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.

Hardware Manual

32

SH7263群

瑞萨32位RISC单片机

SuperH™ RISC engine族/SH7260系列

SH7263

R5S72630P200FP
R5S72631P200FP
R5S72632P200FP
R5S72633P200FP

瑞萨电子
www.renesas.com

Rev.1.00 2008.09

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 life
   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

注意

本文只是参考译文，前页所载英文版“Cautions”具有正式效力。

关于利用本资料时的注意事项

1. 本资料是为了让用户根据用途选择合适的本公司产品的参考资料，对于本资料中所记载的技术信息，并非意味着对本公司或者第三者的知识产权及其他权利做出保证或对实施权力进行的承诺。
2. 对于因使用本资料所记载的产品数据、图、表、程序、算法及其他应用电路例而引起的损害或者对第三者的知识产权及其他权利造成侵犯，本公司不承担任何责任。
3. 不能将本资料所记载的产品和技术用于大规模破坏性武器的开发等目的、军事目的或其他的军需用途方面。另外，在出口时必须遵守日本的《外汇及外国贸易法》及其他出口的相关法令并履行这些法令中规定的必要手续。
4. 本资料所记载的产品数据、图、表、程序、算法以及其他应用电路例等所有信息均为本资料发行时的内容，本公司有可能在未做事先通知的情况下，对本资料所记载的产品或者产品规格进行更改。所以在购买和使用本公司的半导体产品之前，请事先向本公司的营业窗口确认最新的信息并经常留意本公司通过公司主页(http://www.renesas.com)等公开的最新信息。
5. 对于本资料中所记载的信息，制作时我们尽力保证出版时的精确性，但不承担因本资料的叙述不当而致使顾客遭受损失等的任何相关责任。
6. 在使用本资料所记载的产品数据、图、表等所示的技术内容、程序、算法及其他应用电路例时，不仅要对所使用的技术信息进行单独评价，还要对整个系统进行充分的评价。请顾客自行负责，进行是否适用的判断。本公司对于是否适用不负任何责任。
7. 本资料中所记载的产品并非针对万一出现故障或是错误运行就会威胁到人的生命或给人体带来危害的机器、系统(如各种安全装置或者运输交通用的、医疗、燃烧控制、航天器械、核能、海底中继用的机器和系统等)而设计和制造的，特别是对于品质和可靠性要求极高的机器和系统等（将本公司指定用于汽车方面的产品用于汽车时除外）。如果要用于上述的目的，请务必事先向本公司的营业窗口咨询。另外，对于用于上述目的而造成的损失等，本公司概不负责。
8. 除上述第7项内容外，不能将本资料中记载的产品用于以下用途。如果用于以下用途而造成的损失，本公司概不负责。
   1）生命维持装置。
   2）植埋于人体使用的装置。
   3）用于治疗（切除患部、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

1. NC引脚的处理
   【注意】 NC引脚什么也不要连接。
   NC(Non-Connection)引脚除了不连接内部电路以外，还有用于测试引脚和降低噪声等目的。因此，对于NC引脚，请什么也不要连接。
   连接时不能保证器件的正常运行。

2. 未使用的输入引脚的处理
   【注意】未使用的输入引脚必须固定成高电平或者低电平。
   CMOS产品的输入引脚一般为高阻抗输入。将未使用的输入引脚处于开放状态运行时，就可能由于周围噪声的感应而产生中间电平，并产生内部浸透电流，引起误动作。
   因此，对于未使用的输入引脚，请固定成高电平或者低电平。

3. 初始化前的处理
   【注意】接通电源时，产品的状态不定。
   从给所有电源引脚外加电压开始，到给复位引脚输入低电平为止，内部电路处于不定状态，并且寄存器的设定和各引脚的输出状态也不定。因此，在进行系统设计时，请采用对策，避免因此不定状态而引起的系统误动作。
   对于具有复位功能的产品，在接通电源后，请首先执行复位运行。

4. 禁止存取未定义的地址或者保留地址
   【注意】禁止存取未定义的地址或者保留地址。
   未定义地址或者保留地址，除了用于将来扩展功能以外，还有分配给测试用的寄存器等情况。因为不能保证存取这些寄存器时的运行和继续运行，所以请不要存取。

5. 各寄存器保留位的读取/写入
   对于各模块中使用的寄存器保留位，在其说明中只要没有指定读取/写入值就请按以下方法处理。
   读出值总是为0。写入时写0或者是写入刚刚读出的值。如果是写入刚刚读出的值，将来对此位分配扩展功能时，就有不影响此扩展功能的优点。

本手册的构成

本手册由以下内容构成：

1. 有关产品的一般注意事项
2. 本手册的构成
3. 前言
4. 目录
5. 概要
6. 各功能模块的说明
   - CPU 和系统控制
   - 内部外围模块
     各模块功能说明的构成因各模块而不同。一般由①特点、②输入 / 输出引脚、③寄存器说明、④运行说明、⑤使用注意事项等章节构成。

> 设计采用本 LSI 的应用系统时，请在充分确认了注意事项的基础上进行设计。请务必阅读各章节中有关说明的注意事项和各章节最后的使用注意事项 (使用注意事项根据需要记载)。

7. 寄存器一览表
8. 电特性
9. 附录
   - 产品型号和外形尺寸图等
10. 本版中修改或者追加的部分 （仅适用于修订版）

> 修订履历汇总了对前版内容的修改和追加的主要部分。
> 记载的内容并非全部修订的内容，请在本手册的正文中确认有关详细内容。

11. 索引

前言

本 LSI 是以瑞萨独创的 RISC 方式 CPU 为核心、集聚了系统构成所需外围功能的 RISC 单片机。

对象：本手册是以设计 “采用本 LSI 的应用系统”的用户为对象。使用本手册的读者需要具备电路、逻辑电路以及微型计算机的基础知识。

目的：本手册是以 “能让用户理解本 LSI 的硬件功能和电特性”为目的。

1. 阅读方法

- 希望了解全部功能时
  → 请按照目录的顺序阅读。
  本手册大致按CPU、系统控制功能、外围功能、电特性的顺序构成。
- 希望了解详细的CPU功能时
  → 请参照 《SH-2A、 SH2A-FPU 软件手册》。
- 知道寄存器名，希望了解寄存器的详细功能时
  → 在本手册的后面附有 “索引”，请从索引检索页号。
  有关地址、位内容及其初始化，汇总在 “第 34 章 寄存器一览”中。

2. Description of Numbers and Symbols

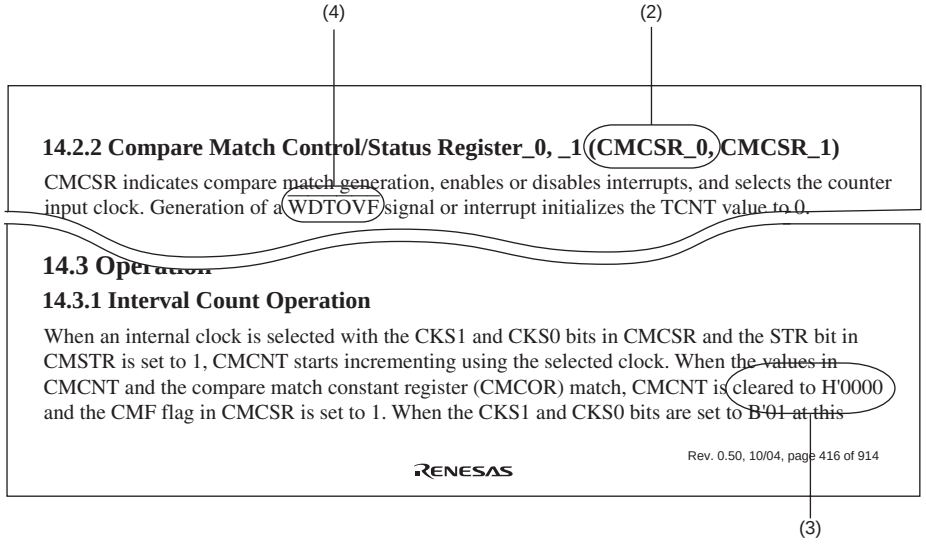
Aspects of the notations for register names, bit names, numbers, and symbolic names in this manual are explained below.

(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] $\overline{\text{WDTOVF}}$

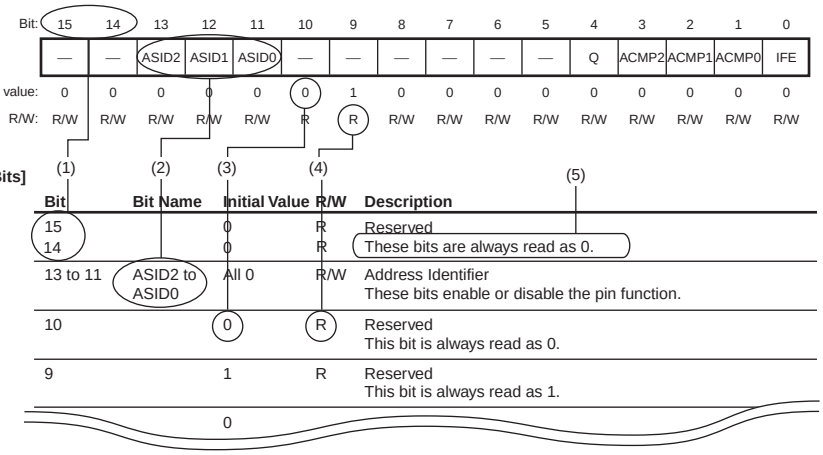


Note: The bit names and sentences in the above figure are examples and have nothing to do with the contents of this manual.

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.

[Bit Chart]



[Table of Bits]

Bit	Bit Name	Initial Value	R/W	Description
15		0	R	Reserved
14		0	R	These bits are always read as 0.
13 to 11	ASID2 to ASID0	All 0	R/W	Address Identifier These bits enable or disable the pin function.
10		0	R	Reserved This bit is always read as 0.
9		1	R	Reserved This bit is always read as 1.
		0		

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

(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.

目　录

第 1 章　概要 1
1.1 SH7263 的特点 1
1.2 产品一览表 7
1.3 框图 7
1.4 引脚排列图 8
1.5 引脚功能 9
1.6 引脚一览表 15

第 2 章　CPU 30
2.1 寄存器结构 30
2.1.1 通用寄存器 30
2.1.2 控制寄存器 31
2.1.3 系统寄存器 32
2.1.4 寄存器组 33
2.1.5 寄存器初始值 33
2.2 数据格式 34
2.2.1 寄存器的数据格式 34
2.2.2 存储器的数据格式 34
2.2.3 立即数的数据格式 34
2.3 指令特点 35
2.3.1 RISC 方式 35
2.3.2 寻址方式 38
2.3.3 指令格式 42
2.4 指令系统 45
2.4.1 指令系统的分类 45
2.4.2 数据传送指令 50
2.4.3 算术运算指令 53
2.4.4 逻辑运算指令 55
2.4.5 移位指令 56
2.4.6 转移指令 57
2.4.7 系统控制指令 58
2.4.8 浮点运算指令 60
2.4.9 有关 FPU 的 CPU 指令 62
2.4.10 位操作指令 63
2.5 处理状态 64

第 3 章　浮点运算单元 （FPU） 66
3.1 特点 66
3.2 数据格式 67
3.2.1 浮点格式 67
3.2.2 非数 （NaN） 69
3.2.3 非规格化数 69
3.3 寄存器的说明 70
3.3.1 浮点寄存器 70
3.3.2 浮点状态 / 控制寄存器 （FPSCR） 71
3.3.3 浮点通信寄存器 （FPUL） 72
3.4 舍入 72
3.5 浮点异常 73
3.5.1 FPU 异常源 73

3.5.2 FPU 异常处理 73

第 4 章 时钟脉冲振荡器 （CPG） 74
4.1 特点 74
4.2 输入 / 输出引脚 77
4.3 时钟运行模式 78
4.4 寄存器说明 81
4.4.1 频率控制寄存器 （FRQCR） 81
4.5 频率的变更方法 83
4.5.1 倍频率的变更 83
4.5.2 分频比的变更 83
4.6 电路板设计的注意事项 84
4.6.1 外部时钟输入时的注意事项 84
4.6.2 晶体振荡器使用时的注意事项 84
4.6.3 有关振荡器的注意事项 84
4.6.4 PLL 振荡电路使用时的注意事项 85
4.7 使用时的注意事项 85

第 5 章 异常处理 86
5.1 概要 86
5.1.1 异常处理的种类和优先级 86
5.1.2 异常处理的运行 87
5.1.3 异常处理向量表 88
5.2 复位 89
5.2.1 输入 / 输出引脚 89
5.2.2 复位的种类 89
5.2.3 上电复位 90
5.2.4 手动复位 91
5.3 地址错误 92
5.3.1 地址错误发生源 92
5.3.2 地址错误异常处理 92
5.4 寄存器组错误 93
5.4.1 寄存器组错误发生源 93
5.4.2 寄存器组错误异常处理 93
5.5 中断 94
5.5.1 中断源 94
5.5.2 中断优先级 95
5.5.3 中断异常处理 96
5.6 由指令引起的异常 97
5.6.1 由指令引起的异常种类 97
5.6.2 陷阱指令 97
5.6.3 槽非法指令 98
5.6.4 一般非法指令 98
5.6.5 整数除法指令 98
5.6.6 浮点运算指令 99
5.7 不接受异常处理时 99
5.8 异常处理后堆栈的状态 100
5.9 使用时的注意事项 101
5.9.1 堆栈指针 （SP）的值 101
5.9.2 向量基址寄存器 （VBR）的值 101
5.9.3 在地址错误异常处理的堆栈存取中发生的地址错误 101

第 6 章　中断控制器 （INTC） 102
6.1　特点 102
6.2　输入 / 输出引脚 104
6.3　寄存器说明 105
6.3.1　中断优先级设定寄存器 01、 02、 05 ～ 17 （IPR01、 IPR02、 IPR05 ～ IPR17） 106
6.3.2　中断控制寄存器 0 （ICR0） 107
6.3.3　中断控制寄存器 1 （ICR1） 108
6.3.4　中断控制寄存器 2 （ICR2） 109
6.3.5　IRQ 中断请求寄存器 （IRQRR） 110
6.3.6　PINT 中断允许寄存器 （PINTER） 111
6.3.7　PINT 中断请求寄存器 （PIRR） 112
6.3.8　存储体控制寄存器 （IBCR） 113
6.3.9　存储体编号寄存器 （IBNR） 114
6.4　中断源 115
6.4.1　NMI 中断 115
6.4.2　用户断点中断 115
6.4.3　H-UDI 中断 115
6.4.4　IRQ 中断 115
6.4.5　PINT 中断 116
6.4.6　内置外围模块中断 116
6.5　中断异常处理向量表和优先级 117
6.6　运行说明 122
6.6.1　中断运行的流程 122
6.6.2　中断异常处理结束后的堆栈状态 123
6.7　中断响应时间 124
6.8　寄存器组 129
6.8.1　存储体的对象寄存器和输入 / 输出方式 129
6.8.2　存储体保存、返回的运行 130
6.8.3　在保存至所有存储体的状态下进行的保存、返回 131
6.8.4　寄存器组异常 131
6.8.5　寄存器组错误异常处理 131
6.9　通过中断请求信号的数据传送 132
6.9.1　中断请求信号作为 CPU 中断源而不作为 DMAC 启动源的情况 132
6.9.2　中断请求信号作为 DMAC 启动源而不作为 CPU 中断源的情况 132
6.10　使用注意事项 133
6.10.1　清除中断源的时序 133
6.10.2　IRQOUT 的取反时序 133

第 7 章　用户断点控制器 （UBC） 134
7.1　特点 134
7.2　输入 / 输出引脚 136
7.3　寄存器说明 136
7.3.1　断点地址寄存器 （BAR） 137
7.3.2　断点地址屏蔽寄存器 （BAMR） 138
7.3.3　断点数据寄存器 （BDR） 138
7.3.4　断点数据屏蔽寄存器 （BDMR） 139
7.3.5　断点总线周期寄存器 （BBR） 140
7.3.6　断点控制寄存器 （BRCR） 141
7.4　运行说明 143
7.4.1　用户断点运行的流程 143
7.4.2　取指令周期的断点 143
7.4.3　数据存取周期的断点 144

7.4.4 被保存的程序计数器值 144
7.4.5 使用例 145
7.5 使用时的注意事项 147

第 8 章 高速缓存 148
8.1 特点 148
8.1.1 高速缓存的结构 148
8.2 寄存器的说明 150
8.2.1 高速缓存控制寄存器 1 （CCR1） 150
8.2.2 高速缓存控制寄存器 2 （CCR2） 151
8.3 运行说明 154
8.3.1 高速缓存的搜索 154
8.3.2 读操作 155
8.3.3 预取操作 （仅限操作数高速缓存） 155
8.3.4 写操作 （仅限操作数高速缓存） 155
8.3.5 回写缓冲器 （仅限操作数高速缓存） 156
8.3.6 高速缓存与外部存储器匹配 156
8.4 存储器分配的高速缓存结构 157
8.4.1 地址阵列 157
8.4.2 数据阵列 158
8.4.3 使用例 159
8.4.4 注意事项 159

第 9 章 总线状态控制器 （BSC） 160
9.1 特点 160
9.2 输入 / 输出引脚 163
9.3 区域概要 164
9.3.1 地址映射 164
9.3.2 各区域的数据总线宽度和引脚功能设定 165
9.4 寄存器说明 166
9.4.1 通用控制寄存器 （CMNCR） 167
9.4.2 CSn 空间总线控制寄存器 （CSnBCR）（n=0 ～ 7） 168
9.4.3 CSn 空间等待控制寄存器 （CSnWCR）（n=0 ～ 7） 171
9.4.4 SDRAM 控制寄存器 （SDCR） 194
9.4.5 刷新定时器的控制 / 状态寄存器 （RTCSR） 196
9.4.6 刷新定时器的计数器 （RTCNT） 197
9.4.7 刷新时间常数寄存器 （RTCOR） 198
9.5 运行说明 199
9.5.1 端法 / 存取长度和数据对齐 199
9.5.2 通常空间接口 203
9.5.3 存取等待的控制 208
9.5.4 CSn 断言期间的扩展 210
9.5.5 MPX-I/O 接口 211
9.5.6 SDRAM 接口 213
9.5.7 突发 ROM （时钟异步）接口 241
9.5.8 带字节选择的 SRAM 接口 242
9.5.9 PCMCIA 接口 247
9.5.10 突发 MPX-I/O 接口 251
9.5.11 突发 ROM （时钟同步）接口 254
9.5.12 存取周期之间的空闲 255
9.5.13 总线仲裁 259
9.5.14 其他 260

第 10 章　直接存储器存取控制器 （DMAC） 262
10.1　特点 262
10.2　输入 / 输出引脚 264
10.3　寄存器说明 265
10.3.1　DMA 源地址寄存器 （SAR） 267
10.3.2　DMA 目标地址寄存器 （DAR） 267
10.3.3　DMA 传送计数寄存器 （DMATCR） 267
10.3.4　DMA 通道控制寄存器 （CHCR） 268
10.3.5　DMA 再装入源地址寄存器 （RSAR） 272
10.3.6　DMA 再装入目标地址寄存器 （RDAR） 272
10.3.7　DMA 再装入传送计数寄存器 （RDMATCR） 272
10.3.8　DMA 操作寄存器 （DMAOR） 273
10.3.9　DMA 扩展资源选择器 0 ～ 3 （DMARS0 ～ DMARS3） 276
10.4　运行说明 278
10.4.1　传送流程 278
10.4.2　DMA 传送请求 280
10.4.3　通道优先级 284
10.4.4　DMA 的传送种类 287
10.4.5　总线周期的状态数和 DREQ 引脚的采样时序 294
10.5　使用时的注意事项 297
10.5.1　半结束标志的置位及半结束中断 297
10.5.2　DACK 输出及 TEND 输出的时序 297

第 11 章　多功能定时器脉冲单元 2 （MTU2） 298
11.1　特点 298
11.2　输入 / 输出引脚 302
11.3　寄存器的说明 303
11.3.1　定时器控制寄存器 （TCR） 306
11.3.2　定时器模式寄存器 （TMDR） 310
11.3.3　定时器 I/O 控制寄存器 （TIOR） 312
11.3.4　定时器中断允许寄存器 （TIER） 329
11.3.5　定时器状态寄存器 （TSR） 331
11.3.6　定时器缓冲运行传送模式寄存器 （TBTM） 334
11.3.7　定时器输入捕捉控制寄存器 （TICCR） 335
11.3.8　定时器同步清除寄存器 （TSYCR） 336
11.3.9　定时器 A/D 转换开始请求控制寄存器 （TADCR） 337
11.3.10　定时器 A/D 转换开始请求周期设定寄存器 （TADCORA/B_4） 339
11.3.11　定时器 A/D 转换开始请求周期设定缓冲寄存器 （TADCOBRA/B_4） 339
11.3.12　定时器计数器 （TCNT） 340
11.3.13　定时器通用寄存器 （TGR） 340
11.3.14　定时器启动寄存器 （TSTR） 341
11.3.15　定时器同步寄存器 （TSYR） 342
11.3.16　定时器读写允许寄存器 （TRWER） 343
11.3.17　定时器输出主控器允许寄存器 （TOER） 344
11.3.18　定时器输出控制寄存器 1 （TOCR1） 345
11.3.19　定时器输出控制寄存器 2 （TOCR2） 347
11.3.20　定时器输出电平缓冲寄存器 （TOLBR） 350
11.3.21　定时器门控寄存器 （TGCR） 351
11.3.22　定时器子计数器 （TCNTS） 352
11.3.23　定时器空载时间数据寄存器 （TDDR） 352
11.3.24　定时器周期数据寄存器 （TCDR） 353
11.3.25　定时器周期缓冲寄存器 （TCBR） 353

11.3.26 定时器跳过中断设定寄存器 （TITCR） 354
11.3.27 定时器跳过中断次数计数器 （TITCNT） 355
11.3.28 定时器缓冲传送设定寄存器 （TBTER） 356
11.3.29 定时器空载时间允许寄存器 （TDER） 357
11.3.30 定时器波形控制寄存器 （TWCR） 358
11.3.31 和总线主控器的接口 358
11.4 运行说明 359
11.4.1 基本运行 359
11.4.2 同步运行 364
11.4.3 缓冲运行 366
11.4.4 级联运行 369
11.4.5 PWM 模式 373
11.4.6 相位计数模式 378
11.4.7 复位同步 PWM 模式 384
11.4.8 互补 PWM 模式 387
11.4.9 A/D 转换开始请求的延迟功能 414
11.4.10 互补 PWM 在 “波峰 / 波谷” 的 TCNT 捕捉运行 418
11.5 中断源 419
11.5.1 中断源和优先级 419
11.5.2 DMAC 的启动 421
11.5.3 A/D 转换器的启动 421
11.6 运行时序 423
11.6.1 输入 / 输出时序 423
11.6.2 中断信号的时序 428
11.7 使用时的注意事项 431
11.7.1 模块待机模式的设定 431
11.7.2 输入时钟的限制事项 431
11.7.3 周期设定的注意事项 431
11.7.4 TCNT 的写入和清除的竞争 432
11.7.5 TCNT 的写入和递增计数的竞争 432
11.7.6 TGR 的写入和比较匹配的竞争 433
11.7.7 缓冲寄存器的写入和比较匹配的竞争 434
11.7.8 缓冲寄存器的写入和 TCNT 清除的竞争 435
11.7.9 TGR 的读取和输入捕捉的竞争 436
11.7.10 TGR 的写入和输入捕捉的竞争 436
11.7.11 缓冲寄存器的写入和输入捕捉的竞争 437
11.7.12 级联中 TCNT_2 的写入和上溢 / 下溢的竞争 438
11.7.13 互补 PWM 模式停止时的计数器值 439
11.7.14 互补 PWM 模式中缓冲运行的设定 439
11.7.15 复位同步 PWM 模式的缓冲运行和比较匹配标志 440
11.7.16 复位同步 PWM 模式的上溢标志 441
11.7.17 上溢 / 下溢和计数器清除的竞争 442
11.7.18 TCNT 的写入和上溢 / 下溢的竞争 442
11.7.19 从正常运行或者 PWM 模式 1 转移到复位同步 PWM 模式时的注意事项 443
11.7.20 互补 PWM 模式或者复位同步 PWM 模式的输出电平 443
11.7.21 模块待机时的中断 443
11.7.22 级联中 TCNT_1 和 TCNT_2 的同时输入捕捉 443
11.8 MTU2 输出引脚的初始化方法 444
11.8.1 运行模式 444
11.8.2 开始复位时的运行 444
11.8.3 因运行中的异常等而重新设定时的运行 445
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 445

第 12 章　比较匹配定时器 （CMT） 468
12.1　特点 468
12.2　寄存器说明 469
12.2.1　比较匹配定时器启动寄存器 （CMSTR） 469
12.2.2　比较匹配定时器的控制 / 状态寄存器 （CMCSR） 470
12.2.3　比较匹配计数器 （CMCNT） 471
12.2.4　比较匹配常数寄存器 （CMCOR） 471
12.3　运行说明 471
12.3.1　期间计数运行 471
12.3.2　CMCNT 的计数时序 472
12.4　中断 472
12.4.1　中断源和 DMA 传送请求 472
12.4.2　比较匹配标志的置位时序 472
12.4.3　比较匹配标志的清除时序 473
12.5　使用时的注意事项 473
12.5.1　CMCNT 的写入和比较匹配的竞争 473
12.5.2　CMCNT 的字写入和递增计数的竞争 473
12.5.3　CMCNT 的字节写入和递增计数的竞争 474

第 13 章　看门狗定时器 （WDT） 475
13.1　特点 475
13.2　输入 / 输出引脚 476
13.3　寄存器说明 476
13.3.1　看门狗定时器的计数器 （WTCNT） 476
13.3.2　看门狗定时器的控制 / 状态寄存器 （WTCSR） 477
13.3.3　看门狗复位的控制 / 状态寄存器 （WRCSR） 478
13.3.4　寄存器存取时的注意事项 479
13.4　WDT 的使用方法 480
13.4.1　软件待机模式解除的步骤 480
13.4.2　频率变更的步骤 480
13.4.3　看门狗定时器模式的用法 481
13.4.4　间隔定时器模式的用法 482
13.5　使用时的注意事项 482
13.5.1　定时器误差 482
13.5.2　禁止将 H'FF 设定到 WTCNT 482
13.5.3　间隔定时器上溢标志 482
13.5.4　WDTOVF 信号引起的系统复位 483
13.5.5　看门狗定时器模式的手动复位 483

第 14 章　实时时钟 （RTC） 484
14.1　特点 484
14.2　输入 / 输出引脚 485
14.3　寄存器说明 485
14.3.1　64Hz 计数器 （R64CNT） 486
14.3.2　秒计数器 （RSECCNT） 486
14.3.3　分计数器 （RMINCNT） 487
14.3.4　小时计数器 （RHRCNT） 487
14.3.5　星期计数器 （RWKCNT） 488
14.3.6　日计数器 （RDAYCNT） 488
14.3.7　月计数器 （RMONCNT） 489
14.3.8　年计数器 （RYRCNT） 489
14.3.9　秒报警寄存器 （RSECAR） 490

14.3.10 分报警寄存器 （RMINAR） 490
14.3.11 小时报警寄存器 （RHRAR） 491
14.3.12 星期报警寄存器 （RWKAR） 491
14.3.13 日报警寄存器 （RDAYAR） 492
14.3.14 月报警寄存器 （RMONAR） 492
14.3.15 年报警寄存器 （RYRAR） 493
14.3.16 RTC 控制寄存器 1 （RCR1） 494
14.3.17 RTC 控制寄存器 2 （RCR2） 495
14.3.18 RTC 控制寄存器 3 （RCR3） 496
14.4 运行说明 496
14.4.1 上电后寄存器的初始设定 496
14.4.2 时间的设定步骤 496
14.4.3 时间的读出步骤 497
14.4.4 报警功能 498
14.5 使用时的注意事项 499
14.5.1 关于 RTC 计数运行时的寄存器写入 499
14.5.2 关于实时时钟 （RTC）周期中断的使用 499
14.5.3 关于寄存器设定后的待机转移 499
14.5.4 RTC 用晶体振荡电路 500

第 15 章 带 FIFO 的串行通信接口 （SCIF） 501
15.1 特点 501
15.2 输入 / 输出引脚 502
15.3 寄存器说明 503
15.3.1 接收移位寄存器 （SCRSR） 504
15.3.2 接收 FIFO 数据寄存器 （SCFRDR） 504
15.3.3 发送移位寄存器 （SCTSR） 505
15.3.4 发送 FIFO 数据寄存器 （SCFTDR） 505
15.3.5 串行模式寄存器 （SCSMR） 505
15.3.6 串行控制寄存器 （SCSCR） 507
15.3.7 串行状态寄存器 （SCFSR） 509
15.3.8 位速率寄存器 （SCBRR） 513
15.3.9 FIFO 控制寄存器 （SCFCR） 521
15.3.10 FIFO 数据计数置位寄存器 （SCFDR） 523
15.3.11 串行端口寄存器 （SCSPTR） 523
15.3.12 线路状态寄存器 （SCLSR） 525
15.3.13 串行扩展模式寄存器 （SCEMR） 526
15.4 运行说明 527
15.4.1 概要 527
15.4.2 异步模式中的运行 528
15.4.3 时钟同步模式中的运行 535
15.5 SCIF 的中断 542
15.6 使用时的注意事项 543
15.6.1 SCFTDR 的写操作和 TDFE 标志 543
15.6.2 SCFRDR 的读操作和 RDF 标志 543
15.6.3 DMAC 使用时的注意事项 543
15.6.4 断点的检测和处理 543
15.6.5 断点的发送 543
15.6.6 异步模式的接收数据采样时序和接收容限 544
15.6.7 选择异步基本时钟 544

第 16 章　同步串行通信单元 （SSU） 545
16.1　特点 545
16.2　输入 / 输出引脚 546
16.3　寄存器说明 547
16.3.1　SS 控制寄存器 H （SSCRH） 548
16.3.2　SS 控制寄存器 L （SSCRL） 549
16.3.3　SS 模式寄存器 （SSMR） 550
16.3.4　SS 允许寄存器 （SSER） 551
16.3.5　SS 状态寄存器 （SSSR） 552
16.3.6　SS 控制寄存器 2 （SSCR2） 554
16.3.7　SS 发送数据寄存器 0 ～ 3 （SSTDR0 ～ SSTDR3） 555
16.3.8　SS 接收数据寄存器 0 ～ 3 （SSRDR0 ～ SSRDR3） 556
16.3.9　SS 移位寄存器 （SSTRSR） 557
16.4　运行说明 557
16.4.1　传送时钟 557
16.4.2　时钟的相位、极性和数据的关系 557
16.4.3　数据输入 / 输出引脚和移位寄存器的关系 558
16.4.4　各通信模式和引脚功能 559
16.4.5　SSU 模式 560
16.4.6　SCS 引脚控制和冲突错误 567
16.4.7　时钟同步通信模式 568
16.5　SSU 的中断源和 DMAC 573
16.6　使用时的注意事项 573
16.6.1　模块待机模式的设定 573

第 17 章　I²C 总线接口 3 （IIC3） 574
17.1　特点 574
17.2　输入 / 输出引脚 576
17.3　寄存器说明 577
17.3.1　I²C 总线控制寄存器 1 （ICCR1） 578
17.3.2　I²C 总线控制寄存器 2 （ICCR2） 580
17.3.3　I²C 总线模式寄存器 （ICMR） 581
17.3.4　I²C 总线中断允许寄存器 （ICIER） 582
17.3.5　I²C 总线状态寄存器 （ICSR） 584
17.3.6　从属地址寄存器 （SAR） 586
17.3.7　I²C 总线发送数据寄存器 （ICDRT） 586
17.3.8　I²C 总线接收数据寄存器 （ICDRR） 586
17.3.9　I²C 总线移位寄存器 （ICDRS） 587
17.3.10　NF2CYC 寄存器 （NF2CYC） 587
17.4　运行说明 588
17.4.1　I²C 总线格式 588
17.4.2　主发送运行 589
17.4.3　主接收运行 590
17.4.4　从属发送运行 592
17.4.5　从属接收运行 593
17.4.6　时钟同步串行格式 595
17.4.7　噪声消除电路 597
17.4.8　使用例 598
17.5　中断请求 601
17.6　位同步电路 602
17.7　使用时的注意事项 603
17.7.1　设定 ICCR1.CKS[3:0] 时的注意事项 603

17.7.2 多主操作时的注意事项 603
17.7.3 主接收模式时的注意事项 603
17.7.4 主接收模式、 ACKBT 设定时的注意事项 603
17.7.5 仲裁失败时 MST 和 TRN 位状态的注意事项 603

第 18 章 串行音频接口 （SSI） 604
18.1 特点 604
18.2 输入 / 输出引脚 606
18.3 寄存器的说明 607
18.3.1 控制寄存器 （SSICR） 608
18.3.2 状态寄存器 （SSISR） 612
18.3.3 发送数据寄存器 （SSITDR） 615
18.3.4 接收数据寄存器 （SSIRDR） 615
18.4 运行说明 616
18.4.1 总线格式 616
18.4.2 非压缩模式 616
18.4.3 运行模式 623
18.4.4 发送运行 624
18.4.5 接收运行 626
18.4.6 在发送时的暂时停止、重新开始步骤 628
18.4.7 串行位时钟控制 628
18.5 使用时的注意事项 629
18.5.1 在接收 DMA 运行中溢出时的限制事项 629
18.5.2 使用过采样时钟时的注意事项 629

第 19 章 控制器局域网 （RCAN-TL1） 630
19.1 特点 630
19.1.1 RCAN-TL1 的特点 630
19.1.2 本单片机的特点 631
19.2 构成 631
19.2.1 框图 631
19.2.2 各块的功能 632
19.2.3 引脚构成 633
19.2.4 存储器映像 634
19.3 邮箱 635
19.3.1 邮箱的构成 635
19.3.2 信息控制字段 639
19.3.3 局部接收过滤器屏蔽 （LAFM） 642
19.3.4 信息数据字段 643
19.3.5 时戳 644
19.3.6 发送触发时间 （TTT）与时间触发控制 645
19.4 RCAN-TL1 的控制寄存器 648
19.4.1 主控制寄存器 （MCR） 648
19.4.2 通用状态寄存器 （GSR） 652
19.4.3 位结构寄存器 0、 1 （BCR0、 BCR1） 653
19.4.4 中断请求寄存器 （IRR） 657
19.4.5 中断屏蔽寄存器 （IMR） 661
19.4.6 发送错误计数器 （TEC） / 接收错误计数器 （REC） 662
19.5 RCAN-TL1 邮箱寄存器 663
19.5.1 发送等候寄存器 1、 0 （TXPR1、 TXPR0） 664
19.5.2 发送取消寄存器 1、 0 （TXCR1、 TXCR0） 666
19.5.3 发送应答寄存器 1、 0 （TXACK1、 TXACK0） 667

19.5.4 中断应答寄存器 1、 0 （ABACK1、 ABACK0） 668
19.5.5 数据帧接收完成寄存器 1、 0 （RXPR1、 RXPR0） 669
19.5.6 远程帧接收完成寄存器 1、 0 （RFPR1、 RFPR0） 670
19.5.7 邮箱中断屏蔽寄存器 1、 0 （MBIMR1、 MBIMR0） 671
19.5.8 未读信息状态寄存器 1、 0 （UMSR1、 UMSR0） 672
19.6 定时器寄存器 673
19.6.1 定时触发控制寄存器 0 （TTCR0） 674
19.6.2 最大周期 /Tx_Enable_Window 寄存器 （CMAX_TEW） 676
19.6.3 基准触发偏移寄存器 （RFTROFF） 677
19.6.4 定时器状态寄存器 （TSR） 678
19.6.5 周期计数器寄存器 （CCR） 680
19.6.6 定时器计数器寄存器 （TCNTR） 681
19.6.7 周期时间寄存器 （CYCTR） 682
19.6.8 基准标记寄存器 （RFMK） 682
19.6.9 定时器比较匹配寄存器 0 ～ 2 （TCMR0 ～ 2） 682
19.6.10 发送触发时间选择寄存器 （TTTSEL） 684
19.7 运行说明 684
19.7.1 RCAN-TL1 的设定 684
19.7.2 测试模式的设定 688
19.7.3 信息发送顺序 689
19.7.4 信息接收顺序 699
19.7.5 邮箱的再设定 700
19.8 中断源 702
19.9 DMAC 接口 703
19.10 CAN 总线接口 704
19.11 RCAN-TL1 引脚端口设定 704
19.12 使用时的注意事项 705
19.12.1 关于 1 个通道 64 个邮箱的端口设定注意事项 705

第 20 章 IEBus™ 控制器 （IEB） 706
20.1 特点 706
20.1.1 IEBus 通信协议 707
20.1.2 传输协议 709
20.1.3 传输数据 （数据字段的内容） 713
20.1.4 位格式 716
20.1.5 构成 717
20.2 输入 / 输出引脚 718
20.3 寄存器的说明 719
20.3.1 IEBus 控制寄存器 （IECTR） 720
20.3.2 IEBus 命令寄存器 （IECMR） 721
20.3.3 IEBus 主控制寄存器 （IEMCR） 722
20.3.4 IEBus 主单元地址寄存器 1 （IEAR1） 723
20.3.5 IEBus 主单元地址寄存器 2 （IEAR2） 723
20.3.6 IEBus 从属地址设定寄存器 1 （IESA1） 724
20.3.7 IEBus 从属地址设定寄存器 2 （IESA2） 724
20.3.8 IEBus 发送信息长度寄存器 （IETBFL） 725
20.3.9 IEBus 接收基本地址寄存器 1 （IEMA1） 725
20.3.10 IEBus 接收基本地址寄存器 2 （IEMA2） 726
20.3.11 IEBus 接收控制字段寄存器 （IERCTL） 726
20.3.12 IEBus 接收信息长度寄存器 （IERBFL） 727
20.3.13 IEBus 锁定地址寄存器 1 （IELA1） 727
20.3.14 IEBus 锁定地址寄存器 2 （IELA2） 727

20.3.15 IEBus 通用标志寄存器 （IEFLG） 728
20.3.16 IEBus 发送状态寄存器 （IETSR） 730
20.3.17 IEBus 发送中断允许寄存器 （IEIET） 732
20.3.18 IEBus 接收状态寄存器 （IERSR） 733
20.3.19 IEBus 接收中断允许寄存器 （IEIER） 736
20.3.20 IEBus 时钟选择寄存器 （IECKSR） 737
20.3.21 IEBus 发送数据缓冲器 001 ～ 128 （IETB001 ～ IETB128） 738
20.3.22 IEBus 接收数据缓冲器 001 ～ 128 （IERB001 ～ IERB128） 738
20.4 数据格式 739
20.4.1 发送格式 739
20.4.2 接收格式 740
20.5 软件控制流程 741
20.5.1 初始设定 741
20.5.2 主发送 742
20.5.3 从属接收 743
20.5.4 主接收 744
20.5.5 从属发送 745
20.6 运行时序 746
20.6.1 主发送 746
20.6.2 从属接收 747
20.6.3 主接收 748
20.6.4 从属发送 749
20.7 中断源 750
20.8 使用时的注意事项 751
20.8.1 最大传输字节长度内，通信没结束时的注意事项 751

第 21 章 CD-ROM 解码器 （ROM-DEC） 753
21.1 特点 753
21.1.1 ROM-DEC 格式 753
21.2 框图 754
21.3 寄存器的说明 758
21.3.1 ROMDEC 允许控制寄存器 （CROMEN） 760
21.3.2 同步码的同步控制寄存器 （CROMSY0） 761
21.3.3 解码模式控制寄存器 （CROMCTL0） 762
21.3.4 EDC、 ECC 校验的控制寄存器 （CROMCTL1） 763
21.3.5 解码处理自动停止控制寄存器 （CROMCTL3） 764
21.3.6 解码选项设定控制寄存器 （CROMCTL4） 765
21.3.7 HEAD20 ～ 22 表示控制寄存器 （CROMCTL5） 766
21.3.8 同步码状态寄存器 （CROMST0） 766
21.3.9 ECC 后的扇区头对应的错误状态寄存器 （CROMST1） 767
21.3.10 ECC 后子头对应的错误状态寄存器 （CROMST3） 767
21.3.11 扇区头、子头数据妥当性判定状态寄存器 （CROMST4） 768
21.3.12 模式判定结果和连接扇区检测状态寄存器 （CROMST5） 769
21.3.13 ECC、 EDC 错误状态寄存器 （CROMST6） 770
21.3.14 缓冲器状态寄存器 （CBUFST0） 771
21.3.15 解码中止源状态寄存器 （CBUFST1） 771
21.3.16 缓冲溢出状态寄存器 （CBUFST2） 772
21.3.17 ECC 校正前扇区头 -MINUTES 数据寄存器 （HEAD00） 772
21.3.18 ECC 校正前扇区头 -SECONDS 数据寄存器 （HEAD01） 772
21.3.19 ECC 校正前扇区头 -FRAMES （1/75 秒）数据寄存器 （HEAD02） 773
21.3.20 ECC 校正前扇区头 -MODE 数据寄存器 （HEAD03） 773
21.3.21 ECC 校正前子头 - 文件号码 （BYTE-16）数据寄存器 （SHEAD00） 773

21.3.22 ECC 校正前子头 - 通道号码 （BYTE-17）数据寄存器 （SHEAD01） 774
21.3.23 ECC 校正前子头 - 子模式 （BYTE-18）数据寄存器 （SHEAD02） 774
21.3.24 ECC 校正前子头 - 数据类型 （BYTE-19）数据寄存器 （SHEAD03） 774
21.3.25 ECC 校正前子头 - 文件号码 （BYTE-20）数据寄存器 （SHEAD04） 775
21.3.26 ECC 校正前子头 - 通道号码 （BYTE-21）数据寄存器 （SHEAD05） 775
21.3.27 ECC 校正前子头 - 子模式 （BYTE-22）数据寄存器 （SHEAD06） 775
21.3.28 ECC 校正前子头 - 数据类型 （BYTE-23）数据寄存器 （SHEAD07） 776
21.3.29 ECC 校正后扇区头 -MINUTES 数据寄存器 （HEAD20） 776
21.3.30 ECC 校正后扇区头 -SECONDS 数据寄存器 （HEAD21） 776
21.3.31 ECC 校正后扇区头 -FRAMES （1/75 秒）数据寄存器 （HEAD22） 777
21.3.32 ECC 校正后扇区头 -MODE 数据寄存器 （HEAD23） 777
21.3.33 ECC 校正后子头 - 文件号码 （BYTE-16）数据寄存器 （SHEAD20） 777
21.3.34 ECC 校正后子头 - 通道号码 （BYTE-17）数据寄存器 （SHEAD21） 778
21.3.35 ECC 校正后子头 - 子模式 （BYTE-18）数据寄存器 （SHEAD22） 778
21.3.36 ECC 校正后子头 - 数据类型 （BYTE-19）数据寄存器 （SHEAD23） 778
21.3.37 ECC 校正后子头 - 文件号码 （BYTE-20）数据寄存器 （SHEAD24） 779
21.3.38 ECC 校正后子头 - 通道号码 （BYTE-21）数据寄存器 （SHEAD25） 779
21.3.39 ECC 校正后子头 - 子模式 （BYTE-22）数据寄存器 （SHEAD26） 779
21.3.40 ECC 校正后子头 - 数据类型 （BYTE-23）数据寄存器 （SHEAD27） 780
21.3.41 自动缓冲设定控制寄存器 （CBUFCTL0） 780
21.3.42 自动缓冲开始扇区设定 -MINUTES 控制寄存器 （CBUFCTL1） 781
21.3.43 自动缓冲开始扇区设定 -SECONDS 控制寄存器 （CBUFCTL2） 781
21.3.44 自动缓冲开始扇区设定 -FRAMES 控制寄存器 （CBUFCTL3） 781
21.3.45 ISY 中断源屏蔽控制寄存器 （CROMST0M） 782
21.3.46 CD-ROM 解码器模块复位控制寄存器 （ROMDECRST） 782
21.3.47 CD-ROM 解码器模块复位状态寄存器 （RSTSTAT） 783
21.3.48 SSI 数据控制寄存器 （SSI） 784
21.3.49 中断标志寄存器 （INTHOLD） 785
21.3.50 中断源屏蔽控制寄存器 （INHINT） 786
21.3.51 CD-ROM 解码器数据流输入寄存器 （STRMDIN0） 787
21.3.52 CD-ROM 解码器数据流输入寄存器 （STRMDIN2） 787
21.3.53 CD-ROM 解码器数据流输出寄存器 （STRMDOUT0） 787
21.4 运行说明 788
21.4.1 输入数据流字节序变换功能 788
21.4.2 同步码保护功能 789
21.4.3 纠错 792
21.4.4 自动解码停止功能 793
21.4.5 缓冲格式 793
21.4.6 目标扇区缓冲功能 795
21.5 中断源 797
21.5.1 中断及 DMA 传送请求信号 797
21.5.2 状态寄存器的更新时序 798
21.6 使用时的注意事项 799
21.6.1 在解码运行状态，仅缓冲停止 / 再次开始时的注意事项 799
21.6.2 同步码状态寄存器 （CROMST0）设定时的注意 799
21.6.3 连接块的注意 799
21.6.4 CD-DSP 停止 / 再次开始时的注意点 800
21.6.5 IREADY 标志清除的注意点 800
21.6.6 在数据流传输时的注意点 （1） 800
21.6.7 在数据流传输时的注意点 （2） 800

第 22 章　A/D 转换器 （ADC） 801
22.1　特点 801
22.2　输入 / 输出引脚 802
22.3　寄存器的说明 802
22.3.1　A/D 数据寄存器 A ～ H （ADDRA ～ ADDRH） 803
22.3.2　A/D 控制 / 状态寄存器 （ADCSR） 804
22.4　运行说明 806
22.4.1　单通道模式 806
22.4.2　多通道模式 808
22.4.3　扫描模式 810
22.4.4　通过外部触发、 MTU2 启动 A/D 转换器 812
22.4.5　输入采样和 A/D 转换时间 812
22.4.6　外部触发输入时序 813
22.5　中断源和 DMAC 传送请求 814
22.6　A/D 转换精度的定义 815
22.7　使用时的注意事项 816
22.7.1　模块待机模式的设定 816
22.7.2　模拟电压的设定 816
22.7.3　电路板设计的注意事项 816
22.7.4　模拟输入引脚的处理 816
22.7.5　容许信号源阻抗 818
22.7.6　对绝对精度的影响 818
22.7.7　深度待机模式时的 A/D 转换 818
22.7.8　使用扫描模式以及多通道模式时的注意事项 818

第 23 章　D/A 转换器 （DAC） 819
23.1　特点 819
23.2　输入 / 输出引脚 819
23.3　寄存器的说明 820
23.3.1　D/A 数据寄存器 0、 1 （DADR0、 DADR1） 820
23.3.2　D/A 控制寄存器 （DACR） 820
23.4　运行说明 821
23.5　使用时的注意事项 822
23.5.1　模块待机模式的设定 822
23.5.2　软件待机模式时的 D/A 输出保持功能 822
23.5.3　模拟电压的设定 822
23.5.4　深度待机模式时的 D/A 转换 822

第 24 章　AND/NAND 闪存控制器 （FLCTL） 823
24.1　特点 823
24.2　输入 / 输出引脚 826
24.3　寄存器的说明 827
24.3.1　通用控制寄存器 （FLCMNCR） 828
24.3.2　命令控制寄存器 （FLCMDCR） 830
24.3.3　命令码寄存器 （FLCMCDR） 831
24.3.4　地址寄存器 （FLADR） 832
24.3.5　地址寄存器 2 （FLADR2） 833
24.3.6　数据计数器寄存器 （FLDTCNTR） 834
24.3.7　数据寄存器 （FLDATAR） 835
24.3.8　中断 DMA 控制寄存器 （FLINTDMACR） 836
24.3.9　就绪忙超时设定寄存器 （FLBSYTMR） 839
24.3.10　就绪忙超时计数器 （FLBSYCNT） 840

24.3.11 数据 FIFO 寄存器 （FLDTFIFO） 841
24.3.12 管理码 FIFO 寄存器 （FLECFIFO） 841
24.3.13 传送控制寄存器 （FLTRCR） 842
24.4 运行说明 843
24.4.1 存取步骤 843
24.4.2 运行模式 844
24.4.3 寄存器设定步骤 844
24.4.4 命令存取模式 845
24.4.5 扇区存取模式 849
24.4.6 ECC 的纠错 851
24.4.7 状态读 851
24.5 中断处理 853
24.6 DMA 传送的设定 853

第 25 章 USB2.0 主机 / 功能模块 （USB） 854
25.1 特点 854
25.2 输入 / 输出引脚 856
25.3 寄存器说明 857
25.3.1 系统结构控制寄存器 （SYSCFG） 859
25.3.2 系统结构状态寄存器 （SYSSTS） 860
25.3.3 器件状态控制寄存器 （DVSTCTR） 862
25.3.4 测试模式寄存器 （TESTMODE） 864
25.3.5 FIFO 端口结构寄存器 （CFBCFG、 D0FBCFG、 D1FBCFG） 865
25.3.6 FIFO 端口寄存器 （CFIFO、 D0FIFO、 D1FIFO） 867
25.3.7 FIFO 端口选择寄存器 （CFIFOSEL、 D0FIFOSEL、 D1FIFOSEL） 868
25.3.8 FIFO 端口控制寄存器 （CFIFOCTR、 D0FIFOCTR、 D1FIFOCTR） 872
25.3.9 CFIFO 端口 SIE 寄存器 （CFIFOSIE） 873
25.3.10 处理计数器寄存器 （D0FIFOTRN、 D1FIFOTRN） 874
25.3.11 中断允许寄存器 0 （INTENB0） 874
25.3.12 中断允许寄存器 1 （INTENB1） 876
25.3.13 BRDY 中断允许寄存器 （BRDYENB） 877
25.3.14 NRDY 中断允许寄存器 （NRDYENB） 878
25.3.15 BEMP 中断允许寄存器 （BEMPENB） 879
25.3.16 中断状态寄存器 0 （INTSTS0） 880
25.3.17 中断状态寄存器 1 （INTSTS1） 882
25.3.18 BRDY 中断状态寄存器 （BRDYSTS） 884
25.3.19 NRDY 中断状态寄存器 （NRDYSTS） 885
25.3.20 BEMP 中断状态寄存器 （BEMPSTS） 886
25.3.21 帧号寄存器 （FRMNUM） 887
25.3.22 μ 帧号寄存器 （UFRMNUM） 888
25.3.23 USB 地址寄存器 （USBADDR） 889
25.3.24 USB 请求类型寄存器 （USBREQ） 889
25.3.25 USB 请求价值寄存器 （USBVAL） 890
25.3.26 USB 请求变址寄存器 （USBINDX） 890
25.3.27 USB 请求长度寄存器 （USBLENG） 890
25.3.28 DCP 结构寄存器 （DCPCFG） 891
25.3.29 DCPMax 包大小寄存器 （DCPMAXP） 892
25.3.30 DCP 控制寄存器 （DCPCTR） 893
25.3.31 管道窗口选择寄存器 （PIPESEL） 895
25.3.32 管道结构寄存器 （PIPECFG） 896
25.3.33 管道缓冲器指定寄存器 （PIPEBUF） 898
25.3.34 管道 Max 包大小寄存器 （PIPEMAXP） 899

25.3.35 管道周期控制寄存器 （PIPEPERI） 900
25.3.36 管道 n 控制寄存器 （PIPEnCTR）（n=1 ～ 7） 901
25.3.37 USB AC 特性转换寄存器 （USBACSWR） 902
25.4 运行说明 903
25.4.1 系统控制及振荡控制 903
25.4.2 中断功能 905
25.4.3 管道控制 916
25.4.4 缓冲存储器 921
25.4.5 控制传送 （DCP） 931
25.4.6 批量传送 （管道 1 ～ 5） 933
25.4.7 中断传送 （管道 6、 7） 935
25.4.8 同步传送 （管道 1、 2） 936
25.4.9 SOF 插值功能 941
25.4.10 管道安排 942
25.5 使用时的注意事项 943
25.5.1 使用同步 OUT 传送时的注意事项 943
25.5.2 USB 收发器的设定步骤 943
25.5.3 清除中断源的时序 944

第 26 章 LCD 控制器 （LCDC） 945
26.1 特点 945
26.2 输入 / 输出引脚 946
26.3 寄存器的说明 947
26.3.1 LCDC 输入时钟寄存器 （LDICKR） 948
26.3.2 LCDC 模块类型寄存器 （LDMTR） 949
26.3.3 LCDC 数据格式寄存器 （LDDFR） 951
26.3.4 LCDC 扫描模式寄存器 （LDSMR） 952
26.3.5 LCDC 上部显示面板用数据输入开始地址寄存器 （LDSARU） 953
26.3.6 LCDC 下部显示面板用数据输入开始地址寄存器 （LDSARL） 954
26.3.7 LCDC 显示面板用输入数据行地址偏移量寄存器 （LDLAOR） 954
26.3.8 LCDC 调色板控制寄存器 （LDPALCR） 955
26.3.9 调色板数据寄存器 00 ～ FF （LDPR00 ～ LDPRFF） 955
26.3.10 LCDC 水平字符数寄存器 （LDHCNR） 956
26.3.11 LCDC 水平同步信号寄存器 （LDHSYNR） 957
26.3.12 LCDC 垂直显示行数寄存器 （LDVDLNR） 958
26.3.13 LCDC 垂直总行数寄存器 （LDVTLNR） 959
26.3.14 LCDC 垂直同步信号寄存器 （LDVSYNR） 960
26.3.15 LCDCAC 调制信号交替行数寄存器 （LDACLNR） 961
26.3.16 LCDC 中断控制寄存器 （LDINTR） 961
26.3.17 LCDC 电源管理模式寄存器 （LDPMMR） 963
26.3.18 LCDC 通电顺序期间寄存器 （LDPSPR） 964
26.3.19 LCDC 控制寄存器 （LDCNTR） 965
26.3.20 LCDC 用户指定中断控制寄存器 （LDUINTR） 966
26.3.21 LCDC 用户指定中断行数寄存器 （LDUINTLNR） 967
26.3.22 LCDC 存储器存取间隔号寄存器 （LDLIRNR） 968
26.4 运行说明 969
26.4.1 关于用 LCDC 能显示的液晶模块的尺寸 969
26.4.2 旋转显示分辨率 / 突发长度及连接存储器 （SDRAM）的限制事项 970
26.4.3 关于彩色调色板规格 976
26.4.4 数据格式 977
26.4.5 显示分辨率的设定 980
26.4.6 电源控制顺序处理 980

26.4.7 硬件旋转运行说明 984
26.5 时钟和 LCD 数据信号例子 986
26.6 使用时的注意事项 994
26.6.1 关于存取用于存储显示数据的 VRAM （区域 3 的同步 DRAM）时的停止步骤 994

第 27 章 采样频率转换器 （SRC） 995
27.1 特点 995
27.2 寄存器的说明 996
27.2.1 SRC 输入数据寄存器 （SRCID） 996
27.2.2 SRC 输出数据寄存器 （SRCOD） 997
27.2.3 SRC 输入数据控制寄存器 （SRCIDCTRL） 998
27.2.4 SRC 输出数据控制寄存器 （SRCODCTRL） 999
27.2.5 SRC 控制寄存器 （SRCCTRL） 1000
27.2.6 SRC 状态寄存器 （SRCSTAT） 1002
27.3 运行说明 1003
27.3.1 初始设定 1003
27.3.2 数据输入 1004
27.3.3 数据输出 1005
27.4 中断 1006
27.5 使用时的注意事项 1006

第 28 章 SD 主机接口 （SDHI） 1007

第 29 章 引脚功能控制器 （PFC） 1008
29.1 特点 1013
29.2 寄存器的说明 1014
29.2.1 端口 B/IO 寄存器 L （PBIORL） 1015
29.2.2 端口 B 控制寄存器 L1 ～ L4 （PBCRL1 ～ PBCRL4） 1015
29.2.3 端口 C/IO 寄存器 L （PCIORL） 1019
29.2.4 端口 C 控制寄存器 L1 ～ L4 （PCCRL1 ～ PCCRL4） 1019
29.2.5 端口 D/IO 寄存器 L （PDIORL） 1023
29.2.6 端口 D 控制寄存器 L1 ～ L4 （PDCRL1 ～ PDCRL4） 1023
29.2.7 端口 E/IO 寄存器 L （PEIORL） 1030
29.2.8 端口 E 控制寄存器 L1 ～ L4 （PECRL1 ～ PECRL4） 1031
29.2.9 端口 F/IO 寄存器 H、 L （PFIORH、 PFIORL） 1035
29.2.10 端口 F 控制寄存器 H1 ～ H4、 L1 ～ L4 （PFCRH1 ～ PFCRH4、 PFCRL1 ～ PFCRL4） 1035
29.2.11 RQOUT 功能控制寄存器 （IFCR） 1043
29.2.12 SSI 过采样时钟选择寄存器 （SCSR） 1043
29.3 关于端口 A 的功能切换 1044
29.4 使用时的注意事项 1045

第 30 章 I/O 端口 1046
30.1 特点 1046
30.2 端口 A 1046
30.2.1 寄存器说明 1046
30.2.2 端口 A 数据寄存器 L （PADRL） 1047
30.3 端口 B 1048
30.3.1 寄存器说明 1048
30.3.2 端口 B 数据寄存器 L （PBDRL） 1048
30.3.3 端口 B 端口寄存器 L （PBPRL） 1050
30.4 端口 C 1050
30.4.1 寄存器说明 1051

30.4.2 端口 C 数据寄存器 L （PCDRL） 1051
30.4.3 端口 C 端口寄存器 L （PCPRL） 1052
30.5 端口 D 1053
30.5.1 寄存器说明 1053
30.5.2 端口 D 数据寄存器 L （PDDRL） 1054
30.5.3 端口 D 端口寄存器 L （PDPRL） 1055
30.6 端口 E 1056
30.6.1 寄存器说明 1056
30.6.2 端口 E 数据寄存器 L （PEDRL） 1057
30.6.3 端口 E 端口寄存器 L （PEPRL） 1058
30.7 端口 F 1059
30.7.1 寄存器说明 1059
30.7.2 端口 F 数据寄存器 H、 L （PFDRH、 PFDRL） 1060
30.7.3 端口 F 端口寄存器 H、 L （PFPRH、 PFPRL） 1062
30.8 使用时的注意事项 1064

第 31 章 内置 RAM 1065
31.1 特点 1065
31.2 使用时的注意事项 1066
31.2.1 页竞争 1066
31.2.2 关于 RAME 位、 RAMWE 位 1066
31.2.3 指令配置禁止领域 1066

第 32 章 低功耗模式 1067
32.1 特点 1067
32.1.1 低功耗模式的种类 1067
32.2 寄存器的说明 1068
32.2.1 待机控制寄存器 （STBCR） 1068
32.2.2 待机控制寄存器 2 （STBCR2） 1069
32.2.3 待机控制寄存器 3 （STBCR3） 1070
32.2.4 待机控制寄存器 4 （STBCR4） 1071
32.2.5 待机控制寄存器 5 （STBCR5） 1072
32.2.6 待机控制寄存器 6 （STBCR6） 1073
32.2.7 系统控制寄存器 1 （SYSCR1） 1074
32.2.8 系统控制寄存器 2 （SYSCR2） 1075
32.2.9 系统控制寄存器 3 （SYSCR3） 1076
32.2.10 深度待机控制寄存器 （DSCTR） 1077
32.2.11 深度待机控制寄存器 2 （DSCTR2） 1078
32.2.12 深度待机解除源选择寄存器 （DSSSR） 1079
32.2.13 深度待机解除源标志寄存器 （DSFR） 1080
32.2.14 保持用内置 RAM 调整寄存器 （DSRTR） 1081
32.3 运行说明 1082
32.3.1 睡眠模式 1082
32.3.2 软件待机模式 1082
32.3.3 软件待机模式的应用例子 1084
32.3.4 深度待机模式 1085
32.3.5 模块待机功能 1088
32.4 使用时的注意事项 1088

第 33 章 用户调试接口 （H-UDI） 1089
33.1 特点 1089
33.2 输入 / 输出引脚 1090

33.3 寄存器说明 1090
33.3.1 旁路寄存器 （SDBPR） 1090
33.3.2 指令寄存器 （SDIR） 1091
33.4 运行说明 1092
33.4.1 TAP 控制器 1092
33.4.2 复位结构 1093
33.4.3 TDO 输出时序 1093
33.4.4 H-UDI 复位 1094
33.4.5 H-UDI 中断 1094
33.5 使用时的注意事项 1094

第 34 章 寄存器一览 1095
34.1 寄存器地址一览 （按功能模块、手册章节编号顺序） 1096
34.2 寄存器位一览 1118
34.3 各运行模式下的寄存器状态一览 1165

第 35 章 电特性 1167
35.1 绝对最大额定值 1167
35.2 通电、断电顺序 1168
35.3 DC 特性 1169
35.4 AC 特性 1174
35.4.1 时钟时序 1175
35.4.2 控制信号时序 1178
35.4.3 总线时序 1180
35.4.4 UBC 时序 1214
35.4.5 DMAC 时序 1214
35.4.6 MTU2 时序 1215
35.4.7 WDT 时序 1216
35.4.8 SCIF 时序 1216
35.4.9 SSU 时序 1217
35.4.10 IIC3 时序 1220
35.4.11 SSI 时序 1221
35.4.12 RCAN-TL1 时序 1222
35.4.13 ADC 时序 1223
35.4.14 FLCTL 时序 1224
35.4.15 USB 时序 1232
35.4.16 LCDC 时序 1234
35.4.17 SDHI 时序 1235
35.4.18 I/O 端口时序 1236
35.4.19 H-UDI 时序 1237
35.4.20 AC 特性测量条件 1238
35.5 A/D 转换器特性 1238
35.6 D/A 转换器特性 1239
35.7 使用时的注意事项 1239

附录 1240
附录 A. 引脚状态 1240
附录 B. 外形尺寸图 1245

索引 1246

第 1 章 概要

1.1 SH7263 的特点

本 LSI 是以瑞萨科技独创的 RISC （精简指令系统计算机）方式 CPU 为内核，集成了系统构成所需外围功能的 RISC 单片机。

本 LSI 的 CPU 采用 SH-2A CPU，该 CPU 的特点是与 SH-1、SH-2、SH-2E 微型计算机在目标代码级向上兼容。拥有 RISC 方式的指令系统并且采用超标量体系结构和哈佛体系结构，因此大大提高了指令的执行速度。另外，采用与直接存储器存取控制器 （DMAC）独立的内部 32 位总线结构，强化了数据处理功能。通过使用本 LSI 的 CPU 可实现以前的单片机不能实现的高速性实时控制等应用功能，并且能构成更低成本、更高性能和更高功能的系统。

本 LSI 内置浮点运算单元（FPU）和高速缓存。而且作为系统构成所必需的外围功能，本 LSI 内置了 64KB 的高速内置 RAM 和用于 16KB 数据保持的 RAM、多功能定时脉冲单元 2（MTU2）、比较匹配定时器（CMT）、实时时钟 （RTC）、带 FIFO 的串行通信接口 （SCIF）、同步串行通信单元 （SSU）、I^2C 总线接口 3 （IIC3）、串行音频接口 （SSI）、控制器局域网 （RCAN-TL1）、IEBus™*1 控制器 （IEB）*2、CD-ROM 解码器 （ROM-DEC）、A/D 转换器、D/A 转换器、AND/NAND 闪存控制器 （FLCTL）、USB2.0 主机 / 功能模块 （USB）、取样频率转换器 （SRC）、 SD 主机接口 （SDHI）*3、中断控制器 （INTC）、 I/O 端口等。

通过外部存储器存取支持功能，本 LSI 可直接连接到存储器和外围 LSI，从而能大幅降低系统的成本。而且在本 LSI 的 I/O 引脚带有抑制引脚的电位变为中间电位状态的弱保持电路，因此不需要固定输入电平的外接电路，可以大幅减少零件数。

本 LSI 的特点如**表 1.1** 所示。

【注】 *1 IEBus™ （Inter Equipment Bus™）是 NEC 电子的商标。
*2 R5S72632P200FP 和 R5S72633P200FP 中内置。
*3 R5S72631P200FP 和 R5S72633P200FP 中内置。

表 1.1　SH7263 的特点

项目	特　　点
CPU	• 瑞萨独创的 SuperH 体系结构 • 在目标代码级与 SH-1、 SH-2、 SH-2E 兼容 • 32 位内部数据总线 • 通用寄存器体系结构 通用寄存器：32 位 ×16 个 控制寄存器：32 位 ×4 个 系统寄存器：32 位 ×4 个 响应高速中断的寄存器组 • RISC 方式的指令系统 （与 SH 系列向上兼容） 指令长度：提高代码效率的 16 位基本指令与提高性能和使用便利性的 32 位指令加载存储结构 延迟转移指令 基于 C 语言的指令系统 • 含 FPU 的 2 条指令同时执行超标量 • 指令执行时间：最大 2 条指令 / 周期 • 地址空间：4G 字节 • 内置乘法器 • 5 段流水线 • 哈佛体系结构
浮点运算单元 （FPU）	• 内置浮点协处理器 • 支持单精度 （32 位）及双精度 （64 位） • 支持依照 IEEE754 的数据类型及异常 • 舍入模式：就近舍入及朝 0 舍入 • 非规格化数的处理：下舍入为 0 • 浮点寄存器 浮点寄存器：32 位 ×16 个 （单精度 ×16 字或双精度 ×8 字） 浮点系统寄存器：32 位 ×2 个 • 支持 FMAC （乘法运算及累加）指令 • 支持 FDIV （除法运算） /FSQRT （平方根）指令 • 支持 FLDI0/FLDI1 （加载常数 0/1）指令 • 指令执行时间 等待时间 （FMAC/FADD/FSUB/FMUL）：3 个周期 （单精度）、 8 个周期 （双精度） 节距 （FMAC/FADD/FSUB/FMUL）：1 个周期 （单精度）、 6 个周期 （双精度） 【注】 FMAC 只支持单精度。 • 5 段流水线
高速缓冲存储器	• 指令高速缓存：8K 字节 • 操作数高速缓存：8K 字节 • 各有 128 个入口、 4 路集联、 16 字节块长 • 回写、直写、 LRU 置换算法 • 有高速缓存锁定功能 （仅限操作数高速缓存）：可锁定路 2、路 3
中断控制器 （INTC）	• 17 个外部中断引脚 （NMI、 IRQ7 ～ IRQ0、 PINT7 ～ PINT0） • 内置外围中断：对每个模块设定优先级 • 可设定 16 个级别的优先级 • 通过寄存器组可在进行中断处理的同时高速执行寄存器的保存 / 返回

项目	特　　点
总线状态控制器 （BSC）	• 地址空间支持 8 个最大 64MB 的区域 （区域 0 ～ 7） • 各区域可独立设定以下功能 总线宽 （8、 16、 32 位）。但每个区域支持的宽度不同 存取等待周期数 （有可给读 / 写单独设定等待的区域） 空闲等待周期的设定 （相同区域 / 其他区域） 按区域指定连接的存储器，可直接连接 SRAM、带字节选择的 SRAM、 SDRAM 及突发 ROM （时钟同步 / 异步）。另外，支持地址 / 数据多路复用 I/O （MPX）接口及突发 MPX-I/O 接口 支持 PCMCIA 接口 对目标区域输出片选信号 （$\overline{CS0}$ ～ $\overline{CS7}$） （可通过编程选择 $\overline{CS}$ 断言 / 取反时序） • SDRAM 刷新功能 支持自动刷新和自刷新模式 • SDRAM 突发存取功能
直接存储器存取控制器 （DMAC）	• 8 个通道。其中 4 个通道可用于外部请求 • 可从内置外围模块启动 • 突发模式和周期挪用模式 • 支持间歇模式 （支持 16/64 周期） • 可自动再装入传送信息
时钟脉冲振荡器 （CPG）	• 时钟模式：可从外部输入 （EXTAL、 CKIO 或 USB_X1）或晶体振荡器选择输入时钟 • 可通过内部 PLL 电路实现输入时钟最大 16 倍频 • 生成 3 种时钟 CPU 时钟：最大 200MHz 总线时钟：最大 66MHz 外围时钟：最大 33MHz
看门狗定时器 （WDT）	• 1 个通道的看门狗定时器 • 通过计数器的溢出，可将本 LSI 复位
低功耗模式	• 本 LSI 为降低功耗，支持 4 种低功耗模式 睡眠模式 软件待机模式 深度待机模式 模块待机模式

项目	特　　点
多功能定时器脉冲单元 2（MTU2）	• 以 16 位定时器的 5 个通道为基础，最多能输入 / 输出 16 种脉冲 • 18 个输出比较寄存器兼输入捕捉寄存器 • 输入捕捉功能 • 脉冲输出模式 交替 /PWM/ 互补 PWM/ 复位同步 PWM • 多个计数器的同步功能 • 互补 PWM 输出模式 输出用于控制 3 相反相器的非重叠波形 自动设定空载时间 可在 0 ～ 100% 任意范围内设定 PWM 占空比 A/D 转换请求的延迟功能 波峰 / 波谷的跳过中断功能 • 复位同步 PWM 模式 3 相输出任意占空比的正相 / 反相 PWM 波形 • 相位计数模式 可处理 2 相编码器的计数
比较匹配定时器 （CMT）	• 2 个通道 16 位计数器 • 可选择 4 种时钟 （Pϕ/8、 Pϕ/32、 Pϕ/128、 Pϕ/512） • 比较匹配时，可选择产生 DMA 传送请求或者中断请求
实时时钟 （RTC）	• 内部时钟、日历功能、报警功能 • 1/256 秒的最大分辨率 （中断周期）的内置 32.768kHz 晶体振荡器电路
带 FIFO 的串行通信接口（SCIF）	• 4 个通道 • 可选择时钟同步 / 异步模式 • 可同时接收和发送 （全双工） • 内置专用波特率发生器 • 内置分别用于接收和发送的 16 字节 FIFO • 调制解调器控制功能 （异步模式时）
同步串行通信单元 （SSU）	• 可选择主模式和从属模式 • 可选择标准模式和双向模式 • 可选择 8/16/32 位长度的接收 / 发送数据 • 可同时进行接收和发送 （全双工） • 可连续进行串行通信 • 2 个通道
I^2C 总线接口 3 （IIC3）	• 4 个通道 • 内置主模式 / 从属模式
串行音频接口 （SSI）	• 4 个通道的双向串行传送 • 支持多种实时音频格式 • 支持主 / 从属功能 • 生成可编程字时钟、位时钟 • 多通道格式功能 • 支持 8/16/18/20/22/24/32 位数据格式

项目	特　　点
控制器局域网 （RCAN-TL1）	• 2 个通道 • 所有的通道均支持 TTCAN 级 1 • 支持 Bosch 2.0B active • 缓冲器大小：接收 / 发送 ×31、发送 ×1 • 为增加 32 个通道输入的缓冲器，可将多个 RCAN 通道分配给 1 个总线 • 可给发送或接收设定 31 个信箱 • CAN 总线的抖动：0.3% （目标值）
IEBus™ 控制器 （IEB） 【注】根据型号分为具有 IEB 的产品和不具有 IEB 的产品	• 支持 IEBus 的协议控制 （层 2） 半双工异步通信 多主模式 广播通信功能 可选择 3 种不同传送速度的模式 • 能连续发送 / 接收字节数最大为 128 字节的用于数据发送 / 接收的内置缓冲器（双端口 RAM），该 128 字节是模式 2 的最大传送字节数。 工作频率： 12MHz、 12.58MHz （使用 Pϕ 或 AUDIO_X1、 AUDIO_X2 时钟的 1/2 分频） 18MHz、 18.87MHz （使用 Pϕ 或 AUDIO_X1、 AUDIO_X2 时钟的 1/3 分频） 24MHz、 25.16MHz （使用 Pϕ 或 AUDIO_X1、 AUDIO_X2 时钟的 1/4 分频） 30MHz、 31.45MHz （使用 Pϕ 或 AUDIO_X1、 AUDIO_X2 时钟的 1/5 分频） 36MHz、 37.74MHz （使用 AUDIO_X1、 AUDIO_X2 时钟的 1/6 分频）
CD-ROM 解码器 （ROM-DEC）	• 支持 Mode0、 Mode1、 Mode2、 Mode2 Form1、 Mode2 Form2 的 5 种格式 • 同步码的检测 / 保护 （保护：无法检测时自动插入） • 解扰 • ECC 校正 P、 Q、 PQ、 QP 校正 PQ、 QP 校正最多可重复 3 次 • EDC 校验 ECC 前后校验 • Mode、 Form 自动检测 • 连接扇区自动检测 • 缓冲数据控制 将同步码以后的 CD-ROM 数据 （为解扰、 ECC 校正、 EDC 校验后的数据）通过特定格式传送
AND/NAND 闪存控制器 （FLCTL）	• 与 AND/NAND 型闪存直接连接的存储器接口 • 扇区单位的读取 / 写入 • 指令存取模式和扇区存取模式 （512 字节数据 +16 字节管理码：带 ECC） 2 种传送模式 • 有中断请求、 DMAC 传送请求 • 支持 5 字节地址 （超过 2G 位）的闪存
USB 2.0 主机 / 功能模块 （USB）	• 依照 USB 版本 2.0 • 支持 480Mbps 和 12Mbps 的传送率 • 可作为功能器件使用 • 可通过软件设定 • 内置 8K 字节 RAM 作为通信缓冲器

项目	特　　点
取样频率转换器 （SRC）	• 数据格式：立体 32 位 （L/R 各 16 位）、单声道 16 位 • 输入取样频率：8/11.025/12/16/22.05/24/32/44.1/48kHz • 输出取样频率：44.1/48kHz
LCD 控制器 （LCDC）	• 支持 16×1 ～ 1,024×1,024 位 • 支持 4、 8、 15、 16bpp 颜色模式 • 支持 1、 2、 4、 6bpp 灰度模式 • 支持 TFT/DSTN/STN 显示器 • 可设定信号极性 • 24 位调色板存储器 （24 位中 16 位有效 ;R:5/G:6/B:5） • 统一图形存储器体系结构
SD 主机接口 （SDHI） 【注】根据型号分为具有 SDHI 的产品和不具有 SDHI 的产品	• SD 存储器 /IO 卡接口 （1 位 /4 位 SD 总线） • 错误校验功能：CRC7 （指令）、 CRC16 （数据） • MMC （MultiMediaCard）存取 • 中断请求：卡存取中断、 SDIO 存取中断、卡检测中断 • DMA 传送请求：SD_BUF 写、 SD_BUF 读 • 支持卡检测功能、写保护
I/O 端口	• 82 个 （输入 / 输出）、 16 个 （输入）、 1 个 （输出） • 输入 / 输出端口可按每个位切换输入 / 输出 • 内置弱保持电路
A/D 转换器 （ADC）	• 分辨率：10 位 • 输入：8 个通道 • 通过外部触发器 / 定时触发器，可启动 A/D 转换
D/A 转换器 （DAC）	• 分辨率：8 位 • 输出 2 个通道
用户断点控制器 （UBC）	• 断点通道 ×2 个通道 • 可将地址、数据值、存取格式和数据长度设定为断点条件
用户调试接口 （H-UDI）	• 支持 E10A 仿真器 • JTAG 标准引脚配置
内置 RAM	• 内置用于高速存取的 64K 字节存储器 （16KB×4 面） • 内置用于数据保持的 16K 字节存储器 （4KB×4 面）
电源电压	• V_{CC}：1.1 ～ 1.3V、 PV_{CC}：3.0 ～ 3.6V
封装	• QFP3232-240Cu （0.5 间距）

1.2 产品一览表

表 1.2 产品一览

产品分类	产品型号	IEB	SDHI	封装
SH7263	R5S72630P200FP	不具有	不具有	QFP3232-240Cu
	R5S72631P200FP	不具有	具有	
	R5S72632P200FP	具有	不具有	
	R5S72633P200FP	具有	具有	

1.3 框图

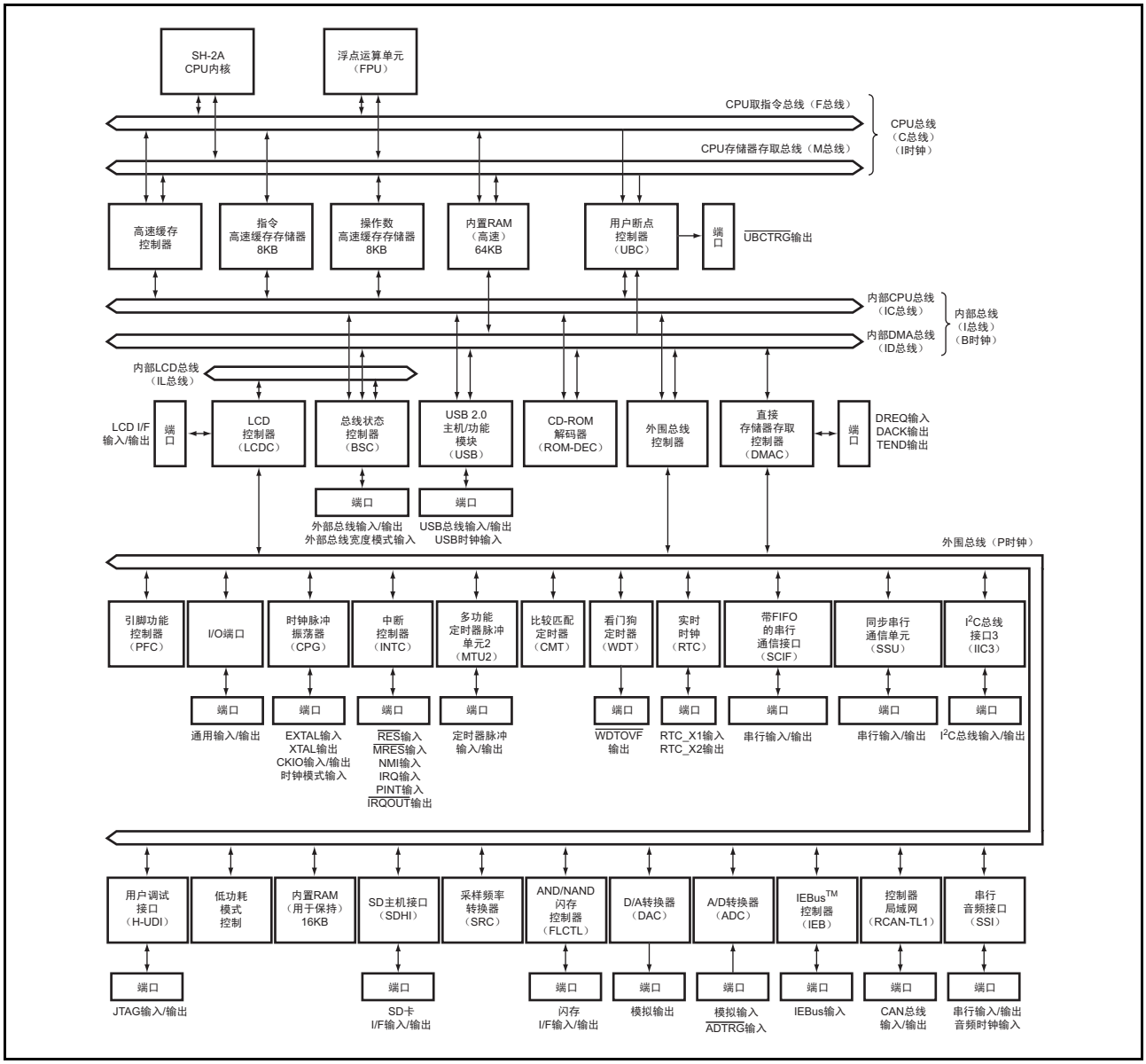


图 1.1 框图

1.4 引脚排列图

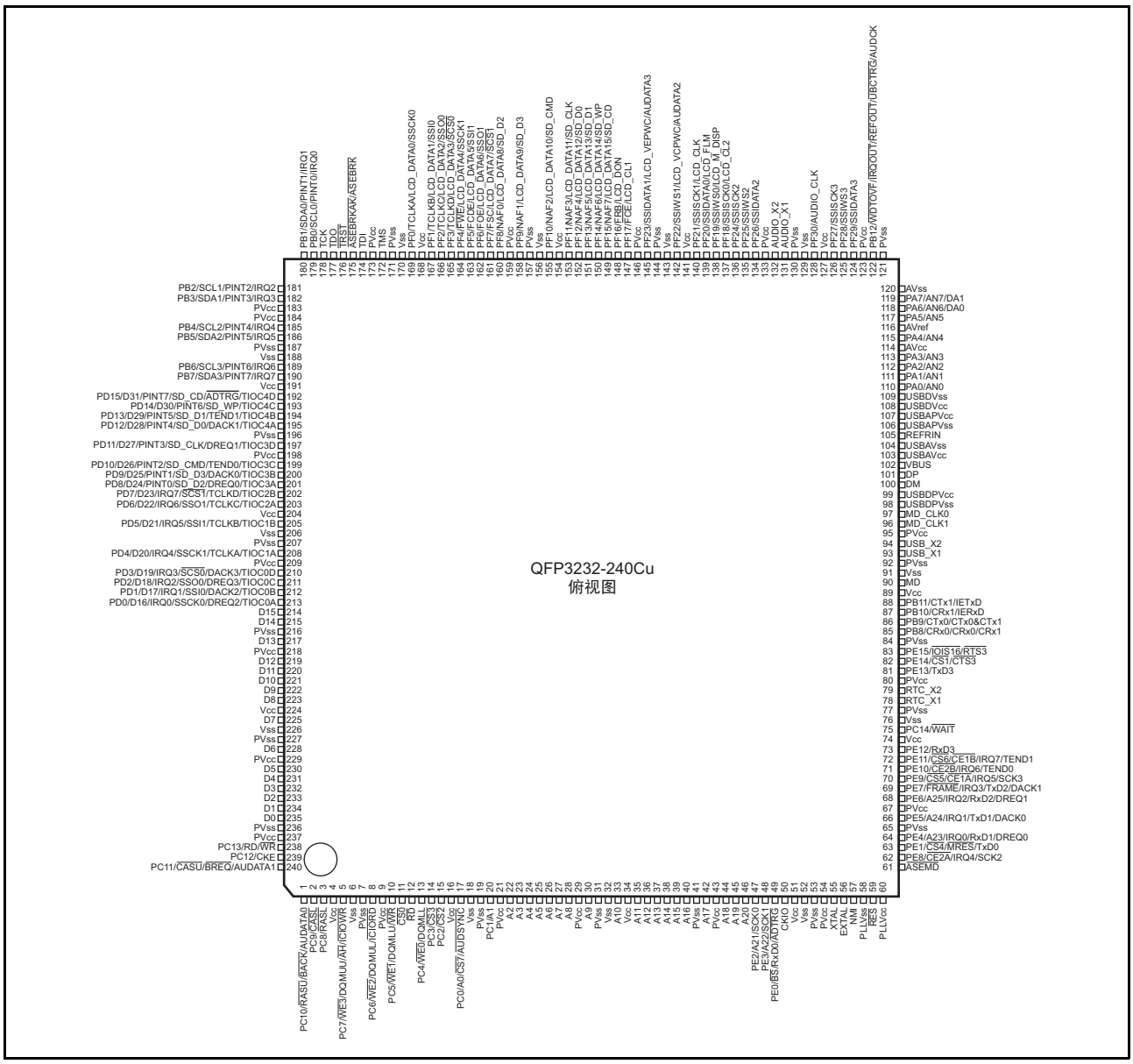


图 1.2 引脚排列图

1.5　引脚功能

表 1.3　引脚功能

分类	引脚名	输入 / 输出	名称	功　　能
电源	V_{CC}	输入	电源	是电源引脚。请将全部 V_{CC} 引脚连接到系统电源。如果有开路引脚，就不运行。
	V_{SS}	输入	接地	是接地引脚。请将全部 V_{SS} 引脚连接到系统电源（0V）。如果有开路引脚，就不运行。
	PV_{CC}	输入	用于输入 / 输出电路的电源	是用于输入 / 输出引脚的电源引脚。请将全部 PV_{CC} 引脚连接到系统电源。如果有开路引脚，就不运行。
	PV_{SS}	输入	用于输入 / 输出电路的接地	是用于输入 / 输出引脚的接地引脚。请将全部 PV_{SS} 引脚连接到系统电源 （0V）。如果有开路引脚，就不运行。
	$PLLV_{CC}$	输入	用于 PLL 的电源	是用于内置 PLL 振荡器的电源。
	$PLLV_{SS}$	输入	用于 PLL 的接地	是用于内置 PLL 振荡器的接地引脚。
时钟	EXTAL	输入	外部时钟	连接晶体振荡器。 EXTAL 引脚也可输入外部时钟。
	XTAL	输出	晶体	连接晶体振荡器。
	CKIO	输入 / 输出	系统时钟输入 / 输出	向外部时钟输入引脚或外部器件提供系统时钟。
运行模式的控制	MD	输入	模式设定	设定运行模式。请不要在运行中改变此引脚。
	MD_CLK1、MD_CLK0	输入	时钟模式设定	设定时钟运行模式。请不要在运行中改变此引脚。
	$\overline{ASEMD}$	输入	ASE 模式	如果在 $\overline{RES}$ 引脚断言期间输入低电平，就变为 ASE 模式。如果输入高电平就变为产品芯片模式。 ASE 模式中 E10A-USB 仿真器功能有效。不使用仿真器功能时，请固定为高电平。
系统控制	$\overline{RES}$	输入	上电复位	当此引脚为低电平时，处于上电复位状态。
	$\overline{MRES}$	输入	手动复位	当此引脚为低电平时，处于手动复位状态。
	$\overline{WDTOVF}$	输出	看门狗定时器的溢出	是来自 WDT 的溢出输出信号。
	$\overline{BREQ}$	输入	总线权请求	在外部器件请求释放总线权时，置为低电平。
	$\overline{BACK}$	输出	总线权请求的应答	表示已将总线权释放给外部器件。能得知输出 $\overline{BREQ}$ 信号的器件已接收 $\overline{BACK}$ 信号，并获得总线权。
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，就请固定为高电平。
	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。
	PINT7 ～ PINT0	输入	中断请求 7 ～ 0	是可屏蔽的中断请求引脚。 只能选择电平输入。
	$\overline{IRQOUT}$	输出	中断请求输出	表示已产生中断源。在总线释放过程中也可得知中断的产生。
地址总线	A25 ～ A0	输出	地址总线	输出地址。
数据总线	D31 ～ D0	输入 / 输出	数据总线	是双方向的数据总线。

分类	引脚名	输入 / 输出	名称	功 能
总线控制	$\overline{CS_7}$ ～ $\overline{CS_0}$	输出	片选 7 ～ 0	用于外部存储器或者器件的片选信号。
	$\overline{RD}$	输出	读出	表示从外部器件读出。
	RD/$\overline{WR}$	输出	读 / 写	是读 / 写信号。
	$\overline{BS}$	输出	总线开始	是总线周期开始信号。
	$\overline{AH}$	输出	地址保持	保持地址 / 数据多路复用 I/O 的地址的信号。
	$\overline{FRAME}$	输出	$\overline{FRAME}$ 信号	突发 MPX-I/O 接口时，连接 $\overline{FRAME}$ 信号。
	$\overline{WAIT}$	输入	等待	存取外围空间时，给总线周期插入等待周期。
	$\overline{WE0}$	输出	指定字节	表示写入外部存储器或器件数据的 bit7 ～ 0。
	$\overline{WE1}$	输出	指定字节	表示写入外部存储器或器件数据的 bit15 ～ 8。
	$\overline{WE2}$	输出	指定字节	表示写入外部存储器或器件数据的 bit23 ～ 16
	$\overline{WE3}$	输出	指定字节	表示写入外部存储器或器件数据的 bit31 ～ 24。
	DQMLL	输出	指定字节	连接 SDRAM 时，选择 D7 ～ D0。
	DQMLU	输出	指定字节	连接 SDRAM 时，选择 D15 ～ D8。
	DQMUL	输出	指定字节	连接 SDRAM 时，选择 D23 ～ D16。
	DQMUU	输出	指定字节	连接 SDRAM 时，选择 D31 ～ D24。
	$\overline{RASU}$、$\overline{RASL}$	输出	RAS	连接 SDRAM 时，连接 $\overline{RAS}$ 引脚。
	$\overline{CASU}$、$\overline{CASL}$	输出	CAS	连接 SDRAM 时，连接 $\overline{CAS}$ 引脚。
	CKE	输出	CK 允许	连接 SDRAM 时，连接 CKE 引脚。
	$\overline{CE1A}$、$\overline{CE1B}$	输出	选择 PCMCIA 卡的低位字节	连接 PCMCIA 卡的选择信号 D7 ～ D0。
	$\overline{CE2A}$、$\overline{CE2B}$	输出	选择 PCMCIA 卡的高位字节	连接 PCMCIA 卡的选择信号 D15 ～ D8。
	$\overline{ICIOWR}$	输出	PCMCIA 写选通	连接 PCMCIA I/O 写选通。
	$\overline{ICIORD}$	输出	PCMCIA 读选通	连接 PCMCIA I/O 读选通。
	$\overline{WE}$	输出	PCMCIA 存储器写选通	连接 PCMCIA 存储器写选通。
	$\overline{IOIS16}$	输入	PCMCIA 动态总线宽度变更	表示 PCMCIA 的 16 位 I/O。
	$\overline{REFOUT}$	输出	刷新请求	执行刷新的请求信号。
直接存储器存取控制器（DMAC）	DREQ0 ～ DREQ3	输入	DMA 传送请求	是接收外部 DMA 传送请求的输入引脚。
	DACK0 ～ DACK3	输出	接收 DMA 传送请求	是对外部 DMA 传送请求输出接收请求的引脚。
	TEND0、TEND1	输出	DMA 传送结束输出	是 DMA 传送结束的输出引脚。

分类	引脚名	输入 / 输出	名称	功 能
多功能定时器脉冲单元 2（MTU2）	TCLKA、TCLKB、TCLKC、TCLKD	输入	MTU2 定时器的时钟输入	是定时器的外部时钟输入引脚。
	TIOC0A、TIOC0B、TIOC0C、TIOC0D	输入 / 输出	MTU2 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 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。
	TIOC4A、TIOC4B、TIOC4C、TIOC4D	输入 / 输出	MTU2 的输入捕捉 / 输出比较 （通道 4）	是 TGRA_4、 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。
实时时钟（RTC）	RTC_X1	输入	用于 RTC 的晶体振荡器	连接 32.768kHz 的晶体振荡器。
	RTC_X2	输出		
带 FIFO 的串行通信接口（SCIF）	T_xD3 ～ T_xD0	输出	发送数据	是数据输出引脚。
	R_xD3 ～ R_xD0	输入	接收数据	是数据输入引脚。
	SCK3 ～ SCK0	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。
	$\overline{CTS3}$	输入	发送允许	是调制解调器的控制引脚。
同步串行通信单元 （SSU）	SSO1、 SSO0	输入 / 输出	数据	是数据的输入 / 输出引脚。
	SSI1、 SSI0	输入 / 输出	数据	是数据的输入 / 输出引脚。
	SSCK1、SSCK0	输入 / 输出	时钟	是时钟的输入 / 输出引脚。
	$\overline{SCS1}$、 $\overline{SCS0}$	输入 / 输出	片选	是片选的输入 / 输出引脚。
I^2C 总线接口 3（IIC3）	SCL3 ～ SCL0	输入 / 输出	串行时钟引脚	是串行时钟的输入 / 输出引脚。
	SDA3 ～ SDA0	输入 / 输出	串行数据引脚	是串行数据的输入 / 输出引脚。

分类	引脚名	输入 / 输出	名称	功 能
串行音频接口（SSI）	SSIDATA3 ～ SSIDATA0	输入 / 输出	SSI 数据输入 / 输出	是串行数据的输入 / 输出引脚。
	SSISCK3 ～ SSISCK0	输入 / 输出	SSI 时钟输入 / 输出	是串行时钟的输入 / 输出引脚。
	SSIWS3 ～ SSIWS0	输入 / 输出	SSI 时钟 LR 输入 / 输出	是字选择的输入 / 输出引脚。
	AUDIO_CLK	输入	用于 SSI 音频的外部时钟	输入用于音频的外部时钟。 选择 AUDIO_X1、 AUDIO_X2 的振荡时钟，输入分频器。
	AUDIO_X1	输入	用于 SSI 音频的晶体振荡器 / 外部时钟	连接用于音频的晶体振荡器。 另外， AUDIO_X1 引脚可输入外部时钟。 选择 AUDIO_CLK 输入时钟，输入分频器。
	AUDIO_X2	输出		
控制器局域网（RCAN-TL1）	CT_x0、 CT_x1	输出	CAN 总线发送数据	是用于 CAN 总线发送的引脚。
	CR_x0、 CR_x1	输入	CAN 总线接收数据	是用于 CAN 总线接收的引脚。
IEBus™ 控制器 （IEB）	IET_xD	输出	IEB 发送数据	是 IEB 的发送数据输出引脚。
	IER_xD	输入	IEB 接收数据	是 IEB 的接收数据输入引脚。
AND/NAND 闪存控制器（FLCTL）	FOE	输出	闪存输出允许	地址锁定允许：地址输出时断言。数据输入 / 输出时取反。 输出允许：数据输入时 / 状态为读取时断言。
	FSC	输出	闪存串行时钟	读允许：在下降沿读数据。 串行时钟：使信号同步，输入 / 输出数据。
	$\overline{FCE}$	输出	闪存芯片选通	芯片选通：允许本 LSI 连接的闪存。
	FCDE	输出	闪存命令数据允许	命令锁定允许：命令输出时断言。 命令数据允许：命令输出时断言。
	FRB	输入	闪存就绪 / 忙	就绪 / 忙：高电平时表示为就绪状态，低电平时为忙状态。
	$\overline{FWE}$	输出	闪存写允许	写允许：在下降沿，闪存锁定命令、地址和数据。
	NAF7 ～ NAF0	输入 / 输出	闪存数据	是数据输入 / 输出引脚。

分类	引脚名	输入 / 输出	名称	功 能
USB2.0 主机 / 功能模块（USB）	DP	输入 / 输出	USB D+ 数据	是 USB 总线的 D+ 数据。
	DM	输入 / 输出	USB D- 数据	是 USB 总线的 D- 数据。
	VBUS	输入	VBUS 输入	请连接 USB 总线的 Vbus。
	REFRIN	输入	基准输入	请通过 T.B.D. kΩ 电阻连接 $USBAPV_{SS}$。
	USB_X1	输入	用于 USB 的晶体振荡器 / 外部时钟	连接用于 USB 的晶体振荡器。 USB_X1 引脚可输入外部时钟。
	USB_X2	输出		
	$USBAPV_{CC}$	输入	收发器模拟引脚电源	是用于引脚的电源。
	$USBAPV_{SS}$	输入	收发器模拟引脚接地	是用于引脚的接地。
	$USBDPV_{CC}$	输入	收发器数字引脚电源	是用于引脚的电源。
	$USBDPV_{SS}$	输入	收发器数字引脚接地	是用于引脚的接地。
	$USBAV_{CC}$	输入	收发器模拟内核电源	是用于内核的电源。
	$USBAV_{SS}$	输入	收发器模拟内核接地	是用于内核的接地。
	$USBDV_{CC}$	输入	收发器数字内核电源	是用于内核的电源。
	$USBDV_{SS}$	输入	收发器数字内核接地	是用于内核的接地。
LCD 控制器（LCDC）	LCD_DATA15 ～ LCD_DATA0	输出	LCD 数据	是用于 LCD 面板的数据输入 / 输出引脚。
	LCD_CL1	输出	移位时钟	是 LCD 移位时钟 1/ 水平同步信号引脚。
	LCD_CL2	输出	移位时钟	是 LCD 移位时钟 2/ 点时钟引脚。
	LCD_CLK	输入	时钟源	是 LCD 时钟源输入 / 输出引脚。
	LCD_FLM	输出	线路标识	是起始线路标识 / 垂直同步信号引脚。
	LCD_DON	输出	LCD 显示开始	是 LCD 显示开始信号引脚。
	LCD_VCPWC	输出	电源控制	是 LCD 模块电源控制 （VCC）引脚。
	LCD_VEPWC	输出	电源控制 （VEE）	是 LCD 模块电源控制 （VEE）引脚。
	LCD_M_DISP	输出	LCD 电流交替	是 LCD 电流交替信号引脚。
SD 主机接口（SDHI）	SD_CLK	输出	SD 时钟	是 SD 时钟输出引脚。
	SD_CMD	输入 / 输出	SD 命令	是 SD 命令输出、响应输入信号。
	SD_D3 ～ SD_D0	输入 / 输出	SD 数据	是 SD 数据总线信号。
	SD_CD	输入	SD 卡的检测	是 SD 卡的检测。
	SD_WP	输入	SD 写保护	是 SD 写保护信号。
A/D 转换器（ADC）	AN7 ～ AN0	输入	模拟输入引脚	是模拟输入引脚。
	$\overline{ADTRG}$	输入	A/D 转换触发输入	开始 A/D 转换的外部触发输入引脚。
D/A 转换器（DAC）	DA1、 DA0	输出	模拟输出引脚	是模拟输出引脚。
模拟通用	AV_{CC}	输入	模拟电源	是 A/D 转换器和 D/A 转换器的电源引脚。
	AV_{SS}	输入	模拟接地	是 A/D 转换器和 D/A 转换器的接地引脚。
	AV_{ref}	输入	模拟基准电压	是 A/D 转换器和 D/A 转换器的基准电压引脚。

分类	引脚名	输入 / 输出	名称	功 能
I/O 端口	PB11 ～ PB8、PC14 ～ PC0、PD15 ～ PD0、PE15 ～ PE0、PF30 ～ PF0	输入 / 输出	通用端口	是 82 位的通用输入 / 输出端口引脚。
	PA7 ～ PA0、PB7 ～ PB0	输入	通用端口	是 16 位的通用输入端口引脚。
	PB12	输出	通用端口	是 1 位的通用输出端口引脚。
用户调试接口（H-UDI）	TCK	输入	测试时钟	是测试时钟输入引脚。
	TMS	输入	测试模式选择	是测试模式选择信号输入引脚。
	TDI	输入	测试数据输入	是指令和数据的串行输入引脚。
	TDO	输出	测试数据输出	是指令和数据的串行输出引脚。
	$\overline{\text{TRST}}$	输入	测试复位	是初始化信号输入引脚。
仿真器接口	AUDATA3 ～ AUDATA0	输出	AUD 数据	是转移地址 / 转移源地址输出引脚。
	AUDCK	输出	AUD 时钟	是同步时钟输出引脚。
	$\overline{\text{AUDSYNC}}$	输出	AUD 同步信号	是数据起始位置识别信号输出引脚。
	$\overline{\text{ASEBRKAK}}$	输出	断点模式应答	表示 E10A-USB 仿真器进入断点模式。
	$\overline{\text{ASEBRK}}$	输入	断点请求	是 E10A-USB 仿真器断点输入。
用户断点控制器（UBC）	$\overline{\text{UBCTRG}}$	输出	用户断点触发输入	是 UBC 条件相同的触发输出。

1.6　引脚一览表

表 1.4　引脚一览表

引脚编号	功能 1		功能 2		功能 3	
	引脚名	I/O	引脚名	I/O	引脚名	I/O
1	PC10	I/O	$\overline{\text{RASU}}$	O	$\overline{\text{BACK}}$	O
2	PC9	I/O	$\overline{\text{CASL}}$	O	—	—
3	PC8	I/O	$\overline{\text{RASL}}$	O	—	—
4	V_{CC}					
5	PC7	I/O	$\overline{\text{WE3}}$/DQMUU/ $\overline{\text{AH}}$/$\overline{\text{ICIOWR}}$	O	—	—
6	V_{SS}					
7	PV_{SS}					
8	PC6	I/O	$\overline{\text{WE2}}$/DQMUL/$\overline{\text{ICIORD}}$	O	—	—
9	PV_{CC}					
10	PC5	I/O	$\overline{\text{WE1}}$/DQMLU/$\overline{\text{WR}}$	O	—	—
11	$\overline{\text{CS0}}$	O	—	—	—	—
12	$\overline{\text{RD}}$	O	—	—	—	—
13	PC4	I/O	$\overline{\text{WE0}}$/DQMLL	O	—	—
14	PC3	I/O	$\overline{\text{CS3}}$	O	—	—
15	PC2	I/O	$\overline{\text{CS2}}$	O	—	—

引脚编号	功能 4		功能 5		功能 6		弱保持	上拉
	引脚名	I/O	引脚名	I/O	引脚名	I/O		
1	AUDATA0	O	—	—	—	—	○	
2	—	—	—	—	—	—	○	
3	—	—	—	—	—	—	○	
4								
5	—	—	—	—	—	—	○	
6								
7								
8	—	—	—	—	—	—	○	
9								
10	—	—	—	—	—	—	○	
11	—	—	—	—	—	—	○	
12	—	—	—	—	—	—	○	
13	—	—	—	—	—	—	○	
14	—	—	—	—	—	—	○	
15	—	—	—	—	—	—	○	

引脚编号	功能 1		功能 2		功能 3	
	引脚名	I/O	引脚名	I/O	引脚名	I/O
16	V_{CC}					
17	PC0	I/O	A0	O	$\overline{CS7}$	O
18	V_{SS}					
19	PV_{SS}					
20	PC1	I/O	A1	O	—	—
21	PV_{CC}					
22	A2	O	—	—	—	—
23	A3	O	—	—	—	—
24	A4	O	—	—	—	—
25	A5	O	—	—	—	—
26	A6	O	—	—	—	—
27	A7	O	—	—	—	—
28	A8	O	—	—	—	—
29	PV_{CC}					
30	A9	O	—	—	—	—
31	PV_{SS}					
32	V_{SS}					

引脚编号	功能 4		功能 5		功能 6		弱保持	上拉
	引脚名	I/O	引脚名	I/O	引脚名	I/O		
16								
17	$\overline{AUDSYNC}$	O	—	—	—	—	○	
18								
19								
20	—	—	—	—	—	—	○	
21								
22	—	—	—	—	—	—	○	
23	—	—	—	—	—	—	○	
24	—	—	—	—	—	—	○	
25	—	—	—	—	—	—	○	
26	—	—	—	—	—	—	○	
27	—	—	—	—	—	—	○	
28	—	—	—	—	—	—	○	
29								
30	—	—	—	—	—	—	○	
31								
32								

引脚编号	功能 1		功能 2		功能 3	
	引脚名	I/O	引脚名	I/O	引脚名	I/O
33	A10	O	—	—	—	—
34	V_{CC}					
35	A11	O	—	—	—	—
36	A12	O	—	—	—	—
37	A13	O	—	—	—	—
38	A14	O	—	—	—	—
39	A15	O	—	—	—	—
40	A16	O	—	—	—	—
41	PV_{SS}					
42	A17	O	—	—	—	—
43	PV_{CC}					
44	A18	O	—	—	—	—
45	A19	O	—	—	—	—
46	A20	O	—	—	—	—
47	PE2	$I_{(s)}$/O	A21	O	—	—
48	PE3	$I_{(s)}$/O	A22	O	—	—
49	PE0	$I_{(s)}$/O	$\overline{BS}$	O	—	—

引脚编号	功能 4		功能 5		功能 6		弱保持	上拉
	引脚名	I/O	引脚名	I/O	引脚名	I/O		
33	—	—	—	—	—	—	○	
34								
35	—	—	—	—	—	—	○	
36	—	—	—	—	—	—	○	
37	—	—	—	—	—	—	○	
38	—	—	—	—	—	—	○	
39	—	—	—	—	—	—	○	
40	—	—	—	—	—	—	○	
41								
42	—	—	—	—	—	—	○	
43								
44	—	—	—	—	—	—	○	
45	—	—	—	—	—	—	○	
46	—	—	—	—	—	—	○	
47	SCK0	$I_{(s)}$/O	—	—	—	—	○	
48	SCK1	$I_{(s)}$/O	—	—	—	—	○	
49	R_xD0	$I_{(s)}$	$\overline{ADTRG}$	$I_{(s)}$	—	—	○	

引脚编号	功能 1		功能 2		功能 3	
	引脚名	I/O	引脚名	I/O	引脚名	I/O
50	CKIO	I/O	—	—	—	—
51	V_{CC}					
52	V_{SS}					
53	PV_{SS}					
54	PV_{CC}					
55	XTAL	O	—	—	—	—
56	EXTAL	$I_{(s)}$	—	—	—	—
57	NMI	$I_{(s)}$	—	—	—	—
58	$PLLV_{SS}$					
59	$\overline{RES}$	$I_{(s)}$	—	—	—	—
60	$PLLV_{CC}$					
61	$\overline{ASEMD}$	$I_{(s)}$	—	—	—	—
62	PE8	$I_{(s)}$/O	$\overline{CE2A}$	O	IRQ4	$I_{(s)}$
63	PE1	$I_{(s)}$/O	$\overline{CS4}$	O	$\overline{MRES}$	$I_{(s)}$
64	PE4	$I_{(s)}$/O	A23	O	IRQ0	$I_{(s)}$
65	PV_{SS}					
66	PE5	$I_{(s)}$/O	A24	O	IRQ1	$I_{(s)}$

引脚编号	功能 4		功能 5		功能 6		弱保持	上拉
	引脚名	I/O	引脚名	I/O	引脚名	I/O		
50	—	—	—	—	—	—		
51								
52								
53								
54								
55	—	—	—	—	—	—		
56	—	—	—	—	—	—		
57	—	—	—	—	—	—		
58								
59	—	—	—	—	—	—		
60								
61	—	—	—	—	—	—		
62	SCK2	$I_{(s)}$/O	—	—	—	—	○	
63	T_xD0	O	—	—	—	—	○	
64	R_xD1	$I_{(s)}$	DREQ0	$I_{(s)}$	—	—	○	
65								
66	T_xD1	O	DACK0	O	—	—	○	

引脚编号	功能 1		功能 2		功能 3	
	引脚名	I/O	引脚名	I/O	引脚名	I/O
67	PV_{CC}					
68	PE6	$I_{(s)}/O$	A25	O	IRQ2	$I_{(s)}$
69	PE7	$I_{(s)}/O$	$\overline{FRAME}$	O	IRQ3	$I_{(s)}$
70	PE9	$I_{(s)}/O$	$\overline{CS5}/\overline{CE1A}$	O	IRQ5	$I_{(s)}$
71	PE10	$I_{(s)}/O$	$\overline{CE2B}$	O	IRQ6	$I_{(s)}$
72	PE11	$I_{(s)}/O$	$\overline{CS6}/\overline{CE1B}$	O	IRQ7	$I_{(s)}$
73	PE12	$I_{(s)}/O$	—	—	—	—
74	V_{CC}					
75	PC14	I/O	$\overline{WAIT}$	I	—	—
76	V_{SS}					
77	PV_{SS}					
78	RTC_X1	I	—	—	—	—
79	RTC_X2	O	—	—	—	—
80	PV_{CC}					
81	PE13	$I_{(s)}/O$	—	—	—	—
82	PE14	$I_{(s)}/O$	$\overline{CS1}$	O	—	—
83	PE15	$I_{(s)}/O$	$\overline{IOIS16}$	$I_{(s)}$	—	—

引脚编号	功能 4		功能 5		功能 6		弱保持	上拉
	引脚名	I/O	引脚名	I/O	引脚名	I/O		
67								
68	R_xD2	$I_{(s)}$	DREQ1	$I_{(s)}$	—	—	○	
69	T_xD2	O	DACK1	O	—	—	○	
70	SCK3	$I_{(s)}/O$	—	—	—	—	○	
71	—	—	TEND0	O	—	—	○	
72	—	—	TEND1	O	—	—	○	
73	R_xD3	$I_{(s)}$	—	—	—	—	○	
74								
75	—	—	—	—	—	—	○	
76								
77								
78	—	—	—	—	—	—		
79	—	—	—	—	—	—		
80								
81	T_xD3	O	—	—	—	—	○	
82	$\overline{CTS3}$	$I_{(s)}/O$	—	—	—	—	○	
83	$\overline{RTS3}$	$I_{(s)}/O$	—	—	—	—	○	

引脚编号	功能 1		功能 2		功能 3	
	引脚名	I/O	引脚名	I/O	引脚名	I/O
84	PV_{SS}					
85	PB8	I/O	CR_x0	I	CR_x0/CR_x1	I
86	PB9	I/O	CT_x0	O	$CT_x0\&CT_x1$	O
87	PB10	I/O	CR_x1	I	IER_xD	I
88	PB11	I/O	CT_x1	O	IET_xD	O
89	V_{CC}					
90	MD	$I_{(s)}$	—	—	—	—
91	V_{SS}					
92	PV_{SS}					
93	USB_X1	I	—	—	—	—
94	USB_X2	O	—	—	—	—
95	PV_{CC}					
96	MD_CLK1	$I_{(s)}$	—	—	—	—
97	MD_CLK0	$I_{(s)}$	—	—	—	—
98	$USBDPV_{SS}$					
99	$USBDPV_{CC}$					
100	DM	I/O	—	—	—	—

引脚编号	功能 4		功能 5		功能 6		弱保持	上拉
	引脚名	I/O	引脚名	I/O	引脚名	I/O		
84								
85	—	—	—	—	—	—	○	
86	—	—	—	—	—	—	○	
87	—	—	—	—	—	—	○	
88	—	—	—	—	—	—	○	
89								
90	—	—	—	—	—	—		
91								
92								
93	—	—	—	—	—	—		
94	—	—	—	—	—	—		
95								
96	—	—	—	—	—	—		
97	—	—	—	—	—	—		
98								
99								
100	—	—	—	—	—	—		

引脚编号	功能 1		功能 2		功能 3	
	引脚名	I/O	引脚名	I/O	引脚名	I/O
101	DP	I/O	—	—	—	—
102	VBUS	I	—	—	—	—
103	$USBAV_{CC}$					
104	$USBAV_{SS}$					
105	REFRIN	I	—	—	—	—
106	$USBAPV_{SS}$					
107	$USBAPV_{CC}$					
108	$USBDV_{CC}$					
109	$USBDV_{SS}$					
110	PA0	I	AN0	I(a)	—	—
111	PA1	I	AN1	I(a)	—	—
112	PA2	I	AN2	I(a)	—	—
113	PA3	I	AN3	I(a)	—	—
114	AV_{CC}					
115	PA4	I	AN4	I(a)	—	—
116	AV_{ref}					
117	PA5	I	AN5	I(a)	—	—

引脚编号	功能 4		功能 5		功能 6		弱保持	上拉
	引脚名	I/O	引脚名	I/O	引脚名	I/O		
101	—	—	—	—	—	—		
102	—	—	—	—	—	—		
103								
104								
105	—	—	—	—	—	—		
106								
107								
108								
109								
110	—	—	—	—	—	—		
111	—	—	—	—	—	—		
112	—	—	—	—	—	—		
113	—	—	—	—	—	—		
114								
115	—	—	—	—	—	—		
116								
117	—	—	—	—	—	—		

引脚编号	功能 1		功能 2		功能 3	
	引脚名	I/O	引脚名	I/O	引脚名	I/O
118	PA6	I	AN6	I(a)	DA0	O(a)
119	PA7	I	AN7	I(a)	DA1	O(a)
120	AV_{SS}					
121	PV_{SS}					
122	PB12	O	$\overline{WDTOVF}$	O	$\overline{IRQOUT}$/ $\overline{REFOUT}$	O
123	PV_{CC}					
124	PF29	I/O	SSIDATA3	I/O	—	—
125	PF28	I/O	SSIWS3	I/O	—	—
126	PF27	I/O	SSISCK3	I/O	—	—
127	V_{CC}					
128	PF30	I/O	AUDIO_CLK	I	—	—
129	V_{SS}					
130	PV_{SS}					
131	AUDIO_X1	I	—	—	—	—
132	AUDIO_X2	O	—	—	—	—
133	PV_{CC}					
134	PF26	I/O	SSIDATA2	I/O	—	—

引脚编号	功能 4		功能 5		功能 6		弱保持	上拉
	引脚名	I/O	引脚名	I/O	引脚名	I/O		
118	—	—	—	—	—	—		
119	—	—	—	—	—	—		
120								
121								
122	$\overline{UBCTRG}$	O	AUDCK	O	—	—	○	
123								
124	—	—	—	—	—	—	○	
125	—	—	—	—	—	—	○	
126	—	—	—	—	—	—	○	
127								
128	—	—	—	—	—	—	○	
129								
130								
131	—	—	—	—	—	—		
132	—	—	—	—	—	—		
133								
134	—	—	—	—	—	—	○	

引脚编号	功能 1		功能 2		功能 3	
	引脚名	I/O	引脚名	I/O	引脚名	I/O
135	PF25	I/O	SSIWS2	I/O	—	—
136	PF24	I/O	SSISCK2	I/O	—	—
137	PF18	I/O	SSISCK0	I/O	LCD_CL2	O
138	PF19	I/O	SSIWS0	I/O	LCD_M_DISP	O
139	PF20	I/O	SSIDATA0	I/O	LCD_FLM	O
140	PF21	I/O	SSISCK1	I/O	LCD_CLK	I
141	V_{CC}					
142	PF22	I/O	SSIWS1	I/O	LCD_VCPWC	O
143	V_{SS}					
144	PV_{SS}					
145	PF23	I/O	SSIDATA1	I/O	LCD_VEPWC	O
146	PV_{CC}					
147	PF17	I/O	$\overline{FCE}$	O	LCD_CL1	O
148	PF16	I/O	FRB	I	LCD_DON	O
149	PF15	I/O	NAF7	I/O	LCD_DATA15	O
150	PF14	I/O	NAF6	I/O	LCD_DATA14	O
151	PF13	I/O	NAF5	I/O	LCD_DATA13	O

引脚编号	功能 4		功能 5		功能 6		弱保持	上拉
	引脚名	I/O	引脚名	I/O	引脚名	I/O		
135	—	—	—	—	—	—	○	
136	—	—	—	—	—	—	○	
137	—	—	—	—	—	—	○	
138	—	—	—	—	—	—	○	
139	—	—	—	—	—	—	○	
140	—	—	—	—	—	—	○	
141								
142	AUDATA2	O	—	—	—	—	○	
143								
144								
145	AUDATA3	O	—	—	—	—	○	
146								
147	—	—	—	—	—	—	○	
148	—	—	—	—	—	—	○	
149	SD_CD	I	—	—	—	—	○	
150	SD_WP	I	—	—	—	—	○	
151	SD_D1	I/O	—	—	—	—	○	

引脚编号	功能 1		功能 2		功能 3	
	引脚名	I/O	引脚名	I/O	引脚名	I/O
152	PF12	I/O	NAF4	I/O	LCD_DATA12	O
153	PF11	I/O	NAF3	I/O	LCD_DATA11	O
154	V_{CC}					
155	PF10	I/O	NAF2	I/O	LCD_DATA10	O
156	V_{SS}					
157	PV_{SS}					
158	PF9	I/O	NAF1	I/O	LCD_DATA9	O
159	PV_{CC}					
160	PF8	I/O	NAF0	I/O	LCD_DATA8	O
161	PF7	$I_{(s)}$/O	FSC	O	LCD_DATA7	O
162	PF6	$I_{(s)}$/O	FOE	O	LCD_DATA6	O
163	PF5	$I_{(s)}$/O	FCDE	O	LCD_DATA5	O
164	PF4	$I_{(s)}$/O	$\overline{FWE}$	O	LCD_DATA4	O
165	PF3	$I_{(s)}$/O	TCLKD	$I_{(s)}$	LCD_DATA3	O
166	PF2	$I_{(s)}$/O	TCLKC	$I_{(s)}$	LCD_DATA2	O
167	PF1	$I_{(s)}$/O	TCLKB	$I_{(s)}$	LCD_DATA1	O
168	V_{CC}					

引脚编号	功能 4		功能 5		功能 6		弱保持	上拉
	引脚名	I/O	引脚名	I/O	引脚名	I/O		
152	SD_D0	I/O	—	—	—	—	○	
153	SD_CLK	O	—	—	—	—	○	
154								
155	SD_CMD	I/O	—	—	—	—	○	
156								
157								
158	SD_D3	I/O	—	—	—	—	○	
159								
160	SD_D2	I/O	—	—	—	—	○	
161	$\overline{SCS1}$	$I_{(s)}$/O	—	—	—	—	○	
162	SSO1	$I_{(s)}$/O	—	—	—	—	○	
163	SSI1	$I_{(s)}$/O	—	—	—	—	○	
164	SSCK1	$I_{(s)}$/O	—	—	—	—	○	
165	$\overline{SCS0}$	$I_{(s)}$/O	—	—	—	—	○	
166	SSO0	$I_{(s)}$/O	—	—	—	—	○	
167	SSI0	$I_{(s)}$/O	—	—	—	—	○	
168								

引脚编号	功能 1		功能 2		功能 3	
	引脚名	I/O	引脚名	I/O	引脚名	I/O
169	PF0	$I_{(s)}/O$	TCLKA	$I_{(s)}$	LCD_DATA0	O
170	V_{SS}					
171	PV_{SS}					
172	TMS	I	—	—	—	—
173	PV_{CC}					
174	TDI	I	—	—	—	—
175	$\overline{\text{ASEBRKAK}}/\overline{\text{ASEBRK}}$	$I_{(s)}/O$	—	—	—	—
176	$\overline{\text{TRST}}$	$I_{(s)}$	—	—	—	—
177	TDO	O	—	—	—	—
178	TCK	I	—	—	—	—
179	PB0	$I_{(s)}$	SCL0	$I_{(s)}/O_{(o)}$	PINT0	$I_{(s)}$
180	PB1	$I_{(s)}$	SDA0	$I_{(s)}/O_{(o)}$	PINT1	$I_{(s)}$
181	PB2	$I_{(s)}$	SCL1	$I_{(s)}/O_{(o)}$	PINT2	$I_{(s)}$
182	PB3	$I_{(s)}$	SDA1	$I_{(s)}/O_{(o)}$	PINT3	$I_{(s)}$
183	PV_{CC}					
184	PV_{CC}					
185	PB4	$I_{(s)}$	SCL2	$I_{(s)}/O_{(o)}$	PINT4	$I_{(s)}$

引脚编号	功能 4		功能 5		功能 6		弱保持	上拉
	引脚名	I/O	引脚名	I/O	引脚名	I/O		
169	SSCK0	$I_{(s)}/O$	—	—	—	—	○	
170								
171								
172	—	—	—	—	—	—		○
173								
174	—	—	—	—	—	—		○
175	—	—	—	—	—	—	○	
176	—	—	—	—	—	—		○
177	—	—	—	—	—	—		
178	—	—	—	—	—	—		○
179	IRQ0	$I_{(s)}$	—	—	—	—		
180	IRQ1	$I_{(s)}$	—	—	—	—		
181	IRQ2	$I_{(s)}$	—	—	—	—		
182	IRQ3	$I_{(s)}$	—	—	—	—		
183								
184								
185	IRQ4	$I_{(s)}$	—	—	—	—		

引脚编号	功能 1		功能 2		功能 3	
	引脚名	I/O	引脚名	I/O	引脚名	I/O
186	PB5	$I_{(s)}$	SDA2	$I_{(s)}/O_{(o)}$	PINT5	$I_{(s)}$
187	PV_{SS}					
188	V_{SS}					
189	PB6	$I_{(s)}$	SCL3	$I_{(s)}/O_{(o)}$	PINT6	$I_{(s)}$
190	PB7	$I_{(s)}$	SDA3	$I_{(s)}/O_{(o)}$	PINT7	$I_{(s)}$
191	V_{CC}					
192	PD15	I/O	D31	I/O	PINT7	$I_{(s)}$
193	PD14	I/O	D30	I/O	PINT6	$I_{(s)}$
194	PD13	I/O	D29	I/O	PINT5	$I_{(s)}$
195	PD12	I/O	D28	I/O	PINT4	$I_{(s)}$
196	PV_{SS}					
197	PD11	I/O	D27	I/O	PINT3	$I_{(s)}$
198	PV_{CC}					
199	PD10	I/O	D26	I/O	PINT2	$I_{(s)}$
200	PD9	I/O	D25	I/O	PINT1	$I_{(s)}$
201	PD8	I/O	D24	I/O	PINT0	$I_{(s)}$
202	PD7	I/O	D23	I/O	IRQ7	$I_{(s)}$

引脚编号	功能 4		功能 5		功能 6		弱保持	上拉
	引脚名	I/O	引脚名	I/O	引脚名	I/O		
186	IRQ5	$I_{(s)}$	—	—	—	—		
187								
188								
189	IRQ6	$I_{(s)}$	—	—	—	—		
190	IRQ7	$I_{(s)}$	—	—	—	—		
191								
192	SD_CD	I	$\overline{\text{ADTRG}}$	$I_{(s)}$	TIOC4D	$I_{(s)}/O$	○	
193	SD_WP	I	—	—	TIOC4C	$I_{(s)}/O$	○	
194	SD_D1	I/O	TEND1	O	TIOC4B	$I_{(s)}/O$	○	
195	SD_D0	I/O	DACK1	O	TIOC4A	$I_{(s)}/O$	○	
196								
197	SD_CLK	O	DREQ1	$I_{(s)}$	TIOC3D	$I_{(s)}/O$	○	
198								
199	SD_CMD	I/O	TEND0	O	TIOC3C	$I_{(s)}/O$	○	
200	SD_D3	I/O	DACK0	O	TIOC3B	$I_{(s)}/O$	○	
201	SD_D2	I/O	DREQ0	$I_{(s)}$	TIOC3A	$I_{(s)}/O$	○	
202	$\overline{\text{SCS1}}$	$I_{(s)}/O$	TCLKD	$I_{(s)}$	TIOC2B	$I_{(s)}/O$	○	

引脚编号	功能 1		功能 2		功能 3	
	引脚名	I/O	引脚名	I/O	引脚名	I/O
203	PD6	I/O	D22	I/O	IRQ6	$I_{(s)}$
204	V_{CC}					
205	PD5	I/O	D21	I/O	IRQ5	$I_{(s)}$
206	V_{SS}					
207	PV_{SS}					
208	PD4	I/O	D20	I/O	IRQ4	$I_{(s)}$
209	PV_{CC}					
210	PD3	I/O	D19	I/O	IRQ3	$I_{(s)}$
211	PD2	I/O	D18	I/O	IRQ2	$I_{(s)}$
212	PD1	I/O	D17	I/O	IRQ1	$I_{(s)}$
213	PD0	I/O	D16	I/O	IRQ0	$I_{(s)}$
214	D15	I/O	—	—	—	—
215	D14	I/O	—	—	—	—
216	PV_{SS}					
217	D13	I/O	—	—	—	—
218	PV_{CC}					
219	D12	I/O	—	—	—	—

引脚编号	功能 4		功能 5		功能 6		弱保持	上拉
	引脚名	I/O	引脚名	I/O	引脚名	I/O		
203	SSO1	$I_{(s)}$/O	TCLKC	$I_{(s)}$	TIOC2A	$I_{(s)}$/O	○	
204								
205	SSI1	$I_{(s)}$/O	TCLKB	$I_{(s)}$	TIOC1B	$I_{(s)}$/O	○	
206								
207								
208	SSCK1	$I_{(s)}$/O	TCLKA	$I_{(s)}$	TIOC1A	$I_{(s)}$/O	○	
209								
210	$\overline{SCS0}$	$I_{(s)}$/O	DACK3	O	TIOC0D	$I_{(s)}$/O	○	
211	SSO0	$I_{(s)}$/O	DREQ3	$I_{(s)}$	TIOC0C	$I_{(s)}$/O	○	
212	SSI0	$I_{(s)}$/O	DACK2	O	TIOC0B	$I_{(s)}$/O	○	
213	SSCK0	$I_{(s)}$/O	DREQ2	$I_{(s)}$	TIOC0A	$I_{(s)}$/O	○	
214	—	—	—	—	—	—	○	
215	—	—	—	—	—	—	○	
216								
217	—	—	—	—	—	—	○	
218								
219	—	—	—	—	—	—	○	

引脚编号	功能 1		功能 2		功能 3	
	引脚名	I/O	引脚名	I/O	引脚名	I/O
220	D11	I/O	—	—	—	—
221	D10	I/O	—	—	—	—
222	D9	I/O	—	—	—	—
223	D8	I/O	—	—	—	—
224	V_{CC}					
225	D7	I/O	—	—	—	—
226	V_{SS}					
227	PV_{SS}					
228	D6	I/O	—	—	—	—
229	PV_{CC}					
230	D5	I/O	—	—	—	—
231	D4	I/O	—	—	—	—
232	D3	I/O	—	—	—	—
233	D2	I/O	—	—	—	—
234	D1	I/O	—	—	—	—
235	D0	I/O	—	—	—	—
236	PV_{SS}					

引脚编号	功能 4		功能 5		功能 6		弱保持	上拉
	引脚名	I/O	引脚名	I/O	引脚名	I/O		
220	—	—	—	—	—	—	○	
221	—	—	—	—	—	—	○	
222	—	—	—	—	—	—	○	
223	—	—	—	—	—	—	○	
224								
225	—	—	—	—	—	—	○	
226								
227								
228	—	—	—	—	—	—	○	
229								
230	—	—	—	—	—	—	○	
231	—	—	—	—	—	—	○	
232	—	—	—	—	—	—	○	
233	—	—	—	—	—	—	○	
234	—	—	—	—	—	—	○	
235	—	—	—	—	—	—	○	
236								

引脚编号	功能 1		功能 2		功能 3	
	引脚名	I/O	引脚名	I/O	引脚名	I/O
237	PV_{CC}					
238	PC13	I/O	RD/$\overline{WR}$	O	—	—
239	PC12	I/O	CKE	O	—	—
240	PC11	I/O	$\overline{CASU}$	O	$\overline{BREQ}$	I

引脚编号	功能 4		功能 5		功能 6		弱保持	上拉
	引脚名	I/O	引脚名	I/O	引脚名	I/O		
237								
238	—	—	—	—	—	—	○	
239	—	—	—	—	—	—	○	
240	AUDATA1	O	—	—	—	—	○	

【符号说明】

(s)：施密特

(a)：模拟

(o)：漏极开路

第 2 章 CPU

2.1 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 ×4 个）和系统寄存器 （32 位 ×4 个）共 3 种。

2.1.1 通用寄存器

通用寄存器如图 2.1 所示。通用寄存器为 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）。

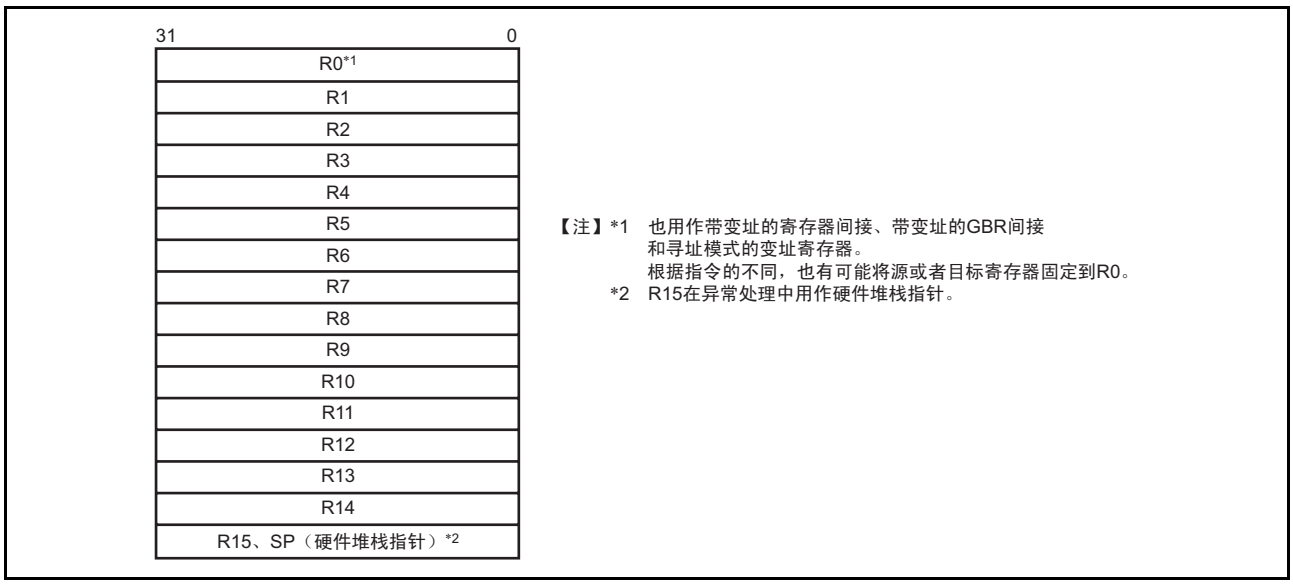


图 2.1 通用寄存器

2.1.2 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）、向量基址寄存器 （VBR）和转移表基址寄存器 （TBR）共 4 个。

SR 表示各种指令的处理状态。

GBR 作为 GBR 间接寻址模式的基地址，用于内置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。

TBR 用作函数表区域的基地址。

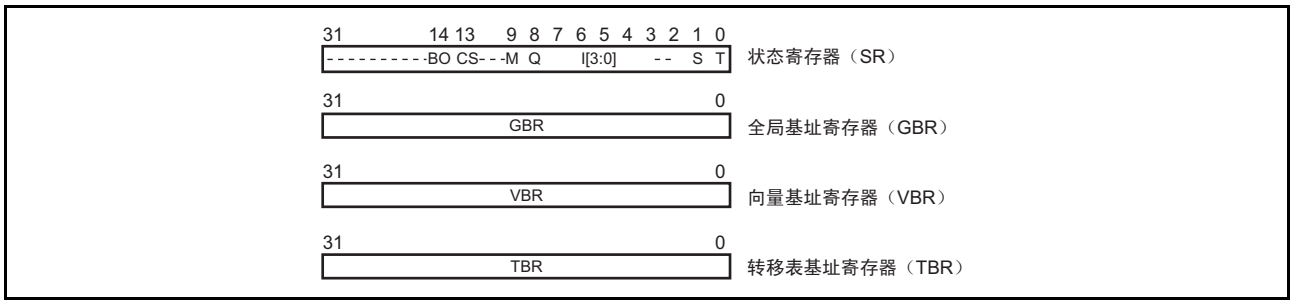


图 2.2 控制寄存器

（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
	-	BO	CS	-	-	-	M	Q	I[3:0]				-	-	S	T
初始值：	0	0	0	0	0	0	-	-	1	1	1	1	0	0	-	-
R/W：	R	R/W	R/W	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 ～ 15	—	均为 0	R	保留位 读写的值总是 0。
14	BO	0	R/W	BO 位 表示寄存器组上溢。
13	CS	0	R/W	CS 位 表示在 CLIP 指令的过程中，超过饱和上限值或者低于饱和下限值。
12 ～ 10	—	均为 0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说明
9	M	—	R/W	M 位 Q 位 通过 DIV0S、 DIV0U、 DIV1 指令使用。
8	Q	—	R/W	
7 ～ 4	I[3:0]	1111	R/W	中断屏蔽级别
3、 2	—	均为 0	R	保留位 读写的值总是 0。
1	S	—	R/W	S 位 指定 MAC 指令的饱和运行。
0	T	—	R/W	T 位 真 / 伪条件或者进位 / 借位位

（2） 全局基址寄存器 （GBR）

GBR 参照为 “GBR 参照 MOV 指令”的基地址。

（3） 向量基址寄存器 （VBR）

在产生异常及中断时， VBR 参照为转移目标的基地址。

（4） 转移表基址寄存器 （TBR）

在表参照子程序调用指令 JSR/N@@(disp8,TBR) 中，转移表基址寄存器参照为配置给存储器的函数表的起始地址。

2.1.3 系统寄存器

系统寄存器为 32 位长，有乘加寄存器 （MACH、 MACL）、过程寄存器 （PR）和程序计数器 （PC）共 4 个。MACH、MACL 保存乘法或者乘加运算的结果。PR 保存子程序过程的返回地址。PC 表示正在执行的程序地址，并控制处理的流程。

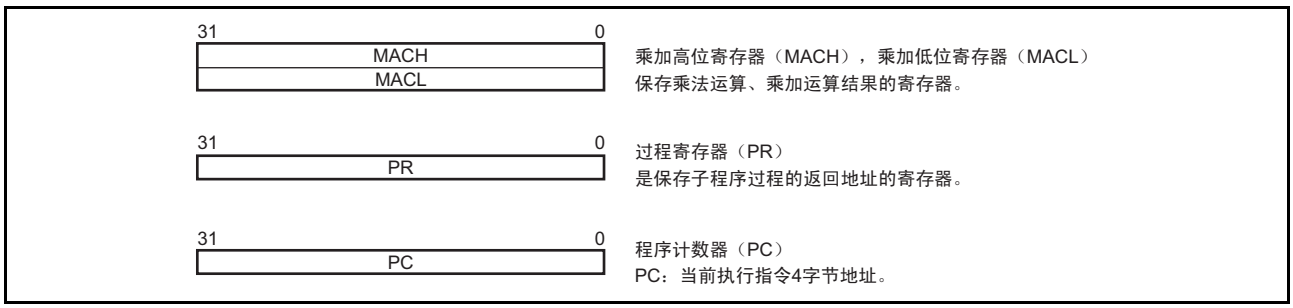


图 2.3 系统寄存器

（1） 乘加高位寄存器 （MACH）、乘加低位寄存器 （MACL）

MACH 及 MACL 用作 MAC 指令的加法运算值，同时也用于保存 MAC 指令、 MUL 指令的运算结果。

（2） 过程寄存器 （PR）

PR 保存使用 BSR、 BSRF、 JSR 指令的子程序调用的返回地址。 PR 通过子程序的返回指令被参照。

（3） 程序计数器 （PC）

PC 表示执行中的指令地址。

2.1.4 寄存器组

通用寄存器的 R0 ～ R14、控制寄存器的 GBR、系统寄存器的 MACH、MACL、PR 共 19 个 32 位寄存器可以使用寄存器组进行高速的寄存器保存和返回。向 CPU 接收到使用寄存器组的中断后，自动进行对存储体的保存。通过向中断处理程序发送 RESBANK 指令，执行从存储体的返回。

本 LSI 有 15 个存储体。详细请参照 《SH-2A、 SH2A-FPU 软件手册》、“6.8 寄存器组”。

2.1.5 寄存器初始值

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	I[3:0] 为 1111 （H'F）， BO、 CS 为 0，保留位为 0，其他位不定
	GBR、 TBR	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

2.2 数据格式

2.2.1 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），就符号扩展为长字，保存到寄存器。



图 2.4 寄存器的数据格式

2.2.2 存储器的数据格式

有字节、字和长字数据格式。存储器可以用 8 位字节、16 位字和 32 位长字中的任意形式存取。未满 32 位的存储器操作数则在符号扩展或者零扩展后，保存到寄存器中。

字操作数请从字边界 （2 字节的偶数地址：2n 地址）、长字操作数请从长字边界 （4 字节的偶数地址：4n 地址）进行存取。否则将发生地址错误。字节操作数可以从任意地址存取。

数据格式只可选择大端法的字节顺序。

存储器的数据格式如图 2.5 所示。

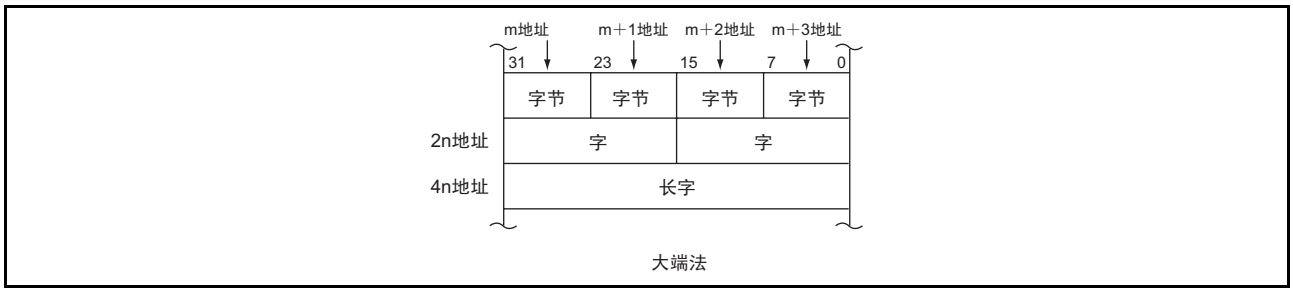


图 2.5 存储器的数据格式

2.2.3 立即数的数据格式

字节 （8 位）的立即数分配在操作码中。

MOV、ADD、CMP/EQ 指令将立即数进行符号扩展后作为长字进行运算，TST、AND、OR、XOR 指令将立即数进行零扩展后作为长字进行运算。因此， AND 指令使用立即数时，总是清除目标寄存器的高 24 位。

20 位的立即数分配在 32 位长的传送指令 MOVI20 及 MOVI20S 的码中。MOVI20 指令将立即数符号扩展并保存到目标寄存器。 MOVI20S 指令将立即数向高位移动 8 位，符号扩展并保存到目标寄存器。

字和长字的立即数不分配在操作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。

具体事例请参照 “2.3.1 （10） 立即数”。

2.3 指令特点

2.3.1 RISC 方式

指令为 RISC 方式，特点如下：

（1） 16 位固定长指令

基本指令是 16 位固定长，因此能提高程序码的效率。

（2） 32 位固定长指令

SH-2A 中追加了 32 位固定长指令。因此提高了性能和使用的便利性。

（3） 1 条指令 /1 个状态

采用流水线方式，能在 1 个状态周期执行 1 条基本指令。

（4） 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，在进行符号扩展后作为长字进行运算；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，作为长字进行运算。

表 2.2 字数据的符号扩展

SH-2A CPU		说明	其他 CPU 的例子
MOV.W ADD . . . . . . .DATA.W	@(disp,PC),R1 R1,R0 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。

（5） 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，进行与存储器的运算（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。

（6） 延迟转移

无条件转移指令等是除去一部分指令的延迟转移指令。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。

延迟转移中，在槽指令执行之后发生转移。但是，指令 （寄存器更新等）仍然按延迟转移指令 → 延迟槽指令的顺序执行。例如，即使在延迟槽中更改保存了转移地址的寄存器，但转移地址仍然是更改前的寄存器内容。

表 2.3 延迟转移指令

SH-2A CPU		说明	其他 CPU 的例子
BRA ADD	TRGET R1,R0	在转移到 TRGET 前执行 ADD。	ADD.W R1,R0 BRA TRGET

（7） 无延迟槽的无条件转移指令

SH-2A 中追加了不执行延迟槽指令的无条件转移指令。因此可削减无用的 NOP 指令，并能减少代码长度。

（8） 乘法 / 乘加运算

以 1 ～ 2 个状态执行 16×16→32 的乘法运算，以 2 ～ 3 个状态执行 16×16+64→64 的乘加运算，以 2 ～ 4 个状态执行 32×32→64 的乘法运算和 32×32+64→64 的乘加运算。

（9） T 位

比较结果反映在状态寄存器（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

SH-2A CPU	说明	其他 CPU 的例子
CMP/GE R1,R0 BT TRGET0 BF TRGET1	当 R0 ≥ R1 时， T 位被置位。 当 R0 ≥ 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

（10）立即数

字节的立即数分配在操作码中。字和长字的立即数不分配在操作码中而分配在存储器的表中。通过使用了带位移量 PC 相对寻址模式的立即数的数据传送指令 （MOV），参照存储器的表。

在 SH-2A 中， 17 ～ 28 位的立即数也可分配在操作码中。但是，关于 21 ～ 28 位的立即数，在寄存器传送后需要执行 OR 指令。

表 2.5 立即数的参照

区分	SH-2A CPU	其他的 CPU 例子
8 位立即数	MOV #H'12,R0	MOV.B #H'12,R0
16 位立即数	MOVI20 #H'1234,R0	MOV.W #H'1234,R0
20 位立即数	MOVI20 #H'12345,R0	MOV.L #H'12345,R0
28 位立即数	MOVI20S #H'12345,R0 OR #H'67,R0	MOV.L #H'1234567,R0
32 位立即数	MOV.L @(disp,PC),R0 …… …… .DATA.L H'12345678	MOV.L #H'12345678,R0

【注】 用 @(disp,PC) 访问立即数。

（11）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。

另外在 SH-2A 中，使用小于等于 28 位的绝对地址参照数据时，也可将分配在操作码中的立即数转送到寄存器，并通过寄存器间接寻址方式参照数据。但在使用 21 ～ 28 位的绝对地址参照数据时，在寄存器传送后需要使用 OR 指令。

表 2.6 绝对地址的参照

区分	SH-2A CPU	其他 CPU 的例子
小于等于 20 位	MOVI20 #H'12345,R1 MOV.B @R1,R0	MOV.B @H'12345,R0
21 ～ 28 位	MOVI20S #H'12345,R1 OR #H'67,R1 MOV.B @R1,R0	MOV.B @H'1234567,R0
大于等于 29 位	MOV.L @(disp,PC),R1 MOV.B @R1,R0 . . . . . . .DATA.L H'12345678	MOV.B @H'12345678,R0

（12）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，使用装入立即数的方法将该值传送到寄存器，并通过带变址的寄存器间接寻址方式参照数据。

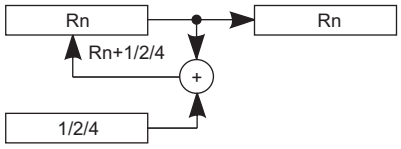
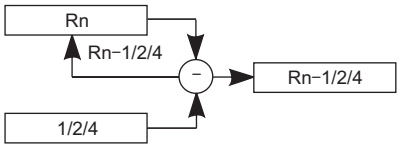
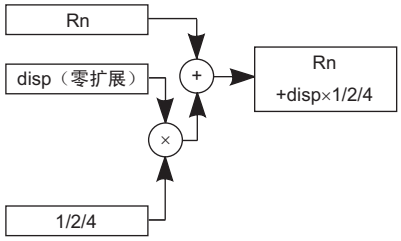
表 2.7 位移量的参照

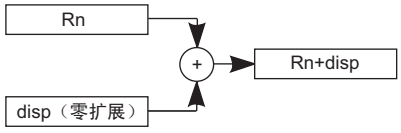
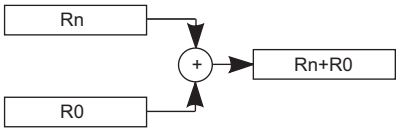
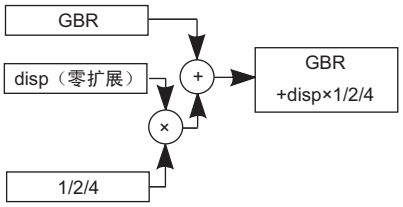
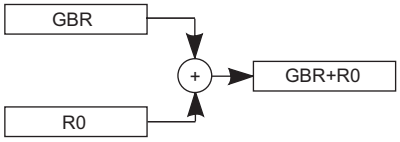
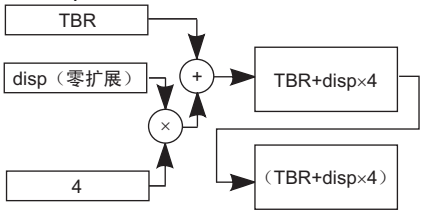
区分	SH-2A CPU	其他 CPU 的例子
16 位的位移量	MOV.W @(disp,PC),R0 MOV.W @(R0,R1),R2 . . . . . . .DATA.W H'1234	MOV.W @(H'1234,R1),R2

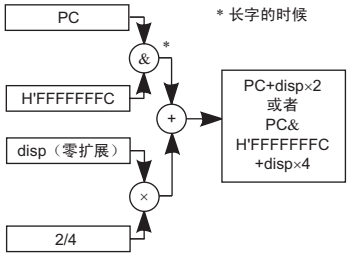
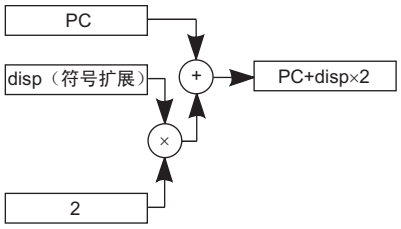
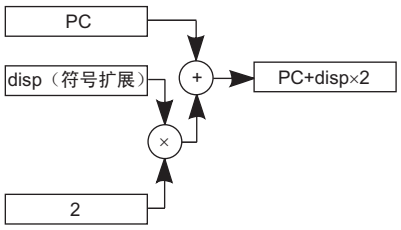
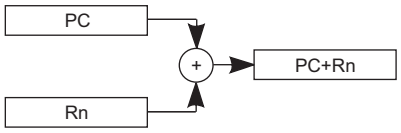
2.3.2 寻址方式

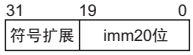
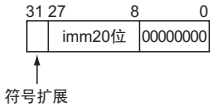
寻址方式和有效地址的计算方法如下所示。

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。 	Rn 指令执行后 字节：Rn+1→Rn 字：Rn+2→Rn 长字：Rn+4→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→Rn 字：Rn–2→Rn 长字：Rn–4→Rn （通过计算后的 Rn 执行指令）
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×2 长字：Rn+disp×4

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:12,Rn)	有效地址为寄存器 Rn 加上 12 位位移量 disp 后的内容。 disp 进行零扩展。 	字节：Rn+disp 字：Rn+disp 长字：Rn+disp
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×2 长字：GBR+disp×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
带位移量的 TBR 双重间接	@@(disp:8,TBR)	有效地址为寄存器 TBR 加上 8 位位移量 disp 后的内容。 disp 在进行零扩展后乘 4。 	（TBR+disp×4）地址的内容

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)	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。 	字：PC+disp×2 长字： PC&H'FFFFFFFC+disp×4
PC 相对	disp:8	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。 	PC+disp×2
	disp:12	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。 	PC+disp×2
	Rn	有效地址为寄存器 PC 加上 Rn 后的内容。 	PC+Rn

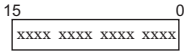
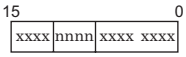
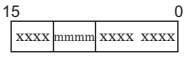
寻址方式	指令格式	有效地址的计算方法	计算式
立即数	#imm:20	MOVI20 指令的 20 位立即数 imm 进行符号扩展。 	—
		MOVI20S 指令的 20 位立即数 imm 向左移 8 位，高位进行符号扩展，低位进行零填充。 	—
	#imm:8	TST、 AND、 OR、 XOR 指令的 8 位立即数 imm 进行零扩展。	—
	#imm:8	MOV、 ADD、 CMP/EQ 指令的 8 位立即数 imm 进行符号扩展。	—
	#imm:8	TRAPA 指令的 8 位立即数 imm 进行零扩展后乘 4。	—
	#imm:3	BAND、 BOR、 BXOR、 BST、 BLD、 BSET、 BCLR 指令的 3 位立即数 imm 表示位的位置。	—

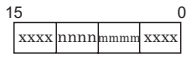
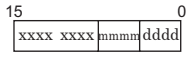
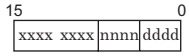
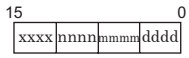
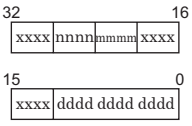
2.3.3 指令格式

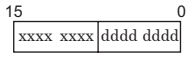
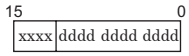
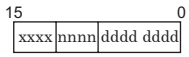
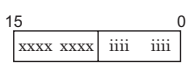
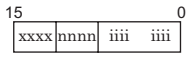
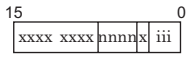
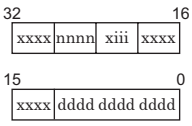
本章节说明指令格式、源操作数和目标操作数。操作数的含义因操作码而不同。符号如下：

xxxx：操作码
mmmm：源寄存器
nnnn：目标寄存器
iiii：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	MOV T Rn
		控制寄存器 或者系统寄存器	nnnn：寄存器直接	STS MACH,Rn
		R0 （寄存器直接）	nnnn：寄存器直接	DIVU R0, 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
		mmmm：寄存器直接	R15 （先减寄存器间接）	MOVMU.L Rm,@-R15
		R15 （后增寄存器间接）	nnnn：寄存器直接	MOVMU.L @R15+,Rn
		R0 （寄存器直接）	nnnn：后增寄存器间接	MOV.L R0,@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：先减寄存器间接	R0 （寄存器直接）	MOV.L @-Rm,R0
		mmmm：使用 Rm 的 PC 相对	—	BRAF Rm

指令格式		源操作数	目标操作数	指令例子
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：寄存器直接	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)
md 格式		mmmmdddd： 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）	nnnndddd： 带位移量的寄存器间接	MOV.B R0,@(disp,Rn)
nmd 格式		mmmm：寄存器直接	nnnndddd： 带位移量的寄存器间接	MOV.L Rm,@(disp,Rn)
		mmmmdddd： 带位移量的寄存器间接	nnnn：寄存器直接	MOV.L @(disp,Rm),Rn
nmd12 格式		mmmm：寄存器直接	nnnndddd： 带位移量的寄存器间接	MOV.L Rm,@(disp12,Rn)
		mmmmdddd： 带位移量的寄存器间接	nnnn：寄存器直接	MOV.L @(disp12,Rm),Rn

指令格式		源操作数	目标操作数	指令例子
d 格式		dddddddd： 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）	dddddddd： 带位移量的 GBR 间接	MOV.L　R0,@(disp,GBR)
		dddddddd： 带位移量的 PC 相对	R0 （寄存器直接）	MOVA　@(disp,PC),R0
		dddddddd： 带位移量的 TBR 双重间接	—	JSR/N　@@(disp8,TBR)
		dddddddd：PC 相对	—	BF　label
d12 格式		ddddddddddd： PC 相对	—	BRA　label (label=disp+PC)
nd8 格式		dddddddd： 带位移量的 PC 相对	nnnn：寄存器直接	MOV.L　@(disp,PC),Rn
i 格式		iiiiiiii：立即数	带变址的 GBR 间接	AND.B　#imm,@(R0,GBR)
		iiiiiiii：立即数	R0 （寄存器直接）	AND　#imm,R0
		iiiiiiii：立即数	—	TRAPA　#imm
ni 格式		iiiiiiii：立即数	nnnn：寄存器直接	ADD　#imm,Rn
ni3 格式		nnnn：寄存器直接 iii：立即数	—	BLD　#imm3,Rn
		—	nnnn：寄存器直接 iii：立即数	BST　#imm3,Rn
ni20 格式		iiiiiiiiiiiiiiiiiiii：立即数	nnnn：寄存器直接	MOVI20　#imm20,Rn
nid 格式		nnnnddddddddddddd： 带位移量的寄存器间接 iii：立即数	—	BLD.B　#imm3,@(disp12,Rn)
		—	nnnnddddddddddddd： 带位移量的寄存器间接 iii：立即数	BST.B　#imm3,@(disp12,Rn)

【注】 *　在乘加指令中 nnnn 为源寄存器。

2.4 指令系统

2.4.1 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	13	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 传送反向堆栈	62
		MOVA	传送有效地址	
		MOVI20	传送 20 位立即数	
		MOVI20S	传送 20 位立即数 左移 8 位	
		MOVML	R0 ～ Rn 的寄存器保存・返回	
		MOVMU	Rn ～ R14、 PR 的寄存器保存・返回	
		MOVRT	T 位取反并传送到 Rn	
		MOVT	传送 T 位	
		MOVU	传送无符号数据	
		NOTT	T 位取反	
		PREF	预取操作数高速缓存	
		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	26	ADD	2 进制加法	40
		ADDC	带进位的 2 进制加法	
		ADDV	带上溢的 2 进制加法	
		CMP/cond	比较	
		CLIPS	带符号的饱和值比较	
		CLIPU	无符号的饱和值比较	
		DIVS	带符号除法 （32 ÷ 32）	
		DIVU	无符号除法 （32 ÷ 32）	
		DIV1	单步除法	
		DIV0S	带符号的单步除法初始化	

分类	指令的种类	操作码	功能	指令数
算术运算指令	26	DIV0U	无符号的单步除法初始化	40
		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	
		MULR	带 Rn 结果保存符号的乘法	
		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	
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	12	ROTL	左循环 1 位	16
		ROTR	右循环 1 位	
		ROTCL	带 T 位的左循环 1 位	
		ROTCR	带 T 位的右循环 1 位	
		SHAD	动态算术移位	
		SHAL	算术左移 1 位	
		SHAR	算术右移 1 位	
		SHLD	动态逻辑移位	
		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	

分类	指令的种类	操作码	功能	指令数
转移指令	10	BF	条件转移、带延迟的条件转移 （当 T=0 时进行转移）	15
		BT	条件转移、带延迟的条件转移 （当 T=1 时进行转移）	
		BRA	带延迟的无条件转移	
		BRAF	带延迟的无条件转移	
		BSR	转移到带延迟的子程序过程	
		BSRF	转移到带延迟的子程序过程	
		JMP	带延迟的无条件转移	
		JSR	转移到子程序过程，转移到带延迟的子程序过程	
		RTS	从子程序过程返回，从带延迟的子程序过程返回	
		RTV/N	从带 Rm→R0 传送的子程序过程返回	
系统控制指令	14	CLRT	清除 T 位	36
		CLRMAC	清除 MAC 寄存器	
		LDBANK	从指定寄存器组入口的寄存器返回	
		LDC	装入到控制寄存器	
		LDS	装入到系统寄存器	
		NOP	无操作	
		RESBANK	从寄存器组的寄存器返回	
		RTE	从异常处理返回	
		SETT	T 位的置位	
		SLEEP	转移到低功耗状态	
		STBANK	寄存器保存到指定寄存器组入口	
		STC	控制寄存器的保存	
		STS	系统寄存器的保存	
		TRAPA	陷阱异常处理	
浮点运算指令	19	FABS	浮点数绝对值	48
		FADD	浮点数加法	
		FCMP	浮点数比较	
		FCNVDS	将双精度转换为单精度	
		FCNVSD	将单精度转换为双精度	
		FDIV	浮点数除法	
		FLDI0	浮点数装入立即数 0	
		FLDI1	浮点数装入立即数 1	
		FLDS	将浮点数装入到系统寄存器 FPUL	
		FLOAT	将整数转换为浮点数	
		FMAC	浮点数乘加运算	

分类	指令的种类	操作码	功能	指令数
浮点运算指令	19	FMOV	传送浮点数	48
		FMUL	浮点数乘法	
		FNEG	浮点数符号取反	
		FSCHG	SZ 位取反	
		FSQRT	浮点平方根	
		FSTS	来自系统寄存器 FPUL 的浮点数保存	
		FSUB	浮点数减法	
		FTRC	下舍入为浮点数的整数	
有关 FPU 的 CPU 指令	2	LDS	装入到浮点系统寄存器	8
		STS	来自浮点系统寄存器的保存	
位操作指令	10	BAND	位逻辑与	14
		BCLR	清除位	
		BLD	位装入	
		BOR	位逻辑或	
		BSET	置位	
		BST	位保存	
		BXOR	位逻辑异或	
		BANDNOT	位逻辑与非	
		BORNOT	位逻辑或非	
		BLDNOT	位非装入	
	计 112			253

通过以下格式按照分类顺序说明指令的操作码、操作和执行状态。

指令	操作码	操作概略	执行状态	T 位
用助记符表示。 【符号说明】 Rm ：源寄存器 Rn ：目标寄存器 imm ：立即数 disp ：位移量 *2	按照 MSB←→LSB 的顺序表示。 【符号说明】 mmmm：源寄存器 nnnn：目标寄存器 0000：R0 0001：R1 ····· 1111：R15 iiii：立即数 dddd：位移量	表示操作的概略 【符号说明】 →、←：传送方向 (xx) ：存储器操作数 M/Q/T ：SR 内的标志位 & ：每位的逻辑“与” | ：每位的逻辑“或” ^ ：每位的逻辑“异或” ~ ：每位的逻辑“非” <<n ：左移 n 位 >>n ：右移 n 位	是无等待时的值。 *1	表示执行指令后的 T 位的值。 【符号说明】 —：不变化

【注】 *1 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，指令执行的状态数将会增加。

（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定标 （×1、 ×2、 ×4）。

详细内容请参照 《SH-2A、 SH2A-FPU软件手册》。

2.4.2 数据传送指令

表 2.11 数据传送指令

指令		操作码	操作	执行状态	T 位	适用指令		
						SH2、SH2E	SH4	SH-2A
MOV	#imm,Rn	1110nnnniiiiiiii	imm→ 符号扩展 →Rn	1	—	○	○	○
MOV.W	@(disp,PC),Rn	1001nnnndddddddd	(disp×2+PC)→ 符号扩展 →Rn	1	—	○	○	○
MOV.L	@(disp,PC),Rn	1101nnnndddddddd	(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	—	○	○	○

指令		操作码	操作	执行状态	T 位	适用指令		
						SH2、SH2E	SH4	SH-2A
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	—	○	○	○
MOV.B	R0,@Rn+	0100nnnn10001011	R0→(Rn),Rn+1→Rn	1	—			○
MOV.W	R0,@Rn+	0100nnnn10011011	R0→(Rn),Rn+2→Rn	1	—			○
MOV.L	R0,@Rn+	0100nnnn10101011	R0→(Rn),Rn+4→Rn	1	—			○
MOV.B	@-Rm,R0	0100mmmm11001011	Rm-1→Rm,(Rm)→ 符号扩展 →R0	1	—			○
MOV.W	@-Rm,R0	0100mmmm11011011	Rm-2→Rm,(Rm)→ 符号扩展 →R0	1	—			○
MOV.L	@-Rm,R0	0100mmmm11101011	Rm-4→Rm,(Rm)→R0	1	—			○
MOV.B	Rm,@(disp12,Rn)	0011nnnnmmmm0001 0000dddddddddddd	Rm→(disp+Rn)	1	—			○
MOV.W	Rm,@(disp12,Rn)	0011nnnnmmmm0001 0001dddddddddddd	Rm→(disp×2+Rn)	1	—			○
MOV.L	Rm,@(disp12,Rn)	0011nnnnmmmm0001 0010dddddddddddd	Rm→(disp×4+Rn)	1	—			○
MOV.B	@(disp12,Rm),Rn	0011nnnnmmmm0001 0100dddddddddddd	(disp+Rm)→ 符号扩展 →Rn	1	—			○
MOV.W	@(disp12,Rm),Rn	0011nnnnmmmm0001 0101dddddddddddd	(disp×2+Rm)→ 符号扩展 →Rn	1	—			○
MOV.L	@(disp12,Rm),Rn	0011nnnnmmmm0001 0110dddddddddddd	(disp×4+Rm)→Rn	1	—			○
MOVA	@(disp,PC),R0	11000111dddddddd	disp×4+PC→R0	1	—	○	○	○
MOVI20	#imm20,Rn	0000nnnniiii0000 iiiiiiiiiiiiiiii	imm→ 符号扩展 →Rn	1	—			○
MOVI20S	#imm20,Rn	0000nnnniiii0001 iiiiiiiiiiiiiiii	imm<<8→ 符号扩展 →Rn	1	—			○
MOVML.L	Rm,@-R15	0100mmmm11110001	R15-4→R15,Rm→(R15) R15-4→R15,Rm-1→(R15) ： R15-4→R15,R0→(R15) ※ Rm=R15 时， Rm 读作 PR	1 ～ 16	—			○

指令	操作码	操作	执行状态	T 位	适用指令		
					SH2、SH2E	SH4	SH-2A
MOVML.L @R15+,Rn	0100nnnn11110101	(R15)→R0,R15+4→R15 (R15)→R1,R15+4→R15 ： (R15)→Rn ※ Rn=R15 时， Rn 读作 PR	1 ～ 16	—			○
MOVMU.L Rm,@-R15	0100mmmm11110000	R15-4→R15,PR→(R15) R15-4→R15,R14→(R15) ： R15-4→R15,Rm→(R15) ※ Rm=R15 时， Rm 读作 PR	1 ～ 16	—			○
MOVMU.L @R15+,Rn	0100nnnn11110100	(R15)→Rn,R15+4→R15 (R15)→Rn+1,R15+4→R15 ： (R15)→R14,R15+4→R15 (R15)→PR ※ Rn=R15 时， Rn 读作 PR	1 ～ 16	—			○
MOVRT Rn	0000nnnn00111001	~T→Rn	1	—			○
MOVT Rn	0000nnnn00101001	T→Rn	1	—	○	○	○
MOVU.B @(disp12,Rm),Rn	0011nnnnmmmm0001 1000dddddddddddd	(disp+Rm)→ 零扩展 →Rn	1	—			○
MOVU.W @(disp12,Rm),Rn	0011nnnnmmmm0001 1001dddddddddddd	(disp×2+Rm)→ 零扩展 →Rn	1	—			○
NOTT	0000000001101000	~T→T	1	运算结果			○
PREF @Rn	0000nnnn10000011	(Rn)→ 操作数高速缓存	1	—		○	○
SWAP.B Rm,Rn	0110nnnnmmmm1000	Rm→ 低位 2 字节的高低字节交换 →Rn	1	—	○	○	○
SWAP.W Rm,Rn	0110nnnnmmmm1001	Rm→ 交换高低字 →Rn	1	—	○	○	○
XTRCT Rm,Rn	0010nnnnmmmm1101	Rm:Rn 的中间 32 位 →Rn	1	—	○	○	○

2.4.3 算术运算指令

表 2.12 算术运算指令

指令		操作码	操作	执行状态	T 位	适用指令		
						SH2、SH2E	SH4	SH-2A
ADD	Rm,Rn	0011nnnnmmmm1100	Rn+Rm→Rn	1	—	○	○	○
ADD	#imm,Rn	0111nnnniiiiiiii	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	10001000iiiiiiii	R0=imm 时 1→T 此外 0→T	1	比较结果	○	○	○
CMP/EQ	Rm,Rn	0011nnnnmmmm0000	Rn=Rm 时 1→T 此外 0→T	1	比较结果	○	○	○
CMP/HS	Rm,Rn	0011nnnnmmmm0010	无符号 Rn ≥ Rm 时 1→T 此外 0→T	1	比较结果	○	○	○
CMP/GE	Rm,Rn	0011nnnnmmmm0011	有符号 Rn ≥ Rm 时 1→T 此外 0→T	1	比较结果	○	○	○
CMP/HI	Rm,Rn	0011nnnnmmmm0110	无符号 Rn > Rm 时 1→T 此外 0→T	1	比较结果	○	○	○
CMP/GT	Rm,Rn	0011nnnnmmmm0111	有符号 Rn > Rm 时 1→T 此外 0→T	1	比较结果	○	○	○
CMP/PL	Rn	0100nnnn00010101	Rn > 0 时 1→T 此外 0→T	1	比较结果	○	○	○
CMP/PZ	Rn	0100nnnn00010001	Rn ≥ 0 时 1→T 此外 0→T	1	比较结果	○	○	○
CMP/STR	Rm,Rn	0010nnnnmmmm1100	当任意字节相等时 1→T 此外 0→T	1	比较结果	○	○	○
CLIPS.B	Rn	0100nnnn10010001	Rn > (H'0000007F) 时、(H'0000007F)→Rn,1→CS Rn < (H'FFFFFF80) 时、(H'FFFFFF80)→Rn,1→CS	1	—			○
CLIPS.W	Rn	0100nnnn10010101	Rn > (H'00007FFF) 时、(H'00007FFF)→Rn,1→CS Rn < (H'FFFF8000) 时、(H'FFFF8000)→Rn,1→CS	1	—			○
CLIPU.B	Rn	0100nnnn10000001	Rn > (H'000000FF) 时、(H'000000FF)→Rn,1→CS	1	—			○
CLIPU.W	Rn	0100nnnn10000101	Rn > (H'0000FFFF) 时、(H'0000FFFF)→Rn,1→CS	1	—			○
DIV1	Rm,Rn	0011nnnnmmmm0100	单步除法 (Rn ÷ Rm)	1	计算结果	○	○	○

指令		操作码	操作	执行状态	T 位	适用指令		
						SH2、SH2E	SH4	SH-2A
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	○	○	○
DIVS	R0,Rn	0100nnnn10010100	带符号 Rn ÷ R0→Rn 32 ÷ 32→32 位	36	—			○
DIVU	R0,Rn	0100nnnn10000100	无符号 Rn ÷ R0→Rn 32 ÷ 32→32 位	34	—			○
DMULS.L	Rm,Rn	0011nnnnmmmm1101	带符号 Rn×Rm→MACH,MACL 32×32→64 位	2	—	○	○	○
DMULU.L	Rm,Rn	0011nnnnmmmm0101	无符号 Rn×Rm→MACH,MACL 32×32→64 位	2	—	○	○	○
DT	Rn	0100nnnn00010000	Rn-1→Rn,Rn 为 0 时 1→T 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	4	—	○	○	○
MAC.W	@Rm+,@Rn+	0100nnnnmmmm1111	带符号 (Rn)×(Rm)+MAC→MAC 16×16+64→64 位	3	—	○	○	○
MUL.L	Rm,Rn	0000nnnnmmmm0111	Rn×Rm→MACL 32×32→32 位	2	—	○	○	○
MULR	R0,Rn	0100nnnn10000000	R0×Rn→Rn 32×32→32 位	2				○
MULS.W	Rm,Rn	0010nnnnmmmm1111	带符号 Rn×Rm→MACL 16×16→32 位	1	—	○	○	○
MULU.W	Rm,Rn	0010nnnnmmmm1110	无符号 Rn×Rm→MACL 16×16→32 位	1	—	○	○	○
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	上溢	○	○	○

2.4.4 逻辑运算指令

表 2.13 逻辑运算指令

指令	操作码	操作	执行状态	T 位	适用指令		
					SH2、SH2E	SH4	SH-2A
AND Rm,Rn	0010nnnnmmmm1001	Rn&Rm→Rn	1	—	○	○	○
AND #imm,R0	11001001iiiiiiii	R0&imm→R0	1	—	○	○	○
AND.B #imm,@(R0,GBR)	11001101iiiiiiii	(R0+GBR)&imm→(R0+GBR)	3	—	○	○	○
NOT Rm,Rn	0110nnnnmmmm0111	~Rm→Rn	1	—	○	○	○
OR Rm,Rn	0010nnnnmmmm1011	Rn | Rm→Rn	1	—	○	○	○
OR #imm,R0	11001011iiiiiiii	R0 | imm→R0	1	—	○	○	○
OR.B #imm,@(R0,GBR)	11001111iiiiiiii	(R0+GBR) | imm→(R0+GBR)	3	—	○	○	○
TAS.B @Rn	0100nnnn00011011	(Rn) 为 0 时 1→T, 此外 0→T, 1→MSB of(Rn)	3	测试结果	○	○	○
TST Rm,Rn	0010nnnnmmmm1000	Rn&Rm, 结果为 0 时 1→T, 其他 0→T	1	测试结果	○	○	○
TST #imm,R0	11001000iiiiiiii	R0&imm, 结果为 0 时 1→T 其他 0→T	1	测试结果	○	○	○
TST.B #imm,@(R0,GBR)	11001100iiiiiiii	(R0+GBR)&imm, 结果为 0 时 1→T，其他 0→T	3	测试结果	○	○	○
XOR Rm,Rn	0010nnnnmmmm1010	Rn^Rm→Rn	1	—	○	○	○
XOR #imm,R0	11001010iiiiiiii	R0^imm→R0	1	—	○	○	○
XOR.B #imm,@(R0,GBR)	11001110iiiiiiii	(R0+GBR)^imm→(R0+GBR)	3	—	○	○	○

2.4.5 移位指令

表 2.14 移位指令

指令	操作码	操作	执行状态	T 位	适用指令		
					SH2、SH2E	SH4	SH-2A
ROTL Rn	0100nnnn00000100	T←Rn←MSB	1	MSB	○	○	○
ROTR Rn	0100nnnn00000101	LSB→Rn→T	1	LSB	○	○	○
ROTCL Rn	0100nnnn00100100	T←Rn←T	1	MSB	○	○	○
ROTCR Rn	0100nnnn00100101	T→Rn→T	1	LSB	○	○	○
SHAD Rm,Rn	0100nnnnmmmm1100	Rm ≥ 0 时 Rn<<Rm→Rn Rm < 0 时 Rn>>|Rm|→[MSB→Rn]	1	—		○	○
SHAL Rn	0100nnnn00100000	T←Rn←0	1	MSB	○	○	○
SHAR Rn	0100nnnn00100001	MSB→Rn→T	1	LSB	○	○	○
SHLD Rm,Rn	0100nnnnmmmm1101	Rm ≥ 0 时 Rn<<Rm→Rn Rm < 0 时 Rn>>|Rm|→[0→Rn]	1	—		○	○
SHLL Rn	0100nnnn00000000	T←Rn←0	1	MSB	○	○	○
SHLR Rn	0100nnnn00000001	0→Rn→T	1	LSB	○	○	○
SHLL2 Rn	0100nnnn00001000	Rn<<2→Rn	1	—	○	○	○
SHLR2 Rn	0100nnnn00001001	Rn>>2→Rn	1	—	○	○	○
SHLL8 Rn	0100nnnn00011000	Rn<<8→Rn	1	—	○	○	○
SHLR8 Rn	0100nnnn00011001	Rn>>8→Rn	1	—	○	○	○
SHLL16 Rn	0100nnnn00101000	Rn<<16→Rn	1	—	○	○	○
SHLR16 Rn	0100nnnn00101001	Rn>>16→Rn	1	—	○	○	○

2.4.6 转移指令

表 2.15 转移指令

指令	操作码	操作	执行状态	T 位	适用指令		
					SH2、SH2E	SH4	SH-2A
BF label	10001011dddddddd	T=0 时 disp×2+PC→PC, T=1 时 nop	3/1*	—	○	○	○
BF/S label	10001111dddddddd	延迟转移 ,T=0 时 disp×2+PC→PC,T=1 时 nop	2/1*	—	○	○	○
BT label	10001001dddddddd	T=1 时 disp×2+PC→PC, T=0 时 nop	3/1*	—	○	○	○
BT/S label	10001101dddddddd	延迟转移 ,T=1 时 disp×2+PC→PC,T=0 时 nop	2/1*	—	○	○	○
BRA label	1010dddddddddddd	延迟转移， disp×2+PC→PC	2	—	○	○	○
BRAF Rm	0000mmmm00100011	延迟转移， Rm+PC→PC	2	—	○	○	○
BSR label	1011dddddddddddd	延迟转移， PC→PR,disp×2+PC→PC	2	—	○	○	○
BSRF Rm	0000mmmm00000011	延迟转移， PC→PR,Rm+PC→PC	2	—	○	○	○
JMP @Rm	0100mmmm00101011	延迟转移， Rm→PC	2	—	○	○	○
JSR @Rm	0100mmmm00001011	延迟转移， PC→PR,Rm→PC	2	—	○	○	○
JSR/N @Rm	0100mmmm01001011	PC-2→PR,Rm→PC	3	—			○
JSR/N @@(disp8,TBR)	10000011dddddddd	PC-2→PR,(disp×4+TBR)→PC	5	—			○
RTS	0000000000001011	延迟转移， PR→PC	2	—	○	○	○
RTS/N	0000000001101011	PR→PC	3	—			○
RTV/N Rm	0000mmmm01111011	Rm→R0,PR→PC	3	—			○

【注】 * 在不转移时为一个状态。

2.4.7 系统控制指令

表 2.16 系统控制指令

指令		操作码	操作	执行状态	T 位	适用指令		
						SH2、SH2E	SH4	SH-2A
CLRT		0000000000001000	0→T	1	0	○	○	○
CLRMAC		0000000000101000	0→MACH,MACL	1	—	○	○	○
LDBANK	@Rm,R0	0100mmmm11100101	（指定寄存器组入口）→R0	6	—			○
LDC	Rm,SR	0100mmmm00001110	Rm→SR	3	LSB	○	○	○
LDC	Rm,TBR	0100mmmm01001010	Rm→TBR	1	—			○
LDC	Rm,GBR	0100mmmm00011110	Rm→GBR	1	—	○	○	○
LDC	Rm,VBR	0100mmmm00101110	Rm→VBR	1	—	○	○	○
LDC.L	@Rm+,SR	0100mmmm00000111	(Rm)→SR,Rm+4→Rm	5	LSB	○	○	○
LDC.L	@Rm+,GBR	0100mmmm00010111	(Rm)→GBR,Rm+4→Rm	1	—	○	○	○
LDC.L	@Rm+,VBR	0100mmmm00100111	(Rm)→VBR,Rm+4→Rm	1	—	○	○	○
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	—	○	○	○
RESBANK		0000000001011011	存储体 →R0 ～ R14, GBR,MACH,MACL,PR	9*	—			○
RTE		0000000000101011	延迟转移、堆栈区 →PC/SR	6	—	○	○	○
SETT		0000000000011000	1→T	1	1	○	○	○
SLEEP		0000000000011011	睡眠	5	—	○	○	○
STBANK	R0,@Rn	0100nnnn11100001	R0→ （指定寄存器组入口）	7	—			○
STC	SR,Rn	0000nnnn00000010	SR→Rn	2	—	○	○	○
STC	TBR,Rn	0000nnnn01001010	TBR→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)	2	—	○	○	○
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	—	○	○	○
STS	PR,Rn	0000nnnn00101010	PR→Rn	1	—	○	○	○

指令		操作码	操作	执行状态	T 位	适用指令		
						SH2、SH2E	SH4	SH-2A
STS.L	MACH,@-Rn	0100nnnn00000010	Rn-4→Rn,MACH→(Rn)	1	—	○	○	○
STS.L	MACL,@-Rn	0100nnnn00010010	Rn-4→Rn,MACL→(Rn)	1	—	○	○	○
STS.L	PR,@-Rn	0100nnnn00100010	Rn-4→Rn,PR→(Rn)	1	—	○	○	○
TRAPA	#imm	11000011iiiiiiii	PC/SR→ 堆栈区、(imm×4+VBR)→PC	5	—	○	○	○

【注】 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，指令执行的状态数将会增加。

（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使用的寄存器相同时

* 存储体上溢时，状态数为 19。

2.4.8 浮点运算指令

表 2.17 浮点运算指令

指令		操作码	操作	执行状态	T 位	适用指令		
						SH2E	SH4	SH-2A/ SH2A-FPU
FABS	FRn	1111nnnn01011101	|FRn|→FRn	1	—	○	○	○
FABS	DRn	1111nnn001011101	|DRn|→DRn	1	—		○	○
FADD	FRm,FRn	1111nnnnmmmm0000	FRn+FRm→FRn	1	—	○	○	○
FADD	DRm,DRn	1111nnn0mmm00000	DRn+DRm→DRn	6	—		○	○
FCMP/EQ	FRm,FRn	1111nnnnmmmm0100	(FRn=FRm)? 1:0→T	1	比较结果	○	○	○
FCMP/EQ	DRm,DRn	1111nnn0mmm00100	(DRn=DRm)? 1:0→T	2	比较结果		○	○
FCMP/GT	FRm,FRn	1111nnnnmmmm0101	(FRn ＞ FRm)? 1:0→T	1	比较结果	○	○	○
FCMP/GT	DRm,DRn	1111nnn0mmm00101	(DRn ＞ DRm)? 1:0→T	2	比较结果		○	○
FCNVDS	DRm,FPUL	1111mmm010111101	(float)DRm→FPUL	2	—		○	○
FCNVSD	FPUL,DRn	1111nnn010101101	(double)FPUL→DRn	2	—		○	○
FDIV	FRm,FRn	1111nnnnmmmm0011	FRn/FRm→FRn	10	—	○	○	○
FDIV	DRm,DRn	1111nnn0mmm00011	DRn/DRm→DRn	23	—		○	○
FLDI0	FRn	1111nnnn10001101	0×00000000→FRn	1	—	○	○	○
FLDI1	FRn	1111nnnn10011101	0×3F800000→FRn	1	—	○	○	○
FLDS	FRm,FPUL	1111mmmm00011101	FRm→FPUL	1	—	○	○	○
FLOAT	FPUL,FRn	1111nnnn00101101	(float)FPUL→FRn	1	—	○	○	○
FLOAT	FPUL,DRn	1111nnn000101101	(double)FPUL→DRn	2	—		○	○
FMAC	FR0,FRm,FRn	1111nnnnmmmm1110	FR0×FRm+FRn→FRn	1	—	○	○	○
FMOV	FRm,FRn	1111nnnnmmmm1100	FRm→FRn	1	—	○	○	○
FMOV	DRm,DRn	1111nnn0mmm01100	DRm→DRn	2	—		○	○
FMOV.S	@(R0,Rm),FRn	1111nnnnmmmm0110	(R0+Rm)→FRn	1	—	○	○	○
FMOV.D	@(R0,Rm),DRn	1111nnn0mmmm0110	(R0+Rm)→DRn	2	—		○	○
FMOV.S	@Rm+,FRn	1111nnnnmmmm1001	(Rm)→FRn,Rm+=4	1	—	○	○	○
FMOV.D	@Rm+,DRn	1111nnn0mmmm1001	(Rm)→DRn,Rm+=8	2	—		○	○
FMOV.S	@Rm,FRn	1111nnnnmmmm1000	(Rm)→FRn	1	—	○	○	○
FMOV.D	@Rm,DRn	1111nnn0mmmm1000	(Rm)→DRn	2	—		○	○
FMOV.S	@(disp12,Rm),FRn	0011nnnnmmmm0001 0111dddddddddddd	(disp×4+Rm)→FRn	1	—			○

指令		操作码	操作	执行状态	T 位	适用指令		
						SH2E	SH4	SH-2A/ SH2A-FPU
FMOV.D	@(disp12,Rm),DRn	0011nnn0mmmm0001 0111dddddddddddd	(disp×8+Rm)→DRn	2	—			○
FMOV.S	FRm,@(R0,Rn)	1111nnnnmmmm0111	FRm→(R0+Rn)	1	—	○	○	○
FMOV.D	DRm,@(R0,Rn)	1111nnnnmmm00111	DRm→(R0+Rn)	2	—		○	○
FMOV.S	FRm,@-Rn	1111nnnnmmmm1011	Rn-=4,FRm→(Rn)	1	—	○	○	○
FMOV.D	DRm,@-Rn	1111nnnnmmm01011	Rn-=8,DRm→(Rn)	2	—		○	○
FMOV.S	FRm,@Rn	1111nnnnmmmm1010	FRm→(Rn)	1	—	○	○	○
FMOV.D	DRm,@Rn	1111nnnnmmm01010	DRm→(Rn)	2	—		○	○
FMOV.S	FRm,@(disp12,Rn)	0011nnnnmmmm0001 0011dddddddddddd	FRm→(disp×4+Rn)	1	—			○
FMOV.D	DRm,@(disp12,Rn)	0011nnnnmmm00001 0011dddddddddddd	DRm→(disp×8+Rn)	2	—			○
FMUL	FRm,FRn	1111nnnnmmmm0010	FRn×FRm→FRn	1	—	○	○	○
FMUL	DRm,DRn	1111nnn0mmm00010	DRn×DRm→DRn	6	—		○	○
FNEG	FRn	1111nnnn01001101	-FRn→FRn	1	—	○	○	○
FNEG	DRn	1111nnn001001101	-DRn→DRn	1	—		○	○
FSCHG		1111001111111101	FPSCR.SZ=~FPSCR.SZ	1	—		○	○
FSQRT	FRn	1111nnnn01101101	√ FRn→FRn	9	—		○	○
FSQRT	DRn	1111nnn001101101	√ DRn→DRn	22	—		○	○
FSTS	FPUL,FRn	1111nnnn00001101	FPUL→FRn	1	—	○	○	○
FSUB	FRm,FRn	1111nnnnmmmm0001	FRn-FRm→FRn	1	—	○	○	○
FSUB	DRm,DRn	1111nnn0mmm00001	DRn-DRm→DRn	6	—		○	○
FTRC	FRm,FPUL	1111mmmm00111101	(long)FRm→FPUL	1	—	○	○	○
FTRC	DRm,FPUL	1111mmm000111101	(long)DRm→FPUL	2	—		○	○

2.4.9 有关 FPU 的 CPU 指令

表 2.18 有关 FPU 的 CPU 指令

指令		操作码	操作	执行状态	T 位	适用指令		
						SH2E	SH4	SH-2A/ SH2A-FPU
LDS	Rm,FPSCR	0100mmmm01101010	Rm→FPSCR	1	—	○	○	○
LDS	Rm,FPUL	0100mmmm01011010	Rm→FPUL	1	—	○	○	○
LDS.L	@Rm+,FPSCR	0100mmmm01100110	(Rm)→FPSCR,Rm+=4	1	—	○	○	○
LDS.L	@Rm+,FPUL	0100mmmm01010110	(Rm)→FPUL,Rm+=4	1	—	○	○	○
STS	FPSCR,Rn	0000nnnn01101010	FPSCR→Rn	1	—	○	○	○
STS	FPUL,Rn	0000nnnn01011010	FPUL→Rn	1	—	○	○	○
STS.L	FPSCR,@-Rn	0100nnnn01100010	Rn-=4,FPSCR→(Rn)	1	—	○	○	○
STS.L	FPUL,@-Rn	0100nnnn01010010	Rn-=4,FPUL→(Rn)	1	—	○	○	○

2.4.10 位操作指令

表 2.19 位操作指令

指令		操作码	操作	执行状态	T 位	适用指令		
						SH2、SH2E	SH4	SH-2A
BAND.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0100dddddddddddd	(imm of (disp+Rn))&T → T	3	运算结果			○
BANDNOT.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 1100dddddddddddd	~(imm of(disp+Rn))&T → T	3	运算结果			○
BCLR.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0000dddddddddddd	0→(imm of(disp+Rn))	3	—			○
BCLR	#imm3,Rn	10000110nnnn0iii	0→imm of Rn	1	—			○
BLD.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0011dddddddddddd	(imm of (disp+Rn))→T	3	运算结果			○
BLD	#imm3,Rn	10000111nnnn1iii	imm of Rn→T	1	运算结果			○
BLDNOT.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 1011dddddddddddd	~(imm of(disp+Rn))→T	3	运算结果			○
BOR.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0101dddddddddddd	(imm of(disp+Rn))| T→T	3	运算结果			○
BORNOT.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 1101dddddddddddd	~(imm of(disp+Rn)) | T→T	3	运算结果			○
BSET.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0001dddddddddddd	1→(imm of (disp+Rn))	3	—			○
BSET	#imm3,Rn	10000110nnnn1iii	1→imm of Rn	1	—			○
BST.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0010dddddddddddd	T→(imm of (disp+Rn))	3	—			○
BST	#imm3,Rn	10000111nnnn0iii	T→imm of Rn	1	—			○
BXOR.B	#imm3,@(disp12,Rn)	0011nnnn0iii1001 0110dddddddddddd	(imm of (disp+Rn)) ^ T→ T	3	运算结果			○

2.5 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 5 种。状态间的转移如图 2.6 所示。

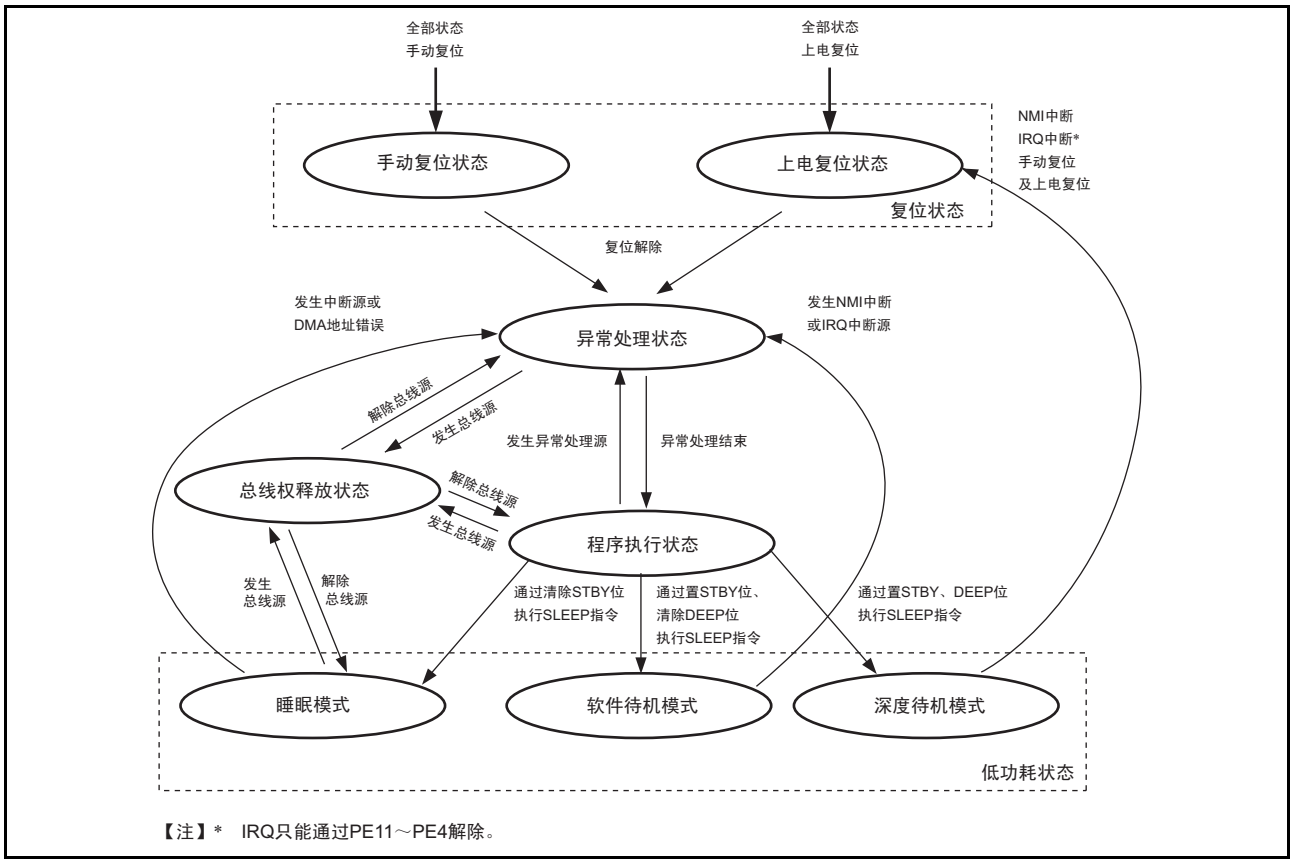


图 2.6 处理状态的状态转移图

（1） 复位状态

为 CPU 被复位的状态。复位有上电复位和手动复位两种。

（2） 异常处理状态

该状态为 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。。

复位时，从异常处理向量表取出作为程序计数器 （PC）初始值的执行起始地址与堆栈指针 （SP）的初始值并分别保存，转移到起始地址后开始执行程序。

中断时，参照 SP，将 PC 与状态寄存器（SR）保存到堆栈区。从异常处理向量表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。

之后，处理状态为程序执行状态。

（3） 程序执行状态

为 CPU 依次执行程序的状态。

（4） 低功耗状态

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、软件待机模式或深度待机模式。

（5） 总线权释放状态

CPU 对请求总线权的器件释放总线的状态

第 3 章 浮点运算单元 （FPU）

3.1 特点

FPU 有以下的特点：

- 依据IEEE754规格
- 16个单精度浮点寄存器 （也可作为8个双精度寄存器参照）
- 2种舍入模式：就近舍入和朝0舍入
- 非规格化数处理模式：下舍入为0 （刷新为0）
- 5个异常源：
  无效运算、被0除、上溢、下溢、不正确
- 包括指令：
  单精度、双精度、系统控制

3.2　数据格式

3.2.1　浮点格式

浮点由以下 3 个字段构成：

- 符号 （s）
- 指数 （e）
- 小数 （f）

本 LSI 可使用图 3.1 和图 3.2 所示的格式处理单精度、双精度浮点。

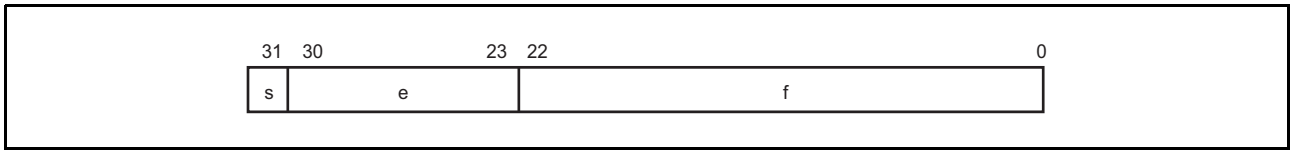


图 3.1　单精度浮点格式

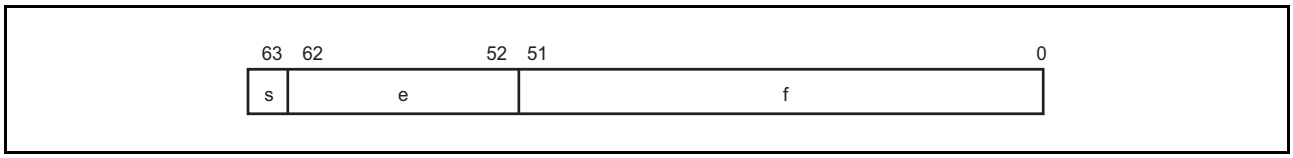


图 3.2　双精度浮点格式

指数为带偏置的格式表示如下：

e=E + bias

无偏置的指数 E 的范围是从 E_{min}-1 至 E_{max}+1。（E_{min}-1 和 E_{max}+1） 2 个值的区别如下：E_{min}-1 表示 0（正 / 负符号）和非规格化数， E_{max}+1 表示正 / 负无穷大和非数 （NaN）。 E_{min} 和 E_{max} 的值如表 3.1 所示。

表 3.1　浮点的格式和参数

参数	单精度	双精度
总位宽	32 位	64 位
符号位	1 位	1 位
指数字段	8 位	11 位
小数字段	23 位	52 位
精度	24 位	53 位
偏置	+127	+1023
E_{max}	+127	+1023
E_{min}	-126	-1022

浮点数的值 v 可由以下决定。

$E=E_{max}+1$ 且 f ≠ 0 时，与符号 s 无关 v 为非数 （NaN）。
$E=E_{max}+1$ 且 f=0 时， v 为 $(-1)^s$ （无穷大）“正 / 负无穷大”。
$E_{min} \leqslant E \leqslant E_{max}$ 时， v 为 $(-1)^s 2^E$ （1.f）“规格化数”。
$E=E_{min}-1$ 且 f ≠ 0 时， v 为 $(-1)^s 2^{Emin}$ （0.f）“非规格化数”。
$E=E_{min}-1$ 且 f=0 时、 v 为 $(-1)^s 0$ “正 / 负 0”。

16 进制中各数的范围如**表 3.2** 所示。

表 3.2　浮点的范围

类型	单精度	双精度
信令非数	H'7FFF FFFF ～ H'7FC0 0000	H'7FFF FFFF FFFF FFFF ～ H'7FF8 0000 0000 0000
静态非数	H'7FBF FFFF ～ H'7F80 0001	H'7FF7 FFFF FFFF FFFF ～ H'7FF0 0000 0000 0001
正无穷大	H'7F80 0000	H'7FF0 0000 0000 0000
正的规格化数	H'7F7F FFFF ～ H'0080 0000	H'7FEF FFFF FFFF FFFF ～ H'0010 0000 0000 0000
正的非规格化数	H'007F FFFF ～ H'0000 0001	H'000F FFFF FFFF FFFF ～ H'0000 0000 0000 0001
正 0	H'0000 0000	H'0000 0000 0000 0000
负 0	H'8000 0000	H'8000 0000 0000 0000
负的非规格化数	H'8000 0001 ～ H'807F FFFF	H'8000 0000 0000 0001 ～ H'800F FFFF FFFF FFFF
负的规格化数	H'8080 0000 ～ H'FF7F FFFF	H'8010 0000 0000 0000 ～ H'FFEF FFFF FFFF FFFF
负无穷大	H'FF80 0000	H'FFF0 0000 0000 0000
静态非数	H'FF80 0001 ～ H'FFBF FFFF	H'FFF0 0000 0000 0001 ～ H'FFF7 FFFF FFFF FFFF
信令非数	H'FFC0 0000 ～ H'FFFF FFFF	H'FFF8 0000 0000 0000 ～ H'FFFF FFFF FFFF FFFF

3.2.2 非数 （NaN）

非数 （NaN）的位模式如图 3.3 所示。以下情况时值为 NaN。

- 符号位：Don't care
- 指数字段：所有位为 1
- 小数字段：至少 1 位为 1

小数字段的 MSB 为 1 时， NaN 为信令非数 （sNaN）； 0 时， NaN 为静态非数 （qNaN）。

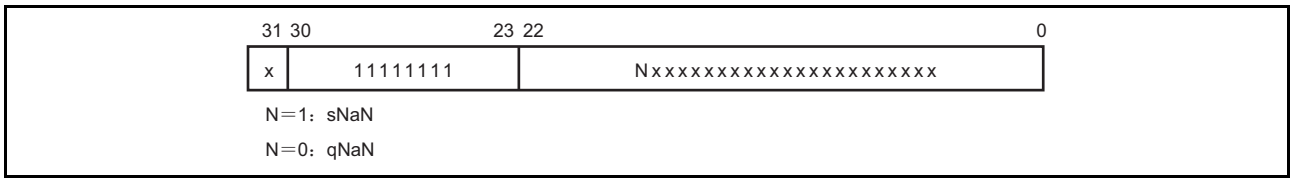


图 3.3 单精度的 NaN 位模式

在生成除复制、 FABS 和 FNEG 以外的浮点值的运算中输入 sNaN。

- FPSCR 的 EN.V 位为 0 时，运算结果 （输出）为 qNaN。
- FPSCR 的 EN.V 位为 1 时，产生无效运算异常。此时，不改变运算目标寄存器的内容。

在生成浮点值的运算中输入 qNaN，且在此运算中不输入 sNaN 时，与 FPSCR 的 EN.V 位的设定无关，总是输出 qNaN。此时不产生异常。

作为运算结果生成的 qNaN，通常为以下的值。

- 单精度 qNaN：H'7FBF FFFF
- 双精度 qNaN：H'7FF7 FFFF FFFF FFFF

输入非数 （NaN）时，浮点运算的详细情况请参照各指令的说明。

3.2.3 非规格化数

非规格化数的浮点值，指数字段作为 0，小数字段作为 0 以外的值表示。

由于在 SH2A-FPU 的状态寄存器 FPSCR 的 DN 位通常为 1，因此非规格化数 （源操作数或者运算结果）在生成 （复制、 FNEG、 FABS 以外的运算）值的浮点运算中总是被刷新为 0。

输入非规格化数时，浮点运算的详细情况请参照各指令的说明。

3.3　寄存器的说明

3.3.1　浮点寄存器

浮点寄存器的构成如图 3.4 所示。有 16 个 32 位浮点寄存器 FPR0 ～ FPR15。这 16 个寄存器参照为 FR0 ～ FR15、 DR0/2/4/6/8/l0/12/14。由 FPSCR 的 PR 位和 SZ 位决定 FPRn 与参照名的对应。请参照图 3.4。

1. 浮点寄存器：FPRi （16个寄存器）
   FPR0, FPR l, FPR2, FPR3, FPR4, FPR5, FPR6, FPR7,
   FPR8, FPR9, FPR10, FPR11, FPR12, FPR13, FPR14, FPR15
2. 单精度浮点寄存器：FRi （16个寄存器）
   FR0～FR15分配至FPR0～FPR15。
3. 双精度浮点寄存器或单精度浮点寄存器对　DRi （8个寄存器）
   DR寄存器由2个FR寄存器构成。
   DR0={FR0,FR1}、 DR2={FR2,FR3}、 DR4={FR4,FR5}、 DR6={FR6,FR7}、
   DR8={FR8,FR9}、 DR10={FR10,FR11}、 DR12={FR12,FR13}、 DR14={FR14,FR15}

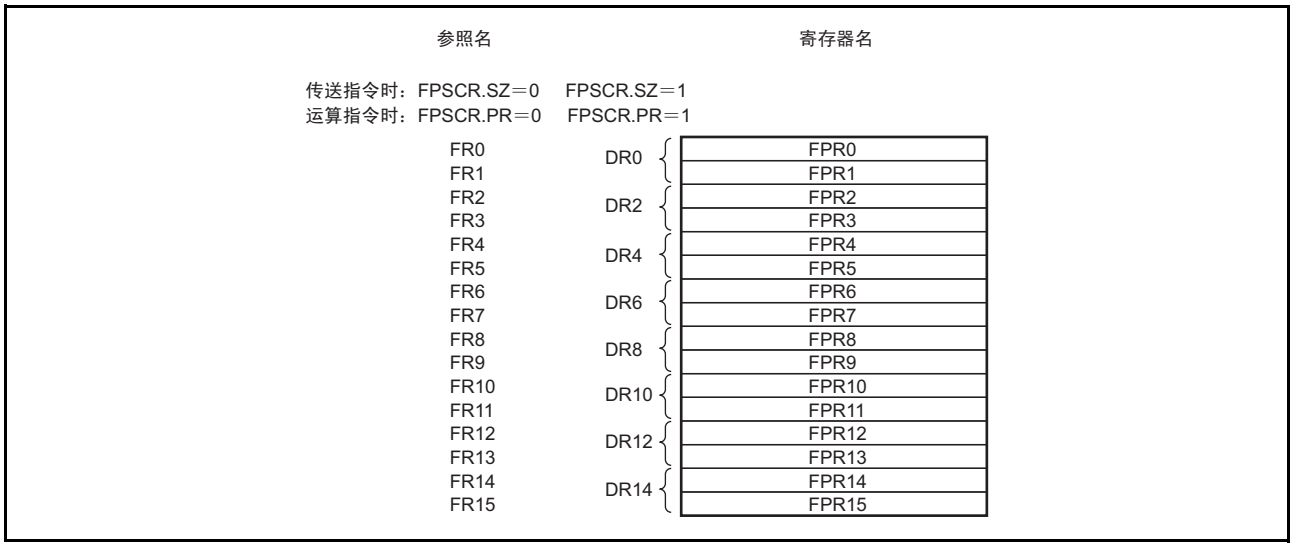


图 3.4　浮点寄存器

3.3.2　浮点状态 / 控制寄存器 （FPSCR）

FPSCR 是 32 位寄存器，选择浮点指令的控制、 FPU 异常的设定和舍入的使用方法。

位：	31	30	29	28	27	26	25	24	23	22	21	21	19	18	17	16
	-	-	-	-	-	-	-	-	-	QIS	-	SZ	PR	DN	Cause	
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W：	R	R	R	R	R	R	R	R	R	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
	Cause				Enable					Flag					RM1	RM0
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W：	R/W	R/W	R/W	R/W	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 ～ 23	—	全 0	R	保留位 读写的值总是 0。
22	QIS	0	R/W	非数处理模式 0：直接处理 qNaN 或±∞。 1：qNaN 或±∞进行与 sNaN 相同的处理 （只有在 FPSCR.Enable.V=1 时有效）。
21	—	0	R	保留位 读写的值总是 0。
20	SZ	0	R/W	传送长度模式 0：FMOV 指令的数据长度为 32 位。 1：FMOV 指令的数据长度为 32 位对 （64 位）。
19	PR	0	R/W	精度模式 0：以单精度运算执行浮点指令。 1：以双精度运算执行浮点指令。
18	DN	1	R	非规格化模式 （SH2A-FPU 中通常固定为 1） 1：非规格化数作为 0 处理。
17 ～ 12	Cause	全 0	R/W	FPU 异常源字段 FPU 异常允许字段 FPU 异常标志字段 执行 FPU 运算指令时，最初 FPU 异常源字段被清 0。接着产生 FPU 异常时，将对应 FPU 异常源字段以及 FPU 异常标志字段的位置 1。 FPU 异常标志字段保持 1 的值直到被软件清 0 为止。 各字段位的分配请参照表 3.3。
11 ～ 7	Enable	全 0	R/W	
6 ～ 2	Flag	全 0	R/W	
1 0	RM1 RM0	0 1	R/W R/W	舍入模式 选择舍入的方法。 00：就近舍入 01：朝 0 舍入 10：保留 11：保留

表 3.3　与 FPU 异常处理有关的位的分配

		FPU 错误（E）	无效运算（V）	被 0 除（Z）	上溢（O）	下溢（U）	不正确（I）
Cause	FPU 异常源字段	bit17	bit16	bit15	bit14	bit13	bit12
Enable	FPU 异常允许字段	无	bit11	bit10	bit9	bit8	bit7
Flag	FPU 异常标志字段	无	bit6	bit5	bit4	bit3	bit2

【注】 SH2A-FPU 中不发生 FPU 错误。

3.3.3　浮点通信寄存器 （FPUL）

FPU 与 CPU 之间的信息通过 FPUL 进行传递。FPUL 为 32 位的系统寄存器，通过 LDS、STS 指令从 CPU 存取。例如，将通用寄存器 R1 存储的整数转换为单精度浮点的处理流程如下所示。

R1→ （LDS 指令）→ FPUL → （单精度 FLOAT 指令） → FR1

3.4　舍入

在浮点指令中，从中间结果生成最终运算结果时执行舍入。因此，像 FMAC 这样的组合指令的结果与仅使用 FADD、 FSUB、 FMUL 等基本指令的结果不同。这是因为 FMAC 执行 1 次舍入，而 FADD、 FSUB、 FMUL 执行 2 次舍入，舍入的次数不相同。

舍入有 2 种方法，使用方法由 FPSCR 的 RM 字段决定。

FPSCR.RM[1:0]=00：就近舍入

FPSCR.RM[1:0]=01：朝 0 舍入

（1） 就近舍入

运算结果舍入至最近的可表示值。最近的可表示值为 2 个时， LSB 选择 0。

如果舍入前的值大于等于 $2^{Emax}(2-2^{-p})$ 时，结果为无穷，符号与舍入前的值相同。在此， Emax 与 p 在单精度时分别为 127 与 24，双精度时分别为 1023 与 53。

（2） 朝 0 舍入

将舍入前值的舍入位以下的位下舍入。

但舍入前的值比可表示的最大绝对值大时，变为可表示的最大绝对值。

3.5 浮点异常

3.5.1 FPU 异常源

异常源如下所示：

- FPU错误 （E） ：FPSCR的DN位为0，并且输入非规格化数时 （SH2A-FPU中不产生该错误）
- 无效运算 （V） ：无效运算 （如NaN输入）时
- 被0除 （Z） ：被除数0除
- 上溢 （O） ：运算结果上溢时
- 下溢 （U） ：运算结果下溢时
- 不正确异常 （I） ：发生上溢、下溢、舍入时

FPSCR 的 FPU 异常源字段包含对应上叙所有 E、V、Z、O、U、I 的位，FPSCR 的标志和允许字段包含对应 V、 Z、 O、 U、 I 的位，但不包含对应 E 的位。这样 FPU 错误无法禁止。

如果产生 FPU 异常，FPU 异常源字段对应的位被置 1，并给对应 FPU 异常源字段的位累加 1。不产生 FPU 异常时， FPU 异常源字段对应的位被清 0，对应 FPU 异常标志字段的位不改变。

3.5.2 FPU 异常处理

FPU 异常在以下情况时产生。

- FPU错误 （E） ：FPSCR的DN位为0，且输入非规格化数时 （SH2A-FPU中不产生该错误）
- 无效运算 （V） ：FPSCR的Enable的V位为1，且无效运算时
- 被0除 （Z） ：FPSCR的Enable的Z位为1，且被除数0除
- 上溢 （O） ：FPSCR的Enable的O位为1，且运算结果可能上溢的指令
- 下溢 （U） ：FPSCR的Enable的U位为1，且运算结果可能下溢的指令
- 不正确异常 （I） ：FPSCR的Enable的I位为1，且运算结果可能不正确的指令

在各指令的说明中有各种可能性的说明。因 FPU 运算而引发的所有异常现象都作为相同的异常现象被分配。通过读取 FPSCR 并解释其中的信息，由软件决定异常的含义。FPU 异常允许字段的 O、U、I 和 V 位的其中一个或多个位被置位时，如果 FPSCR 的 FPU 异常源字段中所有位都没有被置位，就表示不发生实际的 FPU 异常。另外，不管 FPU 异常处理如何运行，目标寄存器不改变。

除上述以外，在所有处理中对应 V、 Z、 O、 U、 I 的位置 1，生成默认值，作为运算结果。

- 无效运算 （V） ：生成结果qNaN。
- 被0除 （Z） ：生成带符号 （与舍入前的值相同）的无穷数。
- 上溢 （O） ：
  - — 朝0舍入时，生成带符号 （与舍入前的值相同）的最大规格数。
  - — 就近舍入时，生成带符号 （与舍入前的值相同）的无穷数。
- 下溢 （U） ：生成带符号 （与舍入前的值相同）的0。
- 不正确异常 （I） ：生成不正确的结果。

第 4 章 时钟脉冲振荡器 （CPG）

本 LSI 内置时钟脉冲振荡器，生成内部时钟 （Iϕ）、外围时钟 （Pϕ）和总线时钟 （Bϕ）。时钟脉冲振荡器由晶体振荡器、 PLL 电路和分频电路构成。

4.1 特点

- 4种时钟运行模式
  可根据使用的频率范围或者时钟输入类型（外接晶体振荡器、输入外部时钟、输入用于USB的时钟），从4种时钟运行模式中选择。
- 3种时钟
  能独立生成用于CPU、高速缓存的内部时钟 （Iϕ）、用于外围模块的外围时钟 （Pϕ）以及用于外部总线接口的总线时钟 （Bϕ=CKIO）。
- 频率变更功能
  能通过CPG的内部PLL （Phase Locked Loop）电路和分频电路独立更改内部时钟和外围时钟的频率。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
  可在停止睡眠模式、软件待机模式和深度待机模式中停止时钟，并可通过模块待机功能来停止特定模块。关于低功耗模式的控制，请参照 **“第32章 低功耗模式”**。

时钟脉冲振荡器的框图如图 4.1 所示。

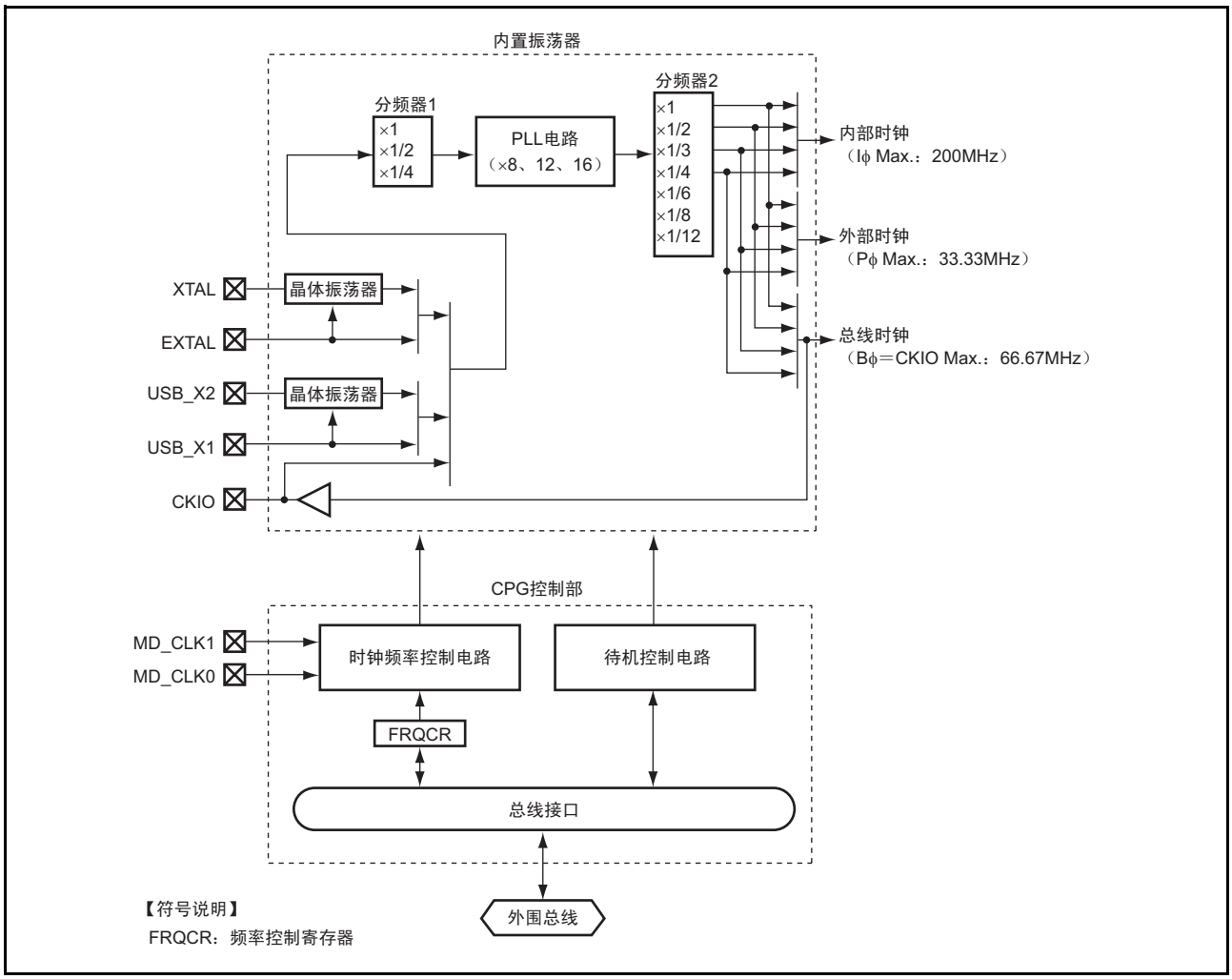


图 4.1　时钟脉冲振荡器的框图

时钟脉冲振荡器各块的功能如下：

（1） 晶体振荡器

这是将晶体振荡器连接到 XTAL、EXTAL 引脚及 USB_X1、USB_X2 引脚时的振荡电路。可根据时钟运行模式的设定使用晶体振荡器。

（2） 分频器 1

分频器 1 可对来自 EXTAL 引脚和 CKIO 引脚的时钟以及 USB_X1 输入时钟进行分频。时钟运行模式决定分频比。

（3） PLL 电路

PLL 电路可对来自晶体振荡器或者 EXTAL 引脚、CKIO 引脚的时钟及 USB_X1 输入时钟的频率递增 8 倍、12 倍、或者 16 倍。

通过频率控制寄存器设定倍频率。此时，控制内部时钟上升沿的相位使其与 CKIO 引脚的上升沿的相位相同。

时钟运行模式决定输入时钟。通过 MD_CLK0、 MD_CLK1 引脚设定时钟运行模式。关于时钟运行模式请参照**表 4.2**。

（4） 分频器 2

分频器 2 生成内部时钟或者外围时钟所使用的运行频率的时钟。在不低于 CKIO 引脚的时钟频率范围内，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、1/6 倍、1/8 倍、或 1/12 倍。通过频率控制寄存器设定分频比。

（5） 时钟频率的控制电路

时钟频率控制电路通过 MD_CLK0、 MD_CLK1 引脚及频率控制寄存器 （FRQCR）来控制时钟频率。

（6） 待机控制电路

待机控制电路控制时钟转换时或者睡眠模式、软件待机模式及深度待机模式时内部振荡电路的状态以及其他模块的状态。

另外还有控制其他模块的低功耗模式的待机控制寄存器。关于待机控制寄存器，请参照**“第 32 章 低功耗模式”**。

（7） 频率控制寄存器 （FRQCR）

频率控制寄存器 （FRQCR）中分配有以下控制位：在软件待机模式下是否从 CKIO 引脚输出时钟的控制位、 PLL 电路倍频率的控制位、内部时钟及外围时钟 （Pϕ）分频比的控制位。

4.2　输入 / 输出引脚

时钟脉冲振荡器的引脚结构和功能如表 4.1 所示。

表 4.1　振荡电路的引脚结构与功能

名称	引脚名	输入 / 输出	功能 （时钟运行模式 0、 1）	功能 （时钟运行模式 2）	功能 （时钟运行模式 3）
模式控制引脚	MD_CLK0	输入	设定时钟运行模式。		
	MD_CLK1	输入	设定时钟运行模式。		
晶体输入 / 输出引脚 （时钟输入引脚）	XTAL	输出	连接晶体振荡器。 （不使用晶体振荡器时，请将引脚置为开路）	请将引脚置为开路。	请将引脚置为开路。
	EXTAL	输入	连接晶体振荡器或者用作外部时钟的输入引脚。	请上拉引脚。	请上拉引脚。
时钟输入 / 输出引脚	CKIO	输入 / 输出	成为时钟输出引脚。	成为时钟输入引脚。	成为时钟输出引脚。
USB 用晶体 输入 / 输出引脚 （时钟输入引脚）	USB_X1	输入	作为USB专用时钟输入连接晶体振荡器，或者用作外部时钟输入引脚。不使用 USB 时，请上拉引脚。	作为USB专用时钟输入连接晶体振荡器，或者用作外部时钟输入引脚。不使用 USB 时，请上拉引脚。	作为 USB 兼本 LSI 的时钟输入连接晶体振荡器，或者用作外部时钟输入引脚。
	USB_X2	输出	连接 USB 用晶体振荡器。 （不使用晶体振荡器时，请将引脚置为开路）	连接 USB 用晶体振荡器。 （不使用晶体振荡器时，请将引脚置为开路）	连接 USB 兼本 LSI 用晶体振荡器。 （不使用晶体振荡器时，请将引脚置为开路）

4.3　时钟运行模式

模式控制引脚 （MD_CLK1、 MD_CLK0）的组合与时钟运行模式的关系如**表 4.2** 所示。
时钟运行模式可使用的频率范围如**表 4.3** 所示。

表 4.2　时钟运行模式

模式	引脚组合		时钟输入 / 输出		分频器 1	PLL 电路	CKIO 的频率
	MD_CLK1	MD_CLK0	供给源	输出			
0	0	0	EXTAL/ 晶体振荡器	CKIO	1	ON （×8、 12、 16）	（EXTAL/ 晶体） ×4
1	0	1	EXTAL/ 晶体振荡器	CKIO	1/2	ON （×8、 12、 16）	（EXTAL/ 晶体） ×2
2	1	0	CKIO	—	1/4	ON （×8、 12、 16）	（CKIO）
3	1	1	USB_X1/ 晶体振荡器	CKIO	1/4	ON （×8、 12、 16）	（USB_X1/ 晶体）

- 模式0
  在模式0中，从EXTAL引脚或者晶体振荡器输入时钟。在PLL电路根据波形形成及频率控制寄存器的设定进行倍频后，提供给本LSI。EXTAL引脚输入和晶体振荡器可使用10MHz~16.67MHz的振荡频率，CKIO的频率范围是40MHz~66.67MHz。另外，为减少消耗电流，在不使用USB时，请上拉USB_X1引脚，并将USB_X2引脚置为开路。
- 模式1
  在模式1中，从EXTAL引脚或者晶体振荡器输入时钟。在PLL电路根据波形形成及频率控制寄存器的设定进行倍频后，提供给本LSI。EXTAL引脚输入和晶体振荡器可使用20MHz~33.33MHz的振荡频率，CKIO的频率范围是40MHz~66.67MHz。另外，为减少消耗电流，在不使用USB时，请上拉USB_X1引脚，并将USB_X2引脚置为开路。
- 模式2
  在模式2中，CKIO引脚为输入，对此引脚输入外部时钟，在PLL电路根据波形形成及频率控制寄存器的设定进行倍频后，提供给本LSI。 CKIO的频率范围是40MHz~66.67MHz。另外，为减少消耗电流，在模式2中使用时，请上拉EXTAL引脚，并将XTAL引脚置为开路。并且，不使用USB时，请上拉USB_X1引脚，并将USB_X2引脚置为开路。
- 模式3
  在模式3中，可从USB_X1引脚或者晶体振荡器输入时钟。输入此引脚的外部时钟，在PLL电路根据波形形成及频率控制寄存器的设定进行倍频后，提供给本LSI。 CKIO 的频率为 （USB_X1/ 晶体）（48MHz）。另外，为减少消耗电流，在模式3中使用时，请上拉EXTAL引脚，并将XTAL引脚置为开路。并且，不使用USB晶体时，请将USB_X2引脚置为开路。

表 4.3　时钟运行模式与可设定的频率范围

时钟运行模式	FRQCR 寄存器设定值 *1	PLL 倍频率 / PLL 电路	内部时钟比（I：B：P）*2	可设定的频率范围 （MHz）				
				输入时钟 *3	输出时钟（CKIO 引脚）	内部时钟（Iϕ）	总线时钟（Bϕ）	外围时钟（Pϕ）
0	H'x003	ON （×8）	8：4：2	10 ～ 16.67	40 ～ 66.67	80 ～ 133.36	40 ～ 66.67	20 ～ 33.33
	H'x004	ON （×8）	8：4：4/3	10 ～ 16.67	40 ～ 66.67	80 ～ 133.36	40 ～ 66.67	13.33 ～ 22.22
	H'x005	ON （×8）	8：4：1	10 ～ 16.67	40 ～ 66.67	80 ～ 133.36	40 ～ 66.67	10 ～ 16.67
	H'x006	ON （×8）	8：4：2/3	10 ～ 16.67	40 ～ 66.67	80 ～ 133.36	40 ～ 66.67	6.67 ～ 11.11
	H'x104	ON （×12）	12：4：2	10 ～ 16.67	40 ～ 66.67	120 ～ 200	40 ～ 66.67	20 ～ 33.33
	H'x106	ON （×12）	12：4：1	10 ～ 16.67	40 ～ 66.67	120 ～ 200	40 ～ 66.67	10 ～ 16.67
	H'x205	ON （×16）	16：4：2	10 ～ 12.5	40 ～ 50	160 ～ 200	40 ～ 50	20 ～ 25
	H'x206	ON （×16）	16：4：4/3	10 ～ 12.5	40 ～ 50	160 ～ 200	40 ～ 50	13.33 ～ 16.67
	H'x215	ON （×16）	8：4：2	10 ～ 12.5	40 ～ 50	80 ～ 100	40 ～ 50	20 ～ 25
	H'x216	ON （×16）	8：4：4/3	10 ～ 12.5	40 ～ 50	80 ～ 100	40 ～ 50	13.33 ～ 16.67
1	H'x003	ON （×8）	4：2：1	20 ～ 33.33	40 ～ 66.67	80 ～ 133.36	40 ～ 66.67	20 ～ 33.33
	H'x004	ON （×8）	4：2：2/3	20 ～ 33.33	40 ～ 66.67	80 ～ 133.36	40 ～ 66.67	13.33 ～ 22.22
	H'x005	ON （×8）	4：2：1/2	20 ～ 33.33	40 ～ 66.67	80 ～ 133.36	40 ～ 66.67	10 ～ 16.67
	H'x006	ON （×8）	4：2：1/3	20 ～ 33.33	40 ～ 66.67	80 ～ 133.36	40 ～ 66.67	6.67 ～ 11.11
	H'x104	ON （×12）	6：2：1	20 ～ 33.33	40 ～ 66.67	120 ～ 200.0	40 ～ 66.67	20 ～ 33.33
	H'x106	ON （×12）	6：2：1/2	20 ～ 33.33	40 ～ 66.67	120 ～ 200.0	40 ～ 66.67	10 ～ 16.67
	H'x205	ON （×16）	8：2：1	20 ～ 25	40 ～ 50	160 ～ 200	40 ～ 50	20 ～ 25
	H'x206	ON （×16）	8：2：2/3	20 ～ 25	40 ～ 50	160 ～ 200	40 ～ 50	13.33 ～ 16.67
	H'x215	ON （×16）	4：2：1	20 ～ 25	40 ～ 50	80 ～ 100	40 ～ 50	20 ～ 25
	H'x216	ON （×16）	4：2：2/3	20 ～ 25	40 ～ 50	80 ～ 100	40 ～ 50	13.33 ～ 16.67
2	H'x003	ON （×8）	2：1：1/2	40 ～ 66.67	—	80 ～ 133.36	40 ～ 66.67	20 ～ 33.33
	H'x004	ON （×8）	2：1：1/3	40 ～ 66.67	—	80 ～ 133.36	40 ～ 66.67	13.33 ～ 22.22
	H'x005	ON （×8）	2：1：1/4	40 ～ 66.67	—	80 ～ 133.36	40 ～ 66.67	10 ～ 16.67
	H'x006	ON （×8）	2：1：1/6	40 ～ 66.67	—	80 ～ 133.36	40 ～ 66.67	6.67 ～ 11.11
	H'x104	ON （×12）	3：1：1/2	40 ～ 66.67	—	120 ～ 200.0	40 ～ 66.67	20 ～ 33.33
	H'x106	ON （×12）	3：1：1/4	40 ～ 66.67	—	120 ～ 200.0	40 ～ 66.67	10 ～ 16.67
	H'x205	ON （×16）	4：1：1/2	40 ～ 50	—	160 ～ 200	40 ～ 50	20 ～ 25
	H'x206	ON （×16）	4：1：1/3	40 ～ 50	—	160 ～ 200	40 ～ 50	13.33 ～ 16.67
	H'x215	ON （×16）	2：1：1/2	40 ～ 50	—	80 ～ 100	40 ～ 50	20 ～ 25
	H'x216	ON （×16）	2：1：1/3	40 ～ 50	—	80 ～ 100	40 ～ 50	13.33 ～ 16.67

时钟运行模式	FRQCR 寄存器设定值 *1	PLL 倍频率 PLL 电路	内部时钟比（I：B：P）*2	可设定的频率范围 （MHz） 输入时钟 *3	输出时钟（CKIO 引脚）	内部时钟（Iϕ）	总线时钟（Bϕ）	外围时钟（Pϕ）
3	H'x003	ON （×8）	2：1：1/2	48	48	96	48	24
	H'x004	ON （×8）	2：1：1/3	48	48	96	48	16
	H'x005	ON （×8）	2：1：1/4	48	48	96	48	12
	H'x006	ON （×8）	2：1：1/6	48	48	96	48	8
	H'x104	ON （×12）	3：1：1/2	48	48	144	48	24
	H'x106	ON （×12）	3：1：1/4	48	48	144	48	12
	H'x205	ON （×16）	4：1：1/2	48	48	192	48	24
	H'x206	ON （×16）	4：1：1/3	48	48	192	48	16
	H'x215	ON （×16）	2：1：1/2	48	48	96	48	24
	H'x216	ON （×16）	2：1：1/3	48	48	96	48	16

【注】 *1 FRQCR 寄存器设定值 x 由 bit12、 13 的设定值决定。

*2 是假设输入时钟频率为 1 时的时钟比。

*3 模式 0、 1 时为 EXTAL 引脚的时钟输入或者晶体振荡器的频率。
模式2时为CKIO引脚的时钟输入频率。
模式3时为USB_X1引脚的时钟输入或者晶体振荡器的频率。

【注意事项】

1. 模式0时，内部时钟的频率=EXTAL引脚的频率×PLL电路的倍频率×分频器1的分频比。模式1时，频率=（EXTAL引脚的频率×1/2）×PLL电路的倍频率×分频器1的分频比。模式2时，频率=（CKIO引脚的频率×1/4）×PLL电路的倍频率×分频器1的分频比。模式3时，频率=（USB_X1引脚的频率×1/4）×PLL电路的倍频率×分频器1的分频比。内部时钟频率不能低于CKIO引脚的频率。
2. 模式0时，外围时钟的频率=EXTAL引脚的频率×PLL电路的倍频率×分频器1的分频比。模式1时，频率=（EXTAL引脚的频率×1/2）×PLL电路的倍频率×分频器1的分频比。模式2时，频率=（CKIO引脚的频率×1/4）×PLL电路的倍频率×分频器1的分频比。模式3时，频率=（USB_X1引脚的频率×1/4）×PLL电路的倍频率×分频器1的分频比。
外围时钟频率请设定在小于等于33.33MHz的范围内。同时不能高于CKIO引脚的频率。
3. PLL电路的倍频率可选择×8、×12或者×16。分频器1的分频比可选择×1、×1/2、×1/3、×1/4、×1/6、×1/8或者×1/12。以上由各自的频率控制寄存器 （FRQCR）设定。
4. 模式0时， PLL电路的输出频率=EXTAL引脚的频率¥PLL电路的倍频率。模式1时，频率=（EXTAL引脚的频率×1/2）×PLL电路的倍频率。模式2、模式3时，频率=（CKIO引脚或者USB_X1引脚的频率×1/4）×PLL电路的倍频率。此频率请设定在小于等于200MHz的范围内。

4.4　寄存器说明

时钟脉冲振荡器有以下寄存器。

表 4.4　寄存器结构

寄存器名	简称	R/W	初始值	地址	存取长度
频率控制寄存器	FRQCR	R/W	H'0003	H'FFFE0010	16

4.4.1　频率控制寄存器 （FRQCR）

FRQCR 是 16 位可读写寄存器，可指定通常时、总线权释放时、软件待机模式时和待机解除时是否从 CKIO 引脚输出时钟、 PLL 电路的倍频率、内部时钟以及外围时钟 （Pϕ）的分频比。 FRQCR 只能进行字存取。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	CKOEN2	CKOEN[1:0]		-	-	STC[1:0]		-	-	-	IFC	-	PFC[2:0]		
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W：	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R	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	CKOEN2	0	R/W	时钟输出允许 2 CKOEN2 指定 PLL 电路的倍频率变更时是从 CKIO 引脚输出时钟还是将 CKIO 引脚固定为低电平。 设为 1 时，在 PLL 电路的倍频率变更时，CKIO 引脚被固定为低电平。据此可防止 PLL 电路的倍频率变更时，由于 CKIO 时钟不稳定而导致的外部电路误动作。时钟运行模式 2 时，与此位的值无关， CKIO 引脚为输入。 0：输出时钟 1：输出低电平
13、 12	CKOEN[1:0]	00	R/W	时钟输出允许 CKOEN[1:0] 指定通常时、总线释放时、待机模式时和待机解除时是从 CKIO 引脚输出时钟，还是固定 CKIO 引脚电平，还是设置为高阻抗状态（Hi-Z）。 设定为 01 时，在待机模式时及待机解除期间， CKIO 引脚被固定为低电平。据此可防止待机退解除时，由于 CKIO 时钟不稳定而导致的外部电路错误运行。时钟工作模式 2 时，与此位的值无关， CKIO 引脚为输入。深度待机模式时，保持通常时的状态。 通常时　　总线权释放时　　待机模式时 00：输出　　输出 off （Hi-Z）　　输出 off （Hi-Z） 01：输出　　输出　　输出低电平 10：输出　　输出　　输出 （输出不稳定时钟） 11：输出 off （Hi-Z）　输出 off （Hi-Z）　输出 off （Hi-Z）
11、 10	—	全 0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说明
9、8	STC[1:0]	00	R/W	PLL 电路的倍频率 00：×8 倍 01：×12 倍 10：×16 倍 11：保留 （禁止设定）
7 ～ 5	—	全 0	R	保留位 读写的值总是 0。
4	IFC	0	R/W	内部时钟频率的分频比 指定与 PLL 电路的输出频率相关的内部时钟的分频比。 0：×1 倍 1：×1/2 倍
3	—	0	R	保留位 读写的值总是 0。
2 ～ 0	PFC[2:0]	011	R/W	外围时钟频率的分频比 指定与 PLL 电路的输出频率相关的内部时钟的分频比。 000：保留 （禁止设定） 001：保留 （禁止设定） 010：保留 （禁止设定） 011：×1/4 倍 100：×1/6 倍 101：×1/8 倍 110：×1/12 倍

4.5　频率的变更方法

可通过改变 PLL 电路的倍频率和分频器的分频比来改变内部时钟 （Iϕ）及外围时钟 （Pϕ）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：

4.5.1　倍频率的变更

变更 PLL 电路的倍频率时，需要 PLL 稳定时间。通过内部 WDT 对稳定时间进行计数。

1. 在初始状态下， PLL 电路的倍频率为 8。
2. 给 WDT 设定一个将达到指定振荡稳定时间的值，停止 WDT。需要进行如下设定。
   WTCSR.TME=0：WDT 停止
   WTCSR.CKS[2:0]：WDT 计数时钟的分频比
   WTCNT：计数器的初始值
   （使用设定后的时钟进行 WDT 的递增计数。）
3. 给 STC[1:0] 设定目标值。同时也可对 IFC、 PFC[2:0] 设定分频比。
4. 本 LSI 内部暂时停止，开始 WDT 的递增计数。停止内部时钟和外围时钟，只向 WDT 供给时钟。同时在 CKIO 引脚持续输出时钟。本状态与软件待机模式状态相同，可根据模块的不同进行寄存器的初始化。详细请参照 “34.3 各运行模式下的寄存器状态一览”。
5. 开始提供设定在 WDT 计数上溢时的时钟，本 LSI 重新开始运行。 WDT 在上溢后停止。

4.5.2　分频比的变更

只变更分频比，而不同时进行倍频率的变更时，不进行 WDT 的计数。

1. 初始状态下， IFC=B'0， PFC[2:0]=B'011。
2. 给 IFC、PFC[2:0] 位设定目标值。由于时钟模式及 PLL 电路倍频率的关系，可设定的值有限。注意设定错误值时本 LSI 会发生误动作。
3. 设定寄存器的各位 （IFC、 PFC[2:0]）时，切换为被设定的时钟。

【注】 1. 在频率变更后执行 SLEEP 指令时，读取 3 次频率控制寄存器 （FRQCR）后执行 SLEEP 指令。
2. 当 PLL 电路的倍频率变更时及软件待机模式解除后的振荡稳定时间内，在时钟模式 0、 1、 3，输出不稳定的 CKIO。为防止由此导致的误动作，请变更 FRQCR 寄存器的 bit14、 13、 12。

4.6　电路板设计的注意事项

4.6.1　外部时钟输入时的注意事项

外部时钟输入的连接示例如图 4.2 所示。 XTAL 引脚为开路状态时，寄生电容必须小于等于 10pF。即使在输入外部时钟的情况下，为确保 PLL 的稳定时间，在通电时或解除待机时，等待时间必须大于等于振荡稳定时间。

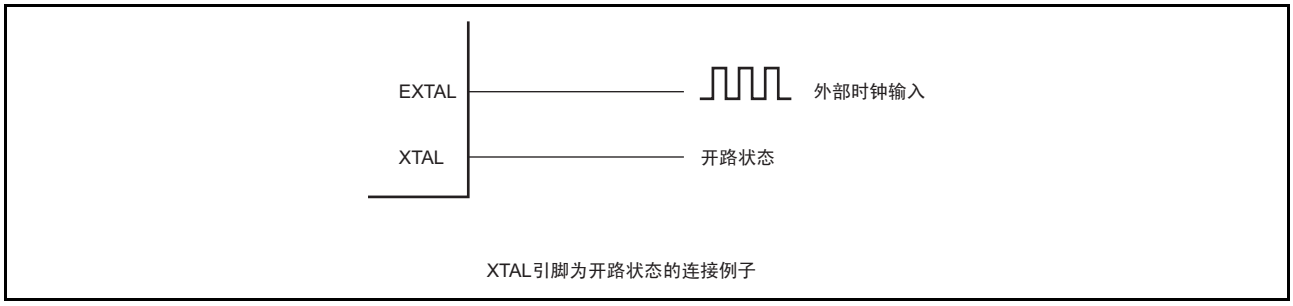


图 4.2　外部时钟的连接例

4.6.2　晶体振荡器使用时的注意事项

晶体振荡器与电容 CL1、 CL2，请尽量置于 XTAL 引脚和 EXTAL 引脚附近。为避免感应，并进行正确振荡，需与振荡器附加电容的接地点相同，并且不可在这些器件附近放置布线图形。

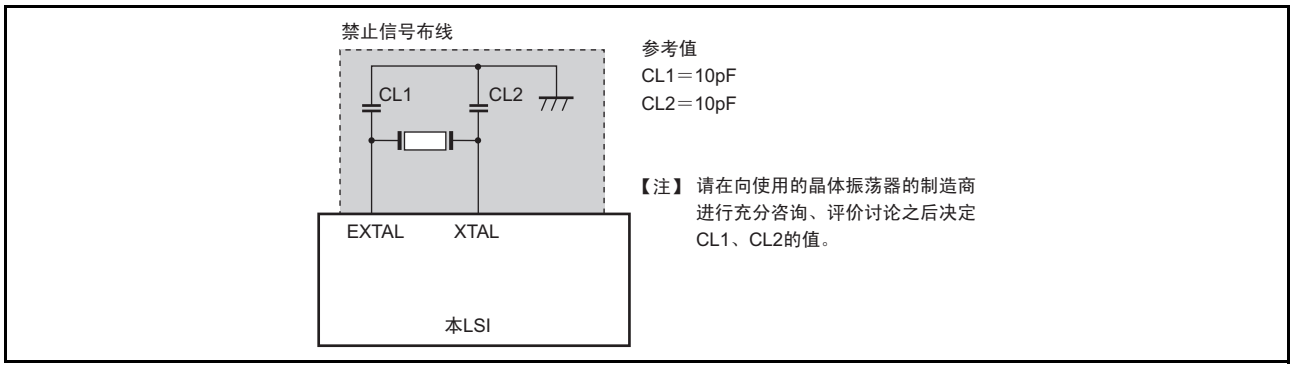


图 4.3　晶体振荡器使用时的注意事项

4.6.3　有关振荡器的注意事项

因为有关振荡器的各种特性密切关系到用户的电路板设计，所以希望用户参考本章介绍的振荡器的连接例子，在经过充分评价后使用。振荡器的电路额定值因振荡器以及安装电路的寄生电容等而不同，所以请与振荡器制造商充分商谈后再作决定。振荡引脚的外加电压不可超过最大额定值。

4.6.4　PLL 振荡电路使用时的注意事项

在用于 PLL 的 $PLLV_{CC}$ 与 $PLLV_{SS}$ 的连接图形中，请减少从电路板电源供给引脚的布线长度、加宽图形宽度并减少感应系数。

时钟运行模式 2、 3 时，请上拉 EXTAL 引脚并将 XTAL 引脚置为开路。

由于 PLL 的模拟电源对噪声等敏感，因此可能会出现因其他电源的干扰导致全系统产生误动作的现象。所以，绝对不能在电路板上向本模拟电源与 V_{CC}、 PV_{CC} 的数字电源供给同一电源。

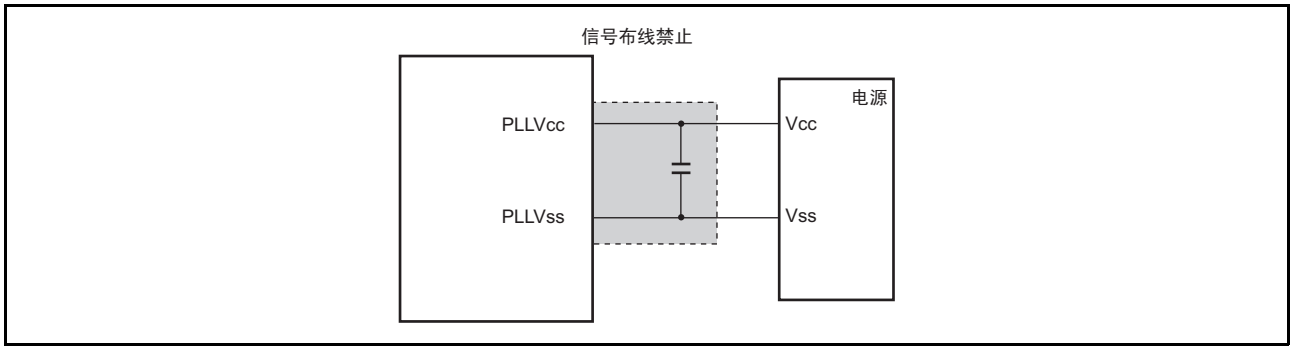


图 4.4　PLL 振荡电路使用时的注意事项

4.7　使用时的注意事项

如果在时钟模式 0、 1、 3 中使用本 LSI， $\overline{RES}$ 引脚取反后， CKIO 输出 1 个周期的不稳定时钟。

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、寄存器组错误、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

种类	异常处理		优先级
复位	上电复位		高
	手动复位		
地址错误	CPU 地址错误		
	DMAC 地址错误		
指令	FPU 异常		
	整数除法异常 （被 0 除）		
	整数除法异常 （上溢）		
寄存器组错误	存储体下溢		
	存储体上溢		
中断	NMI		
	用户断点		
	H-UDI		
	IRQ		
	PINT		
	内置外围模块	直接存储器存取控制器 （DMAC）	
		USB2.0 主 / 功能模块 （USB）	
		LCD 控制器 （LCDC）	
		比较匹配定时器 （CMT）	
		总线状态控制器 （BSC）	
		看门狗定时器 （WDT）	
		多功能定时器脉冲单元 2 （MTU2）	
		A/D 转换器 （ADC）	
		I^2C 总线接口 3 （IIC3）	
		带 FIFO 的串行通信接口 （SCIF）	
		同步串行通信单元 （SSU）	
		串行音频接口 （SSI）	
		CD-ROM 解码器 （ROM-DEC）	
		AND/NAND 闪存控制器 （FLCTL）	
		SD 主接口 （SDHI）	
		实时时钟 （RTC）	
		控制器局域网 （RCAN-TL1）	
		取样速率转换器 （SRC）	
		IEBus™ 控制器 （IEB）	
指令	陷阱指令 （TRAPA 指令）		
	一般非法指令 （未定义代码）		
	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 （FPU 模块待机时，包含 FPU 指令和与 FPU 有关的 CPU 指令）、 PC 改写指令 *2、 32 位指令 *3、 RESBANK 指令、 DIVS 指令或 DIVU 指令）		低

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

*2　PC 改写指令：JMP、JSR、BRA、BSR、RTS、RTE、BT、BF、TRAPA、BF/S、BT/S、BSRF、BRAF、JSR/N、 RTV/N

*3　32 位指令：BAND.B、 BANDNOT.B、 BCLR.B、 BLD.B、 BLDNOT.B、 BOR.B、 BORNOT.B、 BSET.B、BST.B、 BXOR.B、 MOV.B@disp12、 MOV.W@disp12、 MOV.L@disp12、 MOVI20、 MOVI20S、MOVU.B、 MOVU.W

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		源检测和异常处理时序
复位	上电复位	在 $\overline{\text{RES}}$ 引脚从低电平变为高电平，H-UDI 复位断言命令置位后，设置 H-UDI 复位取反命令或者 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		在指令的解码时被检测，并在执行中的指令结束后，开始处理。
中断		
寄存器组错误	存储体下溢	在没有执行向寄存器组的保存时，如果打算执行 RESBANK 指令，就开始处理
	存储体上溢	通过中断控制器设定为接受寄存器组上溢异常 （INTC 的 IBNR 的 BOVE 位 =1），产生使用寄存器组的中断且被 CPU 接受，以及已经保存到寄存器组的全部区域时，开始处理。
指令	陷阱指令	通过执行 TRAPA 指令开始处理。
	一般非法指令	除了紧接延迟转移指令 （延迟槽）以外的未定义代码 （包含 FPU 模块待机时的 FPU 指令以及与 FPU 有关的 CPU 指令）被解码时，开始处理。
	槽非法指令	当分配在紧接延迟转移指令 （延迟槽）之后的未定义代码 （包含 FPU 模块待机时的 FPU 指令以及与 FPU 有关的 CPU 指令）、PC 改写指令、32 位指令、RESBANK 指令、 DIVS 指令或 DIVU 指令被解码时，开始处理。
	整数除法指令	检测出被零除异常或者因用 –1 除负数的最大值 (H'80000000) 引起的上溢异常时，开始处理。
	浮点运算指令	通过浮点运算指令的无效运算异常 （IEEE754 规则）、被零除异常、上溢、下溢或不正确异常，开始处理。另外，FPSCR 的 QIS 位被置位时，如果将 qNaN 或±∞输入浮点运算指令源，就开始处理。

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，设定向量基址寄存器（VBR）为 H'00000000，状态寄存器（SR）的中断屏蔽位 （I3 ～ I0）为 H'F （B'1111），并将 BO 位和 CS 位初始化为 0，中断控制器 （INTC）的 IBNR 的 BN 位初始化为 0。而且上电复位时，将 FPSCR 初始化为 H'00040001，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、寄存器组错误、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 NMI 以及用户断点以外的中断异常处理中，执行使用寄存器组的设定时，将通用寄存器 R0 ～ R14、控制寄存器 GBR、系统寄存器 MACH、MACL、PR 和执行的中断异常处理的向量表地址偏移量保存到寄存器组。在由地址错误、寄存器组错误、NMI 中断、用户断点中断、指令引起的异常处理时，不执行向寄存器组的保存。另外，如果已进行向寄存器组中全部存储体的保存，就替代寄存器组自动保存到堆栈。此时，需要在中断控制器中设定不接受寄存器组上溢异常（INTC 的 IBNR 的 BOVE 位 =0）。如果设定接受寄存器组上溢异常 （INTC 的 IBNR 的 BOVE 位 =1），就会产生寄存器组上溢异常。在中断异常处理时，将中断优先级写到 SR 的 I3 ～ I0 位。在由地址错误或者指令引起异常处理时，I3 ～ I0 位不受影响。从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。

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 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
FPU 异常		13	H'00000034 ～ H'00000037
H-UDI		14	H'00000038 ～ H'0000003B
存储体上溢		15	H'0000003C ～ H'0000003F
存储体下溢		16	H'00000040 ～ H'00000043
整数除法异常 （被 0 除）		17	H'00000044 ～ H'00000047
整数除法异常 （上溢）		18	H'00000048 ～ H'0000004B
（系统保留）		19 ； 31	H'0000004C ～ H'0000004F ： H'0000007C ～ H'0000007F
陷阱指令 （用户向量）		32 ； 63	H'00000080 ～ H'00000083 ： H'000000FC ～ H'000000FF
外部中断 （IRQ、 PINT）、内置外围模块 *		64 ； 511	H'00000100 ～ H'00000103 ： H'000007FC ～ H'000007FF

【注】 * 有关外部中断、各内置外围模块的中断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 6.4。

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、寄存器组错误、中断、指令	向量表地址 =VBR+（向量表地址偏移量） =VBR+（向量号）×4

【注】 1. 向量表地址偏移量：参照表 5.3
2. 向量号：参照表 5.3

5.2　复位

5.2.1　输入 / 输出引脚

与复位相关的引脚构成如表 5.5 所示。

表 5.5　引脚构成

名称	引脚名	输入 / 输出	功　　能
上电复位	$\overline{\text{RES}}$	输入	通过向引脚输入低电平，转换为上电复位处理。
手动复位	$\overline{\text{MRES}}$	输入	通过向引脚输入低电平，转换为手动复位处理。

5.2.2　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 5.6 所示，通过上电复位或者手动复位，都会初始化 CPU 状态。FPU 状态在上电复位时被初始化，而在手动复位时不被初始化。另外，内置外围模块的寄存器（除去少部分寄存器）在上电复位时被初始化，而在手动复位时不被初始化。

表 5.6　复位状态

种类	转移到复位状态的条件				内部状态	
	$\overline{\text{RES}}$	H-UDI 命令	$\overline{\text{MRES}}$	WDT 上溢	CPU	CPU 以外的模块
上电复位	低电平	—	—	—	初始化	初始化
	高电平	设置 H-UDI 复位断言命令	—	—	初始化	初始化
	高电平	设置 H-UDI 复位断言以外的命令	—	上电	初始化	*
手动复位	高电平	设置 H-UDI 复位断言以外的命令	低电平	—	初始化	*
	高电平	设置 H-UDI 复位断言以外的命令	高电平	手动	初始化	*

【注】 * 请参照“34.3 各运行模式下的寄存器状态一览”。

5.2.3 上电复位

（1） 由 RES 引脚引起的上电复位

将 RES 引脚置为低电平时，本 LSI 就进入上电复位状态。为了确保本 LSI 的复位，RES 引脚的低电平在通电或者软件待机时 （时钟停止时），必须至少保持振荡稳定时间，而在时钟运行时，必须至少保持 20tcyc。在上电复位的状态下，CPU 的内部状态和内置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 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）的中断屏蔽级位 （I3～I0）初始化为 H'F （B'1111）， BO 位和 CS 位初始化为 0。另外， INTC 的 IBNR 的 BN 位初始化为 0， FPSCR 初始化为 H'00040001。
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 H-UDI 复位断言命令引起的上电复位

设置 H-UDI 复位断言命令时，就进入上电复位状态。 H-UDI 复位断言命令与由 RES 引脚引起的上电复位相同。通过设置 H-UDI 复位取反命令，可解除上电复位状态。 H-UDI 复位断言命令与 H-UDI 复位取反命令之间所需的时间，与为了开始上电复位保持 RES 引脚为低电平的时间相同。在通过 H-UDI 复位断言命令进行上电复位的状态下，如果设置 H-UDI 复位取反命令，就开始上电复位异常处理。此时 CPU 的运行与由 RES 引脚引起的上电复位时相同。

（3） 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产生上电复位，就在 WDT 的 WTCNT 发生上溢时进入上电复位状态。

此时，由 WDT 引起的复位信号不初始化 WDT 的 WRCSR 以及 CPG 的 FRQCR。

另外，如果同时产生由 RES 引脚、H-UDI 复位断言命令引起的复位和由 WDT 上溢引起的复位，就优先进行由 RES 引脚、H-UDI 复位断言命令引起的复位，并将 WRCSR 的 WOVF 位清 0。当开始由 WDT 引起的上电复位异常处理时， CPU 的运行与通过 RES 引脚引起的上电复位时相同。

5.2.4 手动复位

（1） 由 $\overline{\text{MRES}}$ 引脚引起的手动复位

将 $\overline{\text{MRES}}$ 引脚置为低电平时，本 LSI 就进入手动复位状态。为了确保本 LSI 的复位， $\overline{\text{MRES}}$ 引脚的低电平必须至少保持 20tcyc。在手动复位状态下， CPU 的内部状态被初始化，而内置外围模块的寄存器不被初始化。在手动复位状态下，$\overline{\text{MRES}}$ 引脚保持一定期间的低电平后变为高电平时，开始手动复位异常处理。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）的中断屏蔽级位 （I3 ～ I0）初始化为 H'F （B'1111）， BO 位和 CS 位初始化为 0。另外， INTC 的 IBNR 的 BN 位初始化为 0。
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。

（2） 由 WDT 引起的手动复位

如果设定为在 WDT 看门狗定时器模式中产生手动复位，就在 WDT 的 WTCNT 发生上溢时进入手动复位状态。

开始由 WDT 引起的手动复位异常处理时的 CPU 运行与由 $\overline{\text{MRES}}$ 引脚引起的手动复位时相同。

产生手动复位时，保持总线周期。在总线权释放或 DMAC 突发传送过程中产生手动复位时，到 CPU 获得总线权为止，保留手动复位异常处理。通过手动复位初始化 CPU 和 INTC 的 IBNR 的 BN 位。 FPU 和其他模块不被初始化。

5.3 地址错误

5.3.1 地址错误发生源

如表 5.7 总线周期和地址错误所示，在取指令或者读写数据时发生地址错误。

表 5.7 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
		从内置外围模块空间 *、内置 RAM 空间 * 的 H'F0000000 ～ H'F5FFFFFF 以外取命令	无 （正常）
		从内置外围模块空间 *、内置 RAM 空间 * 的 H'F0000000 ～ H'F5FFFFFF 取命令	发生地址错误
读写数据	CPU 或 DMAC	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
		在内置外围模块空间 * 内存取字数据和字节数据	无 （正常）
		在 16 位内置外围模块空间 * 存取长字数据	无 （正常）
		在 8 位内置外围模块空间 * 存取长字数据	无 （正常）

【注】 * 关于内置外围模块空间和内置 RAM 空间的情况，请参照 “第 9 章 总线状态控制器 （BSC）”。

5.3.2 地址错误异常处理

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. 从异常处理向量表中取出对应发生地址错误的异常服务程序的起始地址。
2. 将状态寄存器 （SR）保存到堆栈。
3. 将程序计数器 （PC）保存到堆栈。保存的PC值是已执行的最后一条命令的下一条命令的起始地址。
4. 转移到从异常处理向量表中取出的异常服务程序的起始地址，并开始执行程序。此时的转移不是延迟转移。

5.4 寄存器组错误

5.4.1 寄存器组错误发生源

（1） 存储体上溢

通过中断控制器设定为接受寄存器组上溢异常 （INTC 的 IBNR 的 BOVE 位 =1），产生使用寄存器组的中断且被 CPU 接受，以及已经保存到寄存器组的全部区域时。

（2） 存储体下溢

在没有执行向寄存器组的保存时，打算执行 RESBANK 命令的情况下。

5.4.2 寄存器组错误异常处理

发生寄存器组错误时，发生寄存器组错误异常处理。此时 CPU 的运行如下：

1. 从异常处理向量表中取出对应发生寄存器组错误的异常服务程序的起始地址。
2. 将状态寄存器 （SR）保存到堆栈。
3. 将程序计数器 （PC）保存到堆栈。当存储体上溢时，保存的 PC 值是已执行的最后一条命令的下一个指令的起始地址，在存储体下溢时，保存的PC值是已执行的RESBANK命令的起始地址。
   为了防止存储体上溢时产生多重中断，将成为存储体上溢源的中断级写入状态寄存器（SR）的中断屏蔽级位 （I3～I0）。
4. 转移到从异常处理向量表中取出的异常服务程序的起始地址，并开始执行程序。此时的转移不是延迟转移。

5.5 中断

5.5.1 中断源

如表 5.8 所示，启动异常处理的中断源有 NMI、用户断点、 H-UDI、 IRQ、 PINT 和内置外围模块。

表 5.8 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
H-UDI	用户调试接口 （H-UDI）	1
IRQ	IRQ0 ～ IRQ7 引脚 （来自外部的输入）	8
PINT	PINT0 ～ PINT7 引脚 （来自外部的输入）	8
内置外围模块	直接存储器存取控制器 （DMAC）	16
	USB2.0 主 / 功能模块 （USB）	1
	LCD 控制器 （LCDC）	1
	比较匹配定时器 （CMT）	2
	总线状态控制器 （BSC）	1
	看门狗定时器 （WDT）	1
	多功能定时器脉冲单元 2 （MTU2）	25
	A/D 转换器 （ADC）	1
	I^2C 总线接口 3 （IIC3）	20
	带 FIFO 的串行通信接口 （SCIF）	16
	同步串行通信单元 （SSU）	6
	串行音频接口 （SSI）	4
	CD-ROM 解码器 （ROM-DEC）	5
	AND/NAND 闪存控制器 （FLCTL）	4
	SD 主接口 （SDHI）	3
	实时时钟 （RTC）	3
	控制器局域网 （RCAN-TL1）	10
	取样速率转换器 （SRC）	3
	IEBus™ 控制器 （IEB）	1

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 6.4。

5.5.2　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。用户断点中断和 H-UDI 的优先级为 15。能通过 INTC 的中断优先级设定寄存器 01、02、05 ～ 17（IPR01、IPR02、IPR05 ～ IPR17）自由设定 IRQ 中断、PINT 中断和内置外围模块中断的优先级（表 5.9）。能设定的优先级为 0 ～ 15，不能设定优先级 16。有关 IPR01、IPR02、IPR05 ～ IPR17，请参照“6.3.1 中断优先级设定寄存器 01、 02、 05 ～ 17 （IPR01、 IPR02、 IPR05 ～ IPR17）”。

表 5.9　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
H-UDI	15	固定优先级
IRQ	0 ～ 15	通过中断优先级设定寄存器 01、 02、 05 ～ 17（IPR01、 IPR02、 IPR05 ～ IPR17）来设定
PINT		
内置外围模块		

5.5.3 中断异常处理

发生中断时，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）的中断屏蔽级位 （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 从异常处理向量表中取出与被接受的中断相对应的异常服务程序的起始地址，将 SR 和程序计数器 （PC）保存到堆栈。在 NMI、用户断点以外的中断异常处理中，如果进行使用寄存器组的设定，就将通用寄存器 R0 ～ R14、控制寄存器 GBR、系统寄存器 MACH、MACL、PR 和被执行的异常处理向量表地址偏移量保存到寄存器组。在由地址错误、NMI 中断、用户断点中断及指令引起的异常处理中，不进行向寄存器组的保存。另外，已进行向寄存器组所有存储体 （0 ～ 14）的保存时，就代替寄存器组自动保存至堆栈。此时，需要在中断控制器中设定不接受寄存器组上溢异常（INTC 的 IBNR 的 BOVE 位 =0）。设定为接受寄存器组上溢异常时 （INTC 的 IBNR 的 BOVE 位 =1），就发生寄存器组上溢异常。然后将接受的中断的优先级值写入 SR 的 I3 ～ I0 位。但为 NMI 时优先级为 16，而 I3 ～ I0 位设定的值为 H'F （优先级 15）。之后，转移到从异常处理向量表中取出的中断异常服务程序的起始地址，并开始执行程序。此时的转移不是延迟转移。有关中断异常处理的详细情况请参照 “6.6 运行说明”。

5.6 由指令引起的异常

5.6.1 由指令引起的异常种类

如表 5.10 所示，启动异常处理的指令有陷阱指令、槽非法指令、一般非法指令及整数除法运算异常。

表 5.10 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 （在 FPU 模块待机时，含 FPU 指令和与 FPU 有关的 CPU 指令）、 PC 改写指令、 32 位指令、RESBANK 指令、DIVS 指令或者 DIVU 指令	延迟转移指令：JMP、JSR、BRA、BSR、RTS、RTE、 BF/S、 BT/S、 BSRF、 BRAF PC 改写指令：JMP、JSR、BRA、BSR、RTS、RTE、BT、BF、TRAPA、BF/S、BT/S、BSRF、BRAF、 JSR/N、 RTV/N 32 位指令：BAND.B、 BANDNOT.B、 BCLR.B、BLD.B、 BLDNOT.B、 BOR.B、 BORNOT.B、BSET.B、 BST.B、 BXOR.B、 MOV.B@disp12、MOV.W@disp12、 MOV.L@disp12、 MOVI20、MOVI20S、 MOVU.B、 MOVU.W
一般非法指令	在延迟槽以外的未定义代码 （在 FPU 模块待机时，含 FPU 指令和与 FPU 有关的 CPU 指令）	
整数除法异常	被 0 除	DIVU、 DIVS
	负的最大值÷ （–1）	DIVS
浮点运算指令	引起 IEEE754 规格中定义的无效运算异常或者被零除异常的指令，以及可能引起上溢、下溢或不正确异常的指令。	FADD、FSUB、FMUL、FDIV、FMAC、FCMP/EQ、 FCMP/GT、 FLOAT、 FTRC、 FCNVDS、FCNVSD、 FSQRT

5.6.2 陷阱指令

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

1. 从异常处理向量表中取出与TRAPA指令指定的向量号相对应的异常服务程序的起始地址。
2. 将状态寄存器 （SR）保存到堆栈。
3. 将程序计数器 （PC）保存到堆栈。保存的PC值是TRAPA指令的下一条指令的起始地址。
4. 转移到从异常处理向量表中取出的异常服务程序的起始地址，并开始执行程序。此时的转移不是延迟转移。

5.6.3 槽非法指令

将分配在紧接着延迟转移指令之后的指令称为 “分配给延迟槽的指令”。分配给延迟槽的指令为未定义代码（含在 FPU 模块待机时的 FPU 指令以及与 FPU 有关的 CPU 指令）、PC 改写指令、32 位指令、RESBANK 指令、DIVS 指令或 DIVU 指令时，如果这些指令被解码，就开始槽非法指令异常处理。而且，FPU 为模块待机状态时，浮点指令或与 FPU 有关的 CPU 指令作为未定义代码处理。分配给延迟槽时，如果该指令被解码，就开始槽非法指令异常处理。

进行槽非法指令异常处理时， CPU 的运行如下：

1. 从异常处理向量表中取出异常服务程序的起始地址。
2. 将状态寄存器 （SR）保存到堆栈。
3. 将程序计数器 （PC）保存到堆栈。保存的PC值为未定义代码、PC改写指令、32位指令、RESBANK指令、 DIVS指令或DIVU指令之前的延迟转移指令的转移地址。
4. 转移到从异常处理向量表中取出的异常服务程序的起始地址，并开始执行程序。此时的转移不是延迟转移。

5.6.4 一般非法指令

除紧接延迟转移指令（延迟槽）以外的未定义代码（含在 FPU 模块待机时的 FPU 指令以及与 FPU 有关的 CPU 指令）进行解码时，就开始一般非法指令的异常处理。另外， FPU 为模块待机状态时，浮点指令和与 FPU 有关的 CPU 指令作为未定义代码处理，分配在除紧接延迟转移指令（延迟槽）以外时，如果该指令被解码，就开始一般非法指令的异常处理。

进行一般非法指令的异常处理时， CPU 按照和槽非法指令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 值是此未定义代码的起始地址。

5.6.5 整数除法指令

如果整数除法指令执行被零除或者整数除法运算的结果上溢时，就发生整数除法异常。成为被零除异常源的指令是 DIVU 和 DIVS。成为上溢异常源的指令只有 DIVS，只有用 -1 除负的最大值时才发生该异常。整数除法异常发生时， CPU 的运行如下：

1. 从异常处理向量表中取出与发生的整数除法异常相对应的异常服务程序的起始地址。
2. 将状态寄存器 （SR）保存到堆栈。
3. 将程序计数器 （PC）保存到堆栈。保存的PC值为发生异常的整数除法指令的起始地址。
4. 转移到从异常处理向量表中取出的异常服务程序的起始地址，并开始执行程序。此时的转移不是延迟转移。

5.6.6 浮点运算指令

浮点状态 / 控制寄存器（FPSCR）的 FPU 异常允许字段（Enable）中的 V、Z、O、U 或 I 被置位时，发生 FPU 异常。这表示浮点运算指令引起了 IEEE754 规格中定义的无效运算异常、被零除异常、上溢（有可能引起上溢的指令）、下溢 （有可能引起下溢的指令）以及不正确异常 （有可能引起引起不正确异常的指令）。

成为异常源的浮点运算指令如下所示：

FADD、FSUB、FMUL、FDIV、FMAC、FCMP/EQ、FCMP/GT、FLOAT、FTRC、FCNVDS、FCNVSD、FSQRT

只有对应的允许位被置位时，才发生 FPU 异常。FPU 检测出异常源时，停止 FPU 的运行并通知 CPU 发生异常。开始异常处理时， CPU 的运行如下：

1. 从异常处理向量表中取出与发生的FPU异常相对应的异常服务程序的起始地址。
2. 将状态寄存器 （SR）保存到堆栈。
3. 将程序计数器 （PC）保存到堆栈。保存的PC值为已执行的最后一条指令的下一条指令的起始地址。
4. 转移到从异常处理向量表中取出的异常服务程序的起始地址，并开始执行程序。此时的转移不是延迟转移。

FPSCR 的 FPU 异常标志字段 （Flag）与是否接受 FPU 异常无关，不断被更新，直到用户用明示性指令清除前一直保持被置位状态。每执行一次 FPU 指令， FPSCR 的 FPU 异常源字段 （Cause）就会发生变化。

另外， FPSCR 的 FPU 异常允许字段 （Enable）中的 V 位被置位，且 FPSCR 的 QIS 位被置位时，将 qNaN 和±∞输入浮点运算指令源时，就开始 FPU 异常。

5.7 不接受异常处理时

如表 5.11 所示，地址错误、FPU 异常、寄存器组错误（上溢）和中断在紧接延迟转移指令之后发生时，有不被立即接受而被保留的情况。此时，当能接受异常的指令被解码时，才接受这些异常。

表 5.11 发生紧接延迟转移指令之后的异常源

发生时间	异常源			
	地址错误	FPU 异常	寄存器组错误 （上溢）	中断
紧接着延迟转移指令 * 之后	×	×	×	×

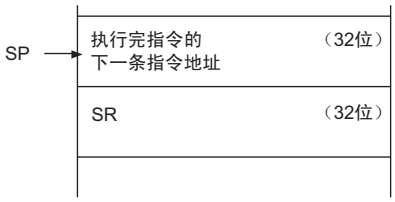
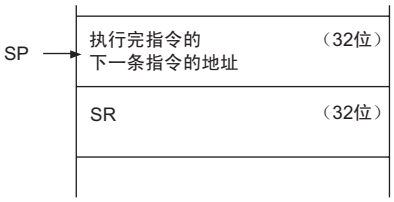
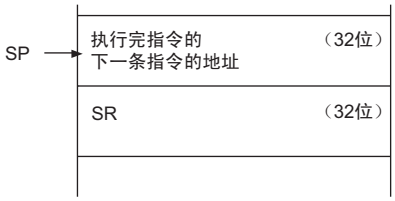
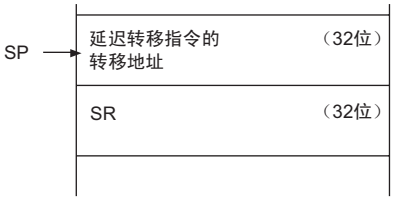
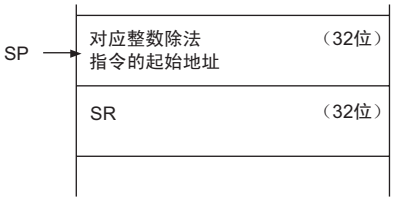
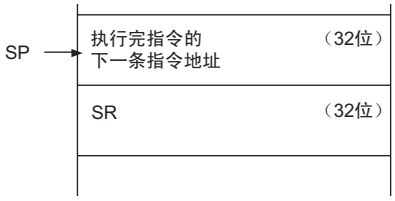
【符号说明】×：不被接受

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

5.8　异常处理后堆栈的状态

异常处理结束后的堆栈状态如表 5.12 所示。

表 5.12　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		中断	
寄存器组错误（上溢）		寄存器组错误（下溢）	
陷阱指令		槽非法指令	
一般非法指令		整数除法指令	
FPU 异常			

5.9 使用时的注意事项

5.9.1 堆栈指针 （SP）的值

SP 的值必须是 4 的倍数。否则，如果在异常处理期间存取堆栈，就会发生地址错误。

5.9.2 向量基址寄存器 （VBR）的值

VBR 的值必须是 4 的倍数。否则，如果在异常处理期间存取堆栈，就会发生地址错误。

5.9.3 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，在该异常处理结束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（写入）。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 SP 的值，因此输出发生错误的地址。此时，堆栈存取的写数据不定。

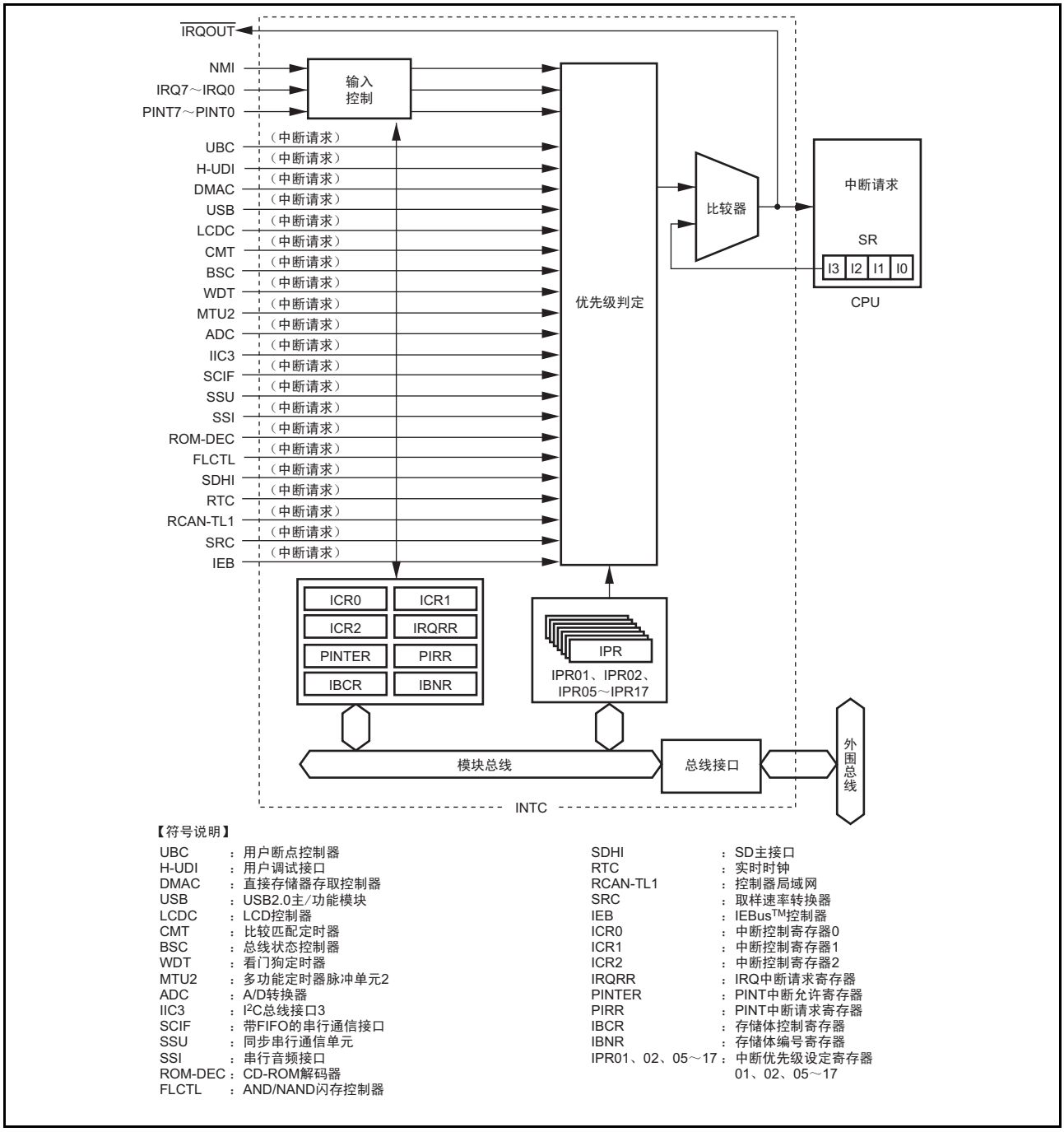
第 6 章 中断控制器 （INTC）

中断控制器 （INTC）判断中断源的优先级以及控制对 CPU 的中断请求。 INTC 有设定各中断优先级的寄存器，用户根据该寄存器中设定的优先级来处理中断请求。

6.1 特点

- 能将中断优先级设定为16级
  能通过15个中断优先级设定寄存器，将IRQ中断、PINT中断和内置外围模块中断的优先级根据请求源用最多16个级别进行设定。
- NMI的噪声消除功能
  有表示NMI引脚状态的NMI输入电平位。引脚状态可通过在中断异常服务程序读取此位来确认，用作噪声消除功能。
- 发生中断时，能将该信息输出到外部 （$\overline{\text{IRQOUT}}$引脚）
  例如：本LSI释放总线权时，能将内置外围模块中断的发生告知外部总线主控器，并请求总线权。
- 寄存器组
  本LSI内置寄存器组，能在进行中断处理的同时，高速进行寄存器的保存、返回。

INTC 的框图如图 6.1 所示。



【符号说明】

UBC	：用户断点控制器	SDHI	：SD主接口
H-UDI	：用户调试接口	RTC	：实时时钟
DMAC	：直接存储器存取控制器	RCAN-TL1	：控制器局域网
USB	：USB2.0主/功能模块	SRC	：取样速率转换器
LCDC	：LCD控制器	IEB	：IEBus™控制器
CMT	：比较匹配定时器	ICR0	：中断控制寄存器0
BSC	：总线状态控制器	ICR1	：中断控制寄存器1
WDT	：看门狗定时器	ICR2	：中断控制寄存器2
MTU2	：多功能定时器脉冲单元2	IRQRR	：IRQ中断请求寄存器
ADC	：A/D转换器	PINTER	：PINT中断允许寄存器
IIC3	：I^2C总线接口3	PIRR	：PINT中断请求寄存器
SCIF	：带FIFO的串行通信接口	IBCR	：存储体控制寄存器
SSU	：同步串行通信单元	IBNR	：存储体编号寄存器
SSI	：串行音频接口	IPR01、02、05～17	：中断优先级设定寄存器 01、02、05～17
ROM-DEC	：CD-ROM解码器		
FLCTL	：AND/NAND闪存控制器		

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	IRQ7 ～ IRQ0	输入	输入可屏蔽的中断请求信号
	PINT7 ～ PINT0	输入	
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。通过这些寄存器进行中断优先级的设定和外部中断输入信号的检测控制等。

表 6.2　寄存器结构

寄存器名	简称	R/W	初始值	地址	存取长度
中断控制寄存器 0	ICR0	R/W	*1	H'FFFE0800	16、32
中断控制寄存器 1	ICR1	R/W	H'0000	H'FFFE0802	16、32
中断控制寄存器 2	ICR2	R/W	H'0000	H'FFFE0804	16、32
IRQ 中断请求寄存器	IRQRR	R/(W)*2	H'0000	H'FFFE0806	16、32
PINT 中断允许寄存器	PINTER	R/W	H'0000	H'FFFE0808	16、32
PINT 中断请求寄存器	PIRR	R	H'0000	H'FFFE080A	16、32
存储体控制寄存器	IBCR	R/W	H'0000	H'FFFE080C	16、32
存储体编号寄存器	IBNR	R/W	H'0000	H'FFFE080E	16、32
中断优先级设定寄存器 01	IPR01	R/W	H'0000	H'FFFE0818	16、32
中断优先级设定寄存器 02	IPR02	R/W	H'0000	H'FFFE081A	16、32
中断优先级设定寄存器 05	IPR05	R/W	H'0000	H'FFFE0820	16、32
中断优先级设定寄存器 06	IPR06	R/W	H'0000	H'FFFE0C00	16、32
中断优先级设定寄存器 07	IPR07	R/W	H'0000	H'FFFE0C02	16、32
中断优先级设定寄存器 08	IPR08	R/W	H'0000	H'FFFE0C04	16、32
中断优先级设定寄存器 09	IPR09	R/W	H'0000	H'FFFE0C06	16、32
中断优先级设定寄存器 10	IPR10	R/W	H'0000	H'FFFE0C08	16、32
中断优先级设定寄存器 11	IPR11	R/W	H'0000	H'FFFE0C0A	16、32
中断优先级设定寄存器 12	IPR12	R/W	H'0000	H'FFFE0C0C	16、32
中断优先级设定寄存器 13	IPR13	R/W	H'0000	H'FFFE0C0E	16、32
中断优先级设定寄存器 14	IPR14	R/W	H'0000	H'FFFE0C10	16、32
中断优先级设定寄存器 15	IPR15	R/W	H'0000	H'FFFE0C12	16、32
中断优先级设定寄存器 16	IPR16	R/W	H'0000	H'FFFE0C14	16、32
中断优先级设定寄存器 17	IPR17	R/W	H'0000	H'FFFE0C16	16、32

【注】　*1　NMI 引脚为高电平时：H'8000，为低电平时：H'0000。

*2　为了清除标志，只能在读 1 后写 0。

6.3.1　中断优先级设定寄存器 01、02、05 ～ 17（IPR01、IPR02、IPR05 ～ IPR17）

IPR01、IPR02、IPR05 ～ IPR17 均为 16 位可读写寄存器，设定 IRQ 中断、PINT 中断和内置外围模块中断的优先级 （0 ～ 15 级）。中断请求源与 IPR01、 IPR02、 IPR05 ～ IPR17 各位的对应关系如表 6.3 所示。

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

表 6.3　中断请求源与 IPR01、 IPR02、 IPR05 ～ IPR17

寄存器名	位			
	15 ～ 12	11 ～ 8	7 ～ 4	3 ～ 0
中断优先级设定寄存器 01	IRQ0	IRQ1	IRQ2	IRQ3
中断优先级设定寄存器 02	IRQ4	IRQ5	IRQ6	IRQ7
中断优先级设定寄存器 05	PINT7 ～ PINT0	保留	保留	保留
中断优先级设定寄存器 06	DMAC0	DMAC1	DMAC2	DMAC3
中断优先级设定寄存器 07	DMAC4	DMAC5	DMAC6	DMAC7
中断优先级设定寄存器 08	USB	LCDC	CMT0	CMT1
中断优先级设定寄存器 09	BSC	WDT	MTU0 (TGI0A ～ TGI0D)	MTU0 (TCI0V、 TGI0E、 TGI0F)
中断优先级设定寄存器 10	MTU1 (TGI1A、 TGI1B)	MTU1 (TCI1V、 TCI1U)	MTU2 (TGI2A、 TGI2B)	MTU2 (TCI2V、 TCI2U)
中断优先级设定寄存器 11	MTU3 (TGI3A ～ TGI3D)	MTU3 (TCI3V)	MTU4 (TGI4A ～ TGI4D)	MTU4 (TCI4V)
中断优先级设定寄存器 12	ADC	IIC3-0	IIC3-1	IIC3-2
中断优先级设定寄存器 13	IIC3-3	SCIF0	SCIF1	SCIF2
中断优先级设定寄存器 14	SCIF3	SSU0	SSU1	SSI0
中断优先级设定寄存器 15	SSI1	SSI2	SSI3	ROM-DEC
中断优先级设定寄存器 16	FLCTL	SDHI	RTC	RCAN0
中断优先级设定寄存器 17	RCAN1	SRC	IEB	保留

如表 6.3 所示，通过向 bit15 ～ 12、 bit11 ～ 8、 bit7 ～ 4、 bit3 ～ 0 的各 4 位设定从 H'0 （0000）到 H'F （1111）范围内的值，来设定各自所对应的中断优先级。设定 H'0 时为中断优先级 0 （最低），设定 H'F 时为中断优先级 15 （最高）。

6.3.2　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引脚为高电平时为1、低电平时为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。能通过读取本位得知 NMI 引脚的电平。 写操作无效。 0：NMI 引脚输入低电平 1：NMI 引脚输入高电平
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 选择在 NMI 输入的下降沿或者上升沿的检测中断请求信号。 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。

6.3.3　中断控制寄存器 1 （ICR1）

ICR1 是对外部中断输入引脚 IRQ7 ～ IRQ0，单独指定低电平、下降沿、上升沿、双边沿检测模式的 16 位寄存器。

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	IRQ 的检测选择 选择是在低电平还是在下降沿、上升沿、双边沿检测 IRQ7 ～ IRQ0 引脚的中断信号。 00：在 IRQn 输入的低电平检测中断请求。 01：在 IRQn 输入的下降沿检测中断请求。 10：在 IRQn 输入的上升沿检测中断请求。 11：在 IRQn 输入的双边沿检测中断请求。
14	IRQ70S	0	R/W	
13	IRQ61S	0	R/W	
12	IRQ60S	0	R/W	
11	IRQ51S	0	R/W	
10	IRQ50S	0	R/W	
9	IRQ41S	0	R/W	
8	IRQ40S	0	R/W	
7	IRQ31S	0	R/W	
6	IRQ30S	0	R/W	
5	IRQ21S	0	R/W	
4	IRQ20S	0	R/W	
3	IRQ11S	0	R/W	
2	IRQ10S	0	R/W	
1	IRQ01S	0	R/W	
0	IRQ00S	0	R/W	

【符号说明】 n=7 ～ 0

6.3.4　中断控制寄存器 2 （ICR2）

ICR2 是对外部中断输入引脚 PINT7 ～ PINT0 单独指定低电平、高电平检测模式的 16 位寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PINT7S	PINT6S	PINT5S	PINT4S	PINT3S	PINT2S	PINT1S	PINT0S
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
15 ～ 8	—	全 0	R	保留位 读写的值总是 0。
7	PINT7S	0	R/W	PINT 的检测选择 选择在低电平还是在高电平检测 PINT7 ～ PINT0 引脚的中断信号。 0：在 PINTn 输入的低电平检测中断请求。 1：在 PINTn 输入的高电平检测中断请求。
6	PINT6S	0	R/W	
5	PINT5S	0	R/W	
4	PINT4S	0	R/W	
3	PINT3S	0	R/W	
2	PINT2S	0	R/W	
1	PINT1S	0	R/W	
0	PINT0S	0	R/W	

【符号说明】 n=7 ～ 0

6.3.5　IRQ 中断请求寄存器 （IRQRR）

IRQRR 是 16 位寄存器，表示外部中断输入引脚 IRQ7 ～ IRQ0 的中断请求。将 IRQ7 ～ IRQ0 中断设定为边沿检测时，通过读取 IRQ7F ～ IRQ0F=1 后，对 IRQ7 ～ IRQ0 写入 0，可以取销被保持的中断请求。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
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)*

【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 ～ 8	—	全 0	R	保留位 读写的值总是 0。
7	IRQ7F	0	R/(W)*	IRQ 中断请求 表示 IRQ7 ～ IRQ0 中断请求的状态。 电平检测时 0：不存在 IRQn 中断请求。 ［清除条件］ • IRQn 输入为高电平时 1：存在 IRQn 中断请求。 ［置位条件］ • IRQn 输入为低电平时 边沿检测时 0：未检测出 IRQn 中断请求。 ［清除条件］ • 在读 IRQnF=1 的状态后写 0 时 • 已执行 IRQn 中断异常处理时 1：检测出 IRQn 中断请求。 ［置位条件］ • 在 IRQn 引脚发生对应 ICR1 的 IRQn1S、 IRQn0S 的边沿时
6	IRQ6F	0	R/(W)*	
5	IRQ5F	0	R/(W)*	
4	IRQ4F	0	R/(W)*	
3	IRQ3F	0	R/(W)*	
2	IRQ2F	0	R/(W)*	
1	IRQ1F	0	R/(W)*	
0	IRQ0F	0	R/(W)*	

【符号说明】 n=7 ～ 0

6.3.6　PINT 中断允许寄存器 （PINTER）

PINTER 是对外部中断输入引脚 PINT7 ～ PINT0 允许输入中断请求的 16 位寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PINT7E	PINT6E	PINT5E	PINT4E	PINT3E	PINT2E	PINT1E	PINT0E
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
15 ～ 8	—	全 0	R	保留位 读写的值总是 0。
7	PINT7E	0	R/W	PINT 允许位 选择是否允许对 PINT7 ～ PINT0 引脚输入中断请求。 0：禁止 PINTn 输入中断请求。 1：允许 PINTn 输入中断请求。
6	PINT6E	0	R/W	
5	PINT5E	0	R/W	
4	PINT4E	0	R/W	
3	PINT3E	0	R/W	
2	PINT2E	0	R/W	
1	PINT1E	0	R/W	
0	PINT0E	0	R/W	

【符号说明】 n=7 ～ 0

6.3.7　PINT 中断请求寄存器 （PIRR）

PIRR 为 16 位寄存器，表示外部中断 PINT7 ～ PINT0 的中断请求。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PINT7R	PINT6R	PINT5R	PINT4R	PINT3R	PINT2R	PINT1R	PINT0R
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 ～ 8	—	全 0	R	保留位 读写的值总是 0。
7	PINT7R	0	R	PINT 中断请求 表示 PINT7 ～ PINT0 中断请求。 0：PINTn 引脚无中断。 1：PINTn 引脚有中断。
6	PINT6R	0	R	
5	PINT5R	0	R	
4	PINT4R	0	R	
3	PINT3R	0	R	
2	PINT2R	0	R	
1	PINT1R	0	R	
0	PINT0R	0	R	

【符号说明】 n=7 ～ 0

6.3.8　存储体控制寄存器 （IBCR）

IBCR 能对中断优先级设定允许 / 禁止使用寄存器组。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	-
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

位	位名	初始值	R/W	说明
15	E15	0	R/W	允许 对中断优先级 15 ～ 1 设定允许 / 禁止使用寄存器组。但用户断点中断总是禁止使用寄存器组。 0：禁止使用寄存器组。 1：允许使用寄存器组。
14	E14	0	R/W	
13	E13	0	R/W	
12	E12	0	R/W	
11	E11	0	R/W	
10	E10	0	R/W	
9	E9	0	R/W	
8	E8	0	R/W	
7	E7	0	R/W	
6	E6	0	R/W	
5	E5	0	R/W	
4	E4	0	R/W	
3	E3	0	R/W	
2	E2	0	R/W	
1	E1	0	R/W	
0	—	0	R	保留位 读写的值总是 0。

6.3.9　存储体编号寄存器 （IBNR）

IBNR 设定允许 / 禁止使用寄存器组，以及允许 / 禁止寄存器组上溢异常。同时表示通过 BN3 ～ BN0 下一个要保存的存储体编号。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BE[1:0]		BOVE	-	-	-	-	-	-	-	-	-	BN[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	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15、 14	BE[1:0]	00	R/W	寄存器组允许 设定允许 / 禁止使用寄存器组。 00：禁止所有中断使用存储体。忽视 IBCR 的设定。 01：对于 NMI、用户断点以外的所有中断，允许存储体的使用。忽视 IBCR 的设定。 10：保留 （禁止设定） 11：根据 IBCR 的设定，使用寄存器组。
13	BOVE	0	R/W	寄存器组上溢允许 设定允许 / 禁止寄存器组上溢异常。 0：禁止发生寄存器组上溢异常。 1：允许发生寄存器组上溢异常。
12 ～ 4	—	全 0	R	保留位 读写的值总是 0。
3 ～ 0	BN[3:0]	0000	R	存储体编号 表示下一个要保存的存储体编号。接受到使用寄存器组的中断时，保存到 BN3 ～ BN0 所示的寄存器组，然后将 BN+1。通过执行寄存器组返回指令， BN-1 后，从寄存器组返回。

6.4 中断源

中断源分为 NMI、用户断点、H-UDI、IRQ、PINT、内置外围模块 6 种。各中断的优先级用中断优先级的值 （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 时，该中断总是被屏蔽。

6.4.1 NMI 中断

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

NMI中断优先级为16，但可通过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。

6.4.2 用户断点中断

用户断点中断是在用户断点控制器 （UBC）中设定的断点条件成立时发生的中断，优先级为 15。通过用户断点中断异常处理将 SR 的 I3 ～ I0 位设定为 15。关于用户断点，请参照“**第 7 章 用户断点控制器（UBC）**”。

6.4.3 H-UDI 中断

用户调试接口 （H-UDI）中断优先级为 15。串行输入时，发生 H-UDI 中断指令。在边沿检测出 H-UDI 中断请求，并被保持到被接受为止。通过 H-UDI 中断异常处理，将 SR 的 I3 ～ I0 位设定为 15。关于 H-UDI 中断，请参照 “**第 33 章 用户调试接口 （H-UDI）**”。

6.4.4 IRQ 中断

IRQ 中断是由 IRQ7 ～ IRQ0 引脚输入引起的中断。IRQ 中断能通过设定中断控制寄存器 1（ICR1）的 IRQ 检测选择位 （IRQ71S ～ IRQ01S、IRQ70S ～ IRQ00S），给每个引脚选择低电平、下降沿、上升沿或双边沿检测。另外，能通过中断优先级设定寄存器 01、02（IPR01、IPR02），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 IRQ7 ～ IRQ0 引脚为低电平期间，将中断请求信号发送给 INTC。IRQ7 ～ IRQ0 引脚变为高电平时，停止对 INTC 发送中断请求信号。能通过读取 IRQ 中断请求寄存器（IRQRR）的 IRQ 中断请求位 （IRQ7F ～ IRQ0F）确认中断请求。

如果将 IRQ 中断设定为边沿检测，就通过 IRQ7 ～ IRQ0 引脚的变化检测中断请求，并将中断请求信号发送给 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，能通过读取 IRQRR 的 IRQ7F ～ IRQ0F 位确认是否检测到 IRQ 中断请求，能通过在读出 1 后写入 0 来取消 IRQ 中断请求的检测结果。

IRQ 中断异常处理将 SR 的 I3 ～ I0 位设定为已接受的 IRQ 中断优先级的值。

从 IRQ 中断异常服务程序返回时，为避免再次错误接受，请确认在 IRQ 中断请求寄存器 （IRQRR）清除中断请求后，再执行 RTE 指令。

6.4.5 PINT 中断

PINT 中断是由 PINT7 ～ PINT0 引脚输入引起的中断。通过 PINT 中断允许寄存器 （PINTER）的 PINT 允许位 （PINT7E ～ PINT0E）来允许中断请求输入。 PINT7 ～ PINT0 能通过设定中断控制寄存器 2 （ICR2）的 PINT 检测选择位（PINT7S ～ PINT0S），给每个引脚选择低电平或高电平检测。优先级能通过中断优先级设定寄存器 05 （IPR05）的 bit15 ～ 12，在 0 ～ 15 范围内给 PINT7 ～ PINT0 统一设定优先级。

如果将 PINT7 ～ PINT0 设定为低电平检测，就在 PINT 引脚为低电平期间，将中断请求信号发送给 INTC。PINT 引脚变为高电平时，停止对 INTC 发送中断请求信号。能通过读取 PINT 中断请求寄存器（PIRR）的 PINT 中断请求位 （PINT7R ～ PINT0R）确认中断请求的等级。除极性相反的情况以外，在设定为高电平检测时也相同。 PINT 中断异常处理将 SR 的 I3 ～ I0 设定为 PINT 中断的优先级值。

从 PINT 中断异常服务程序返回时，为避免再次错误接受，请确认在 PINT 中断请求寄存器 （PIRR）清除中断请求后，再执行 RTE 指令。

6.4.6 内置外围模块中断

内置外围模块中断是由以下所示的内置外围模块发生的中断。

- 直接存储器存取控制器 （DMAC）
- USB2.0 主/功能模块 （USB）
- LCD 控制器 （LCDC）
- 比较匹配定时器 （CMT）
- 总线状态控制器 （BSC）
- 看门狗定时器 （WDT）
- 多功能定时器脉冲单元 2 （MTU2）
- A/D 转换器 （ADC）
- I^2C 总线接口 3 （IIC3）
- 带 FIFO 的串行通信接口 （SCIF）
- 同步串行通信单元 （SSU）
- 串行音频接口 （SSI）
- CD-ROM 解码器 （ROM-DEC）
- AND/NAND 闪存控制器 （FLCTL）
- SD 主接口 （SDHI）
- 实时时钟 （RTC）
- 控制器局域网 （RCAN-TL1）
- 取样速率转换器 （SRC）
- IEBus™ 控制器 （IEB）

因为给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。能通过中断优先级设定寄存器 05 ～ 17 （IPR05 ～ IPR17），在 0 ～ 15 的范围内给每个模块设定优先级。内置外围模块中断的异常处理将 SR 的 I3 ～ I0 位设定为已接受的内置外围模块中断优先级的值。

6.5　中断异常处理向量表和优先级

中断源、向量号、向量表地址偏移量和中断优先级如**表 6.4** 所示。

给各中断源分别分配了不同的向量号和向量表地址偏移量，向量表地址由向量号和向量表地址偏移量算出。通过中断异常处理从向量表地址所示的向量表中取出中断异常服务程序的起始地址。有关向量表地址的算法，请参照 **“第 5 章　异常处理”**的 **“表 5.4　异常处理向量表地址的计算方法”**。

能通过中断优先级设定寄存器 01、02、05 ～ 17（IPR01、IPR02、IPR05 ～ IPR17），在 0 ～ 15 的范围内给每个引脚或者模块任意设定 IRQ 中断、PINT 中断和内置外围模块中断的优先级。但是如**表 6.4** 的 IPR 设定单位内的优先级所示，在 IPR05 ～ IPR17 中发生同一 IPR 内的多个中断时，优先级已经被设定不能更改。通过上电复位，将 IRQ 中断、PINT 中断和内置外围模块中断的优先级设定为 0。在将多个中断源的优先顺序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 6.4** 所示的默认优先级进行处理。

表 6.4　中断异常向量和优先级

中断源		中断向量		中断优先级（初始值）	对应的 IPR（位）	IPR 设定单位内的优先级	默认优先级
		向量	向量表地址偏移量				
NMI		11	H'0000002C ～ H'0000002F	16	—	—	高
用户断点		12	H'00000030 ～ H'00000033	15	—	—	
H-UDI		14	H'00000038 ～ H'0000003B	15	—	—	
IRQ	IRQ0	64	H'00000100 ～ H'00000103	0 ～ 15(0)	IPR01(15 ～ 12)	—	
	IRQ1	65	H'00000104 ～ H'00000107	0 ～ 15(0)	IPR01(11 ～ 8)	—	
	IRQ2	66	H'00000108 ～ H'0000010B	0 ～ 15(0)	IPR01(7 ～ 4)	—	
	IRQ3	67	H'0000010C ～ H'0000010F	0 ～ 15(0)	IPR01(3 ～ 0)	—	
	IRQ4	68	H'00000110 ～ H'00000113	0 ～ 15(0)	IPR02(15 ～ 12)	—	
	IRQ5	69	H'00000114 ～ H'00000117	0 ～ 15(0)	IPR02(11 ～ 8)	—	
	IRQ6	70	H'00000118 ～ H'0000011B	0 ～ 15(0)	IPR02(7 ～ 4)	—	
	IRQ7	71	H'0000011C ～ H'0000011F	0 ～ 15(0)	IPR02(3 ～ 0)	—	
PINT	PINT0	80	H'00000140 ～ H'00000143	0 ～ 15(0)	IPR05(15 ～ 12)	1	
	PINT1	81	H'00000144 ～ H'00000147			2	
	PINT2	82	H'00000148 ～ H'0000014B			3	
	PINT3	83	H'0000014C ～ H'0000014F			4	
	PINT4	84	H'00000150 ～ H'00000153			5	
	PINT5	85	H'00000154 ～ H'00000157			6	
	PINT6	86	H'00000158 ～ H'0000015B			7	
	PINT7	87	H'0000015C ～ H'0000015F			8	低

中断源			中断向量		中断优先级（初始值）	对应的 IPR（位）	IPR 设定单位内的优先级	默认优先级
			向量	向量表地址偏移量				
DMAC	DMAC0	DEI0	108	H'000001B0 ～ H'000001B3	0 ～ 15(0)	IPR06(15 ～ 12)	1	高
		HEI0	109	H'000001B4 ～ H'000001B7			2	
	DMAC1	DEI1	112	H'000001C0 ～ H'000001C3	0 ～ 15(0)	IPR06(11 ～ 8)	1	
		HEI1	113	H'000001C4 ～ H'000001C7			2	
	DMAC2	DEI2	116	H'000001D0 ～ H'000001D3	0 ～ 15(0)	IPR06(7 ～ 4)	1	
		HEI2	117	H'000001D4 ～ H'000001D7			2	
	DMAC3	DEI3	120	H'000001E0 ～ H'000001E3	0 ～ 15(0)	IPR06(3 ～ 0)	1	
		HEI3	121	H'000001E4 ～ H'000001E7			2	
	DMAC4	DEI4	124	H'000001F0 ～ H'000001F3	0 ～ 15(0)	IPR07(15 ～ 12)	1	
		HEI4	125	H'000001F4 ～ H'000001F7			2	
	DMAC5	DEI5	128	H'00000200 ～ H'00000203	0 ～ 15(0)	IPR07(11 ～ 8)	1	
		HEI5	129	H'00000204 ～ H'00000207			2	
	DMAC6	DEI6	132	H'00000210 ～ H'00000213	0 ～ 15(0)	IPR07(7 ～ 4)	1	
		HEI6	133	H'00000214 ～ H'00000217			2	
	DMAC7	DEI7	136	H'00000220 ～ H'00000223	0 ～ 15(0)	IPR07(3 ～ 0)	1	
		HEI7	137	H'00000224 ～ H'00000227			2	
USB	USBI		140	H'00000230 ～ H'00000233	0 ～ 15(0)	IPR08(15 ～ 12)	—	
LCDC	LCDCI		141	H'00000234 ～ H'00000237	0 ～ 15(0)	IPR08(11 ～ 8)	—	
CMT	CMI0		142	H'00000238 ～ H'0000023B	0 ～ 15(0)	IPR08(7 ～ 4)		
	CMI1		143	H'0000023C ～ H'0000023F	0 ～ 15(0)	IPR08(3 ～ 0)	—	
BSC	CMI		144	H'00000240 ～ H'00000243	0 ～ 15(0)	IPR09(15 ～ 12)	—	
WDT	ITI		145	H'00000244 ～ H'00000247	0 ～ 15(0)	IPR09(11 ～ 8)	—	低

中断源			中断向量		中断优先级（初始值）	对应的 IPR（位）	IPR 设定单位内的优先级	默认优先级
			向量	向量表地址偏移量				
MTU2	MTU0	TGI0A	146	H'00000248 ～ H'0000024B	0 ～ 15(0)	IPR09(7 ～ 4)	1	高
		TGI0B	147	H'0000024C ～ H'0000024F			2	▲
		TGI0C	148	H'00000250 ～ H'00000253			3	
		TGI0D	149	H'00000254 ～ H'00000257			4	
		TCI0V	150	H'00000258 ～ H'0000025B	0 ～ 15(0)	IPR09(3 ～ 0)	1	
		TGI0E	151	H'0000025C ～ H'0000025F			2	
		TGI0F	152	H'00000260 ～ H'00000263			3	
	MTU1	TGI1A	153	H'00000264 ～ H'00000267	0 ～ 15(0)	IPR10(15 ～ 12)	1	
		TGI1B	154	H'00000268 ～ H'0000026B			2	
		TCI1V	155	H'0000026C ～ H'0000026F	0 ～ 15(0)	IPR10(11 ～ 8)	1	
		TCI1U	156	H'00000270 ～ H'00000273			2	
	MTU2	TGI2A	157	H'00000274 ～ H'00000277	0 ～ 15(0)	IPR10(7 ～ 4)	1	
		TGI2B	158	H'00000278 ～ H'0000027B			2	
		TCI2V	159	H'0000027C ～ H'0000027F	0 ～ 15(0)	IPR10(3 ～ 0)	1	
		TCI2U	160	H'00000280 ～ H'00000283			2	
	MTU3	TGI3A	161	H'00000284 ～ H'00000287	0 ～ 15(0)	IPR11(15 ～ 12)	1	
		TGI3B	162	H'00000288 ～ H'0000028B			2	
		TGI3C	163	H'0000028C ～ H'0000028F			3	
		TGI3D	164	H'00000290 ～ H'00000293			4	
		TCI3V	165	H'00000294 ～ H'00000297	0 ～ 15(0)	IPR11(11 ～ 8)	—	
	MTU4	TGI4A	166	H'00000298 ～ H'0000029B	0 ～ 15(0)	IPR11(7 ～ 4)	1	
		TGI4B	167	H'0000029C ～ H'0000029F			2	
		TGI4C	168	H'000002A0 ～ H'000002A3			3	
		TGI4D	169	H'000002A4 ～ H'000002A7			4	
		TCI4V	170	H'000002A8 ～ H'000002AB	0 ～ 15(0)	IPR11(3 ～ 0)	—	▼
ADC	ADI		171	H'000002AC ～ H'000002AF	0 ～ 15(0)	IPR12(15 ～ 12)	—	低

中断源			中断向量		中断优先级（初始值）	对应的 IPR（位）	IPR 设定单位内的优先级	默认优先级
			向量	向量表地址偏移量				
IIC3	IIC3-0	STPI0	172	H'000002B0 ～ H'000002B3	0 ～ 15(0)	IPR12(11 ～ 8)	1	高
		NAKI0	173	H'000002B4 ～ H'000002B7			2	
		RXI0	174	H'000002B8 ～ H'000002BB			3	
		TXI0	175	H'000002BC ～ H'000002BF			4	
		TEI0	176	H'000002C0 ～ H'000002C3			5	
	IIC3-1	STPI1	177	H'000002C4 ～ H'000002C7	0 ～ 15(0)	IPR12(7 ～ 4)	1	
		NAKI1	178	H'000002C8 ～ H'000002CB			2	
		RXI1	179	H'000002CC ～ H'000002CF			3	
		TXI1	180	H'000002D0 ～ H'000002D3			4	
		TEI1	181	H'000002D4 ～ H'000002D7			5	
	IIC3-2	STPI2	182	H'000002D8 ～ H'000002DB	0 ～ 15(0)	IPR12(3 ～ 0)	1	
		NAKI2	183	H'000002DC ～ H'000002DF			2	
		RXI2	184	H'000002E0 ～ H'000002E3			3	
		TXI2	185	H'000002E4 ～ H'000002E7			4	
		TEI2	186	H'000002E8 ～ H'000002EB			5	
	IIC3-3	STPI3	187	H'000002EC ～ H'000002EF	0 ～ 15(0)	IPR13(15 ～ 12)	1	
		NAKI3	188	H'000002F0 ～ H'000002F3			2	
		RXI3	189	H'000002F4 ～ H'000002F7			3	
		TXI3	190	H'000002F8 ～ H'000002FB			4	
		TEI3	191	H'000002FC ～ H'000002FF			5	
SCIF	SCIF0	BRI0	192	H'00000300 ～ H'00000303	0 ～ 15(0)	IPR13(11 ～ 8)	1	
		ERI0	193	H'00000304 ～ H'00000307			2	
		RXI0	194	H'00000308 ～ H'0000030B			3	
		TXI0	195	H'0000030C ～ H'0000030F			4	
	SCIF1	BRI1	196	H'00000310 ～ H'00000313	0 ～ 15(0)	IPR13(7 ～ 4)	1	
		ERI1	197	H'00000314 ～ H'00000317			2	
		RXI1	198	H'00000318 ～ H'0000031B			3	
		TXI1	199	H'0000031C ～ H'0000031F			4	
	SCIF2	BRI2	200	H'00000320 ～ H'00000323	0 ～ 15(0)	IPR13(3 ～ 0)	1	
		ERI2	201	H'00000324 ～ H'00000327			2	
		RXI2	202	H'00000328 ～ H'0000032B			3	
		TXI2	203	H'0000032C ～ H'0000032F			4	
	SCIF3	BRI3	204	H'00000330 ～ H'00000333	0 ～ 15(0)	IPR14(15 ～ 12)	1	
		ERI3	205	H'00000334 ～ H'00000337			2	
		RXI3	206	H'00000338 ～ H'0000033B			3	
		TXI3	207	H'0000033C ～ H'0000033F			4	
SSU	SSU0	SSERI0	208	H'00000340 ～ H'00000343	0 ～ 15(0)	IPR14(11 ～ 8)	1	
		SSRXI0	209	H'00000344 ～ H'00000347			2	
		SSTXI0	210	H'00000348 ～ H'0000034B			3	
	SSU1	SSERI1	211	H'0000034C ～ H'0000034F	0 ～ 15(0)	IPR14(7 ～ 4)	1	
		SSRXI1	212	H'00000350 ～ H'00000353			2	
		SSTXI1	213	H'00000354 ～ H'00000357			3	低

中断源			中断向量		中断优先级（初始值）	对应的 IPR（位）	IPR 设定单位内的优先级	默认优先级
			向量	向量表地址偏移量				
SSI	SSI0	SSII0	214	H'00000358 ～ H'0000035B	0 ～ 15(0)	IPR14(3 ～ 0)	—	高
	SSI1	SSII1	215	H'0000035C ～ H'0000035F	0 ～ 15(0)	IPR15(15 ～ 12)	—	
	SSI2	SSII2	216	H'00000360 ～ H'00000363	0 ～ 15(0)	IPR15(11 ～ 8)	—	
	SSI3	SSII3	217	H'00000364 ～ H'00000367	0 ～ 15(0)	IPR15(7 ～ 4)	—	
ROM-DEC	ISY		218	H'00000368 ～ H'0000036B	0 ～ 15(0)	IPR15(3 ～ 0)	1	
	IERR		219	H'0000036C ～ H'0000036F			2	
	IARG		220	H'00000370 ～ H'00000373			3	
	ISEC		221	H'00000374 ～ H'00000377			4	
	IBUF		222	H'00000378 ～ H'0000037B			5	
	IREADY		223	H'0000037C ～ H'0000037F			6	
FLCTL	FLSTEI		224	H'00000380 ～ H'00000383	0 ～ 15(0)	IPR16(15 ～ 12)	1	
	FLTENDI		225	H'00000384 ～ H'00000387			2	
	FLTREQ0I		226	H'00000388 ～ H'0000038B			3	
	FLTREQ1I		227	H'0000038C ～ H'0000038F			4	
SDHI	SDHI3		228	H'00000390 ～ H'00000393	0 ～ 15(0)	IPR16(11 ～ 8)	1	
	SDHI0		229	H'00000394 ～ H'00000397			2	
	SDHI1		230	H'00000398 ～ H'0000039B			3	
RTC	ARM		231	H'0000039C ～ H'0000039F	0 ～ 15(0)	IPR16(7 ～ 4)	1	
	PRD		232	H'000003A0 ～ H'000003A3			2	
	CUP		233	H'000003A4 ～ H'000003A7			3	
RCAN-TL1	RCAN0	ERS0	234	H'000003A8 ～ H'000003AB	0 ～ 15(0)	IPR16(3 ～ 0)	1	
		OVR0	235	H'000003AC ～ H'000003AF			2	
		RM00	236	H'000003B0 ～ H'000003B3			3	
		RM10	237	H'000003B4 ～ H'000003B7			4	
		SLE0	238	H'000003B8 ～ H'000003BB			5	
	RCAN1	ERS1	239	H'000003BC ～ H'000003BF	0 ～ 15(0)	IPR17(15 ～ 12)	1	
		OVR1	240	H'000003C0 ～ H'000003C3			2	
		RM01	241	H'000003C4 ～ H'000003C7			3	
		RM11	242	H'000003C8 ～ H'000003CB			4	
		SLE1	243	H'000003CC ～ H'000003CF			5	
SRC	OVF		244	H'000003D0 ～ H'000003D3	0 ～ 15(0)	IPR17(11 ～ 8)	1	
	ODFI		245	H'000003D4 ～ H'000003D7			2	
	IDEI		246	H'000003D8 ～ H'000003DB			3	
IEB	IEBI		247	H'000003DC ～ H'000003DF	0 ～ 15(0)	IPR17(7 ～ 4)	—	低

6.6　运行说明

6.6.1　中断运行的流程

以下说明产生中断时的运行流程，运行流程如图 6.2 所示。

1. 各中断请求源将中断请求信号发送给中断控制器。
2. 中断控制器按照中断优先级设定寄存器01、02、05～17（IPR01、IPR02、IPR05～IPR17），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，如果产生多个设定为同一优先级的中断或者同一IPR设定内的中断，就根据表6.4中所示的默认优先级和IPR设定单位内的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）的中断屏蔽级位 （I3 ～ I0）比较。忽视相同或者低于在I3～I0位设定的优先级的中断，而只接受高于I3～I0位的优先级的中断，并向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要执行的指令进行解码时，检测到从中断控制器送来的中断请求，并将执行该指令替换为执行中断异常处理 （参照图6.4）。
6. 从与接受的中断相对应的异常处理向量表中取出中断异常服务程序的起始地址。
7. 将状态寄存器 （SR）保存到堆栈，将接受的中断的优先级写到SR的I3～I0位。
8. 将程序计数器 （PC）保存到堆栈。
9. 转移到取出的中断异常服务程序的起始地址，并开始执行程序。此时的转移不是延迟转移。
10. 从 $\overline{\text{IRQOUT}}$ 引脚输出高电平。但是在中断控制器接收的其他中断的中断优先级高于正在接收的中断时， $\overline{\text{IRQOUT}}$引脚仍为低电平。

【注】 请在中断处理程序中清除中断源标志。从清除中断源标志到实际取消对 CPU 的中断源，需要表 6.5 记载的 “从发生中断请求，到中断控制器判定优先级以及与 SR 屏蔽位的比较后，对 CPU 发送中断请求信号的时间”。为了不错误地再次接受应该已被清除的中断源，必须在清除后读取中断源标志，然后再执行 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 中断请求寄存器（IRQRR）取消。详细请参照 “6.4.4 IRQ 中断”。
另外，通过上电复位清除由边沿检测保留的中断。

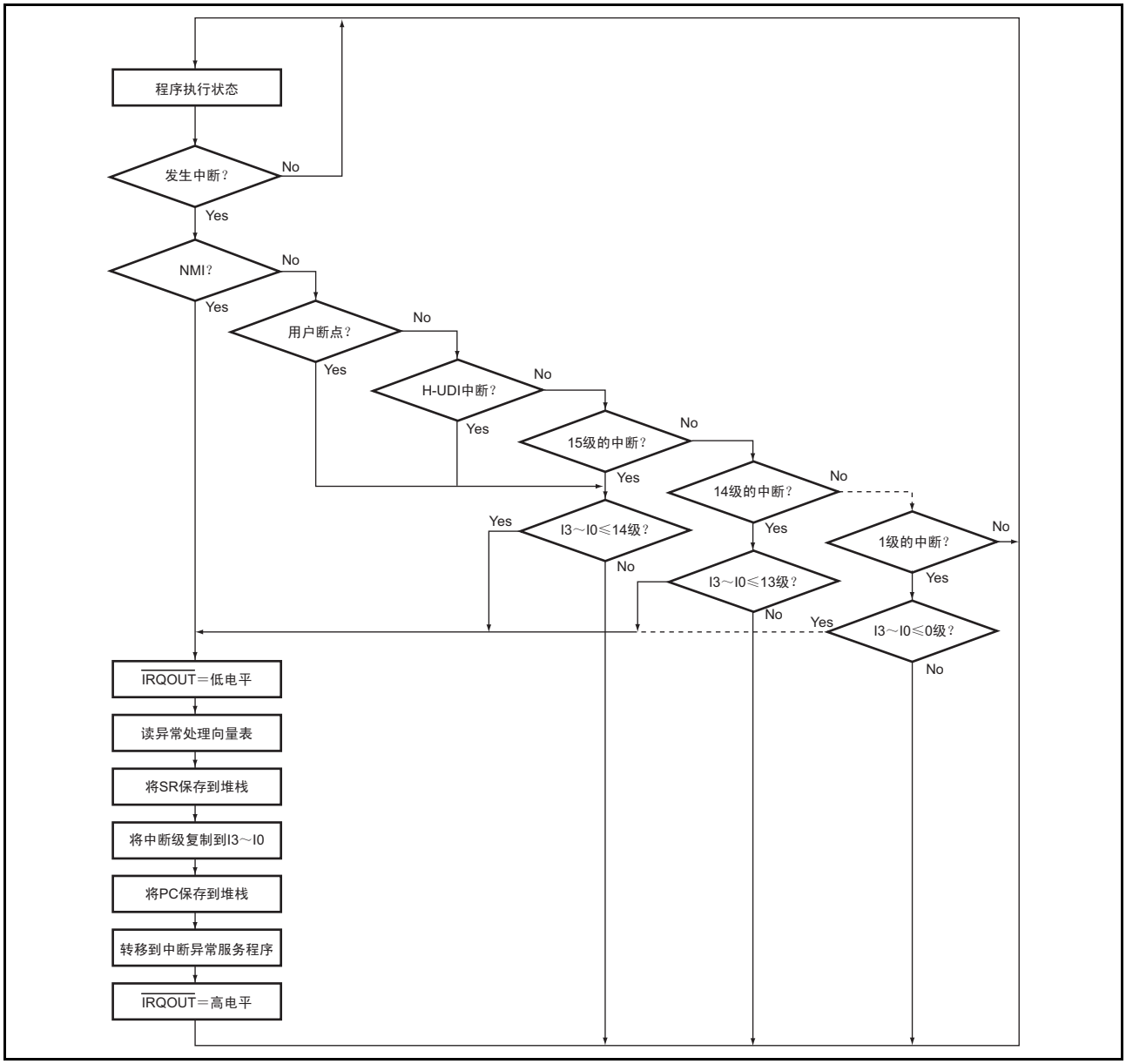


图 6.2　中断运行流程

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 6.3 所示。

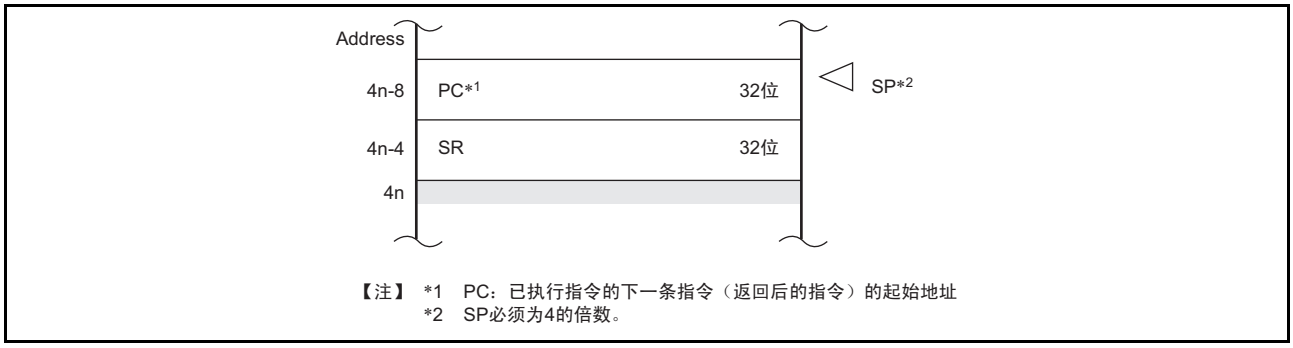


图 6.3　中断异常处理结束后的堆栈状态

6.7　中断响应时间

从发生中断请求到进行中断异常处理、开始取中断异常服务程序的起始指令为止的时间 （中断响应时间）如表 6.5 所示。中断处理在无堆积、有堆积且无寄存器组上溢、有堆积且有寄存器组上溢时运行不同。无堆积时的流水线运行例如图 6.4、图 6.5 所示。有堆积且无寄存器组上溢时的流水线运行例如图 6.6、图 6.7 所示。有堆积且有寄存器组上溢时的流水线运行例如图 6.8、图 6.9 所示。

表 6.5　中断响应时间

<table>
<tr><th colspan="3" rowspan="2">项目</th><th colspan="6">状态数</th><th rowspan="2">备注</th></tr>
<tr><th>NMI</th><th>用户断点</th><th>H-UDI</th><th>IRQ、PINT</th><th>USB</th><th>外围模块（USB 以外）</th></tr>
<tr><td colspan="3">从发生中断请求，到中断控制器判定优先级以及与SR屏蔽位比较后，对 CPU 发送中断请求信号的时间</td><td>$2I_{cyc}$+ $2B_{cyc}$+ $1P_{cyc}$</td><td>$3I_{cyc}$</td><td>$2I_{cyc}$+ $1P_{cyc}$</td><td>$2I_{cyc}$+ $3B_{cyc}$+ $1P_{cyc}$</td><td>$2I_{cyc}$+ $4B_{cyc}$</td><td>$2I_{cyc}$+ $2B_{cyc}$</td><td></td></tr>
<tr><td rowspan="6">从对CPU输入中断请求信号，到执行中的顺序结束后，开始中断异常处理，取中断异常服务程序第一条指令的时间</td><td rowspan="2">无寄存器组。</td><td>最小值</td><td colspan="5">$3I_{cyc}$+m1+m2</td><td rowspan="2">最小值在中断等待时间 0 时。最大值在中断异常处理中，发生更高位的中断请求时。</td></tr>
<tr><td>最大值</td><td colspan="5">$4I_{cyc}$+2(m1+m2)+m3</td></tr>
<tr><td rowspan="2">有寄存器组。无寄存器组上溢。</td><td>最小值</td><td colspan="2">—</td><td colspan="3">$3I_{cyc}$+m1+m2</td><td rowspan="2">最小值在中断等待时间 0 时。最大值在RESBANK指令执行过程中产生中断请求时。</td></tr>
<tr><td>最大值</td><td colspan="2">—</td><td colspan="3">$12I_{cyc}$+m1+m2</td></tr>
<tr><td rowspan="2">有寄存器组。有寄存器组上溢。</td><td>最小值</td><td colspan="2">—</td><td colspan="3">$3I_{cyc}$+m1+m2</td><td rowspan="2">最小值在中断等待时间 0 时。最大值在RESBANK指令执行过程中产生中断请求时。</td></tr>
<tr><td>最大值</td><td colspan="2">—</td><td colspan="3">$3I_{cyc}$+m1+m2+19(m4)</td></tr>
</table>

项目			状态数						备注
			NMI	用户断点	H-UDI	IRQ、PINT	USB	外围模块（USB 以外）	
响应时间	无寄存器组。	最小值	$5I_{cyc}$+ $2B_{cyc}$+ $1P_{cyc}$+ m1+m2	$6I_{cyc}$+m1 +m2	$5I_{cyc}$+ $1P_{cyc}$+m 1+m2	$5I_{cyc}$+ $3B_{cyc}$+ $1P_{cyc}$+ m1+m2	$5I_{cyc}$+ $4B_{cyc}$+ m1+m2	$5I_{cyc}$+$2B_{cyc}$ +m1+m2	200MHz 运行时 *1*2： 0.040 ～ 0.110μs
		最大值	$6I_{cyc}$+ $2B_{cyc}$+ $1P_{cyc}$+2 (m1+m2) +m3	$7I_{cyc}$+2 (m1+m2) +m3	$6I_{cyc}$+ $1P_{cyc}$+2 (m1+m2) +m3	$6I_{cyc}$+ $3B_{cyc}$+ $1P_{cyc}$+2 (m1+m2) +m3	$6I_{cyc}$+ $4B_{cyc}$+ 2(m1+m2) +m3	$6I_{cyc}$+$2B_{cyc}$ +2(m1+m2) +m3	200MHz 运行时 *1*2： 0.060 ～ 0.130μs
	有寄存器组。无寄存器组上溢。	最小值	—	—	$5I_{cyc}$+1P $_{cyc}$+m1+ m2	$5I_{cyc}$+ $3B_{cyc}$+ $1P_{cyc}$+ m1+m2	$5I_{cyc}$+ $4B_{cyc}$+ m1+m2	$5I_{cyc}$+$2B_{cyc}$ +m1+m2	200MHz 运行时 *1*2： 0.070 ～ 0.110μs
		最大值	—	—	$14I_{cyc}$+ $1P_{cyc}$+ m1+m2	$14I_{cyc}$+ $3B_{cyc}$+ $1P_{cyc}$+ m1+m2	$14I_{cyc}$+ $4B_{cyc}$+ m1+m2	$14I_{cyc}$+ $2B_{cyc}$+ m1+m2	200MHz 运行时 *1*2： 0.120 ～ 0.155μs
	有寄存器组。有寄存器组上溢。	最小值	—	—	$5I_{cyc}$+ $1P_{cyc}$+ m1+m2	$5I_{cyc}$+ $3B_{cyc}$+ $1P_{cyc}$+ m1+m2	$5I_{cyc}$+ $4B_{cyc}$+ m1+m2	$5I_{cyc}$+$2B_{cyc}$ +m1+m2	200MHz 运行时 *1*2： 0.065 ～ 0.110μs
		最大值	—	—	$5I_{cyc}$+ $1P_{cyc}$+ m1+m2 +19(m4)	$5I_{cyc}$+ $3B_{cyc}$+ $1P_{cyc}$+ m1+m2 +19(m4)	$5I_{cyc}$+ $4B_{cyc}$+ m1+m2+ 19(m4)	$5I_{cyc}$+$2B_{cyc}$ +m1+m2+ 19(m4)	200MHz 运行时 *1*2： 0.160 ～ 0.205μs

【注】 m1 ～ m4 为下述存储器存取时需要的状态数：

m1：读向量地址 （读取长字）

m2：保存 SR （写入长字）

m3：保存 PC （写入长字）

m4：从存储体对象寄存器 （R0 ～ R14、 GBR、 MACH、 MACL、 PR）返回

*1 ：m1=m2=m3=m4=1I_{cyc} 时

*2 ：（Iϕ、 Bϕ、 Pϕ） =（200MHz、 66MHz、 33MHz）时

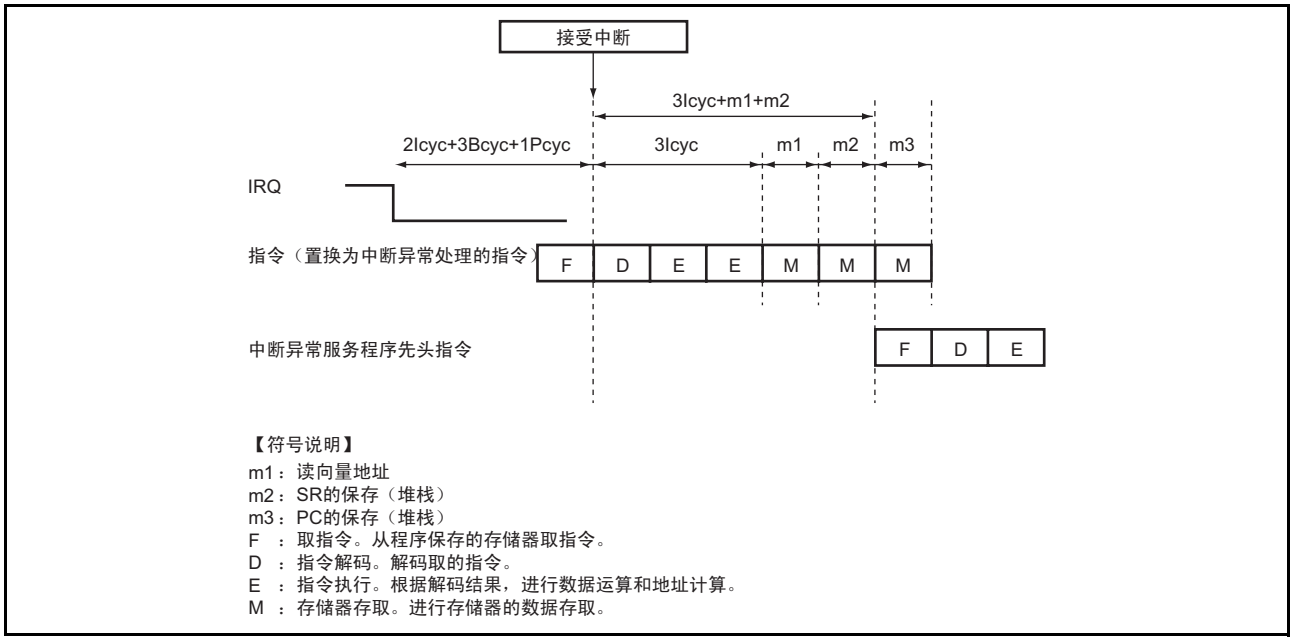


图 6.4　接受 IRQ 中断时的流水线运行例 （无寄存器组）

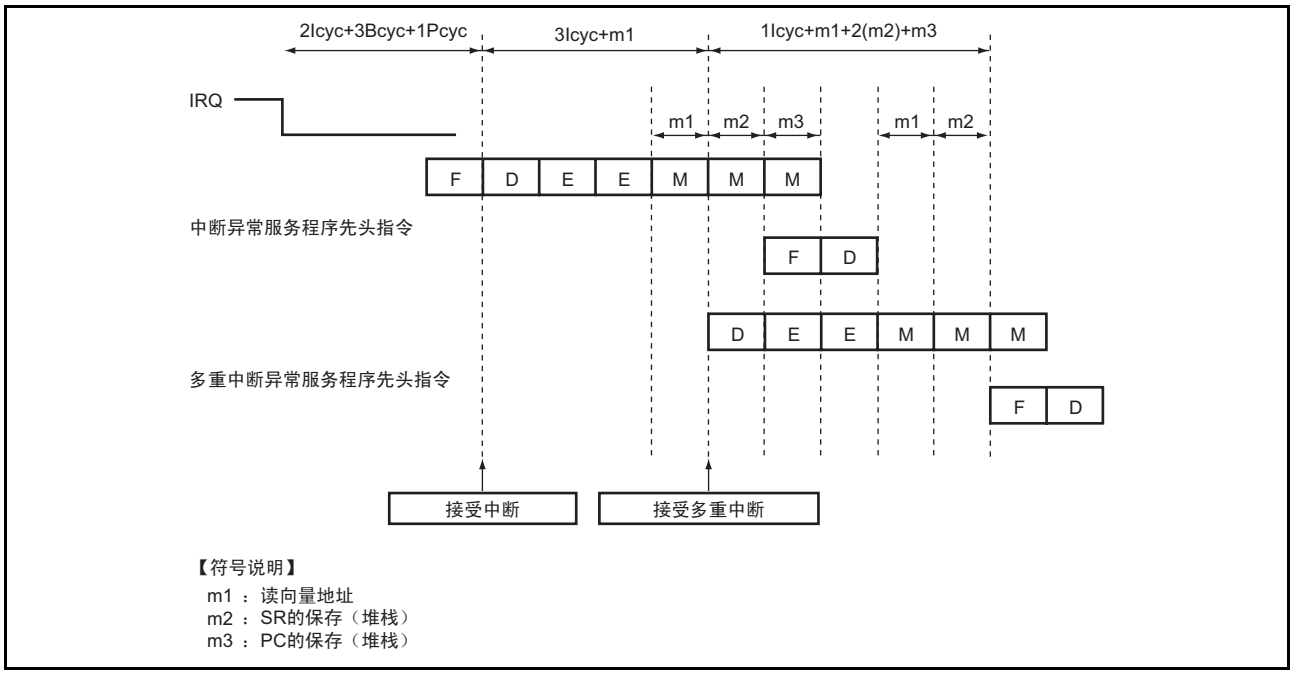


图 6.5　多重中断时的流水线运行例 （无寄存器组）

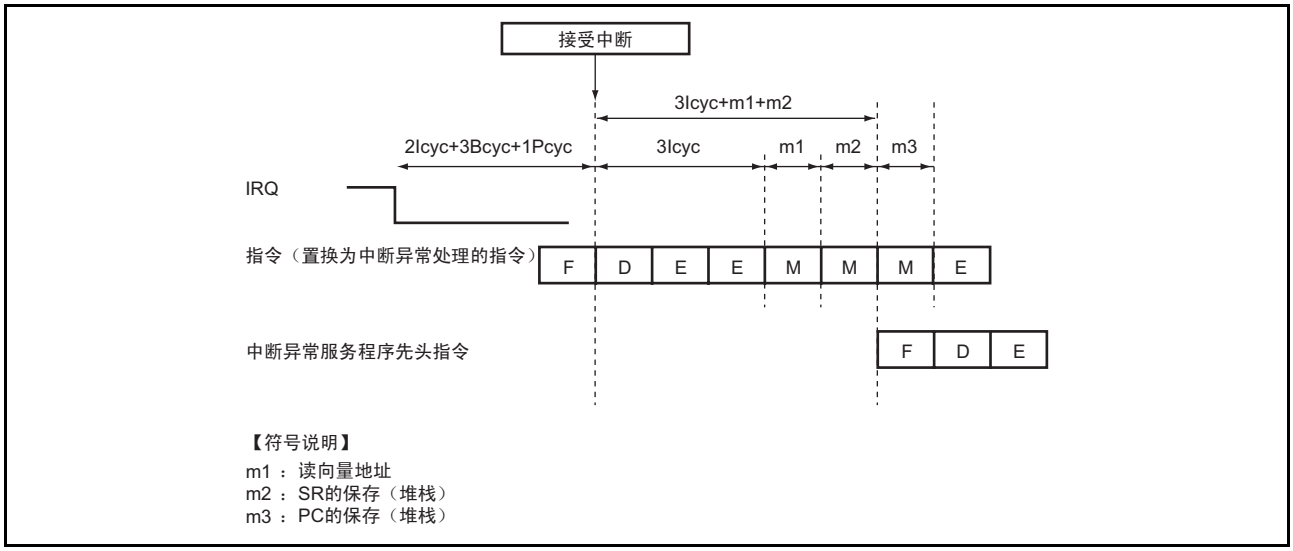


图 6.6　接受 IRQ 中断时的流水线运行例 （有寄存器组，无寄存器组上溢）

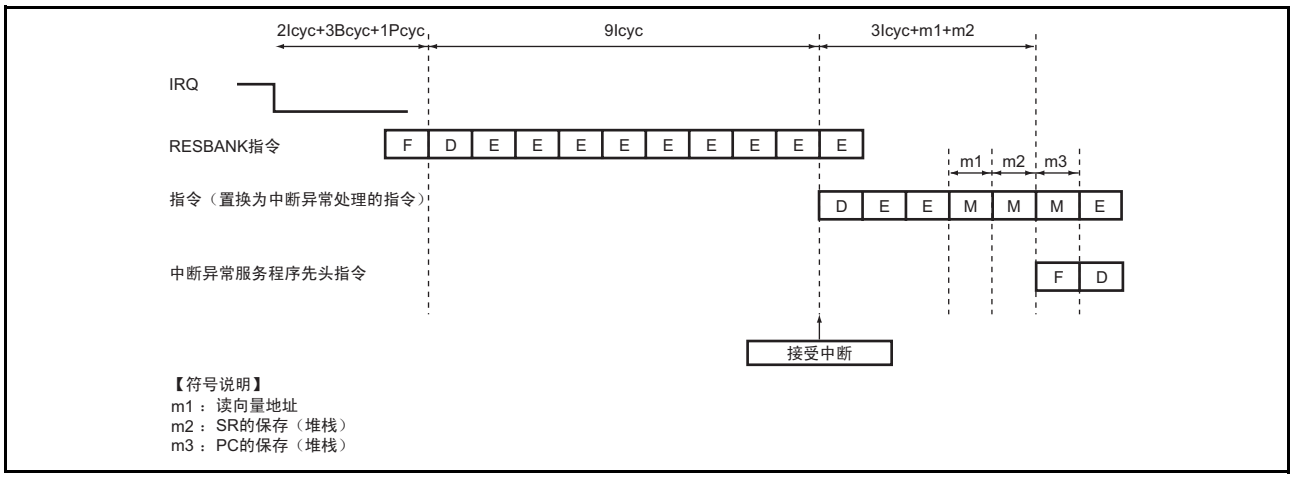


图 6.7　RESBANK 指令时接受中断时的流水线运行例 （有寄存器组，无寄存器组上溢）

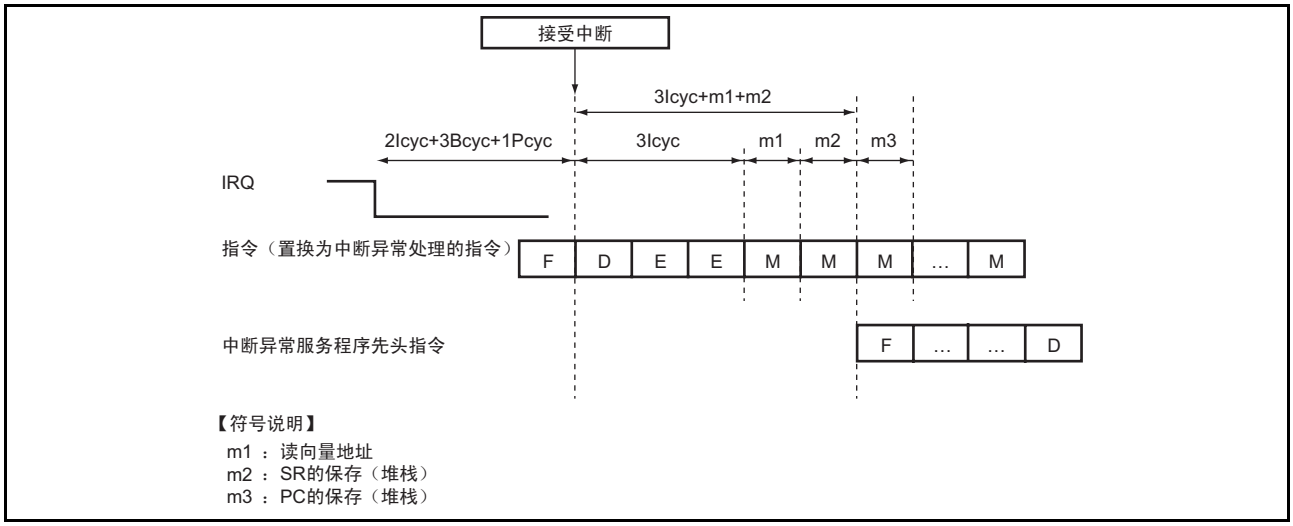


图 6.8　接受 IRQ 中断时的流水线运行例 （有寄存器组，有寄存器组上溢）

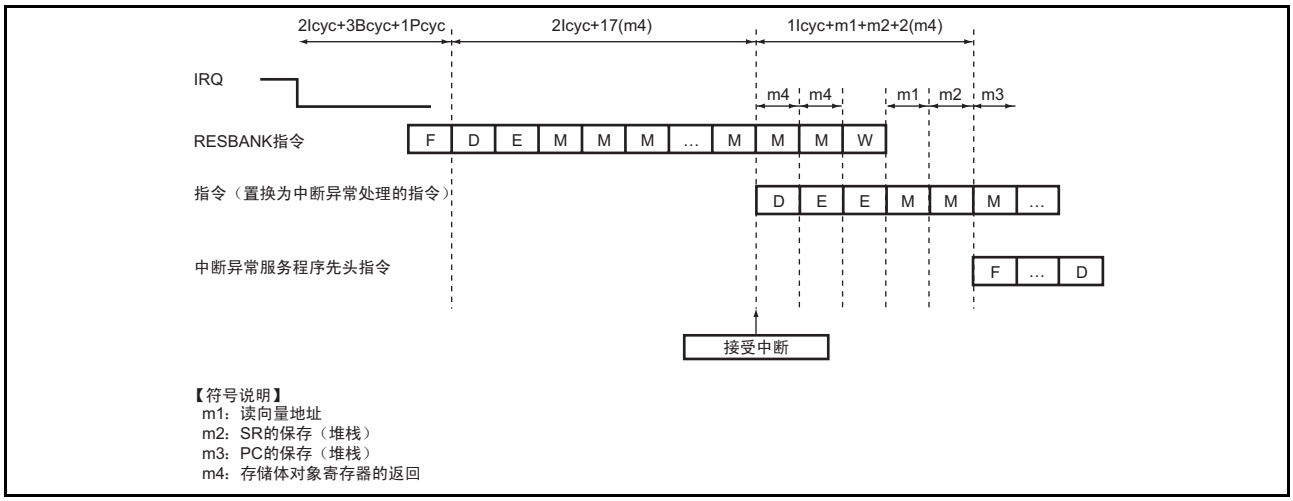


图 6.9　RESBANK 指令时接受中断时的流水线运行例 （有寄存器组，有寄存器组上溢）

6.8　寄存器组

本 LSI 内置有在进行中断处理的同时高速进行寄存器的保存与返回的 15 个寄存器组。寄存器组的结构如图 6.10 所示。

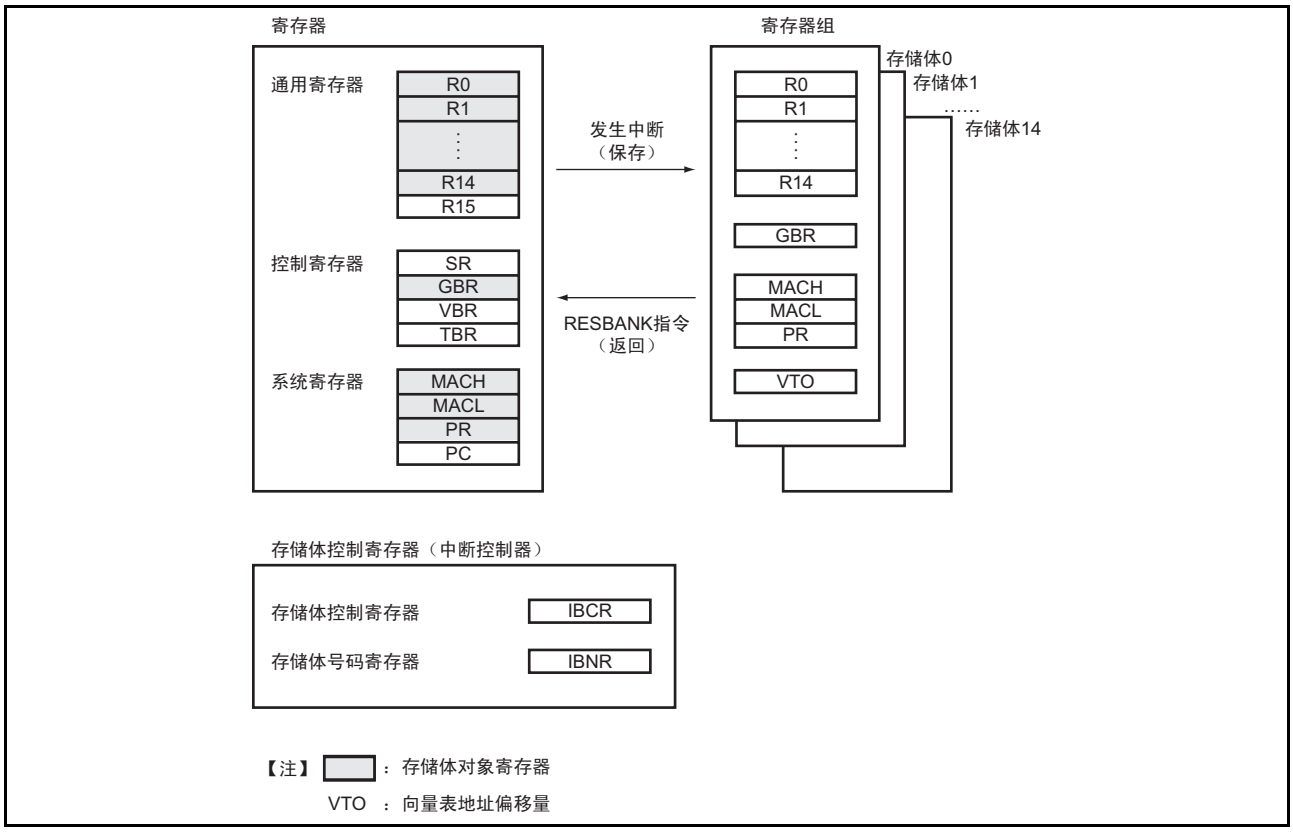


图 6.10　寄存器组结构的概要

6.8.1　存储体的对象寄存器和输入 / 输出方式

（1） 存储体的对象寄存器

将通用寄存器 （R0 ～ R14）、全局基址寄存器 （GBR）、乘加寄存器 （MACH， MACL）、过程寄存器 （PR）和向量表地址偏移量作为存储体的对象。

（2） 存储体的输入 / 输出方式

寄存器组有 15 个存储体 （存储体 0~ 存储体 14）。寄存器组为先进后出 （FILO）式的堆栈，保存从存储体 0 开始按顺序进行，返回从最后保存的存储体开始进行。

