦除的执行

在进行闪存的擦除中，需要将用户 MAT 中的擦除块号传递给已下载的擦除程序，并将它设定到 FEBS 参数（通用寄存器 R4）。

从 0 ~ 15 的块号中指定 1 块。

有关擦除处理过程的详细内容在“23.5.2 用户编程模式”中说明。

- 闪存擦除块的选择参数（FEBS：CPU 的通用寄存器 R4）

指定擦除块号，不能指定多个块号。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	EBS[7:0]							
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 8	—	不定	R/W	未使用位 必须置 0。
7 ~ 0	EBS[7:0]	不定	R/W	<ul style="list-style-type: none"> <li>为 512KB 闪存时 在 0 ~ 15 的范围内设定擦除块号。0 和 15 分别对应块 EB0 和块 EB15，如果设定 0 ~ 15（H'00 ~ H'0F）以外的数就出错。</li> <li>为 256KB 闪存时 在 0 ~ 11 的范围内设定擦除块号。0 和 11 分别对应块 EB0 和块 EB11，如果设定 0 ~ 11（H'00 ~ H'0B）以外的数就出错。</li> </ul>

- 闪存的正常 / 失败结果参数 (FPFR: CPU 的通用寄存器 R0)  
在此说明作为擦除处理结果返回值的 FPFR。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	MD	EE	FK	EB	-	-	SF
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
31 ~ 7	—	不定	R/W	未使用位 返回 0。
6	MD	不定	R/W	擦除模式相关设定的错误检测位 返回 FWE 引脚的输入值为高电平并且不是错误保护状态的检查结果。在 FWE 引脚为低电平时或者在错误保护状态时写 1，并能通过 FCCS 寄存器的 bit7 (FWE) 和 bit4 (FLER) 确认这些状态。另外，有关错误保护状态的转移条件请参照“23.6.3 错误保护”。 0: FWE、FLER 状态正常 (FWE=1、FLER=0) 1: FWE=0 或者 FLER=1，处于不能擦除的状态
5	EE	不定	R/W	擦除时的错误检测位 如果不能擦除用户 MAT 或者从用户分支处理返回时一部分闪存相关寄存器被改写，此位就返回 1。如果因这些原因而此位为 1，用户 MAT 就很有可能在途中被擦除，所以必须在解除错误原因后重新进行擦除。 另外，即使在 FMATS 寄存器的值为 H'AA (用户引导 MAT 选择状态) 时进行擦除，也在擦除执行时出现错误。此时，用户 MAT 和用户引导 MAT 都没被擦除。 用户引导 MAT 的擦除必须在引导模式或者编程器模式中进行。 0: 擦除处理正常结束 1: 擦除处理异常结束 (不能保证擦除结果)
4	FK	不定	R/W	闪存键寄存器的错误检测位 返回开始擦除处理前的 FKEY 寄存器值的检查结果。 0: FKEY 寄存器的设定值正常 (FKEY=H'5A) 1: FKEY 寄存器的设定值错误 (FKEY 不为 H'5A)
3	EB	不定	R/W	擦除块的选择错误检测位 返回被指定的擦除块号是否在用户 MAT 的块范围内的检查结果。 0: 擦除块号的设定值正常 1: 擦除块号的设定值异常
2、1	—	不定	R/W	未使用位 返回 0。
0	SF	不定	R/W	成功 / 失败位 返回擦除处理是否已正常结束。 0: 正常结束 (无错误) 1: 异常结束 (发生错误)

### 23.4.4 RAM 仿真寄存器 (RAMER)

在对用户 MAT 的改写进行实时仿真时，RAMER 设定与一部分内部 RAM 重叠的用户 MAT 区。必须在用户模式或者用户编程模式中进行 RAM 仿真。

有关用户 MAT 区的分割方法，请参照表 23.7。另外，为了准确地执行仿真功能，不能在改写本寄存器后立即存取 RAM 仿真的对象 MAT。否则，就不能保证正常的存取。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	RAMS	RAM[2:0]		
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 4	—	全 0	R	保留位 读写的值总是 0。
3	RAMS	0	R/W	RAM 选择 是选择还是不选择通过 RAM 进行用户 MAT 仿真。当 RAMS=1 时，用户 MAT 的全部块为编程 / 擦除保护。 0: 不选择仿真 用户 MAT 全部块的编程 / 擦除保护无效 1: 选择仿真 用户 MAT 全部块的编程 / 擦除保护有效
2 ~ 0	RAM[2:0]	000	R/W	用户 MAT 区的选择 和 bit3 一起使用，选择与内部 RAM 重叠的用户 MAT 区（参照表 23.7）。

表 23.7 RAM 区和用户 MAT 区的重叠

RAM 区	块名	RAMS	RAM2	RAM1	RAM0
H'FFFA000 ~ H'FFFAFFF	RAM 区 (4KB)	0	x	x	x
H'00000000 ~ H'00000FFF	EB0 (4KB)	1	0	0	0
H'00001000 ~ H'00001FFF	EB1 (4KB)	1	0	0	1
H'00002000 ~ H'00002FFF	EB2 (4KB)	1	0	1	0
H'00003000 ~ H'00003FFF	EB3 (4KB)	1	0	1	1
H'00004000 ~ H'00004FFF	EB4 (4KB)	1	1	0	0
H'00005000 ~ H'00005FFF	EB5 (4KB)	1	1	0	1
H'00006000 ~ H'00006FFF	EB6 (4KB)	1	1	1	0
H'00007000 ~ H'00007FFF	EB7 (4KB)	1	1	1	1

【注】 x: Don't care



## 23.5 板上编程模式

如果在板上编程模式中设定引脚并启动复位，就转移到能编程 / 擦除内部闪存的板上编程状态。板上编程模式有用户编程模式、用户引导模式和引导模式 3 种运行模式。

各模式转移的引脚设定方法请参照表 23.1，闪存各模式的状态转移图请参照图 23.2。

### 23.5.1 引导模式

引导模式是使用内部 SCI 从主机发送控制命令和编程数据的方式，进行用户 MAT 或者用户引导 MAT 的编程 / 擦除。需要预先给主机准备发送控制命令的工具和编程数据，并将使用的 SCI 通信模式设定为异步模式。如果在将本 LSI 的引脚设定为引导模式后启动复位，就启动预先嵌入在单片机内部的引导程序，并在自动调整 SCI 位速率后通过控制命令方式和主机进行通信。

引导模式时的系统构成如图 23.6 所示，引导模式的引脚设定请参照表 23.1。虽然引导模式的 NMI 和其他中断被忽视，但是也必须使中断不发生。另外，必须注意：在引导模式运行中不能使用 AUD。

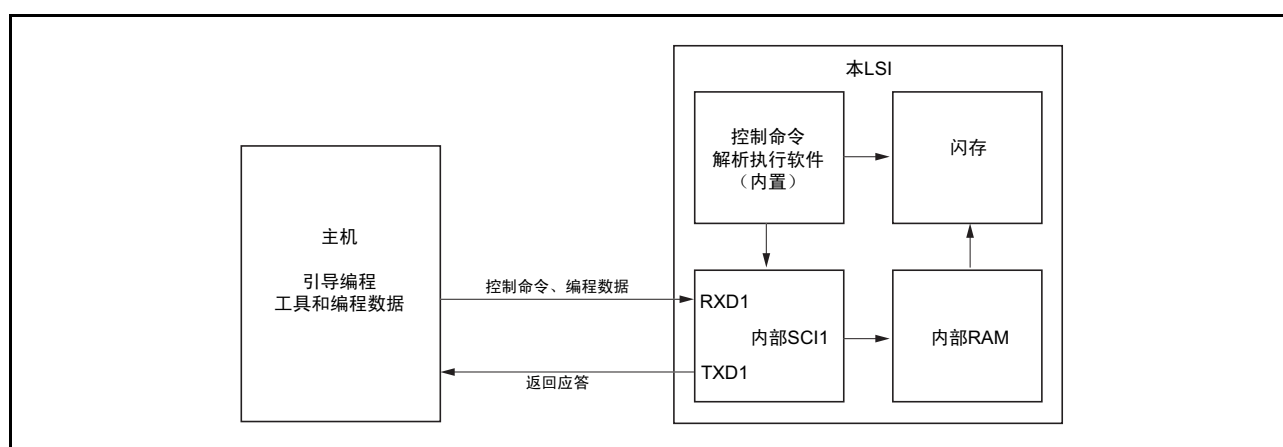


图 23.6 引导模式时的系统构成图

#### (1) 主机的 SCI 接口设定

如果启动引导模式，本 LSI 就测量主机连续发送来的异步 SCI 通信数据（H'00）的 Low 期间。必须将此时的 SCI 发送 / 接收格式设定为“8 位数据、1 个停止位和无奇偶校验”。本 LSI 根据测量的 Low 期间计算主机的发送位速率，并将位调整结束信号（1 字节的 H'00）发送给主机，主机必须在确认已正常接收到此调整结束信号（H'00）后，向本 LSI 发送 1 字节的 H'55。如果不能正常接收，就必须重新启动（复位）引导模式，进行上述操作。根据主机发送的位速率和本 LSI 的系统时钟的频率，主机和本 LSI 的位速率会产生误差，所以为了使 SCI 正常运行，必须将主机的传送位速率设定为 9,600bps 或者 19,200bps。

主机的传送位速率和本 LSI 的位速率能自动匹配的系统时钟的频率如表 23.8 所示，必须在此系统时钟的范围内启动引导模式。另外，在引导模式中，不支持各内部时钟的分频比为  $\times 1/3$  倍的设定。



图 23.7 位速率的自动匹配

表 23.8 本 LSI 能自动匹配的系统时钟的频率

主机的位速率	本 LSI 的位速率能自动匹配的外围时钟 (Pφ) 频率
9,600bps	10 ~ 40MHz
19,200bps	10 ~ 40MHz

【注】 在引导模式中，不支持各内部时钟的分频比为  $\times 1/3$  倍的设定。

## (2) 状态转移图

启动引导模式后的状态转移概要如图 23.8 所示，有关引导模式的详细内容请参照“23.9.1 引导模式的标准串行通信接口规格”。

### 1. 位速率的匹配

在启动引导模式后进行主机和 SCI 接口的位速率匹配。

### 2. 等待查询选择命令

将查询用户 MAT 容量、用户 MAT 结构、MAT 起始地址和支持信息等时所需要的信息发送到主机。

### 3. 自动擦除全部的用户 MAT 和用户引导 MAT

如果在查询选择结束后发送编程 / 擦除的状态转移命令，就自动擦除全部的用户 MAT 和用户引导 MAT。

### 4. 等待编程 / 擦除命令

- 如果接收到“编程选择命令”，就转移到编程数据的等待状态。必须在编程命令后面连续发送编程的起始地址和编程数据，当编程结束时，必须将编程的起始地址设定为 H'FFFFFFF 后发送。据此从编程数据的等待状态返回到编程 / 擦除命令的等待状态。
- 如果接收到“擦除选择命令”，就转移到擦除块数据的等待状态。必须在擦除命令后面连续发送擦除块号，当擦除结束时，必须将擦除块号设定为 H'FF 后发送。据此从擦除块数据的等待状态返回到编程 / 擦除命令的等待状态。另外，在引导模式中，当编程后不需启动复位而只要改写特定块时，必须进行擦除。当通过 1 次操作就能完成编程时，因为在转移到编程 / 擦除 / 其他命令的等待状态之前已将全部块擦除，所以不需要此擦除操作。
- 除编程 / 擦除以外，还有用户 MAT/ 用户引导 MAT 的校验和命令、用户 MAT/ 用户引导 MAT 的空白检查（擦除检查）命令、用户 MAT/ 用户引导 MAT 的存储器读取命令以及当前状态的信息取得命令。

必须注意：用户 MAT/ 用户引导 MAT 的存储器读取只能在自动擦除全部的用户 MAT/ 用户引导 MAT 后，读已被编程的数据。

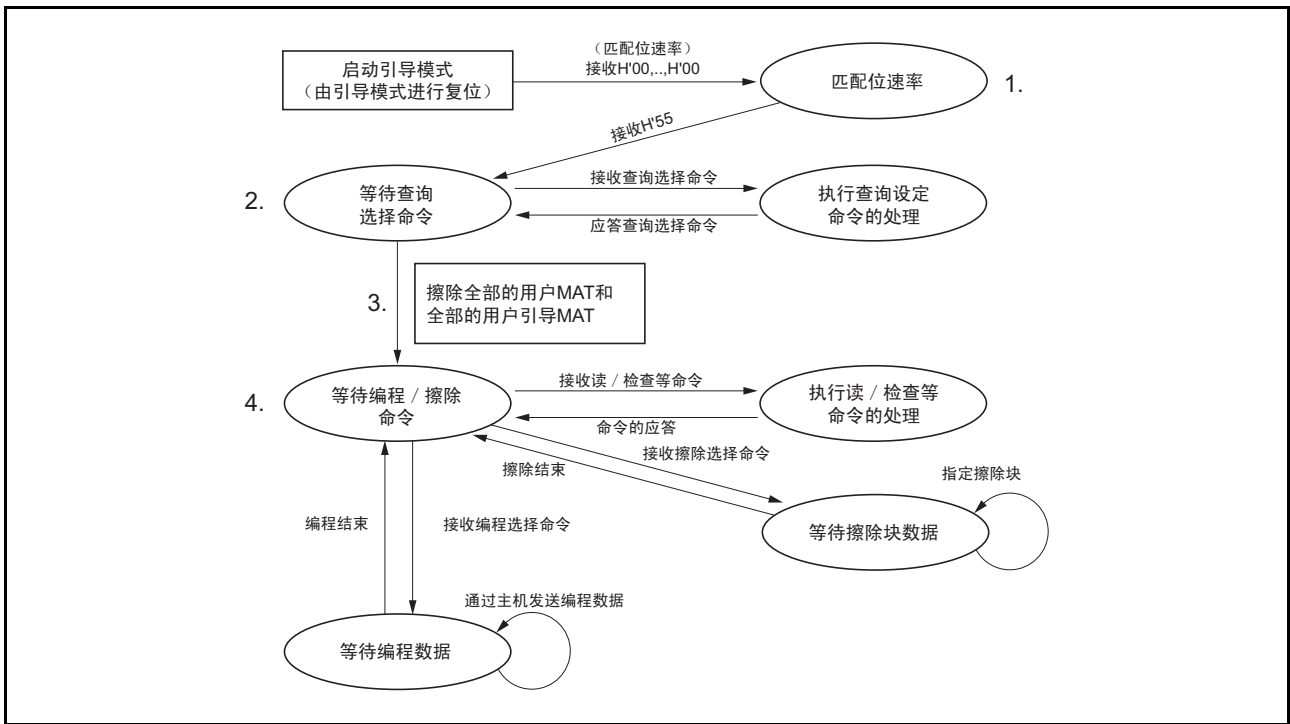


图 23.8 引导模式状态转移的概略图

### 23.5.2 用户编程模式

能在用户编程模式中编程 / 擦除用户 MAT（不能进行用户引导 MAT 的编程 / 擦除）。

预先将内部程序下载到单片机内，然后进行编程 / 擦除。

概略流程如图 23.9 所示。

另外，因为在编程 / 擦除处理中闪存内部加有高电压，所以不能在编程 / 擦除处理中进行复位，否则有可能损伤甚至破坏闪存。如果错误地进行了复位，必须在长于通常的 100 $\mu$ s 复位输入期间后进行复位解除。

有关编程步骤和擦除步骤，请分别参照后述的“(2) 用户编程模式的编程步骤”和“(3) 用户编程模式的擦除步骤”。

有关使用 FTDAR 寄存器，将编程 / 擦除程序分别下载到内部 RAM 区，进行擦除和编程的重复处理的概略，说明在“(4) 用户编程模式的擦除 / 编程步骤”。

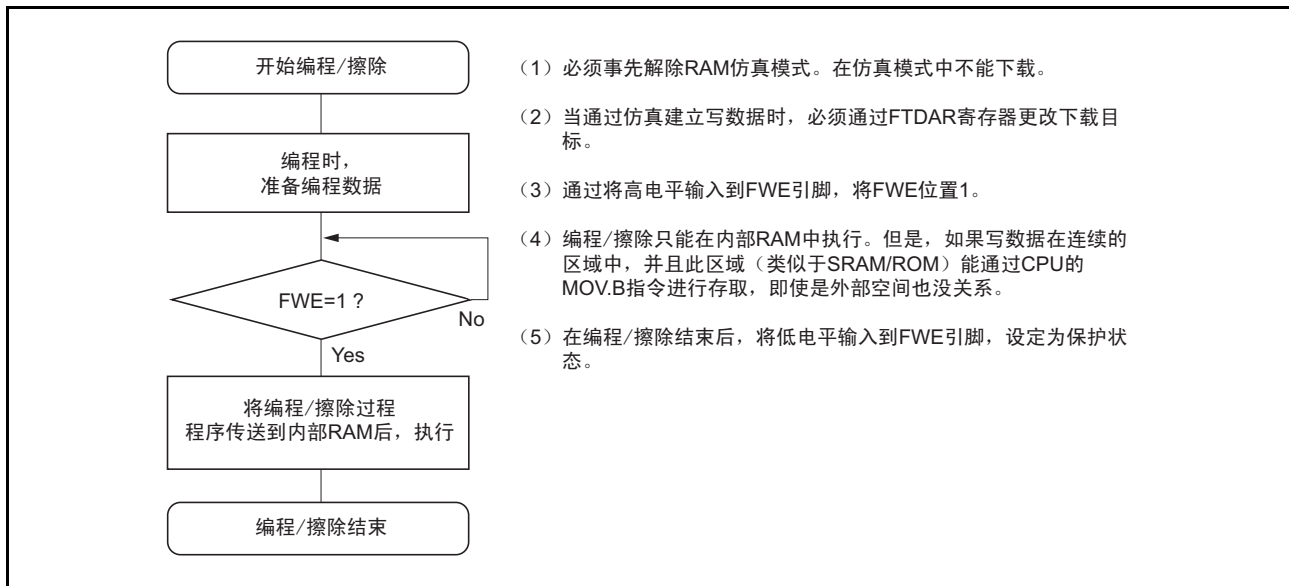


图 23.9 编程 / 擦除的概略流程

## (1) 进行编程 / 擦除时的内部 RAM 的地址映像

必须在内部 RAM 中执行用户建立的下载请求、编程 / 擦除的步骤和结果判断等一部分过程程序。另外，由于被下载的内部程序全部存放在内部 RAM 中，所以为了使这些程序不重叠，必须注意内部 RAM 的区域管理。

被下载的程序区如图 23.10 所示。

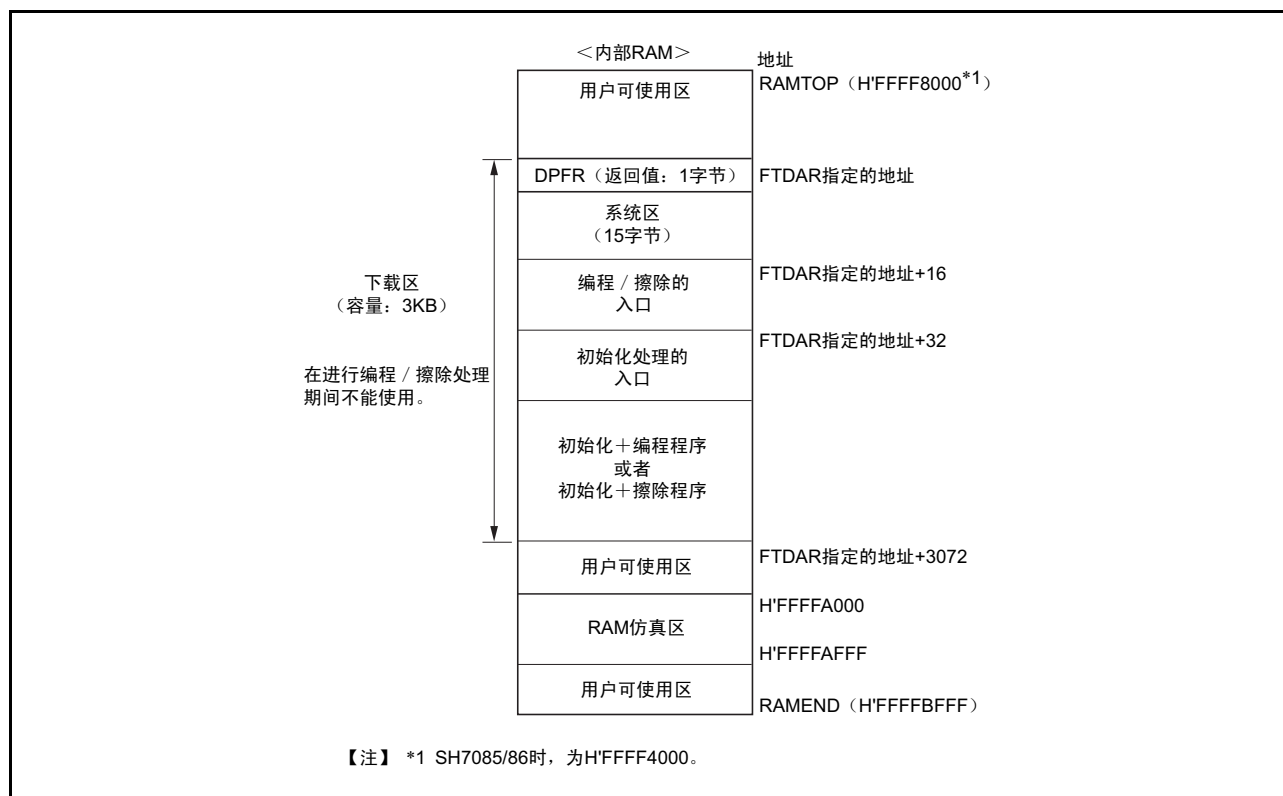


图 23.10 下载后的内部 RAM 映像

## (2) 用户编程模式的编程步骤

下载、初始化和编程的步骤如图 23.11 所示。

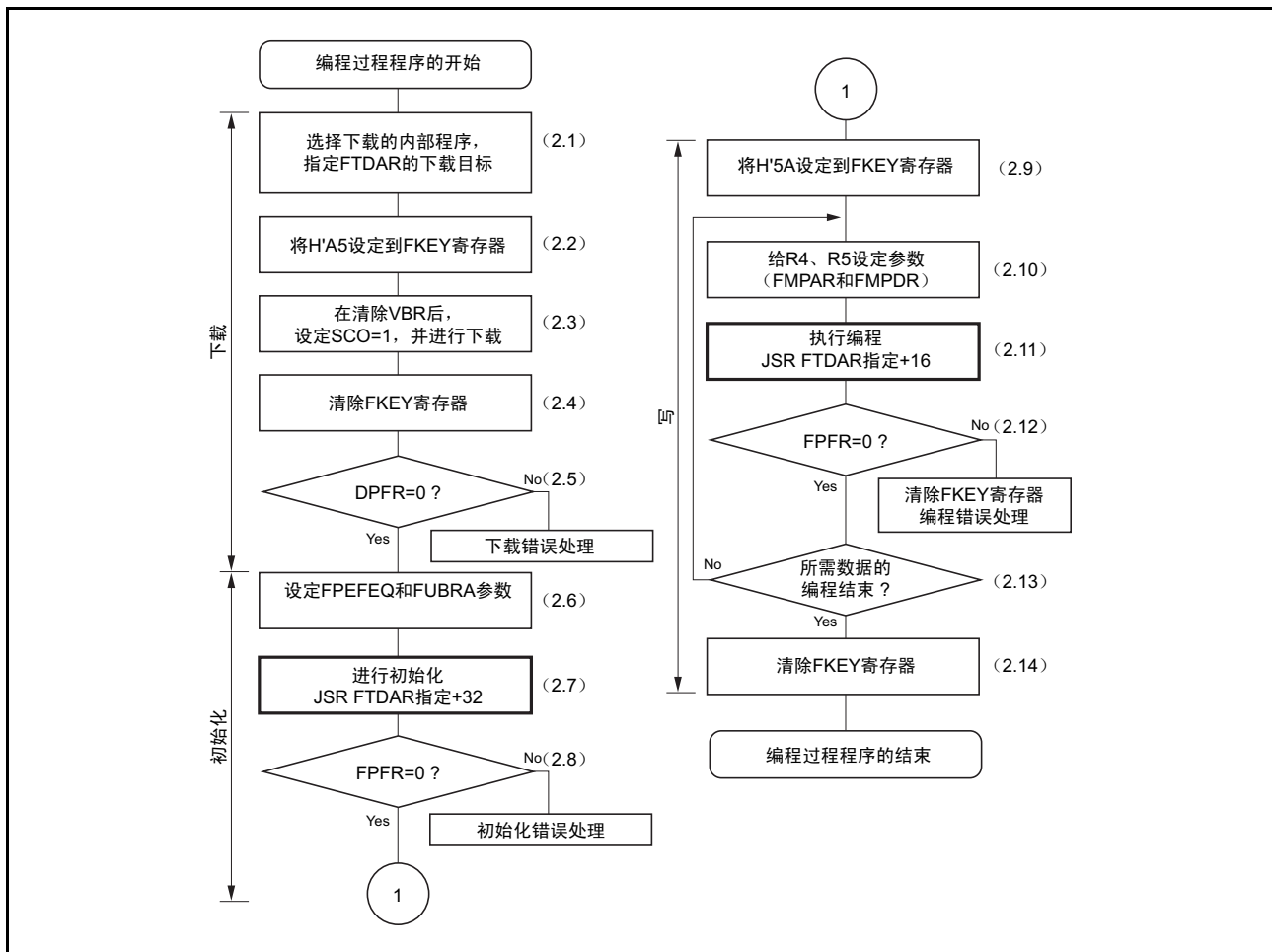


图 23.11 编程步骤

在此说明编程步骤的详细内容。必须在编程对象的闪存以外的区域执行过程程序，尤其是为了下载而将 FCCS 寄存器的 SCO 位设定为 1 的部分，必须在内部 RAM 中执行。另外，通过频率控制寄存器（FRQCR）设定的内部时钟（I $\phi$ ）、总线时钟（B $\phi$ ）、外围时钟（P $\phi$ ）的分频比都为 1/4（初始值）。

编程 / 擦除的程序下载结束，并且将 SCO 位清 0 后，就可以将频率控制寄存器（FRQCR）的设定改为任意的值。

可单步执行用户过程程序的区域（内部 RAM、用户 MAT、外部空间等）表示在“23.9.2 过程程序或者编程数据的可保存区”中。

以下说明的前提是：已擦除用户 MAT 中的编程对象区，并且编程数据也已准备在连续区。如果还没有擦除，必须在编程前擦除。

在 1 次编程处理中进行 128 字节的编程。当超过 128 字节时，以 128 字节为单位更新编程目标地址 / 编程数据的参数，反复进行编程。

当未 128 字节时，需要填写无效数据补齐 128 字节。如果填写的无效数据为 H'FF，就能缩短编程的处理时间。

(2.1) 选择下载的内部程序以及指定下载的目标地址。

如果将 FPCS 寄存器的 PPVS 位设定为 1，就选择编程程序。

不能选择多个编程 / 擦除程序，否则就不下载而由 DPF<sub>R</sub> 参数的源选择检测位 (SS) 报告下载错误。通过 FTDAR 寄存器指定下载目标的起始地址。

(2.2) 将 H'A5 写到 FKEY 寄存器。

如果为了保护而未将 H'A5 写到 FKEY 寄存器，就不能给下载请求的 SCO 位写 1。

(2.3) 设定 VBR 寄存器并给 FCCS 寄存器的 SCO 位写 1，进行下载。

在设定 SCO 位之前，必须将 VBR 寄存器清除为 H'84000000。

为了给 SCO 位写 1，需要满足以下的全部条件：

1. 已解除 RAM 仿真模式
2. 已将 H'A5 写到 FKEY 寄存器。
3. 已在内部 RAM 中进行了 SCO 位的写操作。

因为当 SCO 位为 1 时就自动开始下载，当返回到用户的过程程序时 SCO 位就被清 0 (SCO=0)，所以在用户过程程序中不能确认 SCO=1。

因为只能通过 DPF<sub>R</sub> 参数的返回值确认下载结果，所以在 SCO=1 之前，必须将 FTDAR (作为 DPF<sub>R</sub> 参数的) 所指定的内部 RAM 的起始 1 字节设定为返回值以外的值 (H'FF 等)，以免发生误判断。

在进行下载时，因为单片机的内部处理进行如下所示的伴有存储体转换的特殊中断处理，所以需要将 VBR 设定为 H'84000000。另外，必须在设定 SCO=1 的指令后面立刻执行 4 条 NOP 指令。

1. 将用户 MAT 空间转换到内部程序保存区。
2. 在检查下载程序的选择条件和 FTDAR 的指定地址等后，对 FTDAR 指定的内部 RAM 进行传送处理。
3. 将 FPCS 寄存器、FECS 寄存器和 FCCS 寄存器的 SCO 位清 0。
4. 给 DPF<sub>R</sub> 参数设定返回值。
5. 在将内部程序保存区返回到用户 MAT 空间后，返回到用户过程程序。

在下载结束并返回到用户过程程序后，能重新设定 VBR。

有关下载中的注意事项如下所述：

在下载处理中 CPU 的通用寄存器的值被保存。

不能在下载处理中发生中断。有关下载和中断请参照“23.8.2 执行编程 / 擦除过程中的中断”。

因为最多使用 128 字节的堆栈区，所以在使 SCO=1 之前，至少必须确保 128 字节的堆栈区。

如果在下载中通过 DMAC 或者 DTC 存取闪存，就不能保证运行，所以必须注意：不能通过 DMAC 或者 DTC 进行存取。

(2.4) 将 FKEY 寄存器清除为 H'00，进行保护。

(2.5) 检查 DPF<sub>R</sub> 参数的值，确认下载结果。

建议使用以下方法确认下载结果：

1. 检查 DPF<sub>R</sub> 参数 (FTDAR 指定的下载目标起始地址的 1 字节) 的值。如果值为 H'00，就表示下载正常进行，否则就按以下步骤检查不能下载的原因。
2. 如果 DPF<sub>R</sub> 参数的值和下载前的设定值 (H'FF 等) 相同，FTDAR 的下载目标地址的设定可能异常，所以必须确认 FTDAR 的 bit7 (TDER 位)。
3. 如果 DPF<sub>R</sub> 参数的值和下载前的设定值不同，就必须通过 DPF<sub>R</sub> 参数的 bit2 (SS 位) 和 bit1 (FK 位) 确认下载程序的选择或者 FKEY 寄存器的设定是否正常。

(2.6) 给 FPEFEQ 和 FUBRA 参数设定工作频率和用户分支目标进行初始化。

4. 给 FPEFEQ 参数（通用寄存器：R4）设定当前的 CPU 时钟频率。FPEFEQ 参数的可设定范围请参照“28.3.1 时钟时序”。

如果将频率设定在此范围以外，就由初始化程序的 FPFR 参数报告错误并且不进行初始化。频率的设定方法请参照“23.4.3 编程 / 擦除的接口参数”的“闪存编程 / 擦除的频率控制参数（FPEFEQ: CPU 的通用寄存器 R4）”的说明。

5. 给 FUBRA 参数（通用寄存器：R5）设定用户分支目标的起始地址。

当不需要用户分支处理时，必须给 FUBRA 置 0。

当进行用户分支时，必须在编程对象的闪存以外的区域执行转移目标，并且不能设定到下载内部程序的区域。

必须使用 RTS 指令从用户分支处理返回到编程处理。

请参照“23.4.3 编程 / 擦除的接口参数”的“闪存用户分支地址的设定参数（FUBRA: CPU 的通用寄存器 R5）”的说明。

### (2.7) 执行初始化

在下载编程程序时，初始化程序一起被下载到内部 RAM。因为在从 FTDAR 设定的下载起始地址 +32 字节开始的区域中有初始化程序的入口点，所以必须通过以下方法调用子程序进行初始化。

MOV.L	#DLTOP+32,R1	;	将入口地址设定到 R1
JSR	@R1	;	调用初始化程序
NOP			

1. 在初始化程序中保存 R0 以外的通用寄存器。
2. R0 是 FPFR 参数的返回值。
3. 因为在初始化程序中使用堆栈区，所以必须预先在 RAM 中至少确保 128 字节的堆栈区。
4. 能在执行初始化程序中接受中断，但是不能破坏内部 RAM 中的程序保存区、堆栈区和寄存器的值。

(2.8) 判断初始化程序的返回值 FPFR（通用寄存器 R0）。

(2.9) 必须给 FKEY 寄存器设定 H'5A 以便能对用户 MAT 进行编程。

(2.10) 设定编程所需要的参数。

将用户 MAT 的编程目标起始地址（FMPAR）和编程数据保存区的起始地址（FMPDR）分别设定给通用寄存器 R5 和 R4。

1. FMPAR 的设定

因为 FMPAR 指定编程起始地址，所以当指定用户 MAT 区以外的地址时，即使执行编程程序也不进行编程，而由返回值参数 FPFR 报告错误。另外，因为是以 128 字节为单位，所以低 8 位（MOA7 ~ MOA0）需要为 H'00 或者 H'80 的 128 字节的边界。

2. FMPDR 的设定

当编程数据的保存目标在闪存中时，即使执行编程执行程序也不进行编程，而由 FPFR 参数报告错误。此时必须暂时先将编程数据传送到内部 RAM，然后进行编程。



## (2.11) 执行编程处理

因为在从 FTDAR 指定的下载目标起始地址 +16 字节开始的区域中有编程程序的入口点，所以必须通过以下方法调用子程序进行编程处理。

MOV.L	#DLTOP+16,R1	;	将入口地址设定到 R1
JSR	@R1	;	调用编程程序
NOB			

1. 在编程程序中保存 R0 以外的通用寄存器。
2. R0 是 FPFR 参数的返回值。
3. 因为在编程程序中使用堆栈区，所以必须预先在 RAM 中至少确保 128 字节的堆栈区。

## (2.12) 判断编程程序的返回值 FPFR（通用寄存器 R0）。

## (2.13) 判断所需数据的编程是否已结束。

当编程超过 128 字节的数据时，以 128 字节为单位更新 FMPAR、FMPDR 的设定，然后重复上述 (2.10) ~ (2.13) 的处理。必须正确进行编程目标地址的 128 字节的递增和编程数据指针的更新。如果对已被编程的地址进行重复编程，不仅会引起编程错误，还会对闪存造成损伤。

## (2.14) 必须在编程结束后清除 FKEY 寄存器并加软件保护。

如果在结束对用户 MAT 的编程后立刻通过上电复位重新启动，必须设定长于通常的 100 $\mu$ s 复位期间 ( $\overline{\text{RES}}=0$  的期间)。

(3) 用户编程模式的擦除步骤

下载、初始化和擦除的步骤如图 23.12 所示。

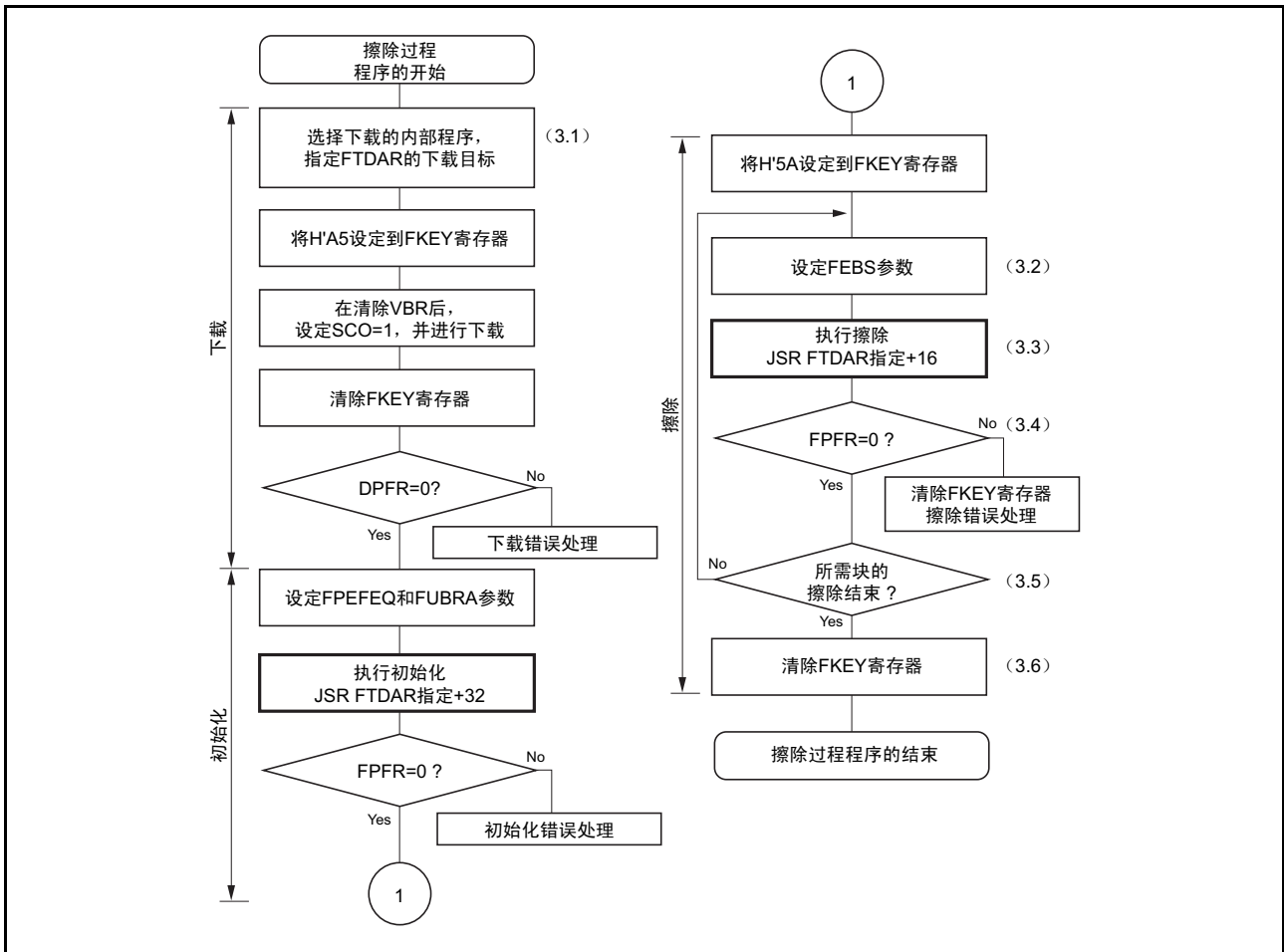


图 23.12 擦除步骤

在此说明擦除步骤的详细内容。必须在擦除对象的闪存以外的区域执行过程程序。尤其是为了下载而将 FCCS 寄存器的 SCO 位设定为 1 的部分，必须在内部 RAM 中执行。另外，通过频率控制寄存器（FRQCR）设定的内部时钟（I $\phi$ ）、总线时钟（B $\phi$ ）、外围时钟（P $\phi$ ）的分频比都为 1/4（初始值）。

编程 / 擦除的程序下载结束，并且将 SCO 位清 0 后，就可以将频率控制寄存器（FRQCR）的设定改为任意的值。

可单步执行用户过程程序的区域（内部 RAM、用户 MAT、外部空间等）表示在“23.9.2 过程程序或者编程数据的可保存区”中。

有关被下载的内部程序的区域，请参照图 23.10 的下载后的内部 RAM 映像。

在 1 次擦除处理中进行 1 个分割块的擦除，有关块分割请参照图 23.4。当进行 2 块以上的擦除时，先更新擦除块号再重复擦除。

(3.1) 选择下载的内部程序以及指定下载的目标地址。

将 FECS 寄存器的 EPVB 位设定为 1。

不能选择多个编程 / 擦除程序，否则就不下载而由 DPFR 参数的源选择错误检测位 (SS) 报告下载错误。

通过 FTDAR 寄存器指定下载目标的起始地址。

设定 FKEY 寄存器以后的下载和初始化等过程和编程步骤相同，请参照“23.5.2 (2) 用户编程模式的编程步骤”。

设定擦除程序参数以后的步骤如下所示：

(3.2) 设定擦除时所需要的 FEBS 参数。

将用户 MAT 的擦除块号设定到闪存擦除块的选择参数 FEBS (通用寄存器 R4)。如果设定用户 MAT 分割块号以外的值，即使执行擦除处理程序也不擦除，而由返回值参数 FPFR 报告错误。

(3.3) 执行擦除处理

和编程相同，因为在从 FTDAR 指定的下载目标起始地址 +16 字节开始的区有擦除程序的入口点，所以必须通过以下方法调用子程序进行擦除处理。

MOV.L	#DLTOP+16,R1	;	将入口地址设定到 R1
JSR	@R1	;	调用擦除程序
NOP			

1. 在擦除程序中保存 R0 以外的通用寄存器。
2. R0 是 FPFR 参数的返回值。
3. 因为在擦除程序中使用堆栈区，所以必须预先在 RAM 中至少确保 128 字节的堆栈区。

(3.4) 判断擦除程序的返回值 FPFR (通用寄存器 R0)。

(3.5) 判断所需块的擦除是否已结束。

当擦除多个块时，更新 FEBS 参数的设定，然后重复上述 (3.2) ~ (3.5) 的处理。能对已擦除的块进行擦除。

(3.6) 必须在结束擦除后清除 FKEY 寄存器并加软件保护。

如果在结束用户 MAT 的擦除后立刻通过上电复位重新启动，必须设定长于通常的 100μs 通常复位期间 (RES=0 的期间)。

#### (4) 用户编程模式的擦除 / 编程步骤

通过更改 FTDAR 寄存器中的下载目标的内部 RAM 地址，能将擦除程序和编程程序事先下载到各自的内部 RAM 区。

重复进行 RAM 仿真、擦除 / 编程时的使用例子如图 23.13 所示。

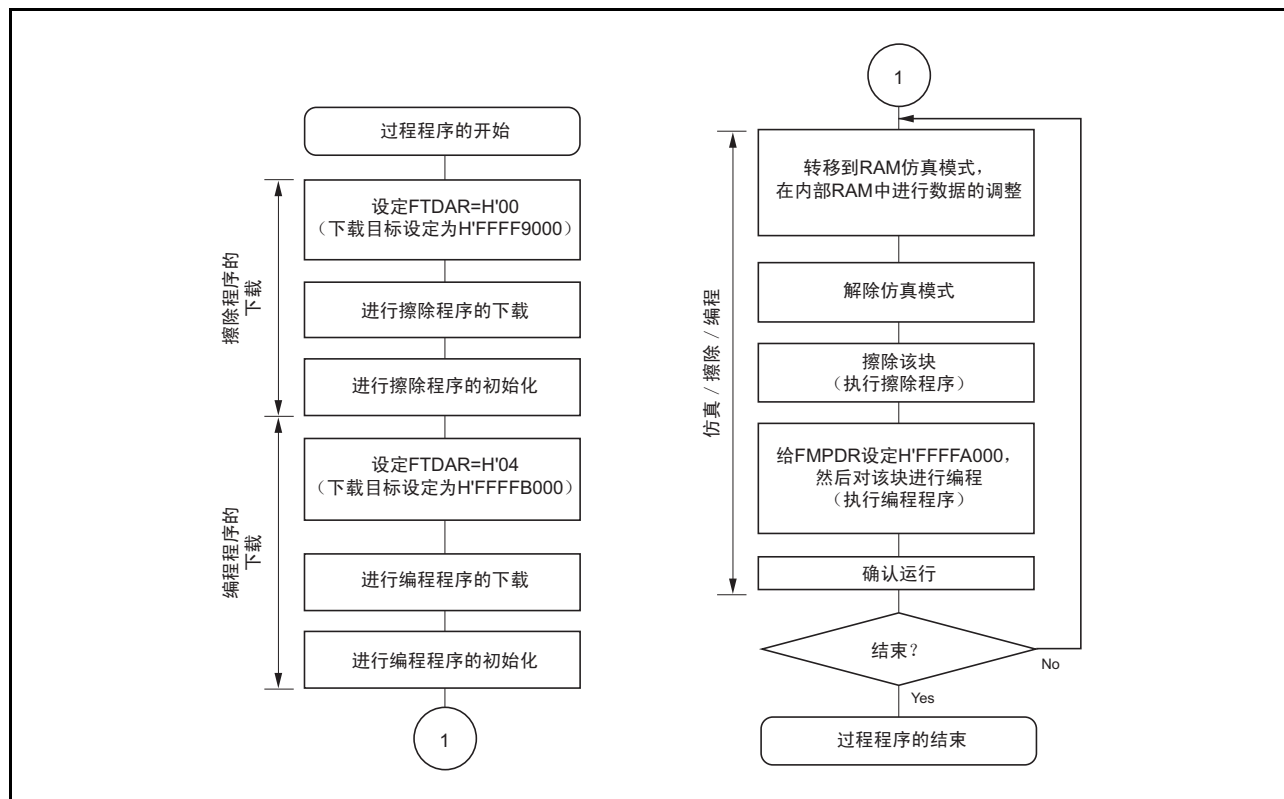


图 23.13 RAM 仿真、擦除 / 编程的重复例子（概要）

在本例中，为了进行 RAM 仿真，将擦除 / 编程程序下载到 H'FFFA000 ~ H'FFFAFFF 以外的区域。另外，只在最初进行 1 次下载和初始化。

在执行本例的步骤时，必须注意以下事项：

1. 必须注意不要因重复设定而破坏内部 RAM 区。

在内部 RAM 中除了 RAM 仿真区、擦除程序区、编程程序区以外，还有用户建立的过程程序区、作业区、堆栈区等，所以不能破坏这些区域。

2. 必须进行擦除程序和编程程序的初始化。

必须在擦除程序和编程程序中进行设定 FPEFEQ 参数和 FUBRA 参数的初始化。初始化的入口地址必须设定为擦除程序的下载地址 +32（本例为 H'FFFF9020）和编程程序的下载地址 +32（本例为 H'FFFF9820）。

### 23.5.3 用户引导模式

本 LSI 有通过与用户编程模式和引导模式所不同的模式引脚设定进行启动的用户引导模式，能实现与使用内部 SCI 的引导模式所不同的用户任意的引导模式。

在用户引导模式中，只有用户 MAT 才能编程 / 擦除，而用户引导 MAT 的编程 / 擦除必须在引导模式或者编程器模式中进行。

(1) 用户引导模式的启动

用于用户引导模式启动的模式引脚设定如表 23.1 所示。

如果以用户引导模式开始复位，就执行闪存相关寄存器的检查程序，此程序使用的 RAM 容量约为 1.2KB（从地址 H'FFFF9800 开始）和堆栈使用的 4 字节（从地址 H'FFFAFFC 开始）。在此程序的执行期间，不接受 NMI 和其他中断，并且也不能使用 AUD。当以 40MHz 的内部频率运行时，此执行时间约为 100μs。

此后，从用户引导 MAT 中的复位向量的执行起始地址开始处理，因为此时执行 MAT 为用户引导 MAT，所以闪存 MAT 选择寄存器 FMATS 的设定为 H'AA。

(2) 在用户引导模式中进行用户 MAT 的编程

在用户引导模式中进行用户 MAT 编程时，需要追加以下的步骤：

- 通过 FMATS 寄存器从用户引导 MAT 选择状态转换到用户 MAT 选择状态
  - 在编程结束后，再次从用户 MAT 选择状态返回到用户引导 MAT 选择状态
- 在用户引导模式中进行用户 MAT 编程的步骤如图 23.14 所示。

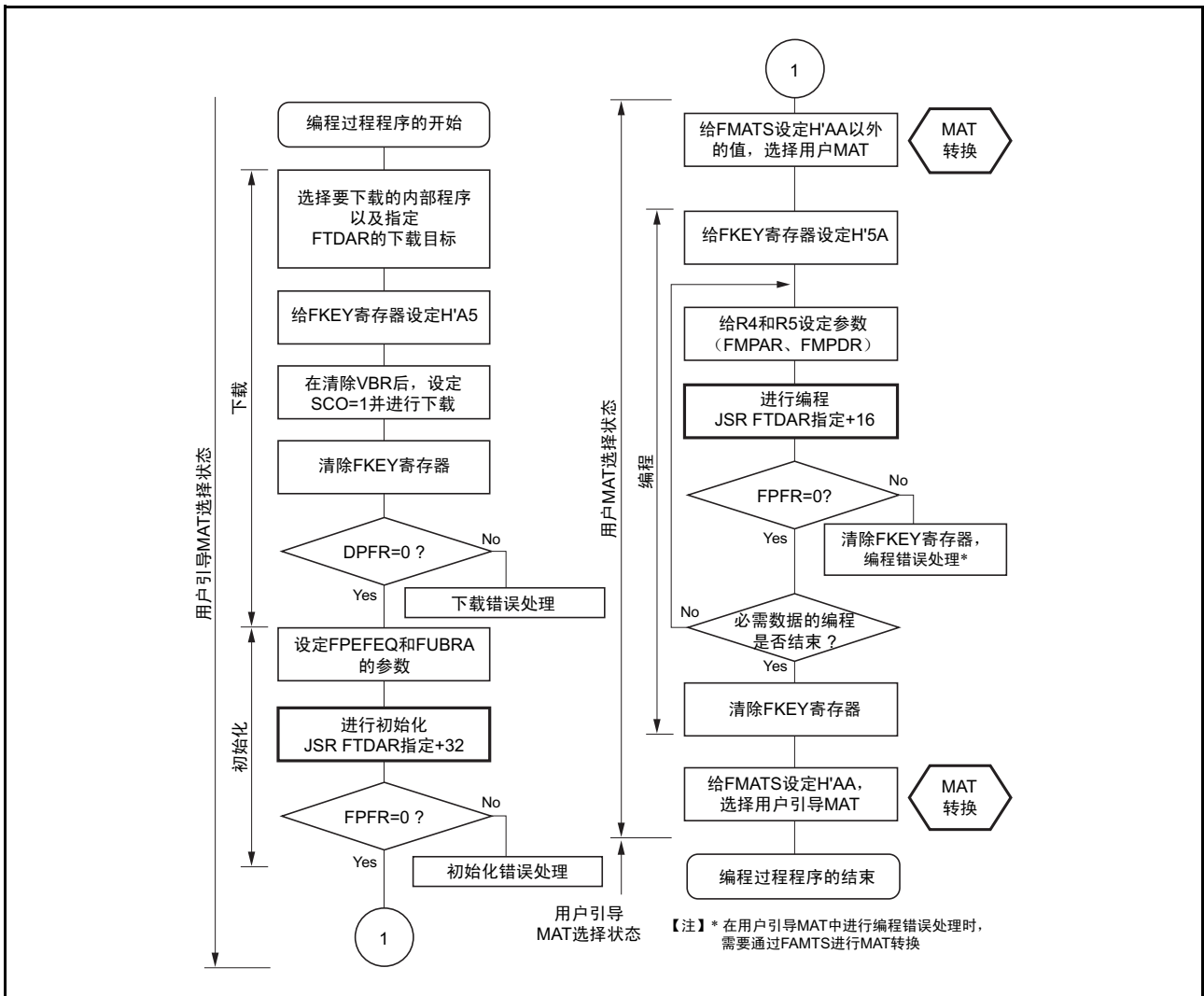


图 23.14 在用户引导模式中进行用户 MAT 编程的步骤

如图 23.14 所示，用户编程模式和用户引导模式的编程步骤的不同在于是否进行 MAT 转换。

在用户引导模式中，能看到闪存空间中的用户引导 MAT，而用户 MAT 隐藏在背后。只在用户 MAT 的编程处理期间，用户 MAT 和用户引导 MAT 进行转换。在编程处理中，用户引导 MAT 为隐藏状态，而用户 MAT 为编程状态，所以需要在闪存以外的区域执行程序。如果编程处理结束，就再次进行 MAT 转换，返回到最初的状态。

能通过给 FMATS 寄存器写规定的值，实现 MAT 的转换，但是在 MAT 转换完全结束前不能存取 MAT，并且如果在 MAT 转换中发生中断，就会出现不确定从哪个 MAT 读取中断向量的不稳定状态。有关 MAT 转换，请按照“23.8.1 用户 MAT 和用户引导 MAT 的转换”的说明进行。

除了 MAT 转换以外，编程步骤与用户编程模式相同。

可单步执行用户过程程序的区域（内部 RAM、用户 MAT、外部空间等）表示在“23.9.2 过程程序或者编程数据的可保存区”中。

(3) 在用户引导模式中进行用户 MAT 的擦除

在用户引导模式中进行用户 MAT 擦除时，需要追加以下的步骤：

- 通过 FMATS 寄存器从用户引导 MAT 选择状态转换到用户 MAT 选择状态
  - 在擦除结束后，再次从用户 MAT 选择状态返回到用户引导 MAT 选择状态
- 在用户引导模式中进行用户 MAT 擦除的步骤如图 23.15 所示。

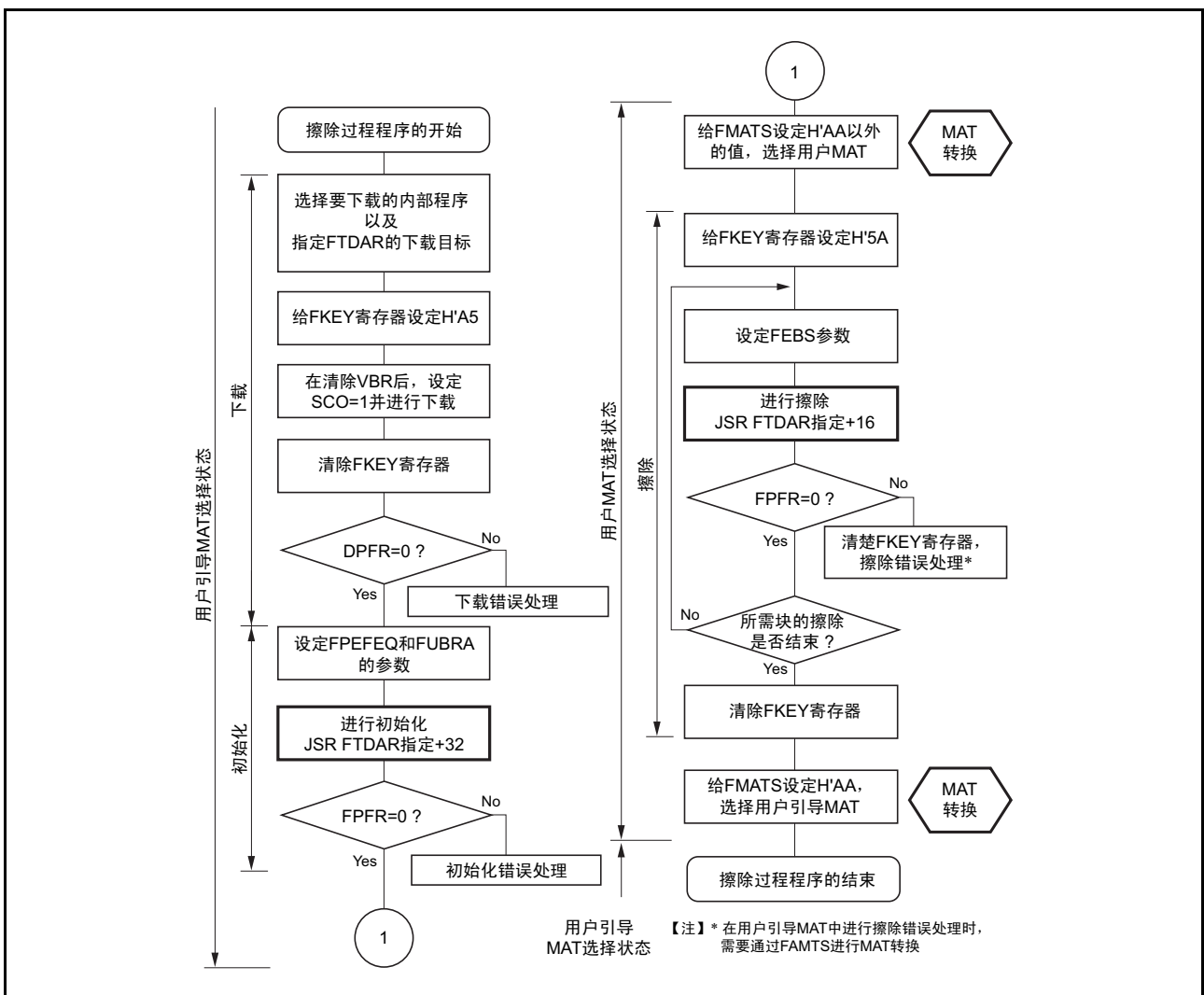


图 23.15 在用户引导模式中进行用户 MAT 擦除的步骤

如图 23.15 所示，用户编程模式和用户引导模式的擦除步骤的不同在于是否进行 MAT 转换。

能通过给 FMATS 寄存器写规定的值，实现 MAT 的转换，但是在 MAT 转换完全结束前不能存取 MAT，并且如果在 MAT 转换中发生中断，就会出现从哪个 MAT 读取中断向量的不稳定状态。有关 MAT 转换，请按照“23.8.1 用户 MAT 和用户引导 MAT 的转换”的说明进行。

除了 MAT 转换以外，擦除步骤与用户编程模式相同。

可单步执行用户过程程序的区域（内部 RAM、用户 MAT、外部空间等）表示在“23.9.2 过程程序或者编程数据的可保存区”中。

## 23.6 保护

闪存的编程 / 擦除保护有硬件保护、软件保护和错误保护共 3 种。

### 23.6.1 硬件保护

硬件保护是指强制禁止或者强制中断闪存编程 / 擦除的状态。能下载内部程序并进行初始化，但是即使启动编程 / 擦除程序也不能进行用户 MAT 的编程 / 擦除，而由 FPCR 参数报告编程 / 擦除的错误。

表 23.9 硬件保护

项目	说明	受保护的功能	
		下载	编程和擦除
FWE 引脚的保护	<ul style="list-style-type: none"> <li>当给 FWE 引脚输入 Low 电平时，就清除 FCCS 寄存器的 FWE 位，变为编程 / 擦除的保护状态。</li> </ul>	—	○
复位和待机的保护	<ul style="list-style-type: none"> <li>在上电复位（包括由 WDT 产生的上电复位）和待机时，对编程 / 擦除的接口寄存器进行初始化，变为编程 / 擦除的保护状态。</li> <li>通过 RES 引脚进行复位时，如果不将 RES 引脚的 Low 电平保持到接通电源后的振荡稳定为止，就不进入复位状态。另外，运行中的复位必须在 AC 特性规定的 RES 脉宽之间将 RES 引脚保持为 Low 电平。不能保证编程 / 擦除运行中的闪存值。此时，必须在擦除后重新进行编程。</li> </ul>	○	○

### 23.6.2 软件保护

软件保护有内部编程 / 擦除程序下载后的保护、通过键代码的保护和通过 RAM 仿真的保护。

表 23.10 软件保护

项目	说明	受保护的功能	
		下载	编程和擦除
SCO 位的保护	<ul style="list-style-type: none"> <li>因将 FCCS 寄存器的 SCO 位清 0 而不能下载编程 / 擦除的程序，所以变为编程 / 擦除的保护状态。</li> </ul>	○	○
FKEY 寄存器的保护	<ul style="list-style-type: none"> <li>如果不给 FKEY 寄存器写键代码，就不能下载和编程 / 擦除。对于下载和编程 / 擦除，需要设定不同的键代码。</li> </ul>	○	○
仿真保护	<ul style="list-style-type: none"> <li>通过将 RAM 仿真保护寄存器（RAMER）的 RAMS 位设定为 1，变为编程 / 擦除保护状态。</li> </ul>	○	○

### 23.6.3 错误保护

错误保护是在闪存的编程 / 擦除中因检测到单片机失控或者没有按规定的编程 / 擦除步骤运行而强制中断编程 / 擦除运行的保护。通过中断编程 / 擦除运行能防止因重复编程或者重复擦除而造成闪存的损伤。

如果在闪存的编程 / 擦除中单片机运行异常，FCCS 寄存器的 FLER 位就被置 1，并转移到错误保护状态，中断编程 / 擦除。

FLER 位的置位条件如下所示：

1. 在编程 / 擦除中已读取闪存的该存储体区时（包括读向量和取指令）
2. 在编程 / 擦除中已执行 SLEEP 指令时（包括软件待机）

只在上电复位时解除错误保护（清除 FLER 位）。

另外，此时必须在长于通常的 100μs 通常复位输入期间后解除复位。因为在编程 / 擦除中闪存加有高电压，有可能在向错误保护状态转移时不能完全去除外加电压，所以需要通过延长复位期间去除外加电压，减少对闪存的损伤。

错误保护状态的状态转移如图 23.16 所示。

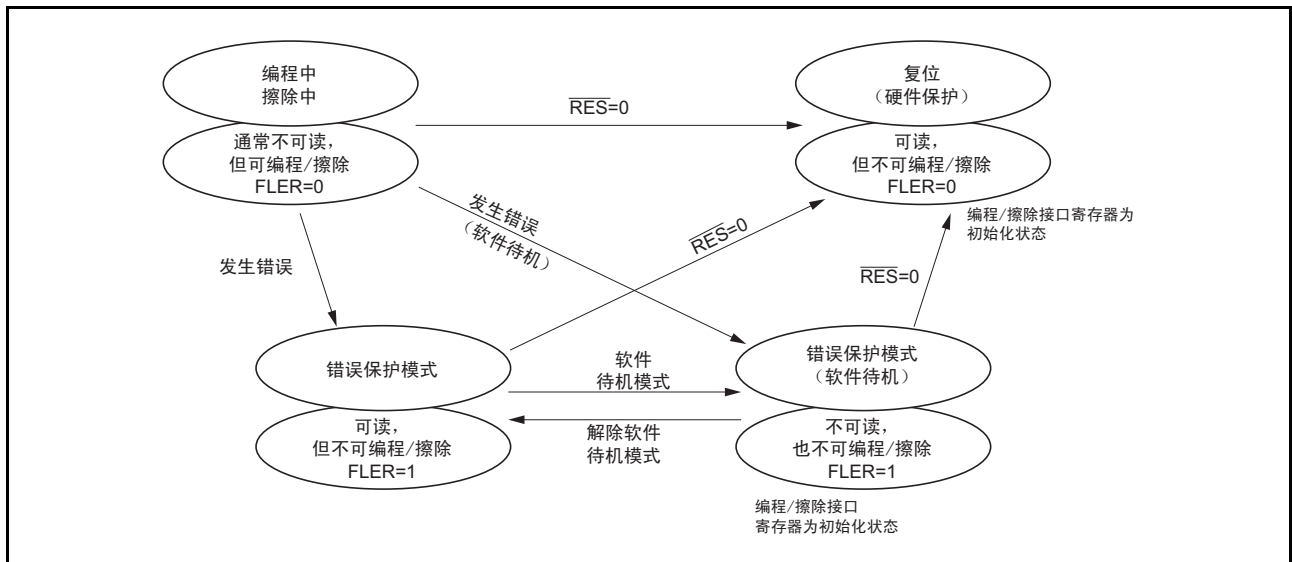


图 23.16 错误保护状态的状态转移图



## 23.7 通过 RAM 对闪存进行仿真

为了用 RAM 对闪存数据的改写进行实时仿真，能将一部分 RAM 和 RAM 仿真寄存器（RAMER）设定的闪存区（用户 MAT）重叠使用。在设定 RAMER 后，能从用户 MAT 区和与其重叠的 RAM 区进行存取。可仿真的模式为用户模式和用户编程模式。

对用户 MAT 的改写进行实时仿真的例子如图 23.17 所示。

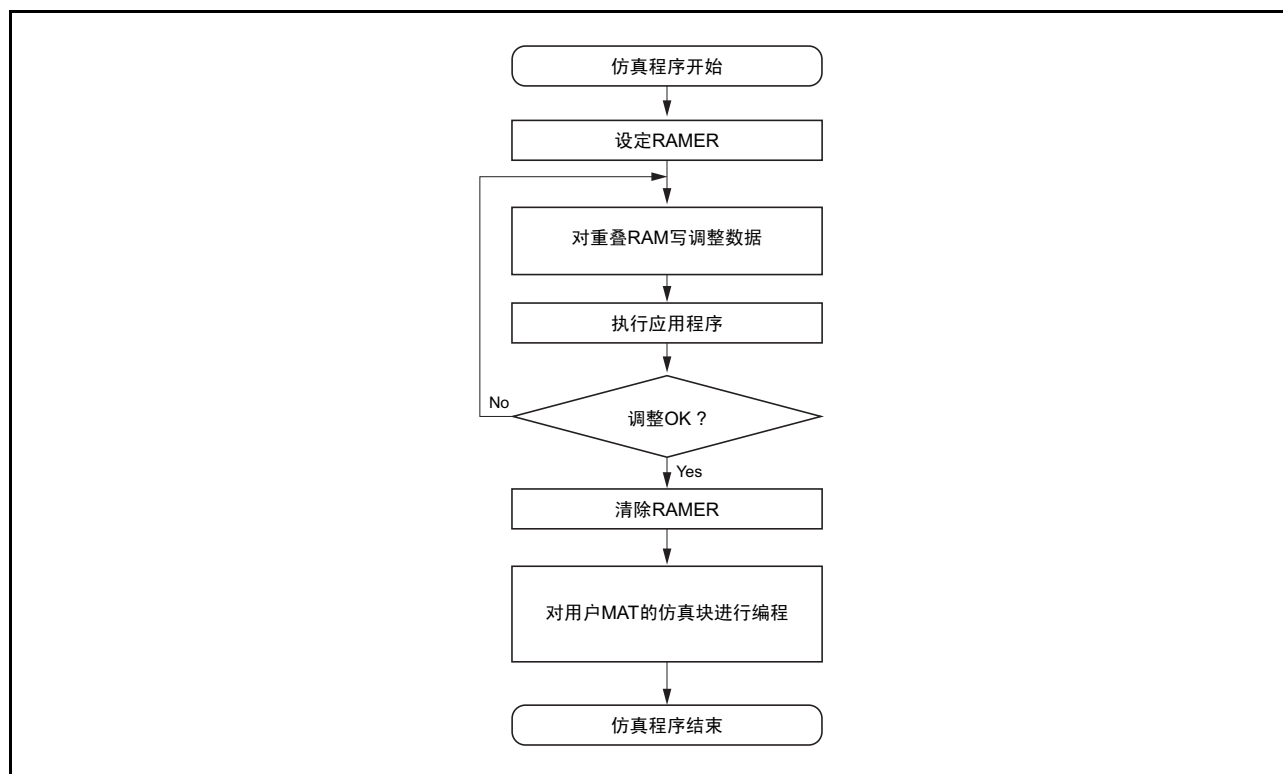


图 23.17 通过 RAM 进行的仿真

使闪存的块区 EB0 重叠的例子如图 23.18 所示。

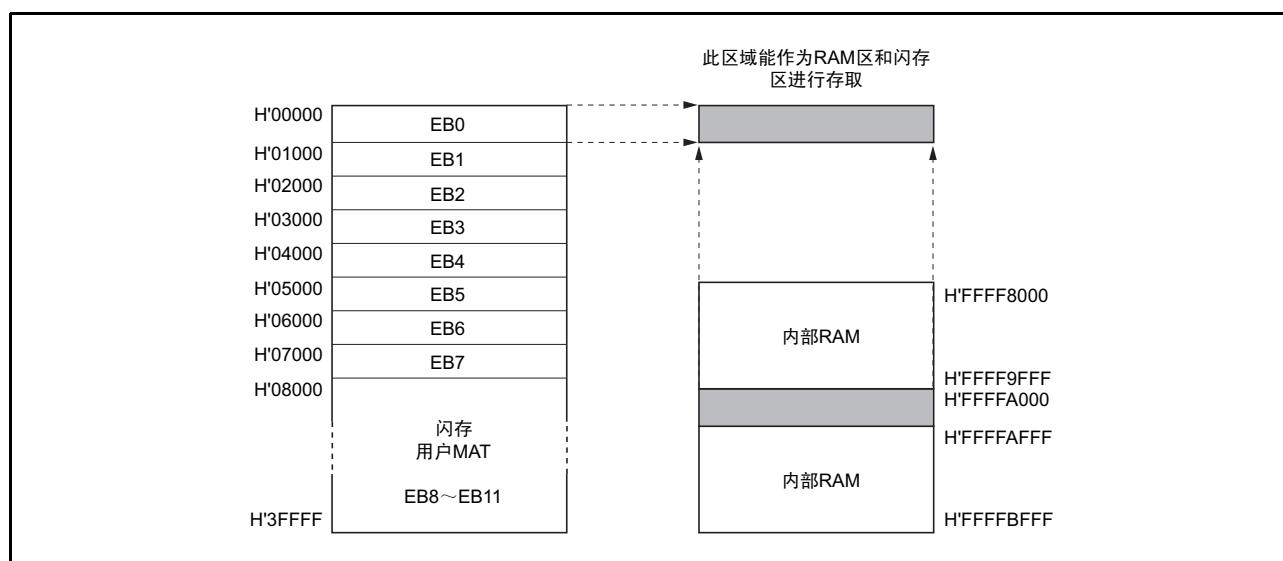


图 23.18 RAM 的重叠运行例子（SH7083（闪存 256KB 版）的情况）

可仿真的闪存区为用户 MAT 的 EB0 ~ EB7 的 8 个区域中的 1 个区域（用 RAMER 寄存器的 RAM2 ~ RAM0 位选择）。

1. 为了使一部分 RAM 与实时改写所需的区域 EB0 重叠，必须将 RAMER 的 RAMS 位设定为 1，并且将 RAM2 ~ RAM0 位设定为 0、0、0。
2. 使用重叠的 RAM，进行实时改写。

对于用户 MAT 的编程 / 擦除，需要执行包含内部程序下载的一系列过程程序。此时，为了使 RAM 的重叠区域和下载的内部程序区不重叠，必须用 FTDAR 寄存器设定下载区。

对用户 MAT 的 EB0 区进行仿真结束后的数据编程的例子如图 23.19 所示。

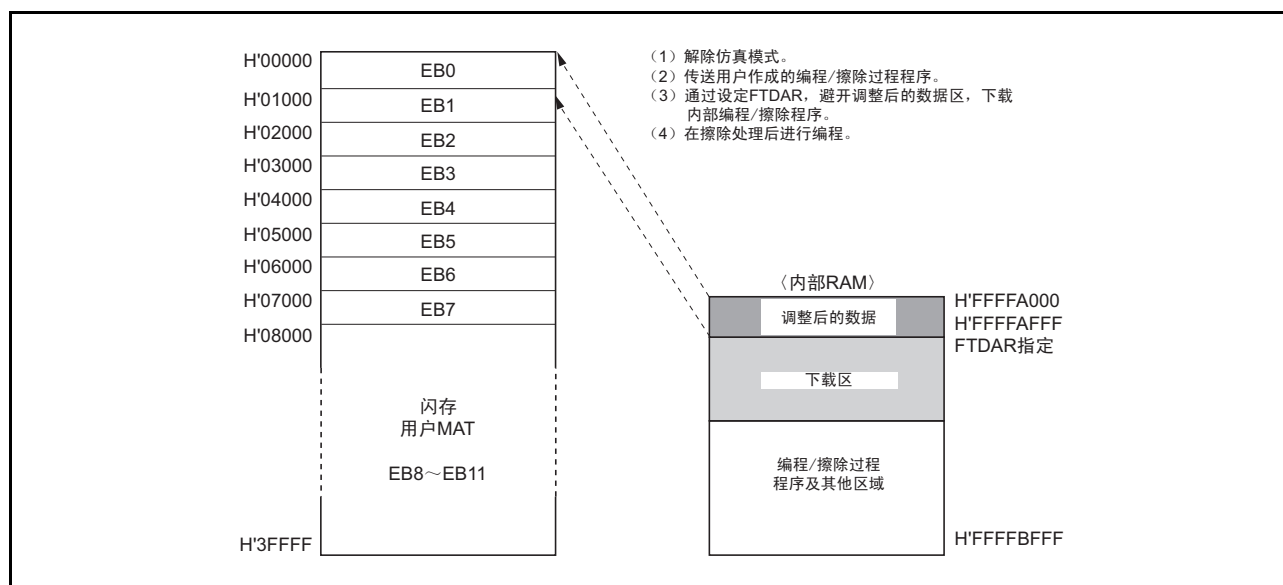


图 23.19 调整结束后的数据编程例子（SH7083（闪存 256KB 版）的情况）

1. 在确定要改写的的数据后，清除 RAMS 位，解除 RAM 的重叠。如果解除仿真模式，仿真保护也被解除。
2. 将用户建立的编程 / 擦除过程程序传送到 RAM。
3. 启动 RAM 中的编程 / 擦除过程程序，将单片机的内部编程 / 擦除程序下载到 RAM。  
此时，为了不使调整结束后的数据区和下载区重叠，必须通过设定 FTDAR 寄存器，指定下载起始地址。
4. 在还没有擦除用户 MAT 的 EB0 区时，必须在擦除后进行编程。必须对编程处理参数 FMPAR 和 FMPDR 指定调整后的数据，然后进行编程处理。

**【注】** 如果将 RAMS 位置 1，就与 RAM2 ~ RAM0 的值无关，闪存的全部块为编程 / 擦除保护状态（仿真保护）。在进行实际编程 / 擦除时，必须清除 RAMS 位。

即使在选择用户引导 MAT 时，也能进行 RAM 仿真，但是用户引导 MAT 的擦除 / 编程只能在引导模式或者编程器模式中进行。

## 23.8 使用注意事项

### 23.8.1 用户 MAT 和用户引导 MAT 的转换

用户 MAT 和用户引导 MAT 能进行相互转换，但是因为它们被分配到相同的 0 地址，所以需要以下的步骤：  
（在转换到用户引导 MAT 的状态下，不能进行编程 / 擦除。用户引导 MAT 的改写必须在引导模式或者编程器模式中进行。）

1. 必须在内部 RAM 中通过 FMATS 寄存器进行 MAT 的转换。  
SH 单片机预取要执行的指令，所以如果在用户 MAT 中的程序执行期间进行 MAT 的转换，就会出现是预取用户 MAT 中的指令码还是预取转换后的用户引导 MAT 中的指令的不稳定操作。
2. 为了在转换后保证 MAT 的存取，必须在改写内部 RAM 中的 FMATS 寄存器后立即执行在内部 RAM 中的 4 条 NOP 指令（这是为了不在转换中存取闪存）。
3. 如果在转换中发生中断，就不能保证哪个存储器 MAT 被存取。  
在 MAT 转换前，必须屏蔽可屏蔽的中断，并且在 MAT 转换中，不能让系统发生 NMI 中断。
4. 必须注意：在 MAT 转换结束后，各种中断的向量表区也被转换。  
在 MAT 转换前后进行同样的中断处理时，或者在不能禁止中断发生时，必须将中断处理程序传送到内部 RAM，并且还必须通过设定 VBR 寄存器，将中断向量表设定到内部 RAM 中。此时，必须注意 VBR 寄存器的更改和中断发生的竞争。
5. 用户 MAT 的存储容量和用户引导 MAT 不同。不能存取超过 12KB 空间的用户引导 MAT，否则就会读取不定值。

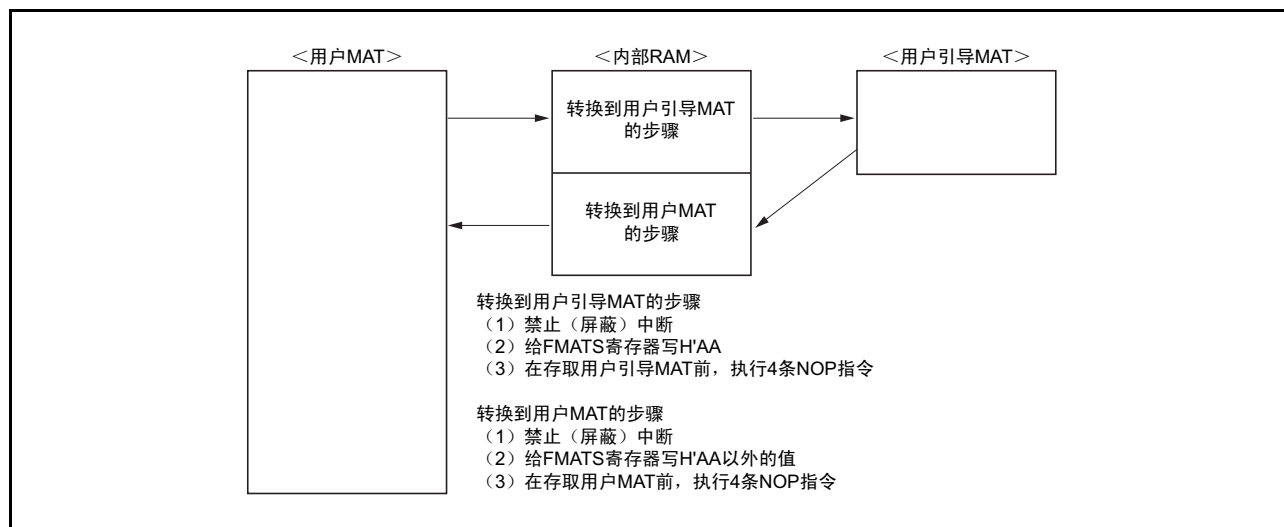


图 23.20 用户 MAT 和用户引导 MAT 的转换

## 23.8.2 执行编程 / 擦除过程中的中断

### (1) 执行内部程序的下载

#### (1.1) VBR 的变更

在下载内部程序前，需要将 VBR 寄存器置为 H'84000000。在设定 H'84000000 以外的值而使用 VBR 的情况下，如果将 VBR 设定为 H'84000000，中断向量表就变为用户 MAT（FMATS≠H'AA 时）或者用户引导 MAT（FMATS=H'AA 时）。

另外，如果变更 VBR 和发生中断出现竞争，就有可能因参照 VBR 变更前或变更后的向量表而发生问题。

因此，在有可能发生和中断竞争的情况下，还必须在用户 MAT 或者用户引导 MAT 的起始部分准备 VBR=H'84000000（初始值）时的参照向量表。

#### (1.2) SCO 下载请求和中断请求

在将 FCCS 寄存器的 SCO 位设定为 1 后下载内部的编程 / 擦除程序时，会产生伴有 MAT 转换的特殊中断。以下说明 SCO 下载请求和中断请求出现竞争时的运行：

##### 1. SCO 下载请求和中断请求的竞争

将 FCCS 寄存器的 SCO 位设定为 1 的指令执行和接受中断的竞争时序如图 23.21 所示。

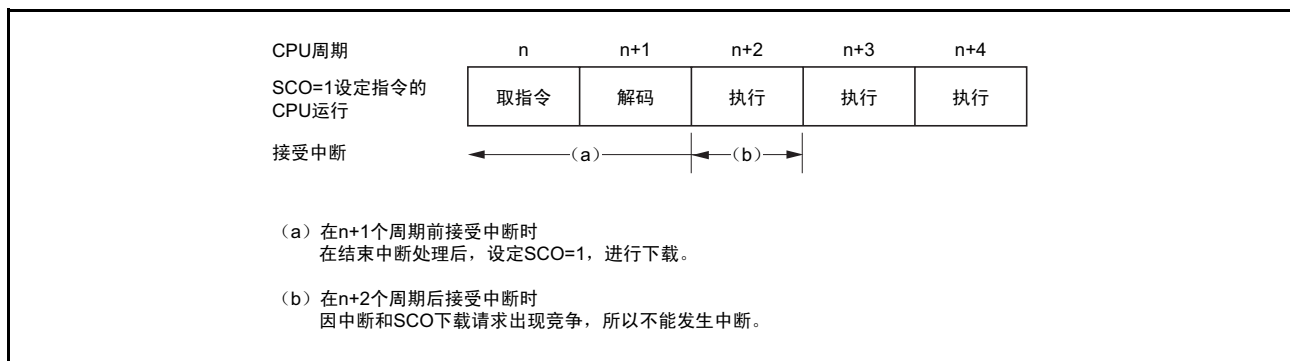


图 23.21 SCO 下载请求和中断请求的竞争时序

##### 2. 在下载中发生的中断请求

不能在进行 SCO 下载中发生中断。

### (2) 编程 / 擦除处理中的中断

能实时处理在下载内部程序的编程 / 擦除中发生的中断，但是有以下的限制和注意事项：

- 对于正在编程 / 擦除中的闪存，不能存取用户 MAT/ 用户引导 MAT。必须将中断向量表和中断处理程序准备在如内部 RAM 或者外部存储器中。即使在中断处理程序中也不能存取正在编程 / 擦除中的闪存。如果读闪存，就不能保证读取的值。另外，如果在编程 / 擦除中存取闪存的该存储体，就转移到错误保护状态，中断编程 / 擦除处理。当存取该存储器以外的区域时，不转移到错误保护，但是不能保证读取的值。
- 不能更改 FMPDR 参数指定的编程数据。在通过中断处理准备编程数据时，必须先将数据准备到其他区域，然后在确认编程已结束转移到 FMPDR 区或者将 FMPDR 更改为已准备好的其他区域。
- 不能在中断处理程序中破坏本章的闪存相关寄存器和已下载的内部程序区。另外，在中断处理中，不能进行 RAM 仿真，也不能多重进行 SCO 请求的内部程序下载或者编程 / 擦除。
- 必须在中断处理程序的起始位置保存 CPU 的寄存器，并且在返回前恢复这些寄存器。
- 如果在中断处理程序中转移到睡眠状态或者软件待机状态，就进入错误保护状态，中断编程 / 擦除。另外，当转移到复位状态时，为了减少对闪存的损伤，必须在 100μs 以上的复位状态后解除复位。