6.8.2　存储体保存、返回的运行

（1） 向存储体的保存

向寄存器组的保存的运行如图 6.11 所示。产生中断并允许 CPU 所接受的中断使用寄存器组时，运行如下：

（a）假设产生中断前的存储体编号寄存器 （IBNR）的存储体编号位 （BN）的值为 i。

（b）将寄存器 R0 ～ R14、 GBR、 MACH、 MACL、 PR 和被接受中断的向量表地址偏移量 （VTO）保存到 BN 所示的存储体 i。

（c） BN 的值 +1。

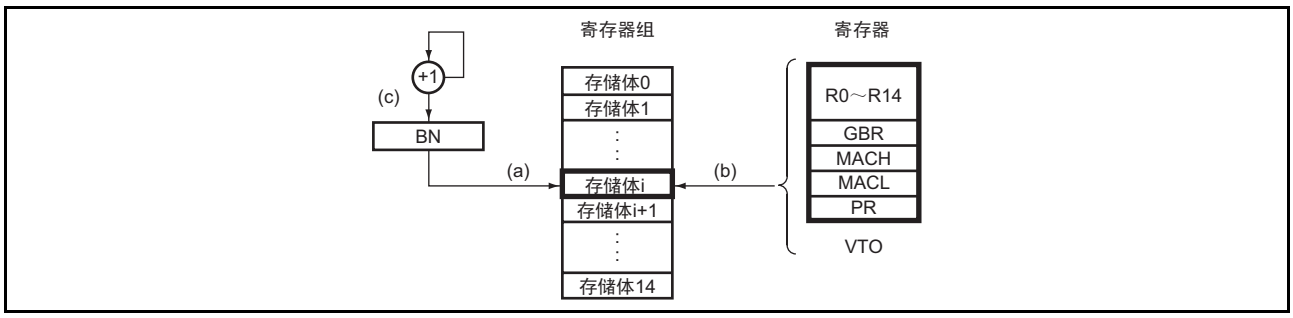


图 6.11　存储体保存的运行

寄存器组保存的时序如图 6.12 所示。从开始中断异常处理到开始取中断异常服务程序的第一条指令期间，执行向寄存器组的保存。

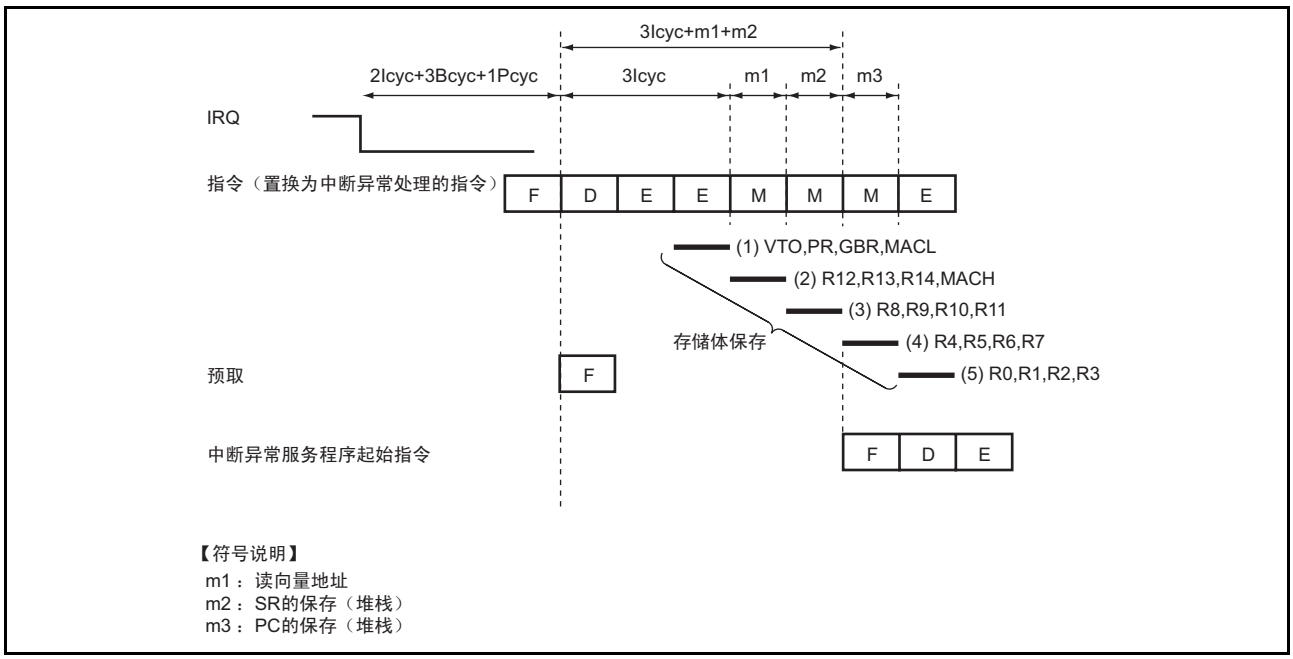


图 6.12　存储体保存的时序

（2） 从存储体的返回

使用存储体返回指令 RESBANK 返回保存到存储体的数据。在中断异常服务程序的最后，通过 RESBANK 指令进行存储体返回后，请通过 RTE 指令进行从中断异常服务程序的返回。

6.8.3　在保存至所有存储体的状态下进行的保存、返回

在保存到寄存器组所有存储体的状态下，产生中断，并允许 CPU 所接受的中断使用寄存器组时，如果存储体编号寄存器（IBNR）的 BOVE 位为 0，就代替寄存器组自动保存到堆栈。IBNR 的 BOVE 位置 1 时，产生寄存器组上溢异常，但不进行向堆栈的保存。

使用堆栈时保存、返回的运行如下：

（1） 向堆栈的保存

1. 中断异常处理时，对状态寄存器 （SR）、程序计数器 （PC）保存到堆栈。
2. 将存储体对象寄存器 （R0～R14、 GBR、 MACH、 MACL、 PR）保存到堆栈。保存到堆栈的寄存器顺序为：MACL、 MACH、 GBR、 PR、 R14、 R13、……、 R1、 R0
3. 将SR的寄存器组上溢位 （BO）置1。
4. 存储体编号寄存器 （IBNR）的存储体编号位 （BN）保持为最大值15。

（2） 从堆栈的返回

在 SR 的寄存器组上溢位 （BO）置 1 的状态下，执行存储体返回指令 RESBANK 时，运行如下：

1. 将存储体对象寄存器 （R0～R14、 GBR、 MACH、 MACL、 PR）从堆栈返回。从堆栈返回的寄存器顺序为：R0、 R1、……、 R13、 R14、 PR、 GBR、 MACH、 MACL。
2. 存储体编号寄存器 （IBNR）的存储体编号位 （BN）保持为最大值15。

6.8.4　寄存器组异常

寄存器组异常 （寄存器组错误）有寄存器组上溢和寄存器组下溢 2 种。

（1） 寄存器组上溢

在保存到寄存器组所有存储体的状态下，产生中断，并允许 CPU 所接受的中断使用寄存器组时，如果将存储体编号寄存器 （IBNR）的 BOVE 位置 1，就产生此异常。此时，存储体编号寄存器 （IBNR）的存储体编号位 （BN）保持为存储体数 15，不进行向寄存器组的保存。

（2） 寄存器组下溢

在完全没有向寄存器组进行保存的状态下，执行寄存器组返回指令时发生此异常。此时 R0 ～ R14、GBR、MACH、 MACL、 PR 的值不变。同时存储体号码寄存器 （IBNR）的存储体号码位 （BN）保持为 0 不变。

6.8.5　寄存器组错误异常处理

发生寄存器组错误时，开始寄存器组错误异常处理。此时 CPU 的运行如下：

1. 从与发生寄存器组错误相对应的异常处理向量表中取出异常服务程序的起始地址。
2. 将状态寄存器 （SR）保存到堆栈。
3. 将程序计数器 （PC）保存到堆栈。寄存器组上溢时保存的 PC 值为已执行的最后一条指令的下一条指令的起始地址。寄存器组下溢时保存的PC值为该RESBANK指令的起始地址。同时，为了防止产生寄存器组上溢时的多重中断，将成为寄存器存储体上溢源的中断级写入状态寄存器（SR）的中断屏蔽级位 （I3～I0）。
4. 从异常服务程序的起始地址开始执行程序。

6.9　通过中断请求信号的数据传送

能通过中断请求信号启动 DMAC 传送数据。

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

屏蔽条件 =DME・（DE0・源选择 0+DE1・源选择 1+DE2・源选择 2+DE3・源选择 3+DE4・源选择 4+DE5・源选择 5+DE6・源选择 6+DE7・源选择 7）

中断控制框图如图 6.13 所示。

在这里 DME 为 DMAC 的 DMAOR 的 bit0，DEn（n=0 ～ 7）为 DMAC 的 CHCR0 ～ CHCR7 的 bit0。详细请参照 “第 10 章 直接存储器存取控制器 （DMAC）”。

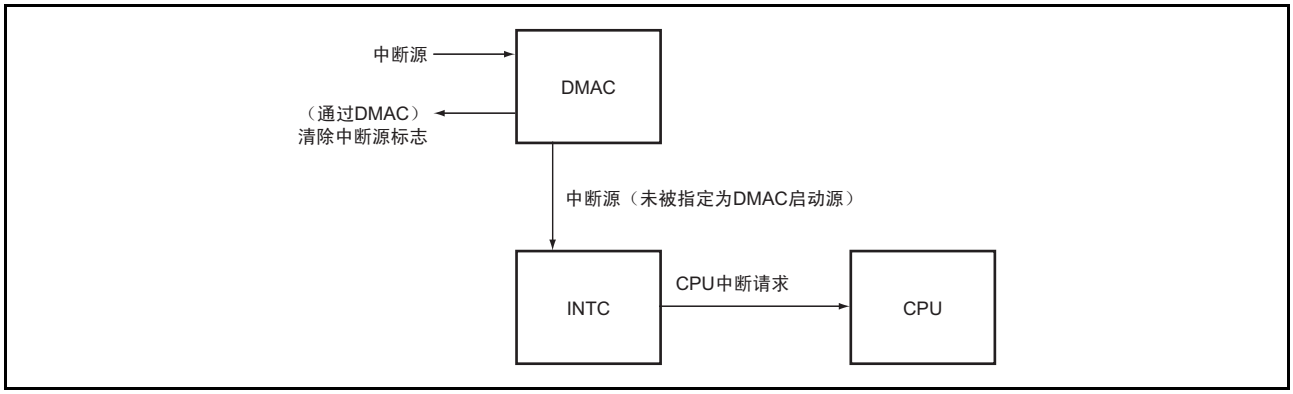


图 6.13　中断控制框图

6.9.1　中断请求信号作为 CPU 中断源而不作为 DMAC 启动源的情况

1. 不选择DMAC的启动源，或者将DME位清0。在选择DMAC的启动源时，将DMAC对应通道的DE位清0。
2. 产生中断时，向CPU请求中断。
3. 在中断异常服务程序， CPU清除中断源并进行必要的处理。

6.9.2　中断请求信号作为 DMAC 启动源而不作为 CPU 中断源的情况

1. 选择DMAC的启动源，设定DE=1、DME=1。与中断优先级设定寄存器的设定无关，屏蔽CPU中断源。
2. 产生中断时，将启动源传送给DMAC。
3. 在传送时， DMAC清除启动源。

6.10　使用注意事项

6.10.1　清除中断源的时序

请在中断异常服务程序中清除中断源标志。从清除中断源标志到实际取消对 CPU 的中断源，需要**表** 6.5 记载的“从产生中断请求，到中断控制器判定优先级以及与 SR 的屏蔽位比较后，向 CPU 发出中断请求信号的时间”。因此为了避免错误地再次接受应该已被清除的中断源，请在清除后读取 * 中断源标志，然后再执行 RTE 执令。

【注】 *　清除 USB 的中断源标志时，请在清除后读取 3 次中断源标志。

6.10.2　$\overline{\text{IRQOUT}}$ 的取反时序

中断控制器接受中断请求时，$\overline{\text{IRQOUT}}$ 引脚输出低电平，转移到中断异常服务程序的起始地后，$\overline{\text{IRQOUT}}$ 引脚输出高电平。

但是，中断控制器接受中断请求，从 $\overline{\text{IRQOUT}}$ 引脚输出低电平后，如果在转移到中断异常服务程序的起始地址之前取消中断请求，$\overline{\text{IRQOUT}}$ 引脚持续输出低电平直到转移到下一个中断请求的中断异常服务程序的起始地址。

第 7 章　　用户断点控制器 （UBC）

用户断点控制器 （UBC）提供程序调试功能。通过使用此功能，可轻松建立自监控调试程序，即使不使用在线仿真器，也能通过本 LSI 简单地调试程序。作为能给 UBC 设定的断点条件，支持取指令或者数据的读写 （读 / 写数据时为总线周期（CPU，DMAC））、数据长度、数据内容、地址值以及取指令时的停止时序。由于本 LSI 采用哈佛体系结构，通过对取指令总线 （F 总线）发送总线周期，执行 CPU 总线 （C 总线）上的取指令；通过对存储器存取总线 （M 总线）发送总线周期，执行 C 总线上的数据存取。同时，内部总线 （I 总线）有内部 CPU 总线和内部 DMA 总线，CPU 对内部 CPU 总线、DMA 对内部 DMA 总线发送总线周期。UBC 监控这些 C 总线和 I 总线。

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 0 和 1）
   能将用户断点独立设定为通道 0、 1。
   - 地址
     能按位屏蔽 32 位 （地址）的比较。
     能选择 4 种地址总线 （F 总线地址 （FAB）、 M 总线地址 （MAB）、内部 CPU 总线地址 （ICAB）、内部 DMA 总线地址 （IDAB））。
   - 数据
     能按位屏蔽 32 位 （数据）的比较。
     能选择 3 种数据总线 （M 总线数据 （MDB）、内部 CPU 总线数据 （ICDB）、内部 DMA 总线数据 （IDDB））中的一种。
   - 选择 I 总线时的总线选择
     内部 CPU 总线、内部 DMA 总线
   - 总线周期
     取指令 （仅限选择 C 总线时）或者存取数据
   - 读取或者写入
   - 操作数长度
     支持字节、字和长字。
2. 在取指令周期中，能将开始执行用户断点中断异常处理的时间设定在指令执行前或在指令执行后。
3. 断点条件成立时能对 $\overline{\text{UBCTRG}}$ 引脚输出信号。

UBC 的框图如**图 7.1** 所示。

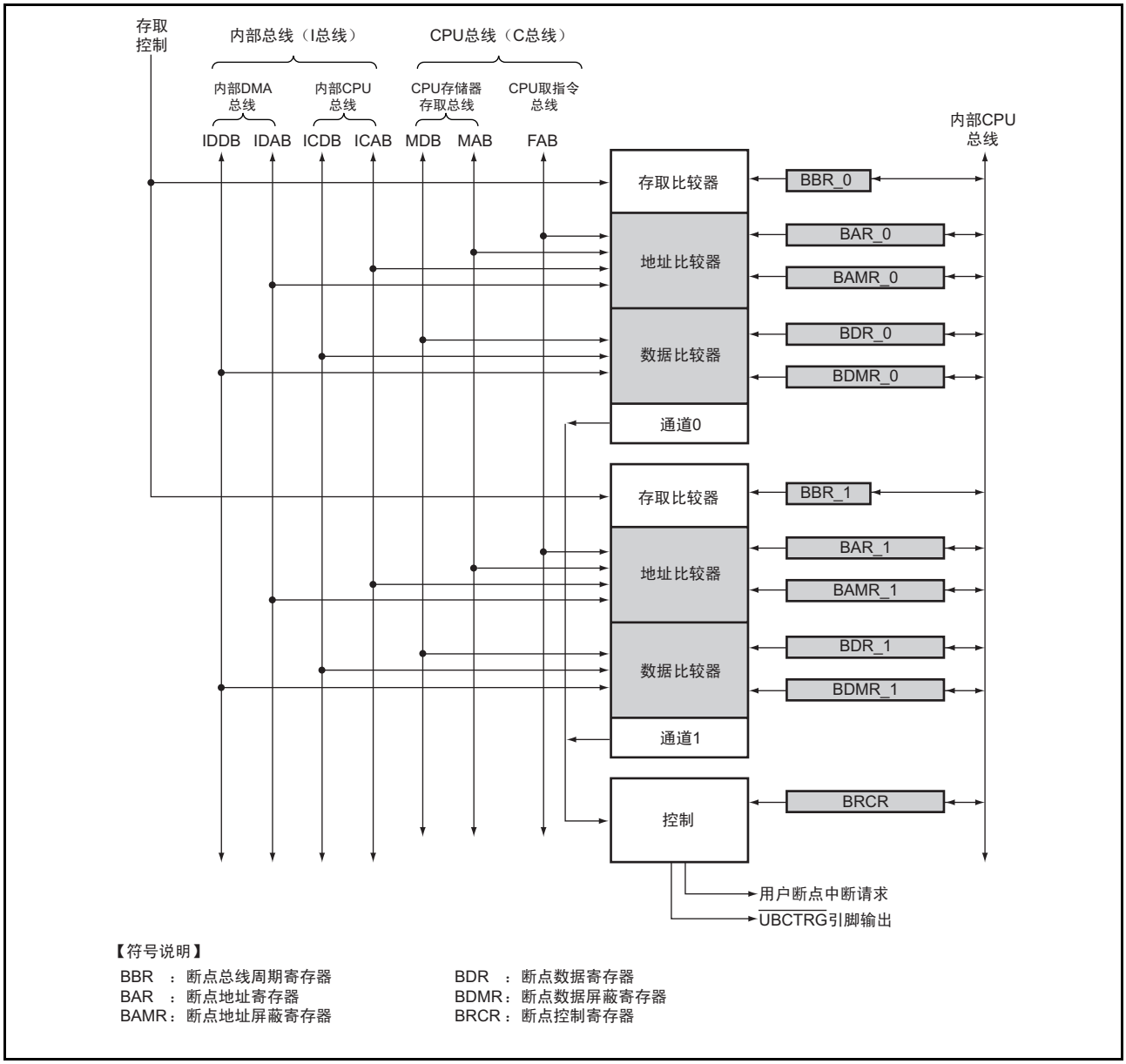

存取
控制
内部总线（I总线）
CPU总线（C总线）
内部DMA
总线
内部CPU
总线
CPU存储器
存取总线
CPU取指令
总线
IDDB
IDAB
ICDB
ICAB
MDB
MAB
FAB
内部CPU
总线
存取比较器
地址比较器
数据比较器
通道0
BBR_0
BAR_0
BAMR_0
BDR_0
BDMR_0
存取比较器
地址比较器
数据比较器
通道1
BBR_1
BAR_1
BAMR_1
BDR_1
BDMR_1
控制
BRCR
用户断点中断请求
UBCTRG引脚输出
【符号说明】
BBR ：断点总线周期寄存器
BAR ：断点地址寄存器
BAMR：断点地址屏蔽寄存器
BDR ：断点数据寄存器
BDMR：断点数据屏蔽寄存器
BRCR：断点控制寄存器

图 7.1　UBC 的框图

7.2　输入 / 输出引脚

UBC 的引脚结构如表 7.1 所示。

表 7.1　引脚结构

名称	引脚名	输入 / 输出	功能
UBC 触发	$\overline{\text{UBCTRG}}$	输出	表示在 UBC 的通道 0 或者 1，设定条件成立

7.3　寄存器说明

UBC 有以下寄存器。每通道 5 个控制寄存器和 1 个通道 0、1 共有的控制寄存器。关于各通道的寄存器，例如通道 0 的 BAR 表示为 BAR_0。

表 7.2　寄存器构成

通道	寄存器名	简称	R/W	初始值	地址	存取长度
0	断点地址寄存器 _0	BAR_0	R/W	H'00000000	H'FFFC0400	32
	断点地址屏蔽寄存器 _0	BAMR_0	R/W	H'00000000	H'FFFC0404	32
	断点总线周期寄存器 _0	BBR_0	R/W	H'0000	H'FFFC04A0	16
	断点数据寄存器 _0	BDR_0	R/W	H'00000000	H'FFFC0408	32
	断点数据屏蔽寄存器 _0	BDMR_0	R/W	H'00000000	H'FFFC040C	32
1	断点地址寄存器 _1	BAR_1	R/W	H'00000000	H'FFFC0410	32
	断点地址屏蔽寄存器 _1	BAMR_1	R/W	H'00000000	H'FFFC0414	32
	断点总线周期寄存器 _1	BBR_1	R/W	H'0000	H'FFFC04B0	16
	断点数据寄存器 _1	BDR_1	R/W	H'00000000	H'FFFC0418	32
	断点数据屏蔽寄存器 _1	BDMR_1	R/W	H'00000000	H'FFFC041C	32
共用	断点控制寄存器	BRCR	R/W	H'00000000	H'FFFC04C0	32

7.3.1　　断点地址寄存器 （BAR）

BAR 为 32 位可读写寄存器。BAR 指定作为各通道断点条件的地址。断点条件对象的地址总线有 4 种，通过断点总线周期寄存器 （BBR）的控制位的 CD[1:0] 和 CP[1:0] 选择。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24	BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
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
	BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8	BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
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	BA31 ～ BA0	均为 0	R/W	断点地址 保存指定断点条件的 CPU 地址总线（FAB 或者 MAB）或者内部地址总线（ICAB 或者 IDAB 的）的地址。 通过 BBR 选择 C 总线和取指令周期时，对 BA31 ～ BA0 指定 FAB 的地址。 通过 BBR 选择 C 总线和数据存取周期时，对 BA31 ～ BA0 指定 MAB 的地址。 通过 BBR 选择 I 总线和内部 CPU 总线时，对 BA31 ～ BA0 指定 ICAB 的地址。 通过 BBR 选择 I 总线和内部 DMA 总线时，对 BA31 ～ BA0 指定 IDAB 的地址。

【注】 作为断点条件设定取指令周期时，请将 BAR 的 LSB 清 0。

7.3.2　断点地址屏蔽寄存器 （BAMR）

BAMR 为 32 位可读 / 写寄存器。在 BAR 指定的断点地址位中， BAMR 指定要屏蔽的位。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAM31	BAM30	BAM29	BAM28	BAM27	BAM26	BAM25	BAM24	BAM23	BAM22	BAM21	BAM20	BAM19	BAM18	BAM17	BAM16
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
	BAM15	BAM14	BAM13	BAM12	BAM11	BAM10	BAM9	BAM8	BAM7	BAM6	BAM5	BAM4	BAM3	BAM2	BAM1	BAM0
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	BAM31 ～ BAM0	均为 0	R/W	断点地址屏蔽 在 BAR （BA31 ～ BA0）指定的断点地址位中，指定要屏蔽的位。 0：断点地址位 BAn 包含在断点条件中 1：断点地址位 BAn 被屏蔽，不包含在断点条件中 【注】 n=31 ～ 0

7.3.3　断点数据寄存器 （BDR）

BDR 为 32 位可读写寄存器。作为断点条件对象的数据总线有 3 种，通过断点总线周期寄存器 （BBR）的控制位 CD[1:0] 和 CP[1:0] 选择。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BD31	BD30	BD29	BD28	BD27	BD26	BD25	BD24	BD23	BD22	BD21	BD20	BD19	BD18	BD17	BD16
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
	BD15	BD14	BD13	BD12	BD11	BD10	BD9	BD8	BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0
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	BD31 ～ BD0	均为 0	R/W	断点数据位 保存指定断点条件的数据。 通过 BBR 选择 C 总线时，对 BD31 ～ BD0 指定 MDB 的断点数据。 通过 BBR 选择 I 总线和内部 CPU 总线时，对 BD31 ～ BD0 指定 ICDB 的地址。 通过 BBR 选择 I 总线和内部 DMA 总线时，对 BD31 ～ BD0 指定 IDDB 的地址。

【注】 1. 在断点条件中包含数据总线的值时，请指定操作数长度。

2. 指定字节大小为断点条件时，请对 bit31 ～ 24、 23 ～ 16、 15 ～ 8、 7 ～ 0 设定相同的字节数据作为 BDR 的断点数据。同样在指定字大小时，也请对 bit31 ～ 16、 15 ～ 0 设定同样的字数据。

7.3.4　断点数据屏蔽寄存器 （BDMR）

BDMR 为 32 位可读 / 写寄存器。在 BDR 指定的断点数据位中， BDMR 指定要屏蔽的位。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BDM31	BDM30	BDM29	BDM28	BDM27	BDM26	BDM25	BDM24	BDM23	BDM22	BDM21	BDM20	BDM19	BDM18	BDM17	BDM16
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
	BDM15	BDM14	BDM13	BDM12	BDM11	BDM10	BDM9	BDM8	BDM7	BDM6	BDM5	BDM4	BDM3	BDM2	BDM1	BDM0
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	BDM31 ～ BDM0	均为 0	R/W	断点数据屏蔽 在 BDR （BD31 ～ BD0）指定的断点数据位中，指定要屏蔽的位。 0：断点数据位 BDn 包含在断点条件中 1：断点数据位 BDn 被屏蔽，不包含在断点条件中 【注】 n=31 ～ 0

【注】 1. 在断点条件中包含数据总线的值时，请指定操作数长度。

2. 指定字节大小为断点条件时，请对 bit31 ～ 24、 23 ～ 16、 15 ～ 8、 7 ～ 0 设定相同的字节数据作为 BDMR 的断点屏蔽数据，同样在指定字大小时，请对 bit31 ～ 16、 15 ～ 0 设定同样的字数据。

7.3.5　断点总线周期寄存器 （BBR）

BBR 是 16 位可读 / 写寄存器，指定（1）禁止 / 允许用户断点中断请求、（2）包含 / 不包含数据总线的值、（3）内部 CPU 总线或者内部 DMA 总线、（4） C 总线周期或者 I 总线周期、（5）取指令或者数据存取、（6）读取或者写入、（7）操作数长度为断点条件。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	UBID	DBE	-	-	CP[1:0]		CD[1:0]		ID[1:0]		RW[1:0]		SZ[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/W	R/W	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、 14	—	均为 0	R	保留位 读写的值总是 0。
13	UBID	0	R/W	用户断点中断禁止 指定条件匹配时禁止 / 允许用户断点中断请求。 0：允许用户断点中断请求 1：禁止用户断点中断请求
12	DBE	0	R/W	数据断点允许 选择断点条件中是否包含数据总线条件。 0：断点条件中不包含数据总线条件 1：断点条件中包含数据总线条件
11、 10	—	均为 0	R	保留位 读写的值总是 0。
9、 8	CP[1:0]	00	R/W	I 总线的总线选择 选择断点条件的总线周期是 I 总线时的总线。但是，总线周期为 C 总线周期时，本位无效 （仅限 CPU 周期）。 00：不进行条件比较 01：断点条件为内部 CPU 总线 10：断点条件为内部 DMA 总线 11：断点条件为内部 CPU 总线
7、 6	CD[1:0]	00	R/W	C 总线周期 /I 总线周期的选择 选择 C 总线周期或者 I 总线周期为断点条件的总线周期。 00：不进行条件比较 01：断点条件为 C 总线 （F 总线、 M 总线）周期 10：断点条件为 I 总线周期 11：断点条件为 C 总线 （F 总线、 M 总线）周期
5、 4	ID[1:0]	00	R/W	取指令 / 数据存取的选择 选择取指令周期或者数据存取周期为断点条件的总线周期。 选择取指令周期时请选择 C 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

位	位名	初始值	R/W	说明
3、 2	RW[1:0]	00	R/W	读 / 写的选择 选择读周期或者写周期为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：断点条件为读周期或者写周期
1、 0	SZ[1:0]	00	R/W	操作数长度的选择 选择断点条件总线周期的操作数长度。 00：断点条件不含操作数长度 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

7.3.6　断点控制寄存器 （BRCR）

BRCR 设定如下条件：

1. 指定通过取指令周期开始执行用户断点中断异常处理的时间是设定在指令执行前还是设定在指令执行后。
2. 设定断点条件匹配时的$\overline{\text{UBCTRG}}$输出的脉冲宽度。
3. 设定断点条件匹配时是否进行$\overline{\text{UBCTRG}}$输出。

BRCR 是 32 位可读写寄存器，有设定断点条件匹配标志和其他断点条件的位。因为只有 bit15 ～ 12 的条件匹配标志为写 1 无效 （保持之前的值）而只能写 0 的位，所以清除时请对要清除的标志位写 0，对其以外的标志位写 1。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	UTOD1	UTOD0	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	R	R	R	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
	SCMFC0	SCMFC1	SCMFD0	SCMFD1	-	-	-	-	-	PCB1	PCB0	-	-	-	-	-
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/W	R/W	R	R	R	R	R

位	位名	初始值	R/W	说明
31 ～ 20	—	均为 0	R	保留位 读写的值总是 0。
19	UTOD1	0	R/W	$\overline{\text{UBCTRG}}$ 输出禁止 1 指定通道 1 的条件匹配时是否进行 $\overline{\text{UBCTRG}}$ 输出。 0：通道 1 的条件匹配时向 $\overline{\text{UBCTRG}}$ 输出 1：通道 1 的条件匹配时不向 $\overline{\text{UBCTRG}}$ 输出
18	UTOD0	0	R/W	$\overline{\text{UBCTRG}}$ 输出禁止 0 指定通道 0 的条件匹配时是否进行 $\overline{\text{UBCTRG}}$ 输出。 0：通道 0 的条件匹配时向 $\overline{\text{UBCTRG}}$ 输出 1：通道 0 的条件匹配时不向 $\overline{\text{UBCTRG}}$ 输出

位	位名	初始值	R/W	说明
17、16	CKS[1:0]	00	R/W	时钟选择 指定条件匹配时对 $\overline{\text{UBCTRG}}$ 输出的脉冲宽度。 00：$\overline{\text{UBCTRG}}$ 的脉冲宽度为 1 个总线时钟周期 01：$\overline{\text{UBCTRG}}$ 的脉冲宽度为 2 个总线时钟周期 10：$\overline{\text{UBCTRG}}$ 的脉冲宽度为 4 个总线时钟周期 11：$\overline{\text{UBCTRG}}$ 的脉冲宽度为 8 个总线时钟周期
15	SCMFC0	0	R/W	C 总线周期条件匹配标志 0 当满足给通道 0 设定的断点条件 （C 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 0 的 C 总线周期条件不匹配 1：通道 0 的 C 总线周期条件匹配
14	SCMFC1	0	R/W	C 总线周期条件匹配标志 1 当满足给通道 1 设定的断点条件 （C 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 1 的 C 总线周期条件不匹配 1：通道 1 的 C 总线周期条件匹配
13	SCMFD0	0	R/W	I 总线周期条件匹配标志 0 当满足给通道 0 设定的断点条件 （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 0 的 I 总线周期条件不匹配 1：通道 0 的 I 总线周期条件匹配
12	SCMFD1	0	R/W	I 总线周期条件匹配标志 1 当满足给通道 1 设定的断点条件 （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 1 的 I 总线周期条件不匹配 1：通道 1 的 I 总线周期条件匹配
11 ～ 7	—	均为 0	R	保留位 读写的值总是 0。
6	PCB1	0	R/W	PC 断点选择 1 选择通道 1 取指令周期的断点时序是在指令执行前还是指令执行后。 0：将通道 1 的 PC 断点设定到指令执行前 1：将通道 1 的 PC 断点设定到指令执行后
5	PCB0	0	R/W	PC 断点选择 0 选择通道 0 取指令周期的断点时序是在指令执行前还是指令执行后。 0：将通道 0 的 PC 断点设定到指令执行前 1：将通道 0 的 PC 断点设定到指令执行后
4 ～ 0	—	均为 0	R	保留位 读写的值总是 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中断异常处理的运行流程如下：

1. 将断点地址指定到断点地址寄存器（BAR）。将要屏蔽的地址指定到断点地址屏蔽寄存器（BAMR）。将断点数据指定到断点数据寄存器（BDR）。将要屏蔽的数据指定到断点数据屏蔽寄存器（BDMR）。将总线断点条件指定到断点总线周期寄存器 （BBR）。 BBR 的 3 对控制位 （即 C 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 对为 00 时，就不发生用户断点暂停。在 BRCR 的位指定断点控制。必须在设定全部断点相关寄存器后进行 BBR 的设定，并在读取最后写入的寄存器后发生转移。从转移目标的指令新写入寄存器的值有效。
2. 当满足断点条件并且设定允许用户断点中断请求时，UBC 在将用户断点中断请求通知给 INTC 的同时，设置各通道的 C 总线条件匹配标志 （SCMFC）和 I 总线条件匹配标志 （SCMFD），并且按照 CKS[1:0] 位设定的脉冲宽度对 $\overline{\text{UBCTRG}}$ 引脚进行输出。同时能通过将 BBR 的 UBID 位置 1，在外部监控触发输出而不请求用户断点中断。
3. 接受用户断点中断请求信号时，INTC 进行优先级判定。用户断点中断为优先级 15，因此在状态寄存器（SR）的中断屏蔽级位 （I3 ～ I0）不超过优先级 14 时，接受该中断。 I3 ～ I0 位为优先级 15 时，不接受用户断点中断，但是进行条件判定，如果匹配，条件匹配标志就被置位。关于优先级判定，请参照“**第 6 章中断控制器 （INTC）**”。
4. 能使用对应的条件匹配标志 （SCMFC、 SCMFD）检查设定条件是否匹配。请在用户断点中断异常处理程序中进行清除条件匹配标志位的运行。如果不进行该运行就会再次产生中断。
5. 有时几乎同时发生通道 0 和通道 1 设定的断点，此时即使对 INTC 的用户断点中断请求只有 1 个，也有可能设置 2 个断点通道匹配标志。
6. 选择 I 总线为断点条件时，请注意如下事项：
   - 高速缓存的设定决定 CPU 在 C 总线上发送的存取是否也在内部 CPU 总线上发送。关于高速缓存条件下的 I 总线运行，请参照 “**第 8 章高速缓存**”的**表 8.8**。
   - 对 I 总线设定断点条件时只监视数据存取周期，不监视取指令周期 （含高速缓存更新周期）。
   - 内部 DMA 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 执行指令引起的内部 CPU 总线周期中条件匹配，也不能决定是在哪条指令接受用户断点中断请求。

7.4.2　取指令周期的断点

1. 如果将 “C 总线 / 取指令 / 读取 / 字或者长字”设定到断点总线周期寄存器 （BBR），断点条件就为 FAB 总线的取指令。能通过对应通道的断点控制寄存器 （BRCR）的 PCB0 或者 PCB1 位，选择是在指令执行前还是指令执行后开始执行用户断点中断异常处理。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 （BAR）的 BA0 位清零。如果此位被置 1，就不发生断点。
2. 在将取指令的断点设定在执行该指令前的状态下，如果条件匹配就取出指令，然后在确定执行指令的时序发生断点。因此，通过溢出 （在转移或者中断转换中取出但不执行的指令）所取得的指令不发生断点。如果对延迟转移指令的延迟槽设定此类断点条件，到转移目标的首条指令执行前不接受用户断点中断请求。

   【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断点设定在指令执行后，就在执行和断点条件匹配的指令后并在执行下一条指令前发生断点。和执行前的断点情况一样，预取指令不发生断点。如果对延迟转移指令及其延迟槽设定这种断点条件，到转移目标的首条指令执行前不接受用户断点中断请求。
4. 如果设定取指令周期，就忽视断点数据寄存器 （BDR）。因此，不能给取指令周期的断点设定断点数据。
5. 给取指令周期的断点设定 I 总线时无效。

7.4.3　数据存取周期的断点

1. 对数据存取断点，如果将C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断点暂停；如果将I总线指定为断点条件，就对总线 （由I总线选择指定）上的数据存取周期的地址 （和数据）进行条件比较并发生断点暂停。有关在内部CPU总线上发送的CPU总线周期，请参照 “7.4.1用户断点运行的流程”的第6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各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 （BAR）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 （BBR）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匹配时发生断点暂停。此时，为了指定字节数据，必须给断点数据寄存器（BDR）和断点数据屏蔽寄存器（BDMR）的 bit31 ～ 24、bit23 ～ 16、bit15 ～ 8、bit7 ～ 0 这 4 个字节指定相同的数据。为了指定字数据，请给 bit31 ～ 16、 bit15 ～ 0 这 2 个字指定相同的数据。
4. 将PREF指令作为无存取数据的长字节读出存取处理。因此，如果PREF指令的断点条件中包含数据值就不发生断点暂停。
5. 选择数据存取周期时，不能特别指定发生断点暂停的指令。

7.4.4　被保存的程序计数器值

接受用户断点中断请求时，将应该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 C 总线（FAB）/ 取指令指定为断点条件，就能决定发生断点暂停的指令。如果将 C 总线 / 数据存取周期或者 I 总线 / 数据存取周期指定为断点条件，就不能决定发生断点暂停的指令。

1. 将C总线 （FAB） /取指令 （指令执行前）指定为断点条件时
   将与断点条件匹配的指令地址保存到堆栈。不执行条件匹配的指令，而在执行前发生断点暂停。但是，如果延迟槽指令发生条件匹配，就执行这些指令，并将延迟转移指令的地址保存到堆栈。
2. 将C总线 （FAB） /取指令 （指令执行后）指定为断点条件时
   将与断点条件匹配指令的下一条指令地址保存到堆栈。执行条件匹配的指令，并在执行下一条指令前发生断点暂停。如果延迟转移指令或者延迟槽发生条件匹配，就执行这些指令，并将转移目标地址保存到堆栈。
3. 将C总线/数据存取周期或者I总线/数据存取周期指定为断点条件时
   将数条指令 （与断点条件匹配的指令）执行后的地址保存到堆栈。

7.4.5　使用例

（1） 指定为 C 总线取指令周期的断点条件

（例 1-1）

- 寄存器指定

  BAR_0=H'00000404、 BAMR_0=H'00000000、 BBR_0=H'0054、 BAR_1=H'00008010、
  BAMR_1=H'00000006、 BBR_1=H'0054、 BDR_1=H'00000000、 BDMR_1=H'00000000、
  BRCR=H'00000020

＜通道 0 ＞

地址：H'00000404、地址屏蔽：H'00000000

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

＜通道 1 ＞

地址：H'00008010、地址屏蔽：H'00000006

数据：H'00000000、数据屏蔽：H'00000000

总线周期：C 总线/取指令 （执行指令前） /读取 （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暂停。

（例 1-2）

- 寄存器指定

  BAR_0=H'00027128、 BAMR_0=H'00000000、 BBR_0=H'005A、 BAR_1=H'00031415、
  BAMR_1=H'00000000、 BBR_1=H'0054、 BDR_1=H'00000000、 BDMR_1=H'00000000、
  BRCR=H'00000000

＜通道 0 ＞

地址：H'00027128、地址屏蔽：H'00000000

总线周期：C 总线/取指令 （执行指令前） /写入/字

＜通道 1 ＞

地址：H'00031415、地址屏蔽：H'00000000

数据：H'00000000、数据屏蔽：H'00000000

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

在通道 0 中，因为取指令不是写周期，所以不发生用户断点暂停；在通道 1 中，因为对偶数地址取指令，所以也不发生用户断点暂停。

（例 1-3）

- 寄存器指定

  BAR_0=H'00008404、 BAMR_0=H'00000FFF、 BBR_0=H'0054、 BAR_1=H'00008010、
  BAMR_1=H'00000006、 BBR_1=H'0054、 BDR_1=H'00000000、 BDMR_1=H'00000000、
  BRCR=H'00000020

＜通道 0 ＞

地址：H'00008404、地址屏蔽：H'00000FFF

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

＜通道 1 ＞

地址：H'00008010、地址屏蔽：H'00000006

数据：H'00000000、数据屏蔽：H'00000000

总线周期：C 总线/取指令 （执行指令前） /读取 （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暂停。

（2） 指定为 C 总线数据存取周期的断点条件

（例 2-1）

- 寄存器指定

  BAR_0=H'00123456、 BAMR_0=H'00000000、 BBR_0=H'0064、 BAR_1=H'000ABCDE、 BAMR_1=H'000000FF、 BBR_1=H'106A、 BDR_1=H'A512A512、 BDMR_1=H'00000000、 BRCR=H'00000000

＜通道 0 ＞

地址：H'00123456、地址屏蔽：H'00000000

总线周期：C 总线/数据存取/读取 （断点条件不包含操作数长度）

＜通道 1 ＞

地址：H'000ABCDE、地址屏蔽：H'000000FF

数据：H'0000A512、数据屏蔽：H'00000000

总线周期：C 总线/数据存取/写入/字

在通道 0 中，在读取地址 H'00123456 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暂停；在通道 1 中，在给 H'000ABC00～H'000ABCFE 写字数据 H'A512 时发生用户断点暂停。

（3） 指定为 I 总线数据存取周期的断点条件

（例 3-1）

- 寄存器指定：

  BAR_0=H'00314156、 BAMR_0=H'00000000、 BBR_0=H'0194、 BAR_1=H'00055555、 BAMR_1=H'00000000、 BBR_1=H'12A9、 BDR_1=H'78787878、 BDMR_1=H'0F0F0F0F、 BRCR=H'00000000

＜通道 0 ＞

地址：H'00314156、地址屏蔽：H'00000000

总线周期：内部CPU 总线/取指令/读取 （断点条件不包含操作数长度）

＜通道 1 ＞

地址：H'00055555、地址屏蔽：H'00000000

数据：H'00000078、数据屏蔽：H'0000000F

总线周期：内部DMA 总线/数据存取/写入/字节

在通道 0 中，内部 CPU 总线上的取指令设定为无效。在通道 1 中，在内部 DMA 总线上 DMAC 将字节 H'7x 写到地址 H'00055555 时发生用户断点暂停 （内部CPU 总线的存取时不发生）。

7.5 使用时的注意事项

1. 通过内部CPU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断点暂停。为得知更改UBC 寄存器的时序，必须读取最后写入的寄存器。此后的指令对新写入的寄存器值有效。
2. UBC不能用相同的通道监视C总线周期、内部CPU总线周期及内部DMA总线周期。
3. 在同一指令发生用户断点中断请求和其他异常源时，按照“**第5章异常处理**”的**表5.1**所规定的优先级进行判定。发生更高优先级的异常源时，也不接受用户断点中断请求。
4. 在延迟槽中发生断点暂停时，有以下注意事项：
   给延迟槽指令设定指令执行前的断点时，在执行其转移目标前不接受用户断点中断请求。
5. 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。
6. 不能把放置中断优先级为15以上 （含用户断点中断）的中断异常处理程序的地址设定为断点地址。
7. 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。
8. 对32位指令设定断点地址时，请设定在前16位。断点地址设定在后16位时，即使作为断点条件设定在指令执行前也当作指令执行后处理。
9. 不能对DIVU、DIVS指令的下一条指令设定指令执行前断点。对DIVU、DIVS指令的下一条指令设定指令执行前断点时，在DIVU、 DIVS指令的执行过程中，就会发生异常或中断，并且即使DIVU、DIVS指令的执行被中断，也会发生下一条指令的指令执行前断点暂停。

第 8 章　高速缓存

8.1　特点

- 容量
  指令高速缓存：8K 字节
  操作数高速缓存：8K 字节
- 结构：指令/数据分离、 4路集联
- 锁定功能 （仅限操作数高速缓存）：可锁定路2、路3
- 线尺寸：16字节
- 入口数：128个入口
- 写入方式：可选择回写方式和直写方式
- 替换方式：LRU替换算法

8.1.1　高速缓存的结构

高速缓存是指令 / 数据分离型的 4 路集联方式。由 4 个路 （存储体）组成，每路被分成地址和数据。

每个地址和数据由 128 个入口组成。入口的数据被称为线。每 1 线是 16 字节 （4 字节 ×4）。每 1 路的数据容量是 2K 字节 （16 字节 ×128 个入口），高速缓存整体 （4 路）为 8K 字节的容量。

操作数高速缓存的结构如**图 8.1** 所示。指令高速缓存除没有 U 位外，结构与操作数高速缓存相同。

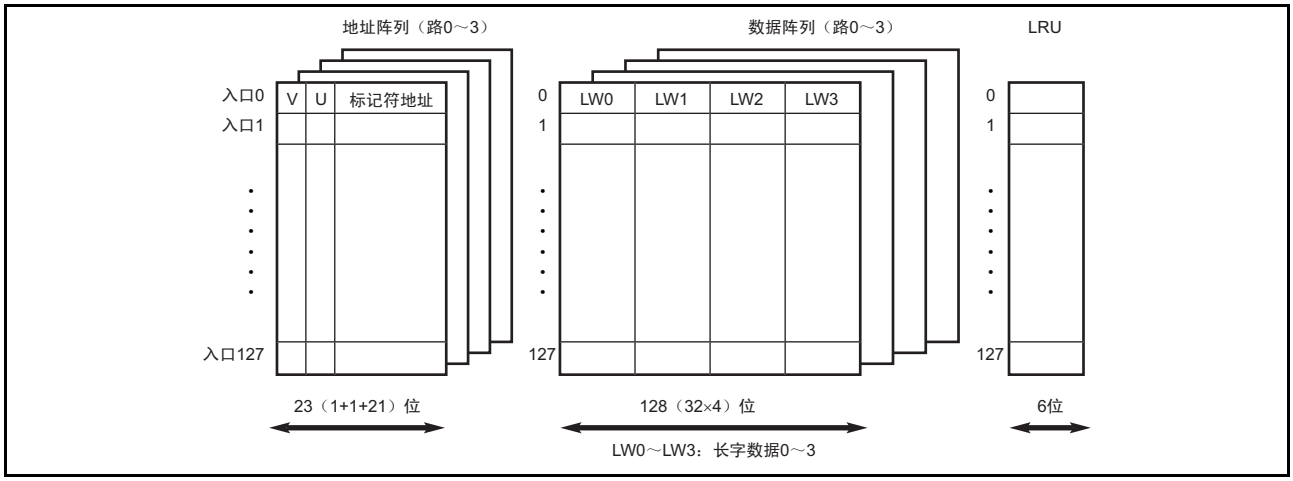


图 8.1　操作数高速缓存的结构

（1） 地址阵列

V 位表示入口的数据是否有效。 V 位为 1 表示有效，为 0 表示无效。

U 位 （仅限操作数高速缓存），表示在回写模式中该入口有写入。 U 位为 1 表示有写入，为 0 表示没有写入。

标记符地址保存用于外部存储器存取的地址。由高速缓存搜索时用于比较的 21 个位 （地址 31 ～ 11）组成。本 LSI 中，高速缓存有效空间的地址为 H'00000000 ～ H'1FFFFFFF （参照 “第 9 章 总线状态控制器（BSC）”），因此将标记符地址的高 3 位清 0。

V、 U 位通过上电复位初始化为 0，但不通过手动复位初始化，软件待机模式时也不被初始化。标记符地址不通过上电复位以及手动复位初始化，软件待机模式时也不被初始化。

（2） 数据阵列

保存 16 字节的指令或者数据。高速缓存入口的登录单位，按线单位 （16 字节单位）进行。

数据阵列不通过上电复位以及手动复位初始化，软件待机模式时也不被初始化。

（3） LRU

通过 4 路集联方式，最多可向高速缓存登录 4 条入口地址相同的指令或数据。登录入口时， LRU 位表示登录到 4 路中的哪一路。 LRU 位由 6 个位组成，通过硬件控制。作为路选择算法，使用 LRU （Least Recently Used）算法，该算法选择最近被存取的路。

通过 LRU 位 （6 个位）指定高速缓存未中时被替换的路。不使用高速缓存锁定功能 （仅限操作数高速缓存）时， LRU 位与被替换的路的关系如表 8.1 所示（使用高速缓存锁定功能时，请参照 “8.2.2 高速缓存控制寄存器 2（CCR2）”）。通过软件指定表 8.1 所示以外的 LRU 位时，高速缓存就无法正常运行。通过软件变更 LRU 位时，请设定如表 8.1 所示的模式。

LRU 位通过上电复位初始化为 B'000000，但不通过手动复位初始化，软件待机模式时也不被初始化。

表 8.1 LRU 位与被替换的路 （不使用高速缓存锁定功能时）

LRU （bit5 ～ 0）	被替换的路
000000、 000100、 010100、 100000、 110000、 110100	3
000001、 000011、 001011、 100001、 101001、 101011	2
000110、 000111、 001111、 010110、 011110、 011111	1
111000、 111001、 111011、 111100、 111110、 111111	0

8.2　寄存器的说明

高速缓存中有以下寄存器。

表 8.2　寄存器构成

寄存器名	简称	R/W	初始值	地址	存取长度
高速缓存控制寄存器 1	CCR1	R/W	H'00000000	H'FFFC1000	32
高速缓存控制寄存器 2	CCR2	R/W	H'00000000	H'FFFC1004	32

8.2.1　高速缓存控制寄存器 1 （CCR1）

通过 ICE 位指定指令高速缓存的允许或者禁止。ICF 位控制指令高速缓存所有入口的无效化。通过 OCE 位指定操作数高速缓存的允许或者禁止。OCF 位控制操作数高速缓存所有入口的无效化。WT 位切换操作数高速缓存的直写模式和回写模式。

请将变更 CCR1 内容的程序分配在高速缓存无效空间，读出 CCR1 的内容后请存取高速缓存有效空间。

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
	-	-	-	-	ICF	-	-	ICF	-	-	-	-	OCF	-	WT	OCE
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	R	R/W	R	R	R	R	R/W	R	R/W	R/W

位	位名	初始值	R/W	说　明
31 ～ 12	—	均为 0	R	保留位 读写的值总是为 0 。
11	ICF	0	R/W	指令高速缓存闪存 写入 1 时，指令高速缓存所有入口的 V 位和 LRU 位清 0（闪存）。读出时值为 0。闪存时不对外部存储器进行回写。
10、 9	—	均为 0	R	保留位 读写的值总是为 0 。
8	ICE	0	R/W	指令高速缓存有效 指定指令高速缓存功能的允许 / 禁止。 0：指令高速缓存禁止 1：指令高速缓存允许
7 ～ 4	—	均为 0	R	保留位 读写的值总是为 0 。
3	OCF	0	R/W	操作数高速缓存闪存 写 1 时，操作数高速缓存所有全入口的 V、U、LRU 位清 0（闪存）。读取时值为 0。闪存时不对外部存储器进行回写。
2	—	0	R	保留位 读写的值总是为 0 。

位	位名	初始值	R/W	说　明
1	WT	0	R/W	直写 回写 / 直写的切换 0：回写模式 1：直写模式
0	OCE	0	R/W	操作数高速缓存有效 指定操作数高速缓存功能的允许 / 禁止。 0：操作数高速缓存禁止 1：操作数高速缓存允许

8.2.2　高速缓存控制寄存器 2 （CCR2）

CCR2 是控制操作数高速缓存的高速缓存锁定功能的寄存器。高速缓存锁定功能只有在高速缓存锁定模式时才有效。所谓高速缓存锁定模式就是 CCR2 的锁定允许位 （LE 位） =1 的状态。在非高速缓存锁定模式下，高速缓存锁定功能无效。

在高速缓存锁定模式中执行预取指令 （PREF@Rn）时，如果高速缓存未中，则根据 CCR2 的 bit9、8（W3LOAD、W3LOCK）以及 bit1、0（W2LOAD、W2LOCK）的设定，将 Rn 所指示的 1 条线的数据输入高速缓存。执行预取指令时，各位的设定与被替换的路的关系如**表 8.3** 所示。另一方面如果执行预取指令，并且高速缓存命中时，不进行新数据的输入，保持已经有效的入口。例如，Rn 所指示的 1 条线的数据已经存在于路 0 时，如果在高速缓存锁定模式，设定 W3LOAD=1 且 W3LOCK=1，并执行预取指令，就成为高速缓存命中，不对路 3 输入数据。

除高速缓存锁定模式下的预取指令以外，高速缓存存取限制通过W3LOCK、W2LOCK位被替换的路。CCR2 各位的设定与被替换的路的关系如**表 8.4** 所示。

请将变更 CCR2 内容的程序分配在高速缓存无效空间，读出 CCR2 的内容后请存取高速缓存有效空间。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LE
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

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	W3 LOAD*	W3 LOCK	-	-	-	-	-	-	W2 LOAD*	W2 LOCK
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	R	R	R	R	R	R/W	R/W

【注】* W3LOAD与W2LOAD不能同时置1。

位	位名	初始值	R/W	说　　明
31 ～ 17	—	均为 0	R	保留位 读写的值总是为 0。
16	LE	0	R/W	锁定允许 控制高速缓存锁定模式。 0：非高速缓存锁定模式 1：高速缓存锁定模式
15 ～ 10	—	均为 0	R	保留位 读写的值总是为 0。
9 8	W3LOAD* W3LOCK	0 0	R/W R/W	路 3 装入 路 3 锁定 W3LOCK=1、W3LOAD=1、且高速缓存锁定模式时，由预取指令引起的高速缓存未中的数据总是被读入到路 3。其他的所有条件中，高速缓存未中的数据被读入到 LRU 所示的路。
7 ～ 2	—	均为 0	R	保留位 读写的值总是为 0 。
1 0	W2LOAD* W2LOCK	0 0	R/W R/W	路 2 装入 路 2 锁定 W2LOCK=1、W2LOAD=1、且高速缓存锁定模式时，由预取指令引起的高速缓存未中的数据总是被读入到路 2。其他的全部条件中，高速缓存未中的数据被读入到 LRU 所示的路。

【注】 * W3LOAD 与 W2LOAD 不能被同时置 1。

表 8.3　PREF 指令在高速缓存未中时被替换的路

LE	W3LOAD*	W3LOCK	W2LOAD*	W2LOCK	被替换的路
0	x	x	x	x	根据 LRU （表 8.1）
1	x	0	x	0	根据 LRU （表 8.1）
1	x	0	0	1	根据 LRU （表 8.5）
1	0	1	x	0	根据 LRU （表 8.6）
1	0	1	0	1	根据 LRU （表 8.7）
1	0	x	1	1	路 2
1	1	1	0	x	路 3

【符号说明】 x：Don't care

【注】 * 不能同时设定 W3LOAD=1 且 W2LOAD=1。

表 8.4 除 PREF 指令以外在高速缓存未中时被替换的路

LE	W3LOAD*	W3LOCK	W2LOAD*	W2LOCK	被替换的路
0	×	×	×	×	根据 LRU （表 8.1）
1	×	0	×	0	根据 LRU （表 8.1）
1	×	0	×	1	根据 LRU （表 8.5）
1	×	1	×	0	根据 LRU （表 8.6）
1	×	1	×	1	根据 LRU （表 8.7）

【記号说明】 ×：Don't care

【注】 * 不能同时设定 W3LOAD=1 且 W2LOAD=1。

表 8.5 LRU 位与被替换的路 （W2LOCK=1 且 W3LOCK=0 时）

LRU （bit5 ～ 0）	被替换的路
000000，000001，000100，010100，100000，100001，110000，110100	3
000011，000110，000111，001011，001111，010110，011110，011111	1
101001，101011，111000，111001，111011，111100，111110，111111	0

表 8.6 LRU 位与被替换的路 （W2LOCK=0 且 W3LOCK=1 时）

LRU （bit5 ～ 0）	被替换的路
000000，000001，000011，001011，100000，100001，101001，101011	2
000100，000110，000111，001111，010100，010110，011110，011111	1
110000，110100，111000，111001，111011，111100，111110，111111	0

表 8.7 LRU 位与被替换的路 （W2LOCK=1 且 W3LOCK=1 时）

LRU （bit5 ～ 0）	被替换的路
000000，000001，000011，000100，000110，000111，001011，001111，010100，010110，011110，011111	1
100000，100001，101001，101011，110000，110100，111000，111001，111011，111100，111110，111111	0

8.3　运行说明

对操作数高速缓存进行说明。除地址阵列中没有 U 位、预取操作、写操作及回写缓冲器外，指令高速缓存与操作数高速缓存相同。

8.3.1　高速缓存的搜索

操作数高速缓存为允许时（CCR1 寄存器的 OCE 位 =1），存取高速缓存有效空间的数据时，就搜索高速缓存，调查目标数据是否存在于高速缓存中。高速缓存搜索方法的概念图如图 8.2 所示。

通过存储器存取地址的 bit10 ～ 4 选择入口，并读出该入口的标记符地址。此时，标记符地址的高 3 位总是为 0。存储器存取地址的 bit31 ～ 11 与读出的标记符地址进行比较。地址的比较使用 4 个路 （全部）。如果比较结果匹配且被比较的入口为有效 （V=1）时，就成为高速缓存命中。除此以外就成为高速缓存未中。路 1 命中时的情况如图 8.2 所示。

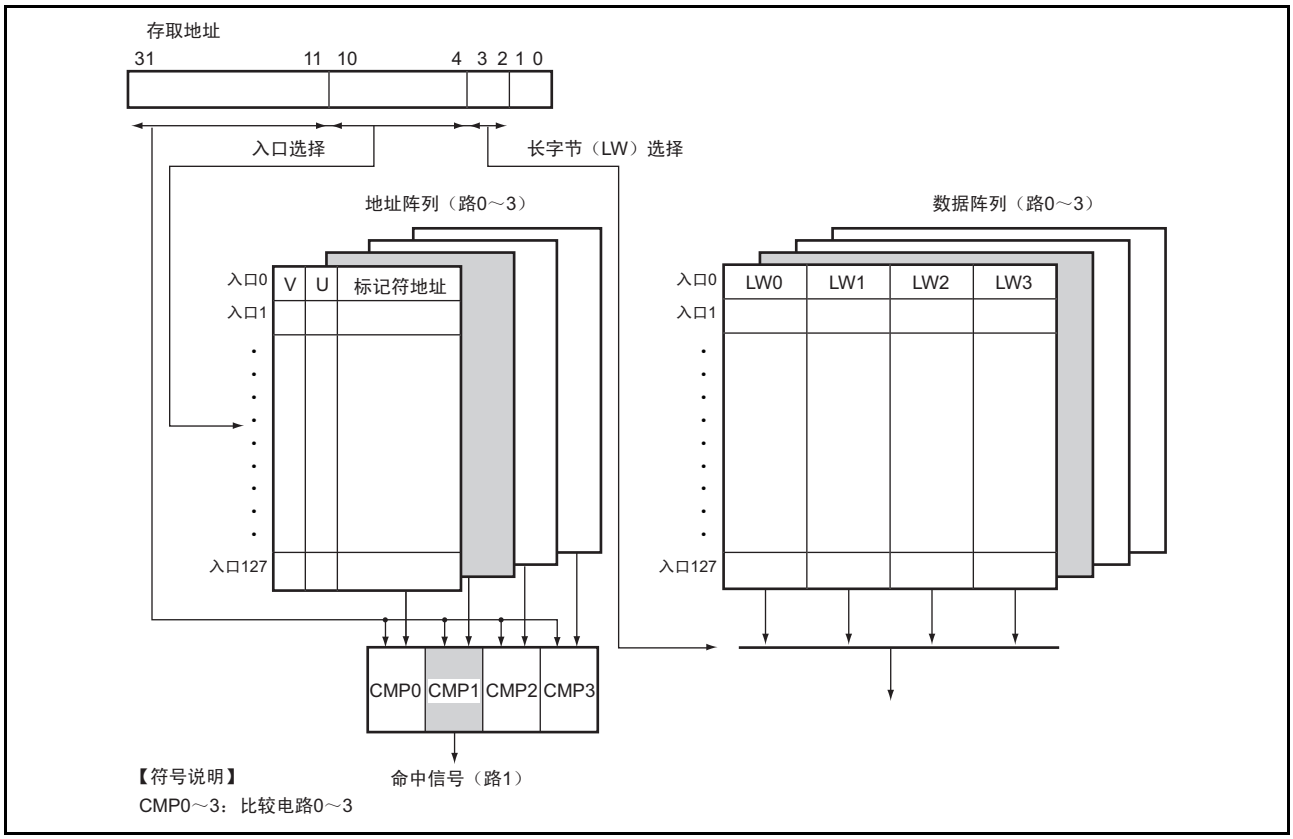


图 8.2　高速缓存的搜索方法

8.3.2 读操作

（1） 读命中

将数据从高速缓存传送给 CPU。更新 LRU 以便命中的路为最新。

（2） 读未中

启动外部总线周期，并更新入口。替换的路如**表 8.4** 所示。入口的更新单位是 16 字节。目标数据从外部存储器登录到高速缓存的同时，该数据被传送到 CPU。登录到高速缓存时，V 位被置 1，更新 LRU 以便被替换的路为最新。在操作数高速缓存中， U 位被清 0，在回写模式中由于入口的更新而被替换的入口的 U 位为 1 时，在该入口被传送到回写缓冲器后开始高速缓存更新周期。高速缓存更新周期结束后，传送到回写缓冲器的入口被回写到存储器。回写的单位是 16 字节。

8.3.3 预取操作 （仅限操作数高速缓存）

（1） 预取命中

更新 LRU 以便命中的路为最新。不变更其他高速缓存的内容。不向 CPU 传送数据。

（2） 预取未中

不向 CPU 传送数据，替换的路如**表 8.3** 所示。其他的运行与读未中时相同。

8.3.4 写操作 （仅限操作数高速缓存）

（1） 写命中

在回写模式中，将数据写入高速缓存，不发送对外部存储器的写周期。被写入口的 U 位被置 1，更新 LRU 以便命中的路为最新。

在直写模式中，将数据写入高速缓存，发送对外部存储器的写周期。不更新被写入口的 U 位，更新 LRU 以便命中的路为最新。

（2） 写未中

在回写模式中，写入未中时启动外部总线周期，并更新入口。替换的路如**表 8.4** 所示。由于入口的更新而被替换的入口的 U 位为 1 时，在该入口被传送到回写缓冲器后开始高速缓存更新周期。数据写入高速缓存后，U 位与 V 位被置 1。更新 LRU 以便被替换的路为最新。高速缓存更新周期结束后，传送到回写缓冲器的入口被回写到存储器。回写的单位是 16 字节。

在直写模式中，写未中时不对高速缓存进行写入，只对外部存储器进行写入。

8.3.5　回写缓冲器 （仅限操作数高速缓存）

在回写模式中，被替换入口的 U 位为 1 时，需要对外部存储器进行回写。为提高性能，首先将被替换的入口传送到回写缓冲器，并优先对高速缓存新入口的输入进行回写。对高速缓存新入口的输入结束后，回写缓冲器对外部存储器进行回写。在该回写中，高速缓存可进行存取。

回写缓冲器可保存高速缓存 1 条线的数据 （16 字节）和其地址。回写缓冲器的结构如图 8.3 所示。

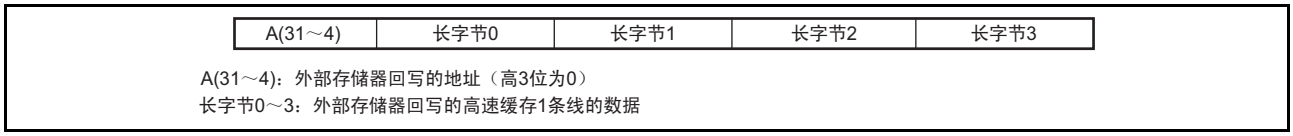


图 8.3　回写缓冲器的结构

上述 8.3.2 ～ 8.3.5 的运行汇总如表 8.8 所示。

表 8.8　高速缓存运行汇总

高速缓存的种类	CPU 周期	命中 / 未中	回写模式 / 直写模式	U 位	对外部存储器的存取（通过内部总线）	高速缓存的内容
指令高速缓存	取指令	命中	—	—	不发生。	不更新。
		未中	—	—	发生高速缓存更新周期。	通过高速缓存更新周期更新。
操作数高速缓存	预取 / 读	命中	任意模式	×	不发生。	不更新。
		未中	直写模式	—	发生高速缓存更新周期。	通过高速缓存更新周期更新。
			回写模式	0	发生高速缓存更新周期。	通过高速缓存更新周期更新。
				1	发生高速缓存更新周期。 之后发生回写缓冲器的回写周期。	通过高速缓存更新周期更新。
	写	命中	直写模式	—	发生 CPU 发送的写周期。	通过 CPU 发送的写周期的内容更新。
			回写模式	×	不发生。	通过 CPU 发送的写周期的内容更新。
		未中	直写模式	—	发生 CPU 发送的写周期。	不更新。 *
			回写模式	0	发生高速缓存更新周期。	通过高速缓存更新周期更新后，又通过 CPU 发送的写周期的内容更新。
				1	发生高速缓存更新周期。 之后发生回写缓冲器的回写周期。	通过高速缓存更新周期更新后，又通过 CPU 发送的写周期的内容更新。

【符号说明】×：Don't care

【注】 高速缓存更新周期：16 字节的读存取
回写缓冲器的回写周期：16字节的写存取
* 也不更新 LRU。除此以外的情况都更新 LRU。

8.3.6　高速缓存与外部存储器匹配

请用软件保证高速缓存与外部存储器相匹配。

如果将本 LSI 与其他器件的共有存储器分配在高速缓存有效空间时，请根据需要进行存储器分配的高速缓存，并进行无效化以及回写。关于本 LSI 内的 CPU 与 DMAC 共有的存储器，也请进行相同的操作。

8.4 存储器分配的高速缓存结构

因为用软件管理高速缓存，所以可通过 MOV 指令读写高速缓存的内容。指令高速缓存的地址阵列分配在 H'F0000000 ～ H'F07FFFFF，数据阵列分配在 H'F1000000 ～ H'F17FFFFF。操作数高速缓存的地址阵列分配在 H'F0800000 ～ H'F0FFFFFF，数据阵列分配在 H'F1800000 ～ H'F1FFFFFF。地址阵列与数据阵列的存取长度固定为长字节，不能进行取指令。

8.4.1 地址阵列

需要对地址阵列的存取指定 32 位的地址 （读 / 写时）和 32 位的数据 （写时）。

在地址中，指定选择入口的入口地址、选择路的 W 位以及指定有无关联动作的 A 位。 W 位的 B'00 表示路 0、B'01 表示路 1、B'10 表示路 2、B'11 表示路 3。因为存取长度固定为长字节，所以请对地址的 bit1 ～ 0 指定 B'00。

在数据中指定标记符地址、 LRU 位、 U 位 （仅限操作数高速缓存）以及 V 位。总是将标记符地址的高 3 位 （bit31 ～ 29）指定为 0。

关于地址及数据的格式，请参照**图 8.4**。

对地址阵列可进行下面 3 种操作。

（1） 读取地址阵列

从地址所指定的入口地址以及与路相对应的入口读出标记符地址、LRU 位、U 位（仅限操作数高速缓存）以及 V 位。读取时，地址所指定的关联位 （A 位）无论是 1 还是 0 都不进行关联运行。

（2） 写入地址阵列 （没有关联）

将地址的关联位 （A 位）清 0 并写入时，对地址所指定的入口地址以及与路相对应的入口，写入数据所指定的标记符地址、 LRU 位、 U 位 （仅限操作数高速缓存）以及 V 位。如果对 U 位 =1、 V 位 =1 的高速缓存线进行了写操作 （对操作数高速缓存地址阵列的写入）时，在该高速缓存线进行回写后，写入数据所指定的标记符地址、 LRU 位、 U 位以及 V 位。但对 V 位写 0 时，必须对该入口的 U 位也写 0。

（3） 写入地址阵列 （有关联）

如果将地址的关联位 （A 位）置 1 并写入时，对地址所指定的所有入口 （4 路）与数据所指定的标记符地址进行比较。对比较结果匹配的路，将数据所指定的 U 位 （仅限操作数高速缓）和 V 位写入入口。但标记符地址和 LRU 位不发生改变。没有一个路匹配时不进行写入，成为空操作。此运行用于高速缓存特定入口的无效化。

操作数高速缓存时，如果命中入口的 U 位是 1，就发生回写。但给 V 位写 0 时，也必须对该入口的 U 位写 0。

8.4.2　数据阵列

数据阵列的存取，必须指定 32 位的地址 （读 / 写时）和 32 位的数据 （写入时）。在地址中指定选择存取入口的信息，在数据中指定写入数据阵列的长字数据。

在地址中指定选择入口的入口地址、表示 1 条线 （16 字节）中的长字位置的 L 位、指定路的 W 位。 L 位中，B'00 表示长字 0，B'01 表示长字 1，B'10 表示长字 2，B'11 表示长字 3。W 位中，B'00 表示路 0，B'01 表示路 1， B'10 表示路 2， B'11 表示路 3。因为存取长度固定为长字，所以请给存取地址的 bit1 ～ 0 指定 B'00。

地址及数据格式，请参照图 8.4。

对数据阵列可进行下面 2 种操作。另外不通过此操作改变地址阵列的信息。

（1） 读取数据阵列

从地址所指定的入口地址及与路相对应的入口，读出通过地址的 L 位指定的数据。

（2） 写入数据阵列

在地址所指定的入口地址及与路相对应的入口中，向通过地址的 L 位指定的位置，写入数据所指定的长字数据。

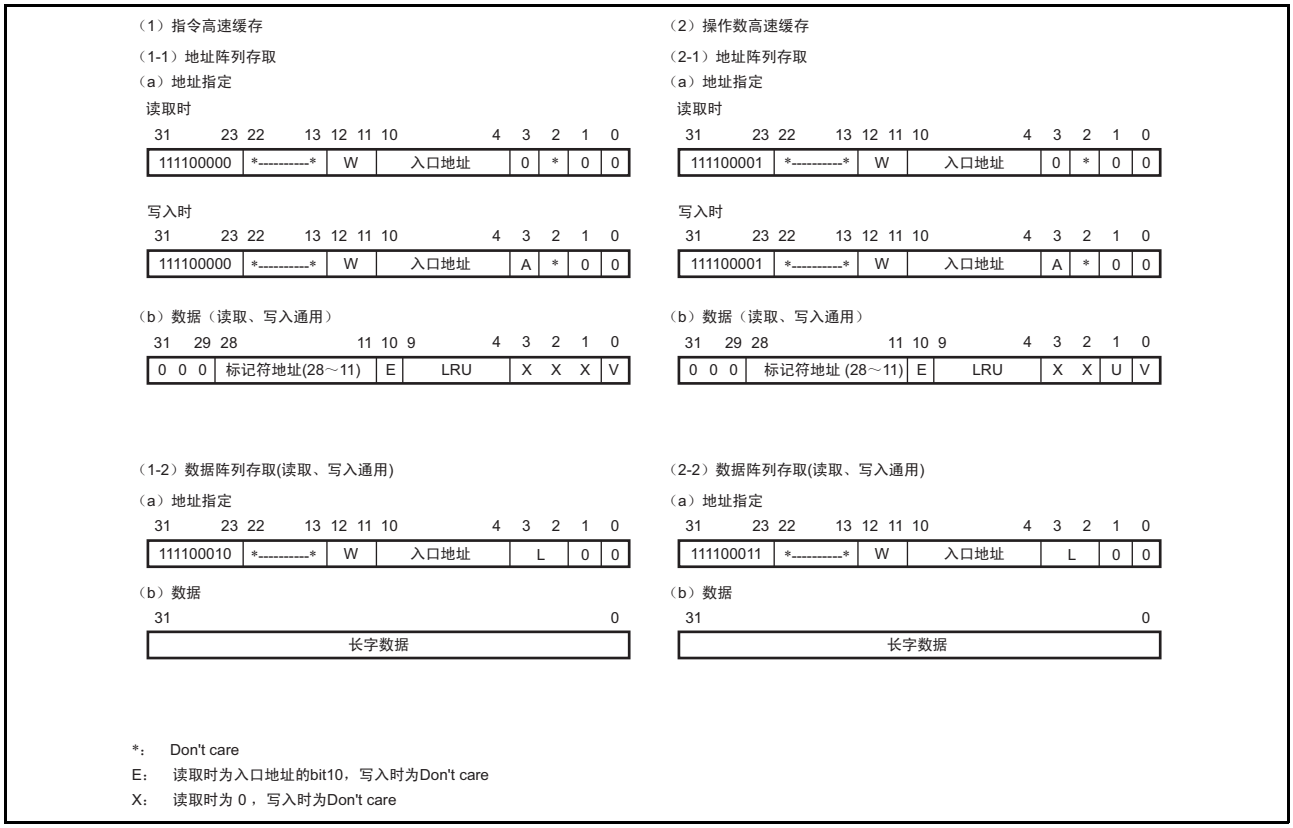


图 8.4　存储器分配高速缓存存取的地址、数据指定方法

8.4.3 使用例

（1） 特定入口的无效化

在存储器分配高速缓存存取中，通过对该入口的 V 位写 0 可实现高速缓存特定入口的无效化。将 A 位置 1，对写入数据所指定的标记符地址与入口地址所选择的高速缓存中的标记符地址进行比较，匹配时，写入 V 位与 U 位 （由写入数据指定）。不匹配时为空操作。将地址阵列某个入口的 V 位置 0 时，如果该入口的 U 位为 1，该入口就被回写。

下面是向 R0 指定写入数据，向 R1 指定地址的例子。

```
；  R0=H'0110 0010 ；标记符地址 （28 ～ 11） =B'0 0001 0001 0000 0000 0、 U=0、 V=0
；  R1=H'F080 0088 ；操作数高速缓存地址阵列存取、入口 =B'000 1000、 A=1
；
   MOV.L   R0，  @R1
```

（2） 读取特定入口的数据

可通过存储器分配高速缓存存取读取特定入口的数据。图 8.4 的数据阵列中数据区所示的长字被读取到寄存器。

下面是向 R0 指定地址读取到 R1 的例子。

```
        ；  R0=H'F100 004C ；指令高速缓存数据阵列存取、入口 =B'000 0100、路 =0、
          长字节地址 =3
；
   MOV.L   @ R0，  R1
```

8.4.4 注意事项

1. 存取存储器分配高速缓存的程序，请分配在高速缓存无效空间。
2. 禁止改写地址阵列的内容以便多个路同时命中。不保证改写地址阵列内容时的运行以便多个路同时命中。
3. 寄存器以及存储器分配高速缓存只能通过CPU存取，而不能通过DMAC存取。

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。因此，本 LSI 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 64M 字节
   - 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 的存储器种类和 PCMCIA 接口
   - 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）及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
   - 支持低频率模式、掉电模式
   - 支持发送 MRS 命令、 EMRS 命令

6. PCMCIA 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

7. 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

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. 总线仲裁
   - 与其他的CPU共有所有资源，可接受来自外部的总线权请求以及输出总线使用允许

11. 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

12. 将刷新计数器用作间隔定时器
   - 能通过比较匹配定时器产生中断请求

BSC 的框图如图 9.1 所示。

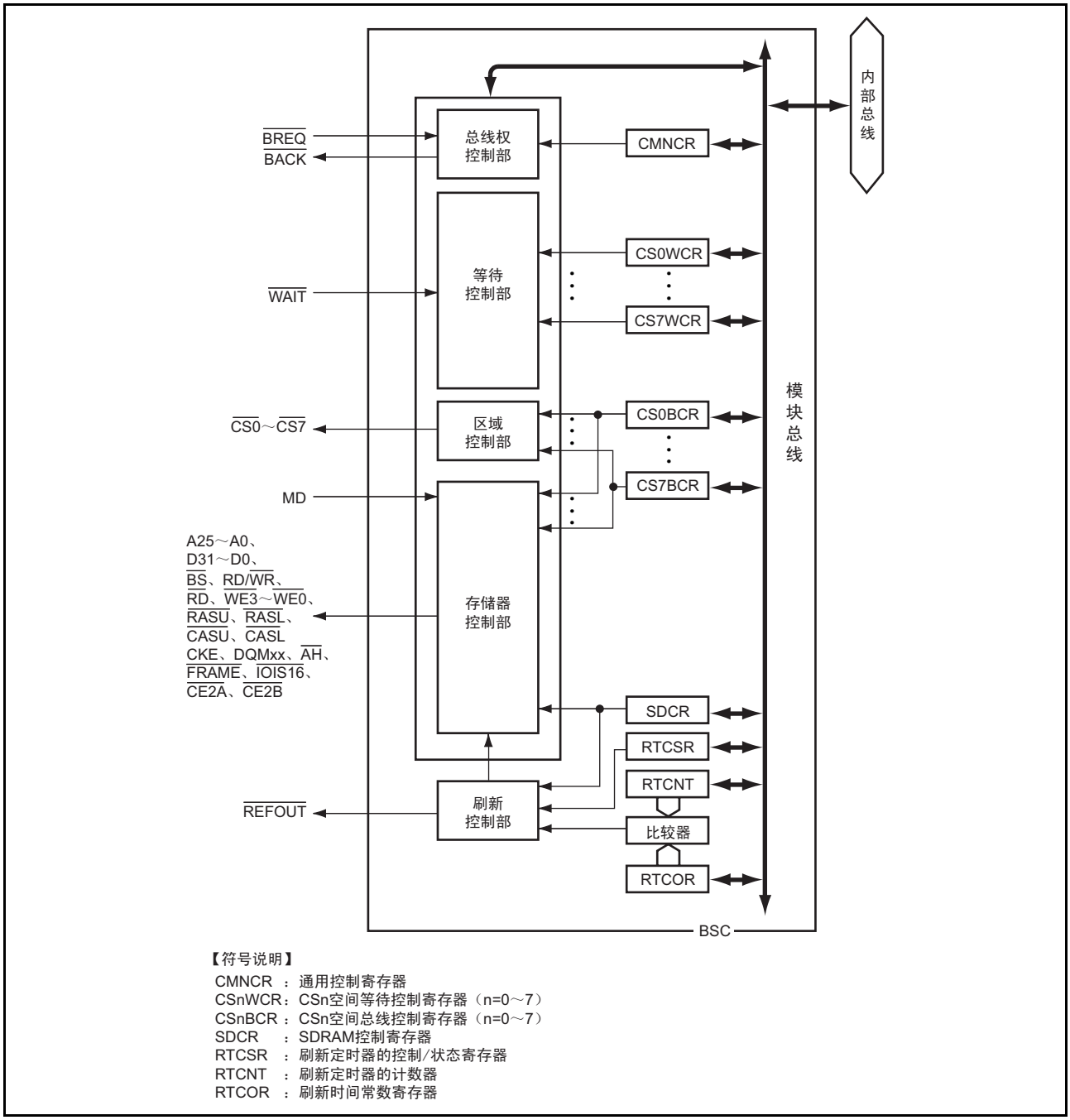


图 9.1　BSC 的框图

9.2　输入 / 输出引脚

BSC 的引脚结构如表 9.1 所示。

表 9.1　引脚构成

引脚名	输入 / 输出	功　　能
A25 ～ A0	输出	地址总线
D31 ～ D0	输入 / 输出	数据总线
$\overline{BS}$	输出	表示总线周期开始的信号
$\overline{CS0}$ ～ $\overline{CS4}$、 $\overline{CS7}$	输出	片选
$\overline{CS5}$/$\overline{CE1A}$、 $\overline{CS6}$/$\overline{CE1B}$	输出	片选 在使用 PCMCIA 时，为 PCMCIA 卡的选择信号，对应 D7 ～ D0
$\overline{CE2A}$、 $\overline{CE2B}$	输出	PCMCIA 卡的选择信号，对应 D15 ～ D8
RD/$\overline{WR}$	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 $\overline{WE}$ 引脚
$\overline{RD}$	输出	读取脉冲信号 （读数据输出的允许信号） 在使用 PCMCIA 时，为表示存储器读周期的选通信号
$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$/$\overline{AH}$	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 在连接 SDRAM 时，为对应 D31 ～ D24 的选择信号 在使用 PCMCIA 时，为表示 I/O 写操作的选通信号 在使用 MPX-I/O 时，为用于保持地址的信号
$\overline{WE2}$/DQMUL/$\overline{ICIORD}$	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 在连接 SDRAM 时，为对应 D23 ～ D16 的选择信号 在使用 PCMCIA 时，为表示 I/O 读操作的选通信号
$\overline{WE1}$/DQMLU/$\overline{WE}$	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 在连接 SDRAM 时，为对应 D15 ～ D8 的选择信号 在使用 PCMCIA 时，为表示存储器写周期的选通信号
$\overline{WE0}$/DQMLL	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 在连接 SDRAM 时，对应 D7 ～ D0 的选择信号
$\overline{RASU}$、 $\overline{RASL}$	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
$\overline{FRAME}$	输出	使用突发 MPX-I/O 接口时为 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
$\overline{BACK}$	输出	总线使用允许的输出
$\overline{REFOUT}$	输出	总线释放时，输出刷新执行请求
$\overline{IOIS16}$	输入	表示 PCMCIA16 位 I/O 的信号 只在小端法时有效。大端法时，变为低电平。
MD	输入	选择区域 0 的总线宽度，选择区域 1 ～ 7 的总线宽度初始值

9.3　区域概要

9.3.1　地址映射

本 LSI 的体系结构拥有 32 位地址空间。通过高位分配为高速缓存有效空间、高速缓存无效空间、内置空间 （内置 RAM、内置外围模块、保留）。

CS0 ～ CS7 的外部地址空间，在内部地址 A29=0 时高速缓存有效，在 A29=1 时高速缓存无效。

给各个空间指定连接的存储器种类和数据总线宽度。外部地址空间的地址映射如下表所示。

表 9.2　地址映射

内部地址	空间	存储器种类	高速缓存
H'00000000 ～ H'03FFFFFF	CS0	通常空间，突发 ROM （异步、同步）	有效
H'04000000 ～ H'07FFFFFF	CS1	通常空间，带字节选择的 SRAM	
H'08000000 ～ H'0BFFFFFF	CS2	通常空间，带字节选择的 SRAM、 SDRAM	
H'0C000000 ～ H'0FFFFFFF	CS3	通常空间，带字节选择的 SRAM、 SDRAM	
H'10000000 ～ H'13FFFFFF	CS4	通常空间，带字节选择的 SRAM、突发 ROM （异步）	
H'14000000 ～ H'17FFFFFF	CS5	通常空间，带字节选择的 SRAM、 MPX-I/O、 PCMCIA	
H'18000000 ～ H'1BFFFFFF	CS6	通常空间，带字节选择的 SRAM、突发 MPX-I/O、 PCMCIA	
H'1C000000 ～ H'1FFFFFFF	CS7	通常空间，带字节选择的 SRAM	
H'20000000 ～ H'23FFFFFF	CS0	通常空间，突发 ROM （异步、同步）	无效
H'24000000 ～ H'27FFFFFF	CS1	通常空间，带字节选择的 SRAM	
H'28000000 ～ H'2BFFFFFF	CS2	通常空间，带字节选择的 SRAM、 SDRAM	
H'2C000000 ～ H'2FFFFFFF	CS3	通常空间，带字节选择的 SRAM、 SDRAM	
H'30000000 ～ H'33FFFFFF	CS4	通常空间，带字节选择的 SRAM、突发 ROM （异步）	
H'34000000 ～ H'37FFFFFF	CS5	通常空间，带字节选择的 SRAM、 MPX-I/O、 PCMCIA	
H'38000000 ～ H'3BFFFFFF	CS6	通常空间，带字节选择的 SRAM、突发 MPX-I/O、 PCMCIA	
H'3C000000 ～ H'3FFFFFFF	CS7	通常空间，带字节选择的 SRAM	
H'80000000 ～ H'FFFBFFFF	其他	内置 RAM、保留区域 *	—
H'FFFC0000 ～ H'FFFFFFFF	其他	内置外围模块、保留区域 *	—

【注】 * 内置 RAM 空间请存取 “第 31 章 内置 RAM” 中所示的地址。内置外围模块空间请存取 “第 34 章 寄存器一览” 中所示的地址。请勿存取没记载的地址，存取时不能保证正常运行。

9.3.2　各区域的数据总线宽度和引脚功能设定

在本 LSI 中，可在上电复位时使用外部引脚，将区域 0 的数据总线宽度和区域 1 ～ 7 的初始状态的数据总线宽度任意设定为 16 位或 32 位。区域 0 的总线宽度在上电复位后不能改变。虽然区域 1 ～ 7 的初始状态的数据总线宽度与区域 0 相同，但通过寄存器的设定，在程序中可任意变更为 8 位、 16 位、 32 位。请注意，根据设定的存储器类型，可能会限制数据总线宽度。

在上电复位后，启动保存在外部存储器（分配在区域 0 中）中的程序。假设区域 0 的外部存储器为 ROM，因此就成为地址总线、数据总线、 $\overline{\text{CS0}}$、 $\overline{\text{RD}}$ 等基本的引脚功能。在本章记载的存取波形例子中，显示了 $\overline{\text{BS}}$、RD/$\overline{\text{WR}}$、 $\overline{\text{WEn}}$ 等引脚，但这些是通过引脚功能控制器设定引脚功能时的例子。在通过程序完成引脚的设定之前，只能执行区域 0 的读存取。按照 32 位数据总线宽度启动 LSI 后，区域 0 以外的空间总线宽度变为 16 位时，需要设定 A1 引脚，变为 8 位时，需要设定 A1、A0 引脚。请注意，使用区域 7 空间时， $\overline{\text{CS7}}$ 和 A0 功能分配给同一引脚，因此不能设定 8 位总线宽度。

引脚设定的详细情况请参照 “第 29 章 引脚功能控制器 （PFC）”。

表 9.3　外部引脚 （MD）设定和数据总线宽度的关系

MD	数据总线宽度
1	32 位
0	16 位

9.4　寄存器说明

BSC 有以下寄存器。

在结束连接存储器的接口设定之前，请勿存取区域 0 以外的空间。

表 9.4　寄存器结构

寄存器名	简称	R/W	初始值	地址	存取长度
通用控制寄存器	CMNCR	R/W	H'00001010	H'FFFC0000	32
CS0 空间总线控制寄存器	CS0BCR	R/W	H'36DB0600*	H'FFFC0004	32
CS1 空间总线控制寄存器	CS1BCR	R/W	H'36DB0600*	H'FFFC0008	32
CS2 空间总线控制寄存器	CS2BCR	R/W	H'36DB0600*	H'FFFC000C	32
CS3 空间总线控制寄存器	CS3BCR	R/W	H'36DB0600*	H'FFFC0010	32
CS4 空间总线控制寄存器	CS4BCR	R/W	H'36DB0600*	H'FFFC0014	32
CS5 空间总线控制寄存器	CS5BCR	R/W	H'36DB0600*	H'FFFC0018	32
CS6 空间总线控制寄存器	CS6BCR	R/W	H'36DB0600*	H'FFFC001C	32
CS7 空间总线控制寄存器	CS7BCR	R/W	H'36DB0600*	H'FFFC0020	32
CS0 空间等待控制寄存器	CS0WCR	R/W	H'00000500	H'FFFC0028	32
CS1 空间等待控制寄存器	CS1WCR	R/W	H'00000500	H'FFFC002C	32
CS2 空间等待控制寄存器	CS2WCR	R/W	H'00000500	H'FFFC0030	32
CS3 空间等待控制寄存器	CS3WCR	R/W	H'00000500	H'FFFC0034	32
CS4 空间等待控制寄存器	CS4WCR	R/W	H'00000500	H'FFFC0038	32
CS5 空间等待控制寄存器	CS5WCR	R/W	H'00000500	H'FFFC003C	32
CS6 空间等待控制寄存器	CS6WCR	R/W	H'00000500	H'FFFC0040	32
CS7 空间等待控制寄存器	CS7WCR	R/W	H'00000500	H'FFFC0044	32
SDRAM 控制寄存器	SDCR	R/W	H'00000000	H'FFFC004C	32
刷新定时器的控制 / 状态寄存器	RTCSR	R/W	H'00000000	H'FFFC0050	32
刷新定时器的计数器	RTCNT	R/W	H'00000000	H'FFFC0054	32
刷新时间常数寄存器	RTCOR	R/W	H'00000000	H'FFFC0058	32

【注】 * 为通过外部引脚 （MD）（总线宽度设定为 32 位）启动 LSI 时的初始值。总线宽度设定为 16 位时，为 H'36DB0400。

9.4.1　通用控制寄存器 （CMNCR）

CMNCR 是控制各区域通用部分的 32 位寄存器。

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
	-	-	-	-	BLOCK	DPRTY[1:0]		DMAIW[2:0]			DMA IWA	-	-	-	HIZ MEM	HIZ CNT
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/W	R/W	R/W	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
31 ～ 13	—	均为 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	BLOCK	0	R/W	总线锁定位 指定接受或不接受 $\overline{\text{BREQ}}$。 0：接受 $\overline{\text{BREQ}}$。 1：不接受 $\overline{\text{BREQ}}$。
10、9	DPRTY[1:0]	00	R/W	DMA 突发传送优先级 该位指定 DMA 突发传送中刷新请求 / 总线权使用请求的优先级。 00：在 DMA 突发传送中接受刷新请求 / 总线权使用请求。 01：在 DMA 突发传送中接受刷新请求，但不接受总线权使用请求。 10：在 DMA 突发传送中不接受刷新请求 / 总线权使用请求。 11：保留 （禁止设定）
8 ～ 6	DMAIW[2:0]	000	R/W	指定 DMA 单地址传送时存取周期之间的等待 该位指定 DMA 单地址传送时，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后面所述 DMAIWA 位的指定而不同。 000：不插入空闲周期 001：插入 1 个空闲周期 010：插入 2 个空闲周期 011：插入 4 个空闲周期 100：插入 6 个空闲周期 101：插入 8 个空闲周期 110：插入 10 个空闲周期 111：插入 12 个空闲周期

位	位名	初始值	R/W	说　　明
5	DMAIWA	0	R/W	指定 DMA 单地址传送时存取周期之间的等待插入方法 该位指定空闲周期 （由 DMAIW[2:0] 位指定）的插入方法。 当该位为 0 时，带 DACK 的外部器件驱动数据总线后，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。当该位为 1 时，即使连续存取带 DACK 的外部器件，也必须在 1 次存取结束后插入空闲周期。 0：带 DACK 的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	—	均为 0	R	保留位 读写的值总是 0。
1	HIZMEM	0	R/W	High-Z 存储器控制 该位指定软件待机模式和深度待机模式时的 A25 ～ A0、$\overline{BS}$、$\overline{CSn}$、$\overline{CE2x}$、RD/$\overline{WR}$、$\overline{WEn}$/DQMxx/$\overline{AH}$、$\overline{RD}$、$\overline{FRAME}$ 引脚状态。释放总线时，与该位无关，这些引脚为高阻抗状态。 0：在软件待机模式和深度待机模式时，引脚为高阻抗状态 1：在软件待机模式和深度待机模式时，引脚为驱动状态
0	HIZCNT*	0	R/W	High-Z 控制 该位指定软件待机模式、深度待机模式和释放总线权时的 CKE、$\overline{RASU}$、$\overline{RASL}$、$\overline{CASU}$、$\overline{CASL}$ 引脚状态。 0：在软件待机模式、深度待机模式和释放总线权时， CKE、$\overline{RASU}$、$\overline{RASL}$、$\overline{CASU}$、$\overline{CASL}$ 引脚为高阻抗状态 1：在软件待机模式、深度待机模式和释放总线权时， CKE、$\overline{RASU}$、$\overline{RASL}$、$\overline{CASU}$、$\overline{CASL}$ 引脚为驱动状态

【注】 *　CKIO 的 High-Z 控制请参照 “第 4 章　时钟脉冲振荡器 （CPG）”。

9.4.2　CSn 空间总线控制寄存器 （CSnBCR）（n=0 ～ 7）

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，请勿存取区域 0 以外的外部存储器。
即使指定无空闲周期，也有插入空闲周期的情况。详细情况请参照 “9.5.12 存取周期之间的空闲”。

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

【注】* 在上电复位时对指定数据总线宽度的外部引脚（MD）的值进行采样。

位	位名	初始值	R/W	说　　明
31	—	0	R	保留位 读写的值总是 0。
30 ～ 28	IWW[2:0]	011	R/W	指定写－读 / 写－写周期之间的空闲 该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。 对象周期为写－读与写－写周期时。 000：不插入空闲周期 001：插入 1 个空闲周期 010：插入 2 个空闲周期 011：插入 4 个空闲周期 100：插入 6 个空闲周期 101：插入 8 个空闲周期 110：插入 10 个空闲周期 111：插入 12 个空闲周期
27 ～ 25	IWRWD[2:0]	011	R/W	指定不同空间读－写周期之间的空闲 该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。 对象周期为在不同空间的连续读－写周期。 000：不插入空闲周期 001：插入 1 个空闲周期 010：插入 2 个空闲周期 011：插入 4 个空闲周期 100：插入 6 个空闲周期 101：插入 8 个空闲周期 110：插入 10 个空闲周期 111：插入 12 个空闲周期
24 ～ 22	IWRWS[2:0]	011	R/W	指定同一空间读－写周期之间的空闲 该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。 000：不插入空闲周期 001：插入 1 个空闲周期 010：插入 2 个空闲周期 011：插入 4 个空闲周期 100：插入 6 个空闲周期 101：插入 8 个空闲周期 110：插入 10 个空闲周期 111：插入 12 个空闲周期

位	位名	初始值	R/W	说　　明
21 ～ 19	IWRRD[2:0]	011	R/W	指定不同空间读－读周期之间的空闲 该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。 000：不插入空闲周期 001：插入 1 个空闲周期 010：插入 2 个空闲周期 011：插入 4 个空闲周期 100：插入 6 个空闲周期 101：插入 8 个空闲周期 110：插入 10 个空闲周期 111：插入 12 个空闲周期
18 ～ 16	IWRRS[2:0]	011	R/W	指定同一空间读－读周期之间的空闲 该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。 000：不插入空闲周期 001：插入 1 个空闲周期 010：插入 2 个空闲周期 011：插入 4 个空闲周期 100：插入 6 个空闲周期 101：插入 8 个空闲周期 110：插入 10 个空闲周期 111：插入 12 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	该位设定连接空间的存储器种类。 000：通常空间 001：突发 ROM （时钟异步） 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。 【注】 如果给 CS0 空间连接突发 ROM，请通过使用 CS0WCR 寄存器的突发 ROM 变更为需要的设定后，给突发 ROM 设定 TYPE[2:0]。
11	ENDIAN	0	R/W	指定端法 该位指定空间的数据排列。 0：按照大端法运行 1：按照小端法运行

位	位名	初始值	R/W	说　　明
10、 9	BSZ[1:0]	11*	R/W	指定数据总线宽度 该位指定空间的数据总线宽度。 00：保留 （禁止设定） 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 【注】 1. 当区域 5 设定为 MPX-I/O 时，如果将该位设定为 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）。另外，固定总线宽度可设定为 8 位或 16 位。 2. 通过外部引脚设定区域 0 ～ 7 初始状态的数据总线宽度。此时，忽视 CS0BCR 的 BSZ[1:0] 位的写操作，可改变 CS1BCR ～ CS7BCR 的总线宽度。 3. 当区域 6 设定为突发 MPX-I/O 时，仅可将总线宽度设定为 32 位。 4. 当区域 5 或者区域 6 设定为 PCMCIA 空间时，总线宽度可选择为 8 位或者 16 位。 5. 当区域 2 或者区域 3 设定为 SDRAM 空间时，总线宽度可选择为 16 位或者 32 位。 6. 当区域 0 设定为时钟同步突发 ROM 空间时，总线宽度可选择为 16 位或者 32 位。 7. 使用 8 位数据总线宽度时，不能使用区域 7。在使用区域 7 时，请将使用的所有区域设定为 16 位或者 32 位数据总线宽度。
8 ～ 0	—	均为 0	R	保留位 读写的值总是 0。

【注】 * 在上电复位时对指定数据总线宽度的外部引脚 （MD）的值进行采样。

9.4.3 CSn 空间等待控制寄存器 （CSnWCR）（n=0 ～ 7）

CSnWCR 设定有关存储器存取的各种等待周期。该寄存器的位结构因 CSn 空间总线控制寄存器（CSnBCR）设定的存储器种类 （TYPE[2:0]）而发生以下变化。请在存取前设定对象区域，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、 MPX-I/O

- CS0WCR

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/W	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]		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	—	均为 0	R	保留位 读写的值总是 0。
21、 20	—*	均为 0	R/W	保留位 为通常空间接口时置 0。
19、 18	—	均为 0	R	保留位 读写的值总是 0。
17、 16	—*	均为 0	R/W	保留位 为通常空间接口时置 0。
15 ～ 13	—	均为 0	R	保留位 读写的值总是 0。
12、 11	SW[1:0]	00	R/W	地址、 $\overline{\text{CS0}}$ 断言 →$\overline{\text{RD}}$、 $\overline{\text{WEn}}$ 断言的延迟周期数 该位指定从地址、 $\overline{\text{CS0}}$ 断言到 $\overline{\text{RD}}$、 $\overline{\text{WEn}}$ 断言的延迟周期数。 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	存取等待周期数 该位指定读写存取所需的等待周期数。 0000：不插入等待周期 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 （禁止设定） 1110：保留 （禁止设定） 1111：保留 （禁止设定）
6	WM	0	R/W	外部等待屏蔽的指定 该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，该位的设定也有效 0：外部等待输入有效 1：忽视外部等待输入
5 ～ 2	—	均为 0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说　　明
1、 0	HW[1:0]	00	R/W	$\overline{RD}$、 $\overline{WEn}$ 取反 → 地址、 $\overline{CS0}$ 取反的延迟周期数 该位指定从 $\overline{RD}$、 $\overline{WEn}$ 取反到地址、 $\overline{CS0}$ 取反的延迟周期数。 00：0.5 个周期 01：1.5 个周期 10：2.5 个周期 11：3.5 个周期

【注】 * 在 CS0 空间连接突发 ROM 并启动后，切换为突发 ROM 接口时，通过 bit21、 20 和 bit17、 16 分别指定突发数和突发等待周期数后，设定 CS0BCR 的 TYPE[2:0]。请勿对上述以外的保留位写 1。

- CS1WCR、 CS7WCR

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	—	均为 0	R	保留位 读写的值总是 0。
20	BAS	0	R/W	带字节选择的 SRAM　字节存取选择 该位设定带字节选择的 SRAM 接口时 $\overline{WEn}$ 和 RD/$\overline{WR}$ 信号的时序。 0：$\overline{WEn}$ 在读写时序时断言， RD/$\overline{WR}$ 在写存取周期时断言 1：$\overline{WEn}$ 在读写存取周期时断言、 RD/$\overline{WR}$ 在写时序时断言
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 该位指定写存取所需的周期数。 000：与 WR[3:0] 设定的 （读存取等待数）周期相同 001：不插入等待周期 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	—	均为 0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说　　明
12、 11	SW[1:0]	00	R/W	地址、 $\overline{\text{CSn}}$ 断言 →$\overline{\text{RD}}$、 $\overline{\text{WEn}}$ 断言的延迟周期数 该位指定从地址、 $\overline{\text{CSn}}$ 断言到 $\overline{\text{RD}}$、 $\overline{\text{WEn}}$ 断言的延迟周期数。 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 该位指定读存取所需的等待周期数。 0000：不插入等待周期 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 （禁止设定） 1110：保留 （禁止设定） 1111：保留 （禁止设定）
6	WM	0	R/W	指定外部等待屏蔽 该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，该位的设定也有效 0：外部等待输入有效 1：忽视外部等待输入
5 ～ 2	—	均为 0	R	保留位 读写的值总是 0。
1、 0	HW[1:0]	00	R/W	$\overline{\text{RD}}$、 $\overline{\text{WEn}}$ 取反 → 地址、 $\overline{\text{CSn}}$ 取反的延迟周期数 该位指定从 $\overline{\text{RD}}$、 $\overline{\text{WEn}}$ 取反到地址、 $\overline{\text{CSn}}$ 取反的延迟周期数。 00：0.5 个周期 01：1.5 个周期 10：2.5 个周期 11：3.5 个周期

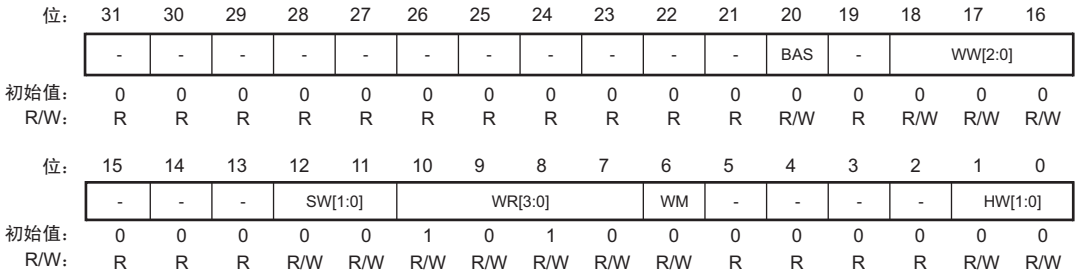
- CS2WCR、 CS3WCR

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	BAS	-	-	-	-
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	R

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	WR[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 ～ 21	—	均为 0	R	保留位 读写的值总是 0。
20	BAS	0	R/W	带字节选择的 SRAM　字节存取选择 该位设定带字节选择的 SRAM 接口时 $\overline{\text{WEn}}$ 和 RD/$\overline{\text{WR}}$ 信号的时序。 0：$\overline{\text{WEn}}$ 在读写时序时断言， RD/$\overline{\text{WR}}$ 在写存取周期时断言 1：$\overline{\text{WEn}}$ 在读写存取周期时断言、 RD/$\overline{\text{WR}}$ 在写时序时断言
19 ～ 11	—	均为 0	R	保留位 读写的值总是 0。
10 ～ 7	WR[3:0]	1010	R/W	存取等待周期数 该位指定读写存取所需的等待周期数。 0000：不插入等待周期 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 （禁止设定） 1110：保留 （禁止设定） 1111：保留 （禁止设定）
6	WM	0	R/W	指定外部等待屏蔽 该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，该位的设定也有效。 0：外部等待输入有效 1：忽视外部等待输入
5 ～ 0	—	均为 0	R	保留位 读写的值总是 0。

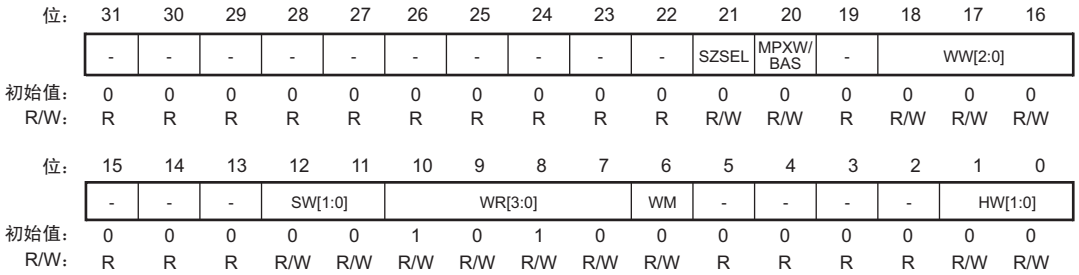
- CS4WCR



位	位名	初始值	R/W	说　　明
31 ～ 21	—	均为 0	R	保留位 读写的值总是 0。
20	BAS	0	R/W	带字节选择的 SRAM　字节存取选择 该位设定带字节选择的 SRAM 接口时 $\overline{\text{WEn}}$ 和 RD/$\overline{\text{WR}}$ 信号的时序。 0：$\overline{\text{WEn}}$ 在读写时序时断言， RD/$\overline{\text{WR}}$ 在写存取周期时断言 1：$\overline{\text{WEn}}$ 在读写存取周期时断言、 RD/$\overline{\text{WR}}$ 在写时序时断言
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 该位指定写存取所需的等待周期数。 000：与 WR[3:0] 设定的 （读存取等待数）周期相同 001：不插入等待周期 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	—	均为 0	R	保留位 读写的值总是 0。
12、 11	SW[1:0]	00	R/W	地址、 $\overline{\text{CS4}}$ 断言 →$\overline{\text{RD}}$、 $\overline{\text{WEn}}$ 断言的延迟周期数 该位指定从地址、 $\overline{\text{CS4}}$ 断言到 $\overline{\text{RD}}$、 $\overline{\text{WEn}}$ 断言的延迟周期数。 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	读存取的等待周期数 该位指定读存取所需的等待周期数。 0000：不插入等待周期 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 （禁止设定） 1110：保留 （禁止设定） 1111：保留 （禁止设定）
6	WM	0	R/W	指定外部等待屏蔽 该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，该位的设定也有效。 0：外部等待输入有效 1：忽视外部等待输入
5 ～ 2	—	均为 0	R	保留位 读写的值总是 0。
1、0	HW[1:0]	00	R/W	$\overline{\text{RD}}$、$\overline{\text{WEn}}$ 取反 → 地址、$\overline{\text{CSn}}$ 取反的延迟周期数 该位指定从 $\overline{\text{RD}}$、$\overline{\text{WEn}}$ 取反到地址、$\overline{\text{CSn}}$ 取反的延迟周期数。 00：0.5 个周期 01：1.5 个周期 10：2.5 个周期 11：3.5 个周期

- CS5WCR



位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	SZSEL	MPXW/ 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/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	—	均为 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 该位指定要选择总线宽度 （将 CS5BCR 的 BSZ[1:0] 设定为 11 时）的地址。 该设定只在将区域 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 SZSEL 位和通过 A14、 A21 选择总线宽度的关系如下所示。
20	MPXW	0	R/W	MPX-I/O 接口地址等待 该设定只在将区域 5 设定为 MPX-I/O 时有效。 该位设定 MPX-I/O 接口地址周期插入等待。 0：不插入等待 1：插入 1 个周期等待
	BAS	0	R/W	带字节选择的 SRAM　字节存取选择 该设定只在将区域 5 设定为带字节选择的 SRAM 时有效。 该位设定带字节选择的 SRAM 接口时 $\overline{\text{WEn}}$ 和 RD/$\overline{\text{WR}}$ 信号的时序。 0：$\overline{\text{WEn}}$ 在读写时序时断言， RD/$\overline{\text{WR}}$ 在写存取周期时断言 1：$\overline{\text{WEn}}$ 在读写存取周期时断言、 RD/$\overline{\text{WR}}$ 在写时序时断言
19	—	0	R	保留位 读写的值总是 0。

SZSEL	A14	A21	说明
0	0	没有影响	8位总线宽度
	1	没有影响	16位总线宽度
1	没有影响	0	8位总线宽度
	没有影响	1	16位总线宽度

位	位名	初始值	R/W	说　　明
18 ～ 16	WW[2:0]	000	R/W	写存取的等待周期数 该位指定写存取所需的周期数。 000：与 WR[3:0] 设定的 （读存取等待数）周期相同 001：不插入等待周期 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	—	均为 0	R	保留位 读写的值总是 0。
12、 11	SW[1:0]	00	R/W	地址、 $\overline{\text{CS5}}$ 断言 →$\overline{\text{RD}}$、 $\overline{\text{WEn}}$ 断言的延迟周期数 该位指定从地址、 $\overline{\text{CS5}}$ 断言到 $\overline{\text{RD}}$、 $\overline{\text{WEn}}$ 断言的延迟周期数。 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 该位指定读存取所需的等待周期数。 0000：不插入等待周期 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 （禁止设定） 1110：保留 （禁止设定） 1111：保留 （禁止设定）
6	WM	0	R/W	指定外部等待屏蔽 该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，该位的设定也有效。 0：外部等待输入有效 1：忽视外部等待输入
5 ～ 2	—	均为 0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说　　明
1、 0	HW[1:0]	00	R/W	$\overline{RD}$、 $\overline{WEn}$ 取反 → 地址、 $\overline{CS5}$ 取反的延迟周期数 该位指定从 $\overline{RD}$、 $\overline{WEn}$ 取反到地址、 $\overline{CS5}$ 取反的延迟周期数。 00：0.5 个周期 01：1.5 个周期 10：2.5 个周期 11：3.5 个周期

- CS6WCR

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	BAS	-	-	-	-
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	R

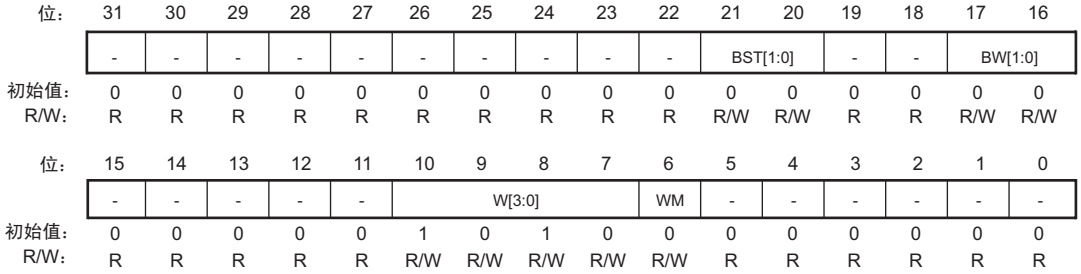
位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SW[1:0]		WR[3:0]				WM	-	-	-	-	HW[1:0]	
初始值：	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W：	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

位	位名	初始值	R/W	说　　明
31 ～ 21	—	均为 0	R	保留位 读写的值总是 0。
20	BAS	0	R/W	带字节选择的 SRAM　字节存取选择 该位设定带字节选择的 SRAM 接口时 $\overline{WEn}$ 和 RD/$\overline{WR}$ 信号的时序。 0：$\overline{WEn}$ 在读写时序时断言， RD/$\overline{WR}$ 在写存取周期时断言 1：$\overline{WEn}$ 在读写存取周期时断言、 RD/$\overline{WR}$ 在写时序时断言
19 ～ 13	—	均为 0	R	保留位 读写的值总是 0。
12、 11	SW[1:0]	00	R/W	地址、 $\overline{CS6}$ 断言 →$\overline{RD}$、 $\overline{WEn}$ 断言的延迟周期数 该位指定从地址、 $\overline{CS6}$ 断言到 $\overline{RD}$、 $\overline{WEn}$ 断言的延迟周期数。 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	存取等待周期数 该位指定读写存取所需的等待周期数。 0000：不插入等待周期 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 （禁止设定） 1110：保留 （禁止设定） 1111：保留 （禁止设定）
6	WM	0	R/W	指定外部等待屏蔽 该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，该位的设定也有效。 0：外部等待输入有效 1：忽视外部等待输入
5 ～ 2	—	均为 0	R	保留位 读写的值总是 0。
1、0	HW[1:0]	00	R/W	$\overline{\text{RD}}$、$\overline{\text{WEn}}$ 取反 → 地址、$\overline{\text{CS6}}$ 取反的延迟周期数 该位指定从 $\overline{\text{RD}}$、$\overline{\text{WEn}}$ 取反到地址、$\overline{\text{CS6}}$ 取反的延迟周期数。 00：0.5 个周期 01：1.5 个周期 10：2.5 个周期 11：3.5 个周期

（2） 突发 ROM （时钟异步）

- CS0WCR



位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	BST[1:0]		-	-	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	R	R/W	R/W

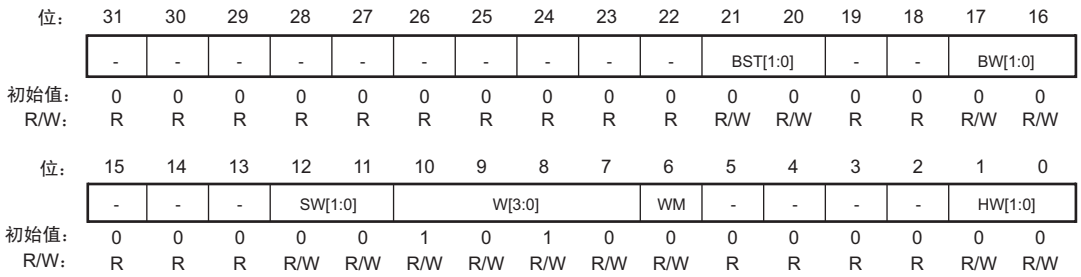
位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	W[3:0]				WM	-	-	-	-	-	-
初始值：	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W：	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

位	位名	初始值	R/W	说　　明
31 ～ 22	—	均为 0	R	保留位 读写的值总是 0。
21、 20	BST[1:0]	00	R/W	指定突发数 该位指定在发生 16 字节存取时的突发数。 BST[1:0] 的 B'11 设定为保留，因此不能设定。
19、 18	—	均为 0	R	保留位 读写的值总是 0。
17、 16	BW[1:0]	00	R/W	突发等待周期数 该位指定要插入到第 2 次或以后存取周期 （突发存取时）的等待周期数。 00：不插入等待周期 01：1 个周期 10：2 个周期 11：3 个周期
15 ～ 11	—	均为 0	R	保留位 读写的值总是 0。

总线宽度	BST[1:0]	突发数
8位	00	16突发×1次
	01	4突发×4次
16位	00	8突发×1次
	01	2突发×4次
	10	4-4或2-4-2突发
32位	xx	4突发×1次

位	位名	初始值	R/W	说　　明
10 ～ 7	W[3:0]	1010	R/W	存取等待周期数 该位指定要插入到第 1 次存取周期的等待周期数。 0000：不插入等待周期 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 （禁止设定） 1110：保留 （禁止设定） 1111：保留 （禁止设定）
6	WM	0	R/W	指定外部等待屏蔽 该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，该位的设定也有效。 0：外部等待输入有效 1：忽视外部等待输入
5 ～ 0	—	均为 0	R	保留位 读写的值总是 0。

- CS4WCR



位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	BST[1:0]		-	-	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	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 ～ 22	—	均为 0	R	保留位 读写的值总是 0。
21、 20	BST[1:0]	00	R/W	突发数的指定 该位指定在发生 16 字节存取时的突发数。 BST[1:0] 的 B'11 设定为保留，因此不能设定。
19、 18	—	均为 0	R	保留位 读写的值总是 0。
17、 16	BW[1:0]	00	R/W	突发等待周期数 该位指定要插入到第 2 次或以后存取周期 （突发存取时）的等待周期数。 00：不插入等待周期 01：1 个周期 10：2 个周期 11：3 个周期
15 ～ 13	—	均为 0	R	保留位 读写的值总是 0。
12、 11	SW[1:0]	00	R/W	地址、 $\overline{\text{CS4}}$ 断言 →$\overline{\text{RD}}$、 $\overline{\text{WEn}}$ 断言的延迟周期数 该位指定从地址、 $\overline{\text{CS4}}$ 断言到 $\overline{\text{RD}}$、 $\overline{\text{WEn}}$ 断言的延迟周期数。 00：0.5 个周期 01：1.5 个周期 10：2.5 个周期 11：3.5 个周期

总线宽度	BST[1:0]	突发数
8位	00	16突发×1次
	01	4突发×4次
16位	00	8突发×1次
	01	2突发×4次
	10	4-4或2-4-2突发
32位	xx	4突发×1次

位	位名	初始值	R/W	说　　明
10 ～ 7	W[3:0]	1010	R/W	存取等待周期数 该位指定要插入到第 1 次存取周期的等待周期数。 0000：不插入等待周期 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 （禁止设定） 1110：保留 （禁止设定） 1111：保留 （禁止设定）
6	WM	0	R/W	指定外部等待屏蔽 该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，该位的设定也有效。 0：外部等待输入有效 1：忽视外部等待输入
5 ～ 2	—	均为 0	R	保留位 读写的值总是 0。
1、0	HW[1:0]	00	R/W	$\overline{\text{RD}}$、$\overline{\text{WEn}}$ 取反 → 地址、$\overline{\text{CS4}}$ 取反的延迟周期数 该位指定从 $\overline{\text{RD}}$、$\overline{\text{WEn}}$ 取反到地址、$\overline{\text{CS4}}$ 取反的延迟周期数。 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） SDRAM*

- 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	—	均为 0	R	保留位 读写的值总是 0。
10	—	1	R	保留位 读写的值总是 0。
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 等待时间 该位指定区域 2 的 CAS 等待时间。 00：1 个周期 01：2 个周期 10：3 个周期 11：4 个周期
6 ～ 0	—	均为 0	R	保留位 读写的值总是 0。

【注】 * 只有 1 个区域连接 SDRAM 时，请将区域 3 设定为 SDRAM。此时将区域 2 设定为通常空间或带字节选择的 SRAM。

- 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

【注】* 区域2和区域3都设定为SDRAM时，将WTRP[1:0]、WTRCD[1:0]、TRWL[1:0]、WTRC[1:0] 位设定为通用。

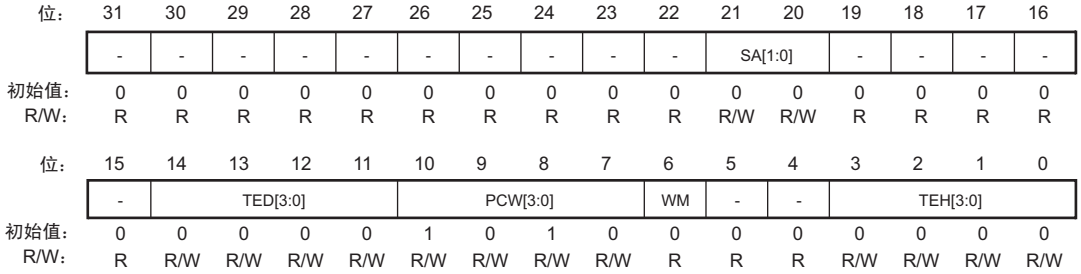
位	位名	初始值	R/W	说　　明
31 ～ 15	—	均为 0	R	保留位 读写的值总是 0。
14、 13	WTRP[1:0]*	00	R/W	预充电结束的等待周期数 指定以下的等待预充电结束的最小周期数。 • 从启动自动预充电到对同一存储体发送 ACTV 命令 • 从发送 PRE/PALL 命令到对同一存储体发送 ACTV 命令 • 转移为掉电模式 / 深度掉电模式之前 • 自动刷新时从发送 PALL 命令到发送 REF 命令 • 自刷新时从发送 PALL 命令到发送 SELF 命令 区域 2 和区域 3 的设定是通用的。 00：不插入等待周期 01：1 个周期 10：2 个周期 11：3 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 该位指定从发送 ACTV 命令后到发送 READ （A） /WRIT （A）命令前的最小等待周期数。 区域 2 和区域 3 的设定是通用的。 00：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 的等待时间 该位指定区域 3 的 CAS 等待时间。 00：1 个周期 01：2 个周期 10：3 个周期 11：4 个周期

位	位名	初始值	R/W	说　　明
6、5	—	均为 0	R	保留位 读写的值总是 0。
4、3	TRWL[1:0]*	00	R/W	启动预充电的等待周期数 指定以下的等待预充电启动的最小周期数。 • 从本 LSI 发送 WRITA 命令到在 SDRAM 内启动自动预充电的周期数发送 WRITEA 命令后，到对同一存储体发送 ACTV 命令之前的周期数。另外，请通过各 SDRAM 的数据表确认在 SDRAM 内从接受 WRITA 命令到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。 • 本 LSI 从发送 WRIT 命令到发送 PRE 命令的周期数 在存储体激活模式中，存取同一存储体内的不同行地址时。 区域 2 和区域 3 的设定是通用的。 00：不插入等待周期 01：1 个周期 10：2 个周期 11：3 个周期
2	—	0	R	保留位 读写的值总是 0。
1、0	WTRC[1:0]*	00	R/W	REF 命令 / 自刷新解除 →ACTV/REF/MRS 命令之间的空闲周期数 指定以下命令之间的最小空闲周期数。 • 从发送 REF 命令到发送 ACTV/REF/MRS 命令 • 从解除自刷新到发送 ACTV/REF/MRS 命令 区域 2 和区域 3 的设定是通用的。 00：2 个周期 01：3 个周期 10：5 个周期 11：8 个周期

【注】 * 区域 2 和区域 3 都设定为 SDRAM 时， WTRP[1:0]、 WTRCD[1:0]、 TRWL[1:0]、 WTRC[1:0] 位成为通用设定。
只有区域1连接SDRAM时，请将区域3设定为SDRAM。此时将区域2设定为通常空间或带字节选择的SRAM。

（4） 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

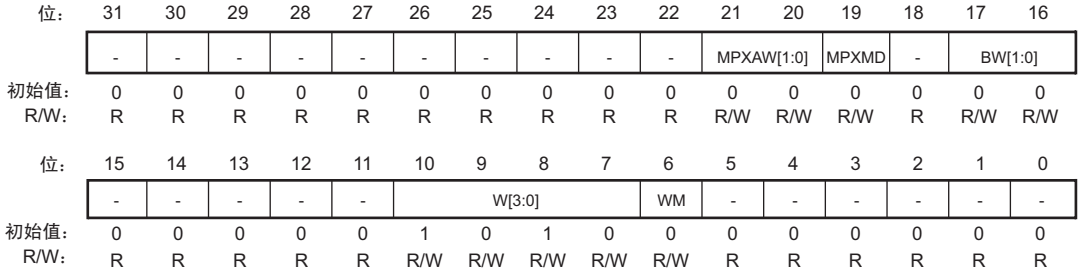
位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	TED[3:0]				PCW[3:0]				WM	-	-	TEH[3:0]			
初始值：	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W：	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说　　明
31 ～ 22	—	均为 0	R	保留位 读写的值总是 0。
21、 20	SA[1:0]	00	R/W	指定空间属性 设定 PCMCIA 接口时，该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	—	均为 0	R	保留位 读写的值总是 0。
14 ～ 11	TED[3:0]	0000	R/W	地址 -$\overline{RD}$/$\overline{WE}$ 断言的延迟 该位设定（在 PCMCIA 接口）从地址输出到存储器卡时的 $\overline{RD}$/$\overline{WE}$ 断言或到 I/O 卡时的 $\overline{ICIORD}$/$\overline{ICIOWR}$ 断言的延迟时间。 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	存取的等待周期数 该位设定要插入的等待周期数。 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
6	WM	0	R/W	指定外部等待屏蔽 该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，该位的设定也有效。 0：外部等待输入有效 1：忽视外部等待输入
5、4	—	均为 0	R	保留位 读写的值总是 0。
3 ～ 0	TEH[3:0]	0000	R/W	$\overline{RD}$/$\overline{WE}$ 取反—地址延迟 该位设定 （在 PCMCIA 接口）来自存储器卡时 $\overline{RD}$/$\overline{WE}$ 取反或来自 I/O 卡时 $\overline{ICIORD}$ 和 $\overline{ICIOWR}$ 取反的地址保持时间。 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

（5） 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	—	均为 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 该位设定要插入到地址周期的等待数。 00：不插入等待周期 01：1 个周期 10：2 个周期 11：3 个周期
19	MPXMD	0	R/W	指定突发 MPX-I/O 接口的模式 该位指定 16 字节存取时的存取模式。 0：以 16 字节 （传送长度）进行 4 突发存取 1 次 1：以 8 字节 （传送长度为四字）进行 2 突发存取 2 次 MPXMD=0 时的传送长度

D31	D30	D29	传送长度
0	0	0	字节（1字节）
		1	字（2字节）
	1	0	长字（4字节）
		1	保留四字（8字节）
1	0	0	16 字节
		1	保留（32 字节）
	1	0	保留（64 字节）

MPXMD=1 时的传送长度

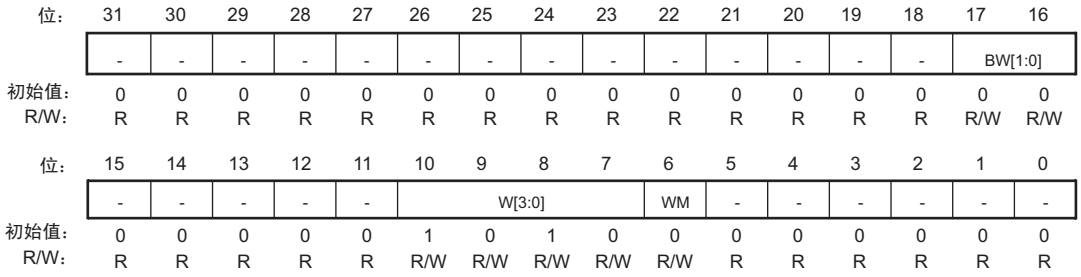
D31	D30	D29	传送长度
0	0	0	字节（1字节）
		1	字（2字节）
	1	0	长字（4字节）
		1	四字（8字节）
1	0	0	保留（32字节）

18	—	0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说　　明
17、 16	BW[1:0]	00	R/W	突发的等待周期数 该位指定要插入到第 2 次或以后存取周期 （突发存取时）的等待周期数。 00：不插入等待周期 01：1 个周期 10：2 个周期 11：3 个周期
15 ～ 11	—	均为 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 该位指定要插入到第 1 次突发存取周期或者单存取的等待周期数。 0000：不插入等待周期 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 （禁止设定） 1110：保留 （禁止设定） 1111：保留 （禁止设定）
6	WM	0	R/W	指定外部等待屏蔽 该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，该位的设定也有效。 0：外部等待输入有效 1：忽视外部等待输入
5 ～ 0	—	均为 0	R	保留位 读写的值总是 0。

（6） 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	—	均为 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 该位指定要插入到第 2 次或以后存取周期 （突发存取时）的等待周期数。 00：不插入等待周期 01：1 个周期 10：2 个周期 11：3 个周期
15 ～ 11	—	均为 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 该位指定要插入到第 1 次存取周期的等待周期数。 0000：不插入等待周期 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 （禁止设定） 1110：保留 （禁止设定） 1111：保留 （禁止设定）
6	WM	0	R/W	指定外部等待屏蔽 该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，该位的设定也有效。 0：外部等待输入有效 1：忽视外部等待输入
5 ～ 0	—	均为 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、存取方法和要连接的 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
	-	-	DEEP	SLOW	RFSH	RMODE	PDOWN	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/W	R/W	R/W	R/W	R/W	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	—	均为 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 该位指定区域 2 行地址的位数。 00：11 位 01：12 位 10：13 位 11：保留 （禁止设定）
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 该位指定区域 2 列地址的位数。 00：8 位 01：9 位 10：10 位 11：保留 （禁止设定）
15、 14	—	均为 0	R	保留位 读写的值总是 0。
13	DEEP	0	R/W	深度掉电模式 只对低功耗 SDRAM 有效。在该位置 1 的状态下，将 RFSH 位或 RMODE 位置 1 时，发送深度掉电入口命令，低功率 SDRAM 转换为深度掉电模式。 0：自刷新模式 1：深度掉电模式

位	位名	初始值	R/W	说　　明
12	SLOW	0	R/W	低频率模式 该位指定对 SDRAM 输出命令、地址、写数据的时序，和输入读数据的时序。 该位设定为 1 时，在比正常慢半个周期的 CKIO 的下降沿同步输出 SDRAM 的命令、地址和写数据。另外，在比正常快半个周期的 CKIO 的下降沿输入来自 SDRAM 的读数据。因此可延长命令、地址、写数据和读数据的保持时间。该模式适合 SDRAM 低频率运行时。 0：在 CKIO 的上升沿同步输出对 SDRAM 的命令、地址和写数据。 在 CKIO 的上升沿同步输入来自 SDRAM 的读数据。 1：在 CKIO 的下降沿同步输出对 SDRAM 的命令、地址和写数据。 在 CKIO 的下降沿同步输入来自 SDRAM 的读数据。
11	RFSH	0	R/W	刷新控制 该位指定是否刷新 SDRAM。 0：不刷新 1：刷新
10	RMODE	0	R/W	刷新控制 该位指定当 RFSH 位为 1 时，是进行自动刷新还是进行自刷新。 当 RFSH 位和该位都设定为 1 时，设定后立即进入自刷新模式。 当RFSH位设定为1而该位设定为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	PDOWN	0	R/W	掉电模式 该位指定对 SDRAM 的存取结束后， SDRAM 是否成为掉电模式。 该位设定为 1 时，存取结束后 CKE 引脚为低电平， SDRAM 为掉电模式。 0：存取结束后， SDRAM 不是掉电模式 1：存取结束后， SDRAM 是掉电模式
8	BACTV	0	R/W	存储体激活模式 该位指定是在自动预充电模式下（使用 READA 命令和 WRITA 命令）还是在存储体激活模式下 （使用 READ 命令和 WRIT 命令）进行存取。 0：自动预充电模式 （使用 READA 命令和 WRITA 命令） 1：存储体激活模式 （使用 READ 命令和 WRIT 命令） 【注】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。 将区域 2 和区域 3 都设定为 SDRAM 时，就必须设定为自动预充电模式。
7 ～ 5	—	均为 0	R	保留位 读写的值总是 0。
4、3	A3ROW[1:0]	00	R/W	区域 3 行地址的位数 该位指定区域 3 行地址的位数。 00：11 位 01：12 位 10：13 位 11：保留 （禁止设定）

位	位名	初始值	R/W	说　　明
2	—	0	R	保留位 读写的值总是 0。
1、0	A3COL[1:0]	00	R/W	区域 3 列地址的位数 该位指定区域 3 列地址的位数。 00：8 位 01：9 位 10：10 位 11：保留 （禁止设定）

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。

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

请注意：因为对刷新定时器的计数器 （RTCNT）进行递增计数的时钟只有在上电复位时才进行相位的匹配，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	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
	-	-	-	-	-	-	-	-	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 ～ 8	—	均为 0	R	保留位 读写的值总是 0。
7	CMF	0	R/W	比较匹配标志 该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）的值匹配的状态标志。 在以下项目下进行置位 / 清除。 0：清除项目：在 CMF=1 的状态下，读 RTCSR 后给 CMF 写 0 时 1：置位项目：当 RTCNT=RTCOR 时
6	CMIE	0	R/W	比较匹配的中断允许 该位设定在 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

位	位名	初始值	R/W	说　　明
5 ～ 3	CKS[2:0]	000	R/W	时钟选择 该位选择对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：Bϕ/4 010：Bϕ/16 011：Bϕ/64 100：Bϕ/256 101：Bϕ/1024 110：Bϕ/2048 111：Bϕ/4096
2 ～ 0	RRC[2:0]	000	R/W	刷新次数 该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）的值匹配后产生刷新请求时的连续刷新次数。通过该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 （禁止设定） 110：保留 （禁止设定） 111：保留 （禁止设定）

9.4.6　刷新定时器的计数器 （RTCNT）

RTCNT 是 8 位计数器，对通过 RTCSR 的 CKS[2:0] 位选择的时钟进行递增计数。 RTCNT 和 RTCOR 的值匹配时， RTCNT 被清 0。当递增计数到 255 时，就返回 0。在写入 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	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
	-	-	-	-	-	-	-	-								
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 ～ 8	—	均为 0	R	保留位 读取时值总是 0。
7 ～ 0		均为 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 的值匹配时， RTCSR 的 CMF 位被置 1， RTCNT 被清 0。

将 SDCR 的 RFSH 位置 1 时，通过该匹配信号产生刷新请求。刷新请求被保持到实际刷新为止。在下一个匹配信号前还没有处理刷新请求时，前一个请求就无效。

总线释放过程中产生刷新请求时，可断言 $\overline{\text{REFOUT}}$ 信号。详细情况请参照 **“9.5.6 （9） 刷新请求和总线周期的关系”**或 **“9.5.13 总线仲裁”**。

将 RTCSR 的 CMIE 位置 1 时，通过该匹配信号产生中断请求。在 RTCSR 的 CMF 位被清除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，可边进行刷新边使用中断对刷新请求的次数进行计数等，也可同时设定刷新和间隔定时器的中断。

在写入 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	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
	-	-	-	-	-	-	-	-								
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 ～ 8	—	均为 0	R	保留位 读取时的值总是 0。
7 ～ 0		均为 0	R/W	8 位寄存器

9.5　运行说明

9.5.1　端法 / 存取长度和数据对齐

本 LSI 支持的字节数据排列方法为大端法和小端法，大端法的高位字节 （MSB）靠近地址 0 方向，而小端法的低位字节 （LSB）靠近地址 0 方向。在上电复位后，所有区域的初始状态均为大端法。区域 0 空间不能为小端法，但区域 1 ～ 7 空间在不存取对象空间时，可通过 CSnBCR 寄存器的设定改变端法。

另外，对区域 1 ～ 7，通用存储器和带字节选择的 SRAM 可选择 8 位、16 位或 32 位数据总线宽度。SDRAM 可选择 16 位或 32 位数据总线。PCMCIA 接口可选择 8 位或 16 位数据总线宽度。MPX-I/O 为 8 位或 16 位固定宽度的数据总线，或根据存取地址为 8 位或 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对齐，因此欲从 8 位宽的器件中读取长字数据时，就必须读 4 次。在本 LSI 的各接口之间自动进行数据的对齐和数据长度的转换。在区域 0 空间中，通过上电复位时 MD 引脚的设定，固定为 16 或 32 位数据总线宽度。

器件的数据宽度和存取单位的关系如**表 9.5 ～表 9.10** 所示。请注意， 32 位总线宽度和 16 位总线宽度时的选通信号的对应地址，因大端法和小端法而不同。例如，在 32 位总线宽度下为大端法时， $\overline{\text{WE3}}$ 指向地址 0 ；为小端法时， $\overline{\text{WE0}}$ 指向地址 0。通过小端法存取指令代码时，取长字和取字混在一起，而且可能以字节为单位读取数据，因此请将外部存储器设置为 8 位宽度。考虑到存取效率，建议用大端法执行外部存储器的取指令。

表 9.5 （大端法） 32 位外部器件的存取和数据对齐

操作	数据总线				选通信号			
	D31 ～ D24	D23 ～ D16	D15 ～ D8	D7 ～ D0	$\overline{\text{WE3}}$、DQMUU	$\overline{\text{WE2}}$、DQMUL	$\overline{\text{WE1}}$、DQMLU	$\overline{\text{WE0}}$、DQMLL
地址 0 的字节存取	数据 7 ～ 0	—	—	—	断言	—	—	—
地址 1 的字节存取	—	数据 7 ～ 0	—	—	—	断言	—	—
地址 2 的字节存取	—	—	数据 7 ～ 0	—	—	—	断言	—
地址 3 的字节存取	—	—	—	数据 7 ～ 0	—	—	—	断言
地址 0 的字存取	数据 15 ～ 8	数据 7 ～ 0	—	—	断言	断言	—	—
地址 2 的字存取	—	—	数据 15 ～ 8	数据 7 ～ 0	—	—	断言	断言
地址 0 的长字存取	数据 31 ～ 24	数据 23 ～ 16	数据 15 ～ 8	数据 7 ～ 0	断言	断言	断言	断言

表 9.6 （大端法） 16 位外部器件的存取和数据对齐

操作		数据总线				选通信号			
		D31 ～ D24	D23 ～ D16	D15 ～ D8	D7 ～ D0	$\overline{WE3}$、DQMUU	$\overline{WE2}$、DQMUL	$\overline{WE1}$、DQMLU	$\overline{WE0}$、DQMLL
地址 0 的字节存取		—	—	数据 7 ～ 0	—	—	—	断言	—
地址 1 的字节存取		—	—	—	数据 7 ～ 0	—	—	—	断言
地址 2 的字节存取		—	—	数据 7 ～ 0	—	—	—	断言	—
地址 3 的字节存取		—	—	—	数据 7 ～ 0	—	—	—	断言
地址 0 的字存取		—	—	数据 15 ～ 8	数据 7 ～ 0	—	—	断言	断言
地址 2 的字存取		—	—	数据 15 ～ 8	数据 7 ～ 0	—	—	断言	断言
地址 0 的长字存取	第 1 次（地址 0）	—	—	数据 31 ～ 24	数据 23 ～ 16	—	—	断言	断言
	第 2 次（地址 2）	—	—	数据 15 ～ 8	数据 7 ～ 0	—	—	断言	断言

表 9.7 （大端法） 8 位外部器件的存取和数据对齐

操作		数据总线				选通信号			
		D31 ～ D24	D23 ～ D16	D15 ～ D8	D7 ～ D0	$\overline{WE3}$、DQMUU	$\overline{WE2}$、DQMUL	$\overline{WE1}$、DQMLU	$\overline{WE0}$、DQMLL
地址 0 的字节存取		—	—	—	数据 7 ～ 0	—	—	—	断言
地址 1 的字节存取		—	—	—	数据 7 ～ 0	—	—	—	断言
地址 2 的字节存取		—	—	—	数据 7 ～ 0	—	—	—	断言
地址 3 的字节存取		—	—	—	数据 7 ～ 0	—	—	—	断言
地址 0 的字存取	第 1 次 （地址 0）	—	—	—	数据 15 ～ 8	—	—	—	断言
	第 2 次 （地址 1）	—	—	—	数据 7 ～ 0	—	—	—	断言
地址 2 的字存取	第 1 次 （地址 2）	—	—	—	数据 15 ～ 8	—	—	—	断言
	第 2 次 （地址 3）	—	—	—	数据 7 ～ 0	—	—	—	断言
地址 0 的长字存取	第 1 次 （地址 0）	—	—	—	数据 31 ～ 24	—	—	—	断言
	第 2 次 （地址 1）	—	—	—	数据 23 ～ 16	—	—	—	断言
	第 3 次 （地址 2）	—	—	—	数据 15 ～ 8	—	—	—	断言
	第 4 次 （地址 3）	—	—	—	数据 7 ～ 0	—	—	—	断言

表 9.8　（小端法） 32 位外部器件的存取和数据对齐

操作	数据总线				选通信号			
	D31 ～ D24	D23 ～ D16	D15 ～ D8	D7 ～ D0	$\overline{WE3}$、DQMUU	$\overline{WE2}$、DQMUL	$\overline{WE1}$、DQMLU	$\overline{WE0}$、DQMLL
地址 0 的字节存取	—	—	—	数据 7 ～ 0	—	—	—	断言
地址 1 的字节存取	—	—	数据 7 ～ 0	—	—	—	断言	—
地址 2 的字节存取	—	数据 7 ～ 0	—	—	—	断言	—	—
地址 3 的字节存取	数据 7 ～ 0	—	—	—	断言	—	—	—
地址 0 的字存取	—	—	数据 15 ～ 8	数据 7 ～ 0	—	—	断言	断言
地址 2 的字存取	数据 15 ～ 8	数据 7 ～ 0	—	—	断言	断言	—	—
地址 0 的长字存取	数据 31 ～ 24	数据 23 ～ 16	数据 15 ～ 8	数据 7 ～ 0	断言	断言	断言	断言

表 9.9　（小端法） 16 位外部器件的存取和数据对齐

操作		数据总线				选通信号			
		D31 ～ D24	D23 ～ D16	D15 ～ D8	D7 ～ D0	$\overline{WE3}$、DQMUU	$\overline{WE2}$、DQMUL	$\overline{WE1}$、DQMLU	$\overline{WE0}$、DQMLL
地址 0 的字节存取		—	—	—	数据 7 ～ 0	—	—	—	断言
地址 1 的字节存取		—	—	数据 7 ～ 0	—	—	—	断言	—
地址 2 的字节存取		—	—	—	数据 7 ～ 0	—	—	—	断言
地址 3 的字节存取		—	—	数据 7 ～ 0	—	—	—	断言	—
地址 0 的字存取		—	—	数据 15 ～ 8	数据 7 ～ 0	—	—	断言	断言
地址 2 的字存取		—	—	数据 15 ～ 8	数据 7 ～ 0	—	—	断言	断言
地址 0 的长字存取	第 1 次（地址 0）	—	—	数据 15 ～ 8	数据 7 ～ 0	—	—	断言	断言
	第 2 次（地址 2）	—	—	数据 31 ～ 24	数据 23 ～ 16	—	—	断言	断言

表 9.10 （小端法） 8 位外部器件的存取和数据对齐

操作		数据总线				选通信号			
		D31 ～ D24	D23 ～ D16	D15 ～ D8	D7 ～ D0	$\overline{WE3}$、DQMUU	$\overline{WE2}$、DQMUL	$\overline{WE1}$、DQMLU	$\overline{WE0}$、DQMLL
地址 0 的字节存取		—	—	—	数据 7 ～ 0	—	—	—	断言
地址 1 的字节存取		—	—	—	数据 7 ～ 0	—	—	—	断言
地址 2 的字节存取		—	—	—	数据 7 ～ 0	—	—	—	断言
地址 3 的字节存取		—	—	—	数据 7 ～ 0	—	—	—	断言
地址 0 的字存取	第 1 次（地址 0）	—	—	—	数据 7 ～ 0	—	—	—	断言
	第 2 次（地址 1）	—	—	—	数据 15 ～ 8	—	—	—	断言
地址 2 的字存取	第 1 次（地址 2）	—	—	—	数据 7 ～ 0	—	—	—	断言
	第 2 次（地址 3）	—	—	—	数据 15 ～ 8	—	—	—	断言
地址 0 的长字存取	第 1 次（地址 0）	—	—	—	数据 7 ～ 0	—	—	—	断言
	第 2 次（地址 1）	—	—	—	数据 15 ～ 8	—	—	—	断言
	第 3 次（地址 2）	—	—	—	数据 23 ～ 16	—	—	—	断言
	第 4 次（地址 3）	—	—	—	数据 31 ～ 24	—	—	—	断言

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 而输出选通信号。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 个周期断言。

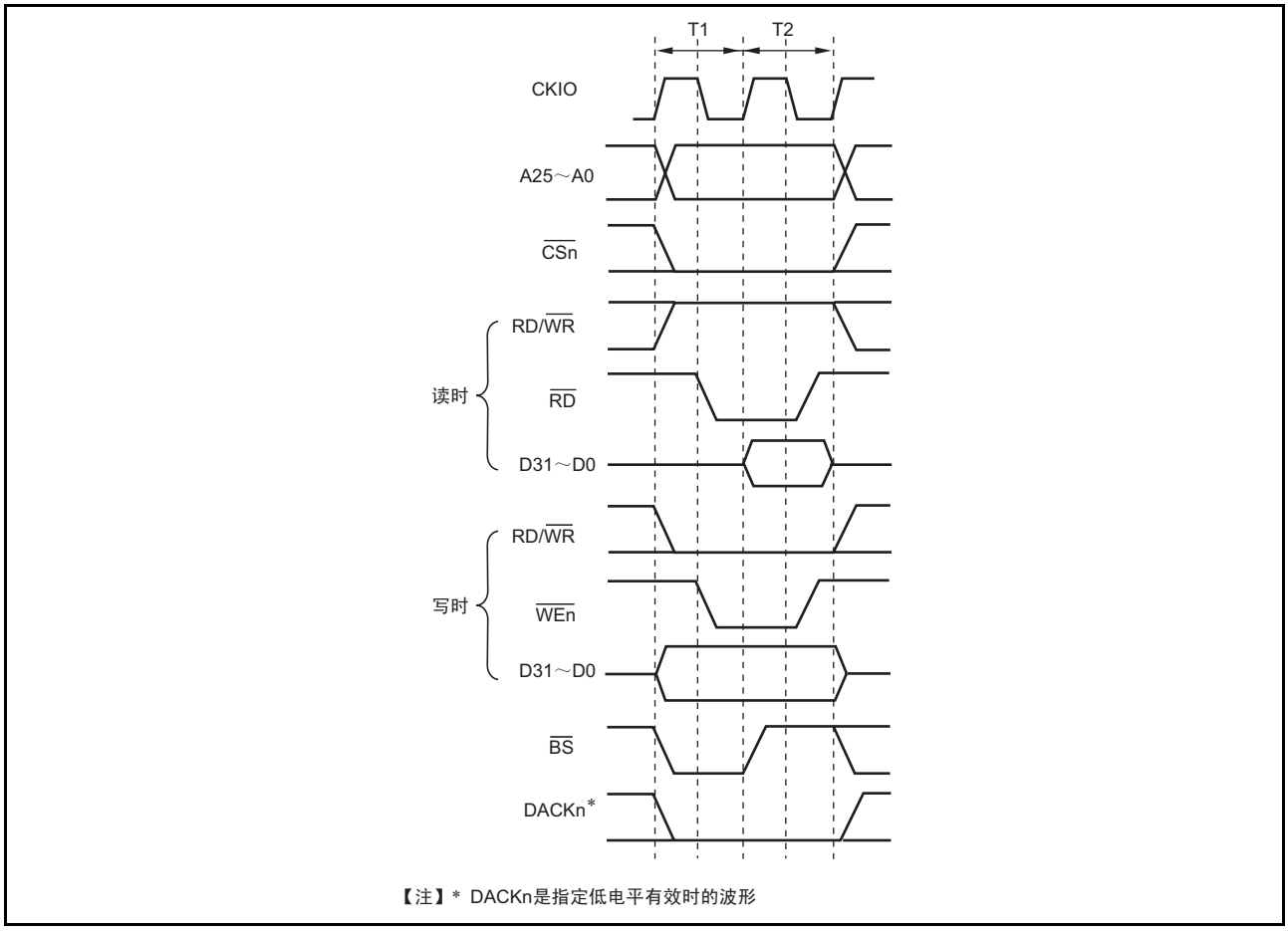


图 9.2　通常空间的基本存取 （存取等待 0）

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有效位却没有指定存取长度，所以对于 32 位器件总是读出 32 位数据，而对于 16 位器件总是读出 16 位数据。在进行写操作时，只有被写入字节的 $\overline{WEn}$ 信号才断言。

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。请注意：因为在没有存取时 RD/$\overline{WR}$ 信号为读取状态 （高电平输出），如果使用 RD/$\overline{WR}$ 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、图 9.4 所示。将 CSnWCR 的 WM 位设定为 0 时，为了评价存取设定 CSn 空间后的外部等待而插入 1 个 Tnop 周期 （图 9.3）。将 CSnWCR 的 WM 位设定为 1 时，忽视外部等待并能抑止 Tnop 周期的插入 （图 9.4）。

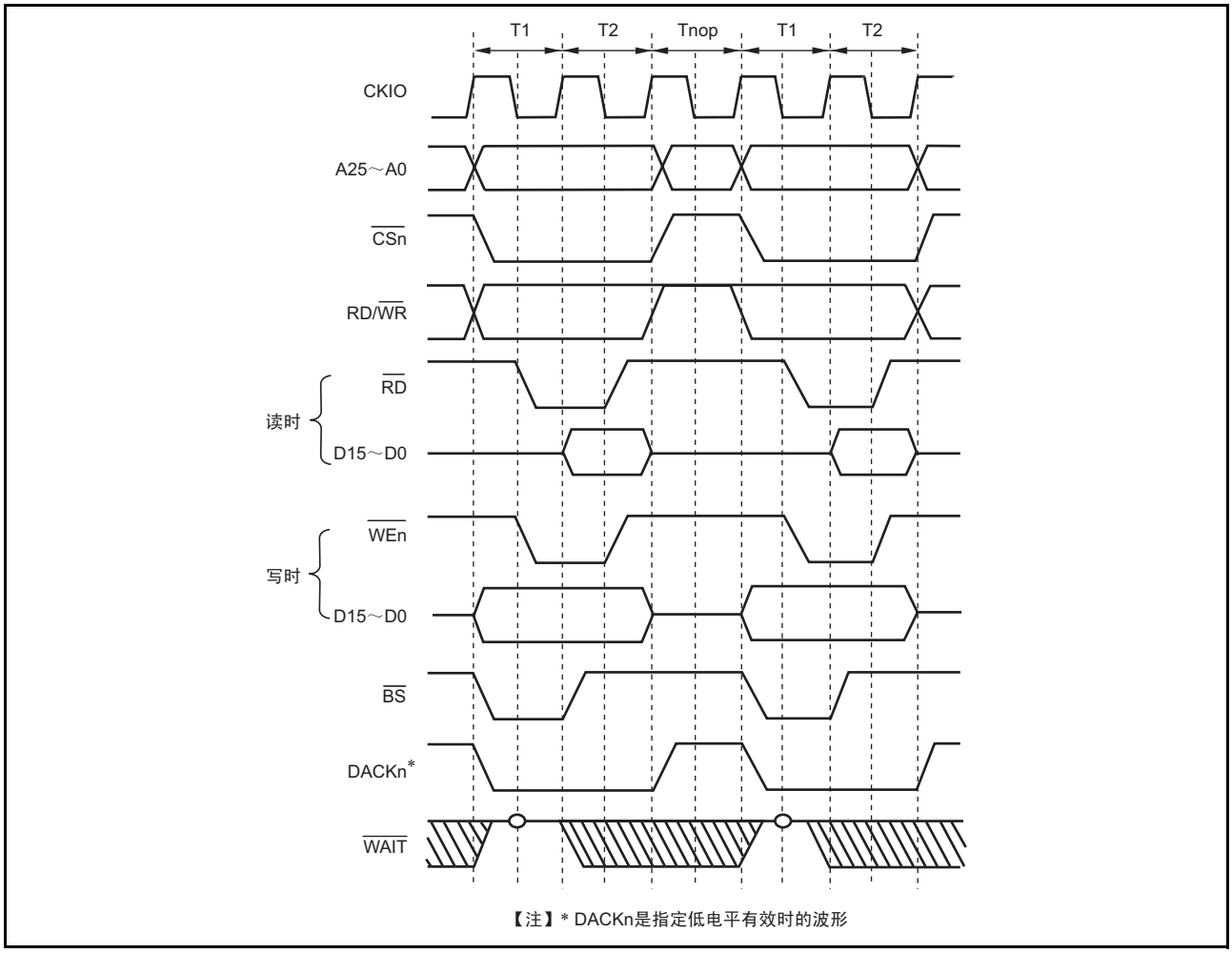


图 9.3　通常空间的连续存取例 1
16 位总线宽度、长字存取、 CSnWCR.WM 位 =0
（存取周期 0、周期之间的等待 0）

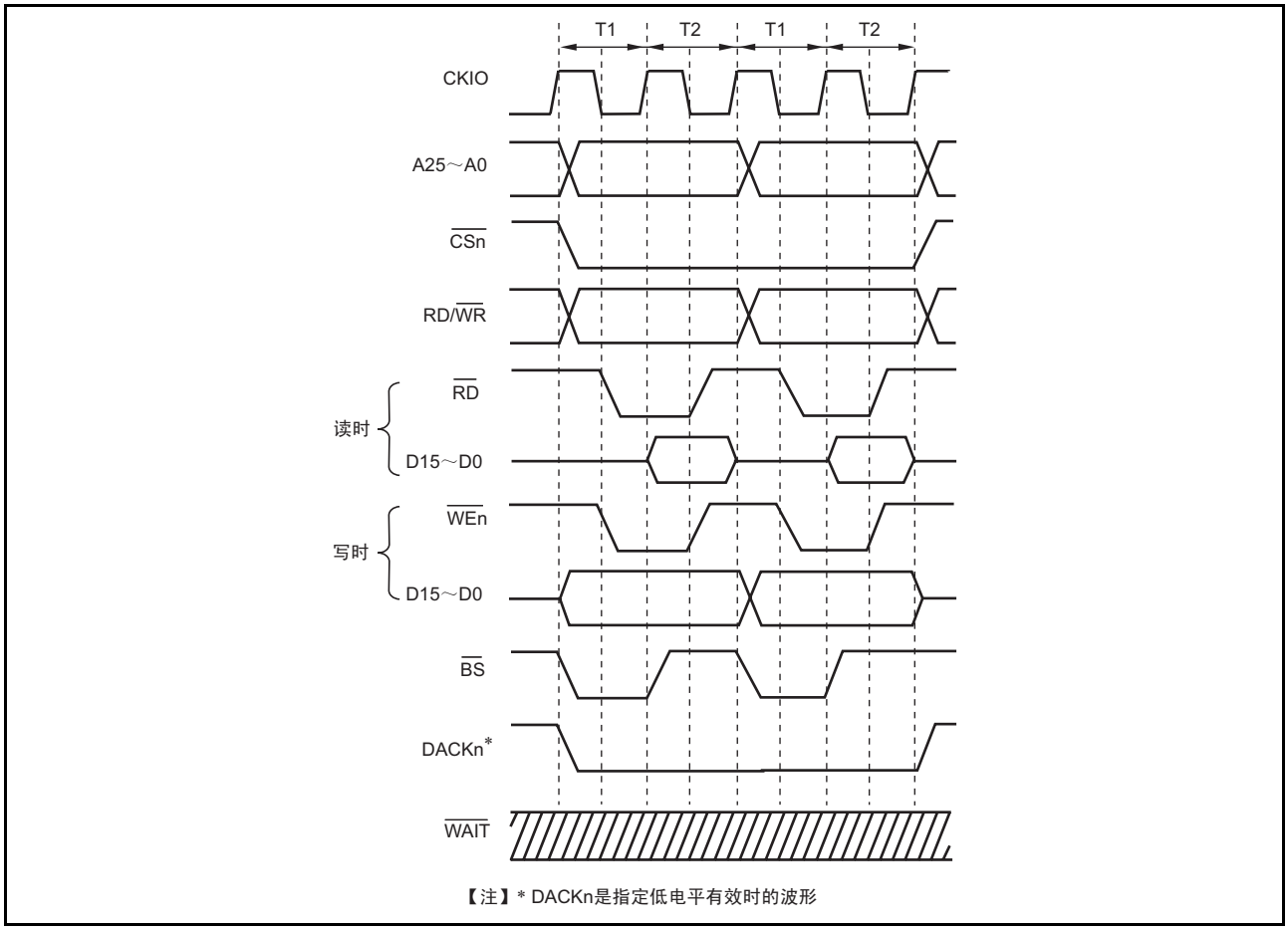


图 9.4　通常空间的连续存取例 2
16 位总线宽度、长字存取、 CSnWCR.WM 位 =1
（存取等待 0、周期之间的等待 0）

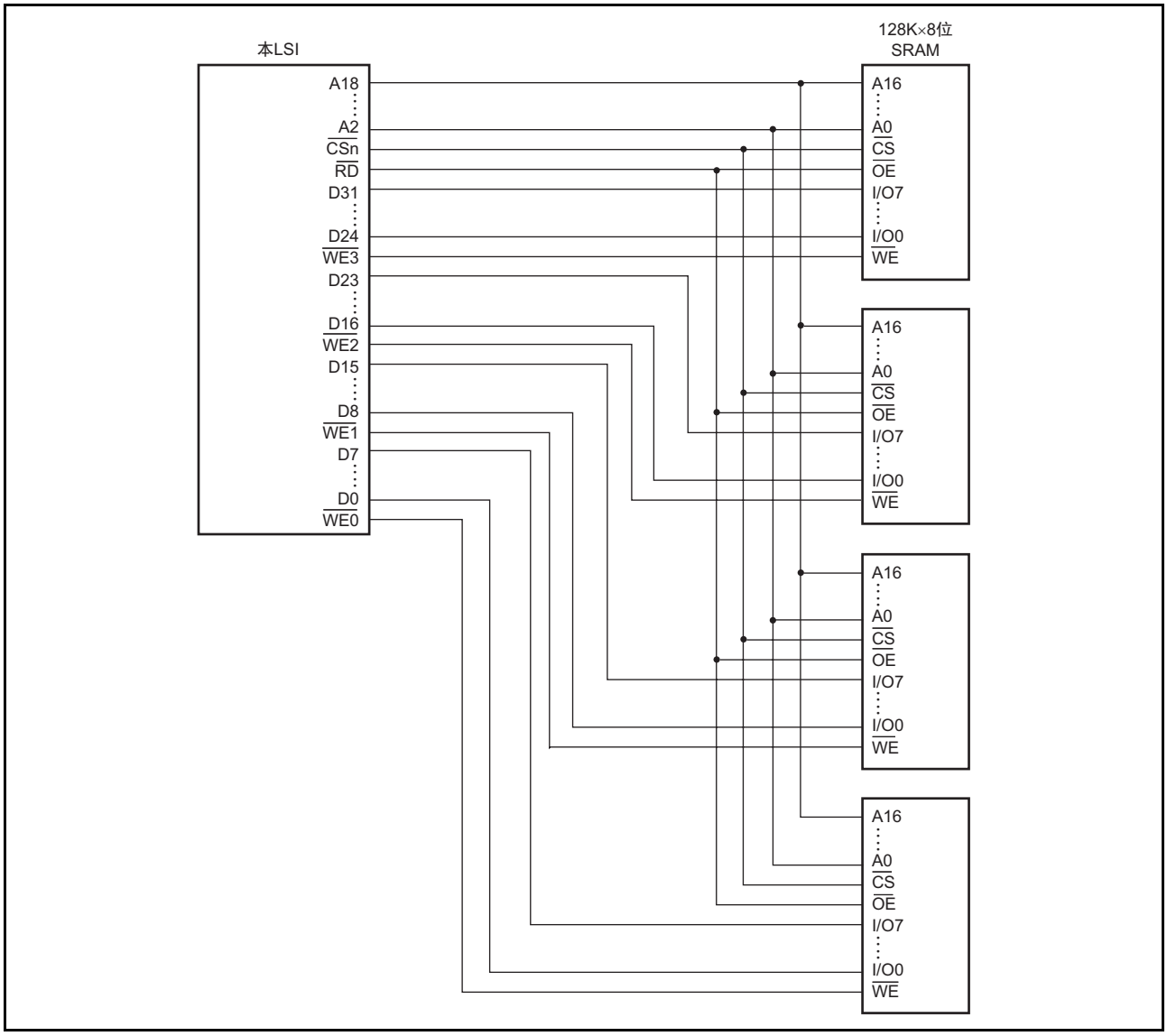


图 9.5　32 位数据宽度的 SRAM 连接例子

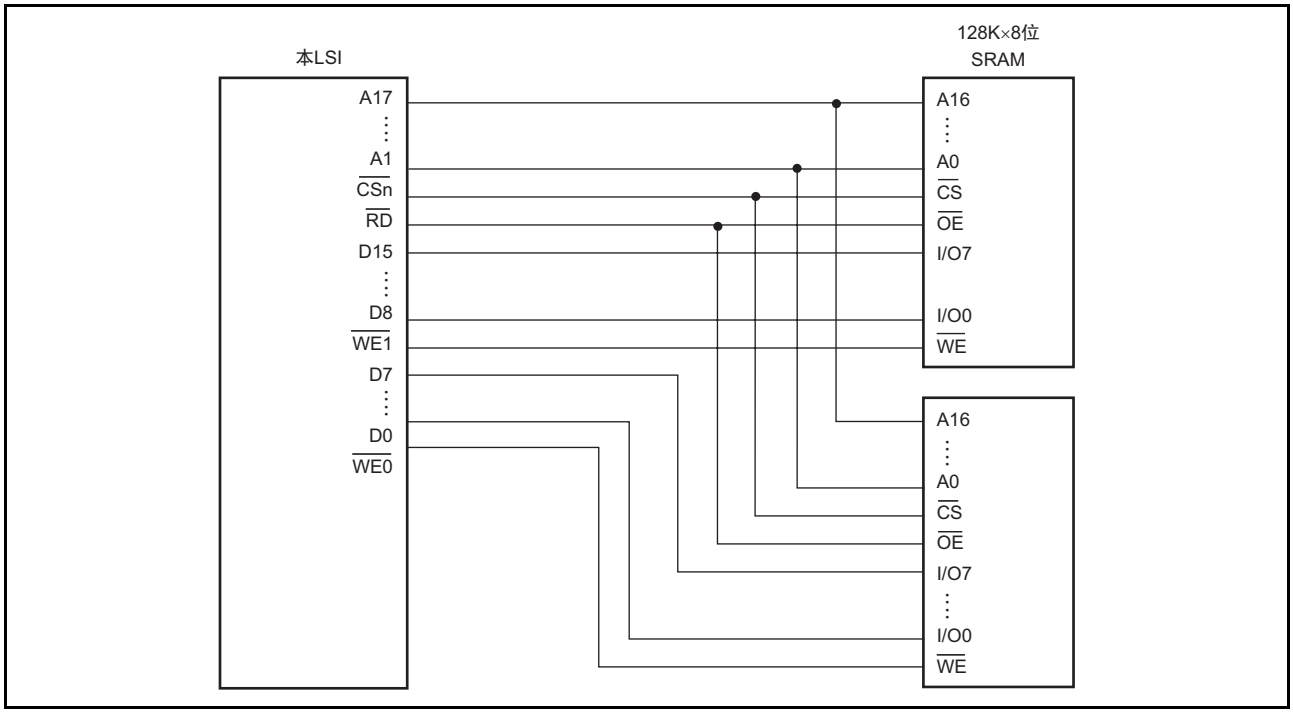


图 9.6　16 位数据宽度的 SRAM 连接例子

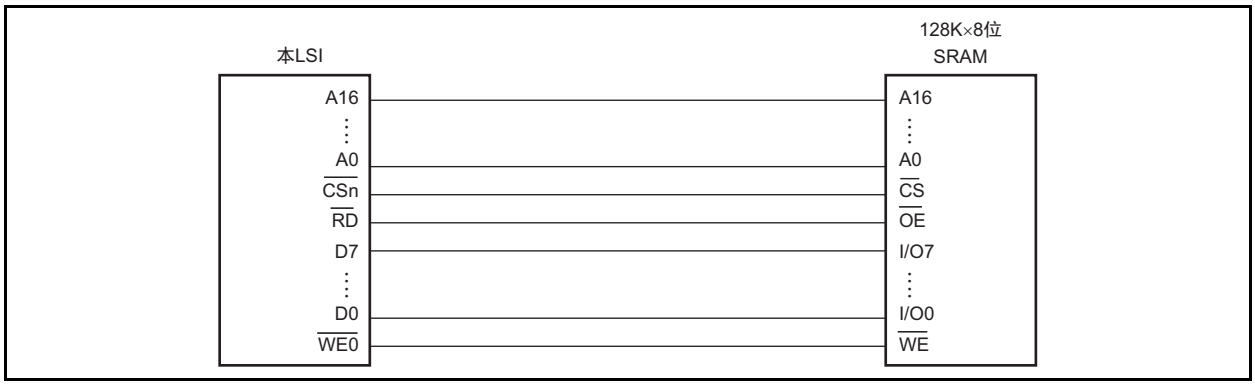


图 9.7　8 位数据宽度的 SRAM 连接例子

9.5.3　存取等待的控制

能通过设定 CSnWCR 的 WR[3:0] 位，控制通常空间存取的等待周期的插入。对于区域 1、4、5、7 和区域 8，能在读 / 写存取中独立插入等待周期。区域 0、2、3 和区域 6 的存取等待在读 / 写周期中通用。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 Tw 周期数。

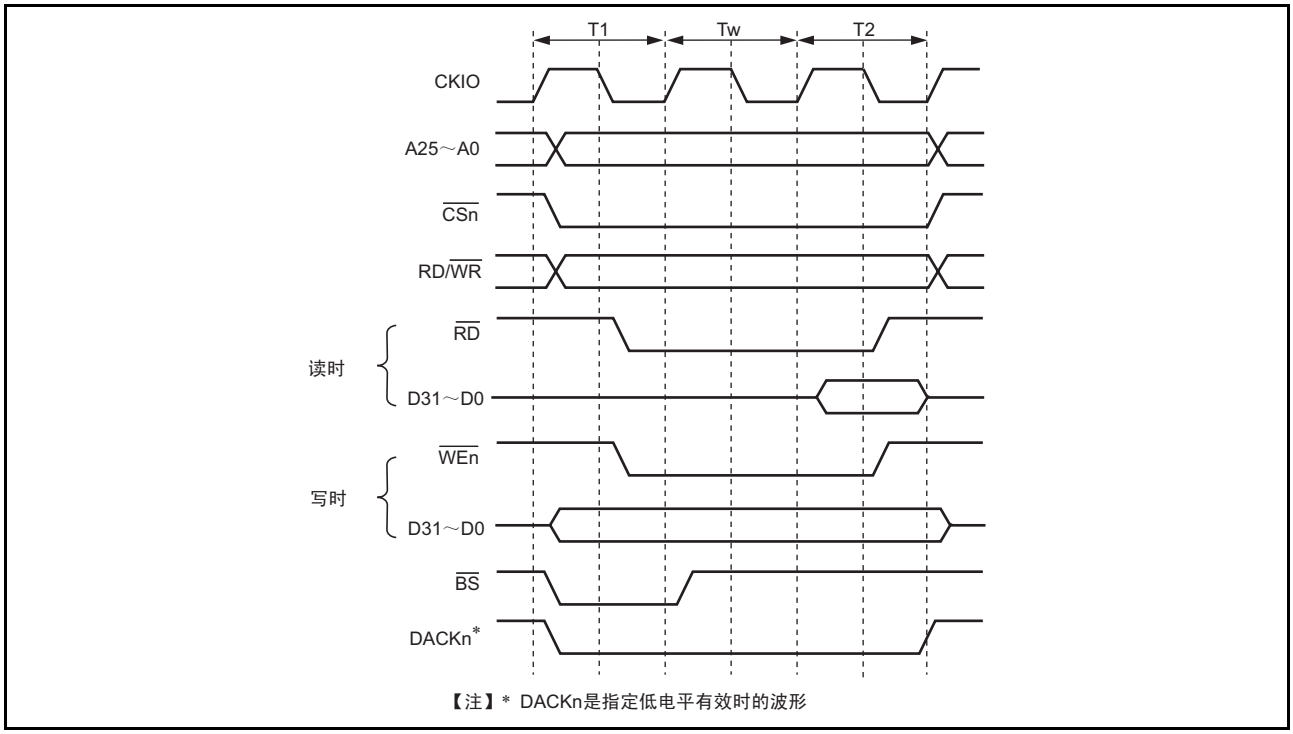


图 9.8　通常空间存取的等待时序 （只限软件等待）

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 CKIO 的下降沿对 $\overline{\text{WAIT}}$ 信号进行采样。

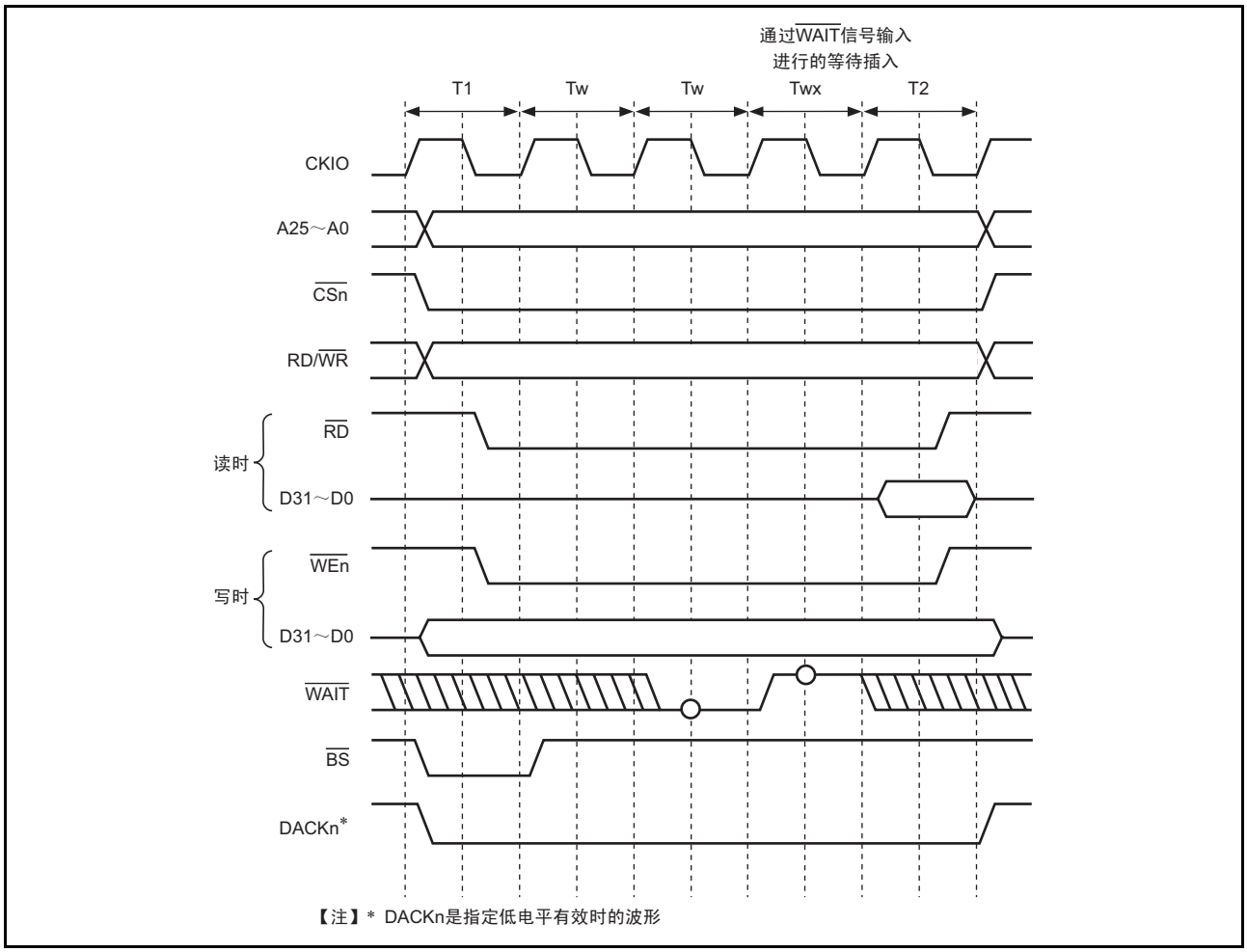


图 9.9　通常空间存取的等待时序 （通过 $\overline{\text{WAIT}}$ 信号进行的等待插入）

9.5.4　$\overline{\text{CSn}}$ 断言期间的扩展

能通过设定 CSnWCR 的 SW[1:0] 位，指定从 $\overline{\text{CSn}}$ 断言到 $\overline{\text{RD}}$ 和 $\overline{\text{WEn}}$ 断言的周期数。还能通过设定 HW[1:0] 位，指定从 $\overline{\text{RD}}$ 和 $\overline{\text{WEn}}$ 取反到 $\overline{\text{CSn}}$ 取反的周期数。因此，能获得外部器件的灵活性接口。在如图 9.10 所示的例子中，将 Th 周期和 Tf 周期分别附加在通常周期的前后。在这些周期中，除了 $\overline{\text{RD}}$ 和 $\overline{\text{WEn}}$ 不是断言以外，其他信号都断言。另外，因为数据被延长输出到 Tf 周期为止，所以对写操作较慢的器件等有效。

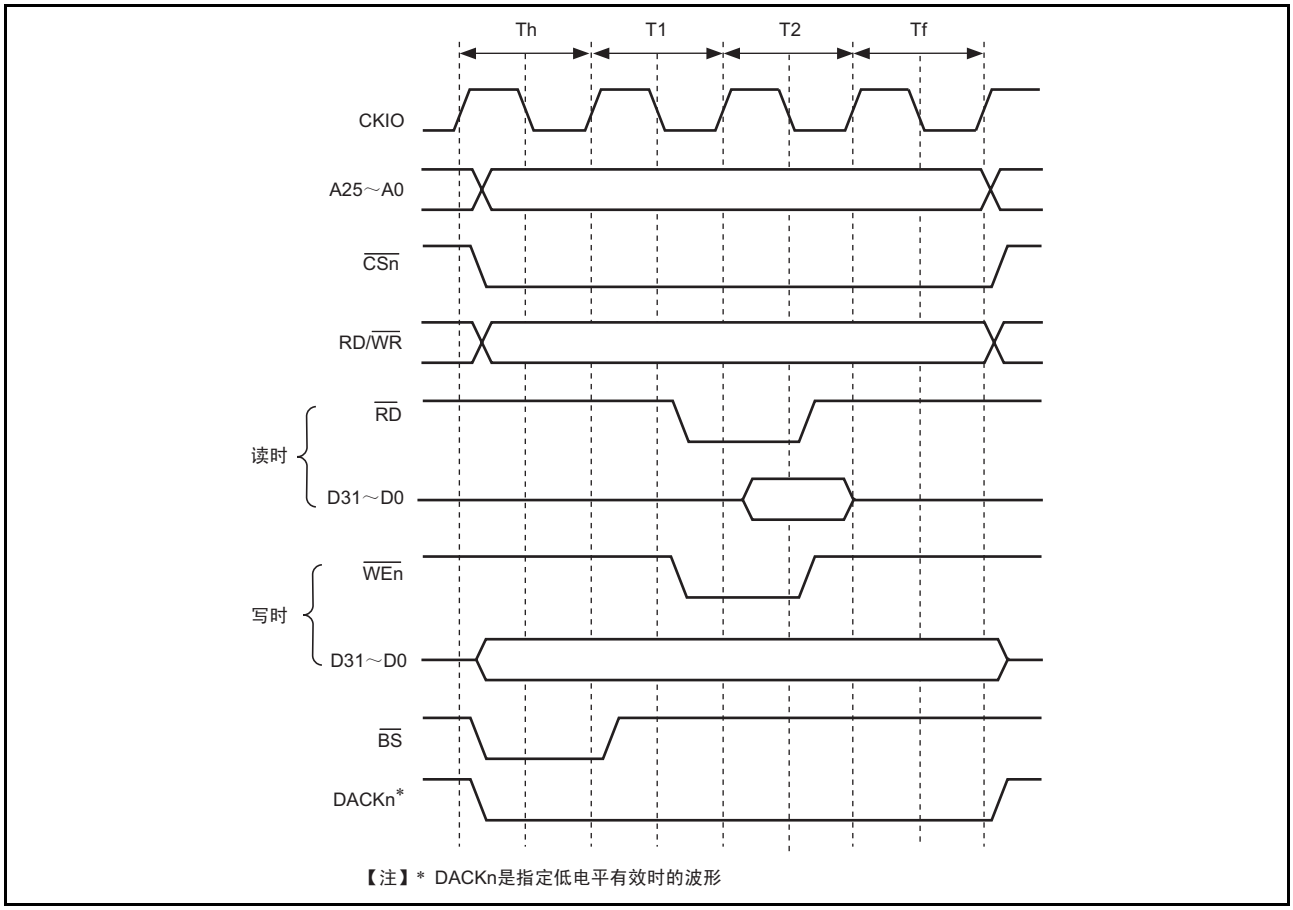


图 9.10　$\overline{\text{CSn}}$ 断言期间的扩展

9.5.5　MPX-I/O 接口

MPX 空间的存取时序如下所示。在 MPX 空间中，通过 $\overline{CS5}$、$\overline{AH}$、$\overline{RD}$ 和 $\overline{WEn}$ 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 16 位可变宽度。

在 Ta2 周期到 Ta3 周期之间从 D15 ～ D0 或者 D7 ～ 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 的 MPXW 位设定为 1，地址输出变为 3 个周期。

RD/$\overline{WR}$ 信号和 $\overline{CS5}$ 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 9.12**、**图 9.13** 所示。

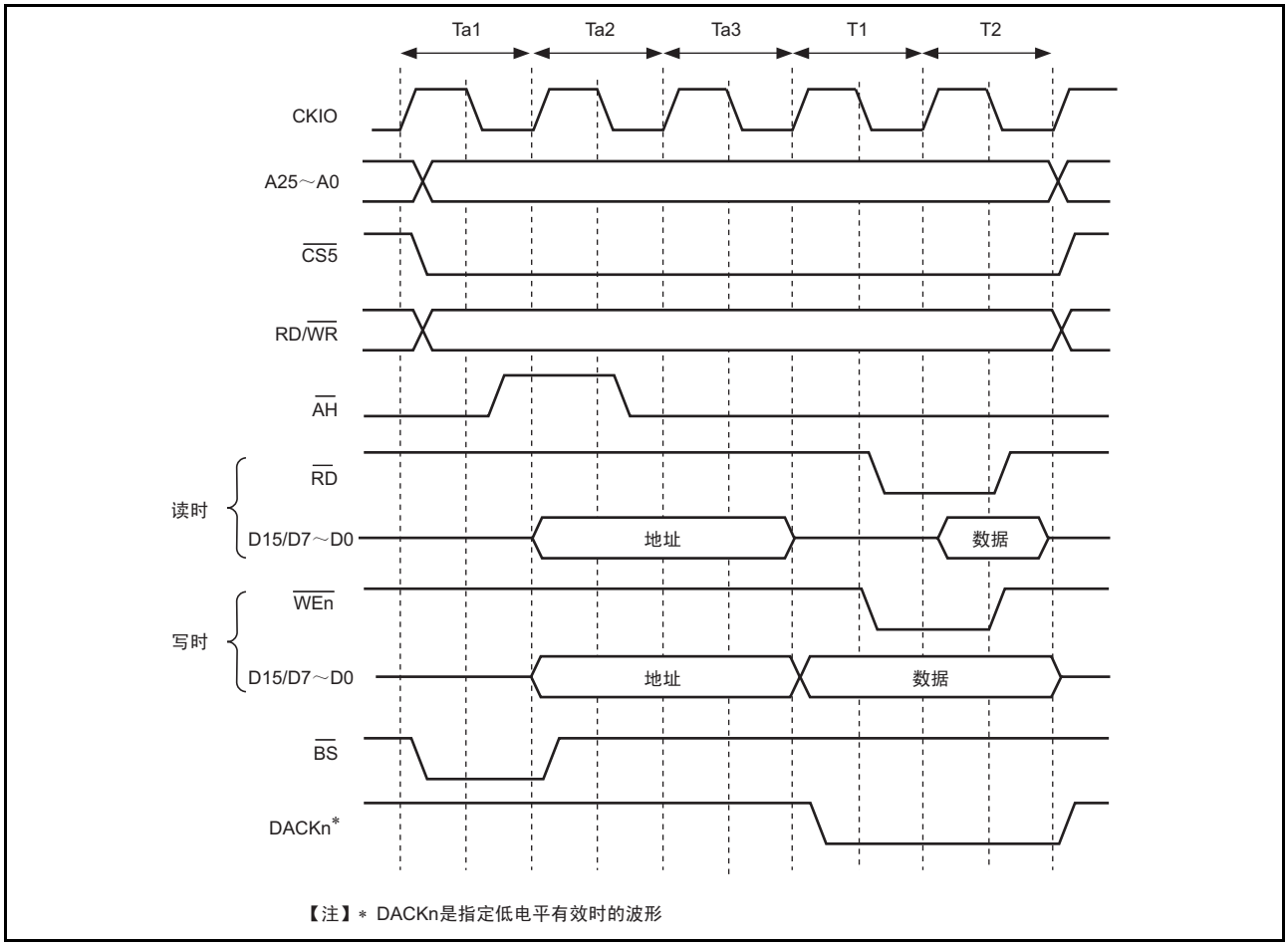


图 9.11　MPX 空间的存取时序 （地址周期无等待、数据周期无等待）

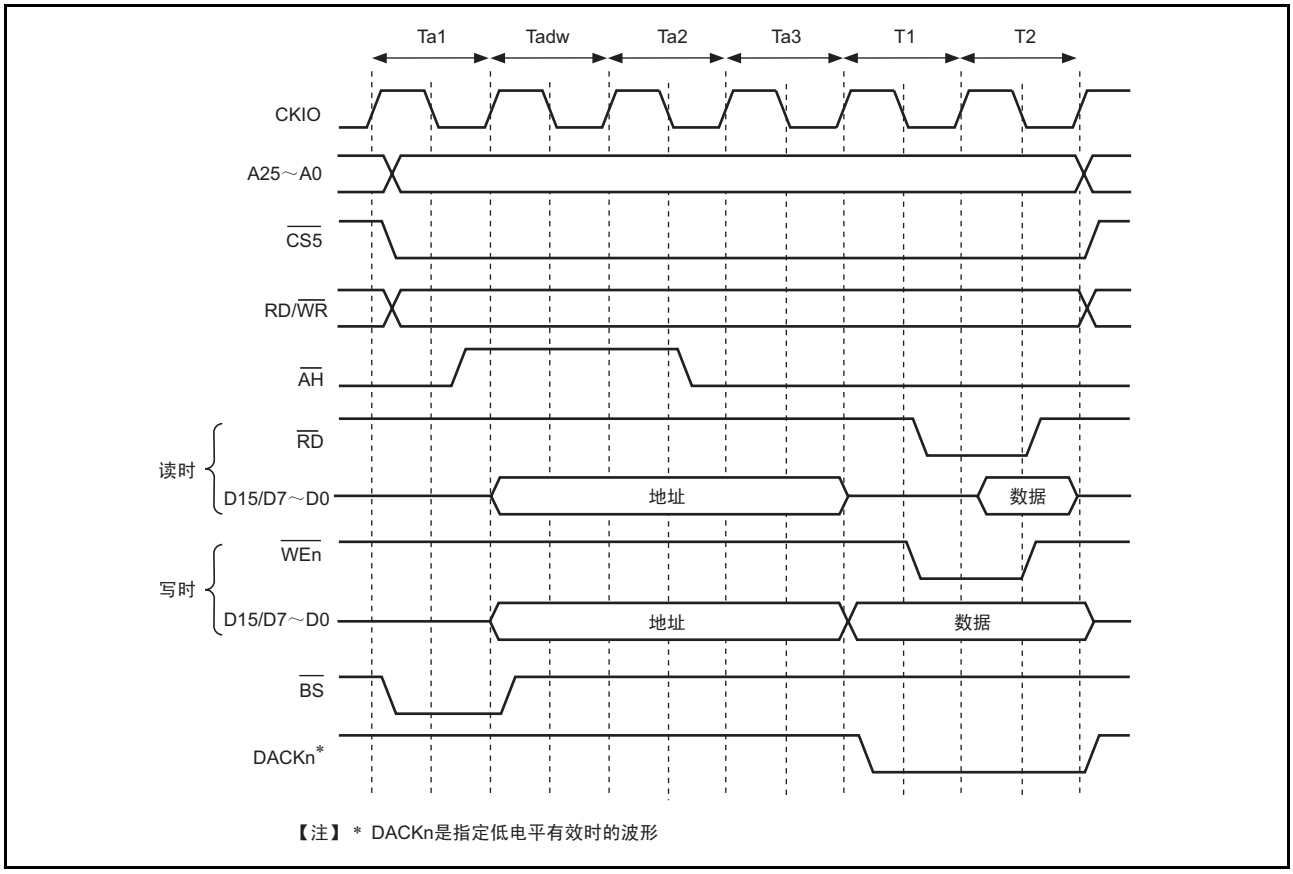


图 9.12　MPX 空间的存取时序 （地址周期等待 1、数据周期无等待）

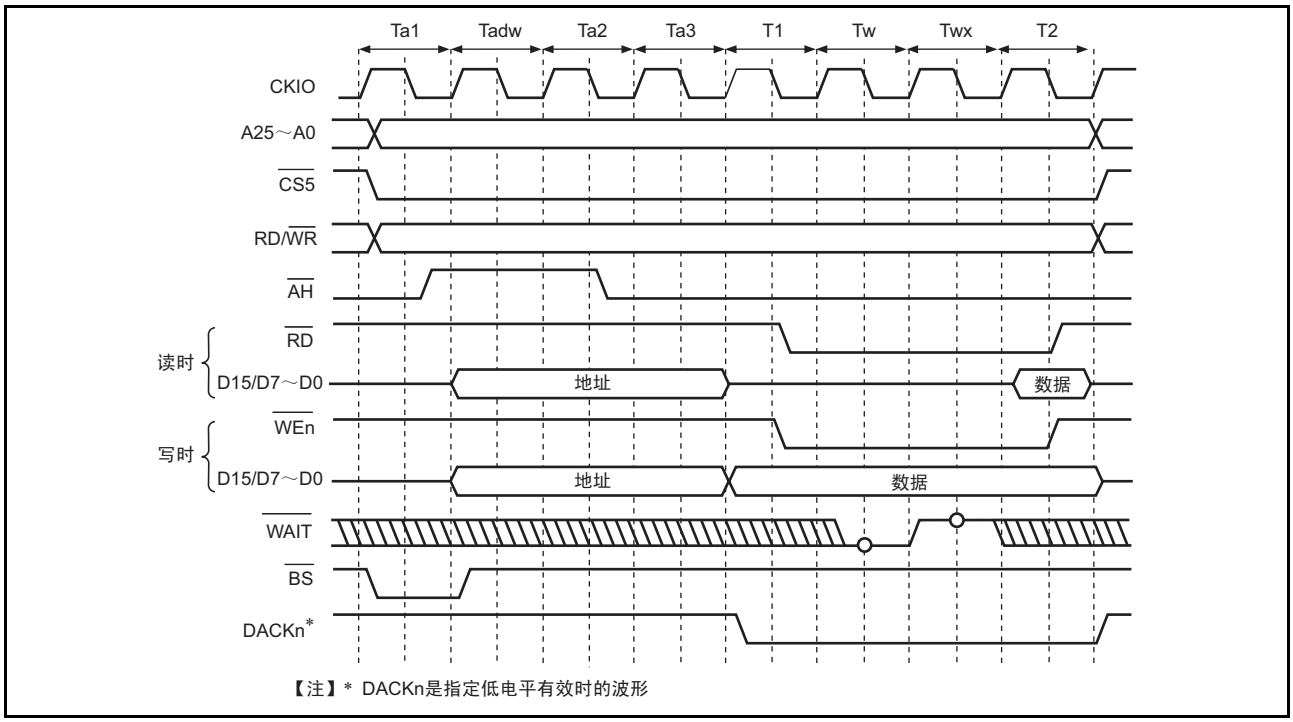


图 9.13　MPX 空间的存取时序
（地址周期的存取等待 1、数据周期等待 1、外部等待 1）

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。

用于直接连接 SDRAM 的控制信号有 $\overline{\text{RASU}}$、$\overline{\text{RASL}}$、$\overline{\text{CASU}}$、$\overline{\text{CASL}}$、RD/$\overline{\text{WR}}$、DQMUU、DQMUL、DQMLU、DQMLL、CKE、$\overline{\text{CS2}}$ 和 $\overline{\text{CS3}}$。除 $\overline{\text{CS2}}$ 和 $\overline{\text{CS3}}$ 以外的信号对各空间是通用的，并且只在 $\overline{\text{CS2}}$ 和 $\overline{\text{CS3}}$ 断言时 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 $\overline{\text{RASU}}$、$\overline{\text{RASL}}$、$\overline{\text{CASU}}$、$\overline{\text{CASL}}$、RD/$\overline{\text{WR}}$ 和特定的地址信号指定 SDRAM 的命令。命令支持：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）和模式寄存器的写 （MRS、 EMRS）等。

通过 DQMUU、 DQMUL、 DQMLU 和 DQMLL 指定要存取的字节。对应的 DQMxx 对低电平的字节进行读 / 写。 DQMxx 和要存取字节的关系请参照 “9.5.1 端法 / 存取长度和数据对齐”。

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 $\overline{\text{RASU}}$、$\overline{\text{RASL}}$、$\overline{\text{CASU}}$ 以及 $\overline{\text{CASL}}$，能使本 LSI 连接 2 个最大 32M 字节的 SDRAM。此时，在同一 CS 空间内共有 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。存取 A25=0 的地址存取时， $\overline{\text{RASL}}$ 和 $\overline{\text{CASL}}$ 断言；存取 A25=1 的地址存取时， $\overline{\text{RASU}}$ 和 $\overline{\text{CASU}}$ 断言。

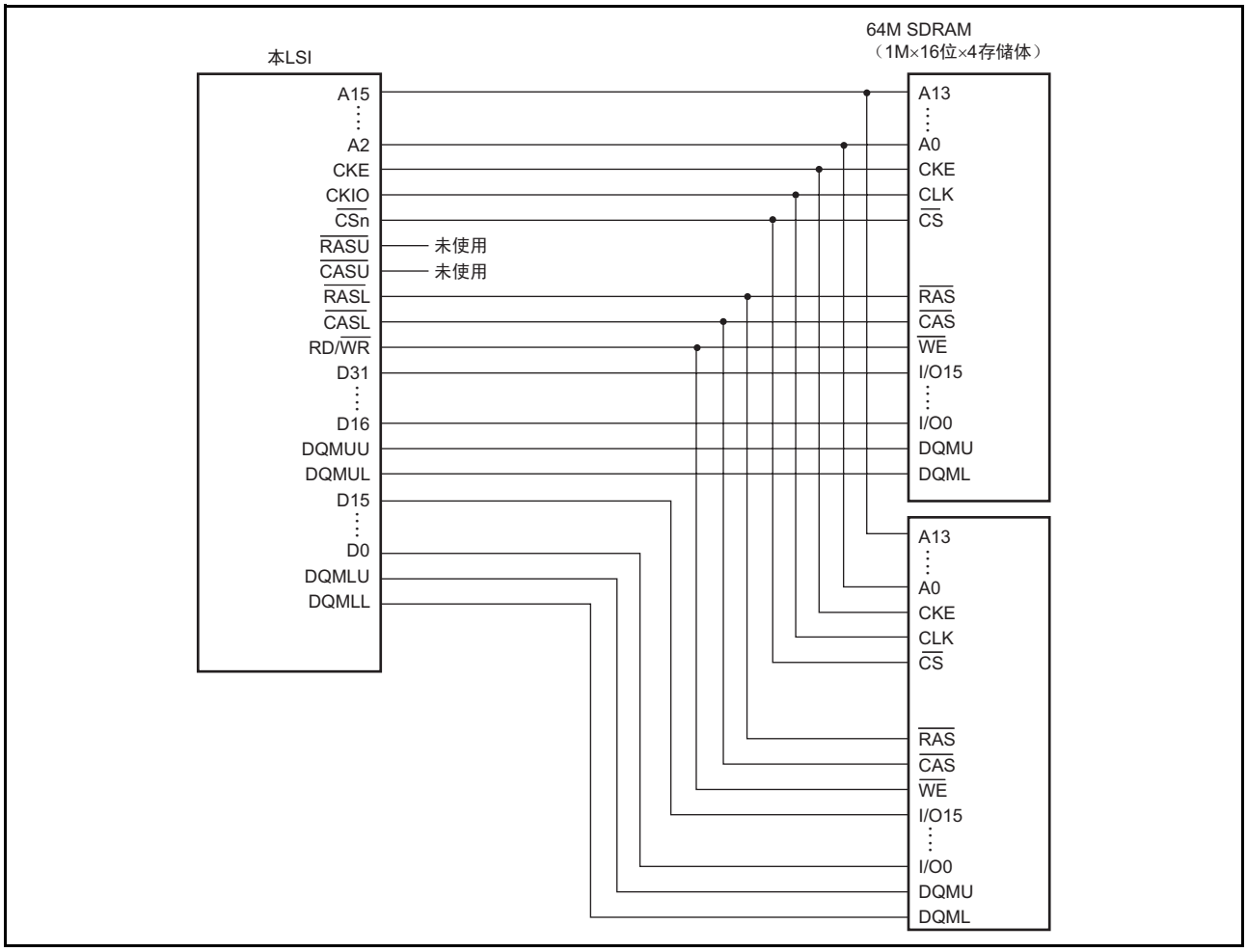


图 9.14　32 位数据宽度的 SDRAM 连接例子 （未使用 $\overline{\text{RASU}}$、 $\overline{\text{CASU}}$）

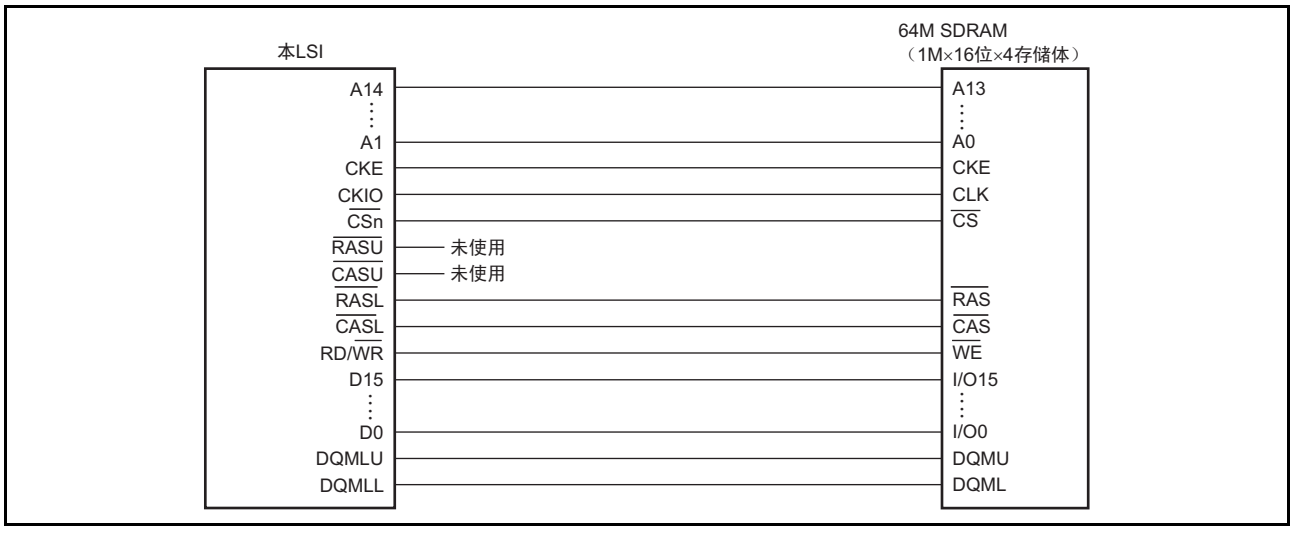


图 9.15　16 位数据宽度的 SDRAM 连接例子 （未使用 $\overline{\text{RASU}}$、 $\overline{\text{CASU}}$）

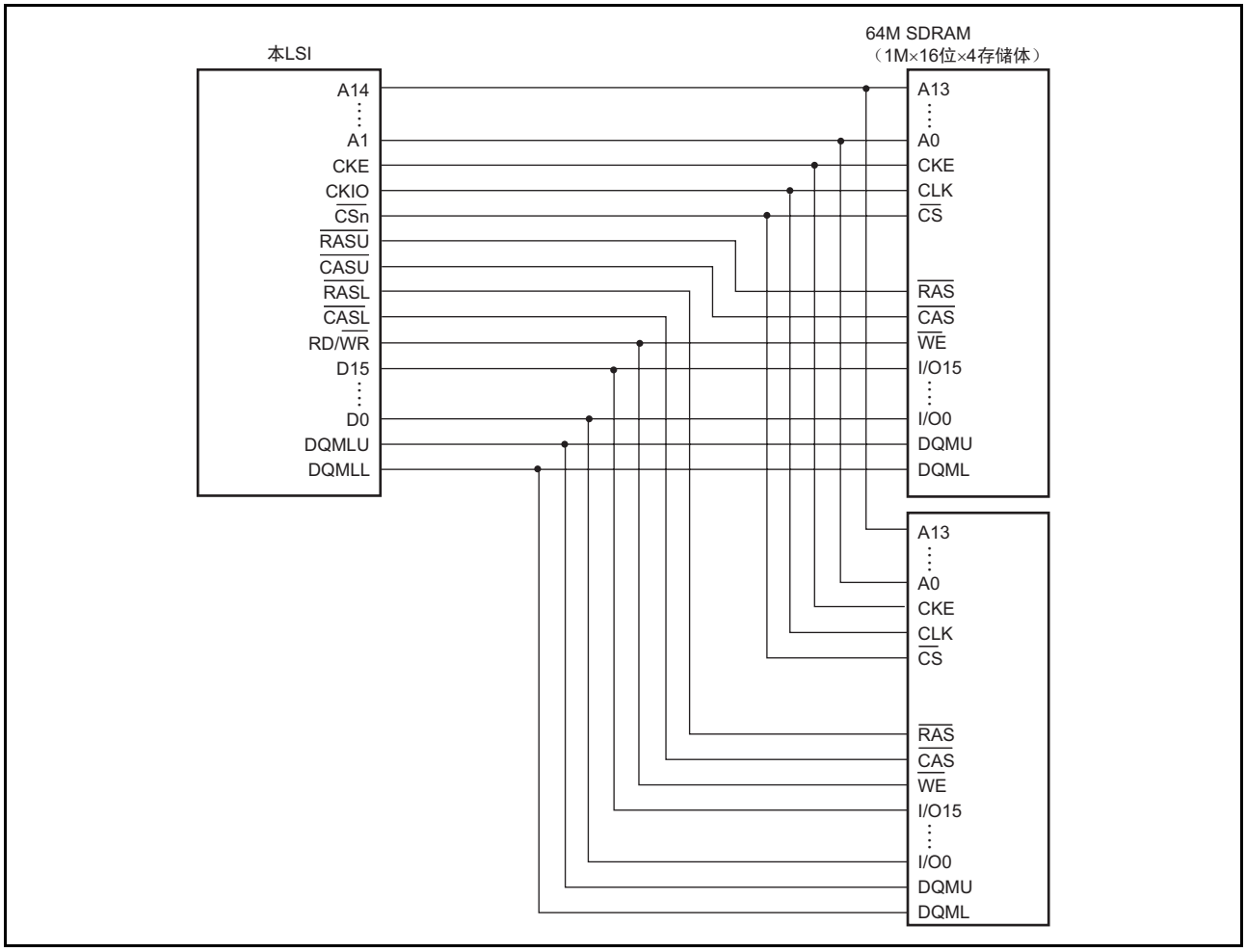


图 9.16　16 位数据宽度的 SDRAM 连接例子 （使用 $\overline{\text{RASU}}$、 $\overline{\text{CASU}}$）

（2） 地址的多路复用

根据 CSnBCR 的 BSZ[1:0] 位、SDCR 的 A2ROW[1:0] 位、A2COL[1:0] 位、A3ROW[1:0] 位和 A3COL[1:0] 位的设定进行地址的多路复用，以便不需要外接地址多路复用的电路就能连接SDRAM。BSZ[1:0]、A2ROW[1:0] 位、A2COL[1:0] 位、A3ROW[1:0] 位和 A3COL[1:0] 位的设定和输出到地址引脚的位的关系如**表 9.11 ～表 9.16** 所示。不能进行此表以外的设定，否则就不保证运行。 A25 ～ A18 不进行多路复用，总是输出原地址。

当数据总线宽度为 16 位时 （BSZ[1:0]=B'10）， SDRAM 的 A0 引脚指定字地址。因此请将 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 位时 （BSZ[1:0]=B'11）， SDRAM 的 A0 引脚指定长字地址。因此请将 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 9.11　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	SDRAM 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
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	SDRAM 的引脚	功能
A17	A25	A17		未使用
A16	A24	A16		
A15	A23*2	A23*2	A13(BA1)	指定存储体
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
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 是用于指定指令的位，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 9.12　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	SDRAM 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	SDRAM 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 $\overline{\text{RASL}}$ 断言，而 $\overline{\text{RASU}}$ 不断言。

表 9.13　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）		
本 LSI 的输出引脚	行地址的输出周期	列地址的输出周期	SDRAM 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 个存储体、列 9 位产品） 1 个 256M 位产品 （4M 字 ×16 位 ×4 个存储体、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 $\overline{\text{RASL}}$ 断言，而 $\overline{\text{RASU}}$ 不断言。

表 9.14　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	SDRAM 的引脚	功能
A17	A25	A17		未使用
A16	A24	A16		
A15	A23	A15		
A14	A22	A14		
A13	A21	A21		
A12	A20*2	A20*2	A11(BA0)	指定存储体
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		未使用
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	SDRAM 的引脚	功能
A17	A25	A17		未使用
A16	A24	A16		
A15	A23	A15		
A14	A22*2	A22*2	A13(BA1)	指定存储体
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		未使用
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 9.15　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	SDRAM 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	SDRAM 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 9.16　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	SDRAM 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	SDRAM 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 $\overline{\text{RASL}}$ 断言，而 $\overline{\text{RASU}}$ 不断言。

（3） 突发读

本 LSI 产生突发读的项目如下：

1. 当读取的存取长度大于数据总线宽度时
2. 高速缓存未中的情况下进行16字节传送时
3. 当通过DMAC进行16字节传送时
4. 进行LCDC的16字节～128字节传送时*

本 LSI 总是以突发长度 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 的突发读。存取长度和突发数的关系如表 9.17 所示。

【注】 * 详细情况请参照 “第 26 章 LCD 控制器 （LCDC）”。

表 9.17　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 字节 *	16
	64 字节 *	32
	128 字节 *	64
32 位	8 位	1
	16 位	1
	32 位	1
	16 字节	4
	32 字节 *	8
	64 字节 *	16
	128 字节 *	32

【注】 * 使用 LCDC 时发生 32、64、128 字节存取。
详细情况请参照 “第26章 LCD控制器 （LCDC）”。

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 Tr 周期输出 ACTV 命令，接着在 Tc1、Tc2、 Tc3 周期发送 READ 命令、在 Tc4 周期发送 READA 命令，并且在从 Td1 到 Td4 的周期中在外部时钟（CKIO）的上升沿接收读数据。 Tap 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发送新命令。但是可存取不同的 CS 空间或者同一 SDRAM 的不同存储体。通过指定 CS3WCR 的 WTRP[1:0] 位决定 Tap 的周期数。

本 LSI 为了用各种频率连接 SDRAM，能通过设定 CS3WCR 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 WTRCD[1:0] 指定从 ACTV 命令输出周期 Tr 到 READ 命令输出周期 Tc1 之间的周期数。将 WTRCD[1:0] 设定为至少 1 个周期时，就在 Tr 周期和 Tc1 周期之间插入 NOP 命令发送的 Trw 周期。能通过 CS2WCR 的 A2CL[1:0] 位和 CS3WCR 的 A3CL[1:0] 位，在 CS2 和 CS3 空间内分别独立指定从 READ 命令输出周期 Tc1 到读数据输入周期 Td1 之间的周期数。此周期数相当于 SDRAM 的 CAS 等待时间。SDRAM 的 CAS 等待时间最多为 3 个周期，但是本 LSI 能设定 1~4 个周期。这是为了在本 LSI 和 SDRAM 之间连接含锁存器的电路。

Tde 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 Tde 周期。

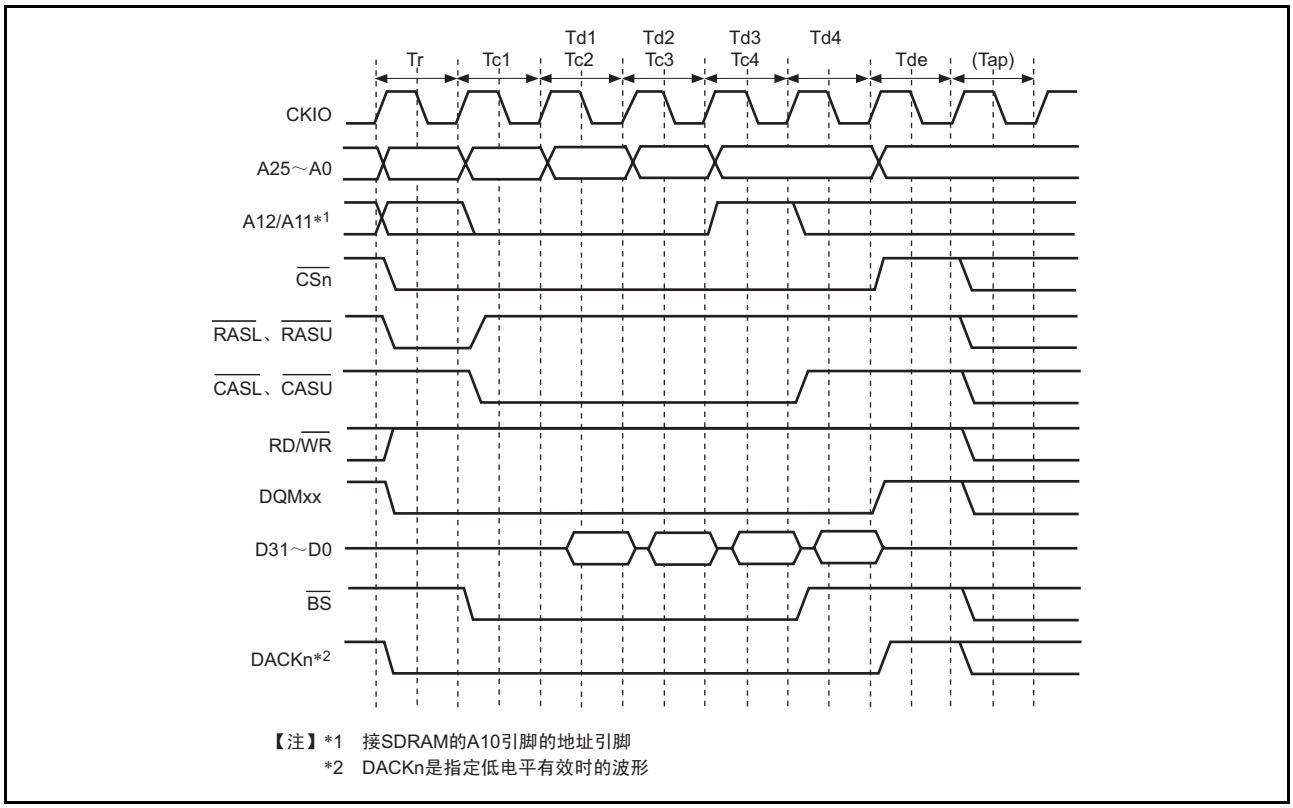
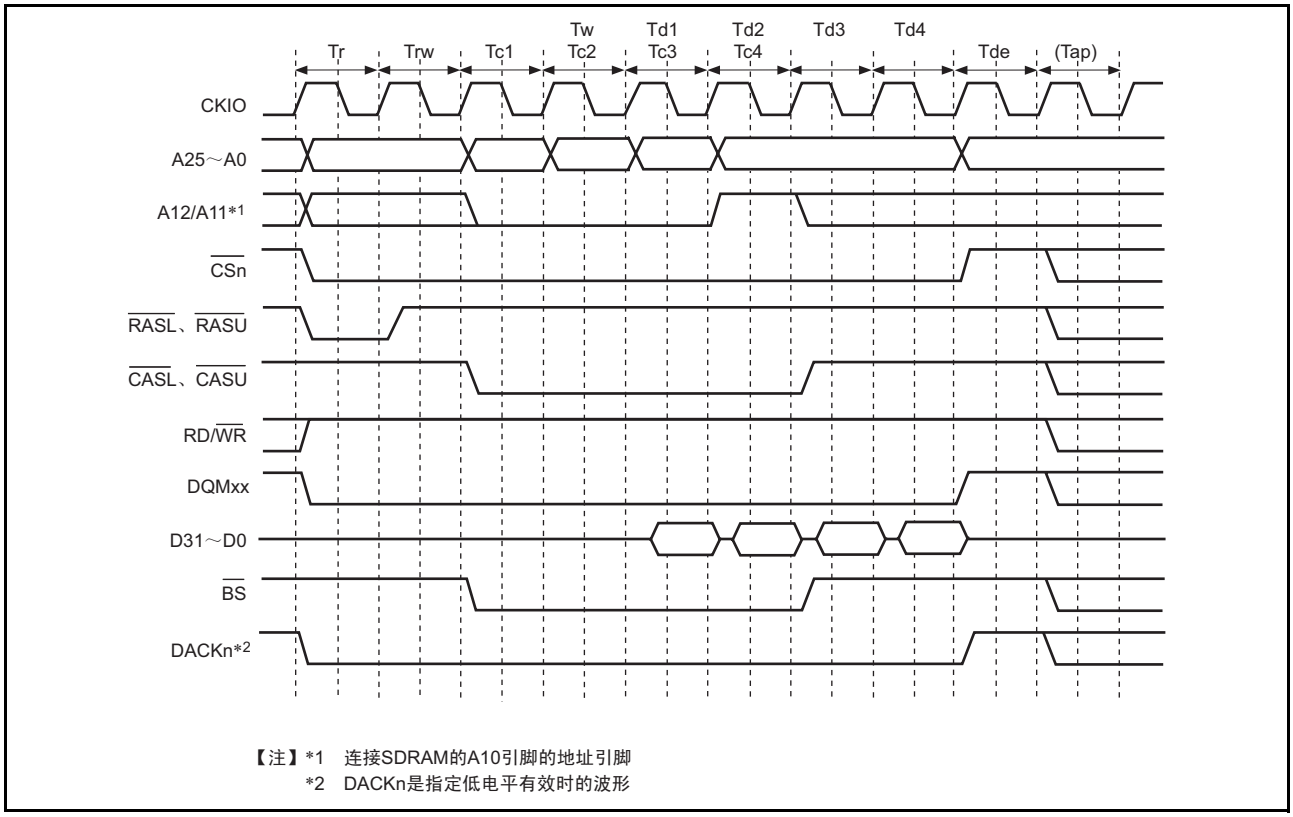


图 9.17　突发读的基本时序 （CAS 等待时间 1、自动预充电）



【注】*1　连接SDRAM的A10引脚的地址引脚
*2　DACKn是指定低电平有效时的波形

图 9.18　突发读的等待指定时序
（CAS 等待时间 2、 WTRCD[1:0]=1 周期、自动预充电）

（4） 单次读

当在高速缓存无效空间中并且数据总线宽度不小于存取长度时，读存取进行 1 次就结束。因为 SDRAM 设定为突发长度 1 的突发读，所以只输出需要的数据。进行一次就结束的读存取称为单次读。

单次读的基本时序图如**图 9.19** 所示。

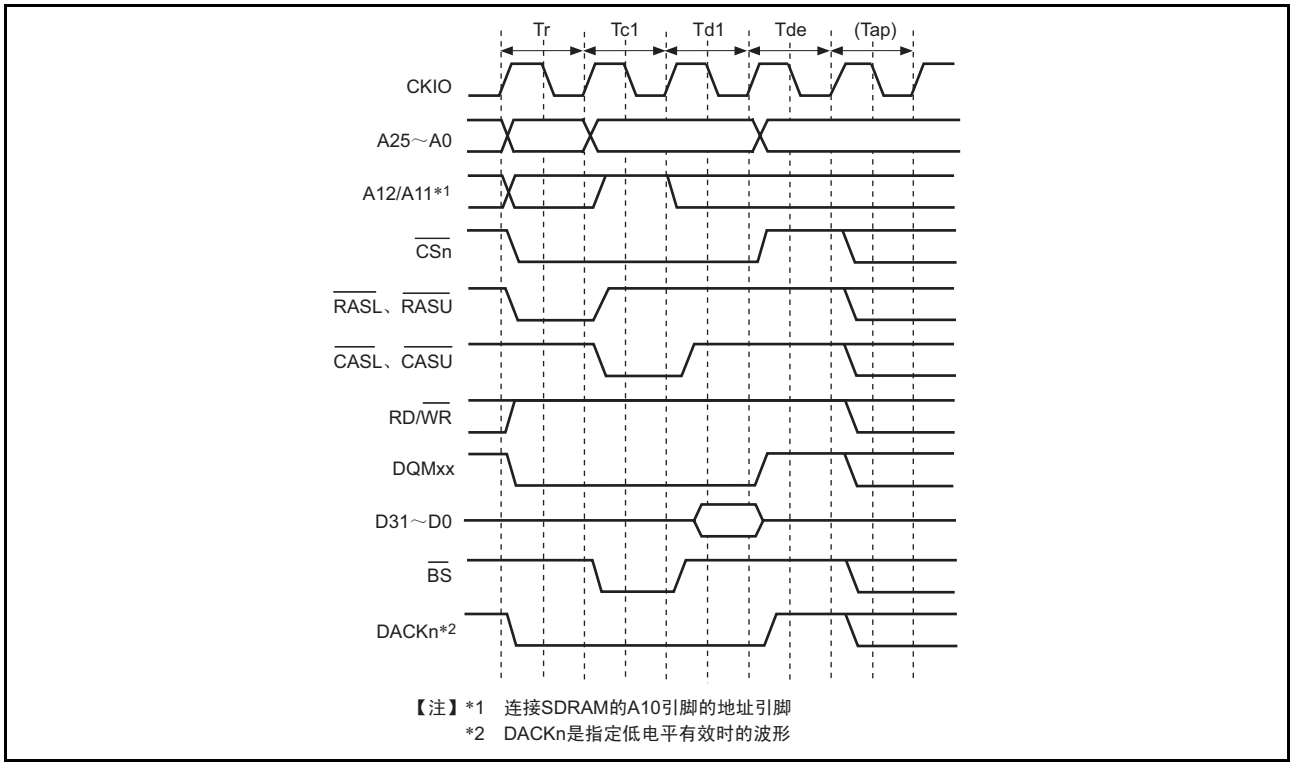


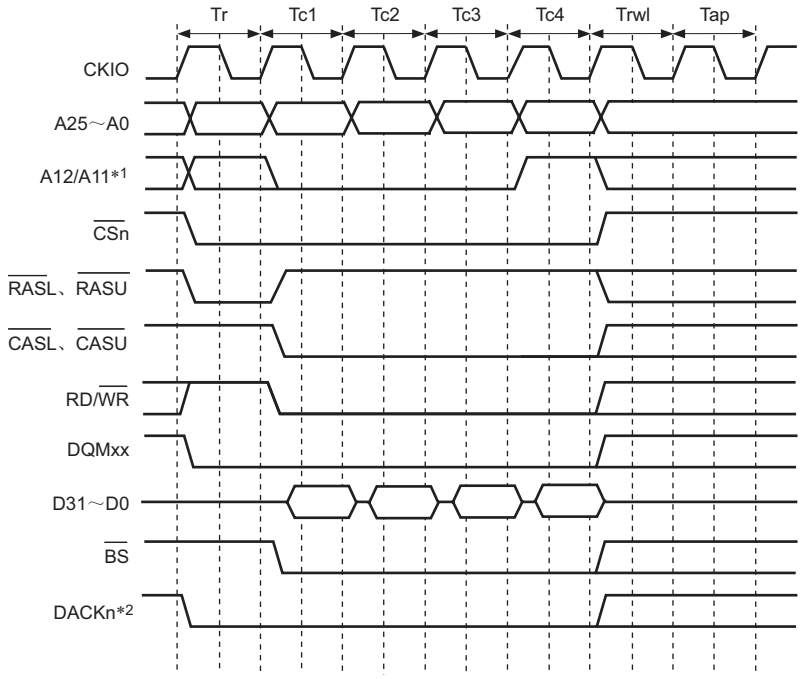
图 9.19　单次读的基本时序 （CAS 等待时间 1、自动预充电）

（5） 突发写

本 LSI 产生突发写的项目如下：

1. 当写入的存取长度大于数据总线宽度时
2. 产生高速缓存的回写时
3. 通过DMAC进行16字节传送时

本 LSI 总是以突发长度 1 存取 SDRAM。例如，当从连接 32 位数据总线的 SDRAM 连续写入 16 字节的数据时，连续进行 4 次突发长度为 1 的写操作。此时的存取称为突发数 4 的突发写。存取长度和突发数的关系如**表 9.17** 所示。突发写的时序图如**图 9.20** 所示。在突发写过程中，在 Tr 周期输出 ACTV 命令，在 Tc1、 Tc2、Tc3 周期发送 WRIT 命令，在 Tc4 周期发送进行自动预充电的 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 Trw1 周期 （等待自动预充电的启动）和 Tap 周期 （等待自动预充电的结束）。 Tap 周期是等待 SDRAM 内部自动预充电结束 （由 WRITA 命令引起的）的周期。在 Trwl 周期和 Tap 周期之间不对同一存储体发送新命令。但是可存取不同的 CS 空间或同一 SDRAM 的不同存储体。通过指定 CS3WCR 的 TRWL[1:0] 位决定 Trw1 周期，通过指定 CS3WCR 的 WTRP[1:0] 位决定 Tap 周期。



【注】*1　连接SDRAM的A10引脚的地址引脚
*2　DACKn是指定低电平有效时的波形

图 9.20　突发写的基本时序 （自动预充电）

（6） 单次写

当在高速缓存无效空间中且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。因为 SDRAM 设定为单次写或突发长度为 1 的突发写，所以只写入需要的数据。进行 1 次就结束的写存取称为单次写。单次写的基本时序图如图 9.21 所示。

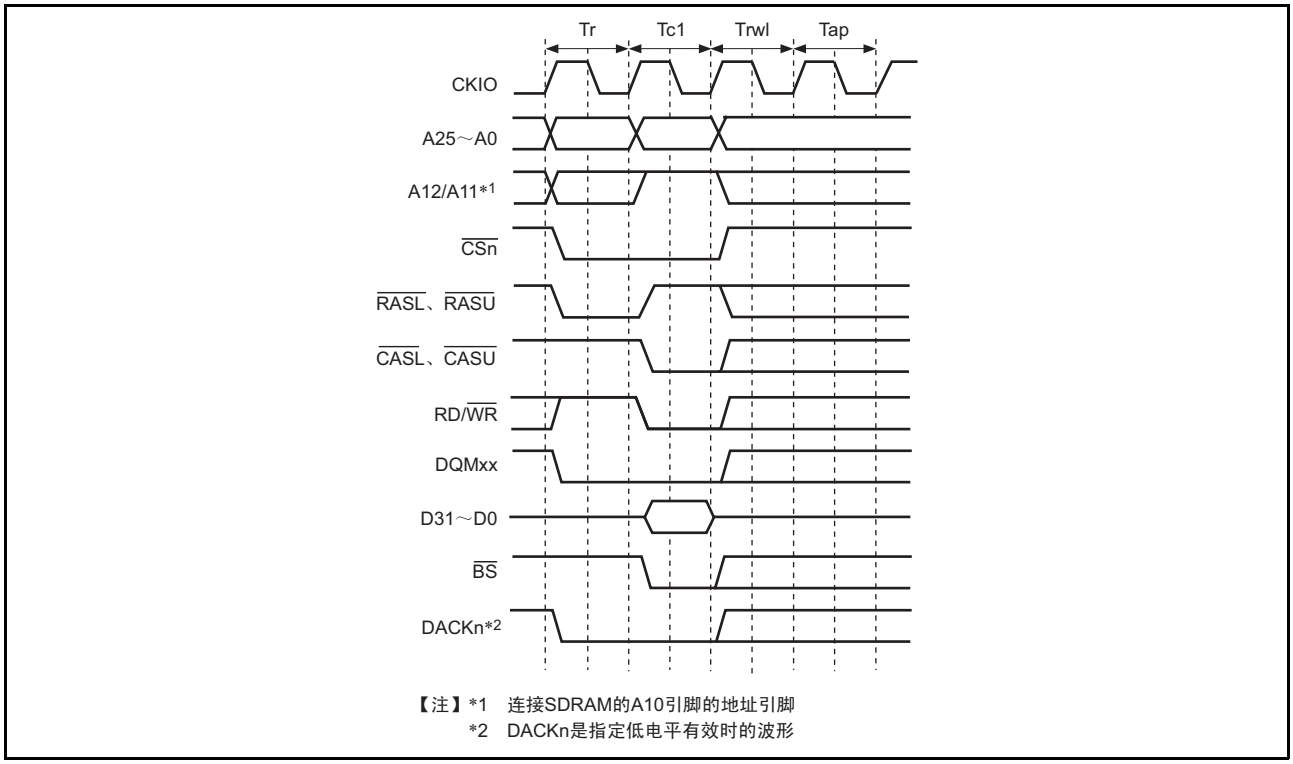


图 9.21　单次写的基本时序 （自动预充电）

（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 位为 1 时，使用无自动预充电的命令 （READ 或 WRIT）进行存取。这就是存储体激活功能。但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，请将区域 2 空间设定为通常空间或带字节选择的 SRAM。当区域 2 和区域 3 都设定为 SDRAM 时，请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。当存取同一存储体的相同行地址时，能不发送 ACTV 命令而直接发送 READ 命令或 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。当下一次存取是对不同的行地址进行的存取时，首先发送 PRE 命令然后进行相应存储体的预充电，在预充电结束后按顺序发送 ACTV 命令、READ 命令或 WRIT 命令。当继续存取不同的行地址时，有可能因为在产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] 位指定从发送 PRE 命令到发送 ACTV 命令的周期数。

如果在写操作时进行自动预充电，就不能在发送 WRITA 命令后的 Trwl+Tap 周期内对同一存储体发送命令。如果使用存储体激活模式时，为同一行地址时就能连续发送 READ 命令或 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。在执行程序的过程中，如果不能保证在限制所规定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。

无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。另外，无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，在发送 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 读取数据时，使指定读取字节的 DQMxx 信号保持 2 个周期的等待时间。当 CAS 等待时间不小于 2 个周期时，即使在 Tc 周期以后断言 DQMxx 信号，也可保证 2 个周期的等待时间，所以不插入 Tnop 周期。

只从存取空间 （被设定存储体激活功能）的各存储体来看，只要连续存取同一行地址，就从图 9.22 或图 9.25 所示的周期开始运行，重复图 9.23 或图 9.26 所示的周期。此时即使存取不同空间或不同存储体也无影响。如果在存储体激活模式中存取不同的行地址，执行图 9.24 或图 9.27 的总线周期，而不是图 9.23 或图 9.26 的总线周期。即使在存储体激活模式中，刷新周期后或通过总线仲裁释放总线后，所有存储体仍为非激活状态。

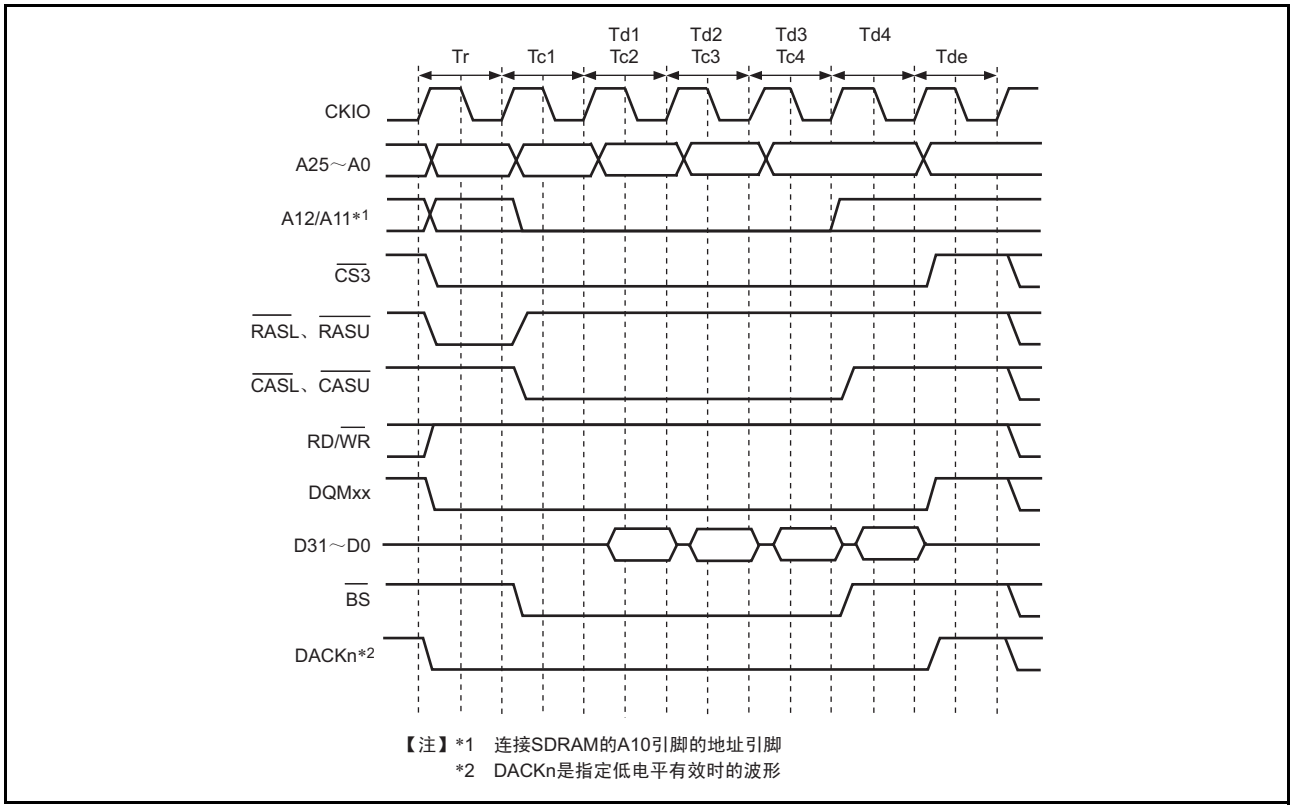


图 9.22　突发读的时序 （存储体激活、不同存储体、 CAS 等待时间 1）

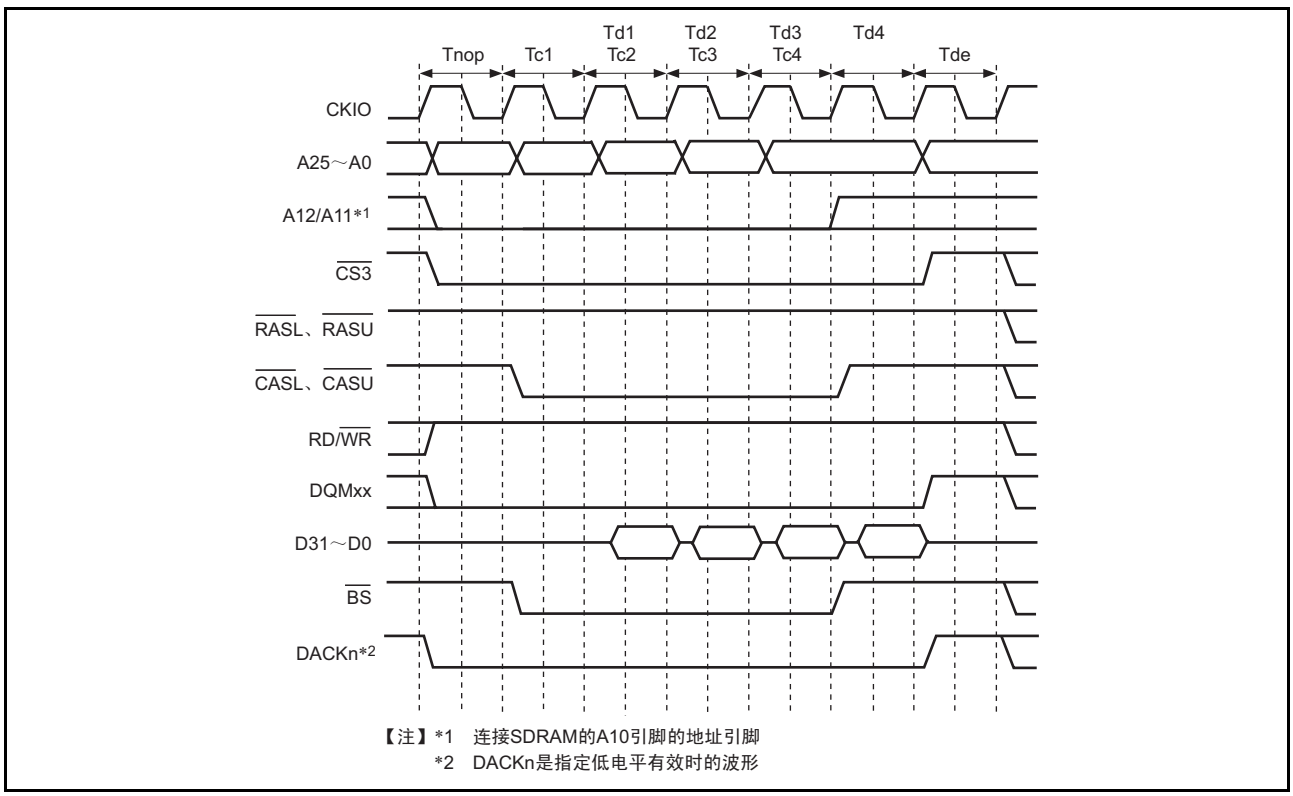


图 9.23　突发读的时序 （存储体激活、同一存储体中的同一行地址、 CAS 等待时间 1）

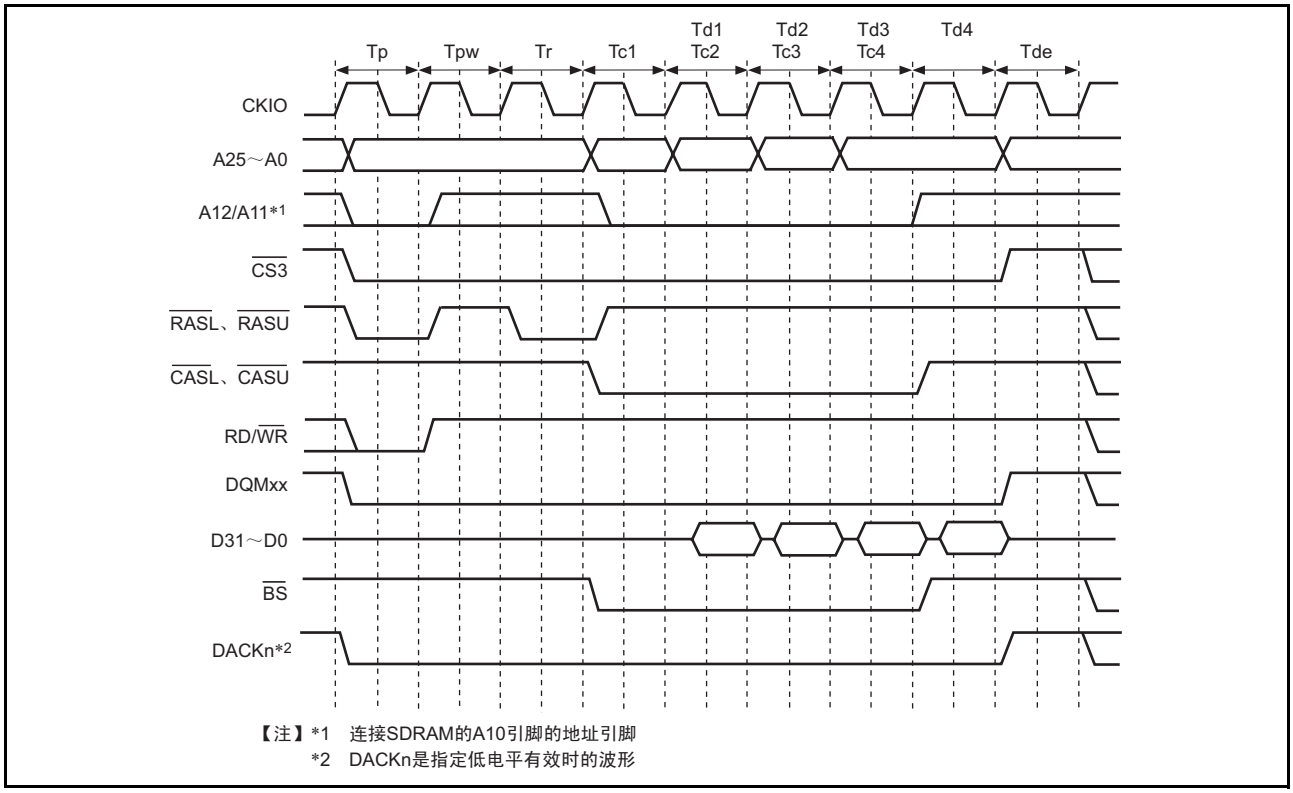


图 9.24　突发读的时序
（存储体激活、同一存储体中的不同行地址、 CAS 等待时间 1）

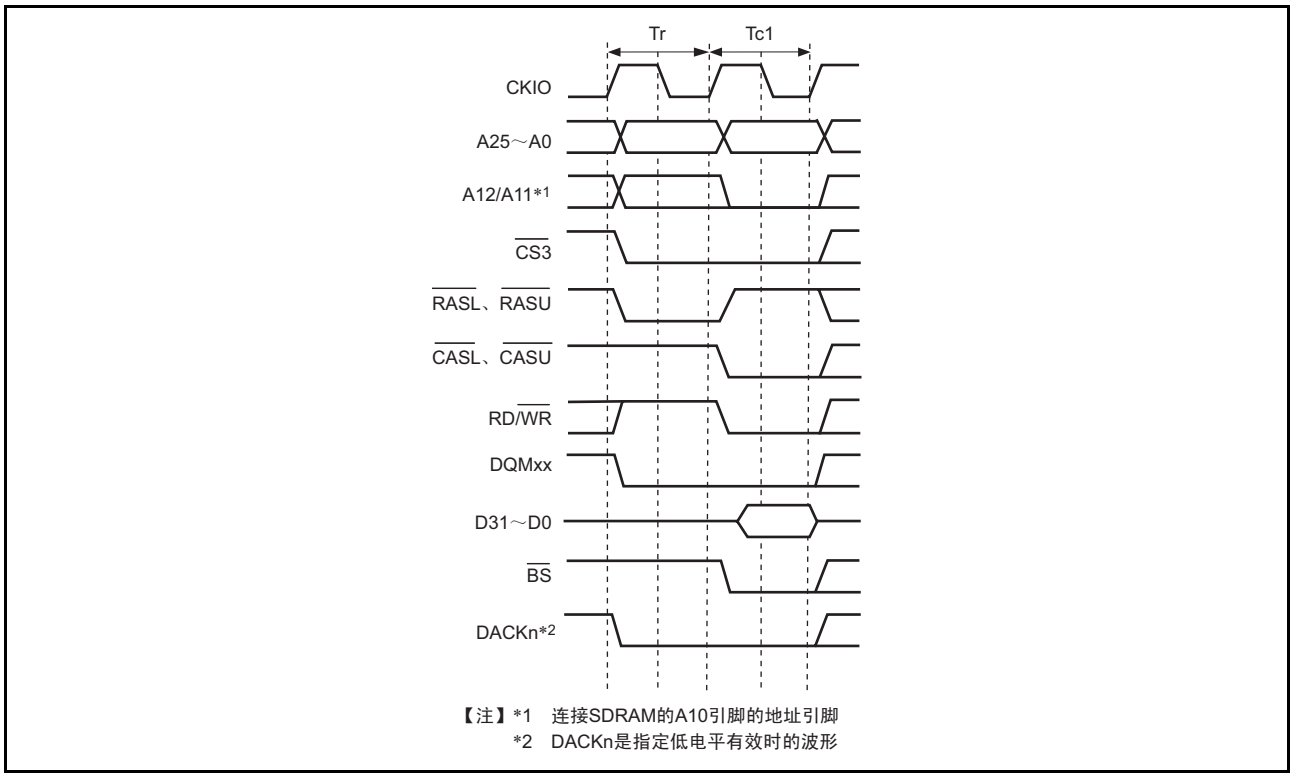
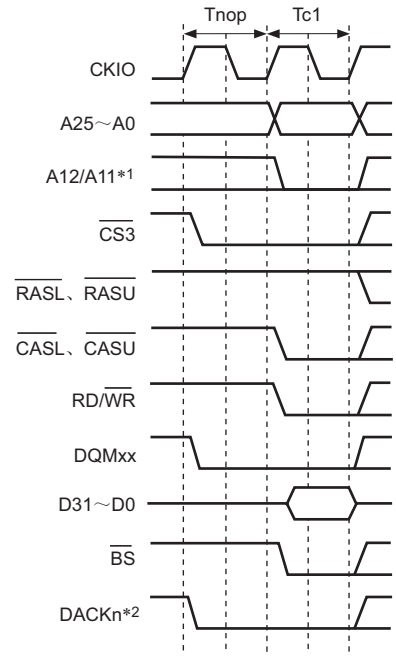
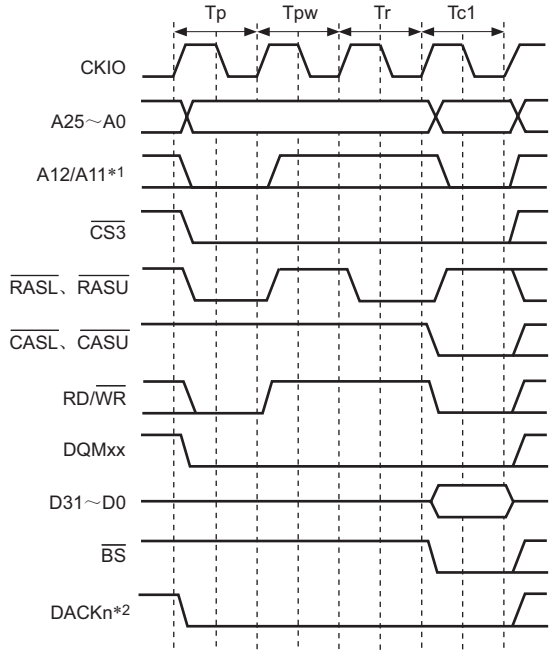


图 9.25　单次写的时序 （存储体激活、不同存储体）



【注】*1　连接SDRAM的A10引脚的地址引脚
*2　DACKn是指定低电平有效时的波形

图 9.26　单次写的时序 （存储体激活、同一存储体中的同一行地址）



【注】*1　连接SDRAM的A10引脚的地址引脚
*2　DACKn是指定低电平有效时的波形

图 9.27　单次写的时序 （存储体激活、同一存储体中的不同行地址）

（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 位同时置 1，启动低功耗的自刷新。

（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] 位设定次数的刷新。为了符合所使用的 SDRAM 的刷新间隔规定，请设定各寄存器。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] 位选择输入时钟， 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，如果两者的值相同，就产生刷新请求并执行 RRC[2:0] 位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。启动自动刷新时，如果有正在预充电的存储体，在等待自动刷新结束后，在 Tp 周期中发送 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] 位所定的空闲周期数，然后在 Trr 周期中发送 REF 命令。在 Trr 周期之后和 CS3WCR 的 WTRC[1:0] 位指定的周期数之间不发送新命令。为了符合 SDRAM 刷新周期时间的规定（tRC），需设定 WTRC[1:0] 位。当 CS3WCR 的 WTRP[1:0] 位的设定值至少是 1 个周期时，就在 Tp 周期和 Trr 周期之间插入空闲周期。

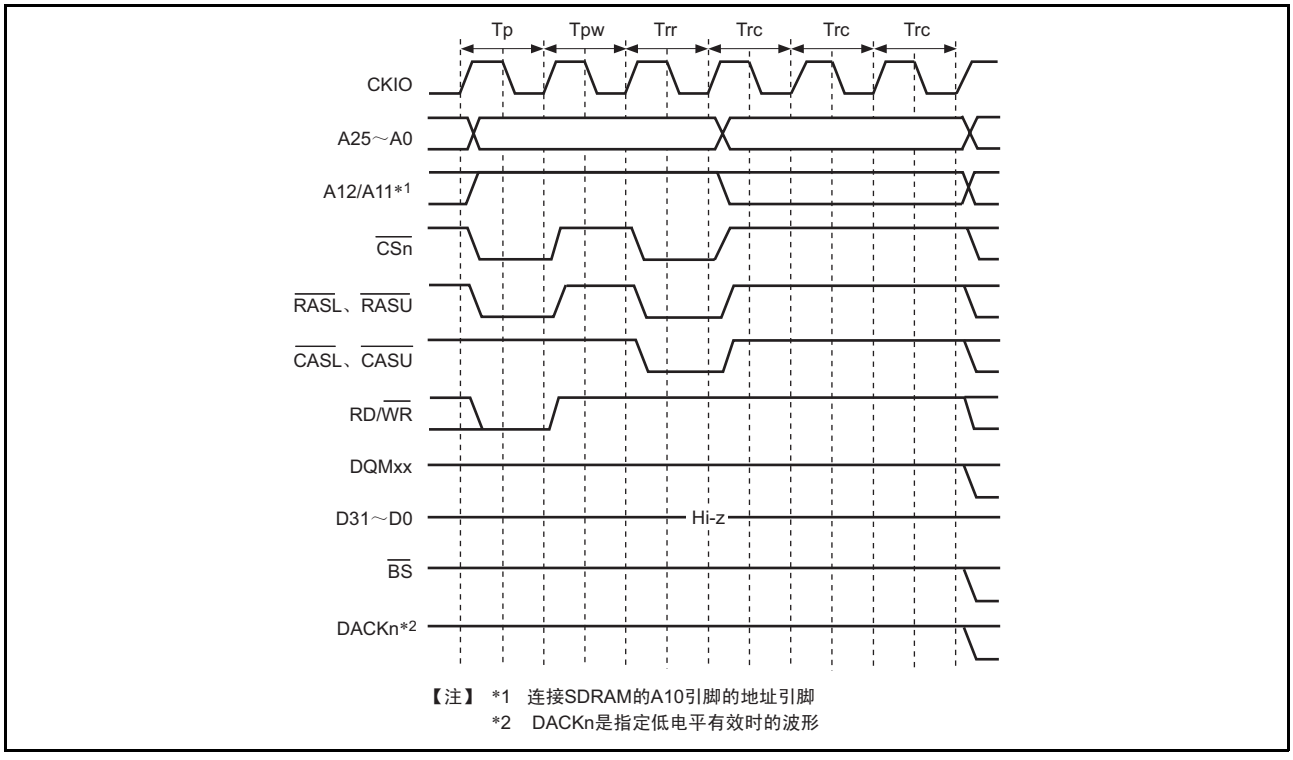


图 9.28　自动刷新的时序

（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 位同时置 1 进行启动。启动自刷新时，如果有正在预充电的存储体，就等待自刷新结束后，在 Tp 周期中发送 PALL 命令。接着，在插入 CS3WCR 的 WTRP[1:0] 位设定的空闲周期后发送 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，自刷新解除后在 CS3WCR 的 WTRC[1:0] 位指定的周期数之间不发送命令。

自刷新的时序如**图 9.29** 所示。在解除自刷新后，请立即进行设定以确保能以正确的间隔执行自动刷新。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。

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。但将 CMNCR 寄存器的 HIZCNT 位置 1，即使在待机状态下，也需要驱动 CKE 引脚。

另外，改变 PLL 电路的倍频率时， CKIO 输出不稳定，变为低电平。关于 CKIO 输出请参照 **“第 4 章　时钟脉冲振荡器** （CPG）”。在倍频率变更前设置为自刷新状态，可保持 SDRAM 的内容。

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。

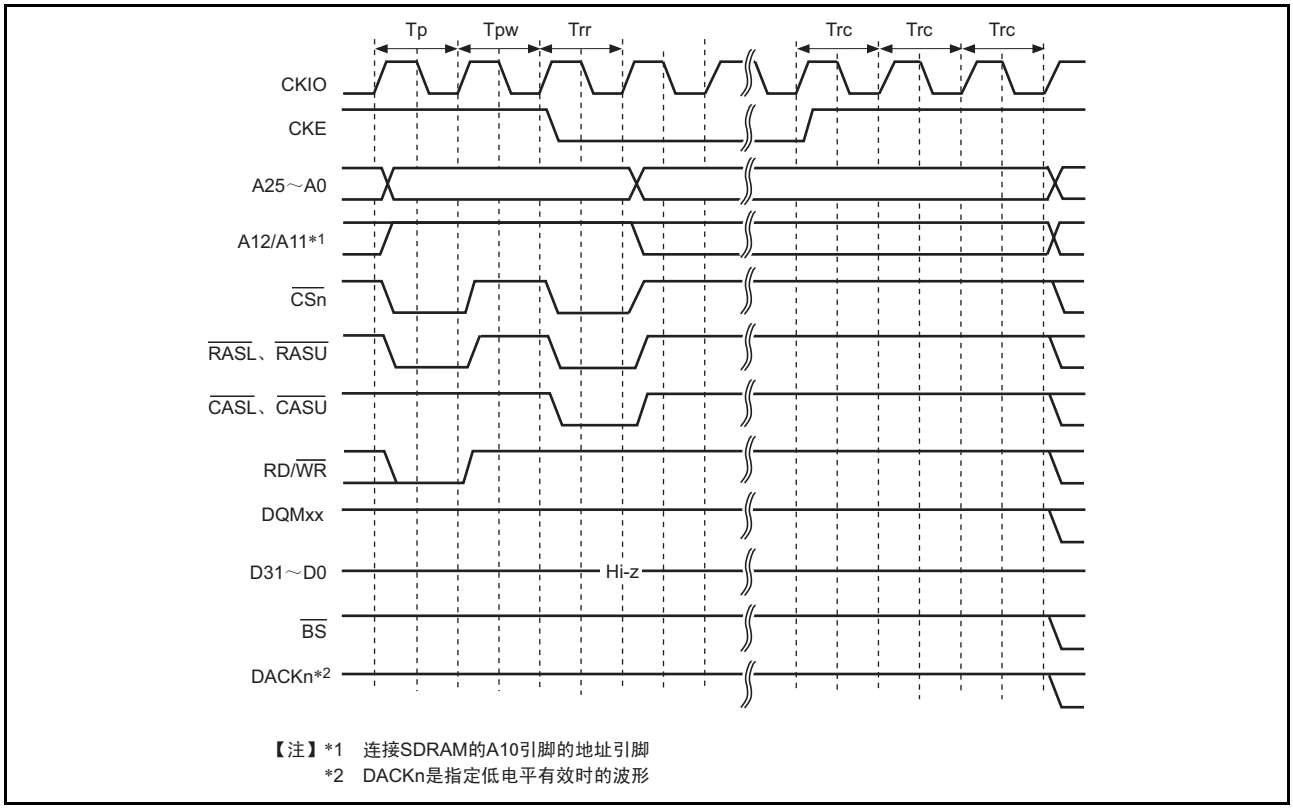


图 9.29　自刷新的时序

（9） 刷新请求和总线周期的关系

在执行总线周期的过程中产生刷新请求时，等到总线周期结束后执行刷新。另外，如果在通过总线仲裁功能释放总线时产生刷新请求，就在等到获得总线权后执行刷新。本 LSI 设有 $\overline{\text{REFOUT}}$ 引脚作为在等待执行刷新期间的总线权请求信号。$\overline{\text{REFOUT}}$ 引脚的选择请参照“**第 29 章　引脚功能控制器**（PFC）”。在获得总线权前，$\overline{\text{REFOUT}}$ 一直断言 （低电平）。

因为 $\overline{\text{REFOUT}}$ 断言，外部器件必须使 $\overline{\text{BREQ}}$ 取反并返回总线权。请注意，当超过刷新间隔的规定时间，外部器件仍没有返回总线权时，不能进行刷新并无法保证 SDRAM 的内容。

在等待执行刷新的状态下产生新的刷新请求时，取消前一个刷新请求。为了正确进行刷新，需要比刷新间隔长的总线周期或不占有总线权。

即使在自刷新时产生总线权请求，也不在解除自刷新前释放总线。

（10）低频率模式

如果将 SDCR 的 SLOW 位置 1，就在适合以低频率运行 SDRAM 的时序输出命令、地址、写数据和输入读数据。

低频率模式中的存取时序如**图 9.30** 所示。该模式在比正常慢半个周期的 CKIO 下降沿同步输出命令、地址、写数据。另外，在比正常快半个周期的 CKIO 的下降沿输入读数据。因此可延长命令、地址、写数据及读数据的保持时间。

如果将 SLOW 位置 1，并以高频率运行 SDRAM 时，就可能无法确保命令、地址、写数据及读数据的设置时间。请针对运行频率或电路板的时序决定 SLOW 位的设定。

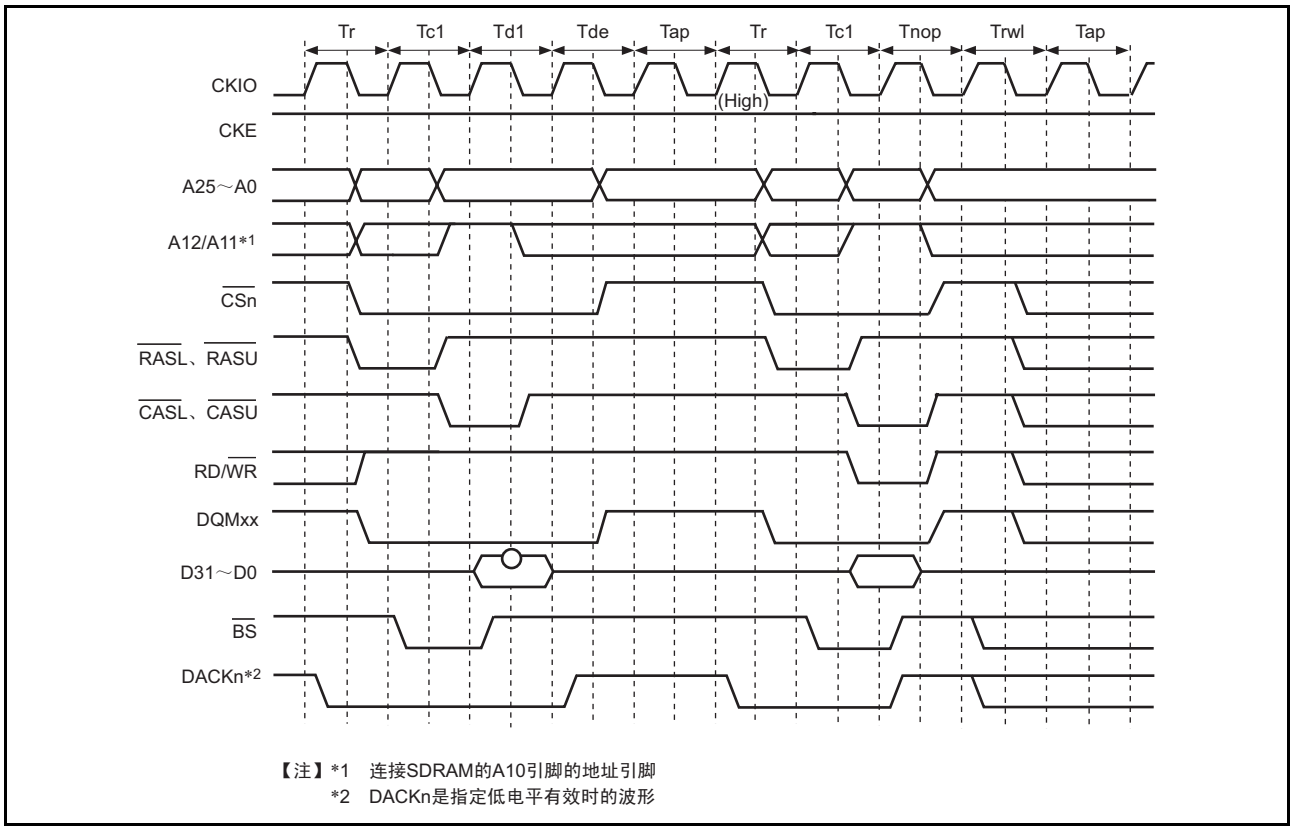


图 9.30　低频率模式中的存取时序

（11）掉电模式

将 SDCR 的 PDOWN 位置 1 时，在非存取时将 CKE 设置为 L 电平，并将 SDRAM 转换为掉电模式。因此在非存取时可大幅降低功耗。但在发生存取时，为了解除 SDRAM 的掉电模式需要插入使 CKE 断言的周期，因此会发生 1 个周期的消耗。掉电模式中的存取时序如图 9.31 所示。

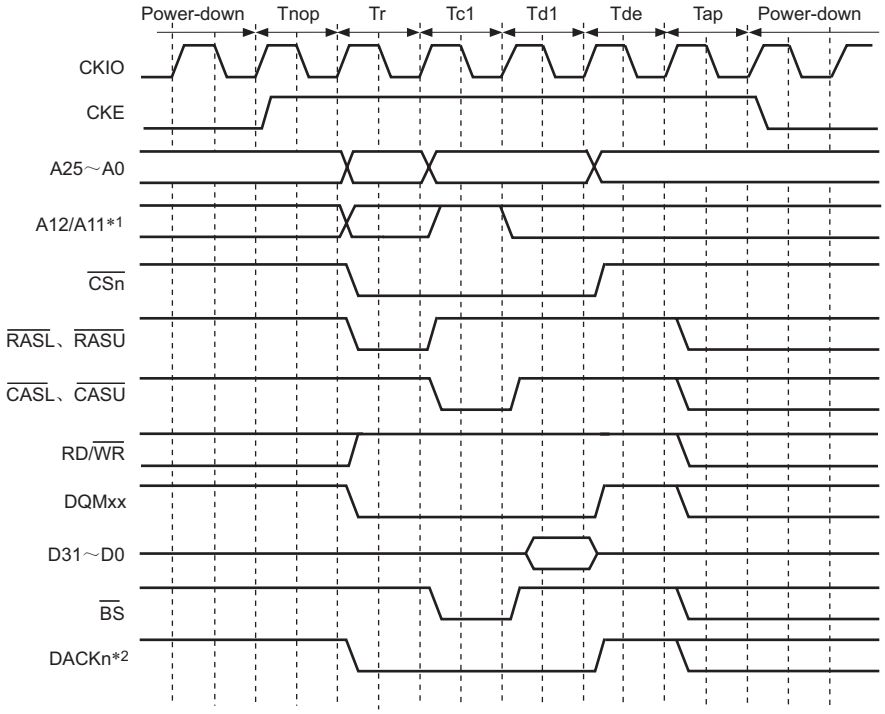


图 9.31　掉电模式中的存取时序

（12）上电顺序

在使用 SDRAM 时，需要在上电后并等待至少 100μs 后对 SDRAM 进行模式设定。请通过上电复位产生电路或软件等实现至少 100μs 的间隔。

为了正确初始化 SDRAM，先设定 BSC 的寄存器，然后对 SDRAM 的模式寄存器进行写入。在设定 SDRAM 的模式寄存器时，通过 $\overline{\text{CSn}}$、$\overline{\text{RASU}}$、$\overline{\text{RASL}}$、$\overline{\text{CASU}}$、$\overline{\text{CASL}}$ 和 RD/$\overline{\text{WR}}$ 的组合，将当时的地址信号值输入到 SDRAM。假设要设定的值为 X，通过对 X+（区域 2：H'FFFC4000、区域 3：H'FFFC5000）进行字写入，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 1）或突发读 / 突发写 （突发长度为 1）、CAS 等待时间 2 ～ 3、Wrap 型 = 顺序以及突发长度 1 时，将任意数据字写到表 9.18 所示的存取地址。此时给外部地址引脚 A12 以上的位输出 0。

表 9.18　写 SDRAM 模式寄存器时的存取地址

- 区域2的设定
  突发读/单次写 （突发长度为1）时

数据总线宽度	CAS 等待时间	存取地址	外部地址引脚
16 位	2	H'FFFC4440	H'0000440
	3	H'FFFC4460	H'0000460
32 位	2	H'FFFC4880	H'0000880
	3	H'FFFC48C0	H'00008C0

突发读/突发写 （突发长度为1）时

数据总线宽度	CAS 等待时间	存取地址	外部地址引脚
16 位	2	H'FFFC4040	H'0000040
	3	H'FFFC4060	H'0000060
32 位	2	H'FFFC4080	H'0000080
	3	H'FFFC40C0	H'00000C0

- 区域3的设定
  突发读/单次写 （突发长度为1）的情况

数据总线宽度	CAS 等待时间	存取地址	外部地址引脚
16 位	2	H'FFFC5440	H'0000440
	3	H'FFFC5460	H'0000460
32 位	2	H'FFFC5880	H'0000880
	3	H'FFFC58C0	H'00008C0

突发读 / 突发写 （突发长度为 1）时

数据总线宽度	CAS 等待时间	存取地址	外部地址引脚
16 位	2	H'FFFC5040	H'0000040
	3	H'FFFC5060	H'0000060
32 位	2	H'FFFC5080	H'0000080
	3	H'FFFC50C0	H'00000C0

模式寄存器的设定时序如**图 9.32** 所示。首先发送全部存储体的预充电命令 （PALL），其次发送 8 次自动刷新命令 （REF）。最后发送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。另外，在 MRS 和下一次发送的命令之间至少插入 1 个空闲周期。

在接通电源后并且在进行全部存储体的预充电 （PALL）前，SDRAM 必须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。

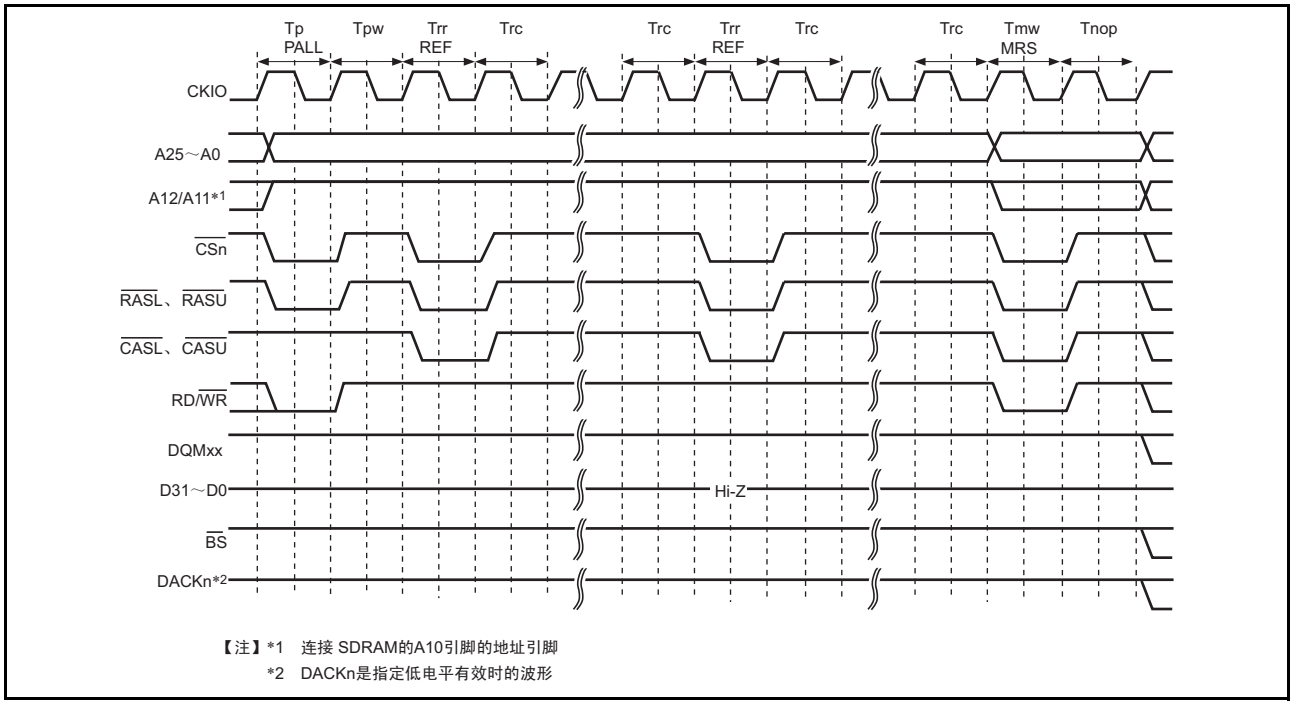


图 9.32　SDRAM 模式寄存器的写入时序 （以 JEDEC 为基准）

（13）低功率 SDRAM

低功率 SDRAM 是可通过与通常 SDRAM 相同的协议进行存取的存储器。

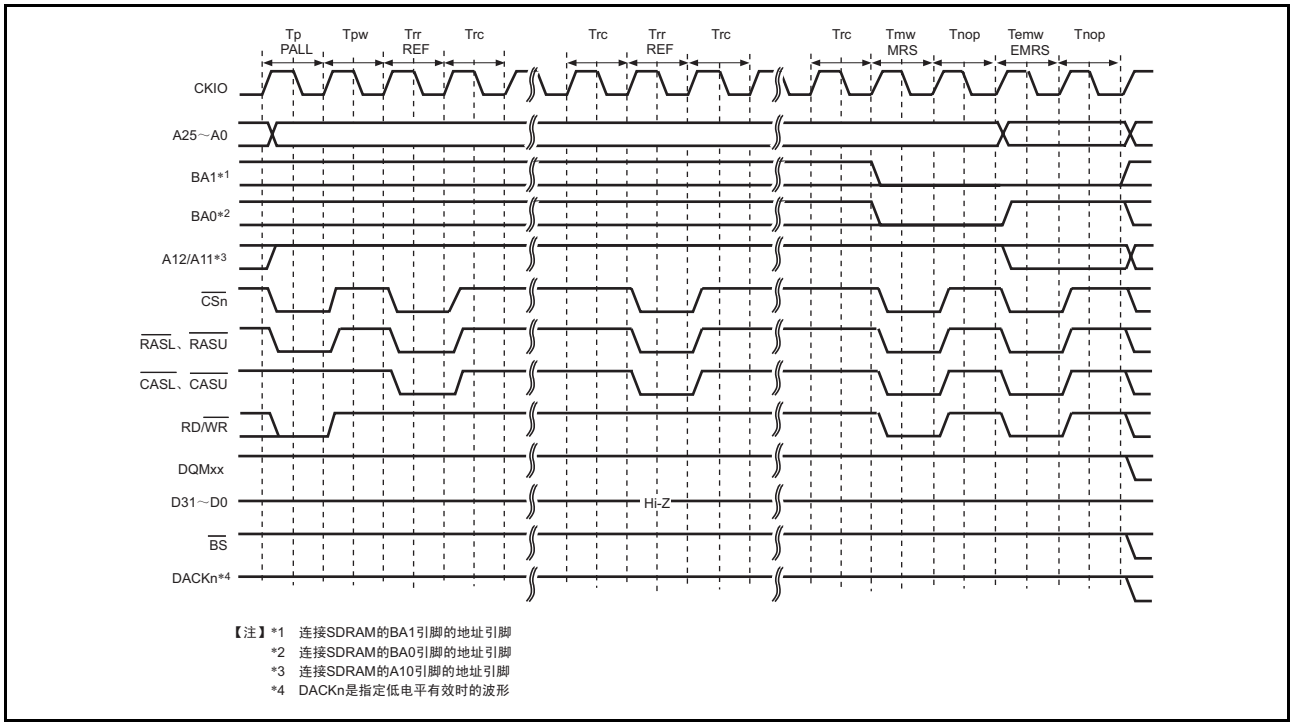
低功率 SDRAM 与通常 SDRAM 规格的区别是：可详细设定在自刷新时使存储器的一部分进入自刷新状态的部分刷新，以及按照用户使用项目（温度）进行刷新时的低功率化等控制。部分刷新对即使在工作区（除特定区域外）内丢失数据也没问题的系统有效。详细情况请参照使用的低功率 SDRAM 数据表。

在低功率 SDRAM 中具有模式寄存器（与通常 SDRAM 相同）和扩展模式寄存器。扩展模式寄存器写入命令称为 EMRS，本 LSI 支持发送 EMRS 命令。

按下表发送 EMRS。例如用长字将 H'0YYYYYYY 的数据写入 H'FFFC5XX0 时，就对 CS3 空间按 PALL→REF×8→MRS→EMRS 的顺序发送命令。因此发送 MRS 时的地址为 H'0000XX0，发送 EMRS 时的地址为 H'YYYYYYY。另外用长字将 H'1YYYYYYY 的数据写入 H'FFFC5XX0 时，就对 CS3 空间按 PALL→MRS→EMRS 的顺序发送命令。

表 9.19　发送 EMRS 命令时的输出地址

发送命令	存取命令	存取数据	写存取长度	发送 MRS 命令时的地址	发送 EMRS 命令时的地址
CS2　MRS	H'FFFC4XX0	H'********	16 位	H'0000XX0	--------
CS3　MRS	H'FFFC5XX0	H'********	16 位	H'0000XX0	--------
CS2　MRS+EMRS（有刷新）	H'FFFC4XX0	H'0YYYYYYY	32 位	H'0000XX0	H'YYYYYYY
CS3　MRS+EMRS（有刷新）	H'FFFC5XX0	H'0YYYYYYY	32 位	H'0000XX0	H'YYYYYYY
CS2　MRS+EMRS（无刷新）	H'FFFC4XX0	H'1YYYYYYY	32 位	H'0000XX0	H'YYYYYYY
CS3　MRS+EMRS（无刷新）	H'FFFC5XX0	H'1YYYYYYY	32 位	H'0000XX0	H'YYYYYYY



【注】*1　连接SDRAM的BA1引脚的地址引脚
*2　连接SDRAM的BA0引脚的地址引脚
*3　连接SDRAM的A10引脚的地址引脚
*4　DACKn是指定低电平有效时的波形

图 9.33　发送 EMRS 命令的时序

- 深度掉电模式

  在低功率SDRAM中有被称为深度掉电模式的低功耗模式。

  部分自刷新只对某特定区域进行自刷新，而深度掉电模式中存储器的任何区域都不进行自刷新。

  该模式对将整个存储器作为工作区域的系统有效。

  在SDCR的DEEP位置1、RFSH位置1的状态下，向RMODE位写入1时，低功率SDRAM转换为深度掉电模式。向RMODE位写入0时，CKE成为高电平，并解除深度掉电模式。请在重新进行上电顺序后，进行解除深度掉电后的存取。

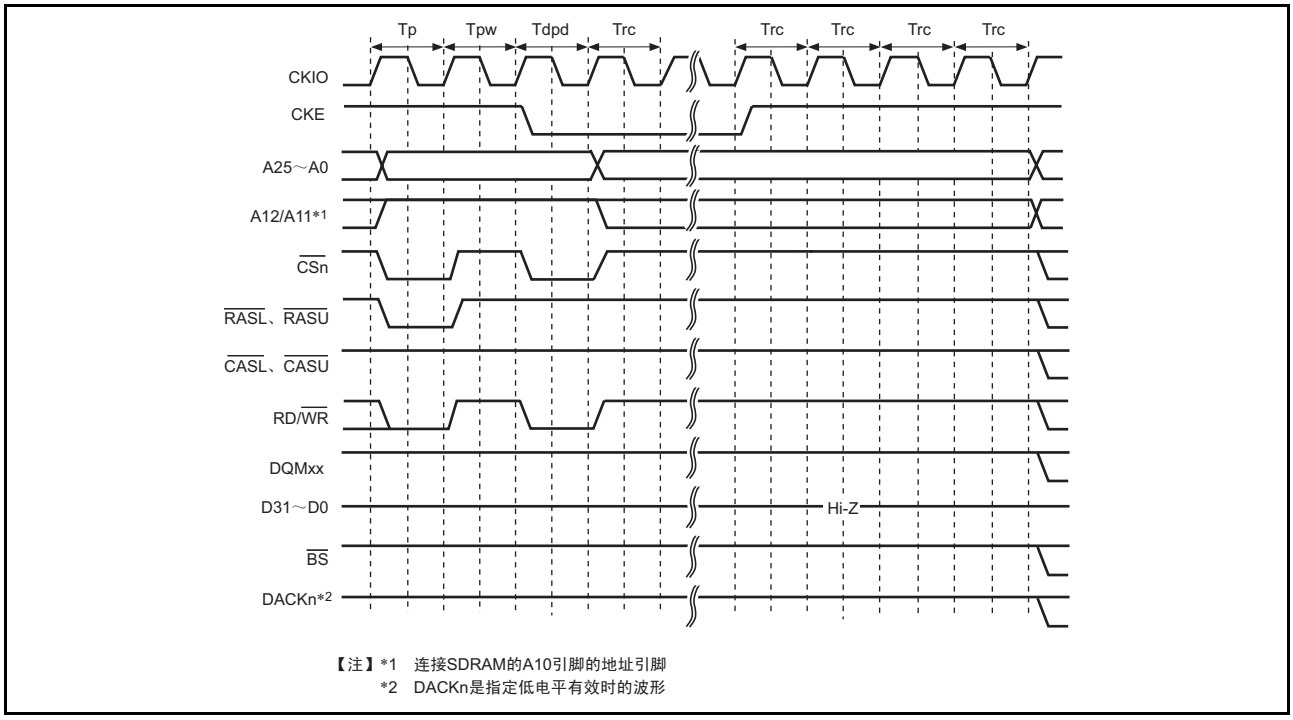


图 9.34　深度掉电模式转换时序

9.5.7　突发 ROM （时钟异步）接口

突发 ROM（时钟异步）接口通过称为突发模式或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 $\overline{RD}$ 信号置为取反而只转换地址，进行第 2 次及以后的存取。第 2 次及以后的存取在 CKIO 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 次及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 （时钟异步）时，$\overline{BS}$ 信号只对最初的存取周期断言。另外，外部等待输入也只对最初的存取周期有效。

当通过突发 ROM （时钟异步）接口进行单次存取或写存取 （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 9.20**、时序图如**图 9.35** 所示。

表 9.20　总线宽度、存取长度和突发数的关系

总线宽度	存钱长度	CSnWCR.BST[1:0] 位	突发数	存取次数
8 位	8 位	没有影响	1	1
	16 位	没有影响	2	1
	32 位	没有影响	4	1
	16 字节	00	16	1
		01	4	4
16 位	8 位	没有影响	1	1
	16 位	没有影响	1	1
	32 位	没有影响	2	1
	16 字节	00	8	1
		01	2	4
		10*	4	2
			2、4、2	3
32 位	8 位	没有影响	1	1
	16 位	没有影响	1	1
	32 位	没有影响	1	1
	16 位	没有影响	4	1

【注】 * 总线宽度为 16 位、存取长度为 16 字节、CSnWCR.BST[1:0] 位为“10”时，突发数和存取次数因存取起始地址而发生变化，当地址为 H'xxx0、H'xxx8 时进行 4-4 突发存取，当地址为 H'xxx4、H'xxxC 时进行 2-4-2 突发存取。

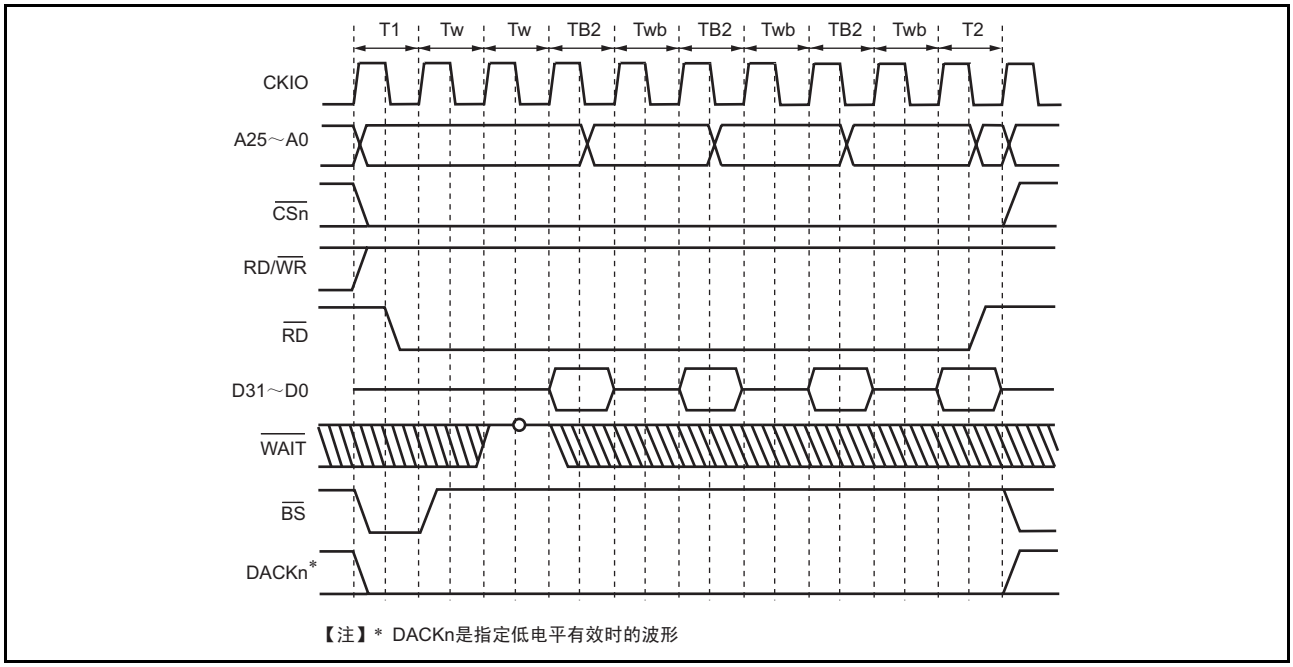


图 9.35　突发 ROM （时钟异步）的存取
（32 位总线宽度、 16 字节传送 （突发数 4）、初次存取等待 2、第 2 次及以后的存取等待 1）

9.5.8　带字节选择的 SRAM 接口

带字节选择的 SRAM 接口是在读或写的任意总线周期中输出字节选择引脚信号（$\overline{\text{WEn}}$）的存储器接口。此接口有 16 位数据引脚，用于存取带高位字节选择引脚和低位字节选择引脚 （如 UB 或 LB）的 SRAM。

当 CSnWCR 的 BAS 位为 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， $\overline{\text{WEn}}$ 引脚的时序和通常空间接口不同，从 $\overline{\text{WEn}}$ 引脚输出字节选择信号。基本存取时序如图 9.36 所示。特别是在进行写操作时，按照字节选择引脚 （$\overline{\text{WEn}}$）的时序写存储器。所以请确认所使用的存储器数据表。

当 CSnWCR 的 BAS 位为 1 时， $\overline{\text{WEn}}$ 引脚和 RD/$\overline{\text{WR}}$ 引脚的时序发生变化。基本存取时序如图 9.37 所示。特别是在进行写操作时，按照写允许引脚 （RD/$\overline{\text{WR}}$）时序写入存储器。请通过设定 CSnWCR 的 HW[1:0] 位，确保从 RD/$\overline{\text{WR}}$ 取反到写数据的数据保持时间。软件等待设定时的存取时序如图 9.38 所示。

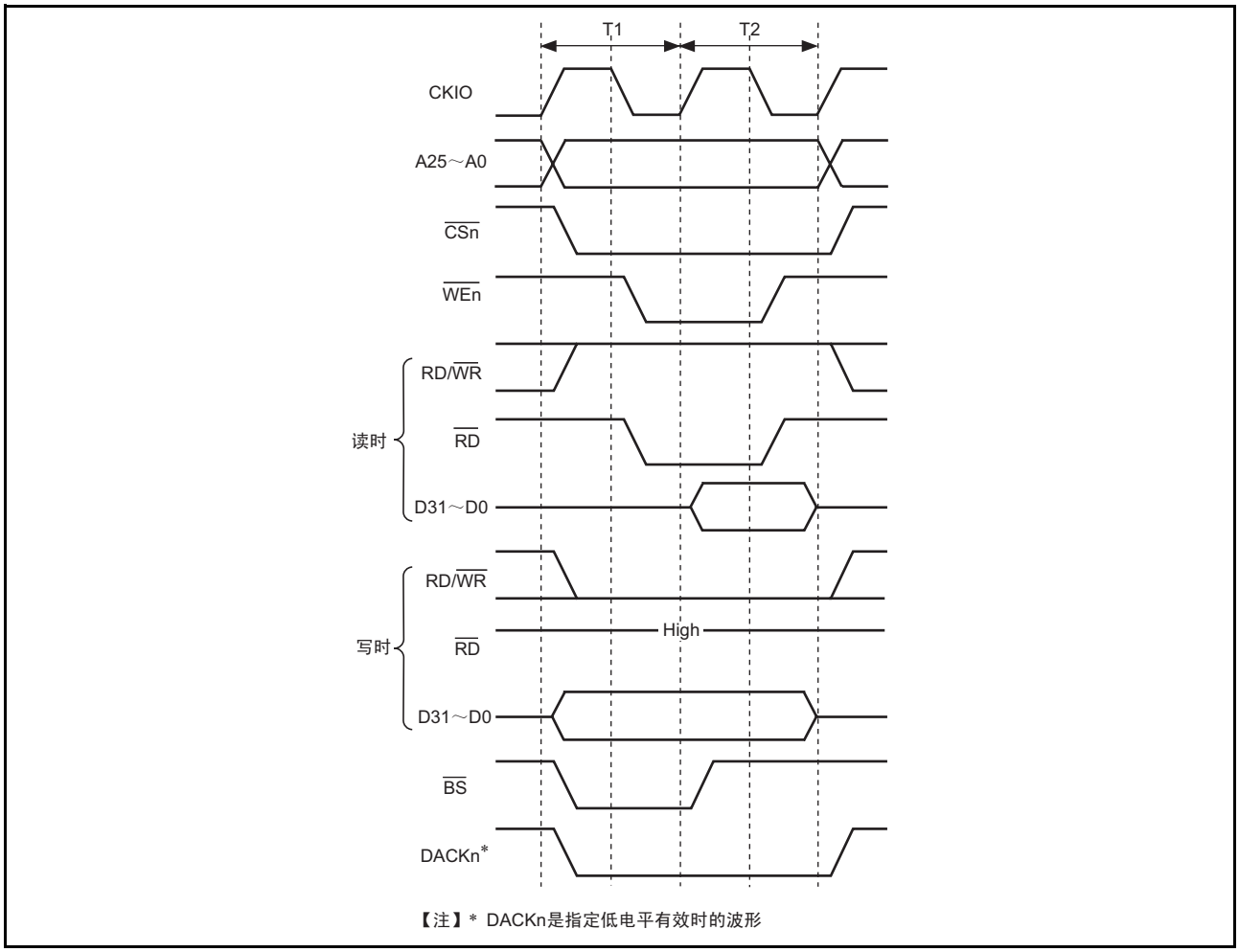


图 9.36　BAS=0、带字节选择的 SRAM 的基本存取时序

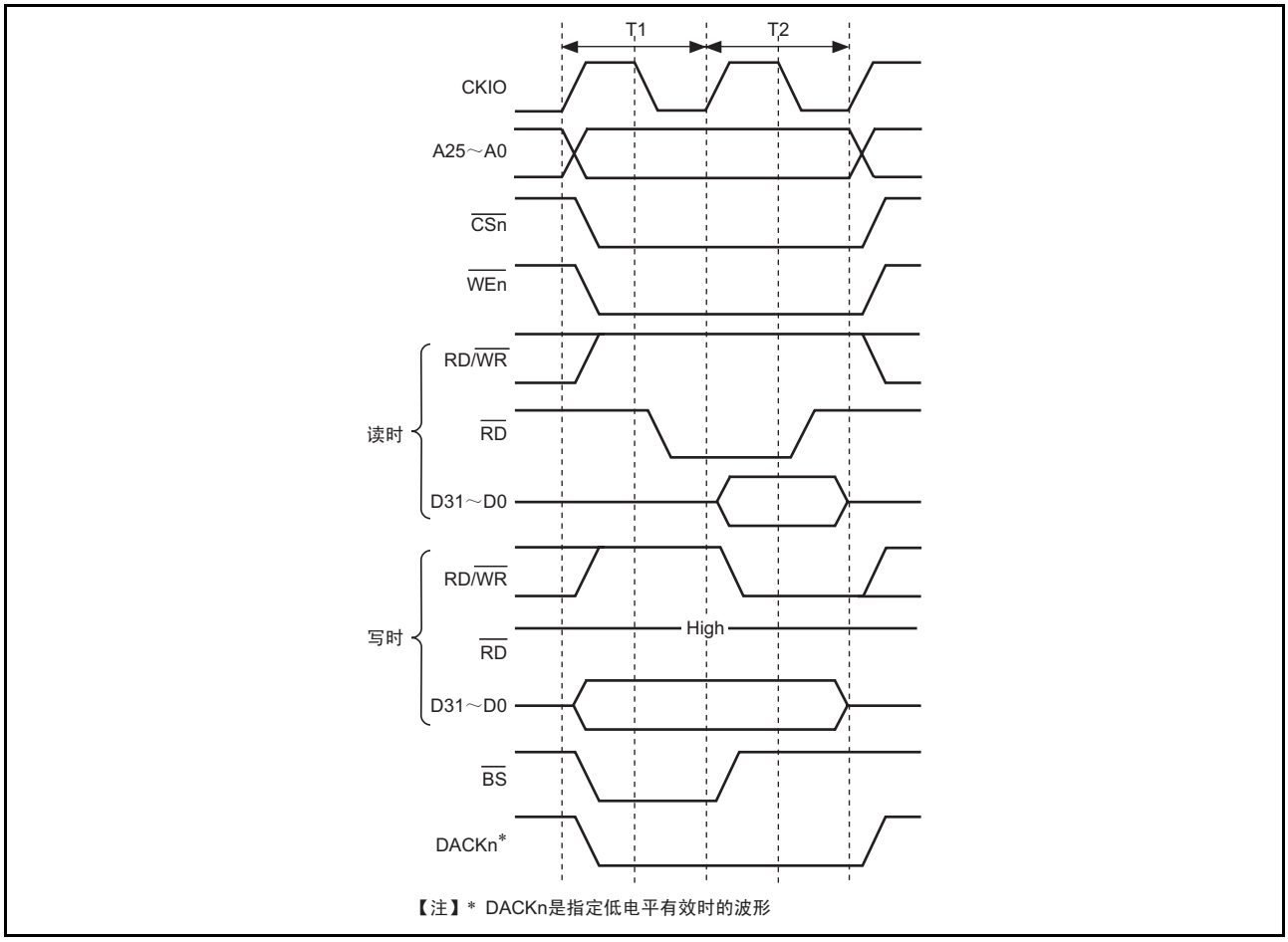


图 9.37　BAS=1、带字节选择的 SRAM 的基本存取时序

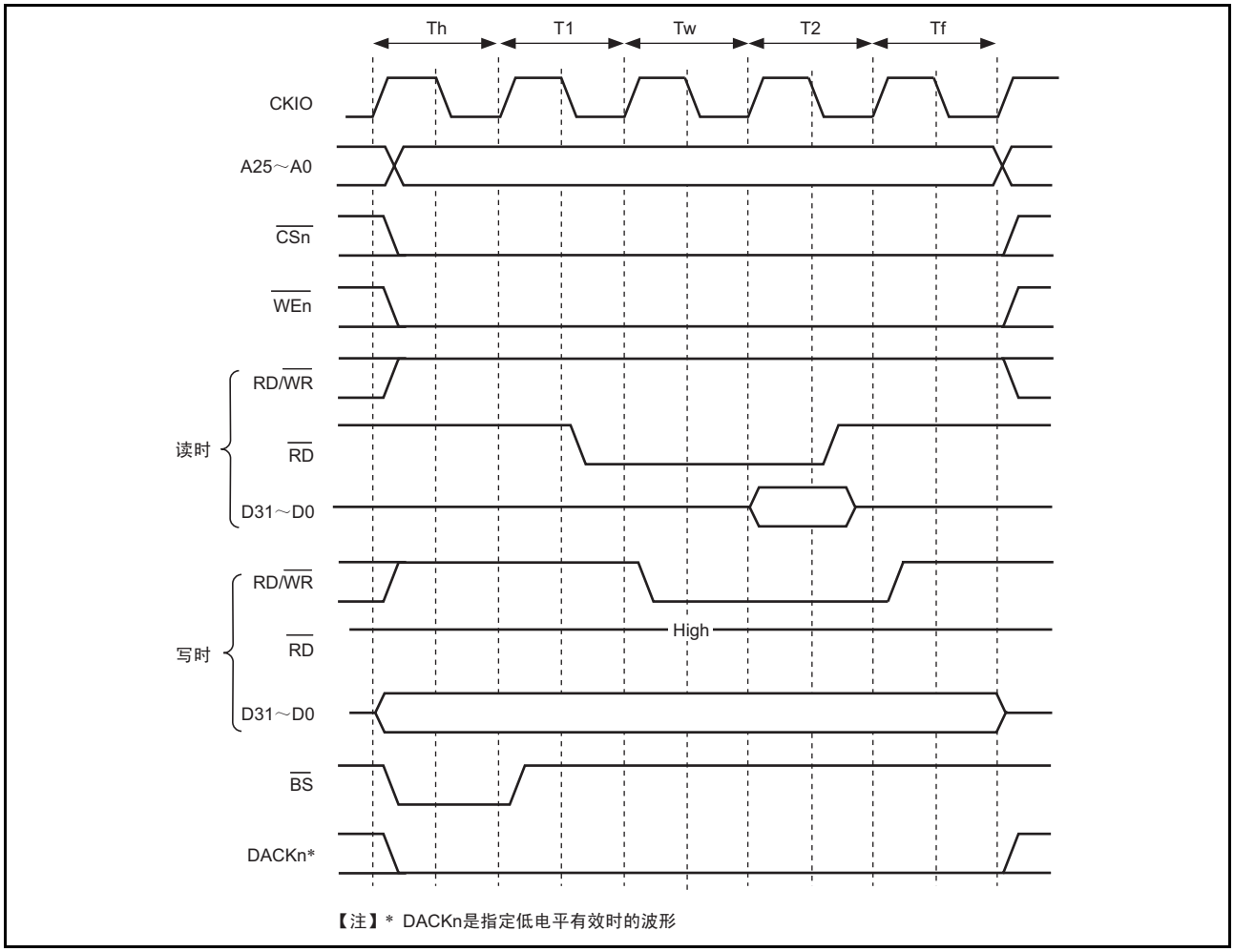


图 9.38　BAS=1、带字节选择的 SRAM 的等待时序
（SW[1:0]=01、 WR[3:0]=0001、 HW[1:0]=01）

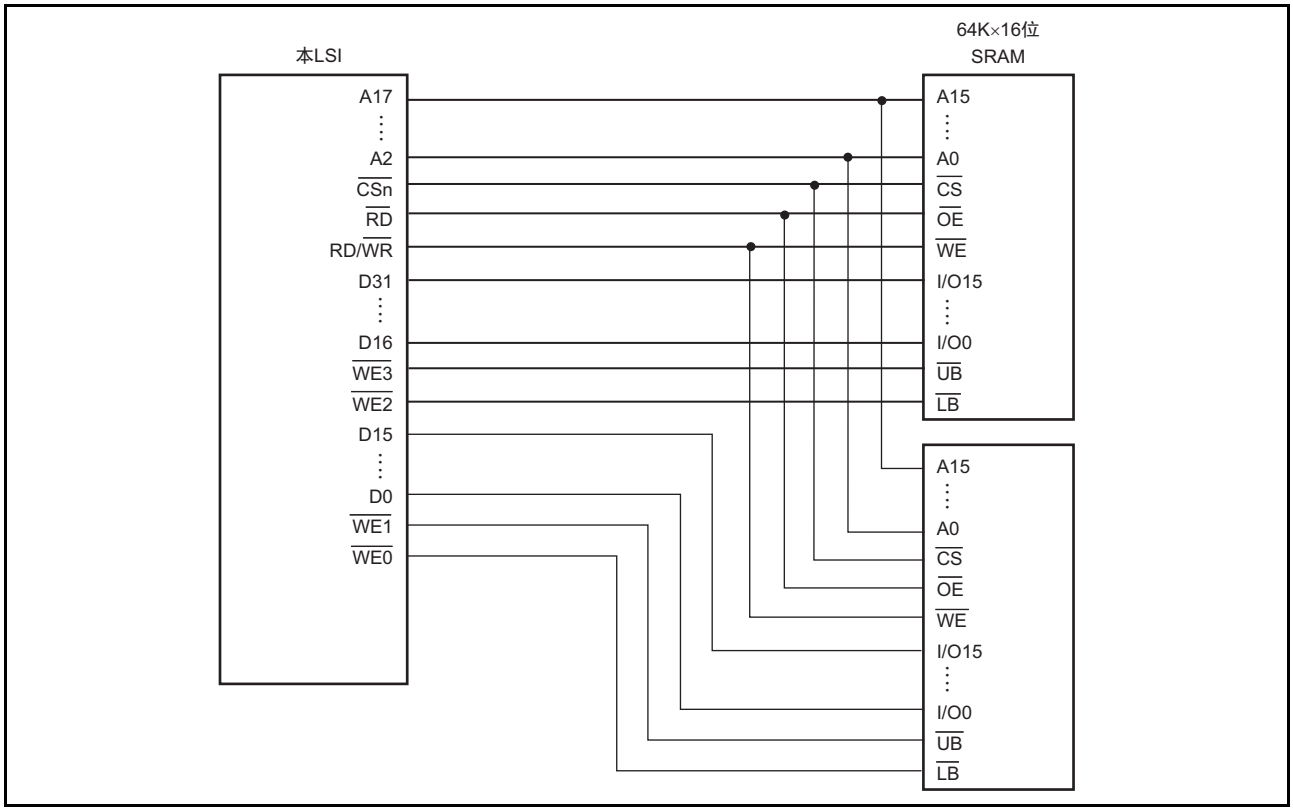


图 9.39　带字节选择的 SRAM 连接例子 （32 位数据宽度）

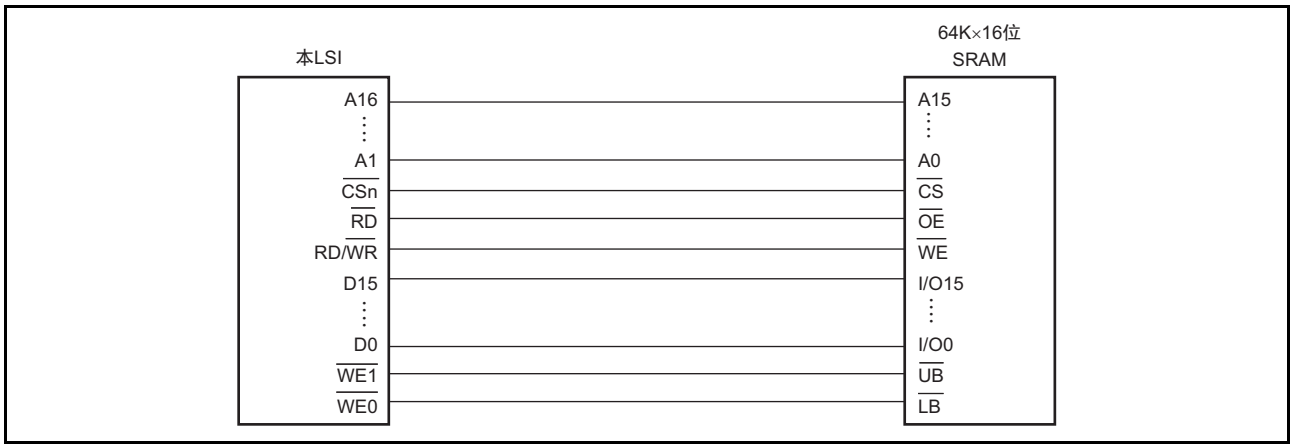


图 9.40　带字节选择的 SRAM 连接例子 （16 位数据宽）

9.5.9　PCMCIA 接口

本 LSI 通过将 CSnBCR （n=5、 6）的 TYPE[2:0] 位设定为 B'101，将区域 5 和区域 6 设定为符合 JEIDA 规格 Ver4.2 （PCMCIA2.1Rev2.1）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 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，区域 5 的前 32MB 和后 32MB 分别成为 IC 存储卡或 I/O 卡接口。

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

本 LSI 和 PCMCIA 卡的连接例子如**图 9.41** 所示。为了能进行 PCMCIA 卡的热插拔 （在提供系统电源时插拔卡），需要在本 LSI 的总线接口和 PCMCIA 卡之间连接 3 态缓冲器。

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接口为独立规定的接口规格。

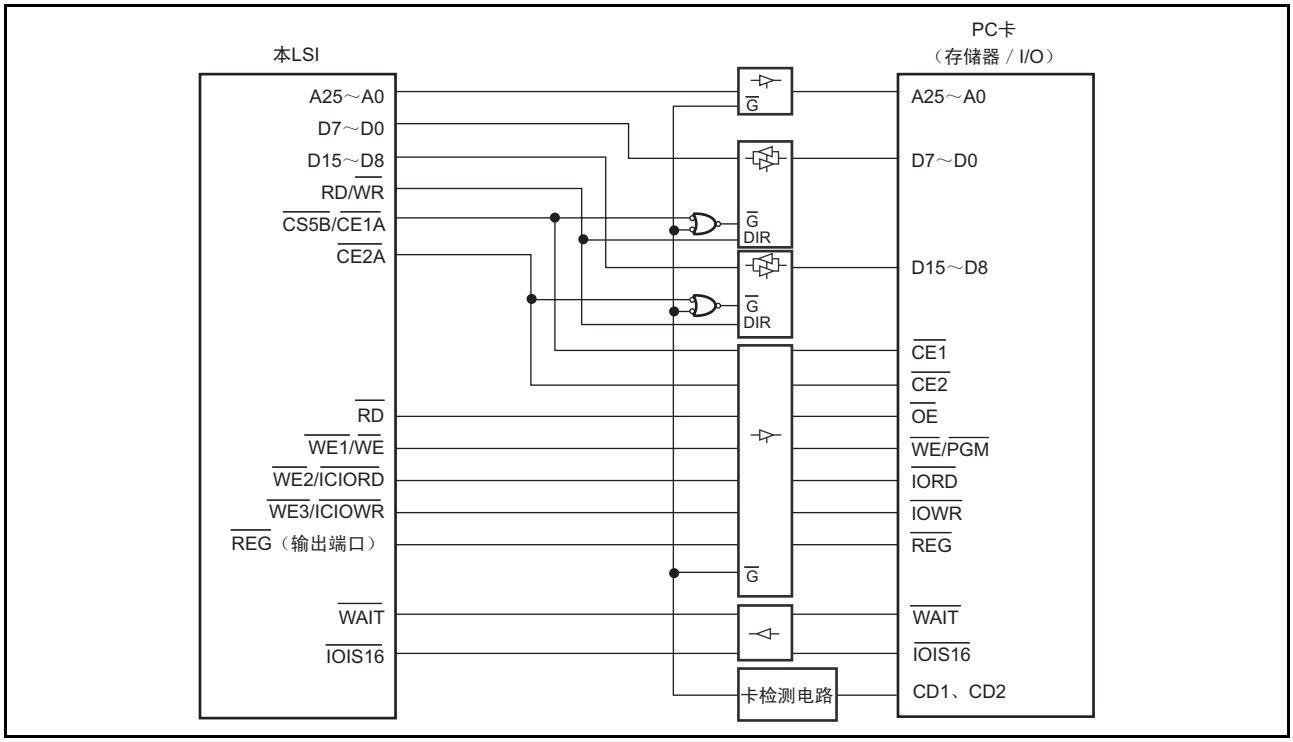


图 9.41　PCMCIA 接口的连接例子

（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 9.42** 所示。在将区域 5 和区域 6 设定为 PCMCIA 接口的情况下，根据 CS5WCR 和 CS6WCR 的 SA[1:0] 位的设定，可作为 IC 存储卡接口进行总线存取。如果外部总线频率（CKIO）变高，对于 $\overline{RD}$ 和 $\overline{WE}$ 来说，地址（A25 ～ A0）、卡的允许信号（$\overline{CE1A}$、$\overline{CE2A}$、$\overline{CE1B}$、$\overline{CE2B}$）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。对此，本 LSI 能通过 CS5WCR 或 CS6WCR，分别给区域 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 $\overline{WAIT}$ 引脚产生的硬件等待。 PCMCIA 存储器的总线等待时序如**图 9.43** 所示。

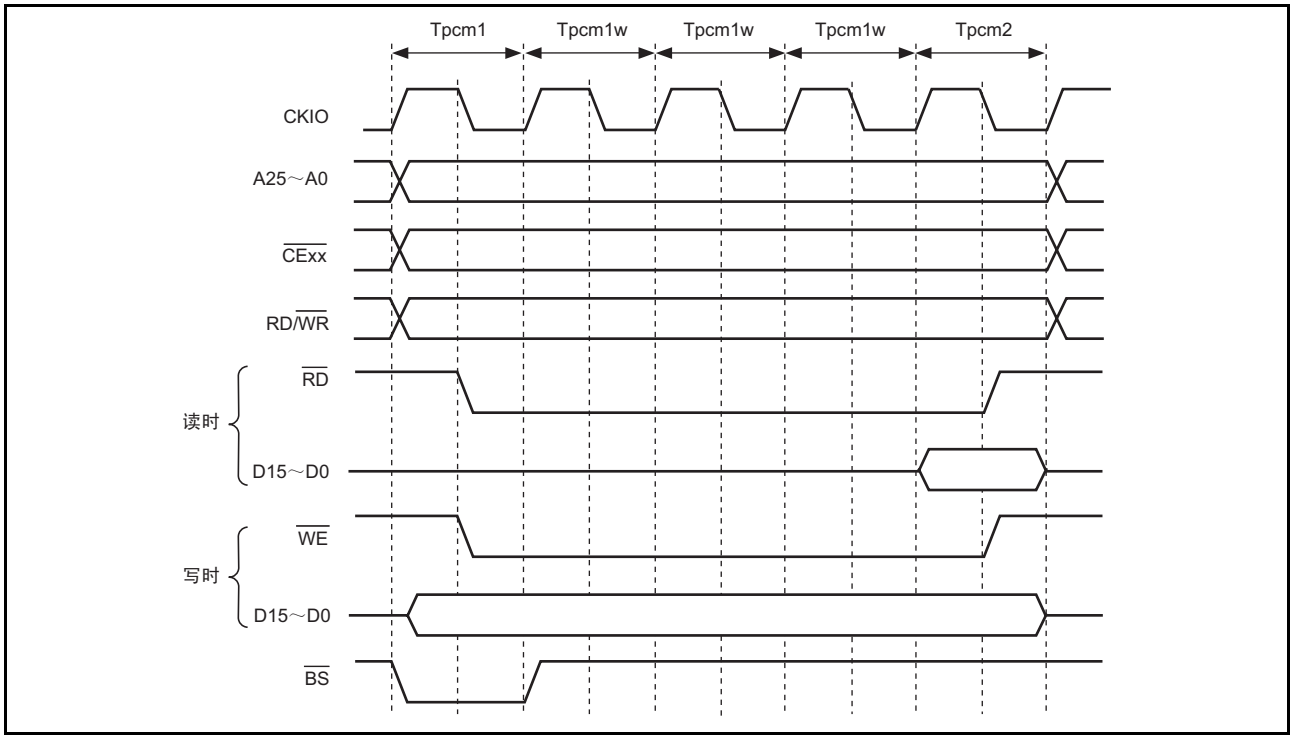


图 9.42　PCMCIA 存储卡接口的基本时序

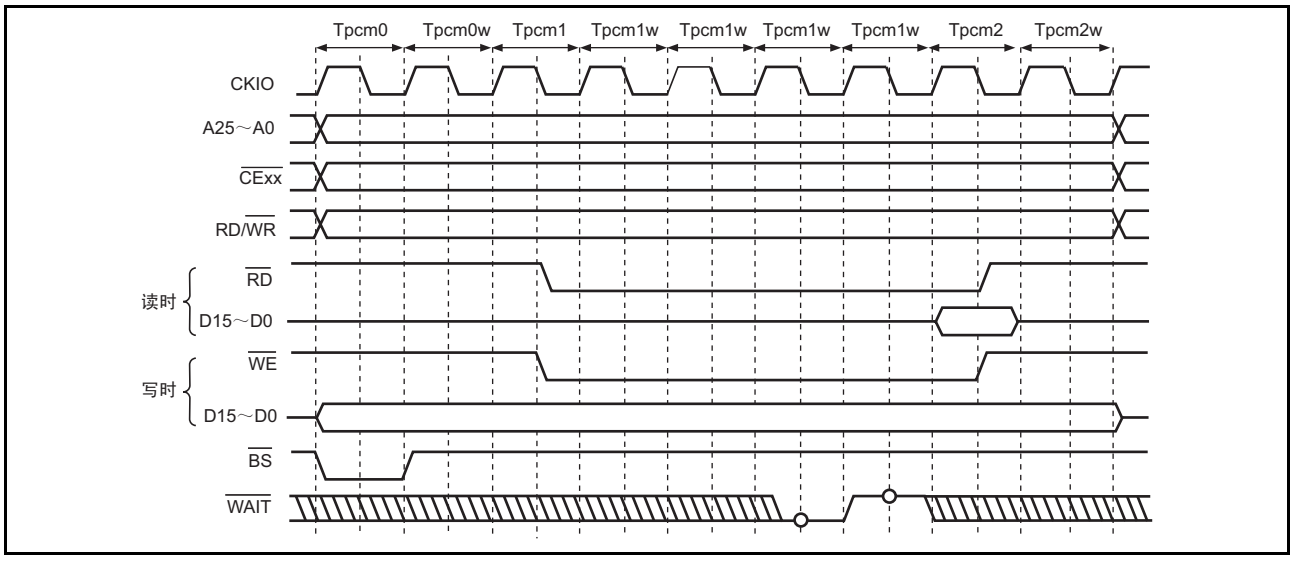


图 9.43　PCMCIA 存储卡接口的等待时序
（TED[3:0]=B'0010、 PCW[3:0]=B'0000、 TEH[3:0]=B'0001、硬件等待 1）

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，如图 9.44 所示，在通用存储器和属性存储器总共的存储空间不超过 32M 字节并且够用时，通过将 A24 引脚用作 $\overline{\text{REG}}$ 信号，可将存储空间用作 16M 字节的通用存储空间和 16M 字节的属性存储空间。

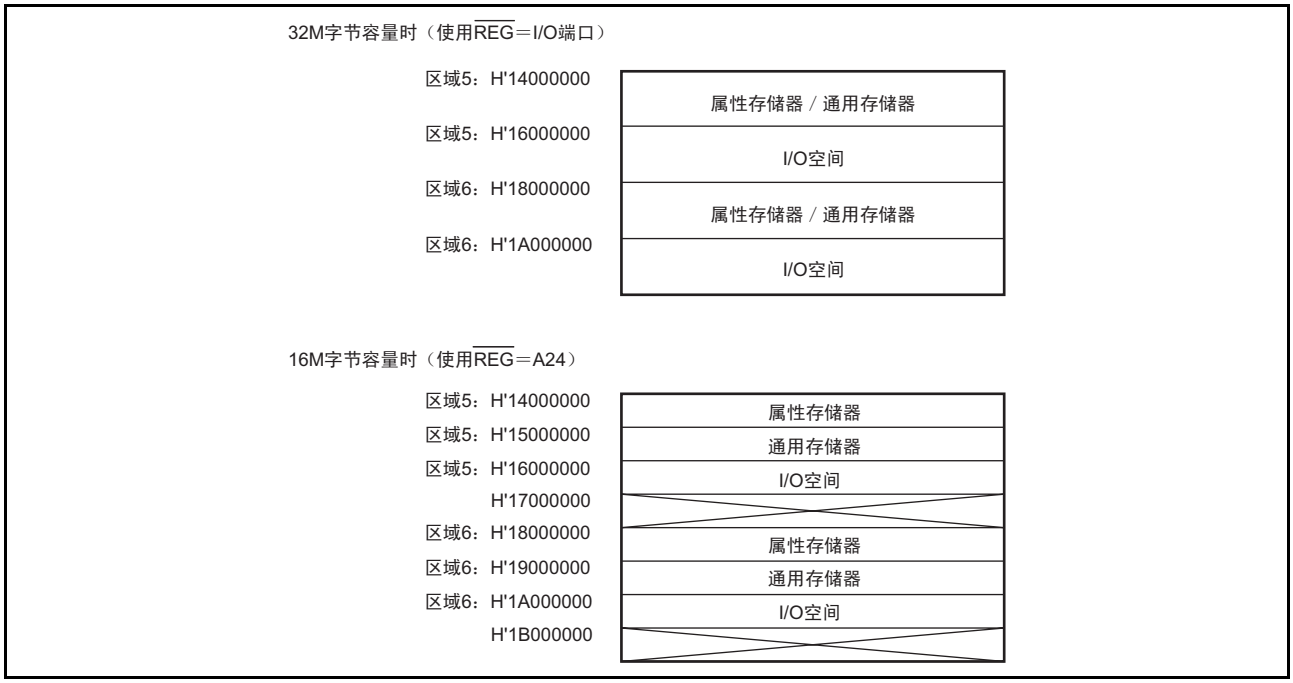


图 9.44　PCMCIA 空间的分配例子 （CS5WCR.SA[1:0]=B'10、 CS6WCR.SA[1:0]=B'10）

（2） I/O 卡接口的基本时序

PCMCIA 的 I/O 卡接口时序如图 9.45 和图 9.46 所示。

在存取 PCMCIA 的 I/O 卡时，请存取高速缓存无效区域。

根据 CS5WCR 以及 CS6WCR 的 SA[1:0] 位的设定和存取的地址，切换 I/O 卡接口和 IC 存储器卡接口。

在小端法模式中将 PCMCIA 卡作为 I/O 卡接口存取时，可使用 $\overline{\text{IOIS16}}$ 引脚，进行 I/O 总线宽度的动态总线宽度变更。区域 5 或区域 6 的总线宽度设定为 16 位时，如果在字长度的 I/O 卡总线周期中 $\overline{\text{IOIS16}}$ 信号为高电平，总线宽度就被识别为 8 位，并且在执行中的 I/O 卡总线周期中只进行 8 位数据存取，并自动继续进行剩余的 8 位数据存取。

在 Tpci0 以及所有 Tpci0w 周期（将 TED[3:0] 位设定为大于等于 1.5 周期时）CKIO 的下降沿，采样 $\overline{\text{IOIS16}}$ 信号，并从 Tpci0 的采样点开始 CKIO1.5 周期后，反映在 $\overline{\text{CE2A}}$ 或 $\overline{\text{CE2B}}$ 信号。必须在所有的采样点确定 $\overline{\text{IOIS16}}$ 信号，且不能在中途发生变化。

请设定 TED[3:0] 位，以便满足所用 PC 卡从 $\overline{\text{ICIORD}}$ 和 $\overline{\text{ICIOWR}}$ 到 $\overline{\text{CE1}}$ 或 $\overline{\text{CE2}}$ 的设置规格。。

动态总线宽度变更的基本波形如图 9.46 所示。

而且在大端法模式下不支持 $\overline{\text{IOIS16}}$ 信号。当为大端法模式时，请将 $\overline{\text{IOIS16}}$ 信号固定为低电平。

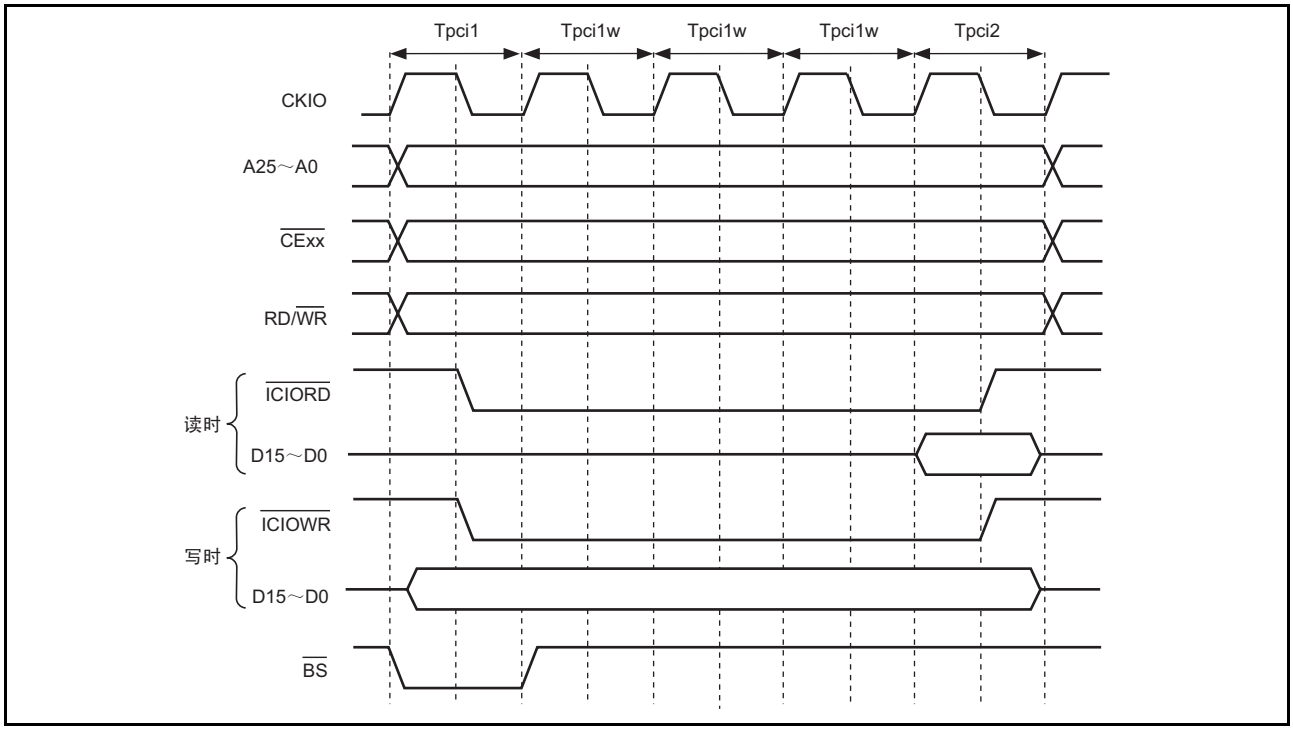


图 9.45　PCMCIA I/O 卡接口的基本时序

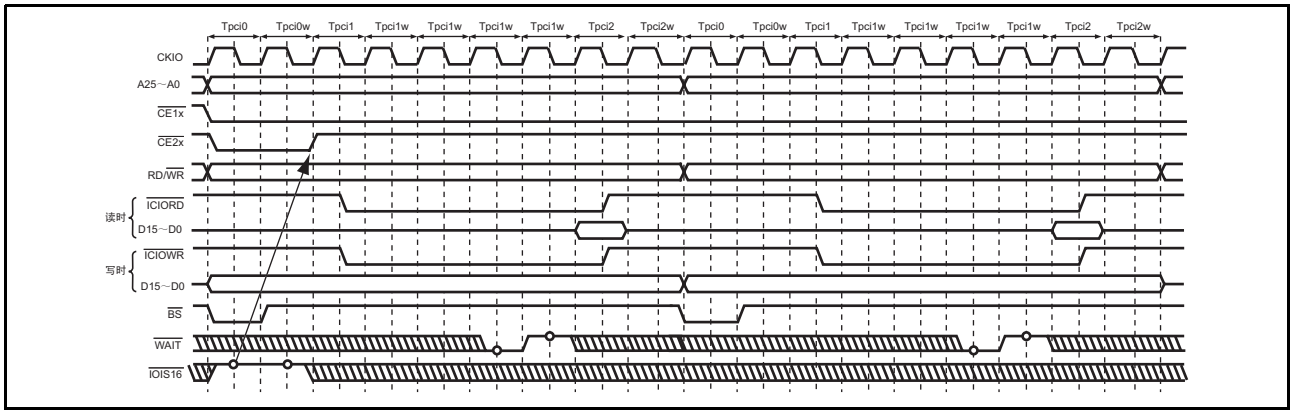


图 9.46　PCMCIA I/O 卡接口的动态总线宽度变更时序
（TED[3:0]=B'0010、 PCW[3:0]=B'0000、 TEH[3:0]=B'0001、硬件等待 1）

9.5.10　突发 MPX-I/O 接口

突发 MPX 器件的连接列子和突发 MPX 空间的存取时序分别如图 9.47 和图 9.48 ～图 9.51 所示。

通过设定 CS6BCR 的 TYPE[2:0] 位，能将区域 6 设定为地址 / 数据多路复用 I/O 接口。通过本接口能很容易地连接使用地址 / 数据多路复用 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关系请参照“9.4.3 CSn 空间等待控制寄存器 （CSnWCR）（n=0 ～ 7）（5） 突发 MPX-I/O”的 CS6WCR 说明。

将通常的地址输出到地址引脚 A25 ～ A0。

在使用突发 MPX-I/O 接口时，总线宽度固定为 32 位。所以请将 CS6BCR 的 BSZ[1:0] 位设定为 32 位。另外，能插入软件等待以及 $\overline{\text{WAIT}}$ 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。

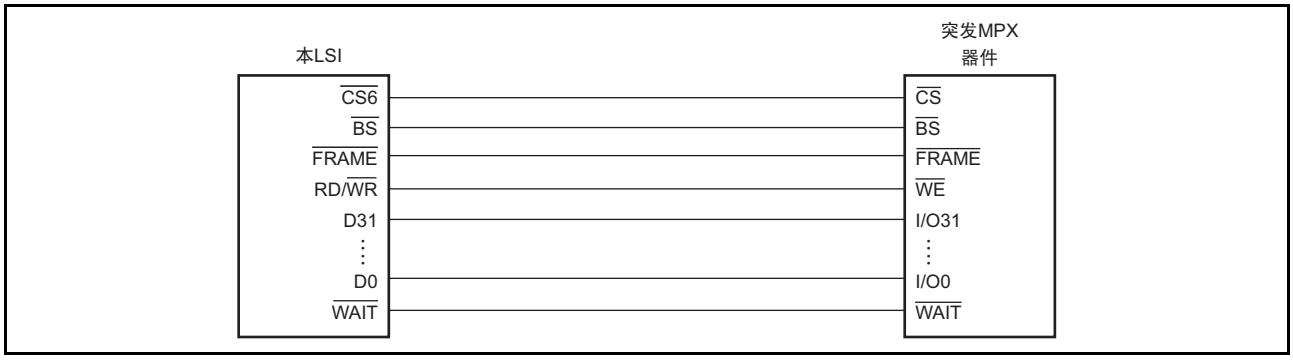


图 9.47　突发 MPX 器件的连接例子

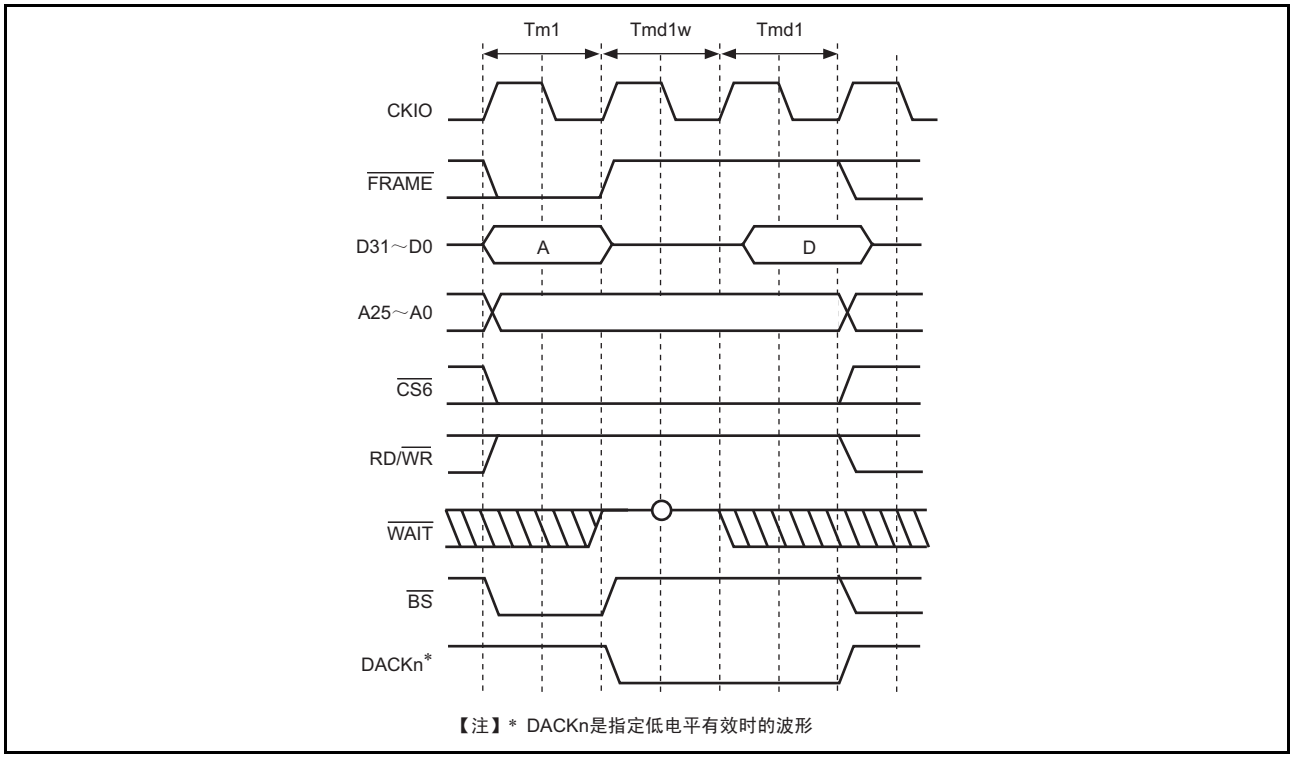


图 9.48　突发 MPX 空间的存取时序
（单次读、无等待或者软件等待 1）

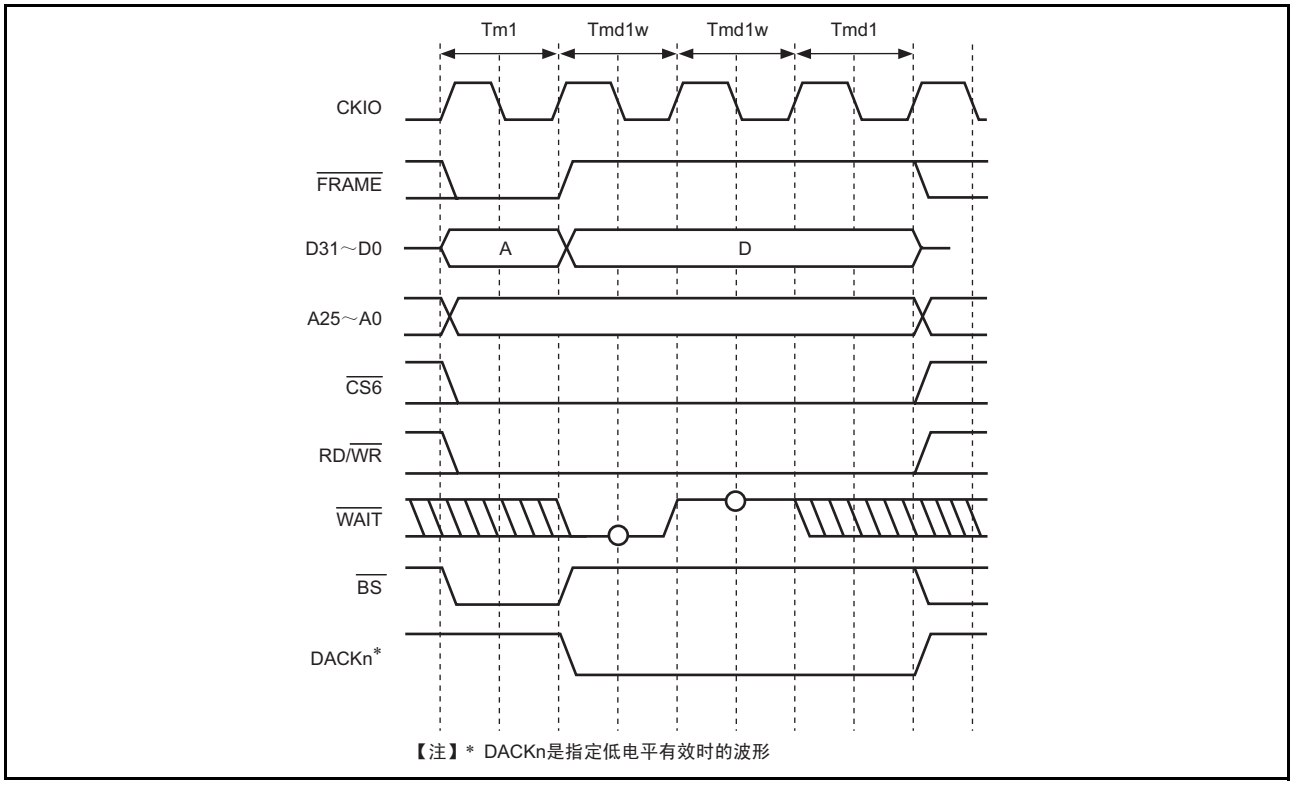


图 9.49　突发 MPX 空间的存取时序
（单次写、软件等待 1、硬件等待 1）

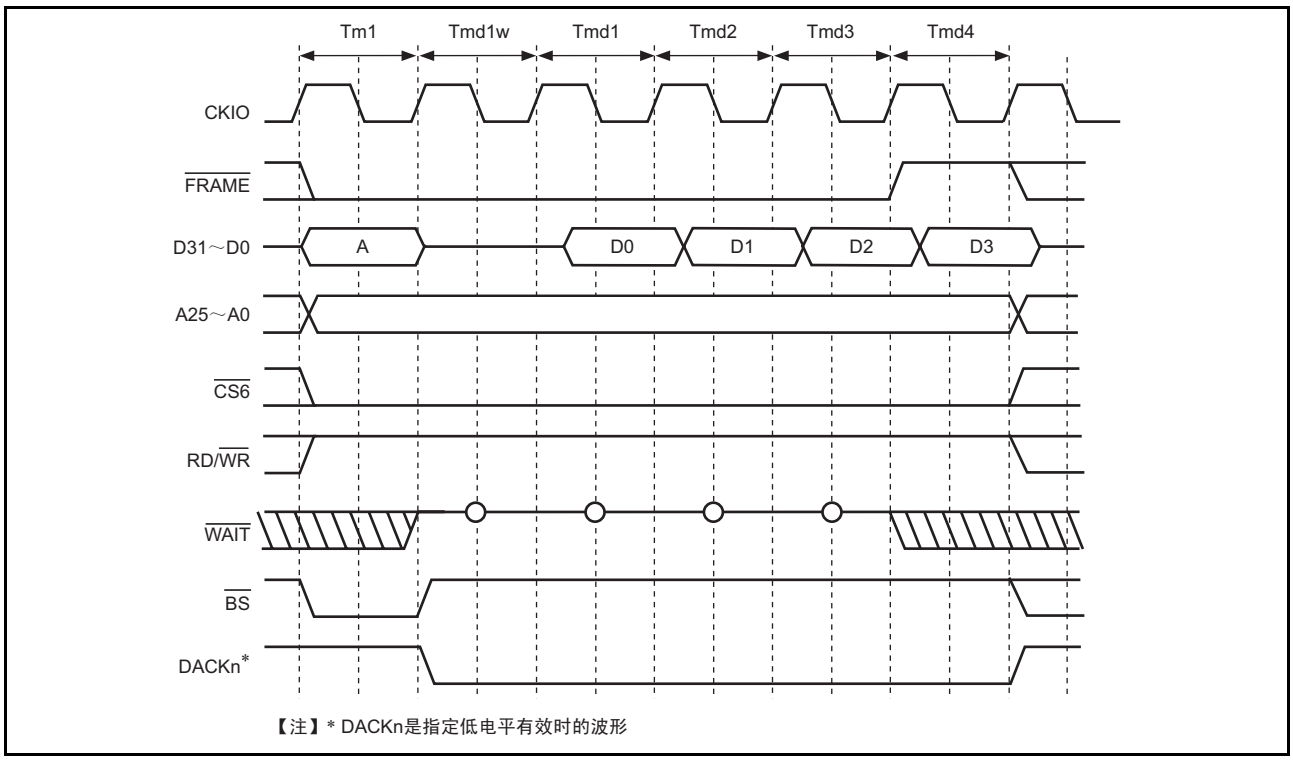


图 9.50　突发 MPX 空间的存取时序 （突发读、无等待或者软件等待 1、 CS6WCR.MPXMD=0）

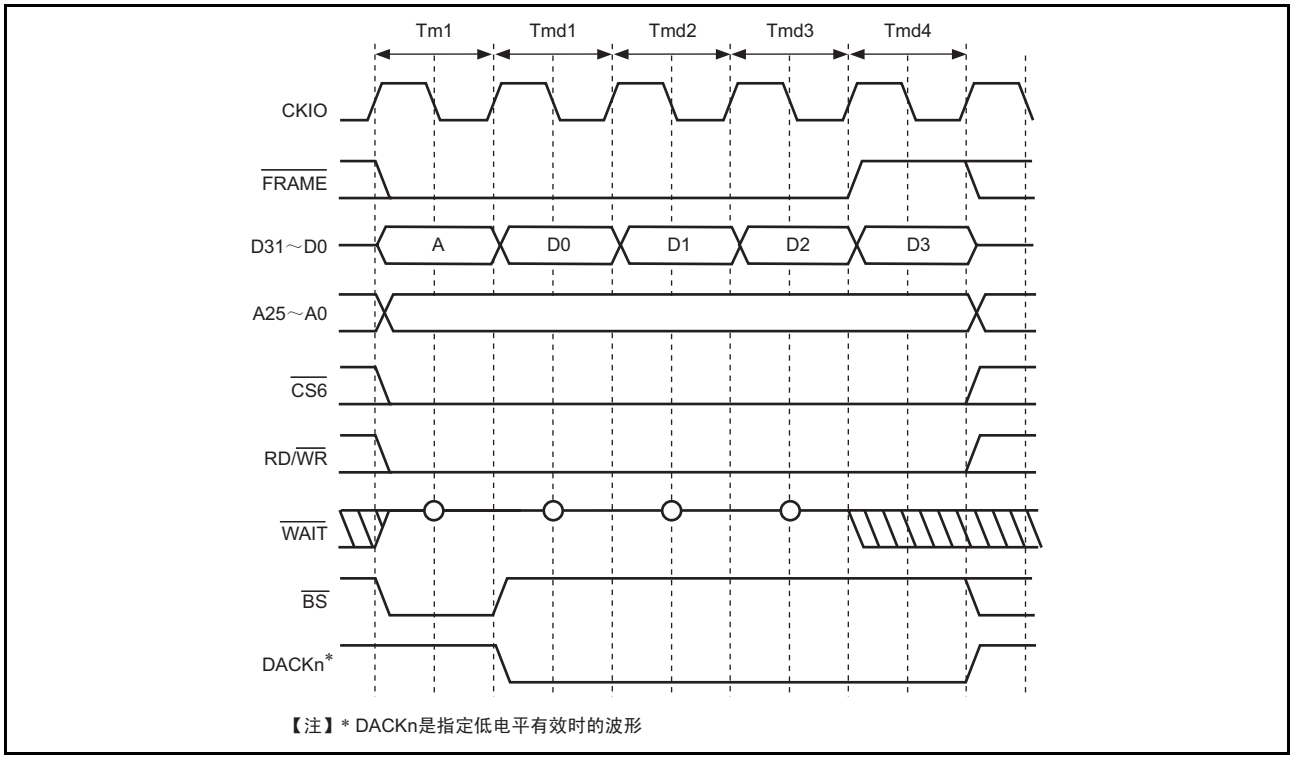


图 9.51 突发 MPX 空间的存取时序 （突发写、无等待、 CS6WCR.MPXMD=0）

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。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] 位所设定个数的等待周期。在第 2 次以后的存取周期中插入 CS0WCR 的 BW[1:0] 位所设定个数的等待周期。

当存取突发 ROM（时钟同步）时，$\overline{BS}$ 信号只对最初的存取周期断言。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，请将突发长度设定为 8，当总线宽度为 32 位时，请将突发长度设定为 4。不支持 8 位总线。本接口对全部读存取进行突发运行。例如，以 16 位总线宽度进行长字存取时，在读取 2 个所需的数据后，空读剩余的 6 个无用数据。

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，所以必须有效利用在高速缓存有效空间中的高速缓存充填引起 16 字节读操作，或 DMA 引起的 16 字节读操作。在写存取时，与通常空间存取时序相同。

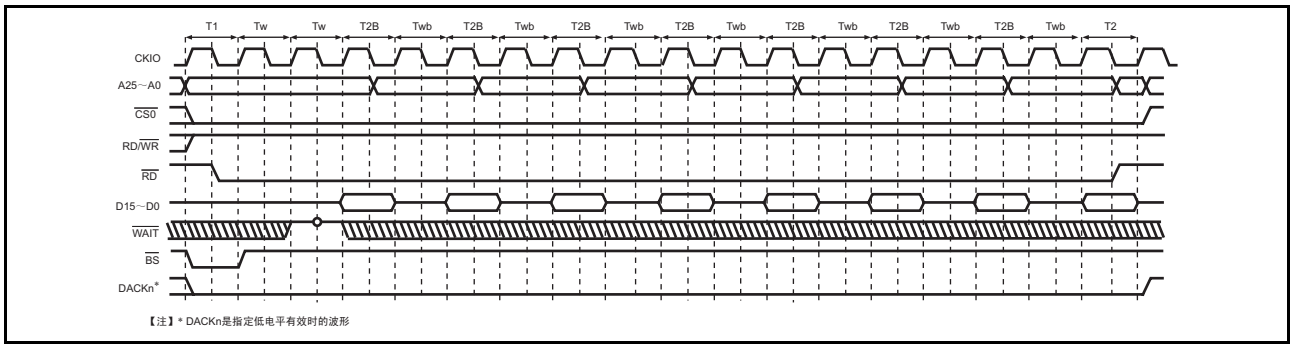


图 9.52　突发 ROM （时钟同步）的存取
（突发长度 8、初次存取等待 2、第 2 次及以后的存取等待 1）

9.5.12　存取周期之间的空闲

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空闲 （等待），以避免数据的冲突。

通过 CSnWCR 的 WM 位以及 CSnBCR 的 IWW[2:0] 位、 IWRWD[2:0] 位、 IWRWS[2:0] 位、 IWRRD[2:0] 位、IWRRS[2:0] 位和 CMNCR 的 DMAIW[2:0] 位、 DMAIWA 位指定存取周期之间的空闲周期数。可在存取周期之间插入空闲周期的项目如下：

1. 连续存取为写－读、写－写
2. 连续存取为不同空间的读－写
3. 连续存取为相同空间的读－写
4. 连续存取为不同空间的读－读
5. 连续存取为相同空间的读－读
6. 在通过 DMA 单地址传送，从外部器件输出数据的周期后，从其他器件 （包括本 LSI）输出数据时 （DMAIWA=0）
7. 在通过DMA单地址传送，从外部器件输出数据的周期后，发生存取时 （DMAIWA=1）

有关上述存取周期之间空闲周期数的指定，请参照各寄存器的说明。

除存取周期 （由这些寄存器指定）之间的等待空闲周期以外，为确保连接内部总线的接口或多路复用引脚（$\overline{\text{WEn}}$）的最小脉冲宽度，也有可能插入空闲周期。以下是关于空闲周期详细情况和空闲周期数计算方法的说明。

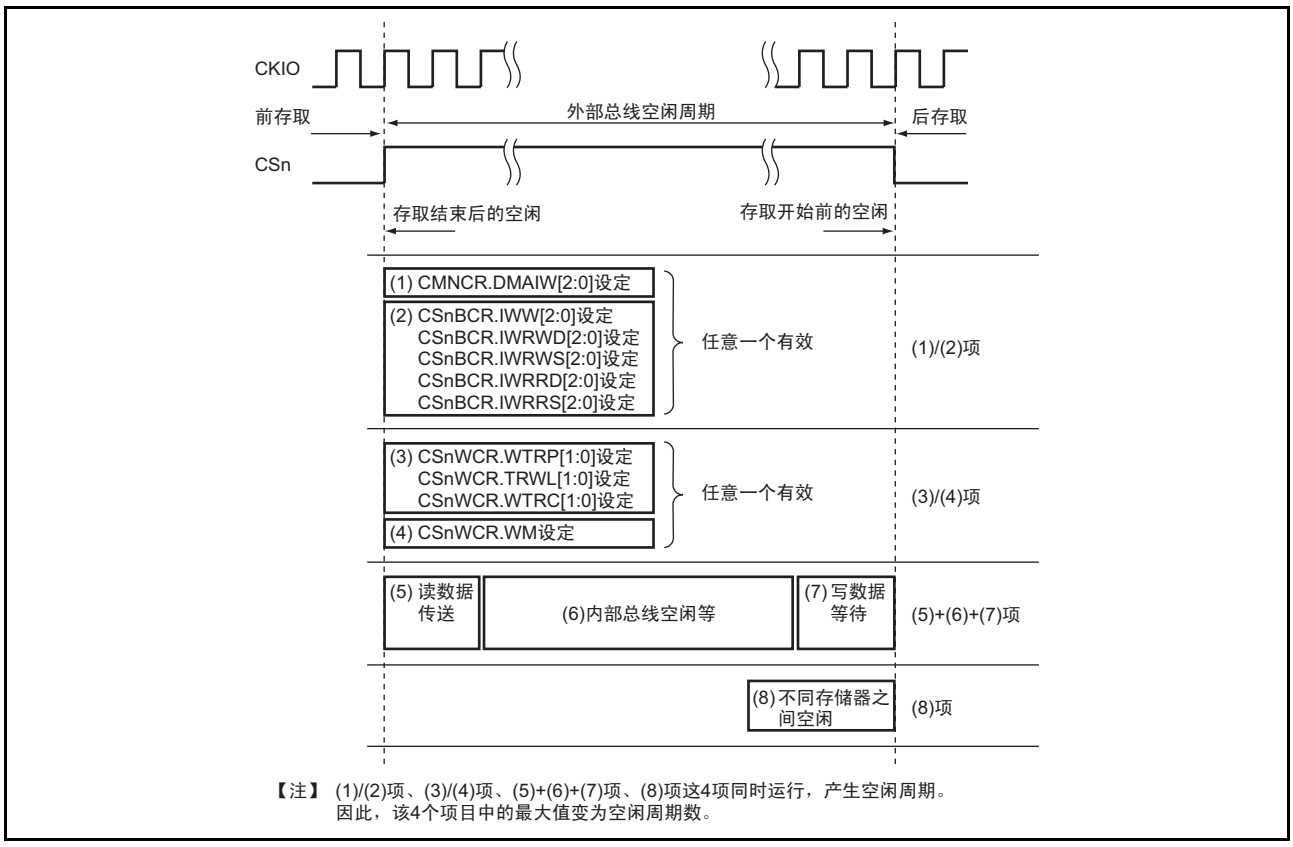
说明从$\overline{\text{CSn}}$取反到$\overline{\text{CSn}}$或$\overline{\text{CSm}}$断言的外部总线空闲周期数。此时$\overline{\text{CSn}}$和$\overline{\text{CSm}}$中含PCMCIA的$\overline{\text{CE2A}}$、$\overline{\text{CE2B}}$。决定外部总线空闲周期数的项目有 8 项，如**表 9.21** 所示。它们之间的关系如**图 9.53** 所示。

表 9.21　决定空闲周期数的项目

编号	内容	说明	范围	注意事项
(1)	CMNCR.DMAIW[2:0] 设定	指定 DMA 单地址传送时的空闲周期数。该项目只在单地址传送时有效，为存取结束后产生的空闲周期。	0 ～ 12	请注意，空闲数设定为 0 时，有 DACK 信号连续断言的情况，当带 DACK 的器件的识别周期数与 DMAC 传送数不匹配时，可导致误动作。
(2)	CSnBCR.IW***[2:0] 设定	指定不是单地址传送时的空闲周期数。可给每个前后周期的组合独立指定空闲周期数。例如在读取 CS1 空间后读取其他 CS 空间时，如果想将空闲数设定为至少 6 个周期，就将 CS1BCR.IWRRD[2:0] 设定为 B'100。该项目只在不为单地址传送时有效，为存取结束后产生的空闲周期。	0 ～ 12	请注意，为无法连续存取的存储器类型时，不能设定为 0。
(3)	CSnWCR 的 SDRAM 相关设定	指定 SDRAM 存取时的预充电结束 / 启动等待、命令之间的空闲数。该项目只在 SDRAM 存取时有效，为存取结束后产生的空闲周期。	0 ～ 3	请根据使用的 SDRAM 的规格设定。
(4)	CSnWCR.WM 位设定	在 SDRAM 以外的存储器中，可将外部 $\overline{\text{WAIT}}$ 引脚输入设定为有效 / 无效。为 0 （外部 $\overline{\text{WAIT}}$ 有效）时，在存取结束后为了评价外部 $\overline{\text{WAIT}}$ 引脚状态插入 1 个空闲周期。为 1 （无效）时，不产生本空闲周期。	0 ～ 1	

编号	内容	说明	范围	注意事项
(5)	读数据传送周期	为读存取结束后产生的 1 个空闲周期。在被分割存取的最初和中途不产生空闲周期。另外， CSnWCR.HW[1:0] 不是 B'00 时，也不产生。	0 ～ 1	在 SDRAM、 PCMCIA 的读周期中肯定会产生 1 个空闲周期。
(6)	内部总线空闲等	通过内部总线，传递来自 CPU、 DMAC 等的外部总线存取请求和结果。在内部总线的空闲周期以及外部总线以外的存取中，外部总线成为空闲状态。存取长度超过外部数据总线宽度时，通过 BSC 进行分割存取，但在分割周期之间，没有内部总线空闲周期等的影响。	0 ～	Iϕ：根据 Bϕ 的时钟比，内部总线空闲数有时不为“0”。时钟比与内部总线的最小空闲数的关系如表 9.22、表 9.23 所示。
(7)	写数据到达等待周期	在写存取时，等待写数据到达后产生外部总线的写周期。该写数据等待为写周期前产生的空闲周期。但在上一个周期为写周期时，如果内部总线空闲周期比上一个存取的写周期短时，为了与上一个存取并行处理，就不产生本空闲周期 （写缓冲器效果）。	0 ～ 1	在写→写以及写→读存取时，由于左边所述的写缓冲器效果，容易产生连续存取。不能连续存取时，请通过 CSnBCR 等指定周期之间空闲的最低数。
(8)	不同存储器之间的空闲	为确保多路引脚的最小脉冲宽度，在发生切换存储器类型后的存取前，有时会插入空闲周期。根据存储器类型，即使不切换存储器类型，也会产生空闲周期。	0 ～ 2.5	由每个存储器类型决定。请参照表 9.24。

由于 (1)/(2) 项 （任意一项为有效）、(3)/(4) 项 （任意一项为有效）、(5)+(6)+(7) 项 （由于按顺序发生，所以进行加法运算）和 (8) 这四项同时发生，其中最大的一项成为外部总线空闲数。确保最低空闲数时，请进行 (1)/(2) 项寄存器的设定。



【注】 (1)/(2)项、(3)/(4)项、(5)+(6)+(7)项、(8)项这4项同时运行，产生空闲周期。
因此，该4个项目中的最大值变为空闲周期数。

图 9.53　空闲周期的结构

表 9.22　内部总线的最小空闲数 （CPU 运行）

CPU 运行	时钟比 （Iϕ：Bϕ）					
	8:1	6:1	4:1	3:1	2:1	1:1
写 → 写	1	1	2	2	2	3
写 → 读	0	0	0	0	0	1
读 → 写	1	1	2	2	2	3
读 → 读	0	0	0	0	0	1

表 9.23　内部总线的最小空闲数 （DMAC 运行）

DMAC 运行	传送模式	
	双地址	单地址
写 → 写	0	2
写 → 读	0 或 2	0
读 → 写	0	0
读 → 读	0	2

【注】 1. 双地址传送的写 → 写、读 → 读的运行在被分割的周期中进行。

2. 双地址传送的写 → 读的 0 为连续启动不同的通道时， 2 为连续启动相同通道时。

3. 单次地址的写 → 读、读 → 写为连续启动不同的通道时。
“写”是从带DACK的器件→外部存储器的传送、“读”是从外部存储器→带DACK的器件的传送。

表 9.24　进行不同存储器之间的存取之前插入的空闲周期数

		后周期									
		SRAM	突发 ROM （异步）	MPX-I/O	字节 SRAM （BAS=0）	字节 SRAM （BAS=1）	SDRAM	SDRAM （低频率模式）	PCMCIA	突发 MPX	突发 ROM （同步）
前周期	SRAM	0	0	1	0	1	1	1.5	0	0	0
	突发 ROM （异步）	0	0	1	0	1	1	1.5	0	0	0
	MPX-I/O	1	1	0	1	1	1	1.5	1	1	1
	字节 SRAM （BAS=0）	0	0	1	0	1	1	1.5	0	0	0
	字节 SRAM （BAS=1）	1	1	2	1	0	0	1.5	1	1	1
	SDRAM	1	1	2	1	0	0	—	1	1	1
	SDRAM （低频率模式）	1.5	1.5	2.5	1.5	0.5	—	1	1.5	1.5	1.5
	PCMCIA	0	0	1	0	1	1	1.5	0	0	0
	突发 MPX	0	0	1	0	1	1	1.5	0	0	0
	突发 ROM （同步）	0	0	1	0	1	1	1.5	0	0	0

估算周期之间最低空闲数的例子如**图 9.54** 所示。实际运行中，空闲周期可能比估算值小（根据写缓冲器的效果），或者比估算值大（由于执行 CPU 指令或 CPU 寄存器竞争引起的拆分而导致产生内部总线空闲周期），所以在使用估算值时，必须考虑这些误差。

周期之间空闲数的估算例子

考虑通过CPU存取，将数据从CS1空间传送到CS2空间的例子。传送为重复CS1读→CS1读→CS2写→CS2写→CS1读→…。

- 条件
  将CS1BCR和CS2BCR的周期之间空闲指定为0。
  CS1WCR和CS2WCR的WM位置1（外部WAIT引脚无效）、HW[1:0]置00（不延长CS取反）。
  Iϕ:Bϕ设置为4:1，传送时不执行其他处理。
  CS1和CS2都可连接通常SRAM，总线宽度为32位，存取长度也为32位。

在各周期间估算决定空闲数的项目。下表的R为读、W为写。

项目	R→R	R→W	W→W	W→R	备　注
(1)/(2)	0	0	0	0	CSnBCR置0
(3)/(4)	0	0	0	0	WM位置1
(5)	1	1	0	0	在读周期后发生
(6)	0	2	2	0	参照表8.19的Iϕ:Bϕ＝4:1部分
(7)	0	1	0	0	因为写缓冲器效果，不发生第2次的空闲周期
(5)+(6)+(7)	1	4	2	0	
(8)	0	0	0	0	通过SRAM→SRAM得到的结果
估算空闲周期数	1	4	2	0	(1)/(2)项、(3)/(4)项、(5)+(6)+(7)项、(8)项中的最大值
实际产生的空闲数	1	4	2	1	在W→R中产生不同周期数的原因是，将(6)的内部空闲数估算为0，但实际为执行循环判断指令，就产生了内部空闲。

图 9.54　空闲周期数的估算例子与实际的比较

9.5.13　总线仲裁

本 LSI 中的总线仲裁在通常状态下占有总线权，在接受来自其它器件的总线权请求后释放总线。

在总线周期的边界进行总线权的传递。在有总线权请求时，如果没有执行总线周期，就立即释放总线权。如果正在执行总线周期，就等到总线周期结束，再释放总线权。即使从 LSI 外部观察没有执行总线周期，但因为有在存取周期之间插入等待等而开始内部总线周期的情况，所以根据 $\overline{\text{CSn}}$ 信号及其他的总线控制信号，不能立刻判断是否释放总线。不进行总线释放的状态如下所示：

1. 由高速缓存未中引起的16字节传送中
2. 回写高速缓存存储器时
3. 在TAS指令的读周期和写周期之间
4. 由数据总线宽度小于存取长度而产生的多个总线周期 （例如：在 8 位数据总线宽度的存储器中，进行长字存取时的总线周期之间）
5. 通过DMAC传送16字节
6. 将CMNCR的BLOCK位置1时
7. 传送LCDC的16字节～128字节时

而且，通过 CMNCR 的 DPRTY[1:0] 位，可在 DMAC 突发传送时选择是否接受总线权使用请求。

本 LSI 只要不接受总线权请求，就占有总线权。接受外部的总线权请求 $\overline{\text{BREQ}}$ 断言 （低电平），执行完总线周期后释放总线，并使总线使用允许 $\overline{\text{BACK}}$ 为断言（低电平）。接受表示外部器件已释放总线的 $\overline{\text{BREQ}}$ 取反（高电平），使 $\overline{\text{BACK}}$ 为取反后 （高电平），重新使用总线。

使用 SDRAM 接口时，如果有激活的存储体，就发送所有存储体预充电命令 （PALL），结束后释放总线。

具体的总线释放顺序如下所示。首先，与 CKIO 的上升沿同步将地址总线和数据总线设置为高阻抗。0.5 个周期后使总线使用允许信号与 CKIO 的下降沿同步断言。接着在 CKIO 的上升沿，将总线控制信号（$\overline{\text{BS}}$、$\overline{\text{CSn}}$、$\overline{\text{RASU}}$、$\overline{\text{RASL}}$、$\overline{\text{CASU}}$、$\overline{\text{CASL}}$、CKE、DQMxx、$\overline{\text{WEn}}$、$\overline{\text{RD}}$ 和 RD/$\overline{\text{WR}}$）设置为高阻抗。这些总线控制信号，最迟在设置为高阻抗的 1 个周期前要成为高电平。在 CKIO 的下降沿进行总线权请求信号的采样。而且，CKE、$\overline{\text{RASU}}$、$\overline{\text{RASL}}$、$\overline{\text{CASU}}$、$\overline{\text{CASL}}$ 通过 CMNCR 的 HIZCNT 位的设定，即使在总线释放过程中，也可用之前的值继续驱动。

从外部器件再次获得总线权的顺序如下所示。在 CKIO 的下降沿检测出 $\overline{\text{BREQ}}$ 取反时，1.5 个周期后，开始以高电平驱动总线控制信号。接着在时钟的下降沿，总线使用允许信号取反。之后在 CKIO 的上升沿开始地址总线和数据总线的驱动。最早从与驱动地址及数据信号相同时钟的上升沿开始将总线控制信号设置为断言，并实际开始总线周期。总线仲裁时序如**图 9.55** 所示。

当在释放总线权的过程中需要刷新 SDRAM 时，就必须返回总线权，将 $\overline{\text{REFOUT}}$ 设定为该请求信号。关于 $\overline{\text{REFOUT}}$ 引脚的选择请参照**“第 29 章　引脚功能控制器（PFC）”**。$\overline{\text{REFOUT}}$ 在获得总线权之前一直在低电平保持断言。外部器件通过 $\overline{\text{REFOUT}}$ 断言，使 $\overline{\text{BREQ}}$ 为取反并返回总线权。请注意，如果超过刷新间隔的规定时间后还没返回总线权时，就不能执行刷新，因此无法确保 SDRAM 的内容。

到本 LSI 获得总线权为止，不执行释放总线权时的 SLEEP 指令（转移至睡眠模式、深度待机模式或软件待机模式）。释放总线权时的手动复位也同样。

软件待机模式或深度待机模式时，忽视 $\overline{\text{BREQ}}$ 输入、$\overline{\text{BACK}}$ 输出变为高阻抗状态。该状态中需要总线权请求时，通过将 $\overline{\text{BACK}}$ 引脚设置为下拉，在转移至软件待机模式或深度待机模式时，成为总线权释放状态。

请在允许使用总线后 （$\overline{\text{BACK}}$ 的低电平断言），进行总线权请求 （$\overline{\text{BREQ}}$ 的低电平断言）后的总线权释放（$\overline{\text{BREQ}}$ 的高电平取反）。在 $\overline{\text{BACK}}$ 断言前设置 $\overline{\text{BREQ}}$ 为取反时，根据 $\overline{\text{BREQ}}$ 取反时序， $\overline{\text{BACK}}$ 只有 1 个周期为断言，因此可能在外部器件和本 LSI 之间产生总线冲突。

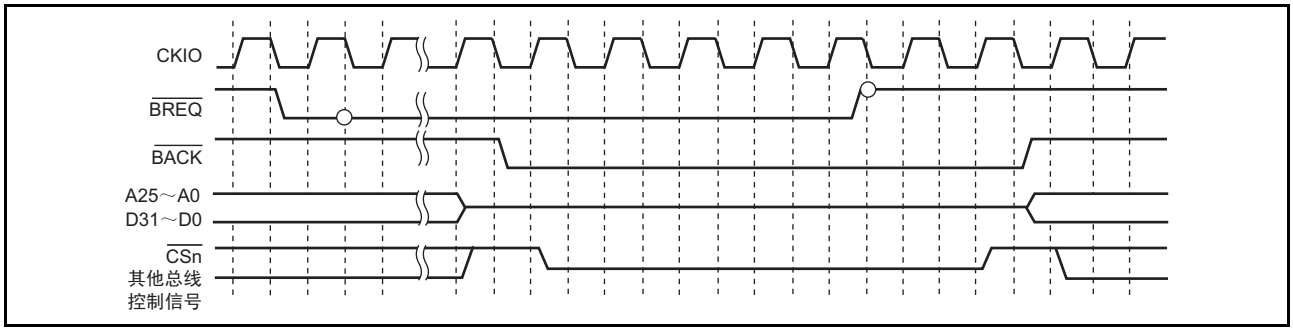


图 9.55　总线仲裁 （时钟模式 7）

9.5.14　其他

（1） 复位

只在上电复位时才对总线状态控制器（BSC）进行完全初始化。在上电复位时，使复位与内部时钟同步后，与总线周期状态 （是否在执行中）无关，将全部信号设为取反，并关闭输出缓冲器。另外对全部控制寄存器进行初始化。在软件待机模式、睡眠和手动复位时，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，就只执行当前正在执行的总线周期。在手动复位信号断言时，RTCNT 也能递增计数，所以能产生刷新请求并启动刷新周期。

（2） 从 LSI 内部总线主控器的存取

本 LSI 内部分为 CPU 总线、内部总线和外围总线共 3 种总线。 CPU 和高速缓存存储器连接 CPU 总线、除 CPU 外的内部总线主控器和总线状态控制器连接内部总线、低速的外围模块连接外围总线。另外，高速缓存存储器以外的内部存储器连接 CPU 总线和内部总线。可从 CPU 总线存取内部总线，但不能反相进行存取。因此发生以下的情况。

可从除 CPU 之外的内置总线主控器 （DMAC 等）存取高速缓存存储器以外的内部存储器，但不能存取高速缓存存储器。通过除 CPU 之外的内置总线主控器进行外部存储器的写操作，但结果有可能发生外部存储器的内容和高速缓存的内容不符的情况。通过除 CPU 之外的内部总线主控器进行外部存储器的写操作时，传送到该地址的数据可能在高速缓存时，需要以软件清除高速缓存存储器。

在高速缓存有效空间时，如果 CPU 开始读存取，就搜索高速缓存。数据保留在高速缓存中时，CPU 输入数据并完成存取。高速缓存中没有数据时，通过内部总线进行高速缓存数据的充填，所以启动 4 个连续长字读取。关于字节或字操作数存取时以及转移到奇数字边界 （4n+2）时的命中与未中，在芯片外部接口上必须通过 4 个长字存取进行填充。在高速缓存无效空间时，根据实际的存取地址进行存取。存取为对偶数字边界 （4n）的取指令时为长字存取，为对奇数字边界 （4n+2）的取指令时为字存取。

内置外围模块的读周期时，通过内部总线和外围总线启动读周期。读出的数据经过外围总线、内部总线、CPU 总线传送到 CPU。

高速缓存有效空间的写周期根据高速缓存的写方式，有不同的运行。

回写模式中，有对应地址 （搜索高速缓存的地址）的数据时，就对高速缓存进行写入。到发生对应地址的置换为止，不对存取器进行实际写入。没有对应地址的数据时，更新高速缓存。首先，置换对象的数据保存到内部缓冲器，之后读出包含对应地址数据的 16 字节数据，更新对应地址的数据。接着，进行最初保存的 16 字节数据的回写周期。

直写模式中，当有对应地址 （搜索高速缓存的地址）的数据时，就在向高速缓存进行写入的同时，通过内部总线进行实际写入。没有对应地址的数据时，不更新高速缓存，而只通过内部总线进行实际写入。

由于总线状态控制器里有一段写缓冲器，因此即使在写周期中芯片外部的总线周期没有结束，也可用内部总线进行其他存取。对芯片外部的低速存储器进行写入后，在读取或写入内置外围模块时，可以不等待至低速存储器的写入结束，就存取内置外围模块。

因为在读周期时，CPU 总是处于等待状态直到读操作结束，所以想要在确认对实际器件的数据写入已经结束后继续处理时，只需继续对相同地址进行虚读就能确认写入已经结束。

对于 DMAC 等其他总线主控器的存取，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，不必等待至写周期结束，就可以启动下一个读周期。但 DMA 的源地址和目标地址都在外部存储器空间时，就需等到前一个写周期结束后才开始下一个读周期。

如果在写缓冲器运行时更改 BSC 的寄存器就不能正确进行写存取，所以请勿在写存取后立即更改 BSC 的寄存器。如果需要，请在进行写数据的虚读后更改 BSC 的寄存器。

（3） 内置外围模块的存取

在存取内置外围模块的寄存器时，内部总线至少需要 2 个周期以上的外围模块时钟 （Pϕ）。另一方面，从 CPU 向内部外围模块进行写入时， CPU 不等待寄存器写操作结束，就执行之后的指令。

以为了降低功耗转移至软件待机模式为例进行说明。为实现转移，需要在 STBCR 寄存器的 STBY 位置 1 后执行 SLEEP 指令，但在执行 SLEEP 指令前必须虚读 STBCR 寄存器。如果不虚读，在 STBY 位置 1 前，由于 CPU 执行 SLEEP 指令，就不转移到软件待机模式，而转移至睡眠模式。为等待 STBY 位的写入，需要虚读 STBCR 寄存器。

如此例，执行后续指令时，想要通过内部外围寄存器反映所产生的变化时，需要在寄存器写指令后实施相同寄存器的虚读，然后再执行该后续指令。

第 10 章　直接存储器存取控制器 （DMAC）

直接存储器存取控制器 （DMAC）能代替 CPU 在带 DACK （传送请求的接收信号）的外部器件、外部存储器、内置存储器、存储器映像的外部器件以及内置外围模块之间进行高速数据传送。

10.1　特点

- 通道数：8个通道 （CH0～CH7）
  只有4个通道 （CH0～CH3）能接受外部请求。
- 地址空间：在体系结构上为4G字节
- 传送数据单位：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 （24位）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3种 （外部请求、内置外围模块请求、自动请求）
  以下模块能发送内置外围模块请求：
  SCIF：8个源、 IIC3：8个源、 A/D转换器：1个源、 MTU2：5个源、 CMT：2个源、
  USB：2个源、 FLCTL：2个源、 RCAN-TL1：2个源、 SSI：4个源、 SRC：2个源、
  SSU：4个源、 ROM-DEC：1个源、 SDHI：2个源
- 总线模式：可选择周期挪用模式 （普通模式与间歇模式）及突发模式
- 优先顺序：可选择通道优先顺序的固定模式或者循环模式
- 中断请求：能在数据传送1/2结束或者数据传送结束时向CPU发生中断请求
  DMAC传送通过CHCR的HE位与HIE位，在初始设定的1/2次结束时设定对CPU的中断。
- 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・传送结束信号：DACK及TEND能设定有效电平
- 由于DMA设定的寄存器具备再装入功能，因此能够不重新设定而反复执行与执行中的DMA传送相同设定的DMA传送。通过在DMA传送中预先设定再装入寄存器，能用不同的设定执行下次的DMA传送。
  该再装入功能可对每个通道以及再装入寄存器设定ON/OFF。

DMAC 的框图如图 10.1 所示。

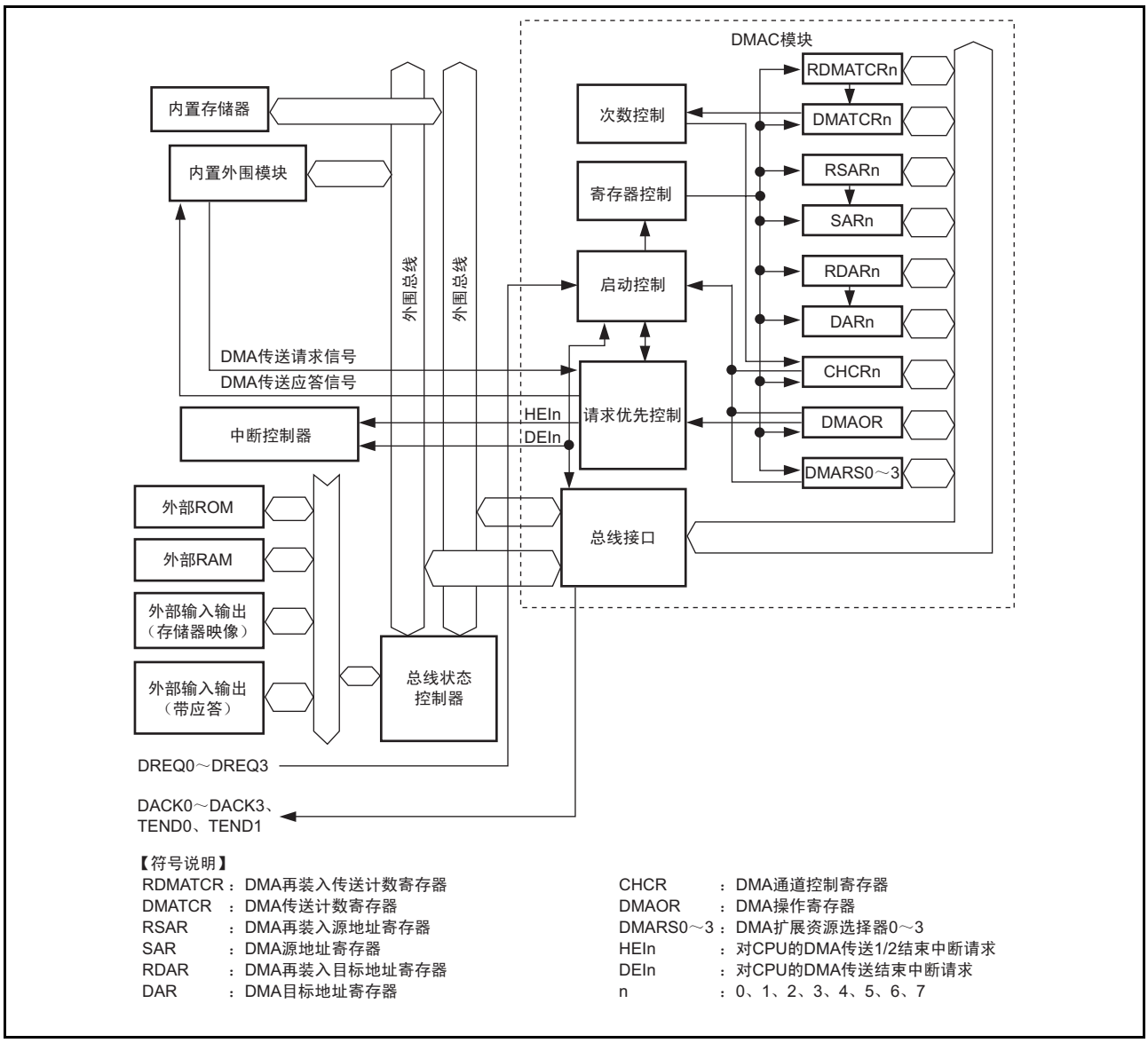


图 10.1　DMAC 的框图

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 （CH0 ～ CH3）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
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
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
0	DMA 传送结束	TEND0	输出	输出 DMAC 通道 0 的 DMA 传送结束
1	DMA 传送结束	TEND1	输出	输出 DMAC 通道 1 的 DMA 传送结束

10.3　寄存器说明

DMAC 有以下寄存器。各通道有 4 个控制寄存器和 3 个再装入寄存器，全部通道通用的控制寄存器有 1 个。并且每 2 通道有 1 个扩展资源选择器寄存器。关于各通道的寄存器，例如通道 0 的 SAR 表示为 SAR_0。

表 10.2　寄存器构成

通道	寄存器名	简称	R/W	初始值	地址	存取长度
0	DMA 源地址寄存器 _0	SAR0	R/W	H'00000000	H'FFFE1000	16、32
	DMA 目标地址寄存器 _0	DAR0	R/W	H'00000000	H'FFFE1004	16、32
	DMA 传送计数寄存器 _0	DMATCR0	R/W	H'00000000	H'FFFE1008	16、32
	DMA 通道控制寄存器 _0	CHCR0	R/W*1	H'00000000	H'FFFE100C	8、16、32
	DMA 再装入源地址寄存器 _0	RSAR0	R/W	H'00000000	H'FFFE1100	16、32
	DMA 再装入目标地址寄存器 _0	RDAR0	R/W	H'00000000	H'FFFE1104	16、32
	DMA 再装入传送计数寄存器 _0	RDMATCR0	R/W	H'00000000	H'FFFE1108	16、32
1	DMA 源地址寄存器 _1	SAR1	R/W	H'00000000	H'FFFE1010	16、32
	DMA 目标地址寄存器 _1	DAR1	R/W	H'00000000	H'FFFE1014	16、32
	DMA 传送计数寄存器 _1	DMATCR1	R/W	H'00000000	H'FFFE1018	16、32
	DMA 通道控制寄存器 _1	CHCR1	R/W*1	H'00000000	H'FFFE101C	8、16、32
	DMA 再装入源地址寄存器 _1	RSAR1	R/W	H'00000000	H'FFFE1110	16、32
	DMA 再装入目标地址寄存器 _1	RDAR1	R/W	H'00000000	H'FFFE1114	16、32
	DMA 再装入传送计数寄存器 _1	RDMATCR1	R/W	H'00000000	H'FFFE1118	16、32
2	DMA 源地址寄存器 _2	SAR2	R/W	H'00000000	H'FFFE1020	16、32
	DMA 目标地址寄存器 _2	DAR2	R/W	H'00000000	H'FFFE1024	16、32
	DMA 传送计数寄存器 _2	DMATCR2	R/W	H'00000000	H'FFFE1028	16、32
	DMA 通道控制寄存器 _2	CHCR2	R/W*1	H'00000000	H'FFFE102C	8、16、32
	DMA 再装入源地址寄存器 _2	RSAR2	R/W	H'00000000	H'FFFE1120	16、32
	DMA 再装入目标地址寄存器 _2	RDAR2	R/W	H'00000000	H'FFFE1124	16、32
	DMA 再装入传送计数寄存器 _2	RDMATCR2	R/W	H'00000000	H'FFFE1128	16、32
3	DMA 源地址寄存器 _3	SAR3	R/W	H'00000000	H'FFFE1030	16、32
	DMA 目标地址寄存器 _3	DAR3	R/W	H'00000000	H'FFFE1034	16、32
	DMA 传送计数寄存器 _3	DMATCR3	R/W	H'00000000	H'FFFE1038	16、32
	DMA 通道控制寄存器 _3	CHCR3	R/W*1	H'00000000	H'FFFE103C	8、16、32
	DMA 再装入源地址寄存器 _3	RSAR3	R/W	H'00000000	H'FFFE1130	16、32
	DMA 再装入目标地址寄存器 _3	RDAR3	R/W	H'00000000	H'FFFE1134	16、32
	DMA 再装入传送计数寄存器 _3	RDMATCR3	R/W	H'00000000	H'FFFE1138	16、32

通道	寄存器名	简称	R/W	初始值	地址	存取长度
4	DMA 源地址寄存器 _4	SAR4	R/W	H'00000000	H'FFFE1040	16、32
	DMA 目标地址寄存器 _4	DAR4	R/W	H'00000000	H'FFFE1044	16、32
	DMA 传送计数寄存器 _4	DMATCR4	R/W	H'00000000	H'FFFE1048	16、32
	DMA 通道控制寄存器 _4	CHCR4	R/W*1	H'00000000	H'FFFE104C	8、16、32
	DMA 再装入源地址寄存器 _4	RSAR4	R/W	H'00000000	H'FFFE1140	16、32
	DMA 再装入目标地址寄存器 _4	RDAR4	R/W	H'00000000	H'FFFE1144	16、32
	DMA 再装入传送计数寄存器 _4	RDMATCR4	R/W	H'00000000	H'FFFE1148	16、32
5	DMA 源地址寄存器 _5	SAR5	R/W	H'00000000	H'FFFE1050	16、32
	DMA 目标地址寄存器 _5	DAR5	R/W	H'00000000	H'FFFE1054	16、32
	DMA 传送计数寄存器 _5	DMATCR5	R/W	H'00000000	H'FFFE1058	16、32
	DMA 通道控制寄存器 _5	CHCR5	R/W*1	H'00000000	H'FFFE105C	8、16、32
	DMA 再装入源地址寄存器 _5	RSAR5	R/W	H'00000000	H'FFFE1150	16、32
	DMA 再装入目标地址寄存器 _5	RDAR5	R/W	H'00000000	H'FFFE1154	16、32
	DMA 再装入传送计数寄存器 _5	RDMATCR5	R/W	H'00000000	H'FFFE1158	16、32
6	DMA 源地址寄存器 _6	SAR6	R/W	H'00000000	H'FFFE1060	16、32
	DMA 目标地址寄存器 _6	DAR6	R/W	H'00000000	H'FFFE1064	16、32
	DMA 传送计数寄存器 _6	DMATCR6	R/W	H'00000000	H'FFFE1068	16、32
	DMA 通道控制寄存器 _6	CHCR6	R/W*1	H'00000000	H'FFFE106C	8、16、32
	DMA 再装入源地址寄存器 _6	RSAR6	R/W	H'00000000	H'FFFE1160	16、32
	DMA 再装入目标地址寄存器 _6	RDAR6	R/W	H'00000000	H'FFFE1164	16、32
	DMA 再装入传送计数寄存器 _6	RDMATCR6	R/W	H'00000000	H'FFFE1168	16、32
7	DMA 源地址寄存器 _7	SAR7	R/W	H'00000000	H'FFFE1070	16、32
	DMA 目标地址寄存器 _7	DAR7	R/W	H'00000000	H'FFFE1074	16、32
	DMA 传送计数寄存器 _7	DMATCR7	R/W	H'00000000	H'FFFE1078	16、32
	DMA 通道控制寄存器 _7	CHCR7	R/W*1	H'00000000	H'FFFE107C	8、16、32
	DMA 再装入源地址寄存器 _7	RSAR7	R/W	H'00000000	H'FFFE1170	16、32
	DMA 再装入目标地址寄存器 _7	RDAR7	R/W	H'00000000	H'FFFE1174	16、32
	DMA 再装入传送计数寄存器 _7	RDMATCR7	R/W	H'00000000	H'FFFE1178	16、32
通用	DMA 操作寄存器	DMAOR	R/W*2	H'0000	H'FFFE1200	8、16
0/1	DMA 扩展资源选择器 0	DMARS0	R/W	H'0000	H'FFFE1300	16
2/3	DMA 扩展资源选择器 1	DMARS1	R/W	H'0000	H'FFFE1304	16
4/5	DMA 扩展资源选择器 2	DMARS2	R/W	H'0000	H'FFFE1308	16
6/7	DMA 扩展资源选择器 3	DMARS3	R/W	H'0000	H'FFFE130C	16

【注】 *1　为了清除标志， CHCRn 的 HE、 TE 位只能在读 1 后写 0。

*2　为了清除标志， DMAOR 的 AE、 NMIF 位只能在读 1 后写 0。

10.3.1　DMA 源地址寄存器 （SAR）

SAR 是 32 位可读写寄存器，指定 DMA 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。

进行字 （2 字节）、长字 （4 字节）、 16 字节单位的数据传送时，请分别指定 2 字节、 4 字节、 16 字节边界的地址。

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/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 （DAR）

DAR 是 32 位可读写寄存器，指定 DMA 传送目标地址，在 DMA 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。

进行字 （2 字节）、长字 （4 字节）、 16 字节单位的数据传送时，请分别指定 2 字节、 4 字节、 16 字节边界的地址。

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/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 （DMATCR）

DMATCR 是 32 位可读写寄存器，指定 DMA 的传送次数。当设定值为 H'00000001 时，传送次数为 1 次；当设定值为 H'00FFFFFF 时，传送次数为 16,777,215 次；当设定值为 H'00000000 时，传送次数为 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。

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 （CHCR）

CHCR 是 32 位可读写寄存器，控制 DMA 的传送模式。

关于决定外部引脚 DREQ、DACK 规格的位（DO、AM、AL、DL、DS），在通道 0 ～ 3 可读写，但是在通道 4 ～ 7 对应的位为保留位。

关于决定外部引脚 TEND 规格的位（TL），在通道 0、1 为可读写，但是在通道 2 ～ 7 对应的位为保留位。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TC	-	RLD SAR	RLD DAR	-	DAF	SAF	-	DO	TL	-	TE MASK	HE	HIE	AM	AL
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R	R/W	R/W	R	R/W	R/W	R	R/W	R/W	R	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
	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	TC	0	R/W	传送计数模式 设定 1 次传送请求是传送 1 次还是传送 DMATCR 的设定次数。本功能只在内置外围模块请求时有效。设定为 TC=0 时，TB 位不能设定为 1（突发模式）。传送请求源设定为 SCIF、 IIC3、 SSI、 SRC、 SDHI、 FLCTL 时，不能设定为 TC=1。 0：1 次传送请求传送 1 次 1：1 次传送请求传送 DMATCR 的设定次数
30	—	0	R	保留位 读写的值总是 0。
29	RLDSAR	0	R/W	SAR 再装入功能 ON/OFF 设定对 SAR 及 DMATCR 的再装入功能是有效 （ON）还是无效 （OFF）。 0：对 SAR 及 DMATCR 的再装入功能无效 （OFF） 1：对 SAR 及 DMATCR 的再装入功能有效 （ON）
28	RLDDAR	0	R/W	DAR 再装入功能 ON/OFF 设定对 DAR 及 DMATCR 的再装入功能是有效 （ON）还是无效 （OFF）。 0：对 DAR 及 DMATCR 的再装入功能无效 （OFF） 1：对 DAR 及 DMATCR 的再装入功能有效 （ON）
27	—	0	R	保留位 读写的值总是 0。
26	DAF	0	R/W	固定目标地址 16 字节传送 传送长度 （TS[1:0]）是 16 字节且目标地址模式 （DM[1:0]）是固定地址时有效 0：从设定在 DAR 寄存器的地址传送 16 字节。 1：从设定在 DAR 寄存器的地址传送 4 次 4 字节。本功能仅用于 ROM-DEC。
25	SAF	0	R/W	固定源地址 16 字节传送 传送长度 （TS[1:0]）是 16 字节且源地址模式 （SM[1:0]）是固定地址时有效。 0：从设定在 SAR 寄存器的地址传送 16 字节。 1：从设定在 SAR 寄存器的地址传送 4 次 4 字节。本功能仅用于 ROM-DEC。
24	—	0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说明
23	DO	0	R/W	DMA 溢出 选择是用溢出 0 检测 DREQ，还是用溢出 1 检测 DREQ。本位只在 CHCR_0 ～ 3 电平检测时有效。在 CHCR_4 ～ 7 是保留位，读写的值总是 0。 0：用溢出 0 检测 DREQ 1：用溢出 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。本位只在 CHCR_0、 1 有效。在 CHCR_2 ～ 7 是保留位，读写的值总是 0。 0：TEND 输出低电平有效 1：TEND 输出高电平有效
21	—	0	R	保留位 读写的值总是 0。
20	TEMASK	0	R/W	TE 设置屏蔽 指示 TE 位被置 1 时， DMA 传送不停止。通过 SAR 再装入功能或者与 DAR 再装入功能共同设定本位，可在到停止传送请求的这段时间内执行 DMA 传送。检测出自动请求及外部请求的上升 / 下降沿时，忽视本位的设定， TE 位被置位时，停止 DMA 传送。另外，本功能在 RLDSAR 位被置 1 时或者 RLDDAR 位被置 1 时有效。 0：TE 位被置位时，结束 DMA 传送 1：即使 TE 位被置位，也继续 DMA 传送
19	HE	0	R/(W)*	半结束标志 传送次数为传送开始前设置的 DMATCR 值一半以上时，HE 位被置 1。传送次数没达到传送开始前设置的 DMATCR 一半时，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 DMAOR 的 DME 位而结束传送， HE 位就不被置位。 HE 位被置位后，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 DMAOR 的 DME 位而结束传送， HE 位就保持置位状态。要清除 HE 位，需要在读出 HE 位 =1 后写 0。 0：DMA 传送中或者 DMA 传送中断后， DMATCR> （传送前设置的 DMATCR） /2 ［清除条件］ • 读出 HE 位 =1 后写 0 1：DMATCR ≤ （传送前设置的 DMATCR） /2
18	HIE	0	R/W	半结束中断允许 指定传送次数为传送开始前设置的 DMATCR 值一半时，是否向 CPU 请求中断。如果将 HIE 位置 1 并且 HE 位被置位，就向 CPU 请求中断。 0：DMATCR= （传送前设置的 DMATCR） /2 时，禁止中断请求 1：DMATCR= （传送前设置的 DMATCR） /2 时，允许中断请求
17	AM	0	R/W	应答模式 选择是在双地址模式的数据读周期输出 DACK 及 TEND，还是在双地址模式的写周期输出 DACK 及 TEND。 单地址模式时，与本位的指定无关，总是输出 DACK 及 TEND。 本位只在 CHCR_0 ～ 3 有效。在 CHCR_4 ～ 7 是保留位，读写的值总是 0。 0：在读周期输出 DACK 及 TEND （双地址模式） 1：在写周期输出 DACK 及 TEND （双地址模式）

位	位名	初始值	R/W	说明
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。本位只在 CHCR_0 ～ 3 有效。在 CHCR_4 ～ 7 是保留位，读写的值总是 0。 0：DACK 输出低电平有效 1：DACK 输出高电平有效
15、 14	DM[1:0]	00	R/W	目标地址模式 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：固定目标地址 01：增加目标地址 （以字节为单位传送时 +1，以字为单位传送时 +2，以长字为单位传送时 +4，以 16 字节为单位传送时 +16） 10：减少目标地址 （以字节为单位传送时 –1，以字为单位传送时 –2，以长字为单位传送时 –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 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：固定源地址 01：增加源地址 （以字节为单位传送时 +1，以字为单位传送时 +2，以长字为单位传送时 +4，以 16 字节为单位传送时 +16） 10：减少源地址 （以字节为单位传送时 –1，以字为单位传送时 –2，以长字为单位传送时 –4，以 16 字节为单位传送时禁止设定） 11：禁止设定
11 ～ 8	RS[3:0]	0000	R/W	资源选择 指定对 DMAC 输入的传送请求源。必须在 DMA 允许位 （DE）为 0 的状态下更改传送请求源。 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：禁止设定 0111：禁止设定 1000：DMA 扩展资源选择器 1001：RCAN-TL10 1010：RCAN-TL11 1011：禁止设定 1100：禁止设定 1101：禁止设定 1110：禁止设定 1111：禁止设定 【注】外部请求的指定只在 CHCR_0 ～ 3 有效。在 CHCR_4 ～ 7 即使指定外部请求也不执行任何操作。

位	位名	初始值	R/W	说明
7	DL	0	R/W	DREQ 电平
6	DS	0	R/W	DREQ 边沿选择 选择 DREQ 输入的检测方法和检测电平。 本位只在 CHCR_0 ～ 3 有效。在 CHCR_4 ～ 7 是保留位，读写的值必须总为 0。 另外，将传送请求源指定为内置外围模块或者自动请求时本位无效。 00：低电平检测 01：下降沿检测 10：高电平检测 11：上升沿检测
5	TB	0	R/W	传送总线模式 选择 DMA 传送的总线模式。但是，设定 TC=0 时，请勿选择突发模式。 0：周期挪用模式 1：突发模式
4、3	TS[1:0]	00	R/W	传送长度 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置外围模块的寄存器时，必须选择该传送长度。 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 （长字传送 ×4）
2	IE	0	R/W	中断允许 指定在 DMA 传送结束时是否向 CPU 请求中断。如果将 IE 位置 1 并且 TE 位被置位，就向 CPU 请求中断 （DEI）。 0：禁止中断请求 1：允许中断请求
1	TE	0	R/(W)*	传送结束标志 当 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 =1 后写 0。 TEMASK 位为 0，且 TE 位被置位时，即使将 DE 位置 1 也不允许传送。 0：DMA 传送中或者 DMA 传送中断 ［清除条件］ • 在读 TE 位 =1 后写 0 1：（DMATCR=0） DMA 传送结束
0	DE	0	R/W	DMA 允许 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 位置 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 传送请求时，就开始传送。在外部请求的低电平检测或者高电平检测及外围模块请求中，如果 TEMASK 位为 1，NMIF 位及 AE 位必须为 0。如果 TEMASK 位为 0， TE 位也必须为 0。外部请求的上升沿检测或者下降沿检测与自动请求模式相同，TE 位、NMIF 位、AE 位必须全部为 0。清除 DE 位时，能中断传送。 0：禁止 DMA 传送 1：允许 DMA 传送

【注】 * 为了清除标志，只能在读 1 后写 0。

10.3.5　DMA 再装入源地址寄存器 （RSAR）

RSAR 为 32 位可读 / 写寄存器。

将 SAR 再装入功能设定为 ON 的情况下，在当前 DMA 传送结束时， RSAR 的内容被写到源地址寄存器（SAR）。此时，因为在 DMA 传送中预先进行了设定，所以能预置下次 DMA 传送的设定。将 SAR 再装入功能设定为 OFF 的情况下，对运行没有任何影响。

以字 （2 字节）、长字 （4 字节）、 16 字节为单位进行数据传送时，请分别指定 2 字节、 4 字节、 16 字节边界的地址。

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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

10.3.6　DMA 再装入目标地址寄存器 （RDAR）

RDAR 为 32 位可读 / 写寄存器。

将 DAR 再装入功能设定为 ON 的情况下，在当前 DMA 传送结束时，RDAR 的内容被写到目标地址寄存器（DAR）。此时，因为在 DMA 传送中预先进行了设定，所以能预置下次 DMA 传送的设定。将 DAR 再装入功能设定为 OFF 的情况下，对运行没有任何影响。

以字 （2 字节）、长字 （4 字节）、 16 字节为单位进行数据传送时，请分别指定 2 字节、 4 字节、 16 字节边界的地址。

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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

10.3.7　DMA 再装入传送计数寄存器 （RDMATCR）

RDMATCR 为 32 位可读 / 写寄存器。

将 SAR 再装入功能或者 DAR 再装入功能设定为 ON 的情况下，在当前 DMA 传送结束时， RDMATCR 的内容被写到传送计数寄存器（DMATCR）。此时，因为在 DMA 传送中预先进行了设定，所以能预置下次 DMA 传送的设定。将 SAR 再装入功能或者 DAR 再装入功能设定为 OFF 的情况下，对运行没有任何影响。

RDMATCR 的高 8 位读写的值总是 0。

与 DMATCR 相同，当设定值为 H'00000001 时，传送次数为 1 次；当设定值为 H'00FFFFFF 时，传送次数为 16,777,215 次；当设定值为 H'00000000 时，传送次数为 16,777,216 次（最大传送次数）。16 字节传送时，传送 1 次 16 字节 （128 位）时，计数 1 次。

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

10.3.8　DMA 操作寄存器 （DMAOR）

DMAOR 为 16 位可读 / 写寄存器，指定 DMA 传送时通道的优先级。也表示 DMA 的传送状态 （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	—	均为 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 选择周期挪用模式时的普通模式或者间歇模式。 为了使间歇模式有效，所有通道的总线模式必须是周期挪用模式。 00：普通模式 01：禁止设定 10：间歇模式 16 Bϕ 时钟的每 16 个时钟 （周期）执行 1 次 DMA 传送 11：间歇模式 64 Bϕ 时钟的每 64 个时钟 （周期）执行 1 次 DMA 传送
11、 10	—	均为 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 如果多个通道同时产生传送请求，此位决定要执行的通道优先级。 00：固定模式 1：CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7 01：固定模式 2：CH0 > CH4 > CH1 > CH5 > CH2 > CH6 > CH3 > CH7 10：禁止设定 11：循环模式 （只支持 CH0 ～ CH3）
7 ～ 3	—	均为 0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说明
2	AE	0	R/(W)*	地址错误标志 表示产生由 DMAC 引起的地址错误。 AE 位被置位时，即使将 CHCR 的 DE 位和 DMAOR 的 DME 位置 1 也不允许 DMA 传送。为了清除 AE 位，必须在读 AE 位 =1 后写 0。 0：无由 DMAC 引起的地址错误 1：产生由 DMAC 引起的地址错误 ［清除条件］ • 读 AE 位 =1 后写 0
1	NMIF	0	R/(W)*	NMI 标志 表示产生 NMI 中断。 NMIF 位被置位时，即使将 CHCR 的 DE 位和 DMAOR 的 DME 位置 1 也不允许 DMA 传送。为了清除 NMIF 位，必须在读 NMIF 位为 1 后写 0。输入 NMI 时，能在 1 个传送单位完成执行中的 DMA 传送。 DMAC 不运行时，即使输入 NMI 中断， NMIF 位也被置 1。 0：无 NMI 中断 1：产生 NMI 中断 ［清除条件］ • 读 NMIF 位 =1 后写 0
0	DME	0	R/W	DMA 主控允许 允许或者禁止所有通道的 DMA 传送。 DME 位及 CHCR 的 DE 位被置 1 时，允许 DMA 传送。但是，进行传送的通道的 CHCR 的 TE 位和 DMAOR 的 NMIF 位、AE 位必须全部为 0。清除 DME 位时，中断所有通道的 DMA 传送。 0：禁止所有通道的 DMA 传送 1：允许所有通道的 DMA 传送

【注】 * 为了清除标志，只能在读 1 后写 0。

结束一个传送后，改变优先权模式位的设定时，初始化优先级。

例如，重新设定固定模式 2 时，优先级为 CH0 > CH4 > CH1 > CH5 > CH2 > CH6 > CH3 > CH7，重新设定固定模式 1 时，优先级为 CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7。重新设定循环模式时，传送结束通道被复位。

优先权模式位的各模式（模式 0 ～ 2）的优先级转移如**表 10.3** 所示。根据传送结束的通道，各模式中，接受下一个请求的通道的优先级最多变化为 3 种方式。

例如，传送结束的通道为 CH1 时，接受下一个请求的通道优先级为 CH2 > CH3 > CH0 > CH1 > CH4 > CH5 > CH6 > CH7。因为传送结束的通道为 CH4、 CH5、 CH6、 CH7 时为非循环对象，所以即使通道 4、通道 5、通道 6 以及通道 7 传送结束，传送优先级也不变化。

另外，发生地址错误时 DMAC 的内部处理运行如下：

- 不发生地址错误时：Read （传送源 →DMAC 内部） →Write （DMAC 内部 → 传送目标）
- 在源地址发生地址错误：Nop→Nop
- 在目标地址发生地址错误：Read→Nop

表 10.3　优先权模式位的组合

模式	传送结束	优先权模式位		传送结束后的优先级：高 ←→ 低							
	CH No.	PR[1]	PR[0]	优先级 0	优先级 1	优先级 2	优先级 3	优先级 4	优先级 5	优先级 6	优先级 7
模式 0 （固定模式 1）	任意	0	0	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
模式 1 （固定模式 2）	任意	0	1	CH0	CH4	CH1	CH5	CH2	CH6	CH3	CH7
模式 2 （循环模式）	CH0	1	1	CH1	CH2	CH3	CH0	CH4	CH5	CH6	CH7
	CH1	1	1	CH2	CH3	CH0	CH1	CH4	CH5	CH6	CH7
	CH2	1	1	CH3	CH0	CH1	CH2	CH4	CH5	CH6	CH7
	CH3	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
	CH4	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
	CH5	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
	CH6	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
	CH7	1	1	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7

10.3.9　DMA 扩展资源选择器 0 ～ 3 （DMARS0 ～ DMARS3）

DMARS 为 16 位可读 / 写寄存器，对每个通道指定外围模块的 DMA 传送请求源。设定 DMARS0 为通道 0 以及 1、DMARS1 为通道 2 以及 3、DMARS2 为通道 4 以及 5、DMARS3 为通道 6 以及 7。可设定的组合如表 10.4 所示。

本寄存器，能设定以下传送请求源以接受传送请求。

SCIF：8 个源、 IIC3：8 个源、 A/D 转换器：1 个源、 MTU2：5 个源、 CMT：2 个源、 USB：2 个源、 FLCTL：2 个源、 SSI：4 个源、 SRC：2 个源、 SSU：4 个源、 ROM-DEC：1 个源、 SDHI：2 个源

并且，只有 2 个 RCAN-TL1 源产生的传送请求接受信号，能用 DMA 通道控制寄存器的 RS[3:0] 设定，不需要用 DMA 扩展资源选择器设定。

- DMARS0

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH1 MID[5:0]						CH1 RID[1:0]		CH0 MID[5:0]						CH0 RID[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

- DMARS1

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH3 MID[5:0]						CH3 RID[1:0]		CH2 MID[5:0]						CH2 RID[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

- DMARS2

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH5 MID[5:0]						CH5 RID[1:0]		CH4 MID[5:0]						CH4 RID[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

- DMARS3

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CH7 MID[5:0]						CH7 RID[1:0]		CH6 MID[5:0]						CH6 RID[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

各模块的传送请求设定以下的 MID、 RID。

表 10.4　DMARS 的设定

外围模块	1 通道的设定值 （{MID、 RID}）	MID	RID	功能
USB_0	H'03	B'000000	B'11	—
USB_1	H'07	B'000001	B'11	—
SDHI	H'11	B'000100	B'01	发送
	H'12	B'000100	B'10	接收
SSI_0	H'23	B'001000	B'11	—
SSI_1	H'27	B'001001	B'11	—
SSI_2	H'2B	B'001010	B'11	—
SSI_3	H'2F	B'001011	B'11	—

外围模块	1 通道的设定值 （{MID、 RID}）	MID	RID	功能
SRC	H'41	B'010000	B'01	输入数据空
	H'42		B'10	输出数据满
SSU_0	H'51	B'010100	B'01	发送
	H'52		B'10	接收
SSU_1	H'55	B'010101	B'01	发送
	H'56		B'10	接收
IIC3_0	H'61	B'011000	B'01	发送
	H'62		B'10	接收
IIC3_1	H'65	B'011001	B'01	发送
	H'66		B'10	接收
IIC3_2	H'69	B'011010	B'01	发送
	H'6A		B'10	接收
IIC3_3	H'6D	B'011011	B'01	发送
	H'6E		B'10	接收
ROM-DEC	H'73	B'011100	B'11	—
SCIF_0	H'81	B'100000	B'01	发送
	H'82		B'10	接收
SCIF_1	H'85	B'100001	B'01	发送
	H'86		B'10	接收
SCIF_2	H'89	B'100010	B'01	发送
	H'8A		B'10	接收
SCIF_3	H'8D	B'100011	B'01	发送
	H'8E		B'10	接收
A/D 转换器 _0	H'B3	B'101100	B'11	—
FLCTL_0	H'BB	B'101110	B'11	发送 / 接收数据
FLCTL_1	H'BF	B'101111	B'11	发送 / 接收管理码
MTU2_0	H'E3	B'111000	B'11	—
MTU2_1	H'E7	B'111001	B'11	—
MTU2_2	H'EB	B'111010	B'11	—
MTU2_3	H'EF	B'111011	B'11	—
MTU2_4	H'F3	B'111100	B'11	—
CMT_0	H'FB	B'111110	B'11	—
CMT_1	H'FF	B'111111	B'11	—

不能保证设定表 10.4 以外的 MID 或者 RID 时的运行。 DMARS 寄存器的传送请求只在设定为 CHCR_0 ～ 7 寄存器的资源选择位 （RS[3:0]） =B'1000 时有效。 B'1000 以外时，即使设定 DMARS 也不能作为传送请求源接受。

10.4　运行说明

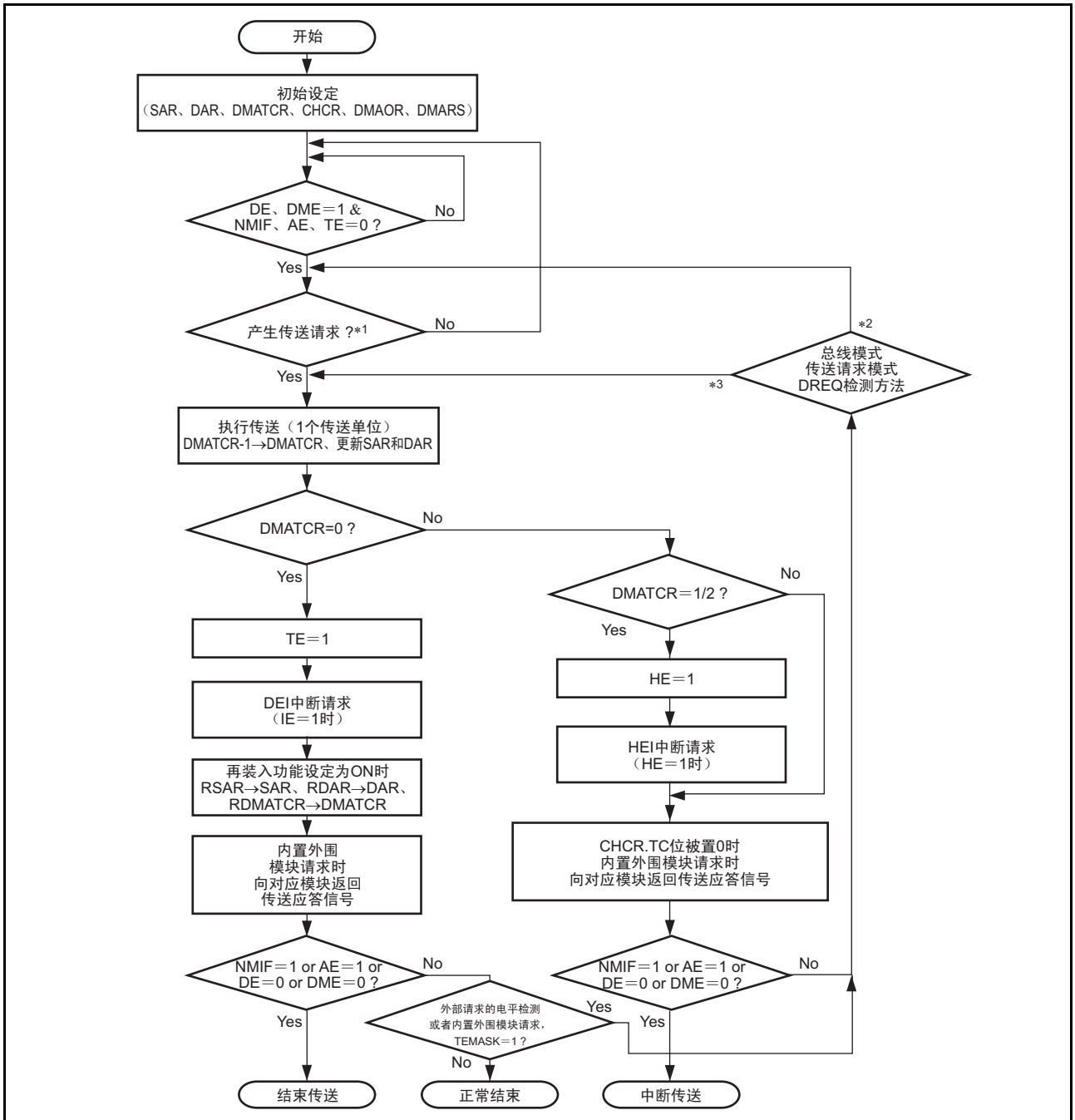
当产生 DMA 传送请求时，DMAC 按照规定的通道优先级开始传送，满足传送结束条件时，DMAC 结束传送。传送请求有自动请求、外部请求和内置外围模块请求 3 种模式。可选择突发模式或者周期挪用模式的总线模式。

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） 、DMA 操作寄存器 （DMAOR） 、3 个再装入寄存器 （RSAR、RDAR、RDMATCR）和 DMA 扩展资源选择 （DMARS）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、TEMASK=0 且 TE=0 或者 TEMASK=1、AE=0、NMIF=0）。
2. 在传送允许状态下产生传送请求时，传送 1 个传送单位的数据 （取决于 TS[1:0] 位的设定）。在自动请求模式中， DE 位和 DME 位被置 1 时，自动开始传送。每进行 1 次传送， DMATCR 的值就递减 1。具体的传送流程因地址模式和总线模式而不同。
3. 超过指定次数一半的传送 （DMATCR 的值为初始值的 1/2）时，如果 CHCR 的 HIE 位置 1，就向 CPU 产生 HEI 中断。
4. TEMASK=0 时，如果结束指定次数的传送 （DMATCR 的值为 0），传送就正常结束。此时如果 CHCR 的 IE 位置 1，就向 CPU 产生 DEI 中断。TEMASK=1 时，如果 DMATCR 的值为 0，设定 TE=1 后将指定的 RSAR、 RDAR、 RDMATCR 值再装入到 SAR、 DAR、 DMATCR，继续进行传送直到取消传送请求。
5. 如果产生 DMAC 引起的地址错误或者 NMI 中断，就中断传送。即使 CHCR 的 DE 位或者 DMAOR 的 DME 位清 0，也将中断传送。

上述的流程图如**图 10.2** 所示。



【注】*1　在自动请求模式中，如果在NMIF、AE和TE位都为0时将DE位和DME位置1，就开始传送。
*2　突发模式中的DREQ电平检测（外部请求）、或者周期挪用模式。
*3　突发模式中的DREQ边沿检测（外部请求）、或者突发模式中的自动请求模式。

图 10.2　DMA 传送流程图

10.4.2　DMA 传送请求

基本的使用方法是使 DMA 传送请求产生在数据的传送源或者传送目标，但是也有使 DMA 传送请求产生在既不是传送源又不是传送目标的外部器件或者内置外围模块的使用方法。

传送请求有自动请求、外部请求和内置外围模块请求 3 种。通过 CHCR_0 ～ CHCR_7 的 RS[3:0] 位及 DMARS0、 DMARS1、 DMARS2、 DMARS3 寄存器选择传送请求。

（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置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。CHCR_0 ～ CHCR_7 的 DE 位及 DMAOR 的 DME 位被置 1 时，就开始传送。但是， CHCR_0 ～ CHCR_7 的 TE 位、 DMAOR 的 AE、 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 10.5 所示的模式中选择 1 种使用。允许 DMA 传送（电平检测时，DE=1、DME=1、TEMASK=0 且 TE=0 或者 TEMASK=1、AE=0、NMIF=0 ；边沿检测时，DE=1、DME=1、TE=0、AE=0、NMIF=0）时，如果输入 DREQ，就开始 DMA 传送。

表 10.5　通过 RS 位选择外部请求模式

RS[3]	RS[2]	RS[1]	RS[0]	地址模式	传送源	传送目标
0	0	0	0	双地址模式	任意	任意
0	0	1	0	单地址模式	外部存储器或者 存储器映像的外部器件	带 DACK 的外部器件
			1		带 DACK 的外部器件	外部存储器或者 存储器映像的外部器件

如表 10.6 所示，通过 CHCR_0 ～ CHCR_3 的 DL 位和 DS 位，选择是通过边沿还是通过电平来检测 DREQ。传送请求源不一定必须是数据的传送源或者传送目标。在突发模式下进行上升沿检测或者下降沿检测时，持续传送直到通过 1 次传送请求使 DMATCR=0。周期挪用模式下，通过 1 次传送请求进行 1 次传送。

表 10.6　通过 DL 或者 DS 位选择外部请求检测

CHCR		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

接受 DREQ 时， DREQ 引脚处于不能接受请求的状态 （死区）。对已接受的 DREQ 输出应答 DACK 后，DREQ 引脚再次成为能接受请求的状态。

通过电平检测使用 DREQ 时，根据从输出 DACK 到检测出下一个 DREQ 的时序，有 2 种情况：①执行和请求次数相同的传送后产生中断 （溢出 0）；②执行比请求次数多 1 次的传送后产生中断 （溢出 1）。通过 CHCR 的 DO 位选择溢出 0 或者溢出 1。

表 10.7　通过 DO 位选择外部请求检测

CHCR 的 DO 位	外部请求
0	溢出 0
1	溢出 1

（3） 内置外围模块请求模式

在内置外围模块请求模式中，通过内置外围模块的 DMA 传送请求信号进行传送。

内置外围模块对 DMAC 发出的传送请求信号一览如表 10.8 所示。

选择内置外围模块请求模式时，如果是 DMA 传送允许状态 （DE=1、 DME=1、 TEMASK=0 且 TE=0 或者 TEMASK=1、 AE=0、 NMIF=0），就通过传送请求信号进行传送。

内置外围模块请求时，有传送源、传送目标被固定的情况。请参照表 10.8。

表 10.8　通过 RS3 ～ RS0 位选择内置外围模块请求模式

CHCR	DMARS		DMA 传送请求源	DMA 传送请求信号	传送源	传送目标	总线模式
RS[3:0]	MID	RID					
1001	任意	任意	RCAN-TL10 接收	DM0 （接收完成）	RCAN0 MB0	任意	周期挪用
1010	任意	任意	RCAN-TL11 接收	DM0 （接收完成）	RCAN1 MB0	任意	
1000	000000	11	USB	USB_DMA0 （接收 FIFO 满）	D0FIFO	任意	周期挪用 / 突发
				USB_DMA0 （发送 FIFO 空）	任意	D0FIFO	
	000001	11	USB	USB_DMA1 （接收 FIFO 满）	D1FIFO	任意	
				USB_DMA1 （发送 FIFO 空）	任意	D1FIFO	
	000100	01	SDHI 发送	TXI （发送数据空）	任意	数据寄存器	周期挪用
		10	SDHI 接收	RXI （接收数据满）	数据寄存器	任意	

CHCR	DMARS		DMA 传送请求源	DMA 传送请求信号	传送源	传送目标	总线模式
RS[3:0]	MID	RID					
1000	001000	11	SSI_0	DMA0 （发送模式）	任意	SSITDR0	周期挪用
				DMA0 （接收模式）	SSIRDR0	任意	
	001001	11	SSI_1	DMA1 （发送模式）	任意	SSITDR1	
				DMA1 （接收模式）	SSIRDR1	任意	
	001010	11	SSI_2	DMA2 （发送模式）	任意	SSITDR2	
				DMA2 （接收模式）	SSIRDR2	任意	
	001011	11	SSI_3	DMA3 （发送模式）	任意	SSITDR3	
				DMA3 （接收模式）	SSIRDR3	任意	
	010000	01	SRC 输入	IDEI （输入数据空）	任意	SRCIDR	
		10	SRC 输出	ODFI （输出数据满）	SRCODR	任意	
	010100	01	SSU_0 发送	SSTXI0 （发送空或者发送结束）	任意	SSTDR0 ～ 3	周期挪用
		10	SSU_0 接收	SSRXI0 （接收满）	SSRDR0 ～ 3	任意	
	010101	01	SSU_1 发送	SSTXI1 （发送空或者发送结束）	任意	SSTDR0 ～ 3	
		10	SSU_1 接收	SSRXI1 （接收满）	SSRDR0 ～ 3	任意	
	011000	01	IIC3_0 发送	TXI0 （发送数据空）	任意	ICDRT0	
		10	IIC3_0 接收	RXI0 （接收数据满）	ICDRR0	任意	
	011001	01	IIC3_1 发送	TXI1 （发送数据空）	任意	ICDRT1	
		10	IIC3_1 接收	RXI1 （接收数据满）	ICDRR1	任意	
	011010	01	IIC3_2 发送	TXI2 （发送数据空）	任意	ICDRT2	
		10	IIC3_2 接收	RXI2 （接收数据满）	ICDRR2	任意	
	011011	01	IIC3_3 发送	TXI3 （发送数据空）	任意	ICDRT3	周期挪用
		10	IIC3_3 接收	RXI3 （接收数据满）	ICDRR3	任意	
	011100	11	ROM-DEC	IREADY （解码结束）	STRMDOUT	任意	周期挪用 / 突发

CHCR	DMARS		DMA 传送请求源	DMA 传送请求信号	传送源	传送目标	总线模式
RS[3:0]	MID	RID					
1000	100000	01	SCIF_0 发送	TXI0 （发送 FIFO 数据空）	任意	SCFTDR_0	周期挪用
		10	SCIF_0 接收	RXI0 （接收 FIFO 数据满）	SCFRDR_0	任意	
	100001	01	SCIF_1 发送	TXI1 （发送 FIFO 数据空）	任意	SCFTDR_1	
		10	SCIF_1 接收	RXI1 （接收 FIFO 数据满）	SCFRDR_1	任意	
	100010	01	SCIF_2 发送	TXI2 （发送 FIFO 数据空）	任意	SCFTDR_2	
		10	SCIF_2 接收	RXI2 （接收 FIFO 数据满）	SCFRDR_2	任意	
	100011	01	SCIF_3 发送	TXI3 （发送 FIFO 数据空）	任意	SCFTDR_3	
		10	SCIF_3 接收	RXI3 （接收 FIFO 数据满）	SCFRDR_3	任意	
	101100	11	A/D 转换器	ADI （A/D 转换结束）	ADDR	任意	周期挪用
	101110	11	FLCTL 数据部发送	发送 FIFO 数据空	任意	FLDTFIFO	周期挪用
			FLCTL 数据部接收	接收 FIFO 数据满	FLDTFIFO	任意	
	101111	11	FLCTL 管理码部发送	发送 FIFO 数据空	任意	FLECFIFO	
			FLCTL 管理码部接收	接收 FIFO 数据满	FLECFIFO	任意	
	111000	11	MTU2_0	TGI0A （输入捕捉 / 比较匹配）	任意	任意	周期挪用 / 突发
	111001	11	MTU2_1	TGI1A （输入捕捉 / 比较匹配）	任意	任意	
	111010	11	MTU2_2	TGI2A （输入捕捉 / 比较匹配）	任意	任意	
	111011	11	MTU2_3	TGI3A （输入捕捉 / 比较匹配）	任意	任意	周期挪用 / 突发
	111100	11	MTU2_4	TGI4A （输入捕捉 / 比较匹配）	任意	任意	
	111110	11	CMT_0	CMI0 （比较匹配）	任意	任意	
	111111	11	CMT_1	CMI1 （比较匹配）	任意	任意	

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。从固定模式 1、固定模式 2 和循环模式这 3 种模式中，选择通道优先级。通过 DMAOR 的 PR 位和 PR0 位选择模式。

（1） 固定模式

在固定模式 1 及 2 中，通道优先级不变。

各模式优先级如下：

- 固定模式1：CH0＞CH1＞CH2＞CH3＞CH4＞CH5＞CH6＞CH7
- 固定模式2：CH0＞CH4＞CH1＞CH5＞CH2＞CH6＞CH3＞CH7

通过 DMAOR 的 PR1 位和 PR0 位进行选择。

（2） 循环模式

在循环模式中，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，使该通道优先级在循环对象通道内变为最低。另外，循环对象的通道只有 CH0 ～ CH3 的 4 个通道。此外的通道即使在循环模式也不变更优先级。运行如图 10.3 所示。另外，刚复位的循环模式优先级为 CH0 ＞ CH1 ＞ CH2 ＞ CH3 ＞ CH4 ＞ CH5 ＞ CH6 ＞ CH7。

当指定循环模式时，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。

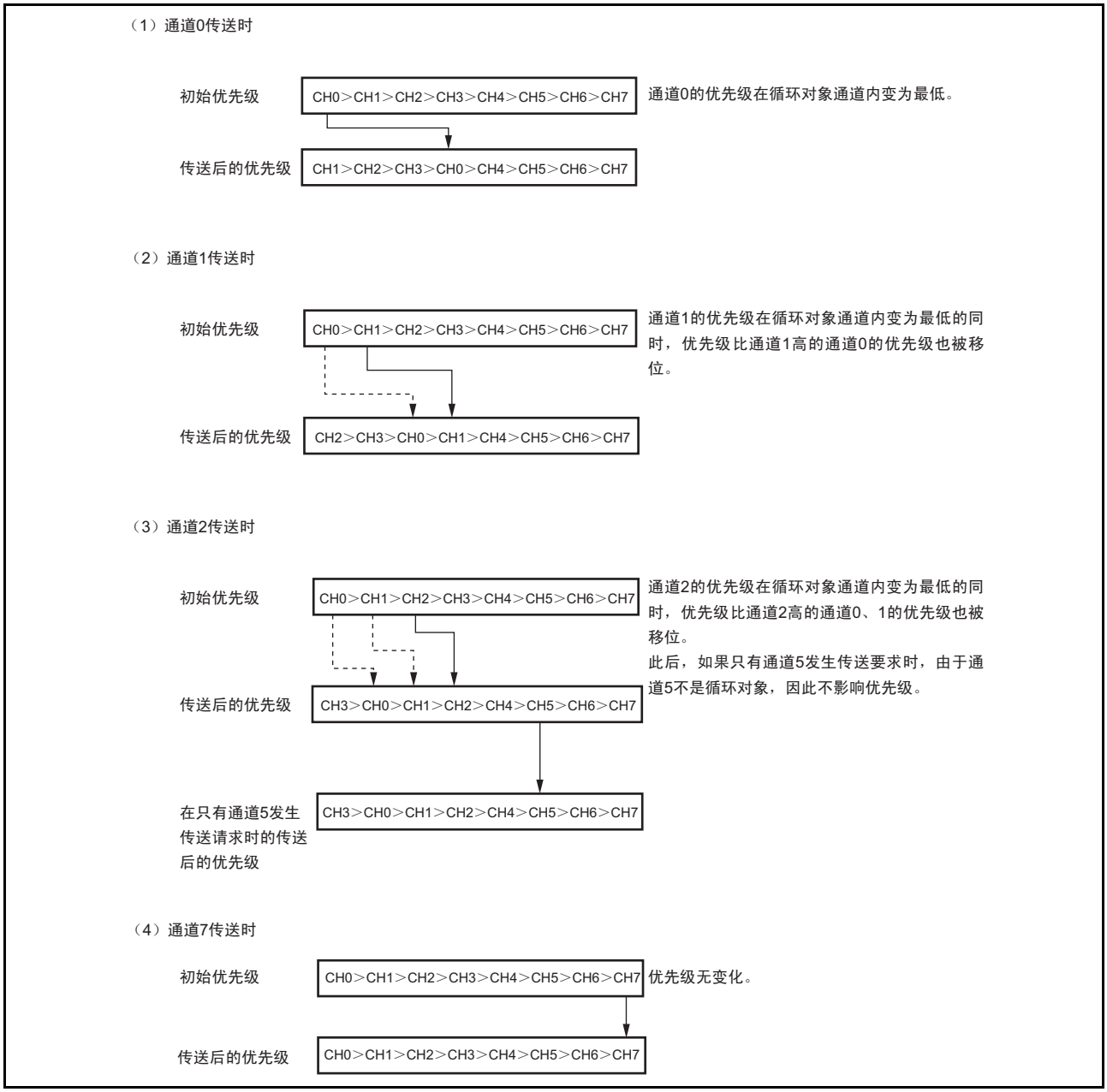

（1）通道0传送时
初始优先级
CH0＞CH1＞CH2＞CH3＞CH4＞CH5＞CH6＞CH7
通道0的优先级在循环对象通道内变为最低。
传送后的优先级
CH1＞CH2＞CH3＞CH0＞CH4＞CH5＞CH6＞CH7
（2）通道1传送时
初始优先级
CH0＞CH1＞CH2＞CH3＞CH4＞CH5＞CH6＞CH7
通道1的优先级在循环对象通道内变为最低的同时，优先级比通道1高的通道0的优先级也被移位。
传送后的优先级
CH2＞CH3＞CH0＞CH1＞CH4＞CH5＞CH6＞CH7
（3）通道2传送时
初始优先级
CH0＞CH1＞CH2＞CH3＞CH4＞CH5＞CH6＞CH7
通道2的优先级在循环对象通道内变为最低的同时，优先级比通道2高的通道0、1的优先级也被移位。
此后，如果只有通道5发生传送要求时，由于通道5不是循环对象，因此不影响优先级。
传送后的优先级
CH3＞CH0＞CH1＞CH2＞CH4＞CH5＞CH6＞CH7
在只有通道5发生传送请求时的传送后的优先级
CH3＞CH0＞CH1＞CH2＞CH4＞CH5＞CH6＞CH7
（4）通道7传送时
初始优先级
CH0＞CH1＞CH2＞CH3＞CH4＞CH5＞CH6＞CH7
优先级无变化。
传送后的优先级
CH0＞CH1＞CH2＞CH3＞CH4＞CH5＞CH6＞CH7

图 10.3　循环模式

当通道 0 和通道 3 同时产生传送请求，并且在通道 0 传送时产生通道 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同时产生传送请求。
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产生传送请求 （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的优先级在循环对象通道内变为最低。
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的优先级在循环对象通道内变为最低。
7. 开始通道3的传送。
8. 当通道3的传送结束时，通道3的优先级在循环对象通道内变为最低，同时通道2的优先级也变低。

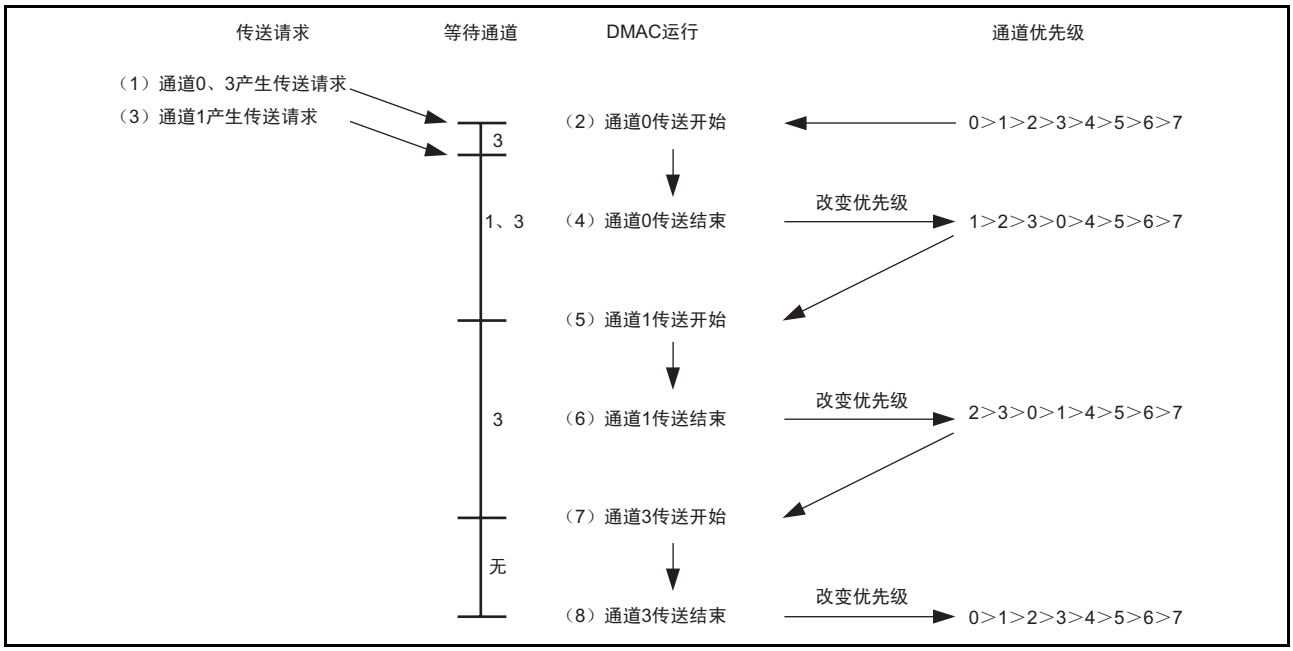


图 10.4　循环模式中的通道优先级

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 10.9 示。

表 10.9　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	内置外围模块	内置存储器
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
存储器映像的外部器件	双、单	双	双	双	双
内置外围模块	不能	双	双	双	双
内置存储器	不能	双	双	双	双

【注】 1. 双：双地址模式

2. 单：单地址模式

3. 内置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 （选择）传送源和传送目标的模式。传送源和传送目标可以是外部也可以是内部。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 10.5 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。

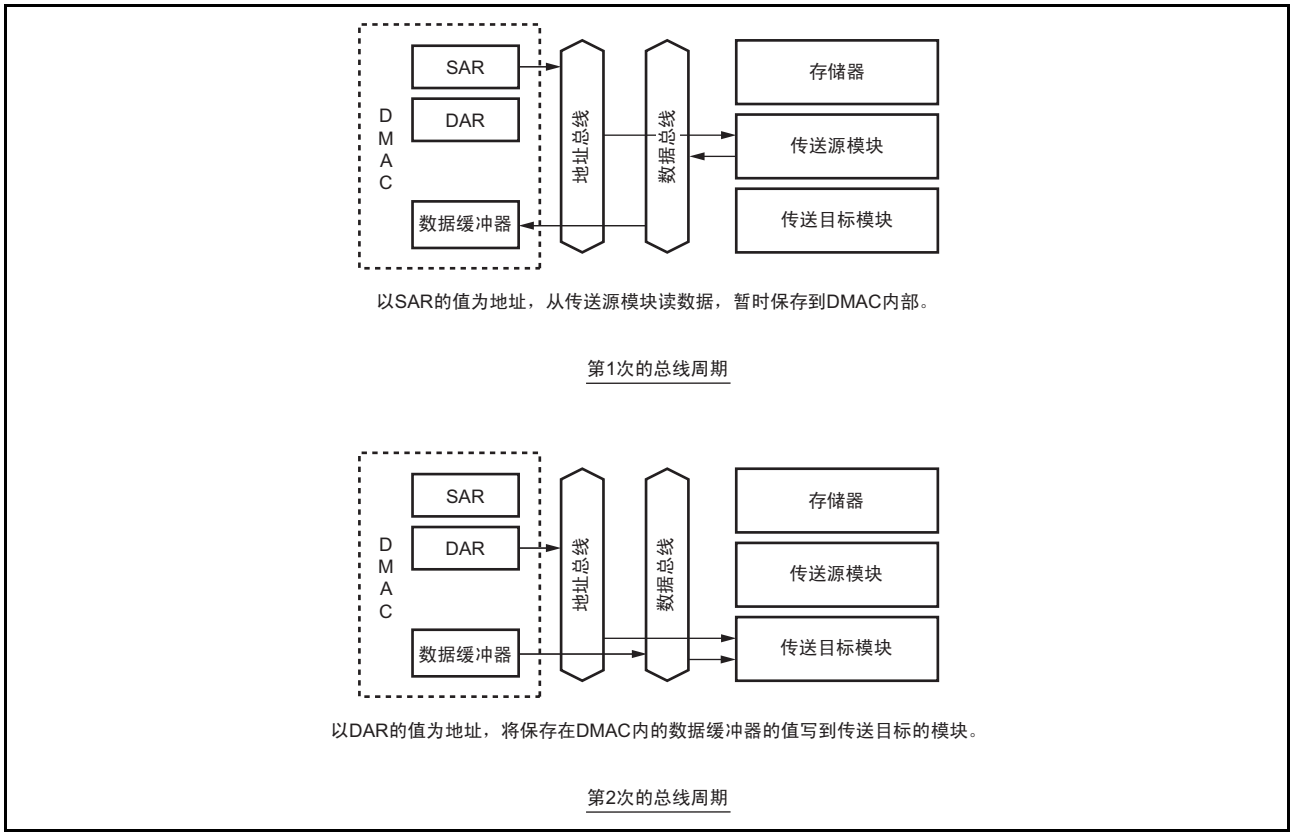


图 10.5　双地址模式的数据流程

传送请求可以是自动请求、外部请求或者内置外围模块请求。在双地址模式中，能在读周期或者写周期中输出 DACK。能通过 CHCR 的 AM 位设定 DACK 是在读周期中还是在写周期中输出。

双地址模式中的 DMA 传送时序例子如图 10.6 所示。

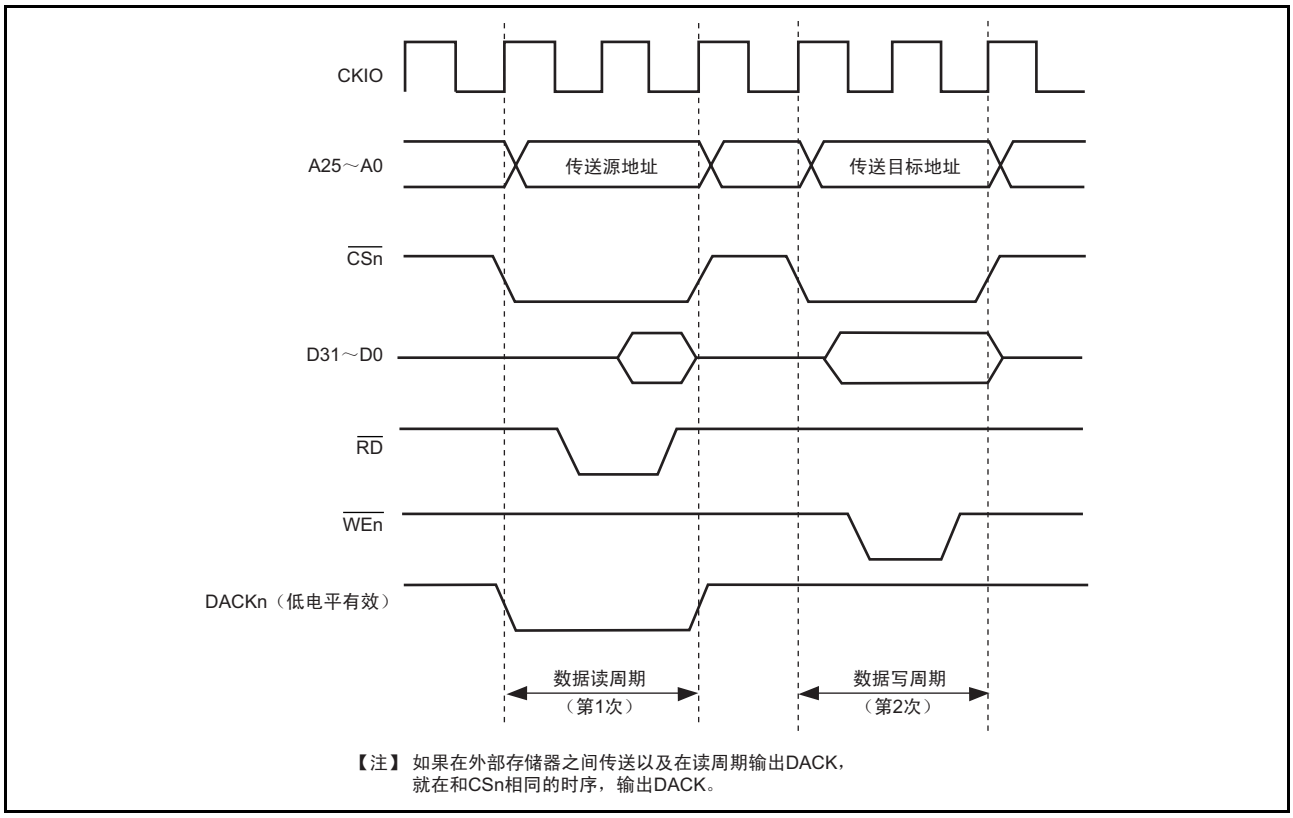


图 10.6　双地址模式中的 DMA 传送时序例 （传送源：普通存储器，传送目标：普通存储器）

（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，将地址输出到传送目标，通过 1 个总线周期进行 DMA 传送。例如，对于图 10.7 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相同的总线周期中，将该数据写到外部存储器。

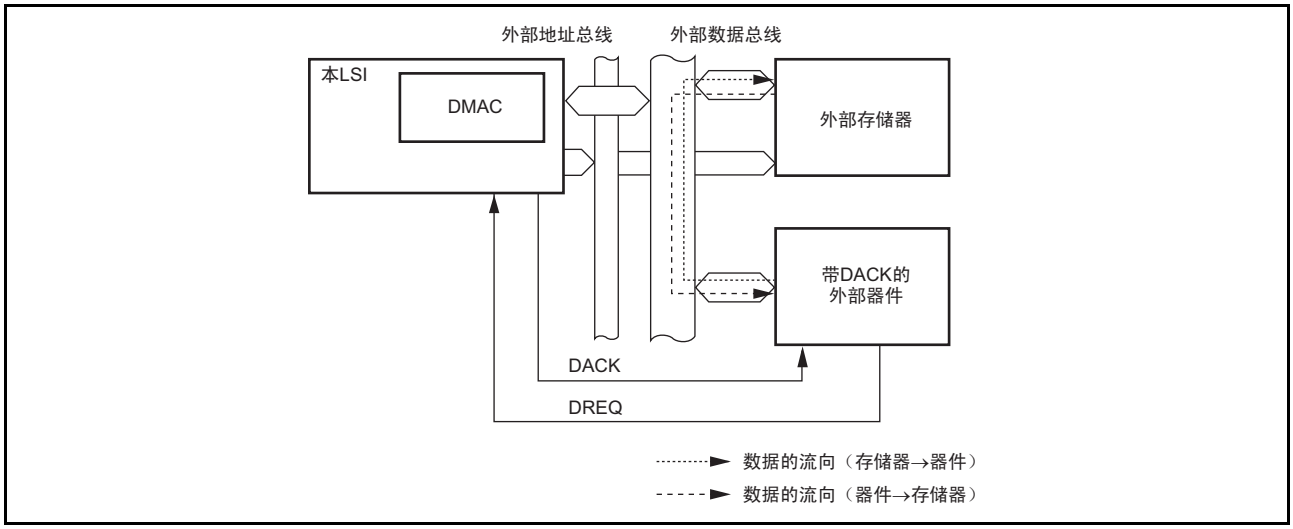


图 10.7　单地址模式的数据流程

能在单地址模式中进行（1）带 DACK 的外部器件和存储器映像的外部器件之间的传送、（2）带 DACK 的外部器件和外部存储器之间的传送。无论哪种情况，传送请求都只是外部请求 （DREQ）。

单地址模式中的 DMA 传送时序例如图 10.8 所示。

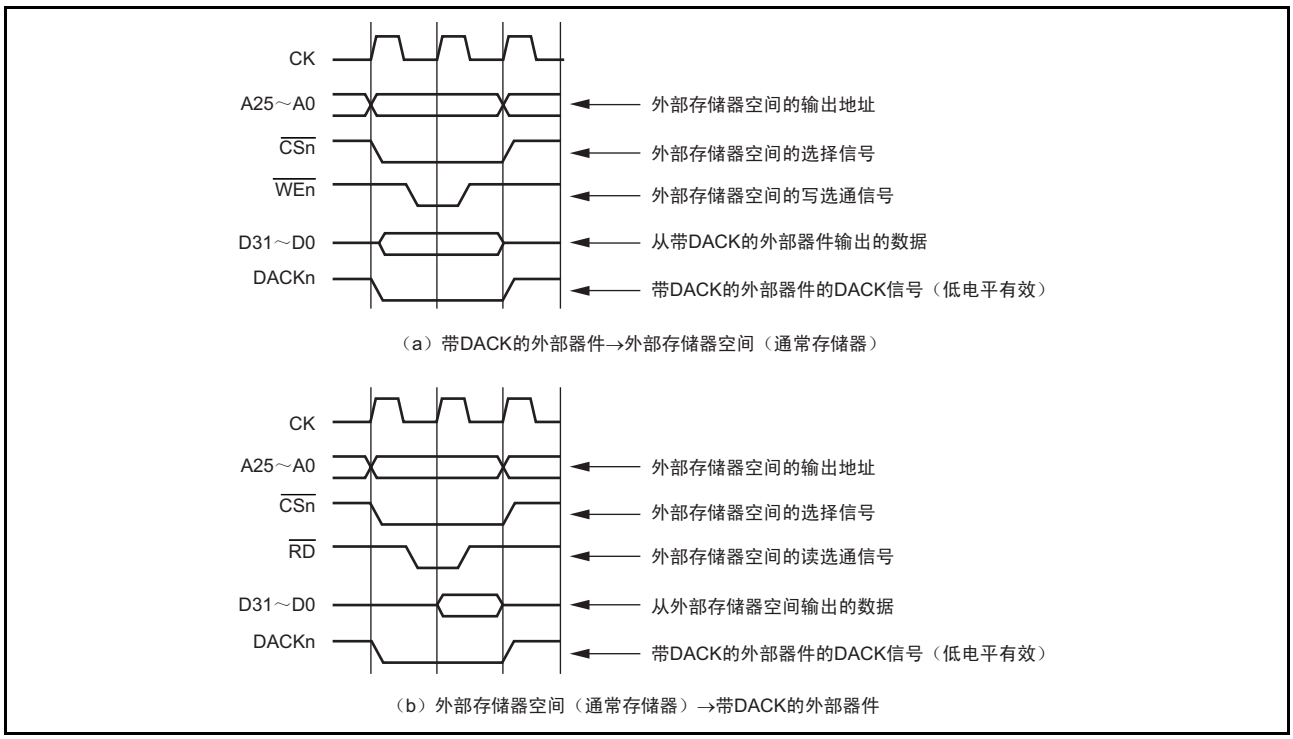


图 10.8　单地址模式中的 DMA 传送时序例

（2） 总线模式

总线模式有周期挪用模式和突发模式，通过 CHCR 的 TB 位进行选择。

（a） 周期挪用模式

- 普通模式

在周期挪用的普通模式中，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器。如果此后产生传送请求，就从其他总线主控器取回总线权，再次进行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器。重复此操作直到满足传送结束条件为止。

周期挪用的普通模式与传送请求源、传送源和传送目标无关，可在所有传送区间内使用。

周期挪用普通模式中的DMA传送时序例子如**图**10.9所示，图中的传送条件如下：

- 双地址模式
- DREQ低电平检测

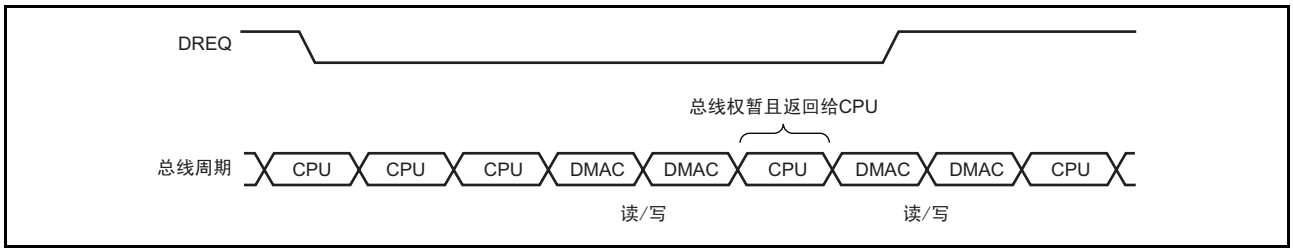


图 10.9　周期挪用普通模式中的 DMA 传送例 （双地址、 DREQ 低电平检测）

- 间歇模式16、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器。如果此后产生传送请求，就通过Bϕ时钟计数等待16个时钟或者 64 个时钟，然后从其他总线主控器取回总线权，再次进行 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器。重复此操作直到满足结束条件为止。因此，和周期挪用的普通模式相比，能降低DMA传送的总线占有率。

DMAC再次取回总线权时，在进行由于高速缓存未中引起的入口更新等时， DMA传送可能需要进一步延迟。

间歇模式与传送请求源、传送源和传送目标无关，可在所有传送区间内使用，但是所有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

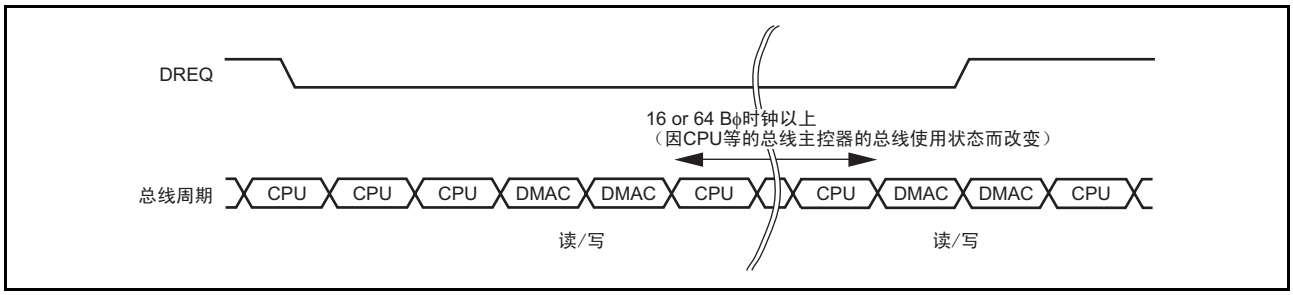


图 10.10　周期挪用间歇模式中的 DMA 传送例 （双地址、 DREQ 低电平检测）

（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出 DREQ 时，如果 DREQ 变为无效电平，即使尚未满足传送结束条件，也将在结束已接受请求的 DMAC 传送请求后，将总线权交给其他总线主控器。

突发模式中的 DMA 传送时序如图 10.11 所示。

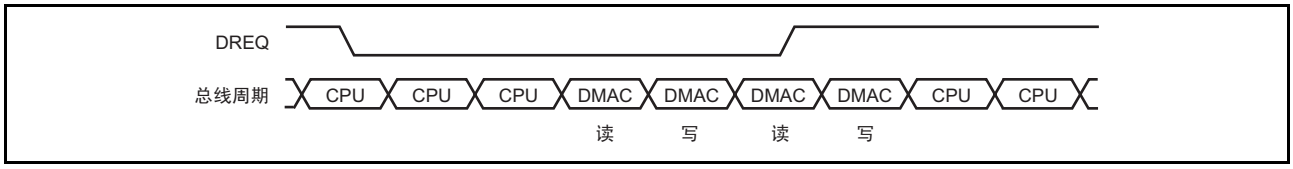


图 10.11　突发模式中的 DMA 传送例 （双地址、 DREQ 低电平检测）

（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 10.10 所示。

表 10.10　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 的外部器件和存储器映像的外部器件	外部	B/C	8/16/32/128	0 ～ 3
	外部存储器和外部存储器	均可 *4	B/C	8/16/32/128	0 ～ 7*3
	外部存储器和存储器映像的外部器件	均可 *4	B/C	8/16/32/128	0 ～ 7*3
	存储器映像的外部器件和存储器映像的外部器件	均可 *4	B/C	8/16/32/128	0 ～ 7*3
	外部存储器和内置外围模块	均可 *1	B/C*5	8/16/32/128*2	0 ～ 7*3
	存储器映像的外部器件和内置外围模块	均可 *1	B/C*5	8/16/32/128*2	0 ～ 7*3
	内置外围模块和内置外围模块	均可 *1	B/C*5	8/16/32/128*2	0 ～ 7*3
	内置存储器和内置存储器	均可 *4	B/C	8/16/32/128	0 ～ 7*3
	内置存储器和存储器映像的外部器件	均可 *4	B/C	8/16/32/128	0 ～ 7*3
	内置存储器和内置外围模块	均可 *1	B/C*5	8/16/32/128*2	0 ～ 7*3
	内置存储器和外部存储器	均可 *4	B/C	8/16/32/128	0 ～ 7*3
单地址模式	带 DACK 的外部器件和外部存储器	外部	B/C	8/16/32/128	0 ～ 3
	带 DACK 的外部器件和存储器映像的外部器件	外部	B/C	8/16/32/128	0 ～ 3

【符号说明】

B　：突发

C　：周期挪用

【注】 *1　外部请求、自动请求和内置外围模块请求均可。
但是在内置外围模块请求时，传送请求源除了MTU2以及CMT，传送源或者传送目标必须为各自的请求源寄存器。

*2　为传送源或者传送目标的内置外围模块的寄存器允许的存取尺寸。

*3　传送请求为外部请求时只能利用通道 0 ～ 3。

*4　外部请求、自动请求和内置外围模块请求均可。
但是，在内置外围模块请求的情况下，只能利用MTU2、 CMT。

*5　在内置外围模块请求的情况下，除了 USB、SSI、ROM-DEC、SRC、MTU2 及 CMT，传送请求源只有周期挪用。

（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 产生传送请求，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，通道 1 继续传送。

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 10.12 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器。

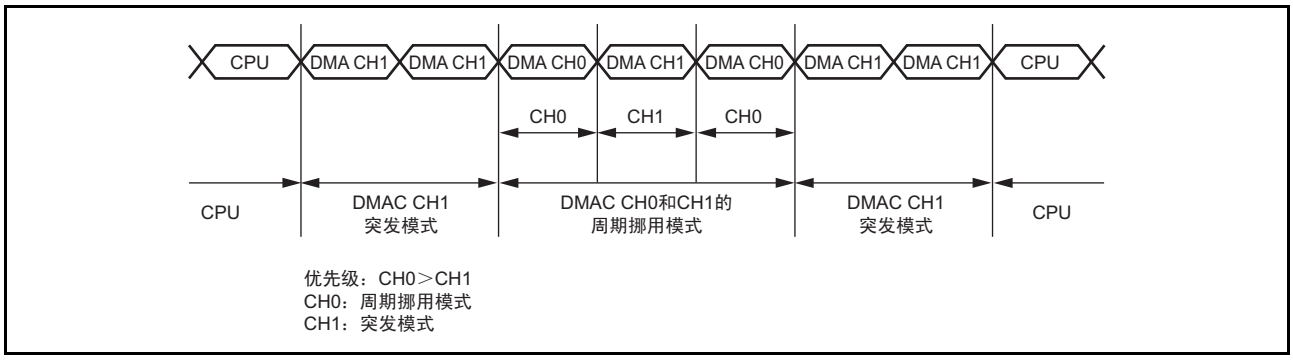


图 10.12　多个通道运行时的总线状态

在循环模式中，按照图 10.13 所示的规则，优先级发生变化。但是，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 为总线主控器时和 CPU 为总线主控器时一样，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 10.13 ～图 10.16 所示。

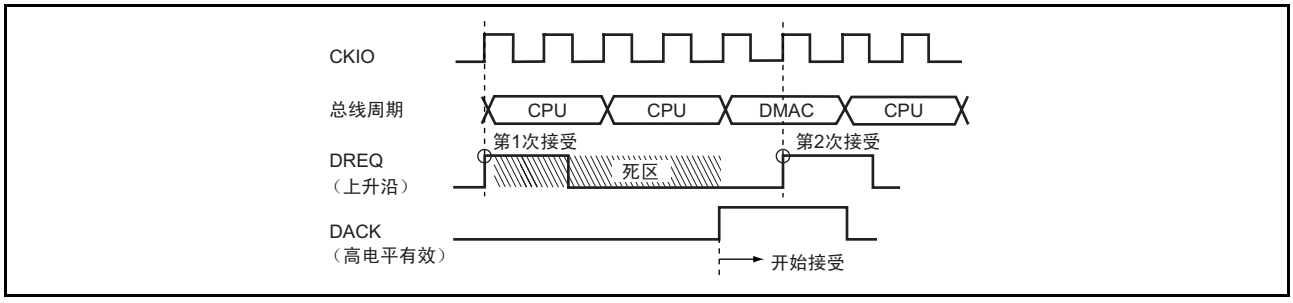


图 10.13　周期挪用模式、边沿检测时的 DREQ 输入检测时序

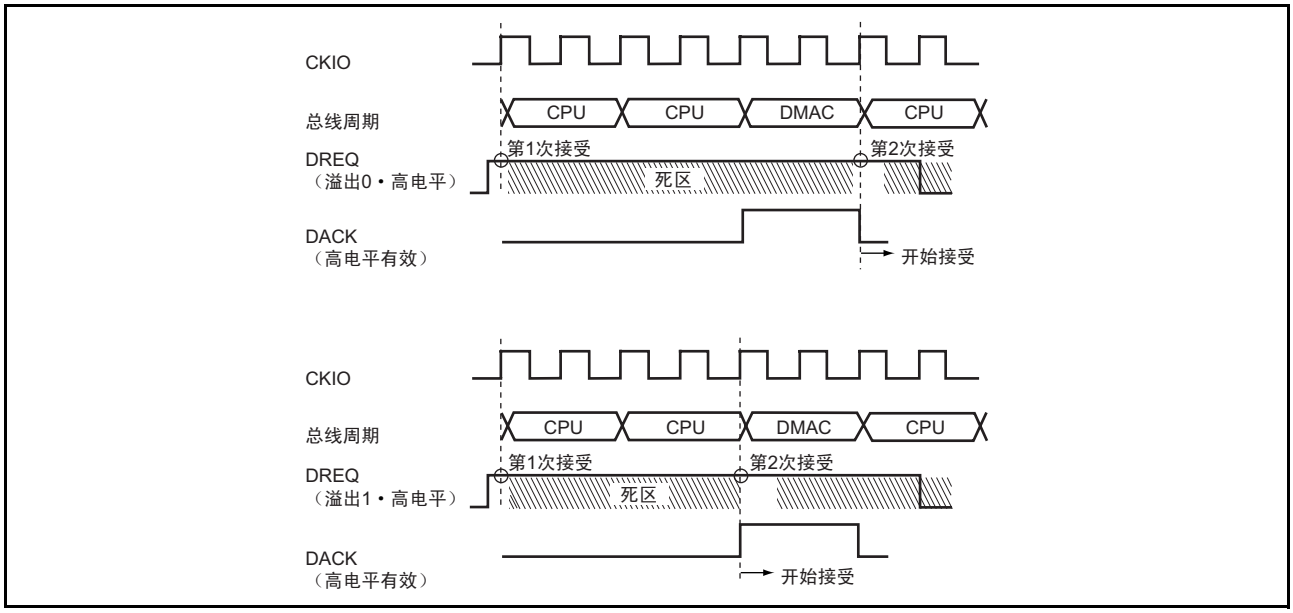


图 10.14　周期挪用模式、电平检测时的 DREQ 输入检测时序

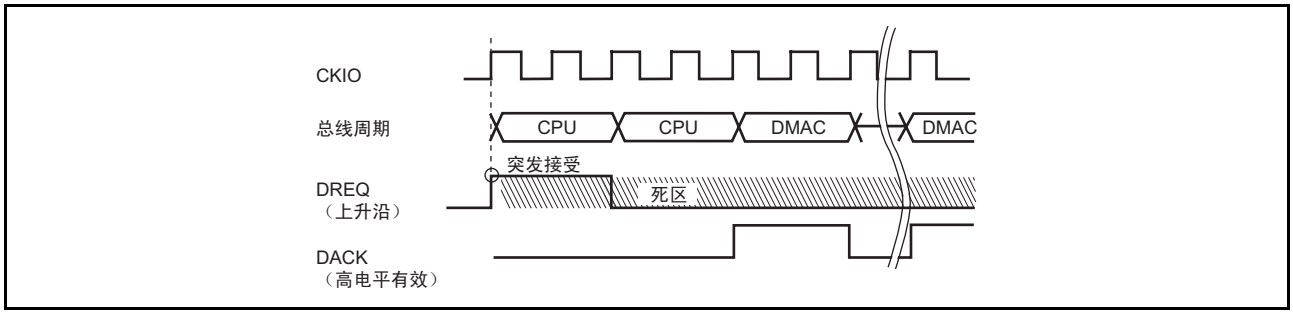


图 10.15　突发模式、边沿检测时的 DREQ 输入检测时序

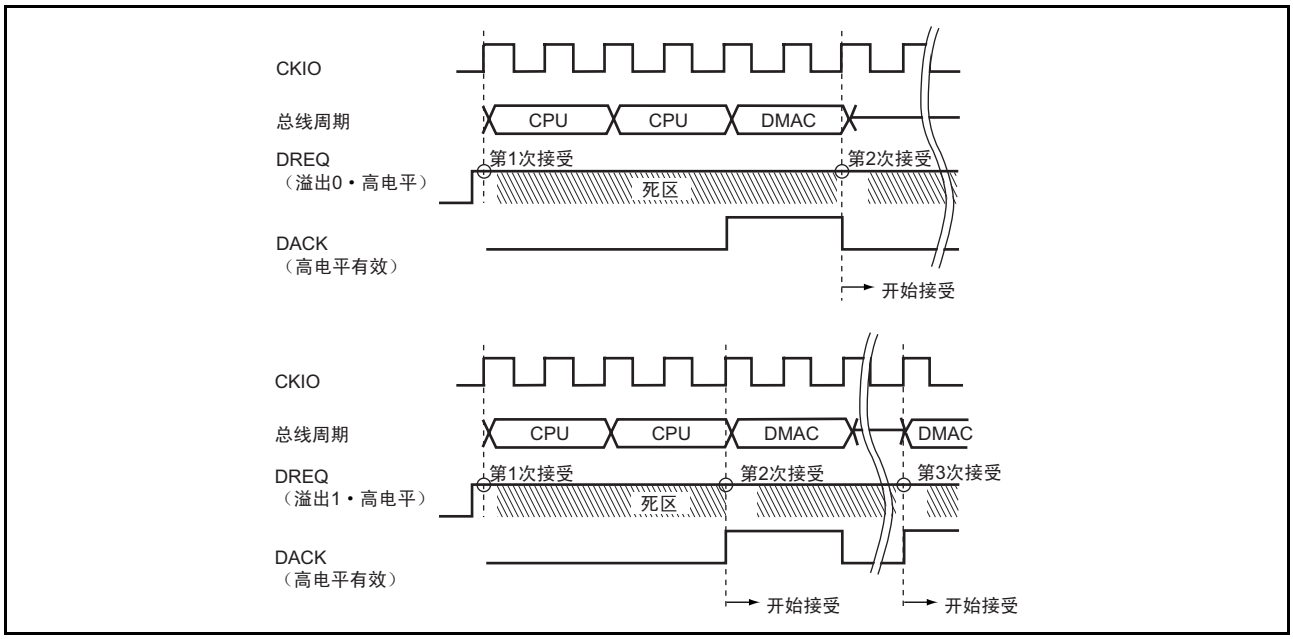


图 10.16　突发模式、电平检测时的 DREQ 输入检测时序

TEND 的输出时序如图 10.17 所示。

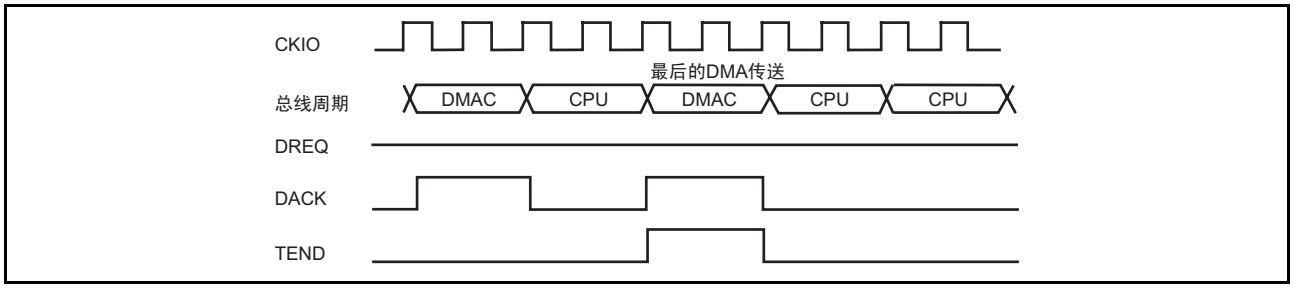


图 10.17　DMA 传送结束信号时序 （周期挪用、电平检测）

在对 8 位、16 位、32 位外部器件进行 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。请注意在 DMA 传送单位被分割为多个总线周期，且总线周期之间将 $\overline{CS}$ 设定为取反时，为了对齐数据，和 $\overline{CS}$ 一样 DACK 输出和 TEND 输出被分割。此例如图 10.18 所示。另外， DMA 传送时不分割 DACK、 TEND 的情况如图 10.13 ～图 10.17 所示。

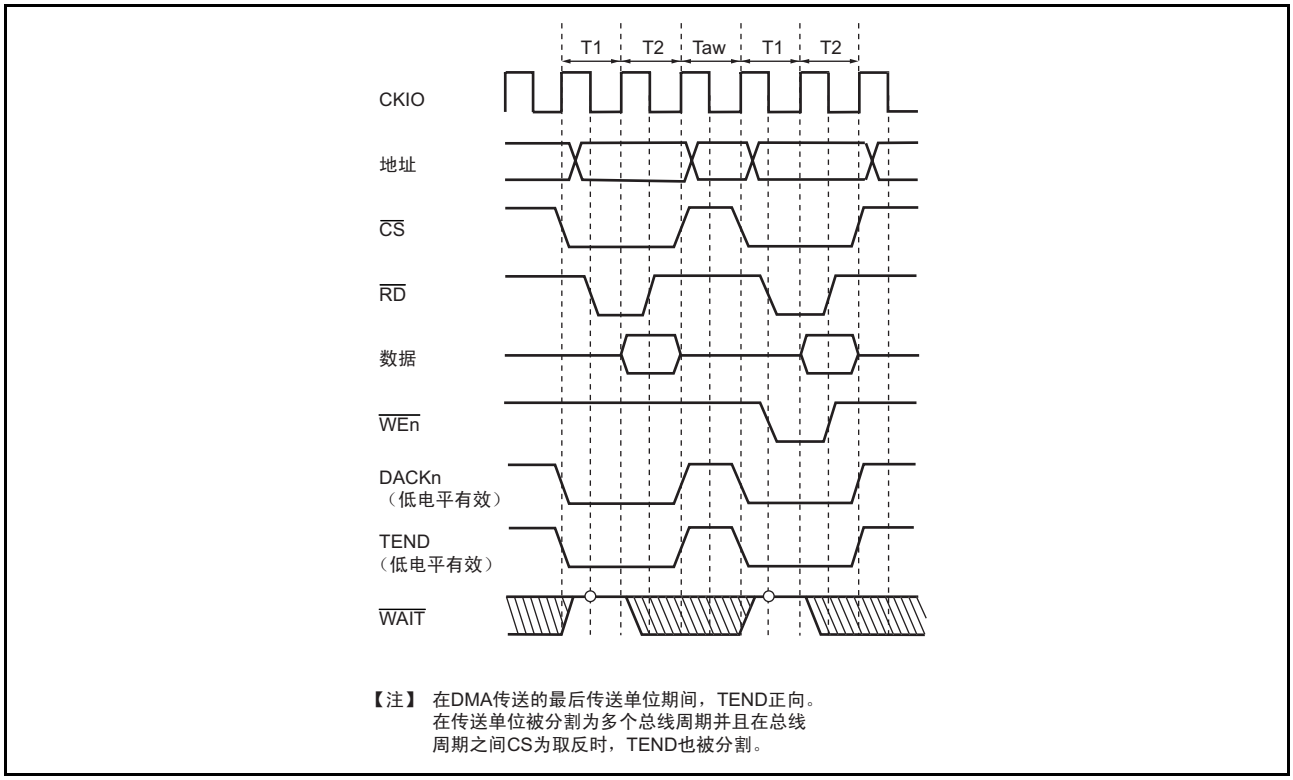


图 10.18　BSC 普通存储器存取
（无等待、空闲周期 =1、 16 位器件的长字存取

10.5　使用时的注意事项

10.5.1　半结束标志的置位及半结束中断

参照 CHCR 寄存器的半结束标志状态及使用半结束中断时，并且与再装入功能共同使用时，请注意以下几点：

请设定再装入传送次数（设定在 RDMATCR 的值）等同于最初设定的传送次数（设定在 DMATCR 的值）。第 2 次及以后的传送使用的 RDMATCR 设定值与最初的 DMATCR 设定值不相同时，就可能有设置半结束标志的时序快于传送次数的一半，或者不设置半结束标志的情况。半结束中断也相同。

10.5.2　DACK 输出及 TEND 输出的时序

外部存储器为 MPX-I/O 或者突发 MPX-I/O 时， DACK 输出通过数据周期的时序被置为断言。详细请参照“第 9 章 总线状态控制器 （BSC）”的 “9.5.5 MPX-I/O 接口”或者 “9.5.10 突发 MPX-I/O 接口”的各图。

MPX-I/O 及突发 MPX-I/O 以外的存储器时， DACK 输出与相应的 CS 在相同时序断言。

与存储器无关， TEND 输出总是与相应的 CS 在相同时序断言。

第 11 章　多功能定时器脉冲单元 2 （MTU2）

本 LSI 内置由 5 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
- 各通道可选择8种计数器输入时钟
- 能设定以下运行：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、多个定时器计数器 （TCNT）的同时写入、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而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
- 通过内部16位总线进行的高速存取
- 28种中断源
- 能进行寄存器数据的自动传送
- 能产生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
- 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
计数时钟		Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 TCLKA TCLKB TCLKC TCLKD	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/256 TCLKA TCLKB	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/1024 TCLKA TCLKB TCLKC	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/256 Pϕ/1024 TCLKA TCLKB	Pϕ/1 Pϕ/4 Pϕ/16 Pϕ/64 Pϕ/256 Pϕ/1024 TCLKA TCLKB
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
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
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
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		○	—	—	○	○

项目	通道 0	通道 1	通道 2	通道 3	通道 4
DMAC 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 上溢 / 下溢
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 互补 PWM 模式时 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
A/D 转换的开始请求延迟功能	—	—	—	—	• TADCORA_4 和 TCNT_4 匹配时，请求开始 A/D 转换 • TADCORB_4 和 TCNT_4 匹配时，请求开始 A/D 转换
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。

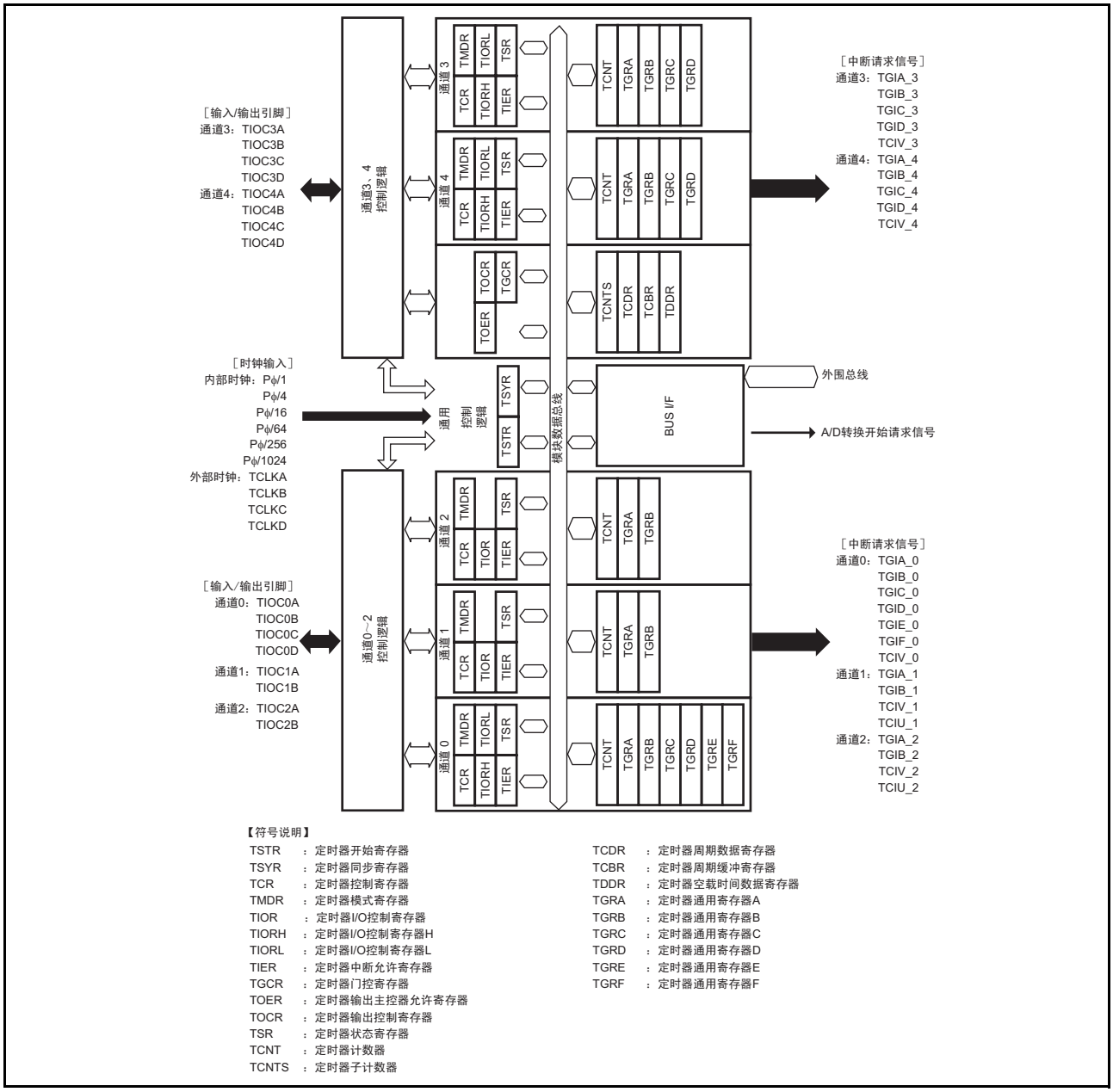


图 11.1　MTU2 的框图

11.2　输入 / 输出引脚

表 11.2　引脚构成

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

【注】　互补 PWM 模式引脚的结构请参照 “11.4.8　互补 PWM 模式”的表 11.54。

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 34 章 寄存器一览”。为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

通道	寄存器名	简称	R/W	初始值	地址	存取长度
0	定时器控制寄存器 _0	TCR_0	R/W	H'00	H'FFFE4300	8
	定时器模式寄存器 _0	TMDR_0	R/W	H'00	H'FFFE4301	8
	定时器 I/O 控制寄存器 H_0	TIORH_0	R/W	H'00	H'FFFE4302	8
	定时器 I/O 控制寄存器 L_0	TIORL_0	R/W	H'00	H'FFFE4303	8
	定时器中断允许寄存器 _0	TIER_0	R/W	H'00	H'FFFE4304	8
	定时器状态寄存器 _0	TSR_0	R/W	H'C0	H'FFFE4305	8
	定时器计数器 _0	TCNT_0	R/W	H'0000	H'FFFE4306	16
	定时器通用寄存器 A_0	TGRA_0	R/W	H'FFFF	H'FFFE4308	16
	定时器通用寄存器 B_0	TGRB_0	R/W	H'FFFF	H'FFFE430A	16
	定时器通用寄存器 C_0	TGRC_0	R/W	H'FFFF	H'FFFE430C	16
	定时器通用寄存器 D_0	TGRD_0	R/W	H'FFFF	H'FFFE430E	16
	定时器通用寄存器 E_0	TGRE_0	R/W	H'FFFF	H'FFFE4320	16
	定时器通用寄存器 F_0	TGRF_0	R/W	H'FFFF	H'FFFE4322	16
	定时器中断允许寄存器 2_0	TIER2_0	R/W	H'00	H'FFFE4324	8
	定时器状态寄存器 2_0	TSR2_0	R/W	H'C0	H'FFFE4325	8
	定时器缓冲运行传送模式寄存器 _0	TBTM_0	R/W	H'00	H'FFFE4326	8
1	定时器控制寄存器 _1	TCR_1	R/W	H'00	H'FFFE4380	8
	定时器模式寄存器 _1	TMDR_1	R/W	H'00	H'FFFE4381	8
	定时器 I/O 控制寄存器 _1	TIOR_1	R/W	H'00	H'FFFE4382	8
	定时器中断允许寄存器 _1	TIER_1	R/W	H'00	H'FFFE4384	8
	定时器状态寄存器 _1	TSR_1	R/W	H'C0	H'FFFE4385	8
	定时器计数器 _1	TCNT_1	R/W	H'0000	H'FFFE4386	16
	定时器通用寄存器 A_1	TGRA_1	R/W	H'FFFF	H'FFFE4388	16
	定时器通用寄存器 B_1	TGRB_1	R/W	H'FFFF	H'FFFE438A	16
	定时器输入捕捉控制寄存器	TICCR	R/W	H'00	H'FFFE4390	8

通道	寄存器名	简称	R/W	初始值	地址	存取长度
2	定时器控制寄存器 _2	TCR_2	R/W	H'00	H'FFFE4000	8
	定时器模式寄存器 _2	TMDR_2	R/W	H'00	H'FFFE4001	8
	定时器 I/O 控制寄存器 _2	TIOR_2	R/W	H'00	H'FFFE4002	8
	定时器中断允许寄存器 _2	TIER_2	R/W	H'00	H'FFFE4004	8
	定时器状态寄存器 _2	TSR_2	R/W	H'C0	H'FFFE4005	8
	定时器计数器 _2	TCNT_2	R/W	H'0000	H'FFFE4006	16
	定时器通用寄存器 A_2	TGRA_2	R/W	H'FFFF	H'FFFE4008	16
	定时器通用寄存器 B_2	TGRB_2	R/W	H'FFFF	H'FFFE400A	16
3	定时器控制寄存器 _3	TCR_3	R/W	H'00	H'FFFE4200	8
	定时器模式寄存器 _3	TMDR_3	R/W	H'00	H'FFFE4202	8
	定时器 I/O 控制寄存器 H_3	TIORH_3	R/W	H'00	H'FFFE4204	8
	定时器 I/O 控制寄存器 L_3	TIORL_3	R/W	H'00	H'FFFE4205	8
	定时器中断允许寄存器 _3	TIER_3	R/W	H'00	H'FFFE4208	8
	定时器状态寄存器 _3	TSR_3	R/W	H'C0	H'FFFE422C	8
	定时器计数器 _3	TCNT_3	R/W	H'0000	H'FFFE4210	16
	定时器通用寄存器 A_3	TGRA_3	R/W	H'FFFF	H'FFFE4218	16
	定时器通用寄存器 B_3	TGRB_3	R/W	H'FFFF	H'FFFE421A	16
	定时器通用寄存器 C_3	TGRC_3	R/W	H'FFFF	H'FFFE4224	16
	定时器通用寄存器 D_3	TGRD_3	R/W	H'FFFF	H'FFFE4226	16
	定时器缓冲运行传送模式寄存器 _3	TBTM_3	R/W	H'00	H'FFFE4238	8
4	定时器控制寄存器 _4	TCR_4	R/W	H'00	H'FFFE4201	8
	定时器模式寄存器 _4	TMDR_4	R/W	H'00	H'FFFE4203	8
	定时器 I/O 控制寄存器 H_4	TIORH_4	R/W	H'00	H'FFFE4206	8
	定时器 I/O 控制寄存器 L_4	TIORL_4	R/W	H'00	H'FFFE4207	8
	定时器中断允许寄存器 _4	TIER_4	R/W	H'00	H'FFFE4209	8
	定时器状态寄存器 _4	TSR_4	R/W	H'C0	H'FFFE422D	8
	定时器计数器 _4	TCNT_4	R/W	H'0000	H'FFFE4212	16
	定时器通用寄存器 A_4	TGRA_4	R/W	H'FFFF	H'FFFE421C	16
	定时器通用寄存器 B_4	TGRB_4	R/W	H'FFFF	H'FFFE421E	16
	定时器通用寄存器 C_4	TGRC_4	R/W	H'FFFF	H'FFFE4228	16
	定时器通用寄存器 D_4	TGRD_4	R/W	H'FFFF	H'FFFE422A	16
	定时器缓冲运行传送模式寄存器 _4	TBTM_4	R/W	H'00	H'FFFE4239	8

通道	寄存器名	简称	R/W	初始值	地址	存取长度
4	定时器 A/D 转换开始请求控制寄存器	TADCR	R/W	H'0000	H'FFFE4240	16
	定时器 A/D 转换开始请求周期设定寄存器 A_4	TADCORA_4	R/W	H'FFFF	H'FFFE4244	16
	定时器 A/D 转换开始请求周期设定寄存器 B_4	TADCORB_4	R/W	H'FFFF	H'FFFE4246	16
	定时器 A/D 转换开始请求周期设定缓冲寄存器 A_4	TADCOBRA_4	R/W	H'FFFF	H'FFFE4248	16
	定时器 A/D 转换开始请求周期设定缓冲寄存器 B_4	TADCOBRB_4	R/W	H'FFFF	H'FFFE424A	16
通用	定时器启动寄存器	TSTR	R/W	H'00	H'FFFE4280	8
	定时器同步寄存器	TSYR	R/W	H'00	H'FFFE4281	8
	定时器读 / 写允许寄存器	TRWER	R/W	H'01	H'FFFE4284	8
3/4 通用	定时器输出主控器允许寄存器	TOER	R/W	H'C0	H'FFFE420A	8
	定时器输出控制寄存器 1	TOCR1	R/W	H'00	H'FFFE420E	8
	定时器输出控制寄存器 2	TOCR2	R/W	H'00	H'FFFE420F	8
	定时器门控制寄存器	TGCR	R/W	H80	H'FFFE420D	8
	定时器周期数据寄存器	TCDR	R/W	H'FFFF	H'FFFE4214	16
	定时器空载时间数据寄存器	TDDR	R/W	H'FFFF	H'FFFE4216	16
	定时器子计数器	TCNTS	R	H'0000	H'FFFE4220	16
	定时器周期缓冲寄存器	TCBR	R/W	H'FFFF	H'FFFE4222	16
	定时器跳过中断设定寄存器	TITCR	R/W	H'00	H'FFFE4230	8
	定时器跳过中断次数计数器	TITCNT	R	H'00	H'FFFE4231	8
	定时器缓冲传送设定寄存器	TBTER	R/W	H'00	H'FFFE4232	8
	定时器空载时间允许寄存器	TDER	R/W	H'01	H'FFFE4234	8
	定时器同步清除寄存器	TSYCR	R/W	H'00	H'FFFE4250	8
	定时器波形控制寄存器	TWCR	R/W	H'00	H'FFFE4260	8
	定时器输出电平缓冲寄存器	TOLBR	R/W	H'00	H'FFFE4236	8

11.3.1　定时器控制寄存器（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，共 5 个 TCR。请在停止 TCNT 运行的状态下，进行 TCR 的设定。

位：	7	6	5	4	3	2	1	0
	CCLR[2:0]			CKEG[1:0]		TPSC[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 ～ 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、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：Pϕ/4 的双边沿 =Pϕ/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 Pϕ/4 或者慢于 Pϕ/4 时有效。如果给输入时钟选择 Pϕ/1 或者其他通道的上溢 / 下溢，虽然能写值，但运行时为初始值。 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	定时器预分频器 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 11.9。

【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	禁止清除 TCNT
	0	0	1	通过 TGRA 的比较匹配 / 输入捕捉来清除 TCNT
	0	1	0	通过 TGRB 的比较匹配 / 输入捕捉来清除 TCNT
	0	1	1	通过正在进行同步清除 / 同步运行的其他通道的计数器清除，来清除 TCNT*1
	1	0	0	禁止清除 TCNT
	1	0	1	通过 TGRC 的比较匹配 / 输入捕捉来清除 TCNT*2
	1	1	0	通过 TGRD 的比较匹配 / 输入捕捉来清除 TCNT*2
	1	1	1	通过正在进行同步清除 / 同步运行的其他通道的计数器清除，来清除 TCNT*1

【注】 *1　通过将 TSYR 的 SYNC 位置 1 来设定同步运行。

*2　在将 TGRC 或者 TGRD 用作缓冲寄存器时，优先设定缓冲寄存器。为了不发生比较匹配 / 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	禁止清除 TCNT
	0	0	1	通过 TGRA 的比较匹配 / 输入捕捉来清除 TCNT
	0	1	0	通过 TGRB 的比较匹配 / 输入捕捉来清除 TCNT
	0	1	1	通过正在进行同步清除 / 同步运行的其他通道的计数器清除，来清除 TCNT*1

【注】 *1　通过将 TSYR 的 SYNC 位置 1 来设定同步运行。

*2　在通道 1、2 中，bit7 为保留位。读时值总是 0。写无效。

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 Pϕ/1 进行计数
	0	0	1	内部时钟：通过 Pϕ/4 进行计数
	0	1	0	内部时钟：通过 Pϕ/16 进行计数
	0	1	1	内部时钟：通过 Pϕ/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 Pϕ/1 进行计数
	0	0	1	内部时钟：通过 Pϕ/4 进行计数
	0	1	0	内部时钟：通过 Pϕ/16 进行计数
	0	1	1	内部时钟：通过 Pϕ/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 Pϕ/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，该设定无效。

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 Pϕ/1 进行计数
	0	0	1	内部时钟：通过 Pϕ/4 进行计数
	0	1	0	内部时钟：通过 Pϕ/16 进行计数
	0	1	1	内部时钟：通过 Pϕ/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 Pϕ/1024 进行计数

【注】 在通道 2 为相位计数模式时，该设定无效。

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 Pϕ/1 进行计数
	0	0	1	内部时钟：通过 Pϕ/4 进行计数
	0	1	0	内部时钟：通过 Pϕ/16 进行计数
	0	1	1	内部时钟：通过 Pϕ/64 进行计数
	1	0	0	内部时钟：通过 Pϕ/256 进行计数
	1	0	1	内部时钟：通过 Pϕ/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

11.3.2　定时器模式寄存器 （TMDR）

TMDR 是 8 位可读 / 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 5 个 TMDR。请在停止 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	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	R	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，该位为保留位。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。将 TGRD 用作缓冲寄存器时，除了互补 PWM 模式以外，不发生 TGRD 的输入捕捉 / 输出比较。但在互补 PWM 模式中发生 TGRD 的比较匹配。另外，在互补 PWM 模式的 Tb 区间发生比较匹配时，TGRD 被置位，所以请将定时器中断允许寄存器 3/4 （TIER_3/4）的 TGIED 位置 0。 在没有 TGRD 的通道 1、 2 中该位为保留位。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。将 TGRC 用作缓冲寄存器时，除了互补 PWM 模式以外，不发生 TGRC 的输入捕捉 / 输出比较。但在互补 PWM 模式中发生 TGRC 的比较匹配。另外，在互补 PWM 模式的 Tb 区间发生通道 4 的比较匹配时，TGFC 被置位，所以请将定时器中断允许寄存器 _4 （TIER_4）的 TGIEC 位置 0。 在没有 TGRC 的通道 1、 2 中，该位为保留位。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 11.10。

表 11.10　通过 MD3 ～ MD0 位设定运行模式

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　不能在通道 3、 4 设定 PWM 模式 2。

*2　不能在通道 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。
对通道3设定复位同步PWM模式或者互补PWM模式时，通道4的设定无效，自动服从通道3的设定。但不能对通道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 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，共 8 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。

将 TGRC 或 TGRD 设定为缓冲运行时，该设定无效，作为缓冲寄存器运行。

- TIORH_0、 TIOR_1、 TIOR_2、 TIORH_3、 TIORH_4

位：	7	6	5	4	3	2	1	0
	IOB[3:0]				IOA[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 ～ 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 11.11　TIOR_1：表 11.13　TIOR_2：表 11.14 TIORH_3：表 11.15　TIORH_4：表 11.17
3 ～ 0	IOA[3:0]	0000	R/W	I/O 控制 A3 ～ A0 IOA3 ～ IOA0 位设定 TGRA 的功能。 请参照下述表。 TIORH_0：表 11.19　TIOR_1：表 11.21　TIOR_2：表 11.22 TIORH_3：表 11.23　TIORH_4：表 11.25

- TIORL_0、 TIORL_3、 TIORL_4

位：	7	6	5	4	3	2	1	0
	IOD[3:0]				IOC[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 ～ 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 11.12　TIORL_3：表 11.16　TIORL_4：表 11.18
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 11.20　TIORL_3：表 11.24　TIORL_4：表 11.26

表 11.11　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	输出捕捉寄存器	在上升沿进行输出捕捉
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 11.12　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：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	在上升沿进行输出捕捉
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　将 TMDR_0 的 BFB 位置 1， TGRD_0 用作缓冲寄存器时，本设定无效并且不发生输入捕捉 / 输出比较。

表 11.13　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	在上升沿进行输出捕捉
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 11.14　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	在上升沿进行输出捕捉
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 11.15　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	在上升沿进行输出捕捉
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　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	在上升沿进行输出捕捉
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　将 TMDR_3 的 BFB 位置 1， TGRD_3 用作缓冲寄存器时，本设定无效并且不发生输入捕捉 / 输出比较。

表 11.17　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	在上升沿进行输出捕捉
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 11.18　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	在上升沿进行输出捕捉
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　将 TMDR_4 的 BFB 位置 1， TGRD_4 用作缓冲寄存器时，本设定无效并且不发生输入捕捉 / 输出比较。

表 11.19　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	在上升沿进行输出捕捉
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 11.20　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	在上升沿进行输出捕捉
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　将 TMDR_0 的 BFA 位置 1， TGRC_0 用作缓冲寄存器时，本设定就无效并且不发生输入捕捉 / 输出比较。

表 11.21　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	在上升沿进行输出捕捉
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 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 11.22　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	在上升沿进行输出捕捉
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 11.23　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	在上升沿进行输出捕捉
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　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	在上升沿进行输出捕捉
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　将 TMDR_3 的 BFA 位置 1， TGRC_3 用作缓冲寄存器时，本设定就无效并且不发生输入捕捉 / 输出比较。

表 11.25　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	在上升沿进行输出捕捉
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 11.26　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	在上升沿进行输出捕捉
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　将 TMDR_4 的 BFA 位置 1， TGRC_4 用作缓冲寄存器时，本设定就无效并且不发生输入捕捉 / 输出比较。

11.3.4　定时器中断允许寄存器 （TIER）

TIER 是 8 位可读 / 写寄存器，允许或禁止各通道的中断请求。在 MTU2 中，通道 0 有 2 个，通道 1 ～ 4 各有 1 个，共 6 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：禁止产生 A/D 转换开始请求 1：允许产生 A/D 转换开始请求
6	TTGE2	0	R/W	A/D 转换开始请求的允许 2 在互补 PWM 模式中，允许或禁止因 TCNT_4 的下溢 （波谷）产生的 A/D 转换请求。在通道 0 ～ 3 中，该位为保留位。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	允许下溢中断 在通道 1、 2 中，当 TSR 的 TCFU 标志被置 1 时，允许或禁止 TCFU 标志的中断请求 （TCIU）。在通道 0、 3、 4 中，该位为保留位。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	允许上溢中断 当 TSR 的 TCFV 标志被置 1 时，允许或禁止 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）

位	位名	初始值	R/W	说　　明
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，允许或禁止 TGFD 位的中断请求 （TGID）。在通道 1、 2 中，该位为保留位。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，允许或禁止 TGFC 位的中断请求 （TGIC）。在通道 1、 2 中，该位为保留位。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）
1	TGIEB	0	R/W	TGR 中断允许 B 当 TSR 的 TGFB 位被置 1 时，允许或禁止 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	—	均为 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）

11.3.5　定时器状态寄存器（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 ～ 4 各有 1 个，共 6 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，该位为保留位。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 是表示通道 1、2 为相位计数模式时，发生 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，该位为保留位。读写的值总是 0。 ［清除条件］ • 在 TCFU=1 的状态下，当读 TCFU 后，给 TCFU 写 0 时 *2 ［置位条件］ • 当 TCNT 的值发生下溢（H'0000→H'FFFF）时
4	TCFV	0	R/(W)*1	上溢标志 • 是表示发生 TCNT 上溢的状态标志。为了清除标志，只能写 0。 ［清除条件］ • 在 TCFV=1 的状态下，当读 TCFV 后，给 TCFV 写 0 时 *2 ［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。

位	位名	初始值	R/W	说　　明
3	TGFD	0	R/(W)*1	输入捕捉 / 输出比较标志 D 是表示通道 0、3、4 的 TGRD 发生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，该位为保留位。读写的值总是 0。 ［清除条件］ • 在 TGFD=1 的状态下，当读 TGFD 后，给 TGFD 写 0 时 *2 ［置位条件］ • 在 TGRD 用作输出比较寄存器的情况下，当 TCNT=TGRD 时 • 在 TGRD 用作输入捕捉寄存器的情况下，当通过输入捕捉信号将 TCNT 的值传送到 TGRD 时
2	TGFC	0	R/(W)*1	输入捕捉 / 输出比较标志 C 是表示通道 0、3、4 的 TGRC 发生输入捕捉或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，该位为保留位。读写的值总是 0。 ［清除条件］ • 在 TGFC=1 的状态下，当读 TGFC 后，给 TGFC 写 0 时 *2 ［置位条件］ • 在 TGRC 用作输出比较寄存器的情况下，当 TCNT=TGRC 时 • 在 TGRC 用作输入捕捉寄存器的情况下，当通过输入捕捉信号将 TCNT 的值传送到 TGRC 时
1	TGFB	0	R/(W)*1	输入捕捉 / 输出比较标志 B 是表示 TGRB 发生输入捕捉或比较匹配的状态标志。为了清除标志，只能写 0。 ［清除条件］ • 在 TGFB=1 的状态下，当读 TGFB 后，给 TGFB 写 0 时 *2 ［置位条件］ • 在 TGRB 用作输出比较寄存器的情况下，当 TCNT=TGRB 时 • 在 TGRB 用作输入捕捉寄存器的情况下，当通过输入捕捉信号将 TCNT 的值传送到 TGRB 时
0	TGFA	0	R/(W)*1	输入捕捉 / 输出比较标志 A 是表示 TGRA 发生输入捕捉或比较匹配的状态标志。为了清除标志，只能写 0。 ［清除条件］ • 在通过 TGIA 中断启动 DMAC 时 • 在 TGFA=1 的状态下，当读 TGFA 后，给 TGFA 写 0 时 *2 ［置位条件］ • 在 TGRA 用作输出比较寄存器的情况下，当 TCNT=TGRA 时 • 在 TGRA 用作输入捕捉寄存器的情况下，当通过输入捕捉信号将 TCNT 的值传送到 TGRA 时

【注】 *1　为了清除标志，只能在读 1 后写 0。

*2　定时器状态寄存器 （TSR）的写操作，读 1 后只能对需要清除的位写 0，对其他位写 1。但实际并不写 1，而保留前面的值。

- 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)*1	R/(W)*1

【注】*1　为清除标志，只能在读1后写0。

位	位名	初始值	R/W	说　　明
7、6	—	均为 1	R	保留位 读写的值总是 1。
5 ～ 2	—	均为 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 发生比较匹配的状态标志。 ［清除条件］ • 在 TGFF=1 的状态下，当读 TGFF 后，给 TGFF 写 0 时 *2 ［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
0	TGFE	0	R/(W)*1	比较匹配标志 E 是表示 TCNT_0 和 TGRE_0 发生比较匹配的状态标志。 ［清除条件］ • 在 TGFE=1 的状态下，当读 TGFE 后，给 TGFE 写 0 时 *2 ［置位条件］ • 在将 TGRE_0 用作比较寄存器的情况下，当 TCNT_0=TGRE_0 时

【注】 *1　为了清除标志，只能在读 1 后写 0。

*2　定时器状态寄存器 （TSR）的写操作，读 1 后只能对需要清除的位写 0，对其他位写 1。但实际并不写 1，而保留前面的值。

11.3.6　定时器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通用寄存器的传送时序。在 MTU2 中，通道 0、3、4 各有 1 个，共 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	—	均为 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，该位为保留位。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：当通道 0 发生比较匹配 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，对于在 PWM 以外的模式中使用的通道，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，对于在 PWM 以外的模式中使用的通道，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

11.3.7　定时器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	—	均为 0	R	保留位 读写的值总是 0。
3	I2BE	0	R/W	允许输入捕捉 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	允许输入捕捉 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	允许输入捕捉 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

11.3.8　定时器同步清除寄存器 （TSYCR）

TSYCR 是 8 位可读 / 写寄存器，设定从 MTU2 进行 MTU2S 的 TCNT_3 和 TCNT_4 同步清除条件。MTU2S 的通道 3 有 1 个 TSYCR。但 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	清除允许 0A 设定禁止或允许通过置位 MTU2 的 TSR_0 的 TGFA 标志进行的清除。 0：禁止通过置位 TSR_0 的 TGFA 标志进行的清除 1：允许通过置位 TSR_0 的 TGFA 标志进行的清除
6	CE0B	0	R/W	清除允许 0B 设定禁止或允许通过置位 MTU2 的 TSR_0 的 TGFB 标志进行的清除。 0：禁止通过置位 TSR_0 的 TGFB 标志进行的清除 1：允许通过置位 TSR_0 的 TGFB 标志进行的清除
5	CE0C	0	R/W	清除允许 0C 设定禁止或允许通过置位 MTU2 的 TSR_0 的 TGFC 标志进行的清除。 0：禁止通过置位 TSR_0 的 TGFC 标志进行的清除 1：允许通过置位 TSR_0 的 TGFC 标志进行的清除
4	CE0D	0	R/W	清除允许 0D 设定禁止或允许通过置位 MTU2 的 TSR_0 的 TGFD 标志进行的清除。 0：禁止通过置位 TSR_0 的 TGFD 标志进行的清除 1：允许通过置位 TSR_0 的 TGFD 标志进行的清除
3	CE1A	0	R/W	清除允许 1A 设定禁止或允许通过置位 MTU2 的 TSR_1 的 TGFA 标志进行的清除。 0：禁止通过置位 TSR_1 的 TGFA 标志进行的清除 1：允许通过置位 TSR_1 的 TGFA 标志进行的清除
2	CE1B	0	R/W	清除允许 1B 设定禁止或允许通过置位 MTU2 的 TSR_1 的 TGFB 标志进行的清除。 0：禁止通过置位 TSR_1 的 TGFB 标志进行的清除 1：允许通过置位 TSR_1 的 TGFB 标志进行的清除
1	CE2A	0	R/W	清除允许 2A 设定禁止或允许通过置位 MTU2 的 TSR_2 的 TGFA 标志进行的清除。 0：禁止通过置位 TSR_2 的 TGFA 标志进行的清除 1：允许通过置位 TSR_2 的 TGFA 标志进行的清除
0	CE2B	0	R/W	清除允许 2B 设定禁止或允许通过置位 MTU2 的 TSR_2 的 TGFB 标志进行的清除。 0：禁止通过置位 TSR_2 的 TGFB 标志进行的清除 1：允许通过置位 TSR_2 的 TGFB 标志进行的清除

11.3.9　定时器 A/D 转换开始请求控制寄存器 （TADCR）

TADCR 是 16 位可读 / 写寄存器，允许或禁止 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，不能设定为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 11.27。
13 ～ 8	—	均为 0	R	保留位 读写的值总是 0。
7	UT4AE	0	R/W	允许递增计数 TRG4AN 在进行 TCNT_4 的递增计数时，允许或禁止 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	允许递减计数 TRG4AN 在进行 TCNT_4 的递减计数时，允许或禁止 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	允许递增计数 TRG4BN 在进行 TCNT_4 的的递增计数时，允许或禁止 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）

位	位名	初始值	R/W	说　　明
4	DT4BE	0*	R/W	允许递减计数 TRG4BN 在进行 TCNT_4 的递减计数时，允许或禁止 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）
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。请以 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），请设定为不与跳过中断功能联动 （将定时器 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 11.27　通过 BF1、 BF0 位设定的传送时序

bit7	bit6	说　　明
BF1	BF 0	
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 处于波峰时，或在复位同步 PWM 模式中 TCNT_3 和 TGRA_3 比较匹配时，或在 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。

11.3.10　定时器 A/D 转换开始请求周期设定寄存器 （TADCORA/B_4）

TADCORA/B_4 是 16 位可读 / 写寄存器。当 TADCORA/B_4 和 TCNT_4 匹配时，就发生对应的 A/D 转换开始请求。

TADCORA/B_4 的初始值为 H'FFFF。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
初始值：	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W：	R/W	R/W	R/W	R/W	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】 禁止以8位为单位存取TADCORA/B_4，请以16位为单位进行存取。

11.3.11　定时器 A/D 转换开始请求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。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
初始值：	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W：	R/W	R/W	R/W	R/W	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】 禁止以8位为单位存取TADCOBRA/B_4，请以16位为单位进行存取。

11.3.12　定时器计数器 （TCNT）

TCNT 是 16 位可读 / 写计数器。通道 0 ～ 4 各有 1 个，共 5 个 TCNT。

禁止以 8 位为单位存取 TCNT。请以 16 位为单位进行存取。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
初始值：	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W：	R/W	R/W	R/W	R/W	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】 禁止以8位为单位存取TCNT，请以16位为单位进行存取。

11.3.13　定时器通用寄存器 （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，共 18 个通用寄存器。

TGRA、 TGRB、 TGRC、 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 作为缓冲寄存器来设定其运行。 TGR 和缓冲寄存器的组合为 TGRA–TGRC、 TGRB–TGRD。

TGRE_0、 TGRF_0 用作比较匹配寄存器，当 TCNT_0 和 TGRE_0 匹配时，就会产生 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。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
初始值：	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W：	R/W	R/W	R/W	R/W	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】 禁止以8位为单位存取TGR，请以16位为单位进行存取。TGR初始值为H'FFFF。

11.3.14　定时器启动寄存器 （TSTR）

TSTR 是 8 位可读 / 写寄存器，选择通道 0 ～ 4 的 TCNT 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，请停止 TCNT 的计数器运行。

位：	7	6	5	4	3	2	1	0
	CST4	CST3	-	-	-	CST2	CST1	CST0
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	CST4	0	R/W	计数器启动 4、 3 选择 TCNT 的运行或者停止。 TIOC 引脚在输出状态下运行时，如果给 CST 位写 0，就停止计数器，但保持 TIOC 引脚输出比较的输出电平。如果在 CST 位为 0 的状态下对 TIOR 进行写入，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
6	CST3	0	R/W	
5 ～ 3	—	均为 0	R	保留位 读写的值总是 0。
2	CST2	0	R/W	计数器启动 2 ～ 0 选择 TCNT 的运行或者停止。 TIOC 引脚在输出状态下运行时，如果给 CST 位写 0，就停止计数器，但保持 TIOC 引脚输出比较的输出电平。如果在 CST 位为 0 的状态下对 TIOR 进行写入，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
1	CST1	0	R/W	
0	CST0	0	R/W	

11.3.15　定时器同步寄存器 （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	定时器同步 4、 3 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 位置 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
6	SYNC3	0	R/W	
5 ～ 3	—	均为 0	R	保留位 读写的值总是 0。
2	SYNC2	0	R/W	定时器同步 2 ～ 0 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 位置 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
1	SYNC1	0	R/W	
0	SYNC0	0	R/W	

11.3.16　定时器读写允许寄存器（TRWER）

TRWER 是 8 位可读 / 写寄存器。设定允许或者禁止存取通道 3、4 的误写防止对象寄存器 / 计数器。

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	—	均为 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22个寄存器。

11.3.17　定时器输出主控器允许寄存器 （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	—	均为 1	R	保留位 读写的值总是 1。
5	OE4D	0	R/W	主控器允许 TIOC4D 允许 / 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	主控器允许 TIOC4C 允许 / 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	主控器允许 TIOC3D 允许 / 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	主控器允许 TIOC4B 允许 / 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	主控器允许 TIOC4A 允许 / 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	主控器允许 TIOC3B 允许 / 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输出控制寄存器 1/2 （TOCR1/2）的设定。详细内容请参照 “11.3.18 定时器输出控制寄存器 1 （TOCR1）” 和 “11.3.19 定时器输出控制寄存器 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。

11.3.18　定时器输出控制寄存器 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	允许 PWM 同步输出 允许 / 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	—	均为 0	R	保留位 读写的值总是 0。
3	TOCL	0	R/(W)*	TOC 寄存器的写入禁止位 *1 禁止 / 允许向 TOCR1 寄存器的 TOCS 位、 OLSN 位、 OLSP 位进行写入。 0：允许向 TOCS 位、 OLSN 位、 OLSP 位进行写入 1：禁止向 TOCS 位、 OLSN 位、 OLSP 位进行写入
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 11.28。
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 11.29。

【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 11.28　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 11.29　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，互补 PWM 模式的输出例 （1 相）如图 11.2 所示。

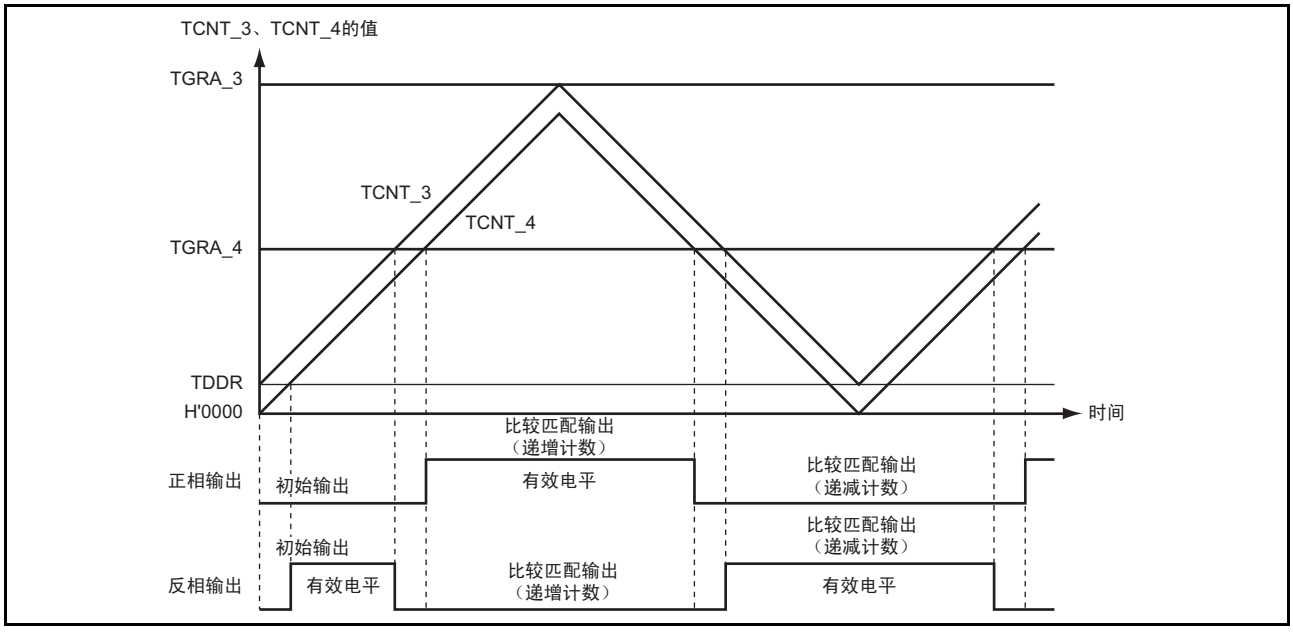


图 11.2　互补 PWM 模式的输出电平的例子

11.3.19　定时器输出控制寄存器 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 选择从 TOLBR 到 TOCR2 的缓冲传送时序。 详细内容请参照表 11.30。
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 11.31。
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 11.32。
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 11.33。
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 11.34。
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 11.35。
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 11.36。

【注】 *　通过将 TOCR1 的 TOCS 位设定为 1，本设定有效。

表 11.30　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）的数据传送到 TOCR2	不将缓冲寄存器（TOLBR）的数据传送到 TOCR2
0	1	在 TCNT_4 的波峰，将缓冲寄存器（TOLBR）的数据传送到 TOCR2	在清除 TCNT_3/4 计数器时，将缓冲寄存器（TOLBR）的数据传送到 TOCR2
1	0	在 TCNT_4 的波谷，将缓冲寄存器（TOLBR）的数据传送到 TOCR2	禁止设定
1	1	在 TCNT_4 的波峰和波谷，将缓冲寄存器（TOLBR）的数据传送到 TOCR2	禁止设定

表 11.31　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 11.32　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 11.33　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 11.34　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 11.35　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 11.36　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

11.3.20　定时器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	—	均为 0	R	保留位 读写的值总是 0。
5	OLS3N	0	R/W	请给 TOCR2 的 OLS3N 位设定缓冲传送的值。
4	OLS3P	0	R/W	请给 TOCR2 的 OLS3P 位设定缓冲传送的值。
3	OLS2N	0	R/W	请给 TOCR2 的 OLS2N 位设定缓冲传送的值。
2	OLS2P	0	R/W	请给 TOCR2 的 OLS2P 位设定缓冲传送的值。
1	OLS1N	0	R/W	请给 TOCR2 的 OLS1N 位设定缓冲传送的值。
0	OLS1P	0	R/W	请给 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，设定步骤如图 11.3 所示。

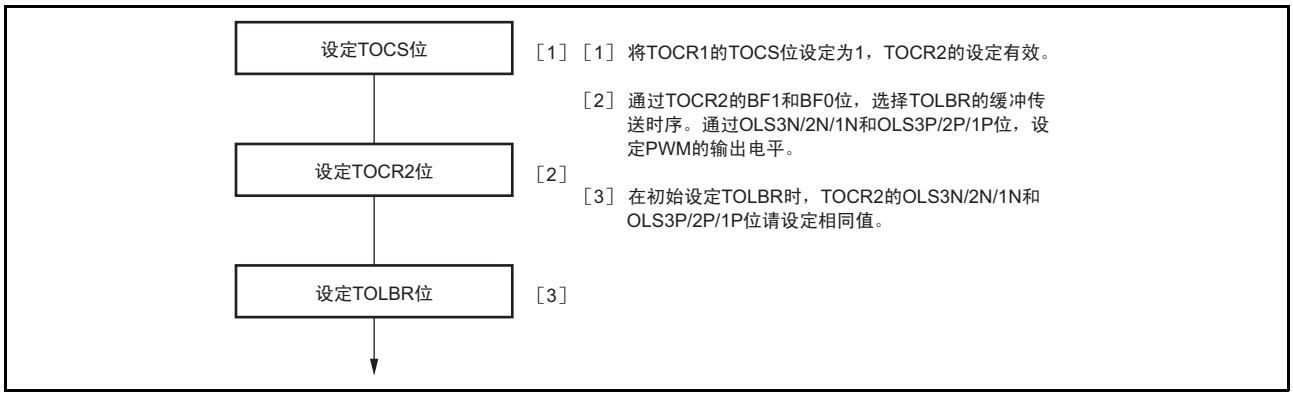


图 11.3　在缓冲运行中设定 PWM 输出电平时的设定步骤例

11.3.21　定时器门控寄存器 （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 当输出反相引脚（TIOC3D 引脚、 TIOC4C 引脚、 TIOC4D 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 当输出正相引脚（TIOC3B 引脚、 TIOC4A 引脚、 TIOC4B 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	允许外部反馈信号 选择是通过 MTU2/ 通道 0 的 TGRA、 TGRB、 TGRC 的输入捕捉信号自动进行还是通过给 TGCR 的 bit2 ～ 0 写 0 或者写 1 进行正相 / 反相的输出转换。 0：通过外部输入进行输出转换 （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 11.37。
1	VF	0	R/W	
0	UF	0	R/W	

表 11.37　输出电平的选择功能

bit2	bit1	bit0	功　　能					
WF	VF	UF	TIOC3B	TIOC4A	TIOC4B	TIOC3D	TIOC4C	TIOC4D
			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

11.3.22　定时器子计数器 （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。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
初始值：	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W：	R/W	R/W	R/W	R/W	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】 禁止以8位为单位存取TCNTS，请以16位为单位进行存取。

11.3.23　定时器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 TCNT_3 计数器并开始计数。 TDDR 的初始值为 H'FFFF。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
初始值：	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W：	R/W	R/W	R/W	R/W	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】 禁止以8位为单位存取TDDR，请以16位为单位进行存取。

11.3.24　定时器周期数据寄存器 （TCDR）

TCDR 是只在互补 PWM 模式中使用的 16 位寄存器。请将 TCDR 寄存器的值设定为 PWM 载流子周期的 1/2。本寄存器在互补 PWM 模式中随时和 TCNTS 计数器比较，如果匹配， 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。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
初始值：	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W：	R/W	R/W	R/W	R/W	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】 禁止以8位为单位存取TCDR，请以16位为单位进行存取。

11.3.25　定时器周期缓冲寄存器 （TCBR）

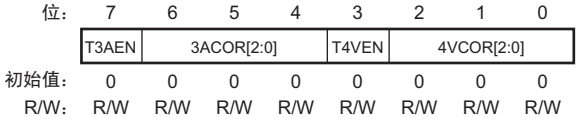
TCBR 是只在互补 PWM 模式中使用的 16 位寄存器，用作 TCDR 寄存器的缓冲寄存器。在 TMDR 寄存器设定的传送时序，将 TCBR 寄存器的值传送到 TCDR 寄存器。 TCBR 的初始值为 H'FFFF。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
初始值：	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W：	R/W	R/W	R/W	R/W	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】 禁止以8位为单位存取TCBR，请以16位为单位进行存取。

11.3.26　定时器跳过中断设定寄存器 （TITCR）

TITCR 是 8 位可读 / 写寄存器，禁止 / 允许跳过中断功能以及跳过中断次数。 MTU2 有 1 个 TITCR。



位	位名	初始值	R/W	说　　明
7	T3AEN	0	R/W	T3AEN 禁止 / 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	设定 TGIA_3 的跳过中断次数 （0 ～ 7 次）。 * 详细内容请参照表 11.38。
3	T4VEN	0	R/W	T4VEN 禁止 / 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	设定 TCIV_4 的跳过中断次数 （0 ～ 7 次）。 * 详细内容请参照表 11.39。

【注】 * 如果将跳过中断次数设定 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 11.38　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 11.39　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

11.3.27　定时器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 读取时值总是 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，就在产生 TGIA_3 中断源时进行递增计数（加 1）。 ［清除条件］ • TITCR 的 3ACOR2 ～ 3ACOR0 和 TITCNT 的 3ACNT2 ～ 3ACNT0 匹配时 • TITCR 的 T3AEN 位为 0 时 • 当 TITCR 的 3ACOR2 ～ 3ACOR0 为 0 时
3	—	0	R	保留位 读取时值总是 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，就在产生 TCIV_4 中断源时进行递增计数（加 1）。 ［清除条件］ • 当 TITCR 的 4VCOR2 ～ 4VCOR0 和 TITCNT 的 4VCNT2 ～ 4VCNT0 匹配时 • 当 TITCR 的 T4VEN 位为 0 时 • 当 TITCR 的 4VCOR2 ～ 4VCOR0 为 0 时

【注】 要清除 TITCNT 的值时，请将 TITCR 的 T3AEN 位和 T4VEN 位清 0。

11.3.28　定时器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	—	均为 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 *（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。 详细内容请参照表 11.40。

【注】 * 对象缓冲寄存器
TGRC_3、 TGRD_3、 TGRC_4、 TGRD_4、 TCBR

表 11.40　BTE1、 BTE0 位的设定

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、 4VCOR）设定为 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 设定为 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。

11.3.29　定时器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，能控制生成互补 PWM 模式的空载时间。 MTU2 有 1 个 TDER。请在 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	—	均为 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

【注】 * 请设定为 TDDR ≥ 1。

11.3.30　定时器波形控制寄存器 （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控制，及是否进行由 TGRA_3 比较匹配引起的计数器清除。请在停止 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	-	-	-	-	-	-	WRE
初始值：	0*	0	0	0	0	0	0	0
R/W：	R/(W)	R	R	R	R	R	R	R/(W)

【注】* 除了互补PWM模式1以外，不能设定为1。

位	位名	初始值	R/W	说　　明
7	CCE	0*	R/(W)	允许比较匹配清除 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。 0：不进行由 TGRA_3 比较匹配引起的计数器清除 1：进行由 TGRA_3 比较匹配引起的计数器清除 ［置位条件］ • CCE=0 的状态下，当读 CCE 后，给 CCE 写 1 时
6 ～ 1	—	均为 0	R	保留位 读写的值总是 0。
0	WRE	0	R/(W)	允许波形保持 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 刚启动的波谷 Tb 区间发生同步清除，也将输出 TOCR 寄存器设定的初始值。 有关互补 PWM 模式波谷的 Tb 区间，请参照图 11.40。 0：输出 TOCR 寄存器设定的初始输出值 1：保持同步清除前的波形 ［置位条件］ • WRE=0 的状态下，当读 WRE 后，给 WRE 写 1 时

【注】 * 当不是互补 PWM 模式 1 时，请勿设定为 1。

11.3.31　和总线主控器的接口

定时器计数器 （TCNT） 、通用寄存器 （TGR） 、定时器子计数器 （TCNTS） 、定时器周期缓冲寄存器 （TCBR）、定时器空载时间数据寄存器 （TDDR）、定时器周期数据寄存器 （TCDR）、定时器 A/D 转换开始请求控制寄存器 （TADCR）、定时器 A/D 转换开始请求周期设定寄存器 （TADCOR）以及定时器 A/D 转换开始请求周期设定缓冲寄存器 （TADCOBR）是 16 位寄存器。因为和总线主控器之间的数据总线为 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。请以 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 位置 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。

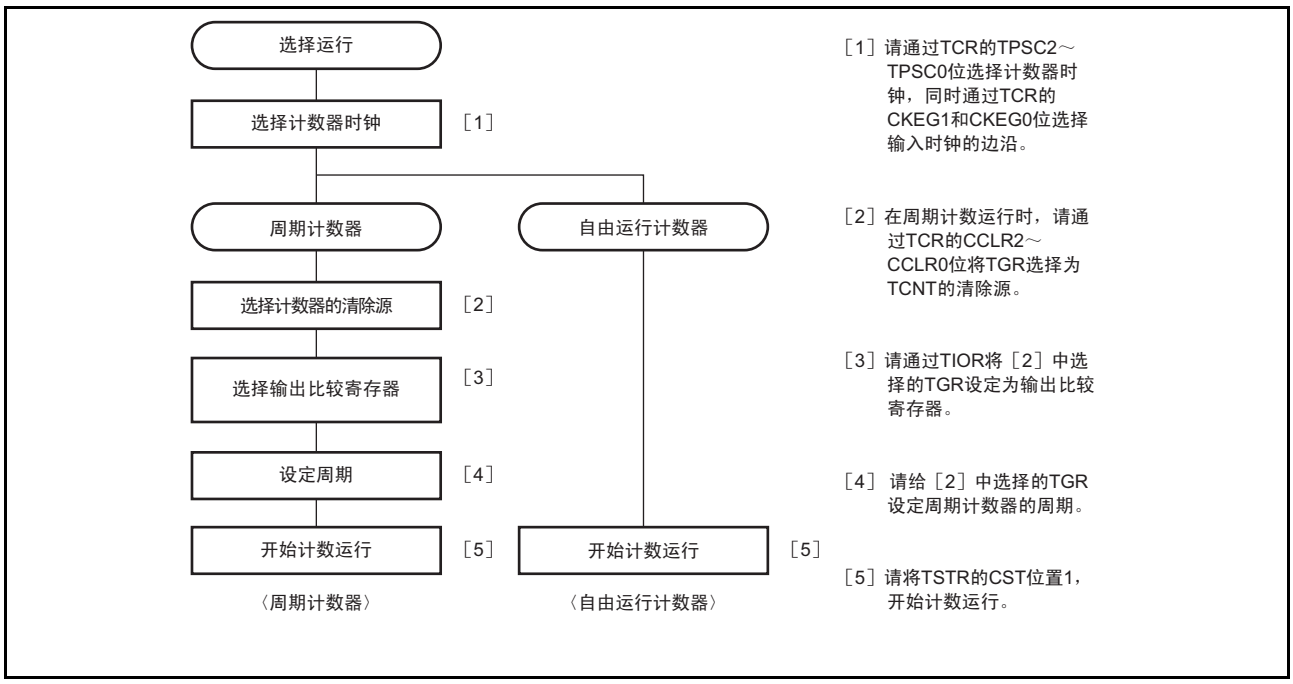


图 11.4　计数器运行的设定步骤例子

（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。

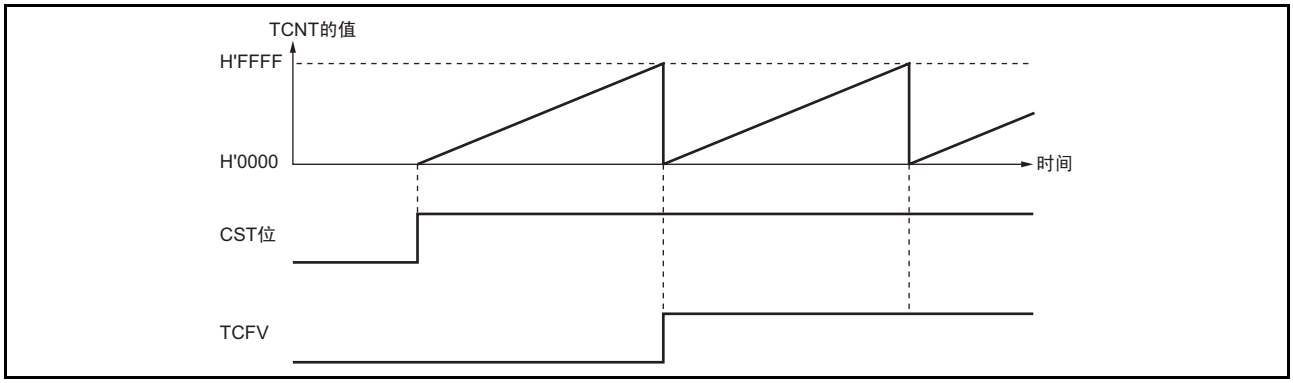


图 11.5　自由运行计数器的运行

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 位选择比较匹配引起的计数器清除。在设定后，如果将 TSTR 的对应位置 1，就作为周期计数器开始递增计数运行。当计数值和 TGR 的值匹配时， TSR 的 TGF 位就被置 1， TCNT 被清除为 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。

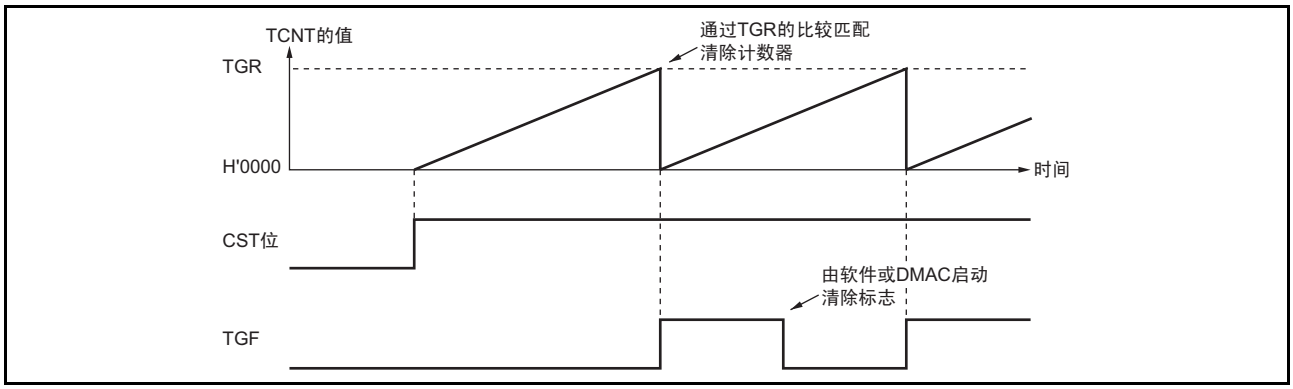


图 11.6　周期计数器的运行

（2） 通过比较匹配进行波形输出的功能

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。

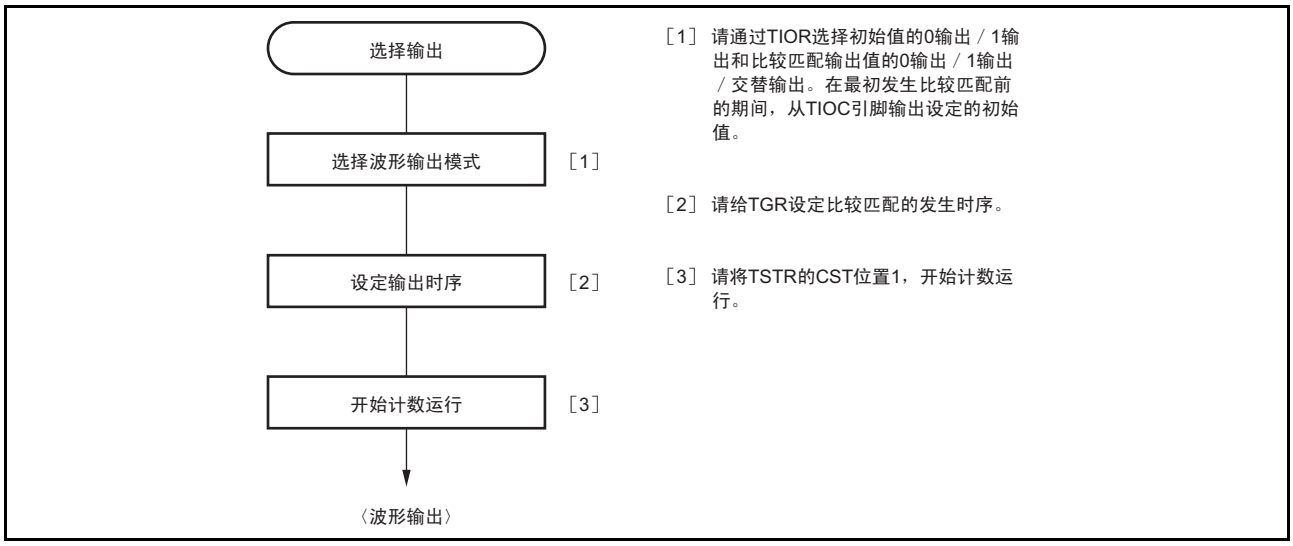


图 11.7　通过比较匹配进行波形输出的运行例子

（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 作为自由运行计数器运行，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匹配时，引脚的电平不变。

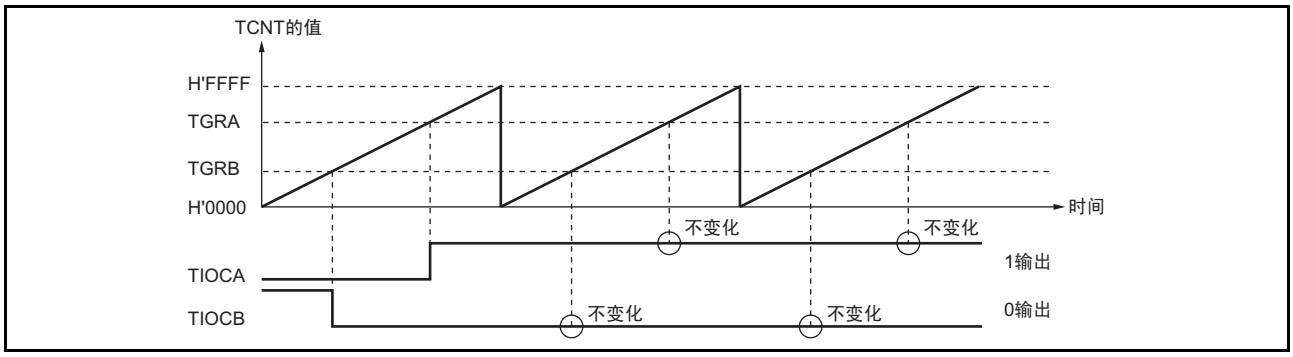


图 11.8　0 输出 /1 输出的运行例子

交替输出的例子如图 11.9 所示。

在此例中，将 TCNT 作为周期计数器运行（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。

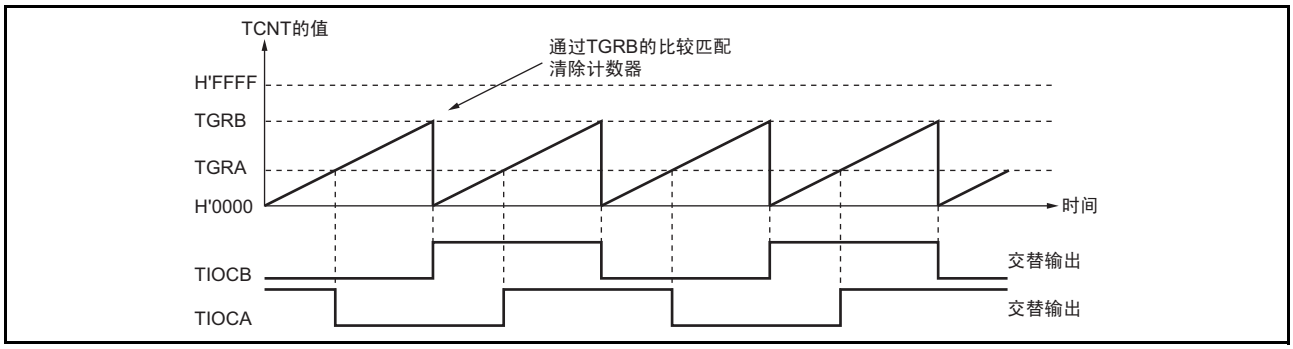


图 11.9　交替输出的运行例子

（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出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输入时，就不能选择 Pϕ/1 作为输入捕捉输入的计数器输入时钟。如果选择 Pϕ/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。

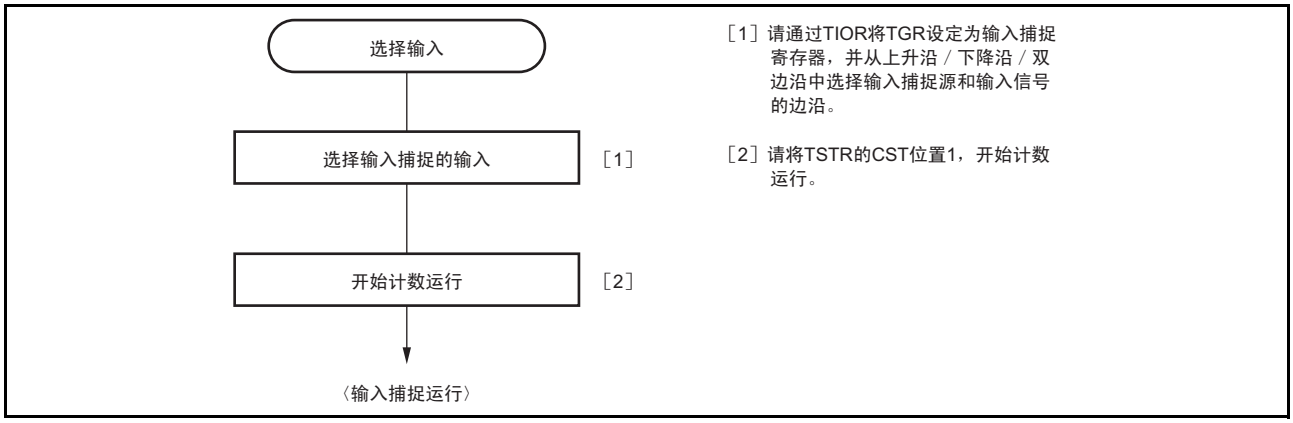


图 11.10　输入捕捉运行的设定例子

（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。

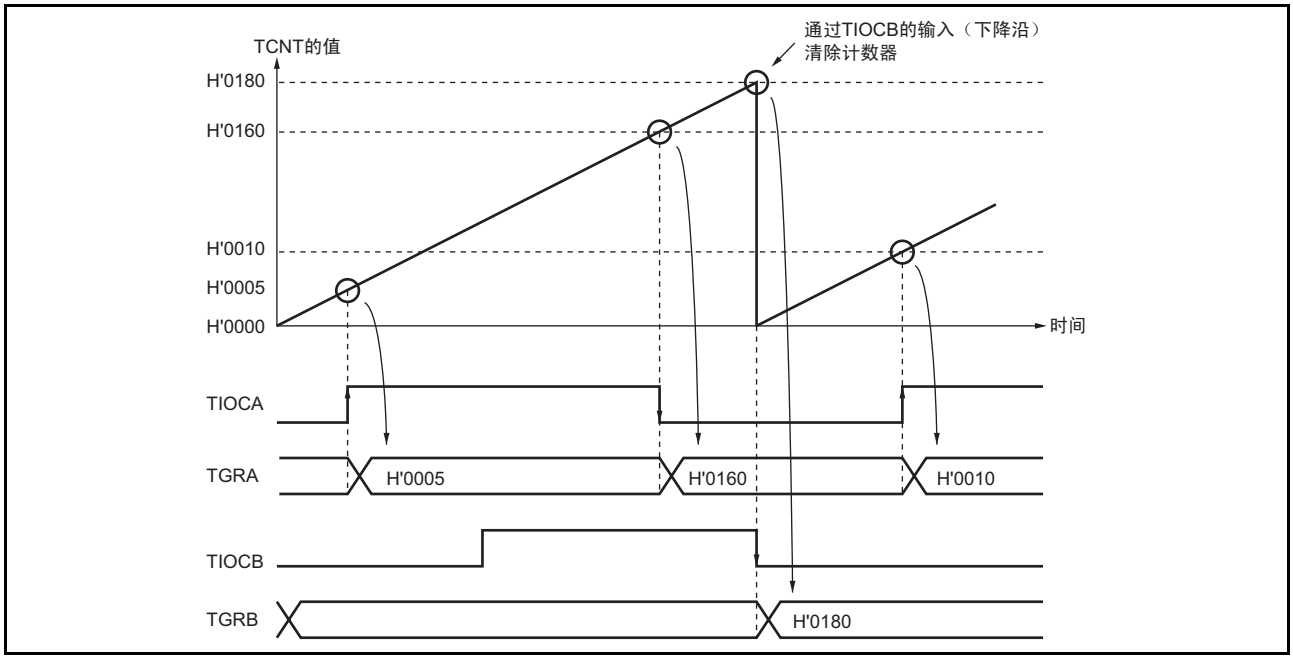


图 11.11　输入捕捉的运行例子

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。

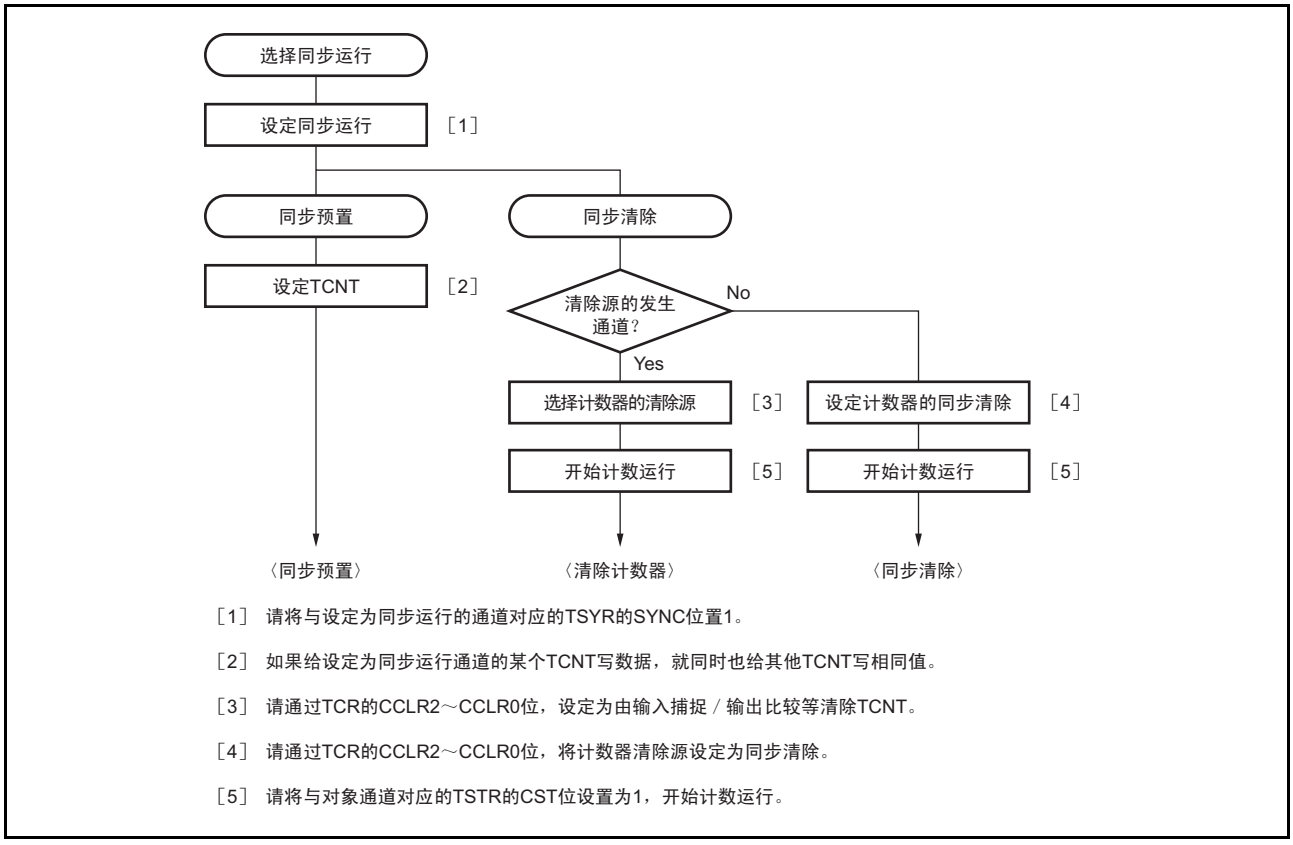


图 11.12　同步运行的设定步骤例子

（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”。

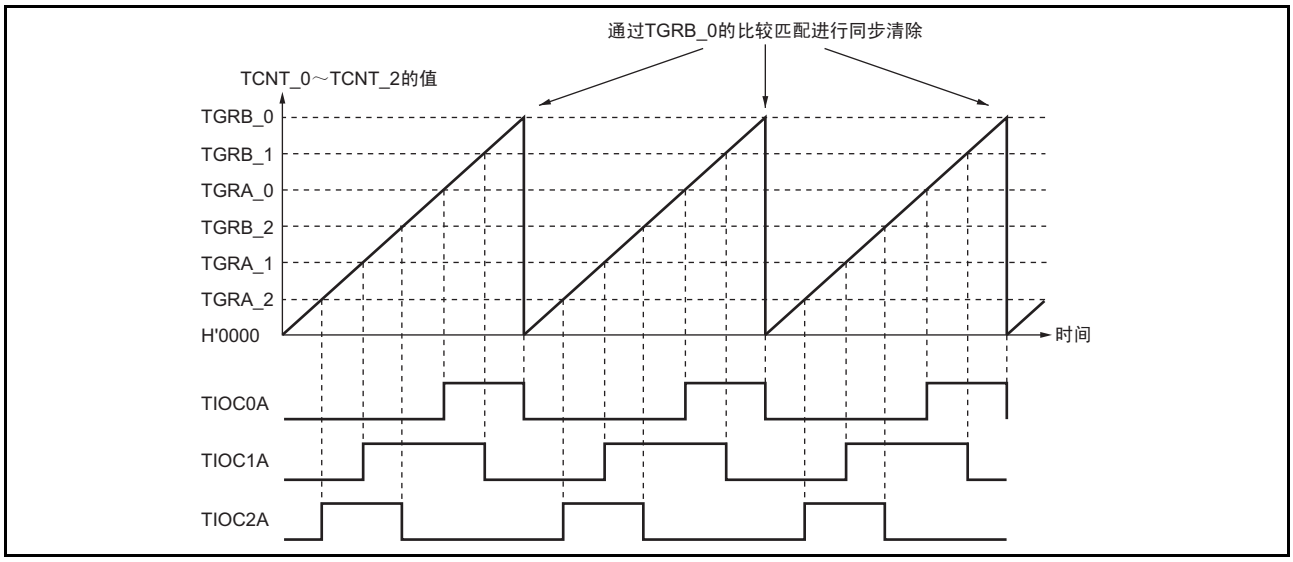


图 11.13　同步运行的例子

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。

TGR 设定为输入捕捉寄存器时与设定为比较匹配寄存器时，缓冲运行的内容有所不同。

【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 11.41 所示。

表 11.41　寄存器的组合

通道	定时器通用寄存器	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 为输出比较寄存器时
  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。

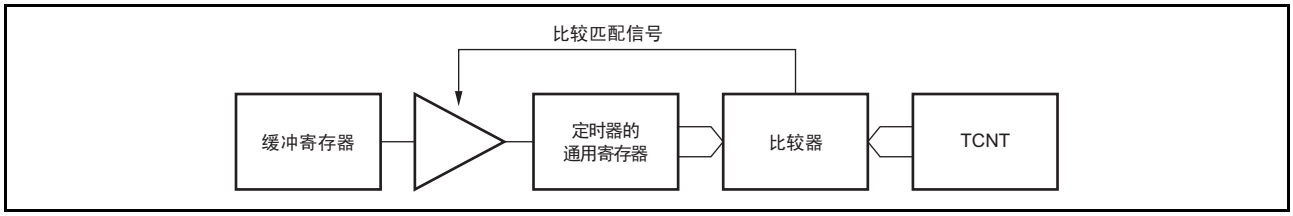


图 11.14　比较匹配的缓冲运行

- TGR 为输入捕捉寄存器时
  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。

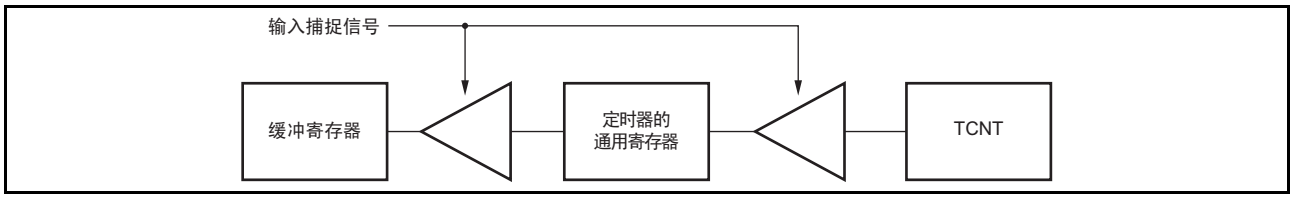


图 11.15　输入捕捉的缓冲运行

（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。

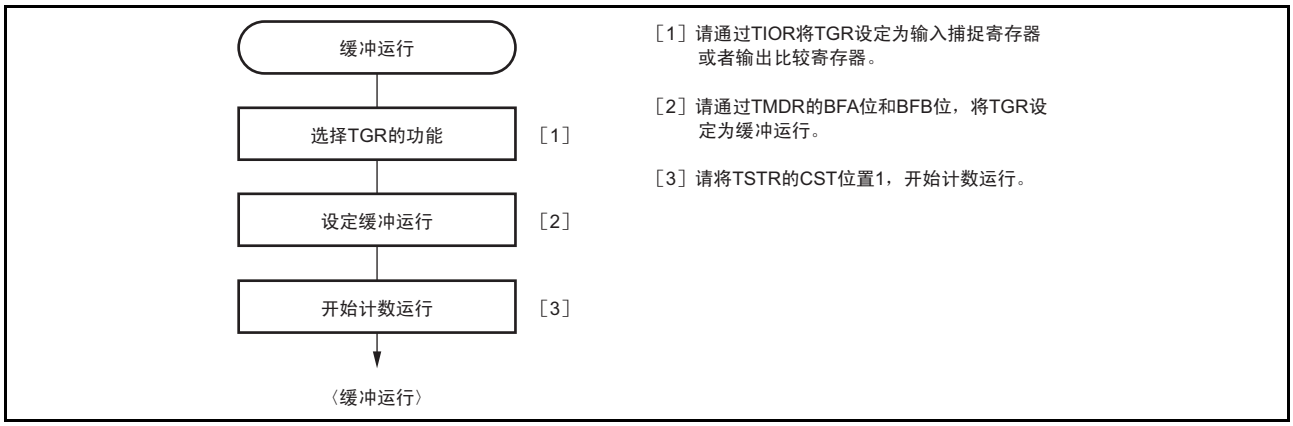


图 11.16　缓冲运行的设定步骤例子

（2） 缓冲运行的例子

（a） TGR 为输出比较寄存器时

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”。

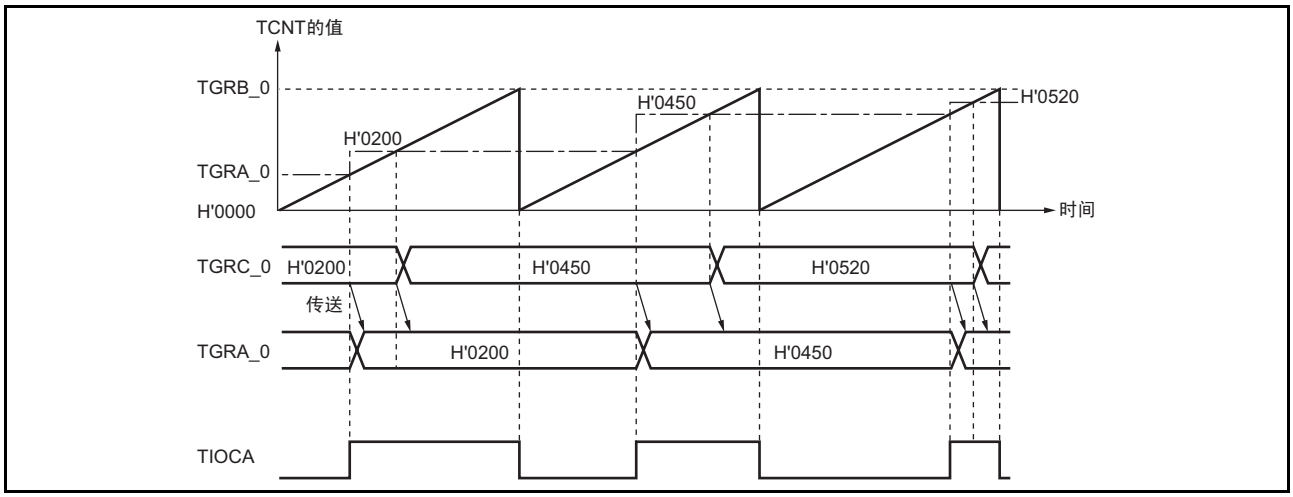


图 11.17　缓冲运行的例子 （1）

（b） TGR 为输入捕捉寄存器时

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。

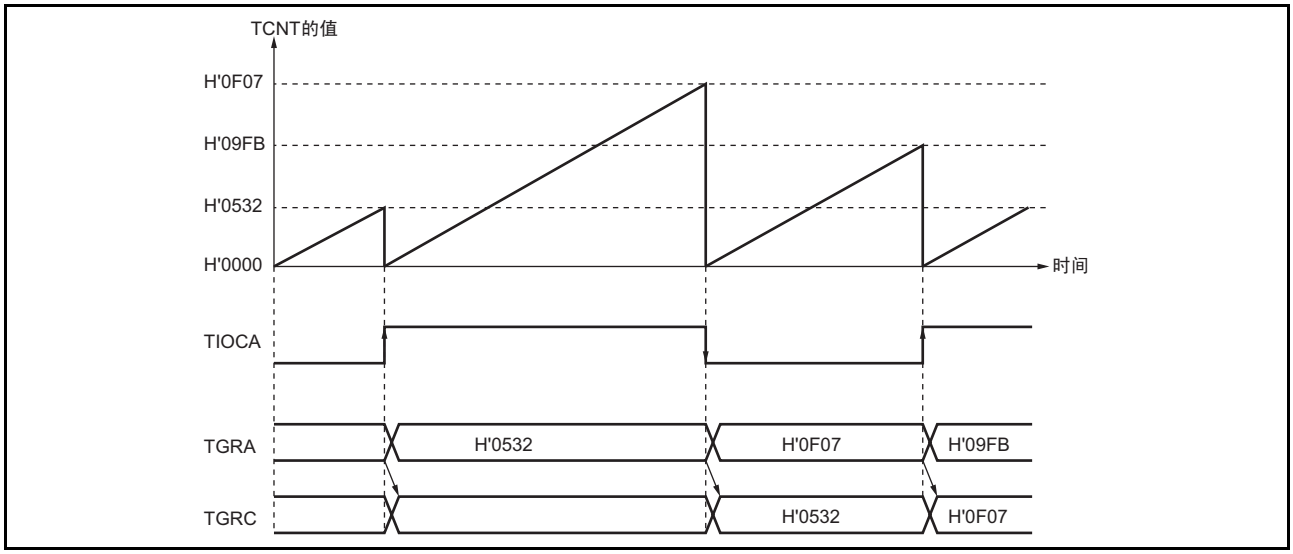


图 11.18　缓冲运行的例子 （2）

（3） 选择缓冲运行时从缓冲寄存器到定时器通用寄存器的传送时序

能通过设定缓冲运行传送模式寄存器（TBTM_0、TBTM_3、TBTM_4），选择缓冲寄存器到定时器通用寄存器的传送时序（通道 0 为 PWM 模式 1、2，通道 3、4 为 PWM 模式 1）。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清除为H'0000时

【注】　请在停止 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。

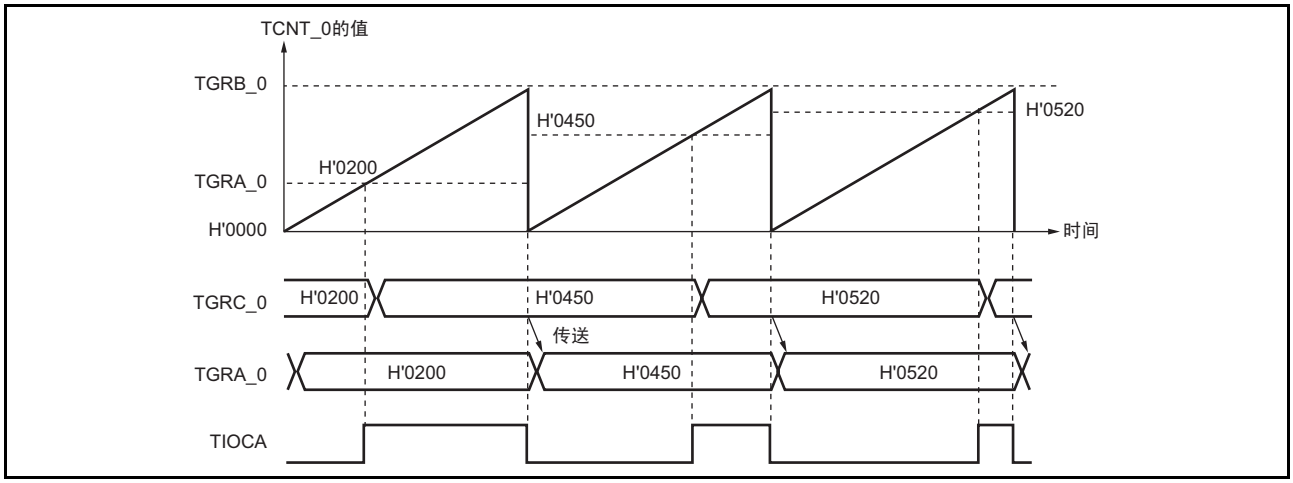


图 11.19　将 TGRC_0 到 TGRA_0 的缓冲传送时序选择为清除 TCNT_0 时的运行例子

11.4.4　级联运行

级联运行是将 2 个通道的 16 位计数器连接作为 32 位计数器运行的功能。

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”，来运行此功能。

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 11.42 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 11.42　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，追加输入引脚（输入引脚为输入捕捉条件）。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 设定值和输入捕捉输入引脚的对应关系如**表 11.43** 所示。

表 11.43　TICCR 设定值和输入捕捉输入引脚的对应关系

对象输入捕捉	TICCR 设定值	输入捕捉输入引脚
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。

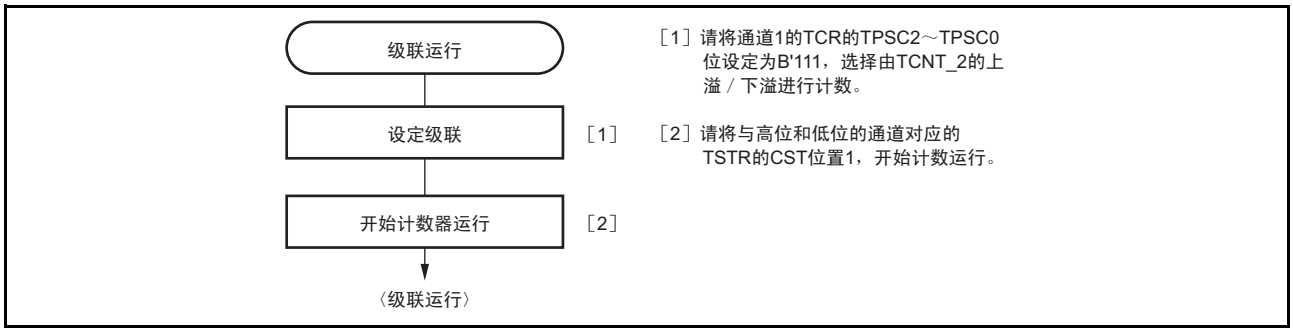


图 11.20　级联运行的设定步骤

（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。

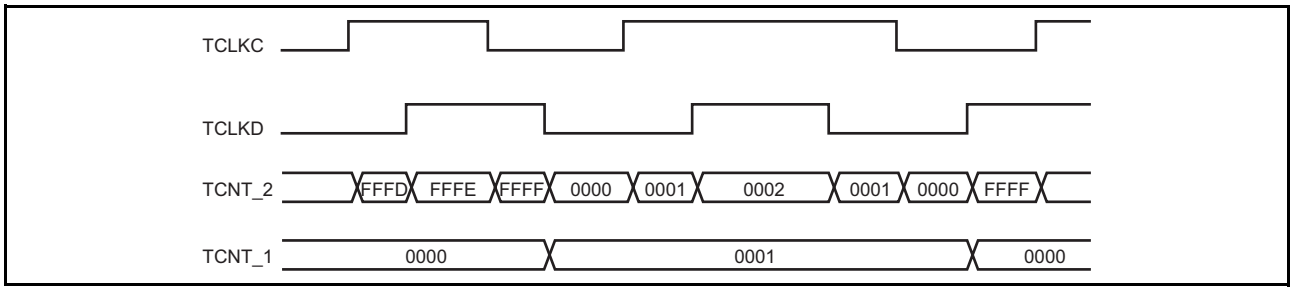


图 11.21　级联的运行例子 （a）

（3） 级联运行的例子 （b）

将 TCNT_1 和 TCNT_2 进行级联并将 TICCR 的 I2AE 位置 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。

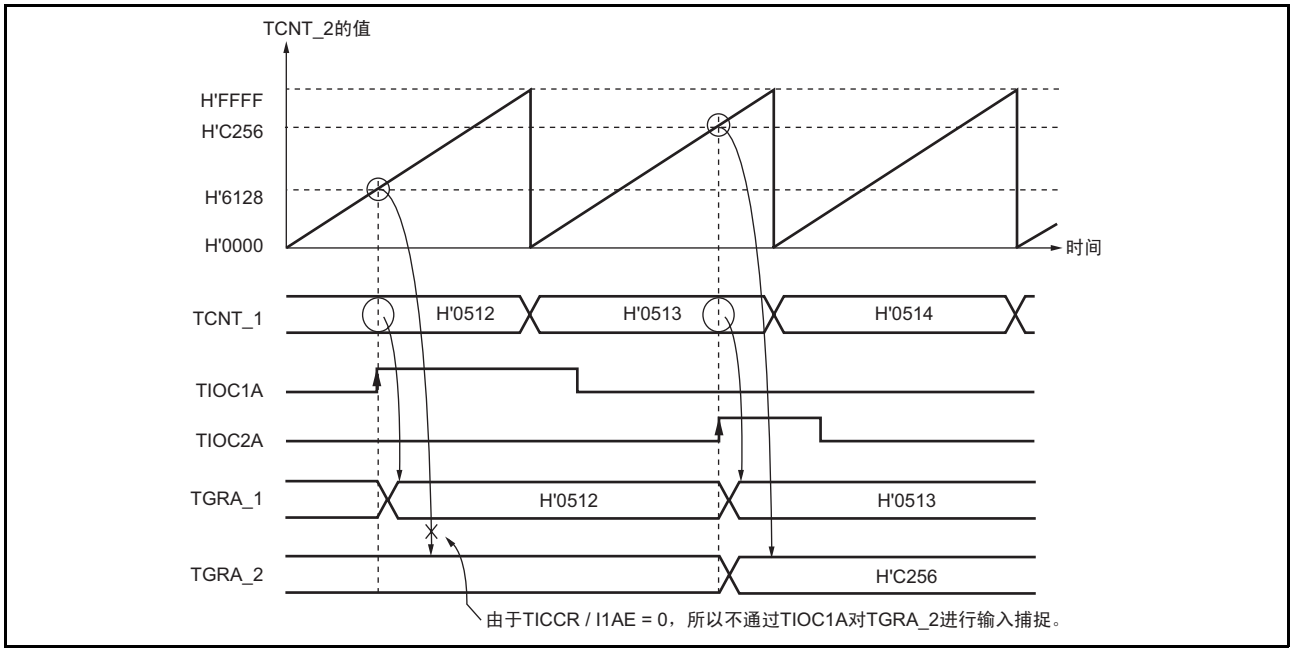


图 11.22　级联的运行例子 （b）

（4） 级联运行的例子 （c）

将 TCNT_1 和 TCNT_2 进行级联并将 TICCR 的 I2AE 位和 I1AE 位置 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。

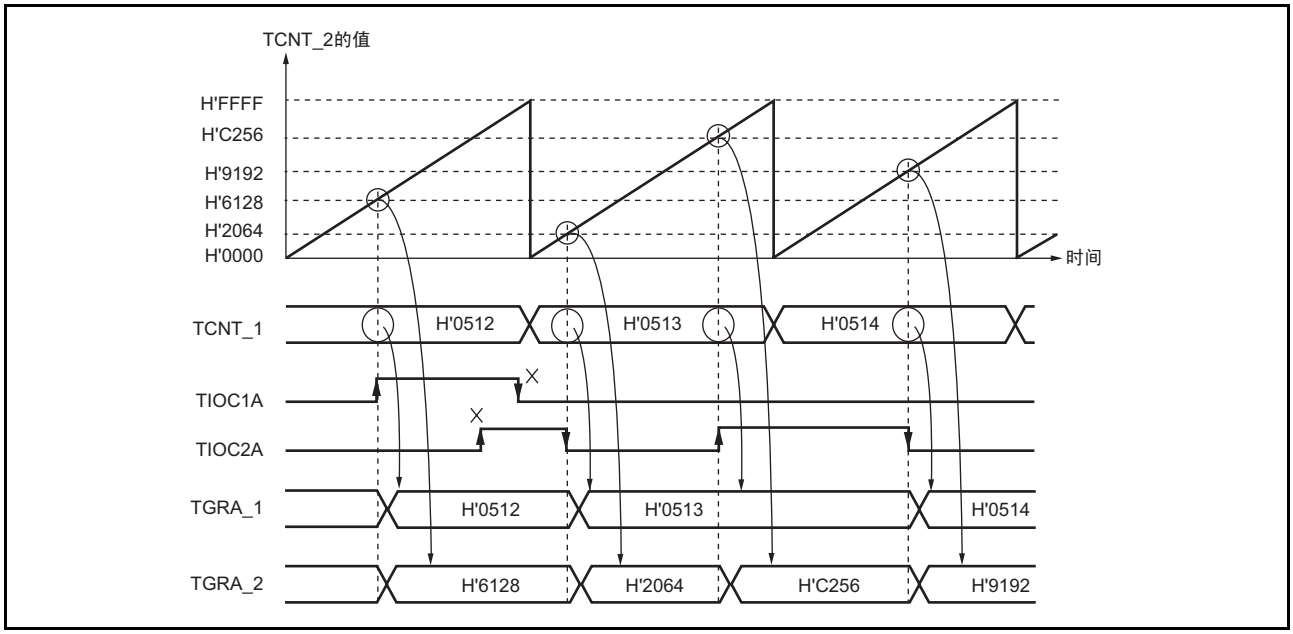


图 11.23　级联的运行例子 （c）

（5） 级联运行的例子 （d）

将 TCNT_1 和 TCNT_2 进行级联并给 TICCR 的 I2AE 位置 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 / 输入捕捉进行输入捕捉。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，所以即使将 TICCR 的 I2AE 位置 1， TIOC2A 的边沿也不为 TGRA_1 的输入捕捉条件。

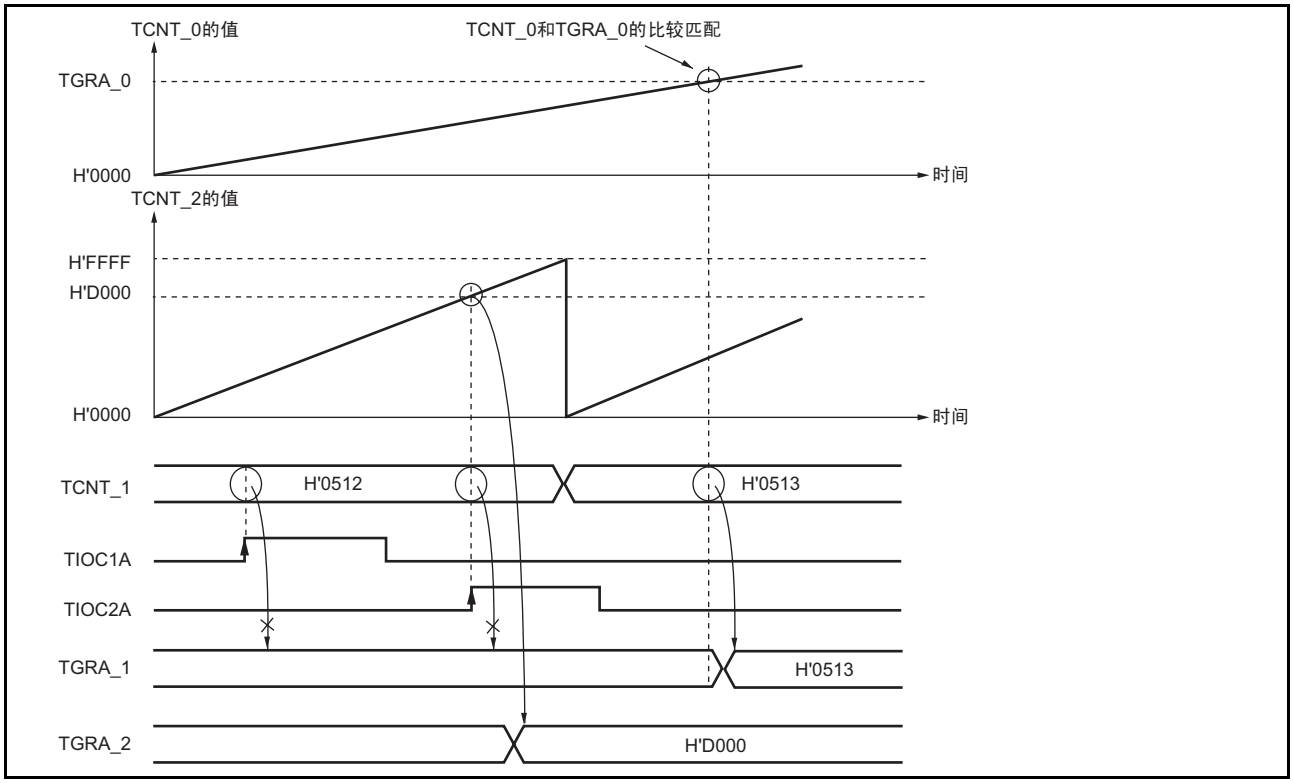


图 11.24　级联的运行例子 （d）

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 / 交替输出中选择各个 TGR 比较匹配的输出电平。

能通过设定各个 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时器清除源，给该寄存器设定周期。所有通道均能独立设定为 PWM 模式。也能进行同步运行。

PWM 模式有以下 2 种。

- 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的设定值相同，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。

- 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，通过与同伴运行并用，能进行最多8相的PWM输出。
  PWM输出引脚和寄存器的对应关系如**表11.44**所示。

表 11.44　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	
	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。

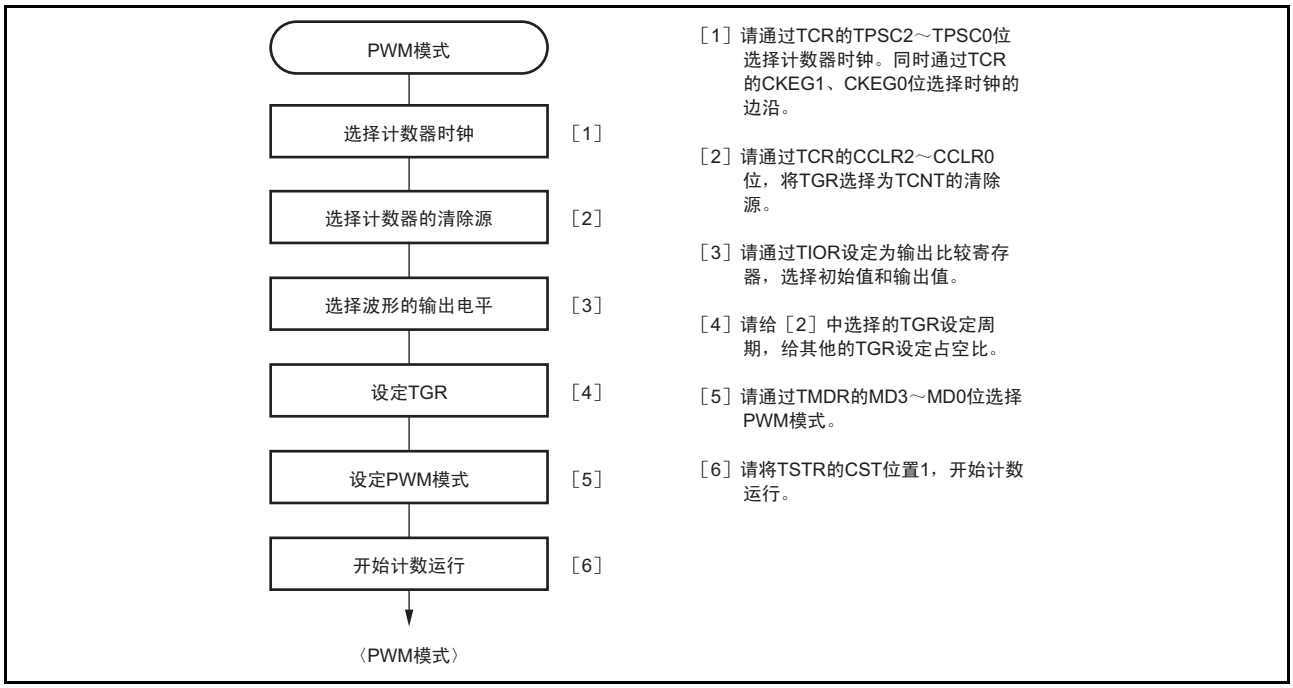


图 11.25　PWM 模式的设定步骤例子

（2） PWM 模式的运行例子

PWM 模式 1 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设定为 0， TGRB 的输出值设定为 1。

此时，设定在 TGRA 的值为周期，设定在 TGRB 的值为占空比。

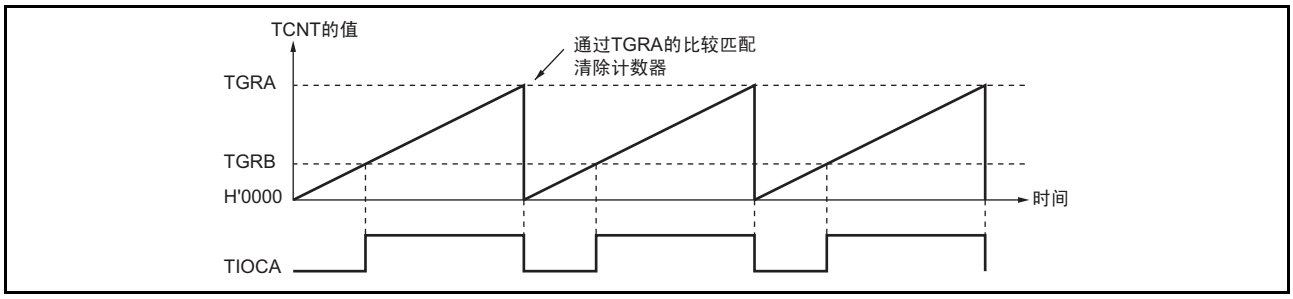


图 11.26　PWM 模式的运行例子

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）的初始输出值设定为 0、输出值设定为 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。

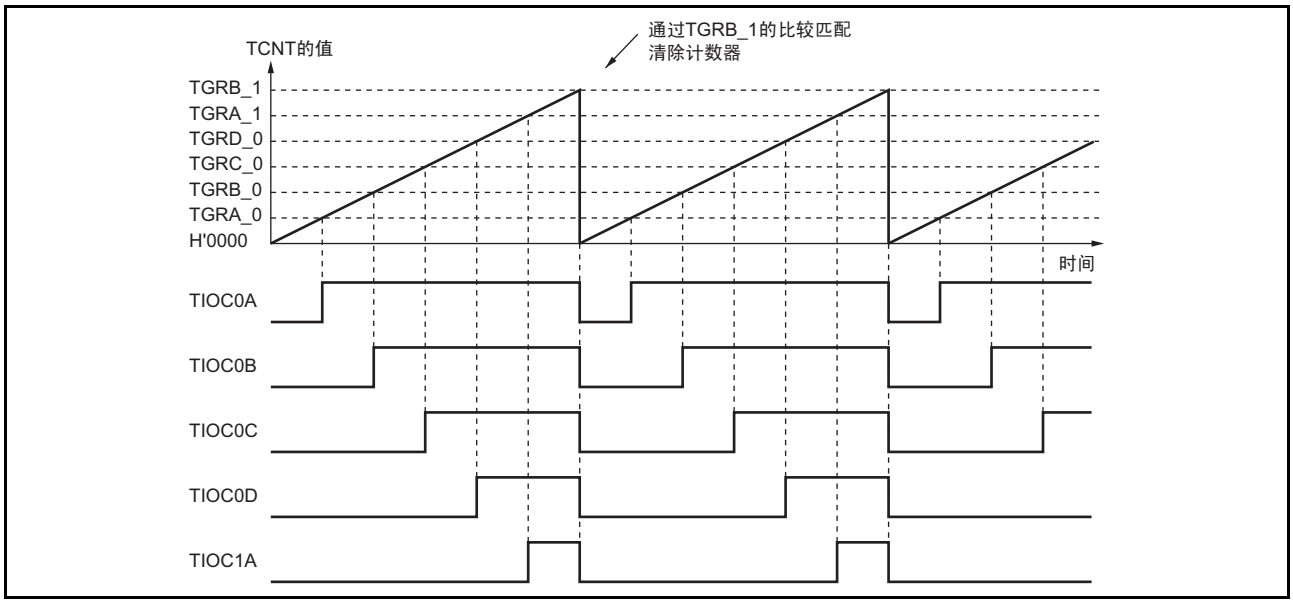


图 11.27　PWM 模式的运行例子

在 PWM 模式中，输出占空比为 0%和占空比为 100%的 PWM 波形例子如图 11.28 所示。

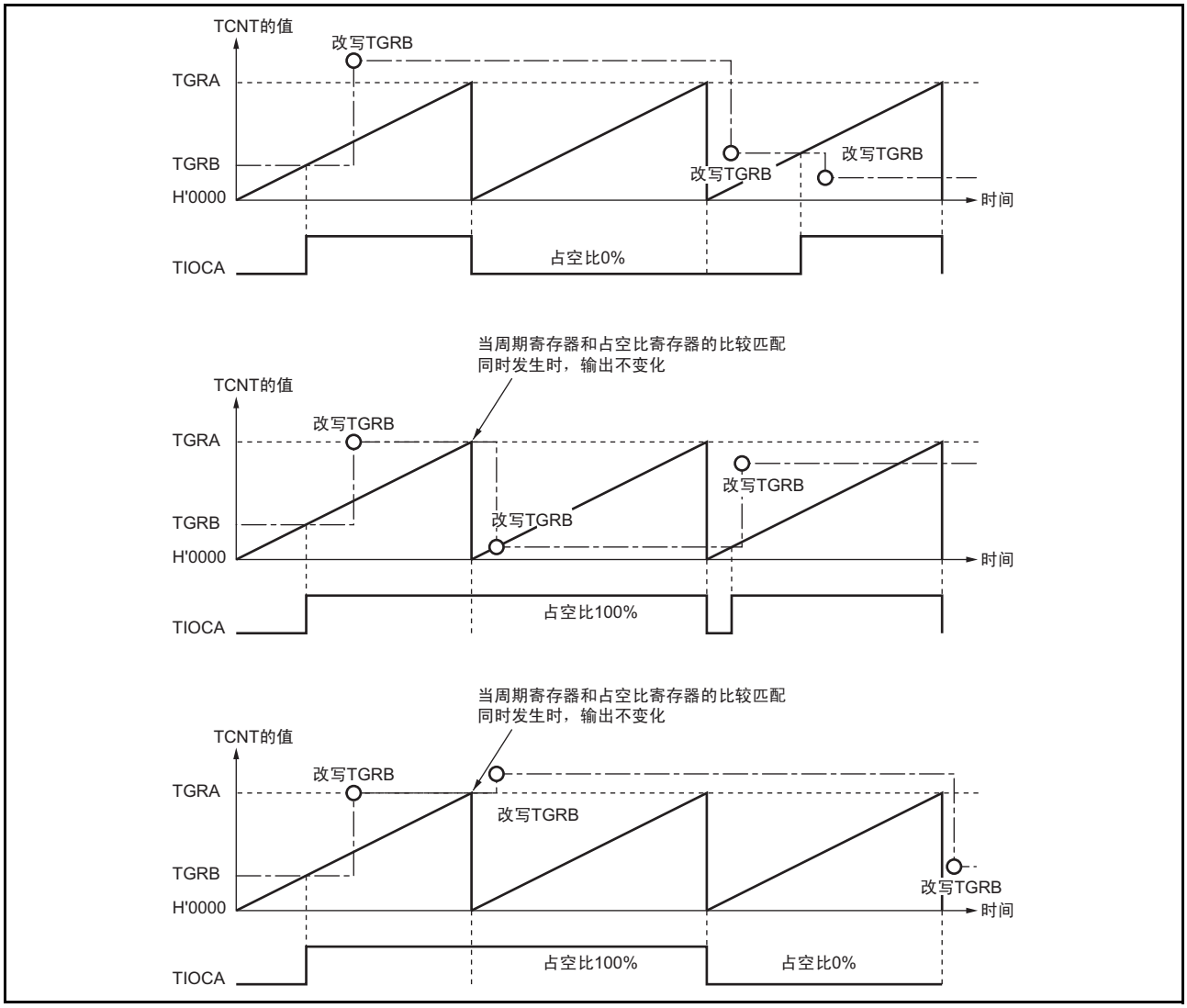


图 11.28　PWM 模式的运行例子

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，并对 TCNT 进行递增 / 递减计数。

设定为相位计数模式时，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。通过读取 TCFD 标志，能确认 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关系如**表 11.45** 所示。

表 11.45　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。

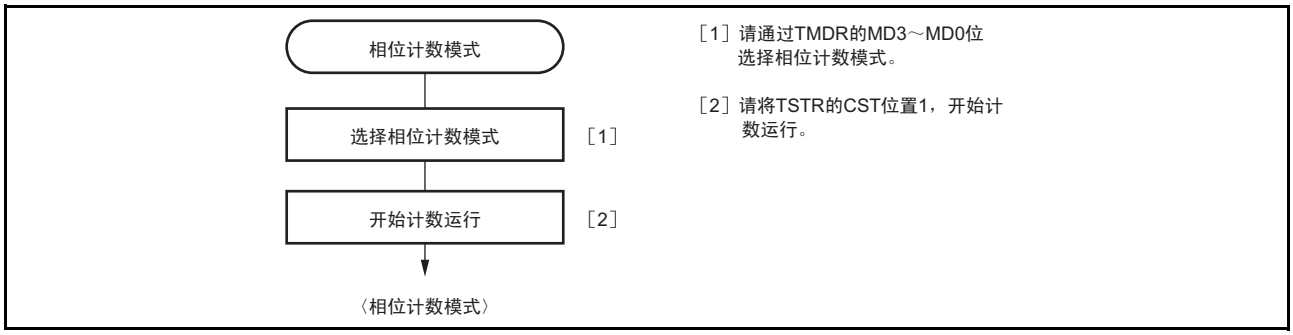


图 11.29　相位计数模式的设定步骤例子

（2） 相位计数模式的运行例子

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 / 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 的递增 / 递减计数条件如**表 11.46** 所示。

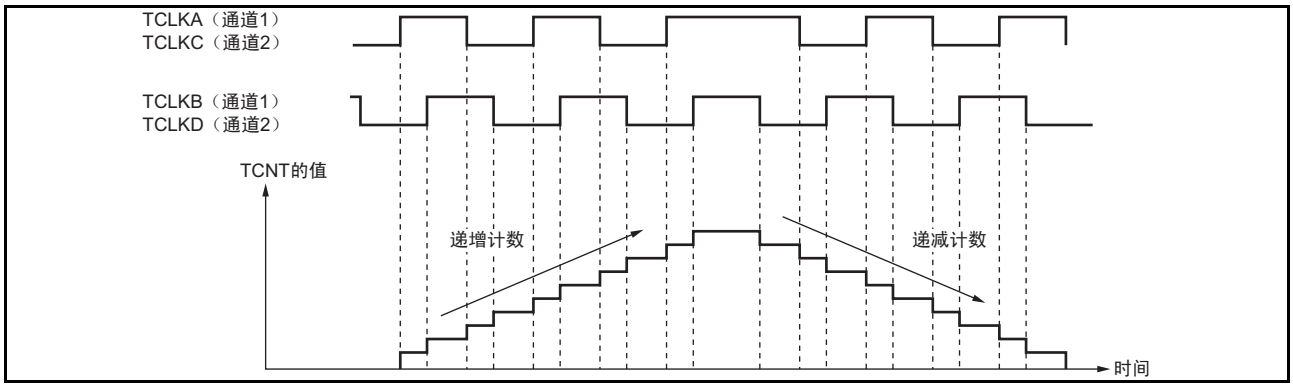


图 11.30　相位计数模式 1 的运行例子

表 11.46　相位计数模式 1 的递增 / 递减计数条件

TCLKA （通道 1） TCLKC （通道 2）	TCLKB （通道 1） TCLKD （通道 2）	运行内容
High 电平	↑	递增计数
Low 电平	↓	
↑	Low 电平	
↓	High 电平	
High 电平	↓	递减计数
Low 电平	↑	
↑	High 电平	
↓	Low 电平	

【符号说明】

↑ ：上升沿

↓ ：下降沿

（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 的递增 / 递减计数条件如**表 11.47** 所示。

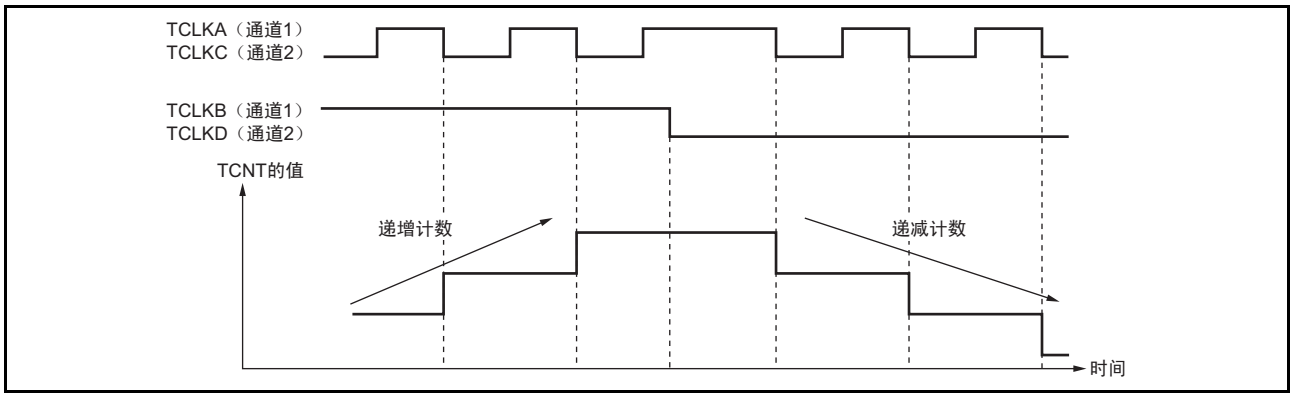


图 11.31　相位计数模式 2 的运行例子

表 11.47　相位计数模式 2 的递增 / 递减计数条件

TCLKA （通道 1） TCLKC （通道 2）	TCLKB （通道 1） TCLKD （通道 2）	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 的递增 / 递减计数条件如表 11.48 所示。

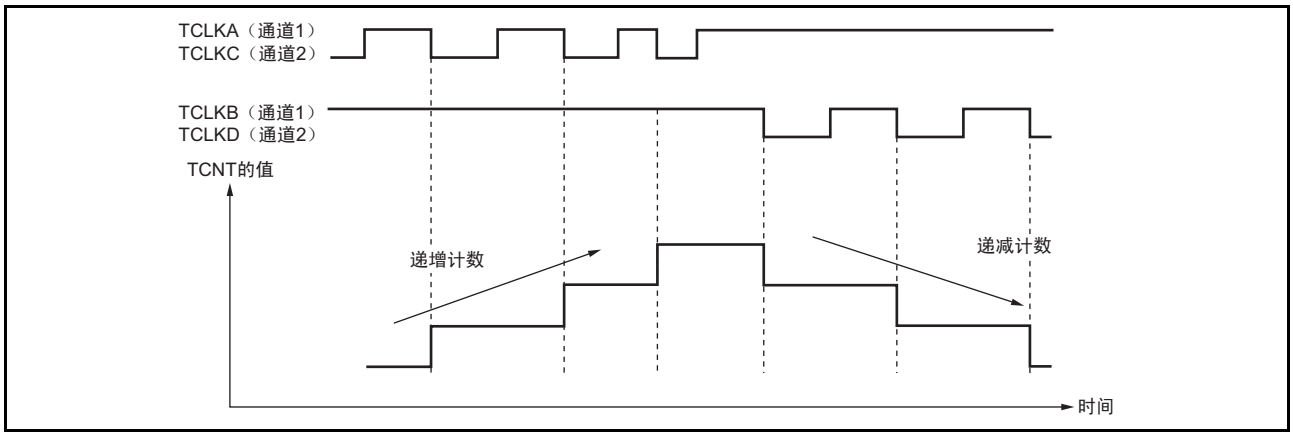


图 11.32　相位计数模式 3 的运行例子

表 11.48　相位计数模式 3 的递增 / 递减计数条件

TCLKA （通道 1） TCLKC （通道 2）	TCLKB （通道 1） TCLKD （通道 2）	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	↓	递减计数
Low 电平	↑	不计数 （Don't care）
↑	High 电平	不计数 （Don't care）
↓	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 的运行例子如**图 11.33** 所示， TCNT 的递增 / 递减计数条件如**表 11.49** 所示。

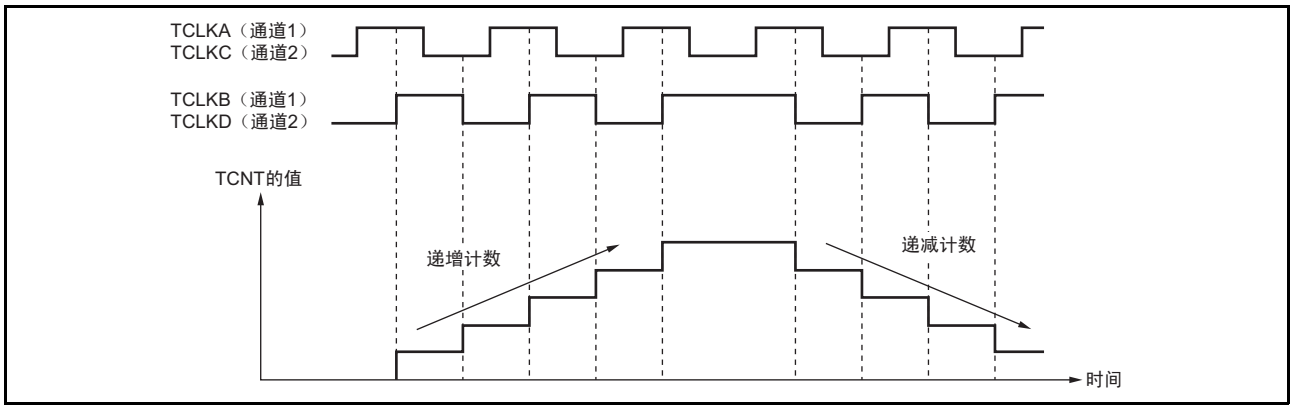


图 11.33　相位计数模式 4 的运行例子

表 11.49　相位计数模式 4 的递增 / 递减计数条件

TCLKA （通道 1） TCLKC （通道 2）	TCLKB （通道 1） TCLKD （通道 2）	运行内容
High 电平	↑	递增计数
Low 电平	↓	
↑	Low 电平	不计数 （Don't care）
↓	High 电平	
High 电平	↓	递减计数
Low 电平	↑	
↑	High 电平	不计数 （Don't care）
↓	Low 电平	

【符号说明】
↑ ： 上升沿
↓ ： 下降沿

（3） 相位计数模式的应用例子

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 的计时器输入时钟，检测 2 相解码器的 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。

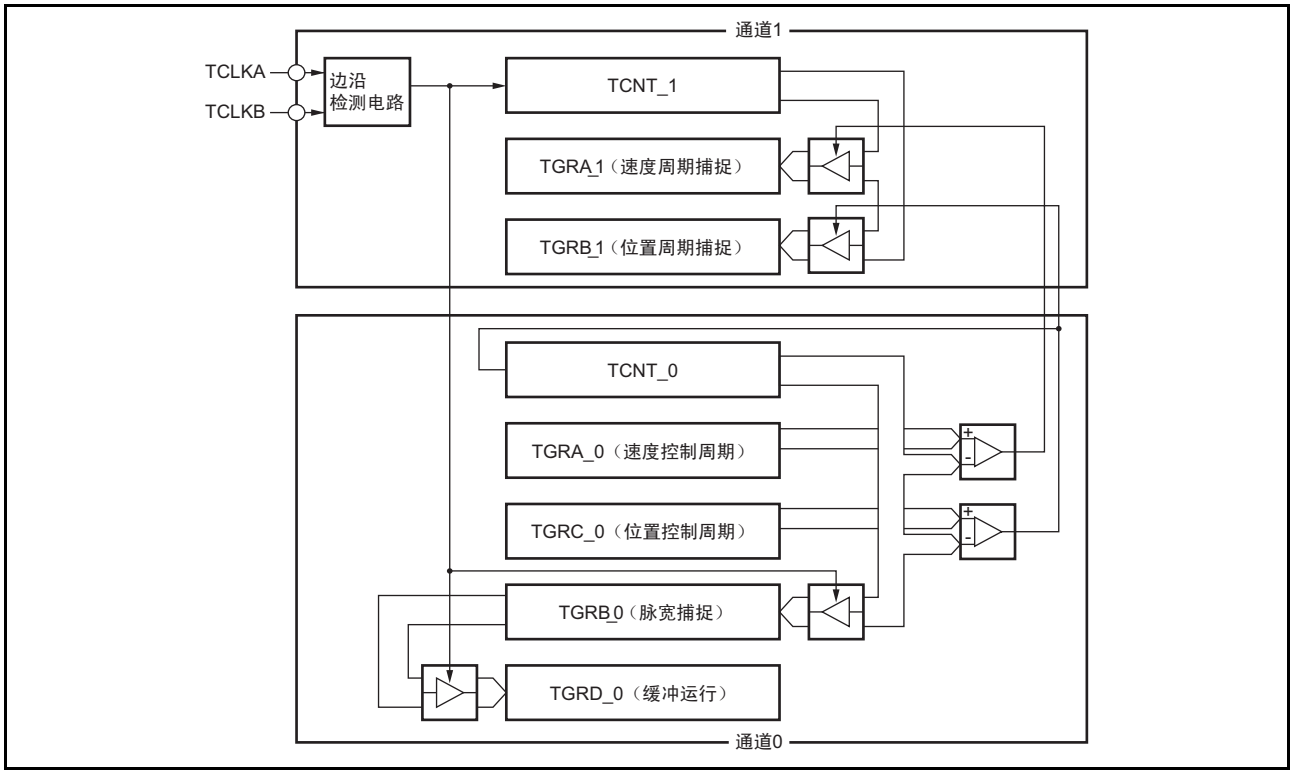


图 11.34　相位计数模式的应用例子

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 11.50 和表 11.51 所示。

表 11.50　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 11.51　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 波形的转换点
TGRA_4	设定从 TIOC4A、 TIOC4C 引脚输出的 PWM 波形的转换点
TGRB_4	设定从 TIOC4B、 TIOC4D 引脚输出的 PWM 波形的转换点

（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。

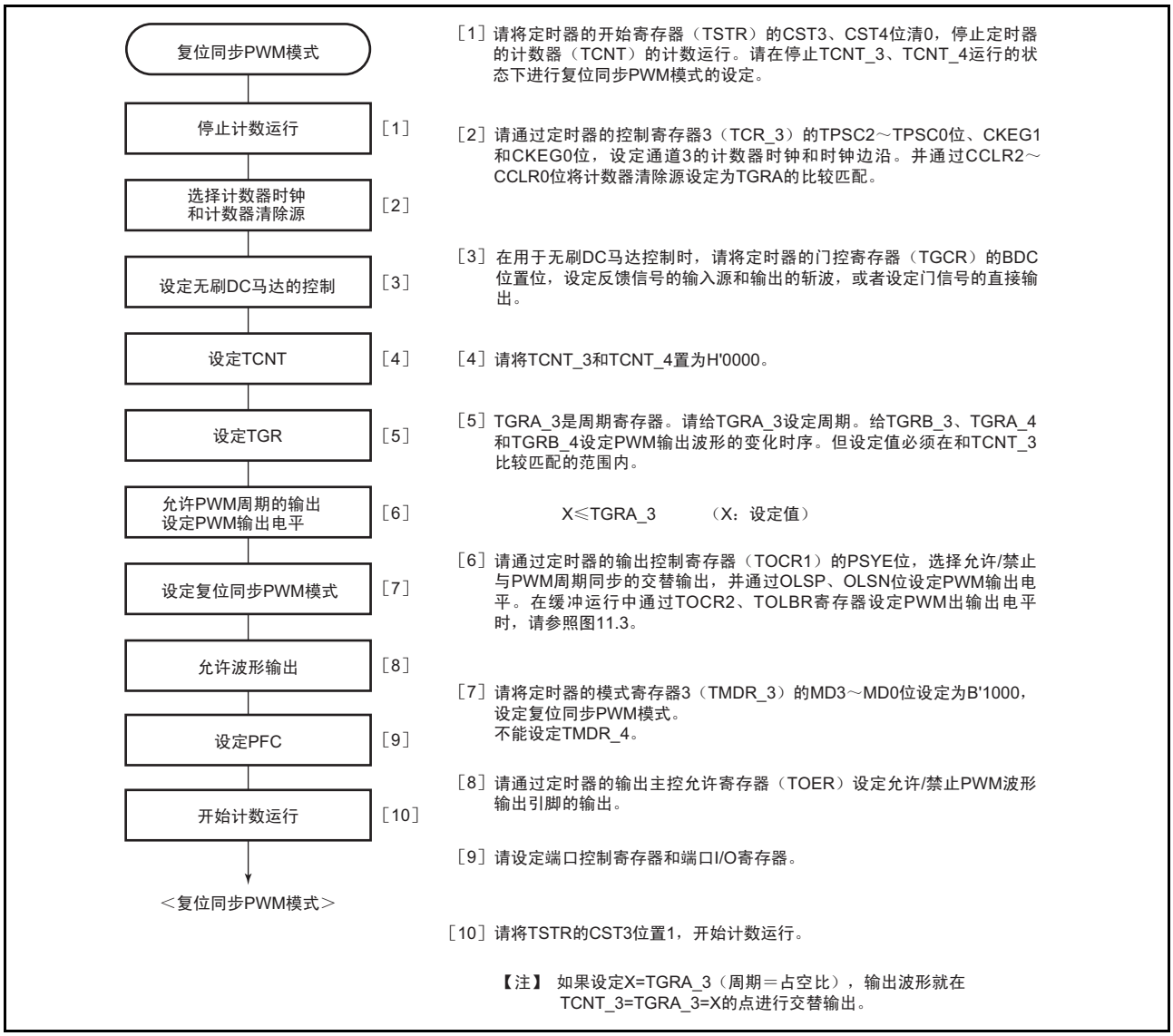


图 11.35　复位同步 PWM 模式的设定步骤例子

（2） 复位同步 PWM 模式的运行例子

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。

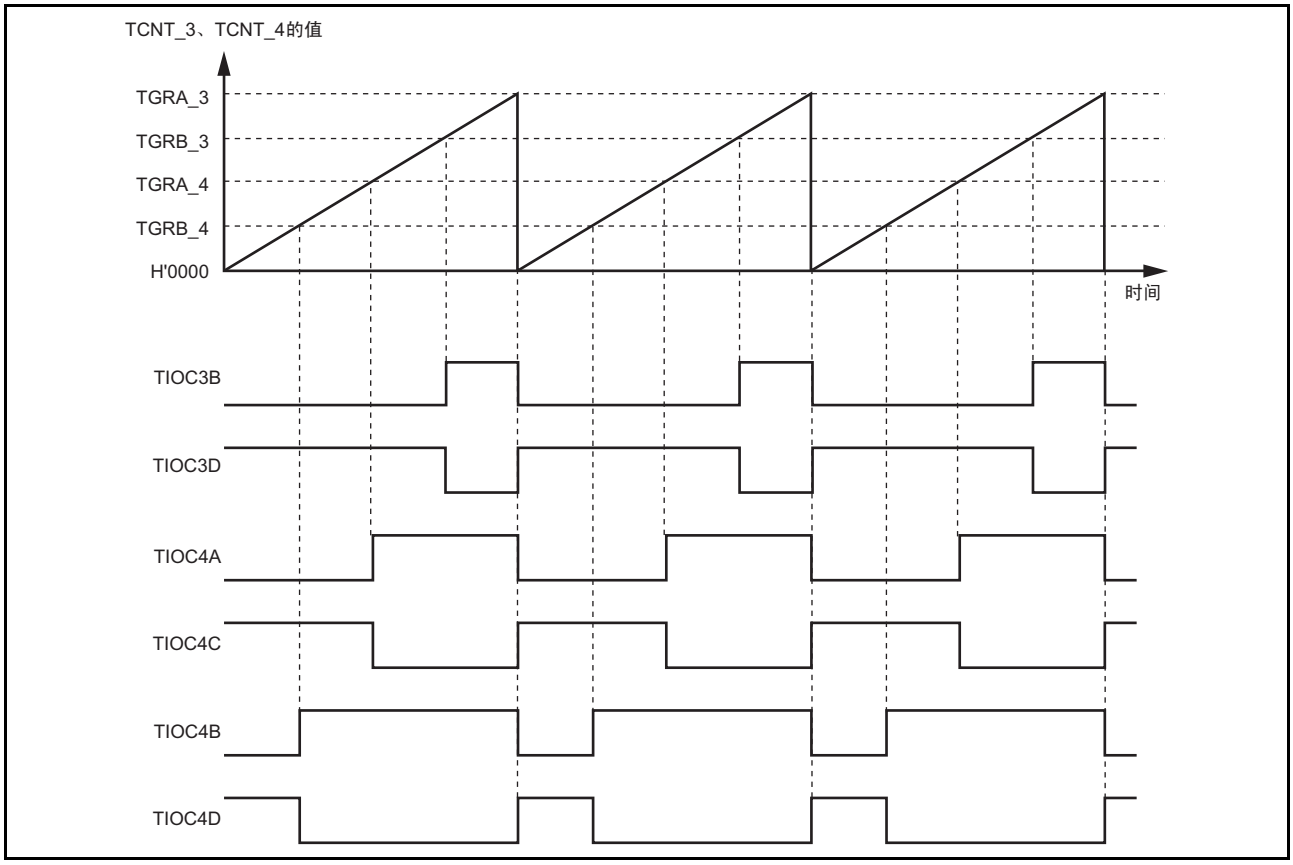


图 11.36　复位同步 PWM 模式的运行例子 （当设定 TOCR 的 OLSN=1、 OLSP=1 时）

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 输出引脚和寄存器的设定分别如表 11.52 和表 11.53 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 11.52　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'（与 PWM 输出 1 有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'（与 PWM 输出 2 有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'（与 PWM 输出 3 有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，请勿将 TIOC3C 引脚设定为定时器的输入 / 输出引脚。

表 11.53　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

通道	计数器 / 寄存器	说　　明	由 CPU 进行读写
定时器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 （定时器读写允许寄存器），允许或者禁止存取。

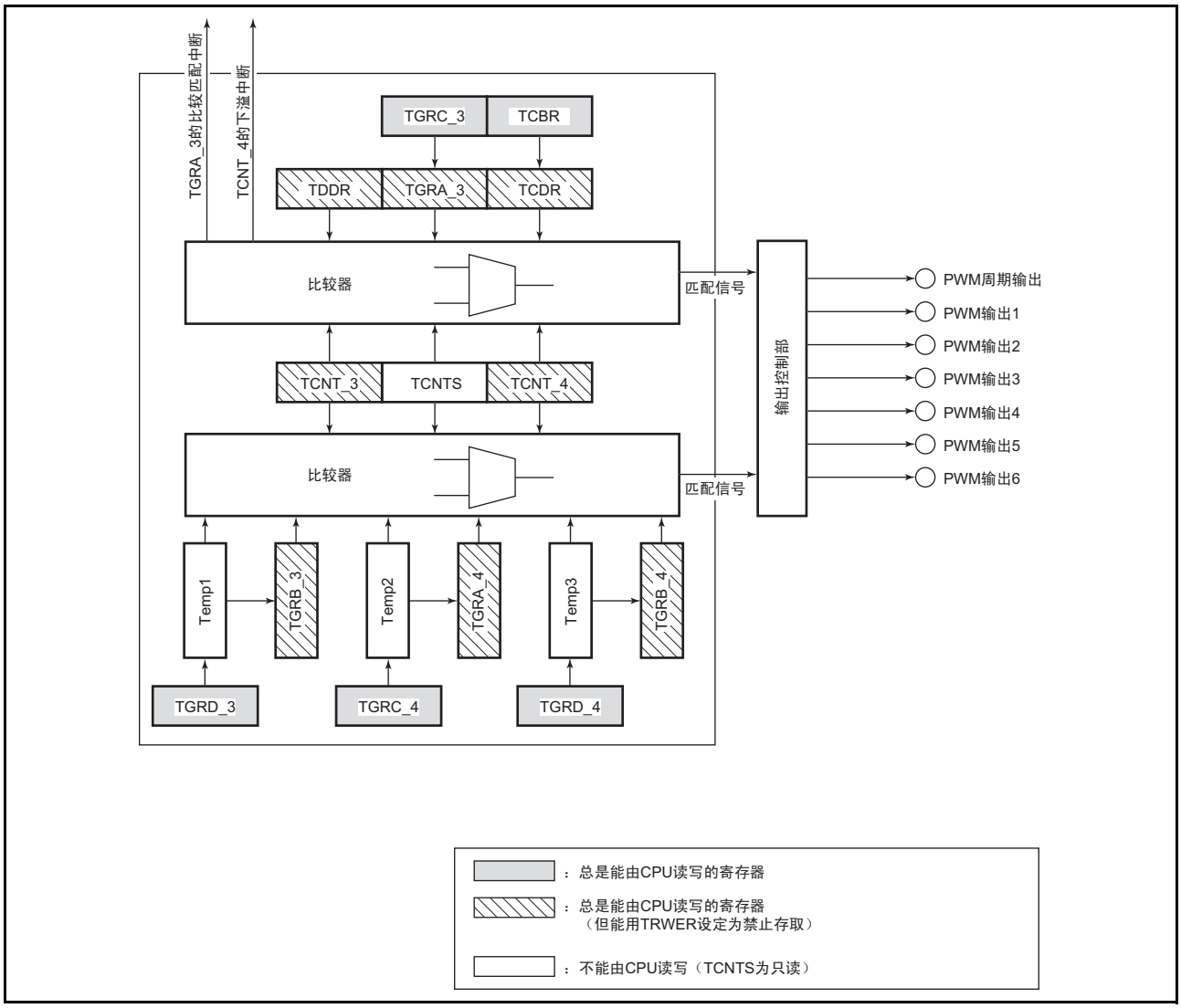


图 11.37　互补 PWM 模式中的通道 3、 4 框图

（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互补 PWM 模式的运行例子分别如图 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 / 递减计数。

当 TCNT_3 被设定为互补 PWM 模式并且 TSTR 的 CST 位为 0 时，设定在 TDDR 的值就自动设定为初始值。

当 CST 位被设定为 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 匹配时， TCNT_3 就转换为递减计数。当 TCNT_3 和 TDDR 匹配时， TCNT_3 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 位被设定为 1 时，TCNT_4 就与 TCNT_3 同步进行递增计数。当 TCNT_4 和 TCDR 匹配时，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 匹配， TCNTS 就开始递减计数。当 TCNTS 和 TCDR 匹配时，TCNTS 就转换为递增计数。另外，当 TCNTS 和 TGRA_3 匹配时，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 匹配，TCNTS 就开始递增计数。当 TCNTS 和 TDDR 匹配时，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。

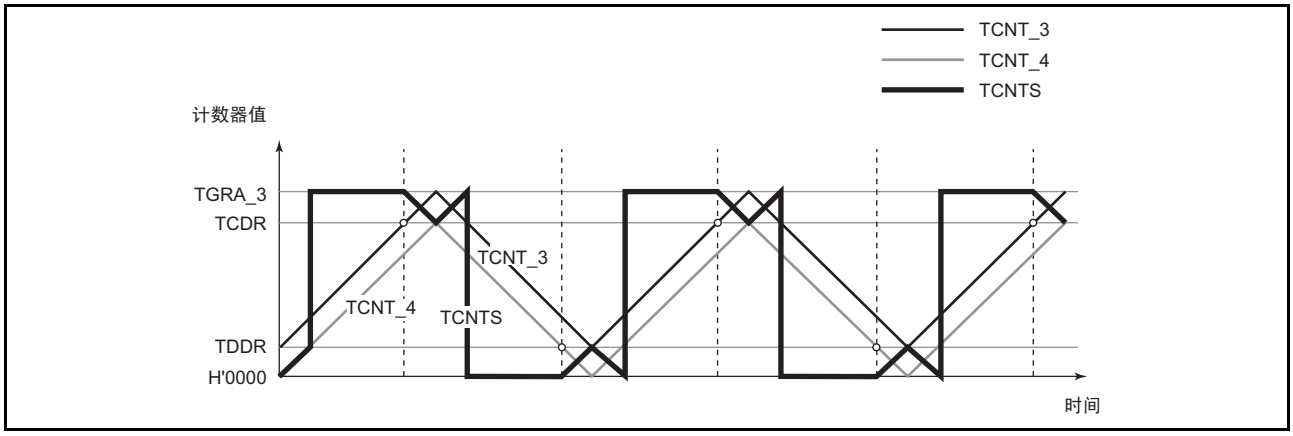


图 11.39　互补 PWM 模式的计数器运行

（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匹配时，就输出定时器输出控制寄存器 （TOCR）的 OLSN、 OLSP 位设定的值。

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。

为更改比较寄存器的数据，请给对应的缓冲寄存器写入更改数据。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。 Tb 区间结束后，将在此区间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 匹配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， 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。在此区间中， TCNT_3、TCNT_4、TCNTS（3 个计数器）与比较寄存器、暂存器（2 个寄存器）比较，控制 PWM 输出。

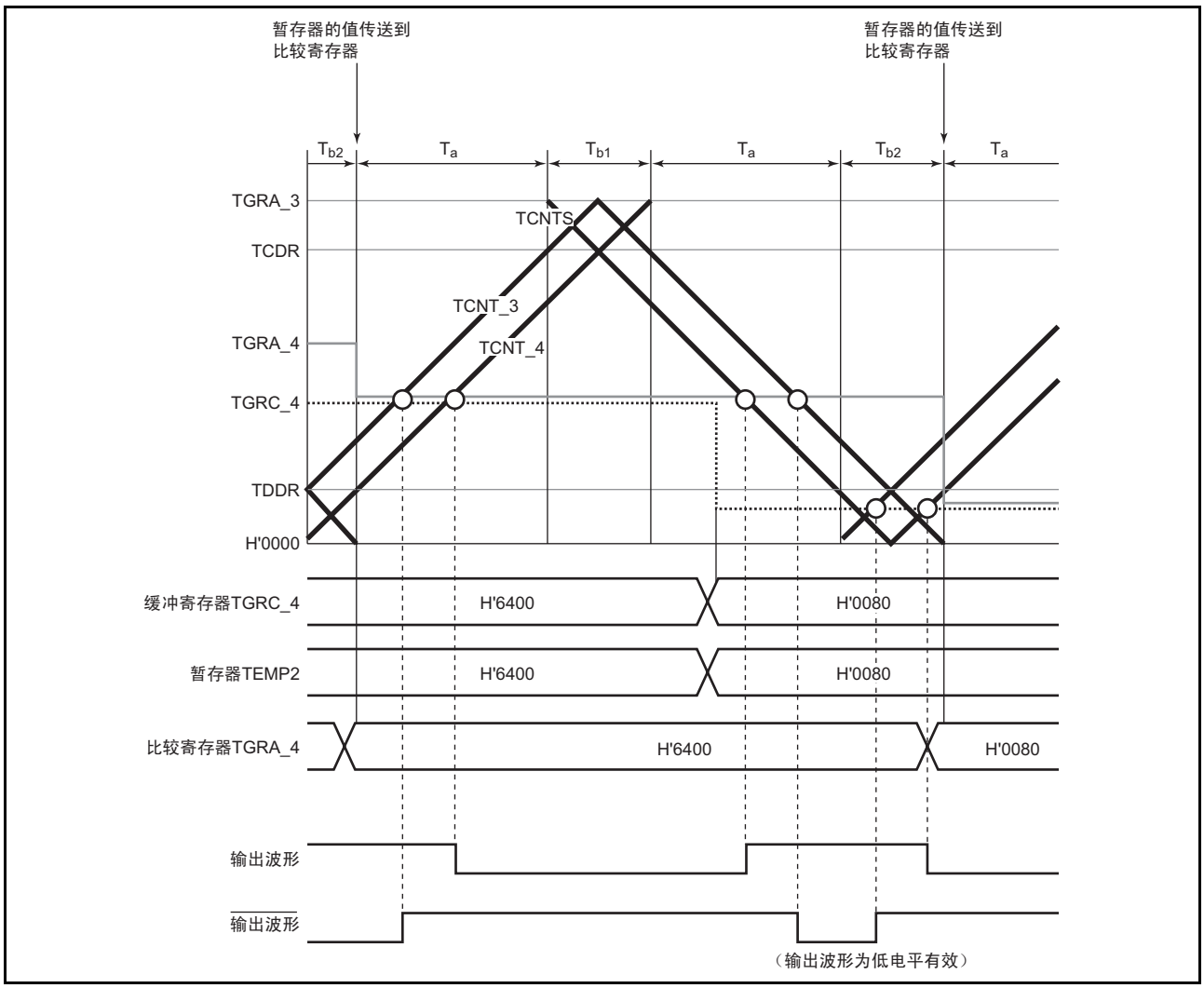


图 11.40　互补 PWM 模式运行例子

（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）。

在通过定时器模式寄存器 （TMDR）的 MD3 ～ MD0 位设定为互补 PWM 模式前，请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。定时器周期缓冲寄存器 （TCBR）用作定时器周期数据寄存器 （TCDR）的缓冲寄存器，设定 1/2 的 PWM 载波周期。还给定时器空载时间数据寄存器 （TDDR）设定空载时间 Td。

如果不生成空载时间，就将定时器空载时间允许寄存器 （TDER）的 TDER 位设定为 0，将 TGRC_3 和 TGRA_3 设定为 1/2 的 PWM 载波周期 +1， TDDR 设定为 1。

给缓冲寄存器 TGRD_3、 TGRC_4、 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，请在设定为互补 PWM 模式前将 TCNT_4 设定为 H'0000。

表 11.54　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 （通过 TDER 设定为不生成空载时间时，就为 1/2 的 PWM 载波周期 +1）
TDDR	空载时间 Td （通过 TDER 设定为不生成空载时间时，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。但通过TDER设定为不生成空载时间时，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，通过定时器输出控制寄存器 1 （TOCR1）的 OLSN、 OLSP 位或者定时器输出控制寄存器 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，请在解除互补 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。请在解除互补 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 设定为 0。

将 TGRA_3、 TGRC_3 设定为 1/2 的 PWM 载波周期 +1，并且将定时器空载时间数据寄存器 （TDDR）设定为 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。不生成空载时间的运行例子如图 11.41 所示。

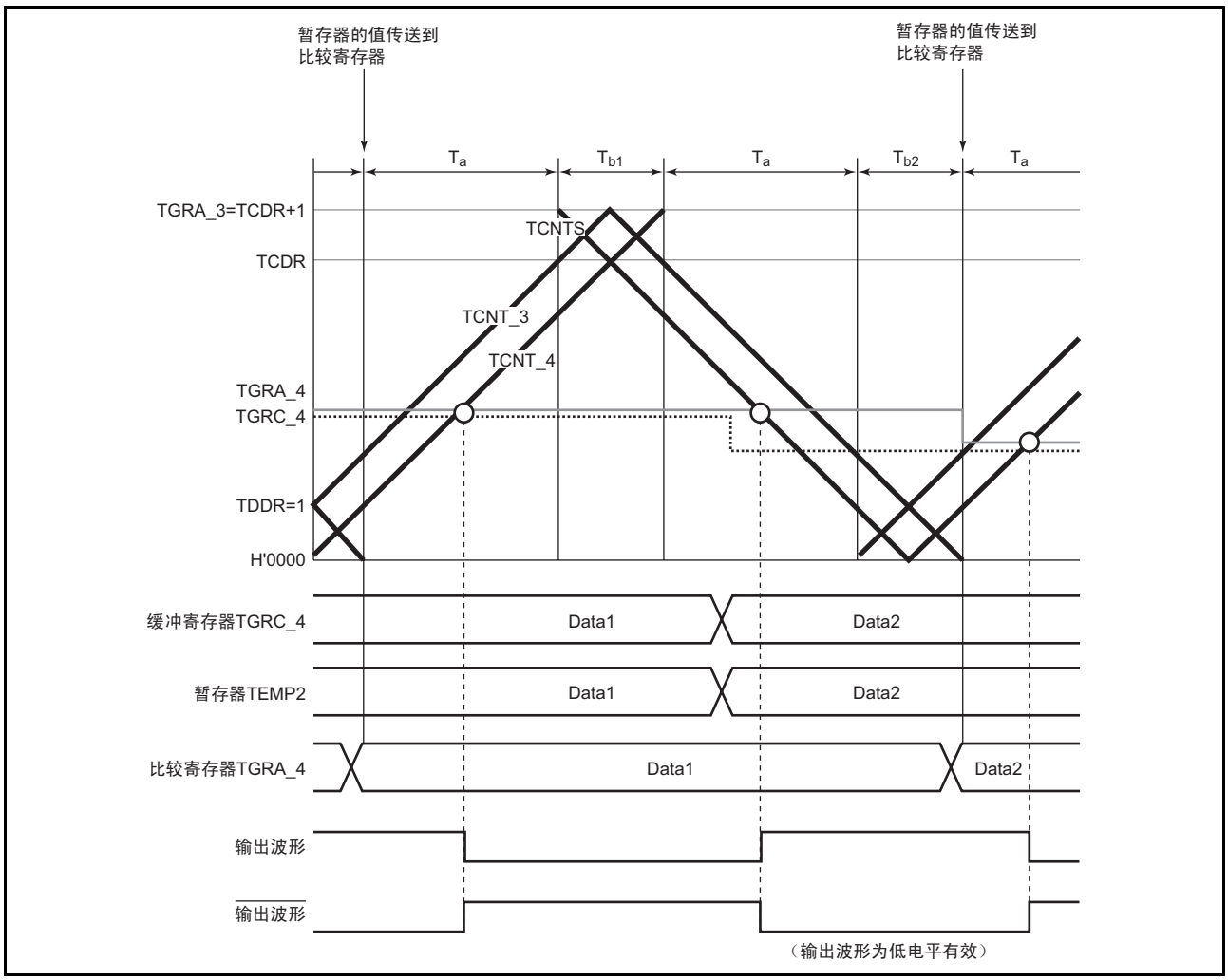


图 11.41　不生成空载时间的运行例子

（g） PWM 周期的设定

在互补 PWM 模式中，将 PWM 脉冲的周期设定在 TGRA_3（设定 TCNT3 的上限值）和 TCDR（设定 TCNT_4 的上限值）2 个寄存器中。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，请通过给缓冲寄存器的 TGRC_3、TCBR 设定值，来进行 TGRA_3、TCDR 的设定。在定时器模式寄存器（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”。

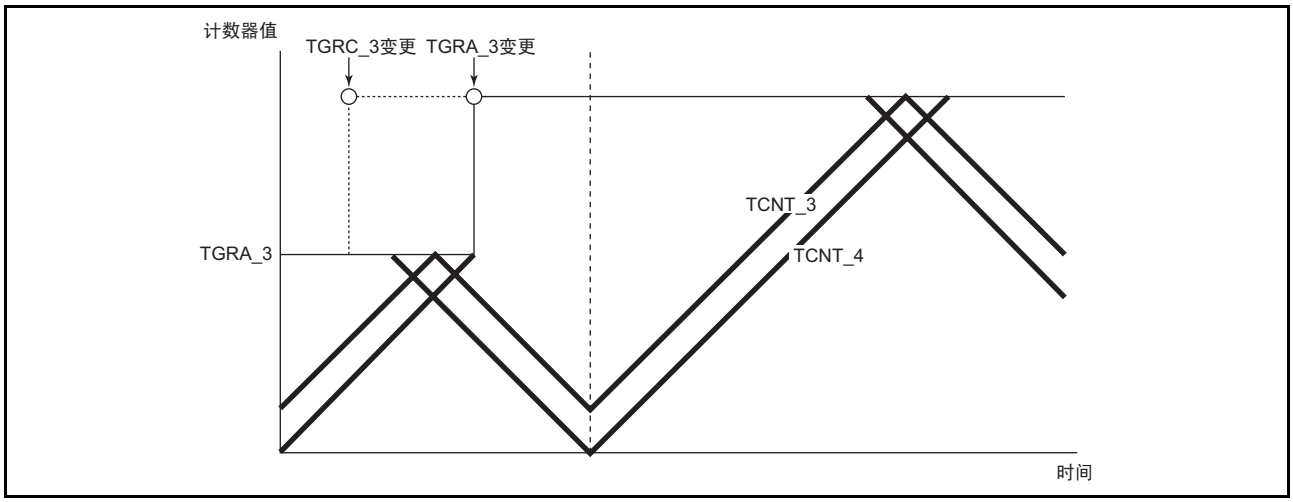


图 11.42　PWM 周期的变更例子

（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，有 5 个用于 PWM 占空比和载波周期的寄存器 （这些寄存器有缓冲寄存器并在运行过程中能更改）。

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，如果更新缓冲寄存器的数据，暂存器的值也会被改写。TCNTS 在进行计数时，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，最好必须对 TGRD_4 进行写入。在写入 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 5 个寄存器不进行全部更新或者不更新 TGRD_4 的数据，就必须在写入要更新的寄存器数据后写入 TGRD_4。此时，写到 TGRD_4 的数据必须和写之前的数据相同。

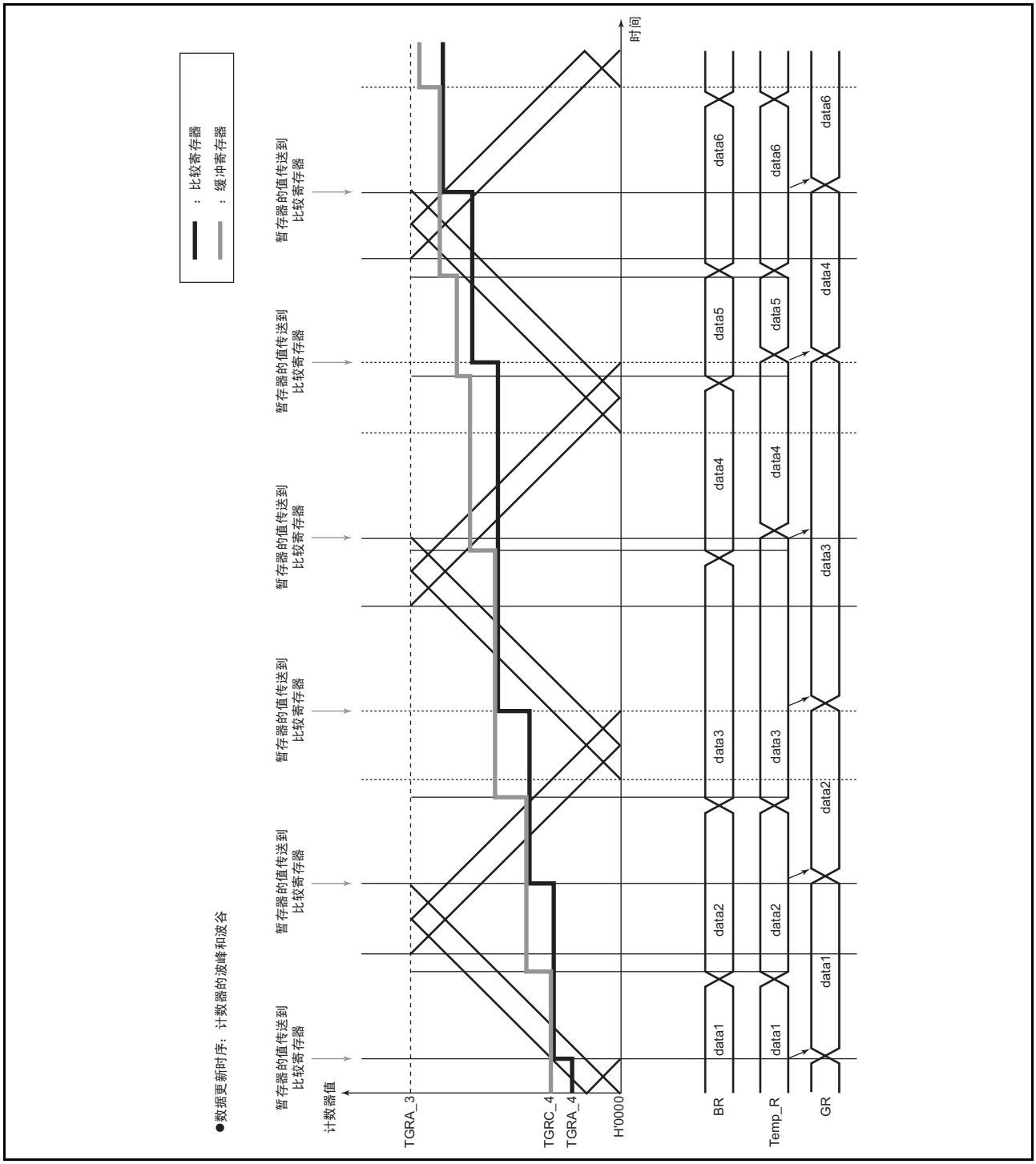


图 11.43　互补 PWM 模式的数据更新例子

（i）　互补 PWM 模式的初始输出

在互补 PWM 模式中，通过设定定时器输出控制寄存器 1（TOCR1）的 OLSN、OLSP 位或者定时器输出控制寄存器 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。

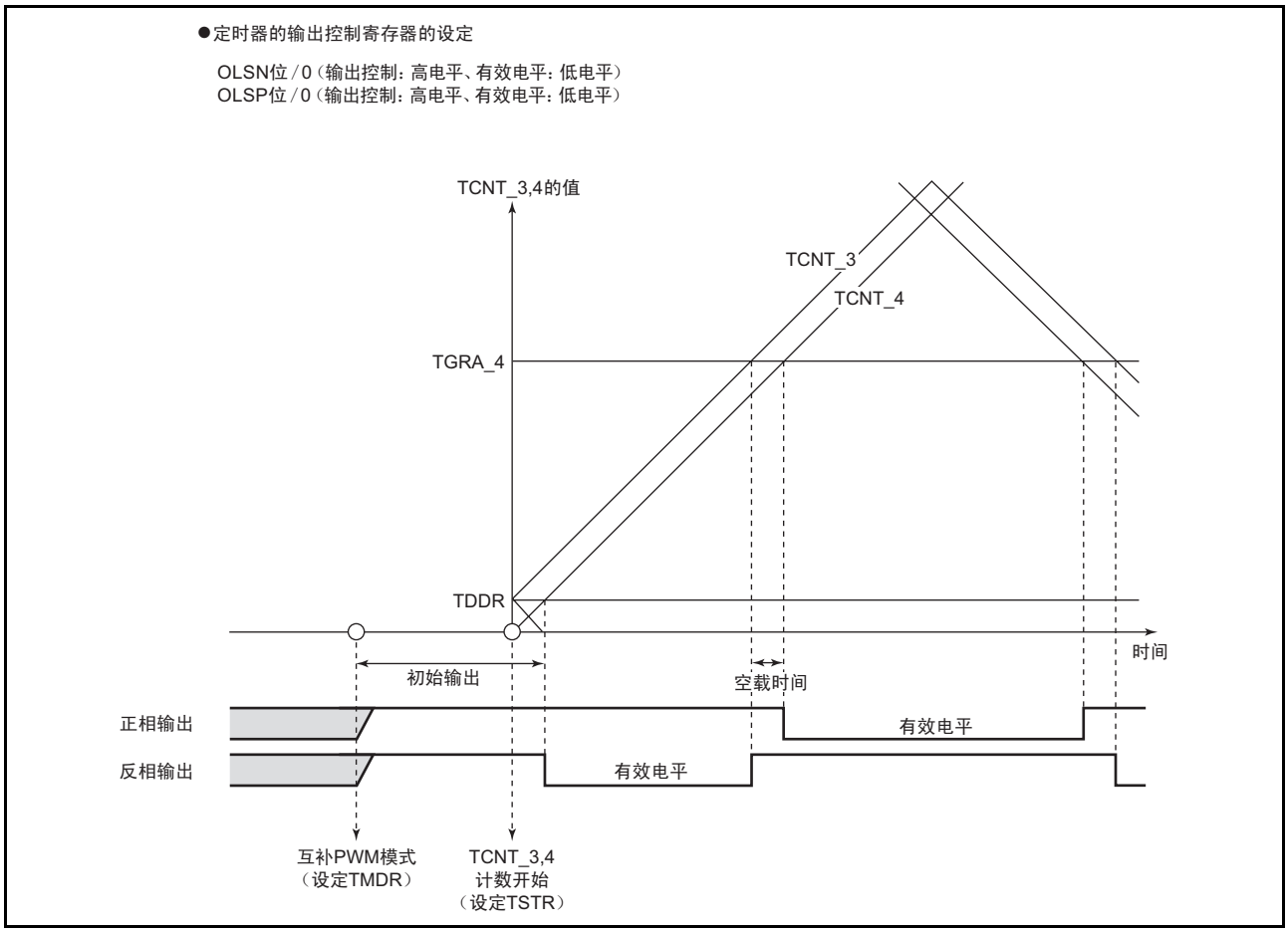


图 11.44　互补 PWM 模式的初始输出例子 （1）

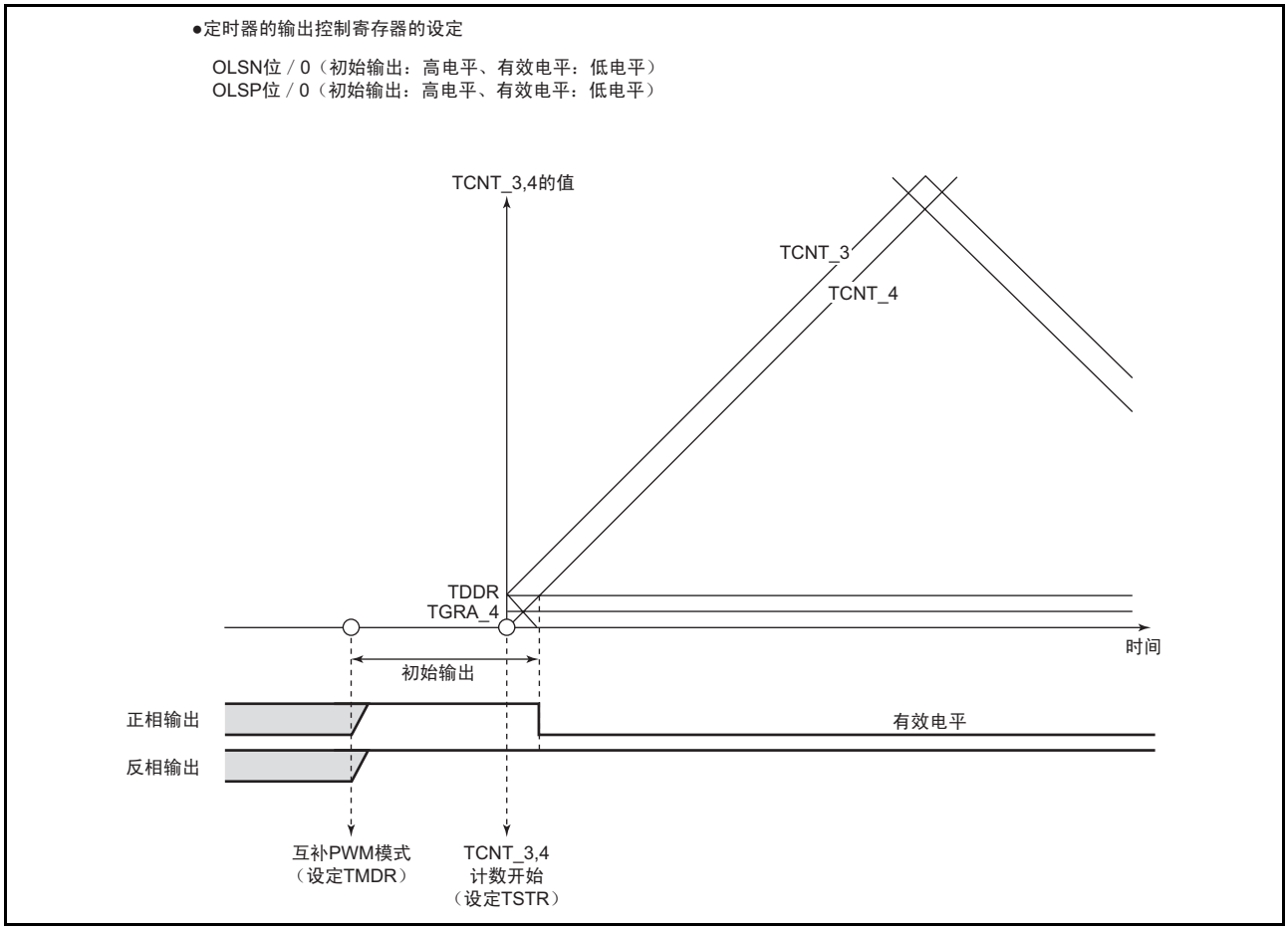


图 11.45　互补 PWM 模式的初始输出例子 （2）

（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，产生 ON、OFF 比较匹配的时序都会改变，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，通过与点线计数器（用比实线计数器延迟空载时间长度运行的点线计数器）的比较匹配生成 ON 时序。在 T1 期间，最优先进行反相 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，最优先进行正相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 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，在 a 的比较匹配之后先产生 c 的比较匹配时，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 c 的比较匹配 （正相 OFF），优先使正相 OFF （因此波形不发生变化，因为正相是 OFF 到 OFF）。

同样地，在**图 11.48** 所示的例子中，比起 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 之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。

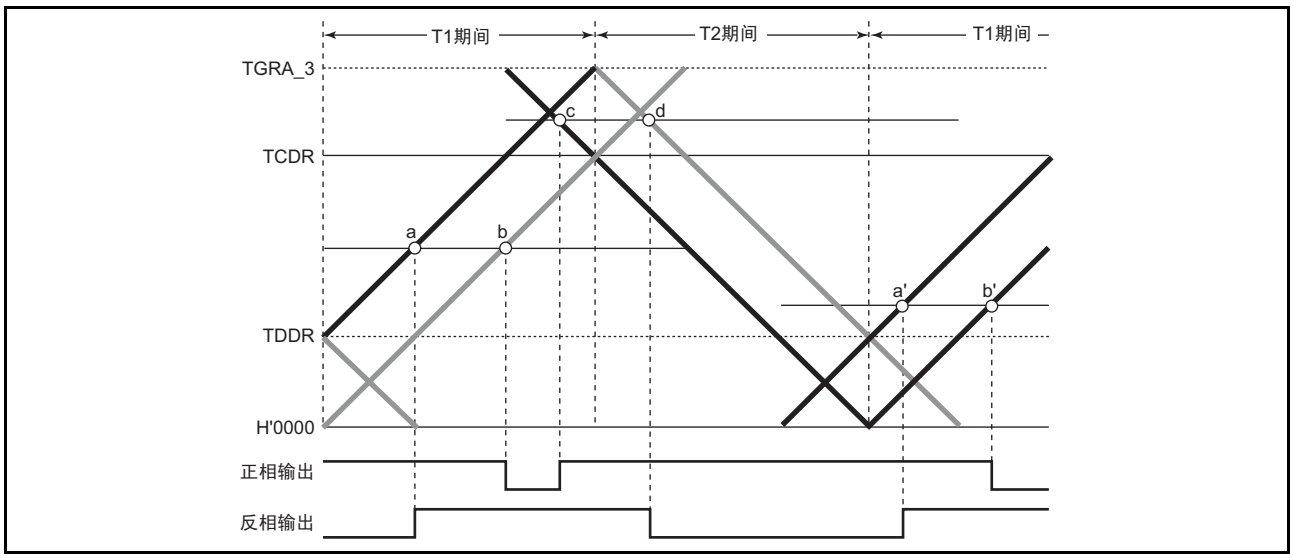


图 11.46　互补 PWM 模式的波形输出例子 （1）

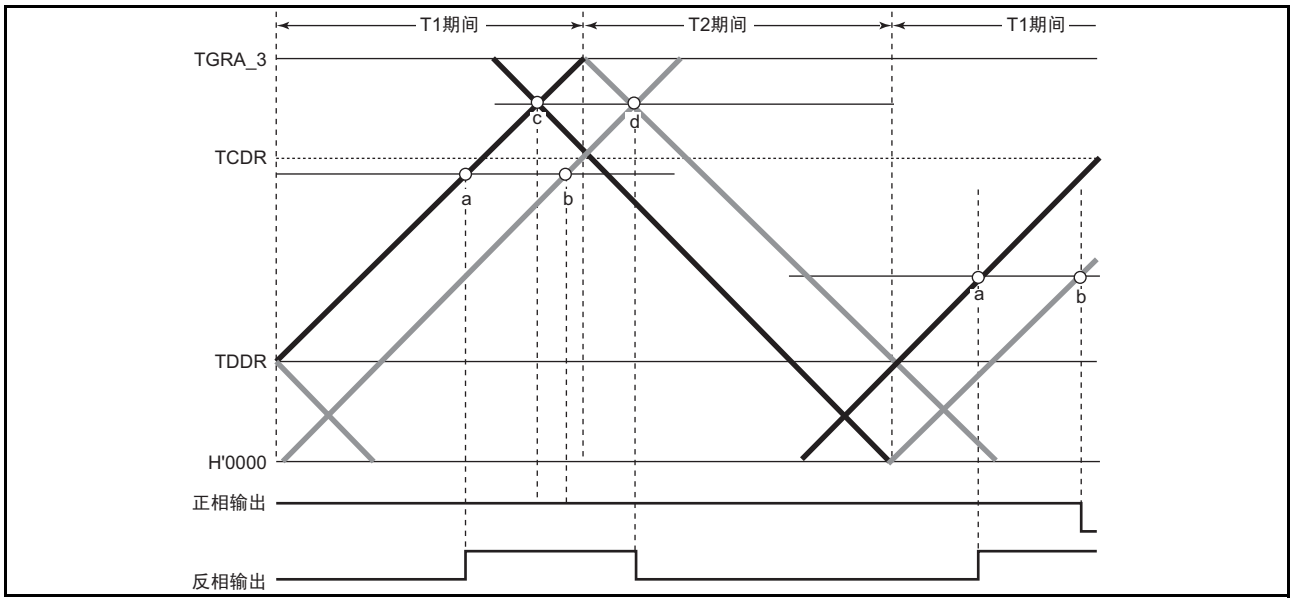


图 11.47　互补 PWM 模式的波形输出例子 （2）

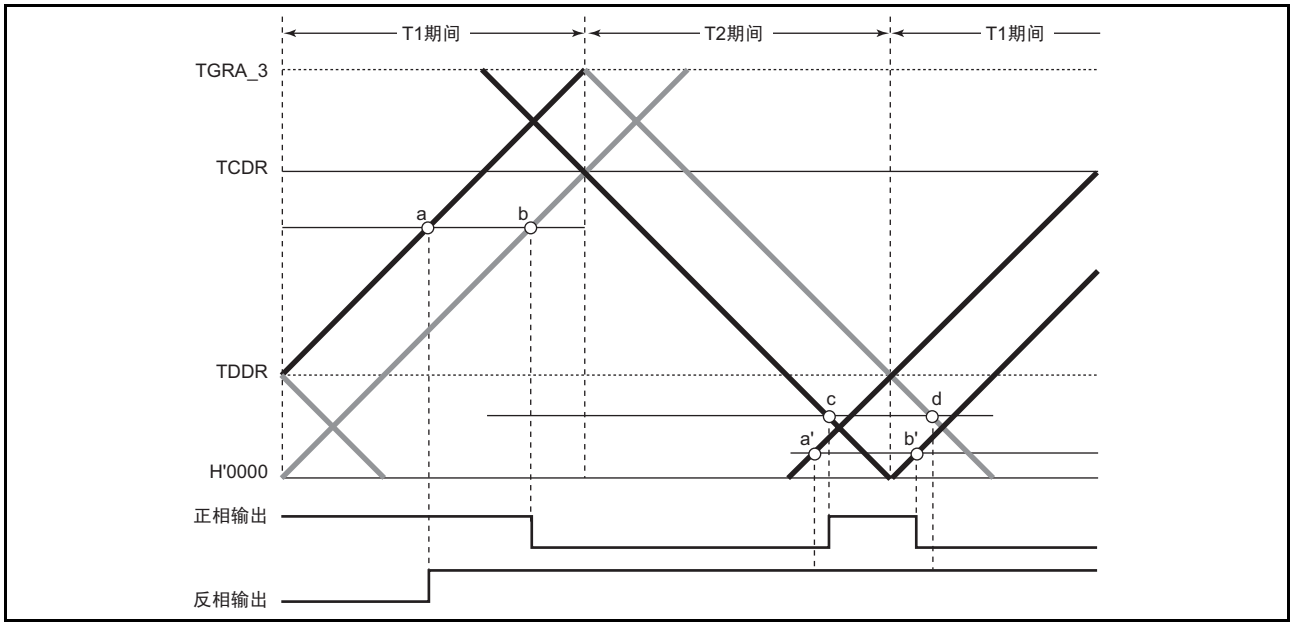


图 11.48　互补 PWM 模式的波形输出例子 （3）

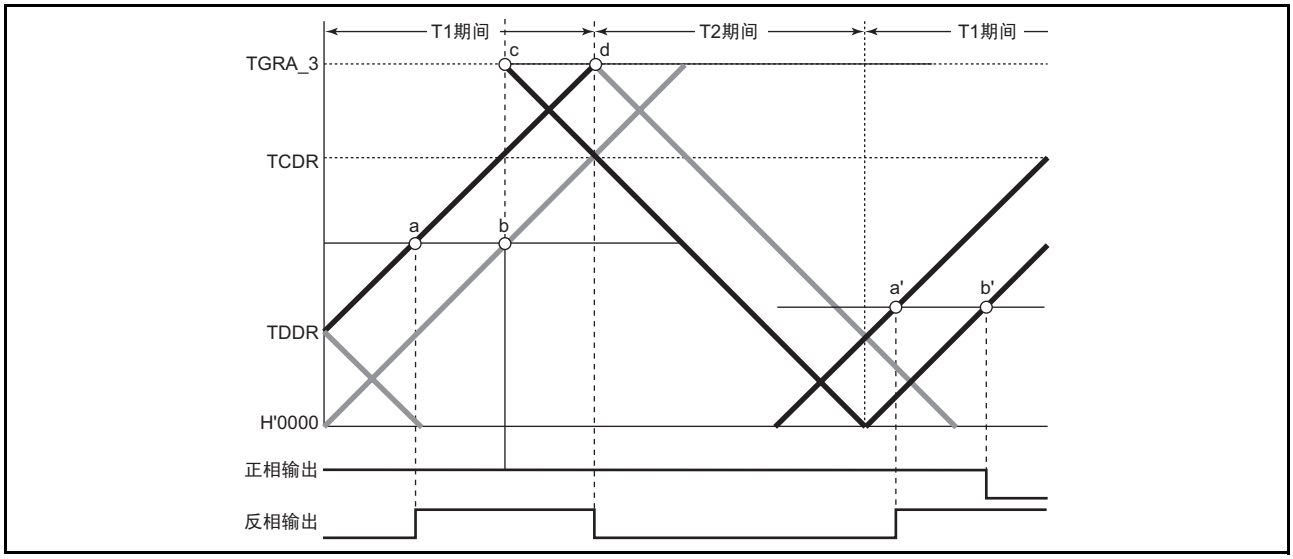


图 11.49　互补 PWM 模式的 0%、 100% 波形输出例子 （1）

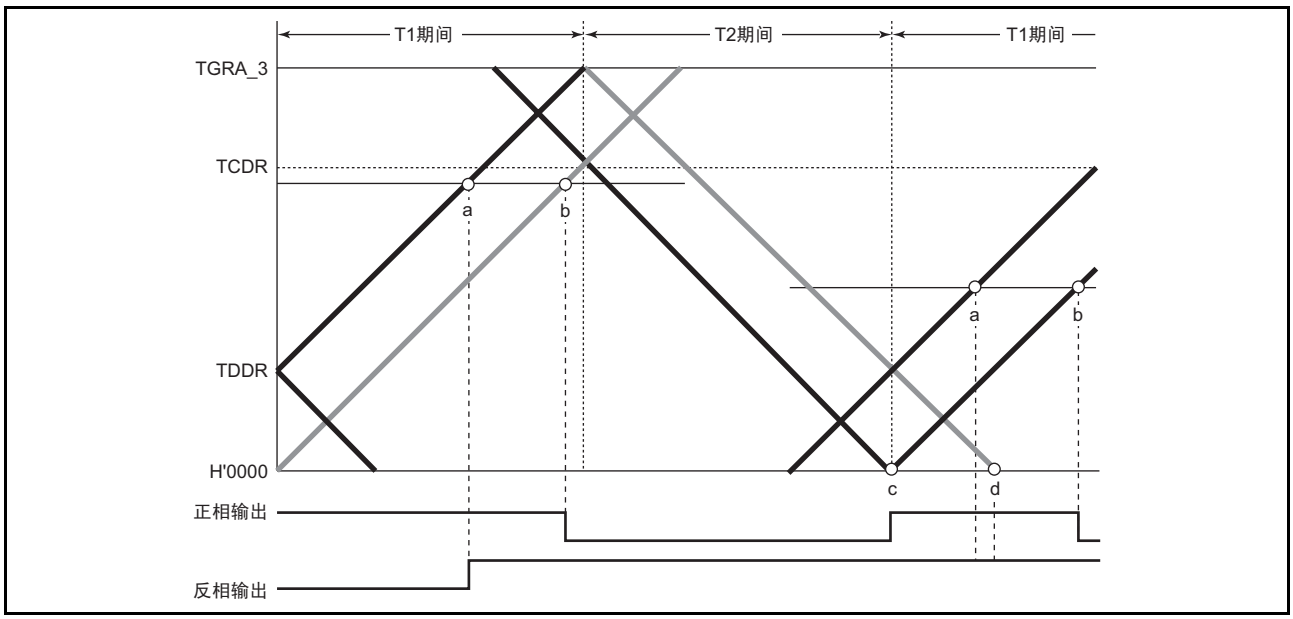


图 11.50　互补 PWM 模式的 0%、 100% 波形输出例子 （2）

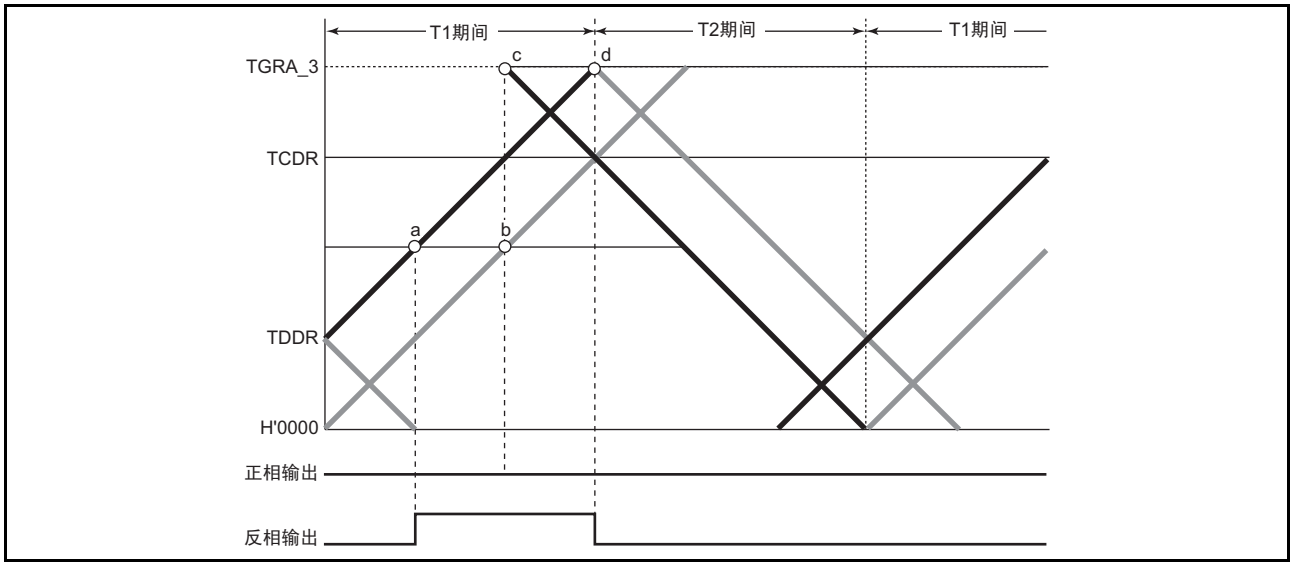


图 11.51　互补 PWM 模式的 0%、 100% 波形输出例子 （3）

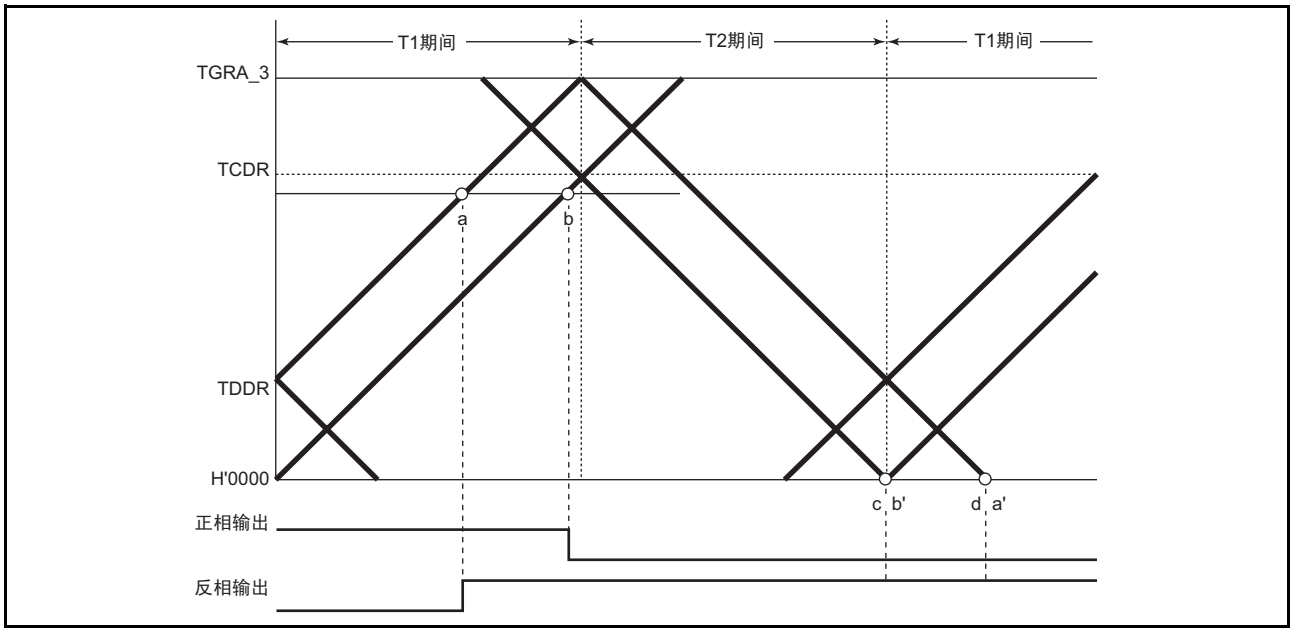


图 11.52　互补 PWM 模式的 0%、 100% 波形输出例子 （4）

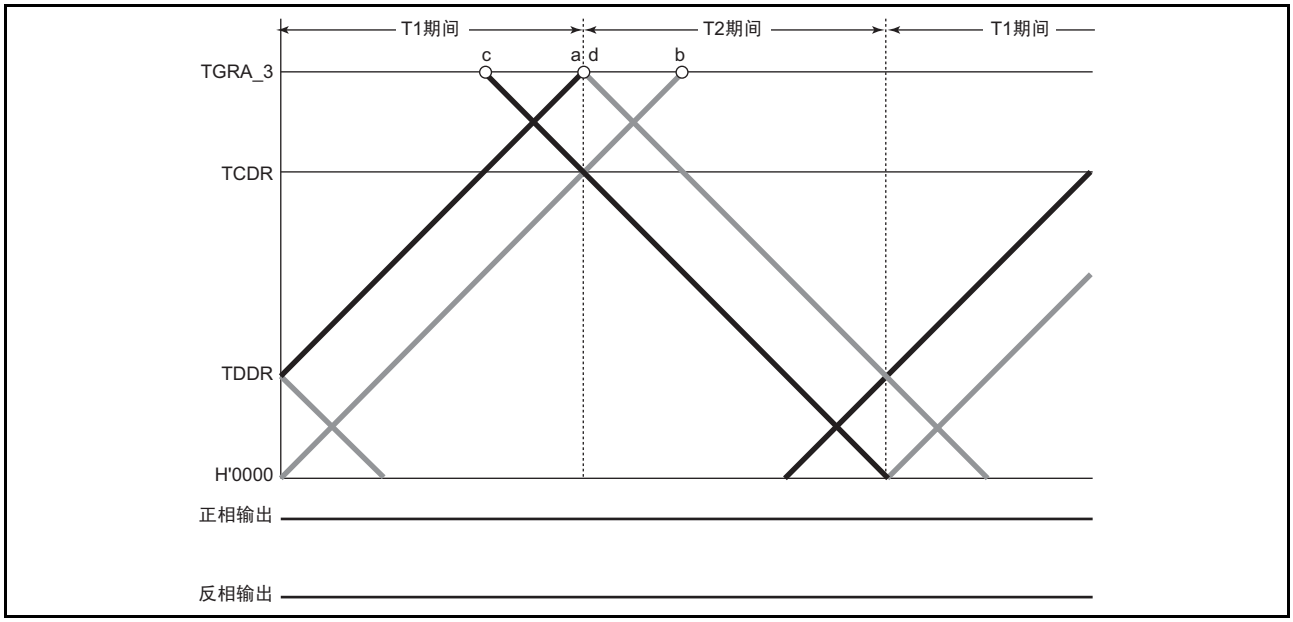


图 11.53　互补 PWM 模式的 0%、 100% 波形输出例子 （5）

（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。

（l） 与 PWM 周期同步的交替输出

在互补 PWM 模式中，能通过将定时器输出控制寄存器 （TOCR）的 PSYE 位置 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。

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为 TIOC3A 引脚。初始输出为 1 输出。

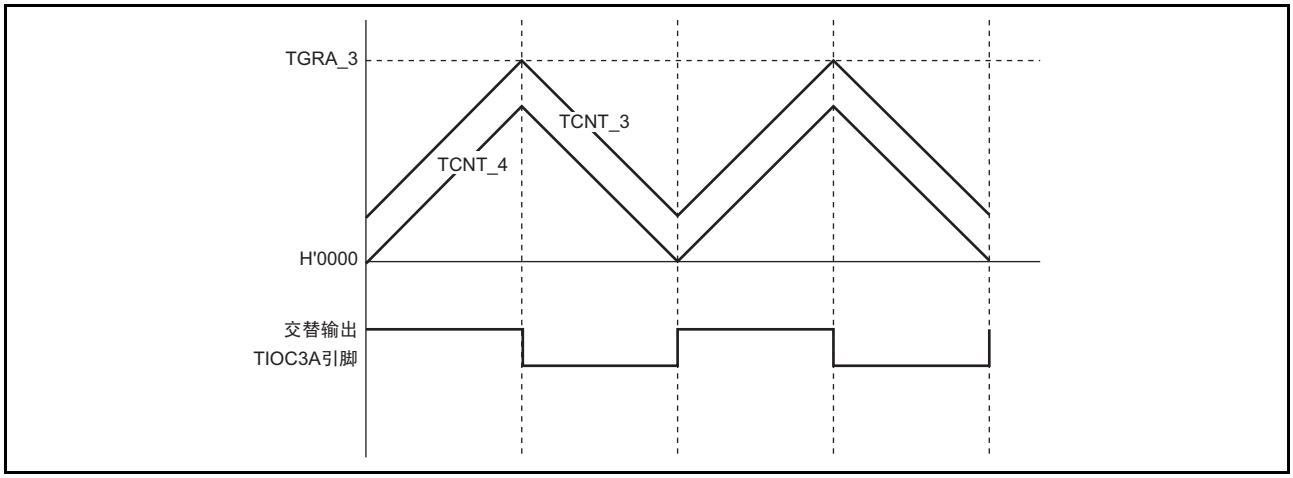


图 11.54　与 PWM 输出同步的交替输出的波形例子

（m）通过其他通道进行的计数器清除

在互补 PWM 模式中，通过定时器同步寄存器 （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。

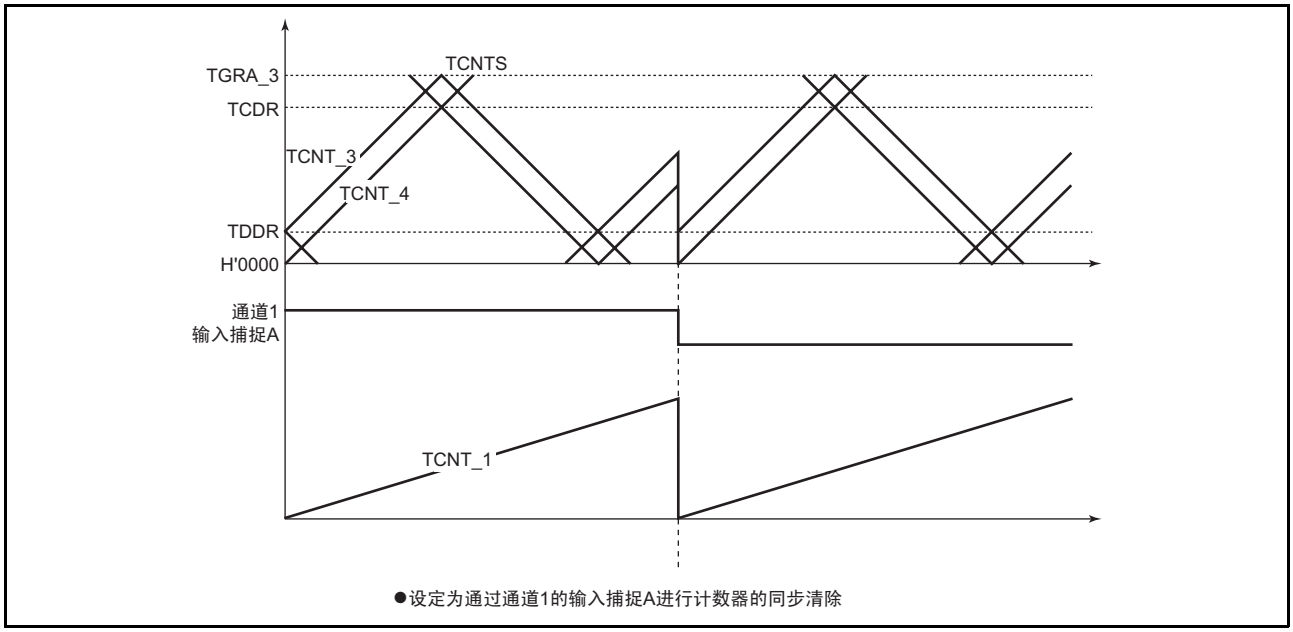


图 11.55　与其他通道同步的计数器清除

（n） 互补 PWM 模式中，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 （10）、（11）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。

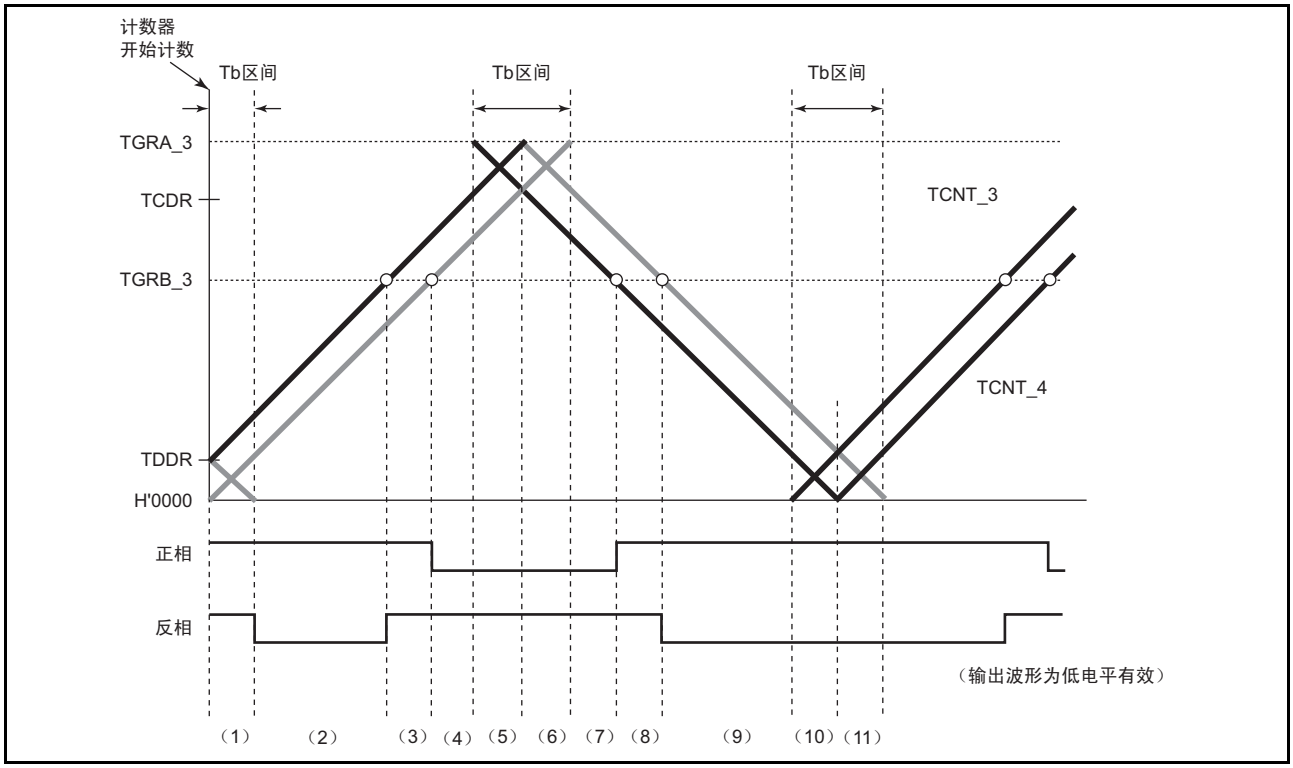


图 11.56　同步计数器清除的时序

- 互补PWM模式中，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，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。

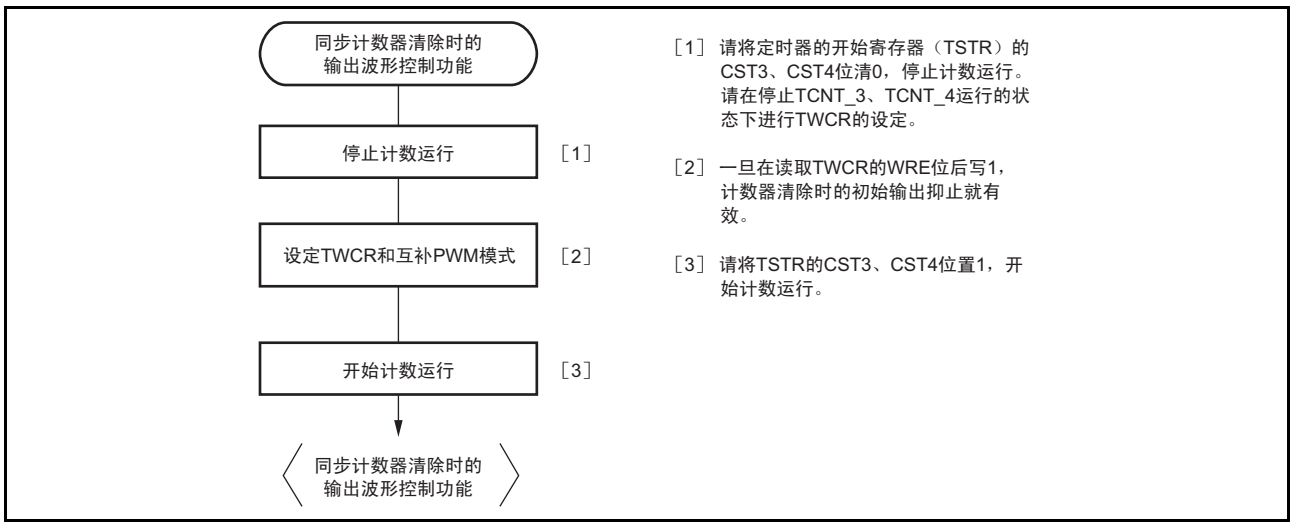


图 11.57　互补 PWM 模式中，同步计数器清除时的输出波形控制的设定步骤例子

- 互补PWM模式中，同步计数器清除时的输出波形控制的运行例子
  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。

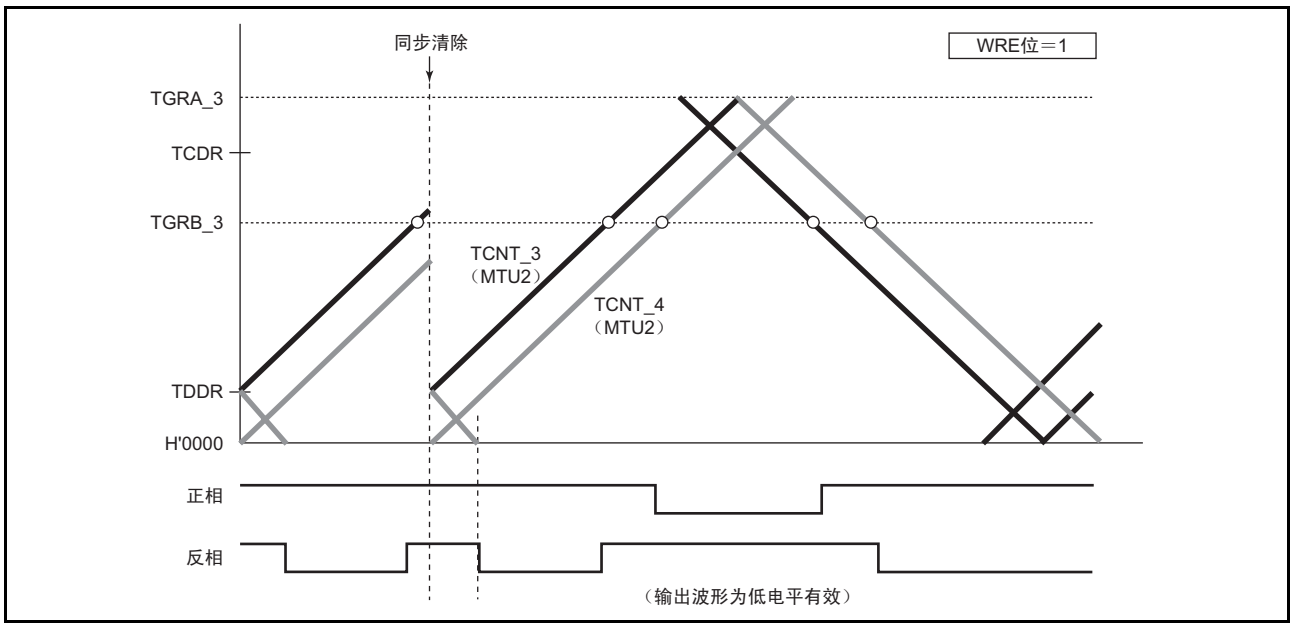


图 11.58　在递增计数中的空载时间发生同步清除
（图 11.56 的时序 （3）、 MTU2 的 TWCR 寄存器的 WRE 位 =1）

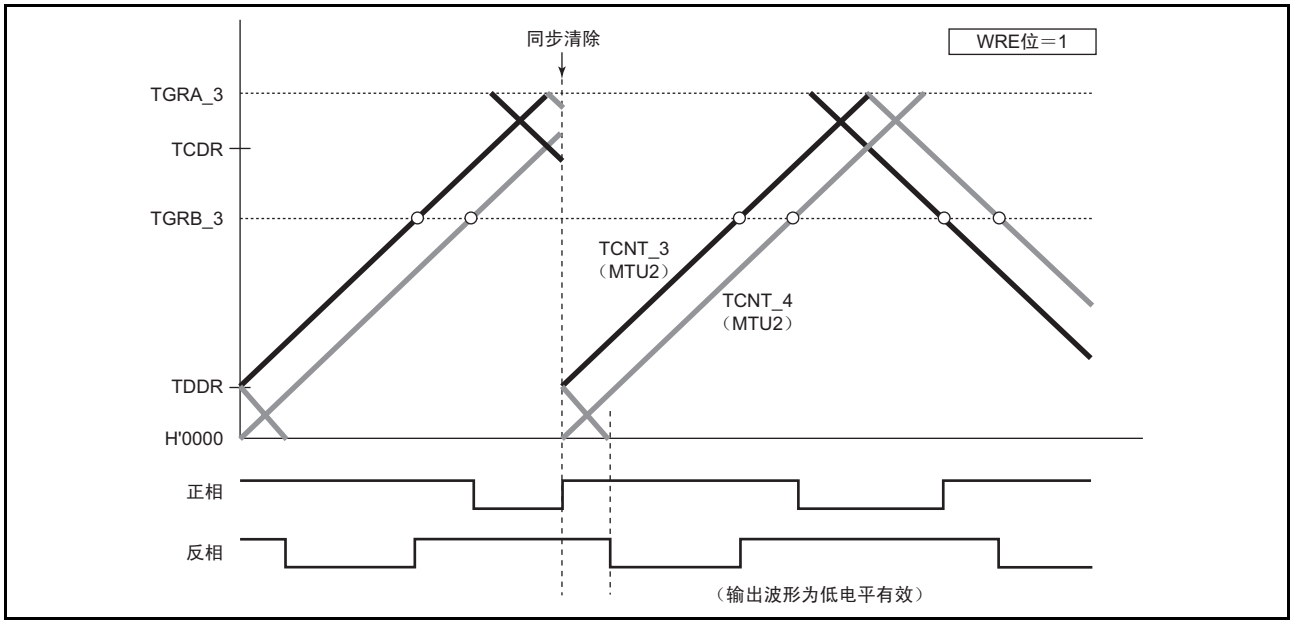


图 11.59　在波峰的 Tb 区间发生同步清除
（图 11.56 的时序 （6）、 MTU2 的 TWCR 寄存器的 WRE 位 =1）

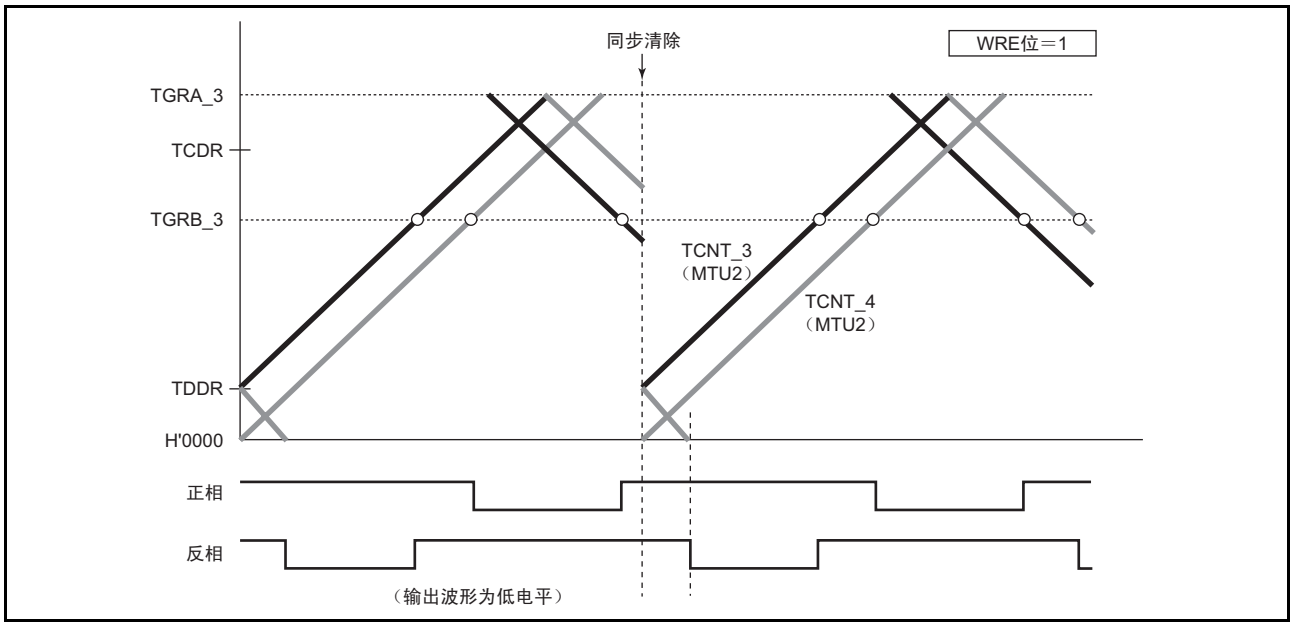


图 11.60　在递减计数中的空载时间发生同步清除
（图 11.56 的时序 （8）、 TWCR 寄存器的 WRE 位 =1）

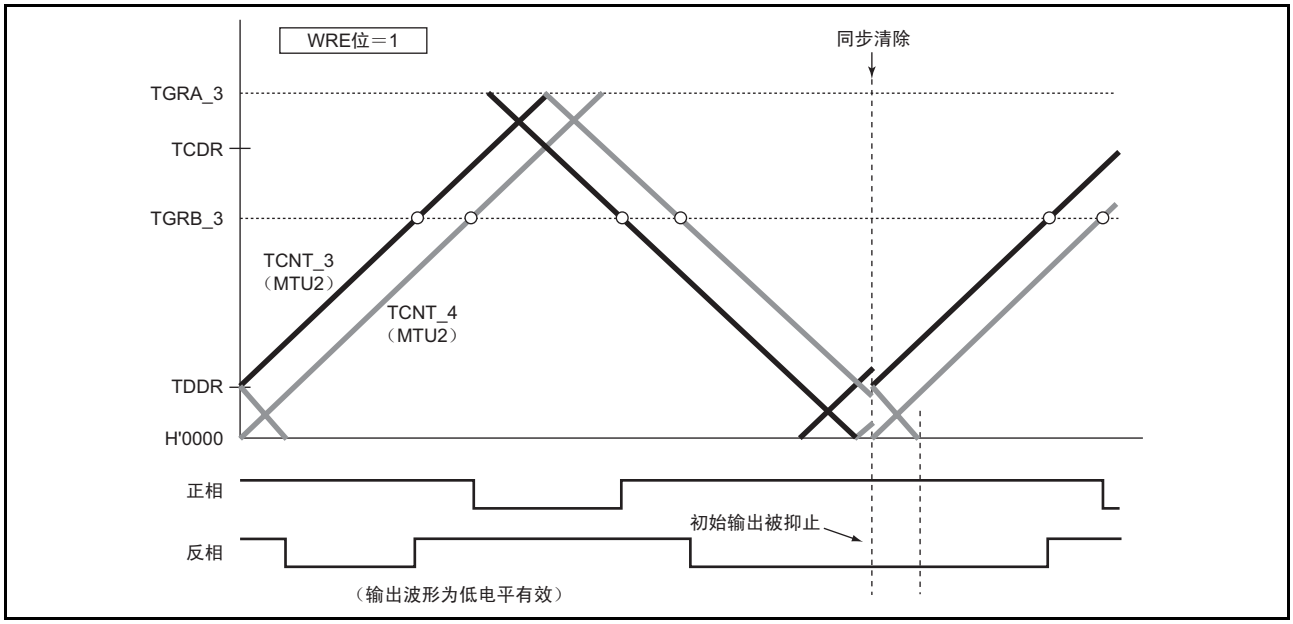


图 11.61　在波谷的 Tb 区间发生同步清除
（图 11.56 的时序 （11）、 TWCR 寄存器的 WRE 位 =1）

（o） 通过 TGRA_3 的比较匹配清除计数器

在互补 PWM 模式中，能通过设定定时器波形控制寄存器（TWCR）的 CCE 位，在 TGRA_3 的比较匹配时清除 TCNT_3、 TCNT_4 和 TCNTS。

运行例子如图 11.62 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。（不能将定时器同步寄存器 （TSYR）的 SYNC0 ～ SYNC4 位和定时器同步清除寄存器 （TSYCR）的 CE0A/B/C/D、 CE1A/B/C/D 位设定为 1）
3. PWM 占空比不能设定为 H'0000。
4. 请勿将定时器输出控制寄存器 1 （TOCR1）的 PSYE 位设定为 1。

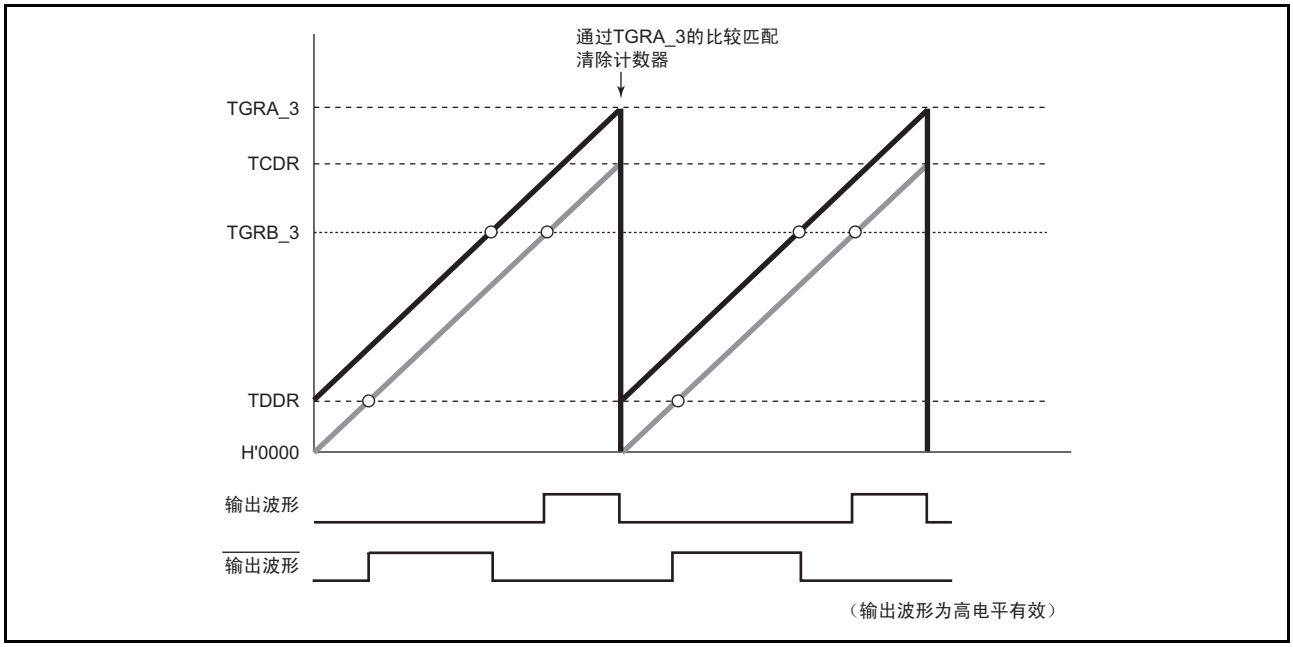


图 11.62　通过 TGRA_3 的比较匹配清除计数器的运行例子

（p） AC 同步马达 （无刷 DC 马达）的驱动波形输出例子

在互补 PWM 模式中，能使用定时器门控寄存器（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 11.63 ～图 11.66 所示。

为了转换 3 相无刷 DC 马达的输出相，在使用霍尔器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通道 0 的定时器输入引脚 TIOC0A、TIOC0B、TIOC0C 引脚（请通过 PFC 进行设定）。如果在 TIOC0A、 TIOC0B、 TIOC0C 这 3 个引脚检测到边沿，输出的 ON/OFF 就自动转换。

当 FB 位为 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 相输出引脚输出驱动波形。

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，进行斩波输出。在 N 位或者 P 位为 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，能通过定时器输出控制寄存器 （TOCR）的 OLSN 位和 OLSP 位进行设定。