### 23.8.3 其他注意事项

#### (1) 内部程序的下载执行时间

包含初始化程序的编程程序或者包含初始化程序的擦除程序的代码长度都在 3KB 以内。因此，当 CPU 的时钟频率为 20MHz 时，各程序的下载时间最大约为 10ms。

#### (2) 用户分支处理的间隔

执行用户分支处理的间隔因编程 / 擦除和处理阶段而不同。CPU 的时钟频率为 80MHz 时的最大启动间隔如表 23.11 所示。

表 23.11 用户分支处理的启动间隔

	最大间隔
编程处理	约 2ms
擦除处理	约 15ms

但是，在 CPU 时钟以 80MHz 运行时的最初用户分支处理为止的时间最大值如表 23.12 所示。

表 23.12 用户分支处理时间

	最大
编程处理	约 2ms
擦除处理	约 15ms

#### (3) 通过 DMAC/DTC 对闪存相关寄存器的写操作

只要在内部 RAM 中的指令执行期间，也能通过 DMAC/DTC 对下载所需的 FCCS 寄存器的 SCO 位或者 MAT 转换的 FMATS 寄存器进行写操作。如果没注意对这些寄存器进行了写操作，就可能会进行下载，并且进行 RAM 的破坏或者 MAT 的转换，导致 CPU 失控等。

#### (4) 中断的忽视状态

在以下的模式或者期间中，即使发生中断也被忽视，不执行也不保存中断源。

- 引导模式运行中
- 编程器模式运行中

#### (5) 用户 MAT 在使用 256KB 的产品进行编程时的注意事项

用户 MAT 使用 256KB 的产品进行大于等于 256KB 的编程时，不保证 256KB 以后的编程内容，请注意。

#### (6) 和以前的 F-ZTAT SH 单片机的编程 / 擦除程序的兼容性

不支持由 SCO 传送请求进行的内部程序的下载方式，本 LSI 不能运行以前的 F-ZTAT SH 单片机使用的闪存编程 / 擦除程序。

必须在下载内部程序后进行本 LSI 的闪存编程 / 擦除。

## (7) 使用 WDT 对失控等的监视

和以前的 F-ZTAT SH 单片机不同，在通过下载的内部程序进行编程 / 擦除中，不实施由 WDT 对失控等的对策。

根据需要，必须使用考虑了编程 / 擦除执行时间的 WDT 使用对策（使用用户分支程序或者定期的定时器中断等）。

## 23.9 附录

### 23.9.1 引导模式的标准串行通信接口规格

在引导模式中启动的引导程序使用主机和 LSI 内部 SCI 进行接收和发送。主机和引导程序的串行通信接口规格如下所示：

#### 23.9.1.1 状态

引导程序有 3 个状态：

##### 1. 位速率匹配状态

这是使主机和接收 / 发送的位速率进行匹配的状态。当启动引导模式时，引导程序就被启动，进入位速率匹配状态，接收主机的命令，并进行位速率的匹配。当匹配结束时，就转移到查询选择状态。

##### 2. 查询选择状态

这是应答主机查询命令的状态。在此状态中选择器件、时钟模式和位速率。当选择结束时，通过编程 / 擦除状态转移命令转移到编程 / 擦除状态。在转移到编程 / 擦除状态前，引导程序将擦除程序传送到 RAM，并擦除用户 MAT 和用户引导 MAT。

##### 3. 编程 / 擦除状态

这是进行编程 / 擦除的状态。按照主机的命令，将编程 / 擦除程序传送到 RAM，并进行编程 / 擦除。通过命令进行校验和及空白检查。

引导程序的处理流程如图 23.22 所示。

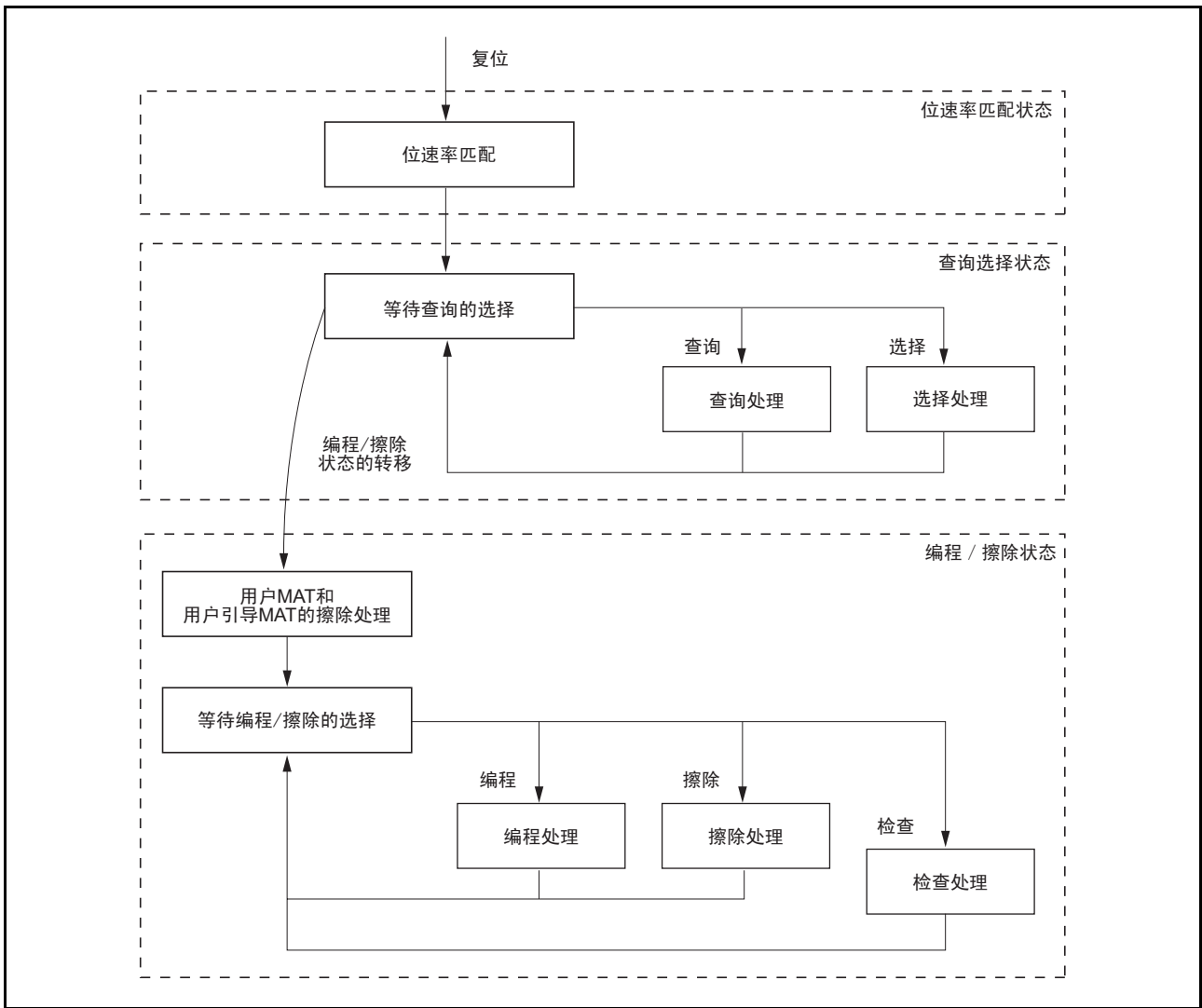


图 23.22 引导程序的处理流程

### 23.9.1.2 位速率匹配状态

位速率匹配测量主机发送的 H'00 的低电平区间，计算位速率。能通过新的位速率选择命令更改位速率。当位速率匹配结束时，引导程序转移到查询选择状态。位速率匹配的顺序如图 23.23 所示。



图 23.23 位速率匹配的顺序

### 23.9.1.3 通信协议

位速率匹配结束后的主机和引导程序的串行通信协议如下：

1. 1 个字符命令或者 1 个字符应答

因为命令或者应答只有 1 个字符，所以有查询和正常结束的 ACK。

2. n 个字符命令或者 n 个字符应答

因为命令和应答需要 n 字节的数据，所以有选择的命令和对应查询的应答。

因为对编程数据另外规定数据长，所以省略数据长。

3. 错误应答

这是对命令的错误应答，错误应答和错误码为 2 字节。

4. 128 字节的编程

这是没有长度的命令，能通过编程长度查询的应答，得知数据的长度。

5. 存储器读取的应答

这是含有 4 字节长度信息的应答。



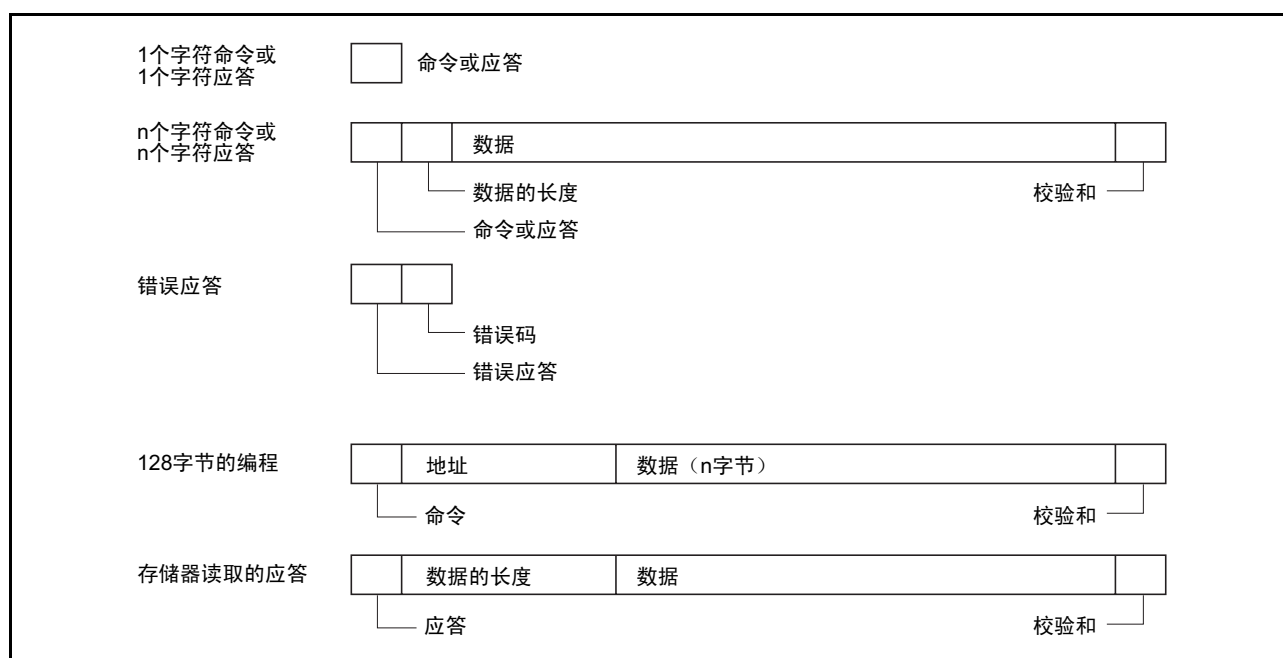


图 23.24 通信协议格式

- 命令（1 字节）：查询、选择、编程、擦除、检查等命令
- 应答（1 字节）：对查询的应答
- 长度（1 字节或者 2 字节）：除命令 / 应答、长度、校验和以外的接收和发送数据的长度
- 数据（n 字节）：命令和应答的详细数据
- 校验和（1 字节）：加算从命令到校验和的总和，并通过设定使低位 1 字节为 H'00
- 错误应答（1 字节）：对指令的错误应答
- 错误码（1 字节）：发生的错误种类
- 地址（4 字节）：编程地址
- 数据（n 字节）：编程数据。n 能从编程长度查询命令的应答得知
- 数据的长度（4 字节）：对存储器读取的 4 字节应答

#### 23.9.1.4 查询选择状态

在查询选择状态中，引导程序对主机的查询命令进行闪存 ROM 信息的应答，对选择的命令选择器件、时钟模式和位速率。

查询选择命令一览表如表 23.13 所示。

表 23.13 查询选择命令一览表

命令	命令名	功能
H'20	支持器件的查询	查询器件的代码和产品名
H'10	器件选择	选择器件代码
H'21	时钟模式查询	查询时钟模式数和各自的值
H'11	时钟模式选择	通知已选择的时钟模式
H'22	倍增比查询	查询倍增比或者分频比种类数和各自的个数及其值
H'23	工作频率查询	查询主时钟和外围时钟的最小值和最大值
H'24	用户引导 MAT 信息查询	查询用户引导 MAT 的个数以及各自的起始地址和最终地址
H'25	用户 MAT 信息查询	查询用户 MAT 的个数以及各自的起始地址和最终地址
H'26	擦除块信息查询	查询块数以及各自的起始地址和最终地址
H'27	编程长度查询	查询编程时的数据长
H'3F	新位速率选择	选择新位速率
H'40	编程 / 擦除状态转移	擦除用户 MAT 和用户引导 MAT, 转移到编程 / 擦除状态
H'4F	引导程序状态查询	查询引导的处理状态

必须按器件的选择 (H'10)、时钟模式的选择 (H'11)、新位速率的选择 (H'3F) 的顺序从主机发送选择命令。当同一选择命令发送 2 次以上时, 后发送的选择命令有效。

除引导程序状态查询 (H'4F) 以外, 这些命令在接受编程 / 擦除状态转移 (H'40) 之前都有效, 主机能从上述的命令中选择需要的命令进行查询。引导程序状态查询 (H'4F) 在接受编程 / 擦除状态转移 (H'40) 后也有效。

### (1) 支持器件的查询

对于支持器件的查询, 引导程序应答可支持器件的器件代码和产品名。

命令 

H'20
------

- 命令 [H'20] (1 字节): 支持器件的查询

应答	H'30	长度	器件数	
	字符数	器件代码		产品名
	...			
	SUM			

- 应答 [H'30] (1 字节): 对支持器件的查询的应答
- 长度 (1 字节): 命令、长度、校验和除外的接收 / 发送数据的长度。在此为器件数、字符数、器件代码和产品名的合计长度
- 器件数 (1 字节): 单片机内的引导程序所支持的品种数
- 字符数 (1 字节): 器件代码和引导程序产品名的字符数
- 器件代码 (4 字节): 支持的产品名的 ASCII 代码
- 产品名 (n 字节): 引导程序产品型号 (ASCII 代码)
- SUM (1 字节): 校验和  
命令到 SUM 的总和, 并使其结果为 H'00。

## (2) 器件选择

对于器件选择，引导程序将支持器件设定到指定的支持器件，对此后的查询，应答已选择的器件信息。

命令	H'10	长度	器件代码	SUM
----	------	----	------	-----

- 命令 [H'10] (1 字节)：器件选择
- 长度 (1 字节)：器件代码的字符数 (是固定值，为 2)
- 器件代码 (4 字节)：支持器件的查询时应答的器件代码 (ASCII 代码)
- SUM (1 字节)：校验和

应答	H'06
----	------

- 应答 [H'06] (1 字节)：对器件选择的应答  
当指定的器件代码和支持器件一致时返回 ACK

错误

应答	H'90	ERROR
----	------	-------

- 错误应答 [H'90] (1 字节)：对器件选择的错误应答
- ERROR: (1 字节)：错误码  
H'11: 校验和错误  
H'21: 器件代码不一致错误

## (3) 时钟模式查询

对于时钟模式查询，引导程序应答可选择的时钟模式。

指令	H'21
----	------

- 命令 [H'21] (1 字节)：时钟模式查询

应答	H'31	长度	模式	...	SUM
----	------	----	----	-----	-----

- 应答 [H'31] (1 字节)：对时钟模式查询的应答
- 长度 (1 字节)：模式数和模式的合计长度
- 模式 (1 字节)：可选择的时钟模式 (例：H'01 时钟模式 1)
- SUM (1 字节)：校验和

## (4) 时钟模式选择

对于时钟模式选择，引导程序将时钟模式设定到指定的时钟模式，对此后的查询，应答已选择的时钟模式的信息。

必须在发送器件选择命令后，发送时钟模式选择命令。

命令	H'11	长度	模式	SUM
----	------	----	----	-----

- 命令 [H'11] (1 字节)：时钟模式选择
- 长度 (1 字节)：模式的字符数 (是固定值，为 1)
- 模式 (1 字节)：时钟模式查询时应答的时钟模式
- SUM (1 字节)：校验和

应答	H'06
----	------

- 应答 [H'06] (1 字节)：对时钟模式选择的应答  
当指定的时钟模式和可选择的时钟模式一致时返回 ACK

错误

应答	H'91	ERROR
----	------	-------

- 错误应答 [H'91] (1 字节)：对时钟模式选择的错误应答
- ERROR: (1 字节)：错误码  
H'11: 校验和错误  
H'22: 时钟模式不一致错误

## (5) 倍增比查询

对于倍增比查询，引导程序应答可选择的倍增比或者分频比。

命令	H'22
----	------

- 命令 [H'22] (1 字节)：倍增比查询

应答	H'32	长度	频率种类数						
	倍增比数	倍增比	...						
	...								
	SUM								

- 应答 [H'32] (1 字节)：对倍增比查询的应答
- 长度 (1 字节)：频率种类数、倍增比数和倍增比的合计长度
- 频率种类数 (1 字节)：器件可选择的倍增比的种类数  
(如果是主工作频率和外围模块工作频率 2 种，为 H'02)
- 倍增比数 (1 字节)：各工作频率可选择的倍增比数  
主模块和外围模块可选择的倍增比数
- 倍增比 (1 字节)  
倍增比：倍增的数值 (例 4 倍增: H'04)  
分频比：分频的数值，为负数 (例 2 分频: H'FE[-2])  
倍增比的重复次数为倍增比数，倍增比数和倍增比的组合重复次数为频率种类数。
- SUM (1 字节)：校验和

## (6) 工作频率查询

对于工作频率查询，引导程序应答工作频率的种类数及其最小值和最大值。

命令 

H'23
------

- 命令 [H'23] (1 字节)：工作频率查询

应答	H'33	长度	频率种类数
	工作频率最小值		工作频率最大值
	...		
	SUM		

- 应答 [H'33] (1 字节)：对工作频率查询的应答
- 长度 (1 字节)：频率种类数、工作频率最小值和工作频率最大值的合计长度
- 频率种类数 (1 字节)：器件所需要的工作频率的种类数  
例如，在主工作频率和外围模块工作频率的情况下为 2
- 工作频率最小值 (2 字节)：倍增或者分频时钟的最小值  
工作频率的最小值和最大值是将频率 (MHz) 小数点 2 位为止的值放大 100 倍的值 (例如，当频率为 20.00MHz 时，放大 100 倍后为 2000，设定为 H'07D0)。
- 工作频率最大值 (2 字节)：倍增或者分频时钟的最大值  
工作频率的最小值和最大值的数据持续次数为频率种类数
- SUM (1 字节)：校验和

## (7) 用户引导 MAT 信息查询

对于用户引导 MAT 信息查询，引导程序应回答用户引导 MAT 的区域数和地址。

命令 

H'24
------

- 命令 [H'24] (1 字节)：用户引导 MAT 信息查询

应答	H'34	长度	区域数
	区域起始地址		区域最终地址
	...		
	SUM		

- 应答 [H'34] (1 字节)：对用户引导 MAT 信息查询的应答
- 长度 (1 字节)：区域数、区域起始地址和区域最终地址的合计长度
- 区域数 (1 字节)：用户引导 MAT 的区域数  
当用户引导 MAT 的区域连续时为 H'01
- 区域起始地址 (4 字节)：区域的起始地址
- 区域最终地址 (4 字节)：区域的最终地址  
区域起始地址和区域最终地址的数据重复次数为区域数
- SUM (1 字节)：校验和

## (8) 用户 MAT 信息查询

对于用户 MAT 信息查询，引导程序应响应用户 MAT 的区域数和地址。

命令 

H'25
------

- 命令 [H'25] (1 字节)：用户 MAT 信息查询

应答	H'35	长度	区域数	
	区域起始地址			区域最终地址
	...			
	SUM			

- 应答 [H'35] (1 字节)：对用户 MAT 信息查询的应答
- 长度 (1 字节)：区域数、区域起始地址和区域最终地址的合计长度
- 区域数 (1 字节)：用户 MAT 的区域数  
当用户 MAT 的区域连续时为 H'01
- 区域起始地址 (4 字节)：区域的起始地址
- 区域最终地址 (4 字节)：区域的最终地址  
区域起始地址和区域最终地址的数据重复次数为区域数
- SUM (1 字节)：校验和

## (9) 擦除块信息查询

对于擦除块信息查询，引导程序应响应用户 MAT 的擦除块的块数和地址。

命令 

H'26
------

- 命令 [H'26] (1 字节)：擦除块信息查询

应答	H'36	长度	块数	
	块起始地址			块最终地址
	...			
	SUM			

- 应答 [H'36] (1 字节)：对擦除块信息查询的应答
- 长度 (2 字节)：块数、块起始地址和块最终地址的合计长度
- 块数 (1 字节)：闪存的擦除块数
- 块起始地址 (4 字节)：块的起始地址
- 块最终地址 (4 字节)：块的最终地址  
块起始地址和块最终地址的数据重复次数为块数
- SUM (1 字节)：校验和

## (10) 编程长度查询

对于编程长度查询，引导程序应答编程数据的编程单位。

命令 

H'27
------

- 命令 [H'27] (1 字节)：编程长度查询

应答 

H'37	长度	编程长度	SUM
------	----	------	-----

- 应答 [H'37] (1 字节)：对编程长度查询的应答
- 长度 (1 字节)：编程单位长度的字符数 (是固定值，为 2)
- 编程长度 (2 字节)：编程单位的长度  
以此长度接受编程数据
- SUM (1 字节)：校验和

## (11) 新位速率选择

对于新位速率选择，引导程序将选择更改被指定的位速率，对于确认应答新位速率。必须在发送时钟模式选择命令后，发送新位速率选择命令。

命令 

H'3F	长度	位速率	输入频率
倍增比数	倍增比 1	倍增比 2	
SUM			

- 命令 [H'3F] (1 字节)：新位速率选择
- 长度 (1 字节)：位速率、输入频率、倍增比数和倍增比的合计长度
- 位速率 (2 字节)：新位速率  
为 1/100 的值 (例如，当位速率为 19200bps 时为 192，设定为 H'00C0)
- 输入频率 (2 字节)：输入到引导程序的时钟频率  
为频率 (MHz) 小数点 2 位为止的值 (例如，当时钟频率为 28.882MHz 时，将小数点 2 位为止的值放大 100 倍后为 2888，设定为 H'0B48)
- 倍增比数 (1 字节)：器件可选择的倍增比数  
通常是主工作频率和外围模块工作频率，为 2
- 倍增比 1 (1 字节)：主工作频率的倍增比或者分频比  
倍增比：倍增的数值 (例 4 倍增：H'04)  
分频比：分频的数值，为负数 (例 2 分频：H'FE[-2])
- 倍增比 2 (1 字节)：外围工作频率的倍增比或者分频比  
倍增比：倍增的数值 (例 4 倍增：H'04)  
分频比：分频的数值，为负数 (例 2 分频：H'FE[-2])
- SUM (1 字节)：校验和

应答 

H'06
------

- 应答 [H'06] (1 字节)：对新位速率选择的应答  
当选择指定的位速率时作为 ACK 发送

错误

应答 

H'BF	ERROR
------	-------

- 错误应答 [H'BF] (1 字节)：对新位速率选择的错误应答

- **ERROR:** (1 字节): 错误码
  - H'11: 校验和错误
  - H'24: 不能选择位速率的错误  
不能选择被指定的位速率
  - H'25: 输入频率的错误  
输入频率不在最小值和最大值的范围内
  - H'26: 倍增比错误  
倍增比不一致
  - H'27: 工作频率错误  
工作频率不在最小值和最大值的范围内

接收数据的检查方法如下所示:

1. 输入频率

检查接收的输入频率的值是否在输入频率的最小值和最大值的范围内 (对于已选择器件的时钟模式), 如果不在范围内, 就为输入频率错误。

2. 倍增比

检查已接收的倍增比或者分频比的值是否和倍增比或者分频比 (对于已选择器件的时钟模式) 一致。如果不一致, 就为倍增比错误。

3. 工作频率

由接收的输入频率以及倍增比或者分频比计算工作频率。输入频率是提供给 LSI 的频率, 工作频率是 LSI 实际的工作频率, 计算式如下所示:

$$\text{工作频率} = \text{输入频率} \times \text{倍增比}$$

$$\text{或者 工作频率} = \text{输入频率} \div \text{分频比}$$

检查此计算的工作频率是否在工作频率的最小值和最大值的范围内 (对于已选择器件的时钟模式), 如果不在范围内, 就为工作频率错误。

4. 位速率

由外围工作频率 ( $P\phi$ ) 和位速率 ( $B$ ) 计算串行模式寄存器 (SCSMR) 的时钟选择 (CKS) 的值 ( $n$ )、位速率寄存器 (SCBRR) 的值 ( $N$ ) 以及误差。检查误差是否小于 4%, 如果误差大于等于 4%, 就为位速率选择错误。误差的计算如下:

$$\text{误差} (\%) = \left\{ \left[ \frac{P\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} \right] - 1 \right\} \times 100$$

如果能选择新位速率, 就在应答 ACK 后将新位速率的值设定到寄存器。主机以新位速率发送 ACK, 引导程序以新位速率应答。

确认

- 确认 [H'06] (1 字节): 新位速率的确认

应答

- 应答 [H'06] (1 字节): 对确认新位速率的应答  
选择新位速率的顺序如图 23.25 所示。





图 23.25 选择新位速率的顺序

### (12) 编程 / 擦除状态转移

对于编程 / 擦除状态转移，引导程序传送擦除程序，并按用户 MAT、用户引导 MAT 的顺序擦除 MAT 的数据。当擦除结束时，应答 ACK，变为编程 / 擦除状态。

在发送编程选择命令的编程数据前，主机必须通过器件选择命令、时钟模式选择命令和新位速率选择命令，选择 LSI 的器件、时钟模式和新位速率，并将编程 / 擦除状态转移命令发送到引导程序。

命令 

H'40
------

- 命令 [H'40] (1 字节)：编程 / 擦除状态转移

应答 

H'06
------

- 应答 [H'06] (1 字节)：对编程 / 擦除状态转移的应答。在传送擦除程序后，当用户引导 MAT 和用户 MAT 正常擦除数据后返回 ACK

错误

应答 

H'C0	H'51
------	------

- 错误应答 [H'C0] (1 字节)：对编程 / 擦除状态转移的错误应答
- 错误码 [H'51] (1 字节)：擦除错误  
因发生错误而不能擦除

#### 23.9.1.5 命令错误

如果未定义命令、命令顺序不正确或者不能接受命令，就为命令错误。例如，器件选择前的时钟模式选择命令或者编程 / 擦除状态转移命令后的查询命令为命令错误。

错误

应答 

H'80	H'xx
------	------

- 错误应答 [H'80] (1 字节)：命令错误
- 命令 [H'xx] (1 字节)：已接收的命令

## 23.9.1.6 命令顺序

在查询选择状态下的命令顺序的例子如下：

1. 必须通过支持器件的查询（H'20），查询支持的器件。
2. 必须从被应答的器件信息中，进行器件选择（H'10）。
3. 必须通过时钟模式查询（H'21），查询时钟模式。
4. 必须从被应答的时钟模式中，进行时钟模式选择（H'11）。
5. 当器件选择和时钟模式选择结束时，必须通过倍增比查询（H'22）和工作频率查询（H'23），查询新位速率选择所需要的信息。
6. 必须按照倍增比和工作频率的信息，选择新位速率（H'3F）。
7. 当器件选择和时钟模式选择结束时，必须通过用户引导 MAT 信息查询（H'24）、用户 MAT 信息查询（H'25）、擦除块信息查询（H'26）和编程长度查询（H'27），查询用户引导 MAT 和用户 MAT 的编程 / 擦除信息。
8. 当查询和新位速率选择结束时，必须进行编程 / 擦除状态转移（H'40），转移到编程 / 擦除状态。

## 23.9.1.7 编程 / 擦除状态

在编程 / 擦除状态中，引导程序通过编程选择命令进行编程方法的选择，通过 128 字节编程命令进行数据的编程，并且通过擦除选择和块擦除命令进行块的擦除。编程 / 擦除命令一览表如表 23.14 所示。

表 23.14 编程 / 擦除命令一览表

命令	命令名	功能
H'42	用户 MAT 编程选择	选择用户 MAT 的编程程序
H'43	用户引导 MAT 编程选择	选择用户引导 MAT 的编程程序
H'50	128 字节编程	128 字节编程
H'48	擦除选择	选择擦除程序
H'58	块擦除	块数据的擦除
H'52	存储器读取	读存储器
H'4A	用户引导 MAT 的校验和	用户引导 MAT 的校验和
H'4B	用户 MAT 的校验和	用户 MAT 的校验和
H'4C	用户引导 MAT 的空白检查	用户引导 MAT 的空白检查
H'4D	用户 MAT 的空白检查	用户 MAT 的空白检查
H'4F	引导程序状态查询	查询引导的处理状态

## 23.9.1.8 编程

使用编程选择命令和 128 字节编程命令进行编程。

首先，主机发送编程选择命令，选择编程方式和编程 MAT。根据编程区域和编程方式有以下 2 个编程选择命令：

1. 用户引导 MAT 编程选择
2. 用户 MAT 编程选择

其次，发送 128 字节编程命令，接在选择命令后面的 128 字节编程命令分别被解释为选择命令所指定的编程方式的编程数据。当编程超过 128 字节的数据时，必须重复 128 字节的编程命令。如果要结束编程，必须由主机发送地址为 H'FFFFFFF 的 128 字节编程命令。当编程结束时，变为编程 / 擦除选择命令的等待状态。

当继续进行其他方式和其他 MAT 的编程时，从编程选择命令开始。

编程选择命令和 128 字节编程命令的顺序如图 23.26 所示。



图 23.26 编程顺序

## (1) 用户引导 MAT 编程选择

对于用户引导 MAT 编程选择，引导程序传送编程程序，并通过传送的编程程序对用户引导 MAT 进行编程。

命令 H'42

- 命令 [H'42] (1 字节)：用户引导 MAT 编程选择

应答 H'06

- 应答 [H'06] (1 字节)：对用户引导 MAT 编程选择的应答。当编程程序传送结束时返回 ACK

错误

应答 H'C2 ERROR

- 错误应答 [H'C2] (1 字节)：对用户引导 MAT 编程选择的错误应答
- ERROR: (1 字节)：错误码

H'54: 选择处理错误 (发生传送错误, 处理未结束)

## (2) 用户 MAT 编程选择

对于用户 MAT 编程选择，引导程序传送编程程序，并通过传送的编程程序对用户 MAT 进行编程。

命令 

H'43
------

- 命令 [H'43] (1 字节)：用户 MAT 编程选择

应答 

H'06
------

- 应答 [H'06] (1 字节)：对用户 MAT 编程选择的应答。当编程程序传送结束时返回 ACK

错误

应答 

H'C3	ERROR
------	-------

- 错误应答 [H'C3] (1 字节)：对用户 MAT 编程选择的错误应答
- ERROR: (1 字节)：错误码  
H'54: 选择处理错误 (发生传送错误, 处理未结束)

## (3) 128 字节编程

对于 128 字节编程，引导程序通过编程选择传送的编程程序，对用户引导 MAT 或者用户 MAT 进行编程。

命令	H'50	编程地址						
	数据	...						
	...							
	SUM							

- 命令 [H'50] (1 字节)：128 字节编程
- 编程地址 (4 字节)：编程的起始地址  
必须指定 128 字节边界的地址。  
例) H'00、H'01、H'00、H'00: H'00010000
- 编程数据 (n 字节)：编程数据  
编程数据的长度是“编程长度查询”应答的长度
- SUM (1 字节)：校验和

应答 

H'06
------

- 应答 [H'06] (1 字节)：对 128 字节编程的应答  
当编程结束时返回 ACK

错误

应答 

H'D0	ERROR
------	-------

- 错误应答 [H'D0] (1 字节)：对 128 字节编程的错误应答
- ERROR: (1 字节)：错误码  
H'11: 校验和错误  
H'2A: 地址错误 (地址不在指定 MAT 的范围内)  
H'53: 编程错误 (发生编程错误, 不能编程)

必须按照数据编程长度指定边界地址。例如，当数据编程长度为 128 字节时，必须将地址的低位字节设定为 H'00 或者 H'80。

主机必须将 128 字节中没有编程数据的部分填写 H'FF，然后发送。

当结束编程处理时，必须发送地址 H'FFFFFFF 的 128 字节编程命令。对地址 H'FFFFFFF 的 128 字节编程命令，引导程序判断数据已结束，变为编程 / 擦除选择命令的等待状态。

命令	H'50	编程地址	SUM
----	------	------	-----

- 命令 [H'50] (1 字节)：128 字节编程
- 编程地址 (4 字节)：结束码 (H'FF、H'FF、H'FF、H'FF)
- SUM (1 字节)：校验和

应答	H'06
----	------

- 应答 [H'06] (1 字节)：对 128 字节编程的应答  
当编程处理结束时返回 ACK

错误

应答	H'D0	ERROR
----	------	-------

- 错误应答 [H'D0] (1 字节)：对 128 字节编程的错误应答
- ERROR: (1 字节)：错误码  
H'11：校验和错误  
H'53：编程错误

### 23.9.1.9 擦除

使用擦除选择命令和块擦除命令，进行擦除。

首先通过擦除选择命令，选择擦除，其次对块擦除命令所指定的块进行擦除。当有多块擦除块时，重复块擦除命令。当结束擦除处理时，必须从主机发送块号为 H'FF 的块擦除命令。当擦除结束时，变为编程 / 擦除选择命令的等待状态。

擦除选择命令和块擦除命令的顺序如图 23.27 所示。

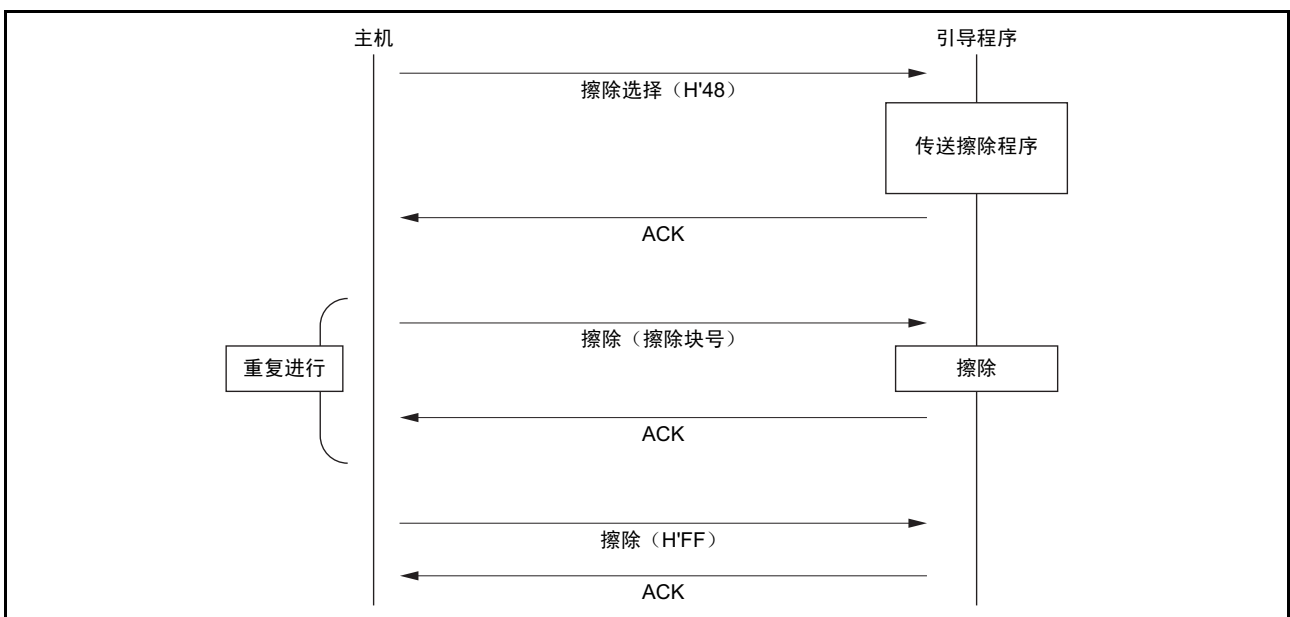


图 23.27 擦除顺序

## (1) 擦除选择

对于擦除选择，引导程序传送擦除程序，并通过传送的擦除程序对用户 MAT 的数据进行擦除。

命令 

H'48
------

- 命令 [H'48] (1 字节)：擦除选择

应答 

H'06
------

- 应答 [H'06] (1 字节)：对擦除选择的应答  
当擦除程序传送结束时返回 ACK

错误

应答 

H'C8	ERROR
------	-------

- 错误应答 [H'C8] (1 字节)：对擦除选择的错误应答
- ERROR: (1 字节)：错误码  
H'54: 选择处理错误 (发生传送错误, 处理未结束)

## (2) 块擦除

对于擦除，引导程序擦除被指定的用户 MAT 块。

命令 

H'58	长度	块号码	SUM
------	----	-----	-----

- 命令 [H'58] (1 字节)：擦除
- 长度 (1 字节)：擦除块号的字符数 (是固定值, 为 1)
- 块号码 (1 字节)：擦除数据的擦除块号
- SUM (1 字节)：校验和

应答 

H'06
------

- 应答 [H'06] (1 字节)：对擦除的应答  
当擦除结束时返回 ACK

错误

应答 

H'D8	ERROR
------	-------

- 错误应答 [H'D8] (1 字节)：对擦除的错误应答
- ERROR: (1 字节)：错误码  
H'11: 校验和错误  
H'29: 块号错误  
块号不正确  
H'51: 擦除错误  
在擦除中发生错误

对于块号 H'FF，引导程序结束擦除处理，变为选择命令的等待状态。

命令 

H'58	长度	块号码	SUM
------	----	-----	-----

- 命令 [H'58] (1 字节)：擦除
- 长度 (1 字节)：擦除块号码的字符数 (是固定值, 为 1)
- 块号码 (1 字节)：H'FF (擦除处理的结束码)

- SUM (1 字节): 校验和

应答 

H'06
------

- 应答 [H'06] (1 字节): 对擦除结束的应答 ACK  
在由 H'FF 指定块号后重新进行擦除时, 从擦除选择开始执行。

### 23.9.1.10 存储器读取

对存储器读取, 引导程序读取被指定的地址数据。

命令	H'52	长度	区域	读的起始地址			
	读的长度				SUM		

- 命令 [H'52] (1 字节): 存储器读取
- 长度 (1 字节): 区域、读的起始地址和读的长度的合计长度 (是固定值, 为 9)
- 区域 (1 字节)

H'00: 用户引导 MAT

H'01: 用户 MAT

当指定的区域不正确时为地址错误

- 读的起始地址 (4 字节): 读的起始地址
- 读的长度 (4 字节): 读数据的长度
- SUM (1 字节): 校验和

应答	H'52	读的长度					
	数据	...					
	SUM						

- 应答 [H'52] (1 字节): 对存储器读取的应答
- 读的长度 (4 字节): 读数据的长度
- 数据 (n 字节): 从读的起始地址开始, 个数为读的长度的数据
- SUM (1 字节): 校验和

错误

应答 

H'D2	ERROR
------	-------

- 错误应答 [H'D2] (1 字节): 对存储器读的错误应答
- ERROR: (1 字节): 错误码
  - H'11: 校验和错误
  - H'2A: 地址错误  
读的起始地址不在 MAT 的范围内
  - H'2B: 长度错误  
读的长度超过 MAT 的范围, 或者由读的起始地址和读的长度计算的读的最终地址不在 MAT 的范围内, 或者读的长度为 0

## 23.9.1.11 用户引导 MAT 的校验和

对于用户引导 MAT 的校验和，引导程序将用户引导 MAT 的数据相加，并应答其结果。

命令 

H'4A
------

- 命令 [H'4A] (1 字节)：用户引导 MAT 的校验和

应答 

H'5A	长度	MAT 的校验和	SUM
------	----	----------	-----

- 应答 [H'5A] (1 字节)：对用户引导 MAT 的校验和的应答
- 长度 (1 字节)：校验和数据字符数 (是固定值，为 4)
- MAT 的校验和 (4 字节)：用户引导 MAT 的校验和值，以字节为单位相加
- SUM (1 字节)：校验和 (发送数据)

## 23.9.1.12 用户 MAT 的校验和

对于用户 MAT 的校验和，引导程序将用户 MAT 的数据相加，并应答其结果。

命令 

H'4B
------

- 命令 [H'4B] (1 字节)：用户 MAT 的校验和

应答 

H'5B	长度	MAT 的校验和	SUM
------	----	----------	-----

- 应答 [H'5B] (1 字节)：对用户 MAT 的校验和的应答
- 长度 (1 字节)：校验和数据字符数 (是固定值，为 4)
- MAT 的校验和 (4 字节)：用户 MAT 的校验和值，以字节为单位相加
- SUM (1 字节)：校验和 (发送数据)

## 23.9.1.13 用户引导 MAT 的空白检查

对于用户引导 MAT 的空白检查，引导程序检查用户引导 MAT 是否全部为空白，并应答其结果。

命令 

H'4C
------

- 命令 [H'4C] (1 字节)：用户引导 MAT 的空白检查

应答 

H'06
------

- 应答 [H'06] (1 字节)：对用户引导 MAT 的空白检查的应答。当区域全部为空白 (H'FF) 时返回 ACK

错误

应答 

H'CC	H'52
------	------

- 错误应答 [H'CC] (1 字节)：对用户引导 MAT 的空白检查的错误应答
- 错误码 [H'52] (1 字节)：未擦除错误



## 23.9.1.14 用户 MAT 的空白检查

对于用户 MAT 的空白检查，引导程序检查用户 MAT 是否全部为空白，并应答其结果。

命令 

H'4D
------

- 命令 [H'4D] (1 字节)：用户 MAT 的空白检查

应答 

H'06
------

- 应答 [H'06] (1 字节)：对用户 MAT 的空白检查的应答。当区域全部为空白 (H'FF) 时返回 ACK

错误

应答 

H'CD	H'52
------	------

- 错误应答 [H'CD] (1 字节)：对用户 MAT 的空白检查的错误应答
- 错误码 [H'52] (1 字节)：未擦除错误

## 23.9.1.15 引导程序状态查询

对于引导程序状态查询，引导程序应答当前的状态和错误状态。此查询在查询选择状态和编程 / 擦除状态中都有效。

命令 

H'4F
------

- 命令 [H'4F] (1 字节)：引导程序状态查询

应答 

H'5F	长度	STATUS	ERROR	SUM
------	----	--------	-------	-----

- 应答 [H'5F] (1 字节)：对引导程序状态查询的应答
- 长度 (1 字节)：数据的字符数 (是固定值, 为 2)
- STATUS (1 字节)：标准引导程序的状态  
请参照表 23.15。
- ERROR (1 字节)：错误状态  
当 ERROR=0 时, 正常  
当 ERROR 不为 0 时, 异常  
请参照表 23.16。
- SUM (1 字节)：校验和

表 23.15 状态码

代码	内容
H'11	等待器件的选择
H'12	等待时钟模式的选择
H'13	等待位速率的选择
H'1F	等待编程 / 擦除的状态转移 (位速率选择结束)
H'31	正在擦除用户 MAT 和用户引导 MAT
H'3F	等待编程 / 擦除的选择 (擦除结束)
H'4F	等待编程数据的接收 (编程结束)
H'5F	等待擦除块的指定 (擦除结束)

表 23.16 错误码

代码	内容
H'00	无错误
H'11	校验和错误
H'21	器件代码不一致错误
H'22	时钟模式不一致错误
H'24	位速率的选择错误
H'25	输入频率错误
H'26	倍增比错误
H'27	工作频率错误
H'29	块号错误
H'2A	地址错误
H'2B	数据长错误
H'51	擦除错误
H'52	未擦除错误
H'53	编程错误
H'54	选择处理错误
H'80	命令错误
H'FF	位速率匹配确认错误

### 23.9.2 过程程序或者编程数据的可保存区

前几节说明的编程 / 擦除过程程序和编程数据的可保存区为内部 RAM，但是如果满足以下条件，也可在其他区域（如外部空间区等）执行。

1. 因为从 FTDAR 寄存器指定的内部 RAM 地址下载并执行内部编程 / 擦除程序，所以此区域不能使用。
2. 因为内部编程 / 擦除程序使用 128 字节以上的堆栈区，所以必须确保该区域。
3. 因为在将 SCO 位设定为 1 进行下载请求的处理中产生 MAT 转换，所以必须在内部 RAM 中进行。
4. 在开始编程 / 擦除前（在判断下载结果前）能存取闪存。对于如单芯片模式不能存取外部空间的模式，必须在此之前将需要的过程程序、中断向量、中断处理程序 and 用户分支处理程序等传送到内部 RAM。
5. 因为在编程 / 擦除处理中不能存取闪存，所以通过已被下载到内部 RAM 中的程序来执行。启动此程序的过程程序、编程 / 擦除中的用户分支目标的用户程序、中断的向量表和中断处理程序的执行区也需要置于内部 RAM 上。
6. 从编程 / 擦除结束后到清除 FKEY 寄存器的期间，禁止存取闪存。  
在编程 / 擦除结束后立刻更改 LSI 模式进行复位时，必须设有 100 $\mu$ s 以上的复位期间（ $\overline{\text{RES}}=0$  的期间）。另外，在编程 / 擦除处理中禁止转移到复位状态，如果错误地进行了复位，就必须在长于通常的 100 $\mu$ s 复位期间后解除复位。
7. 在用户引导模式中进行用户 MAT 的编程 / 擦除时，需要在内部 RAM 中通过 FMATS 进行 MAT 转换（请参照“23.8.1 用户 MAT 和用户引导 MAT 的转换”）。  
对于 MAT 的转换，必须注意现在选择了哪个 MAT。
8. 如果编程处理的参数 FMPDR 所指示的编程数据保存区在闪存中，就判断为错误，所以必须暂时将编程数据传送到内部 RAM，并将 FMPDR 指示的地址设定在闪存空间以外。

基于这些条件，按照各运行模式 / 处理内容的组合，编程数据保存区和可执行区如下表所示：

表 23.17 可执行的 MAT

处理	启动模式	
	用户编程模式	用户引导模式 *
编程	表 23.18 (1)	表 23.18 (3)
擦除	表 23.18 (2)	表 23.18 (4)

【注】 \* 可对用户 MAT 进行编程 / 擦除。

表 23.18 (1) 在用户编程模式中编程处理可使用的区域

	项目	可保存 / 执行的区域			选择的 MAT	
		内部 RAM	用户 MAT	外部空间	用户 MAT	嵌入式程序 保存 MAT
编程 步骤 ↓	编程数据的保存区	○	× *	○	—	—
	下载内部程序的选择处理	○	○	○	○	
	对键寄存器写 H'A5 的处理	○	○	○	○	
	设定 FCCS 的 SCO=1 (下载)	○	×	×		○
	键寄存器的清除处理	○	○	○	○	
	下载结果的判断	○	○	○	○	
	下载的错误处理	○	○	○	○	
	初始化参数的设定处理	○	○	○	○	
	初始化的执行	○	×	×	○	
	初始化结果的判断	○	○	○	○	
	初始化的错误处理	○	○	○	○	
	中断处理程序	○	×	○	○	
	对键寄存器写 H'5A 的处理	○	○	○	○	
	编程参数的设定处理	○	×	○	○	
	编程的执行	○	×	×	○	
	编程结果的判断	○	×	○	○	
	编程的错误处理	○	×	○	○	
	键寄存器的清除处理	○	×	○	○	

【注】 \* 只要预先传送到内部 RAM，就可使用。

表 23.18 (2) 在用户编程模式中擦除处理可使用的区域

项目	可保存 / 执行的区域			选择的 MAT	
	内部 RAM	用户 MAT	外部空间	用户 MAT	嵌入式程序 保存 MAT
下载内部程序的选择处理	○	○	○	○	
对键寄存器写 H'5A 的处理	○	○	○	○	
设定 FCCS 的 SCO=1 (下载)	○	×	×		○
键寄存器的清除处理	○	○	○	○	
下载结果的判断	○	○	○	○	
下载的错误处理	○	○	○	○	
初始化参数的设定处理	○	○	○	○	
初始化的执行	○	×	×	○	
初始化结果的判断	○	○	○	○	
初始化的错误处理	○	○	○	○	
中断处理程序	○	×	○	○	
对键寄存器写 H'5A 的处理	○	○	○	○	
擦除参数的设定处理	○	×	○	○	
擦除的执行	○	×	×	○	
擦除结果的判断	○	×	○	○	
擦除的错误处理	○	×	○	○	
键寄存器的清除处理	○	×	○	○	

擦除步骤



表 23.18 (3) 在用户引导模式中编程处理可使用的区域

项目	可保存 / 执行的区域			选择的 MAT		
	内部 RAM	用户 MAT	外部空间	用户 MAT	用户引导 MAT	嵌入式程序保存 MAT
编程数据的保存区	○	× *1	○	—	—	—
下载内部程序的选择处理	○	○	○		○	
对键寄存器写 H'A5 的处理	○	○	○		○	
设定 FCCS 的 SCO=1 (下载)	○	×	×			○
键寄存器的清除处理	○	○	○		○	
下载结果的判断	○	○	○		○	
下载的错误处理	○	○	○		○	
初始化参数的设定处理	○	○	○		○	
初始化的执行	○	×	×		○	
初始化结果的判断	○	○	○		○	
初始化的错误处理	○	○	○		○	
中断处理程序	○	×	○		○	
通过 FMATS 的 MAT 转换	○	×	×	○		
对键寄存器写 H'5A 的处理	○	×	○	○		
编程参数的设定处理	○	×	○	○		
编程的执行	○	×	×	○		
编程结果的判断	○	×	○	○		
编程的错误处理	○	× *2	○	○		
键寄存器的清除处理	○	×	○	○		
通过 FMATS 的 MAT 转换	○	×	×		○	

【注】 \*1 只要预先传送到内部 RAM，就可使用。

\*2 只要在内部 RAM 中转换 FMATS 后，就可使用。

表 23.18 (4) 在用户引导模式中擦除处理可使用的区域

	项目	可保存 / 执行的区域			选择的 MAT		
		内部 RAM	用户 MAT	外部空间	用户 MAT	用户引导 MAT	嵌入式程序保存 MAT
擦除步骤	下载内部程序的选择处理	○	○	○		○	
	对键寄存器写 H'5A 的处理	○	○	○		○	
	设定 FCCS 的 SCO=1 (下载)	○	×	×			○
	键寄存器的清除处理	○	○	○		○	
	下载结果的判断	○	○	○		○	
	下载的错误处理	○	○	○		○	
	初始化参数的设定处理	○	○	○		○	
	初始化的执行	○	×	×		○	
	初始化结果的判断	○	○	○		○	
	初始化的错误处理	○	○	○		○	
	中断处理程序	○	×	○		○	
	通过 FMATS 的 MAT 转换	○	×	×		○	
	对键寄存器写 H'5A 的处理	○	×	○	○		
	擦除参数的设定处理	○	×	○	○		
	擦除的执行	○	×	×	○		
	擦除结果的判断	○	×	○	○		
	擦除的错误处理	○	× *	○	○		
	键寄存器的清除处理	○	×	○	○		
通过 FMATS 的 MAT 转换	○	×	×		○		

【注】\* 只要在内部 RAM 中转换 FMATS 后，就可使用。

### 23.10 编程器模式

在编程器模式中，和单一闪存相同，能通过插座适配器使用 PROM 编程器进行编程和擦除。请使用支持瑞萨 256K/512K 字节闪存内置型单片机芯片（F-ZTATxxxx）的 PROM 编程器。

## 第 24 章 掩模型 ROM

本 LSI 内置 256KB 的掩模型 ROM。通过宽度为 32 位的数据总线将内部 ROM 连接在 CPU、直接存储器存取控制器 (DMAC)、数据传送控制器 (DTC) 上 (图 24.1)。CPU、DMAC、DTC 可用宽度为 8、16 或 32 位存取内部 ROM。CPU 总是以 1 个状态存取内部 ROM 的数据。

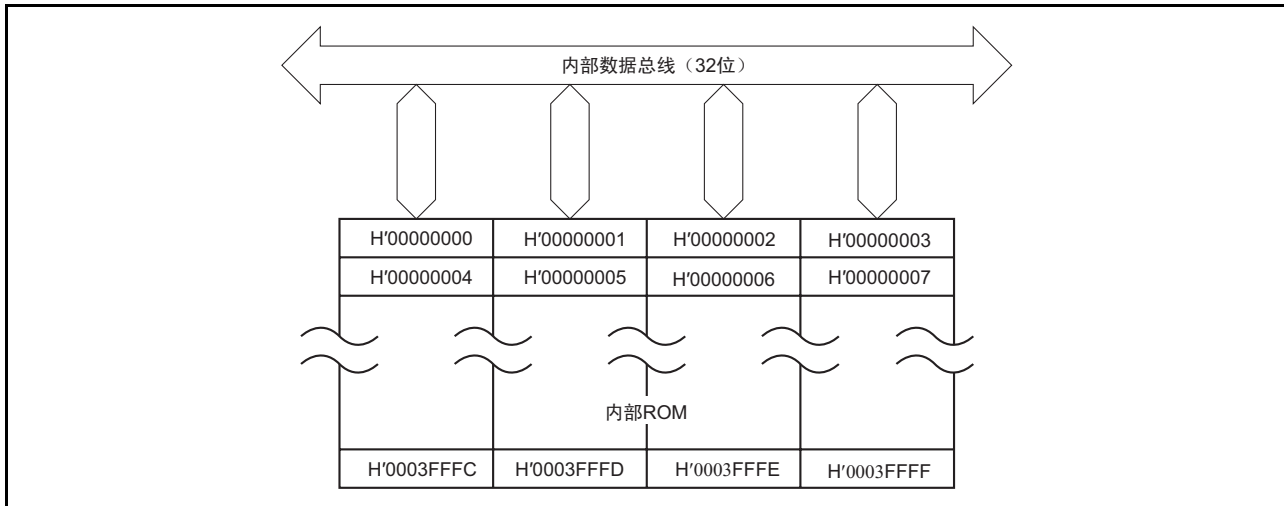


图 24.1 掩模型 ROM 的块图

内部 ROM 有效还是无效由运行模式决定。通过模式设定引脚 FWE、MD1、MD0 选择运行模式。使用内部 ROM 时必须选择模式 2 或模式 3，不使用时则必须选择模式 0 或模式 1。内部 ROM 被分配到存储区 0 的地址 H'00000000 ~ H'0003FFFF 里。

### 24.1 使用注意事项

#### 24.1.1 模块待机模式的设定

通过待机控制寄存器可以设定禁止或者允许掩模型 ROM 的存取。在初始值中，允许掩模型 ROM 的存取。在设定模块待机模式后，禁止掩模型 ROM 的存取。详细内容请参照“第 26 章 低功耗模式”。



## 第 25 章 RAM

本 LSI 内置高速静态 RAM。内部 RAM 通过 32 位数据总线（L 总线）连接到 CPU、通过 32 位数据总线（I 总线）连接到直接存储器存取控制器（DMAC）和数据传送控制器（DTC）。能以 8 位、16 位或者 32 位存取内部 RAM。根据各产品，将内部 RAM 分配到图 25.1 所示的地址，按地址分为 page0 和 page1。CPU（经过 L 总线）和 DMAC/DTC（经过 I 总线）能存取 RAM。如果不同的总线同时对同一页产生存取请求，优先级为 I 总线（DMAC/DTC）>L 总线（CPU）。因为这样的竞争会降低 RAM 的存取性能，为了尽量不发生竞争，推荐使用软件对策。例如，按总线存取不同的页，就不会发生竞争。只要不发生页竞争，L 总线（CPU）的存取就为 1 个周期。I 总线（DMAC/DTC）的存取因内部时钟（I $\phi$ ）和总线时钟（B $\phi$ ）的比或者 DMAC/DTC 的运行状态等而不同。在睡眠模式、软件待机模式、上电复位和手动复位时保持内部 RAM 的内容，但是在深度软件待机模式中不保持内部 RAM 的内容。

通过 RAM 控制寄存器（RAMCR）的 RAME 位，能控制 RAM 的有效或者无效，有关 RAMCR 请参照“26.3.7 RAM 控制寄存器（RAMCR）”。

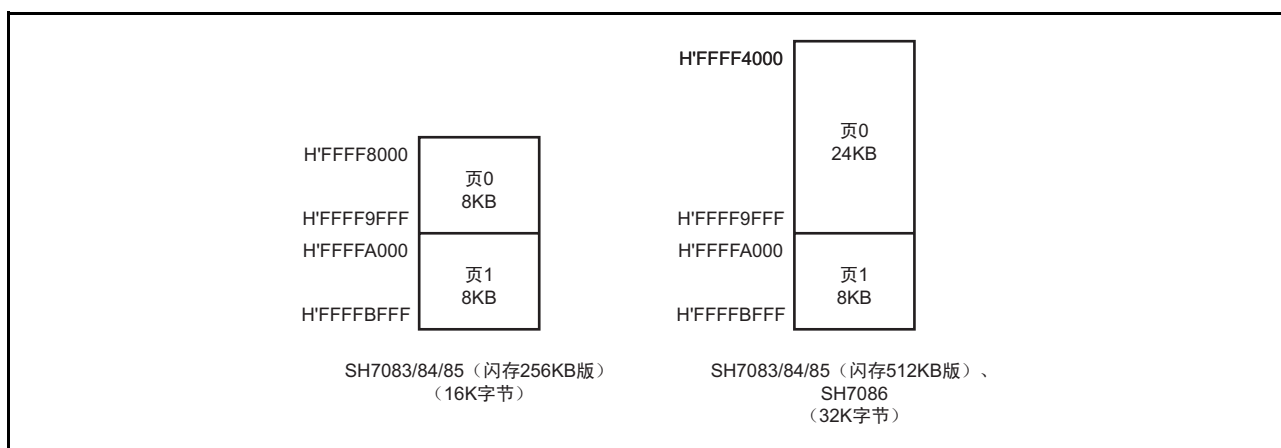


图 25.1 内部 RAM 地址

### 25.1 使用注意事项

#### 25.1.1 模块待机模式的设定

能通过待机控制寄存器，允许或者禁止 RAM 的存取，初始值为允许 RAM 的存取。通过设定模块待机模式，禁止 RAM 的存取。详细内容请参照“第 26 章 低功耗模式”。

#### 25.1.2 地址错误

如果在写 RAM 时发生地址错误，就有可能破坏 RAM 的内容。

#### 25.1.3 RAM 的初始值

上电后，直到对 RAM 进行编程为止，RAM 的初始值不定。

## 第 26 章 低功耗模式

本 LSI 支持低功耗模式的睡眠模式、软件待机模式、深度软件待机模式和模块待机功能。

### 26.1 特点

- 支持睡眠模式、软件待机模式、模块待机功能和深度软件待机模式。

#### 26.1.1 低功耗模式的种类

低功耗模式有如下的模式和功能：

1. 睡眠模式
2. 软件待机模式
3. 深度软件待机模式
4. 模块待机功能

从程序的执行状态转移到各模式的方法、各模式中的 CPU 和外围模式等的状态以及各模式的解除方法如表 26.1 所示。

表 26.1 低功耗模式的状态

低功耗模式	转移方法	状 态					解除方法
		CPG	CPU	CPU 寄存器	内部 存储器	内部 外围模块	
睡眠模式	在 STBCR1 的 STBY 位为 0 的状态下执行 SLEEP 指令	运行	停止	保持	运行	运行	复位
软件待机模式	在 STBCR1 的 STBY 位和 STBCR6 的 STBYMD 位都为 1 的状态下执行 SLEEP 指令	停止	停止	保持	停止 (保持内容)	停止	(1) 由 NMI 或者 IRQ 引起的中断 (2) 由 $\overline{\text{RES}}$ 引脚引起的上电复位 (3) 由 $\overline{\text{MRES}}$ 引脚进行的手动复位
深度软件待机模式	在 STBCR1 的 STBY 位和 STBCR6 的 STBYMD 位分别为 1 和 0 的状态下执行 SLEEP 指令	停止	停止	不定	停止 (内容不定)	停止	由 $\overline{\text{RES}}$ 引脚引起的上电复位
模块待机功能	将 STBCR2 ~ 5 的 MSTP 位设定为 1	运行	运行	保持	停止指定的模块 (保持内容)	停止指定的模块	(1) MSTP 位清 0 (2) 上电复位 (MSTP 位的初始值为 0 的模块)

【注】 有关各模式中的内部外围模块的寄存器状态和引脚状态请分别参照“27.3 各运行模式的寄存器状态”和“附录 A. 引脚状态”。

## 26.2 输入 / 输出引脚

低功耗模式相关的引脚结构如表 26.2 所示。

表 26.2 引脚结构

名称	引脚名	输入 / 输出	功 能
上电复位	$\overline{\text{RES}}$	输入	是上电复位的输入信号。通过低电平进行上电复位。
手动复位	$\overline{\text{MRES}}$	输入	是手动复位的输入信号。通过低电平进行手动复位。

## 26.3 寄存器说明

低功耗模式相关的寄存器有以下寄存器。有关这些寄存器的地址和各处理模式中的寄存器状态请参照“第 27 章 寄存器一览”。

表 26.3 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
待机控制寄存器 1	STBCR1	R/W	H'00	H'FFFE802	8
待机控制寄存器 2	STBCR2	R/W	H'38	H'FFFE804	8
待机控制寄存器 3	STBCR3	R/W	H'FF	H'FFFE806	8
待机控制寄存器 4	STBCR4	R/W	H'FF	H'FFFE808	8
待机控制寄存器 5	STBCR5	R/W	H'03	H'FFFE80A	8
待机控制寄存器 6	STBCR6	R/W	H'00	H'FFFE80C	8
RAM 控制寄存器	RAMCR	R/W	H'10	H'FFFE880	8

### 26.3.1 待机控制寄存器 1 (STBCR1)

STBCR1 是 8 位可读写寄存器，指定低功耗模式的状态。

位:	7	6	5	4	3	2	1	0
	STBY	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
7	STBY	0	R/W	待机 指定向软件待机模式的转移。 0: 通过执行 SLEEP 指令, 转移到睡眠模式 1: 通过执行 SLEEP 指令, 转移到软件待机模式 / 深度软件待机模式
6 ~ 0	—	全 0	R	保留位 读写的值总是 0。

### 26.3.2 待机控制寄存器 2 (STBCR2)

STBCR2 是 8 位可读写寄存器，控制低功耗时各模块的运行。

位:	7	6	5	4	3	2	1	0
	MSTP 7	MSTP 6	-	MSTP 4	MSTP 3	-	-	-
初始值:	0	0	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R	R	R

位	位名	初始值	R/W	说 明
7	MSTP7	0	R/W	模块停止位 7 如果将本位设定为 1，就停止向 RAM 提供时钟。 0: RAM 运行 1: 停止向 RAM 提供时钟
6	MSTP6	0	R/W	模块停止位 6 如果将本位设定为 1，就停止向 ROM 提供时钟。 0: ROM 运行 1: 停止向 ROM 提供时钟
5	—	1	R	保留位 读写的值总是 1。
4	MSTP4	1	R/W	模块停止位 4 如果将本位设定为 1，就停止向 DTC 提供时钟。 0: DTC 运行 1: 停止向 DTC 提供时钟
3	MSTP3	1	R/W	模块停止位 3 如果将本位设定为 1，就停止向 DMAC 提供时钟。 0: DMAC 运行 1: 停止向 DMAC 提供时钟
2 ~ 0	—	全 0	R	保留位 读写的值总是 0。

## 26.3.3 待机控制寄存器 3 (STBCR3)

STBCR3 是 8 位可读写寄存器，控制低功耗时各模块的运行。

位:	7	6	5	4	3	2	1	0
	MSTP 15	MSTP 14	MSTP 13	MSTP 12	MSTP 11	MSTP 10	-	-
初始值:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R

位	位名	初始值	R/W	说 明
7	MSTP15	1	R/W	模块停止位 15 如果将本位设定为 1，就停止向 I <sup>2</sup> C2 提供时钟。 0: I <sup>2</sup> C2 运行 1: 停止向 I <sup>2</sup> C2 提供时钟
6	MSTP14	1	R/W	模块停止位 14 如果将本位设定为 1，就停止向 SCIF 提供时钟。 0: SCIF 运行 1: 停止向 SCIF 提供时钟
5	MSTP13	1	R/W	模块停止位 13 如果将本位设定为 1，就停止向 SCI_2 提供时钟。 0: SCI_2 运行 1: 停止向 SCI_2 提供时钟
4	MSTP12	1	R/W	模块停止位 12 如果将本位设定为 1，就停止向 SCI_1 提供时钟。 0: SCI_1 运行 1: 停止向 SCI_1 提供时钟
3	MSTP11	1	R/W	模块停止位 11 如果将本位设定为 1，就停止向 SCI_0 提供时钟。 0: SCI_0 运行 1: 停止向 SCI_0 提供时钟
2	MSTP10	1	R/W	模块停止位 10 如果将本位设定为 1，就停止向 SSU 提供时钟。 0: SSU 运行 1: 停止向 SSU 提供时钟
1、0	—	全 1	R	保留位 读写的值总是 1。

## 26.3.4 待机控制寄存器 4 (STBCR4)

STBCR4 是 8 位可读写寄存器，控制低功耗时各模块的运行。

位:	7	6	5	4	3	2	1	0
	MSTP 23	MSTP 22	MSTP 21	-	-	MSTP 18	MSTP 17	MSTP 16
初始值:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7	MSTP23	1	R/W	模块停止位 23 如果将本位设定为 1，就停止向 MTU2S 提供时钟。 0: MTU2S 运行 1: 停止向 MTU2S 提供时钟
6	MSTP22	1	R/W	模块停止位 22 如果将本位设定为 1，就停止向 MTU2 提供时钟。 0: MTU2 运行 1: 停止向 MTU2 提供时钟
5	MSTP21	1	R/W	模块停止位 21 如果将本位设定为 1，就停止向 CMT 提供时钟。 0: CMT 运行 1: 停止向 CMT 提供时钟
4、3	—	全 1	R	保留位 读的值总是 1。
2	MSTP18	1	R/W	模块停止位 18 如果将本位设定为 1，就停止向 A/D_2 提供时钟。 0: A/D_2 运行 1: 停止向 A/D_2 提供时钟
1	MSTP17	1	R/W	模块停止位 17 如果将本位设定为 1，就停止向 A/D_1 提供时钟。 0: A/D_1 运行 1: 停止向 A/D_1 提供时钟
0	MSTP16	1	R/W	模块停止位 16 如果将本位设定为 1，就停止向 A/D_0 提供时钟。 0: A/D_0 运行 1: 停止向 A/D_0 提供时钟

## 26.3.5 待机控制寄存器 5 (STBCR5)

STBCR5 是 8 位可读写寄存器，指定低功耗模式的状态。

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	MSTP 25	MSTP 24
初始值:	0	0	0	0	0	0	1	1
R/W:	R	R	R	R	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
7 ~ 2	—	全 0	R	保留位 读写的值总是 0。
1	MSTP25	1	R/W	模块停止位 25 如果将本位设定为 1，就停止向 AUD 提供时钟。 0: AUD 运行 1: 停止向 AUD 提供时钟
0	MSTP24	1	R/W	模块停止位 24 如果将本位设定为 1，就停止向 UBC 提供时钟。 0: UBC 运行 1: 停止向 UBC 提供时钟

## 26.3.6 待机控制寄存器 6 (STBCR6)

STBCR6 是 8 位可读写寄存器，指定低功耗模式的状态。

位:	7	6	5	4	3	2	1	0
	AUD SRST	HIZ	-	-	-	-	STBY MD	-
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R/W	R

位	位名	初始值	R/W	说 明
7	AUDSRST	0	R/W	AUD 软件复位 通过软件控制 AUD 的复位。 如果给本位写 0，AUD 模块就进入上电复位状态。 0: 设定为 AUD 复位状态 1: 解除 AUD 的复位 将该位置 1 时，必须先将 STBCR5 的 MSTP25 位置 0。
6	HIZ	0	R/W	端口的高阻抗状态 选择在软件待机模式时是保持引脚状态还是设定为高阻抗状态。 0: 在软件待机模式时保持引脚状态 1: 在软件待机模式时将引脚状态设定为高阻抗
5 ~ 2	—	全 0	R	保留位 读写的值总是 0。
1	STBYMD	0	R/W	软件待机模式的选择 在 STBCR1 的 STBY 位为 1 的状态下，选择在执行 SLEEP 指令时是转移到软件待机模式还是转移到深度软件待机模式。 0: 转移到深度软件待机模式 1: 转移到软件待机模式
0	—	0	R	保留位 读写的值总是 0。



### 26.3.7 RAM 控制寄存器 (RAMCR)

RAMCR 是 8 位可读写寄存器，允许或者禁止对内部 RAM 的存取。

位:	7	6	5	4	3	2	1	0
	-	-	-	RAME	-	-	-	-
初始值:	0	0	0	1	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R

位	位名	初始值	R/W	说 明
7 ~ 5	—	全 0	R	保留位 读写的值总是 0。
4	RAME	1	R/W	RAM 允许 选择内部 RAM 有效或者无效。 0: 内部 RAM 无效 1: 内部 RAM 有效 如果将本位清 0，就不能对内部 RAM 进行存取。此时，读内部 RAM 和取指令的值就不定，忽视内部 RAM 的写操作。 在将本位清 0（将内部 RAM 设定为无效）时，在 RAMCR 的写指令之后不能紧接内部 RAM 的存取指令。否则，就不能保证正常的存取。 在将本位设定为 1（将内部 RAM 置为有效）时，在 RAMCR 的写指令之后必须紧接 RAMCR 的读指令。如果在 RAMCR 的写指令之后紧接内部 RAM 的存取指令，就不能保证正常的存取。
3 ~ 0	—	全 0	R	保留位 读写的值总是 0。

## 26.4 睡眠模式

### 26.4.1 向睡眠模式的转移

如果在 STBCR1 的 STBY 位为 0 的状态下执行 SLEEP 指令，就从程序的执行状态转移到睡眠模式，但是在释放总线的期间（将低电平输入到  $\overline{\text{BREQ}}$  引脚）不能转移到睡眠模式。CPU 在执行 SLEEP 指令后停止，但是保持 CPU 的寄存器内容，并且内部外围模块继续运行。

### 26.4.2 睡眠模式的解除

通过复位解除睡眠模式。

不要通过中断解除睡眠模式。

#### (1) 通过复位的解除

通过  $\overline{\text{RES}}$  引脚的上电复位、 $\overline{\text{MRES}}$  引脚的手动复位或者 WDT 的内部上电复位 / 内部手动复位，解除睡眠模式。

## 26.5 软件待机模式

### 26.5.1 向软件待机模式的转移

如果在 STBCR1 的 STBY 位和 STBCR6 的 STBYMD 位都为 1 的状态下执行 SLEEP 指令，就从程序的执行状态转移到软件待机模式，但是在释放总线的期间（将低电平输入到  $\overline{\text{BREQ}}$  引脚）不能转移到软件待机模式。另外，必须在停止 DMAC 和 DTC 的运行后执行 SLEEP 指令。在软件待机模式中，不仅 CPU 而且时钟和内部外围模块都停止。

保持 CPU 的寄存器内容和内部 RAM 的数据，内部外围模块的寄存器可能被初始化。有关软件待机模式中的外围模块的寄存器状态和引脚状态请分别参照“27.3 各运行模式的寄存器状态”和“附录 A. 引脚状态”。

转移到软件待机模式的步骤如下所示：

1. 将 WDT 定时器的控制寄存器（WTCSR）的 TME 位设定为 0，停止 WDT。
2. 将 WDT 定时器的计数器（WTCNT）设定为 0，并给 WTCSR 寄存器的 CKS2 ~ CKS0 位设定与指定的振荡稳定时间对应的值。
3. 当 DMAC 和 DTC 在运行时，停止 DMAC 和 DTC 的运行。
4. 在总线为释放状态（将低电平输入到  $\overline{\text{BREQ}}$  引脚）时，获取总线（将高电平输入到  $\overline{\text{BREQ}}$  引脚）。
5. 在将 STBCR1 的 STBY 位和 STBCR6 的 STBYMD 位设定为 1 后，执行 SLEEP 指令。
6. 进入软件待机模式，停止 LSI 内部的时钟。

### 26.5.2 软件待机模式的解除

通过中断（NMI、IRQ）或者复位，解除软件待机模式。

#### (1) 通过中断的解除

能通过内部 WDT 进行热启动。如果检测到 NMI 或者 IRQ 中断（边沿检测），就在经过 WDT 定时器的控制 / 状态寄存器设定的时间后给整个 LSI 提供时钟，并且解除软件待机模式。然后，执行中断异常处理。

但是，如果 IRQ 中断优先级低于 CPU 的状态寄存器（SR）设定的中断屏蔽级，就不接受中断也不解除软件待机模式。

另外，如果将 NMI 引脚设定为下降沿检测，就必须在转移到软件待机模式前将 NMI 引脚置为高电平；如果将 NMI 引脚设定为上升沿检测，就必须在转移到软件待机模式前将 NMI 引脚置为低电平。

同样地，如果将 IRQ 引脚设定为下降沿检测，就必须在转移到软件待机模式前将 IRQ 引脚置为高电平；如果将 IRQ 引脚设定为上升沿检测，就必须在转移到软件待机模式前将 IRQ 引脚置为低电平。

#### (2) 上电复位的解除

通过  $\overline{\text{RES}}$  引脚的上电复位，解除软件待机模式。 $\overline{\text{RES}}$  引脚必须在时钟振荡稳定前保持低电平。

#### (3) 手动复位的解除

通过  $\overline{\text{MRES}}$  引脚进行手动复位，就能解除软件待机模式。 $\overline{\text{MRES}}$  引脚在时钟振荡稳定前必须保持低电平。

## 26.6 深度软件待机模式

### 26.6.1 向深度软件待机模式的转移

如果在 STBCR1 的 STBY 位和 STBCR6 的 STBYMD 位分别为 1 和 0 的状态下执行 SLEEP 指令，就从程序的执行状态转移到深度软件待机模式，但是在释放总线的期间（将低电平输入到  $\overline{\text{BREQ}}$  引脚）不能转移到深度软件待机模式。另外，必须在停止 DMAC 和 DTC 的运行后执行 SLEEP 指令。在深度软件待机模式中，不仅 CPU 而且时钟和内部外围模块都停止，并且还截断本 LSI 的内部电源。

CPU 的寄存器内容和内部 RAM 的数据处于不定状态，内部外围模块的寄存器还被初始化。有关深度软件待机模式中的引脚状态请参照“附录 A. 引脚状态”。

转移到深度软件待机模式的步骤如下所示：

1. 将 WDT 定时器的控制寄存器（WTCSR）的 TME 位设定为 0，停止 WDT。
2. 当 DMAC 和 DTC 在运行时，停止 DMAC 和 DTC 的运行。
3. 在总线为释放状态（将低电平输入到  $\overline{\text{BREQ}}$  引脚）时，获取总线（将高电平输入到  $\overline{\text{BREQ}}$  引脚）。
4. 在将 STBCR1 的 STBY 位和 STBCR6 的 STBYMD 位分别设定为 1 和 0 后，执行 SLEEP 指令。
5. 进入深度软件待机模式，停止 LSI 内部的时钟，并且截断本 LSI 的内部电源。

### 26.6.2 深度软件待机模式的解除

通过  $\overline{\text{RES}}$  引脚进行上电复位，就能解除深度软件待机模式。 $\overline{\text{RES}}$  引脚在时钟振荡稳定前必须保持低电平。

## 26.7 模块待机功能

### 26.7.1 向模块待机功能的转移

能通过将待机控制寄存器 2～5（STBCR2～5）的各 MSTP 位设定为 1，停止向各自对应的内部外围模块提供时钟。通过使用此功能，能降低正常模式中的功耗。

不能存取被设定为模块待机模式的外围模块寄存器。有关模块待机模式中外围模块的寄存器状态，请参照“27.3 各运行模式的寄存器状态”。

### 26.7.2 模块待机功能的解除

通过将 STBCR2～5 的各 MSTP 位清 0，解除模块待机功能。对于 MSTP 位的初始值为 0 的模块，也能通过上电复位进行解除。

## 26.8 使用注意事项

### 26.8.1 振荡稳定待机中的功耗

在振荡稳定待机中，功耗会增加。

### 26.8.2 深度软件待机模式

请不要使用深度软件待机模式。

### 26.8.3 执行睡眠指令时

通过睡眠指令转移到睡眠模式或软件待机模式时，必须执行以下对策中的任何一个。

对策 A. 在执行睡眠指令前，必须使 DMAC/DTC 的运行停止以及设定为不发生由内部外围模块产生的中断、IRQ 中断、NMI 中断后，再执行睡眠指令。

对策 B. 在执行睡眠指令前，将 FRQCR 的值改写为初始值 H'36DB、并在虚读 2 次 FRQCR 后，再执行睡眠指令。

## 第 27 章 寄存器一览

寄存器地址一览表汇总了有关内部寄存器地址、位结构以及各运行模式状态的信息。表的记述方法如下：

### 1. 寄存器地址一览表（地址顺序）

- 从分配地址小的寄存器顺序记载。
- 在寄存器名称部用“—”表示保留的地址。  
不能存取保留的地址。
- 在地址为 16 位或者 32 位时，记载 MSB 侧的地址。
- 根据模块名分类。
- 表示存取长度。

### 2. 位结构一览表

- 按照“寄存器地址一览（地址顺序）”的顺序记载位结构。
- 在位名部用“—”表示保留位
- 表中位名为空白的栏，表示该寄存器全部分配给计数器或者数据。
- 在 16 位或者 32 位寄存器时，从 MSB 侧的位开始记载。

### 3. 各运行模式的寄存器状态

- 按照“寄存器地址一览（地址顺序）”的顺序记载位结构。
- 表示基本运行模式的寄存器状态。在有内部模块固有的复位等时，请参照内部模块的章节。

## 27.1 寄存器地址一览表（地址顺序）

存取长度表示位数。

存取状态数表示指定的基准时钟的状态数。

但是，这些值是 B:8 位存取时、W:16 位存取时和 L:32 位存取时的值。

【注】 禁止存取未定义或者保留的地址。由于不保证存取这些寄存器时的运行及其后续运行，因此禁止存取。

寄存器名	略称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
串行模式寄存器_0	SCSMR_0	8	H'FFFC000	SCI (通道 0)	8	P $\phi$ 基准 B:2	16 位
位速率寄存器_0	SCBRR_0	8	H'FFFC002		8		
串行控制寄存器_0	SCSCR_0	8	H'FFFC004		8		
发送数据寄存器_0	SCTDR_0	8	H'FFFC006		8		
串行状态寄存器_0	SCSSR_0	8	H'FFFC008		8		
接收数据寄存器_0	SCRDR_0	8	H'FFFC00A		8		
串行方向控制寄存器_0	SCSDCR_0	8	H'FFFC00C		8		
串行端口寄存器_0	SCSPTR_0	8	H'FFFC00E		8		
串行模式寄存器_1	SCSMR_1	8	H'FFFC080	SCI (通道 1)	8	P $\phi$ 基准 B:2	16 位
位速率寄存器_1	SCBRR_1	8	H'FFFC082		8		
串行控制寄存器_1	SCSCR_1	8	H'FFFC084		8		
发送数据寄存器_1	SCTDR_1	8	H'FFFC086		8		
串行状态寄存器_1	SCSSR_1	8	H'FFFC088		8		
接收数据寄存器_1	SCRDR_1	8	H'FFFC08A		8		
串行方向控制寄存器_1	SCSDCR_1	8	H'FFFC08C		8		
串行端口寄存器_1	SCSPTR_1	8	H'FFFC08E		8		
串行模式寄存器_2	SCSMR_2	8	H'FFFC100	SCI (通道 2)	8	P $\phi$ 基准 B:2	16 位
位速率寄存器_2	SCBRR_2	8	H'FFFC102		8		
串行控制寄存器_2	SCSCR_2	8	H'FFFC104		8		
发送数据寄存器_2	SCTDR_2	8	H'FFFC106		8		
串行状态寄存器_2	SCSSR_2	8	H'FFFC108		8		
接收数据寄存器_2	SCRDR_2	8	H'FFFC10A		8		
串行方向控制寄存器_2	SCSDCR_2	8	H'FFFC10C		8		
串行端口寄存器_2	SCSPTR_2	8	H'FFFC10E		8		
串行模式寄存器_3	SCSMR_3	16	H'FFFC180	SCIF (通道 3)	16	P $\phi$ 基准 B:2、W:2	16 位
位速率寄存器_3	SCBRR_3	8	H'FFFC182		8		
串行控制寄存器_3	SCSCR_3	16	H'FFFC184		16		
发送 FIFO 数据寄存器_3	SCFTDR_3	8	H'FFFC186		8		
串行状态寄存器_3	SCFSR_3	16	H'FFFC188		16		
接收 FIFO 数据寄存器_3	SCFRDR_3	8	H'FFFC18A		8		
FIFO 控制寄存器_3	SCFCR_3	16	H'FFFC18C		16		