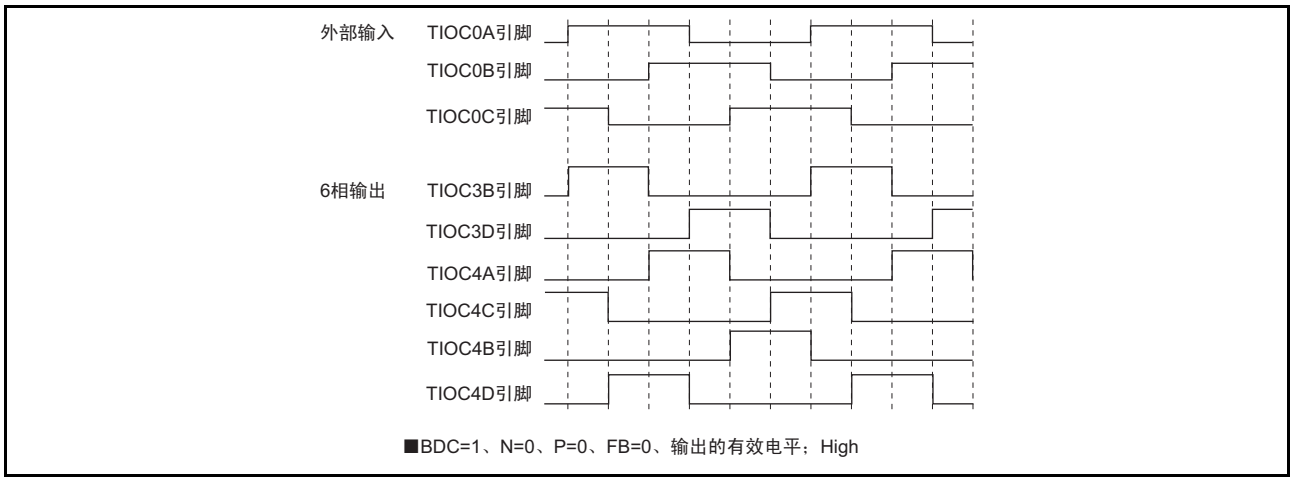


图 11.63　通过外部输入进行输出相转换的运行例子 （1）

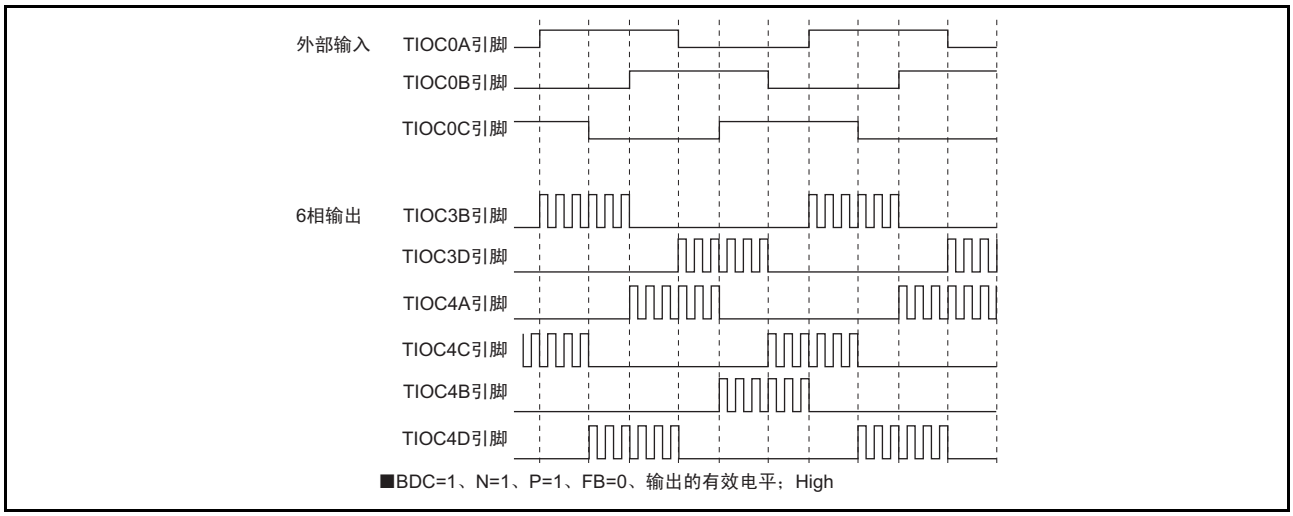


图 11.64　通过外部输入进行输出相转换的运行例子 （2）

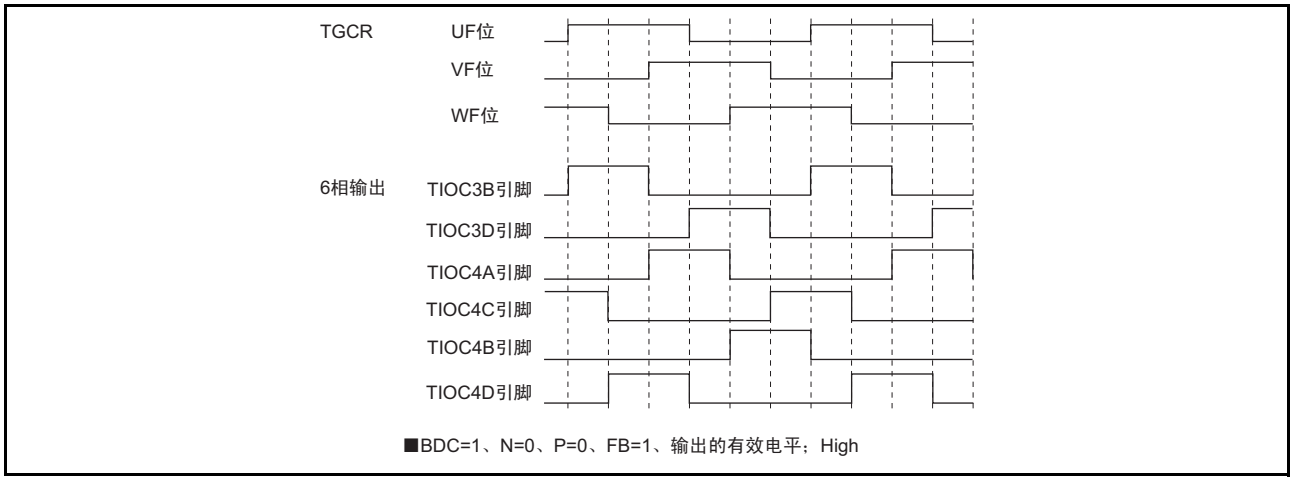


图 11.65　通过设定 UF、 VF、 WF 位进行输出相转换的运行例子 （1）

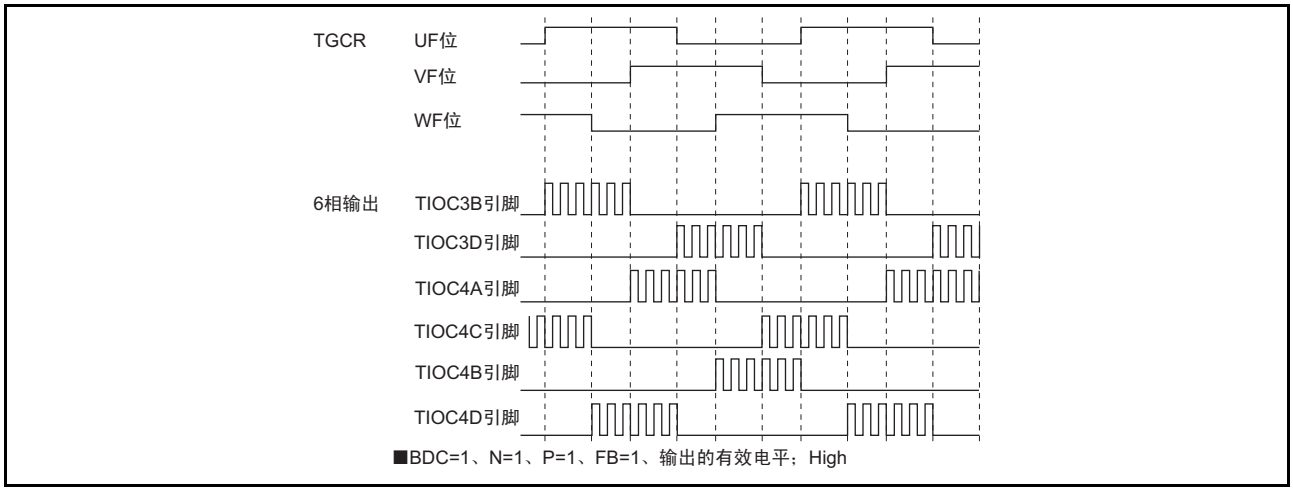


图 11.66　通过设定 UF、 VF、 WF 位进行输出相转换的运行例子 （2）

（q） 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。

能通过将定时器中断允许寄存器（TIER）的 TTGE 位置 1，设定 A/D 转换的开始请求。并能通过将 TIER_4 的 TTGE2 位置 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）及 TCIV_4 （波谷的中断）跳过 7 次中断。

通过设定定时器缓冲传送寄存器 （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的 A/D 转换开始请求联动，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”。

请通过设定 TIER_3、 TIER_4 寄存器将 TGIA_3 和 TCIV_4 中断请求设定为禁止状态后，并且在不发生由比较匹配引起的 TGFA_3、 TCFV_4 标志置位的状态下，设定定时器跳过中断设定寄存器 （TITCR）。另外，在更改跳过次数前，请将 T3AEN 和 T4VEN 位置 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 11.67 和图 11.68 所示。

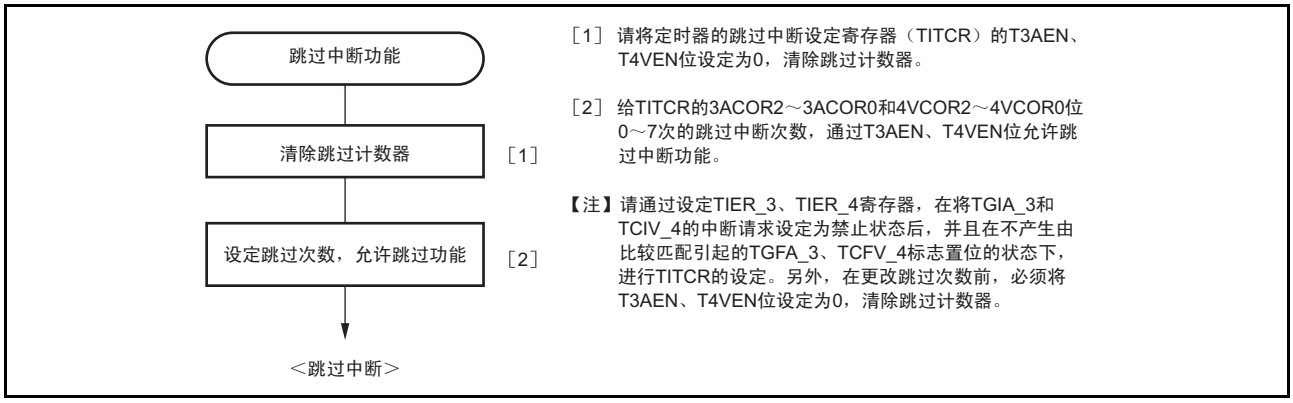


图 11.67　跳过中断功能的设定步骤例子

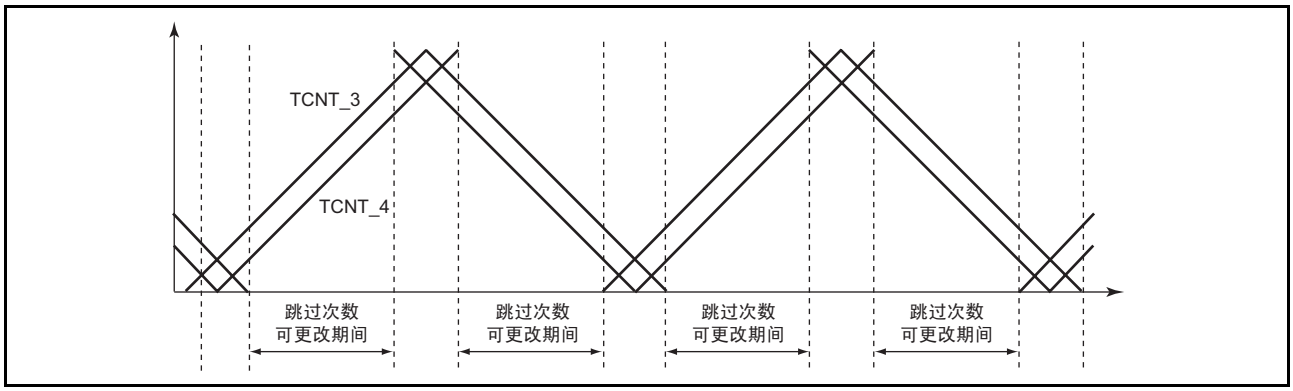


图 11.68　跳过中断次数的可更改期间

（b） 跳过中断功能的运行例子

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子如图 11.69 所示。

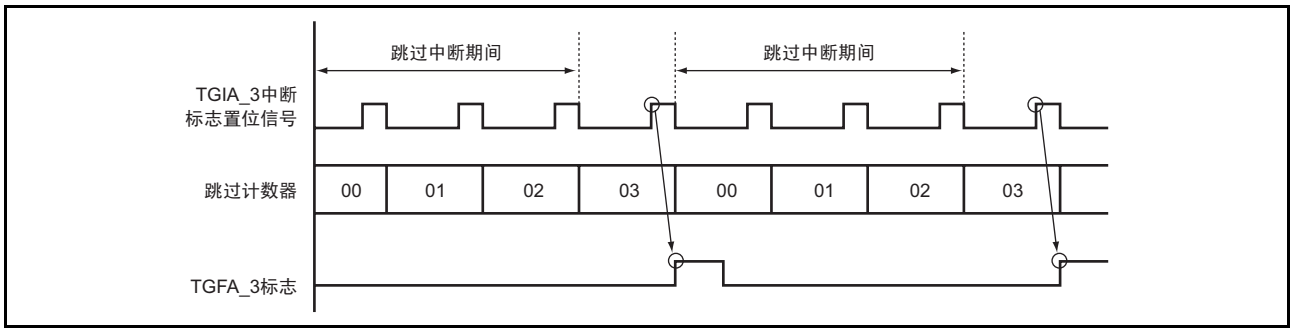


图 11.69　跳过中断功能的运行例子

（c） 与跳过中断功能联动的缓冲传送控制

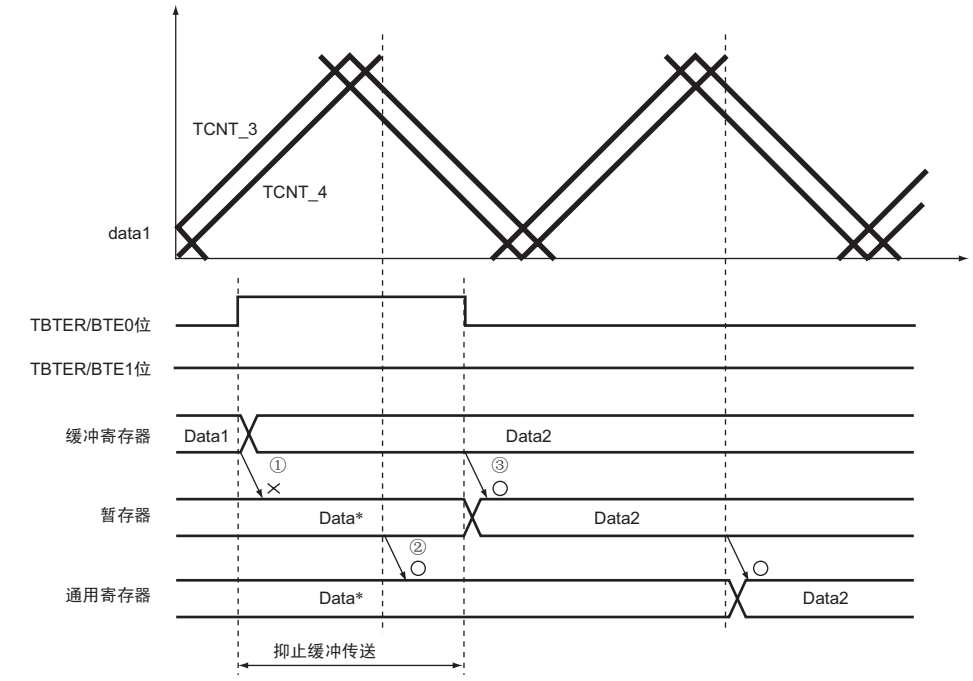
通过设定定时器缓冲传送设定寄存器（TBTER）的 BTE1、BTE0 位，可选择在互补 PWM 模式中是否进行缓冲寄存器到暂存器的缓冲传送，或者选择是否与跳过中断功能联动。

设定为抑制缓冲传送 （BTE1=0、 BTE0=1）时的运行例子如图 11.70 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 11.71 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，将定时器跳过中断设定寄存器（TITCR）的 T3AEN 位、T4VEN 位、T3AEN/T4VEN 位设定为 1 时，各缓冲传送允许期间不同。 TITCR 的 T3AEN、 T4VEN 位的设定和缓冲传送允许期间的关系如图 11.72 所示。

【注】 本功能请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、 4VCOR）设定为0时）必须设定为缓冲传送不与跳过中断功能联动 （将定时器缓冲传送设定寄存器 （TBTER）的BTE1设定为0），否则就不进行缓冲传送。



【符号说明】

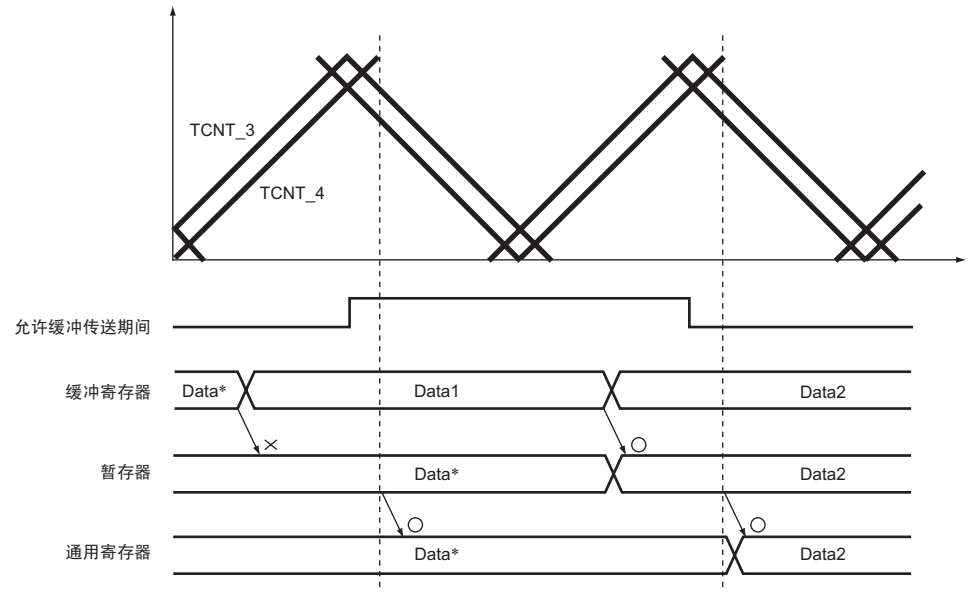
① 在缓冲传送的抑止期间中（TBTER的BTE1/0位设定为0、1）
缓冲寄存器的值不传送到暂存器。

② 即使在缓冲传送的抑止期间中，暂存器的值也传送到通用寄存器。

③ 在解除缓冲传送的抑止后，缓冲寄存器的值传送到暂存器。

【注】* 选择在波峰进行缓冲传送时

图 11.70　设定为抑制缓冲传送 （BTE1=0、 BTE0=1）时的运行例子



【注】* 选择在波峰进行缓冲传送跳过次数设定为3次T3AEN设定为1

图 11.71　设定为缓冲传送与跳过中断功能联动 （BTE1=1、 BTE0=0）时的运行例子

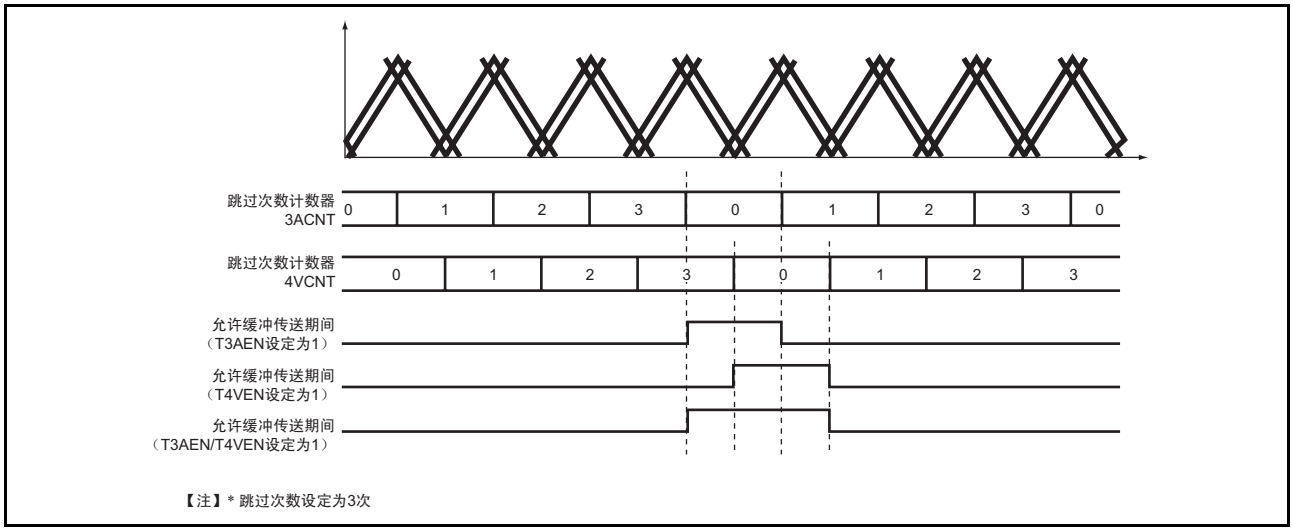


图 11.72　定时器跳过中断设定寄存器 （TITCR）的 T3AEN、 T4VEN 位的设定和缓冲传送允许期间的关系

（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的保护功能。

（a） 寄存器、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 / 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 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取对象寄存器时，读出不定值并且写操作无效。

11.4.9　A/D 转换开始请求的延迟功能

能通过设定通道 4 的定时器 A/D 转换开始请求控制寄存器 （TADCR）、定时器 A/D 转换开始请求周期寄存器 （TADCORA_4、 TADCORB_4）和定时器 A/D 转换开始请求周期缓冲寄存器 （TADCOBRA_4、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 匹配时， 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/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11.73所示。

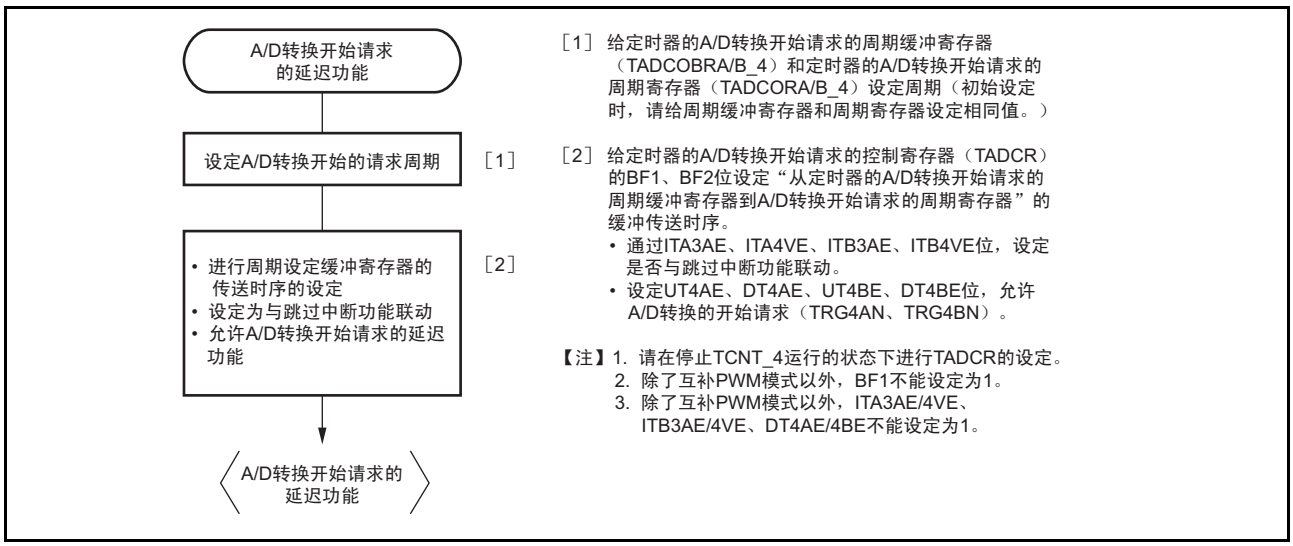


图 11.73　A/D 转换开始请求的延迟功能的设定步骤例子

- A/D转换开始请求的延迟功能的基本运行例子
  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11.74所示。

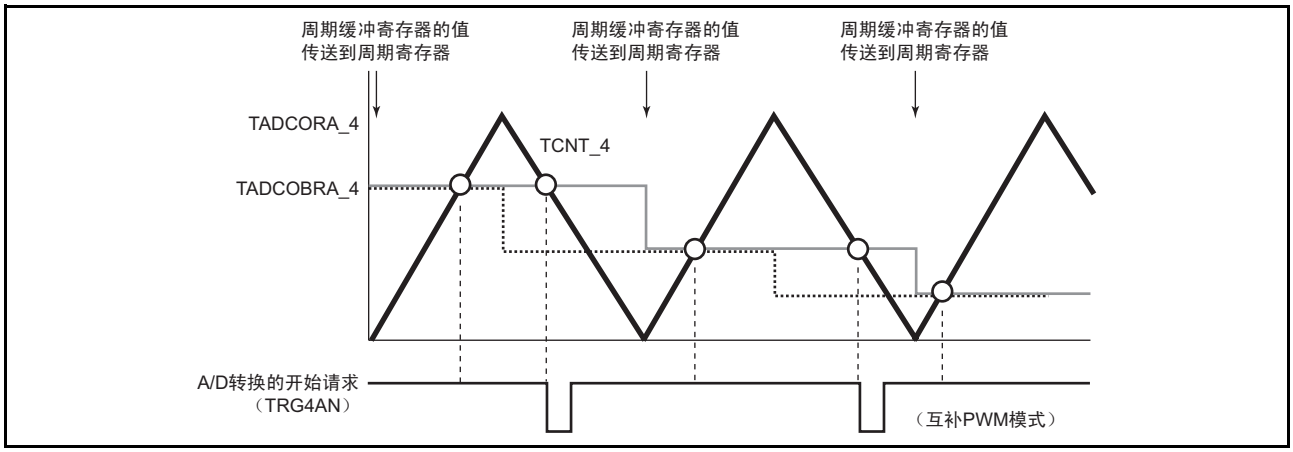


图 11.74　A/D 转换的开始请求信号（TRG4AN）的基本运行例子

- 缓冲传送
  通过将数据写到定时器A/D转换开始请求周期设定缓冲寄存器 （TADCOBRA/B_4），更新定时器A/D转换开始请求周期设定寄存器 （TADCORA/B_4）的数据。能通过设定定时器A/D转换开始请求控制寄存器 （TADCR_4）的BF1、 BF0位，选择从定时器A/D转换开始请求周期设定缓冲寄存器到定时器A/D转换开始请求周期设定寄存器的传送时序。

- 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A/D转换开始请求控制寄存器（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、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11.75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11.76所示。

【注】 请将本功能和跳过中断功能配合使用。
在禁止跳过中断功能时 （将定时器跳过中断设定寄存器 （TITCR）的T3AEN、 T4VEN位或者TITCR跳过次数的设定位 （3ACOR、 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A/D转换开始请求控制寄存器 （TADCR）的ITA3AE、 ITA4VE、 ITB3AE和ITB4VE位设定为0）。

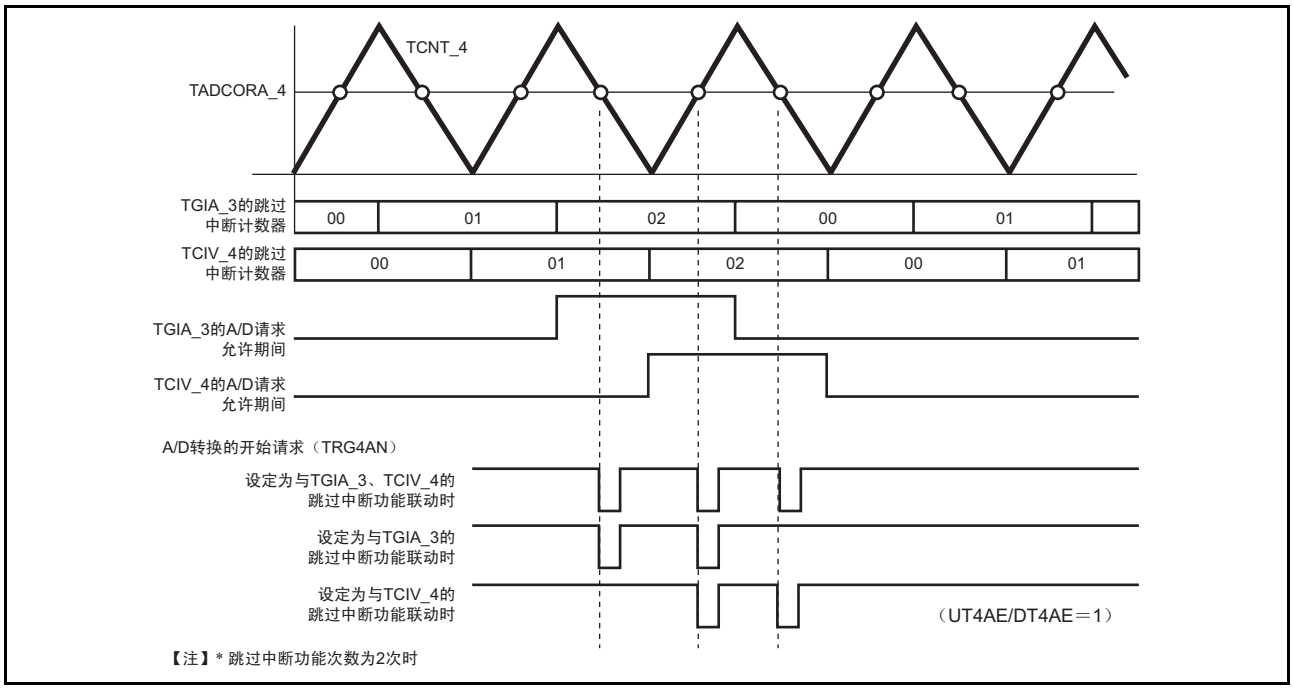


图 11.75　与跳过中断功能联动时 A/D 转换的开始请求信号 （TRG4AN）的运行例子

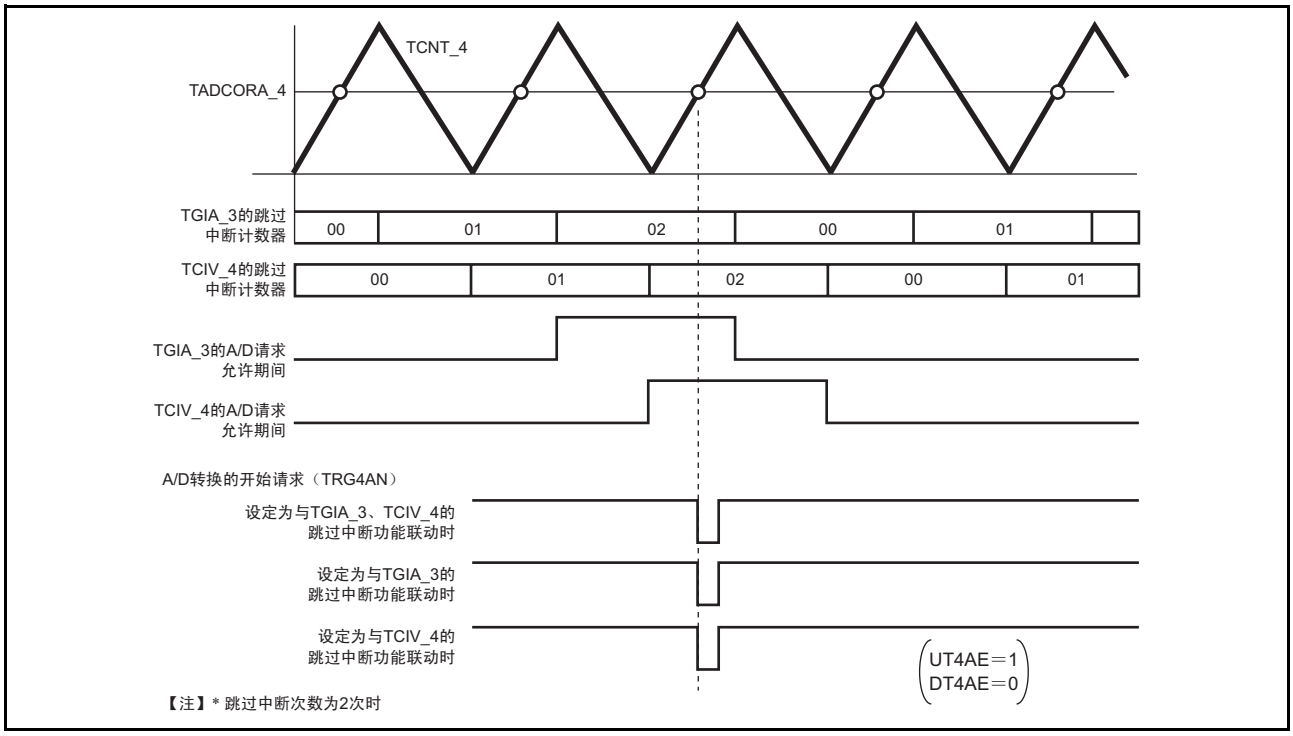


图 11.76　与跳过中断功能联动时 A/D 转换的开始请求信号 （TRG4AN）的运行例子

11.4.10　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 / 波谷” 将 TCNT 的值保存到 TGR。通过 TIOR 选择要捕捉到 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” 在 TGR 进行捕捉的运行例子如图 11.77 所示。

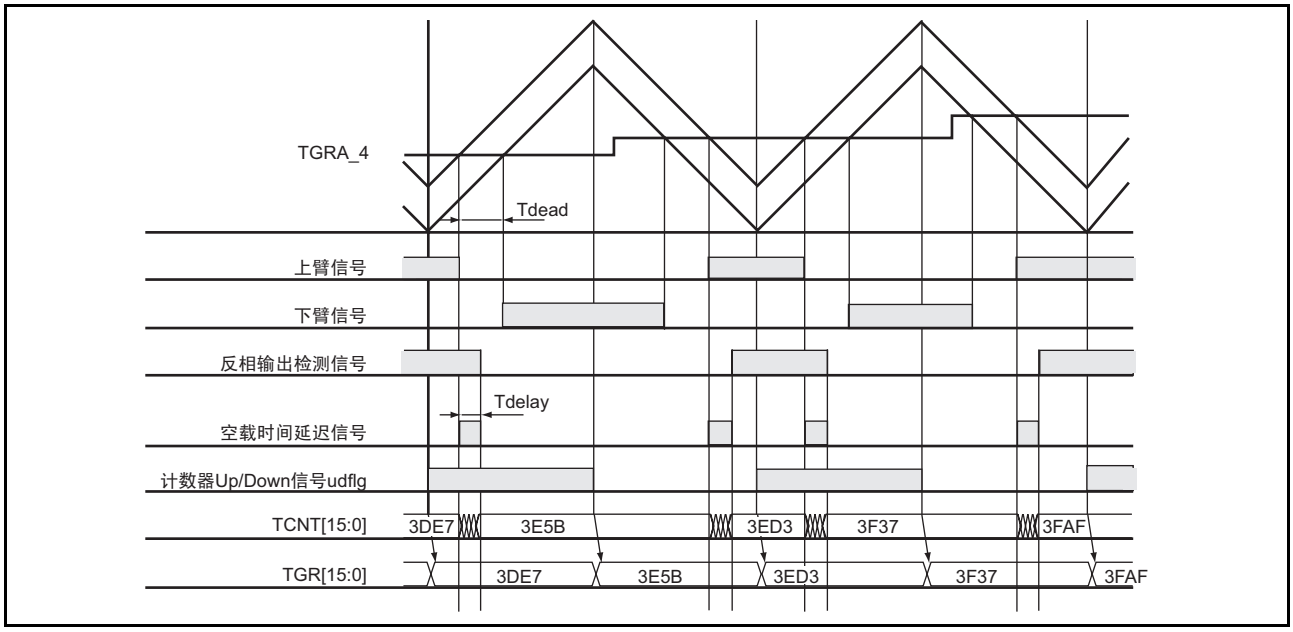


图 11.77　互补 PWM 在 “波峰 / 波谷” 的 TCNT 捕捉运行

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。

当产生中断源时， 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。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 11.55 所示。

表 11.55　MTU2 中断源

通道	名称	中断源	中断标志	DMAC 的启动	优先级
0	TGIA_0	TGRA_0 的输入捕捉 / 比较匹配	TGFA_0	可	高
	TGIB_0	TGRB_0 的输入捕捉 / 比较匹配	TGFB_0	不可	
	TGIC_0	TGRC_0 的输入捕捉 / 比较匹配	TGFC_0	不可	
	TGID_0	TGRD_0 的输入捕捉 / 比较匹配	TGFD_0	不可	
	TCIV_0	TCNT_0 的上溢	TCFV_0	不可	
	TGIE_0	TGRE_0 的比较匹配	TGFE_0	不可	
	TGIF_0	TGRF_0 的比较匹配	TGFF_0	不可	
1	TGIA_1	TGRA_1 的输入捕捉 / 比较匹配	TGFA_1	可	
	TGIB_1	TGRB_1 的输入捕捉 / 比较匹配	TGFB_1	不可	
	TCIV_1	TCNT_1 的上溢	TCFV_1	不可	
	TCIU_1	TCNT_1 的下溢	TCFU_1	不可	
2	TGIA_2	TGRA_2 的输入捕捉 / 比较匹配	TGFA_2	可	
	TGIB_2	TGRB_2 的输入捕捉 / 比较匹配	TGFB_2	不可	
	TCIV_2	TCNT_2 的上溢	TCFV_2	不可	
	TCIU_2	TCNT_2 的下溢	TCFU_2	不可	
3	TGIA_3	TGRA_3 的输入捕捉 / 比较匹配	TGFA_3	可	
	TGIB_3	TGRB_3 的输入捕捉 / 比较匹配	TGFB_3	不可	
	TGIC_3	TGRC_3 的输入捕捉 / 比较匹配	TGFC_3	不可	
	TGID_3	TGRD_3 的输入捕捉 / 比较匹配	TGFD_3	不可	
	TCIV_3	TCNT_3 的上溢	TCFV_3	不可	低

通道	名称	中断源	中断标志	DMAC 的启动	优先级
4	TGIA_4	TGRA_4 的输入捕捉 / 比较匹配	TGFA_4	可	高
	TGIB_4	TGRB_4 的输入捕捉 / 比较匹配	TGFB_4	不可	▲
	TGIC_4	TGRC_4 的输入捕捉 / 比较匹配	TGFC_4	不可	▼
	TGID_4	TGRD_4 的输入捕捉 / 比较匹配	TGFD_4	不可	
	TCIV_4	TCNT_4 的上溢 / 下溢	TCFV_4	不可	低

【注】 表示刚复位后的初始状态。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，共 18 个输入捕捉 / 比较匹配中断。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　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。

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关系如**表 11.56** 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，在将 TIER_4 的 TTGE2 位置 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，且 TIER 的 TTGE 位已被置 1 的情况下
- 在将 TIER_4 的 TTGE2 位置 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

如果将 A/D 转换开始请求控制寄存器（TADCR）的 TAD4AE、TAD4BE 位置 1，能通过 TADCORA、TADCORB 和 TCNT_4 的匹配产生 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 11.56　各中断源和 A/D 转换开始请求的对应关系

对　　象	中断源	A/D 转换的开设请求
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 11.78 所示。另外，外部时钟运行 （普通模式）时的 TCNT 计数时序如图 11.79 所示，外部时钟运行 （相位计数模式）时的 TCNT 计数时序如图 11.80 所示。

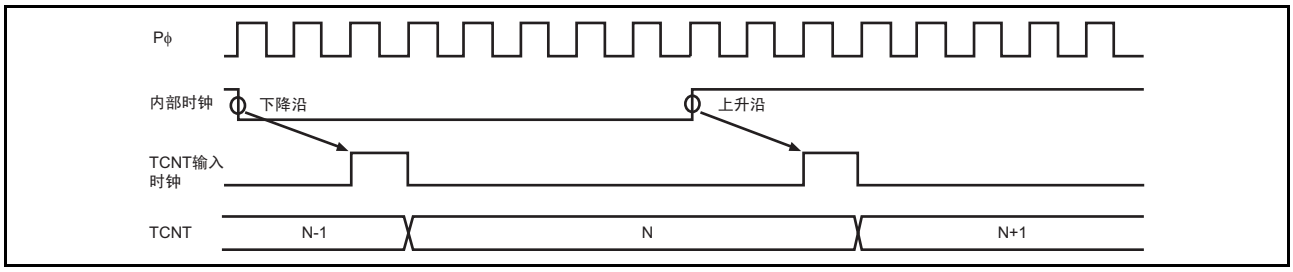


图 11.78　内部时钟运行时的计数时序

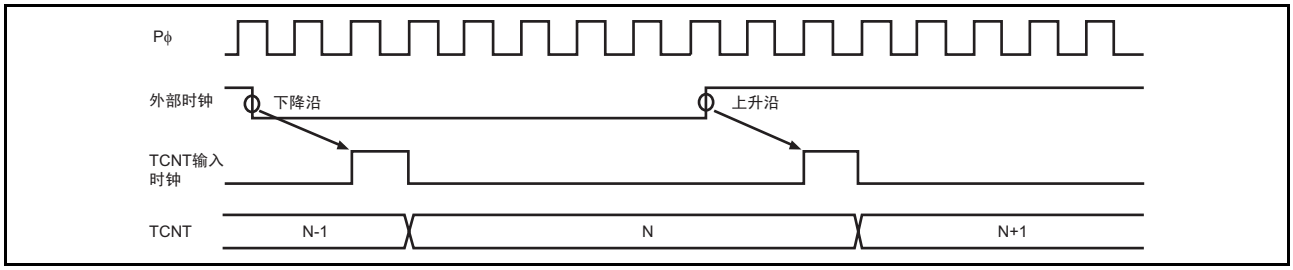


图 11.79　外部时钟运行时的计数时序

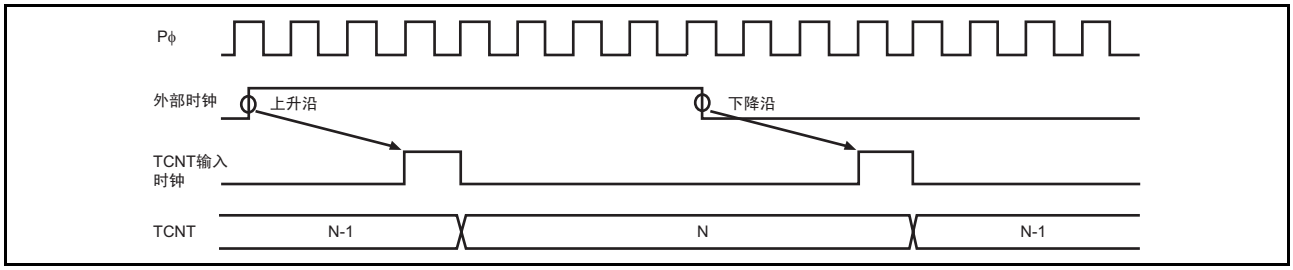


图 11.80　外部时钟运行时的计数时序 （相位计数模式）

（2） 输出比较的输出时序

在 TCNT 和 TGR 匹配的最后状态 （更新 TCNT 匹配后的计数值时）产生比较匹配信号。在发生比较匹配信号时，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 匹配后到产生 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（普通模式、PWM 模式）如图 11.81 所示，输出比较的输出时序（互补 PWM 模式、复位同步 PWM 模式）如图 11.82 所示。

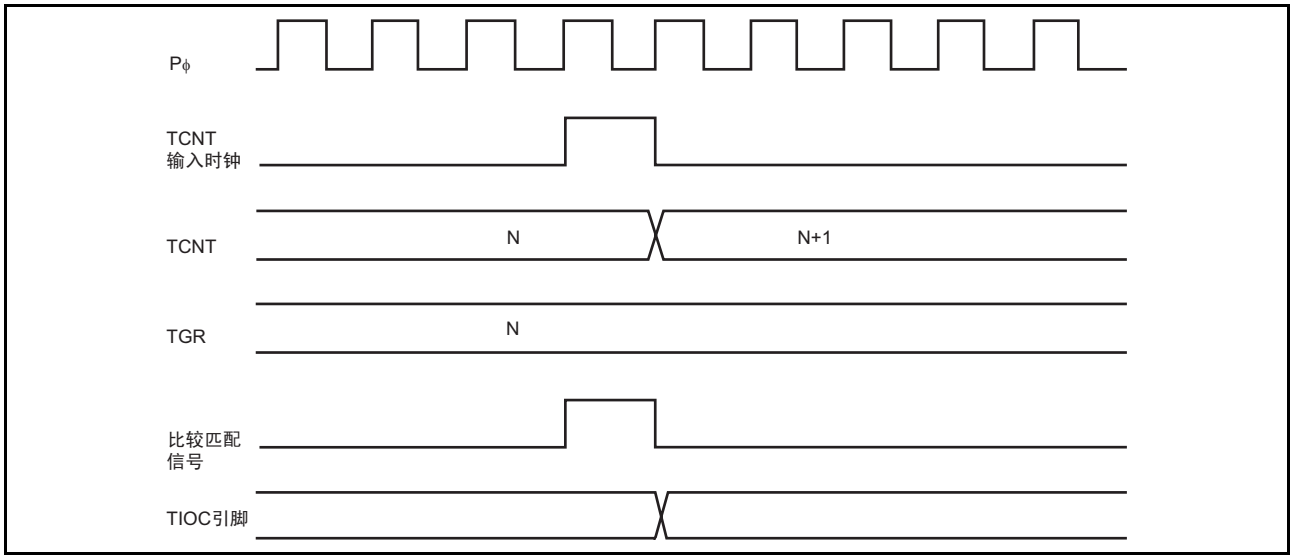


图 11.81　输出比较的输出时序 （普通模式、 PWM 模式）

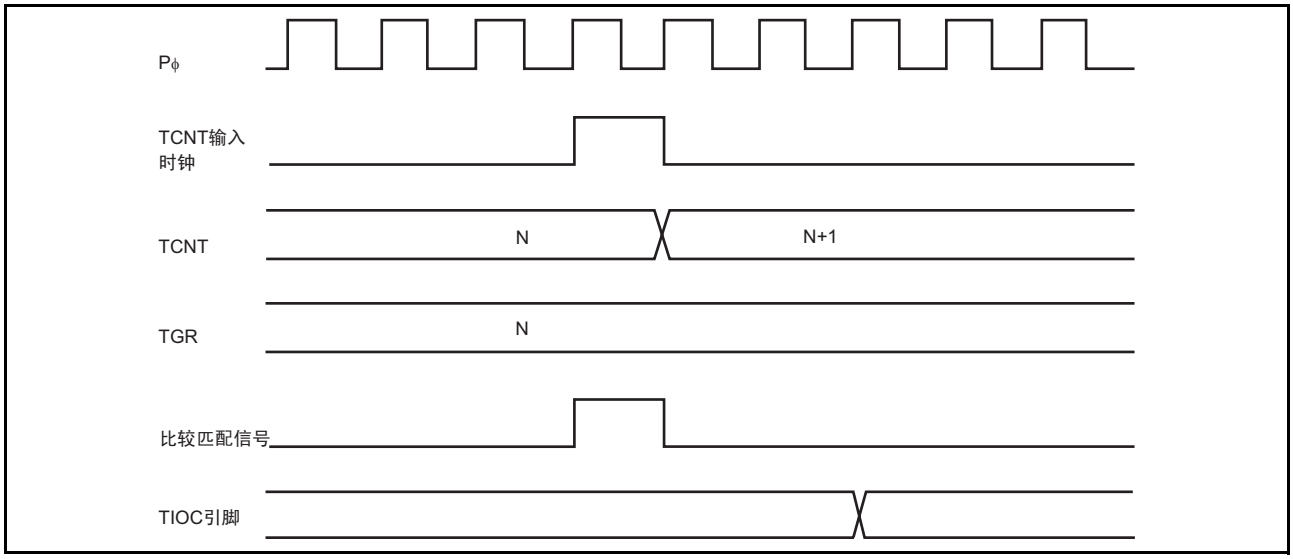


图 11.82　输出比较的输出时序 （互补 PWM 模式、复位同步 PWM 模式）

（3） 输入捕捉信号的时序

输入捕捉的时序如图 11.83 所示。

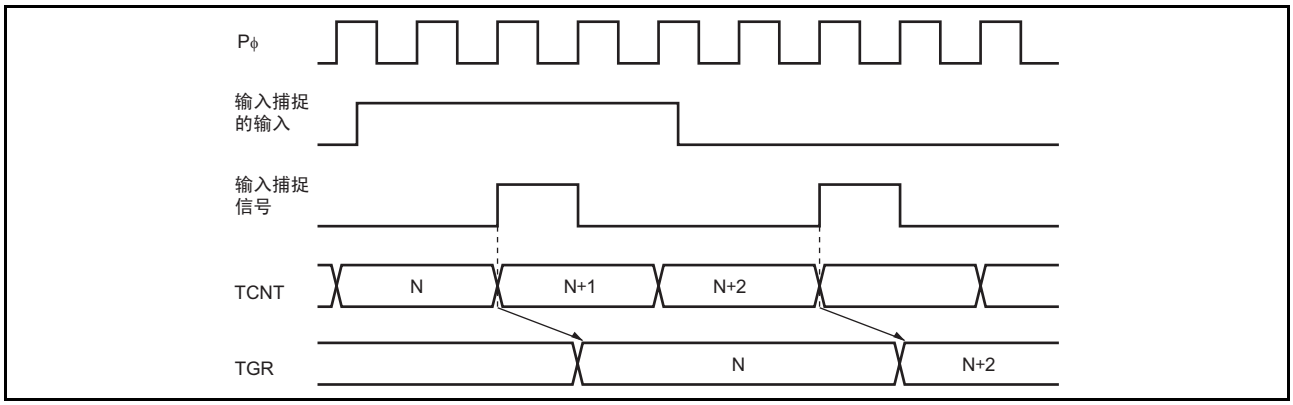


图 11.83　输入捕捉的输入信号时序

（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 11.84 所示。
指定通过输入捕捉进行计数器清除时的时序如图 11.85 所示。

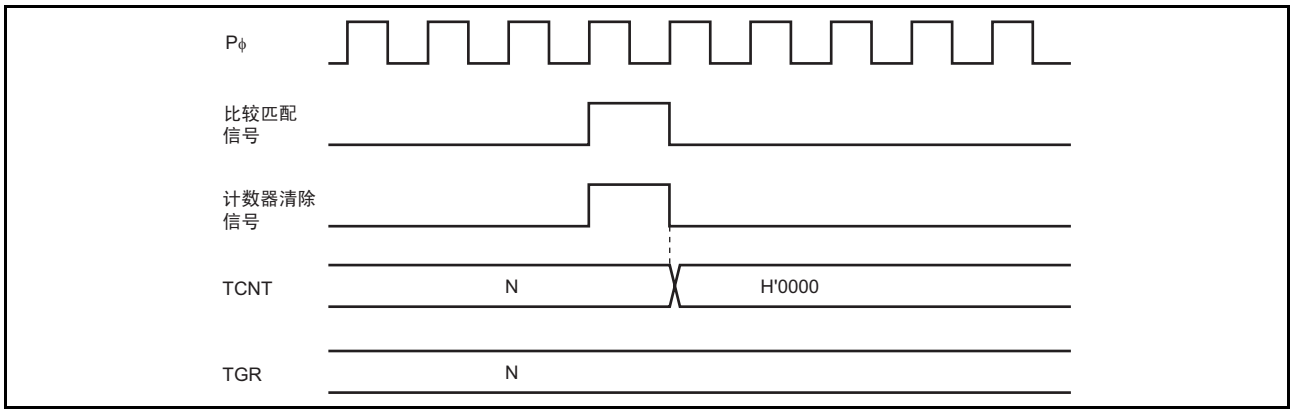


图 11.84　计数器清除的时序 （比较匹配）

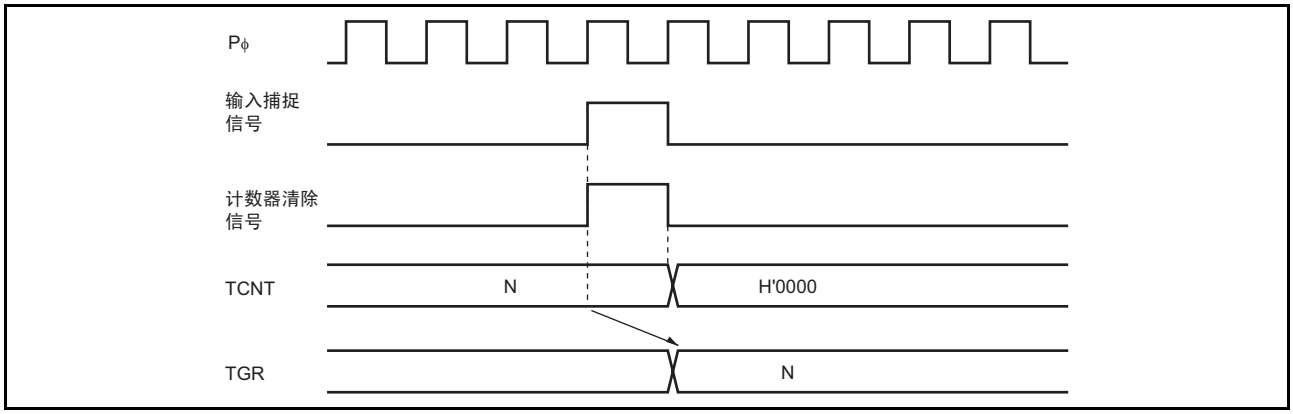


图 11.85　计数器清除的时序 （输入捕捉）

（5） 缓冲运行的时序

缓冲运行的时序如图 11.86 ～图 11.88 所示。

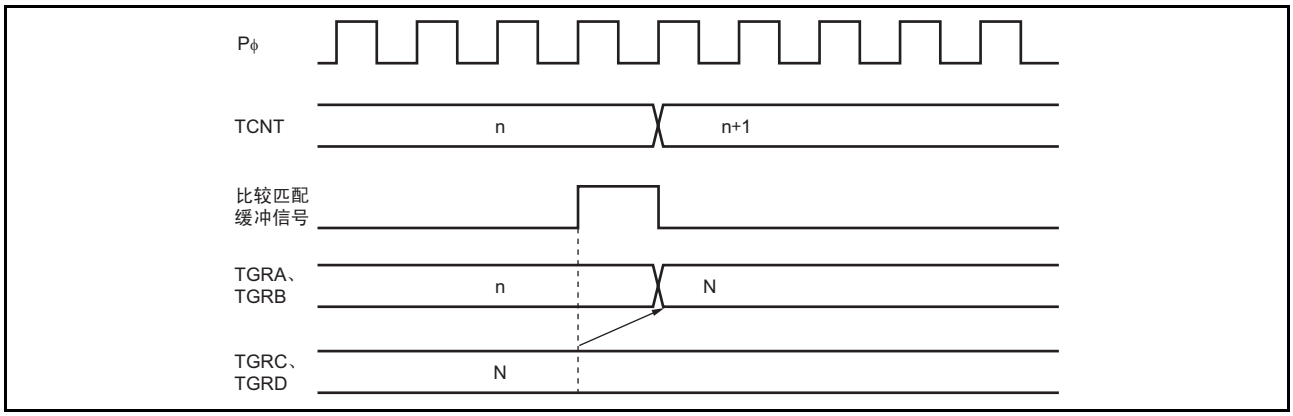


图 11.86　缓冲运行的时序 （比较匹配）

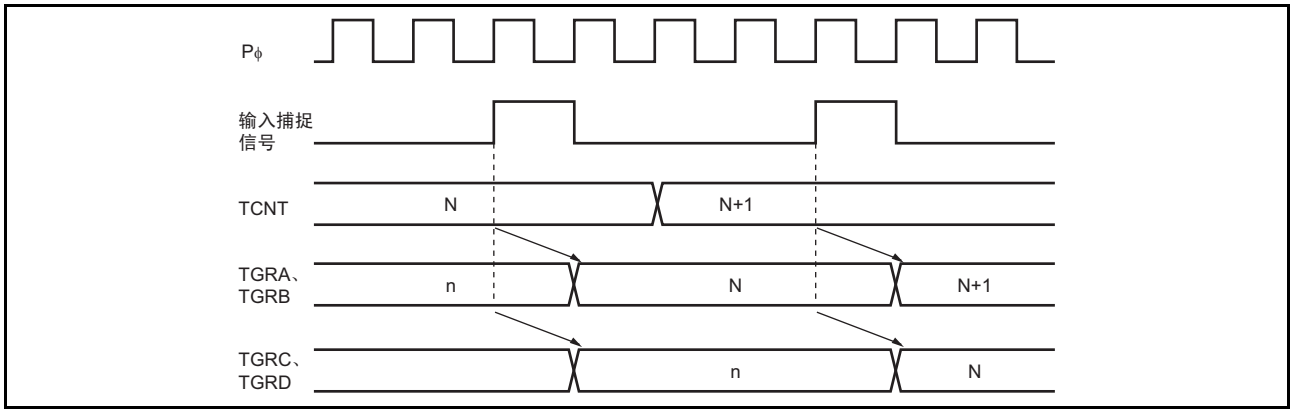


图 11.87　缓冲运行的时序 （输入捕捉）

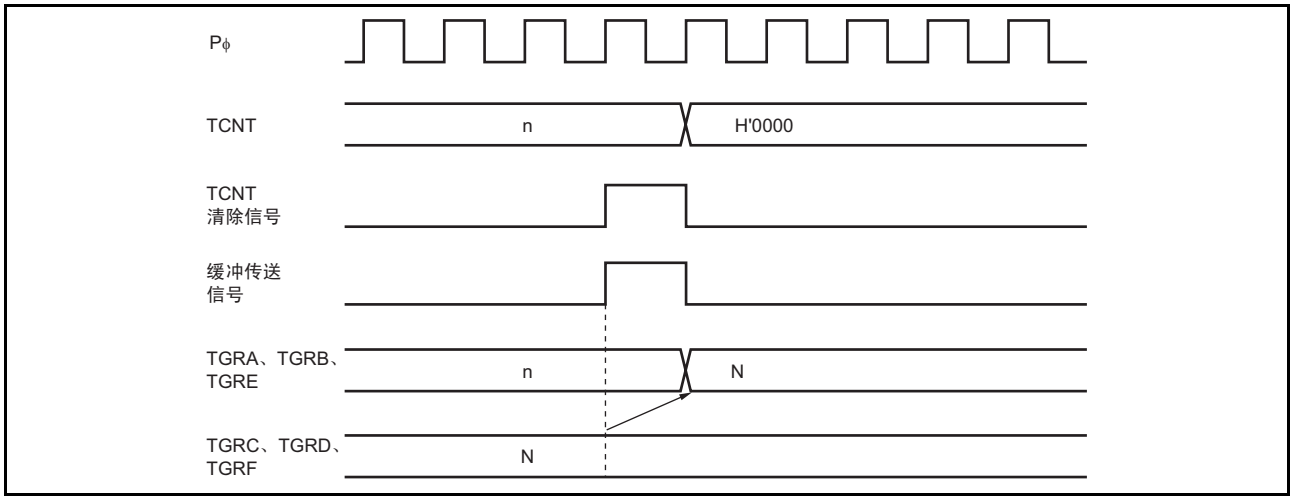


图 11.88　缓冲运行的时序 （清除 TCNT 时）

（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 11.89 ～图 11.91 所示。

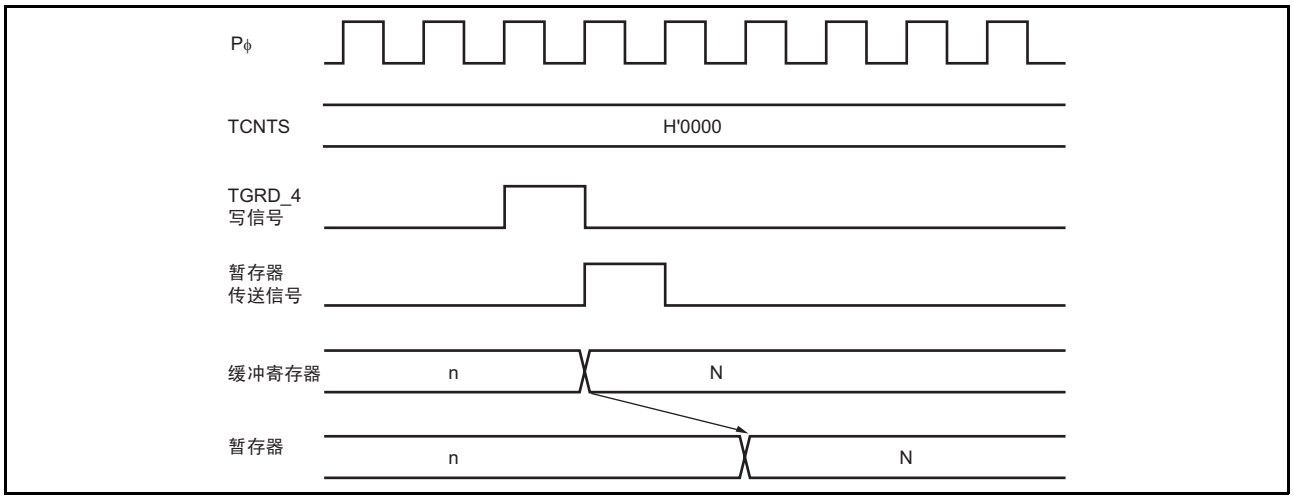


图 11.89　缓冲寄存器到暂存器的传送时序 （TCNTS 停止时）

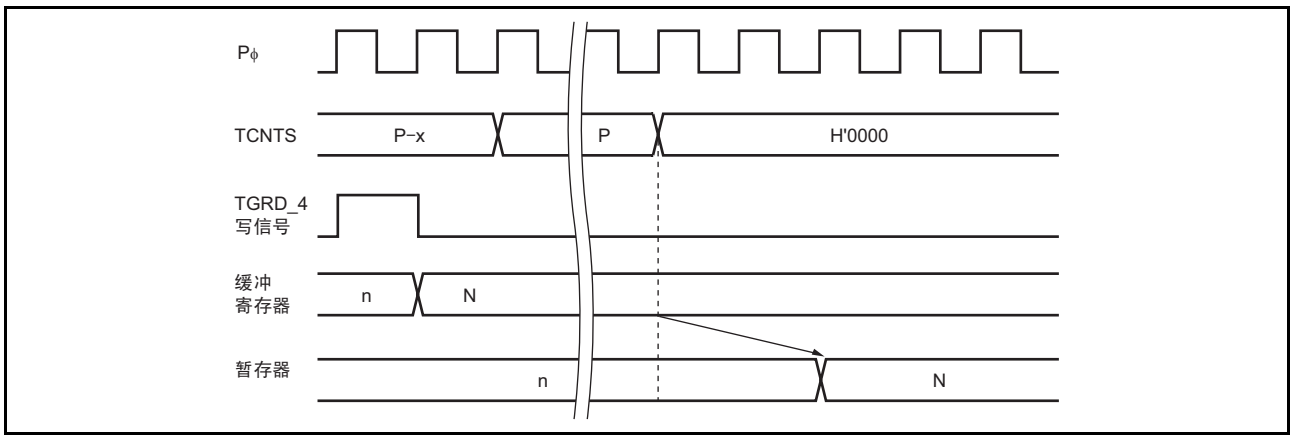


图 11.90　缓冲寄存器到暂存器的传送时序 （TCNTS 运行时）

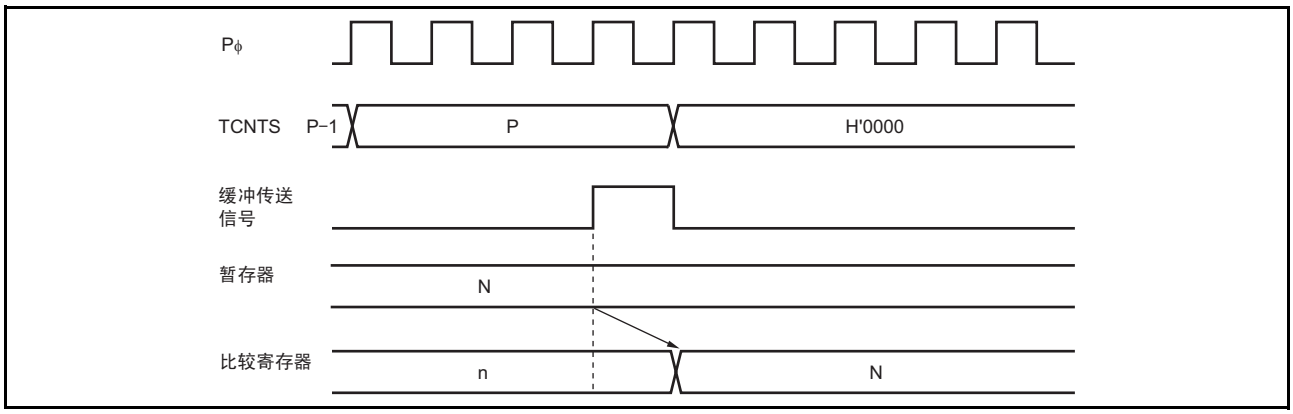


图 11.91　暂存器到比较寄存器的传送时序

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 11.92 所示。

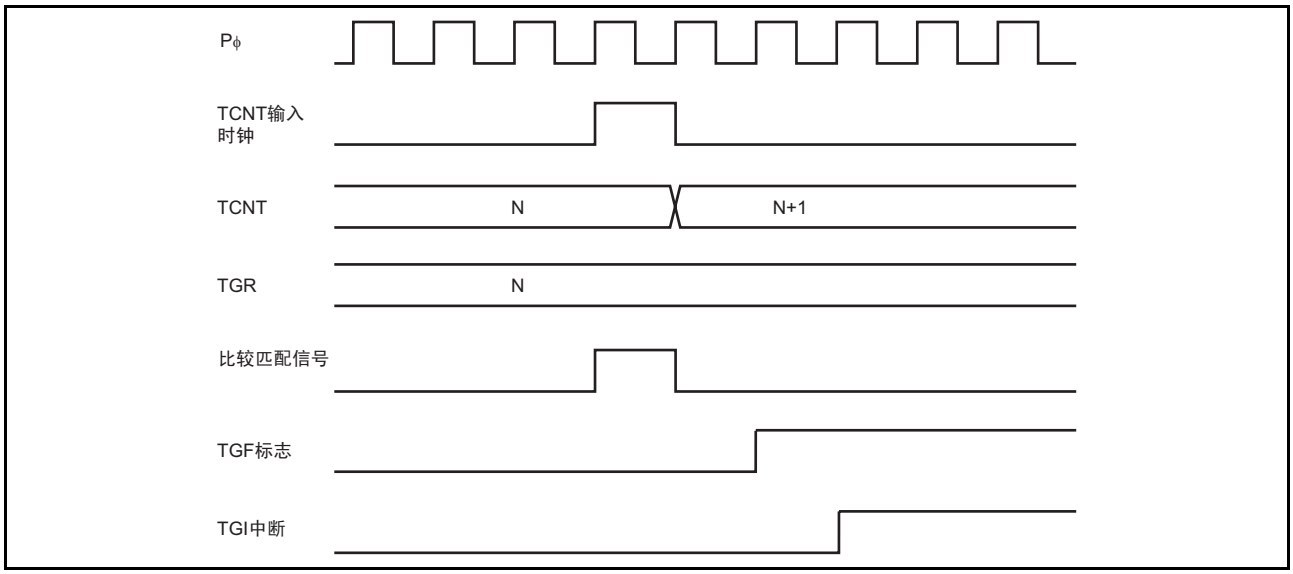


图 11.92　TGI 中断时序 （比较匹配）

（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 11.93 所示。

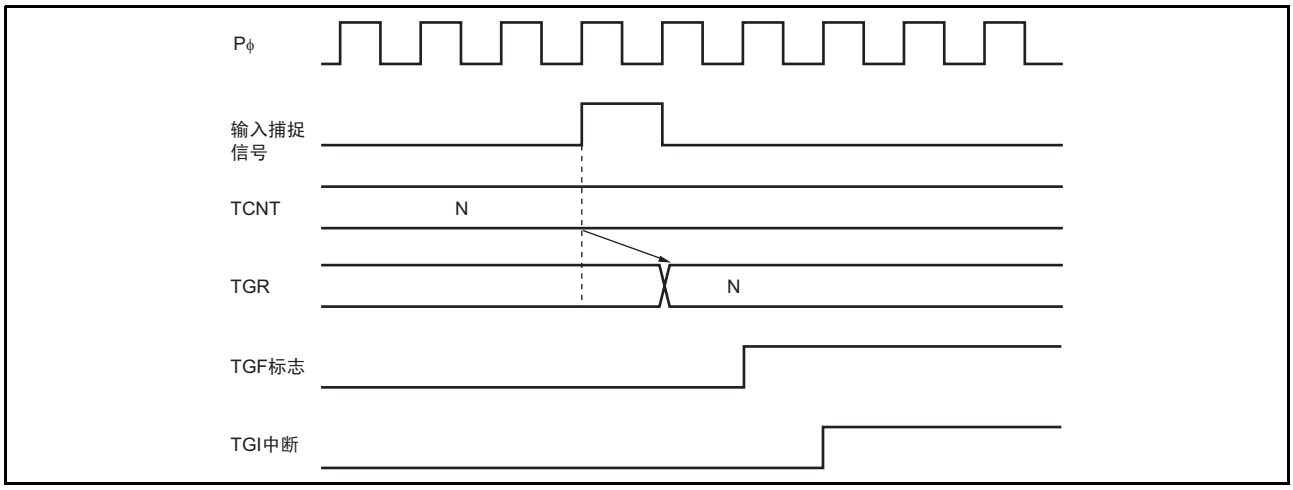


图 11.93　TGI 中断时序 （输入捕捉）

（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 11.94 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 11.95 所示。

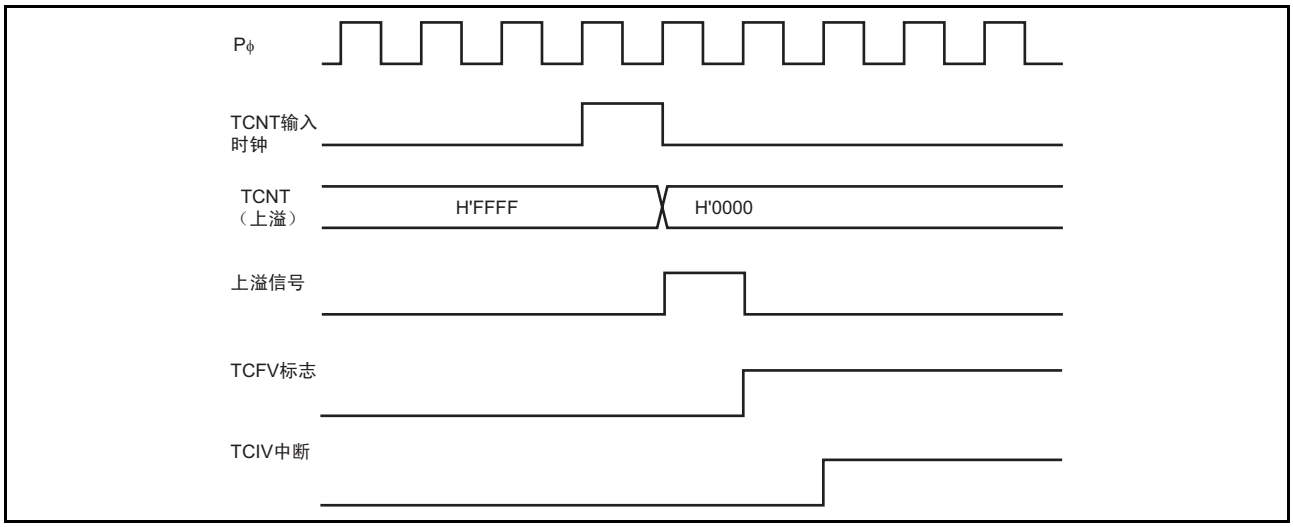


图 11.94　TCIV 中断的置位时序

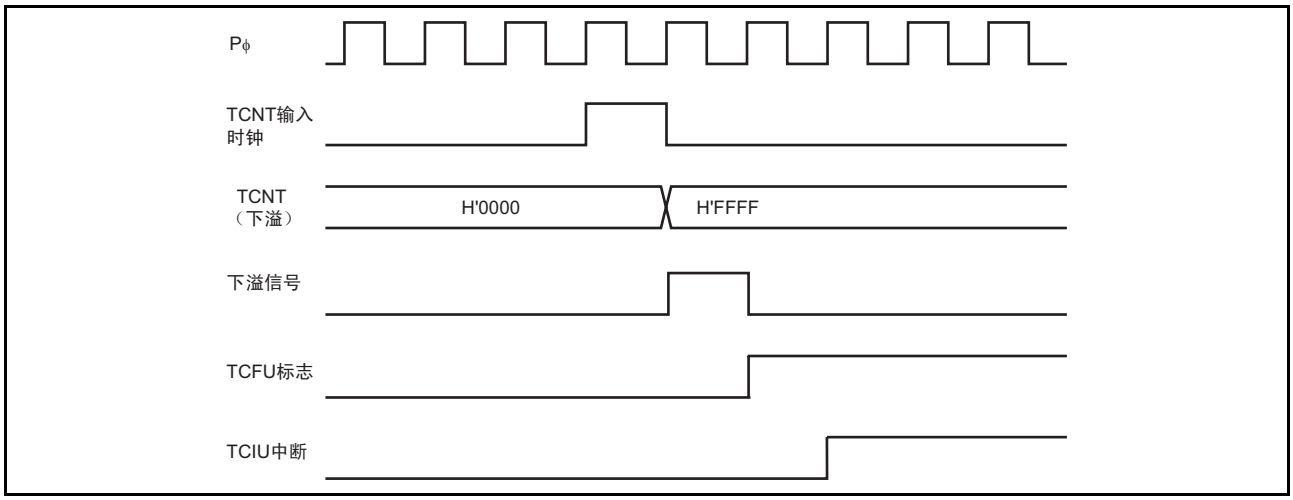


图 11.95　TCIU 中断的置位时序

（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 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 11.96 所示，通过 DMAC 清除状态标志的时序如图 11.97 所示。

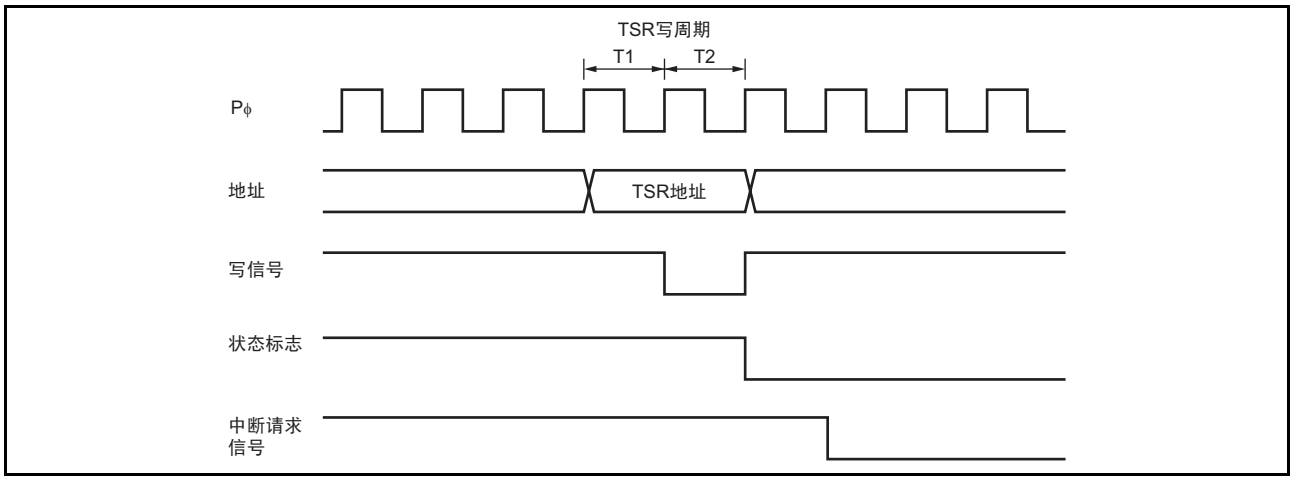


图 11.96　通过 CPU 清除状态标志的时序

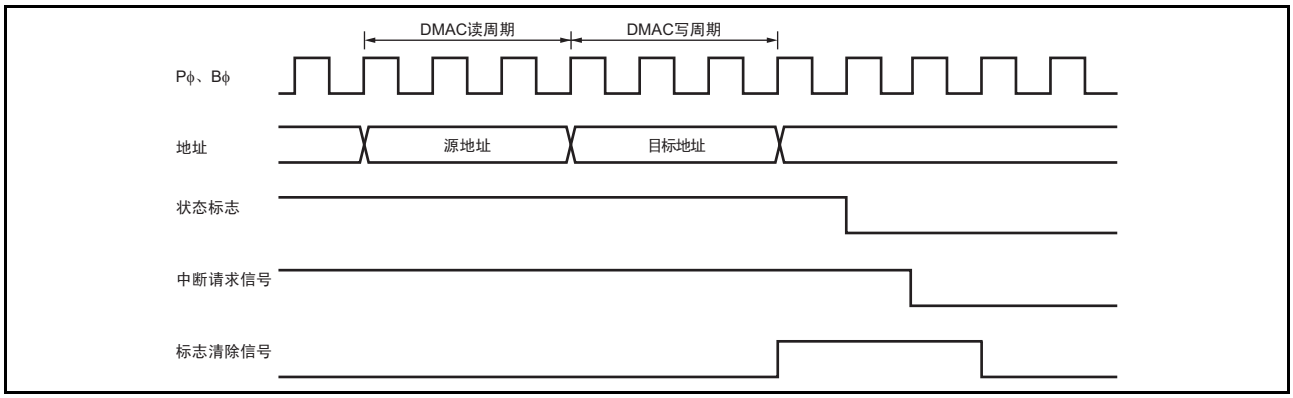


图 11.97　通过 DMAC 的启动清除状态标志的时序

11.7　使用时的注意事项

11.7.1　模块待机模式的设定

MTU2 能通过待机控制寄存器禁止 / 允许本模块的运行。初始值时，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 32 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，输入时钟的脉宽至少为 1.5 个状态。在双边沿的情况下，输入时钟的脉宽至少为 2.5 个状态。请注意，如果小于下述脉宽就不能正常运行。

为相位计数模式时，2 个输入时钟的相位差和重叠都至少为 1.5 个状态，脉宽至少为 2.5 个状态。相位计数模式的输入时钟条件如图 11.98 所示。

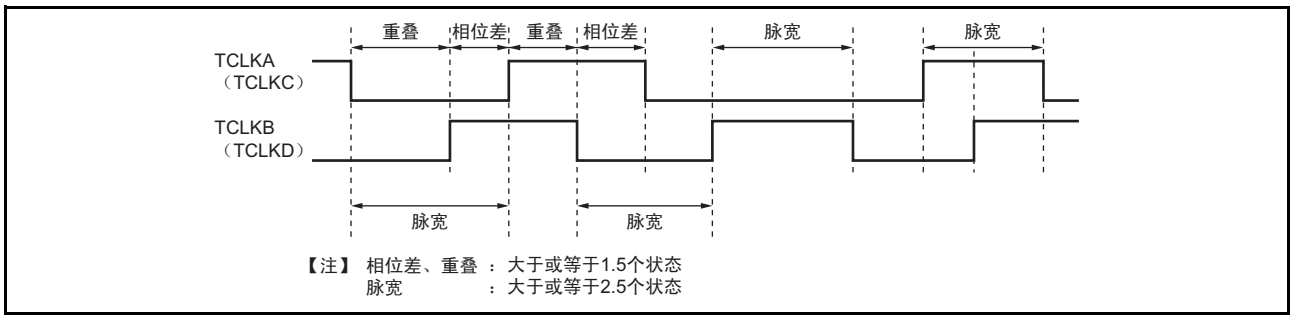


图 11.98　相位计数模式中的相位差、重叠和脉宽

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 的值匹配的最后状态（更新 TCNT 匹配后的计数值时序）清除 TCNT。因此，实际的计数器频率为以下的公式。

$$f = \frac{P\phi}{(N+1)}$$

f　：计数器频率

Pϕ ：外围时钟运行频率

N　：TGR 的设定值

11.7.4　TCNT 的写入和清除的竞争

如果在 TCNT 的写周期中的 T2 状态产生计数器清除信号，就不进行 TCNT 的写入而优先清除 TCNT。此时序如图 11.99 所示。

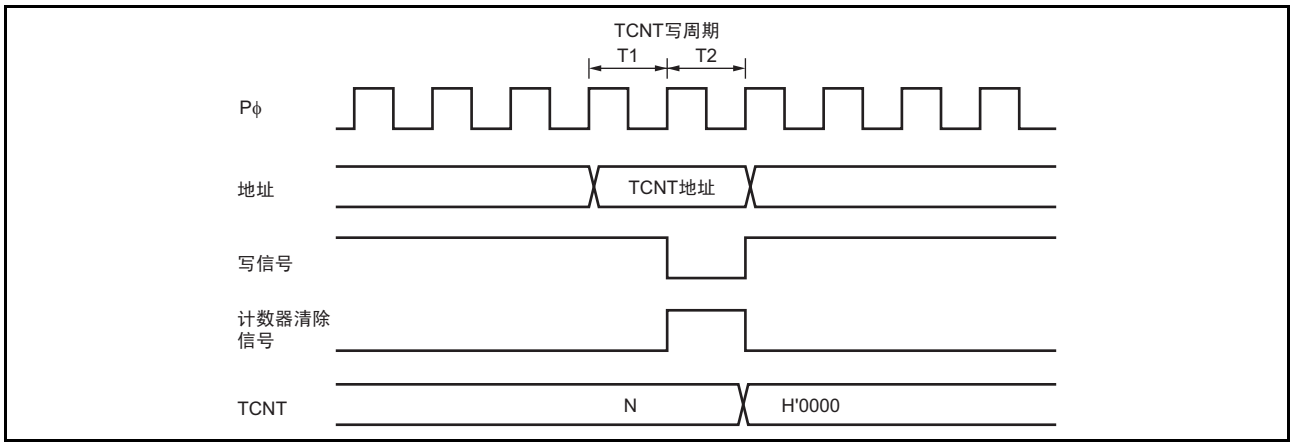


图 11.99　TCNT 的写入和清除的竞争

11.7.5　TCNT 的写入和递增计数的竞争

即使在 TCNT 写周期中的 T2 状态产生递增计数信号，也不进行递增计数而优先写入 TCNT。此时序如图 11.100 所示。

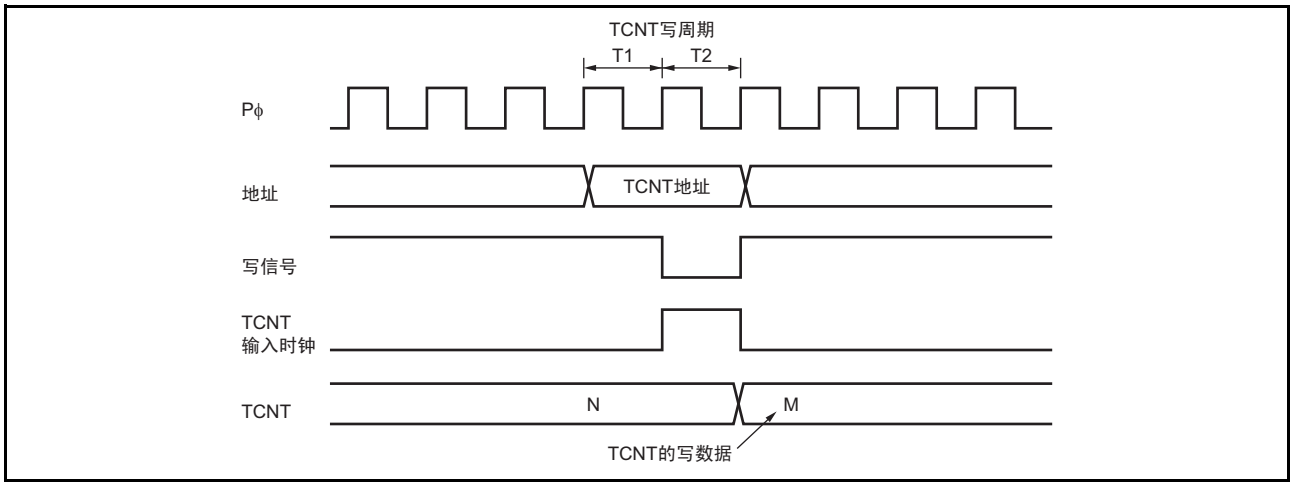


图 11.100　TCNT 的写入和递增计数的竞争

11.7.6　TGR 的写入和比较匹配的竞争

如果在 TGR 写周期中的 T2 状态产生比较匹配，就写入 TGR 并且产生比较匹配信号。此时序如图 11.101 所示。

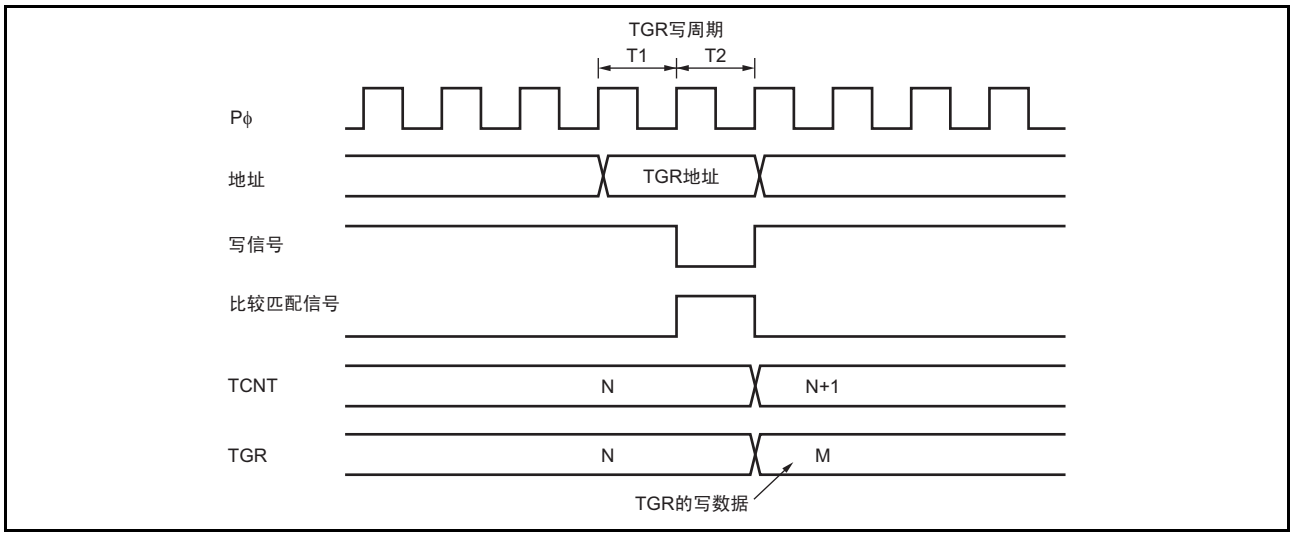


图 11.101　TGR 的写入和比较匹配的竞争

11.7.7　缓冲寄存器的写入和比较匹配的竞争

如果在 TGR 写周期中的 T2 状态产生比较匹配，由缓冲运行传送到 TGR 的数据就为写入前的数据。此时序如图 11.102 所示。

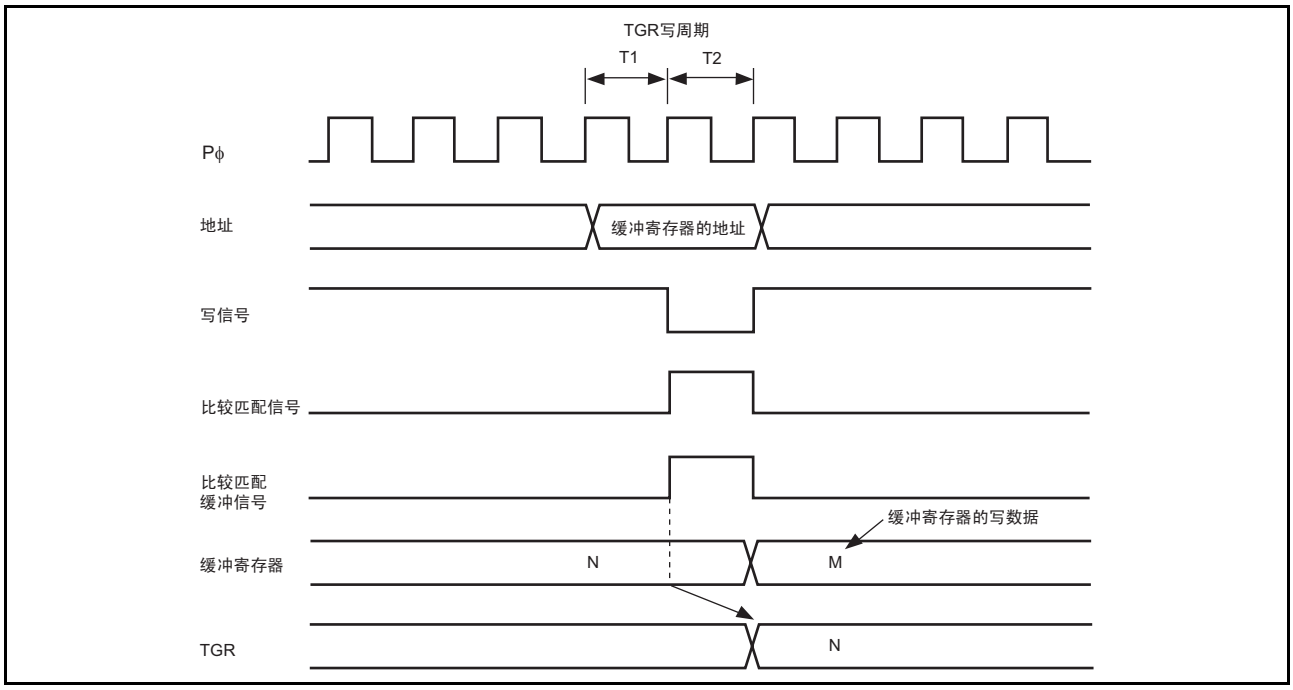


图 11.102　缓冲寄存器的写入和比较匹配的竞争

11.7.8　缓冲寄存器的写入和 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，由缓冲运行传送的数据就为写入前的数据。

此时序如图 11.103 所示。

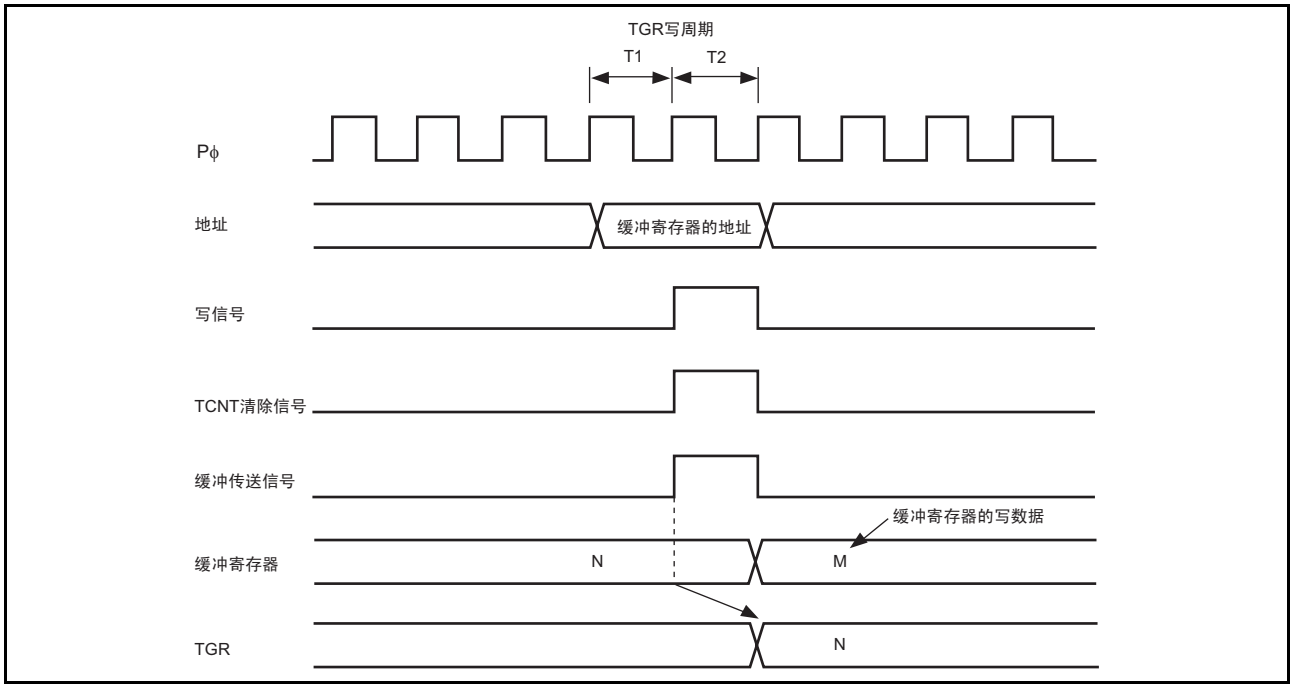


图 11.103　缓冲寄存器的写入和 TCNT 清除的竞争

11.7.9　TGR 的读取和输入捕捉的竞争

如果在 TGR 读周期中的 T1 状态产生输入捕捉信号，读出的数据为输入捕捉传送前的数据。此时序如图 11.104 所示。

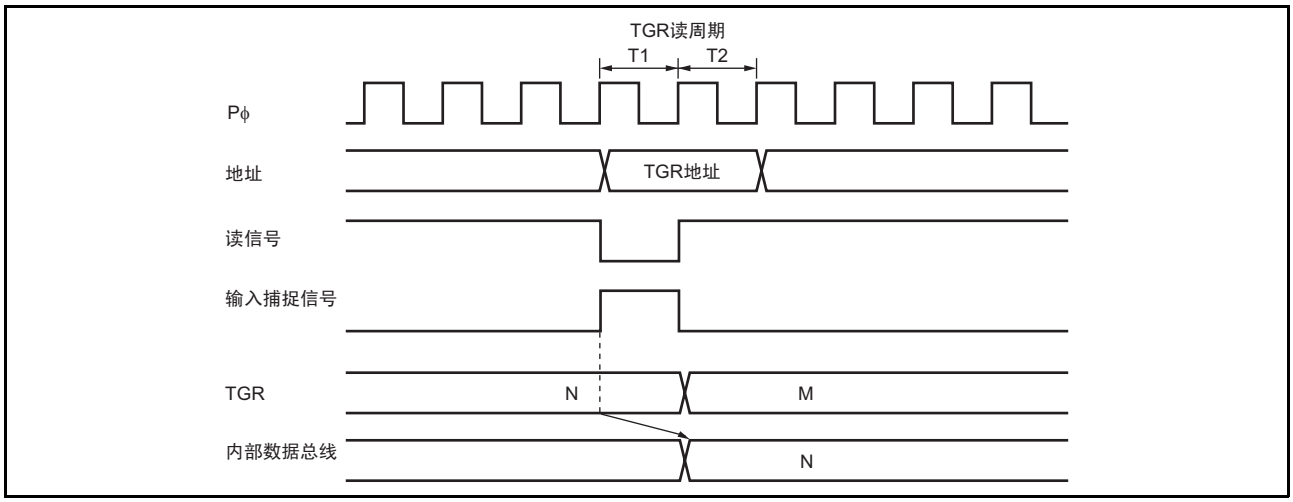


图 11.104　TGR 的读取和输入捕捉的竞争

11.7.10　TGR 的写入和输入捕捉的竞争

如果在 TGR 写周期中的 T2 状态产生输入捕捉信号，不写入 TGR 而优先输入捕捉。此时序如图 11.105 所示。

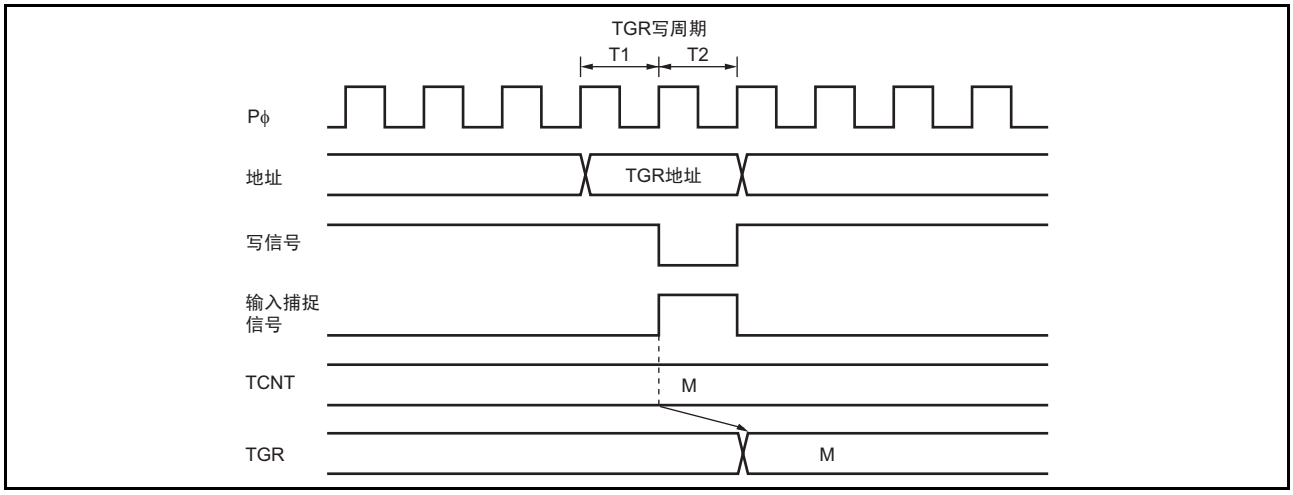


图 11.105　TGR 的写入和输入捕捉的竞争

11.7.11　缓冲寄存器的写入和输入捕捉的竞争

如果在缓冲器写周期中的 T2 状态产生输入捕捉信号，就不写入缓冲寄存器而优先进行缓冲运行。此时序如图 11.106 所示。

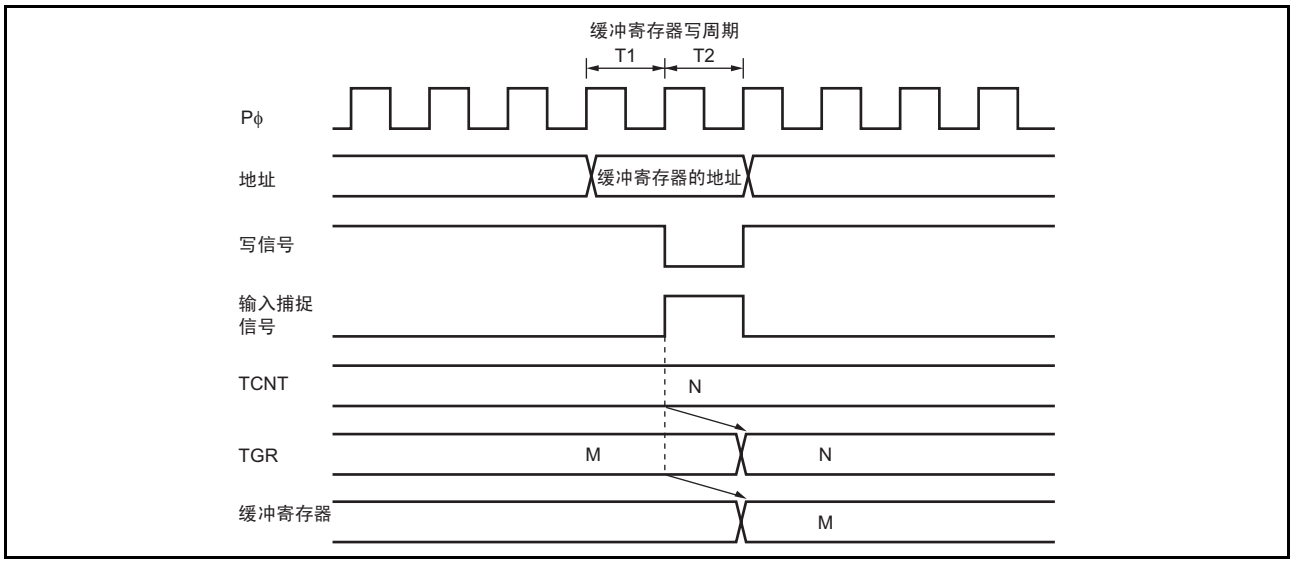


图 11.106　缓冲寄存器的写入和输入捕捉的竞争

11.7.12　级联中 TCNT_2 的写入和上溢 / 下溢的竞争

将定时器计数器 （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，就写入 TCNT_2，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 的值匹配时，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 11.107 所示。

另外，在级联运行中设定 TCNT 的清除时，请进行通道 1 和通道 2 的同步设定。

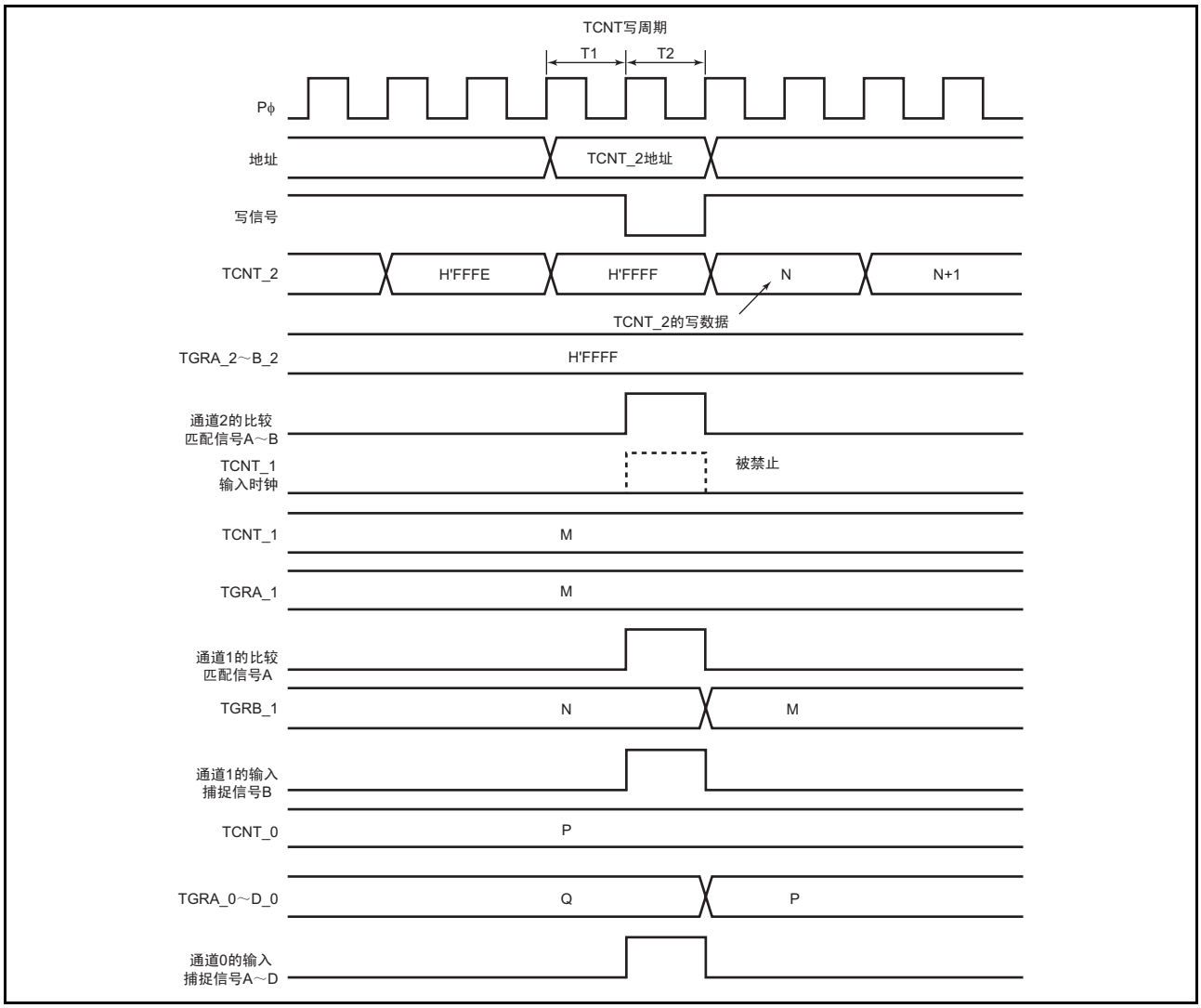


图 11.107　级联中 TCNT_2 的写入和上溢 / 下溢的竞争

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 11.108 所示。

另外，在其他运行模式中开始计数时，请给 TCNT_3 和 TCNT_4 设定计数的初始值。

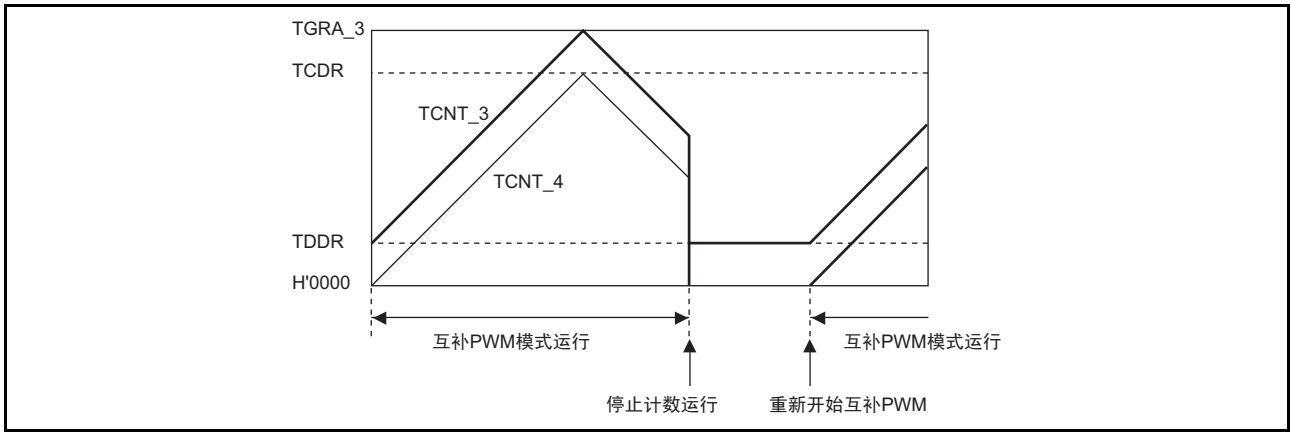


图 11.108　互补 PWM 模式停止时的计数器值

11.7.14　互补 PWM 模式中缓冲运行的设定

在互补 PWM 模式中，请在缓冲运行中改写 PWM 周期设定寄存器 （TGRA_3）、定时器周期数据寄存器（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 位置 1，TGRC_3 就用作 TGRA_3 的缓冲寄存器。同时 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

在复位同步 PWM 模式中设定缓冲运行时，请将 TMDR_4 的 BFA 位和 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 位和 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 位置 1， TGRC_3 就用作 TGRA_3 的缓冲寄存器。同时 TGRC_4 用作 TGRA_4 的缓冲寄存器。

当 TGRC_3、 TGRD_3 用作缓冲寄存器时， TSR_3、 TSR_4 的 TGFC 位和 TGFD 位就不被置位。

在将 TMDR_3 的 BFA 位和 BFB 位置 1，并且 TMDR_4 的 BFA 位和 BFB 位置 0 时， TGR_3、 TGR_4、 TIOC3、 TIOC4 的运行例子如图 11.109 所示。

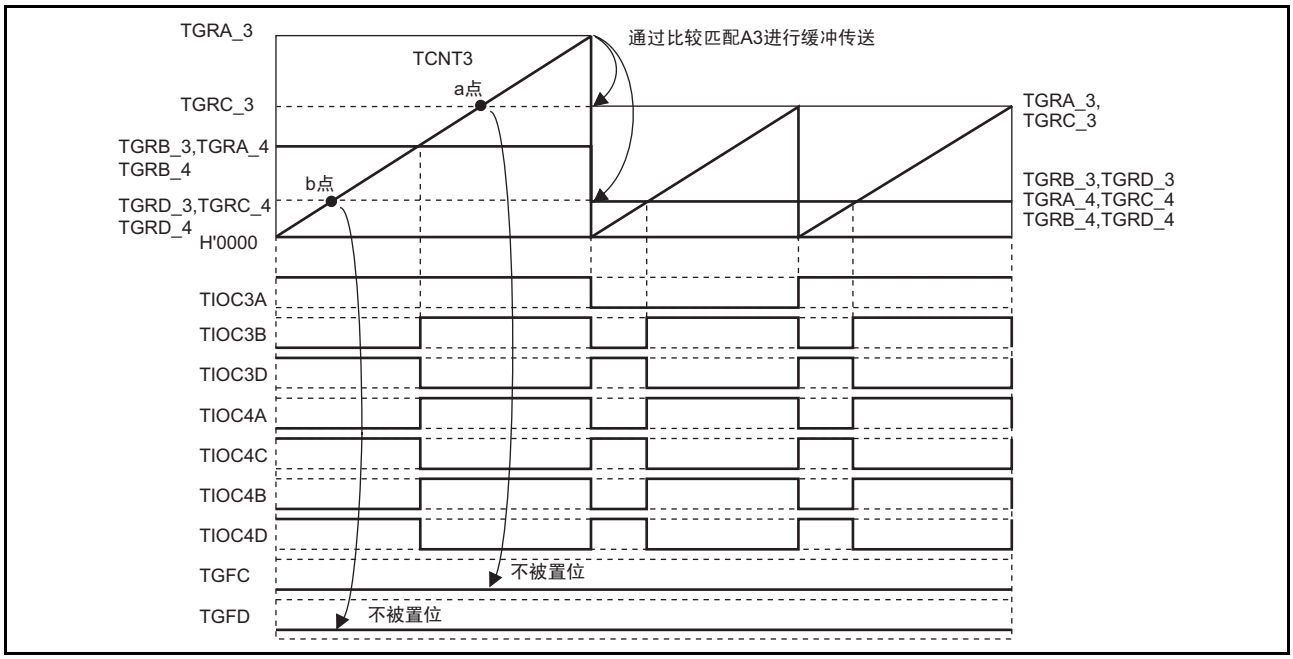


图 11.109　复位同步 PWM 模式的缓冲运行和比较匹配标志

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 11.110 所示。

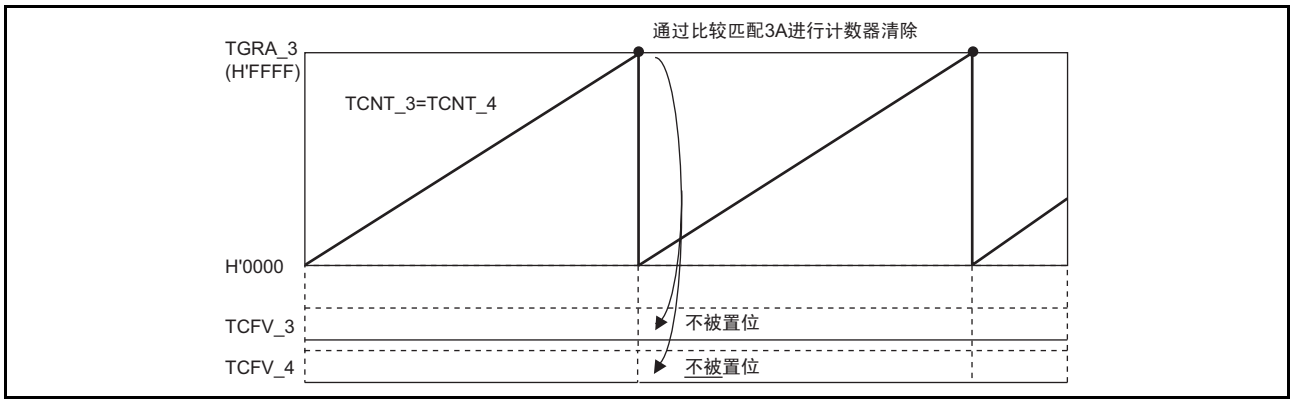


图 11.110　复位同步 PWM 模式的上溢标志

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 11.111 所示。

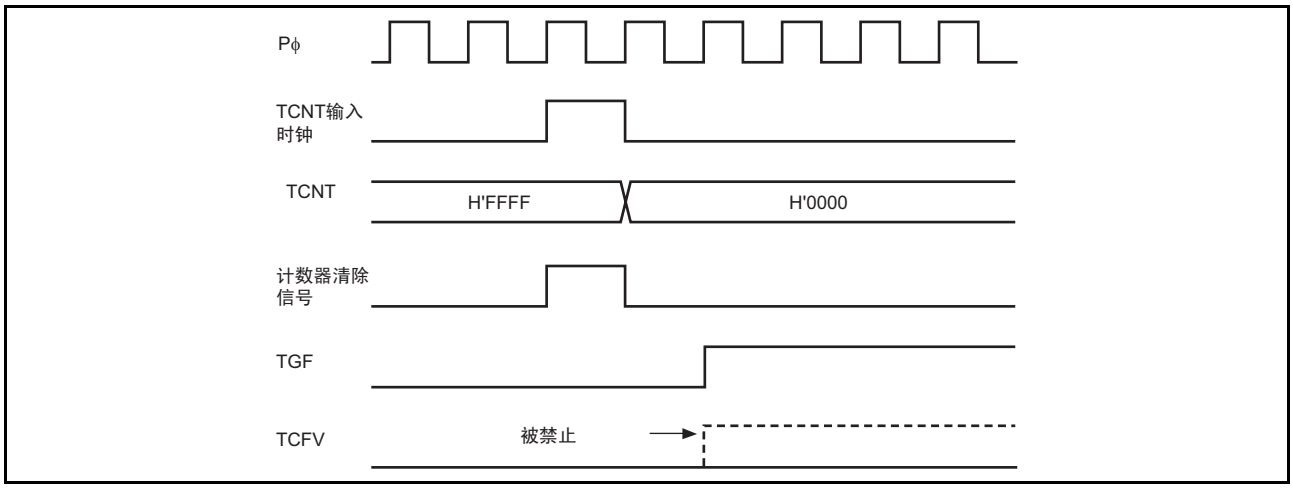


图 11.111　上溢和计数器清除的竞争

11.7.18　TCNT 的写入和上溢 / 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，也优先写入 TCNT 而不将 TSR 的 TCFV/TCFU 标志置位。

TCNT 的写入和上溢发生竞争时的运行时序如图 11.112 所示。

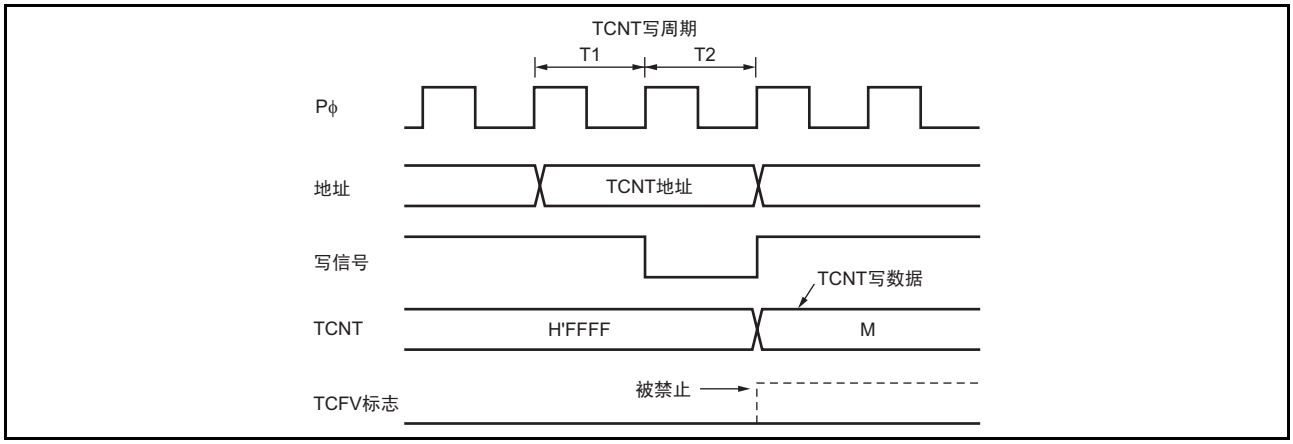


图 11.112　TCNT 的写入和上溢的竞争

11.7.19 从正常运行或者 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 的正常运行或者 PWM 模式 1 转移到复位同步 PWM 模式时，在将输出引脚（TIOC3B、TIOC3D、TIOC4A、 TIOC4C、 TIOC4B、 TIOC4D）置为高电平的状态下停止计数器。请注意，如果在转移到复位同步 PWM 模式后运行，就不能正确地进行引脚的初始输出。

从正常运行转移到复位同步 PWM 模式时，请将 H'11 写到 TIORH_3、 TIORL_3、 TIORH_4、 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。

从 PWM 模式 1 转移到复位同步 PWM 模式时，请暂且转移为正常运行，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，通过定时器输出控制寄存器（TOCR）的 OLSP 位和 OLSN 位设定 PWM 波形的输出电平。在互补 PWM 模式或者复位同步 PWM 模式时，请将 TIOR 置 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 DMAC 的启动源。

请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输入内部时，有可能因 TIOC1A、TIOC2A 或者 TIOC1B、TIOC2B 的输入时序产生偏差，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，但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，但会误传送 TCNT_1=H'FFF0 和 TCNT_2=H'0000 的值。

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。

- 普通模式 （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时，因为系统马上就运行，所以请在结束 MTU2 的输出引脚的初始设定后进行 PFC 的设定。

【注】 * 通道编号 + 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，就请通过系统截止 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。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 11.57** 所示。

表中使用下述的符号表示。

Normal：普通模式　　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 11.57　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）	none	none
PCM	（17）	（18）	（19）	（20）	none	none
CPWM	（21）	（22）	none	none	（23）（24）	（25）
RPWM	（26）	（27）	none	none	（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

- 在通过定时器的I/O控制寄存器（TIOR）的设定转移到选择引脚输出电平的模式（Normal、PWM1、PWM2、PCM）时，请通过设定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，请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，请在普通模式中进行初始化后转移到PWM模式2。
- 在普通模式或者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，请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，请在解除缓冲模式进行初始化后重新设定缓冲模式。
- 在通过定时器输出控制寄存器（TOCR）的设定转移到选择引脚输出电平的模式（CPWM、RPWM）时，请转移到普通模式并且通过TIOR进行初始化，在将TIOR返回到初始值后，通过定时器输出主控器允许寄存器（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、TOER的设定）运行。

【注】 本项记述中的 * 为通道号。

按照组合 No. 初始化引脚的步骤如**表 11.57** 所示。另外，有效电平为低电平。

（1） 在普通模式的运行中发生异常并在普通模式中重新开始时的运行

在普通模式中发生异常，重新设定后在普通模式中重新开始时的说明如**图 11.113** 所示。

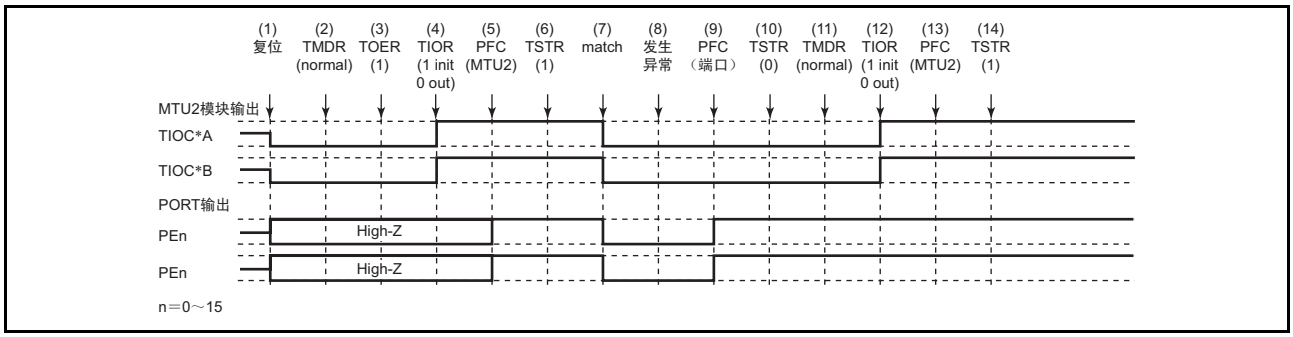


图 11.113　在普通模式中发生异常并在普通模式中恢复时

（1）通过复位， MTU2 输出为低电平，端口为高阻抗状态。
（2）通过复位， TMDR 为普通模式。
（3）在通道 3、 4 中通过 TIOR 初始化引脚前，请通过 TOER 允许输出。
（4）请通过 TIOR 初始化引脚 （例如，初始输出为高电平，比较匹配为低电平输出）。
（5）请通过 PFC 设定为 MTU2 输出。
（6）通过 TSTR 开始计数运行。
（7）通过产生比较匹配，输出低电平。
（8）发生了异常。
（9）请通过 PFC 设定为端口输出，输出有效电平的反相信号。
（10）通过 TSTR 停止计数运行。
（11）在普通模式中重新开始时不需要此步骤。
（12）请通过 TIOR 初始化引脚。
（13）请通过 PFC 设定为 MTU2 输出。
（14）通过 TSTR 重新开始。

（2） 在普通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

在普通模式中发生异常，重新设定后在 PWM 模式 1 中重新开始时的说明如图 11.114 所示。

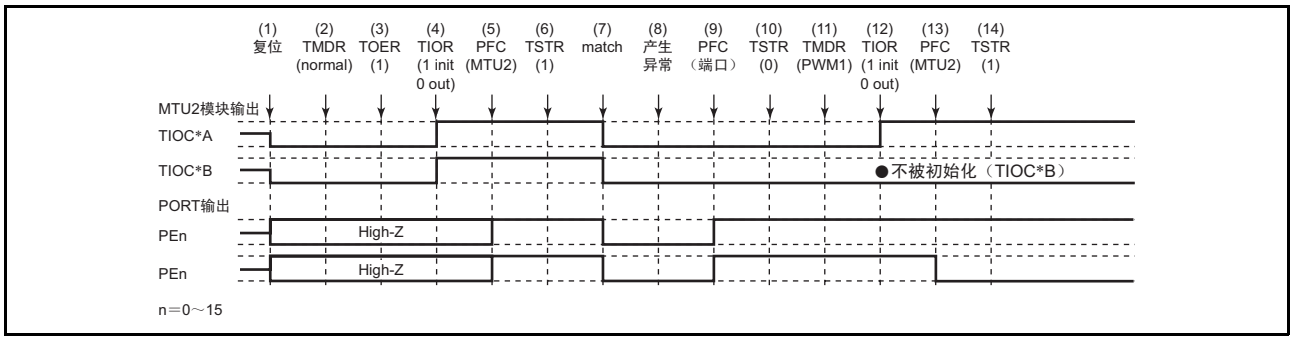


图 11.114　在普通模式中发生异常并在 PWM 模式 1 中恢复时

（1）～（10）和图 11.113 通用。
（11）设定 PWM 模式 1。
（12）请通过 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，请在普通模式中进行初始化后转移到 PWM 模式 1）。
（13）请通过 PFC 设定为 MTU2 输出。
（14）通过 TSTR 重新开始。

（3） 在普通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

在普通模式中发生异常，重新设定后在 PWM 模式 2 中重新开始时的说明如图 11.115 所示。

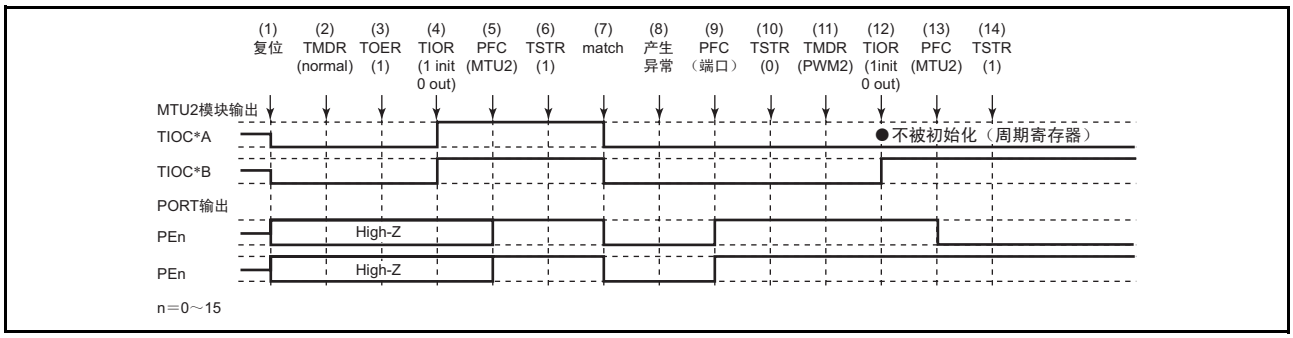


图 11.115　在普通模式中发生异常并在 PWM 模式 2 中恢复时

（1）～（10）和图 11.113 通用。
（11）设定 PWM 模式 2。
（12）请通过 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，请在普通模式中进行初始化后转移到 PWM 模式 2）。
（13）通过 PFC 设定为 MTU2 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） 在普通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

在普通模式中发生异常，重新设定后在相位计数模式中重新开始时的说明如图 11.116 所示。

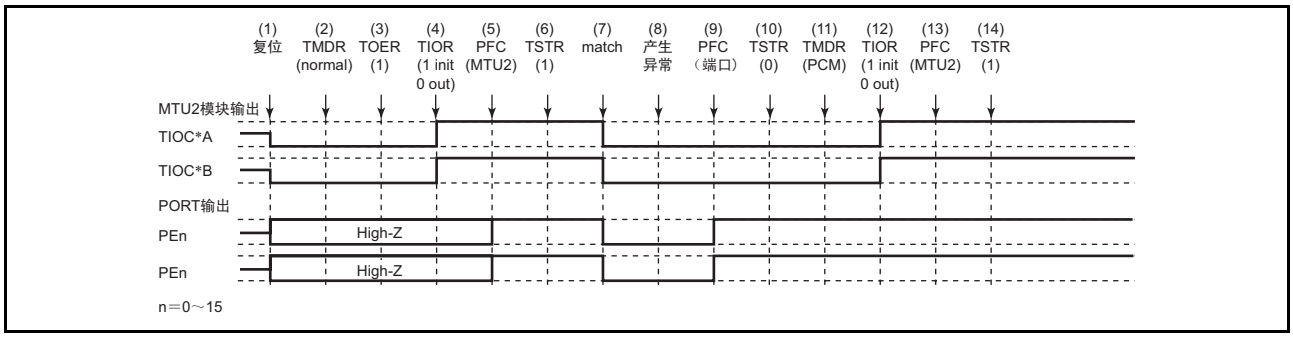


图 11.116　在普通模式中发生异常并在相位计数模式中恢复时

（1）～（10）和图 11.113 通用。
（11）设定相位计数模式。
（12）请通过 TIOR 初始化引脚。
（13）请通过 PFC 设定为 MTU2 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） 在普通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

在普通模式中发生异常，重新设定后在互补 PWM 模式中重新开始时的说明如图 11.117 所示。

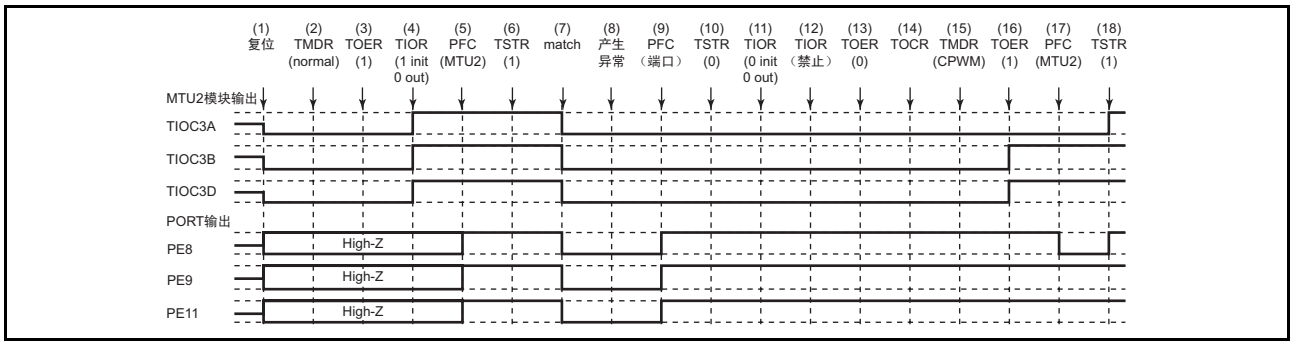


图 11.117　在普通模式中发生异常并在互补 PWM 模式中恢复时

（1）～（10）和图 11.113 通用。
（11）请通过 TIOR 初始化普通模式的波形生成部。
（12）请通过 TIOR 禁止普通模式的波形生成部的运行。
（13）请通过 TOER 禁止通道 3、4 的输出。
（14）请通过 TOCR 选择互补 PWM 的输出电平并允许 / 禁止周期输出。
（15）设定互补 PWM。
（16）请通过 TOER 允许通道 3、4 的输出。
（17）请通过 PFC 设定为 MTU2 输出。
（18）通过 TSTR 重新开始。

（6） 在普通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

在普通模式中发生异常，重新设定后在复位同步 PWM 模式中重新开始时的说明如图 11.118 所示。

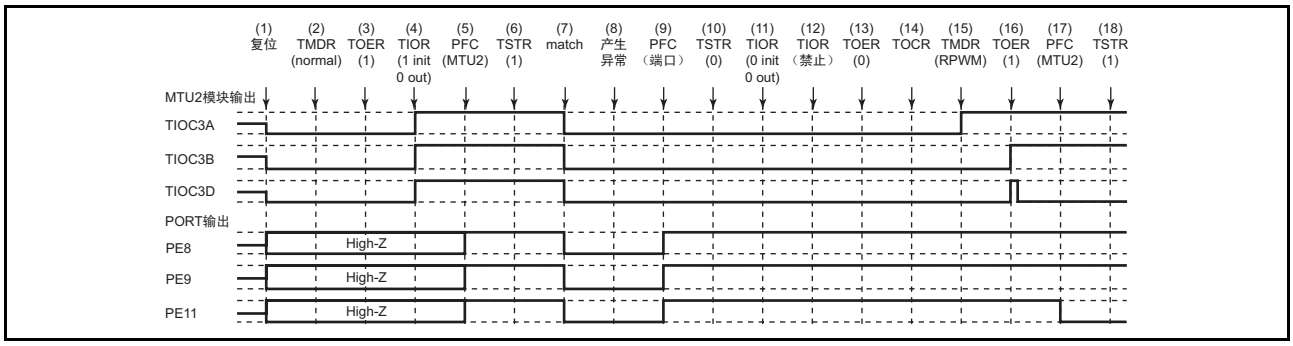


图 11.118　在普通模式中发生异常并在复位同步 PWM 模式中恢复时

（1）～（13）和图 11.113 通用。
（14）请通过 TOCR 选择复位同步 PWM 的输出电平和允许 / 禁止周期输出。
（15）设定复位同步 PWM。
（16）请通过 TOER 允许通道 3、 4 的输出。
（17）请通过 PFC 设定为 MTU2 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普通模式中重新开始时的运行

在 PWM 模式 1 中发生异常，重新设定后在普通模式中重新开始时的说明如图 11.119 所示。

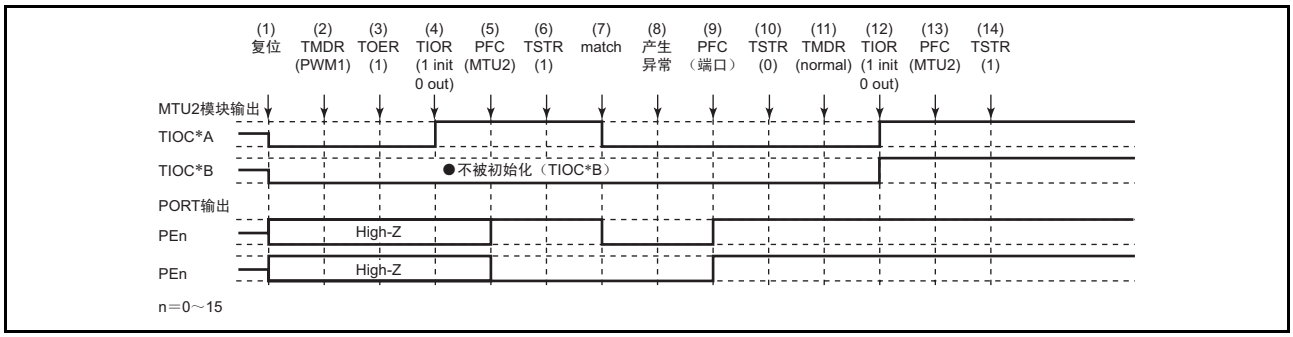


图 11.119　在 PWM 模式 1 中发生异常并在普通模式中恢复时

（1）通过复位， MTU2 输出为低电平，端口为高阻抗状态。
（2）请设定 PWM 模式 1。
（3）在通道 3、 4 中通过 TIOR 初始化引脚前，请通过 TOER 允许输出。
（4）请通过 TIOR 初始化引脚 （例如，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 输出。
（6）通过 TSTR 开始计数运行。
（7）通过产生比较匹配，输出低电平。
（8）发生了异常。
（9）请通过 PFC 设定为端口输出，输出有效电平的反相信号。
（10）通过 TSTR 停止计数运行。
（11）请设定普通模式。
（12）请通过 TIOR 初始化引脚。
（13）请通过 PFC 设定为 MTU2 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 11.120 所示。

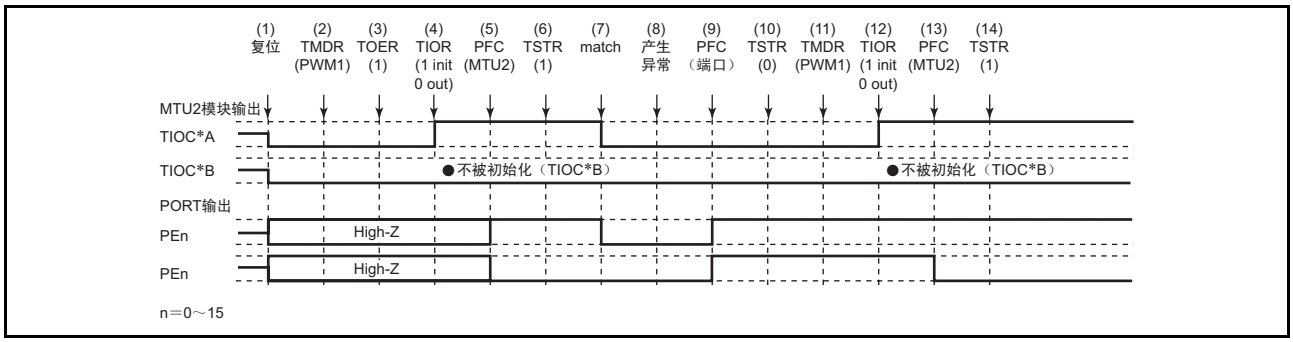


图 11.120　在 PWM 模式 1 中发生异常并在 PWM 模式 1 中恢复时

（1）～（10）和图 11.119 通用。
（11）在 PWM 模式 1 中重新开始时不需要此步骤。
（12）请通过 TIOR 初始化引脚 （在 PWM 模式 1 中不初始化 TIOC*B）。
（13）请通过 PFC 设定为 MTU2 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 中重新开始时的说明如图 11.121 所示。

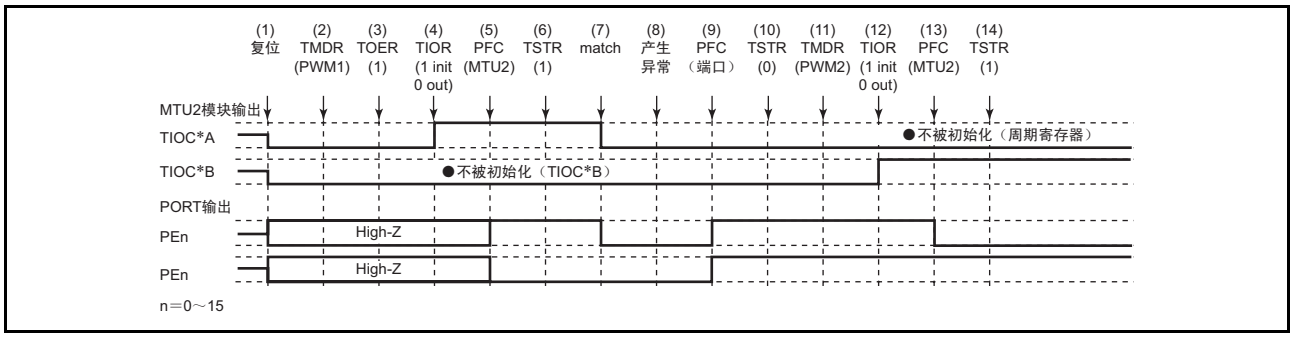


图 11.121　在 PWM 模式 1 中发生异常并在 PWM 模式 2 中恢复时

（1）～（10）和图 11.119 通用。
（11）设定 PWM 模式 2。
（12）请通过 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）请通过 PFC 设定为 MTU2 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 11.122 所示。

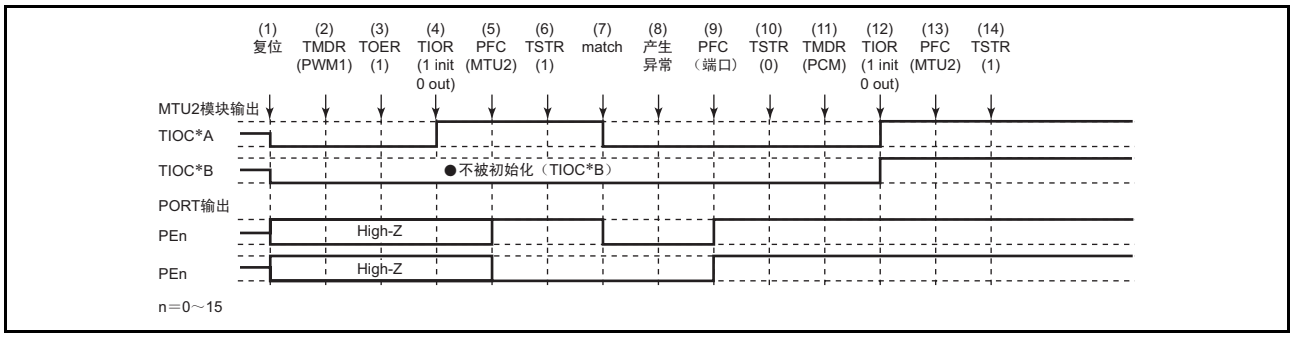


图 11.122　在 PWM 模式 1 中发生异常并在相位计数模式中恢复时

（1）～（10）和图 11.119 通用。
（11）设定相位计数模式。
（12）请通过 TIOR 初始化引脚。
（13）请通过 PFC 设定为 MTU2 输出。
（14）通过 TSTR 重新开始。

【注】 只能在通道 1、2 中设定位相计数模式，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 模式中重新开始时的说明如图 11.123 所示。

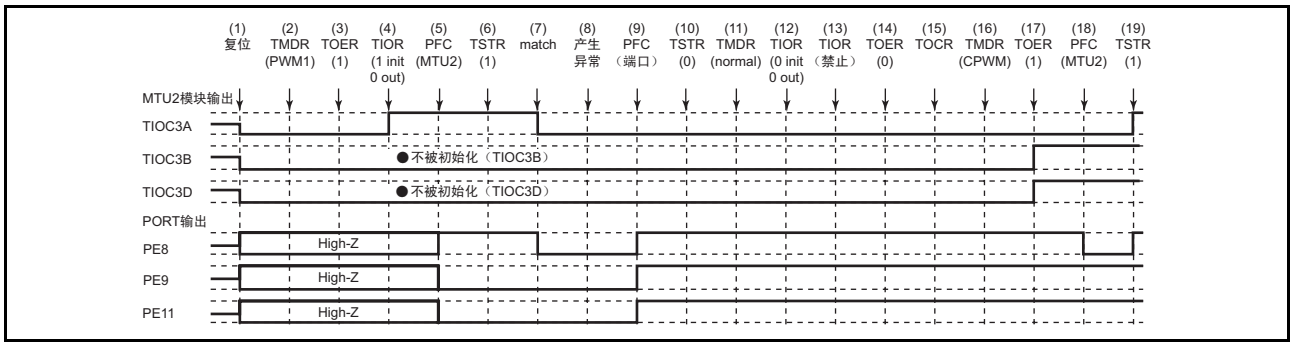


图 11.123　在 PWM 模式 1 中发生异常并在互补 PWM 模式中恢复时

（1）～（10）和图 11.119 通用。
（11）为了初始化波形生成部，请设定普通模式。
（12）请通过 TIOR 初始化 PWM 模式 1 的波形生成部。
（13）请通过 TIOR 禁止 PWM 模式 1 的波形生成部的运行
（14）请通过 TOER 禁止通道 3、 4 的输出。
（15）请通过 TOCR 选择 PWM 的输出电平和允许 / 禁止周期输出。
（16）设定互补 PWM。
（17）请通过 TOER 允许通道 3、 4 的输出。
（18）请通过 PFC 设定为 MTU2 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 模式中重新开始时的说明如图 11.124 所示。

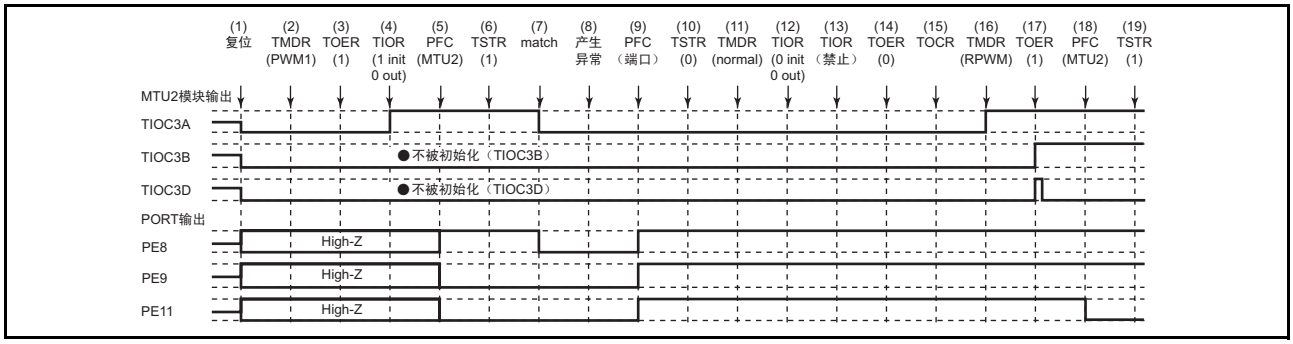


图 11.124　在 PWM 模式 1 中发生异常并在复位同步 PWM 模式中恢复时

（1）～（14）和图 11.123 通用。
（15）请通过 TOCR 选择复位同步 PWM 的输出电平并允许或者禁止周期输出。
（16）设定复位同步 PWM。
（17）请通过 TOER 允许通道 3、 4 的输出。
（18）通过 PFC 设定为 MTU2 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普通模式中重新开始时的运行

在 PWM 模式 2 中发生异常，重新设定后在普通模式中重新开始时的说明如图 11.125 所示。

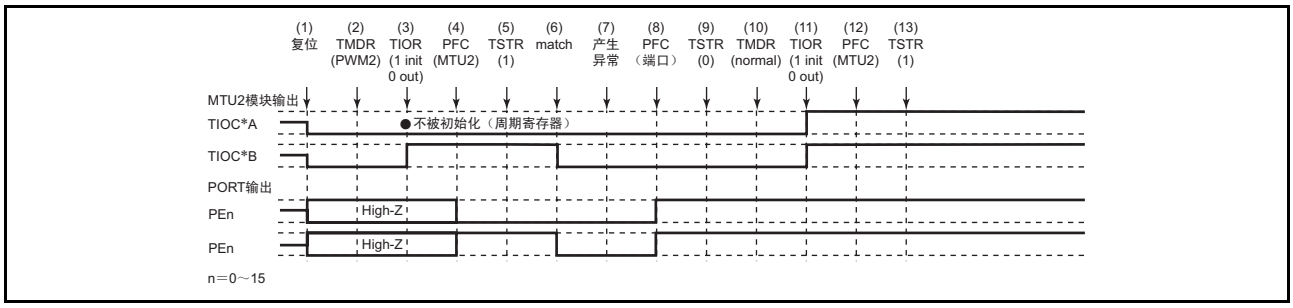


图 11.125　在 PWM 模式 2 中发生异常并在普通模式中恢复时

（1）通过复位， MTU2 输出为低电平，端口为高阻抗状态。
（2）请设定 PWM 模式 2。
（3）请通过 TIOR 初始化引脚 （例如，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。例如， TIOC*A 为周期寄存器）。
（4）请通过 PFC 设定为 MTU2 输出。
（5）通过 TSTR 开始计数运行。
（6）通过产生比较匹配，输出低电平。
（7）发生了异常。
（8）请通过 PFC 设定端口输出，输出有效电平的反相信号。
（9）通过 TSTR 停止计数运行。
（10）请设定普通模式。
（11）请通过 TIOR 初始化引脚。
（12）请通过 PFC 设定为 MTU2 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 中重新开始时的说明如图 11.126 所示。

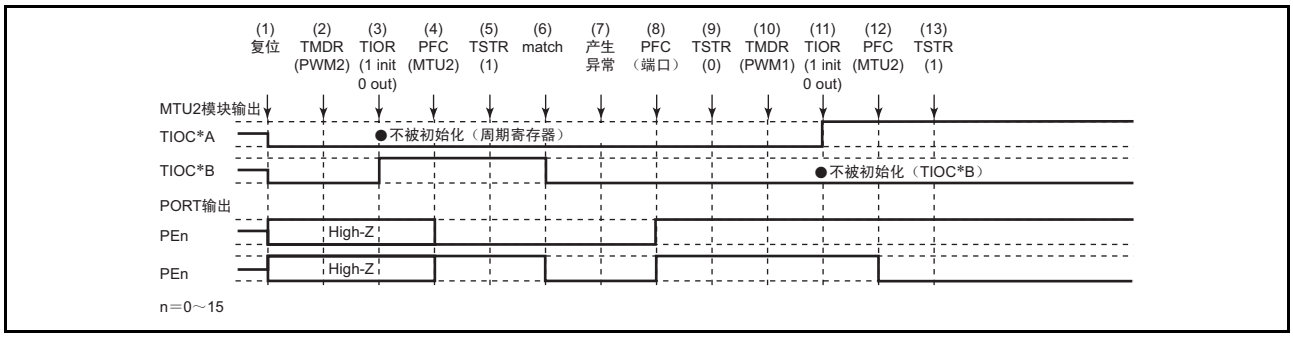


图 11.126　在 PWM 模式 2 中发生异常并在 PWM 模式 1 中恢复时

（1）～（9）和图 11.125 通用。
（10）设定 PWM 模式 1。
（11）请通过 TIOR 初始化引脚 （在 PWM 模式 1 中不初始化 TIOC*B）。
（12）请通过 PFC 设定为 MTU2 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 11.127 所示。

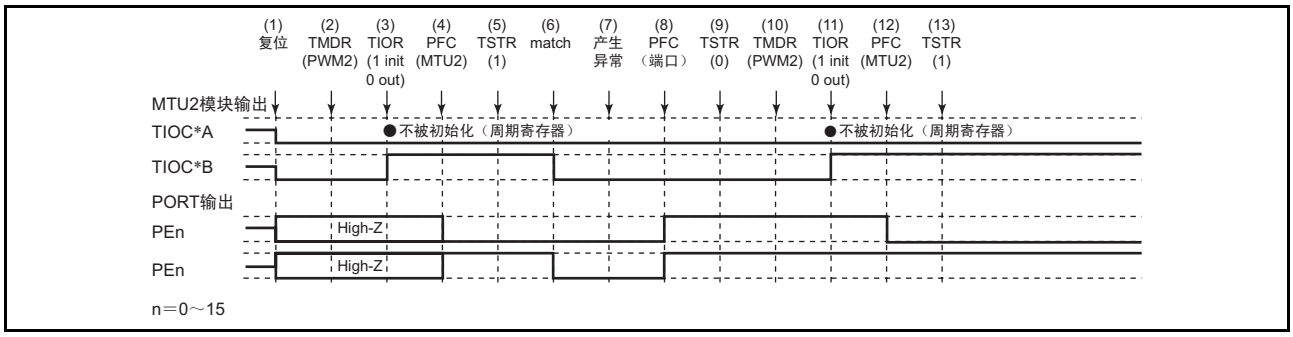


图 11.127　PWM 模式 2 中发生异常并在 PWM 模式 2 中恢复时

（1）～（9）和图 11.125 通用。
（10）在 PWM 模式 2 中重新开始时不需要此步骤。
（11）请通过 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）请通过 PFC 设定为 MTU2 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，重新设定后在相位计数模式中重新开始时的说明如图 11.128 所示。

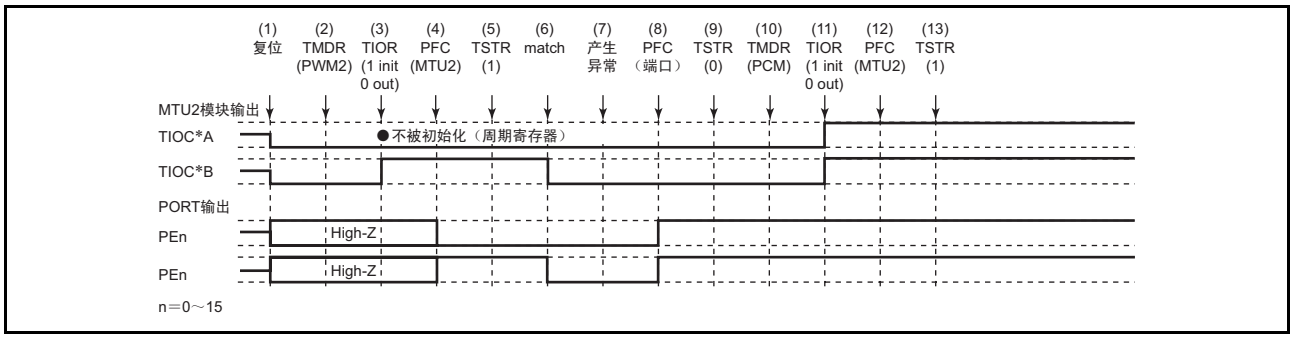


图 11.128　在 PWM 模式 2 中发生异常并在相位计数模式中恢复时

（1）～（9）和图 11.125 通用。
（10）设定相位计数模式。
（11）请通过 TIOR 初始化引脚。
（12）请通过 PFC 设定为 MTU2 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普通模式中重新开始时的运行

在相位计数模式中发生异常，重新设定后在普通模式中重新开始时的说明如图 11.129 所示。

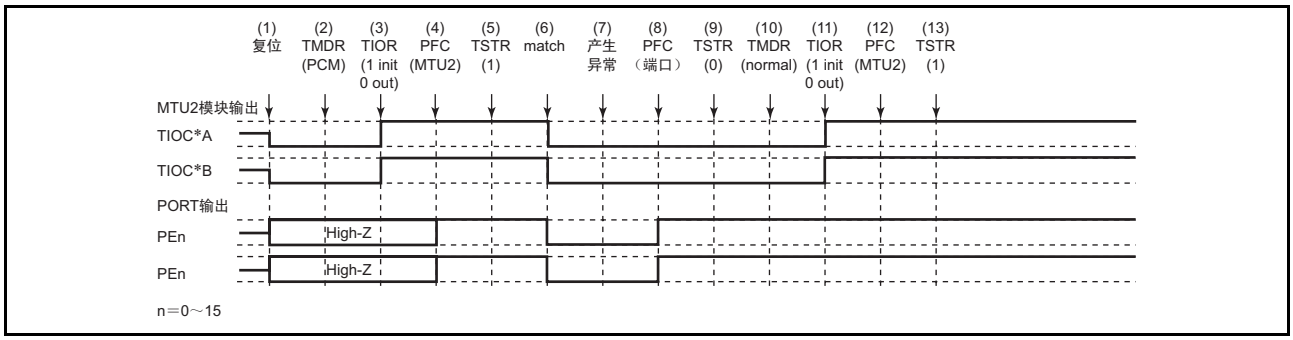


图 11.129　在相位计数模式中发生异常并在普通模式中恢复时

（1）通过复位， MTU2 输出为低电平，端口为高阻抗状态。
（2）请设定相位计数模式。
（3）请通过 TIOR 初始化引脚 （例如，初始输出为高电平，通过比较匹配输出低电平）。
（4）请通过 PFC 设定为 MTU2 输出。
（5）通过 TSTR 开始计数运行。
（6）通过产生比较匹配，输出低电平。
（7）发生了异常。
（8）通过 PFC 设定端口输出，输出有效电平的反相信号。
（9）通过 TSTR 停止计数运行。
（10）请在正常模式中进行设定。
（11）请通过 TIOR 初始化引脚。
（12）请通过 PFC 设定为 MTU2 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 11.130 所示。

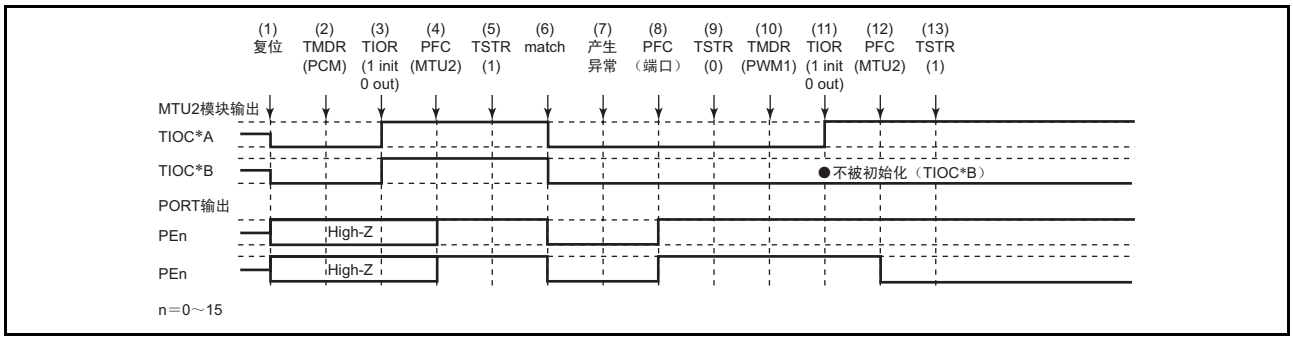


图 11.130　在相位计数模式中发生异常并在 PWM 模式 1 中恢复时

（1）～（9）和图 11.129 通用。
（10）设定 PWM 模式 1。
（11）请通过 TIOR 初始化引脚 （在 PWM 模式 1 中不初始化 TIOC*B）。
（12）请通过 PFC 设定为 MTU2 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 11.131 所示。

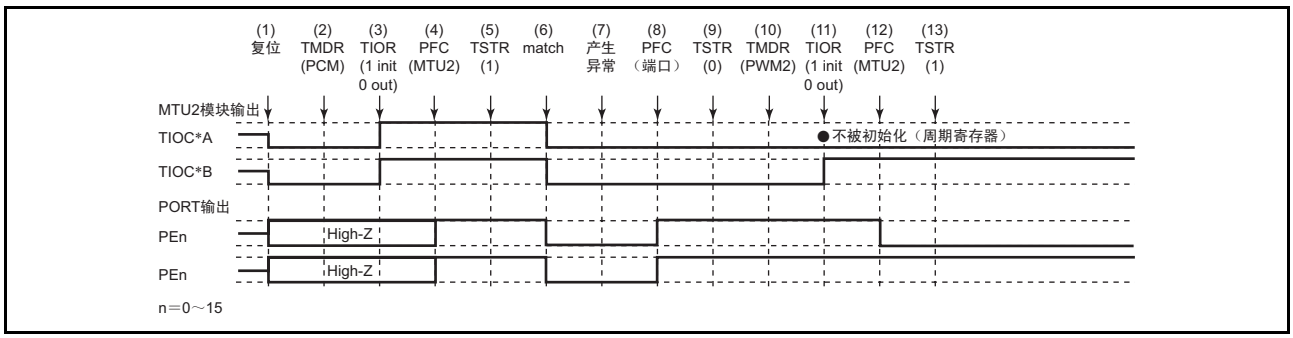


图 11.131　在相位计数模式中发生异常并在 PWM 模式 2 中恢复时

（1）～（9）和图 11.129 通用。
（10）设定 PWM 模式 2。
（11）请通过 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）请通过 PFC 设定为 MTU2 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 11.132 所示。

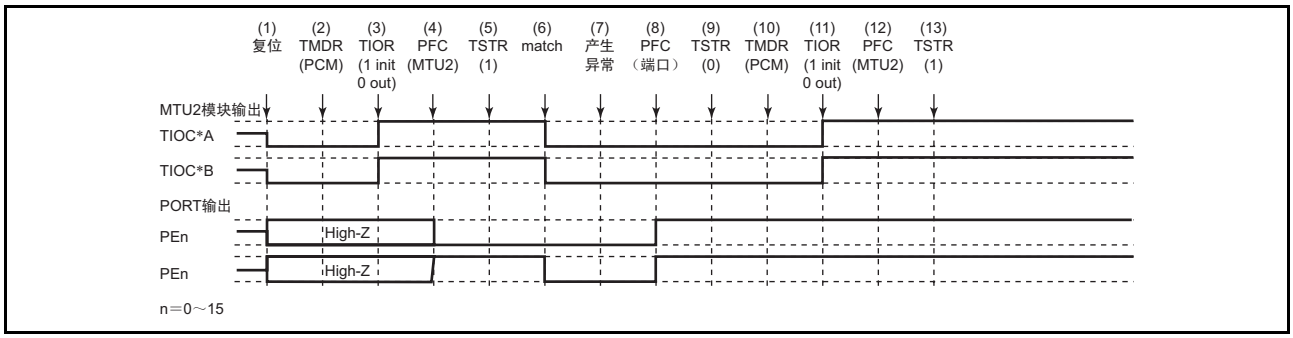


图 11.132　在相位计数模式中发生异常并在相位计数模式中恢复时

（1）～（9）和图 11.129 通用。
（10）在相位计数模式中重现开始时不需要此步骤。
（11）请通过 TIOR 初始化引脚。
（12）请通过 PFC 设定为 MTU2 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普通模式中重新开始时的运行

在互补 PWM 模式中发生异常，重新设定后在普通模式中重新开始时的说明如图 11.133 所示。

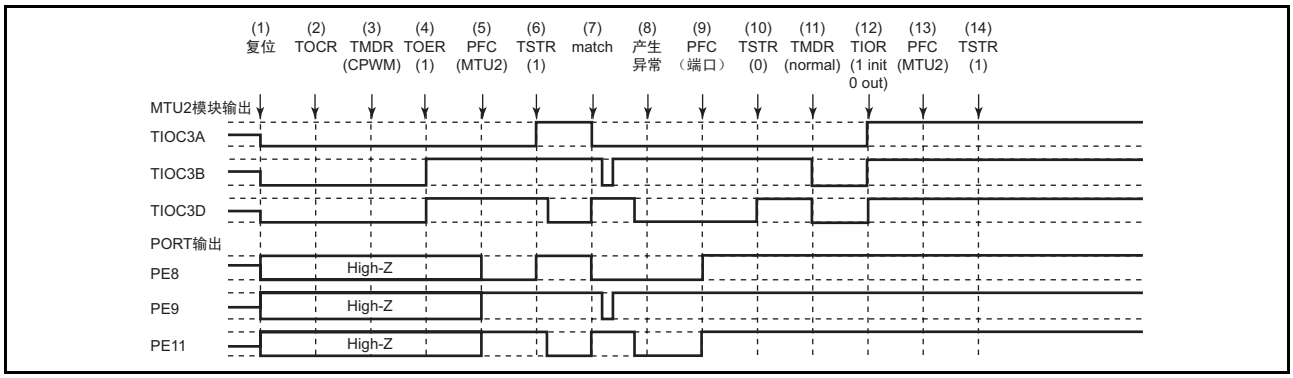


图 11.133　在互补 PWM 模式中发生异常并在普通模式中恢复时

（1）通过复位， MTU2 输出为低电平，端口为高阻抗状态。
（2）请通过 TOCR 选择互补 PWM 的输出电平并允许 / 禁止周期输出。
（3）设定互补 PWM。
（4）请通过 TOER 允许通道 3、 4 的输出。
（5）请通过 PFC 设定为 MTU2 输出。
（6）通过 TSTR 开始计数运行。
（7）通过产生比较匹配，输出互补 PWM 波形。
（8）发生了异常。
（9）请通过 PFC 设定为端口输出，输出有效电平的反相信号。
（10）通过 TSTR 停止计数运行 （MTU2 输出为互补 PWM 输出的初始值）。
（11）请设定普通模式 （MTU2 的输出为低电平）。
（12）请通过 TIOR 初始化引脚。
（13）请通过 PFC 设定为 MTU2 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 11.134 所示。

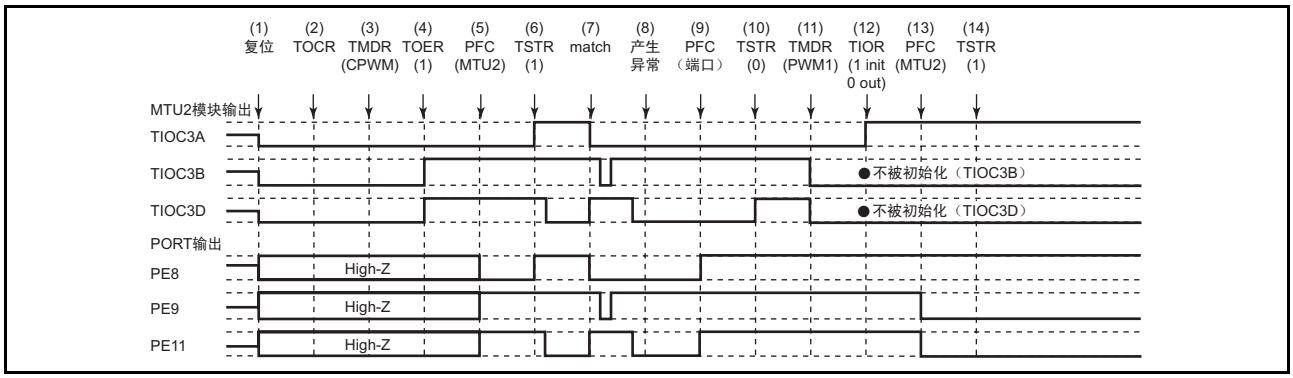


图 11.134　在互补 PWM 模式中发生异常并在 PWM 模式 1 中恢复时

（1）～（10）和图 11.133 通用。
（11）请设定 PWM 模式 1 （MTU2 输出为低电平）。
（12）请通过 TIOR 初始化引脚 （在 PWM 模式 1 中不初始化 TIOC*B）。
（13）请通过 PFC 设定为 MTU2 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（用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 11.135 所示。

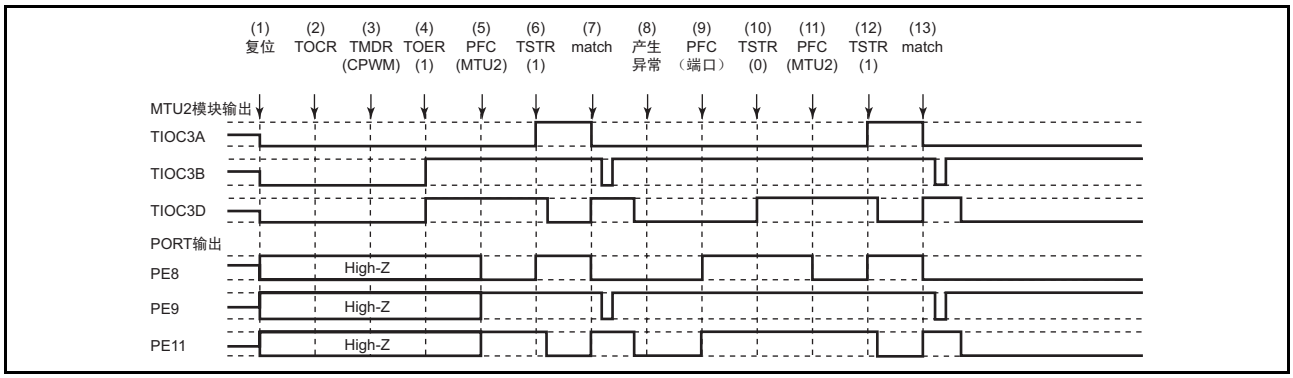


图 11.135　在互补 PWM 模式中发生异常并在互补 PWM 模式中恢复时

（1）～（10）和图 11.133 通用。
（11）请通过 PFC 设定为 MTU2 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 （用周期和占空比的新设定值重新开始）时的说明如图 11.136 所示。

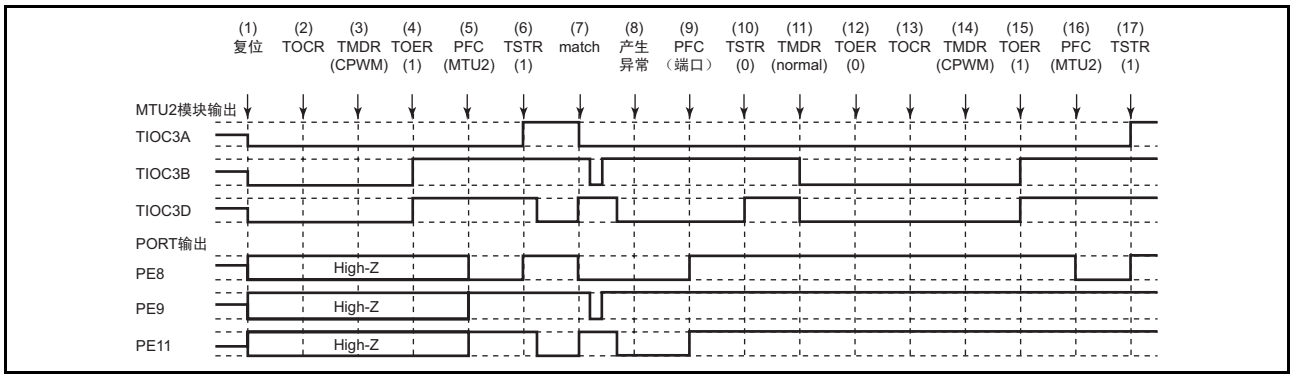


图 11.136　在互补 PWM 模式中发生异常并在互补 PWM 模式中恢复时

（1）～（10）和图 11.133 通用。
（11）请设定普通模式和新的设定值 （MTU2 输出为低电平）。
（12）请通过 TOER 禁止通道 3、 4 的输出。
（13）请通过 TOCR 选择互补 PWM 模式的输出电平并允许 / 禁止周期输出。
（14）设定互补 PWM。
（15）请通过 TOER 允许通道 3、 4 的输出。
（16）请通过 PFC 设定为 MTU2 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 11.137 所示。

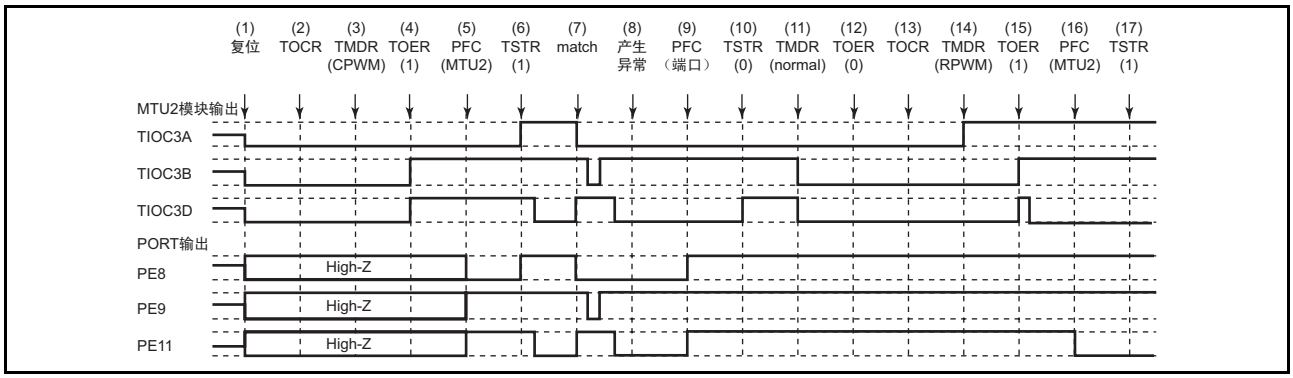


图 11.137　在互补 PWM 模式中发生异常并在复位同步 PWM 模式中恢复时

（1）～（10）和图 11.133 通用。
（11）请设定普通模式 （MTU2 输出为低电平）。
（12）请通过 TOER 禁止通道 3、 4 的输出。
（13）请通过 TOCR 选择复位同步 PWM 模式的输出电平并允许 / 禁止周期输出。
（14）设定复位同步 PWM。
（15）请通过 TOER 允许通道 3、 4 的输出。
（16）请通过 PFC 设定为 MTU2 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普通模式中重新开始时的运行

在复位同步 PWM 模式中发生异常，重新设定后在普通模式中重新开始时的说明如图 11.138 所示。

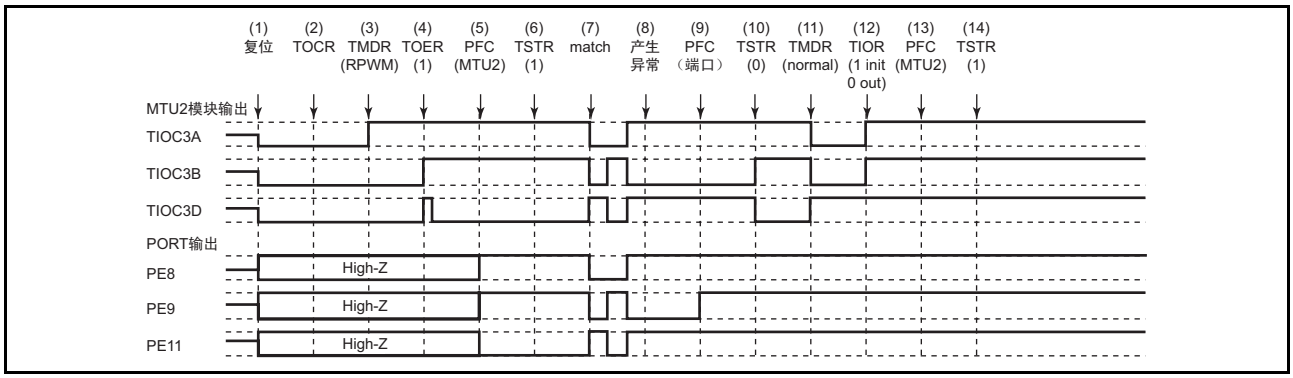


图 11.138　在复位同步 PWM 模式中发生异常并在普通模式中恢复时

（1）通过复位， MTU2 输出为低电平，端口为高阻抗状状态。
（2）请通过 TOCR 选择复位同步 PWM 的输出电平并允许 / 禁止周期输出。
（3）设定复位同步 PWM。
（4）请通过 TOER 允许通道 3、 4 的输出。
（5）请通过 PFC 设定为 MTU2 输出。
（6）通过 TSTR 开始计数运行。
（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）请通过 PFC 设定为端口输出，输出有效电平的反相信号。
（10）通过 TSTR 停止计数运行 （MTU2 输出为复位同步 PWM 输出的初始值）。
（11）请设定普通模式 （MTU2 输出的正相为低电平，反相为高电平）。
（12）请通过 TIOR 初始化引脚。
（13）请通过 PFC 设定为 MTU2 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 中重新开始时的说明如图 11.139 所示。

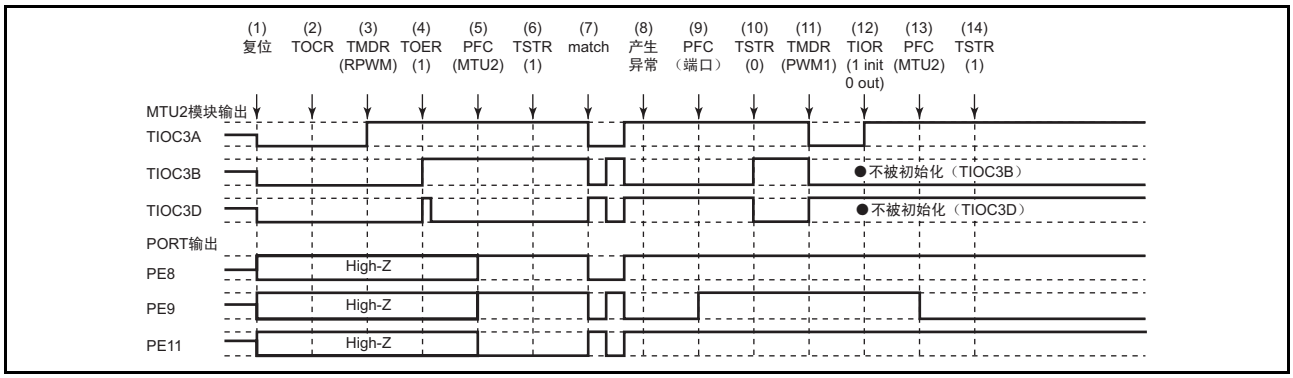


图 11.139　在复位同步 PWM 模式中发生异常并在 PWM 模式 1 中恢复时

（1）～（10）和图 11.138 通用。

（11）请设定 PWM 模式 1 （MTU2 输出的正相为低电平，反相为高电平）。

（12）请通过 TIOR 初始化引脚 （在 PWM 模式 1 中不初始化 TIOC*B）。

（13）请通过 PFC 设定为 MTU2 输出。

（14）通过 TSTR 重新开始。

（28）在复位同步 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 11.140 所示。

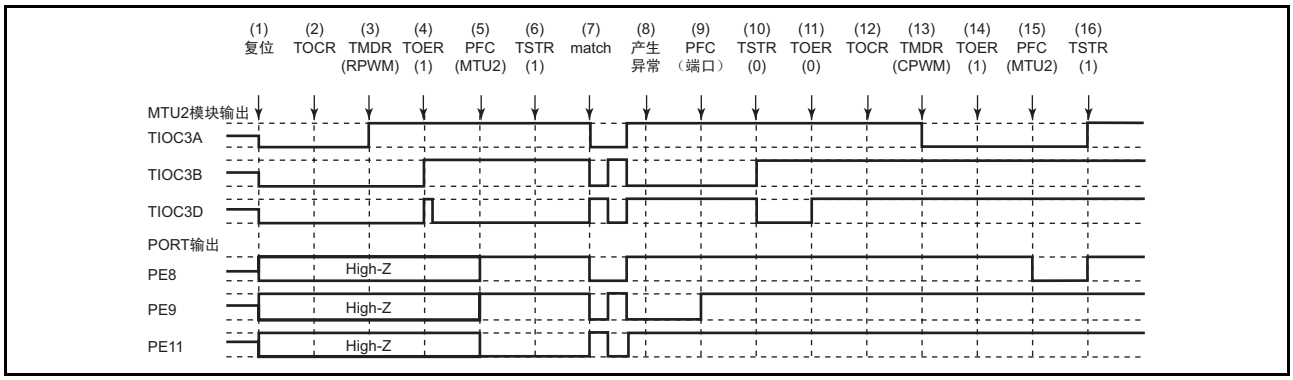


图 11.140　在复位同步 PWM 模式中发生异常并在互补 PWM 模式中恢复时

（1）～（10）和图 11.138 通用。
（11）请通过 TOER 禁止通道 3、 4 的输出。
（12）请通过 TOCR 选择互补 PWM 的输出电平并允许 / 禁止周期输出。
（13）设定互补 PWM （MTU2 的周期输出引脚为低电平）。
（14）请通过 TOER 允许通道 3、 4 的输出。
（15）请通过 PFC 设定为 MTU2 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 11.141 所示。

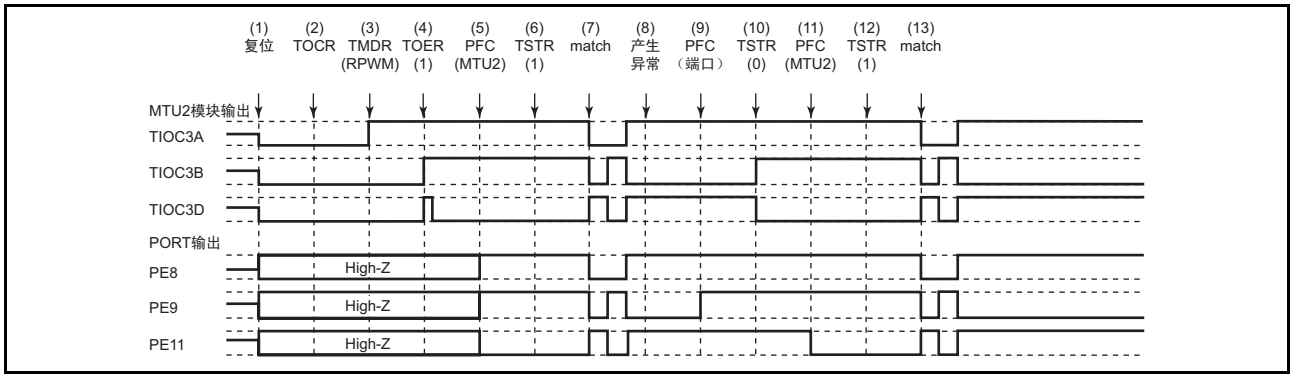


图 11.141　在复位同步 PWM 模式中发生异常并在复位同步 PWM 模式中恢复时

（1）～（10）和图 11.138 通用。
（11）请通过 PFC 设定为 MTU2 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　比较匹配定时器 （CMT）

本 LSI 内置由 2 个通道 16 位定时器构成的比较匹配定时器（CMT）。CMT 有 16 位计数器，能按各设定的周期产生中断。

12.1　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）
- 比较匹配时，可通过设定DMAC，选择产生DMAC传送请求或者中断请求
- 不使用CMT时，为降低功耗可以中止向CMT供给时钟使运行停止。

CMT 的框图如图 12.1 所示。

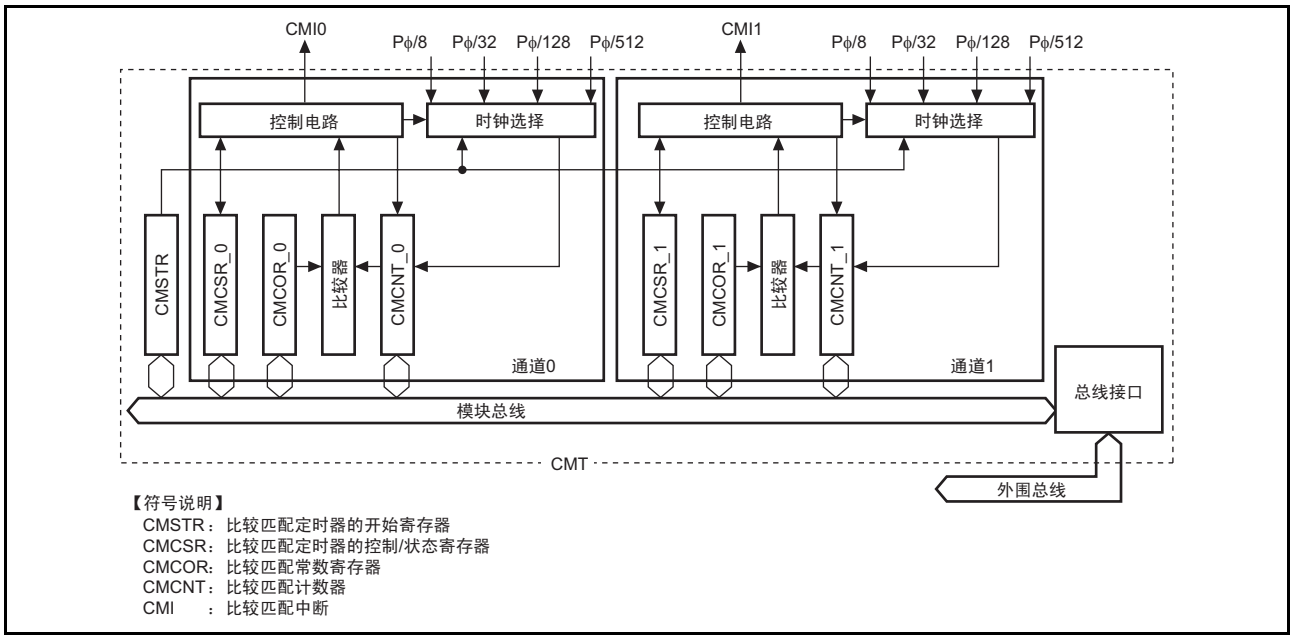


图 12.1　CMT 的框图

12.2　寄存器说明

CMT 中有以下寄存器。

表 12.1　寄存器结构

通道	寄存器名	简称	R/W	初始值	地址	存取长度
通用	比较匹配定时器启动寄存器	CMSTR	R/W	H'0000	H'FFFEC000	16
0	比较匹配定时器控制 / 状态寄存器 _0	CMCSR_0	R/W	H'0000	H'FFFEC002	16
	比较匹配计数器 _0	CMCNT_0	R/W	H'0000	H'FFFEC004	8、16
	比较匹配常数寄存器 _0	CMCOR_0	R/W	H'FFFF	H'FFFEC006	8、16
1	比较匹配定时器控制 / 状态寄存器 _1	CMCSR_1	R/W	H'0000	H'FFFEC008	16
	比较匹配计数器 _1	CMCNT_1	R/W	H'0000	H'FFFEC00A	8、16
	比较匹配常数寄存器 _1	CMCOR_1	R/W	H'FFFF	H'FFFEC00C	8、16

12.2.1　比较匹配定时器启动寄存器 （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	—	均为 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

12.2.2　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，允许 / 禁止中断以及设定计数器输入时钟。

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)*	R/W	R	R	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 ～ 8	—	均为 0	R	保留位 读写的值总是 0。
7	CMF	0	R/(W)*	比较匹配标志 表示 CMCNT 和 CMCOR 的值是否匹配。 0：CMCNT 和 CMCOR 的值不匹配 [清除条件] • 当读 CMF=1 后，对 CMF 写 0 时 1：CMCNT 和 CMCOR 的值匹配
6	CMIE	0	R/W	比较匹配中断允许 在 CMCNT 和 CMCOR 值匹配时 （CMF=1），选择允许还是禁止产生比较匹配中断（CMI）。 0：禁止比较匹配中断 （CMI） 1：允许比较匹配中断 （CMI）
5 ～ 2	—	均为 0	R	保留位 读写的值总是 0。
1、0	CKS[1:0]	00	R/W	时钟选择 从分频外围时钟 （Pϕ）的 4 种内部时钟中，选择输入 CMCNT 的时钟。当 CMSTR 的 STR 位被置 1 时， CMCNT 通过 CKS[1:0] 位所选择的时钟开始计数。 00：Pϕ/8 01：Pϕ/32 10：Pϕ/128 11：Pϕ/512

12.2.3　比较匹配计数器 （CMCNT）

CMCNT 是 16 位寄存器，用作递增计数器。通过 CMCSR 的 CKS[1:0] 位选择计数器输入时钟，并且将 CMST 的 STR 位置 1 时， CMCNT 通过所选的时钟开始计数。 CMCNT 的值和比较匹配常数寄存 （CMCOR）的值匹配时， CMCNT 被清除为 H'0000，并且 CMCSR 的 CMF 标志被置 1。

将比较匹配定时器启动寄存器 （CMSTR）对应通道的计数开始位从 1 清除为 0 时， CMCNT 被初始化为 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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

12.2.4　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。

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

12.3　运行说明

12.3.1　期间计数运行

通过 CMCSR 的 CKS[1:0] 位选择内部时钟，并且将 CMSTR 的 STR 位置 1 时，CMCNT 通过所选的时钟开始递增计数。当 CMCNT 的值和 CMCOR 的值匹配时，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 位被置 1，则请求比较匹配中断（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 12.2** 所示。

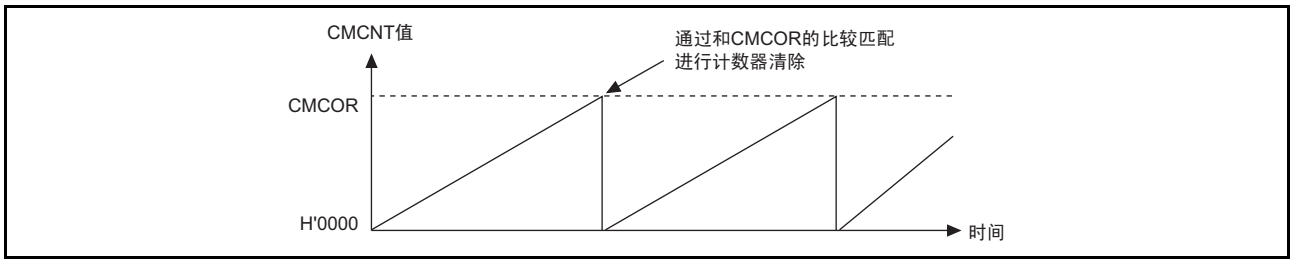


图 12.2　计数器的运行

12.3.2　CMCNT 的计数时序

通过 CMCSR 的 CKS[1:0] 位，能从分频外围时钟 （Pϕ）所得到的 4 种时钟 （Pϕ/8、 Pϕ/32、 Pϕ/128、 Pϕ/512）中选择一种。该时序如图 12.3 所示。

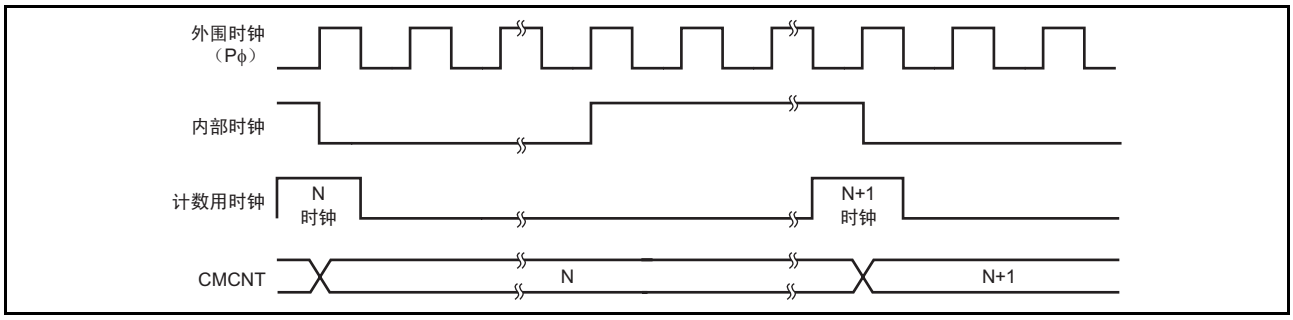


图 12.3　计数时序

12.4　中断

12.4.1　中断源和 DMA 传送请求

CMT 有各通道的比较匹配中断，分别分配了独立的向量地址。当比较匹配标志 （CMF）和中断允许位（CMIE）都被置 1 时，就输出相应的中断请求。通过中断请求启动 CPU 中断时，通过设定中断控制器，可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）”。

请在用户异常处理程序中将 CMF 位清 0，否则会再次产生中断。同时，在请求比较匹配中断时，通过设定直接存储器存取控制器 （DMAC），可启动 DMAC。此时不对 CPU 产生中断。不进行 DMAC 的启动设定时，对 CPU 产生中断请求。通过 DMAC 传送数据时， CMF 位自动清 0。

12.4.2　比较匹配标志的置位时序

CMCOR 和 CMCNT 匹配时，在匹配的最后状态（将 CMCNT 的值更新为 H'0000 的时序）产生比较匹配信号，并且将 CMCSR 的 CMF 位置 1。即，在 CMCOR 和 CMCNT 匹配后，不输入用于 CMCNT 计数器的时钟时，不产生比较匹配信号。 CMF 位的置位时序如图 12.4 所示。

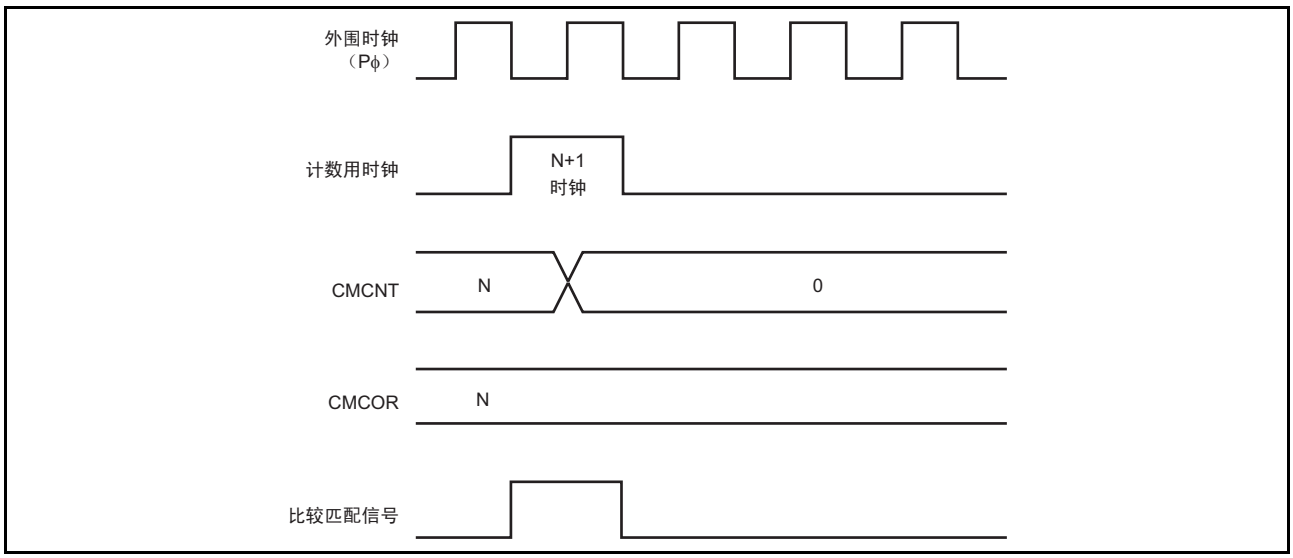


图 12.4　CMF 置位时序

12.4.3　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。但是如果启动了 DMAC，就在 DMAC 进行数据传送的时候，自动清除 CMF 位。

12.5　使用时的注意事项

12.5.1　CMCNT 的写入和比较匹配的竞争

如果在 CMCNT 计数器的写周期中的 T2 状态产生比较匹配信号，就不向 CMCNT 计数器进行写入而优先清除 CMCNT 计数器。此时序如图 12.5 所示。

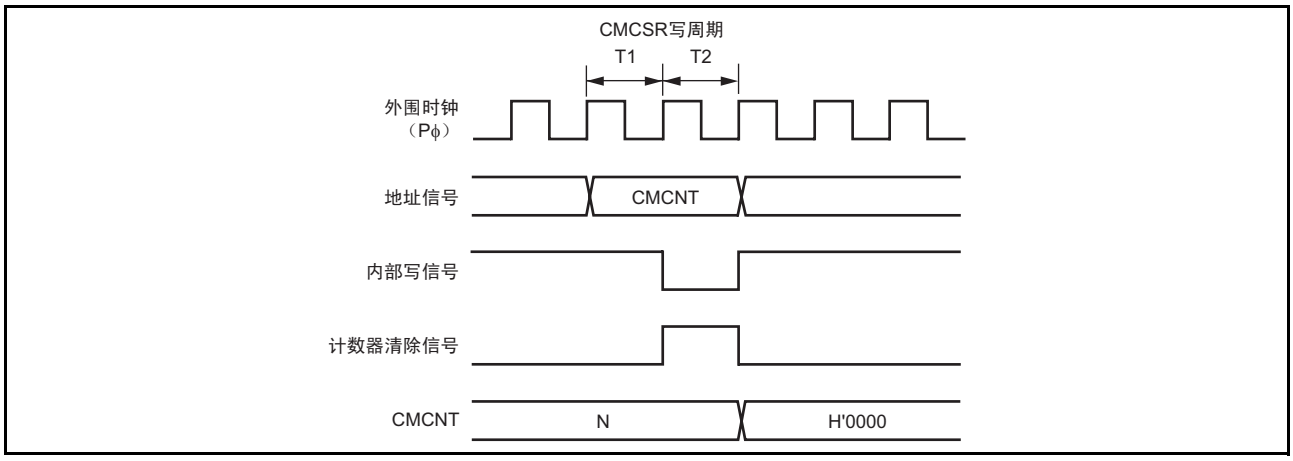


图 12.5　CMCNT 的写入和比较匹配的竞争

12.5.2　CMCNT 的字写入和递增计数的竞争

即使在 CMCNT 计数器的字写入周期中的 T2 状态发生递增计数，也不进行递增计数而优先写入计数器。此时序如图 12.6 所示。

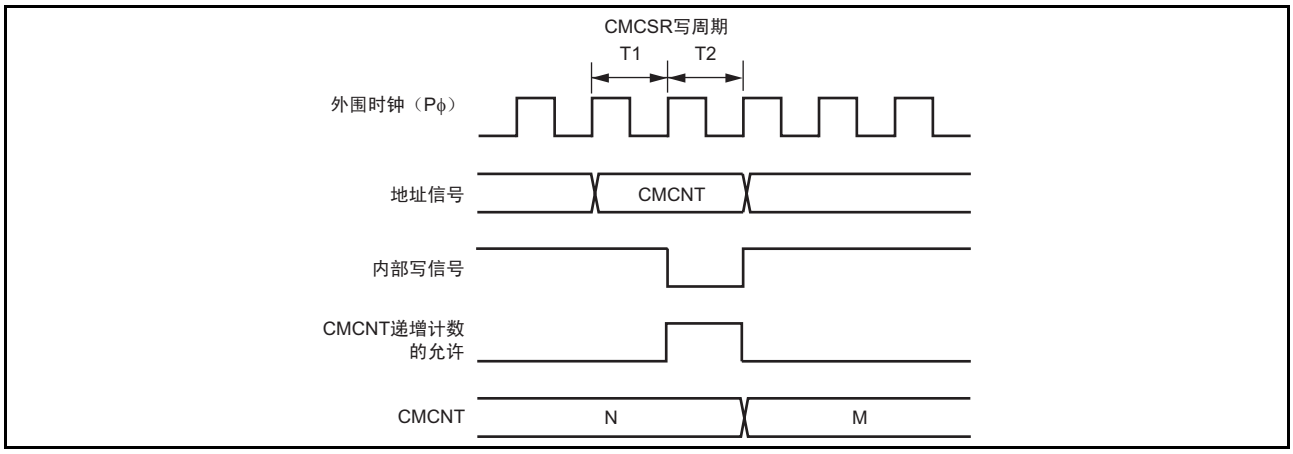


图 12.6　CMCNT 的字写入和递增计数的竞争

12.5.3　CMCNT 的字节写入和递增计数的竞争

即使在 CMCNT 的字节写入周期中的 T2 状态发生递增计数，也不进行写数据 （已进行写入）的递增计数而优先写计数器。并且，也不进行字节数据 （不进行写入）的递增计数而为写入之前的内容。

CMCNTH 写入周期中的 T2 状态发生递增计数时的时序如图 12.7 所示。

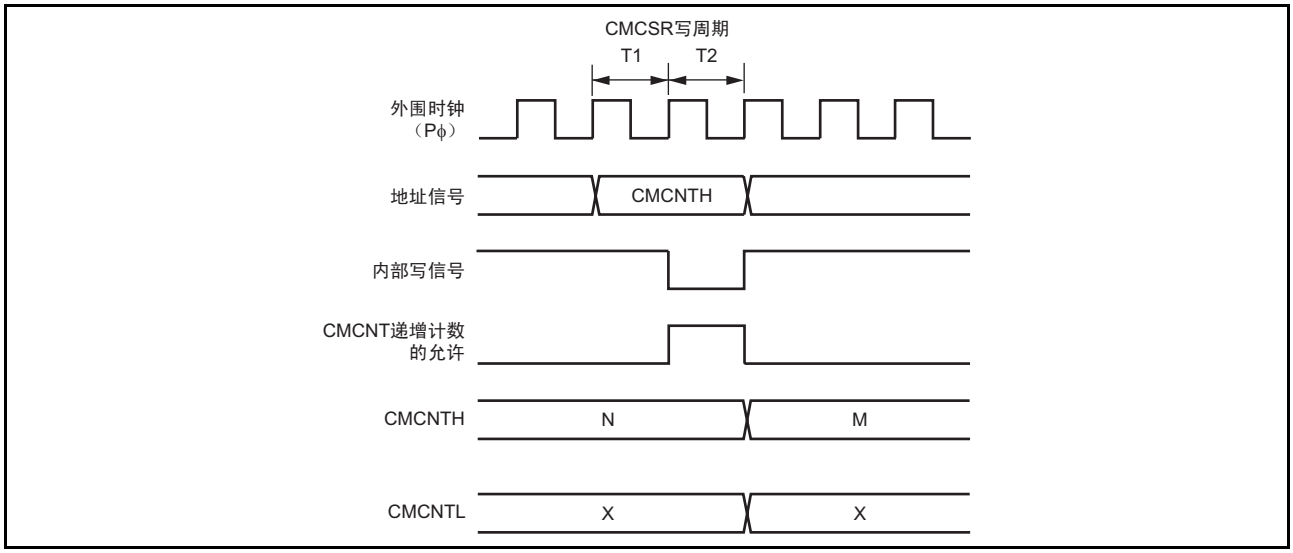


图 12.7　CMCNT 的字节写入和递增计数的竞争

第 13 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不能改写计数器的值而发生上溢时，就向外部输出上溢信号 （$\overline{\text{WDTOVF}}$）。同时，能产生本 LSI 内部复位信号。

WDT 是 1 个通道的定时器，用于计数解除软件待机模式或者频率变更时暂时待机状态的时钟振荡稳定时间。也可用作通常的看门狗定时器或者间隔定时器。

13.1　特点

- 能用于确保时钟振荡稳定时间
  在解除软件待机模式或者暂时待机状态 （时钟频率变更时）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
- 看门狗定时器模式时，输出$\overline{\text{WDTOVF}}$信号
  计数器上溢时，对外部输出$\overline{\text{WDTOVF}}$信号。此时，能选择是否同时复位本LSI内部。该内部复位能选择上电复位或者手动复位。
- 间隔定时器模式时发生中断
  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 （Pϕ×1～×1/16384）中选择外围时钟。

WDT 的框图如图 13.1 所示。

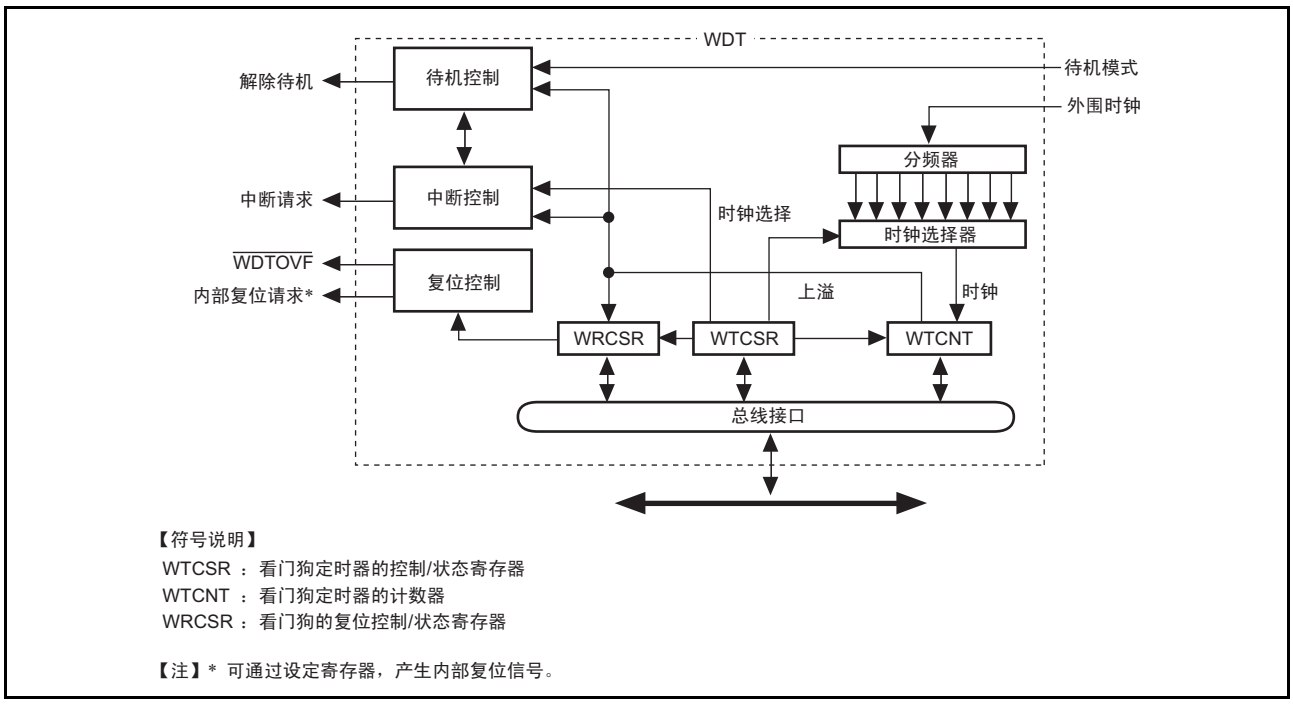


图 13.1　WDT 的框图

13.2　输入 / 输出引脚

WD 的引脚如表 13.1 所示。

表 13.1　引脚结构

名称	引脚名	输入 / 输出	功能
看门狗定时器的上溢	$\overline{\text{WDTOVF}}$	输出	看门狗定时器模式时输出计数器上溢信号

13.3　寄存器说明

WDT 有以下寄存器。

表 13.2　寄存器结构

寄存器名	简称	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'FFFE0002	16*
看门狗定时器的控制 / 状态寄存器	WTCSR	R/W	H'18	H'FFFE0000	16*
看门狗复位的控制 / 状态寄存器	WRCSR	R/W	H'1F	H'FFFE0004	16*

【注】 *　存取长度请参照 “13.3.4 寄存器存取时的注意事项”。

13.3.1　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递增计数的计数器。如果发生上溢，就会在看门狗定时器模式时产生看门狗定时器上溢信号 （$\overline{\text{WDTOVF}}$），并在间隔定时器模式时产生中断。

请在将高位字节置为 H'5A 后以字单位向 WTCNT 写入，以字节单位读取 WTCNT。

【注】 为了本寄存器不被误改写，写入方法和一般寄存器不同，详细内容请参照 “13.3.4 寄存器存取时的注意事项”。

位：	7	6	5	4	3	2	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

13.3.2　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，由选择计数使用时钟的位、上溢标志位和允许位构成。

用于计数软件待机模式解除时的时钟振荡稳定时间时，在计数器发生上溢后仍保持该值。

请将高位字节置为 H'A5 后以字单位向 WTCSR 写入。请以字节单位读 WTCSR 寄存器。

【注】 为了本寄存器不被误改写，写入方法和一般寄存器不同。详细内容请参照 “13.3.4 寄存器存取时的注意事项”。

位：	7	6	5	4	3	2	1	0
	IOVF	WT/$\overline{IT}$	TME	-	-	CKS[2:0]		
初始值：	0	0	0	1	1	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	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，在看门狗定时器模式中此位不被置位。 0：没有上溢 1：在间隔定时器模式中， WTCNT 已发生上溢 ［清除条件］ • 读 IOVF 后写 0
6	WT/$\overline{IT}$	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 【注】看门狗定时器模式中， WTCNT 上溢时向外部输出 $\overline{WDTOVF}$ 信号。如果在 WDT 运行时改写 WT/$\overline{IT}$，就有可能无法正确进行递增计数。
5	TME	0	R/W	定时器的允许 设定开始或者停止定时器的运行。如果在软件待机模式或者时钟频率变更时使用 WDT，请将此位设定为 0。 0：禁止定时器的运行。停止递增计数，保持 WTCNT 的值 1：允许定时器的运行
4、3	—	均为 1	R	保留位 读写的值总是 1。
2 ～ 0	CKS[2:0]	000	R/W	时钟选择 从分频外围时钟 （Pϕ）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 （Pϕ） =33MHz 的上溢周期值。 时钟分频比上溢周期 000：1×Pϕ （7.7μs） 001：1/64×Pϕ （500μs） 010：1/128×Pϕ （1.0ms） 011：1/256×Pϕ （2.0ms） 100：1/512×Pϕ （4.0ms） 101：1/1024×Pϕ （8.0ms） 110：1/4096×Pϕ （32ms） 111：1/16384×Pϕ （128ms） 【注】 如果在 WDT 运行时改写 CKS[2:0] 位，就有可能无法正确地进行递增计数。改写 CKS[2:0] 位时，必须使 WDT 停止运行。

13.3.3　看门狗复位的控制 / 状态寄存器 （WRCSR）

WRCSR 是 8 位可读 / 写寄存器，控制由看门狗定时器的计数器 （WTCNT）上溢而引起的内部复位信号的产生。

【注】 为了本寄存器不被误改写，写入方法和一般寄存器不同。详细内容请参照 “13.3.4 寄存器存取时的注意事项”。

位：	7	6	5	4	3	2	1	0
	WOVF	RSTE	RSTS	-	-	-	-	-
初始值：	0	0	0	1	1	1	1	1
R/W：	R/(W)	R/W	R/W	R	R	R	R	R

位	位名	初始值	R/W	说明
7	WOVF	0	R/(W)	看门狗定时器上溢 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：没有上溢 1：在看门狗定时器中 WTCNT 已发生上溢 ［清除条件］ • 读 WOVF 后写 0
6	RSTE	0	R/W	复位允许 看门狗定时器模式中， WTCNT 上溢时，选择是否产生复位本 LSI 内部的信号。 间隔定时器模式时，忽略设定值。 0：WTCNT 上溢时，不复位内部 * 1：WTCNT 上溢时，复位内部 【注】 * 本 LSI 内部不被复位，但 WDT 内的 WTCNT、 WTCSR 被复位。
5	RSTS	0	R/W	复位选择 看门狗定时器模式中， WTCNT 上溢时选择复位种类。 间隔定时器模式时，忽略设定值。 0：上电复位 1：手动复位
4 ～ 0	—	均为 1	R	保留位 读写的值总是 1。

13.3.4　寄存器存取时的注意事项

为了看门狗定时器计数器 （WTCNT）、看门狗定时器控制 / 状态寄存器和看门狗复位控制 / 状态寄存器（WRCSR）不易被改写，写入方法和一般寄存器不同。请按照以下方法进行读取或者写入。

（1） 向 WTCNT 及 WTCSR 的写入

在向 WTCNT 及 WTCSR 写入时，必须使用字传送指令。不能通过字节传送及长字传送指令进行写入。

如图 13.2 所示，在向 WTCNT 写入时，请将高位字节置为 H'5A、低位字节置为写入数据后进行传送；在向 WTCSR 写入时，请将高位字节置为 H'A5、低位字节置为写入后进行传送。如此进行传送，就能将低位字节的数据写到 WTCNT 或者 WTCSR。

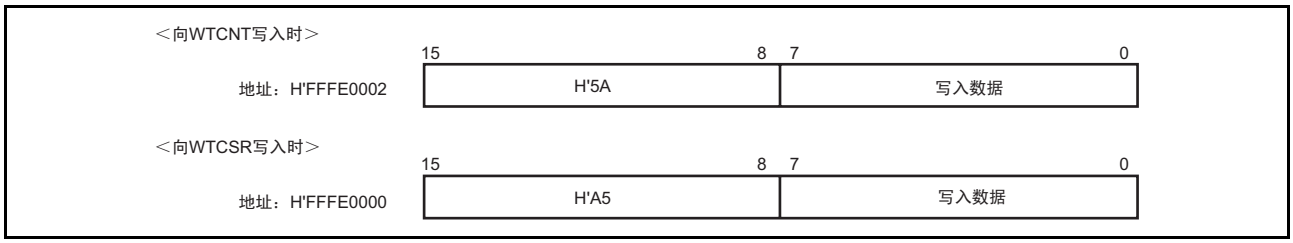


图 13.2　写 WTCNT 及 WTCSR

（2） 向 WRCSR 的写入

写入 WRCSR 时，请对地址 H'FFFE0004 进行字传送。不能通过字节传送指令及长字节传送指令进行写入。

在对 WOVF 位（bit7）写 0 时以及写 RSTE 位（bit6）和 RSTS 位（bit5）时，如图 13.3 所示，写入方法不同。

在对 WOVF 位写 0 时，请将高位字节置为 H'5A、低位字节置为 H'00 后进行传送。如此进行传送，就能将 WOVF 位清 0。此时 RSTE、RSTS 位不受影响。在对 RSTE、RSTS 位写入时，请将高位字节置为 H'5A、低位字节置为写数据进行传送。如此进行传送，低位字节的 bit6 和 bit5 的值就被各自写入到 RSTE 位和 RSTS 位。此时 WOVF 位不受影响。

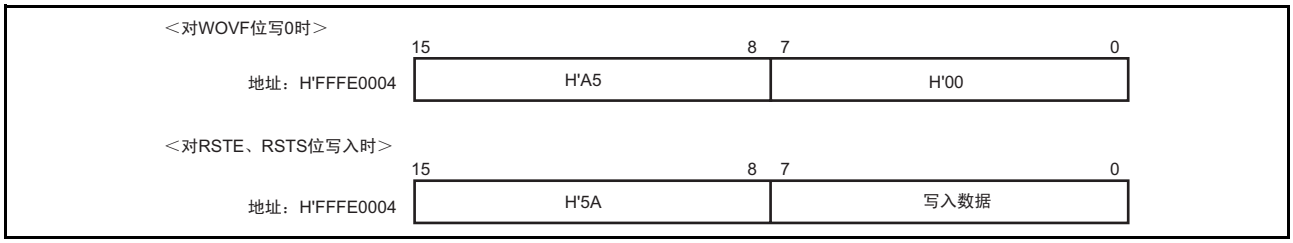


图 13.3　向 WRCSR 的写入

（3） 从 WTCNT、 WTCSR、 WRCSR 的读取

能用与普通寄存器相同的方法读取。WTCSR 分配到地址 H'FFFE0000、WTCNT 分配到地址 H'FFFE0002、WRCSR 分配到地址 H'FFFE0004。读取时，必须使用字节传送指令。

13.4 WDT 的使用方法

13.4.1 软件待机模式解除的步骤

通过 NMI 等中断解除软件待机模式时使用 WDT。步骤如下所示（通过复位进行解除时因为 WDT 不运行，所以在时钟振荡稳定前，请保持 $\overline{\text{RES}}$ 引脚和 $\overline{\text{MRES}}$ 引脚的低电平）。

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种类和计数器的初始值分别设定到WTCSR的CKS[2:0]位和WTCNT。在设定这些值时，计数上溢前的时间必须大于等于时钟振荡稳定时间。
3. 将待机控制寄存器（STBCR：参照“**第32章 低功耗模式**”）的STBY位设定为1后，通过执行SLEEP指令转移到软件待机模式，并且停止时钟。
4. 通过中断检测NMI信号变化的边沿等，开始WDT计数。
5. WDT计数上溢时， CPG开始提供时钟，本LSI重新开始运行。此时， WRCSR的WOVF不被置位。

13.4.2 频率变更的步骤

变更使用 PLL 的频率时，使用 WDT。当只是因转换分频器而引起频率变更时，不使用 WDT。

1. 必须在频率变更前，将WTCSR的TME位设定为0。将TME位设定为1时，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种类和计数器的初始值分别设定到WTCSR寄存器CKS[2:0]位和WTCNT计数器。在设定这些值时，计数上溢前的时间必须大于等于时钟振荡稳定时间。但是， WDT的递增计数通过设定后的时钟进行。
3. 改写频率控制寄存器 （FRQCR）时，处理器内部暂时停止， WDT开始计数。
4. WDT计数上溢时，CPG再次开始提供时钟，本LSI重新开始运行。此时，WRCSR的WOVF不被置位。
5. 计数器在H'00的值时停止运行。
6. 如果要在频率变更指令之后改写WTCNT，请在确认读取WTCNT并且为H'00后再进行改写。

13.4.3　看门狗定时器模式的用法

1. 将WTCSR的WT/$\overline{IT}$位设定为1，将计数器时钟的种类设定到CKS[2:0]，将是否复位本LSI的内部设定到WRCSR的RSTE位，将此时的复位类型设定到RSTS位，并将初始值设定到WTCNT。
2. WTCSR的TME位被设定为1时，在看门狗定时器模式开始计数。
3. 在看门狗定时器模式中运行时，为了使计数器不上溢请定期将计数器改写为H'00。
4. 计数器上溢时，WDT将WRCSR的WOVF置1，$\overline{WDTOVF}$信号被输出到外部，如**图13.4**所示。能使用此$\overline{WDTOVF}$信号复位系统。在64×Pϕ时钟之间输出$\overline{WDTOVF}$信号。
5. 将WRCSR的RSTE位置1时，能与$\overline{WDTOVF}$信号同时产生复位本LSI内部的信号。此复位能通过设定WRCSR的RSTS位，选择上电复位或者手动复位。在128×Pϕ时钟之间输出内部复位信号。
6. 同时发生$\overline{RES}$引脚输入信号引起的复位以及WDT上溢引起的复位时，优先进行$\overline{RES}$引脚引起的复位，WRCSR的WOVF位被清0。

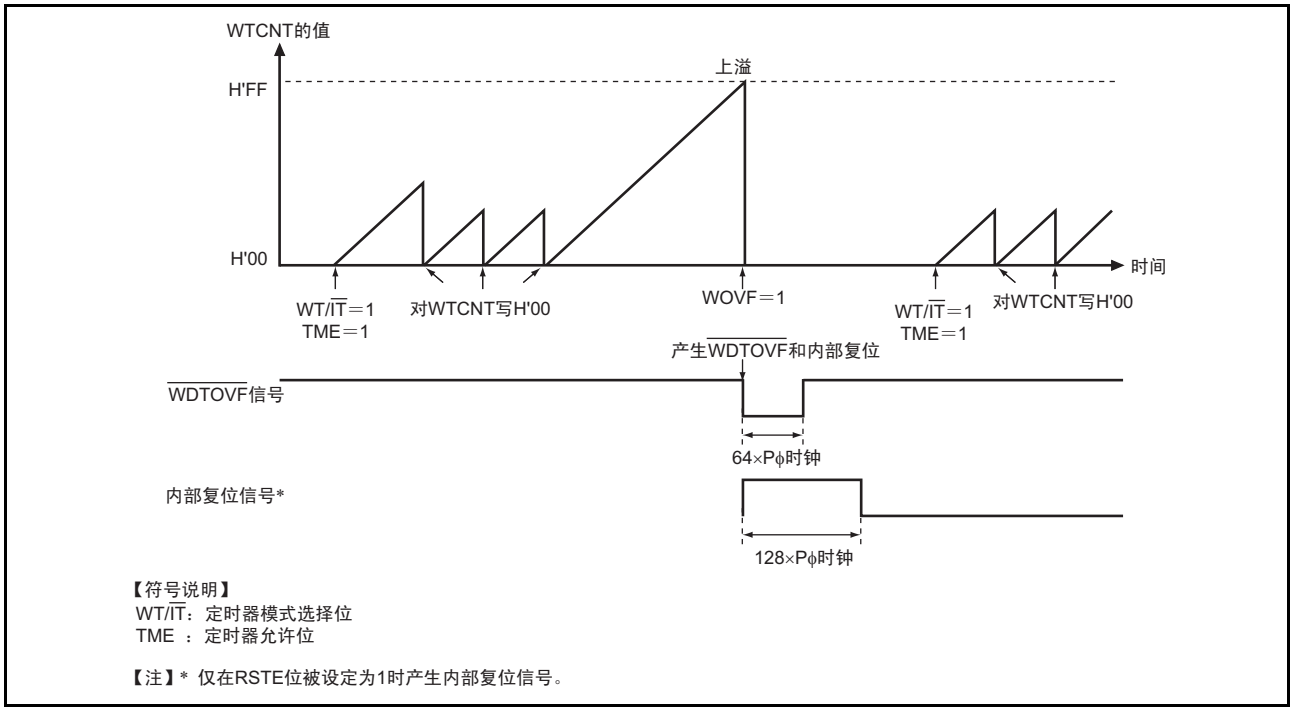


图 13.4　看门狗定时器模式时的运行

13.4.4　间隔定时器模式的用法

在间隔定时器模式中运行时，每当计数器发生上溢时，就产生间隔定时器中断。因此能每隔一定时间产生中断。

1. 将WTCSR的WT/$\overline{IT}$位置0，并将计数时钟的种类设定到CKS[2:0]位、初始值设定到WTCNT。
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。计数器继续计数。

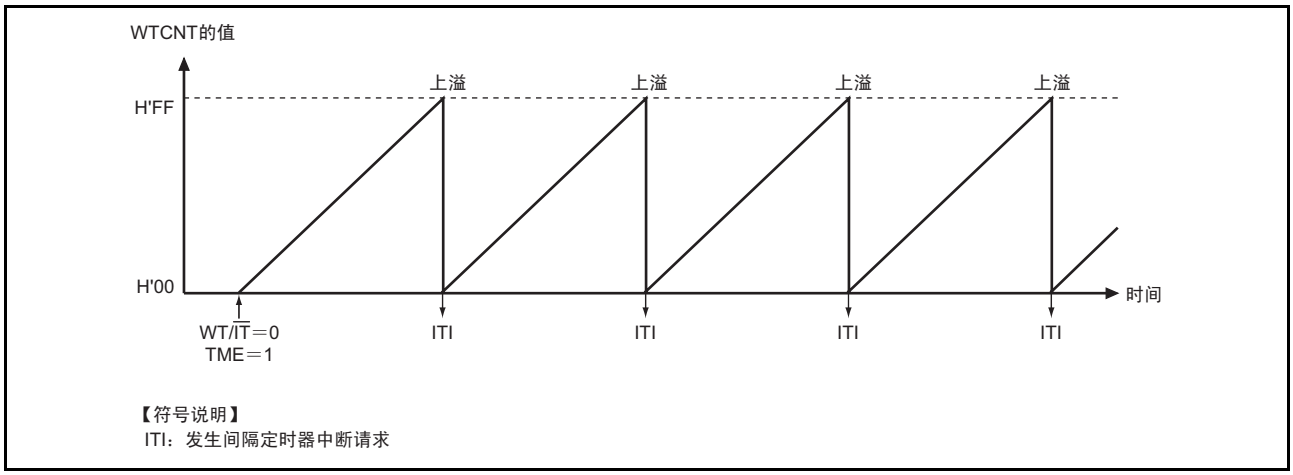


图 13.5　间隔定时器模式时的运行

13.5　使用时的注意事项

在间隔定时器模式及看门狗定时器模式中，有以下注意事项。

13.5.1　定时器误差

定时器运行开始后，根据以上电复位为基点的 WTCSR 寄存器的 TME 位的置位时序，WTCNT 寄存器最初的递增计数时序为从 Pϕ 的 1 个周期后 （最短）到通过 CKS[2:0] 选择的分频时序 （最长）的期间。第 2 次以后的递增计数时序为选择的分频时序。因此，到上述最初的递增计数为止的时间差为定时器误差。定时器运行中，改写 WTCNT 寄存器后，最初的递增计数时序也同样。

13.5.2　禁止将 H'FF 设定到 WTCNT

当 WTCNT 的值为 H'FF 时， WDT 就判定发生上溢。因此，在 WTCNT 设定 H'FF 时，与 CKS[2:0] 的选择时钟无关，会立即发生间隔定时器中断或者 WDT 复位。

13.5.3　间隔定时器上溢标志

WTCNT 的值为 H'FF 时，不能清除 WTCSR 的 IOVF 标志。

WTCNT 的值为 H'00 后请进行清除，或者将 WTCNT 的值改写为 H'FF 以外的值后清除 IOVF 标志。

13.5.4　$\overline{\text{WDTOVF}}$ 信号引起的系统复位

对本 LSI 的 $\overline{\text{RES}}$ 引脚输入 $\overline{\text{WDTOVF}}$ 信号时，不能正确初始化本 LSI。

不能逻辑性的对本 LSI 的 $\overline{\text{RES}}$ 引脚输入 $\overline{\text{WDTOVF}}$ 信号。 $\overline{\text{WDTOVF}}$ 信号系统整体复位时，请按图 13.6 所示的电路进行。

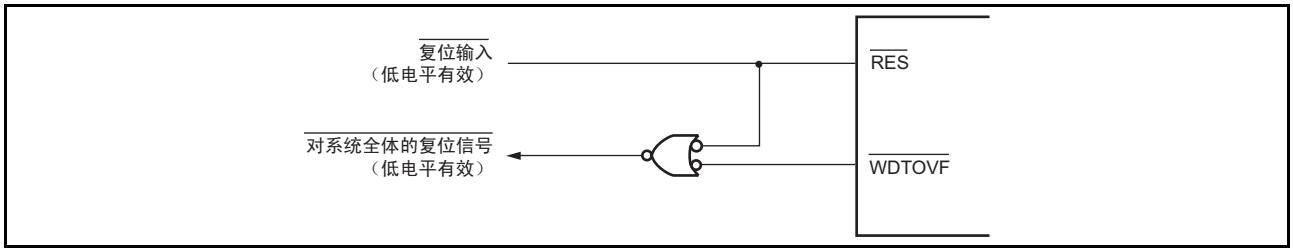


图 13.6　$\overline{\text{WDTOVF}}$ 信号引起的系统复位电路例

13.5.5　看门狗定时器模式的手动复位

发生看门狗定时器模式引起的手动复位时，保持总线周期。在总线释放中和 DMAC 突发传送中发生手动复位时，保留手动复位异常处理直到 CPU 获得总线权为止。

第 14 章　实时时钟 （RTC）

本 LSI 内置实时时钟 （RTC：Real Time Clock）及 RTC 用的 32.768kHz 晶体振荡电路。

14.1　特点

- 具有时钟/日历功能 （BCD表示）
  进行秒、分、小时、星期、日、月、年的计数
- 具有1～64Hz定时器 （二进制表示）
  664Hz计数器表示RTC分频电路中1Hz～64Hz的状态。
- 启动/停止功能
- 30秒调整功能
- 报警中断
  能选择是否与秒、分、小时、星期、日、月、年任一个进行比较作为报警中断条件
- 周期中断
  能从1/256秒、 1/64秒、 1/16秒、 1/4秒、 1/2秒、 1秒、 2秒周期中选择中断周期
- 进位中断
  表示秒计数器进位或者读取64Hz计数器时发生的64Hz计数器进位的进位中断功能
- 闰年自动修正功能

RTC 的框图如**图 14.1** 所示。

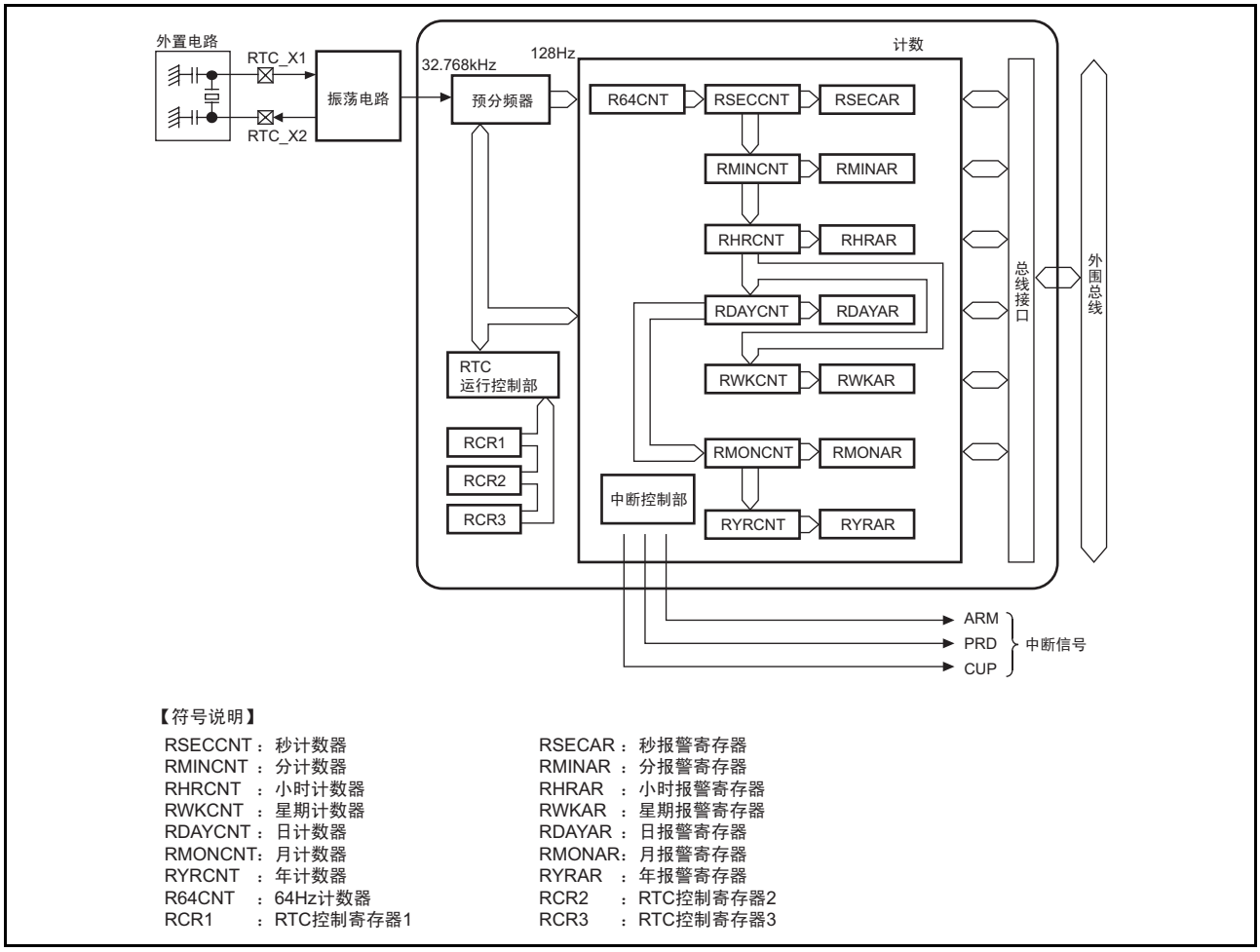


图 14.1　RTC 的框图

14.2 输入 / 输出引脚

RTC 的引脚结构如表 14.1 所示。

表 14.1 引脚结构

名称	引脚名	输入 / 输出	功能
RTC 用晶体振荡器引脚	RTC_X1	输入	给 RTC 连接 32.768kHz 的晶体振荡器。
RTC 用晶体振荡器引脚	RTC_X2	输出	给 RTC 连接 32.768kHz 的晶体振荡器。

14.3 寄存器说明

RTC 有以下寄存器：

表 14.2 寄存器结构

寄存器名	简称	R/W	初始值	地址	存取长度
64Hz 计数器	R64CNT	R	H'xx	H'FFFF2000	8
秒计数器	RSECCNT	R/W	H'xx	H'FFFF2002	8
分计数器	RMINCNT	R/W	H'xx	H'FFFF2004	8
小时计数器	RHRCNT	R/W	H'xx	H'FFFF2006	8
星期计数器	RWKCNT	R/W	H'0x	H'FFFF2008	8
日计数器	RDAYCNT	R/W	H'xx	H'FFFF200A	8
月计数器	RMONCNT	R/W	H'xx	H'FFFF200C	8
年计数器	RYRCNT	R/W	H'xxxx	H'FFFF200E	16
秒报警寄存器	RSECAR	R/W	H'xx	H'FFFF2010	8
分报警寄存器	RMINAR	R/W	H'xx	H'FFFF2012	8
小时报警寄存器	RHRAR	R/W	H'xx	H'FFFF2014	8
星期报警寄存器	RWKAR	R/W	H'0x	H'FFFF2016	8
日报警寄存器	RDAYAR	R/W	H'xx	H'FFFF2018	8
月报警寄存器	RMONAR	R/W	H'xx	H'FFFF201A	8
年报警寄存器	RYRAR	R/W	H'xxxx	H'FFFF2020	16
RTC 控制寄存器 1	RCR1	R/W	H'00	H'FFFF201C	8
RTC 控制寄存器 2	RCR2	R/W	H'09	H'FFFF201E	8
RTC 控制寄存器 3	RCR3	R/W	H'00	H'FFFF2024	8

14.3.1　64Hz 计数器 （R64CNT）

R64CNT 表示 RTC 分频电路中 1Hz ～ 64Hz 的状态。

发生来自 128Hz 分频段的进位时，如果读取此寄存器，就表示 RTC 控制寄存器 1 （RCR1）的 CF 位被置 1、64Hz 计数器的读取与进位重合。此时读出的值无效，因此在对 RCR1 的 CF 位写 0 并清除后，需要再次读取 R64CNT。

将 RTC 控制寄存器 2 （RCR2）的 RESET 位置 1 或者将 RCR2 的 ADJ 位置 1 时，初始化 RTC 的分频电路与 R64CNT。

位：	7	6	5	4	3	2	1	0
	-	1Hz	2Hz	4Hz	8Hz	16Hz	32Hz	64Hz
初始值：	0	不定	不定	不定	不定	不定	不定	不定
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 读写的值总是 0。
6	1Hz	不定	R	表示 RTC 分频电路中 1Hz ～ 64Hz 的状态。
5	2Hz	不定	R	
4	4Hz	不定	R	
3	8Hz	不定	R	
2	16Hz	不定	R	
1	32Hz	不定	R	
0	64Hz	不定	R	

14.3.2　秒计数器 （RSECCNT）

RSECCNT 是用于设定 / 计数秒 （RTC 被 BCD 码化的秒部分）的计数器，根据 64Hz 计数器每秒的进位进行计数运行。

可设定的范围为 00 ～ 59 （十进制 （BCD））。设定的值不是 00 ～ 59 （十进制 （BCD））时，不能正常运行。请通过设置 RCR2 的 START 位停止计数运行之后，再进行写入。

位：	7	6	5	4	3	2	1	0
	-	10秒			1秒			
初始值：	0	不定	不定	不定	不定	不定	不定	不定
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	—	0	R	保留位 读写的值总是 0。
6 ～ 4	10 秒	不定	R/W	秒十位计数 秒十位从 0 到 5 计数，进行 60 秒的计数。
3 ～ 0	1 秒	不定	R/W	秒个位计数 秒个位对每秒从 0 到 9 计数。发生进位时，秒十位加 1。

14.3.3　分计数器 （RMINCNT）

RMINCNT 是用于设定 / 计数分（RTC 被 BCD 码化的分钟部分）的计数器，根据秒计数器每分的进位进行计数运行。

可设定的范围为 00 ～ 59 （10 进制 （BCD））。设定的值不是 00 ～ 59 （十进制 （BCD））时，不能正常运行。请通过设置 RCR2 的 START 位停止计数运行之后，再进行写入。

位：	7	6	5	4	3	2	1	0
	-	10分			1分			
初始值：	0	不定	不定	不定	不定	不定	不定	不定
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	—	0	R	保留位 读写的值总是 0。
6 ～ 4	10 分	不定	R/W	分十位计数 分十位从 0 到 5 计数，进行 60 分的计数。
3 ～ 0	1 分	不定	R/W	分个位计数 分个位对每分钟从 0 到 9 计数。发生进位时，分十位加 1。

14.3.4　小时计数器 （RHRCNT）

RHRCNT 是用于设定 / 计数小时 （RTC 被 BCD 码化的小时部分）的计数器，根据分计数器每小时的进位进行计数运行。

可设定的范围为 00 ～ 23 （10 进制 （BCD））。设定的值不是 00 ～ 23 （十进制 （BCD））时，不能正常运行。请通过设置 RCR2 的 START 位停止计数运行之后，再进行写入。

位：	7	6	5	4	3	2	1	0
	-	-	10小时		1小时			
初始值：	0	0	不定	不定	不定	不定	不定	不定
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	—	均为 0	R	保留位 读写的值总是 0。
5、4	10 小时	不定	R/W	小时十位计数 小时十位从 0 到 2 计数。
3 ～ 0	1 小时	不定	R/W	小时个位计数 小时个位对每小时从 0 到 9 计数。发生进位时，小时十位加 1。

14.3.5　星期计数器 （RWKCNT）

RWKCNT 是用于设定 / 计数星期 （RTC 被 BCD 码化的星期部分）的计数器，根据小时计数器每天的进位进行计数运行。

可设定的范围为 0 ～ 6（10 进制 （BCD））。设定的值不是 0 ～ 6（十进制 （BCD））时，不能正常运行。请通过设置 RCR2 的 START 位停止计数运行之后，再进行写入。

位：	7	6	5	4	3	2	1	0
	-	-	-	-	-	星期几		
初始值：	0	0	0	0	0	不定	不定	不定
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	—	均为 0	R	保留位 读写的值总是 0。
2 ～ 0	星期	不定	R/W	星期计数 用二进制码表示星期。 000：星期日 001：星期一 010：星期二 011：星期三 100：星期四 101：星期五 110：星期六 111：保留 （禁止设定）

14.3.6　日计数器 （RDAYCNT）

RDAYCNT 是用于设定 / 计数日 （RTC 被 BCD 码化的日部分）的计数器，根据小时计数器每天的进位进行计数运行。

可设定的范围为 01 ～ 31 （10 进制 （BCD））。设定的值不是 01 ～ 31 （十进制 （BCD））时，不能正常运行。请通过设置 RCR2 的 START 位停止计数运行之后，再进行写入。

因为 RDAYCNT 的设定范围根据每月及闰年变化，所以请在确认之后再作设定。闰年将年计数器（RYRCNT）作为西历，通过能否被 400、 100、 4 除尽来计算。

位：	7	6	5	4	3	2	1	0
	-	-	10日		1日			
初始值：	0	0	不定	不定	不定	不定	不定	不定
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	—	均为 0	R	保留位 读写的值总是 0。
5、4	10 日	不定	R/W	日十位计数
3 ～ 0	1 日	不定	R/W	日个位计数 日个位对每日从 0 到 9 计数。发生进位时，日十位加 1。

14.3.7　月计数器 （RMONCNT）

RMONCNT 是用于设定 / 计数月 （RTC 被 BCD 码化的月部分）的计数器，根据日计数器每月的进位进行计数运行。

可设定的范围为 01 ～ 12 （10 进制 （BCD））。设定的值不是 01 ～ 12 （十进制 （BCD））时，不能正常运行。请通过设置 RCR2 的 START 位停止计数运行之后，再进行写入。



位	位名	初始值	R/W	说明
7 ～ 5	—	均为 0	R	保留位 读写的值总是 0。
4	10 月	不定	R/W	月十位计数
3 ～ 0	1 月	不定	R/W	月个位计数 月个位对每月从 0 到 9 计数。发生进位时，月十位加 1。

14.3.8　年计数器 （RYRCNT）

RYRCNT 是用于设定 / 计数年 （RTC 被 BCD 码化的年部分）的计数器，根据月计数器每年的进位进行计数运行。

可设定的范围为 0000 ～ 9999 （10 进制 （BCD））。设定的值不是 0000 ～ 9999 （十进制 （BCD））时，不能正常运行。请通过设置 RCR2 的 START 位停止计数运行之后，再进行写入。



位	位名	初始值	R/W	说明
15 ～ 12	1000 年	不定	R/W	年千位计数
11 ～ 8	100 年	不定	R/W	年百位计数
7 ～ 4	10 年	不定	R/W	年十位计数
3 ～ 0	1 年	不定	R/W	年个位计数

14.3.9　秒报警寄存器 （RSECAR）

RSECAR 是对应秒（RTC 被 BCD 码化的秒部分）计数器 RSECCNT 的报警寄存器。将 ENB 位置 1 时，与 RSECCNT 的值进行比较。报警寄存器 （RSECAR、 RMINAR、 RHRAR、 RWKAR、 RDAYAR、 RMONAR、 RCR3）中，只有 ENB 位被置 1 的报警寄存器才进行计数器和报警寄存器的比较，各自全部匹配时，将 RCR1 报警标志置 1。

可设定的范围为 00 ～ 59+ENB 位 （10 进制 （BCD））， 设定值不是 00 ～ 59+ENB 位 （10 进制 （BCD））时，不能正常运行。

位：	7	6	5	4	3	2	1	0
	ENB	10秒			1秒			
初始值：	0	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	ENB	0	R/W	置 1 时，与 RSECCNT 的值进行比较。
6 ～ 4	10 秒	不定	R/W	秒十位的设定值
3 ～ 0	1 秒	不定	R/W	秒个位的设定值

14.3.10　分报警寄存器 （RMINAR）

RMINAR 是对应分（RTC 的被 BCD 码化的分部分）计数器 RMINCNT 的报警寄存器。将 ENB 位置 1 时，与 RMINCNT 的值进行比较。报警寄存器（RSECAR、RMINAR、RHRAR、RWKAR、RDAYAR、RMONAR、RCR3）中，只有 ENB 位被置 1 的报警寄存器才进行计数器和报警寄存器的比较，各自全部匹配时，将 RCR1 报警标志置 1。

可设定的范围为 00 ～ 59+ENB （10 进制 （BCD））位，设定的值不是 00 ～ 59+ENB （10 进制 （BCD））时，不能正常运行。

位：	7	6	5	4	3	2	1	0
	ENB	10分			1分			
初始值：	0	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	ENB	0	R/W	置 1 时，与 RMINCNT 的值进行比较。
6 ～ 4	10 分	不定	R/W	分十位的设定值
3 ～ 0	1 分	不定	R/W	秒个位的设定值

14.3.11　小时报警寄存器 （RHRAR）

RHRAR 是对应小时 （RTC 被 BCD 码化的小时部分）计数器 RHRCNT 的报警寄存器。将 ENB 位置 1 时，与 RHRCNT 的值进行比较。报警寄存器（RSECAR、RMINAR、RHRAR、RWKAR、RDAYAR、RMONAR、RCR3）中，只有 ENB 位被置 1 的报警寄存器才进行计数器和报警寄存器的比较、各自全部匹配时，将 RCR1 报警标志置 1。

可设定的范围为 00 ～ 23+ENB 位（10 进制（BCD）），设定的值不是 00 ～ 23+ENB 位（10 进制（BCD））时，不能正常运行。

位：	7	6	5	4	3	2	1	0
	ENB	-	10小时		1小时			
初始值：	0	0	不定	不定	不定	不定	不定	不定
R/W：	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7	ENB	0	R/W	置 1 时，与 RHRCNT 的值进行比较。
6	—	0	R	保留位　读写的值总是 0。
5、4	10 小时	不定	R/W	小时十位的设定值
3 ～ 0	1 小时	不定	R/W	小时个位的设定值

14.3.12　星期报警寄存器 （RWKAR）

RWKAR 是对应星期（RTC 被 BCD 码化的星期部分）计数器 RWKCNT 的报警寄存器。将 ENB 位置 1 时，与 RWKCNT 的值进行比较。报警寄存器（RSECAR、RMINAR、RHRAR、RWKAR、RDAYAR、RMONAR、RCR3）中，只有 ENB 位被置 1 的报警寄存器才进行计数器和报警寄存器的比较，各自全部匹配时，将 RCR1 报警标志置 1。

可设定的范围为 0 ～ 6+ENB 位 （10 进制 （BCD）），设定的值不是 0 ～ 6+ENB 位 （10 进制 （BCD））时，不能正常运行。

位：	7	6	5	4	3	2	1	0
	ENB	-	-	-	-	星期几		
初始值：	0	0	0	0	0	不定	不定	不定
R/W：	R/W	R	R	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7	ENB	0	R/W	置 1 时，与 RWKCNT 的值进行比较。
6 ～ 3	—	均为 0	R	保留位 读写的值总是 0。
2 ～ 0	星期	不定	R/W	星期几的设定值 000：星期日 001：星期一 010：星期二 011：星期三 100：星期四 101：星期五 110：星期六 111：保留 （禁止设定）

14.3.13　日报警寄存器 （RDAYAR）

RDAYAR 是对应日 （RTC 被 BCD 码化的日部分）计数器 RDAYCNT 的报警寄存器。将 ENB 位置 1 时，与 RDAYCNT 的值进行比较。报警寄存器（RSECAR、RMINAR、RHRAR、RWKAR、RDAYAR、RMONAR、RCR3）中，只有 ENB 位被置 1 的报警寄存器才进行计数器和报警寄存器的比较，各自全部匹配时，将 RCR1 报警标志置 1。

可设定的范围为 00 ～ 31+ENB 位（10 进制（BCD））， 设定的值不是 00 ～ 31+ENB 位（10 进制（BCD））时，不能正常运行。

位：	7	6	5	4	3	2	1	0
	ENB	-	10日		1日			
初始值：	0	0	不定	不定	不定	不定	不定	不定
R/W：	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7	ENB	0	R/W	置 1 时，与 RDAYCNT 的值进行比较。
6	—	0	R	保留位 读写的值总是 0。
5、4	10 日	不定	R/W	日十位的设定值
3 ～ 0	1 日	不定	R/W	日个位的设定值

14.3.14　月报警寄存器 （RMONAR）

RMONAR 是对应月 （RTC 被 BCD 码化的月部分）计数器 RMONCNT 的报警寄存器。将 ENB 位置 1 时，与 RMONCNT 的值进行比较。报警寄存器（RSECAR、RMINAR、RHRAR、RWKAR、RDAYAR、RMONAR、RCR3）中，只有 ENB 位被置 1 的报警寄存器才进行计数器和报警寄存器的比较，各自全部匹配时，将 RCR1 报警标志置 1。

可设定的范围为 01 ～ 12+ENB 位（10 进制（BCD））， 设定的值不是 01 ～ 12+ENB 位（10 进制（BCD））时，不能正常运行。

位：	7	6	5	4	3	2	1	0
	ENB	-	-	10月	1月			
初始值：	0	0	0	不定	不定	不定	不定	不定
R/W：	R/W	R	R	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7	ENB	0	R/W	置 1 时，与 RMONCNT 的值进行比较。
6、5	—	均为 0	R	保留位　读写的值总是 0。
4	10 月	不定	R/W	月十位的设定值
3 ～ 0	1 月	不定	R/W	月个位的设定值

14.3.15　年报警寄存器 （RYRAR）

RYRAR 是对应年 （RTC 被 BCD 码化的年部分）计数器 RYRCNT 的报警寄存器。

可设定的范围为 0000 ～ 9999 （10 进制 （BCD）），设定的值不是 0000 ～ 9999 （10 进制 （BCD））时，不能正常运行。



位	位名	初始值	R/W	说明
15 ～ 12	1000 年	不定	R/W	年千位的设定值
11 ～ 8	100 年	不定	R/W	年百位的设定值
7 ～ 4	10 年	不定	R/W	年十位的设定值
3 ～ 0	1 年	不定	R/W	年个位的设定值

14.3.16　RTC 控制寄存器 1 （RCR1）

RCR1 是关于进位及报警标志的寄存器。同时，对于各自的标志能选择是否产生中断。

分频电路被复位 （将 RCR2 的 RESET 位和 ADJ 位置 1）前， CF 标志不定。使用 CF 标志时，必须在使用前复位分频电路。

位：	7	6	5	4	3	2	1	0
	CF	-	-	CIE	AIE	-	-	AF
初始值：	不定	0	0	0	0	0	0	0
R/W：	R/W	R	R	R/W	R/W	R	R	R/W

位	位名	初始值	R/W	说明
7	CF	不定	R/W	进位标志 表示将该标志置 1 时，秒计数器进位或者读取 64Hz 计数器时发生的 64Hz 计数器进位，此时不能保证读出的计数寄存器的值。需要再次读出。 0：没有秒计数器进位或读取 64Hz 计数器时的 64Hz 计数器进位 ［清除条件］对 CF 写 0 时 1：有秒计数器进位或读取 64Hz 计数器时的 64Hz 计数器进位 ［置位条件］秒计数器进位或者读取 64Hz 计数器进位时的 64Hz 计数器进位；或者对 CF 写 1 时
6、5	—	均为 0	R	保留位 读写的值总是 0。
4	CIE	0	R/W	进位中断允许标志 将进位标志 （CF）置 1 时，允许产生中断的位。 0：将 CF 标志被置 1 时，不允许发生进位中断 1：将 CF 标志置 1 时，允许发生进位中断
3	AIE	0	R/W	报警中断允许标志 将报警标志 （AF）置 1 时，允许产生中断的位。 0：将 AF 标志置 1 时，不允许产生报警中断 1：将 AF 标志被置为 1 时，允许产生报警中断
2、1	—	均为 0	R	保留位 读写的值总是 0。
0	AF	0	R/W	报警标志 AF 标志是在报警寄存器 （RSECAR、 RMINAR、 RHRAR、 RWKAR、 RDAYAR、 RMONAR、 RYRAR）设定的报警时间 （仅限 ENB 位设定为 1 的寄存器）与计数器匹配时，被置 1 的标志。 0：报警寄存器与计数器不匹配 ［清除条件］对 AF 写 0 时 1：报警寄存器与计数器匹配 * ［置位条件］报警寄存器 （仅限 ENB 位设定为 1 的寄存器）与计数器匹配时 【注】 * 写 1 时，保持原值。

14.3.17　RTC 控制寄存器 2 （RCR2）

RCR2 是关于周期中断控制、 30 秒调整、分频电路复位及 RTC 计数控制的寄存器。

RCR2 通过上电复位或在深度待机模式时被初始化。手动复位时，初始化 RTCEN 位以及 START 位以外的位。

位：	7	6	5	4	3	2	1	0
	PEF	PES[2:0]			RTCEN	ADJ	RESET	START
初始值：	0	0	0	0	1	0	0	1
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	PEF	0	R/W	周期中断标志 表示在 PES2 ～ PES0 位设定的周期中产生中断的标志。将该标志置 1 时，产生周期中断。 0：在 PES2 ～ PES0 位设定的周期中未产生中断 ［清除条件］对 PEF 写 0 时 1：在 PES2 ～ PES0 位设定的周期中产生中断 ［置位条件］在 PES2 ～ PES0 位设定的周期中产生中断时，或者对 PEF 写 1 时
6 ～ 4	PES[2:0]	000	R/W	中断允许标志 设定周期中断的周期。 000：没有产生周期中断 001：产生周期中断的周期为每 1/256 秒 010：产生周期中断的周期为每 1/64 秒 011：产生周期中断的周期为每 1/16 秒 100：产生周期中断的周期为每 1/4 秒 101：产生周期中断的周期为每 1/2 秒 110：产生周期中断的周期为每 1 秒 111：产生周期中断的周期为每 2 秒
3	RTCEN	1	R/W	振荡器有效 控制 RTC 用晶体振荡器的运行。 0：使 RTC 用晶体振荡器停止 1：使 RTC 用晶体振荡器运行
2	ADJ	0	R/W	30 秒调整 用于 30 秒调整。通过写 1，在 29 秒以前舍为 00 秒、 30 秒以后进位为 1 分。此时，分频电路（RTC 预分频器以及 R64CNT）也同时被复位。因为此 ADJ 位自动变为 0，所以不需要特意写 0，读位时总是读出 0。 0：通常的时钟运行 1：进行 30 秒的调整
1	RESET	0	R/W	复位 通过写 1，初始化分频电路。另外，写 1 时，分频电路（RTC 预分频器以及 R64CNT）被复位后，此 RESET 位自动变为 0，所以不需要特意写 0，读位时总是读出 0。 0：通常的时钟运行 1：复位分频电路

位	位名	初始值	R/W	说明
0	START	1	R/W	START 位 停止以及重新启动计数器 （时钟）运行的位。 0：秒、分、小时、日、星期、月、年计数器停止 * 1：秒、分、小时、日、星期、月、年计数器正常运行 * 【注】* 只有不通过 RTCEN 位使 64Hz 计数器停止，该计数器才会持续运行。

14.3.18　RTC 控制寄存器 3 （RCR3）

将 ENB 位置 1 时， RCR3 与 RYRCNT 的值进行比较。报警寄存器 （RSECAR、 RMINAR、 RHRAR、 RWKAR、RDAYAR、RMONAR、RCR3）中，只有 ENB 位被置 1 的报警寄存器才进行计数器和报警寄存器的比较，各自全部匹配时，将 RCR1 报警标志置 1。

位：	7	6	5	4	3	2	1	0
	ENB	-	-	-	-	-	-	-
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	ENB	0	R/W	置 1 时，与 RYRCNT 的值进行比较。
6 ～ 0	—	均为 0	R	保留位 读写的值总是 0。

14.4　运行说明

RTC 的使用例。

14.4.1　上电后寄存器的初始设定

上电后请对所有寄存器进行初始设定。

14.4.2　时间的设定步骤

时间设定步骤例如图 14.2 所示。

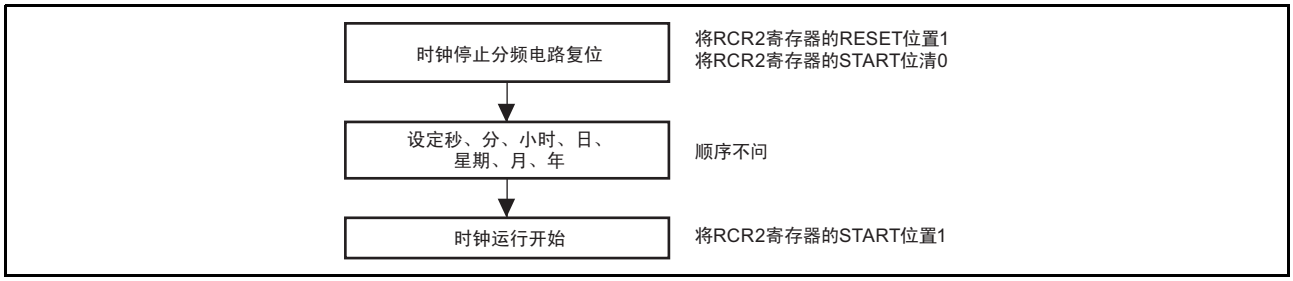


图 14.2　时间设定步骤

14.4.3　时间的读出步骤

时间的读出步骤如图 14.3 所示。

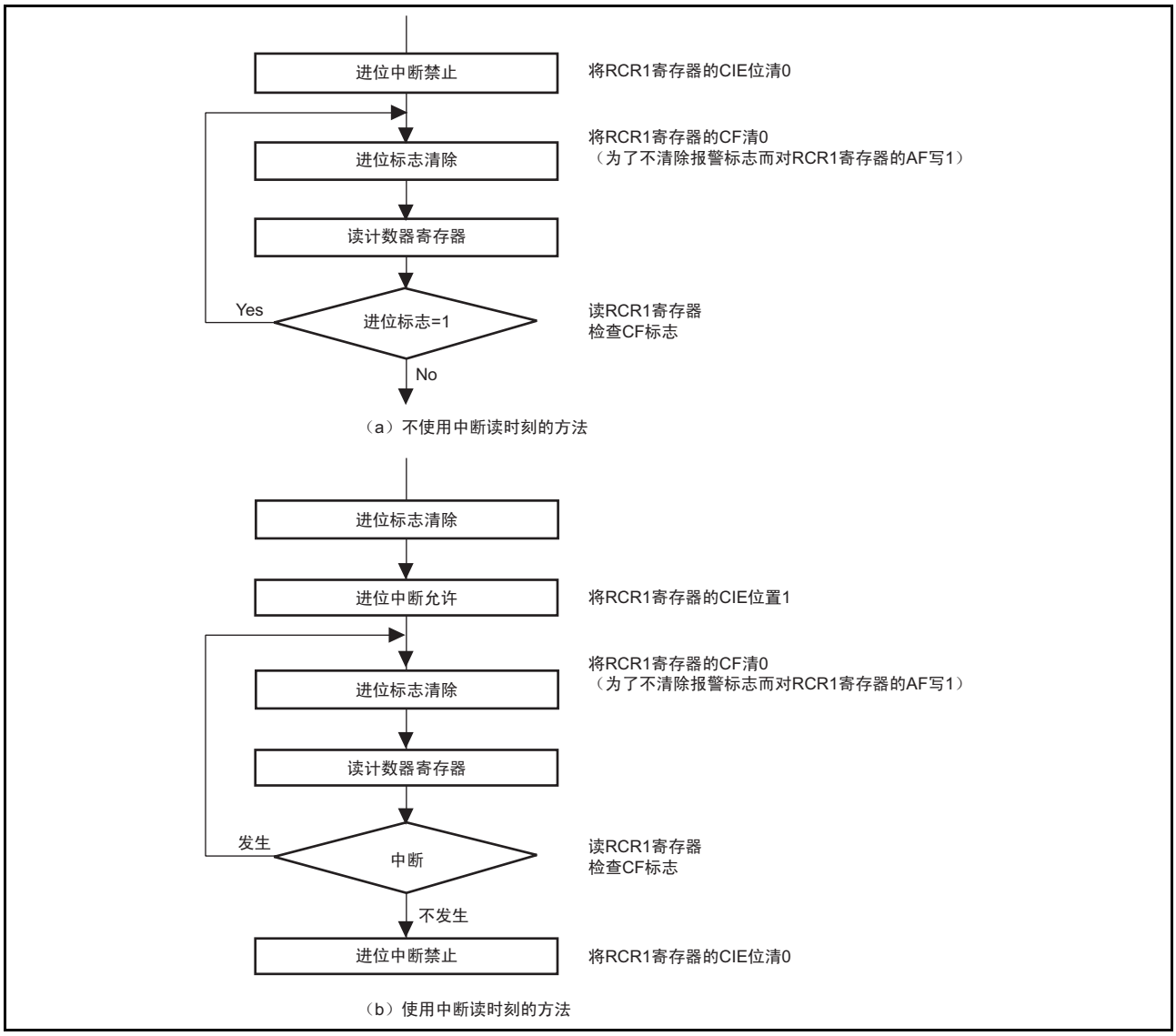


图 14.3　时间的读出步骤

如果在读取时间的过程中产生进位，就会无法得到正确的时间，因此必须重新读取。不使用中断的方法如图 14.3 （a），使用进位中断的方法如图 14.3 （b）所示。通常为了使程序简便，利用不使用中断的方法。

14.4.4　报警功能

报警功能的使用例如图 14.4 所示。

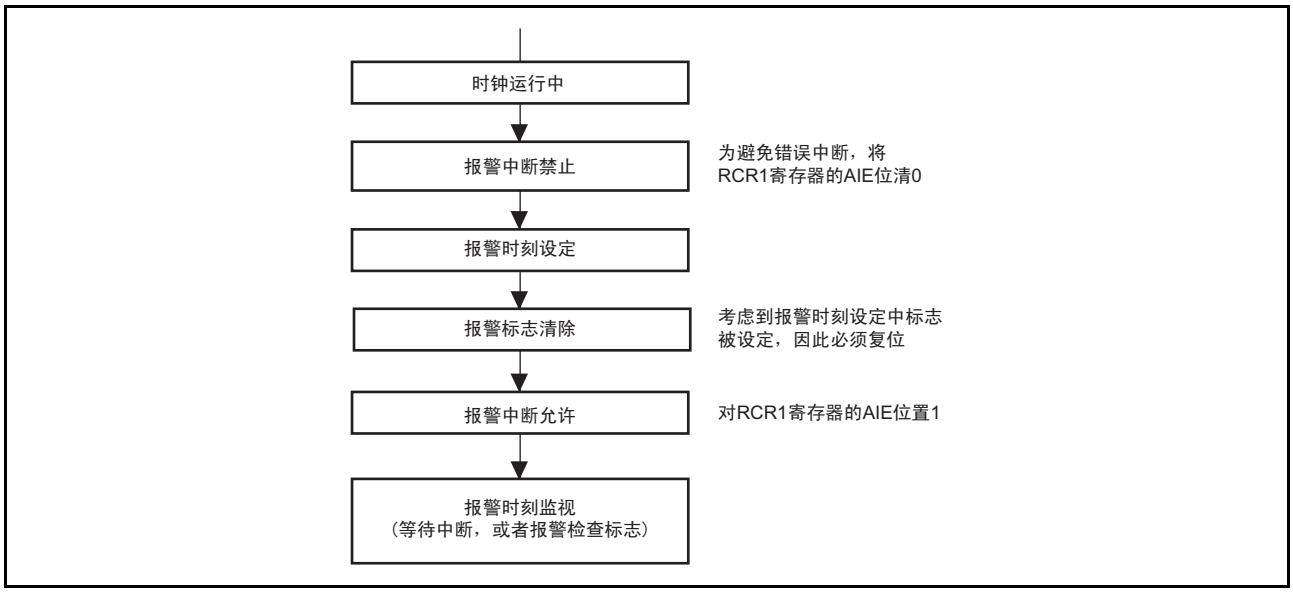


图 14.4　报警功能的使用方法

能通过秒、分、小时、日、星期、月、年中任意一个，或者组合发生报警。对报警对象的报警寄存器的 ENB 位写 1，并且对低位设定报警时间。对不是报警对象的寄存器的 ENB 位写 0。

计数器与报警时间匹配时，将 RCR1 寄存器的 AF 位置 1。能通过读取此位确认报警检测，但通常通过中断进行。对 RCR1 寄存器的 AIE 位写 1 时，能产生报警中断并能检测报警。

计数器与报警时间匹配时，将报警标志置位。通过写 0，清除报警标志。

14.5 使用时的注意事项

14.5.1 关于 RTC 计数运行时的寄存器写入

RTC 计数运行时 （RCR2 寄存器的 START 位 =1 时），不能对以下寄存器进行写入。

RSECCNT、 RMINCNT、 RHRCNT、 RDAYCNT、 RWKCNT、 RMONCNT、 RYRCONT

对上述寄存器进行写时，请在停止 RTC 的计数运行后进行。

14.5.2 关于实时时钟 （RTC）周期中断的使用

周期中断的使用方法如图 14.5 所示。

能通过 RCR2 寄存器的 PES[2:0] 位设定的周期定期产生周期中断。经过 PES[2:0] 位设定的时间后，将 PEF 置 1。

PEF 在设定 PES[2:0] 位及产生周期中断时被清 0。虽然能通过读取此位确认周期中断的产生，但是通常使用中断功能。

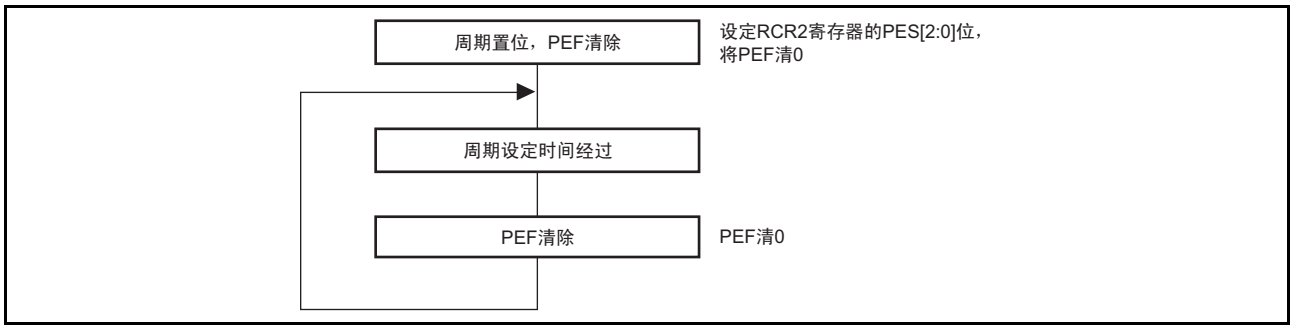


图 14.5 周期中断功能的使用方法

14.5.3 关于寄存器设定后的待机转移

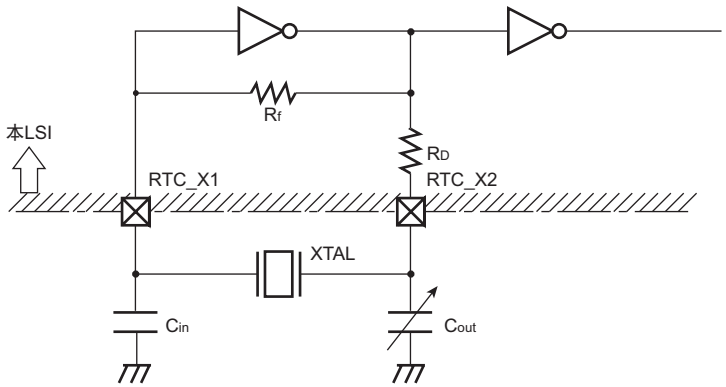
设定 RTC 内的寄存器后转移到待机状态时，可能出现不能正确计数的情况。必须在寄存器设定后等待不低于 2RTC 时钟之后，再转移到待机状态。

14.5.4　RTC 用晶体振荡电路

RTC 用晶体振荡电路的各常数 （推荐值）如**表 14.3**， RTC 用晶体振荡电路如**图 14.6** 所示。

表 14.3　推荐振荡电路的常数 （推荐值）

f_{osc}	C_{in}	C_{out}
32.768kHz	10 ～ 22pF	10 ～ 22pF



【注】 1. 按照调整范围和稳定性等要求决定用于频率调整的可变电容是在Cin端还是在Cout端。
2. 内置的电阻值Rf（Typ值）=10MΩ，RD（Typ值）=400kΩ
3. Cin与Cout 的值包含由于布线产生的寄生电容值。使用接地层电路板时请注意。
4. 由于晶体振荡的振荡稳定时间因安装电路常数、寄生电容等而不同，请在与晶体振荡器制造商充分咨询后再决定。
5. 请尽量将晶体振荡器及负载电容Cin、Cout配置在LSI附近，并尽可能的缩短布线。振荡电路附近不能通过信号线。
（在RTC_X1、RTC_X2引脚上有外部噪声时，就有可能不能正确振
6. 荡。）
晶体振荡器连接引脚（RTC_X1、RTC_X2）布线时请尽量远离除GND以
7. 外的电源线或者信号线。
不使用RTC用水晶振荡电路时，请将RTC_X1引脚连接到GND，并将RTC_X2引脚置为开路。

图 14.6　RTC 用晶体振荡电路连接例

第 15 章　带 FIFO 的串行通信接口 （SCIF）

本 LSI 内置支持异步通信和时钟同步通信 2 种方式的 4 通道带 FIFO 的串行通信接口 （SCIF）。同时，各通道还内置用于独立发送 / 接收的 16 段 FIFO 寄存器，可提高本 LSI 的效率及实现高速连续通信。

15.1　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 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
  断点的检测　　：当发生帧错误并且继续有至少 1 帧长的空间 0 （低电平）时，就检测到断点；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断点。
- 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，所以能同时进行发送和接收，而且发送部和接收部都是 16 段 FIFO 缓冲结构，所以能高速连续发送和连续接收串行数据。
- 可通过内置波特率发生器选择任意的位速率
- 内部或者外部的发送 / 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 数据空中断、断点中断、接收 FIFO 数据满中断以及接收错误中断的 4 种中断，能分别独立请求中断。
- 不使用 SCIF 时，为降低功耗可停止给 SCIF 供给时钟并使其运行停止。
- 在异步模式中，内置 （仅通道 3）调制解调器控制功能 （$\overline{\text{RTS}}$ 及 $\overline{\text{CTS}}$）。
- 能检测到发送和接收 FIFO 数据寄存器的数据个数以及接收 FIFO 数据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）。
- 在异步模式中，可选择 16/8 倍位速率的基本时钟运行
- 在异步模式中，内部时钟被选择为时钟源以及 SCK 引脚用作输入引脚时，能选择波特率发生器的普通 / 倍速模式。

SCIF 的框图如图 15.1 所示。

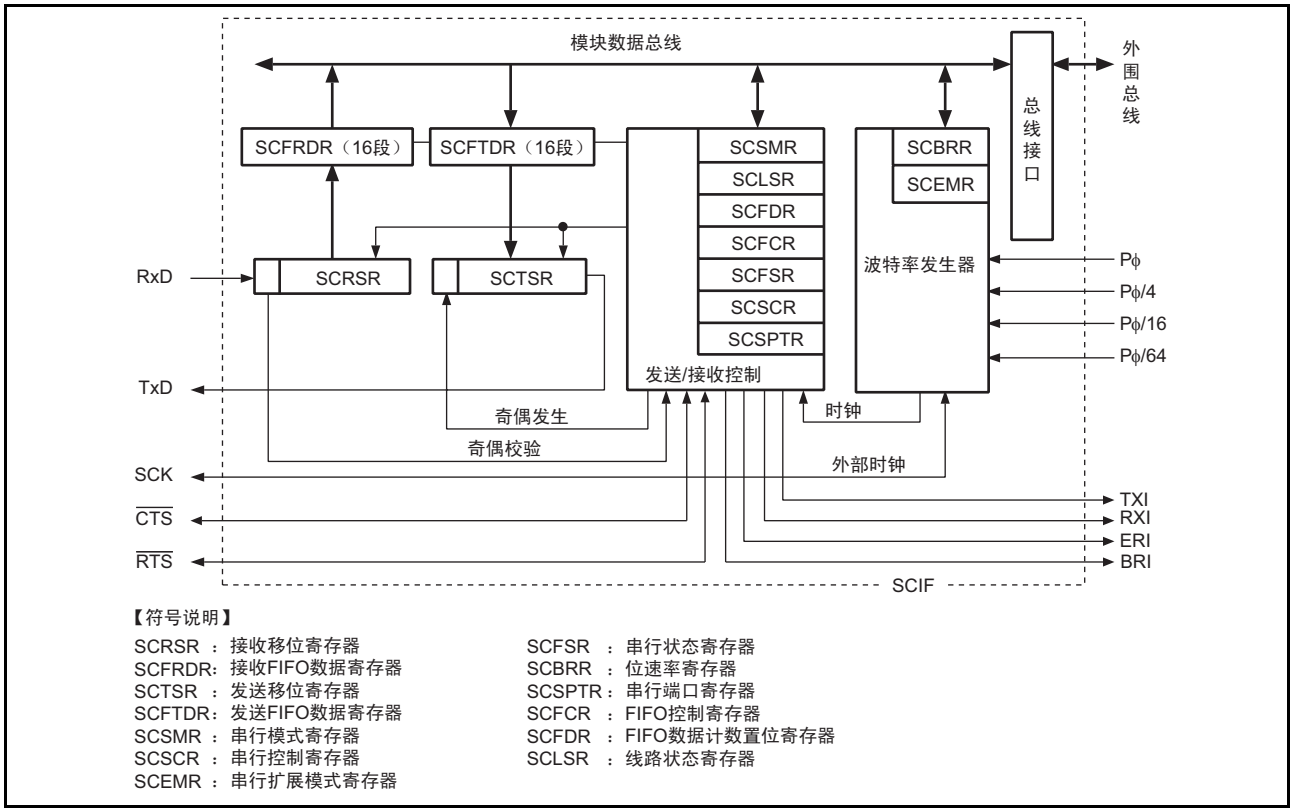


图 15.1　SCIF 的框图

15.2　输入 / 输出引脚

SCIF 的引脚构成如表 15.1 所示。

表 15.1　引脚构成

通道	名称	引脚名称	输入 / 输出	功能
0 ～ 3	串行时钟引脚	SCK0 ～ SCK3	输入 / 输出	输入 / 输出时钟
	接收数据引脚	RxD0 ～ RxD3	输入	输入接收数据
	发送数据引脚	TxD0 ～ TxD3	输出	输出发送数据
3	请求发送引脚	$\overline{\text{RTS3}}$	输入 / 输出	请求发送
	清除发送引脚	$\overline{\text{CTS3}}$	输入 / 输出	清除发送

15.3　寄存器说明

SCIF 有以下寄存器。

表 15.2　寄存器构成

通道	寄存器名	简称	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'0000	H'FFFE8000	16
	位速率寄存器 _0	SCBRR_0	R/W	H'FF	H'FFFE8004	8
	串行控制寄存器 _0	SCSCR_0	R/W	H'0000	H'FFFE8008	16
	发送 FIFO 数据寄存器 _0	SCFTDR_0	W	不定	H'FFFE800C	8
	串行状态寄存器 _0	SCFSR_0	R/(W)*1	H'0060	H'FFFE8010	16
	接收 FIFO 数据寄存器 _0	SCFRDR_0	R	不定	H'FFFE8014	8
	FIFO 控制寄存器 _0	SCFCR_0	R/W	H'0000	H'FFFE8018	16
	FIFO 数据计数置位寄存器 _0	SCFDR_0	R	H'0000	H'FFFE801C	16
	串行端口寄存器 _0	SCSPTR_0	R/W	H'0050	H'FFFE8020	16
	线路状态寄存器 _0	SCLSR_0	R/(W)*2	H'0000	H'FFFE8024	16
	串行扩展模式寄存器 _0	SCEMR_0	R/W	H'0000	H'FFFE8028	16
1	串行模式寄存器 _1	SCSMR_1	R/W	H'0000	H'FFFE8800	16
	位速率寄存器 _1	SCBRR_1	R/W	H'FF	H'FFFE8804	8
	串行控制寄存器 _1	SCSCR_1	R/W	H'0000	H'FFFE8808	16
	发送 FIFO 数据寄存器 _1	SCFTDR_1	W	不定	H'FFFE880C	8
	串行状态寄存器 _1	SCFSR_1	R/(W)*1	H'0060	H'FFFE8810	16
	接收 FIFO 数据寄存器 _1	SCFRDR_1	R	不定	H'FFFE8814	8
	FIFO 控制寄存器 _1	SCFCR_1	R/W	H'0000	H'FFFE8818	16
	FIFO 数据计数置位寄存器 _1	SCFDR_1	R	H'0000	H'FFFE881C	16
	串行端口寄存器 _1	SCSPTR_1	R/W	H'0050	H'FFFE8820	16
	线路状态寄存器 _1	SCLSR_1	R/(W)*2	H'0000	H'FFFE8824	16
	串行扩展模式寄存器 _1	SCEMR_1	R/W	H'0000	H'FFFE8828	16
2	串行模式寄存器 _2	SCSMR_2	R/W	H'0000	H'FFFE9000	16
	位速率寄存器 _2	SCBRR_2	R/W	H'FF	H'FFFE9004	8
	串行控制寄存器 _2	SCSCR_2	R/W	H'0000	H'FFFE9008	16
	发送 FIFO 数据寄存器 _2	SCFTDR_2	W	不定	H'FFFE900C	8
	串行状态寄存器 _2	SCFSR_2	R/(W)*1	H'0060	H'FFFE9010	16
	接收 FIFO 数据寄存器 _2	SCFRDR_2	R	不定	H'FFFE9014	8
	FIFO 控制寄存器 _2	SCFCR_2	R/W	H'0000	H'FFFE9018	16
	FIFO 数据计数置位寄存器 _2	SCFDR_2	R	H'0000	H'FFFE901C	16
	串行端口寄存器 _2	SCSPTR_2	R/W	H'0050	H'FFFE9020	16
	线路状态寄存器 _2	SCLSR_2	R/(W)*2	H'0000	H'FFFE9024	16
	串行扩展模式寄存器 _2	SCEMR_2	R/W	H'0000	H'FFFE9028	16

通道	寄存器名	简称	R/W	初始值	地址	存取长度
3	串行模式寄存器 _3	SCSMR_3	R/W	H'0000	H'FFFE9800	16
	位速率寄存器 _3	SCBRR_3	R/W	H'FF	H'FFFE9804	8
	串行控制寄存器 _3	SCSCR_3	R/W	H'0000	H'FFFE9808	16
	发送 FIFO 数据寄存器 _3	SCFTDR_3	W	不定	H'FFFE980C	8
	串行状态寄存器 _3	SCFSR_3	R/(W)*1	H'0060	H'FFFE9810	16
	接收 FIFO 数据寄存器 _3	SCFRDR_3	R	不定	H'FFFE9814	8
	FIFO 控制寄存器 _3	SCFCR_3	R/W	H'0000	H'FFFE9818	16
	FIFO 数据计数置位寄存器 _3	SCFDR_3	R	H'0000	H'FFFE981C	16
	串行端口寄存器 _3	SCSPTR_3	R/W	H'0050	H'FFFE9820	16
	线路状态寄存器 _3	SCLSR_3	R/(W)*2	H'0000	H'FFFE9824	16
	串行扩展模式寄存器 _3	SCEMR_3	R/W	H'0000	H'FFFE9828	16

【注】 *1　为了清除标志只可写 0。 bit15 ～ 8、 3、 2 为只读，不能写。

*2　为了清除标志只可写 0。 bit15 ～ 1 为只读，不能写。

15.3.1　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。当接收完 1 字节的数据时，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：	-	-	-	-	-	-	-	-

15.3.2　接收 FIFO 数据寄存器 （SCFRDR）

SCFRDR 是保存已接收串行数据的 16 段 FIFO 寄存器。当接收完 1 字节的串行数据时，SCIF 就将已接收的串行数据从接收移位寄存器 （SCRSR）传送到 SCFRDR 并进行保存，然后结束接收操作。在结束 16 字节的保存前可连续接收， CPU 能读但不能写入 SCFRDR。如果在接收 FIFO 数据寄存器中没有接收数据的状态下读数据，则读的值不定。

如果 SCFRDR 接收数据满，其后接收的串行数据就会丢失。

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

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 FIFO 数据寄存器 （SCFTDR）传送到 SCTSR，SCIF 就通过从 LSB （位 0）按顺序将数据发送到 TxD 引脚来发送串行数据。当发送完 1 字节的数据时，下一个发送数据就自动从 SCFTDR 传送到 SCTSR，并开始发送。

CPU 不能直接读写 SCTSR。

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

15.3.4　发送 FIFO 数据寄存器 （SCFTDR）

SCFTDR 是保存串行发送数据的 16 段 FIFO 寄存器。如果检测到发送移位寄存器（SCTSR）为空，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 发送数据为空。CPU 总是能写入 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

15.3.5　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寄存器。

CPU 总是能读写 SCSMR。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	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	位名称	初始值	R/W	说明
15 ～ 8	—	均为 0	R	保留位 读写的值总是 0。
7	C/$\overline{A}$	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 /8 位数据。在时钟同步模式下，数据长固定为 8 位数据，与 CHR 的设定无关。 0：8 位数据 1：7 位数据 * 【注】 * 如果选择 7 位数据，发送 FIFO 数据寄存器的 MSB （bit7）就不发送。

位	位名称	初始值	R/W	说明
5	PE	0	R/W	允许奇偶校验 在异步模式下，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。在时钟同步模式下，不附加也不检查奇偶校验位，与 PE 位的设定无关。 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 * 【注】 * 如果将 PE 位置 1，就在发送时将 O/$\overline{E}$ 位所指定的偶校验位或者奇校验位附加给发送数据后进行发送，而在接收时检查已接收的奇偶校验位是否为 O/$\overline{E}$ 位所指定的偶校验位或者奇校验位。
4	O/$\overline{E}$	0	R/W	奇偶校验模式 选择是通过偶校验还是通过奇校验来附加或检查奇偶校验位。只在异步模式下将 PE 位置 1 并且允许附加或检查奇偶校验位时，O/$\overline{E}$ 位的设定才有效；在时钟同步模式或者异步模式下禁止附加或检查奇偶校验位时， O/$\overline{E}$ 位的指定无效。 0：偶校验 *1 1：奇校验 *2 【注】*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。
3	STOP	0	R/W	停止位长 异步模式下的停止位长可选择 1 位 /2 位。STOP 位的设定只在异步模式下有效。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	选择时钟 选择内置波特率发生器的内部时钟源。 有关时钟源、位速率寄存器的设定值及波特率的关系，请参照 “15.3.8 位速率寄存器（SCBRR）”。 00：Pϕ 时钟 01：Pϕ/4 时钟 10：Pϕ/16 时钟 11：Pϕ/64 时钟 【注】 Pϕ：外围时钟

15.3.6　串行控制寄存器 （SCSCR）

SCSCR 是进行 SCIF 的发送 / 接收、允许 / 禁止中断请求以及选择发送 / 接收时钟源的寄存器。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	位名称	初始值	R/W	说明
15 ～ 8	—	均为 0	R	保留位 读写的值总是 0。
7	TIE	0	R/W	允许发送中断 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 / 禁止发生 FIFO 数据空的中断 （TXIF）请求。 0：禁止发送 FIFO 数据空的中断 （TXI）请求 1：允许发送 FIFO 数据空的中断 （TXI）请求 * 【注】* 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 TXI。
6	RIE	0	R/W	允许接收中断 允许 / 禁止以下的中断请求： 1. 串行状态寄存器（SCFSR）的 RDF 标志或者 DR 标志被置 1 时的接收 FIFO 数据满的中断 （RXI）请求。 2. SCFSR 的 ER 标志被置 1 时的接收错误中断 （ERI）请求。 3. SCFSR 的 BRK 标志或者线路状态寄存器 （SCLSR）的 ORER 标志被置 1 时的断点中断 （BRI）请求。 0：禁止接收 FIFO 数据满中断 （RXI）请求、接收错误中断 （ERI）请求和断点中断 （BRI）请求 1：允许 * 接收 FIFO 数据满中断 （RXI）请求、接收错误中断 （ERI）请求和断点中断 （BRI）请求 【注】* 通过在读 DR 标志或者 RDF 标志的 1 后将该位清 0、或者通过将 RIE 位清 0 来解除 RXI 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 ERI、 BRI 中断。
5	TE	0	R/W	允许发送 允许 / 禁止串行发送。 0：禁止发送 1：允许发送 * 【注】* 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，设定发送格式以及对发送 FIFO 进行复位。

位	位名称	初始值	R/W	说明
4	RE	0	R/W	允许接收 允许 / 禁止串行接收。 0：禁止接收 *1 1：允许接收 *2 【注】*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，如果在异步模式时检测到起始位以及在时钟同步模式下检测到同步时钟，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），设定接收格式以及对接收 FIFO 进行复位。
3	REIE	0	R/W	允许接收错误中断 允许 / 禁止接收错误的中断 （ERI）请求和断点中断 （BRI）请求。但是 REIE 位的设定只在 RIE 位是 0 时有效。 0：禁止接收错误的中断 （ERI）请求和断点中断 （BRI）请求 1：允许接收错误的中断 （ERI）请求和断点中断 （BRI）请求 * 【注】* 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 ERI、BRI 中断请求。如果 REIE 位被置 1，即使将 RIE 位设定为 0，也发生 ERI 中断或者 BRI 中断。在 DMAC 传送时，需要向中断控制器通知 ERI、 BRI 中断请求时进行设定。
2	—	0	R	保留位 读写的值总是 0。
1、0	CKE[1:0]	00	R/W	允许时钟 选择 SCIF 时钟源以及禁止 / 允许 SCK 引脚的输出时钟。通过 CKE[1:0] 位将 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。在时钟同步模式下设定同步时钟输出时，请在 SCSMR 的 C/$\overline{A}$ 位置 1 后再设定 CKE[1:0]。 • 异步模式 00：内部时钟 /SCK 引脚为输入引脚 （忽视输入信号） 01：内部时钟 /SCK 引脚为时钟输出引脚 （输出频率是位速率的 16/8 倍的时钟） 10：外部时钟 /SCK 引脚为时钟输入引脚 （输入频率是位速率的 16/8 倍的时钟） 11：禁止设定 • 时钟同步模式 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：禁止设定

15.3.7　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 运行状态的状态标志。

CPU 总是对 SCFSR 进行读写，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。 PER （bit15 ～ 12、 2）标志和 FER （bit11 ～ 8、 3）标志为只读标志，不能写。

位：	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】* 为清除标志，只能在读1后，写0。

位	位名称	初始值	R/W	说明
15 ～ 12	PER[3:0]	0000	R	奇偶校验错误数 表示保存到接收 FIFO 数据寄存器 （SCFRDR）的接收数据中发生奇偶校验错误的数据个数。 将 SCFSR 的 ER 位置位后， bit15 ～ 12 所示的值表示发生奇偶校验错误的数据个数。SCFRDR 的 16 字节接收数据均发生奇偶校验错误时，PER[3:0] 就表示 0000。
11 ～ 8	FER[3:0]	0000	R	帧错误数 表示保存到接收 FIFO 数据寄存器 （SCFRDR）的接收数据中发生帧错误的数据个数。 将SCFSR的ER位置位后，bit11～8所示的值表示发生帧错误的数据个数。SCFRDR 的 16 字节接收数据均发生帧错误时， FER[3:0] 就表示 0000。
7	ER	0	R/(W)*	接收错误 表示发生了帧错误或者在接收含奇偶性的数据时发生了奇偶校验错误。 *1 0：表示正在接收或者已正常接收完 ［清除条件］ • 上电复位 • 在读 ER=1 的状态后写入 0 时 1：表示在接收时发生了帧错误或者奇偶校验错误 ［置位条件］ • 在 1 次数据接收完时，检查接收数据的最后停止位是否为 1，且确定最后停止位为 0 时 *2 • 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/$\overline{E}$ 位所指定的偶 / 奇校验的设定不匹配时 【注】 *1 SCSCR 的 RE 位清 0 时， ER 标志不受影响继续保持原来的状态。即使发生接收错误，还是将接收数据传送到 SCFRDR 并继续接收运行。能通过 SCFSR 的 FER 位或者 PER 位来判断从 SCFRDR 读取的数据中是否有接收错误。 *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。

位	位名称	初始值	R/W	说明
6	TEND	1	R/(W)*	发送结束 表示在发送字符的最后一位时 SCFTDR 中无有效数据而终止发送。 0：表示正在发送 ［清除条件］ • 当给 SCFTDR 写发送数据、读取 TEND=1 的状态后给 TEND 标志写 0 时 *1 1：表示发送已结束 ［置位条件］ • 上电复位 • 当 SCSCR 的 TE 位是 0 时 • 当发送 1 字节串行发送字符的最后一位时， SCFTDR 中无发送数据时 【注】*1　DMAC 通过 TXI 中断请求将数据写到 SCFTDR 时，请勿用作发送结束标志。
5	TDFE	1	R/(W)*	发送 FIFO 数据空 表示将数据从发送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 TTRG[1:0] 所指定的发送触发数据个数、以及允许能将发送数据写入 SCFTDR。 0：表示写入 SCFTDR 的发送数据个数大于指定的发送触发个数 ［清除条件］ • 当读取 TDFE=1 的状态后，将大于指定发送触发个数的数据写入 SCFTDR 并将 0 写入 TDFE 时 • 通过发送 FIFO 数据空中断 （TXI）启动 DMAC，将大于指定发送触发个数的数据写入 SCFTDR 时 1：表示写入 SCFTDR 的发送数据个数小于等于指定发送触发个数 *1 ［置位条件］ • 上电复位 • 当写入发送结果 SCFTDR 的发送数据个数小于等于指定发送触发个数时 【注】*1　因为 SCFTDR 是 16 字节 FIFO 寄存器，所以在 TDFE=1 的状态时，能写入的最大数据个数［为 16 减去指定发送触发个数之后的数］。即使想写入更多的数据，此数据仍被忽视。 SCFDR 的高 8 位表示 SCFTDR 的数据个数。
4	BRK	0	R/(W)*	断点的检测 表示在接收数据中已检测到断点信号。 0：无断点信号 ［清除条件］ • 上电复位 • 读 BRK=1 的状态后给 BRK 标志写 0 时 1：接收到断点信号 *1 ［置位条件］ • 接收含有帧错误的数据，且继续有大于等于 1 帧长的空间 0 （低电平）时 【注】*1　检测到断点时，在检测后停止向 SCFRDR 传送接收数据 （H'00）。当断点结束并且接收信号变为标记 1 时，重新开始传送接收数据。

位	位名称	初始值	R/W	说明
3	FER	0	R	表示帧错误 表示在异步模式下，从接收 FIFO 数据寄存器 （SCFRDR）读取的数据中是否有帧错误。 0：表示下一次从 SCFRDR 读取的接收数据中未发生帧错误 ［清除条件］ • 上电复位 • 下一次从 SCFRDR 读取的数据无帧错误 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］ • 下一个 SCFRDR 读数据中有帧错误
2	PER	0	R	表示奇偶校验错误 表示在异步模式中，从接收 FIFO 数据寄存器 （SCFRDR）读取的数据是否有奇偶校验错误。 0：表示下一次从 SCFRDR 读取的接收数据未发生奇偶校验错误 ［清除条件］ • 上电复位 • 下一个 SCFRDR 读数据无奇偶校验错误 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］ • 下一个 SCFRDR 读数据有奇偶校验错误
1	RDF	0	R/(W)*	接收 FIFO 数据满 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 RTRG[1:0] 位所指定的接收触发个数。 0：表示写入 SCFRDR 的接收数据个数小于指定的接收触发个数 ［清除条件］ • 上电复位 • 在读取 RDF=1 后，读 SCFRDR 直到 SCFRDR 的接收数据个数小于指定接收触发个数，然后给 RDF 写 0 时 • 通过接收 FIFO 数据满中断（RXI）启动 DMAC，读 SCFRDR 直到 SCFRDR 的接收数据个数小于指定接收触发个数时 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］ • 当将大于等于指定接收触发个数的接收数据个数保存在 SCFRDR 时 *1 【注】*1　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接收触发个数。在读取 SCFRDR 的全部数据后，如果继续读数据，则值不定。SCFDR 的低 8 位表示 SCFRDR 的接收数据个数

位	位名称	初始值	R/W	说明
0	DR	0	R/(W)*	接收数据就绪 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）并且经过 15ETU 的时间后还未从最后的停止位接收到下一个数据。在时钟同步模式中，此位不置位。 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］ • 上电复位 • 读取 DR=1 的状态后、已读取 SCFRDR 的全部接收数据并且写 0 时 • 通过接收 FIFO 数据满中断（RXI）启动 DMAC，读出 SCFRDR 的全部接收数据时 1：表示没有接收到下一个接收数据 ［置位条件］ • 当将小于指定接收触发个数的数据保存到 SCFRDR 并且经过 15ETU 的时间 *1 后还没有从最后的停止位接收到下一个数据时 【注】 *1　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）

【注】 *　为清除标志，只能在读 1 后写 0。

15.3.8　位速率寄存器 （SCBRR）

SCBRR 是 8 位寄存器，结合串行模式寄存器 （SCSMR）的 CKS[1:0] 与串行扩展模式寄存器 （SCEMR）的 BGDM 位及 ABCS 位，设定串行发送和接收的位速率。

CPU 随时能对 SCBRR 进行读写。上电复位时，SCBRR 被初始化为 H'FF。另外，因为每个通道的波特率发生器的控制是独立的，因此能分别对 3 个通道设定不同的值。

位：	7	6	5	4	3	2	1	0
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/W	R/W

通过以下公式求 SCBRR 的设定值：

【异步模式】

- 波特率发生器普通模式 （SCEMR 的 BGDM=0）

  $N = (P\phi / (64 \times 2^{2n-1} \times B)) \times 10^6 - 1$

  （以频率为位速率的 16 倍的基本时钟运行时）

  $N = (P\phi / (32 \times 2^{2n-1} \times B)) \times 10^6 - 1$

  （以频率为位速率的 8 倍的基本时钟运行时）
- 波特率发生器倍速模式 （SCEMR 的 BGDM=1）

  $N = (P\phi / (32 \times 2^{2n-1} \times B)) \times 10^6 - 1$

  （以频率为位速率的 16 倍的基本时钟运行时）

  $N = (P\phi / (16 \times 2^{2n-1} \times B)) \times 10^6 - 1$

  （以频率为位速率的 8 倍的基本时钟运行时）

【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 SCBRR 的设定值 （$0 \leqslant N \leqslant 255$）

（请设定为满足电特性的值。）

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

n	时钟	SCSMR 的设定值	
		CKS[1]	CKS[0]
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

通过以下公式求异步模式的位速率误差：

- 波特率发生器普通模式 （SCEMR的BGDM=0）
  误差 （%） = {（（Pϕ×10^6） / （（N+1） ×B×64×2^{2n-1}））−1} ×100
  （以频率为位速率的16倍的基本时钟运行时）
  误差 （%） = {（（Pϕ×10^6） / （（N+1） ×B×32×2^{2n-1}））−1} ×100
  （以频率为位速率的8倍的基本时钟运行时）
- 波特率发生器倍速模式 （SCEMR的BGDM=1）
  误差 （%） = {（（Pϕ×10^6） / （（N+1） ×B×32×2^{2n-1}））−1} ×100
  （以频率为位速率的16倍的基本时钟运行时）
  误差 （%） = {（（Pϕ×10^6） / （（N+1） ×B×16×2^{2n-1}））−1} ×100
  （以频率为位速率的8倍的基本时钟运行时）

当波特率发生器普通模式 （SCEMR 的 BGDM=0）、 16 倍的基本时钟运行 （SCEMR 的 ABCS=0）时，异步模式 SCBRR 的设定例子**表 15.4** 所示。另外，时钟同步模式 SCBRR 的设定例子如**表 15.5** 所示。

表 15.4　位速率的 SCBRR 设定例子 〔异步模式、 BGDM=0、 ABCS=0〕（1）

位速率 （bit/s）	Pϕ （MHz）											
	8			9.8304			10			12		
	n	N	误差 （%）	n	N	误差 （%）	n	N	误差 （%）	n	N	误差 （%）
110	2	141	0.03	2	174	–0.26	2	177	–0.25	2	212	0.03
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16
9600	0	25	0.16	0	31	0.00	0	32	–1.36	0	38	0.16
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	0.16
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11	0.00
38400	0	6	–6.99	0	7	0.00	0	7	1.73	0	9	–2.34

表 15.4　位速率的 SCBRR 设定例子 〔异步模式、 BGDM=0、 ABCS=0〕（2）

位速率 （bit/s）	Pϕ （MHz）											
	12.288			14.7456			16			19.6608		
	n	N	误差 （%）	n	N	误差 （%）	n	N	误差 （%）	n	N	误差 （%）
110	2	217	0.08	3	64	0.70	3	70	0.03	3	86	0.31
150	2	159	0.00	2	191	0.00	2	207	0.16	2	255	0.00
300	2	79	0.00	2	95	0.00	2	103	0.16	2	127	0.00
600	1	159	0.00	1	191	0.00	1	207	0.16	1	255	0.00
1200	1	79	0.00	1	95	0.00	1	103	0.16	1	127	0.00
2400	0	159	0.00	0	191	0.00	0	207	0.16	0	255	0.00
4800	0	79	0.00	0	95	0.00	0	103	0.16	0	127	0.00
9600	0	39	0.00	0	47	0.00	0	51	0.16	0	63	0.00
19200	0	19	0.00	0	23	0.00	0	25	0.16	0	31	0.00
31250	0	11	2.40	0	14	–1.70	0	15	0.00	0	19	–1.70
38400	0	9	0.00	0	11	0.00	0	12	0.16	0	15	0.00

表 15.4　位速率的 SCBRR 设定例子 〔异步模式、 BGDM=0、 ABCS=0〕（3）

位速率（bit/s）	Pϕ （MHz）											
	20			24			24.576			28.7		
	n	N	误差 （%）	n	N	误差 （%）	n	N	误差 （%）	n	N	误差 （%）
110	3	88	–0.25	3	106	–0.44	3	108	0.08	3	126	0.31
150	3	64	0.16	3	77	0.16	3	79	0.00	3	92	0.46
300	2	129	0.16	2	155	0.16	2	159	0.00	2	186	–0.08
600	2	64	0.16	2	77	0.16	2	79	0.00	2	92	0.46
1200	1	129	0.16	1	155	0.16	1	159	0.00	1	186	–0.08
2400	1	64	0.16	1	77	0.16	1	79	0.00	1	92	0.46
4800	0	129	0.16	0	155	0.16	0	159	0.00	0	186	–0.08
9600	0	64	0.16	0	77	0.16	0	79	0.00	0	92	0.46
19200	0	32	–1.36	0	38	0.16	0	39	0.00	0	46	–0.61
31250	0	19	0.00	0	23	0.00	0	24	–1.70	0	28	–1.03
38400	0	15	1.73	0	19	–2.34	0	19	0.00	0	22	1.55

表 15.4　位速率的 SCBRR 设定例子 〔异步模式、 BGDM=0、 ABCS=0〕（4）

位速率（bit/s）	Pϕ （MHz）					
	30			33		
	n	N	误差 （%）	n	N	误差 （%）
110	3	132	0.13	3	145	0.33
150	3	97	–0.35	3	106	0.39
300	2	194	0.16	2	214	–0.07
600	2	97	–0.35	2	106	0.39
1200	1	194	0.16	1	214	–0.07
2400	1	97	–0.35	1	106	0.39
4800	0	194	–1.36	0	214	–0.07
9600	0	97	–0.35	0	106	0.39
19200	0	48	–0.35	0	53	–0.54
31250	0	29	0.00	0	32	0.00
38400	0	23	1.73	0	26	–0.54

【注】　请尽量将误差设定在 1% 以内。

表 15.5　位速率的 SCBRR 设定例子 〔时钟同步模式〕

位速率（bit/s）	Pϕ（MHz）									
	8		16		28.7		30		33	
	n	N	n	N	n	N	n	N	n	N
110	—	—	—	—	—	—	—	—	—	—
250	3	124	3	249	—	—	—	—	—	—
500	2	249	3	124	3	223	3	233	3	255
1k	2	124	2	249	3	111	3	116	3	125
2.5k	1	199	2	99	2	178	2	187	2	200
5k	1	99	1	199	2	89	2	93	2	100
10k	0	199	1	99	1	178	1	187	1	200
25k	0	79	0	159	1	71	1	74	1	80
50k	0	39	0	79	0	143	0	149	0	160
100k	0	19	0	39	0	71	0	74	0	80
250k	0	7	0	15	—	—	0	29	0	31
500k	0	3	0	7	—	—	0	14	0	15
1M	0	1	0	3	—	—	—	—	0	7
2M	0	0*	0	1	—	—	—	—	—	—

【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 / 接收。

使用波特率发生器时异步模式各频率的最大位速率如表 15.6 所示。另外，输入外部时钟时异步模式的最大位速率如表 15.7 所示、输入 （t_{Scyc}=12t_{pcyc} 时 *）外部时钟时时钟同步模式的最大位速率如表 15.8 所示。

【注】 * 请确认满足本 LSI 与通信对象的电特性。

表 15.6　使用波特率发生器时各频率的最大位速率 （异步模式）

Pϕ （MHz）	设定值				最大位速率 （bit/s）
	BGDM	ABCS	n	N	
8	0	0	0	0	250000
		1	0	0	500000
	1	0	0	0	500000
		1	0	0	1000000
9.8304	0	0	0	0	307200
		1	0	0	614400
	1	0	0	0	614400
		1	0	0	1228800
12	0	0	0	0	375000
		1	0	0	750000
	1	0	0	0	750000
		1	0	0	1500000
14.7456	0	0	0	0	460800
		1	0	0	921600
	1	0	0	0	921600
		1	0	0	1843200
16	0	0	0	0	500000
		1	0	0	1000000
	1	0	0	0	1000000
		1	0	0	2000000
19.6608	0	0	0	0	614400
		1	0	0	1228800
	1	0	0	0	1228800
		1	0	0	2457600
20	0	0	0	0	625000
		1	0	0	1250000
	1	0	0	0	1250000
		1	0	0	2500000
24	0	0	0	0	750000
		1	0	0	1500000
	1	0	0	0	1500000
		1	0	0	3000000

Pϕ （MHz）	设定值				最大位速率 （bit/s）
	BGDM	ABCS	n	N	
24.576	0	0	0	0	768000
		1	0	0	1536000
	1	0	0	0	1536000
		1	0	0	3072000
28.7	0	0	0	0	896875
		1	0	0	1793750
	1	0	0	0	1793750
		1	0	0	3587500
30	0	0	0	0	937500
		1	0	0	1875000
	1	0	0	0	1875000
		1	0	0	3750000
33	0	0	0	0	1031250
		1	0	0	2062500
	1	0	0	0	2062500
		1	0	0	4125000

表 15.7　输入外部时钟时的最大位速率 （异步模式）

Pϕ （MHz）	外部输入时钟 （MHz）	设定值 ABCS	最大位速率 （bit/s）
8	2.0000	0	125000
		1	250000
9.8304	2.4576	0	153600
		1	307200
12	3.0000	0	187500
		1	375000
14.7456	3.6864	0	230400
		1	460800
16	4.0000	0	250000
		1	500000
19.6608	4.9152	0	307200
		1	614400
20	5.0000	0	312500
		1	625000
24	6.0000	0	375000
		1	750000
24.576	6.1440	0	384000
		1	768000
28.7	4.9152	0	448436
		1	896872
30	7.5000	0	468750
		1	937500
33	8.2500	0	515625
		1	1031250

表 15.8　输入外部时钟时的最大位速率 （时钟同步模式、 $t_{Scyc}=12t_{pcyc}$ 时）

Pϕ （MHz）	外部输入时钟 （MHz）	最大位速率 （bit/s）
8	0.6666	666666.6
16	1.3333	1333333.3
24	2.0000	2000000.0
28.7	2.3916	2391666.6
30	2.5000	2500000.0
33	2.7500	2750000.0

15.3.9　FIFO 控制寄存器 （SCFCR）

SCFCR 是复位发送 FIFO 数据寄存器与接收 FIFO 数据寄存器的数据个数并设定触发数据个数的寄存器，SCFCR 还具有回送测试的允许位。

CPU 总能对 SCFCR 进行读写。

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	位名称	初始值	R/W	说明
15 ～ 11	—	均为 0	R	保留位 读写的值总是 0。
10 ～ 8	RSTRG[2:0]	000	R/W	$\overline{\text{RTS}}$ 输出有效触发 当保存在接收 FIFO 数据寄存器 （SCFRDR）的接收数据个数不小于下表所示的触发设定个数时， $\overline{\text{RTS}}$ 信号为高电平。 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 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 1。 • 异步模式 00：1 01：4 10：8 11：14 • 时钟同步模式 00：1 01：2 10：8 11：14 【注】时钟同步模式时，如果通过 DMAC 传送接收数据，请将接收触发个数设定为 1。不将其设定为 1 时，需要通过 CPU 读出 SCFRDR 中剩余的接收数据。

位	位名称	初始值	R/W	说明
5、4	TTRG[1:0]	00	R/W	发送 FIFO 数据个数的触发 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） 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 1。 00：8 （8） * 01：4 （12） * 10：2 （14） * 11：0 （16） * 【注】 * （）内的数值表示 TDFE 标志被置 1 时 SCFTDR 寄存器的空字节数。
3	MCE	0	R/W	允许调制解调器的控制 允许 / 禁止调制解调器的控制信号 $\overline{\text{CTS}}$ 和 $\overline{\text{RTS}}$。 请在通道 0 ～ 2、时钟同期模式中，总是将 MCE 置 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， $\overline{\text{CTS}}$ 被固定为有效 0， $\overline{\text{RTS}}$ 也被固定为 0。
2	TFRST	0	R/W	复位发送 FIFO 数据寄存器 使发送 FIFO 数据寄存器内的发送数据无效，且将数据复位为空状态。 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	复位接收 FIFO 数据寄存器 使接收 FIFO 数据寄存器内的接收数据无效，且将数据复位为空状态。 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 允许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 $\overline{\text{CTS}}$ 引脚连接并进行回送测试。 0：禁止回送测试 1：允许回送测试

15.3.10　FIFO 数据计数置位寄存器 （SCFDR）

SCFDR 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 接收数据的个数。 CPU 总是能读 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	位名称	初始值	R/W	说明
15 ～ 13	—	均为 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 表示无发送数据， H'10 表示发送数据均保存在 SCFTDR。
7 ～ 5	—	均为 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 表示无接收数据， H'10 表示接收数据均保存在 SCFRDR。

15.3.11　串行端口寄存器 （SCSPTR）

SCSPT 对（被多路复用为 SCIF 的引脚）端口的输入 / 输出及数据进行控制。对于 $\overline{\text{RTS}}$ 引脚，bit7、6 能读取数据以及写入输出数据。对于 $\overline{\text{CTS}}$ 引脚，bit5、4 能读取数据以及写入输出数据。对于 SCK 引脚，bit3、2 能读取数据以及写入输出数据。能通过 bit1、0 从 RxD 引脚读取输入数据、向 TxD 引脚写入输出数据、控制串行发送 / 接收的断点。

CPU 随时能读写 SCSPTR。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT
初始值：	0	0	0	0	0	0	0	0	0	1	0	1	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	位名称	初始值	R/W	说明
15 ～ 8	—	均为 0	R	保留位 读写的值总是 0。
7	RTSIO	0	R/W	输入 / 输出 $\overline{\text{RTS}}$ 端口 指定串行端口的 $\overline{\text{RTS}}$ 引脚的输入 / 输出。将 $\overline{\text{RTS}}$ 引脚实际作为端口输出引脚输出 RTSDT 位所设定的值时，请将 SCFCR 的 MCE 位置 0。 0：不将 RTSDT 位的值输出到 $\overline{\text{RTS}}$ 引脚 1：将 RTSDT 位的值输出到 $\overline{\text{RTS}}$ 引脚

位	位名称	初始值	R/W	说明
6	RTSDT	1	R/W	$\overline{\text{RTS}}$ 端口数据 指定串行端口的 $\overline{\text{RTS}}$ 引脚的输入 / 输出数据。通过 RTSIO 位指定为输入或者输出。输出时，将 RTSDT 位的值输出到 $\overline{\text{RTS}}$ 引脚。与 RTSIO 位的值无关，从 RTSDT 位读出 $\overline{\text{RTS}}$ 引脚的值。但是需要在 PFC 预先设定 $\overline{\text{RTS}}$ 输入 / 输出。 0：输入 / 输出数据为低电平 1：输入 / 输出数据为高电平
5	CTSIO	0	R/W	输入 / 输出 $\overline{\text{CTS}}$ 端口 指定串行端口的 $\overline{\text{CTS}}$ 引脚的输入 / 输出。将 $\overline{\text{CTS}}$ 引脚实际作为端口输出引脚输出 CTSDT 位的值时，请将 SCFCR 的 MCE 位置 0。 0：不将 CTSDT 位的值输出到 $\overline{\text{CTS}}$ 引脚 1：将 CTSDT 位的值输出到 $\overline{\text{CTS}}$ 引脚
4	CTSDT	1	R/W	$\overline{\text{CTS}}$ 端口数据 指定串行端口的 $\overline{\text{CTS}}$ 引脚的输入 / 输出数据。通过 CTSIO 位指定为输入或者输出。输出时，将 CTSDT 位的值输出到 $\overline{\text{CTS}}$ 引脚。与 CTSIO 位的值无关，从 CTSDT 位读出 $\overline{\text{CTS}}$ 引脚的值。但是需要在 PFC 预先设定 $\overline{\text{CTS}}$ 输入 / 输出。 0：输入 / 输出数据为低电平 1：输入 / 输出数据为高电平
3	SCKIO	0	R/W	SCK 端口输入 / 输出 指定串行端口的 SCK 引脚的输入 / 输出。将 SCK 引脚实际作为端口输出引脚输出 SCKDT 位的值时，请将 SCSCR 的 CKE1、 CKE0 位置 0。 0：不将 SCKDT 位的值输出到 SCK 引脚 1：将 SCKDT 位的值输出到 SCK 引脚
2	SCKDT	0	R/W	SCK 端口数据 指定串行端口的 SCK 引脚的输入 / 输出数据。通过 SCKIO 位指定为输入或者输出。输出时，将 SCKDT 位的值被输出到 SCK 引脚。与 SCKIO 位的值无关，从 SCKDT 位读出 SCK 引脚的值。但是需要在 PFC 预先设定 SCK 输入 / 输出。 0：输入 / 输出数据为低电平 1：输入 / 输出数据为高电平
1	SPB2IO	0	R/W	串行端口断点输入 / 输出 指定串行端口的 TxD 引脚的输出条件。将 TxD 引脚实际作为端口输出引脚输出 SPB2DT 位的值时，请将 SCSCR 的 TE 位置 0。 0：不将 SPB2DT 位的值输出到 TxD 引脚 1：将 SPB2DT 位的值输出到 TxD 引脚
0	SPB2DT	0	R/W	串行端口断点数据 指定串行端口的 RxD 引脚的输入数据以及 TxD 引脚的输出数据。通过 SPB2IO 位指定为输入或者输出。设定 TxD 引脚为输出时，将 SPB2DT 位的值输出到 TxD 引脚。与 SPB2IO 位的值无关，从 SPB2DT 位读出 RxD 引脚的值。但是需要在 PFC 预先设定 RxD 输入、 TxD 输出。 0：输入 / 输出数据为低电平 1：输入 / 输出数据为高电平

15.3.12　线路状态寄存器 （SCLSR）

CPU 能随时读写 SCLSR。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】* 为清除标志，只能在读1后，写0。

位	位名称	初始值	R/W	说明
15 ～ 1	—	均为 0	R	保留位 读写的值总是 0。
0	ORER	0	R/(W)*	溢出错误 表示在接收时发生溢出错误而结束了异常。 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 中均为 16 字节数据的状态下，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，不能继续进行以后的串行接收

15.3.13　串行扩展模式寄存器 （SCEMR）

CPU 能随时读写 SCEMR。通过将 BGDM 位置 1，当选择异步模式 （SCSMR 的 C/$\overline{A}$=0）且内部时钟被选择为时钟源以及 SCK 引脚用作输入引脚 （SCSCR 的 CKE[1:0]=00）时， SCIF 内部的波特率发生器以倍速模式运行。

另外，能通过改变 ABCS 位的设定，选择在异步模式 1 位期间的基本时钟。

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

位	位名称	初始值	R/W	说明
15 ～ 8	—	均为 0	R	保留位 读写的值总是 0。
7	BGDM	0	R/W	波特率发生器倍速模式 BGDM 位置 1 时， SCIF 内部的波特率发生器以倍速模式运行。 BGDM 的设定只在选择异步模式（SCSMR 的 C/$\overline{A}$=0）且内部时钟被选择为时钟源以及 SCK 引脚用作输入引脚 （SCSCR 的 CKE[1:0]=00）时有效。上述设定以外时请使用普通模式。 0：普通模式 1：倍速模式
6 ～ 1	—	均为 0	R	保留位 读写的值总是 0。
0	ABCS	0	R/W	选择异步基本时钟 选择在异步模式 1 位期间的基本时钟。 ABCS 位的设定只在异步模式 （SCSMR 的 C/$\overline{A}$=0）时有效。 0：以频率为位速率的 16 倍的基本时钟运行 1：以频率为位速率的 8 倍的基本时钟运行

15.4　运行说明

15.4.1　概要

SCIF 能以异步模式 （以字符为单位一边同步一边通信）和时钟同步模式 （通过时钟脉冲一边同步一边通信）这 2 种方式进行串行通信。

SCIF 内置了分别用于接收 / 发送的 16 段 FIFO 寄存器，能减少 CPU 的开销以及能进行高速连续通信。而且通道 3 还具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 / 接收格式，如表 15.9 所示。另外， SCIF 时钟源取决于串行控制寄存器 （SCSCR）的 CKE[1:0] 位的组合，如表 15.10 所示。

1. 异步模式
   - 数据长：可选择7位或者8位
   - 可选择附加奇偶性和附加1位/2位的停止位 （根据这些组合，决定发送/接收格式及字符长）
   - 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、接收数据就绪和断点
   - 表示各发送/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/外部时钟
     选择内部时钟时：通过波特率发生器时钟运行SCIF
     选择外部时钟时：需要输入频率为16/8倍位速率的时钟 （不使用内置波特率发生器）
2. 时钟同步模式
   - 发送/接收格式：固定为8位数据
   - 接收时，能检测到溢出错误
   - SCIF的时钟源：可选择内部时钟/外部时钟
     选择内部时钟时：通过波特率发生器的时钟运行SCIF，并且将此时钟作为同步时钟输出到外部
     选择外部时钟时：不使用内置波特率发生器，而使用输入的外部同步时钟运行SCIF

表 15.9　SCSMR 的设定值和 SCIF 的发送 / 接收格式

SCSMR				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
C/$\overline{\text{A}}$	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 15.10　SCSMR、 SCSCR 的设定值和 SCIF 时钟源的选择

SCSMR	SCSCR	模式	时钟源	SCK 引脚的功能
bit7	bit1、 0			
C/A	CKE[1:0]			
0	00	异步模式	内部	SCIF 不使用 SCK 引脚
	01			输出频率为 16/8 倍位速率的时钟
	10		外部	输出频率为 16/8 倍位速率的时钟
	11		禁止设定	
1	0x	时钟同步模式	内部	输出同步时钟
	10		外部	输入同步时钟
	11		禁止设定	

【符号说明】x：Don't care

【注】 使用波特率发生器倍速模式 （BGDM=1）时，请选择异步模式 （C/$\overline{A}$=0）且选择内部时钟为时钟源以及请勿使用 SCK 引脚 （CKE[1:0]=00）

15.4.2　异步模式中的运行

异步模式是以字符为单位一边同步一边进行串行通信的模式，用来发送 / 接收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的字符。

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 / 接收中读写数据，因此可进行连续发送 / 接收。

异步模式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）。SCIF 监视通信线路，将变为空状态（低电平）的位视为起始位，开始串行通信。

以起始位 （低电平）开始，数据 （LSB 先：从最低有效位开始）、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 在接收时在起始位的下降沿进行同步。另外，因为 SCIF 在 1 个位期间频率为 16/8 倍位速率时钟的第 8/4 个时钟采样数据，所以在各位的中央获取通信数据。

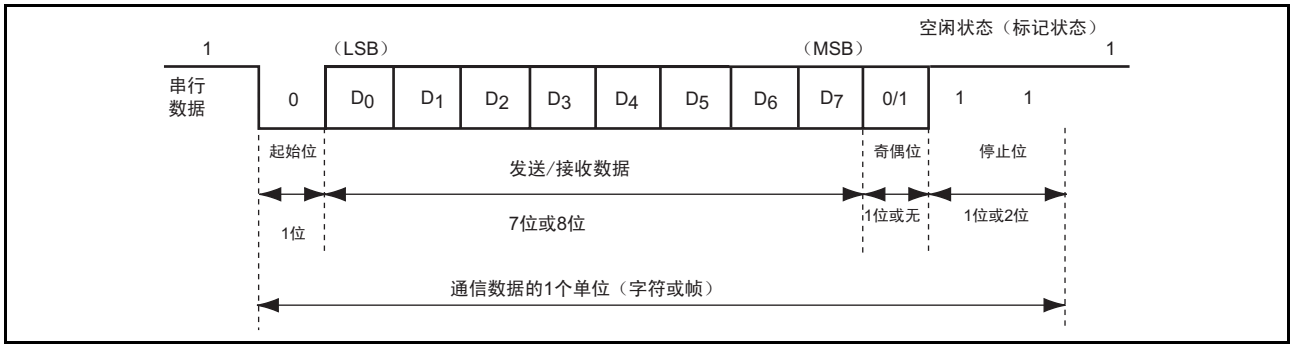


图 15.2　异步通信的数据格式
（8 位数据 / 有奇偶性 /2 个停止位的例子）

（1） 发送 / 接收格式

能设定的发送 / 接收格式如**表 15.11** 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 15.11　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
		1	START 8位数据 STOP STOP
	1	0	START 8位数据 P STOP
		1	START 8位数据 P STOP STOP
1	0	0	START 7位数据 STOP
		1	START 7位数据 STOP STOP
	1	0	START 7位数据 P STOP
		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 C/$\overline{A}$ 位和串行控制寄存器 （SCSCR）的 CKE[1:0] 位，从内置波特率发生器生成的内部时钟或者从 SCK 引脚输入的外部时钟中选择 SCIF 的发送 / 接收时钟。有关 SCIF 时钟源的选择请参照**表 15.10**。

在给 SCK 引脚输入外部时钟时，请输入频率是 16/8 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 16/8 倍。

（3） 数据的发送 / 接收

初始化 SCIF （异步模式）

在数据的发送/接收前，请在将串行控制寄存器（SCSCR）的TE位及RE位清0后，按以下顺序初始化SCIF。

在更改运行模式和通信格式等时，请在将 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。但是必须注意：即使将 TE 位和 RE 位清 0，也不初始化串行状态寄存器 （SCFSR）、发送 FIFO 数据寄存器 （SCFTDR）、接收 FIFO 数据寄存器 （SCFRDR）而保持其内容。请在发送所有的发送数据并且置位 SCFSR 的 TEND 标志后，将 TE 位清 0。即使在发送过程也能将 TE 位清 0，而清 0 后发送数据为标记状态。另外，重新将 TE 位置 1 开始发送前，请将 SCFCR 的 TFRST 位置 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 15.3 所示。

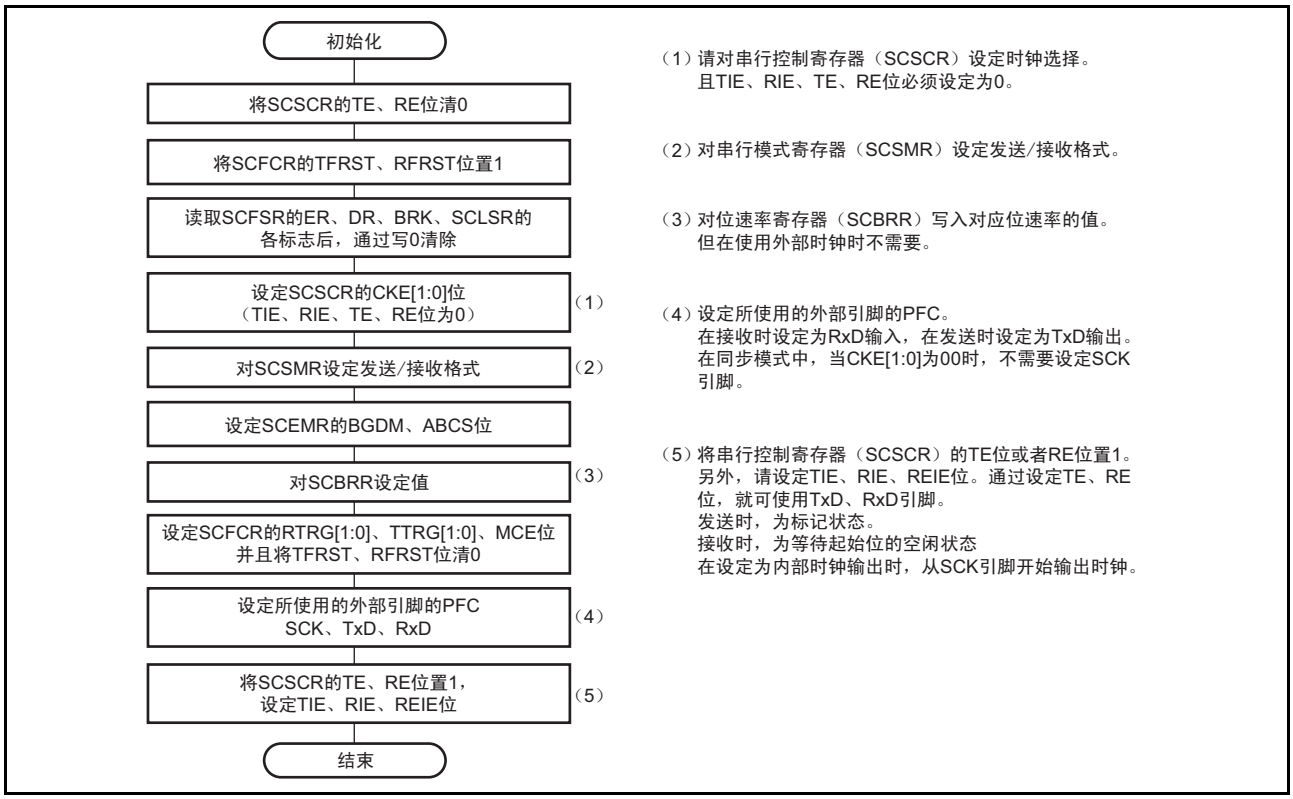


图 15.3　SCIF 初始化流程图例子

发送串行数据 （异步模式）

串行发送的流程图例子如图 15.4 所示。

在将 SCIF 设定为可发送状态后，请按照以下步骤发送串行数据。

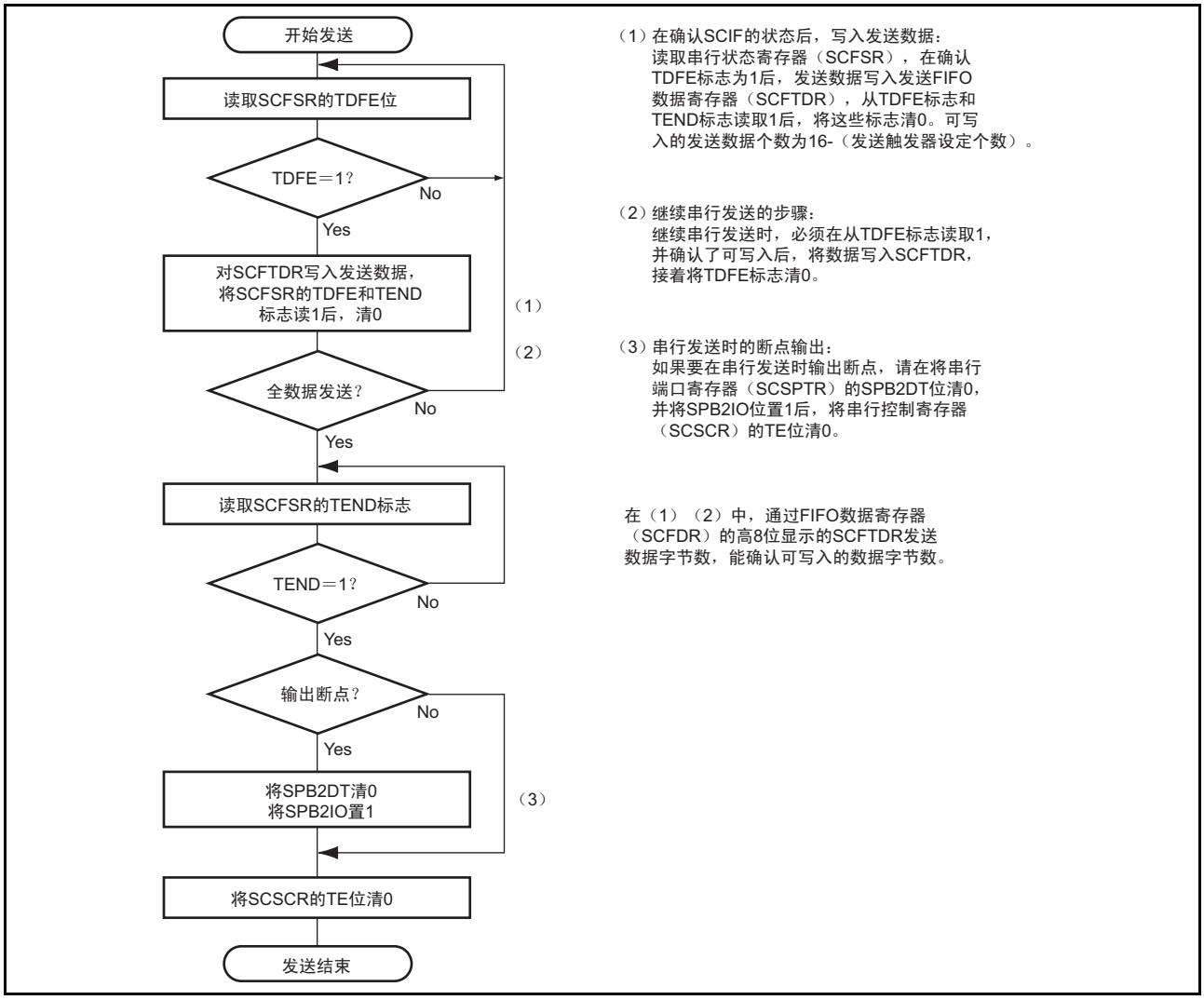


图 15.4　串行发送的流程图例子

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，请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，如果将串行控制寄存器（SCSCR）的TIE位置1，就产生发送FIFO数据空中断（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. 奇偶校验位　：输出1位的奇偶校验位 （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　：在输出开始下一次发送的起始位前连续输出1。
3. SCIF在输出停止位时检查SCFTDR的发送数据。如果有数据，就将数据从SCFTDR传送到SCTSR，在输出停止位后，开始下一帧的串行发送。

异步模式下发送时的运行例子如图15.5所示。

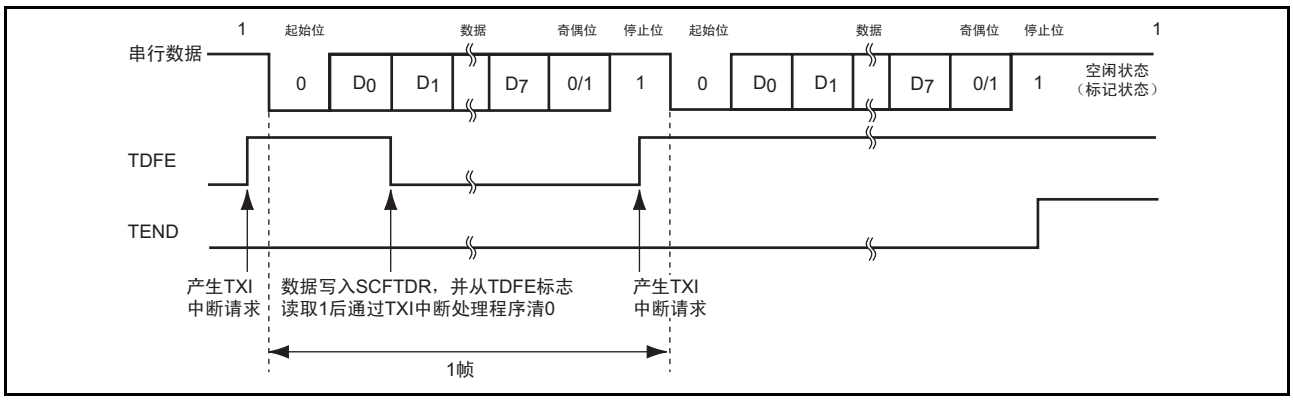


图 15.5　发送时的运行例 （8 位数据 / 有奇偶性 /1 个停止位）

4. 如果在通道3允许调制解调器控制，就能通过$\overline{\text{CTS}}$输入值停止/重新开始发送。将$\overline{\text{CTS}}$置1时，如果是正在发送，就在发送完1帧后进入标记状态。将$\overline{\text{CTS}}$置0时，就从起始位开始输出下一个发送数据。

使用调制解调器控制的运行例子如图15.6所示。

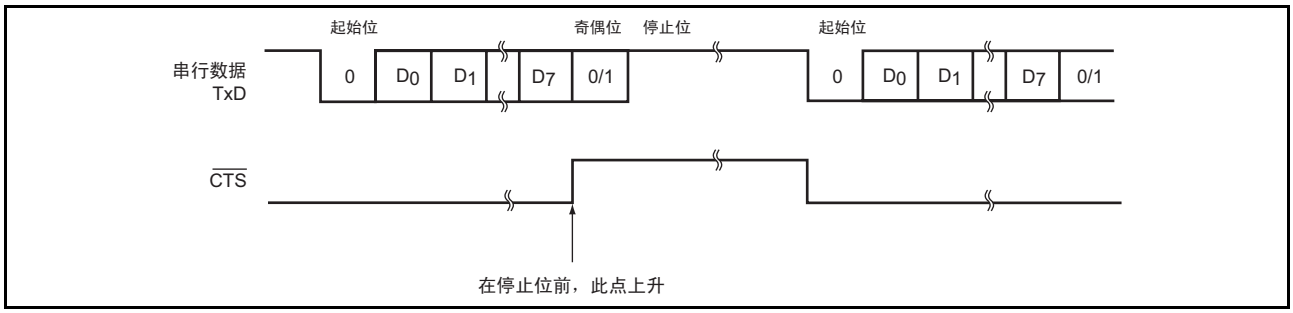


图 15.6　使用调制解调器控制的运行例 （$\overline{\text{CTS}}$）

接收串行数据 （异步模式）

串行接收的流程图例子如**图 15.7** 和**图 15.8** 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。

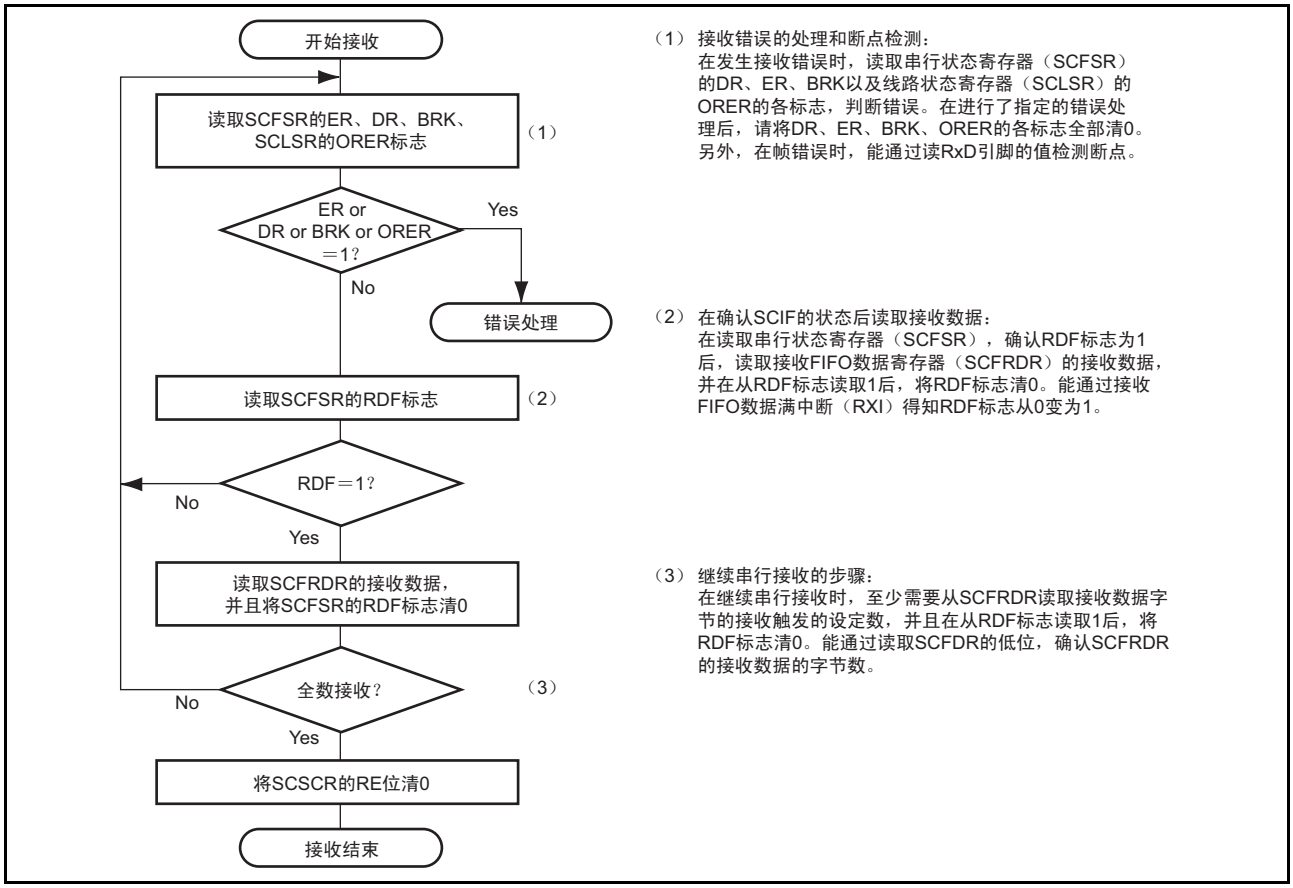


图 15.7　串行接收的流程图例子 （1）

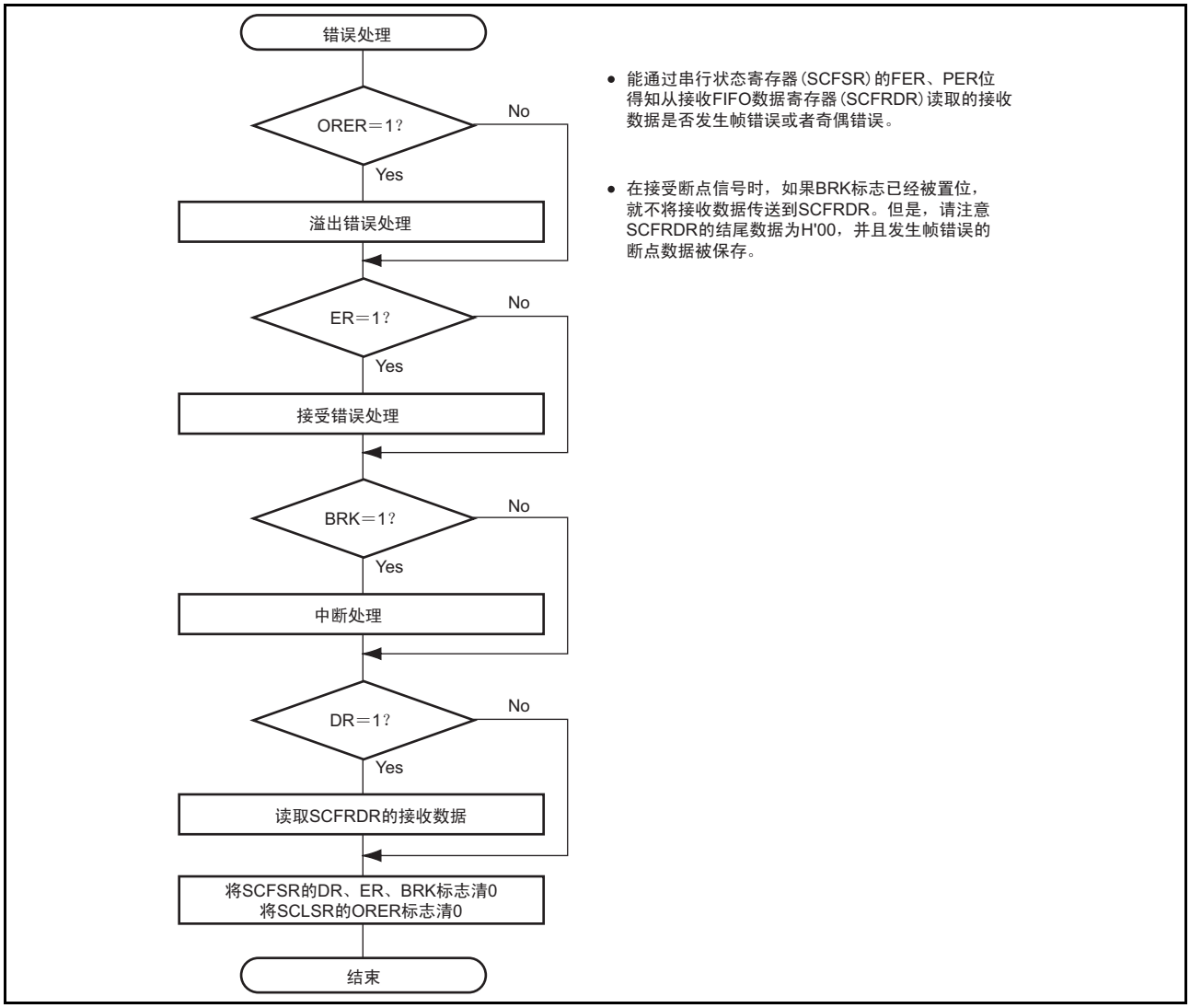


图 15.8　串行接收的流程图例子 （2）

接收时SCIF运行如下：

1. SCIF监视通信线路，如果检测到起始位的0，就同步内部且开始接收。
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。

   接收后， SCIF进行以下检查：

   a. 检查停止位：检查停止位是否为1。但是2个停止位时，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. 检查溢出错误：检查表示未发生溢出错误地ORER标志是否为0。
   d. 检查断点：检查表示未设定断点状态的BRK标志是否为0。