寄存器名	略称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
FIFO 数据数寄存器_3	SCFDR_3	16	H'FFFFC18E	SCIF (通道 3)	16	P $\phi$ 基准 B:2、W:2	16 位
串行端口寄存器_3	SCSPTR_3	16	H'FFFFC190		16		
线路状态寄存器_3	SCLSR_3	16	H'FFFFC192		16		
定时器的控制寄存器_3	TCR_3	8	H'FFFFC200	MTU2	8、16、32	MP $\phi$ 基准 B:2、W:2、L:4	16 位
定时器的控制寄存器_4	TCR_4	8	H'FFFFC201		8		
定时器的模式寄存器_3	TMDR_3	8	H'FFFFC202		8、16		
定时器的模式寄存器_4	TMDR_4	8	H'FFFFC203		8		
定时器的 I/O 控制寄存器 H_3	TIORH_3	8	H'FFFFC204		8、16、32		
定时器的 I/O 控制寄存器 L_3	TIORL_3	8	H'FFFFC205		8		
定时器的 I/O 控制寄存器 H_4	TIORH_4	8	H'FFFFC206		8、16		
定时器的 I/O 控制寄存器 L_4	TIORL_4	8	H'FFFFC207		8		
定时器的中断允许寄存器_3	TIER_3	8	H'FFFFC208		8、16		
定时器的中断允许寄存器_4	TIER_4	8	H'FFFFC209		8		
定时器的输出主控允许寄存器	TOER	8	H'FFFFC20A		8		
定时器的门控寄存器	TGCR	8	H'FFFFC20D		8		
定时器的输出控制寄存器 1	TOCR1	8	H'FFFFC20E		8、16		
定时器的输出控制寄存器 2	TOCR2	8	H'FFFFC20F		8		
定时器的计数器_3	TCNT_3	16	H'FFFFC210		16、32		
定时器的计数器_4	TCNT_4	16	H'FFFFC212		16		
定时器的周期数据寄存器	TCDR	16	H'FFFFC214		16、32		
定时器的空载时间数据寄存器	TDDR	16	H'FFFFC216		16		
定时器的通用寄存器 A_3	TGRA_3	16	H'FFFFC218		16、32		
定时器的通用寄存器 B_3	TGRB_3	16	H'FFFFC21A		16		
定时器的通用寄存器 A_4	TGRA_4	16	H'FFFFC21C		16、32		
定时器的通用寄存器 B_4	TGRB_4	16	H'FFFFC21E		16		
定时器的子计数器	TCNTS	16	H'FFFFC220		16、32		
定时器的周期缓冲寄存器	TCBR	16	H'FFFFC222		16		
定时器的通用寄存器 C_3	TGRC_3	16	H'FFFFC224		16、32		
定时器的通用寄存器 D_3	TGRD_3	16	H'FFFFC226		16		
定时器的通用寄存器 C_4	TGRC_4	16	H'FFFFC228		16、32		
定时器的通用寄存器 D_4	TGRD_4	16	H'FFFFC22A		16		
定时器的状态寄存器_3	TSR_3	8	H'FFFFC22C		8、16		
定时器的状态寄存器_4	TSR_4	8	H'FFFFC22D		8		
定时器的跳过中断设定寄存器	TITCR	8	H'FFFFC230	8、16			
定时器的跳过中断次数计数器	TITCNT	8	H'FFFFC231	8			
定时器的缓冲传送设定寄存器	TBTER	8	H'FFFFC232	8			

寄存器名	略称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
定时器的空载时间允许寄存器	TDER	8	H'FFFFC234	MTU2	8	MP $\phi$ 基准 B:2、W:2、L:4	16 位
定时器的输出电平缓冲寄存器	TOLBR	8	H'FFFFC236		8		
定时器的缓冲运行传送模式寄存器_3	TBTM_3	8	H'FFFFC238		8、16		
定时器的缓冲运行传送模式寄存器_4	TBTM_4	8	H'FFFFC239		8		
定时器的 A/D 转换开始请求的控制寄存器	TADCR	16	H'FFFFC240		16		
定时器的 A/D 转换开始请求的周期设定寄存器 A_4	TADCORA_4	16	H'FFFFC244		16、32		
定时器的 A/D 转换开始请求的周期设定寄存器 B_4	TADCORB_4	16	H'FFFFC246		16		
定时器的 A/D 转换开始请求的周期设定缓冲寄存器 A_4	TADCOBRA_4	16	H'FFFFC248		16、32		
定时器的 A/D 转换开始请求的周期设定缓冲寄存器 B_4	TADCOBRB_4	16	H'FFFFC24A		16		
定时器的波形控制寄存器	TWCR	8	H'FFFFC260		8		
定时器的开始寄存器	TSTR	8	H'FFFFC280		8、16		
定时器的同步寄存器	TSYR	8	H'FFFFC281		8		
定时器的计数器同步开始寄存器	TCSYSTR	8	H'FFFFC282		8		
定时器的读写允许寄存器	TRWER	8	H'FFFFC284		8		
定时器的控制寄存器_0	TCR_0	8	H'FFFFC300		8、16、32		
定时器的模式寄存器_0	TMDR_0	8	H'FFFFC301		8		
定时器的 I/O 控制寄存器 H_0	TIORH_0	8	H'FFFFC302		8、16		
定时器的 I/O 控制寄存器 L_0	TIORL_0	8	H'FFFFC303		8		
定时器的中断允许寄存器_0	TIER_0	8	H'FFFFC304		8、16、32		
定时器的状态寄存器_0	TSR_0	8	H'FFFFC305		8		
定时器的计数器_0	TCNT_0	16	H'FFFFC306		16		
定时器的通用寄存器 A_0	TGRA_0	16	H'FFFFC308		16、32		
定时器的通用寄存器 B_0	TGRB_0	16	H'FFFFC30A		16		
定时器的通用寄存器 C_0	TGRC_0	16	H'FFFFC30C		16、32		
定时器的通用寄存器 D_0	TGRD_0	16	H'FFFFC30E		16		
定时器的通用寄存器 E_0	TGRE_0	16	H'FFFFC320		16、32		
定时器的通用寄存器 F_0	TGRF_0	16	H'FFFFC322		16		
定时器的中断允许寄存器 2_0	TIER2_0	8	H'FFFFC324		8、16		
定时器的状态寄存器 2_0	TSR2_0	8	H'FFFFC325		8		
定时器的缓冲运行传送模式寄存器_0	TBTM_0	8	H'FFFFC326		8		

寄存器名	略称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
定时器的控制寄存器_1	TCR_1	8	H'FFFC380	MTU2	8、16	MP $\phi$ 基准 B:2、W:2、L:4	16 位
定时器的模式寄存器_1	TMDR_1	8	H'FFFC381		8		
定时器的 I/O 控制寄存器_1	TIOR_1	8	H'FFFC382		8		
定时器的中断允许寄存器_1	TIER_1	8	H'FFFC384		8、16、32		
定时器的状态寄存器_1	TSR_1	8	H'FFFC385		8		
定时器的计数器_1	TCNT_1	16	H'FFFC386		16		
定时器的通用寄存器 A_1	TGRA_1	16	H'FFFC388		16、32		
定时器的通用寄存器 B_1	TGRB_1	16	H'FFFC38A		16		
定时器的输入捕捉控制寄存器	TICCR	8	H'FFFC390		8		
定时器的控制寄存器_2	TCR_2	8	H'FFFC400		8、16		
定时器的模式寄存器_2	TMDR_2	8	H'FFFC401		8		
定时器的 I/O 控制寄存器_2	TIOR_2	8	H'FFFC402		8		
定时器的中断允许寄存器_2	TIER_2	8	H'FFFC404		8、16、32		
定时器的状态寄存器_2	TSR_2	8	H'FFFC405		8		
定时器的计数器_2	TCNT_2	16	H'FFFC406		16		
定时器的通用寄存器 A_2	TGRA_2	16	H'FFFC408		16、32		
定时器的通用寄存器 B_2	TGRB_2	16	H'FFFC40A		16		
定时器的计数器 U_5	TCNTU_5	16	H'FFFC480		16、32		
定时器的通用寄存器 U_5	TGRU_5	16	H'FFFC482		16		
定时器的控制寄存器 U_5	TCRU_5	8	H'FFFC484		8		
定时器的 I/O 控制寄存器 U_5	TIORU_5	8	H'FFFC486		8		
定时器的计数器 V_5	TCNTV_5	16	H'FFFC490		16、32		
定时器的通用寄存器 V_5	TGRV_5	16	H'FFFC492		16		
定时器的控制寄存器 V_5	TCRV_5	8	H'FFFC494		8		
定时器的 I/O 控制寄存器 V_5	TIORV_5	8	H'FFFC496		8		
定时器的计数器 W_5	TCNTW_5	16	H'FFFC4A0		16、32		
定时器的通用寄存器 W_5	TGRW_5	16	H'FFFC4A2		16		
定时器的控制寄存器 W_5	TCRW_5	8	H'FFFC4A4		8		
定时器的 I/O 控制寄存器 W_5	TIORW_5	8	H'FFFC4A6		8		
定时器的状态寄存器_5	TSR_5	8	H'FFFC4B0		8		
定时器的中断允许寄存器_5	TIER_5	8	H'FFFC4B2	8			
定时器的开始寄存器_5	TSTR_5	8	H'FFFC4B4	8			
定时器的比较匹配清除寄存器	TCNTCMPCLR	8	H'FFFC4B6	8			
定时器的控制寄存器_3S	TCR_3S	8	H'FFFC600	MTU2S	8、16、32	MP $\phi$ 基准 B:2、W:2、L:4	
定时器的控制寄存器_4S	TCR_4S	8	H'FFFC601		8		
定时器的模式寄存器_3S	TMDR_3S	8	H'FFFC602		8、16		



寄存器名	略称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
定时器的模式寄存器_4S	TMDR_4S	8	H'FFFC603	MTU2S	8	MP <sub>φ</sub> 基准 B:2、W:2、L:4	16 位
定时器的 I/O 控制寄存器 H_3S	TIORH_3S	8	H'FFFC604		8、16、32		
定时器的 I/O 控制寄存器 L_3S	TIORL_3S	8	H'FFFC605		8		
定时器的 I/O 控制寄存器 H_4S	TIORH_4S	8	H'FFFC606		8、16		
定时器的 I/O 控制寄存器 L_4S	TIORL_4S	8	H'FFFC607		8		
定时器的中断允许寄存器_3S	TIER_3S	8	H'FFFC608		8、16		
定时器的中断允许寄存器_4S	TIER_4S	8	H'FFFC609		8		
定时器的输出主控允许寄存器 S	TOERS	8	H'FFFC60A		8		
定时器的门控寄存器 S	TGCRS	8	H'FFFC60D		8		
定时器的输出控制寄存器 1S	TOCR1S	8	H'FFFC60E		8、16		
定时器的输出控制寄存器 2S	TOCR2S	8	H'FFFC60F		8		
定时器的计数器_3S	TCNT_3S	16	H'FFFC610		16、32		
定时器的计数器_4S	TCNT_4S	16	H'FFFC612		16		
定时器的周期数据寄存器 S	TCDRS	16	H'FFFC614		16、32		
定时器的空载时间数据寄存器 S	TDDRS	16	H'FFFC616		16		
定时器的通用寄存器 A_3S	TGRA_3S	16	H'FFFC618		16、32		
定时器的通用寄存器 B_3S	TGRB_3S	16	H'FFFC61A		16		
定时器的通用寄存器 A_4S	TGRA_4S	16	H'FFFC61C		16、32		
定时器的通用寄存器 B_4S	TGRB_4S	16	H'FFFC61E		16		
定时器的子计数器 S	TCNTSS	16	H'FFFC620		16、32		
定时器的周期缓冲寄存器 S	TCBRS	16	H'FFFC622		16		
定时器的通用寄存器 C_3S	TGRC_3S	16	H'FFFC624		16、32		
定时器的通用寄存器 D_3S	TGRD_3S	16	H'FFFC626		16		
定时器的通用寄存器 C_4S	TGRC_4S	16	H'FFFC628		16、32		
定时器的通用寄存器 D_4S	TGRD_4S	16	H'FFFC62A		16		
定时器的状态寄存器_3S	TSR_3S	8	H'FFFC62C		8、16		
定时器的状态寄存器_4S	TSR_4S	8	H'FFFC62D		8		
定时器的跳过中断设定寄存器 S	TITCRS	8	H'FFFC630		8、16		
定时器的跳过中断次数计数器 S	TITCNTS	8	H'FFFC631	8			

寄存器名	略称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
定时器的缓冲传送设定寄存器 S	TBTERS	8	H'FFFFC632	MTU2S	8	MP $\phi$ 基准 B:2、W:2、L:4	16 位
定时器的空载时间允许寄存器 S	TDERS	8	H'FFFFC634		8		
定时器的输出电平缓冲寄存器 S	TOLBRS	8	H'FFFFC636		8		
定时器的缓冲运行传送模式寄存器_3S	TBTM_3S	8	H'FFFFC638		8、16		
定时器的缓冲运行传送模式寄存器_4S	TBTM_4S	8	H'FFFFC639		8		
定时器的 A/D 转换开始请求的控制寄存器 S	TADCRS	16	H'FFFFC640		16		
定时器的 A/D 转换开始请求的周期设定寄存器 A_4S	TADCORA_4S	16	H'FFFFC644		16、32		
定时器的 A/D 转换开始请求的周期设定寄存器 B_4S	TADCORB_4S	16	H'FFFFC646		16		
定时器的 A/D 转换开始请求的周期设定缓冲寄存器 A_4S	TADCOBRA_4S	16	H'FFFFC648		16、32		
定时器的 A/D 转换开始请求的周期设定缓冲寄存器 B_4S	TADCOBRB_4S	16	H'FFFFC64A		16		
定时器的同步清除寄存器 S	TSYCRS	8	H'FFFFC650		8		
定时器的波形控制寄存器 S	TWCRS	8	H'FFFFC660		8		
定时器的开始寄存器 S	TSTRS	8	H'FFFFC680		8、16		
定时器的同步寄存器 S	TSYRS	8	H'FFFFC681		8		
定时器的读写允许寄存器 S	TRWERS	8	H'FFFFC684		8		
定时器的计数器 U_5S	TCNTU_5S	16	H'FFFFC880		16、32		
定时器的通用寄存器 U_5S	TGRU_5S	16	H'FFFFC882		16		
定时器的控制寄存器 U_5S	TCRU_5S	8	H'FFFFC884		8		
定时器的 I/O 控制寄存器 U_5S	TIORU_5S	8	H'FFFFC886		8		
定时器的计数器 V_5S	TCNTV_5S	16	H'FFFFC890		16、32		
定时器的通用寄存器 V_5S	TGRV_5S	16	H'FFFFC892		16		
定时器的控制寄存器 V_5S	TCRV_5S	8	H'FFFFC894		8		
定时器的 I/O 控制寄存器 V_5S	TIORV_5S	8	H'FFFFC896		8		
定时器的计数器 W_5S	TCNTW_5S	16	H'FFFFC8A0		16、32		
定时器的通用寄存器 W_5S	TGRW_5S	16	H'FFFFC8A2		16		
定时器的控制寄存器 W_5S	TCRW_5S	8	H'FFFFC8A4		8		

寄存器名	略称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
定时器的 I/O 控制寄存器 W_5S	TIORW_5S	8	H'FFFFC8A6	MTU2S	8	MP $\phi$ 基准 B:2、W:2、L:4	16 位
定时器的状态寄存器_5S	TSR_5S	8	H'FFFFC8B0		8		
定时器的中断允许寄存器_5S	TIER_5S	8	H'FFFFC8B2		8		
定时器的开始寄存器_5S	TSTR_5S	8	H'FFFFC8B4		8		
定时器的比较匹配清除寄存器 S	TCNTCMPCLRS	8	H'FFFFC8B6		8		
A/D 数据寄存器 0	ADDR0	16	H'FFFFC900	A/D (通道 0)	16	P $\phi$ 基准 B:2、W:2	16 位
A/D 数据寄存器 1	ADDR1	16	H'FFFFC902		16		
A/D 数据寄存器 2	ADDR2	16	H'FFFFC904		16		
A/D 数据寄存器 3	ADDR3	16	H'FFFFC906		16		
A/D 控制 / 状态寄存器_0	ADCSR_0	16	H'FFFFC910		16		
A/D 控制寄存器_0	ADCR_0	16	H'FFFFC912		16		
A/D 数据寄存器 4	ADDR4	16	H'FFFFC980	A/D (通道 1)	16	P $\phi$ 基准 B:2、W:2	16 位
A/D 数据寄存器 5	ADDR5	16	H'FFFFC982		16		
A/D 数据寄存器 6	ADDR6	16	H'FFFFC984		16		
A/D 数据寄存器 7	ADDR7	16	H'FFFFC986		16		
A/D 控制 / 状态寄存器_1	ADCSR_1	16	H'FFFFC990		16		
A/D 控制寄存器_1	ADCR_1	16	H'FFFFC992		16		
A/D 数据寄存器 8	ADDR8	16	H'FFFFCA00	A/D (通道 2)	16	P $\phi$ 基准 B:2、W:2	16 位
A/D 数据寄存器 9	ADDR9	16	H'FFFFCA02		16		
A/D 数据寄存器 10	ADDR10	16	H'FFFFCA04		16		
A/D 数据寄存器 11	ADDR11	16	H'FFFFCA06		16		
A/D 数据寄存器 12	ADDR12	16	H'FFFFCA08		16		
A/D 数据寄存器 13	ADDR13	16	H'FFFFCA0A		16		
A/D 数据寄存器 14	ADDR14	16	H'FFFFCA0C		16		
A/D 数据寄存器 15	ADDR15	16	H'FFFFCA0E		16		
A/D 控制 / 状态寄存器_2	ADCSR_2	16	H'FFFFCA10		16		
A/D 控制寄存器_2	ADCR_2	16	H'FFFFCA12		16		
闪存代码控制 / 状态寄存器	FCCS	8	H'FFFFCC00	FLASH	8	P $\phi$ 基准 B:5	16 位
闪存编程代码选择寄存器	FPCS	8	H'FFFFCC01		8		
闪存擦除代码选择寄存器	FECS	8	H'FFFFCC02		8		
闪存键代码寄存器	FKEY	8	H'FFFFCC04		8		
闪存 MAT 选择寄存器	FMATS	8	H'FFFFCC05		8		
闪存传送目标地址寄存器	FTDAR	8	H'FFFFCC06		8		

寄存器名	略称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
DTC 允许寄存器 A	DTCERA	16	H'FFFFCC80	DTC	8、16	P $\phi$ 基准 B:2、W:2、 L:4	16 位
DTC 允许寄存器 B	DTCERB	16	H'FFFFCC82		8、16		
DTC 允许寄存器 C	DTCERC	16	H'FFFFCC84		8、16		
DTC 允许寄存器 D	DTCERD	16	H'FFFFCC86		8、16		
DTC 允许寄存器 E	DTCERE	16	H'FFFFCC88		8、16		
DTC 控制寄存器	DTCCR	8	H'FFFFCC90		8		
DTC 向量基址寄存器	DTCVBR	32	H'FFFFCC94		8、16、32		
I <sup>2</sup> C 总线控制寄存器 1	ICCR1	8	H'FFFFCD80	I <sup>2</sup> C2	8	P $\phi$ 基准 B:2	16 位
I <sup>2</sup> C 总线控制寄存器 2	ICCR2	8	H'FFFFCD81		8		
I <sup>2</sup> C 总线模式寄存器	ICMR	8	H'FFFFCD82		8		
I <sup>2</sup> C 总线中断允许寄存器	ICIER	8	H'FFFFCD83		8		
I <sup>2</sup> C 总线状态寄存器	ICSR	8	H'FFFFCD84		8		
从属地址寄存器	SAR	8	H'FFFFCD85		8		
I <sup>2</sup> C 总线发送数据寄存器	ICDRT	8	H'FFFFCD86		8		
I <sup>2</sup> C 总线接收数据寄存器	ICDRR	8	H'FFFFCD87		8		
NF2CYC 寄存器	NF2CYC	8	H'FFFFCD88		8		
SS 控制寄存器 H	SSCRH	8	H'FFFFCD00		SSU		
SS 控制寄存器 L	SSCRL	8	H'FFFFCD01	8			
SS 模式寄存器	SSMR	8	H'FFFFCD02	8、16			
SS 允许寄存器	SSER	8	H'FFFFCD03	8			
SS 状态寄存器	SSSR	8	H'FFFFCD04	8、16			
SS 控制寄存器 2	SSCR2	8	H'FFFFCD05	8			
SS 发送数据寄存器 0	SSTDR0	8	H'FFFFCD06	8、16			
SS 发送数据寄存器 1	SSTDR1	8	H'FFFFCD07	8			
SS 发送数据寄存器 2	SSTDR2	8	H'FFFFCD08	8、16			
SS 发送数据寄存器 3	SSTDR3	8	H'FFFFCD09	8			
SS 接收数据寄存器 0	SSRDR0	8	H'FFFFCD0A	8、16			
SS 接收数据寄存器 1	SSRDR1	8	H'FFFFCD0B	8			
SS 接收数据寄存器 2	SSRDR2	8	H'FFFFCD0C	8、16			
SS 接收数据寄存器 3	SSRDR3	8	H'FFFFCD0D	8			
比较匹配定时器的开始寄存器	CMSTR	16	H'FFFCE00	CMT		8、16、32	P $\phi$ 基准 B:2、W:2、 L:4
比较匹配定时器的控制 / 状态寄存器 _0	CMCSR_0	16	H'FFFCE02		8、16		
比较匹配计数器 _0	CMCNT_0	16	H'FFFCE04		8、16、32		
比较匹配常数寄存器 _0	CMCOR_0	16	H'FFFCE06		8、16		

寄存器名	略称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
比较匹配定时器的控制 / 状态寄存器 _1	CMCSR_1	16	H'FFFFCE08	CMT	8、16、32	P $\phi$ 基准 B:2、W:2、 L:4	16 位
比较匹配计数器 _1	CMCNT_1	16	H'FFFFCE0A		8、16		
比较匹配常数寄存器 _1	CMCOR_1	16	H'FFFFCE0C		8、16、32		
输入电平控制 / 状态寄存器 1	ICSR1	16	H'FFFFD000	POE	8、16、32	P $\phi$ 基准 B:2、W:2、 L:4	16 位
输出电平控制 / 状态寄存器 1	OCSR1	16	H'FFFFD002		8、16		
输入电平控制 / 状态寄存器 2	ICSR2	16	H'FFFFD004		8、16、32		
输出电平控制 / 状态寄存器 2	OCSR2	16	H'FFFFD006		8、16		
输入电平控制 / 状态寄存器 3	ICSR3	16	H'FFFFD008		8、16		
软件端口输出允许寄存器	SPOER	8	H'FFFFD00A		8		
端口输出允许控制寄存器 1	POECR1	8	H'FFFFD00B		8		
端口输出允许控制寄存器 2	POECR2	16	H'FFFFD00C		8、16		
端口 A 的数据寄存器 H	PADRH	16	H'FFFFD100		I/O		
端口 A 的数据寄存器 L	PADRL	16	H'FFFFD102	I/O	8、16		
端口 A 的 IO 寄存器 H	PAIORH	16	H'FFFFD104	PFC	8、16、32		
端口 A 的 IO 寄存器 L	PAIORL	16	H'FFFFD106		8、16		
端口 A 的控制寄存器 H4	PACRH4	16	H'FFFFD108		8、16、32		
端口 A 的控制寄存器 H3	PACRH3	16	H'FFFFD10A		8、16		
端口 A 的控制寄存器 H2	PACRH2	16	H'FFFFD10C		8、16、32		
端口 A 的控制寄存器 H1	PACRH1	16	H'FFFFD10E		8、16		
端口 A 的控制寄存器 L4	PACRL4	16	H'FFFFD110		8、16、32		
端口 A 的控制寄存器 L3	PACRL3	16	H'FFFFD112		8、16		
端口 A 的控制寄存器 L2	PACRL2	16	H'FFFFD114		8、16、32		
端口 A 的控制寄存器 L1	PACRL1	16	H'FFFFD116		8、16		
端口 A 的端口寄存器 H	PAPRH	16	H'FFFFD11C		I/O	8、16、32	
端口 A 的端口寄存器 L	PAPRL	16	H'FFFFD11E	8、16			
端口 B 的数据寄存器 L	PBDRL	16	H'FFFFD182	PFC	8、16		
端口 B 的 IO 寄存器 L	PBIORL	16	H'FFFFD186		8、16		
端口 B 的控制寄存器 L3	PBCRL3	16	H'FFFFD192		8、16		
端口 B 的控制寄存器 L2	PBCRL2	16	H'FFFFD194		8、16、32		
端口 B 的控制寄存器 L1	PBCRL1	16	H'FFFFD196		8、16		
端口 B 的端口寄存器 L	PBPRL	16	H'FFFFD19E	I/O	8、16		
端口 C 的数据寄存器 H	PCDRH	16	H'FFFFD200		8、16、32		
端口 C 的数据寄存器 L	PCDRL	16	H'FFFFD202		8、16		
端口 C 的 IO 寄存器 H	PCIORH	16	H'FFFFD204	PFC	8、16、32		
端口 C 的 IO 寄存器 L	PCIORL	16	H'FFFFD206		8、16		

寄存器名	略称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
端口 C 的控制寄存器 H3	PCCR3H	16	H'FFFFD20A	PFC	8、16	P $\phi$ 基准 B:2、W:2、 L:4	16 位
端口 C 的控制寄存器 H2	PCCR3H2	16	H'FFFFD20C		8、16、32		
端口 C 的控制寄存器 H1	PCCR3H1	16	H'FFFFD20E		8、16		
端口 C 的控制寄存器 L4	PCCRL4	16	H'FFFFD210		8、16、32		
端口 C 的控制寄存器 L3	PCCRL3	16	H'FFFFD212		8、16		
端口 C 的控制寄存器 L2	PCCRL2	16	H'FFFFD214		8、16、32		
端口 C 的控制寄存器 L1	PCCRL1	16	H'FFFFD216		8、16		
端口 C 的端口寄存器 H	PCPRH	16	H'FFFFD21C	I/O	8、16、32		
端口 C 的端口寄存器 L	PCPRL	16	H'FFFFD21E		8、16		
端口 D 的数据寄存器 H	PDDR3H	16	H'FFFFD280		8、16、32		
端口 D 的数据寄存器 L	PDDR3L	16	H'FFFFD282		8、16		
端口 D 的 IO 寄存器 H	PDIOR3H	16	H'FFFFD284	PFC	8、16、32		
端口 D 的 IO 寄存器 L	PDIOR3L	16	H'FFFFD286		8、16		
端口 D 的控制寄存器 H4	PDCR3H4	16	H'FFFFD288		8、16、32		
端口 D 的控制寄存器 H3	PDCR3H3	16	H'FFFFD28A		8、16		
端口 D 的控制寄存器 H2	PDCR3H2	16	H'FFFFD28C		8、16、32		
端口 D 的控制寄存器 H1	PDCR3H1	16	H'FFFFD28E		8、16		
端口 D 的控制寄存器 L4	PDCR3L4	16	H'FFFFD290		8、16、32		
端口 D 的控制寄存器 L3	PDCR3L3	16	H'FFFFD292	8、16			
端口 D 的控制寄存器 L2	PDCR3L2	16	H'FFFFD294	8、16、32			
端口 D 的控制寄存器 L1	PDCR3L1	16	H'FFFFD296	8、16			
端口 D 的端口寄存器 H	PDPR3H	16	H'FFFFD29C	I/O	8、16、32		
端口 D 的端口寄存器 L	PDPR3L	16	H'FFFFD29E		8、16		
端口 E 的数据寄存器 H	PEDR3H	16	H'FFFFD300		8、16、32		
端口 E 的数据寄存器 L	PEDR3L	16	H'FFFFD302		8、16		
端口 E 的 IO 寄存器 H	PEIOR3H	16	H'FFFFD304	PFC	8、16、32		
端口 E 的 IO 寄存器 L	PEIOR3L	16	H'FFFFD306		8、16		
端口 E 的控制寄存器 H2	PECR3H2	16	H'FFFFD30C		8、16、32		
端口 E 的控制寄存器 H1	PECR3H1	16	H'FFFFD30E		8、16		
端口 E 的控制寄存器 L4	PECR3L4	16	H'FFFFD310		8、16、32		
端口 E 的控制寄存器 L3	PECR3L3	16	H'FFFFD312		8、16		
端口 E 的控制寄存器 L2	PECR3L2	16	H'FFFFD314		8、16、32		
端口 E 的控制寄存器 L1	PECR3L1	16	H'FFFFD316	8、16			
端口 E 的端口寄存器 H	PEPR3H	16	H'FFFFD31C	I/O	8、16、32		
端口 E 的端口寄存器 L	PEPR3L	16	H'FFFFD31E		8、16		

寄存器名	略称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
大电流端口控制寄存器	HCPCR	16	H'FFFFD320	PFC	8、16、32	P $\phi$ 基准 B:2、W:2、 L:4	16 位
IRQOUT 功能控制寄存器	IFCR	16	H'FFFFD322		8、16		
端口 F 的数据寄存器 L	PFDR_L	16	H'FFFFD382	I/O	8、16		
频率控制寄存器	FRQCR	16	H'FFFFE800	CPG	16	P $\phi$ 基准 W:2	
待机控制寄存器 1	STBCR1	8	H'FFFFE802	低功耗	8	P $\phi$ 基准 B:2	16 位
待机控制寄存器 2	STBCR2	8	H'FFFFE804		8		
待机控制寄存器 3	STBCR3	8	H'FFFFE806		8		
待机控制寄存器 4	STBCR4	8	H'FFFFE808		8		
待机控制寄存器 5	STBCR5	8	H'FFFFE80A		8		
待机控制寄存器 6	STBCR6	8	H'FFFFE80C		8		
看门狗定时器的计数器	WTCNT	8	H'FFFFE810	WDT *1: 读时 *2: 写时	8*1、16*2	P $\phi$ 基准 B:2*1、W:2*2	16 位
看门狗定时器的控制 / 状态寄存器	WTCSR	8	H'FFFFE812		8*1、16*2		
振荡停止检测控制寄存器	OSCCR	8	H'FFFFE814	CPG	8	P $\phi$ 基准 B:2	16 位
RAM 控制寄存器	RAMCR	8	H'FFFFE880	低功耗	8	P $\phi$ 基准 B:2	16 位
A/D 触发选择寄存器 0	ADTSR_0	16	H'FFFFE890	A/D	8、16	P $\phi$ 基准 B:2、W:2	16 位
A/D 触发选择寄存器 1	ADTSR_1	16	H'FFFFE892		8、16		
总线功能扩展寄存器	BSCEHR	16	H'FFFFE89A	BSC	8、16	P $\phi$ 基准 B:2、W:2	16 位
中断控制寄存器 0	ICR0	16	H'FFFFE900	INTC	8、16	P $\phi$ 基准 B:2、W:2	16 位
IRQ 控制寄存器	IRQCR	16	H'FFFFE902		8、16		
IRQ 状态寄存器	IRQSR	16	H'FFFFE904		8、16		
中断优先级寄存器 A	IPRA	16	H'FFFFE906		8、16		
中断优先级寄存器 B	IPRB	16	H'FFFFE908		8、16		
中断优先级寄存器 C	IPRC	16	H'FFFFE980		16		
中断优先级寄存器 D	IPRD	16	H'FFFFE982		16		
中断优先级寄存器 E	IPRE	16	H'FFFFE984		16		
中断优先级寄存器 F	IPRF	16	H'FFFFE986		16		
中断优先级寄存器 H	IPRH	16	H'FFFFE98A		16		
中断优先级寄存器 I	IPRI	16	H'FFFFE98C		16		
中断优先级寄存器 J	IPRJ	16	H'FFFFE98E		16		
中断优先级寄存器 K	IPRK	16	H'FFFFE990		16		
中断优先级寄存器 L	IPRL	16	H'FFFFE992		16		
中断优先级寄存器 M	IPRM	16	H'FFFFE994	16			

寄存器名	略称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
DMA 源地址寄存器_0	SAR_0	32	H'FFFFEB20	DMAC	16、32	P $\phi$ 基准 B:2、W:2、 L:4	16 位
DMA 目标地址寄存器_0	DAR_0	32	H'FFFFEB24		16、32		
DMA 传送计数寄存器_0	DMATCR_0	32	H'FFFFEB28		16、32		
DMA 通道控制寄存器_0	CHCR_0	32	H'FFFFEB2C		8、16、32		
DMA 源地址寄存器_1	SAR_1	32	H'FFFFEB30		16、32		
DMA 目标地址寄存器_1	DAR_1	32	H'FFFFEB34		16、32		
DMA 传送计数寄存器_1	DMATCR_1	32	H'FFFFEB38		16、32		
DMA 通道控制寄存器_1	CHCR_1	32	H'FFFFEB3C		8、16、32		
DMA 源地址寄存器_2	SAR_2	32	H'FFFFEB40		16、32		
DMA 目标地址寄存器_2	DAR_2	32	H'FFFFEB44		16、32		
DMA 传送计数寄存器_2	DMATCR_2	32	H'FFFFEB48		16、32		
DMA 通道控制寄存器_2	CHCR_2	32	H'FFFFEB4C		8、16、32		
DMA 源地址寄存器_3	SAR_3	32	H'FFFFEB50		16、32		
DMA 目标地址寄存器_3	DAR_3	32	H'FFFFEB54		16、32		
DMA 传送计数寄存器_3	DMATCR_3	32	H'FFFFEB58		16、32		
DMA 通道控制寄存器_3	CHCR_3	32	H'FFFFEB5C		8、16、32		
DMA 操作寄存器	DMAOR	16	H'FFFFEB60		8、16		
通用控制寄存器	CMNCR	32	H'FFFFFF00	BSC	32	B $\phi$ 基准 L:2	16 位
CS0 空间总线控制寄存器	CS0BCR	32	H'FFFFFF04		32		
CS1 空间总线控制寄存器	CS1BCR	32	H'FFFFFF08		32		
CS2 空间总线控制寄存器	CS2BCR	32	H'FFFFFF0C		32		
CS3 空间总线控制寄存器	CS3BCR	32	H'FFFFFF10		32		
CS4 空间总线控制寄存器	CS4BCR	32	H'FFFFFF14		32		
CS5 空间总线控制寄存器	CS5BCR	32	H'FFFFFF18		32		
CS6 空间总线控制寄存器	CS6BCR	32	H'FFFFFF1C		32		
CS7 空间总线控制寄存器	CS7BCR	32	H'FFFFFF20		32		
CS8 空间总线控制寄存器	CS8BCR	32	H'FFFFFF24		32		
CS0 空间等待控制寄存器	CS0WCR	32	H'FFFFFF28		32		
CS1 空间等待控制寄存器	CS1WCR	32	H'FFFFFF2C		32		
CS2 空间等待控制寄存器	CS2WCR	32	H'FFFFFF30		32		
CS3 空间等待控制寄存器	CS3WCR	32	H'FFFFFF34		32		
CS4 空间等待控制寄存器	CS4WCR	32	H'FFFFFF38		32		
CS5 空间等待控制寄存器	CS5WCR	32	H'FFFFFF3C		32		
CS6 空间等待控制寄存器	CS6WCR	32	H'FFFFFF40		32		
CS7 空间等待控制寄存器	CS7WCR	32	H'FFFFFF44		32		
CS8 空间等待控制寄存器	CS8WCR	32	H'FFFFFF48		32		



寄存器名	略称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
SDRAM 控制寄存器	SDCR	32	H'FFFFFF04C	BSC	32	B $\phi$ 基准 L:2	16 位
刷新定时器的控制 / 状态寄存器	RTCSR	32	H'FFFFFF050		32		
刷新定时器的计数器	RTCNT	32	H'FFFFFF054		32		
刷新时间常数寄存器	RTCOR	32	H'FFFFFF058		32		
RAM 仿真寄存器	RAMER	16	H'FFFFFF108	FLASH	16	B $\phi$ 基准 W:2	16 位
断点地址寄存器 A	BARA	32	H'FFFFFF300	UBC	32	B $\phi$ 基准 B:2、W:2、 L:4	16 位
断点地址屏蔽寄存器 A	BAMRA	32	H'FFFFFF304		32		
断点总线周期寄存器 A	BBRA	16	H'FFFFFF308		16		
断点数据寄存器 A	BDRA	32	H'FFFFFF310		32		
断点数据屏蔽寄存器 A	BDMRA	32	H'FFFFFF314		32		
断点地址寄存器 B	BARB	32	H'FFFFFF320		32		
断点地址屏蔽寄存器 B	BAMRB	32	H'FFFFFF324		32		
断点总线周期寄存器 B	BBRB	16	H'FFFFFF328		16		
断点数据寄存器 B	BDRB	32	H'FFFFFF330		32		
断点数据屏蔽寄存器 B	BDMRB	32	H'FFFFFF334		32		
断点控制寄存器	BRCR	32	H'FFFFFF3C0		32		
转移源寄存器	BRSR	32	H'FFFFFF3D0		32		
转移目标寄存器	BRDR	32	H'FFFFFF3D4		32		
执行次数断点寄存器	BETR	16	H'FFFFFF3DC		16		

## 27.2 寄存器位一览

内部外围模块寄存器的地址和位名如下所示：

对于 16 位、32 位寄存器，分别以 8 位分 2 行或者 4 行表示。

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
SCSMR_0	C/ $\bar{A}$	CHR	PE	O/ $\bar{E}$	STOP	MP	CKS[1:0]		SCI (通道 0)
SCBRR_0									
SCSCR_0	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]		
SCTDR_0									
SCSSR_0	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
SCRDR_0									
SCSDCR_0	—	—	—	—	DIR	—	—	—	
SCSPTR_0	EIO	—	—	—	SPB1IO	SPB1DT	SPB0IO	SPB0DT	
SCSMR_1	C/ $\bar{A}$	CHR	PE	O/ $\bar{E}$	STOP	MP	CKS[1:0]		SCI (通道 1)
SCBRR_1									
SCSCR_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]		
SCTDR_1									
SCSSR_1	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
SCRDR_1									
SCSDCR_1	—	—	—	—	DIR	—	—	—	
SCSPTR_1	EIO	—	—	—	SPB1IO	SPB1DT	SPB0IO	SPB0DT	
SCSMR_2	C/ $\bar{A}$	CHR	PE	O/ $\bar{E}$	STOP	MP	CKS[1:0]		SCI (通道 2)
SCBRR_2									
SCSCR_2	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]		
SCTDR_2									
SCSSR_2	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
SCRDR_2									
SCSDCR_2	—	—	—	—	DIR	—	—	—	
SCSPTR_2	EIO	—	—	—	SPB1IO	SPB1DT	SPB0IO	SPB0DT	
SCSMR_3	—	—	—	—	—	—	—	—	SCIF (通道 3)
	C/ $\bar{A}$	CHR	PE	O/ $\bar{E}$	STOP	—	CKS[1:0]		
SCBRR_3									
SCSCR_3	—	—	—	—	—	—	—	—	
	TIE	RIE	TE	RE	REIE	—	CKE[1:0]		
SCFTDR_3									
SCFSR_3	PER[3:0]				FER[3:0]				
	ER	TEND	TDFE	BRK	FER	PER	RDF	DR	

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
SCFRDR_3									SCIF (通道 3)
SCFCR_3	—	—	—	—	—	RSTRG[2:0]			
	RTRG[1:0]		TTRG[1:0]		MCE	TFRST	RFRST	LOOP	
SCFDR_3	—	—	—	T[4:0]					
	—	—	—	R[4:0]					
SCSPTR_3	—	—	—	—	—	—	—	—	
	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPBIO	SPBDT	
SCLSR_3	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	ORER	
TCR_3	CCLR[2:0]		CKEG[1:0]		TPSC[2:0]			MTU2	
TCR_4	CCLR[2:0]		CKEG[1:0]		TPSC[2:0]				
TMDR_3	—	—	BFB	BFA	MD[3:0]				
TMDR_4	—	—	BFB	BFA	MD[3:0]				
TIORH_3	IOB[3:0]				IOA[3:0]				
TIORL_3	IOD[3:0]				IOC[3:0]				
TIORH_4	IOB[3:0]				IOA[3:0]				
TIORL_4	IOD[3:0]				IOC[3:0]				
TIER_3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB		TGIEA
TIER_4	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB		TGIEA
TOER	—	—	OE4D	OE4C	OE3D	OE4B	OE4A		OE3B
TGCR	—	BDC	N	P	FB	WF	VF		UF
TOCR1	—	PSYE	—	—	TOCL	TOCS	OLSN		OLSP
TOCR2	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N		OLS1P
TCNT_3									
TCNT_4									
TCDR									
TDDR									
TGRA_3									
TGRB_3									

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
TGRA_4									MTU2
TGRB_4									
TCNTS									
TCBR									
TGRC_3									
TGRD_3									
TGRC_4									
TGRD_4									
TSR_3	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TSR_4	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TITCR	T3AEN	3ACOR[2:0]			T4VEN	4VCOR[2:0]			
TITCNT	—	3ACNT[2:0]			—	4VCNT[2:0]			
TBTER	—	—	—	—	—	—	BTE[1:0]		
TDER	—	—	—	—	—	—	—	TDER	
TOLBR	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P	
TBTM_3	—	—	—	—	—	—	TTSB	TTSA	
TBTM_4	—	—	—	—	—	—	TTSB	TTSA	
TADCR	BF[1:0]		—	—	—	—	—	—	
	UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE	
TADCOBRA_4									
TADCOBRB_4									
TADCOBRA_4									
TADCOBRB_4									
TWCR	CCE	—	—	—	—	—	—	WRE	

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
TSTR	CST4	CST3	—	—	—	CST2	CST1	CST0	MTU2
TSYR	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0	
TCSYSTR	SCH0	SCH1	SCH2	SCH3	SCH4	—	SCH3S	SCH4S	
TRWER	—	—	—	—	—	—	—	RWE	
TCR_0	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]			
TMDR_0	—	BFE	BFB	BFA	MD[3:0]				
TIORH_0	IOB[3:0]				IOA[3:0]				
TIORL_0	IOD[3:0]				IOC[3:0]				
TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_0									
TGRA_0									
TGRB_0									
TGRC_0									
TGRD_0									
TGRE_0									
TGRF_0									
TIER2_0	TTGE2	—	—	—	—	—	TGIEF	TGIEE	
TSR2_0	—	—	—	—	—	—	TGFF	TGFE	
TBTM_0	—	—	—	—	—	TTSE	TTSB	TTSA	
TCR_1	—	CCLR[1:0]		CKEG[1:0]		TPSC[2:0]			
TMDR_1	—	—	—	—	MD[3:0]				
TIOR_1	IOB[3:0]				IOA[3:0]				
TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_1									
TGRA_1									

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块	
TGRB_1									MTU2	
TICCR	—	—	—	—	I2BE	I2AE	I1BE	I1AE		
TCR_2	—	CCLR[1:0]		CKEG[1:0]		TPSC[2:0]				
TMDR_2	—	—	—	—	MD[3:0]					
TIOR_2	IOB[3:0]				IOA[3:0]					
TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA		
TSR_2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA		
TCNT_2										
TGRA_2										
TGRB_2										
TCNTU_5										
TGRU_5										
TCRU_5	—	—	—	—	—	—	TPSC[1:0]			
TIORU_5	—	—	—	IOC[4:0]						
TCNTV_5										
TGRV_5										
TCRV_5	—	—	—	—	—	—	TPSC[1:0]			
TIORV_5	—	—	—	IOC[4:0]						
TCNTW_5										
TGRW_5										
TCRW_5	—	—	—	—	—	—	TPSC[1:0]			
TIORW_5	—	—	—	IOC[4:0]						
TSR_5	—	—	—	—	—	CMFU5	CMFV5	CMFW5		
TIER_5	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W		
TSTR_5	—	—	—	—	—	CSTU5	CSTV5	CSTW5		
TCNTCMLR	—	—	—	—	—	CMPCLR5U	CMPCLR5V	CMPCLR5W		

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
TCR_3S	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]			MTU2S
TCR_4S	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]			
TMDR_3S	—	—	BFB	BFA	MD[3:0]				
TMDR_4S	—	—	BFB	BFA	MD[3:0]				
TIORH_3S	IOB[3:0]				IOA[3:0]				
TIORL_3S	IOD[3:0]				IOC[3:0]				
TIORH_4S	IOB[3:0]				IOA[3:0]				
TIORL_4S	IOD[3:0]				IOC[3:0]				
TIER_3S	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TIER_4S	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TOERS	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B	
TGCRS	—	BDC	N	P	FB	WF	VF	UF	
TOCR1S	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP	
TOCR2S	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P	
TCNT_3S									
TCNT_4S									
TCDRS									
TDDRS									
TGRA_3S									
TGRB_3S									
TGRA_4S									
TGRB_4S									
TCNTSS									
TCBRS									
TGRC_3S									

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
TGRD_3S									MTU2S
TGRC_4S									
TGRD_4S									
TSR_3S	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TSR_4S	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TITCRS	T3AEN	3ACOR[2:0]			T4VEN	4VCOR[2:0]			
TITCNTS	—	3ACNT[2:0]			—	4VCNT[2:0]			
TBTERS	—	—	—	—	—	—	BTE[1:0]		
TDERS	—	—	—	—	—	—	—	TDER	
TOLBRS	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P	
TBTM_3S	—	—	—	—	—	—	TTSB	TTSA	
TBTM_4S	—	—	—	—	—	—	TTSB	TTSA	
TADCRS	BF[1:0]		—	—	—	—	—	—	
	UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE	
TADCORA_4S									
TADCORB_4S									
TADCOBRA_4S									
TADCOBRB_4S									
TSYCRS	CE0A	CE0B	CE0C	CE0D	CE1A	CE1B	CE2A	CE2B	
TWCRS	CCE	—	—	—	—	—	SCC	WRE	
TSTRS	CST4	CST3	—	—	—	CST2	CST1	CST0	
TSYRS	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0	
TRWERS	—	—	—	—	—	—	—	RWE	
TCNTU_5S									
TGRU_5S									
TCRU_5S	—	—	—	—	—	—	TPSC[1:0]		
TIORU_5S	—	—	—	IOC[4:0]					



寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
TCNTV_5S									MTU2S
TGRV_5S									
TCRV_5S	—	—	—	—	—	—	TPSC[1:0]		
TIORV_5S	—	—	—	IOC[4:0]					
TCNTW_5S									
TGRW_5S									
TCRW_5S	—	—	—	—	—	—	TPSC[1:0]		
TIORW_5S	—	—	—	IOC[4:0]					
TSR_5S	—	—	—	—	—	CMFU5	CMFV5	CMFW5	
TIER_5S	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W	
TSTR_5S	—	—	—	—	—	CSTU5	CSTV5	CSTW5	
TCNTCMPCLRS	—	—	—	—	—	CMPCLR5U	CMPCLR5V	CMPCLR5W	
ADDR0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D (通道 0)
	AD1	AD0	—	—	—	—	—	—	
ADDR1	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR2	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR3	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADCSR_0	ADF	ADIE	—	—	TRGE	—	CONADF	STC	
	CKSL[1:0]		ADM[1:0]		ADCS	CH[2:0]			
ADCR_0	—	—	ADST	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
ADDR4	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D (通道 1)
	AD1	AD0	—	—	—	—	—	—	
ADDR5	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR6	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR7	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
ADCSR_1	ADF	ADIE	—	—	TRGE	—	CONADF	STC	A/D (通道 1)
	CKSL[1:0]		ADM[1:0]		ADCS	CH[2:0]			
ADCR_1	—	—	ADST	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
ADDR8	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D (通道 2)
	AD1	AD0	—	—	—	—	—	—	
ADDR9	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR10	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR11	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR12	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR13	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR14	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR15	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADCSR_2	ADF	ADIE	—	—	TRGE	—	CONADF	STC	
	CKSL[1:0]		ADM[1:0]		ADCS	CH[2:0]			
ADCR_2	—	—	ADST	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
FCCS	FWE	MAT	—	FLER	—	—	—	SCO	FLASH
FPCS	—	—	—	—	—	—	—	PPVS	
FECS	—	—	—	—	—	—	—	EPVB	
FKEY	K[7:0]								
FMATS	MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0	
FTDAR	TDER	TDA[6:0]							
DTCERA	DTCERA15	DTCERA14	DTCERA13	DTCERA12	DTCERA11	DTCERA10	DTCERA9	DTCERA8	
	—	—	—	—	—	—	—	—	
DTCERB	DTCERB15	DTCERB14	DTCERB13	DTCERB12	DTCERB11	DTCERB10	DTCERB9	DTCERB8	
	DTCERB7	DTCERB6	DTCERB5	DTCERB4	DTCERB3	DTCERB2	DTCERB1	DTCERB0	
DTCERC	DTCERC15	DTCERC14	DTCERC13	DTCERC12	—	—	—	—	
	—	—	—	—	DTCERC3	DTCERC2	DTCERC1	DTCERC0	

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
DTCERD	DTCERD15	DTCERD14	DTCERD13	DTCERD12	DTCERD11	DTCERD10	DTCERD9	DTCERD8	DTC
	DTCERD7	DTCERD6	DTCERD5	DTCERD4	DTCERD3	—	—	—	
DTCERE	DTCERE15	DTCERE14	DTCERE13	DTCERE12	DTCERE11	DTCERE10	DTCERE9	DTCERE8	
	DTCERE7	DTCERE6	DTCERE5	DTCERE4	—	—	—	—	
DTCCR	—	—	—	RRS	RCHNE	—	—	ERR	
DTCVBR									
					—	—	—	—	
	—	—	—	—	—	—	—	—	
ICCR1	ICE	RCVD	MST	TRS	CKS[3:0]			I <sup>2</sup> C2	
ICCR2	BBSY	SCP	SDAO	SDAOP	SCLO	—	IICRST		
ICMR	MLS	WAIT	—	—	BCWP	BC[2:0]			
ICIER	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR		ACKBT
ICSR	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS		ADZ
SAR	SVA[6:0]								FS
ICDRT									
ICDRR									
NF2CYC	—	—	—	—	—	—	—		NF2CYC
SSCRH	MSS	BIDE	—	SOL	SOLP	—	CSS[1:0]		SSU
SSCRL	FCLRM	SSUMS	SRES	—	—	—	DATS[1:0]		
SSMR	MLS	CPOS	CPHS	—	—	CKS[2:0]			
SSER	TE	RE	—	—	TEIE	TIE	RIE	CEIE	
SSSR	—	ORER	—	—	TEND	TDRE	RDRF	CE	
SSCR2	—	—	—	TENDSTS	SCSATS	SSODTS	—	—	
SSTDR0									
SSTDR1									
SSTDR2									
SSTDR3									
SSRDR0									
SSRDR1									
SSRDR2									
SSRDR3									
CMSTR	—	—	—	—	—	—	—	—	CMT
	—	—	—	—	—	—	STR1	STR0	
CMCSR_0	—	—	—	—	—	—	—	—	
	CMF	CMIE	—	—	—	—	CKS[1:0]		

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
CMCNT_0									CMT
CMCOR_0									
CMCSR_1	—	—	—	—	—	—	—	—	
	CMF	CMIE	—	—	—	—	CKS[1:0]		
CMCNT_1									
CMCOR_1									
ICSR1	POE3F	POE2F	POE1F	POE0F	—	—	—	PIE1	POE
	POE3M[1:0]		POE2M[1:0]		POE1M[1:0]		POE0M[1:0]		
OCSR1	OSF1	—	—	—	—	—	OCE1	OIE1	
	—	—	—	—	—	—	—	—	
ICSR2	POE7F	POE6F	POE5F	POE4F	—	—	—	PIE2	
	POE7M[1:0]		POE6M[1:0]		POE5M[1:0]		POE4M[1:0]		
OCSR2	OSF2	—	—	—	—	—	OCE2	OIE2	
	—	—	—	—	—	—	—	—	
ICSR3	—	—	—	POE8F	—	—	POE8E	PIE3	
	—	—	—	—	—	—	POE8M[1:0]		
SPOER	—	—	—	—	—	MTU2SHIZ	MTU2CH0HIZ	MTU2CH34HIZ	
POECSR1	—	—	—	—	MTU2PE3ZE	MTU2PE2ZE	MTU2PE1ZE	MTU2PE0ZE	
POECSR2	—	MTU2P1CZE	MTU2P2CZE	MTU2P3CZE	—	MTU2SP1CZE	MTU2SP2CZE	MTU2SP3CZE	
	—	MTU2SP4CZE	MTU2SP5CZE	MTU2SP6CZE	—	MTU2SP7CZE	MTU2SP8CZE	MTU2SP9CZE	
PADRH*1	—	—	PA29DR	PA28DR	PA27DR	PA26DR	PA25DR	PA24DR	I/O
	PA23DR	PA22DR	PA21DR	PA20DR	PA19DR	PA18DR	PA17DR	PA16DR	
PADRL*1	PA15DR	PA14DR	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR	PA8DR	
	PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR	
PAIORH	—	—	PA29IOR	PA28IOR	PA27IOR	PA26IOR	PA25IOR	PA24IOR	PFC
	PA23IOR	PA22IOR	PA21IOR	PA20IOR	PA19IOR	PA18IOR	PA17IOR	PA16IOR	
PAIORL	PA15IOR	PA14IOR	PA13IOR	PA12IOR	PA11IOR	PA10IOR	PA9IOR	PA8IOR	
	PA7IOR	PA6IOR	PA5IOR	PA4IOR	PA3IOR	PA2IOR	PA1IOR	PA0IOR	
PACRH4*1	—	—	—	—	—	—	—	—	
	—	—	PA29MD1	PA29MD0	—	—	PA28MD1	PA28MD0	
PACRH3*1	—	—	PA27MD1	PA27MD0	—	—	PA26MD1	PA26MD0	
	—	—	PA25MD1	PA25MD0	—	—	PA24MD1	PA24MD0	

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块	
PACRH2*1	—	—	PA23MD1	PA23MD0	—	—	PA22MD1	PA22MD0	PFC	
	—	—	PA21MD1	PA21MD0	—	—	PA20MD1	PA20MD0		
PACRH1*1	—	—	PA19MD1	PA19MD0	—	—	PA18MD1	PA18MD0		
	—	—	PA17MD1	PA17MD0	—	PA16MD2	PA16MD1	PA16MD0		
PACRL4*1	—	PA15MD2	PA15MD1	PA15MD0	—	PA14MD2	PA14MD1	PA14MD0		
	—	PA13MD2	PA13MD1	PA13MD0	—	PA12MD2	PA12MD1	PA12MD0		
PACRL3*1	—	PA11MD2	PA11MD1	PA11MD0	—	PA10MD2	PA10MD1	PA10MD0		
	—	PA9MD2	PA9MD1	PA9MD0	—	PA8MD2	PA8MD1	PA8MD0		
PACRL2*1	—	PA7MD2	PA7MD1	PA7MD0	—	PA6MD2	PA6MD1	PA6MD0		
	—	PA5MD2	PA5MD1	PA5MD0	—	PA4MD2	PA4MD1	PA4MD0		
PACRL1*1	—	PA3MD2	PA3MD1	PA3MD0	—	PA2MD2	PA2MD1	PA2MD0		
	—	PA1MD2	PA1MD1	PA1MD0	—	PA0MD2	PA0MD1	PA0MD0		
PAPRH*1	—	—	PA29PR	PA28PR	PA27PR	PA26PR	PA25PR	PA24PR		I/O
	PA23PR	PA22PR	PA21PR	PA20PR	PA19PR	PA18PR	PA17PR	PA16PR		
PAPRL*1	PA15PR	PA14PR	PA13PR	PA12PR	PA11PR	PA10PR	PA9PR	PA8PR		
	PA7PR	PA6PR	PA5PR	PA4PR	PA3PR	PA2PR	PA1PR	PA0PR		
PBDRL*1	—	—	—	—	—	—	PB9DR	PB8DR		
	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR		
PBIORL	—	—	—	—	—	—	PB9IOR	PB8IOR	PFC	
	PB7IOR	PB6IOR	PB5IOR	PB4IOR	PB3IOR	PB2IOR	PB1IOR	PB0IOR		
PBCRL3	—	—	—	—	—	—	—	—		
	—	PB9MD2	PB9MD1	PB9MD0	—	PB8MD2	PB8MD1	PB8MD0		
PBCRL2	—	PB7MD2	PB7MD1	PB7MD0	—	PB6MD2	PB6MD1	PB6MD0		
	—	PB5MD2	PB5MD1	PB5MD0	—	PB4MD2	PB4MD1	PB4MD0		
PBCRL1	—	PB3MD2	PB3MD1	PB3MD0	—	PB2MD2	PB2MD1	PB2MD0		
	—	PB1MD2	PB1MD1	PB1MD0	—	PB0MD2	PB0MD1	PB0MD0		
PBPRL	—	—	—	—	—	—	PB9PR	PB8PR		I/O
	PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	PB0PR		
PCDRH	—	—	—	—	—	—	PC25DR	PC24DR		
	PC23DR	PC22DR	PC21DR	PC20DR	PC19DR	PC18DR	—	—		
PCDRL	PC15DR	PC14DR	PC13DR	PC12DR	PC11DR	PC10DR	PC9DR	PC8DR		
	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR		
PCIORH	—	—	—	—	—	—	PC25IOR	PC24IOR		PFC
	PC23IOR	PC22IOR	PC21IOR	PC20IOR	PC19IOR	PC18IOR	—	—		
PCIORL	PC15IOR	PC14IOR	PC13IOR	PC12IOR	PC11IOR	PC10IOR	PC9IOR	PC8IOR		
	PC7IOR	PC6IOR	PC5IOR	PC4IOR	PC3IOR	PC2IOR	PC1IOR	PC0IOR		

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块	
PCCR3*1	—	—	—	—	—	—	—	—	PFC	
	—	—	—	PC25MD0	—	—	—	PC24MD0		
PCCR2*1	—	—	—	PC23MD0	—	—	—	PC22MD0		
	—	—	—	PC21MD0	—	—	—	PC20MD0		
PCCR1*1	—	—	—	PC19MD0	—	—	—	PC18MD0		
	—	—	—	—	—	—	—	—		
PCCRL4	—	—	—	PC15MD0	—	—	—	PC14MD0		
	—	—	—	PC13MD0	—	—	—	PC12MD0		
PCCRL3	—	—	—	PC11MD0	—	—	—	PC10MD0		
	—	—	—	PC9MD0	—	—	—	PC8MD0		
PCCRL2	—	—	—	PC7MD0	—	—	—	PC6MD0		
	—	—	—	PC5MD0	—	—	—	PC4MD0		
PCCRL1	—	—	—	PC3MD0	—	—	—	PC2MD0		
	—	—	—	PC1MD0	—	—	—	PC0MD0		
PCPRH*1	—	—	—	—	—	—	PC25PR	PC24PR		I/O
	PC23PR	PC22PR	PC21PR	PC20PR	PC19PR	PC18PR	—	—		
PCPRL	PC15PR	PC14PR	PC13PR	PC12PR	PC11PR	PC10PR	PC9PR	PC8PR		
	PC7PR	PC6PR	PC5PR	PC4PR	PC3PR	PC2PR	PC1PR	PC0PR		
PDDRH*1	PD31DR	PD30DR	PD29DR	PD28DR	PD27DR	PD26DR	PD25DR	PD24DR		
	PD23DR	PD22DR	PD21DR	PD20DR	PD19DR	PD18DR	PD17DR	PD16DR		
PDDRL	PD15DR	PD14DR	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR	PD8DR		
	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR		
PDIORH	PD31IOR	PD30IOR	PD29IOR	PD28IOR	PD27IOR	PD26IOR	PD25IOR	PD24IOR	PFC	
	PD23IOR	PD22IOR	PD21IOR	PD20IOR	PD19IOR	PD18IOR	PD17IOR	PD16IOR		
PDIORL	PD15IOR	PD14IOR	PD13IOR	PD12IOR	PD11IOR	PD10IOR	PD9IOR	PD8IOR		
	PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR		
PDCRH4*1	—	—	PD31MD1	PD31MD0	—	—	PD30MD1	PD30MD0		
	—	—	PD29MD1	PD29MD0	—	—	PD28MD1	PD28MD0		
PDCRH3*1	—	—	PD27MD1	PD27MD0	—	—	PD26MD1	PD26MD0		
	—	—	PD25MD1	PD25MD0	—	—	PD24MD1	PD24MD0		
PDCRH2*1	—	—	PD23MD1	PD23MD0	—	PD22MD2	PD22MD1	PD22MD0		
	—	PD21MD2	PD21MD1	PD21MD0	—	PD20MD0	PD20MD1	PD20MD0		
PDCRH1*1	—	PD19MD2	PD19MD1	PD19MD0	—	PD18MD2	PD18MD1	PD18MD0		
	—	PD17MD2	PD17MD1	PD17MD0	—	PD16MD2	PD16MD1	PD16MD0		
PDCRL4	—	—	PD15MD1	PD15MD0	—	—	PD14MD1	PD14MD0		
	—	—	PD13MD1	PD13MD0	—	—	PD12MD1	PD12MD0		

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
PDCRL3	—	—	PD11MD1	PD11MD0	—	PD10MD2	PD10MD1	PD10MD0	PFC
	—	PD9MD2	PD9MD1	PD9MD0	—	PD8MD2	PD8MD1	PD8MD0	
PDCRL2	—	PD7MD2	PD7MD1	PD7MD0	—	PD6MD2	PD6MD1	PD6MD0	
	—	PD5MD2	PD5MD1	PD5MD0	—	PD4MD2	PD4MD1	PD4MD0	
PDCRL1	—	PD3MD2	PD3MD1	PD3MD0	—	PD2MD2	PD2MD1	PD2MD0	
	—	PD1MD2	PD1MD1	PD1MD0	—	PD0MD2	PD0MD1	PD0MD0	
PDPRH*1	PD31PR	PD30PR	PD29PR	PD28PR	PD27PR	PD26PR	PD25PR	PD24PR	I/O
	PD23PR	PD22PR	PD21PR	PD20PR	PD19PR	PD18PR	PD17PR	PD16PR	
PDPRL	PD15PR	PD14PR	PD13PR	PD12PR	PD11PR	PD10PR	PD9PR	PD8PR	
	PD7PR	PD6PR	PD5PR	PD4PR	PD3PR	PD2PR	PD1PR	PD0PR	
PEDRH*1	—	—	—	—	—	—	—	—	
	—	—	PE21DR	PE20DR	PE19DR	PE18DR	PE17DR	PE16DR	
PEDRL*1	PE15DR	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR	
	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR	
PEIORH	—	—	—	—	—	—	—	—	PFC
	—	—	PE21IOR	PE20IOR	PE19IOR	PE18IOR	PE17IOR	PE16IOR	
PEIOLR	PE15IOR	PE14IOR	PE13IOR	PE12IOR	PE11IOR	PE10IOR	PE9IOR	PE8IOR	
	PE7IOR	PE6IOR	PE5IOR	PE4IOR	PE3IOR	PE2IOR	PE1IOR	PE0IOR	
PECRH2*1	—	—	—	—	—	—	—	—	
	—	—	PE21MD1	PE21MD0	—	—	PE20MD1	PE20MD0	
PECRH1*1	—	—	PE19MD1	PE19MD0	—	—	PE18MD1	PE18MD0	
	—	—	PE17MD1	PE17MD0	—	PE16MD2	PE16MD1	PE16MD0	
PECRL4	—	PE15MD2	PE15MD1	PE15MD0	—	PE14MD2	PE14MD1	PE14MD0	
	—	—	PE13MD1	PE13MD0	—	PE12MD2	PE12MD1	PE12MD0	
PECRL3*1	—	PE11MD2	PE11MD1	PE11MD0	—	PE10MD2	PE10MD1	PE10MD0	
	—	PE9MD2	PE9MD1	PE9MD0	—	PE8MD2	PE8MD1	PE8MD0	
PECRL2*1	—	PE7MD2	PE7MD1	PE7MD0	—	PE6MD2	PE6MD1	PE6MD0	
	—	PE5MD2	PE5MD1	PE5MD0	—	PE4MD2	PE4MD1	PE4MD0	
PECRL1	—	PE3MD2	PE3MD1	PE3MD0	—	PE2MD2	PE2MD1	PE2MD0	
	—	PE1MD2	PE1MD1	PE1MD0	—	—	PE0MD1	PE0MD0	
PEPRH*1	—	—	—	—	—	—	—	—	I/O
	—	—	PE21PR	PE20PR	PE19PR	PE18PR	PE17PR	PE16PR	
PEPRL*1	PE15PR	PE14PR	PE13PR	PE12PR	PE11PR	PE10PR	PE9PR	PE8PR	
	PE7PR	PE6PR	PE5PR	PE4PR	PE3PR	PE2PR	PE1PR	PE0PR	
HCPCR	—	—	—	—	—	—	—	—	PFC
	—	—	—	—	MZIZDH	MZIZDL	MZIZEH	MZIZEL	

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
IFCR	—	—	—	—	—	—	—	—	PFC
	—	—	—	—	IRQMD3	IRQMD2	IRQMD1	IRQMD0	
PFDR <sup>*1</sup>	PF15DR	PF14DR	PF13DR	PF12DR	PF11DR	PF10DR	PF9DR	PF8DR	I/O
	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR	
FRQCR	—	IFC[2:0]			BFC[2:0]			PFC[2]	CPG
	PFC[1:0]		MIFC[2:0]			MPFC[2:0]			
STBCR1	STBY	—	—	—	—	—	—	—	低功耗
STBCR2	MSTP7	MSTP6	—	MSTP4	MSTP3	—	—	—	
STBCR3	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	—	—	
STBCR4	MSTP23	MSTP22	MSTP21	—	—	MSTP18	MSTP17	MSTP16	
STBCR5	—	—	—	—	—	—	MSTP25	MSTP24	
STBCR6	AUDSRST	HIZ	—	—	—	—	STBYMD	—	
WTCNT									WDT
WTCSR	TME	WT/IT	RSTS	WOVF	IOVF	CKS[2:0]			
OSCCR	—	—	—	—	—	OSCSTOP	—	OSCERS	CPG
RAMCR	—	—	—	RAME	—	—	—	—	低功耗
ADTSR_0	TRG11S[3:0]				TRG01S[3:0]				A/D
	TRG1S[3:0]				TRG0S[3:0]				
ADTSR_1	TRG2S[3:0]				—	—	—	—	
	—	—	—	—	—	—	—	—	
BSCEHR	DTLOCK	CSSTP1	—	CSSTP2	DTBST	DTSA	CSSTP3	DTPR	BSC
	—	—	—	DMMTU4	DMMTU3	DMMTU2	DMMTU1	DMMTU0	
ICR0	NMIL	—	—	—	—	—	—	NMIE	INTC
	—	—	—	—	—	—	—	—	
IRQCR	IRQ71S	IRQ70S	IRQ61S	IRQ60S	IRQ51S	IRQ50S	IRQ41S	IRQ40S	
	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S	
IRQSR	IRQ7L	IRQ6L	IRQ5L	IRQ4L	IRQ3L	IRQ2L	IRQ1L	IRQ0L	
	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F	
IPRA	IRQ0	IRQ0	IRQ0	IRQ0	IRQ1	IRQ1	IRQ1	IRQ1	
	IRQ2	IRQ2	IRQ2	IRQ2	IRQ3	IRQ3	IRQ3	IRQ3	
IPRB	IRQ4	IRQ4	IRQ4	IRQ4	IRQ5	IRQ5	IRQ5	IRQ5	
	IRQ6	IRQ6	IRQ6	IRQ6	IRQ7	IRQ7	IRQ7	IRQ7	
IPRC	DMAC_0	DMAC_0	DMAC_0	DMAC_0	DMAC_1	DMAC_1	DMAC_1	DMAC_1	
	DMAC_2	DMAC_2	DMAC_2	DMAC_2	DMAC_3	DMAC_3	DMAC_3	DMAC_3	
IPRD	MTU2_0	MTU2_0	MTU2_0	MTU2_0	MTU2_0	MTU2_0	MTU2_0	MTU2_0	
	MTU2_1	MTU2_1	MTU2_1	MTU2_1	MTU2_1	MTU2_1	MTU2_1	MTU2_1	



寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
IPRE	MTU2_2	MTU2_2	MTU2_2	MTU2_2	MTU2_2	MTU2_2	MTU2_2	MTU2_2	INTC
	MTU2_3	MTU2_3	MTU2_3	MTU2_3	MTU2_3	MTU2_3	MTU2_3	MTU2_3	
IPRF	MTU2_4	MTU2_4	MTU2_4	MTU2_4	MTU2_4	MTU2_4	MTU2_4	MTU2_4	
	MTU2_5	MTU2_5	MTU2_5	MTU2_5	POE(MTU2)	POE(MTU2)	POE(MTU2)	POE(MTU2)	
IPRH	—	—	—	—	I <sup>2</sup> C2	I <sup>2</sup> C2	I <sup>2</sup> C2	I <sup>2</sup> C2	
	MTU2S_3	MTU2S_3	MTU2S_3	MTU2S_3	MTU2S_3	MTU2S_3	MTU2S_3	MTU2S_3	
IPRI	MTU2S_4	MTU2S_4	MTU2S_4	MTU2S_4	MTU2S_4	MTU2S_4	MTU2S_4	MTU2S_4	
	MTU2S_5	MTU2S_5	MTU2S_5	MTU2S_5	POE(MTU2S)	POE(MTU2S)	POE(MTU2S)	POE(MTU2S)	
IPRJ	CMT_0	CMT_0	CMT_0	CMT_0	CMT_1	CMT_1	CMT_1	CMT_1	
	BSC	BSC	BSC	BSC	WDT	WDT	WDT	WDT	
IPRK	A/D_0,1	A/D_0,1	A/D_0,1	A/D_0,1	A/D_2	A/D_2	A/D_2	A/D_2	
	—	—	—	—	—	—	—	—	
IPRL	SCI_0	SCI_0	SCI_0	SCI_0	SCI_1	SCI_1	SCI_1	SCI_1	
	SCI_2	SCI_2	SCI_2	SCI_2	SCIF	SCIF	SCIF	SCIF	
IPRM	SSU	SSU	SSU	SSU	I <sup>2</sup> C2	I <sup>2</sup> C2	I <sup>2</sup> C2	I <sup>2</sup> C2	
	—	—	—	—	—	—	—	—	
SAR_0									DMAC
DAR_0									
DMATCR_0									
CHCR_0	—	—	—	—	—	—	—	—	
	DO	TL	—	—	—	—	AM	AL	
	DM[1:0]		SM[1:0]		RS[3:0]				
	DL	DS	TB	TS[1:0]		IE	TE	DE	
SAR_1									

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
DAR_1									DMAC
DMATCR_1									
CHCR_1	—	—	—	—	—	—	—	—	
	DO	TL	—	—	—	—	AM	AL	
	DM[1:0]		SM[1:0]		RS[3:0]				
	DL	DS	TB	TS[1:0]		IE	TE	DE	
SAR_2									
DAR_2									
DMATCR_2									
CHCR_2	—	—	—	—	—	—	—	—	
	DO	TL	—	—	—	—	AM	AL	
	DM[1:0]		SM[1:0]		RS[3:0]				
	DL	DS	TB	TS[1:0]		IE	TE	DE	
SAR_3									
DAR_3									

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
DMATCR_3									DMAC
CHCR_3	—	—	—	—	—	—	—	—	
	DO	TL	—	—	—	—	AM	AL	
	DM[1:0]		SM[1:0]		RS[3:0]				
	DL	DS	TB	TS[1:0]		IE	TE	DE	
DMAOR	—	—	CMS[1:0]		—	—	PR[1:0]		
	—	—	—	—	—	AE	NMIF	DME	
CMNCR	—	—	—	—	—	—	—	—	BSC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—		—	—	
	DMAIW[1:0]		DMAIWA	—	—	—	HIZMEM	HIZCNT	
CS0BCR	—	—	IWW[1:0]		—	IWRWD[1:0]		—	
	IWRWS[1:0]		—	IWRRD[1:0]		—	IWRRS[1:0]		
	—	TYPE[2:0]			—	BSZ[1:0]		—	
	—	—	—	—	—	—	—	—	
CS1BCR	—	—	IWW[1:0]		—	IWRWD[1:0]		—	
	IWRWS[1:0]		—	IWRRD[1:0]		—	IWRRS[1:0]		
	—	TYPE[2:0]			—	BSZ[1:0]		—	
	—	—	—	—	—	—	—	—	
CS2BCR	—	—	IWW[1:0]		—	IWRWD[1:0]		—	
	IWRWS[1:0]		—	IWRRD[1:0]		—	IWRRS[1:0]		
	—	TYPE[2:0]			—	BSZ[1:0]		—	
	—	—	—	—	—	—	—	—	
CS3BCR	—	—	IWW[1:0]		—	IWRWD[1:0]		—	
	IWRWS[1:0]		—	IWRRD[1:0]		—	IWRRS[1:0]		
	—	TYPE[2:0]			—	BSZ[1:0]		—	
	—	—	—	—	—	—	—	—	
CS4BCR	—	—	IWW[1:0]		—	IWRWD[1:0]		—	
	IWRWS[1:0]		—	IWRRD[1:0]		—	IWRRS[1:0]		
	—	TYPE[2:0]			—	BSZ[1:0]		—	
	—	—	—	—	—	—	—	—	

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
CS5BCR	—	—	IWW[1:0]		—	IWRWD[1:0]		—	BSC
	IWRWS[1:0]		—	IWRRD[1:0]		—	IWRRS[1:0]		
	—	TYPE[2:0]			—	BSZ[1:0]		—	
	—	—	—	—	—	—	—	—	
CS6BCR	—	—	IWW[1:0]		—	IWRWD[1:0]		—	
	IWRWS[1:0]		—	IWRRD[1:0]		—	IWRRS[1:0]		
	—	TYPE[2:0]			—	BSZ[1:0]		—	
	—	—	—	—	—	—	—	—	
CS7BCR	—	—	IWW[1:0]		—	IWRWD[1:0]		—	
	IWRWS[1:0]		—	IWRRD[1:0]		—	IWRRS[1:0]		
	—	TYPE[2:0]			—	BSZ[1:0]		—	
	—	—	—	—	—	—	—	—	
CS8BCR	—	—	IWW[1:0]		—	IWRWD[1:0]		—	
	IWRWS[1:0]		—	IWRRD[1:0]		—	IWRRS[1:0]		
	—	TYPE[2:0]			—	BSZ[1:0]		—	
	—	—	—	—	—	—	—	—	
CS0WCR*2	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	WW[2:0]			
	—	—	—	SW[1:0]		WR[3:1]			
	WR[0]	WM	—	—	—	—	HW[1:0]		
CS0WCR*3	—	—	—	—	—	—	—	—	
	—	—	—	BEN	—	BW[1:0]			
	—	—	—	SW[1:0]		W[3:1]			
	W[0]	WM	—	—	—	—	HW[1:0]		
CS0WCR*4	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	BW[1:0]		
	—	—	—	—	—	W[3:1]			
	W[0]	WM	—	—	—	—	—	—	
CS1WCR*2	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	WW[2:0]			
	—	—	—	SW[1:0]		WR[3:1]			
	WR[0]	WM	—	—	—	—	HW[1:0]		
CS2WCR*2	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	WW[2:0]			
	—	—	—	SW[1:0]		WR[3:1]			
	WR[0]	WM	—	—	—	—	HW[1:0]		

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
CS2WCR*5	—	—	—	—	—	—	—	—	BSC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	A2CL[1]	
	A2CL[0]	—	—	—	—	—	—	—	
CS3WCR*2	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	WW[2:0]			
	—	—	—	SW[1:0]		WR[3:1]			
	WR[0]	WM	—	—	—	—	HW[1:0]		
CS3WCR*5	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	WTRP[1:0]		—	WTRCD[1:0]		—	A3CL[1]	
	A3CL[0]	—	—	TRWL[1:0]		—	WTRC[1:0]		
CS4WCR*2	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	WW[2:0]			
	—	—	—	SW[1:0]		WR[3:1]			
	WR[0]	WM	—	—	—	—	HW[1:0]		
CS4WCR*3	—	—	—	—	—	—	—	—	
	—	—	—	BEN	—	BW[1:0]			
	—	—	—	SW[1:0]		W[3:1]			
	W[0]	WM	—	—	—	—	HW[1:0]		
CS5WCR*2	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	WW[2:0]			
	—	—	—	SW[1:0]		WR[3:1]			
	WR[0]	WM	—	—	—	—	HW[1:0]		
CS5WCR*6	—	—	—	—	—	—	—	—	
	—	—	SZSEL	MPXW	—	WW[2:0]			
	—	—	—	SW[1:0]		WR[3:1]			
	WR[0]	WM	—	—	—	—	HW[1:0]		
CS5WCR*7	—	—	—	—	—	—	—	—	
	—	—	SA[1:0]		—	—	—	—	
	—	TED[3:0]			PCW[3:1]				
	PCW[0]	WM	—	—	THE[3:0]				
CS6WCR*2	—	—	—	—	—	—	—	—	
	—	—	—	BAS	—	WW[2:0]			
	—	—	—	SW[1:0]		WR[3:1]			
	WR[0]	WM	—	—	—	—	HW[1:0]		

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
CS6WCR*7	—	—	—	—	—	—	—	—	BSC
	—	—	SA[1:0]		—	—	—	—	
	—	TED[3:0]			PCW[3:1]			—	
	PCW[0]	WM	—	—	THE[3:0]				
CS6WCR*8	—	—	—	—	—	—	—	—	BSC
	—	—	MPXAW[1:0]		MPXMD	—	BW[1:0]		
	—	—	—	—	—	W[3:1]			
	W[0]	WM	—	—	—	—	—	—	
CS7WCR*2	—	—	—	—	—	—	—	—	BSC
	—	—	—	BAS	—	WW[2:0]			
	—	—	—	SW[1:0]		WR[3:1]			
	WR[0]	WM	—	—	—	HW[1:0]			
CS8WCR*2	—	—	—	—	—	—	—	—	BSC
	—	—	—	BAS	—	WW[2:0]			
	—	—	—	SW[1:0]		WR[3:1]			
	WR[0]	WM	—	—	—	HW[1:0]			
SDCR	—	—	—	—	—	—	—	—	BSC
	—	—	—	A2ROW[1:0]		—	A2COL[1:0]		
	—	—	—	—	RFSH	RMODE	—	BACTV	
	—	—	—	A3ROW[1:0]		—	A3COL[1:0]		
RTCSCR	—	—	—	—	—	—	—	—	BSC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	CMF	CMIE	CKS[2:0]			RRC[2:0]			
RTCNT	—	—	—	—	—	—	—	—	BSC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
RTCOR	—	—	—	—	—	—	—	—	BSC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
RAMER	—	—	—	—	—	—	—	—	FLASH
	—	—	—	—	RAMS	RAM[2:0]			

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
BARA	BAA31	BAA30	BAA29	BAA28	BAA27	BAA26	BAA25	BAA24	UBC
	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16	
	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8	
	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0	
BAMRA	BAMA31	BAMA30	BAMA29	BAMA28	BAMA27	BAMA26	BAMA25	BAMA24	
	BAMA23	BAMA22	BAMA21	BAMA20	BAMA19	BAMA18	BAMA17	BAMA16	
	BAMA15	BAMA14	BAMA13	BAMA12	BAMA11	BAMA10	BAMA9	BAMA8	
	BAMA7	BAMA6	BAMA5	BAMA4	BAMA3	BAMA2	BAMA1	BAMA0	
BBRA	—	—	—	—	—	CPA[2:0]			
	CDA[1:0]		IDA[1:0]		RWA[1:0]		SZA[1:0]		
BDRA	BDA31	BDA30	BDA29	BDA28	BDA27	BDA26	BDA25	BDA24	
	BDA23	BDA22	BDA21	BDA20	BDA19	BDA18	BDA17	BDA16	
	BDA15	BDA14	BDA13	BDA12	BDA11	BDA10	BDA9	BDA8	
	BDA7	BDA6	BDA5	BDA4	BDA3	BDA2	BDA1	BDA0	
BDMRA	BDMA31	BDMA30	BDMA29	BDMA28	BDMA27	BDMA26	BDMA25	BDMA24	
	BDMA23	BDMA22	BDMA21	BDMA20	BDMA19	BDMA18	BDMA17	BDMA16	
	BDMA15	BDMA14	BDMA13	BDMA12	BDMA11	BDMA10	BDMA9	BDMA8	
	BDMA7	BDMA6	BDMA5	BDMA4	BDMA3	BDMA2	BDMA1	BDMA0	
BARB	BAB31	BAB30	BAB29	BAB28	BAB27	BAB26	BAB25	BAB24	
	BAB23	BAB22	BAB21	BAB20	BAB19	BAB18	BAB17	BAB16	
	BAB15	BAB14	BAB13	BAB12	BAB11	BAB10	BAB9	BAB8	
	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1	BAB0	
BAMRB	BAMB31	BAMB30	BAMB29	BAMB28	BAMB27	BAMB26	BAMB25	BAMB24	
	BAMB23	BAMB22	BAMB21	BAMB20	BAMB19	BAMB18	BAMB17	BAMB16	
	BAMB15	BAMB14	BAMB13	BAMB12	BAMB11	BAMB10	BAMB9	BAMB8	
	BAMB7	BAMB6	BAMB5	BAMB4	BAMB3	BAMB2	BAMB1	BAMB0	
BBRB	—	—	—	—	—	CPB[2:0]			
	CDB[1:0]		IDB[1:0]		RWB[1:0]		SZB[1:0]		
BDRB	BDB31	BDB30	BDB29	BDB28	BDB27	BDB26	BDB25	BDB24	
	BDB23	BDB22	BDB21	BDB20	BDB19	BDB18	BDB17	BDB16	
	BDB15	BDB14	BDB13	BDB12	BDB11	BDB10	BDB9	BDB8	
	BDB7	BDB6	BDB5	BDB4	BDB3	BDB2	BDB1	BDB0	
BDMRB	BDMB31	BDMB30	BDMB29	BDMB28	BDMB27	BDMB26	BDMB25	BDMB24	
	BDMB23	BDMB22	BDMB21	BDMB20	BDMB19	BDMB18	BDMB17	BDMB16	
	BDMB15	BDMB14	BDMB13	BDMB12	BDMB11	BDMB10	BDMB9	BDMB8	
	BDMB7	BDMB6	BDMB5	BDMB4	BDMB3	BDMB2	BDMB1	BDMB0	

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
BRCR	—	—	—	—	—	—	—	—	UBC
	—	—	UTRGW[1:0]		UBIDB	—	UBIDA	—	
	SCMFCA	SCMFCB	SCMFDA	SCMFDB	PCTE	PCBA	—	—	
	DBEA	PCBB	DBEB	—	SEQ	—	—	ETBE	
BRSR	SVF	—	—	—	BSA27	BSA26	BSA25	BSA24	
	BSA23	BSA22	BSA21	BSA20	BSA19	BSA18	BSA17	BSA16	
	BSA15	BSA14	BSA13	BSA12	BSA11	BSA10	BSA9	BSA8	
	BSA7	BSA6	BSA5	BSA4	BSA3	BSA2	BSA1	BSA0	
BRDR	DVF	—	—	—	BDA27	BDA26	BDA25	BDA24	
	BDA23	BDA22	BDA21	BDA20	BDA19	BDA18	BDA17	BDA16	
	BDA15	BDA14	BDA13	BDA12	BDA11	BDA10	BDA9	BDA8	
	BDA7	BDA6	BDA5	BDA4	BDA3	BDA2	BDA1	BDA0	
BETR	—	—	—	—	BET[11:8]				
	BET[7:0]								

【注】 \*1 寄存器的位的内容因产品而不同。详细内容请参照各寄存器的说明。

\*2 是存储器种类设定为通常空间、带字节选择的 SRAM 的情况。

\*3 是存储器种类设定为突发 ROM（异步）的情况。

\*4 是存储器种类设定为突发 ROM（时钟同步）的情况。

\*5 是存储器种类设定为 SDRAM 的情况。

\*6 是存储器种类设定为 MPX-I/O 的情况。

\*7 是存储器种类设定为 PCMCIA 的情况。

\*8 是存储器种类设定为突发 MPX-I/O 的情况。



## 27.3 各运行模式的寄存器状态

寄存器略称	上电 复位	手动 复位	软件 待机	深度 软件待机	模块 待机	睡眠	模块
SCSMR_0	初始化	保持	初始化	初始化	初始化	保持	SCI (通道 0)
SCBRR_0	初始化	保持	初始化	初始化	初始化	保持	
SCSCR_0	初始化	保持	初始化	初始化	初始化	保持	
SCTDR_0	初始化	保持	初始化	初始化	初始化	保持	
SCSSR_0	初始化	保持	初始化	初始化	初始化	保持	
SCRDR_0	初始化	保持	初始化	初始化	初始化	保持	
SCSDCR_0	初始化	保持	初始化	初始化	初始化	保持	
SCSPTR_0	初始化	保持	初始化	初始化	初始化	保持	
SCSMR_1	初始化	保持	初始化	初始化	初始化	保持	SCI (通道 1)
SCBRR_1	初始化	保持	初始化	初始化	初始化	保持	
SCSCR_1	初始化	保持	初始化	初始化	初始化	保持	
SCTDR_1	初始化	保持	初始化	初始化	初始化	保持	
SCSSR_1	初始化	保持	初始化	初始化	初始化	保持	
SCRDR_1	初始化	保持	初始化	初始化	初始化	保持	
SCSDCR_1	初始化	保持	初始化	初始化	初始化	保持	
SCSPTR_1	初始化	保持	初始化	初始化	初始化	保持	
SCSMR_2	初始化	保持	初始化	初始化	初始化	保持	SCI (通道 2)
SCBRR_2	初始化	保持	初始化	初始化	初始化	保持	
SCSCR_2	初始化	保持	初始化	初始化	初始化	保持	
SCTDR_2	初始化	保持	初始化	初始化	初始化	保持	
SCSSR_2	初始化	保持	初始化	初始化	初始化	保持	
SCRDR_2	初始化	保持	初始化	初始化	初始化	保持	
SCSDCR_2	初始化	保持	初始化	初始化	初始化	保持	
SCSPTR_2	初始化	保持	初始化	初始化	初始化	保持	
SCSMR_3	初始化	保持	初始化	初始化	初始化	保持	SCIF (通道 3)
SCBRR_3	初始化	保持	初始化	初始化	初始化	保持	
SCSCR_3	初始化	保持	初始化	初始化	初始化	保持	
SCFTDR_3	初始化	保持	初始化	初始化	初始化	保持	
SCFSR_3	初始化	保持	初始化	初始化	初始化	保持	
SCFRDR_3	初始化	保持	初始化	初始化	初始化	保持	
SCFCR_3	初始化	保持	初始化	初始化	初始化	保持	
SCFDR_3	初始化	保持	初始化	初始化	初始化	保持	
SCSPTR_3	初始化	保持	初始化	初始化	初始化	保持	
SCLSR_3	初始化	保持	初始化	初始化	初始化	保持	

寄存器略称	上电 复位	手动 复位	软件 待机	深度 软件待机	模块 待机	睡眠	模块
TCR_3	初始化	保持	初始化	初始化	初始化	保持	MTU2
TCR_4	初始化	保持	初始化	初始化	初始化	保持	
TMDR_3	初始化	保持	初始化	初始化	初始化	保持	
TMDR_4	初始化	保持	初始化	初始化	初始化	保持	
TIORH_3	初始化	保持	初始化	初始化	初始化	保持	
TIORL_3	初始化	保持	初始化	初始化	初始化	保持	
TIORH_4	初始化	保持	初始化	初始化	初始化	保持	
TIORL_4	初始化	保持	初始化	初始化	初始化	保持	
TIER_3	初始化	保持	初始化	初始化	初始化	保持	
TIER_4	初始化	保持	初始化	初始化	初始化	保持	
TOER	初始化	保持	初始化	初始化	初始化	保持	
TGCR	初始化	保持	初始化	初始化	初始化	保持	
TOCR1	初始化	保持	初始化	初始化	初始化	保持	
TOCR2	初始化	保持	初始化	初始化	初始化	保持	
TCNT_3	初始化	保持	初始化	初始化	初始化	保持	
TCNT_4	初始化	保持	初始化	初始化	初始化	保持	
TCDR	初始化	保持	初始化	初始化	初始化	保持	
TDDR	初始化	保持	初始化	初始化	初始化	保持	
TGRA_3	初始化	保持	初始化	初始化	初始化	保持	
TGRB_3	初始化	保持	初始化	初始化	初始化	保持	
TGRA_4	初始化	保持	初始化	初始化	初始化	保持	
TGRB_4	初始化	保持	初始化	初始化	初始化	保持	
TCNTS	初始化	保持	初始化	初始化	初始化	保持	
TCBR	初始化	保持	初始化	初始化	初始化	保持	
TGRC_3	初始化	保持	初始化	初始化	初始化	保持	
TGRD_3	初始化	保持	初始化	初始化	初始化	保持	
TGRC_4	初始化	保持	初始化	初始化	初始化	保持	
TGRD_4	初始化	保持	初始化	初始化	初始化	保持	
TSR_3	初始化	保持	初始化	初始化	初始化	保持	
TSR_4	初始化	保持	初始化	初始化	初始化	保持	
TITCR	初始化	保持	初始化	初始化	初始化	保持	
TITCNT	初始化	保持	初始化	初始化	初始化	保持	
TBTER	初始化	保持	初始化	初始化	初始化	保持	
TDER	初始化	保持	初始化	初始化	初始化	保持	
TOLBR	初始化	保持	初始化	初始化	初始化	保持	
TBTM_3	初始化	保持	初始化	初始化	初始化	保持	

寄存器略称	上电 复位	手动 复位	软件 待机	深度 软件待机	模块 待机	睡眠	模块
TBTM_4	初始化	保持	初始化	初始化	初始化	保持	MTU2
TADCR	初始化	保持	初始化	初始化	初始化	保持	
TADCORA_4	初始化	保持	初始化	初始化	初始化	保持	
TADCORB_4	初始化	保持	初始化	初始化	初始化	保持	
TADCOBRA_4	初始化	保持	初始化	初始化	初始化	保持	
TADCOBRB_4	初始化	保持	初始化	初始化	初始化	保持	
TWCR	初始化	保持	初始化	初始化	初始化	保持	
TSTR	初始化	保持	初始化	初始化	初始化	保持	
TSYR	初始化	保持	初始化	初始化	初始化	保持	
TCSYSTR	初始化	保持	初始化	初始化	初始化	保持	
TRWER	初始化	保持	初始化	初始化	初始化	保持	
TCR_0	初始化	保持	初始化	初始化	初始化	保持	
TMDR_0	初始化	保持	初始化	初始化	初始化	保持	
TIORH_0	初始化	保持	初始化	初始化	初始化	保持	
TIORL_0	初始化	保持	初始化	初始化	初始化	保持	
TIER_0	初始化	保持	初始化	初始化	初始化	保持	
TSR_0	初始化	保持	初始化	初始化	初始化	保持	
TCNT_0	初始化	保持	初始化	初始化	初始化	保持	
TGRA_0	初始化	保持	初始化	初始化	初始化	保持	
TGRB_0	初始化	保持	初始化	初始化	初始化	保持	
TGRC_0	初始化	保持	初始化	初始化	初始化	保持	
TGRD_0	初始化	保持	初始化	初始化	初始化	保持	
TGRE_0	初始化	保持	初始化	初始化	初始化	保持	
TGRF_0	初始化	保持	初始化	初始化	初始化	保持	
TIER2_0	初始化	保持	初始化	初始化	初始化	保持	
TSR2_0	初始化	保持	初始化	初始化	初始化	保持	
TBTM_0	初始化	保持	初始化	初始化	初始化	保持	
TCR_1	初始化	保持	初始化	初始化	初始化	保持	
TMDR_1	初始化	保持	初始化	初始化	初始化	保持	
TIOR_1	初始化	保持	初始化	初始化	初始化	保持	
TIER_1	初始化	保持	初始化	初始化	初始化	保持	
TSR_1	初始化	保持	初始化	初始化	初始化	保持	
TCNT_1	初始化	保持	初始化	初始化	初始化	保持	
TGRA_1	初始化	保持	初始化	初始化	初始化	保持	
TGRB_1	初始化	保持	初始化	初始化	初始化	保持	
TICCR	初始化	保持	初始化	初始化	初始化	保持	

寄存器略称	上电 复位	手动 复位	软件 待机	深度 软件待机	模块 待机	睡眠	模块
TCR_2	初始化	保持	初始化	初始化	初始化	保持	MTU2
TMDR_2	初始化	保持	初始化	初始化	初始化	保持	
TIOR_2	初始化	保持	初始化	初始化	初始化	保持	
TIER_2	初始化	保持	初始化	初始化	初始化	保持	
TSR_2	初始化	保持	初始化	初始化	初始化	保持	
TCNT_2	初始化	保持	初始化	初始化	初始化	保持	
TGRA_2	初始化	保持	初始化	初始化	初始化	保持	
TGRB_2	初始化	保持	初始化	初始化	初始化	保持	
TCNTU_5	初始化	保持	初始化	初始化	初始化	保持	
TGRU_5	初始化	保持	初始化	初始化	初始化	保持	
TCRU_5	初始化	保持	初始化	初始化	初始化	保持	
TIORU_5	初始化	保持	初始化	初始化	初始化	保持	
TCNTV_5	初始化	保持	初始化	初始化	初始化	保持	
TGRV_5	初始化	保持	初始化	初始化	初始化	保持	
TCRV_5	初始化	保持	初始化	初始化	初始化	保持	
TIORV_5	初始化	保持	初始化	初始化	初始化	保持	
TCNTW_5	初始化	保持	初始化	初始化	初始化	保持	
TGRW_5	初始化	保持	初始化	初始化	初始化	保持	
TCRW_5	初始化	保持	初始化	初始化	初始化	保持	
TIORW_5	初始化	保持	初始化	初始化	初始化	保持	
TSR_5	初始化	保持	初始化	初始化	初始化	保持	
TIER_5	初始化	保持	初始化	初始化	初始化	保持	
TSTR5	初始化	保持	初始化	初始化	初始化	保持	
TCNTCMPCLR	初始化	保持	初始化	初始化	初始化	保持	
TCR_3S	初始化	保持	初始化	初始化	初始化	保持	MTU2S
TCR_4S	初始化	保持	初始化	初始化	初始化	保持	
TMDR_3S	初始化	保持	初始化	初始化	初始化	保持	
TMDR_4S	初始化	保持	初始化	初始化	初始化	保持	
TIORH_3S	初始化	保持	初始化	初始化	初始化	保持	
TIORL_3S	初始化	保持	初始化	初始化	初始化	保持	
TIORH_4S	初始化	保持	初始化	初始化	初始化	保持	
TIORL_4S	初始化	保持	初始化	初始化	初始化	保持	
TIER_3S	初始化	保持	初始化	初始化	初始化	保持	
TIER_4S	初始化	保持	初始化	初始化	初始化	保持	
TOERS	初始化	保持	初始化	初始化	初始化	保持	
TGCRS	初始化	保持	初始化	初始化	初始化	保持	

寄存器略称	上电 复位	手动 复位	软件 待机	深度 软件待机	模块 待机	睡眠	模块
TOCR1S	初始化	保持	初始化	初始化	初始化	保持	MTU2S
TOCR2S	初始化	保持	初始化	初始化	初始化	保持	
TCNT_3S	初始化	保持	初始化	初始化	初始化	保持	
TCNT_4S	初始化	保持	初始化	初始化	初始化	保持	
TCDRS	初始化	保持	初始化	初始化	初始化	保持	
TDDRS	初始化	保持	初始化	初始化	初始化	保持	
TGRA_3S	初始化	保持	初始化	初始化	初始化	保持	
TGRB_3S	初始化	保持	初始化	初始化	初始化	保持	
TGRA_4S	初始化	保持	初始化	初始化	初始化	保持	
TGRB_4S	初始化	保持	初始化	初始化	初始化	保持	
TCNTSS	初始化	保持	初始化	初始化	初始化	保持	
TCBRS	初始化	保持	初始化	初始化	初始化	保持	
TGRC_3S	初始化	保持	初始化	初始化	初始化	保持	
TGRD_3S	初始化	保持	初始化	初始化	初始化	保持	
TGRC_4S	初始化	保持	初始化	初始化	初始化	保持	
TGRD_4S	初始化	保持	初始化	初始化	初始化	保持	
TSR_3S	初始化	保持	初始化	初始化	初始化	保持	
TSR_4S	初始化	保持	初始化	初始化	初始化	保持	
TITCRS	初始化	保持	初始化	初始化	初始化	保持	
TITCNTS	初始化	保持	初始化	初始化	初始化	保持	
TBTERS	初始化	保持	初始化	初始化	初始化	保持	
TDERS	初始化	保持	初始化	初始化	初始化	保持	
TOLBRS	初始化	保持	初始化	初始化	初始化	保持	
TBTM_3S	初始化	保持	初始化	初始化	初始化	保持	
TBTM_4S	初始化	保持	初始化	初始化	初始化	保持	
TADCRS	初始化	保持	初始化	初始化	初始化	保持	
TADCORA_4S	初始化	保持	初始化	初始化	初始化	保持	
TADCORB_4S	初始化	保持	初始化	初始化	初始化	保持	
TADCOBRA_4S	初始化	保持	初始化	初始化	初始化	保持	
TADCOBRB_4S	初始化	保持	初始化	初始化	初始化	保持	
TSYCRS	初始化	保持	初始化	初始化	初始化	保持	
TWCRS	初始化	保持	初始化	初始化	初始化	保持	
TSTRS	初始化	保持	初始化	初始化	初始化	保持	
TSYRS	初始化	保持	初始化	初始化	初始化	保持	
TRWERS	初始化	保持	初始化	初始化	初始化	保持	
TCNTU_5S	初始化	保持	初始化	初始化	初始化	保持	

寄存器略称	上电 复位	手动 复位	软件 待机	深度 软件待机	模块 待机	睡眠	模块
TGRU_5S	初始化	保持	初始化	初始化	初始化	保持	MTU2S
TCRU_5S	初始化	保持	初始化	初始化	初始化	保持	
TIORU_5S	初始化	保持	初始化	初始化	初始化	保持	
TCNTV_5S	初始化	保持	初始化	初始化	初始化	保持	
TGRV_5S	初始化	保持	初始化	初始化	初始化	保持	
TCRV_5S	初始化	保持	初始化	初始化	初始化	保持	
TIORV_5S	初始化	保持	初始化	初始化	初始化	保持	
TCNTW_5S	初始化	保持	初始化	初始化	初始化	保持	
TGRW_5S	初始化	保持	初始化	初始化	初始化	保持	
TCRW_5S	初始化	保持	初始化	初始化	初始化	保持	
TIORW_5S	初始化	保持	初始化	初始化	初始化	保持	
TSR_5S	初始化	保持	初始化	初始化	初始化	保持	
TIER_5S	初始化	保持	初始化	初始化	初始化	保持	
TSTR_5S	初始化	保持	初始化	初始化	初始化	保持	
TCNTCMPCLRS	初始化	保持	初始化	初始化	初始化	保持	
ADDR0	初始化	保持	初始化	初始化	初始化	保持	A/D (通道 0)
ADDR1	初始化	保持	初始化	初始化	初始化	保持	
ADDR2	初始化	保持	初始化	初始化	初始化	保持	
ADDR3	初始化	保持	初始化	初始化	初始化	保持	
ADCSR_0	初始化	保持	初始化	初始化	初始化	保持	
ADCR_0	初始化	保持	初始化	初始化	初始化	保持	
ADDR4	初始化	保持	初始化	初始化	初始化	保持	A/D (通道 1)
ADDR5	初始化	保持	初始化	初始化	初始化	保持	
ADDR6	初始化	保持	初始化	初始化	初始化	保持	
ADDR7	初始化	保持	初始化	初始化	初始化	保持	
ADCSR_1	初始化	保持	初始化	初始化	初始化	保持	
ADCR_1	初始化	保持	初始化	初始化	初始化	保持	
ADDR8	初始化	保持	初始化	初始化	初始化	保持	A/D (通道 2)
ADDR9	初始化	保持	初始化	初始化	初始化	保持	
ADDR10	初始化	保持	初始化	初始化	初始化	保持	
ADDR11	初始化	保持	初始化	初始化	初始化	保持	
ADDR12	初始化	保持	初始化	初始化	初始化	保持	
ADDR13	初始化	保持	初始化	初始化	初始化	保持	
ADDR14	初始化	保持	初始化	初始化	初始化	保持	
ADDR15	初始化	保持	初始化	初始化	初始化	保持	
ADCSR_2	初始化	保持	初始化	初始化	初始化	保持	

寄存器略称	上电 复位	手动 复位	软件 待机	深度 软件待机	模块 待机	睡眠	模块	
ADCR_2	初始化	保持	初始化	初始化	初始化	保持	A/D (通道 2)	
FCCS	初始化	保持	初始化	初始化	初始化	保持	FLASH	
FPCS	初始化	保持	初始化	初始化	初始化	保持		
FECS	初始化	保持	初始化	初始化	初始化	保持		
FKEY	初始化	保持	初始化	初始化	初始化	保持		
FMATS	初始化	保持	初始化	初始化	初始化	保持		
FTDAR	初始化	保持	初始化	初始化	初始化	保持		
DTCERA	初始化	保持	保持	初始化	保持	保持		DTC
DTCERB	初始化	保持	保持	初始化	保持	保持		
DTCERC	初始化	保持	保持	初始化	保持	保持		
DTCERD	初始化	保持	保持	初始化	保持	保持		
DTCERE	初始化	保持	保持	初始化	保持	保持		
DTCCR	初始化	保持	保持	初始化	保持	保持		
DTCVBR	初始化	保持	保持	初始化	保持	保持		
ICCR1	初始化	保持	保持	初始化	保持	保持	I <sup>2</sup> C2	
ICCR2	初始化	保持	保持	初始化	保持	保持		
ICMR	初始化	保持	保持	初始化	保持	保持		
ICIER	初始化	保持	保持	初始化	保持	保持		
ICSR	初始化	保持	保持	初始化	保持	保持		
SAR	初始化	保持	保持	初始化	保持	保持		
ICDRT	初始化	保持	保持	初始化	保持	保持		
ICDRR	初始化	保持	保持	初始化	保持	保持		
NF2CYC	初始化	保持	保持	初始化	保持	保持		
SSCRH	初始化	保持	初始化	初始化	初始化	保持		SSU
SSCRL	初始化	保持	初始化	初始化	初始化	保持		
SSMR	初始化	保持	初始化	初始化	初始化	保持		
SSER	初始化	保持	初始化	初始化	初始化	保持		
SSSR	初始化	保持	初始化	初始化	初始化	保持		
SSCR2	初始化	保持	初始化	初始化	初始化	保持		
SSTDR0	初始化	保持	初始化	初始化	初始化	保持		
SSTDR1	初始化	保持	初始化	初始化	初始化	保持		
SSTDR2	初始化	保持	初始化	初始化	初始化	保持		
SSTDR3	初始化	保持	初始化	初始化	初始化	保持		
SSRDR0	初始化	保持	初始化	初始化	初始化	保持		
SSRDR1	初始化	保持	初始化	初始化	初始化	保持		

寄存器略称	上电 复位	手动 复位	软件 待机	深度 软件待机	模块 待机	睡眠	模块
SSRDR2	初始化	保持	初始化	初始化	初始化	保持	SSU
SSRDR3	初始化	保持	初始化	初始化	初始化	保持	
CMSTR	初始化	保持	初始化	初始化	初始化	保持	CMT
CMCSR_0	初始化	保持	初始化	初始化	初始化	保持	
CMCNT_0	初始化	保持	初始化	初始化	初始化	保持	
CMCOR_0	初始化	保持	初始化	初始化	初始化	保持	
CMCSR_1	初始化	保持	初始化	初始化	初始化	保持	
CMCNT_1	初始化	保持	初始化	初始化	初始化	保持	
CMCOR_1	初始化	保持	初始化	初始化	初始化	保持	CMT
ICSR1	初始化	保持	保持	初始化	—	保持	POE
OCSR1	初始化	保持	保持	初始化	—	保持	
ICSR2	初始化	保持	保持	初始化	—	保持	
OCSR2	初始化	保持	保持	初始化	—	保持	
ICSR3	初始化	保持	保持	初始化	—	保持	
SPOER	初始化	保持	保持	初始化	—	保持	
POECSR1	初始化	保持	保持	初始化	—	保持	
POECSR2	初始化	保持	保持	初始化	—	保持	
PADRH	初始化	保持	保持	初始化	—	保持	I/O
PADRL	初始化	保持	保持	初始化	—	保持	
PAIORH	初始化	保持	保持	初始化	—	保持	PFC
PAIORL	初始化	保持	保持	初始化	—	保持	
PACRH4	初始化	保持	保持	初始化	—	保持	
PACRH3	初始化	保持	保持	初始化	—	保持	
PACRH2	初始化	保持	保持	初始化	—	保持	
PACRH1	初始化	保持	保持	初始化	—	保持	
PACRL4	初始化	保持	保持	初始化	—	保持	
PACRL3	初始化	保持	保持	初始化	—	保持	
PACRL2	初始化	保持	保持	初始化	—	保持	
PACRL1	初始化	保持	保持	初始化	—	保持	
PAPRH	初始化	保持	保持	初始化	—	保持	
PAPRL	初始化	保持	保持	初始化	—	保持	
PBDRL	初始化	保持	保持	初始化	—	保持	
PBIORL	初始化	保持	保持	初始化	—	保持	PFC
PBCRL3	初始化	保持	保持	初始化	—	保持	
PBCRL2	初始化	保持	保持	初始化	—	保持	
PBCRL1	初始化	保持	保持	初始化	—	保持	



寄存器略称	上电 复位	手动 复位	软件 待机	深度 软件待机	模块 待机	睡眠	模块
PBPRL	初始化	保持	保持	初始化	—	保持	I/O
PCDRH	初始化	保持	保持	初始化	—	保持	
PCDRL	初始化	保持	保持	初始化	—	保持	
PCIORH	初始化	保持	保持	初始化	—	保持	PFC
PCIORL	初始化	保持	保持	初始化	—	保持	
PCCR3H3	初始化	保持	保持	初始化	—	保持	
PCCR3H2	初始化	保持	保持	初始化	—	保持	
PCCR3H1	初始化	保持	保持	初始化	—	保持	
PCCRL4	初始化	保持	保持	初始化	—	保持	
PCCRL3	初始化	保持	保持	初始化	—	保持	
PCCRL2	初始化	保持	保持	初始化	—	保持	
PCCRL1	初始化	保持	保持	初始化	—	保持	
PCPRH	初始化	保持	保持	初始化	—	保持	
PCPRL	初始化	保持	保持	初始化	—	保持	
PDDRH	初始化	保持	保持	初始化	—	保持	
PDDRL	初始化	保持	保持	初始化	—	保持	
PDIORH	初始化	保持	保持	初始化	—	保持	PFC
PDIORL	初始化	保持	保持	初始化	—	保持	
PDCRH4	初始化	保持	保持	初始化	—	保持	
PDCRH3	初始化	保持	保持	初始化	—	保持	
PDCRH2	初始化	保持	保持	初始化	—	保持	
PDCRH1	初始化	保持	保持	初始化	—	保持	
PDCRL4	初始化	保持	保持	初始化	—	保持	
PDCRL3	初始化	保持	保持	初始化	—	保持	
PDCRL2	初始化	保持	保持	初始化	—	保持	
PDCRL1	初始化	保持	保持	初始化	—	保持	
PDPRH	初始化	保持	保持	初始化	—	保持	I/O
PDPRL	初始化	保持	保持	初始化	—	保持	
PEDRH	初始化	保持	保持	初始化	—	保持	
PEDRL	初始化	保持	保持	初始化	—	保持	
PEIORH	初始化	保持	保持	初始化	—	保持	PFC
PEIORL	初始化	保持	保持	初始化	—	保持	
PECRH2	初始化	保持	保持	初始化	—	保持	
PECRH1	初始化	保持	保持	初始化	—	保持	
PECRL4	初始化	保持	保持	初始化	—	保持	
PECRL3	初始化	保持	保持	初始化	—	保持	

寄存器略称	上电 复位	手动 复位	软件 待机	深度 软件待机	模块 待机	睡眠	模块
PECRL2	初始化	保持	保持	初始化	—	保持	PFC
PECRL1	初始化	保持	保持	初始化	—	保持	
PEPRH	初始化	保持	保持	初始化	—	保持	I/O
PEPRL	初始化	保持	保持	初始化	—	保持	
HCPCR	初始化	保持	保持	初始化	—	保持	PFC
IFCR	初始化	保持	保持	初始化	—	保持	
PFDRL	初始化	保持	保持	初始化	—	保持	I/O
FRQCR	初始化 *1	保持	保持	初始化	—	保持	CPG
STBCR1	初始化	保持	保持	初始化	—	保持	低功耗
STBCR2	初始化	保持	保持	初始化	—	保持	
STBCR3	初始化	保持	保持	初始化	—	保持	
STBCR4	初始化	保持	保持	初始化	—	保持	
STBCR5	初始化	保持	保持	初始化	—	保持	
STBCR6	初始化	保持	保持	初始化	—	保持	
WTCNT	初始化 *1	保持	保持	初始化	—	保持	WDT
WTCSR	初始化 *1	保持	保持	初始化	—	保持	
OSCCR	初始化 *2	保持	保持 *3	初始化	—	保持	CPG
RAMCR	初始化	保持	保持	初始化	—	保持	低功耗
ADTSR_0	初始化	保持	保持	初始化	保持	保持	A/D
ADTSR_1	初始化	保持	保持	初始化	保持	保持	
BSCEHR	初始化	保持	保持	初始化	—	保持	BSC
ICR0	初始化	初始化	保持	初始化	—	保持	INTC
IRQCR	初始化	初始化	保持	初始化	—	保持	
IRQSR	初始化	初始化	保持	初始化	—	保持	
IPRA	初始化	初始化	保持	初始化	—	保持	
IPRB	初始化	初始化	保持	初始化	—	保持	
IPRC	初始化	初始化	保持	初始化	—	保持	
IPRD	初始化	初始化	保持	初始化	—	保持	
IPRE	初始化	初始化	保持	初始化	—	保持	
IPRF	初始化	初始化	保持	初始化	—	保持	
IPRH	初始化	初始化	保持	初始化	—	保持	
IPRI	初始化	初始化	保持	初始化	—	保持	
IPRJ	初始化	初始化	保持	初始化	—	保持	
IPRK	初始化	初始化	保持	初始化	—	保持	
IPRL	初始化	初始化	保持	初始化	—	保持	
IPRM	初始化	初始化	保持	初始化	—	保持	

寄存器略称	上电 复位	手动 复位	软件 待机	深度 软件待机	模块 待机	睡眠	模块
SAR_0	初始化	保持	保持	初始化	保持	保持	DMAC
DAR_0	初始化	保持	保持	初始化	保持	保持	
DMATCR_0	初始化	保持	保持	初始化	保持	保持	
CHCR_0	初始化	保持	保持	初始化	保持	保持	
SAR_1	初始化	保持	保持	初始化	保持	保持	
DAR_1	初始化	保持	保持	初始化	保持	保持	
DMATCR_1	初始化	保持	保持	初始化	保持	保持	
CHCR_1	初始化	保持	保持	初始化	保持	保持	
SAR_2	初始化	保持	保持	初始化	保持	保持	
DAR_2	初始化	保持	保持	初始化	保持	保持	
DMATCR_2	初始化	保持	保持	初始化	保持	保持	
CHCR_2	初始化	保持	保持	初始化	保持	保持	
SAR_3	初始化	保持	保持	初始化	保持	保持	
DAR_3	初始化	保持	保持	初始化	保持	保持	
DMATCR_3	初始化	保持	保持	初始化	保持	保持	
CHCR_3	初始化	保持	保持	初始化	保持	保持	
DMAOR	初始化	保持	保持	初始化	保持	保持	
CMNCR	初始化	保持	保持	初始化	—	保持	BSC
CS0BCR	初始化	保持	保持	初始化	—	保持	
CS1BCR	初始化	保持	保持	初始化	—	保持	
CS2BCR	初始化	保持	保持	初始化	—	保持	
CS3BCR	初始化	保持	保持	初始化	—	保持	
CS4BCR	初始化	保持	保持	初始化	—	保持	
CS5BCR	初始化	保持	保持	初始化	—	保持	
CS6BCR	初始化	保持	保持	初始化	—	保持	
CS7BCR	初始化	保持	保持	初始化	—	保持	
CS8BCR	初始化	保持	保持	初始化	—	保持	
CS0WCR	初始化	保持	保持	初始化	—	保持	
CS1WCR	初始化	保持	保持	初始化	—	保持	
CS2WCR	初始化	保持	保持	初始化	—	保持	
CS3WCR	初始化	保持	保持	初始化	—	保持	
CS4WCR	初始化	保持	保持	初始化	—	保持	
CS5WCR	初始化	保持	保持	初始化	—	保持	
CS6WCR	初始化	保持	保持	初始化	—	保持	
CS7WCR	初始化	保持	保持	初始化	—	保持	
CS8WCR	初始化	保持	保持	初始化	—	保持	

寄存器略称	上电 复位	手动 复位	软件 待机	深度 软件待机	模块 待机	睡眠	模块
SDCR	初始化	保持	保持	初始化	—	保持	BSC
RTCSR	初始化	保持	保持	初始化	—	保持	
RTCNT	初始化	保持	保持	初始化	—	保持	
RTCOR	初始化	保持	保持	初始化	—	保持	
RAMER	初始化	初始化	保持	初始化	保持	保持	FLASH
BARA	初始化	保持	保持	初始化	初始化	保持	UBC
BAMRA	初始化	保持	保持	初始化	初始化	保持	
BBRA	初始化	保持	保持	初始化	初始化	保持	
BDRA*4	初始化	保持	保持	初始化	初始化	保持	
BDMRA*4	初始化	保持	保持	初始化	初始化	保持	
BARB	初始化	保持	保持	初始化	初始化	保持	
BAMRB	初始化	保持	保持	初始化	初始化	保持	
BBRB	初始化	保持	保持	初始化	初始化	保持	
BDRB*4	初始化	保持	保持	初始化	初始化	保持	
BDMRB*4	初始化	保持	保持	初始化	初始化	保持	
BRCR	初始化	保持	保持	初始化	初始化	保持	
BRSR*4	初始化	初始化	保持	初始化	初始化	保持	
BRDR*4	初始化	初始化	保持	初始化	初始化	保持	
BETR*4	初始化	保持	保持	初始化	初始化	保持	

【注】 \*1 在通过 WDT 进行上电复位时，不进行初始化。

\*2 在通过 WDT 进行上电复位时，OSCSTOP 位不被初始化。

\*3 OSCSTOP 位被初始化。

\*4 只限 F-ZTAT 版。

## 第 28 章 电特性

### 28.1 绝对最大额定值

绝对最大额定值如表 28.1 所示

表 28.1 绝对最大额定值

项 目		符号	额定值	单位
电源电压		V <sub>CC</sub>	-0.3 ~ +7.0	V
输入电压（模拟输入引脚除外、SCL/SDA 引脚除外）		V <sub>in</sub>	-0.3 ~ V <sub>CC</sub> +0.3	V
输入电压（SCL/SDA 引脚）		V <sub>in</sub>	-0.3 ~ +7.0	V
模拟电源电压		AV <sub>CC</sub>	-0.3 ~ +7.0	V
模拟基准电压		AV <sub>ref</sub>	-0.3 ~ AV <sub>CC</sub> +0.3	V
模拟输入电压		V <sub>an</sub>	-0.3 ~ AV <sub>CC</sub> +0.3	V
工作温度	民用产品	T <sub>opr</sub>	-20 ~ +85	°C
	工业用产品		-40 ~ +85	°C
保存温度		T <sub>stg</sub>	-55 ~ +125	°C

#### 【使用注意事项】

如果使用 LSI 时超过绝对最大额定值，就会产生 LSI 的永久性破坏。

## 28.2 DC 特性

DC 特性如表 28.2 和表 28.3 所示。

表 28.2 DC 特性

条件:  $V_{CC}=3.0V \sim 3.6V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{ref}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$  (民用产品)、 $T_a=-40 \sim +85^{\circ}C$  (工业用产品)

项 目	符 号	Min.	Typ.	Max.	单 位	测 量 条 件
输入高电平的电压 (施密特触发输入 引脚除外)	$\overline{RES}$ 、 $\overline{MRES}$ 、 $\overline{NMI}$ 、 $\overline{FWE}$ 、 $\overline{MD1}$ 、 $\overline{MD0}$ 、 $\overline{ASEMD0}$ 、 $\overline{EXTAL}$	$V_{CC}-0.5$	—	$V_{CC}+0.3$	V	
	模拟兼用端口	2.2	—	$AV_{CC}+0.3$	V	
	其他的输入引脚	2.2	—	$V_{CC}+0.3$	V	
输入低电平的电压 (施密特触发输入 引脚除外)	$\overline{RES}$ 、 $\overline{MRES}$ 、 $\overline{NMI}$ 、 $\overline{FWE}$ 、 $\overline{MD1}$ 、 $\overline{MD0}$ 、 $\overline{ASEMD0}$ 、 $\overline{EXTAL}$	-0.3	—	0.5	V	
	其他的输入引脚	-0.3	—	0.8	V	
施密特触发 输入电压	$\overline{IRQ7} \sim \overline{IRQ0}$ 、 $\overline{POE8} \sim \overline{POE0}$ 、 $\overline{TCLKA} \sim \overline{TCLKD}$ 、 $\overline{TIOC0A} \sim \overline{TIOC0D}$ 、 $\overline{TIOC1A}$ 、 $\overline{TIOC1B}$ 、 $\overline{TIOC2A}$ 、 $\overline{TIOC2B}$ 、 $\overline{TIOC3A} \sim \overline{TIOC3D}$ 、 $\overline{TIOC4A} \sim \overline{TIOC4D}$ 、 $\overline{TIC5U}$ 、 $\overline{TIC5V}$ 、 $\overline{TIC5W}$ 、 $\overline{TIOC3AS} \sim \overline{TIOC3DS}$ 、 $\overline{TIOC4AS} \sim \overline{TIOC4DS}$ 、 $\overline{TIC5US}$ 、 $\overline{TIC5VS}$ 、 $\overline{TIC5WS}$ 、 $\overline{SCK0} \sim \overline{SCK3}$ 、 $\overline{RXD0} \sim \overline{RXD3}$ 、 $\overline{CTS3}$ 、 $\overline{SSCK}$ 、 $\overline{SCS}$ 、 $\overline{SSI}$ 、 $\overline{SSO}$ 、 $\overline{SCL}$ 、 $\overline{SDA}$	$V_{T+}$	$V_{CC}-0.5$	—	V	
		$V_{T-}$	—	0.5	V	
		$V_{T+}-V_{T-}$	0.2	—	—	V
输入漏泄电流	全部的输入引脚 ( $\overline{ASEMD0}$ 除外)	$ I_{in} $	—	1.0	$\mu A$	
输入上拉 MOS 电流	$\overline{ASEMD0}$	$-I_{pu}$	—	350	$\mu A$	$V_{in}=0V$
三态漏泄电流 (off 状态)	端口 A、B、C、D、E	$ I_{tsi} $	—	1.0	$\mu A$	
输出高电平的电压	全部的输出引脚 (SH7084/85/86 的 PB2、 PB3 除外)	$V_{OH}$	$V_{CC}-0.5$	—	V	$I_{OH}=-200\mu A$
		$V_{OH}$	$V_{CC}-1.0$	—	V	$I_{OH}=-1mA$
	PB2、PB3 (仅 SH7084/85/86)	$V_{OH}$	1.0	—	V	$I_{OH}=-200\mu A$
	PD9、PD11 ~ PD15、 PD24 ~ PD29、 PE9、PE11 ~ PE21	$V_{OH}$	$V_{CC}-1.0$	—	V	$I_{OH}=-5mA$

项 目	符 号	Min.	Typ.	Max.	单 位	测 量 条 件	
输出低电平的电压	全部的输出引脚	—	—	0.4	V	I <sub>OL</sub> =1.6mA	
	SCL、SDA	—	—	0.4	V	I <sub>OL</sub> =3mA	
		—	—	0.5	V	I <sub>OL</sub> =8mA	
	PD9、PD11 ~ PD15、 PD24 ~ PD29、 PE9、PE11 ~ PE21	—	—	0.9	V	I <sub>OL</sub> =15mA	
输入电容	全部的输入引脚	C <sub>in</sub>	—	—	20	pF	V <sub>in</sub> =0V、 f=1MHz、 T <sub>a</sub> =25°C
消费电流	通常运行时	I <sub>CC</sub>	—	100 (150)*	135 (165)*	mA	I <sub>φ</sub> =80MHz B <sub>φ</sub> =40MHz P <sub>φ</sub> =40MHz MP <sub>φ</sub> =40MHz MI <sub>φ</sub> =80MHz
	睡眠时		—	65 (140)*	110 (150)*	mA	B <sub>φ</sub> =40MHz P <sub>φ</sub> =40MHz MP <sub>φ</sub> =40MHz MI <sub>φ</sub> =80MHz
	软件待机时		—	10 (20)*	40 (60)*	mA	T <sub>a</sub> ≤ 50°C
			—	—	80 (120)*	mA	50°C < T <sub>a</sub>
	深度软件待机时		—	5 (20)*	30 (50)*	μA	T <sub>a</sub> ≤ 50°C
			—	—	80 (120)*	μA	50°C < T <sub>a</sub>
模拟电源电流 (SH7084 除外)	A/D 转换中	A <sub>Icc</sub>	—	2	3.5	mA	每个模块 A/D 转换的值
	A/D 转换待机时		—	—	1	mA	
	待机时		—	—	10	μA	
参考电源电流 (SH7084 除外)	A/D 转换中	A <sub>Iref</sub>	—	—	2.5	mA	每个模块 A/D 转换的值
	A/D 转换待机时		—	—	2.5	mA	
	待机时		—	—	10	μA	
模拟电源电流 (SH7084)	A/D 转换中	A <sub>Icc</sub>	—	3	6	mA	每个模块 A/D 转换的值
	A/D 转换待机时		—	—	3.5	mA	
	待机时		—	—	10	μA	
RAM 待机电压	VRAM	2.0	—	—	V	V <sub>CC</sub>	

## 【使用注意事项】

- 在不使用 A/D 转换器时，不能将 AV<sub>CC</sub>、AV<sub>SS</sub> 和 AV<sub>REF</sub> 引脚置为开路。
- 消费电流是在 V<sub>IH</sub>(Min.)= V<sub>CC</sub>-0.5V、V<sub>IL</sub>(Max.)= 0.5V 的条件下全部输出引脚为无负载状态时的值。

【注】 \* 只限支持 E10A 全功能的 F-ZTAT 版。

表 28.3 DC 特性

条件:  $V_{CC}=4.5V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{ref}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$  (民用产品)、 $T_a=-40 \sim +85^{\circ}C$  (工业用产品)

项 目	符号	Min.	Typ.	Max.	单位	测量条件	
输入高电平的电压 (施密特触发输入 引脚除外)	$\overline{RES}$ 、 $\overline{MRES}$ 、NMI、 $\overline{FWE}$ 、MD1、MD0、 $\overline{ASEMD0}$ 、EXTAL	$V_{CC}-0.7$	—	$V_{CC}+0.3$	V		
	模拟兼用端口	2.2	—	$AV_{CC}+0.3$	V		
	其他的输入引脚	2.2	—	$V_{CC}+0.3$	V		
输入低电平的电压 (施密特触发输入 引脚除外)	$\overline{RES}$ 、 $\overline{MRES}$ 、NMI、 $\overline{FWE}$ 、MD1、MD0、 $\overline{ASEMD0}$ 、EXTAL	-0.3	—	0.5	V		
	其他的输入引脚	-0.3	—	0.8	V		
施密特触发 输入电压	IRQ7 ~ IRQ0、 POE8 ~ POE0、 TCLKA ~ TCLKD、 TIOC0A ~ TIOC0D、 TIOC1A、TIOC1B、 TIOC2A、TIOC2B、 TIOC3A ~ TIOC3D、 TIOC4A ~ TIOC4D、 TIC5U、TIC5V、TIC5W、 TIOC3AS ~ TIOC3DS、 TIOC4AS ~ TIOC4DS、 TIC5US、TIC5VS、 TIC5WS、 SCK0 ~ SCK3、 RXD0 ~ RXD3、 $\overline{CTS3}$ 、SSCK、 $\overline{SCS}$ 、 SSI、SSO、SCL、SDA	$V_{T+}$	$V_{CC}-0.5$	—	—	V	
		$V_{T-}$	—	—	1.0	V	
		$V_{T+}-V_{T-}$	0.4	—	—	V	
输入漏泄电流	全部的输入引脚 ( $\overline{ASEMD0}$ 除外)	$ I_{in} $	—	—	1.0	$\mu A$	
输入上拉 MOS 电流	$\overline{ASEMD0}$	$-I_{pu}$	—	—	800	$\mu A$ $V_{in}=0V$	
三态漏泄电流 (off 状态)	端口 A 的、B、C、D、E	$ I_{tsi} $	—	—	1.0	$\mu A$	
输出高电平的电压	全部的输出引脚 (SH7084/85/86 的 PB2、 PB3 除外)	$V_{OH}$	$V_{CC}-0.5$	—	—	V	$I_{OH}=-200\mu A$
			$V_{CC}-1.0$	—	—	V	$I_{OH}=-1mA$
	PB2、PB3 (仅 SH7084/85/86)		1.0	—	—	V	$I_{OH}=-200\mu A$
	PD9、PD11 ~ PD15、 PD24 ~ PD29、 PE9、PE11 ~ PE21		$V_{CC}-1.0$	—	—	V	$I_{OH}=-5mA$



项 目	符 号	Min.	Typ.	Max.	单 位	测 量 条 件	
输出低电平电压	全部的输出引脚	—	—	0.4	V	I <sub>OL</sub> =1.6mA	
	SCL、SDA	—	—	0.4	V	I <sub>OL</sub> =3mA	
		—	—	0.5	V	I <sub>OL</sub> =8mA	
	PD9、PD11 ~ PD15、 PD24 ~ PD29、 PE9、PE11 ~ PE21	—	—	1.4	V	I <sub>OL</sub> =15mA	
输入电容	全部的输入引脚	C <sub>in</sub>	—	—	20	pF	V <sub>in</sub> =0V、 f=1MHz、 T <sub>a</sub> =25°C
消费电流	通常运行时	I <sub>CC</sub>	—	100 (150)*	135 (165)*	mA	I <sub>φ</sub> =80MHz B <sub>φ</sub> =40MHz P <sub>φ</sub> =40MHz MP <sub>φ</sub> =40MHz MI <sub>φ</sub> =80MHz
	睡眠时		—	65 (150)*	110 (150)*	mA	B <sub>φ</sub> =40MHz P <sub>φ</sub> =40MHz MP <sub>φ</sub> =40MHz MI <sub>φ</sub> =80MHz
	软件待机时		—	10 (20)*	40 (60)*	mA	T <sub>a</sub> ≤ 50°C
			—	—	80 (120)*	mA	50°C < T <sub>a</sub>
	深度软件待机时		—	5 (20)*	30 (50)*	μA	T <sub>a</sub> ≤ 50°C
			—	—	80 (120)*	μA	50°C < T <sub>a</sub>
模拟电源电流 (SH7084 除外)	A/D 转换中	A <sub>Icc</sub>	—	2	3.5	mA	每个模块 A/D 转换的值
	A/D 转换待机时		—	—	1	mA	
	待机时		—	—	10	μA	
参考电源电流 (SH7084 除外)	A/D 转换中	A <sub>Iref</sub>	—	—	2.5	mA	每个模块 A/D 转换的值
	A/D 转换待机时		—	—	2.5	mA	
	待机时		—	—	10	μA	
模拟电源电流 (SH7084)	A/D 转换中	A <sub>Icc</sub>	—	3	6	mA	每个模块 A/D 转换的值
	A/D 转换待机时		—	—	3.5	mA	
	待机时		—	—	10	μA	
RAM 待机电压	VRAM	2.0	—	—	V	V <sub>CC</sub>	

## 【使用注意事项】

- 在不使用 A/D 转换器时，不能将 AV<sub>CC</sub>、AV<sub>SS</sub> 和 AV<sub>ref</sub> 引脚置为开路。
- 消费电流是在 V<sub>IH</sub>(Min.)= V<sub>CC</sub>-0.5V、V<sub>IL</sub>(Max.)= 0.5V 的条件下全部输出引脚为无负载状态时的值。

【注】 \* 只限支持 E10A 全功能的 F-ZTAT 版。

表 28.4 输出允许电流值

条件:  $V_{CC}=3.0V \sim 3.6V$  或者  $4.5V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{ref}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$  (民用产品)、 $T_a=-40 \sim +85^{\circ}C$  (工业用产品)

项 目	符号	Min.	Typ.	Max.	单位
输出低电平的容许电流 (每个引脚)	I <sub>OL</sub>	—	—	2.0*	mA
输出低电平的容许电流 (总和)	ΣI <sub>OL</sub>	—	—	80	mA
输出高电平的容许电流 (每个引脚)	-I <sub>OH</sub>	—	—	2.0*	mA
输出高电平的容许电流 (总和)	Σ-I <sub>OH</sub>	—	—	25	mA

【使用注意事项】为了确保 LSI 的可靠性，输出的电流值不能超过表 28.4 中的值。

【注】 \* PD9、PD11 ~ PD15、PD24 ~ PD29、PE9、PE11 ~ PE21 为 I<sub>OL</sub>=15mA(Max.) / -I<sub>OH</sub>=5mA(Max.)。SCL、SDA 为 I<sub>OL</sub>=8mA(Max.)。但是，这些引脚中同时超过 2.0mA 流过 I<sub>OL</sub> / -I<sub>OH</sub> 的引脚只能在 3 个以内。

## 28.3 AC 特性

原则上输入到本 LSI 的信号是时钟同步输入信号。只要没有特殊理由，必须保持各输入信号的准备和保持时间。

表 28.5 最大工作频率

条件:  $V_{CC}=3.0V \sim 3.6V$  或者  $4.5V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{ref}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$  (民用产品)、 $T_a=-40 \sim +85^{\circ}C$  (工业用产品)

项 目		符号	Min.	Typ.	Max.	单位	备注
工作频率	CPU (I $\phi$ )	f	10	—	80	MHz	
	外部总线 (B $\phi$ )		10	—	40		
	外围模块 (P $\phi$ )		10	—	40		
	MTU2 (MP $\phi$ )		10	—	40		
	MTU2S (MI $\phi$ )		10	—	80		

## 28.3.1 时钟时序

表 28.6 时钟时序

条件:  $V_{CC}=3.0V \sim 3.6V$  或者  $4.5V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{ref}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$  (民用产品)、 $T_a=-40 \sim +85^{\circ}C$  (工业用产品)

项 目	符号	Min.	Max.	单位	参照图
EXTAL 时钟的输入频率	$f_{EX}$	5	12.5	MHz	图 28.1
EXTAL 时钟的输入周期时间	$t_{EXcyc}$	80	200	ns	
EXTAL 时钟的输入低电平脉宽	$t_{EXL}$	20	—	ns	
EXTAL 时钟的输入高电平脉宽	$t_{EXH}$	20	—	ns	
EXTAL 时钟的输入上升时间	$t_{EXr}$	—	5	ns	
EXTAL 时钟的输入下降时间	$t_{EXf}$	—	5	ns	
CK 时钟输出频率	$f_{oP}$	10	40	MHz	图 28.2
CK 时钟的输出周期时间	$t_{cyc}$	25	100	ns	
CK 时钟的输出低电平脉宽	$t_{CKL}$	$1/2 t_{cyc}-7.5$	—	ns	
CK 时钟的输出高电平脉宽	$t_{CKH}$	$1/2 t_{cyc}-7.5$	—	ns	
CK 时钟的输出上升时间	$t_{CKr}$	—	5	ns	
CK 时钟的输出下降时间	$t_{CKf}$	—	5	ns	
上电振荡稳定时间	$t_{OSC1}$	10	—	ms	图 28.3
待机返回的振荡稳定时间 1	$t_{OSC2}$	10	—	ms	图 28.4
待机返回的振荡稳定时间 2	$t_{OSC3}$	10	—	ms	图 28.5

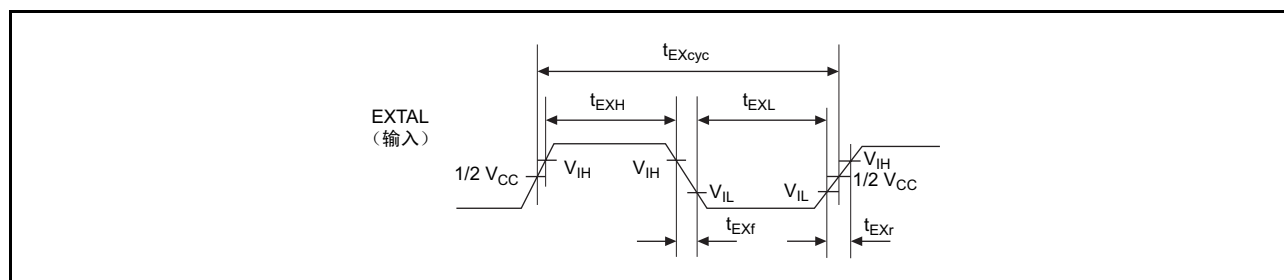


图 28.1 EXTAL 时钟的输入时序

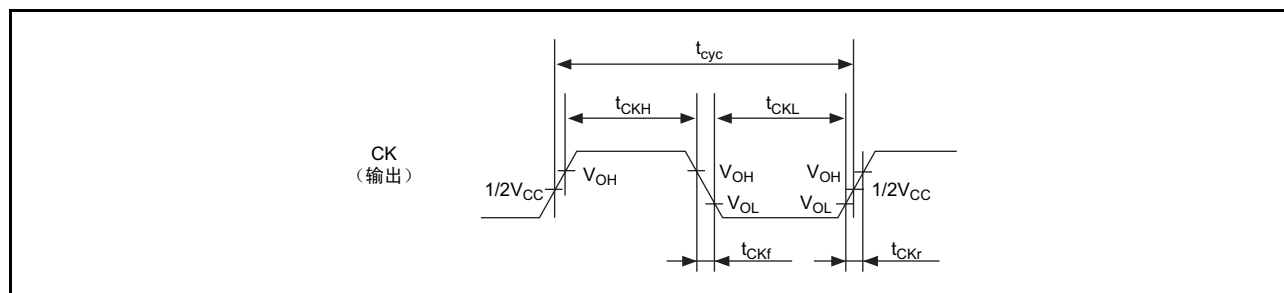


图 28.2 CK 时钟的输出时序

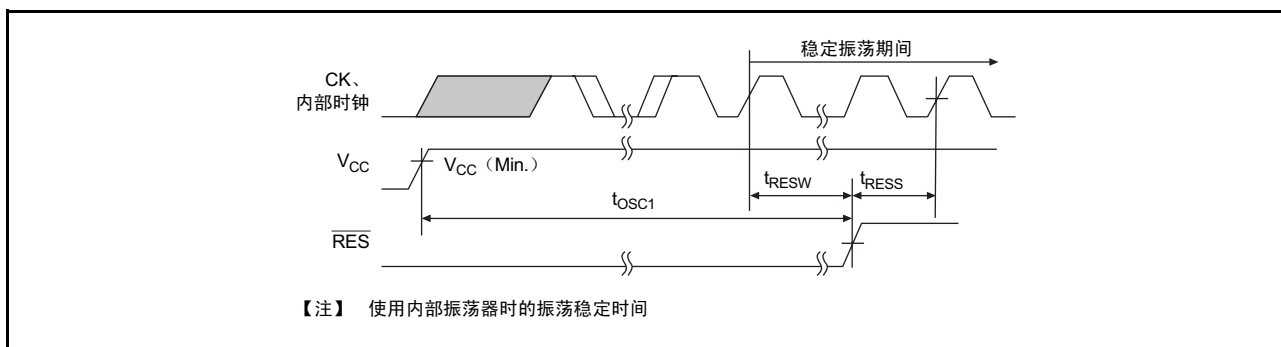


图 28.3 上电振荡的稳定时间

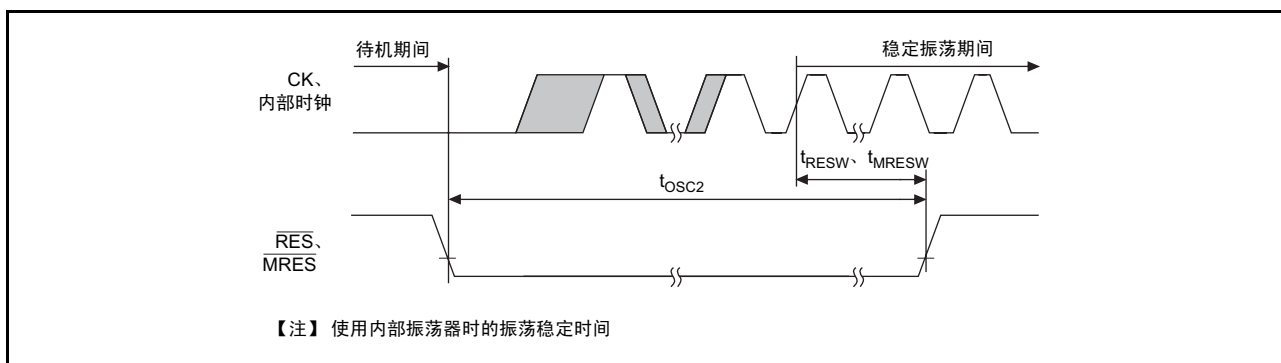


图 28.4 待机返回时的振荡稳定时间（通过复位的返回）

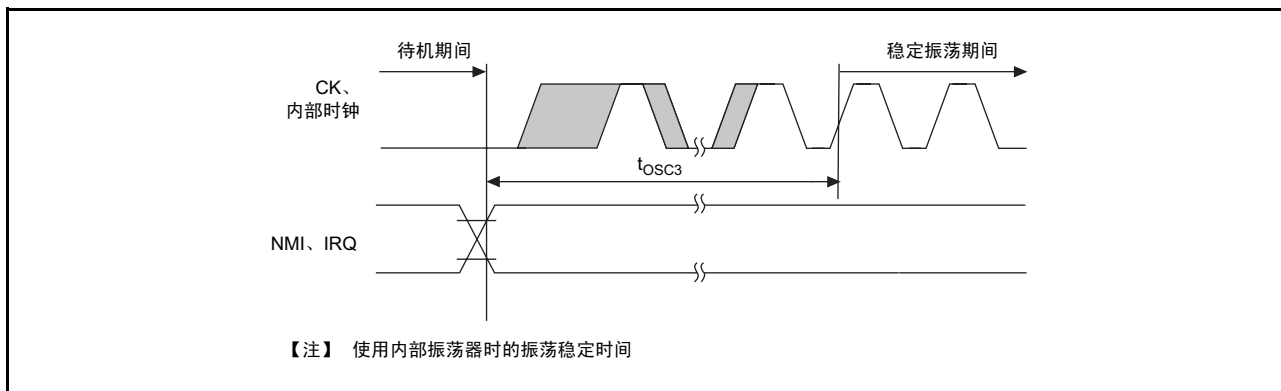


图 28.5 待机返回时的振荡稳定时间（通过 NMI、IRQ 的返回）

## 28.3.2 控制信号的时序

表 28.7 控制信号的时序

条件:  $V_{CC}=3.0V \sim 3.6V$  或者  $4.5V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{ref}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$  (民用产品)、 $T_a=-40 \sim +85^{\circ}C$  (工业用产品)

项 目	符号	Min.	Max.	单位	参照图
$\overline{RES}$ 脉宽	$t_{RESW}$	20*2	—	$t_{Bcyc}$ *4	图 28.3
$\overline{RES}$ 准备时间 *1	$t_{RESS}$	65	—	ns	图 28.4
$\overline{RES}$ 保持时间	$t_{RESH}$	15	—	ns	图 28.6
$\overline{MRES}$ 脉宽	$t_{MRESW}$	20*3	—	$t_{Bcyc}$ *4	图 28.7
$\overline{MRES}$ 准备时间 *1	$t_{MRESS}$	25	—	ns	
$\overline{MRES}$ 保持时间	$t_{MRESH}$	15	—	ns	
MD1、MD0、FWE 准备时间	$t_{MDS}$	20	—	$t_{Bcyc}$ *4	图 28.6
$\overline{BREQ}$ 准备时间	$t_{BREQS}$	$1/2t_{Bcyc}+15$	—	ns	图 28.9
$\overline{BREQ}$ 保持时间	$t_{BREQH}$	$1/2t_{Bcyc}+10$	—	ns	
NMI 准备时间 *1	$t_{NMIS}$	60	—	ns	图 28.7
NMI 保持时间	$t_{NMIH}$	10	—	ns	
IRQ7 ~ IRQ0 准备时间 *1	$t_{IRQS}$	35	—	ns	
IRQ7 ~ IRQ0 保持时间	$t_{IRQH}$	35	—	ns	
$\overline{IRQOUT}$ 输出延迟时间	$t_{IRQOD}$	—	100	ns	图 28.8
$\overline{BACK}$ 延迟时间	$t_{BACKD}$	—	$1/2t_{Bcyc}+20$	ns	图 28.9
总线三态延迟时间	$t_{BOFF}$	0	100	ns	图 28.10
总线缓冲器 ON 时间	$t_{BON}$	0	100	ns	

【注】 \*1  $\overline{RES}$ 、 $\overline{MRES}$ 、NMI、 $\overline{BREQ}$  和 IRQ7 ~ IRQ0 是异步信号。如果满足在此所示的准备时间，就在时钟上升沿检测到信号的变化；否则，信号变化的检测就会推迟到下一个时钟的上升沿。

\*2 在待机模式时，为  $t_{RESW}=t_{OSC2}$  (10ms)。

\*3 在待机模式时，为  $t_{MRESW}=t_{OSC2}$  (10ms)。

\*4  $t_{Bcyc}$  表示外部总线时钟 ( $B\phi=CK$ ) 的周期。

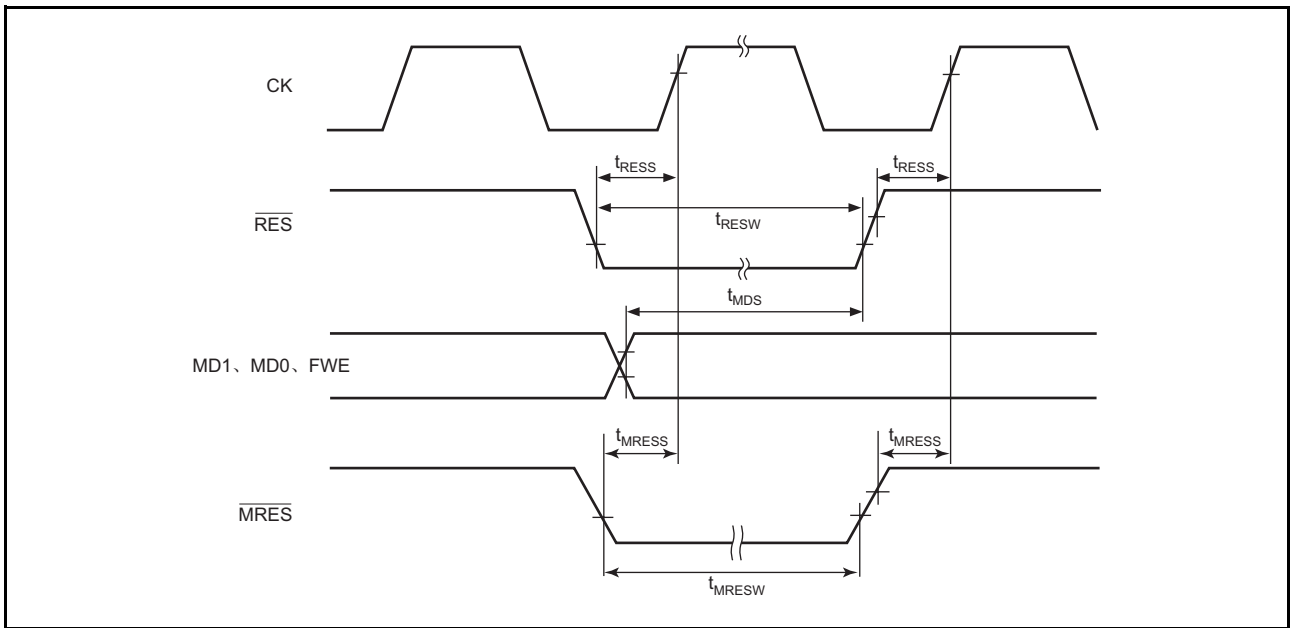


图 28.6 复位的输入时序

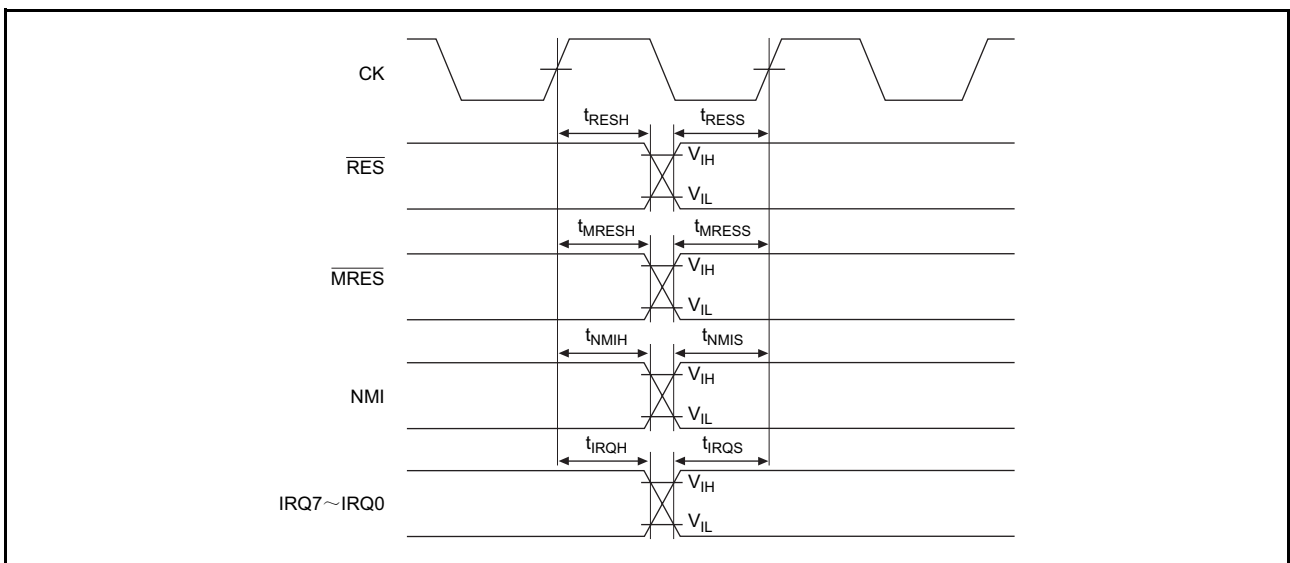


图 28.7 中断信号的输入时序

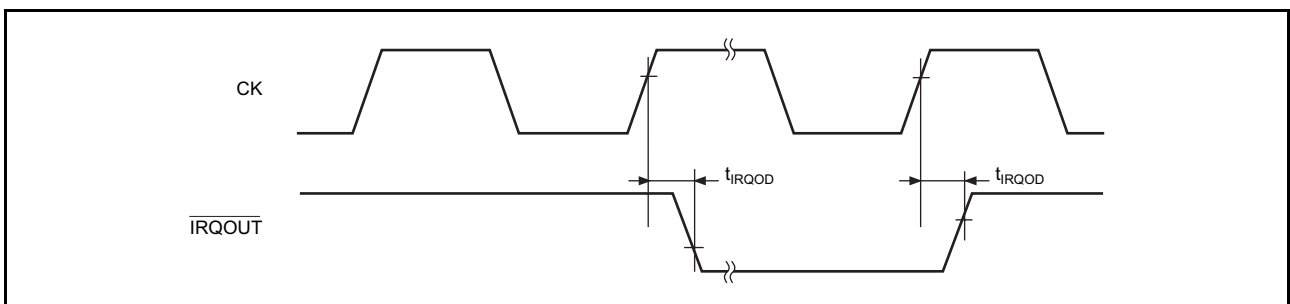


图 28.8 中断信号的输出时序

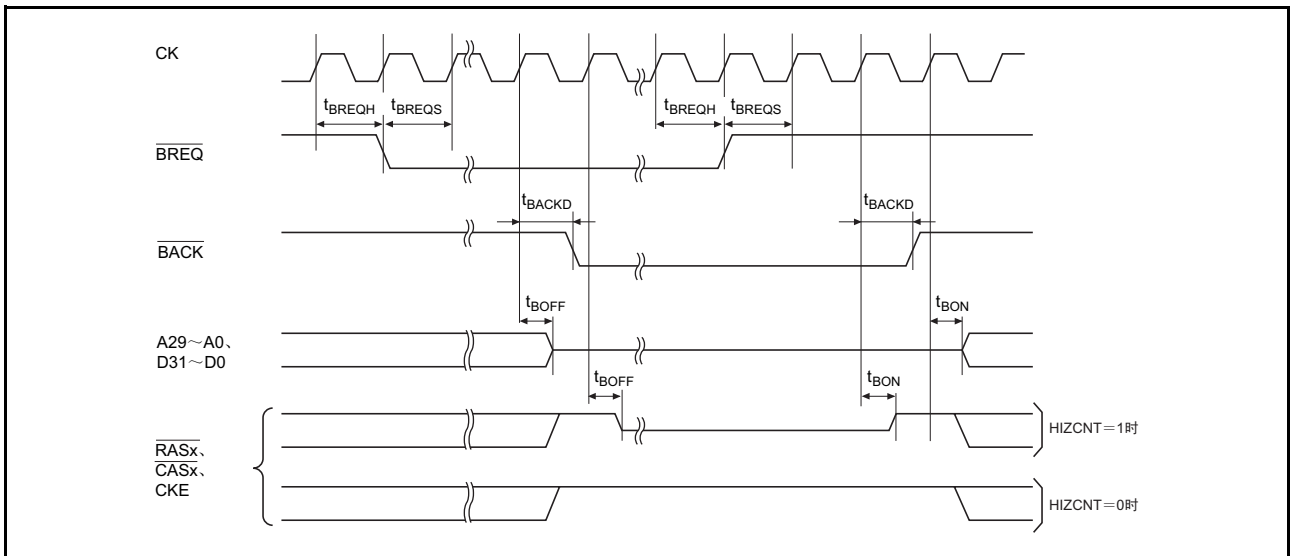


图 28.9 总线权的释放时序

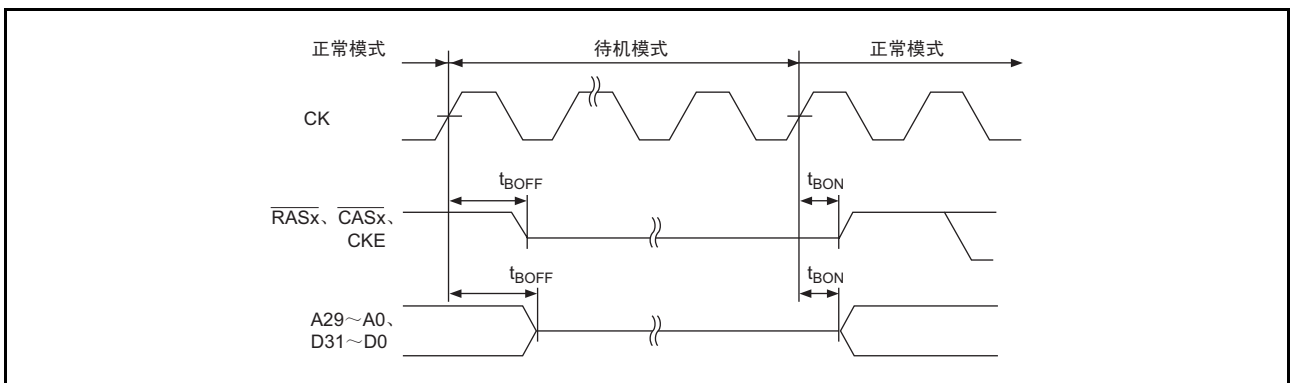


图 28.10 待机时的引脚驱动时序

## 28.3.3 AC 总线的时序规格

表 28.8 总线时序

条件:  $V_{CC}=3.0V \sim 3.6V$  或者  $4.5V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{ref}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$  (民用产品)、 $T_a=-40 \sim +85^{\circ}C$  (工业用产品)

项 目	符号	Min.	Max.	单位	参照图
地址延迟时间 1	$t_{AD1}$	1	18	ns	图 28.11 ~ 28.44
地址延迟时间 2	$t_{AD2}$	$1/2t_{Bcyc}+1$	$1/2t_{Bcyc}+18$	ns	图 28.23
地址准备时间	$t_{AS}$	0	—	ns	图 28.11 ~ 28.14、28.18
地址保持时间	$t_{AH}$	0	—	ns	图 28.11 ~ 28.14、28.18
$\overline{BS}$ 延迟时间	$t_{BSD}$	—	18	ns	图 28.11 ~ 28.37、28.41 ~ 28.44
$\overline{CS}$ 延迟时间	$t_{CSD}$	1	18	ns	图 28.11 ~ 28.44
$\overline{CS}$ 准备时间	$t_{CSS}$	0	—	ns	图 28.11 ~ 28.14
$\overline{CS}$ 保持时间	$t_{CSH}$	0	—	ns	图 28.11 ~ 28.14
读写延迟时间	$t_{RWD}$	1	18	ns	图 28.11 ~ 28.44
读选通的延迟时间	$t_{RSD}$	$1/2t_{Bcyc}+1$	$1/2t_{Bcyc}+18$	ns	图 28.11 ~ 28.18、28.23、28.41、28.42
读数据的准备时间 1	$t_{RDS1}$	$1/2t_{Bcyc}+18$	—	ns	图 28.11 ~ 28.18、28.41 ~ 28.44
读数据的准备时间 2	$t_{RDS2}$	19	—	ns	图 28.20 ~ 28.22、28.24 ~ 28.27、28.32 ~ 28.34
读数据的准备时间 3	$t_{RDS3}$	$1/2t_{Bcyc}+18$	—	ns	图 28.32
读数据的保持时间 1	$t_{RDH1}$	0	—	ns	图 28.11 ~ 28.18、28.41 ~ 28.44
读数据的保持时间 2	$t_{RDH2}$	2	—	ns	图 28.20 ~ 28.22、28.24 ~ 28.27、28.32 ~ 28.34
读数据的保持时间 3	$t_{RDH3}$	0	—	ns	图 28.32
读数据的存取时间	$t_{ACC}^*2$	$t_{Bcyc} \times (n+1.5) - 33^*1$	—	ns	图 28.11 ~ 28.17
读选通开始的存取时间	$t_{OE}^*2$	$t_{Bcyc} \times (n+1) - 31^*1$	—	ns	图 28.11 ~ 28.17
写选通的延迟时间 1	$t_{WSD1}$	$1/2t_{Bcyc}+1$	$1/2t_{Bcyc}+18$	ns	图 28.11 ~ 28.16、28.41、28.42
写选通的延迟时间 2	$t_{WSD2}$	—	20	ns	图 28.17
写数据的延迟时间 1	$t_{WDD1}$	—	20	ns	图 28.11 ~ 28.22、28.41 ~ 28.44
写数据的延迟时间 2	$t_{WDD2}$	—	20	ns	图 28.28 ~ 28.31、28.35 ~ 28.37
写数据的保持时间 1	$t_{WDH1}$	1	11	ns	图 28.11 ~ 28.22、28.41 ~ 28.44
写数据的保持时间 2	$t_{WDH2}$	1	—	ns	图 28.28 ~ 28.31、28.35 ~ 28.37
写数据的保持时间	$t_{WRH}$	0	—	ns	图 28.11 ~ 28.14、28.18
$\overline{WAIT}$ 准备时间	$t_{WTS}$	$1/2t_{Bcyc}+17$	—	ns	图 28.12 ~ 28.23、28.42、28.44
$\overline{WAIT}$ 保持时间	$t_{WTH}$	$1/2t_{Bcyc}+7$	—	ns	图 28.12 ~ 28.23、28.42、28.44
$\overline{RAS}$ 延迟时间	$t_{RASD}$	1	18	ns	图 28.24 ~ 28.35、28.37 ~ 28.40
$\overline{CAS}$ 延迟时间	$t_{CASD}$	1	18	ns	图 28.24 ~ 28.40
DQM 延迟时间	$t_{DQMD}$	1	18	ns	图 28.24 ~ 28.37
CKE 延迟时间	$t_{CKED}$	1	18	ns	图 28.39



项 目	符号	Min.	Max.	单位	参照图
AH 延迟时间	$t_{AHD}$	$1/2t_{Bcyc}+1$	$1/2t_{Bcyc}+18$	ns	图 28.18
多路地址延迟时间	$t_{MAD}$	—	18	ns	图 28.18
多路地址保持时间	$t_{MAH}$	1	—	ns	图 28.18
DACK、TEND 延迟时间	$t_{DACD}$	1	18	ns	图 28.11 ~ 28.35
FRAME 延迟时间	$t_{FMD}$	1	18	ns	图 28.19 ~ 28.22
ICIORD 延迟时间	$t_{ICRSD}$	$1/2t_{Bcyc}+1$	$1/2t_{Bcyc}+18$	ns	图 28.43、28.44
ICIOWR 延迟时间	$t_{ICWSD}$	$1/2t_{Bcyc}+1$	$1/2t_{Bcyc}+18$	ns	图 28.43、28.44
IOIS16 准备时间	$t_{IO16S}$	$1/2t_{Bcyc}+13$	—	ns	图 28.44
IOIS16 保持时间	$t_{IO16H}$	$1/2t_{Bcyc}+10$	—	ns	图 28.44