   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. RDF 或者 DR 标志已经是 1 时，如果 SCSCR 的 RIE 位被置 1，就发生接收 FIFO 数据满中断 （RXI）请求。 ER标志也已经是1时，如果SCSCR的RIE位或者REIE位被置1，就产生接收错误中断 （ERI）请求； BRK 标志或者 ORER 标志已经是 1 时，如果 SCSCR 的 RIE 位或者 REIE 位被置 1，就产生断点接收中断 （BRI）请求。

异步模式接收时的运行例如图 15.9 所示。

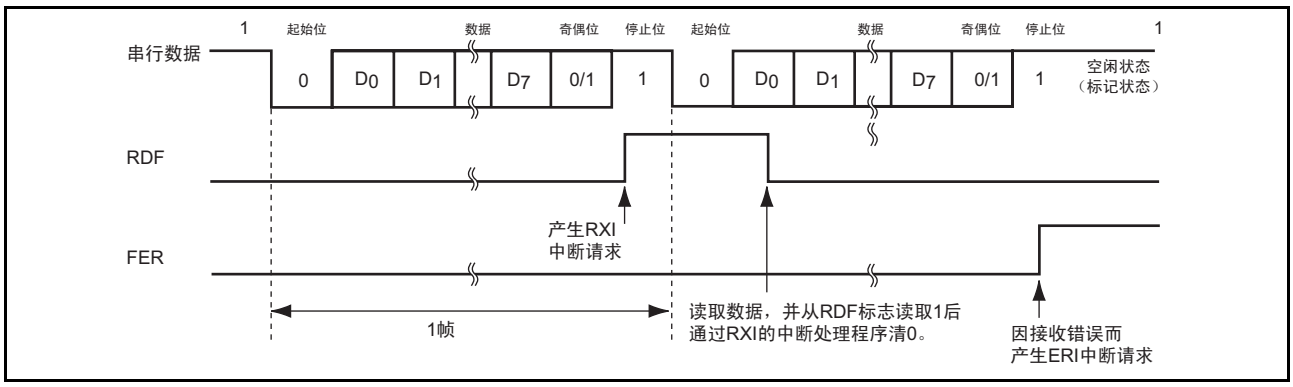


图 15.9　SCIF 接收时的运行例 （8 位数据 / 有奇偶 /1 个停止位）

5. 当通道 3 的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 的数据个数大于等于 RTS 输出有效触发设定个数。

使用调制解调器控制时的运行例如图 15.10 所示。

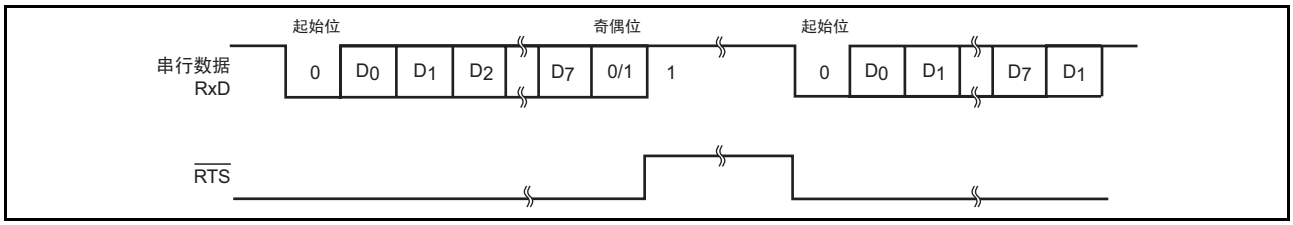


图 15.10　使用调制解调器控制时的运行例 （$\overline{\text{RTS}}$）

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 / 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。另外，发送部和接收部都是 16 段 FIFO 缓冲结构，所以能在发送 / 接收中读写数据，进行连续的发送 / 接收。

时钟同步串行通信的一般格式如图 15.11 所示。

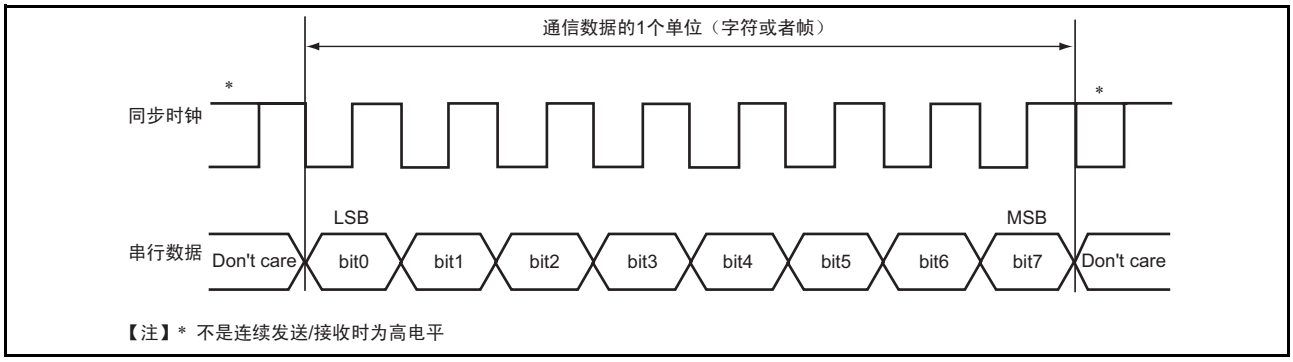


图 15.11　时钟同步通信的数据格式

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输出通信线路的数据，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 C/$\overline{A}$ 位和 SCSCR 的 CKE[1:0] 位，选择内置波特率发生器生成的内部时钟或者从 SCK 引脚输入的外部同步时钟。

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 / 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 位为 1 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。

（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/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。在改变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。将 TE 位清 0 时，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 各标志及接收数据寄存器 （SCRDR）的内容。

SCIF 的初始化流程图例子如图 15.12 所示。

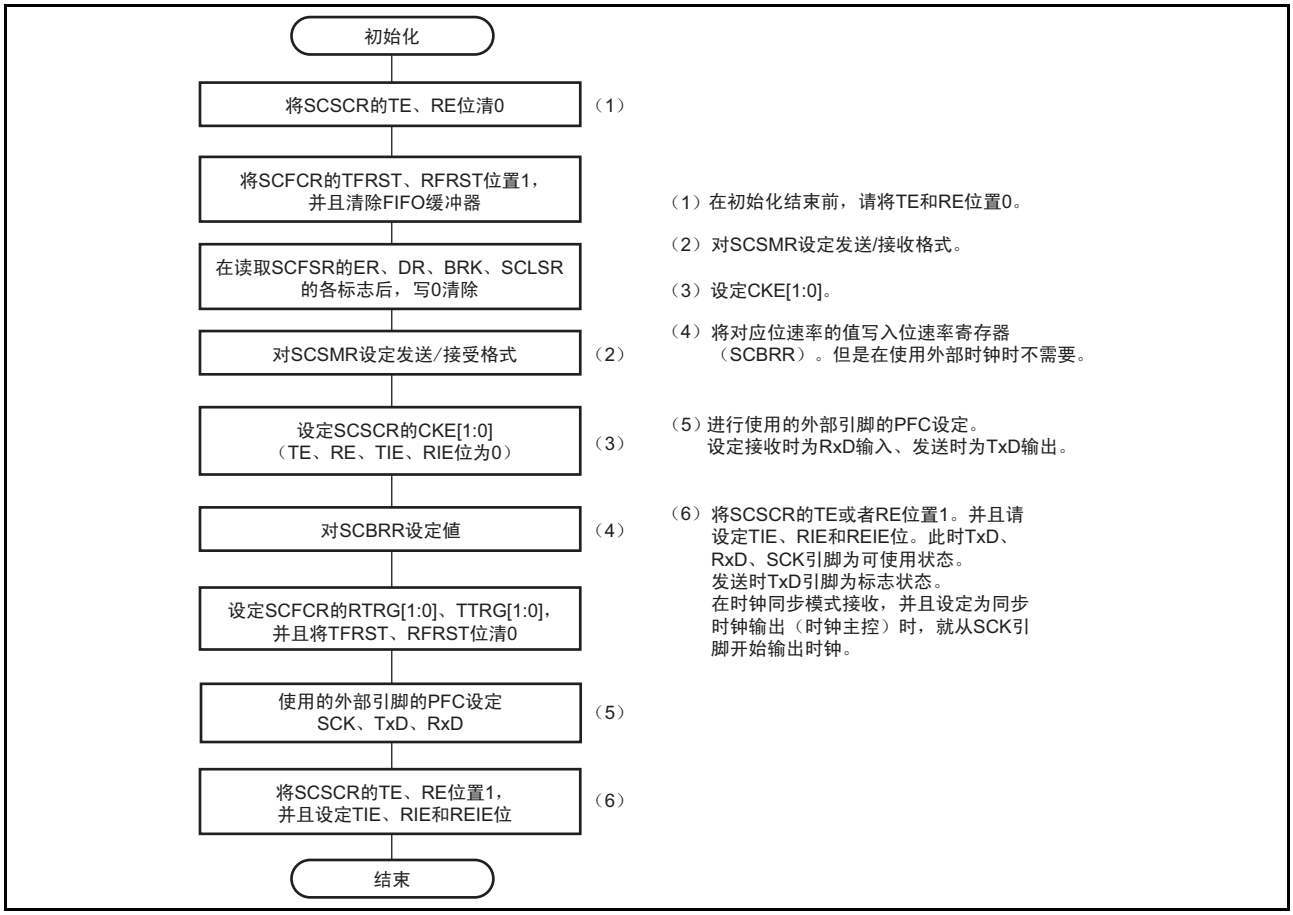


图 15.12　SCIF 的初始化流程图例子

发送串行数据的发送 （时钟同步模式）

串行发送的流程图例子如图 15.13 所示。

在将 SCIF 设定为可发送状态后，请按照以下步骤发送串行数据。

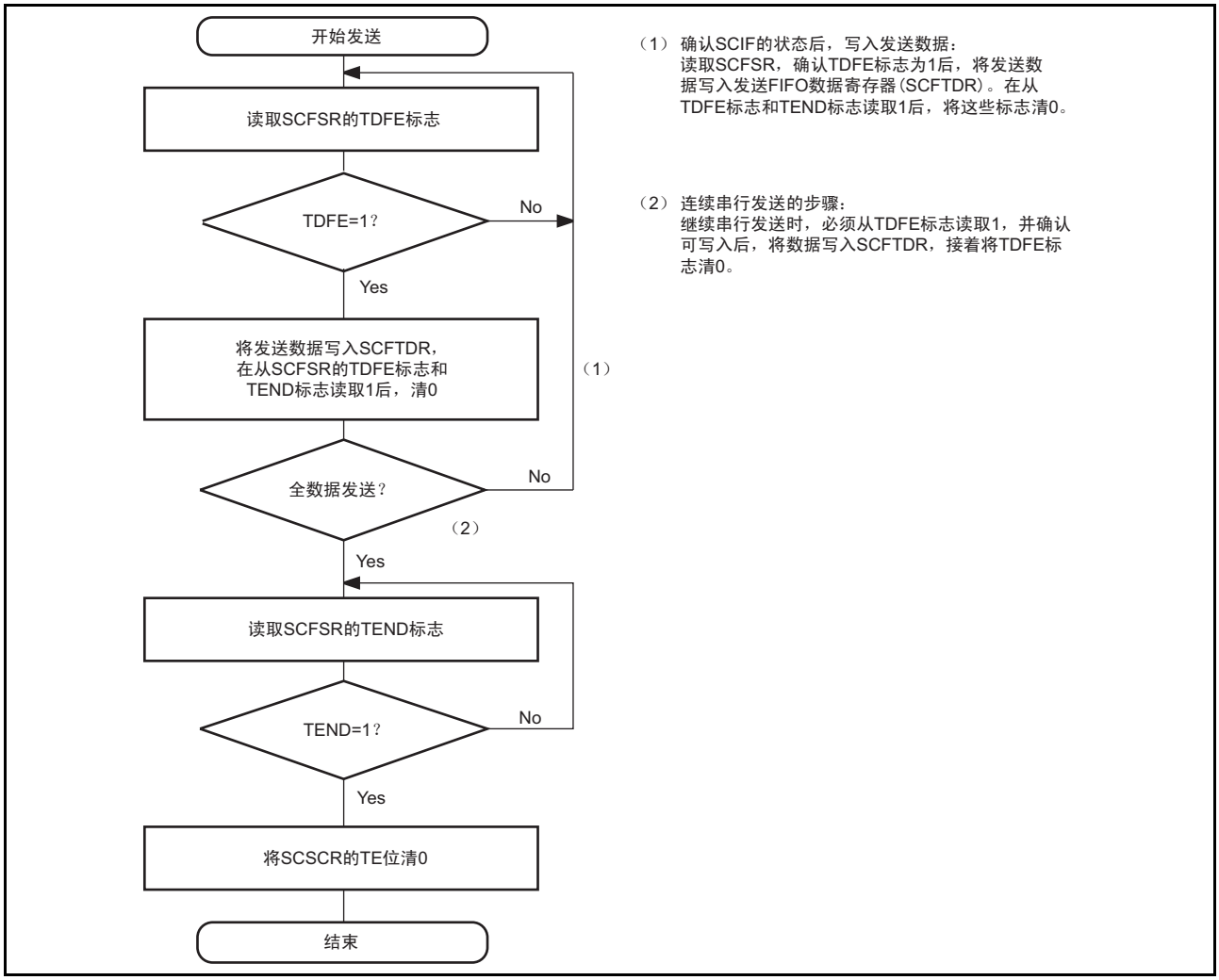


图 15.13　串行发送的流程图例子

串行发送时 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，请确认是否将串行状态寄存器（SCFSR）的 TDFE 标志置 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等于 FIFO 控制寄存器（SCFCR）设定的发送触发个数时，置位 TDFE 标志。此时，如果将串行控制寄存器 （SCSCR）的 TIE 位置 1，就产生发送 FIFO 数据空中断（TXI）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 （bit0）～MSB （bit7）的顺序从 TxD 引脚输出串行发送数据。
3. SCIF 在输出最后位时检查 SCFTDR 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；如果没有数据，就将 SCFSR 的 TEND 标志置 1，在输出最后位后， 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 发送时的运行例如图 15.14 所示。

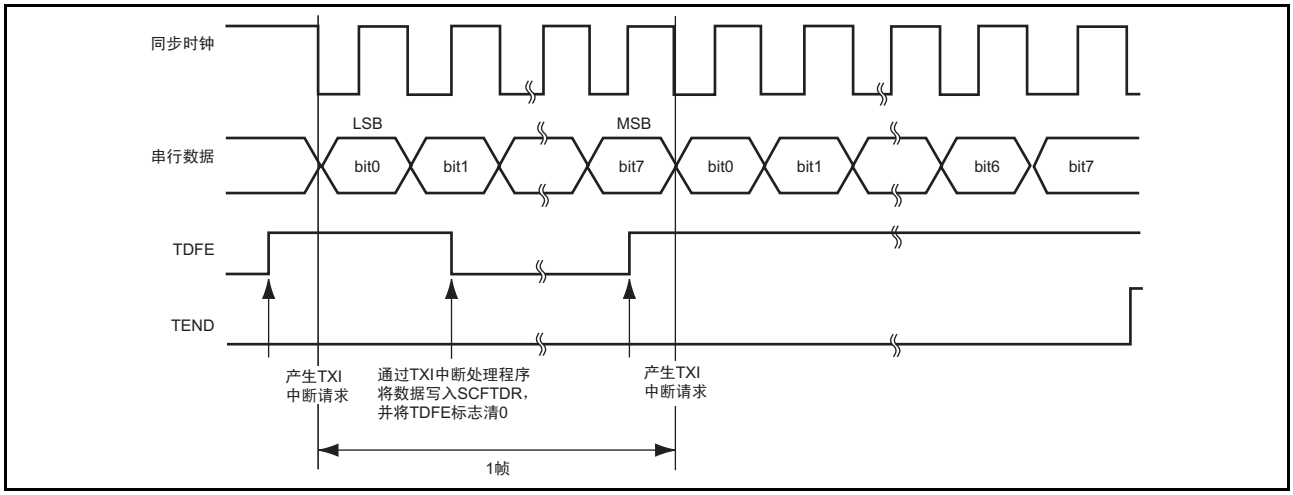


图 15.14　SCIF 发送时的运行例

接收串行数据 （时钟同步模式）

串行接收的流程图例子如图 15.15 和图 15.16 所示。

在将 SCIF 设定为可接收状态后，请按照以下步骤接收串行数据。

在不初始化 SCIF 并将运行模式从异步模式转换为时钟同步模式时，必须确认是否已将 ORER、PER、FER 各标志清 0。

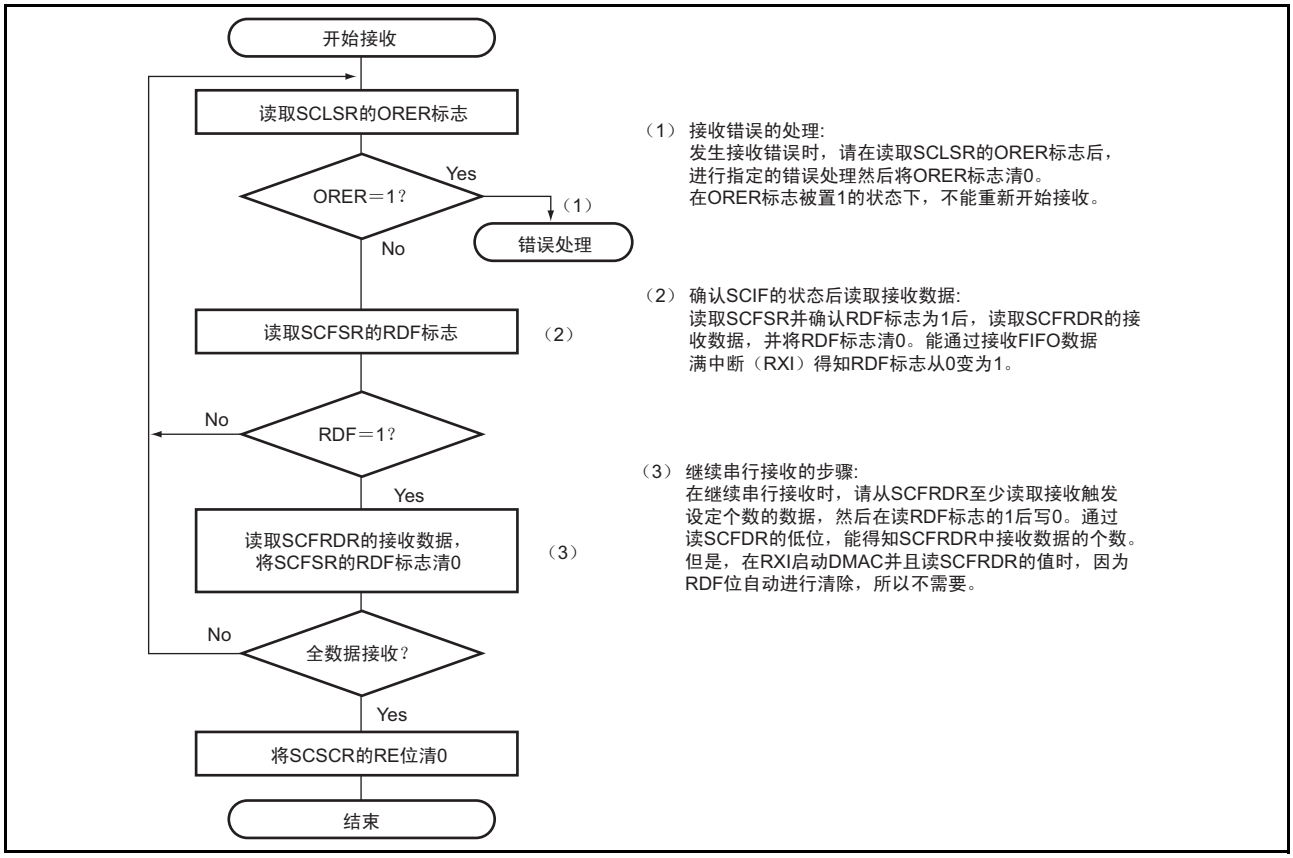


图 15.15　串行接收的流程图例子 （1）

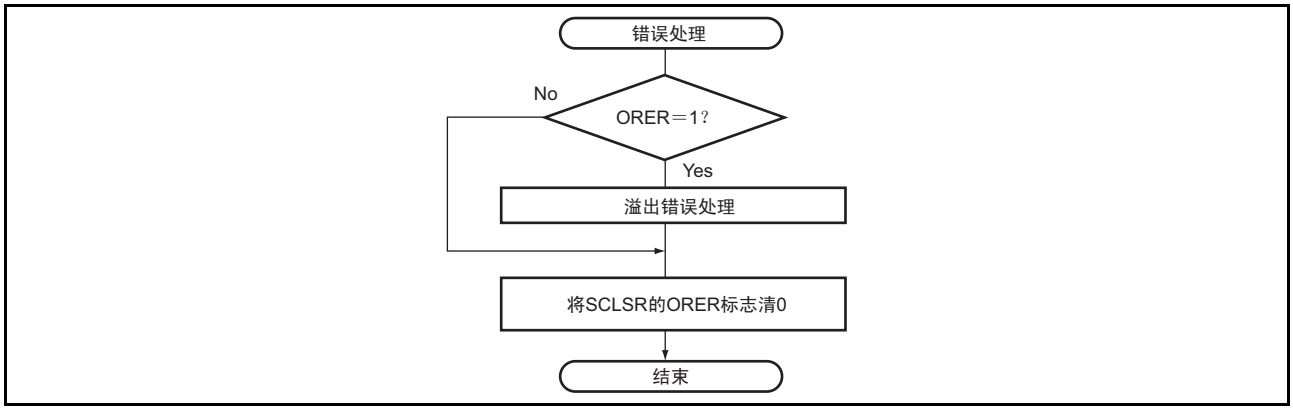


图 15.16　串行接收的流程图例子 （2）

串行接收时 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 RDF 标志置 1，并将接收数据保存到 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. RDF 标志是 1 时，如果将串行控制寄存器（SCSCR）的 RIE 位置 1，就产生接收 FIFO 数据满中断（RXI）请求； ORER 标志是 1 时，如果将 SCSCR 的 RIE 位或者 REIE 位置 1，就产生断点的中断 （BRI）请求。

SCIF 接收时的运行例如图 15.17。

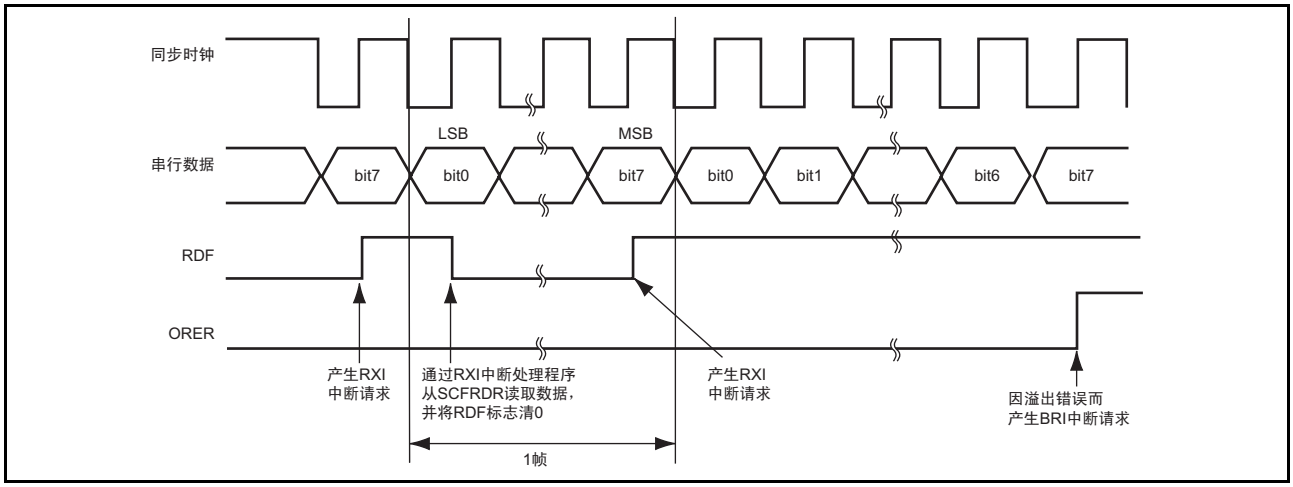


图 15.17　SCIF 接收时的运行例

同时进行串行数据的发送 / 接收 （时钟同步模式）

同时进行串行发送 / 接收的流程图例子如**图 15.18** 所示。

在将 SCIF 设定为可发送 / 接收状态后，必须按照以下步骤同时发送 / 接收串行数据。

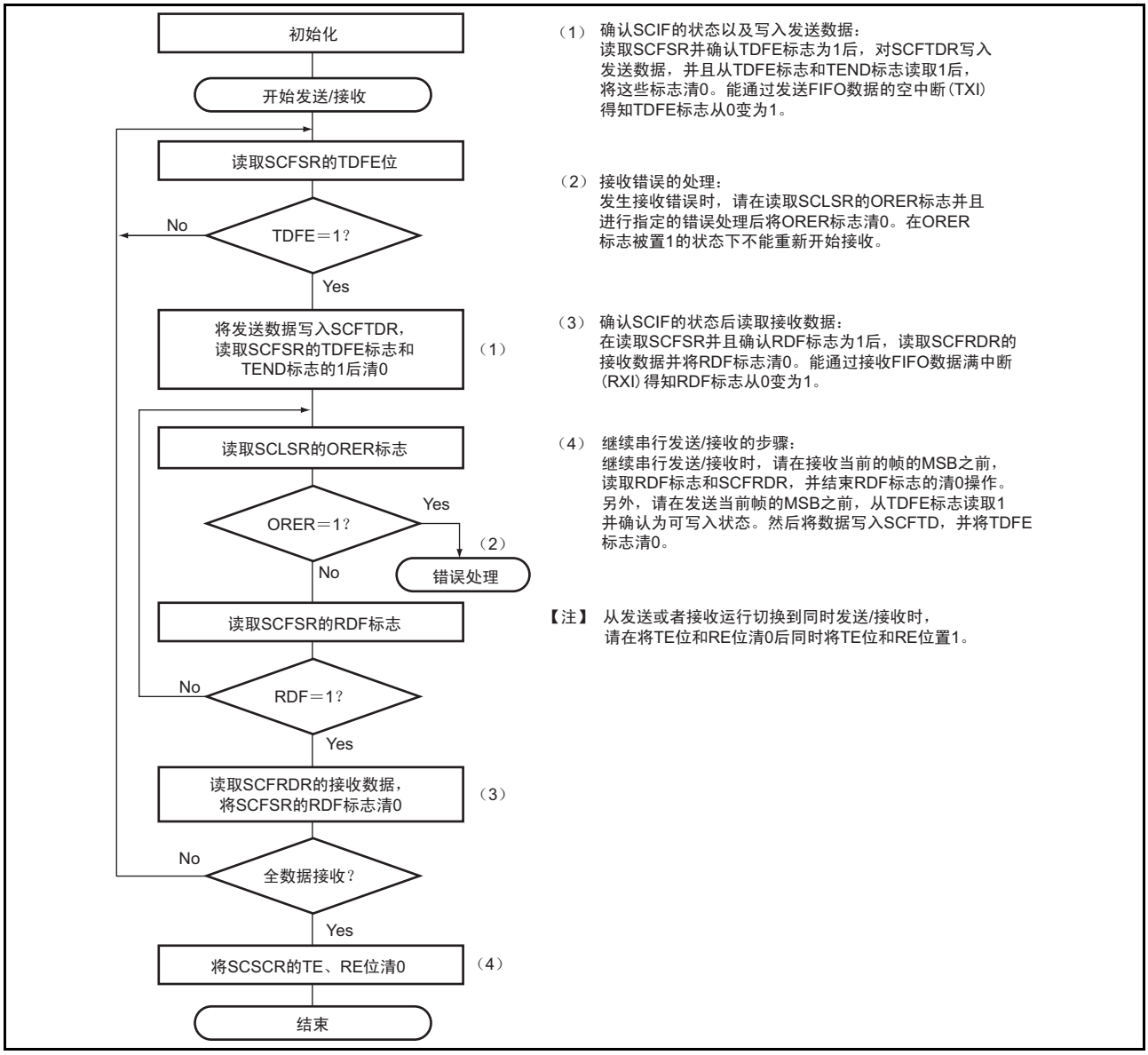


图 15.18　串行数据发送和接收的流程图例子

15.5　SCIF 的中断

SCIF 有发送 FIFO 数据空中断 （TXI）请求、接收错误中断 （ERI）请求、接收 FIFO 数据满中断 （RXI）请求和断点的中断 （BRI）请求共 4 种中断源。

各中断源和优先级如**表 15.12** 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。

通过 TIE 位允许 TXI 中断时，如果将串行状态寄存器 （SCFSR）的 TDFE 标志置 1，就产生 TXI 中断。能通过此 TXI 中断请求启动 DMAC，并进行数据传送。此时，不发生对 CPU 的中断请求。

通过 RIE 位允许 RXI 中断时，如果将 SCFSR 的 RDF 标志或者 DR 标志置 1，就产生 RXI 中断请求。能通过此 RXI 中断请求启动 DMAC，并进行数据传送。此时，不产生对 CPU 的中断请求。但是只在异步模式时，才能通过将 DR 标志置 1 产生 RXI 中断请求。

如果将 SCSCR 的 RIE 位置 0，并且将 REIE 位置 1，就不能产生 RXI 中断请求而只能产生 ERI 中断请求。

另外， TXI 表示能写发送数据， RXI 表示 SCFRDR 中有接收数据。

表 15.12　SCIF 中断源

中断源	内　　容	启动 DMAC	解除复位时 优先级
BRI	断点 （BRK）或者溢出 （ORER）引起的中断	不可	高 ↕ 低
ERI	接收错误 （ER）引起的中断	不可	
RXI	接收 FIFO 数据满 （RDF）或者数据就绪 （DR）引起的中断	可	
TXI	发送 FIFO 数据空 （TDFE）引起的中断	可	

15.6　使用时的注意事项

在使用 SCIF 时，请注意以下几点。

15.6.1　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TTRG[1:0]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。置位 TDFE 标志后，能在 SCFTDR 变空之前写发送数据，进行高效率的连续发送。

但是，如果写到 SCFTDR 的数据字节数小于等于发送触发个数，即使在读 1 后将 TDFE 标志清 0，TDFE 标志也将被再次置 1。因此，写入 SCFTDR 的数据字节数超过发送触发个数时，必须在读 1 后将 TDFE 清 0。

能通过 FIFO 数据计数寄存器 （SCFDR）的高 8 位得知 SCFTDR 的发送数据字节数。

15.6.2　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等于 FIFO 控制寄存器 （SCFCR）的 RTRG[1:0] 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 标志被置位后，能从 SCFRDR 读取触发数个接收数据，进行高效率的连续接收。

但是，如果 SCFRDR 的数据字节数超过接收触发个数，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 数据计数寄存器 （SCFDR）高 8 位得知 SCFRDR 的接收数据字节数。

15.6.3　DMAC 使用时的注意事项

DMAC 通过 TXI 中断请求向 SCFTDR 写入数据时， TEND 标志的状态不定。因此，此时不能将 TEND 标志作为传送结束标志使用。

15.6.4　断点的检测和处理

检测帧错误 （FER）时，能通过直接读取 RxD 引脚的值检测断点。断点时， RxD 引脚的输入始终为 0，因此置位 FER 标志，并且也可能置位奇偶校验错误标志 （PER）。

如果检测到断点， SCIF 就停止向 SCFRDR 传送接收数据，而还能继续接收。

15.6.5　断点的发送

能通过串行端口寄存器（SCSPTR）的 SPB2IO 位和 SPB2DR 位决定 TxD 引脚的输入 / 输出的条件和电平，所以能利用这一功能发送断点。

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 SPB2DT 位的值替代。因此，先要将 SPB2IO 和 SPB2DT 位置 1 （输出高电平）。

为了在串行发送时发送断点信号，就必须在将 SPB2DT 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。

15.6.6　异步模式的接收数据采样时序和接收容限

SCIF 通过频率为 16/8 倍位速率的基本时钟运行。接收时， SCIF 通过基本时钟采样起始位的下降沿，同步内部，而在基本时钟的第 8/4 个时钟的上升沿在内部获取接收数据。以 16 倍频率的基本时钟运行时的时序如图 15.19 所示。

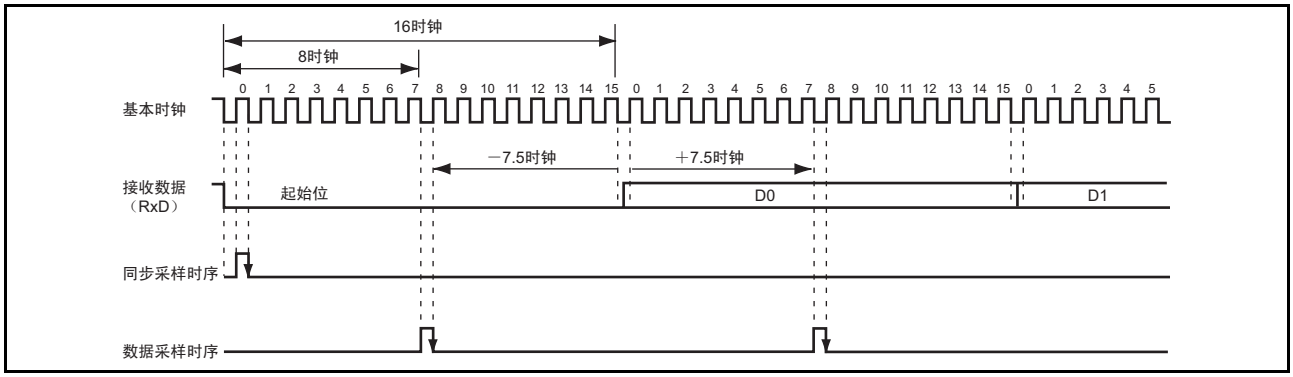


图 15.19　异步模式的接收数据采样时序 （以频率为 16 倍位速率的基本时钟运行）

因此，能用公式 （1）表示异步模式的接收容限。

$$M = \{(0.5 - \frac{1}{2N}) - (L-0.5)F - \frac{|D-0.5|}{N}(1+F)\} \times 100[\%] \quad \cdots\text{公式（1）}$$

M：接收容限 （%）

N：时钟频率与位速率的比 （N=16/8）

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

如果公式 （1）中 F=0、 D=0.5 并且 N=16，根据公式 （2），接收容限为 46.875%

D＝0.5、F＝0时

$$M=(0.5-1/(2\times16))\times100\%=46.875\% \quad \cdots\text{公式（2）}$$

但是，此值为计算值，所以在系统设计时必须留有 20 ～ 30% 的余地。

15.6.7　选择异步基本时钟

本 LSI 能通过变更串行扩展模式寄存器 （SCEMR）的 ABCS 位，将异步模式时 1 位期间的基本时钟频率设置为位速率的 16/8 倍。

但是，由 “15.6.6 异步模式的接收数据采样时序和接收容限”的公式 （1）可知，必须注意基本时钟的频率低于位速率的 8 倍时接收容限就会减少。

如果只能通过串行模式寄存器 （SCSMR）的 CKS[1:0]、位速率寄存器 （SCBRR）的设定实现期望的位速率时，建议将 1 位期间的基本时钟频率设定为 16 倍位速率 （SCEMR 的 ABCS=0）。如果选者内部时钟为时钟源以及不使用 SCK 引脚，就能通过设定为波特率发生器倍速模式 （SCEMR 的 BGDM=1），在不减少接收容限的情况下提高位速率。

第 16 章 同步串行通信单元 （SSU）

本 LSI 具有 2 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。

16.1 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优先方式或者MSB优先方式
- 可选择7种内部时钟（Pϕ/4、 Pϕ/8、 Pϕ/16、 Pϕ/32、 Pϕ/64、 Pϕ/128、 Pϕ/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，启动直接存储器存取控制器 （DMAC）进行数据传送。
- 能设定模块待机模式
  为降低功耗，可通过停止向SSU提供时钟来中止运行。

SSU 的框图如图 16.1 所示。

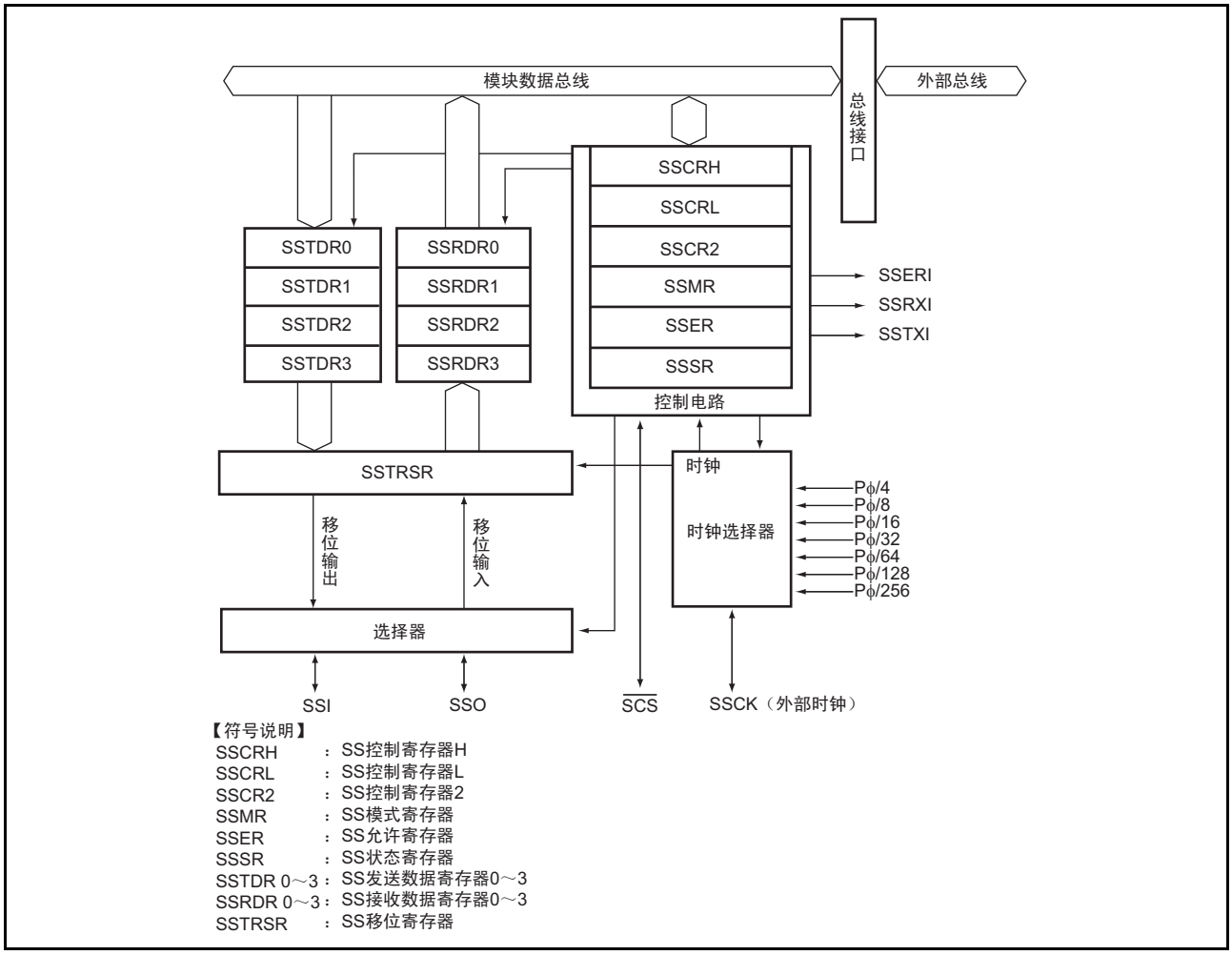


图 16.1　SSU 的框图

16.2　输入 / 输出引脚

SSU 有表 16.1 中的输入 / 输出引脚。

表 16.1　引脚结构

通道	引脚名	输入 / 输出	功能
0、 1	SSCK0、 SSCK1	输入 / 输出	SSU 的时钟输入 / 输出引脚
	SSI0、 SSI1	输入 / 输出	SSU 的数据输入 / 输出引脚
	SSO0、 SSO1	输入 / 输出	SSU 的数据输入 / 输出引脚
	$\overline{\text{SCS0}}$、 $\overline{\text{SCS1}}$	输入 / 输出	SSU 的片选输入 / 输出引脚

16.3　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下寄存器的状态，请参照 “第 34 章 寄存器一览”。

表 16.2　寄存器结构

通道	寄存器名	简称	R/W	初始值	地址	存取长度
0	SS 控制寄存器 H_0	SSCRH_0	R/W	H'0D	H'FFFE7000	8、16
	SS 控制寄存器 L_0	SSCRL_0	R/W	H'00	H'FFFE7001	8
	SS 模式寄存器 _0	SSMR_0	R/W	H'00	H'FFFE7002	8、16
	SS 允许寄存器 _0	SSER_0	R/W	H'00	H'FFFE7003	8
	SS 状态寄存器 _0	SSSR_0	R/W	H'04	H'FFFE7004	8、16
	SS 控制寄存器 2_0	SSCR2_0	R/W	H'00	H'FFFE7005	8
	SS 发送数据寄存器 0_0	SSTDR0_0	R/W	H'00	H'FFFE7006	8、16
	SS 发送数据寄存器 1_0	SSTDR1_0	R/W	H'00	H'FFFE7007	8
	SS 发送数据寄存器 2_0	SSTDR2_0	R/W	H'00	H'FFFE7008	8、16
	SS 发送数据寄存器 3_0	SSTDR3_0	R/W	H'00	H'FFFE7009	8
	SS 接收数据寄存器 0_0	SSRDR0_0	R	H'00	H'FFFE700A	8、16
	SS 接收数据寄存器 1_0	SSRDR1_0	R	H'00	H'FFFE700B	8
	SS 接收数据寄存器 2_0	SSRDR2_0	R	H'00	H'FFFE700C	8、16
	SS 接收数据寄存器 3_0	SSRDR3_0	R	H'00	H'FFFE700D	8
1	SS 控制寄存器 H_1	SSCRH_1	R/W	H'0D	H'FFFE7800	8、16
	SS 控制寄存器 L_1	SSCRL_1	R/W	H'00	H'FFFE7801	8
	SS 模式寄存器 _1	SSMR_1	R/W	H'00	H'FFFE7802	8、16
	SS 允许寄存器 _1	SSER_1	R/W	H'00	H'FFFE7803	8
	SS 状态寄存器 _1	SSSR_1	R/W	H'04	H'FFFE7804	8、16
	SS 控制寄存器 2_1	SSCR2_1	R/W	H'00	H'FFFE7805	8
	SS 发送数据寄存器 0_1	SSTDR0_1	R/W	H'00	H'FFFE7806	8、16
	SS 发送数据寄存器 1_1	SSTDR1_1	R/W	H'00	H'FFFE7807	8
	SS 发送数据寄存器 2_1	SSTDR2_1	R/W	H'00	H'FFFE7808	8、16
	SS 发送数据寄存器 3_1	SSTDR3_1	R/W	H'00	H'FFFE7809	8
	SS 接收数据寄存器 0_1	SSRDR0_1	R	H'00	H'FFFE780A	8、16
	SS 接收数据寄存器 1_1	SSRDR1_1	R	H'00	H'FFFE780B	8
	SS 接收数据寄存器 2_1	SSRDR2_1	R	H'00	H'FFFE780C	8、16
	SS 接收数据寄存器 3_1	SSRDR3_1	R	H'00	H'FFFE780D	8

16.3.1　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 $\overline{\text{SCS}}$ 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	选择主器件 / 从属器件 选择是将 SSU 用作主模式还是用作从属模式。用作主模式时，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	允许双向模式 选择是使用串行数据的输入引脚和输出引脚 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 “16.4.3 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 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	选择串行数据输出值 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，但是能在发送前或者发送后更改串行数据的输出电平。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 要更改串行数据的输出电平时，请用 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 对本位写 0 时，请在确认读 1 后写 0。
2	—	1	R	保留位 读写的值总是 1。
1、0	CSS[1:0]	01	R/W	选择 $\overline{\text{SCS}}$ 引脚 选择是将 $\overline{\text{SCS}}$ 引脚用作 SCS 输入还是用作 SCS 输出。 00：禁止设定 01：用作 $\overline{\text{SCS}}$ 输入 10：用作 $\overline{\text{SCS}}$ 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 $\overline{\text{SCS}}$ 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）

16.3.2　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 选择是在存取相应寄存器时还是在 DMAC 传送结束时清除中断标志。在使用 DMAC 时，请将此位设定为 1。 0：在存取寄存器时 1：在 DMAC 传送结束时
6	SSUMS	0	R/W	选择 SSU 模式和时钟同步通信模式。 0：SSU 模式 1：时钟同步通信模式
5	SRES	0	R/W	软件复位 如果将此位置 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，请给此位写 1，对内部定序器进行复位。
4 ～ 2	—	均为 0	R	保留位 读写的值总是 0 。
1、 0	DATS[1:0]	00	R/W	选择发送 / 接收数据长度 选择串行数据的长度。 00：8 位数据长度 01：16 位数据长度 10：32 位数据长度 11：设定无效

16.3.3　SS 模式寄存器 （SSMR）

SSMR 选择 MSB 优先 /LSB 优先、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	选择 MSB 优先 /LSB 优先 选择是以 MSB 先还是以 LSB 先进行串行数据的传送。 0：LSB 优先 1：MSB 优先
6	CPOS	0	R/W	选择时钟极性 选择 SSCK 时钟的极性。 0：空闲时为 High 电平输出，有效时为 Low 电平输出 1：空闲时为 Low 电平输出，有效时为 High 电平输出
5	CPHS	0	R/W	选择时钟相位 （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，锁存数据
4、3	—	均为 0	R	保留位 读写的值总是 0 。
2 ～ 0	CKS[2:0]	000	R/W	选择传送时钟速率 在选择内部时钟时，选择传送时钟速率 （预分频器的分频比）。 000：保留 001：Pϕ/4 010：Pϕ/8 011：Pϕ/16 100：Pϕ/32 101：Pϕ/64 110：Pϕ/128 111：Pϕ/256

16.3.4　SS 允许寄存器 （SSER）

SSER 设定发送允许、接收允许以及中断请求允许。

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	发送允许 将此位置 1 时，能进行发送。
6	RE	0	R/W	允许接收 将此位置 1 时，能进行接收。
5、4	—	均为 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 将此位置 1 时，允许发送结束时的 SSTXI 中断请求。
2	TIE	0	R/W	发送中断的允许 将此位置 1 时，允许发送数据空时的 SSTXI 中断请求。
1	RIE	0	R/W	接收中断的允许 将此位置 1 时，允许 SSRXI 中断和溢出错误时的 SSERI 中断请求。
0	CEIE	0	R/W	冲突错误中断的允许 将此位置 1 时，允许冲突错误时的 SSERI 中断请求。

16.3.5　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，下一个串行接收的 1 字节数据结束时 ［清除条件］ • 读 1 的状态后写 0 时
5、4	—	均为 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］ • SSCR2 的 TENDSTS 是 0 并且在 TDRE=1 的状态下，对发送数据的最后位进行发送时 • SSCR2 的 TENDSTS 是 1 并且在 TDRE=1 的状态下，对发送数据的最后位进行发送后 ［清除条件］ • 读 TEND=1 的状态后，给 TEND 标志写 0 时 • 将数据写入 SSTDR 时

位	位名	初始值	R/W	说明
2	TDRE	1	R/W	发送数据空 表示 SSTDR 内有无数据。 ［置位条件］ • SSER 的 TE 是 0 时 • 将数据从 SSTDR 传送到 SSTRSR 并且能将数据写入 SSTDR 时 ［清除条件］ • 读 TDRE=1 的状态后，给 TDRE 标志写 0 时 • TE=1 并且将数据写入 SSTDR 时 • 通过 SSTXI 中断启动 DMAC，通过 DMAC 传送将传送数据写入 SSTDR 时
1	RDRF	0	R/W	接收数据满 表示 SSRDR 内有无数据。 ［置位条件］ • 串行接收正常结束并且将接收数据从 SSTRSR 传送到 SSRDR 时 ［清除条件］ • 读 RDRF=1 的状态后，给 RDRF 标志写 0 时 • 从 SSRDR 读接收数据时 • 通过 SSRXI 中断启动 DMAC，通过 DMAC 传送从 SSRDR 读接收数据时
0	CE	0	R/W	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 $\overline{\text{SCS}}$ 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 $\overline{\text{SCS}}$ 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。请在重新开始传送前，将 SSCRL 的 SRES 置 1，对内部定序器进行复位。 ［置位条件］ • 为主器件 （SSCRH 的 MSS=1）并且给 SCS 引脚输入 Low 电平时 • 为从属器件 （SSCRH 的 MSS=0）并且在传送中 SCS 引脚变为 1 时 ［清除条件］ • 读到 1 的状态后写 0 时

16.3.6　SS 控制寄存器 2 （SSCR2）

SSCR2 是设定 $\overline{\text{SCS}}$ 引脚的断言时序、 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
	-	-	-	TENDSTS	SCSATS	SSODTS	-	-
初始值：	0	0	0	0	0	0	0	0
R/W：	R	R	R	R/W	R/W	R/W	R	R

位	位名	初始值	R/W	说明
7 ～ 5	—	均为 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：发送最后位时，置位 TEND 位 1：发送最后位后，置位 TEND 位
3	SCSATS	0	R/W	选择 $\overline{\text{SCS}}$ 引脚的断言时序 （只在设定 SSU 模式和主模式时有效） 0：t_{LEAD} 和 t_{LAG} 的输出期间 Min. 为 1/2×tSUcyc 1：t_{LEAD} 和 t_{LAG} 的输出期间 Min. 为 3/2×tSUcyc
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	—	均为 0	R	保留位 读写的值总是 0 。

16.3.7　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。通过设定 SSCRL 的 DATS1 位和 DATS0 位来选择 8 位数据长度时，SSTDR0 有效；选择 16 位数据长度时，SSTDR0 和 SSTDR1 有效；选择 32 位数据长度时，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入 SSTDR，就能进行连续串行发送。

CPU 和 DMAC 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：	7	6	5	4	3	2	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 ～ 0		均为 0	R/W	串行发送数据

表 16.3　DATS 位的设定和 SSTDR 的对应表

SSTDR	DATS[1:0] （SSCRL[1:0]）			
	00	01	10	11 （设定无效）
0	有效	有效	有效	无效
1	无效	有效	有效	无效
2	无效	无效	有效	无效
3	无效	无效	有效	无效

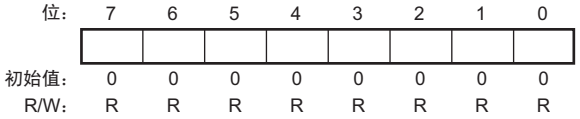
16.3.8　SS 接收数据寄存器 0 ～ 3 （SSRDR0 ～ SSRDR3）

SSRDR 是保存接收数据的 8 位寄存器。通过设定 SSCRL 的 DATS1 位和 DATS0 位来选择 8 位数据长度时，SSRDR0 有效；当选择 16 位数据长时，SSRDR0 和 SSRDR1 有效；当选择 32 位数据长度时，SSRDR0、SSRDR1、SSRDR2 和 SSRDR3 有效。不能存取无效的 SSRDR。

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。

请在确认 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	位名	初始值	R/W	说明
7 ～ 0		均为 0	R	串行接收数据

表 16.4　DATS 位的设定和 SSRDR 的对应表

SSRDR	DATS[1:0] （SSCRL[1:0]）			
	00	01	10	11 （设定无效）
0	有效	有效	有效	无效
1	无效	有效	有效	无效
2	无效	无效	有效	无效
3	无效	无效	有效	无效

16.3.9　SS 移位寄存器 （SSTRSR）

SSTRSR 是发送和接收串行数据的移位寄存器。

将发送数据从 SSTDR 传送到 SSTRSR 时，如果 SSMR 的 MLS=0，就将 SSTDR 的 bit0 （LSB 优先通信）传送到 SSTRSR 的 bit0；如果 MLS=1，就将 SSTDR 的 bit7（MSB 优先通信）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，并进行串行数据的发送。

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。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：	-	-	-	-	-	-	-	-

16.4　运行说明

16.4.1　传送时钟

可从 7 种内部时钟和外部时钟中选择传送时钟。使用本模块时，需要通过 PFC 预先将 SSCK 引脚设定为有效。SSCRH 的 MSS=1 时，选择内部时钟并且 SSCK 引脚为输出引脚，开始传送时，就从 SSCK 引脚输出 SSMR 的 CKS2 ～ CKS0 所设定的传送率的时钟；在 MSS=0 时，选择外部时钟并且 SSCK 引脚为输入引脚。

16.4.2　时钟的相位、极性和数据的关系

SSCRL 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 16.2 所示。当 SSUMS=1 时， CPOS 的设定有效，但是 CPHS 的设定无效。

另外，能通过设定 SSMR 的 MLS，选择是以 MSB 优先还是以 LSB 优先进行传送； MLS=0 时，按照从 LSB 到 MSB 的顺序进行传送； MLS=1 时，按照从 MSB 到 LSB 的顺序进行传送。

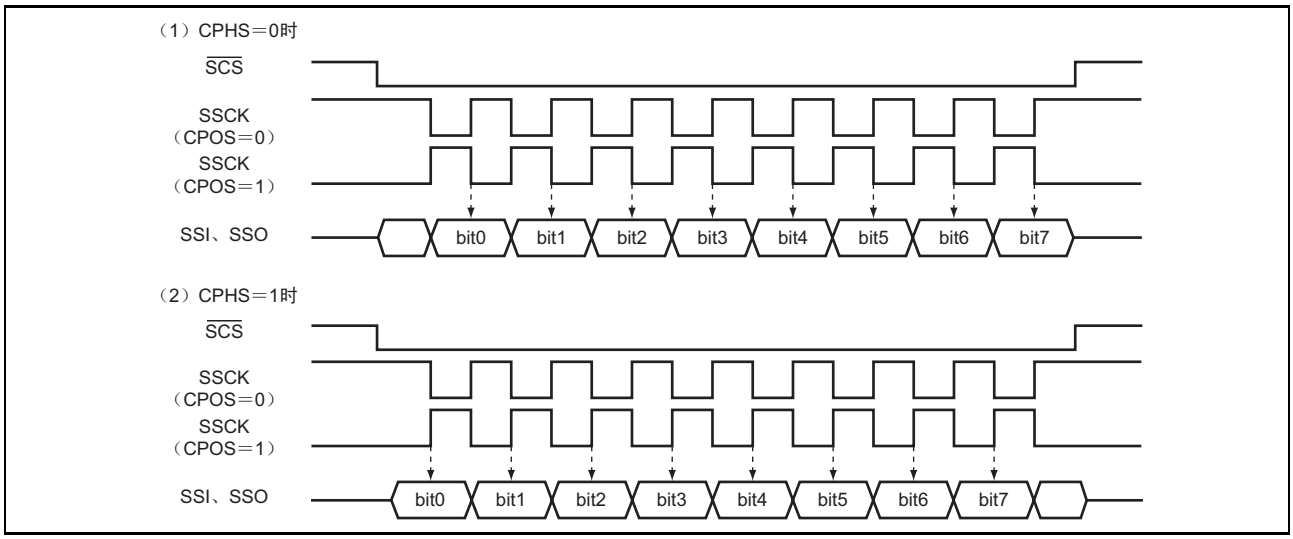


图 16.2　时钟的相位、极性和数据的关系

16.4.3　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 16.3 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 16.3 （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 16.3 （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 16.3 （3）、图 16.3 （4））。

但是不能同时将 TE 和 RE 置 1 并进行同时发送和接收，必须选择 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。MSS=1 时，SSU 就从 SSCK 引脚输出内部时钟； MSS=0 时， SSCK 引脚为输入引脚 （图 16.3 （5）、图 16.3 （6））。

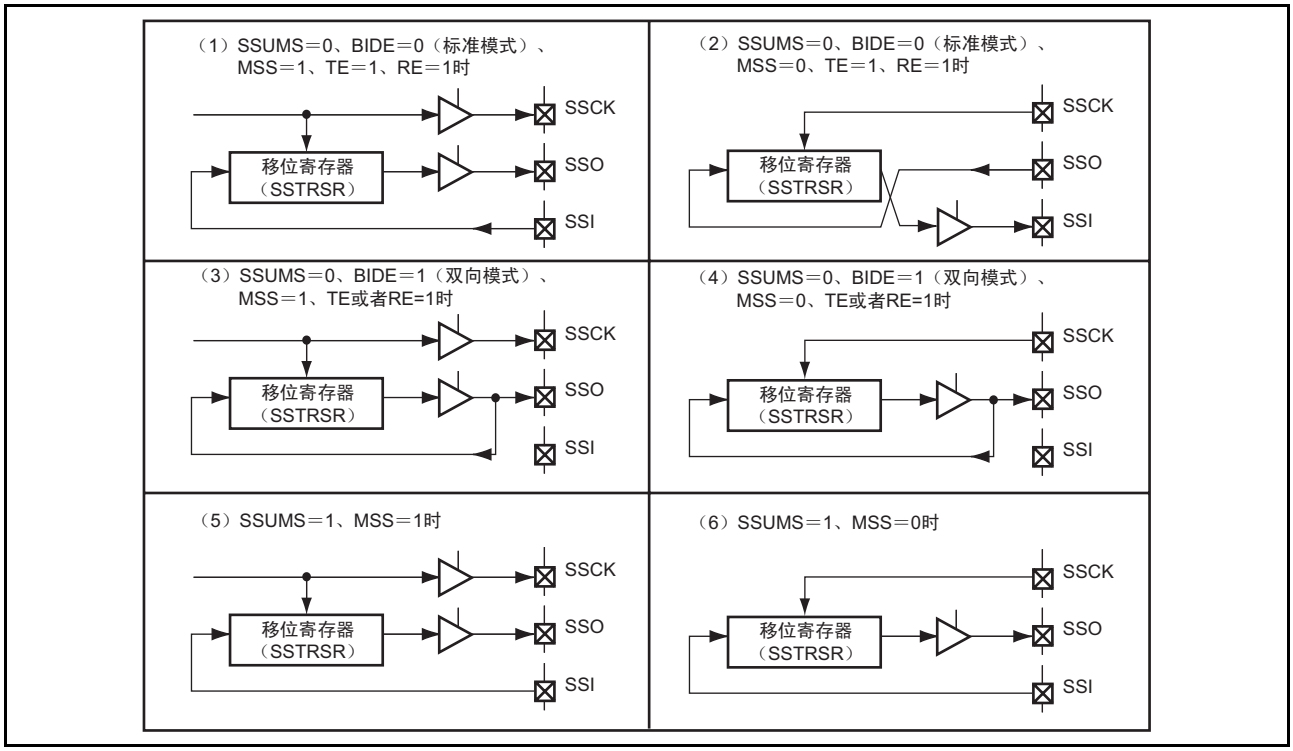


图 16.3　数据输入 / 输出引脚和移位寄存器的关系

16.4.4　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。各通信模式和输入 / 输出引脚的关系如表 16.5 ～表 16.7 所示。

表 16.5　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	输出	输入
			1	0	1	输入	—
				1	0	—	输出
					1	输入	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	输入	输出
			1	0	1	输入	—
				1	0	—	输出
					1	输入	输出

【符号说明】 —：不用作 SSU 引脚 （能用作 I/O 端口）

表 16.6　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

【符号说明】 —：不用作 SSU 引脚 （能用作 I/O 端口）

表 16.7　各通信模式和 $\overline{SCS}$ 引脚的状态

通信模式	寄存器状态				引脚状态
	SSUMS	MSS	CSS1	CSS0	$\overline{SCS}$
SSU 通信模式	0	0	x	x	输入
		1	0	0	—
			0	1	输入
			1	0	自动输入 / 输出
			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 （能用作 I/O 端口）

16.4.5　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 （$\overline{SCS}$）的 4 个总线进行数据传送。

另外，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 模式的初始设定例如图 16.4 所示。在发送或者接收数据前，必须将 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。

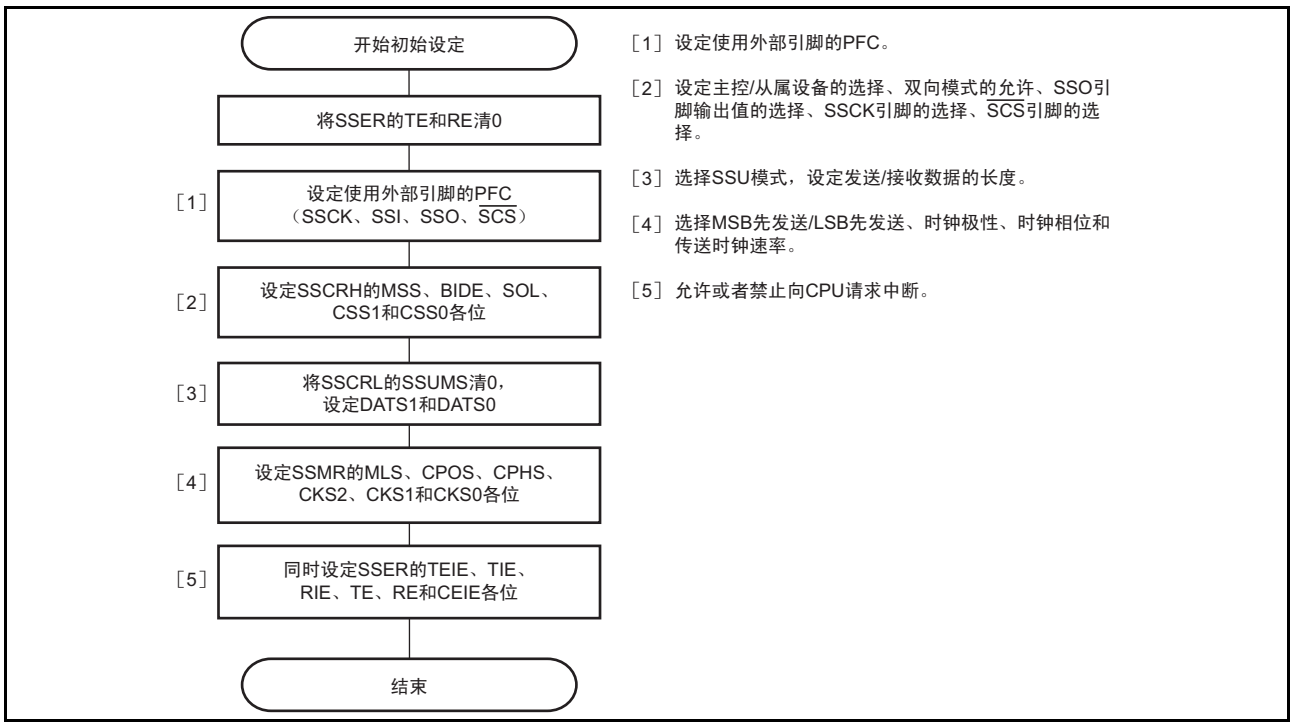


图 16.4　SSU 模式的初始设定例

（2） 发送数据

发送时的运行例和数据发送的流程图例分别如图 16.5 和图 16.6 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，如果将发送数据写入 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 并开始发送。此时，如果 SSER 的 TIE 已被置 1，就发生 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， SSMR 的 CPOS=0 时， SSCK 引脚固定为高电平； CPOS=1 时， SSCK 引脚固定为低电平。

另外，不能在 SSSR 的 ORER 被置 1 的状态下发送数据，请在发送前确认 ORER 是否已被清 0。

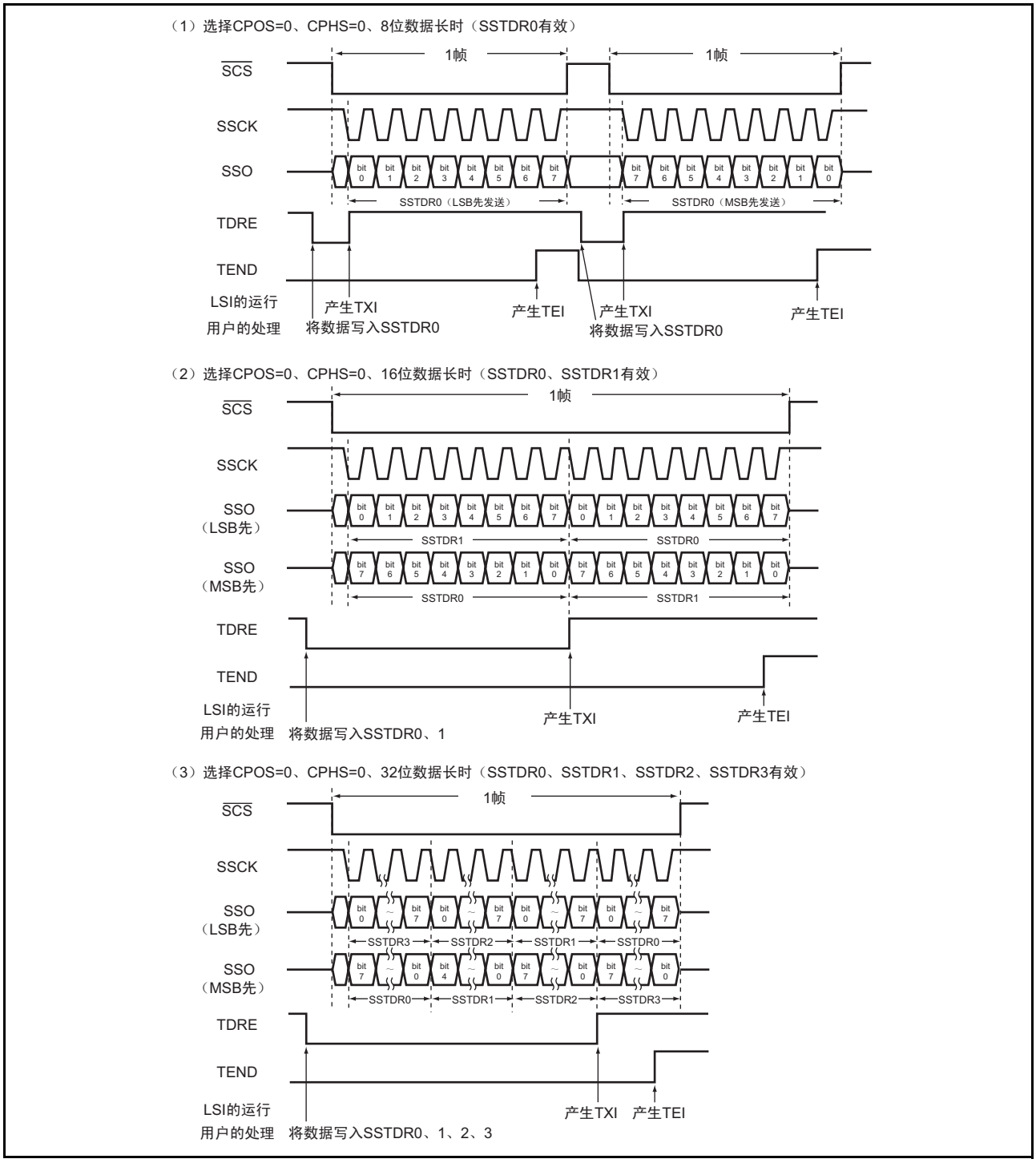


图 16.5　发送时的运行例 （SSU 模式）

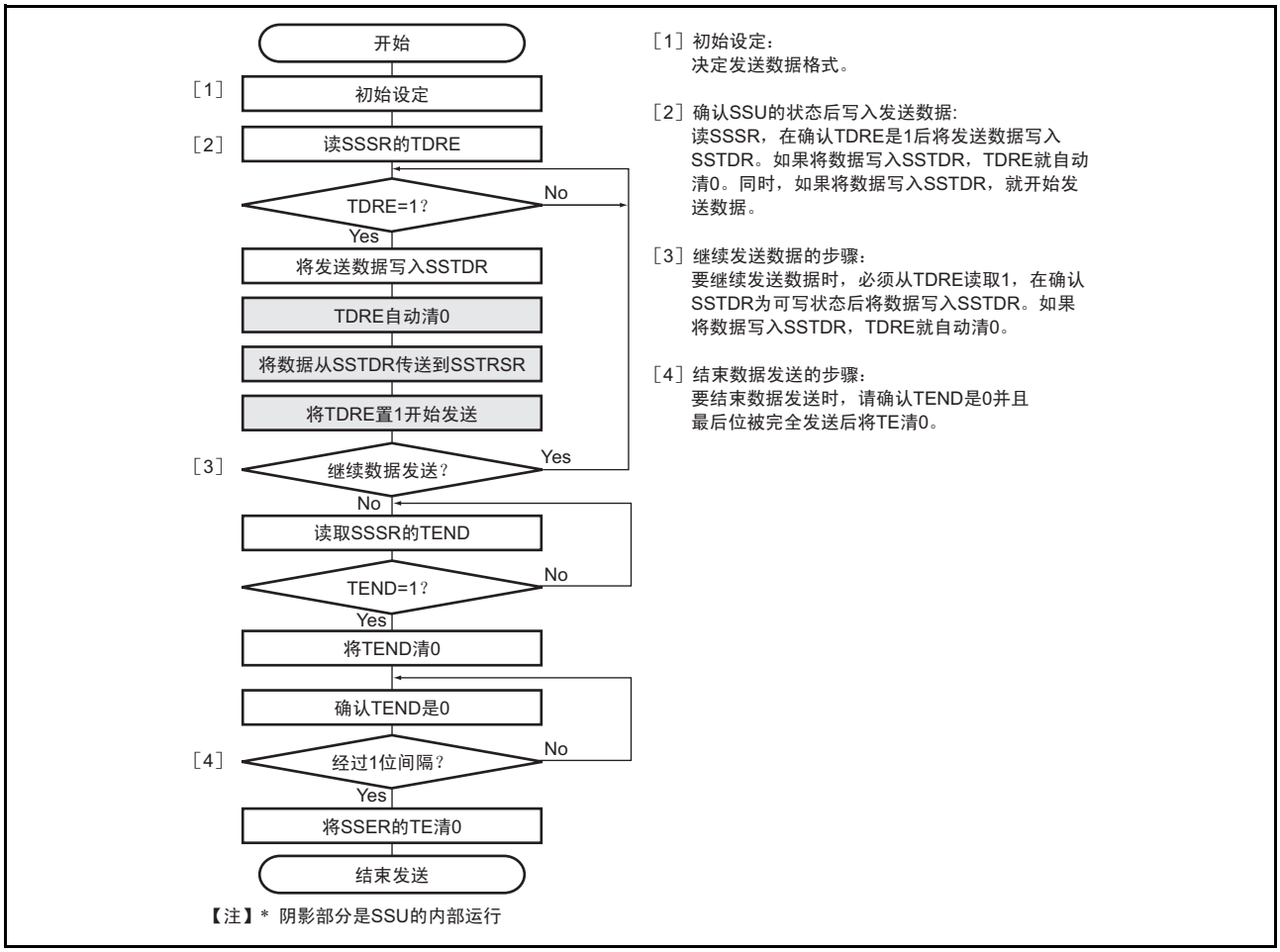


图 16.6　数据发送的流程图例 （SSU 模式）

（3） 接收数据

接收时的运行例和数据接收的流程图例分别如**图 16.7** 和**图 16.8** 所示。在接收数据时， SSU 的运行如下：

将 SSER 的 RE 置 1 并虚读 SSRDR 时， SSU 开始接收运行。

将 SSU 设定为主器件时，输出传送时钟并且输入接收数据；设定为从属器件时，如果给 $\overline{\text{SCS}}$ 引脚输入低电平并且从 SSCK 引脚输入传送时钟， SSU 就和此传送时钟同步输入接收数据。

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，所以请在重新开始接收时，将 ORER 清 0。

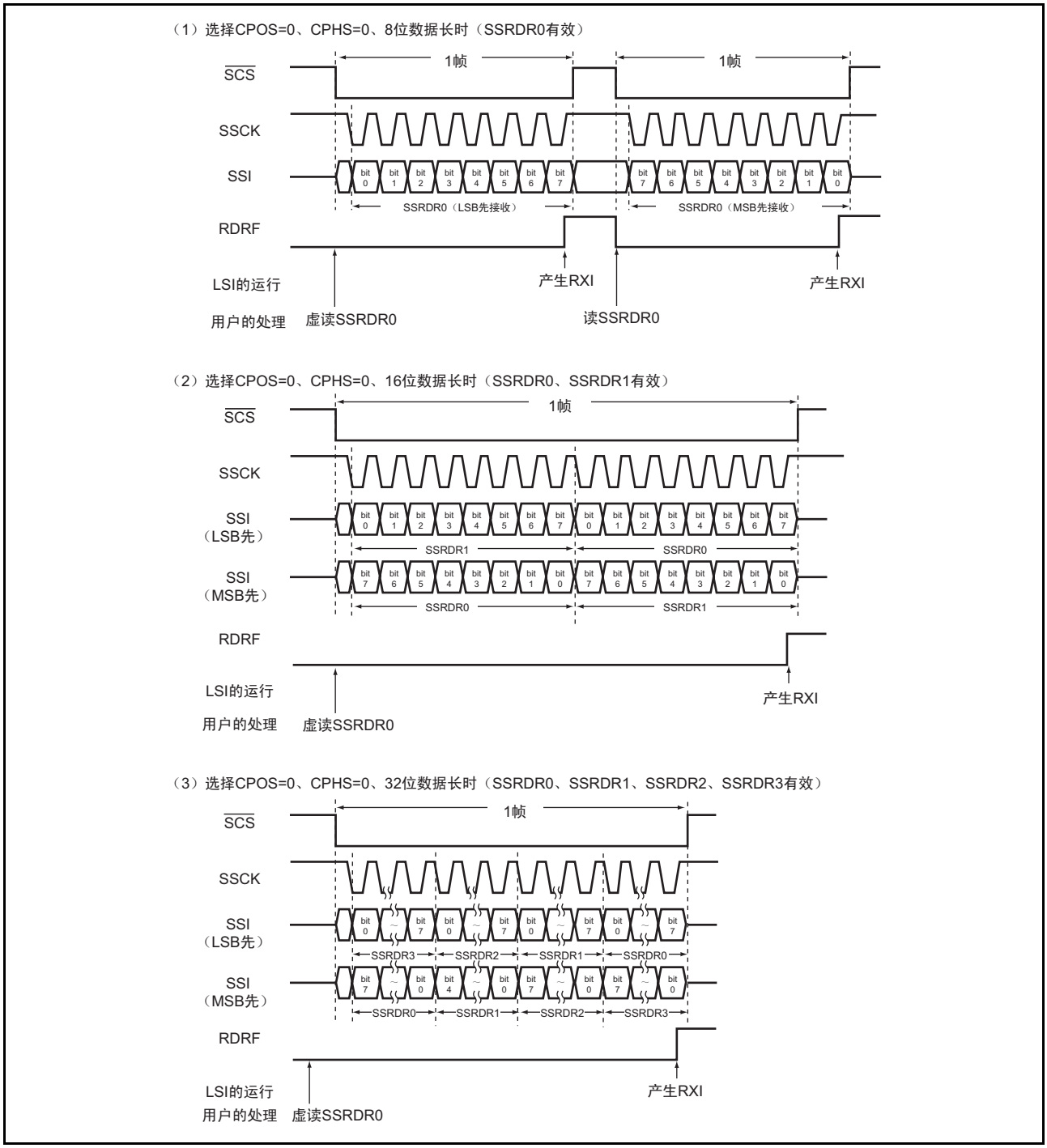


图 16.7　接收时的运行例 （SSU 模式）

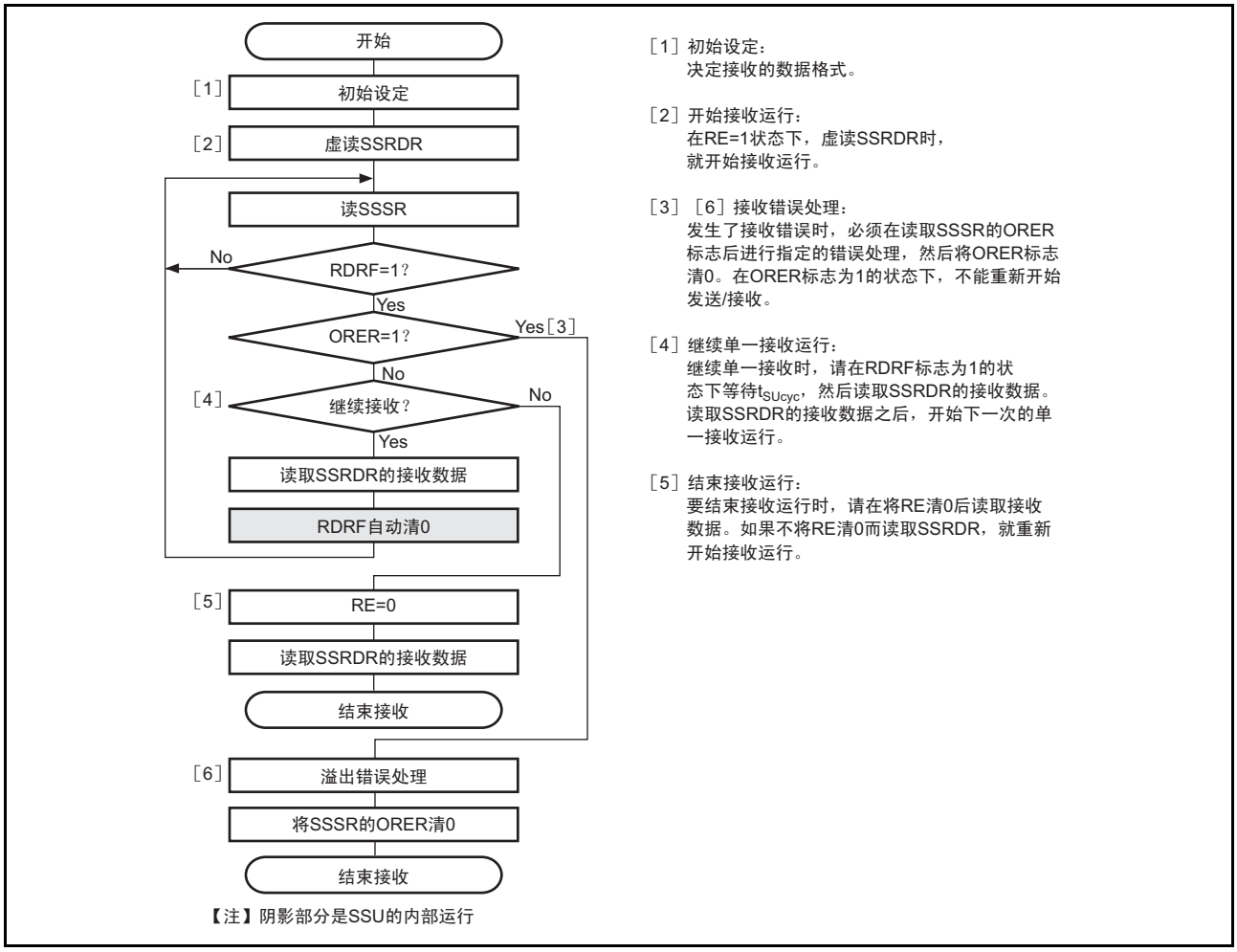


图 16.8　数据接收的流程图例 （SSU 模式）

（4） 发送 / 接收数据

数据同时发送 / 接收的流程图例如图 16.9 所示，数据的发送 / 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，通过将发送数据写入 SSTDR，开始发送 / 接收数据。

另外，在从发送模式 （TE=1）或者接收模式 （RE=1）转换为发送 / 接收模式 （TE=RE=1）时，请将 TE 和 RE 清 0 后进行转换。同时，请确认 TEND、 RDRF 和 ORER 已被清 0 后将 TE 和 RE 置 1。

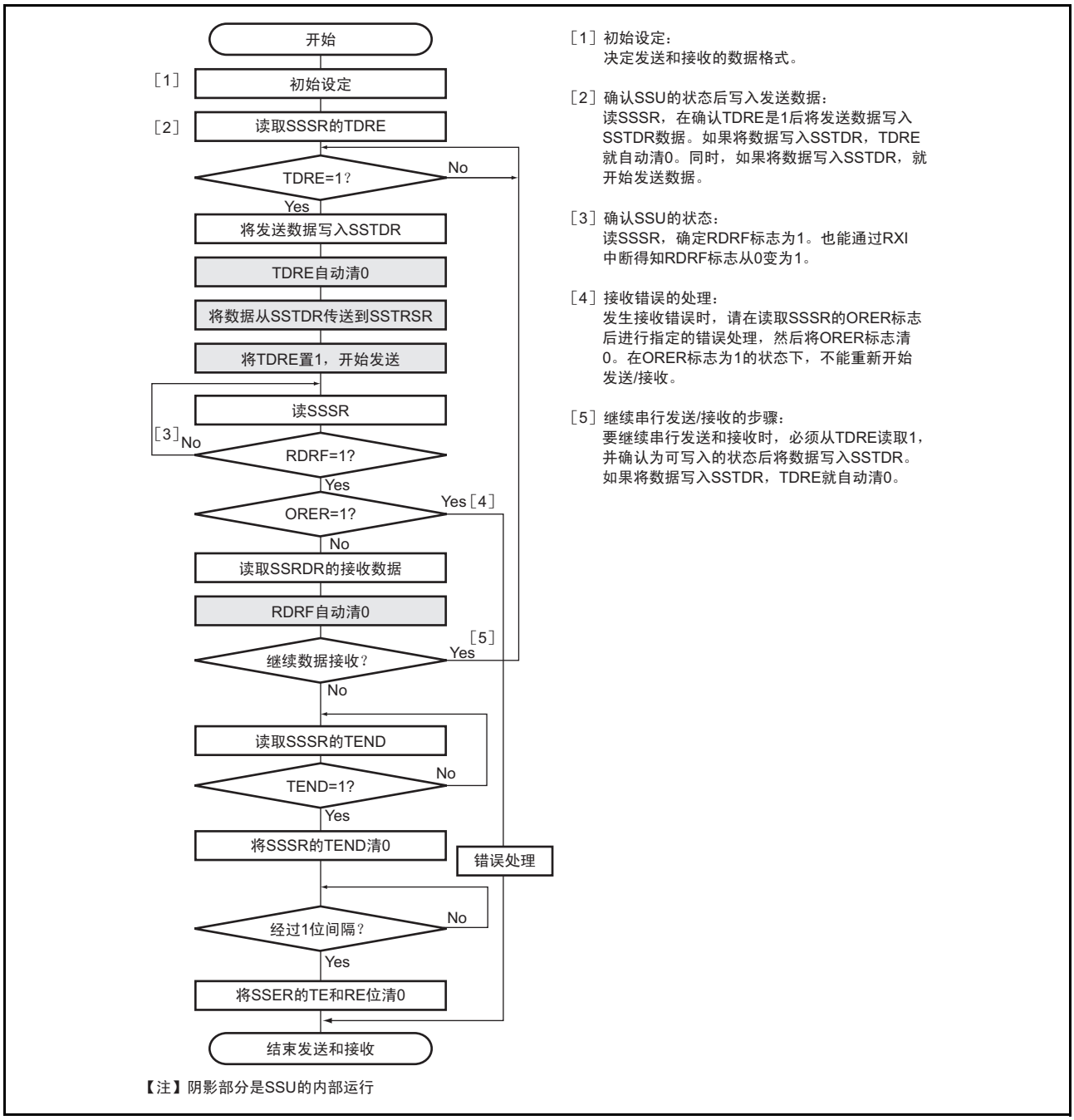


图 16.9　数据同时发送 / 接收的流程图例 （SSU 模式）

16.4.6　$\overline{\text{SCS}}$ 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，$\overline{\text{SCS}}$ 引脚为输入引脚 （Hi-Z），并且检测冲突错误。在此期间，如果从 $\overline{\text{SCS}}$ 引脚输入低电平，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 / 接收。在开始发送 / 接收前，必须将 CE 清 0。

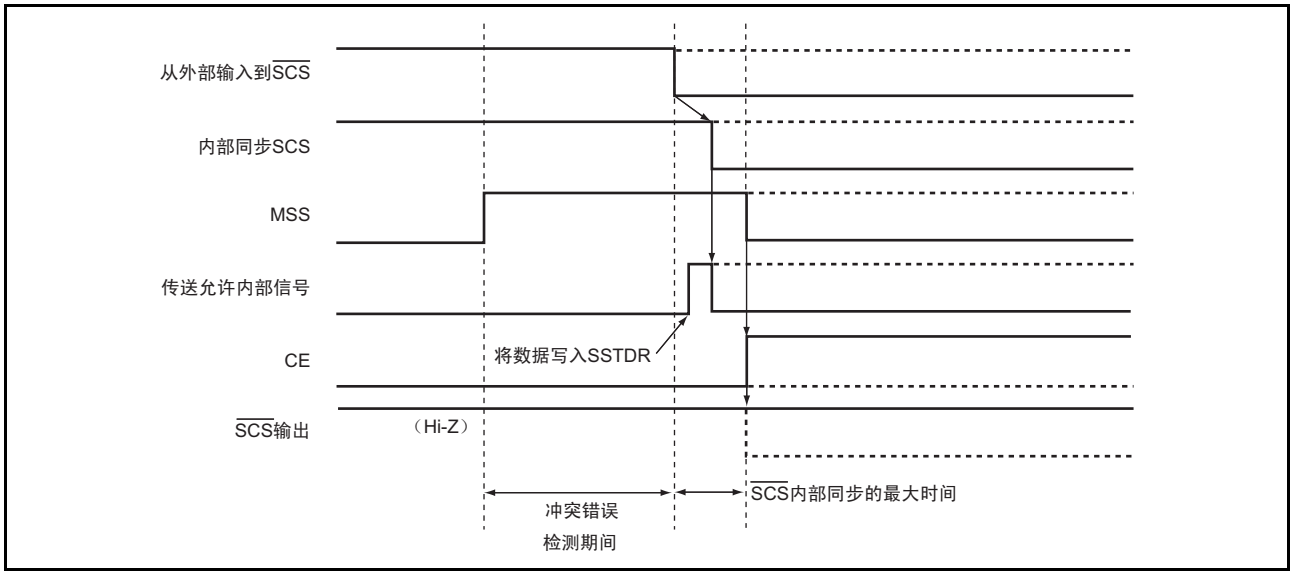


图 16.10　冲突错误的检测时序 （传送前）

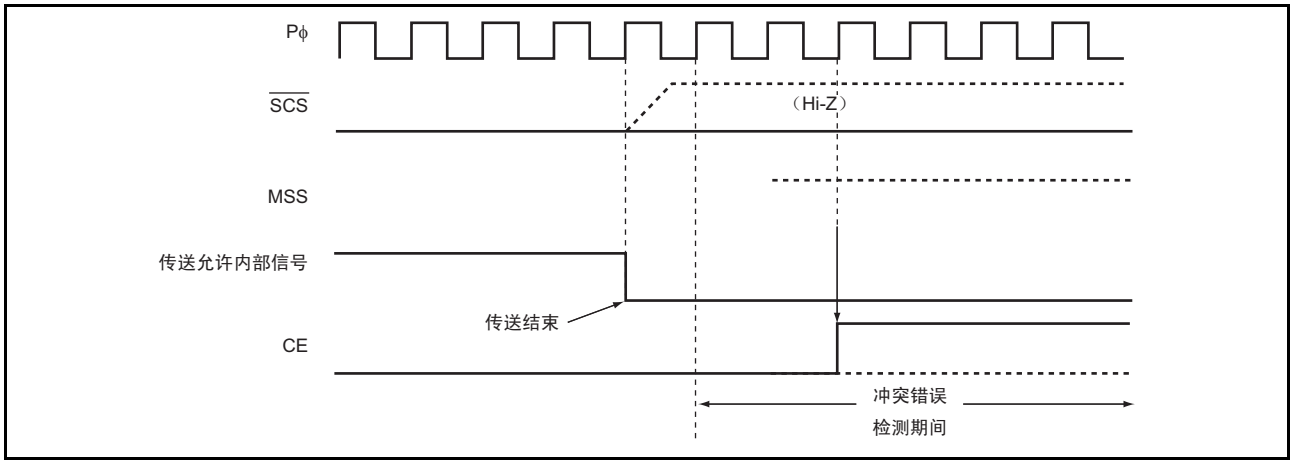


图 16.11　冲突错误的检测时序 （传送结束后）

16.4.7　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） 3 个总线进行数据传送。

（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如图 16.12 所示，在发送 / 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，就必须将 TE 和 RE 清 0 后进行更改。请注意：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。

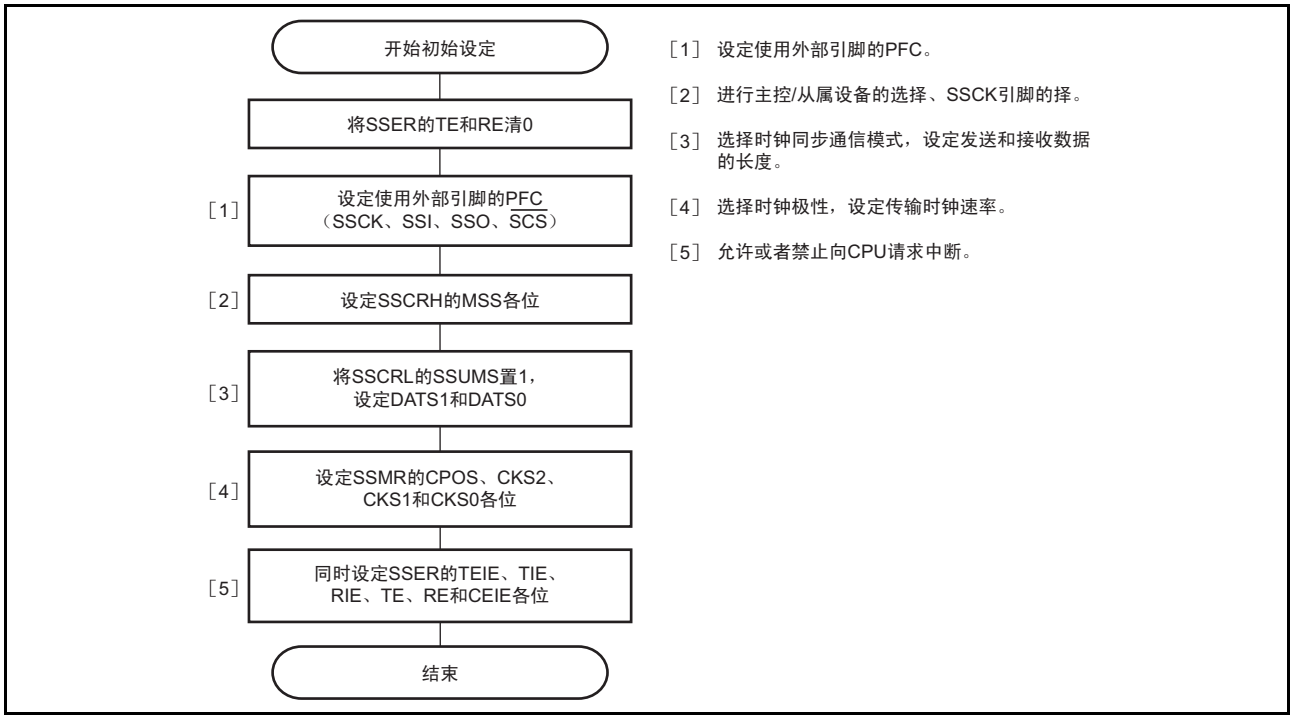


图 16.12　时钟同步通信模式的初始设定例

（2） 发送数据

发送时的运行例和发送数据的流程图例分别如图 16.13 和图 16.14 所示。在时钟同步通信模式中发送数据时， SSU 的运行如下：

将 SSU 设定为主器件时，输出传送时钟和数据； 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，如果将发送数据写入 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 并开始发送。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。

另外，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。

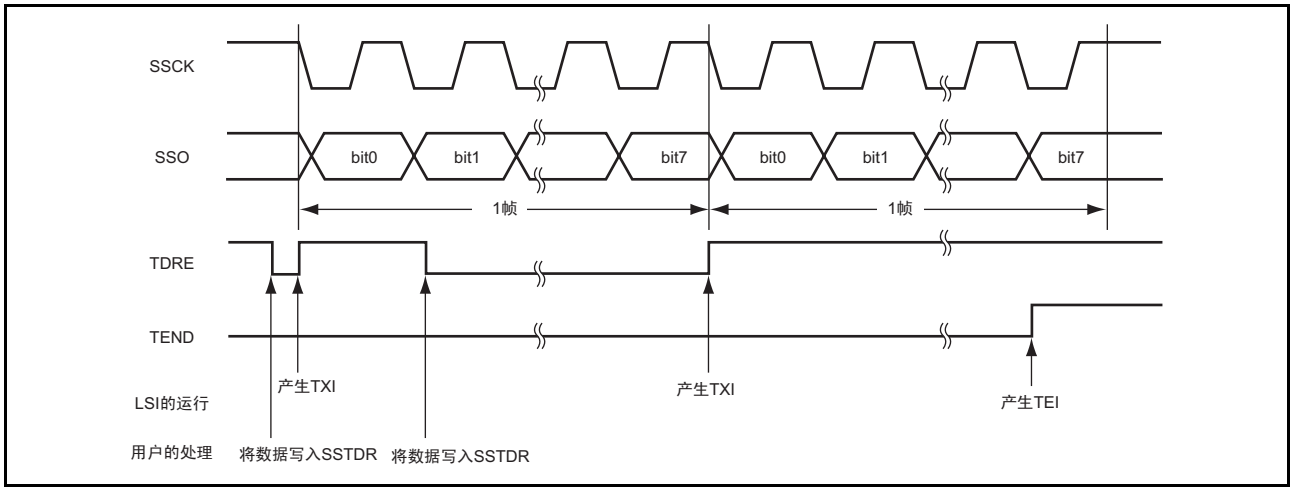


图 16.13　发送时的运行例 （时钟同步通信模式）

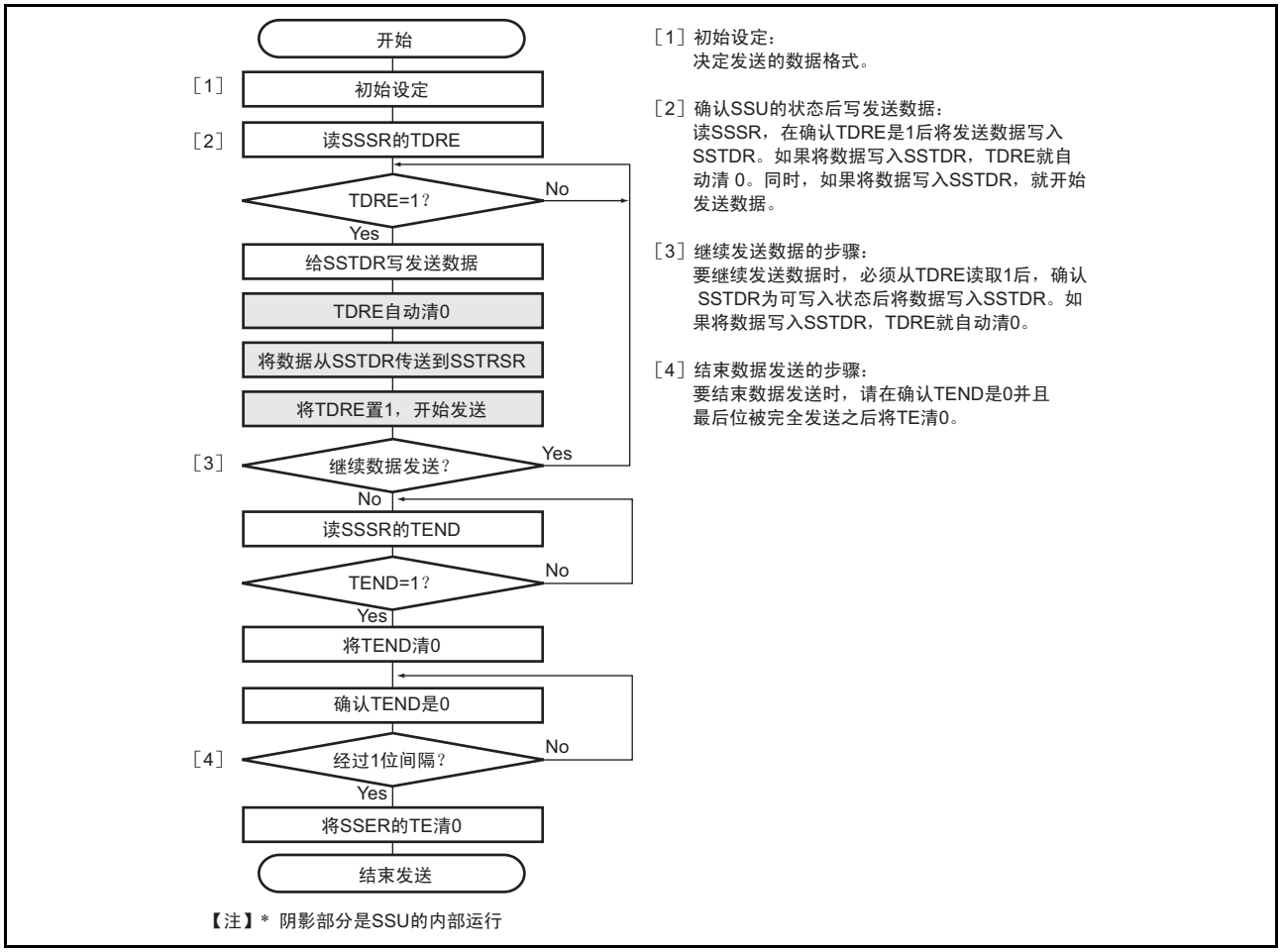


图 16.14　发送数据的流程图例 （时钟同步通信模式）

（3） 接收数据

接收时的运行例和接收数据的流程图例分别如图 16.15 和图 16.16 所示。在接收数据时，SSU 的运行如下：

将 SSER 的 RE 置 1 时， SSU 开始接收运行。

将 SSU 设定为主器件时，输出传送时钟并且输入接收数据；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，所以请在重新开始接收时，将 ORER 清 0。

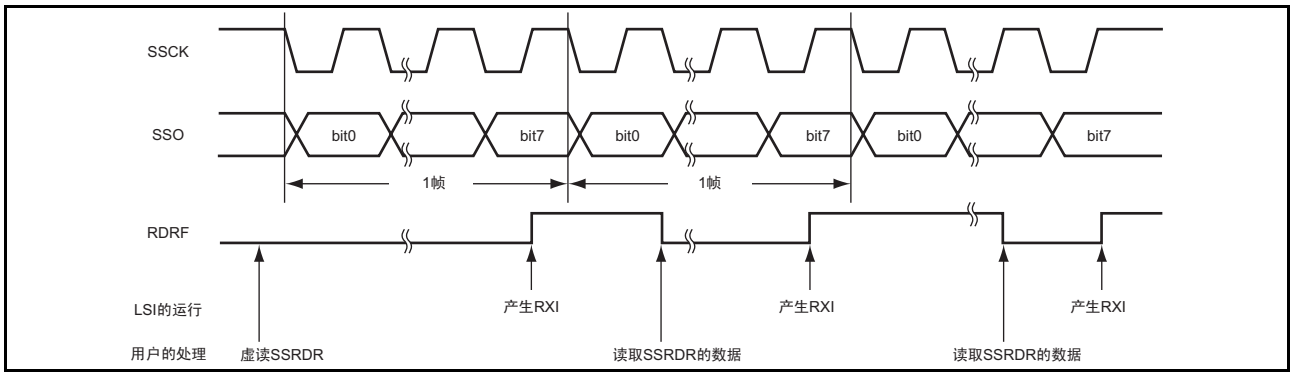


图 16.15　接收时的运行例 （时钟同步通信模式）

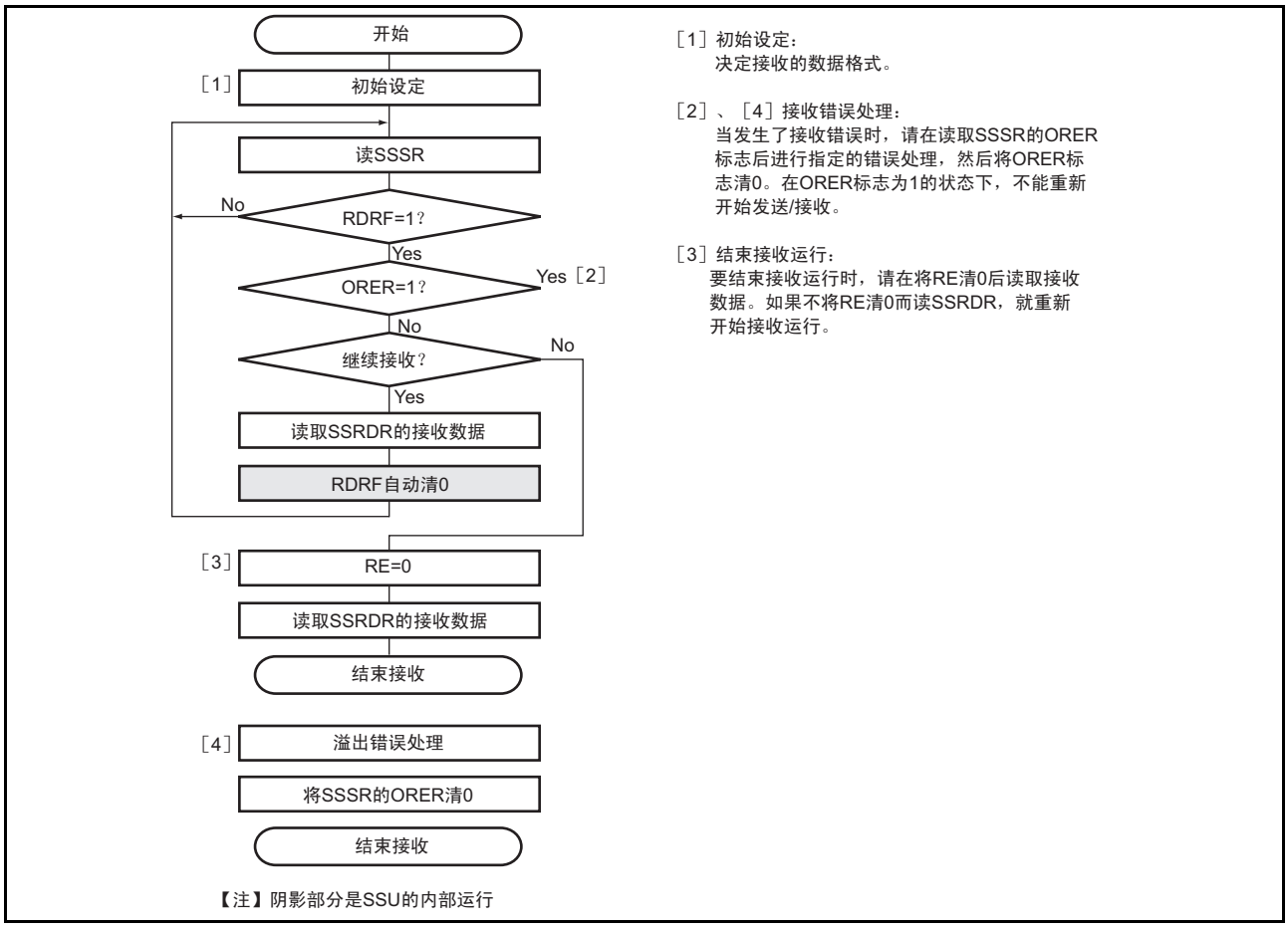


图 16.16　接收数据的流程图例 （时钟同步通信模式）

（4） 发送 / 接收数据

同时发送 / 接收的流程图例如图 16.17 所示，数据的发送 / 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，通过将发送数据写入 SSTDR，开始发送 / 接收数据。

另外，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，请在将 TE 和 RE 清 0 后进行转换。同时，请在确认 TEND、 RDRF 和 ORER 已被清 0 后将 TE 和 RE 置 1。

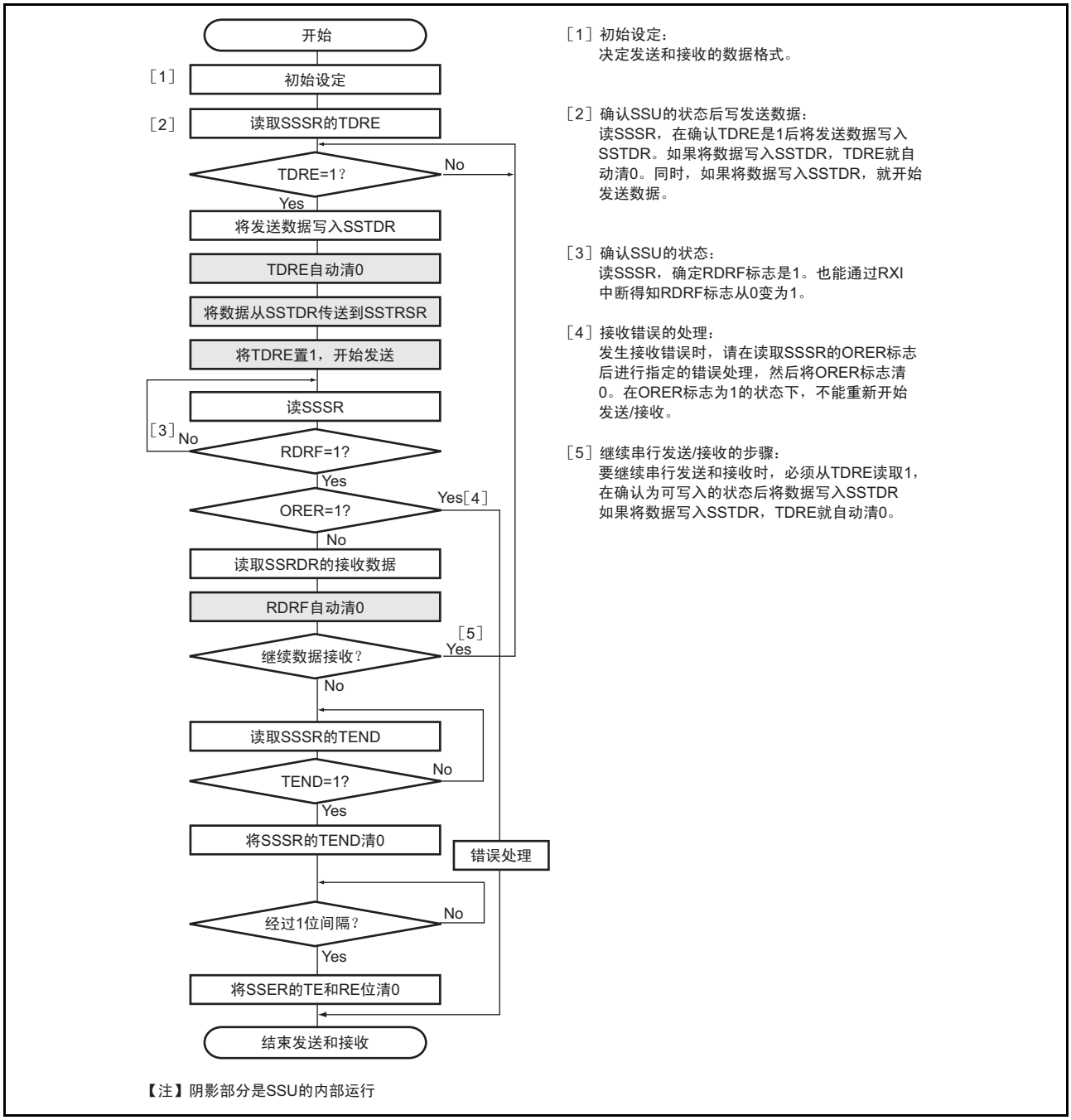


图 16.17　数据同时发送 / 接收的流程图例 （时钟同步通信模式）

16.5　SSU 的中断源和 DMAC

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 DMAC 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 16.8** 所示。

当**表 16.8** 的中断条件成立时，就产生中断请求。请通过 CPU 或者 DMAC 的数据传送来清除中断源。

表 16.8　SSU 中断源

名称	中断源	中断条件	DMAC 的启动
SSERI	溢出错误	（RIE=1）•（ORER=1）+（CEIE=1）•（CE=1）	—
	冲突错误		
SSRXI	接收数据满	（RIE=1）•（RDRF=1）	○
SSTXI	发送数据空	（TIE=1）•（TDRE=1）+（TEIE=1）•（TEND=1）	○
	发送结束		

16.6　使用时的注意事项

16.6.1　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 32 章　低功耗模式**”。

第 17 章　I²C 总线接口 3 （IIC3）

I²C 总线接口 3 依据飞利浦公司提倡的 I²C 总线 （Inter IC Bus）接口方式，配备了子集功能。但需要注意的是，控制 I²C 总线的寄存器结构与飞利浦斯公司有些部分不同。

另外，有 4 个通道。

17.1　特点

- 可以选择I²C总线格式或者时钟同步串行格式
- 可以连续发送/接收
  由于移位寄存器、发送数据寄存器和接收数据寄存器各自独立，因此能连续发送/接收。

I²C 总线格式

- 在主模式中，自动生成开始条件和停止条件
- 接收时，可以选择应答的输出电平
- 发送时，自动装入应答位
- 内置位同步功能
  在主模式中，按位监控SCL的状态，自动取得同步。在完成传送准备前，置SCL为低电平，使SCL处于待机状态。
- 中断源：6种
  发送数据空 （包含从属地址匹配时）、发送结束、接收数据满 （包含从属地址匹配时）、仲裁失败、检测出NACK、检测出停止条件。
- 能通过发送数据空中断和接收数据满中断，启动直接存储器存取控制器 （DMAC）进行数据的传送。
- 能直接驱动总线
  在选择总线驱动功能时， SCL和SDA的2个引脚为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
- 能通过发送数据空中断和接收数据满中断，启动直接存储器存取控制器 （DMAC）进行数据的传送。

I²C 总线接口 3 的框图如图 17.1 所示。

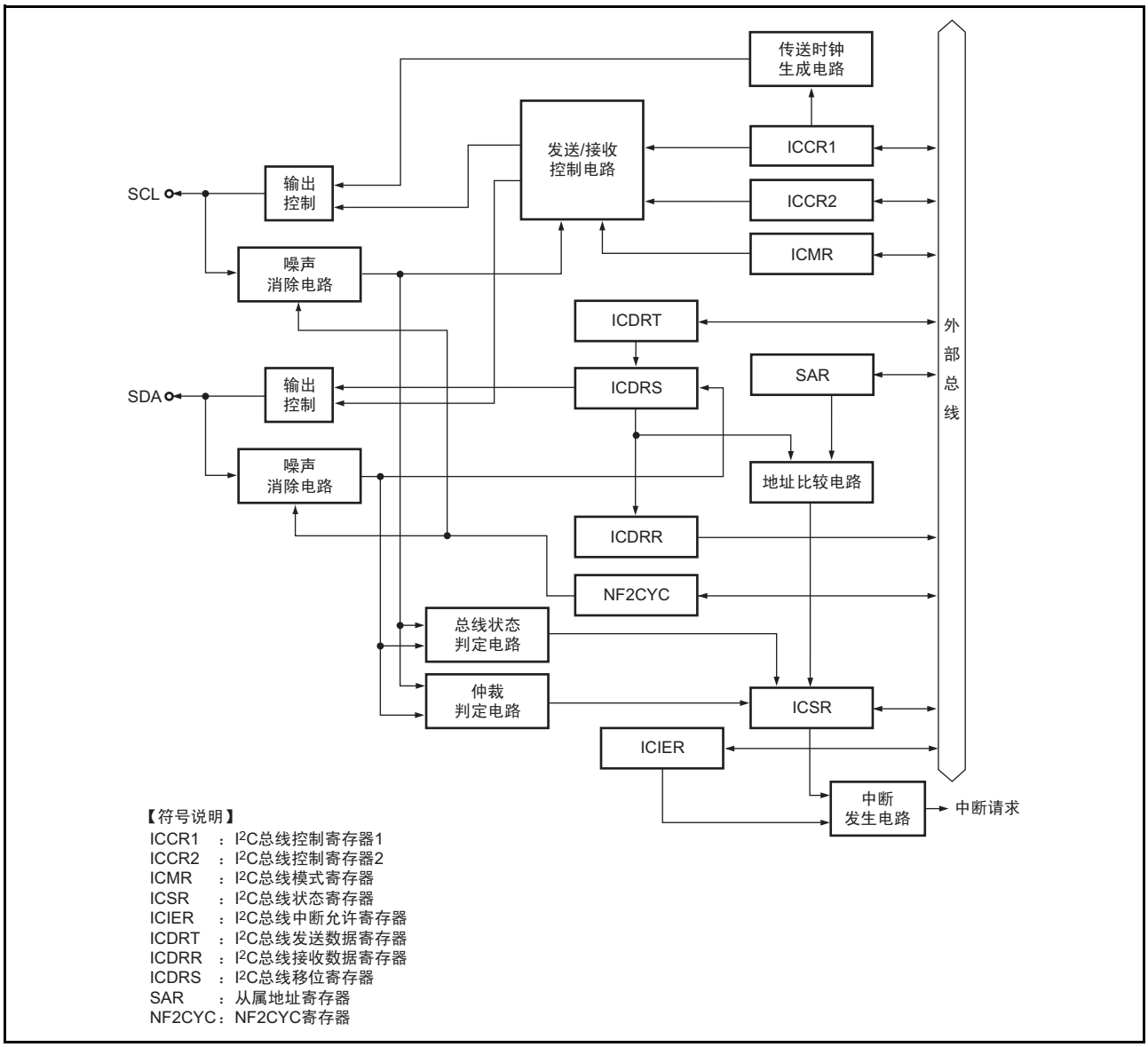


图 17.1　I²C 总线接口 3 的框图

17.2　输入 / 输出引脚

I²C 总线接口 3 使用的引脚结构如表 17.1 所示。

表 17.1　引脚结构

名称	引脚名	输入 / 输出	功能
串行时钟引脚	SCL0 ～ SCL3	输入 / 输出	I²C 串行时钟输入 / 输出引脚
串行数据引脚	SDA0 ～ SDA3	输入 / 输出	I²C 串行数据输入 / 输出引脚

输入 / 输出引脚的外部电路连接例如图 17.2 所示。

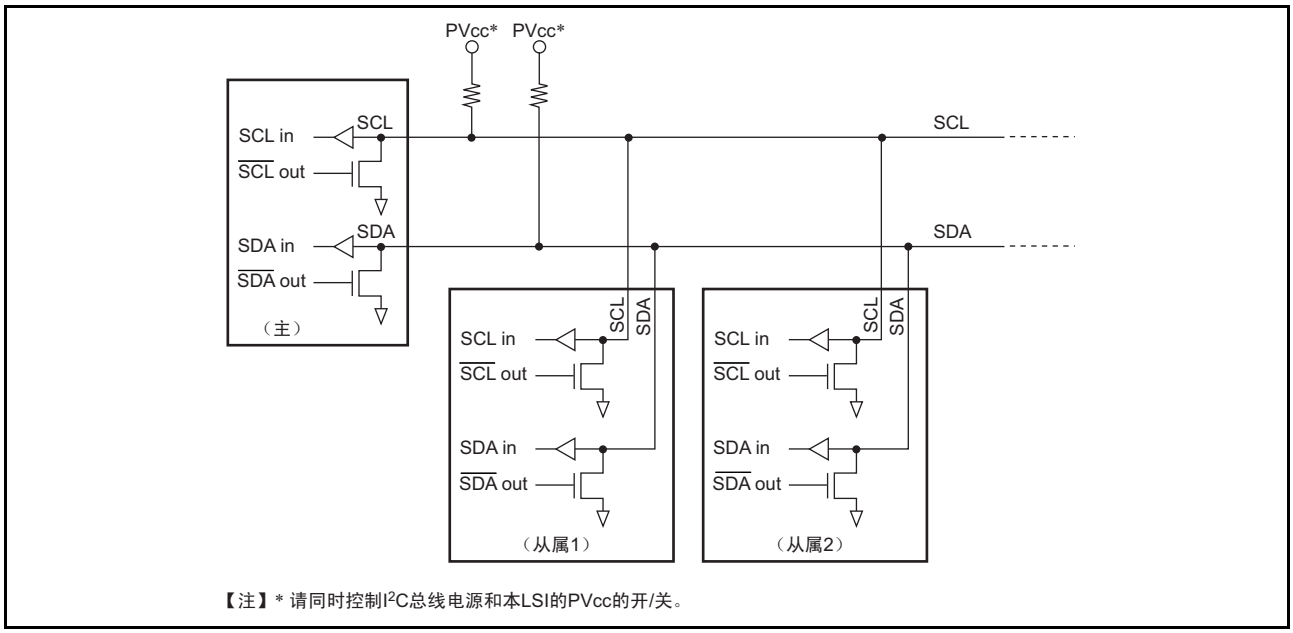


图 17.2　输入 / 输出引脚的外部电路连接例

17.3　寄存器说明

I²C 总线接口 3 有以下寄存器。

表 17.2 寄存器结构

通道	寄存器名	简称	R/W	初始值	地址	存取长度
0	I²C 总线控制寄存器 1	ICCR1_0	R/W	H'00	H'FFFEE000	8
	I²C 总线控制寄存器 2	ICCR2_0	R/W	H'7D	H'FFFEE001	8
	I²C 总线模式寄存器	ICMR_0	R/W	H'38	H'FFFEE002	8
	I²C 总线中断允许寄存器	ICIER_0	R/W	H'00	H'FFFEE003	8
	I²C 总线状态寄存器	ICSR_0	R/W	H'00	H'FFFEE004	8
	从属地址寄存器	SAR_0	R/W	H'00	H'FFFEE005	8
	I²C 总线发送数据寄存器	ICDRT_0	R/W	H'FF	H'FFFEE006	8
	I²C 总线接收数据寄存器	ICDRR_0	R/W	H'FF	H'FFFEE007	8
	NF2CYC 寄存器	NF2CYC_0	R/W	H'00	H'FFFEE008	8
1	I²C 总线控制寄存器 1	ICCR1_1	R/W	H'00	H'FFFEE400	8
	I²C 总线控制寄存器 2	ICCR2_1	R/W	H'7D	H'FFFEE401	8
	I²C 总线模式寄存器	ICMR_1	R/W	H'38	H'FFFEE402	8
	I²C 总线中断允许寄存器	ICIER_1	R/W	H'00	H'FFFEE403	8
	I²C 总线状态寄存器	ICSR_1	R/W	H'00	H'FFFEE404	8
	从属地址寄存器	SAR_1	R/W	H'00	H'FFFEE405	8
	I²C 总线发送数据寄存器	ICDRT_1	R/W	H'FF	H'FFFEE406	8
	I²C 总线接收数据寄存器	ICDRR_1	R/W	H'FF	H'FFFEE407	8
	NF2CYC 寄存器	NF2CYC_1	R/W	H'00	H'FFFEE408	8
2	I²C 总线控制寄存器 1	ICCR1_2	R/W	H'00	H'FFFEE800	8
	I²C 总线控制寄存器 2	ICCR2_2	R/W	H'7D	H'FFFEE801	8
	I²C 总线模式寄存器	ICMR_2	R/W	H'38	H'FFFEE802	8
	I²C 总线中断允许寄存器	ICIER_2	R/W	H'00	H'FFFEE803	8
	I²C 总线状态寄存器	ICSR_2	R/W	H'00	H'FFFEE804	8
	从属地址寄存器	SAR_2	R/W	H'00	H'FFFEE805	8
	I²C 总线发送数据寄存器	ICDRT_2	R/W	H'FF	H'FFFEE806	8
	I²C 总线接收数据寄存器	ICDRR_2	R/W	H'FF	H'FFFEE807	8
	NF2CYC 寄存器	NF2CYC_2	R/W	H'00	H'FFFEE808	8

通道	寄存器名	简称	R/W	初始值	地址	存取长度
3	I²C 总线控制寄存器 1	ICCR1_3	R/W	H'00	H'FFFEEC00	8
	I²C 总线控制寄存器 2	ICCR2_3	R/W	H'7D	H'FFFEEC01	8
	I²C 总线模式寄存器	ICMR_3	R/W	H'38	H'FFFEEC02	8
	I²C 总线中断允许寄存器	ICIER_3	R/W	H'00	H'FFFEEC03	8
	I²C 总线状态寄存器	ICSR_3	R/W	H'00	H'FFFEEC04	8
	从属地址寄存器	SAR_3	R/W	H'00	H'FFFEEC05	8
	I²C 总线发送数据寄存器	ICDRT_3	R/W	H'FF	H'FFFEEC06	8
	I²C 总线接收数据寄存器	ICDRR_3	R/W	H'FF	H'FFFEEC07	8
	NF2CYC 寄存器	NF2CYC_3	R/W	H'00	H'FFFEEC08	8

17.3.1　I²C 总线控制寄存器 1 （ICCR1）

ICCR1 是 8 位可读 / 写寄存器，选择 I²C 总线接口 3 的运行 / 停止、发送 / 接收控制、主模式 / 从属模式、发送 / 接收以及主模式传送时钟频率。

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	I²C 总线接口 3 允许 0：本模块为功能停止状态 1：本模块为传送运行可能状态 （SCL/SDA 为总线驱动状态）
6	RCVD	0	R/W	接收禁止 在 TRS=0 的状态下，读取 ICDRR 时，设定继续 / 禁止下一次运行。 0：继续下一次的接收运行 1：禁止下一次的接收运行

位	位名	初始值	R/W	说明
5 4	MST TRS	0 0	R/W R/W	主 / 从属选择 发送 / 接收选择 在 I²C 总线格式的主模式，如果总线竞争失败，MST 和 TRS 就被硬件复位，变为从属接收模式。请在传送帧之间改变 TRS。 另外，在从属接收模式，如果开始条件后的 7 位与设定在 SAR 的从属地址匹配，并且第 8 位是 1， TRS 就被自动置 1。在时钟同步串行格式的主接收模式，如果发生溢出错误， MST 就被清 0，变为从属接收模式。 通过 MST 和 TRS 的组合，变为以下的运行模式。另外，在选择时钟同步串行格式的情况下，当 MST=1 时，变为时钟输出。 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	传送时钟选择 在主模式时，请对照需要的传送率 （参照表 17.3）进行设定。

表 17.3　传送率

bit3	bit2	bit1	bit0	时钟	传送率 （kHz）				
CKS[3]	CKS[2]	CKS[1]	CKS[0]		Pϕ =16.7MHz	Pϕ =20.0MHz	Pϕ =25.0MHz	Pϕ =26.7MHz	Pϕ =33.3MHz
0	0	0	0	Pϕ/44	379	455	568	606	758
			1	Pϕ/52	321	385	481	513	641
		1	0	Pϕ/64	260	313	391	417	521
			1	Pϕ/72	231	278	347	370	463
	1	0	0	Pϕ/84	198	238	298	317	397
			1	Pϕ/92	181	217	272	290	362
		1	0	Pϕ/100	167	200	250	267	333
			1	Pϕ/108	154	185	231	247	309
1	0	0	0	Pϕ/176	94.7	114	142	152	189
			1	Pϕ/208	80.1	96.2	120	128	160
		1	0	Pϕ/256	65.1	78.1	97.7	104	130
			1	Pϕ/288	57.9	69.4	86.8	92.6	116
	1	0	0	Pϕ/336	49.6	59.5	74.4	79.4	99.2
			1	Pϕ/368	45.3	54.3	67.9	72.5	90.6
		1	0	Pϕ/400	41.7	50.0	62.5	66.7	83.3
			1	Pϕ/432	38.6	46.3	57.9	61.7	77.2

【注】 设定时，请满足外部规格。

17.3.2　I²C 总线控制寄存器 2 （ICCR2）

ICCR2 为 8 位可读 / 写寄存器，控制发送开始 / 停止条件、SDA 引脚的操作、SCL 引脚的监控以及 I²C 总线的控制部的复位。

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 I²C 总线占有 / 释放状态的标志功能、发送主模式开始 / 停止条件的功能 2 种。在时钟同步串行格式时，读取此位时，总读出 0。在 I²C 总线格式，SCL= 高电平的状态下，如果 SDA 从高电平变为低电平，就认为开始条件被发送，将此位置 1。SCL= 高电平的状态下，如果 SDA 从低电平变为高电平，就认为停止条件被发送，将此位清 0。在发送开始条件时，给 BBSY 写 1 且给 SCP 写 0。再次发送开始条件时也同样进行此操作。发送停止条件时，给 BBSY 写 0 且给 SCP 写 0。
6	SCP	1	R/W	禁止发送开始 / 停止条件 SCP 位控制在主模式下发送开始条件 / 停止条件。在发送开始条件时，给 BBSY 写 1 且给 SCP 写 0。再次发送开始条件时也同样。在发送停止条件时，给 BBSY 写 0 且给 SCP 写 0。读取此位时，总读出 1。即使写 1，数据也不被保存。
5	SDAO	1	R/W	控制 SDA 输出值 在改变 SDA 的输出电平时， SDAO 和 bit4 的 SDAOP 配合使用。另外，请勿在传送中操作此位。 0：读取时， SDA 引脚输出为低电平 写入时，将 SDA 引脚输出变为低电平 1：读取时， SDA 引脚输出为高电平 写入时，将 SDA 引脚输出变为 Hi-Z（通过外部上拉电阻，输出高电平）
4	SDAOP	1	R/W	SDAO 写保护 通过改写 SDAO 位，控制 SDA 引脚输出电平的变更。改变输出电平时，给 SDAO 写 0 且给 SDAOP 写 0，或者给 SDAO 写 1 且给 SDAOP 写 0。读取此位时，总读出 1。
3	SCLO	1	R	SCL 输出电平 SCLO 监控 SCL 的输出电平。读取时，如果 SCLO 是 1， SCL 引脚输出就为高电平，如果 SCLO 是 0， SCL 引脚输出就为低电平。
2	—	1	R	保留位 读写的值总是 1。
1	IICRST	0	R/W	IIC 控制部复位 IICRST 对除了 I²C 总线的寄存器以外的控制部进行复位。在 I²C 总线的运行中，由于通信不良而中止时，如果 IICRST 位置 1，就能复位 IIC3 的一部分寄存器和控制部。
0	—	1	R	保留位 读写的值总是 1。

17.3.3　I²C 总线模式寄存器 （ICMR）

ICMR 为 8 位可读 / 写寄存器，选择 MSB 优先 /LSB 优先和传送位数。
通过 ICCR2 的 IICRST，初始化 BC[2:0] 为 H'0。

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	选择 MSB 优先 /LSB 优先 0：MSB 优先 1：LSB 优先 在 I²C 总线格式使用时，请设定为 0。
6	—	0	R	保留位 读写的值总是 0。
5、4	—	均为 1	R	保留位 读写的值总是 1。
3	BCWP	1	R/W	BC 写保护 控制 BC[2:0] 的写入。在改写 BC[2:0] 时，请将此位设定为 0。另外，在时钟同步串行格式，请勿改写 BC[2:0]。 0：写入时，设定 BC[2:0] 的值 1：读取时，总读出 1 写入时， BC[2:0] 的设定值无效
2 ～ 0	BC[2:0]	000	R/W	位计数器 指定下一次传送数据的位数。如果读取此位，就能知道剩余的传送位数。在 I²C 总线格式，给数据加上 1 个应答位后进行传送。请在传送帧之间进行设定。另外，如果设定 B'000 以外的值，请在 SCL 为低电平状态时进行。在包含应答的数据传送结束后，这些位自动返回到 B'000。同时，在检测出停止条件后自动变为 B'111。通过将上电复位、软件待机模式、模块待机模式以及将 ICCR2 的 IICRST 设定为 1，清除本位。在选择时钟同步串行格式时，请勿改写。 I²C 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

17.3.4　I²C 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	发送中断允许 当 ICSR 的 TDRE 被置位时，允许 / 禁止发送数据空中断 （TXI）。 0：禁止发送数据空中断请求 （TXI） 1：允许发送数据空中断请求 （TXI）
6	TEIE	0	R/W	发送结束中断允许 在 ICSR 的 TDRE 为 1 的状态下，并且在第 9 个时钟上升时，TEIE 选择允许 / 禁止发送结束中断 （TEI）。另外，能通过将 TEND 清 0 或者将 TEIE 清 0，解除 TEI。 0：禁止发送结束中断请求 （TEI） 1：允许发送结束中断请求 （TEI）
5	RIE	0	R/W	接收中断允许 当接收数据从 ICDRS 被传送到 ICDRR，并且在 ICSR 的 RDRF 被置 1 时，RIE 选择允许 / 禁止接收数据满中断请求 （RXI）和选择允许 / 禁止时钟同步格式时的溢出错误中断请求 （ERI）。另外，能通过将 RDRF 清 0 或者将 RIE 清 0，解除 RXI。 0：禁止接收数据满中断请求 （RXI） 1：允许接收数据满中断请求 （RXI）
4	NAKIE	0	R/W	NACK 接收中断允许 当 ICSR 的 NACKF 或者 AL/OVE 被置位时， NAKIE 选择允许 / 禁止 NACK 检测中断请求 （NAKI）和选择允许 / 禁止时钟同步格式时的溢出错误（ICSR 的 OVE 置位）中断请求（ERI）。另外，能通过将 NACKF 或者 AL/OVE 清 0，或者将 NAKIE 清 0，解除 NAKI。 0：禁止 NACK 接收中断请求 （NAKI） 1：允许 NACK 接收中断请求 （NAKI）

位	位名	初始值	R/W	说明
3	STIE	0	R/W	停止条件检测中断允许 ICSR 的 STOP 被置位时，STIE 选择允许 / 禁止停止条件检测中断请求（STPI）。 0：禁止停止条件检测中断请求 （STPI） 1：允许停止条件检测中断请求 （STPI）
2	ACKE	0	R/W	应答位判定选择 0：忽略接收应答的内容进行连续传送。 1：当接收应答为 1 时，中止传送。
1	ACKBR	0	R	接收应答 发送模式时，此位保存从接收器件接收的应答位的内容，写入无效。在 ICCR2 的 BBSY 被置 1 时，清除本位。 0：接收应答 =0 1：接收应答 =1
0	ACKBT	0	R/W	发送应答 接收模式时，设定在应答时序发送的位。 0：在应答时序发送 0 1：在应答时序发送 1

17.3.5　I2C 总线状态寄存器 （ICSR）

ICSR 是 8 位可读 / 写寄存器，确认各中断请求标志和状态。

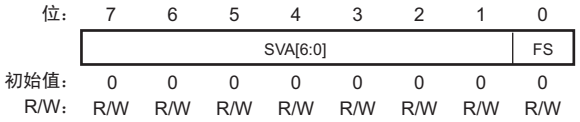
位：	7	6	5	4	3	2	1	0
	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
初始值：	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7	TDRE	0	R/W	发送数据空 ［清除条件］ • 在读出 1 的状态后，写 0 时 • 在给 ICDRT 写数据时 ［置位条件］ • 将数据从 ICDRT 传送到 ICDRS，并且 ICDRT 变为空时 • 将 TRS 置位时 • 发送开始条件 （包含再送）时 • 在从属模式，从接收模式变为发送模式时
6	TEND	0	R/W	发送结束 ［清除条件］ • 在读出 1 的状态后，写 0 时 • 在给 ICDRT 写数据时 ［置位条件］ • I2C 总线格式时，在 TDRE 为 1 的状态下 SCL 的第 9 个时钟上升时 • 在时钟同步串行格式，发出发送帧的最后位时
5	RDRF	0	R/W	接收数据满 ［清除条件］ • 在读出 1 的状态后，写 0 时 • 在读取 ICDRR 时 ［置位条件］ • 将接收数据从 ICDRS 传送到 ICDRR 时

位	位名	初始值	R/W	说明
4	NACKF	0	R/W	无应答检测标志 ［清除条件］ • 在读出 1 的状态后，写 0 时 ［置位条件］ • 在 ICIER 的 ACKE=1 的状态下，发送时没有来自接收器件的应答时
3	STOP	0	R/W	停止条件检测标志 ［清除条件］ • 在读出 1 的状态后，写 0 时 ［置位条件］ • 结束帧传送后检测出停止条件时
2	AL/OVE	0	R/W	仲裁失败标志 / 溢出错误标志 在 I²C 总线格式的主模式， AL/OVE 表示总线竞争失败。同时，在时钟同步串行格式，并且在 RDRF=1 的状态下， AL/OVE 表示接收到最后位。 在多个主器件几乎同时要占有总线时， I²C 总线接口 3 监控 SDA，如果与自己发送的数据不同，就将 AL 标志置 1，表示总线已被其它主器件占有。 ［清除条件］ • 在读出 1 的状态后，写 0 时 ［置位条件］ • 在主发送模式，并且在 SCL 的上升沿，内部 SDA 和 SDA 引脚的电平不匹配时 • 在主模式，当检测出开始条件，并且 SDA 引脚为高电平时 • 在时钟同步串行格式，并且在 RDRF=1 的状态下接收到最后位时
1	AAS	0	R/W	从属地址识别标志 在从属接收模式，当开始条件后的第一帧与 SAR 的 SVA6 ～ SVA0 匹配时，此标志被置位。 ［清除条件］ • 在读出 1 的状态后，写 0 时 ［置位条件］ • 在从属接收模式，检测出从属地址时 • 在从属接收模式，检测出一般调用地址时
0	ADZ	0	R/W	一般调用地址识别标志 在 I²C 总线格式的从属接收模式时有效 ［清除条件］ • 在读出 1 的状态后，写 0 时 ［置位条件］ • 从属接收模式，检测出一般调用地址时

17.3.6　从属地址寄存器 （SAR）

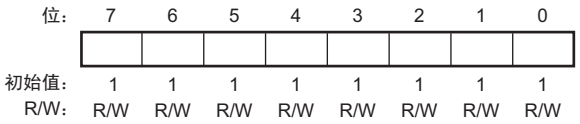
SAR 是 8 位可读 / 写寄存器，选择格式和设定从属地址。在 I2C 总线格式是从属模式时，当开始条件后送来的第 1 帧的高 7 位和 SAR 的高 7 位匹配时，作为从属器件运行。



位	位名	初始值	R/W	说明
7 ～ 1	SVA[6:0]	0000000	R/W	从属地址 设定与连接到 I2C 总线的其它从属器件不同的唯一地址。
0	FS	0	R/W	格式选择 0：选择 I2C 总线格式 1：选择时钟同步串行格式

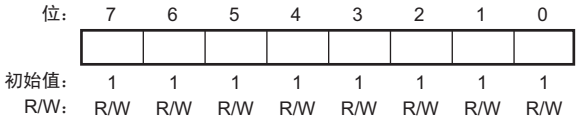
17.3.7　I2C 总线发送数据寄存器 （ICDRT）

ICDRT 是保存发送数据的 8 位可读 / 写寄存器，当检测出移位寄存器（ICDRS）为空时，将写在 ICDRT 的发送数据传送到 ICDRS，开始数据发送。在 ICDRS 的数据发送中，如果把下次发送的数据预先写到 ICDRT，就能连续发送。



17.3.8　I2C 总线接收数据寄存器 （ICDRR）

ICDRR 是保存接收数据的 8 位寄存器。如果 1 字节数据接收结束，就将接收到的数据从 ICDRS 传送到 ICDRR，成为能接收下一个数据的状态。由于 ICDRR 是接收专用寄存器，不能从 CPU 写入。



17.3.9　I²C 总线移位寄存器 （ICDRS）

ICDRS 是用于发送 / 接收数据的寄存器。发送时，将发送数据从 ICDRT 传送到 ICDRS，从 SDA 引脚发送数据。接收时，如果 1 字节数据接收结束，就将数据从 ICDRS 传送到 ICDRR。另外，不能直接从 CPU 读取本寄存器。

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

17.3.10　NF2CYC 寄存器 （NF2CYC）

NF2CYC 是 8 位可读 / 写寄存器，能选择 SCL 引脚、 SDA 引脚的噪声消除范围。详细请参照 “17.4.7 噪声消除电路”。

位：	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PRS	NF2CYC
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	—	均为 0	R	保留位 读写的值总是 0。
1	PRS	0	R/W	选择脉冲宽度比率 指定 SCL 的 High 宽度和 Low 宽度的比率。但是，在 ICCR1 的 CKS[3:0]=H'7 或者 H'F 时，请勿设定 PRS=1。 0：High：Low=0.5：0.5 1：High：Low ≒ 0.4：0.6
0	NF2CYC	0	R/W	选择噪声消除范围 0：能除去外围时钟 1 个周期以内的噪声 1：能除去外围时钟 2 个周期以内的噪声

17.4　运行说明

I²C 总线接口 3 通过设定 SAR 的 FS，能在 I²C 总线格式和时钟同步串行格式进行通信。

17.4.1　I²C 总线格式

I²C 总线格式如图 17.3 所示， I²C 总线时序如图 17.4 所示。开始条件后的第 1 帧由 8 位构成。

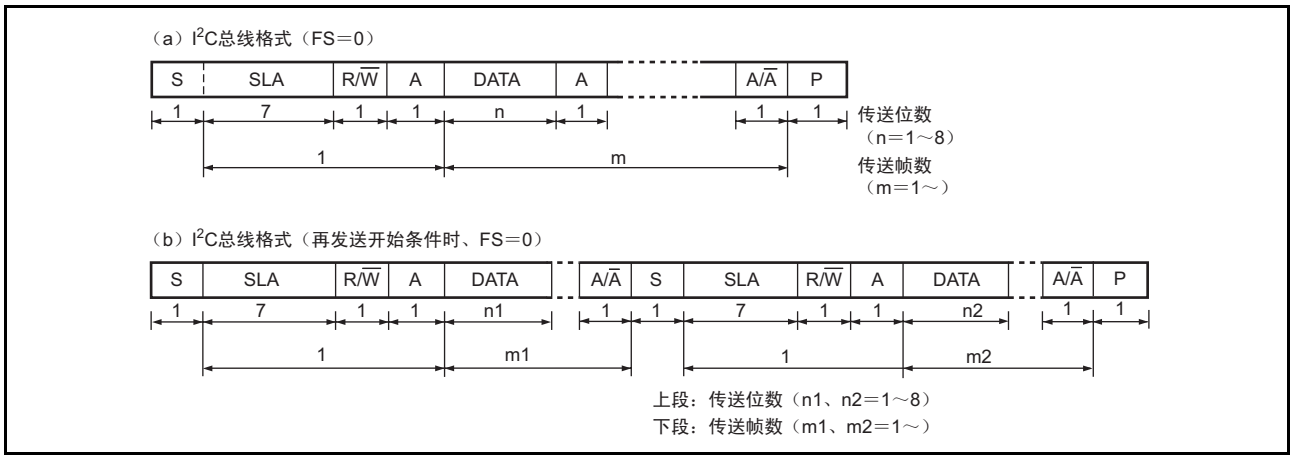


图 17.3　I²C 总线格式

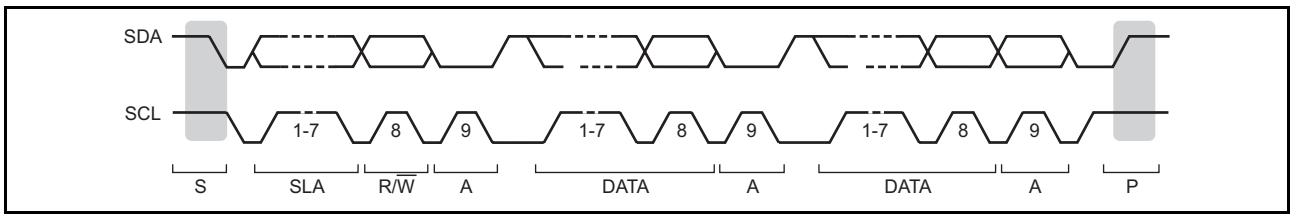


图 17.4　I²C 总线时序

【符号说明】

S　：开始条件。在 SCL= 高电平状态下，主器件将 SDA 从高电平变为低电平。

SLA ：从属地址

R/W ：表示发送 / 接收方向。为 1 时，将数据从从属器件发送到主器件；为 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 置为低电平。

DATA：发送 / 接收数据

P　：停止条件。在 SCL= 高电平状态下，主器件将 SDA 从低电平变为高电平。

17.4.2　主发送运行

在主发送模式，主器件输出发送时钟和发送数据，从属器件返回应答。关于主发送模式的运行时序，请参照图 17.5 和图 17.6。主发送模式的发送步骤和运行如下所示：

1. 将ICCR1的ICE位置1，并且设定ICCR1的CKS[3:0]等 （初始设定）。
2. 读取ICCR2的BBSY标志，并在确认总线处于释放状态后，将ICCR1的MST、TRS设定为主发送模式。然后，写BBSY=1和SCP=0 （发送开始条件），由此，生成开始条件。
3. 在确定ICSR的TDRE被置位后，给ICDRT写发送数据 （第1字节是表示从属地址和R/$\overline{W}$的数据）。此时， TDRE自动清0，在将数据从ICDRT传送到ICDRS后，再次将TDRE置位。
4. 在TDRE被置位的状态下，结束1字节的发送，在发送时钟的第9个时钟的上升沿，将ICSR的TEND置位。读取ICIER的ACKBR，在确认选择了从属器件后，将第2字节的数据写到ICDRT。由于ACKBR为1时无法确认从属器件，因此发送停止条件。通过写BBSY=0和SCP=0，发送停止条件。另外，在完成数据准备前或者在发送停止条件前， SCL被固定为低电平。
5. 每当TDRE被置位时，就将第2字节以后的发送数据写到ICDRT。
6. 如果将发送字节数写到ICDRT，就在TDRE被置位的状态下，等待TEND置位 （最后字节发送结束），或者在ICIER的ACKE已置位的状态下，等待来自接收器件的NACK （ICSR的NACKF=1）。然后，发送停止条件，清除TEND或者NACKF。
7. 如果ICSR的STOP被置位，就返回从属接收模式。

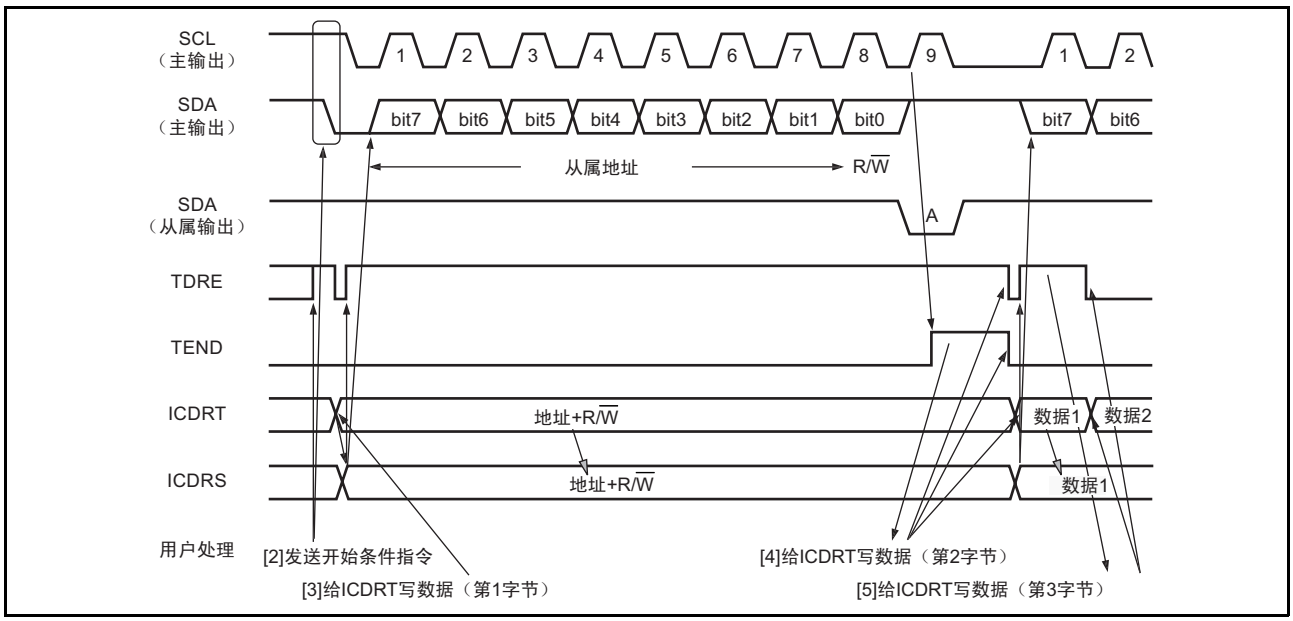


图 17.5　主发送模式的运行时序 （1）

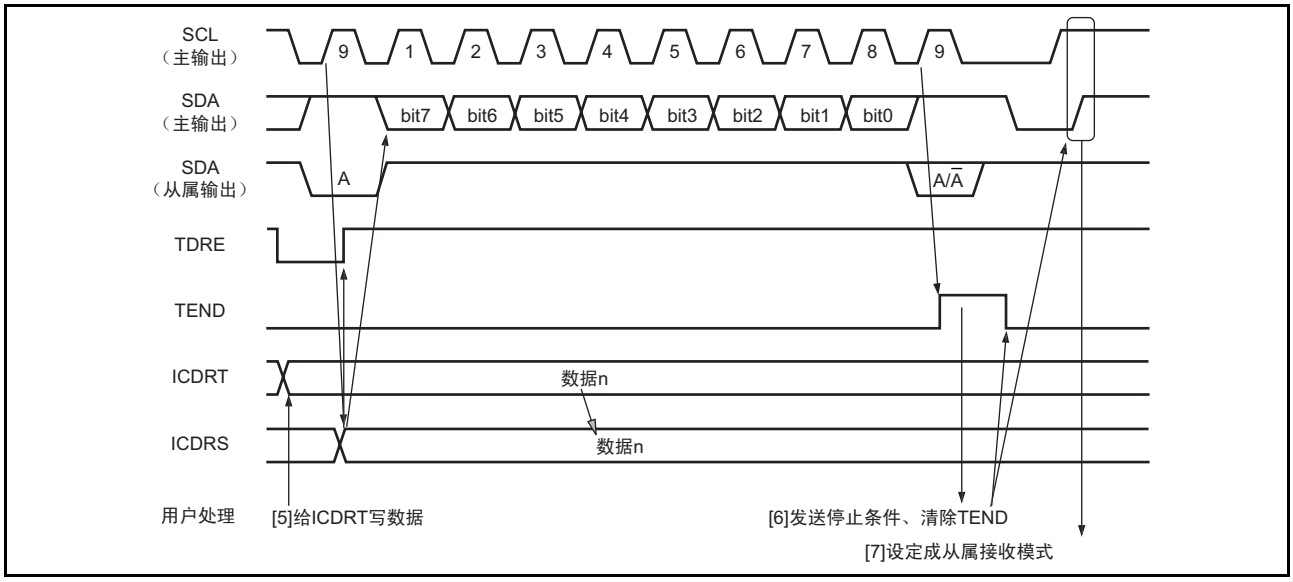


图 17.6　主发送模式的运行时序（2）

17.4.3　主接收运行

在主接收模式，主器件输出接收时钟，接收来自从属器件的数据，并且返回应答。关于主接收模式的运行时序，请参照图 17.7 和图 17.8。

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，从主发送模式转换到主接收模式。然后清除TDRE。
2. 如果虚读ICDRR，就开始*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，主器件将由ICIER的ACKBT设定的电平输出到SDA。
3. 1帧数据的接收结束，在接收时钟的第 9 个时钟的上升沿，将 ICSR 的 RDRF 置位。此时，能通过读取 ICDRR，读出接收到的数据，同时清除RDRF。
4. 每当RDRF被置位时，都能通过读取ICDRR连续接收数据。另外，由于其它处理，在RDRF置位的状态下，如果ICDRR的读取延迟到第8个时钟下降以后，在读取ICDRR之前， SCL被固定为低电平。
5. 当下一个接收数据为最后帧时，在读取ICDRR之前，将ICCR1的RCVD置位。由此，在下次接收后，成为可发送停止条件的状态。
6. 在接收时钟的第9个时钟的上升沿，如果RDRF被置位，就发送停止条件。
7. 如果ICSR的STOP被置位，就读取ICDRR，然后清除RCVD。
8. 返回从属接收模式。

【注】 *　只需要接收 1 字节时，请在将 ICCR1 的 RCVD 置位后进行 ICDRR 的虚读。

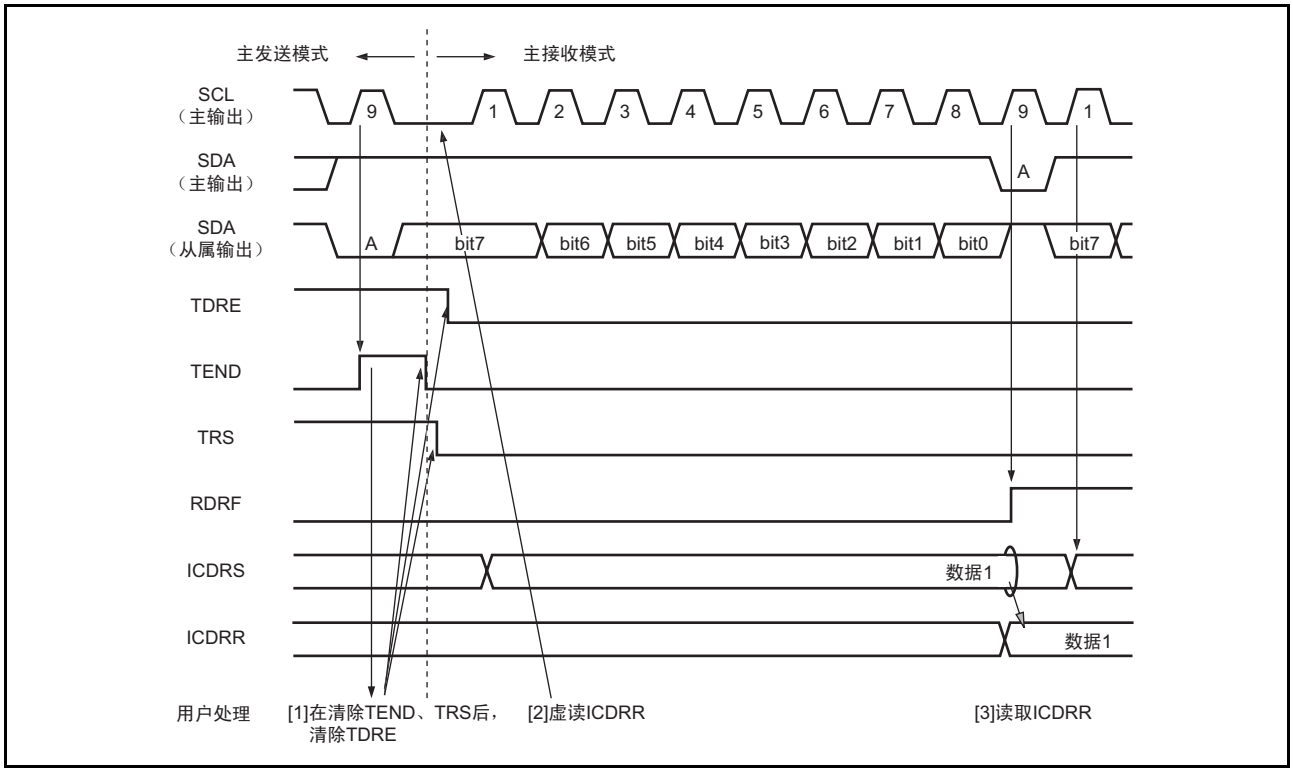


图 17.7　主接收模式的运行时序 （1）

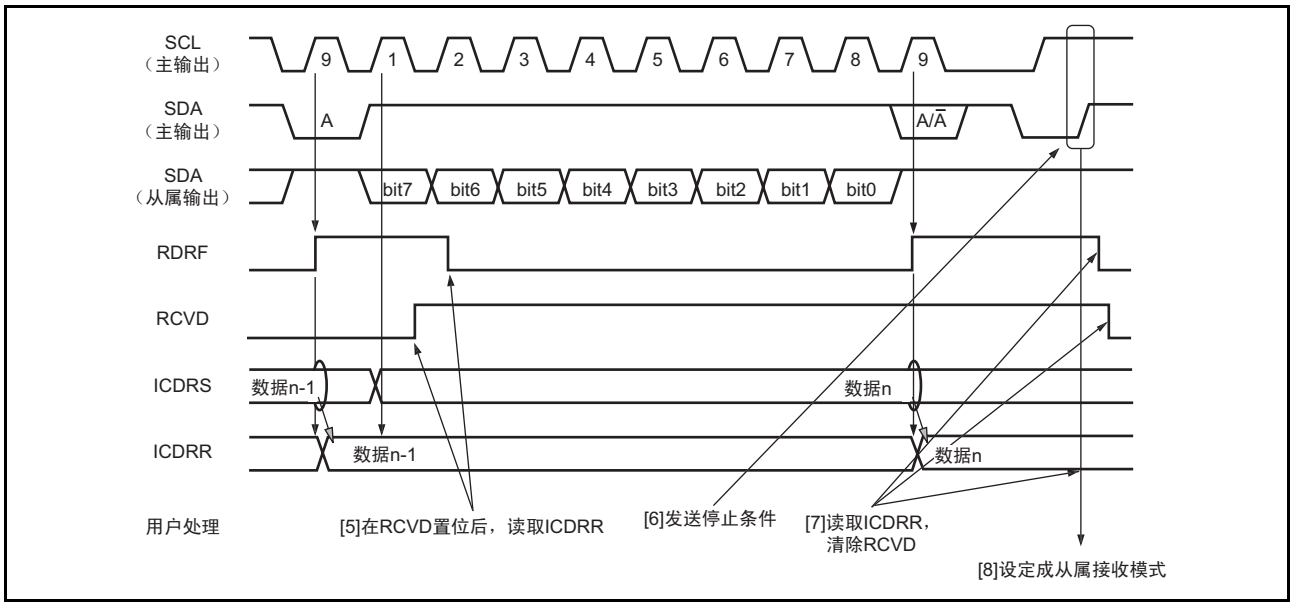


图 17.8　主接收模式的运行时序 （2）

17.4.4　从属发送运行

在从属发送模式，从属器件输出发送数据，主器件输出接收时钟，并且返回应答。关于从属发送模式的运行时序，请参照图 17.9 和图 17.10。

从属发送模式的发送步骤和运行如下所示：

1. 将ICCR1的ICE位置1，并且设定ICMR的CKS[3:0]等（初始设定）。将ICCR1的MST、TRS设定为从属接收模式，等待从属地址匹配。
2. 在检测出开始条件后的第 1 帧，从属地址匹配时，在第 9 个时钟的上升沿，从属器件将由 ICIER 的ACKBT设定的电平输出到SDA。此时，在第8位的数据（R/W）为1时，ICCR1的TRS和ICSR的TDRE被置位，自动转换到从属发送模式。每当TDRE被置位时，如果给ICDRT写发送数据，就能连续发送。
3. 将最后的发送数据写到ICDRT后，当TDRE被置位时，在TDRE=1的状态下，等待ICSR的TEND被置位。如果TEND被置位，就清除TEND。
4. 清除为了结束处理的TRS，并且虚读ICDRR。由此来释放SCL。
5. 清除TDRE。

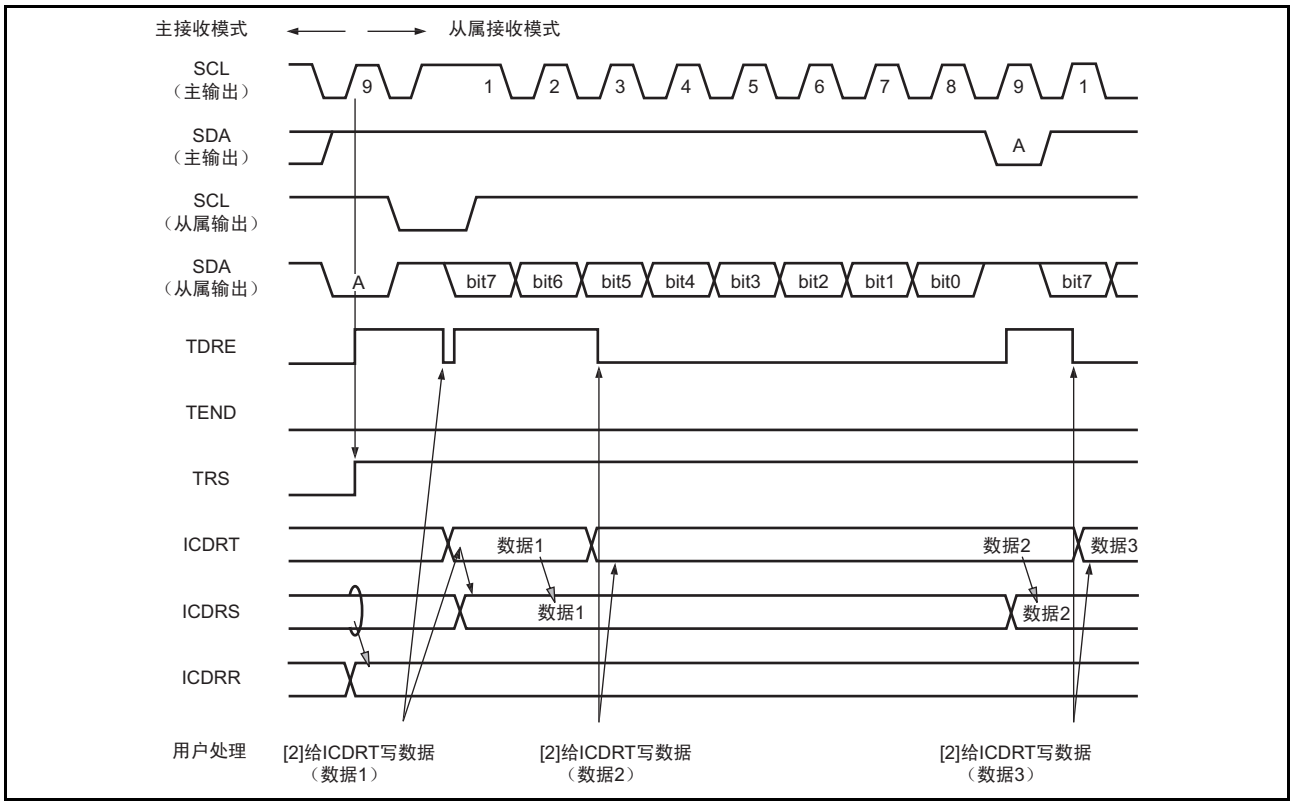


图 17.9　从属发送模式的运行时序（1）

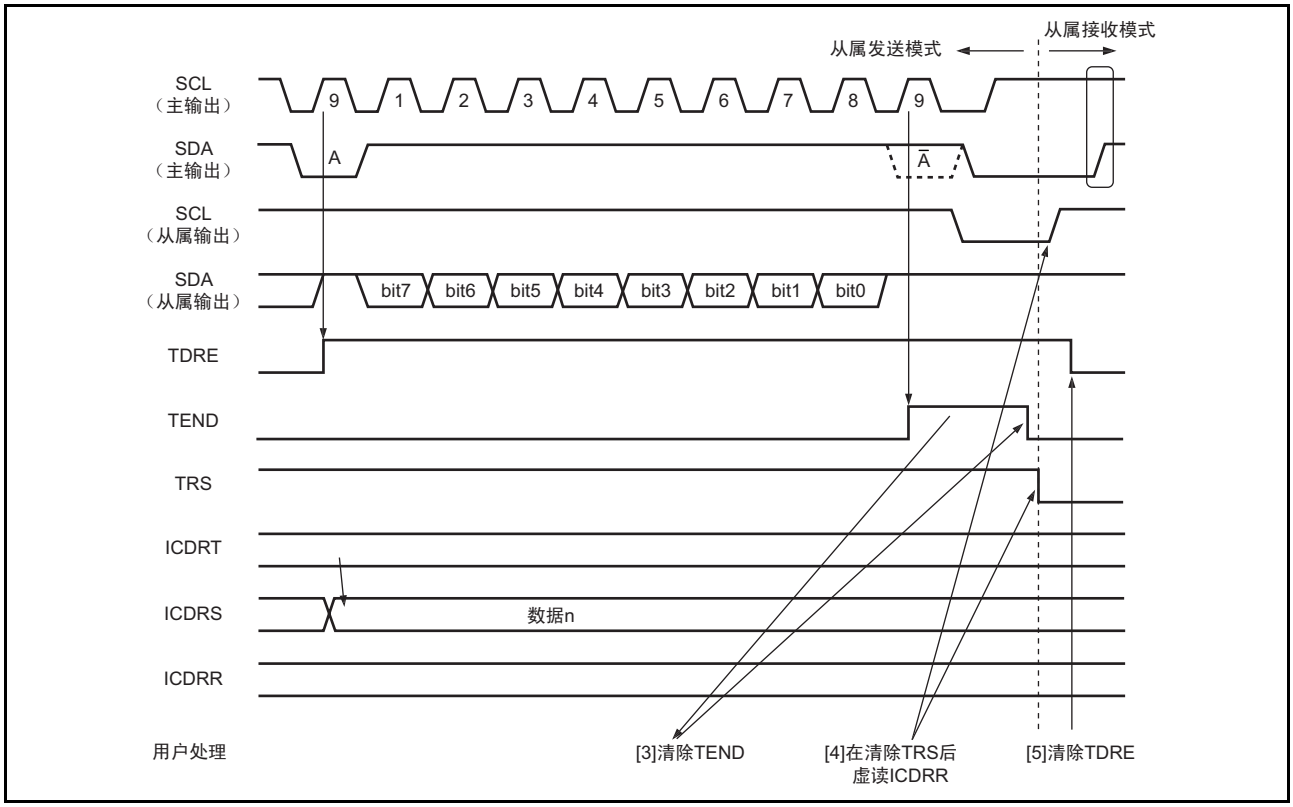


图 17.10　从属发送模式的运行时序（2）

17.4.5　从属接收运行

在从属接收模式，主器件输出发送时钟和发送数据，从属器件返回应答。关于从属接收模式的运行时序，请参照图 17.11 和图 17.12。从属接收模式的接收步骤和运行如下所示：

1. 将ICCR1的ICE位置1，并且设定ICMR的CKS[3:0]等（初始设定）。将ICCR1的MST和TRS设为从属接收模式，等待从属地址匹配。
2. 在检测出开始条件后的第一帧，从属地址匹配时，在第 9 个时钟的上升沿，从属器件将由 ICIER 的ACKBT设定的电平输出到SDA。由于ICSR的RDRF同时被置位，虚读ICDRR （因为读数据表示从属地址+R/W，所以不用）。
3. 每当RDRF被置位时，读取ICDRR。在RDRF被置位的状态下，如果第8个时钟下降，就在读取ICDRR之前，将SCL固定成低电平。应答设定 （在读取ICDRR前返回主器件）的变更反映在下一个传送帧。
4. 最后字节的读取同样通过读取ICDRR来进行。

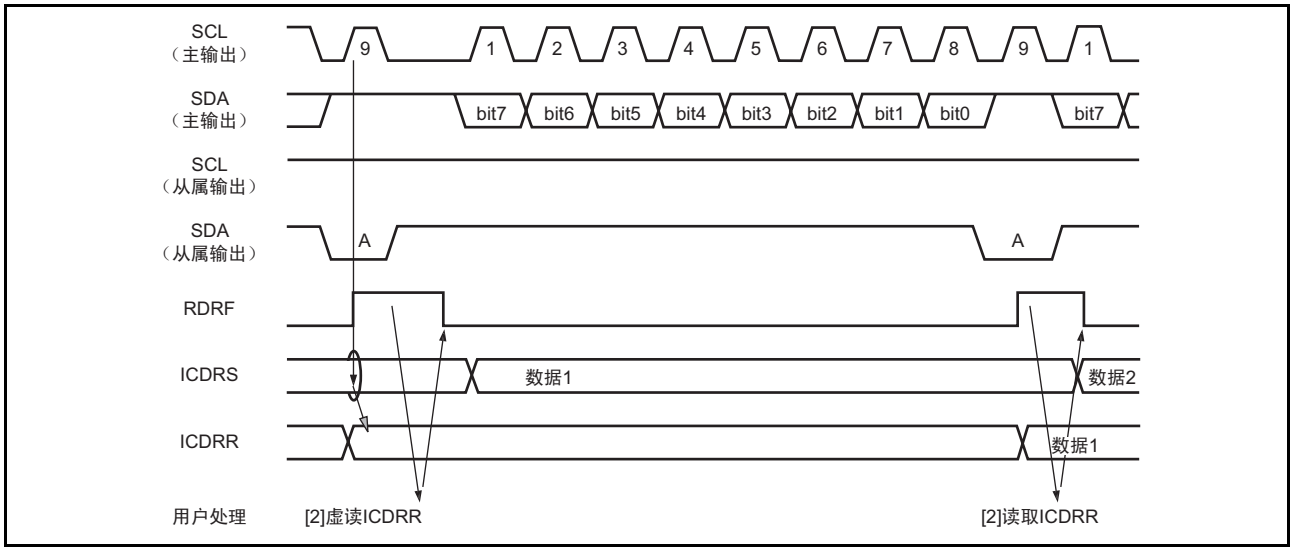


图 17.11　从属接收模式的运行时序 （1）

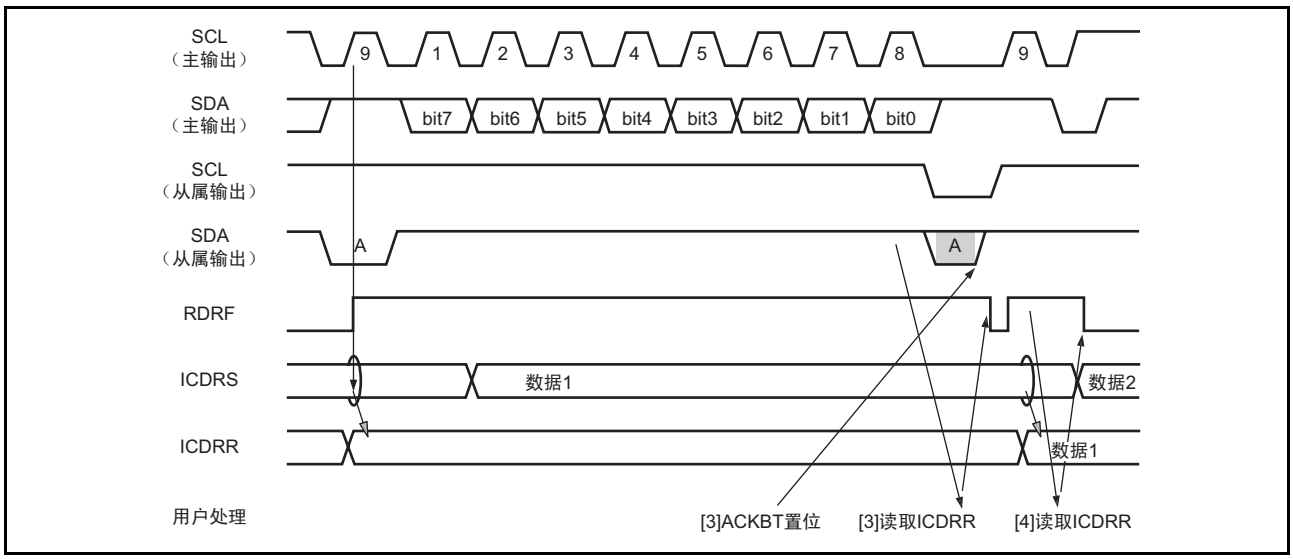


图 17.12　从属接收模式的运行时序 （2）

17.4.6　时钟同步串行格式

本模块能通过将 SAR 的 FS 置 1，作为时钟同步串行格式运行。在 ICCR1 的 MST=1 时，从 SCL 输出传送时钟；在 MST=0 时，输入传送时钟。

（1） 数据传送格式

时钟同步串行格式的传送格式如图 17.13 所示。

从 SCL 时钟的一个下降沿开始到下一个下降沿为止输出传送数据，保证 SCL 时钟上升沿的数据。数据的传送顺序能通过 ICMR 的 MLS，选择 MSB 优先或者 LSB 优先。另外，能通过 ICCR2 的 SDAO，在传送等待中改变 SDA 的输出电平。

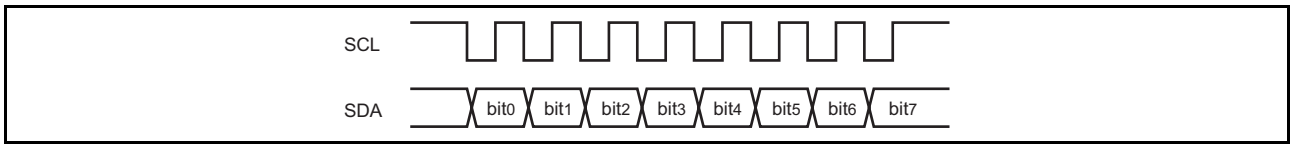


图 17.13　时钟同步串行格式的传送格式

（2） 发送运行

在发送模式，与传送时钟的下降同步从 SDA 输出发送数据。在 ICCR1 的 MST=1 时，传送时钟为输出；在 MST=0 时，传送时钟为输入。发送模式的运行时序，请参照图 17.14。发送模式的步骤和运行如下所示：

1. ICCR1 的 ICE 位置 1，并且设定 ICCR1 的 MST、 CKS[3:0]等 （初始设定）。
2. 设定 ICCR1 的 TRS，使其成为发送模式。由此， ICSR 的 TDRE 被置位。
3. 如果确认 TDRE 被置位，就给 ICDRT 写发送数据。由此，数据从 ICDRT 被传送到 ICDRS， TDRE 被自动置位。每当 TDRE 被置位时，如果给 ICDRT 写数据，就能连续发送。另外，从发送模式转换到接收模式时，请在 TDRE 置位的状态下清除 TRS。

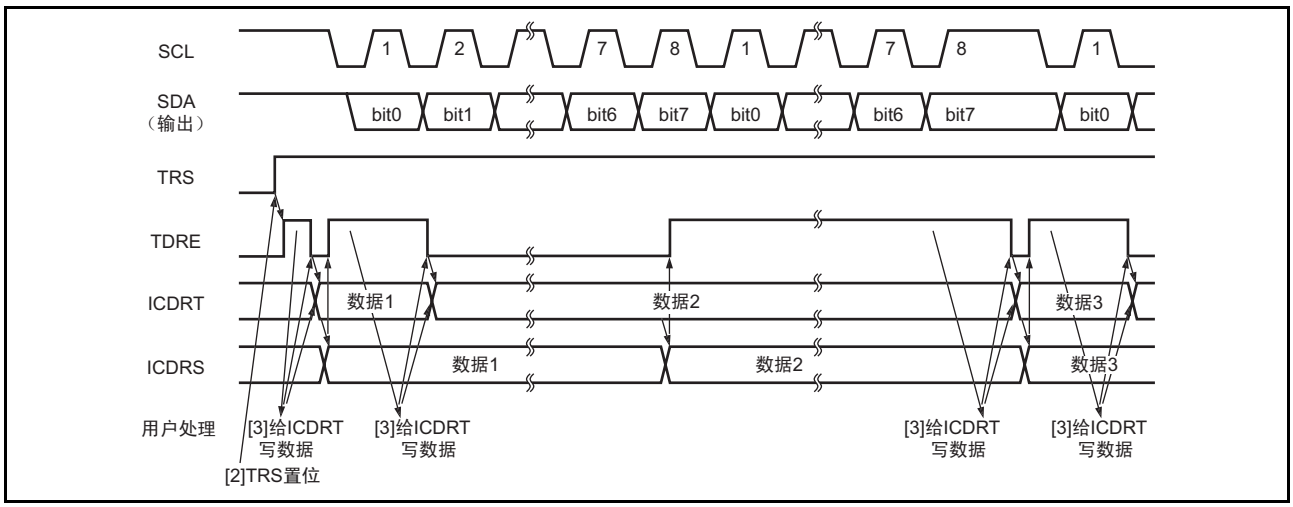


图 17.14　发送模式的运行时序

（3） 接收运行

在接收模式，在传送时钟的上升锁存数据。 ICCR1 的 MST=1 时，传送时钟为输出； MST=0 时，传送时钟为输入。关于接收模式的运行时序，请参照图 17.15。接收模式的步骤和运行如下所示：

1. 将ICCR1的ICE位置1，并且设定ICCR1的CKS[3:0]等 （初始设定）。
2. 在输出传送时钟时，设定MST=1。由此，开始输出接收时钟。
3. 当接收结束时，数据从ICDRS被传送到ICDRR， ICSR的RDRF被置位。在MST=1时，因为下一字节处于可接收状态，所以连续输出时钟。每当RDRF被置位时，通过读取ICDRR，可以连续接收。在RDRF被置位的状态下，如果第8个时钟上升，就检测出溢出，将ICSR的AL/OVE置位。此时，ICDRR的值保持以前的接收数据。
4. MST=1时，为了停止接收，在将ICCR1的RCVD置位后，读取ICDRR。由此，在下一字节接收结束后， SCL被固定成高电平。

【注】 MST=1 时，如果只需要接收 1 字节，请按照下列步骤进行。关于运行时序请参照图 17.16。

1. 将 ICCR1 的 ICE 位置 1，并设定 ICCR1 的 CKS[3:0] 等 （初始设定）。
2. 在 ICCR1 的 RCVD 位为 0 状态下，设定 MST=1。由此开始输出接收时钟。
3. 请在确认 ICMR 的 BC[2] 位被置 1 后，设定 ICCR1 的 RCVD=1。由此，在接收时钟输出 1 字节后， SCL 被固定为高电平。

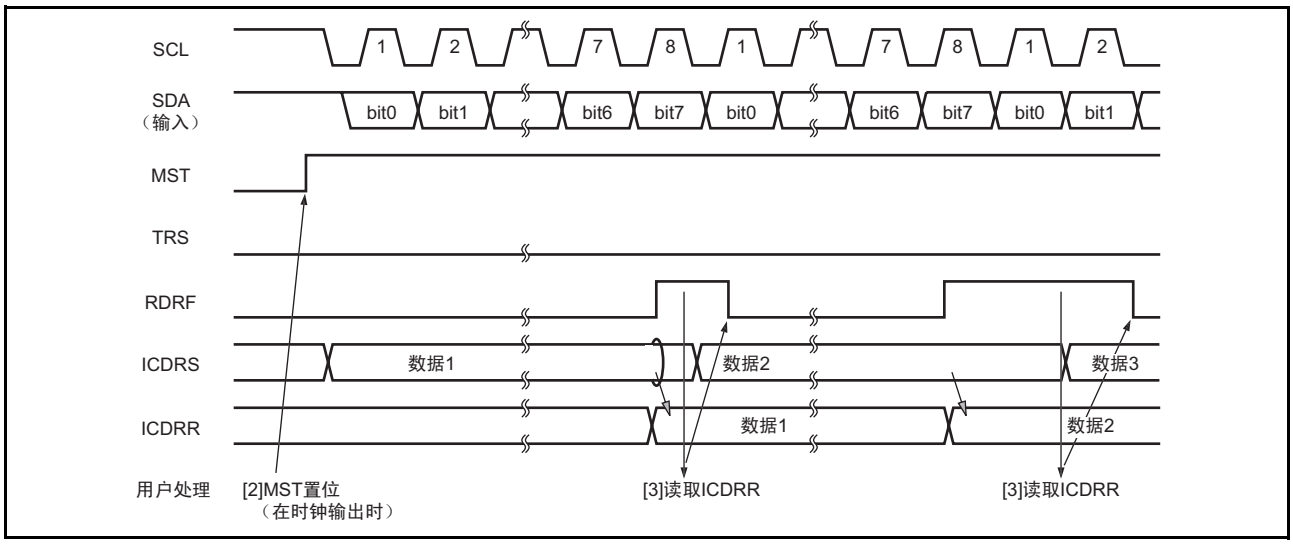


图 17.15　接收模式的运行时序

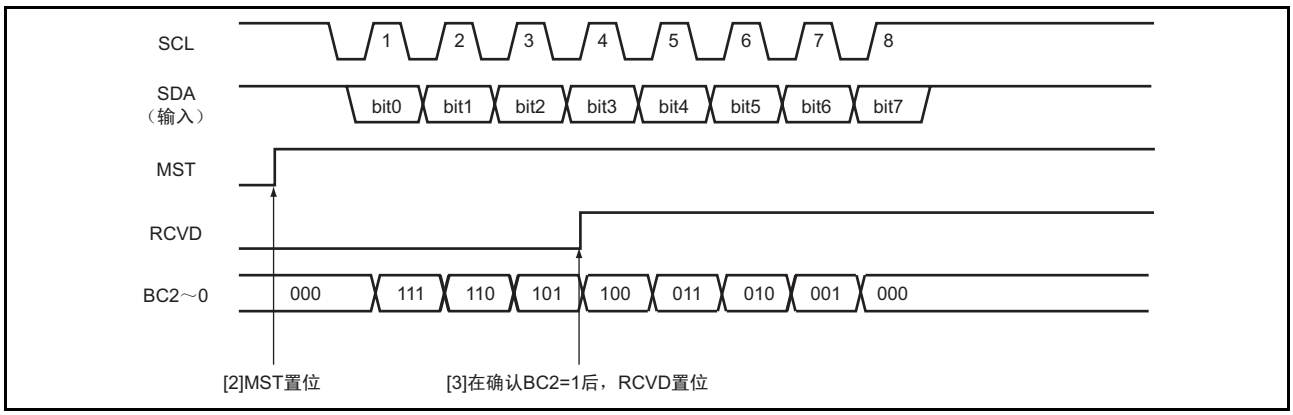


图 17.16　1 字节接收运行的时序 （MST=1）

17.4.7　噪声消除电路

SCL 引脚和 SDA 引脚的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 17.17 所示。

噪声消除电路由 3 段串联的锁存电路和匹配检测电路构成。用外部时钟采样 SCL 引脚的输入信号（或者 SDA 引脚的输入信号）。NF2CYC 寄存器是“0”，且 2 个锁存器输出匹配时，将电平传递给后面电路。NF2CYC 寄存器是 “1”，且 3 个锁存器输出匹配时，将电平传递给后面电路。不匹配时，保持以前的值。

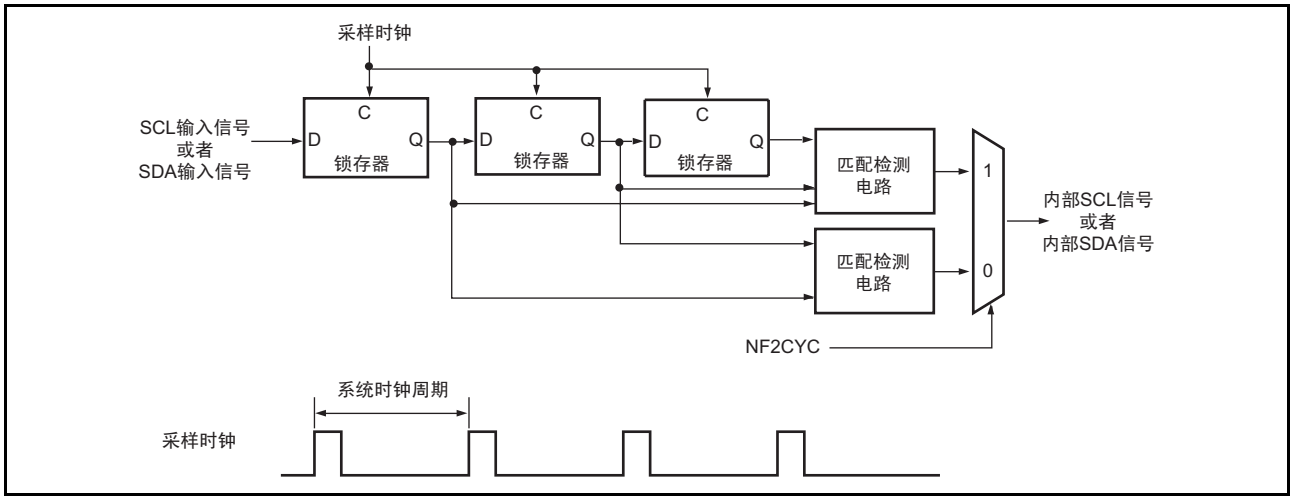


图 17.17　噪声消除电路的框图

17.4.8　使用例

使用 I²C 总线接口 3 时，各模式的流程图例如图 17.18 ～图 17.21 所示。

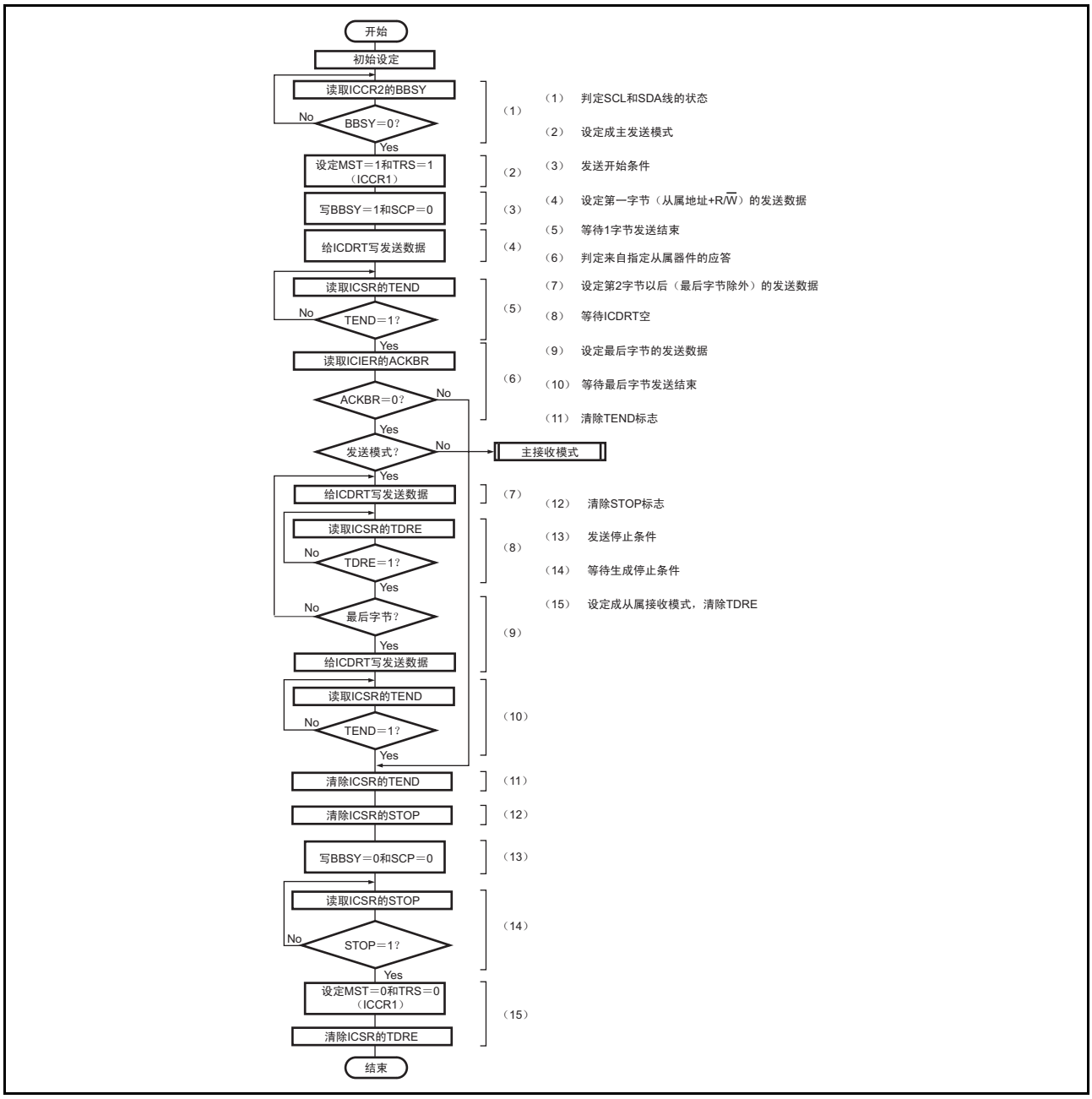


图 17.18　主发送模式的流程图例

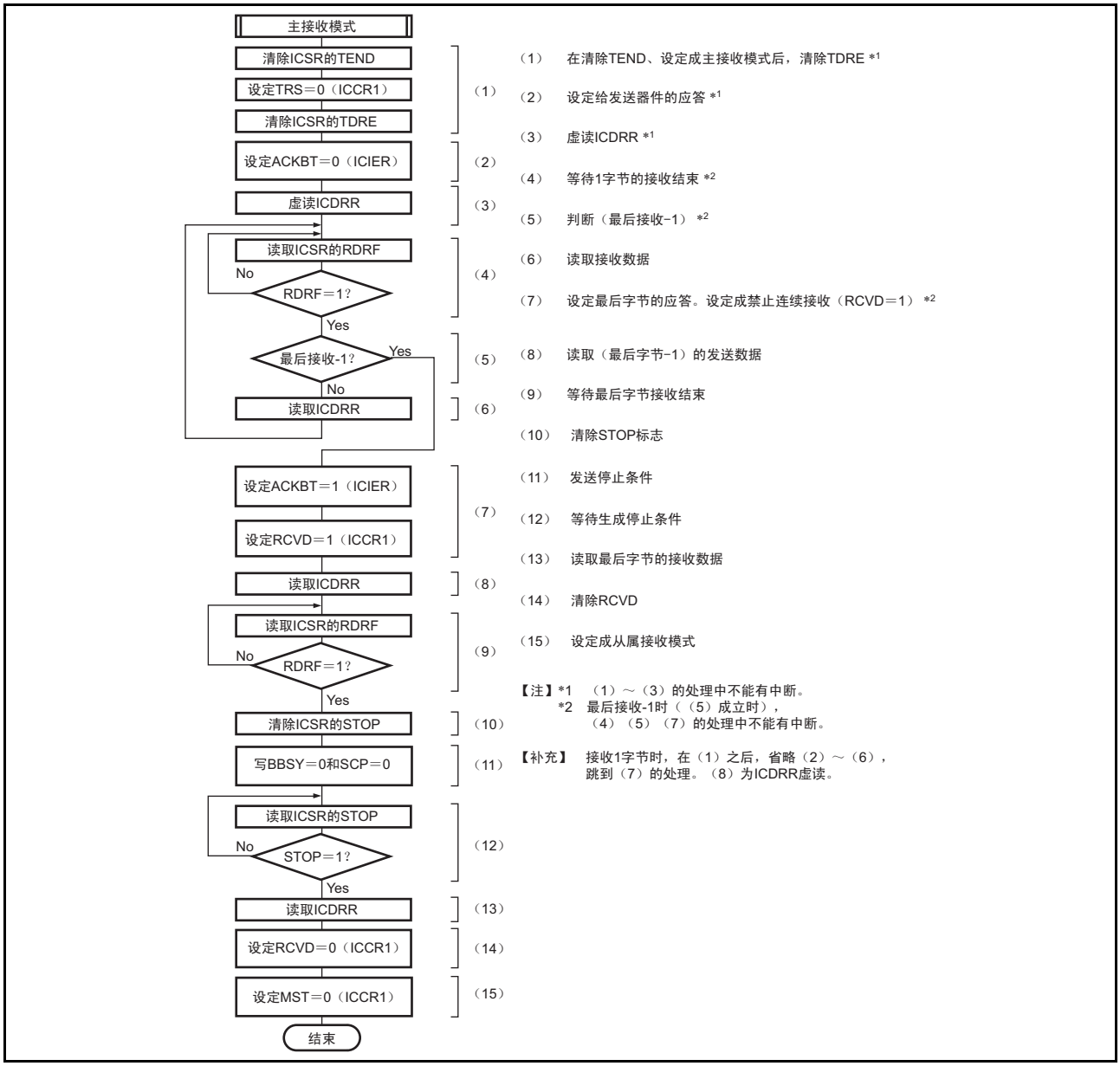


图 17.19　主接收模式的流程图例

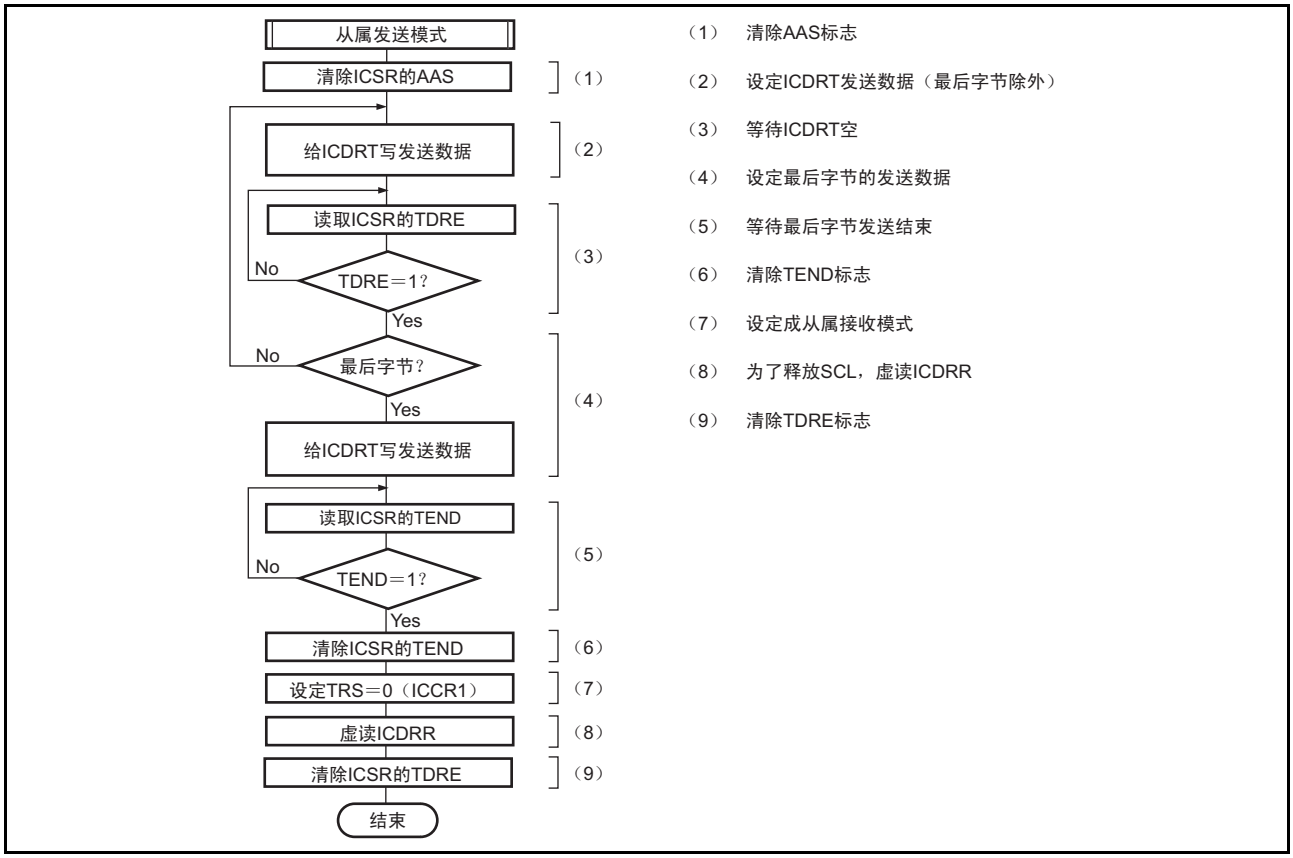


图 17.20　从属发送模式的流程图例

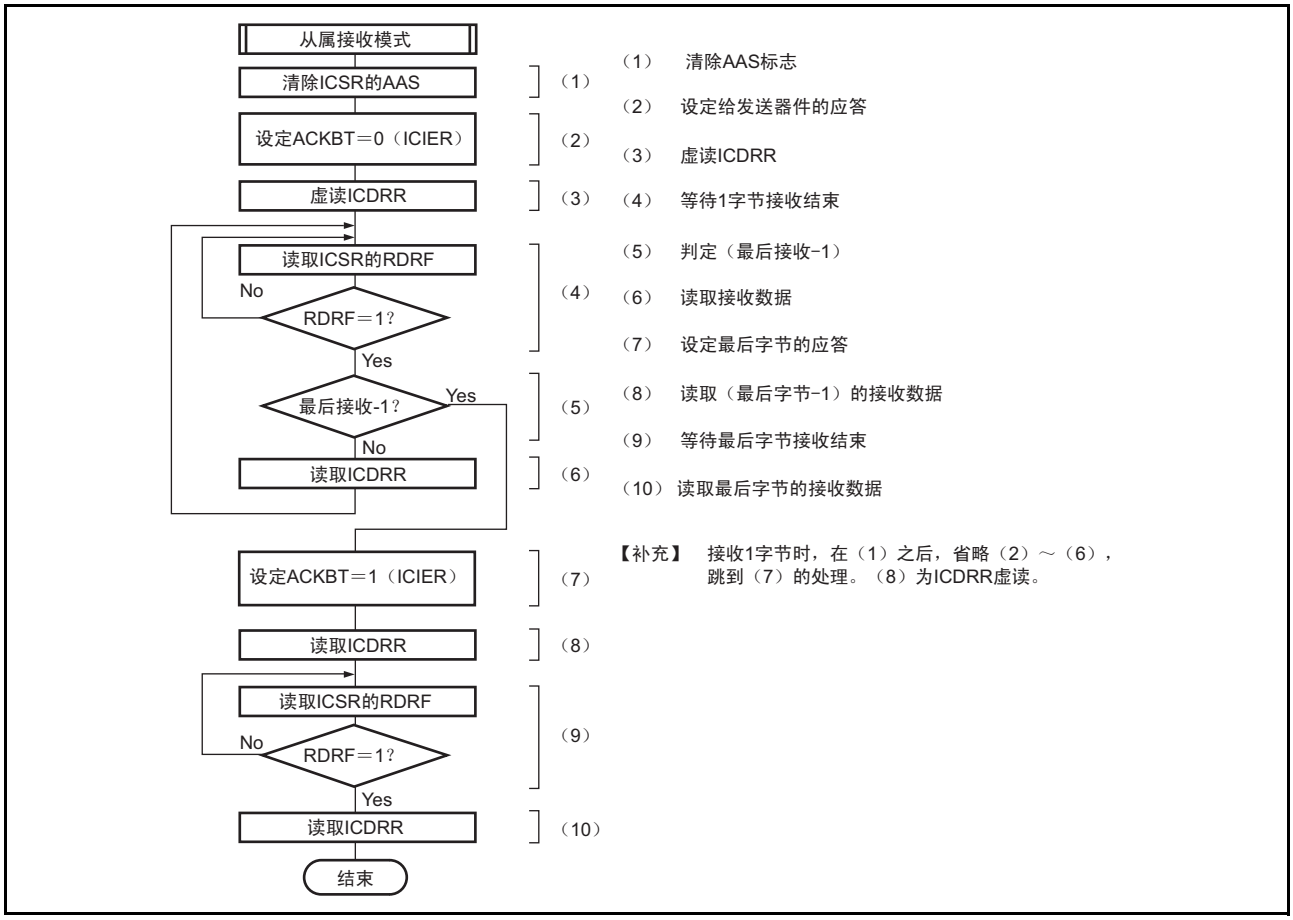


图 17.21　从属接收模式的流程图例

17.5　中断请求

本模块的中断请求有 6 种：发送数据空、发送结束、接收数据满、检测出 NACK、检测出停止条件、仲裁失败 / 溢出错误。各中断请求的内容如表 17.4 所示。

表 17.4　中断请求一览表

中断请求	简称	中断条件	I²C 总线格式	时钟同步串行格式
发送数据空	TXI	(TDRE=1) • (TIE=1)	○	○
发送结束	TEI	(TEND=1) • (TEIE=1)	○	○
接收数据满	RXI	(RDRF=1) • (RIE=1)	○	○
检测出停止条件	STPI	(STOP=1) • (STIE=1)	○	×
检测出 NACK	NAKI	{(NACKF=1)+(AL=1)} • (NAKIE=1)	○	×
仲裁失败 / 溢出错误			○	○

当表 17.4 的中断条件为 1 时，CPU 执行中断异常处理。另外，进行 DMAC 的启动设定时，TXI 和 RXI 能启动 DMAC。此时，不对 CPU 产生中断。在异常处理中，请务必清除各种中断源。但请注意：通过给 ICDRT 写发送数据，TDRE 和 TEND 将被自动清除；通过读取 ICDRR，RDRF 将被自动清除。特别是在给 ICDRT 写发送数据的同时， TDRE 再次被置位，并且在清除 TDRE 时，有时会多发送 1 个字节数据。

17.6　位同步电路

在设定主模式时，本模块有以下 2 种情况：

- SCL 被从属器件拉成低电平时
- 由于SCL线的负载（负载电容、上拉电阻），SCL的上升变得迟缓时

由于高电平期间可能变短，因此监控 SCL，按位取得同步的同时进行通信。

位同步电路的时序如图 17.22 所示。在将 SCL 从低电平输出改变到 Hi-Z 开始，到监控 SCL 为止的时间如表 17.5 所示。

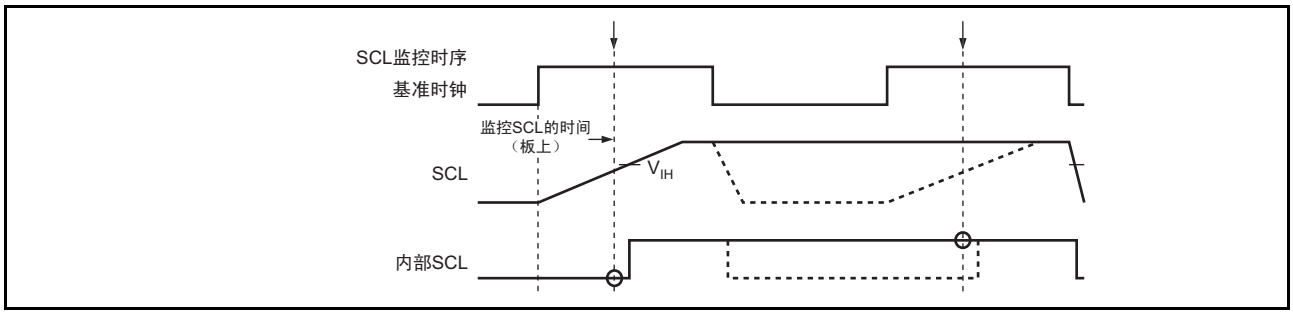


图 17.22　位同步电路的时序

表 17.5　监控 SCL 的时间

CKS[3]	CKS[2]	监控 SCL 的时间 *1
0	0	9tpcyc*2
	1	21tpcyc*2
1	0	19tpcyc*2
	1	43tpcyc*2

【注】　*1　从“SCL 监控时序基准时钟”上升沿计数并监控“监控 SCL 的时间”（pcyc）后的“（板上）SCL”电平。

*2　pcyc=Pϕ×cyc

17.7 使用时的注意事项

17.7.1 设定 ICCR1.CKS[3:0] 时的注意事项

请勿设定 ICCR1.CKS[3:0]=H'7 或者 H'F，以及 NF2CYC.PRS=1。

17.7.2 多主操作时的注意事项

在多主操作，本 LSI 的 IIC 传送速率的设定 （ICCR1 CKS[3:0]）比其他主操作慢时，偶尔会出现对 SCL 输出预料外宽度的 SCL。

有必要将传送速率设定为大于等于其他主操作最快传送速率的 1/1.8。

17.7.3 主接收模式时的注意事项

在第 8 个时钟的下降沿附近读取 ICDRR 时，有可能出现不能取接收数据的情况。

另外，接收缓冲器满并且在第 8 个时钟的下降沿附近设定 RCVD=1 时，有可能出现不能发送停止条件的情况。

请使用以下 1 或者 2 的方法解决。

1. 主接收模式时，请在8个时钟的下降沿之前读取ICDRR。
2. 主接收模式时，设置RCVD=1，按每1个字节的通信进行处理。

17.7.4 主接收模式、 ACKBT 设定时的注意事项

主接收模式运行时，请在连续转送的最后数据的第 8 个 SCL 的下降沿之前，设定 ACKBT。从属发送器件有溢出的可能。

17.7.5 仲裁失败时 MST 和 TRN 位状态的注意事项

在多主操作时，如果将 MST、 TRS 按顺序进行位操作并设定为主发送时，根据 TRS 的位操作指令执行中仲裁失败发生时序，有可能成为 ICSR 的 AL=1 且象主发送模式 （MST=1、 TRS=1）那样的矛盾状态。

为了回避该现象，有如下的方法：

- 多主操作时，请使用MOV指令设定MST、 TRS。
- 仲裁失败时，请确认MST=0、 TRS=0。

一旦不是 MST=0、 TRS=0 的状态，请重新设定 MST=0、 TRS=0。

第 18 章　串行音频接口 （SSI）

串行音频接口 （SSI）是发送 / 接收和飞利浦格式兼容的各种器件及音频数据的模块。除了通用格式，也支持多通道格式。

18.1　特点

- 通道数：4个通道
- 运行模式：非压缩模式
  非压缩模式支持分配至通道的串行音频流。
- 即可作为发送器也可作为接收器运行
- 可使用串行总线格式
- 在数据缓冲器和移位寄存器之间为异步传送
- 可选择串行总线接口使用的时钟分频比
- 通过DMAC或中断，可控制数据的发送/接收
- 可从以下的引脚选择过采样时钟
  EXTAL、 XTAL （时钟运行模式0、 1）
  CKIO （时钟运行模式2）
  AUDIO_CLK
  AUDIO_X1、 AUDIO_X2

SSI 模块的概况图如图 18.1 所示。

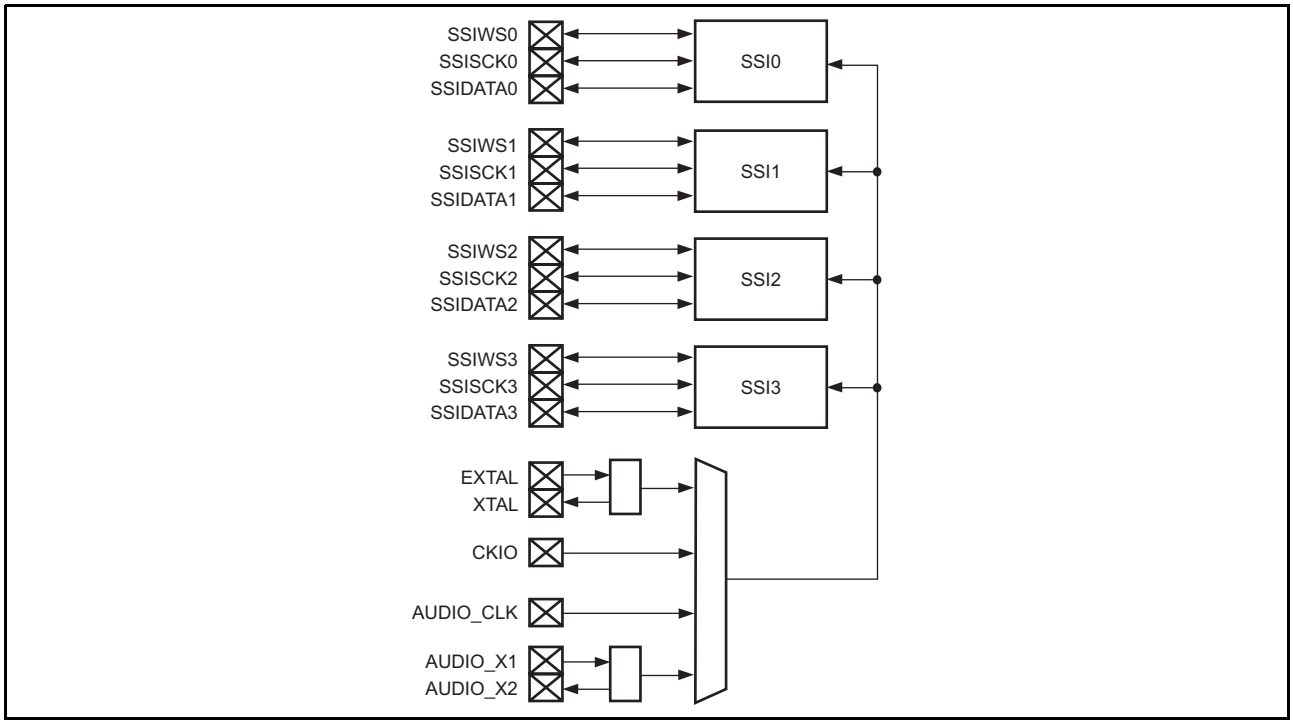


图 18.1　SSI 模块的概况图

SSI 的框图如图 18.2 所示。

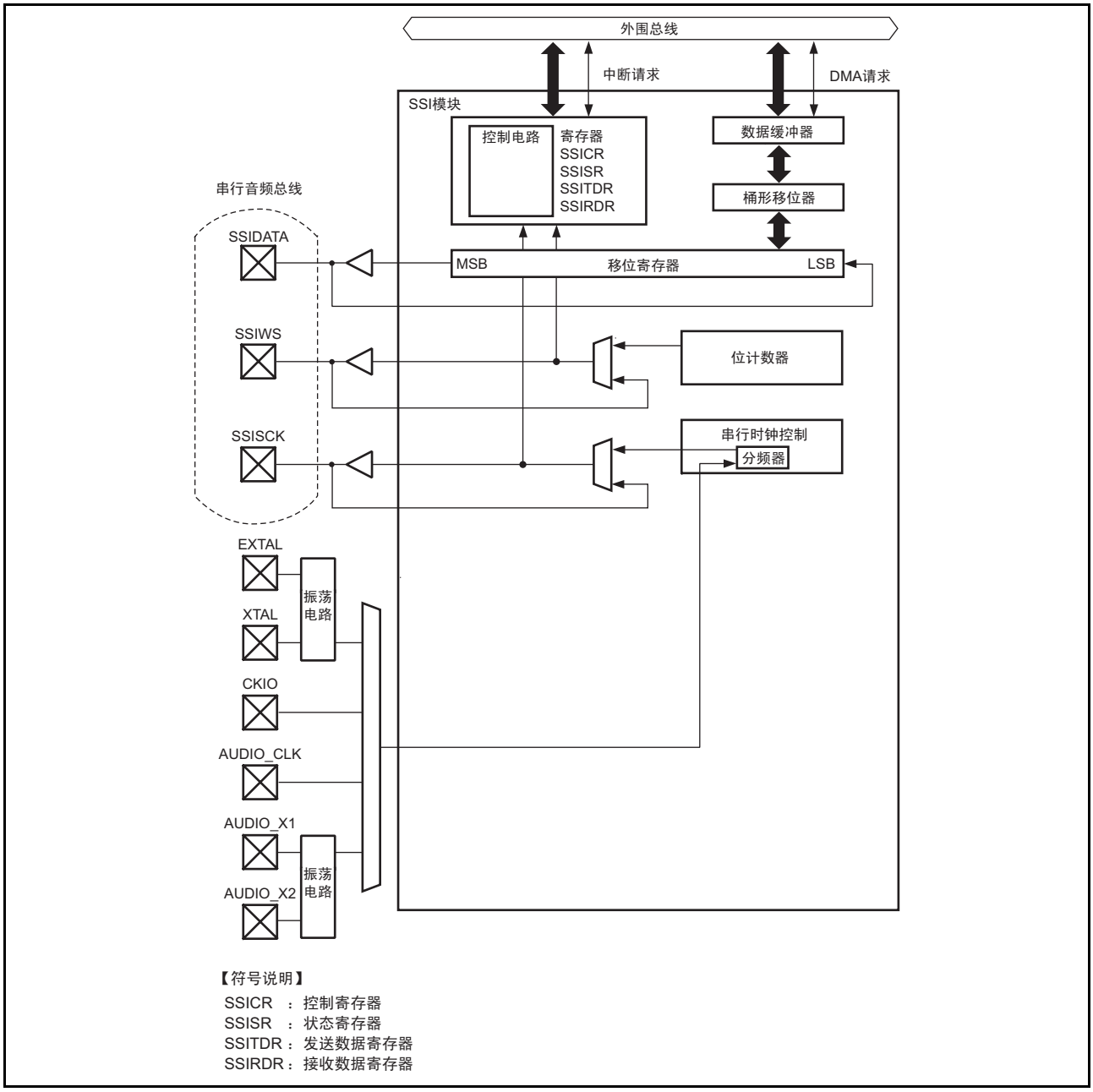


图 18.2　SSI 的框图

18.2　输入 / 输出引脚

与 SSI 模块有关的引脚构成如表 18.1 所示。

表 18.1　引脚构成

名称	数量	输入 / 输出	功　　能
SSISCK0	1	输入 / 输出	串行位时钟
SSIWS0	1	输入 / 输出	字选择
SSIDATA0	1	输入 / 输出	串行数据输入 / 输出
SSISCK1	1	输入 / 输出	串行位时钟
SSIWS1	1	输入 / 输出	字选择
SSIDATA1	1	输入 / 输出	串行数据输入 / 输出
SSISCK2	1	输入 / 输出	串行位时钟
SSIWS2	1	输入 / 输出	字选择
SSIDATA2	1	输入 / 输出	串行数据输入 / 输出
SSISCK3	1	输入 / 输出	串行位时钟
SSIWS3	1	输入 / 输出	字选择
SSIDATA3	1	输入 / 输出	串行数据输入 / 输出
AUDIO_CLK	1	输入	用于音频的外部时钟 （输入过采样时钟）
AUDIO_X1	1	输入	用于音频的晶体振荡器 （输入过采样时钟）
AUDIO_X2	1	输出	

18.3 寄存器的说明

SSI 有下列的寄存器。另外，在本文的中，简单说明了由通道引起的区别。

表 18.2 寄存器构成

通道	寄存器名	简称	R/W	初始值	地址	存取大小
0	控制寄存器 0	SSICR_0	R/W	H'00000000	H'FFFFC000	32
	状态寄存器 0	SSISR_0	R/W*	H'02000003	H'FFFFC004	32
	传送数据寄存器 0	SSITDR_0	R/W	H'00000000	H'FFFFC008	32
	接收数据寄存器 0	SSIRDR_0	R	H'00000000	H'FFFFC00C	32
1	控制寄存器 1	SSICR_1	R/W	H'00000000	H'FFFFC800	32
	状态寄存器 1	SSISR_1	R/W*	H'02000003	H'FFFFC804	32
	传送数据寄存器 1	SSITDR_1	R/W	H'00000000	H'FFFFC808	32
	接收数据寄存器 1	SSIRDR_1	R	H'00000000	H'FFFFC80C	32
2	控制寄存器 2	SSICR_2	R/W	H'00000000	H'FFFFD000	32
	状态寄存器 2	SSISR_2	R/W*	H'02000003	H'FFFFD004	32
	传送数据寄存器 2	SSITDR_2	R/W	H'00000000	H'FFFFD008	32
	接收数据寄存器 2	SSIRDR_2	R	H'00000000	H'FFFFD00C	32
3	控制寄存器 3	SSICR_3	R/W	H'00000000	H'FFFFD800	32
	状态寄存器 3	SSISR_3	R/W*	H'02000003	H'FFFFD804	32
	传送数据寄存器 3	SSITDR_3	R/W	H'00000000	H'FFFFD808	32
	接收数据寄存器 3	SSIRDR_3	R	H'00000000	H'FFFFD80C	32

【注】 * 该寄存器的 bit27 和 bit26 可读 / 写，其他位为只读。详细情况请参照 “18.3.2 状态寄存器 （SSISR）”。

18.3.1　控制寄存器 （SSICR）

SSICR 为 32 位可读 / 写寄存器，控制 IRQ、选择各极性状态、设定运行模式。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	DMEN	UIEN	OIEN	IIEN	DIEN	CHNL[1:0]		DWL[2:0]			SWL[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/W	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
	SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL	-	CKDV[2:0]			MUEN	-	TRMD	EN
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	R/W	R/W	R/W	R/W	R	R/W	R/W

位	位名	初始值	R/W	说　明
31 ～ 29	—	均为 0	R	保留位 读写的值总是 0。
28	DMEN	0	R/W	DMA 允许 允许 / 禁止 DMA 请求。 0：禁止 DMA 请求 1：允许 DMA 请求
27	UIEN	0	R/W	下溢中断允许 0：禁止下溢中断 1：允许下溢中断
26	OIEN	0	R/W	上溢中断允许 0：禁止上溢中断 1：允许上溢中断
25	IIEN	0	R/W	空闲模式中断允许 0：禁止空闲模式中断 1：允许空闲模式中断
24	DIEN	0	R/W	数据中断允许 0：禁止数据中断 1：允许数据中断
23、22	CHNL[1:0]	00	R/W	通道 表示各系统字的通道数。 00：各系统字由 1 个通道构成。 01：各系统字由 2 个通道构成。 10：各系统字由 3 个通道构成。 11：各系统字由 4 个通道构成。
21 ～ 19	DWL[2:0]	000	R/W	数据的字长 表示数据字的位数。 000：8 位 001：16 位 010：18 位 011：20 位 100：22 位 101：24 位 110：32 位 111：禁止设定

<table>
<tr><th>位</th><th>位名</th><th>初始值</th><th>R/W</th><th>说　　明</th></tr>
<tr><td>18 ～ 16</td><td>SWL[2:0]</td><td>000</td><td>R/W</td><td>系统的字长
表示系统字的位数。
000：8 位
001：16 位
010：24 位
011：32 位
100：48 位
101：64 位
110：128 位
111：256 位</td></tr>
<tr><td>15</td><td>SCKD</td><td>0</td><td>R/W</td><td>串行位时钟方向
0：串行位时钟输入，从属模式
1：串行位时钟输出，主模式
【注】 只能设定为 （SCKD、SWSD）=（0、0）和 （1、1）。禁止除此以外的设定。</td></tr>
<tr><td>14</td><td>SWSD</td><td>0</td><td>R/W</td><td>串行 WS 方向
0：串行字选择输入，从属模式
1：串行字选择输出，主模式
【注】 只能设定为 （SCKD、SWSD）=（0、0）和 （1、1）。禁止除此以外的设定。</td></tr>
<tr><td>13</td><td>SCKP</td><td>0</td><td>R/W</td><td>串行位时钟极性
0：SSIWS 和 SSIDATA 在 SSISCK 的下降沿变化
（在 SCK 的上升沿进行采样）
1：SSIWS 和 SSIDATA 在 SSISCK 的上升沿变化
（在 SCK 的下降沿进行采样）
<table>
<tr><th></th><th>SCKP＝0</th><th>SCKP＝1</th></tr>
<tr><td>接收时（TRMD＝0）SSIDATA
输入采样时序</td><td>SSISCK
上升沿</td><td>SSISCK
下降沿</td></tr>
<tr><td>发送时（TRMD＝1）SSIDATA
输出变化时许</td><td>SSISCK
下降沿</td><td>SSISCK
上升沿</td></tr>
<tr><td>从属模式时（SWSD＝0）
SSIWS　输入采样时序</td><td>SSISCK
上升沿</td><td>SSISCK
下降沿</td></tr>
<tr><td>主模式时（SWSD＝1）
SSIWS　输出变化时许</td><td>SSISCK
下降沿</td><td>SSISCK
上升沿</td></tr>
</table></td></tr>
<tr><td>12</td><td>SWSP</td><td>0</td><td>R/W</td><td>串行 WS 极性
0：SSIWS 在第 1 通道为低电平，在第 2 通道为高电平
1：SSIWS 在第 1 通道为高电平，在第 2 通道为低电平</td></tr>
<tr><td>11</td><td>SPDP</td><td>0</td><td>R/W</td><td>串行填充极性
0：填充位为低电平
1：填充位为高电平
【注】 当 MUEN=1 时，填充位变为低电平。（优先静噪功能）</td></tr>
</table>

位	位名	初始值	R/W	说　　明
10	SDTA	0	R/W	串行数据对齐 0：按串行数据、填充位的顺序发送 / 接收 1：按填充位、串行数据的顺序发送 / 接收
9	PDTA	0	R/W	并行数据对齐 数据的字长为 32、 16、 8 位时，该位无意义。 该位适用于接收模式时的 SSIRDR 和发送模式时的 SSITDR。 0：向左对齐并行数据 （SSITDR、 SSIRDR） 1：向右对齐并行数据 （SSITDR、 SSIRDR） • 当 DWL=000 （数据的字长为 8 位时），忽略 PDTA 设定 在串行音频总线使用 SSIRDR 或 SSITDR 的所有数据位。在各 32 位存取时发送 / 接收 4 个数据字。第 1 数据字保存在 bit7 ～ 0、第 2 数据字保存在 bit15 ～ 8、第 3 数据字保存在 bit23 ～ 16、第 4 数据字保存在 bit31 ～ 24。 • 当 DWL=001 （数据的字长为 16 位时），忽略 PDTA 设定 在串行音频总线使用 SSIRDR 或 SSITDR 的所有数据位。在各 32 位存取时发送 / 接收 2 个数据字。第 1 数据字保存在 bit15 ～ 0、第 2 数据字保存在 bit31 ～ 16。 • 当 DWL=010、011、100、101（数据的字长为 18、20、22、24 位时）、 PDTA=0 （向左对齐） 在 SSIRDR 或 SSITDR 的数据位使用以下位。 bit31 ～位 （通过 32-DWL 设定的数据字长的位数） 也就是，当 DWL=011 时，数据的字长为 20 位；使用 SSIRDR 或 SSITDR 的 bit31 ～ 12。其他所有的位被忽视或为保留位。 • 当 DWL=010、011、100、101（数据的字长为 18、20、22、24 位时）、 PDTA=1 （向右对齐） 在 SSIRDR 或 SSITDR 的数据位使用以下位。 bit （根据 DWL 设定的数据字长的位数 –1）～ bit0 也就是，当 DWL=011 时，数据的字长为 20 位；使用 SSIRDR 或 SSITDR 的 bit19 ～ 0。其他所有的位被忽视或为保留位。 • 当 DWL=110 （数据的字长为 32 位时），忽视 PDTA 设定 在串行音频总线可使用 SSIRDR 或 SSITDR 的所有数据位。
8	DEL	0	R/W	串行数据延迟 0：在 SSIWS 和 SSIDATA 之间有 1 个时钟周期的延迟 1：在 SSIWS 和 SSIDATA 之间无延迟
7	—	0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说　　明
6 ～ 4	CKDV[2:0]	000	R/W	串行过采样时钟分频比 设定过采样时钟 * 与串行位时钟的分频比。当 SCKD=0 时，忽视这些位。串行位时钟在移位寄存器中使用，并由 SSISCK 引脚提供。 000：串行位时钟频率 = 过采样时钟频率 /1 001：串行位时钟频率 = 过采样时钟频率 /2 010：串行位时钟频率 = 过采样时钟频率 /4 011：串行位时钟频率 = 过采样时钟频率 /8 100：串行位时钟频率 = 过采样时钟频率 /16 101：串行位时钟频率 = 过采样时钟频率 /6 110：串行位时钟频率 = 过采样时钟频率 /12 111：禁止设定 【注】 * 根据 PFC 内的 SCSR 的设定选择过采样时钟。请参照 “第 29 章 引脚功能控制器 （PFC）”。
3	MUEN	0	R/W	静噪允许 0：SSI 模块不为静噪状态 1：SSI 模块为静噪状态
2	—	0	R	保留位 读写的值总是 0。
1	TRMD	0	R/W	选择发送 / 接收模式 0：SSI 模块为接收模式 1：SSI 模块为发送模式
0	EN	0	R/W	SSI 模块允许 0：禁止 SSI 模块运行 1：允许 SSI 模块运行

18.3.2　状态寄存器 （SSISR）

SSISR 由表示 SSI 模块运行状态的状态标志和表示现在的通道编号与字编号的位构成。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	DMRQ	UIRQ	OIRQ	IIRQ	DIRQ	-	-	-	-	-	-	-	-
初始值：	0	0	0	0	0	0	1	0	不定	不定	不定	不定	不定	不定	不定	不定
R/W：	R	R	R	R	R/W*	R/W*	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
	-	-	-	-	-	-	-	-	-	-	-	-	CHNO[1:0]		SWNO	IDST
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	0	0	1	1
R/W：	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

【注】* 可读/写。写0时初始化该位，但忽略1的写入。

位	位名	初始值	R/W	说　明
31 ～ 29	—	均为 0	R	保留位 读写的值总是 0。
28	DMRQ	0	R	DMA 请求状态标志 根据该状态标志， CPU 可得知 SSI 模块的 DMA 请求状态。 ［TRMD=0 （接收模式）时］ • 当 DMRQ=1 时， SSIRDR 中有未读数据。 • 当读取 SSIRDR 时，到下一个未读数据出现为止 DMRQ=0。 ［TRMD=1 （发送模式）时］ • 当 DMRQ=1 时，为了能继续串行音频总线上的传送 SSITDR 请求写入数据。 • 给 SSITDR 写数据时，到下一个发送数据的请求出现为止 DMRQ=0。
27	UIRQ	0	R/W*	下溢错误中断状态标志 该状态标志表示用比请求速率低的速率提供数据。 与 UIEN 位的设定无关，该位被置 1。清 0 时，请写 0。 当 UIRQ=1 且 UIEN=1 时，产生中断。 ［TRMD=0 （接收模式）时］ 当 UIRQ=1 时，在 DMRQ 或 DIRQ 位显示存在有新的未读数据前，显示已读取 SSIRDR。此时，根据主机的不同，相同接收数据有可能被存储 2 次，且有可能导致多通道数据的破坏。 ［TRMD=1 （发送模式）时］ 当 UIRQ=1 时，表示在发送前没有向 SSITDR 写入发送数据。因此相同的数据的发送可能多于 1 次，且有可能导致多通道数据的破坏。其结果是输出错误的 SSI 数据，因此该错误比接收模式的下溢更严重。 【注】 如果发生下溢错误，到写入下一个数据为止，发送数据缓冲器中的数据。

位	位名	初始值	R/W	说　　明
26	OIRQ	0	R/W*	上溢错误中断状态标志 该状态标志表示用比请求速率高的速率提供数据 与 OIEN 位的设定无关，该位被置 1。清 0 时，请写 0 当 OIRQ=1 且 OIEN=1 时，产生中断。 ［TRMD=0 （接收模式）时］ 当 OIRQ=1 时，显示在向 SSIRDR 写入新的未读数据之前，还没有读出以前的未读数据。这样有可能损失数据，且有可能导致多通道数据的破坏。 【注】 如果发生上溢错误，从 SSI 接口发送来的下一个数据改写数据缓冲器中的数据。 ［当 TRMD=1 （发送模式）时］ 当 OIRQ=1 时，显示 SSITDR 的数据传送到移位寄存器前，已向 SSITDR 写入数据。因此可能损失数据，且有可能导致多通道数据的破坏。
25	IIRQ	1	R	空闲模式中断状态标志 该状态标志表示 SSI 模块是否为空闲状态。为了可以查询，与 IIEN 位的设定无关，将该位置 1。 通过将 IIEN 位清 0，可屏蔽中断，但即使该位写 0 也不能清除中断。 当 IIRQ=1 且 IIEN=1 时，产生中断。 0：SSI 模块不为空闲状态 1：SSI 模块为空闲状态
24	DIRQ	0	R	数据中断状态标志 该状态标志表示 SSI 模块需要读写数据。 为了可以查询，与 DIEN 位的设定无关，将该位置 1 通过将 DIEN 位清 0，可屏蔽中断，但即使该位写 0 也不能清除中断。 当 DIRQ=1 且 DIEN=1 时，产生中断。 ［当 TRMD=0 （接收模式）时］ 0：SSIRDR 没有未读数据 1：SSIRDR 有未读数据 ［当 TRMD=1 （发送模式）时］ 0：发送缓冲器满 1：发送缓冲器空，请求向 SSITDR 写数据
23 ～ 4	—	不定	R	保留位 读写的值总是 0。
3、2	CHNO[1:0]	00	R	通道编号 表示现在的通道。 ［当 TRMD=0 （接收模式）时］ 该位表示目前在 SSIRDR 内的数据是属于哪个通道的。通过移位寄存器的传送更新 SSIRDR 时，该值发生变化。 ［当 TRMD=1 （发送模式）时］ 该位表示应向 SSITDR 写入哪个通道的数据。如果数据复制到移位寄存器，与是否写入 SSITDR 无关，该值发生变化。

位	位名	初始值	R/W	说　　明
1	SWNO	1	R	串行字编号 表示目前的字编号。 ［当 TRMD=0 （接收模式）时］ 该位表示现在 SSIRDR 内的数据为哪个系统字。与是否读取 SSIRDR 无关，通过移位寄存器的传送更新 SSIRDR 时，该值发生变化。 ［当 TRMD=1 （发送模式）时］ 该位表示该向 SSITDR 写入哪个系统字。如果数据复制到移位寄存器，与是否写入 SSITDR 无关，该值发生变化。
0	IDST	1	R	空闲模式状态标志 该状态标志表示串行总线为停止状态。 当 EN=1 且串行总线为运行中时，该位被清除。 该位为以下条件时自动置 1。 ［当 SSI 为主传送器 （SWSD=1 且 TRMD=1）时］ 清除 EN 位，如果写入 SSITDR 的数据从串行数据输入 / 输出引脚（SSIDATA）输出结束 （如果结束系统字长的输出），该位就置 1。 ［当 SSI 为主接收器 （SWSD=1 且 TRMD=0）时］ 如果清除 EN 位，结束目前的系统字，该位就置 1。 ［当 SSI 为从属传送器 / 接收器 （SWSD=0）时］ 如果清除 EN 位，结束目前的系统字，该位就置 1。 【注】 在结束目前的系统字前，如果外部器件停止串行总线时钟，该位就不被置位。

【注】 * 可读 / 写。如果写 0，该位被初始化，但忽略 1 的写入。

18.3.3　发送数据寄存器 （SSITDR）

SSITDR 是 32 位寄存器，保存发送数据。

如果有发送请求，就将写入该寄存器的数据传送到移位寄存器。数据的字长未满 32 位时，根据 SSICR 的 PDTA 控制位的设定进行对齐。

通过读该寄存器，可获得缓冲器的数据。

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

18.3.4　接收数据寄存器 （SSIRDR）

SSIRDR 为 32 位的寄存器，保存接收数据。

每次接收数据字，就从移位寄存器传送该寄存器的数据。数据的字长未满 32 位时，根据 SSICR 的 PDTA 控制位的设定进行对齐。

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
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

18.4　运行说明

18.4.1　总线格式

SSI 模块可作为发送器也可作为接收器运行，任何模式都可使用多个串行总线格式。

总线格式可从 8 个主要模式中选择，如表 18.3 所示。

表 18.3　SSI 模块的总线格式

	TRMD	SCKD	SWSD	EN	MUEN	DIEN	IIEN	OIEN	UIEN	DEL	PDTA	SDTA	SPDP	SWSP	SCKP	SWL[2:0]	DWL[2:0]	CHNL[1:0]
非压缩从属接收器	0	0	0	控制位						结构位								
非压缩从属发送器	1	0	0															
非压缩主接收器	0	1	1															
非压缩主发送器	1	1	1															

18.4.2　非压缩模式

非压缩模式支持分配至通道的串行音频流。除了飞利浦、索尼或松下模式外，还支持多个改良版本。

（1） 从属接收器

在该模式中可从其他器件接收串行数据。从外部器件提供用于串行数据流的时钟和字选择信号。当这些信号与 SSI 模块设定的格式不同时，无法保证运行。

（2） 从属发送器

在该模式中可向其他器件发送串行数据。从外部器件提供用于串行数据流的时钟和字选择信号。当这些信号与 SSI 模块设定的格式不同时，无法保证运行。

（3） 主接收器

在该模式中可从其他器件接收串行数据。由过采样时钟在内部生成时钟和字选择信号。这些信号的格式取决于 SSI 模块的设定。其他器件发送的数据与设定的格式不同时，无法保证运行。

（4） 主发送器

在该模式中可向其他器件发送串行数据。由过采样时钟在内部生成时钟和字选择信号。这些信号的格式取决于 SSI 模块结构位的设定。

（5） 运行设定 - 字长的关系

在非压缩模式中，与 SSICR 的字长有关的所有位为有效。 SSI 模块可支持多个结构，以下对飞利浦、索尼、松下的格式进行说明。

- 飞利浦格式

有填充与无填充的飞利浦格式分别如图 18.3、图 18.4 所示。数据字长比系统字长短时，产生填充。

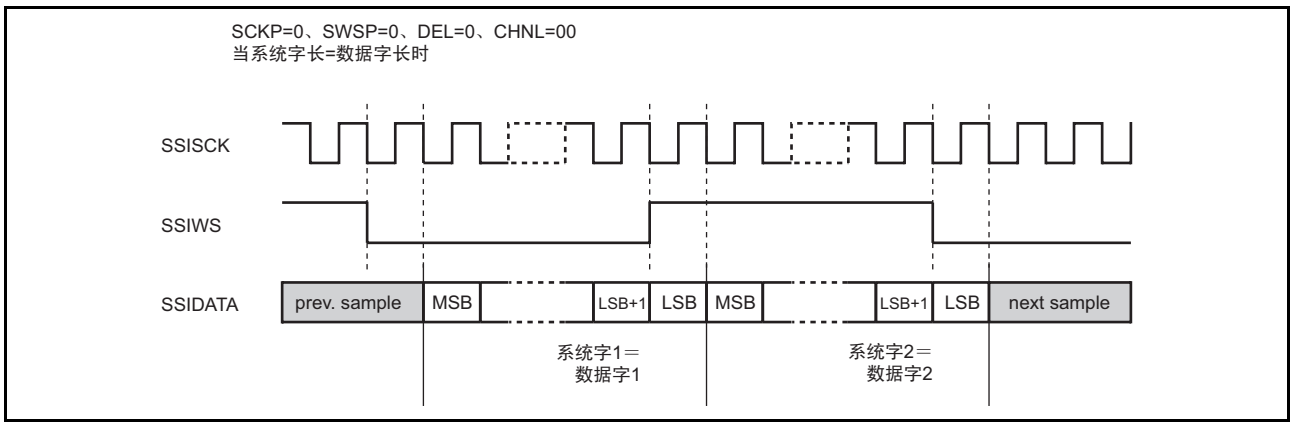


图 18.3　飞利浦格式 （无填充）

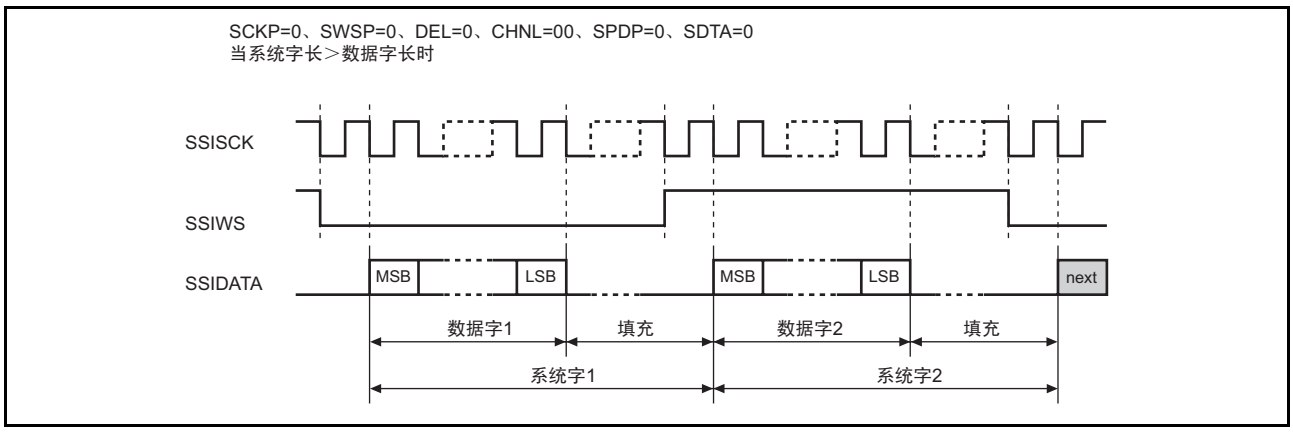


图 18.4　飞利浦格式 （有填充）

索尼和松下的格式分别如图 18.5、图 18.6 所示。都是有填充的例子，但当系统的字长和数据的字长相同时会出现无填充的情况。

- 索尼格式

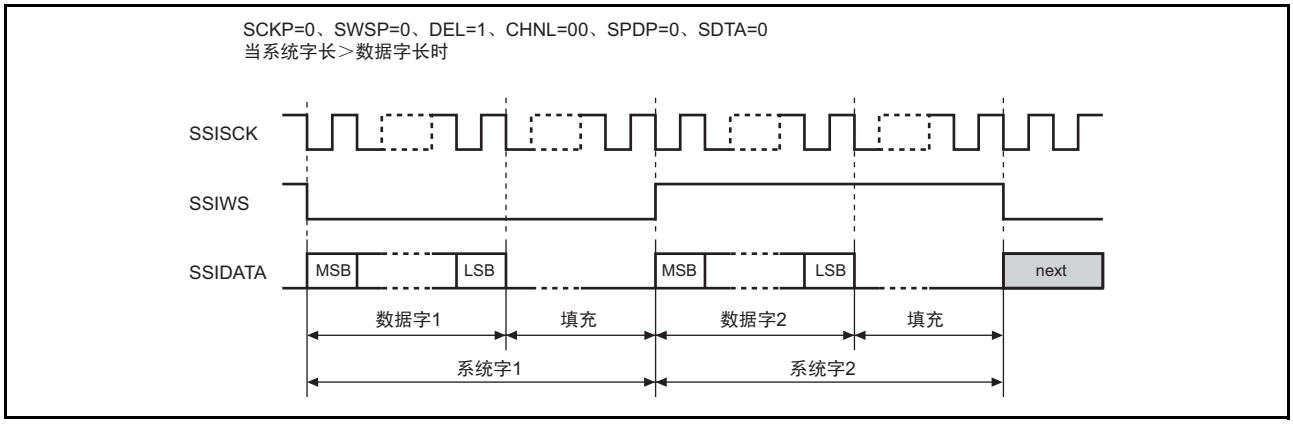


图 18.5　索尼格式 （按串行数据、填充位的顺序发送 / 接收）

- 松下格式

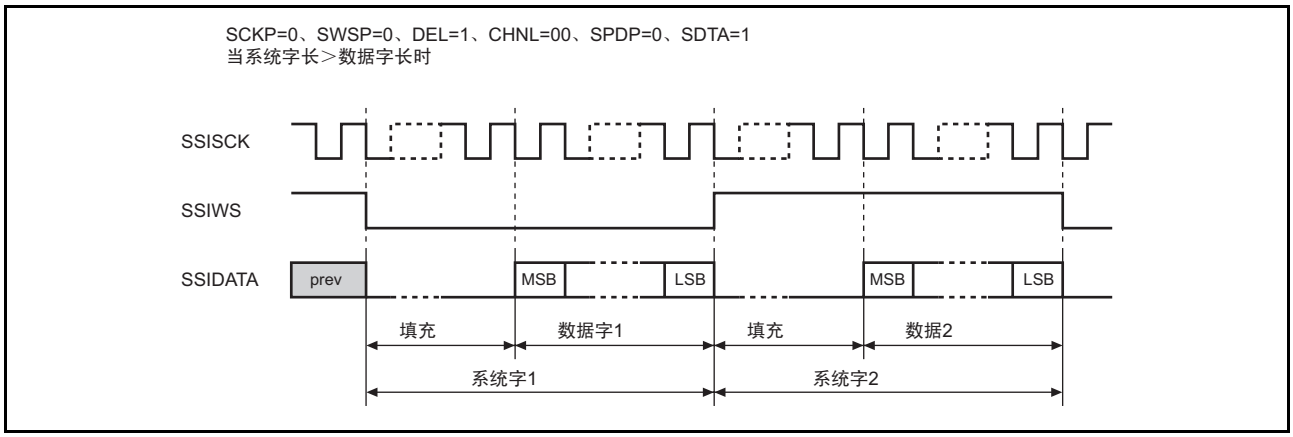


图 18.6　松下格式 （按填充位、串行数据的顺序发送 / 接收）

（6） 多通道格式

有些器件扩展飞利浦规格的定义，在 2 个系统字中进行超过 2 个通道的传送。

SSI 模块使用 CHNL、SWL 和 DWL 位执行 4、6、8 个通道的传送。但这种情况仅限于系统字长 （SWL）不小于数据字长 （DWL）与通道数 （CHNL）的乘积。

有效设定和填充位数如表 18.4 所示。无效设定代替数字用 “—” 表示。

表 18.4　有效的设定和填充位数

每个系统字的填充位数			DWL[2:0]	000	001	010	011	100	101	110
CHNL [1:0]	每个系统字解码的通道	SWL[2:0]	解码的字长	8	16	18	20	22	24	32
00	1	000	8	0	—	—	—	—	—	—
		001	16	8	0	—	—	—	—	—
		010	24	16	8	6	4	2	0	—
		011	32	24	16	14	12	10	8	0
		100	48	40	32	30	28	26	24	16
		101	64	56	48	46	44	42	40	32
		110	128	120	112	110	108	106	104	96
		111	256	248	240	238	236	234	232	224
01	2	000	8	—	—	—	—	—	—	—
		001	16	0	—	—	—	—	—	—
		010	24	8	—	—	—	—	—	—
		011	32	16	0	—	—	—	—	—
		100	48	32	16	12	8	4	0	—
		101	64	48	32	28	24	20	16	0
		110	128	112	96	92	88	84	80	64
		111	256	240	224	220	216	212	208	192
10	3	000	8	—	—	—	—	—	—	—
		001	16	—	—	—	—	—	—	—
		010	24	0	—	—	—	—	—	—
		011	32	8	—	—	—	—	—	—
		100	48	24	0	—	—	—	—	—
		101	64	40	16	10	4	—	—	—
		110	128	104	80	74	68	62	56	32
		111	256	232	208	202	196	190	184	160
11	4	000	8	—	—	—	—	—	—	—
		001	16	—	—	—	—	—	—	—
		010	24	—	—	—	—	—	—	—
		011	32	0	—	—	—	—	—	—
		100	48	16	—	—	—	—	—	—
		101	64	32	0	—	—	—	—	—
		110	128	96	64	56	48	40	32	0
		111	256	224	192	184	176	168	160	128

SSI 模块作为发送器运行时，写入 SSITDR 的各字，按照写入的顺序发送到串行音频总线。SSI 模块作为接收器运行时，串行音频总线接收的各字，按照从 SSIRDR 接收的顺序读出。

4、6、8 个通道的数据是如何传送到串行音频总线的如图 18.7 ～图 18.9 所示。图 18.7 为没有填充位时，图 18.8 为向左对齐时，图 18.9 为向右对齐。这些例子都是任意的。

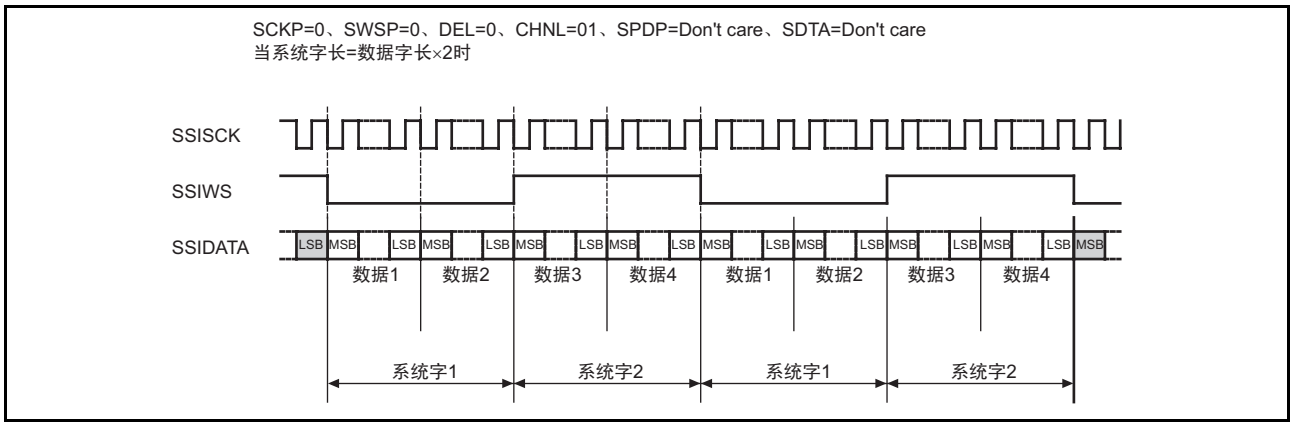


图 18.7　多通道格式 （4 个通道、无填充）

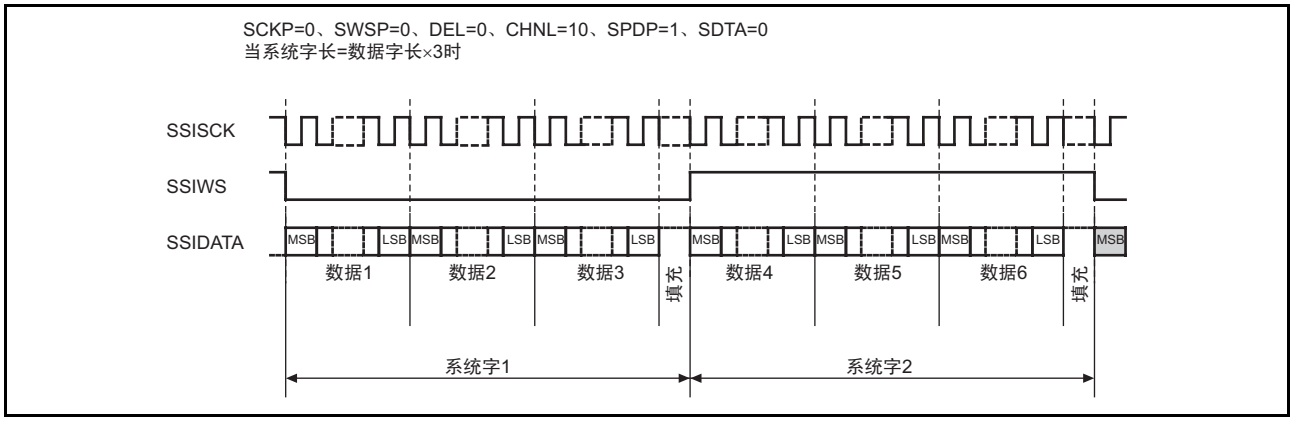


图 18.8　多通道格式 （6 个通道、 High 填充）

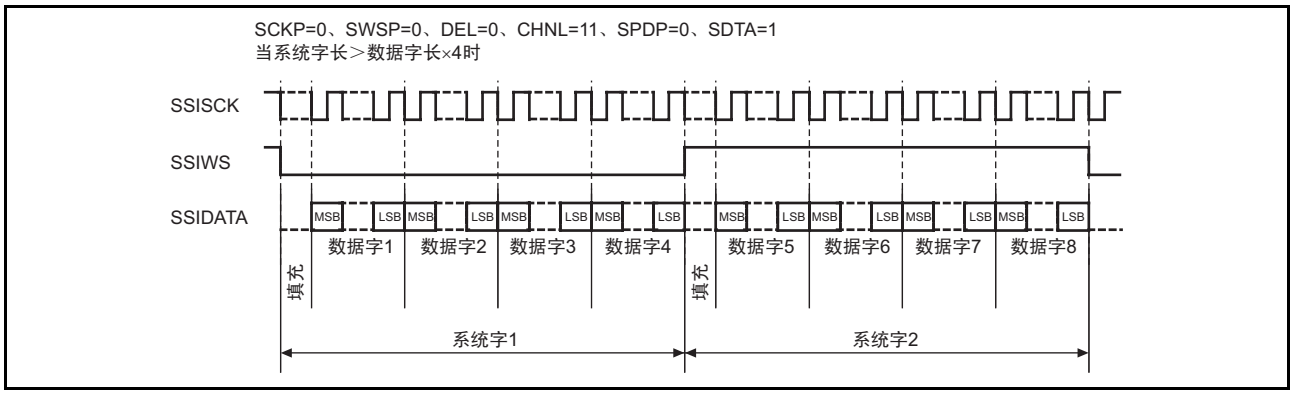


图 18.9　多通道格式 （按 8 个通道、填充位、串行数据的顺序发送 / 接收，有填充）

（7） 运行设定格式设定位

非压缩模式的其他结构位如下所示。这些位不相互排斥，但组合使用时，可能不实用。
参照图 18.10 的基本格式例，对这些结构位进行如下说明。

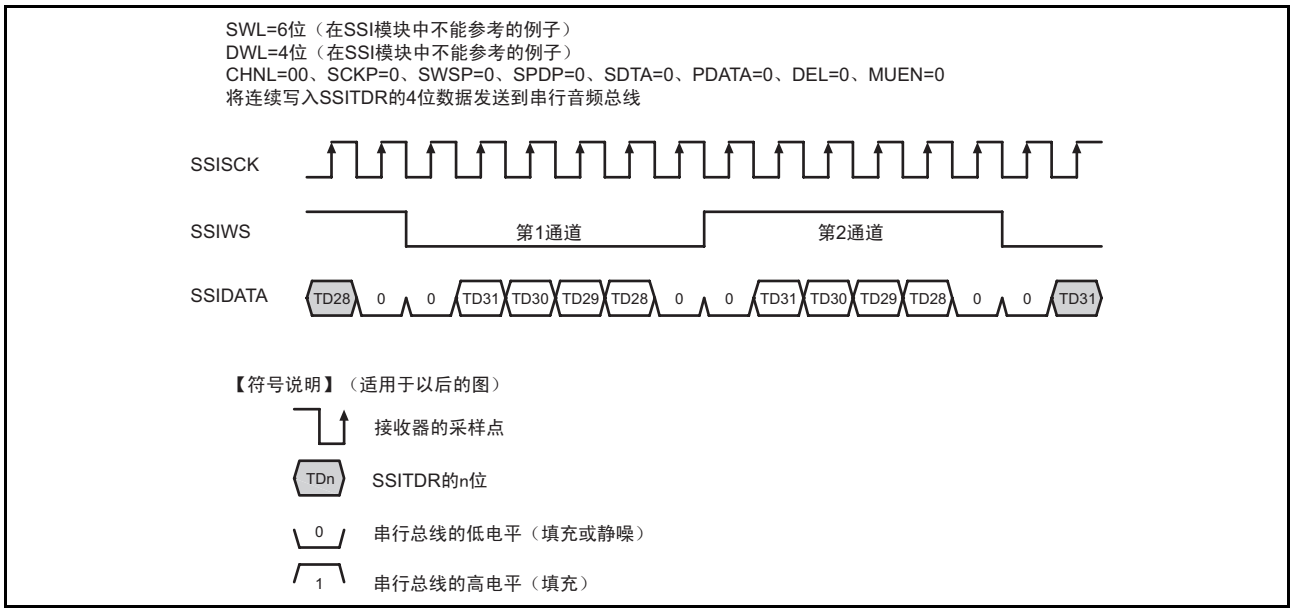


图 18.10　基本格式例 （发送模式、任意系统 / 数据的字长）

在图 18.10 的例子中，使用 6 位的系统字和 4 位的数据字。这些设定在 SSI 模块中无法实现，在这里举例是为了说明其他的设定位。

- 反相时钟

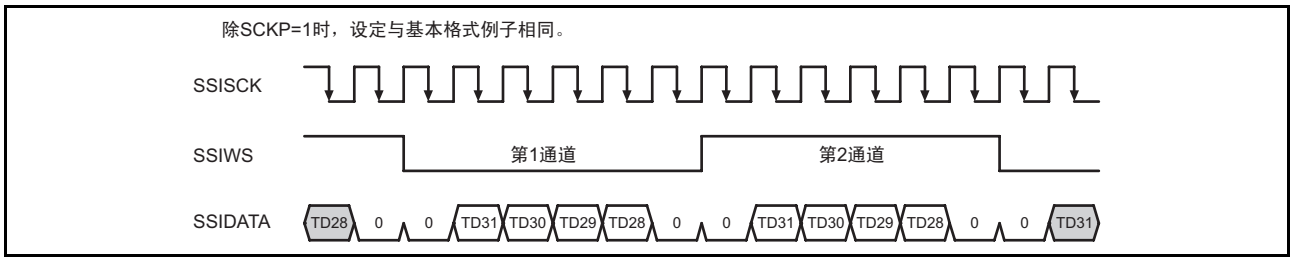


图 18.11　反相时钟

- 反相字选择信号

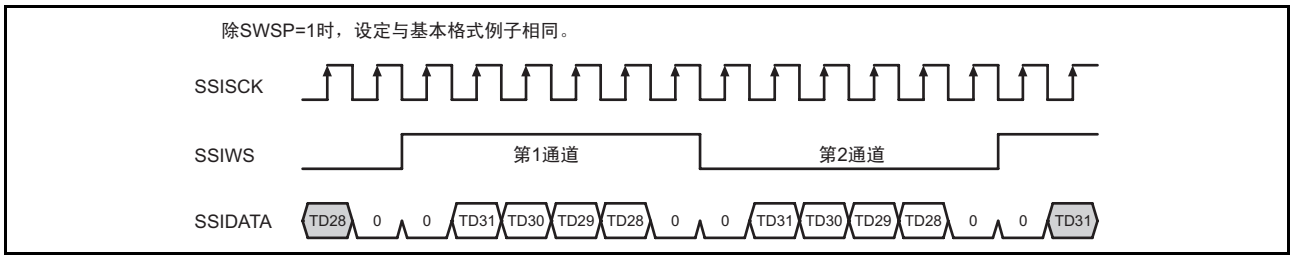


图 18.12　反相字选择信号

- 反相填充极性

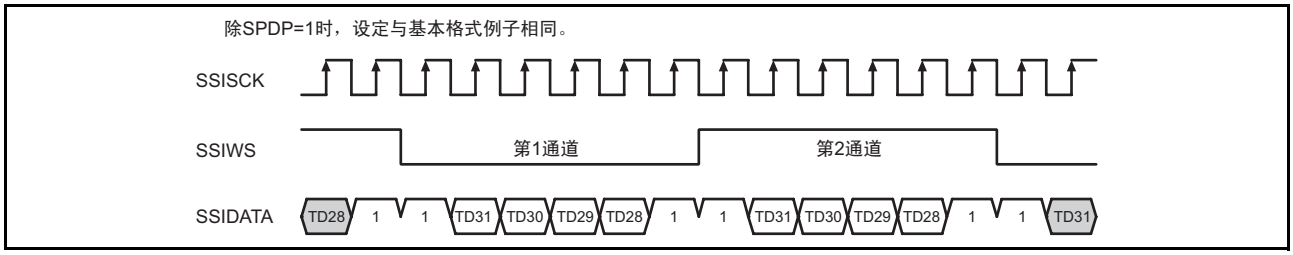


图 18.13　反相填充极性

- 按照填充位、串行数据的顺序发送/接收，有延迟

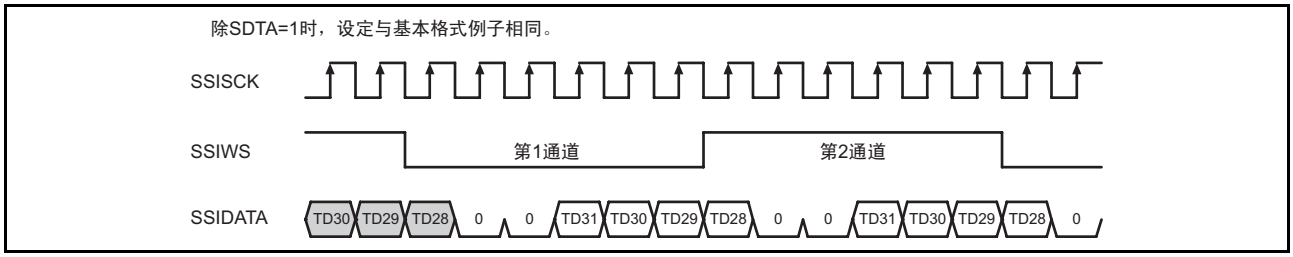


图 18.14　按照填充位、串行数据的顺序发送 / 接收，有延迟

- 按照填充位、串行数据的顺序发送/接收，无延迟

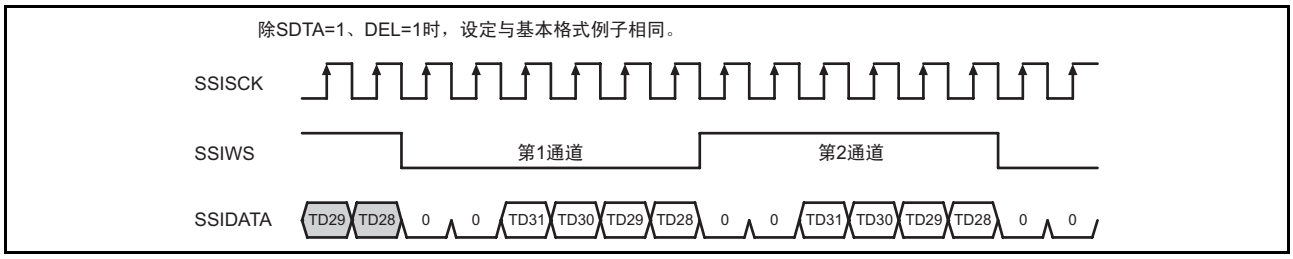


图 18.15　按照填充位、串行数据的顺序发送 / 接收，无延迟

- 按照串行数据、填充位的顺序发送/接收，无延迟

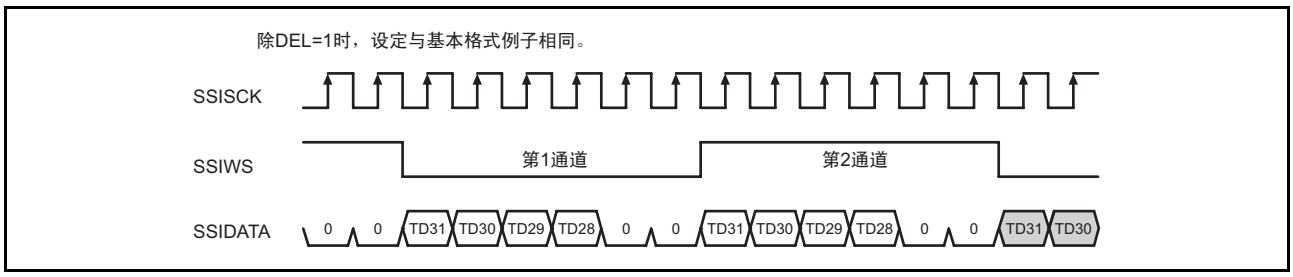


图 18.16　按照串行数据、填充位的顺序发送 / 接收，无延迟

- 填充数据的向右对齐，有延迟

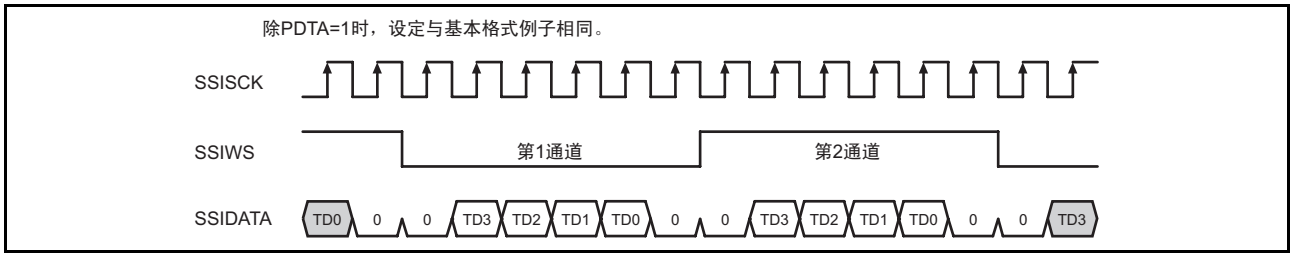


图 18.17　填充数据的向右对齐，有延迟

- 静噪有效

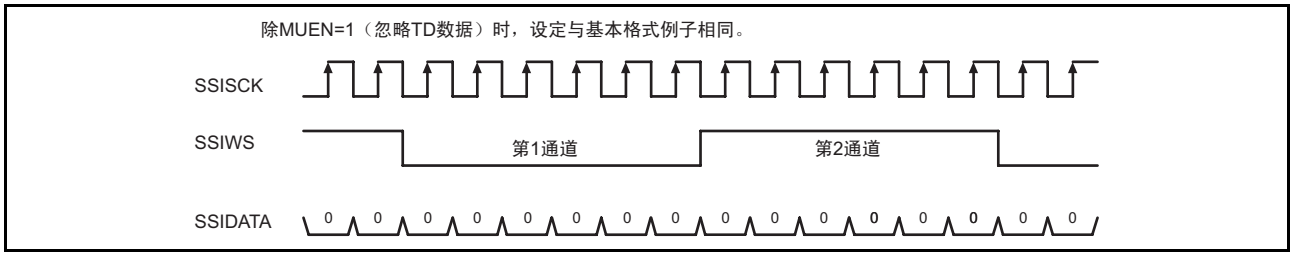


图 18.18　静噪有效

18.4.3　运行模式

有结构、有效和无效 3 种运行模式。运行模式的转移图如图 18.19 所示。

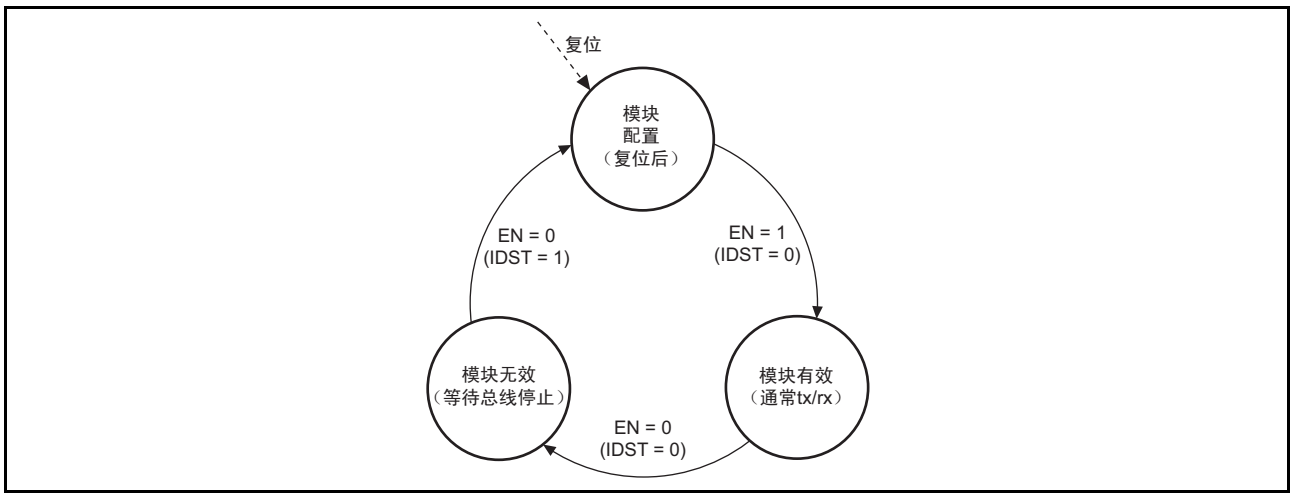


图 18.19　运行模式的转移图

（1） 结构模式

复位解除后进入该模式。在 SSI 模块通过设置 EN 位变为有效前，必须在该模式对控制寄存器进行必要的设定。

如果将 EN 位置位， SSI 模块转移为模块有效模式。

（2） 模块有效模式

该模式的运行依赖于所选择的运行模式。详细情况请参照 “18.4.4 发送运行”和 “18.4.5 接收运行”。

18.4.4　发送运行

可通过 DMA 或中断控制发送。

通过 DMA 控制，有利于减少 CPU 负荷。在 DMA 控制模式中，发生数据的下溢或上溢时，或 DMAC 传送结束时，通过中断通知。

也有根据需要，使用中断 （SSI 模块为提供数据而生成的中断）的控制方法。由于 SSI 模块仅为双缓冲器结构，而且至少需要在每个系统字写入数据，因此就会在中断控制模式产生高负荷。

SSI 模块无效时，必须持续提供 SSI 时钟 * 直到 IIRQ 位显示 SSI 为空闲状态。

DMA 控制模式的发送运行和中断控制模式的发送运行分别如图 18.20、图 18.21 所示。

【注】 *　当 SCKD=0 时，来自 SSISCK 引脚的输入时钟
　　当 SCKD=1 时，过采样时钟

（1） 使用 DMA 控制器的发送

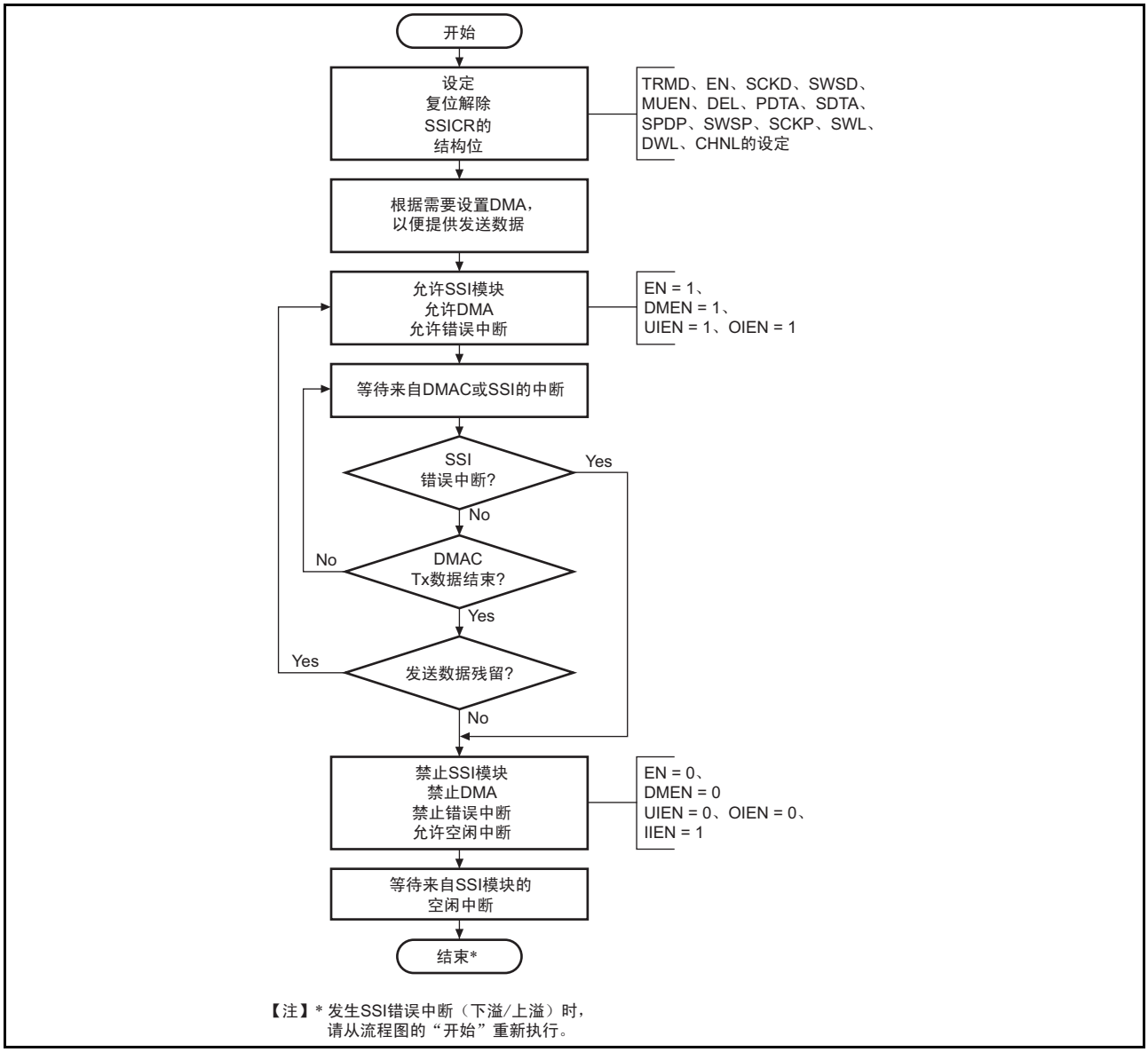


图 18.20　使用 DMA 控制器的发送

（2） 使用中断数据流程控制的发送

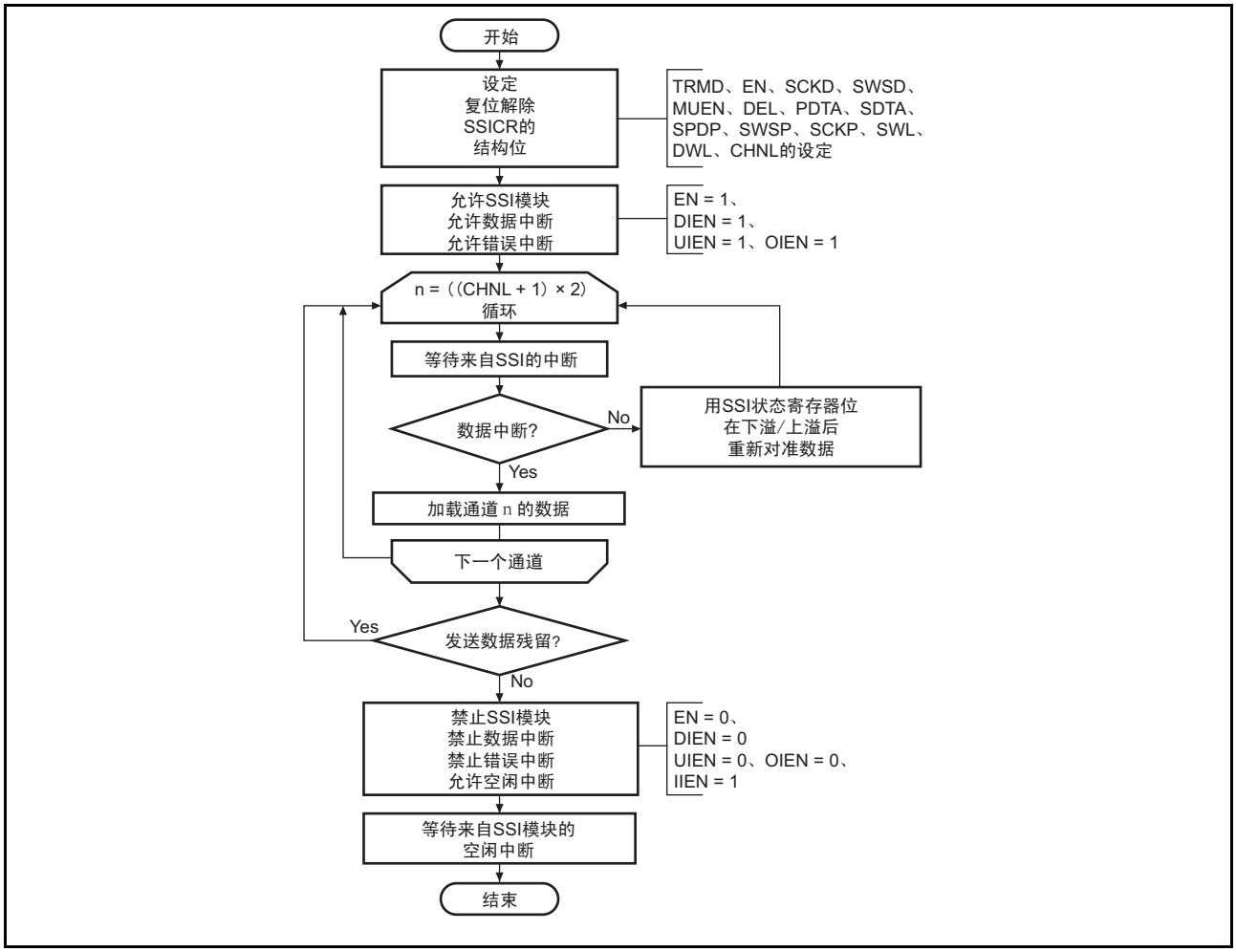


图 18.21　使用中断数据流程控制的发送

18.4.5　接收运行

与发送相同，也能通过 DMA 或中断控制接收。

各运行的流程图分别如图 18.22、图 18.23 所示。

SSI 模块无效时，必须持续提供 SSI 时钟 * 直到 IIRQ 位显示 SSI 为空闲状态。

【注】 *　当 SCKD=0 时，来自 SSISCK 引脚的输入时钟
　　当 SCKD=1 时，过采样时钟

（1） 使用 DMA 控制器的接收

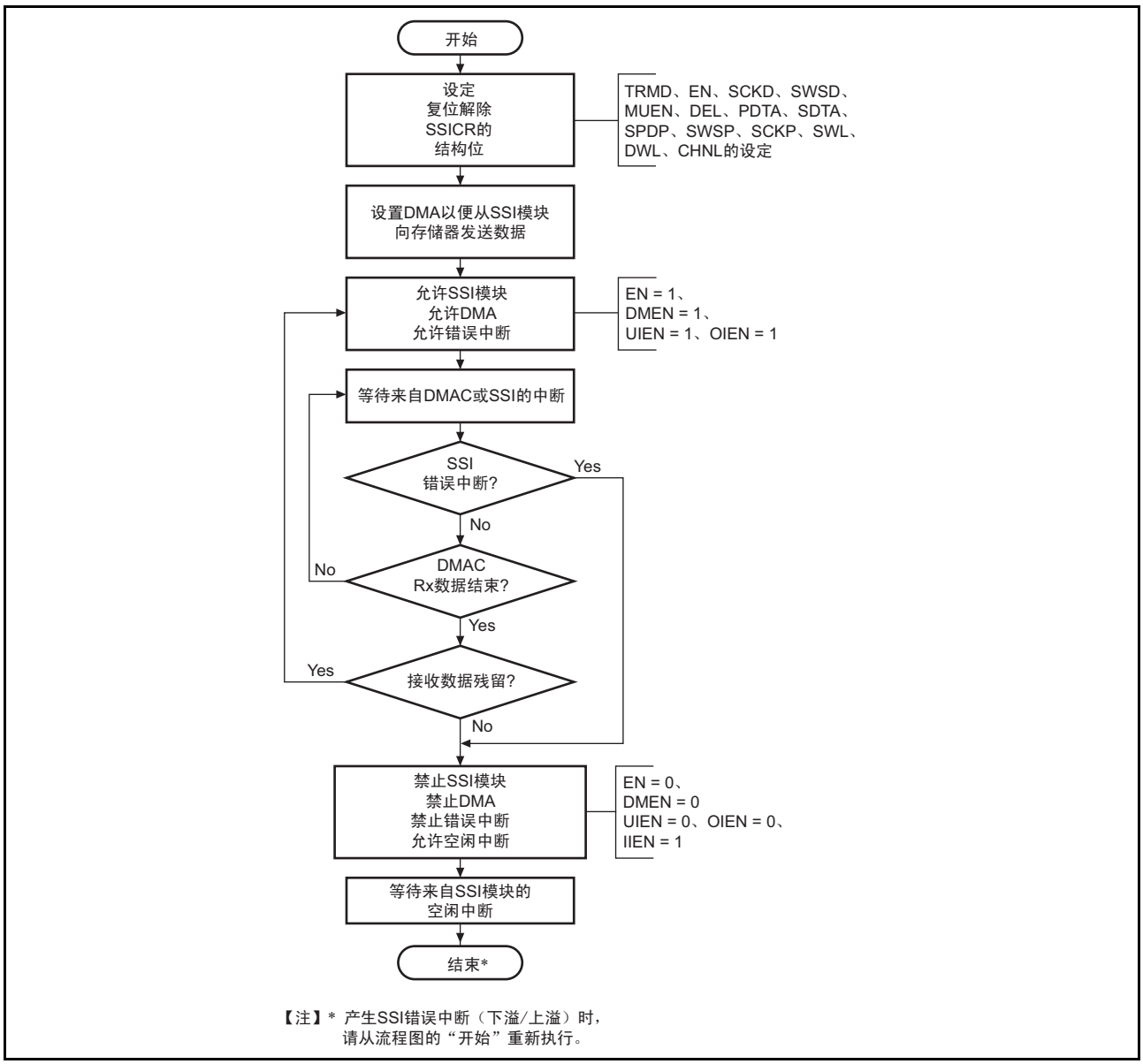


图 18.22　使用 DMA 控制器的接收

（2） 使用中断数据流程控制的接收

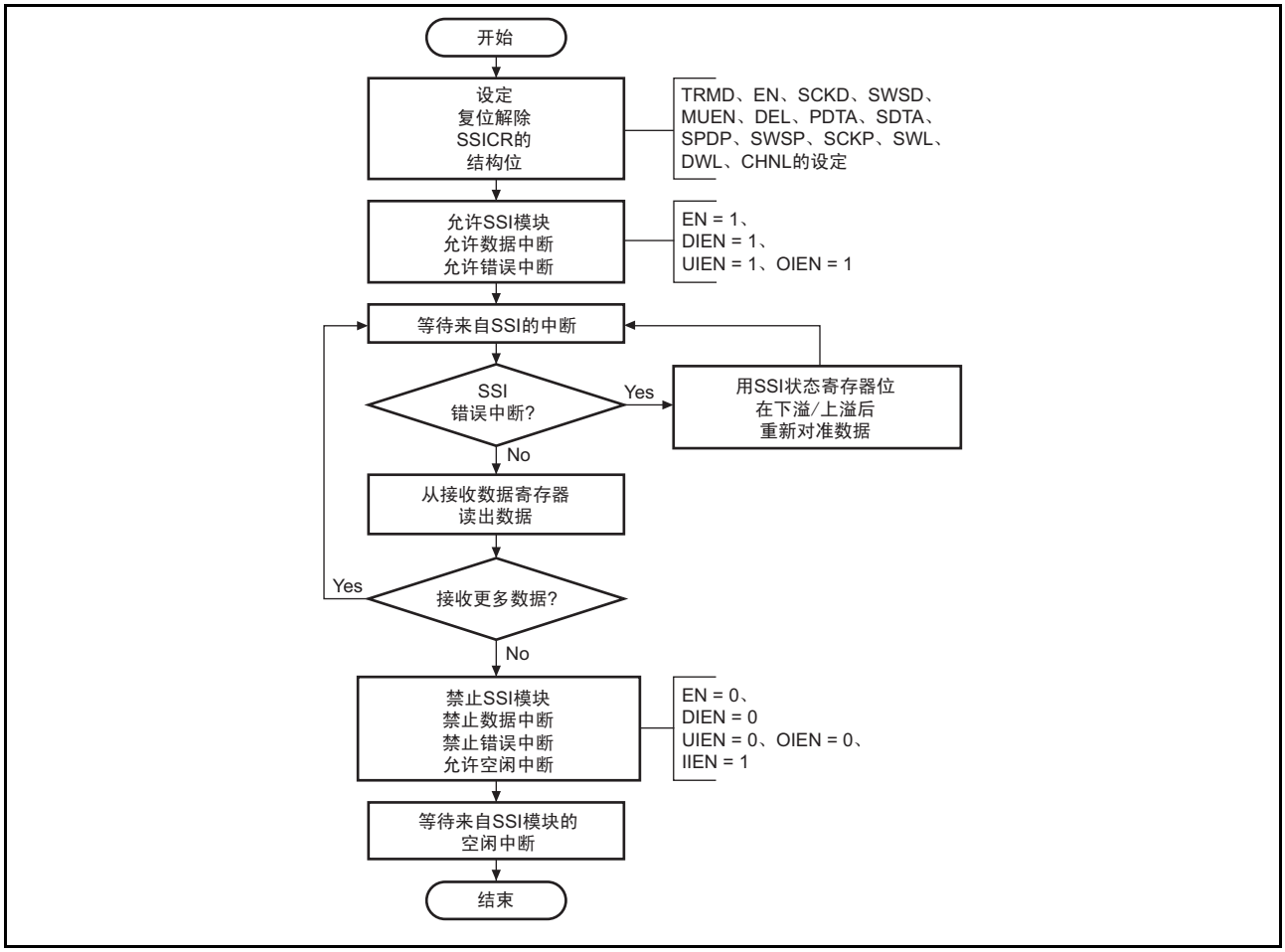


图 18.23　使用中断数据流程控制的接收

当下溢或上溢条件匹配时，可使用 CHNO[1:0] 位和 SWNO 位使 SSI 模块恢复到匹配前的状态。如果发生下溢或上溢，通过主机 CPU 读出通道数和系统字数，可得知串行音频流已到达的位置。作为发送器运行时，主机 CPU 在 SSI 模块到达下一次预计发送的数据前，可跳过发送数据。这样就能和音频数据流再次同步。作为接收器运行时，主机 CPU 在 SSI 模块到能储存表示下次接收的数据前，通过储存无效数据，取得接收数据的匹配性，能和音频数据流再次同步。

18.4.6 在发送时的暂时停止、重新开始步骤

能通过以下的步骤实现。

（1） 不重新设定 DMAC 而重复传送、停止的步骤

1. 为停止DMA传送，使SSICR.DMEN=0 （禁止DMA请求）。
2. 通过查询或中断等，等待SSISR.DIRQ=1 （发送模式：发送缓冲器空）。
3. 通过SSICR.EN=0 （禁止SSI模块运行），停止传送。
4. 重新传送前，确认SSISR.IDST=1。
5. 使SSICR.EN=1 （允许SSI模块运行）。
6. 通过查询或中断等，等待SSISR.DIRQ=1。
7. 通过使SSICR.DMEN=1 （允许DMA请求），重新开始DMA传送。

（2） SSI 停止后，重新设定 DMAC 时的传送步骤

1. 为停止DMA传送，使SSICR.DMEN=0 （禁止DMA请求）。
2. 通过查询或中断等，等待SSISR.DIRQ=1 （发送模式：发送缓冲器空）。
3. 通过SSICR.EN=0 （禁止SSI模块运行），停止传送。
4. 通过DMAC的CHCR，停止DMAC。
5. 重新传送前，确认SSISR.IDST=1。
6. 使SSICR.EN=1 （允许SSI模块运行）。
7. 设定DMAC的各寄存器和开始传送。
8. 通过SSICR.DMEN=1 （允许DMA请求），重新开始DMA传送。

18.4.7 串行位时钟控制

使用串行位时钟的功能，控制和选择在串行总线接口使用的时钟。

如果串行位时钟的方向设定为输入（SCKD=0），SSI 模块就为时钟从属模式，移位寄存器使用的位时钟是输入到 SSISCK 引脚的时钟。

如果串行位时钟的方向设定为输出（SCKD=1），SSI 模块就为时钟主模式，移位寄存器使用的位时钟是过采样时钟或将位时钟分频的时钟。通过 SSICR 的串行过采样时钟分频比 （CKDV）位设定的比率，过采样时钟在分频后用作移位寄存器的位时钟。

在上述的任何情况下， SSISCK 引脚的输出都和位时钟相同。

18.5 使用时的注意事项

18.5.1 在接收 DMA 运行中溢出时的限制事项

在接收 DMA 运行中发生溢出时，需要重新启动模块。SSI 内的接收缓冲器由 L 通道和 R 通道共享的 32 位寄存器构成。因此，通过将控制寄存器 （SSICR）的数据字长 （DWL2 ～ DWL0）和系统字长 （SWL2 ～ DWL0）设定为 32 位时，一旦发生上溢，就会发生在 R 通道接收到了应该在 L 通道接收的数据。

在这里，通过溢出错误中断或溢出错误状态标志（SSISR 的 OIRQ 位）确认溢出时，请通过向 SSICR 的 EN 位和 DMEN 位写 0，禁止 SSI 模块的 DMA，并停止运行 （此时也请停止 DMA 控制器的设定）。此后， OIRQ 位写 0，清除溢出状态，重新设定 DMA 并开始传送。

18.5.2 使用过采样时钟时的注意事项

外部时钟输入用作过采样时钟时，请参照 “4.6.1 外部时钟输入时的注意事项”将 EXTAL、 XTAL 引脚替换为 AUDIO_X1、 AUDIO_X2 引脚。

使用晶体振荡器时，请参照 “4.6.2 晶体振荡器使用时的注意事项”将 EXTAL、 XTAL 引脚替换为 AUDIO_X1、 AUDIO_X2 引脚。

另外，请参照 “4.6.3 有关振荡器的注意事项”。

第 19 章　控制器局域网 （RCAN-TL1）

控制器局域网 （RCAN-TL1：Renesas CAN Time Trigger Level 1）为用于控制以汽车及工业器件系统等实时通信为目标的 CAN （Controller Area Network）的模块。

本章说明 RCAN-TL1 的程序接口。

另外，关于 CAN 的数据链路控制器的功能请参考下述参考文献：

［参考文献］

1. CAN Specification Version 2.0 part A, Robert Bosch GmbH, 1991
2. CAN Specification Version 2.0 part B, Robert Bosch GmbH, 1991
3. Implementation Guide for the CAN Protocol, CAN Specification 2.0 Addendum, CAN In Automation, Erlangen, Germany, 1997
4. Road vehicles-Controller area network （CAN） : Part 1: Data link layer and physical signaling （ISO-11898-1, 2003）
5. Road vehicles - Controller area network （CAN） : Part 4: Time triggered communication （ISO-11898-4, 2004）

19.1　特点

19.1.1　RCAN-TL1 的特点

- 支持CAN规格2.0B
- 位时序根据ISO-11898规格
- 32个邮箱
- 时钟频率：16～33MHz
- 31个可编程发送/接收用邮箱及1个接收用邮箱
- 检测低功耗的CAN睡眠模式及CAN总线激活，自动解除CAN的睡眠模式
- 对应所有邮箱的可编程接收过滤器屏蔽 （标准ID及扩展ID）
- 最大1Mbps的可编程CAN数据率
- 为防止实时应用所引起的优先级反转，备有优先级内部区分功能的传送信息队列
- 丰富的中断源
- 内置测试功能 （接收专用模式、错误被动模式）
- 16位的自由运行定时器 （多种时钟源、预分频器、 3个定时器比较匹配寄存器）
- 时间触发发送用6位周期计数器 （Basic Cycle）
- 通过定时器的比较匹配寄存器产生的中断
- 可进行定时器计数器的清除及设定
- 时间触发用寄存器：Local_Time、 Cycle_time、 Ref_Mark、 Tx_Enable_Window、 Ref_Trigger_Offset
- 发送/接收均支持SOF灵活的时戳功能
- 支持时间触发发送及定周期发送 （包括事件触发发送）
- 可将周期计数器 （Basic Cycle）嵌入CAN帧后进行发送

19.1.2　本单片机的特点

- 具有32个邮箱RCAN×2个通道 （RCAN0、 1）
- 可通过设定引脚功能控制器 （PFC），采用以下连接使用RCAN的2个通道
  32个邮箱×2个通道 （RCAN0、 1）
  64个邮箱*×1个通道 （RCAN0、 1）
- RCAN0、 1均可通过邮箱0的接收信息启动DMAC

【注】 * 在 64 个邮箱的构成方面，有使用时的注意事项，请参考 “19.12 使用时的注意事项”。

19.2　构成

19.2.1　框图

RCAN-TL1 由支持 CAN2.0B Active 与 ISO-11898 的 CAN 帧构成，提供控制的自由度大的简洁的方法。RCAN-TL1 在功能上由微处理单元接口 （MPI）、邮箱、邮箱控制、定时器及 CAN 接口等 5 种块构成。

RCAN-TL1 的框图如图 19.1 所示。

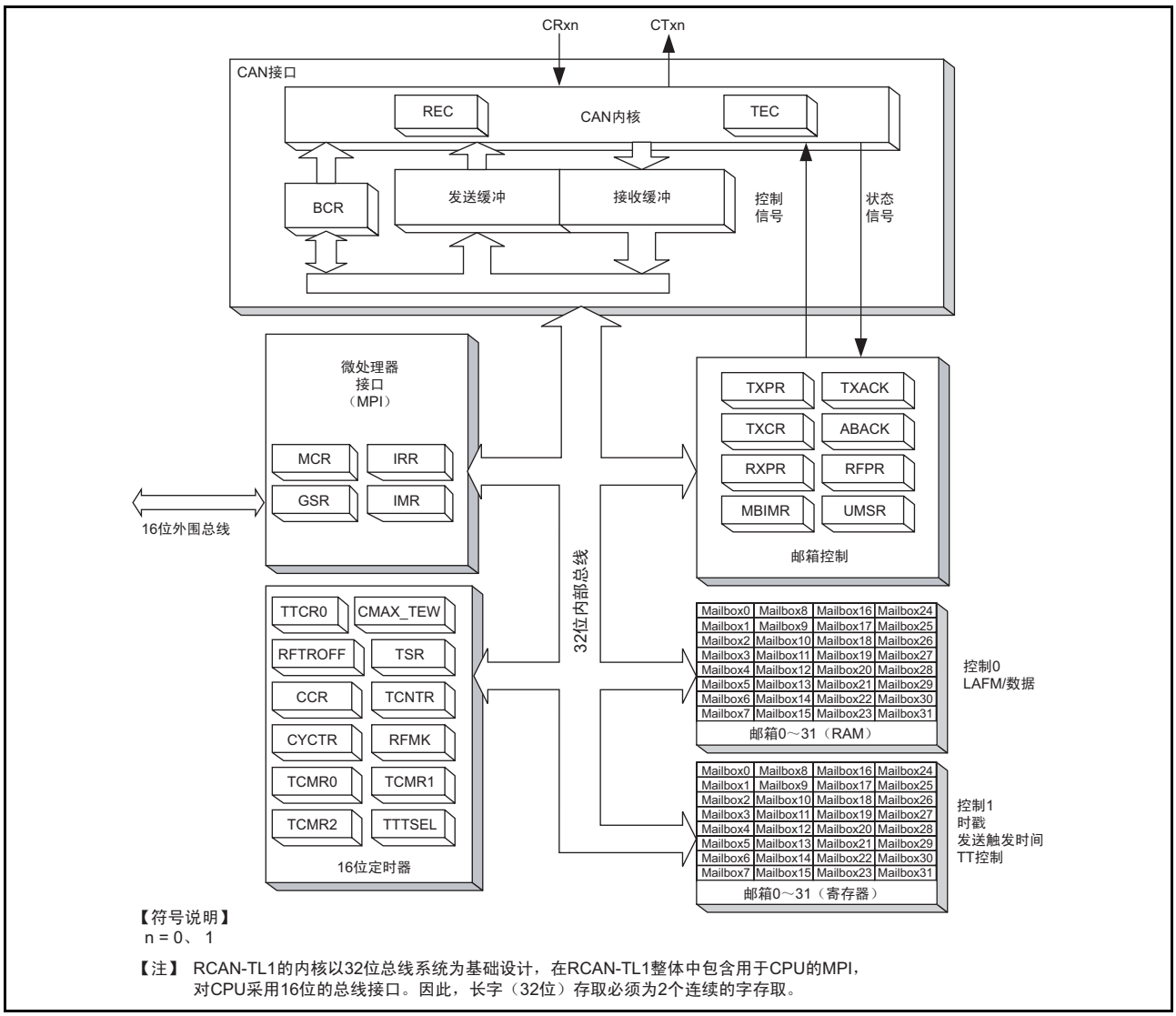


图 19.1　RCAN-TL1 的框图 （每 1 个通道）

19.2.2　各块的功能

（1） 微处理单元接口 （MPI）

MPI 可进行 CPU 与 RCAN-TL1 的寄存器或邮箱之间的通信，并控制存储器接口。另外，MPI 检测 CAN 总线的激活，MPI 或 RCAN-TL1 的其他模块还具有通知 CAN 总线激活的唤醒控制逻辑，所以 RCAN-TL1 可自动解除 CAN 睡眠模式。而且 MPI 的寄存器中具有 MCR、 IRR、 GSR 及 IMR。

（2） 邮箱

邮箱配置在 RAM 及寄存器中作为信息缓冲器。
RAM 与寄存器中分别具有 32 个邮箱，保存以下信息：

［RAM］
- CAN信息控制 （ID、 RTR、 IDE等）
- CAN信息数据 （CAN数据帧用）
- 接收用的本地接收过滤器屏蔽 （LAFM）

［寄存器］
- CAN信息控制 （DLC）
- 信息发送/接收用时戳
- 3位邮箱结构、自动再发送无效位、远程请求用自动发送位、新信息控制位
- 发送触发时间

（3） 邮箱控制

邮箱控制具有以下功能。
- 接收信息时，将ID进行比较，生成用于将来自CAN接口的信息保存至邮箱的RAM地址与数据，并设置或清除对应的寄存器。
- 发送事件触发信息时，进行内部仲裁并选择正确的优先级的信息，从邮箱将信息加载至CAN接口的发送缓冲器。此后，置位或清除对应的寄存器。时间触发发送时，通过发送触发时间的比较匹配加载信息。
- 进行CPU与邮箱控制之间的邮箱存取的仲裁
- 寄存器具有TXPR、 TXCR、 TXACK、 ABACK、 RXPR、 RFPR、 MBIMR及UMSR。

（4） 定时器

本定时器为用于在特定的时间范围内发送信息、记录结果的功能块。为 16 位自由运行递增计数器，可通过 CPU 进行控制。具有 1 个同本地时间进行比较的 16 位比较匹配寄存器与 2 个同周期时间进行比较的比较匹配寄存器。这些比较匹配寄存器可产生中断信号、并且清除计数器。本定时器的时钟可从系统时钟中生成的多个时钟周期中进行选择，但也可通过编程用 CAN 总线的 1 位时序进行计数。本定时器的寄存器中具有 TCNTR、TTCR0、CMAX_TEW、RETROFF、TSR、CCR、CYCTR、RFMK、TCMR0、TCMR1、TCMR2 及 TTTSEL。

（5） CAN 接口

本块支持参考文献 [2] 与 [4] 的 CAN 总线数据链路控制器规格。它满足 OSI 模型所规定的数据链路控制器的所有功能。另外，也向 CAN 总线提供特殊化的寄存器或逻辑。具体地说，为接收错误计数器、发送错误计数器、位结构寄存器等各种测试模式。而且，还具有保存 CAN 数据链路控制器的发送 / 接收的功能。

19.2.3　引脚构成

RCAN-TL1 的引脚构成如表 19.1 所示。

表 19.1　引脚构成

名称	引脚名称	输入 / 输出	功能
发送数据引脚	CTxn	输出	CAN 总线发送用引脚。
接收数据引脚	CRxn	输入	CAN 总线接收用引脚。

【注】 n=0、 1

19.2.4　存储器映像

RCAN-TL1 的存储器映像如图 19.2 所示。

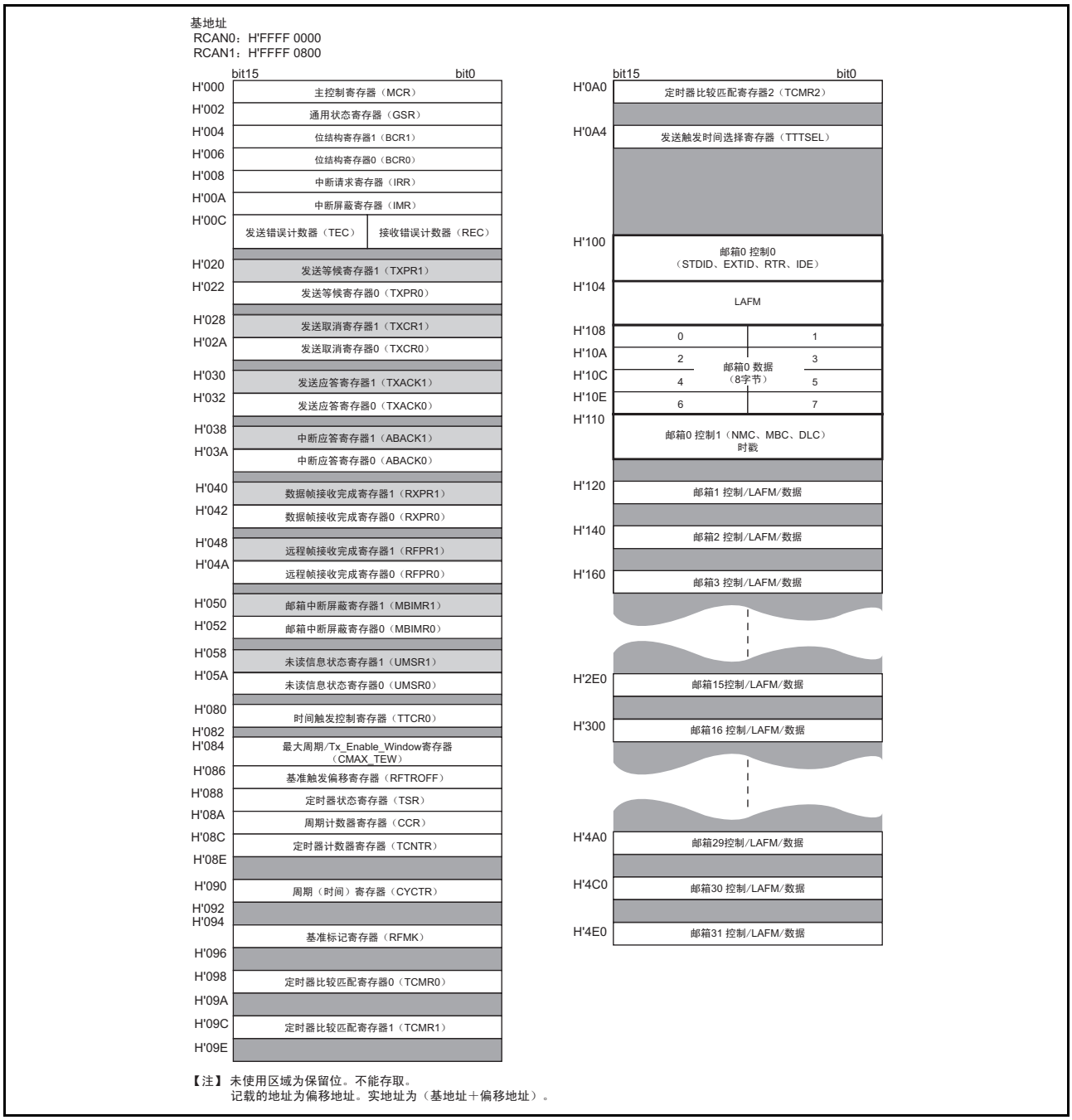


图 19.2　RCAN-TL1 的存储器映像 （每 1 个通道）

19.3　邮箱

19.3.1　　邮箱的构成

邮箱作为用于发送/接收CAN帧的信息缓冲器来运行。各邮箱由信息控制、本地接收过滤器屏蔽（LAFM）、信息数据的 3 个保存字段构成。而且还有具备时戳、时间触发结构及时间触发控制字段的邮箱。

各邮箱的信息控制、 LAFM、信息数据、时戳、发送触发时间及时间触发控制的地址映射如表 19.2 所示。

表 19.2　各邮箱的地址映射 （每 1 个通道）

邮箱	地址						
	控制 0	LAFM	数据	控制 1	时戳	触发时间	TT 控制
	4 字节	4 字节	8 字节	2 字节	2 字节	2 字节	2 字节
0 （仅接收）	100-103	104-107	108-10F	110-111	112-113	—	—
1	120-123	124-127	128-12F	130-131	132-133	—	—
2	140-143	144-147	148-14F	150-151	152-153	—	—
3	160-163	164-167	168-16F	170-171	172-173	—	—
4	180-183	184-187	188-18F	190-191	192-193	—	—
5	1A0-1A3	1A4-1A7	1A8-1AF	1B0-1B1	1B2-1B3	—	—
6	1C0-1C3	1C4-1C7	1C8-1CF	1D0-1D1	1D2-1D3	—	—
7	1E0-1E3	1E4-1E7	1E8-1EF	1F0-1F1	1F2-1F3	—	—
8	200-203	204-207	208-20F	210-211	212-213	—	—
9	220-223	224-227	228-22F	230-231	232-233	—	—
10	240-243	244-247	248-24F	250-251	252-253	—	—
11	260-263	264-267	268-26F	270-271	272-273	—	—
12	280-283	284-287	288-28F	290-291	292-293	—	—
13	2A0-2A3	2A4-2A7	2A8-2AF	2B0-2B1	2B2-2B3	—	—
14	2C0-2C3	2C4-2C7	2C8-2CF	2D0-2D1	2D2-2D3	—	—
15	2E0-2E3	2E4-2E7	2E8-2EF	2F0-2F1	2F2-2F3	—	—
16	300-303	304-307	308-30F	310-311	—	—	—
17	320-323	324-327	328-32F	330-331	—	—	—
18	340-343	344-347	348-34F	350-351	—	—	—
19	360-363	364-367	368-36F	370-371	—	—	—
20	380-383	384-387	388-38F	390-391	—	—	—
21	3A0-3A3	3A4-3A7	3A8-3AF	3B0-3B1	—	—	—
22	3C0-3C3	3C4-3C7	3C8-3CF	3D0-3D1	—	—	—
23	3E0-3E3	3E4-3E7	3E8-3EF	3F0-3F1	—	—	—
24	400-403	404-407	408-40F	410-411	—	414-415	416-417
25	420-423	424-427	428-42F	430-431	—	434-435	436-437
26	440-443	444-447	448-44F	450-451	—	454-455	456-457
27	460-463	464-467	468-46F	470-471	—	474-475	476-477

邮箱	地址						
	控制 0	LAFM	数据	控制 1	时戳	触发时间	TT 控制
	4 字节	4 字节	8 字节	2 字节	2 字节	2 字节	2 字节
28	480-483	484-487	488-48F	490-491	—	494-495	496-497
29	4A0-4A3	4A4-4A7	4A8-4AF	4B0-4B1	—	4B4-4B5	4B6-4B7
30	4C0-4C3	4C4-4C7	4C8-4CF	4D0-4D1	4D2-4D3 （本地时间）	4D4-4D5	—
31	4E0-4E3	4E4-4E7	4E8-4EF	4F0-4F1	4F2-4F3 （本地时间）	—	—

邮箱 0 为接收 （信息）专用。其他所有邮箱通过信息控制的 MBC （邮箱构成）位的设定，均可进行发送 / 接收。邮箱构成的详细内容如图 19.3 ～图 19.5 所示。

表 19.3　邮箱的作用

	事件触发		时间触发		备注	
	发送	接收	发送	接收	时戳寄存器	发送定时器时间寄存器
MB31	可设定	可设定	—	接收时间基准	有	—
MB30	可设定	可设定	主时间控模式时发送时间基准	从时间模式时接收	有	有
MB29 ～ MB24	可设定	可设定	OK	OK	—	有
MB23 ～ MB16	可设定	可设定	—（ET）	OK	—	—
MB15 ～ MB1	可设定	可设定	—（ET）	OK	有	—
MB0	—	可设定	—	OK	有	—

【注】 ET：在时间触发模式下的合并仲裁窗期间内可进行发送

• MB0（带时戳的接收邮箱）

地址	数据总线																存取长度	字段名
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100+N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		16/32	控制0
H'102+N*32	EXTID[15:0]																16	
H'104+N*32	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM[17:16]		16/32	LAFM
H'106+N*32	EXTID_LAFM[15:0]																16	
H'108+N*32	MSG_DATA_0 （初始的Rx/Tx 字节）								MSG_DATA_1								8/16/32	数据
H'10A+N*32	MSG_DATA_2								MSG_DATA_3								8/16	
H'10C+N*32	MSG_DATA_4								MSG_DATA_5								8/16/32	
H'10E+N*32	MSG_DATA_6								MSG_DATA_7								8/16	
H'110+N*32	0	0	NMC	0	0	MBC[2:0]*			0	0	0	0	DLC[3:0]				8/16	控制1
H'112+N*32	TimeStamp[15:0] (CYCTR[15:0] or CCR[5:0]/CYCTR[15:6] at SOF)																16	时戳

• MB15～1（带时戳的邮箱）

地址	数据总线																存取长度	字段名
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100+N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		16/32	控制0
H'102+N*32	EXTID[15:0]																16	
H'104+N*32	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM[17:16]		16/32	LAFM
H'106+N*32	EXTID_LAFM[15:0]																16	
H'108+N*32	MSG_DATA_0 （初始的Rx/Tx 字节）								MSG_DATA_1								8/16/32	数据
H'10A+N*32	MSG_DATA_2								MSG_DATA_3								8/16	
H'10C+N*32	MSG_DATA_4								MSG_DATA_5								8/16/32	
H'10E+N*32	MSG_DATA_6								MSG_DATA_7								8/16	
H'110+N*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				8/16	控制1
H'112+N*32	TimeStamp[15:0] (CYCTR[15:0] or CCR[5:0]/CYCTR[15:6] at SOF)																16	时戳

【注】* MBC[1]＝1固定

1. 灰色表示的位为保留位。写入的值为0。读出的值不一定为0，并不能保证。
2. 在邮箱0中不支持ATX和DART。另外，邮箱0的MBC的设定值是有限的。
3. 由于MCR15位的初始值为1，信息控制及LAFM的STDID、RTR、IDE、EXTID的顺序与HCAN2的不同。

图 19.3　邮箱 （0 ～ 15）的构成

• MB23～16（邮箱：无时戳）

地址	数据总线																存取长度	字段名
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100+N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		16/32	控制0
H'102+N*32	EXTID[15:0]																16	
H'104+N*32	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM[17:16]		16/32	LAFM
H'106+N*32	EXTID_LAFM[15:0]																16	
H'108+N*32	MSG_DATA_0 （初始的Rx/Tx 字节）								MSG_DATA_1								8/16/32	数据
H'10A+N*32	MSG_DATA_2								MSG_DATA_3								8/16	
H'10C+N*32	MSG_DATA_4								MSG_DATA_5								8/16/32	
H'10E+N*32	MSG_DATA_6								MSG_DATA_7								8/16	
H'110+N*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				8/16	控制1

• MB29～24（在时间触发模式时发送时间触发）

地址	数据总线																存取长度	字段名
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100+N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		16/32	控制0
H'102+N*32	EXTID[15:0]																16	
H'104+N*32	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM[17:16]		16/32	LAFM
H'106+N*32	EXTID_LAFM[15:0]																16	
H'108+N*32	MSG_DATA_0 （初始的Rx/Tx 字节）								MSG_DATA_1								8/16/32	数据
H'10A+N*32	MSG_DATA_2								MSG_DATA_3								8/16	
H'10C+N*32	MSG_DATA_4								MSG_DATA_5								8/16/32	
H'10E+N*32	MSG_DATA_6								MSG_DATA_7								8/16	
H'110+N*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				8/16	控制1
H'112+N*32	保留																—	—
H'114+N*32	发送触发时间（TTT）																16	触发时间
H'116+N*32	TTW[1:0]		偏移						0	0	0	0	0	Rep_Factor			16	TT控制

【注】 1. 灰色表示的位为保留位。写入的值为0。读出的值不一定为0，并不能保证。

2. 由于MCR15位的初始值为1，信息控制及LAFM的STDID、RTR、IDE、EXTID的顺序与HCAN2不同。

图 19.4　邮箱 （16 ～ 29）的构成

• MB30（在时间触发模式时发送时间参考）

地址	数据总线																存取长度	字段名
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100+N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		16/32	控制0
H'102+N*32	EXTID[15:0]																16	
H'104+N*32	IDE LAFM	0	0	STDID_LAFM[10:0]											EXTID LAFM[17:16]		16/32	LAFM
H'106+N*32	EXTID_LAFM[15:0]																16	
H'108+N*32	MSG_DATA_0（初始的Rx/Tx 字节）								MSG_DATA_1								8/16/32	数据
H'10A+N*32	MSG_DATA_2								MSG_DATA_3								8/16	
H'10C+N*32	MSG_DATA_4								MSG_DATA_5								8/16/32	
H'10E+N*32	MSG_DATA_6								MSG_DATA_7								8/16	
H'110+N*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				8/16	控制1
H'112+N*32	TimeStamp[15:0]（在SOF的TCNTR值）																16	时戳
H'114+N*32	作为时间参考的发送触发时间（TTT）																16	触发时间

• MB31（在时间触发模式时接收时间参考）

地址	数据总线																存取长度	字段名
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100+N*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		16/32	控制0
H'102+N*32	EXTID[15:0]																16	
H'104+N*32	IDE LAFM	0	0	STDID_LAFM[10:0]											EXTID LAFM[17:16]		16/32	LAFM
H'106+N*32	EXTID_LAFM[15:0]																16	
H'108+N*32	MSG_DATA_0（初始的Rx/Tx 字节）								MSG_DATA_1								8/16/32	数据
H'10A+N*32	MSG_DATA_2								MSG_DATA_3								8/16	
H'10C+N*32	MSG_DATA_4								MSG_DATA_5								8/16/32	
H'10E+N*32	MSG_DATA_6								MSG_DATA_7								8/16	
H'110+N*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				8/16	控制1
H'112+N*32	TimeStamp[15:0]（在SOF的TCNTR值）																16	时戳

【注】1.　灰色表示的位为保留位。写入的值为0。读出的值不一定为0，并不能保证。
2.　由于MCR15位的初始值为1，因此信息控制及LAFM的STDID、RTR、IDE、EXTID的顺序与HCAN2不同。

图 19.5　邮箱 （30、 31）的构成

19.3.2　信息控制字段

寄存器名称	地址	位名称	位名	说明
MB[x]. CONTROL0H	H'100+N*32	15	IDE	ID 扩展 将 CAN 数据帧与远程帧分为标准格式或扩展格式。 0：标准格式 1：扩展格式
		14	RTR	远程发送请求 区分数据帧与远程帧。根据数据帧或远程帧，接收 CAN 帧重写此位。 【重点】通过 MBC=B'001 置位数据帧自动发送（ATX）位时，RTR 则不可置位。接收远程帧时，通过对应的 RFPR 位或 IRR2（远程帧接收中断）位通知 CPU，但 RCAN-TL1 需要将当前的信息作为数据帧进行发送，所以 RTR 位无变化。 【重点】通过 MBC=B'001 设定 ATX=1，接收远程帧的邮箱自动进行发送设定。此时为了发送数据帧，必须将 RTR 设定为 0。 0：数据帧 1：远程帧
		13	—	保留位 初始值不定。写入值常设定为 0。
		12 ～ 2	STDID [10:0]	标准 ID 设定数据帧与远程帧的 ID （标准 ID）的位。
		1、 0	EXTID [17:16]	扩展 ID 设定数据帧与远程帧的 ID （扩展 ID）的位。
MB[x]. CONTROL0L	H'102+N*32	15 ～ 0	EXTID [15:0]	

- 邮箱0

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

【注】 MBC[1] 的值总是 1。

- 邮箱1～31

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

寄存器名称	地址	位名称	位名	说明
MB[x]. CONTROL1	H'110+N*32	15、14	—	保留位 读写的值总是 0。
		13	NMC	新信息控制 此位置 0 时， RXPR 或 RFPR 已置位的邮箱不保存新信息，而保持之前的信息，并置位 UMSR 对应的位。 此位置 1 时， RXPR 或 RFPR 已置位的邮箱重写新信息，并置位 UMSR 对应的位。 【重点】如果可通过数据帧重写远程帧或与其相反时，可将 RXPR 与 RFPR 标志 （与 USMR 同时进行）设置在同一邮箱中。此时，存在于邮箱控制字段中的 RTR 位也进行重写。 【重点】使用时间触发模式时，必须将邮箱 31 的 NMC 置 1，即使不清除 RXPR[31]，也请设定为可接收所有基准信息。 0：溢出模式 1：重写模式
	H'110+N*32	12	ATX	自动发送数据帧 此位置 1，通过邮箱接收远程帧时，保存 DLC， TXPR 自动进行置位。使用由相同邮箱重写的 DLC，自动发送数据帧。 已设定为自动发送的邮箱根据信息发送优先级位 （MCR2）的设定，按照 ID 优先级或邮箱优先级进行调度。使用此功能时，请将 MBC[2:0] 设定为 B'001。使用此功能进行发送时，所使用的数据长度码 （DLC）使用已接收的内容。应用此内容时，远程帧的 DLC 必须对应于所要求的数据帧的 DLC。 【重点】 1. 使用 ATX， MBC 为 B'001 时，与应答信息时相同，远程帧的 ID 必须与数据帧的 ID 完全相同，所以不可使用 IDE 位的过滤器。 2. 使用此功能时，与接收远程帧无关，RTR 位不置位。接收远程帧时，置位结束后，虽通过 RFPR 通知 CPU，但 RCAN-TL1 需要将当前信息作为数据帧进行发送，所以不变更 RTR 位。 3. 有时不开始自动发送远程帧。请注意溢出的状态（NMC=0 时的 UMSR 置位）。 4. 通过远程帧，已设定为 ATX=1 的邮箱为溢出状态时，有时会接受旧信息的自动发送请求。 0：数据帧的自动发送无效 1：数据帧的自动发送有效 【注】此位仅存在于邮箱 1 ～ 31 中。在邮箱 0 中为保留位。
		11	DART	自动重新发送无效 如果此位置 1，发生 CAN 总线错误事件或通过 CAN 总线仲裁失败时，信息的自动重新发送无效。实际上，使用此功能时，在发送开始，自动置位所对应的 TXCR 位。此位置 0 时， RCAN-TL1 通过 TXCR 取消发送请求或在发送正常结束之前请求发送。 0：再发送有效 1：再发送无效 【注】此位仅存在于邮箱 1 ～ 31 中。在邮箱 0 中为保留位。

寄存器名称	地址	位	位名称	说明
MB[x]. CONTROL1	H'110+N*32	10 ～ 8	MBC[2:0]	邮箱结构 这些位如表 19.4 所示设定各邮箱的功能。MBC 为 B'111 时，邮箱无效。即：与 TXPR 或其他的设定无关，不进行信息的发送 / 接收。 禁止将 MBC 设定为 B'100、 B'101、 B'110。如果将 MBC 的值设定为 B'000 或其他值，则可使用 LAFM 字段。 设定 MBC 为接收时，请勿置位 TXPR。无硬件保护功能， TXPR 仍旧进行置位。邮箱 0 的 MBC[1] 为接收专用，因此通过硬件固定为 1。
		7 ～ 4	—	保留位 读写的值总是 0。
		3 ～ 0	DLC[3:0]	数据长度码 通过数据帧将所发送的数据字节数 （0 ～ 8）进行编码。 0000：数据长度 0 字节 0001：数据长度 1 字节 0010：数据长度 2 字节 0011：数据长度 3 字节 0100：数据长度 4 字节 0101：数据长度 5 字节 0110：数据长度 6 字节 0111：数据长度 7 字节 1xxx：数据长度 8 字节 【注】 x：Don't care

表 19.4　邮箱功能的设定

MBC2	MBC1	MBC0	发送数据帧	发送远程帧	接收数据帧	接收远程帧	说明
0	0	0	可	可	不可	不可	• 不可使用邮箱 0 • 可使用时间触发发送
0	0	1	可	可	不可	可	• 可通过 ATX 使用 * • 不可使用邮箱 0 • 可使用 LAFM
0	1	0	不可	不可	可	可	• 可使用邮箱 0 • 可使用 LAFM
0	1	1	不可	不可	可	不可	• 可使用邮箱 0 • 可使用 LAFM
1	0	0	禁止设定				
1	0	1	禁止设定				
1	1	0	禁止设定				
1	1	1	邮箱无效 （初始值）				

【注】 * 为了支持自动再发送， MBC 为 B'001，而且 ATX 为 1 时，请将 RTR 设定为 0。将 ATX 设定为 1 使用时，请勿使用 IDE 过滤器。

19.3.3　局部接收过滤器屏蔽 （LAFM）

MBC 为 B'001、 B'010、 B'011 的值时，此字段作为接收用 LAFM 来使用。 LAFM 允许邮箱接受多个接收 ID。 LAFM 如图 19.6 所示，由 2 个 16 位可读 / 写区域构成。

寄存器名	地址	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	存取长度	字段名
MB[0].LAFMH	H'104+N*32	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM[17:16]		16/32	LAFM
MB[0].LAFML	H'106+N*32	EXTID_LAFM[15:0]																16	

图 19.6　局部接收过滤器屏蔽 （LAFM）

将 1 个位置位至 LAFM 中，通过与 RCAN-TL1 相同的 CAN-ID 搜索邮箱时，忽略已接收的 CAN 的 ID 所对应的位。清除位时，已接收的 CAN 的 ID 所对应的位必须与所保存的邮箱中设定的 STDID/IDE/EXTID 保持匹配。 LAFM 的结构与邮箱的信息控制相同。不使用此功能时，必须为 0。

【注】 1. RCAN-TL1 从邮箱 31 至邮箱 0，进行相符 ID 的搜索。 RCAN-TL1 在搜索出相符 ID 时，不论 NMC 或 RXPR/RFPR 标志，立即结束搜索此信息。这就表示即使使用 LAFM，接收的信息也仅限保存在 1 个邮箱中。

2. 接收一条信息并发现相符的邮箱时，全部信息就保存至邮箱中。使用 LAFM 时， STDID、 RTR、 IDE、 EXTID 更新为所接收信息的 STDID、 RTR、 IDE、 EXTID，所以有时与接收前设定的不同。

寄存器名称	地址	位	位名称	说明
MB[x].LAFMH	H'104+N*32	15	IDE_LAFM	IDE 位的过滤器屏蔽位 0：对应的 IDE 位有效 1：对应的 IDE 位无效
		14、 13	—	保留位 初始值不定。写入值常设定为 0。
		12 ～ 2	STDID_LAFM [10:0]	STDID[10:0] 的过滤器屏蔽位 0：对应的 STDID 位有效 1：对应的 STDID 位无效
		1、 0	EXTID_LAFM [17:16]	EXTID[17:0] 的过滤器屏蔽位 0：对应的 EXTID 位有效 1：对应的 EXTID 位无效
MB[x].LAFML	H'106+N*32	15 ～ 0	EXTID_LAFM [15:0]	

19.3.4　信息数据字段

保存发送 / 接收的 CAN 信息。MSG_DATA_0 对应于进行发送 / 接收的开始数据字节。CAN 总线中的位的排列顺序为从 bit7 至 bit0。

- 时间触发发送注意事项

邮箱 30 通过 CMAX ≠ B'111、 MBC[30]=B'000，将 TXPR[30] 设定为 1 时，可设定为时间基准发送用。此时， DLC 必须设定为大于 0 的值，并且将 RTR 设定为 0 （TTCAN Level 1 中规定）。邮箱 30 开始发送时，将周期计数器的值 （CCR）插入数据字段起始字节 MSG_DATA_0[5:0] 中后，发送信息。

MSG_DATA_0[7:6] 发送保存在邮箱中的值。

需要发送 Next_is_Gap 时，用户可通过将 MSG_DATA_0[7] 设定为 1 来进行发送。

此时，将 CCR 的值插入所发送的帧中，但请注意：未保存在邮箱 30 的信息数据字段中。

通过 CMAX ≠ B'111、 MBC[31]=B'011 设定邮箱 31 时，设定为时间基准接收用。接收有效的基准信息时（ID 匹配、 DLC ＞ 0）， RCAN-TL1 通过重写 RFMK 与周期计数器 （CCR），将 Cycle_Time 与 Basic_Cycle 进行同步化。

MB30、31

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'108+N*32	Next_is_Gap/Cycle_Counter（初始的Rx/Tx字节）								MSG_DATA_1								8/16/32位	数据
H'10A+N*32	MSG_DATA_2								MSG_DATA_3								8/16位	
H'10C+N*32	MSG_DATA_4								MSG_DATA_5								8/16/32位	
H'10E+N*32	MSG_DATA_6								MSG_DATA_7								8/16位	

图 19.7　信息数据字段

19.3.5　时戳

保存发送 / 接收信息中所记录的时戳。时戳用于监控是否按计划发送 / 接收信息。

- 时戳

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8	TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
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

（1） 接收信息

邮箱 0 ～ 15 的接收信息时戳，通过时间触发控制寄存器 0 （TTCR0） bit14 的设定，在接收信息的 SOF 捕捉 CYCTR[15:0]（周期时间寄存器）的值或 CCR[5:0]（周期计数器）+CYCTR[15:6]（周期时间寄存器）的值，并保存至时戳。

邮箱 30、 31 的接收信息时戳，在接收信息的 SOF 捕捉定时器计数器寄存器 （TCNTR）的值，并保存至时戳。

（2） 发送信息

邮箱 1 ～ 15 的发送信息时戳，通过时间触发控制寄存器 0 （TTCR0）的 bit14 的设定，在接收信息的 SOF 捕捉 CYCTR[15:0]（周期时间寄存器）的值或 CCR[5:0]（周期计数器） +CYCTR[15:6]（周期时间寄存器）的值，并保存至时戳。

邮箱 30、 31 的发送信息时戳，在接收信息的 SOF 捕捉定时器计数器寄存器 （TCNTR）的值，并保存至时戳。

【重点】 时戳保存在暂存器中。正常进行发送或接收时，复制其值至邮箱字段。另外，在通过溢出置位 UMSR[N] 的同时， CPU 清除 RXPR[N]/RFPR[N] 时，有时仅时戳进行更新。因此，可读取清除 RXPR[N]/RFPR[N] 之前的正确的时戳值。

19.3.6　发送触发时间 （TTT）与时间触发控制

邮箱 24 ～ 29，通过时间触发模式 （CMAX ≠ B'111）将 MBC 设定为 B'000 时，发送触发时间作为决定时间窗口间的边界的 Time_Mark 来使用。

发送触发时间 （TTT）与时间触发控制 （TT 控制）由下述 2 个 16 位读 / 写寄存器构成。

邮箱 30 中无时间触发控制 （TT 控制），作为 Time_Ref 来使用。

邮箱 24 ～ 30 不用于时间触发模式发送时，可作为接收邮箱来使用。但使用时间触发模式时，不会成为事件触发发送的目标。

- 发送触发时间 （TTT）

发送触发时间 （TTT）指定周期时间内开始信息发送的时间。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TTT15	TTT14	TTT13	TTT12	TTT11	TTT10	TTT9	TTT8	TTT7	TTT6	TTT5	TTT4	TTT3	TTT2	TTT1	TTT0
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

- 时间触发控制 （TT控制）

时间触发控制 （TT 控制）指定窗口的属性、开始发送的系统矩阵中的周期计数 （Basic Cycle）及定周期发送的频率。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TTW[1:0]		Offset[5:0]						0	0	0	0	0	rep_factor[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/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

支持时间触发模式的所有邮箱之间的异同点如图 19.8 所示。

MB29～24

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'114＋N*32	发送触发时间（周期时间）																16位	触发时间
H'116＋N*32	TTW[1:0]		Offset[5:0]						0	0	0	0	0	rep_factor[2:0]			16位	TT控制

MB30

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'114＋N*32	发送触发时间（周期时间）																16位	触发时间

图 19.8　发送触发控制字段

- TTW[1:0] （时间触发窗口）

表示时间窗口的属性。通过 TTW=B'10 设定，已开始的合并仲裁窗口 （Merged Arbitrating Window）必须通过 TTW=B'11 的设定来结束。从合并仲裁窗口的起始到结束之间可使用多个 TTW=B'10 设定的信息。

TTW[1]	TTW[0]	说明
0	0	异或非窗口　　：exclusive window （初始值）
0	1	仲裁窗口　　：arbitrating window
1	0	开始合并仲裁窗口　　：Start of Merged arbitrating window
1	1	结束合并仲裁窗口　　：End of Merged arbitrating window

TTT 寄存器的值与 CYCTR 的值匹配， Offset 的值与 CCR 的值匹配时，从所对应的邮箱尝试发送。为了使此功能有效，请设定为 CMAX ≠ B'111、运行 （TTCR0 的 bit15 为 1）定时器 （TCNTR）、将所对应的邮箱设定为 MBC=B'000、将所对应的 TXPR 位设定为 1。通过软件将 TXPR 的位置位后，继续进行定周期发送，所以 RCAN-TL1 不清除 TXPR 的位 （邮箱 24 ～ 30）。

停止定周期发送时，请通过 TXCR 清除 TXPR。此时，发送完成后未立即清除 TXACK 时，有可能将相同邮箱的 TXACK 与 ABACK 双方置位 （参考**图 19.9**）。

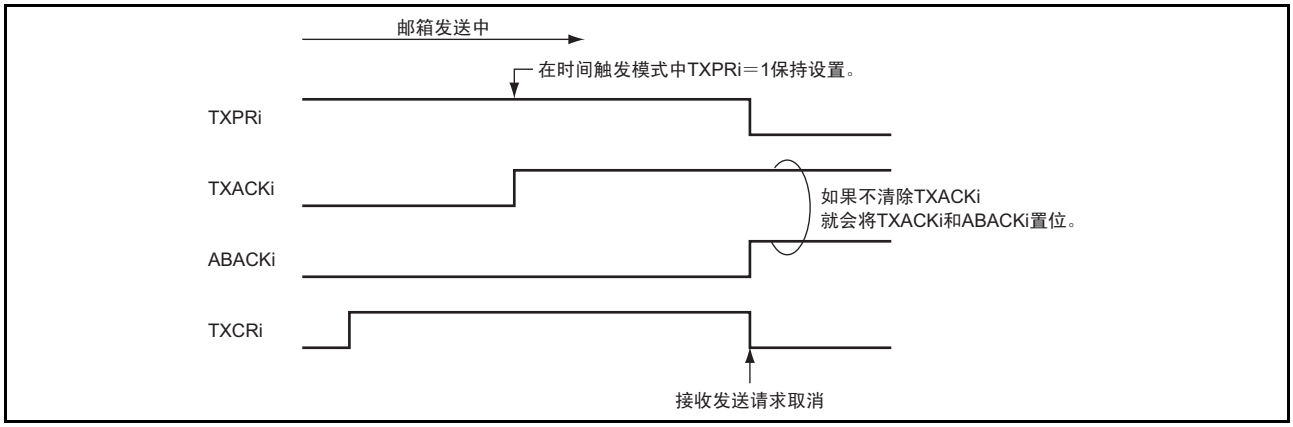


图 19.9　时间触发发送时的 TXACK 与 ABACK

邮箱 30 固定为 TTW=B'01、 Offset=B'000000、 rep_factor=B'000。 rep_factor 与 Offset 的组合如下表所示。

rep_factor	说明
B'000	每 1 个周期计数 （初始值）
B'001	每 2 个周期计数
B'010	每 4 个周期计数
B'011	每 8 个周期计数
B'100	每 16 个周期计数
B'101	每 32 个周期计数
B'110	每 64 个周期计数 （每个系统矩阵进行 1 次）
B'111	保留

Offset 字段决定时间触发邮箱可开始信息发送的初始周期计数器值。

Offset	说明
B'000000	偏移 （Offset） = 初始的周期计数 （初始值）
B'000001	偏移 （Offset） = 第 2 个周期计数
B'000010	偏移 （Offset） = 第 3 个周期计数
B'000011	偏移 （Offset） = 第 4 个周期计数
B'000100	偏移 （Offset） = 第 5 个周期计数
• • •	• • •
B'111110	偏移 （Offset） = 第 63 个周期计数
B'111111	偏移 （Offset） = 第 64 个周期计数

必须满足以下关系：

Cycle_Count_Maximum+1 ＞ =Repeat_Factor ＞ Offset
Cycle_Count_Maximum=2^{CMAX}-1
Repeat_Factor=2^{rep_factor}

CMAX、 Repeat_Factor 和 Offset 为寄存器的值。

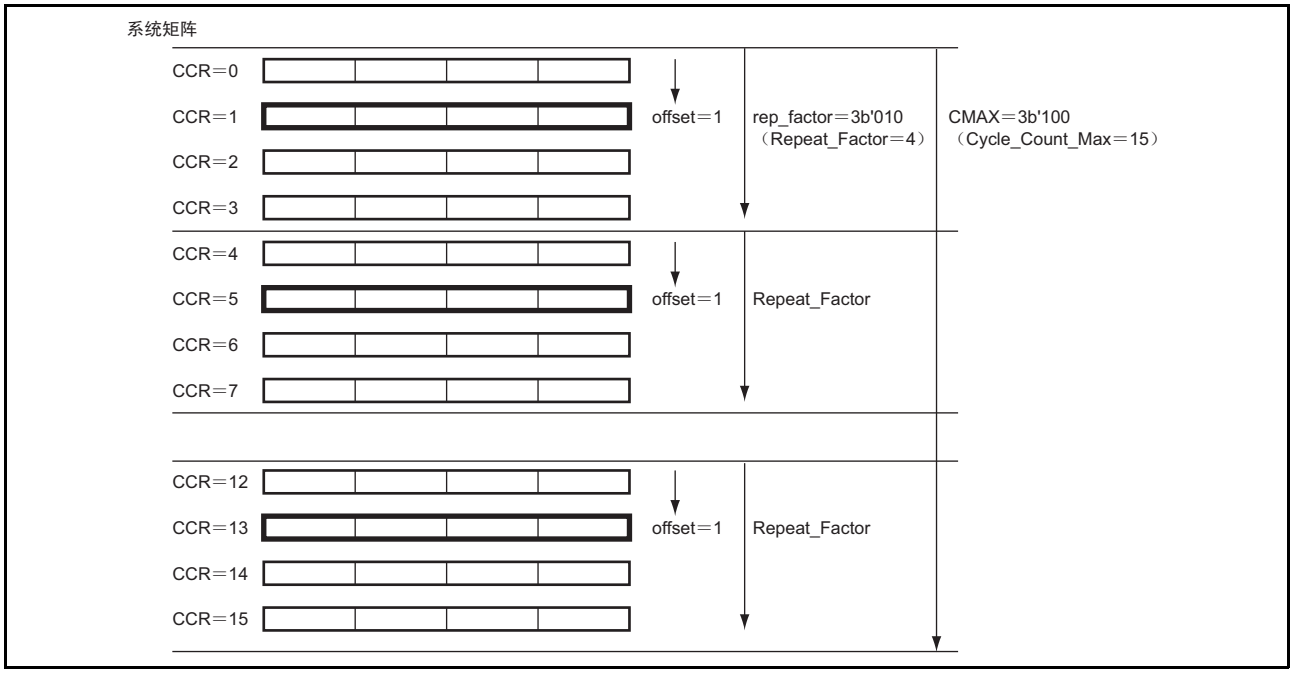


图 19.10　系统矩阵

请将发送触发时间设定为升序。发送触发时间之间的差必须最少确保为 TEW 宽度。

19.4　RCAN-TL1 的控制寄存器

现就 RCAN-TL1 的控制寄存器进行说明。 RCAN-TL1 的控制寄存器仅可通过字长度 （16 位）进行存取。 RCAN-TL1 的控制寄存器如表 19.5 所示。

表 19.5　RCAN-TL1 的控制寄存器的构成

寄存器名称	简称	地址	存取长度 （位）
主控制寄存器	MCR	H'000	16
通用状态寄存器	GSR	H'002	16
位结构寄存器 1	BCR1	H'004	16
位结构寄存器 0	BCR0	H'006	16
中断请求寄存器	IRR	H'008	16
中断屏蔽寄存器	IMR	H'00A	16
发送错误计数器 / 接收错误计数器	TEC/REC	H'00C	16

19.4.1　主控制寄存器 （MCR）

MCR 为 16 位可读 / 写寄存器，控制 RCAN-TL1。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MCR15	MCR14	—	—	—	TST[2:0]			MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0
初始值：	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W：	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W

位	位名称	初始值	R/W	说明
15	MCR15	1	R/W	变更 ID 排列 此位可设定与 HCAN2 具有兼容性的信息控制及 LAFM 的 STDID、 RTR、 IDE、 EXTID 的顺序。而且，此位仅可在复位模式下进行变更。关于变更 ID 排列的顺序请参考图 19.11。 0：RCAN-TL1 与 HCAN2 为相同的顺序 1：RCAN-TL1 与 HCAN2 为不同的顺序
14	MCR14	0	R/W	自动总线断开停机 同时设定此位与 MCR6 时，如果 RCAN-TL1 转移至总线断开状态， MCR1 则立即自动进行置位。此位仅可在复位模式下进行变更。 0：通过正常的返回顺序 （128×11 隐性位）保持 RCAN-TL1 总线断开状态 1：MCR6 置位时，RCAN-TL1 转移至总线断开状态后立即转移至停机模式
13 ～ 11	—	均为 0	R	保留位 读写的值总是 0。

位	位名称	初始值	R/W	说明
10 ～ 8	TST[2:0]	000	R/W	测试模式 此位设定测试模式有效 / 无效。请注意：启动测试模式之前，需要将 RCAN-TL1 设定为停机模式或复位模式。这是为了避免影响向测试模式转移过程中的发送 / 接收。详细内容请参考“19.7.2 测试模式的设定”。测试模式为仅用于诊断以及测试的模式， RCAN-TL1 正常运行时不可使用。 000：普通模式 001：listen-only 模式 （接收专用模式） 010：自测试模式 1 （外部） 011：自测试模式 2 （内部） 100：写错误计数器 101：错误被动模式 110：禁止设定 111：禁止设定
7	MCR7	0	R/W	自动唤醒模式 此位设定 CAN 睡眠模式的自动唤醒模式有效 / 无效。设定此位时， RCAN-TL1 检测 CAN 总线激活 （显性位）后自动解除 CAN 睡眠模式 （MCR5）。清除此位时，RCAN-TL1 不自动解除 CAN 睡眠模式。RCAN-TL1 不可保存唤醒 RCAN-TL1 的信息。 0：通过 CAN 总线激活产生的自动唤醒模式无效 1：通过 CAN 总线激活产生的自动唤醒模式有效 【注】CAN 睡眠模式下不可变更 MCR7 位。
6	MCR6	0	R/W	总线断开时停机 此位设定在总线断开时如果置位 MCR1 则立即转移至停机模式有效或无效。此位仅可在复位或停机模式下进行变更。请注意：总线断开状态下，转移至停机模式时， CAN 控制器也立即返回错误主动状态模式。 0：在总线断开时，不转移至停机模式，等待返回顺序结束 1：在总线断开时，将通过 MCR1 设定产生的停机模式转移设置为有效