【注】  $t_{Bcyc}$  表示外部总线时钟 ( $B\phi=CK$ ) 的周期。

\*1 n 为等待数

\*2 如果满足存取时间，就不需要满足  $t_{RDS1}$ 。

\*3 支持 E10A 全功能的 F-ZTAT 版的运行保证温度范围是 0 ~ +50°C。

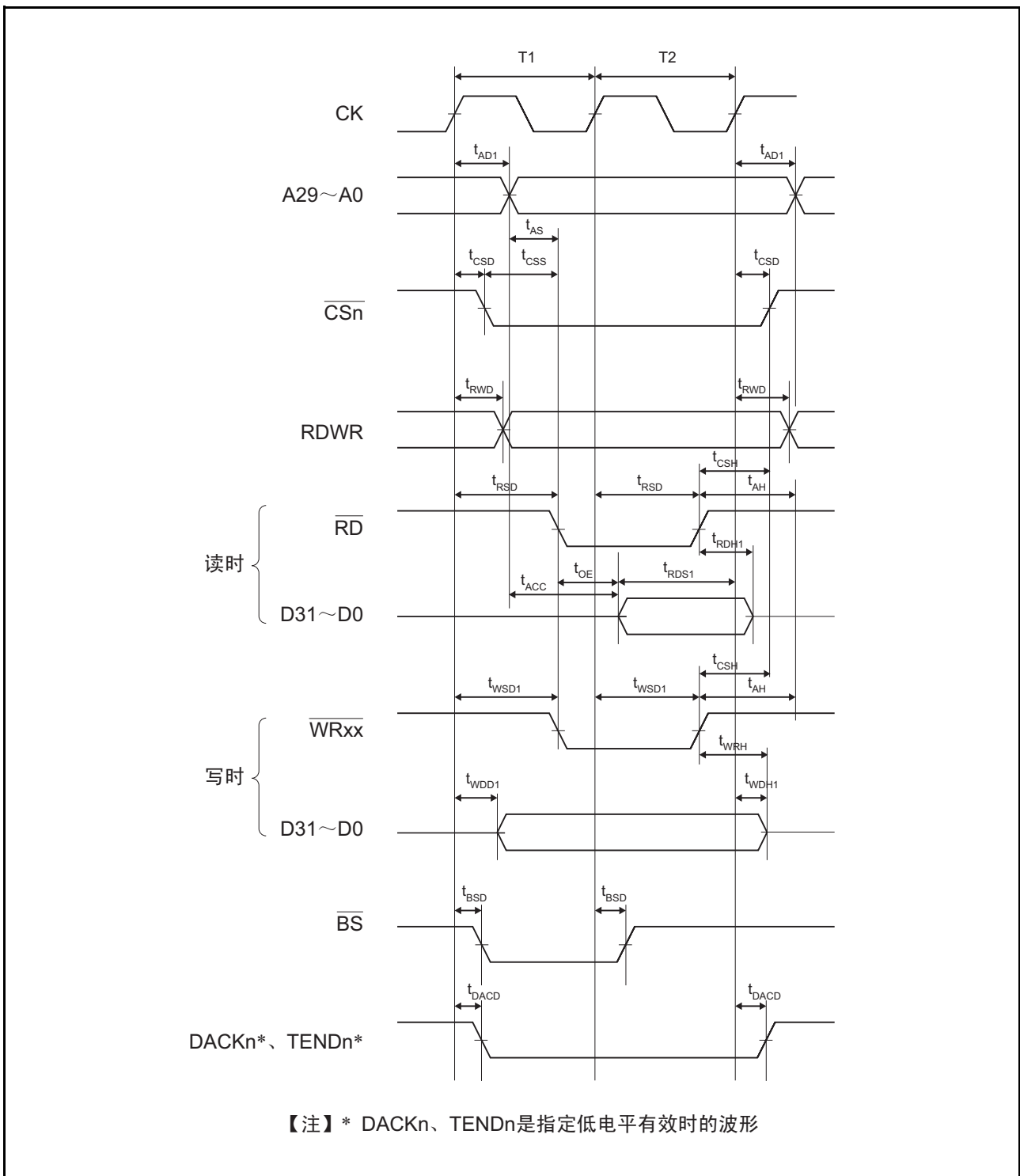


图 28.11 通常空间的基本总线周期（无等待）

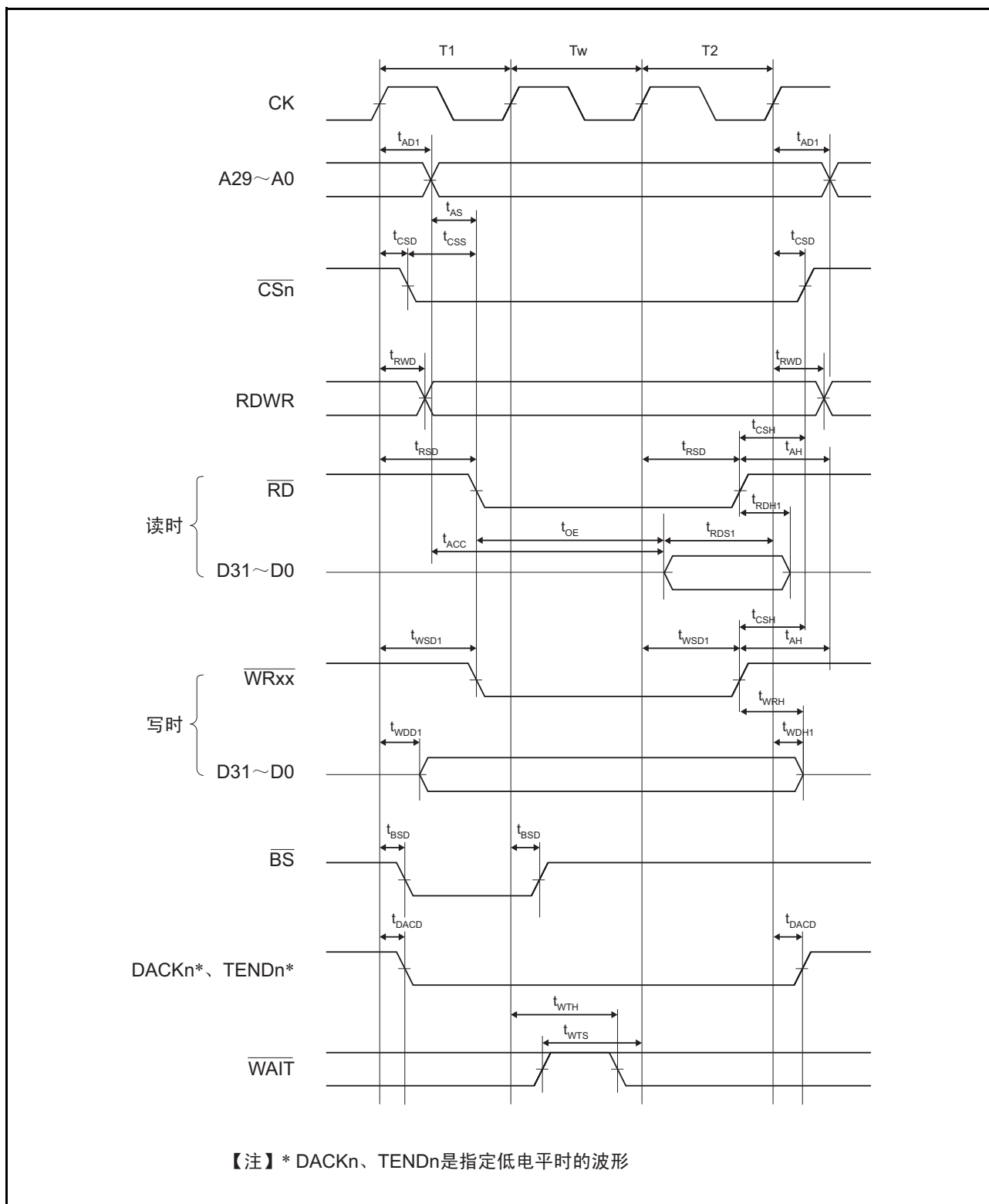


图 28.12 通常空间的基本总线周期（1 个软件等待）

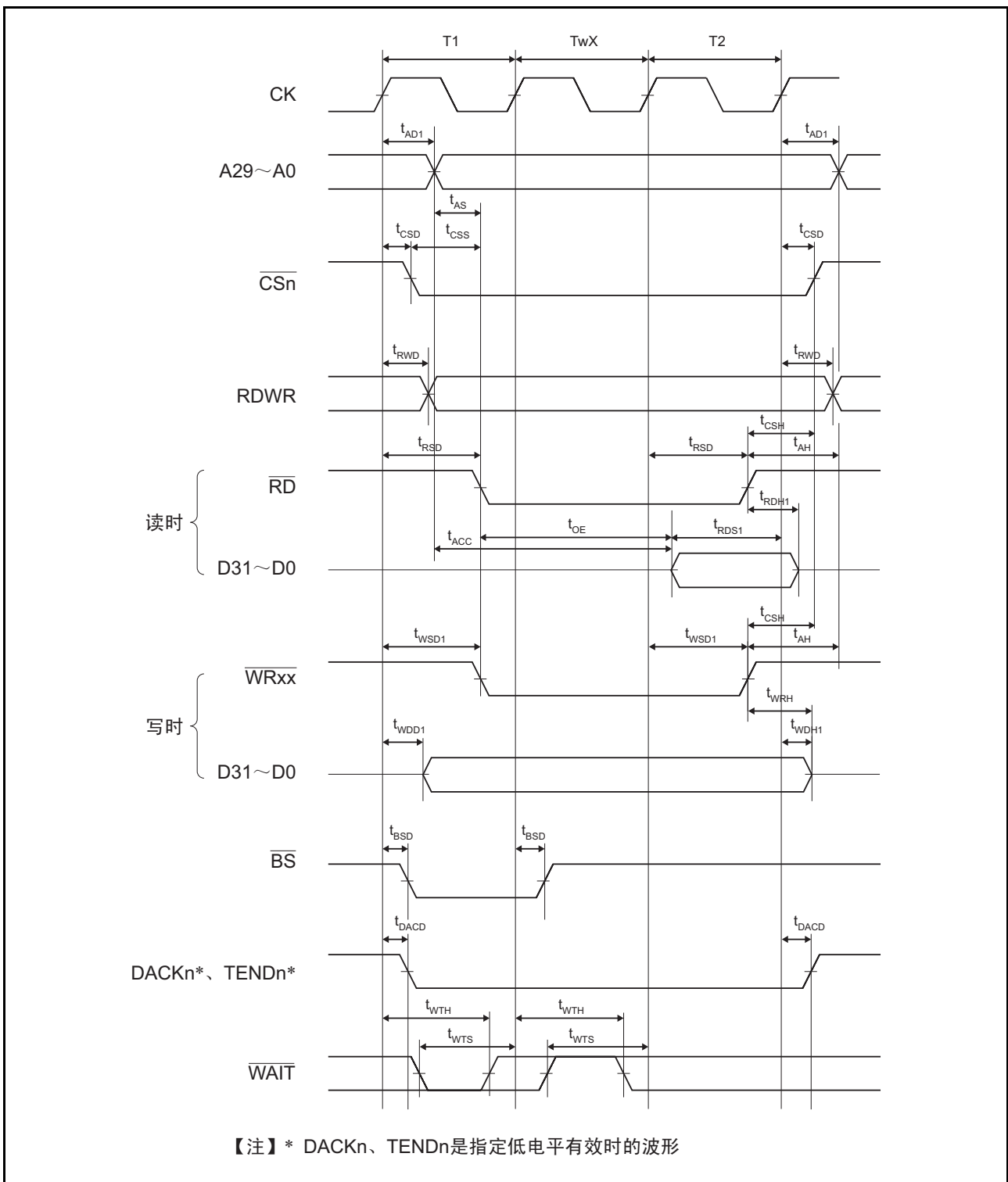


图 28.13 通常空间的基本总线周期（插入 1 个外部等待）

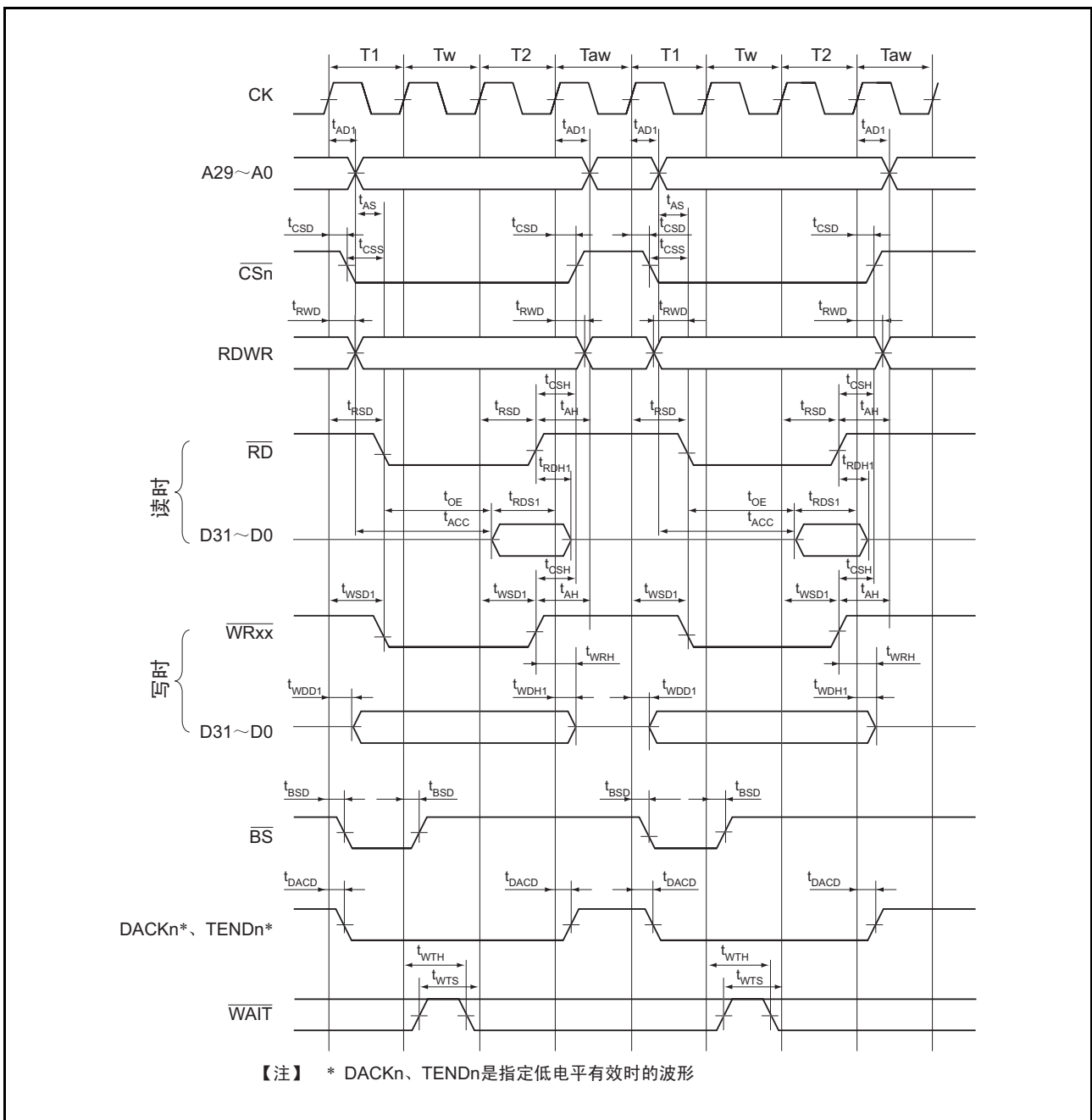


图 28.14 通常空间的基本总线周期  
(1个软件等待、外部等待有效 (WM位=0)、无空闲周期)

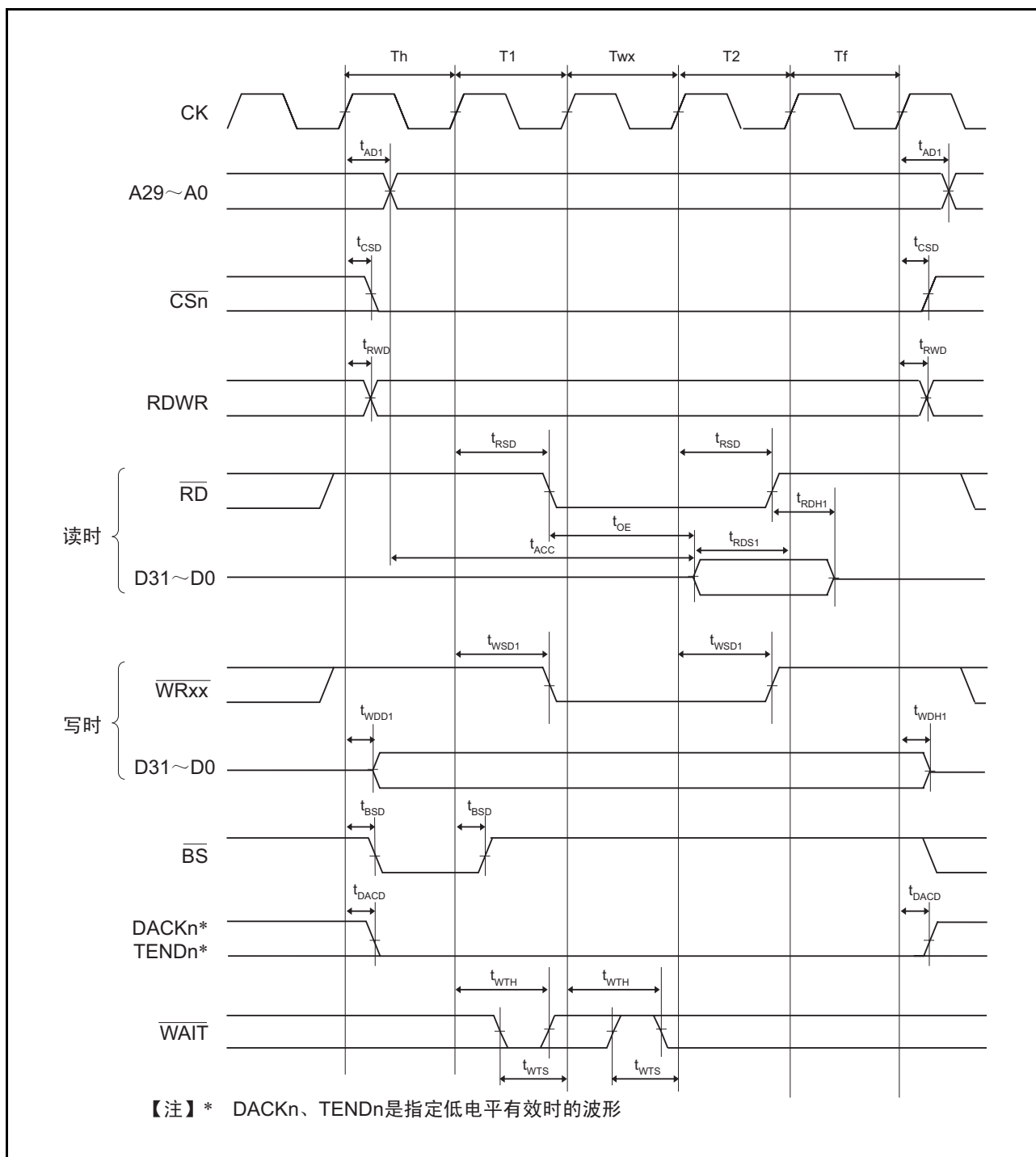


图 28.15 通常空间的 CS 扩展总线周期  
(SW=1 个周期、HW=1 个周期、插入 1 个外部等待)

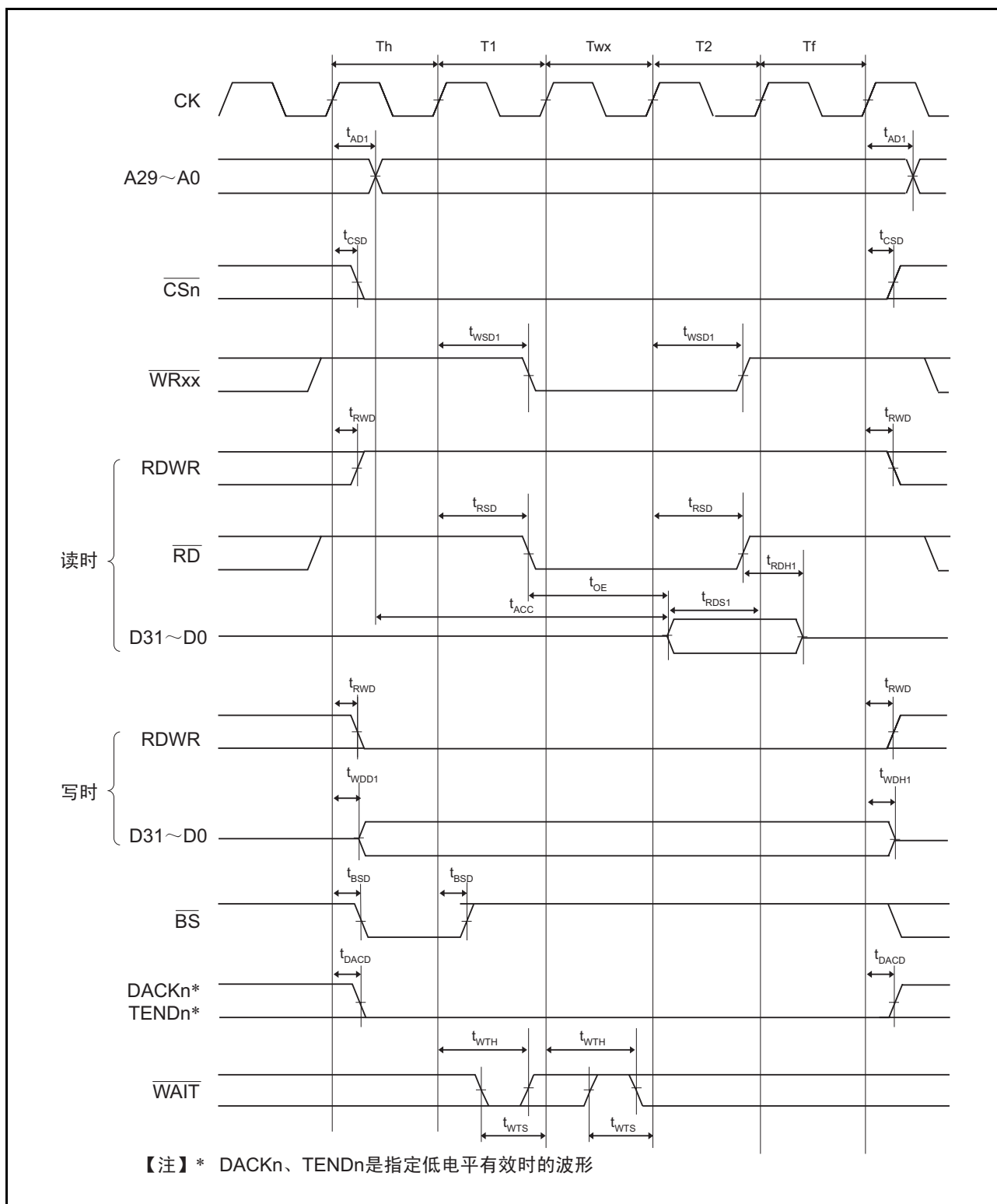


图 28.16 带字节选择的 SRAM 总线周期 (SW=1 个周期、HW=1 个周期、插入 1 个外部等待、BAS=0 (写周期 UB/LB 控制))

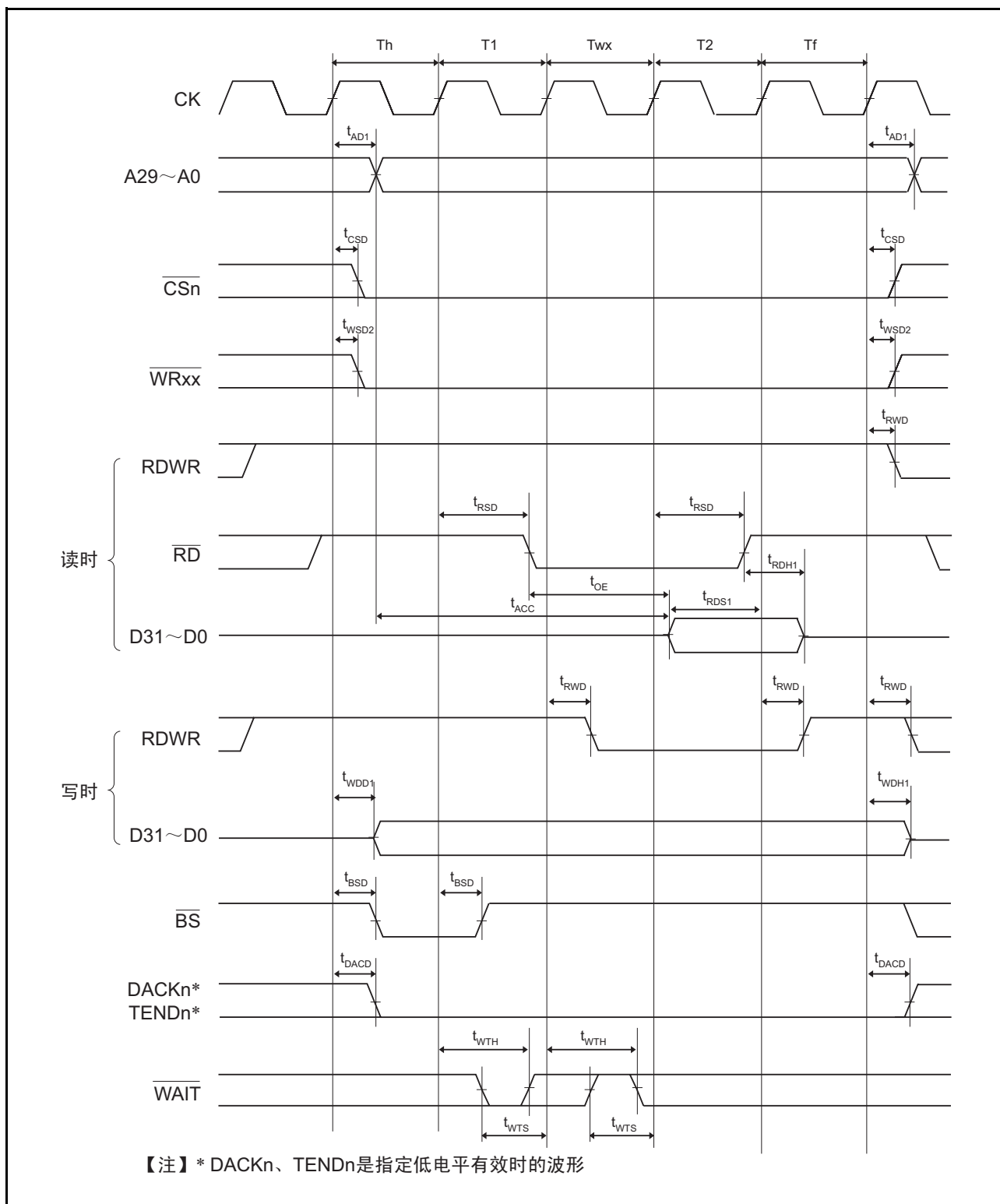


图 28.17 带字节选择的 SRAM 总线周期 (SW=1 个周期、HW=1 个周期、插入 1 个外部等待、BAS=1 (写周期 WE 控制))



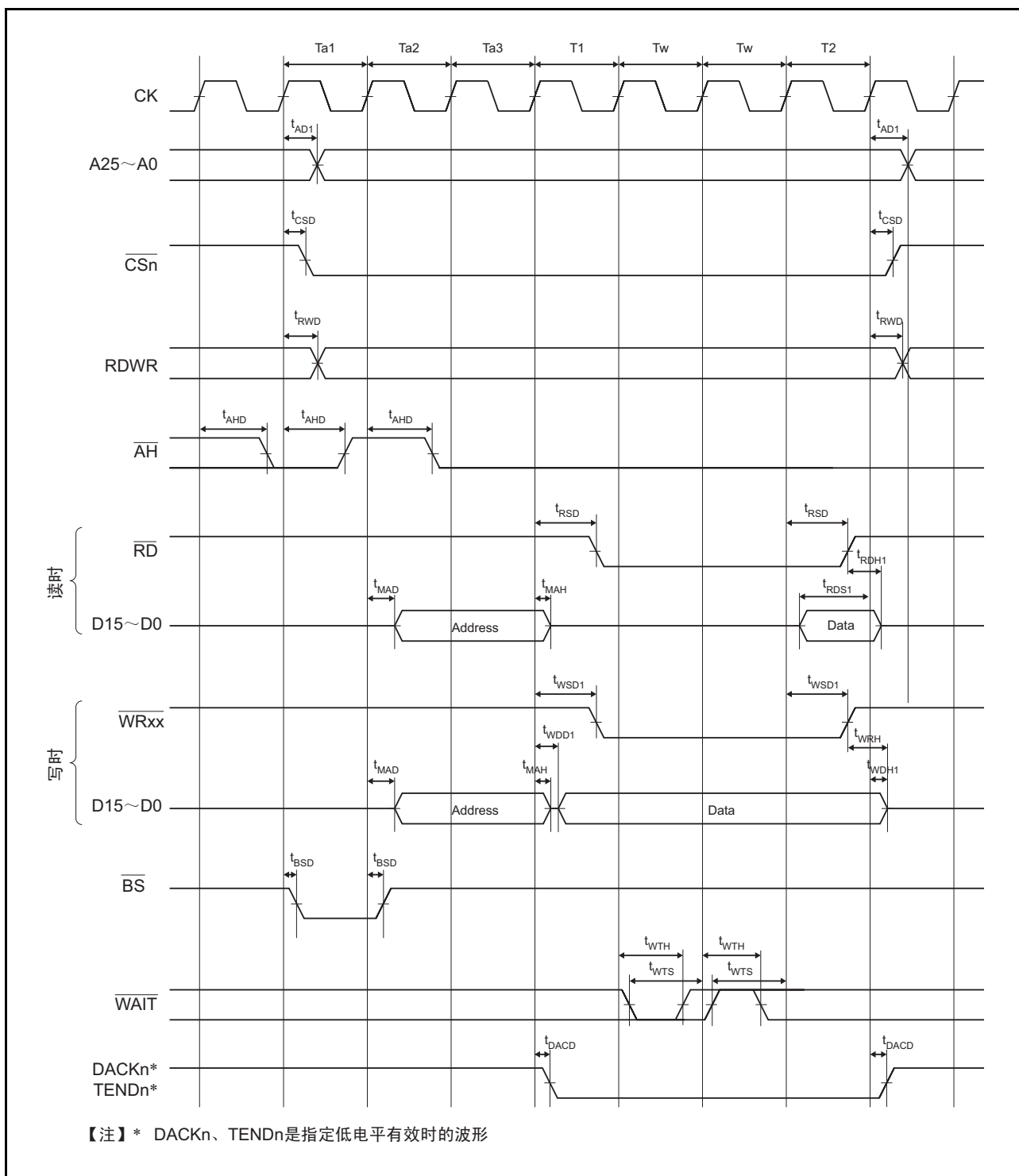


图 28.18 MPX 总 I/O 接口的总线周期  
(3 个地址周期、1 个软件等待、插入 1 个外部等待)

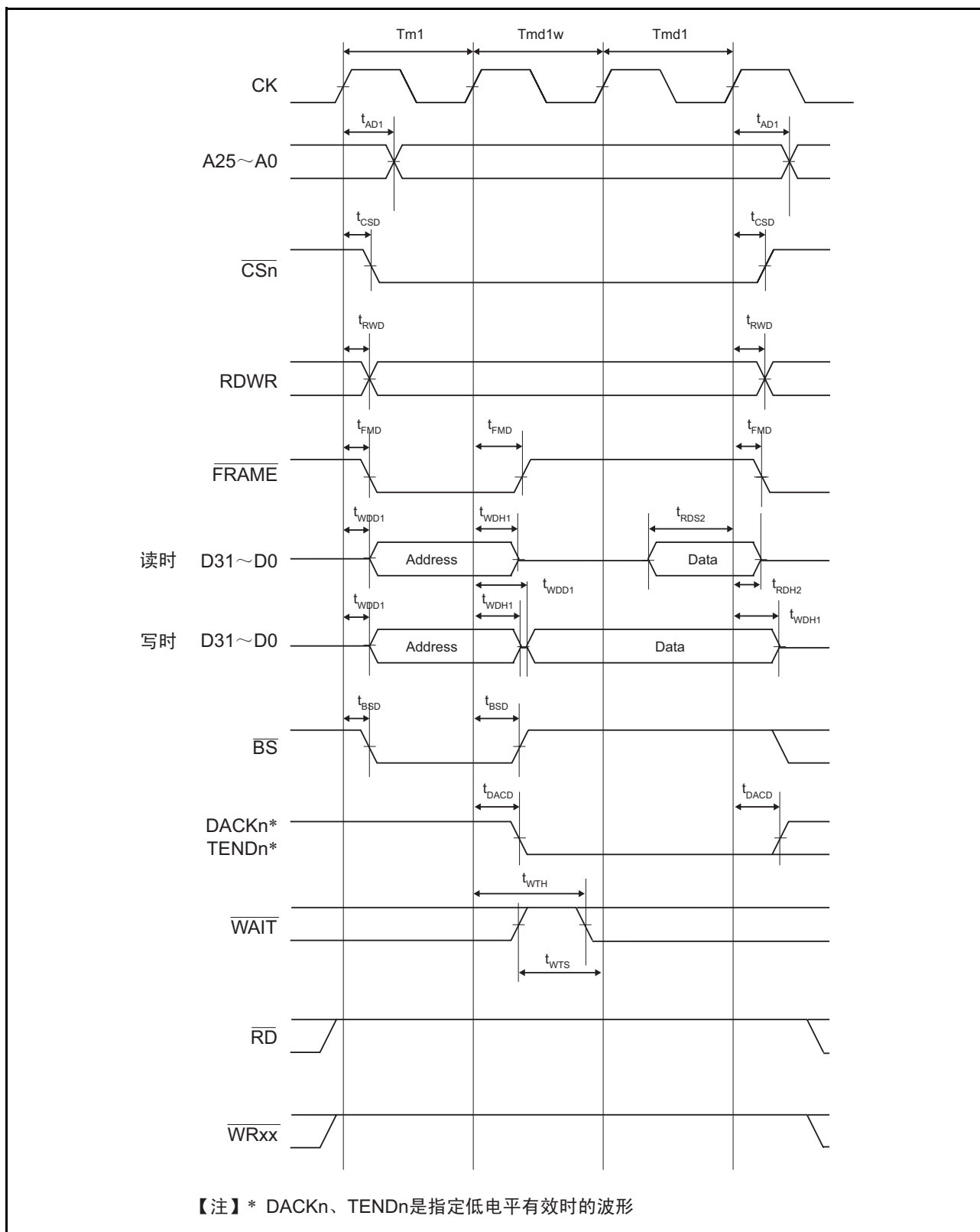


图 28.19 突发 MPX-I/O 接口的总线周期 单次读写  
(1 个地址周期、1 个软件等待)

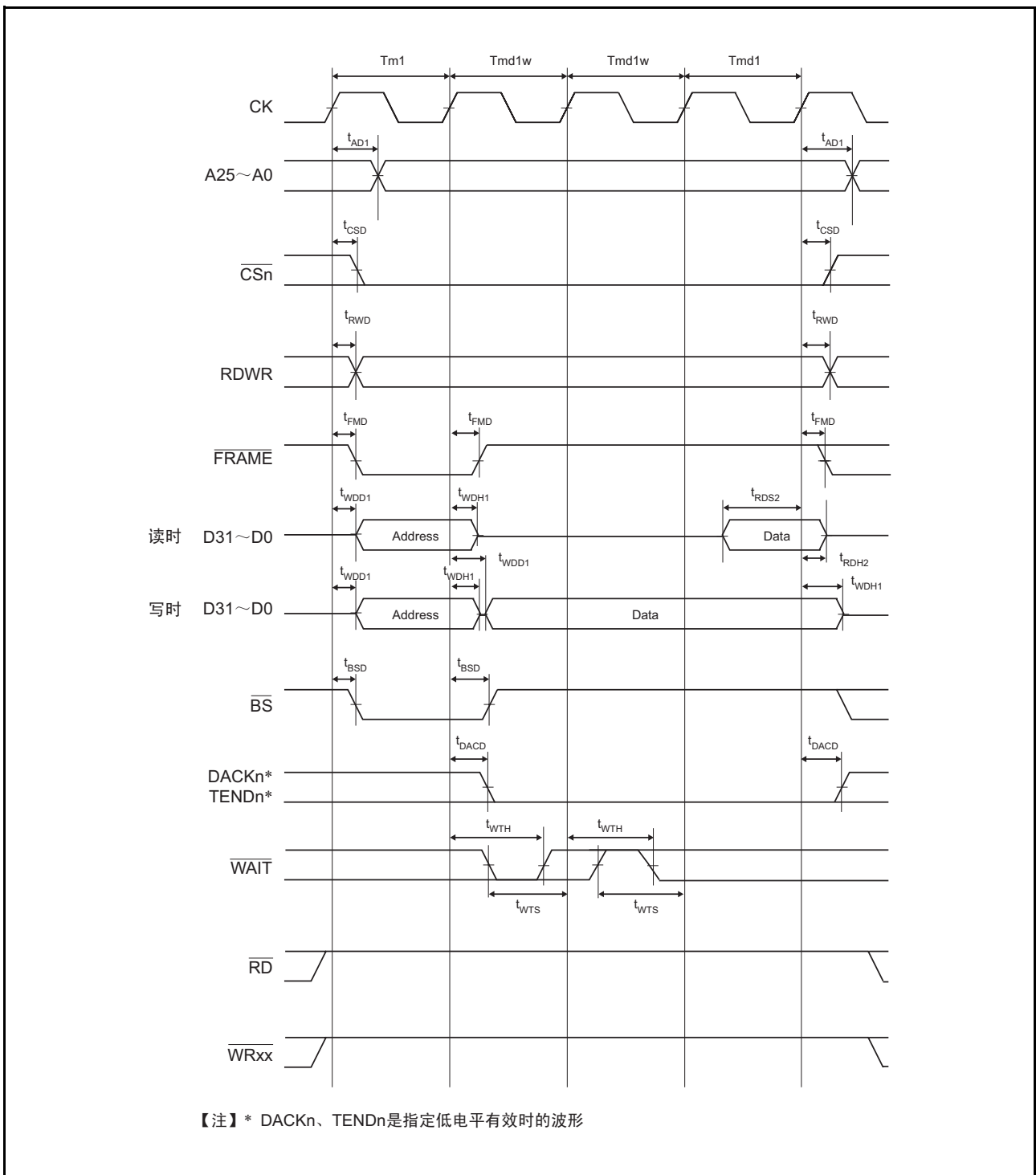


图 28.20 突发 MPX 接口的总线周期 单次读写  
(1 个地址周期、1 个软件等待、插入 1 个外部等待)

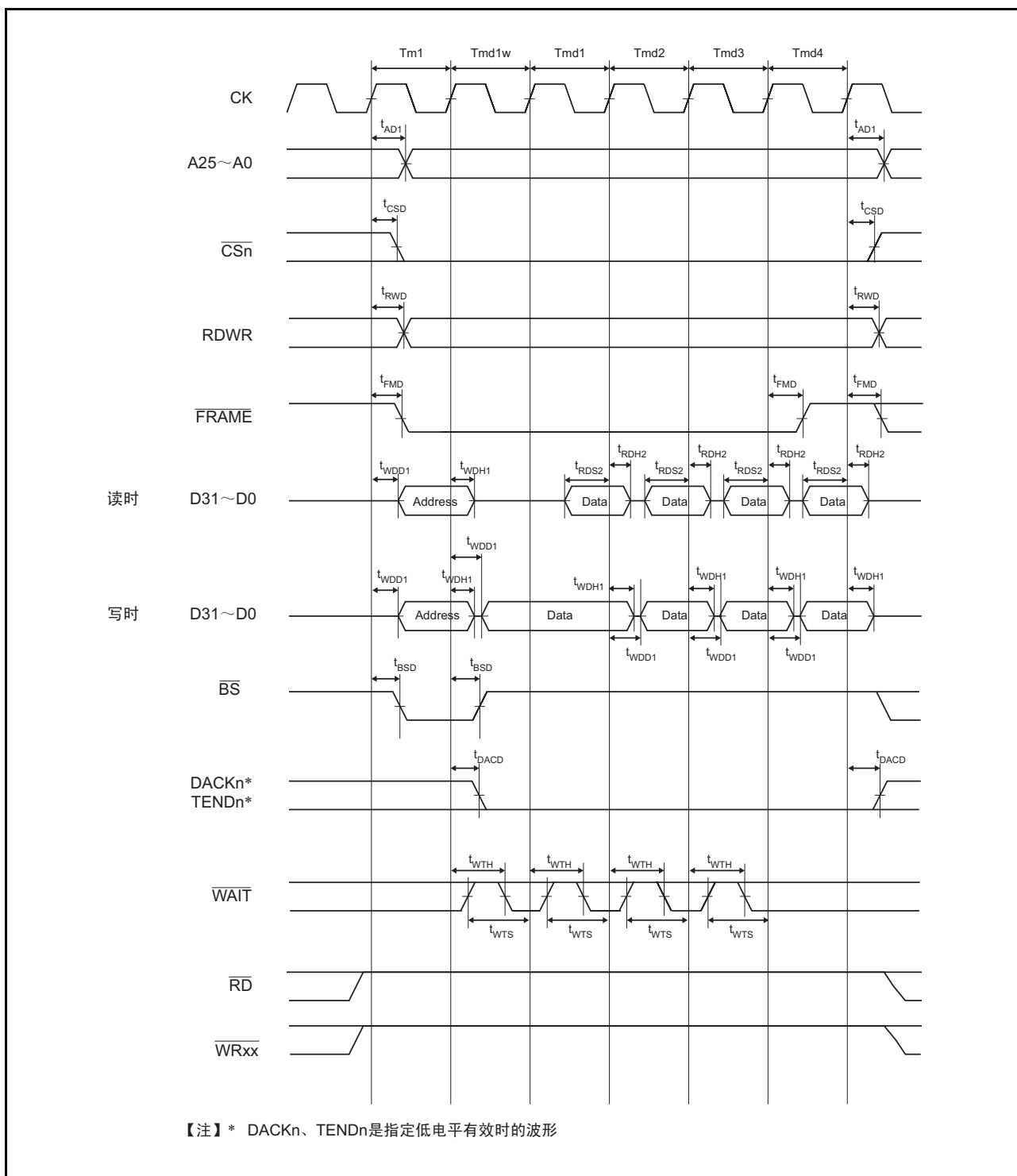


图 28.21 突发 MPX 接口的总线周期 突发读写  
(1 个地址周期、1 个软件等待)

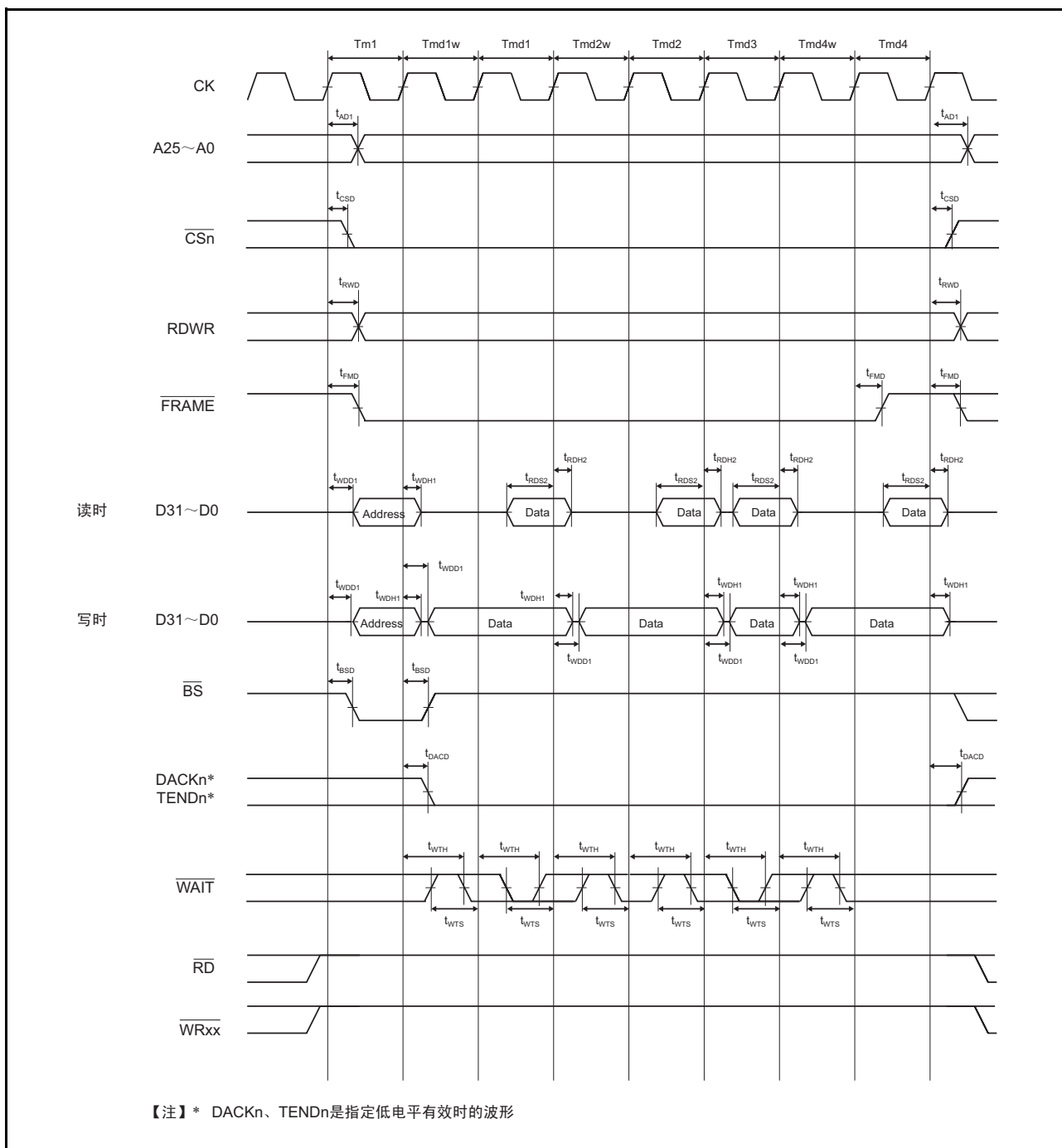


图 28.22 突发 MPX 接口的总线周期 突发读写  
(1 个地址周期、1 个软件等待、有外部等待插入)

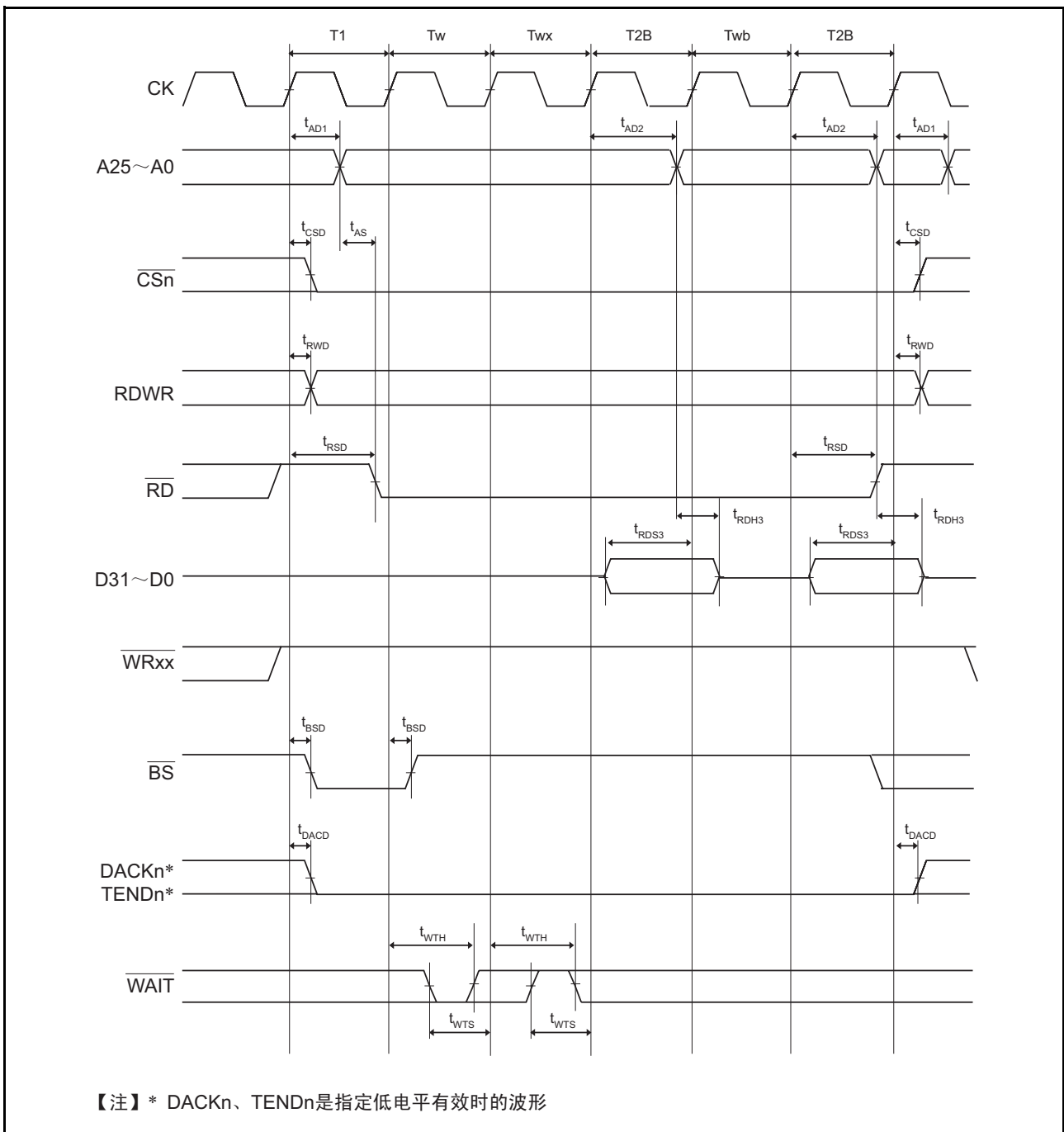


图 28.23 突发 ROM 的读周期  
(1 个软件等待、插入 1 个外部等待、1 个突发等待、2 个突发)

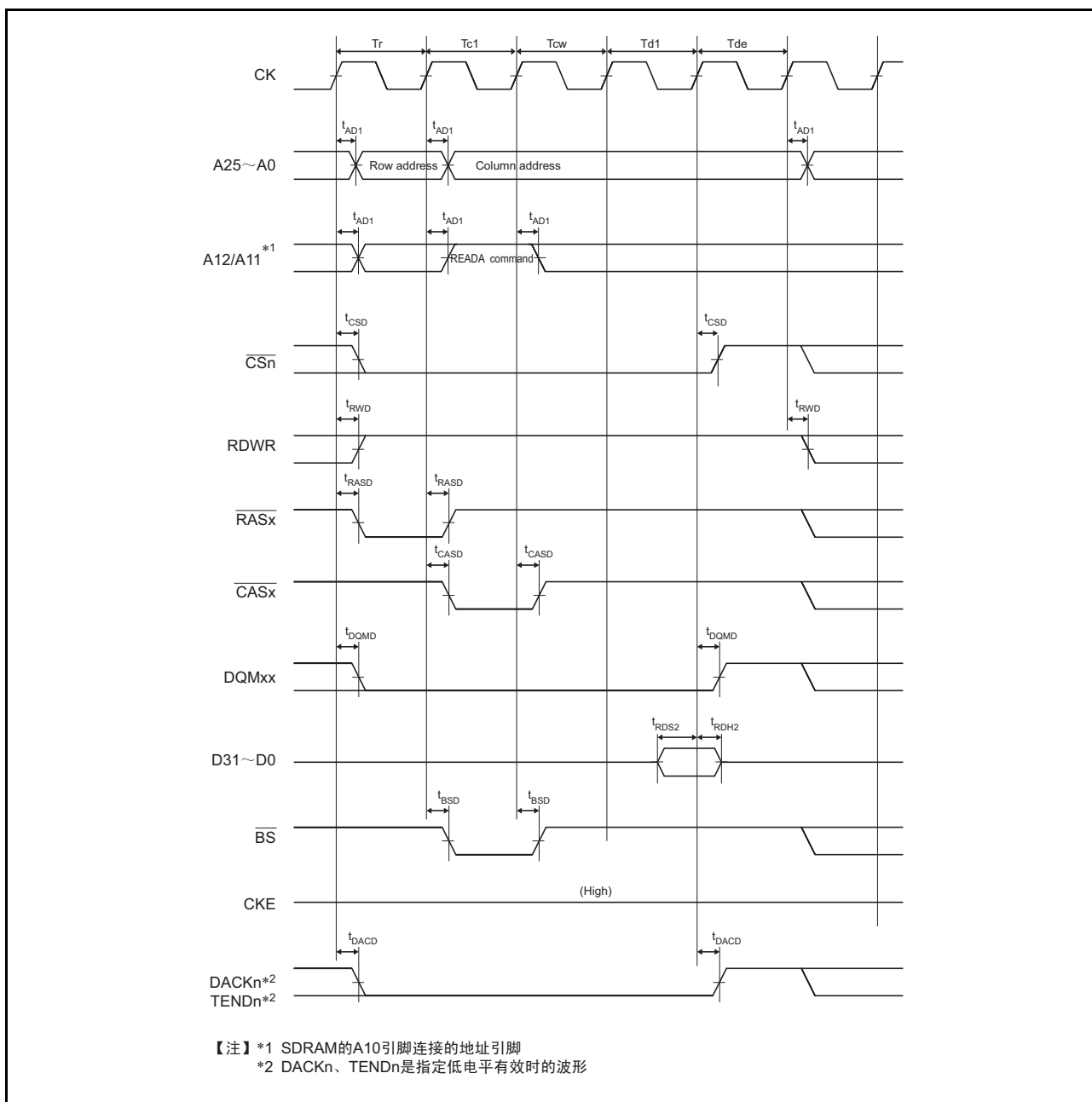


图 28.24 同步 DRAM 单次读的总线周期  
(有自动予充电、2 个 CAS 等待、WTRCD=0 个周期、WTRP=0 个周期)

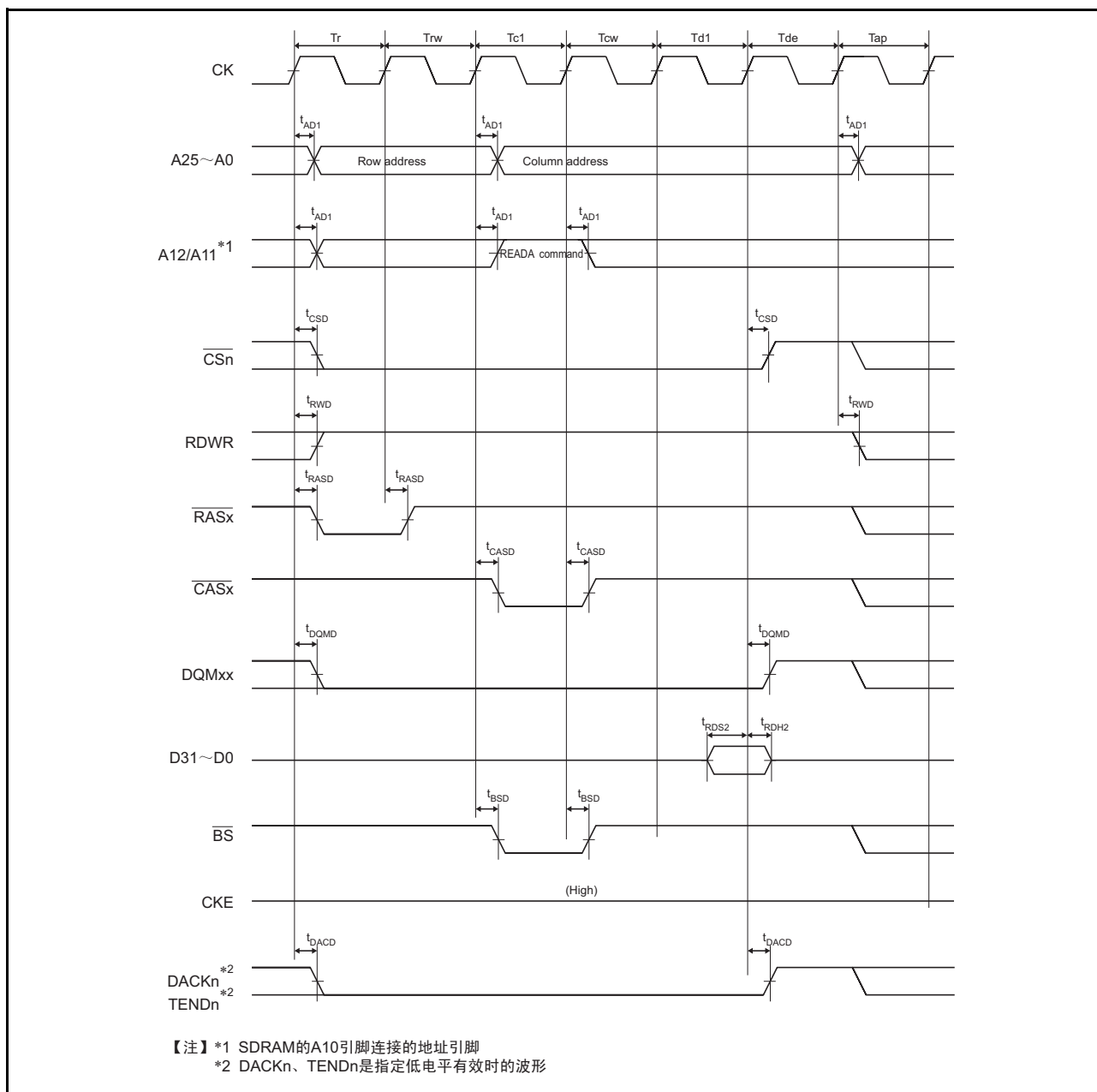


图 28.25 同步 DRAM 单次读的总线周期  
(有自动予充电、2 个 CAS 等待、WTRCD=1 个周期、WTRP=1 个周期)



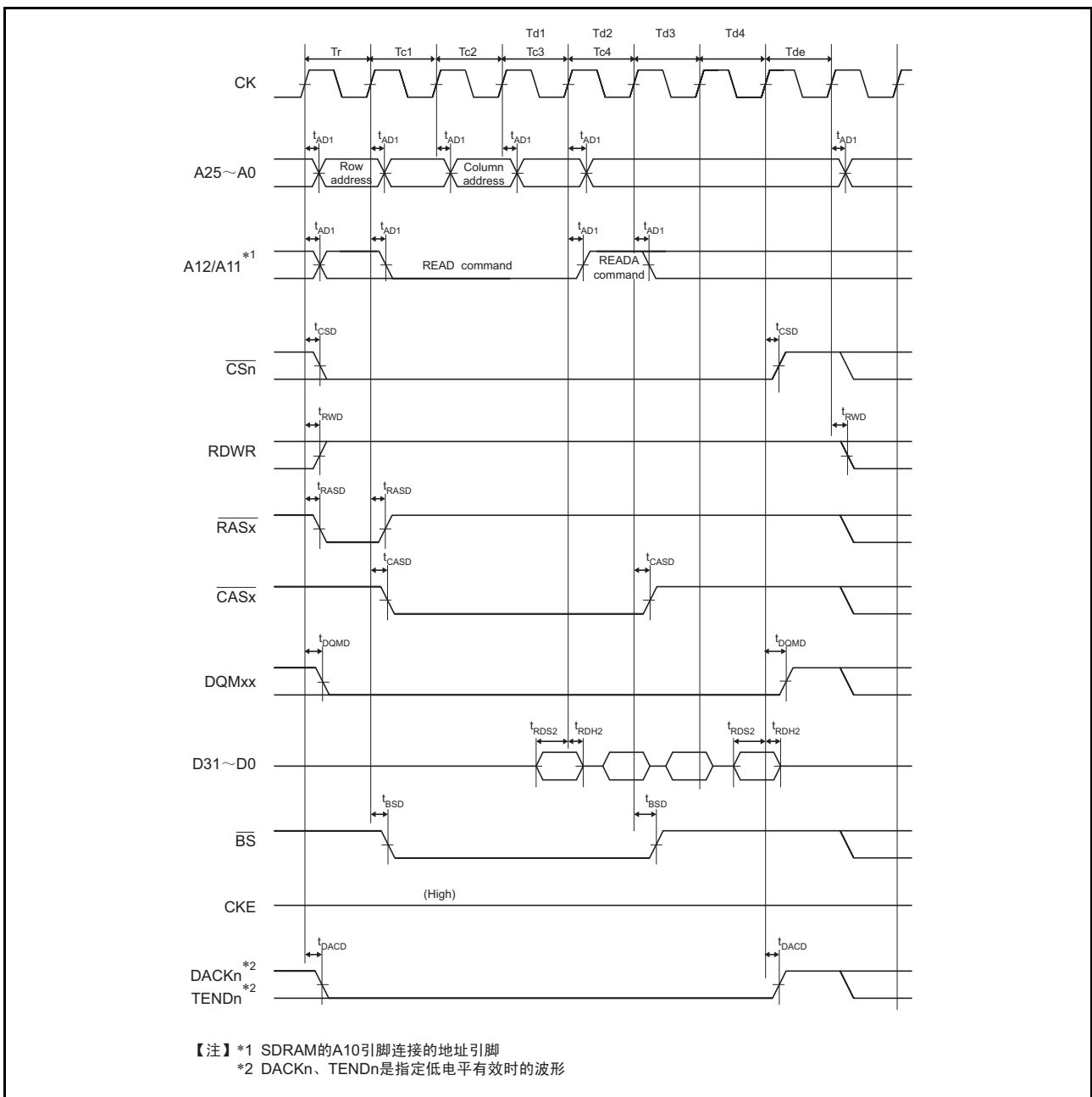


图 28.26 同步 DRAM 突发读的总线周期（4 个读周期）  
（有自动予充电、2 个 CAS 等待、WTRCD=0 个周期、WTRP=1 个周期）

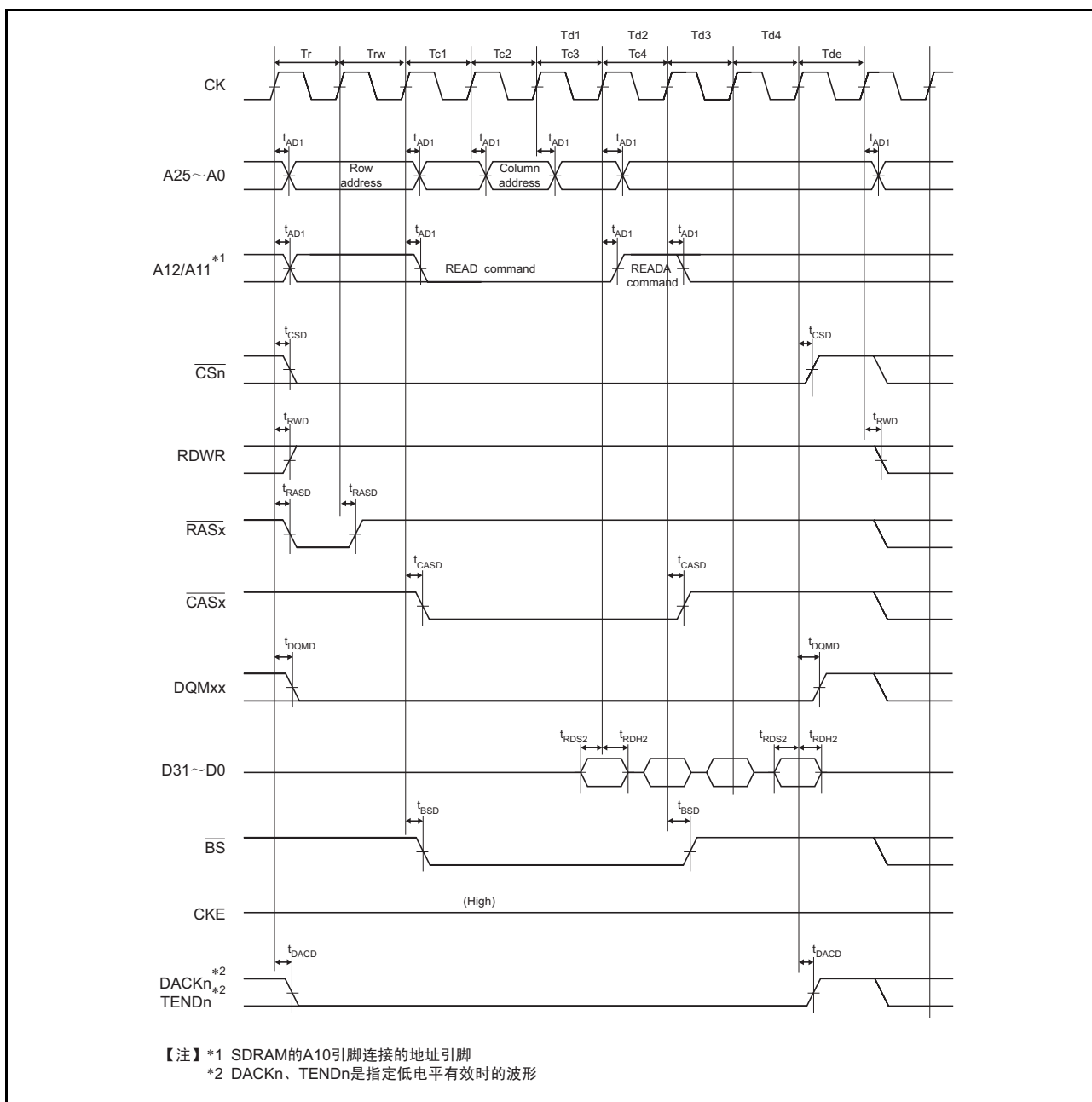


图 28.27 同步 DRAM 突发读的总线周期（4 个读周期）  
（有自动予充电、2 个 CAS 等待、WTRCD=1 个周期、WTRP=0 个周期）

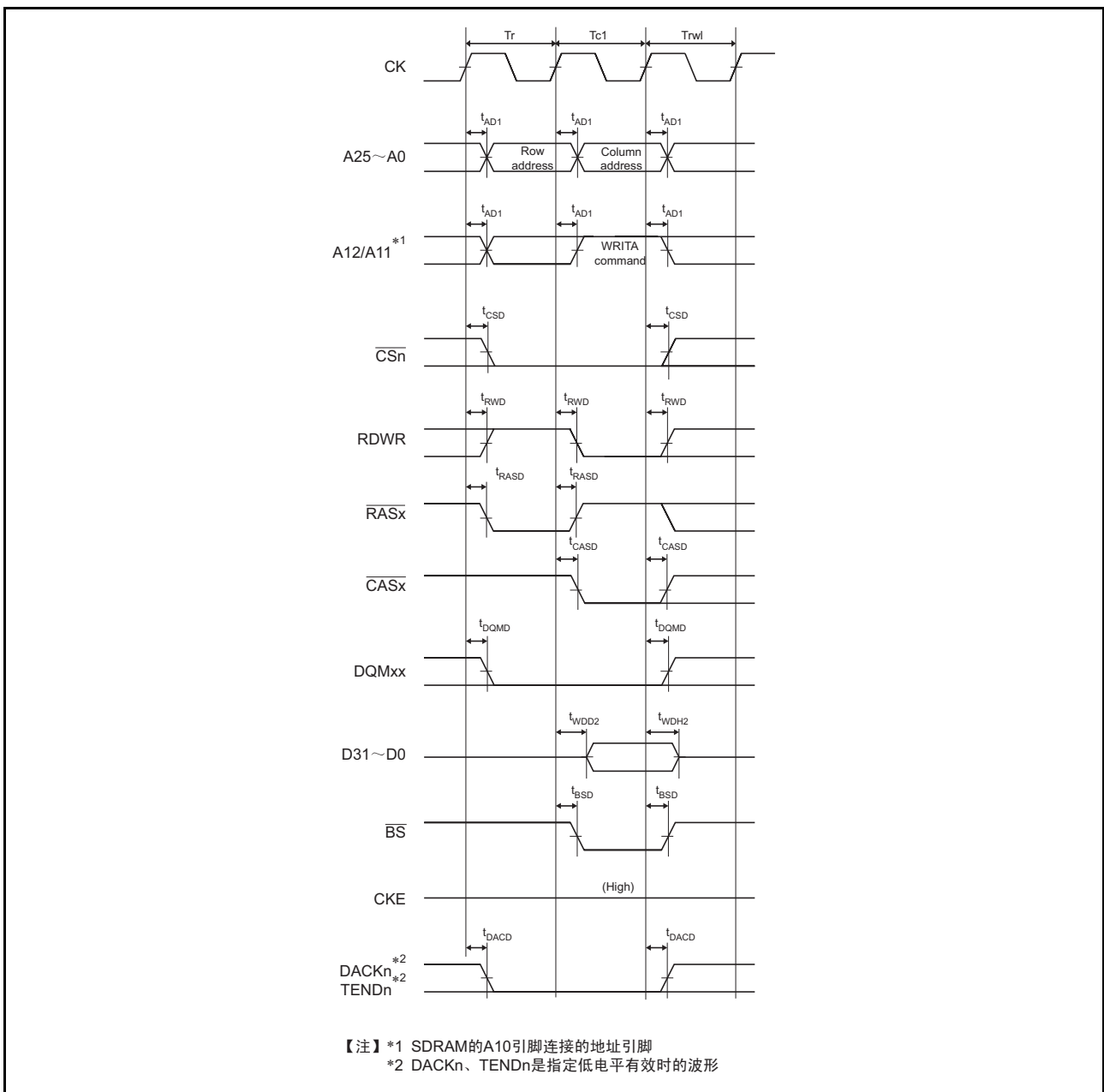


图 28.28 同步 DRAM 单次写的总线周期  
(有自动予充电、TRWL=1 个周期)

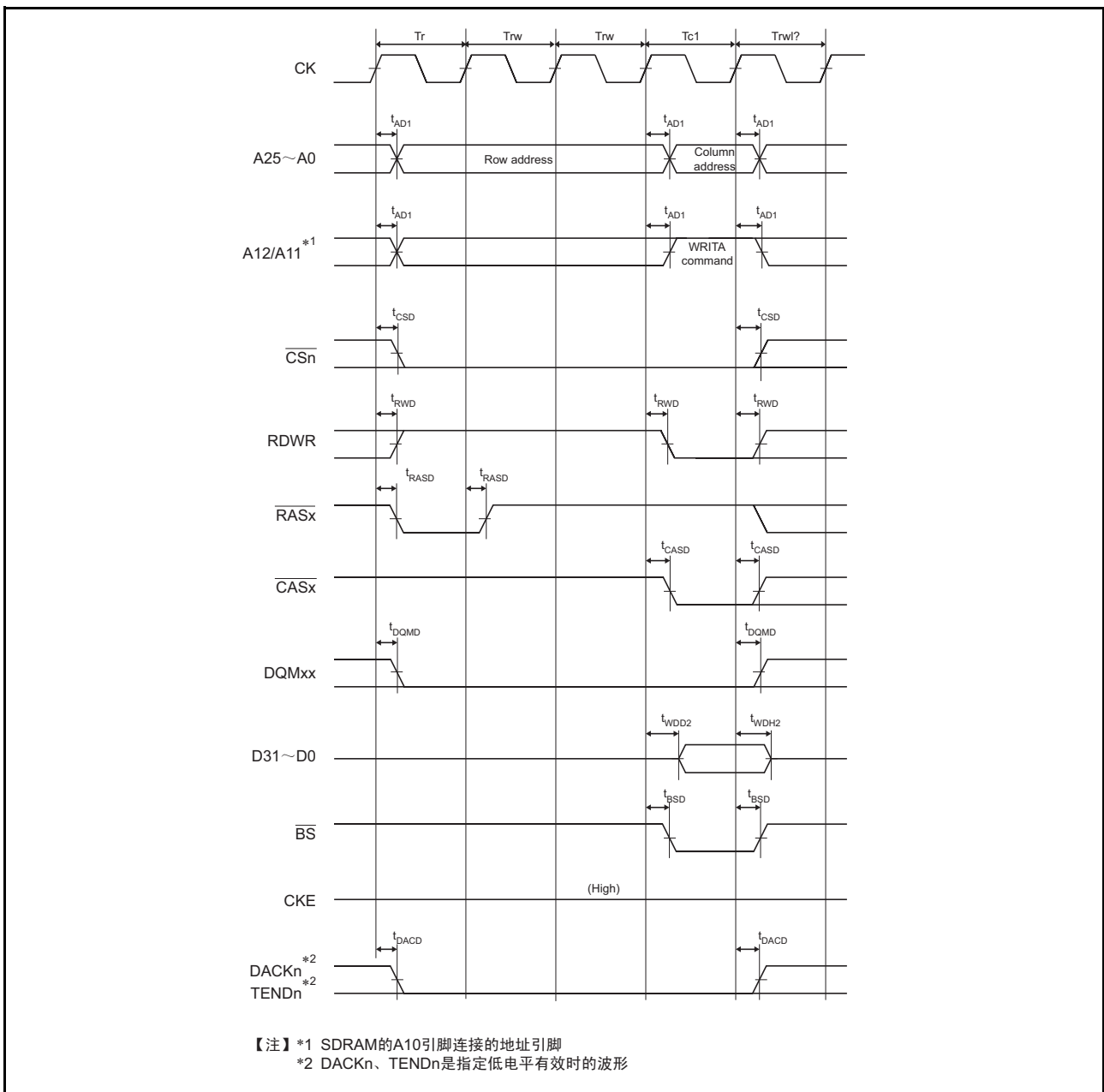


图 28.29 同步 DRAM 单次写的总线周期  
(有自动予充电、WTRCD=2 个周期、TRWL=1 个周期)

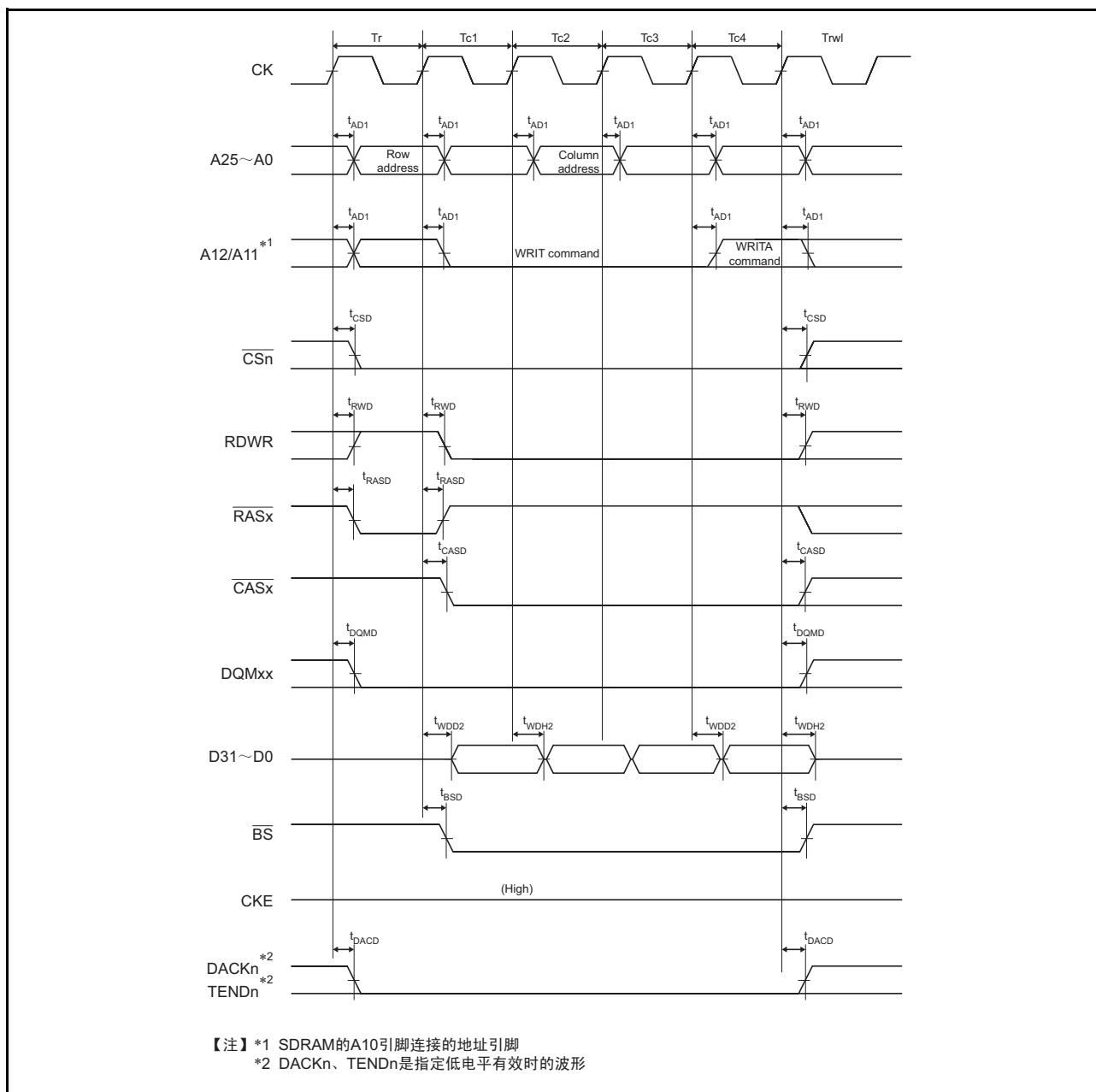


图 28.30 同步 DRAM 突发写的总线周期（4 个写周期）  
（有自动予充电、WTRCD=0 个周期、TRWL=1 个周期）

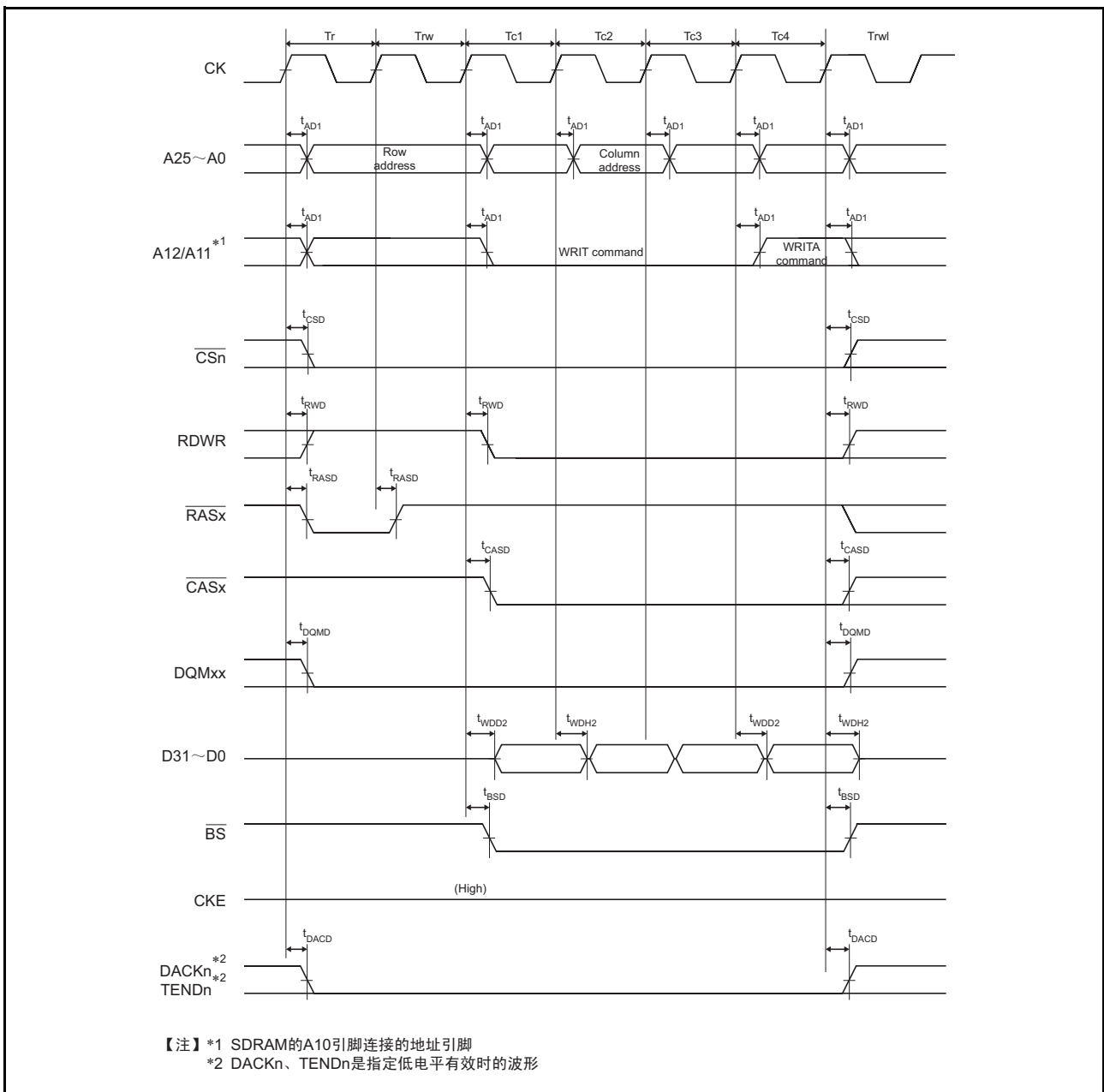


图 28.31 同步 DRAM 突发写的总线周期（4 个写周期）  
（有自动予充电、WTRCD=1 个周期、TRWL=1 个周期）

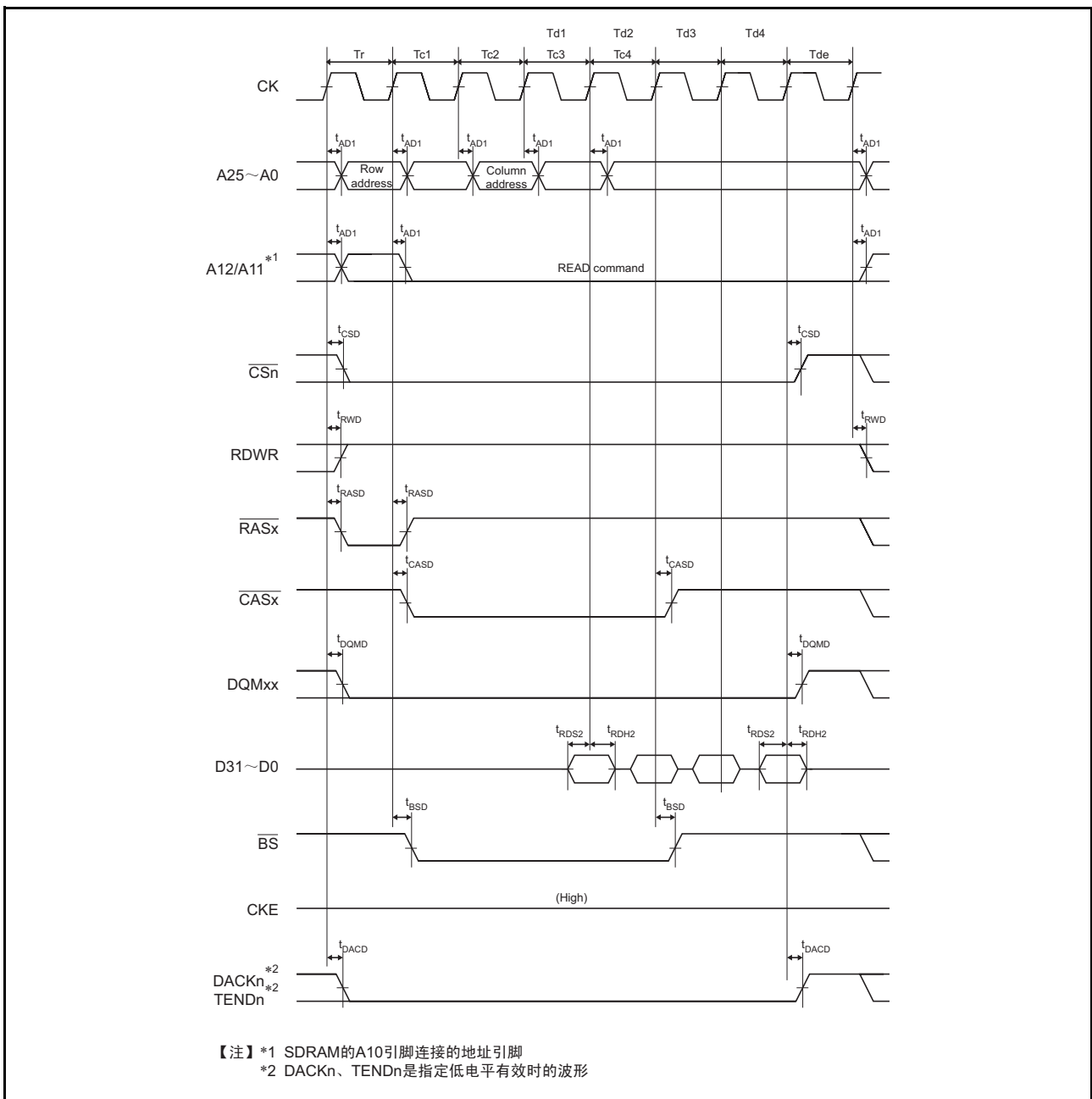


图 28.32 同步 DRAM 突发读的总线周期（4 个读周期）  
（存储体激活模式：ACT+READ 命令、2 个 CAS 等待、WTRCD=0 个周期）

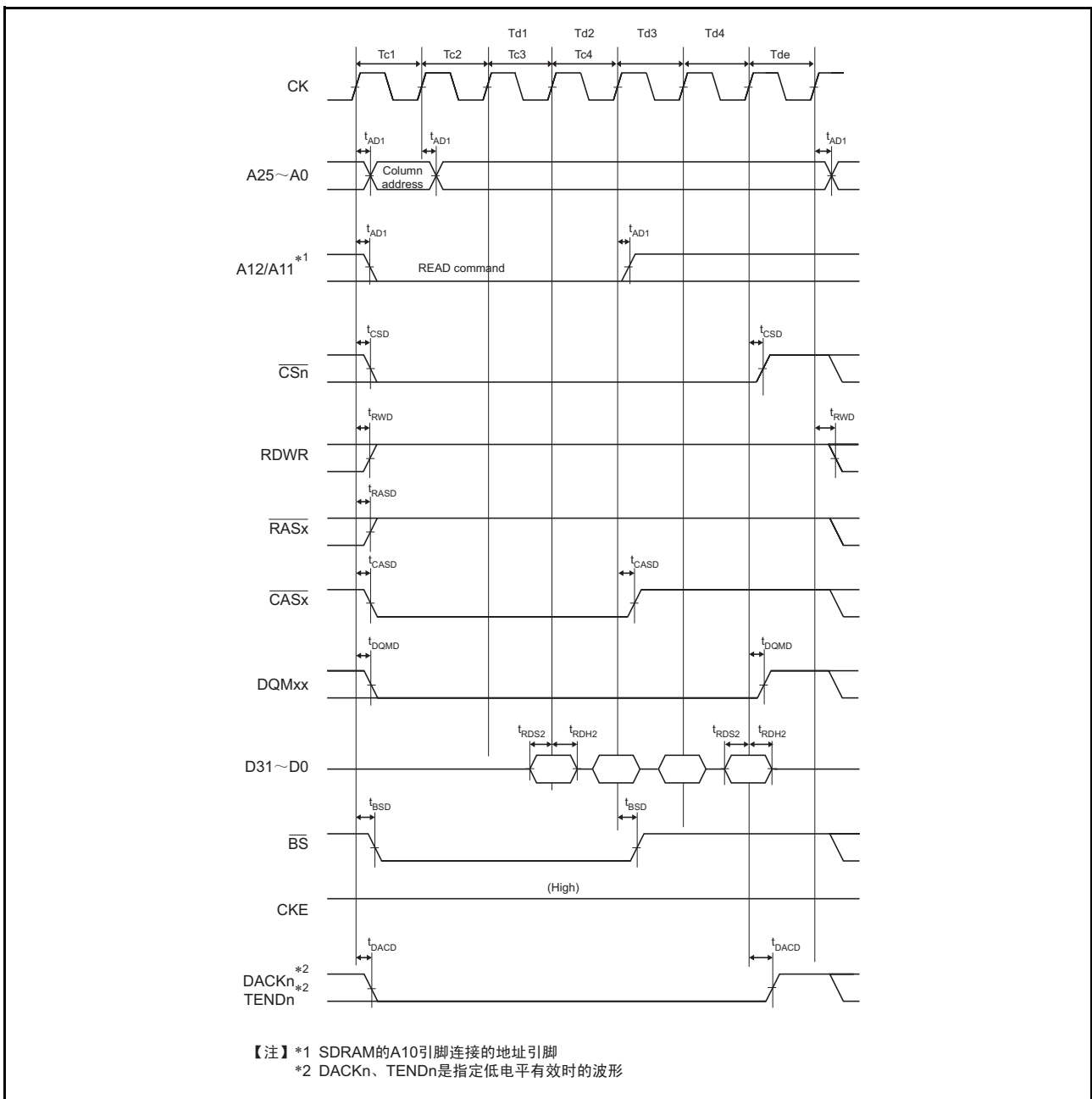


图 28.33 同步 DRAM 突发读的总线周期（4 个读周期）  
（存储体激活模式：READ 命令、同一行地址、2 个 CAS 等待、WTRCD=0 个周期）



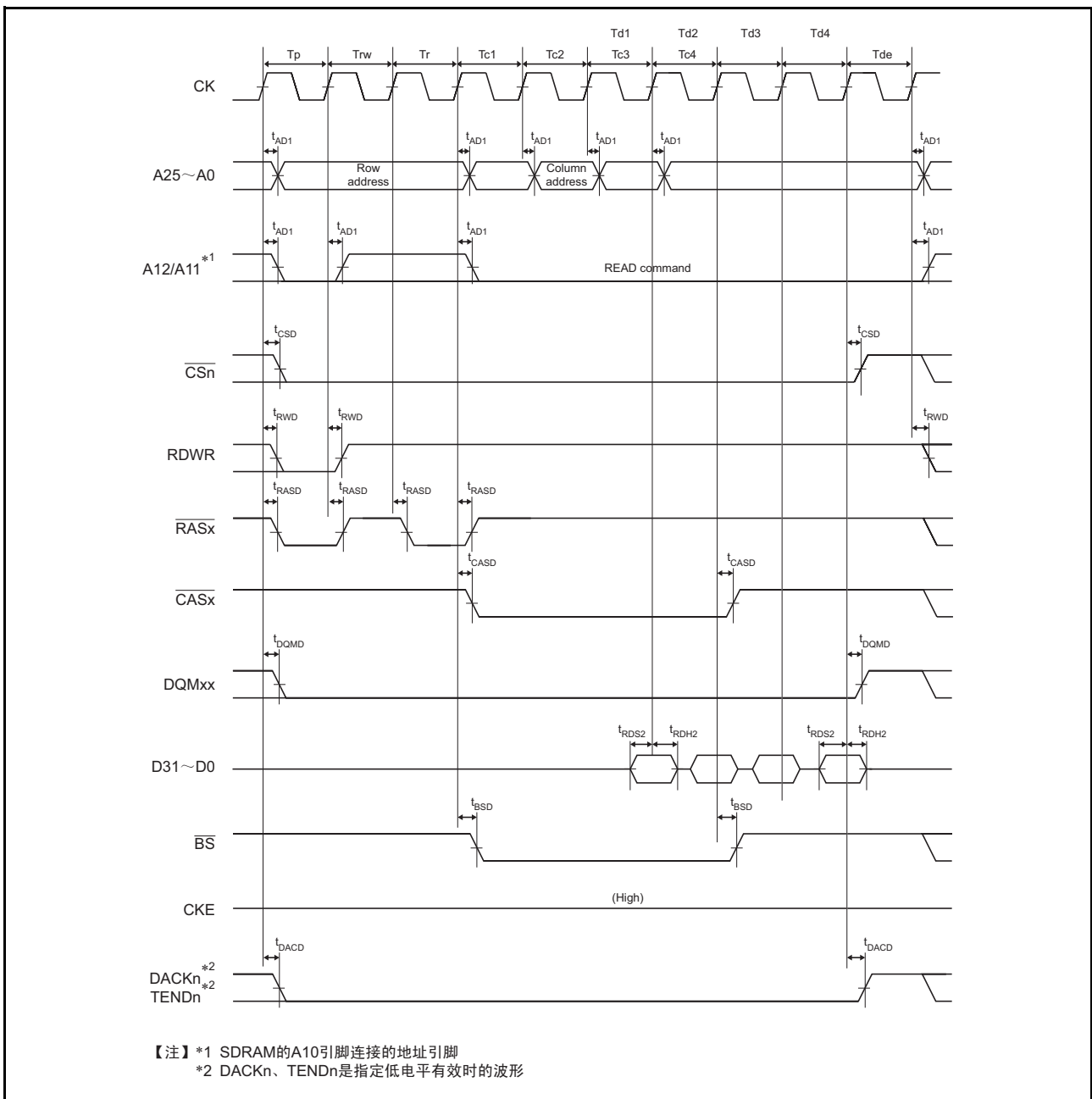


图 28.34 同步 DRAM 突发读的总线周期（4 个读周期）  
（存储体激活模式：PRE+ACT+READ 命令、不同行地址、2 个 CAS 等待、WTRCD=0 个周期）

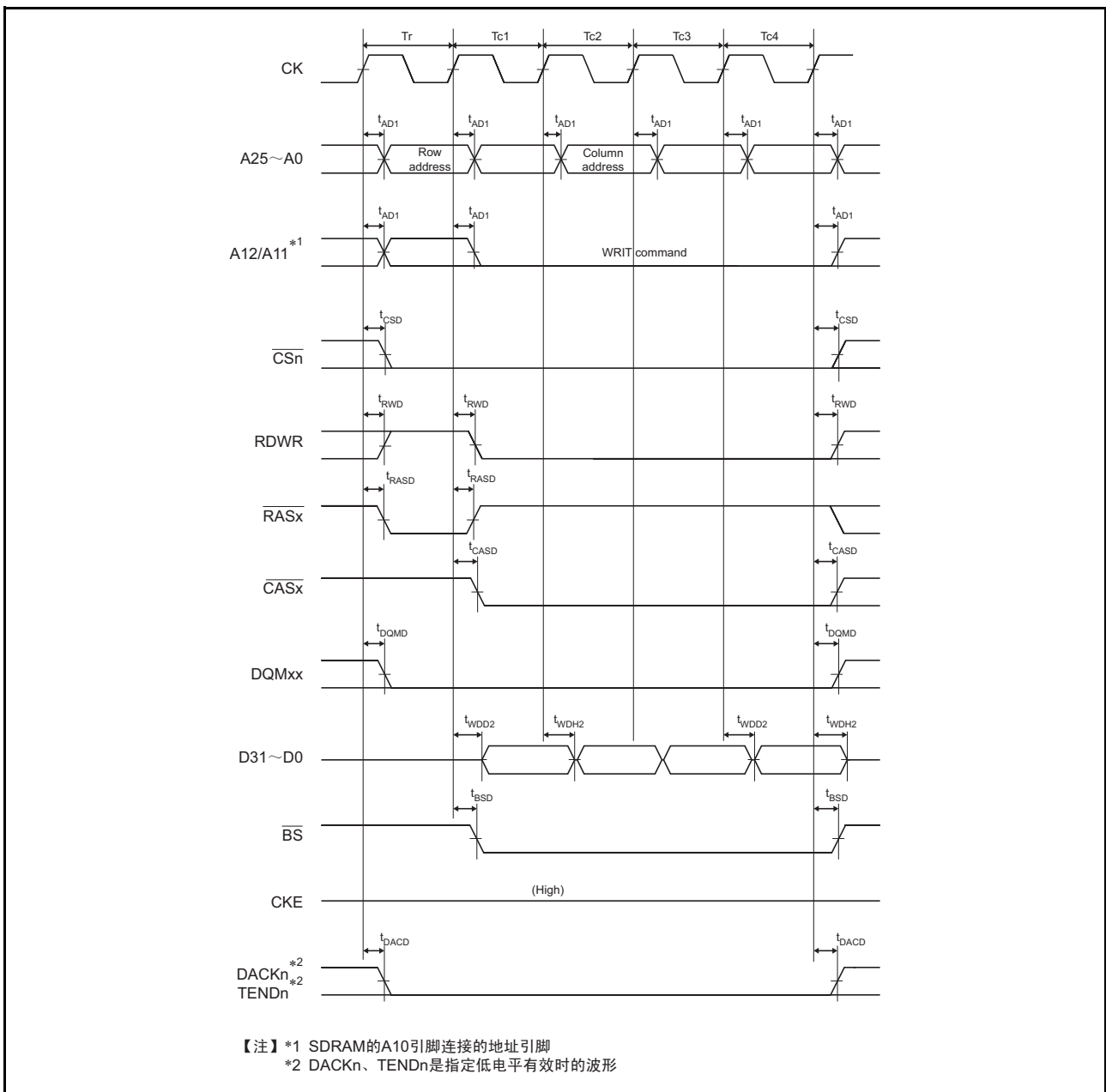


图 28.35 同步 DRAM 突发写的总线周期（4 个写周期）  
（存储体激活模式：ACT+WRITE 命令、WTRCD=0 个周期、TRWL=0 个周期）

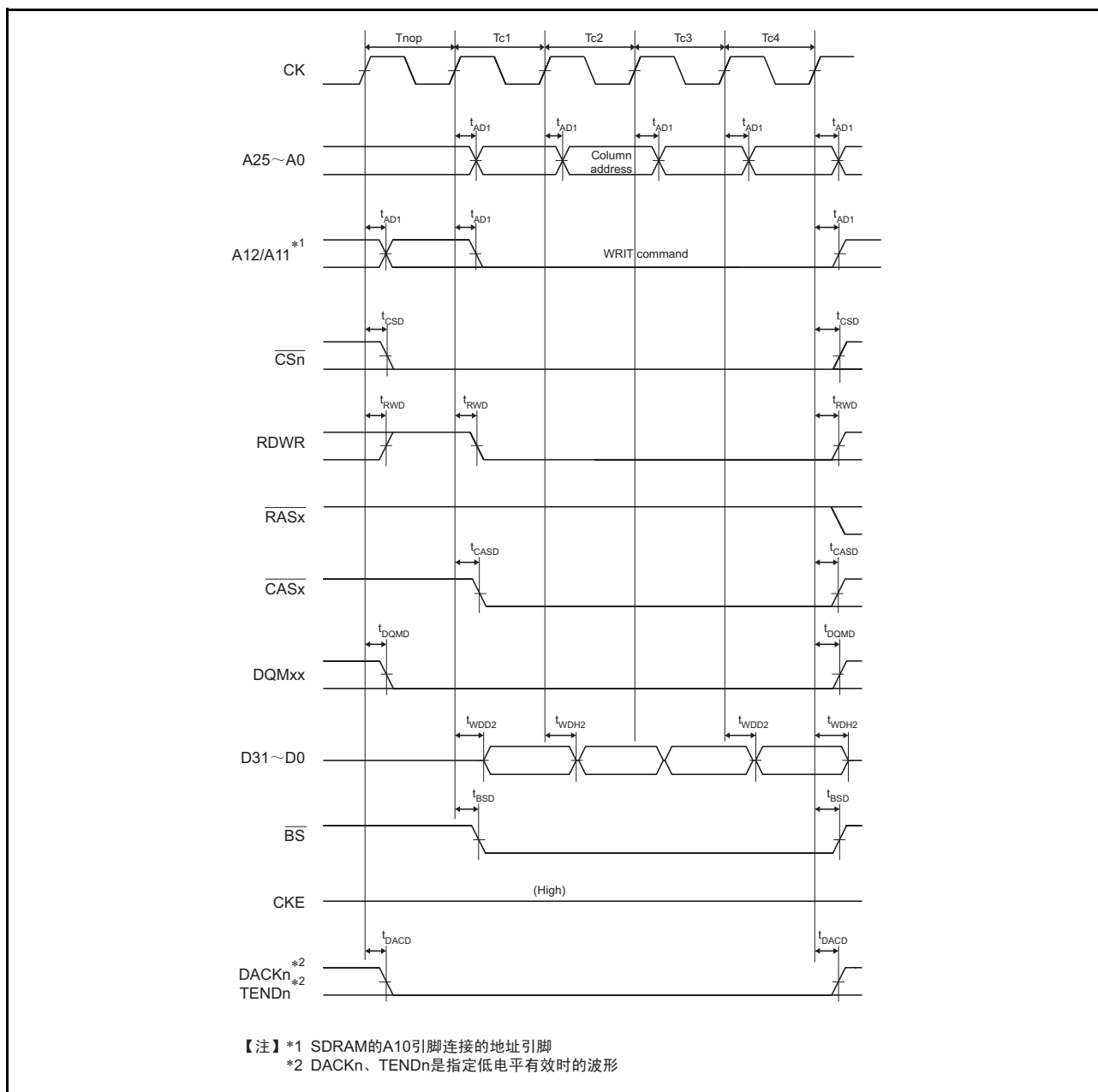


图 28.36 同步 DRAM 突发写的总线周期（4 个写周期）  
（存储体激活模式：WRITE 命令、同一行地址、WTRCD=0 个周期、TRWL=0 个周期）

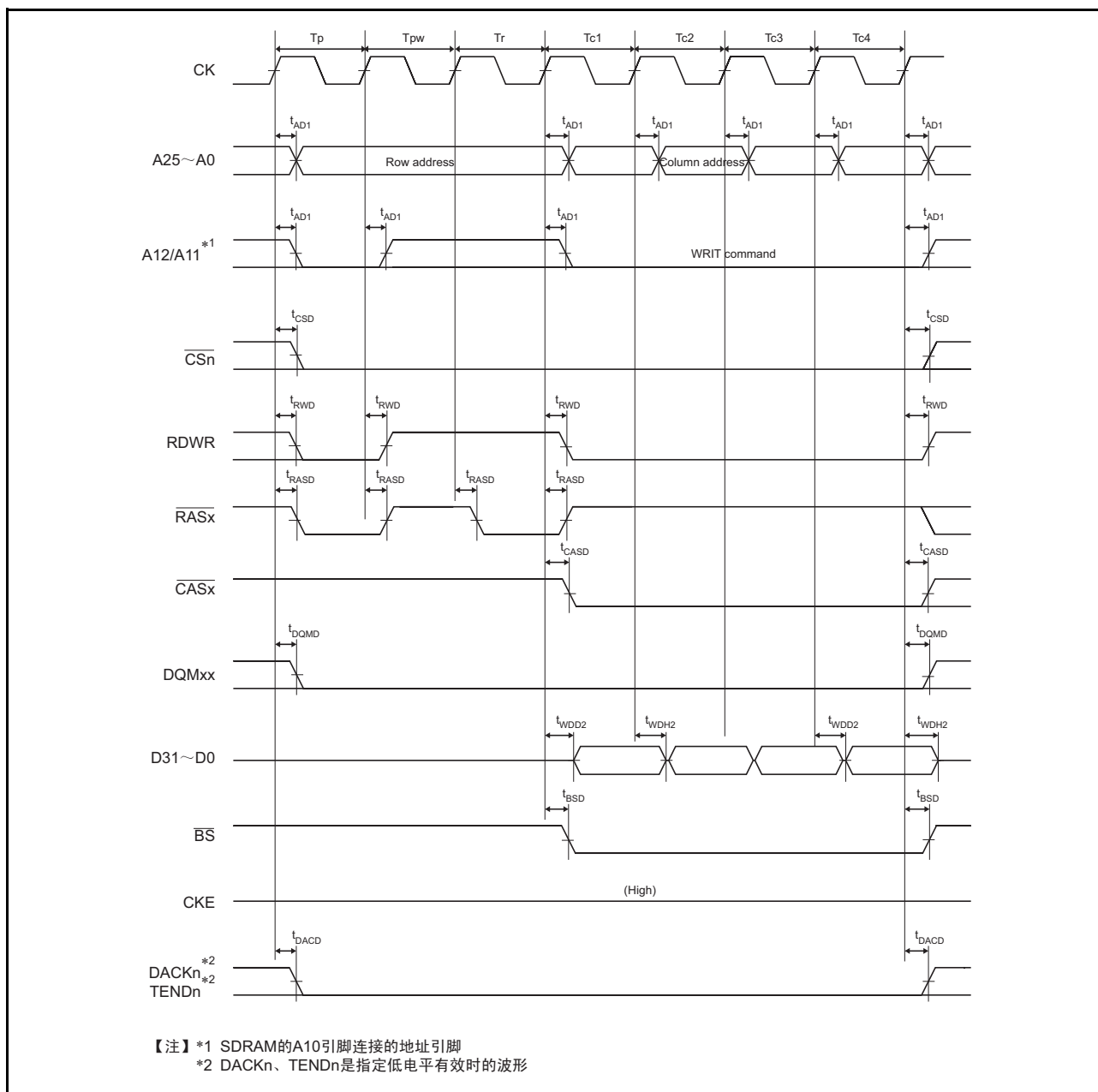


图 28.37 同步 DRAM 突发写的总线周期 (4 个写周期)  
(存储体激活模式: PRE+ACT+WRITE 命令、不同行地址、WTRCD=0 个周期、TRWL=0 个周期)

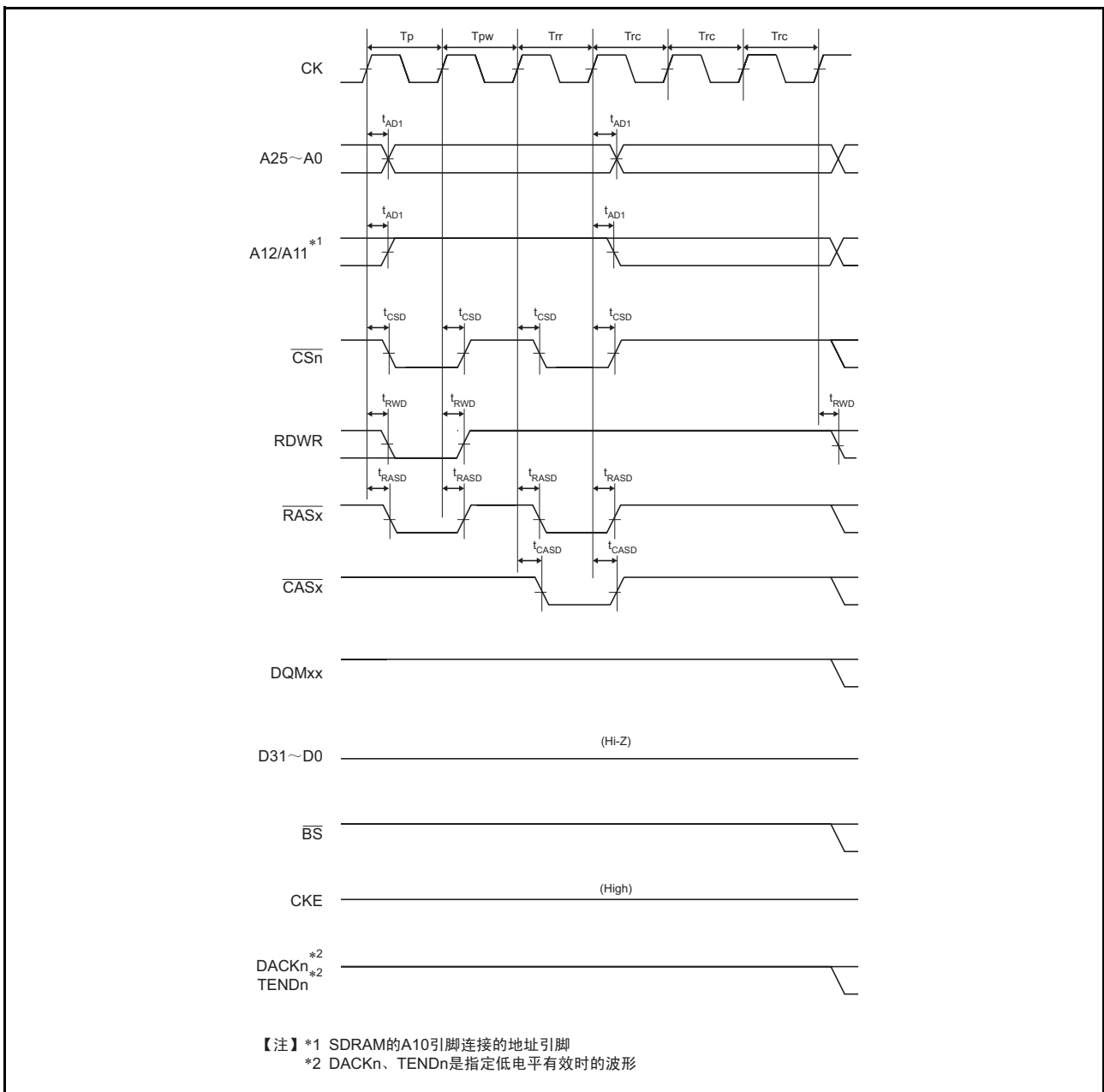


图 28.38 同步 DRAM 的自动刷新时序 (WTRP=1 个周期、WTRC=3 个周期)

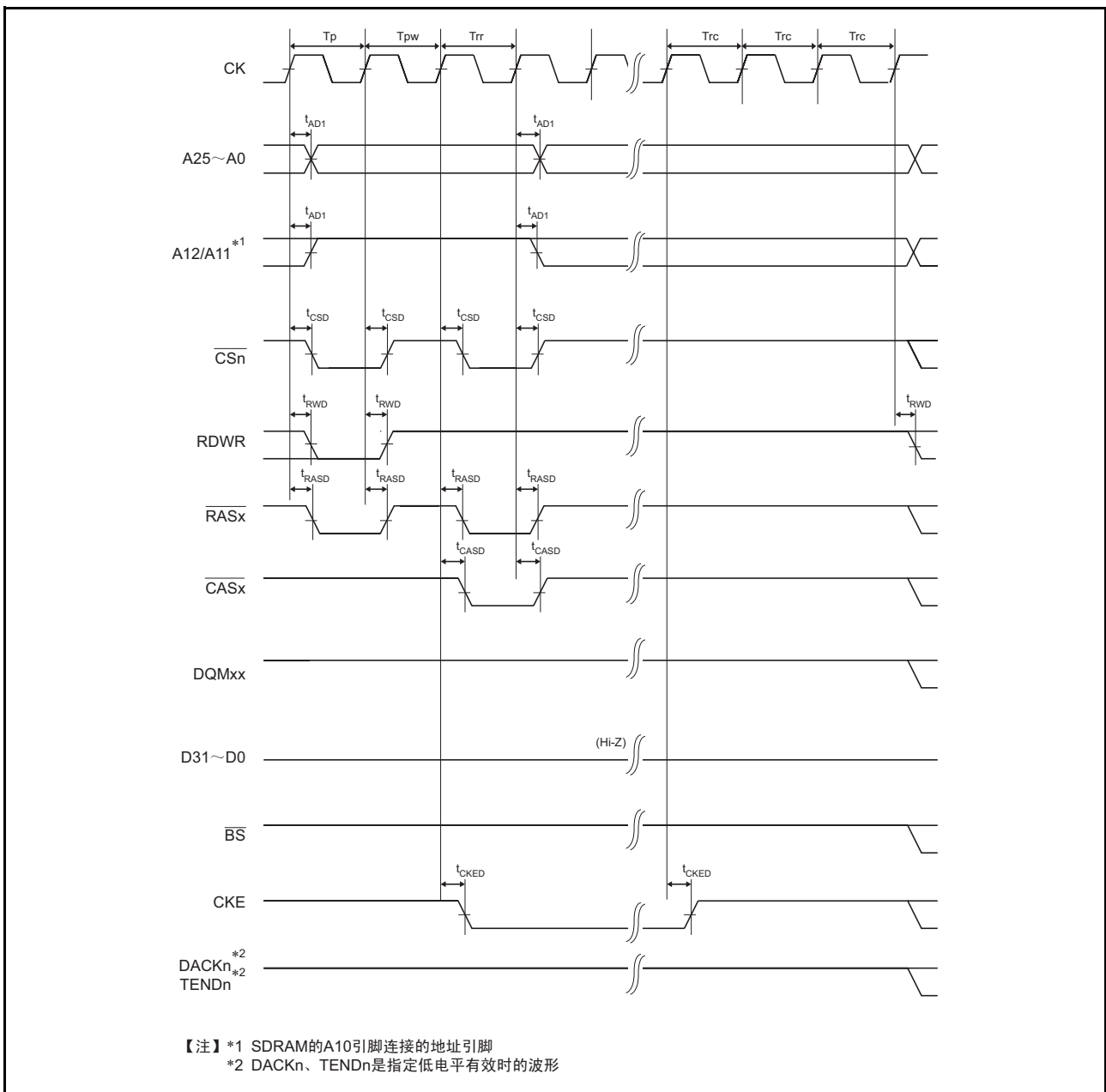


图 28.39 同步 DRAM 的自刷新时序 (WTRP=1 个周期、WTRC=3 个周期)

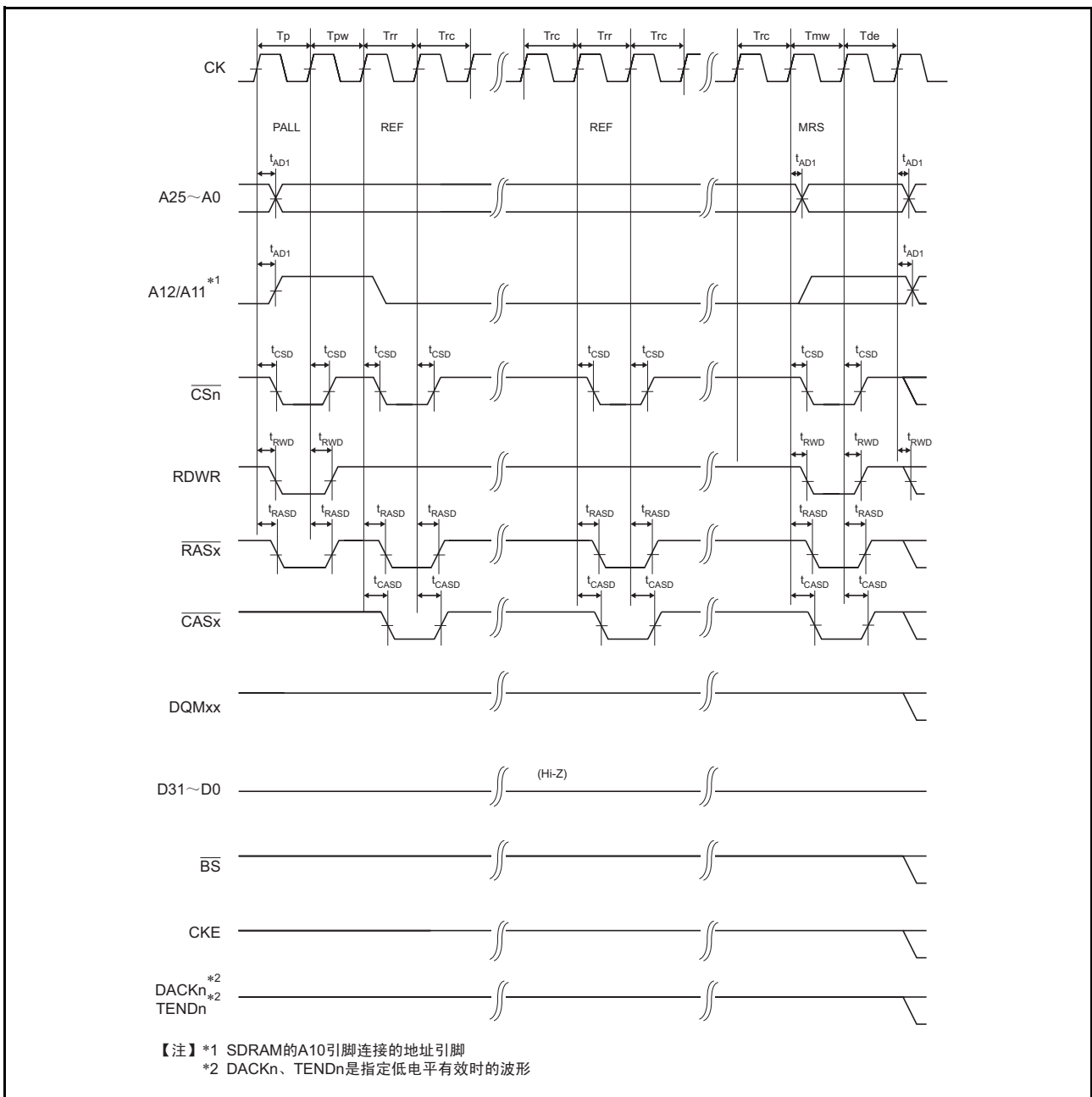


图 28.40 同步 DRAM 模式寄存器的写时序 (WTRP=1 个周期)

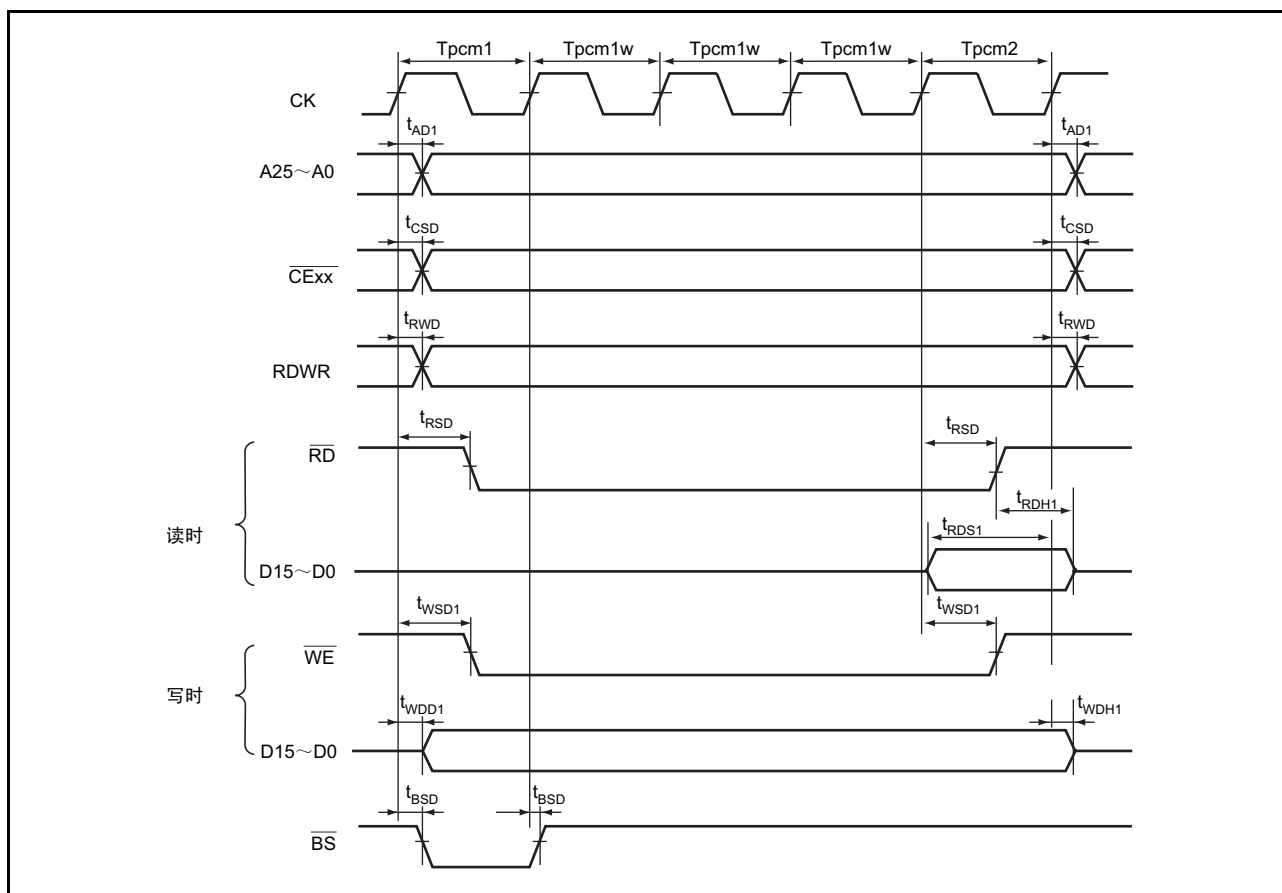


图 28.41 PCMCIA 存储器卡接口的总线时序



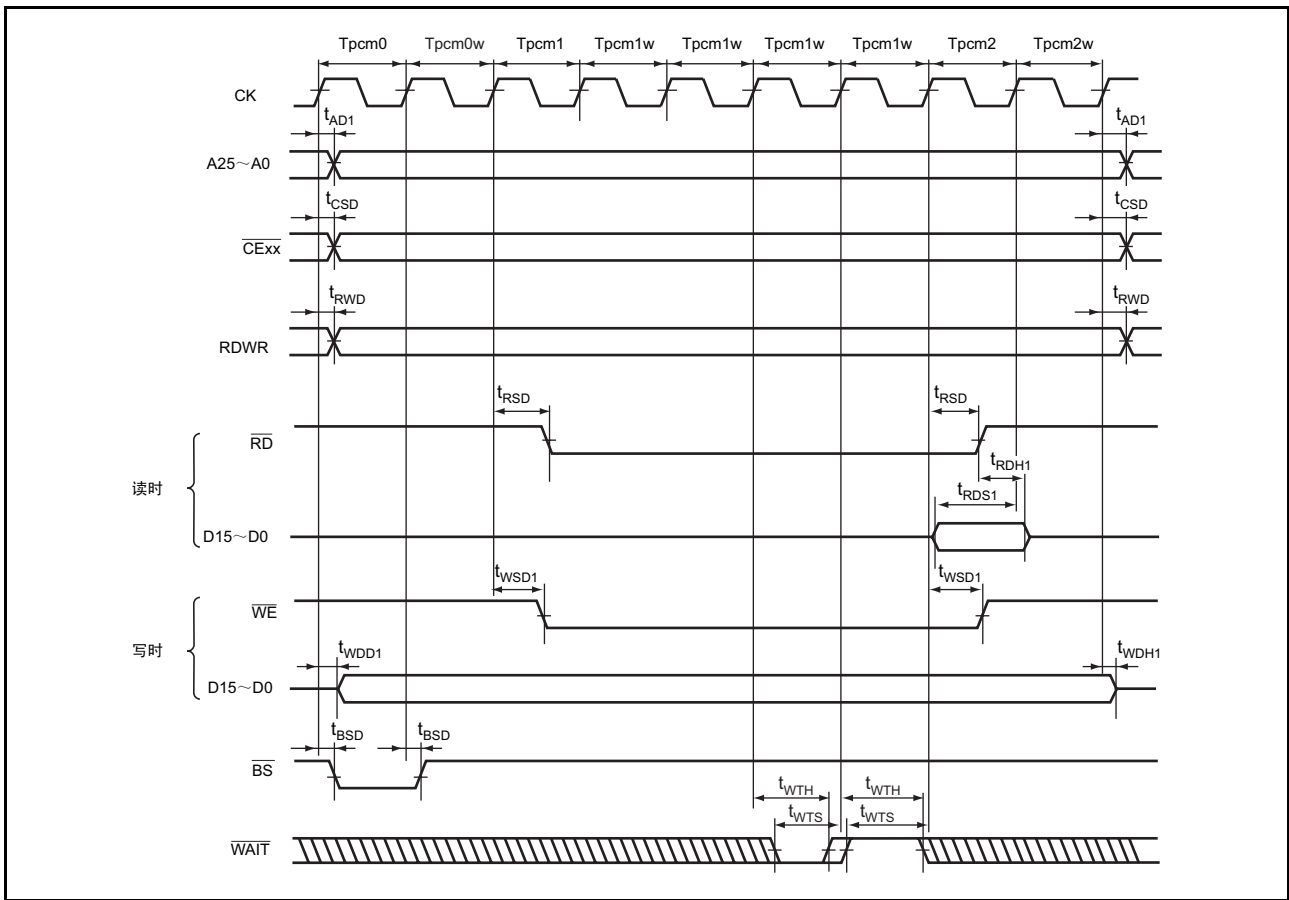


图 28.42 PCMCIA 存储器卡接口的总线时序  
(TED=2.5 个周期、TEH=1.5 个周期、1 个外部等待)

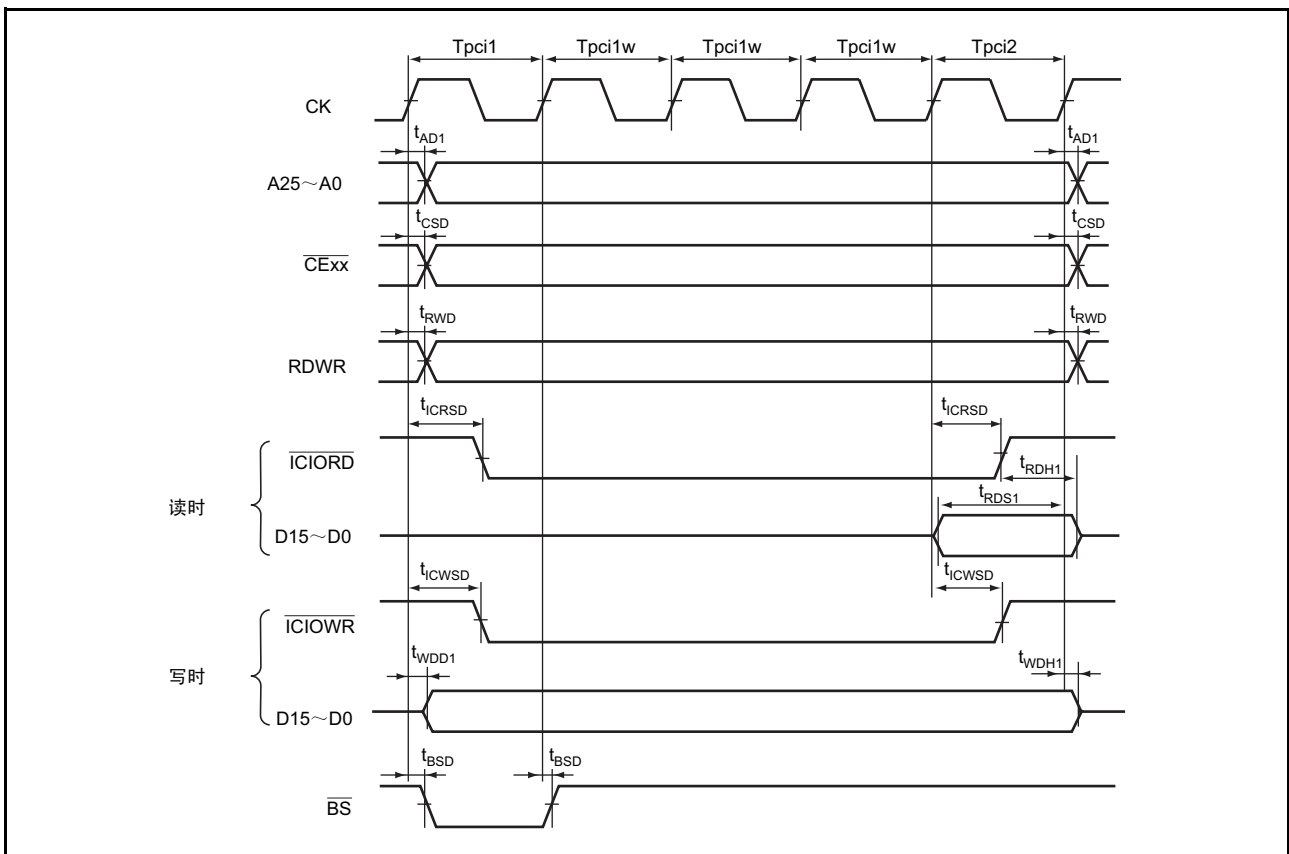


图 28.43 PCMCIA I/O 卡接口的总线时序

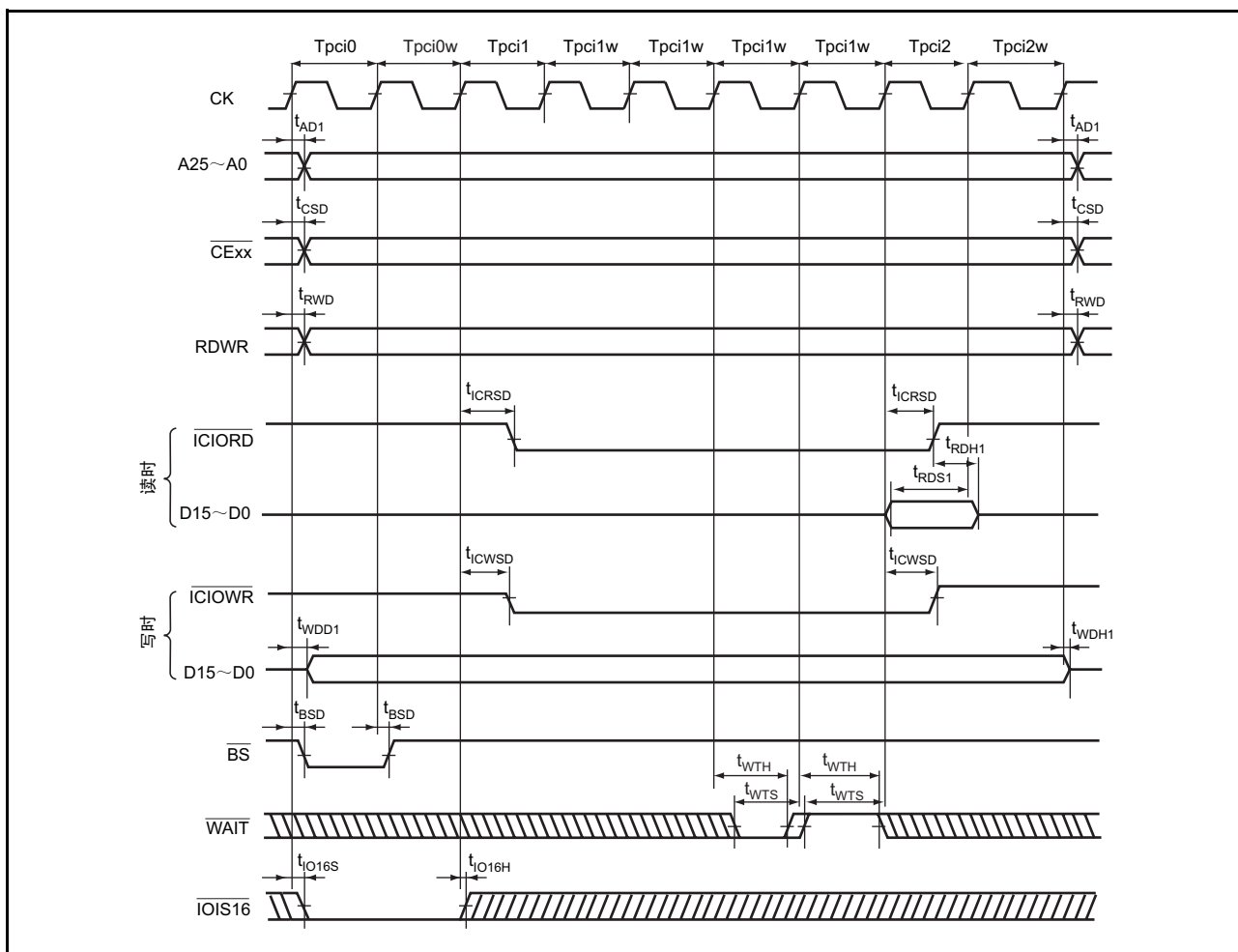


图 28.44 PCMCIA I/O 卡接口的总线时序  
(TED=2.5 周期、TEH=1.5 个周期、1 个外部等待)

### 28.3.4 直接存储器存取控制器 (DMAC) 的时序

表 28.9 直接存储器存取控制器 (DMAC) 的时序

条件:  $V_{CC}=3.0V \sim 3.6V$  或者  $4.5V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{ref}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{SS}=AV_{SS}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$  (民用产品)、 $T_a=-40 \sim +85^{\circ}C$  (工业用产品)

项 目	符号	Min.	Max.	单位	参照图
DREQ 准备时间	$t_{DRQS}$	20	—	ns	图 28.45
DREQ 保持时间	$t_{DRQH}$	20	—	ns	

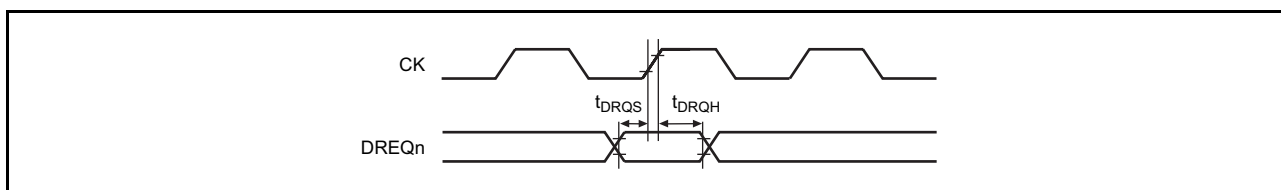


图 28.45 DREQ 的输入时序

## 28.3.5 多功能定时器脉冲单元 2 (MTU2) 的时序

表 28.10 多功能定时器脉冲单元 2 (MTU2) 的时序

条件:  $V_{CC}=3.0V \sim 3.6V$  或者  $4.5V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{ref}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$  (民用产品)、 $T_a=-40 \sim +85^{\circ}C$  (工业用产品)

项 目	符号	Min.	Max.	单位	参照图
输出比较的输出延迟时间	$t_{TOCD}$	—	50	ns	图 28.46
输入捕捉的输入准备时间	$t_{TICS}$	20	—	ns	
输入捕捉的输入脉宽 (指定单边沿时)	$t_{TICW}$	1.5	—	$t_{MPcyc}$	
输入捕捉的输入脉宽 (指定双边沿时)	$t_{TICW}$	2.5	—	$t_{MPcyc}$	
定时器的输入准备时间	$t_{TCKS}$	20	—	ns	图 28.47
定时器的时钟脉宽 (指定单边沿时)	$t_{TCKWH/L}$	1.5	—	$t_{MPcyc}$	
定时器的时钟脉宽 (指定双边沿时)	$t_{TCKWH/L}$	2.5	—	$t_{MPcyc}$	
定时器的时钟脉宽 (位相计数模式)	$t_{TCKWH/L}$	2.5	—	$t_{MPcyc}$	

【注】  $t_{MPcyc}$  表示 MTU2 时钟 (MP $\phi$ ) 的周期。

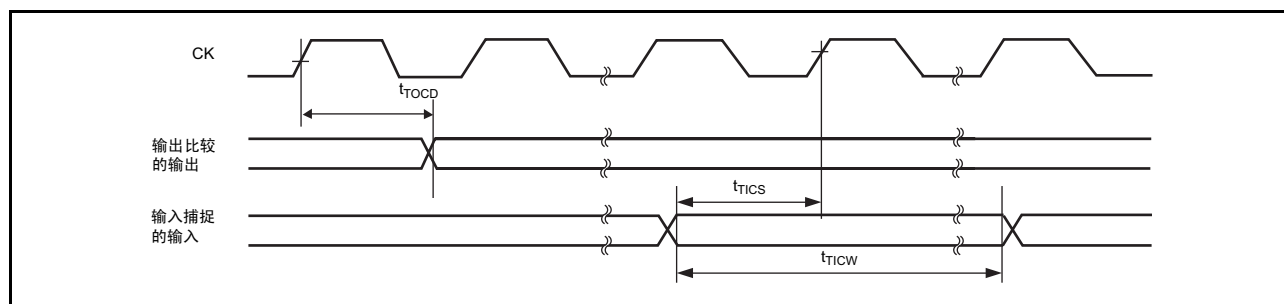


图 28.46 MTU2 的输入 / 输出时序

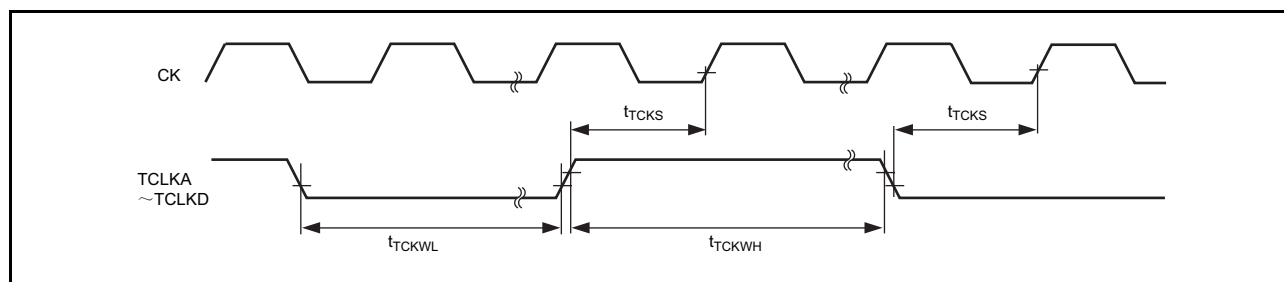


图 28.47 MTU2 的时钟输入时序

## 28.3.6 多功能定时器脉冲单元 2S (MTU2S) 的时序

表 28.11 多功能定时器脉冲单元 2S (MTU2S) 的时序

条件:  $V_{CC}=3.0V \sim 3.6V$  或者  $4.5V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{ref}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$  (民用产品)、 $T_a=-40 \sim +85^{\circ}C$  (工业用产品)

项 目	符号	Min.	Max.	单位	参照图
输出比较的输出延迟时间	$t_{TOCD}$	—	50	ns	图 28.48
输入捕捉的输入准备时间	$t_{TICS}$	20	—	ns	
输入捕捉的输入脉宽 (指定单边沿时)	$t_{TICW}$	1.5	—	$t_{M1cyc}$	
输入捕捉的输入脉宽 (指定双边沿时)	$t_{TICW}$	2.5	—	$t_{M1cyc}$	

【注】  $t_{M1cyc}$  表示 MTU2S 时钟 ( $M1\phi$ ) 的周期。

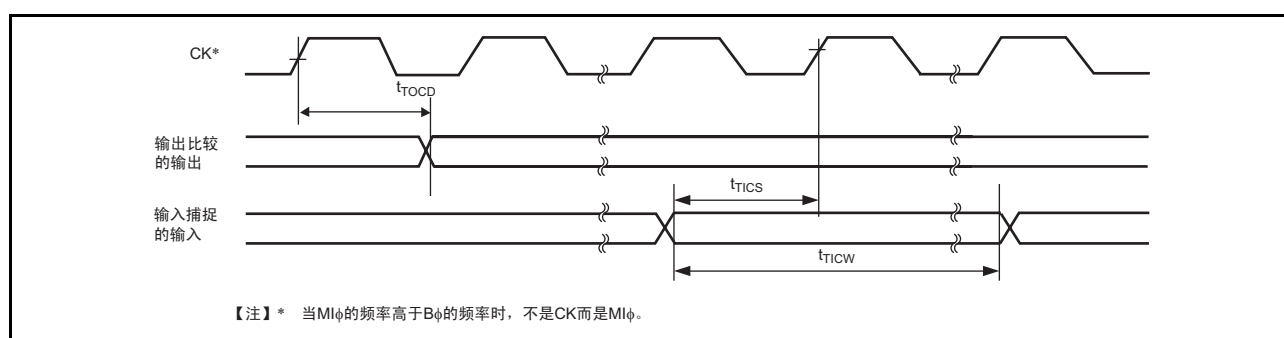


图 28.48 MTU2S 的输入 / 输出时序

## 28.3.7 I/O 端口的时序

表 28.12 I/O 端口的时序

条件:  $V_{CC}=3.0V \sim 3.6V$  或者  $4.5V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{ref}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$  (民用产品)、 $T_a=-40 \sim +85^{\circ}C$  (工业用产品)

项 目	符号	Min.	Max.	单位	参照图
端口输出数据的延迟时间	$t_{PWD}$	—	50	ns	图 28.49
端口输入的保持时间	$t_{PRH}$	20	—	ns	
端口输入的准备时间	$t_{PRS}$	20	—	ns	

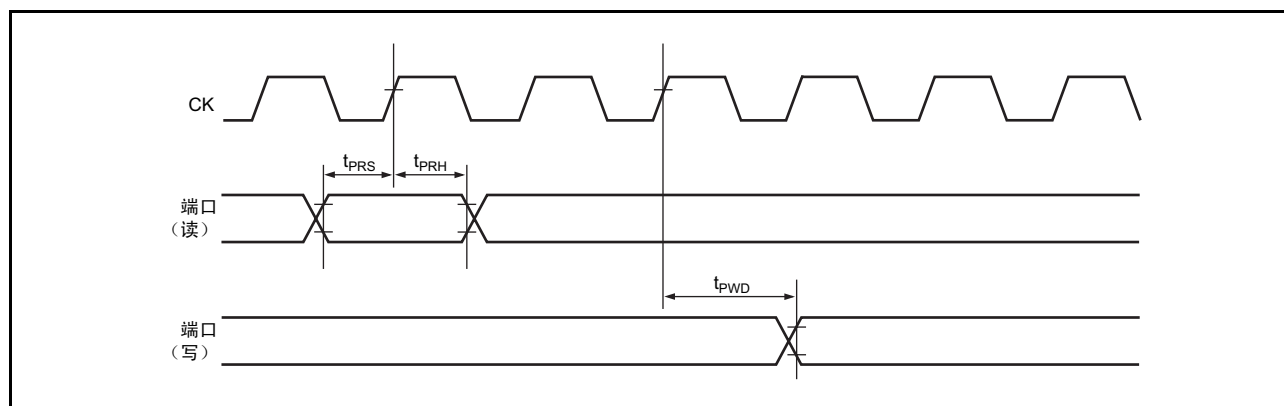


图 28.49 I/O 端口的输入 / 输出时序

## 28.3.8 看门狗定时器 (WDT) 的时序

表 28.13 看门狗定时器的 (WDT) 时序

条件:  $V_{CC}=3.0V \sim 3.6V$  或者  $4.5V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{ref}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$  (民用产品)、 $T_a=-40 \sim +85^{\circ}C$  (工业用产品)

项 目	符号	Min.	Max.	单位	参照图
WDTOVF 延迟时间	$t_{WDOVD}$	—	50	ns	图 28.50

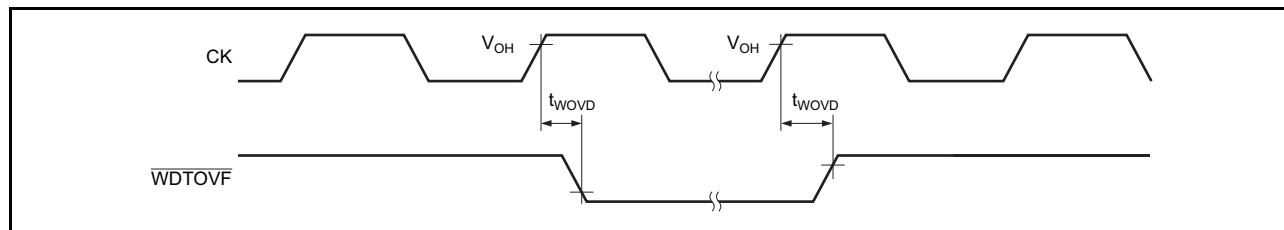


图 28.50 看门狗定时器的时序

## 28.3.9 串行通信接口 (SCI) 的时序

表 28.14 串行通信接口 (SCI) 的时序

条件:  $V_{CC}=3.0V \sim 3.6V$  或者  $4.5V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{ref}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$  (民用产品)、 $T_a=-40 \sim +85^{\circ}C$  (工业用产品)

项 目	符号	Min.	Max.	单位	参照图				
输入时钟周期 (异步)	$t_{scyc}$	4	—	$t_{pcyc}$	图 28.51				
输入时钟周期 (时钟同步)	$t_{scyc}$	6	—	$t_{pcyc}$					
输入时钟脉宽	$t_{sckw}$	0.4	0.6	$t_{scyc}$					
输入时钟的上升时间	$t_{sckr}$	—	1.5	$t_{pcyc}$					
输入时钟的下降时间	$t_{sckf}$	—	1.5	$t_{pcyc}$					
发送数据的延迟时间	$t_{TXD}$	—	$4t_{pcyc}+10$	ns	图 28.52				
接收数据的准备时间						$t_{RXS}$	$4t_{pcyc}$	—	ns
接收数据的保持时间						$t_{RXH}$	$4t_{pcyc}$	—	ns
发送数据的延迟时间	$t_{TXD}$	—	$3t_{pcyc}+10$	ns					
接收数据的准备时间						$t_{RXS}$	$2t_{pcyc}+50$	—	ns
接收数据的保持时间						$t_{RXH}$	$2t_{pcyc}$	—	ns

【注】  $t_{pcyc}$  表示外围时钟 ( $P_{\phi}$ ) 的周期。

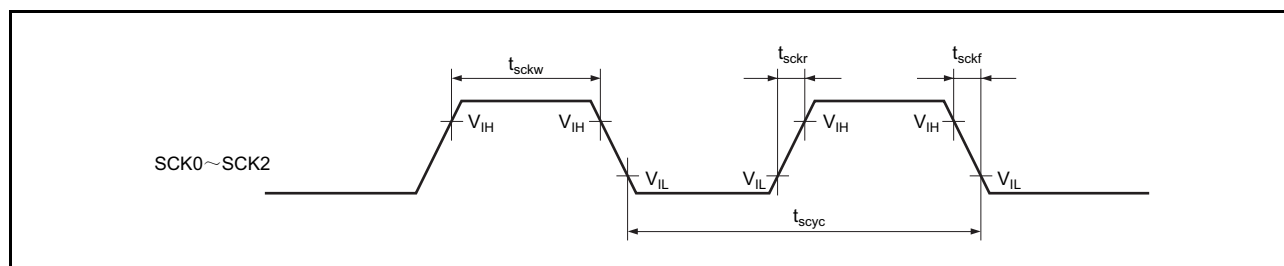


图 28.51 输入时钟的时序

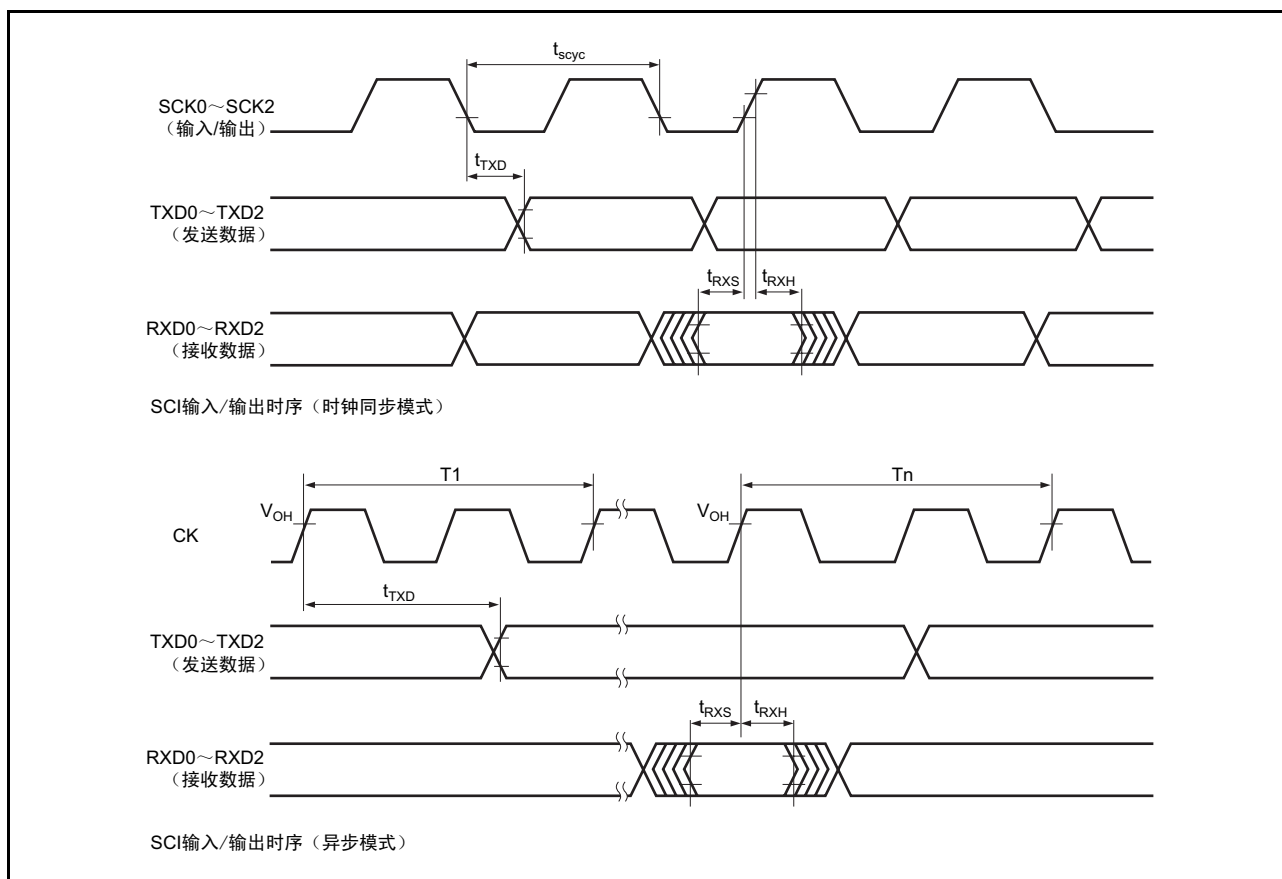


图 28.52 SCI 的输入 / 输出时序

## 28.3.10 带 FIFO 的串行通信接口 (SCIF) 的时序

表 28.15 带 FIFO 的串行通信接口 (SCIF) 的时序

条件:  $V_{CC}=3.0V \sim 3.6V$  或者  $4.5V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{ref}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$  (民用产品)、 $T_a=-40 \sim +85^{\circ}C$  (工业用产品)

项 目	符号	Min.	Max.	单位	参照图				
输入时钟周期 (异步)	$t_{scyc}$	4	—	$t_{pcyc}$	图 28.53				
输入时钟周期 (时钟同步)	$t_{scyc}$	6	—	$t_{pcyc}$					
输入时钟脉宽	$t_{sckw}$	0.4	0.6	tscyc					
输入时钟的上升时间	$t_{sckr}$	—	1.5	$t_{pcyc}$					
输入时钟的下降时间	$t_{sckf}$	—	1.5	$t_{pcyc}$					
发送数据的延迟时间	$t_{TXD}$	—	$4t_{pcyc}+10$	ns	图 28.54				
接收数据的准备时间						$t_{RXS}$	$4t_{pcyc}$	—	ns
接收数据的保持时间						$t_{RXH}$	$4t_{pcyc}$	—	ns
发送数据的延迟时间	$t_{TXD}$	—	$3t_{pcyc}+10$	ns					
接收数据的准备时间						$t_{RXS}$	$2t_{pcyc}+50$	—	ns
接收数据的保持时间						$t_{RXH}$	$2t_{pcyc}$	—	ns
$\overline{RTS}$ 延迟时间	$t_{RTSD}$	—	$4t_{pcyc}+10$	ns					
$\overline{CTS}$ 准备时间					$t_{CTSS}$	$4t_{pcyc}$	—	ns	
$\overline{CTS}$ 保持时间					$t_{CTSH}$	$4t_{pcyc}$	—	ns	

【注】  $t_{pcyc}$  表示外围时钟 ( $P_{\phi}$ ) 的周期。

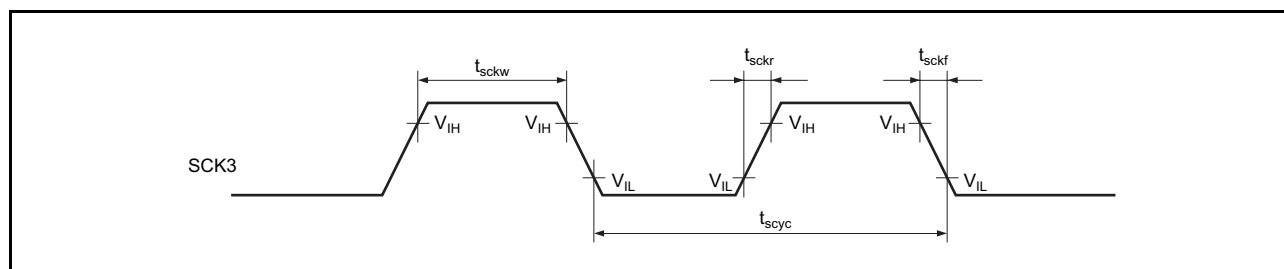


图 28.53 输入时钟的时序



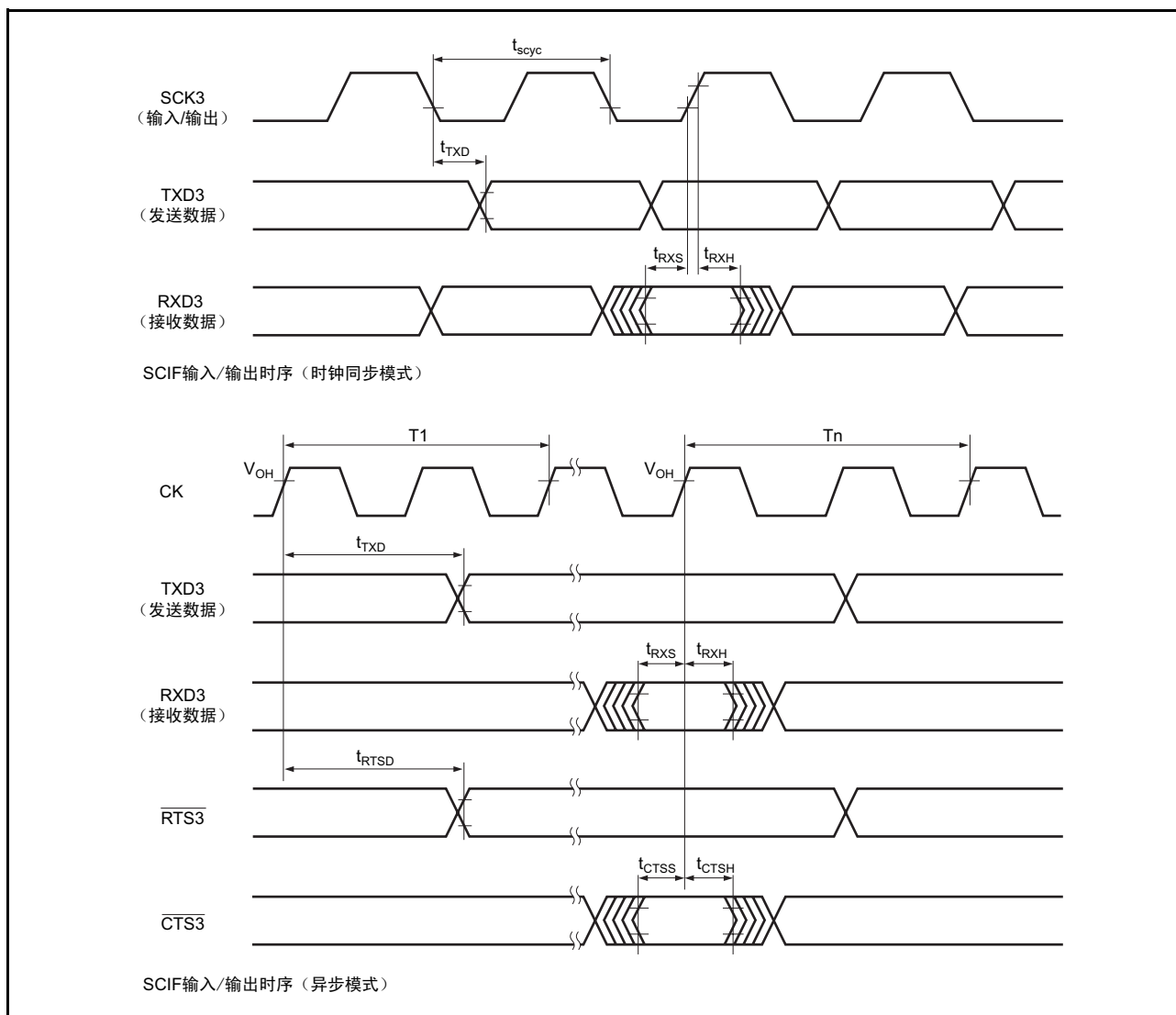


图 28.54 SCIF 的输入 / 输出时序

## 28.3.11 串行通信单元 (SSU) 的时序

表 28.16 串行通信单元 (SSU) 的时序

条件:  $V_{CC}=3.0V \sim 3.6V$  或者  $4.5V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{ref}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$  (民用产品)、 $T_a=-40 \sim +85^{\circ}C$  (工业用产品)

项 目		符号	Min.	Max.	单位	测量条件
时钟周期	主器件	$t_{SUcyc}$	4	256	$t_{pcyc}$	图 28.55
	从属器件		4	256		图 28.56
时钟的高电平脉宽	主器件	$t_{HI}$	60	—	ns	图 28.57
	从属器件		60	—		图 28.58
时钟的低电平脉宽	主器件	$t_{LO}$	60	—	ns	
	从属器件		60	—		
时钟的上升时间		$t_{RISE}$	—	20	ns	
时钟的下降时间		$t_{FALL}$	—	20	ns	
数据输入的准备时间	主器件	$t_{SU}$	25	—	ns	
	从属器件		30	—		
数据输入的保持时间	主器件	$t_H$	10	—	ns	
	从属器件		10	—		
$\overline{SCS}$ 准备时间	主器件	$t_{LEAD}$	1.5	—	$t_{pcyc}$	
	从属器件		1.5	—		
$\overline{SCS}$ 保持时间	主器件	$t_{LAG}$	1.5	—	$t_{pcyc}$	
	从属器件		1.5	—		
数据输出的延迟时间	主器件	$t_{OD}$	—	40	ns	
	从属器件		—	40		
数据输出的保持时间	主器件	$t_{OH}$	30	—	ns	
	从属器件		30	—		
连续发送的延迟时间	主器件	$t_{TD}$	1.5	—	$t_{pcyc}$	
	从属器件		1.5	—		
从属器件的存取时间		$t_{SA}$	—	1	$t_{pcyc}$	图 28.57
从属器件的输出释放时间		$t_{REL}$	—	1	$t_{pcyc}$	图 28.58

【注】  $t_{pcyc}$  表示外围时钟 ( $P_{\phi}$ ) 的周期。

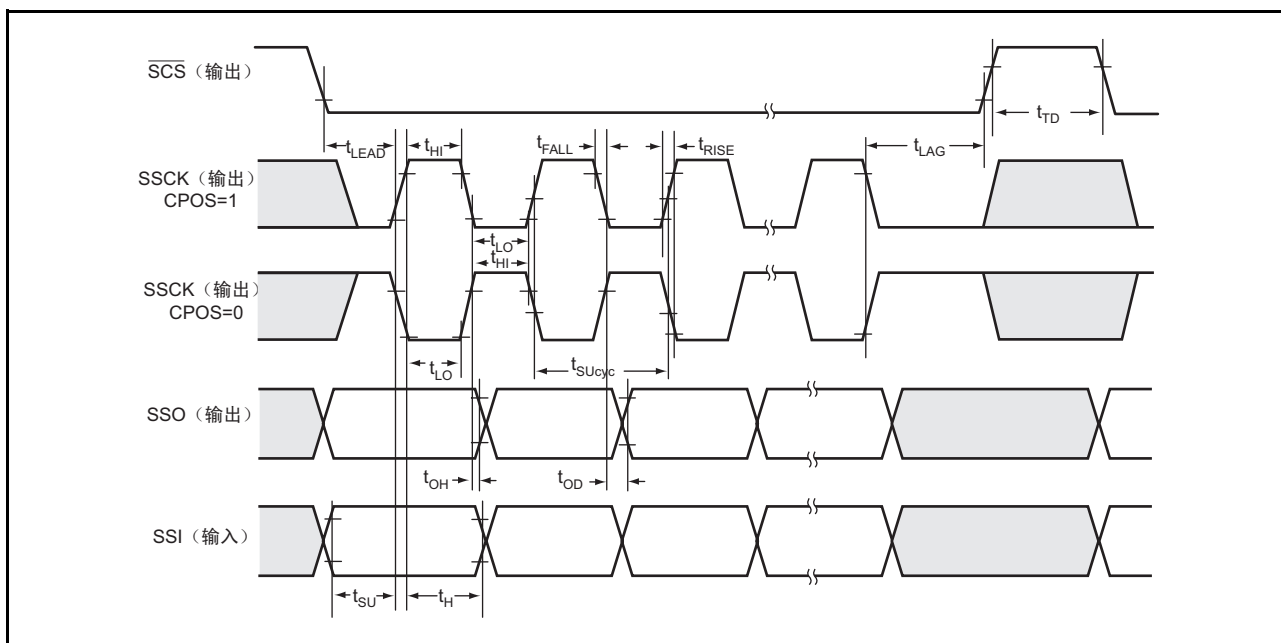


图 28.55 SSU 的时序 (主器件、CPHS=1)

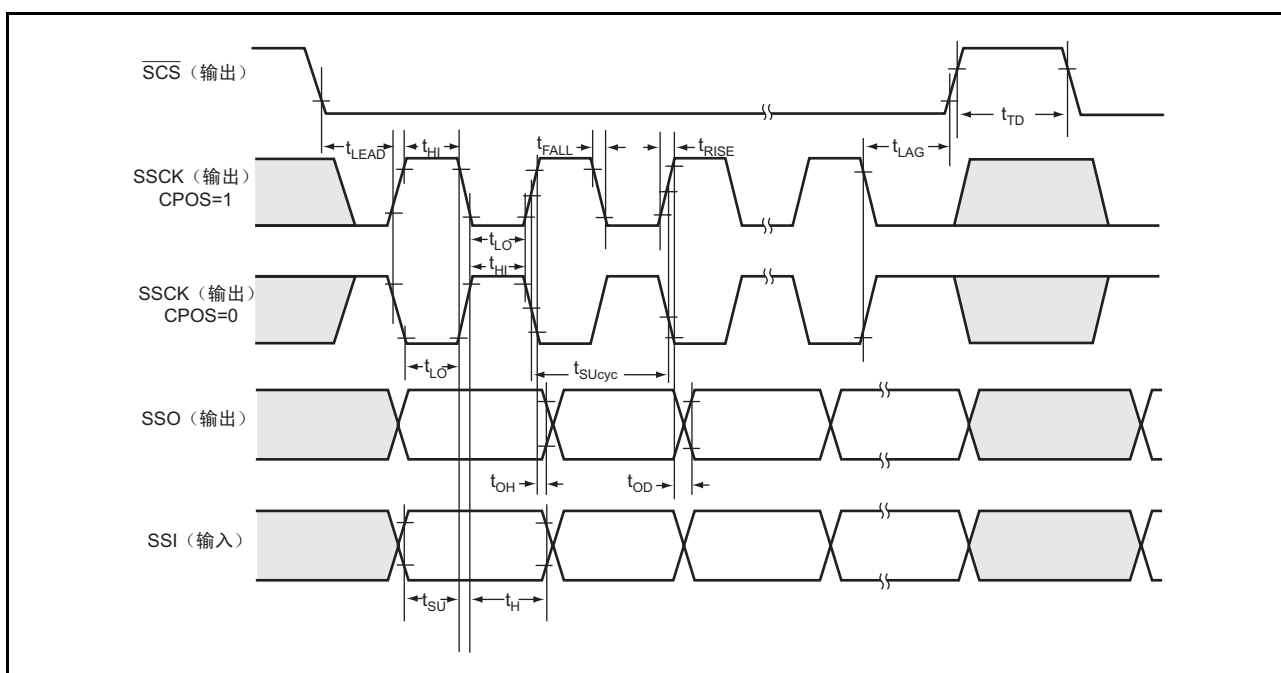


图 28.56 SSU 的时序 (主器件、CPHS=0)

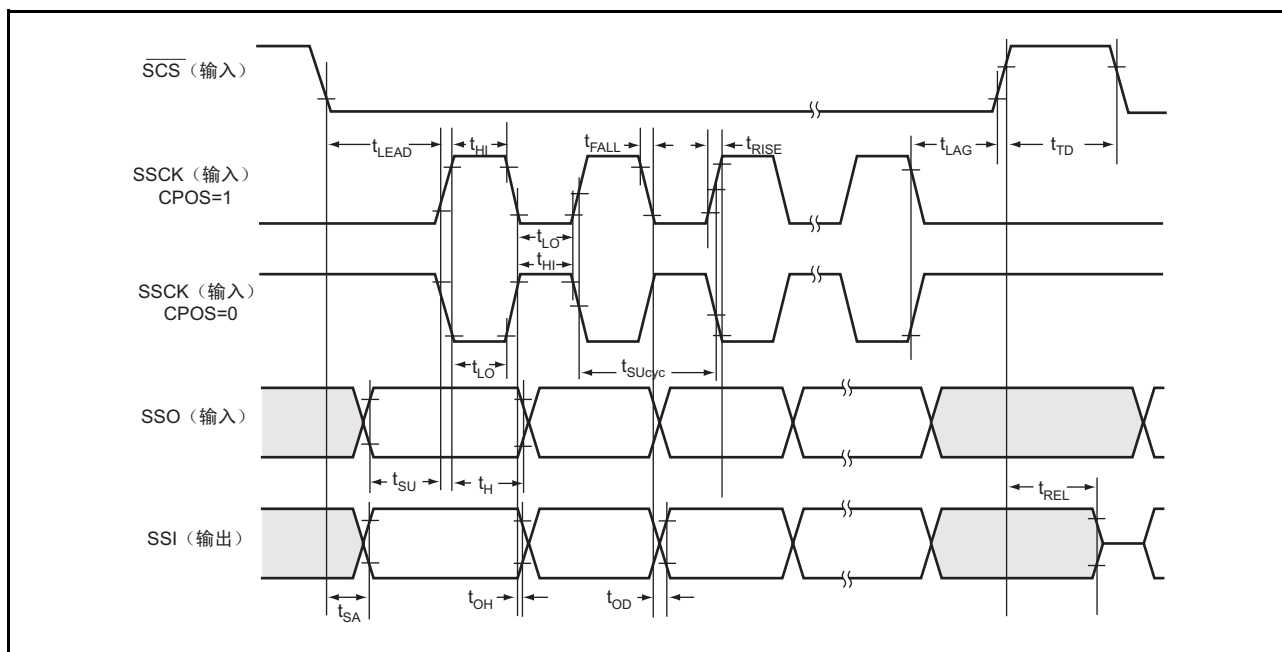


图 28.57 SSU 的时序（从属器件、CPHS=1）

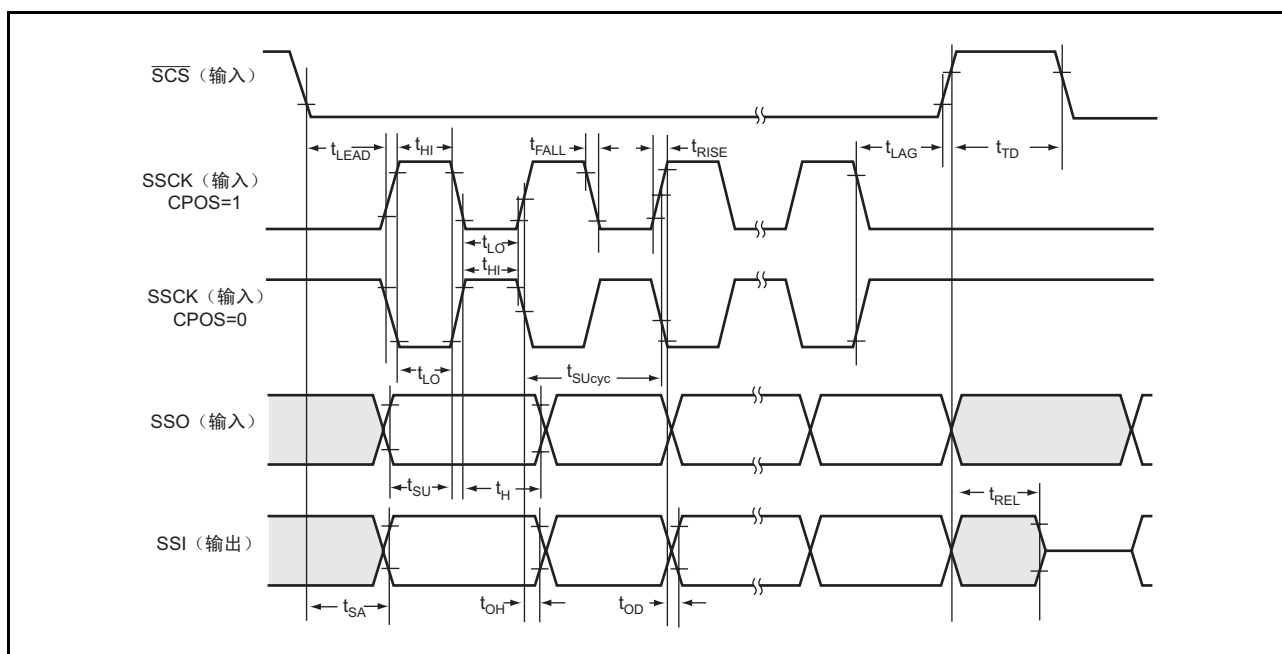


图 28.58 SSU 的时序（从属器件、CPHS=0）

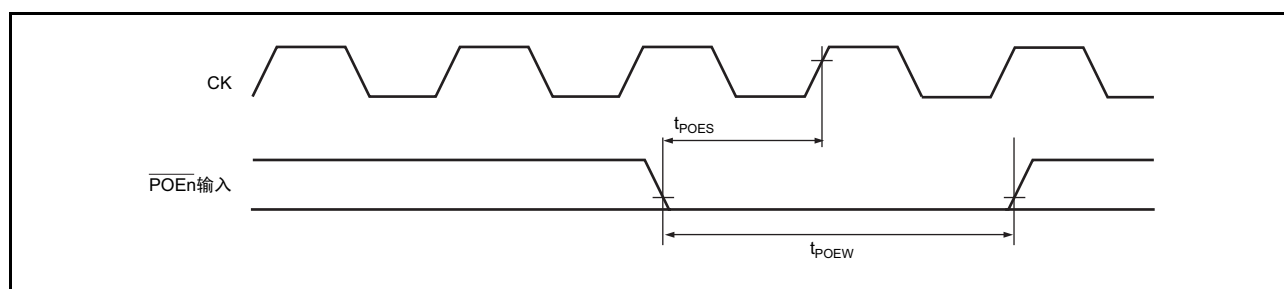
## 28.3.12 端口输出允许 (POE) 的时序

表 28.17 端口输出允许 (POE) 的时序

条件:  $V_{CC}=3.0V \sim 3.6V$  或者  $4.5V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{ref}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$  (民用产品)、 $T_a=-40 \sim +85^{\circ}C$  (工业用产品)