位	位名称	初始值	R/W	说明
5	MCR5	0	R/W	CAN 睡眠模式 此位设定向 CAN 睡眠模式的转移有效 / 无效。 RCAN-TL1 在停机模式下将此位进行置位时，向 CAN 睡眠模式的转移有效。转移至停机模式后允许进行此位的设定。 2 个错误计数器 （REC、 TEC）在 CAN 睡眠模式间不变化。解除 CAN 睡眠模式的方法有 2 种。 • 向 MCR5 位写入 0 • MCR7 有效时，检测 CAN 总线中的显性位 如果自动睡眠解除模式无效，在 CAN 睡眠模式结束之前， RCAN-TL1 忽略所有 CAN 总线的运行。解除 CAN 睡眠模式时， RCAN-TL1 在转移至 CAN 总线运行之前，通过检查 11 个隐性位来与 CAN 总线取得同步。这意味着在使用第 2 种方法时，RCAN-TL1 无法接收最初的信息。同样，CAN 收发器也在结束待机模式时无法接收最初的信息，所以需要采用此方法设计软件。 CAN 睡眠模式下仅可存取 MCR、GSR、IRR、IMR 寄存器。详细内容请参考 “19.7.1 （3） CAN 睡眠模式”。 0：解除 CAN 睡眠模式 1：转移至 CAN 睡眠模式有效 【注】最初设定为停机模式之后，可将 RCAN-TL1 转移至 CAN 睡眠模式。因此，在转移至 CAN 睡眠模式之前， CPU 可清除等待状态中的所有中断。在清除所有中断的同时，必须将RCAN-TL1从停机模式转移至CAN睡眠模式。（具体地说就是向 MCR5 写入 1、同时向 MCR1 写入 0。）
4、3	—	均为 0	R	保留位 读写的值总是 0。
2	MCR2	0	R/W	信息发送优先级 此位选择等候中的发送数据的发送顺序。此位置 1 时，按发送等候寄存器（TXPR）的位顺序发送发送数据。 最优先开始发送邮箱 31，持续到邮箱 1 为止 （将邮箱设定为发送用时）。 请注意：此功能不可用于邮箱 24 ～ 30 的时间触发发送。如果清除此位，所有的发送信息 （通过运行内部仲裁）按照 ID 优先级进行发送。最优先的信息具有最小数字的仲裁字段（STDID+IDE 位 +EXTID（IDE=1 时）+RTR 位），最初进行发送。 内部仲裁包含 RTR 位与 IDE 位 （采用与存在于 2 个 CAN 节点之间的 CAN 总线中的仲裁同时开始转移相同的方法运行内部仲裁）。 此位仅可在复位或停机模式下进行变更。 0：按信息 ID 优先级进行发送 1：按邮箱号码顺序 （邮箱 31→ 邮箱 1）进行发送

位	位名称	初始值	R/W	说明
1	MCR1	0	R/W	停机请求 置位此位时， CAN 控制器在结束当前运行后转移至停机模式（在此，从 CAN 总线分离）。在清除此位之前， RCAN-TL1 依旧保持停机模式。 在停机模式下，CAN 接口与 CAN 总线运行无关，也不进行信息的保存或发送。用于向 CPU 通知停机状态的 IRR0 与 GSR4 除外，保持用户寄存器 （包含邮箱内容及 TEC/REC）的内容。 CAN 总线在空闲或间歇状态下，与 MCR6 无关， RCAN-TL1 在 1 位时间内为停机模式。 MCR6 置位时，总线断开中的停机请求也在 1 位时间内进行运行。此外，到总线断开返回顺序结束之前，不转移至停机模式。转移至停机模式时，通过 IRR0 与 GSR4 进行通知。 MCR14 与 MCR6 置位时，如果 RCAN-TL1 转移至总线断开状态，则立即自动置位此位。 停机模式下，由于 RCAN-TL1 与总线运行无关，所以除位时序设定外，可变更 RCAN-TL1 的结构。为了再次加入 CAN 总线运行，需要将此位清 0。清除后，RCAN-TL1 等待至检测出 11 个隐性位后，加入 CAN 总线。 0：清除停机模式请求 1：停机模式转移请求 【注】1. 发送停机请求后，在向停机模式转移结束之前，CPU 不可存取（通过 IRR0 与 GSR4 进行通知）TXPR 与 TXCR 及清除此位。将 MCR1 置位后，此状态在转移至停机模式之后或仅通过（以软件或硬件的）复位运行进行解除。 2. 仅在 BCR1 与 BCR0 寄存器设定为适当的波特率时，才可进行向停机模式的转移或从停机模式的转移。
0	MCR0	1	R/W	复位请求 此位控制 RCAN-TL1 模块的复位。此位从 0 变为 1 时， RCAN-TL1 控制器进入复位程序，初始化内部逻辑，为了通知复位模式，将 GSR3 与 IRR0 置位。初始化所有用户寄存器。 将此置位时，可重新构成 RCAN-TL1。为了加入 CAN 总线，此位需要写入 0 后，进行清除。清除后， RCAN-TL1 模块等待至检测出 11 个隐性位后，加入 CAN 总线。 为采样 CAN 总线中的值，需要设定适当的波特率。上电复位后，常将此位与 GSR3 置位。这表示请求复位，需要构成 RCAN-TL1。 复位请求相当于上电复位，通过软件进行控制。 0：清除复位模式请求 ［清除条件］将 RCAN-TL1 复位后写入 0 时 1：CAN 接口的复位模式转移请求

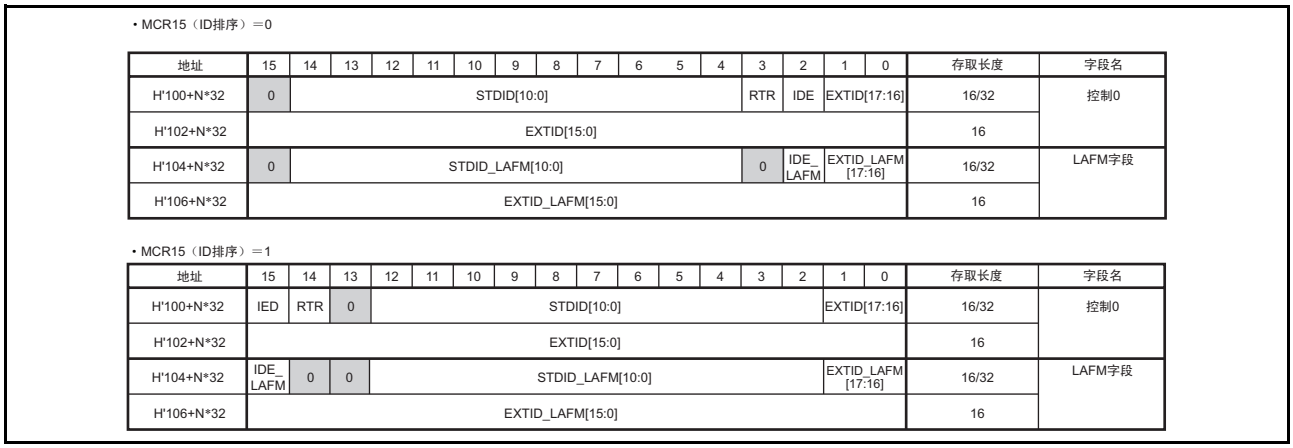


图 19.11　变更 ID 排列

19.4.2　通用状态寄存器 （GSR）

GSR 为 16 位只读寄存器，表示 RCAN-TL1 的状态。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
R/W：	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名称	初始值	R/W	说明
15 ～ 6	—	均为 0	R	保留位 读写的值总是 0。
5	GSR5	0	R	错误被动状态 表示 CAN 接口是否为错误被动。 RCAN-TL1 为错误被动状态时，此位立即置 1，再次成为错误被动状态时，进行清除。这意味着在错误被动中与总线断开中， GSR5 保持为 1。因此，需要得知正确状态时，请检查 GSR5 与 GSR0。 0：RCAN-TL1 不为错误被动或总线断开状态 ［清除条件］ RCAN-TL1 处于错误被动状态中 1：RCAN-TL1 为错误被动（但 GSR0=0 时）或总线断开（但 GSR0=1 时） ［置位条件］在 TEC ≥ 128、 REC ≥ 128 或测试模式下，选择错误被动模式时
4	GSR4	0	R	停机 / 睡眠状态 表示 CAN 控制器是否为停机 / 睡眠状态。请注意：此标志的清除时间与 IRR12 的置位时间不同。此标志为反映 CAN 控制器的状态的标志，但不是充分反映 RCAN-TL1 的状态的标志。结束 CAN 睡眠模式并清除 MCR5 时， RCAN-TL1 可进行存取。经过传送时钟 2 个位后，结束 CAN 控制器的 CAN 睡眠模式。 0：RCAN-TL1 既不是停机模式又不是 CAN 睡眠模式 1：为停机模式 （MCR1=1 时）或 CAN 睡眠模式 （MCR5=1 时） ［置位条件］ MCR1 置位， CAN 总线为间歇或空闲状态时。 MCR5 置位， RCAN-TL1 为停机模式时。 MCR14 和 MCR6 均置位， RCAN-TL1 为转移至总线断开时。

位	位名称	初始值	R/W	说明
3	GSR3	1	R	复位状态 表示 RCAN-TL1 是否为复位状态。 0：RCAN-TL1 不是复位状态 1：RCAN-TL1 为复位状态 ［置位条件］ RCAN-TL1 的软件或硬件复位后
2	GSR2	1	R	信息发送中的标志 为向 CPU 表示 RCAN-TL1 处于总线断开状态或处在发送信息中或正在发送由于发送中检测出的错误原因所引起的错误 / 超载的标志。置位 TXACK 的时序与清除 GSR2 的时序不同。在 EOF 的第 7 位设置 TXACK。无等待发送的信息时，在间歇的第 3 个位设置 GSR2。另外，也可通过仲裁失败、总线空闲、接收、复位、停机转移进行设置。 0：RCAN-TL1 为总线断开状态或发送信息中 1：［置位条件］不在总线断开状态中，也不在发送信息中
1	GSR1	0	R	发送 / 接收警告标志 表示错误警告的标志。 0：［清除条件］ TEC ＜ 96 而 REC ＜ 96 或总线断开时 1：［置位条件］ 96 ≤ TEC ＜ 256 或 96 ≤ REC ＜ 256 时 【注】 由于 REC 计数总线断开返回顺序所必需的 11 个隐性位的反复次数，所以总线断开中增加。但此位在总线断开中不进行置位。
0	GSR0	0	R	总线断开标志 表示 RCAN-TL1 为总线断开状态。 0：［清除条件］在从总线断开状态返回或硬件或软件复位后 1：［置位条件］ TEC ≥ 256 （总线断开状态）

19.4.3　位结构寄存器 0、 1 （BCR0、 BCR1）

BCR0、 BCR1 分别为 16 位可读 / 写寄存器，设定 CAN 位时序参数与 CAN 接口的波特率预分频器。时间量子定义如下：

时间量子 $=2\times BRP/f_{clk}$

BRP （波特率预分频器）为 BCR0 保存值 +1 的值。 f_{clk} 为外围总线时钟频率。

（1） BCR1

关于 TSEG1 与 TSEG2 的设定，请参考**表 19.5**。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG1[3:0]				–	TSG2[2:0]			–	–	SJW[1:0]		–	–	–	BSP
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	R	R/W	R/W	R	R	R	R/W

位	位名称	初始值	R/W	说明
15 ～ 12	TSG1[3:0]	0000	R/W	时间量子 1 由于使用正相位误差补偿 CAN 总线中的边沿，所以这些位用于设定段 TSEG1 （=PRSEG+PHSEG1）。可设定 4 ～ 16 个时间量子 0000：禁止设定 0001：禁止设定 0010：禁止设定 0011：PRSEG+PHSEG1=4 个时间量子 0100：PRSEG+PHSEG1=5 个时间量子 ：　　： 1111：PRSEG+PHSEG1=16 个时间量子
11	—	0	R	保留位 读写的值总是 0。
10 ～ 8	TSG2[2:0]	000	R/W	时间量子 2 由于使用负相位误差补偿 CAN 总线中的边沿，所以这些位用于设定段 TSEG2 （=PRSEG2）。可设定 2 ～ 8 个时间量子 000：禁止设定 001：PHSEG2=2 个时间量子 （根据条件禁止设定。请参考表19.5） 010：PHSEG2=3 个时间量子 011：PHSEG2=4 个时间量子 100：PHSEG2=5 个时间量子 101：PHSEG2=6 个时间量子 110：PHSEG2=7 个时间量子 111：PHSEG2=8 个时间量子
7、6	—	均为 0	R	保留位 读写的值总是 0。
5、4	SJW[1:0]	00	R/W	再次同步转移宽度 设定同步转移宽度。 00：同步转移宽度 =1 个时间量子 01：同步转移宽度 =2 个时间量子 10：同步转移宽度 =3 个时间量子 11：同步转移宽度 =4 个时间量子
3 ～ 1	—	均为 0	R	保留位 读写的值总是 0。
0	BSP	0	R/W	位采样点 设定采样数据的点。 0：在一处进行位采样 （时间量子 1 的最后） 1：在三处进行位采样 （PHSEG1 的最后 3 个时钟周期的上升沿）

（2） BCR0

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	BRP[7:0]							
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	位名称	初始值	R/W	说明
15 ～ 8	—	均为 0	R	保留位 读写的值总是 0。
7 ～ 0	BRP[7:0]	0000000	R/W	波特率预分频器 这些位设定对应于 1 个时间量子的外围总线时钟数。 0000000：2× 外围总线时钟 0000001：4× 外围总线时钟 0000010：6× 外围总线时钟 ： ：2×（寄存器的值 +1）× 外围总线时钟 1111111：512× 外围总线时钟

- 关于位结构寄存器



图 19.12　1 位时间的结构

SYNC_SEG：CAN 总线中的节点的同步的段
（在此段产生正常的位边沿转移）
PRSEG　：调整网络间物理性延迟的段
PHSEG1　：定相漂移 （正方向）的缓冲段 （同步或再次同步时，进行扩展）
PHSEG2　：定相漂移 （负方向）的缓冲段 （同步或再次同步时，进行缩小）
TSEG1　：TSG1+1
TSEG2　：TSG2+1

RCAN-TL1 位速率进行如下计算：

位速率 =fclk/(2×(BRP+1)×(TSEG1+TSEG2+1))

BRP 通过寄存器的值赋予。另外， TSEG1 及 TSEG2 为从上述表中计算出的值，不是实际设定值。上述位速率计算式的时间量子的 “+1” 是根据 SYNC_SEG 的 1 个时间量子得出的。

f*clk*= 外围总线时钟

BCR 设定时的限制事项

TSEG1 （Min.）＞ TSEG2 ≥ SJW （Max.） （SJW=1 ～ 4）

8 ≤ TSEG1+TSEG2+1 ≤ 25 时间量子 （不可 TSEG1+TSEG2+1=7）

TSEG2 ≥ 2

如果位结构寄存器 TSEG1 及 TSEG2 的值在表 19.6 所示的设定值的范围内，则满足上述限制事项。表 19.6 的值为 SJW 可设定的值。以“不可”表示的地方不可进行 TSEG1 与 TSEG2 的组合。

表 19.6　TSG 与 TSEG 的设定

		001	010	011	100	101	110	111	TSG2
		2	3	4	5	6	7	8	TSEG2
TSG1	TSEG1								
0011	4	不可	1-3	不可	不可	不可	不可	不可	
0100	5	1-2	1-3	1-4	不可	不可	不可	不可	
0101	6	1-2	1-3	1-4	1-4	不可	不可	不可	
0110	7	1-2	1-3	1-4	1-4	1-4	不可	不可	
0111	8	1-2	1-3	1-4	1-4	1-4	1-4	不可	
1000	9	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1001	10	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1010	11	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1011	12	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1100	13	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1101	14	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1110	15	1-2	1-3	1-4	1-4	1-4	1-4	1-4	
1111	16	1-2	1-3	1-4	1-4	1-4	1-4	1-4	

例 1：*fclk* 为 40MHz，而且将位速率设定为 500kbps 时， BRP=3、 TSEG1=6、 TSEG2=3 满足条件。
此时，将 H'5200 写入 BCR1，将 H'0003 写入 BCR0。

例 2：*fclk* 为 35MHz，而且将位速率设定为 250kbps 时， BRP=4、 TSEG1=8、 TSEG2=5 满足条件。
此时，将 H'7400 写入 BCR1，将 H'0004 写入 BCR0。

例 3：*fclk* 为 32MHz，而且将位速率设定为 500kbps 时， BRP=1、 TSEG1=11、 TSEG2=4 满足条件。
此时，将 H'A300 写入 BCR1，将 H'0001 写入 BCR0。

例 4：*fclk* 为 20MHz，而且将位速率设定为 500kbps 时， BRP=1、 TSEG1=6、 TSEG2=3 满足条件。
此时，将 H'5200 写入 BCR1，将 H'0001 写入 BCR0。

19.4.4　中断请求寄存器 （IRR）

IRR 为 16 位可读 / 写寄存器，由各中断源的状态标志构成。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRR15	IRR14	IRR13	IRR12	IRR11	IRR10	IRR9	IRR8	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W

位	位名称	初始值	R/W	说明
15	IRR15	0	R/W	定时器比较匹配中断 1 表示产生定时器比较匹配寄存器 1 （TCMR1）的比较匹配。 TCMR1 的设定值与周期时间相同时 （TCMR1=CYCTR），此位进行置位。 0：不产生 TCMR1 的定时器比较匹配 ［清除条件］写入 1 1：产生 TCMR1 的定时器比较匹配 ［置位条件］ TCMR1 与周期时间相同 （TCMR1=CYCTR）
14	IRR14	0	R/W	定时器比较匹配中断 0 表示产生定时器比较匹配寄存器 0 （TCMR0）的比较匹配。 TCMR0 的设定值与本地时间相同时 （TCMR0=TCNTR），此位进行置位。 0：不发生 TCMR0 的定时器比较匹配 ［清除条件］写入 1 1：产生 TCMR0 的定时器比较匹配 ［置位条件］ TCMR0 与定时器的值相同 （TCMR0=TCNTR）
13	IRR13	0	R/W	定时器溢出中断 /Next_is_Gap 接收中断 / 信息错误中断 此中断根据 RCAN-TL1 的模式具有不同的含义，表示如下状态： • 事件触发模式下 （含测试模式）运行时，发生定时器 （TCNTR）溢出。 • 时间触发模式下 （含测试模式）运行时， Next_is_Gap 接收已设置的时间基准信息。 • 测试模式下发生信息错误。但即使在测试模式下产生信息溢出条件，此位也不进行置位。 0：在事件触发模式 （含测试模式）下，不发生定时器 （TCNTR）溢出 时间触发模式 （含测试模式）下，不接收带 Next_is_Gap 的时间基准信息 测试模式下，不发生信息错误 ［清除条件］写入 1 1：［置位条件］事件触发模式 （含测试模式）下，发生定时器 （TCNTR）溢出，并从 H'FFFF 变为 H'0000 定时触发模式 （含测试模式）下，接收带 Next_is_Gap 的时间基准信息测试模式下，发生信息错误

位	位名称	初始值	R/W	说明
12	IRR12	0	R/W	CAN 睡眠模式时总线激活 此位表示 CAN 总线激活的存在。 RCAN-TL1 为 CAN 睡眠模式时，如果在 CAN 总线检测出显性位，则此位进行置位。向此位写入 1 时清除中断。写入 0 时忽略。不使用自动唤醒功能，并且不接收此中断时，必须通过对应的屏蔽位使此中断无效。不使用自动唤醒功能时如果请求此中断，请从 CAN 睡眠模式返回后清除此位。这是为了防止通过接收流水线的新的下降沿再次置位中断。 请注意：此中断的置位时间与 GSR4 的清除时间不同。 0：总线空闲状态 ［清除条件］写入 1 1：CAN 睡眠模式下检测 CAN 总线激活 ［置位条件］ CAN 睡眠模式下，在 CRxn 中检测向显性的位状态变化
11	IRR11	0	R/W	定时器比较匹配中断 2 表示产生定时器比较匹配寄存器 2 （TCMR2）的比较匹配。 TCMR2 的设定值与周期时间相同时 （TCMR2=CYCTR），此位进行置位。 0：不产生 TCMR2 的定时器比较匹配 ［清除条件］写入 1 1：产生 TCMR2 的定时器比较匹配 ［置位条件］ TCMR2 与周期时间相同 （TCMR2=CYCTR）
10	IRR10	0	R/W	起始系统矩阵中断 表示开始下个系统矩阵。当 CCR=0 时，完成时间基准信息的发送 / 接收时，此位进行置位。当 CMAX=0 时，此中断设置为每次周期计数（BasicCycle）。 0：不是新的系统矩阵的起始 ［清除条件］写入 1 1：周期计数器为 0 ［置位条件］ CMAX!=B'111 而且 CCR=0 时，完成时间基准信息的发送 / 接收
9	IRR9	0	R	信息溢出 / 重写中断标志 表示虽然此位已接收信息，但所对应的 RXPR 或 RFPR 已置 1，所以不读取邮箱内的信息，由于 CPU 未清除，所以通过 NMC 位的设定，新接收的信息被放弃 （溢出）或重写。将 1 写入 UMSR 寄存器的对应位时，清除此位。另外，也可通过向 MBIMR 的所有对应位写入 1 来进行清除。对此位的写入无效。 0：无信息溢出 / 重写通知 ［清除条件］清除 UMSR 的所有位，或者对应已经置位的全部 UMSRMBIMR 置位 1：由于发生溢出，放弃接收信息。或者重写信息 ［置位条件］对应的 PXPR 或 RFPR=1，而且 MBIMR=0 时接收信息

位	位名称	初始值	R/W	说明
8	IRR8	0	R	邮箱空中断标志 正常发送设定为发送用的 1 条信息 （所对应的 TXACK 标志置位）或发送中断 （将执行发送取消的信息的 ABACK 标志置位）时，此位进行置位。此时，在事件触发模式下，清除对应的 TXPR，邮箱可接收下一个发送用信息数据。时间触发模式下，为了持续进行每个已编程的周期计数 （Basic Cycle）的发送，在发送完成后，不清除邮箱 24 ～ 30 的 TXPR。 实际上，此位通过未通过 MBIMR 标志进行屏蔽的 TXACK 与已取得 ABACK 位的 OR 的信号进行置位。因此，清除所有的 TXACK 及 ABACK 位时，自动清除此位。另外，即使向 MBIMR 对应的所有位写入 1，也可清除。向此位的写入无效。 0：进行发送或发送取消的信息未在处理中 ［清除条件］清除所有 TXACK 及 ABACK 的位或置位已进行置位的所有的 TXACK 及 ABACK 位对应的 MBIMR 1：发送信息或进行发送中断 （发送取消），可保存下一条信息 （在时间触发模式下，邮箱 24 ～ 30 仅在中断时可保存下一条信息） ［置位条件］ TXACK 或 ABACK 位进行置位时 （对应的 MBIMR=0 时）
7	IRR7	0	R/W	超载帧 表示 RCAN-TL1 检测出超载帧发送的标志。 IRR7 写入 1 时，进行清除。写入 0 时忽略。 0：［清除条件］写 1 1：［置位条件］检测出超载帧条件
6	IRR6	0	R/W	总线断开中断标志 RCAN-TL1 为总线断开状态时或从总线断开状态返回错误主动状态时，此位进行置位。因此，节点的 TEC ≥ 256、总线断开返回顺序结束 （128 次连续接收 11 个隐性位）或从总线断开转移至停止状态 （自动或手动）成为源。RCAN-TL1 解除总线断开后，此位仍为置位状态，所以需要通过软件进行清除。请通过软件读取 GSR0，并判断 RCAN-TL1 为总线断开状态或错误主动状态。即使节点仍为总线断开状态，写入 1 时，仍清除此位。写入 0 时忽略。 0：［清除条件］写入 1 1：通过发送错误转移至总线断开状态或从总线断开状态返回错误主动状态 ［置位条件］ TEC ≥ 256 或 128 次连续接收 11 个隐性位后，结束总线断开或从总线断开转移至停止状态
5	IRR5	0	R/W	错误被动中断标志 表示通过接收或发送错误计数器所产生的错误被动状态或通过测试模式强行进入错误被动状态。此位写入 1 时，进行清除。写入 0 时忽略。有时即使清除此位，节点仍旧保持错误被动状态。需要通过软件检查 GSR0 与 GSR5，判断 RCAN-TL1 为错误被动状态或总线断开状态。 0：［清除条件］写入 1 1：通过发送 / 接收错误产生的错误被动状态 ［置位条件］使用 TEC ≥ 128、 REC ≥ 128 或错误被动测试模式

位	位名	初始值	R/W	说明
4	IRR4	0	R/W	接收错误计数器警告中断标志 RCAN-TL1 不为总线断开状态时，如果接收错误计数器 （REC）超过 95，则进行置位。此位写入 1 时，清除中断。写入 0 时忽略。 0：［清除条件］写入 1 1：通过接收错误产生的错误警告状态 ［置位条件］ RCAN-TL1 为总线断开状态以外， REC ≥ 96 时
3	IRR3	0	R/W	发送错误计数器警告中断标志 RCAN-TL1 不为总线断开状态时，发送错误计数器 （TEC）超过 95 时，进行置位。此位写入 1 时，清除中断。写入 0 时忽略。 0：［清除条件］写入 1 1：通过发送错误产生的错误警告状态 ［置位条件］ TEC ≥ 96
2	IRR2	0	R	远程帧接收中断标志 表示邮箱接收远程帧。至少 1 个对应的 MBIMR 未进行置位的邮箱中保存远程帧发送请求时，进行置位。清除远程帧接收完成寄存器 （RFPR）的所有位时，此位进行清除。另外，向 MBIMR 对应的所有位写入 1 时，进行清除。向此位的写入无效。 0：［清除条件］清除 RFPR 的所有位 1：至少 1 个远程帧请求为处理等待状态 ［置位条件］对应的 MBIMR=0 时，接收远程帧
1	IRR1	0	R	数据帧接收中断标志 表示有处理等待状态的接收数据帧。此位为 1 时，至少 1 个邮箱保存未处理的信息。清除数据帧接收完成寄存器（RXPR）的所有位时（任何接收邮箱中均无未处理的信息），进行清除。设定 MBIMR 不为 0 的各接收邮箱的 RXPR 标志的逻辑和。 另外，向 MBIMR 对应的所有位写入 1 时，进行清除。向此位的写入无效。 0：［清除条件］清除 RXPR 的所有位 1：接收数据帧，保存至邮箱中 ［置位条件］对应的 MBIMR=0 时，接收数据

位	位名称	初始值	R/W	说明
0	IRR0	1	R/W	复位 / 停机 / 睡眠中断标志 通过下述 3 个源进行置位。 • 软件复位 （MCR0）或硬件复位后，转移至复位模式 • 停机请求 （MCR1）后，转移至停机模式 • 停机模式下，发送睡眠请求 （MCR5）后，转移至 CAN 睡眠模式 将此位进行置位后，读取 GSR，可得知 RCAN-TL1 的状态。 【注】 需要发送睡眠模式请求时，请事先使用停机模式。请参考 MCR5 的说明及图 19.8。 GSR3 或 GSR4 从 0 变为 1 时，或者从停机模式转移至 CAN 睡眠模式时，IRR0 进行置位。因此，RCAN-TL1 在解除停机模式之后，立即重新转移至停机模式时，不清除 GSR4，所以 IRR0 不进行置位。同样，从 CAN 睡眠模式直接转移至停机模式时， IRR0 也不进行置位。从停机模式 /CAN 睡眠模式转移至发送 / 接收运行时，在清除 GSR4 之前，需要 [1 位时间 -TSEG2] ～ [1 位时间 ×2-TSEG2] 的时间。 复位模式时，虽 IRR0 进行置位，但通过初始化， IMR0 自动进行置位，所以不支持向 CPU 的中断。 0：［清除条件］写入 1 1：转移至软件复位模式、停机模式或 CAN 睡眠模式 ［置位条件］在复位 （MCR0 或硬件）、停机模式 （MCR1）或 CAN 睡眠模式 MCR5）请求后，完成向复位 / 停机 /CAN 睡眠模式的转移

19.4.5　中断屏蔽寄存器 （IMR）

IMR 为 16 位可读 / 写寄存器，屏蔽中断请求寄存器 （IRR）各位对应的中断 IRQ 输出信号的生成。将位设定为 1 时，屏蔽所对应的中断请求。 IMR 直接控制 IRQ 的生成，但不禁止 IRR 对应位的置位。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
初始值：	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W：	R/W	R/W	R/W	R/W	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	位名称	初始值	R/W	说明
15 ～ 0	IMR[15:0]	均为 1	R/W	屏蔽 IRR[15:0] 的各位对应的中断源 此位进行置位时，即使将 IRR 对应的位进行置位，也不生成此中断信号。 0：不屏蔽对应的 IRR （产生中断源时，生成 IRQ） 1：屏蔽 IRR 对应的中断

19.4.6　发送错误计数器 （TEC） / 接收错误计数器 （REC）

TEC 及 REC 为可读取 / 带条件写入的 16 位寄存器，为显示 CAN 接口中的发送 / 接收信息错误数量的计数器。计数器值在上述参考文献 [1]、[2]、[3]、[4] 中规定。除写错误计数器测试模式以外，此寄存器为只读，仅可通过 CAN 接口进行重写。通过转移至复位请求 （MCR0）或总线断开状态清除此寄存器。

写错误计数器测试模式 （TST[2:0]=B'100）下，可向此寄存器进行写入。 TEC 及 REC 仅可写相同的值。写入 TEC 的值设定至 TEC 及 REC。写入此寄存器时，需要将 RCAN-TL1 设置为停机模式。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
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】 * 仅限在通过 MCR 的 TST[2:0]=B'100 产生的测试模式时，可进行写入。总线断开状态下， REC 计数总线断开返回顺序所必需的 11 个隐性位的接收次数。

位	位名称	初始值	R/W	说明
15 ～ 8	TEC[7:0]	均为 0	R/W*	发送错误计数器 发送中，检测出基于 CAN 规格的错误时，此寄存器的值递增。
7 ～ 0	REC[7:0]	均为 0	R/W*	接收错误计数器 接收中，检测出基于 CAN 规格的错误时，此寄存器的值递增。

19.5　RCAN-TL1 邮箱寄存器

说明 RCAN-TL1 的邮箱寄存器。 RCAN-TL1 邮箱寄存器控制各邮箱，并表示邮箱的状态。

RCAN-TL1 邮箱寄存器如表 19.7 所示。

【重点】　长字节存取，可作为 2 个连续的字存取来进行。

表 19.7　RCAN-TL1 邮箱寄存器

寄存器名称	简称	地址	存取长度 （位）
发送等候寄存器 1	TXPR1	H'020	32
发送等候寄存器 0	TXPR0	H'022	—
		H'024	
		H'026	
发送取消寄存器 1	TXCR1	H'028	16/32
发送取消寄存器 0	TXCR0	H'02A	16
		H'02C	
		H'02E	
发送应答寄存器 1	TXACK1	H'030	16/32
发送应答寄存器 0	TXACK0	H'032	16
		H'034	
		H'036	
中断应答寄存器 1	ABACK1	H'038	16/32
中断应答寄存器 0	ABACK0	H'03A	16
		H'03C	
		H'03E	
数据帧接收完成寄存器 1	RXPR1	H'040	16/32
数据帧接收完成寄存器 0	RXPR0	H'042	16
		H'044	
		H'046	
远程帧接收完成寄存器 1	RFPR1	H'048	16/32
远程帧接收完成寄存器 0	RFPR0	H'04A	16
		H'04C	
		H'04E	
邮箱中断屏蔽寄存器 1	MBIMR1	H'050	16/32
邮箱中断屏蔽寄存器 0	MBIMR0	H'052	16
		H'054	
		H'056	
未读信息状态寄存器 1	UMSR1	H'058	16/32
未读信息状态寄存器 0	UMSR0	H'05A	16
		H'05C	
		H'05E	

19.5.1　发送等候寄存器 1、 0 （TXPR1、 TXPR0）

TXPR1 与 TXPR0 进行连接，构成保存 CAN 模块的发送等候标志的 32 位寄存器。为 16 位总线接口时，长字存取可作为 2 个连续的字存取来进行。

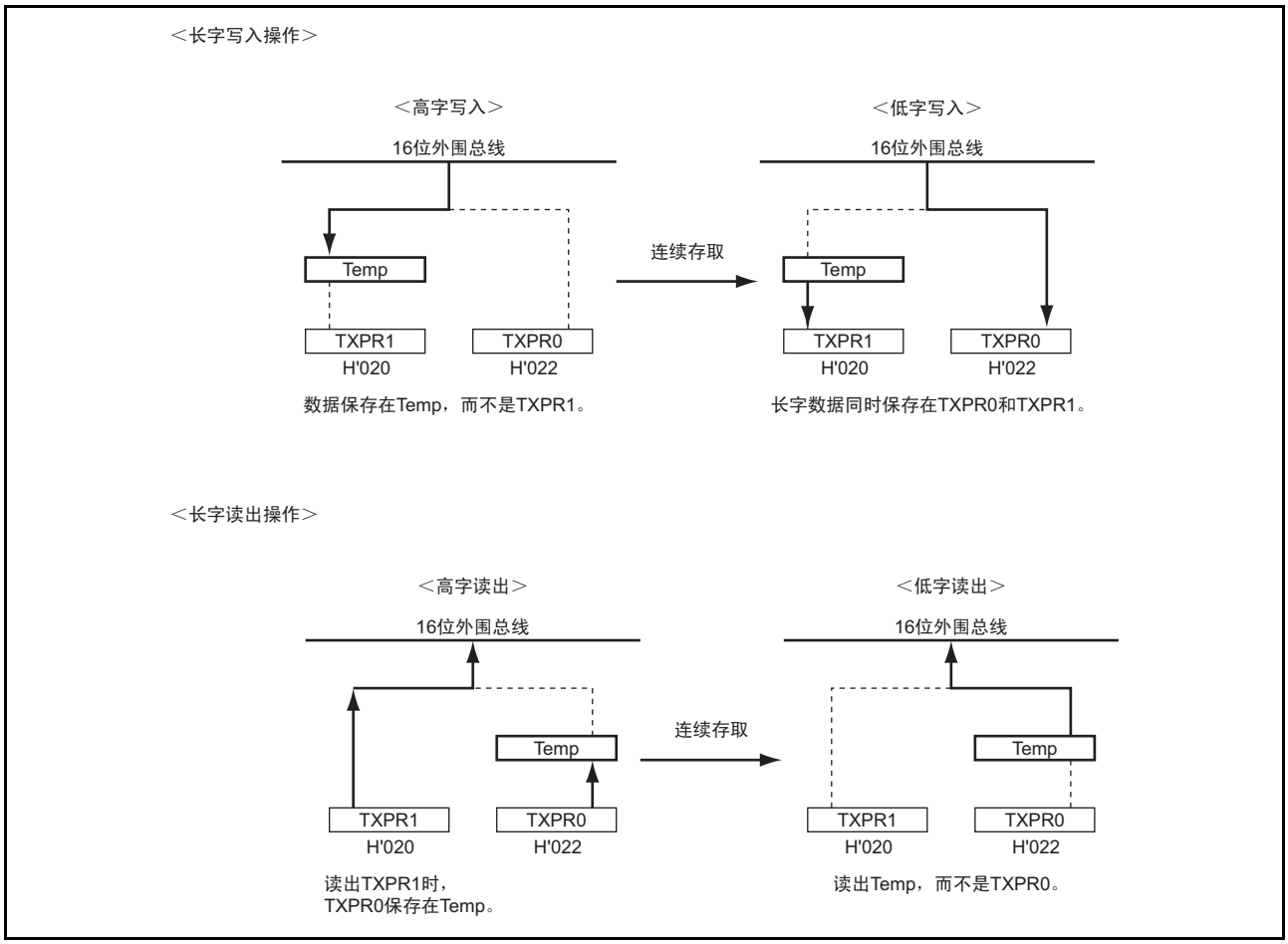


图 19.13　16 位总线接口时的长字存取

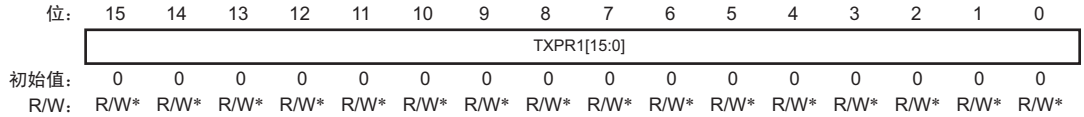
TXPR1 寄存器控制邮箱 31 ～ 16； TXPR0 寄存器控制邮箱 15 ～ 1。通过向位写入 1， CPU 可对所对应的发送信息进行操作。写入 0 无效。清除 TXPR 时，必须将 TXCR 对应的位进行置位。通过读取 TXPR， CPU 可得知发送等候的邮箱及发送中的邮箱。实际上，除邮箱 0 之外的所有邮箱均具有发送等候位。另外，禁止对未设定为发送的邮箱的位写入 1。

事件触发模式下，正常发送信息后或进行来自 TXCR 的发送中断后，RCAN-TL1 清除所对应的发送等候标志。时间触发模式下，为了持续进行每个已编程的周期计数 （Basic Cycle）的发送，发送结束后不清除邮箱 30 ～ 24 的 TXPR。由于 CAN 节点仲裁失败或 CAN 总线发生错误，所以不发送信息时不清除 TXPR 的标志。此时，如果所对应的邮箱的信息控制字段的 DART （自动再发送无效）位未进行置位， RCAN-TL1 则自动尝试再发送。 DART 进行置位时，清除发送，通过邮箱空中断标志 （IRR8）与中断应答寄存器 （ABACK）对应的位通知 CPU。

TXPR 的状态发生变化时，即使在总线仲裁失败或 CAN 总线中发生错误的情况下， RCAN-TL1 仍常处理发送 ID 所示优先级 （MCR2=0）最高的信息。关于详细内容请参考 “19.7 运行说明”。

RCAN-TL1 将 TXPR 位的状态变更为 0 时，有时生成邮箱空中断标志 （IRR8）。这表示邮箱的发送正常结束或中断。信息发送正常结束时，反映在 TXACK 中。信息发送中断时，反映在 ABACK 中。通过检查这些寄存器，可将所对应的邮箱的信息数据字段的内容重写为下次发送用。

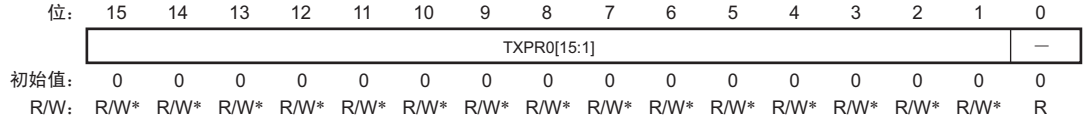
（1） TXPR1



【注】 * 设定为发送用的邮箱所对应的位，仅可写入 1。

位	位名称	初始值	R/W	说明
15 ～ 0	TXPR1[15:0]	H'0000	R/W	请求向对应的邮箱发送 CAN 帧。 bit15 ～ 0 对应于邮箱 31 ～ 16。多个位被置位时的发送顺序，通过 MCR2 位的设定，按照信息 ID 优先级或邮箱号码顺序进行发送。 0：所对应的邮箱为发送信息空闲状态 ［清除条件］信息发送结束 （事件触发信息时）或信息发送中断（自动进行清除） 1：向所对应的邮箱产生发送请求

（2） TXPR0



【注】 * 设定为发送用邮箱所对应的位，仅可写入 1。
TXPR1/TXPR0 的读 / 写，必须以长字存取来进行。

位	位名称	初始值	R/W	说明
15 ～ 1	TXPR0[15:1]	H'0000	R/W	表示向所对应的邮箱产生 CAN 帧发送请求。 bit15 ～ 1 对应于邮箱 15 ～ 1。多个位进行置位时的发送顺序，通过 MCR2 位的设定，按照信息 ID 优先级或邮箱号码的顺序进行发送。 0：所对应的邮箱为发送信息空闲状态 ［清除条件］信息发送结束 （事件触发信息时）或信息发送中断（自动进行清除） 1：向所对应的邮箱产生发送请求
0	—	0	R	保留位 由于为接收专用邮箱，此位常为 0。写 1 时忽略。读取值为 0。

19.5.2　发送取消寄存器 1、 0 （TXCR1、 TXCR0）

TXCR1 与 TXCR0 为 16 位可读取 / 带条件写入的寄存器， TXCR1 控制邮箱 31 ～ 16 ； TXCR0 控制邮箱 15 ～ 1。 CPU 使用 TXCR，取消 TXPR 的发送请求。为了清除 TXPR 的位，请从 CPU 向对应的 TXCR 的位写入 1。写入 0 时忽略。

正常执行中断时，CAN 控制器清除所对应的 TXPR 与 TXCR 的位，并将 ABACK 位进行置位。但邮箱开始发送后，不可通过此寄存器的位来取消。此时，发送正常结束时，CAN 控制器清除 TXPR 与 TXCR 的位，并将 TXACK 的位进行置位。但由于仲裁失败或总线中发生错误，不可进行发送时， CAN 控制器清除对应的 TXPR 与 TXCR 的位，并将 ABACK 位进行置位。即使 CPU 要清除不为发送等待状态的邮箱的发送，仍被忽略。此时， CPU 不可将 TXCR 的标志进行置位。

（1） TXCR1

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXCR1[15:0]															
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*

【注】 * 对应于接受发送请求的发送邮箱的位，仅可写入 1。

位	位名称	初始值	R/W	说明
15 ～ 0	TXCR1[15:0]	H'0000	R/W	向进入发送用队列的邮箱请求发送取消。 bit15 ～ 0 对应于邮箱 31 ～ 16 （及 TXPR1[15:0]）。 0：对应的邮箱为发送信息取消空闲状态 ［清除条件］发送信息取消结束 （自动进行清除） 1：向对应的邮箱请求发送取消

（2） TXCR0

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXCR0[15:1]															—
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

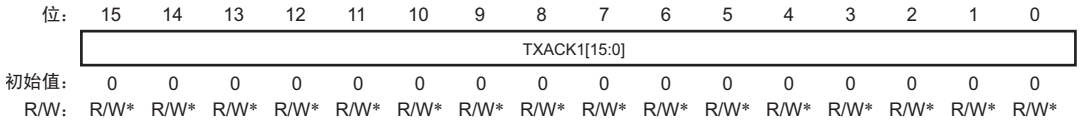
【注】 * 对应于接受发送请求的发送邮箱的位，仅可写入 1。

位	位名称	初始值	R/W	说明
15 ～ 1	TXCR0[15:1]	H'0000	R/W	向进入发送用队列的邮箱请求发送取消。 bit15 ～ 1 对应于邮箱 15 ～ 1 （及 TXPR0[15:1]）。 0：对应的邮箱为发送信息取消空闲状态 ［清除条件］发送信息取消结束 （自动进行清除） 1：向对应的邮箱请求发送取消
0	—	0	R	保留位 由于为接收专用邮箱，此位常为 0。写 1 时忽略，读取值常为 0。

19.5.3　发送应答寄存器 1、 0 （TXACK1、 TXACK0）

TXACK1 与 TXACK0 为 16 位可读取 / 带条件写入的寄存器，用于通知 CPU，邮箱的发送已正常进行。正常进行发送时， RCAN-TL1 将 TXACK 寄存器所对应的位进行置位。 CPU 通过写入 1，可清除 TXACK 的位。写入 0 时忽略。

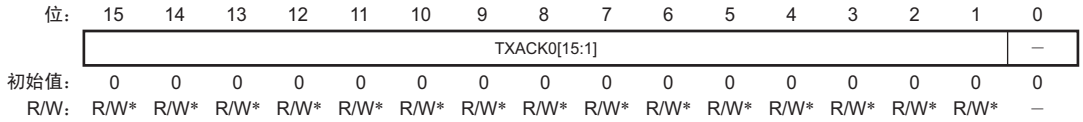
（1） TXACK1



【注】 * 为了清除位，仅可写入 1。

位	位名称	初始值	R/W	说明
15 ～ 0	TXACK1[15:0]	H'0000	R/W	通知对应的邮箱所请求的发送已正常进行。 bit15 ～ 0 对应于邮箱 31 ～ 16。 0：［清除条件］写入 1 1：已正常发送对应的邮箱的信息 （数据帧或远程帧） ［置位条件］完成对应的邮箱的信息发送

（2） TXACK0



【注】 * 为了清除位，仅可写入 1。

位	位名称	初始值	R/W	说明
15 ～ 1	TXACK0[15:1]	H'0000	R/W	通知对应的邮箱所请求的发送已正常进行。 bit15 ～ 1 对应于邮箱 15 ～ 1。 0：［清除条件］写入 1 1：已正常发送对应邮箱的信息 （数据帧或远程帧） ［置位条件］完成对应邮箱的信息发送
0	—	0	R	保留位 由于为接收专用邮箱，此位常为 0。写 1 时忽略，读取值常为 0。

19.5.4　中断应答寄存器 1、 0 （ABACK1、 ABACK0）

ABACK1 与 ABACK0 为 16 位可读取 / 带条件写入的寄存器，用于通知 CPU，按照请求已中断邮箱发送。进行中断时， RCAN-TL1 将 ABACK 寄存器的对应位进行置位。CPU 通过写入 1，可清除 ABACK 的位。写入 0 时忽略。通过 RCAN-TL1 将 ABACK 的位进行置位，确认对应的 TXCR 位已清除 TXPR 位。

（1） ABACK1

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ABACK1[15:0]															
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*

【注】 * 为了清除位，仅可写入 1。

位	位名称	初始值	R/W	说明
15 ～ 0	ABACK1[15:0]	H'0000	R/W	通知对应的邮箱所请求的发送取消已正常进行。 bit15 ～ 0 对应于邮箱 31 ～ 16。 0：［清除条件］写入 1 1：已取消对应邮箱的信息 （数据帧或远程帧）发送 ［置位条件］取消对应邮箱的信息发送

（2） ABACK0

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ABACK0[15:1]															ABCS
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

【注】 * 为了清除位，仅可写入 1。

位	位名称	初始值	R/W	说明
15 ～ 1	ABACK0[15:1]	H'0000	R/W	通知对应的邮箱所请求的发送取消已正常进行。 bit15 ～ 1 对应于邮箱 15 ～ 1。 0：［清除条件］写入 1 1：已取消对应邮箱的信息 （数据帧或远程帧）发送 ［置位条件］取消对应邮箱的信息发送
0	—	0	R	保留位 由于为接收邮箱，此位常为 0。写 1 时忽略，读取值常为 0。

19.5.5　数据帧接收完成寄存器 1、 0 （RXPR1、 RXPR0）

RXPR1 与 RXPR0 为 16 位可读 / 带条件写入的寄存器，设定为接收用的邮箱由表示接收数据帧的标志构成。CAN 数据帧正常地保存至接收邮箱时，置位 RXPR 的对应位。写入 1 时，清除对应的位。写入 0 时无效。但邮箱仅通过 MBC（邮箱结构）设定为接收数据帧时，才进行置位。将 RXPR 的位进行置位时，如果对应的 MBIMR 未进行置位，则 IRR1 （数据帧接收中断标志）仍进行置位。而且如果 IMR1 未进行置位，则生成中断信号。此寄存器的位仅通过接收数据帧进行置位，接收远程帧时，不进行置位。

（1） RXPR1

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXPR1[15:0]															
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*

【注】 * 为了清除位，仅可写入 1。

位	位名称	初始值	R/W	说明
15 ～ 0	RXPR1[15:0]	H'0000	R/W	bit15 ～ 0 对应于设定为接收用的邮箱号码 31 ～ 16 的邮箱。 0：［清除条件］写入 1 1：对应的邮箱已接收 CAN 数据帧 ［置位条件］对应邮箱的数据帧接收结束

（2） RXPR0

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXPR0[15:0]															
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*

【注】 * 为了清除位，仅可写入 1。

位	位名称	初始值	R/W	说明
15 ～ 0	RXPR0[15:0]	H'0000	R/W	bit15 ～ 0 对应于设定为接收用的邮箱号码 15 ～ 0 的邮箱。 0：［清除条件］写入 1 1：对应的邮箱已接收 CAN 数据帧 ［置位条件］对应邮箱的数据帧接收结束

19.5.6　远程帧接收完成寄存器 1、 0 （RFPR1、 RFPR0）

RFPR1 与 RFPR0 为 16 位可读取 / 带条件写入的寄存器，设定为接收用的邮箱由表示接收远程帧的标志构成。 CAN 远程帧正常地保存到接收邮箱时，将 RFPR 的对应位进行置位。写入 1 时，清除对应的位。写入 0 时无效。所有的邮箱均具有位，但仅在邮箱通过 MBC（邮箱结构）设定为接收远程帧时才将位进行置位。将 RFPR 的位进行置位时，如果对应的 MBIMR 未进行置位，则 IRR2 （数据帧接收中断标志）仍进行置位。而且如果 IMR2 未进行置位，则生成中断信号。此寄存器的位仅通过接收远程帧进行置位，接收数据帧时不进行置位。

（1） RFPR1

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFPR1[15:0]															
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*

【注】 * 为了清除位，仅可写入 1。

位	位名称	初始值	R/W	说明
15 ～ 0	RFPR1[15:0]	H'0000	R/W	邮箱 31 ～ 16 的远程请求接收标志。 0：［清除条件］写入 1 1：对应的邮箱已接收 CAN 远程帧 ［置位条件］对应邮箱的远程帧接收结束

（2） RFPR0

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFPR0[15:0]															
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*

【注】 * 为了清除位，仅可写入 1。

位	位名称	初始值	R/W	说明
15 ～ 0	RFPR0[15:0]	H'0000	R/W	邮箱 15 ～ 0 的远程请求接收标志。 0：［清除条件］写入 1 1：对应的邮箱已接收 CAN 远程帧 ［置位条件］对应邮箱的远程帧接收结束

19.5.7　邮箱中断屏蔽寄存器 1、 0 （MBIMR1、 MBIMR0）

MBIMR1 与 MBIMR0 为 16 位可读 / 写寄存器。 MBIMR 为屏蔽邮箱运行有关的 IRR （IRR1：数据帧接收中断、IRR2：远程帧接收中断、IRR8：邮箱空中断、IRR9：信息溢出 / 重写中断）的寄存器。邮箱设定为接收时，屏蔽通过接收中断标志（IRR1、IRR2、IRR9）产生的中断。不影响 RXPR、RFPR、UMSR 位的置位。邮箱设定为发送时，屏蔽通过发送或发送中断（IRR8）产生的中断或邮箱空中断。不影响通过发送产生的 TXPR/TXCR 位的清除与 TXACK 位的置位，通过发送中断产生的 TXPR/TXCR 位的清除与 ABACK 位的置位。

通过向屏蔽的邮箱所对应的位写入 1 来设定屏蔽。复位时，屏蔽所有的邮箱中断。

（1） MBIMR1

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIMR1[15:0]															
初始值：	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W：	R/W	R/W	R/W	R/W	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	位名称	初始值	R/W	说明
15 ～ 0	MBIMR1[15:0]	H'FFFF	R/W	允许 / 禁止来自邮箱 31 ～ 16 的中断请求。 0：允许 IRR1、 IRR2、 IRR8、 IRR9 的中断请求 1：禁止 IRR1、 IRR2、 IRR8、 IRR9 的中断请求

（2） MBIMR0

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIMR0[15:0]															
初始值：	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W：	R/W	R/W	R/W	R/W	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	位名称	初始值	R/W	说明
15 ～ 0	MBIMR0[15:0]	H'FFFF	R/W	允许 / 禁止来自邮箱 15 ～ 0 的中断请求。 0：允许 IRR1、 IRR2、 IRR8、 IRR9 的中断请求 1：禁止 IRR1、 IRR2、 IRR8、 IRR9 的中断请求

19.5.8　未读信息状态寄存器 1、 0 （UMSR1、 UMSR0）

UMSR1 与 UMSR0 为 16 位可读取 / 带条件写入的寄存器。接收新信息时，记录未通过 CPU/DMAC 存取内容的接收邮箱。接收邮箱的新信息时，如果未通过 CPU 清除 RXPR 或 RFPR 的对应位，则 UMSR 的位进行置位。写入 1 时，进行清除。写入 0 时无效。

邮箱设定为发送时，对应的 UMSR 的位不进行置位。

（1） UMSR1

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UMSR1[15:0]															
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*

【注】 * 为了清除位，仅可写入 1。

位	位名称	初始值	R/W	说明
15 ～ 0	UMSR1[15:0]	H'0000	R/W	表示邮箱 31 ～ 16 的未读接收信息发生重写或溢出。 0：［清除条件］写入 1 1：未读接收信息由于新信息发生重写或溢出 ［置位条件］清除 RXPR 或 RFPR 前接收新信息

（2） UMSR0

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UMSR0[15:0]															
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*

【注】 * 为了清除位，仅可写入 1。

位	位名称	初始值	R/W	说明
15 ～ 0	UMSR0[15:0]	H'0000	R/W	表示邮箱 15 ～ 0 的未读接收信息发生重写或溢出。 0：［清除条件］写入 1 1：未读接收信息由于新信息发生重写或溢出 ［置位条件］清除 RXPR 或 RFPR 前接收新信息

19.6　定时器寄存器

RCAN-TL1 为 16 位定时器，支持多种时钟源。通过预分频计数器可降低时钟的速度。另外，具有 3 个比较匹配寄存器 （TCMR2、 TCMR1、 TCMR0）。地址映射如下所示：

【重点】　定时器寄存器仅可存取所有字 （16 位）。

表 19.8　RCAN-TL1 的定时器寄存器的构成

寄存器名称	简称	地址	存取长度 （位）
时间触发控制寄存器 0	TTCR0	H'080	16
最大周期 /Tx_Enable_Window 寄存器	CMAX_TEW	H'084	16
基准触发偏移寄存器	RFTROFF	H'086	16
定时器状态寄存器	TSR	H'088	16
周期计数器寄存器	CCR	H'08A	16
定时器计数器寄存器	TCNTR	H'08C	16
周期时间寄存器	CYCTR	H'090	16
基准标记寄存器	RFMK	H'094	16
定时器比较匹配寄存器 0	TCMR0	H'098	16
定时器比较匹配寄存器 1	TCMR1	H'09C	16
定时器比较匹配寄存器 2	TCMR2	H'0A0	16
发送触发时间选择寄存器	TTTSEL	H'0A4	16

19.6.1　定时触发控制寄存器 0 （TTCR0）

TTCR0 为 16 位可读 / 写寄存器。

控制定时器的运行。请在设定本寄存器并运行定时器后，进行定周期发送或监视用寄存器的设定。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	—	—	—	TCR6	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0
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	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说明
15	TCR15	0	R/W	定时器允许 将此位进行置位时，运行 TCNTR。清除时，清除 TCNTR 与 CCR。 0：清除 TCNTR 与 CCR，停止运行 1：定时器运行
14	TCR14	0	R/W	时戳值 指定使用周期时间 （CYCTR）或使用 CCR[5:0]+CYCTR[15:6] 作为邮箱 15 ～ 0 的发送及接收的时戳。此功能便于在时间触发模式下监视 Rx_Trigger 此寄存器不影响邮箱 30 及 31 的时戳。 0：使用 CYCTR[15:0] 为邮箱 15 ～ 0 的时戳 1：使用 CCR[5:0]+CYCTR[15:6] 为邮箱 15 ～ 0 的时戳
13	TCR13	0	R/W	通过 TCMR2 取消 此位及 bit12 为置位状态， RCAN-TL1 在停机模式以外的模式下，发生比较匹配时，该 TXPR 位对应的所有 TXCR 位自动进行置位。因此取消发送信号中的信息。 0：禁止通过 TCMR2 的比较匹配产生取消发送 1：允许通过 TCMR2 的比较匹配产生取消发送
12	TCR12	0	R/W	比较匹配允许 将此位进行置位时， IRR11 通过 TCMR2 的比较匹配进行置位。 0：IRR11 不通过 TCMR2 的比较匹配进行置位 1：IRR11 通过 TCMR2 的比较匹配进行置位
11	TCR11	0	R/W	比较匹配允许 将此位进行置位时， IRR15 通过 TCMR1 的比较匹配进行置位。 0：IRR15 不通过 TCMR1 的比较匹配进行置位 1：IRR15 通过 TCMR1 的比较匹配进行置位
10	TCR10	0	R/W	比较匹配允许 将此位进行置位时， IRR14 通过 TCMR0 的比较匹配进行置位。 0：IRR14 不通过 TCMR0 的比较匹配进行置位 1：IRR14 通过 TCMR0 的比较匹配进行置位
9 ～ 7	—	均为 0	R	保留位 读写的值总是 0。

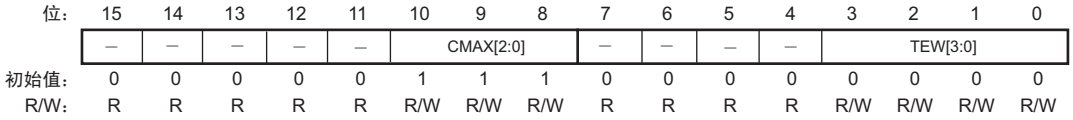
位	位名称	初始值	R/W	说明
6	TCR6	0	R/W	通过 TCMR0 产生的定时器清除控制 指定 TCNTR 与 TCMR0 相同时，是否将定时器清除为 H'0000。另外， TCMR0 也可通过 IRR14 对 CPU 产生中断信号。 【注】RCAN-TL1 在 TTCAN 模式下运行时 （CMAX ≠ B'111），为了不清除本地时间，请将此位设定为 0。 0：不通过 TCMR0 进行定时器清除 1：通过 TCMR0 进行定时器清除
5 ～ 0	TPSC5 ～ TPSC0	均为 0	R/W	定时器预分频器 通过此位的设定，可分频使用定时器的源时钟（4×[RCAN-TL1 的系统时钟]）。 预分频器仅在事件触发模式下有效。 TTCAN 模式下 （CMAX ≠ B'111），自动选择 1 位时序 （=CAN 总线中的 1 位长度）作为 TCNTR 的源时钟。 源时钟周期与定时器周期的关系如下所示： 000000：1X 源时钟 000001：2X 源时钟 000010：3X 源时钟 000011：4X 源时钟 000100：5X 源时钟 ： 111111：64X 源时钟

19.6.2　最大周期 /Tx_Enable_Window 寄存器 （CMAX_TEW）

CMAX_TEW 为 16 位可读 / 写寄存器。

CMAX 指定时间触发发送用周期计数器 （CCR）的最大值，因此设定系统矩阵的周期计数数 （Basic Cycle）。周期计数器为最大值时 （CCR=CMAX）， 1 个周期计数后周期计数器清 0，产生 IRR10 中断。

TEW 指定 Tx_Enable_Window 宽度。



位	位名称	初始值	R/W	说明
15 ～ 11	—	均为 0	R	保留位 读写的值总是 0。
10 ～ 8	CMAX[2:0]	111	R/W	周期计数器最大值 表示 CCR 的最大值。 周期计数器最大值 =2^{CMAX}-1 （CMAX= 寄存器值） 时间触发发送用矩阵周期中可周期计数数为 （周期计数器最大值 +1）。设定 CMAX ≠ B'111 时，RCAN-TL1 为时间触发模式，可使用时间触发功能。 设定 CMAX=B'111 时， RCAN-TL1 为事件触发模式。 000：周期计数器最大值 =0 001：周期计数器最大值 =1 010：周期计数器最大值 =3 011：周期计数器最大值 =7 100：周期计数器最大值 =15 101：周期计数器最大值 =31 110：周期计数器最大值 =63 111：RCAN-TL1 为事件触发模式 （清除 CCR） 【注】使用事件触发模式时，请设定为 CMAX=B'111。
7 ～ 4	—	均为 0	R	保留位 读写的值总是 0。
3 ～ 0	TEW[3:0]	0000	R/W	Tx_Enable_Window 设定 Tx_Enable_Window 宽度。 TEW=B'0000 时，窗口宽度为 1 位时序。 可设定 B'0000 ～ B'1111 的所有的值。 0000：Tx_Enable_Window 宽度 =1 0001：Tx_Enable_Window 宽度 =2 0010：Tx_Enable_Window 宽度 =3 0011：Tx_Enable_Window 宽度 =4 ： 1111：Tx_Enable_Window 宽度 =16 【注】由于 RCAN-TL1 的 CAN 数据链路控制器从发送请求开始发送，通常需要 CAN 总线 1 ～ 2 位时序。因此，上述的值未考虑此精度。

19.6.3　基准触发偏移寄存器 （RFTROFF）

RFTROFF 为 8 位可读 / 写寄存器。

邮箱 30 的发送触发时间 （TTT）可设定偏移为 -127 ～ +127。附加 RFTROFF 寄存器值的符号扩展值后，邮箱 30 的 TTT 与 CYCTR 进行比较。

但不变更 TTT 的值。此偏移值仅影响邮箱 30。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFTROFF[7:0]								—	—	—	—	—	—	—	—
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	R	R	R	R	R	R	R

位	位名称	初始值	R/W	说明
15 ～ 8	RFTROFF [7:0]	均为 0	R/W	表示基准触发偏移值。 00000000：ref_trigger_offset=+0 00000001：ref_trigger_offset=+1 00000010：ref_trigger_offset=+2 ： 01111111：ref_trigger_offset=+127 ： 11111111：ref_trigger_offset=-1 11111110：ref_trigger_offset=-2 ： 10000001：ref_trigger_offset=-127
7 ～ 0	—	均为 0	R	保留位 读写的值总是 0。

19.6.4　定时器状态寄存器 （TSR）

TSR 为 16 位只读寄存器，因此 CPU 可监视定时器比较匹配状态及时间溢出状态。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	TSR4	TSR3	TSR2	TSR1	TSR0
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	位名称	初始值	R/W	说明
15 ～ 5	—	均为 0	R	保留位 读写的值总是 0。
4	TSR4	0	R*	起始系统矩阵 表示开始下一个系统矩阵。 CCR=0 时，此位在时间基准信息的发送 / 接收完成时进行置位。 0：不是下一个系统矩阵的起始 ［清除条件］ 将 1 写入 IRR10 （周期计数器溢出中断标志） 1：周期计数器为 0 ［置位条件］ 周期计数器的值从最大值 （CMAX）变为 H'0 CMAX!=B'111 而且 CCR=0 时，时间基准信息的发送 / 接收完成
3	TSR3	0	R*	定时器比较匹配标志 2 表示发生定时器比较匹配寄存器 2 （TCMR2）的比较匹配。 TCMR2 所设定的值与周期时间寄存器相同 （TCMR2=CYCTR）时，如果 TTCR0 的 bit12 设定为 1，则将此位进行置位。此位为读取专用，在清除 IRR11 （定时器比较匹配中断 2 标志）时进行清除。 0：不发生 TCMR2 的定时器比较匹配 ［清除条件］ 将 1 写入 IRR11 （定时器比较匹配中断 2 标志） 1：发生 TCMR2 的定时器比较匹配 ［置位条件］ TTCR0 的 bit12 为 1 时，TCMR2 与周期时间相同（TCMR2=CYCTR）
2	TSR2	0	R*	定时器比较匹配标志 1 表示发生定时器比较匹配寄存器 1 （TCMR1）的比较匹配。 TCMR1 所设定的值与周期时间寄存器相同 （TCMR1=CYCTR）时，如果 TTCR0 的 bit11 设定为 1，则将此位进行置位。此位为读取专用，在清除 IRR15 （定时器比较匹配中断 1 标志）时进行清除。 0：不发生 TCMR1 的定时器比较匹配 ［清除条件］ 将 1 写入 IRR15 （定时器比较匹配中断 1 标志） 1：发生 TCMR1 的定时器比较匹配 ［置位条件］ TTCR0 的 bit11 为 1 时，TCMR1 与周期时间相同（TCMR1=CYCTR）

位	位名称	初始值	R/W	说明
1	TSR1	0	R*	定时器比较匹配标志 0 表示发生定时器比较匹配寄存器 0 （TCMR0）的比较匹配。 TCM02 所设定的值与定时器的值相同 （TCMR0=TCNTR）时，如果 TTCR0 的 bit10 设定为 1，则将此位进行置位。此位为读取专用，在清除 IRR14 （定时器比较匹配中断 0 标志）时进行清除。 0：未发生 TCMR0 的定时器比较匹配 ［清除条件］ 将 1 写入 IRR14 （定时器比较匹配中断 0 标志） 1：发生 TCMR0 的定时器比较匹配 ［置位条件］ TTCR0 的 bit10 为 1 时，TCMR0 与定时器的值相同（TCMR0=TCNTR）
0	TSR0	0	R*	定时器溢出 /Next_is_Gap 接收 / 信息错误 此标志分为 3 种不同功能： 1. 事件触发模式下运行时发生定时器溢出 2. 时间触发模式下运行时，接收带 Next_is_Gap 的时间基准信息， 3. 测试模式下， CAN 总线中检测出的错误发生。 测试模式与其他设定相比，拥有更高的优先级。 0：事件触发模式下不发生定时器 （TCNTR）溢出 时间触发模式下不接收带 Next_is_Gap 的时间基准信息 在测试模式下不发生信息错误 ［清除条件］ 将 1 写入 IRR13 1：［置位条件］ 事件触发模式下发生定时器（TCNTR）溢出，并从 H'FFFF 变为 H'0000 时间触发模式下接收带 Next_is_Gap 的时间基准信息 测试模式下发生信息错误

【注】 * 此位为读取专用， CPU 可监视周期计数器、定时器及比较匹配寄存器的状态。对此位的写入无效。

19.6.5　周期计数器寄存器 （CCR）

CCR 为 6 位可读 / 写寄存器。

为表示时间触发发送的周期计数值 （Basic Cycle）的寄存器。

CCR 的值，由于 RCAN-TL1 作为潜在主时间进行运行或作为从时间进行运行，所以更新方法不同。 CCR 进行如下更新：

RCAN-TL1 作为主时间 （潜在）进行运行时：

- 周期时间 （CYCTR）与邮箱30的发送触发时间每相同一次，就增加+1

或者

- 每接收一次有效的基准信息，就重写邮箱31的MSG_DATA_0[5:0]的值

RCAN-TL1 作为从时间进行运行时：

- 每接收一次有效的基准信息，就以邮箱31的MSG_DATA_0[5:0]的值进行重写

CMAX=B'111 而且 TTCR0[15]=0 时：

- CCR的值常为B'000000

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

位	位名称	初始值	R/W	说明
15 ～ 6	—	均为 0	R	保留位 读写的值总是 0。
5 ～ 0	CCR[5:0]	均为 0	R/W	周期计数器寄存器 表示时间触发发送用矩阵周期的当前的周期计数数。

19.6.6　定时器计数器寄存器 （TCNTR）

TCNTR 为 16 位可读 / 写寄存器。

TCNTR 为自由运行定时器计数器。通过设定为 TTCR0[15]=1，定时器进行计数。

定时器计数器的预分频器，因 RCAN-TL1 的触发模式而发生变化。

- 事件触发模式时 （CMAX=B'111）：TTCR0[5:0]的TPSC值
- 时间触发模式时 （CMAX≠B'111）：CAN总线1位时序

设定 TTCR0[6]=1 时，如果 TCNTR 的值与定时器比较匹配寄存器 0 （TCMR0）相同， TCNTR 清除为 H'0000，并进行累加计数。

时间触发模式下， TCNTR 作为本地时间 （Local time）使用。因此，请设定为 TTCR0[6]=0，并作为自由运行定时器进行运行。

【注】 1. TTCR0[15]=0 时， TCNTR 常为 H'0000。

2. 从设定允许定时器 （TTCR0[15]=1）之后，到 TCNTR 开始计数为止，有多个时钟周期的延迟。这是由于预分频器的内部逻辑所引起的。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCNTR[15:0]															
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*

【注】 * 仅可在进行定时器允许设定时 （TTCR0[15]=1）写入此寄存器。
时间触发模式 （CMAX 为 B'111 以外）下，禁止写入。

位	位名称	初始值	R/W	说明
15 ～ 0	TCNTR[15:0]	H'0000	R/W	表示自由运行定时器的值。

19.6.7　周期时间寄存器 （CYCTR）

CYCTR 为表示周期时间的 16 位只读寄存器。

Cycle_Time （CYCTR） =Local_Time （TCNTR） -Reference_Mark （RFMK）

事件触发模式下， RFMK 固定为 0，所以此寄存器为 TCNTR 的副本。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CYCTR[15:0]															
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	位名称	初始值	R/W	说明
15 ～ 0	CYCTR[15:0]	H'0000	R	表示周期时间。

19.6.8　基准标记寄存器 （RFMK）

RFMK 为 16 位只读寄存器。该寄存器在正常发送或接收基准信息时，用于捕捉基准信息 SOF 中的本地时间。事件触发模式下不使用此寄存器，常清 0。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFMK[15:0]															
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	位名称	初始值	R/W	说明
15 ～ 0	RFMK[15:0]	H'0000	R	基准标记寄存器 表示时间基准信息 SOF 中的 TCNTR 的值。

19.6.9　定时器比较匹配寄存器 0 ～ 2 （TCMR0 ～ 2）

TCMR0 ～ 2 为 16 位可读 / 写寄存器。

可进行中断信号的发生、定时器值的清除 （仅支持 TCMR0）及发送请求的清除 （仅支持 TCMR2）。

TCMR0 与 TCNTR 进行比较，TCMR1 与 TCMR2 与 CYCTR 进行比较。各比较匹配寄存器的比较值可分别单独设定。将比较匹配的标志进行置位时，需要将 TTCR0 的 bit12 ～ 10 进行置位。时间触发模式下， TMCR0 用于 Init_Watch_Trigger、 TCMR2 用于 Watch_Trigger。

（1） 中断功能

通过设定 TTCR0 的 bit12、 bit11、 bit10，允许各寄存器的中断标志。

发生比较匹配时，将 IRR 对应的中断标志 （bit11、 bit15、 bit14）进行置位。

通过 IMR 的 bit11、 bit15、 bit14，可禁止中断信号的发生。

比较匹配发生时，如果 IRR11 （IRR15 或 IRR14）进行置位，则定时器状态寄存器 （TSR）的 bit3、 bit2 或 bit1 也均进行置位。另外，清除 IRR 的位时，对应的 TSR 的位也进行清除。

（2） 定时器清除功能

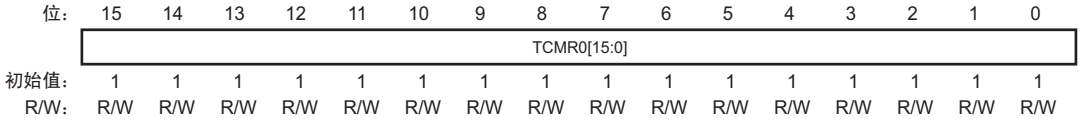
仅 TCMR0 具有通过比较匹配产生的定时器值（TCNTR）的清除功能。设定 TTCR0[6]=1 时有效。TCMR1 与 TCMR2 无此功能。

（3） 请求发送的信息的取消功能

TCMR2 设定为 TTCR0[13:12]=B'11 时，在停机模式以外的模式下，发生 TCMR2 的比较匹配时，进行对应于 TXPR 设定的 TXCR 设定。因此，除发送中的信息外，取消发送等候请求。

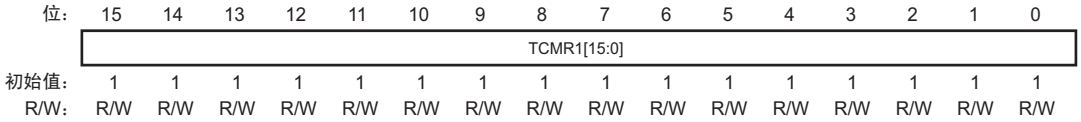
TCMR1 与 TCMR0 无此功能。

- TCMR0



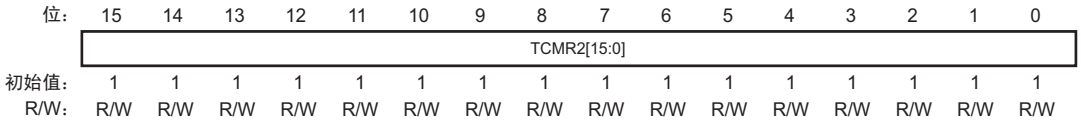
位	位名称	初始值	R/W	说明
15 ～ 0	TCMR0[15:0]	H'FFFF	R/W	定时器比较匹配寄存器 表示比较匹配发生时的 TCNTR 的值。

- TCMR1



位	位名称	初始值	R/W	说明
15 ～ 0	TCMR1[15:0]	H'FFFF	R/W	定时器比较匹配寄存器 表示比较匹配发生时的 CYCTR 的值。

- TCMR2



位	位名称	初始值	R/W	说明
15 ～ 0	TCMR2[15:0]	H'FFFF	R/W	定时器比较匹配寄存器 表示比较匹配发生时的 CYCTR 的值。

19.6.10　发送触发时间选择寄存器 （TTTSEL）

TTTSEL 为 16 位可读 / 写寄存器，指定周期时间与进行比较匹配的发送触发时间。仅 1 个位可进行置位。请勿将多个位进行置位。或者，请清除所有的位。此寄存器仅可在结构模式下进行变更。变更时的算法如图 19.14 所示。

TTTSEL 为测试及诊断专用。正常运行时请勿写入。另外，不保证读取值。

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

【注】 仅 1 位可进行置位。

位	位名称	初始值	R/W	说明
15	—	0	R	保留位 读写的值总是 0。
14 ～ 8	TTTSEL [14:8]	B'1000000	R/W	指定 CYCTR 与进行比较匹配的发送触发时间。 bit14 ～ 8 对应于邮箱 30 ～ 24。
7 ～ 0	—	均为 0	R	保留位 读写的值总是 0。 .

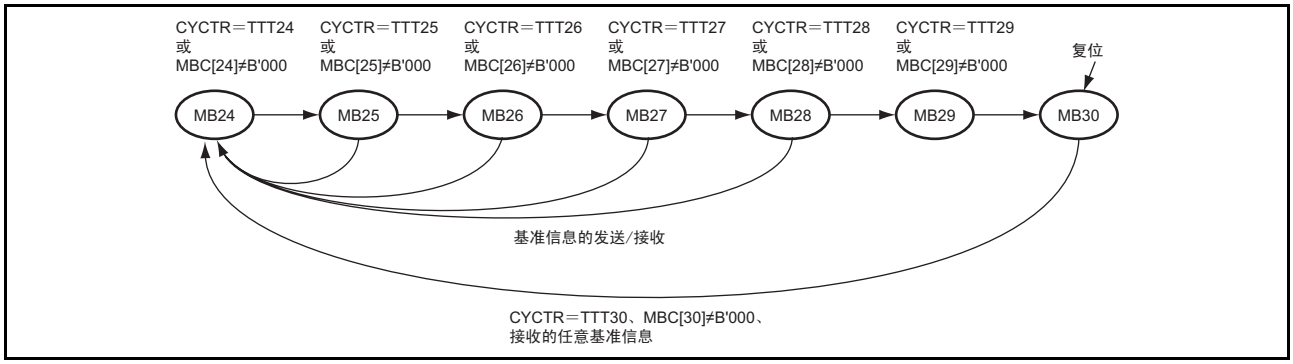


图 19.14　TTTSEL 变更算法

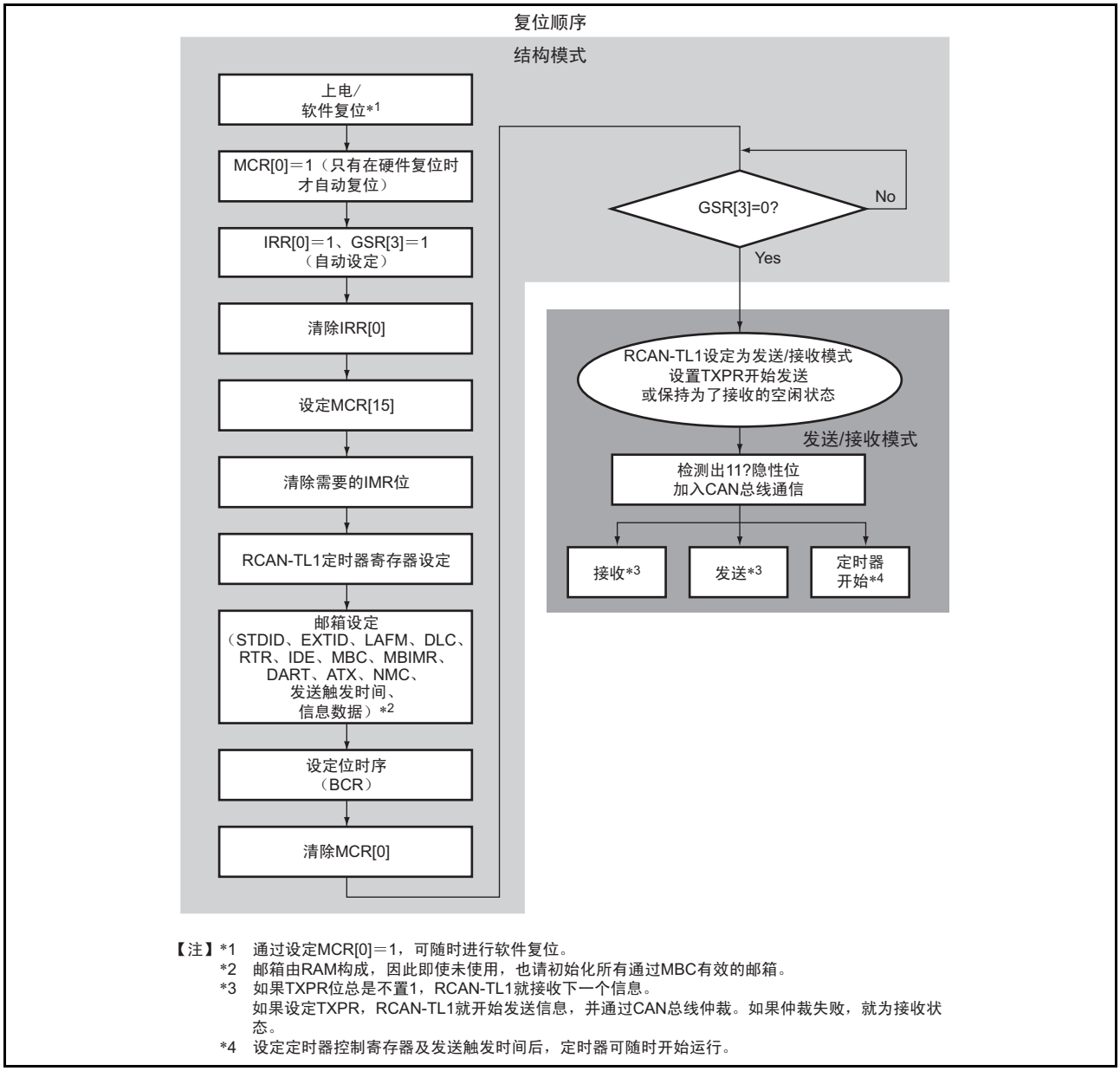
19.7　运行说明

19.7.1　RCAN-TL1 的设定

关于硬件复位（上电复位）或软件复位（MCR0）后的结构模式及停机模式下的 RCAN-TL1 设定，在此予以说明。无论在何种情况下，RCAN-TL1 均不可加入 CAN 总线激活。另外，RCAN-TL1 的设定变更不影响 CAN 总线的通信。

（1） 复位顺序

软件复位或硬件复位后的 RCAN-TL1 的设定步骤如图 19.15 所示。复位后，初始化所有寄存器。因此在加入 CAN 总线激活之前，必须设定 RCAN-TL1。详细内容请参考图中的注释。



【注】*1　通过设定MCR[0]＝1，可随时进行软件复位。
*2　邮箱由RAM构成，因此即使未使用，也请初始化所有通过MBC有效的邮箱。
*3　如果TXPR位总是不置1，RCAN-TL1就接收下一个信息。
如果设定TXPR，RCAN-TL1就开始发送信息，并通过CAN总线仲裁。如果仲裁失败，就为接收状态。
*4　设定定时器控制寄存器及发送触发时间后，定时器可随时开始运行。

图 19.15　复位顺序

（2） 停机模式

RCAN-TL1 为停机模式时，不可加入 CAN 总线激活。因此用户可变更对 CAN 总线通信无影响的所必需的寄存器的设定。在此，重要的是在变更寄存器之前，等候到 RCAN-TL1 成为停机模式。向停机模式的转移未必立刻进行（在 CAN 总线为空闲状态或睡眠状态时转移）。RCAN-TL1 转移至停机模式时，GSR4 位就进行置位。

寄存器设定结束后需解除停机请求。解除停机请求后 （MCR1=0 或 GSR4=0） RCAN-TL1 在 CAN 总线中检测出 11 个隐性位后加入 CAN 总线激活。

（3） CAN 睡眠模式

CAN 睡眠模式下，为了降低消耗电流，停止 RCAN-TL1 主要模块的时钟。但仅可存取 MCR、 GSR、 IRR 及 IMR 寄存器。而且在 CAN 睡眠模式下，不可清除发送与接收有关的中断，所以请预先进行清除。

RCAN-TL1 的 CAN 睡眠模式的流程图如图 19.16 所示。

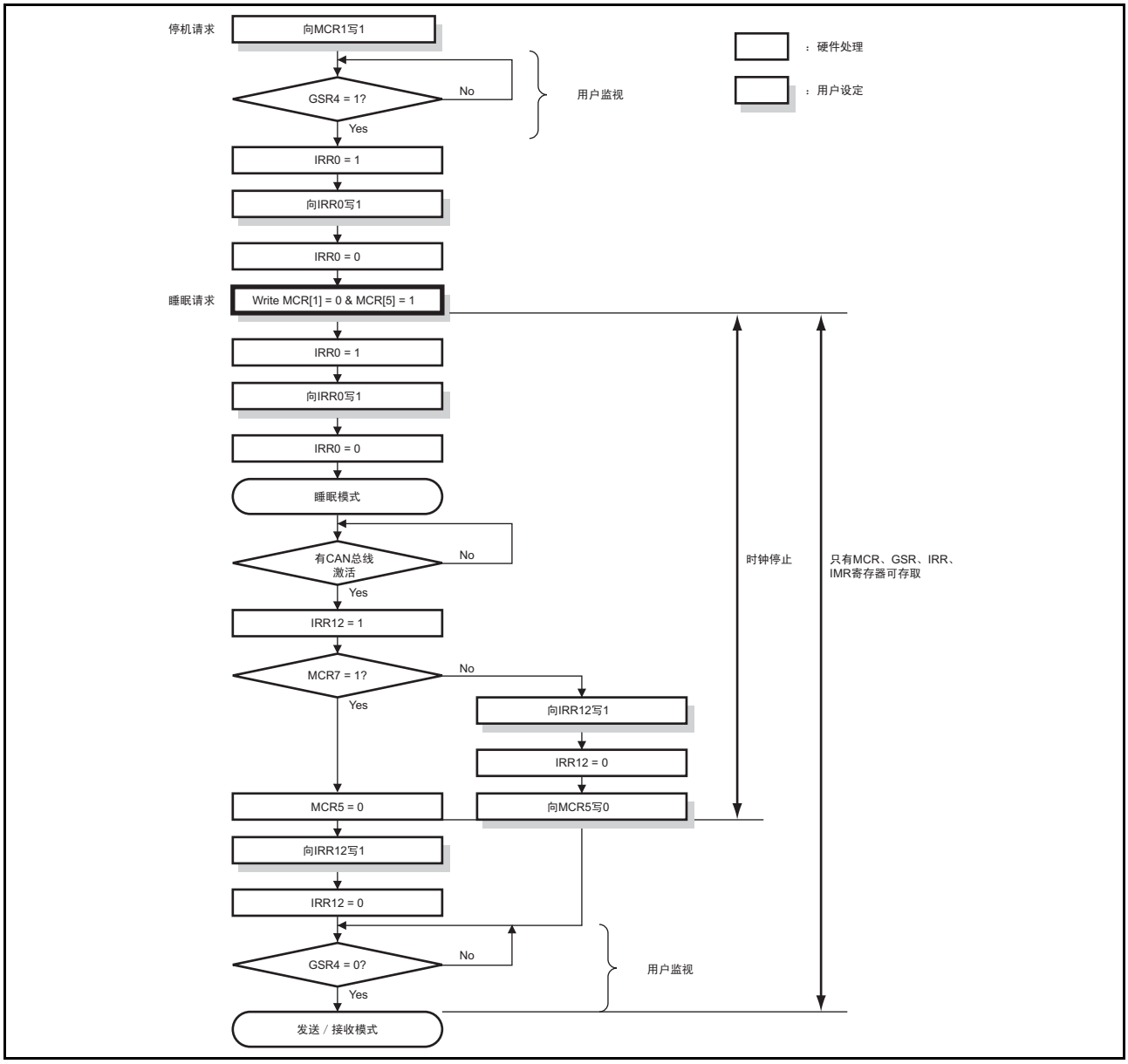
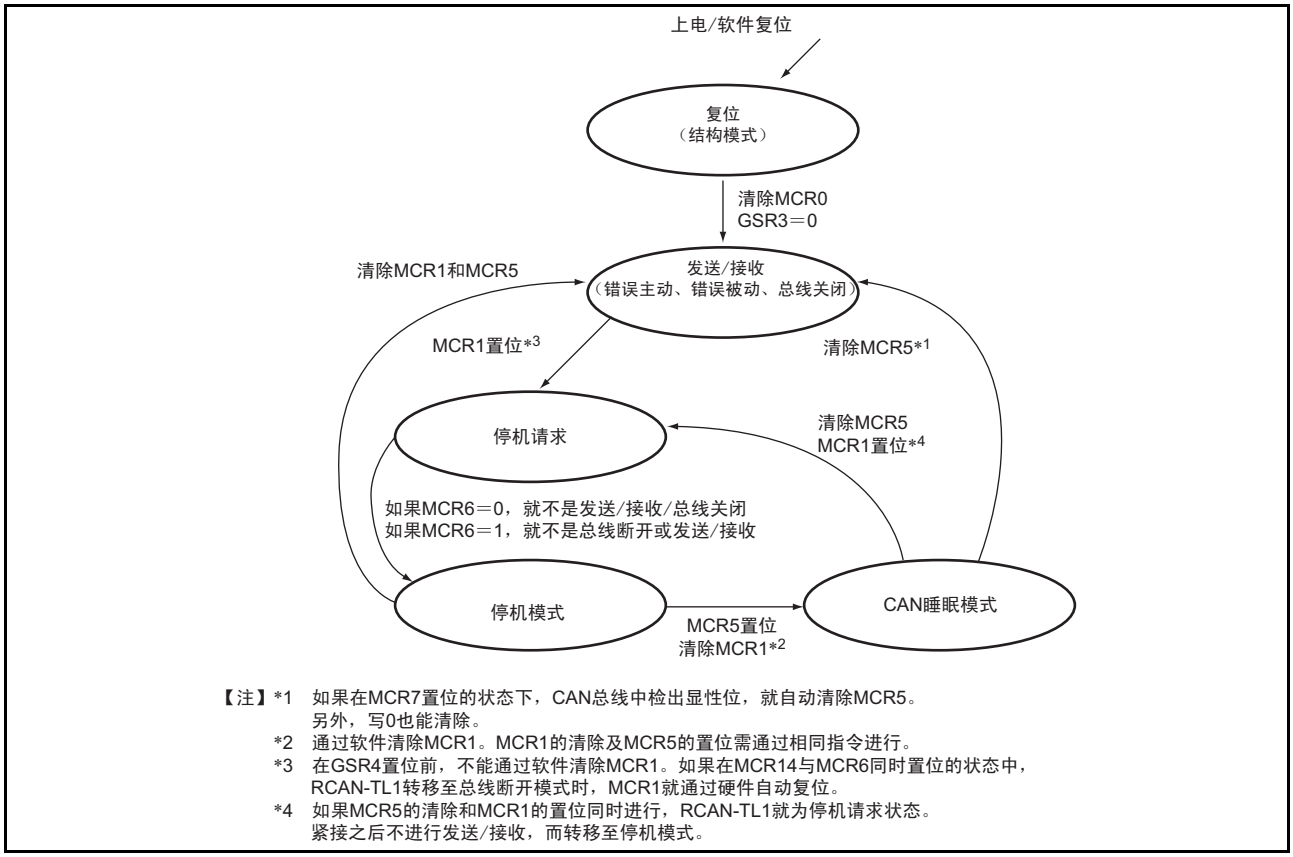


图 19.16　CAN 睡眠模式的流程图

可能的状态转移如图 19.17 所示。

在停机模式以外的模式下， MCR5 （CAN 睡眠模式）位请勿进行置位。

另外，将MCR1位进行置位后，清除MCR1之前，必须确认GSR4的置位，并将RCAN-TL1设定至停机模式。



【注】*1　如果在MCR7置位的状态下，CAN总线中检出显性位，就自动清除MCR5。
另外，写0也能清除。
*2　通过软件清除MCR1。MCR1的清除及MCR5的置位需通过相同指令进行。
*3　在GSR4置位前，不能通过软件清除MCR1。如果在MCR14与MCR6同时置位的状态中，RCAN-TL1转移至总线断开模式时，MCR1就通过硬件自动复位。
*4　如果MCR5的清除和MCR1的置位同时进行，RCAN-TL1就为停机请求状态。紧接之后不进行发送/接收，而转移至停机模式。

图 19.17　状态转移图

允许在各模式下的存取的条件如**表 19.9** 所示。

表 19.9　可存取寄存器

状态模式	寄存器									
	MCR、GSR	IRR、IMR	BCR	MBIMR 定时器 TT 寄存器	标志寄存器	邮箱（控制 0、LAFM）		邮箱（数据）	邮箱（控制 1）		邮箱（触发时间、TT 控制）
复位	Yes	Yes	Yes	Yes	Yes	Yes		Yes	Yes		Yes
发送 / 接收	Yes	Yes	No	Yes	Yes	No	Yes*	Yes*	No	Yes*	Yes*
停机请求	Yes	Yes	No	Yes	Yes	No	Yes*	Yes*	No	Yes*	Yes*
停机模式	Yes	Yes	No	Yes	Yes	Yes		Yes	Yes		Yes
CAN 睡眠模式	Yes	Yes	No	No	No	No		No	No		No

【符号说明】

Yes：允许存取的寄存器

No：禁止存取的寄存器

【注】 *　TXPR0 未进行置位时

19.7.2　测试模式的设定

RCAN-TL1 具有各种测试模式。通过 MCR 寄存器的 TST[2:0] 位选择测试模式。RCAN-TL1 在普通模式下，以默认 （初始值）方式进行运行。

测试模式的设定如**表 19.10** 所示。

仅可在结构模式下选择测试模式。执行所选择的测试模式时，请在选择测试模式后解除结构模式 （确认设定 BCR0/BCR1）。

表 19.10　测试模式的设定

TST2	TST1	TST0	模式
0	0	0	普通模式
0	0	1	listen-only 模式 （接收专用模式）
0	1	0	自测试模式 1 （外部）
0	1	1	自测试模式 2 （内部）
1	0	0	写错误计数器
1	0	1	错误被动模式
1	1	0	禁止设定
1	1	1	禁止设定

- 普通模式
  RCAN-TL1进行正常运行。
- listen-only模式 （接收专用模式）
  为ISO-11898中要求用于波特率检测的模式，。清除错误计数器后禁止运行，不增加TEC/REC的值。另外，禁止CTxn （n=0、 1）输出，禁止通过RCAN-TL1生成的错误帧或应答位。发生信息错误时，将IRR13进行置位。
- 自测试模式 （外部）
  以RCAN-TL1自身生成应答位，如果需要，则将信息保存至RCAN-TL1的接收邮箱。 CRxn/CTxn （n=0、 1）引脚必须连接至CAN总线。
- 自测试模式 （内部）
  以RCAN-TL1自身生成应答位，如果需要，则将信息保存至RCAN-TL1的接收邮箱。因内部CTxn （n=0、 1）循环反馈为内部CRxn （n=0、 1），所以无需将CRxn/CTxn （n=0、 1）引脚连接至CAN总线及其他的外部器件。 CTxn （n=0、 1）引脚仅输出隐性位， CRxn （n=0、 1）引脚无效。
- 写错误计数器
  TEC/REC可在此模式下进行写入。通过将大于127的值写入写错误计数器，可将RCAN-TL1强行设置为错误被动模式。因为将写入TEC中的值写入REC，所以TEC与REC常设定为相同值。同样，通过写入大于95的值，可将RCAN-TL1强行设置为错误警告模式。
  写入TEC/REC时， RCAN-TL1必须为停机模式 （错误计数器写入时MCR1=1）。因CAN规格，在解除停机模式之前，必须解除此测试模式，并可使TEC/REC重新运行。
- 错误被动模式
  RCAN-TL1可强行设置为错误被动模式。
  通过执行错误被动模式，虽有时不改变REC的值，但运行一次错误被动模式时，如果接收错误， REC的值正常增加。此模式下， TEC的值达到256时， RCAN-TL1成为总线断开状态，但使用此模式时，RCAN-TL1不可成为错误主动状态。因此， RCAN-TL1在总线断开返回顺序的末尾，转移至错误被动状态，而不是错误主动状态。

信息错误发生时，在所有的测试模式下，将 IRR13 位进行置位。

19.7.3　信息发送顺序

（1） 信息发送请求

在总线中发送 CAN 帧的顺序例如图 19.18 所示。

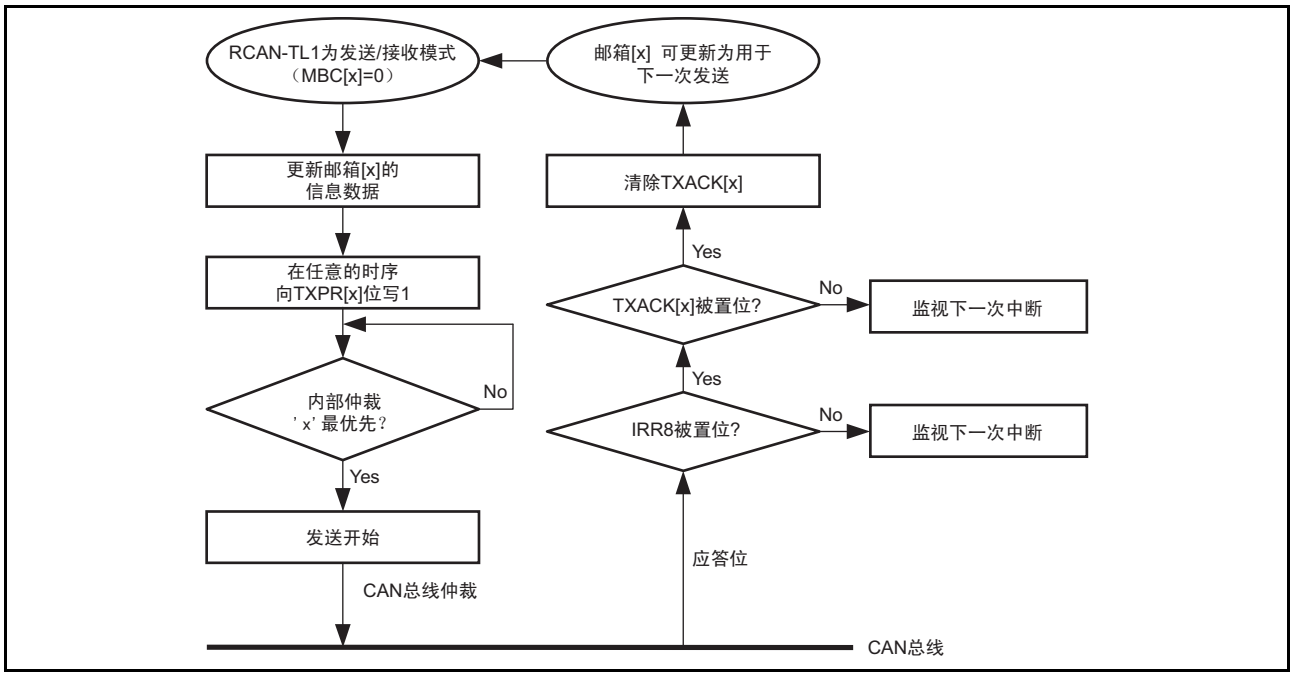


图 19.18　信息发送请求

请注意：如同寄存器章节中说明的那样， TXACK 或 ABACK 位任意一个位进行置位时， IRR8 就进行置位。这就意味着任意邮箱发送或发送中断结束，可更新为下一个发送的状态。另外， GSR2 表示未发生当前发送请求 （所有 TXPR 标志未进行置位）。

（2） 发送用内部仲裁

图 19.19 说明 RCAN-TL1 如何按照信息 ID 的顺序进行发送请求的信息调度。通过内部仲裁，在有发送请求的信息中选取优先级最高的信息。

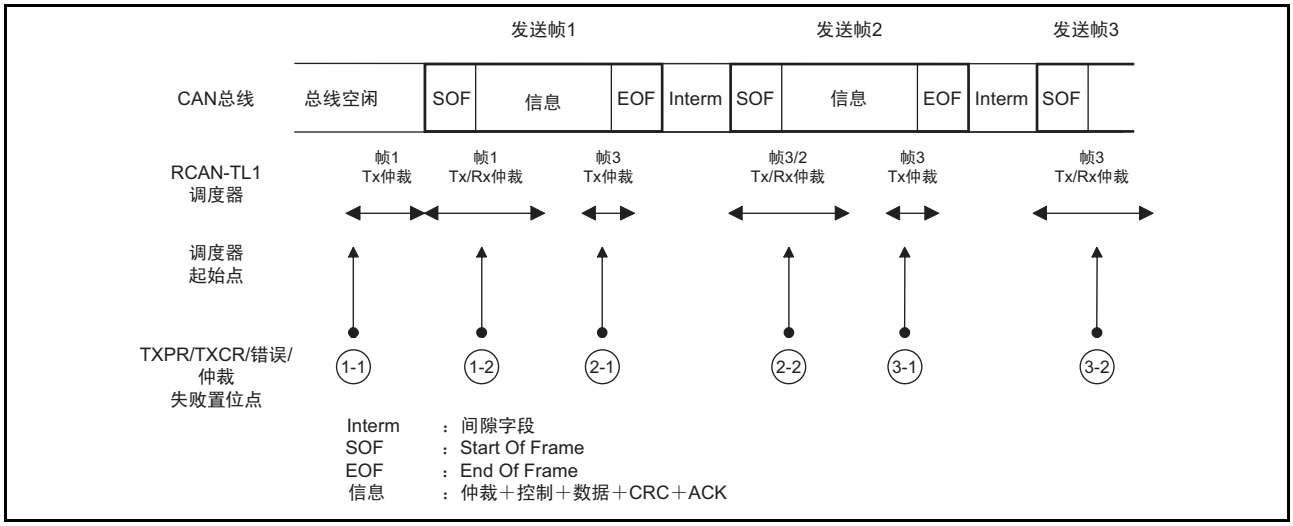


图 19.19　发送用内部仲裁

1-1：CAN 总线为空闲状态时，如果 TXPR 位进行置位，则立即进行内部仲裁，并开始发送。

1-2：在 SOF 开始发送与接收运行。因不是接收帧，所以 RCAN-TL1 进行发送。

2-1：通过 CRC 的定界符，开始进行内部仲裁，搜索下一个发送信息。

2-2：在 SOF 开始发送与接收运行。由于为高优先级的接收帧，所以 RCAN-TL1 进行接收。因此在不发送帧 3 的情况下进行接收。

3-1：通过 CRC 的定界符，开始进行仲裁，搜索下一个发送信息。

3-2：在 SOF 开始发送与接收运行。由于发送帧的优先级比接收帧高，所以 RCAN-TL1 进行发送。

CAN 总线中检测出错误时，在各错误定界符的起始点也可进行下一个发送用内部仲裁。另外，在超载帧之后的错误定界符的起始点也可进行内部仲裁。

通过 CRC 的定界符可进行发送仲裁，因此 ATX=1 的邮箱在接收远程帧请求时，根据此请求，发送的信息加入发送用仲裁的时序为此后的总线空闲、 CRC 定界符或错误定界符。

根据 CAN 总线的状态， TXCR 进行置位后，可进行对应于最大 1CAN 帧延迟之后的信息中断处理。

（3） 时间触发发送

RCAN-TL1 通过硬件支持符合 ISO-11898-4 TTCAN Level 1 规格的时间触发模式通信。

在此记述使用时间触发模式的基本步骤。

- **时间触发模式的设定**

时间触发模式的设置需要进行以下设定：

- 将CMAX_TEW的CMAX设定为B'111以外的值
- 将TTCR0的bit15进行置位，开始TCNTR的运行
- 为了防止清除TCNTR，通过比较匹配清除TTCR0的bit6
- 为了进行周期发送，不清除进行时间触发发送的邮箱的TXPR。

• 各寄存器的作用

使用 RCAN-TL1 的用户寄存器，可处理 TTCAN 规格所要求的主要功能。

TCNTR	Local_Time
RFMK	Ref_Mark
CYCTR	周期时间 =TCNTR-RFMK
RFTROFF	邮箱 30 的 Ref_Trigger_Offset
邮箱 31	时间基准信息专用接收邮箱
邮箱 30	时间基准信息专用发送邮箱 （作为潜在主时间运行时）
邮箱 29 ～ 24	时间触发发送对应的邮箱
邮箱 23 ～ 16	无时戳的接收对应的邮箱 （也可作为事件触发发送用邮箱来使用。）
邮箱 15 ～ 0	带时戳的接收对应的邮箱 （也可作为事件触发发送用邮箱来使用。）
Tx-Trigger Time	指定发送信息时序的 Time_Mark
CMAX	指定作为潜在主时间进行运行时的周期计数数 （Basic Cycle） 的最大值
TEW	指定 Tx_Enable_Window 宽度
TCMR0	Init_Watch_Trigger （本地时间与比较匹配）
TCMR1	用作用户指定事件监视的周期与比较匹配
TCMR2	Watch_Trigger （周期和比较匹配） 可设定为中断全部处于等待状态的发送。
TTW	指定用于发送的时间窗口的属性
TTTSEL	指定下一个发送等待邮箱

- 主时间/从时间

RCAN-TL1 可设定为作为网络的潜在主时间或从时间进行运行。各模式所需的设定与 RCAN-TL1 自动进行的处理如下表所示：

模式	设定	功能
从时间	TXPR[30]=0 & MBC[30] ≠ B'000 & CMAX ≠ B'111 & MBC[31]=B'011	每当在 CAN 总线中检测出 SOF，就采样 TCNTR，并保存至内部寄存器。邮箱 31 接收有效时间基准信息时，将 TCNTR 的值 （保存在 SOF）复制到 Ref_Mark （RFMK）。 将接收到的基准信息的周期计数器值 （CCR）复制到自节点的周期计数器寄存器 （CCR）。 如果 Next_is_Gap=1，则将 IRR13 进行置位。
（潜在）主时间	TXPR[30]=1 & MBC[30]=B'000 & DLC[30] ＞ 0 & CMAX ≠ B'111 & MBC[31]=B'011	有以下 2 种情况： （1）邮箱 31 接收到有效的时间基准信息时，在 SOF 将保存在内部寄存器的 TCNTR 的值复制到 Ref_Mark （RFMK）。将接收到的基准信息的周期计数器值 （CCR）复制到自节点的周期计数器寄存器 （CCR），如果 Next_is_Gap=1，则将 IRR13 进行置位。 （2）从邮箱 30 发送时间基准信息时，在 SOF 将保存在内部寄存器的 TCNTR 的值复制 Ref_Mark （RFMK）。邮箱 30 的 TTT 与 CYCTR 相同时，将周期计数器 （CCR）进行增量。周期计数器 （CCR）嵌入时间基准信息的起始数据字节 （Data0[7:6]、 CCR[5:0]）。

- 发送触发时间的设定

请按以下所示升序设定发送触发时间 （TTT）。 TTT 之间的差异需满足下列公式。公式中的 TEW 为寄存器的值。

TTT （邮箱 24）＜ TTT （邮箱 25）＜ TTT （邮箱 26）＜
TTT （邮箱 27）＜ TTT （邮箱 28）＜ TTT （邮箱 29）＜
TTT （邮箱 30）

而且

TTT （邮箱 i） -TTT （邮箱 i-1）＞ TEW+ 最大帧长度 +9

作为潜在主时间运行时， TTT （邮箱 24）～ TTT （邮箱 29）对应于 Time_Mark ； TTT （邮箱 30）对应于表示 Basic_Cycle length 的 Time_Ref。

上述规定仅适用于设定为时间触发发送的邮箱。

【重点】 根据发送触发时间设定的规定，可分配 1 个时间窗口的邮箱仅为 1 个。

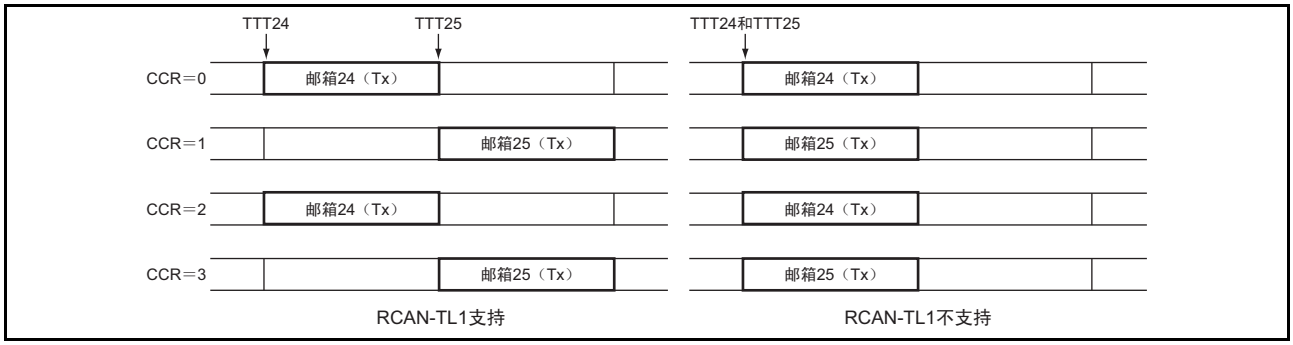


图 19.20　发送触发时间的规定

作为 Watch_Trigger 的 TCMR2 的值必须大于表示 1Basic_Cycle length 的 TTT （邮箱 30）。

（潜在）主时间及从时间的设定例如**图 19.21**、**图 19.22** 所示。图中的 L 为时间基准信息的时间长度。

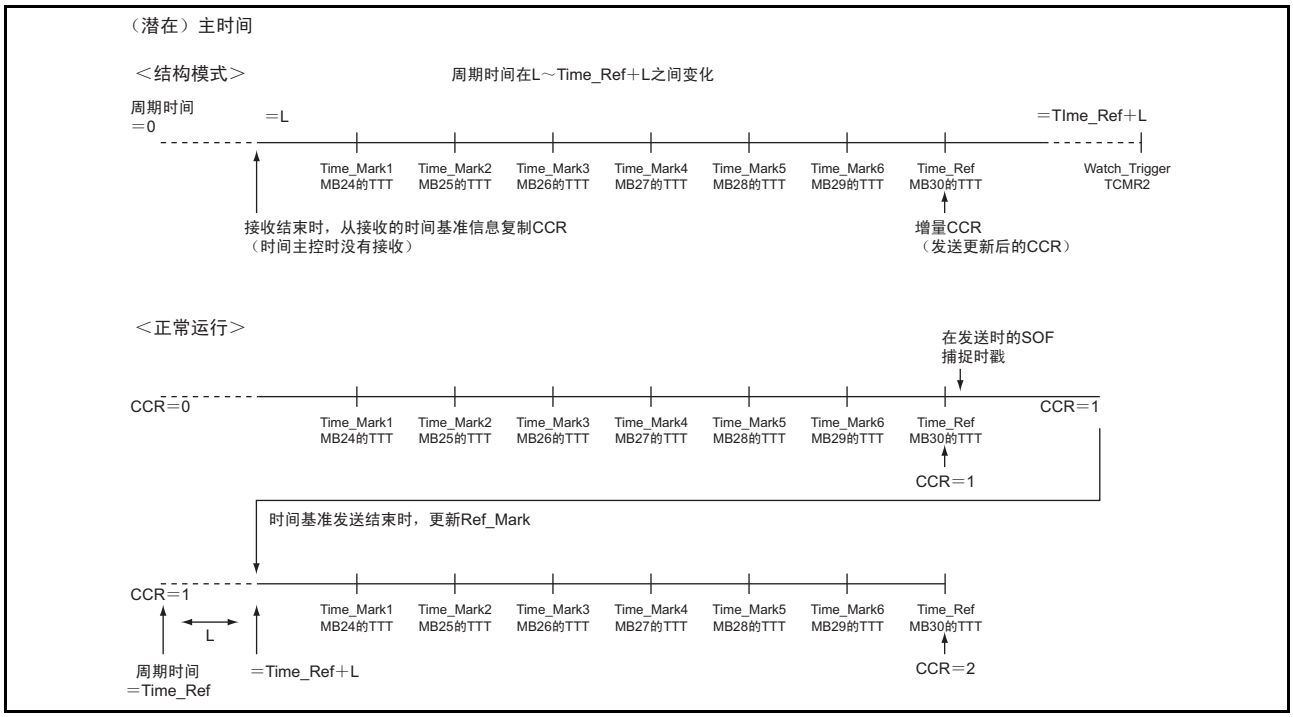


图 19.21　（潜在）主时间

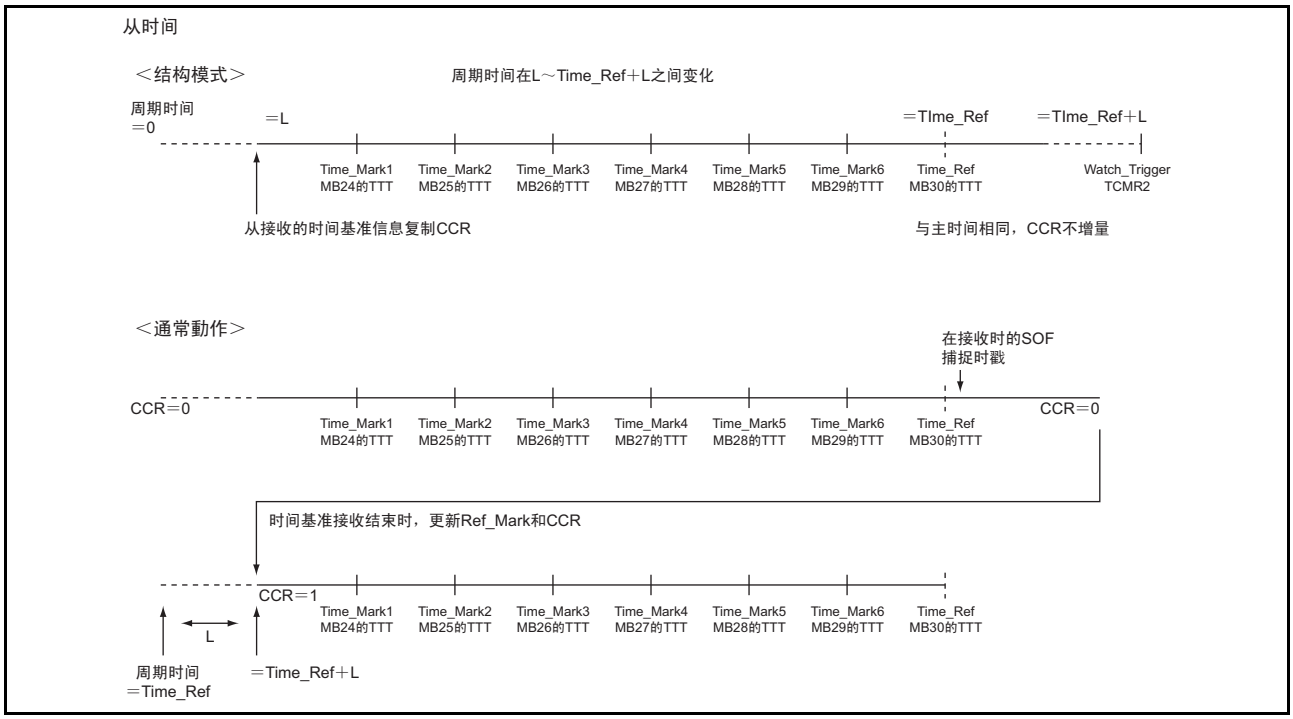


图 19.22　从时间

- 通过软件安装的功能

TTCAN 的功能中有需要通过软件安装的功能。主要功能如下所示。详细内容请参考 ISO-11898-4。

- 由Init_Watch_Trigger变更为Watch_Trigger

作为 Init_Watch_Trigger 及 Watch_Trigger 的硬件支持， RCAN-TL1 分别备有 TCMR0 寄存器与 TCMR2 寄存器。在 CAN 总线中检测出初始基准信息之前，将 TCMR0 设置为有效，而且将 TCMR2 设置为无效，此后，通过软件控制 TCMR0 无效而且 TCMR2 有效。（日程同步状态机）

仅支持 Next_is_Gap 接收中断。应用中，通过将对应的 TXCR 标志进行置位，当前的 Basic_Cycle 结束时，必须停止所有的发送。

主 / 从模式控制

仅支持自动周期时间同步与 CCR 的增量。

- 信息状态计数

必须通过软件对异或非窗口的定周期信息相关的日程错误进行计数。

- **时间触发通信的信息发送请求**

使用时间触发模式时，必须满足 ISO11898-4 的要求事项。

请进行以下步骤：

1. 将RCAN-TL1设置为复位或停机模式
2. 将Init_Watch_Trigger （H'FFFF）设定至TCMR0
3. 通过TTCR0的bit10将TCMR0产生的比较匹配设置为有效
4. 将指定的Watch_Trigger值设定至TCMR2
5. 将TTCR0的bit12保持为0，将通过TCMR2产生的比较匹配设置为无效。
6. 将所需的值设定至CMAX （B'111以外）
7. 将所需的值设定至TEW
8. 将邮箱设定为时间触发发送/接收
9. 对低3位设定邮箱31的LAFM
10. 将所需的值设定至MCR、 BCR1、 BCR0
11. 作为潜在主时间运行时设定如下：
    - 将所需的Init_Ref_Offset值设定至RFTROFF
    - 将邮箱30的TXPR进行置位
    - 将H'4000写入TTTSEL
12. 通过TTCR0的bit15允许定时器TCNTR
13. 转移至普通模式
14. 等待发送或接收有效基准信息，或者等待TCMR0的比较匹配
15. 如果本地时间 （TCNTR）与TCMR0的值相同，则表示到达Init_Watch_Trigger，并在应用中将邮箱30的TXCR进行置位，并重新开始
16. 如果发送基准信息 （将TXACK[30]置位），则将RFTROFF设定为0
17. 如果接收有效基准信息 （将RXPR[31]），则进行以下操作：
    - 如果邮箱31的ID的低3位的优先级比邮箱30的低3位高 （作为主时间电平运行时），则将RFTROFF的值保持为Init_Ref_Offset
    - 如果邮箱31的ID的低3位的优先级比邮箱30的低3位低 （作为主时间电平运行时），则将RFTROFF的值递减1
18. 清除TTCR0的bit10，将通过TCMR0产生的比较匹配设置为无效
19. 将TTCR0的bit12进行置位，将通过TCMR2产生的比较匹配设置为有效
20. 在CAN总线中检测出2个基准信息 （发送或接收）后，应用中可将其他时间触发邮箱的TXPR进行置位。

在 CAN 总线中未检测出基准信息，并且周期时间 CYCTR 达到 TCMR2 的值时， RCAN-TL1 自动中断全部等待状态的发送 （含基准信息）。

时间触发模式下，再次请求发送时的顺序如下所示：

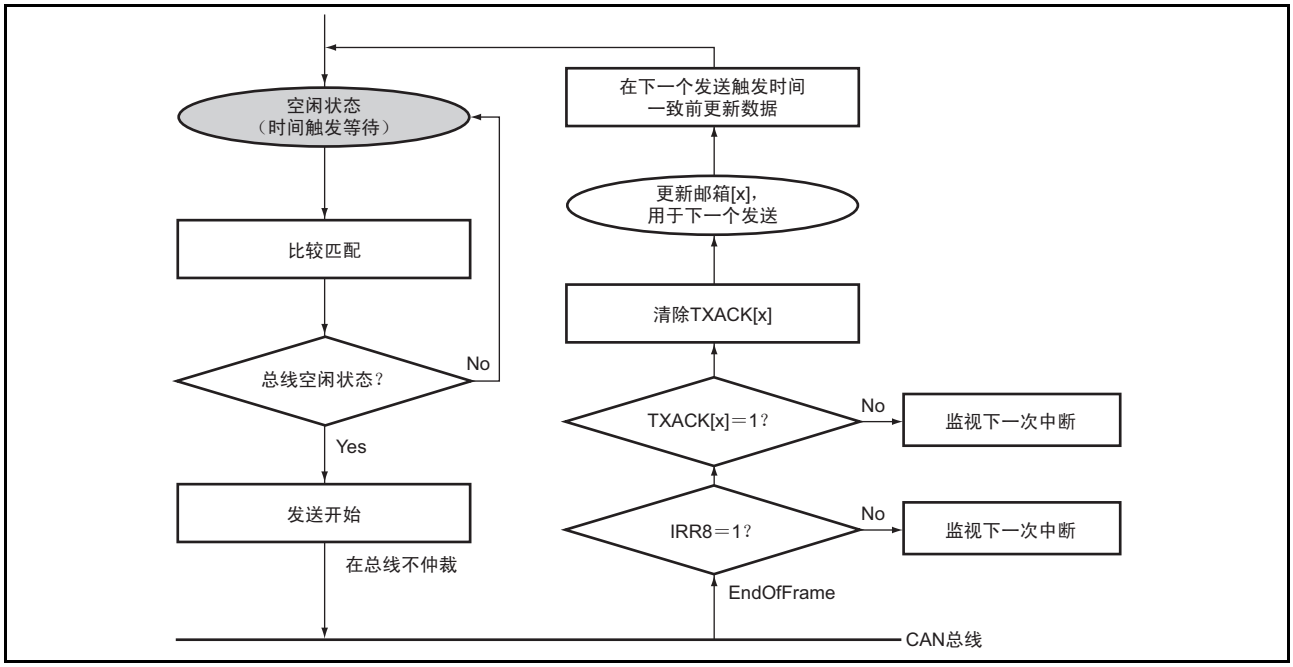


图 19.23　信息发送请求

发生发送触发之前，软件必须正确地进行信息更新。 CYCTR 达到邮箱的 TTT （发送触发时间）， CCR 与已编程的发送用周期相同时， RCAN-TL1 立即向发送缓冲器传送信息。

此时， RCAN-TL1 在指定的 Tx_Enable Window 中尝试发送。错过此时间槽，发送为定周期 （邮箱 24 ～ 30）时， RCAN-TL1 保持对应的 TXPR 位为 1，并在下一个发送触发之前保留发送请求。

RCAN-TL1 有 3 个错过上述时间槽的原因。

1. CAN 总线正在使用中
2. 发送时间触发信息时， CAN 总线中发生错误
3. 发送时间触发信息时，发生仲裁失败

合并仲裁窗口 （Merged Arbitrating Window）时，此发送槽从开始窗口的 （TTW=B'10）邮箱的发送触发开始，到结束窗口 （TTW=B'11）邮箱的 TEW 末尾。 TXPR 随时可进行变更。虽 RCAN-TL1 经常正确地确保按正常日程发送时间触发信息，但为了保证正确的进度，还规定了以下所示的重要规则：

- TTT （发送触发时间）在结构模式下可进行变更。
- 不可进行超过指定 Basic_Cycle length 周期计数器 （Basic Cycle）长度的 Time_Ref 的 TTT 的设定。违反时，调度中会发生问题。
- 定周期发送，不自动清除 TXPR。需要取消定周期发送时，应用中，将对应的 TXCR 位进行置位。

- **时间触发系统例**

使用从时间模式的 RCAN-TL1，运行时间触发系统的简例如下图所示。

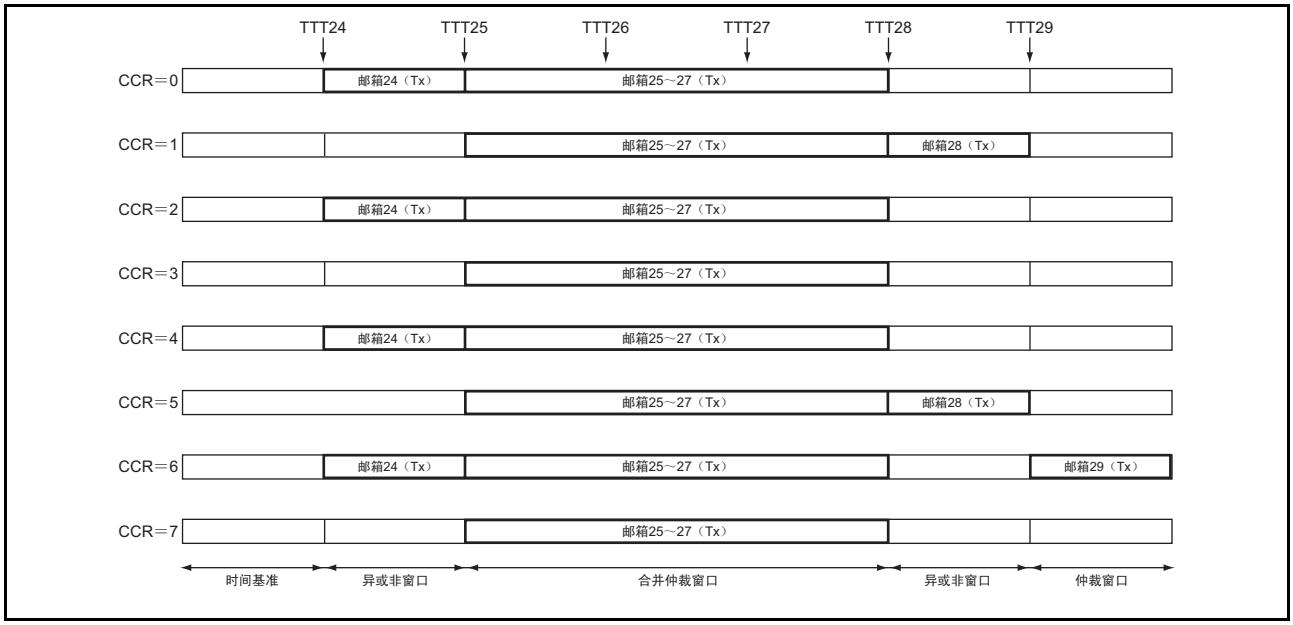


图 19.24　作为从时间的时间触发系统例

图 19.24 例中所使用的值如表 19.11 所示。

表 19.11　例中所使用的设定值

	rep_factor （寄存器）	Offset		TTW[1:0]	MBC[2:0]
邮箱 24	B'001	B'000000		B'00	B'000
邮箱 25	B'000	B'000000		B'10	B'000
邮箱 26	B'000	B'000000		B'10	B'000
邮箱 27	B'000	B'000000		B'11	B'000
邮箱 28	B'010	B'000001		B'00	B'000
邮箱 29	B'011	B'000110		B'01	B'000
邮箱 30	—	—		—	B'111
邮箱 31	—	—		—	B'011

【注】 CMAX=B'011,TXPR[30]=0

合并仲裁窗口中，时间触发发送请求通过 FCFS （First Come First Served）进行处理。例如：邮箱 25 不可在发送触发时间 25（TTT25）与发送触发时间 26（TTT26）之间进行发送时，在 TTT26-TTT28 之间，邮箱 25 优先级高于邮箱 26。

将时间触发发送设置为无效时，将MBC设定为B'111。RCAN-TL1为主时间时，必须设定为MBC[30]=B'000，时间基准窗口自动识别为仲裁窗口。

- 定时器运行

定时器的时序图如图 19.25 所示。设定发送触发时间 =n 时，时间触发发送在 CYCTR=n+2 到 CYCTR=n+3 之间开始。

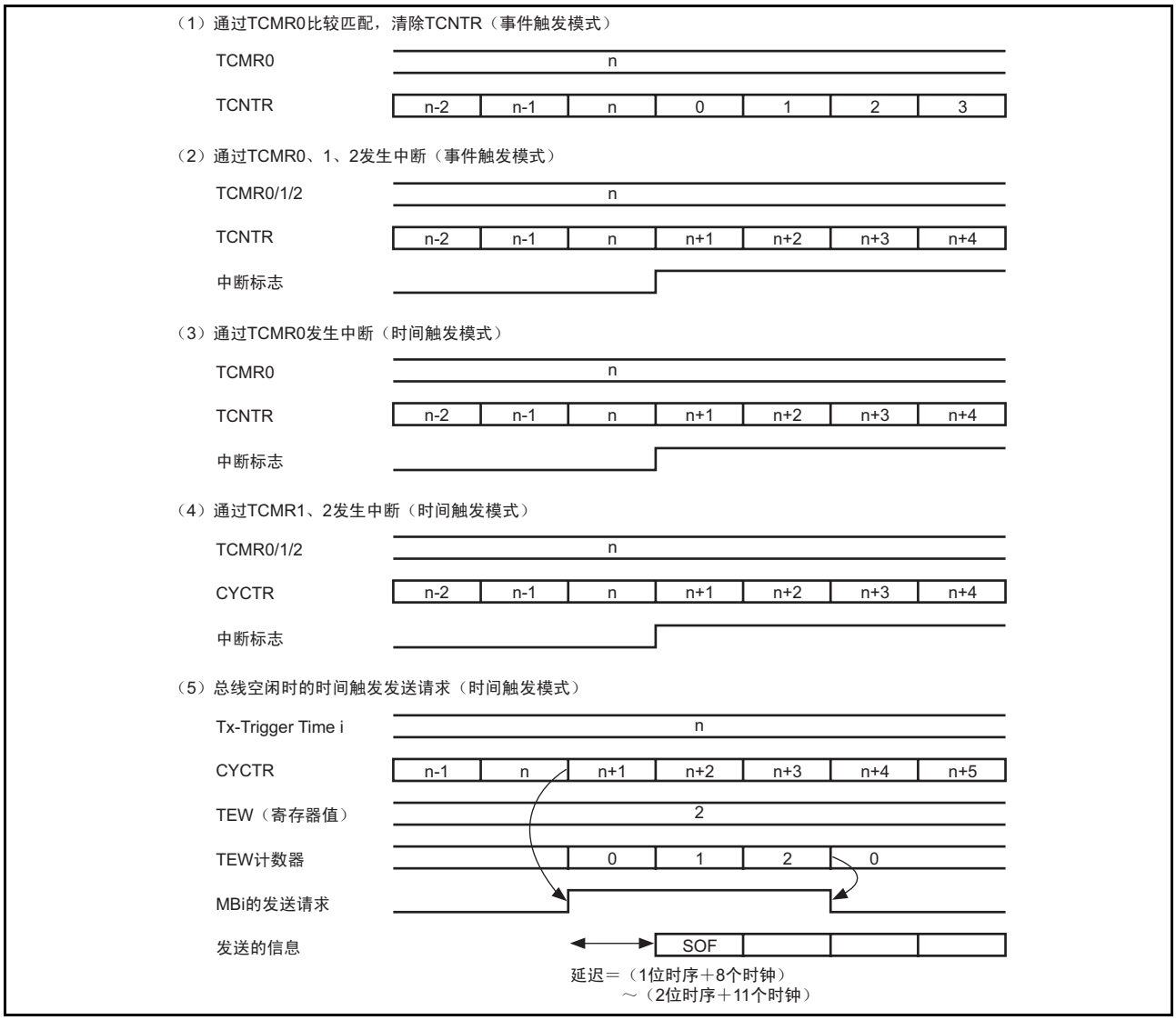


图 19.25　定时器的时序图

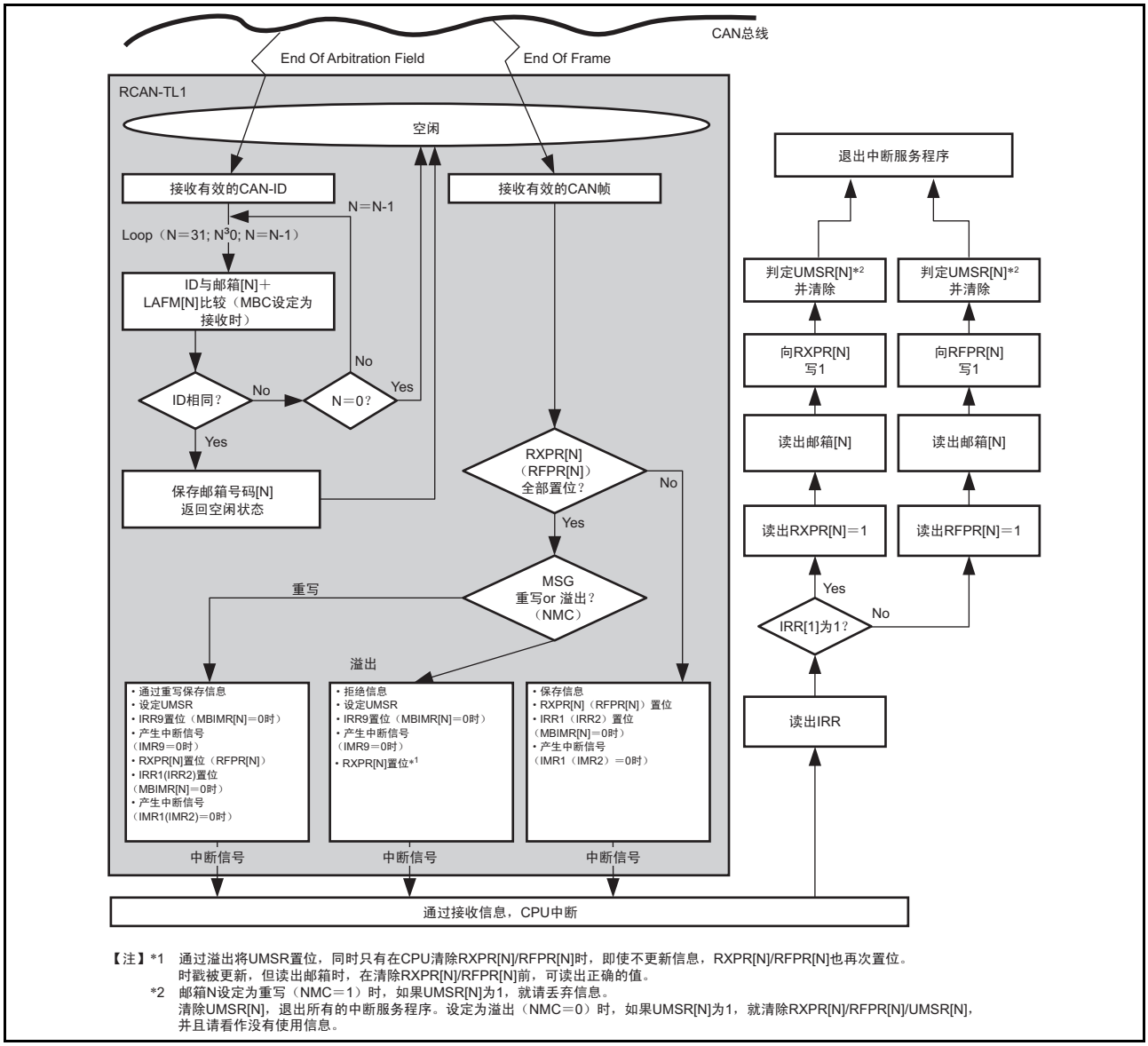
合并仲裁窗口中，事件触发发送在时间触发发送完成后进行处理。例如：邮箱 25 的发送完成时，如果 CYCTR 未到达 TTT26，则通过 MCR2 所指定的信息发送的优先级，开始进行事件触发发送。发送完成后，不清除时间触发发送的 TXPR，但清除事件触发发送的 TXPR。

关闭合并仲裁窗口的邮箱的 TXPR 未进行置位时，在此邮箱的 TTT 之后的 TEW 的末尾，关闭合并仲裁窗口。

请参考 “19.3.1 邮箱的构成”的表 19.3。

19.7.4　信息接收顺序

信息接收顺序如图 19.26 所示。



【注】*1　通过溢出将UMSR置位，同时只有在CPU清除RXPR[N]/RFPR[N]时，即使不更新信息，RXPR[N]/RFPR[N]也再次置位。
时戳被更新，但读出邮箱时，在清除RXPR[N]/RFPR[N]前，可读出正确的值。
*2　邮箱N设定为重写（NMC=1）时，如果UMSR[N]为1，就请丢弃信息。
清除UMSR[N]，退出所有的中断服务程序。设定为溢出（NMC=0）时，如果UMSR[N]为1，就清除RXPR[N]/RFPR[N]/UMSR[N]，并且请看作没有使用信息。

图 19.26　信息接收顺序

接收信息中，RCAN-TL1 识别仲裁字段的末尾时，开始进行已接收的 ID 与邮箱所设定的 ID 的比较。比较顺序为从邮箱 31 到邮箱 0 的顺序。首先，检查 MBC，并检查邮箱是否设定为接收。此后读取 LAFM，接着读取邮箱 31 （设定为接收用时）的 ID，并与已接收的 ID 进行比较。不相同时，对邮箱 30 （设定为接收时）进行同样的检查。发现相同 ID 时，RCAN-TL1 将此邮箱号码（N）保存至内部缓冲器，停止搜索后，返回空闲状态，等候 EOF （End Of Frame）。通知 EOF 的第 6 位时，通过 NMC 位的设定，废除或写入接收信息。

通信中不可变更 RCAN-TL1 的信息 ID 与 LAFM 的设定。作为改变设定的 1 种办法，具有停机模式及结构模式。将接收信息写入对应的邮箱时，由于写入含信息 ID，因此使用 LAFM 时，有时 CAN-ID 通过不同的 CAN-ID 进行重写。这就意味着在已接收信息的 ID 与多个邮箱的 ID+LAFM 相同时，常常将接收信息保存在邮箱号码最大的邮箱中，号码小的邮箱不保存信息。因此，需注意仔细选择 ID 与 LAFM 的设定值。

在如图 19.11 所示的接收数据及远程帧时，读取 IRR 后清除 UMSR 标志是由于在执行中断服务程序中，检测出信息通过保存在相同邮箱的新信息进行重写 （NMC 为 1 时）。在 UMSR 的最后检测中，检测出重写时，必需取消信息，并再次进行读取。

邮箱设定为溢出 （NMC=0）时， UMSR 进行置位时的信息有效。但 CAN 总线中监控的不是最新信息，而是旧信息。请在清除相关的 RXPR/RFPR 标志之前读取信息。

通过数据帧重写接收的远程帧时，将远程帧接收中断 （IRR2）及数据帧接收中断 （IRR1）进行置位，请注意：接收标志 （RXPR、 RFPR）也进行置位。同样，通过远程帧重写数据帧时， IRR2 与 IRR1 也进行置位。

接收信息并保存至邮箱时，将未接收的数据字段全部作为 0 进行保存。接收标准 ID 时也进行相同操作。扩展 ID （EXTID[17:0]）可写入 0。

19.7.5　邮箱的再设定

需要再次设定邮箱时，请按下述步骤进行：

（1） 变更发送箱的设定

有以下 2 种情况：

- ID、 RTR、 IDE、 LAFM、数据、 DLC、 NMC、 ATX、 DART 的变更
  仅在 MBC=B'000 时可进行变更。请确认对应的 TXPR 位未进行置位。可随时变更设定 （除 MBC 外）。
- 从发送箱到接收箱的设定变更
  请确认对应的 TXPR 位未进行置位。仅可在停机模式或复位状态下进行变更。RCAN-TL1 在接收或发送信息时，有时转移至停机状态需要时间 （是由于需等候接收 / 发送结束之后，才转移至停机状态）。另外，请注意：在停机状态下不可进行信息的接收 / 发送。
  RCAN-TL1 为总线断开状态时，向停机状态的转移根据 MCR 寄存器的 bit6 及 bit14 的设定来进行。

（2） 接收箱的 ID、RTR、IDE、LAFM、数据、DLC、NMC、ATX、DART、MBC 的设定变更及从接收箱向发送箱的变更

仅可在停机模式下进行设定变更。信息存在于 CAN 总线中，并且 RCAN-TL1 为接收模式时，不会丢失此信息。RCAN-TL1 在完成当前进行的接收之后转移至停机模式。RCAN-TL1 在接收或发送信息时，有时转移至停机状态需要时间 （是由于需等候接收 / 发送结束之后，才转移至停机状态）。另外，请注意：在停机状态下不可进行信息的接收 / 发送。

RCAN-TL1 为总线断开状态时，向停机状态的转移根据 MCR 寄存器的 bit6 及 bit14 的设定来进行。

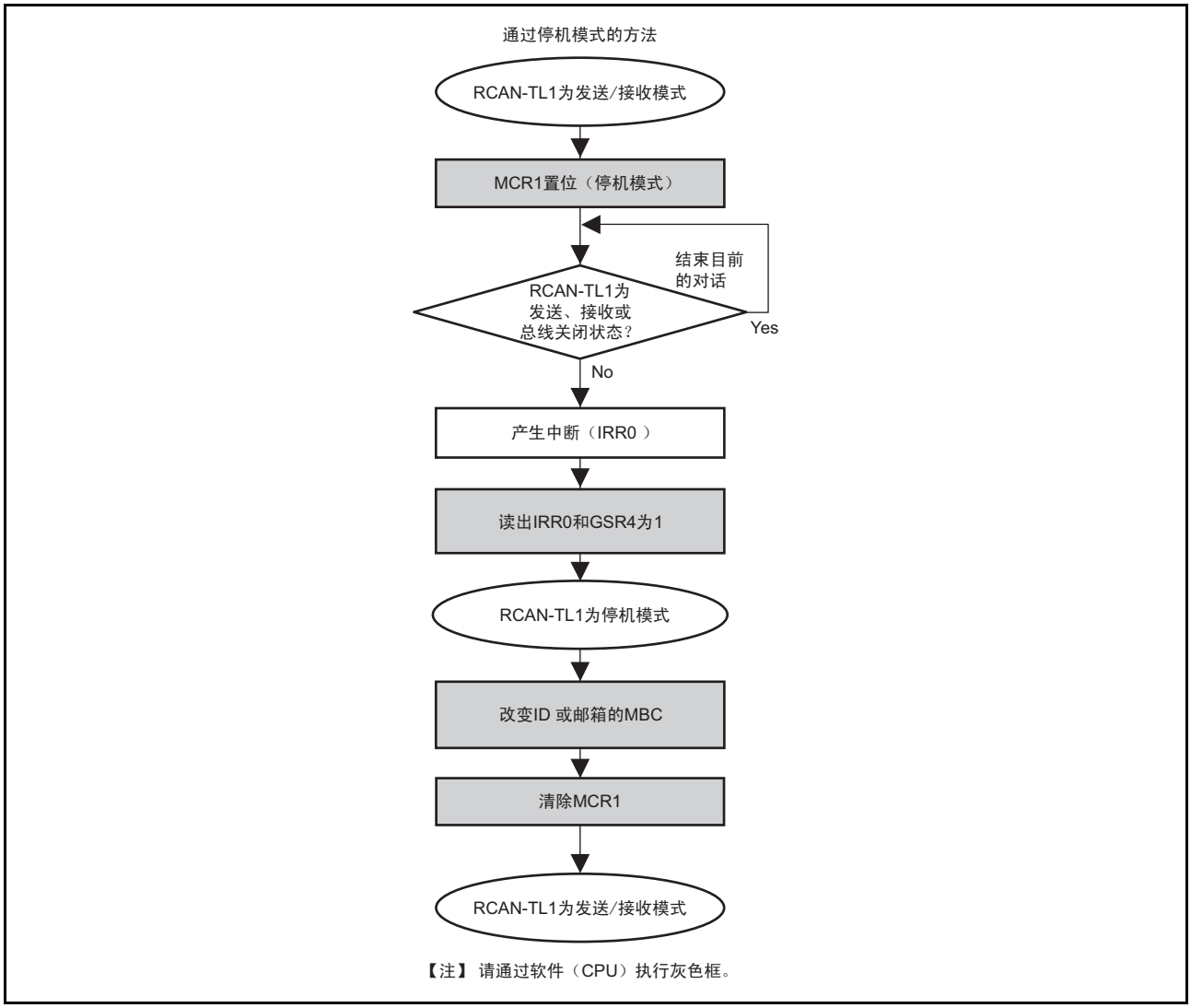


图 19.27　接收箱的 ID 变更 / 从接收箱向发送箱的变更

19.8　中断源

RCAN-TL1 中有表 19.12 所示的中断源。可屏蔽这些中断源。屏蔽时，使用邮箱中断屏蔽寄存器（MBIMR）及中断屏蔽寄存器 （IMR）。关于各中断请求的中断向量请参考 “第 6 章 中断控制器 （INTC）”。

表 19.12　RCAN-TL1 的中断源

模块名称	名称	中断源	中断标志	DMAC 的启动
RCANn*1	ERSn*1	错误被动 （TEC ≥ 128 或 REC ≥ 128）	IRR5	不可
		总线断开 （TEC ≥ 256） / 从总线断开的返回	IRR6	
		错误警告 （TEC ≥ 96）	IRR3	
		错误警告 （REC ≥ 96）	IRR4	
	OVRn*1	复位 / 停机 /CAN 睡眠转移	IRR0	
		超载帧发送	IRR7	
		未读取信息的重写 （溢出）	IRR9	
		起始系统矩阵	IRR10	
		TCMR2 比较匹配	IRR11	
		CAN 睡眠中 CAN 总线运行的检测	IRR12	
		定时器溢出 /Next_is_Gap/ 信息错误	IRR13	
		TCMR0 比较匹配	IRR14	
		TCMR1 比较匹配	IRR15	
	RMn0*1*2 RMn1*1*2	数据帧接收	IRR1*3	可 *4
		远程帧接收	IRR2*3	
	SLEn*1	信息的发送 / 取消发送 （槽空）	IRR8	不可

【注】 *1　n=0、 1

*2　RM0 为通过邮箱 0 的远程帧接收标志 （RFPR0[0]）或数据帧接收标志 （RXPR0[0]）产生的中断； RM1 为通过邮箱 n （n=1 ～ 31）的远程帧接收标志 （RFPR0[n]）或数据帧接收标志 （RXPR0[n]）产生的中断。

*3　IRR1 为邮箱 0 ～ 31 的数据帧接收标志； IRR2 为邮箱 0 ～ 31 的远程帧接收标志。

*4　仅可通过 RMn0 中断启动 DMAC。

19.9　DMAC 接口

将信息接收至各 RCAN-TL1 的邮箱 0 时，可启动 DMAC。而且，设定 DMAC 启动，结束通过 DMAC 进行的传送时，自动清除 RXPR0 与 RFPR0 寄存器的标志。此时，为来自 RCAN-TL1 的接收中断，不产生向 CPU 的中断。 DMAC 的传送流程图如图 19.28 所示。

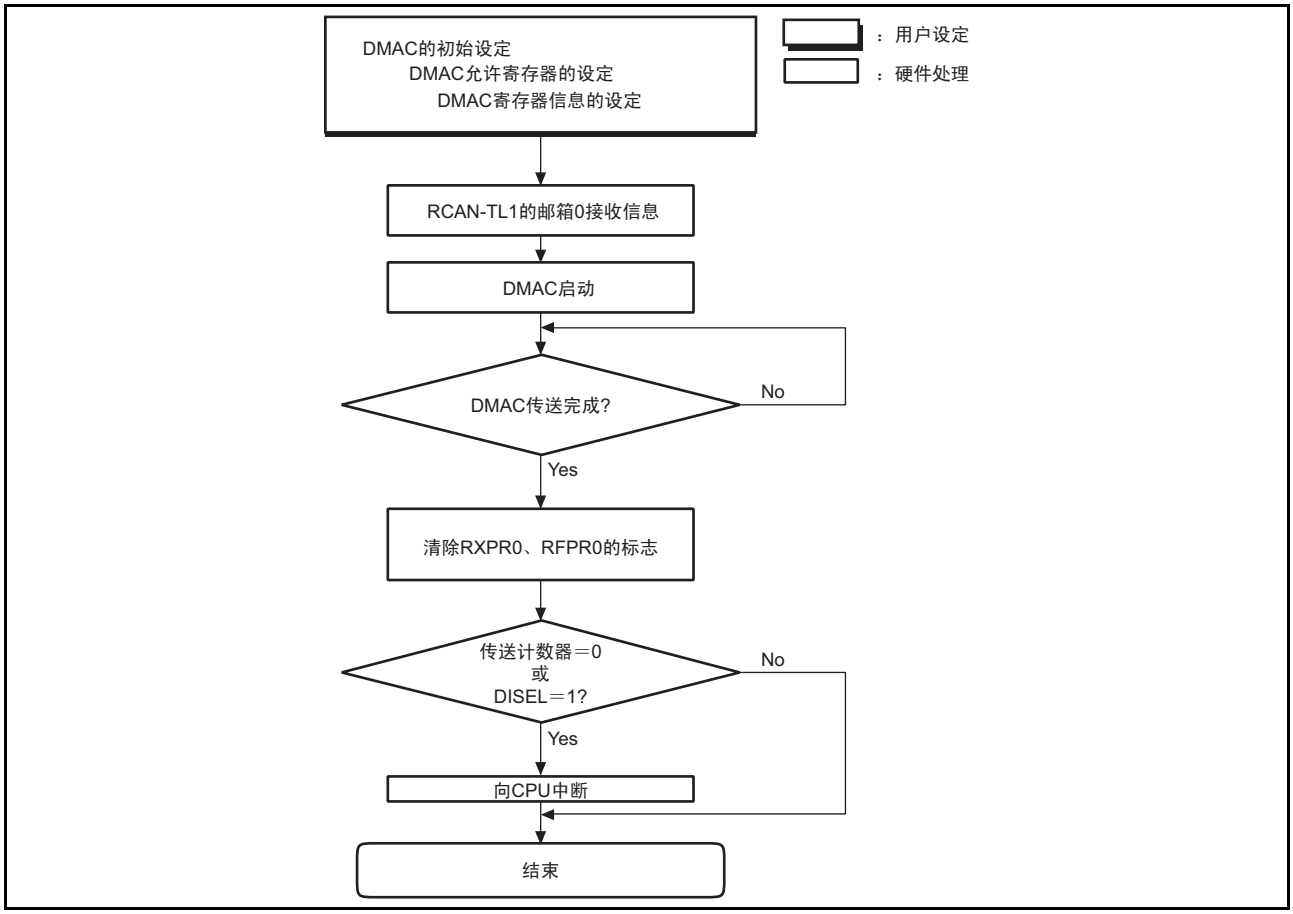


图 19.28　DMAC 的传送流程图

19.10　CAN 总线接口

将本 LSI 与 CAN 总线进行连接时，需要总线收发器 IC。收发器 IC 推荐使用瑞萨 HA13721。使用 HA13721 以外的产品时，请使用与 HA13721 兼容的产品。连接例如图 19.29 所示。

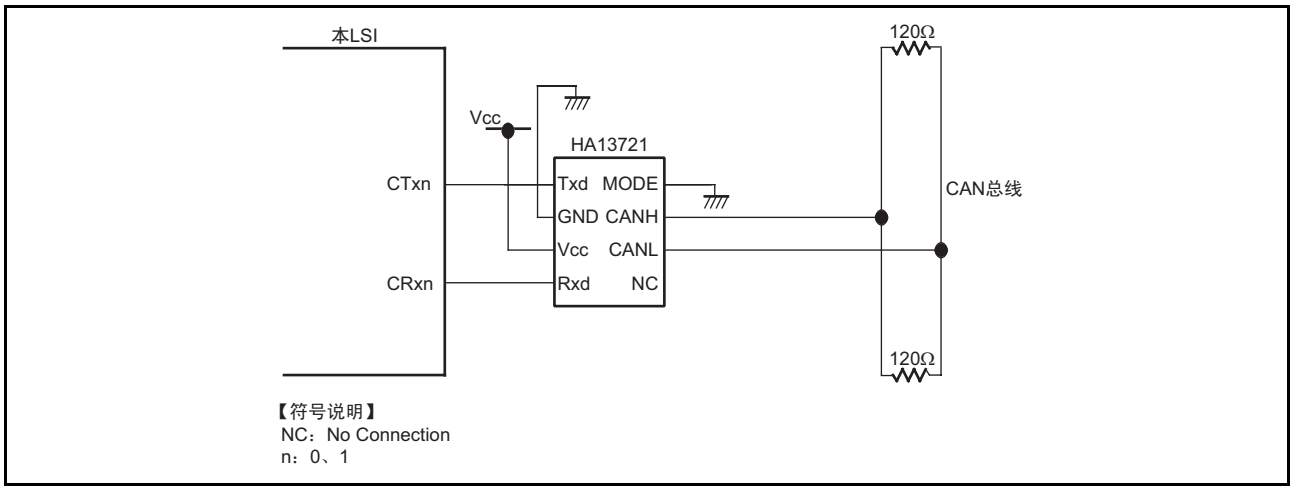


图 19.29　使用 HA13721 的高速 CAN 接口

19.11　RCAN-TL1 引脚端口设定

RCAN-TL1 端口设定必须在结构模式下或在此以前进行。端口设定方法的详细内容请参考“第 29 章　引脚功能控制器 （PFC）”。本 LSI 内置 3 个通道的 RCAN-TL1，有 2 种使用方法。

- 2个通道的32个邮箱 （RCAN0、 1）
- 1个通道的64个邮箱 （RCAN0+1）

【注】 使用 64 个邮箱时必需注意。必须阅读 “19.12.1 关于 1 个通道 64 个邮箱的端口设定注意事项”。

各端口设定的连接例如图 19.30、图 19.31 所示。

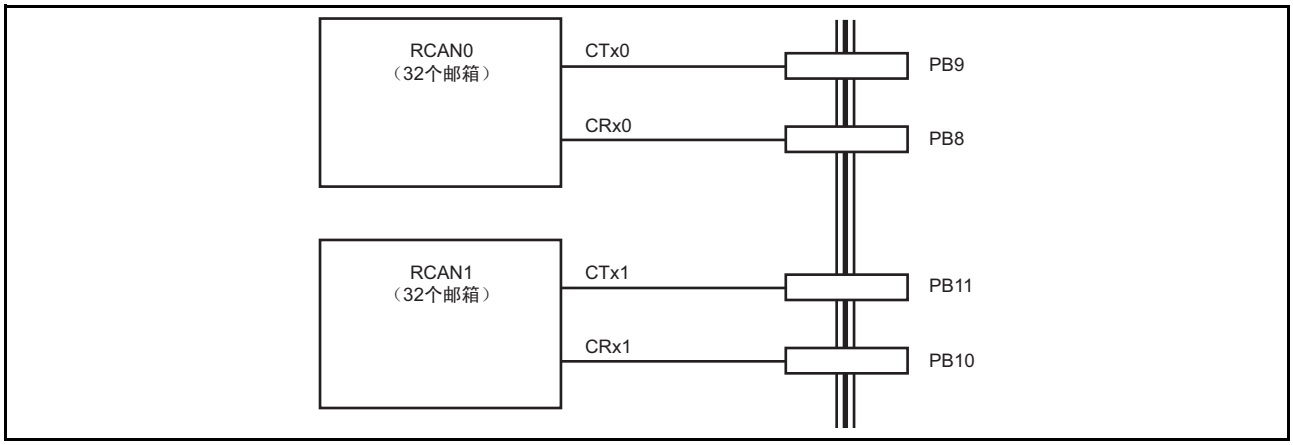


图 19.30　将 RCAN0、 1 作为个别通道来使用时的连接例

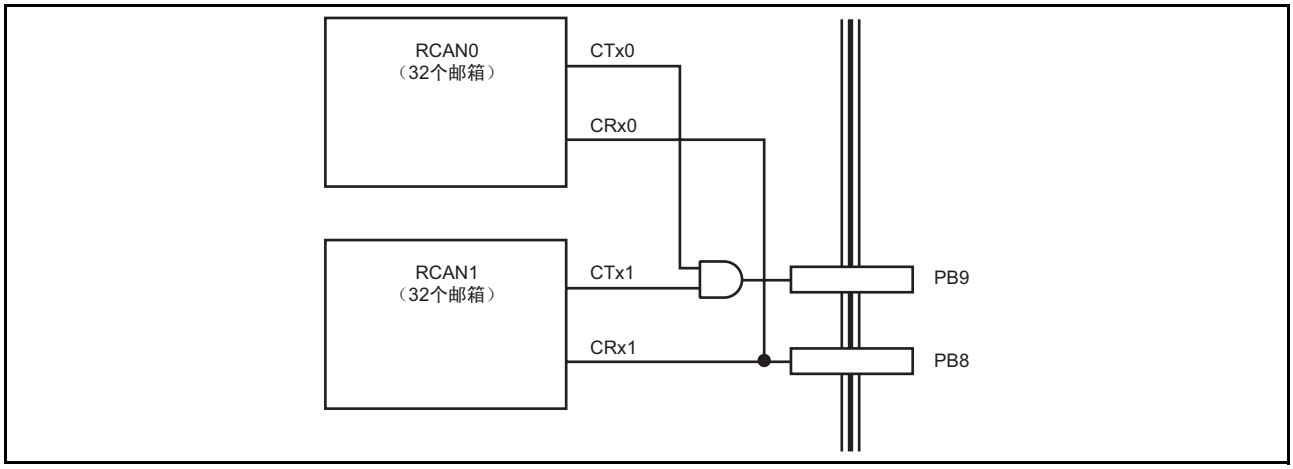


图 19.31　将 RCAN0、 1 作为 64 个邮箱的 1 个通道来使用时的连接例

19.12　使用时的注意事项

19.12.1　　关于 1 个通道 64 个邮箱的端口设定注意事项

本 LSI 内置 2 个通道的 RCAN-TL1。在 1 个通道中，通过 64 个邮箱的设定进行使用时，有以下注意事项：

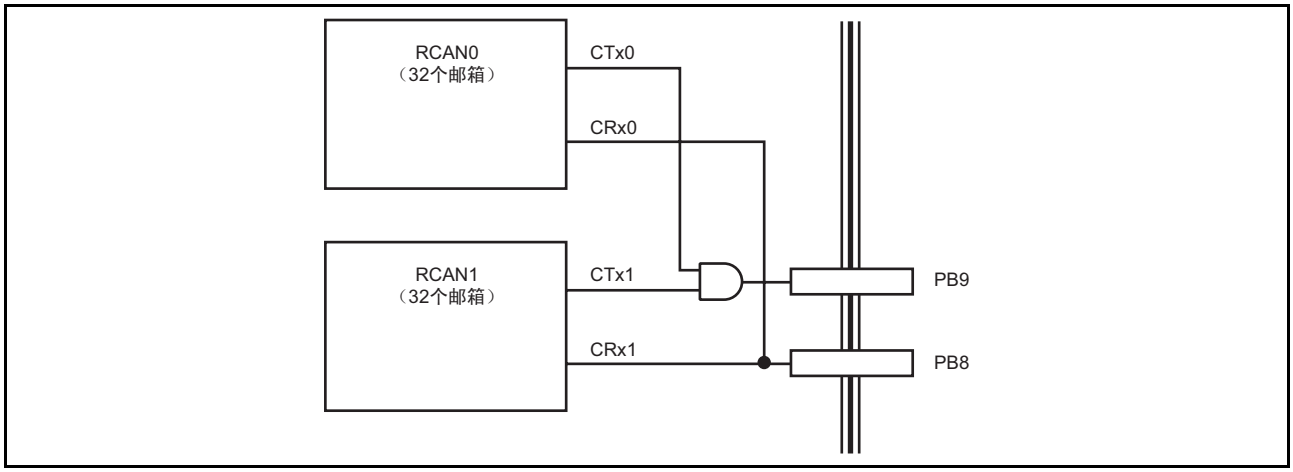


图 19.32　将 RCAN0、 1 作为 64 个邮箱的 1 个通道来使用时的连接例

1. 在CAN总线不连接其他节点的状态下发送信息时，不发生ACK错误。这是由于从上图的RCAN0发送信息时，在ACK字段RCAN1发送ACK。
   RCAN1接收CAN总线中的信息，根据CAN协议，在ACK字段发送ACK， RCAN0接收此ACK。
   请采用对应的方法将不发送信息的RCAN1设置为测试模式的接收专用模式状态 （TST[2:0]=B'001）或复位状态 （设定为MCR0=1）。因此不从不发送信息的通道发送ACK。
2. 决定发送顺序的内部仲裁分别通过RCAN0与RCAN1单独进行。可发送的缓冲器，在RCAN-TL1中有31个邮箱/通道，但在62个缓冲器的范围内不进行内部仲裁。
3. 请勿将相同发送信息ID设定至RCAN0与RCAN1。在CAN总线中进行仲裁后，从2个通道发送信息。

第 20 章　IEBus™ 控制器 （IEB）

本 LSI 内置 1 个通道的 IEBus 控制器（IEB）。IEBus™（Inter Equipment Bus™）* 是以在器件之间传送数据为目的的小规模数字化数据传送系统。

本 LSI 没有内置 IEBus 驱动器 / 接收器，因此需要外接专用的驱动器 / 接收器。另外，IERxD、IETxD 引脚是 3V 的规格，因此需要外接专用的电平移位器。

【注】 * IEBus™ （Inter Equipment Bus™）是 NEC 电子的商标。

20.1　特点

- 支持IEBus的协议控制 （层2）
  半双工异步通信
  多主方式
  广播通信功能
  可选择传输速率不同的3种模式
- 内置数据接收/发送用缓冲器
  发送缓冲器/接收缓冲器各128字节
  可连续接收/发送128字节 （模式2的最大传输字节数）
- 运行频率
  12MHz、 12.58MHz （使用Pϕ或者AUDIO_X1*、 AUDIO_X2*的1/2分频时钟）
  18MHz、 18.87MHz （使用Pϕ或者AUDIO_X1*、 AUDIO_X2*的1/3分频时钟）
  24MHz、 25.16MHz （使用Pϕ或者AUDIO_X1*、 AUDIO_X2*的1/4分频时钟）
  30MHz、 31.45MHz （使用Pϕ或者AUDIO_X1*、 AUDIO_X2*的1/5分频时钟）
  36MHz、 37.74MHz （使用AUDIO_X1*、 AUDIO_X2*的1/6分频时钟）

【注】 * 只有在不用作 SSI 音频的时钟输入时，才能用作 IEB 的时钟输入。

- 可设定模块待机模式

20.1.1　IEBus 通信协议

IEBus 的概要如下：

- 通信方式：半双工异步通信
- 多主方式
  能实现与IEBus连接的全部单元向其他单元传输数据。
- 广播通信功能 （1个单元对多个单元的通信）
  组广播通信　：对组单元的广播通信
  同时广播通信：对全部单元的广播通信
- 可选择传输速率不同的3种模式

表 20.1　3 种模式

模式	IEBϕ*1=12MHz、 18MHz、 24MHz、 30MHz、 36MHz*2	IEBϕ*1=12.58MHz、 18.87MHz、 25.16MHz、 31.45MHz、 37.74MHz*2	最大传输字节数 （字节 / 帧）
0	大约 3.9kbps	大约 4.1kbps	16
1	大约 17kbps	大约 18kbps	32
2	大约 26kbps	大约 27kbps	128

【注】 *1　外围时钟 （Pϕ）或者 AUDIO_X1、 AUDIO_X2 的时钟
*2　使用本 LSI 时的振荡频率

- 存取控制：CSMA/CD （Carrier Sense Multiple Access with Collision Detection）
  总线占有的优先级如下：
  1. 广播通信 （1个单元对多个单元的通信）优先于正常通信 （1个单元对1个单元的通信）
  2. 基本地址最小的优先
- 通信规模
  单元数：最大50
  电缆长：最长150m （使用双绞线电缆时）

【注】 根据外接 IEBus 驱动器 / 接收器的特点与使用电缆的特点，实际系统的通信规模会有不同。

（1） 决定总线占有权 （仲裁）

与 IEBus 连接的器材控制其他的器材时，为了占有总线进行运行。该运行称为仲裁。所谓仲裁就是：如果多个单元同时开始发送，就对其中的 1 个单元进行允许占有总线的处理。

通过仲裁仅能使 1 个器材获得总线占有权，因此可决定以下总线占有的优先条件。

（a） 根据通信种类的优先

广播通信 （1 个单元对多个单元的通信）优先于正常通信 （1 个单元对 1 个单元的通信）。

（b） 根据基本地址的优先

通信种类相同时，基本地址最小的单元优先。

例：基本地址由 12 位构成， H'000 的单元拥有最高优先级， H'FFF 的单元拥有最低优先级。

【注】 如果仲裁失败，可自动进入重新发送模式 （重新发送次数可通过 IEMCR 的 RN 位设定为 0 ～ 7 次）。

（2） 通信模式

IEBus 中准备有传输速率不同的 3 种通信模式。各通信模式的传输速率以及 1 个通信帧中的最大传输字节数，如表 20.2 所示。

表 20.2　各通信模式的传输速率以及最大传输字节数

通信模式	最大传输字节数（字节 / 帧）	实际传输速率 *1 （kbps）	
		IEBϕ*2=12MHz、 18MHz、 24MHz、 30MHz、 36MHz*3	IEBϕ*2=12.58MHz、 18.87MHz、 25.16MHz、 31.45MHz、 37.74MHz*3
0	16	大约 3.9	大约 4.1
1	32	大约 17	大约 18
2	128	大约 26	大约 27

【注】 在进行通信之前预先选择连接 IEBus 的各器件的通信模式。主单元与其通信对象（从属单元）的通信模式如果不相同，通信就不能正确进行。

在 IEBϕ=6MHz 的器件与 IEBϕ=6.29MHz 的器件之间，即使通信模式相同，通信也不能正确进行。因此，必须使用相同的振荡频率进行通信。

*1　传送最大传输字节数时的实际传输速率

*2　外围时钟 （Pϕ）或者 AUDIO_X1、 AUDIO_X2 的时钟

*3　使用本 LSI 时的振荡频率

（3） 通信地址

IEBus 中各器件分配有 12 位固定的通信地址。通信地址的构成如下：

高 4 位：组号码 （各器件所属组的识别号码）

低 8 位：单元号码 （组内各器件的识别号码）

（4） 广播通信

正常发送时，主单元与其通信对象 （从属单元）都是 1 个单元，能进行 1 对 1 的发送或接收。与此对应，广播通信中存在多个从属单元，主单元对多个从属单元进行发送。由于从属单元为多个，因此在通信中不能从从属单元返回应答。

由广播位决定是进行广播通信还是进行正常通信。（关于广播位，请参照“20.1.2 （1）（b） 广播位”）。

有下面 2 种广播通信。

（a） 组广播通信

对组内器件 （通信地址的高 4 位的组号码相同）进行广播通信。

（b） 同时广播通信

与组号码的值无关，对全部器件进行广播通信。

通过从属地址的值，识别是组广播还是同时广播（关于从属地址，请参照“20.1.2（3） 从属地址字段”）。

20.1.2　传输协议

IEBus 传输信号格式如图 20.1 所示。

通信数据作为被称为通信帧的一连串信号传送。 1 个通信帧可传送的数据个数以及传输速率，根据通信模式而不同。

（IEBϕ= 12MHz、18MHz、24MHz、30MHz、36MHz时）

字段名	头部		基本地址字段		从属地址字段			控制字段			信息长度字段			数据 字段					
位数	1	1	12	1	12	1	1	4	1	1	8	1	1	8	1	1	8	1	1
	起始位	广播位	基本地址	P	从属地址	P	A	控制位	P	A	信息长度位	P	A	数据位	P	A	数据位	P	A
传输时间																			
模式0	大约7330ms													大约1590×Nms					
模式1	大约2090ms													大约410×Nms					
模式2	大约1590ms													大约300×Nms					

P：奇偶位（1位）
A：应答位（1位）
　　A = 0时：ACK
　　A = 1时：NAK
N：数据字节数

【注】 广播通讯时，无视应答位的值。

图 20.1　传输信号格式

（1） 头部

头部由起始位以及广播位构成。

（a） 起始位

起始位是向其他单元通知数据传输开始的信号。

准备开始数据传输的单元，按照决定的时间输出低电平信号 （起始位），然后输出广播位。

准备输出起始位时，如果其他单元已经输出起始位，就不输出起始位等待直到其他单元的起始位输出结束，然后与结束的时序同步输出广播位。

不是开始发送的单元在检测出起始位后进入接收状态。

（b） 广播位

广播位用于识别广播通信与正常通信。

广播位为 0 时表示广播通信，为 1 时表示正常通信。广播通信中有组广播和同时广播，根据从属地址的值区分 （关于从属地址，请参照 “20.1.2 （3） 从属地址字段”）。

广播通信时，因为有多个通信对象 （从属单元），因此不返回 （2）及以后各字段的应答位。

2 个及 2 个以上的单元按照相同时序开始发送通信帧时，广播通信优先于正常通信，并赢得仲裁。

（2） 基本地址字段

基本地址字段是向其他单元发送自身单元地址 （基本地址）的字段。基本地址字段由基本地址位和奇偶校验位构成。

基本地址由 12 个位构成，从 MSB 开始输出。

2 个及 2 个以上的单元按照相同时序开始发送相同值的广播位时，仲裁取决于基本地址字段。

在基本地址字段每发送 1 位时，自己输出的数据与总线上的数据进行比较。如果自己输出的基本地址和总线上的数据不同，就判断为仲裁失败，中止发送，进入接收状态。

因为 IEBus 是由线 AND 构成，因此在参加仲裁的单元（仲裁主）中，拥有最小基本地址的单元赢得仲裁。

最终，在输出 12 位的基本地址后，仅 1 个单元作为主单元保持在发送状态。

接下来，该主单元输出奇偶校验位 *，对其他单元确定基本地址，然后开始从属地址字段输出。

【注】 * 奇偶性使用偶校验，基本地址位中 1 位的数是奇数时，奇偶校验位为 1。

（3） 从属地址字段

从属地址字段是发送需要进行通信的单元（从属单元）地址 （从属地址）的字段。从属地址字段由从属地址位、奇偶校验位以及应答位构成。

从属地址由 12 个位构成，从 MSB 开始输出。12 位从属地址发送后，为避免错误接收从属地址，输出奇偶校验位。接下来，为确认从属单元是否存在于总线，主单元检测来自从属单元的应答信号。如果检测出应答信号，就进入控制字段输出。但广播通信时，不检测应答位，直接进入控制字段输出。

如果从属地址匹配，并且检测出基本地址和从属地址的奇偶性为偶数时，从属单元输出应答信号。从属单元奇偶性为奇数时，就判断为基本地址或者从属地址没有正确接收，并且不输出应答信号。此时，主单元为待机 （监控）状态，通信结束。

在广播通信时，从属地址用于识别组广播以及同时广播，如下：

从属地址为 H'FFF 时 ：同时广播

从属地址不是 H'FFF 时：组广播

【注】 组广播通信时的组号码为从属地址高 4 位的值。

（4） 控制字段

控制字段是用于发送以下数据字段的种类和方向的字段。控制字段由控制位、奇偶校验位及应答位构成。

控制位由 4 个位构成，从 MSB 开始输出。

在控制位之后，输出奇偶校验位。奇偶性为偶数且从属单元可执行主单元的请求功能时，从属单元输出应答信号，并进入下面的信息长度字段。但是，在即使奇偶性为偶数且从属单元不能执行主单元的请求时，或者奇偶性是奇数时，从属单元不输出应答信号，返回到待机 （监控）状态。

主单元确认应答信号后，进入下面的信息长度字段。

不能确认应答信号时，主单元为待机状态，通信结束。但是在广播通信时，主单元不确认应答信号，直接进入下面的信息长度字段。

关于控制位的内容请参照 “表 20.4”。

（5） 信息长度字段

信息长度字段用于指定通信数据的字节数。信息长度字段由信息长度位、奇偶校验位以及应答位构成。

信息长度字段由 8 个位构成，从 MSB 开始输出。通信数据的字节数如 “表 20.3”所示。

表 20.3　信息长度位的内容

信息长度位	发送数据 / 字节数
H'01	1 字节
H'02	2 字节
•	•
•	•
H'FF	255 字节
H'00	256 字节

【注】 根据通信模式，设定为超过 1 帧的最大传输字节数时，成为多个帧的通信。此时，第 2 次及其以后，信息长度位为剩余通信数据的字节数。本 LSI 中信息长度位不能设定为超过 1 帧的最大传输字节数。请在以下的范围内设定。

模式 0：1 ～ 16 字节

模式 1：1 ～ 32 字节

模式 2：1 ～ 128 字节

该字段在主发送时 （控制位的 bit3 为 1）的运行与主接收时 （控制位的 bit3 为 0）的运行不同。

（a） 主发送时

主单元输出信息长度位以及奇偶校验位。如果检测出奇偶性为偶数，从属单元就输出应答信号，进入下面的数据字段。但是，在广播通信时，从属单元不输出应答信号。

当奇偶性为奇数时，从属单元就判断没有正确接收信息长度位，不输出应答信号，并返回到待机 （监控）状态。此时，主单元也返回到待机状态，并结束通信。

（b） 主接收时

从属单元输出信息长度位以及奇偶校验位。如果检测出奇偶性为偶数时，主单元输出应答信号。

奇偶性为奇数时，主单元就判断没有正确接收信息长度位，不输出应答信号，并返回到待机状态。此时，从属单元也返回到待机状态，并结束通信。

（6） 数据字段

数据字段是对从属单元发送 / 接收数据的字段。主单元使用数据字段向从属单元发送数据，以及接收来自从属单元的数据。数据字段由数据位、奇偶校验位以及应答位构成。

数据单元由 8 个位构成，从 MSB 开始输出。

数据位后，分别从主单元以及从属单元输出奇偶校验位和应答位。

只在主单元的发送运行时进行广播通信。此时，忽视应答信号。

主发送时和主接收时的运行如下：

（a） 主发送时

由主单元对从属单元进行写入时，主单元对从属单元发送数据位、奇偶校验位。从属单元接收数据位、奇偶校验位，如果奇偶性为偶数且接收缓冲器为空，就输出应答信号。如果奇偶性为奇数或者接收缓冲器不为空时，从属单元拒绝接收对应的数据，并不输出应答信号。

从属单元不输出应答信号时，主单元再次发送相同的数据。到检测出来自从属单元的应答信号或者超过数据最大传输字节数为止，持续该运行。

奇偶性为偶数，并且从属单元输出应答信号时，如果数据有剩余，并没有超过最大传输字节数，主单元就发送下面的数据。

在广播通信时，从属单元不输出应答信号，主单元按每 1 字节为单位传送数据。

（b） 主接收时

主单元从从属单元进行读取时，主单元输出与全部读取位对应的同步信号。

从属单元对应来自主单元的同步信号，将数据、奇偶校验位的内容输出到总线。

主单元读取从属单元输出的数据奇偶校验位，确认奇偶性。

奇偶性为奇数时或者接收缓冲器不为空时，主单元拒绝接收该数据，且不输出应答信号。在 1 个通信帧中，如果数据个数在可发送的最大传输字节数以内，主单元就反复进行相同数据的读取运行。如果奇偶性为偶数，且接收缓冲器为空，主单元就接受数据，并返回应答信号。在 1 个通信帧中，如果数据个数在可发送的最大传输字节数以内，主单元就读取下面的数据。

（7） 奇偶校验位

奇偶校验位用于确认传输数据没有错误。

奇偶校验位附加给基本地址位、从属地址位、控制位、信息长度位、数据位的各数据。

奇偶性为偶校验。数据中 1 的位数是奇数时，奇偶校验位为 1。数据中 1 的位数是偶数时，奇偶校验位为 0。

（8） 应答位

正常通信 （1 个单元对 1 个单元的通信）中，为了确认是否正确接收了数据，在以下位置附加应答位。

- 从属地址字段的最后
- 控制字段的最后
- 信息长度字段的最后
- 数据字段的最后

应答位的定义如下：

- 0：表示识别了传输数据。（ACK）
- 1：表示没有识别传输数据。（NAK）

但是广播通信时，忽视应答位的内容。

（a） 从属地址字段最后的应答位

从属地址字段最后的应答位，在以下情况时，成为 NAK，并中止传送。

- 基本地址位或者从属地址位的奇偶性不正确时
- 发生时序错误 （在位格式有错误）时
- 不存在从属单元时

（b） 控制字段最后的应答位

控制字段最后的应答位，在以下情况时，成为 NAK，并中止传输。

- 控制位的奇偶性不正确时
- 虽然从属接收缓冲器*不为空，但控制位的bit3为1 （写入运行）时
- 虽然从属发送缓冲器*为空，但控制位为读取数据 （H'3、 H'7）的时
- 虽然从属单元被锁定，但没有被锁定的单元请求控制位的H'3、 H'6、 H'7、 H'A、 H'B、 H'E、 H'F时
- 虽然没被锁定，但控制位为读取锁定地址 （H'4、 H'5）的时
- 发生时序错误时
- 未定义的控制位时

【注】 * 请参照 “20.1.3 （1） 从属状态 （SSR）的读取 （控制位：H'0、 H'6）”。

（c） 信息长度字段最后的应答位

信息长度字段最后的应答位，在以下情况时，成为 NAK，并中止传输。

- 信息长度位的奇偶性不正确时
- 发生时序错误时

（d） 数据字段最后的应答位

数据字段最后的应答位，在以下情况时，成为 NAK，并中止传输。

- 数据位的奇偶性不正确时 *
- 在前面应答位传送以后发生时序错误时
- 接收缓冲器为满，不能再接收数据时 *

【注】 * 在这种情况下，如果在 1 个帧中数据个数没有超过可传送的最大传输字节数，就重新执行该数据字段的发送，直到达到最大传输字节数为止。

20.1.3　传输数据 （数据字段的内容）

数据字段的内容为通过控制位表示的数据。

表 20.4　控制位的内容

设定值	bit3*1	bit2	bit1	bit0	功　能 *2
H'0	0	0	0	0	从属状态 （SSR）的读取
H'1	0	0	0	1	未定义
H'2	0	0	1	0	未定义
H'3	0	0	1	1	数据读取和锁定
H'4	0	1	0	0	锁定地址的读取 （低 8 位）
H'5	0	1	0	1	锁定地址的读取 （高 4 位）
H'6	0	1	1	0	从属状态 （SSR）的读取与锁定解除
H'7	0	1	1	1	数据读取
H'8	1	0	0	0	未定义
H'9	1	0	0	1	未定义
H'A	1	0	1	0	命令写入和锁定
H'B	1	0	1	1	数据写入和锁定
H'C	1	1	0	0	未定义
H'D	1	1	0	1	未定义
H'E	1	1	1	0	命令写入
H'F	1	1	1	1	数据写入

【注】 *1 根据 bit3 （MSB）的值，以后的信息长度字段的信息长度位以及数据字段的数据传送方向发生变化。
bit3 为 1 时：主单元向从属单元传送
bit3 为 0 时：从属单元向主单元传送

*2 H'3、 H'6、 H'A、 H'B 是指定设定以及解除锁定的控制位。
发送 H'1、 H'2、 H'8、 H'9、 H'C、 H'D 的未定义值时，不返回应答。

如果从请求锁定的主单元以外接收的控制位不包括在表 20.5 中，通过主单元被锁定的单元就拒绝接收控制位，并且不输出应答位。

表 20.5　被锁定的从属单元的控制字段

设定值	bit3	bit2	bit1	bit0	功　能
H'0	0	0	0	0	从属状态的读取
H'4	0	1	0	0	锁定地址的读取 （低 8 位）
H'5	0	1	0	1	锁定地址的读取 （高 4 位）

（1） 从属状态 （SSR）的读取 （控制位：H'0、 H'6）

通过读取从属状态 （H'0， H'6），主单元可得知从属单元没有回送应答位 （ACK）的理由。从属状态表示从属状态最后进行的通信结果。所有从属单元均能提供从属状态的信息。从属状态如图 20.2 所示。

MSB　　LSB

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

位	值	意　味	
bit7 bit6	00	模式0	表示支持单元的最高位的模式 *1
	01	模式1	
	10	模式2	
	11	用于将来扩展	
bit5	0	0固定	
bit4*2	0	从属发送停止	
	1	可进行从属发送运行	
bit3	0	0固定	
bit2	0	单元没锁定	
	1	单元锁定	
bit1*3	0	从属接收缓冲器空	
	1	从属接收缓冲器不为空	
bit*4	0	从属发送缓冲器空	
	1	从属发送缓冲器不为空	

【注】*1　本LSI可支持到模式2，因此bit7、6固定为10。
*2　可通过IEBus主单元地址寄存器1（IEAR1）的STE位选择bit4的值
*3　从属接收缓冲器为数据编程处理时（控制位:H'A、H'B、H'E、H'F）存取的缓冲器。本LSI对应IEBus接收缓冲寄存器（IERB001～128），bit2为IEBus接收状态寄存器（IERSR）的RXBSY位的值。
*4　从属发送缓冲器为数据写入处理时（控制位：H'3、H'7）存取的缓冲器。本LSI对应IEBus发送缓冲寄存器（IETB001～128），bit0为IEBus通用标志寄存器（IEFLG）SRQ位的值。

图 20.2　从属状态 （SSR）位的构成

（2） 数据命令传送 （控制位：读取 （H'3、 H'7）、写入 （H'A、 H'B、 H'E、 H'F））

数据读取 （H'3、H'7）时，从属单元的数据缓冲器中的数据被读取到主单元。数据写入 （H'B、H'F）或命令写入 （H'A、 H'E）时，从属单元接收的数据按照该从属单元的运行规定进行处理。

【注】 1. 用户可根据系统自由选择数据和命令。
2. H'3、 H'A、 H'B 根据通信条件与状态有被锁定的情况。

（3） 锁定地址的读取 （控制位：H'4、 H'5）

锁定地址的读取处理（H'4、H'5）时，发送锁定指令的主单元地址（12 位）如下所示，以 1 字节为单位构成、读出。

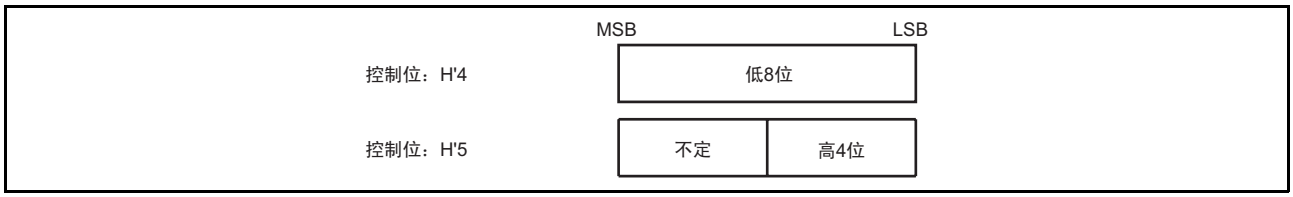


图 20.3　锁定地址的结构

（4） 锁定的设定、解除 （控制位：设定 （H'3、 H'A、 H'B）、解除 （H'6））

锁定功能在通过多个通信帧传送信息时使用。锁定的单元不能从没有锁定的单元接收数据。

锁定的设定以及解除如下所示。

（a） 锁定的设定

在指定锁定的控制位（H'3、H'A、H'B），信息长度字段的应答 bit0 发送 / 接收结束后，在没完成发送 / 接收数据 （信息长度位指定的字节数的数据）前，如果通信帧结束，从属单元就通过主单元被锁定。此时，表示从属状态的字节中与锁定相关的位 （bit2）被置 1。

只有在超过帧的最大传输字节数时才设定锁定，不通过其他错误中止设定锁定。

（b） 锁定的解除

在指定锁定的控制位（H'3、H'A、H'B）或者指定锁定解除的控制位（H'6），在 1 个通信帧内，信息长度位指定的字节数的数据发送 / 接收完成后，从属单元通过主单元解除锁定。此时，表示从属状态的字节中与锁定相关的位 （bit2）被复位为 0。

在广播通信中，不进行锁定的设定以及解除。

【注】 有以下 3 种方法能使指定解除锁定的单元自身解锁。

- 执行上电复位
- 使单元进入深度待机模式
- 通过IEBus命令寄存器 （IECMR）发送锁定状态解除命令

另外，可通过IEFLG的LCK位确认有无锁定状态。

20.1.4　位格式

构成 IEBus 的通信帧的位格式 （概念）如图 20.4 所示。

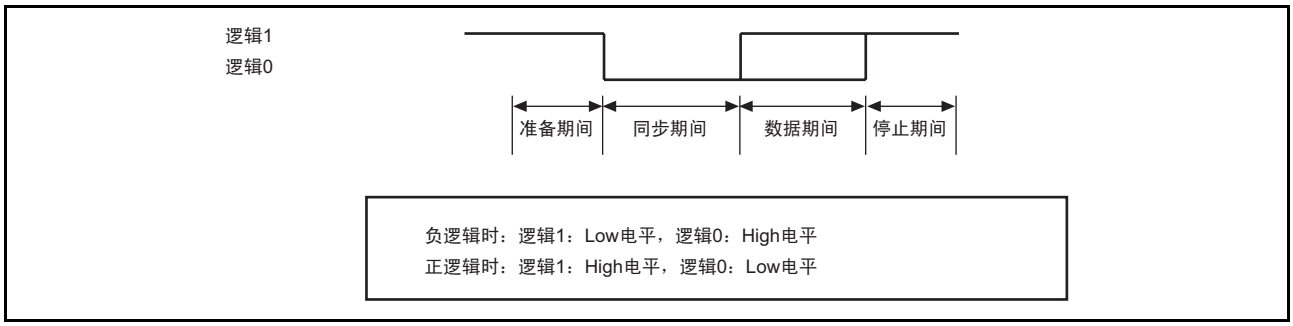


图 20.4　IEBus 的位格式 （概念）

以下是说明正逻辑时位格式的各个期间。

准备期间：最初的逻辑 1 期间 （High 电平）

同步期间：接下来的逻辑 0 期间 （Low 电平）

数据期间：表示位值的期间 （逻辑 1：High 电平、逻辑 0：Low 电平）

停止期间：最后的逻辑 1 期间 （High 电平）

负逻辑时为从正逻辑反转的电平。

同步期间和数据期间的长度大体上相等。

使 IEBus 的每 1 个位取得同步。关于所有位的时间和分配到该位期间的时间规格，根据传输位的种类以及单元 （是主单元还是从属单元）的不同而不同。

20.1.5　构成

IEB 的整体框图如**图** 20.5 所示，各块的功能如**表** 20.6 所示。

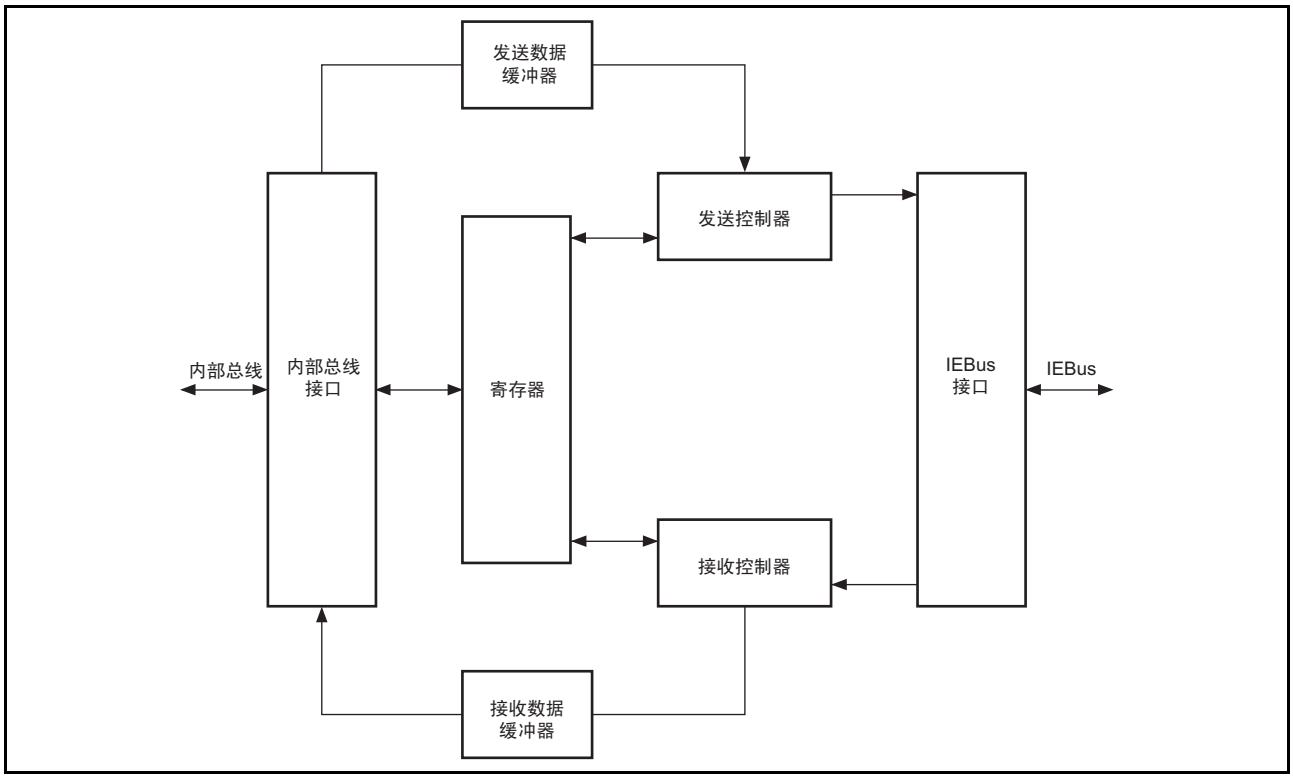


图 20.5　IEB 的框图

表 20.6　各块的功能

块	功　能
内部总线接口	内部总线的接口功能
	• 数据宽度 8 位 • IEB 寄存器存取
IEBus 接口	依据 IEBus 规格的接口功能
	• 将数据按照 IEBus 规格的位格式，从发送控制器发送到 IEBus • 从 IEBus 规格的位格式取出帧数据传送给接收控制器
寄存器	IEB 控制寄存器
	• 控制 IEB 的寄存器 • 可通过内部总线读 / 写
发送控制器	将发送缓冲器里的数据发送到 IEBus
	• 来自寄存器的头部信息与发送缓冲器的数据生成发送帧，并发送 • 检测出发送错误
接收控制器	将来自 IEBus 的数据保存到接收缓冲器
	• 接收的帧的头部信息保存到寄存器，而数据保存到接收缓冲器 • 检测出接收错误
发送数据缓冲器	用于数据发送的缓冲器
	• 保存向 IEBus 发送的数据的缓冲器 • 缓冲器容量为 128 字节
接收数据缓冲器	用于数据接收的缓冲器
	• 保存来自 IEBus 的接收数据的缓冲器 • 缓冲器容量为 128 字节

20.2　输入 / 输出引脚

表 20.7　引脚构成

名称	引脚名	输入 / 输出	功　能
IEB 接收数据引脚	IERxD	输入	接收数据输入引脚
IEB 发送数据引脚	IETxD	输出	发送数据输出引脚

20.3　寄存器的说明

在 IEB 中有以下的寄存器。

各寄存器原则上是 8 位长，并以 8 位单位进行存取。

表 20.8　寄存器构成

寄存器名	简称	R/W	初始值	地址	存取长度
IEBus 控制寄存器	IECTR	R/W	H'00	H'FFFEF000	8
IEBus 命令寄存器	IECMR	W	H'00	H'FFFEF001	8
IEBus 主控制寄存器	IEMCR	R/W	H'00	H'FFFEF002	8
IEBus 主单元地址寄存器 1	IEAR1	R/W	H'00	H'FFFEF003	8
IEBus 主单元地址寄存器 2	IEAR2	R/W	H'00	H'FFFEF004	8
IEBus 从属地址设定寄存器 1	IESA1	R/W	H'00	H'FFFEF005	8
IEBus 从属地址设定寄存器 2	IESA2	R/W	H'00	H'FFFEF006	8
IEBus 发送信息长度寄存器	IETBFL	R/W	H'00	H'FFFEF007	8
IEBus 接收基本地址寄存器 1	IEMA1	R	H'00	H'FFFEF009	8
IEBus 接收基本地址寄存器 2	IEMA2	R	H'00	H'FFFEF00A	8
IEBus 接收控制字段寄存器	IERCTL	R	H'00	H'FFFEF00B	8
IEBus 接收信息长度寄存器	IERBFL	R	H'00	H'FFFEF00C	8
IEBus 锁定地址寄存器 1	IELA1	R	H'00	H'FFFEF00E	8
IEBus 锁定地址寄存器 2	IELA2	R	H'00	H'FFFEF00F	8
IEBus 通用标志寄存器	IEFLG	R	H'00	H'FFFEF010	8
IEBus 发送状态寄存器	IETSR	R/(W)*	H'00	H'FFFEF011	8
IEBus 发送中断允许寄存器	IEIET	R/W	H'00	H'FFFEF012	8
IEBus 接收状态寄存器	IERSR	R/(W)*	H'00	H'FFFEF014	8
IEBus 接收中断允许寄存器	IEIER	R/W	H'00	H'FFFEF015	8
IEBus 时钟选择寄存器	IECKSR	R/W	H'01	H'FFFEF018	8
IEBus 发送数据缓冲器 001 ～ 128	IETB001 ～ IETB128	W	不定	H'FFFEF100 ～ H'FFFEF17F	8
IEBus 接收数据缓冲器 001 ～ 128	IERB001 ～ IERB128	R	不定	H'FFFEF200 ～ H'FFFEF27F	8

【注】 *　为清除标志只能写 1。

20.3.1　IEBus 控制寄存器 （IECTR）

IECTR 是控制 IEB 运行的寄存器。

位：	7	6	5	4	3	2	1	0
	-	IOL	DEE	-	RE	-	-	-
初始值：	0	0	0	0	0	0	0	0
R/W：	R	R/W	R/W	R	R/W	R	R	R

位	位名	初始值	R/W	说　明
7	—	0	R	保留位 读写的值总是 0。
6	IOL	0	R/W	输入 / 输出电平 选择 IERxD、 IETxD 引脚的输入 / 输出引脚的电平 （正逻辑、负逻辑）。 0：引脚的输入 / 输出为负逻辑 （逻辑 1 为 Low 电平，逻辑 0 为 High 电平） 1：引脚的输入 / 输出为正逻辑 （逻辑 1 为 High 电平，逻辑 0 为 Low 电平）
5	DEE	0	R/W	广播接收错误中断允许 将本位置 1 时，如果在广播接收时，接收缓冲器为不能接收的状态（RE 位没有被置 1 或者 RXBSY 标志被置位的状态），就产生广播接收错误中断。此时，基本地址保存在 IEBus 接收基本地址寄存器 1、 2。 本位是 0 时，如果在广播接收时，接收缓冲器为不能接收的状态，就不产生广播接收错误中断，而是中止接收并进入待机状态。不保存基本地址。 0：到控制字段为止不发生广播接收错误。 1：到控制字段为止发生广播接收错误。
4	—	0	R	保留位 读写的值总是 0。
3	RE	0	R/W	接收允许 允许 / 禁止 IEB 的接收。请在帧接收前进行初始设定时，设定本位。 0：禁止接收运行 1：允许接收运行
2 ～ 0	—	均为 0	R	保留位 读写的值总是 0。

20.3.2　IEBus 命令寄存器 （IECMR）

IECMR 发送控制 IEB 通信的命令。本寄存器是写专用寄存器。读取时值不定。

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

位	位名	初始值	R/W	说　明
7 ～ 3	—	均为 0	—	保留位 读取时值不定。写入值总是 0。
2 ～ 0	CMD	000	W	命令位 发送控制 IEB 通信的命令。本命令发送后，IEFLG 的 CMX 位被置位的期间为执行命令的期间。 CMX 为 0 后，进入运行状态。 000：空操作 （不影响运行） 001：解除锁定状态 （其他单元的请求） *1 010：请求作为主的通信 011：中止主通信 *2 100：未定义 *4 101：请求来自从属的数据发送 110：中止来自从属的数据发送 *3 111：未定义 *4

【注】 *1　在从属通信状态禁止执行。

*2　只有在主通信中 （MRQ=1）时，本命令才有效。在其他状态中，忽视该命令的发送。在主通信中发送本命令时，通信控制器立刻进入待机状态。同时，结束发送的主发送请求 （MRQ=0）。

*3　只有在从属发送 （SRQ=1）时，本命令才有效。在其他状态中，忽视该命令的发送。在从属发送中发送本命令时，在进行从属发送之前 SRQ=0，因此不能响应来自主单元的发送请求。在进行从属发送的状态中发送命令时，中止发送运行，并进入待机状态。（SRQ=0）

*4　为未定义位。即使发送本命令，对运行也没有影响。

20.3.3　IEBus 主控制寄存器 （IEMCR）

IEMCR 设定进行主通信时的通信条件。

位：	7	6	5	4	3	2	1	0
	SS	RN			CTL			
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	SS	0	R/W	选择广播 / 正常通信 选择主通信时的广播 / 正常通信。 0：主通信时，广播通信 1：主通信时，正常通信
6 ～ 4	RN	000	R/W	重新发送次数 主通信中如果仲裁失败，自动设定重新发送次数。如果仲裁失败， IETSR 的 TXEAL 位被置位，并且发送错误结束。 000：0 次 001：1 次 010：2 次 011：3 次 100：4 次 101：5 次 110：6 次 111：7 次
3 ～ 0	CTL*1	0000	R/W	控制位 设定控制位 （主发送时，控制字段的控制位）。 0000：读取从属状态 0001：未定义 *3 0010：未定义 *3 0011：数据读取与锁定 *2 0100：读取锁定地址 （低 8 位） 0101：读取锁定地址 （高 4 位） 0110：读取从属状态与锁定解除 *2 0111：数据读取 1000：未定义 *3 1001：未定义 *3 1010：命令写入与锁定 *2 1011：数据写入与锁定 *2 1100：未定义 *3 1101：未定义 *3 1110：命令写入 1111：数据写入

【注】 *1　以后的信息长度字段的信息长度位以及数据字段的数据传送方向根据 bit3 的值变化：
bit3 为 1 时：从主单元向从属单元传送
bit3 为 0 时：从从属单元向主单元传送

*2　为指定锁定以及解除的控制位。

*3　禁止使用。

20.3.4　IEBus 主单元地址寄存器 1 （IEAR1）

IEAR1 设定主单元地址的低 4 位与通信模式。主通信时，主单元地址为基本地址字段的值。从属通信时，与接收的从属地址字段进行比较。

位：	7	6	5	4	3	2	1	0
	IARL4				IMD		-	STE
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 ～ 4	IARL4	0000	R/W	IEBus 主单元地址的低 4 位 设定主单元地址的低 4 位。为基本地址字段的值。从属通信时，与接收的从属地址进行比较。
3、 2	IMD	00	R/W	IEBus 通信模式 选择 IEBus 通信模式。 00：通信模式 0 01：通信模式 1 10：通信模式 2 11：禁止设定
1	—	0	R	保留位 读写的值总是 0。
0	STE	0	R/W	设定从属发送 设定从属状态寄存器 bit4 的值。将本位设定为 1 时，通过从属状态寄存器的发送，将从属单元为可发送状态的信息告知主单元。本位只是设定从属状态寄存器的值，不直接影响从属发送的运行。 0：从属状态的 bit4 为 0 （从属发送为停止状态） 1：从属状态的 bit4 为 1 （从属发送为可能的状态）

20.3.5　IEBus 主单元地址寄存器 2 （IEAR2）

IEAR2 设定主单元地址的高 8 位。主通信时，本寄存器为基本地址字段的值。从属通信时，与接收的从属地址字段进行比较。

位：	7	6	5	4	3	2	1	0
	IARU8							
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 ～ 0	IARU8	均为 0	R/W	IEBus 主单元地址的高 8 位 设定主单元地址的高 8 位。为基本地址字段的值。从属通信时，与接收的从属地址进行比较。

20.3.6　IEBus 从属地址设定寄存器 1 （IESA1）

IESA1 设定通信对象从属单元地址的低 4 位。

位：	7	6	5	4	3	2	1	0
	ISAL4				-	-	-	-
初始值：	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R	R	R	R

位	位名	初始值	R/W	说　明
7 ～ 4	ISAL4	0000	R/W	IEBus 从属地址低 4 位 设定通信对象从属单元地址的低 4 位。
3 ～ 0	—	均为 0	R	保留位 读写的值总是 0。

20.3.7　IEBus 从属地址设定寄存器 2 （IESA2）

IESA2 设定通信对象从属单元地址的高 8 位。

位：	7	6	5	4	3	2	1	0
	ISAU8							
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 ～ 0	ISAU8	均为 0	R/W	IEBus 从属地址的高 8 位 设定通信对象从属单元地址的高 8 位。

20.3.8　IEBus 发送信息长度寄存器 （IETBFL）

IETBFL 设定进行主发送以及从属发送时的信息长度。

位：	7	6	5	4	3	2	1	0
	IBFL							
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 ～ 0	IBFL	均为 0	R/W	发送信息长度位 设定主发送时的信息长度的值。请设定通信模式的最大传输字节数以内的值。 H'01：1 字节 H'02：2 字节 ： H'7F：127 字节 H'80：128 字节 H'81：未定义 * ： H'FF：未定义 * H'00：未定义 *

【注】 *　禁止设定。

20.3.9　IEBus 接收基本地址寄存器 1 （IEMA1）

IEMA1 表示从属 / 广播接收时的通信对象主单元地址的低 4 位。

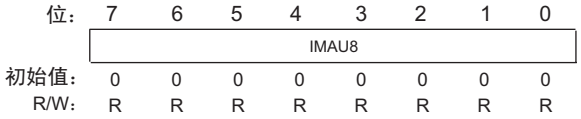
位：	7	6	5	4	3	2	1	0
	IMAL4				-	-	-	-
初始值：	0	0	0	0	0	0	0	0
R/W：	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说　明
7 ～ 4	IMAL4	0000	R	IEBus 接收基本地址的低 4 位 表示从属 / 广播接收时的通信对象主单元地址的低 4 位。从属 / 广播接收开始时，本寄存器为有效 （在 RXS 位被置位时，内容被改写）。 通过 IECTR 的 DEE 位选择广播接收错误时，如果在控制字段接收时，接收缓冲器不为可接收信息的状态，就产生接收错误中断，并且将基本地址低 4 位保存到 IEMA1。
3 ～ 0	—	均为 0	R	保留位 读写的值总是 0。

20.3.10　IEBus 接收基本地址寄存器 2 （IEMA2）

IEMA2 表示从属 / 广播接收时通信对象主单元地址的高 8 位。从属 / 广播接收开始时，本寄存器为有效（IERSR 的 RXS 位被置位时，内容被改写）。

通过 IECTR 的 DEE 位选择广播接收错误中断时，如果在控制字段接收时，接收缓冲器不为可接收信息的状态，就产生接收错误中断，基本地址的高 8 位保存到 IEMA2。对本寄存器的写无效。

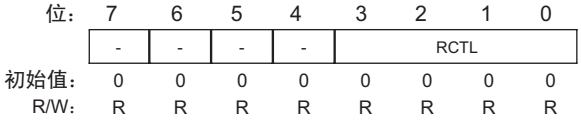


位	位名	初始值	R/W	说　明
7 ～ 0	IMAU8	均为 0	R	IEBus 接收基本地址的高 8 位 表示从属 / 广播接收时的通信对象主单元地址的高 8 位。从属 / 广播接收开始时，本寄存器为有效 （在 RXS 位被置位时，内容被改写）。 通过 IECTR 的 DEE 位选择广播接收错误时，如果在控制字段接收时，接收缓冲器不为可接收信息的状态，就产生接收错误中断，并且将基本地址的高 8 位保存到 IEMA2。

20.3.11　IEBus 接收控制字段寄存器 （IERCTL）

IERCTL 表示从属 / 广播接收时的控制字段的值。从属 / 广播接收开始时，本寄存器为有效（IERSR 的 RXS 位被置位时，内容被改写）。

对本寄存器的写无效。



位	位名	初始值	R/W	说　明
7 ～ 4	—	均为 0	R	保留位 读写的值总是 0。
3 ～ 0	RCTL	0000	R	IEBus 接收控制字段 表示从属 / 广播接收时的控制字段的值。从属 / 广播接收开始时，本寄存器为有效 （RXS 位被置位时，内容被改写）。

20.3.12　IEBus 接收信息长度寄存器 （IERBFL）

IERBFL 表示从属 / 广播接收时的信息长度字段。从属 / 广播接收开始时，本寄存器为有效（IERSR 的 RXS 位被置位时，内容被改写）。

对本寄存器的写无效。

位：	7	6	5	4	3	2	1	0
	RBFL							
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	RBFL	均为 0	R	IEBus 接收信息长度 读出从属 / 广播接收时的信息长度字段的内容。

20.3.13　IEBus 锁定地址寄存器 1 （IELA1）

IELA1 表示锁定时的锁定地址的低 8 位。

位：	7	6	5	4	3	2	1	0
	ILAL8							
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	ILAL8	均为 0	R	IEBus 锁定地址的低 8 位 表示锁定的主单元地址的低 8 位。只有在 IEFLG 的 LCK 位被置位时才有效。

20.3.14　IEBus 锁定地址寄存器 2 （IELA2）

IELA2 表示锁定时的锁定地址的高 4 位。

位：	7	6	5	4	3	2	1	0
	-	-	-	-	ILAU4			
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 ～ 4	—	均为 0	R	保留位 读写的值总是 0。
3 ～ 0	ILAU4	0000	R	IEBus 锁定地址的高 4 位 保存锁定的主单元地址的高 4 位。只有在 IEFLG 的 LCK 位被置位时才有效。

20.3.15　IEBus 通用标志寄存器 （IEFLG）

IEFLG 表示 IEB 命令的执行状态、锁定状态、从属地址匹配、广播接收的检测。

位：	7	6	5	4	3	2	1	0
	CMX	MRQ	SRQ	SRE	LCK	-	RSS	GG
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	CMX	0	R	命令执行状态 表示命令执行状态。 0：命令执行结束 1：命令执行中 ［置位条件］ • 在 MRQ、SRQ、SRE 任意一个置位的条件下，发送主通信请求命令或者从属发送请求命令时 ［清除条件］ • 命令执行结束时。
6	MRQ	0	R	主通信请求 表示该单元作为主单元是在通信请求期间，还是不在通信请求期间。 0：该单元作为主单元不在通信请求期间 1：该单元作为主单元在通信请求期间 ［置位条件］ • 发送主通信请求命令后， CMX 位为 0 时。 ［清除条件］ • 主通信结束时。
5	SRQ	0	R	从属发送请求 表示该单元作为从属单元是在发送请求期间，还是不在发送请求期间。 0：该单元作为从属单元不在发送请求期间 1：该单元作为从属单元在发送请求期间 ［置位条件］ • 发送从属发送请求命令后， CMX 位为 0 时。 ［清除条件］ • 从属发送结束时。
4	SRE	0	R	从属接收状态 表示从属 / 广播接收的执行状态。 0：不是从属 / 广播接收中 1：是从属 / 广播接收中 ［置位条件］ • 在 IECTR 的 RE 位为 1 的状态下，开始从属 / 广播接收时。 ［清除条件］ • 从属 / 广播接收结束时。

位	位名	初始值	R/W	说　明
3	LCK	0	R	表示锁定状态 当该单元通过从主单元接受锁定请求被锁定时，被置位。IELA1、IELA2 的值在本位被置位时有效。 0：锁定解除 1：锁定状态 ［置位条件］ • 从主单元接收锁定设定的控制位，但没接收通过信息长度指定的数据时（只有在信息长度超过帧最大传输字节数时 LCK 才被置位，在其他错误结束时不被置位）。 ［清除条件］ • 锁定解除条件成立时，或者发送锁定解除命令时。
2	—	0	R	保留位 读写的值总是 0。
1	RSS	0	R	接收广播位 表示接收广播位的值。从属 / 广播接收开始时有效 （在 RXS 位被置位时，内容被改写）。 到从属 / 广播接收开始为止，保持上一次的值。 0：接收的广播位为 0 1：接收的广播位为 1
0	GG	0	R	同时广播接收识别 在广播接收中，从属地址识别为 H'FFF 时，被置位。与接收广播位相同，从属 / 广播接收开始时有效 （在 RXS 位被置位时，内容被改写）。 到从属 / 广播接收开始为止，保持上一次的值。从属正常接收时为 0。 0：（1）为从属接收 （2）广播接收时，在从属地址字段没识别 H'FFF 1：广播接收时，在从属地址字段识别 H'FFF

20.3.16　IEBus 发送状态寄存器 （IETSR）

IETSR 检测发送开始、发送正常结束、发送错误结束等的状态。

各源拥有与 IEBus 发送中断允许寄存器（IEIET）对应的位，可设定中断的禁止 / 允许。通过向各位写 1 清除本位。

位：	7	6	5	4	3	2	1	0
	-	TXS	TXF	-	TXEAL	TXETTME	TXERO	TXEACK
初始值：	0	0	0	0	0	0	0	0
R/W：	R	R/(W)*	R/(W)*	R	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	TXS	0	R/(W)*	发送开始 IEB 表示发送开始。 ［置位条件］ • 主发送时，仲裁胜利以及基本地址字段发送结束。 ［清除条件］ • 写 1 时
5	TXF	0	R/(W)*	发送正常结束 表示按信息长度位指定的数据长度进行了发送运行，并正常结束。 ［置位条件］ • 信息长度位指定的数据长度发送结束时 ［清除条件］ • 写 1 时
4	—	0	R	保留位 读写的值总是 0。
3	TXEAL	0	R/(W)*	仲裁失败 如果在主通信时仲裁失败，IEB 就只按照 IEMCR 的 RN 设定的次数，再次从起始位进行发送。设定次数全部仲裁失败时，本位就被置位并进入待机状态。如果在进行设定次数的再次发送时仲裁胜利，本位就不被置 1。在仲裁失败并且通信为待机状态时。本位被置位。 ［置位条件］ • 表示数据发送中发生仲裁失败，并且发送结束。 ［清除条件］ • 写 1 时

位	位名	初始值	R/W	说　明
2	TXETTME	0	R/(W)*	发送时序错误 数据发送中，没有按 IEB 协议规定的时序发送数据时，本位被置位。IEB 设置本位并进入待机状态。 ［置位条件］ • 当数据发送中发生时序错误时。 ［清除条件］ • 写 1 时
1	TXERO	0	R/(W)*	发送帧最大传输字节数溢出 表示数据发送时，由于从接收单元接收 NAK 并重新发送，因此已经发送了通过通信模式定义的最大字节长度；或是由于发送信息长度的值大于最大传输数的值，发送没有结束。 IEB 设置本位并进入待机状态。 ［置位条件］ • 已经发送了通过通信模式定义的最大字节数，但发送没有结束。 ［清除条件］ • 写 1 时
0	TXEACK	0	R/(W)*	应答位 表示在数据字段的应答位接收的数据。 • 数据字段以外的应答位 接收 NAK 时，中止发送并进入待机状态。本位被置 1。 • 在数据字段的应答位 • 在数据字段发送中，如果从接收单元接收　NAK，就在从接收单元接收 ACK 前，再次传送通过通信模式定义的最大字节数。此时，如果在发送中从接收单元接收了 ACK 时，就不设定本位，而继续发送。如果不能接收 ACK，通信结束时，本位被置 1。 【注】该标志在广播通信时无效。 ［置位条件］ • 当检测出应答位 1 （NAK）时。 ［清除条件］ • 写 1 时

【注】 * 为清除标志只能写 1。

20.3.17　IEBus 发送中断允许寄存器 （IEIET）

IEIET 能设定禁止 / 允许 IETSR 的发送开始、发送正常结束、发送错误结束等的各中断源。

位：	7	6	5	4	3	2	1	0
	-	TXSE	TXFE	-	TXEALE	TXE TTMEE	TXEROE	TXE ACKE
初始值：	0	0	0	0	0	0	0	0
R/W：	R	R/W	R/W	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	TXSE	0	R/W	允许发送开始中断 设定发送开始 （TXS）中断的禁止 / 允许。 0：禁止发送开始 （TXS）中断 1：允许发送开始 （TXS）中断
5	TXFE	0	R/W	允许发送正常结束中断 设定发送正常结束 （TXF）中断的禁止 / 允许。 0：禁止发送正常结束 （TXF）中断 1：允许发送正常结束 （TXF）中断
4	—	0	R	保留位 读写的值总是 0。
3	TXEALE	0	R/W	允许仲裁失败中断 设定仲裁失败 （TXEAL）中断的禁止 / 允许。 0：禁止仲裁失败 （TXEAL）中断 1：允许仲裁失败 （TXEAL）中断
2	TXE TTMEE	0	R/W	允许发送时序错误中断 设定发送时序错误 （TXETTMEE）中断的禁止 / 允许。 0：禁止发送时序错误 （TXETTMEE）中断 1：允许发送时序错误 （TXETTMEE）中断
1	TXEROE	0	R/W	允许发送帧最大传输字节数溢出中断 设定发送帧最大传输字节数溢出 （TXEROE）中断的禁止 / 允许。 0：禁止发送帧最大传输字节数溢出 （TXEROE）中断 1：允许发送帧最大传输字节数溢出 （TXEROE）中断
0	TXE ACKE	0	R/W	允许应答位中断 设定应答位 （TXEACKE）中断的禁止 / 允许。 0：禁止应答位 （TXEACKE）中断 1：允许应答位 （TXEACKE）中断

20.3.18　IEBus 接收状态寄存器 （IERSR）

IERSR 检测接收忙、接收开始、接收正常结束、接收错误结束等的状态。各源拥有对应 IEIER 的位，能设定中断的禁止 / 允许。可通过向各位写 1，清除本位。

位：	7	6	5	4	3	2	1	0
	RXBSY	RXS	RXF	RXEDE	RXEOVE	RXE RTME	RXEDLE	RXEPE
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	RXBSY	0	R/(W)*	接收忙 表示接收的数据保存到接收数据缓冲器 （IERB001 ～ IERB128）。在读出全部接收数据之后，请清除。本位被置位的期间，不能接收下面的接收数据。 ［置位条件］ • 接收数据全部写入接收数据缓冲器时 ［清除条件］ • 写 1 时
6	RXS	0	R/(W)*	接收开始 表示 IEB 开始接收。 ［置位条件］ • 从属接收时，数据从主单元到信息长度字段被正确接收时 ［清除条件］ • 写 1 时
5	RXF	0	R/(W)*	接收正常结束 表示按信息长度位指定的数据长度进行了接收运行，并正常结束。 ［置位条件］ • 1：通过信息长度位指定的接收数据字节数已经接收时 ［清除条件］ • 写 1 时

位	位名	初始值	R/W	说　明
4	RXEDE	0	R/(W)*	广播接收错误 表示接收广播接收的控制字段时，接收缓冲器为不能接收状态（RE 位没有被置 1，或者 RXBSY 标志被置位的状态）因此不能接收数据。 IECTR 的 DEE 位是 1 时，本位起作用。 ［置位条件］ • 在广播接收中不能接收数据时 ［清除条件］ • 写 1 时
3	RXEOVE	0	R/(W)*	接收溢出标志 该标志表示数据接收中发生溢出。 在没有清除 RXBSY 标志的状态（即没读出接收数据的状态），开始下面的数据接收时， IEB 设置本标志。此时 IEB 判断发生溢出错误，向通信对象回送 NAK。之后，达到帧最大传输字节数前，通信对象进行再次发送，但是如果 RXBSY 标志处于被置位状态， IEB 就持续发送 NAK。清除 RXBSY 标志时，IEB发送ACK并输入下一个数据。广播接收时，在数据接收开始时如果RXBSY 位被置位，就立刻进入待机状态。接收开始标志 （RXS）被置位后，该标志有效。 ［置位条件］ • 在没有清除 RXBSY 标志的状态中，已接收下一个数据时 ［清除条件］ • 写 1 时
2	RXERTME	0	R/(W)*	接收时序错误 数据接收中，不能按照 IEB 协议规定的时序正确接收数据时，本位被置位。IEB 设置本位并进入待机状态。从接收开始标志（RXS）被置位开始，该标志有效。如果在接收开始标志被置位前发生该错误，就中止通信并进入待机状态。此时，该位不被置位。 ［置位条件］ • 表示在数据接收中发生时序错误 ［清除条件］ • 写 1 时

位	位名	初始值	R/W	说　明
1	RXEDLE	0	R/(W)*	接收帧最大传输字节数溢出 在数据接收时，发生奇偶校验错误或者溢出错误，并进行重新传送引起的接收，因此通信模式定义的最大字节长内，接收没结束，或者，信息长度的值大于最大传输字节数的值，因此接收没结束。IEB 设置本位并进入待机状态。从接收开始标志 （RXS）被置位开始该标志有效。如果在接收开始标志被置位前发生该错误，就中止通信并进入待机状态。此时该位不被置位。 ［置位条件］ • 表示通信模式定义的最大字节数内，接收没结束。 ［清除条件］ • 写 1 时
0	RXEPE	0	R/(W)*	奇偶校验错误 表示数据字段接收中，发生奇偶校验错误。如果在数据字段接收前发生奇偶校验错误，IEB 立刻进入待机状态。RXEPE 不被置位。数据字段接收中，发生奇偶校验错误，达到帧最大传输字节数前没有进行接收时，仍旧不置位 RXEPE。发生奇偶校验错误时，IEB 通过应答位向通信对象回送 NAK。此时，通信对象达到帧最大传输字节数前，进行重新传送，但是在重新接收中，解除奇偶校验错误进行正常接收时，RXEPE 不被置位。不接收信息长度设定的字节数，并且中止接收并进入待机状态时，在没有解除奇偶校验错误的情况下，RXEPE 被置位。广播接收时，如果数据字段接收中发生奇偶校验错误，IEB 设置 RXEPE 并立刻进入待机状态。从接收开始标志（RXS）被置位开始，该标志有效。到接收开始标志被置位前发生该错误，就中止通信并进入待机状态。此时，该位不被置位。 ［置位条件］ • 达到帧最大传输字节数前进行接收，最后接收字节的数据字段奇偶校验位不是偶校验时 ［清除条件］ • 写 1 时

【注】 * 为清除标志只能写 1。

20.3.19　IEBus 接收中断允许寄存器 （IEIER）

IEIER 能设定禁止 / 允许 IERSR 的接收忙、接收开始、接收正常结束、接收错误结束等中断源。

位：	7	6	5	4	3	2	1	0
	RXBYSE	RXSE	RXFE	RXEDEE	RXE OVEE	RXE RTMEE	RXE DLEE	RXEPEE
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	RXBSYE	0	R/W	允许接收忙中断 设定接收忙 （RXBSY）中断的禁止 / 允许。 0：禁止接收忙 （RXBSY）中断 1：允许接收忙 （RXBSY）中断
6	RXSE	0	R/W	允许接收开始中断 设定接收开始 （RXS）中断的禁止 / 允许。 0：禁止接收开始 （RXS）中断 1：允许接收开始 （RXS）中断
5	RXFE	0	R/W	允许接收正常结束中断 设定接收正常结束 （RXF）中断的禁止 / 允许。 0：禁止接收正常结束 （RXF）中断 1：允许接收正常结束 （RXF）中断
4	RXEDEE	0	R/W	允许广播接收错误中断 设定广播接收错误 （RXEDE）中断的禁止 / 允许。 0：禁止广播接收错误 （RXEDE）中断 1：允许广播接收错误 （RXEDE）中断
3	RXEOVEE	0	R/W	允许溢出控制标志中断 设定溢出控制标志 （RXEOVE）中断的禁止 / 允许。 0：禁止溢出控制标志 （RXEOVE）中断 1：允许溢出控制标志 （RXEOVE）中断
2	RXERTMEE	0	R/W	允许接收时序错误中断 设定接收时序错误 （RXERTME）中断的禁止 / 允许。 0：禁止接收时序错误 （RXERTME）中断 1：允许接收时序错误 （RXERTME）中断
1	RXEDLEE	0	R/W	允许接收帧最大传输字节数溢出中断 设定接收帧最大传输字节数溢出 （RXEDLE）中断的禁止 / 允许。 0：禁止接收帧最大传输字节数溢出 （RXEDLE）中断 1：允许接收帧最大传输字节数溢出 （RXEDLE）中断
0	RXEPEE	0	R/W	允许奇偶校验错误中断 设定奇偶校验错误 （RXEPE）中断的禁止 / 允许。 0：禁止奇偶校验错误 （RXEPE）中断 1：允许奇偶校验错误 （RXEPE）中断

20.3.20　IEBus 时钟选择寄存器 （IECKSR）

IECKSR 是 8 位可读 / 写寄存器。设定 IEB 使用的时钟。

位：	7	6	5	4	3	2	1	0
	-	-	-	CKS3	-	CKS[2:0]		
初始值：	0	0	0	0	0	0	0	1
R/W：	R	R	R	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说　明
7 ～ 5	—	均为 0	R	保留位 读写的值总是 0。
4	CKS3	0	R/W	输入时钟选择 3*1*2 选择 IEB 使用的时钟。 0：使用外围时钟 （pϕ）。 1：使用 AUDIO_X1、 AUDIO_X2 的时钟。
3	—	0	R	保留位 读写的值总是 0。
2 ～ 0	CKS[2:0]	001	R/W	输入时钟选择 2 ～ 0*1 选择 IEB 使用的时钟的分频。 000：禁止设定 001：使用通过 CKS3 选择的时钟 （IEBϕ）的 1/2 时钟。 （IEBϕ=12MHz、 12.58MHz） 010：使用通过 CKS3 选择的时钟 （IEBϕ）的 1/3 时钟。 （IEBϕ=18MHz、 18.87MHz） 011：使用通过 CKS3 选择的时钟 （IEBϕ）的 1/4 时钟。 （IEBϕ=24MHz、 25.16MHz） 100：使用通过 CKS3 选择的时钟 （IEBϕ）的 1/5 时钟。 （IEBϕ=30MHz、 31.45MHz） 101：使用通过 CKS3 选择的时钟 （IEBϕ）的 1/6 时钟。 （IEBϕ=36MHz、 37.74MHz） 110：禁止设定 111：禁止设定

【注】 *1　IEBus 发送 / 接收运行中，不能变更 CKS3 ～ CKS0 位的设定。

*2　CKS3 位设定为 1 时，必须将 STBCR3 寄存器的 MSTP36 位设定为 0。
关于STBCR3寄存器的设定方法请参照 “第32章 低功耗模式”。

20.3.21　IEBus 发送数据缓冲器 001 ～ 128 （IETB001 ～ IETB128）

主发送时，写入发送数据的 128 字节 （8×128）缓冲器。

IETB001 ～ IETB128 的初始值不定。

位：	7	6	5	4	3	2	1	0
	TBn							
初始值：	不定	不定	不定	不定	不定	不定	不定	不定
R/W：	W*	W*	W*	W*	W*	W*	W*	W*

[符号说明]　n=001~128

位	位名	初始值	R/W	说　明
7 ～ 0	TBn	不定	W*	IEBus 发送数据缓冲器 主发送时，将在数据字段发送的数据写入 TB001~TB128。按发送的顺序从 TB001 开始 （开始数据为 1 字节），后面是 TB002， TB003…的顺序写入。TB128 为 128 字节发送时的最后数据。

【注】 *　主发送中 （IEFLG 的 MRQ=1）时，禁止写入。

20.3.22　IEBus 接收数据缓冲器 001 ～ 128 （IERB001 ～ IERB128）

从属接收时，保存发送数据的 128 字节 （8×128）缓冲器。

IERB001 ～ IERB128 的初始值不定。

位：	7	6	5	4	3	2	1	0
	RBn							
初始值：	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*

[符号说明]　n=001～128

位	位名	初始值	R/W	说　明
7 ～ 0	RBn	不定	R*	IEBus 接收数据缓冲器 IEBus接收状态寄存器（IERSR）的RXBSY位为1时，能读取RB001~RB128。从 RB001~RB128 读出的数据为从属接收时数据字段的值。按 RB001 （开始数据为 1 字节）、RB002、RB003…的顺序写入接收时的数据。RB128 为 128 字节接收时的最后数据。

【注】 *　从属接收中为 （IEFLG 的 SRE=1 且 IERSR 的 RXBSY=0）时，禁止读出。（读取时值不定）

20.4　数据格式

20.4.1　　发送格式

IEBus 数据发送时的传输格式和各寄存器的关系如图 20.6 所示。

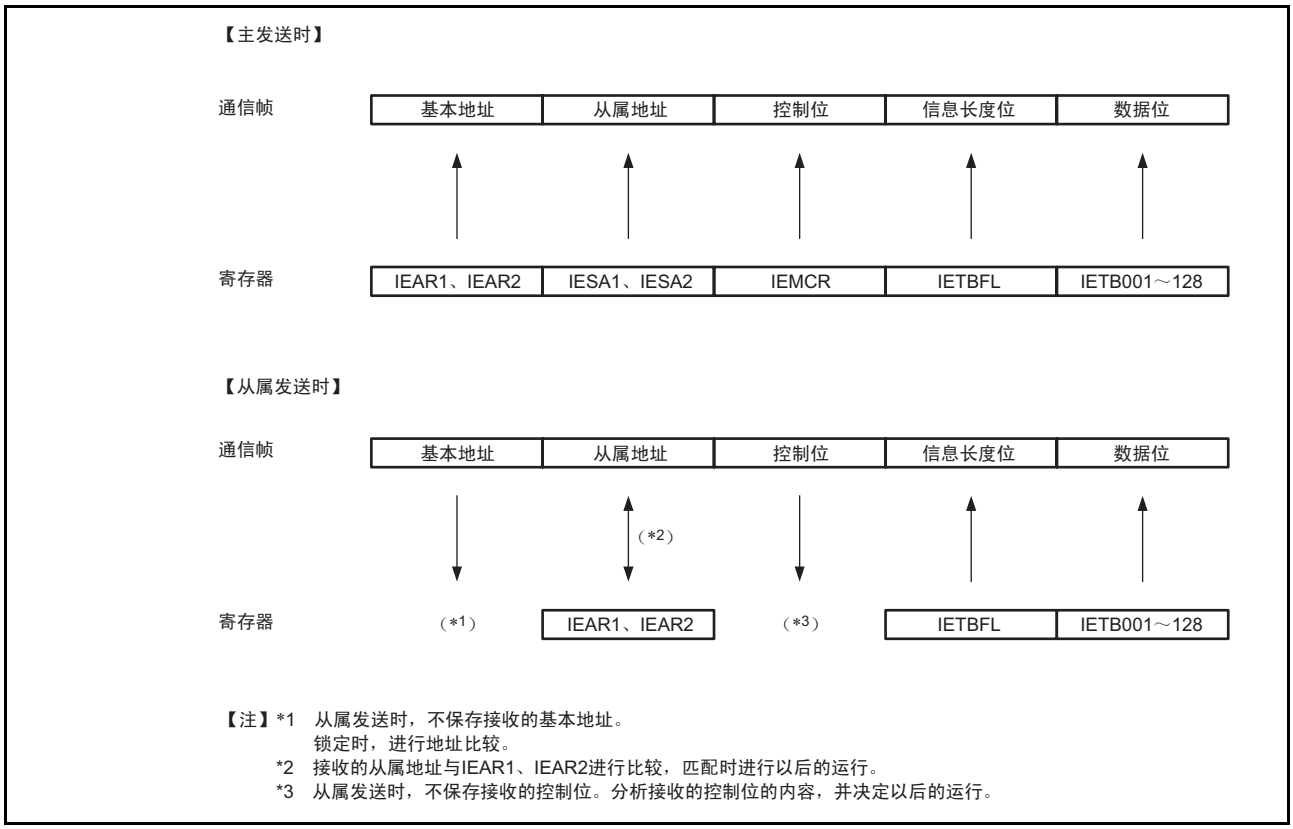


图 20.6　发送时的传输信号格式和各寄存器的关系

20.4.2　接收格式

IEBus 数据接收时的传输格式和各寄存器的关系如图 20.7 所示。

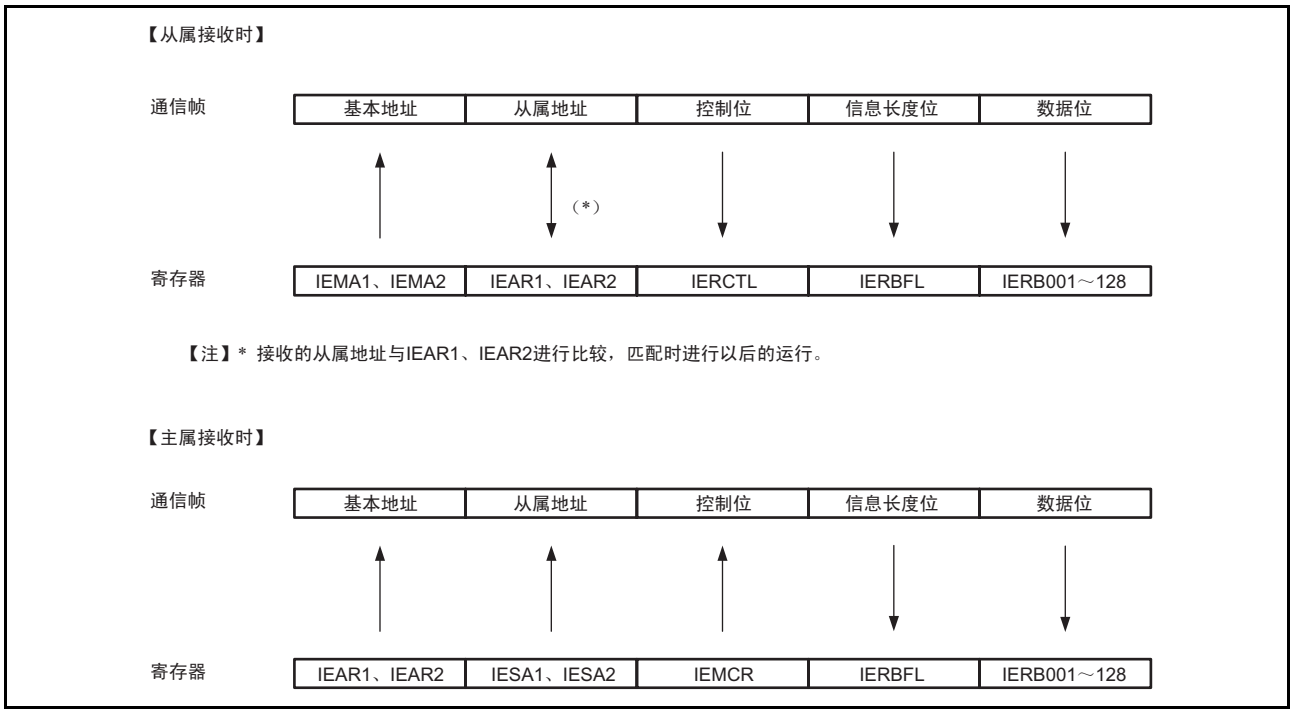
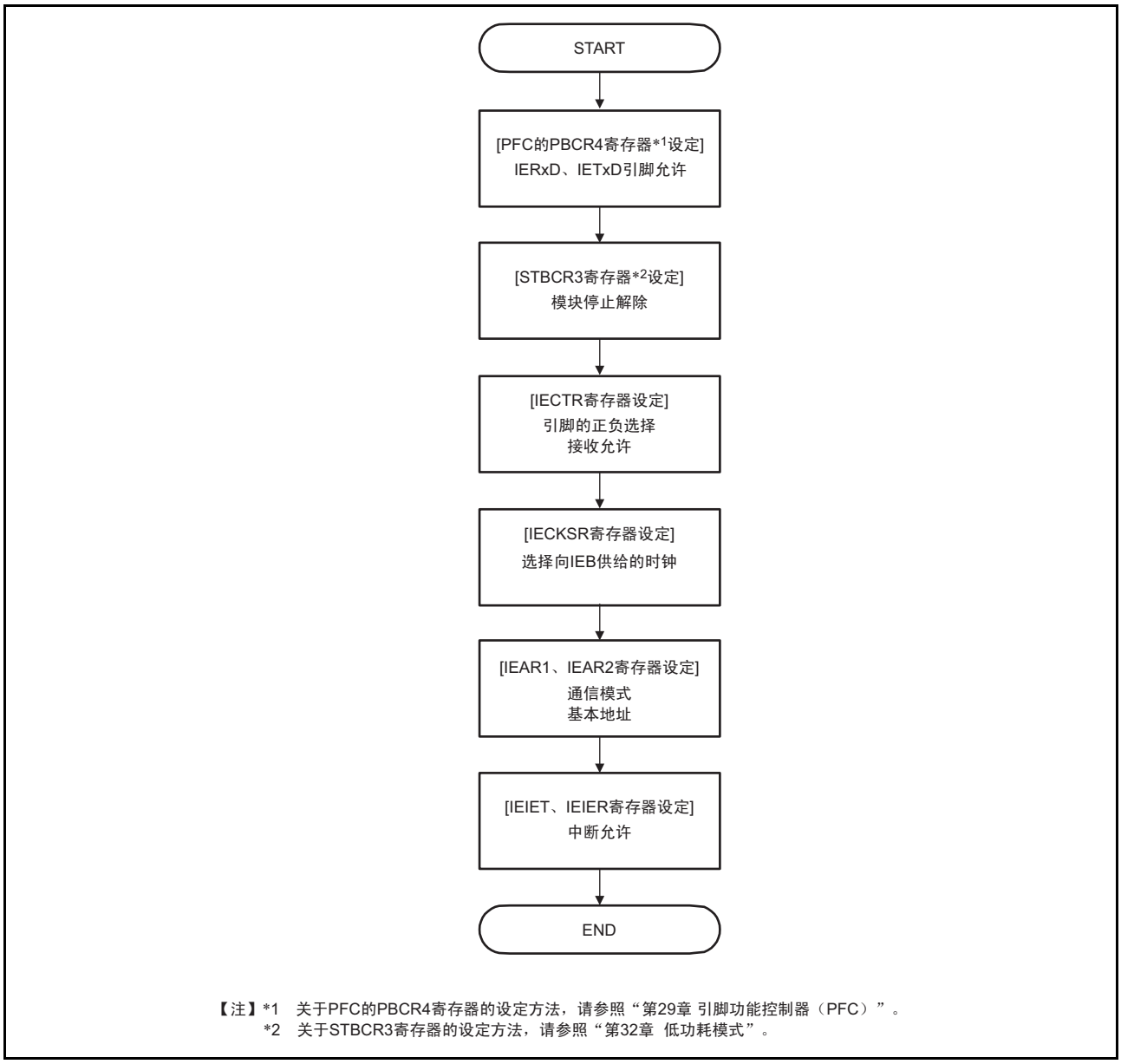


图 20.7　接收时的传输信号格式和各寄存器的关系

20.5　软件控制流程

20.5.1　初始设定

初始设定流程如图 20.8 所示。



【注】*1　关于PFC的PBCR4寄存器的设定方法，请参照“第29章 引脚功能控制器（PFC）”。
*2　关于STBCR3寄存器的设定方法，请参照“第32章　低功耗模式”。

图 20.8　初始设定流程

20.5.2　主发送

主发送流程如图 20.9 所示。

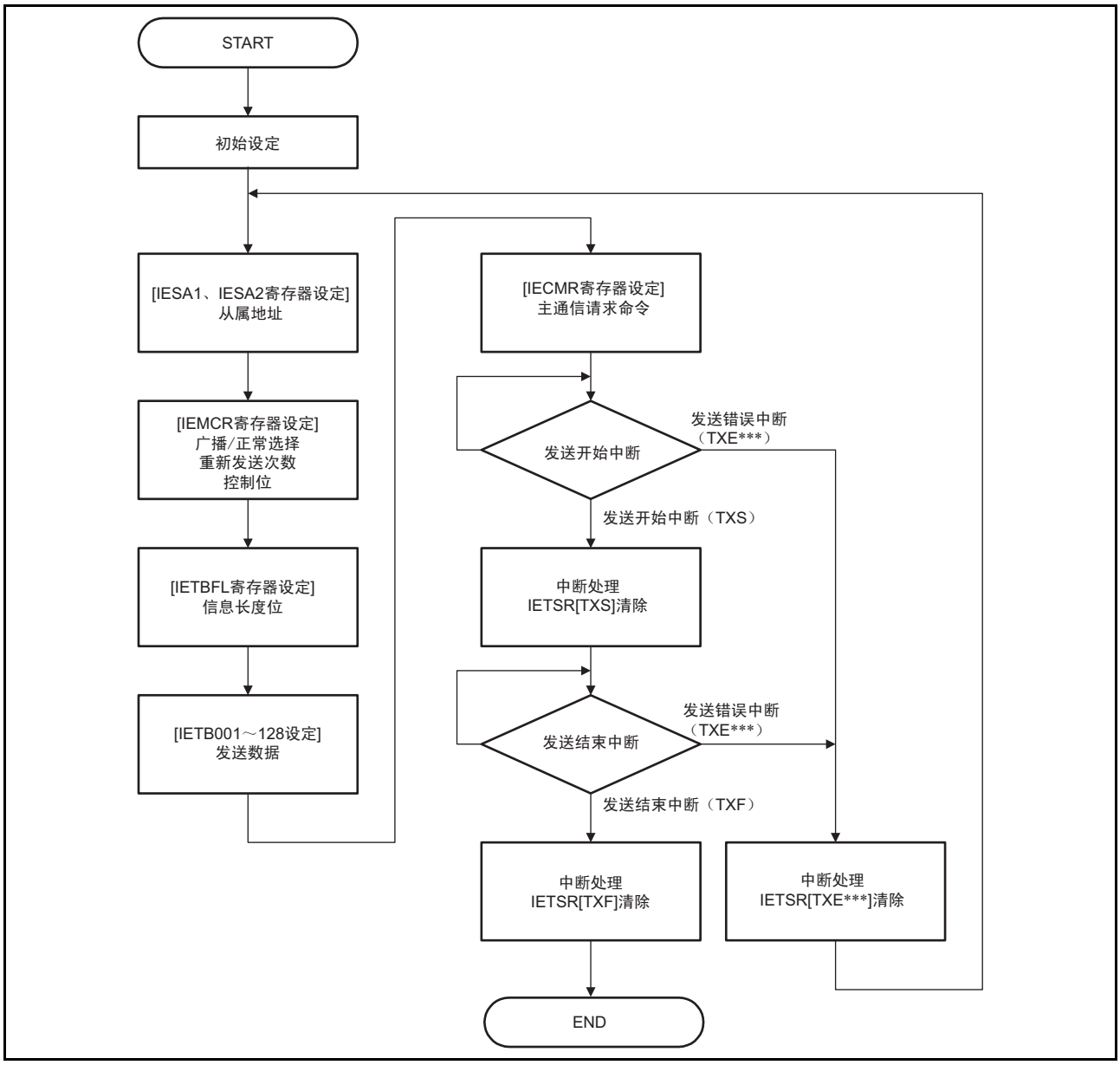


图 20.9　主发送流程

20.5.3　从属接收

从属接收流程如图 20.10 所示。

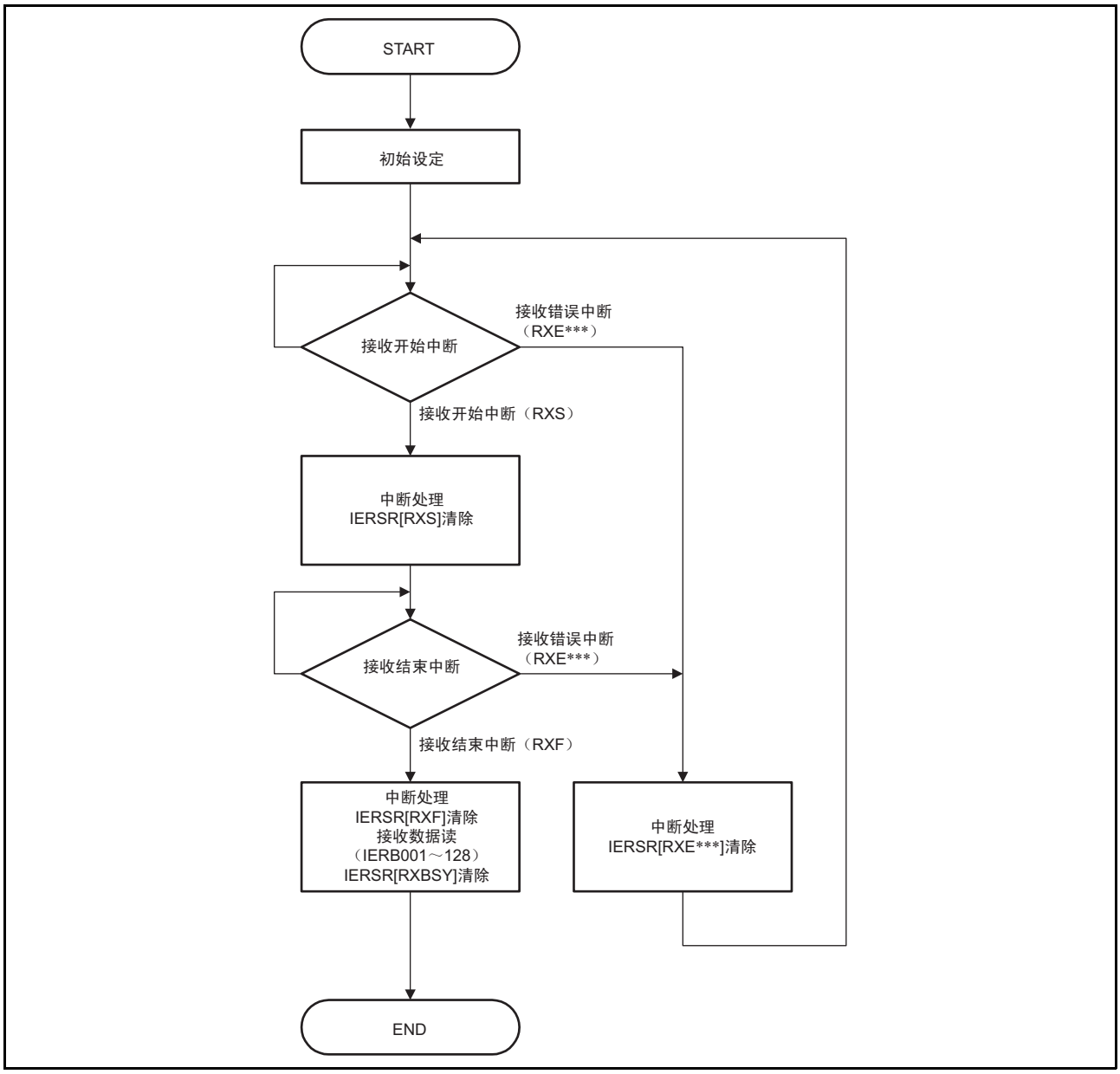


图 20.10　从属接收流程

20.5.4　主接收

主接收流程如图 20.11 所示。

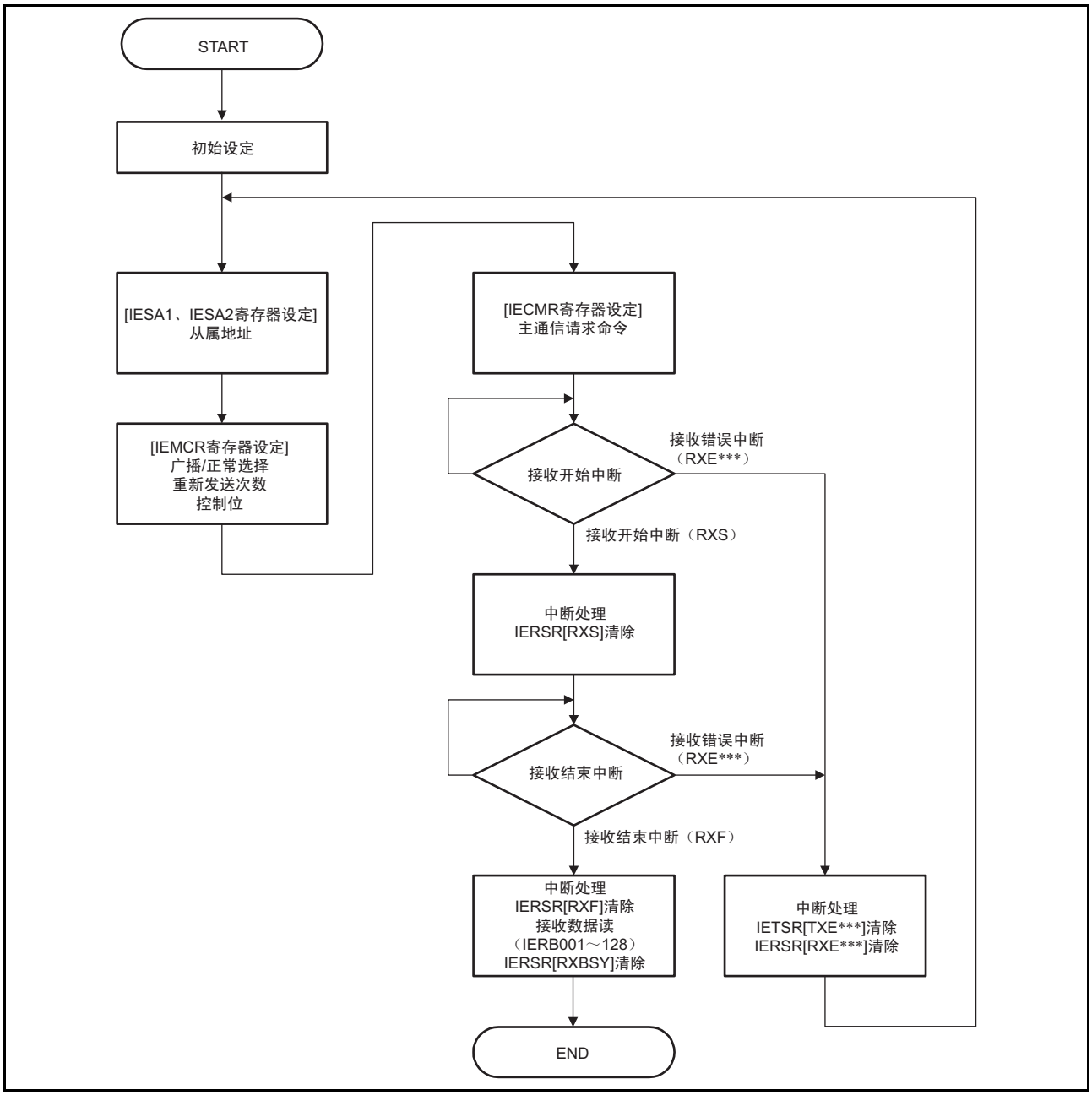


图 20.11　主接收流程

20.5.5　从属发送

从属发送如图 20.12 所示。

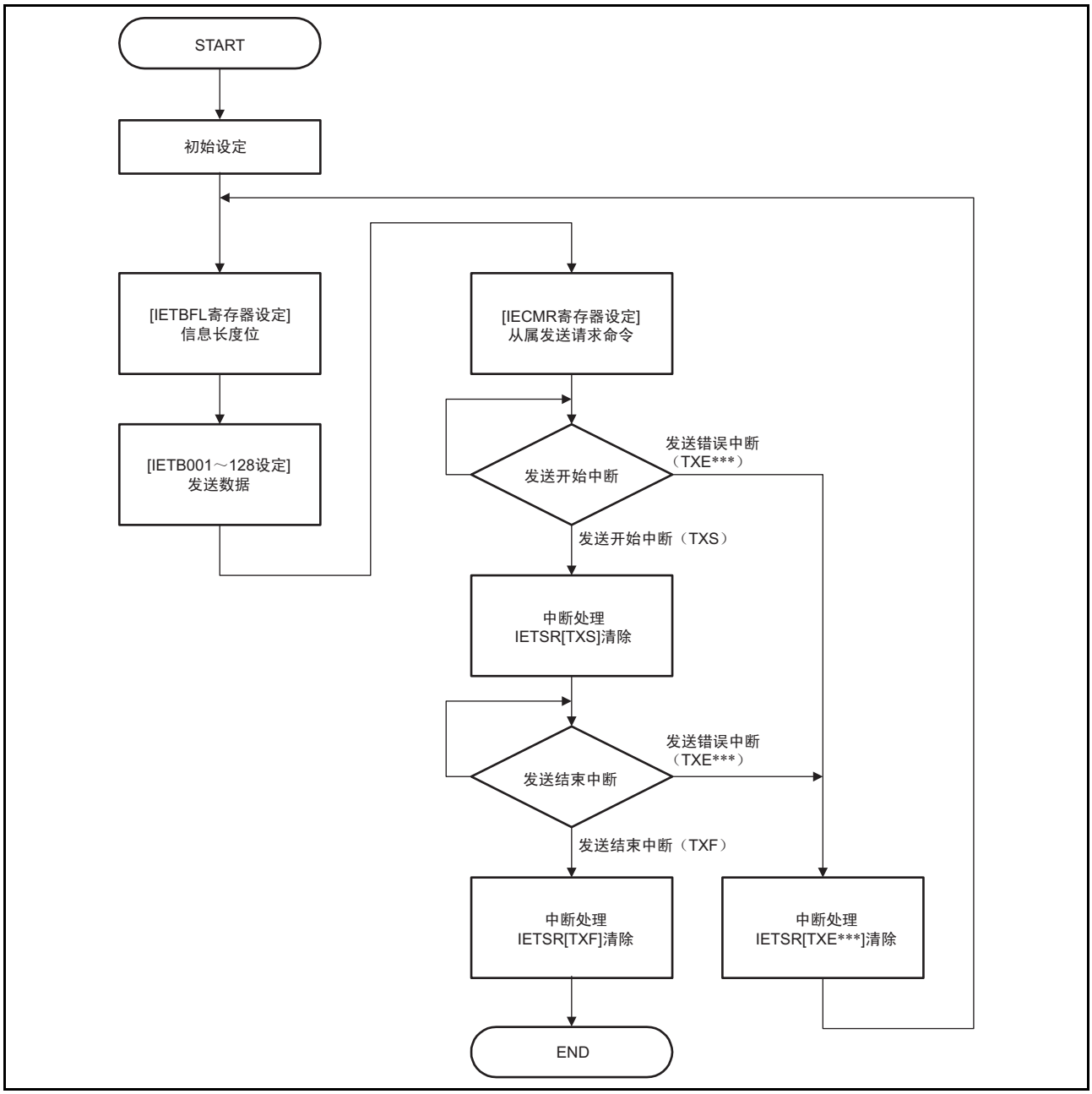


图 20.12　从属发送流程

20.6　运行时序

20.6.1　主发送

主发送运行时序如图 20.13 所示。

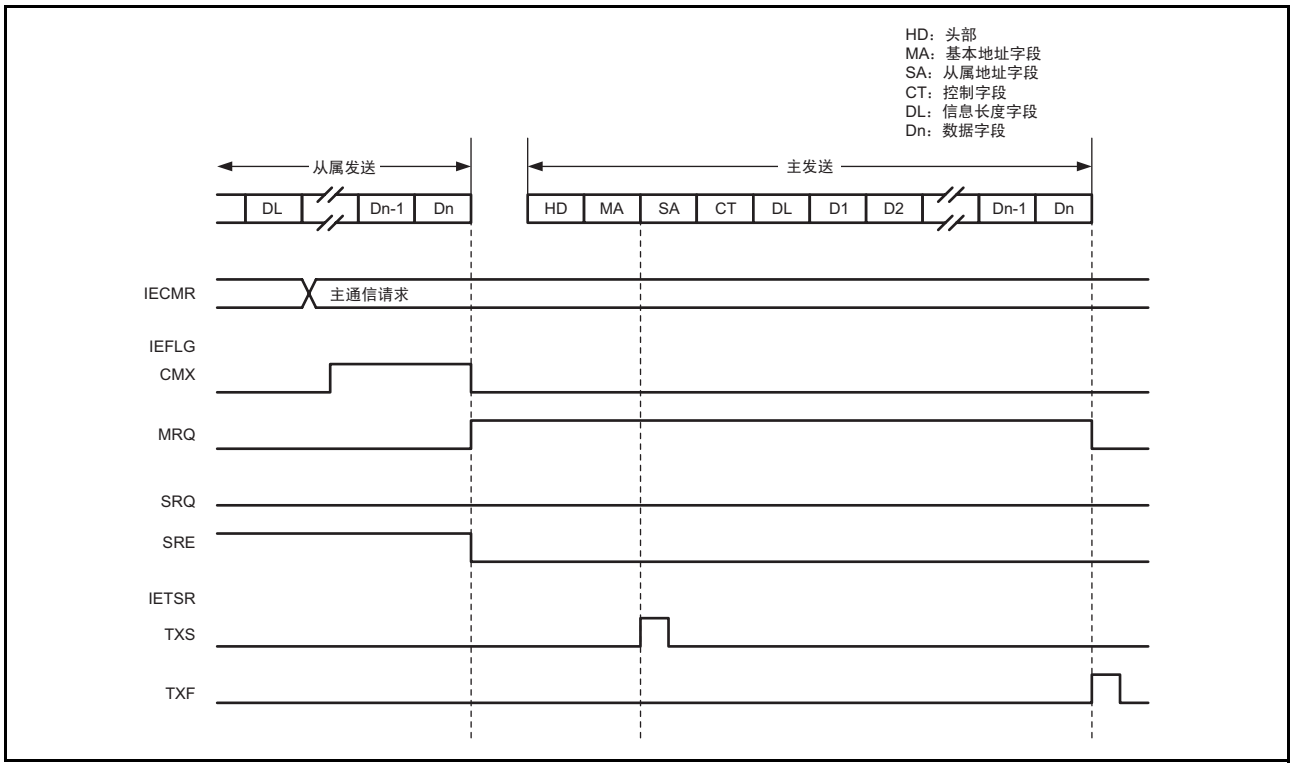


图 20.13　主发送运行时序

20.6.2　从属接收

从属接收运行时序如图 20.14 所示。

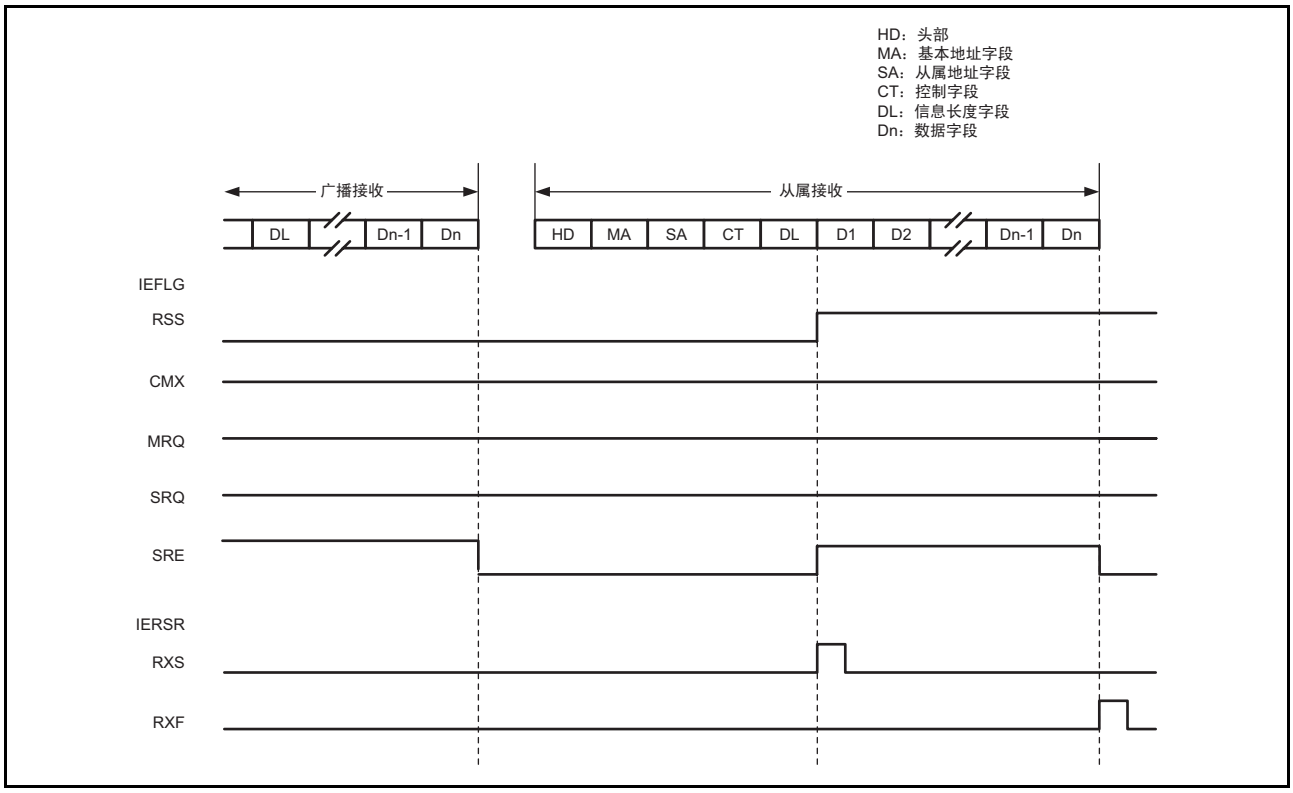


图 20.14　从属接收运行时序

20.6.3　主接收

主接收运行时序如图 20.15 所示。

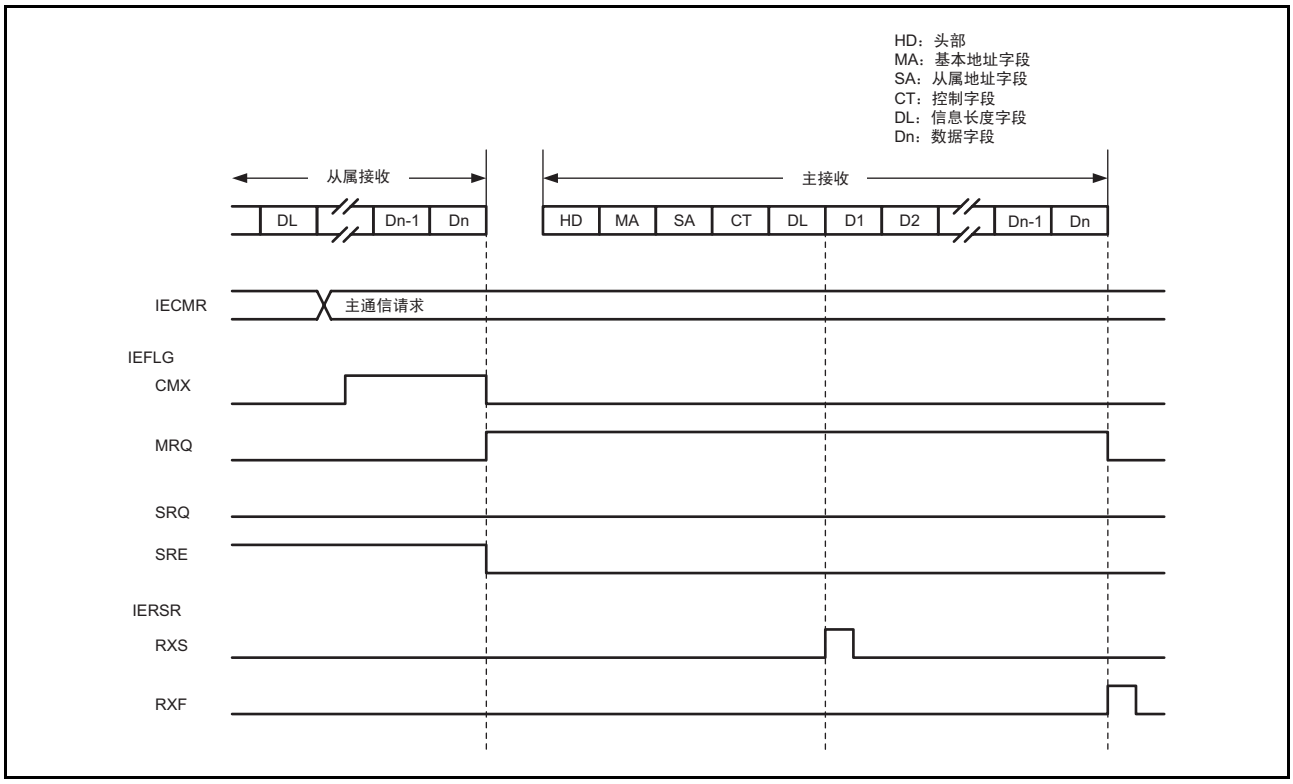


图 20.15　主接收运行时序

20.6.4　从属发送

从属发送运行时序如图 20.16 所示。

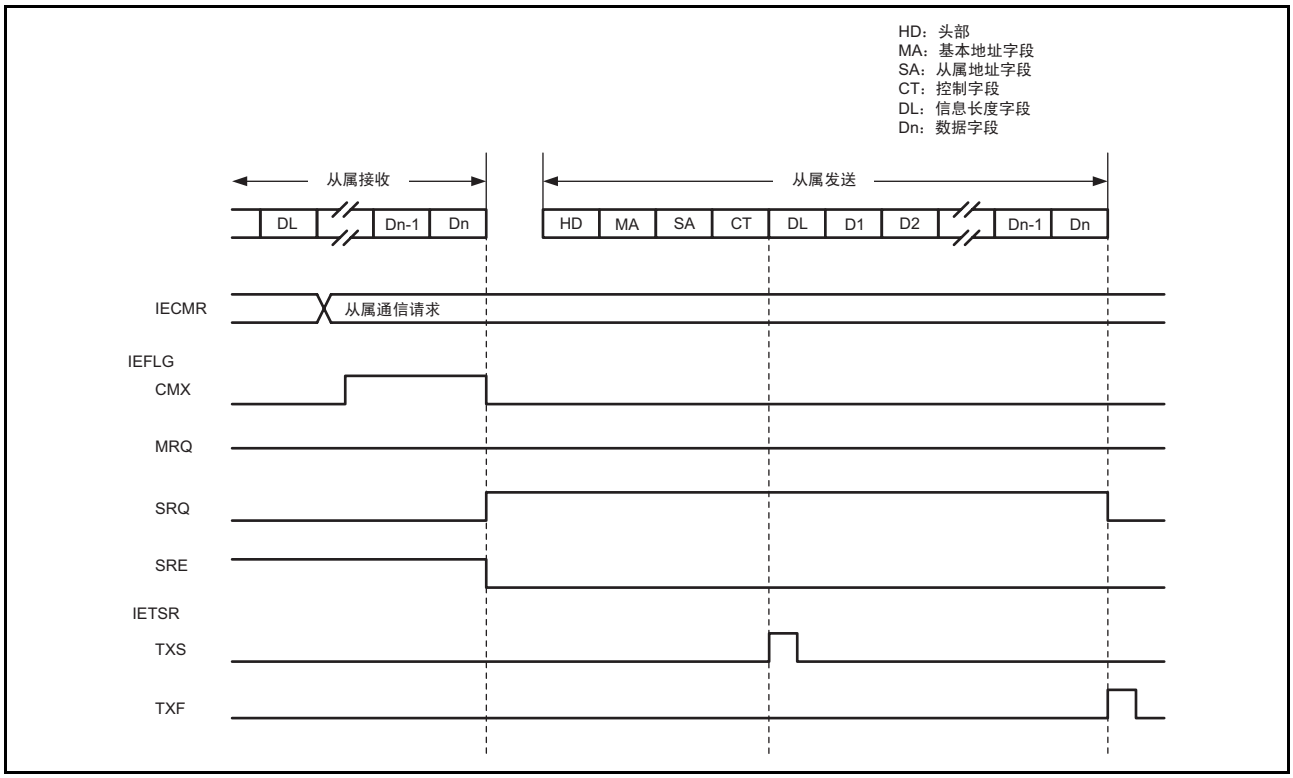


图 20.16　从属发送运行时序

20.7　中断源

IEB 的中断有，发送开始 （TXS）、发送正常结束 （TXF）、仲裁失败 （TXEAL）、发送时序错误 （TXETTME）、发送帧最大传输字节数溢出 （TXERO）、应答位 （TXEACK）、接收忙 （RXBSY）、接收开始 （RXS）、接收正常结束 （RXF）、广播接收错误 （RXEDE）、接收溢出标志 （RXEOVE）、接收时序错误 （RXERTME）、接收帧最大传输字节数溢出 （RXEDLE）、奇偶校验错误 （RXEPE）。

各源拥有对应 IEBus 发送中断允许寄存器 （IEIET）、IEBus 接收中断允许寄存器 （IEIER）的位，能设定中断的禁止 / 允许。而且还拥有对应 IEBus 发送状态寄存器 （IETSR）、IEBus 接收状态寄存器 （IERSR）的状态标志，可通过读取状态标志判定中断源。

IEB 中断源的关系如图 20.17 所示。

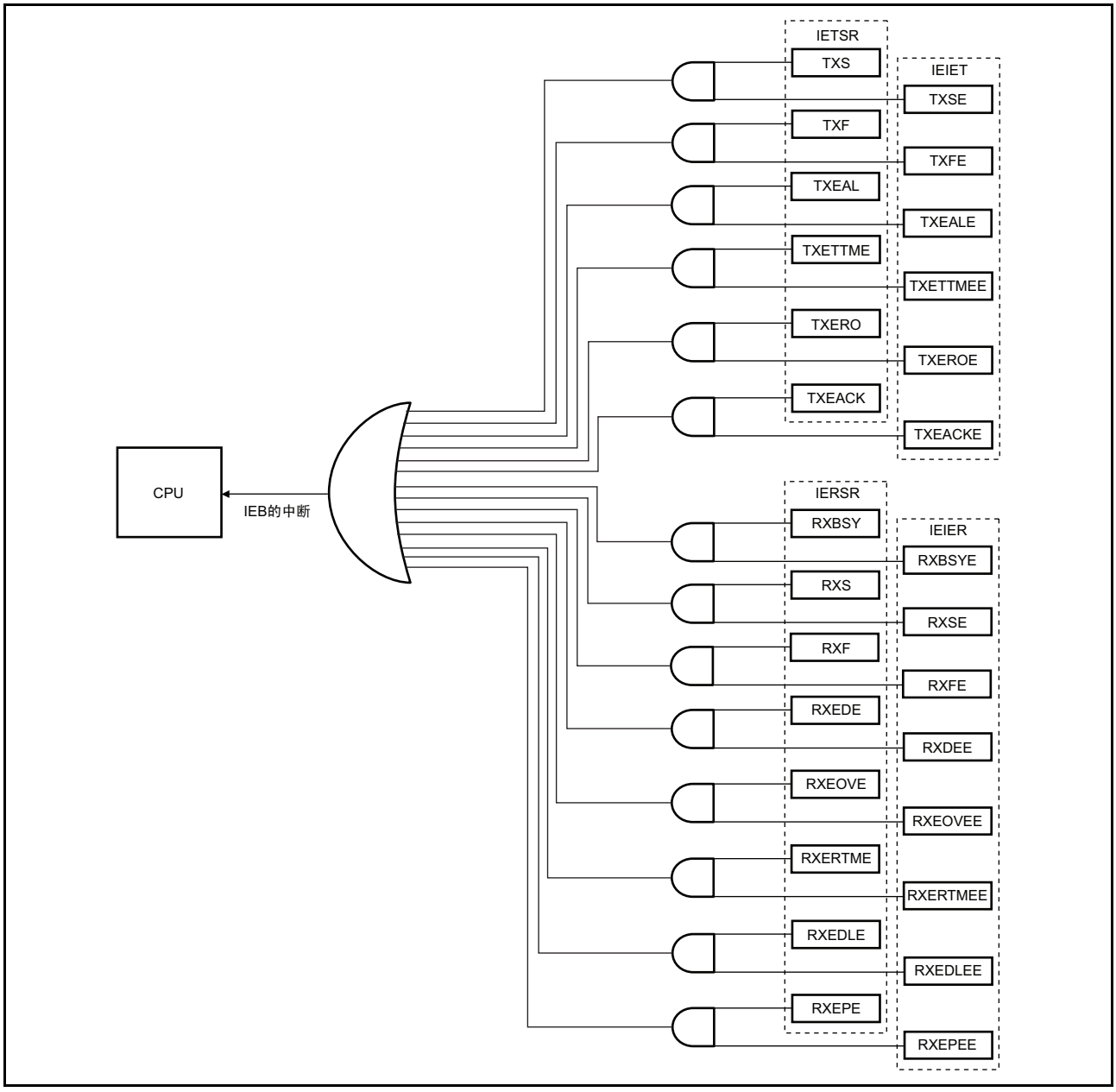


图 20.17　IEB 中断源的关系

20.8　使用时的注意事项

20.8.1　　最大传输字节长度内，通信没结束时的注意事项

（1） 数据发送

表示数据发送时，由于从接收单元接收 NAK，因此已经发送了通过通信模式定义的最大字节长度；或是由于信息长度的值大于最大传输字节数的值，因此在最大字节长度内发送没有结束时，设置 IETSR 的错误标志并进入待机状态。但是此时，发送第 （最大传输字节数 +1）字节。此后，第 （最大传输字节数 +1）字节的应答位接收 NAK 时， TXERO 标志被置位。接收的不是 NAK 而是 ACK 时， TXF 标志被置位。

最大字节长内发送没结束时的运行时序如图 20.18 所示。

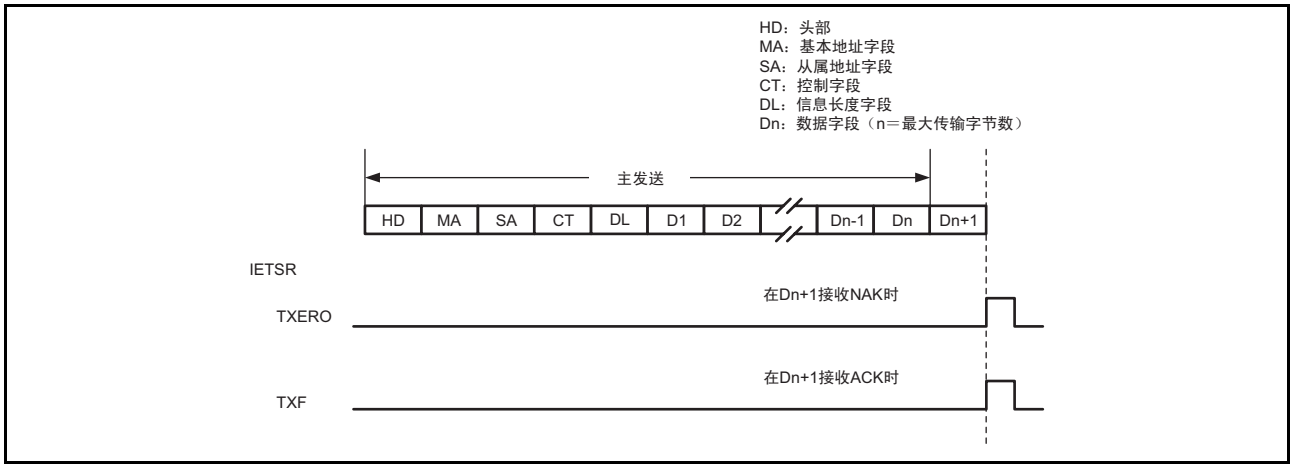


图 20.18　最大字节长内发送没结束时的运行时序

（2） 数据接收

数据接收时，发生奇偶校验错误或者溢出错误，并进行由重新发送引起的接收，因此在通过通信模式定义的最大字节长内接收没结束；或者由于信息长度的值大于最大传输字节数的值，因此接收没结束时，设置 IERSR 的错误标志并进入待机状态。但此时为第（最大传输字节数 +1）字节的数据接收等待状态。因此，不能接收第（最大传输字节数 +1）字节的数据时，检测出接收时序错误，RXERTME 标志被置位。此时，RXEDLE 标志不被置位。如果接收第 （最大传输字节数 +1）字节的数据， RXEDLE 标志就被置位。

接收最大传输字节长，没有解除奇偶校验错误时也同样，不能接收第 （最大传输字节数 +1）字节的数据时， RXERTME 标志被置位。此时， RXEPE 标志不被置位。接收第 （最大传输字节数 +1）字节的数据时，RXEPE 标志被置位。

最大字节长内接收没结束时的运行时序如图 20.19 所示。

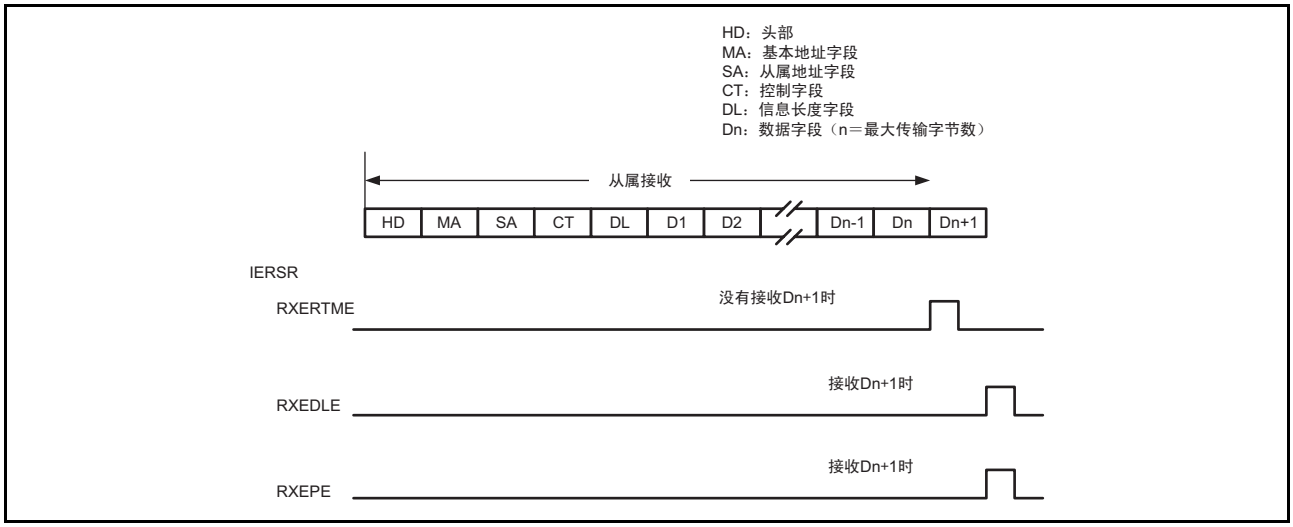


图 20.19　最大字节长内接收没结束时的运行时序

第 21 章　CD-ROM 解码器 （ROM-DEC）

CD-ROM 解码器 （ROM-DEC）进行来自 CD-DSP 的数据流的 ROM 解码处理。 CD-DA*1 时，数据流变为 PCM 数据，因此不输入 CD-ROM 解码器。 CD-ROM*2 时，输入数据流，进行同步码检测 / 保护、解扰、 ECC 校正、 EDC 校检，然后输出数据流。

但前提是数据流经过 SSI 传送 CD-ROM 数据。因此不进行 CD 规格中子代码的处理。

【注】 *1 依据 JIS S 8605 （Red Book）
*2 依据 JIS X 6281 （Yellow Book）

21.1　特点

- 同步码检测/保护
  检测CD-ROM同步码。另外，因划伤等检测不出同步码时，保护同步码 （自动插入）。
  保护的种类有自动同步保护模式、外部同步保护模式、内插同步模式及内插/外部同步保护模式。
- 解扰
- ECC校正
  可进行P校正、 Q校正、 PQ校正、 QP校正。
  可重复进行PQ校正、QP校正，最多能校正3次。但校正次数依据CD的倍速，例如，如果CD-ROM解码器的运行频率为60MHz，且为CD2倍速，就最多能重复校正3次。
  ECC校正有2面的缓冲器， 1面接收数据流的同时，另1面进行ECC校正。
- EDC校检
  在ECC校正前后进行EDC的校验。当ECC前的EDC校检没有错误时，与伴随式计算结果无关，也有不进行ECC校正的模式。
- 缓冲数据控制
  CD-ROM解码器通过将同步代码置于起始的特定格式向缓冲区输出数据。

21.1.1　ROM-DEC 格式

本 LSI 的 CD-ROM 解码器支持图 21.1 中的 5 种格式。



图 21.1　ROM-DEC 格式

21.2　框图

为实现 CD-ROM 解码功能的本 LSI 的 CD-ROM 解码器功能和连接外围总线的总线桥如图 21.2 所示。

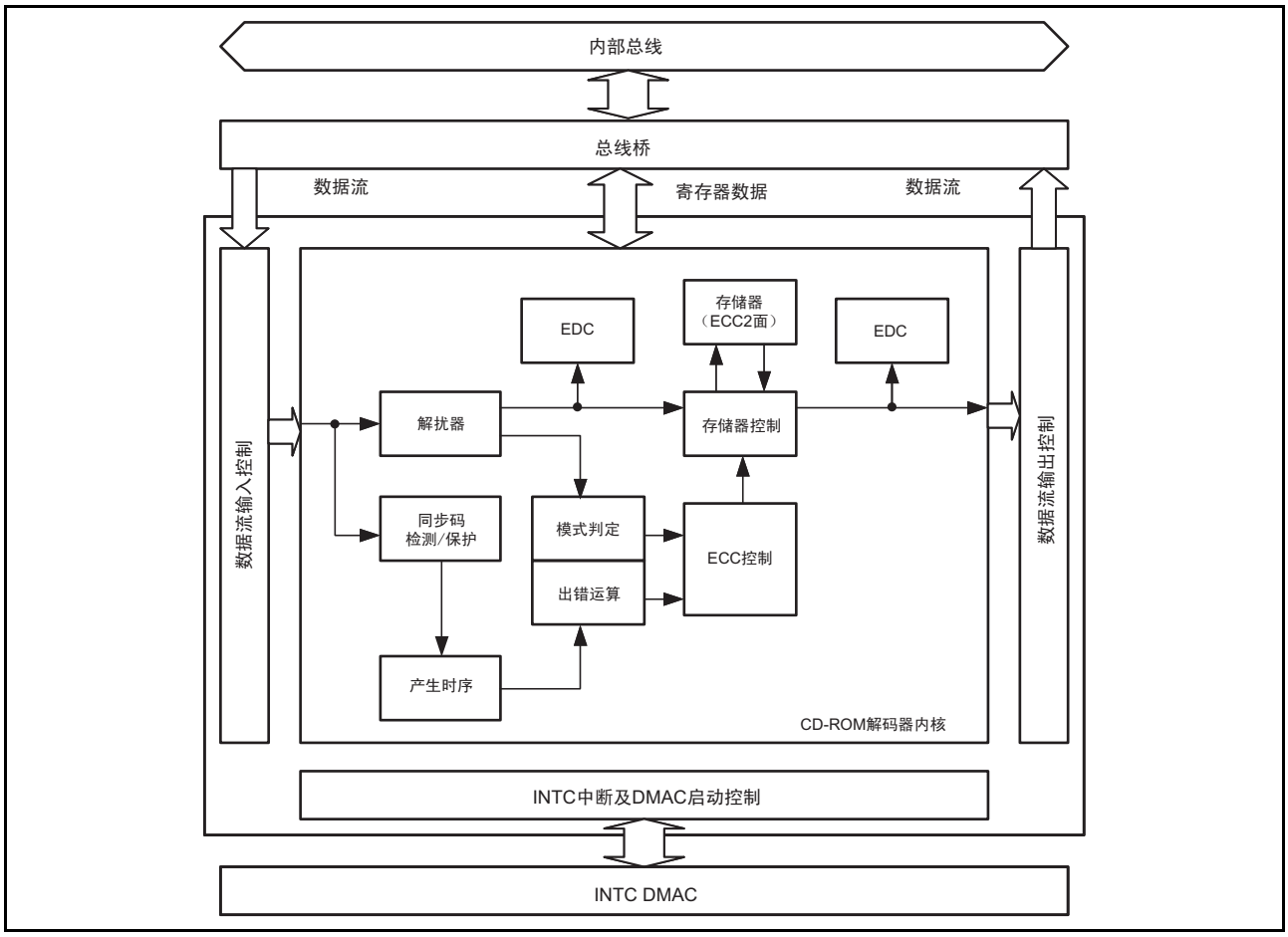


图 21.2　ROM-DEC 的框图

CD-ROM 解码器的内核部分要执行 CD-ROM 解码，就要运行所需的解扰功能、同步码检测、ECC（P、Q 校正）功能、 EDC 校检功能。存储器有 2 个扇区。

来自内部总线数据的输入 / 输出各有 1 个系统，通过总线桥逻辑，分之为寄存器存取端口和数据流端口。

来自 CD-DSP 的数据流经过 SSI 传送到数据流输入控制块。通过 CD-ROM 解码器，在解扰、 ECC 校正、EDC 校检后，得到 1 个扇区的数据。之后，经过数据流输出控制块，传送到数据流缓冲器。通过 DMAC 或 CPU 传送数据流。

总线桥的框图如**图 21.3** 所示。

由于从 SSI 传送输入数据流，因此传送缓慢，但是由于输出数据流已存在于 CD-ROM 解码器内核内，因此可进行高速传送。另外，由于输出数据流在 SDRAM 中缓冲，为降低 SDRAM 的繁忙率，需要进行高速传送。因此，来自内部总线的输出数据流读出请求到达 CD-ROM 解码器前，预先读数据，将数据流存储在总线桥内的寄存器，如果有来自内部总线的请求，立即成为可向内部总线输出数据的状态。因此，有时也会在已经预先读取输出数据流，并保存到寄存器的状态下，接收到其他寄存器读出请求。为此， ROM-DEC 分别拥有用于输出数据流的寄存器和用于读其他寄存器的中继寄存器。

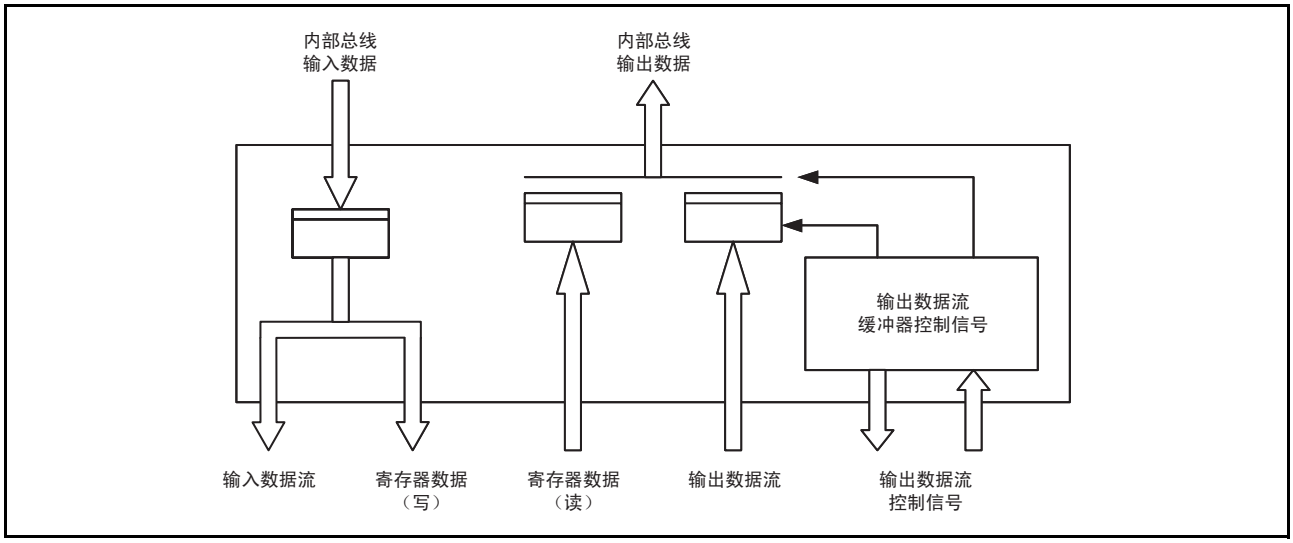


图 21.3　总线桥的框图

数据流输入控制框图如图 21.4 所示。数据流输入控制部分有输入数据流的控制逻辑和改变 CD-ROM 解码器的控制模式的寄存器。

从 SSI 传送来的数据，由于 SSI 的模式不同字节序也有所不同，传送顺序会发生变化，或者会传送填充数据。为对应这些各种各样的数据，预备有改变运行模式的寄存器或生成控制 CD-ROM 解码器内核部分的控制信号。输入数据流保持寄存器有 16 位的 2 面寄存器。通过寄存器设定的模式，可先向 CD-ROM 解码器内核部供给先从 SSI 传送来的 16 位数据，或者先向 CD-ROM 解码器内核部供给后传送来的 16 位数据。同时，也可不向 CD-ROM 解码器内核部份供给填充数据。

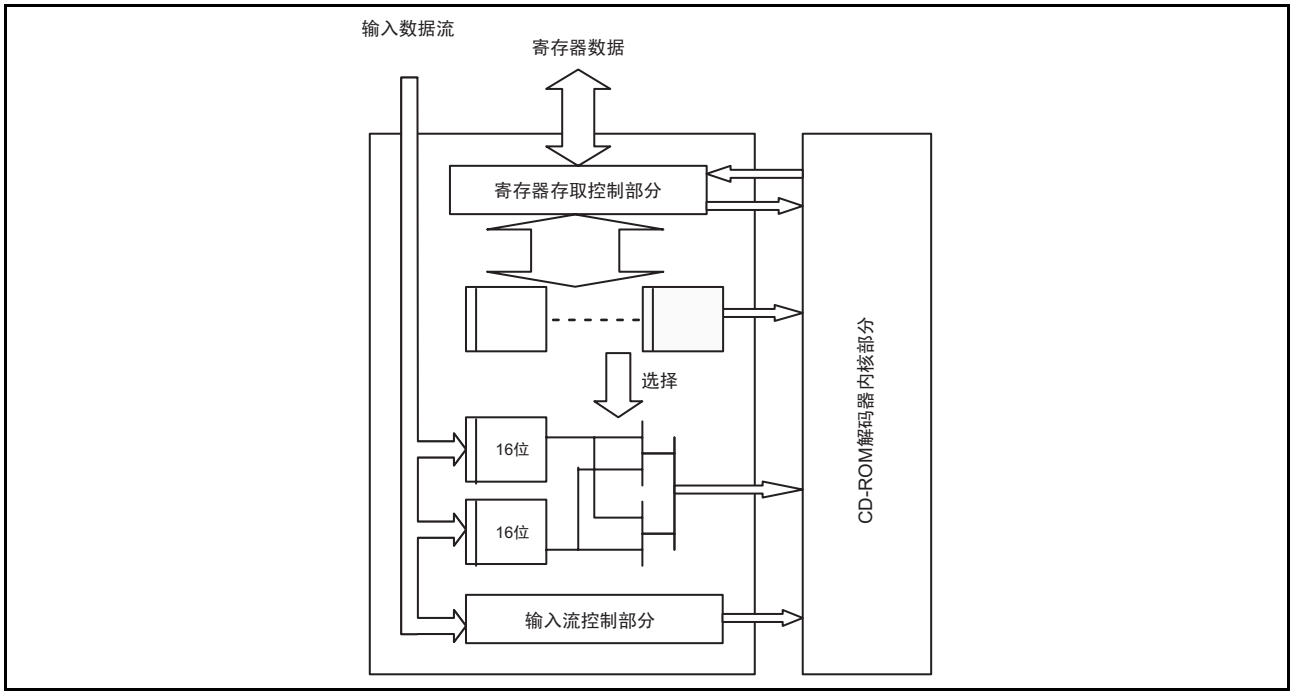


图 21.4　数据流输入控制框图

数据流输出控制框图如图 21.5 所示。

识别 1 扇区的 CD-ROM 数据已存在于 CD-ROM 解码器内核，确认存在于总线桥的输出数据流寄存器为空，从 CD-ROM 解码器内核获取输出数据流。

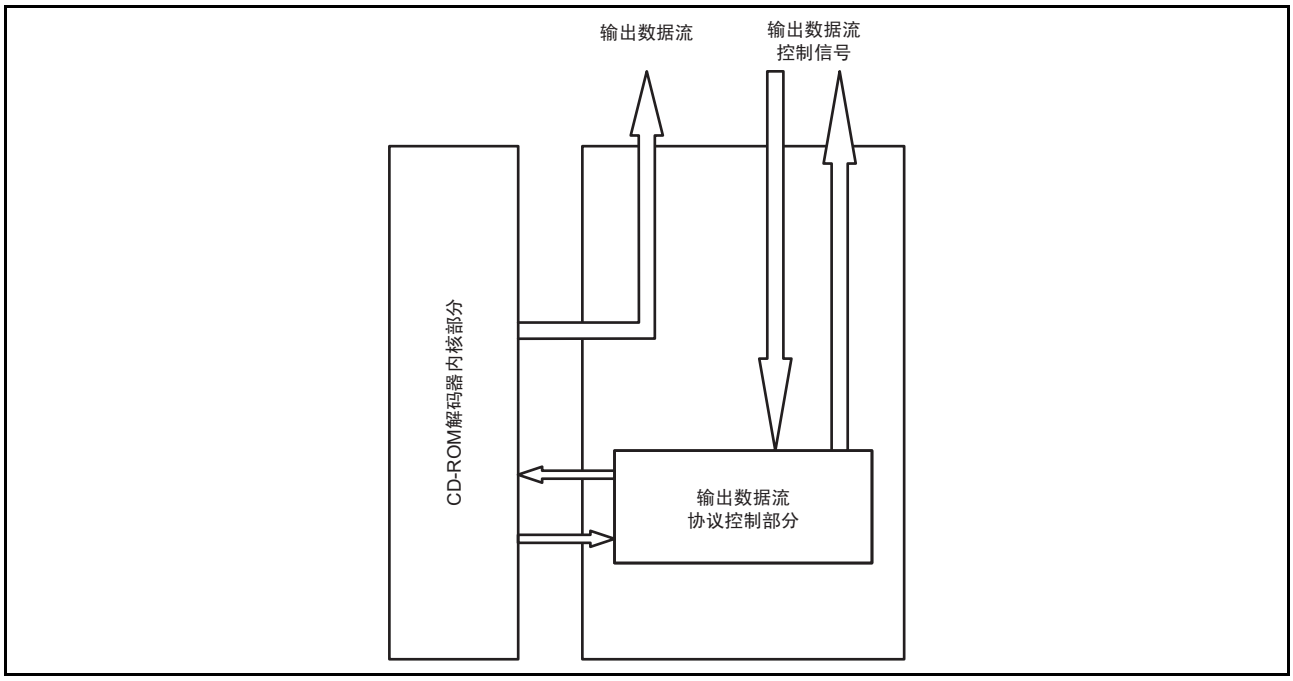


图 21.5　数据流输出控制框图

在 INTC 中断及 DMAC 启动控制中，具有以下功能：

中断的保留或标志清除

中断的屏蔽

DMAC 的启动信号断言

检测传送数据量、启动信号取反

21.3　寄存器的说明

ROM-DEC 有以下的寄存器。

表 21.1　寄存器结构

寄存器名	简称	R/W	初始值	地址	存取长度
ROMDEC 允许控制寄存器	CROMEN	R/W	H'00	H'FFFC2000	8
同步码的同步控制寄存器	CROMSY0	R/W	H'89	H'FFFC2001	8
解码模式控制寄存器	CROMCTL0	R/W	H'82	H'FFFC2002	8
EDC、 ECC 校验的控制寄存器	CROMCTL1	R/W	H'D1	H'FFFC2003	8
解码处理自动停止控制寄存器	CROMCTL3	R/W	H'00	H'FFFC2005	8
解码选项设定控制寄存器	CROMCTL4	R/W	H'00	H'FFFC2006	8
HEAD20 ～ 22 显示控制寄存器	CROMCTL5	R/W	H'00	H'FFFC2007	8
同步码状态寄存器	CROMST0	R	H'00	H'FFFC2008	8
ECC 后的扇区头对应的错误状态寄存器	CROMST1	R	H'00	H'FFFC2009	8
ECC 后的子头对应的错误状态寄存器	CROMST3	R	H'00	H'FFFC200B	8
扇区头、子头数据妥当性判定状态寄存器	CROMST4	R	H'00	H'FFFC200C	8
模式判定结果和连接扇区检测状态寄存器	CROMST5	R	H'00	H'FFFC200D	8
ECC、 EDC 错误状态寄存器	CROMST6	R	H'00	H'FFFC200E	8
缓冲状态寄存器	CBUFST0	R	H'00	H'FFFC2014	8
解码中止源状态寄存器	CBUFST1	R	H'00	H'FFFC2015	8
缓冲溢出状态寄存器	CBUFST2	R	H'00	H'FFFC2016	8
ECC 校正前扇区头 -MINUTES 数据寄存器	HEAD00	R	H'00	H'FFFC2018	8
ECC 校正前扇区头 -SECONDS 数据寄存器	HEAD01	R	H'00	H'FFFC2019	8
ECC 校正前扇区头 -FRAMES （1/75 秒）数据寄存器	HEAD02	R	H'00	H'FFFC201A	8
ECC 校正前扇区头 -MODE 数据寄存器	HEAD03	R	H'00	H'FFFC201B	8
ECC 校正前子头 - 文件号码 （BYTE-16）数据寄存器	SHEAD00	R	H'00	H'FFFC201C	8
ECC 校正前子头 - 通道号码 （BYTE-17）数据寄存器	SHEAD01	R	H'00	H'FFFC201D	8
ECC 校正前子头 - 子模式 （BYTE-18）数据寄存器	SHEAD02	R	H'00	H'FFFC201E	8
ECC 校正前子头 - 数据类型 （BYTE-19）数据寄存器	SHEAD03	R	H'00	H'FFFC201F	8
ECC 校正前子头 - 文件号码 （BYTE-20）数据寄存器	SHEAD04	R	H'00	H'FFFC2020	8
ECC 校正前子头 - 通道号码 （BYTE-21）数据寄存	SHEAD05	R	H'00	H'FFFC2021	8
ECC 校正前子头 - 子模式 （BYTE-22）数据寄存器	SHEAD06	R	H'00	H'FFFC2022	8
ECC 校正前子头 - 数据类型 （BYTE-23）数据寄存器	SHEAD07	R	H'00	H'FFFC2023	8
ECC 校正后扇区头 -MINUTES 数据寄存器	HEAD20	R	H'00	H'FFFC2024	8
ECC 校正后扇区头 -SECONDS 数据寄存器	HEAD21	R	H'00	H'FFFC2025	8
ECC 校正后扇区头 -FRAMES （1/75 秒）数据寄存器	HEAD22	R	H'00	H'FFFC2026	8
ECC 校正后扇区头 -MODE 数据寄存器	HEAD23	R	H'00	H'FFFC2027	8
ECC 校正后子头 - 文件号码 （BYTE-16）数据寄存器	SHEAD20	R	H'00	H'FFFC2028	8

寄存器名	简称	R/W	初始值	地址	存取长度
ECC 校正后子头 - 通道号码 （BYTE-17）数据寄存器	SHEAD21	R	H'00	H'FFFC2029	8
ECC 校正后子头 - 子模式 （BYTE-18）数据寄存器	SHEAD22	R	H'00	H'FFFC202A	8
ECC 校正后子头 - 数据类型 （BYTE-19）数据寄存器	SHEAD23	R	H'00	H'FFFC202B	8
ECC 校正后子头 - 文件号码 （BYTE-20）数据寄存器	SHEAD24	R	H'00	H'FFFC202C	8
ECC 校正后子头 - 通道号码 （BYTE-21）数据寄存器	SHEAD25	R	H'00	H'FFFC202D	8
ECC 校正后子头 - 子模式 （BYTE-22）数据寄存器	SHEAD26	R	H'00	H'FFFC202E	8
ECC 校正后子头 - 数据类型 （BYTE-23）数据寄存器	SHEAD27	R	H'00	H'FFFC202F	8
自动缓冲设定控制寄存器	CBUFCTL0	R/W	H'04	H'FFFC2040	8
自动缓冲开始扇区设定 -MINUTES 控制寄存器	CBUFCTL1	R/W	H'00	H'FFFC2041	8
自动缓冲开始扇区设定 -SECONDS 控制寄存器	CBUFCTL2	R/W	H'00	H'FFFC2042	8
自动缓冲开始扇区设定 -FRAMES 控制寄存器	CBUFCTL3	R/W	H'00	H'FFFC2043	8
ISY 中断源屏蔽控制寄存器	CROMST0M	R/W	H'00	H'FFFC2045	8
CD-ROM 解码器模块复位控制寄存器	ROMDECRST	R/W	H'00	H'FFFC2100	8
CD-ROM 解码器模块复位状态寄存器	RSTSTAT	R	H'00	H'FFFC2101	8
SSI 数据控制寄存器	SSI	R/W	H'18	H'FFFC2102	8
中断标志寄存器	INTHOLD	R/W	H'00	H'FFFC2108	8
中断源屏蔽控制寄存器	INHINT	R/W	H'00	H'FFFC2109	8
CD-ROM 解码器数据流输入寄存器	STRMDIN0	R/W	H'0000	H'FFFC2200	读：16 写：16、 32
CD-ROM 解码器数据流输入寄存器	STRMDIN2	R/W	H'0000	H'FFFC2202	16
CD-ROM 解码器数据流输出寄存器	STRMDOUT0	R	H'0000	H'FFFC2204	16、 32

21.3.1　ROMDEC 允许控制寄存器 （CROMEN）

CROMEN 进行 Subcode 处理允许、 CD-ROM 解码处理允许、 CD-ROM 解码处理强制结束。

位：	7	6	5	4	3	2	1	0
	SUBC_EN	CROM_EN	CROM_STP	-	-	-	-	-
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	SUBC_EN	0	R/W	Subcode 处理允许 请与 CROM_EN 同时进行置位与清除。异常自动停止以及 CROM_STP=1 时，自动清 0。
6	CROM_EN	0	R/W	CD-ROM 解码处理允许 如果该位置 1，检测到有效同步码后开始 CD-ROM 解码处理。如果该位置 0，在当前解码中的扇区的处理结束时，中止解码处理。 通过自动解码停止功能停止时以及 CROM_STP=1 时，自动清 0。
5	CROM_STP	0	R/W	CD-ROM 解码处理强制结束 如果该位置 1，就立即停止 CD-ROM 解码处理。SUBC_EN、CROM_EN 位自动清 0。重新开始解码处理前，须将该位置 0。
4 ～ 0	—	均为 0	R/W	保留位 读写通常为 0。

21.3.2　同步码的同步控制寄存器 （CROMSY0）

CROMSY0 选择同步码保护功能。

位：	7	6	5	4	3	2	1	0
	SY_AUT	SY_IEN	SY_DEN	-	-	-	-	-
初始值：	1	0	0	0	1	0	0	1
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	SY_AUT	1	R/W	CD-ROM 同步码的自动同步保护模式 如果该位置 1， CD-ROM 同步码的同步保护（插入同步码）就为自动。该位置 1 时， SY_IEN、 SY_DEN 的设定无效。
6	SY_IEN	0	R/W	内部同步信号允许 使内部同步信号 （由 CD-ROM 解码器内部的计数器产生）有效。 当 SY_AUT=0 时，如果该位置 1，CD-ROM 数据的同步总是通过内部计数器成为内插模式。
5	SY_DEN	0	R/W	与同步信号的外部同步码同步 监视从输入数据检测的同步码，与内部计数器值无关，总是同步化。 该位的设定在 SY_AUT=0 时有效。
4	—	0	R/W	保留位 读写通常为 0。
3	—	1	R/W	保留位 读写通常为 1。
2、 1	—	均为 0	R/W	保留位 读写通常为 0。
0	—	1	R/W	保留位 读写通常为 1。

表 21.2　同步码保护功能的设定

SY_AUT	SY_IEN	SY_DEN	运行模式
1	—	—	自动同步保护模式
0	0	1	外部同步模式
0	1	0	内插同步模式
0	1	1	内插 / 外部同步模式
0	0	0	禁止设定

21.3.3　解码模式控制寄存器 （CROMCTL0）

CROMCTL0 进行各功能的控制、选择 Mode 判定 /Form 判定以及设定扇区类型。该寄存器的设定在切换扇区时有效。

位：	7	6	5	4	3	2	1	0
	MD_DESC	-	MD_AUTO	MD_AUTOS1	MD_AUTOS2	MD_SEC[2:0]		
初始值：	1	0	0	0	0	0	1	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	MD_DESC	1	R/W	解扰功能控制 0：解扰功能 OFF 1：解扰功能 ON
6	—	0	R/W	保留位 读写通常为 0。
5	MD_AUTO	0	R/W	Mode Form 的自动检测功能控制 0：OFF 1：ON 能检测的格式为 Mode0、Mode1、Mode2（not XA）、Mode2 Form1、Mode2 Form2。不能检测 Mode Form 时，继续上个扇区的 Mode Form。无法检测解码开始的第一个扇区的 Mode Form 时， MD_SEC[2:0] 的设定值作为初始值使用。
4	MD_AUTOS1	0	R/W	当 MD_AUTO=1 时的 Mode 判定基准 0：只在检测同步码时，进行 Mode 判定 1：总是进行 Mode 判定 该位的设定只在 MD_AUTO=1 时有效。无法判定时，继续上个扇区的 Mode。 如果该位置 0，只有在检测对应扇区的同步码时进行 Mode 判定。
3	MD_AUTOS2	0	R/W	当 MD_AUTO=1 时， Mode2 的 Form 判定基准 0：在子头内 2 处 Form 中，如果代码不匹配，为 not XA。 1：不进行 XA 判定。第一个 Form 为有效。但进行 2 处 Form 匹配检查，其结果反映在状态位。 该位的设定只在 MD_AUTO=1 时有效。
2 ～ 0	MD_SEC[2:0]	010	R/W	扇区类型设定 000：禁止设定 001：Mode0 010：Mode1 011：Long （Mode0、 Mode1、 Mode2　EDC/ECC 无数据） 100：禁止设定 101：Mode2 Form1 110：Mode2 Form2 111：Mode2 自动格式检测 当设定为 B'111 时，如果无法判定格式，就作为 Mode2 not XA 处理。

21.3.4　EDC、 ECC 校验的控制寄存器 （CROMCTL1）

CROMCTL1 为控制 EDC、 ECC 校验的寄存器。该寄存器的设定在切换扇区时有效。

位：	7	6	5	4	3	2	1	0
	M2F2 EDC	MD_DEC[2:0]			-	-	MD_PQREP[1:0]	
初始值：	1	1	0	1	0	0	0	1
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	M2F2EDC	1	R/W	在 Mode2 Form2 中，如果 EDC 码为 All 0， EDC 功能就无效。 如果该位置 1， Mode2 Form2 下 EDC 码为 0 时，即使 EDC 校检为 NG，也不发生 IERR 中断。
6 ～ 4	MD_DEC[2:0]	101	R/W	选择 EDC、 ECC 校验模式 000：无校验 001：只有 EDC 010：Q+EDC 011：P+EDC 100：QP+EDC 101：PQ+EDC 110：禁止设定 111：禁止设定
3、 2	—	均为 0	R/W	保留位 读写通常为 0。
1、 0	MD_PQREP[1:0]	01	R/W	PQ、 QP 校正的校正重复次数 通过 MD_DEC 位，选择设定 PQ、 QP 校正时的校正重复次数。 00：禁止设定 01：1 次校正 10：2 次重复校正 11：3 次重复校正

21.3.5　解码处理自动停止控制寄存器 （CROMCTL3）

发生解码异常时， CROMCTL3 可自动停止解码处理。停止时，发生 IBUF 中断，可在 CBUFST1 寄存器确认停止源。该寄存器的设定在切换扇区时有效。

位：	7	6	5	4	3	2	1	0
	STP_ECC	STP_EDC	-	STP_MD	STP_MIN	-	-	-
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	STP_ECC	0	R/W	如果 STP_ECC 位置 1，无法校正 ECC 时，中止 CD-ROM 解码处理。
6	STP_EDC	0	R/W	如果 STP_EDC 位置 1，在 ECC 后的 EDC 发生错误时，停止 CD-ROM 解码处理。
5	—	0	R/W	保留位 读写通常为 0。
4	STP_MD	0	R/W	如果 STP_MD 位置 1， Mode　Form 与之前的扇区不同时，停止解码处理。
3	STP_MIN	0	R/W	如果 STP_MIN 位置 1，当遇到不连续的 MINUTES、 SECONDS、 FRAME （1/75 秒）时，停止解码处理。
2 ～ 0	—	均为 0	R/W	保留位 读写通常为 0。

21.3.6　解码选项设定控制寄存器 （CROMCTL4）

CROMCTL4 进行连接块检测控制、状态寄存器的显示选择及 ECC 校正模式的控制。该寄存器的设定在切换扇区时有效。

位：	7	6	5	4	3	2	1	0
	LINKOFF	LINK2	-	ER0SEL	NO_ECC	-	-	-
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	LINKOFF	0	R/W	连接块检测时的缓冲控制 0：ON 1：OFF 如果该位置 1，检测到连接块时不进行缓冲控制。连接块作为普通扇区处理。 但是要进行连接块的检测 （结果反映在 CROMST5 寄存器的 bit3 ～ 0）。
6	LINK2	0	R/W	连接块检测条件的选择 0：检测出 run-out 1、2 的任意一个或 run-in 3、4 时，就判定为连接块 1：在 run-out 1、2 连接中，检测出 2 个时，就判定为连接块 该位置1时，连接块的缓冲控制无效（作为普通扇区处理）。另外，LINK_ON 置位的条件为解码连接扇区的时序。
5	—	0	R/W	保留位 读写通常为 0。
4	ER0SEL	0	R/W	选择 ROM 数据相关状态寄存器的设定条件 0：表示当前解码中的扇区的信息 1：表示缓冲结束的最新扇区的信息 对象为 CROMST0 寄存器的 bit5 ～ 0 以及 CROMST4、CROMST5 寄存器的 bit7 ～ 1、 HEAD00 ～ 02。
3	NO_ECC	0	R/W	在 ECC 前 EDC 校验为 OK 时，选择 ECC 校正模式 如果该位置 1，在 ECC 前 EDC 校验为 OK 时，与伴随式计算结果无关，不进行 ECC 校正。
2 ～ 0	—	均为 0	R/W	保留位 读写通常为 0。

21.3.7　HEAD20 ～ 22 表示控制寄存器 （CROMCTL5）

CROMCTL5 设定 HEAD20 ～ 22 的表示方式。

位：	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	MSF_LBA_SEL
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 ～ 1	—	均为 0	R/W	保留位 读写通常为 0。
0	MSF_LBA_SEL	0	R/W	HEAD20 ～ 22 的表示方式 0：扇区头的 MSF 直接以 BCD （十进制）表示 1：总扇区值以十六进制表示

21.3.8　同步码状态寄存器 （CROMST0）

CROMST0 表示同步码保护功能时的状态。

位：	7	6	5	4	3	2	1	0
	-	-	ST_SYIL	ST_SYNO	ST_BLKS	ST_BLKL	ST_SESC	ST_SECL
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、6	—	均为 0	R	保留位 读通常为 0。写无效。
5	ST_SYIL	0	R	表示在字计数器（测量同步码间隔）不正确时，虽然检测出同步码，但忽略并且不同步化。 该位只在自动同步保护模式或内插模式有效。
4	ST_SYNO	0	R	表示虽然字计数器为最终值，但没有检测出同步码，并通过内插同步化解码器。 该位只在自动同步保护模式或内插模式有效。
3	ST_BLKS	0	R	表示在字计数器的错误时检测同步码，并据此来同步化解码器。 该位只在自动同步保护模式或外部同步模式有效。
2	ST_BLKL	0	R	表示虽然字计数器为最终值，但没有检测出同步码，延长 1 个扇区的期间。 该位只在外部同步模式时有效。
1	ST_SECS	0	R	表示同步码与内插的同步时序作为短扇区对应。该位置 1 时，立即停止解码，从当前解码的前 1 个扇区进行重试。
0	ST_SECL	0	R	表示同步码和内插同步时序作为长扇区对应。该位置 1 时，立即停止解码，从当前解码的前 2 个扇区进行重试。

21.3.9　ECC 后的扇区头对应的错误状态寄存器 （CROMST1）

CROMST1 表示 ECC 后的扇区头对应的错误状态。

位：	7	6	5	4	3	2	1	0
	-	-	-	-	ER2_HEAD0	ER2_HEAD1	ER2_HEAD2	ER2_HEAD3
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 ～ 4	—	均为 0	R	保留位 读通常为 0。写无效。
3	ER2_HEAD0	0	R	表示 ECC 后的扇区头 MINUTES 对应的错误状态。
2	ER2_HEAD1	0	R	表示 ECC 后的扇区头 SECONDS 对应的错误状态。
1	ER2_HEAD2	0	R	表示 ECC 后的扇区头 FRAME （1/75 秒）对应的错误状态。
0	ER2_HEAD3	0	R	表示 ECC 后的扇区头 MODE 对应的错误状态。

21.3.10　ECC 后子头对应的错误状态寄存器 （CROMST3）

CROMST3 表示 ECC 后子头对应的错误状态。

位：	7	6	5	4	3	2	1	0
	ER2_SHEAD0	ER2_SHEAD1	ER2_SHEAD2	ER2_SHEAD3	ER2_SHEAD4	ER2_HEAD5	ER2_HEAD6	ER2_HEAD7
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	ER2_SHEAD0	0	R	表示 ECC 后子头 （文件号码）对应的错误状态。 表示 SHEAD20 寄存器的错误。
6	ER2_SHEAD1	0	R	表示 ECC 后子头 （通道号码）对应的错误状态。 表示 SHEAD21 寄存器的错误。
5	ER2_SHEAD2	0	R	表示 ECC 后子头 （子模式）对应的错误状态。 表示 SHEAD22 寄存器的错误。
4	ER2_SHEAD3	0	R	表示 ECC 后子头 （数据类型）对应的错误状态。 表示 SHEAD23 寄存器的错误。
3	ER2_SHEAD4	0	R	表示 ECC 后子头 （文件号码）对应的错误状态。 表示 SHEAD24 寄存器的错误。
2	ER2_SHEAD5	0	R	表示 ECC 后子头 （通道号码）对应的错误状态。 表示 SHEAD25 寄存器的错误。
1	ER2_SHEAD6	0	R	表示 ECC 后子头 （子模式）对应的错误状态。 表示 SHEAD26 寄存器的错误。
0	ER2_SHEAD7	0	R	表示 ECC 后子头 （数据类型）对应的错误状态。 表示 SHEAD27 寄存器的错误。

21.3.11　扇区头、子头数据妥当性判定状态寄存器 （CROMST4）

CROMST4 表示自动模式判定、 Mode2 的 Form 判定的错误。

位：	7	6	5	4	3	2	1	0
	NG_MD	NG_MDCMP1	NG_MDCMP2	NG_MDCMP3	NG_MDCMP4	NG_MDDEF	NG_MDTIM1	NG_MDTIM2
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	NG_MD	0	R	表示按照自动模式判定基准，不能判定模式。
6	NG_MDCMP1	0	R	表示判定 Mode2 的 Form 时，由于文件号码 （BYTE-16 与 BYTE-20）而发生比较错误。
5	NG_MDCMP2	0	R	表示判定 Mode2 的 Form 时，由于通道号码 （BYTE-17 与 BYTE-21）而发生比较错误。
4	NG_MDCMP3	0	R	表示判定 Mode2 的 Form 时，由于子模式 （BYTE-18 与 BYTE-22）而发生比较错误。
3	NG_MDCMP4	0	R	表示判定 Mode2 的 Form 时，由于数据类型 （BYTE-19 与 BYTE-23）而发生比较错误。
2	NG_MDDEF	0	R	表示 Mode、 Form 与之前的扇区不同。
1	NG_MDTIM1	0	R	表示扇区头的 MINUTES、 SECONDS、 FRAMES （1/75 秒）的值中途偏移。 下个扇区的连续检查使用更新值。
0	NG_MDTIM2	0	R	表示扇区头的 MINUTES、 SECONDS、 FRAMES 不是 BCD （十进制）的值。 表示该位不是 BCD 值 （A ～ F）、 HEAD01 的值比 H'59 大、 HEAD02 的值比 H'74 大。 下个扇区的连续检查使用内插值。

21.3.12　模式判定结果和连接扇区检测状态寄存器 （CROMST5）

CROMST5 表示自动模式判定结果及连接块检测。

位：	7	6	5	4	3	2	1	0
	ST_AMD[2:0]			ST_MDX	LINK_ON	LINK_DET	LINK_SDET	LINK_OUT1
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 ～ 5	ST_AMD[2:0]	000	R	自动模式判定结果 表示使用自动模式判定功能时的模式判定结果。 000：不使用自动模式判定功能 001：Mode0 010：Mode1 011：- 100：Mode2 not XA 101：Mode2 Form1 110：Mode2 Form2 111：-
4	ST_MDX	0	R	表示不使用自动模式判定，而手动设定时，设定值与逻辑识别的结果不匹配。手动设定值优先。
3	LINK_ON	0	R	通过连接块判定识别为连接块时，该位置 1。 连接块判定基准参照 CROMCTL4 寄存器的 LINK2 位。 该位置 1 时，按照 CBUFCTL0 寄存器的 CBUF_LINK 位，进行缓冲控制。
2	LINK_DET	0	R	表示检测到连接块 （run-out 1 ～ run-in 4）。 因为用 ECC 校正前的数据检测，所以在与连接块相同的代码发生数据错误时， LINK_DET=1。
1	LINK_SDET	0	R	表示解码开始后的 7 个扇区内检测到连接块。
0	LINK_OUT1	0	R	表示 ECC 处理后判定为 run-out 1 扇区。 该位只有在不发生 IERR 中断 （正常进行 ECC 校正）时有效。

21.3.13　ECC、EDC 错误状态寄存器 （CROMST6）

CROMST6 表示 ECC 处理错误及 ECC 前后的 EDC 校检错误。

位：	7	6	5	4	3	2	1	0
	ST_ERR	-	ST_ECCABT	ST_ECCNG	ST_ECCP	ST_ECCQ	ST_EDC1	ST_EDC2
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	ST_ERR	0	R	表示 ECC 后的解码块即使为 1 字节也有错误。
6	—	0	R	保留位 读通常为 0。写无效。
5	ST_ECCABT	0	R	表示 ECC 处理中断。 在 ECC 校正处理中，发生扇区切换时，该位置 1。根据时序，即使该位置 1，如果 CBUFST2 寄存器的 BUF_NG 位为 0， ECC 校正也运行正常。
4	ST_ECCNG	0	R	表示不能进行错误校正。 检测到短扇区时，该位也置 1。
3	ST_ECCP	0	R	表示在 ECC 处理中不能校正 P 系列。 该位只在同步状态正常 （不是短扇区或长扇区）时有效。 P 系列的伴随值不是 All0 时，该位置 1。
2	ST_ECCQ	0	R	表示不能通过 ECC 处理校正 Q 系列。 该位只在同步状态正常 （不是短扇区或长扇区）时有效。 Q 系列的伴随值不是 All0 时，该位置 1。
1	ST_EDC1	0	R	表示在 ECC 前的 EDC 校检为 NG。 通过 EDC 有效，为短扇区时，该位也置 1。
0	ST_EDC2	0	R	表示 ECC 后的 EDC 校检为 NG。

21.3.14　缓冲器状态寄存器 （CBUFST0）

CBUFST0 表示正在进行缓冲开始扇区搜索或缓冲中。

位：	7	6	5	4	3	2	1	0
	BUF_REF	BUF_ACT	-	-	-	-	-	-
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	BUF_REF	0	R	表示正在搜索缓冲开始扇区。 该位只在使用自动缓冲功能时 （CBUF_AUT=1）有效。
6	BUF_ACT	0	R	表示正在缓冲。
5 ～ 0	—	均为 0	R	保留位 读通常为 0。写无效。

21.3.15　解码中止源状态寄存器 （CBUFST1）

CBUFST1 表示因各错误中止解码 / 缓冲处理。

只有 CROMCTL3 的对应位置 1 时，该寄存器才被置位。

位：	7	6	5	4	3	2	1	0
	BUF_ECC	BUF_EDC	-	BUF_MD	BUF_MIN	-	-	-
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	BUF_ECC	0	R	表示因不能进行 ECC 校正，而中止 ROM 解码 / 缓冲处理。
6	BUF_EDC	0	R	表示通过 ECC 校正后的 EDC 检查，检测出错误，而中止解码 / 缓冲处理。
5	—	0	R	保留位 读通常为 0。写无效。
4	BUF_MD	0	R	表示因 Mode、 Form 与之前的扇区不同，中止解码 / 缓冲处理。
3	BUF_MIN	0	R	因 MINUTES、SECONDS、FRAMES （1/75 秒）不连续，而中止解码 / 缓冲处理。
2 ～ 0	—	均为 0	R	保留位 读通常为 0。写无效。

21.3.16　缓冲溢出状态寄存器 （CBUFST2）

CBUFST2 表示向缓冲器的传送没有结束时，发生扇区切换。

位：	7	6	5	4	3	2	1	0
	BUF_NG	-	-	-	-	-	-	-
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	BUF_NG	0	R	表示向缓冲器的数据传送没有结束时，发生扇区切换。如果不从 CD-ROM 解码器引出输出数据流，输入第 3 个扇区的数据时，该位置 1。不发生中断。该位置 1 时，只有通过 ROMDECRST 寄存器的 LOGICRST 位复位，才能将该位清 0。
6 ～ 0	—	均为 0	R	保留位 读通常为 0。写无效。

21.3.17　ECC 校正前扇区头 -MINUTES 数据寄存器 （HEAD00）

HEAD00 表示 ECC 校正前扇区头 MINUTES 的值。

位：	7	6	5	4	3	2	1	0
	HEAD00[7:0]							
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	HEAD00[7:0]	H'00	R	ECC 校正前扇区头 MINUTES 的值

21.3.18　ECC 校正前扇区头 -SECONDS 数据寄存器 （HEAD01）

HEAD01 表示 ECC 校正前扇区头 SECONDS 的值。

位：	7	6	5	4	3	2	1	0
	HEAD01[7:0]							
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	HEAD01[7:0]	H'00	R	ECC 校正前扇区头 SECONDS 的值

21.3.19　ECC 校正前扇区头 -FRAMES （1/75 秒）数据寄存器 （HEAD02）

HEAD02 表示 ECC 校正前扇区头 FRAMES （1/75 秒）的值。

位：	7	6	5	4	3	2	1	0
	HEAD02[7:0]							
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	HEAD02[7:0]	H'00	R	ECC 校正前扇区头 FRAMES （1/75 秒）的值

21.3.20　ECC 校正前扇区头 -MODE 数据寄存器 （HEAD03）

HEAD03 表示 ECC 校正前扇区头 MODE 的值。

位：	7	6	5	4	3	2	1	0
	HEAD03[7:0]							
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	HEAD03[7:0]	H'00	R	ECC 校正前扇区头 MODE 的值

21.3.21　ECC 校正前子头 - 文件号码 （BYTE-16）数据寄存器 （SHEAD00）

SHEAD00 表示 ECC 校正前子头文件号码的值 （BYTE-16）。

位：	7	6	5	4	3	2	1	0
	SHEAD00[7:0]							
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	SHEAD00[7:0]	H'00	R	ECC 校正前子头的文件号码的值 （BYTE-16） 表示不是 Mode2 时，对应位置的字节数据。

21.3.22　ECC 校正前子头 - 通道号码 （BYTE-17）数据寄存器 （SHEAD01）

SHEAD01 表示 ECC 校正前子头的通道号码的值 （BYTE-17）。

位：	7	6	5	4	3	2	1	0
	SHEAD01[7:0]							
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	SHEAD01[7:0]	H'00	R	ECC 校正前子头的通道号码的值 （BYTE-17） 表示不是 Mode2 时，对应位置的字节数据。

21.3.23　ECC 校正前子头 - 子模式 （BYTE-18）数据寄存器 （SHEAD02）

SHEAD02 表示 ECC 校正前子头子模式的值 （BYTE-18）。

位：	7	6	5	4	3	2	1	0
	SHEAD02[7:0]							
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	SHEAD02[7:0]	H'00	R	ECC 校正前子头子模式的值 （BYTE-18） 表示不是 Mode2 时，对应位置的字节数据。

21.3.24　ECC 校正前子头 - 数据类型 （BYTE-19）数据寄存器 （SHEAD03）

SHEAD03 表示 ECC 校正前子头的数据类型值 （BYTE-19）。

位：	7	6	5	4	3	2	1	0
	SHEAD03[7:0]							
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	SHEAD03[7:0]	H'00	R	ECC 校正前子头的数据类型值 （BYTE-19） 表示不是 Mode2 时，对应位置的字节数据。

21.3.25　ECC 校正前子头 - 文件号码 （BYTE-20）数据寄存器 （SHEAD04）

SHEAD04 表示 ECC 校正前子头文件号码值 （BYTE-20）。

位：	7	6	5	4	3	2	1	0
	SHEAD04[7:0]							
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	SHEAD04[7:0]	H'00	R	ECC 校正前子头文件号码值 （BYTE-20） 表示不是 Mode2 时，对应位置的字节数据。

21.3.26　ECC 校正前子头 - 通道号码 （BYTE-21）数据寄存器 （SHEAD05）

SHEAD05 表示 ECC 校正前子头通道号码值 （BYTE-21）。

位：	7	6	5	4	3	2	1	0
	SHEAD05[7:0]							
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	SHEAD05[7:0]	H'00	R	ECC 校正前子头通道号码值 （BYTE-21） 表示不是 Mode2 时，对应位置的字节数据。

21.3.27　ECC 校正前子头 - 子模式 （BYTE-22）数据寄存器 （SHEAD06）

SHEAD06 表示 ECC 校正前子头子模式值 （BYTE-22）。

位：	7	6	5	4	3	2	1	0
	SHEAD06[7:0]							
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	SHEAD06[7:0]	H'00	R	ECC 校正前子头的子模式值 （BYTE-22） 表示不是 Mode2 时，对应位置的字节数据。

21.3.28　ECC 校正前子头 - 数据类型 （BYTE-23）数据寄存器 （SHEAD07）

SHEAD07 表示 ECC 校正前子头数据类型值 （BYTE-23）。

位：	7	6	5	4	3	2	1	0
	SHEAD07[7:0]							
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	SHEAD07[7:0]	H'00	R	ECC 校正前子头数据类型值 （BYTE-23） 表示不是 Mode2 时，对应位置的字节数据。

21.3.29　ECC 校正后扇区头 -MINUTES 数据寄存器 （HEAD20）

HEAD20 表示 ECC 校正后扇区头 MINUTES 值。

位：	7	6	5	4	3	2	1	0
	HEAD20[7:0]							
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	HEAD20[7:0]	H'00	R	ECC 校正后扇区头 MINUTES 值 表示 MSF_LBA_SEL=1 时， M、 S、 F 的总扇区值 （1/3）。

21.3.30　ECC 校正后扇区头 -SECONDS 数据寄存器 （HEAD21）

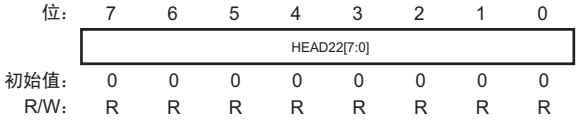
HEAD21 表示 ECC 校正后扇区头 SECONDS 值。

位：	7	6	5	4	3	2	1	0
	HEAD21[7:0]							
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	HEAD21[7:0]	H'00	R	ECC 校正后扇区头 SECONDS 值 表示 MSF_LBA_SEL=1 时， M、 S、 F 的总扇区值 （2/3）。

21.3.31　ECC 校正后扇区头 -FRAMES （1/75 秒）数据寄存器 （HEAD22）

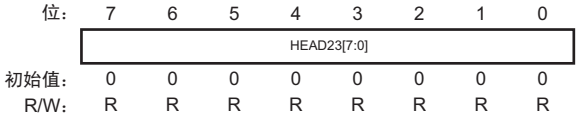
HEAD22 表示 ECC 校正后扇区头 FRAMES （1/75 秒）值。



位	位名	初始值	R/W	说　　明
7 ～ 0	HEAD22[7:0]	H'00	R	ECC 校正后扇区头 FRAMES （1/75 秒）值 表示 MSF_LBA_SEL=1 时， M、 S、 F 的总扇区值 （3/3）。

21.3.32　ECC 校正后扇区头 -MODE 数据寄存器 （HEAD23）

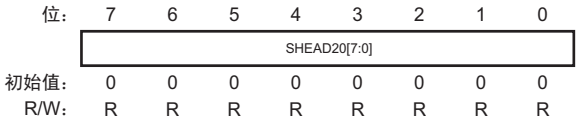
HEAD23 表示 ECC 校正后扇区头 MODE 值。



位	位名	初始值	R/W	说　　明
7 ～ 0	HEAD23[7:0]	H'00	R	ECC 校正后扇区头 MODE 值

21.3.33　ECC 校正后子头 - 文件号码 （BYTE-16）数据寄存器 （SHEAD20）

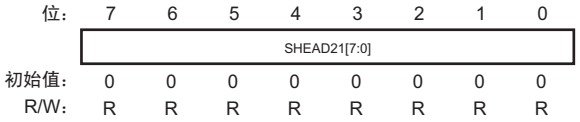
SHEAD20 表示 ECC 校正后子头文件号码值 （BYTE-16）。



位	位名	初始值	R/W	说　　明
7 ～ 0	SHEAD20[7:0]	H'00	R	ECC 校正后子头文件号码值 （BYTE-16）

21.3.34　ECC 校正后子头 - 通道号码 （BYTE-17）数据寄存器 （SHEAD21）

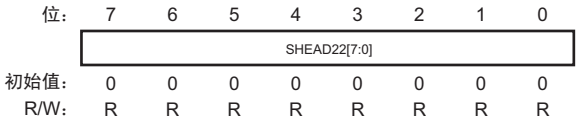
SHEAD21 表示 ECC 校正后子头通道号码值 （BYTE-17）。



位	位名	初始值	R/W	说　　明
7 ～ 0	SHEAD21[7:0]	H'00	R	ECC 校正后子头通道号码值 （BYTE-17）

21.3.35　ECC 校正后子头 - 子模式 （BYTE-18）数据寄存器 （SHEAD22）

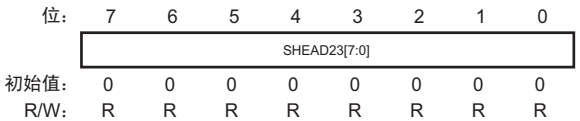
SHEAD22 表示 ECC 校正后子头子模式值 （BYTE-18）。



位	位名	初始值	R/W	说　　明
7 ～ 0	SHEAD22[7:0]	H'00	R	ECC 校正后子头子模式值 （BYTE-18）

21.3.36　ECC 校正后子头 - 数据类型 （BYTE-19）数据寄存器 （SHEAD23）

SHEAD23 表示 ECC 校正后子头的数据类型值 （BYTE-19）。



位	位名	初始值	R/W	说　　明
7 ～ 0	SHEAD23[7:0]	H'00	R	ECC 校正后子头的数据类型值 （BYTE-19）

21.3.37　ECC 校正后子头 - 文件号码 （BYTE-20）数据寄存器 （SHEAD24）

SHEAD24 表示 ECC 校正后子头的文件号码值 （BYTE-20）。

位：	7	6	5	4	3	2	1	0
	SHEAD24[7:0]							
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	SHEAD24[7:0]	H'00	R	ECC 校正后子头的文件号码值 （BYTE-20）

21.3.38　ECC 校正后子头 - 通道号码 （BYTE-21）数据寄存器 （SHEAD25）

SHEAD25 表示 ECC 校正后子头的通道号码值 （BYTE-21）。

位：	7	6	5	4	3	2	1	0
	SHEAD25[7:0]							
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	SHEAD25[7:0]	H'00	R	ECC 校正后子头的通道号码值 （BYTE-21）

21.3.39　ECC 校正后子头 - 子模式 （BYTE-22）数据寄存器 （SHEAD26）

SHEAD26 表示 ECC 校正后子头子模式值 （BYTE-22）。

位：	7	6	5	4	3	2	1	0
	SHEAD26[7:0]							
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	SHEAD26[7:0]	H'00	R	ECC 校正后子头子模式值 （BYTE-22）

21.3.40　ECC 校正后子头 - 数据类型 （BYTE-23）数据寄存器 （SHEAD27）

SHEAD27 表示 ECC 校正后子头的数据类型值 （BYTE-23）。

位：	7	6	5	4	3	2	1	0
	SHEAD27[7:0]							
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	SHEAD27[7:0]	H'00	R	ECC 校正后子头的数据类型值 （BYTE-23）

21.3.41　自动缓冲设定控制寄存器 （CBUFCTL0）

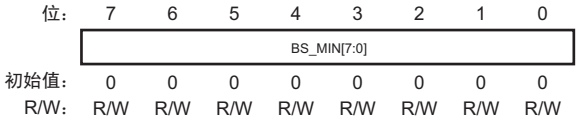
位：	7	6	5	4	3	2	1	0
	CBUF_AUT	CBUF_EN	CBUF_LINK	CBUF_MD[1:0]		CBUF_TS	CBUF_Q	-
初始值：	0	0	0	0	0	1	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说　　明
7	CBUF_AUT	0	R/W	自动缓冲功能控制 设定 CROM_EN=1 的状态，该位 ON/OFF 时，CBUF_EN 也同时进行 ON/OFF。如果不这样，就不能保证 CBUFST0、CBUFST1、CBUFST2 的表示。 0：自动缓冲 OFF 1：自动缓冲 ON
6	CBUF_EN	0	R/W	对缓冲器 RAM 的缓冲控制 该位在自动 / 手动缓冲模式中，进行缓冲的 ON/OFF。手动缓冲时，在发生 ISEC 中断后，请将该位置位。　自动缓冲停止时，自动复位。 0：缓冲 OFF 1：缓冲 ON
5	CBUF_LINK	0	R/W	连接块检测时的缓冲控制 0：确保 7 个扇区的区域 1：装入连接块，进行缓冲
4、3	CBUF_MD [1:0]	00	R/W	设定使用自动缓冲功能时的初始扇区检测模式 00：在上个扇区检测与当前扇区检测中连续性 （扇区头值） OK 01：在当前扇区检测中与内插值的连续性 OK 10：当前扇区检测 OK 11：即使当前扇区未检测，根据内插值判定
2	CBUF_TS	1	R/W	CBUFCTL1 ～ 3 的设定方法 0：CBUFCTL1 ～ 3：BCD （十进制） 1：总扇区值 （十六进制）

位	位名	初始值	R/W	说　　明
1	CBUF_Q	0	R/W	指定 QCODE 的 CRC-NG 时的 QCODE 缓冲数据 0：缓冲 CRC-OK 的最终扇区的值 1：直接缓冲 NG 数据 【注】因为本 LSI 不能输入子代码，所以本位总是设定为 1。
0	—	0	R/W	保留位 读写值总是为 0。

21.3.42　自动缓冲开始扇区设定 -MINUTES 控制寄存器 （CBUFCTL1）

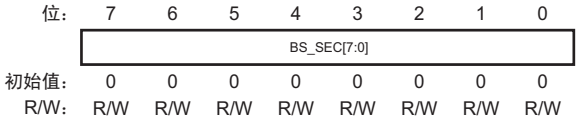
CBUFCTL1 表示缓冲开始扇区的扇区头 -MINUTES 值。



位	位名	初始值	R/W	说　　明
7 ～ 0	BS_MIN[7:0]	H'00	R/W	缓冲开始扇区的扇区头 -MINUTES 值

21.3.43　自动缓冲开始扇区设定 -SECONDS 控制寄存器 （CBUFCTL2）

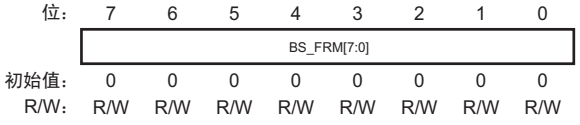
CBUFCTL2 表示缓冲开始扇区的扇区头 -SECONDS 值。



位	位名	初始值	R/W	说　　明
7 ～ 0	BS_SEC[7:0]	H'00	R/W	缓冲开始扇区的扇区头 -SECONDS 值

21.3.44　自动缓冲开始扇区设定 -FRAMES 控制寄存器 （CBUFCTL3）

CBUFCTL3 表示缓冲开始扇区的扇区头 -FRAMES （1/75 秒）值。



位	位名	初始值	R/W	说　　明
7 ～ 0	BS_FRM[7:0]	H'00	R/W	缓冲开始扇区的扇区头 -FRAMES （1/75 秒）值

21.3.45　ISY 中断源屏蔽控制寄存器 （CROMST0M）

CROMST0M 屏蔽由同步码状态寄存器 （CROMST0）的各位引起的 ISY 中断源。

位：	7	6	5	4	3	2	1	0
	-	-	ST_SYILM	ST_SYNOM	ST_BLKSM	ST_BLKLM	ST_SECSM	ST_SECLM
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	—	均为 0	R/W	保留位 读写通常为 0。
5	ST_SYILM	0	R/W	屏蔽 ISY 中断 ST_SYIL （CROMST0 寄存器的 bit5）源
4	ST_SYNOM	0	R/W	屏蔽 ISY 中断 ST_SYNO （CROMST0 寄存器的 bit4）源
3	ST_BLKSM	0	R/W	屏蔽 ISY 中断 ST_BLKS （CROMST0 寄存器的 bit3）源
2	ST_BLKLM	0	R/W	屏蔽 ISY 中断 ST_BLKL （CROMST0 寄存器的 bit2）源
1	ST_SECSM	0	R/W	屏蔽 ISY 中断 ST_SECS （CROMST0 寄存器的 bit1）源
0	ST_SECLM	0	R/W	屏蔽 ISY 中断 ST_SECL （CROMST0 寄存器的 bit0）源

21.3.46　CD-ROM 解码器模块复位控制寄存器 （ROMDECRST）

ROMDECRST 进行 CD-ROM 解码器的随机逻辑的复位，以及 CD-ROM 解码器的 RAM 清除。

位：	7	6	5	4	3	2	1	0
	LOGICRST	RAMRST	-	-	-	-	-	-
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	LOGICRST	0	R/W	CD-ROM 解码器的随机逻辑的复位信号 在该寄存器中置 1 时，输出复位信号。
6	RAMRST	0	R/W	CD-ROM 解码器具有的 RAM 清除信号 请通过 RSTSTAT 寄存器的 RAMCLRST 位，确认 RAM 清除结束。
5 ～ 0	—	均为 0	R/W	保留位 读写通常为 0。

【注】 LOGICRST 置 1 时，在确认 RAMRST 位清 0 后，请将该寄存器写 B'10000000。

21.3.47　CD-ROM 解码器模块复位状态寄存器 （RSTSTAT）

RSTSTAT 表示 CD-ROM 解码器的 RAM 的清除状态。

位：	7	6	5	4	3	2	1	0
	RAM CLRST	-	-	-	-	-	-	-
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	RAMCLRST	0	R	ROMDECRST 寄存器的 RAMRST=1 后，如果 RAM 的清除结束，该位置 1。通过向 RAMRST 写 0，清除该位。
6 ～ 0	—	均为 0	R	保留位 读通常为 0。写无效。

21.3.48　SSI 数据控制寄存器 （SSI）

SSI 进行数据流相关的各种设定。关于设定该寄存器时的运行情况请参照 “21.4.1 输入数据流字节序变换功能”。

位：	7	6	5	4	3	2	1	0
	BYTEND	BITEND	BUFEND0[1:0]		BUFEND1[1:0]		-	-
初始值：	0	0	0	1	1	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	BYTEND	0	R/W	改变从 SSI 输入数据流的字节序。 如果置 1，改换 STRMDIN0、 1、 STRMDIN2、 3 的字节 0 和字节 1。
6	BITEND	0	R/W	改变从 SSI 输入数据流的位顺序。 如果置 1，字节内的位顺序从 0→7 改换为 7→0。
5、 4	BUFEND0[1:0]	01	R/W	选择更换从 SSI 传送的数据流的顺序，或抑制数据流。在 SSI 中可选择“填充模式”“无填充模式”。在“无填充模式”中，从 SSI 把 32 位的数据作为 CD-ROM 数据传送。在 CD-ROM 解码器内部，有 2 组 16 位的输入数据流寄存器，可更换该顺序。另外，在“填充模式”中，从 SSI 传送填充附带的 32 位数据。由于填充部无意义，因此作为数据流需进行抑制，可通过该寄存器的设定进行抑制。 CD-ROM 解码作为 16 位的数据流处理，在该寄存器中，抑制来自 SSI 的 32 位传送数据中的首先输入的 16 位。 00：抑制首先处理的数据流 01：来自 SSI 的 32 位数据流中，首先将高 16 位数据作为数据流处理 10：来自 SSI 的 32 位数据流中，首先将低 16 位数据作为数据流处理 11：禁止设定
3、 2	BUFEND1[1:0]	10	R/W	选择更换从 SSI 传送的数据流的顺序或抑制数据流。在 SSI 中可选择“填充模式”“无填充模式”。在“无填充模式”中，从 SSI 把 32 位数据作为 CD-ROM 数据传送。在 CD-ROM 解码器内部，有 2 组 16 位的输入数据流寄存器，可更换该顺序。另外，在 “填充模式”中，从 SSI 传送填充附带的 32 位数据。由于填充部无意义，因此作为数据流需进行抑制，能通过该寄存器的设定进行抑制。 CD-ROM 解码作为 16 位的数据流处理，在该寄存器中，抑制来自 SSI 的 32 位传送数据中的后面输入的 16 位。 00：抑制后面处理的数据流 01：来自 SSI 的 32 位数据流中，将高 16 位数据作为后面的数据流处理 10：来自 SSI 的 32 位数据流中，将低 16 位数据作为后面的数据流处理 11：禁止设定
1、 0	—	均为 0	R/W	保留位 读写通常为 0。

21.3.49　中断标志寄存器 （INTHOLD）

INTHOLD 由各种中断标志构成。

位：	7	6	5	4	3	2	1	0
	ISEC	ITARG	ISY	IERR	IBUF	IREADY	-	-
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	ISEC	0	R/W	ISEC 中断标志 不读取 1，就不能写入 0。
6	ITARG	0	R/W	ITARG 中断标志 不读取 1，就不能写入 0。
5	ISY	0	R/W	ISY 中断标志 不读取 1，就不能写入 0。
4	IERR	0	R/W	IERR 中断标志 不读取 1，就不能写入 0
3	IBUF	0	R/W	IBUF 中断标志 不读取 1，就不能写入 0
2	IREADY	0	R/W	IREADY 中断标志 不读取 1，就不能写入 0
1、0	—	均为 0	R/W	保留位 读写通常为 0。

21.3.50　中断源屏蔽控制寄存器 （INHINT）

INHINT 控制 CD-ROM 解码器的各种中断请求。

位：	7	6	5	4	3	2	1	0
	INH ISEC	INH ITARG	INH ISY	INH IERR	INH IBUF	INH IREADY	PREINH REQDM	PREINH IREADY
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	INHISEC	0	R/W	ISEC 中断屏蔽 该位置 1 时，禁止 ISEC 中断请求。
6	INHITARG	0	R/W	ITARG 中断屏蔽 该位置 1 时，禁止 ITARG 中断请求。
5	INHISY	0	R/W	ISY 中断屏蔽 该位置 1 时，禁止 ISY 中断请求。
4	INHIERR	0	R/W	IERR 中断屏蔽 该位置 1 时，禁止 IERR 中断请求。
3	INHIBUF	0	R/W	IBUF 中断屏蔽 该位置 1 时，禁止 IBUF 中断请求。
2	INHIREADY	0	R/W	IREADY 中断屏蔽 该位置 1 时，禁止 IREADY 中断请求。
1	PREINHREQDM	0	R/W	禁止输出数据流的 DMA 传送请求中断标志置位。 该位置 1 时，不保持 DMA 传送请求中断源。
0	PREINHIREADY	0	R/W	禁止 IREADY 中断的标志置位。 该位置 1 时，在 IREADY 标志不保持中断源。

21.3.51　CD-ROM 解码器数据流输入寄存器 （STRMDIN0）

STRMDIN0 表示输入 CD-ROM 解码器的 4 个字节数据中从 MSB 开始的 2 个字节。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STRMDIN[31:16]															
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	位名	初始值	R/W	说　　明
15 ～ 0	STRMDIN [31:16]	H'0000	R/W	输入 CD-ROM 解码器的 4 个字节数据中从 MSB 开始的 2 个字节 CD-ROM 解码器将 4 字节宽度的数据窗口作为寄存器，将输入该寄存器的数据作为数据流处理。 1 个扇区的数据量为 2352 字节。

21.3.52　CD-ROM 解码器数据流输入寄存器 （STRMDIN2）

STRMDIN2 表示输入 CD-ROM 解码器的 4 个字节数据中从 LSB 开始的 2 个字节。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STRMDIN[15:0]															
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 ～ 0	STRMDIN [15:0]	H'0000	R/W	输入 CD-ROM 解码器的 4 个字节数据中从 LSB 开始的 2 个字节 CD-ROM 解码器将 4 字节宽度的数据窗口作为寄存器，将输入该寄存器的数据作为数据流处理。 1 个扇区的数据量为 2352 字节。

21.3.53　CD-ROM 解码器数据流输出寄存器 （STRMDOUT0）

STRMDOUT0 表示输入 CD-ROM 解码器的 4 个字节数据中从 MSB 开始的 2 个字节。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	STRMDOUT[15:0]															
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	STRMDOUT [15:0]	H'0000	R	从 CD-ROM 解码器输出 4 个字节数据中从 MSB 开始的 2 个字节 CD-ROM 解码器将 4 字节宽度的数据窗口作为寄存器，从该寄存器输出的数据为 ROM 解码处理后的数据。存取该寄存器时，按照分别定义的输出格式，依次输出存取长度的数据。 1 个扇区的数据为 2768 字节。必须读 2768 字节。

21.4　运行说明

21.4.1　输入数据流字节序变换功能

必须按照 CD-ROM 数据格式规格将数据流输入 CD-ROM 解码器内核。但在一些系统里，需要更换来自 SSI 的数据顺序或者传送填充部。因此在数据流输入控制部有更换数据顺序以及不将填充数据输入 CD-ROM 解码器内核的功能。

这些都是通过 SSI 数据控制寄存器（SSI）控制。图 21.6 所示的是将“填充数据 + 同步码起始 2 字节”（也就是将 H'000000FF 以 16 位为单位按照相反的数据顺序 H'00FF0000）作为数据流输入 CD-ROM 解码器。

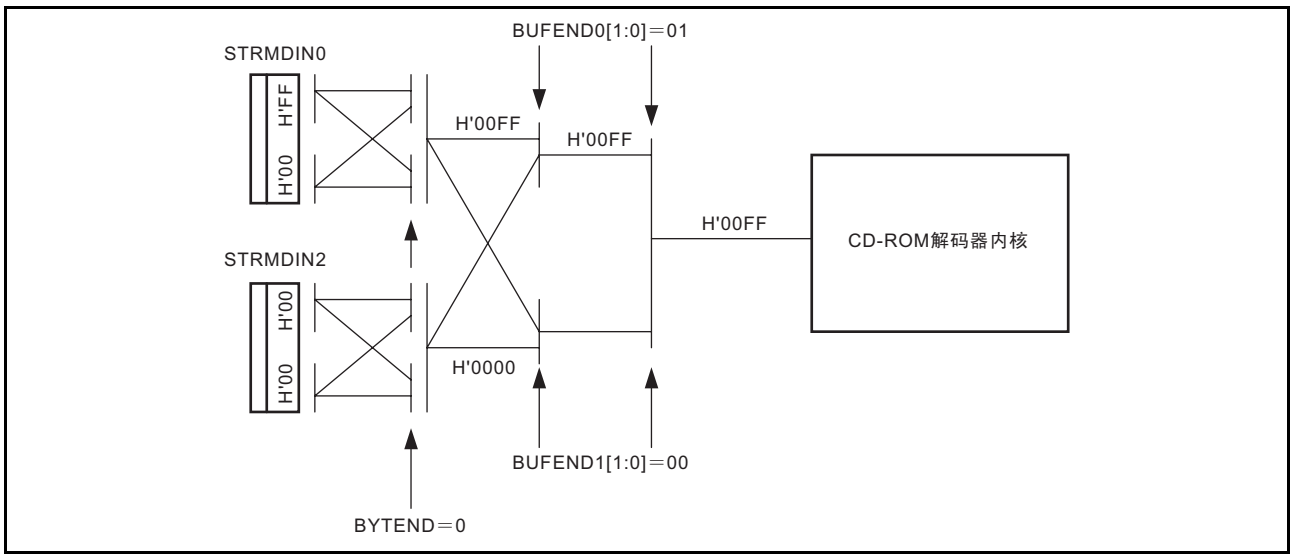


图 21.6　填充数据带数据流的 SSI 寄存器控制例子

图 21.7 所示的是将数据（H'56781234）作为数据流输入 CD-ROM 解码器。H'56781234 为将无填充数据的 H'12345678 以 16 位为单位更换顺序后的数据。

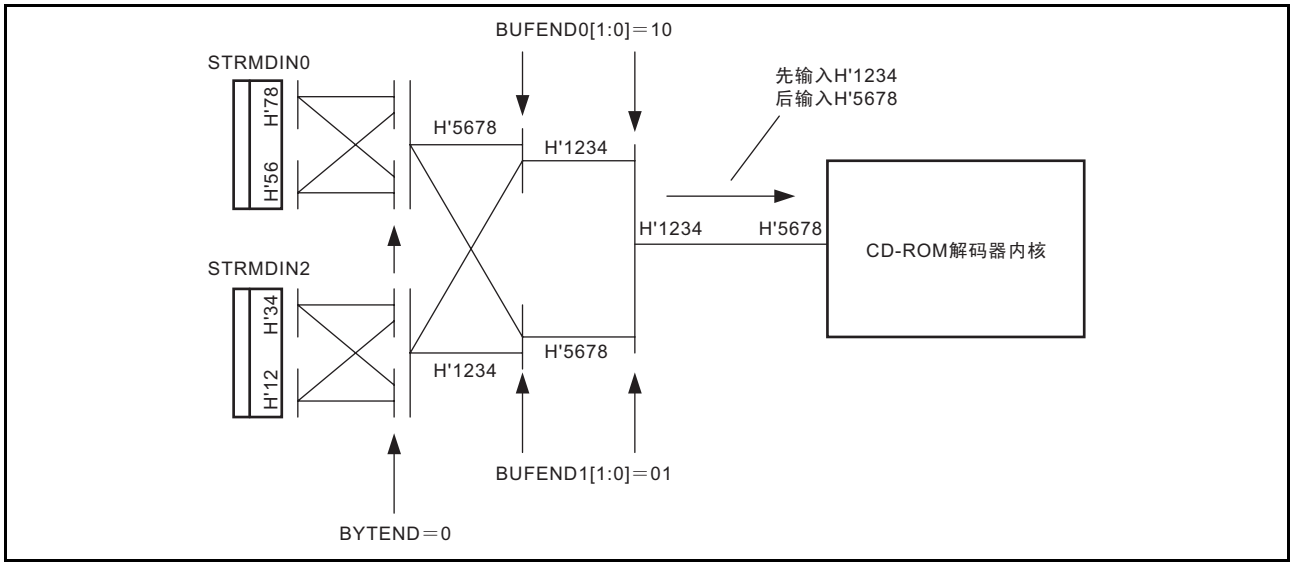


图 21.7　数据流 （无填充数据）的 SSI 寄存器控制例子

21.4.2　同步码保护功能

以 H'00FFFFFFFFFFFFFFFFFFFF00（同步码）开始的 2352 字节为 CD-ROM 的 1 个扇区的数据，但因划伤等原因，也有在错误的时序识别同步码，相反在应该检测出同步码的时序不能检测出同步码的情况。因此，在该 CD-ROM 解码器中拥有以下功能：

1. 在错误时序检测到同步码时，忽略该同步码
2. 在应该检测出同步码的时序无法检测出同步码时，保护同步码。
   同步码保护功能有以下模式。设定的详细情况请参照 “21.3.2 同步码的同步控制寄存器（CROMSY0）”及表21.2。

- 自动同步保护模式
- 外部同步模式
- 内插同步模式
- 内插/外部同步模式

（1） 自动同步保护模式

自动同步保护模式忽略 1 个扇区（2352 字节）期间中检测到的同步码，在下个扇区的起始不能检测到同步码时，保护同步码。另外如果同步时序变化，在同步时序变化后，在 2352 字节内检测出同步码时，再次进行同步。

因此在排除异常同步模式，并且跟随同步时序变更时有效。但不能跟随同步时序变更时的第一个扇区。

正常时检测同步码时的运行如图 21.8 所示，在 1 个扇区内检测同步码时检测到同步码时的运行图 21.9 所示，在超过 1 个扇区期间的时序检测到同步码时的运行如图 21.10 所示。

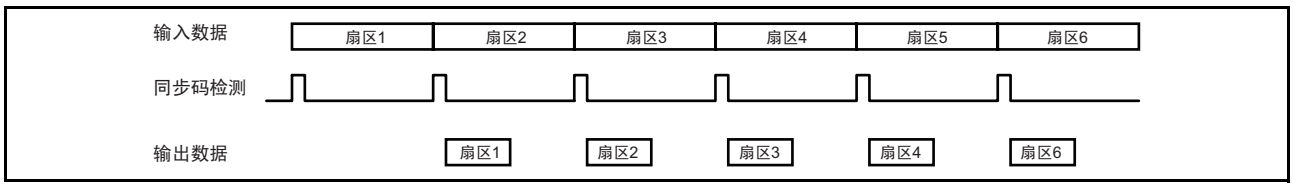


图 21.8　在正常时序的自动同步保护模式运行

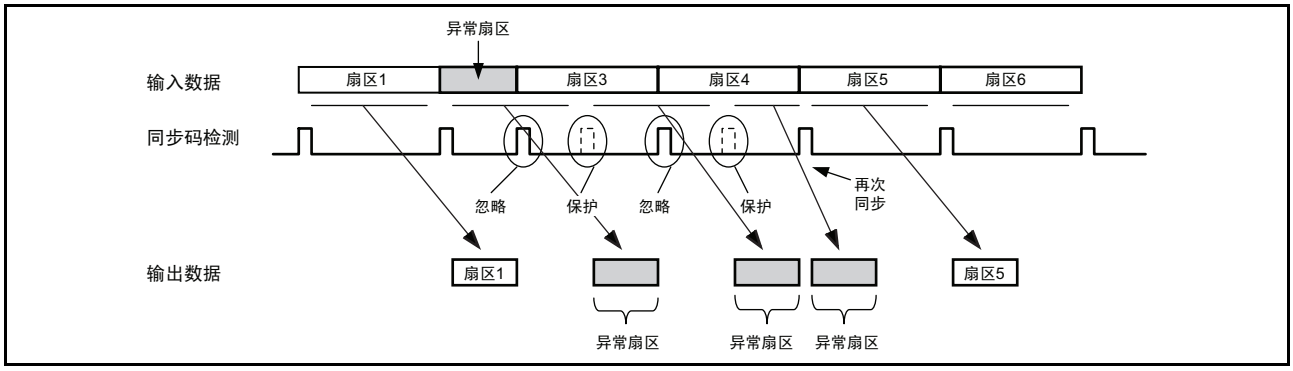


图 21.9　在短扇区的自动同步保护模式运行

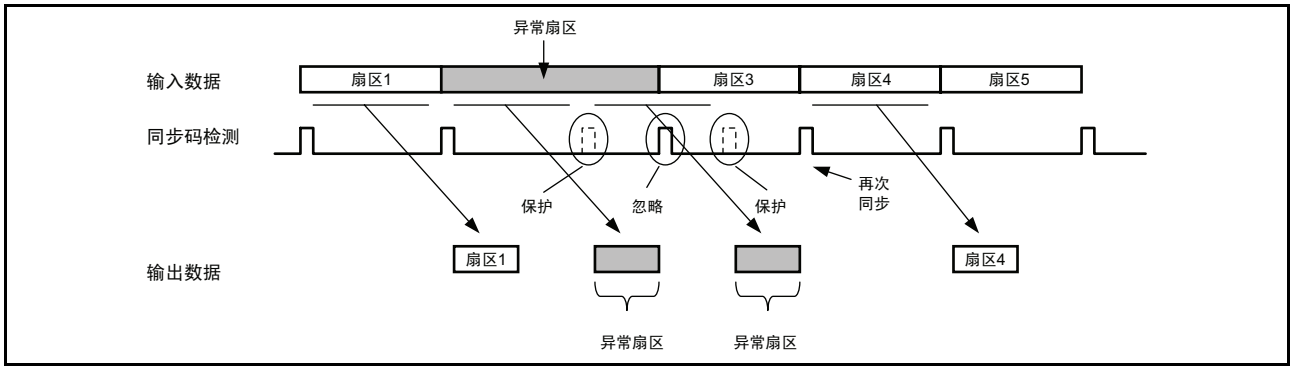


图 21.10　在长扇区的自动同步保护模式运行

（2） 外部同步模式

在外部同步模式，总是与输入数据的同步码同步。在 2352 字节没发现下个同步码时，检测到同步码前不解码。

因此在追随外部同步时序时有效。但如果输入异常同步码模式，就不能正常解码。

在外部同步模式的运行如图 21.11 所示。

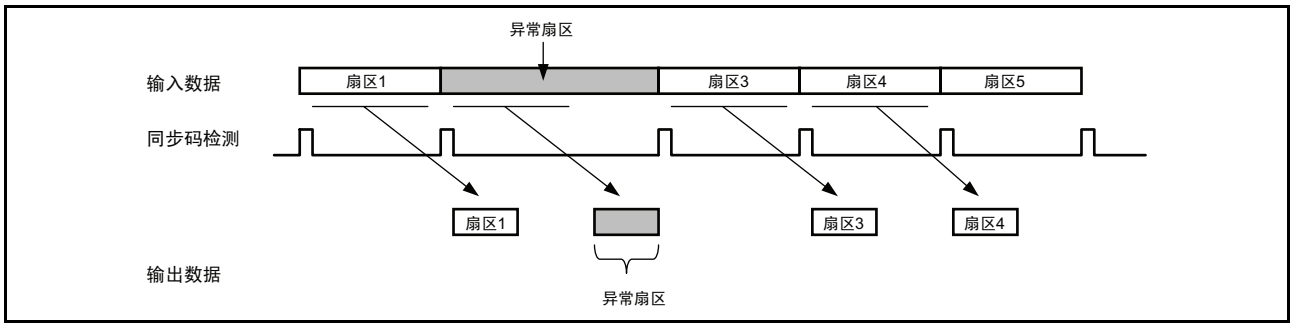


图 21.11　外部同步模式运行

（3） 内插同步模式

当检测出解码开始时的同步码模式后，内插同步模式，总是通过内部计数器同步。因此同步模式失灵时有效。

但如果同步码时序改变，就不能追随，并且不能正常解码。

在内插同步模式的运行如图 21.12 所示。

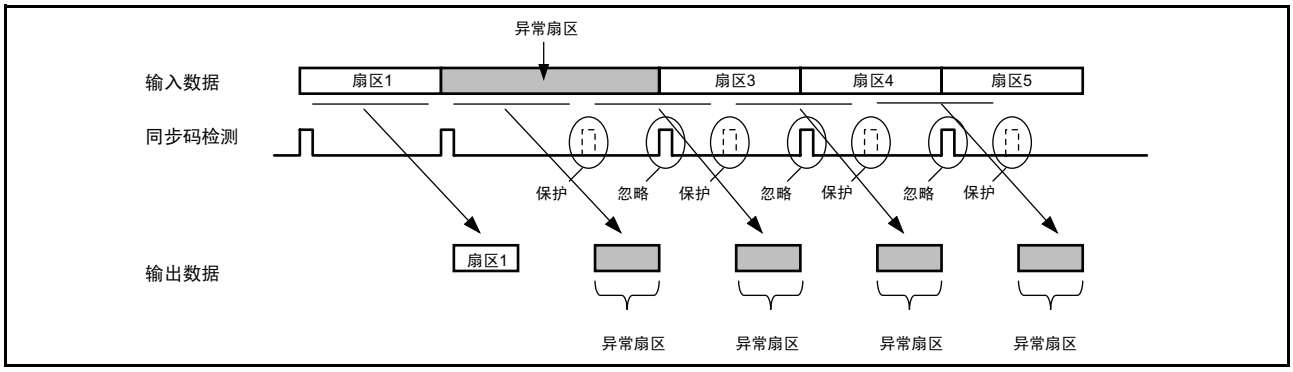


图 21.12　内插同步模式运行

（4） 内插 / 外部同步模式

在内插 / 外部同步模式，如果检测出同步码模式时就同步，在 2352 字节没检测到同步模式时保护同步码。与自动同步保护模式相比，它能够在改变同步时序后立即追随。

但如果在异常时序进入同步模式，就不能正常解码。

短扇区发生时的内插 / 外部同步模式的运行，长扇区发生时的内插 / 外部同步模式的运行分别如图 21.13、图 21.14 所示。

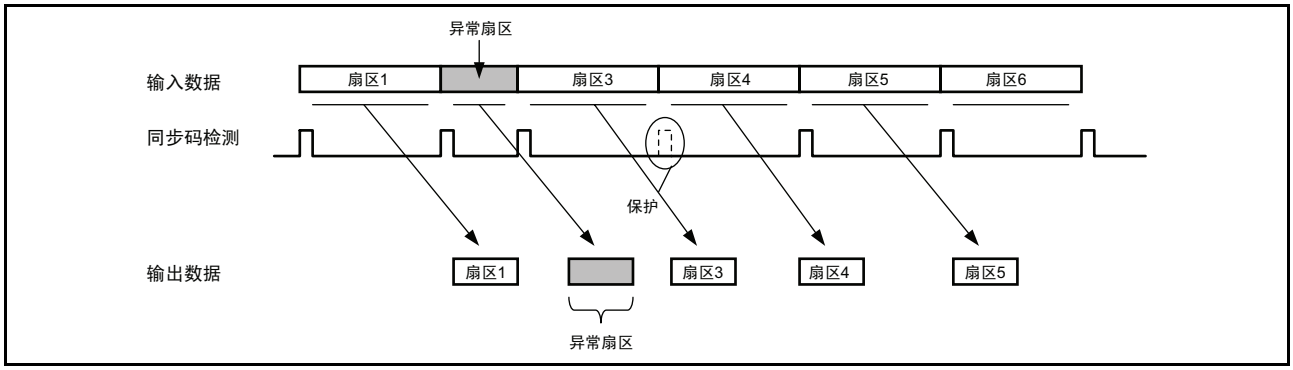


图 21.13　在短扇区的内插 / 外部同步模式运行

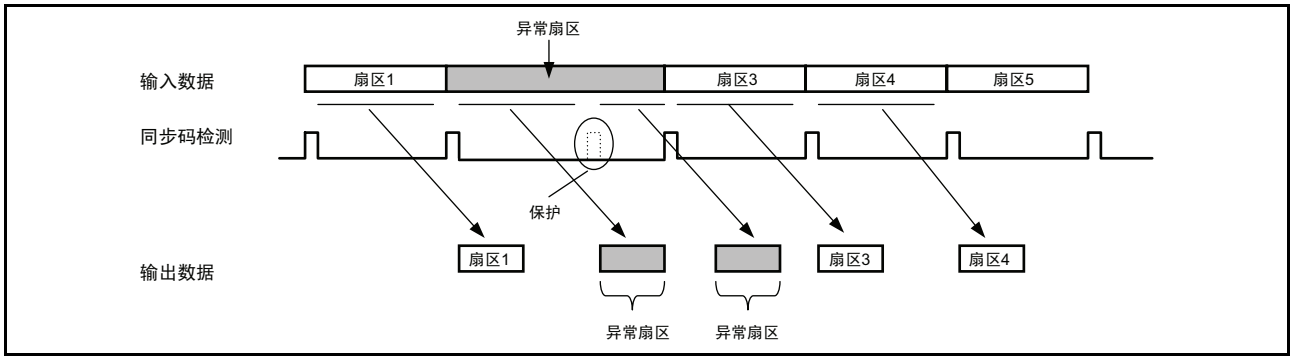


图 21.14　在长扇区的内插 / 外部同步模式运行

21.4.3　纠错

CD-ROM 解码器具有校正错误的 EDC、 P 奇偶性、 Q 奇偶性信息的格式。在本 CD-ROM 解码器中有以下的纠错功能。

- 伴随式计算功能
- ECC校正功能
- EDC校验功能

（1） 伴随式计算功能

CD-ROM 解码器为 Mode1 及 Mode2 Form1 时，输入 1 个扇区的数据后，如果判定出现错误 （伴随式计算结果不为 0），进行 ECC 校正。ECC 校正后，修正的数据的伴随式计算结果输出到 CROMST6 寄存器的 ST_ECCP 位 （P 系列）、 ST_ECCQ 位 （Q 系列）。

（2） ECC 校正及 EDC 校验

对具有 EDC、 P 奇偶性、 Q 奇偶性的 CD-ROM 格式数据，进行 ECC 校正、 EDC 校验。 ECC 的校正模式对应 P 校正、Q 校正、PQ 校正（P 校正后，Q 校正）、QP 校正（Q 校正后，P 校正），PQ 校正、QP 校正可最多重复 3 次 （通过倍速限制）校正。

在 ECC 校正前后进行 2 次 EDC 校验。

通过CROMCTL1寄存器的MD_DEC[2:0]位，设定ECC校正、EDC校验的校验模式。另外，通过CROMCTL1寄存器的 MD_PQREP[1:0] 位设定选择 PQ 校正、 QP 校正模式时的校正重复次数。

使用自动模式 / 格式检测功能时，按照判定的模式进行 ECC 校正、EDC 校验。判定为无 P 奇偶性、Q 奇偶性、无 EDC 的 Mode0、Mode2 时，不进行 ECC 校正、EDC 校验。另外，判定 Mode2 Form2 时，不进行 ECC 校正。

（a） ECC 校正

使用 ECC 校正时，如果判定为无法校正，就发生 IERR 中断，并且 CROMST6 寄存器的 ST_ECCNG 位置 1。即使检测出短扇区时，该位也置 1。

设定 CROMCTL4 寄存器的 NO_ECC 位为 1 时，如果 ECC 前的 EDC 校验为 OK，与伴随式计算结果无关，不进行 ECC 校正。

（b） EDC 校验

使用 EDC 校验时，按照设定的模式 / 格式进行 EDC 校验。另外，使用自动模式 / 格式检测功能时，按照判定的模式 / 格式，进行 EDC 校验。

ECC 校正前后的 EDC 校验结果分别反映在 CROMST6 寄存器的 ST_EDC1、 ST_EDC2 位。另外， ECC 校正后的 EDC 校验结果为 NG 时，发生 IERR 中断。

21.4.4　自动解码停止功能

在 CD-ROM 数据解码中发生异常时，可自动停止解码。

自动停止源有以下的情况。在 CROMCTL3 寄存器设定有效的源。

- 无法进行ECC校正时
- ECC后的EDC校验为NG时
- 模式、格式改变时
- MSF （分、秒、帧 （1/75秒））不连续时

因前述停止源自动停止时，输出发生停止源的扇区后，停止解码。

发生 CROMCTL3 寄存器设定的停止源，引起停止解码时，能够在 CBUFST1 寄存器确认引起自动停止的源的类型。

另外，在 CD-ROM 解码器中，有 2 个扇区的缓冲器。如果不读输出数据流，而继续输入数据流，在输入第 3 个扇区的数据时，CD-ROM 解码器停止。此时，CBUFST2 寄存器的 BUF_NG 位置 1。不发生中断。CBUFST2 寄存器的 BUF_NG 位置 1 时，由于 ROMDECRST 寄存器的 LOGICRST 位，不清除 CD-ROM 解码器功能时，就不能恢复。通过将 ROMDECRST 寄存器的 LOGICRST 位置 1，输出复位信号，并且被设定的寄存器也被清除为初始值。

21.4.5　缓冲格式

CD-ROM 解码后的输出数据流格式如图 21.15 所示。

CD-ROM 解码器的输出具有 4 字节宽度的窗口寄存器 STRMDOUT0，如果 CD-ROM 解码结束后存取该寄存器，就从同步码开始按顺序输出。

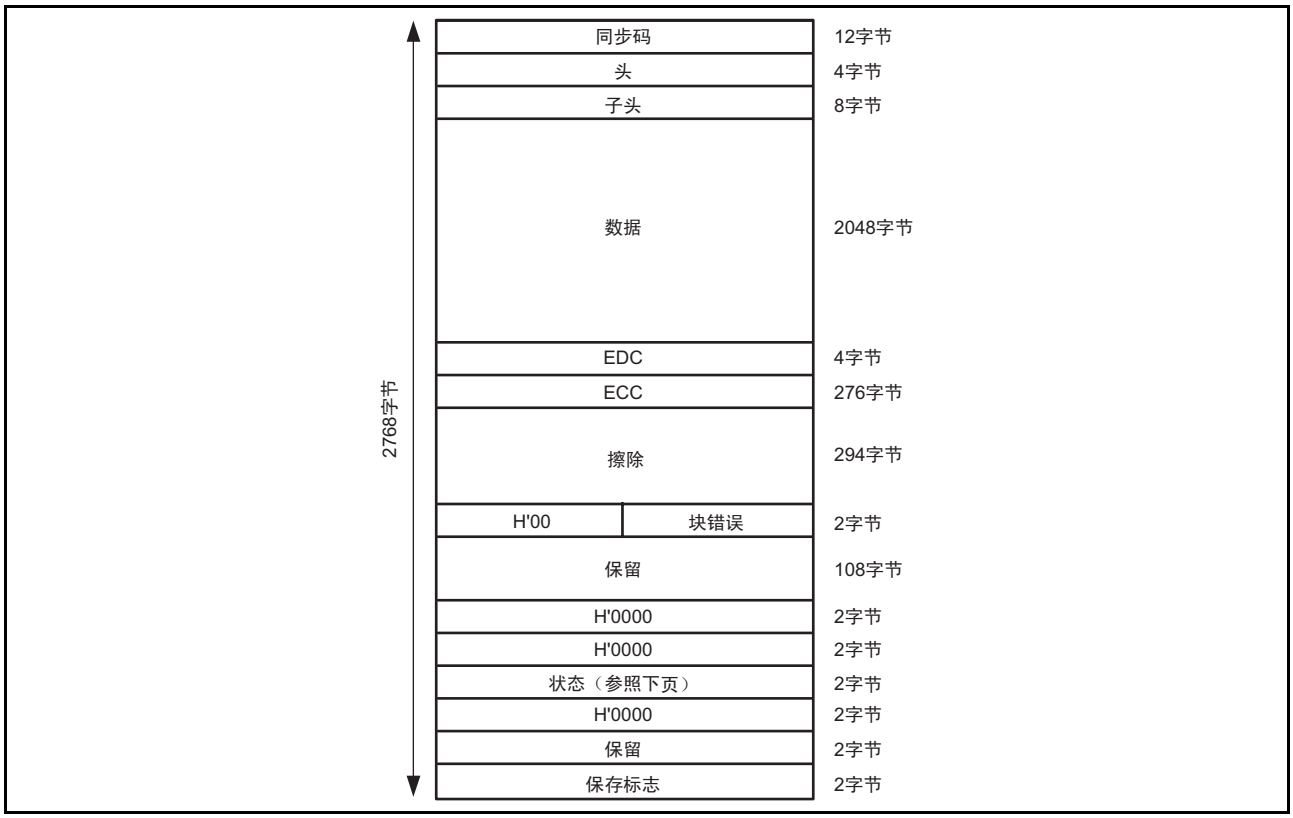


图 21.15　输出数据流格式

图 21.15 的状态的 2 字节有以下的含义。未分配的位为不定。

状态															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PERR	QERR	EDCE	—	—	—	—	—	SD	SY	FM[2:0]			HD	—	—

PERR：表示 P 系列留有错误。

QERR：表示 Q 系列留有错误。

EDCE：表示 ECC 校正后 EDC 校验为 NG。

SD：表示变为短扇区。

SY：表示内插同步码。

FM：表示数据格式。

001：Mode0

010：Mode1

011：Long （无 EDC、 ECC 格式）

100：Mode2 notXA

101：Mode2 Form1

111：Mode2 Form21

HD：扇区头的连续性 （分、秒、帧 （1/75 秒）为 NG）

每输出 1 个扇区的数据，图 21.15 中的保存标志就累加计数（H'0000 ～ H'FFFF），H'FFFF 后，返回 H'0000。

21.4.6　目标扇区缓冲功能

在 CD-ROM 解码器中，有以下 2 种指定需要输出的扇区的方法：

1. 预先设定目标扇区， CD-ROM解码器自身检测出目标扇区的自动缓冲功能
2. 软件识别在CD-ROM解码器内缓冲的扇区，同时通过软件指示目标扇区的输出的手动缓冲功能。

以下所示为进行自动以及手动缓冲的 CD-ROM 解码器内寄存器的设定方法。

（1） 自动缓冲设定

自动缓冲的设定例子如**图 21.16** 所示。

如果设定 CD-ROM 解码器寄存器，并输入数据流； CD-ROM 解码器就检测目标扇区，开始输出数据流。

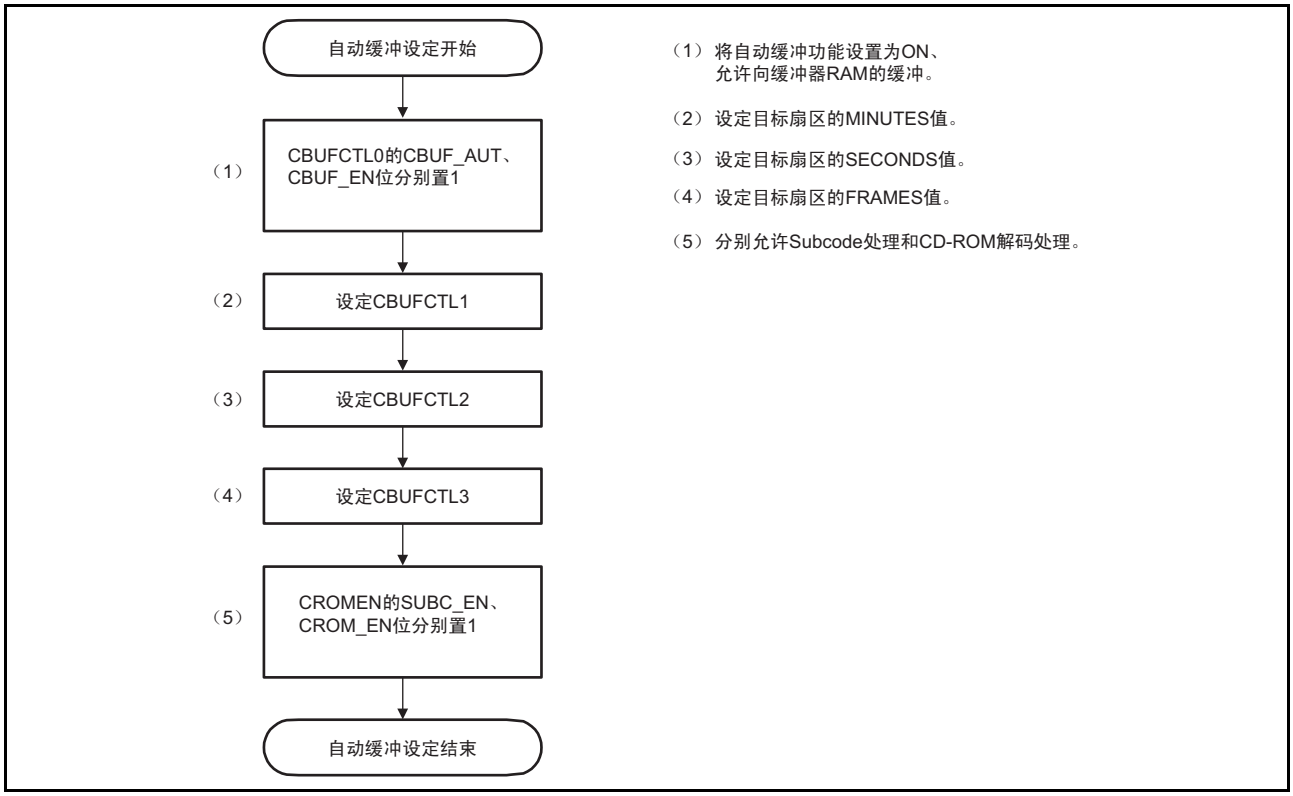


图 21.16　自动缓冲的设定例子

（2） 手动缓冲设定

手动缓冲的设定例子如图 21.17 所示。

每次产生 ISEC 中断时，判定是否有目标扇区，为目标扇区后，缓冲设置为 ON。

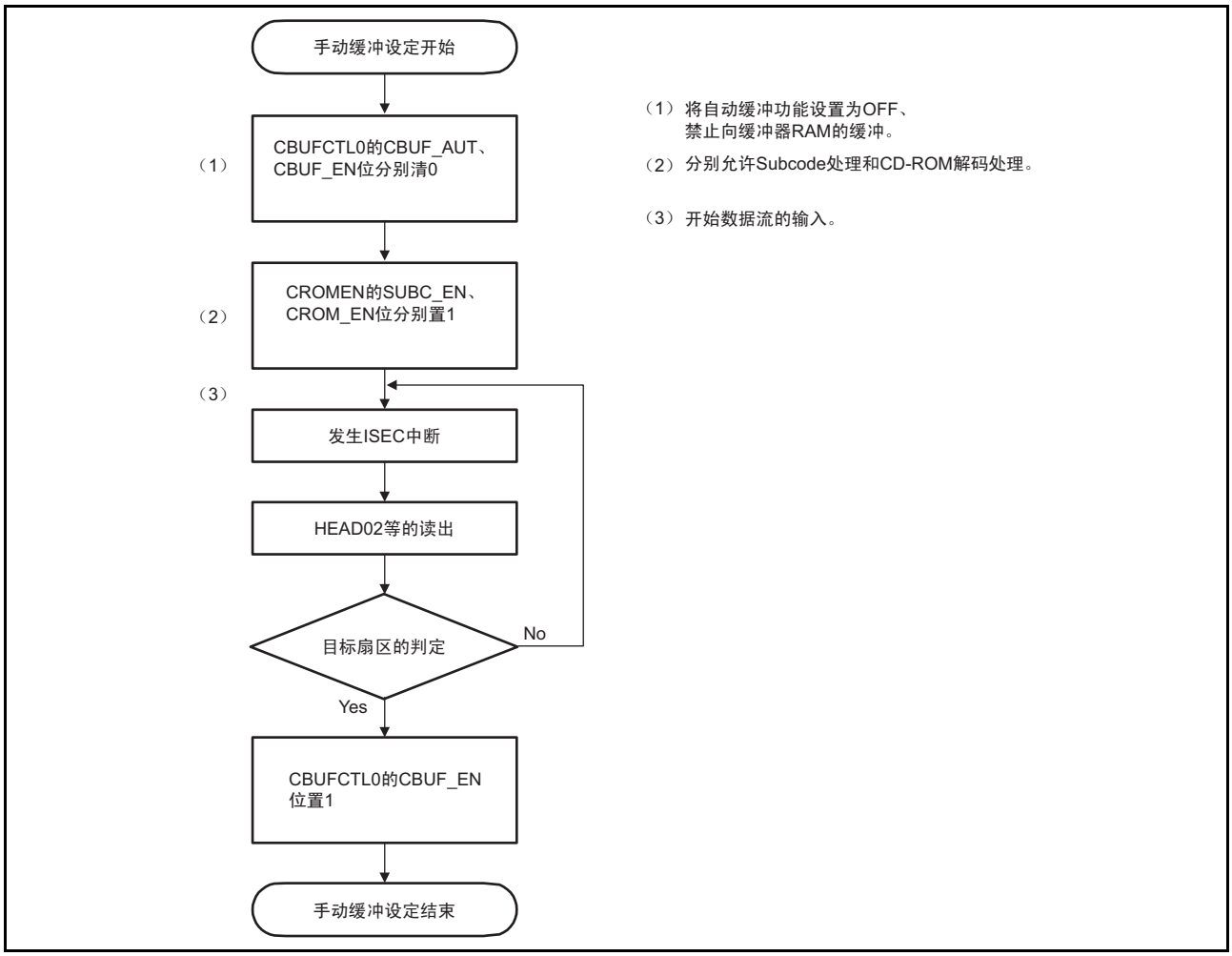


图 21.17　手动缓冲的设定例子

21.5　中断源

21.5.1　　中断及 DMA 传送请求信号

CD-ROM 解码器输出的中断信号与请求 DMA 传送的信号的含义和连接目标如表 21.3 所示。

表 21.3　中断及 DMA 请求一览

名称	条　　件	连接目标
ISEC	切换扇区时	INTC
ITARG	CD-ROM 的存取扇区不是预期的目标扇区时	INTC
ISY	CD-ROM 同步码错误时序	INTC
IERR	不能进行 ECC 校正，或因 ECC 校正后的 EDC 校验发生错误时	INTC
IBUF	向缓冲器传送数据时状态变化	INTC
IREADY	向 CD-ROM 缓冲器的数据传送请求	INTC
DMA 传送请求	向 CD-ROM 缓冲器的数据传送请求	DMAC

（1） ISEC 中断

从同步码开始 1 个扇区切换到下一个扇区时发生。

（2） ITARG 中断

从 CD-DSP 传送的数据流，不为目标扇区的数据时发生。在 CD-ROM 解码器中，检查子代码中的时间，在正常状态下，虽然期待从目标扇区稍微之前开始数据传送，但在以下情况发生中断。

- 传送目标扇区早期的数据时
- 传送目标扇区之后的数据时

该中断从子代码检测 ITARG，但在该 LSI 中从 SSI 传送 CD-ROM 数据，因此该中断无意义。

（3） ISY 中断

在以下的情况发生中断。

- 在字计数器 （同步码间隔检查计数器）不正确时检测出同步码，但忽略该同步码时
- 字计数器为最终值，但不检测同步码，进行内插 （保护）时
- 在字计数器不正确时检测出同步码，并且通过该同步码进行再同步时
- 字计数器为最终值，但不检测同步码，延长 1 个扇区的期间时
- 同步码和内插时序为短扇区时
- 同步码和内插时序为长扇区时

（4） IERR 中断

在以下的情况发生中断。

- ECC 不能校正时
- ECC 校正为 OK，但在 ECC 校正后的 EDC 检查时发生错误

（5） IBUF 中断

在以下的切换中发生中断。

- 向环形缓冲器传送数据时→数据传送结束 （搜索下一个传送数据）
- 搜索向环形缓冲器传送的数据时→数据传送开始

（6） IREADY 中断

1 个扇区的 CD-ROM 解码结束时发生。通过 CPU 向 SDRAM 缓冲输出数据流时，该中断作为启动信号使用。

（7） DMA 传送请求

DMA 的启动源和 IREADY 相同。如果准备好 1 个扇区的输出数据流，就发生中断请求。如果传送如图 21.5 所示的 2768 字节的数据并结束时，取反一次请求信号。这是由于在准备好下一个扇区的输出数据流之前需要时间，所以停止一次来自 DMAC 的数据传送请求。

21.5.2　状态寄存器的更新时序

在每次 ISEC 中断，更新 CD-ROM 解码器具有的状态信息的寄存器。通过 CROMCTL4 寄存器的 ER0SEL 位设定反映哪个扇区的信息

21.6　使用时的注意事项

21.6.1　在解码运行状态，仅缓冲停止 / 再次开始时的注意事项

保持 CD-ROM 解码器运行，但不读出输出数据流，CBUFST2 寄存器的 BUF_NG 位就为 1，之后 CD-ROM 解码器无法运行。

只有缓冲停止时，才将 CBUFCTL0 寄存器的 CBUF_EN 位设定为 0。另外，使用自动缓冲功能时，也同时将 CBUFCTL0 寄存器的 CBUF_AUT 位设定为 0。此时请读出当前缓冲中的扇区。

再次开始自动缓冲时，如果使用自动缓冲功能， CBUFCTL0 寄存器的 CBUF_AUT 位和 CBUF_EN 位也同时置 1。

21.6.2　同步码状态寄存器 （CROMST0）设定时的注意

1. CROMST0 寄存器的 ST_SECS 位为 1 时，立即停止解码，从当前解码的前 1 个扇区开始重试。
2. CROMST0 寄存器的 ST_SECL 位为 1 时，立即停止解码，从当前解码的前 2 个扇区开始重试。

21.6.3　连接块的注意

使用 ECC 校正前的扇区头信息，判定本 CD-ROM 解码器的连接块检测功能。因此，输入数据流错误时，可能错误地检测出连接块。为防止这样的错误，请通过软件实现以下的对策。

- 在缓冲中（CBUFST0 寄存器 BUF_ACT=1）、每次 ISEC 中断时确认 CROMST5 寄存器 LINK_OUT1 位，为 1 时，保存 HEAD20～23 寄存器的 MSF 值。此时，请确认是否发生 IERR 中断。如果发生 IERR 中断，就不保存 MSF 值。
- 确认前述的 LINK_OUT1 位为 1 后，在 7 个扇区 （ISEC 中断产生 7 次）间进行以下的处理。

1. 每次 ISEC 中断时，确认 CROMST5 寄存器的 LINK_ON=1，在之后的 2 个扇区间再次检测到 LINK_ON=1 时
2. 1 次也没检测出 LINK_ON=1

在前述 1.2. 时，强制结束解码，设置 CROMSY0 寄存器为外部同步模式，将上述保存的目标扇区的 MSF 值 +7（紧接连接块之后）作为目标扇区进行重试。

在设定为 CBUFCTL0 寄存器的 CBUF_LINK=0 时，起始扇区地址为 run out 的保存地址 +7。设定为 CBUF_LINK=1 时，起始扇区地址为 run out 的保存地址。

21.6.4　CD-DSP 停止 / 再次开始时的注意点

停止 / 再次开始向 CD-ROM 解码器输入数据流时，不要使停止时的数据流在同步码之前立即停止，之后再次开始时，有可能被识别为无效的数据流。这是由于停止时正在输入的数据流和再次开始时输入的数据流混在一起。所以停止 / 再次开始输入数据流时需要注意。

21.6.5　IREADY 标志清除的注意点

通过中断处理等清除 IREADY 标志时，必须在读出 1 个扇区 （2768 字节）的数据后清 0。如果在结束 1 个扇区的读出前将 IREADY 标志清 0，之后的扇区就不能解码。此时，请在恢复时将 CD-ROM 解码器模块复位控制寄存器 （ROMDECRST）的 LOGICRST 位写 1 后写 0，并进行存取。

21.6.6　在数据流传输时的注意点 （1）

数据流的读出比数据流的写入慢时，CD-ROM 解码器就陷入缓冲溢出状态，并异常停止。请一定注意数据流的写入与读出。数据流传送设定的组合例子如下所示。

表 21.4　数据流传送设定的组合例子

流输入	流输出
通过 DMA 传送 LW/ 周期挪用 （无填充）	（1）通过 DMA 传送 16 字节 / 周期挪用 （16 字节 *） （2）通过 DMA 传送突发 （16 字节 *、长字、字）
通过 DMA 传送 LW/ 周期挪用 （有填充）	（1）通过 DMA 传送周期挪用 （16 字节 *、长字） （2）通过 DMA 传送突发 （16 字节 *、长字、字）
通过 CPU 写 LW	（1）通过 DMA 传送周期挪用 （16 字节 *、长字、字） （2）通过 DMA 传送突发 （16 字节 *、长字、字）

【注】 *　将 DMA 通道控制寄存器 （CHCRn）的 bit25 置 1，并且进行正常的 16 字节传送设定。

21.6.7　在数据流传输时的注意点 （2）

必须在 DMA 或 CPU 任意一个中进行数据流的读出。当 DMA 和 CPU 的读出混在一起时，数据流可能无法识别为 CD-ROM 的格式。

第 22 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。可选择最大 8 个通道的模拟输入。

22.1　特点

- 分辨率：10位
- 输入通道：8个通道
- 最小转换时间：平均1个通道3.9μs （在Pϕ时钟33MHz运行时）
- 绝对精度：±4LSB
- 运行模式：3种
  单通道模式：1个通道的A/D转换
  多通道模式：1～4个通道的A/D转换或者1～8个通道的A/D转换
  扫描模式：1～4个通道的连续A/D转换或者1～8个通道的连续A/D转换
- 数据寄存器：8个
  将A/D转换结果保存到对应各通道的16位数据寄存器
- 带采样和保持功能
- A/D转换的开始方法：3种
  软件
  通过多功能定时器脉冲单元2 （MTU2）触发开始转换
  外部触发信号
- 中断源
  A/D转换结束时，可产生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 22.1 所示。

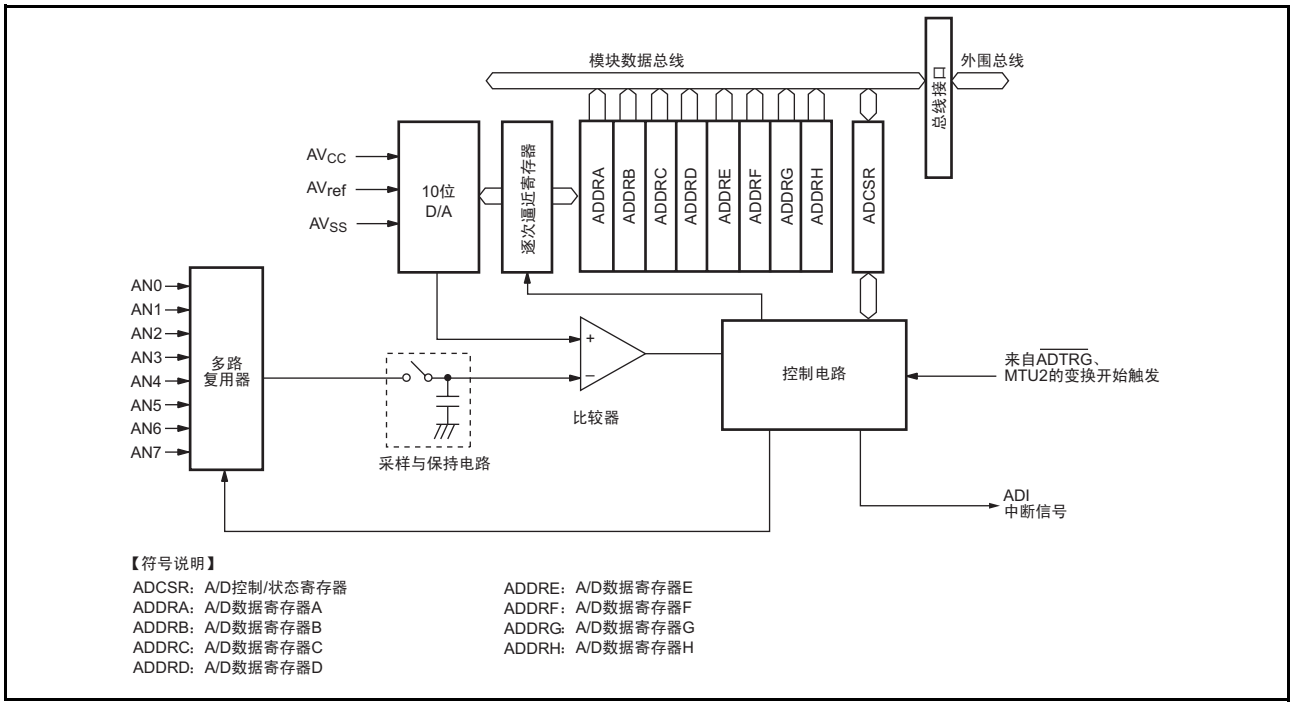


图 22.1　A/D 转换器的框图

22.2　输入 / 输出引脚

A/D 转换器使用的输入引脚如表 22.1 所示。

表 22.1　引脚结构

名称	引脚名	输入 / 输出	功　能
模拟电源引脚	AV_{CC}	输入	模拟部的电源引脚
模拟接地引脚	AV_{SS}	输入	模拟部的接地引脚以及 A/D 变换的基准接地
模拟基准电压引脚	AV_{ref}	输入	A/D 转换器的基准电压引脚
模拟输入引脚 0	AN0	输入	模拟输入
模拟输入引脚 1	AN1	输入	
模拟输入引脚 2	AN2	输入	
模拟输入引脚 3	AN3	输入	
模拟输入引脚 4	AN4	输入	
模拟输入引脚 5	AN5	输入	
模拟输入引脚 6	AN6	输入	
模拟输入引脚 7	AN7	输入	
A/D 外部触发输入引脚	$\overline{ADTRG}$	输入	开始 A/D 变换的外部触发输入

22.3　寄存器的说明

A/D 转换器有以下寄存器。

表 22.2　寄存器结构

寄存器名	简称	R/W	初始值	地址	存取尺寸
A/D 数据寄存器 A	ADDRA	R	H'0000	H'FFFE5800	16
A/D 数据寄存器 B	ADDRB	R	H'0000	H'FFFE5802	16
A/D 数据寄存器 C	ADDRC	R	H'0000	H'FFFE5804	16
A/D 数据寄存器 D	ADDRD	R	H'0000	H'FFFE5806	16
A/D 数据寄存器 E	ADDRE	R	H'0000	H'FFFE5808	16
A/D 数据寄存器 F	ADDRF	R	H'0000	H'FFFE580A	16
A/D 数据寄存器 G	ADDRG	R	H'0000	H'FFFE580C	16
A/D 数据寄存器 H	ADDRH	R	H'0000	H'FFFE580E	16
A/D 控制 / 状态寄存器	ADCSR	R/W	H'0040	H'FFFE5820	16

22.3.1　A/D 数据寄存器 A ～ H （ADDRA ～ ADDRH）

ADDR 是用于保存 A/D 转换结果的 16 位只读寄存器，有 8 个寄存器 （ADDRA ～ ADDRH）。

A/D 转换的数据为 10 位数据，传送并保存到对应转换通道的 ADDR 的 bit15 ～ 6。读取 bit5 ～ 0 时，总是读出 0。

ADDR 禁止 8 位单位的存取。请用 16 位单位存取。

模拟输入通道与 ADDR 的对应关系如**表 22.3** 所示。

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		均为 0	R	位数据 （10 位）
5 ～ 0	—	均为 0	R	保留位 读写的值总是 0。

表 22.3　模拟输入通道与 ADDR 的对应关系

模拟输入通道	保存转换结果的 A/D 数据寄存器
AN0	ADDRA
AN1	ADDRB
AN2	ADDRC
AN3	ADDRD
AN4	ADDRE
AN5	ADDRF
AN6	ADDRG
AN7	ADDRH

22.3.2　A/D 控制 / 状态寄存器 （ADCSR）

ADCSR 为 16 位可读 / 写寄存器，选择运行模式，控制 A/D 转换运行以及允许 / 禁止通过输入外部触发开始 A/D 转换。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADF	ADIE	ADST	-	TRGS[3:0]				CKS[1:0]		MDS[2:0]			CH[2:0]		
初始值：	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
R/W：	R/(W)*	R/W	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

【注】* 为清除标志，只能读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 转换结束的状态标志。 ［清除条件］ • 在 ADF=1 的状态读取 ADF 标志后，向 ADF 标志写 0 时 • 通过 ADI 中断启动 DMAC 并读取 ADDR 时 ［置位条件］ • 在单通道模式中，当 A/D 转换结束时 • 在多通道模式中，当所选通道的 A/D 转换全部结束时 • 在扫描模式中，当所选通道的 A/D 转换全部结束时
14	ADIE	0	R/W	允许 A/D 中断 允许或者禁止 A/D 转换结束引起的中断 （ADI）请求。请在转换停止时，进行 ADIE 位的设定。 0：禁止 A/D 变换结束引起的中断 （ADI）请求 1：允许 A/D 变换结束引起的中断 （ADI）请求
13	ADST	0	R/W	A/D 开始 选择 A/D 转换的开始或者停止。 A/D 转换中保持 1。 0：停止 A/D 转换 1：单通道模式时：开始 A/D 转换。指定通道的 A/D 转换结束时自动清除。 多通道模式时：开始 A/D 转换。指定的所有通道进行完一轮 A/D 转换时，自动清 0。 扫描模式时：开始 A/D 转换。直到通过软件、上电复位或者向深度待机模式、软件待机模式、模块待机模式的转移被清 0 为止，持续 A/D 转换。

位	位名	初始值	R/W	说　明
12	—	0	R	保留位 读写的值总是 0。
11 ～ 8	TRGS[3:0]	0000	R/W	定时器触发选择 允许或者禁止通过触发信号开始 A/D 转换。 0000：禁止通过外部触发开始 A/D 转换 0001：通过来自 MTU2 的转换触发 TRGAN 开始 A/D 转换 0010：通过来自 MTU2 的转换触发 TRG0N 开始 A/D 转换 0011：通过来自 MTU2 的转换触发 TRG4AN 开始 A/D 转换 0100：通过来自 MTU2 的转换触发 TRG4BN 开始 A/D 转换 1001：通过 $\overline{\text{ADTRG}}$ 开始 A/D 转换 上述以外：禁止设定
7、6	CKS[1:0]	01	R/W	时钟选择 进行 A/D 转换时间的设定。请在转换停止时（ADST=0）设定转换时间。 00：转换时间 =138 状态 （最大值） 以 Pϕ/4 运行 01：转换时间 =274 状态 （最大值） 以 Pϕ/8 运行 10：转换时间 =546 状态 （最大值） 以 Pϕ/16 运行 11：禁止设定
5 ～ 3	MDS[2:0]	000	R/W	多通道扫描模式 选择 A/D 转换的运行模式。 0xx：单通道模式 100：多通道模式。 1 ～ 4 个通道的 A/D 转换 101：多通道模式。 1 ～ 8 个通道的 A/D 转换 110：扫描模式。 1 ～ 4 个通道的 A/D 转换 111：扫描模式。 1 ～ 8 个通道的 A/D 转换
2 ～ 0	CH[2:0]	000	R/W	通道选择 该位与 ADCSR 的 MDS 位共同选择模拟输入。 MDS2=0 时 / MDS2=1、MDS0=0 时 / MDS2=1、MDS0=1 时 000：AN0 / 000：AN0 / 000：AN0 001：AN1 / 001：AN0、 AN1 / 001：AN0、 AN1 010：AN2 / 010：AN0 ～ AN2 / 010：AN0 ～ AN2 011：AN3 / 011：AN0 ～ AN3 / 011：AN0 ～ AN3 100：AN4 / 100：AN4 / 100：AN0 ～ AN4 101：AN5 / 101：AN4、 AN5 / 101：AN0 ～ AN5 110：AN6 / 110：AN4 ～ AN6 / 110：AN0 ～ AN6 111：AN7 / 111：AN4 ～ AN7 / 111：AN0 ～ AN7

【符号说明】x：Don't care

【注】 *　为清除标志，只能在读出 1 后写 0。

22.4 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、多通道模式以及扫描模式。为了避免误动作，请在 ADCSR 的 ADST 位为 0 的状态下转换运行模式或者模拟输入通道。运行模式和模拟输入通道的变更能与 ADST 位的置位同时进行。

22.4.1 单通道模式

只进行 1 个通道的 A/D 转换时，选择单通道模式。

单通道模式中，对指定的 1 个通道的模拟输入进行 1 次 A/D 转换，运行如下：

1. 如果通过软件、MTU2或者外部触发输入将ADCSR的ADST位置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. 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0。 A/D转换器进入待机状态。

在 A/D 转换中，为避免误动作，请在将 ADST 位清 0，停止 A/D 转换的状态下，切换模式以及模拟输入通道。更新之后， ADST 位置 1 时 （模式以及通道的切换与 ADST 位的置位可同时进行），再次开始 A/D 转换。

在单通道模式中，选择通道 1（AN1）的运行例如下所示。另外，此时的运行时序如**图 22.2** 所示（在运行例中指定的位属于 ADCSR 寄存器）。

1. 将运行模式设定为单通道模式，将输入通道设定为 AN1 （CH[2:0]=001），并设定 A/D 中断请求允许（ADIE=1），就开始A/D转换 （ADST=1）。
2. A/D转换结束时，将A/D转换结果传送到ADDRB。同时， ADF=1、 ADST=0， A/D转换器进入转换待机状态。
3. 因为ADF=1、 ADIE=1，所以产生ADI中断请求。
4. 开始A/D中断处理程序。
5. 读ADF=1之后，向ADF写0。
6. 读出A/D转换结果 （ADDRB），并处理。
7. 结束执行A/D中断处理程序。之后，将ADST位置1时，就开始A/D转换，执行步骤2.～7.。

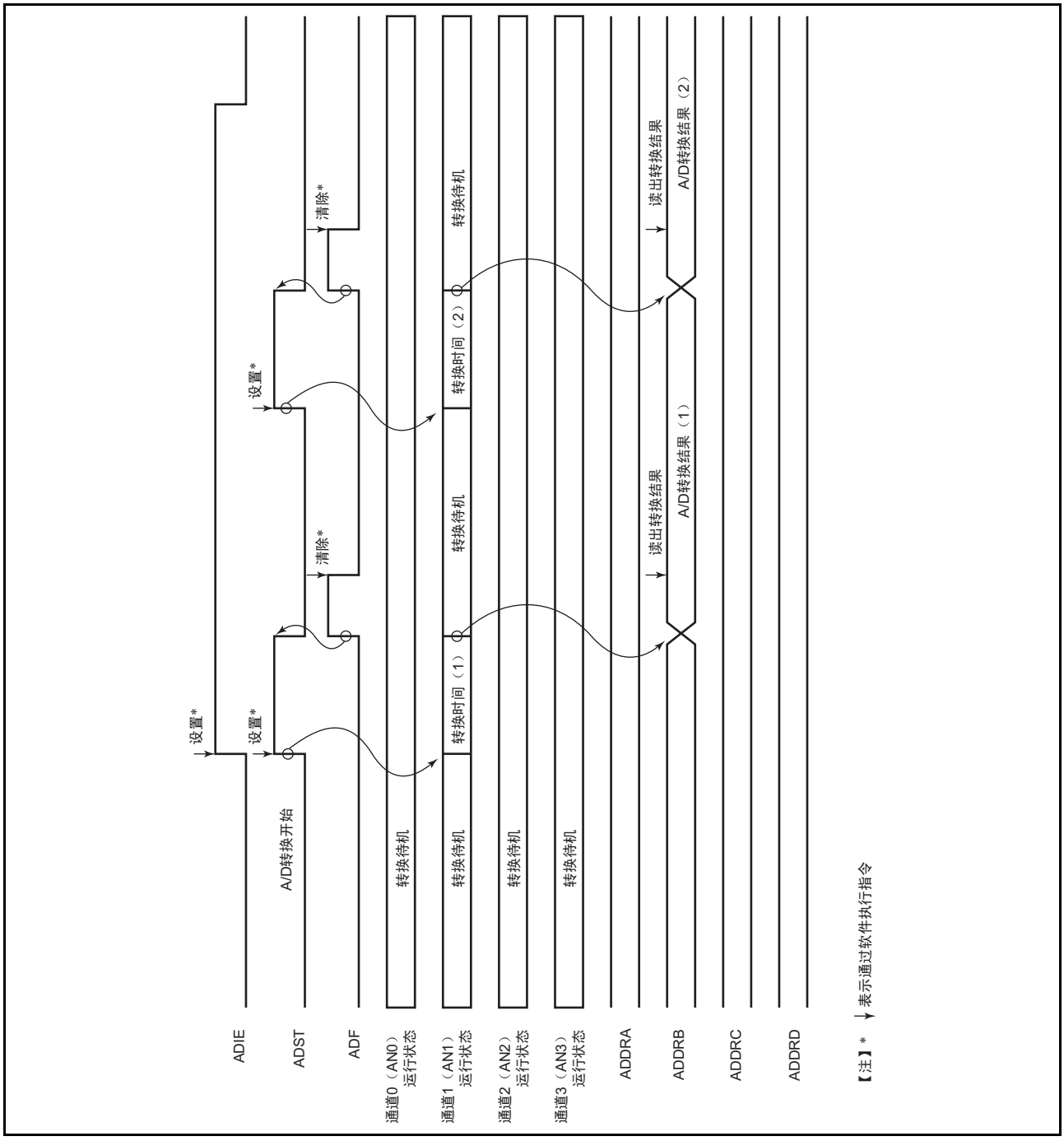


图 22.2 A/D 转换器的运行例 （单通道模式，选择通道 1 时）

22.4.2 多通道模式

多通道模式依次变换 1 次多个通道 （包含 1 个通道）的各模拟输入。

多通道模式将指定的最大 8 个通道的模拟输入进行 1 次 A/D 转换，如下：

1. 如果通过软件、 MTU2或者外部触发输入将ADCSR的ADST位置1，就按照模拟输入通道编号从小到大的顺序 （例如：AN0、 AN1…AN3）进行A/D转换。
2. 如果各通道的A/D转换结束，就将A/D转换结果按顺序传送到对应通道的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0， A/D转换器进入待机状态。 A/D转换中将ADST位清0时，就停止转换， A/D转换器进入待机状态。在读出ADF=1之后，向ADF位写0时，清除ADF位。

指定的全部通道均进行一次 A/D 转换。转换的结果传送并保存到与各通道对应的 ADDR。

在 A/D 转换中，为了避免误动作，请在将 ADST 位清 0，并停止 A/D 转换的状态下，进行模式和模拟输入通道的切换。变更后， ADST 位置 1 时 （模式及通道的变更和 ADST 位的置位可同时进行），选择第 1 通道，再次开始 A/D 转换。

在多通道模式中，选择 3 个通道 （AN0 ～ AN2）进行 A/D 转换的运行例如下所示。同时，此时的运行时序如图 22.3 所示。

1. 将运行模式设定为多通道模式（MDS[2]=1、MDS[1]=0），模拟输入通道设定为AN0～AN2（CH[2:0]=010），开始A/D转换 （ADST=1）。
2. 开始第1通道 （AN0）的A/D转换。 A/D转换结束时，将转换结果传送到ADDRA。
3. 接下来，自动选择第2通道 （AN1），开始转换。
4. 然后，按相同方式进行第3通道 （AN2）的转换。
5. 如果所选通道 （AN0～AN2）的转换全部结束， ADF=1，将ADST位置0，结束转换。
6. 此时， ADIE位是1时， A/D转换结束后，产生ADI中断。

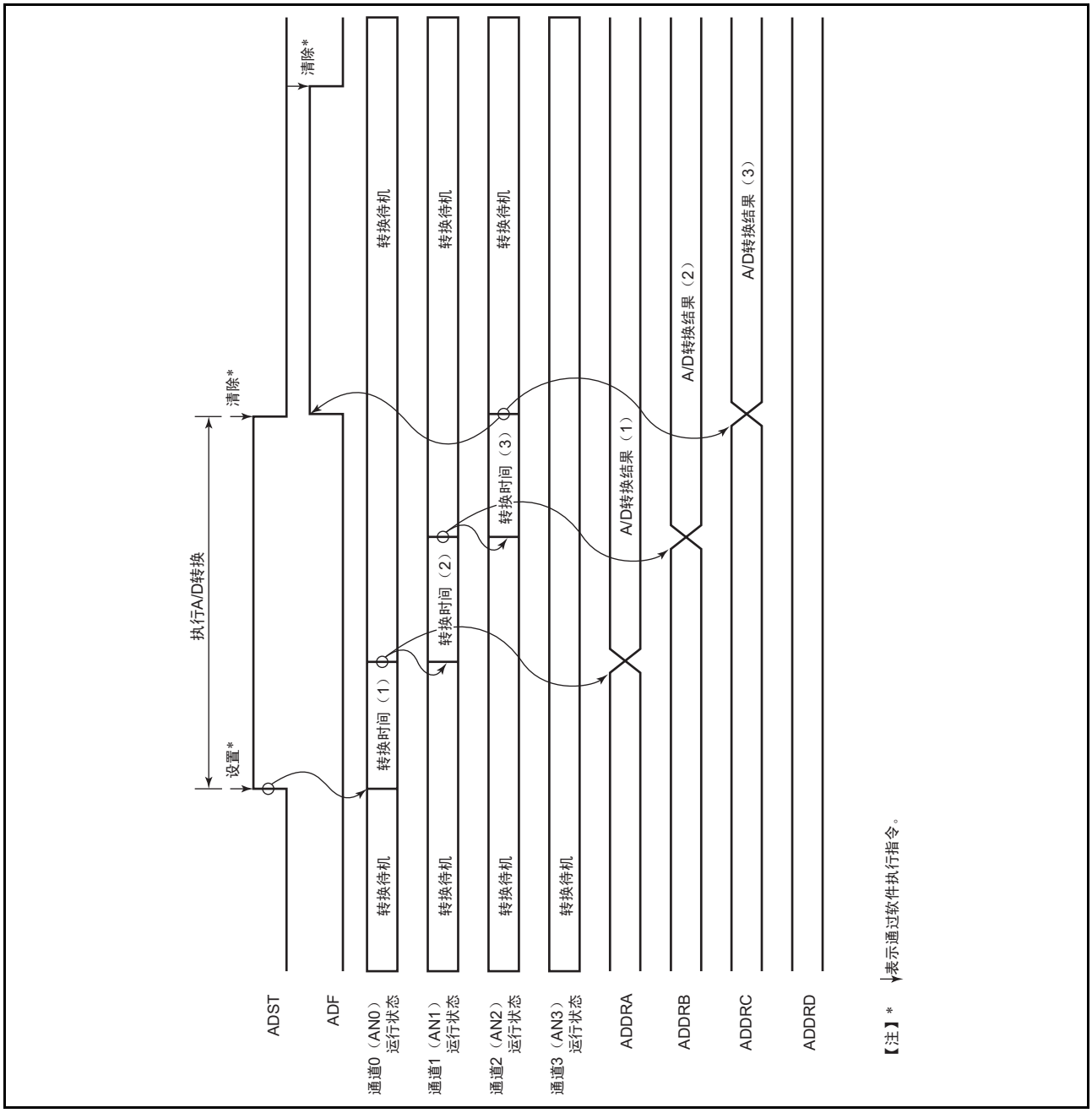


图 22.3 A/D 转换器的运行例子 （多通道模式，选择 3 个通道 （AN0 ～ AN2）时）

22.4.3 扫描模式

扫描模式适合经常监控多个通道 （包含 1 个通道）的模拟输入的系统。在扫描模式中，对指定的最大 8 个通道的模拟输入依次连续进行 A/D 转换，顺序如下：

1. 如果通过软件、 MTU2或者外部触发输入将ADCSR的ADST位置1，就按照模拟输入通道编号从小到大的顺序 （例如， AN0、 AN1…AN3）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转换器再次按照通道编号从小到大的顺序开始A/D转换。
4. ADST位不自动清除，在置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转换器进入待机状态。

读出 ADF=1 后，向 ADF 位写 0 时，清除 ADF 位。

在 A/D 转换中，为了避免误动作，请在将 ADST 位清 0，并停止 A/D 转换的状态下，进行模式和模拟输入通道的切换。变更后， ADST 位置 1 时 （模式及通道的变更和 ADST 位的置位可同时进行），选择第 1 通道，再次开始 A/D 转换。

在扫描模式中，选择 3 个通道 （AN0 ～ AN2）进行 A/D 转换的运行例子如下所示。同时，此时的运行时序如图 22.4 所示。

1. 将运行模式设定为扫描模式（MDS[2]=1、MDS[1]=1），模拟输入通道设定为AN0～AN2（CH[2:0]=010），开始A/D转换 （ADST=1）。
2. 开始第1通道 （AN0）的A/D转换。 A/D转换结束时，将转换结果传送到ADDRA。
3. 接下来，自动选择第2通道 （AN1），开始转换。
4. 然后，按相同方式进行第3通道 （AN2）的转换。
5. 如果所选通道（AN0～AN2）的转换全部结束，ADF=1，重新选择第1通道（AN0），连续进行转换。此时， ADIE位是1时， A/D转换 （第3通道的转换）结束后，产生ADI中断。
6. 不自动清除ADST位，ADST位被置1期间重复2.～4.。重复期间保持ADF=1。ADST位清0时，停止A/D转换。读出ADF=1后，向ADF位写0时，清除ADF位。

在重复 2. ～ 4. 的期间，当 ADF=1 并且 ADIE=1 时，经常产生 ADI 中断。在第 3 通道的转换结束时，如果需要产生中断，请在中断产生后，将 ADF 位清 0。

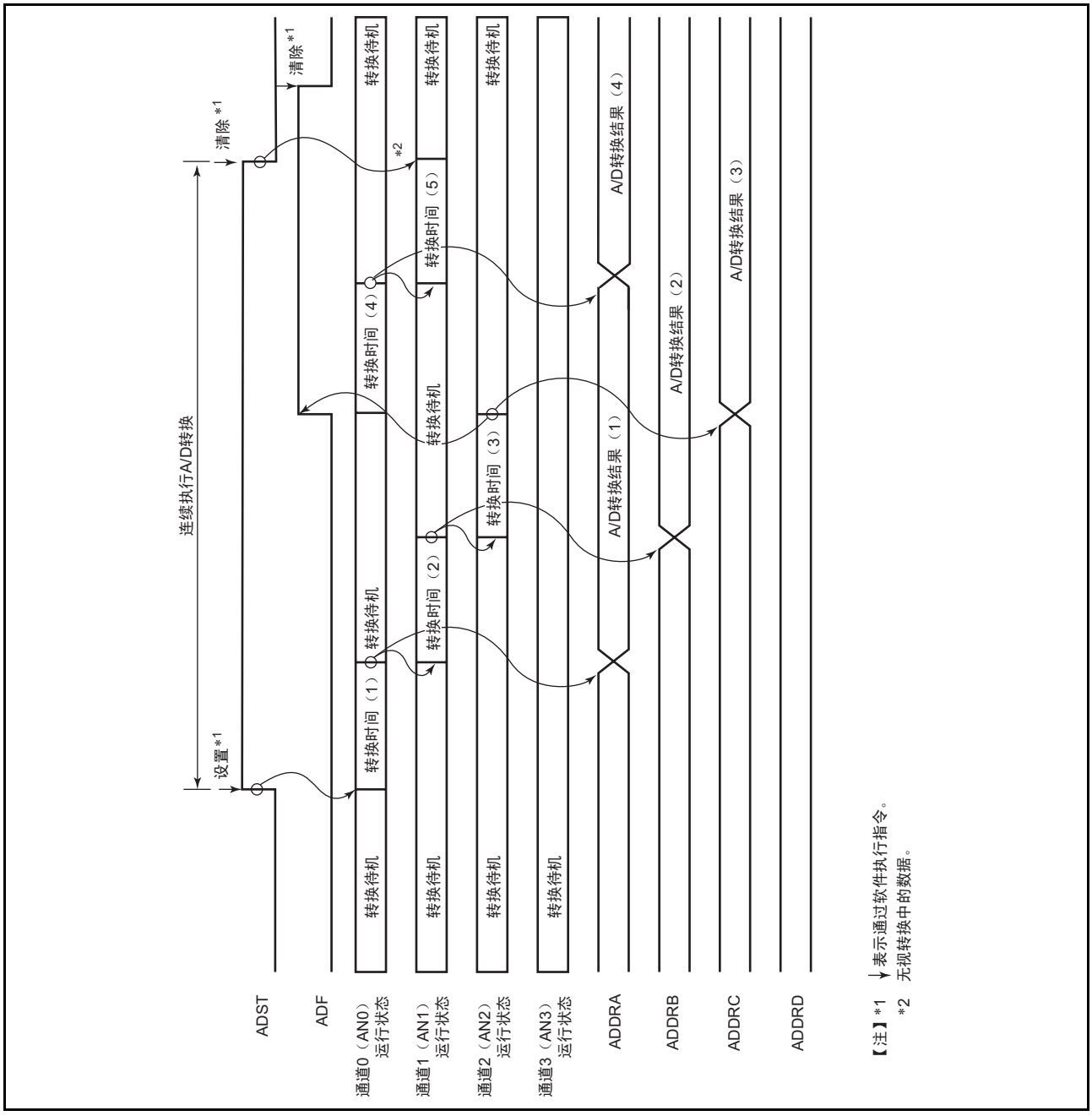


图 22.4 A/D 转换器的运行例子 （扫描模式，选择 3 个通道 （AN0 ～ AN2）时）

22.4.4　通过外部触发、 MTU2 启动 A/D 转换器

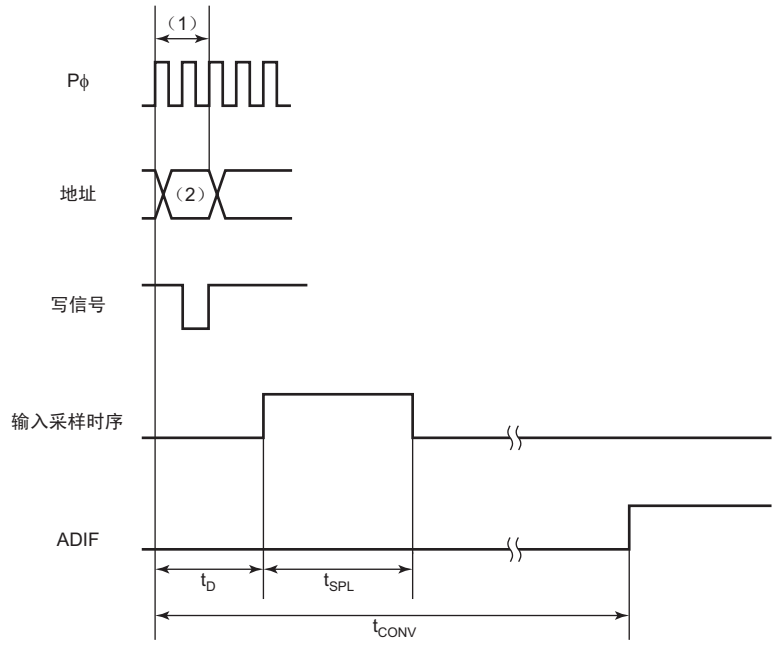
能通过外部触发、MTU2 的 A/D 转换请求，独立启动 A/D 转换器。通过外部触发、MTU2 启动 A/D 转换器时，需要设定 A/D 触发允许位 （TRGS3 ～ TRGS0）。在此状态下，如果产生外部触发、 MTU2 的 A/D 转换请求，就将 ADST 位置 1，这样就开始 A/D 转换。通过 ADCSR 的 CH2 ～ CH0 位决定转换的通道。从 ADST 位被置 1 后到 A/D 转换开始前的时序与通过软件给 ADST 位写 1 时的时序相同。

22.4.5　输入采样和 A/D 转换时间

A/D 转换器内置采样和保持电路。如果在将 ADCSR 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 22.5**、 A/D 转换时间如**表 22.4** 所示。

如**图 22.5** 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 由写入 ADCSR 的时序决定，而不是定值。因此，转换时间在**表 22.4** 所示的范围内发生变化。

关于多通道模式和扫描模式的转换时间，**表 22.4** 所示的值为第 1 次转换时间，**表 22.5** 所示的值为第 2 次及以后的转换时间。



【符号说明】
（1） ：ADCSR写周期
（2） ：ADCSR的地址
t_D ：A/D转换开始延迟时间
t_{SPL} ：输入采样时间
t_{CONV}：A/D转换时间

图 22.5　A/D 转换时序

表 22.4　A/D 转换时间 （单通道模式）

项目	符号	CKS1=0						CKS1=1		
		CKS0=0			CKS0=1			CKS0=0		
		Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.
A/D 转换开始的延迟时间	t_D	11	—	14	19	—	26	35	—	50
输入采样时间	t_{SPL}	—	33	—	—	65	—	—	129	—
A/D 转换时间	t_{CONV}	135	—	138	267	—	274	531	—	546

【注】 表中数值的单位是状态。

表 22.5　A/D 转换时间 （多通道模式 / 扫描模式）

CKS1	CKS0	转换时间 （状态）
0	0	128 （固定）
	1	256 （固定）
1	0	512 （固定）

【注】 表中数值的单位是状态。

22.4.6　外部触发输入时序

也能通过外部触发输入开始 A/D 转换。在 ADCSR 的 TRGS3 ～ TRGS0 位被置为 B'1001 时，从 $\overline{\text{ADTRG}}$ 引脚输入外部触发。在 $\overline{\text{ADTRG}}$ 的下降沿，ADCSR 的 ADST 位被置 1，开始 A/D 转换。与单通道模式 / 多通道模式 / 扫描模式无关，其他运行都与通过软件将 ADST 位置 1 的情况相同。此时序如图 22.6 所示。

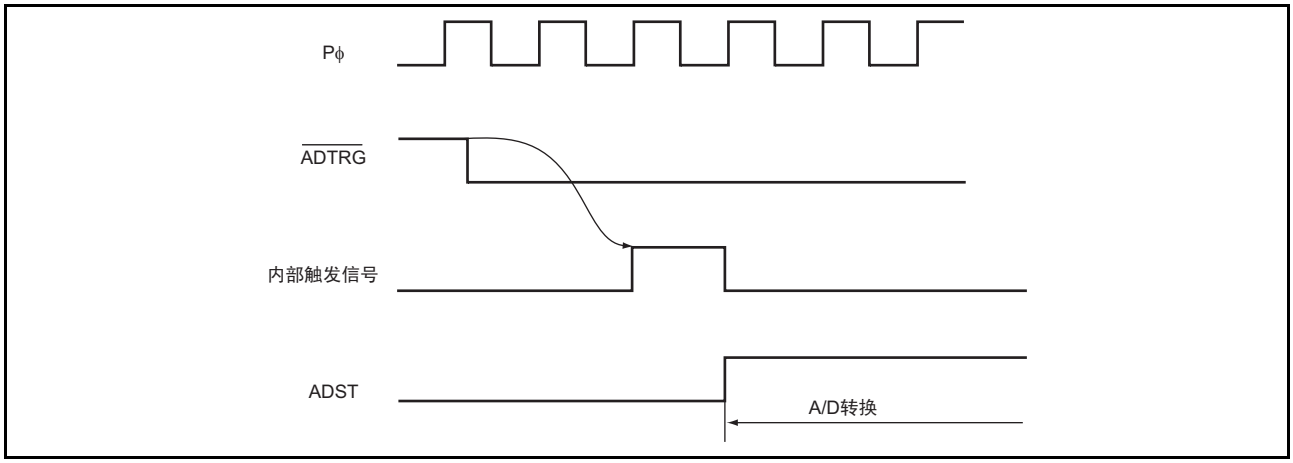


图 22.6　外部触发输入时序

22.5 中断源和 DMAC 传送请求

A/D 转换结束时， A/D 转换器产生 A/D 转换中断 （ADI）。 A/D 转换结束后， ADCSR 的 ADF 被置 1，此时，如果 ADIE 位被置 1，就产生 ADI 中断请求。另外，根据直接存储器存取控制器 （DMAC）的设定，可通过 ADI 中断启动 DMAC。此时，不对 CPU 产生中断请求。不进行 DMAC 的启动设定时，对 CPU 产生中断请求。使用 DMAC 读取通过 ADI 中断转换的数据时，可实现连续转换但不会有软件负担。

单通道模式中，请设定由 ADI 中断引起的 DMA 传送仅为 1 次。对于扫描模式或者多通道模式下的多个通道的 A/D 转换，设定 DMA 传送次数为 1 次时，只进行 1 个通道的数据传送后， DMA 传送就结束。需要通过 DMAC传送全部转换数据时，请将传送源地址设定为保存A/D转换数据的ADDR，传送次数设定为转换通道数。

如果通过 ADI 启动 DMAC，通过 DMAC 传送数据时，自动清除 ADCSR 的 ADF 位。

表 22.6 中断源和 DMAC 传送请求的关系

名称	中断源	中断标志	DMAC 的启动
ADI	A/D 转换结束	ADCSR 的 ADF	可

22.6　A/D 转换精度的定义

A/D 转换器将从模拟输入通道输入的模拟值与模拟基准电压比较，同时转换为 10 位的数字值。此时， A/D 转换的绝对精度，即输入模拟值与输出数字值的偏差包含以下误差：

1. 偏移误差
2. 满刻度误差
3. 量化误差
4. 非线性误差

通过**图 22.7**，对上述 1. ～ 4. 的误差进行说明。但是，为了使图容易理解，将 10 位的 A/D 转换器简化为 3 位的 A/D 转换器。数字输出值从最小值（0 电压）B'0000000000（图中是 000）向 B'0000000001（图中是 001）变化时，实际 A/D 转换特征与理想 A/D 转换特征的偏差（**图 22.7**（1））为偏移误差。数字输出值从 B'1111111110（图中是 110）向最大值（满量程电压）B'1111111111（图中是 111）变化时，实际 A/D 转换特征与理想 A/D 转换特征的偏差 （**图 22.7** （2））为满刻度误差。量化误差为 A/D 转换器固有的偏差，表示为 1/2LSB （**图 22.7** （3））。 0 电压到满刻度电压之间实际 A/D 转换特征与理想 A/D 转换特征的偏差 （**图 22.7** （4））为非线性误差，但不包含偏移误差、满刻度误差及量化误差。

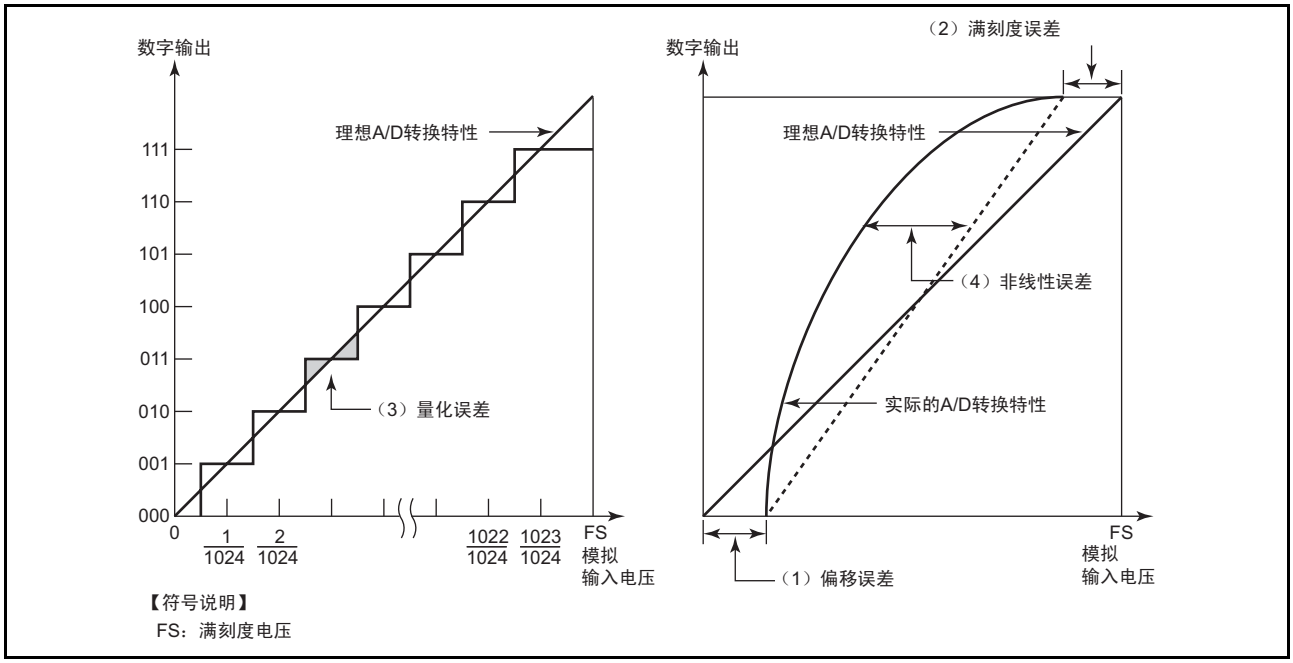


图 22.7　A/D 转换精度的定义

22.7　使用时的注意事项

使用 A/D 转换器时，请注意以下几点：

22.7.1　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 A/D 转换器的运行，初始值时，停止 A/D 转换器的运行。通过解除模块待机模式，可存取寄存器。详细内容请参照 **“第 32 章　低功耗模式”**。

22.7.2　模拟电压的设定

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。

1. 模拟输入电压的范围
   在A/D转换中，请将模拟输入引脚ANn的外加电压设定在AV_{SS}≤ANn≤AV_{CC}的范围内。（n=0～7）
2. AV_{CC}、 AV_{SS}输入电压
   AV_{CC}、 AV_{SS}的输入电压请设定为PV_{CC}－0.3V≤AV_{CC}≤PV_{CC}、 AV_{SS}=PV_{SS}。并且，在不使用A/D转换器或者D/A转换器时，以及在软件待机模式时，不能将AV_{CC}、AV_{SS}引脚设置为开路。不使用时，必须将AV_{CC}连接到电源 （PV_{CC}）， AV_{SS}接地 （PV_{SS}）。
3. AV_{ref}的设定范围
   AV_{ref}引脚的基准电压范围请设定为3.0V≤AV_{ref}≤AV_{CC}。

22.7.3　电路板设计的注意事项

在设定电路板时，请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。

模拟输入信号 （AN0 ～ AN7）、模拟基准电压 （AV_{ref}）、模拟电源 （AV_{CC}）必须通过模拟接地 （AV_{SS}）和数字电路分开，并且模拟接地 （AV_{SS}）必须单点连接到电路板上稳定的数字接地 （PV_{SS}）。

22.7.4　模拟输入引脚的处理

为了防止因过大电涌等异常电压引起对模拟输入引脚 （AN0 ～ AN7）的破坏，请连接如**图 22.8** 所示的保护电路。该图的电路也兼有 CR 过滤器功能 （抑制由噪声引起的误差）。另外，图中电路是设计例子，因此请在考虑实际的使用条件之后，决定电路常数。

模拟输入引脚的等效电路如**图 22.9** 所示，模拟输入引脚的规格如**表 22.7** 所示。

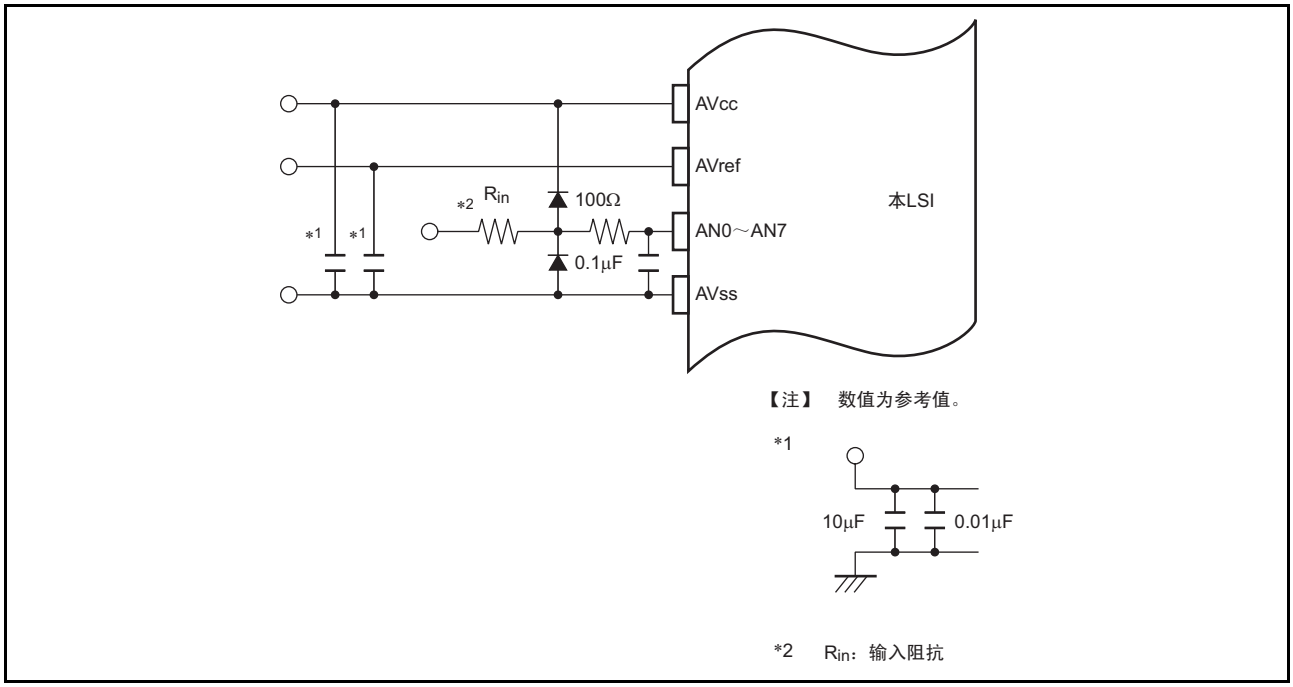


图 22.8　模拟输入引脚的保护电路例子

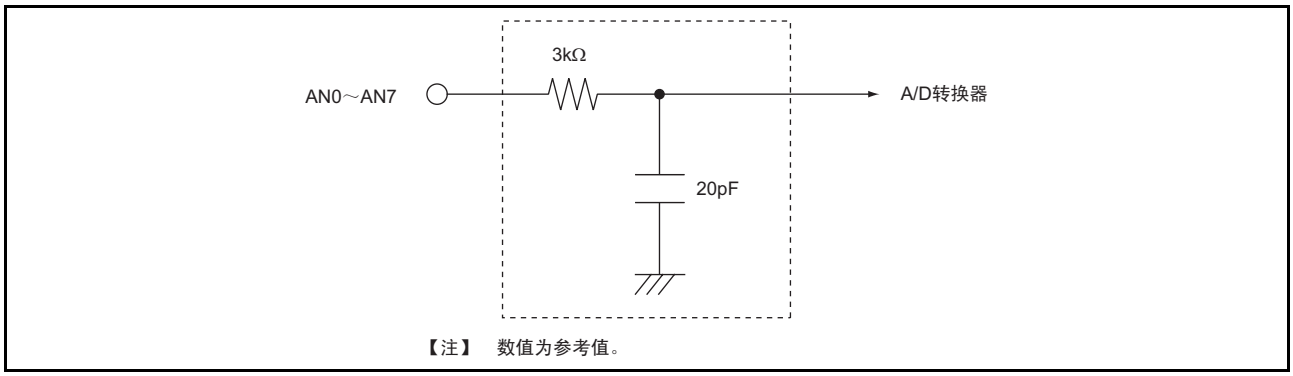


图 22.9　模拟输入引脚的等效电路

表 22.7　模拟输入引脚的规格

项目	Min.	Max.	单位
模拟输入电容	—	20	pF
容许信号源阻抗	—	5	kΩ

22.7.5　容许信号源阻抗

对于信号源阻抗不超过 5kΩ 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。传感器的输出阻抗超过 5kΩ 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外接大电容的情况下，因为输入负载实际上只有 3kΩ 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，大于等于 5mV/μs）（图 22.10）。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。

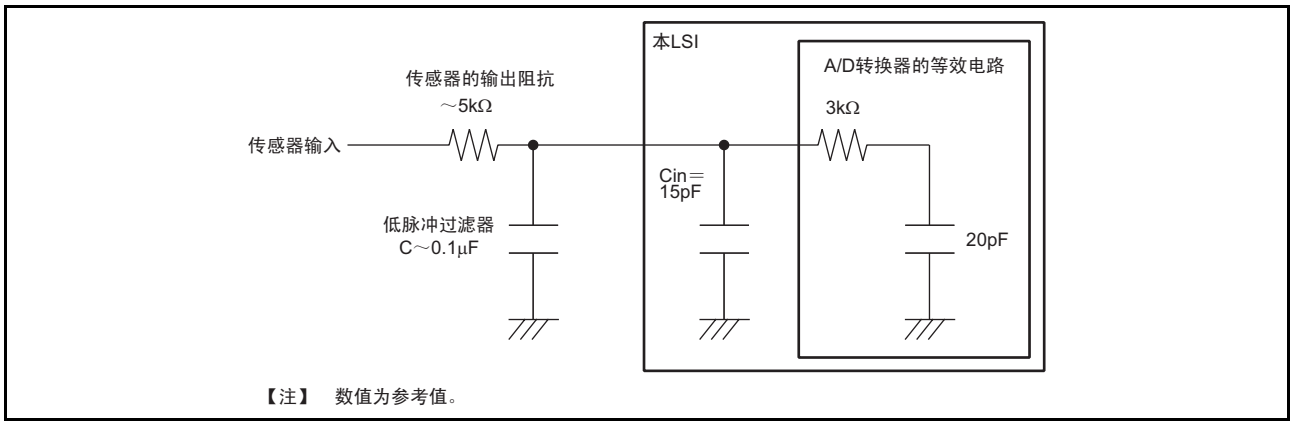


图 22.10　模拟输入电路的例子

22.7.6　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 AV_{SS} 等必须与电稳定的 GND 连接。

另外，必须注意：在安装电路板上滤波器电路不要干扰数字信号也不要充当天线。

22.7.7　深度待机模式时的 A/D 转换

转移至深度待机模式时，请通过将 ADST 位清 0 禁止 A/D 转换。在 A/D 转换为允许的状态下，本 LSI 为深度待机模式时，不能保证 A/D 的引脚状态。

22.7.8　使用扫描模式以及多通道模式时的注意事项

扫描模式以及多通道模式停止之后，立即变更到单通道模式并开始转换时，在单通道模式中有表示错误转换结果的情况。

连续进行单通道模式变换时，设置 ADST=0 之后，在超过 1 个通道的 A/D 转换时间后，请启动转换（ADST=1）。（1 个通道的转换时间根据 ADC 的分频寄存器设定会有所不同）。

第 23 章　D/A 转换器 （DAC）

23.1　特点

- 分辨率：8 位
- 输出通道：2 通道
- 最小转换时间：10μs （负载电容 20pF 时）
- 输出电压：0V ～ AV_{ref}
- 软件待机模式时的 D/A 输出保持功能
- 能设定模块待机模式

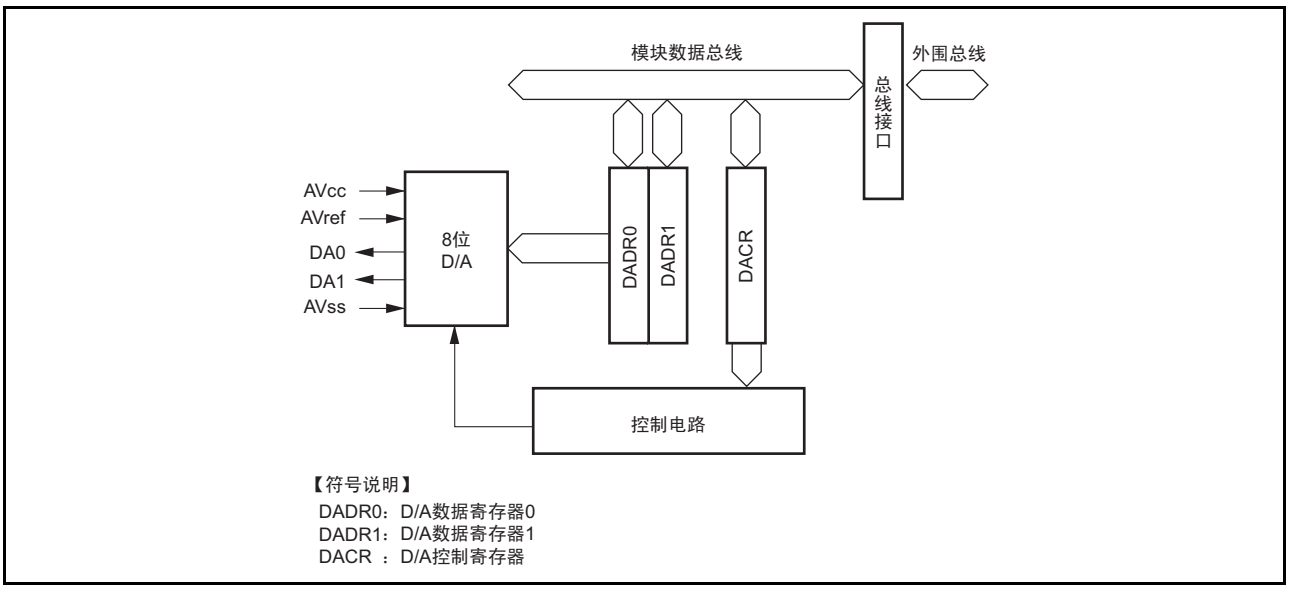


图 23.1　D/A 转换器的框图

23.2　输入 / 输出引脚

D/A 转换器使用的输入 / 输出引脚如表 23.1 所示。

表 23.1 引脚结构

名称	引脚名	输入 / 输出	功能
模拟电源引脚	AV_{CC}	输入	模拟部的电源引脚
模拟接地引脚	AV_{SS}	输入	模拟部的接地引脚
模拟基准电压引脚	AV_{ref}	输入	D/A 转换器的基准电压引脚
模拟输出引脚 0	DA0	输出	通道 0 的模拟输出
模拟输出引脚 1	DA1	输出	通道 1 的模拟输出

23.3　寄存器的说明

D/A 转换器有以下寄存器。

表 23.2　寄存器结构

寄存器名	简称	R/W	初始值	地址	存取长度
D/A 数据寄存器 0	DADR0	R/W	H'00	H'FFFE6800	8、16
D/A 数据寄存器 1	DADR1	R/W	H'00	H'FFFE6801	8、16
D/A 控制寄存器	DACR	R/W	H'1F	H'FFFE6802	8、16

23.3.1　D/A 数据寄存器 0、1 （DADR0、DADR1）

DADR 是用于保存 D/A 转换数据的 8 位可读 / 写寄存器。如果允许模拟输出，就转换 DADR 的值，并输出到模拟输出引脚。

通过上电复位及模块待机模式， DADR 初始化为 H'00。

位：	7	6	5	4	3	2	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

23.3.2　D/A 控制寄存器 （DACR）

DACR 是 8 位可读 / 写寄存器、控制 D/A 转换器的运行。

通过上电复位及模块待机模式， DACR 初始化为 H'1F。

位：	7	6	5	4	3	2	1	0
	DAOE1	DAOE0	DAE	-	-	-	-	-
初始值：	0	0	0	1	1	1	1	1
R/W：	R/W	R/W	R/W	-	-	-	-	-

位	位名	初始值	R/W	说　　明
7	DAOE1	0	R/W	D/A 输出允许 1 控制通道 1 的 D/A 转换和模拟输出。 0：禁止通道 1 的模拟输出 （DA1） 1：允许通道 1 的 D/A 转换。允许通道 1 的模拟输出 （DA1）
6	DAOE0	0	R/W	D/A 输出允许 0 控制通道 0 的 D/A 转换和模拟输出。 0：禁止通道 0 的模拟输出 （DA0） 1：允许通道 0 的 D/A 转换。允许通道 0 的模拟输出 （DA0）
5	DAE	0	R/W	D/A 允许 通过组合 DAOE0、DAOE1 位、控制 D/A 转换。通过 DAOE0、DAOE1 位控制转换结果的输出。请参照表 23.3。 0：独立控制通道 0 和通道 1 的 D/A 转换 1：统一控制通道 0 和通道 1 的 D/A 转换
4 ～ 0	—	均为 1	—	保留位 读取时总是读出 1。写无效。

表 23.3　D/A 转换的控制

bit5	bit7	bit6	说　　明
DAE	DAOE1	DAOE0	
0	0	0	禁止 D/A 转换
		1	允许通道 0 的 D/A 转换。禁止通道 1 的 D/A 转换
	1	0	允许通道 1 的 D/A 转换。禁止通道 0 的 D/A 转换
		1	允许通道 0、 1 的 D/A 转换
1	0	0	禁止 D/A 转换
		1	允许通道 0、 1 的 D/A 转换
	1	0	
		1	

23.4　运行说明

能分别单独进行 2 通道的 D/A 转换。如果 DACR 的 DAOE 位置 1，就允许 D/A 转换，并输出转换结果。进行通道 0 的 D/A 转换时运行例子如下所示。此时的运行时序如图 23.2 所示。

1. 向DADR0写入转换数据。
2. 如果DACR的DAOE0位置1，就开始D/A转换。经过t_{DCONV}时间后，通过模拟输出引脚DA0输出转换结果。改写DADR0或DAOE0位清0为止，继续输出该转换结果。以下公式表示输出值。

$$\frac{\text{DADR的内容}}{256} \times \text{AVref}$$

3. 改写DADR0后立即开始转换。经过t_{DCONV}时间后，输出转换结果。
4. 如果DAOE0位清0，就禁止模拟输出。

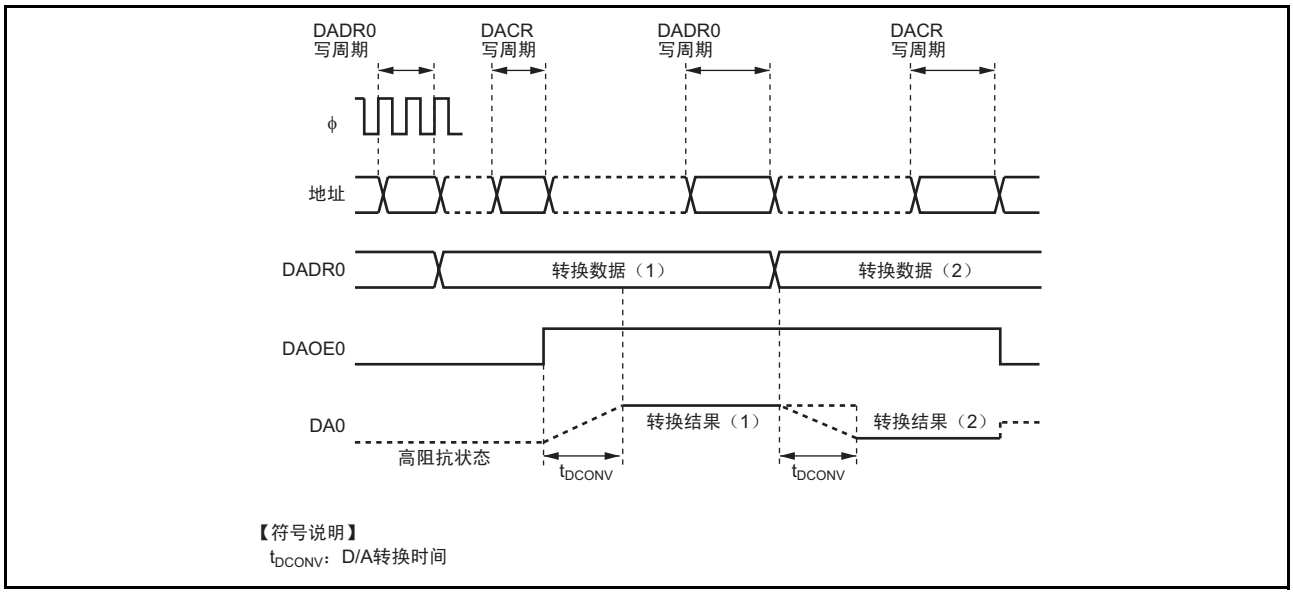


图 23.2　D/A 转换器的运行例子

23.5 使用时的注意事项

23.5.1 模块待机模式的设定

通过待机控制寄存器能禁止 / 允许 D/A 转换器的运行。初始值时，停止 D/A 转换的运行。通过解除模块待机模式，能进行寄存器的存取。详细情况请参照 “第 32 章 低功耗模式”。

23.5.2 软件待机模式时的 D/A 输出保持功能

在允许 D/A 转换的状态下，如果本 LSI 变为软件待机模式，就保持 D/A 输出，模拟电源电流与 D/A 转换时相同。在软件待机模式中需要降低模拟电源电流时，请将 DAOE0、 DAOE1、 DAE 位清 0，并禁止 D/A 输出。

23.5.3 模拟电压的设定

使用 LSI 时如果超过以下所示电压的设定范围，就会影响 LSI 的可靠性。

1. AV_{CC}、 AV_{SS}输入电压
   请将AV_{CC}、 AV_{SS}输入电压设置为PV_{CC}–0.3V ≤ AV_{CC} ≤ PV_{CC}、 AV_{SS}=PV_{SS}。而且，在软件待机模式或者不使用A/D转换器及D/A转换器时，请勿将AV_{CC}、AV_{SS}引脚设置为开路。不使用时，必须将AV_{CC}连接电源 （PV_{CC}），将AV_{SS}接地 （PV_{SS}）。
2. AV_{ref}的设定范围
   设定AV_{ref}引脚的基准电压范围为3.0V ≤ AV_{ref} ≤ AV_{CC}。

23.5.4 深度待机模式时的 D/A 转换

转移至深度待机模式时，请将 DAOE0、DAOE1、DAE 位全部清 0，并禁止 D/A 转换。在允许 D/A 转换的状态下，如果本 LSI 变为深度待机模式，就不能保证 D/A 引脚的状态。

第 24 章　AND/NAND 闪存控制器 （FLCTL）

AND/NAND 闪存控制器（FLCTL），提供外置 AND 型闪存与 NAND 型闪存的存储器接口。同时，为解决闪存特有的读出问题， FLCT 具有 ECC 符号生成以及错误检测功能。

【注】 请注意，在闪存中有多层单元 （MLC）的产品，但本 LSI 不支持。

24.1　特点

（1） AND 型闪存的存储器接口

- 可与AND型闪存直接连结的接口
- 以扇区 （512+16字节）为单位读写。执行ECC处理。
  在AND型闪存数据表中，虽然某些产品是以2048+64字节的存取单位为一页进行记述的，但在本文中512+16字节统一为1个扇区。关于1页2048+64字节的产品，将每一页分割成512+16字节单位 （1页4个扇区）进行处理。
- 以字节为单位进行读写
- 通过扩充为5字节地址，来支持超过2G位的地址

（2） NAND 型闪存的存储器接口

- 可与NAND型闪存直接连结的接口
- 以扇区 （512+16字节）为单位读写。执行ECC处理。
  在NAND型闪存数据表中，虽然某些产品是以2048+64字节的存取单位为一页记述的，但在本文中512+16字节统一为1个扇区。关于1页2048+64字节的产品，将每一页分割成512+16字节单位 （1页4个扇区）进行处理。
- 以字节为单位进行读写
- 通过扩充为5字节地址，来支持超过2G位的地址

（3） 存取模式 :FLCTL 可选择以下 2 种模式：

- 命令存取模式：
  通过在寄存器中指定本 FLCTL 对闪存发送的命令、地址、输入 / 输出数据长度，可进行连续存取。因此，可进行无ECC处理的数据的读写以及擦除
- 扇区存取模式：
  通过指定扇区执行扇区单位的读写并控制ECC符号的生成/校验。通过指定扇区数，可对连续扇区进行读写。

（4） 扇区与管理码

- 1个扇区是存取的基本单位，由512字节的数据和16字节的管理码构成。管理码中，包含8字节的ECC。
- 可通过4字节单位指定管理码内ECC的位置。
- 除ECC以外的管理码中，能写入用户信息。

（5） ECC

- 对扇区 （数据：512字节+管理码:16字节）生成8字节的ECC码以及进行错误校验。
  （但是， 16字节管理码中，生成ECC及进行错误校验的对象字节数根据设定会有所不同。）
- 纠错能力：可订正任意3处错误。
- 写入时，数据以及ECC之前的管理码为ECC符号生成对象。 ECC后的管理码不是ECC对象。
- 读取时，数据及ECC之前的管理码为ECC错误判定对象。 FIFO内管理码中的ECC通过ECC电路替换为判定结果，而不是从闪存读出的ECC码。
- 产生ECC错误时，不进行纠错。请用软件进行处理。

（6） 数据错误时

- 产生编程/擦除错误时，错误反映在错误源标志上。可指定独立源的中断。
- 产生读取错误时，管理码内的ECC不为0。此时，读取错误反映在ECC错误源标志上。
- 如果产生ECC错误，就进行纠错，并指定替代扇区。请根据需要将块的内容复制到替代扇区。

（7） 数据传送用 FIFO 和数据寄存器

- 为传送闪存数据，内置224字节的数据FIFO寄存器 （FLDTFIFO）
- 为传送管理码数据，内置32字节的管理码FIFO寄存器 （FLECFIFO）

（8） DMA 传送

- 通过向DMA控制器分别指定闪存数据和管理码的传送目的地，可向不同区域传送数据和管理码。

（9） 存取时间

- 通过分频外围时钟 （Pϕ）使用AND/NAND型闪存引脚的运行时钟 （FCLK）。可通过通用控制寄存器（FLCMNCR）的FCKSEL位和QTSEL位指定分频率。
- 变更CPG的设定时，请在通过模块停止使FLCTL停止的状态下进行。
- NAND型闪存中， FSC引脚、 $\overline{\text{FWE}}$引脚通过FCLK的运行频率运行。 AND型闪存中， FSC引脚通过FCLK的运行频率运行， $\overline{\text{FWE}}$引脚通过1/2FCLK频率运行。设定时，请勿超过所连接存储器的最大运行频率。

FLCTL 的框图如图 24.1 所示。

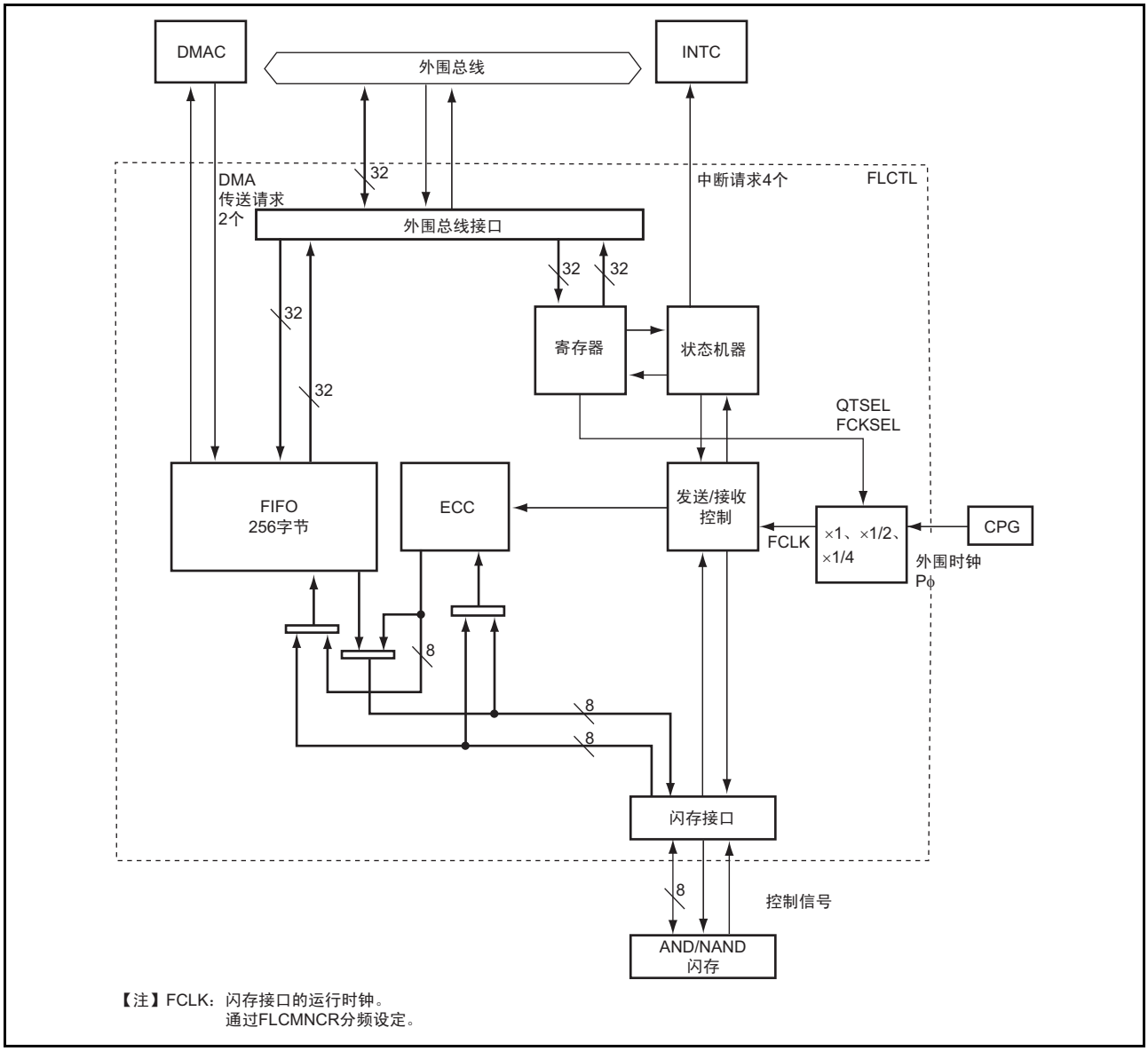


图 24.1　FLCTL 的框图

24.2　输入 / 输出引脚

FLCTL 的引脚结构如表 24.1 所示。

表 24.1　引脚结构

引脚名	输入 / 输出	相应的闪存引脚		功　能
		NAND 型	AND 型	
$\overline{\text{FCE}}$	输出	$\overline{\text{CE}}$	$\overline{\text{CE}}$	芯片选通 允许本 LSI 连接的闪存。
NAF7 ～ 0	输入 / 输出	I/O7 ～ I/O0	I/O7 ～ I/O0	数据输入 / 输出 命令、地址、数据的输入 / 输出引脚。
FCDE	输出	CLE	$\overline{\text{CDE}}$	命令锁定允许 （CLE） 命令输出时断言。 命令数据允许 （$\overline{\text{CDE}}$） 命令输出时断言。
FOE	输出	ALE	$\overline{\text{OE}}$	地址锁定允许 （ALE） 地址输出时断言。 数据输入 / 输出时取反。 输出允许 （$\overline{\text{OE}}$） 数据输入时 / 状态读时断言。
FSC	输出	$\overline{\text{RE}}$	SC	读允许 （$\overline{\text{RE}}$） 在 $\overline{\text{RE}}$ 的下降沿读数据。 串行时钟 （SC） 与 SC 同步进行数据的输入 / 输出。
$\overline{\text{FWE}}$	输出	$\overline{\text{WE}}$	$\overline{\text{WE}}$	写允许 在 $\overline{\text{WE}}$ 上升沿，闪存锁定命令、地址以及数据。
FRB	输入	R/$\overline{\text{B}}$	R/$\overline{\text{B}}$	就绪 / 忙 高电平表示就绪状态，低电平表示忙的状态。
—*	—	$\overline{\text{WP}}$	$\overline{\text{RES}}$	写保护 / 复位 可防止在低电平时，通电 / 断电时偶发的擦除 / 编程。
—*	—	$\overline{\text{SE}}$	—	备用区域允许 是用于存取备用区域的引脚。如果使用扇区存取模式，请固定为低电平。

【注】 *　本 LSI 不支持。

24.3　寄存器的说明

FLCTL 的寄存器结构如表 24.2 所示。

表 24.2　寄存器结构

寄存器名	简称	R/W	初始值	地址	存取尺寸
通用控制寄存器	FLCMNCR	R/W	H'00000000	H'FFFFF000	32
命令控制寄存器	FLCMDCR	R/W	H'00000000	H'FFFFF004	32
命令码寄存器	FLCMCDR	R/W	H'00000000	H'FFFFF008	32
地址寄存器	FLADR	R/W	H'00000000	H'FFFFF00C	32
地址寄存器 2	FLADR2	R/W	H'00000000	H'FFFFF03C	32
数据寄存器	FLDATAR	R/W	H'00000000	H'FFFFF010	32
数据计数器寄存器	FLDTCNTR	R/W	H'00000000	H'FFFFF014	32
中断 DMA 控制寄存器	FLINTDMACR	R/W	H'00000000	H'FFFFF018	32
就绪忙超时设定寄存器	FLBSYTMR	R/W	H'00000000	H'FFFFF01C	32
就绪忙超时计数器	FLBSYCNT	R	H'00000000	H'FFFFF020	32
数据 FIFO 寄存器	FLDTFIFO	R/W	H'xxxxxxxx	H'FFFFF050	32
管理码 FIFO 寄存器	FLECFIFO	R/W	H'xxxxxxxx	H'FFFFF060	32
传送控制寄存器	FLTRCR	R/W	H'00	H'FFFFF02C	8

24.3.1　通用控制寄存器 （FLCMNCR）

FLCMNCR 是 32 位可读 / 写寄存器，指定闪存的类型 （AND/NAND）及存取模式等。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	SNAND	QT SEL	-
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/W	R/W	R

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FCK SEL	-	ECCPOS[1:0]		ACM[1:0]		NAND WF	-	-	-	-	-	CE	-	-	TYPE SEL
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R	R	R/W

位	位名	初始值	R/W	说　明
31 ～ 19	—	均为 0	R	保留位 读写的值总是 0。
18	SNAND	0	R/W	大容量 NAND 闪存选择位 用于在超过 1G 位的闪存中，指定 1 页的结构为 2048+64 字节的 NAND 闪存或超过 1G 位的 AG-AND 闪存。 0：使用 1 页结构为 512+16 字节的闪存或 AND 型闪存时，请设定为 0 1：使用 1 页结构为 2048+64 字节的 NAND 型闪存或超过 1G 位的 AG-AND 闪存时，请设定为 1 【注】 TYPESEL=0 时，禁止设定为 1。
17	QTSEL	0	R/W	闪存时钟分频选择位 是闪存内使用时钟 FCLK 的分频选择位。配合 FCKSEL 使用。 QTSEL=0、 FCKSEL=0：将 CPG 提供的时钟 （Pϕ） 1/2 分频，作为 FCLK 使用 QTSEL=0、FCKSEL=1：将 CPG 提供的时钟（Pϕ）作为 FCLK 使用。 QTSEL=1、 FCKSEL=0：将 CPG 提供的时钟 （Pϕ） 1/4 分频，作为 FCLK 使用。 QTSEL=1、 FCKSEL=1：禁止设定
16	—	0	R	保留位 读写的值总是 0。
15	FCKSEL	0	R/W	闪存时钟选择位 闪存内使用时钟 FCLK 的分频选择位。 配合 QTSEL 使用。请参照 QTSEL 的说明部分。

位	位名	初始值	R/W	说　明
14	—	0	R	保留位 读写的值总是 0。
13、 12	ECCPOS[1:0]	00	R/W	ECC 位置指定位 1、 0 指定管理码区域内 ECC 的位置 （第 0/4/8 字节）。 00：在管理码区域的第 0 ～ 7 字节放置 ECC 01：在管理码区域的第 4 ～ 11 字节放置 ECC 10：在管理码区域的第 8 ～ 15 字节放置 ECC 11：禁止设定
11、 10	ACM[1:0]	00	R/W	存取模式指定位 1、 0 指定存取模式。 00：命令存取模式 01：扇区存取模式 10：禁止设定 11：禁止设定
9	NANDWF	0	R/W	NAND 等待插入运行位 0：在 1 个 FCLK 周期进行地址以及数据的输入 / 输出 1：在 2 个 FCLK 周期进行地址以及数据的输入 / 输出
8 ～ 4	—	均为 0	R	保留位 读写的值总是 0。
3	CE	0	R/W	芯片允许位 0：禁止 （向 $\overline{\text{FCE}}$ 引脚输出高电平。） 1：允许 （向 $\overline{\text{FCE}}$ 引脚输出低电平。）
2、 1	—	均为 0	R	保留位 读写的值总是 0。
0	TYPESEL	0	R/W	存储器选择位 0：选择 AND 型闪存。 1：选择 NAND 型闪存或者 AG-AND。

24.3.2　命令控制寄存器 （FLCMDCR）

FLCMDCR 为 32 位可读 / 写寄存器，能指定是否在命令存取模式发送命令、地址，还能指定数据的输入 / 输出目的地。在扇区存取模式中，能指定扇区传送次数。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADR CNT2	SCTCNT[19:16]				ADR MD	CDS RC	DOSR	-	-	SEL RW	DOA DR	ADRCNT[1:0]		DOC MD2	DOC MD1
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	R	R/W	R/W	R/W	R/W	R/W	R/W

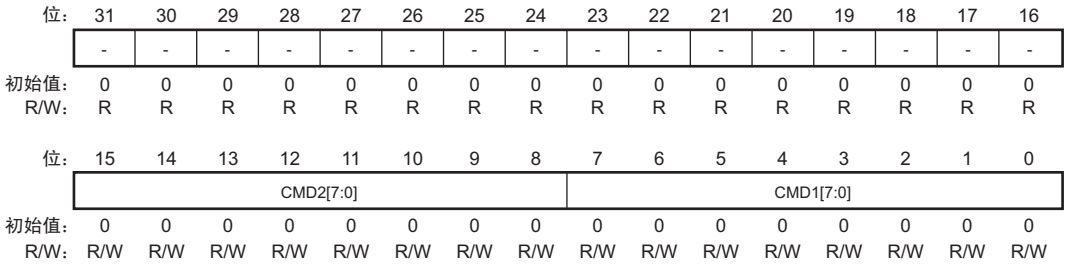
位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCTCNT[15:0]															
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说　明
31	ADRCNT2	0	R/W	地址发送字节数指定位 2 指定在地址阶段发送的地址数据的字节数。配合 ADRCNT[1:0] 位使用。 0：只发送通过 ADRCNT[1:0] 指定的字节数地址。 1：发送 5 字节的地址　ADRCNT[1:0] 请设定为 00。
30 ～ 27	SCTCNT [19:16]	均为 0	R/W	扇区传送次数指定位 [19:16] 是扇区传送次数指定位 SCTCNT[15:0] 的扩充位。 SCTCNT[19:16] 与 SCTCNT[15:0] 配合，作为 SCTCNT[19:0] 的 20 位计数器运行。
26	ADRMD	0	R/W	扇区存取地址指定位 命令存取模式时，该位无效。只有在扇区存取模式时有效。 0：地址寄存器的值作为物理扇区编号处理。扇区存取时，通常请使用该值。 1：地址寄存器的值作为闪存的地址输出。 【注】连续扇区存取时，请设定为 0。
25	CDSRC	0	R/W	数据缓冲器指定位 命令存取模式时，指定数据阶段读 / 写的数据缓冲器。 0：指定 FLDATAR 作为数据缓冲器 1：指定 FLDTFIFO 作为数据缓冲器
24	DOSR	0	R/W	状态读取检验位 命令存取模式时，在发送第 2 个命令后，指定是否进行状态读取。 0：不进行状态读取 1：执行状态读取
23、22	—	均为 0	R	保留位 读写的值总是 0。
21	SELRW	0	R/W	数据读写指定位 指定在数据阶段的读写方向。 0：读 1：写
20	DOADR	0	R/W	地址阶段执行指定位 命令存取模式时，指定是否执行地址阶段。 0：不执行地址阶段 1：执行地址阶段

位	位名	初始值	R/W	说明
19、18	ADRCNT [1:0]	00	R/W	地址发送字节数指定位 1、0 指定在地址阶段发送的地址数据的字节数。 00：发送 1 字节的地址 01：发送 2 字节的地址 10：发送 3 字节的地址 11：发送 4 字节的地址
17	DOCMD2	0	R/W	第 2 命令阶段执行指定位 命令存取模式时，指定是否执行第 2 命令阶段。 0：不执行第 2 命令阶段 1：执行第 2 命令阶段
16	DOCMD1	0	R/W	第 1 命令阶段执行指定位 命令存取模式时，指定是否执行第 1 命令阶段。 0：不执行第 1 命令阶段 1：执行第 1 命令阶段
15 ～ 0	SCTCNT [15:0]	均为 0	R/W	扇区传送次数指定位 [15:0] 指定扇区存取模式连续读出的扇区数。每结束 1 扇区的传送就进行递减计数，为 0 时停止。 与 SCTCNT[19:16] 配合使用。 命令存取模式时，运行中为 H'00001。

24.3.3　命令码寄存器 （FLCMCDR）

FLCMCDR 是 32 位可读 / 写寄存器，指定命令存取、扇区存取时发送的命令的值。

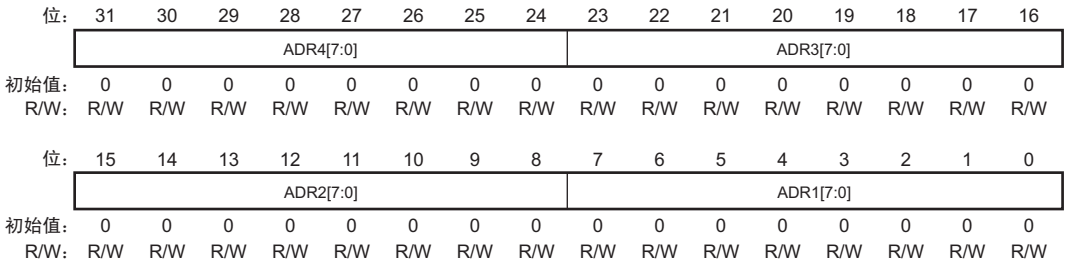


位	位名	初始值	R/W	说　明
31 ～ 16	—	均为 0	R	保留位 读写的值总是 0。
15 ～ 8	CMD2[7:0]	均为 0	R/W	第 2 命令数据位 指定在第 2 命令阶段发送的命令码。
7 ～ 0	CMD1[7:0]	均为 0	R/W	第 1 命令数据位 指定在第 1 命令阶段发送的命令码。

24.3.4　地址寄存器 （FLADR）

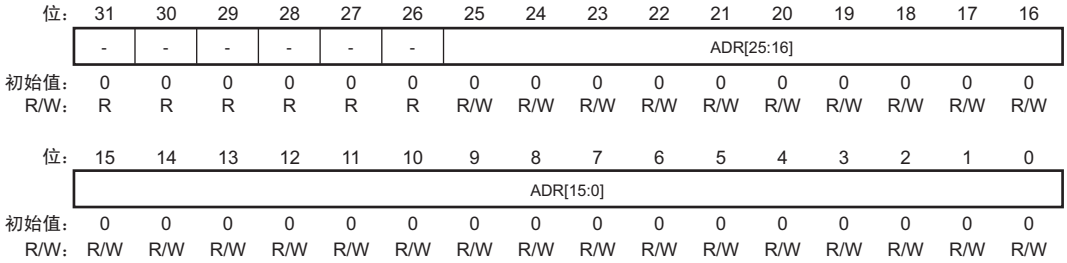
FLADR 是 32 位可读 / 写寄存器，指定要输出的地址。从 ADR1 按顺序将通过命令寄存器指定的字节数以字节为单位输出 （作为地址）。另外，可通过命令控制寄存器的扇区存取地址指定位 （ADRMD）指定是否输出将地址数据位指定的扇区编号转换为地址的值。

- ADRMD=1 时



位	位名	初始值	R/W	说　明
31 ～ 24	ADR4[7:0]	均为 0	R/W	第 4 地址数据位 ADRMD=1 时，指定作为地址第 4 次输出到闪存的数据。
23 ～ 16	ADR3[7:0]	均为 0	R/W	第 3 地址数据位 ADRMD=1 时，指定作为地址第 3 次输出到闪存的数据。
15 ～ 8	ADR2[7:0]	均为 0	R/W	第 2 地址数据位 ADRMD=1 时，指定作为地址第 2 次输出到闪存的数据。
7 ～ 0	ADR1[7:0]	均为 0	R/W	第 1 地址数据位 ADRMD=1 时，指定作为地址第 1 次输出到闪存的数据。

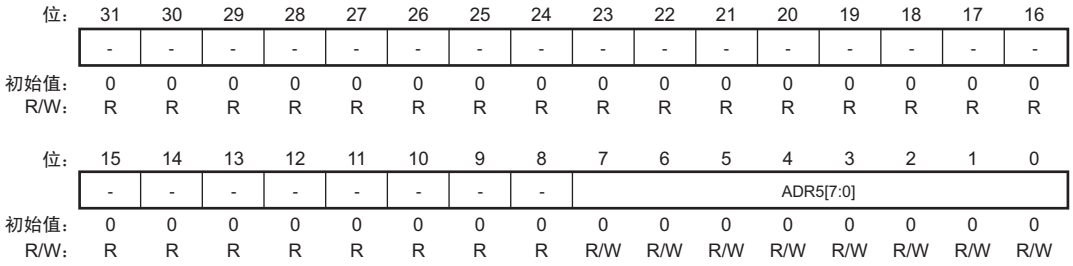
- ADRMD=0 时



位	位名	初始值	R/W	说　明
31 ～ 26	—	均为 0	R	保留位 读写的值总是 0。
25 ～ 0	ADR[25:0]	均为 0	R/W	扇区地址指定位 ADRMD=0 时，指定访问的扇区编号。扇区编号转换为地址输出到闪存。FLCMDCR 的 ADRCNT2 位为 1 时，ADR[25:0] 有效。ADRCNT2 位为 0 时， ADR[17:0] 有效。详细请参照图 24.15。 • 大块产品 （2048+64 字节） ADR[25:2] 可指定页地址，ADR[1:0] 可指定扇区单位的列地址。 ADR[1:0]=00：第 0 字节 （0 扇区） ADR[1:0]=01：第 512+16 字节 （1 扇区） ADR[1:0]=10：第 1024+32 字节 （2 扇区） ADR[1:0]=11：第 1536+48 字节 （3 扇区） • 小块产品 （512+16 字节） 仅指定页地址。

24.3.5　地址寄存器 2 （FLADR2）

FLADR2为32位可读/写寄存器，FLCMDCR的ADRCNT2位为1时有效。在命令存取模式指定要输出的地址。



位	位名	初始值	R/W	说　明
31 ～ 8	—	均为 0	R	保留位 读写的值总是 0。
7 ～ 0	ADR5[7:0]	均为 0	R/W	第 5 地址数据位 ADRMD=1 时，指定作为地址第 5 次输出到闪存的数据。

24.3.6　数据计数器寄存器 （FLDTCNTR）

FLDTCNTR 为 32 位可读 / 写寄存器。命令存取模式时，指定读写的字节数。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECFLW[7:0]								DTFLW[7:0]							
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：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	DTCNT[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
31 ～ 24	ECFLW[7:0]	均为 0	R	FLECFIFO 存取数位 表示 FLECFIFO 的可读写的长字数。 可在通过 CPU 读写 FLECFIFO 时使用。 表示读取 FLECFIFO 时， FLECFIFO 内可读出的数据的长字数。 表示写入 FLECFIFO 时， FLECFIFO 内可写入的空长字数。
23 ～ 16	DTFLW[7:0]	均为 0	R	FLDTFIFO 存取数位 表示 FLDTFIFO 的可读写的长字数。 可在通过 CPU 读写 FLDTFIFO 时使用。 表示读取 FLDTFIFO 时， FLDTFIFO 内可读出的数据的长字数。 表示写入 FLDTFIFO 时， FLDTFIFO 内可写入的空长字数。
15 ～ 12	—	均为 0	R	保留位 读写的值总是 0。
11 ～ 0	DTCNT[11:0]	均为 0	R/W	数据数指定位 在命令存取模式，指定读写数据的字节数（最大可指定 2048+64 字节）。

24.3.7　数据寄存器 （FLDATAR）

FLDATAR 为 32 位可读 / 写寄存器。

该寄存器用于保存在命令存取模式中，FLCMDCR 的 CDSRC 位设定为 0 时使用的输入 / 输出数据。读写大于等于 5 字节的连续数据时，不能使用该寄存器。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DT4[7:0]								DT3[7:0]							
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：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DT2[7:0]								DT1[7:0]							
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
31 ～ 24	DT4[7:0]	均为 0	R/W	第 4 数据位 指定通过 NAF7 ～ 0 输入 / 输出的第 4 数据。 写入时：指定写入的数据。 读出时：保存读出数据。
23 ～ 16	DT3[7:0]	均为 0	R/W	第 3 数据位 指定通过 NAF7 ～ 0 输入 / 输出的第 3 数据。 写入时：指定写入的数据。 读出时：保存读出数据。
15 ～ 8	DT2[7:0]	均为 0	R/W	第 2 数据位 指定通过 NAF7 ～ 0 输入 / 输出的第 2 数据。 写入时：指定写入的数据。 读出时：保存读出数据。
7 ～ 0	DT1[7:0]	均为 0	R/W	第 1 数据位 指定通过 NAF7 ～ 0 输入 / 输出的第 1 数据。 写入时：指定写入的数据。 读出时：保存读出数据。

24.3.8　中断 DMA 控制寄存器 （FLINTDMACR）

FLINTDMACR 为 32 位可读 / 写寄存器，设定 DMA 传送请求及中断的允许 / 禁止。各存取模式的运行开始后，产生从 FLCTL 到 DMAC 的传送请求。

同时，bit9 ～ 5 是表示有无闪存存取时的各种错误以及来自 FIFO 的传送请求的标志位，只能写 0。清除标志时，请对需要清除的标志位写 0，其他的标志位写 1。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	ECER INTE	-	-	FIFOTRG [1:0]		AC1 CLR	AC0 CLR	DREQ1 EN	DREQ0 EN
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/W	R	R	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
	-	-	-	-	-	-	EC ERB	ST ERB	BTO ERB	TRR EQF1	TRR EQF0	STER INTE	RBER INTE	TE INTE	TR INTE1	TR INTE0
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

【注】* 只有写0才有效

位	位名	初始值	R/W	说　明
31 ～ 25	—	均为 0	R	保留位 读写的值总是 0。
24	ECERINTE	0	R/W	ECC 错误中断允许位 0：禁止 ECC 错误发生时的中断 1：允许 ECC 错误发生时的中断
23、22	—	均为 0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说　明
21、 20	FIFOTRG[1:0]	00	R/W	FIFO 触发设定位 切换 FLDTFIFO、 FLECFIFO 的传送请求发生条件。 • 闪存读出时 00：当 FLDTFIFO 已保存有 4 字节数据时，向 CPU 产生中断。 或者，产生DMA传送请求。 01：当 FLDTFIFO 已保存有 16 字节数据时，向 CPU 产生中断。 或者，产生DMA传送请求。 10：当 FLDTFIFO 已保存有 128 字节数据时，向 CPU 产生中断。 或者，已保存有4字节数据时，产生DMA传送请求。 11：当 FLDTFIFO 已保存有 128 字节数据时，向 CPU 产生中断。 或者，已保存有 16 字节数据时，产生 DMA 传送请求。 【注】 * 关于 FLECFIFO，只能使用 FIFOTRG[0] 的值切换 0：当 FLECFIFO 已保存有 4 字节数据时，向 CPU 产生中断。 或者，产生DMA传送请求。 1：当 FLECFIFO 已保存有 16 字节数据时，向 CPU 产生中断。 或者，产生DMA传送请求。 • 闪存写入时 00：FLDTFIFO 有大于等于 4 字节的空白时，向 CPU 产生中断 （不能设定 DMA 传送）。 01：FLDTFIFO 有大于等于 16 字节的空白时，向 CPU 产生中断。 或者，产生DMA传送请求。 10：FLDTFIFO 有大于等于 128 字节的空白时，向 CPU 产生中断。 （不能设定DMA传送） 11：FLDTFIFO 有大于等于 128 字节的空白时，向 CPU 产生中断。 或者，有大于等于16字节的空白时，产生DMA传送请求。 【注】 * 关于 FLECFIFO，只能使用 FIFOTRG[0] 的值切换 0：FLECFIFO 有大于等于 4 字节的空白时，向 CPU 产生中断。 （不能设定DMA传送） 1：FLECFIFO 有大于等于 16 字节的空白时，向 CPU 产生中断。 或者，产生DMA传送请求。
19	AC1CLR	0	R/W	FLECFIFO 清除位 清除 FLECFIFO。改变读写方向时，请清除 FIFO。 0：保持 FLECFIFO 的值。闪存存取时，请设定为 0。 1：清除 FLECFIFO。清除后，请设定为 0。
18	AC0CLR	0	R/W	FLDTFIFO 清除位 清除数据区 FLDTFIFO。改变读写方向时，请清除 FIFO。 0：保持 FLDTFIFO 的值。闪存存取时，请设定为 0。 1：清除 FLDTFIFO 的值。清除后，请设定为 0。
17	DREQ1EN	0	R/W	FLECFIFODMA 请求允许位 选择允许 / 禁止发送来自 FLECFIFO 区域的 DMA 传送请求。 0：禁止发送来自 FLECFIFO 区域的 DMA 传送请求 1：允许发送来自 FLECFIFO 区域的 DMA 传送请求

位	位名	初始值	R/W	说　明
16	DREQ0EN	0	R/W	FLDTFIFODMA 请求允许位 选择允许 / 禁止发送来自 FLDTFIFO 区域的 DMA 传送请求。 0：禁止发送来自 FLDTFIFO 区域的 DMA 传送请求 1：允许发送来自 FLDTFIFO 区域的 DMA 传送请求
15 ～ 10	—	均为 0	R	保留位 读写的值总是 0。
9	ECERB	0	R/(W)*	ECC 错误位 表现 ECC 错误检测结果。在扇区存取模式读取闪存时，如果发生其中的 ECC 错误，本位就被置 1。 本位为标志位，因此不能写 1。为了清除标志，只能写 0。 0：没有 ECC 错误 （表示输入的 ECC 全部为 0） 1：表示发生 ECC 错误。
8	STERB	0	R/(W)*	状态错误位 表示状态读取的结果。读取状态时，如果 FLBSYCNT 的 STAT[7:0] 位的特定位为 1，本位就被置 1。 本位为标志位，因此不能写 1。为了清除标志，只能写 0。 0：没有状态错误 （表示 FLBSYCNT 的 STAT[7:0] 位的特定位为 0） 1：表示发生状态错误　关于特定位，请参照 “24.4.7 状态读”
7	BTOERB	0	R/(W)*	R/$\overline{B}$ 超时错误位 发生 R/$\overline{B}$ 超时错误时（FLBSYCNT 的 RBTIMCNT[19:0] 位在递减计数后为 0），本位被置 1。本位为标志位，因此不能写 1。为了清除标志，只能写 0。 0：没有 R/$\overline{B}$ 超时错误 1：表示发生 R/$\overline{B}$ 超时错误。
6	TRREQF1	0	R/(W)*	FLECFIFO 传送请求标志位 表示产生来自 FLECFIFO 的传送请求。 本位为标志位，因此不能写 1。为了清除标志，只能写 0。 0：表示没有产生来自 FLECFIFO 的传送请求 1：表示产生来自 FLECFIFO 的传送请求
5	TRREQF0	0	R/(W)*	FLDTFIFO 传送请求标志位 表示产生来自 FLDTFIFO 的传送请求。 本位为标志位，因此不能写 1。为了清除标志，只能写 0。 0：表示没有产生来自 FLDTFIFO 的传送请求 1：表示产生来自 FLDTFIFO 的传送请求
4	STERINTE	0	R/W	状态错误发生时的中断允许位 选择允许 / 禁止在发生状态错误时对 CPU 产生中断。 0：禁止在发生状态错误时对 CPU 产生中断 1：允许在发生状态错误时对 CPU 产生中断
3	RBERINTE	0	R/W	R/$\overline{B}$ 超时错误发生时的中断允许位 选择允许 / 禁止在发生 R/$\overline{B}$ 超时错误时对 CPU 产生中断 0：禁止在发生 R/$\overline{B}$ 超时错误时对 CPU 产生中断 1：允许在发生 R/$\overline{B}$ 超时错误时对 CPU 产生中断

位	位名	初始值	R/W	说　明
2	TEINTE	0	R/W	传送结束中断允许位 选择允许 / 禁止在传送结束 （FLTRCR 的 TREND 位）时对 CPU 产生中断。 0：禁止在传送结束 （FLTRCR 的 TREND 位）时对 CPU 产生中断 1：允许在传送结束 （FLTRCR 的 TREND 位）时对 CPU 产生中断
1	TRINTE1	0	R/W	对 CPU 的 FLECFIFO 传送请求允许位 选择允许 / 禁止在发布来自 FLECFIFO 的传送请求时对 CPU 产生中断。 0：禁止在发布来自 FLECFIFO 的传送请求时对 CPU 产生中断 1：允许在发布来自 FLECFIFO 的传送请求时对 CPU 产生中断 允许 DMA 传送时，请将本位设定为 0。
0	TRINTE0	0	R/W	对 CPU 的 FLDTFIFO 传送请求允许位 选择允许 / 禁止在发布来自 FLDTFIFO 的传送请求时对 CPU 产生中断。 0：禁止在发布来自 FLDTFIFO 的传送请求时对 CPU 产生中断 1：允许在发布来自 FLDTFIFO 的传送请求时对 CPU 产生中断 允许 DMA 传送时，请将本位设定为 0。

【注】 *　只有写 0 才有效。

24.3.9　就绪忙超时设定寄存器 （FLBSYTMR）

FLBSYTMR 为 32 位可读 / 写寄存器，设定 FRB 引脚忙时的超时时间。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	RBTMOUT[19:16]			
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/W	R/W	R/W	R/W

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RBTMOUT[15:0]															
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
31 ～ 20	—	均为 0	R	保留位 读写的值总是 0。
19 ～ 0	RBTMOUT [19:0]	均为 0	R/W	就绪忙超时位 设定忙状态的超时时间 （Pϕ 的时钟数）。设定为 0 时，不发生超时。

24.3.10　就绪忙超时计数器 （FLBSYCNT）

FLBSYCNT 为 32 位只读寄存器。

通过状态读运行，将读出的闪存状态保存到 STAT[7:0]。

FRB 引脚为忙状态时，将 FLBSYTMR 的 RBTMOUT[19:0] 位设定的超时时间复制到 RBTIMCNT[19:0] 位，并开始递减计数。RBTIMCNT[19:0] 位的值为 0 时，FLINTDMACR 的 BTOERB 位置 1，并通知发生超时错误。此时，如果通过 FLINTDMACR 的 RBERINTE 位允许中断，就能发送 FLSTE 中断。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	STAT[7:0]								-	-	-	-	RBTIMCNT[19:16]			
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
	RBTIMCNT[15:0]															
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

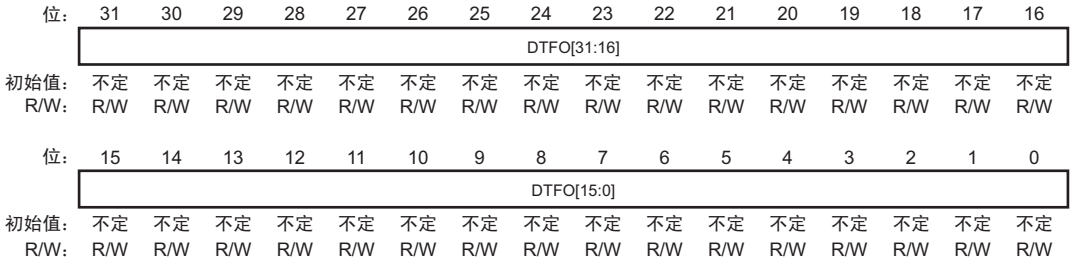
位	位名	初始值	R/W	说　明
31 ～ 24	STAT[7:0]	H'00	R	表示通过状态读取得到的闪存状态值。
23 ～ 20	—	均为 0	R	保留位 读取时总是读出 0。
19 ～ 0	RBTIMCNT [19:0]	均为 0	R	就绪忙超时计数器位 FRB 引脚为忙状态时，FLBSYTMR 的 RBTMOUT[19:0] 位的设定值复制到本位。 此后 FRB 引脚为忙状态时，本位的值被递减计数，如果为 0，就发生超时错误。

24.3.11　数据 FIFO 寄存器 （FLDTFIFO）

FLDTFIFO 是用于读写数据 FIFO 区域的寄存器。

DMA 传送时，请指定本寄存器为数据的传送目的地 （源）。16 字节 DMA 传送时，请从 16 字节地址边界的地址存取 FLDTFIFO。

请使 FLCMDCR 的 SELRW 位指定的读写方向与本寄存器的读写存取的方向相同。同时，如果改变读写方向，请在通过 FLINTDMACR 的 AC0CLR 位清除 FLDTFIFO 之后使用。



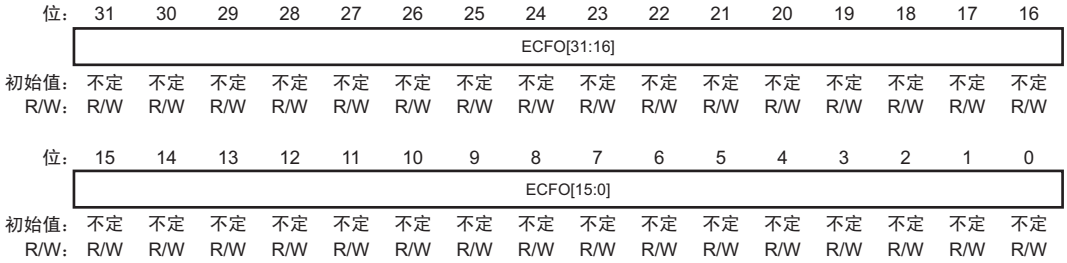
位	位名	初始值	R/W	说　明
31 ～ 0	DTFO[31:0]	H'xxxxxxxx	R/W	用于读写数据 FIFO 区域的寄存器 写入时：将数据写入数据 FIFO 区域。 读出时：从数据 FIFO 区域读出数据。

24.3.12　管理码 FIFO 寄存器 （FLECFIFO）

FLECFIFO 是用于读写管理码 FIFO 区域的寄存器。

DMA 传送时，请指定本寄存器为管理码的传送目的地 （源）。16 字节 DMA 传送时，请从 16 字节地址边界的地址存取 FLECFIFO。

请使 FLCMDCR 的 SELRW 位指定的读写方向与本寄存器读写存取方向相同。同时，如果改变读写方向，请通过 FLINTDMACR 的 AC1CLR 位清除 FLECFIFO 后使用。



位	位名	初始值	R/W	说　明
31 ～ 0	ECFO[31:0]	H'xxxxxxxx	R/W	用于读写管理码 FIFO 区域的寄存器 写入时：将数据写入管理码 FIFO 区域。 读出时：从管理码 FIFO 区域读出数据。

24.3.13　传送控制寄存器 （FLTRCR）

通过将 TRSTRT 位置 1，开始存取闪存。可通过 TREND 位确认存取是否结束。在传送中 （从 TRSTRT 位被置 1 到 TREND 位 1 被置 1 的期间）不能强制结束 （TRSTRT 位置 0）。同时，读取闪存时，在完成闪存的读取时将 TREND 置位。如果在 FIFO 里残留读出数据，就不能在 FIFO 的读取结束前强制结束。

位：	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	TREND	TRSTRT
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	—	均为 0	R	保留位 读写的值总是 0。
1	TREND	0	R/W	处理结束标志位 表示指定存取模式的处理已经结束。 写入时，请写 0。
0	TRSTRT	0	R/W	传送开始位 TREND 为 0 时，通过将 TRSTR 从 0 设定到 1，开始存取模式指定位 ACM[1:0] 指定的存取模式的处理。 0：传送停止 1：传送开始

24.4　运行说明

24.4.1　存取步骤

FLCTL 的存取分成一连串的存取步骤以及独立的阶段。

例如，对 AND 型闪存进行写入时，有以下 5 个阶段。

- 第1命令发送阶段 （写入设置命令）
- 地址发送阶段 （写入地址）
- 数据阶段 （输出）
- 第2命令发送阶段 （写入启动命令）
- 状态读

按此 5 个阶段的顺序执行存取，最后的阶段 （此时为状态读）结束时，结束对闪存的存取。

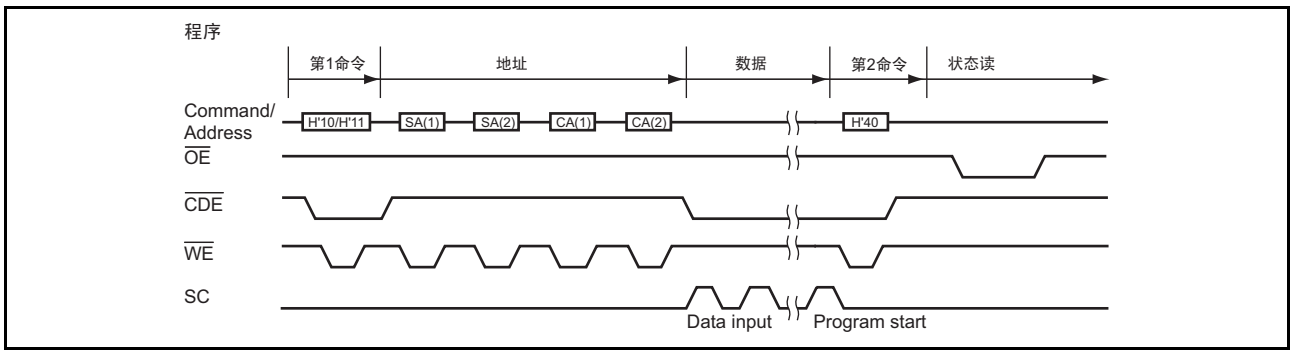


图 24.2　AND 型闪存的写操作按阶段区分的例子

关于详细情况以及 AND 型闪存的读， NAND 型闪存的读 / 写操作，请参照 “24.4.4 命令存取模式”。

24.4.2　运行模式

运行模式有命令存取模式与扇区存取模式 2 种。在扇区存取模式执行 ECC 的生成 / 错误校验。

24.4.3　寄存器设定步骤

存取闪存时所必需的寄存器设定流程如图 24.3 所示。

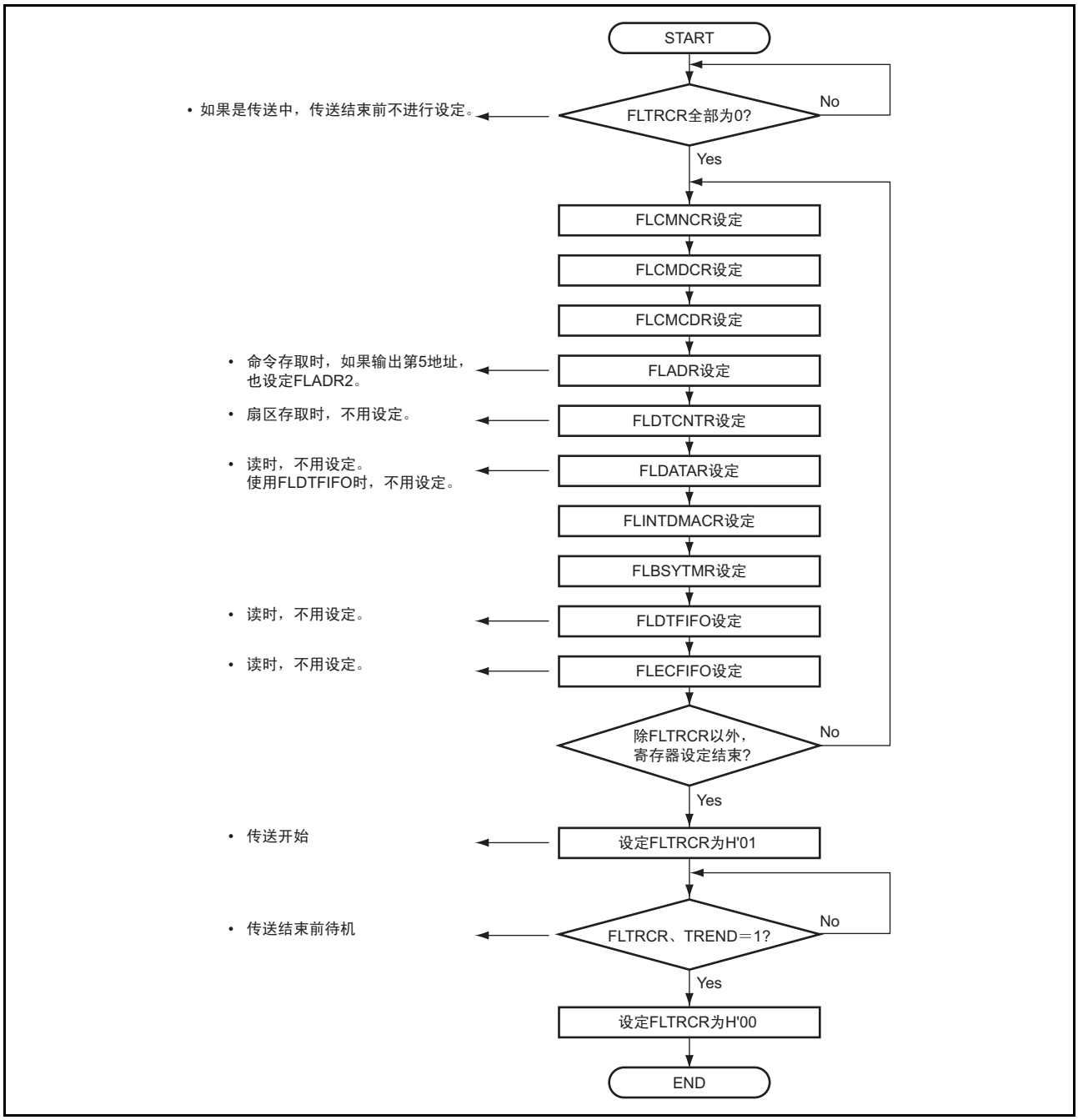


图 24.3　寄存器设定流程

24.4.4　命令存取模式

命令存取模式是通过向寄存器设定对闪存发送的命令、地址、数据、读 / 写方向以及次数等，对闪存进行存取的模式。输入 / 输出数据可使用 FLDTFIFO 进行 DMA 传送。

（1） AND 型闪存的存取

对 AND 型闪存进行读操作的例子如图 24.4、图 24.5 所示。本例中，指定 H'00 为第 1 条命令，指定地址数据长度为 2 字节 SA1、SA2（只指定 SA1、2，不指定 CA1、2）。在数据计数器指定 4 字节为读字节数，并指定 H'FF 为第 2 条命令。

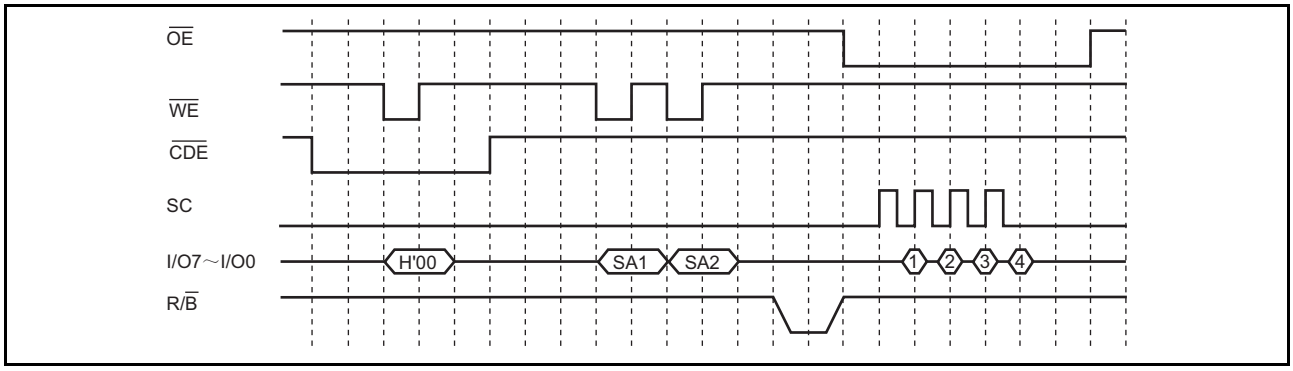


图 24.4　AND 型闪存的读操作时序 （1）

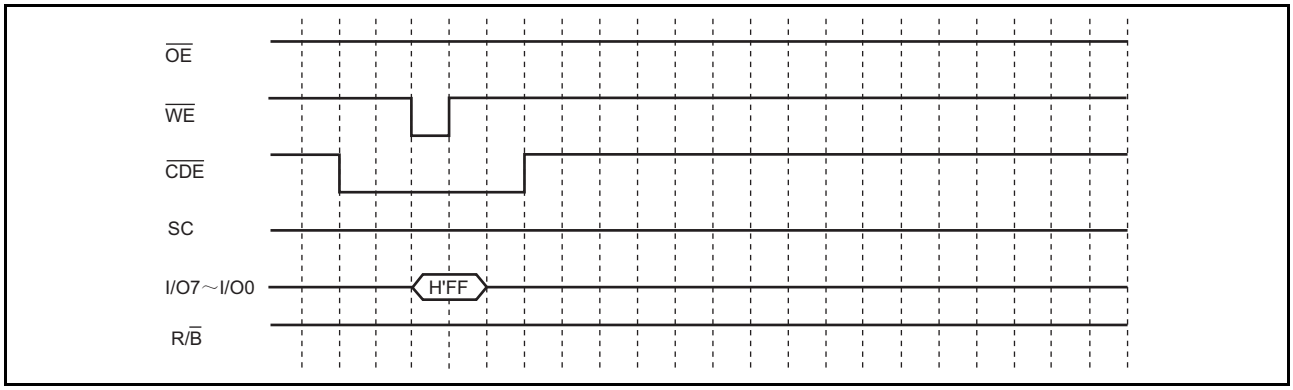


图 24.5　AND 型闪存的读操作时序 （2）

其次，如果对 AND 型闪存进行写操作时的波形如图 24.6、图 24.7 所示。

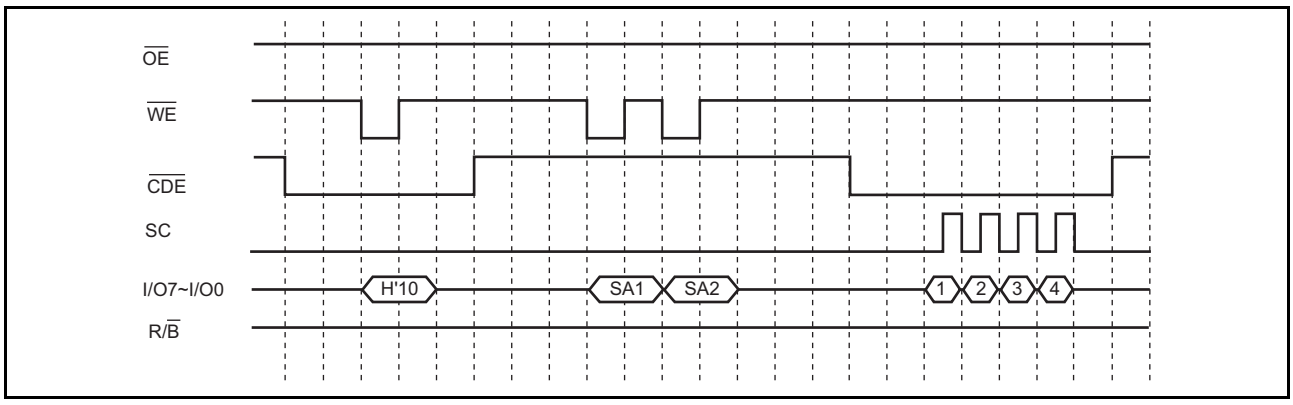


图 24.6　AND 型闪存的写操作时序 （1）

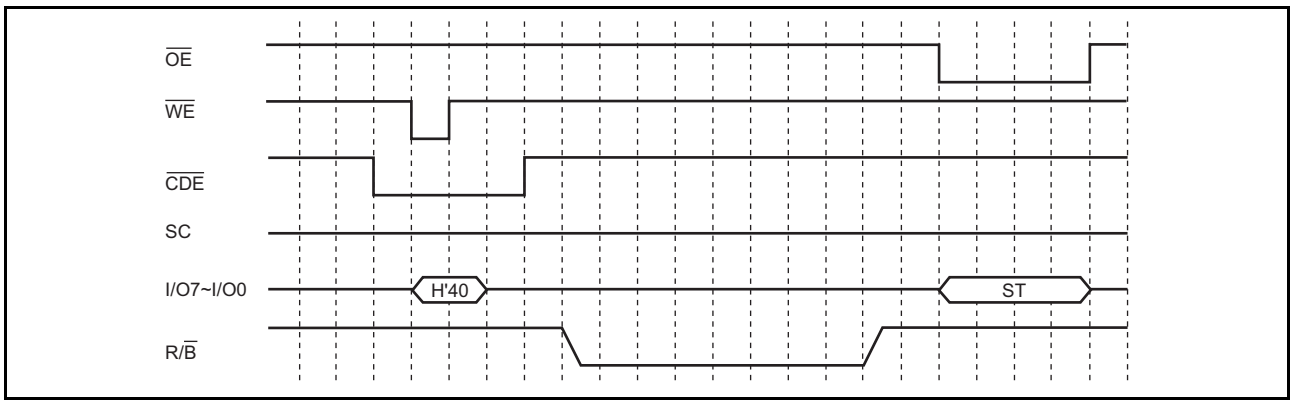


图 24.7　AND 型闪存的写操作时序 （2）

（2） NAND 型闪存的存取

对 NAND 型闪存进行读操作的例子如图 24.8 所示。本例中，指定 H'00 为第 1 命令，指定地址长为 3 字节。在数据计数器指定 8 字节为读字节数。

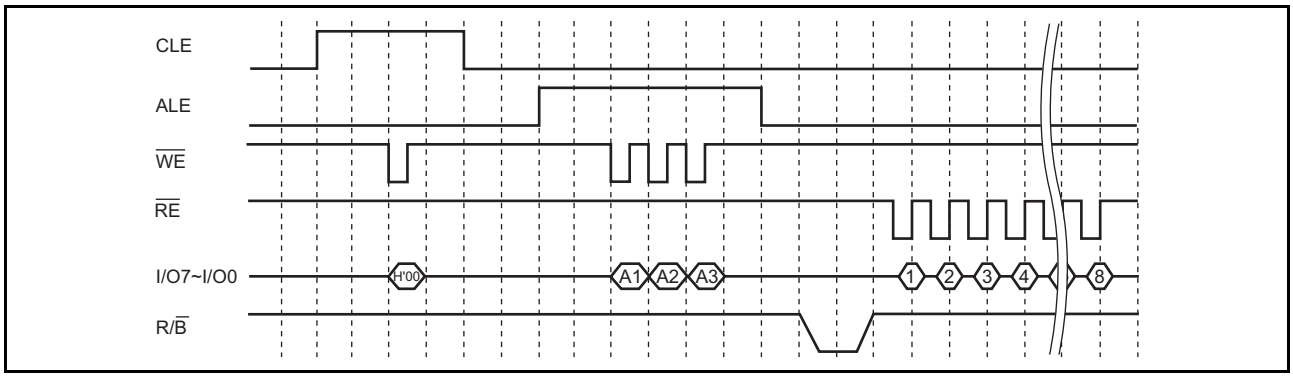


图 24.8　NAND 型闪存的读操作时序

对 NAND 型闪存进行写操作时的波形如图 24.9、图 24.10 所示。

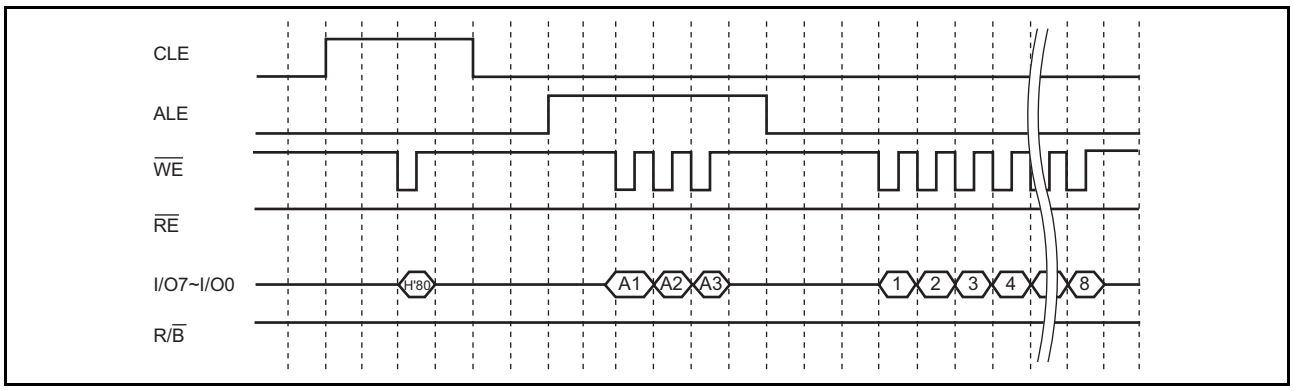


图 24.9　NAND 型闪存的写操作时序 （1）

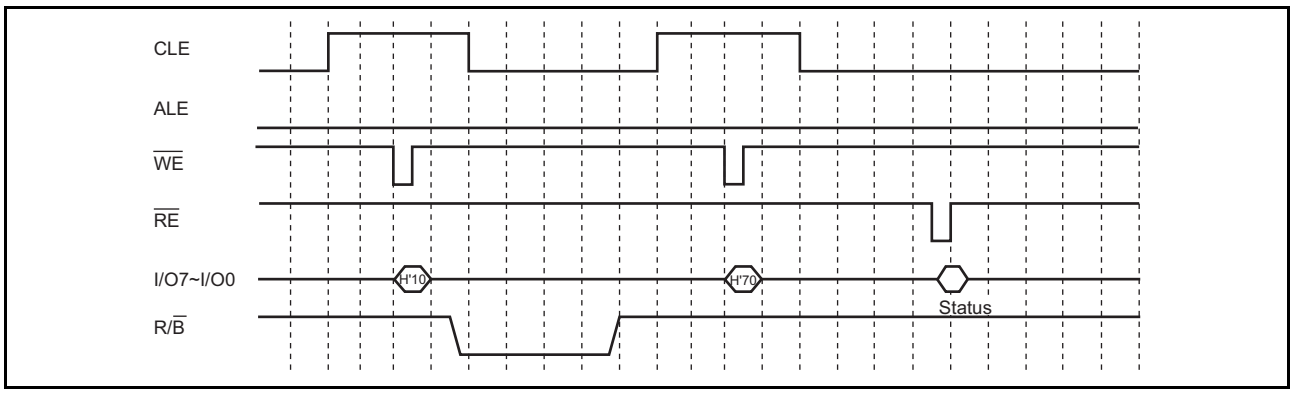


图 24.10　NAND 型闪存的写操作时序 （2）

（3） NAND 型闪存 （2048+64 字节）的存取

对 NAND 型闪存（2048+64 字节）进行读操作的例子如图 24.11 所示。本例中；指定第 1 命令为 H'00，指定第 2 命令为 H'30，指定地址长为 4 字节。在数据计数器指定 4 字节为读字节数。

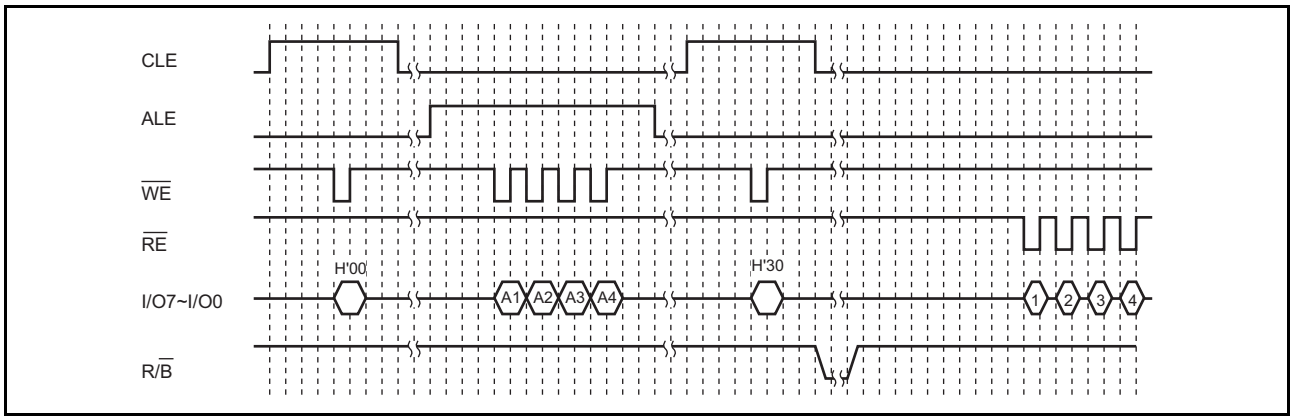


图 24.11　NAND 型闪存的读操作时序

对 NAND 型闪存 （2048+64 字节）进行写操作的波形如图 24.12、图 24.13 所示。

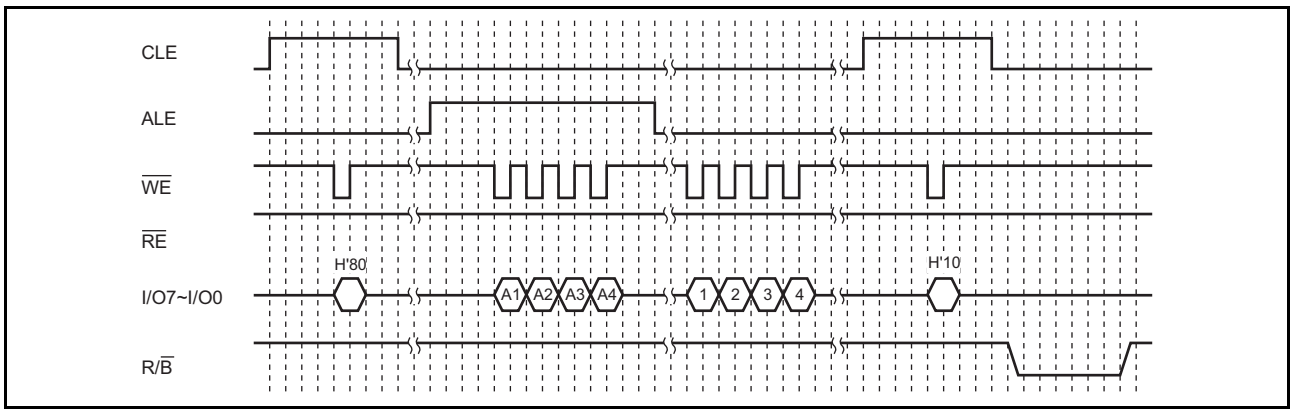


图 24.12　NAND 型闪存的写操作时序 （1）

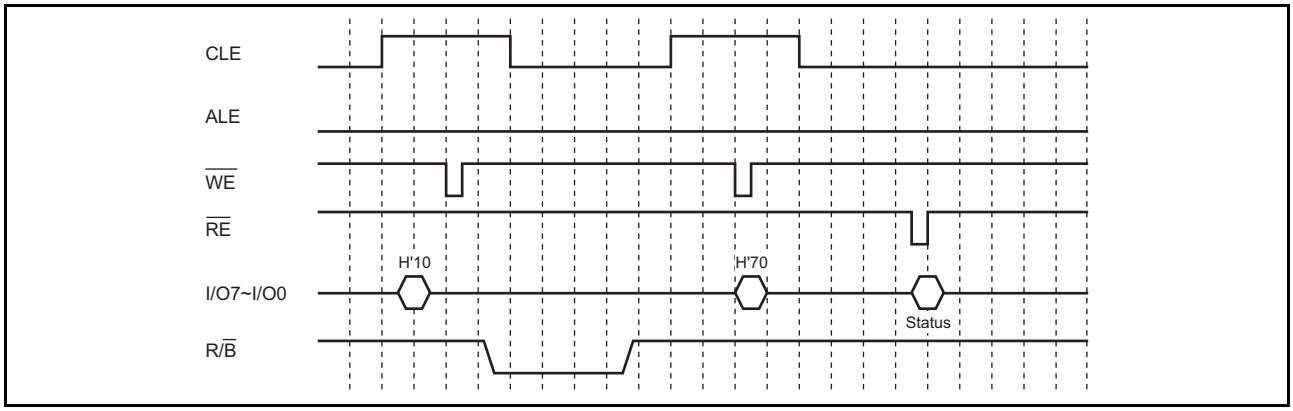


图 24.13　NAND 型闪存的写操作时序 （2）

24.4.5　扇区存取模式

扇区存取模式中，通过指定存取的物理扇区编号，可读写扇区单位。另外，写入时附加 ECC，读出时执行 ECC 错误校验 （检测）处理。

512 字节的数据保存到 FLDTFIFO， 16 字节的管理码保存到 FLECFIFO，因此能设定 FLINTDMACR 的 DREQ1EN 与 DREQ0EN，各自进行 DMA 传送。

闪存内的扇区 （数据 + 管理码）与地址空间上的存储器之间的 DMA 传送关系如图 24.14 所示。

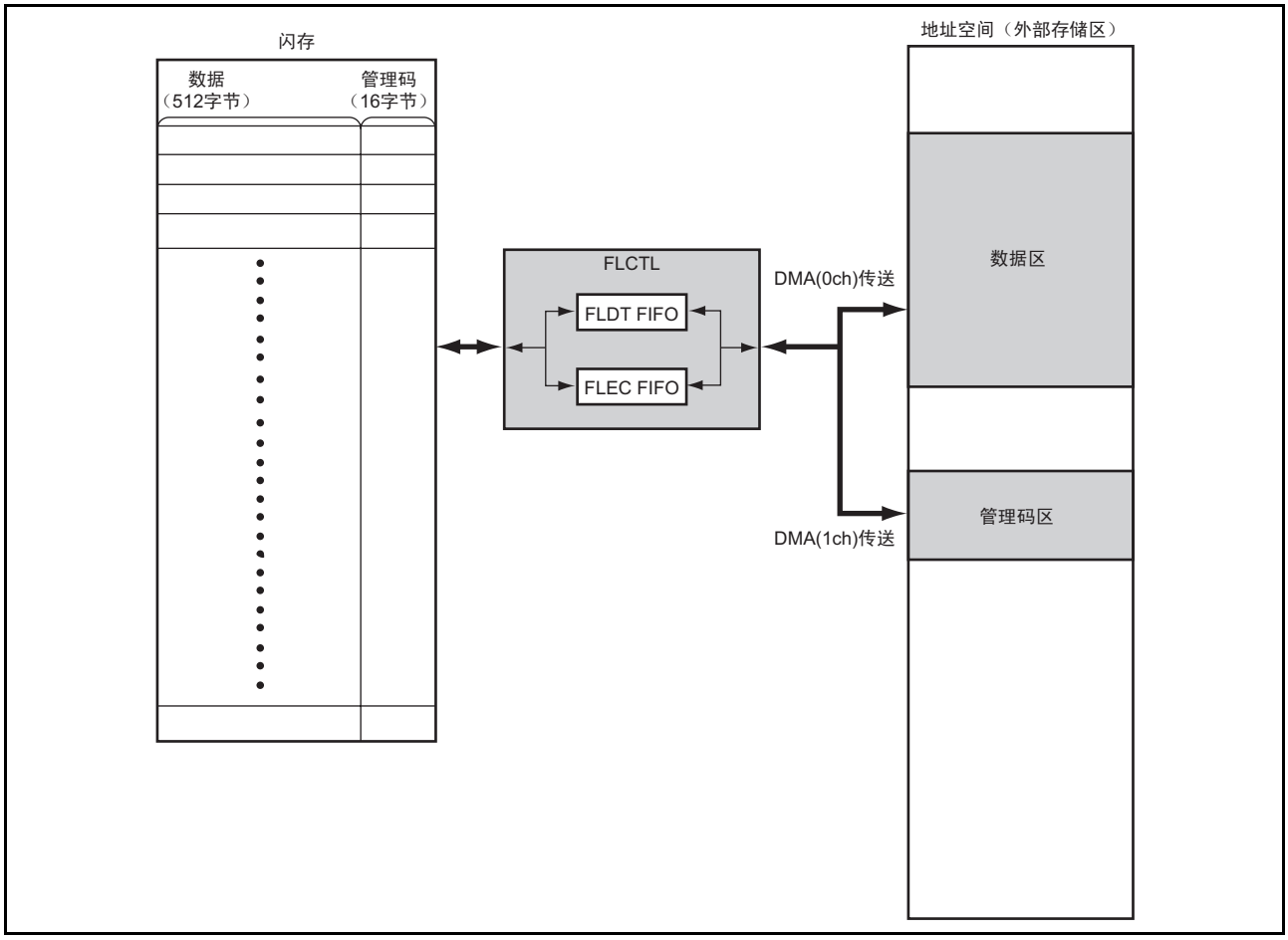


图 24.14　DMA 传送与扇区 （数据，管理码）与存储器与 DMA 传送的关联模式图

（1） 物理扇区

AND/NAND 型闪存的物理扇区地址和闪存的地址的关系如图 24.15 所示。

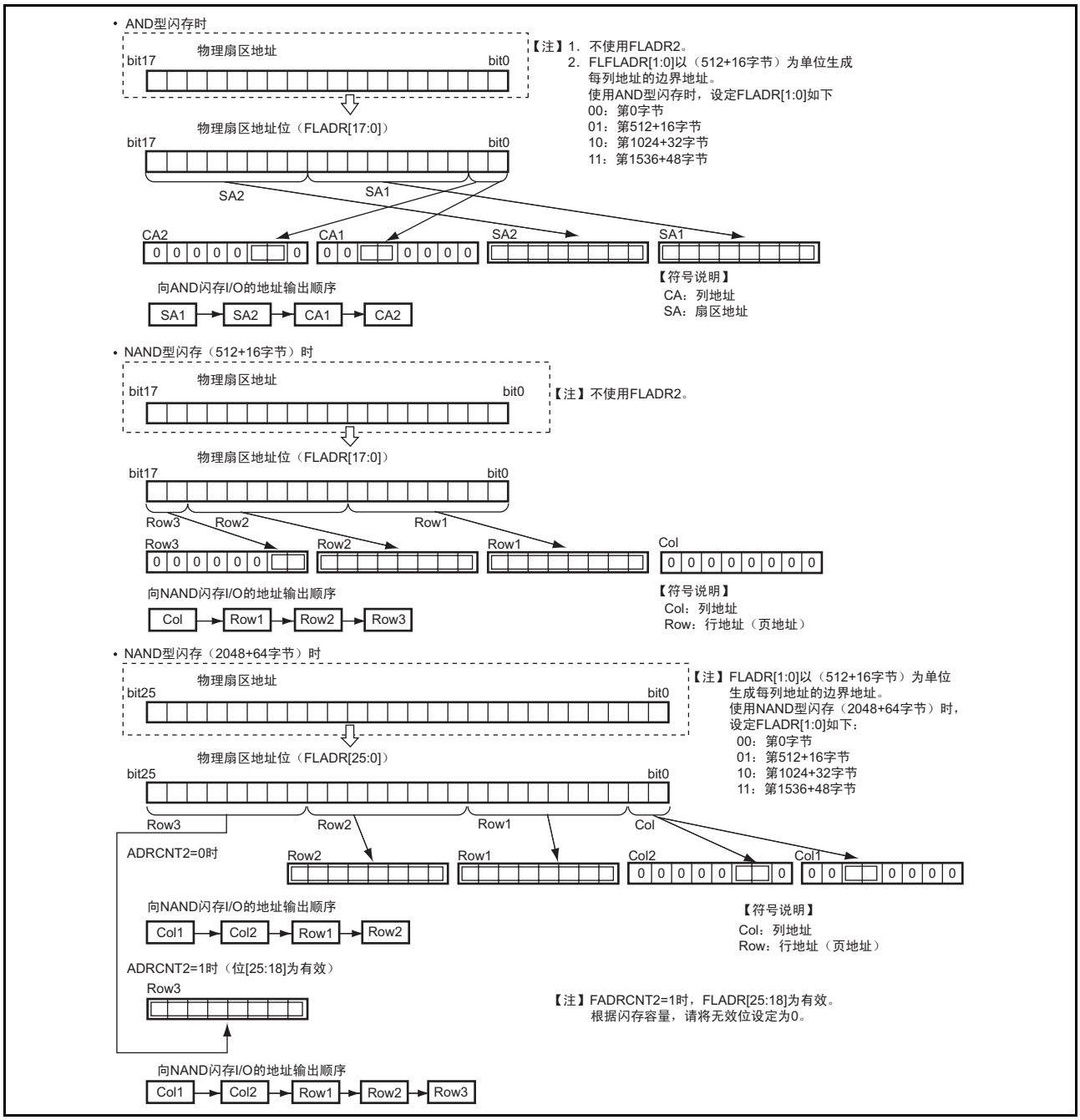


图 24.15　扇区编号与 AND/NAND 型闪存地址的展开例子

（2） 连续扇区存取

通过指定 NAND 型闪存的起始物理扇区与扇区传送次数，可进行连续的物理扇区的读 / 写。当传送 0 ～ 40 的逻辑扇区 （途中存在坏扇区，致使物理扇区不连续）时，物理扇区指定寄存器和扇区传送次数指定寄存器的设定例如图 24.16 所示。

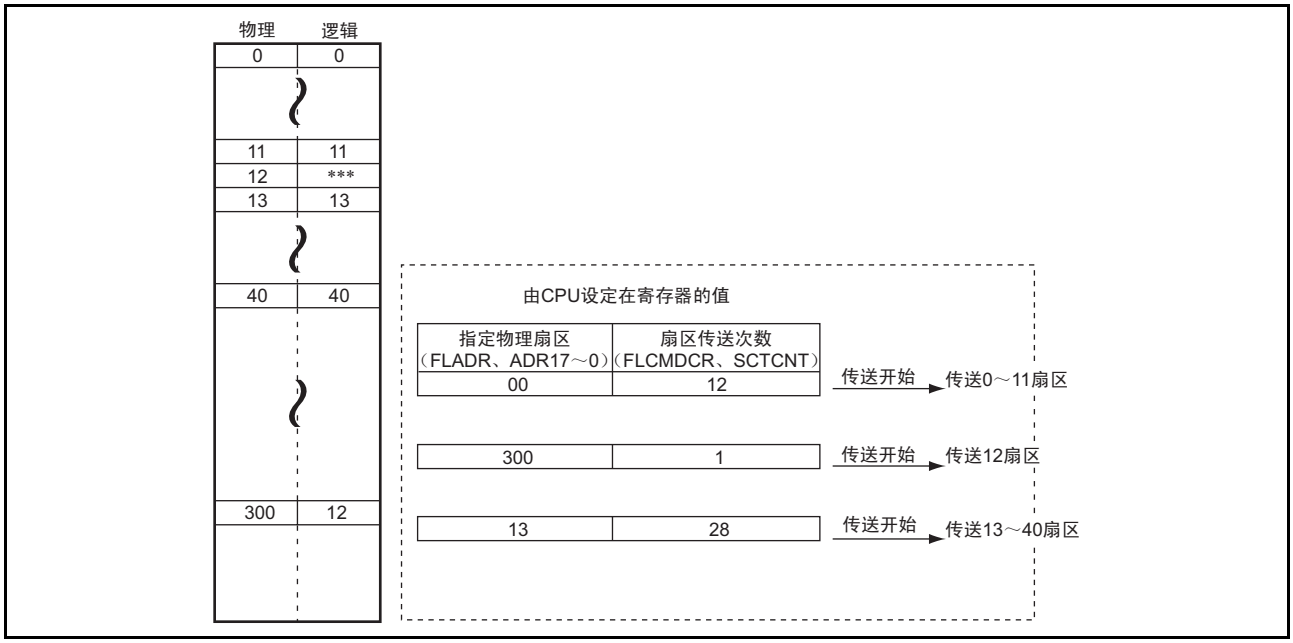


图 24.16　存在坏扇区时的扇区存取例子

24.4.6　ECC 的纠错

在扇区存取模式下进行写入时，本FLCTL生成以及附加ECC码，读出时进行ECC错误校验，但不进行纠错。必需通过软件进行纠错。

24.4.7　状态读

FLCTL 能读出 AND 型闪存与 NAND 型闪存的状态寄存器的值。 AND/NAND 闪存的状态寄存器的值从 I/O7 ～ 0 输入，并保存在 FLBSYCNT 的 STAT[7:0] 位。 FLBSYCNT 的 STAT[7:0] 位可从 CPU 读出。状态寄存器的值保存在 FLBSYCNT 的 STAT[7:0] 位时，如果检测出编程错误和擦除错误， FLINTDMACR 的 STERB 位就被置 1。如果 FLINTDMACR 的 STERINTE 位被允许就对 CPU 产生中断。

（1） AND 型闪存的状态读

可通过断言输出允许信号 $\overline{OE}$ （$\overline{OE}$=0），来实现 AND 型闪存的状态读。如果将 FLCMDCR 的 DOSR 位置 1，并在命令存取模式或者扇区存取模式执行写入， FLCTL 就自动断言 $\overline{OE}$，并进行状态读。 AND 型闪存状态读时，从 I/O7 ～ 0 输入的状态寄存器各位的含义如表 24.3 所示。

表 24.3　AND 型闪存的状态读

I/O	状态 （definition）	说　明
I/O7	就绪 / 忙	0：忙状态 1：就绪状态
I/O6	保留	—
I/O5	擦除校验	0：Pass （擦除） 1：Fail （擦除失败）
I/O4	编程校验	0：Pass （编程） 1：Fail （编程失败）
I/O3 ～ 0	保留	—

（2） NAND 型闪存的状态读

可通过对 NAND 型闪存输入命令 H'70，来实现 NAND 型闪存的状态读。如果将 FLCMDCR 的 DOSR 位置 1，并在命令存取模式或者扇区存取模式进行写入， FLCTL 就自动将 H'70 输入到 NAND 型闪存，并进行状态读。 NAND 型闪存状态读时，从 I/O7 ～ 0 输入的状态寄存器各位的含义如表 24.4 所示。

表 24.4　NAND 型闪存的状态读

I/O	状态 （definition）	说　明
I/O7	写保护	0：不能写入 1：可写入
I/O6	就绪 / 忙	0：忙状态 1：就绪状态
I/O5 ～ 1	保留	—
I/O0	编程 / 擦除	0：Pass （成功） 1：Fail （失败）

24.5　中断处理

FLCTL 有 6 种中断源（状态错误、就绪 / 忙超时错误、ECC 错误、传送结束、FIFO0 传送请求、FIFO1 传送请求）。全部的中断源拥有独立的中断标志，如果通过中断允许位被允许，就对 CPU 产生独立的中断请求。状态错误与就绪 / 忙超时错误、 ECC 错误，对 CPU 使用通用的 FLSTE 中断。

表 24.5　FLCTL 的中断请求

中断源	中断标志	允许位	含　义	优先级
FLSTE 中断	STERB	STERINTE	状态错误	高
	BTOERB	RBERINTE	就绪 / 忙超时错误	
	ECERB	ECERINTE	ECC 错误	
FLTEND 中断	TREND	TEINTE	传送结束	
FLTRQ0 中断	TRREQF0	TRINTE0	FIFO0 传送请求	
FLTRQ1 中断	TRREQF1	TRINTE1	FIFO1 传送请求	低

24.6　DMA 传送的设定

FLCTL 可分别从数据区 FLDTFIFO 与管理码区域 FLECFIFO 请求 DMA 传送。在各存取模式中 DMA 传送的能 / 不能，如表 24.6 所示。

表 24.6　DMA 传送的设定

	扇区存取模式	命令存取模式
FLDTFIFO	能	能
FLECFIFO	能	不能

在使用小端法时，如果使用 16 字节 DMA 传送就会在长字单位发生数据的替换，因此不能使用。

关于 DMAC 的设定请参照 “第 10 章 直接存储器存取控制器 （DMAC）”。

第 25 章　USB2.0 主机 / 功能模块 （USB）

本模块是支持 USB 规格 2.0 的高速传送及全速传送，并具备 USB 主机 / 功能的模块。同时，本模块内置 * USB 收发器，支持 USB 规格所定义的全部传送类型。

内置用于数据传送的 8K 字节缓冲存储器，最多能使用 8 个管道。还能对管道 1 ～ 7 分配进行通信的外围器件以及符合用户系统的任意端点编号。

【注】 * 使用本模块时，需要事先进行内部 USB 收发器的设定。详细内容请参照 “25.5.2 USB 收发器的设定步骤”。

25.1　特点

（1） 内置支持 USB 高速运行的主机控制器和功能控制器

- 内置USB主机控制器和功能控制器
- 可通过寄存器的设定切换USB主机控制器和功能控制器
- 支持高速传送 （480Mbps）和全速传送 （12Mbps）
- 内置高速/全速USB收发器 （主机/功能共用）

（2） 可减少外部器件并且节省安装空间

- 内置D+上拉电阻 （功能部件运行时）
- 内置D+、 D-的下拉电阻 （主机运行时）
- 内置D+、 D-终端电阻 （高速运行时）
- 内置D+、 D-输出电阻 （全速运行时）

（3） 支持 USB 通信的全部数据传送类型

- 控制传送
- 批量传送
- 中断传送 （不支持High Bandwidth）
- 同步传送 （不支持High Bandwidth）

（4） 内部总线接口

- 内置2个DMA接口通道

（5） 管道结构

- 内置8K字节的USB通信用缓冲存储器
- 最多可选择8个管道 （含默认控制管道）
- 可编程的管道结构
- 能给管道1～7分配任意端点编号
- 各管道的可设定传送条件如下：
  管道0：控制传送专用管道、连续传送模式、 256字节固定单缓冲
  管道1、 2：可选择批量传送或者同步传送的管道、连续传送模式、缓冲器大小可编程 （最大为2K字节：能指定双缓冲）
  管道3～5：批量传送专用管道、连续传送模式、缓冲器大小可编程（最大为2K字节：能指定双缓冲）
  管道6、 7：中断传送专用管道、 64字节固定单缓冲

【注】 使用同步 OUT 时，请参照 “25.5.1 使用同步 OUT 传送时的注意事项”。

（6） 选择主机控制器时的特点

- 能与外围设备进行1对1连接的通信
- 自动安排SOF、包的发送时序
- 同步传送、中断传送的传送间隔设定功能

（7） 选择功能控制器时的特点

- 控制传送阶段管理功能
- 器件状态管理功能
- SET_ADDRESS请求的自动响应功能
- NAK响应中断功能 （NRDY）

（8） 其他功能

- 通过复位信号交换自动响应，自动识别高速运行或者全速运行
- 通过处理计数来结束传送的功能
- DMA传送的结束功能
- SOF插值功能
- DMA传送结束时的Zero-Length包附加功能 （DEZPM）
- BRDY中断事件通知时序变更功能 （BFRE）
- 在DnFIFO （n=0、 1）端口读出指定的管道数据后，自动清除缓冲存储器的功能 （DCLRM）
- 通过传输结束产生的响应PID的NAK设定功能 （SHTNAK）

25.2　输入 / 输出引脚

USB 的引脚结构如表 25.1 所示。

不使用本模块时，请按照下面的方法处理。

- 必须外加电源。
- 请将DP、 DM、 VBUS连接到$USBDPV_{SS}$。
- 通过5.6kΩ±1%的电阻，将REFRIN连接到$USBAPV_{SS}$。
- 关于USB_X1、 USB_X2，请参照 “4.3 时钟运行模式”。

表 25.1　USB 的引脚结构

区分	名称	引脚名	输入 / 输出	功能
USB 总线接口	USB D+ 数据	DP	输入 / 输出	为 USB 内部收发器 D+ 输入 / 输出。 请连接到 USB 总线的 D+ 引脚。
	USB D- 数据	DM	输入 / 输出	为 USB 内部收发器 D- 输入 / 输出。 请连接到 USB 总线的 D- 引脚。
VBUS 监控输入	VBUS 输入	VBUS	输入	为 USB 电缆连接监视引脚。 请连接到 USB 总线的 V bus。能检测出 V bus 的连接 / 断开。不连接 USB 总线的 V bus 时，固定为 5V。选择主机控制器功能时，也提供 5V。 【注】不能给连接的器件提供 V bus。
基准电阻	参考输入	REFRIN	输入	基准电阻连接用引脚。 必须通过 5.6kΩ ± 1%电阻连接到 $USBAPV_{SS}$。
时钟	USB 用晶体谐振器 / 外部时钟	USB_X1	输入	连接 USB 用晶体谐振器。另外， USB_X1 引脚也能输入外部时钟。
		USB_X2	输出	
电源	收发器部 模拟引脚电源	$USBAPV_{CC}$	输入	为引脚用电源。
	收发器部 模拟引脚接地	$USBAPV_{SS}$	输入	为引脚用接地。
	收发器部 数字引脚电源	$USBDPV_{CC}$	输入	为引脚用电源。
	收发器部 数字引脚接地	$USBDPV_{SS}$	输入	为引脚用接地。
	收发器部 模拟核电源	$USBAV_{CC}$	输入	为核用电源。
	收发器部 模拟核接地	$USBAV_{SS}$	输入	为核用接地。
	收发器部 数字核电源	$USBDV_{CC}$	输入	为核用电源。
	收发器部 数字核接地	$USBDV_{SS}$	输入	为核用接地。

25.3　寄存器说明

USB 的寄存器结构如表 25.2 所示。

表 25.2　寄存器结构

寄存器名	简称	R/W	初始值	地址	存取长度
系统结构控制寄存器	SYSCFG	R/W	H'0000	H'FFFC1C00	16
系统结构状态寄存器	SYSSTS	R	H'040x	H'FFFC1C02	16
器件状态控制寄存器	DVSTCTR	R/W	H'0000	H'FFFC1C04	16
测试模式寄存器	TESTMODE	R/W	H'0000	H'FFFC1C06	16
CPU-FIFO 总线结构寄存器	CFBCFG	R/W	H'000F	H'FFFC1C0A	16
DMA0-FIFO 总线结构寄存器	D0FBCFG	R/W	H'000F	H'FFFC1C0C	16
DMA1-FIFO 总线结构寄存器	D1FBCFG	R/W	H'000F	H'FFFC1C0E	16
CFIFO 端口寄存器	CFIFO	R/W	H'00000000	H'FFFC1C10	8、16、32
D0FIFO 端口寄存器	D0FIFO	R/W	H'00000000	H'FFFC1C14	8、16、32
D1FIFO 端口寄存器	D1FIFO	R/W	H'00000000	H'FFFC1C18	8、16、32
CFIFO 端口选择寄存器	CFIFOSEL	R/W	H'0000	H'FFFC1C1E	16
CFIFO 端口控制寄存器	CFIFOCTR	R/W	H'0000	H'FFFC1C20	16
CFIFO 端口 SIE 寄存器	CFIFOSIE	R/W	H'0000	H'FFFC1C22	16
D0FIFO 端口选择寄存器	D0FIFOSEL	R/W	H'0000	H'FFFC1C24	16
D0FIFO 端口控制寄存器	D0FIFOCTR	R/W	H'0000	H'FFFC1C26	16
D0 处理计数器寄存器	D0FIFOTRN	R/W	H'0000	H'FFFC1C28	16
D1FIFO 端口选择寄存器	D1FIFOSEL	R/W	H'0000	H'FFFC1C2A	16
D1FIFO 端口控制寄存器	D1FIFOCTR	R/W	H'0000	H'FFFC1C2C	16
D1 处理计数器寄存器	D1FIFOTRN	R/W	H'0000	H'FFFC1C2E	16
中断允许寄存器 0	INTENB0	R/W	H'0000	H'FFFC1C30	16
中断允许寄存器 1	INTENB1	R/W	H'0000	H'FFFC1C32	16
BRDY 中断允许寄存器	BRDYENB	R/W	H'0000	H'FFFC1C36	16
NRDY 中断允许寄存器	NRDYENB	R/W	H'0000	H'FFFC1C38	16
BEMP 中断允许寄存器	BEMPENB	R/W	H'0000	H'FFFC1C3A	16
中断状态寄存器 0	INTSTS0	R/W	H'00x0	H'FFFC1C40	16
中断状态寄存器 1	INTSTS1	R/W	H'0000	H'FFFC1C42	16
BRDY 中断状态寄存器	BRDYSTS	R/W	H'0000	H'FFFC1C46	16
NRDY 中断状态寄存器	NRDYSTS	R/W	H'0000	H'FFFC1C48	16
BEMP 中断状态寄存器	BEMPSTS	R/W	H'0000	H'FFFC1C4A	16
帧号寄存器	FRMNUM	R/W	H'0000	H'FFFC1C4C	16
μ 帧号寄存器	UFRMNUM	R/W	H'0000	H'FFFC1C4E	16

寄存器名	简称	R/W	初始值	地址	存取长度
USB 地址寄存器	USBADDR	R	H'0000	H'FFFC1C50	16
USB 请求类型寄存器	USBREQ	R	H'0000	H'FFFC1C54	16
USB 请求值寄存器	USBVAL	R	H'0000	H'FFFC1C56	16
USB 请求变址寄存器	USBINDX	R	H'0000	H'FFFC1C58	16
USB 请求长度寄存器	USBLENG	R	H'0000	H'FFFC1C5A	16
DCP 结构寄存器	DCPCFG	R/W	H'0000	H'FFFC1C5C	16
DCPMax 包大小寄存器	DCPMAXP	R/W	H'0040	H'FFFC1C5E	16
DCP 控制寄存器	DCPCTR	R/W	H'0040	H'FFFC1C60	16
管道窗口选择寄存器	PIPESEL	R/W	H'0000	H'FFFC1C64	16
管道结构寄存器	PIPECFG	R/W	H'0000	H'FFFC1C66	16
管道缓冲器指定寄存器	PIPEBUF	R/W	H'0000	H'FFFC1C68	16
管道 Max 包大小寄存器	PIPEMAXP	R/W	H'0xxx	H'FFFC1C6A	16
管道周期控制寄存器	PIPEPERI	R/W	H'0000	H'FFFC1C6C	16
管道 1 控制寄存器	PIPE1CTR	R/W	H'0000	H'FFFC1C70	16
管道 2 控制寄存器	PIPE2CTR	R/W	H'0000	H'FFFC1C72	16
管道 3 控制寄存器	PIPE3CTR	R/W	H'0000	H'FFFC1C74	16
管道 4 控制寄存器	PIPE4CTR	R/W	H'0000	H'FFFC1C76	16
管道 5 控制寄存器	PIPE5CTR	R/W	H'0000	H'FFFC1C78	16
管道 6 控制寄存器	PIPE6CTR	R/W	H'0000	H'FFFC1C7A	16
管道 7 控制寄存器	PIPE7CTR	R/W	H'0000	H'FFFC1C7C	16
USB AC 特性转换寄存器	USBACSWR	R/W	H'00000000	H'FFFC1C84	32

25.3.1　系统结构控制寄存器 （SYSCFG）

SYSCFG 是进行是否允许高速运行、选择主机控制器功能或者功能控制器功能、控制 DP 及 DM 引脚、控制全速接收器以及本模块软件复位的寄存器。

通过上电复位初始化本寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	HSE	DCFM	DMRPD	DPRPU	—	FSRPC	—	USBE
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	R/W	R	R/W

位	位名	初始值	R/W	说明
15 ～ 8	—	均为 0	R	保留位 读写的值总是 0。
7	HSE	0	R/W	高速运行允许 0：禁止高速运行 （只允许全速运行） 1：允许高速运行 （本模块检测）
6	DCFM	0	R/W	控制器功能选择 选择主机控制器功能或者功能控制器功能。 0：选择功能控制器功能 1：选择主机控制器功能
5	DMRPD	0	R/W	D- 线电阻控制
4	DPRPU	0	R/W	D+ 线电阻控制 设定 D-， D+ 线的电阻。设定本位前，必须设定 HSE 位以及 DCFM 位。 00：D- 为开路， D+ 为开路 01：D- 为开路， D+ 为上拉 10：D- 为下拉， D+ 为下拉 11：D- 为下拉， D+ 为上拉
3	—	0	R	保留位 读写的值总是 0。
2	FSRPC	0	R/W	全速接收器运行允许 允许全速接收器的运行。 0：硬件控制全速接收器的运行 1：通过软件允许全速接收器的运行
1	—	0	R	保留位 读写的值总是 0。
0	USBE	0	R/W	允许 USB 块运行 通过本位能进行本模块的软件复位。设定 USBE=0 时，将软件复位初始化对象寄存器复位为初始设定值。另外，在 USBE=0 设定中，不进行向软件复位初始化对象寄存器以及位的写入。软件复位后，请设定 USBE=1，允许本模块的运行。 0：禁止 USB 块运行 （软件复位） 1：允许 USB 块运行

25.3.2　系统结构状态寄存器 （SYSSTS）

SYSSTS 监控 USB 数据总线的线状态 （D+ 以及 D- 线）。

通过上电复位、软件复位或 USB 复位初始化本寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	SOFEN	—	—	—	LNST[1:0]	
初始值：	0	0	0	0	0	1	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 ～ 11	—	均为 0	R	保留位 读写的值总是 0。
10	—	1	R	保留位 读取值不定。写无效。
9 ～ 6	—	均为 0	R	保留位 读写的值总是 0。
5	SOFEN	0	R	SOF 发送允许 表示主机模式运行时，软件对 DVSTCTR 寄存器的 UACT 位进行写操作后，本模块的内部电路为可发送 SOF 或者为 SOF 发送停止状态。 0：不对 USB 端口发送 SOF 1：对 USB 端口发送 SOF
4 ～ 2	—	均为 0	R	保留位 读写的值总是 0。
1、0	LNST[1:0]	—	R	USB 数据线路状态 USB 数据总线的线状态表如表 25.3 所示。通过设定本位，来监控 USB 数据总线的线状态 （D+ 线及 D- 线）。 用全速接收器确认线状态。通过提供 USBCLK，本模块自动进行全速接收器的控制，但是如果设定 SYSCFG 寄存器的 FSRPC 位，即使不提供 USBCLK，也能通过软件允许全速接收器的控制。上电复位后，在提供 USBCLK 前确认 D+ 以及 D- 线的状态时，请将 FSRPC 位置 1。另外，一旦提供 USBCLK 后，就不需要软件设定。

【注】 *　取决于 D+ 以及 D- 线的状态。

表 25.3　USB 数据总线线状态表

LNST1	LNST0	全速运行时	高速运行时	Chirp 运行时
0	0	SE0	Squelch	Squelch
0	1	J-State	not Squelch	Chirp J
1	0	K-State	Invalid	Chirp K
1	1	SE1	Invalid	Invalid

【符号说明】

Chirp：允许高速运行的状态 （SYSCFG.HSE=1）下，执行复位信号交换协议 （RHSP）时

Squelch：SE0 或者空闲状态

not Squelch：高速 J-State 或者高速 K-State

Chirp J：Chirp J-State

Chirp K：Chirp K-State

Invalid：无效

25.3.3　器件状态控制寄存器 （DVSTCTR）

DVSTCTR 控制及确认 USB 数据总线的状态。

通过上电复位初始化本寄存器。另外在软件复位后， WKUP 位以外的位被初始化，但是 WKUP 位不定。USB 总线复位后， WKUP 位被初始化， RESUME 位不定。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UACKEY0	—	—	UACKEY1	—	—	—	WKUP	RWUPE	USBRST	RESUME	UACT	—	—	RHST[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	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R

位	位名	初始值	R/W	说明
15	UACKEY0	0	R/W	USBAC 键 0 为了对测试寄存器的 HOSTPCC 位进行写入，需要设定本位。详细请参照“25.5.2 USB 收发器的设定步骤”。
14、13	—	均为 0	R	保留位 读写的值总是 0。
12	UACKEY1	0	R/W	USBAC 键 1 为了对测试寄存器的 HOSTPCC 位进行写入，需要设定本位。详细请参照“25.5.2 USB 收发器的设定步骤”。
11 ～ 9	—	均为 0	R	保留位 读写的值总是 0。
8	WKUP	0	R/W	唤醒输出 本位用于对 USB 总线上的远程唤醒信号输出进行控制。本模块管理远程唤醒信号的输出时间。本位置 1 时，在本模块输出 10ms 的 K-State 后，将本位清 0。 在 USB 规格中，远程唤醒信号发送前，需要保持最短 5ms 的 USB 总线空闲状态。因此，本模块即使在检测出挂起状态后立即对本位写 1，也在等待 2ms 后才输出 K-State。 0：输出无信号 1：输出远程唤醒信号 【注】除器件状态为挂起 （INTSTS0.DVSQ=1xx），并且 USB 主机允许远程唤醒的情况以外不能对本位写 1。同时，如果将本位置 1，即使是挂起也不能停止 USBCLK。
7	RWUPE	0	R/W	唤醒检测允许 将本位置 1 时，如果检测出远程唤醒信号就对下行端口输出恢复信号。此时，本模块将 RESUME 位置 1。 0：禁止下行端口唤醒 1：允许下行端口唤醒 【注】将本位置 1 时，即使是挂起也不能停止 USBCLK。
6	USBRST	0	R/W	总线复位输出 通过将本位置 1，输出 USB 总线复位信号。 USB 总线复位信号输出时间，需要通过软件进行时间管理。经过 USB 总线复位时间后，将本位设定为 0。 0：USB 总线复位信号输出停止 1：USB 总线复位信号输出

位	位名	初始值	R/W	说明
5	RESUME	0	R/W	恢复输出 通过将本位置 1，对 USB 总线输出恢复信号。 0：停止恢复信号输出 1：恢复信号输出
4	UACT	0	R/W	USB 总线允许 控制对 USB 总线发送 SOF 或者 μSOF 包。本模块管理 SOF 包的发送间隔。对本位写 0 时，在下一个 SOF 发送后，成为总线空闲状态。 0：禁止下行端口 （禁止 SOF/μSOF 发送） 1：允许下行端口 （允许 SOF/μSOF 发送）
3、2	—	均为 0	R	保留位 读写的值总是 0。
1、0	RHST[1:0]	00	R	复位信号交换 通过参照本位，能确认与连接的主机控制器之间的传输速度 （通信位速率）。将高速运行设定为禁止状态 （SYSCFG.HSE=0）时，本模块不执行复位信号交换协议而变成全速运行。将高速运行设定为允许状态时，本模块执行（执行中时 RHST=01）复位信号交换协议，并且把执行结果反映给本位 （高速运行时 RHST=11 或者全速运行时 RHST=10）。 选择主机控制器功能时，USB 总线复位解除后，在下面的二个时序将结果反映到本位。其一，全速模式时， USB 总线通过 USB 总线复位输出从 SE0-State 变为 J-State 时。其二，高速模式时，通过复位信号交换将终端电阻切换为高速模式时。但是，必须注意：不管是哪个时序，连接的外围器件为全速对应器件时，在反映之前需要等待时间。 00：传输速度不定 01：复位信号交换处理中 10：确定全速运行 11：确定高速运行 【注】 USB 总线复位结束后 （设定 USBRST=0 后），经过充分的等待时间后 RHST 仍不确定时， USB 总线复位中 USB 线可能被断开。此时，请通过 LNST 位确认 USB 总线的状态。

【注】 选择功能控制器功能时， RWUPE、 USBRST、 RESUME、 UACT 位的值必须全部清 0。
选择主机控制器功能时， WKUP 位的值必须清 0。

25.3.4　测试模式寄存器 （TESTMODE）

TESTMODE 控制高速运行时的 USB 测试信号输出。
通过上电复位或软件复位初始化本寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HOSTPCC	—	—	—	—	—	—	—	—	—	—	—	UTST[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	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	HOSTPCC	0	R/W	断开检波器功率转换 进行 USB 收发器的设定。 * 要设定本位，需要设定器件控制寄存器的 UACKEY0、 UACKEY1 位。
14 ～ 4	—	均为 0	R	保留位 读写的值总是 0。
3 ～ 0	UTST[3:0]	0000	R/W	测试模式 本模块的测试模式运行表如表 25.4 所示。本位控制高速运行时的 USB 测试信号输出。 本位只在高速运行时有效。必须在确认 DVSTCTR.RHST=11 之后才能使用。另外，本位的 bit3 （UTST3）因控制器功能选择位 （DCFM）而异。必须根据 DCFM 位设定。同时，通过设定本位实施测试后，需要通过上电复位返回。

【注】 *　详细内容请参照 “25.5.2 USB 收发器的设定步骤”。

表 25.4　测试模式运行表

测试模式	UTST 位设定	
	选择功能控制器功能时	选择主机控制器功能时
正常运行	0000	0000
Test_J	0001	1001
Test_K	0010	1010
Test_SE0_NAK	0011	1011
Test_Packet	0100	1100
Reserved	0101 ～ 0111	1101 ～ 1111

25.3.5　FIFO 端口结构寄存器 （CFBCFG、 D0FBCFG、 D1FBCFG）

进行 FIFO 端口的存取控制。FIFO 端口有 CPU-FIFO、DMA0-FIFO 以及 DMA1-FIFO 共 3 个端口，相应的结构寄存器控制各自端口的存取。

通过上电复位初始化本寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TENDE	FEND	—	—	—	—	FWAIT[3:0]			
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/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15 ～ 10	—	均为 0	R	保留位 读写的值总是 0。
9	TENDE	0	R/W	允许 DMA 传送结束采样 控制 DMA 传送结束时，是否接收从直接存储器存取控制器 （DMAC）发送的 DMA 传送结束信号。 0：不采样 DMA 传送结束信号 1：采样 DMA 传送结束信号
8	FEND	0	R/W	FIFO 端口端法 设定存取 FIFO 端口时的字节端法。端法运行表如表 25.5 ～表 25.7 所示。 本 LSI 为大端法运行。发送 / 接收不同端法的数据时，请设定本位。
7 ～ 4	—	均为 0	R	保留位 读写的值总是 0。
3 ～ 0	FWAIT[3:0]	1111	R/W	指定 FIFO 端口存取等待 指定相应 FIFO 端口的存取等待数。 FIFO 端口的最小存取周期为 2 个周期。 0000：0 等待 （2 个存取周期） ： 0010：2 等待 （4 个存取周期） ： 0100：4 等待 （6 个存取周期） ： 1111：15 等待 （17 个存取周期）

【注】 TEND 位只在 D0FBCFG 和 D1FBCFG 中存在。

表 25.5　端法运行 （32 位宽存取时）

FEND	bit31 ～ bit24	bit23 ～ bit16	bit15 ～ bit8	bit7 ～ bit0
0	N+0 地址	N+1 地址	N+2 地址	N+3 地址
1	N+3 地址	N+2 地址	N+1 地址	N+0 地址

表 25.6　端法运行 （16 位宽存取时）

FEND	bit31 ～ bit24	bit23 ～ bit16	bit15 ～ bit8	bit7 ～ bit0
0	偶数地址	奇数地址	写：无效 读：禁止 *	
1	写：无效 读：禁止 *		奇数地址	偶数地址

【注】 *　禁止以字为单位读取无效寄存器。

表 25.7　端法运行 （8 位宽存取时）

FEND	bit31 ～ bit24	bit23 ～ bit16	bit15 ～ bit8	bit7 ～ bit0
0	写：有效 读：有效	写：无效 读：无效 *		
1	写：无效 读：禁止 *			写：有效 读：有效

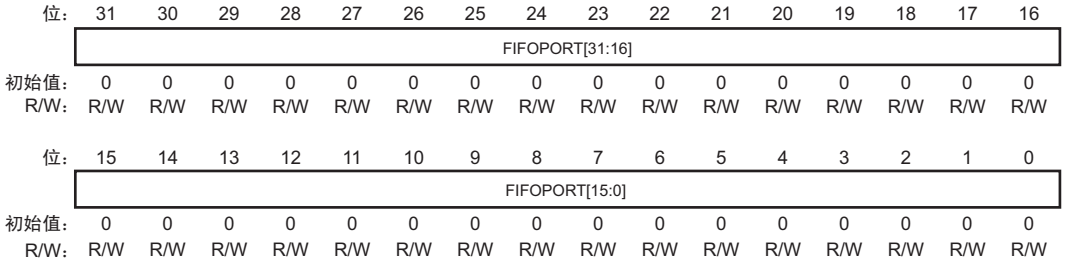
【注】 *　禁止以字节为单位读取无效寄存器。

25.3.6　FIFO 端口寄存器 （CFIFO、 D0FIFO、 D1FIFO）

该寄存器是向 FIFO 缓冲存储器读 / 写数据的端口寄存器。

FIFO 端口有 CFIFO、D0FIFO 和 D1FIFO 共 3 个端口。各 FIFO 端口与由向缓冲存储器读 / 写数据的端口寄存器、选择分配给 FIFO 端口的管道的寄存器、控制寄存器以及端口功能专门寄存器 （CFIFO 端口专用的 SIE 寄存器和 DnFIFO 端口专用的处理计数器寄存器）构成的寄存器群有关。

通过上电复位或软件复位初始化本寄存器。



位	位名	初始值	R/W	说明
31 ～ 0	FIFOPORT[31:0]	均为 0	R/W	FIFO 端口 从缓冲存储器读取接收数据或者向缓冲存储器写入发送数据。

【注】 1. 只有 DCP 为 CFIFO 端口时，才能存取缓冲存储器。
只有D0FIFO端口和D1FIFO端口才能通过DMA传送存取缓冲存储器。

2. 通过 CPU 存取 DnFIFO 端口时，也要按照 DnFIFO 端口的功能及限制 （利用处理计数器等）进行。
3. 使用 FIFO 端口固有的功能时，不能使用选择管道 （利用处理计数器等）。
4. 构成 FIFO 端口的寄存器群，不影响其他 FIFO 端口。
5. 不能将同一管道分配给不同的 FIFO 端口。
6. 缓冲存储器时，存取权可能在 CPU 或者 SIE。缓冲存储器的存取权在 SIE 时，不能从 CPU 正确存取。
7. 不能改变在 FIFO 端口选择的管道的管道结构。

25.3.7　FIFO 端口选择寄存器 （CFIFOSEL、 D0FIFOSEL、 D1FIFOSEL）

该寄存器选择分配给 FIFO 端口的管道，并控制对相应端口的存取。

不能向 CFIFOSEL、 D0FIFOSEL 和 D1FIFOSEL 寄存器的 CURPIPE 位指定同一管道。同时， D0FIFOSEL 和 D1FIFOSEL 寄存器的 CURPIPE 位设定为 B'000 时，为无管道指定。

另外，不能在 DMA 传送允许状态下进行管道编号的变更。

通过上电复位或软件复位初始化本寄存器。

（1） CFIFOSEL

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RCNT	REW	—	—	MBW[1:0]		—	—	—	—	ISEL	—	—	CURPIPE[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/W	R	R	R	R	R/W	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15	RCNT	0	R/W	读计数模式 0：读出全部接收数据后，清除 DTLN 位 1：读出接收数据后， DTLN 位递减计数
14	REW	0	R/W*	缓冲器指针回绕 0：缓冲器指针不回绕 1：缓冲器指针回绕
13、 12	—	均为 0	R	保留位 读写的值总是 0。
11、 10	MBW[1:0]	00	R/W	FIFO 端口存取位宽 00：8 位宽 01：16 位宽 10：32 位宽 11：禁止设定 【注】一旦开始缓冲存储器的读出处理，在全部数据读出结束前，不能进行 FIFO 端口存取位宽的变更。同时，执行对缓冲内存的写处理时，不能进行从 8 位宽切换到 16 位宽 /32 位宽，或者从 16 位宽切换到 32 位宽的位宽。
9 ～ 6	—	均为 0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说　　明
5	ISEL	0	R/W	选择 DCP 时的 FIFO 端口存取方向 0：缓冲存储器读出选择 1：缓冲存储器写入选择 本位只在用 CURPIPE 位选择 DCP 时有效。 请按以下步骤设定本位： • 对 CURPIPE 位的 DCP 设定 （CURPIPE=000）和对 ISEL 位写入同时进行 • 对 CURPIPE 位的 DCP 进行设定（CURPIPE=000）后，等待 200ns，再对 ISEL 位进行设定 详细请参照 “25.4.4 缓冲存储器”。
4、3	—	均为 0	R	保留位 读写的值总是 0。
2 ～ 0	CURPIPE [2:0]	000	R/W	指定 FIFO 端口存取管道 000：DCP 001：管道 1 010：管道 2 011：管道 3 100：管道 4 101：管道 5 110：管道 6 111：管道 7

【注】 *　只有读 0 和写 1 有效。

（2） D0FIFOSEL、 D1FIFOSEL

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RCNT	REW	DCLRM	DREQE	MBW[1:0]		TRENB	TRCLR	DEZPM	—	—	—	—	CURPIPE[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/W	R/W	R/W	R/W	R/W	R/W*	R/W	R	R	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15	RCNT	0	R/W	读计数模式 0：读出全部接收数据后，清除 DTLN 位 1：读出接收数据后， DTLN 位递减计数
14	REW	0	R/W*	缓冲器指针回绕 0：缓冲器指针不回绕 1：缓冲器指针回绕
13	DCLRM	0	R/W	读出指定管道的数据后，自动进入缓冲存储器清除模式 本位只在设定 CURPIPE 位指定的管道为接收方向（从缓冲存储器读出）时有效。 0：禁止自动缓冲器清除模式 1：允许自动缓冲器清除模式
12	DREQE	0	R/W	允许 DMA 传送请求 0：禁止 DMA 传送请求 1：允许 DMA 传送请求
11、 10	MBW[1:0]	00	R/W	FIFO 端口存取位宽 00：8 位宽 01：16 位宽 10：32 位宽 11：禁止设定 【注】一旦开始缓冲存储器的读出处理，在全部数据读出结束前，不能进行 FIFO 端口存取位宽的变更。同时，执行对缓冲存储器的写入处理时，不能进行从 8 位宽切换到 16 位宽 /32 位宽，或者从 16 位宽切换到 32 位宽的位宽。
9	TRENB	0	R/W	处理计数器允许 本位只在设定 CURPIPE 位指定的管道为接收方向（从缓冲存储器读出）时有效。 0：处理计数器功能无效 1：处理计数器功能有效
8	TRCLR	0	R/W	清除处理计数器 本位只在设定 CURPIPE 位指定的管道为接收方向（从缓冲存储器读出）时有效。 0：无效 1：清除当前计数器

位	位名	初始值	R/W	说明
7	DEZPM	0	R/W	Zero-Length 包附加模式 本位只在设定 CURPIPE 位指定的管道为发送方向（向缓冲存储器写入）时有效。 0：无付加 1：有付加
6 ～ 3	—	均为 0	R	保留位 读写的值总是 0。
2 ～ 0	CURPIPE [2:0]	000	R/W	指定 FIFO 端口存取管道 000：无指定 001：管道 1 010：管道 2 011：管道 3 100：管道 4 101：管道 5 110：管道 6 111：管道 7

【注】 *　只有读 0 和写 1 有效。

25.3.8　FIFO 端口控制寄存器 （CFIFOCTR、 D0FIFOCTR、 D1FIFOCTR）

该寄存器是设定缓冲存储器的写入是否结束、是否在 CPU 清除缓冲器以及可否存取 FIFO 端口的寄存器。本寄存器中，对应各 FIFO 端口的有 CFIFOCTR、 D0FIFOCTR 以及 D1FIFOCTR。

通过上电复位及软件复位初始化本寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BVAL	BCLR	FRDY	—	DTLN[11:0]											
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	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15	BVAL	0	R/W*1	缓冲存储器有效标志 只在数据包方向为发送方向时（向缓冲存储器写入时），向本位写 1 才有效。为接收方向时，请设定 BVAL=0。 0：无效 1：写入结束
14	BCLR	0	R/W*1	CPU 缓冲器清除 *2 请在通过管道结构设定为管道无效 （PID=NAK）的情况下，通过本位进行缓冲器清除 0：无效 1：CPU 侧缓冲存储器清除
13	FRDY	0	R	FIFO 端口就绪 确认本位，需要在选择管道后至少 450ns 个存取周期。 0：不可存取 FIFO 端口 1：可存取 FIFO 端口
12	—	0	R	保留位 读写的值总是 0。
11 ～ 0	DTLN[11:0]	H'000	R	接收数据长 *3 能确认接收数据长。

【注】 *1　只有读 0 和写 1 有效。

*2　选择 DCP 以外的管道时， BCLR 位对 CPU 侧的缓冲存储器有效。必须在确认 FRDY=1 后，设定 BCLR=1。选择 DCP 为管道时，也进行 SIE 侧的缓冲存储器清除。此时，不需要确认 FRDY=1。

*3　DTLN 位仅对 CPU 侧的缓冲存储器有效。必须在确认 FRDY=1 后，再参照 DTLN 位。

25.3.9　CFIFO 端口 SIE 寄存器 （CFIFOSIE）

该寄存器控制 CFIFO 端口的 SIE 功能，切换 CPU 侧与 SIE 侧的存取权、清除 SIE 侧缓冲存储器以及确认 SIE 侧缓冲器忙 / 空闲。选择 DCP 时，本寄存器不起作用。

通过上电复位及软件复位初始化本寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TGL	SCLR	SBUSY	—	—	—	—	—	—	—	—	—	—	—	—	—
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	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15	TGL	0	R/W*	切换存取权 本位的功能是将 SIE 侧的缓冲存储器设置到 CPU 侧。设定 PID=NAK，并由 SBUSY 位确认 SIE 没有存取缓冲器 （SBUSY=0）后，写入 TGL 位 （交替操作）。同时，本位只对设定为接收方向 （从缓冲存储器读取）的管道有效。 0：无效 1：切换存取权
14	SCLR	0	R/W*	SIE 缓冲器清除 本位的功能是清除 SIE 侧的缓冲存储器。设定 PID=NAK，并由 SBUSY 位确认 SIE 没有存取缓冲器 （SBUSY=0）后，清除缓冲器。另外，本位只对设定为发送方向 （向缓冲存储器写入）的管道有效。 0：无效 1：清除 SIE 侧缓冲存储器
13	SBUSY	0	R	SIE 缓冲器忙 0：SIE 不进行存取的状态 1：SIE 进行存取的状态
12 ～ 0	—	均为 0	R	保留位 读写的值总是 0。

【注】 *　只有读 0 和写 1 有效。

25.3.10　处理计数器寄存器 （D0FIFOTRN、 D1FIFOTRN）

该寄存器是设定 DMA 传送的处理次数及读出处理次数的寄存器。

通过上电复位及软件复位初始化本寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRNCNT[15:0]															
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 ～ 0	TRNCNT[15:0]	H'0000	R/W	处理计数器 本位只在从缓冲存储器读出数据时有效。 只有在 DnFIFOSEL 寄存器的 TRENB 位为 1 时，才能读出计数时的处理次数， TRENB=0 时能读出设定的处理次数。 W：设定 DMA 传送的处理次数 R：读出处理次数

25.3.11　中断允许寄存器 0 （INTENB0）

该寄存器进行各中断屏蔽指定、器件状态转移中断源屏蔽指定（URST 位、SADR 位、SCFG 位、SUSP 位）和控制传送阶段中断源屏蔽的指定 （WDST 位、 RDST 位、 CMPL 位、 SERR 位）。

通过上电复位及软件复位初始化本寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	URST	SADR	SCFG	SUSP	WDST	RDST	CMPL	SERR
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	VBSE	0	R/W	允许 VBUS 中断 0：禁止中断输出 1：允许中断输出
14	RSME	0	R/W	允许恢复中断 0：禁止中断输出 1：允许中断输出
13	SOFE	0	R/W	允许帧号更新中断 0：禁止中断输出 1：允许中断输出
12	DVSE	0	R/W	允许器件状态转移中断 0：禁止中断输出 1：允许中断输出
11	CTRE	0	R/W	允许控制传送阶段转移中断 0：禁止中断输出 1：允许中断输出

位	位名	初始值	R/W	说明
10	BEMPE	0	R/W	允许缓冲器空中断 0：禁止中断输出 1：允许中断输出
9	NRDYE	0	R/W	允许缓冲器未就绪响应中断 0：禁止中断输出 1：允许中断输出
8	BRDYE	0	R/W	允许缓冲器就绪中断 0：禁止中断输出 1：允许中断输出
7	URST	0	R/W	允许默认状态转移通知 0：默认状态转移时，禁止 DVST 中断 1：默认状态转移时，允许 DVST 中断
6	SADR	0	R/W	允许地址状态转移通知 0：地址状态转移时，禁止 DVST 中断 1：地址状态转移时，允许 DVST 中断
5	SCFG	0	R/W	允许结构状态转移通知 0：结构状态转移时，禁止 DVST 中断 1：结构状态转移时，允许 DVST 中断
4	SUSP	0	R/W	允许挂起状态转移通知 0：挂起状态转移时，禁止 DVST 中断 1：挂起状态转移时，允许 DVST 中断
3	WDST	0	R/W	允许控制写阶段转移通知 0：控制写阶段转移时，禁止 CTRT 中断 1：控制写阶段转移时，允许 CTRT 中断
2	RDST	0	R/W	允许控制读阶段转移通知 0：控制读阶段转移时，禁止 CTRT 中断 1：控制读阶段转移时，允许 CTRT 中断
1	CMPL	0	R/W	允许控制传送结束通知 0：检测出控制传送结束时，禁止 CTRT 中断 1：检测出控制传送结束时，允许 CTRT 中断
0	SERR	0	R/W	允许控制传送顺序错误通知 0：检测出控制传送顺序错误时，禁止 CTRT 中断 1：检测出控制传送顺序错误时，允许 CTRT 中断

【注】 清除各中断状态后，允许 / 禁止相应的中断时，请至少空出 80ns 的间隔。

25.3.12　中断允许寄存器 1 （INTENB1）

该寄存器指定各中断屏蔽，并控制各管道的 BRDY 中断状态清除时序。

通过上电复位初始化本寄存器。同时，通过软件复位，初始化 BRDYM 位以外的位。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	BCHGE	—	DTCHE	—	—	—	—	—	—	SIGNE	SACKE	—	BRDYM	—	—
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	R/W	R	R	R	R	R	R	R/W	R/W	R	R/W	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	BCHGE	0	R/W	允许 USB 总线变化中断 0：禁止中断输出 1：允许中断输出
13	—	0	R	保留位 读写的值总是 0。
12	DTCHE	0	R/W	全速运行时允许断开检测中断 通过本位的断开检测只在选择主机控制器功能和全速运行时有效。高速运行时，请通过软件 （检测出功能控制器无响应等）检测断开。 0：禁止中断输出 1：允许中断输出
11 ～ 6	—	均为 0	R	保留位 读写的值总是 0。
5	SIGNE	0	R/W	允许设置处理错误中断 0：禁止中断输出 1：允许中断输出
4	SACKE	0	R/W	允许设置处理正常响应中断 0：禁止中断输出 1：允许中断输出
3	—	0	R	保留位 读写的值总是 0。
2	BRDYM	0	R/W	各管道的 BRDY 中断状态清除时序控制 0：通过软件来清除状态 1: 本模块通过FIFO缓冲器的读出或者对FIFO缓冲器的写入操作来清除状态
1、 0	—	均为 0	R	保留位 读写的值总是 0。

25.3.13　BRDY 中断允许寄存器 （BRDYENB）

该寄存器允许各管道的 BRDY 中断。

通过上电复位及软件复位初始化本寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PIPE7 BRDYE	PIPE6 BRDYE	PIPE5 BRDYE	PIPE4 BRDYE	PIPE3 BRDYE	PIPE2 BRDYE	PIPE1 BRDYE	PIPE0 BRDYE
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
15 ～ 8	—	均为 0	R	保留位 读写的值总是 0。
7	PIPE7BRDYE	0	R/W	允许管道 7 的 BRDY 中断 0：禁止中断输出 1：允许中断输出
6	PIPE6BRDYE	0	R/W	允许管道 6 的 BRDY 中断 0：禁止中断输出 1：允许中断输出
5	PIPE5BRDYE	0	R/W	允许管道 5 的 BRDY 中断 0：禁止中断输出 1：允许中断输出
4	PIPE4BRDYE	0	R/W	允许管道 4 的 BRDY 中断 0：禁止中断输出 1：允许中断输出
3	PIPE3BRDYE	0	R/W	允许管道 3 的 BRDY 中断 0：禁止中断输出 1：允许中断输出
2	PIPE2BRDYE	0	R/W	允许管道 2 的 BRDY 中断 0：禁止中断输出 1：允许中断输出
1	PIPE1BRDYE	0	R/W	允许管道 1 的 BRDY 中断 0：禁止中断输出 1：允许中断输出
0	PIPE0BRDYE	0	R/W	允许管道 0 的 BRDY 中断 0：禁止中断输出 1：允许中断输出

【注】 清除各中断状态后，允许 / 禁止对应的中断时，请至少空出 80ns 的间隔。

25.3.14　NRDY 中断允许寄存器 （NRDYENB）

该寄存器允许各管道的 NRDY 中断。

通过上电复位及软件复位初始化本寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PIPE7 NRDYE	PIPE6 NRDYE	PIPE5 NRDYE	PIPE4 NRDYE	PIPE3 NRDYE	PIPE2 NRDYE	PIPE1 NRDYE	PIPE0 NRDYE
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
15 ～ 8	—	均为 0	R	保留位 读写的值总是 0。
7	PIPE7NRDYE	0	R/W	允许管道 7 的 NRDY 中断 0：禁止中断输出 1：允许中断输出
6	PIPE6NRDYE	0	R/W	允许管道 6 的 NRDY 中断 0：禁止中断输出 1：允许中断输出
5	PIPE5NRDYE	0	R/W	允许管道 5 的 NRDY 中断 0：禁止中断输出 1：允许中断输出
4	PIPE4NRDYE	0	R/W	允许管道 4 的 NRDY 中断 0：禁止中断输出 1：允许中断输出
3	PIPE3NRDYE	0	R/W	允许管道 3 的 NRDY 中断 0：禁止中断输出 1：允许中断输出
2	PIPE2NRDYE	0	R/W	允许管道 2 的 NRDY 中断 0：禁止中断输出 1：允许中断输出
1	PIPE1NRDYE	0	R/W	允许管道 1 的 NRDY 中断 0：禁止中断输出 1：允许中断输出
0	PIPE0NRDYE	0	R/W	允许管道 0 的 NRDY 中断 0：禁止中断输出 1：允许中断输出

【注】 清除各中断状态后，允许 / 禁止对应的中断时，请至少空出 80ns 的间隔。

25.3.15　BEMP 中断允许寄存器 （BEMPENB）

该寄存器允许各管道的 BEMP 中断。

通过上电复位及软件复位初始化本寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PIPE7 BEMPE	PIPE6 BEMPE	PIPE5 BEMPE	PIPE4 BEMPE	PIPE3 BEMPE	PIPE2 BEMPE	PIPE1 BEMPE	PIPE0 BEMPE
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
15 ～ 8	—	均为 0	R	保留位 读写的值总是 0。
7	PIPE7BEMPE	0	R/W	允许管道 7 的 BEMP 中断 0：禁止中断输出 1：允许中断输出
6	PIPE6BEMPE	0	R/W	允许管道 6 的 BEMP 中断 0：禁止中断输出 1：允许中断输出
5	PIPE5BEMPE	0	R/W	允许管道 5 的 BEMP 中断 0：禁止中断输出 1：允许中断输出
4	PIPE4BEMPE	0	R/W	允许管道 4 的 BEMP 中断 0：禁止中断输出 1：允许中断输出
3	PIPE3BEMPE	0	R/W	允许管道 3 的 BEMP 中断 0：禁止中断输出 1：允许中断输出
2	PIPE2BEMPE	0	R/W	允许管道 2 的 BEMP 中断 0：禁止中断输出 1：允许中断输出
1	PIPE1BEMPE	0	R/W	允许管道 1 的 BEMP 中断 0：禁止中断输出 1：允许中断输出
0	PIPE0BEMPE	0	R/W	允许管道 0 的 BEMP 中断 0：禁止中断输出 1：允许中断输出

【注】 清除各中断状态后，允许 / 禁止对应的中断时，请至少空出 80ns 的间隔

25.3.16　中断状态寄存器 0 （INTSTS0）

该寄存器是确认各中断状态的寄存器。

通过上电复位及软件复位初始化本寄存器。同时，通过 USB 总线复位初始化 DVSQ[2:0] 位。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ[2:0]			VALID	CTSQ[2:0]		
初始值：	0	0	0	0	0	0	0	0	*3	*4	*4	*4	0	0	0	0
R/W：	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R	R	R	R	R	R	R	R/W*1	R	R	R

位	位名	初始值	R/W	说明
15	VBINT	0	R/W*1	VBUS 中断状态 *2 0：未产生 VBUS 中断 1：产生 VBUS 中断
14	RESM	0	R/W*1	恢复中断状态 *2 0：未产生恢复中断 1：产生恢复中断
13	SOFR	0	R/W*1	帧号更新中断状态 *2 0：未产生 SOF 中断 1：产生 SOF 中断
12	DVST	0	R/W*1	器件状态转移中断状态 *2 0：未产生器件状态转移中断 1：产生器件状态转移中断
11	CTRT	0	R/W*1	控制传送阶段转移中断状态 *2 0：未产生控制传送阶段转移中断 1：产生控制传送阶段转移中断
10	BEMP	0	R	缓冲器空中断状态 在 BEMPSTS 寄存器的全部位被清除时，清除本位。 0：未产生 BEMP 中断 1：产生 BEMP 中断
9	NRDY	0	R	缓冲器未就绪中断状态 在 NRDYSTS 寄存器的全部位被清除时，清除本位。 0：未产生 NRDY 中断 1：产生 NRDY 中断
8	BRDY	0	R	缓冲器就绪中断状态 在 BRDYSTS 寄存器的全部位被清除时，清除本位。 0：未产生 BRDY 中断 1：产生 BRDY 中断
7	VBSTS	*3	R	VBUS 输入状态 本位监控 BUS 引脚的状态。但需要通过控制用程序去除震颤。 0：VBUS 引脚为低电平 1：VBUS 引脚为高电平

位	位名	初始值	R/W	说明
6 ～ 4	DVSQ[2:0]	*4	R	器件状态 000：上电状态 001：默认状态 010：地址状态 011：结构状态 1xx：暂停状态
3	VALID	0	R/W*1	设置包接收 0：未检测出 1：设置包接收
2 ～ 0	CTSQ[2:0]	000	R	控制传送阶段 000：空闲或者设置阶段 001：控制读数据阶段 010：控制读状态阶段 011：控制写数据阶段 100：控制写状态阶段 101：控制写 （NoData）状态阶段 110：控制传送顺序错误 111：禁止设定

【注】 *1　只有写 0 有效。

*2　VBINT 位、 RESM 位、 SOFR 位、 DVST 位以及 CTRT 位中产生多个源时，如果不是同时而是连续清除各位，就需要至少 140ns+ 总线时钟 3 个周期的存取周期。

*3　VBUS 引脚为高电平时为 1，为低电平时为 0。

*4　上电复位、软件复位时为 B'000， USB 总线复位时为 B'001。

25.3.17　中断状态寄存器 1 （INTSTS1）

该寄存器为确认各中断状态的寄存器。SOFR、BEMP、NRDY、BRDY 位为 INTSTS0 寄存器的镜像位。读取时，能读出与 INTSTS0 寄存器的同一位相同的值。另外，写入时，写入的值反映到 INTSTS0 和 INTSTS1 寄存器。

另外，选择外围控制器功能时利用 INTSTS0 寄存器，选择主机控制器功能时利用 INTSTS1 寄存器，由此可通过参照任何一方的寄存器得知中断的产生。

通过上电复位及软件复位初始化本寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	BCHG	SOFR	DTCH	—	BEMP	NRDY	BRDY	—	—	SIGN	SACK	—	—	—	—
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/W*	R/W*	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	BCHG	0	R/W*	USB 总线变化中断状态 0：未产生 BCHG 中断 1：产生 BCHG 中断
13	SOFR	0	R/W*	帧号更新中断状态 0：未产生 SOF 中断 1：产生 SOF 中断
12	DTCH	0	R/W*	全速运行时，断开检测中断状态 通过本位的断开检测只在选择主机控制器功能和全速运行时有效。高速运行时，请通过软件 （检测出外围设备无响应等）检测断开 0：未产生 DTCH 中断 1：产生 DTCH 中断
11	—	0	R	保留位 读写的值总是 0。
10	BEMP	0	R	缓冲器空中断状态 0：未产生 BEMP 中断 1：产生 BEMP 中断
9	NRDY	0	R	缓冲器未就绪中断状态 0：未产生 NRDY 中断 1：产生 NRDY 中断
8	BRDY	0	R	缓冲器就绪中断状态 0：未产生 BRDY 中断 1：产生 BRDY 中断
7、6	—	均为 0	R	保留位 读写的值总是 0。
5	SIGN	0	R/W*	设置处理错误中断状态 0：未产生 SIGN 中断 1：产生 SIGN 中断

位	位名	初始值	R/W	说明
4	SACK	0	R/W*	设置处理正常响应中断状态 0：未产生 SACK 中断 1：产生 SACK 中断
3 ～ 0	—	均为 0	R	保留位 读写的值总是 0。

【注】 * 只有写 0 有效。

25.3.18　BRDY 中断状态寄存器 （BRDYSTS）

该寄存器是确认各管道的 BRDY 中断状态的寄存器。

通过上电复位及软件复位初始化本寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PIPE7 BRDY	PIPE6 BRDY	PIPE5 BRDY	PIPE4 BRDY	PIPE3 BRDY	PIPE2 BRDY	PIPE1 BRDY	PIPE0 BRDY
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*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

位	位名	初始值	R/W	说明
15 ～ 8	—	均为 0	R	保留位 读写的值总是 0。
7	PIPE7BRDY	0	R/W*1	管道 7 的 BRDY 中断状态 *2 0：未产生中断 1：产生中断
6	PIPE6BRDY	0	R/W*1	管道 6 的 BRDY 中断状态 *2 0：未产生中断 1：产生中断
5	PIPE5BRDY	0	R/W*1	管道 5 的 BRDY 中断状态 *2 0：未产生中断 1：产生中断
4	PIPE4BRDY	0	R/W*1	管道 4 的 BRDY 中断状态 *2 0：未产生中断 1：产生中断
3	PIPE3BRDY	0	R/W*1	管道 3 的 BRDY 中断状态 *2 0：未产生中断 1：产生中断
2	PIPE2BRDY	0	R/W*1	管道 2 的 BRDY 中断状态 *2 0：未产生中断 1：产生中断
1	PIPE1BRDY	0	R/W*1	管道 1 的 BRDY 中断状态 *2 0：未产生中断 1：产生中断
0	PIPE0BRDY	0	R/W*1	管道 0 的 BRDY 中断状态 *2 0：未产生中断 1：产生中断

【注】　*1　只有写 0 有效。

*2　产生多个管道的源时，如果不是同时而是连续清除各位，就需要至少 140ns+ 总线时钟 3 个周期的存取周期。

25.3.19　NRDY 中断状态寄存器 （NRDYSTS）

该寄存器是确认各管道的 NRDY 中断状态的寄存器。

通过上电复位及软件复位初始化本寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PIPE7 NRDY	PIPE6 NRDY	PIPE5 NRDY	PIPE4 NRDY	PIPE3 NRDY	PIPE2 NRDY	PIPE1 NRDY	PIPE0 NRDY
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*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

位	位名	初始值	R/W	说明
15 ～ 8	—	均为 0	R	保留位 读写的值总是 0。
7	PIPE7NRDY	0	R/W*1	管道 7 的 NRDY 中断状态 *2 0：未产生中断 1：产生中断
6	PIPE6NRDY	0	R/W*1	管道 6 的 NRDY 中断状态 *2 0：未产生中断 1：产生中断
5	PIPE5NRDY	0	R/W*1	管道 5 的 NRDY 中断状态 *2 0：未产生中断 1：产生中断
4	PIPE4NRDY	0	R/W*1	管道 4 的 NRDY 中断状态 *2 0：未产生中断 1：产生中断
3	PIPE3NRDY	0	R/W*1	管道 3 的 NRDY 中断状态 *2 0：未产生中断 1：产生中断
2	PIPE2NRDY	0	R/W*1	管道 2 的 NRDY 中断状态 *2 0：未产生中断 1：产生中断
1	PIPE1NRDY	0	R/W*1	管道 1 的 NRDY 中断状态 *2 0：未产生中断 1：产生中断
0	PIPE0NRDY	0	R/W*1	管道 0 的 NRDY 中断状态 *2 0：未产生中断 1：产生中断

【注】　*1　只有写 0 有效。

*2　产生多个管道的源时，如果不是同时而是连续清除各位，就需要至少 140ns+ 总线时钟 3 个周期的存取周期。

25.3.20　BEMP 中断状态寄存器 （BEMPSTS）

该寄存器是确认各管道的 BEMP 中断状态的寄存器。

通过上电复位及软件复位初始化本寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PIPE7 BEMP	PIPE6 BEMP	PIPE5 BEMP	PIPE4 BEMP	PIPE3 BEMP	PIPE2 BEMP	PIPE1 BEMP	PIPE0 BEMP
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*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

位	位名	初始值	R/W	说明
15 ～ 8	—	均为 0	R	保留位 读写的值总是 0。
7	PIPE7BEMP	0	R/W*1	管道 7 的 BEMP 中断状态 *2 0：未产生中断 1：产生中断
6	PIPE6BEMP	0	R/W*1	管道 6 的 BEMP 中断状态 *2 0：未产生中断 1：产生中断
5	PIPE5BEMP	0	R/W*1	管道 5 的 BEMP 中断状态 *2 0：未产生中断 1：产生中断
4	PIPE4BEMP	0	R/W*1	管道 4 的 BEMP 中断状态 *2 0：未产生中断 1：产生中断
3	PIPE3BEMP	0	R/W*1	管道 3 的 BEMP 中断状态 *2 0：未产生中断 1：产生中断
2	PIPE2BEMP	0	R/W*1	管道 2 的 BEMP 中断状态 *2 0：未产生中断 1：产生中断
1	PIPE1BEMP	0	R/W*1	管道 1 的 BEMP 中断状态 *2 0：未产生中断 1：产生中断
0	PIPE0BEMP	0	R/W*1	管道 0 的 BEMP 中断状态 *2 0：未产生中断 1：产生中断

【注】　*1　只有写 0 有效。

*2　产生多个管道的源时，如果不是同时而是连续清除各位，就需要至少 140ns+ 总线时钟 3 个周期的存取周期。

25.3.21　帧号寄存器 （FRMNUM）

该寄存器进行同步错误通知的源辨别、 SOFR 中断运行模式的选择和帧号的显示。
通过上电复位及软件复位初始化本寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OVRN	CRCE	—	—	SOFRM	FRNM[10:0]										
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	R	R/W	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15	OVRN	0	R/W*1	上溢 / 下溢 *2 0：无错误 1：发生错误 表示由管道 （同步传送中）的 NRDY 中断引起的错误通知源为数据缓冲器错误。 详细内容请参照表 25.8 及表 25.9。
14	CRCE	0	R/W*1	接收数据错误 *2 0：无错误 1：发生错误 表示由管道（同步传送中）的 NRDY 中断引起的错误通知源为包错误。 详细内容请参照表 25.8 及表 25.9。
13、 12	—	均为 0	R	保留位 读写的值总是 0。
11	SOFRM	0	R/W	帧号更新中断输出模式 • 选择功能控制器功能时 0：在 SOF 接收且定时器插值时中断断言 1：SOF 破损或者丢失时中断断言 • 选择主机控制器功能时 0：SOF 发送时断言 1：禁止设定 在检测出 UFRMNUM.UFRNM=000 以外的 μSOF 包时，不产生帧号更新中断。
10 ～ 0	FRNM[10:0]	H'000	R	帧号 能确认帧号。 选择功能控制器功能时，本模块在 SOF 包接收时序更新帧号。由于包破损等原因模块不能检测出 SOF 包时，在接收新的 SOF 包前，保持 FRNM 的值。 依据 SOF 插值定时器的 FRNM 位不进行更新。

【注】 *1　只有写 0 有效。

*2　产生 OVRN 和 CRCE 的源时，如果不是同时而是连续清除各位，就需要至少 140ns+ 总线时钟 3 个周期的存取周期。

表 25.8　同步 OUT 传送中产生 NRDY 中断时的错误信息

位状态	发生时序	发生条件	检测出错误	运行
OVRN=1	接收数据包	缓冲存储器读出结束前接收到新的数据包	接收数据缓冲器的溢出	废弃接收数据
CRCE=1	接收数据包	检测出 CRC 错误或者位填充错误	接收包错误	废弃接收数据

表 25.9　同步 IN 传送中产生 NRDY 中断时的错误信息

位状态	发生时序	发生条件	检测出错误	运行
OVRN=1	IN-Token 接收	缓冲存储器写入结束前 接收了 IN-Token	发送数据缓冲器的下溢	发送 Zero-Length 包
CRCE=1	不产生			

25.3.22　μ 帧号寄存器 （UFRMNUM）

该寄存器表示 μ 帧号。

通过上电复位及软件复位初始化本寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	UFRNM[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	R	R	R

位	位名	初始值	R/W	说明
15 ～ 3	—	均为 0	R	保留位 读写的值总是 0。
2 ～ 0	UFRNM[2:0]	000	R	μ 帧 能确认 μ 帧号。 通过接收 uSOF 包， UFRNM 位递增。 全速运行时，本位总读出 B'000。

25.3.23　USB 地址寄存器 （USBADDR）

该寄存器表示 USB 地址。

本寄存器，只在选择功能控制器功能时有效。选择主机控制器功能时，必须使用 PIPEMAXP 寄存器的 DEVSEL 位设定外围地址。

通过上电复位、软件复位及 USB 总线复位初始化本寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	USBADDR[6:0]						
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 ～ 7	—	均为 0	R	保留位 读写的值总是 0。
6 ～ 0	USBADDR[6:0]	H'00	R	USB 地址 表示 USB 地址。

25.3.24　USB 请求类型寄存器 （USBREQ）

该寄存器保存控制传送的设置请求。

选择功能控制器功能时，保存接收的 bRequest 及 bmRequestType 的值。选择主机控制器功能时，设定发送的 bRequest 及 bmRequestType 的值。

通过上电复位、软件复位及 USB 总线复位初始化本寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BREQUEST[7:0]								BMREQUESTTYPE[7:0]							
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 ～ 8	BREQUEST[7:0]	H'00	R/W*	请求 保存 USB 请求 bRequest 的值。
7 ～ 0	BMREQUEST TYPE[7:0]	H'00	R/W*	请求类型 保存 USB 请求 bmRequestType 的值。

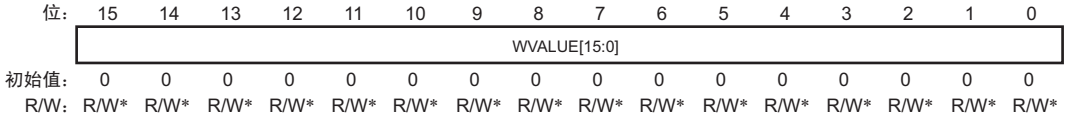
【注】 *　选择功能控制器功能时，只能读。另一方面，选择主机控制器功能时，能读 / 写。

25.3.25　USB 请求价值寄存器 （USBVAL）

该寄存器保存控制传送的设置请求。

选择外围控制器功能时，保存接收的 wValue 的值。选择主机控制器功能时，设定发送的 wValue 的值。

通过上电复位、软件复位及 USB 总线复位初始化本寄存器。



位	位名	初始值	R/W	说明
15 ～ 0	WVALUE[15:0]	H'0000	R/W*	值 保存 USB 请求 wValue 的值。

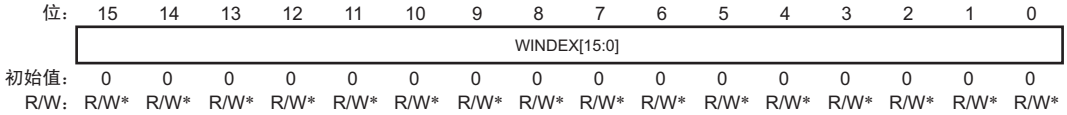
【注】 *　选择功能控制器功能时，只能读。另一方面，选择主机控制器功能时，能读 / 写。

25.3.26　USB 请求变址寄存器 （USBINDX）

该寄存器保存控制传送的设置请求。

选择功能控制器功能时，保存接收的 wIndex 的值。选择主机控制器功能时，设定发送的 wIndex 的值。

通过上电复位、软件复位及 USB 总线复位初始化本寄存器。



位	位名	初始值	R/W	说明
15 ～ 0	WINDEX[15:0]	H'0000	R/W*	索引 保存 USB 请求 wIndex 的值。

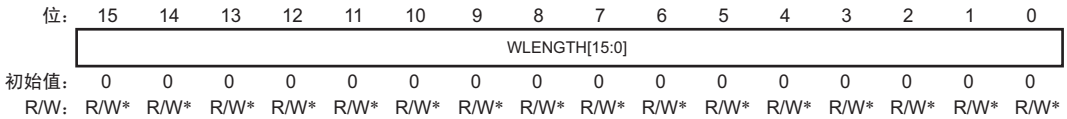
【注】 *　选择功能控制器功能时，只能读。另一方面，选择主控制器功能时，能读 / 写。

25.3.27　USB 请求长度寄存器 （USBLENG）

该寄存器保存控制传送的设置请求。

选择外围控制器功能时，保存接收的 wLength 的值。选择主机控制器功能时，设定发送的 wLength 的值。

通过上电复位、软件复位及 USB 总线复位初始化本寄存器。



位	位名	初始值	R/W	说明
15 ～ 0	WLENGTH[15:0]	H'0000	R/W*	长度 保存 USB 请求 wLength 的值。

【注】 *　选择功能控制器功能时，只能读。另一方面，选择主机控制器功能时，能读 / 写。

25.3.28　DCP 结构寄存器 （DCPCFG）

对于默认控制管道（DCP），该寄存器选择是连续传送模式还是非连续传送模式、是否继续 PIPE 以及传送结束时是否禁止管道运行。

通过上电复位及软件复位初始化本寄存器。

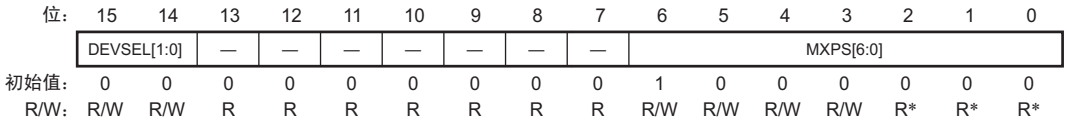
位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CNTMD	SHTNAK	—	—	DIR	—	—	—	—
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R	R	R	R

位	位名	初始值	R/W	说明
15 ～ 9	—	均为 0	R	保留位 读写的值总是 0。
8	CNTMD	0	R/W	连续传送模式 0：非连续传送模式 1：连续传送模式 由于控制读传送及控制写传送共同使用 DCP 缓冲存储器，所以不管 CNTMD 位为哪个传送方向都为共同的位。
7	SHTNAK	0	R/W	DCP 的传送结束时，管道禁止模式 0：传送结束时，管道继续 1：传送结束时，禁止管道 （响应 PID 为 NAK）
6、5	—	0	R	保留位 读写的值总是 0。
4	DIR	0	R/W	传送方向 选择主机控制器功能时，设定控制传送的数据阶段及状态阶段的传送方向。选择功能控制器功能时，必须将本位清 0。 0：数据接收方向 1：数据发送方向
3 ～ 0	—	均为 0	R	保留位 读写的值总是 0。

25.3.29　DCPMax 包大小寄存器 （DCPMAXP）

该寄存器对 DCP 指定 Max 包大小。

通过上电复位及软件复位初始化本寄存器。



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

位	位名	初始值	R/W	说明
15、 14	DEVSEL[1:0]	00	R/W	器件选择 指定选择主机控制器功能时的通信对象的器件地址。选择功能控制器功能时，将本位的值设定为 B'00。 00：地址 00 01：地址 01 10：地址 10 11：地址 11
13 ～ 7	—	均为 0	R	保留位 读写的值总是 0。
6 ～ 0	MXPS[6:0]	H'40	R/W*	Max 包大小 指定 DCP 的 Max 包大小。 不能进行 USB 规格以外的设定。同时， MXPS[2:0] 被固定为 0。

【注】 *　对 MXPS[2:0] 的写入无效。

25.3.30　DCP 控制寄存器 （DCPCTR）

该寄存器对 DCP 进行缓冲存储器状态的确认、数据 PID 顺序位的变更和确认以及响应 PID 的设定。通过上电复位及软件复位初始化本寄存器。另外，通过 USB 总线复位初始化 CCPL 和 PID[2:0] 位。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BSTS	SUREQ	—	—	—	—	—	SQCLR	SQSET	SQMON	—	—	—	CCPL	PID[1:0]	
初始值：	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
R/W：	R	R/W*2	R	R	R	R	R	R*1/ W*2	R*1/ W*2	R	R	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15	BSTS	0	R	缓冲器状态 0：不能存取缓冲器 1：能存取缓冲器 通过 CFIFOSEL.ISEL 位的设定值决定缓冲器存取的方向是写还是读。
14	SUREQ	0	R/W*2	发送设置权标 通过将本位置 1，发送设置包。设置处理结束后，本模块将本位清 0。另外，本位的值为 1 期间，不能对 USBREQ、USBVAL、USBINDX 和 USBLENG 寄存器进行写入。 0：无效 1：发送设置包
13 ～ 9	—	均为 0	R	保留位 读写的值总是 0。
8	SQCLR	0	R*1/W*2	交替位清除 *3*4 0：无效 1：指定 DATA0
7	SQSET	0	R*1/W*2	交替位置位 *3*4 0：无效 1：指定 DATA1
6	SQMON	1	R	交替位确认 0：DATA0 1：DATA1 选择功能控制器功能时，在接收控制传送的 SETUP 权标后，立即通过本模块将本位初期化为 1。
5 ～ 3	—	均为 0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说明
2	CCPL	0	R/W	允许控制传送结束 0：无效 1：控制传送结束 选择功能控制器功能时，在接收 SETUP 权标后本位立即被清 0。选择主机控制器功能时，请将本位清 0。
1、0	PID[1:0]	00	R/W	响应 PID 00：NAK 响应 01：BUF 响应 （根据缓冲器状态） 10：STALL 响应 11：STALL 响应 选择功能控制器功能时，本位在接收 SETUP 权标后立即被清除为 B'00。同时，发生传送错误等时，通过本控制器设定本位，结束传送。

【注】 *1　只有读 0 有效。

*2　只有写 1 有效。

*3　不能同时将 SQCLR 位和 SQSET 位置 1。另外，位操作前必须设定 PID=NAK。

*4　连续变更 SQSET 位或者 SQCLR 位，以及 PIPEnCTR 寄存器的 SQSET 位或者 SQCLR 位时 （连续变更多个管道的数据 PID 顺序交替位时），需要 120ns+ 外围时钟 5 个时钟以上的存取周期。

25.3.31　管道窗口选择寄存器 （PIPESEL）

在管道 1 ～ 7 中，选择使用的管道。选择管道之后，对 PIPECFG 寄存器、 PIPEBUF 寄存器、 PIPEMAXP 寄存器和 PIPEPERI 寄存器进行各管道的功能设定。 PIPEnCTR 寄存器可设定为与通过 PIPESEL 寄存器的管道选择无关。

上电复位、软件复位及 USB 总线复位时，不仅是选择管道，全部管道的寄存器的相应位都被初始化。

通过上电复位及软件复位初始化本寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PIPESEL[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	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15 ～ 3	—	均为 0	R	保留位 读写的值总是 0。
2 ～ 0	PIPESEL[2:0]	000	R/W	管道窗口选择 000：未选择 001：管道 1 010：管道 2 011：管道 3 100：管道 4 101：管道 5 110：管道 6 111：管道 7 设定 PIPESEL=000 时，PIPECFG 寄存器、PIPEBUF 寄存器、PIPEMAXP 寄存器、 PIPEPERI 寄存器及 PIPEnCTR 寄存器的各位全部读出 0。

25.3.32　管道结构寄存器 （PIPECFG）

对管道 1 ～ 7 选择各管道的传送类型、缓冲存储器的存取方向及端点编号的指定、是连续传送模式还是非连续传送模式、是单缓冲器还是双缓冲器以及是否禁止传送结束时的管道运行。

通过上电复位及软件复位初始化本寄存器。另外，通过 USB 总线复位初始化 TYPE[1:0]。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TYPE[1:0]		—	—	—	BFRE	DBLB	CNTMD	SHT NAK	—	—	DIR	EPNUM[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	R	R	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15、 14	TYPE[1:0]	00	R/W	传送类型 • 管道 1、 2 时 00：不能使用管道 01：批量传送 10：禁止设定 11：同步传送 * • 管道 3 ～ 5 时 00：不能使用管道 01：批量传送 10：禁止设定 11：禁止设定 • 管道 6、 7 时 00：不能使用管道 01：禁止设定 10：中断传送 11：禁止设定
13 ～ 11	—	均为 0	R	保留位 读写的值总是 0。
10	BFRE	0	R/W	指定 BRDY 中断运行 0：通过发送 / 接收数据，指定 BRDY 中断 1：通过读取数据，指定 BRDY 中断 如果设定 BFRE=1，为数据写入方向时不产生 BRDY 中断。

位	位名	初始值	R/W	说明
9	DBLB	0	R/W	双缓冲器模式 0：单缓冲器 1：双缓冲器 本位在选择管道 1 ～ 5 时有效。 对某管道动态切换本位的设定时，请遵守以下步骤。 • 从单缓冲器向双缓冲器切换时 （DBLB=0→DBLB=1） 1. 将相应管道的响应 PID 设定为 NAK 2. PIPEnCTR.ACLRM=1 3. 用软件等待 100ns 4. ACLRM=0 5. 变更 DBLB 位 6. 将响应 PID 设定为 BUF • 从双缓冲器向单缓冲器切换时 （DBLB=1→DBLB=0） 1. 将相应管道的响应 PID 设定为 NAK 2. 变更 DBLB 位 3. PIPEnCTR.ACLRM=1 4. 用软件等待 100ns 5. ACLRM=0 6. 将响应 PID 设定为 BUF
8	CNTMD	0	R/W	连续传送模式 本位在管道 1 ～ 5 选择批量传送时 （TYPE=01）有效。选择同步传送时（TYPE=11）不能设定 CNTMD=1。另外，在选择了管道 6、7 时，不能设定 CNTMD=1。 0：非连续传送模式 1：连续传送模式
7	SHTNAK	0	R/W	传送结束时的管道禁止 0：传送结束时管道继续 1：传送结束时管道禁止
6、5	—	均为 0	R	保留位 读写的值总是 0。
4	DIR	0	R/W	传送方向 0：接收方向 1：发送方向
3 ～ 0	EPNUM[3:0]	0000	R/W	端点编号 指定该管道的端点编号。

【注】 * 使用同步 OUT 传送时，请参照 “25.5.1 使用同步 OUT 传送时的注意事项”。

25.3.33　管道缓冲器指定寄存器 （PIPEBUF）

该寄存器对管道 1 ～ 7 指定缓冲器大小及缓冲器编号。

通过上电复位或软件复位初始化本寄存器。

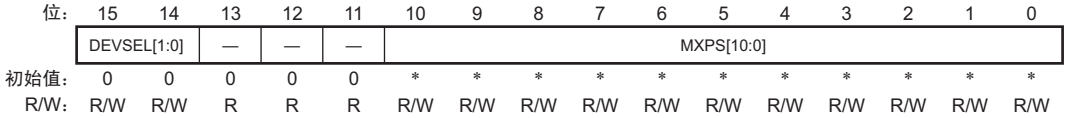
位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	BUFSIZE[4:0]					—	—	—	BUFNMB[6:0]						
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写的值总是 0。
14 ～ 10	BUFSIZE [4:0]	H'00	R/W	缓冲器大小 指定相应管道的缓冲器大小。 （从 0：64 字节到 H'1F：2K 字节） 根据 PIPESEL 寄存器的 PIPESEL 位选择的管道，本位能设定的值有所不同。 管道 1 ～ 5 时：设定 BUFSIZE=H'00 ～ H'1F。 管道 6、 7 时：设定 BUFSIZE=H'00。
9 ～ 7	—	均为 0	R	保留位 读写的值总是 0。
6 ～ 0	BUFNMB [6:0]	H'00	R/W	缓冲器编号 指定相应管道的缓冲器编号 （H'04 ～ H'7F）。 在选择管道 1 ～ 5 时，能配合系统设定 BUFNMB 位。 BUFNMB=0 ～ 3 为 DCP 专用。 BUFNMB=4、 5 被配置到管道 6、 7。 管道 1 ～ 5 时：设定 BUFNMB=H'06 ～ H'7F。 但是，不使用 PIPE7 时， 能设定 BUFNMB=H'05 ～ H'7F。 不使用管道 6、 7 时， 能设定 BUFNMB=H'04 ～ H'7F。 管道 6 时：对本位写无效，读出值总为 BUFNMB=4。 管道 7 时：对本位写无效，读出值总为 BUFNMB=5。

25.3.34　管道 Max 包大小寄存器 （PIPEMAXP）

该寄存器对管道 1 ～ 7 指定 Max 包大小。

通过上电复位或软件复位初始化本寄存器。



位	位名	初始值	R/W	说明
15、14	DEVSEL[1:0]	00	R/W	器件选择 选择主机控制器功能时指定外围器件地址。选择功能控制器功能时，设定本位的值为 B'00。 00：地址 00 01：地址 01 10：地址 10 11：地址 11
13 ～ 11	—	均为 0	R	保留位 读写的值总是 0。
10 ～ 0	MXPS[10:0]	*	R/W	Max 包大小 指定相应管道的 Max 包大小。 按传送类型给本位设定 USB 规格定义范围内的值。

【注】 * 不用 PIPESEL 寄存器的 PIPESEL 位选择管道时为 H'000，用 PIPESEL 寄存器的 PIPESEL 位选择时为 H'040。

25.3.35　管道周期控制寄存器 （PIPEPERI）

该寄存器对管道 1、2 选择否通过同步 IN 传送时的间隔错误运行缓冲器闪存功能，以及设定间隔错误的检测间隔。

通过上电复位或软件复位初始化本寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	IFIS	—	—	—	—	—	—	—	—	—	IITV[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/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15 ～ 13	—	均为 0	R	保留位 读写的值总是 0。
12	IFIS	0	R/W	同步 IN 缓冲器闪存 0：不清空缓冲器 1：清空缓冲器 本位只在选择功能控制器功能时有效。 使用本位时，需要按照以下步骤设定。 • 同步 IN 传送开始时 1. IFIS=1 2. PIPEnCTR.PID[1:0]=01 （BUF） 3. 将发送数据写到 Iso-IN PIPE 的 FIFO 缓冲器 【注】在不使用 IFIS 的传送时，没有必要进行上述步骤。 • 同步 IN 传送结束时 1. PID[1:0]=00 （NAK） 2. PIPEnCTR.ACLR=1 3. 等待 100ns 以上 4. ACLR=0 【注】在不使用 IFIS 的传送时，没有必要进行 ACLR 的设定。
11 ～ 3	—	均为 0	R	保留位 读写的值总是 0。
2 ～ 0	IITV[2:0]	000	R/W	间隔错误检测间隔 本位按照帧时序的 2 的 n 次方指定间隔时序 这些位只在选择功能控制器功能、同步传送时有效。即，只有选择管道 1、2 时才能设定。 设定 OUT 方向时：在本位设定的间隔，如果不接收来自主机的 OUT 权标时，就在 NRDY 中断检测出间隔错误，并发生 NRDY 中断。 设定 IN 方向时　：在本位设定的间隔，如果不接收来自主机的 IN 权标时，检测出间隔错误，并且 IFIS=1 就进行缓冲器闪存 （清除缓冲器）。 选择主机控制器功能时，本位对同步传送和中断传送有效。

25.3.36　管道 n 控制寄存器 （PIPEnCTR）（n=1 ～ 7）

该寄存器对管道 1 ～ 7，确认相应管道的缓冲存储器状态、变更和确认数据 PID 顺序位、选择是否设定为自动响应模式、选择是否设定为自动缓冲器清除模式以及设定响应 PID。本设定与通过 PIPESEL 寄存器的管道选择无关。

通过上电复位或软件复位初始化本寄存器。另外，通过 USB 总线复位初始化 PID[1:0] 位。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR	SQSET	SQMON	—	—	—	—	PID[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/W	R/W	R/W*1	R/W*1	R	R	R	R	R	R/W	R/W

位	位名	初始值	R/W	说明
15	BSTS	0	R	缓冲状态 0：不能存取缓冲器 1：能存取缓冲器 通过 PIPECFG 寄存器的 DIR 位决定缓冲器存取方向是写还是读。详细请参照 “25.4 运行说明”。
14	INBUFM	0	R	发送缓冲器监控 本位表示将相应管道设定为发送方向时有效的值。 0：缓冲存储器中没有能发送的数据 1：缓冲存储器中有能发送的数据 【注】本位对管道 1 ～ 5 有效。
13 ～ 11	—	均为 0	R	保留位 读写的值总是 0。
10	ATREPM	0	R/W	自动响应模式 0：普通模式 1：自动响应模式 【注】本位对管道 1 ～ 5 有效。
9	ACLRM	0	R/W	自动缓冲器清除模式 0：禁止 1：允许 （初始化全部缓冲器） 不能对 CFIFOSEL/DnFIFOSEL 寄存器的 CURPIPE 位选择的管道设定 ACLRM=1。
8	SQCLR	0	R/W*1	清除交替位 *2*3 0：无效 1：指定 DATA0
7	SQSET	0	R/W*1	交替位置位 *2*3 0：无效 1：指定 DATA1
6	SQMON	0	R	确认交替位 0：DATA0 1：DATA1

位	位名	初始值	R/W	说明
5 ～ 2	—	均为 0	R	保留位 读写的值总是 0。
1、 0	PID[1:0]	00	R/W	响应 PID*3 00：NAK 响应 01：BUF 响应 （根据缓冲器状态） 10：STALL 响应 11：STALL 响应 选择主机控制器功能时，如果不是 PID=BUF 设定，就不发送权标。发生传送错误时等，通过控制器设定 PID 位并结束传送。

【注】 *1　只读 0、写 1 有效。

*2　连续变更 SQSET 位或者 SQCLR 位，以及 DCPCTR 寄存器的 SQSET 位或者 SQCLR 位时 （连续变更多个管道的数据 PID 的顺序交替位时），需要至少 120ns+ 总线时钟 5 个周期的存取周期。

*3　不能同时将 SQCLR 位及 SQSET 位置 1。另外，位操作前必须设定 PID=NAK。传送类型设定为同步传送（TYPE=11）时，对 SQSET 位的写入无效。

25.3.37　USB AC 特性转换寄存器 （USBACSWR）

USBACSWR 设定本模块内置的 USB 收发器。

通过上电复位初始化本寄存器。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	UACS23	-	-	-	-	-	-	-
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	R	R	R	R	R	R

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
31 ～ 24	—	均为 0	R	保留位 读写的值总是 0。
23	UACS23	0	R/W	USB AC 特性开关 进行 USB 收发器的设定。 *
22 ～ 0	—	均为 0	R	保留位 读写的值总是 0。

【注】 *　使用本模块时，必须给本位写 1。详细内容请参照 “25.5.2 USB 收发器的设定步骤”。

25.4　运行说明

25.4.1　系统控制及振荡控制

本节对本模块的初始设定所必需的寄存器操作以及进行功耗控制所必需的寄存器进行说明。

（1） 复位

本模块的复位类别一览如**表 25.10** 所示。另外，关于各复位操作后的寄存器初始化状态，请参照“**25.3 寄存器说明**”。

表 25.10　复位类别一览表

名称	操　　作
上电复位	从 $\overline{\text{RST}}$ 引脚输入低电平
软件复位	使用 SYSCFG 寄存器的 USBE 位进行操作
USB 总线复位	选择功能控制器功能时，本模块自动从 D+、 D- 线检测

（2） 选择控制器功能

本模块能选择主机控制器功能或者功能控制器功能。控制器功能的选择必须用 SYSCFG 寄存器的 DCFM 位进行。 DCFM 位的变更，必须在上电复位后的初始设定时进行。

（3） 允许高速运行

本模块能通过软件设定 USB 通信速度（通信位速率）是高速运行还是全速运行。允许本模块高速运行时，请将 SYSCFG 寄存器的 HSE 位置 1。请在上电复位后的初始设定或者 D+ 的上拉禁止状态 （DPRPU=0）时变更 HSE 位。

允许高速运行时，本模块执行复位信号交换协议，并自动设定 USB 通信速度。能通过 DVSTCTR 寄存器的 RHST 位确认复位信号交换的结果。

禁止高速运行时，本模块只能全速运行。

（4） USB 数据总线电阻控制

本模块与 USB 连接器的连接图如图 25.1 所示。

本模块内置 D+ 信号的上拉电阻和 D+、D- 信号的下拉电阻。请通过 SYSCFG 寄存器的 DPRPU、DMRPD 位设定各信号的上拉、下拉。

同时，本模块控制 D+、D- 信号高速运行时的终端电阻和全速运行时的输出电阻。本模块通过复位信号交换、挂起、恢复检测，自动进行与主机控制器或者外围器件连接后的内置电阻切换。检测出来自主机控制器或者外围器件的断开时，请通过软件复位 （SYSCFG.USBE=0）将本模块置为初始状态。

另外，选择功能控制器功能，并且在与主机控制器通信时，如果设定 SYSCFG 寄存器的 DPRPU 位为 0，因为 USB 数据线的上拉电阻 （或者终端电阻器）为禁止，所以在连接 USB 线的状态下，能通过软件控制器件的断开和连接。

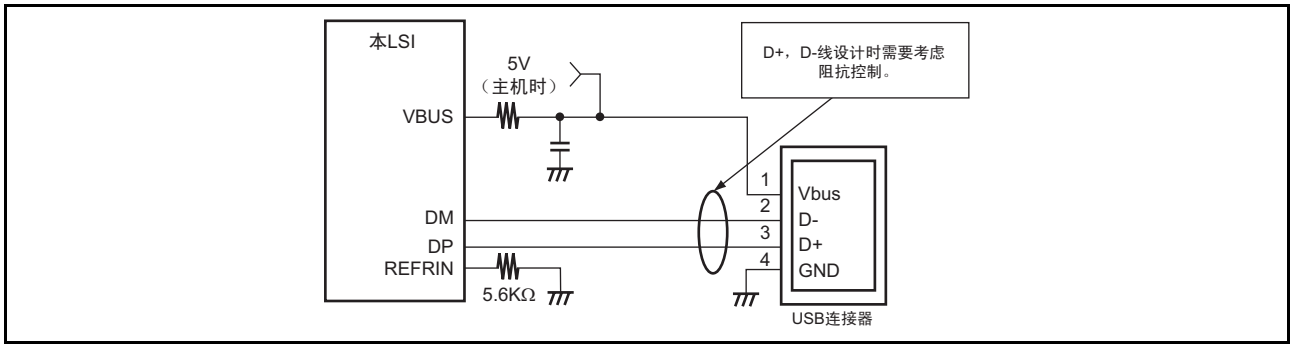


图 25.1　USB 连接器连接图

25.4.2　中断功能

本模块的中断产生条件一览表如**表 25.11** 所示。

这些中断产生条件成立，并且通过对应的中断允许寄存器设定允许中断输出时，本模块向中断控制器（INTC）输出 USB 中断请求信号。

表 25.11　中断产生条件一览表

位	中断名称	中断产生条件	产生的功能	相关状态
VBINT	VBUS 中断	• 检测出 VBUS 输入引脚的状态变化时（L→H、 H→L）	主机、功能	VBSTS
RESM	恢复中断	• 在挂起状态，检测出 USB 总线的状态变化时（J-State→K-State 或者 J-State→SE0）	功能	—
SOFR	帧号更新中断	［选择主机控制器功能时］ • 发送了与帧号不同的 SOF 包时 ［选择功能控制器功能时］ • SOFRM=0：接收了与帧号不同的 SOF 包时 • SOFRM=1：由于 μ 帧号为 0 时的 SOF 包破损等原因而不能接收时	主机、功能	—
DVST	器件状态转移中断	• 检测出器件状态转移时 检测出 USB 总线复位 检测出挂起状态 接收到 Set Address 请求 接收到 Set Configuration 请求	功能	DVSQ
CTRT	控制传送阶段转移中断	• 检测出控制传送的阶段转移时 设置阶段结束 控制写入传送状态阶段转移 控制读出传送状态阶段转移 控制传送结束 发生控制传送顺序错误	功能	CTSQ
BEMP	缓冲器空中断	• 发送缓冲存储器中的全部数据，缓冲器为空时 • 接收的包超过 Max 包大小时	主机、功能	BEMPSTS. PIPEBEMP
NRDY	缓冲器未就绪中断	［选择主机控制器功能时］ • 对于发送的权标，接收到来自功能器件的 STALL 时 • 对于发送的权标，没有来自功能器件的响应时（无响应） • 在同步传送中发生上溢 / 下溢时 • ［选择功能控制器功能时］ • 接收 IN 权标时，没有能向缓冲存储器发送的数据时 • 接收 OUT 权标时，由于缓冲存储器中没有数据保存区域，而不能接收时 • 在同步传送中接收数据时，发生 CRC 错误、位填充错误时	主机、功能	NRDYSTS. PIPENRDY
BRDY	缓冲器就绪中断	• 为缓冲器就绪 （可读或者可写状态）时	主机、功能	BRDYSTS. PIPEBRDY

位	中断名称	中断发生条件	发生的功能	关联状态
BCHG	总线变化中断	• 检测出 USB 总线状态变化时	主机、功能	—
DTCH	全速运行时的断开检测	• 全速运行时检测出功能器件断开时	主机	—
SACK	SETUP 正常	• 接收到设置处理的正常响应 （ACK）时	主机	—
SIGN	SETUP 错误	• 检测出设置处理的错误 （无响应、 ACK 包破损）时	主机	—

本模块的中断相关图如图 25.2 所示。

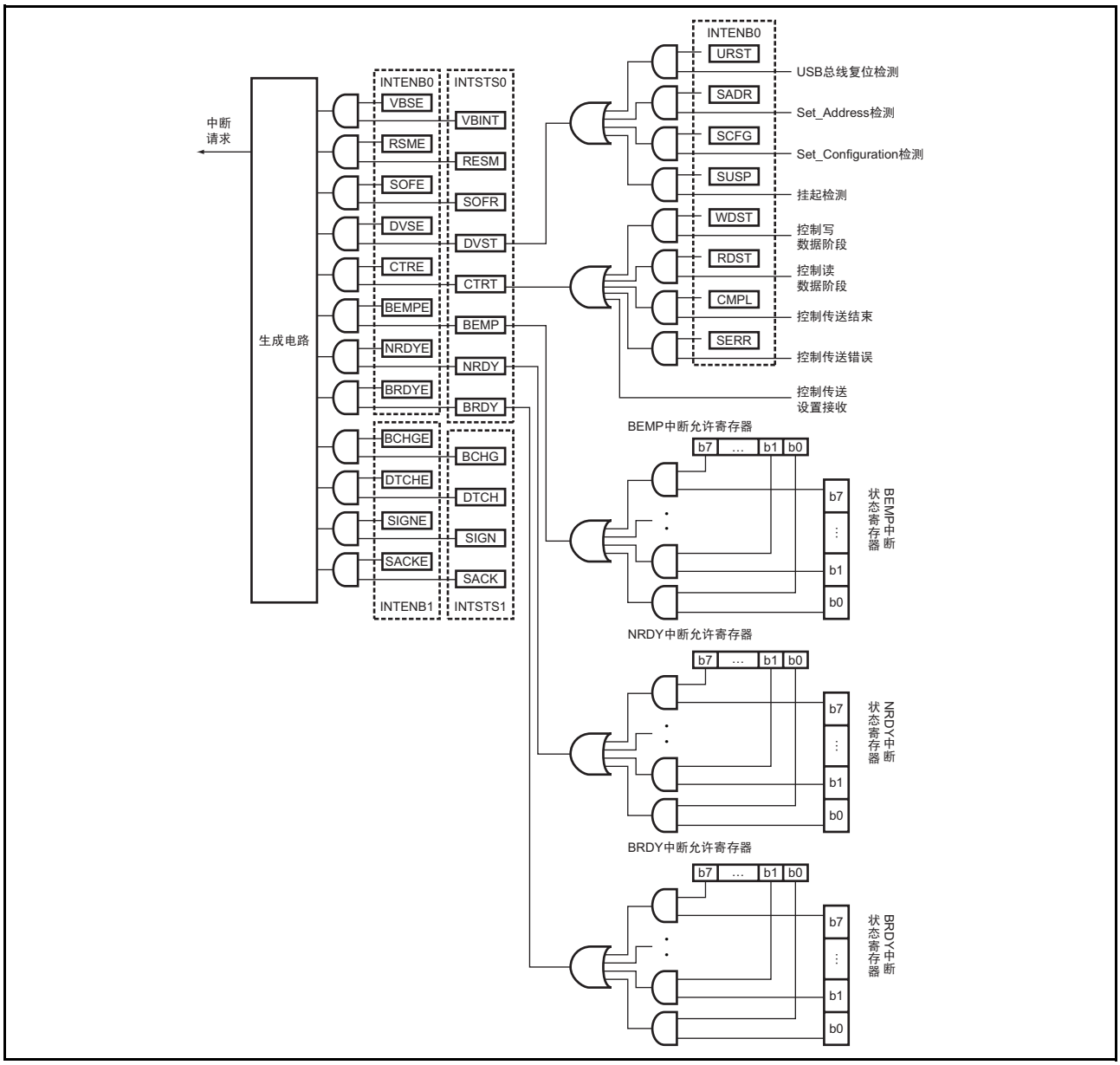


图 25.2　中断的相关图

（1） BRDY 中断

在选择主机控制器或功能控制器的功能时产生 BRDY 中断。各管道满足表 25.12 的条件时，本模块将 BRDYSTS 寄存器的对应位置 1。此时，如果软件将对应相应管道的 BRDYENB 寄存器的 PIPEBRDYE 位置 1，并且将 INTENB0 寄存器的 BRDYE 位置 1，本模块就产生 BRDY 中断。

BRDY 中断产生时序图如图 25.3 所示。

本模块清除 INTSTS0 寄存器的 BRDY 位的条件，因 INTENB1 寄存器的 BRDYM 位的设定值而不同。BRDY 位清除条件如表 25.13 所示。

另外，选择功能控制器功能时，下列 1 的条件中总对 IN 权标发送 Zero-Length 包，且不产生 BRDY 中断。

1. 设定传送类型为批量IN传送、设定PID=BUF，并且设定相应管道的PIPEnCTR寄存器的ATREPM位为H'01时

表 25.12　BRDY 中断产生条件表

存取方向	传送方向	管道	BFRE	DBLB	BRDY 中断产生条件
读出	接收	DCP	—	—	进行下列接收时 • 接收含 Zero-Length 包的短包 • 接收数据直到缓冲器满
		1 ～ 7	0	0	进行下列接收时 • 接收含 Zero-Length 包的短包 • 接收数据直到缓冲器满 * • 缓冲器未满，但处理计数器结束
				1	满足下列任何一个条件时 • 在缓冲器两侧都为等待接收的状态下，用 1. ～ 3. 的任何一种方法进行接收 1. 接收含 Zero-Length 包的短包 2. 接收数据直到缓冲器的一侧为满 * 3. 缓冲器未满，但处理计数器结束 • 在缓冲器两侧都为等待读出的状态下，读完一侧的接收数据 • 在缓冲器两侧都为等待读出的状态下，通过软件设置 BCLR=1，清除一侧的接收数据 • 设定为连续传送模式时 （PIPECFG.CNTMD=1），在 SIE 侧的缓冲器有数据的状态下，设定 CFIFOSIE.TGL=1
			1	Don't Care	满足下列任何一个条件时 • 接收 Zero-Length 包 • 接收短包，且完成相应短包的数据读出 • 处理计数器结束，且完成最后包的数据读出

【注】 * 该缓冲器满即：在设定为非连续传送 （CNTMD=0）时接收到 Max 包大小的数据，在设定为连续传送（CBTMD=1）时接收到缓冲器大小的数据。

存取方向	传送方向	管道	BFRE	DBLB	BRDY 中断产生条件
写入	发送	DCP	—	—	不产生
		1 ～ 7	0	0	满足下列任何一个条件时 • 通过软件将传送方向从接收变更为发送 • 在缓冲器有可发送数据的状态下，完成相应数据的发送 • 在缓冲器有可发送数据的状态下，软件设定 ACLRM=1 • 在缓冲器有可发送数据的状态下，软件设定 SCLR=1
				1	满足下列任何一个条件时 • 通过软件将传送方向从接收变更为发送 • 缓冲器没有可发送数据的状态下，用以下 1. ～ 3. 的任何一种方法使数据为可发送状态 1. 将 Max 包大小的 n 倍的数据写入缓冲器，缓冲器变满 （非连续传送时：n=1） 2. 软件设定 DnFIFOCTR.BVAL=1，使缓冲器能进行发送 3. 在 DMA 传送中写入结束 • 缓冲器两侧有可发送数据的状态下，完成一侧数据的发送 • 缓冲器两侧有可发送数据的状态下，软件设定 ACLRM=1 • 缓冲器两侧有可发送数据的状态下，软件设定 SCLR=1
			1	Don't Care	不产生

接收到 Zero-Length 包时， BRDYSTS 寄存器的相应位为 1，但不能读出该包的数据。请在清除 BRDYSTS 寄存器之后进行缓冲器清除 （BCLR=1）。

同时在管道 1 ～ 7，如果在读出方向使用 DMA 传送，就能通过将 PIPECFG 寄存器的 BFRE 位置 1，按传输单位产生中断。

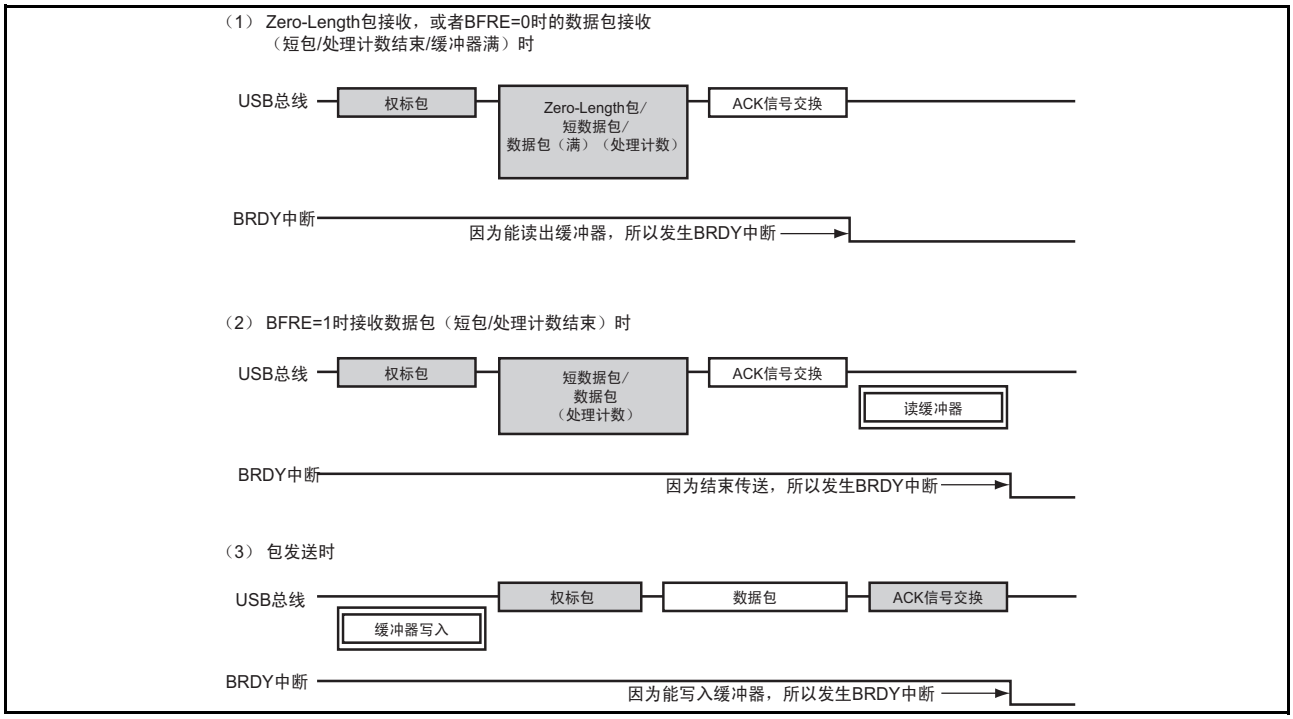


图 25.3　BRDY 中断产生时序图

表 25.13　本模块的 BRDY 位清除条件表

BRDYM	BRDY 位清除条件
0	如果软件清除 BRDYSTS 寄存器的全部位，本模块就清除 INTSTS0 寄存器的 BRDY 位。
1	所有管道的 BSTS 位为 0 时，本模块清除 INTSTS0 寄存器的 BRDY 位。

（2） NRDY 中断

各管道满足下列条件时，本模块将 NRDYSTS 寄存器的相应位置 1。此时，如果软件将对应该管道的 NRDYENB 寄存器的 PIPENRDYE 位置 1，并且将 INTENB0 寄存器的 NRDYE 位置 1，本模块就产生 NRDY 中断。如果软件清除 NRDYSTS 寄存器的全部位，本模块就清除 INTSTS0 寄存器的 NRDY 位。

（a） 选择主机控制器功能时

通过以下任何一个条件产生 NRDY 中断。此时，硬件设定 PID 位，并停止发行权标。关于 PID 位的运行，请参照 “25.4.3 （4） 响应 PID”。

- 对于发送的权标，接收来自外围的STALL时
- 对于发送的权标，没有来自外围的响应时 （无响应）
- 在同步传送中发生上溢/下溢错误时

但在设置处理中不能接收外围的 ACK 时，产生 SIGN 中断。

（b） 选择功能控制器功能时

通过以下条件产生 NRDY 中断

1. 发送数据时
   - 在PIPEnCTR寄存器的PID位设定为BUF，并且缓冲存储器中没有发送数据的状态下，接收到IN权标 （下溢）时
2. 接收数据时
   - 在PIPEnCTR寄存器的PID位设定为BUF，并且缓冲存储器中没有保存接收数据的区域的状态下，接收到OUT权标或者PING权标 （上溢）时
   - 传送类型为批量传送时，未设定Max包大小 （MXPS=0）的管道接收到OUT权标或者PING权标时
   - 同步传送中，发生CRC错误、位填充错误时
   - 同步传送中，在间隔帧以外接收了权标时 （间隔错误）

选择功能控制器功能时的 NRDY 中断产生时序图如图 25.4 所示。

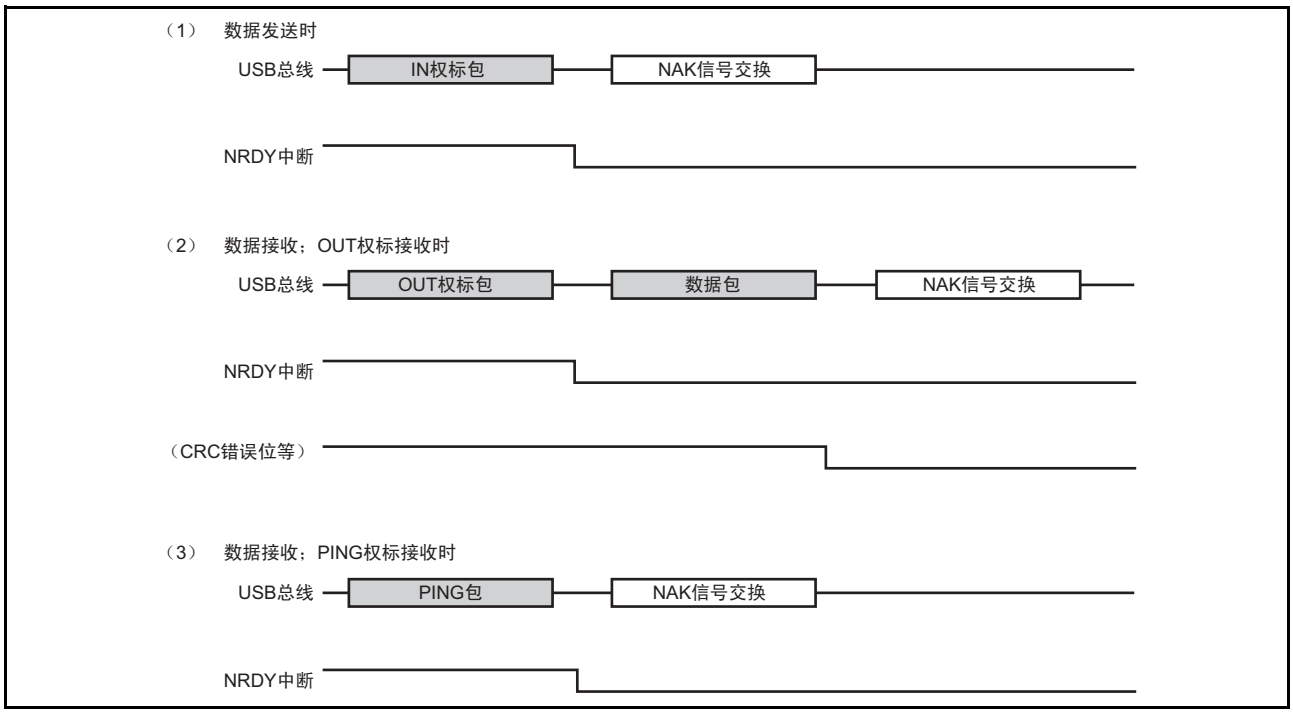


图 25.4　选择功能控制器功能时的 NRDY 中断产生时序图

（3） BEMP 中断

各管道满足下列条件时，本模块将 BEMPSTS 寄存器的相应位置 1。此时，如果软件将该管道对应的 BEMPENB 寄存器的 PIPEBEMPE 位置 1，并且将 INTENB0 寄存器的 BEMPE 位置 1，本模块就产生 BEMP 中断。如果软件清除 BEMPSTS 寄存器的全部位，本模块就清除 INTSTS0 寄存器的 BEMP 位。

1. 设定发送方向 （写入缓冲存储器）时
   发送了保存到缓冲存储器的全部数据时。
   但是缓冲存储器用作双缓冲器时，遵从以下条件。
   - 一侧的缓冲器为空状态，且另一侧的缓冲器的数据发送结束时，产生BEMP中断。
   - 对一侧的缓冲器写入的数据不满8字节，且另一侧的缓冲器数据发送结束时，产生BEMP中断。
2. 设定发送方向 （读取缓冲存储器）时
   接收的数据包大小超过设定的Max包大小时。
   此时，如果将Max包大小设定不为0 （MXPS≠0）时，本模块就将相应管道的PID位设定为STALL。

选择功能控制器功能时的 BEMP 中断产生时序图如图 25.5 所示。

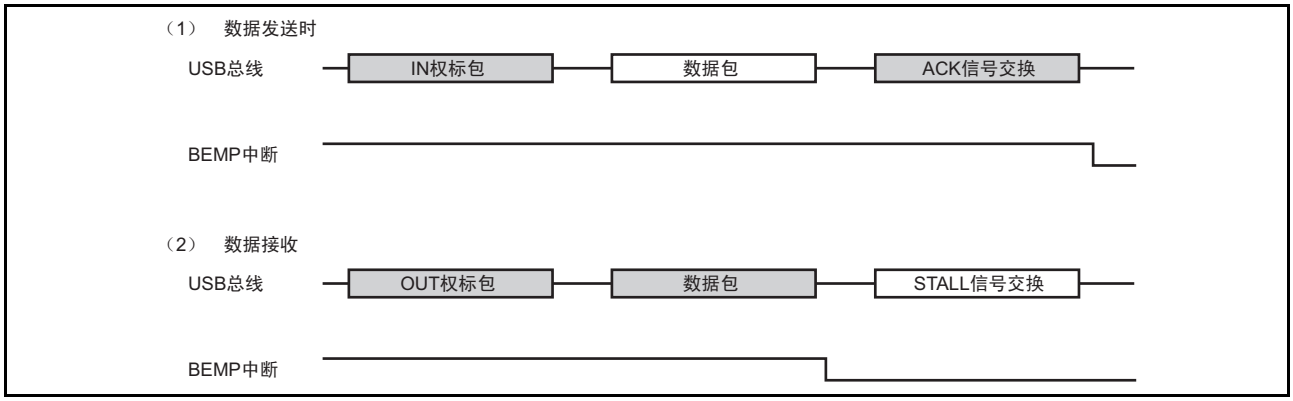


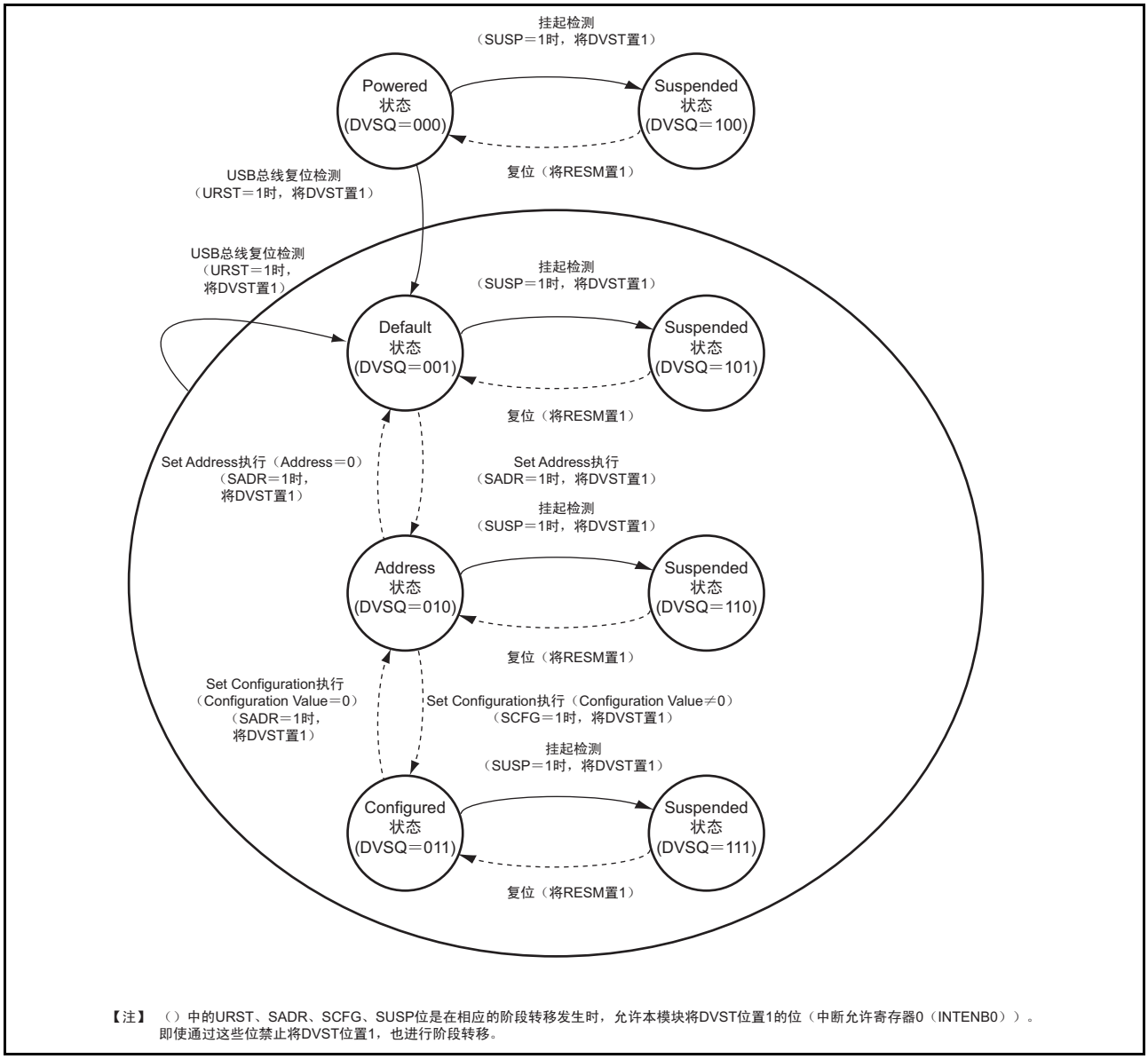
图 25.5　选择功能控制器功能时的 BEMP 中断产生时序图

（4） 器件状态转移中断

本模块的器件状态转移图如图 25.6 所示。本模块管理器件状态，并产生器件状态转移中断。但是，由恢复中断检测从挂起状态的返回（恢复信号检测）。通过 INTENB0 寄存器能个别禁止或者允许器件状态转移中断。同时，能通过 INTSTS0 寄存器的 DVSQ 位确认转移的器件状态。

转移到默认状态时，在复位信号交换协议结束后，产生器件状态转移中断。

只能在选择功能控制器功能时进行器件状态的管理。器件状态转移中断也只能在选择功能控制器功能时产生。



【注】 （）中的URST、SADR、SCFG、SUSP位是在相应的阶段转移发生时，允许本模块将DVST位置1的位（中断允许寄存器0（INTENB0））。
即使通过这些位禁止将DVST位置1，也进行阶段转移。

图 25.6　器件状态转移图

（5） 控制传送阶段转移中断

本模块的控制传送阶段转移图如图 25.7 所示。本模块管理控制传送的顺序，并产生控制传送阶段转移中断。通过 INTENB0 寄存器能个别允许或者禁止控制传送阶段转移中断。同时，能通过 INTSTS0 寄存器的 CTSQ 位确认转移的传送阶段。

控制传送的顺序错误如下所示。发生错误时， DCPCTR 寄存器的 PID 位为 B'1x （STALL 响应）。

1. 控制读传送时
   - 在数据完全没有传送的状态下，对于数据阶段的IN权标，接收OUT或者PING权标
   - 在状态阶段接收IN权标
   - 在状态阶段接收数据包为DATAPID=DATA0的包
2. 控制写传送时
   - 在完全没有ACK应答的状态下，对于数据阶段的OUT权标，接收IN权标
   - 在状态阶段接收最初的数据包为DATAPID=DATA0的包
   - 在状态阶段接收OUT或者PING权标
3. 控制写无数据控制传送时
   - 在状态阶段接收OUT或者PING权标

另外，在控制写传送数据阶段，接收数据数超过 USB 请求的 wLength 值时，不能识别为控制传送顺序错误。同时，在控制读传送状态阶段，要接收 Zero-Length 包以外的包，应进行应答响应并正常结束。

由于顺序错误产生 CTRT 中断时 （设定 SERR=1），从系统写入 CTRT=0 （清除中断状态）之前保持 CTSQ=110 的值。因此，即使在保持 CTSQ=110 的状态下接收新的 USB 请求，也不产生设置阶段结束的 CTRT 中断 （本模块保持设置阶段结束，并通过软件清除中断状态后，产生设置阶段结束中断）。

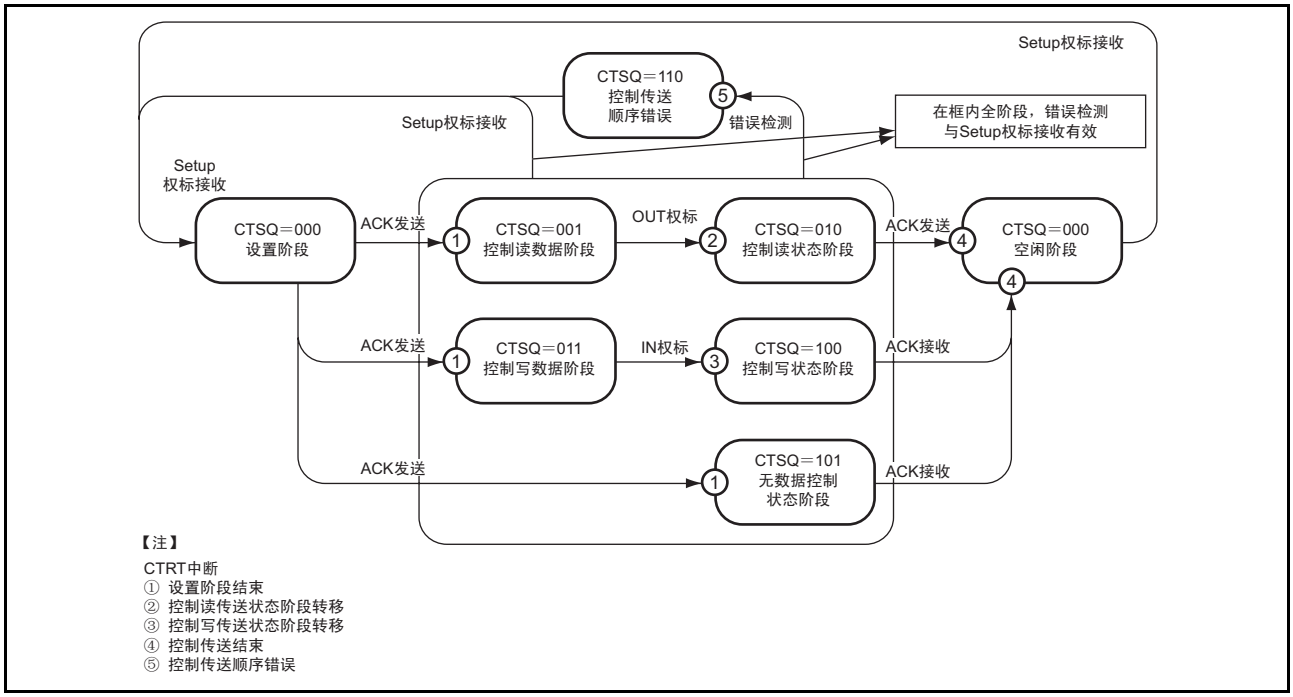


图 25.7　控制传送阶段转移图

（6） 帧更新中断

本模块的 SOFR 中断输出时序例子如图 25.8 所示。

选择主机控制器功能时，在帧号更新时产生时序中断。选择功能控制器功能时，在更新帧号或者检测出SOF包破损时产生 SOFR 中断。请通过 FRMNUM 寄存器的 SOFRM 位指定中断运行。

但在选择主机控制器功能时，请不要设定 SOFRM=1。

1. 选择SOFRM=0时
   在帧号更新时 （约1ms的间隔）产生SOFR中断。 SOF包破损或者丢失时，通过内部插值功能产生中断。高速通信过程中，在帧号更新时 （约1ms的间隔）产生中断。
2. 选择SOFRM=1时
   SOF包破损及丢失时产生SOFR中断。另外在高速通信中，有相同帧号的μSOF包的第一个包破损或者丢失时产生中断
   （通过SOF插值功能识别SOF破损或丢失。详细内容请参照 “25.4.9 SOF插值功能”）。

选择功能控制器功能时，如果本模块在全速运行中检测出新的 SOF 包，就更新帧号并产生 SOFR 中断。但是，如果高速运行中没有成为 μSOF 锁定状态，就不更新帧号且不产生 SOFR 中断。同时， SOF 的插值功能也不运行。所谓 μSOF 锁定状态，即连续 2 次准确接收不同帧号的 μSOF 包。

另外， μSOF 锁定监视开始条件及 μSOF 锁定监视停止条件如下列 1.、 2. 所示。

1. μSOF锁定监视开始条件
   USBE=1
2. μSOF锁定监视停止条件
   USBE=0 （软件复位）、接收USB总线复位或者检测到挂起

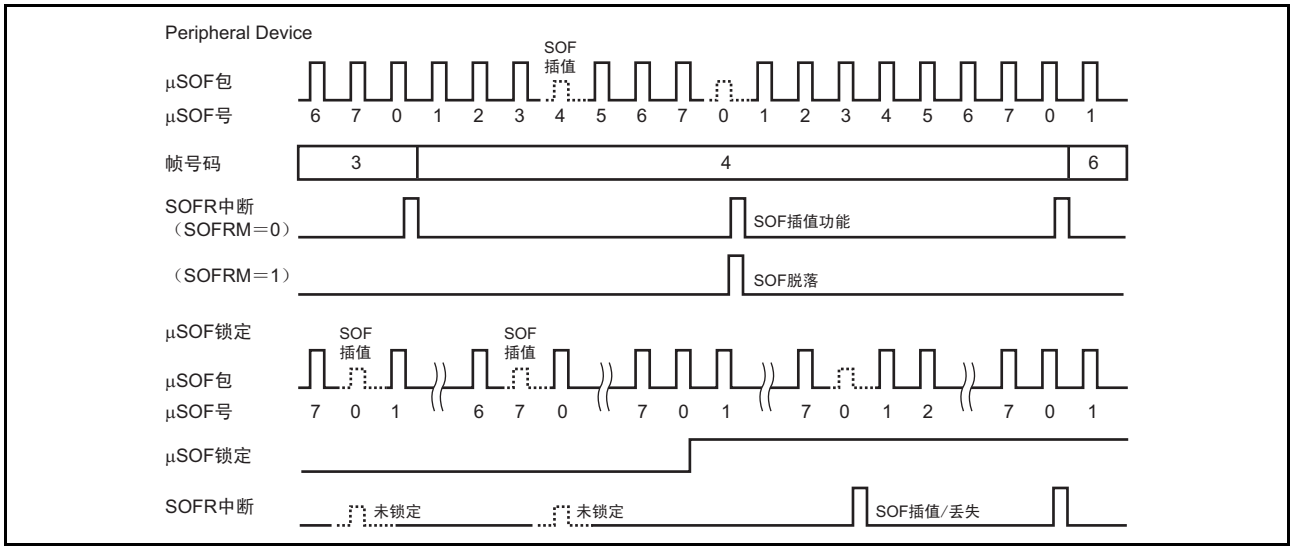


图 25.8　SOFR 中断输出时序例

（7） VBUS 中断

如果 VBUS 引脚发生变化就产生 VBUS 中断。能通过 INTSTS0 寄存器的 VBSTS 位确认 VBUS 引脚的电平。能通过 VBUS 中断确认主机控制器是连接还是断开。但是，如果在主机控制器连接的状态下启动系统，由于 VBUS 引脚不变，所以不发生最初的 VBUS 中断。

（8） 恢复中断

选择功能控制器功能时，器件状态为挂起状态下，如果 USB 总线状态发生变化 （J-State→K-State 或者 J-State→SE0），就产生恢复中断。通过恢复中断检测从挂起状态的返回。

选择主机控制器功能时，不产生恢复中断。请使用 BCHG 中断检测 USB 总线的变化。

（9） BCHG 中断

USB 总线状态发生变化时，产生 BCHG 中断。该中断用于检测是否连接功能器件 （选择主机控制器功能时）和检测远程唤醒。在选择主机控制器功能或者功能控制器功能时，都将产生 BCHG 中断。

（10）DTCH 中断

选择主机控制器功能时，如果在全速运行中检测出器件断开，就产生 DTCH 中断。请注意在高速模式时不会产生该中断。要在高速模式中检测断开，需要定期进行标准请求的控制传送，及在没有外围的响应时判断断开等处理。

请使用 BCHG 中断检测挂起中的分离。

（11）SACK 中断

选择主机控制器功能时，如果对发送的设置包接收来自外围的 ACK 响应，就产生 SACK 中断。能通过 SACK 中断得知设置处理是否正常结束。

（12）SIGN 中断

选择主机控制器功能时，如果对发送的设置包未接收到来自外围的 ACK 响应，就产生 SIGN 中断。中断能用于检测外围没有发送 ACK （无响应）或 ACK 包破损。

25.4.3　管道控制

本模块的管道设定项目一览表如**表 25.14** 所示。USB 数据传送必须通过被称作端点的逻辑管道进行数据通信。本模块有 8 个用于数据传送的管道。请根据系统规格设定各管道。

表 25.14　管道设定项目一览表

寄存器名	位名	设定内容	备注
DCPCFG PIPECFG	TYPE	指定传送类型	请参照 “25.4.3 （1） 传送类型”。
	BFRE	选择 BRDY 中断模式	管道 1 ～ 5：能设定
	DBLB	选择单缓冲器或双缓冲器	管道 1 ～ 5：能设定
	CNTMD	选择连续传送或者非连续传送	DCP：能设定 管道 1、 2：能设定 （只在选择批量传送时能设定） 管道 3 ～ 5：能设定 连续发送 / 接收，设定缓冲器大小为负载的整数倍
	DIR	选择传送方向 （读或者写）	能设定 IN 或者 OUT
	EPNUM	端点编号	请参照 “25.4.3 （2） 端点编号”。
	SHTNAK	选择管道禁止 （传送结束时）	管道 1、 2：能设定 （只在选择批量传送时能设定） 管道 3 ～ 5：能设定
PIPEBUF	BUFSIZE	缓冲存储器大小	DCP：不能设定 （固定 256 字节） 管道 1 ～ 5：能设定 （64 字节为单位最大能指定 2K 字节） 管道 6、 7：不能设定 （固定 64 字节）
	BUFNMB	缓冲存储器编号	DCP：不能设定 （固定 H'0 ～ H'3 区域） 管道 1 ～ 5：能设定 （能在 H'6 ～ H'7F 区域指定） 管道 6、 7：不能设定 （固定 H'4、 H'5 区域）
DCPMAXP PIPEMAXP	MXPS	Max 包大小	请参照 “25.4.3 （3） Max 包大小设定”。
PIPEPERI	IFIS	缓冲器闪存	管道 1、 2：能设定 （只能在选择同步传送时） 管道 3 ～ 7：不能设定
	IITV	间隔计数器	管道 1、 2：能设定 （只能在选择同步传送时） 管道 3 ～ 7：不能设定
DCPCTR PIPEnCTR	BSTS	缓冲器状态	与 DIR 位 /ISEL 位也有关系。
	INBUFM	IN 缓冲器监控	与 DIR 位 /ISEL 位也有关系。
	ACLRM	自动缓冲器清除	设定缓冲存储器读出时，能设定允许 / 禁止该功能。
	SQCLR	顺序清除	清除数据交替位
	SQSET	顺序设定	置位数据交替位
	SQMON	顺序确认	确认数据交替位
	PID	响应 PID	请参照 “25.4.3 （4） 响应 PID”。

（1） 传送类型

- 通过PIPEPCFG寄存器的TYPE位设定各管道的传送类型。各管道能设定的传送类型如下所示。
- DCP：不用设定 （固定为控制传送）。
- 管道1、 2：请设定批量传送或者同步传送。
- 管道3～5：请设定批量传送。
- 管道6、 7：请设定中断传送。

（2） 端点编号

通过 PIPECFG 寄存器的 EPNUM 位设定各管道的端点编号。 DCP 固定为端点 0。其他管道，能设定为端点 1 ～端点 15。

- DCP：不用设定 （固定为端点0）。
- 管道1～7：请选择并设定 （1～15）。
  但是不能使DIR位和EPNUM位的组合重复。

（3） Max 包大小设定

能通过 DCPMAXP 寄存器及 PIPEMAXP 寄存器的 MXPS 位设定各管道的 Max 包大小。DCP 及管道 1 ～ 5 能设定为 USB 规格定义的全部 Max 包大小。管道 6、 7 的 Max 包大小上限为 64 字节。请在开始传送之前（PID=BUF）设定 Max 包大小。

- DCP：高速运行时请设定64。
- DCP：全速运行时，请从8、 16、 32、 64选择并设定。
- 管道1～5：全速批量传送时，请从0或者512中选择并设定。
- 管道1～5：全速批传送时，请从0、 8、 16、 32、 64选择设定。
- 管道1、 2：高速同步传送时，请设定1～1024的值。
- 管道1、 2：全速同步传送时，请设定1～1023的值。
- 管道6、 7：请设定1～64的值。

不支持中断传送及同步传送的 High Bandwidth。

（4） 响应 PID

能通过 DCPCTR 寄存器及 PIPEnCTR 寄存器的 PID 位设定各管道的响应 PID。
各设定中本模块的运行如下所示。

1. 选择主机控制器功能时的响应PID设定
响应PID指定处理的执行情况。
- NAK设定：管道禁止状态。不执行处理。
- BUF设定：根据缓冲存储器的状况，执行处理。
OUT方向时，如果缓冲存储器中有发送数据，就发送OUT权标。
IN方向时，如果缓冲存储器有空间能进行接收，就发送IN权标。
- STALL设定：管道禁止状态。不执行处理。

【注】 用 SUREQ 位设定 DCP 的设置处理。

2. 选择功能控制器功能时的响应PID设定
响应PID用于指定对来自主机处理的响应。
- NAK设定：总是对发生的处理进行NAK应答。
- BUF设定：根据缓冲存储器的状况应答处理。
- STALL设定：总是对发生的处理进行STALL应答。

【注】 对于设置处理，与 PID 的设定无关，总是进行 ACK 应答，并将 USB 请求保存在寄存器。

根据处理结果，本模块有可能对 PID 位进行写入。本模块对 PID 位进行写入的情况如下。

1. 选择主机控制器功能时，如果硬件设定响应PID
- NAK设定：以下情况为PID=NAK，并自动停止权标的发送。
在同步以外的传送中，对发送的权标无响应时
对于发送的权标接收到破损包时
在控制读传送的数据阶段中，接收到短包时
批量传送中，将PIPECFG寄存器的SHTNAK位置1时，接收到短包时
批量传送中将SHTNAK位置1，结束处理计数器时
- BUF设定：没有BUF写入 （通过本模块）。
- STALL设定：以下情况为PID=STALL，自动停止权标的发送。
对于发送的权标接收到STALL时
接收的数据包超过Max包大小时

2. 选择功能控制器功能时，如果硬件设定响应PID
- NAK设定：以下情况为PID=NAK，对处理总是进行NAK应答。
正常接收SETUP权标时 （仅DCP）
批量传送中将PIPECFG寄存器的SHTNAK位置1时，如果处理计数器结束，或者接收到短包
- BUF设定：没有BUF写入 （通过本模块）。
- STALL设定：以下情况为PID=STALL，对处理总是进行STALL应答。
在接收数据包检测出Max包大小溢出错误时
检测出控制传送顺序错误时

（5） USB 通信允许 （PID=BUF）的状态下为设定禁止的寄存器

- CFIFOSEL 寄存器的 ISEL 位 （只适用选择 DCP 时）
- CFIFOSIE 寄存器的 TGL 位、 SCLR 位
- DnFIFOSEL 寄存器的 DCLRM 位、 TRENB 位、 TRCLR 位、 DEZPM 位
- DnFIFOTRN 寄存器的 TRNCNT 位
- DCPCFG 寄存器的各位
- DCPMAXP 寄存器的各位
- DCPCTR 寄存器的各位 （除 CCPL 位）
- PIPECFG 寄存器的各位
- PIPEBUF 寄存器的各位
- PIPEMAXP 寄存器的各位
- PIPEPERI 寄存器的各位
- PIPEnCTR 寄存器的各位

（6） 数据 PID 顺序位

在控制传送的数据阶段、批量传送、中断传送中正常进行数据传送时，数据 PID 的顺序位通过本模块自动交替运行。其次被发送的数据 PID 的顺序位，能通过 DCPCTR 寄存器及 PIPEnCTR 寄存器的 SQMON 位确认。发送数据时，在 ACK 信号交换接收时序切换顺序位，接收数据时，在 ACK 信号交换发送时序切换顺序位。同时，能通过 DCPCTR 寄存器及 PIPEnCTR 寄存器的 SQCLR 位、 SQSET 位改变数据 PID 顺序位。

如果选择功能控制器功能并进行控制传送，阶段转移时，本模块自动设定顺序位。设置阶段结束时为 DATA0 应答，状态阶段为 DATA1 应答。因此，没有必要通过软件设定。如果选择主机控制器功能并进行控制传送，阶段转移时需要通过软件设定顺序位。

请注意不管是选择主机控制器还是功能控制器的功能，发送或者接收 ClearFeature 请求等时，都必须通过软件设定数据 PID 顺序位。

另外，同步传送设定管道不能通过 SQSET 位进行顺序位操作。

（7） 响应 PID=NAK 功能

通过将 PIPECFG 寄存器的 SHTNAK 位置 1，本模块能在接收到传输的最后一个 （模块根据短包接收或者处理计数器自动识别）数据包时禁止管道运行 （响应 PID=NAK）。

通过使用该功能，在缓冲存储器为双缓冲时，能按传输单位接收数据包。同时，如果禁止管道运行，就需要再次通过软件进行管道允许 （响应 PID=BUF）设定。

另外，只能在批量传送时运行本功能。

（8） 自动响应模式

在批量传送的管道（管道 1 ～ 5）中，如果将 PIPEnCTR 寄存器的 ATREPM 位置 1，就成为自动响应模式。OUT 传送时 （DIR=0）为 OUT-NAK 模式， IN 传送时 （DIR=1）为 Null 自动响应模式。

- OUT-NAK模式

在批量 OUT 传送的管道中，如果将 ATREPM 位置 1，就对 OUT 权标或者 PING 权标进行 NAK 应答并输出 NRDY 中断。要从普通模式设定为 OUT-NAK 模式，请在管道运行禁止状态（响应 PID=NAK）下设定 OUT-NAK 模式，并允许管道运行（响应 PID=BUF）。允许管道运行后，OUT-NAK 模式有效。但是在管道运行禁止前接收 OUT 权标时，就正常接收该权标的数据，对主机进行 ACK 应答。

要从 OUT-NAK 模式转移至普通模式，请在管道运行禁止状态（响应 PID=NAK）下解除 OUT-NAK 模式，并允许管道运行（响应 PID=BUF）。在普通模式能接收 OUT 数据，如果缓冲器能接收数据就返回 ACK 以响应 PING 权标。

- Null自动响应模式

在批量 IN 传送的管道中，如果将 ATREPM 位置 1，就连续发送 Zero-Length 包。

要从普通模式设定为 Null 自动响应模式，请在管道运行禁止状态 （响应 PID=NAK）下设定 Null 自动响应模式，并允许管道运行（响应 PID=BUF）。允许管道运行后，Null 自动响应模式有效。但在设定 Null 自动响应模式时，缓冲器内必须为空。请确认 INBUFM 位 =0。在 INBUFM 位为 1 时，因为缓冲器内有数据，需要通过 ACLRM 位清空缓冲器。同时在设定 Null 自动响应模式时，不能从 FIFO 端口写数据。

要从 Null 自动响应模式转换至普通模式，请在管道运行禁止状态（响应 PID=NAK）下等待 Zero-Length 包发送期间 （全速时：10µs，高速时：3µs）后，解除 Null 自动响应模式。在普通模式，能从 FIFO 端口写数据，并可通过允许管道运行 （响应 PID=BUF）向主机发送数据包。

25.4.4　缓冲存储器

（1） 缓冲存储器分配

本模块的缓冲存储器映射例如图 25.9 所示。缓冲存储器是 CPU 与本模块共同使用的区域。缓冲存储器的存取权有在系统 （CPU 侧）和在本模块 （SIE 侧）的情况。

缓冲存储器对每个管道设定独立区域。存储区通过块起始编号和块数 （通过 PIPEBUF 寄存器的 BUFNMB 位及 BUFSIZE 位指定）设定为 1 个块 64 字节。同时，对缓冲存储器的存取（数据读写）使用 3 个 FIFO 端口。分配给 FIFO 端口的管道通过 C/DnFIFOSEL 寄存器的 CURPIPE 位指定管道编号。

能通过 DCPCTR 寄存器及 PIPEnCTR 寄存器的 BSTS 位、INBUFM 位确认各管道的缓冲器状态。同时，能通过 C/DnFIFOCTR 寄存器的 FRDY 位确认 FIFO 端口的存取权。

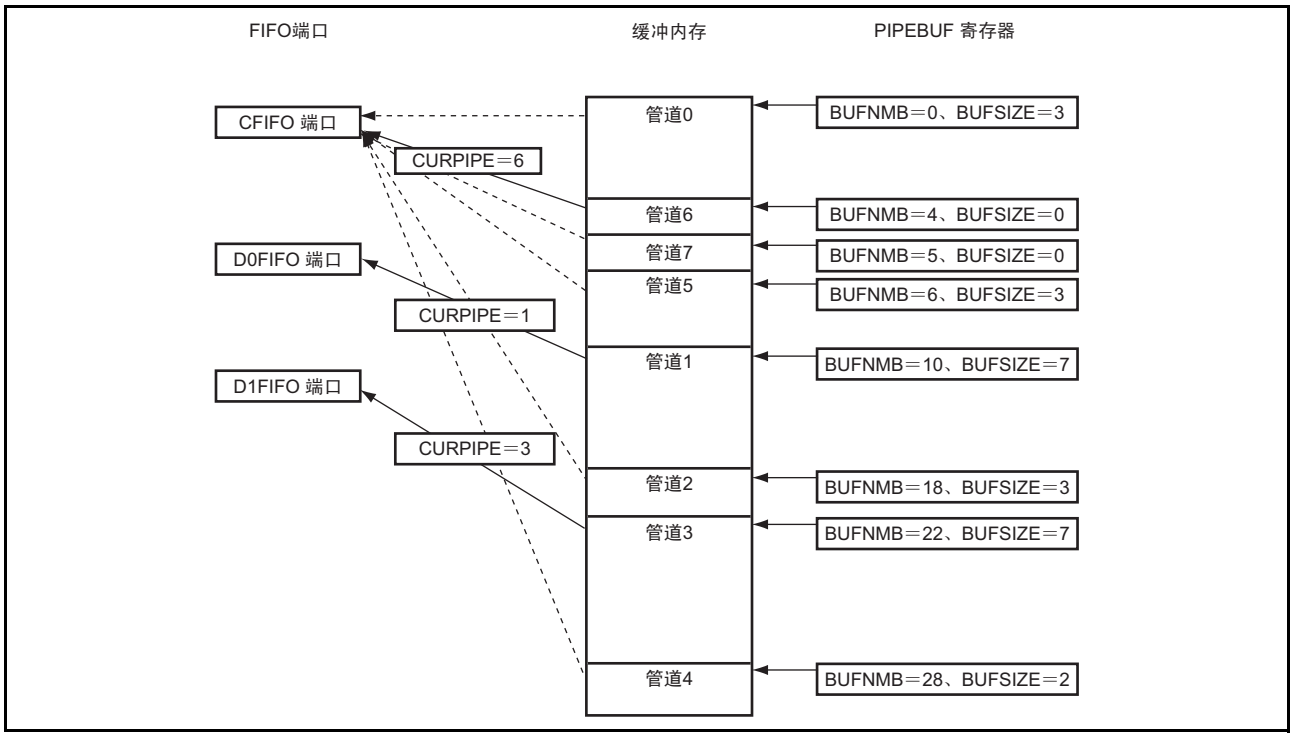


图 25.9　缓冲存储器映射例

（a） 缓冲器状态

本模块的缓冲器状态表如表 25.15 及表 25.16 所示。能通过 DCPCTR.BSTS 位及 PIPEnCTR.INBUFM 位确认缓冲存储器状态。通过 PIPEnCFG 寄存器的 DIR 位或者 CFIFOSEL 寄存器的 ISEL 位（选择 DCP 时）指定缓冲存储器的存取方向。

另外， INBUFM 位仅对处于发送方向的管道 1 ～ 5 有效。

设定发送侧的传送管道为双缓冲时， BSTS 位用于判断 CPU 侧的缓冲器状态：INBUFM 位用于判断 SIE 侧的缓冲器状态。

因 CPU（DMAC）对 FIFO 端口的写入延迟而不能通过 BEMP 中断辨别缓冲器是否为空时，能通过 INBUFM 位确认发送结束。

表 25.15　通过 BSTS 位判断的缓冲器状态表

ISEL 或者 DIR	BSTS	缓冲存储器的状态
0 （接收方向）	0	无接收数据，或者接收中 禁止从 FIFO 端口读取数据
0 （接收方向）	1	有接收数据，或者接收 Zero-Length 包 允许从 FIFO 端口读取数据 但由于在接收 Zero-Length 包时不能读取，因此需要缓冲器清除
1 （发送方向）	0	未完成发送 禁止对 FIFO 端口写入数据
1 （发送方向）	1	发送结束 能写入 CPU

表 25.16　通过 INBUFM 位判断的缓冲器状态表

DIR	INBUFM	缓冲存储器的状态
0 （接收方向）	无效	无效
1 （发送方向）	0	发送完可发送的数据 无可发送的数据
1 （发送方向）	1	从 FIFO 端口写入可发送的数据 有可发送的数据