项 目	符号	Min.	Max.	单位	参照图
$\overline{POE}$ 输入的准备时间	$t_{POES}$	50	—	ns	图 28.59
$\overline{POE}$ 输入脉宽	$t_{POEW}$	1.5	—	$t_{pcyc}$	

【注】  $t_{pcyc}$  表示外围时钟 (P $\phi$ ) 的周期。

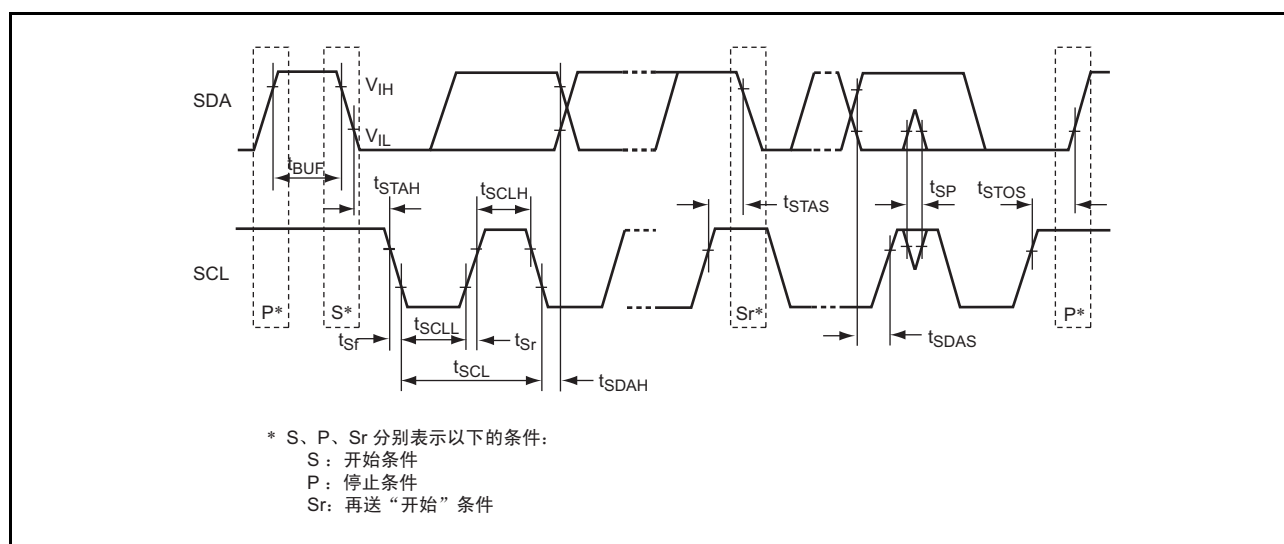
图 28.59  $\overline{POE}$  输入的时序

28.3.13 I<sup>2</sup>C 总线接口 2 (I<sup>2</sup>C2) 的时序表 28.18 I<sup>2</sup>C 总线接口 2 (I<sup>2</sup>C2) 的时序

条件:  $V_{CC}=3.0V \sim 3.6V$  或者  $4.5V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{ref}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$  (民用产品)、 $T_a=-40 \sim +85^{\circ}C$  (工业用产品)

项 目	符号	Min.	Typ.	Max.	单位	参照图
SCL 输入周期时间	$t_{SCL}$	$12t_{pcyc}+600$	—	—	ns	图 28.60
SCL 输入 High 脉宽	$t_{SCLH}$	$3t_{pcyc}+300$	—	—	ns	
SCL 输入 Low 脉宽	$t_{SCLL}$	$5t_{pcyc}+300$	—	—	ns	
SCL、SDA 输入的下陷时间	$t_{sf}$	—	—	300	ns	
SCL、SDA 输入的尖峰脉冲消除时间	$t_{SP}$	—	—	$1t_{pcyc}$	ns	
SDA 输入的总线自由时间	$t_{BUF}$	5	—	—	$t_{pcyc}$	
开始条件输入的保持时间	$t_{STAH}$	3	—	—	$t_{pcyc}$	
再送开始条件输入的准备时间	$t_{STAS}$	3	—	—	$t_{pcyc}$	
停止条件输入的准备时间	$t_{STOS}$	3	—	—	$t_{pcyc}$	
数据输入的准备时间	$t_{SDAS}$	$1t_{pcyc}+20$	—	—	ns	
数据输入的保持时间	$t_{SDAH}$	0	—	—	ns	
SCL、SDA 的电容负载	$C_b$	0	—	400	pF	
SCL、SDA 输出的下陷时间	$t_{sf}$	—	—	250	ns	

【注】  $t_{pcyc}$  表示外围时钟 ( $P\phi$ ) 的周期。

图 28.60 I<sup>2</sup>C 总线接口 2 的输入 / 输出时序

## 28.3.14 UBC 的触发时序

表 28.19 UBC 的触发时序

条件:  $V_{CC}=3.0V \sim 3.6V$  或者  $4.5V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{ref}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{SS}=AV_{SS}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$  (民用产品)、 $T_a=-40 \sim +85^{\circ}C$  (工业用产品)

项 目	符号	Min.	Max.	单位	参照图
$\overline{UBCTRG}$ 延迟时间	$t_{UBCTGD}$	—	150	ns	图 28.61

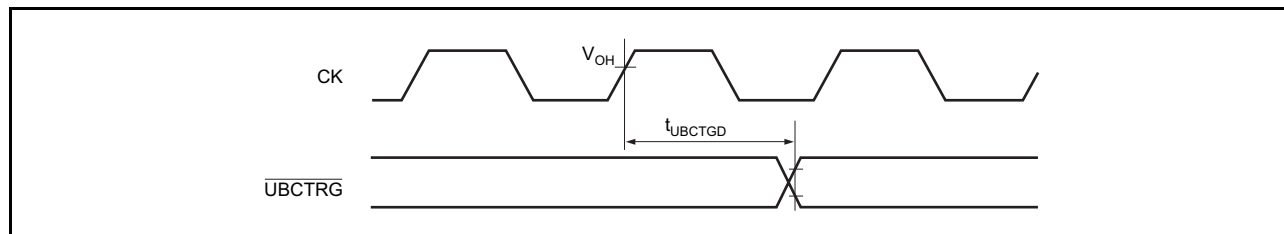


图 28.61 UBC 的触发时序

## 28.3.15 A/D 转换器的时序

表 28.20 A/D 转换器的时序

条件:  $V_{CC}=3.0V \sim 3.6V$  或者  $4.5V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{ref}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{SS}=AV_{SS}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$  (民用产品)、 $T_a=-40 \sim +85^{\circ}C$  (工业用产品)

项 目	符号	Min.	Typ.	Max.	单位	参照图
外部触发输入的开始延迟时间	$t_{TRGS}$	25	—	—	ns	图 28.62

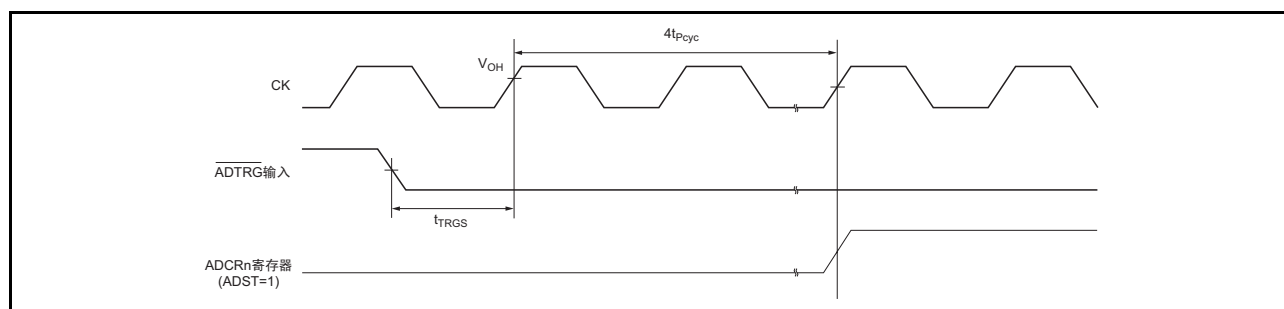


图 28.62 外部触发的输入时序

## 28.3.16 AC 特性的测量条件

- 输入信号电平:  $V_{IL}(\text{Max.})/V_{IH}(\text{Min.})$
- 输出信号参照电平: 高电平: 2.0V、低电平: 0.8V

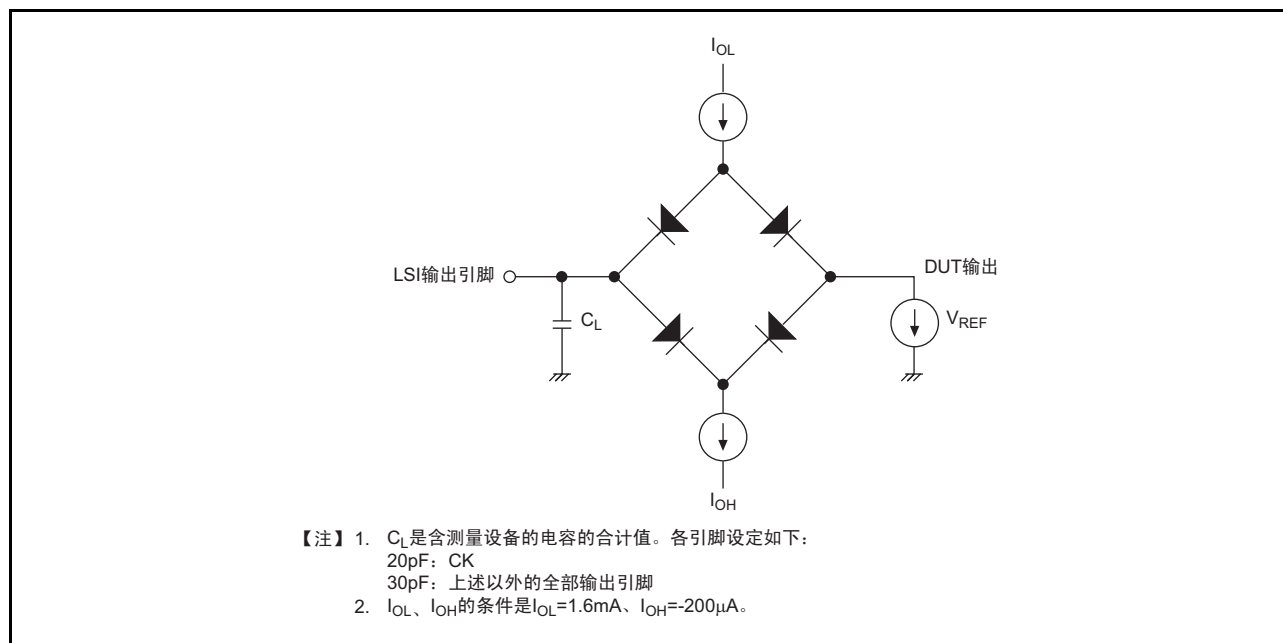


图 28.63 输出负载电路

## 28.4 A/D 转换器特性

表 28.21 A/D 转换器特性

条件:  $V_{CC}=3.0\text{V} \sim 3.6\text{V}$  或者  $4.5\text{V} \sim 5.5\text{V}$ 、 $AV_{CC}=4.5\text{V} \sim 5.5\text{V}$ 、 $AV_{ref}=4.5\text{V} \sim AV_{CC}$ 、 $V_{SS}=\text{PLL}V_{SS}=\text{AV}_{SS}=0\text{V}$ 、 $T_a=-20 \sim +85^\circ\text{C}$  (民用产品)、 $T_a=-40 \sim +85^\circ\text{C}$  (工业用产品)

项 目	Min.	Typ.	Max.	单位
分辨率	10	10	10	位
转换时间	2.0	—	—	$\mu\text{s}$
模拟输入电容	—	—	20	pF
允许信号源阻抗	—	—	$1*1/3*2$	$\text{k}\Omega$
非线性误差	—	—	$\pm 3.0*1/\pm 5.0*2$	LSB
偏移误差	—	—	$\pm 3.0*1/\pm 5.0*2$	LSB
满刻度误差	—	—	$\pm 3.0*1/\pm 5.0*2$	LSB
量化误差	—	—	$\pm 0.5$	LSB
绝对精度	—	—	$\pm 4.0*1/\pm 6.0*2$	LSB

【注】 \*1 转换时间  $\geq 4.0\mu\text{s}$  时

\*2 转换时间  $< 4.0\mu\text{s}$  时



## 28.5 闪存特性

表 28.22 闪存特性

条件:  $V_{CC}=3.0V \sim 3.6V$  或者  $4.5V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{ref}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$  (民用产品)、 $T_a=-40 \sim +85^{\circ}C$  (工业用产品)

项 目	符号	Min.	Typ.	Max.	单位
编程时间 *1*2*4	$t_P$	—	1	10	ms/128 字节
擦除时间 *1*2*4	$t_E$	—	30	100	ms/4K 字节块
		—	250	800	ms/32K 字节块
		—	500	1600	ms/64K 字节块
编程时间 (总和) *1*2*4	$\Sigma t_P$	—	5	14	s/512K 字节
		—	2.5	7	s/256K 字节
擦除时间 (总和) *1*2*4	$\Sigma t_E$	—	5	14	s/512K 字节
		—	2.5	7	s/256K 字节
编程、擦除时间 (总和) *1*2*4	$\Sigma t_{PE}$	—	10	28	s/512K 字节
		—	5	14	s/256K 字节
改写次数	NWEC	100*3	—	—	次

【注】 \*1 编程和擦除时间取决于数据。

\*2 编程和擦除时间不包含数据的传送时间。

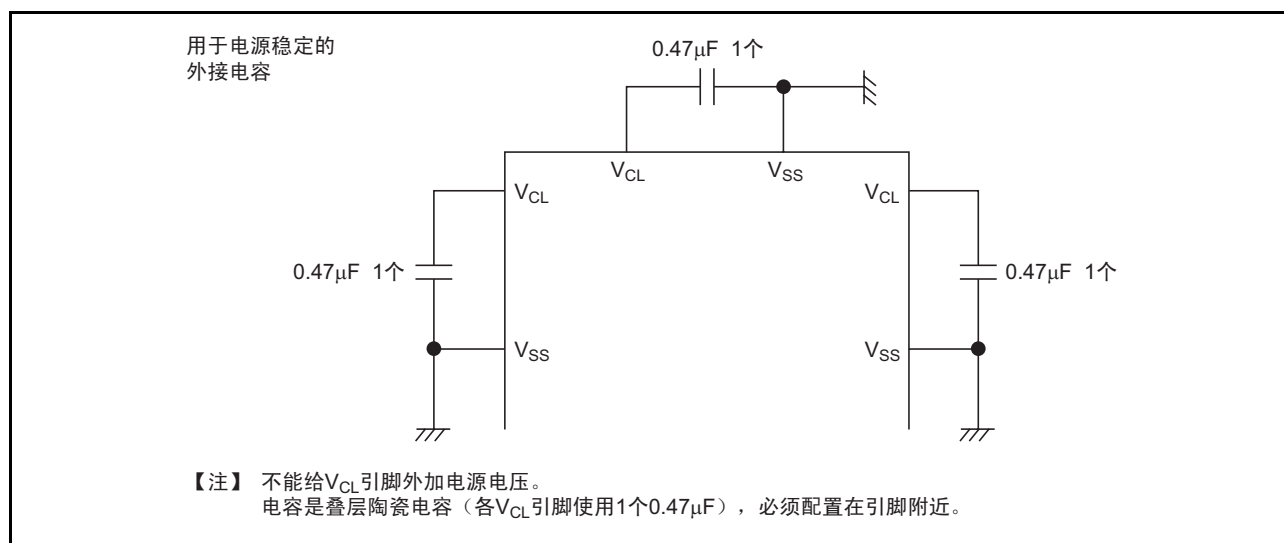
\*3 是保证改写后所有特性的 Min. 次数 (保证范围: 1 ~ Min. 值)。

\*4 是在包含 Min. 值的使用范围内进行改写时的特性。

## 28.6 使用注意事项

28.6.1 V<sub>CL</sub> 电容连接方法

本 LSI 内置内部降压电路, 将单片机内部的电源电压自动下降到适合的电平。需要在此内部降压电源 (V<sub>CL</sub> 引脚) 和 V<sub>SS</sub> 引脚之间连接用于稳定内部电压的电容 (0.47 $\mu$ F)。外接电容的连接方法如图 28.64 所示。外接电容必须配置在引脚的附近, 不能给 V<sub>CL</sub> 引脚外加电源电压。

图 28.64 V<sub>CL</sub> 电容连接方法

## 附录

## 附录 A. 引脚状态

引脚的初始值因 MCU 运行模式而不同，详细内容请参照“第 21 章 引脚功能控制器（PFC）”。

表 A.1 引脚状态（SH7083）

引脚功能		引脚状态										
分类	引脚名	复位状态				低功耗状态			总线权 释放 状态	检测到 振荡 停止时	使用 POE 功能时	
		上电			手动	深度 软件 待机	软件待机	睡眠				
		无 ROM 扩展		有 ROM 扩展								单 芯 片
		8bit	16bit									
时钟	CK	O		Z	O	Z	H* <sup>1</sup>	O	O	O	O	
	XTAL	O			O	L	L	O	O	O	O	
	EXTAL	I			I	Z	I	I	I	I	I	
系统 控制	$\overline{\text{RES}}$	I			I	I	I	I	I	I	I	
	$\overline{\text{MRES}}$	Z			I	Z	I* <sup>7</sup>	I	I	I* <sup>7</sup>	I	
	$\overline{\text{WDTOVF}}$	O* <sup>8</sup>			O	O	O	O	O	O	O	
	$\overline{\text{BREQ}}$	Z			I	Z	Z	I	I	I	I	
	$\overline{\text{BACK}}$	Z			O	Z	Z	O	L	O	O	
运行模 式控制	MD0、MD1	I			I	I	I	I	I	I	I	
	$\overline{\text{ASEMD0}}$	I* <sup>9</sup>			I* <sup>9</sup>	I* <sup>9</sup>	I* <sup>9</sup>	I* <sup>9</sup>	I* <sup>9</sup>	I* <sup>9</sup>	I* <sup>9</sup>	
	FWE	I			I	I	I	I	I	I	I	
中断	NMI	I			I	I	I	I	I	I	I	
	IRQ0 ~ IRQ7	Z			I	Z	I	I	I	I	I	
	$\overline{\text{IRQOUT}}$	Z			O	Z	Z (HCPCR 的 MZIZEL=0) H* <sup>1</sup> (HCPCR 的 MZIZEL=1)	O	O	O* <sup>7</sup>	O	
地址 总线	A0 ~ A17	O	Z		O	Z	Z* <sup>3</sup>	O	Z	O	O	
	A18 ~ A24	Z			O	Z	Z* <sup>3</sup>	O	Z	O	O	
数据 总线	D0 ~ D8、D10	Z			I/O	Z	Z	I/O	Z	I/O	I/O	
	D9、D11 ~ D15	Z			I/O	Z	Z	I/O	Z	I/O* <sup>5</sup>	I/O	
总线 控制	$\overline{\text{WAIT}}$	Z			I	Z	Z	I	Z	I	I	
	$\overline{\text{CS0}}$	H	Z		O	Z	Z* <sup>3</sup>	O	Z	O	O	
	$\overline{\text{CS3}}$ 、 $\overline{\text{CS7}}$	Z			O	Z	Z* <sup>3</sup>	O	Z	O	O	
	$\overline{\text{BS}}$	Z			O	Z	Z* <sup>3</sup>	O	Z	O	O	
	$\overline{\text{RASL}}$	Z			O	Z	Z* <sup>2</sup>	O	Z* <sup>2</sup>	O	O	

引脚功能		引脚状态										
分类	引脚名	复位状态				低功耗状态			总线权 释放 状态	检测到 振荡 停止时	使用 POE 功能时	
		上电			手动	深度 软件 待机	软件待机	睡眠				
		无 ROM 扩展		有 ROM 扩展								单 芯片
		8bit	16bit									
总线 控制	CASL	Z				O	Z	Z*2	O	Z*2	O	O
	DQMLU、DQMLL	Z				O	Z	Z*3	O	Z	O	O
	RDWR	Z				O	Z	Z*3	O	Z	O	O
	$\overline{\text{RD}}$	H	Z			O	Z	Z*3	O	Z	O	O
	$\overline{\text{WRH}}$ 、 $\overline{\text{WRL}}$	H	Z			O	Z	Z*3	O	Z	O	O
	CKE (PE15)	Z				O	Z	Z (HCPCR 的 MZIZEL=0)	O	Z*2	O*7	O
								Z*2 (HCPCR 的 MZIZEL=1)				
CKE (PA9)	Z				O	Z	Z*2	O	Z*2	O	O	
DMAC	DREQ0、DREQ1	Z				I	Z	Z	I	I	I	I
	DACK0、DACK1	Z				O	Z	Z (HCPCR 的 MZIZEL=0)	O	O	O*7	O
								O*1 (HCPCR 的 MZIZEL=1)				
	TEND0、TEND1	Z				O	Z	O*1	O	O	O	O
MTU2	TCLKB ~ TCLKD	Z				I	Z	Z	I	I	I	I
	TIOC0A ~ TIOC0D	Z				I/O	Z	K*1	I/O	I/O	I/O	Z
	TIOC1A	Z				I/O	Z	K*1	I/O	I/O	I/O	I/O
	TIOC2A、TIOC2B	Z				I/O	Z	K*1	I/O	I/O	I/O	I/O
	TIOC3A、TIOC3C	Z				I/O	Z	K*1	I/O	I/O	I/O	I/O
	TIOC4A ~ TIOC4D	Z				I/O	Z	Z (HCPCR 的 MZIZEL=0)	I/O	I/O	I/O*7	Z
								K*1 (HCPCR 的 MZIZEL=1)				
TIC5U、TIC5V、 TIC5W	Z				I	Z	Z	I	I	I	I	
MTU2S	TIOC3AS、 TIOC3CS	Z				I/O	Z	K*1	I/O	I/O	I/O	I/O

引脚功能		引脚状态										
分类	引脚名	复位状态				低功耗状态			总线权 释放 状态	检测到 振荡 停止时	使用 POE 功能时	
		上电			手动	深度 软件 待机	软件待机	睡眠				
		无 ROM 扩展		有 ROM 扩展								单 芯片
		8bit	16bit									
MTU2S	TIOC3BS、 TIOC3DS	Z			I/O	Z	Z (HCPCR 的 MZIZDL=0)	I/O	I/O	I/O*5	Z	
							K*1 (HCPCR 的 MZIZDL=1)					
	TIOC4AS ~ TIOC4DS	Z			I/O	Z	Z (HCPCR 的 MZIZDL=0)	I/O	I/O	I/O*5	Z	
							K*1 (HCPCR 的 MZIZDL=1)					
	TIC5US、 TIC5VS、 TIC5WS	Z			I	Z	Z	I	I	I	I	
POE	$\overline{\text{POE0}}$ 、 $\overline{\text{POE2}} \sim \overline{\text{POE4}}$ 、 $\overline{\text{POE6}} \sim \overline{\text{POE8}}$	Z			I	Z	Z	I	I	I	I	
SCI	SCK0 ~ SCK2	Z			I/O	Z	Z	I/O	I/O	I/O	I/O	
	RXD0 ~ RXD2	Z			I	Z	Z	I	I	I	I	
	TXD0 ~ TXD2	Z			O	Z	O*1	O	O	O	O	
SCIF	SCK3	Z			I/O	Z	Z	I/O	I/O	I/O	I/O	
	RXD3	Z			I	Z	Z	I	I	I	I	
	TXD3	Z			O	Z	Z (HCPCR 的 MZIZEL=0)	O	O	O*7	O	
							O*1 (HCPCR 的 MZIZEL=1)					
SSU	SSCK	Z			I/O	Z	Z	I/O	I/O	I/O	I/O	
	$\overline{\text{SCS}}$	Z			I/O	Z	Z	I/O	I/O	I/O*7	I/O	
	SSI	Z			I/O	Z	Z	I/O	I/O	I/O	I/O	
	SSO	Z			I/O	Z	Z	I/O	I/O	I/O	I/O	
UBC	$\overline{\text{UBCTRG}}$	Z			O	Z	O*1	O	O	O	O	
A/D 转换器	AN0 ~ AN7	Z			I	Z	Z	I	I	I	I	
	$\overline{\text{ADTRG}}$	Z			I	Z	Z	I	I	I	I	
I/O 端口	PA3 ~ PA5、 PA7 ~ PA10、 PA12 ~ PA15	Z			I/O	Z	K*1	I/O	I/O	I/O	I/O	

引脚功能		引脚状态											
分类	引脚名	复位状态				手动	低功耗状态			总线权 释放 状态	检测到 振荡 停止时	使用 POE 功能时	
		上电			无 ROM 扩展		有 ROM 扩展	单 芯片	深度 软件 待机				软件待机
		8bit	16bit										
I/O 端口	PB0 ~ PB2、 PB4 ~ PB9	Z				I/O	Z	K*1	I/O	I/O	I/O	I/O	
	PC0 ~ PC15	Z				I/O	Z	K*1	I/O	I/O	I/O	I/O	
	PD0 ~ PD8、 PD10	Z				I/O	Z	K*1	I/O	I/O	I/O	I/O	
	PD9、 PD11 ~ PD15	Z				I/O	Z	Z (HCPCR 的 MZIZDL=0)	I/O	I/O	I/O*5	Z	
								K*1 (HCPCR 的 MZIZDL=1)					
	PE0 ~ PE3	Z				I/O	Z	K*1	I/O	I/O	I/O	Z	
	PE4、 PE6 ~ PE8、 PE10	Z				I/O	Z	K*1	I/O	I/O	I/O	I/O	
	PE12 ~ PE15	Z				I/O	Z	Z (HCPCR 的 MZIZEL=0)	I/O	I/O	I/O*7	Z	
								K*1 (HCPCR 的 MZIZEL=1)					
PF0 ~ PF7	Z				I	Z	Z	I	I	I	I		

## 【符号说明】

- I : 输入  
O : 输出  
H : 高电平输出  
L : 低电平输出  
Z : 高阻抗  
K : 输入引脚为高阻抗，输出引脚为状态保持

【注】\*1 如果将待机控制寄存器 6 (STBCR6) 的 HIZ 位置为 1，输出引脚就为高阻抗。

\*2 如果将共用控制寄存器 (CMNCR) 的 HIZCNT 位置为 1，就为输出。

\*3 如果将共用控制寄存器 (CMNCR) 的 HIZMEM 位置为 1，就为输出。

\*4 如果将大电流端口控制寄存器 (HCPCR) 的 MZIZDH 位置为 0，就为高阻抗。

\*5 如果将大电流端口控制寄存器 (HCPCR) 的 MZIZDL 位置为 0，就为高阻抗。

\*6 如果将大电流端口控制寄存器 (HCPCR) 的 MZIZEH 位置为 0，就为高阻抗。

\*7 如果将大电流端口控制寄存器 (HCPCR) 的 MZIZEL 位置为 0，就为高阻抗。

\*8 在上电复位中为输入状态。为了防止误动作，必须进行上拉。需要进行下拉时，必须至少用 1MΩ 的电阻值进行下拉。

\*9 什么也不输入时，在内部进行上拉。

表 A.2 引脚状态 (SH7084)

引脚功能		引脚状态										
分类	引脚名	复位状态				低功耗状态			总线权 释放 状态	检测到 振荡 停止时	使用 POE 功能时	
		上电			手动	深度软 件待机	软件待机	睡眠				
		无 ROM 扩展		有 ROM 扩展								单 芯 片
		8bit	16bit									
时钟	CK	O		Z	O	Z	H* <sup>1</sup>	O	O	O	O	
	XTAL	O			O	L	L	O	O	O	O	
	EXTAL	I			I	Z	I	I	I	I	I	
系统 控制	$\overline{\text{RES}}$	I			I	I	I	I	I	I	I	
	$\overline{\text{MRES}}$	Z			I	Z	I* <sup>7</sup>	I	I	I* <sup>7</sup>	I	
	$\overline{\text{WDTOVF}}$	O* <sup>8</sup>			O	O	O	O	O	O	O	
	$\overline{\text{BREQ}}$	Z			I	Z	Z	I	I	I	I	
	$\overline{\text{BACK}}$	Z			O	Z	Z	O	L	O	O	
运行 模式 控制	MD0、MD1	I			I	I	I	I	I	I	I	
	$\overline{\text{ASEMD0}}$	I* <sup>9</sup>			I* <sup>9</sup>	I* <sup>9</sup>	I* <sup>9</sup>	I* <sup>9</sup>	I* <sup>9</sup>	I* <sup>9</sup>	I* <sup>9</sup>	
	FWE	I			I	I	I	I	I	I	I	
中断	NMI	I			I	I	I	I	I	I	I	
	IRQ0 ~ IRQ7	Z			I	Z	I	I	I	I	I	
	$\overline{\text{IRQOUT}}$	Z			O	Z	Z (HCPCR 的 MZIZEL=0) H* <sup>1</sup> (HCPCR 的 MZIZEL=1)	O	O	O* <sup>7</sup>	O	
地址 总线	A0 ~ A17	O	Z		O	Z	Z* <sup>3</sup>	O	Z	O	O	
	A18 ~ A25	Z			O	Z	Z* <sup>3</sup>	O	Z	O	O	
数据 总线	D0 ~ D8、D10	Z			I/O	Z	Z	I/O	Z	I/O	I/O	
	D9、D11 ~ D15	Z			I/O	Z	Z	I/O	Z	I/O* <sup>5</sup>	I/O	
总线 控制	$\overline{\text{WAIT}}$	Z			I	Z	Z	I	Z	I	I	
	$\overline{\text{CS0}}$ 、 $\overline{\text{CS1}}$	H	Z		O	Z	Z* <sup>3</sup>	O	Z	O	O	
	$\overline{\text{CS2}}$ ~ $\overline{\text{CS7}}$	Z			O	Z	Z* <sup>3</sup>	O	Z	O	O	
	$\overline{\text{BS}}$	Z			O	Z	Z* <sup>3</sup>	O	Z	O	O	
	$\overline{\text{RASL}}$	Z			O	Z	Z* <sup>2</sup>	O	Z* <sup>2</sup>	O	O	
	$\overline{\text{CASL}}$	Z			O	Z	Z* <sup>2</sup>	O	Z* <sup>2</sup>	O	O	
	DQMLU、DQMLL	Z			O	Z	Z* <sup>3</sup>	O	Z	O	O	
	$\overline{\text{AH}}$ (PA16)	Z			O	O	Z	Z* <sup>3</sup>	O	Z	O	

引脚功能		引脚状态										
分类	引脚名	复位状态				低功耗状态			总线权 释放 状态	检测到 振荡 停止时	使用 POE 功能时	
		上电			手动	深度软 件待机	软件待机	睡眠				
		无 ROM 扩展		有 ROM 扩展								单 芯 片
		8bit	16bit									
总线 控制	$\overline{\text{AH}}$ (PE14)	Z			O	Z	Z (HCPCR 的 MZIZEL=0) Z*3 (HCPCR 的 MZIZEL=1)	O	Z	O*7	O	
	RDWR	Z			O	Z	Z*3	O	Z	O	O	
	$\overline{\text{RD}}$	H	Z		O	Z	Z*3	O	Z	O	O	
	$\overline{\text{WRH}}$ 、 $\overline{\text{WRL}}$	H	Z		O	Z	Z*3	O	Z	O	O	
	CKE (PE15)	Z			O	Z	Z (HCPCR 的 MZIZEL=0) Z*2 (HCPCR 的 MZIZEL=1)	O	Z*2	O*7	O	
	CKE (PA9/PA16)	Z			O	Z	Z*2	O	Z*2	O	O	
DMAC	DREQ0、DREQ1	Z			I	Z	Z	I	I	I	I	
	DACK0、DACK1	Z			O	Z	Z (HCPCR 的 MZIZEL=0) O*1 (HCPCR 的 MZIZEL=1)	O	O	O*7	O	
	TEND0、TEND1	Z			O	Z	O*1	O	O	O	O	
MTU2	TCLKA ~ TCLKD	Z			I	Z	Z	I	I	I	I	
	TIOC0A ~ TIOC0D	Z			I/O	Z	K*1	I/O	I/O	I/O	Z	
	TIOC1A、TIOC1B	Z			I/O	Z	K*1	I/O	I/O	I/O	I/O	
	TIOC2A、TIOC2B	Z			I/O	Z	K*1	I/O	I/O	I/O	I/O	
	TIOC3A、TIOC3C	Z			I/O	Z	K*1	I/O	I/O	I/O	I/O	
	TIOC3B、TIOC3D	Z			I/O	Z	Z (HCPCR 的 MZIZEL=0) K*1 (HCPCR 的 MZIZEL=1)	I/O	I/O	I/O*7	Z	

引脚功能		引脚状态										
分类	引脚名	复位状态				低功耗状态			总线权 释放 状态	检测到 振荡 停止时	使用 POE 功能时	
		上电			手动	深度软 件待机	软件待机	睡眠				
		无 ROM 扩展		有 ROM 扩展								单 芯 片
		8bit	16bit									
MTU2	TIOC4A ~ TIOC4D	Z			I/O	Z	Z (HCPCR 的 MZIZEL=0) K*1 (HCPCR 的 MZIZEL=1)	I/O	I/O	I/O*7	Z	
	TIC5U、TIC5V、 TIC5W	Z			I	Z	Z	I	I	I	I	
	TIOC3AS、 TIOC3CS	Z			I/O	Z	K*1	I/O	I/O	I/O	I/O	
	TIOC3BS、 TIOC3DS	Z			I/O	Z	Z (HCPCR 的 MZIZDL=0) K*1 (HCPCR 的 MZIZDL=1)	I/O	I/O	I/O*5	Z	
	TIOC4AS ~ TIOC4DS	Z			I/O	Z	Z (HCPCR 的 MZIZDL=0 时) K*1 (HCPCR 的 MZIZDL=1)	I/O	I/O	I/O*5	Z	
	TIC5US、TIC5VS、 TIC5WS	Z			I	Z	Z	I	I	I	I	
	POE	$\overline{\text{POE0}} \sim \overline{\text{POE8}}$	Z			I	Z	Z	I	I	I	I
SCI	SCK0 ~ SCK2	Z			I/O	Z	Z	I/O	I/O	I/O	I/O	
	RXD0 ~ RXD2	Z			I	Z	Z	I	I	I	I	
	TXD0 ~ TXD2	Z			O	Z	O*1	O	O	O	O	
SCIF	SCK3 (PE6)	Z			I/O	Z	Z	I/O	I/O	I/O	I/O	
	SCK3 (PE9)	Z			I/O	Z	Z	I/O	I/O	I/O*7	I/O	
	RXD3 (PE4)	Z			I	Z	Z	I	I	I	I	
	RXD3 (PE11)	Z			I	Z	Z	I	I	I*7	I	
	TXD3 (PE5)	Z			O	Z	O*1	O	O	O	O	



引脚功能		引脚状态										
分类	引脚名	复位状态				低功耗状态			总线权 释放 状态	检测到 振荡 停止时	使用 POE 功能时	
		上电			手动	深度软 件待机	软件待机	睡眠				
		无 ROM 扩展		有 ROM 扩展								单 芯 片
		8bit	16bit									
SCIF	TXD3 (PE12)	Z			O	Z	Z (HCPCR 的 MZIZEL=0)	O	O	O*7	O	
							O*1 (HCPCR 的 MZIZEL=1)					
	RTS3	Z			O	Z	Z (HCPCR 的 MZIZEL=0)	O	O	O*7	O	
							O*1 (HCPCR 的 MZIZEL=1)					
	CTS3	Z			I	Z	Z	I	I	I*7	I	
SSU	SSCK	Z			I/O	Z	Z	I/O	I/O	I/O	I/O	
	SCS	Z			I/O	Z	Z	I/O	I/O	I/O*7	I/O	
	SSI	Z			I/O	Z	Z	I/O	I/O	I/O	I/O	
	SSO	Z			I/O	Z	Z	I/O	I/O	I/O	I/O	
I <sup>2</sup> C2	SCL	Z			I/O	Z	Z	I/O	I/O	I/O	I/O	
	SDA	Z			I/O	Z	Z	I/O	I/O	I/O	I/O	
UBC	UBCTRG	Z			O	Z	O*1	O	O	O	O	
A/D 转换器	AN0 ~ AN7	Z			I	Z	Z	I	I	I	I	
	ADTRG	Z			I	Z	Z	I	I	I	I	
I/O 端口	PA0 ~ PA17	Z			I/O	Z	K*1	I/O		I/O	I/O	
	PB0 ~ PB9	Z			I/O	Z	K*1	I/O		I/O	I/O	
	PC0 ~ PC15	Z			I/O	Z	K*1	I/O		I/O	I/O	
	PD0 ~ PD8, PD10	Z			I/O	Z	K*1	I/O		I/O	I/O	
	PD9, PD11 ~ PD15	Z			I/O	Z	Z (HCPCR 的 MZIZDL=0)	I/O	I/O	I/O*5	Z	
							K*1 (HCPCR 的 MZIZDL=1)					
		PE0 ~ PE3	Z			I/O	Z	K*1	I/O	I/O	I/O	Z
	PE4 ~ PE8, PE10	Z			I/O	Z	K*1	I/O	I/O	I/O	I/O	

引脚功能		引脚状态										
分类	引脚名	复位状态				手动	低功耗状态			总线权 释放 状态	检测到 振荡 停止时	使用 POE 功能时
		上电			单 芯 片		深度软 件待机	软件待机	睡眠			
		无 ROM 扩展		有 ROM 扩展								
		8bit	16bit									
I/O 端口	PE9、 PE11 ~ PE15	Z			I/O	Z	Z (HCPCR 的 MZIZEL=0)	I/O	I/O	I/O*7	Z	
							K*1 (HCPCR 的 MZIZEL=1)					
	PF0 ~ PF7	Z			I	Z	Z	I	I	I	I	

## 【符号说明】

- I : 输入  
O : 输出  
H : 高电平输出  
L : 低电平输出  
Z : 高阻抗  
K : 输入引脚为高阻抗，输出引脚为状态保持

【注】 \*1 如果将待机控制寄存器 6 (STBCR6) 的 HIZ 位置为 1，输出引脚就为高阻抗。

\*2 如果将共用控制寄存器 (CMNCR) 的 HIZCNT 位置为 1，就为输出。

\*3 如果将共用控制寄存器 (CMNCR) 的 HIZMEM 位置为 1，就为输出。

\*4 如果将大电流端口控制寄存器 (HCPCR) 的 MZIZDH 位置为 0，就为高阻抗。

\*5 如果将大电流端口控制寄存器 (HCPCR) 的 MZIZDL 位置为 0，就为高阻抗。

\*6 如果将大电流端口控制寄存器 (HCPCR) 的 MZIZEH 位置为 0，就为高阻抗。

\*7 如果将大电流端口控制寄存器 (HCPCR) 的 MZIZEL 位置为 0，就为高阻抗。

\*8 在上电复位中为输入状态。为了防止误动作，必须进行上拉。需要进行下拉时，必须至少用 1MΩ 的电阻值进行下拉。

\*9 什么也不输入时，在内部进行上拉。

表 A.3 引脚状态 (SH7085)

引脚功能		引脚状态										
分类	引脚名	复位状态				低功耗状态			总线 权 释放 状态	检测 到 振荡 停止 时	使用 POE 功能 时	
		上电			手动	深度 软件 待机	软件待机	睡眠				
		无 ROM 扩展		有 ROM 扩展								单 芯 片
		16bit	32bit									
时钟	CK	O		Z	O	Z	H* <sup>1</sup>	O	O	O	O	
	XTAL	O			O	L	L	O	O	O	O	
	EXTAL	I			I	Z	I	I	I	I	I	
系统 控制	$\overline{\text{RES}}$	I			I	I	I	I	I	I	I	
	$\overline{\text{MRES}}$	Z			I	Z	I* <sup>7</sup>	I	I	I* <sup>7</sup>	I	
	$\overline{\text{WDTOVF}}$	O* <sup>8</sup>			O	O	O	O	O	O	O	
	$\overline{\text{BREQ}}$	Z			I	Z	Z	I	I	I	I	
	$\overline{\text{BACK}}$	Z			O	Z	Z	O	L	O	O	
运行 模式 控制	MD0、MD1	I			I	I	I	I	I	I	I	
	$\overline{\text{ASEMD0}}$	I* <sup>9</sup>			I* <sup>9</sup>	I* <sup>9</sup>	I* <sup>9</sup>	I* <sup>9</sup>	I* <sup>9</sup>	I* <sup>9</sup>	I* <sup>9</sup>	
	FWE	I			I	I	I	I	I	I	I	
中断	NMI	I			I	I	I	I	I	I	I	
	IRQ0 ~ IRQ7	Z			I	Z	I	I	I	I	I	
	$\overline{\text{IRQOUT}}$ (PE15)	Z			O	Z	Z (HCPCR 的 MZIZEL=0)	O	O	O* <sup>7</sup>	O	
							H* <sup>1</sup> (HCPCR 的 MZIZEL=1)					
	$\overline{\text{IRQOUT}}$ (PD30)	Z			O	Z	H* <sup>1</sup>	O	O	O	O	
地址 总线	A0 ~ A17	O	Z		O	Z	Z* <sup>3</sup>	O	Z	O	O	
	A18 ~ A25	Z			O	Z	Z* <sup>3</sup>	O	Z	O	O	
数据 总线	D0 ~ D8、D10、 D16 ~ D23、 D30、D31	Z			I/O	Z	Z	I/O	Z	I/O	I/O	
	D9、D11 ~ D15	Z			I/O	Z	Z	I/O	Z	I/O* <sup>5</sup>	I/O	
	D24 ~ D29	Z			I/O	Z	Z	I/O	Z	I/O* <sup>4</sup>	I/O	
总线 控制	$\overline{\text{WAIT}}$	Z			I	Z	Z	I	Z	I	I	
	CS0、CS1	H	Z		O	Z	Z* <sup>3</sup>	O	Z	O	O	
	$\overline{\text{CS2}}$ (PA6)、 $\overline{\text{CS3}}$ (PA7)、 $\overline{\text{CS4}}$ ~ $\overline{\text{CS7}}$	Z			O	Z	Z* <sup>3</sup>	O	Z	O	O	

引脚功能				引脚状态								
分类	引脚名	复位状态				低功耗状态			总线 权 释放 状态	检测 到 振荡 停止 时	使用 POE 功能 时	
		上电			手动	深度 软件 待机	软件待机	睡眠				
		无 ROM 扩展		有 ROM 扩展								单 芯 片
		16bit	32bit									
总线 控制	$\overline{\text{CS2}}$ (PD28)、 $\overline{\text{CS3}}$ (PD29)	Z			O	Z	Z (HCPCR 的 MZIZDH=0) Z* <sup>3</sup> (HCPCR 的 MZIZDH=1)	O	Z	O* <sup>4</sup>	O	
	$\overline{\text{CE1A}}$ 、 $\overline{\text{CE1B}}$ 、 $\overline{\text{CE2A}}$ 、 $\overline{\text{CE2B}}$	Z			O	Z	Z* <sup>3</sup>	O	Z	O	O	
	$\overline{\text{BS}}$	Z			O	Z	Z* <sup>3</sup>	O	Z	O	O	
	$\overline{\text{RASU}}$ 、 $\overline{\text{RASL}}$	Z			O	Z	Z* <sup>2</sup>	O	Z* <sup>2</sup>	O	O	
	$\overline{\text{CASU}}$ 、 $\overline{\text{CASL}}$	Z			O	Z	Z* <sup>2</sup>	O	Z* <sup>2</sup>	O	O	
	DQMUU (PA23/PA16)、 DQMUL、DQMLU、 DQMLL	Z			O	Z	Z* <sup>3</sup>	O	Z	O	O	
	DQMUU (PE14)	Z			O	Z	Z (HCPCR 的 MZIZEL=0) Z* <sup>3</sup> (HCPCR 的 MZIZEL=1)	O	Z	O* <sup>7</sup>	O	
	$\overline{\text{AH}}$ (PA23/PA16)	Z			O	Z	Z* <sup>3</sup>	O	Z	O	O	
	$\overline{\text{AH}}$ (PE14)	Z			O	Z	Z (HCPCR 的 MZIZEL=0) Z* <sup>3</sup> (HCPCR 的 MZIZEL=1)	O	Z	O* <sup>7</sup>	O	
	$\overline{\text{FRAME}}$	Z			O	Z	Z* <sup>3</sup>	O	Z	O	O	
	$\overline{\text{RDWR}}$	Z			O	Z	Z* <sup>3</sup>	O	Z	O	O	
	$\overline{\text{RD}}$	H	Z		O	Z	Z* <sup>3</sup>	O	Z	O	O	
	$\overline{\text{ICIORD}}$	Z			O	Z	Z* <sup>3</sup>	O	Z	O	O	
	$\overline{\text{WRHH}}$ (PA23)、 $\overline{\text{WRHL}}$	Z	H	Z	O	Z	Z* <sup>3</sup>	O	Z	O	O	

引脚功能		引脚状态										
分类	引脚名	复位状态				低功耗状态			总线 权 释放 状态	检测 到 振荡 停止 时	使用 POE 功能 时	
		上电			手动	深度 软件 待机	软件待机	睡眠				
		无 ROM 扩展		有 ROM 扩展								单 芯 片
		16bit	32bit									
总线 控制	$\overline{\text{WRHH}}$ (PE14)	Z			O	Z	Z (HCPCR 的 MZIZEL=0)	O	Z	O*7	O	
	$\overline{\text{WRHH}}$ (PA16)	Z			O	Z	Z*3	O	Z	O	O	
	$\overline{\text{WRH}}$ 、 $\overline{\text{WRL}}$	H	Z		O	Z	Z*3	O	Z	O	O	
	$\overline{\text{WE}}$	Z			O	Z	Z*3	O	Z	O	O	
	$\overline{\text{ICIOWR}}$ (PA23/PA16)	Z			O	Z	Z*3	O	Z	O	O	
	$\overline{\text{ICIOWR}}$ (PE14)	Z			O	Z	Z (HCPCR 的 MZIZEL=0)	O	Z	O*7	O	
		Z					Z*3 (HCPCR 的 MZIZEL=1)					
	$\overline{\text{IOIS16}}$	Z			I	Z	Z	I	I	I	I	
	$\overline{\text{CKE}}$ (PE15)	Z			O	Z	Z (HCPCR 的 MZIZEL=0)	O	Z*2	O*7	O	
	Z					Z*2 (HCPCR 的 MZIZEL=1)						
$\overline{\text{CKE}}$ (PA9/PA16)	Z			O	Z	Z*2	O	Z*2	O	O		
DMAC	DREQ0 (PD24)、 DREQ1 (PD25)	Z			I	Z	Z	I	I	I*4	I	
	DREQ0 (PA2/PE0)、 DREQ1 (PA5/PE2)、 DREQ2、DREQ3	Z			I	Z	Z	I	I	I	I	

引脚功能				引脚状态								
分类	引脚名	复位状态				低功耗状态			总线 权 释放 状态	检测 到 振荡 停止 时	使用 POE 功能 时	
		上电			手动	深度 软件 待机	软件待机	睡眠				
		无 ROM 扩展		有 ROM 扩展								单 芯 片
		16bit	32bit									
DMAC	DACK0 (PD26)、 DACK1 (PD27)	Z				O	Z	Z (HCPCR 的 MZIZDH=0)	O	O	O*4	O
							O*1 (HCPCR 的 MZIZDH=1)					
	DACK0 (PE14)、 DACK1 (PE15)	Z				O	Z	Z (HCPCR 的 MZIZEL=0)	O	O	O*7	O
							O*1 (HCPCR 的 MZIZEL=1)					
	DACK2、DACK3		Z				O	Z	O*1	O	O	O
TEND0、TEND1		Z				O	Z	O*1	O	O	O	O
MTU2	TCLKA ~ TCLKD	Z				I	Z	Z	I	I	I	I
	TIOC0A ~ TIOC0D	Z				I/O	Z	K*1	I/O	I/O	I/O	Z
	TIOC1A、TIOC1B	Z				I/O	Z	K*1	I/O	I/O	I/O	I/O
	TIOC2A、TIOC2B	Z				I/O	Z	K*1	I/O	I/O	I/O	I/O
	TIOC3A、TIOC3C	Z				I/O	Z	K*1	I/O	I/O	I/O	I/O
	TIOC3B、TIOC3D	Z				I/O	Z	Z (HCPCR 的 MZIZEL=0)	I/O	I/O	I/O*7	Z
								K*1 (HCPCR 的 MZIZEL=1)				
TIOC4A ~ TIOC4D	Z				I/O	Z	Z (HCPCR 的 MZIZEL=0)	I/O	I/O	I/O*7	Z	
							K*1 (HCPCR 的 MZIZEL=1)					
TIC5U、TIC5V、 TIC5W		Z				I	Z	Z	I	I	I	I
MTU2S	TIOC3AS、 TIOC3CS	Z				I/O	Z	K*1	I/O	I/O	I/O	I/O

引脚功能		引脚状态										
分类	引脚名	复位状态				低功耗状态			总线 权 释放 状态	检测 到 振荡 停止 时	使用 POE 功能 时	
		上电			手动	深度 软件 待机	软件待机	睡眠				
		无 ROM 扩展		有 ROM 扩展								单 芯 片
		16bit	32bit									
MTU2S	TIOC3BS (PD9)、 TIOC3DS (PD11)	Z			I/O	Z	Z (HCPCR 的 MZIZDL=0)	I/O	I/O	I/O* <sup>5</sup>	Z	
						K* <sup>1</sup> (HCPCR 的 MZIZDL=1)						
	TIOC3B (PD29)、 TIOC3DS (PD28)	Z			I/O	Z	Z (HCPCR 的 MZIZDH=0)	I/O	I/O	I/O* <sup>4</sup>	Z	
						K* <sup>1</sup> (HCPCR 的 MZIZDH=1)						
	TIOC4AS (PD12)、 TIOC4BS (PD13)、 TIOC4CS (PD14)、 TIOC4DS (PD15)、	Z			I/O	Z	Z (HCPCR 的 MZIZDL=0)	I/O	I/O	I/O* <sup>5</sup>	Z	
						K* <sup>1</sup> (HCPCR 的 MZIZDL=1)						
TIOC4AS (PD27)、 TIOC4BS (PD26)、 TIOC4CS (PD25)、 TIOC4DS (PD24)、	Z			I/O	Z	Z (HCPCR 的 MZIZDH=0)	I/O	I/O	I/O* <sup>4</sup>	Z		
					K* <sup>1</sup> (HCPCR 的 MZIZDH=1)							
TIC5US、TIC5VS、 TIC5WS	Z			I	Z	Z	I	I	I	I		
POE	$\overline{\text{POE0}} \sim \overline{\text{POE8}}$	Z			I	Z	Z	I	I	I	I	
SCI	SCK0 ~ SCK2	Z			I/O	Z	Z	I/O	I/O	I/O	I/O	
	RXD0 ~ RXD2	Z			I	Z	Z	I	I	I	I	
	TXD0 ~ TXD2	Z			O	Z	O* <sup>1</sup>	O	O	O	O	
SCIF	SCK3 (PE6)	Z			I/O	Z	Z	I/O	I/O	I/O	I/O	
	SCK3 (PE9)	Z			I/O	Z	Z	I/O	I/O	I/O* <sup>7</sup>	I/O	
	RXD3 (PE4)	Z			I	Z	Z	I	I	I	I	
	RXD3 (PE11)	Z			I	Z	Z	I	I	I* <sup>7</sup>	I	

引脚功能				引脚状态								
分类	引脚名	复位状态				低功耗状态			总线 权 释放 状态	检测 到 振荡 停止 时	使用 POE 功能 时	
		上电			手动	深度 软件 待机	软件待机	睡眠				
		无 ROM 扩展		有 ROM 扩展								单 芯 片
		16bit	32bit									
SCIF	TXD3 (PE5)	Z				O	Z	O* <sup>1</sup>	O	O	O	O
	TXD3 (PE12)	Z				O	Z	Z (HCPCR 的 MZIZEL=0)	O	O	O* <sup>7</sup>	O
		Z						O* <sup>1</sup> (HCPCR 的 MZIZEL=1)				
	RTS3	Z				O	Z	Z (HCPCR 的 MZIZEL=0)	O	O	O* <sup>7</sup>	O
		Z						O* <sup>1</sup> (HCPCR 的 MZIZEL=1)				
CTS3	Z				I	Z	Z	I	I	I* <sup>7</sup>	I	
SSU	SSCK	Z				I/O	Z	Z	I/O	I/O	I/O	I/O
	SCS	Z				I/O	Z	Z	I/O	I/O	I/O* <sup>7</sup>	I/O
	SSI	Z				I/O	Z	Z	I/O	I/O	I/O	I/O
	SSO	Z				I/O	Z	Z	I/O	I/O	I/O	I/O
I <sup>2</sup> C2	SCL	Z				I/O	Z	Z	I/O	I/O	I/O	I/O
	SDA	Z				I/O	Z	Z	I/O	I/O	I/O	I/O
UBC	UBCTRG	Z				O	Z	O* <sup>1</sup>	O	O	O	O
A/D 转换器	AN0 ~ AN7	Z				I	Z	Z	I	I	I	I
	ADTRG	Z				I	Z	Z	I	I	I	I
I/O 端口	PA0 ~ PA25	Z				I/O	Z	K* <sup>1</sup>	I/O	I/O	I/O	I/O
	PB0 ~ PB9	Z				I/O	Z	K* <sup>1</sup>	I/O	I/O	I/O	I/O
	PC0 ~ PC15	Z				I/O	Z	K* <sup>1</sup>	I/O	I/O	I/O	I/O
	PD0 ~ PD8、PD10、 PD16 ~ PD23、 PD30、PD31	Z				I/O	Z	K* <sup>1</sup>	I/O	I/O	I/O	I/O
	PD9、 PD11 ~ PD15	Z				I/O	Z	Z (HCPCR 的 MZIZDL=0)	I/O	I/O	I/O* <sup>5</sup>	Z
Z				K* <sup>1</sup> (HCPCR 的 MZIZDL=1)								



引脚功能				引脚状态								
分类	引脚名	复位状态				低功耗状态			总线 权 释放 状态	检测 到 振荡 停止 时	使用 POE 功能 时	
		上电			手 动	深 度 软 件 待 机	软 件 待 机	睡 眠				
		无 ROM 扩展		有 ROM 扩 展								单 芯 片
		16bit	32bit									
I/O 端 口	PD24 ~ PD29	Z				I/O	Z	Z (HCPCR 的 MZIZDH=0) K*1 (HCPCR 的 MZIZDH=1)	I/O	I/O	I/O*4	Z
	PE0 ~ PE3	Z				I/O	Z	K*1	I/O	I/O	I/O	Z
	PE4 ~ PE8、 PE10	Z				I/O	Z	K*1	I/O	I/O	I/O	I/O
	PE9、 PE11 ~ PE15	Z				I/O	Z	Z (HCPCR 的 MZIZEL=0) K*1 (HCPCR 的 MZIZEL=1)	I/O	I/O	I/O*7	Z
	PF0 ~ PF7	Z				I	Z	Z	I	I	I	I

## 【符号说明】

- I : 输入  
O : 输出  
H : 高电平输出  
L : 低电平输出  
Z : 高阻抗  
K : 输入引脚为高阻抗，输出引脚为状态保持

【注】\*1 如果将待机控制寄存器 6 (STBCR6) 的 HIZ 位置为 1，输出引脚就为高阻抗。

\*2 如果将共用控制寄存器 (CMNCR) 的 HIZCNT 位置为 1，就为输出。

\*3 如果将共用控制寄存器 (CMNCR) 的 HIZMEM 位置为 1，就为输出。

\*4 如果将大电流端口控制寄存器 (HCPCR) 的 MZIZDH 位置为 0，就为高阻抗。

\*5 如果将大电流端口控制寄存器 (HCPCR) 的 MZIZDL 位置为 0，就为高阻抗。

\*6 如果将大电流端口控制寄存器 (HCPCR) 的 MZIZEH 位置为 0，就为高阻抗。

\*7 如果将大电流端口控制寄存器 (HCPCR) 的 MZIZEL 位置为 0，就为高阻抗。

\*8 在上电复位中为输入状态。为了防止误动作，必须进行上拉。需要进行下拉时，必须至少用 1MΩ 的电阻值进行下拉。

\*9 什么也不输入时，在内部进行上拉。

表 A.4 引脚状态 (SH7086)

引脚功能		引脚状态										
分类	引脚名	复位状态					低功耗状态			总线权释放状态	检测到振荡停止时	使用 POE 功能时
		上电			手动	深度软件待机	软件待机	睡眠				
		无 ROM 扩展		有 ROM 扩展					单芯片			
		16bit	32bit									
时钟	CK	O		Z	O	Z	H* <sup>1</sup>	O	O	O	O	
	XTAL	O			O	L	L	O	O	O	O	
	EXTAL	I			I	Z	I	I	I	I	I	
系统控制	RES	I			I	I	I	I	I	I	I	
	MRES	Z			I	Z	I* <sup>7</sup>	I	I	I* <sup>7</sup>	I	
	WDTOVF	O* <sup>8</sup>			O	O	O	O	O	O	O	
	BREQ	Z			I	Z	Z	I	I	I	I	
	BACK	Z			O	Z	Z	O	L	O	O	
运行模式控制	MD0、MD1	I			I	I	I	I	I	I	I	
	ASEMD0	I* <sup>9</sup>			I* <sup>9</sup>	I* <sup>9</sup>	I* <sup>9</sup>	I* <sup>9</sup>	I* <sup>9</sup>	I* <sup>9</sup>	I* <sup>9</sup>	
	FWE	I			I	I	I	I	I	I	I	
中断	NMI	I			I	I	I	I	I	I	I	
	IRQ0 ~ IRQ7	Z			I	Z	I	I	I	I	I	
	IRQOUT (PE15)	Z			O	Z	Z (HCPCR 的 MZIZEL=0) H* <sup>1</sup> (HCPCR 的 MZIZEL=1)	O	O	O* <sup>7</sup>	O	
	IRQOUT (PD30)	Z			O	Z	H* <sup>1</sup>	O	O	O	O	
地址总线	A0 ~ A25	O	Z		O	Z	Z* <sup>3</sup>	O	Z	O	O	
	A26 ~ A29	Z			O	Z	Z* <sup>3</sup>	O	Z	O	O	
数据总线	D0 ~ D8、D10、D16 ~ D23、D30、D31	Z			I/O	Z	Z	I/O	Z	I/O	I/O	
	D9、D11 ~ D15	Z			I/O	Z	Z	I/O	Z	I/O* <sup>5</sup>	I/O	
	D24 ~ D29	Z			I/O	Z	Z	I/O	Z	I/O* <sup>4</sup>	I/O	
总线控制	WAIT	Z			I	Z	Z	I	Z	I	I	
	CS0、CS1	H	Z		O	Z	Z* <sup>3</sup>	O	Z	O	O	
	CS2 (PA6)、CS3 (PA7)、CS4 ~ CS7	Z			O	Z	Z* <sup>3</sup>	O	Z	O	O	

引脚功能		引脚状态										
分类	引脚名	复位状态				低功耗状态			总线权释放状态	检测到振荡停止时	使用 POE 功能时	
		上电			手动	深度软件待机	软件待机	睡眠				
		无 ROM 扩展		有 ROM 扩展								单芯片
		16bit	32bit									
总线控制	$\overline{\text{CS2}}$ (PD28)、 $\overline{\text{CS3}}$ (PD29)	Z			O	Z	Z (HCPCR 的 MZIZDH=0) Z*3 (HCPCR 的 MZIZDH=1)	O	Z	O*4	O	
	$\overline{\text{CS8}}$	Z			O	Z	Z (HCPCR 的 MZIZEH=0) Z*3 (HCPCR 的 MZIZEH=1)	O	Z	O*6	O	
	$\overline{\text{CE1A}}$ 、 $\overline{\text{CE1B}}$ 、 $\overline{\text{CE2A}}$ 、 $\overline{\text{CE2B}}$	Z			O	Z	Z*3	O	Z	O	O	
	$\overline{\text{BS}}$	Z			O	Z	Z*3	O	Z	O	O	
	$\overline{\text{RASU}}$ 、 $\overline{\text{RASL}}$	Z			O	Z	Z*2	O	Z*2	O	O	
	$\overline{\text{CASU}}$ 、 $\overline{\text{CASL}}$	Z			O	Z	Z*2	O	Z*2	O	O	
	DQMUU (PA23/PA16)、DQMUL、DQMLU、DQMLL	Z			O	Z	Z*3	O	Z	O	O	
	DQMUU (PE14)	Z			O	Z	Z (HCPCR 的 MZIZEL=0) Z*3 (HCPCR 的 MZIZEL=1)	O	Z	O*7	O	
	$\overline{\text{AH}}$ (PA23/PA16)	Z			O	Z	Z*3	O	Z	O	O	
	$\overline{\text{AH}}$ (PE14)	Z			O	Z	Z (HCPCR 的 MZIZEL=0) Z*3 (HCPCR 的 MZIZEL=1)	O	Z	O*7	O	
	$\overline{\text{FRAME}}$	Z			O	Z	Z*3	O	Z	O	O	
	$\overline{\text{RDWR}}$	Z			O	Z	Z*3	O	Z	O	O	
	$\overline{\text{RD}}$	H	Z		O	Z	Z*3	O	Z	O	O	
	$\overline{\text{ICIORD}}$	Z			O	Z	Z*3	O	Z	O	O	
	$\overline{\text{WRHH}}$ (PA23)、 $\overline{\text{WRHL}}$	Z	H	Z	O	Z	Z*3	O	Z	O	O	

引脚功能		引脚状态										
分类	引脚名	复位状态				低功耗状态			总线权 释放状 态	检测到 振荡停 止时	使用 POE 功能时	
		上电			手动	深度软 件待机	软件待 机	睡眠				
		无 ROM 扩展		有 ROM 扩展								单芯 片
		16bit	32bit									
总线 控制	WRHH (PE14)	Z			O	Z	Z (HCPCR 的 MZIZEL=0)	O	Z	O*7	O	
		Z			O	Z	Z*3 (HCPCR 的 MZIZEL=1)	O	Z	O	O	
	WRHH (PA16)	Z			O	Z	Z*3	O	Z	O	O	
	WRH、WRL	H	Z		O	Z	Z*3	O	Z	O	O	
	WE	Z			O	Z	Z*3	O	Z	O	O	
	ICIORW (PA23/PA16)	Z			O	Z	Z*3	O	Z	O	O	
	ICIORW (PE14)	Z			O	Z	Z (HCPCR 的 MZIZEL=0)	O	Z	O*7	O	
		Z			O	Z	Z*3 (HCPCR 的 MZIZEL=1)	O	Z	O	O	
	IOIS16	Z			I	Z	Z	I	I	I	I	
	CKE (PE15)	Z			O	Z	Z (HCPCR 的 MZIZEL=0)	O	Z*2	O*7	O	
Z			O	Z	Z*2 (HCPCR 的 MZIZEL=1)	O	Z*2	O	O			
CKE (PA9/PA16)	Z			O	Z	Z*2	O	Z*2	O	O		
DMAC	DREQ0 (PD24)、 DREQ1 (PD25)	Z			I	Z	Z	I	I	I*4	I	
	DREQ0 (PA2/PE0)、 DREQ1 (PA5/PE2)、 DREQ2、DREQ3	Z			I	Z	Z	I	I	I	I	
	DACK0 (PD26)、 DACK1 (PD27)	Z			O	Z	Z (HCPCR 的 MZIZDH=0)	O	O	O*4	O	
Z			O	Z	O*1 (HCPCR 的 MZIZDH=1)	O	O	O	O			

引脚功能		引脚状态										
分类	引脚名	复位状态				低功耗状态			总线权 释放状 态	检测到 振荡停 止时	使用 POE 功能时	
		上电			手动	深度软 件待机	软件待 机	睡眠				
		无 ROM 扩展		有 ROM 扩展								单芯 片
		16bit	32bit									
DMAC	DACK0 (PE14)、 DACK1 (PE15)	Z			O	Z	Z (HCPCR 的 MZIZEL=0) O*1 (HCPCR 的 MZIZEL=1)	O	O	O*7	O	
	DACK2、DACK3	Z			O	Z	O*1	O	O	O	O	
	TEND0、TEND1	Z			O	Z	O*1	O	O	O	O	
MTU2	TCLKA ~ TCLKD	Z			I	Z	Z	I	I	I	I	
	TIOC0A ~ TIOC0D	Z			I/O	Z	K*1	I/O	I/O	I/O	Z	
	TIOC1A、TIOC1B	Z			I/O	Z	K*1	I/O	I/O	I/O	I/O	
	TIOC2A、TIOC2B	Z			I/O	Z	K*1	I/O	I/O	I/O	I/O	
	TIOC3A、TIOC3C	Z			I/O	Z	K*1	I/O	I/O	I/O	I/O	
	TIOC3B、 TIOC3D	Z			I/O	Z	Z (HCPCR 的 MZIZEL=0) K*1 (HCPCR 的 MZIZEL=1)	I/O	I/O	I/O*7	Z	
	TIOC4A ~ TIOC4D	Z			I/O	Z	Z (HCPCR 的 MZIZEL=0) K*1 (HCPCR 的 MZIZEL=1)	I/O	I/O	I/O*7	Z	
TIC5U、TIC5V、 TIC5W	Z			I	Z	Z	I	I	I	I		
MTU2S	TIOC3AS、 TIOC3CS	Z			I/O	Z	K*1	I/O	I/O	I/O	I/O	
	TIOC3BS (PD9)、 TIOC3DS (PD11)	Z			I/O	Z	Z (HCPCR 的 MZIZDL=0) K*1 (HCPCR 的 MZIZDL=1)	I/O	I/O	I/O*5	Z	
	TIOC3BS (PD29)、 TIOC3DS (PD28)	Z			I/O	Z	Z (HCPCR 的 MZIZDH=0) K*1 (HCPCR 的 MZIZDH=1)	I/O	I/O	I/O*4	Z	

引脚功能		引脚状态									
分类	引脚名	复位状态				低功耗状态			总线权 释放状 态	检测到 振荡停 止时	使用 POE 功能时
		上电			手动	深度软 件待机	软件待 机	睡眠			
		无 ROM 扩展	有 ROM 扩展	单芯 片							
MTU2S	TIOC3BS (PE16)、 TIOC3DS (PE17)	Z			I/O	Z	Z (HCPCR 的 MZIZEH=0)	I/O	I/O	I/O*6	Z
		Z			I/O	Z	K*1 (HCPCR 的 MZIZEH=1)	I/O	I/O	I/O*5	Z
	TIOC4AS (PD12)、 TIOC4BS (PD13)、 TIOC4CS (PD14)、 TIOC4DS (PD15)、	Z			I/O	Z	Z (HCPCR 的 MZIZDL=0)	I/O	I/O	I/O*4	Z
		Z			I/O	Z	K*1 (HCPCR 的 MZIZDL=1)	I/O	I/O	I/O*6	Z
	TIOC4AS (PD27)、 TIOC4BS (PD26)、 TIOC4CS (PD25)、 TIOC4DS (PD24)、	Z			I/O	Z	Z (HCPCR 的 MZIZDH=0)	I/O	I/O	I/O*6	Z
		Z			I/O	Z	K*1 (HCPCR 的 MZIZDH=1)	I/O	I/O	I/O*6	Z
TIOC4AS (PE18)、 TIOC4BS (PE19)、 TIOC4CS (PE20)、 TIOC4DS (PE21)、	Z			I/O	Z	Z (HCPCR 的 MZIZEH=0)	I/O	I/O	I/O*6	Z	
TIC5US、 TIC5VS、 TIC5WS	Z			I	Z	Z (HCPCR 的 MZIZEH=1)	I	I	I	I	
POE	$\overline{POE0} \sim \overline{POE8}$	Z			I	Z	Z	I	I	I	I
SCI	SCK0 ~ SCK2	Z			I/O	Z	Z	I/O	I/O	I/O	I/O
	RXD0 ~ RXD2	Z			I	Z	Z	I	I	I	I
	TXD0 ~ TXD2	Z			O	Z	O*1	O	O	O	O
SCIF	SCK3 (PE6)	Z			I/O	Z	Z	I/O	I/O	I/O	I/O
	SCK3 (PE9)	Z			I/O	Z	Z	I/O	I/O	I/O*7	I/O
	RXD3 (PE4)	Z			I	Z	Z	I	I	I	I
	RXD3 (PE11)	Z			I	Z	Z	I	I	I*7	I
	TXD3 (PE5)	Z			O	Z	O*1	O	O	O	O
	TXD3 (PE12)	Z			O	Z	Z (HCPCR 的 MZIZEL=0)	O	O	O*7	O
							O*1 (HCPCR 的 MZIZEL=1)				

引脚功能		引脚状态										
分类	引脚名	复位状态				低功耗状态			总线权 释放状 态	检测到 振荡停 止时	使用 POE 功能时	
		上电			手动	深度软 件待机	软件待 机	睡眠				
		无 ROM 扩展		有 ROM 扩展								单芯 片
		16bit	32bit									
SCIF	$\overline{\text{RTS3}}$	Z			O	Z	Z (HCPCR 的 MZIZEL=0)	O	O	O*7	O	
							O*1 (HCPCR 的 MZIZEL=1)					
	$\overline{\text{CTS3}}$	Z			I	Z	Z	I	I	I*7	I	
SSU	SSCK	Z			I/O	Z	Z	I/O	I/O	I/O	I/O	
	$\overline{\text{SCS}}$	Z			I/O	Z	Z	I/O	I/O	I/O*7	I/O	
	SSI	Z			I/O	Z	Z	I/O	I/O	I/O	I/O	
	SSO	Z			I/O	Z	Z	I/O	I/O	I/O	I/O	
I <sup>2</sup> C2	SCL	Z			I/O	Z	Z	I/O	I/O	I/O	I/O	
	SDA	Z			I/O	Z	Z	I/O	I/O	I/O	I/O	
UBC	$\overline{\text{UBCTRG}}$	Z			O	Z	O*1	O	O	O	O	
A/D 转换器	AN0 ~ AN15	Z			I	Z	Z	I	I	I	I	
	$\overline{\text{ADTRG}}$	Z			I	Z	Z	I	I	I	I	
I/O 端口	PA0 ~ PA29	Z			I/O	Z	K*1	I/O	I/O	I/O	I/O	
	PB0 ~ PB9	Z			I/O	Z	K*1	I/O	I/O	I/O	I/O	
	PC0 ~ PC15、 PC18 ~ PC25	Z			I/O	Z	K*1	I/O	I/O	I/O	I/O	
	PD0 ~ PD8、 PD10、 PD16 ~ PD23、 PD30、PD31	Z			I/O	Z	K*1	I/O	I/O	I/O	I/O	
	PD9、 PD11 ~ PD15	Z			I/O	Z	Z (HCPCR 的 MZIZDL=0)	I/O	I/O	I/O*5	Z	
					I/O		K*1 (HCPCR 的 MZIZDL=1)					
	PD24 ~ PD29	Z			I/O	Z	Z (HCPCR 的 MZIZDH=0)	I/O	I/O	I/O*4	Z	
							K*1 (HCPCR 的 MZIZDH=1)					
PE0 ~ PE3	Z			I/O	Z	K*1	I/O	I/O	I/O	Z		
PE4 ~ PE8、 PE10	Z			I/O	Z	K*1	I/O	I/O	I/O	I/O		

引脚功能		引脚状态										
分类	引脚名	复位状态				低功耗状态			总线权释放状态	检测到振荡停止时	使用 POE 功能时	
		上电			手动	深度软件待机	软件待机	睡眠				
		无 ROM 扩展		有 ROM 扩展								单芯片
		16bit	32bit									
I/O 端口	PE9、PE11 ~ PE15	Z			I/O	Z	Z (HCPCR 的 MZIZEL=0)	I/O	I/O	I/O*7	Z	
		Z				K*1 (HCPCR 的 MZIZEL=1)						
	PE16 ~ PE21	Z			I/O	Z	Z (HCPCR 的 MZIZEH=0)	I/O	I/O	I/O*6	Z	
Z			K*1 (HCPCR 的 MZIZEH=1)									
	PF0 ~ PF15	Z			I	Z	Z	I	I	I	I	

## 【符号说明】

- I : 输入  
O : 输出  
H : 高电平输出  
L : 低电平输出  
Z : 高阻抗  
K : 输入引脚为高阻抗, 输出引脚为状态保持

【注】 \*1 如果将待机控制寄存器 6 (STBCR6) 的 HIZ 位置为 1, 输出引脚就为高阻抗。

\*2 如果将共用控制寄存器 (CMNCR) 的 HIZCNT 位置为 1, 就为输出。

\*3 如果将共用控制寄存器 (CMNCR) 的 HIZMEM 位置为 1, 就为输出。

\*4 如果将大电流端口控制寄存器 (HCPCR) 的 MZIZDH 位置为 0, 就为高阻抗。

\*5 如果将大电流端口控制寄存器 (HCPCR) 的 MZIZDL 位置为 0, 就为高阻抗。

\*6 如果将大电流端口控制寄存器 (HCPCR) 的 MZIZEH 位置为 0, 就为高阻抗。

\*7 如果将大电流端口控制寄存器 (HCPCR) 的 MZIZEL 位置为 0, 就为高阻抗。

\*8 在上电复位中为输入状态。为了防止误动作, 必须进行上拉。需要进行下拉时, 必须至少用 1MΩ 的电阻值进行下拉。

\*9 什么也不输入时, 在内部进行上拉。



## 附录 B. 总线相关信号的引脚状态

表 B.1 总线相关信号的引脚状态 (1)

引脚名		内部 ROM 空间	内部 RAM 空间	内部外围模块空间
$\overline{CS0} \sim \overline{CS8}$		H	H	H
$\overline{CE1A}$ 、 $\overline{CE1B}$ 、 $\overline{CE2A}$ 、 $\overline{CE2B}$		H	H	H
$\overline{BS}$		H	H	H
$\overline{RASU}$ 、 $\overline{RASL}$		H	H	H
$\overline{CASU}$ 、 $\overline{CASL}$		H	H	H
DQMUU		H	H	H
DQMUL		H	H	H
DQMLU		H	H	H
DQMLL		H	H	H
$\overline{AH}$		L	L	L
$\overline{FRAME}$		H	H	H
RDWR	R	H	H	H
	W	—	H	H
$\overline{RD}$	R	H	H	H
	W	—	H	H
$\overline{ICIORD}$	R	H	H	H
	W	—	H	H
$\overline{WRHH}$	R	H	H	H
	W	—	H	H
$\overline{WRHL}$	R	H	H	H
	W	—	H	H
$\overline{WRH}$	R	H	H	H
	W	—	H	H
$\overline{WRL}$	R	H	H	H
	W	—	H	H
$\overline{WE}$	R	H	H	H
	W	—	H	H
$\overline{ICIOWR}$	R	H	H	H
	W	—	H	H
A29 ~ A0		地址 *	地址 *	地址 *
D31 ~ D24		High-Z	High-Z	High-Z
D23 ~ D16		High-Z	High-Z	High-Z
D15 ~ D8		High-Z	High-Z	High-Z
D7 ~ D0		High-Z	High-Z	High-Z

【符号说明】 R: 读 W: 写

【注】 \* 以前存取的外部空间的地址值

表 B.1 总线相关信号的引脚状态 (2)

引脚名	外部空间 (通常空间)			
	8 位空间	16 位空间		
		高位字节	低位字节	字 / 长字
$\overline{\text{CS0}} \sim \overline{\text{CS8}}$	有效	有效	有效	有效
$\overline{\text{CE1A}}$ 、 $\overline{\text{CE1B}}$ 、 $\overline{\text{CE2A}}$ 、 $\overline{\text{CE2B}}$	H	H	H	H
$\overline{\text{BS}}$	L	L	L	L
$\overline{\text{RASU}}$ 、 $\overline{\text{RASL}}$	H	H	H	H
$\overline{\text{CASU}}$ 、 $\overline{\text{CASL}}$	H	H	H	H
DQMUU	H	H	H	H
DQMUL	H	H	H	H
DQMLU	H	H	H	H
DQMLL	H	H	H	H
$\overline{\text{AH}}$	L	L	L	L
$\overline{\text{FRAME}}$	H	H	H	H
RDWR	R	H	H	H
	W	L	L	L
$\overline{\text{RD}}$	R	L	L	L
	W	H	H	H
$\overline{\text{ICIORD}}$	R	H	H	H
	W	H	H	H
$\overline{\text{WRHH}}$	R	H	H	H
	W	H	H	H
$\overline{\text{WRHL}}$	R	H	H	H
	W	H	H	H
$\overline{\text{WRH}}$	R	H	H	H
	W	H	L	L
$\overline{\text{WRL}}$	R	H	H	H
	W	L	H	L
$\overline{\text{WE}}$	R	H	H	H
	W	H	H	H
$\overline{\text{ICIOWR}}$	R	H	H	H
	W	H	H	H
A29 ~ A0	地址	地址	地址	地址
D31 ~ D24	High-Z	High-Z	High-Z	High-Z
D23 ~ D16	High-Z	High-Z	High-Z	High-Z
D15 ~ D8	High-Z	数据	High-Z	数据
D7 ~ D0	数据	High-Z	数据	数据

【符号说明】 R: 读 W: 写

有效: 对应存取区域的片选信号 =L, 其他的片选信号 =H

表 B.1 总线相关信号的引脚状态 (3)

引脚名		外部空间 (通常空间)						
		32 位空间						
		最高位字节	第 2 字节	第 3 字节	最低位字节	高位字	低位字	长字
$\overline{\text{CS0}} \sim \overline{\text{CS8}}$		有效	有效	有效	有效	有效	有效	有效
$\overline{\text{CE1A}}$ 、 $\overline{\text{CE1B}}$ 、 $\overline{\text{CE2A}}$ 、 $\overline{\text{CE2B}}$		H	H	H	H	H	H	H
$\overline{\text{BS}}$		L	L	L	L	L	L	L
$\overline{\text{RASU}}$ 、 $\overline{\text{RASL}}$		H	H	H	H	H	H	H
$\overline{\text{CASU}}$ 、 $\overline{\text{CASL}}$		H	H	H	H	H	H	H
DQMUU		H	H	H	H	H	H	H
DQMUL		H	H	H	H	H	H	H
DQMLU		H	H	H	H	H	H	H
DQMLL		H	H	H	H	H	H	H
$\overline{\text{AH}}$		L	L	L	L	L	L	L
$\overline{\text{FRAME}}$		H	H	H	H	H	H	H
RDWR	R	H	H	H	H	H	H	H
	W	L	L	L	L	L	L	L
$\overline{\text{RD}}$	R	L	L	L	L	L	L	L
	W	H	H	H	H	H	H	H
$\overline{\text{ICIORD}}$	R	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H
$\overline{\text{WRHH}}$	R	H	H	H	H	H	H	H
	W	L	H	H	H	L	H	L
$\overline{\text{WRHL}}$	R	H	H	H	H	H	H	H
	W	H	L	H	H	L	H	L
$\overline{\text{WRH}}$	R	H	H	H	H	H	H	H
	W	H	H	L	H	H	L	L
$\overline{\text{WRL}}$	R	H	H	H	H	H	H	H
	W	H	H	H	L	H	L	L
$\overline{\text{WE}}$	R	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H
$\overline{\text{ICIOWR}}$	R	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H
A29 ~ A0		地址	地址	地址	地址	地址	地址	地址
D31 ~ D24		数据	High-Z	High-Z	High-Z	数据	High-Z	数据
D23 ~ D16		High-Z	数据	High-Z	High-Z	数据	High-Z	数据
D15 ~ D8		High-Z	High-Z	数据	High-Z	High-Z	数据	数据
D7 ~ D0		High-Z	High-Z	High-Z	数据	High-Z	数据	数据

【符号说明】 R: 读 W: 写

有效: 对应存取区域的片选信号=L, 其他的片选信号=H

表 B.1 总线相关信号的引脚状态 (4)

引脚名	外部空间 (带字节选择的 SRAM)		
	16 位空间		
	高位字节	低位字节	字 / 长字
$\overline{\text{CS0}} \sim \overline{\text{CS8}}$		有效	有效
$\overline{\text{CE1A}}$ 、 $\overline{\text{CE1B}}$ 、 $\overline{\text{CE2A}}$ 、 $\overline{\text{CE2B}}$		H	H
$\overline{\text{BS}}$		L	L
$\overline{\text{RASU}}$ 、 $\overline{\text{RASL}}$		H	H
$\overline{\text{CASU}}$ 、 $\overline{\text{CASL}}$		H	H
DQMUU		H	H
DQMUL		H	H
DQMLU		H	H
DQMLL		H	H
$\overline{\text{AH}}$		L	L
$\overline{\text{FRAME}}$		H	H
RDWR	R	H	H
	W	L	L
$\overline{\text{RD}}$	R	L	L
	W	H	H
$\overline{\text{ICIORD}}$	R	H	H
	W	H	H
$\overline{\text{WRHH}}$	R	H	H
	W	H	H
$\overline{\text{WRHL}}$	R	H	H
	W	H	H
$\overline{\text{WRH}}$	R	L	L
	W	L	L
$\overline{\text{WRL}}$	R	H	L
	W	H	L
$\overline{\text{WE}}$	R	H	H
	W	H	H
$\overline{\text{ICIOWR}}$	R	H	H
	W	H	H
A29 ~ A0		地址	地址
D31 ~ D24		High-Z	High-Z
D23 ~ D16		High-Z	High-Z
D15 ~ D8		数据	数据
D7 ~ D0		High-Z	数据

【符号说明】 R: 读 W: 写

有效: 对应存取区域的片选信号 =L, 其他的片选信号 =H

表 B.1 总线相关信号的引脚状态 (5)

引脚名	外部空间 (带字节选择的 SRAM)						
	32 位空间						
	最高位字节	第 2 字节	第 3 字节	最低位字节	高位字	低位字	长字
$\overline{CS0} \sim \overline{CS8}$	有效	有效	有效	有效	有效	有效	有效
$\overline{CE1A}$ 、 $\overline{CE1B}$ 、 $\overline{CE2A}$ 、 $\overline{CE2B}$	H	H	H	H	H	H	H
$\overline{BS}$	L	L	L	L	L	L	L
$\overline{RASU}$ 、 $\overline{RASL}$	H	H	H	H	H	H	H
$\overline{CASU}$ 、 $\overline{CASL}$	H	H	H	H	H	H	H
DQMUU	H	H	H	H	H	H	H
DQMUL	H	H	H	H	H	H	H
DQMLU	H	H	H	H	H	H	H
DQMLL	H	H	H	H	H	H	H
$\overline{AH}$	L	L	L	L	L	L	L
$\overline{FRAME}$	H	H	H	H	H	H	H
RDWR	R	H	H	H	H	H	H
	W	L	L	L	L	L	L
$\overline{RD}$	R	L	L	L	L	L	L
	W	H	H	H	H	H	H
$\overline{ICIORD}$	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
$\overline{WRHH}$	R	L	H	H	H	L	L
	W	L	H	H	H	L	L
$\overline{WRHL}$	R	H	L	H	H	L	L
	W	H	L	H	H	L	L
$\overline{WRH}$	R	H	H	L	H	H	L
	W	H	H	L	H	H	L
$\overline{WRL}$	R	H	H	H	L	H	L
	W	H	H	H	L	H	L
$\overline{WE}$	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
$\overline{ICIOWR}$	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
A29 ~ A0	地址	地址	地址	地址	地址	地址	地址
D31 ~ D24	数据	High-Z	High-Z	High-Z	数据	High-Z	数据
D23 ~ D16	High-Z	数据	High-Z	High-Z	数据	High-Z	数据
D15 ~ D8	High-Z	High-Z	数据	High-Z	High-Z	数据	数据
D7 ~ D0	High-Z	High-Z	High-Z	数据	High-Z	数据	数据

【符号说明】 R: 读 W: 写

有效: 对应存取区域的片选信号=L, 其他的片选信号=H

表 B.1 总线相关信号的引脚状态 (6)

引脚名	外部空间 (突发 ROM (时钟异步))			
	8 位空间	16 位空间		
		高位字节	低位字节	字 / 长字
$\overline{CS0} \sim \overline{CS8}$	有效	有效	有效	有效
$\overline{CE1A}$ 、 $\overline{CE1B}$ 、 $\overline{CE2A}$ 、 $\overline{CE2B}$	H	H	H	H
$\overline{BS}$	L	L	L	L
$\overline{RASU}$ 、 $\overline{RASL}$	H	H	H	H
$\overline{CASU}$ 、 $\overline{CASL}$	H	H	H	H
DQMUU	H	H	H	H
DQMUL	H	H	H	H
DQMLU	H	H	H	H
DQMLL	H	H	H	H
$\overline{AH}$	L	L	L	L
$\overline{FRAME}$	H	H	H	H
$\overline{RDWR}$	R	H	H	H
	W	—	—	—
$\overline{RD}$	R	L	L	L
	W	—	—	—
$\overline{ICIORD}$	R	H	H	H
	W	—	—	—
$\overline{WRHH}$	R	H	H	H
	W	—	—	—
$\overline{WRHL}$	R	H	H	H
	W	—	—	—
$\overline{WRH}$	R	H	H	H
	W	—	—	—
$\overline{WRL}$	R	H	H	H
	W	—	—	—
$\overline{WE}$	R	H	H	H
	W	—	—	—
$\overline{ICIOWR}$	R	H	H	H
	W	—	—	—
A29 ~ A0	地址	地址	地址	地址
D31 ~ D24	High-Z	High-Z	High-Z	High-Z
D23 ~ D16	High-Z	High-Z	High-Z	High-Z
D15 ~ D8	High-Z	数据	High-Z	数据
D7 ~ D0	数据	High-Z	数据	数据

【符号说明】 R: 读 W: 写

有效: 对应存取区域的片选信号 =L, 其他的片选信号 =H

表 B.1 总线相关信号的引脚状态 (7)

引脚名	外部空间 (突发 ROM (时钟异步))						
	32 位空间						
	最高位字节	第 2 字节	第 3 字节	最低位字节	高位字	低位字	长字
$\overline{CS0} \sim \overline{CS8}$	有效	有效	有效	有效	有效	有效	有效
$\overline{CE1A}$ 、 $\overline{CE1B}$ 、 $\overline{CE2A}$ 、 $\overline{CE2B}$	H	H	H	H	H	H	H
$\overline{BS}$	L	L	L	L	L	L	L
$\overline{RASU}$ 、 $\overline{RASL}$	H	H	H	H	H	H	H
$\overline{CASU}$ 、 $\overline{CASL}$	H	H	H	H	H	H	H
DQMUU	H	H	H	H	H	H	H
DQMUL	H	H	H	H	H	H	H
DQMLU	H	H	H	H	H	H	H
DQMLL	H	H	H	H	H	H	H
$\overline{AH}$	L	L	L	L	L	L	L
$\overline{FRAME}$	H	H	H	H	H	H	H
RDWR	R	H	H	H	H	H	H
	W	—	—	—	—	—	—
$\overline{RD}$	R	L	L	L	L	L	L
	W	—	—	—	—	—	—
$\overline{ICIORD}$	R	H	H	H	H	H	H
	W	—	—	—	—	—	—
$\overline{WRHH}$	R	H	H	H	H	H	H
	W	—	—	—	—	—	—
$\overline{WRHL}$	R	H	H	H	H	H	H
	W	—	—	—	—	—	—
$\overline{WRH}$	R	H	H	H	H	H	H
	W	—	—	—	—	—	—
$\overline{WRL}$	R	H	H	H	H	H	H
	W	—	—	—	—	—	—
$\overline{WE}$	R	H	H	H	H	H	H
	W	—	—	—	—	—	—
$\overline{ICIOWR}$	R	H	H	H	H	H	H
	W	—	—	—	—	—	—
A29 ~ A0	地址	地址	地址	地址	地址	地址	地址
D31 ~ D24	数据	High-Z	High-Z	High-Z	数据	High-Z	数据
D23 ~ D16	High-Z	数据	High-Z	High-Z	数据	High-Z	数据
D15 ~ D8	High-Z	High-Z	数据	High-Z	High-Z	数据	数据
D7 ~ D0	High-Z	High-Z	High-Z	数据	High-Z	数据	数据

【符号说明】 R: 读 W: 写

有效: 对应存取区域的片选信号=L, 其他的片选信号=H

表 B.1 总线相关信号的引脚状态 (8)

引脚名	外部空间 (突发 ROM (时钟同步))			
	16 位空间			
	高位字节	低位字节	字 / 长字	
$\overline{CS0} \sim \overline{CS8}$		有效	有效	有效
$\overline{CE1A}$ 、 $\overline{CE1B}$ 、 $\overline{CE2A}$ 、 $\overline{CE2B}$		H	H	H
$\overline{BS}$		L	L	L
$\overline{RASU}$ 、 $\overline{RASL}$		H	H	H
$\overline{CASU}$ 、 $\overline{CASL}$		H	H	H
DQMUU		H	H	H
DQMUL		H	H	H
DQMLU		H	H	H
DQMLL		H	H	H
$\overline{AH}$		L	L	L
$\overline{FRAME}$		H	H	H
RDWR	R	H	H	H
	W	—	—	—
$\overline{RD}$	R	L	L	L
	W	—	—	—
$\overline{ICIORD}$	R	H	H	H
	W	—	—	—
$\overline{WRHH}$	R	H	H	H
	W	—	—	—
$\overline{WRHL}$	R	H	H	H
	W	—	—	—
$\overline{WRH}$	R	H	H	H
	W	—	—	—
$\overline{WRL}$	R	H	H	H
	W	—	—	—
$\overline{WE}$	R	H	H	H
	W	—	—	—
$\overline{ICIOWR}$	R	H	H	H
	W	—	—	—
A29 ~ A0		地址	地址	地址
D31 ~ D24		High-Z	High-Z	High-Z
D23 ~ D16		High-Z	High-Z	High-Z
D15 ~ D8		数据	High-Z	数据
D7 ~ D0		High-Z	数据	数据

【符号说明】 R: 读 W: 写

有效: 对应存取区域的片选信号 =L, 其他的片选信号 =H



表 B.1 总线相关信号的引脚状态 (9)

引脚名	外部空间 (突发 ROM (时钟同步))						
	32 位空间						
	最高位字节	第 2 字节	第 3 字节	最低位字节	高位字	低位字	长字
$\overline{CS0} \sim \overline{CS8}$	有效	有效	有效	有效	有效	有效	有效
$\overline{CE1A}$ 、 $\overline{CE1B}$ 、 $\overline{CE2A}$ 、 $\overline{CE2B}$	H	H	H	H	H	H	H
$\overline{BS}$	L	L	L	L	L	L	L
$\overline{RASU}$ 、 $\overline{RASL}$	H	H	H	H	H	H	H
$\overline{CASU}$ 、 $\overline{CASL}$	H	H	H	H	H	H	H
DQMUU	H	H	H	H	H	H	H
DQMUL	H	H	H	H	H	H	H
DQMLU	H	H	H	H	H	H	H
DQMLL	H	H	H	H	H	H	H
$\overline{AH}$	L	L	L	L	L	L	L
$\overline{FRAME}$	H	H	H	H	H	H	H
RDWR	R	H	H	H	H	H	H
	W	—	—	—	—	—	—
$\overline{RD}$	R	L	L	L	L	L	L
	W	—	—	—	—	—	—
$\overline{ICIORD}$	R	H	H	H	H	H	H
	W	—	—	—	—	—	—
$\overline{WRHH}$	R	H	H	H	H	H	H
	W	—	—	—	—	—	—
$\overline{WRHL}$	R	H	H	H	H	H	H
	W	—	—	—	—	—	—
$\overline{WRH}$	R	H	H	H	H	H	H
	W	—	—	—	—	—	—
$\overline{WRL}$	R	H	H	H	H	H	H
	W	—	—	—	—	—	—
$\overline{WE}$	R	H	H	H	H	H	H
	W	—	—	—	—	—	—
$\overline{ICIOWR}$	R	H	H	H	H	H	H
	W	—	—	—	—	—	—
A29 ~ A0	地址	地址	地址	地址	地址	地址	地址
D31 ~ D24	数据	High-Z	High-Z	High-Z	数据	High-Z	数据
D23 ~ D16	High-Z	数据	High-Z	High-Z	数据	High-Z	数据
D15 ~ D8	High-Z	High-Z	数据	High-Z	High-Z	数据	数据
D7 ~ D0	High-Z	High-Z	High-Z	数据	High-Z	数据	数据

【符号说明】 R: 读 W: 写

有效: 对应存取区域的片选信号=L, 其他的片选信号=H

表 B.1 总线相关信号的引脚状态 (10)

引脚名	外部空间 (SDRAM)			
	16 位空间			
	高位字节	低位字节	字 / 长字	
$\overline{CS0} \sim \overline{CS8}$		有效 *1	有效 *1	有效 *1
$\overline{CE1A}$ 、 $\overline{CE1B}$ 、 $\overline{CE2A}$ 、 $\overline{CE2B}$		H	H	H
$\overline{BS}$		L	L	L
$\overline{RASU}$ 、 $\overline{RASL}$		有效 *2	有效 *2	有效 *2
$\overline{CASU}$ 、 $\overline{CASL}$		有效 *2	有效 *2	有效 *2
DQMUU		H	H	H
DQMUL		H	H	H
DQMLU		L	H	L
DQMLL		H	L	L
$\overline{AH}$		L	L	L
$\overline{FRAME}$		H	H	H
RDWR	R	H	H	H
	W	L	L	L
RD	R	H	H	H
	W	H	H	H
$\overline{ICIORD}$	R	H	H	H
	W	H	H	H
$\overline{WRHH}$	R	H	H	H
	W	H	H	H
$\overline{WRHL}$	R	H	H	H
	W	H	H	H
$\overline{WRH}$	R	H	H	H
	W	H	H	H
$\overline{WRL}$	R	H	H	H
	W	H	H	H
$\overline{WE}$	R	H	H	H
	W	H	H	H
$\overline{ICIOWR}$	R	H	H	H
	W	H	H	H
A29 ~ A0		地址	地址	地址
D31 ~ D24		High-Z	High-Z	High-Z
D23 ~ D16		High-Z	High-Z	High-Z
D15 ~ D8		数据	High-Z	数据
D7 ~ D0		High-Z	数据	数据

【符号说明】 R: 读 W: 写

【注】 \*1 对应存取区域的片选信号 =L, 其他的片选信号 =H

\*2 存取 A25=0 的地址时  $\overline{RASL}/\overline{CASL}$ =L, 存取 A25=1 的地址时  $\overline{RASU}/\overline{CASU}$ =L

表 B.1 总线相关信号的引脚状态 (11)

引脚名		外部空间 (SDRAM)						
		32 位空间						
		最高位字节	第 2 字节	第 3 字节	最低位字节	高位字	低位字	长字
$\overline{CS0} \sim \overline{CS8}$		有效 *1	有效 *1	有效 *1	有效 *1	有效 *1	有效 *1	有效 *1
$\overline{CE1A}$ 、 $\overline{CE1B}$ 、 $\overline{CE2A}$ 、 $\overline{CE2B}$		H	H	H	H	H	H	H
$\overline{BS}$		L	L	L	L	L	L	L
$\overline{RASU}$ 、 $\overline{RASL}$		有效 *2	有效 *2	有效 *2	有效 *2	有效 *2	有效 *2	有效 *2
$\overline{CASU}$ 、 $\overline{CASL}$		有效 *2	有效 *2	有效 *2	有效 *2	有效 *2	有效 *2	有效 *2
DQMUU		L	H	H	H	L	H	L
DQMUL		H	L	H	H	L	H	L
DQMLU		H	H	L	H	H	L	L
DQMLL		H	H	H	L	H	L	L
$\overline{AH}$		L	L	L	L	L	L	L
$\overline{FRAME}$		H	H	H	H	H	H	H
RDWR	R	H	H	H	H	H	H	H
	W	L	L	L	L	L	L	L
$\overline{RD}$	R	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H
$\overline{ICIORD}$	R	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H
$\overline{WRHH}$	R	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H
$\overline{WRHL}$	R	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H
$\overline{WRH}$	R	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H
$\overline{WRL}$	R	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H
$\overline{WE}$	R	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H
$\overline{ICIOWR}$	R	H	H	H	H	H	H	H
	W	H	H	H	H	H	H	H
A29 ~ A0		地址	地址	地址	地址	地址	地址	地址
D31 ~ D24		数据	High-Z	High-Z	High-Z	数据	High-Z	数据
D23 ~ D16		High-Z	数据	High-Z	High-Z	数据	High-Z	数据
D15 ~ D8		High-Z	High-Z	数据	High-Z	High-Z	数据	数据
D7 ~ D0		High-Z	High-Z	High-Z	数据	High-Z	数据	数据

【符号说明】 R: 读 W: 写

【注】 \*1 对应存取区域的片选信号 =L, 其他的片选信号 =H

\*2 存取 A25=0 的地址时  $\overline{RASL}/\overline{CASL}=L$ , 存取 A25=1 的地址时  $\overline{RASU}/\overline{CASU}=L$

表 B.1 总线相关信号的引脚状态 (12)

引脚名	外部空间 (MPX-I/O)				
	8 位空间	16 位空间			
		高位字节	低位字节	字 / 长字	
$\overline{CS0} \sim \overline{CS8}$		有效	有效	有效	有效
$\overline{CE1A}$ 、 $\overline{CE1B}$ 、 $\overline{CE2A}$ 、 $\overline{CE2B}$		H	H	H	H
$\overline{BS}$		L	L	L	L
$\overline{RASU}$ 、 $\overline{RASL}$		H	H	H	H
$\overline{CASU}$ 、 $\overline{CASL}$		H	H	H	H
$\overline{DQMUU}$		H	H	H	H
$\overline{DQMUL}$		H	H	H	H
$\overline{DQMLU}$		H	H	H	H
$\overline{DQMLL}$		H	H	H	H
$\overline{AH}$		H	H	H	H
$\overline{FRAME}$		H	H	H	H
$\overline{RDWR}$	R	H	H	H	H
	W	L	L	L	L
$\overline{RD}$	R	L	L	L	L
	W	H	H	H	H
$\overline{ICIORD}$	R	H	H	H	H
	W	H	H	H	H
$\overline{WRHH}$	R	H	H	H	H
	W	H	H	H	H
$\overline{WRHL}$	R	H	H	H	H
	W	H	H	H	H
$\overline{WRH}$	R	H	H	H	H
	W	H	L	H	L
$\overline{WRL}$	R	H	H	H	H
	W	L	H	L	L
$\overline{WE}$	R	H	H	H	H
	W	H	H	H	H
$\overline{ICIOWR}$	R	H	H	H	H
	W	H	H	H	H
A29 ~ A0		地址	地址	地址	地址
D31 ~ D24		High-Z	High-Z	High-Z	High-Z
D23 ~ D16		High-Z	High-Z	High-Z	High-Z
D15 ~ D8		High-Z	地址 / 数据	地址	地址 / 数据
D7 ~ D0		地址 / 数据	地址	地址 / 数据	地址 / 数据

【符号说明】 R: 读 W: 写

有效: 对应存取区域的片选信号 =L, 其他的片选信号 =H

表 B.1 总线相关信号的引脚状态 (13)

引脚名	外部空间 (突发 MPX-I/O)						
	32 位空间						
	最高位字节	第 2 字节	第 3 字节	最低位字节	高位字	低位字	长字
$\overline{\text{CS0}} \sim \overline{\text{CS8}}$	有效	有效	有效	有效	有效	有效	有效
$\overline{\text{CE1A}}$ 、 $\overline{\text{CE1B}}$ 、 $\overline{\text{CE2A}}$ 、 $\overline{\text{CE2B}}$	H	H	H	H	H	H	H
$\overline{\text{BS}}$	L	L	L	L	L	L	L
$\overline{\text{RASU}}$ 、 $\overline{\text{RASL}}$	H	H	H	H	H	H	H
$\overline{\text{CASU}}$ 、 $\overline{\text{CASL}}$	H	H	H	H	H	H	H
$\overline{\text{DQMUU}}$	H	H	H	H	H	H	H
$\overline{\text{DQMUL}}$	H	H	H	H	H	H	H
$\overline{\text{DQMLU}}$	H	H	H	H	H	H	H
$\overline{\text{DQMLL}}$	H	H	H	H	H	H	H
$\overline{\text{AH}}$	L	L	L	L	L	L	L
$\overline{\text{FRAME}}$	L	L	L	L	L	L	L
$\overline{\text{RDWR}}$	R	H	H	H	H	H	H
	W	L	L	L	L	L	L
$\overline{\text{RD}}$	R	L	L	L	L	L	L
	W	H	H	H	H	H	H
$\overline{\text{ICIORD}}$	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
$\overline{\text{WRHH}}$	R	H	H	H	H	H	H
	W	L	H	H	H	L	H
$\overline{\text{WRHL}}$	R	H	H	H	H	H	H
	W	H	L	H	H	L	H
$\overline{\text{WRH}}$	R	H	H	H	H	H	H
	W	H	H	L	H	H	L
$\overline{\text{WRL}}$	R	H	H	H	H	H	H
	W	H	H	H	L	H	L
$\overline{\text{WE}}$	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
$\overline{\text{ICIOWR}}$	R	H	H	H	H	H	H
	W	H	H	H	H	H	H
A29 ~ A0	地址	地址	地址	地址	地址	地址	地址
D31 ~ D24	地址 / 数据	地址	地址	地址	地址 / 数据	地址	地址 / 数据
D23 ~ D16	地址	地址 / 数据	地址	地址	地址 / 数据	地址	地址 / 数据
D15 ~ D8	地址	地址	地址 / 数据	地址	地址	地址 / 数据	地址 / 数据
D7 ~ D0	地址	地址	地址	地址 / 数据	地址	地址 / 数据	地址 / 数据

【符号说明】 R: 读 W: 写

有效: 对应存取区域的片选信号 =L, 其他的片选信号 =H

表 B.1 总线相关信号的引脚状态 (14)

引脚名	外部空间 (PCMCIA 存储器卡接口)			
	8 位空间	16 位空间		
		高位字节	低位字节	字 / 长字
$\overline{CS0} \sim \overline{CS8}$	H	H	H	H
$\overline{CE1A}$ 、 $\overline{CE1B}$ 、 $\overline{CE2A}$ 、 $\overline{CE2B}$	有效	有效	有效	有效
$\overline{BS}$	L	L	L	L
$\overline{RASU}$ 、 $\overline{RASL}$	H	H	H	H
$\overline{CASU}$ 、 $\overline{CASL}$	H	H	H	H
$\overline{DQMUU}$	H	H	H	H
$\overline{DQMUL}$	H	H	H	H
$\overline{DQMLU}$	H	H	H	H
$\overline{DQMLL}$	H	H	H	H
$\overline{AH}$	L	L	L	L
$\overline{FRAME}$	H	H	H	H
$\overline{RDWR}$	R	H	H	H
	W	L	L	L
$\overline{RD}$	R	L	L	L
	W	H	H	H
$\overline{ICIORD}$	R	H	H	H
	W	H	H	H
$\overline{WRHH}$	R	H	H	H
	W	H	H	H
$\overline{WRHL}$	R	H	H	H
	W	H	H	H
$\overline{WRH}$	R	H	H	H
	W	H	H	H
$\overline{WRL}$	R	H	H	H
	W	H	H	H
$\overline{WE}$	R	H	H	H
	W	L	L	L
$\overline{ICIOWR}$	R	H	H	H
	W	H	H	H
A29 ~ A0	地址	地址	地址	地址
D31 ~ D24	High-Z	High-Z	High-Z	High-Z
D23 ~ D16	High-Z	High-Z	High-Z	High-Z
D15 ~ D8	High-Z	数据	High-Z	数据
D7 ~ D0	数据	High-Z	数据	数据

【符号说明】 R: 读 W: 写

有效: 对应存取区域的卡允许信号 =L、其他的卡允许信号 =H

表 B.1 总线相关信号的引脚状态 (15)

引脚名	外部空间 (PCMCIA I/O 卡接口)			
	8 位空间	16 位空间		
		高位字节	低位字节	字 / 长字
$\overline{CS0} \sim \overline{CS8}$	H	H	H	H
$\overline{CE1A}$ 、 $\overline{CE1B}$ 、 $\overline{CE2A}$ 、 $\overline{CE2B}$	有效	有效	有效	有效
$\overline{BS}$	L	L	L	L
$\overline{RASU}$ 、 $\overline{RASL}$	H	H	H	H
$\overline{CASU}$ 、 $\overline{CASL}$	H	H	H	H
$\overline{DQMUU}$	H	H	H	H
$\overline{DQMUL}$	H	H	H	H
$\overline{DQMLU}$	H	H	H	H
$\overline{DQMLL}$	H	H	H	H
$\overline{AH}$	L	L	L	L
$\overline{FRAME}$	H	H	H	H
$\overline{RDWR}$	R	H	H	H
	W	L	L	L
$\overline{RD}$	R	H	H	H
	W	H	H	H
$\overline{ICIORD}$	R	L	L	L
	W	H	H	H
$\overline{WRHH}$	R	H	H	H
	W	H	H	H
$\overline{WRHL}$	R	H	H	H
	W	H	H	H
$\overline{WRH}$	R	H	H	H
	W	H	H	H
$\overline{WRL}$	R	H	H	H
	W	H	H	H
$\overline{WE}$	R	H	H	H
	W	H	H	H
$\overline{ICIOWR}$	R	H	H	H
	W	L	L	L
A29 ~ A0	地址	地址	地址	地址
D31 ~ D24	High-Z	High-Z	High-Z	High-Z
D23 ~ D16	High-Z	High-Z	High-Z	High-Z
D15 ~ D8	High-Z	数据	High-Z	数据
D7 ~ D0	数据	High-Z	数据	数据

【符号说明】 R: 读 W: 写

有效: 对应存取区域的卡允许信号 =L, 其他的卡允许信号 =H

## 附录 C. 型号一览

表 C.1 型号一览

产品名	产品分类					产品型号	封装 (封装代码)
	分类	ROM 容量	RAM 容量	用途	工作温度		
SH7083	F-ZTAT 版	256KB	16KB	民用	-20 ~ +85°C	R5F70834AN80FTV	TQFP1414-100 (TFP-100BV)
				工业用	-40 ~ +85°C	R5F70834AD80FTV	
		512KB	32KB	民用	-20 ~ +85°C	R5F70835AN80FTV	
				工业用	-40 ~ +85°C	R5F70835AD80FTV	
		256KB	16KB	民用	-20 ~ +85°C	R5F70834AN80BGV	P-LFBGA-112 (BP-112V)
				工业用	-40 ~ +85°C	R5F70834AD80BGV	
	512KB	32KB	民用	-20 ~ +85°C	R5F70834AN80BGV		
			工业用	-40 ~ +85°C	R5F70834AD80BGV		
	掩模型 ROM 版	256KB	16KB	民用	-20 ~ +85°C	R5M70834ANXXXFTV* <sup>2</sup>	TQFP1414-100 (TFP-100BV)
				工业用	-40 ~ +85°C	R5M70834ADXXXFTV* <sup>2</sup>	
		256KB	16KB	民用	-20 ~ +85°C	R5M70834ANXXXBGV* <sup>2</sup>	P-LFBGA-112 (BP-112V)
				工业用	-40 ~ +85°C	R5M70834ADXXXBGV* <sup>2</sup>	
	无 ROM 版	0KB	16KB	民用	-20 ~ +85°C	R5S70830AN80FTV	TQFP1414-100 (TFP-100BV)
				工业用	-40 ~ +85°C	R5S70830AD80FTV	
0KB		16KB	民用	-20 ~ +85°C	R5S70830AN80BGV	P-LFBGA-112 (BP-112V)	
			工业用	-40 ~ +85°C	R5S70830AD80BGV		
对应 E10A 全功能的 F-ZTAT 版* <sup>1</sup>	512KB	32KB	系统开发 专用* <sup>1</sup>	0 ~ +50°C	R5E70835RN80FTV	TQFP1414-100 (TFP-100BV)	
SH7084	F-ZTAT 版	256KB	16KB	民用	-20 ~ +85°C	R5F70844AN80FPV	LQFP2020-112 (FP-112EV)
				工业用	-40 ~ +85°C	R5F70844AD80FPV	
		512KB	32KB	民用	-20 ~ +85°C	R5F70845AN80FPV	
				工业用	-40 ~ +85°C	R5F70845AD80FPV	
	掩模型 ROM 版	256KB	16KB	民用	-20 ~ +85°C	R5M70844ANXXXFPV* <sup>2</sup>	
				工业用	-40 ~ +85°C	R5M70844ADXXXFPV* <sup>2</sup>	
	无 ROM 版	0KB	16KB	民用	-20 ~ +85°C	R5S70840AN80FPV	
				工业用	-40 ~ +85°C	R5S70840AD80FPV	
	对应 E10A 全功能的 F-ZTAT 版* <sup>1</sup>	512KB	32KB	系统开发 专用* <sup>1</sup>	0 ~ +50°C	R5E70845RN80FPV	



产品分类						产品型号	封装 (封装代码)
产品名	分类	ROM 容量	RAM 容量	用途	工作温度		
SH7085	F-ZTAT 版	256KB	16KB	民用	-20 ~ +85°C	R5F70854AN80FPV	LQFP2020-144 (FP-144LV)
				工业用	-40 ~ +85°C	R5F70854AD80FPV	
		512KB	32KB	民用	-20 ~ +85°C	R5F70855AN80FPV	
				工业用	-40 ~ +85°C	R5F70855AD80FPV	
	掩模型 ROM 版	256KB	16KB	民用	-20 ~ +85°C	R5M70854ANXXXFPV*2	
				工业用	-40 ~ +85°C	R5M70854ADXXXFPV*2	
	无 ROM 版	0KB	16KB	民用	-20 ~ +85°C	R5S70850AN80FPV	
				工业用	-40 ~ +85°C	R5S70850AD80FPV	
对应 E10A 全功能的 F-ZTAT 版 *1	512KB	32KB	系统开发 专用 *1	0 ~ +50°C	R5E70855RN80FPV		
SH7086	F-ZTAT 版	512KB	32KB	民用	-20 ~ +85°C	R5F70865AN80FPV	LQFP2424-176 (FP-176EV)
				工业用	-40 ~ +85°C	R5F70865AD80FPV	
	对应 E10A 全功能的 F-ZTAT 版 *			系统开发 专用 *	0 ~ +50°C	R5E70865RN80FPV	

【注】 \*1 对应 E10A 全功能的 F-ZTAT 版是用户系统开发的专用产品，可以使用 E10A 内部总线跟踪功能和 AUD 功能。但是，在批量生产时，必须使用通常的 F-ZTAT 版或掩模型 ROM 版。另外，通常的 F-ZTAT 版不能使用 E10A 内部总线跟踪功能和 AUD 功能。对应 E10A 全功能的 F-ZTAT 版不保证可靠性。

\*2 XXX 是 ROM 代码。

附录 D. 外形尺寸图

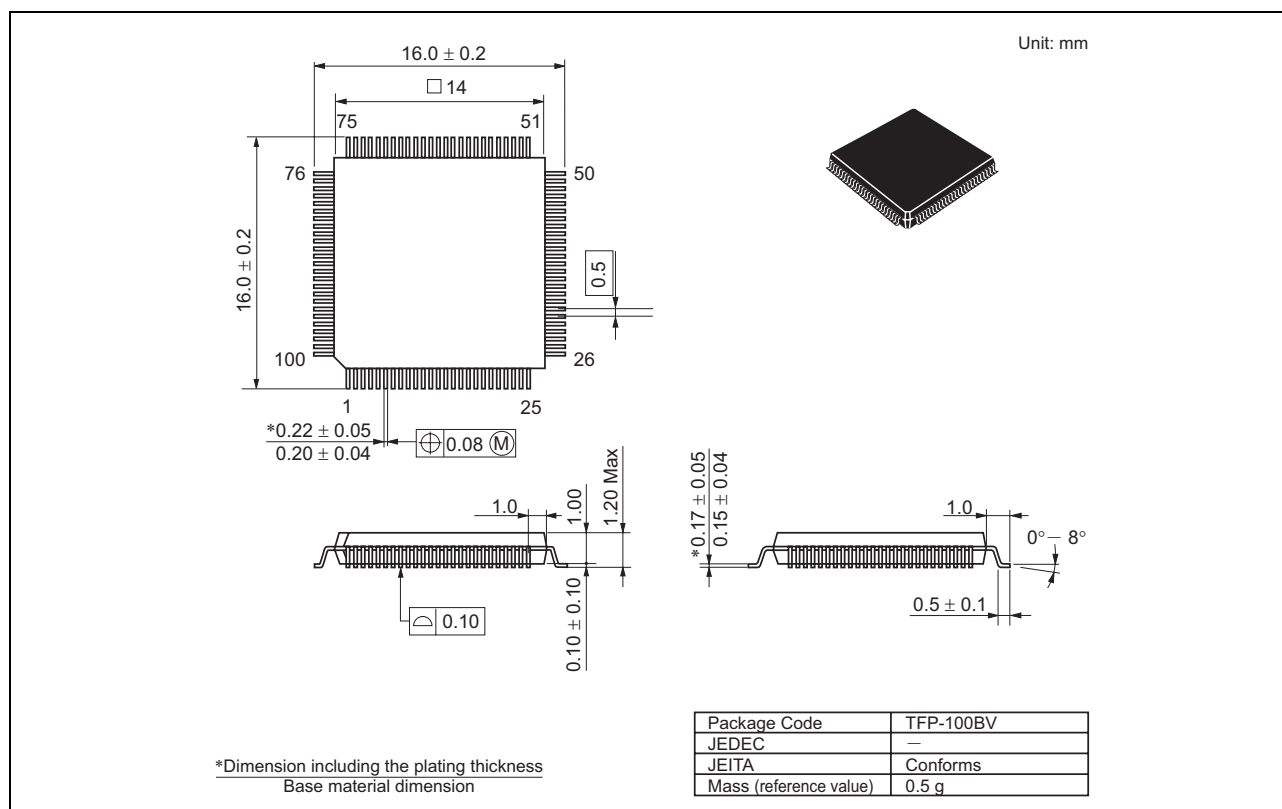


图 D.1 TFP-100BV

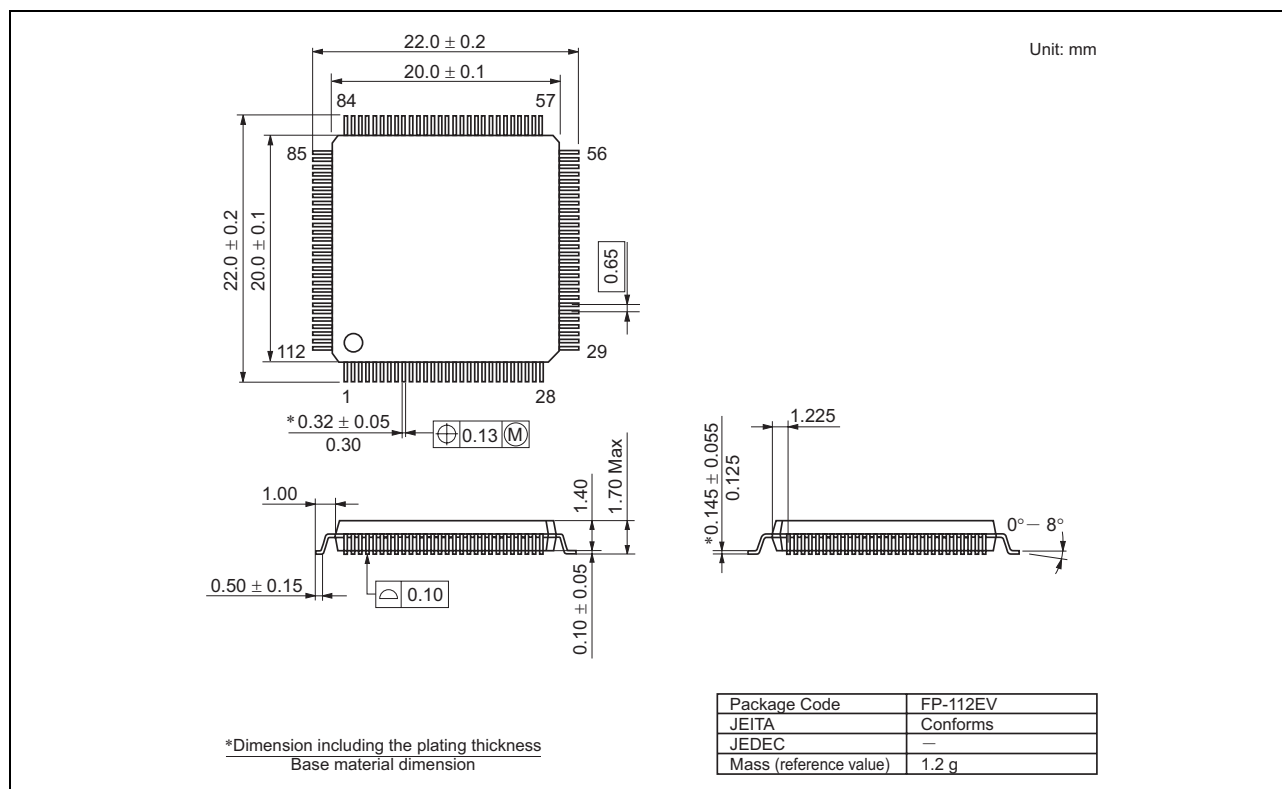


图 D.2 FP-112EV

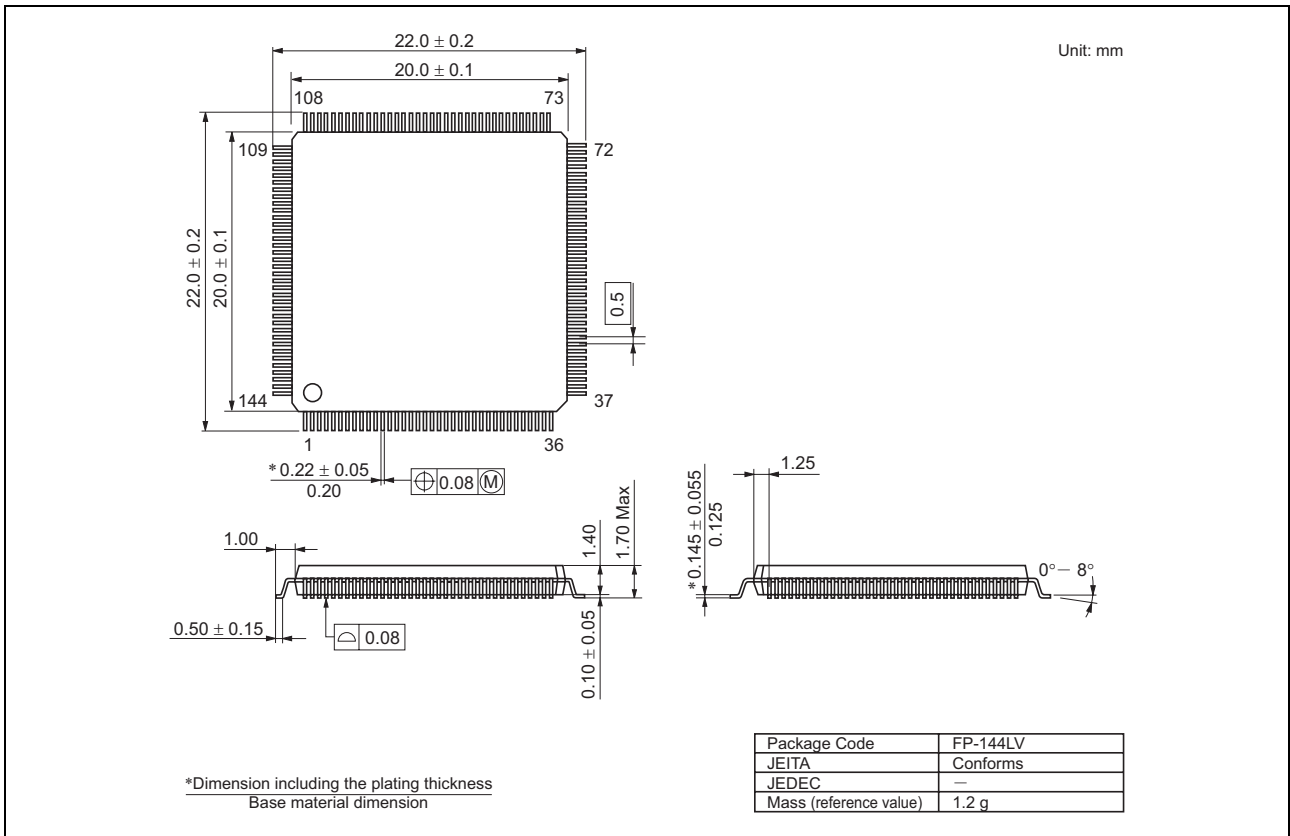


图 D.3 FP-144LV

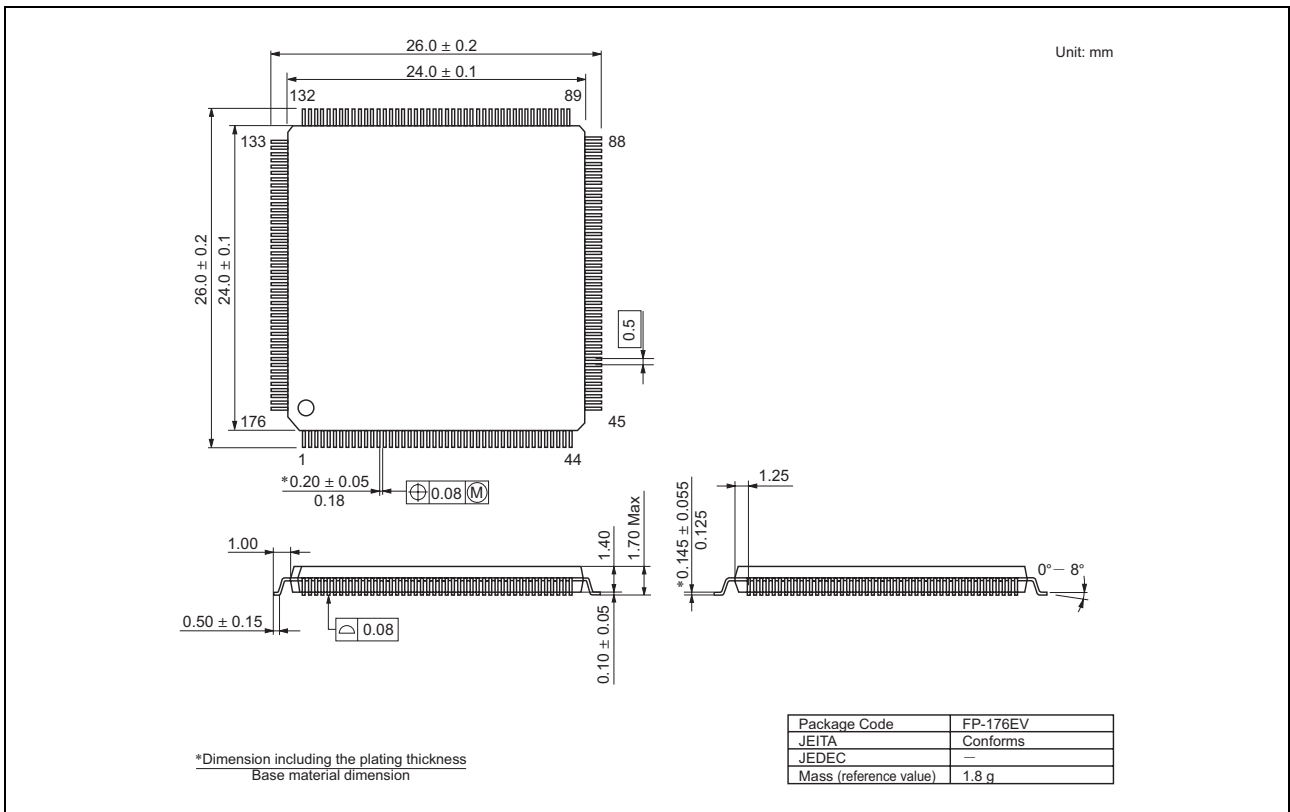


图 D.4 FP-176EV

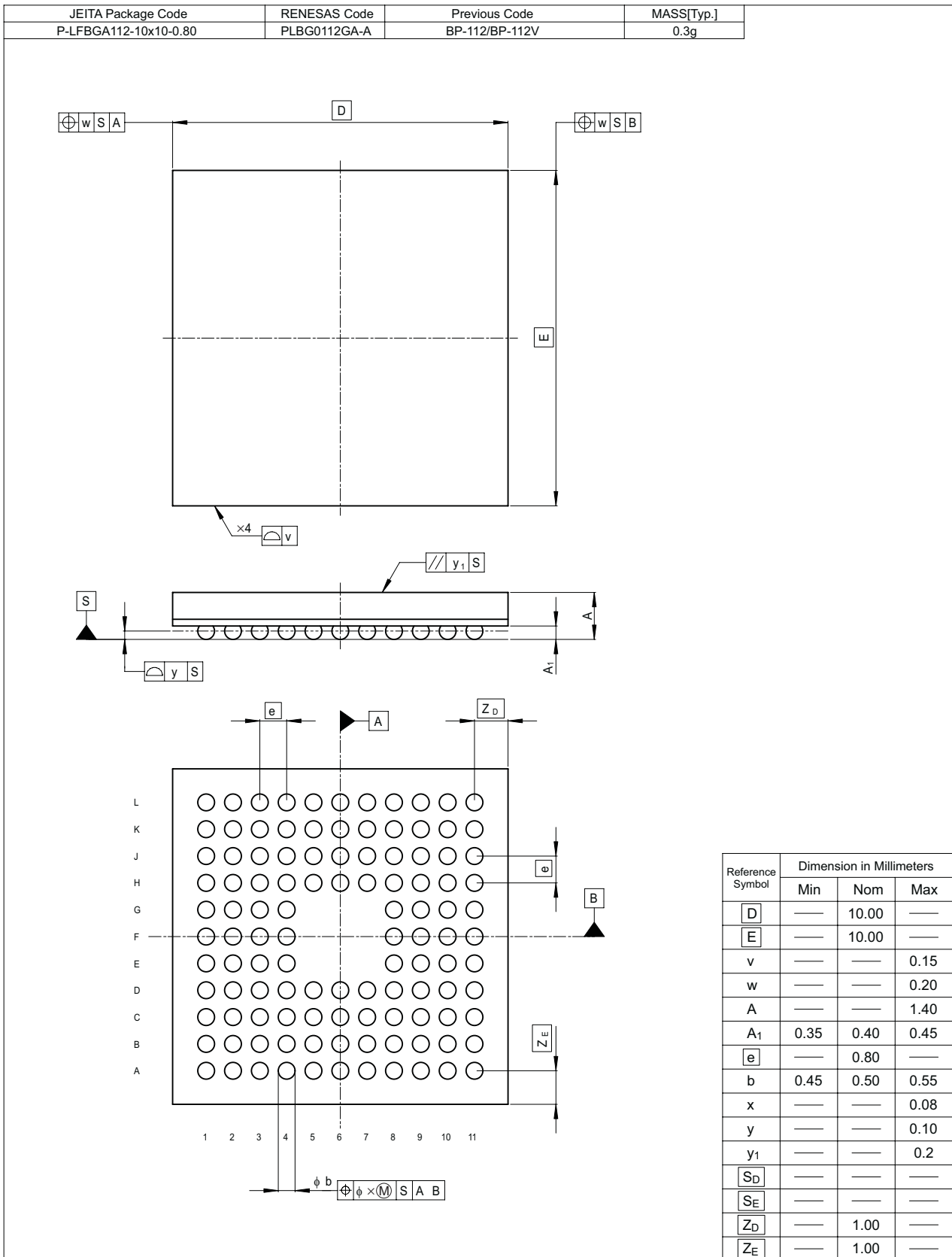


图 D.5 图 D.5BP-112V

# 索引

## A

A/D 转换开始请求的延迟功能 .....	411
A/D 转换器的启动 .....	423
A/D 转换器的中断源 .....	676
A/D 转换器特性 .....	1055
A/D 转换器 (ADC) .....	659
A/D 转换时间 .....	674
AC 特性 .....	1001
AC 特性的测量条件 .....	1055
AC 总线的时序规格 .....	1007

## B

板上编程模式 .....	888
比较匹配定时器 (CMT) .....	681
编程器模式 .....	934

## C

CMT 的中断源 .....	686
CPU .....	18
CSn 有效期间的扩展 .....	202
槽非法指令 .....	72
乘加寄存器 (MACH 和 MACL) .....	21
程序计数器 (PC) .....	21
程序执行状态 .....	39
重复传送模式 .....	135
传送时钟 .....	614
传送信息的分配和 DTC 向量表 .....	127
传送信息的回写省略功能 .....	134
传送信息的跳读功能 .....	133
串行通信接口 (SCI) .....	510
从 LSI 内部总线主控器的存取 .....	252
存储体激活 .....	219
存取长度和数据定位 .....	192
存取等待的控制 .....	200
存取周期之间的等待 .....	240
错误保护 .....	903

## D

DACK 引脚输出的注意事项 .....	286
DC 特性 .....	997
DMA 的传送种类 .....	274
DMA 传送请求 .....	268
DTC/DMAC 的启动 .....	423
DTC 的启动源 .....	126
DTC 的执行状态 .....	139
DTC 的总线权释放时序 .....	141
DTC 向量地址 .....	128
带 FIFO 的串行通信接口 (SCIF) .....	557
带字节选择的 SRAM 接口 .....	228

单次读 .....	216
单次写 .....	218
单地址模式 .....	277
单通道模式 .....	672
单芯片模式 .....	41
单周期扫描模式 .....	672
低功耗模式 .....	937
低功耗状态 .....	39
地址错误 .....	69, 73, 936
地址的多路复用 .....	207
地址映像 .....	151
电路板设计的注意事项 .....	62
断点比较条件 .....	96
端口输出的允许 (POE) .....	484
对绝对精度的影响 .....	679
多处理器通信功能 .....	547
多功能定时器脉冲单元 2S (MTU2S) .....	480
多功能定时器脉冲单元 2 (MTU2) .....	288

## F

非线性误差 .....	677
分频器 .....	52
复位同步 PWM 模式 .....	379
复位状态 .....	39

## G

高阻抗控制对象的引脚及其条件 .....	500
各处理状态的本 LSI 引脚状态 .....	1057
各模块的运行时钟 .....	53
各运行模式的地址映像 .....	42
各运行模式的寄存器状态 .....	984
更改运行模式时的注意事项 .....	49
固定模式 .....	270
过程寄存器 (PR) .....	21

## H

互补 PWM 模式 .....	381
-----------------	-----

## I

I/O 端口 .....	818
I/O 卡接口的基本时序 .....	234
I <sup>2</sup> C 总线格式 .....	642
I <sup>2</sup> C 总线接口 2 (I <sup>2</sup> C2) .....	630
IC 存储卡接口的基本时序 .....	233
IRQ 中断 .....	85

## J

寄存器	
ADCR .....	665
ADCSR .....	663
ADDR0 ~ ADDR15 .....	663

ADTSR .....	668	ICSR1 .....	487
BAMRA .....	99	ICSR2 .....	491
BAMRB .....	102	ICSR3 .....	494
BARA .....	99	IFCR .....	816
BARB .....	102	IPRA ~ IPRF、IPRH ~ IPRM .....	84
BBRA .....	100	IRQCR .....	77
BBRB .....	104	IRQSR .....	79
BDMRA .....	101	MRA .....	121
BDMRB .....	103	MRB .....	122
BDRA .....	101	NF2CYC .....	641
BDRB .....	103	OCSR1 .....	490
BETR .....	107	OCSR2 .....	493
BRCR .....	105	OSCCR .....	59
BRDR .....	109	PACRH1 .....	738
BRSR .....	108	PACRH2 .....	738
BSCEHR .....	126, 189, 265	PACRH3 .....	738
CHCR .....	260	PACRH4 .....	738
CMCNT .....	684	PACRL1 .....	738
CMCOR .....	684	PACRL2 .....	738
CMCSR .....	683	PACRL3 .....	738
CMNCR .....	166	PACRL4 .....	738
CMSTR .....	682	PADRH .....	822
CRA .....	123	PADRL .....	822
CRB .....	124	PAIORH .....	738
CS0BCR ~ CS8BCR .....	167	PAIORL .....	738
CS0WCR ~ CS8WCR .....	170	PAPRH .....	827
DAR (DMAC) .....	259	PAPRL .....	827
DAR (DTC) .....	123	PBCRL1 .....	763
DMAOR .....	264	PBCRL2 .....	763
DMATCR .....	260	PBCRL3 .....	763
DPFR .....	878	PBDRL .....	833
DTCCR .....	125	PBIORL .....	763
DTCERA ~ DTCERE .....	124	PBPRL .....	835
DTCVBR .....	126	PCCRH1 .....	769
FCCS .....	872	PCCRH2 .....	769
FEBS .....	885	PCCRH3 .....	769
FECS .....	874	PCCRL1 .....	769
FKEY .....	875	PCCRL2 .....	769
FMATS .....	875	PCCRL3 .....	769
FMPAR .....	882	PCCRL4 .....	769
FMPDR .....	883	PCDRH .....	838
FPCS .....	874	PCDRL .....	838
FPEFEQ .....	879	PCIORH .....	769
FPFR .....	881, 883, 886	PCIORL .....	769
FRQCR .....	57	PCPRH .....	841
FTDAR .....	876	PCPRL .....	841
FUBRA .....	880	PDCRH1 .....	781
HCPCR .....	815	PDCRH2 .....	781
ICCR1 .....	633	PDCRH3 .....	781
ICCR2 .....	635	PDCRH4 .....	781
ICDRR .....	640	PDCRL1 .....	781
ICDRS .....	641	PDCRL2 .....	781
ICDRT .....	640	PDCRL3 .....	781
ICIER .....	637	PDCRL4 .....	781
ICMR .....	636	PDDRH .....	845
ICR0 .....	77	PDDRL .....	845
ICSR .....	638	PDIORH .....	781

PDIORL	781	SSTDR0 ~ SSTDR3	612
PDPRH	848	SSTRSR	613
PDPRL	848	STBCR1	938
PECRH1	794	STBCR2	939
PECRH2	794	STBCR3	940
PECRL1	794	STBCR4	941
PECRL2	794	STBCR5	942
PECRL3	794	STBCR6	943
PECRL4	794	TADCOBRA_4	332
PEDRH	853	TADCOBRB_4	332
PEDRL	853	TADCORA_4	332
PEIORH	794	TADCORB_4	332
PEIORL	794	TADCR	330
PEPRH	857	TBTER	350
PEPRL	857	TBTM	327
PFDRL	861	TCBR	347
POECR1	496	TCDR	347
POECR2	497	TCNT	332
RAMCR	944	TCNTCMPCLR	319
RAMER	887	TCNTS	346
RTCNT	187	TCR	296
RTCOR	188	TCSYSTR	336
RTCSR	186	TDDR	346
SAR (DMAC)	259	TDER	351
SAR (DTC)	123	TGCR	345
SAR (I2C2)	640	TGR	333
SCBRR (SCIF)	568	TICCR	328
SCBRR (SCI)	523	TIER	320
SCFCR	576	TIOR	301
SCFDR	577	TITCNT	349
SCFRDR	559	TITCR	347
SCFSR	564	TMDR	299
SCFTDR	560	TOCR1	339
SCLSR	580	TOCR2	341
SCRDR	513	TOER	338
SCRSR (SCIF)	559	TOLBR	344
SCRSR (SCI)	513	TRWER	337
SCSCR (SCIF)	562	TSR	323
SCSCR (SCI)	516	TSTR	333
SCSDCR	522	TSYCR	329
SCSMR (SCIF)	561	TSYR	335
SCSMR (SCI)	514	TWCR	351
SCSPTR (SCIF)	578	WTCNT	505
SCSPTR (SCI)	520	WTCSR	506
SCSSR	518	寄存器存取时的注意事项	507
SCTDR	514	寄存器的数据格式	22
SCTSR (SCIF)	560	寄存器地址一览表 (地址顺序)	948
SCTSR (SCI)	513	寄存器位一览	961
SDCR	184	寄存器一览	947
SPOER	495	加电复位	67
SSCR2	611	加电顺序	225
SSCRH	605	间隔定时器模式的用法	509
SSCRL	606	晶体谐振器	52
SSER	608	绝对精度	677
SSMR	607	绝对最大额定值	996
SSRDR0 ~ SSRDR3	613		
SSSR	609		

<b>K</b>	
看门狗定时器模式的用法 .....	508
看门狗定时器 (WDT) .....	504
空载时间的补偿功能 .....	419
控制寄存器的初始值 .....	21
控制信号的时序 .....	1004
块传送模式 .....	136

<b>L</b>	
立即数的数据格式 .....	22
连接晶体谐振器的方法 .....	60
连续扫描模式 .....	672
链传送 .....	137
量化误差 .....	677
逻辑运算指令 .....	35

<b>M</b>	
MCU 扩展模式 .....	41
MCU 运行模式 .....	40
MPX-I/O 接口 .....	203
MTU2 的功能一览表 .....	289
MTU2-MTU2S 的同步运行 .....	414
MTU2 模块的时钟 (MPf) .....	50
MTU2S 的功能一览表 .....	480
MTU2S 模块的时钟 (Mif) .....	50
MTU2 输出引脚的初始化方法 .....	449
MTU2 中断源 .....	422
满刻度误差 .....	677
模块待机功能 .....	946
模块待机模式的设定 .....	146, 556, 601, 629, 658, 679, 687, 936
模拟电源引脚等的设定范围 .....	679

<b>N</b>	
NMI 中断 .....	85
NMI 中断和 DMAC 启动的竞争 .....	287
NMI 中断和 DTC 启动的竞争 .....	147
内部外围模块请求模式 .....	269
内部外围模块中断 .....	86

<b>P</b>	
PC 跟踪 .....	112
PCMCIA 接口 .....	232
偏移误差 .....	677
频率的变更方法 .....	60

<b>Q</b>	
取指令周期的用户断点 .....	110
全局基址寄存器 (GBR) .....	21

<b>R</b>	
RAM .....	936
RISC 方式 .....	23
容许信号源阻抗 .....	679
软件保护 .....	902
软件待机模式 .....	945

<b>S</b>	
SCI 的中断源 .....	552
SCIF 中断源 .....	597
SCSPTR 和 SCIF 引脚的关系 .....	598
SCSPTR 和 SCI 引脚的关系 .....	553
SSU 模式 .....	618
SSU 中断源 .....	629
闪存 .....	863
闪存结构 .....	867
闪存特性 .....	1056
设计电路板的注意事项 .....	680
深度软件待机模式 .....	946
时钟频率的控制电路 .....	52
时钟时序 .....	1002
时钟同步串行格式 (I <sup>2</sup> C2) .....	649
时钟同步模式 .....	510, 540, 590
时钟运行模式 .....	53
时钟振荡器 (CPG) .....	50
手动复位 .....	68
数据存取周期的用户断点 .....	111
数据传送控制器 (DTC) .....	118
数据传送指令 .....	33
输入外部时钟的方法 .....	61
刷新请求和总线周期的关系 .....	224
双地址模式 .....	275
睡眠模式 .....	944
顺序断点 .....	111
算术运算指令 .....	34

<b>T</b>	
同步串行通信单元 (SSU) .....	602
通常空间接口 .....	195
通过 RAM 对闪存进行仿真 .....	904
通过中断启动 DTC .....	144
通用寄存器 .....	20
通用寄存器的初始值 .....	21
突发 MPX-I/O 接口 .....	236
突发模式 .....	279
突发 ROM (时钟同步) 接口 .....	239
突发 ROM (异步) 接口 .....	226
突发写 .....	217

<b>U</b>	
VCL 电容连接方法 .....	1056



## W

外部触发的输入时序 .....	675
外部脉宽的测量功能 .....	418
外部请求模式 .....	268
外围模块的 DMA 传送 .....	286
外围时钟 (Pf) .....	50
外形尺寸图 .....	1096
位同步电路 .....	657

## X

系统寄存器的初始值 .....	21
系统控制指令 .....	37
陷阱指令 .....	72
向量基址寄存器 (VBR) .....	21
型号一览 .....	1095
循环模式 .....	270
寻址方式 .....	25

## Y

掩模型 ROM .....	935
一般非法指令 .....	72
异步模式 .....	510, 532, 582
异常处理 .....	64
异常处理后的堆栈状态 .....	73
异常处理向量表 .....	65, 86
异常处理向量表地址的计算方法 .....	67
异常处理状态 .....	39
移位指令 .....	36
引导模式 .....	888
引脚功能控制器 (PFC) .....	689
硬件保护 .....	902
用户编程模式 .....	891
用户断点控制器 (UBC) .....	96
用户断点运行的流程 .....	109
用户断点中断 .....	86
用户分支处理的启动间隔 .....	908
用户分支处理时间 .....	908
用户 MAT .....	867
用户引导模式 .....	899
有关槽非法指令异常处理的注意事项 .....	74
有关谐振器的注意事项 .....	62

## Z

噪声对策的注意事项 .....	680
噪声消除电路 .....	651
振荡停止检测功能 .....	62
正常传送模式 .....	134
直接存储器存取控制器 (DMAC) .....	256
指令格式 .....	28
指令特点 .....	23
指令系统 .....	30
中断 .....	70
中断控制器 (INTC) .....	75

中断响应时间 .....	92
中断异常处理结束后的堆栈状态 .....	91
中断优先级 .....	71
中断运行的流程 .....	89
中止的发送 .....	554
中止的检测和处理 .....	554
周期挪用模式 .....	278
转移指令 .....	36
状态寄存器 (SR) .....	20
自动请求模式 .....	268
自动刷新 .....	223
自刷新 .....	224
总线关联信号的引脚状态 .....	1080
总线模式和通道优先级 .....	281
总线权释放状态 .....	39
总线时钟 (Bf) .....	50
总线仲裁 .....	248
总线状态控制器 (BSC) .....	148

修订记录	SH7080 群硬件手册
------	--------------

Rev.	发行日	修订内容	
		页	修订处
1.00	2007.06.18	—	初版发行
2.00	2007.07.18	1	在 1.1 中的第 5 段内追加 “和掩模型 ROM 版两种”。
		3	在表内的 “用户调试接口” 下追加 “（只限 F-ZTAT 版）”。
		4	删除 “多功能定时器脉冲单元 2” 中的 “脉冲输出模式” 内的 “单触发”。
		6	在图 1.1 内追加注 1。
		7	在图 1.2 内追加注 1、注 2、注 3；在图 1.2 的标题后追加 “(TQFP1414-100)”。
		8	在图 1.3 内追加注 1、注 2、注 3。
		9	在图 1.4 内追加注 1、注 2、注 3。
		10	在图 1.5 内追加注 1、注 2、注 3。
		11	追加图 1.6。
		16	在表内的 “用户调试接口” 下追加 “（只限 F-ZTAT 版）”；
		17	删除 “高级用户调试接口” 内引脚名为 “AUDMD” 和 “AUDRST” 的部分；在 “E10A 接口” 下追加 “（只限 F-ZTAT 版）”。
		42	修改图 3.1；在图 3.1 的标题内追加 “运行”。
		43	修改图 3.2；在图 3.2 的标题内追加 “运行”
		56	删除注 2 内的 “此频率不能超过 80MHz”；在注 7 内追加 “和总线时钟 (Bφ)”。
		57 ~ 58	在 4.4.1 的表的除 “15” 位的各说明栏内追加 “设定禁止设定的值时，不保证之后的运行” 和 “上述以外：禁止设定”。
		60	在 4.5 的 3 内追加 “且总线时钟 (Bφ) ≥ MTU2 时钟 (MPφ)” 和 “再者，FRQCR 的改写处理必须在内部 ROM 或内部 RAM 上进行程序的执行”；修改 4.5 的 4；追加 “【注】(1 ~ 24n) 根据内部状态而改变”。
		67	将表 5.5 内的 “上电复位” 的 “不初始化” 改为 “初始化”。
		96	在 7 内追加 “另外，掩模型 ROM 只有 L 总线取指令地址断点 (2 个通道)”；在 7.1 的 5 内追加 “(支持 E10A 全功能的 F-ZTAT 版为 8 组)”。
		98	在表 7.2 下追加 “【注】* 只限 F-ZTAT 版”。
		100	修改 7.3.3 的 BBRA 图并在图内追加 “【注】* 在掩模型 ROM 版、无 ROM 版中为保留位。读写的值总是 0”；修改 7.3.3 的 BBRA 的表并在表下追加 “【注】* 在掩模型 ROM 版、无 ROM 版中为保留位。读写的值总是 0”。
		101	在 7.3.4 和 7.3.5 的标题后追加 “（只限 F-ZTAT 版）”。
		103	在 7.3.8 和 7.3.9 的标题后追加 “（只限 F-ZTAT 版）”。
		104	修改 7.3.10 的 BBRB 图并在图内追加 “【注】* 在掩模型 ROM 版、无 ROM 版中为保留位。读写的值总是 0”；修改 7.3.10 的 BBRB 表并在表下追加 “【注】* 在掩模型 ROM 版、无 ROM 版中为保留位。读写的值总是 0”。
		107	在 7.3.12 的标题后追加 “（只限 F-ZTAT 版）”。

Rev.	发行日	修订内容	
		页	修订处
2.00	2007.07.18	108	在 7.3.13 的标题后追加“(只限 F-ZTAT 版)”；在 7.3.13 内追加“或者手动复位”和“(支持 E10A 全功能的 F-ZTAT 版是 8 个)”；将原文中的“8 个”改为“4 个”。
		109	在 7.3.14 的标题后追加“(只限 F-ZTAT 版)”；在 7.3.14 内追加“或者手动复位”和“(支持 E10A 全功能的 F-ZTAT 版是 8 个)”；将原文中的“8 个”改为“4 个”。
		112	将 7.4.6 中的 3 和 4 内的“8 组”改为“4 组”，并追加“(支持 E10A 全功能的 F-ZTAT 版是 8 组)”。
		117	在 7.5 内追加 9。
		139	追加图 8.15。
		141	在 8.5.9 的第 1 段内追加“权”；在表 8.11 内追加“CSSTP2”“CSSTP3”两项；将表 8.11 的注 1 改为“只对读向量后的 CPU 外部存取请求，释放总线权”、注 3 改为“Don't care”、删除注 4。
		147	追加 8.9.11。
		151	在 9.3.1 内追加“在区域 2、3 上连接同步 DRAM 时， $\overline{RASx}$ 、 $\overline{CASx}$ 、 $\overline{RD}/\overline{WR}$ 、 $\overline{DQMxx}$ 等信号也有效。另外在区域 5 和 6 中，选择 PCMCIA 接口时，在相应的存取位上加 $\overline{CS5}$ 、 $\overline{CS6}$ 、 $\overline{CE1A}$ 、 $\overline{CE1B}$ 、 $\overline{CE2A}$ 、 $\overline{CE2B}$ 有效”；修改表 9.2。
		152	修改表 9.3。
		153	修改表 9.4。
		154	修改表 9.5。
		168	在表内的“14 ~ 12”位的说明栏内追加注 2。
		190	在表内的“9”位的说明栏内追加注。
		250	在本页的第 5 段后追加“另外，本 LSI 即使在向外部释放总线权中，只要不产生向外部器件的存取，处理就会继续。之后，当向外部器件的存取产生时，就变为等待总线权恢复的状态”。
		251	在图 9.51 下追加“在向 DMAC 执行总线仲裁时，因为在接收 DMAC 的总线权处理时需要 $IB\phi$ 时间，所以在 I 总线上就会插入 $IB\phi$ 时间的 NOP”和“在向 TDC 执行总线仲裁时，因为在接收 DTC 的总线权处理时不需要插入 NOP 等，所以持续进行总线存取的处理”。
		252	删除 9.5.14 中的 (3)；追加 9.5.15。
		253 ~ 255	追加 9.5.16、9.5.17。
		269	将表 10.6 和注内的“ADDR1”改为“ADDR4 ~ ADDR7”。
		285	追加 10.4.6。
		287	在表 10.9 下追加注 2。
318	在表 11.28 内追加“互补 PWM 模式”。		
322	在表“2”、“1”、“0”位的说明栏内追加“TSR_5 的 CMFU5 位被置 1 时，允许或禁止由 CMFU5 位引起的中断请求 (TGIU_5)”。		
345	在 11.3.23 的表下追加“【注】* 在 MTU2S 上将 BDC 位置 1 时，FB 位一定不能设定为 0”。		
388	在图 11.41 内追加“初始输出”。		
418	替换图 11.92。		

Rev.	发行日	修订内容	
		页	修订处
2.00	2007.07.18	448	在 MTU2 中追加输入 1 个输入捕捉就可以同时捕捉 TCNT_1 和 TCNT_2 的功能。如果使用本功能, 就可以在 TCNT_1 和 TCNT_2 的捕捉时序无偏差的状态下, 取得 32 位计数。详细内容请参照“11.3.8 定时器的输入捕捉控制寄存器 (TICCR)”
		503	将表 13.5 中的“OEI2”改为“OEI3”、“输出的允许中断 2”改为“输出允许中断 3”、“OEI3”改为“OEI2”、“输出的允许中断 3”改为“输出允许中断 2”; 追加 13.6。
		508	在图 14.3 内追加“3Pφ+ 计数器时钟 1 个周期的时间”。
		521	修改表中的部分内容。
		540	在 (2) 内追加“但是, 只进行接收时, 就会发生溢出错误或输出同步时钟直到 RE 位清 0 为止。如果进行 n 个字符数的接收时, 必须将外部时钟作为时钟源。在使用内部时钟时, 必须按照先设定 RE=1 且 TE=1, 再在发送 n 个字符数的虚拟数据的同时接收 n 个字符数的顺序”。
		566	将表中“5”位说明栏内的“1:”内和“注”上方的“小于”改为“小于等于”。
		567	在表中“0”位的说明中追加“通过 DTC 读出 SCFRDR 内的全部接收数据时”。
		592	修改图 16.13 的部分内容。
		596	修改图 16.18 的部分内容。
		601	追加 16.7.8。
		611	删除 17.3.6 中的“设定 SSO 引脚、SSI 引脚、SSCK 引脚、SCS 引脚的漏极开路输出”; 修改 17.3.6 中表的部分内容。
		616	删除表 17.6 中的“SCKS”项。
		618	删除图 17.4 内的部分内容。
		625	删除图 17.12 内的部分内容。
		633	在表“6”位的说明内追加部分内容。
		636	在表“2~0”位的说明内追加“另外, 检测出停止条件后, 就会自动变为 111”。
		658	追加 18.8.3、18.8.4 和 18.8.5。
		666 ~ 667	替换表 19.3。
		668	在 19.3.4 中追加“以及 A/D 模块 1”。
		672	删除 19.4.2 和 19.4.3 内 4 以下的部分。
		676	追加 19.4.7。
		679	将 19.7.4 中的“ $AV_{SS} \leq VAN \leq AV_{CC}$ ”改为“ $AV_{SS} \leq VAN \leq AV_{REF}$ ”。
		697	将表 21.9 中的“ $\overline{AUDRST}$ 输入 (AUD)*”、“AUDMD 输入 (AUD)*”改为“—”。
		699	将表 21.10 中的“ $\overline{AUDRST}$ 输入 (AUD)*”、“AUDMD 输入 (AUD)*”改为“—”。
		700	在表 21.11 下追加“【注】* 只限 F-ZTAT 版”。
		701	在表 21.12 下追加“【注】* 只限 F-ZTAT 版”。
702	在表 21.13 下追加“*2 只限 F-ZTAT 版”; “ $\overline{AUDRST}$ 输入 (AUD)*”、“AUDMD 输入 (AUD)*”改为“—”。		

Rev.	发行日	修订内容	
		页	修订处
2.00	2007.07.18	704	在表 21.14 下追加 “*2 只限 F-ZTAT 版”；“ $\overline{\text{AUDRST}}$ 输入 (AUD) *”、“AUDMD 输入 (AUD) *” 改为 “—”。
		707	将表 21.17 (1) 中引脚号 “47”、“48” 的初始功能引脚名改为 “PD12”、“PD13”、“D12” 和 “D13”。
		708 ~ 711	替换表 21.17 (2)。
		713	将表 21.18 (1) 中引脚号 “56”、“54” 的初始功能引脚名改为 “PD12”、“PD13”、“D12” 和 “D13”。
		714 ~ 717	替换表 21.18 (2)。
		720	将表 21.19 (1) 中引脚号 “67”、“66” 的初始功能引脚名改为 “PD20”、“PD21”、“D20” 和 “D21”。
		721	将表 21.19 (1) 中引脚号 “110”、“111” 的初始功能引脚名改为 “PE1”、“PE2”。
		722 ~ 725	替换表 21.19 (2)。
		729	将表 21.20 (1) 中引脚号 “91”、“90” 的初始功能引脚名改为 “PD20”、“PD21”、“D20” 和 “D21”、引脚号 “134”、“135” 的初始功能引脚名改为 “PE1”、“PE2”。
		731 ~ 735	替换表 21.20 (2)。
		739 ~ 814	追加注 “只有在内部 ROM 有效或者无效的外部扩展模式中才有效的功能。在单芯片模式中不要设定”。
		763	删除 21.1.3 中的 “以及 SCI 的 SCK 输入 / 输出”。
		782	删除 “5” “4” 位说明内的 “/AUDMD” 和 “当使用 E10A 的 AUD 功能时，此引脚被固定为 AUDMD 输入”；删除 “1” “0” 位说明内的 “ $\overline{\text{AUDRST}}$ ” 当使用 E10A 的 AUD 功能时，此引脚被固定为 $\overline{\text{AUDRST}}$ 输入。
		788	删除 “6” “5” “4” 位说明内的 “/AUDMD” 和 “当使用 E10A 的 AUD 功能时，此引脚被固定为 AUDMD 输入”；删除 “2” “1” “0” 位说明内的 “ $\overline{\text{AUDRST}}$ ” 当使用 E10A 的 AUD 功能时，此引脚被固定为 $\overline{\text{AUDRST}}$ 输入。
		794	删除 21.1.9 中的 “、SCI/SCIF 的 SCK 输入 / 输出”。
		808	删除 “10” “9” “8” 位说明内的 $\overline{\text{AUDRST}}$ 当使用 E10A 的 AUD 功能时，此引脚被固定为 $\overline{\text{AUDRST}}$ 输入；删除 “6” “5” “4” 位说明内的 “/AUDMD” 和 “当使用 E10A 的 AUD 功能时，此引脚被固定为 AUDMD 输入”。
		843	删除图 22.9 中的 “/AUDMD (输入)”、“ $\overline{\text{AUDRST}}$ (输入)”。
		844	删除图 22.10 中的 “/AUDMD (输入)”、“ $\overline{\text{AUDRST}}$ (输入)”。
		850	在图 22.11、图 22.12 内追加 “【注】* 只限 F-ZTAT 版”。
		851	删除图 22.13 中的 “ $\overline{\text{AUDRST}}$ (输入) *”、“/AUDMD (输入) *”；追加 “只限 F-ZTAT 版”。
852	删除图 22.14 中的 “ $\overline{\text{AUDRST}}$ (输入) *”、“/AUDMD (输入) *”；追加 “只限 F-ZTAT 版”。		
863	在 23.1 的 “40MHz” 后追加 “(P $\phi$ )”。		
883	在 “闪存的正常 / 失败” 后追加 “结果”。		
884	将表中的 “H'A5” 改为 “H'5A”。		
886	将表中的 “H'A5” 改为 “H'5A”。		

Rev.	发行日	修订内容	
		页	修订处
2.00	2007.07.18	889	将（2）中 2 内的“等待查询设定命令”改为“等待查询选择命令”、3 内的“支持情况”改为“支持信息”。
		890	修改图 23.8。
		893	在图 23.11 下追加“编程 / 擦除的程序下载结束，并且将 SCO 位清 0 后，就可以将频率控制寄存器（FRQCR）的设定改为任意的值”。
		897	在图 23.12 下追加“编程 / 擦除的程序下载结束，并且将 SCO 位清 0 后，就可以将频率控制寄存器（FRQCR）的设定改为任意的值”。
		899	将图 23.13 中的“设定 FTDAR=H'01（下载目标设定为 H'FFFF9800）”改为“设定 FTDAR=H'04（下载目标设定为 H'FFFFB000）”。
		908	将 23.8.3（1）中的“6ms”改为“10ms”；追加 23.8.3（5）。
		911	删除图 23.23 中的“引导的”；将 5 中的“这是长度为 4 字节的应答”改为“这是含有 4 字节长度信息的应答”。
		913	在“长度”内追加“/ 应答”；将“校验和”的解释改为“加算从命令到校验和的总和，并通过设定使低位 1 字节为 H'00”。
		914	将（2）中的“当器件代码一致时返回 ACK”改为“当指定的器件代码和支持器件一致时返回 ACK”；删除（3）中表及解释中有关“模式数”的部分。
		915	将（4）中的“当时钟模式一致时返回 ACK”改为“当指定的时钟模式和可选择的时钟模式一致时返回 ACK”；删除“即使在时钟模式的查询中时钟模式数为 H'00、H'01，也必须分别为该值选择时钟模式”；将（5）中的“种类数”改为“频率种类数”。
		916	将（6）中的“工作频率的种类数”改为“频率种类数”；将（7）中的“连续的用户引导 MAT”改为“用户引导 MAT”。
		917	将（8）中的“连续的用户引导 MAT”改为“用户引导 MAT”。
		918	将（11）中的“可选择时返回 ACK”改为“当选择指定的位速率时作为 ACK 发送”。
		920	将原文中的“23.9.1.5”改为“（12）”。
		923	将（3）中“编程地址”的解释改为“必须指定 128 字节边界的地址”、“H'01000000”改为“H'00010000”。
		924	将 23.9.1.9 中的“擦除数据”改为“块擦除命令”。
		935	追加 24。
		936	追加 25.1.3。
		943	在 26.3.6 中的表内追加“将该位置 1 时，必须先将 STBCR5 的 MSTP25 位置 0”。
		944	在 26.4.2 内追加“不要通过中断解除睡眠模式”。
946	追加 26.8.3。		
948 ~ 960	在 27.1 的表中追加“连接总线宽度”项目。		
970	将表中的“SDOS”、“SSCKOS”、“SCSOS”改为“—”。		
995	将 27.3 中的“BRSR”和“BRDR”的手动复位改为“初始化”；追加“*4 只限 F-ZTAT 版”。		
996	删除 28 中的“【注】请注意：此电特性的规格为暂定值，有可能会不预先通知而改变规格值的情况”。		

Rev.	发行日	修订内容	
		页	修订处
2.00	2007.07.18	998	修改表 28.2 的部分内容，并追加 “【注】 * 只限支持 E10A 全功能的 F-ZTAT 版”。
		1000	修改表 28.3 的部分内容，并追加 “【注】 * 只限支持 E10A 全功能的 F-ZTAT 版”。
		1004	在表 28.7 中的 “MD1、MD0” 后追加 “FWE”；在注 1 内追加 “BREQ”。
		1005	在图 28.6 内追加 “FWE”。
		1007 ~ 1008	修改表 28.8 内的部分内容；并追加 “*3 支持 E10A 全功能的 F-ZTAT 版的运行保证温度范围是 0 ~ +50°C”。
		1045	将表 26.14 中的 “3tpcyc+50” 改为 “2tpcyc+50” 和 “2tpcyc”。
		1047	将表 28.15 中的 “3tpcyc+50” 改为 “2tpcyc+50” 和 “2tpcyc”。
		1054	替换 28.3.15。
		1055	将表 28.21 内的 “2.5” 改为 “2.0”、“1*3/3*4” 改为 “1*1/3*2”；删除注 3、注 4。
		1078 ~ 1079	修改表 A.4 的部分内容。
		1095 ~ 1096	替换表 C.1。
		1099	追加图 D.5。

---

**瑞萨 32 位 RISC 单片机  
硬件手册  
SH7080 群**

Publication Date: Rev1.00, Jun.18, 2007  
Rev2.00, Jul. 18, 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



SH7080群



瑞萨电子株式会社

RCJ09B0039-0200