（b） 缓冲器清除

通过本模块清除缓冲存储器的一览表如**表 25.17** 所示。能通过 BCLR、 SCLR、 DCLRM、 ACLRM 位清除缓冲存储器。

表 25.17　缓冲器清除一览表

位名	BCLR	SCLR	DCLRM	ACLRM
寄存器	CFIFOCTR 寄存器 DnFIFOCTR 寄存器	CFIFOSIE 寄存器	DnFIFOSEL 寄存器	PIPEnCTR 寄存器
功能	清除CPU侧缓冲存储器。	清除SIE侧缓冲存储器。	是读出指定管道的数据后，自动清除缓冲存储器的模式。	是废弃全部接收包的自动缓冲器清除模式。
清除方法	通过写 1 清除	通过写 1 清除	1：模式有效 0：模式无效	1：模式有效 0：模式无效

（c） 缓冲区

本模块的缓冲存储器映射如**表 25.18** 所示。在缓冲存储器中，有预先分配给管道的专用固定区域及用户可设定的用户区域。

DCP 用缓冲器是控制读传送及控制写传送使用的同一专用固定区域。

预先分配管道 6、 7 区域，但不使用管道 6、 7 时，可作为用户区域分配给管道 1 ～ 5 使用。

各管道中的设定区域不能重叠。请注意特别是设定双缓冲时，区域是设定值的二倍。

同时，不能通过小于 Max 包大小的设定值指定缓冲器大小。

表 25.18　缓冲存储器映射

缓冲存储器编号	缓冲器大小	管道设定	备注
H'0 ～ H'3	256 字节	DCP 专用固定区域	单缓冲器、能连续传送
H'4	64 字节	管道 6 用固定区域	单缓冲器
H'5	64 字节	管道 7 用固定区域	单缓冲器
H'6 ～ H'7F	最大 7808 字节	管道 1 ～ 5 用用户区域	能设定双缓冲器、能连续传送

（d） 自动缓冲器清除模式功能

本模块通过将 PIPEnCTR 寄存器的 ACLRM 位置 1，废弃接收的全部数据包。但如果接收了正常的数据包，就对主机控制器进行 ACK 响应。另外，只能在缓冲存储器读出方向设定本功能。

另外，能通过将 ACLRM 位置 1 后再置 0，来清除该管道的缓冲存储器，这与存取方向无关。

作为硬件的内部顺序执行时间，对 ACLRM 位写 1 与写 0 的间隔至少需要 100ns。

（e） 缓冲存储器规格 （单 / 双设定）

管道 1 ～ 5 能通过 PIPEnCFG 寄存器的 DBLB 位选择单缓冲器或者双缓冲器。双缓冲器是给同一管道分配 2 个存储区（通过 PIPEBUF 寄存器的 BUFSIZE 位指定）的功能。本模块的缓冲存储器设定例如图 25.10 所示。

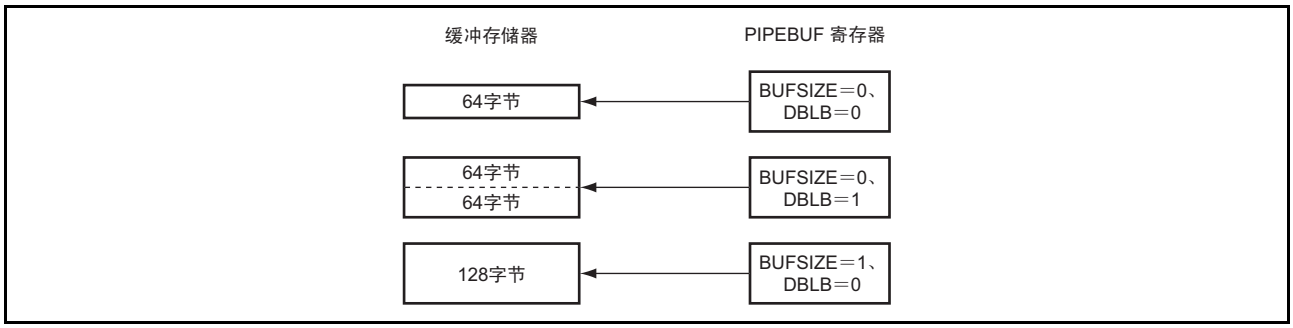


图 25.10　缓冲存储器设定例

（f） 缓冲存储器运行 （连续传送设定）

能通过 DCPCFG 寄存器及 PIPEnCFG 寄存器的 CNTMD 位选择连续传送模式或者非连续传送模式。该选择对管道 0 ～ 5 有效。

连续传送模式功能是连续发送 / 接收多个处理的功能。设定连续传送模式时，能在传送数据达到分配给各管道缓冲器大小前不对 CPU 产生中断。

连续发送模式中，按照 Max 包大小分割写入数据并发送。发送小于缓冲器大小的数据 （短包或者 Max 包大小的整数倍且小于缓冲器大小）时，写入发送数据后需要设定 BVAL=1。

连续接收模式中，接收最大相当于缓冲器大小的包、处理计数结束或者接收短包之前，不产生中断。

本模块的缓冲存储器运行例如图 25.11 所示。

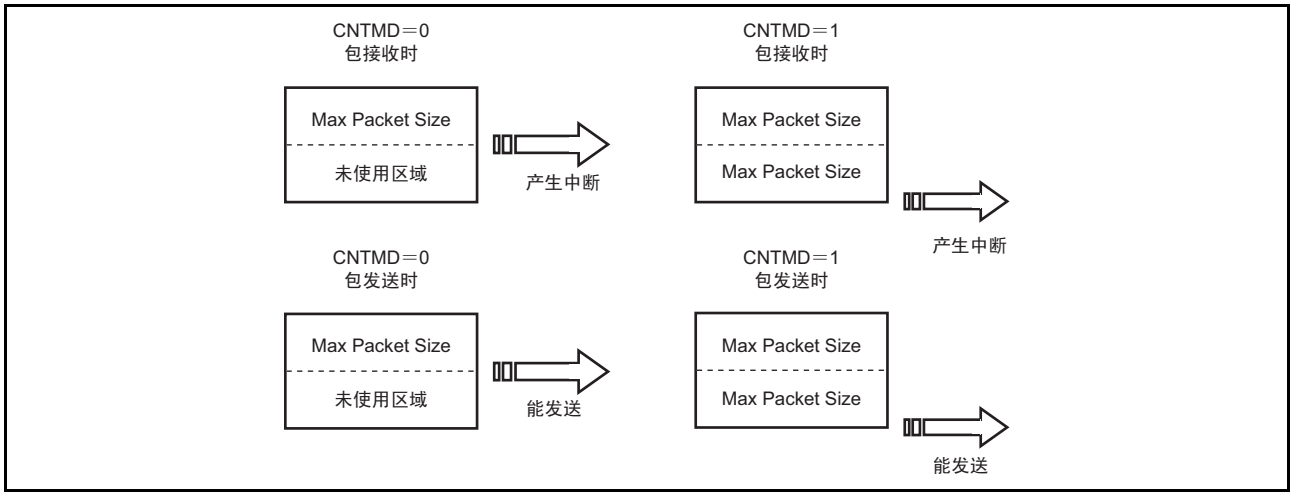


图 25.11　缓冲存储器运行例

（2） FIFO 端口的功能

本模块的 FIFO 端口功能设定表如表 25.19 所示。数据写存取时，如果写入达到缓冲器满 （非连续传送时为 Max 包大小），就自动成为可发送状态。要使小于缓冲器满（非连续传送时为 Max 包大小）的数据成为可发送状态，需要通过 C/DnFIFOCTR 寄存器的 BVAL 位设定写入结束。同时， Zero-Length 包的发送，需要通过相同寄存器的 BCLR 位清除缓冲器后，再通过 BVAL 位设定写入结束。

读存取时，如果读出全部数据，就自动成为可以接收新包的状态。但，接收 Zero-Length 包时（DTLN=0），因为不能读出数据，所以需要通过相同寄存器的 BCLR 位清除缓冲器。通过 C/DnFIFOCTR 寄存器的 DTLN 位确认接收数据长度。

表 25.19　FIFO 端口功能设定表

寄存器名	位名	功能	备注
C/DnFIFOSEL	REW	缓冲存储器回绕 （重读、重写）	
	DCLRM	读出指定管道的接收数据后自动清除	DnFIFO 专用
	DREQE	允许 DMA 传送	DnFIFO 专用
	MBW	FIFO 端口存取位宽	
	TRENB	允许处理计数器运行	DnFIFO 专用
	TRCLR	清除当前处理次数	DnFIFO 专用
	DEZPM	Zero-Length 包补充模式	DMA 传送专用
	ISEL	FIFO 端口存取方向	DCP 专用
C/DnFIFOCTR	BVAL	缓冲存储器写入结束	
	BCLR*	清除 CPU 侧缓冲存储器	
	DTLN	确认接收数据长度	
DnFIFOTRN	TRNCNT	设定接收处理计数	DnFIFO 专用
CFIFOSIE （除 DCP）	TGL	CPU/SIE 缓冲器交替	CFIFO 专用
	SCLR	清除 SIE 侧缓冲存储器	CFIFO 专用

【注】 *　设定 CFIFOSEL.CURPIPE=DCP 时，如果设定 CFIFOCTR.BCLR=1，也能清除 SIE 侧缓冲存储器。

（a） FIFO 端口选择

各 FIFO 端口能选择的管道表如表 25.20 所示。通过 C/DnFIFOSEL 寄存器的 CURPIPE 位选择存取的管道。选择管道后，请确认 FRDY=1，然后存取 FIFO 端口。

同时，请通过 MBW 位选择存取的总线宽度。缓冲存储器存取方向取决于 PIPEnCFG 寄存器的 DIR 位。但只有 DCP 是通过 ISEL 位决定的。

表 25.20　不同管道的 FIFO 端口存取表

管道	存取方法	可使用的端口
DCP	CPU 存取	CFIFO 端口寄存器
管道 1 ～ 7	CPU 存取	CFIFO 端口寄存器 D0FIFO/D1FIFO 端口寄存器
	DMA 存取	D0FIFO/D1FIFO 端口寄存器

（b） REW 位

能在暂时中断当前存取的管道存取后，存取其他的管道，然后再继续进行当前的管道处理。这样的处理需要使用 C/DnFIFOSEL 寄存器的 REW 位。

如果在设定 C/DnFIFOSEL 寄存器的 CURPIPE 位的同时将 REW 位置 1 并进行管道选择，就能复位缓冲存储器的读 / 写指针，并从最初字节进行读 / 写。同时，如果将 REW 位置 0 并进行管道选择，就能够不复位缓冲存储器的读 / 写指针，而接着上次的选择读写数据。

要存取 FIFO 端口，需要在选择管道后确认 FRDY=1。

（c） SIE 侧缓冲存储器的读出 （CFIFO 端口读出方向）

即使在 FRDY=0 状态下不能从缓冲存储器读出数据，本模块也能够通过确认 CFIFOSIE 寄存器的 SBUSY 位，并将 TGL 位置 1，来读存取 SIE 侧的数据。请在设定 PID=NAK 并确认 SBUSY=0 后写 TGL=1。模块就能从 CFIFO 寄存器读出数据。另外，只能在缓冲存储器读出方向使用本功能。同时，通过 TGL 位操作产生 BRDY 中断。

请不要在以下状态时对 TGL 位写 1：

- 选择DCP时
- 读取缓冲存储器时
- 缓冲存储器写入方向的管道

（d） 清除 SIE 侧缓冲存储器 （CFIFO 端口写入方向）

本模块能通过确认 CFIFOSIE 寄存器的 SBUSY 位，并将 SCLR 位置 1，取消准备发送的数据。

请在设定 PID=NAK，并确认 SBUSY=0 后，写 SCLR=1。模块就能从 CFIFO 寄存器写入新数据。另外，只能在缓冲存储器写入方向使用本功能。同时，通过 SCLR 位操作产生 BRDY 中断。

请不要在下列状态中对 SCLR 位写 1：

- 选择DCP时
- 写入缓冲存储器时
- 缓冲存储器读出方向的管道

（e） 处理计数器 （D0FIFO/D1FIFO 端口读出方向）

在数据包接收方向，结束指定次数的处理时，本模块能识别传送结束。在 D0FIFO/D1FIFO 端口选择管道设定为从缓冲存储器读出数据的方向时，运行处理计数器功能。处理计数器有指定处理次数的 TRNCNT 寄存器和在内部对处理进行计数的当前计数器，如果当前计数器与指定次数匹配，缓冲存储器就成为可读状态。能通过 TRCLR 位初始化处理计数器功能的当前计数器，并重新对处理进行计数。TRENB 位的设定决定 TRNCNT 寄存器读出的信息。

- TRENB=0：能读出设定的处理计数器值。
- TRENB=1：能读出当前计数器 （在内部进行计数）的值。

CURPIPE 位的改变条件如下：

- 指定管道的处理结束前，请不要改变CURPIPE位。
- 如果当前计数器未被清除，就不能改变CURPIPE位。

TRCLR 位的操作条件如下：

- 在计数处理的过程中，且PID=BUF时，不能清除当前计数器。
- 在缓冲器内残留有数据的状态下，不能清除当前计数器。

（f） FIFO 端口存取等待指定

关于存取本模块内的 FIFO 端口，有如下规定：

- 传送速度最大不超过48MB/sec

因此，为了消除总线时钟的频率限制，本模块有能通过存取等待指定 （FWAIT）位控制存取周期的功能。能给各 FIFO 端口设定 FWAIT 位，并能根据 CPU 速度及传送对象的存取周期等进行有效设定。

FWAIT 位的设定例：

- 条件

  存取方向　　：对FIFO的写入

  总线时钟频率　　：66MHz

  MBW位设定值　　：10 （32位幅）

  存取类型　　：传送数据从内置存储器 （源侧）读出数据后，写入 FIFO 端口。此时，源侧的存取需要2个时钟周期。

- 计算例

  (2+(FWAIT+2))×1/66MHz ≥ 1/48MHz×4 （32位）

  FWAIT=2 （4个时钟周期）

（g） 对 FIFO 端口的尾数数据存取方法

从 FIFO 端口读取数据时，当读出数据少于 FIFO 端口选择寄存器的 MBW 位指定的位宽度时，就请以 MBW 位指定的位宽度读出数据，并通过软件删除不需要的数据。

对 FIFO 端口写入数据时，当写入数据少于 FIFO 端口选择寄存器的 MBW 位指定的位宽度时，请按下面的例子存取。例：将 FIFO 端口宽度设定为 32 位 （MBW=10）时，写入 24 位数据。

- 写入尾数数据例1：16位宽写1次，然后8位宽写1次

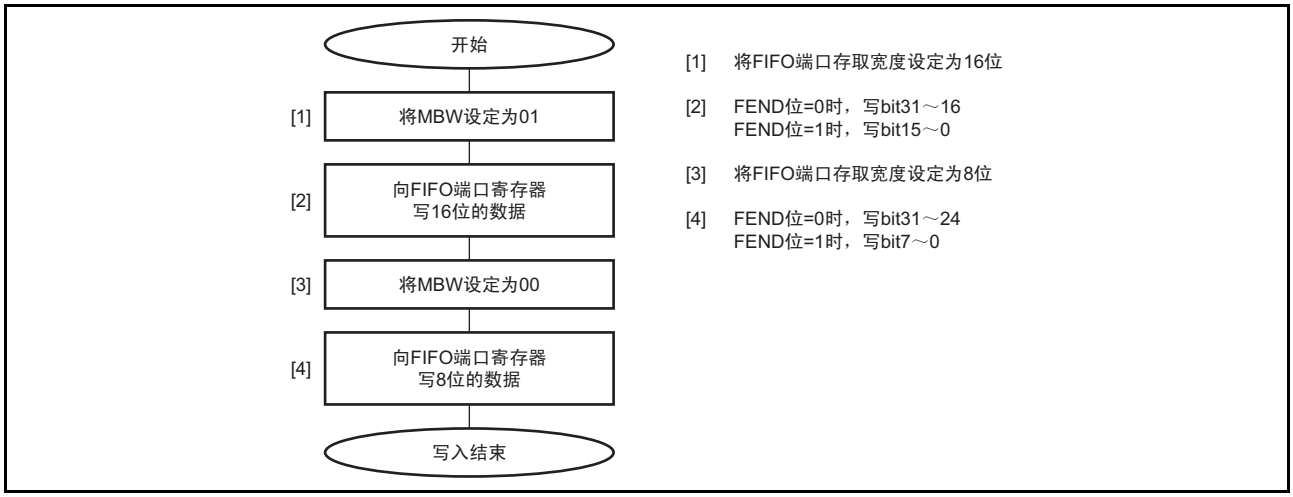


图 25.12　对 FIFO 端口写入尾数数据例 1

- 写入尾数数据例2：8位宽写3次

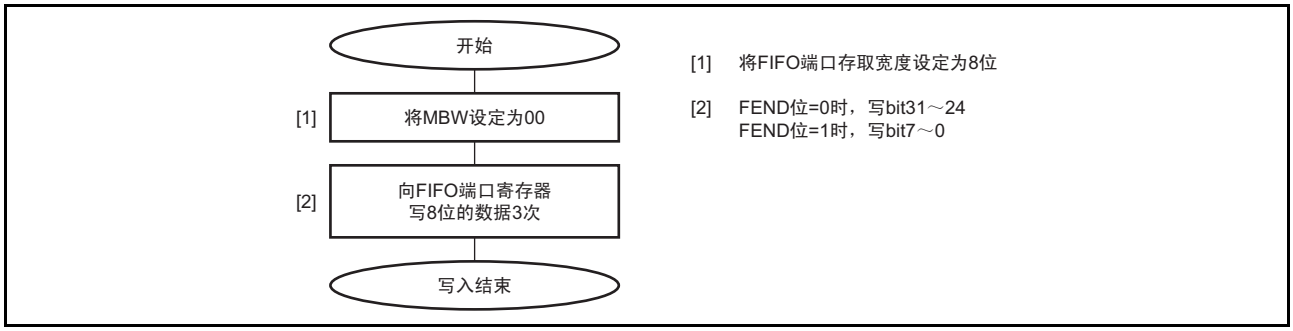


图 25.13　对 FIFO 端口写入尾数数据例 2

（3） DMA 传送 （D0FIFO/D1FIFO 端口）

（a） DMA 传送概要

对于管道 1 ～ 7，能通过 DMAC 存取 FIFO 端口。能存取设定为 DMA 传送的管道的缓冲器时，输出 DMA 传送要求。

请通过 DnFIFOSEL 寄存器的 MBW 位选择对 FIFO 端口的传送单位，通过 CURPIPE 位选择 DMA 传送的管道。另外，在 DMA 传送中请不要变更选择的管道。

（b） DMA 传送结束自动识别

通过控制 DMA 传送结束信号输入，本模块能结束通过 DMA 传送进行的 FIFO 数据写入。如果按照直接存储器存取控制器 （DMAC）的 DMA 传输计数器 （DMATCR）设定的次数进行 DMA 传送，就从 DMAC 输出 DMA 传送结束信号。如果采样 DMA 传送结束信号，就使缓冲存储器进入可发送状态 （与设定 BVAL=1 的状态相同）。能通过 DnFBCFG.TENDE 位设定采样 / 不采样 DMA 传送结束信号。

（c） Zero-Length 包补充模式 （D0FIFO/D1FIFO 端口写入方向）

通过将 DnFIFOSEL 寄存器的 DEZPM 位设定为 1，如果在采样 DMA 传送结束信号时写入到缓冲存储器的数据的字节数是 Max 包大小的整数倍，本模块就能在发送全数据后，补充发送 1 个 Zero-Length 包。只能在缓冲存储器设定为写入方向时 （对 CURPIPE 位设定发送方向的管道时）设定本功能。

（d） DnFIFO 自动清除模式 （D0FIFO/D1FIFO 端口读出方向）

通过将 DnFIFOSEL 寄存器的 DCLRM 位设定为 1，在结束缓冲存储器的数据读出时，本模块自动清除相应管道的缓冲存储器。

表 25.21 所示的是各设定中包接收和通过软件进行的缓冲存储器清除处理之间的关系。如**表 25.21** 所示，根据 BFRE 位的设定值，缓冲器清除条件有所不同，但即使在必须清除的状态中，通过使用 DCLRM 位，就不需要通过软件清除缓冲器，因此可进行不需软件介入的 DMA 传送。

另外，只能在缓冲存储器读出方向设定本功能。

表 25.21　包接收与通过软件进行缓冲存储器清除的关系表

寄存器设定 / 包接收时的缓冲器状态	DCLRM=0		DCLRM=1	
	BFRE=0	BFRE=1	BFRE=0	BFRE=1
缓冲器满	不用清除	不用清除	不用清除	不用清除
Zero-Length 包接收	需要清除	需要清除	不用清除	不用清除
通常的短包接收	不用清除	需要清除	不用清除	不用清除
处理计数结束	不用清除	需要清除	不用清除	不用清除

（e） BRDY 中断时序选择功能

通过设定 PIPECFG 寄存器的 BFRE 位，能在接收 Max 包大小的数据包时不产生 BRDY 中断。

使用 DMA 传送时，能通过该功能只在接收最终数据时产生中断。所谓最终数据，就是表示接收短包或者处理计数结束。设定 BFRE=1 时，在读出接收的数据后，产生 BRDY 中断。通过读出 DnFIFOCTR 寄存器的 DTLN 位，能在产生 BRDY 中断时确认最终接收的数据包的接收数据长度。

本模块的 BRDY 中断产生时序如表 25.22 所示。

表 25.22　BRDY 中断产生时序表

寄存器设定 / 包接收时的缓冲器状态	BFRE=0	BFRE–1
缓冲器满 （通常的包接收）	接收包时	不产生
接收 Zero-Length 包	接收包时	接收包时
接收通常的短包	接收包时	从缓冲存储器读完接收数据时
处理计数结束	接收包时	从缓冲存储器读完接收数据时

【注】 BFRE 位只在从缓冲存储器读出方向有效。写入方向时，请将 BFRE 位固定为 0。

（4） FIFO 端口可存取时序

（a） 管道切换时的 FIFO 端口可存取时序

转换 FIFO 端口指定的管道（改变 C/DnFIFOSEL 寄存器的 CURPIPE 位）时，确定 FRDY 位及 DTLN 位之前的时序图如图 25.14 所示。

改变 CURPIPE 位时，在写入 C/DnFIFOSEL 寄存器后，等待 450ns+ 总线时钟 8 个时钟周期后，进行 FIFO 端口的存取。

而且，改变 ISEL 位时， CFIFO 端口也为相同时序。

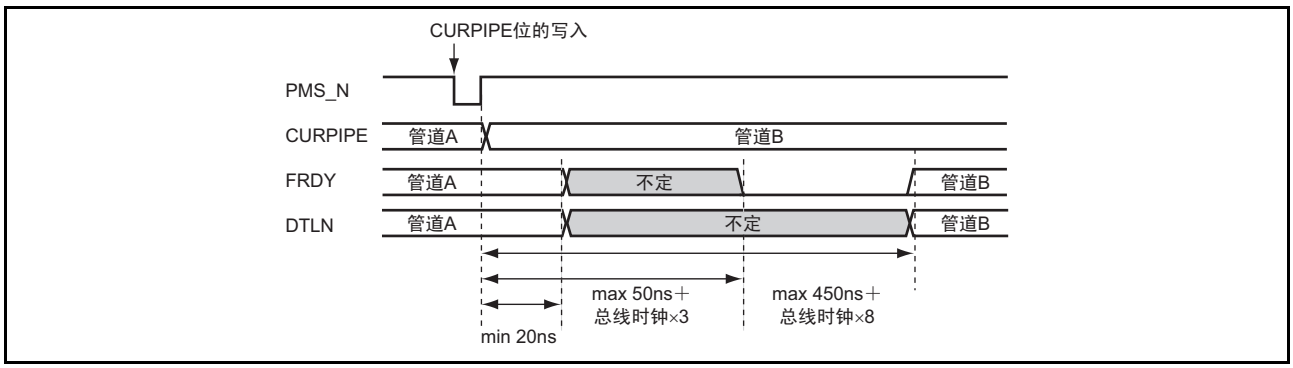


图 25.14　管道变更后的 FRDY、 DTLN 的确定时序

（b） 双缓冲时，读 / 写结束后的 FIFO 端口可存取时序

对于双缓冲器的管道，缓冲器读 / 写结束后，另一个缓冲器变为可存取状态前的时序如图 25.15 所示。

使用双缓冲器时，请在进行交替前的存取之后等待 300ns ＋总线时钟 6 个时钟周期后，进行 FIFO 端口的存取。

而且在使用 IN 方向的管道发送短包 （通过设定 BVAL=1）时也为相同时序。

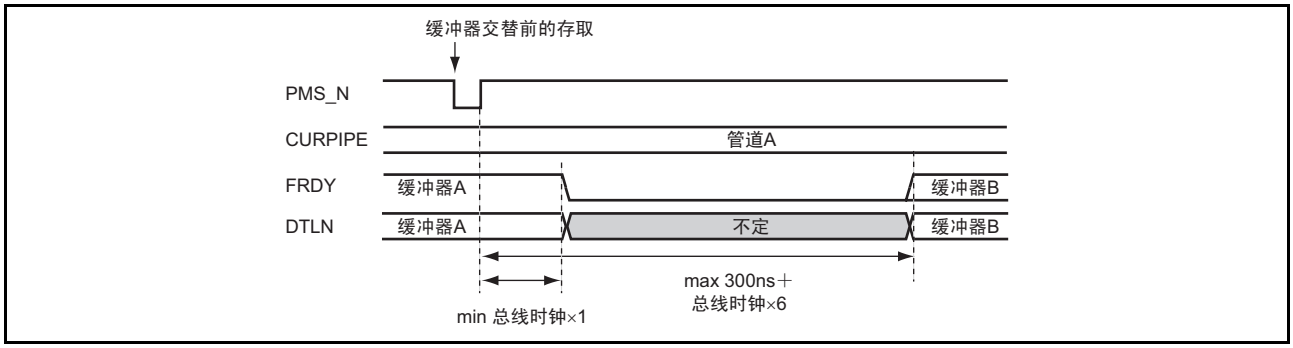


图 25.15　双缓冲器时，读 / 写结束后的 FRDY、 DTLN 的确定时序

25.4.5 控制传送 （DCP）

控制传送的数据阶段的数据传送，使用默认控制管道 （DCP）。在控制读及控制写共享的固定区域， DCP 的缓冲存储器为 256 字节单缓冲器。只能通过 CFIFO 端口存取缓冲存储器。

（1） 选择主机控制器功能时的控制传送

（a） 设置阶段

USBREQ 寄存器、 USBVAL 寄存器、 USBINDX 寄存器及 USBLENG 寄存器用于发送设置处理的 USB 请求。通过将设置包的数据写入寄存器，并给 DCPCTR 寄存器的 SUREQ 位写 1，来输出设定数据 （作为设置处理）。如果处理结束，就将 SUREQ 位清 0。SUREQ=1 时不运行上述 USB 请求寄存器。通过 DCPMAXP 寄存器的 DEVSEL 位指定设置处理的器件地址。

如果输出处理，就根据外围的响应产生中断请求 （INTSTS1 寄存器的 SIGN 位或 SACK 位）。通过该中断请求能确认设置处理结果。

与 DCPCTR 寄存器的 SQMON 位的内容无关，通常发送 DATA0 的数据包 （USB 请求）作为设置处理的数据包。

（b） 数据阶段

使用 DCP 缓冲存储器进行数据传送。

通过 CFIFOSEL 寄存器的 ISEL 位指定对 DCP 缓冲存储器的存取方向。而且，通过 DCPCFG 寄存器的 DIR 位指定传送方向。

数据阶段的第 1 个数据包需要将数据 PID 作为 DATA1 进行通信。通过 DCPCFG 寄存器的 SQSET 位将数据 PID 设置为 DATA1，并且将 PID 位设定为 BUF 来执行处理。通过 BRDY 中断及 BEMP 中断检测数据传送结束。

可通过指定连续传送模式进行多个包的数据传送。请注意：在接收方向设定连续传送时，如果缓冲器未满或不接收短包，就不会产生 BRDY 中断 （为 Max 包大小的整数倍，并且为 256 字节以下的情况）。

另外，控制写传送时，如果发送数据为 Max 包大小的整数倍，请通过软件进行控制使其在最后发送 Zero-Length 包。

如果在高速运行时数据传送为发送方向，就发送 PING 包。 PING 包的控制与批量传送相同。

（c） 状态阶段

该阶段为与数据阶段方向相反的 Zero-Length 包的数据传送。与数据阶段相同，是使用 DCP 缓冲存储器的数据传送。按与数据阶段相同的顺序执行处理。

数据阶段的数据包需要将数据 PID 作为 DATA1 进行通信。通过 DCPCFG 寄存器的 SQSET 位将数据 PID 设置为 DATA1。

另外， Zero-Length 包的接收，需要在产生 BRDY 中断后通过 CFIFOCTR 寄存器的 DTLN 位确认接收数据长度，并通过 BCLR 位进行缓冲存储器清除。

如果在高速运行时数据传送为发送方向，就发送 PING 包。 PING 包的控制与批量传送相同。

（2） 选择功能控制器功能时的控制传送

（a） 设置阶段

对于本模块来说属于正常的设置包，本模块必定对其进行 ACK 应答。设置阶段，本模块运行如下：

1. 如果接收新设置包，本模块就设置以下的位。
   - INTSTS0 寄存器的 VALID 位 =1
   - DCPCTR 寄存器的 PID 位 =NAK
   - DCPCTR 寄存器的 CCPL 位 =0

2. 如果在设置包之后接收数据包，本模块将 USB 请求的参数保存到 USBREQ 寄存器、 USBVAL 寄存器、USBINDX 寄存器及 USBLENG 寄存器。

必须在设定为 VALID=0 后，方可进行控制传送的响应处理。在 VALID=1 状态中不能设定 PID=BUF，并且不能结束数据阶段。

通过 VALID 位的功能，在控制传送过程中接收到新 USB 请求时，本模块可中断处理中的请求处理，响应最新请求。

另外，本模块自动辨别接收的 USB 请求的方向位 （bmRequestType 的 bit8）及请求数据长度 （wLength），并且识别控制读传送、控制写传送及控制写无数据传送，以及管理阶段转移。对于错误的顺序，产生控制传送阶段转移中断的顺序错误，并通知软件。关于本模块的阶段管理请参照图 25.7。

（b） 数据阶段

请使用 DCP 进行与接收到的 USB 请求相对应的数据传送。对 DCP 缓冲存储器进行存取前，请通过 CFIFOSEL 寄存器的 ISEL 位指定存取方向。

传送数据大于 DCP 缓冲存储器的大小时，在控制写传送中使用 BRDY 中断进行数据传送，在控制读传送中使用 BEMP 中断进行数据传送。

高速运行时的控制写传送中，根据缓冲存储器的状态进行 NYET 信号交换响应。关于 NYET 信号交换请参照 “25.4.6 （2） 选择功能控制器功能时的 NYET 信号交换控制”。

（c） 状态阶段

在 DCPCTR 寄存器的 PID 位为 PID=BUF 的状态中，通过将 CCPL 位置 1 来结束控制传送。

在进行上述设定后，按照设置阶段确定的数据传送方向，本模块自动执行状态阶段。具体内容如下所示：

- 控制读传送时
  本模块进行 Zero-Length 包的发送，从 USB 主机接收 ACK 响应。
- 控制写传送、无数据控制传送时
  从 USB 主机接收 Zero-Length 包，并且发送 ACK 响应。

（d） 控制传送自动响应功能

- 本模块自动应答正常的 SET_ADDRESS 请求。 SET_ADDRESS 请求有下列错误时，需要软件的响应。
- 控制读传送以外时：　bmRequestType ≠ H'00
- 请求错误时：　wIndex ≠ H'00
- 无数据控制传送以外时：　wLength ≠ H'00
- 请求错误时：　wValue ＞ H'7F
- 器件状态错误的控制传送：　DVSQ=011 （Configured）

SET_ADDRESS 以外的所有请求，需要相应的软件响应。

25.4.6 批量传送 （管道 1 ～ 5）

批量传送能选择缓冲存储器的使用方法（单 / 双缓冲器设定或连续 / 非连续传送模式设定）。缓冲存储器大小最大能设定到 2K 字节。本模块管理缓冲存储器的状态，并对 PING 包 /NYET 信号交换自动应答。同时在设定 MXPS=0 时，其中断规格与其他管道不同。详细内容请参照 “25.4.3 （3） Max 包大小设定”。

（1） 选择主机控制器功能时的 PING 包控制

通过本模块自动发送出 OUT 方向的 PING 包。

如下所示，初始状态为 PING 包发送的状态下，通过接收 ACK 信号交换发送 OUT 包。接收 NAK 或者 NYET 时，返回到 PING 发送状态。同时，该控制与控制传送的数据阶段、状态阶段相同。

1. 设定OUT数据发送模式
2. 发送PING包发送
3. 接收ACK信号交换
4. 发送OUT数据包
5. 接收ACK信号交换
   （重复4.和5.。）
6. 发送OUT数据包
7. 接收NAK/NYET信号交换
8. 发送PING包送

同时，本控制器返回到 PING 包发送模式的原因为上电复位、软件复位、接收 NYET/NAK 信号交换、置位顺序交替位、清除 （SQSET、 SQCLR）及设定缓冲器清除 （ACLRM）。

（2） 选择功能控制器功能时的 NYET 信号交换控制

本模块的 NYET 信号交换响应表如表 25.23 所示。本模块的 NYET 响应，遵从以下条件。但在接收短包时，不进行 NYET 包响应，而为 ACK 响应。同时，控制写传送的数据阶段也相同。

表 25.23　NYET 信号交换响应表

DCPCTR.PID 位的设定值	缓冲存储器的状态	权标	响应	备注
NAK/STALL	—	SETUP	ACK	—
	—	IN/OUT/PING	NAK/STALL	—
BUF	—	SETUP	ACK	—
	RCV-BRDY1	OUT/PING	ACK	接收 OUT 权标时，接收数据包
	RCV-BRDY2	OUT	NYET	通知不能接收数据包
	RCV-BRDY2	OUT （Short）	ACK	通知不能接收数据包
	RCV-BRDY2	PING	ACK	通知能接收
	RCV-NRDY	OUT/PING	NAK	通知不能接收
	TRN-BRDY	IN	DATA0/1	发送数据包
	TRN-NRDY	IN	NAK	TRN-NRDY

【符号说明】

RCV-BRDY1：接收 OUT/PING 权标时，缓冲存储器中有 2 个包以上的空白区

RCV-BRDY2：接收 OUT 权标时，缓冲存储器中只有 1 个包的空白区

RCV-NRDY：接收 PING 权标时，缓冲存储器中没有空白区

TRN-BRDY：接收 IN 权标时，缓冲存储器中有发送数据

TRN-NRDY：接收 IN 权标时，缓冲存储器中没有发送数据

25.4.7 中断传送 （管道 6、 7）

选择功能控制器功能时，本模块按照主机控制器管理的周期进行中断传送。中断传送时，忽视 PING 包（无响应）。同时，不发送 NYET 信号交换，而进行 ACK、 NAK、 STALL 响应。

选择主机控制器功能时，能通过间隔计数器设定权标发送时序。即使是 OUT 方向的传送，也不发送 PING 权标而发送 OUT 权标。如果从外围接收了 NYET 信号交换，就作为 ACK 接收运行。

另外，本模块不支持中断传送的 High-Bandwidth 传送。

（1） 选择主机控制器功能时，中断传送时的间隔计数器

如果进行中断传送，就对 PIPEPERI 寄存器的 IITV 位设定处理的间隔。本控制器按照设定的间隔发送中断传送的权标。

（a） 计数器的初始化

本控制器初始化间隔计数器的条件如下：

- 上电复位：
  初始化IITV位。
- 软件复位
  初始化IITV位。
- 通过ACLRM初始化缓冲存储器
  不初始化IITV位但初始化计数值。通过将ACLRM位置0，从IITV的设定值开始计数。

另外，请注意在以下情况时不初始化间隔计数器。

- USB总线复位、 USB挂起
  不初始化IITV位。通过将UACT位置1，从进入USB总线复位或USB挂起状态前的值开始计数。

（b） 不能在权标的发生时序进行发送 / 接收时的运行

以下情况时，即使在权标的发生时序也不发生权标。此时，在下一个间隔尝试执行处理。

- 将PID设定为NAK或者STALL时
- IN方向 （接收）的传送时，在权标的发送时序缓冲存储器没有空白区域时
- OUT方向 （发送）的传送时，在权标的发送时序缓冲存储器没有发送数据时

25.4.8 同步传送 （管道 1、 2）

对同步传送，本模块具备以下功能：

- 同步传送的错误信息通知
- 间隔计数器 （IITV位指定）
- 同步IN传送数据设置控制 （IDLY功能）
- 同步IN传送缓冲器闪存功能 （IFIS位指定）

本模块不支持同步传送的 High-Bandwidth 传送。

【注】 使用同步 OUT 传送时，请参照 “25.5.1 使用同步 OUT 传送时的注意事项”。

（1） 同步传送的错误检测

为了使软件能够管理同步传送的错误发生，本模块有以下的错误信息检测功能。关于本模块检测错误的优先级和伴随错误检测产生的中断如表 25.24 及表 25.25 所示。

1. PID错误
   - 接收包的PID错误时

2. CRC错误、位填充错误
   - 接收包的CRC有错误或者位填充错误时

3. 超出Max包大小
   - 接收包的数据大小超过Max包大小的设定值

4. 上溢、下溢错误
   - 选择主机控制器功能时
     IN方向 （接收）的传送时，在权标的发送时序缓冲存储器没有空白区域时
     OUT方向 （发送）的传送时，在权标的发送时序缓冲存储器没有数据时
   - 选择功能控制器功能时
     IN方向 （发送）的传送中，接收IN权标时缓冲存储器没有数据时
     OUT方向 （接收）的传送中，接收了OUT权标但缓冲存储器没有空白区域时

5. 间隔错误
   选择功能控制器功能时，以下情况为间隔错误。
   - 同步IN传送中，在间隔帧不能接收IN权标时
   - 同步OUT传送中，在间隔帧以外接收OUT权标时

表 25.24　接收权标时的错误检测

检测的优先级	错误	产生的中断和状态
1	PID 错误	选择主机 / 功能控制器功能时，均不会产生中断 （作为破损包忽视）。
2	CRC 错误、位填充错误	选择主机 / 功能控制器功能时，均不会产生中断 （作为破损包忽视）。
3	上溢、下溢错误	选择主机 / 功能控制器功能时，均会产生 NRDY 中断，并将 OVRN 位置位。选择主控制器功能时，不发送权标。选择功能控制器功能时，对 IN 权标发送 Zero-Length 包。不对 OUT 权标接收数据包。
4	间隔错误	选择功能控制器功能时，产生 NRDY 中断。选择主机控制器功能时不产生。

表 25.25　接收数据包时的错误检测

检测的优先级	错误	产生的中断和状态
1	PID 错误	不产生中断 （作为破损包忽视）。
2	CRC 错误、位填充错误	选择主机 / 功能控制器功能时，均会产生 NRDY 中断，并设定 CRCE 位。
3	Max 包大小超出错误	选择主机/功能控制器功能时，均会产生BEMP中断，并将PID设定为STALL。

（2） DATA-PID

本模块不支持 High-Bandwidth 传送。选择功能控制器功能时，对接收到的 PID 做如下处理：

1. IN方向
   - DATA0：作为数据包的PID发送。
   - DATA1：不发送。
   - DATA2：不发送。
   - mDATA：不发送。

2. OUT方向 （全速运行时）
   - DATA0：作为数据包的PID正常接收。
   - DATA1：作为数据包的PID正常接收。
   - DATA2：忽视包。
   - mDATA：忽视包。

3. OUT方向 （高速运行时）
   - DATA0：作为数据包的PID正常接收。
   - DATA1：作为数据包的PID正常接收。
   - DATA2：作为数据包的PID正常接收。。
   - mDATA：作为数据包的PID正常接收。

（3） 间隔计数器

能通过 PIPEPERI 寄存器的 IITV 位设定同步传送的间隔。选择功能控制器功能时，间隔计数器能实现表 25.26 所示的功能。选择主机控制器功能时，生成权标的发送时序。选择主机控制器功能时，间隔计数器的运行与中断传送运行相同。。

表 25.26　选择功能控制器功能时的间隔计数器功能

传送方向	功能	检测出条件
IN	发送缓冲器闪存功能	同步 IN 传送中，在间隔帧不能正常接收 IN 权标。
OUT	通知未接收权标	同步 OUT 传送中，在间隔帧不能正常接收 OUT 权标。

因为间隔的计数是在接收 SOF 或者插值 SOF 时进行的，所以即使 SOF 损坏也能保持同步性。能设定的帧间隔为 2^{IITV} 帧或者 $2^{IITV}\mu$ 帧。

（a） 选择功能控制器功能时的计数器初始化

本模块在下列条件下初始化间隔计数器。

- 上电复位
  初始化IITV位。
- 软件复位
  初始化IITV位。
- USB总线复位
  不初始化IITV位但初始化计数值。
- 通过ACLRM初始化缓冲存储器
  不初始化IITV位但初始化计数值。通过将ACLRM位置0，从IITV的设定值开始计数。

初始化间隔计数器后，在正常传送包之后，在下列 1. 或者 2. 的条件下开始间隔计数。

1. PID=BUF的状态下，为了响应IN权标，在数据发送后接收SOF
2. PID=BUF的状态下，接收OUT权标的数据后，接收SOF

另外，在下列条件下不初始化间隔计数器。

1. 设定PID位为NAK或者STALL时
   不停止间隔定时器。尝试在下一个间隔中执行处理。
2. USB挂起
   不初始化IITV位。接收SOF时，从接收前的值开始计数。

（4） 选择功能控制器功能时的同步传送发送数据设置

选择功能控制器功能时，对于本模块的同步数据发送，在向缓冲存储器写入数据后，检测出 SOF 包的下一个帧能发送数据包。该功能称为同步传送发送数据设置功能。能通过该功能指定开始发送的帧。

缓冲存储器为双缓冲器时，即使两个缓冲器侧的写入都结束，成为可传送状态的缓冲存储器也只是首先结束写入的那一侧。因此在同一帧，即使接收到多个 IN 权标，能发送的缓冲存储器大小也只有 1 个包的大小。

接收 IN 权标时，如果缓冲存储器为可发送状态就传送数据并正常应答。可是，如果缓冲存储器为不能发送的状态，就发送 Zero-Length 包并产生下溢错误。

图 25.16 表示设定 IITV=0 （每帧）时本模块使用同步传送发送数据设置功能进行发送的例子。图中阴影 Null 表示 Zero-Length 包发送。

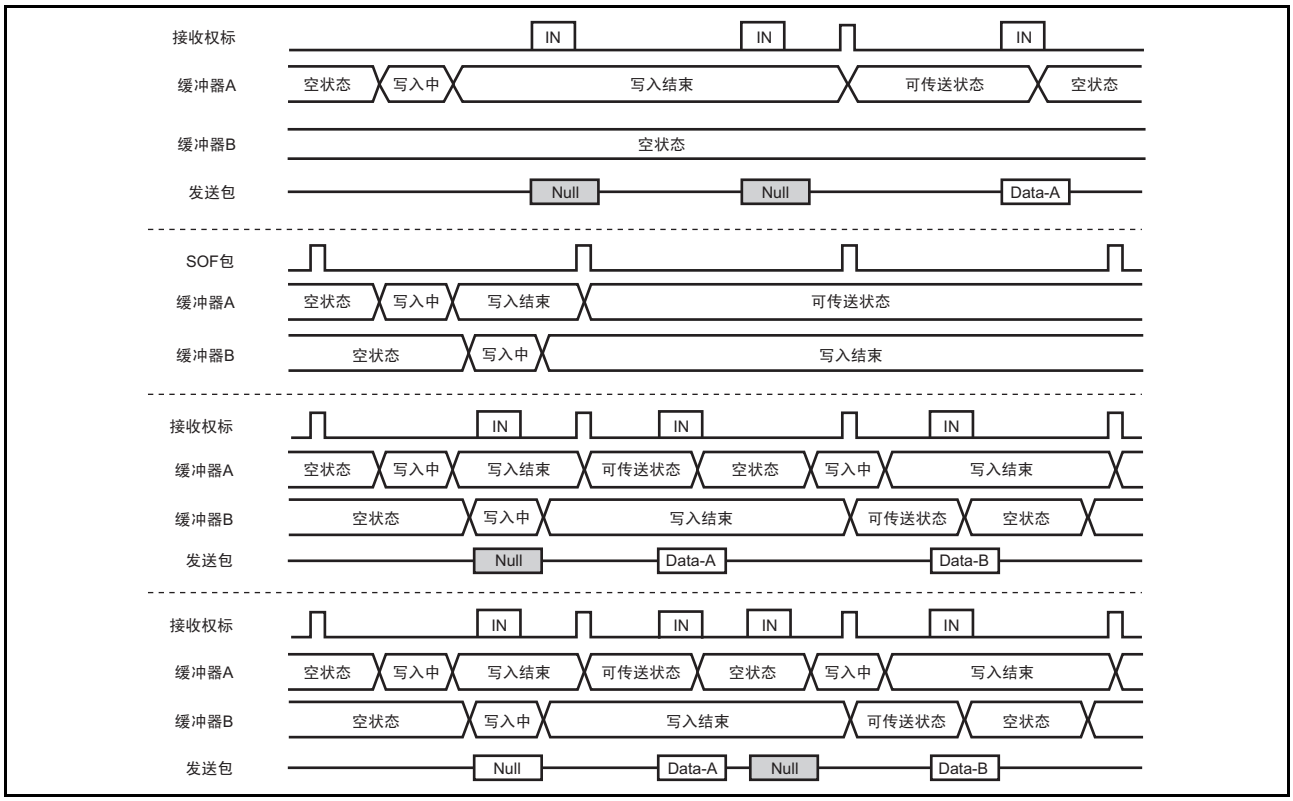


图 25.16　数据设置功能运行例

（5） 选择功能控制器功能时的同步传送发送缓冲器闪存

选择功能控制器功能时，本模块在同步数据发送的间隔帧不接收 IN 权标，接收下一个帧的 SOF 或者 μSOF 包时，作为 IN 权标破损处理，清除为可发送状态的缓冲器，并使该缓冲器为可写入状态。

此时，缓冲存储器为双缓冲器，两个缓冲器侧的写入都结束时，就将废弃的缓冲存储器看作在相同的间隔帧发送的数据，并使 SOF 或者 μSOF 包接收没有废弃的缓冲存储器成为可传送状态。

缓冲器闪存功能的运行开始时序因 IITV 位设定值而不同。

- IITV=0的情况
  从管道有效后的下一个帧开始运行缓冲器闪存。
- IITV=0以外的情况
  在第一个正常处理后运行缓冲器闪存。

本模块的缓冲器闪存功能运行例如**图 25.17** 所示。但在间隔帧之前接收到设定以外的权标时，按照数据设置状态，发送 Zero-Length 包作为写入数据的发送或者下溢错误。

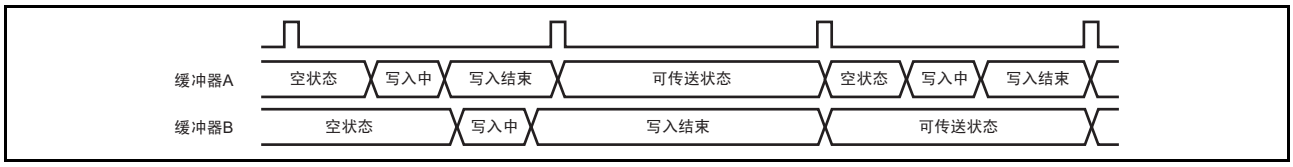


图 25.17　缓冲器闪存功能运行例子

本模块的间隔错误发生例如**图 25.18** 所示。间隔错误有如下 5 种。在图中的①时序发生间隔错误并运行缓冲器闪存功能。

在 IN 传送时，如果发生间隔错误，就运行缓冲器闪存功能，在 OUT 传送时，如果发生间隔错误，就产生 NRDY 中断。

请通过 OVRN 位判定 NRDY 中断 （接收包错误等）和溢出错误的区别。

根据缓冲存储器状态，响应图中阴影部分的权标。

1. IN方向
   - 如果是缓冲器能传送状态就传送数据并正常响应
   - 如果是缓冲器不能传送状态就发送Zero-Length包并发生下溢错误
2. OUT方向
   - 如果是缓冲器能接收状态就接收数据且正常响应
   - 如果是缓冲器不能接收状态就废弃数据并发生溢出错误

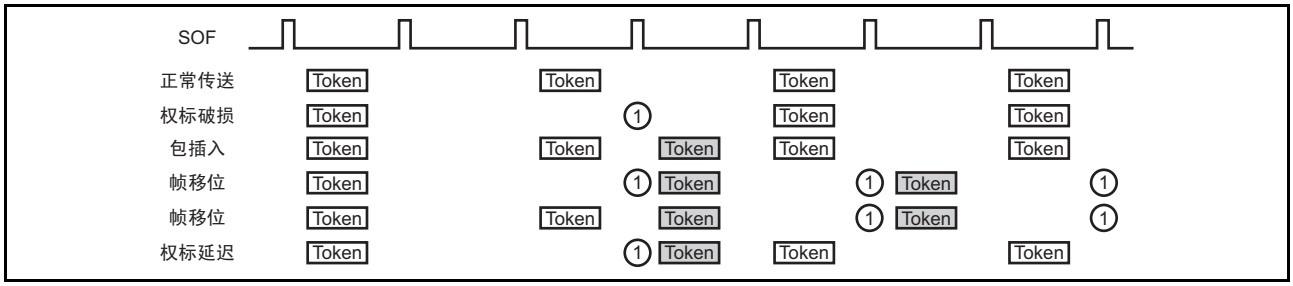


图 25.18　IITV=1 时的间隔错误发生例

25.4.9 SOF 插值功能

选择功能控制器功能时，由于 SOF 包破损或者丢失，在 1ms（全速运行时）或者 125μs（高速运行时）的间隔不能接收 SOF 包时，本模块插值 SOF。 SYSCFG.USBE=1 且接收 SOF 包时，开始运行 SOF 插值功能。同时，在以下条件下初始化插值功能。

- 上电复位
- 软件复位
- USB 总线复位
- 检测出挂起

同时， SOF 插值按照以下规格运行。

- 帧间隔 （125μs 或者 1ms）根据复位信号交换协议的结果。
- 接收 SOF 包前，不运行插值功能。
- 接收最初的 SOF 包后，以 USB 用时钟的 48MHz 计数 125μs 或者 1ms 并进行插值。
- 接收第 2 次以后的 SOF 包后，用上次的接收间隔进行插值。
- 挂起时及 USB 总线复位过程中不进行插值。
  （在高速运行时的挂起转移中，从接收到最终包开始 3ms 后继续插值）

本模块根据 SOF 包的接收情况运行下列功能。由于在 SOF 包丢失时进行 SOF 插值，因此能继续正常运行。

- 帧号及 μ 帧号的更新
- SOFR 中断时序及 μSOF 锁定
- 同步传送间隔计数

如果全速运行时 SOF 包丢失，就不更新 FRMNUM 寄存器的 FRNM 位。如果高速运行时 μSOF 包丢失，就更新 UFRMNUM 寄存器的 UFRNM 位。但如果 UFRNM=000 的 μSOF 包丢失，就不更新 FRNM 位。此时，即使继续正常接收 UFRNM=000 以外的 μSOF 包，也不更新 FRNM 位。

25.4.10　管道安排

（1） 处理产生条件

选择主机控制器功能时，在设定 UACT=1 后，本模块按表 25.27 所示的条件产生处理。

表 25.27　处理产生条件

处理	产生条件				
	DIR	PID	IITV0	缓冲器的状态	SUREQ
设置	—*1	—*1	—*1	—*1	设定 1
控制传送的数据阶段、状态阶段、批量传送	IN	BUF	无效	有接收区域	—*1
	OUT	BUF	无效	有发送数据	—*1
中断传送	IN	BUF	有效	有接收区域	—*1
	OUT	BUF	有效	有发送数据	—*1
同步传送	IN	BUF	有效	*2	—*1
	OUT	BUF	有效	*3	—*1

【注】 *1　表中的 “—”，表示与权标的发送无关的条件。有效表示在中断传送和同步传送中，仅在根据间隔计数器的传送帧才产生处理。无效表示条件的产生与间隔计数器无关。

*2　与有无接收区域无关，产生处理。但没有接收区域时废弃接收数据。

*3　与有无发送数据无关，产生处理。但没有发送数据时，发送 Zero-Length 包。

（2） 传送安排

此处对本模块帧内的传送安排进行了说明。发送 SOF 后，本模块按以下所示的顺序进行传送。

1. 执行周期性传送
   按管道1→管道2→管道6→管道7的顺序进行检索，如果有可产生同步传送或者中断传送处理的管道，就发送处理。

2. 控制传送的设置处理
   确认DCP，并能进行设置处理时，就发送。

3. 批量/控制传送数据阶段、状态阶段的执行
   按照DCP→管道1→管道2→管道3→管道4→管道5的顺序检索管道，如果有可产生批量/控制传送数据阶段、控制传送状态阶段处理的管道，就执行处理。
   产生处理时，来自外围的响应不管是ACK还是NAK，都将转移至下一个管道的处理。同时，如果帧内有进行传送的时间，就重复3.。

（3） 允许 USB 通信

通过将 DVSTCTR 寄存器的 UACT 位置 1，开始 SOF 或者 µSOF 的发送，并可产生处理。

设定 UACT 位为 0 时，停止 SOF 或者 µSOF 的发送并进入挂起状态。设定 UACT 位为 1→0 时，发送下一个 SOF 或者 µSOF 后停止。

25.5　使用时的注意事项

25.5.1　使用同步 OUT 传送时的注意事项

下列 “① and ②” 条件时，就请按照下表的管道设定，使用同步 （以下 ISO） -OUT 传送。

① 主机模式时使用

② 全速通信时

【注】 另外，使用主机模式的高速通信时以及使用功能模式时 （含全速通信时），本注意事项不适用。

	PIPE1	PIPE2	PIPE6
ISO-OUT 传送使用 PIPE1 时	ISO-OUT	未使用或者 BULK-IN/OUT	未使用或者 INT-IN/OUT
ISO-OUT 传送使用 PIPE2 时	未使用或者 ISO-IN 或者 BULK-IN/OUT	ISO-OUT	未使用

【注】 不能同时用PIPE1、PIPE2（2个管道）进行ISO-OUT传送。需要用2个管道进行ISO-OUT传送时，请使用高速通信。

25.5.2　USB 收发器的设定步骤

使用本模块时，需要事先设定内部 USB 收发器。设定方法如下所示。以下步骤的程序例子如图 25.19 所示。

（1） 给 USBAC 特性转换寄存器 （USBACSWR）的 UACS23 位写 1。

（2） 给 TEST 模式寄存器 （TESTMODE）的 HOSTPCC 位写 1。

为了该位不被错误改写，可以通过特殊的顺序进行写入，写入顺序如下所示。

1. 器件状态控制寄存器 （DVSTCTR）的UACKEY0、 UACKEY1位写1。
2. 给测试模式寄存器 （TESTMODE）的HOSTPCC位写1。
3. 给器件状态控制寄存器 （DVSTCTR）的UACKEY0、 UACKEY12位写0。

```
;初始化程序
;设定USBE=1
        MOVI20  #H'FFFC1C00, R0
        MOV.W   #H'0001, R1
        MOV.W   R1, @R0

;（1） 设定UACS23=1
        MOVI20  #H'FFFC1C84, R0
        MOV.L   #H'00800000, R1
        MOV.L   R1, @R0

;（2）设定 HOSTPCC=1
;1. UACKEY0、UACKEY1=1
        MOVI20  #H'FFFC1C04, R0
        MOV.W   #H'9000, R1
        MOV.W   R1, @R0
;2. HOSTPCC=1
        MOVI20  #H'FFFC1C06, R0
        MOV.W   #H'8000, R1
        MOV.W   R1, @R0
;3. UACKEY0、UACKEY1=0
        MOVI20  #H'FFFC1C04, R0
        MOV.W   #H'0000, R1
        MOV.W   R1, @R0
                .
                .
                .
                .
```

图 25.19　USB 收发器的设定步骤

25.5.3 清除中断源的时序

请在中断异常服务程序中清除中断源标志，从清除中断源标志到可以实际撤消对 CPU 的中断源需要一定的时间。因此，为了不错误地再次接收已被清除的中断源，需要在清除后读取 3 次中断源标志，然后再执行 RTE 指令。

第 26 章 LCD 控制器 （LCDC）

LCD 控制器 （LCDC）采用将显示用的图像保存到系统存储器的统一存储器体系结构。 LCDC 模块从系统存储器读出数据，使用调色板存储器决定颜色后，传送至 LCD 面板。可连接除单片机总线接口方式、 NTSC/PAL 方式、 LVDS 接口的液晶模块 * 以外的液晶模块。

【注】 * 通过连接 LVDS 转换 LSI，可连接 LVDS 接口。

26.1 特点

LCDC 有以下特点。

- 面板接口
  串行接口方式
  支持STN/Dual STN/TFT面板 （8/12/16/18位总线宽度）的数据格式*1
- 支持4/8/15/16bpp （位每象素）颜色模式
- 支持1/2/4/6bpp灰度模式
- 支持16×1～1024×1024的液晶面板*2
- 24位彩色调色板存储器 （24位中的16位为有效R:5/G:6/B:5）
- 通过 RGB 的 8 位、 24 位的空间调制 FRC，在容易引起闪烁、阴影的 STN/DSTN 面板实现闪烁较少的65536颜色控制
- 连接CPU的同步DRAM （区域3）的一部分作为保存LCDC显示数据的VRAM使用，不需要专用显示存储器
- 通过2.4K字节的大尺寸行缓冲器，实现稳定显示
- 支持适合液晶面板的信号极性的输出信号、输出信号的电平反转功能
- 支持通过寄存器选择各种数据格式 （字节内的端法设定、反像素方式）
- 可在用户指定的位置产生中断 （通过控制VRAM的更新开始时序，避免闪烁）
- 支持硬件旋转模式 （在横向液晶面板支持纵向液晶面板）（旋转前的宽度不能超过320像素。请参照表26.5）

【注】 *1 连接 18 位总线宽度的 TFT 面板时，未连接的低位线的信号连接 GND 或数据输出的最低位。
*2 详细情况请参照 “26.4.1 关于用 LCDC 能显示的液晶模块的尺寸”。

LCDC 的框图如图 26.1 所示。

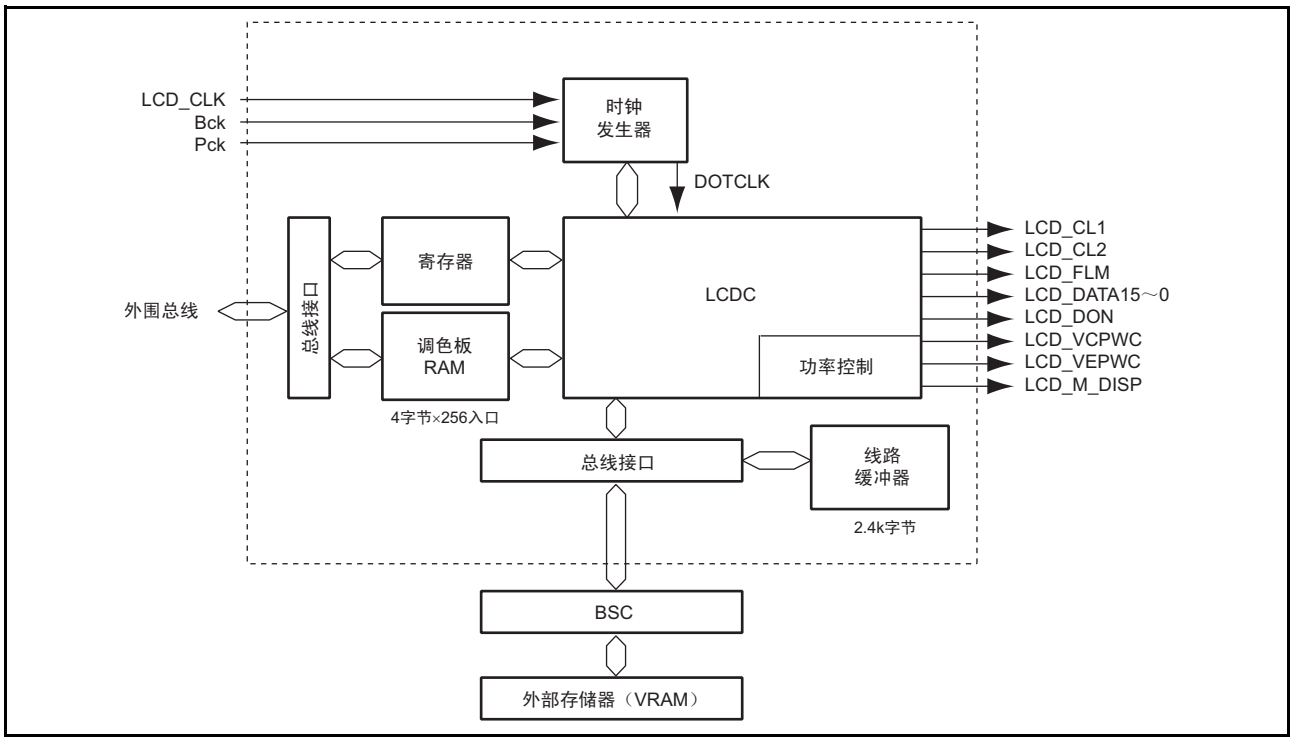


图 26.1　LCDC 的框图

26.2　输入 / 输出引脚

LCDC 的引脚结构如表 26.1 所示。

表 26.1　引脚结构

引脚名	输入 / 输出	功　　能
LCD_DATA15 ～ 0	输出	LCD 面板用数据
LCD_DON	输出	显示开始信号 （DON）
LCD_CL1	输出	移位时钟 1 （STN/DSTN） / 水平同步信号 （HSYNC）（TFT）
LCD_CL2	输出	移位时钟 2 （STN/DSTN） / 点时钟 （DONCLK）（TFT）
LCD_M_DISP	输出	液晶交流信号 /DISP 信号
LCD_FLM	输出	首行标记 / 垂直同步信号 （VSYNC）（TFT）
LCD_VCPWC	输出	液晶模块电源控制 （VCC）
LCD_VEPWC	输出	液晶模块电源控制 （VEE）
LCD_CLK	输入	LCD 时钟源输入

【注】 请确认 “26.5 时钟和 LCD 数据信号例子” 及液晶模块的规格后，决定液晶模块的接线规格。

26.3　寄存器的说明

LCDC 有以下寄存器。这些寄存器的地址及各处理状态中寄存器的状态请参照 “第 34 章　寄存器一览”。

表 26.2　寄存器结构

寄存器名	简称	R/W	初始值	地址	存取长度
LCDC 输入时钟寄存器	LDICKR	R/W	H'0101	H'FFFFFC00	16
LCDC 模块类型寄存器	LDMTR	R/W	H'0109	H'FFFFFC02	16
LCDC 数据格式寄存器	LDDFR	R/W	H'000C	H'FFFFFC04	16
LCDC 扫描模式寄存器	LDSMR	R/W	H'0000	H'FFFFFC06	16
LCDC 上部显示面板用取数据开始地址寄存器	LDSARU	R/W	H'0C000000	H'FFFFFC08	32
LCDC 下部显示面板用取数据开始地址寄存器	LDSARL	R/W	H'0C000000	H'FFFFFC0C	32
LCDC 显示面板用取数据行地址偏移量寄存器	LDLAOR	R/W	H'0280	H'FFFFFC10	16
LCDC 调色板控制寄存器	LDPALCR	R/W	H'0000	H'FFFFFC12	16
调色板数据寄存器 00 ～ FF	LDPR00 ～ FF	R/W	—	H'FFFFF800 ～ H'FFFFFBFC	32
LCDC 水平字符数寄存器	LDHCNR	R/W	H'4F52	H'FFFFFC14	16
LCDC 水平同步信号寄存器	LDHSYNR	R/W	H'0050	H'FFFFFC16	16
LCDC 垂直显示行数寄存器	LDVDLNR	R/W	H'01DF	H'FFFFFC18	16
LCDC 垂直总行数寄存器	LDVTLNR	R/W	H'01DF	H'FFFFFC1A	16
LCDC 垂直同步信号寄存器	LDVSYNR	R/W	H'01DF	H'FFFFFC1C	16
LCDCAC 转变信号交替行数寄存器	LDACLNR	R/W	H'000C	H'FFFFFC1E	16
LCDC 中断控制寄存器	LDINTR	R/W	H'0000	H'FFFFFC20	16
LCDC 电源管理模式寄存器	LDPMMR	R/W	H'0010	H'FFFFFC24	16
LCDC 电源顺序期间寄存器	LDPSPR	R/W	H'F60F	H'FFFFFC26	16
LCDC 控制寄存器	LDCNTR	R/W	H'0000	H'FFFFFC28	16
LCDC 用户指定中断控制寄存器	LDUINTR	R/W	H'0000	H'FFFFFC34	16
LCDC 用户指定中断行数寄存器	LDUINTLNR	R/W	H'004F	H'FFFFFC36	16
LCDC 存储器存取间隔数寄存器	LDLIRNR	R/W	H'0000	H'FFFFFC40	16

26.3.1　LCDC 输入时钟寄存器 （LDICKR）

作为 LCDC 的运行时钟供给源， LCDC 可选择总线时钟、外围时钟或外部时钟。另外，内置 1/1 ～ 1/32 的分频器，可将分频的时钟作为 LCDC 的运行时钟（DOTCLK）使用。LCDC 输出的时钟从该寄存器选择的运行时钟生成液晶面板用的同步时钟输出 （LCD_CL2）。为 TFT 面板时，输出 LCD_CL2=DOTCLK 频率的时钟。为 STN、DSTN 面板时，输出 LCD_CL2=（DOTCLK/ 液晶面板的输出数据总线宽度）频率的时钟。与 LCD_CL2 无关，必须将 LDICKR 设定为输入 LCDC 的时钟不超过 66MHz。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	ICKSEL[1:0]		-	-	-	-	-	-	DCDR[5:0]					
初始值：	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W：	R	R	R/W	R/W	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、 14	—	均为 0	R	保留位 读写的值总是 0。
13、 12	ICKSEL[1:0]	00	R/W	输入时钟选择 设定 DOTCLK 的供给源。 00：选择总线时钟 （Bck） 01：选择外围时钟 （Pck） 10：选择外部时钟 （LCD_CLK） 11：禁止设定
11 ～ 9	—	均为 0	R	保留位 读写的值总是 0。
8	—	均为 1	R	保留位 读写的值总是 1。
7、 6	—	均为 0	R	保留位 读写的值总是 0。
5 ～ 0	DCDR[5:0]	000001	R/W	时钟分频比 设定输入时钟分频比。 设定的详细情况请参照表 26.3。

表 26.3　输入 / 输出时钟频率和分频比

DCDR[5:0]	时钟分频比	输入 / 输出时钟频率 （MHz）		
		50.000	60.000	66.000
000001	1/1	50.000	60.000	66.000
000010	1/2	25.000	30.000	33.000
000011	1/3	16.667	20.000	22.000
000100	1/4	12.500	15.000	16.500
000110	1/6	8.333	10.000	11.000
001000	1/8	6.250	7.500	8.250
001100	1/12	4.167	5.000	5.500
010000	1/16	3.125	3.750	4.125
011000	1/24	2.083	2.500	2.750
100000	1/32	1.563	1.875	2.063

【注】 为上述以外的设定时，时钟分频比为 1/1 （初始值）。

26.3.2　LCDC 模块类型寄存器 （LDMTR）

LDMTR 配合所连接液晶模块的信号极性，设定从 LCDC 输出控制信号以及数据信号的极性。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FLM POL	CL1 POL	DISP POL	DPOL	-	MCNT	CL1CNT	CL2CNT	-	-	MIFTYP[5:0]					
初始值：	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1
R/W：	R/W	R/W	R/W	R/W	R	R/W	R/W	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
15	FLMPOL	0	R/W	FLM （垂直同步）极性选择 选择液晶模块的 LCD_FLM （垂直同步信号、首行标记）的极性。 0：LCD_FLM 脉冲为高电平有效 1：LCD_FLM 脉冲为低电平有效
14	CL1POL	0	R/W	CL1 （水平同步）极性选择 选择液晶模块的 LCD_CL1 （水平同步信号）的极性。 0：LCD_CL1 脉冲为高电平有效 1：LCD_CL1 脉冲为低电平有效
13	DISPPOL	0	R/W	DISP （允许显示）极性选择 选择液晶模块的 LCD_M_DISP （允许显示）的极性。 0：LCD_M_DISP 为高电平有效 1：LCD_M_DISP 为低电平有效
12	DPOL	0	R/W	显示数据极性选择 选择液晶模块的 LCD_DATA （显示数据）的极性。支持液晶模块的反映。 0：LCD_DATA 为高电平有效。透射式液晶面板 1：LCD_DATA 为低电平有效。反射式液晶面板

位	位名	初始值	R/W	说　明
11	—	0	R	保留位 读写的值总是 0。
10	MCNT	0	R/W	M 信号控制 设定液晶模块的液晶交流信号的输出。 0：输出 M （AC 行调制）信号 1：不输出 M 信号
9	CL1CNT	0	R/W	CL1 （水平同步）控制 设定垂直回扫线期间的 LCD_CL1 输出。 0：在垂直回扫线期间输出 LCD_CL1 1：在垂直回扫线期间不输出 LCD_CL1
8	CL2CNT	1	R/W	CL2 （液晶模块的点时钟）控制 设定垂直水平回扫线期间中的 LCD_CL2 输出。 0：在垂直水平回扫线期间中输出 LCD_CL2 1：在垂直水平回扫线期间中不输出 LCD_CL2
7、6	—	均为 0	R	保留位 读写的值总是 0。
5 ～ 0	MIFTYP[5:0]	001001	R/W	模块接口类型选择 设定液晶面板的类型和向液晶面板输出的数据总线宽度。有 3 种液晶面板的类型可选择：STN、 DSTN、 TFT。向液晶面板的输出数据总线宽度可选择：4 位、8 位、12 位、16 位。TFT 的液晶面板的请求数据总线宽度超过 16 位时，请对应面板上存在的数据总线进行连接。与 TFT 不同，在 STN、 DSTN 的液晶面板中，显示色数、显示分辨率和输出数据总线宽度的设定并不是一一对应，因此即使 16bpp 的显示色数也有 8 位的数据总线宽度、 4bpp 的显示色数也有 12 位的数据总线宽度的情况出现。这是由于不是通过数据总线的位数，而是通过数据总线的数据载入方法决定STN、DSTN的显示色数的。关于STN、DSTN 时的数据规格请参照使用液晶面板的规格书。另外，输出数据总线宽度请根据液晶面板的机械接口规格设定。 液晶面板的类型选择 STN 或 DSTN 类型时，与色彩显示、灰度显示的灰度设定无关，通过 LCDC 内置的 RGB 各 8 位的 24 位空调调制 FRC 进行显示控制。因此在 STN 或 DSTN 的显示中，显示从 1600 万颜色中选择 DSPCOLOR 指定的颜色、灰度。使用调色板时，显示在调色板设定的颜色。 000000：STN 单色 4 位数据总线模块 000001：STN 单色 8 位数据总线模块 001000：STN 彩色 4 位数据总线模块 001001：STN 彩色 8 位数据总线模块 001010：STN 彩色 12 位数据总线模块 001011：STN 彩色 16 位数据总线模块 010001：DSTN 单色 8 位数据总线模块 010011：DSTN 单色 16 位数据总线模块 011001：DSTN 彩色 8 位数据总线模块 011010：DSTN 彩色 12 位数据总线模块 011011：DSTN 彩色 16 位数据总线模块 101011：TFT 彩色 16 位数据总线模块 上述以外的设定：设定禁止

26.3.3　LCDC 数据格式寄存器 （LDDFR）

LDDFR 为了配合用于显示的驱动器软件规格，设定在 1 字节中数据的位调整以及显示用数据的型号和色数。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PABD	-	DSPCOLOR[6:0]						
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
R/W：	R	R	R	R	R	R	R	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
15 ～ 9	—	均为 0	R	保留位 读写的值总是 0。
8	PABD	0	R/W	字节数据像素调整 设定 1 字节数据中的像素数据的调整种类。被调整的每个像素的数据内容相同，与该位的内容无关。例如，数据 H'05 不是选择 2 进制的 B'0101 或 B'1010，而是作为 CPU 通过 MOV 指令正常处理状态的 H'05（B'0101）。 0：将字节数据设定为大端法 1：将字节数据设定为小端法
7	—	0	R	保留位 读写的值总是 0。
6 ～ 0	DSPCOLOR [6:0]	0001100	R/W	显示颜色选择 设定显示器的显示色数 （将 0 写入 4、 5、 6bpp 高位）。 有 （通过调色板）相关描述的显示颜色，实际上调色板设定的颜色可通过显示数据选择并显示。 通过显示分辨率，限制旋转显示时可支持的色数。详细情况请参照表 26.5。 0000000：单色、 2 灰度、 1bpp （通过调色板） 0000001：单色、 4 灰度、 2bpp （通过调色板） 0000010：单色、 16 灰度、 4bpp （通过调色板） 0000100：单色、 64 灰度、 6bpp （通过调色板） 0001010：彩色、 16 色、 4bpp （通过调色板） 0001100：彩色、 256 色、 8bpp （通过调色板） 0011101：彩色、 32k 色 （RGB：5-5-5）、 15bpp 0101101：彩色、 64k 色 （RGB：5-6-5）、 16bpp 上述以外的设定：设定禁止

26.3.4　LCDC 扫描模式寄存器 （LDSMR）

LDSMR 选择旋转液晶面板后再使用的硬件旋转功能的开 / 关，以及对确保显示用的 VRAM （区域 3 的同步 DRAM）指定突发长度。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	ROT	-	-	-	AU[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/W	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说　明
15、14	—	均为 0	R	保留位 读写的值总是 0。
13	ROT	0	R/W	选择旋转模块 选择通过硬件旋转显示。但旋转时有以下的限制。 • 模块类型为 STN 或 TFT。不能为 DSTN • 液晶面板的水平 （在液晶面板内部的扫描方向）宽度最大为 320 • 超过LDLAOR显示尺寸，设定2乘方的值（320×240面板旋转为240×320面板，及显示映像的宽度为 240 字节时，请选择 256） 0：无旋转 1：右旋转 90 度 （左边的映像显示在液晶模块的上面）
12 ～ 10	—	均为 0	R	保留位 读写的值总是 0。
9、8	AU[1:0]	00	R/W	选择存取单元 选择 VRAM 的存取单位。只有在 ROT=1（旋转时）时该位有效。ROT=0 时，与 AU 的设定无关，进行 16 突发运行。 00：4 突发 01：8 突发 10：16 突发 11：32 突发 【注】1. 突发长度为 32 位总线宽度，16 位总线宽度时，突发长度变为两倍。 2. 旋转显示时，通过连接 SDRAM 的列地址位和总线宽度，限制突发长度。 详细情况请参照表 26.4 及表 26.5。
7 ～ 0	—	均为 0	R	保留位 读写的值总是 0。

26.3.5　LCDC 上部显示面板用数据输入开始地址寄存器 （LDSARU）

LDSARU 指定将在液晶面板显示的数据输入 LCDC 的开始地址。使用 DSTN 型的液晶面板时，该寄存器指定上部面板的输入开始地址。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	SAU25	SAU24	SAU23	SAU22	SAU21	SAU20	SAU19	SAU18	SAU17	SAU16
初始值：	0	0	0	0	1	1	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：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SAU15	SAU14	SAU13	SAU12	SAU11	SAU10	SAU9	SAU8	SAU7	SAU6	SAU5	SAU4	-	-	-	-
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	R	R	R

位	位名	初始值	R/W	说　明
31 ～ 28	—	均为 0	R	保留位 读写的值总是 0。
27、 26	—	均为 1	R	保留位 读写的值总是 1。
25 ～ 4	SAU25 ～ SAU4	均为 0	R/W	上部面板显示数据的输入开始地址 显示数据的输入开始地址设定在区域 3 的同步 DRAM 区域中。
3 ～ 0	—	均为 0	R	保留位 读写的值总是 0。

【注】 1. 不使用硬件旋转功能时， LDSARU 的最小调整单位为 512 字节。请将低 9 位设定为 0。使用硬件旋转功能时，请设定 LDSARU 的值以便映像左上的地址在 512 字节边界。

2. 使用硬件旋转功能 （ROT=1）时，在该寄存器设定从显示的映像尺寸计算出的映像的左下地址。设定映像为 240×320、 LDLAOR=256 的显示时，可进行以下计算。 LDSARU 的值不是通过面板的尺寸，而是通过显示映像的存储器尺寸计算。请注意，此时 LDLAOR 至少为映像水平尺寸的 2 乘方。此时如果使用 LDSARU 进行倒算，由于 LDSARU–256 （LDLAOR 的值） × （320–1），请将映像左上的地址设定在 512 字节边界。
LDSARU=映像左上的地址 +256 （LDLAOR 的值） ×319 （行）

26.3.6　LCDC 下部显示面板用数据输入开始地址寄存器 （LDSARL）

使用 DSTN 型的液晶面板时， LDSARL 指定面板下部的输入开始地址。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	SAL25	SAL24	SAL23	SAL22	SAL21	SAL20	SAL19	SAL18	SAL17	SAL16
初始值：	0	0	0	0	1	1	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：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SAL15	SAL14	SAL13	SAL12	SAL11	SAL10	SAL9	SAL8	SAL7	SAL6	SAL5	SAL4	-	-	-	-
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	R	R	R

位	位名	初始值	R/W	说　明
31 ～ 28	—	均为 0	R	保留位 读写的值总是 0。
27、 26	—	均为 1	R	保留位 读写的值总是 1。
25 ～ 4	SAL25 ～ SAL4	均为 0	R/W	下部面板显示数据的输入开始地址 在区域 3 的同步 DRAM 区域中设定显示数据的输入开始地址。 STN、 TFT：不使用 DSTN：对应下部面板的显示数据的输入开始地址
3 ～ 0	—	均为 0	R	保留位 读写的值总是 0。

26.3.7　LCDC 显示面板用输入数据行地址偏移量寄存器 （LDLAOR）

LDLAOR 指定 Y 坐标 （用于 LCDC 读出通过图形驱动器识别的映像）增量的地址宽度。指定当 Y 坐标方向增加 1 时，从存储器读数据需要移动多少字节地址。不需要与液晶面板的宽度相同。通过 Ax+By+C 计算 2 维映像上的点 （X、 Y）的存储器地址时，该寄存器等同于与该公式的 B。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LAO15	LAO14	LAO13	LAO12	LAO11	LAO10	LAO9	LAO8	LAO7	LAO6	LAO5	LAO4	LAO3	LAO2	LAO1	LAO0
初始值：	0	0	0	0	0	0	1	0	1	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 ～ 10	LAO15 ～ LAO10	均为 0	R/W	行地址偏移量 最小的调整单位为 16 字节。由于进行 16 字节的处理，因此请设定各寄存器写入值的低 4 位为 0。另外，如果读寄存器的值，低 4 位读为 0。初始值（× 分辨率 =640）连续并准确地配置 VGA（640×480 点）显示数据在行之间不跳过地址。详细情况请参照表 26.4 及表 26.5。 考虑到软件的运行速度，推荐 LDLAOR 的值至少为映像宽度的 2 乘方的值。另外，使用硬件旋转功能时，不是液晶面板的宽度（例如、320×240 面板的 320），而是需要至少映像的宽度 （旋转后，为 240×320 时的 240） 2 乘方的值 （该例子中为 256）。
9	LAO9	1	R/W	
8	LAO8	0	R/W	
7	LAO7	1	R/W	
6 ～ 0	LAO6 ～ LAO0	均为 0	R/W	

26.3.8　LCDC 调色板控制寄存器 （LDPALCR）

LDPALCR 选择是 CPU 还是 LCDC 存取调色板存储器。使用调色板存储器在显示运行中时，在正常显示模式改写调色板存储器的内容时，请设定为彩色调色板设定模式。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	PALS	-	-	-	PALEN
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

位	位名	初始值	R/W	说　　明
15 ～ 5	—	均为 0	R	保留位 读写的值总是 0。
4	PALS	0	R	调色板状态 表示调色板存取权的状态。 0：LCDC 使用调色板。正常显示模式 1：主机 （CPU）使用调色板。调色板设定模式
3 ～ 1	—	均为 0	R	保留位 读写的值总是 0。
0	PALEN	0	R/W	调色板读 / 写允许 请求调色板存取权。 0：请求转移至正常显示模式 1：请求转移至彩色调色板设定模式

26.3.9　调色板数据寄存器 00 ～ FF （LDPR00 ～ LDPRFF）

LDPR 为存取在存储器空间直接配置（4 字节 ×256 地址）的调色板数据的寄存器。存取调色板存储器时请存取该寄存器（LDPR00 ～ LDPRFF）中相应的寄存器。每个调色板寄存器为 32 位寄存器，每个调色板寄存器中的 RGB 分别为 8 位区域。该调色板的详细规格请参照 **“26.4.3 关于彩色调色板规格”**。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	PALDnn 23	PALDnn 22	PALDnn 21	PALDnn 20	PALDnn 19	PALDnn 18	PALDnn 17	PALDnn 16
初始值：	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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
	PALDnn 15	PALDnn 14	PALDnn 13	PALDnn 12	PALDnn 11	PALDnn 10	PALDnn 9	PALDnn 8	PALDnn 7	PALDnn 6	PALDnn 5	PALDnn 4	PALDnn 3	PALDnn 2	PALDnn 1	PALDnn 0
初始值：	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W：	R/W	R/W	R/W	R/W	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 ～ 24	—	—	R	保留位
23 ～ 0	PALDnn23 ～ PALDnn0	—	R/W	调色板数据 bit18 ～ 16、9、8、2 ～ 0 为 RGB 各调色板中的保留位。不能设定，但可根据高位进行扩展使用。

【注】 nn=H'00 ～ H'FF

26.3.10　LCDC 水平字符数寄存器 （LDHCNR）

LDHCNR 为液晶模块的水平 （扫描方向）尺寸，以及指定含水平回扫线期间所有扫描宽度的寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HDCN7	HDCN6	HDCN5	HDCN4	HDCN3	HDCN2	HDCN1	HDCN0	HTCN7	HTCN6	HTCN5	HTCN4	HTCN3	HTCN2	HTCN1	HTCN0
初始值：	0	1	0	0	1	1	1	1	0	1	0	1	0	0	1	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	HDCN7	0	R/W	水平显示字符数
14	HDCN6	1	R/W	设定水平画面的显示字符数 （字符 =8 点单位）。
13	HDCN5	0	R/W	请设定为 （显示字符数） –1 的值。
12	HDCN4	0	R/W	（例）使用宽度 640pixel 的液晶模块时
11	HDCN3	1	R/W	HDCN=(640/8)–1=79=H'4F
10	HDCN2	1	R/W	
9	HDCN1	1	R/W	
8	HDCN0	1	R/W	
7	HTCN7	0	R/W	水平总字符数
6	HTCN6	1	R/W	设定水平画面显示的总字符数 （字符 =8 点）。
5	HTCN5	0	R/W	请设定为（总字符数）–1 的值。　但最小的水平回扫线期间为 3 字符（24 点）。
4	HTCN4	1	R/W	（例）使用宽度 640pixel 的液晶模块时
3	HTCN3	0	R/W	HTCN=[(640/8)–1]+3=82=H'52
2	HTCN2	0	R/W	此时，水平总点数为 664 点、水平回扫线期间为 24 点。
1	HTCN1	1	R/W	
0	HTCN0	0	R/W	

【注】 1. HDCN、HTCN 的设定值必须满足 HTCN ≥ HDCN 的关系。另外、HTCN 的总字符数请设定为偶数 （因为设定值设定为 –1 后的值，所以为奇数）。

2. 根据使用显示器的分辨率， HDCN 的设定如下所示。
1bpp 时：（16 的倍数） –1[1 行为 128pixel 的倍数]
2bpp 时：（8 的倍数） –1[1 行为 64pixel 的倍数]
4bpp 时：（4 的倍数） –1[1 行为 32pixel 的倍数]
6bpp/8bpp 时：（2 的倍数） –1[1 行为 16pixel 的倍数]

26.3.11　LCDC 水平同步信号寄存器 （LDHSYNR）

LDHSYNR 为指定液晶面板模块的水平 （扫描方向）同步信号时序的寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HSYNW3	HSYNW2	HSYNW1	HSYNW0	-	-	-	-	HSYNP7	HSYNP6	HSYNP5	HSYNP4	HSYNP3	HSYNP2	HSYNP1	HSYNP0
初始值：	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	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 14 13 12	HSYNW3 HSYNW2 HSYNW1 HSYNW0	0 0 0 0	R/W R/W R/W R/W	水平同步信号宽度 设定水平画面方向同步信号 （CL1、 Hsync）的宽度 （字符 =8 点）。 请设定为 （水平同步信号宽度） –1 的值。 （例）水平同步信号宽度为 8 点时 HSYNW= （8 点 /8 点 / 字符） –1=0=H'0
11 ～ 8	—	均为 0	R	保留位 读写的值总是 0。
7 6 5 4 3 2 1 0	HSYNP7 HSYNP6 HSYNP5 HSYNP4 HSYNP3 HSYNP2 HSYNP1 HSYNP0	0 1 0 1 0 0 0 0	R/W R/W R/W R/W R/W R/W R/W R/W	水平同步信号输出位置 设定水平画面方向的同步信号的输出位置 （字符 =8 点）。 请设定为 （水平同步信号输出位置） –1 的值。 （例）使用宽度 640pixel 的液晶模块时 HSYNP=[(640/8)+1]–1=80=H'50 此时从第 648 点到第 655 点，水平同步信号为有效。

【注】 请满足以下条件：HTCN ≥ HSYNP+HSYNW+1

HSYNP ≥ HDCN+1

26.3.12　LCDC 垂直显示行数寄存器 （LDVDLNR）

LDVDLNR 为指定液晶面板模块的纵向 （与扫描方向垂直的方向）尺寸的寄存器。为 DSTN 时与上下面板的尺寸无关，请指定大于等于面板模块的纵向尺寸的偶数 （例子：为 640×480 的面板时为 480）。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	VDLN10	VDLN9	VDLN8	VDLN7	VDLN6	VDLN5	VDLN4	VDLN3	VDLN2	VDLN1	VDLN0
初始值：	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1
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	—	均为 0	R	保留位 读写的值总是 0。
10	VDLN10	0	R/W	垂直显示行数 设定垂直的显示行数 （单位：行）。 请设定为 （显示行数） –1 的值。 （例）使用 480 行的液晶模块时 VDLN=480–1=479=H'1DF
9	VDLN9	0	R/W	
8	VDLN8	1	R/W	
7	VDLN7	1	R/W	
6	VDLN6	1	R/W	
5	VDLN5	0	R/W	
4	VDLN4	1	R/W	
3	VDLN3	1	R/W	
2	VDLN2	1	R/W	
1	VDLN1	1	R/W	
0	VDLN0	1	R/W	

26.3.13　LCDC 垂直总行数寄存器 （LDVTLNR）

LDVTLNR 为指定包含液晶面板模块的垂直回扫线期间的所有纵向长度的寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	VTLN10	VTLN9	VTLN8	VTLN7	VTLN6	VTLN5	VTLN4	VTLN3	VTLN2	VTLN1	VTLN0
初始值：	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1
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	—	均为 0	R	保留位 读写的值总是 0。
10	VTLN10	0	R/W	垂直总行数 设定垂直的总行数 （单位：行）。 设定为 （总行数） –1 的值。 最小的垂直总行数为 2 行。 请满足 VTLN ≥ VDLN、 VTLN ≥ 1。 （例）使用 480 行的液晶模块，垂直回扫线期间为 0 行时 VTLN=(480+0)–1=479=H'1DF
9	VTLN9	0	R/W	
8	VTLN8	1	R/W	
7	VTLN7	1	R/W	
6	VTLN6	1	R/W	
5	VTLN5	0	R/W	
4	VTLN4	1	R/W	
3	VTLN3	1	R/W	
2	VTLN2	1	R/W	
1	VTLN1	1	R/W	
0	VTLN0	1	R/W	

26.3.14　LCDC 垂直同步信号寄存器 （LDVSYNR）

LDVSYNR 为指定液晶模块纵向 （与扫描方向垂直的方向）的同步信号时序的寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VSYNW3	VSYNW2	VSYNW1	VSYNW0	-	VSYNP10	VSYNP9	VSYNP8	VSYNP7	VSYNP6	VSYNP5	VSYNP4	VSYNP3	VSYNP2	VSYNP1	VSYNP0
初始值：	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1
R/W：	R/W	R/W	R/W	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

位	位名	初始值	R/W	说　明
15 14 13 12	VSYNW3 VSYNW2 VSYNW1 VSYNW0	0 0 0 0	R/W R/W R/W R/W	垂直同步信号宽度 设定垂直画面方向的同步信号 （FLM、 Vsync）宽度 （单位：行）。 设定为 （垂直同步信号宽度） –1 的值。 （例）垂直同步信号宽度为 1 行时 VSYNW=(1–1)=0=H'0
11	—	0	R	保留位 读写的值总是 0。
10 9 8 7 6 5 4 3 2 1 0	VSYNP10 VSYNP9 VSYNP8 VSYNP7 VSYNP6 VSYNP5 VSYNP4 VSYNP3 VSYNP2 VSYNP1 VSYNP0	0 0 1 1 1 0 1 1 1 1 1	R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W	垂直同步信号输出位置 设定垂直画面方向同步信号 （FLM、 Vsync）的输出位置 （单位：行）。 设定为 （垂直同步信号输出位置） –2 的值。 DSTN 时请设定为奇数值。作为 (设定值 +1)/2 处理。 （例）使用 480 行的液晶模块，回扫线为 0 行， 即 VTLN=479 时，在第 1 行垂直同步信号为有效时 • 单显显示器时 VSYNP=[(1–1)+VTLN]mod(VTLN+1) =[(1–1)+479mod(479+1) =479mod480=479 =H'1DF • 双地址时 VSYNP=[(1–1)×2+VTLN]mod(VTLN+1) =[(1–1)×2+479]mod(479+1) =479mod480=479 =H'1DF

26.3.15　LCDCAC 调制信号交替行数寄存器 （LDACLNR）

LDACLNR 为指定交替液晶模块的 AC 调制信号 （液晶交流信号）时序的寄存器。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	ACLN4	ACLN3	ACLN2	ACLN1	ACLN0
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
R/W：	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说　明
15 ～ 5	—	均为 0	R	保留位 读写的值总是 0。
4	ACLN4	0	R/W	AC 行数
3	ACLN3	1	R/W	设定交替液晶模块交流信号的行数 （单位：行）。
2	ACLN2	1	R/W	设定为 （交替行数） –1 的值。
1	ACLN1	0	R/W	（例）每 13 行进行交替时
0	ACLN0	0	R/W	ACLN=13–1=12=H'0C

【注】 在面板的总行数为偶数时，必须设定偶数以便在奇数行进行交替。

26.3.16　LCDC 中断控制寄存器 （LDINTR）

LDINTR 为指定 Vsync 中断起始点的寄存器。关于中断请参照 “26.3.20 LCDC 用户指定中断控制寄存器 （LDUINTR）” 及 “26.3.21 LCDC 用户指定中断行数寄存器 （LDUINTLNR）”。而且，通过该寄存器设定的运行与通过 LCDC 用户指定中断控制寄存器 （LDUINTR）设定的运行是独立的。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MINT EN	FINT EN	VSINT EN	VEINT EN	MINTS	FINTS	VSINTS	VEINTS	-	-	-	-	-	-	-	-
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	R	R	R	R	R	R	R

位	位名	初始值	R/W	说　明
15	MINTEN	0	R/W	存储器存取中断允许 设定通过 LCDC 存取 VRAM 的垂直回扫线期间的起始点是否产生中断。 0：在 VRAM 存取的垂直回扫线期间的起始点不产生中断。 1：在 VRAM 存取的垂直回扫线期间的起始点产生中断。
14	FINTEN	0	R/W	帧结束中断允许 设定向液晶面板输出帧的最后像素时是否产生中断。 0：向液晶面板输出帧的最后像素时不产生中断。 1：向液晶面板输出帧的最后像素时产生中断。
13	VSINTEN	0	R/W	Vsync 开始中断允许 设定 LCDC Vsync 开始时是否产生中断。 0：LCDC Vsync 开始时不产生中断。 1：LCDC Vsync 开始时产生中断。

位	位名	初始值	R/W	说　　明
12	VEINTEN	0	R/W	Vsync 结束中断允许 设定 LCDC Vsync 结束时是否产生中断。 0：LCDC Vsync 结束时不产生中断。 1：LCDC Vsync 结束时产生中断。
11	MINTS	0	R/W	存储器存取中断状态 表示存储器存取中断的处理状态。 该位在产生 LCDC 存储器存取中断时为 1（置位状态）。在存储器存取中断的处理程序中，通过向寄存器写 0 清除该位。 0：表示 LCDC 没产生存储器存取中断或接收到产生存储器存取中断处理结束的通知。 1：表示 LCDC 没有收到产生存储器存取中断处理结束的通知。
10	FINTS	0	R/W	帧结束中断状态 表示帧结束中断的处理状态。 该位在产生 LCDC 帧结束中断时为 1（置位状态）。在帧结束中断的处理程序中，通过向寄存器写 0 清除该位。 0：表示 LCDC 没产生帧结束中断或接收到产生的帧结束中断处理结束的通知。 1：表示 LCDC 没有接收到产生的帧结束中断处理结束的通知。
9	VSINTS	0	R/W	Vsync 开始中断状态 表示 LCDC Vsync 开始中断的处理状态。 该位在产生 LCDC Vsync 开始中断时为 1 （置位状态）。在 Vsync 开始中断的处理程序中，通过向寄存器写 0 清除该位。 0：表示 LCDC 没产生 Vsync 开始中断或接收到产生 Vsync 开始中断处理结束的通知。 1：表示 LCDC 没有接收到产生 Vsync 开始中断处理结束的通知。
8	VEINTS	0	R/W	Vsync 结束中断状态 表示 LCDC Vsync 结束中断的处理状态。 该位在产生 LCDC Vsync 结束中断时为 1 （置位状态）。在 Vsync 结束中断的处理程序中，通过向寄存器写 0 清除该位。 0：表示 LCDC 不产生 Vsync 结束中断、或接收到产生的 Vsync 结束中断处理结束的通知。 1：表示 LCDC 没有接收到产生 Vsync 结束中断处理结束的通知。
7 ～ 0	—	均为 0	R	保留位 读写的值总是 0。

26.3.17　LCDC 电源管理模式寄存器 （LDPMMR）

LDPMMR 为设定电源电路控制功能的寄存器，该电源电路用于向液晶面板模块提供电源。设定是否使用 2 种电源控制引脚 LCD_VCPWC 和 LCD_VEPWC，以及通电功能的 on/off。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ONC3	ONC2	ONC1	ONC0	OFFD3	OFFD2	OFFD1	OFFD0	-	VCPE	VEPE	DONE	-	-	LPS[1:0]	
初始值：	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R

位	位名	初始值	R/W	说　　明
15 14 13 12	ONC3 ONC2 ONC1 ONC0	0 0 0 0	R/W R/W R/W R/W	LCDC 通电顺序期间 在 LCD 模块的通电顺序中，从 LCD_VEPWC 引脚断言到 LCD_DON 引脚断言期间按照帧周期单位设定。请设定为 （期间） –1 的值。 相当于图 26.4 ～图 26.7 中“电源控制顺序和液晶模块的运行状态”的（c）期间。详细情况请参照表 26.6（设定方法 ONA、 ONB、 OFFD、 OFFE、 OFFF 各寄存器通用）。
11 10 9 8	OFFD3 OFFD2 OFFD1 OFFD0	0 0 0 0	R/W R/W R/W R/W	LCDC 断电顺序期间 在 LCD 模块的断电顺序中，从 LCD_DON 引脚取反到 LCD_VEPWC 引脚取反期间按照帧周期单位设定。请设定 （期间）为 –1 的值。 相当于图 26.4 ～图 26.7 中“电源控制顺序和液晶模块的运行状态”的（d）期间。
7	—	0	R	保留位 读写的值总是 0。
6	VCPE	0	R/W	LCD_VCPWC 引脚允许 设定是否有使用 LCD_VCPWC 引脚的电源控制顺序处理 0：（无处理）屏蔽 LCD_VCPWC 引脚输出并固定为低电平 1：（有处理）按照规定的顺序，断言或取反 LCD_VCPWC 引脚输出
5	VEPE	0	R/W	LCD_VEPWC 引脚允许 设定是否有使用 LCD_VEPWC 引脚的电源控制顺序处理。 0：（无处理）屏蔽 LCD_VEPWC 引脚输出并固定为低电平 1：（有处理）按照规定的顺序，断言或取反 LCD_VEPWC 引脚输出
4	DONE	1	R/W	LCD_DON 引脚允许 设定是否有使用 LCD_DON 引脚电源控制顺序处理。 0：（无处理）屏蔽 LCD_DON 引脚输出并固定为低电平 1：（有处理）按照规定的顺序，断言或取反 LCD_DON 引脚输出
3、 2	—	均为 0	R	保留位 读写的值总是 0。
1、 0	LPS[1:0]	00	R	液晶模块电源输入状态 表示使用电源、控制功能时的液晶模块的通电状态。 00：液晶模块断电 11：液晶模块通电

26.3.18　LCDC 通电顺序期间寄存器 （LDPSPR）

LDPSPR 为设定电源电路控制功能的寄存器，该电源电路用于向液晶模块提供电源。指定 LCD_VEPWC、LCD_VCPWC 引脚以及伴随其时序信号的开始输出时序。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ONA3	ONA2	ONA1	ONA0	ONB3	ONB2	ONB1	ONB0	OFFE3	OFFE2	OFFE1	OFFE0	OFFF3	OFFF2	OFFF1	OFFF0
初始值：	1	1	1	1	0	1	1	0	0	0	0	0	1	1	1	1
R/W：	R/W	R/W	R/W	R/W	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 13 12	ONA3 ONA2 ONA1 ONA0	1 1 1 1	R/W R/W R/W R/W	LCDC 通电顺序期间 在 LCD 模块的通电顺序中，从 LCD_VCPWC 引脚断言到显示数据（LCD_DATA）和时序信号（LCD_FLM、LCD_CL1、LCD_CL2、LCD_M_DISP）输出开始期间，按照帧周期单位进行设定。请设定为（期间）–1 的值。 相当于图 26.4 ～图 26.7 中“电源控制顺序和液晶模块的运行状态”的（a）期间。
11 10 9 8	ONB3 ONB2 ONB1 ONB0	0 1 1 0	R/W R/W R/W R/W	LCDC 通电顺序期间 在 LCD 模块的通电顺序中，从显示数据（LCD_DATA）和时序信号（LCD_FLM、LCD_CL1、LCD_CL2、LCD_M_DISP）输出开始到 LCD_VEPWC 引脚断言期间，按照帧周期单位进行设定。请设定为（期间）–1 的值。 相当于图 26.4 ～图 26.7 中“电源控制顺序和液晶模块的运行状态”的（b）期间。
7 6 5 4	OFFE3 OFFE2 OFFE1 OFFE0	0 0 0 0	R/W R/W R/W R/W	LCDC 断电顺序期间 在液晶模块的断电顺序中，从 LCD_VEPWC 引脚取反到显示数据（LCD_DATA）和时序信号（LCD_FLM、LCD_CL1、LCD_CL2、LCD_M_DISP）输出停止期间，按照帧周期单位进行设定。请设定为（期间）–1 的值。 相当于图 26.4 ～图 26.7 中“电源控制顺序和液晶模块的运行状态”的（e）期间。
3 2 1 0	OFFF3 OFFF2 OFFF1 OFFF0	1 1 1 1	R/W R/W R/W R/W	LCDC 断电顺序期间 在液晶模块的断电顺序中，从显示数据（LCD_DATA）和时序信号（LCD_FLM、LCD_CL1、LCD_CL2、LCD_M_DISP）输出停止到 LCD_VCPWC 引脚取反期间，按照帧周期单位进行设定。请设定为（期间）–1 的值。 相当于图 26.4 ～图 26.7 中“电源控制顺序和液晶模块的运行状态”的（f）期间。

26.3.19　LCDC 控制寄存器 （LDCNTR）

LDCNTR 指定 LCDC 显示运行的开始 / 结束。

DON2 位和 DON 位分别写 1 时，LCDC 开始显示。之后按照在 LDPMMR 及 LDPSPR 设定的顺序，给液晶模块通电。如果 LPS1、LPS0 位从 B'00 变为 B'11 时，规定的顺序就结束。而且到规定的顺序结束为止，不能进行下个 DON 位的操作。

LCDC 结束显示时，设定 DON 位为 0。按照在 LDPMMR 及 LDPSPR 设定的顺序，关掉液晶模块的电源。LCDC 停止运行。如果 LPS[1:0] 从 B'11 变为 B'00 时，规定的顺序就结束。而且到规定的顺序结束为止，不能进行下个 DON 位的操作。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	DON2	-	-	-	DON
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	R/W

位	位名	初始值	R/W	说　　明
15 ～ 5	—	均为 0	R	保留位 读写的值总是 0。
4	DON2	0	R/W	显示器开 2 指定 LCDC 显示运行开始。 0：LCDC 运行 / 结束时 1：开始运行 读取该位时，总是读出 0。请只在开始显示时写 1。开始显示以外时，如果写 1，就无法保证运行。写入 1 时自动恢复为 0，因此该位不需要通过写 0 来清除 1
3 ～ 1	—	均为 0	R	保留位 读写的值总是 0。
0	DON	0	R/W	显示器开 指定 LCDC 显示运行开始 / 结束。 控制顺序的状态可通过参照 LDPMMR 的 LPS[1:0] 值进行确认。 0：显示关模式：LCDC 停止。 1：显示开模式：LCDC 运行。

【注】 1. 显示开始时将 H'0011 写入 LDCNTR，显示结束时将 H'0000 写入 LDCNTR。请不要写入其他的值。

2. 由于 DON2 位写 1 时，调色板 RAM 数据不定。因此在 DON2 位写 1 后，请设定调色板 RAM 的数据。

3. 写入 LDCNTR 后，到实际显示开始 / 结束需要一定时间。因此，写入 LDCNTR 后，在存取 LCDC 的其他寄存器前，请进行 1 次 LDCNTR 的虚读。

26.3.20　LCDC 用户指定中断控制寄存器 （LDUINTR）

LDUINTR 寄存器设定是否产生用户指定中断，并且显示其状态。LCDC 从 VRAM 读完在 LCDC 用户指定中断行数寄存器 （LDUINTLNR）设定的行的映像数据时，产生该中断。

该 LCDC 发送的中断（LCDCI）：通过该寄存器的用户指定中断、通过 LCDC 中断控制寄存器（LDINTR）的存储器存取和 Vsync 中断输出的 OR。而且该寄存器和 LCDC 中断控制寄存器 （LDINTR）的设定独立作用于中断运行。

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

位	位名	初始值	R/W	说　　明
15 ～ 9	—	均为 0	R	保留位 读写的值总是 0。
8	UINTEN	0	R/W	用户指定中断允许 设定是否产生 LCDC 用户指定中断。 0：不产生 LCDC 用户指定中断。 1：产生 LCDC 用户指定中断。
7 ～ 1	—	均为 0	R	保留位 读写的值总是 0。
0	UINTS	0	R/W	用户指定中断状态 该位在产生 LCDC 用户指定中断时为 1（置位状态）。在用户指定中断的处理程序中，通过向寄存器写 0 清除该位。 0：表示 LCDC 没产生用户指定中断、或接收到产生的用户指定中断处理结束的通知。 1：表示 LCDC 没有接收到产生的用户指定中断处理结束的通知。

【注】 中断处理的流程

1. 输入中断信号
2. 读取 LDINTR
3. 如果 MINTS、FINTS、VSINTS、VEINTS=1，该中断就为存储器存取中断、帧结束中断、Vsync 上升沿中断或 Vsync 下降沿中断。对各中断进行处理。
4. 如果 MINTS=FINTS=VSINTS=VEINTS=0，该中断就不为存储器存取中断、帧结束中断、Vsync 上升沿中断或 Vsync 下降沿中断。
5. 读取 UINTS
6. 如果 UINTS=1，则该中断为用户指定中断。对用户指定中断进行处理。
7. 如果 UINTS=0，则该中断不为用户指定中断。进行其他处理。

26.3.21　LCDC 用户指定中断行数寄存器 （LDUINTLNR）

LDUINTLNR 为设定产生用户指定中断位置的寄存器。设定能按照水平行单位进行。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	UINTLN10	UINTLN9	UINTLN8	UINTLN7	UINTLN6	UINTLN5	UINTLN4	UINTLN3	UINTLN2	UINTLN1	UINTLN0
初始值：	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1
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	—	均为 0	R	保留位 读写的值总是 0。
10	UINTLN10	0	R/W	用户指定中断产生行数 设定产生用户指定中断的行 （单位：行）。 请设定为 （中断产生行数） –1 的值。 （例子）在 80 行产生用户指定中断时： HINTLN=160/2–1=79=H'04F
9	UINTLN9	0	R/W	
8	UINTLN8	0	R/W	
7	UINTLN7	0	R/W	
6	UINTLN6	1	R/W	
5	UINTLN5	0	R/W	
4	UINTLN4	0	R/W	
3	UINTLN3	1	R/W	
2	UINTLN2	1	R/W	
1	UINTLN1	1	R/W	
0	UINTLN0	1	R/W	

【注】 1. 使用 STN/TFT 的液晶模块时，该寄存器的设定值请不要超过 LDVDLNR 的垂直显示行数 （VDLN）。

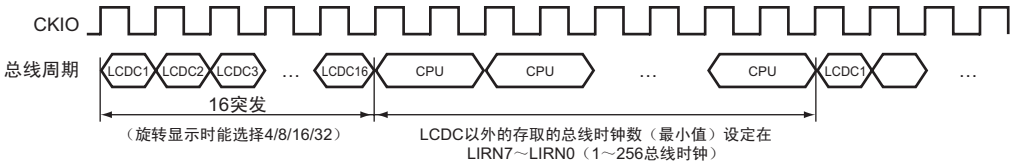
2. 使用 DSTN 液晶模块时，该寄存器的设定值请不要超过 LDVDLNR 的垂直显示行数 （VDLN）的 1/2。此时，LCDC 从 VRAM 读完将下部画面的映像数据设定在该寄存器的行数时，产生用户指定中断。

26.3.22　LCDC 存储器存取间隔号寄存器 （LDLIRNR）

LDLIRNR 控制 LCDC 读取 VRAM 时的总线时钟间隔。在通过 LDLIRNR 设定的总线时钟期间，由于 LCDC 不存取 VRAM，因此，在此期间可通过 CPU/DMAC 进行外部总线存取。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	LIRN7	LIRN6	LIRN5	LIRN4	LIRN3	LIRN2	LIRN1	LIRN0
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
15 ～ 8	—	均为 0	R	保留位 读写的值总是 0。
7 ～ 0	LIRN7 ～ LIRN0	均为 0	R/W	LCDC 的 VRAM 读出总线时钟间隔 指定在 LCDC 读出 VRAM 的突发总线周期期间插入的总线时钟数。 H'00：1 个总线周期 H'01：2 个总线周期 ： H'FF：256 个总线周期



26.4　运行说明

26.4.1　关于用 LCDC 能显示的液晶模块的尺寸

LCDC 可进行 1024×1024 点、16bpp（比特每象素）的显示。但显示的映像保存在 VRAM 与 CPU 共享，在显示前该 LCDC 需要从 VRAM 读出数据。

在该 LSI 中，通过内置最大 32 突发的存储器读和 2.4K 字节的行缓冲器，虽然不容易发生显示失败，但由于组合，有时会出现显示困难的情况。帧速率为60Hz时的推荐尺寸为：16bpp时320×240点或8bpp时640×480点。

大致的标准是，以下所示的总线占有率不能超过 40％。

$$\text{总线占有率}(\%) = \frac{\text{开销系数} \times \text{显示总像素}((\text{HDCN}+1)\times 8\times(\text{VDLN}+1)) \times \text{帧速率}(\text{Hz}) \text{色数}(\text{bpp})}{\text{CKIO}(\text{Hz}) \times \text{总线宽度}(\text{bit})} \times 100$$

开销系数在 CL2 的 SDRAM 为 32 位总线时为 1.375，为 16 位总线时为 1.188。

关于有效的显示和回扫线期间如图 26.2 所示。

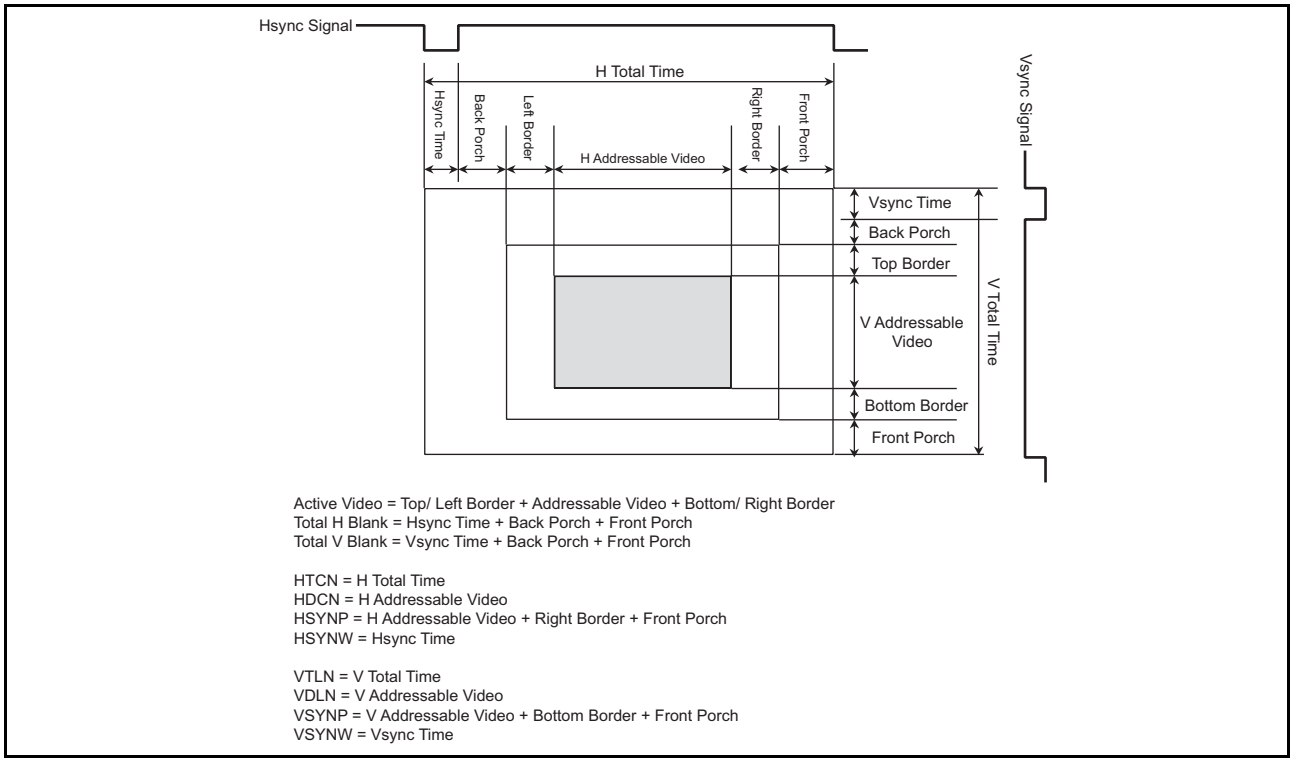


图 26.2　有效的显示和回扫线期间

26.4.2　旋转显示分辨率 / 突发长度及连接存储器 （SDRAM）的限制事项

该 LCDC 可将纵向的显示映像旋转 90 度，并在对应的横向液晶模块上显示。关于各自的分辨率，只能显示如下所示的色数。另外，为了连续读取 SDRAM，对 SDRAM 的尺寸 （列地址位数）及 LCDC 的突发长度有所限制。

显示色数、 SDRAM 列地址数及 LCDC 的突发长度的限制如**表 26.4** 及**表 26.5** 所示。

单色的显示映像需要单色的液晶模块，彩色的显示映像需要彩色的液晶模块。

表 26.4　旋转显示分辨率 / 突发长度及连接存储器的限制 （SDRAM32 位宽度时）

存储器中的显示映像（X 分辨率 × Y 分辨率）	LCDC 模块（X 分辨率 × Y 分辨率）	显示色数		使用 SDRAM 的列地址位数	LCDC 的突发长度（LDSMR*）
240×320	320×240	单色	4bpp（packed）	8 位	小于等于 8 突发
				9 位	小于等于 16 突发
				10 位	—
			4bpp（unpacked）	8 位	4 突发
				9 位	小于等于 8 突发
				10 位	小于等于 16 突发
			6bpp	8 位	4 突发
				9 位	小于等于 8 突发
				10 位	小于等于 16 突发
		彩色	8bpp	8 位	4 突发
				9 位	8 突发以下
				10 位	16 突发以下
			16bpp	8 位	不能使用
				9 位	4 突发
				10 位	小于等于 8 突发
234×320	320×234	单色	6bpp	8 位	4 突发
				9 位	小于等于 8 突发
				10 位	小于等于 16 突发
		彩色	16bpp	8 位	不能使用
				9 位	4 突发
				10 位	小于等于 8 突发

存储器中的显示映像（X 分辨率 × Y 分辨率）	LCDC 模块（X 分辨率 × Y 分辨率）	显示色数		使用 SDRAM 的列地址位数	LCDC 的突发长度（LDSMR*）
80×160	160×80	单色	2bpp	8 位	—
				9 位	—
				10 位	—
			4bpp（packed）	8 位	小于等于 16 突发
				9 位	—
				10 位	—
			4bpp（unpacked）	8 位	小于等于 8 突发
				9 位	小于等于 16 突发
				10 位	—
			6bpp	8 位	小于等于 8 突发
				9 位	小于等于 16 突发
				10 位	—
		彩色	4bpp（packed）	8 位	小于等于 16 突发
				9 位	—
				10 位	—
			4bpp（unpacked）	8 位	小于等于 8 突发
				9 位	小于等于 16 突发
				10 位	—
			8bpp	8 位	小于等于 8 突发
				9 位	小于等于 16 突发
				10 位	—
			16bpp	8 位	4 突发
				9 位	小于等于 8 突发
				10 位	小于等于 16 突发

存储器中的显示映像（X 分辨率 × Y 分辨率）	LCDC 模块（X 分辨率 × Y 分辨率）	显示色数		使用 SDRAM 的列地址位数	LCDC 的突发长度（LDSMR*）
64×128	128×64	单色	1bpp	8 位	—
				9 位	—
				10 位	—
			2bpp	8 位	—
				9 位	—
				10 位	—
			4bpp（packed）	8 位	—
				9 位	—
				10 位	—
			4bpp（unpacked）	8 位	小于等于 16 突发
				9 位	—
				10 位	—
			6bpp	8 位	小于等于 16 突发
				9 位	—
				10 位	—
		彩色	4bpp（packed）	8 位	—
				9 位	—
				10 位	—
			4bpp（unpacked）	8 位	小于等于 16 突发
				9 位	—
				10 位	—
			8bpp	8 位	小于等于 16 突发
				9 位	—
				10 位	—

【注】 * 设定数据时，必须使设定为突发长度的行数的数据存储在 SDRAM 的相同 ROW 地址中。

表 26.5　旋转显示分辨率 / 突发长度及连接存储器的限制 （SDRAM16 位宽度时）

存储器中的显示映像（X 分辨率 × Y 分辨率）	LCDC 模块（X 分辨率 × Y 分辨率）	显示色数		使用 SDRAM 的列地址位数	LCDC 的突发长度（LDSMR*）
240×320	320×240	单色	4bpp（packed）	8 位	小于等于 4 突发
				9 位	小于等于 8 突发
				10 位	小于等于 16 突发
			4bpp（unpacked）	8 位	不能使用
				9 位	4 突发
				10 位	小于等于 8 突发
			6bpp	8 位	不能使用
				9 位	4 突发
				10 位	小于等于 8 突发
		彩色	8bpp	8 位	不能使用
				9 位	4 突发
				10 位	小于等于 8 突发
			16bpp	8 位	不能使用
				9 位	不能使用
				10 位	4 突发
234×320	320×234	单色	6bpp	8 位	不能使用
				9 位	4 突发
				10 位	小于等于 8 突发
		彩色	16bpp	8 位	不能使用
				9 位	不能使用
				10 位	4 突发

存储器中的显示映像（X 分辨率 × Y 分辨率）	LCDC 模块（X 分辨率 × Y 分辨率）	显示色数		使用 SDRAM 的列地址位数	LCDC 的突发长度（LDSMR*）
80×160	160×80	单色	2bpp	8 位	小于等于 16 突发
				9 位	—
				10 位	—
			4bpp（packed）	8 位	小于等于 8 突发
				9 位	小于等于 16 突发
				10 位	—
			4bpp（unpacked）	8 位	4 突发
				9 位	小于等于 8 突发
				10 位	小于等于 16 突发
			6bpp	8 位	4 突发
				9 位	小于等于 8 突发
				10 位	小于等于 16 突发
		彩色	4bpp（packed）	8 位	小于等于 8 突发
				9 位	小于等于 16 突发
				10 位	—
			4bpp（unpacked）	8 位	4 突发
				9 位	小于等于 8 突发
				10 位	小于等于 16 突发
			8bpp	8 位	4 突发
				9 位	小于等于 8 突发
				10 位	小于等于 16 突发
			16bpp	8 位	不能使用
				9 位	4 突发
				10 位	小于等于 8 突发

存储器中的显示映像（X 分辨率 × Y 分辨率）	LCDC 模块（X 分辨率 × Y 分辨率）	显示色数		使用 SDRAM 的列地址位数	LCDC 的突发长度（LDSMR*）
64×128	128×64	单色	1bpp	8 位	—
				9 位	—
				10 位	—
			2bpp	8 位	—
				9 位	—
				10 位	—
			4bpp（packed）	8 位	小于等于 16 突发
				9 位	—
				10 位	—
			4bpp（unpacked）	8 位	小于等于 8 突发
				9 位	小于等于 16 突发
				10 位	—
			6bpp	8 位	小于等于 8 突发
				9 位	小于等于 16 突发
				10 位	—
		彩色	4bpp（packed）	8 位	小于等于 16 突发
				9 位	—
				10 位	—
			4bpp（unpacked）	8 位	小于等于 8 突发
				9 位	小于等于 16 突发
				10 位	—
			8bpp	8 位	小于等于 8 突发
				9 位	小于等于 16 突发
				10 位	—

【注】 * 设定数据时，必须使设定为突发长度的行数的数据存储在 SDRAM 的相同 ROW 地址中。

26.4.3　关于彩色调色板规格

（1） 调色板寄存器

本 LCDC 有通过每个入口输出 24 位数据，同时使用 256 个入口的调色板，通过利用该调色板可同时显示 16M 色中的 256 色。

该调色板根据以下顺序，用户可随时设定。

1. LDPALCR的PALEN位=0 （初始值）：正常显示运行
2. 存取LDPALCR，设定PALEN=1：在外围时钟的3个周期后，向调色板设定模式转移
3. 存取LDPALCR，确认PALS=1
4. 存取LDPR00～FF，在PALD00～FF写入需要的值
5. 存取LDPALCR，设定PALEN=0：在外围时钟的1个周期后，向正常显示模式转移

在 LDPALCR 的 PALS=1 时， LCDC 显示数据输出 （LCD_DATA）为 0 值输出。

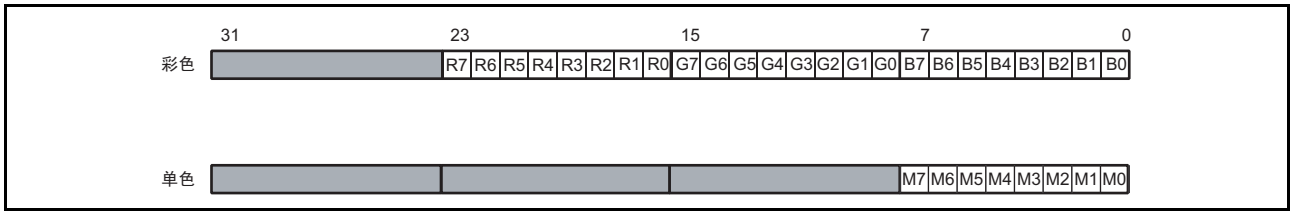


图 26.3　调色板数据格式

PALDnn 的颜色 / 灰度数据设定如上。

彩色显示时， PALDnn[23:16] 设定 R 数据， PALDnn[15:8] 设定 G 数据， PALDnn[7:0] 设定 B 数据。但在 PALDnn[18:16]、PALDnn[9:8]、PALDnn[2:0] 中没有寄存器的位，也没有对应的存储器。因此，PALDnn[18:16]、PALDnn[9:8]、 PALDnn[2:0] 无法保存调色板的数据，因此 R:5 位、 G:6 位、 B:5 位为有效。实际使用时，请写入 24 位 （R:8 位、 G:8 位、 B:8 位）的数据。 PALDnn[23:19]、 PALDnn[15:10]、 PALDnn[7:3] 的值不为 0 时，PALDnn[18:16]、PALDnn[9:8]、PALDnn[2:0] 写 0 或 1，PALDnn[23:19]、PALDnn[15:10]、PALDnn[7:3] 的值为 0 时，通过向 PALDnn[18:16]、 PALDnn[9:8]、 PALDnn[2:0] 写 0，扩展为 24 位。

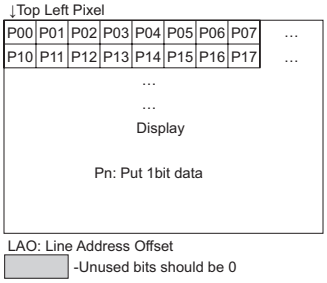
单色显示时，在 PALDnn[7:3] 设定灰度数据。 PALDnn[23:8] 为 don't care。 PALDnn[7:3] 的值不为 0 时，PALDnn[2:0] 写 1， PALDnn[7:3] 的值为 0 时，向 PALDnn[2:0] 写 0，扩展为 8 位。

【符号说明】 nn：H'00 ～ H'FF

26.4.4　数据格式

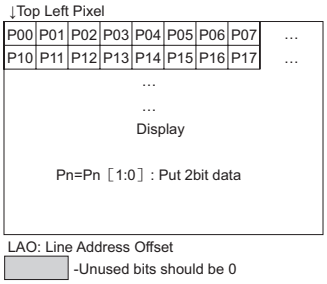
1.Packed 1bpp（Pixel Alignment in Byte is Big Endian）［Windows CE Recommended Format］

Address	MSB 7	6	5	4	3	2	1	LSB 0	［Bit］
+00	P00	P01	P02	P03	P04	P05	P06	P07	（Byte0）
+01	P08								（Byte1）
+02									
+03									
…				…					
+LAO+00	P10	P11	P12	P13	P14	P15	P16	P17	
+LAO+01	P18								
+LAO+02				…					
+LAO+03									
…				Display Memory					



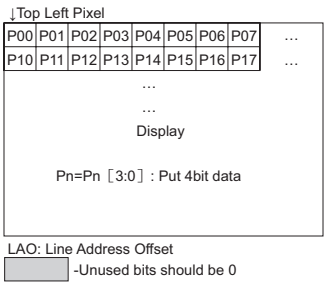
2.Packed 2bpp（Pixel Alignment in Byte is Big Endian）［Windows CE Recommended Format］

Address	MSB 7	6	5	4	3	2	1	LSB 0	［Bit］
+00	P00		P01		P02		P03		（Byte0）
+01	P04		P05		P06		P07		（Byte1）
+02									
+03									
…				…					
+LAO+00	P10		P11		P12		P13		
+LAO+01	P14		P15		P16		P17		
+LAO+02				…					
+LAO+03									
…				Display Memory					



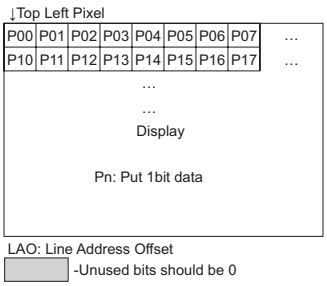
3.Packed 4bpp（Pixel Alignment in Byte is Big Endian）［Windows CE Recommended Format］

Address	MSB 7	6	5	4	3	2	1	LSB 0	［Bit］
+00	P00				P01				（Byte0）
+01	P02				P03				（Byte1）
+02	P04				P05				（Byte2）
+03									
…				…					
+LAO+00	P10				P11				
+LAO+01	P12				P13				
+LAO+02	P14				P15				
+LAO+03				…					
…				Display Memory					



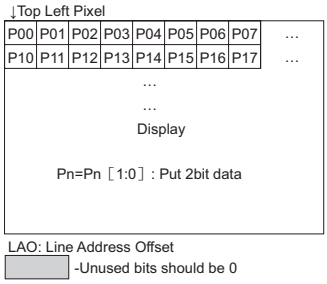
4.Packed 1bpp（Pixel Alignment in Byte is Little Endian）

Address	MSB 7	6	5	4	3	2	1	LSB 0	［Bit］
+00	P07	P06	P05	P04	P03	P02	P01	P00	（Byte0）
+01								P08	（Byte1）
+02									
+03									
…				…					
+LAO+00	P17	P16	P15	P14	P13	P12	P11	P10	
+LAO+01								P18	
+LAO+02				…					
+LAO+03									
…				Display Memory					



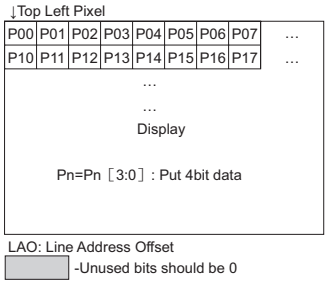
5.Packed 2bpp（Pixel Alignment in Byte is Little Endian）

Address	MSB 7	6	5	4	3	2	1	LSB 0	[Bit]
+00	P03				P02				(Byte0)
+01	P07				P06				(Byte1)
+02									
+03									
…	…								
+LAO+00	P13				P12				
+LAO+01	P17				P16				
+LAO+02	…								
+LAO+03									
…	Display Memory								



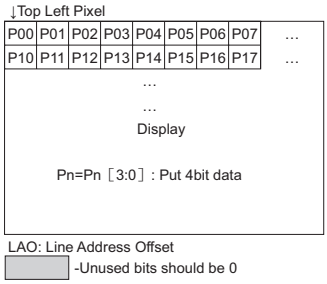
6.Packed 4bpp（Pixel Alignment in Byte is Little Endian）

Address	MSB 7	6	5	4	3	2	1	LSB 0	[Bit]
+00	P01				P00				(Byte0)
+01	P03				P02				(Byte1)
+02	P05				P04				(Byte2)
+03									
…	…								
+LAO+00	P11				P10				
+LAO+01	P13				P12				
+LAO+02	P15				P14				
+LAO+03	…								
…	Display Memory								



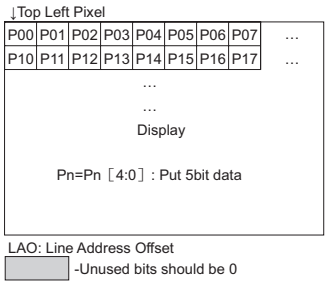
7.Unpacked 4bpp［Windows CE Recommended Format］

Address	MSB 7	6	5	4	3	2	1	LSB 0	[Bit]
+00					P00				(Byte0)
+01					P01				(Byte1)
+02					P02				(Byte2)
+03									
…	…								
+LAO+00					P10				
+LAO+01					P11				
+LAO+02					P12				
+LAO+03	…								
…	Display Memory								

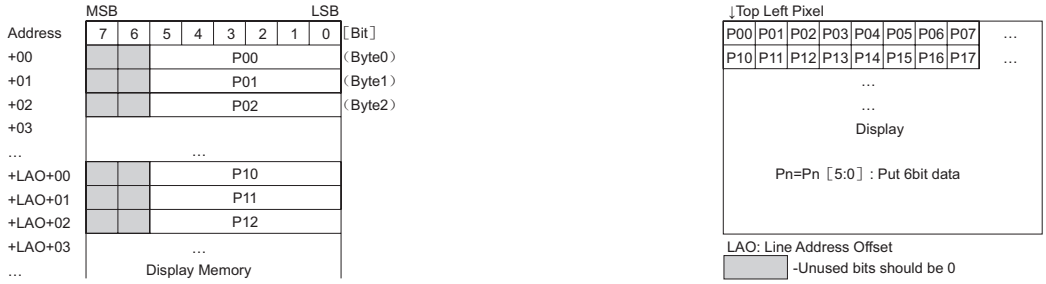


8.Unpacked 5bpp［Windows CE Recommended Format］

Address	MSB 7	6	5	4	3	2	1	LSB 0	[Bit]
+00				P00					(Byte0)
+01				P01					(Byte1)
+02				P02					(Byte2)
+03									
…	…								
+LAO+00				P10					
+LAO+01				P11					
+LAO+02				P12					
+LAO+03	…								
…	Display Memory								



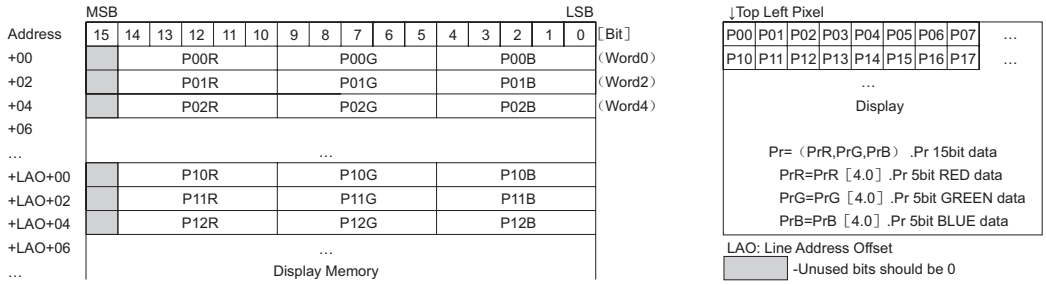
9.Unpacked 6bpp［Windows CE Recommended Format］



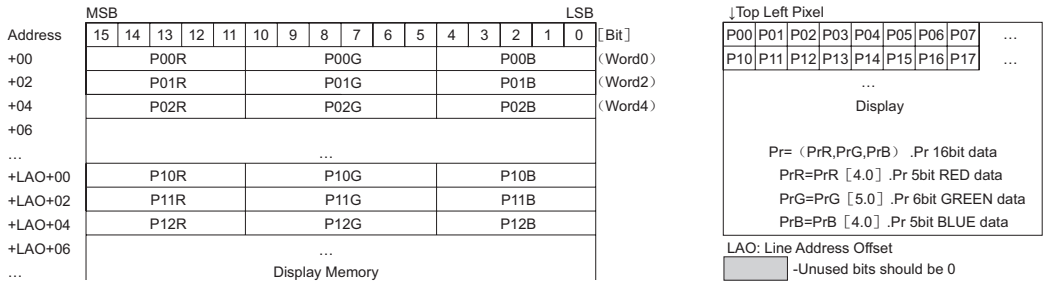
10.Packed 8bpp［Windows CE Recommended Format］



11.Unpacked color 15bpp（RGB 555）［Windows CE Recommended Format］



12.Packed color 16bpp（RGB 565）［Windows CE Recommended Format］



26.4.5　显示分辨率的设定

在 LDHCNR、LDHSYNR、LDVDLNR、LDVTLNR、LDVSYNR 中设定显示分辨率。通过 LDACLNR 设定 STN 或 DSTN 显示时的液晶交流周期。这些寄存器的初始值为 VGA （640×480 点）在 STN 或 DSTN 显示时典型的分辨率设定值。

通过 LDICKR 设定使用的时钟。通过与尺寸相关的寄存器设定的 1 画面的显示画面＋回扫线期间（非显示期间）和使用时钟的频率，决定液晶模块的帧速率。该 LCDC 拥有在每个垂直回扫线期间的起始点（确切说是最后显示行的下一行的起始点）产生中断的 Vsync 中断功能。用 LDINTR 设定该功能。

26.4.6　电源控制顺序处理

通常液晶模块需要进行与电源的接入 / 切断有关的特定顺序处理。可通过设定 LDPMMR、LDPSPR、LDCNTR，使用液晶电源控制引脚 （LCD_VCPWC、LCD_VEPWC、LCD_DON），按照液晶模块的请求执行多种电源控制顺序处理。

电源控制顺序的简略时序图如图 26.4 ～图 26.7 所示，可设定的电源控制顺序期间的说明如表 26.6 所示。

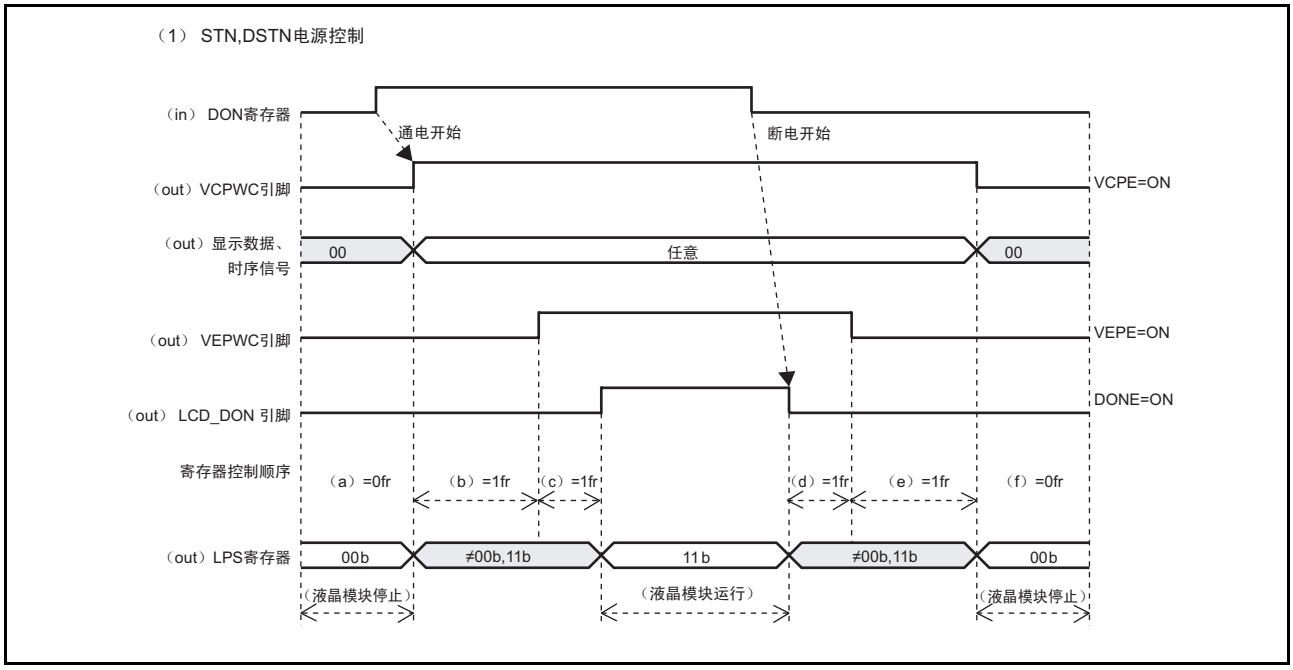


图 26.4　电源控制顺序和液晶模块的运行状态

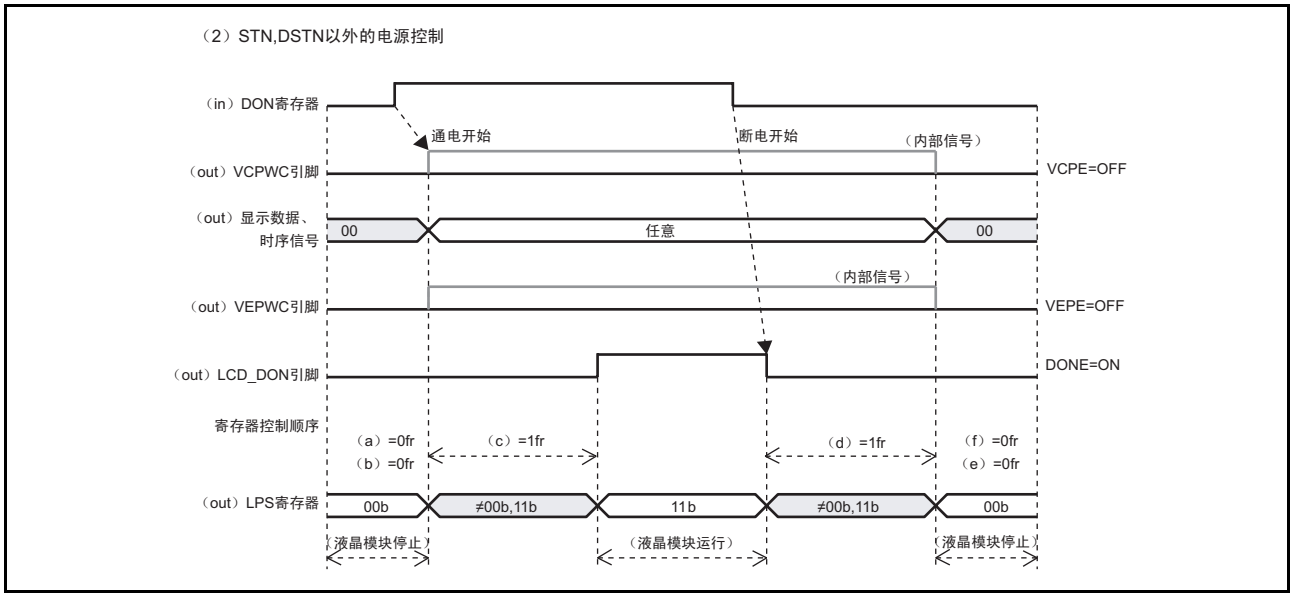


图 26.5　电源控制顺序和液晶模块的运行状态

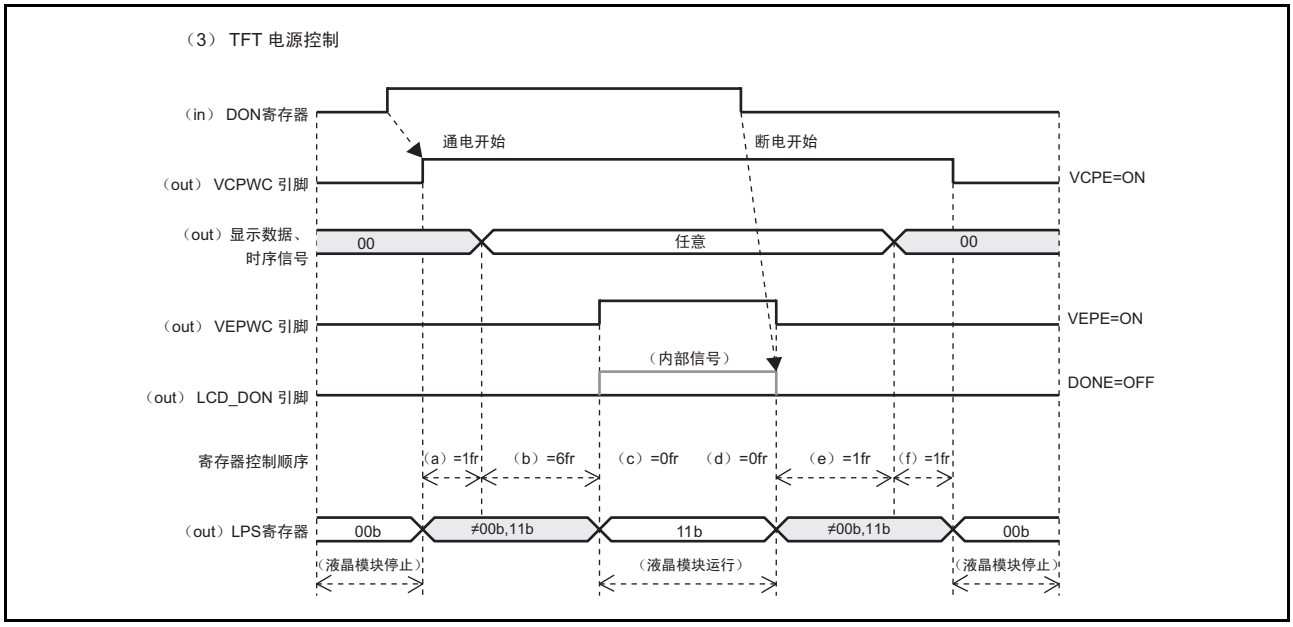


图 26.6　电源控制顺序和液晶模块的运行状态

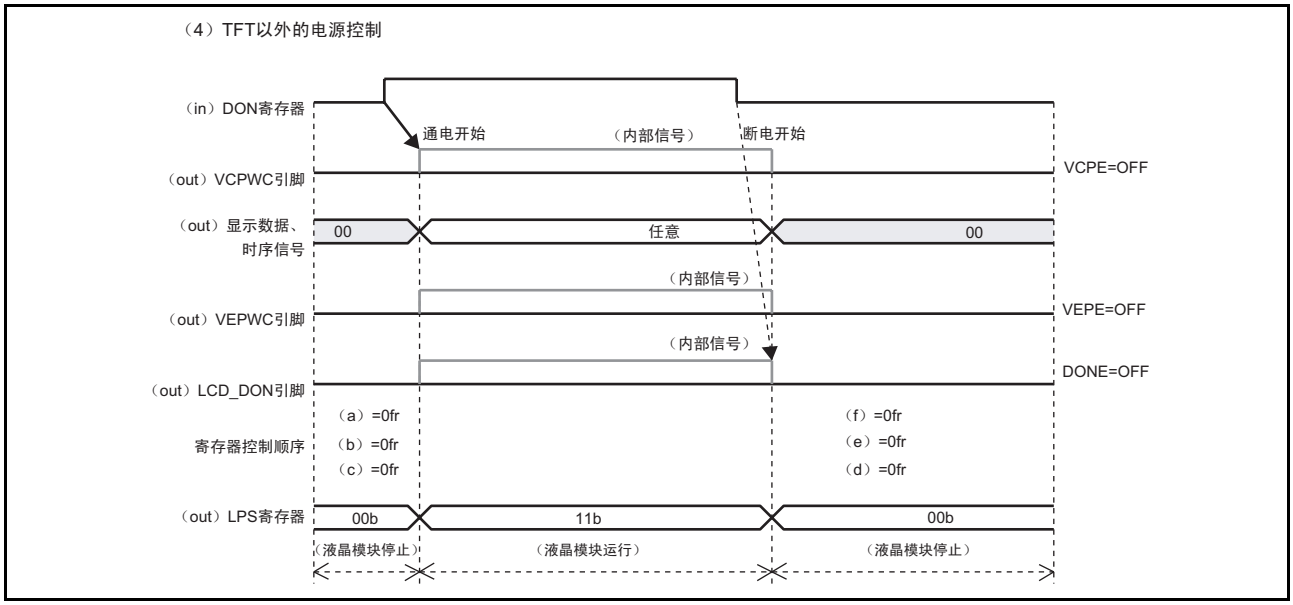


图 26.7 电源控制顺序和液晶模块的运行状态

表 26.6 以典型帧速率，可设定的电源控制顺序期间

ONX、 OFFX 寄存器设定值	帧速率	
	120Hz	60Hz
H'F	(–1+1)/120 = 0.00(ms)	(–1+1)/60 = 0.00(ms)
H'0	(0+1)/120 = 8.33(ms)	(0+1)/60 = 16.67(ms)
H'1	(1+1)/120 = 16.67(ms)	(1+1)/60 = 33.33(ms)
H'2	(2+1)/120 = 25.00(ms)	(2+1)/60 = 50.00(ms)
H'3	(3+1)/120 = 33.33(ms)	(3+1)/60 = 66.67(ms)
H'4	(4+1)/120 = 41.67(ms)	(4+1)/60 = 83.33(ms)
H'5	(5+1)/120 = 50.00(ms)	(5+1)/60 = 100.00(ms)
H'6	(6+1)/120 = 58.33(ms)	(6+1)/60 = 116.67(ms)
H'7	(7+1)/120 = 66.67(ms)	(7+1)/60 = 133.33(ms)
H'8	(8+1)/120 = 75.00(ms)	(8+1)/60 = 150.00(ms)
H'9	(9+1)/120 = 83.33(ms)	(9+1)/60 = 166.67(ms)
H'A	(10+1)/120 = 91.67(ms)	(10+1)/60 = 183.33(ms)
H'B	(11+1)/120 = 100.00(ms)	(11+1)/60 = 200.00(ms)
H'C	(12+1)/120 = 108.33(ms)	(12+1)/60 = 216.67(ms)
H'D	(13+1)/120 = 116.67(ms)	(13+1)/60 = 233.33(ms)
H'E	(14+1)/120 = 125.00(ms)	(14+1)/60 = 250.00(ms)

ONA、ONB、ONC、OFFD、OFFE、OFFF 的各寄存器按照帧周期单位，可设定 0 ～ 15 帧的电源控制顺序期间。设定为寄存器的值 –1， H'0 ～ H'E 设定表示 1 ～ 15 的帧， H'F 设定表示 0 帧。

实际的顺序时间取决于寄存器设定值和显示帧频率。下表为典型的液晶模块所使用的显示帧频率的电源控制顺序期间。

- 显示帧频率为 120Hz， ONB 寄存器设定为 H'6 时
  由于显示帧频率为 120Hz， 1 帧的时间为 8.33(ms)=1/120(sec)
  由于 ONB 寄存器设定为 –1 的值，通电顺序期间为 7 帧。
  因此，此时的顺序时间为 58.33(ms)=8.33(ms)×7。

表 26.7　LCDC 运行模式

模式		功 能
显示 ON （LCDC 运行）	寄存器设定：DON=1	通过规定的分辨率、色数，向液晶模块输出显示数据、时序信号
显示 OFF （LCDC 停止）	寄存器设定：DON=0	可进行寄存器存取。 通过规定的分辨率、色数，不向液晶模块输出显示数据、时序信号

表 26.8　液晶模块电源状态

（STN、 DSTN 模块时）

状　态	逻辑电源	显示数据时序信号	高压电源	DON 信号
对应的控制引脚	LCD_VCPWC	LCD_CL2、LCD_CL1、LCD_FLM、LCD_M_DISP、LCD_DATA	LCD_VEPWC	LCD_DON
运行状态	提供	提供	提供	提供
（过渡状态）	提供	提供	提供	
	提供	提供		
	提供			
停止状态				

（TFT 模块时）

状　态	逻辑电源	显示数据时序信号	高压电源
对应的控制引脚	LCD_VCPWC	LCD_CL2、LCD_CL1、LCD_FLM、LCD_M_DISP、LCD_DATA	LCD_VEPWC
运行状态	提供	提供	提供
（过渡状态）	提供	提供	
	提供		
停止状态			

上述是一般液晶模块的运行状态、停止状态的电源以及显示数据、时序信号的供给状态。但根据模块的不同，也有在模块内部从逻辑电源电压生成高压电源的情况，因此不一定需要提供电源。

- 显示OFF模式 （LCDC停止）的注意事项

通过该 LCDC 使用液晶模块电源控制顺序处理时，如果在处于显示 ON 模式的情况下切断电源时，就无法保证 LCDC 的正常运行。另外，最坏的情况是可能造成连接的液晶模块破损。

26.4.7　硬件旋转运行说明

以下是硬件旋转运行的说明。硬件旋转模式的使用设想为“通过将有横向映像的液晶面板纵向放置，代替纵向的液晶面板使用”。面板的形状不论是横向或纵向都没问题，但宽度不能超过 320。

进行硬件旋转时，必须改变无硬件旋转设定中的以下 5 项 （以下的例子是 8bpp 时的情况。 16bpp 时，每点的存储器尺寸变为 2 倍，因此映像的尺寸以及旋转时寄存器的设定值不同）

1. 映像具备旋转后面板所显示的形状（如果旋转后为240×320像素，就要具备没旋转的240×320尺寸的数据）
2. 改变与映像的地址有关的寄存器设定 （LDSARU、 LDLAOR）。
3. LDLAOR 为2乘方 （旋转后宽度为240像素时，就应将LDLAOR 设置为256像素）。
4. 图形软件与3.的设定相同。
5. 将LDSARU从映像的左上地址变更为左下地址。

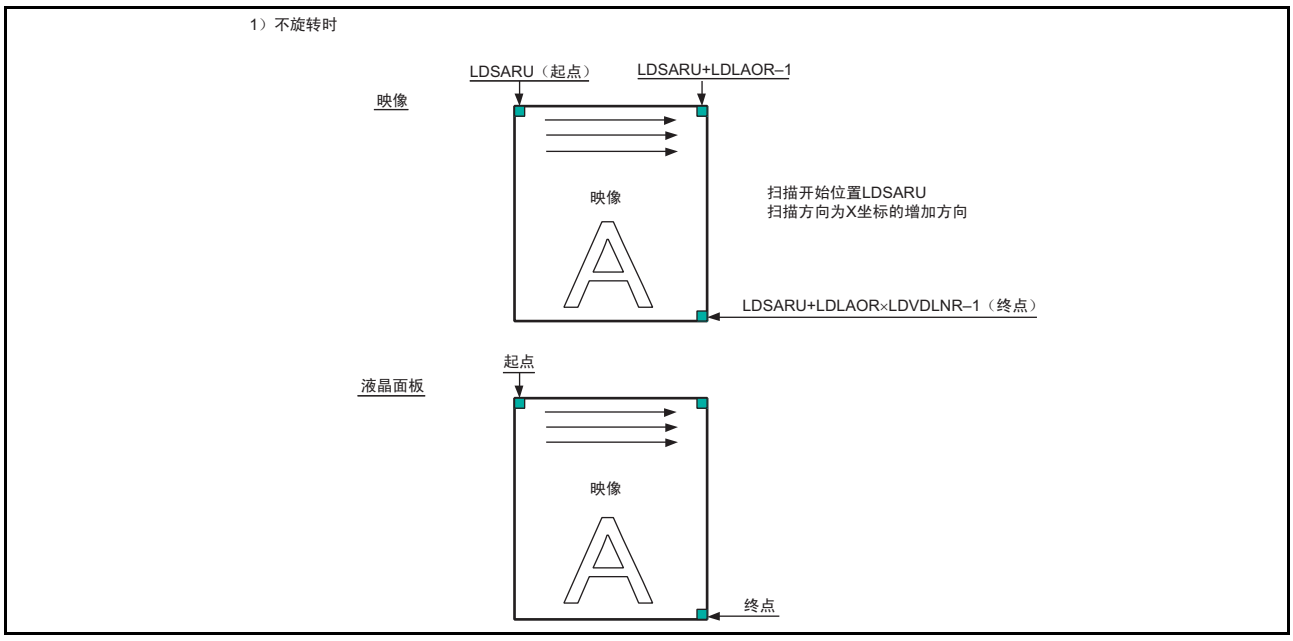


图 26.8　硬件旋转运行 （不旋转时）

例如，在尺寸为 320×240 的液晶面板，已设定显示从 LDSARU=0x0c001000 开始的横向映像 （320×240）的寄存器、并完成图形驱动软件。在该状态中使用硬件旋转，变更为 240×320 显示。此时，如果 LDLAOR=512，图形驱动软件应将用于计算映像 Y 坐标的地址的偏移作为 2 的乘方。该状态中，在 ROT=1 前，映像为了用于 240×320 需自动重画。另外，由于尺寸不同，LDLAOR=256，同时图形驱动软件也将改变。LDSARU 从左上变为左下，因此变更为 LDSARU=0x0c001000+256*319。

【注】 硬件旋转功能是可将液晶面板旋转 90 度的功能，液晶面板的相关设定需要符合旋转前的液晶面板。另外，旋转时可以不考虑图形驱动软件的绘画处理，但图形驱动软件的管理映像的尺寸以及地址偏移的值需要与实际的映像相匹配。

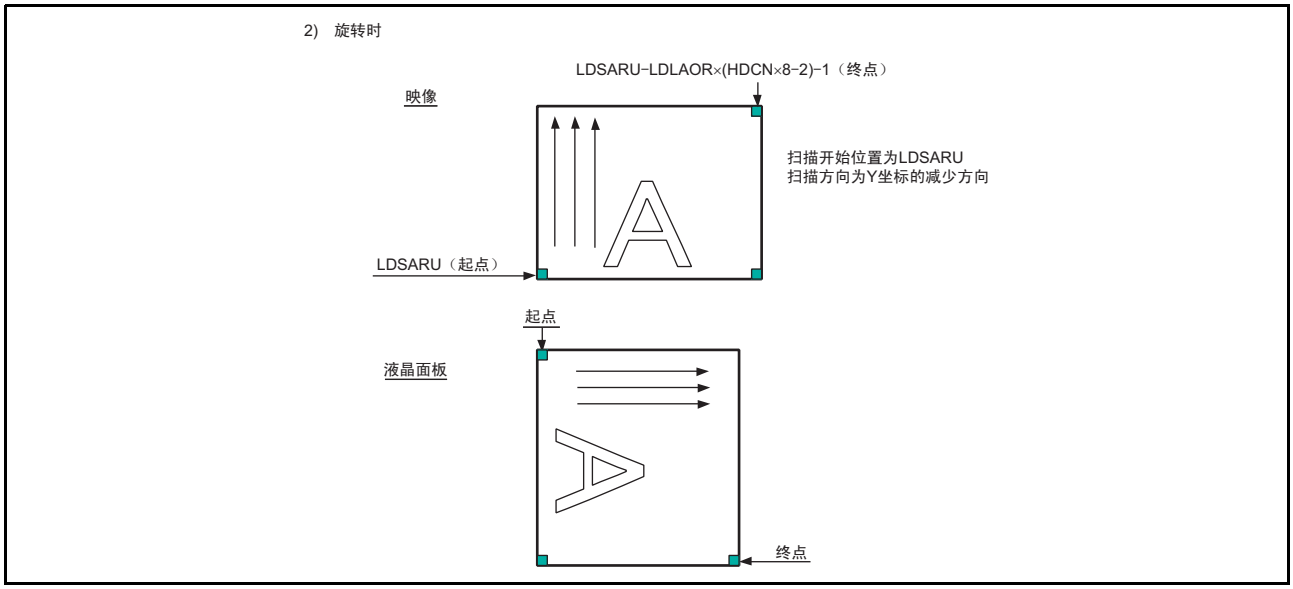


图 26.9　硬件旋转运行 （旋转时）

26.5 时钟和 LCD 数据信号例子

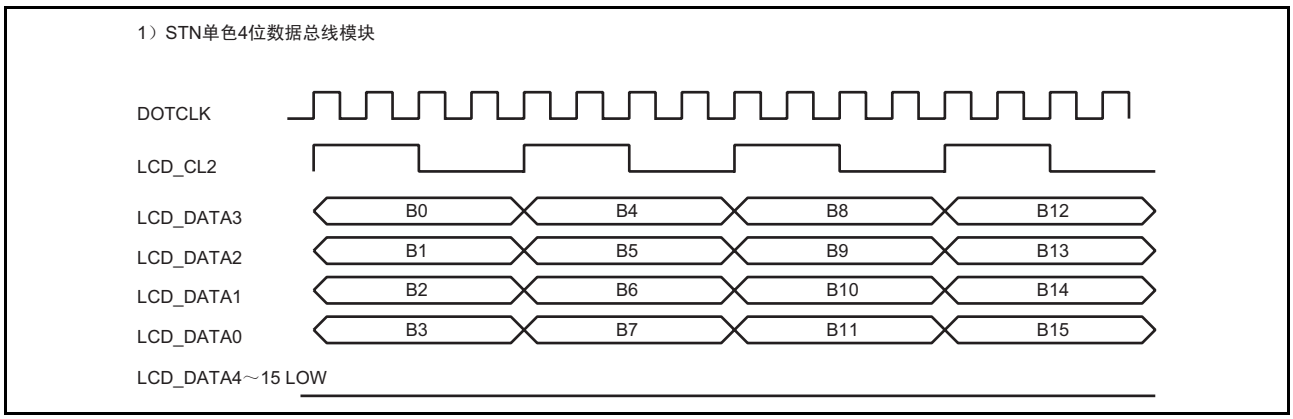


图 26.10 时钟和 LCD 数据信号例子 （STN 单色 4 位数据总线模块）

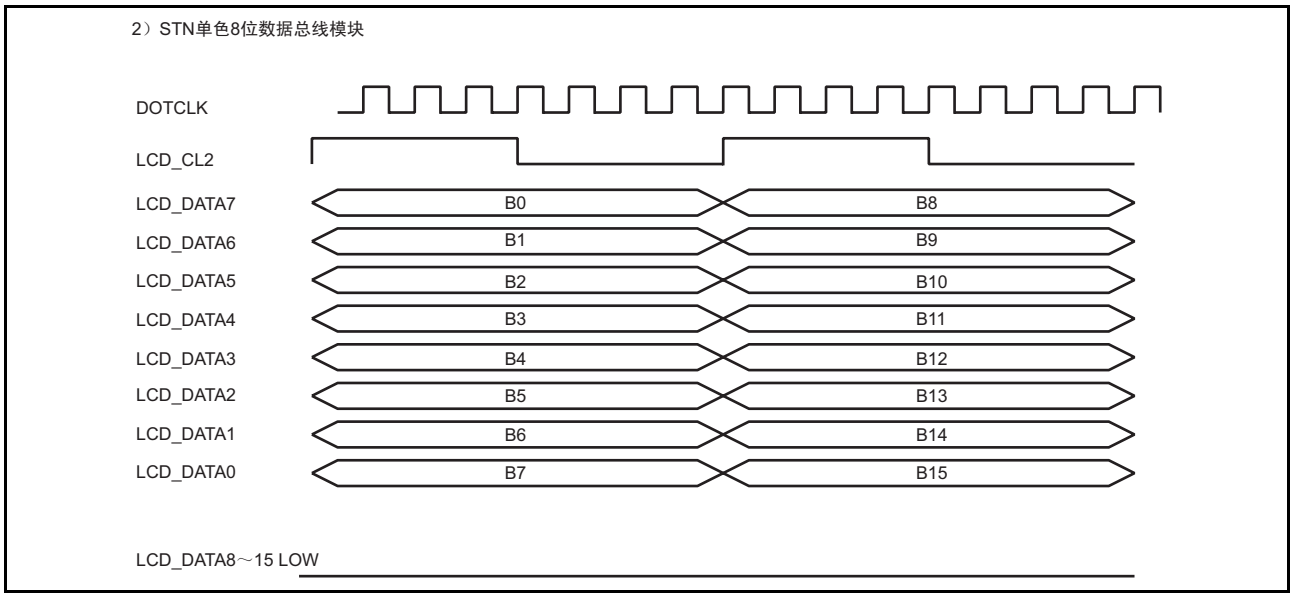


图 26.11 时钟和 LCD 数据信号例子 （STN 单色 8 位数据总线模块）

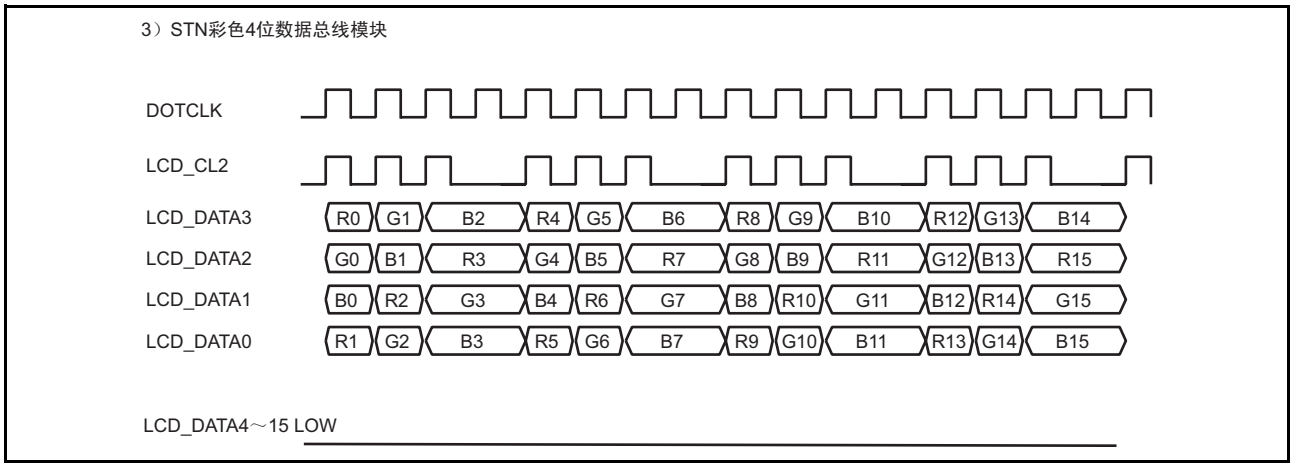


图 26.12 时钟和 LCD 数据信号例子 （STN 彩色 4 位数据总线模块）

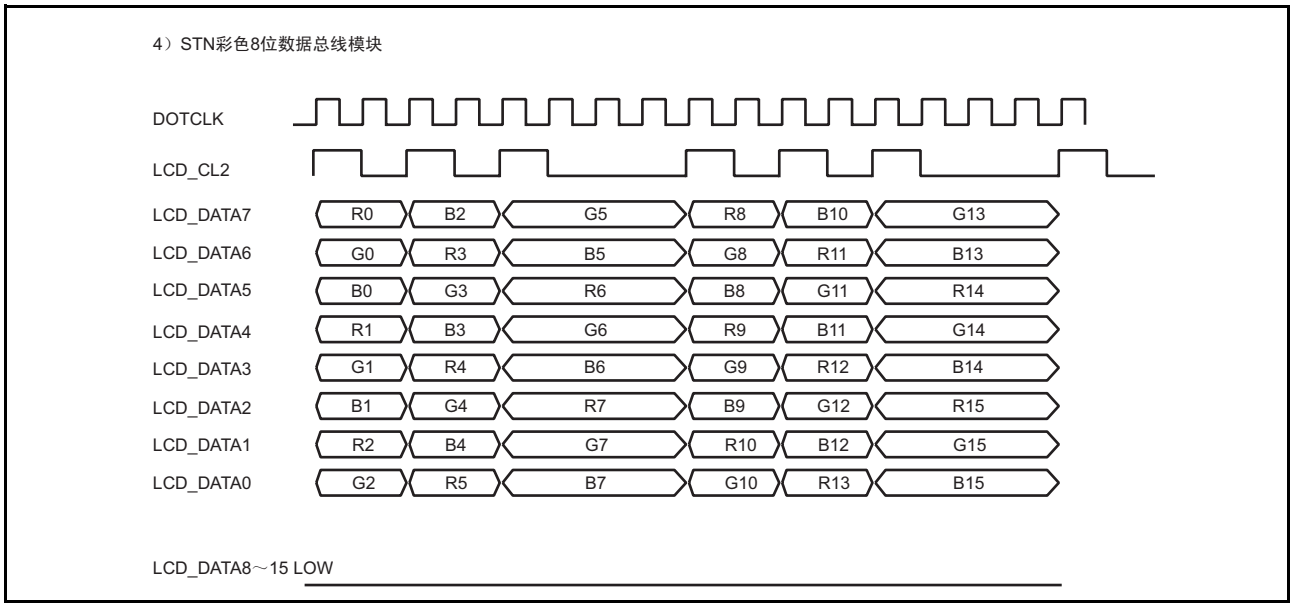


图 26.13 时钟和 LCD 数据信号例子 （STN 彩色 8 位数据总线模块）

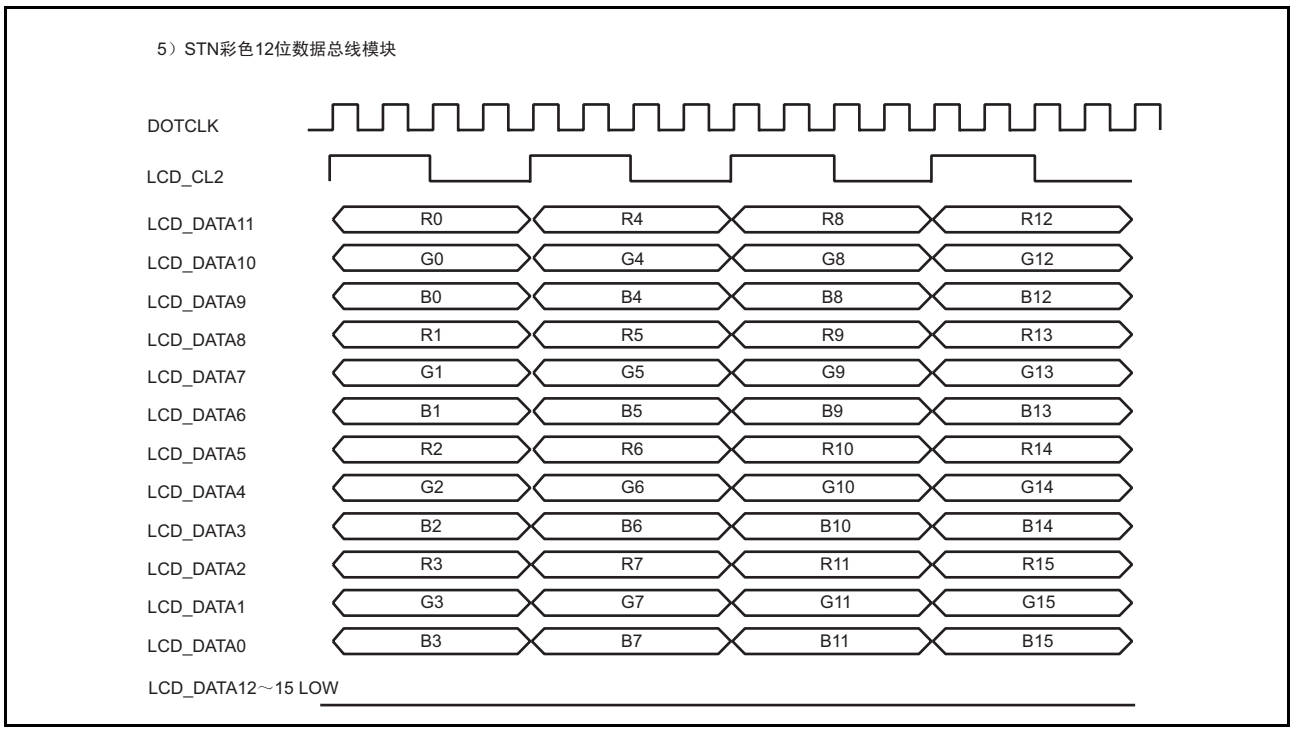


图 26.14 时钟和 LCD 数据信号例子 （STN 彩色 12 位数据总线模块）

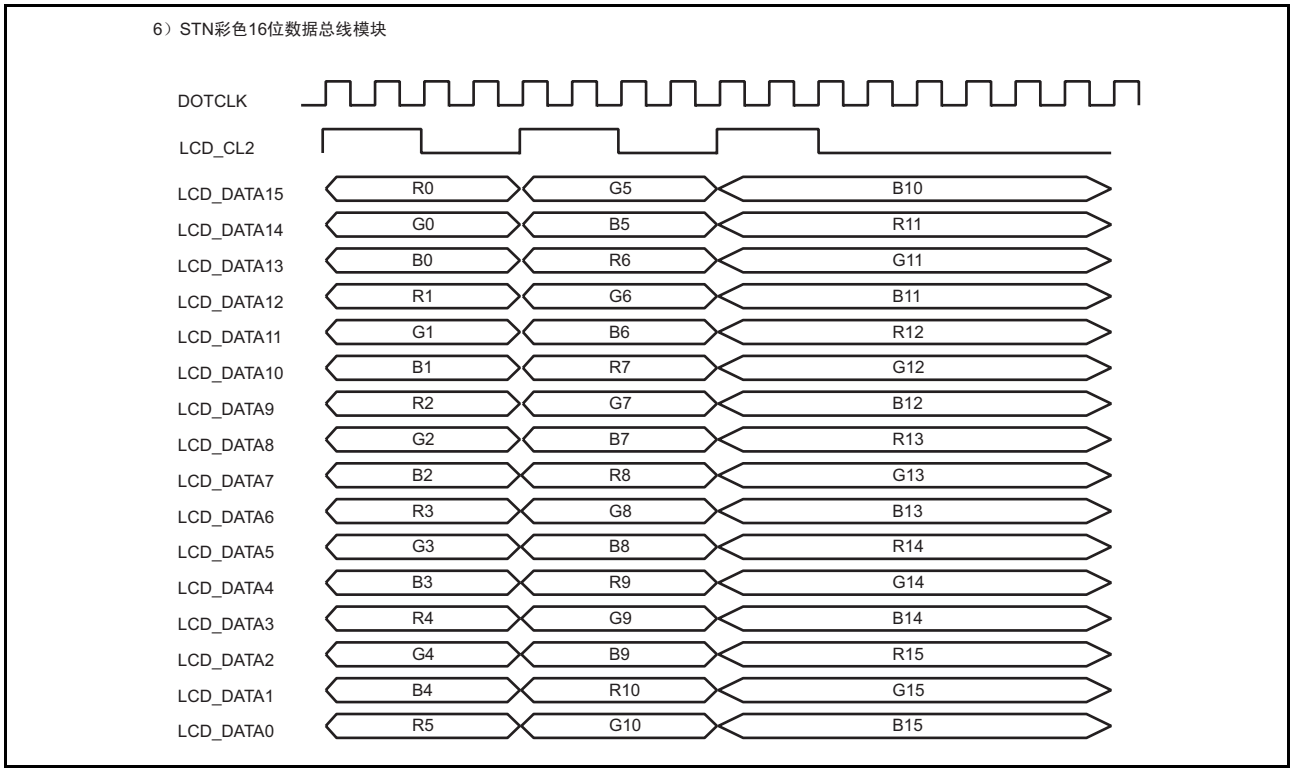


图 26.15　时钟和 LCD 数据信号例子 （STN 彩色 16 位数据总线模块）

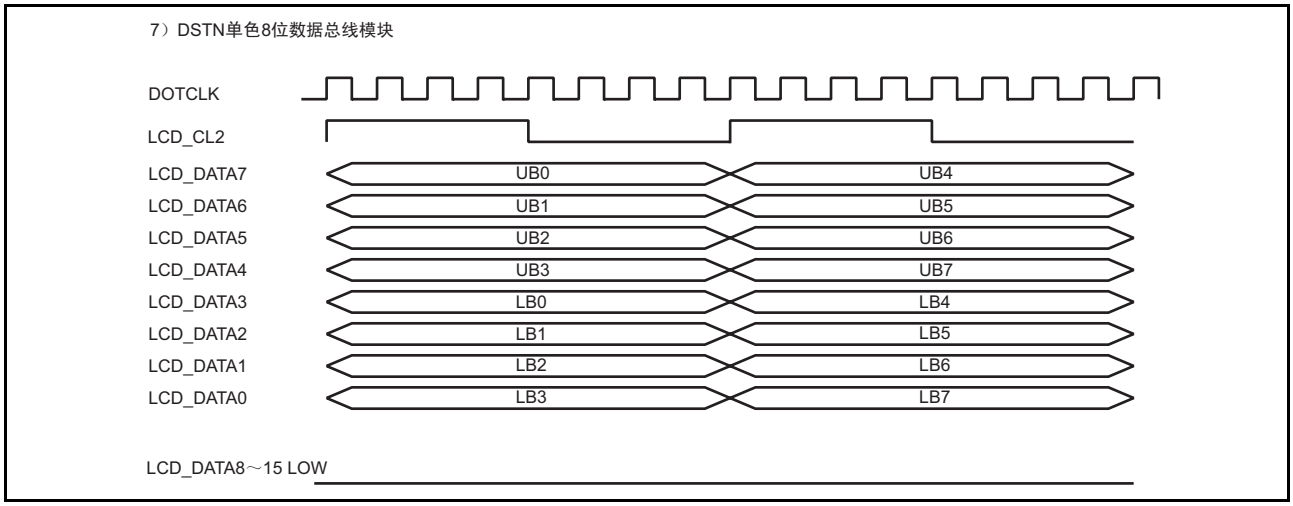


图 26.16　时钟和 LCD 数据信号例子 （DSTN 单色 8 位数据总线模块）

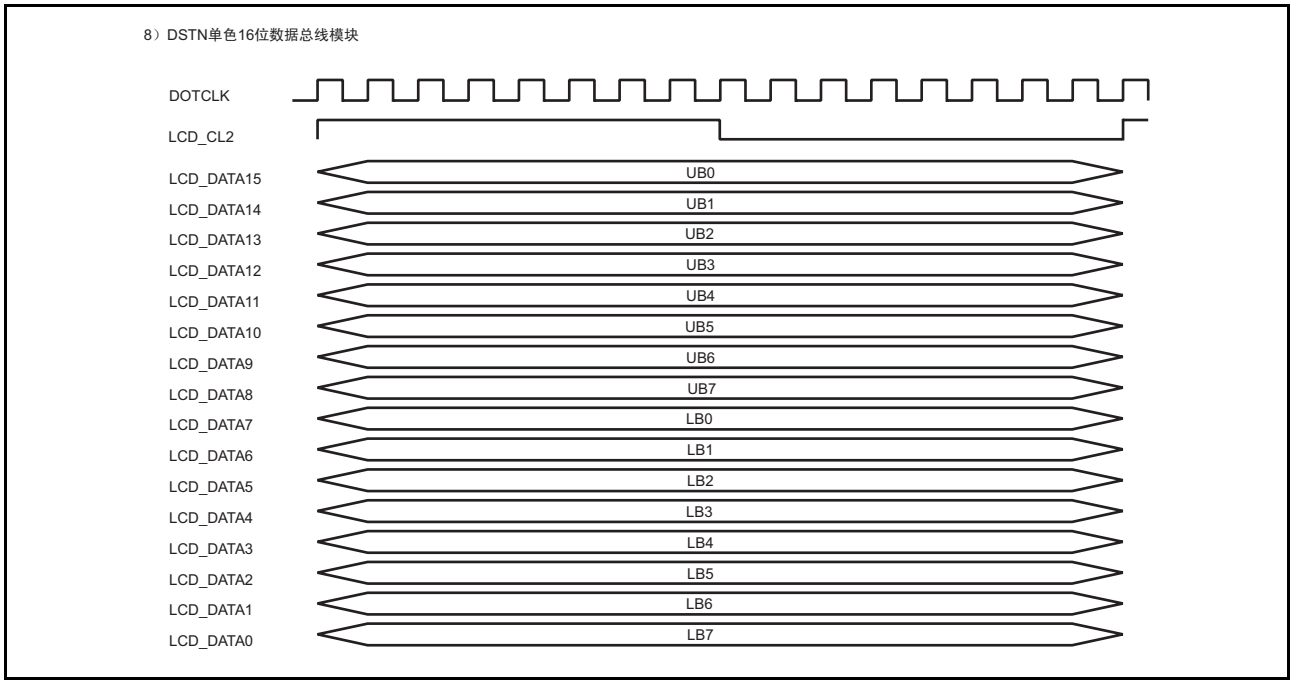


图 26.17 时钟和 LCD 数据信号例子 （DSTN 单色 16 位数据总线模块）

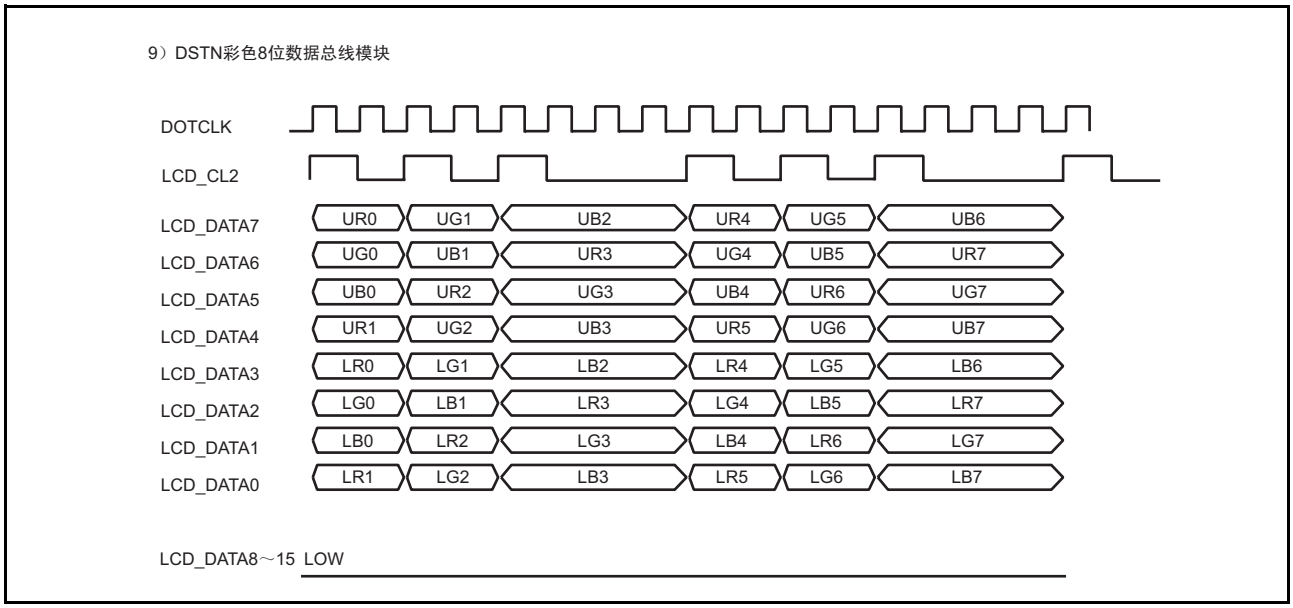


图 26.18 时钟和 LCD 数据信号例子 （DSTN 彩色 8 位数据总线模块）

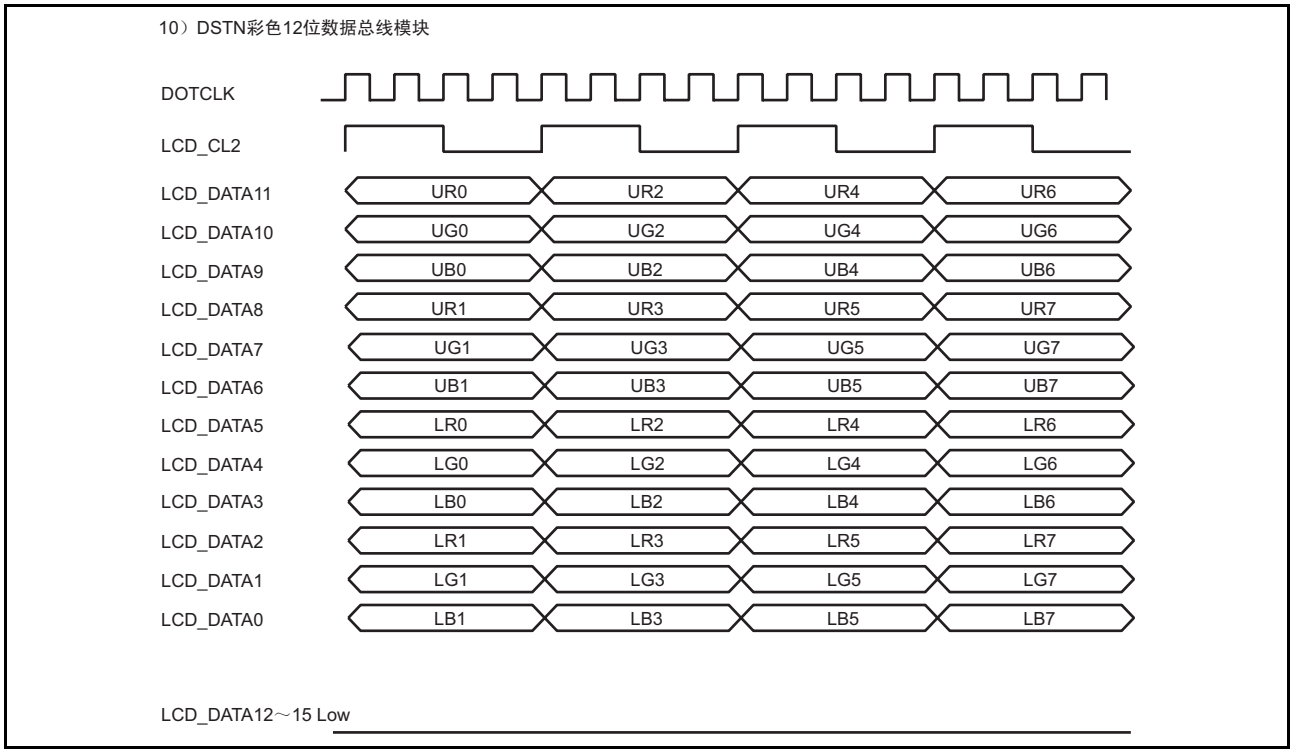


图 26.19　时钟和 LCD 数据信号例子 （DSTN 彩色 12 位数据总线模块）

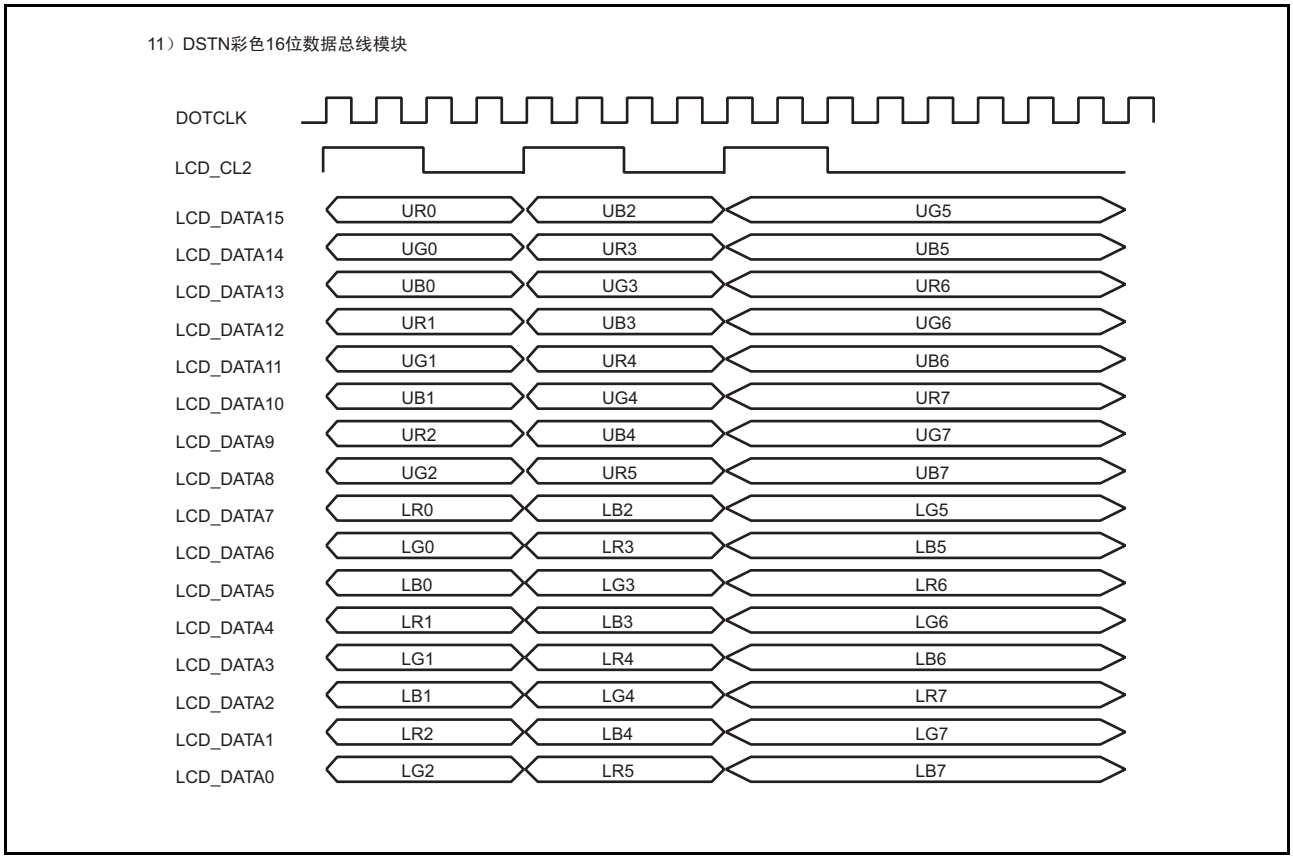


图 26.20　时钟和 LCD 数据信号例子 （DSTN 彩色 16 位数据总线模块）

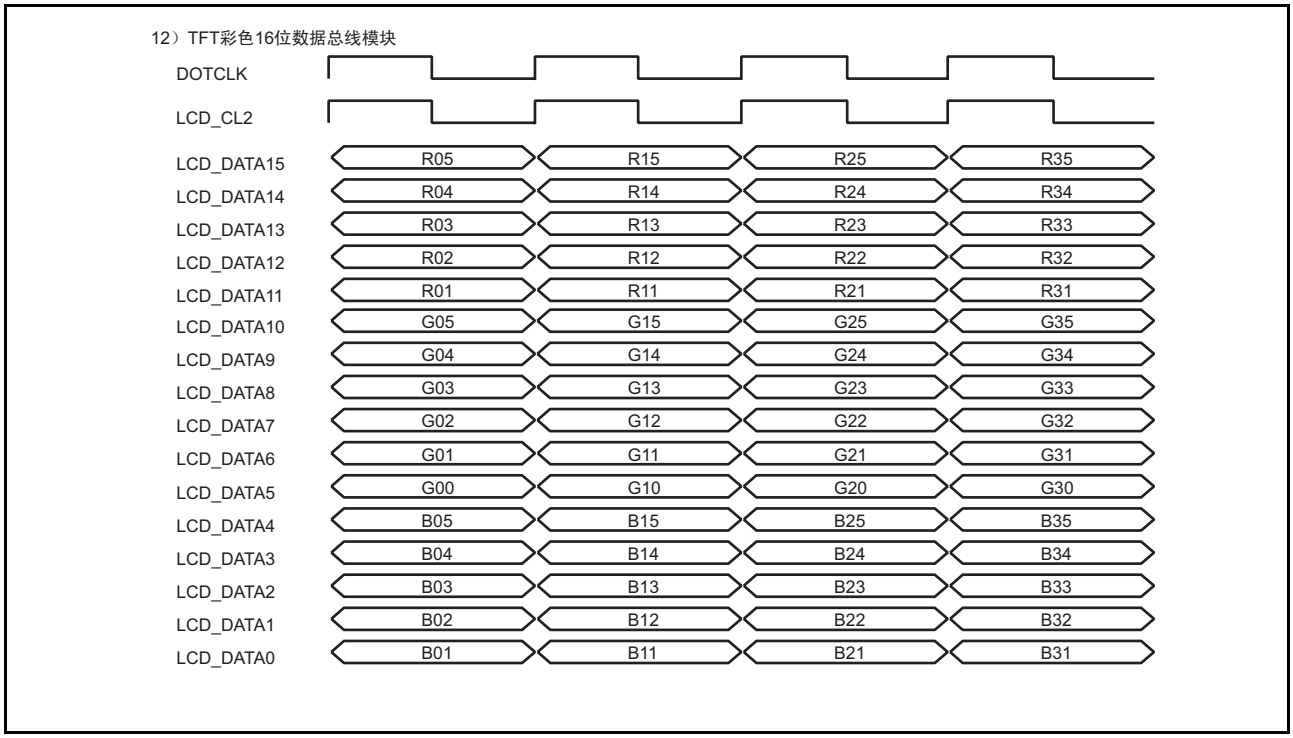


图 26.21 时钟和 LCD 数据信号例子 （TFT 彩色 16 位数据总线模块）

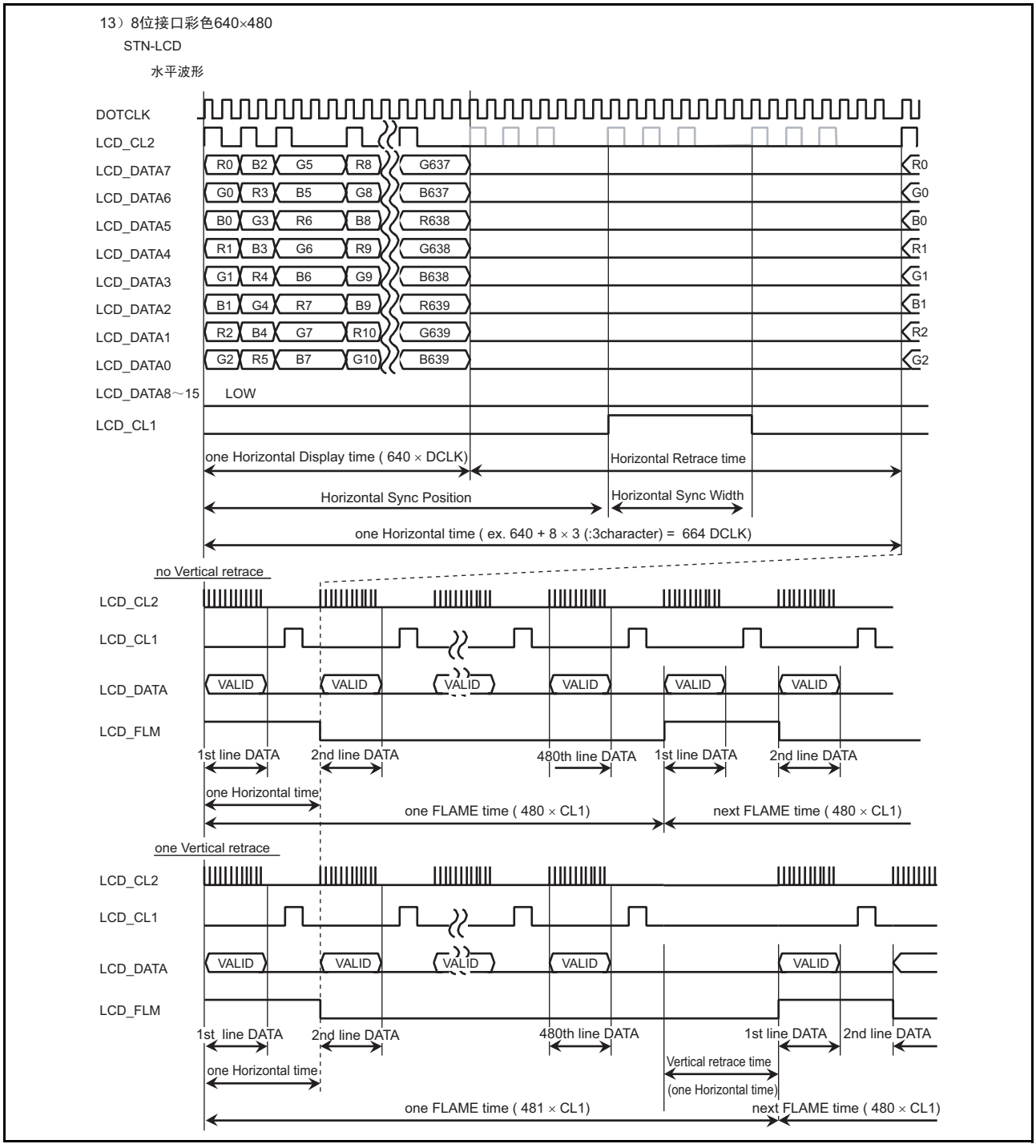


图 26.22 时钟和 LCD 数据信号例子 （8 位接口彩色 640×480）

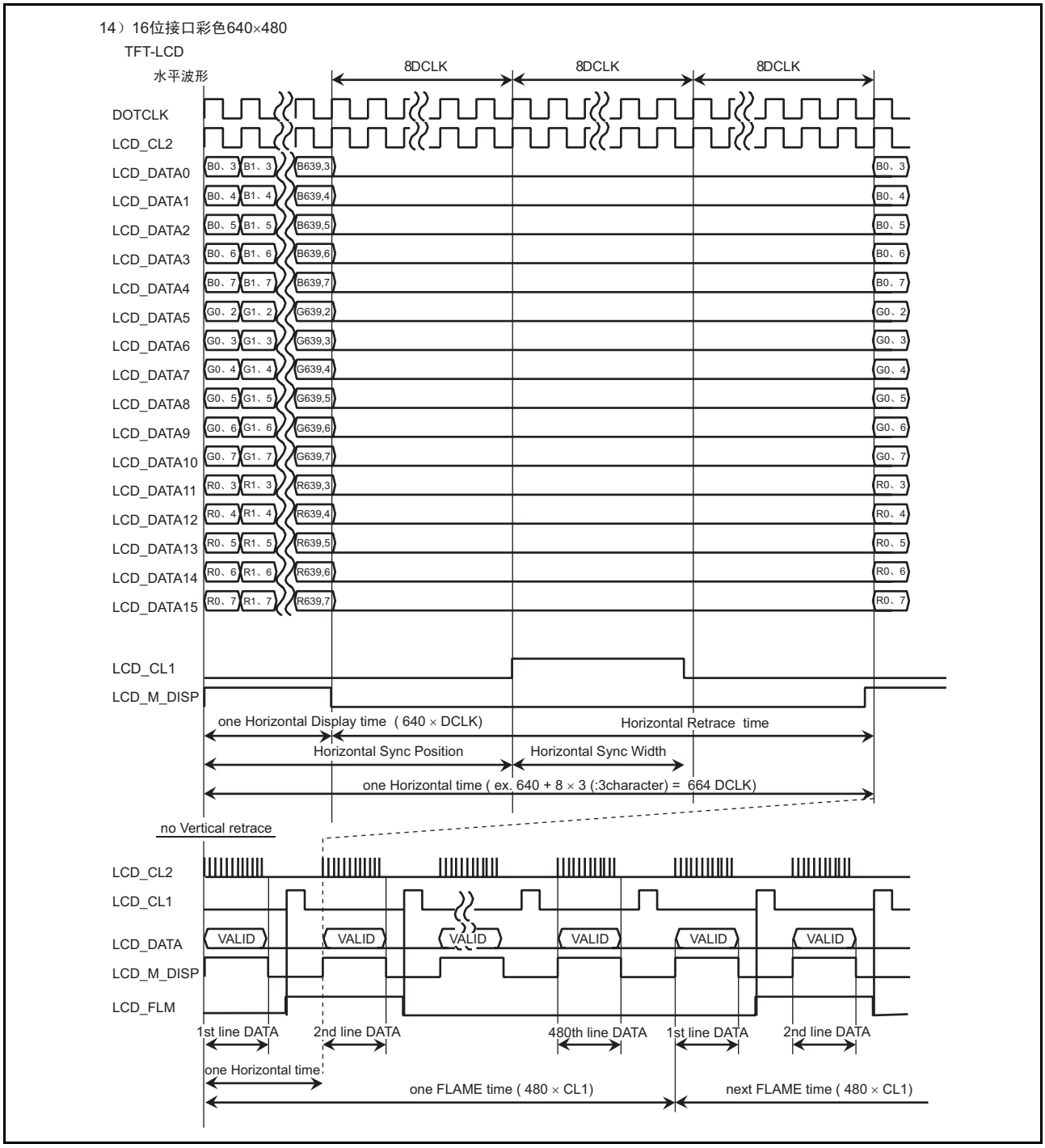


图 26.23 时钟和 LCD 数据信号例子 （16 位接口彩色 640×480）

26.6 使用时的注意事项

26.6.1 关于存取用于存储显示数据的 VRAM （区域 3 的同步 DRAM）时的停止步骤

存取用于存储显示数据的 VRAM （区域 3 的同步 DRAM）时的停止步骤，请按照以下的流程。

- 存取用于存储显示数据的VRAM时的停止顺序
  1. 确认现在的状态为LDPMMR的LPS1和LPS0位为1。
  2. LDCNTR的DON位设定为0 （显示关闭模式）。
  3. 确认LDPMMR的LPS1和LPS0位为0。
  4. 等待1帧的显示时间。

在用于存储显示数据的 VRAM （区域 3 的同步 DRAM）自刷新前，或进入待机模式、模块待机等前，需要该停止顺序。

第 27 章　采样频率转换器 （SRC）

采样频率转换器 （SRC）为对 WMA/MP3/AAC 等各种解码器生成数据的采样频率进行转换的模块。

27.1　特点

- 数据格式：16位 （立体声/单声道）
- 采样频率
  输入：可从8kHz、 11.025kHz、 12kHz、 16kHz、 22.05kHz、 24kHz、 32kHz、 44.1kHz、 48kHz中选择
  输出：可从44.1kHz、 48kHz中选择
- 处理性能：1个采样的输出间隔最大约为8μs （Pϕ=33MHz）
- SNR：大于等于80db
- 中断源：3种
  输入数据FIFO空、输出数据FIFO满、输出数据FIFO重写
- DMA传送源：2种
  输入数据FIFO空、输出数据FIFO满
- 模块待机模式
  不使用时，由于停止向SRC提供时钟，可降低功耗

SRC 的框图如图 27.1 所示。

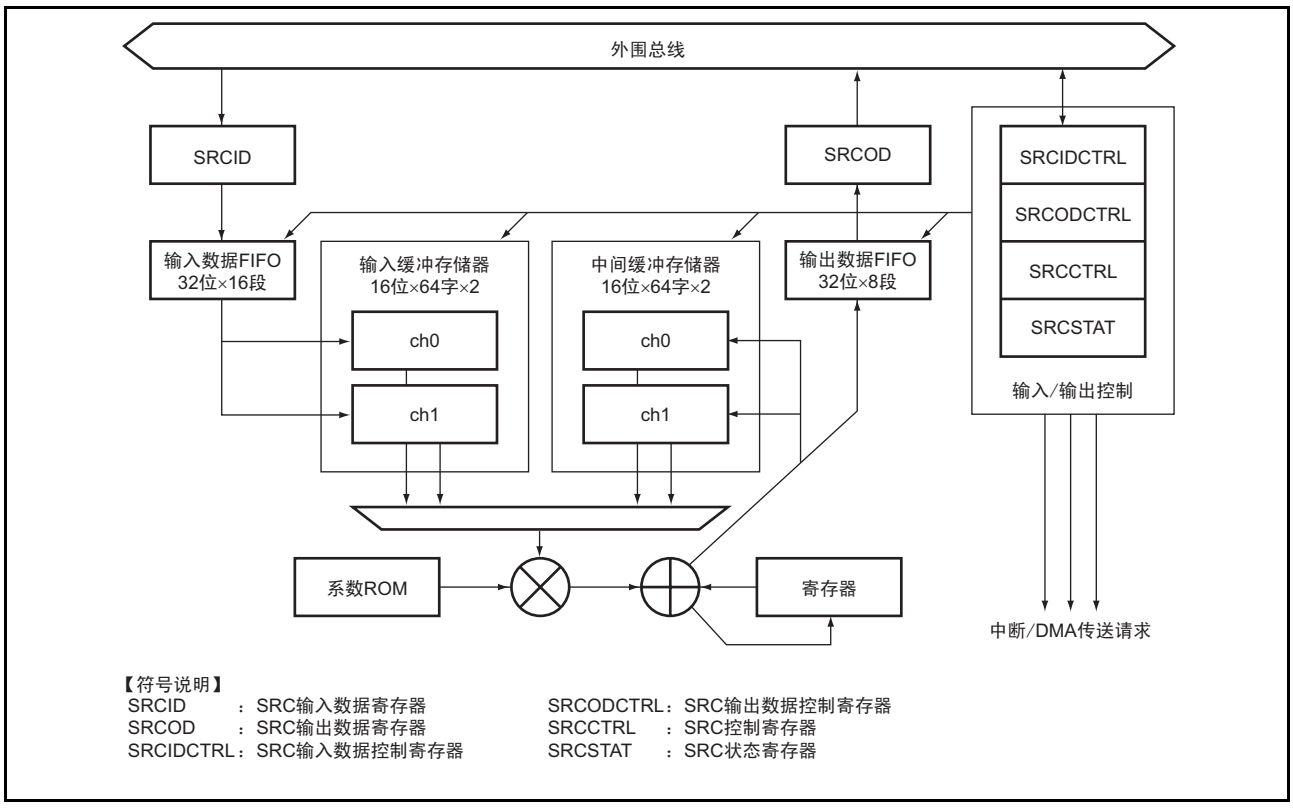


图 27.1　SRC 的框图

27.2　寄存器的说明

SRC 中有以下的寄存器。

表 27.1　寄存器结构

寄存器名称	简称	R/W	初始值	地址	存取长度
SRC 输入数据寄存器	SRCID	R/W	H'00000000	H'FFFF4000	16、32
SRC 输出数据寄存器	SRCOD	R	H'00000000	H'FFFF4004	16、32
SRC 输入数据控制寄存器	SRCIDCTRL	R/W	H'0000	H'FFFF4008	16
SRC 输出数据控制寄存器	SRCODCTRL	R/W	H'0000	H'FFFF400A	16
SRC 控制寄存器	SRCCTRL	R/W	H'0000	H'FFFF400C	16
SRC 状态寄存器	SRCSTAT	R/(W)*	H'0002	H'FFFF400E	16

【注】 *　bit15 ～ 3 只可读取。另外， bit2 ～ 0 读 1 后只能写 0。

27.2.1　SRC 输入数据寄存器 （SRCID）

SRCID 为 32 位可读 / 写寄存器，用于采样频率转换前的输入数据。所有位通常读出 0。向 SRCID 写入的数据保存在 16 段的输入数据 FIFO 中。输入数据 FIFO 的数据数为 16 时，向 SRCID 的写入无效。为立体声数据时， bit31 ～ 16 保存 ch0 的数据， bit15 ～ 0 保存 ch1 的数据。为单声道数据时， bit31 ～ 16 保存的数据有效，bit15 ～ 0 保存的数据无效。

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

为转换处理对象的数据，根据 SRCIDCTRL 的 IED 位的设定值，队列有所不同。 SRCIDCTRL 的 IED 位的设定值和数据队列的关系如表 27.2 所示。

表 27.2　转换数据的队列

IED	ch0[15:8]	ch0[7:0]	ch1[15:8]	ch1[7:0]
0	SRCID[31:24]	SRCID[23:16]	SRCID[15:8]	SRCID[7:0]
1	SRCID[23:16]	SRCID[31:24]	SRCID[7:0]	SRCID[15:8]

27.2.2　SRC 输出数据寄存器 （SRCOD）

SRCOD 为 32 位可读寄存器，用于输出采样频率转换后的数据。能从 SRCOD 读出保存在 8 段输出数据 FIFO 中的数据。

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
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

SRCOD 的内容根据 SRCODCTRL 的 OCH、OED 位的设定值，队列会有所不同。SRCODCTRL 的 OCH、OED 位的设定值和保存在 SRCOD 中的数据的队列关系如**表 27.3** 所示。

表 27.3　SRCOD 的队列

OCH	OED	SRCOD[31:24]	SRCOD[23:16]	SRCOD[15:8]	SRCOD[7:0]
0	0	ch0[15:8]	ch0[7:0]	ch1[15:8]*2	ch1[7:0]*2
	1	ch0[7:0]	ch0[15:8]	ch1[7:0]*2	ch1[15:8]*2
1*1	0	ch1[15:8]	ch1[7:0]	ch0[15:8]	ch0[7:0]
	1	ch1[7:0]	ch1[15:8]	ch0[7:0]	ch0[15:8]

【注】　*1　处理单声道数据时，请勿设定。

*2　处理单声道数据时，为无效数据。

27.2.3　SRC 输入数据控制寄存器 （SRCIDCTRL）

SRCIDCTRL 为 16 位可读 / 写寄存器，设定输入数据的端法格式、中断请求的允许 / 禁止及触发数据数。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	IED	IEN	-	-	-	-	-	-	IFTRG[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/W	R/W	R	R	R	R	R	R	R/W	R/W

位	位名称	初始值	R/W	说　明
15 ～ 10	—	均为 0	R	保留位 读写的值总是 0。
9	IED	0	R/W	指定输入数据端法 指定输入数据的端法格式。 0：大端法 1：小端法
8	IEN	0	R/W	输入数据空中断允许 输入 FIFO 的数据数为小于等于 IFTRG[1:0] 位设定的触发数， SRC 状态寄存器 （SRCSTAT）的 IINT 位置 1 时，允许 / 禁止产生输入数据空中断请求。 0：禁止输入数据空中断请求 1：允许输入数据空中断请求
7 ～ 2	—	均为 0	R	保留位 读写的值总是 0。
1、 0	IFTRG[1:0]	00	R/W	输入 FIFO 数据数触发 指定设置 SRC 状态寄存器 （SRCSTAT）的 IINT 位的条件。保存在输入 FIFO 的输入数据数小于等于以下所示的设定触发数时，IINT 位置 1。 00：0 01：4 10：8 11：12

27.2.4　SRC 输出数据控制寄存器 （SRCODCTRL）

SRCODCTRL 为 16 位可读 / 写寄存器，设定输出数据的通道交换、端法格式、中断请求的允许 / 禁止及触发数据数。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	OCH	OED	OEN	-	-	-	-	-	-	OFTRG[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/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

位	位名称	初始值	R/W	说　明
15 ～ 11	—	均为 0	R	保留位 读写的值总是 0。
10	OCH	0	R/W	输出数据通道转换 指定 SRC 输出数据寄存器（SRCOD）的通道转换。转换为单声道数据时，请勿置 1。 0：不转换通道 （与输入数据的顺序相同） 1：转换通道 （与输入数据的顺序相反）
9	OED	0	R/W	指定输出数据端法 指定输出数据的端法格式。 0：大端法 1：小端法
8	OEN	0	R/W	输出数据满中断允许 输出 FIFO 的数据数为大于等于 OFTRG[1:0] 位设定的触发数，SRC 状态寄存器 （SRCSTAT）的 OINT 位置位时，允许 / 禁止产生输出数据满中断请求。 0：禁止输出数据满中断请求 1：允许输出数据满中断请求
7 ～ 2	—	均为 0	R	保留位 读写的值总是 0。
1、0	OFTRG[1:0]	00	R/W	输出 FIFO 数据数触发 指定 SRC 状态寄存器 （SRCSTAT）的 OINT 位置位条件。保存在输出 FIFO 的输出数据数大于等于以下所示的设定触发数时， OINT 位置 1。 00：1 01：2 10：4 11：6

27.2.5　SRC 控制寄存器 （SRCCTRL）

SRCCTRL 为 16 位可读 / 写寄存器，设定允许 / 禁止模块运行、允许 / 禁止中断请求、闪存处理、内部工作存储器的清除处理及输入 / 输出采样频率。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	SRCEN	-	EEN	FL	CL	IFS[3:0]				-	-	-	OFS
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	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W

位	位名称	初始值	R/W	说　明
15 ～ 13	—	均为 0	R	保留位 读写的值总是 0。
12	SRCEN	0	R/W	SRC 模块允许 设定允许 / 禁止 SRC 模块运行。如果 SRCEN=0 时写入 1，就清除内部工作存储器。 0：禁止 SRC 模块运行 1：允许 SRC 模块运行 【注】 SRCEN=1 时，请勿改变以下位的设定值。
11	—	0	R	保留位 读写的值总是 0。
10	EEN	0	R/W	输出 FIFO 重写中断允许 输出 FIFO 的数据数为 8 时结束下一个数据的转换处理，并在 SRC 状态寄存器（SRCSTAT）的 OVF 位置 1 时，允许 / 禁止产生输出 FIFO 重写中断请求。 0：禁止输出 FIFO 重写中断请求 1：允许输出 FIFO 重写中断请求
9	FL	0	R/W	内部工作存储器闪存 如果写入 1，就开始对保存在输入 FIFO、输入缓冲存储器及中间缓冲存储器的所有数据进行采样频率转换 （闪存处理）。读取时，总是读出 0。SRCEN=0 时，即使写入 1 也不进行闪存处理。 另外，在输入缓冲存储器的数据数小于 64 时，如果 FL 位写入 1，就得不到有效输出数据，因此不进行闪存处理，而清除内部工作存储器。
8	CL	0	R/W	清除内部工作存储器 如果写入 1，就清除输入 FIFO、输出 FIFO、输入缓冲存储器、中间缓冲存储器及累加器。读取时，总是读出0。即使SRCEN=0，写入1时也清除。

寄存器	位	位名
SRCIDCTRL	9	IED
SRCODCTRL	10、9	OCH、OED
SRCCTRL	7～4、0	IFS[3:0]、OFS

位	位名称	初始值	R/W	说　明
7 ～ 4	IFS[3:0]	0000	R/W	输入采样频率 设定输入采样频率。 0000：8.0kHz 0001：11.025kHz 0010：12.0kHz 0011：禁止设定 0100：16.0kHz 0101：22.05kHz 0110：24.0kHz 0111：禁止设定 1000：32.0kHz 1001：44.1kHz 1010：48.0kHz 1011：禁止设定 1100：禁止设定 1101：禁止设定 1110：禁止设定 1111：禁止设定
3 ～ 1	—	均为 0	R	保留位 读写的值总是 0。
0	OFS	0	R/W	输出采样频率 设定输出采样频率。 0：44.1kHz 1：48.0kHz

作为转换结果得到的输出数据数可通过公式（A）或公式（B）求出。IFS、OFS 位的设定值和适用的计算公式的关系如**表 27.4** 所示。

$$\text{输出数据数} = \text{输入数据数} \times \frac{\text{输出采样率}}{\text{输入采样率}} \quad \cdots \text{（A）}$$

$$\text{输出数据数} = \text{输入数据数} \times \frac{\text{输出采样率}}{\text{输入采样率}} - 1 \quad \cdots \text{（B）}$$

表 27.4　采样频率设定和输出数据数的关系

OFS 设定值（输出采样频率 [kHz]）	IFS 设定值 （输入采样频率 [kHz]）								
	0000 (8.0)	0001 (11.025)	0010 (12.0)	0100 (16.0)	0101 (22.05)	0110 (24.0)	1000 (32.0)	1001 (44.1)	1010 (48.0)
0(44.1)	B	A	A	B	A	A	B	—	A
1(48.0)	B	B	A	B	B	A	B	B	—

27.2.6　SRC 状态寄存器 （SRCSTAT）

SRCSTAT 为 16 位可读 / 写寄存器，显示输出 FIFO 及输入 FIFO 的数据数、各中断源的产生状态及进行闪存处理的状态。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OFDN[3:0]					IFDN[4:0]				-	-	FLF	-	OVF	IINT	OINT
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W：	R	R	R	R	R	R	R	R	R	R	R	R	R	R/(W)*	R/ (W)*	R/(W)*

【注】* 读1后只能写0。

位	位名称	初始值	R/W	说　明
15 ～ 12	OFDN[3:0]	0000	R	输出 FIFO 数据计数 表示保存在输出 FIFO 的数据数。
11 ～ 7	IFDN[4:0]	00000	R	输入 FIFO 数据计数 表示保存在输入 FIFO 的数据数。
6、5	—	均为 0	R	保留位 读写的值总是 0。
4	FLF	0	R	闪存处理状态标志 表示正在进行闪存处理。 ［清除条件］ • 结束闪存处理时 • SRCCTRL 的 CL 位写入 1 时 • 当 SRCCTRL 的 SRCEN 位为 0，在 SRCEN 写入 1 时 ［置位条件］ • SRCCTRL 的 FL 位已写入 1 时
3	—	0	R	保留位 读写的值总是 0。
2	OVF	0	R/(W)*	输出 FIFO 重写中断请求标志 表示输出 FIFO 的数据数为 8 时，已结束下个数据的转换处理。清除 OVF 标志前，停止转换处理。 ［清除条件］ • 在 OVF=1 的状态下，读 OVF 后，在 OVF 已写入 0 时 • SRCCTRL 的 CL 位已写入 1 时 • 当 SRCCTRL 的 SRCEN 位为 0，在 SRCEN 已写入 1 时 ［置位条件］ • 输出 FIFO 的数据数为 8、已结束下一个数据的转换处理时

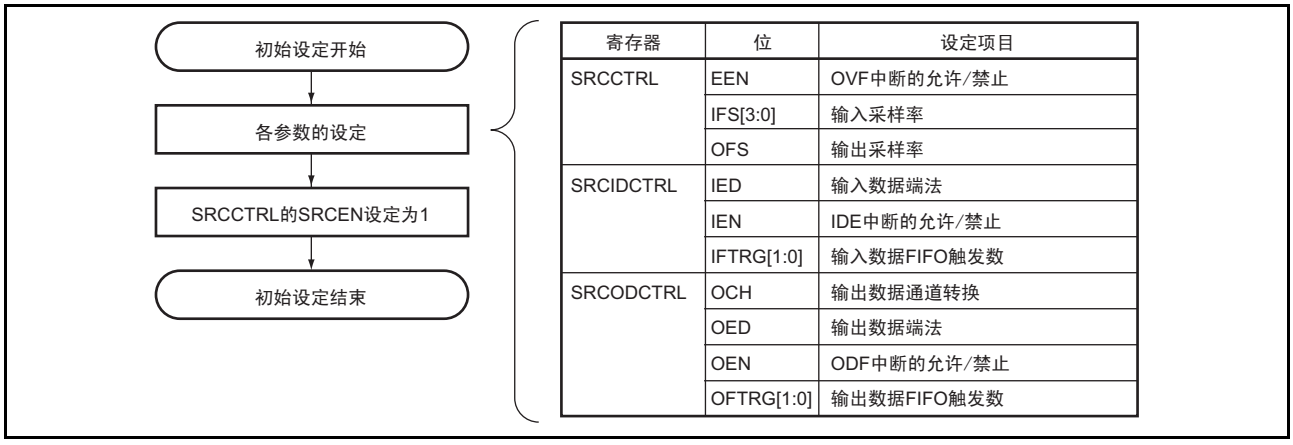
位	位名称	初始值	R/W	说　明
1	IINT	1	R/(W)*	输入 FIFO 空中断请求标志 表示保存在输入 FIFO 的数据数已变为小于等于 SRC 输入数据控制寄存器 （SRCIDCTRL）的 IFTRG[1:0] 位设定的触发数。 ［清除条件］ • 在 IINT=1 时，读出 IINT 后，在 IINT 已写入 0 时 • 通过 DMA 传送，输入 FIFO 的数据数超过设定的触发数时 ［置位条件］ • 保存在输入 FIFO 的数据数已变为小于等于设定的触发数时 • SRCCTRL 的 CL 位已写入 1 时 • 当 SRCCTRL 的 SRCEN 位为 0，在 SRCEN 已写入 1 时
0	OINT	0	R/(W)*	输出 FIFO 满中断请求标志 表示保存在输出 FIFO 的数据数已变为大于等于 SRC 输出数据控制寄存器 （SRCODCTRL）的 OFTRG[1:0] 位设定的触发数。 ［清除条件］ • 在 OINT=1 时，读出 OINT 后，在 OINT 已写入 0 时 • 通过 DMA 传送，输出 FIFO 的数据数小于设定的触发数时 • 在 SRCCTRL 的 CL 位已写入 1 时 • 当 SRCCTRL 的 SRCEN 位为 0，在 SRCEN 已写入 1 时 ［置位条件］ • 保存在输出 FIFO 的数据数大于等于设定的触发数时

【注】 * 读 1 后只能写 0。

27.3 运行说明

27.3.1 初始设定

初始设定的顺序如图 27.2 所示。



寄存器	位	设定项目
SRCCTRL	EEN	OVF中断的允许/禁止
	IFS[3:0]	输入采样率
	OFS	输出采样率
SRCIDCTRL	IED	输入数据端法
	IEN	IDE中断的允许/禁止
	IFTRG[1:0]	输入数据FIFO触发数
SRCODCTRL	OCH	输出数据通道转换
	OED	输出数据端法
	OEN	ODF中断的允许/禁止
	OFTRG[1:0]	输出数据FIFO触发数

图 27.2　初始设定顺序的例子

27.3.2　数据输入

数据输入的顺序如图 27.3 所示。

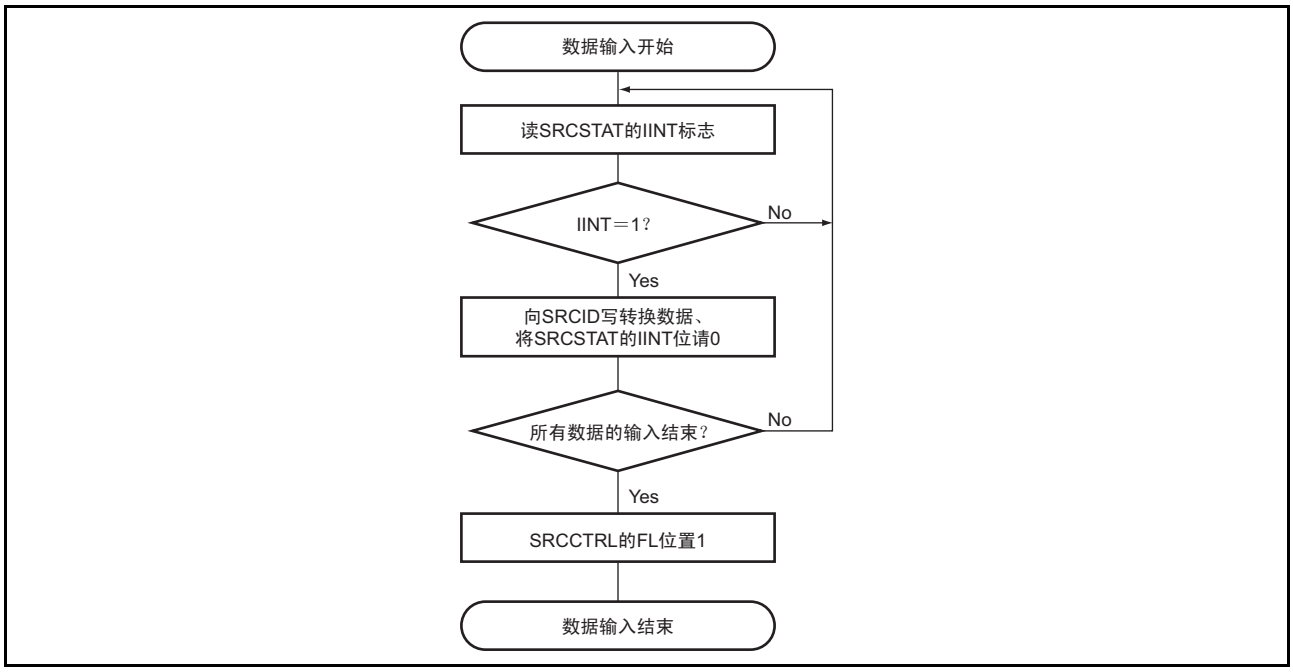


图 27.3　数据输入顺序的例子

（1） 使用对 CPU 的中断时

1. SRCIDCTRL的IEN位设定为1。
2. 如果SRCSTAT的IINT位设定为1，就产生IDE中断请求。在中断处理程序中，读出SRCSTAT的IINT位为1后，向SRCID写入数据，并向SRCSTAT的IINT位写入0。之后从中断处理程序返回。
3. 重复进行2.，如果结束所有的数据输入，就在SRCCTRL的FL位写入1。

（2） 通过中断启动 DMAC 时

1. 向SRC的IDEI分配DMAC的任意一个通道。
2. SRCIDCTRL的IEN位设定为1。
3. 如果SRCSTAT的IINT位设定为1，就产生IDE中断请求，从而启动DMAC。通过DMA传送，向SRCID写入数据。如果输入数据FIFO的数据数大于SRCIDCTRL的IFTRG[1:0]位设定的触发数时，就清除SRCSTAT的IINT位。
4. 重复进行3.，如果结束所有的数据输入，就在SRCCTRL的FL位写入1。

27.3.3　数据输出

数据输出的顺序如图 27.4 所示。

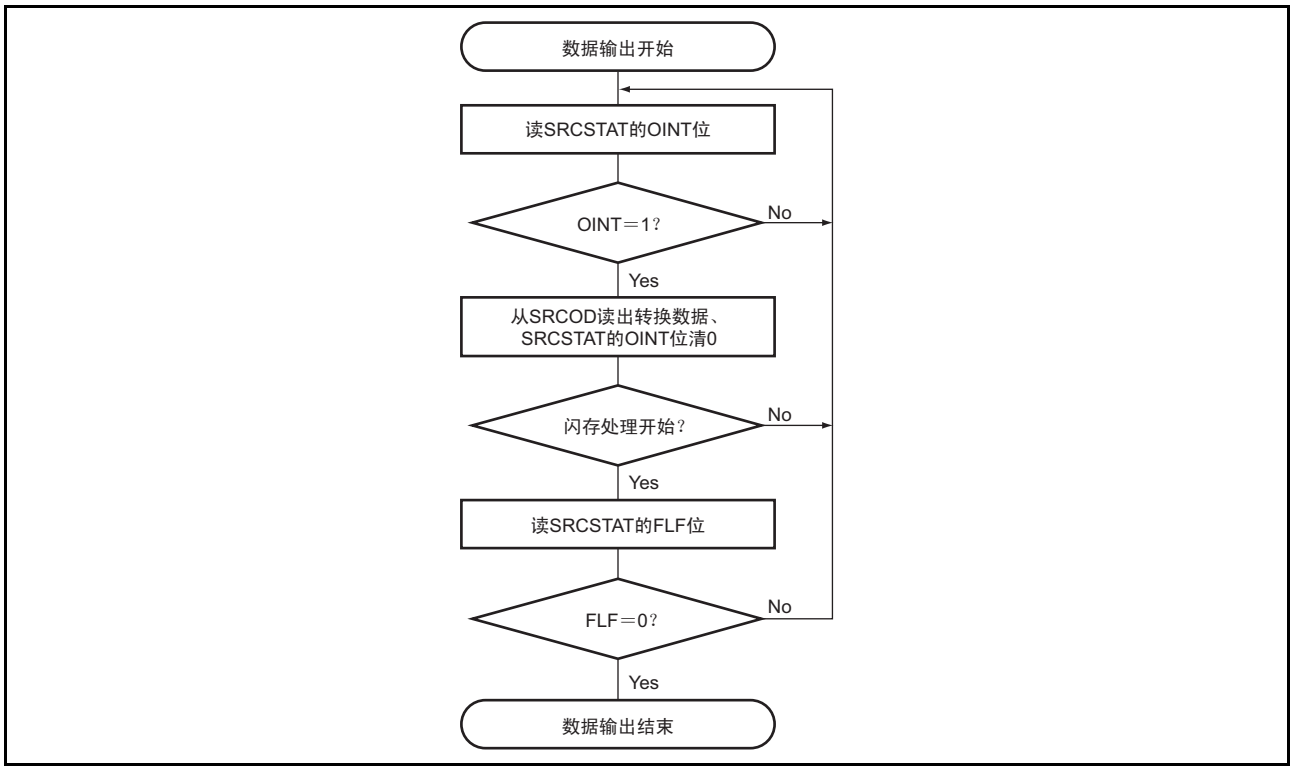


图 27.4　数据输出顺序的例子

（1） 使用对 CPU 的中断时

1. SRCODCTRL 的 OEN 位设定为 1。
2. 如果 SRCSTAT 的 OINT 位设定为 1，就产生 ODF 中断请求。在中断处理程序，读出 SRCSTAT 的 OINT 位为 1 后，从 SRCOD 读取数据，并向 SRCSTAT 的 OINT 位写入 0。之后从中断处理程序返回。
3. 开始闪存处理后，在读出 SRCSTAT 的 FLF 位为 0 前，重复进行 2. 中的操作。

（2） 通过中断启动 DMAC 时

1. 向 SRC 的 ODFI 分配 DMAC 的任意一个通道。
2. SRCODCTRL 的 OEN 位设定为 1。
3. 如果 SRCSTAT 的 OINT 位设定为 1，就产生 ODF 中断请求，从而启动 DMAC。通过 DMA 传送，从 SRCOD 读出数据。如果输出数据 FIFO 的数据数小于 SRCODCTRL 的 OFTRG[1:0] 位设定的触发数时，就清除 SRCSTAT 的 OINT 位。
4. 开始闪存处理后，在读出 SRCSTAT 的 FLF 位为 0 前，重复进行 3. 中的操作。

27.4　中断

SRC 的中断源有输入 FIFO 数据空（IDEI）、输出 FIFO 数据满（ODFI）及输出 FIFO 重写（OVF）3 种。中断的种类和产生条件如表 27.5 所示。

表 27.5　中断请求的种类和产生条件

中断请求	简称	产生条件	启动 DMAC
输入数据 FIFO 空	IDEI	(IINT=1) • (IEN=1) • (SRCEN=1)	可
输出数据 FIFO 满	ODFI	(OINT=1) • (OEN=1) • (SRCEN=1)	可
输出数据 FIFO 重写	OVF	(OVF=1) • (EEN=1) • (SRCEN=1)	不可

如果中断产生条件成立， CPU 就进行中断异常处理。请在中断异常处理程序中清除中断源标志。

通过设定 DMAC， IDEI 中断和 ODFI 中断可启动 DMAC。启动 DMAC 时，不向 CPU 产生来自 SRC 的中断。通过 DMA 传送，向 SRCID 写入数据。输入数据 FIFO 的数据数大于设定触发数时，清除 IINT。同样，从 SRCOD 读取数据，输出数据 FIFO 的数据数小于设定触发数时，清除 OINT。

27.5　使用时的注意事项

SRCCTRL 的 FL 位写 1 时，在 SRCSTAT 的 FLF 位置位前，需要外围时钟 （Pϕ）的 3 个周期。另外，不等到寄存器写入结束，CPU 便执行后续指令，因此通过紧接着向 SRCCTRL 写入指令之后的指令，不能读出 FLF 的置位状态。确认进行闪存处理的状态时，在向 SRCCTRL 写入指令后，请等待通过虚读 SRCCTRL 或 SRCSTAT 来置 FLF 位。

第 28 章 SD 主机接口 （SDHI）

本章内容仅在签署保密合同的前提下予以公开。
详细情况请与我公司销售负责人进行确认。

第 29 章　引脚功能控制器 （PFC）

引脚功能控制器 （PFC）由用于选择多路复用引脚的功能和其输入 / 输出方向的寄存器构成。该 LSI 的多路复用引脚如表 29.1 ～表 29.6 所示。

表 29.1　多路复用一览表 （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）
A	PA7 输入 （端口）	AN7 输入 （ADC）	DA1 输出 （DAC）
	PA6 输入 （端口）	AN6 输入 （ADC）	DA0 输出 （DAC）
	PA5 输入 （端口）	AN5 输入 （ADC）	—
	PA4 输入 （端口）	AN4 输入 （ADC）	—
	PA3 输入 （端口）	AN3 输入 （ADC）	—
	PA2 输入 （端口）	AN2 输入 （ADC）	—
	PA1 输入 （端口）	AN1 输入 （ADC）	—
	PA0 输入 （端口）	AN0 输入 （ADC）	—

【注】 通用输入、 A/D 转换器模拟输入、 D/A 转换器模拟输出的功能自动切换，因此在 PFC 没有用于设定的寄存器。

表 29.2　多路复用一览表 （端口 B）

设定寄存器	模式位 （PBnMD[1:0]）设定值			
	00	01	10	11
	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）
PBCRL4	PB12 输出 （端口）	$\overline{\text{WDTOVF}}$ 输出 （WDT）	$\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$ 输出 （INTC/BSC）	$\overline{\text{UBCTRG}}$ 输出 （UBC）
PBCRL3	PB11 输入 / 输出 （端口）	CTx1 输出 （RCAN-TL1）	IETxD 输出 （IEB）	—
	PB10 输入 / 输出 （端口）	CRx1 输入 （RCAN-TL1）	IERxD 输入 （IEB）	—
	PB9 输入 / 输出 （端口）	CTx0 输出 （RCAN-TL1）	CTx0&CTx1 输出 （RCAN-TL1）	—
	PB8 输入 / 输出 （端口）	CRx0 输入 （RCAN-TL1）	CRx0/CRx1 输入 （RCAN-TL1）	—
PBCRL2	PB7 输入 （端口）	SDA3 输入 / 输出 （IIC3）	PINT7 输入 （INTC）	IRQ7 输入 （INTC）
	PB6 输入 （端口）	SCL3 输入 / 输出 （IIC3）	PINT6 输入 （INTC）	IRQ6 输入 （INTC）
	PB5 输入 （端口）	SDA2 输入 / 输出 （IIC3）	PINT5 输入 （INTC）	IRQ5 输入 （INTC）
	PB4 输入 （端口）	SCL2 输入 / 输出 （IIC3）	PINT4 输入 （INTC）	IRQ4 输入 （INTC）
PBCRL1	PB3 输入 （端口）	SDA1 输入 / 输出 （IIC3）	PINT3 输入 （INTC）	IRQ3 输入 （INTC）
	PB2 输入 （端口）	SCL1 输入 / 输出 （IIC3）	PINT2 输入 （INTC）	IRQ2 输入 （INTC）
	PB1 输入 （端口）	SDA0 输入 / 输出 （IIC3）	PINT1 输入 （INTC）	IRQ1 输入 （INTC）
	PB0 输入 （端口）	SCL0 输入 / 输出 （IIC3）	PINT0 输入 （INTC）	IRQ0 输入 （INTC）

表 29.3　多路复用一览表 （端口 C）

设定寄存器	模式位 （PCnMD[1:0]）设定值			
	00	01	10	11
	功能 1 （相关模块）	功能 2 （相关模块）	功能 3 （相关模块）	—
PCCRL4	PC14 输入 / 输出 （端口）	$\overline{\text{WAIT}}$ 输入 （BSC）	—	—
	PC13 输入 / 输出 （端口）	RD$\overline{\text{WR}}$ 输出 （BSC）	—	—
	PC12 输入 / 输出 （端口）	CKE 输出 （BSC）	—	—
PCCRL3	PC11 输入 / 输出 （端口）	$\overline{\text{CASU}}$ 输出 （BSC）	$\overline{\text{BREQ}}$ 输入 （BSC）	—
	PC10 输入 / 输出 （端口）	$\overline{\text{RASU}}$ 输出 （BSC）	$\overline{\text{BACK}}$ 输出 （BSC）	—
	PC9 输入 / 输出 （端口）	$\overline{\text{CASL}}$ 输出 （BSC）	—	—
	PC8 输入 / 输出 （端口）	$\overline{\text{RASL}}$ 输出 （BSC）	—	—
PCCRL2	PC7 输入 / 输出 （端口）	$\overline{\text{WE3}}$/DQMUU/$\overline{\text{AH}}$/$\overline{\text{ICIOWR}}$ 输出 （BSC）	—	—
	PC6 输入 / 输出 （端口）	$\overline{\text{WE2}}$/DQMUL/$\overline{\text{ICIORD}}$ 输出 （BSC）	—	—
	PC5 输入 / 输出 （端口）	$\overline{\text{WE1}}$/DQMLU/$\overline{\text{WE}}$ 输出 （BSC）	—	—
	PC4 输入 / 输出 （端口）	$\overline{\text{WE0}}$/DQMLL 输出 （BSC）	—	—
PCCRL1	PC3 输入 / 输出 （端口）	$\overline{\text{CS3}}$ 输出 （BSC）	—	—
	PC2 输入 / 输出 （端口）	$\overline{\text{CS2}}$ 输出 （BSC）	—	—
	PC1 输入 / 输出 （端口）	A1 输出 （地址）	—	—
	PC0 输入 / 输出 （端口）	A0 输出 （地址）	$\overline{\text{CS7}}$ 输出 （BSC）	—

表 29.4　多路复用一览表 （端口 D）

设定寄存器	模式位 （PDnMD[2:0]）设定值						
	000	001	010	011	100	101	110/111
	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）	—
PDCRL4	PD15 输入 / 输出 （端口）	D31 输入 / 输出 （数据）	PINT7 输入 （INTC）	SD_CD 输入 （SDHI）	$\overline{\text{ADTRG}}$ 输入 （ADC）	TIOC4D 输入 / 输出 （MTU2）	—
	PD14 输入 / 输出 （端口）	D30 输入 / 输出 （数据）	PINT6 输入 （INTC）	SD_WP 输入 （SDHI）	—	TIOC4C 输入 / 输出 （MTU2）	—
	PD13 输入 / 输出 （端口）	D29 输入 / 输出 （数据）	PINT5 输入 （INTC）	SD_D1 输入 / 输出 （SDHI）	TEND1 输出 （DMAC）	TIOC4B 输入 / 输出 （MTU2）	—
	PD12 输入 / 输出 （端口）	D28 输入 / 输出 （数据）	PINT4 输入 （INTC）	SD_D0 输入 / 输出 （SDHI）	DACK1 输出 （DMAC）	TIOC4A 输入 / 输出 （MTU2）	—
PDCRL3	PD11 输入 / 输出 （端口）	D27 输入 / 输出 （数据）	PINT3 输入 （INTC）	SD_CLK 输出 （SDHI）	DREQ1 输入 （DMAC）	TIOC3D 输入 / 输出 （MTU2）	—
	PD10 输入 / 输出 （端口）	D26 输入 / 输出 （数据）	PINT2 输入 （INTC）	SD_CMD 输入 / 输出 （SDHI）	TEND0 输出 （DMAC）	TIOC3C 输入 / 输出 （MTU2）	—
	PD9 输入 / 输出 （端口）	D25 输入 / 输出 （数据）	PINT1 输入 （INTC）	SD_D3 输入 / 输出 （SDHI）	DACK0 输出 （DMAC）	TIOC3B 输入 / 输出 （MTU2）	—
	PD8 输入 / 输出 （端口）	D24 输入 / 输出 （数据）	PINT0 输入 （INTC）	SD_D2 输入 / 输出 （SDHI）	DREQ0 输入 （DMAC）	TIOC3A 输入 / 输出 （MTU2）	—
PDCRL2	PD7 输入 / 输出 （端口）	D23 输入 / 输出 （数据）	IRQ7 输入 （INTC）	$\overline{\text{SCS1}}$ 输入 / 输出 （SSU）	TCLKD 输入 （MTU2）	TIOC2B 输入 / 输出 （MTU2）	—
	PD6 输入 / 输出 （端口）	D22 输入 / 输出 （数据）	IRQ6 输入 （INTC）	SSO1 输入 / 输出 （SSU）	TCLKC 输入 （MTU2）	TIOC2A 输入 / 输出 （MTU2）	—
	PD5 输入 / 输出 （端口）	D21 输入 / 输出 （数据）	IRQ5 输入 （INTC）	SSI1 输入 / 输出 （SSU）	TCLKB 输入 （MTU2）	TIOC1B 输入 / 输出 （MTU2）	—
	PD4 输入 / 输出 （端口）	D20 输入 / 输出 （数据）	IRQ4 输入 （INTC）	SSCK1 输入 / 输出 （SSU）	TCLKA 输入 （MTU2）	TIOC1A 输入 / 输出 （MTU2）	—
PDCRL1	PD3 输入 / 输出 （端口）	D19 输入 / 输出 （数据）	IRQ3 输入 （INTC）	$\overline{\text{SCS0}}$ 输入 / 输出 （SSU）	DACK3 输出 （DMAC）	TIOC0D 输入 / 输出 （MTU2）	—
	PD2 输入 / 输出 （端口）	D18 输入 / 输出 （数据）	IRQ2 输入 （INTC）	SSO0 输入 / 输出 （SSU）	DREQ3 输入 （DMAC）	TIOC0C 输入 / 输出 （MTU2）	—
	PD1 输入 / 输出 （端口）	D17 输入 / 输出 （数据）	IRQ1 输入 （INTC）	SSI0 输入 / 输出 （SSU）	DACK2 输出 （DMAC）	TIOC0B 输入 / 输出 （MTU2）	—
	PD0 输入 / 输出 （端口）	D16 输入 / 输出 （数据）	IRQ0 输入 （INTC）	SSCK0 输入 / 输出 （SSU）	DREQ2 输入 （DMAC）	TIOC0A 输入 / 输出 （MTU2）	—

表 29.5　多路复用一览表 （端口 E）

设定寄存器	模式位 （PEnMD[2:0]）设定值					
	000	001	010	011	100	101/110/111
	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）	—
PECRL4	PE15 输入 / 输出 （端口）	$\overline{\text{IOIS16}}$ 输入 （BSC）	—	$\overline{\text{RTS3}}$ 输入 / 输出 （SCIF）	—	—
	PE14 输入 / 输出 （端口）	$\overline{\text{CS1}}$ 输出 （BSC）	—	$\overline{\text{CTS3}}$ 输入 / 输出 （SCIF）	—	—
	PE13 输入 / 输出 （端口）	—	—	TxD3 输出 （SCIF）	—	—
	PE12 输入 / 输出 （端口）	—	—	RxD3 输入 （SCIF）	—	—
PECRL3	PE11 输入 / 输出 （端口）	$\overline{\text{CS6}}$/$\overline{\text{CE1B}}$ 输出 （BSC）	IRQ7 输入 （INTC）	—	TEND1 输出 （DMAC）	—
	PE10 输入 / 输出 （端口）	$\overline{\text{CE2B}}$ 输出 （BSC）	IRQ6 输入 （INTC）	—	TEND0 输出 （DMAC）	—
	PE9 输入 / 输出 （端口）	$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$ 输出 （BSC）	IRQ5 输入 （INTC）	SCK3 输入 / 输出 （SCIF）	—	—
	PE8 输入 / 输出 （端口）	$\overline{\text{CE2A}}$ 输出 （BSC）	IRQ4 输入 （INTC）	SCK2 输入 / 输出 （SCIF）	—	—
PECRL2	PE7 输入 / 输出 （端口）	$\overline{\text{FRAME}}$ 输出 （BSC）	IRQ3 输入 （INTC）	TxD2 输出 （SCIF）	DACK1 输出 （DMAC）	—
	PE6 输入 / 输出 （端口）	A25 输出 （地址）	IRQ2 输入 （INTC）	RxD2 输入 （SCIF）	DREQ1 输入 （DMAC）	—
	PE5 输入 / 输出 （端口）	A24 输出 （地址）	IRQ1 输入 （INTC）	TxD1 输出 （SCIF）	DACK0 输出 （DMAC）	—
	PE4 输入 / 输出 （端口）	A23 输出 （地址）	IRQ0 输入 （INTC）	RxD1 输入 （SCIF）	DREQ0 输入 （DMAC）	—
PECRL1	PE3 输入 / 输出 （端口）	A22 输出 （地址）	—	SCK1 输入 / 输出 （SCIF）	—	—
	PE2 输入 / 输出 （端口）	A21 输出 （地址）	—	SCK0 输入 / 输出 （SCIF）	—	—
	PE1 输入 / 输出 （端口）	$\overline{\text{CS4}}$ 输出 （BSC）	$\overline{\text{MRES}}$ 输入 （系统控制）	TxD0 输出 （SCIF）	—	—
	PE0 输入 / 输出 （端口）	$\overline{\text{BS}}$ 输出 （BSC）	—	RxD0 输入 （SCIF）	$\overline{\text{ADTRG}}$ 输入 （ADC）	—

表 29.6　多路复用一览表 （端口 F）

设定寄存器	模式位 （PFnMD[1:0]）设定值			
	00	01	10	11
	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）
PFCRH4	PF30 输入 / 输出 （端口）	AUDIO_CLK 输入 （SSI）	—	—
	PF29 输入 / 输出 （端口）	SSIDATA3 输入 / 输出 （SSI）	—	—
	PF28 输入 / 输出 （端口）	SSIWS3 输入 / 输出 （SSI）	—	—
PFCRH3	PF27 输入 / 输出 （端口）	SSISCK3 输入 / 输出 （SSI）	—	—
	PF26 输入 / 输出 （端口）	SSIDATA2 输入 / 输出 （SSI）	—	—
	PF25 输入 / 输出 （端口）	SSIWS2 输入 / 输出 （SSI）	—	—
	PF24 输入 / 输出 （端口）	SSISCK2 输入 / 输出 （SSI）	—	—
PFCRH2	PF23 输入 / 输出 （端口）	SSIDATA1 输入 / 输出 （SSI）	LCD_VEPWC 输出 （LCDC）	—
	PF22 输入 / 输出 （端口）	SSIWS1 输入 / 输出 （SSI）	LCD_VCPWC 输出 （LCDC）	—
	PF21 输入 / 输出 （端口）	SSISCK1 输入 / 输出 （SSI）	LCD_CLK 输入 （LCDC）	—
	PF20 输入 / 输出 （端口）	SSIDATA0 输入 / 输出 （SSI）	LCD_FLM 输出 （LCDC）	—
PFCRH1	PF19 输入 / 输出 （端口）	SSIWS0 输入 / 输出 （SSI）	LCD_M_DISP 输出 （LCDC）	—
	PF18 输入 / 输出 （端口）	SSISCK0 输入 / 输出 （SSI）	LCD_CL2 输出 （LCDC）	—
	PF17 输入 / 输出 （端口）	$\overline{\text{FCE}}$ 输出 （FLCTL）	LCD_CL1 输出 （LCDC）	—
	PF16 输入 / 输出 （端口）	FRB 输入 （FLCTL）	LCD_DON 输出 （LCDC）	—
PFCRL4	PF15 输入 / 输出 （端口）	NAF7 输入 / 输出 （FLCTL）	LCD_DATA15 输出 （LCDC）	SD_CD 输入 （SDHI）
	PF14 输入 / 输出 （端口）	NAF6 输入 / 输出 （FLCTL）	LCD_DATA14 输出 （LCDC）	SD_WP 输入 （SDHI）
	PF13 输入 / 输出 （端口）	NAF5 输入 / 输出 （FLCTL）	LCD_DATA13 输出 （LCDC）	SD_D1 输入 / 输出 （SDHI）
	PF12 输入 / 输出 （端口）	NAF4 输入 / 输出 （FLCTL）	LCD_DATA12 输出 （LCDC）	SD_D0 输入 / 输出 （SDHI）

设定寄存器	模式位 （PFnMD[1:0]）设定值			
	00	01	10	11
	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）
PFCRL3	PF11 输入 / 输出 （端口）	NAF3 输入 / 输出 （FLCTL）	LCD_DATA11 输出 （LCDC）	SD_CLK 输出 （SDHI）
	PF10 输入 / 输出 （端口）	NAF2 输入 / 输出 （FLCTL）	LCD_DATA10 输出 （LCDC）	SD_CMD 输入 / 输出 （SDHI）
	PF9 输入 / 输出 （端口）	NAF1 输入 / 输出 （FLCTL）	LCD_DATA9 输出 （LCDC）	SD_D3 输入 / 输出 （SDHI）
	PF8 输入 / 输出 （端口）	NAF0 输入 / 输出 （FLCTL）	LCD_DATA8 输出 （LCDC）	SD_D2 输入 / 输出 （SDHI）
PFCRL2	PF7 输入 / 输出 （端口）	FSC 输出 （FLCTL）	LCD_DATA7 输出 （LCDC）	$\overline{\text{SCS1}}$ 输入 / 输出 （SSU）
	PF6 输入 / 输出 （端口）	FOE 输出 （FLCTL）	LCD_DATA6 输出 （LCDC）	SSO1 输入 / 输出 （SSU）
	PF5 输入 / 输出 （端口）	FCDE 输出 （FLCTL）	LCD_DATA5 输出 （LCDC）	SSI1 输入 / 输出 （SSU）
	PF4 输入 / 输出 （端口）	$\overline{\text{FWE}}$ 输出 （FLCTL）	LCD_DATA4 输出 （LCDC）	SSCK1 输入 / 输出 （SSU）
PFCRL1	PF3 输入 / 输出 （端口）	TCLKD 输入 （MTU2）	LCD_DATA3 输出 （LCDC）	$\overline{\text{SCS0}}$ 输入 / 输出 （SSU）
	PF2 输入 / 输出 （端口）	TCLKC 输入 （MTU2）	LCD_DATA2 输出 （LCDC）	SSO0 输入 / 输出 （SSU）
	PF1 输入 / 输出 （端口）	TCLKB 输入 （MTU2）	LCD_DATA1 输出 （LCDC）	SSI0 输入 / 输出 （SSU）
	PF0 输入 / 输出 （端口）	TCLKA 输入 （MTU2）	LCD_DATA0 输出 （LCDC）	SSCK0 输入 / 输出 （SSU）

29.1　特点

- 通过设定控制寄存器，选择多路复用引脚的功能
- 选择通用输入/输出功能或MTU2的TIOC输入/输出功能时，通过设定IO寄存器，选择输入/输出方向
- 端口A通过A/D转换器的A/D控制/状态寄存器 （ADCSR）、 D/A转换器的D/A控制寄存器 （DACR）的设定，切换功能

29.2　寄存器的说明

PFC 有以下的寄存器。

表 29.7　寄存器结构

寄存器名	简称	R/W	初始值	地址	存取长度
端口 B/IO 寄存器 L	PBIORL	R/W	H'0000	H'FFFE3886	8、16
端口 B 控制寄存器 L4	PBCRL4	R/W	H'0001	H'FFFE3890	8*1、16、32
端口 B 控制寄存器 L3	PBCRL3	R/W	H'0000	H'FFFE3892	8、16
端口 B 控制寄存器 L2	PBCRL2	R/W	H'0000	H'FFFE3894	8、16、32
端口 B 控制寄存器 L1	PBCRL1	R/W	H'0000	H'FFFE3896	8、16
IRQOUT 功能控制寄存器	IFCR	R/W	H'0000	H'FFFE38A2	8、16
端口 C/IO 寄存器 L	PCIORL	R/W	H'0000	H'FFFE3906	8、16
端口 C 控制寄存器 L4	PCCRL4	R/W	H'0000	H'FFFE3910	8、16、32
端口 C 控制寄存器 L3	PCCRL3	R/W	H'0000	H'FFFE3912	8、16
端口 C 控制寄存器 L2	PCCRL2	R/W	H'0000	H'FFFE3914	8、16、32
端口 C 控制寄存器 L1	PCCRL1	R/W	H'0000/H'0010*2	H'FFFE3916	8、16
端口 D/IO 寄存器 L	PDIORL	R/W	H'0000	H'FFFE3986	8、16
端口 D 控制寄存器 L4	PDCRL4	R/W	H'0000/H'1111*2	H'FFFE3990	8、16、32
端口 D 控制寄存器 L3	PDCRL3	R/W	H'0000/H'1111*2	H'FFFE3992	8、16
端口 D 控制寄存器 L2	PDCRL2	R/W	H'0000/H'1111*2	H'FFFE3994	8、16、32
端口 D 控制寄存器 L1	PDCRL1	R/W	H'0000/H'1111*2	H'FFFE3996	8、16
端口 E/IO 寄存器 L	PEIORL	R/W	H'0000	H'FFFE3A06	8、16
端口 E 控制寄存器 L4	PECRL4	R/W	H'0000	H'FFFE3A10	8、16、32
端口 E 控制寄存器 L3	PECRL3	R/W	H'0000	H'FFFE3A12	8、16
端口 E 控制寄存器 L2	PECRL2	R/W	H'0000	H'FFFE3A14	8、16、32
端口 E 控制寄存器 L1	PECRL1	R/W	H'0000	H'FFFE3A16	8、16
端口 F/IO 寄存器 H	PFIORH	R/W	H'0000	H'FFFE3A84	8、16、32
端口 F/IO 寄存器 L	PFIORL	R/W	H'0000	H'FFFE3A86	8、16
端口 F 控制寄存器 H4	PFCRH4	R/W	H'0000	H'FFFE3A88	8、16、32
端口 F 控制寄存器 H3	PFCRH3	R/W	H'0000	H'FFFE3A8A	8、16
端口 F 控制寄存器 H2	PFCRH2	R/W	H'0000	H'FFFE3A8C	8、16、32
端口 F 控制寄存器 H1	PFCRH1	R/W	H'0000	H'FFFE3A8E	8、16
端口 F 控制寄存器 L4	PFCRL4	R/W	H'0000	H'FFFE3A90	8、16、32
端口 F 控制寄存器 L3	PFCRL3	R/W	H'0000	H'FFFE3A92	8、16
端口 F 控制寄存器 L2	PFCRL2	R/W	H'0000	H'FFFE3A94	8、16、32
端口 F 控制寄存器 L1	PFCRL1	R/W	H'0000	H'FFFE3A96	8、16
SSI 时钟选择寄存器	SCSR	R/W	H'0000	H'FFFE3AA2	8、16

【注】　*1　8 位存取时，能读不能写。

*2　根据 LSI 的运行模式，初始值会有不同。

29.2.1　端口 B/IO 寄存器 L （PBIORL）

PBIORL 为 16 位可读 / 写寄存器，选择端口 B 引脚的输入 / 输出方向。 PB11IOR ～ PB8IOR 位分别对应 PB11/CTx1/IETxD ～ PB8/CRx0/CRx0/CRx1 引脚。 PBIORL 在端口 B 的引脚功能为通用输入 / 输出 （PB11 ～ PB8）时有效，为其他功能时无效。如果 PBIORL 的位置 1，对应的引脚为输出，如果清 0 就为输入。

PBIORL 的 bit15 ～ 12、 7 ～ 0 是保留位。读写的值总是 0。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PB11 IOR	PB10 IOR	PB9 IOR	PB8 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/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

29.2.2　端口 B 控制寄存器 L1 ～ L4 （PBCRL1 ～ PBCRL4）

PBCRL1 ～ PBCRL4 分别为 16 位可读 / 写寄存器，选择端口 B 多路复用引脚的功能。

（1） 端口 B 控制寄存器 L4 （PBCRL4）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PB12MD[1:0]	
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W：	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

【注】 将bit15～8的值作为H'5A，通过16位或32位存取对PBCRL4进行写操作。
不能进行通过8位存取的写入。

位	位名	初始值	R/W	说　明
15 ～ 2	—	均为 0	R	保留位 读取时，总是读出 0。
1、 0	PB12MD[1:0]	01	R/W	PB12 模式 控制 PB12/$\overline{\text{WDTOVF}}$/$\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$/$\overline{\text{UBCTRG}}$ 引脚的功能。 00：PB12 输出 （端口） 01：$\overline{\text{WDTOVF}}$ 输出 （WDT） 10：$\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$ 输出 （INTC/BSC） 11：$\overline{\text{UBCTRG}}$ 输出 （UBC）

（2） 端口 B 控制寄存器 L3 （PBCRL3）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PB11MD[1:0]		-	-	PB10MD[1:0]		-	-	PB9MD[1:0]		-	-	PB8MD[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/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	—	均为 0	R	保留位 读写的值总是 0。
13、12	PB11MD[1:0]	00	R/W	PB11 模式 控制 PB11/CTx1/IETxD 引脚的功能。 00：PB11 输入 / 输出 （端口） 01：CTx1 输出 （RCAN-TL1） 10：IETxD 输出 （IEB） 11：禁止设定
11、10	—	均为 0	R	保留位 读写的值总是 0。
9、8	PB10MD[1:0]	00	R/W	PB10 模式 控制 PB10/CRx1/IERxD 引脚的功能。 00：PB10 输入 / 输出 （端口） 01：CRx1 输入 （RCAN-TL1） 10：IERxD 输入 （IEB） 11：禁止设定
7、6	—	均为 0	R	保留位 读写的值总是 0。
5、4	PB9MD[1:0]	00	R/W	PB9 模式 控制 PB9/CTx0/CTx0&CTx1 引脚的功能。 00：PB9 输入 / 输出 （端口） 01：CTx0 输出 （RCAN-TL1） 10：CTx0&CTx1 输出 （RCAN-TL1） 11：禁止设定
3、2	—	均为 0	R	保留位 读写的值总是 0。
1、0	PB8MD[1:0]	00	R/W	PB8 模式 控制 PB8/CRx0/CRx0/CRx1 引脚的功能。 00：PB8 输入 / 输出 （端口） 01：CRx0 输入 （RCAN-TL1） 10：CRx0/CRx1 输入 （RCAN-TL1） 11：禁止设定

（3） 端口 B 控制寄存器 L2 （PBCRL2）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PB7MD[1:0]		-	-	PB6MD[1:0]		-	-	PB5MD[1:0]		-	-	PB4MD[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/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	—	均为 0	R	保留位 读写的值总是 0。
13、12	PB7MD[1:0]	00	R/W	PB7 模式 控制 PB7/SDA3/PINT7/IRQ7 引脚的功能。 00：PB7 输入 （端口） 01：SDA3 输入 / 输出 （IIC3） 10：PINT7 输入 （INTC） 11：IRQ7 输入 （INTC）
11、10	—	均为 0	R	保留位 读写的值总是 0。
9、8	PB6MD[1:0]	00	R/W	PB6 模式 控制 PB6/SCL3/PINT6/IRQ6 引脚的功能。 00：PB6 输入 （端口） 01：SCL3 输入 / 输出 （IIC3） 10：PINT6 输入 （INTC） 11：IRQ6 输入 （INTC）
7、6	—	均为 0	R	保留位 读写的值总是 0。
5、4	PB5MD[1:0]	00	R/W	PB5 模式 控制 PB5/SDA2/PINT5/IRQ5 引脚的功能。 00：PB5 输入 （端口） 01：SDA2 输入 / 输出 （IIC3） 10：PINT5 输入 （INTC） 11：IRQ5 输入 （INTC）
3、2	—	0	R	保留位 读写的值总是 0。
1、0	PB4MD[1:0]	00	R/W	PB4 模式 控制 PB4/SCL2/PINT4/IRQ4 引脚的功能。 00：PB4 输入 （端口） 01：SCL2 输入 / 输出 （IIC3） 10：PINT4 输入 （INTC） 11：IRQ4 输入 （INTC）

（4） 端口 B 控制寄存器 L1 （PBCRL1）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PB3MD[1:0]		-	-	PB2MD[1:0]		-	-	PB1MD[1:0]		-	-	PB0MD[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/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	—	均为 0	R	保留位 读写的值总是 0。
13、12	PB3MD[1:0]	00	R/W	PB3 模式 控制 PB3/SDA1/PINT3/IRQ3 引脚的功能。 00：PB3 输入 （端口） 01：SDA1 输入 / 输出 （IIC3） 10：PINT3 输入 （INTC） 11：IRQ3 输入 （INTC）
11、10	—	均为 0	R	保留位 读写的值总是 0。
9、8	PB2MD[1:0]	00	R/W	PB2 模式 控制 PB2/SCL1/PINT2/IRQ2 引脚的功能。 00：PB2 输入 （端口） 01：SCL1 输入 / 输出 （IIC3） 10：PINT2 输入 （INTC） 11：IRQ2 输入 （INTC）
7、6	—	均为 0	R	保留位 读写的值总是 0。
5、4	PB1MD[1:0]	00	R/W	PB1 模式 控制 PB1/SDA0/PINT1/IRQ1 引脚的功能。 00：PB1 输入 （端口） 01：SDA0 输入 / 输出 （IIC3） 10：PINT1 输入 （INTC） 11：IRQ1 输入 （INTC）
3、2	—	0	R	保留位 读写的值总是 0。
1、0	PB0MD[1:0]	00	R/W	PB0 模式 控制 PB0/SCL0/PINT0/IRQ0 引脚的功能。 00：PB0 输入 （端口） 01：SCL0 输入 / 输出 （IIC3） 10：PINT0 输入 （INTC） 11：IRQ0 输入 （INTC）

29.2.3　端口 C/IO 寄存器 L （PCIORL）

PCIORL 为 16 位可读 / 写的寄存器，选择端口 C 引脚的输入 / 输出方向。PC14IOR ～ PC0IOR 位分别对应 PC14/$\overline{\text{WAIT}}$ ～ PC0/A0/$\overline{\text{CS7}}$ 引脚。PCIORL 在端口 C 的引脚功能为通用输入 / 输出（PC14 ～ PC0）时有效，为其他功能时无效。如果 PCIORL 的位置 1，对应的引脚为输出，如果清 0 就为输入。

PCIORL 的 bit15 是保留位。读写的值总是 0。

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

29.2.4　端口 C 控制寄存器 L1 ～ L4 （PCCRL1 ～ PCCRL4）

PCCRL1 ～ PCCRL4 分别为 16 位可读 / 写寄存器，选择端口 C 多路复用引脚的功能。

（1） 端口 C 控制寄存器 L4 （PCCRL4）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PC14 MD0	-	-	-	PC13 MD0	-	-	-	PC12 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/W	R	R	R	R/W	R	R	R	R/W

位	位名	初始值	R/W	说明
15 ～ 9	—	均为 0	R	保留位 读写的值总是 0。
8	PC14MD0	0	R/W	PC14 模式 控制 PC14/$\overline{\text{WAIT}}$ 引脚的功能。 0：PC14 输入 / 输出 （端口） 1：$\overline{\text{WAIT}}$ 输入 （BSC）
7 ～ 5	—	均为 0	R	保留位 读写的值总是 0。
4	PC13MD0	0	R/W	PC13 模式 控制 PC13/RD$\overline{\text{WR}}$ 引脚的功能。 0：PC13 输入 / 输出 （端口） 1：RD$\overline{\text{WR}}$ 输出 （BSC）
3 ～ 1	—	均为 0	R	保留位 读写的值总是 0。
0	PC12MD0	0	R/W	PC12 模式 控制 PC12/CKE 引脚的功能。 0：PC12 输入 / 输出 （端口） 1：CKE 输出 （BSC）

（2） 端口 C 控制寄存器 L3 （PCCRL3）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PC11MD[1:0]		-	-	PC10MD[1:0]		-	-	-	PC9MD0	-	-	-	PC8MD0
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/W	R	R	R	R/W

位	位名	初始值	R/W	说　明
15、14	—	均为 0	R	保留位 读写的值总是 0。
13、12	PC11MD[1:0]	00	R/W	PC11 模式 控制 PC11/$\overline{\text{CASU}}$/$\overline{\text{BREQ}}$ 引脚的功能。 00：PC11 输入 / 输出 （端口） 01：$\overline{\text{CASU}}$ 输出 （BSC） 10：$\overline{\text{BREQ}}$ 输入 （BSC） 11：禁止设定
11、10	—	均为 0	R	保留位 读写的值总是 0。
9、8	PC10MD[1:0]	00	R/W	PC10 模式 控制 PC10/$\overline{\text{RASU}}$/$\overline{\text{BACK}}$ 引脚的功能。 00：PC10 输入 / 输出 （端口） 01：$\overline{\text{RASU}}$ 输出 （BSC） 10：$\overline{\text{BACK}}$ 输出 （BSC） 11：禁止设定
7 ～ 5	—	均为 0	R	保留位 读写的值总是 0。
4	PC9MD0	0	R/W	PC9 模式 控制 PC9/$\overline{\text{CASL}}$ 引脚的功能。 0：PC9 输入 / 输出 （端口） 1：$\overline{\text{CASL}}$ 输出 （BSC）
3 ～ 1	—	均为 0	R	保留位 读写的值总是 0。
0	PC8MD0	0	R/W	PC8 模式 控制 PC8/$\overline{\text{RASL}}$ 引脚的功能。 0：PC8 输入 / 输出 （端口） 1：$\overline{\text{RASL}}$ 输出 （BSC）

（3） 端口 C 控制寄存器 L2 （PCCRL2）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PC7MD0	-	-	-	PC6MD0	-	-	-	PC5MD0	-	-	-	PC4MD0
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	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	—	均为 0	R	保留位 读写的值总是 0。
12	PC7MD0	0	R/W	PC7 模式 控制 PC7/$\overline{WE3}$/DQMUU/$\overline{AH}$/$\overline{ICIOWR}$ 引脚的功能。 0：PC7 输入 / 输出 （端口） 1：$\overline{WE3}$/DQMUU/$\overline{AH}$/$\overline{ICIOWR}$ 输出 （BSC）
11 ～ 9	—	均为 0	R	保留位 读写的值总是 0。
8	PC6MD0	0	R/W	PC6 模式 控制 PC6/$\overline{WE2}$/DQMUL/$\overline{ICIORD}$ 引脚的功能。 0：PC6 输入 / 输出 （端口） 1：$\overline{WE2}$/DQMUL/$\overline{ICIORD}$ 输出 （BSC）
7 ～ 5	—	均为 0	R	保留位 读写的值总是 0。
4	PC5MD0	0	R/W	PC5 模式 控制 PC5/$\overline{WE1}$/DQMLU/$\overline{WE}$ 引脚的功能。 0：PC5 输入 / 输出 （端口） 1：$\overline{WE1}$DQMLU/$\overline{WE}$ 输出 （BSC）
3 ～ 1	—	均为 0	R	保留位 读写的值总是 0。
0	PC4MD0	0	R/W	PC4 模式 控制 PC4/$\overline{WE0}$/DQMLL 引脚的功能。 0：PC4 输入 / 输出 （端口） 1：$\overline{WE0}$/DQMLL 输出 （BSC）

（4） 端口 C 控制寄存器 L1 （PCCRL1）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PC3MD0	-	-	-	PC2MD0	-	-	-	PC1MD0	-	-	PC0MD[1:0]	
初始值：	0	0	0	0	0	0	0	0	0	0	0	0/1*	0	0	0	0
R/W：	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R/W	R/W

【注】* 根据LSI的运行模式，初始值会有不同。

位	位名	初始值	R/W	说明
15 ～ 13	—	均为 0	R	保留位 读写的值总是 0。
12	PC3MD0	0	R/W	PC3 模式 控制 PC3/$\overline{CS3}$ 引脚的功能。 0：PC3 输入 / 输出 （端口） 1：$\overline{CS3}$ 输出 （BSC）
11 ～ 9	—	均为 0	R	保留位 读写的值总是 0。
8	PC2MD0	0	R/W	PC2 模式 控制 PC2/$\overline{CS2}$ 引脚的功能。 0：PC2 输入 / 输出 （端口） 1：$\overline{CS2}$ 输出 （BSC）
7 ～ 5	—	均为 0	R	保留位 读写的值总是 0。
4	PC1MD0	0/1*	R/W	PC1 模式 控制 PC1/A1 引脚的功能。 • 区域 0：32 位模式 0：PC1 输入 / 输出 （端口）（初始值） 1：A1 输出 （地址） • 区域 0：16 位模式 0：禁止设定 1：A1 输出 （地址）（初始值）
3、2	—	均为 0	R	保留位 读写的值总是 0。
1、0	PC0MD[1:0]	00	R/W	PC0 模式 控制 PC0/A0/$\overline{CS7}$ 引脚的功能。 00：PC0 输入 / 输出 （端口） 01：A0 输出 （地址） 10：$\overline{CS7}$ 输出 （BSC） 11：禁止设定

【注】 * 根据 LSI 的运行模式，初始值会有不同。

29.2.5　端口 D/IO 寄存器 L （PDIORL）

PDIORL 为 16 位可读 / 写的寄存器，选择端口 D 引脚的输入 / 输出方向。PD15IOR ～ PD0IOR 位分别对应 PD15/D31/PINT7/SD_WP/ADTRG/TIOC4D ～ PD0/D16/IRQ0/SSCK0/DREQ2/TIOC0A 引脚。 PDIORL 在端口 D 的引脚功能为通用输入 / 输出 （PD15 ～ PD0）或 MTU2 的 TIOC 输入 / 输出时有效、为其他功能时无效。如果 PDIORL 的位置 1，对应的引脚变为输出，如果清 0 就为输入。

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

29.2.6　端口 D 控制寄存器 L1 ～ L4 （PDCRL1 ～ PDCRL4）

PDCRL1 ～ PDCRL4 分别为 16 位可读 / 写寄存器，选择端口 D 多路复用引脚的功能。

（1） 端口 D 控制寄存器 L4 （PDCRL4）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PD15MD[2:0]			-	PD14MD[2:0]			-	PD13MD[2:0]			-	PD12MD[2:0]		
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

【注】* 根据LSI的运行模式，初始值会有不同。

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	PD15MD[2:0]	000/ 001*	R/W	PD15 模式 控制 PD15/D31/PINT7/SD_CD/ADTRG/TIOC4D 引脚的功能。 • 区域 0：32 位模式 000：禁止设定　100：禁止设定 001：D31 输入 / 输出 （数据）（初始值）　101：禁止设定 010：禁止设定　110：禁止设定 011：禁止设定　111：禁止设定 • 区域 0：16 位模式 000：PD15 输入 / 输出 （端口）（初始值）　100：ADTRG 输入 （ADC） 001：D31 输入 / 输出 （数据）　101：TIOC4D 输入 / 输出 （MTU2） 010：PINT7 输入 （INTC）　110：禁止设定 011：SD_CD 输入 （SDHI）　111：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 ～ 8	PD14MD[2:0]	000/001*	R/W	PD14 模式 控制 PD14/D30/PINT6/SD_WP/TIOC4C 引脚的功能。 • 区域 0：32 位模式 000：禁止设定　100：禁止设定 001：D30 输入 / 输出 （数据）（初始值）　101：禁止设定 010：禁止设定　110：禁止设定 011：禁止设定　111：禁止设定 • 区域 0：16 位模式 000：PD14 输入 / 输出 （端口）（初始值）　100：禁止设定 001：D30 输入 / 输出 （数据）　101：TIOC4C 输入 / 输出 （MTU2） 010：PINT6 输入 （INTC）　110：禁止设定 011：SD_WP 输入 （SDHI）　111：禁止设定
7	—	0	R	保留位 读写的值总是 0。
6 ～ 4	PD13MD[2:0]	000/001*	R/W	PD13 模式 控制 PD13/D29/PINT5/SD_D1/TEND1/TIOC4B 引脚的功能。 • 区域 0：32 位模式 000：禁止设定　100：禁止设定 001：D29 输入 / 输出 （数据）（初始值）　101：禁止设定 010：禁止设定　110：禁止设定 011：禁止设定　111：禁止设定 • 区域 0：16 位模式 000：PD13 输入 / 输出 （端口）　100：TEND1 输出 （DMAC） 001：D29 输入 / 输出 （数据）（初始值）　101：TIOC4B 输入 / 输出 （MTU2） 010：PINT5 输入 （INTC）　110：禁止设定 011：SD_D1 输入 / 输出 （SDHI）　111：禁止设定
3	—	0	R	保留位 读写的值总是 0。
2 ～ 0	PD12MD[2:0]	000/001*	R/W	PD12 模式 控制 PD12/D28/PINT4/SD_D0/DACK1/TIOC4A 引脚的功能。 • 区域 0：32 位模式 000：禁止设定　100：禁止设定 001：D28 输入 / 输出 （数据）（初始值）　101：禁止设定 010：禁止设定　110：禁止设定 011：禁止设定　111：禁止设定 • 区域 0：16 位模式 000：PD12 输入 / 输出 （端口）　100：DACK1 输出 （DMAC） 001：D28 输入 / 输出 （数据）（初始值）　101：TIOC4A 输入 / 输出 （MTU2） 010：PINT4 输入 （INTC）　110：禁止设定 011：SD_D0 输入 / 输出 （SDHI）　111：禁止设定

【注】 * 根据 LSI 的运行模式，初始值会有不同。

（2） 端口 D 控制寄存器 L3 （PDCRL3）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PD11MD[2:0]			-	PD10MD[2:0]			-	PD9MD[2:0]			-	PD8MD[2:0]		
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

【注】* 根据LSI的运行模式，初始值会有不同。

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	PD11MD[2:0]	000/ 001*	R/W	PD11 模式 控制 PD11/D27/PINT3/SD_CLK/DREQ1/TIOC3D 引脚的功能。 • 区域 0：32 位模式 000：禁止设定　100：禁止设定 001：D27 输入 / 输出 （数据）（初始值）　101：禁止设定 010：禁止设定　110：禁止设定 011：禁止设定　111：禁止设定 • 区域 0：16 位模式 000：PD11 输入 / 输出 （端口）（初始值）　100：DREQ1 输入 （DMAC） 001：D27 输入 / 输出 （数据）　101：TIOC3D 输入 / 输出 （MTU2） 010：PINT3 输入 （INTC）　110：禁止设定 011：SD_CLK 输出 （SDHI）　111：禁止设定
11	—	0	R	保留位 读写的值总是 0。
10 ～ 8	PD10MD[2:0]	000/ 001*	R/W	PD10 模式 控制 PD10/D26/PINT2/SD_CMD/TEND0/TIOC3C 引脚的功能。 • 区域 0：32 位模式 000：禁止设定　100：禁止设定 001：D26 输入 / 输出 （数据）（初始值）　101：禁止设定 010：禁止设定　110：禁止设定 011：禁止设定　111：禁止设定 • 区域 0：16 位模式 000：PD10 输入 / 输出 （端口）（初始值）　100：TEND0 输出 （DMAC） 001：D26 输入 / 输出 （数据）　101：TIOC3C 输入 / 输出 （MTU2） 010：PINT2 输入 （INTC）　110：禁止设定 011：SD_CMD 输入 / 输出 （SDHI）　111：禁止设定
7	—	0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说明
6 ～ 4	PD9MD[2:0]	000/001*	R/W	PD9 模式 控制 PD9/D25/PINT1/SD_D3/DACK0/TIOC3B 引脚的功能。 • 区域 0：32 位模式 000：禁止设定　100：禁止设定 001：D25 输入 / 输出 （数据）（初始值）　101：禁止设定 010：禁止设定　110：禁止设定 011：禁止设定　111：禁止设定 • 区域 0：16 位模式 000：PD9 输入 / 输出 （端口）　100：DACK0 输出 （DMAC） 001：D25 输入 / 输出 （数据）（初始值）　101：TIOC3B 输入 / 输出 （MTU2） 010：PINT1 输入 （INTC）　110：禁止设定 011：SD_D3 输入 / 输出 （SDHI）　111：禁止设定
3	—	0	R	保留位 读写的值总是 0。
2 ～ 0	PD8MD[2:0]	000/001*	R/W	PD8 模式 控制 PD8/D24/PINT0/SD_D2/DREQ0/TIOC3A 引脚的功能。 • 区域 0：32 位模式 000：禁止设定　100：禁止设定 001：D24 输入 / 输出 （数据）（初始值）　101：禁止设定 010：禁止设定　110：禁止设定 011：禁止设定　111：禁止设定 • 区域 0：16 位模式 000：PD8 输入 / 输出 （端口）　100：DREQ0 输入 （DMAC） 001：D24 输入 / 输出 （数据）（初始值）　101：TIOC3A 输入 / 输出 （MTU2） 010：PINT0 输入 （INTC）　110：禁止设定 011：SD_D2 输入 / 输出 （SDHI）　111：禁止设定

【注】 * 根据 LSI 的运行模式，初始值会有不同。

（3） 端口 D 控制寄存器 L2 （PDCRL2）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PD7MD[2:0]			-	PD6MD[2:0]			-	PD5MD[2:0]			-	PD4MD[2:0]		
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

【注】* 根据LSI的运行模式，初始值会有不同。

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	PD7MD[2:0]	000/ 001*	R/W	PD7 模式 控制 PD7/D23/IRQ7/$\overline{\text{SCS1}}$/TCLKD/TIOC2B 引脚的功能。 • 区域 0：32 位模式 000：禁止设定　100：禁止设定 001：D23 输入 / 输出 （数据）（初始值）　101：禁止设定 010：禁止设定　110：禁止设定 011：禁止设定　111：禁止设定 • 区域 0：16 位模式 000：PD7 输入 / 输出 （端口）（初始值）　100：TCLKD 输入 （MTU2） 001：D23 输入 / 输出 （数据）　101：TIOC2B 输入 / 输出 （MTU2） 010：IRQ7 输入 （INTC）　110：禁止设定 011：$\overline{\text{SCS1}}$ 输入 / 输出 （SSU）　111：禁止设定
11	—	0	R	保留位 读写的值总是 0。
10 ～ 8	PD6MD[2:0]	000/ 001*	R/W	PD6 模式 控制 PD6/D22/IRQ6/SSO1/TCLKC/TIOC2A 引脚的功能。 • 区域 0：32 位模式 000：禁止设定　100：禁止设定 001：D22 输入 / 输出 （数据）（初始值）　101：禁止设定 010：禁止设定　110：禁止设定 011：禁止设定　111：禁止设定 • 区域 0：16 位模式 000：PD6 输入 / 输出 （端口）（初始值）　100：TCLKC 输入 （MTU2） 001：D22 输入 / 输出 （数据）　101：TIOC2A 输入 / 输出 （MTU2） 010：IRQ6 输入 （INTC）　110：禁止设定 011：SSO1 输入 / 输出 （SSU）　111：禁止设定
7	—	0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说明
6 ～ 4	PD5MD[2:0]	000/ 001*	R/W	PD5 模式 控制 PD5/D21/IRQ5/SSI1/TCLKB/TIOC1B 引脚的功能。 • 区域 0：32 位模式 000：禁止设定　　100：禁止设定 001：D21 输入 / 输出 （数据）（初始值）　　101：禁止设定 010：禁止设定　　110：禁止设定 011：禁止设定　　111：禁止设定 • 区域 0：16 位模式 000：PD5 输入 / 输出 （端口）（初始值）　　100：TCLKB 输入 （MTU2） 001：D21 输入 / 输出 （数据）　　101：TIOC1B 输入 / 输出 （MTU2） 010：IRQ5 输入 （INTC）　　110：禁止设定 011：SSI1 输入 / 输出 （SSU）　　111：禁止设定
3	—	0	R	保留位 读写的值总是 0。
2 ～ 0	PD4MD[2:0]	000/ 001*	R/W	PD4 模式 控制 PD4/D20/IRQ4/SSCK1/TCLKA/TIOC1A 引脚的功能。 • 区域 0：32 位模式 000：禁止设定　　100：禁止设定 001：D20 输入 / 输出 （数据）（初始值）　　101：禁止设定 010：禁止设定　　110：禁止设定 011：禁止设定　　111：禁止设定 • 区域 0：16 位模式 000：PD4 输入 / 输出 （端口）（初始值）　　100：TCLKA 输入 （MTU2） 001：D20 输入 / 输出 （数据）　　101：TIOC1A 输入 / 输出 （MTU2） 010：IRQ4 输入 （INTC）　　110：禁止设定 011：SSCK1 输入 / 输出 （SSU）　　111：禁止设定

【注】 * 根据 LSI 的运行模式，初始值会有不同。

（4） 端口 D 控制寄存器 L1 （PDCRL1）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PD3MD[2:0]			-	PD2MD[2:0]			-	PD1MD[2:0]			-	PD0MD[2:0]		
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

【注】* 根据LSI的运行模式，初始值会有不同。

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	PD3MD[2:0]	000/001*	R/W	PD3 模式 控制 PD3/D19/IRQ3/$\overline{\text{SCS0}}$/DACK3/TIOC0D 引脚的功能。 • 区域 0：32 位模式 000：禁止设定　100：禁止设定 001：D19 输入 / 输出 （数据）（初始值）　101：禁止设定 010：禁止设定　110：禁止设定 011：禁止设定　111：禁止设定 • 区域 0：16 位模式 000：PD3 输入 / 输出 （端口）（初始值）　100：DACK3 输出 （DMAC） 001：D19 输入 / 输出 （数据）　101：TIOC0D 输入 / 输出 （MTU2） 010：IRQ3 输入 （INTC）　110：禁止设定 011：$\overline{\text{SCS0}}$ 输入 / 输出 （SSU）　111：禁止设定
11	—	0	R	保留位 读写的值总是 0。
10 ～ 8	PD2MD[2:0]	000/001*	R/W	PD2 模式 控制 PD2/D18/IRQ2/SSO0/DREQ3/TIOC0C 引脚的功能。 • 区域 0：32 位模式 000：禁止设定　100：禁止设定 001：D18 输入 / 输出 （数据）（初始值）　101：禁止设定 010：禁止设定　110：禁止设定 011：禁止设定　111：禁止设定 • 区域 0：16 位模式 000：PD2 输入 / 输出 （端口）（初始值）　100：DREQ3 输入 （DMAC） 001：D18 输入 / 输出 （数据）　101：TIOC0C 输入 / 输出 （MTU2） 010：IRQ2 输入 （INTC）　110：禁止设定 011：SSO0 输入 / 输出 （SSU）　111：禁止设定
7	—	0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说明
6 ～ 4	PD1MD0[2:0]	000/001*	R/W	PD1 模式 控制 PD1/D17/IRQ1/SSI0/DACK2/TIOC0B 引脚的功能。 • 区域 0：32 位模式 000：禁止设定　100：禁止设定 001：D17 输入 / 输出 （数据）（初始值）　101：禁止设定 010：禁止设定　110：禁止设定 011：禁止设定　111：禁止设定 • 区域 0：16 位模式 000：PD1 输入 / 输出 （端口）（初始值）　100：DACK2 输出 （DMAC） 001：D17 输入 / 输出 （数据）　101：TIOC0B 输入 / 输出 （MTU2） 010：IRQ1 输入 （INTC）　110：禁止设定 011：SSI0 输入 / 输出 （SSU）　111：禁止设定
3	—	0	R	保留位 读写的值总是 0。
2 ～ 0	PD0MD0[2:0]	000/001*	R/W	PD0 模式 控制 PD0/D16/IRQ0/SSCK0/DREQ2/TIOC0A 引脚的功能。 • 区域 0：32 位模式 000：禁止设定　100：禁止设定 001：D16 输入 / 输出 （数据）（初始值）　101：禁止设定 010：禁止设定　110：禁止设定 011：禁止设定　111：禁止设定 • 区域 0：16 位模式 000：PD0 输入 / 输出 （端口）（初始值）　100：DREQ2 输入 （DMAC） 001：D16 输入 / 输出 （数据）　101：TIOC0A 输入 / 输出 （MTU2） 010：IRQ0 输入 （INTC）　110：禁止设定 011：SSCK0 输入 / 输出 （SSU）　111：禁止设定

【注】 * 根据 LSI 的运行模式，初始值会有不同。

29.2.7 端口 E/IO 寄存器 L （PEIORL）

PEIORL 为 16 位可读 / 写的寄存器，选择端口 E 引脚的输入 / 输出方向。 PE15IOR ～ PE0IOR 位分别对应 PE15/$\overline{\text{IOIS16}}$/$\overline{\text{RTS3}}$ ～ PE0/$\overline{\text{BS}}$/RxD0/$\overline{\text{ADTRG}}$ 引脚。 PEIORL 在端口 E 的引脚功能为通用输入 / 输出 （PE15 ～ PE0）时有效，在其他功能时无效。如果 PEIORL 的位置 1，对应的引脚为输出，如果清 0 就为输入。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF15 IOR	PF14 IOR	PF13 IOR	PF12 IOR	PF11 IOR	PF10 IOR	PF9 IOR	PF8 IOR	PF7 IOR	PF6 IOR	PF5 IOR	PF4 IOR	PF3 IOR	PF2 IOR	PF1 IOR	PF0 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

29.2.8　端口 E 控制寄存器 L1 ～ L4 （PECRL1 ～ PECRL4）

PECRL1 ～ PECRL4 分别为 16 位可读 / 写寄存器，选择端口 E 多路复用引脚的功能。

（1） 端口 E 控制寄存器 L4 （PECRL4）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PE15MD[1:0]		-	-	PE14MD[1:0]		-	-	PE13MD[1:0]		-	-	PE12MD[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/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	—	均为 0	R	保留位 读写的值总是 0。
13、12	PE15MD[1:0]	00	R/W	PE15 模式 控制 PE15/$\overline{\text{IOIS16}}$//$\overline{\text{RTS3}}$ 引脚的功能。 00：PE15 输入 / 输出 （端口） 01：$\overline{\text{IOIS16}}$ 输入 （BSC） 10：禁止设定 11：$\overline{\text{RTS3}}$ 输入 / 输出 （SCIF）
11、10	—	均为 0	R	保留位 读写的值总是 0。
9、8	PE14MD[1:0]	00	R/W	PE14 模式 控制 PE14/$\overline{\text{CS1}}$/$\overline{\text{CTS3}}$ 引脚的功能。 00：PE14 输入 / 输出 （端口） 01：$\overline{\text{CS1}}$ 输出 （BSC） 10：禁止设定 11：$\overline{\text{CTS3}}$ 输入 / 输出 （SCIF）
7、6	—	均为 0	R	保留位 读写的值总是 0。
5、4	PE13MD[1:0]	00	R/W	PE13 模式 控制 PE13/TxD3 引脚的功能。 00：PE13 输入 / 输出 （端口） 01：禁止设定 10：禁止设定 11：TxD3 输出 （SCIF）
3、2	—	均为 0	R	保留位 读写的值总是 0。
1、0	PE12MD[2:0]	00	R/W	PE12 模式 控制 PE12/RxD3 引脚的功能。 00：PE12 输入 / 输出 （端口） 01：禁止设定 10：禁止设定 11：RxD3 输入 （SCIF）

（2） 端口 E 控制寄存器 L3 （PECRL3）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE11MD[2:0]			-	PE10MD[2:0]			-	-	PE9MD[1:0]		-	-	PE8MD[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/W	R/W	R/W	R	R/W	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	—	0	R	保留位 读写的值总是 0。
14 ～ 12	PE11MD[2:0]	000	R/W	PE11 模式 控制 PE11/$\overline{\text{CS6}}$/$\overline{\text{CE1B}}$/IRQ7/TEND1 引脚的功能。 000：PE11 输入 / 输出 （端口）　100：TEND1 输出 （DMAC） 001：$\overline{\text{CS6}}$/$\overline{\text{CE1B}}$ 输出 （BSC）　101：禁止设定 010：IRQ7 输入 （INTC）　110：禁止设定 011：禁止设定　111：禁止设定
11	—	0	R	保留位 读写的值总是 0。
10 ～ 8	PE10MD[2:0]	000	R/W	PE10 模式 控制 PE10/$\overline{\text{CE2B}}$/IRQ6/TEND0 引脚的功能。 000：PE10 输入 / 输出 （端口）　100：TEND0 输出 （DMAC） 001：$\overline{\text{CE2B}}$ 输出 （BSC）　101：禁止设定 010：IRQ6 输入 （INTC）　110：禁止设定 011：禁止设定　111：禁止设定
7、6	—	均为 0	R	保留位 读写的值总是 0。
5、4	PE9MD[1:0]	00	R/W	PE9 模式 控制 PE9/$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$/IRQ5/SCK3 引脚的功能。 00：PE9 输入 / 输出 （端口） 01：$\overline{\text{CS5}}$/$\overline{\text{CE1A}}$ 输出 （BSC） 10：IRQ5 输入 （INTC） 11：SCK3 输入 / 输出 （SCIF）
3、2	—	均为 0	R	保留位 读写的值总是 0。
1、0	PE8MD[1:0]	00	R/W	PE8 模式 控制 PE8/$\overline{\text{CE2A}}$/IRQ4/SCK2 引脚的功能。 00：PE8 输入 / 输出 （端口） 01：$\overline{\text{CE2A}}$ 输出 （BSC） 10：IRQ4 输入 （INTC） 11：SCK2 输入 / 输出 （SCIF）

（3） 端口 E 控制寄存器 L2 （PECRL2）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE7MD[2:0]			-	PE6MD[2:0]			-	PE5MD[2:0]			-	PE4MD[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/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 ～ 12	PE7MD[2:0]	000	R/W	PE7 模式 控制 PE7/$\overline{\text{FRAME}}$/IRQ3/TxD2/DACK1 引脚的功能。 000：PE7 输入 / 输出 （端口）　100：DACK1 输出 （DMAC） 001：$\overline{\text{FRAME}}$ 输出 （BSC）　101：禁止设定 010：IRQ3 输入 （INTC）　110：禁止设定 011：TxD2 输出 （SCIF）　111：禁止设定
11	—	0	R	保留位 读写的值总是 0。
10 ～ 8	PE6MD[2:0]	000	R/W	PE6 模式 控制 PE6/A25/IRQ2/RxD2/DREQ1 引脚的功能。 000：PE6 输入 / 输出 （端口）　100：DREQ1 输入 （DMAC） 001：A25 输出 （地址）　101：禁止设定 010：IRQ2 输入 （INTC）　110：禁止设定 011：RxD2 输入 （SCIF）　111：禁止设定
7	—	0	R	保留位 读写的值总是 0。
6 ～ 4	PE5MD[2:0]	000	R/W	PE5 模式 控制 PE5/A24/IRQ1/TxD1/DACK0 引脚的功能。 000：PE5 输入 / 输出 （端口）　100：DACK0 输出 （DMAC） 001：A24 输出 （地址）　101：禁止设定 010：IRQ1 输入 （INTC）　110：禁止设定 011：TxD1 输出 （SCIF）　111：禁止设定
3	—	0	R	保留位 读写的值总是 0。
2 ～ 0	PE4MD[2:0]	000	R/W	PE4 模式 控制 PE4/A23/IRQ0/RxD1/DREQ0 引脚的功能。 000：PE4 输入 / 输出 （端口）　100：DREQ0 输入 （DMAC） 001：A23 输出 （地址）　101：禁止设定 010：IRQ0 输入 （INTC）　110：禁止设定 011：RxD1 输入 （SCIF）　111：禁止设定

（4） 端口 E 控制寄存器 L1 （PECRL1）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PE3MD[1:0]		-	-	PE2MD[1:0]		-	-	PE1MD[1:0]		-	PE0MD[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/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	—	均为 0	R	保留位 读写的值总是 0。
13、12	PE3MD[1:0]	00	R/W	PE3 模式 控制 PE3/A22/SCK1 引脚的功能。 00：PE3 输入 / 输出 （端口） 01：A22 输出 （地址） 10：禁止设定 11：SCK1 输入 / 输出 （SCIF）
11、10	—	均为 0	R	保留位 读写的值总是 0。
9、8	PE2MD[1:0]	00	R/W	PE2 模式 控制 PE2/A21/SCK0 引脚的功能。 00：PE2 输入 / 输出 （端口） 01：A21 输出 （地址） 10：禁止设定 11：SCK0 输入 / 输出 （SCIF）
7、6	—	均为 0	R	保留位 读写的值总是 0。
5、4	PE1MD[1:0]	00	R/W	PE1 模式 控制 PE1/$\overline{\text{CS4}}$/$\overline{\text{MRES}}$/TxD0 引脚的功能。 00：PE1 输入 / 输出 （端口） 01：$\overline{\text{CS4}}$ 输出 （BSC） 10：$\overline{\text{MRES}}$ 输入 （系统控制） 11：TxD0 输出 （SCIF）
3	—	0	R	保留位 读写的值总是 0。
2 ～ 0	PE0MD[2:0]	000	R/W	PE0 模式 控制 PE0/$\overline{\text{BS}}$/RxD0/$\overline{\text{ADTRG}}$ 引脚的功能。 000：PE0 输入 / 输出 （端口）　100：$\overline{\text{ADTRG}}$ 输入 （ADC） 001：$\overline{\text{BS}}$ 输出 （BSC）　101：禁止设定 010：禁止设定　110：禁止设定 011：RxD0 输入 （SCIF）　111：禁止设定

29.2.9　端口 F/IO 寄存器 H、 L （PFIORH、 PFIORL）

PFIORH、PFIORL 分别为 16 位可读 / 写寄存器，选择端口 F 引脚的输入 / 输出方向。PF30IOR ～ PF0IOR 位分别对应 PF30/AUDIO_CLK ～ PF0/TCLKA/LCD_DATA0/SSCK0 引脚。 PFIORH、 PFIORL 在端口 F 的引脚功能为通用输入 / 输出（PF30 ～ PF0）时有效、在其他功能时无效。如果 PFIORH、PFIORL 的位置 1，对应的引脚为输出，如果清 0 就为输入。

PFIORH 的 bit15 是保留位。读写的值总是 0。

（1） 端口 F/IO 寄存器 H （PFIORH）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PF30 IOR	PF29 IOR	PF28 IOR	PF27 IOR	PF26 IOR	PF25 IOR	PF24 IOR	PF23 IOR	PF22 IOR	PF21 IOR	PF20 IOR	PF19 IOR	PF18 IOR	PF17 IOR	PF16 IOR
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/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 F/IO 寄存器 L （PFIORL）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF15 IOR	PF14 IOR	PF13 IOR	PF12 IOR	PF11 IOR	PF10 IOR	PF9 IOR	PF8 IOR	PF7 IOR	PF6 IOR	PF5 IOR	PF4 IOR	PF3 IOR	PF2 IOR	PF1 IOR	PF0 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

29.2.10　端口F控制寄存器H1～H4、L1～L4（PFCRH1～PFCRH4、PFCRL1～PFCRL4）

PFCRH1 ～ PFCRH4、PFCRL1 ～ PFCRL4 分别为 16 位可读 / 写寄存器，选择端口 F 多路复用引脚的功能。

（1） 端口 F 控制寄存器 H4 （PFCRH4）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	PF30 MD0	-	-	-	PF29 MD0	-	-	-	PF28 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/W	R	R	R	R/W	R	R	R	R/W

位	位名	初始值	R/W	说明
15 ～ 9	—	均为 0	R	保留位 读写的值总是 0。
8	PF30MD0	0	R/W	PF30 模式 控制 PF30/AUDIO_CLK 引脚的功能。 0：PF30 输入 / 输出 （端口） 1：AUDIO_CLK 输入 （SSI）
7 ～ 5	—	均为 0	R	保留位 读写的值总是 0。
4	PF29MD0	0	R/W	PF29 模式 控制 PF29/SSIDATA3 引脚的功能。 0：PF29 输入 / 输出 （端口） 1：SSIDATA3 输入 / 输出 （SSI）

位	位名	初始值	R/W	说明
3 ～ 1	—	均为 0	R	保留位 读写的值总是 0。
0	PF28MD0	0	R/W	PF28 模式 控制 PF28/SSIWS3 引脚的功能。 0：PF28 输入 / 输出 （端口） 1：SSIWS3 输入 / 输出 （SSI）

（2） 端口 F 控制寄存器 H3 （PFCRH3）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PF27MD0	-	-	-	PF26MD0	-	-	-	PF25MD0	-	-	-	PF24MD0
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	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	—	均为 0	R	保留位 读写的值总是 0。
12	PF27MD0	0	R/W	PF27 模式 控制 PF27/SSISCK3 引脚的功能。 0：PF27 输入 / 输出 （端口） 1：SSISCK3 输入 / 输出 （SSI）
11 ～ 9	—	均为 0	R	保留位 读写的值总是 0。
8	PF26MD0	0	R/W	PF26 模式 控制 PF26/SSIDATA2 引脚的功能。 0：PF26 输入 / 输出 （端口） 1：SSIDATA2 输入 / 输出 （SSI）
7 ～ 5	—	均为 0	R	保留位 读写的值总是 0。
4	PF25MD0	0	R/W	PF25 模式 控制 PF25/SSIWS2 引脚的功能。 0：PF25 输入 / 输出 （端口） 1：SSIWS2 输入 / 输出 （SSI）
3 ～ 1	—	均为 0	R	保留位 读写的值总是 0。
0	PF24MD0	0	R/W	PF24 模式 控制 PF24/SSISCK2 引脚的功能。 0：PF24 输入 / 输出 （端口） 1：SSISCK2 输入 / 输出 （SSI）

（3） 端口 F 控制寄存器 H2 （PFCRH2）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PF23MD[1:0]		-	-	PF22MD[1:0]		-	-	PF21MD[1:0]		-	-	PF20MD[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/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	—	均为 0	R	保留位 读写的值总是 0。
13、12	PF23MD[1:0]	00	R/W	PF23 模式 控制 PF23/SSIDATA1/LCD_VEPWC 引脚的功能。 00：PF23 输入 / 输出 （端口） 01：SSIDATA1 输入 / 输出 （SSI） 10：LCD_VEPWC 输出 （LCDC） 11：禁止设定
11、10	—	均为 0	R	保留位 读写的值总是 0。
9、8	PF22MD[1:0]	00	R/W	PF22 模式 控制 PF22/SSIWS1/LCD_VCPWC 引脚的功能。 00：PF22 输入 / 输出 （端口） 01：SSIWS1 输入 / 输出 （SSI） 10：LCD_VCPWC 输出 （LCDC） 11：禁止设定
7、6	—	均为 0	R	保留位 读写的值总是 0。
5、4	PF21MD[1:0]	00	R/W	PF21 模式 控制 PF21/SSISCK1/LCD_CLK 引脚的功能。 00：PF21 输入 / 输出 （端口） 01：SSISCK1 输入 / 输出 （SSI） 10：LCD_CLK 输入 （LCDC） 11：禁止设定
3、2	—	均为 0	R	保留位 读写的值总是 0。
1、0	PF20MD[1:0]	00	R/W	PF20 模式 控制 PF20/SSIDATA0/LCD_FLM 引脚的功能。 00：PF20 输入 / 输出 （端口） 01：SSIDATA0 输入 / 输出 （SSI） 10：LCD_FLM 输出 （LCDC） 11：禁止设定

（4） 端口 F 控制寄存器 H1 （PFCRH1）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PF19MD[1:0]		-	-	PF18MD[1:0]		-	-	PF17MD[1:0]		-	-	PF16MD[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/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	—	均为 0	R	保留位 读写的值总是 0。
13、12	PF19MD[1:0]	00	R/W	PF19 模式 控制 PF19/SSIWS0/LCD_M_DISP 引脚的功能。 00：PF19 输入 / 输出 （端口） 01：SSIWS0 输入 / 输出 （SSI） 10：LCD_M_DISP 输出 （LCDC） 11：禁止设定
11、10	—	均为 0	R	保留位 读写的值总是 0。
9、8	PF18MD[1:0]	00	R/W	PF18 模式 控制 PF18/SSISCK0/LCD_CL2 引脚的功能。 00：PF18 输入 / 输出 （端口） 01：SSISCK0 输入 / 输出 （SSI） 10：LCD_CL2 输出 （LCDC） 11：禁止设定
7、6	—	均为 0	R	保留位 读写的值总是 0。
5、4	PF17MD[1:0]	00	R/W	PF17 模式 控制 PF17/$\overline{\text{FCE}}$/LCD_CL1 引脚的功能。 00：PF17 输入 / 输出 （端口） 01：$\overline{\text{FCE}}$ 输出 （FLCTL） 10：LCD_CL1 输出 （LCDC） 11：禁止设定
3、2	—	均为 0	R	保留位 读写的值总是 0。
1、0	PF16MD[1:0]	00	R/W	PF16 模式 控制 PF16/FRB/LCD_DON 引脚的功能。 00：PF17 输入 / 输出 （端口） 01：FRB 输入 （FLCTL） 10：LCD_DON 输出 （LCDC） 11：禁止设定

（5） 端口 F 控制寄存器 L4 （PFCRL4）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PF15MD[1:0]		-	-	PF14MD[1:0]		-	-	PF13MD[1:0]		-	-	PF12MD[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/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	—	均为 0	R	保留位 读写的值总是 0。
13、12	PF15MD[1:0]	00	R/W	PF15 模式 控制 PF15/NAF7/LCD_DATA15/SD_CD 引脚的功能。 00：PF15 输入 / 输出 （端口） 01：NAF7 输入 / 输出 （FLCTL） 10：LCD_DATA15 输出 （LCDC） 11：SD_CD 输入 （SDHI）
11、10	—	均为 0	R	保留位 读写的值总是 0。
9、8	PF14MD[1:0]	00	R/W	PF14 模式 控制 PF14/NAF6/LCD_DATA14/SD_WP 引脚的功能。 00：PF14 输入 / 输出 （端口） 01：NAF6 输入 / 输出 （FLCTL） 10：LCD_DATA14 输出 （LCDC） 11：SD_WP 输入 （SDHI）
7、6	—	均为 0	R	保留位 读写的值总是 0。
5、4	PF13MD[1:0]	00	R/W	PF13 模式 控制 PF13/NAF5/LCD_DATA13/SD_D1 引脚的功能。 00：PF13 输入 / 输出 （端口） 01：NAF5 输入 / 输出 （FLCTL） 10：LCD_DATA13 输出 （LCDC） 11：SD_D1 输入 / 输出 （SDHI）
3、2	—	均为 0	R	保留位 读写的值总是 0。
1、0	PF12MD[1:0]	00	R/W	PF12 模式 控制 PF12/NAF4/LCD_DATA12/SD_D0 引脚的功能。 00：PF12 输入 / 输出 （端口） 01：NAF4 输入 / 输出 （FLCTL） 10：LCD_DATA12 输出 （LCDC） 11：SD_D0 输入 / 输出 （SDHI）

（6） 端口 F 控制寄存器 L3 （PFCRL3）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PF11MD[1:0]		-	-	PF10MD[1:0]		-	-	PF9MD[1:0]		-	-	PF8MD[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/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	—	均为 0	R	保留位 读写的值总是 0。
13、12	PF11MD[1:0]	00	R/W	PF11 模式 控制 PF11/NAF3/LCD_DATA11/SD_CLK 引脚的功能。 00：PF11 输入 / 输出 （端口） 01：NAF3 输入 / 输出 （FLCTL） 10：LCD_DATA11 输出 （LCDC） 11：SD_CLK 输出 （SDHI）
11、10	—	均为 0	R	保留位 读写的值总是 0。
9、8	PF10MD[1:0]	00	R/W	PF10 模式 控制 PF10/NAF2/LCD_DATA10/SD_CMD 引脚的功能。 00：PF10 输入 / 输出 （端口） 01：NAF2 输入 / 输出 （FLCTL） 10：LCD_DATA10 输出 （LCDC） 11：SD_CMD 输入 / 输出 （SDHI）
7、6	—	均为 0	R	保留位 读写的值总是 0。
5、4	PF9MD[1:0]	00	R/W	PF9 模式 控制 PF9/NAF1/LCD_DATA9/SD_D3 引脚的功能。 00：PF9 输入 / 输出 （端口） 01：NAF1 输入 / 输出 （FLCTL） 10：LCD_DATA9 输出 （LCDC） 11：SD_D3 输入 / 输出 （SDHI）
3、2	—	均为 0	R	保留位 读写的值总是 0。
1、0	PF8MD[1:0]	00	R/W	PF8 模式 控制 PF8/NAF0/LCD_DATA8/SD_D2 引脚的功能。 00：PF8 输入 / 输出 （端口） 01：NAF0 输入 / 输出 （FLCTL） 10：LCD_DATA8 输出 （LCDC） 11：SD_D2 输入 / 输出 （SDHI）

（7） 端口 F 控制寄存器 F2 （PFCRL2）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PF7MD[1:0]		-	-	PF6MD[1:0]		-	-	PF5MD[1:0]		-	-	PF4MD[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/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	—	均为 0	R	保留位 读写的值总是 0。
13、12	PF7MD[1:0]	00	R/W	PF7 模式 控制 PF7/FSC/LCD_DATA7/$\overline{\text{SCS1}}$ 引脚的功能。 00：PF7 输入 / 输出 （端口） 01：FSC 输出 （FLCTL） 10：LCD_DATA7 输出 （LCDC） 11：$\overline{\text{SCS1}}$ 输入 / 输出 （SSU）
11、10	—	均为 0	R	保留位 读写的值总是 0。
9、8	PF6MD[1:0]	00	R/W	PF6 模式 控制 PF6/FOE/LCD_DATA6/SSO1 引脚的功能。 00：PF6 输入 / 输出 （端口） 01：FOE 输出 （FLCTL） 10：LCD_DATA6 输出 （LCDC） 11：SSO1 输入 / 输出 （SSU）
7、6	—	均为 0	R	保留位 读写的值总是 0。
5、4	PF5MD[1:0]	00	R/W	PF5 模式 控制 PF5/FCDE/LCD_DATA5/SSI1 引脚的功能。 00：PF5 输入 / 输出 （端口） 01：FCDE 输出 （FLCTL） 10：LCD_DATA5 输出 （LCDC） 11：SSI1 输入 / 输出 （SSU）
3、2	—	均为 0	R	保留位 读写的值总是 0。
1、0	PF4MD[1:0]	00	R/W	PF4 模式 控制 PF4/$\overline{\text{FWE}}$/LCD_DATA4/SSCK1 引脚的功能。 00：PF4 输入 / 输出 （端口） 01：$\overline{\text{FWE}}$ 输出 （FLCTL） 10：LCD_DATA4 输出 （LCDC） 11：SSCK1 输入 / 输出 （SSU）

（8） 端口 F 控制寄存器 L1 （PFCRL1）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PF3MD[1:0]		-	-	PF2MD[1:0]		-	-	PF1MD[1:0]		-	-	PF0MD[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/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	—	均为 0	R	保留位 读写的值总是 0。
13、12	PF3MD[1:0]	00	R/W	PF3 模式 控制 PF3/TCLKD/LCD_DATA3/$\overline{\text{SCS0}}$ 引脚的功能。 00：PF3 输入 / 输出 （端口） 01：TCLKD 输入 （MTU2） 10：LCD_DATA3 输出 （LCDC） 11：$\overline{\text{SCS0}}$ 输入 / 输出 （SSU）
11、10	—	均为 0	R	保留位 读写的值总是 0。
9、8	PF2MD[1:0]	00	R/W	PF2 模式 控制 PF2/TCLKC/LCD_DATA2/SSO0 引脚的功能。 00：PF2 输入 / 输出 （端口） 01：TCLKC 输入 （MTU2） 10：LCD_DATA2 输出 （LCDC） 11：SSO0 输入 / 输出 （SSU）
7、6	—	均为 0	R	保留位 读写的值总是 0。
5、4	PF1MD[1:0]	00	R/W	PF1 模式 控制 PF1/TCLKB/LCD_DATA1/SSI0 引脚的功能。 00：PF1 输入 / 输出 （端口） 01：TCLKB 输入 （MTU2） 10：LCD_DATA1 输出 （LCDC） 11：SSI0 输入 / 输出 （SSU）
3、2	—	均为 0	R	保留位 读写的值总是 0。
1、0	PF0MD[1:0]	00	R/W	PF0 模式 控制 PF0/TCLKA/LCD_DATA0/SSCK0 引脚的功能。 00：PF0 输入 / 输出 （端口） 01：TCLKA 输入 （MTU2） 10：LCD_DATA0 输出 （LCDC） 11：SSCK0 输入 / 输出 （SSU）

29.2.11　RQOUT 功能控制寄存器 （IFCR）

IFCR 为 16 位可读 / 写的寄存器，当 $\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$ 输出通过端口 B 控制寄存器 L4（PBCRL4）设定为多路复用功能时，用于控制其输出。 PBCRL4 的设定变为其他功能时，该寄存器的设定不影响引脚的功能。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PB12IRQ[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	位名	初始值	R/W	说明
15 ～ 2	—	均为 0	R	保留位 读写的值总是 0。
1、 0	PB12IRQ[1:0]	00	R/W	PB12IRQOUT 模式 选择 PBCRL4 的 bit1、 0 （PB12MD[1:0]）设定为 （1、 0）时的 $\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$ 引脚功能。 00：中断请求接收输出 01：刷新信号输出 10：中断请求接收输出或刷新信号输出 （根据当时的运行状态，决定是输出中断请求接收输出还是刷新信号输出） 11：通常为高电平输出

29.2.12　SSI 过采样时钟选择寄存器 （SCSR）

SCSR 为 16 位可读 / 写的寄存器，选择在 SSI 使用的过采样时钟的供给源及分频率。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	SSI3CKS[2:0]			-	SSI2CKS[2:0]			-	SSI1CKS[2:0]			-	SSI0CKS[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/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 ～ 12	SSI3CKS[2:0]	000	R/W	SSI ch3 时钟选择 选择在 SSI ch3 使用的过采样时钟的供给源。 参照表 29.8。
11	—	0	R	保留位 读写的值总是 0。
10 ～ 8	SSI2CKS[2:0]	000	R/W	SSI ch2 时钟选择 选择在 SSI ch2 使用的过采样时钟的供给源。 参照表 29.8。
7	—	0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说明
6 ～ 4	SSI1CKS[2:0]	000	R/W	SSI ch1 时钟选择 选择在 SSI ch1 使用的过采样时钟的供给源。 参照表 29.8。
3	—	0	R	保留位 读写的值总是 0。
2 ～ 0	SSI0CKS[2:0]	000	R/W	SSI ch0 时钟选择 选择在 SSI ch0 使用的过采样时钟的供给源。 参照表 29.8。

表 29.8　通过 SSInCKS 位选择过采样时钟供给源

SSInCKS[2:0]*1 设定值	时钟运行模式		
	0 或 1	2	3
000	AUDIO_X1 输入		
001	AUDIO_X1 输入 /4		
010	AUDIO_CLK 输入 *2		
011	AUDIO_CLK 输入 *2/4		
100	EXTAL 输入	CKIO 输入	禁止设定
101	EXTAL 输入 /4	CKIO 输入 /4	禁止设定
110	EXTAL 输入 /2	CKIO 输入 /2	禁止设定
111	EXTAL 输入 /8	CKIO 输入 /8	禁止设定

【注】　*1　n=0 ～ 3

*2　使用 AUDIO_CLK 时，请将端口 F 控制寄存器 H4 （PFCRH4）的 PF30MD0 位设定为 1。

29.3　关于端口 A 的功能切换

在端口 A，A/D 转换器的模拟输入引脚、D/A 转换器的模拟输出引脚被多路复用。通过 A/D 转换器的 A/D 控制 / 状态寄存器、D/A 转换器的 D/A 控制寄存器的设定，自动切换引脚功能 （参照 “第 22 章 A/D 转换器 （ADC）”、“第 23 章 D/A 转换器 （DAC）”）。

表 29.9　端口 A 的功能切换

A/D 转换器	D/A 转换器	引脚功能	
		PA0/AN0 ～ PA5/AN5	PA6/AN6/DA0、 PA7/AN7/DA1
停止时	停止时	PA0 ～ PA5	PA6、 PA7
停止时	运行时	PA0 ～ PA5	DA0、 DA1
运行时	停止时	AN0 ～ AN5	AN6、 AN7

29.4 使用时的注意事项

表 29.1 ～表 29.6 所示的多路复用一览表中，为了抑制引脚的电位成为中间电位状态，除 PA0 ～ PA7、PB0 ～ PB7 外，所有引脚的 I/O 缓冲器均带有弱保持。但请注意，在高阻抗状态时保持的电位，根据噪声等影响会发生变化。

第 30 章 I/O 端口

本 LSI 的端口由 A、 B、 C、 D、 E 和 F 共 6 个端口构成。
各端口的引脚均为与其他功能兼用的多路复用引脚，通过引脚功能控制器（PFC）选择多路复用引脚功能。
端口分别有保存引脚数据的数据寄存器和读出引脚值的端口寄存器。

30.1 特点

1. 端口总数：99个 （输入/输出82个、输入16个、输出1个）
   - 端口A：输入8个
   - 端口B：输入/输出4个、输入8个、输出1个
   - 端口C：输入/输出15个
   - 端口D：输入/输出16个
   - 端口E：输入/输出16个
   - 端口F：输入/输出31个
2. 带弱保持的引脚
   本LSI的以下I/O引脚带有抑制引脚电位成为中间电位状态的弱保持电路。
   - 端口B：PB8～PB12
   - 端口C：PC0～PC14
   - 端口D：PD0～PD15
   - 端口E：PE0～PE15
   - 端口F：PF0～PF30

   弱保持电路内置于I/O引脚，是当没有从外部驱动I/O引脚时，将输入固定为高电平或者低电平的电路。普通CMOS产品中，未使用的输入引脚需要通过外部上拉或者下拉电阻固定输入电平，本LSI中带弱保持的I/O引脚不需要这样的外置电路，因此能够减少部件个数。
   另外，需要通过上拉或者下拉电阻固定引脚电平时，请使电阻值小于等于4.7kΩ。

30.2 端口 A

如图 30.1 所示，端口 A 为有 8 个引脚的输入端口。

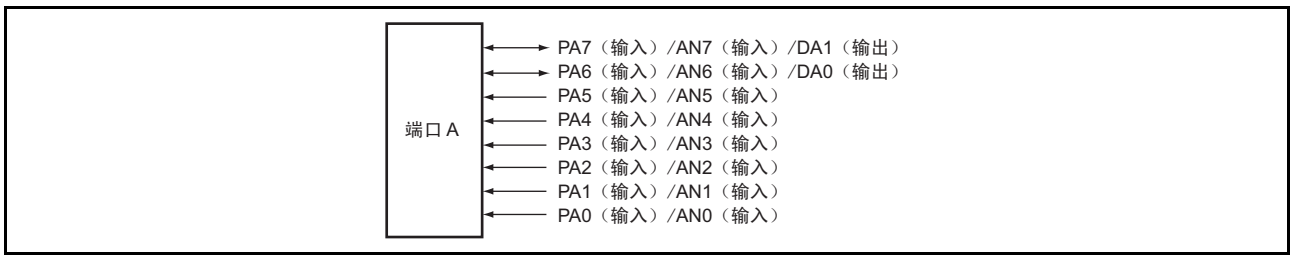


图 30.1 端口 A

30.2.1 寄存器说明

端口 A 的寄存器如表 30.1 所示。

表 30.1 寄存器结构

寄存器名称	简称	R/W	初始值	地址	存取长度
端口 A 数据寄存器 L	PADRL	R	H'00xx	H'FFFE3802	8、 16

30.2.2　端口 A 数据寄存器 L （PADRL）

PADRL 为 16 位只读寄存器， PA7DR ～ PA0DR 位分别对应 PA7/AN7/DA1 ～ PA0/AN0 引脚。 PA7 ～ PA0 通用输入功能只在 A/D 转换器及 D/A 转换器的停止状态时有效。

忽略这些位的写入值，不影响引脚的状态。同时在读取这些位时，不读位的值而直接读出引脚状态。请勿在 A/D 转换器及 D/A 转换器的运行过程中进行读取。 PADRL 的读 / 写操作如表 30.2 所示。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR
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

【注】* 根据外部引脚的状态。

位	位名称	初始值	R/W	说明
15 ～ 8	—	均为 0	R	保留位 读写的值总是 0。
7	PA7DR	引脚的状态	R	参考图 30.2
6	PA6DR	引脚的状态	R	
5	PA5DR	引脚的状态	R	
4	PA4DR	引脚的状态	R	
3	PA3DR	引脚的状态	R	
2	PA2DR	引脚的状态	R	
1	PA1DR	引脚的状态	R	
0	PA0DR	引脚的状态	R	

表 30.2　端口 A 数据寄存器 L （PADRL）的读 / 写操作

- PADRL 的 bit7 ～ 0

引脚功能	读	写
通用输入	引脚的状态	忽略 （不影响引脚的状态）
ANn 输入 /DAn 输出	禁止	忽略 （不影响引脚的状态）

【符号说明】 n=7 ～ 0。但 DA 仅为 DA0 及 DA1。

30.3　端口 B

如图 30.2 所示，端口 B 是有 13 个引脚的输入 / 输出端口。

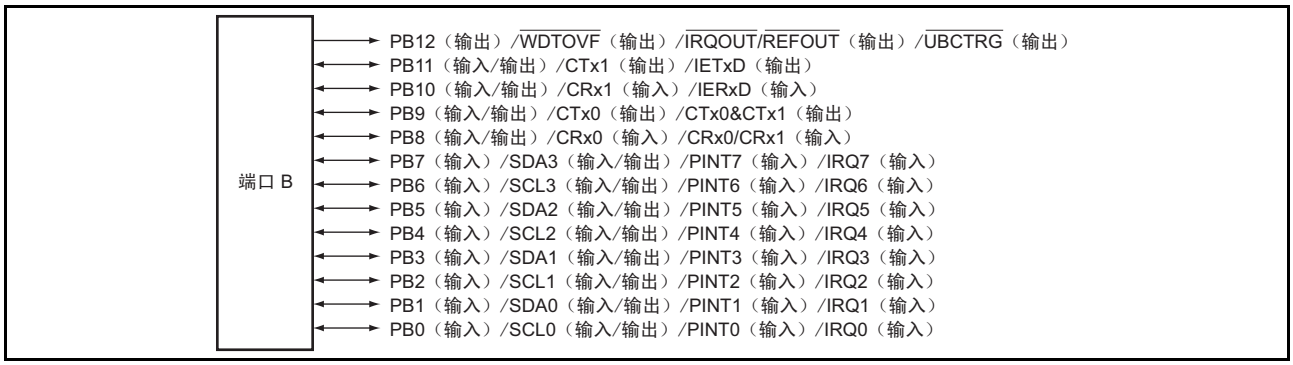


图 30.2　端口 B

30.3.1　寄存器说明

端口 B 的寄存器如表 30.3 所示。

表 30.3　寄存器结构

寄存器名称	简称	R/W	初始值	地址	存取长度
端口 B 数据寄存器 L	PBDRL	R/W	H'00xx	H'FFFE3882	8、16
端口 B 端口寄存器 L	PBPRL	R	H'xxxx	H'FFFE389E	8、16

30.3.2　端口 B 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。PB12DR ～ PB0DR 位分别对应 PB12/WDTOVF/IRQOUT/REFOUT/UBCTRG ～ PB0/SCL0/PINT0/IRQ0 引脚。

引脚功能为通用输出的情况下，如果给 PBDRL 写值，就从引脚输出该值；如果读取 PBDRL，就与引脚的状态无关而直接读出寄存器的值。

引脚功能为通用输入的情况下，如果读取 PBDRL，就不读寄存器的值而直接读出引脚的状态。

在给 PBDRL 写值时，能对 PBDRL 写入该值，但不影响引脚的状态。PBDRL 的读 / 写操作如表 30.4 所示。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	PB12DR	PB11DR	PB10DR	PB9DR	PB8DR	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
初始值：	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	*
R/W：	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

【注】* 根据外部引脚的状态。

位	位名称	初始值	R/W	说明
15 ～ 13	—	均为 0	R	保留位 读写的值总是 0。
12	PB12DR	0	R/W	参考表 30.4
11	PB11DR	0	R/W	
10	PB10DR	0	R/W	
9	PB9DR	0	R/W	
8	PB8DR	0	R/W	
7	PB7DR	引脚的状态	R	
6	PB6DR	引脚的状态	R	
5	PB5DR	引脚的状态	R	
4	PB4DR	引脚的状态	R	
3	PB3DR	引脚的状态	R	
2	PB2DR	引脚的状态	R	
1	PB1DR	引脚的状态	R	
0	PB0DR	引脚的状态	R	

表 30.4 端口 B 数据寄存器 L （PBDRL）的读 / 写操作

- PBDRL 的 bit12

引脚功能	读	写
通用输出	PBDRL 的值	从引脚输出写入值
非通用输出	PBDRL 的值	可以写 PBDRL，但是不影响引脚的状态

- PBDRL 的 bit11 ～ 8

PBIORL	引脚功能	读	写
0	通用输入	引脚的状态	可以写 PBDRL，但是不影响引脚的状态
	非通用输入	引脚的状态	可以写 PBDRL，但是不影响引脚的状态
1	通用输出	PBDRL 的值	从引脚输出写入值
	非通用输出	PBDRL 的值	可以写 PBDRL，但是不影响引脚的状态

- PBDRL 的 bit7 ～ 0

引脚功能	读	写
通用输入	引脚的状态	无效
非通用输入	引脚的状态	无效

30.3.3　端口 B 端口寄存器 L （PBPRL）

PBPRL 为 16 位只读寄存器， PB11PR ～ PB0PR 位分别对应 PB11/CTx1/IETxD、～ PB0/SCL0/PINT0/IRQ0 引脚。与 PFC 的设定无关， PBPRL 总是能读出引脚的值。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	PB11PR	PB10PR	PB9PR	PB8PR	PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	PB0PR
初始值：	0	0	0	0	*	*	*	*	*	*	*	*	*	*	*	*
R/W：	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

【注】* 根据外部引脚的状态。

位	位名称	初始值	R/W	说明
15 ～ 12	—	均为 0	R	保留位 读取时，总是读出 0。写无效。
11	PB11PR	引脚的状态	R	与 PFC 的设定无关，读取时，读出引脚的状态。写无效。
10	PB10PR	引脚的状态	R	
9	PB9PR	引脚的状态	R	
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	

30.4　端口 C

如图 30.3 所示，端口 C 是有 15 个引脚的输入 / 输出端口。

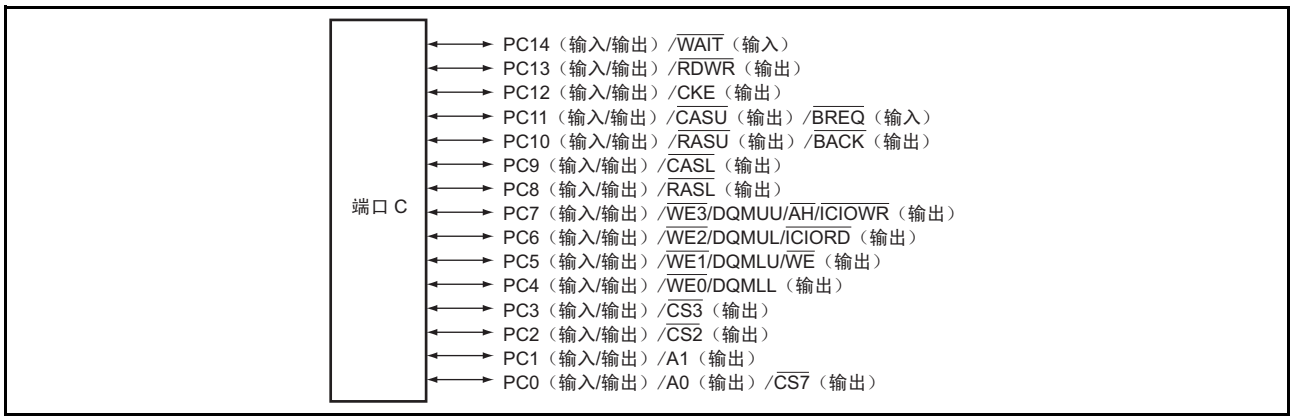


图 30.3　端口 C

30.4.1　寄存器说明

端口 C 的寄存器如表 30.5 所示。

表 30.5　寄存器结构

寄存器名称	简称	R/W	初始值	地址	存取长度
端口 C 数据寄存器 L	PCDRL	R/W	H'xxxx	H'FFFE3902	8、16
端口 C 端口寄存器 L	PCPRL	R	H'xxxx	H'FFFE391E	8、16

30.4.2　端口 C 数据寄存器 L（PCDRL）

PCDRL 为 16 位可读 / 写寄存器，保存端口 C 的数据。PC14DR ～ PC0DR 位各自对应 PC14/$\overline{\text{WAIT}}$ ～ PC0/A0/$\overline{\text{CS7}}$ 引脚。

引脚功能为通用输出的情况下，如果给 PCDRL 写值，就从引脚输出该值；如果去读 PCDRL，就与引脚的状态无关而直接读出寄存器的值。

引脚功能为通用输入的情况下，如果读取 PCDRL，就不读寄存器的值而直接读出引脚状态。在给 PCDRL 写值时，能对 PCDRL 写入该值，但不影响引脚的状态。PCDRL 的读 / 写操作如表 30.6 所示。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PC14DR	PC13DR	PC12DR	PC11DR	PC10DR	PC9DR	PC8DR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说明
15	—	0	R	保留位 读写的值总是 0。
14	PC14DR	0	R/W	参考表 30.6
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 30.6　端口 C 数据寄存器 L （PCDRL）的读 / 写操作

- PCDRL 的 bit14 ～ 0

PCIORL	引脚功能	读	写
0	通用输入	引脚的状态	可以写 PCDRL，但是不影响引脚的状态
	非通用输入	引脚的状态	可以写 PCDRL，但是不影响引脚的状态
1	通用输出	PCDRL 的值	从引脚输出写入值
	非通用输出	PCDRL 的值	可以写 PCDRL，但是不影响引脚的状态

30.4.3　端口 C 端口寄存器 L （PCPRL）

PCPRL 为 16 位只读寄存器，PC14PR ～ PC0PR 位分别对应 PC14/$\overline{\text{WAIT}}$ ～ PC0/A0/$\overline{\text{CS7}}$ 引脚。与 PFC 的设定无关， PCPRL 总能读出引脚的值。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PC14PR	PC13PR	PC12PR	PC11PR	PC10PR	PC9PR	PC8PR	PC7PR	PC6PR	PC5PR	PC4PR	PC3PR	PC2PR	PC1PR	PC0PR
初始值：	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W：	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

【注】* 根据外部引脚的状态。

位	位名称	初始值	R/W	说明
15	—	0	R	保留位 读出的值总是 0。写无效。
14	PC14PR	引脚的状态	R	与 PFC 的设定无关，读取时，读出引脚的状态。写无效。
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	

30.5　端口 D

如图 30.4 所示，端口 D 是有 16 个引脚的输入 / 输出端口。



图 30.4　端口 D

30.5.1　寄存器说明

端口 D 的寄存器如表 30.7 所示。

表 30.7　寄存器结构

寄存器名称	简称	R/W	初始值	地址	存取长度
端口 D 数据寄存器 L	PDDRL	R/W	H'0000	H'FFFE3982	8、16
端口 D 端口寄存器 L	PDPRL	R	H'xxxx	H'FFFE399E	8、16

30.5.2　端口 D 数据寄存器 L （PDDRL）

PDDRL 为 16 位可读 / 写寄存器，保存端口 D 的数据。 PD15DR ～ PD0DR 位各自对应 PD15/D31/PINT7/SD_CD/ADTRG/TIOC4D ～ PD0/D16/IRQ0/SSCK0/DREQ2/TIOC0A 引脚。

引脚功能为通用输出的情况下，如果给 PDDRL 写值，就从引脚输出该值；如果读取 PDDRL，就与引脚的状态无关而直接读出寄存器的值。

引脚功能为通用输入的情况下，如果读取 PDDRL，就不读寄存器的值而直接读出引脚的状态。给 PDDRL 写值时，能对 PDDRL 写该值，但不影响引脚的状态。 PDDRL 的读 / 写操作如表 30.8 所示。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PD15DR	PD14DR	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR	PD8DR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
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	位名称	初始值	R/W	说明
15	PD15DR	0	R/W	参考表 30.8
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 30.8　端口 D 数据寄存器 L （PDDRL）的读 / 写操作

- PDDRL的bit15～0

PDIORL	引脚功能	读	写
0	通用输入	引脚的状态	可以写 PDDRL，但不影响引脚的状态
	非通用输入	引脚的状态	可以写 PDDRL，但不影响引脚的状态
1	通用输出	PDDRL 的值	从引脚输出写入值
	非通用输出	PDDRL 的值	可以写 PDDRL，但不影响引脚的状态

30.5.3　端口 D 端口寄存器 L （PDPRL）

PDPRL 为 16 位只读寄存器， PD15PR ～ PD0PR 位分别对应 PD15/D31/PINT7/SD_CD/$\overline{\text{ADTRG}}$/TIOC4D ～ PD0/D16/IRQ0/SSCK0/DREQ2/TIOC0A 引脚。与 PFC 的设定无关， PDPRL 总能读出引脚的值。

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

【注】* 根据外部引脚的状态。

位	位名称	初始值	R/W	说明
15	PD15PR	引脚的状态	R	与 PFC 的设定无关，读取时，读出引脚的状态。写无效。
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	

30.6　端口 E

如图 30.5 所示，端口 E 是有 16 个引脚的输入 / 输出端口。



图 30.5　端口 E

30.6.1　寄存器说明

端口 E 的寄存器如表 30.9 所示。

表 30.9　寄存器结构

寄存器名称	简称	R/W	初始值	地址	存取长度
端口 E 数据寄存器 L	PEDRL	R/W	H'0000	H'FFFE3A02	8、16
端口 E 端口寄存器 L	PEPRL	R	H'xxxx	H'FFFE3A1E	8、16

30.6.2　端口 E 数据寄存器 L （PEDRL）

PEDRL 为 16 位可读 / 写寄存器，保存端口 E 的数据。PE15DR ～ PE0DR 位分别对应 PE15/$\overline{\text{IOIS16}}$/$\overline{\text{RTS3}}$ ～ PE0/$\overline{\text{BS}}$/RxD0/$\overline{\text{ADTRG}}$ 引脚。

引脚功能为通用输出的情况下，如果给 PEDRL 写值，就从引脚输出该值；如果读取 PEDRL，就与引脚的状态无关而直接读出寄存器的值。

引脚功能为通用输入的情况下，如果读取 PEDRL，就不读寄存器的值而直接读出引脚的状态。可以给 PEDRL 写入值，但对引脚的状态无影响。 PEDRL 的读 / 写操作如表 30.10 所示。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15DR	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
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	位名称	初始值	R/W	说明
15	PE15DR	0	R/W	参考表 30.10
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 30.10　端口 E 数据寄存器 L （PEDRL）的读 / 写操作

- PEDRL 的 bit15 ～ 0

PEIORL	引脚功能	读	写
0	通用输入	引脚的状态	可以写 PEDRL，但是不影响引脚的状态
	非通用输入	引脚的状态	可以写 PEDRL，但是不影响引脚的状态
1	通用输出	PEDRL 的值	从引脚输出写入值
	非通用输出	PEDRL 的值	可以写 PEDRL，但是不影响引脚的状态

30.6.3　端口 E 端口寄存器 L （PEPRL）

PEPRL 为 16 位只读寄存器，PE15PR ～ PE0PR 位分别对应 PE15/$\overline{\text{IOIS16}}$/$\overline{\text{RTS3}}$ ～ PE0/$\overline{\text{BS}}$/RxD0/$\overline{\text{ADTRG}}$ 引脚。与 PFC 的设定无关， PEPRL 总能读出引脚的值。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15PR	PE14PR	PE13PR	PE12PR	PE11PR	PE10PR	PE9PR	PE8PR	PE7PR	PE6PR	PE5PR	PE4PR	PE3PR	PE2PR	PE1PR	PE0PR
初始值：	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W：	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

【注】* 根据外部引脚的状态。

位	位名称	初始值	R/W	说明
15	PE15PR	引脚的状态	R	与 PFC 的设定无关，读取时，读出引脚的状态。写无效。
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	

30.7　端口 F

如图 30.6 所示，端口 F 是有 31 个引脚的输入 / 输出端口。

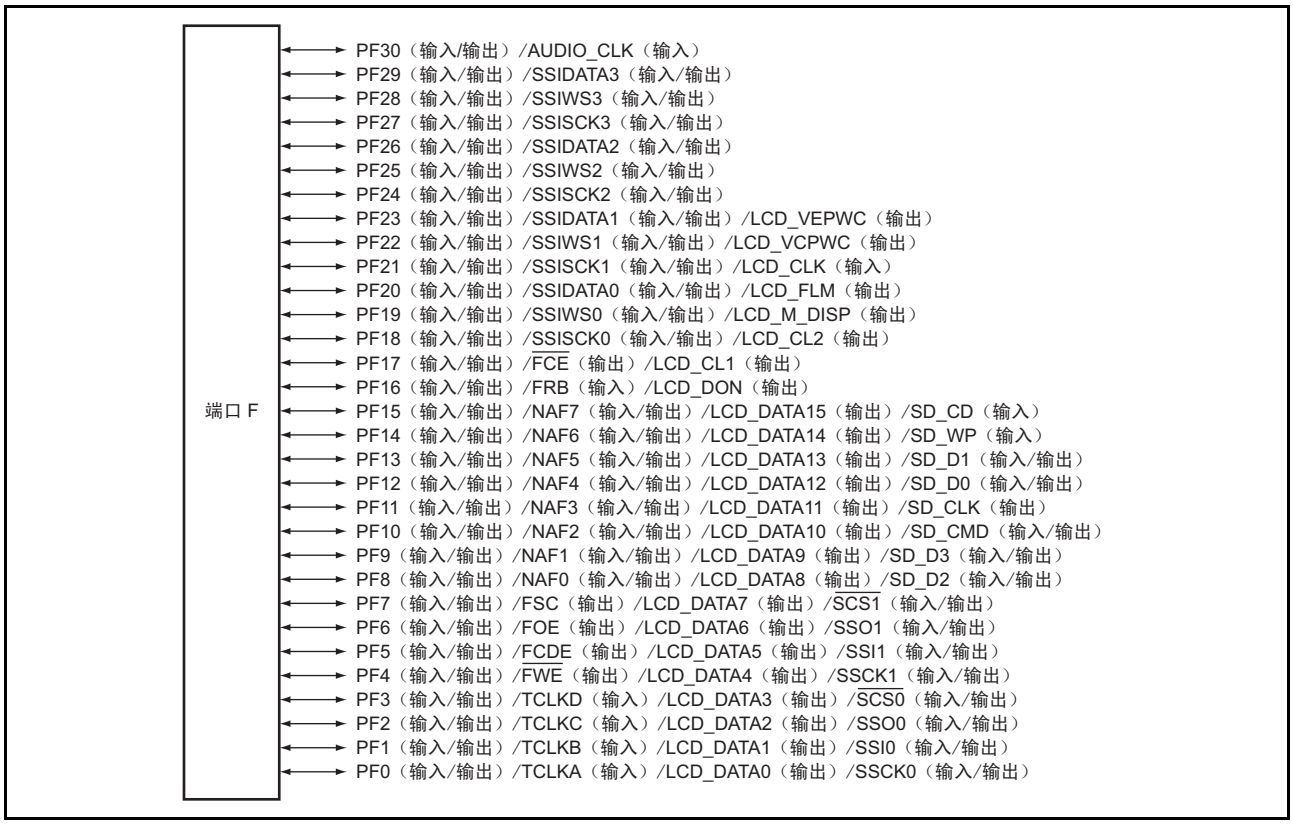


图 30.6　端口 F

30.7.1　寄存器说明

端口 F 的寄存器如表 30.11 所示。

表 30.11　寄存器结构

寄存器名称	简称	R/W	初始值	地址	存取长度
端口 F 数据寄存器 H	PFDRH	R/W	H'0000	H'FFFE3A80	8、16、32
端口 F 数据寄存器 L	PFDRL	R/W	H'0000	H'FFFE3A82	8、16
端口 F 端口寄存器 H	PFPRH	R	H'xxxx	H'FFFE3A9C	8、16、32
端口 F 端口寄存器 L	PFPRL	R	H'xxxx	H'FFFE3A9E	8、16

30.7.2 端口 F 数据寄存器 H、L（PFDRH、PFDRL）

PFDRH、PFDRL 为 16 位可读 / 写寄存器，保存端口 F 的数据。PF30DR ～ PF0DR 位分别对应 PF30/AUDIO_CLK ～ PF0/TCLKA/LCD_DATA0/SSCK0 引脚。

引脚功能为通用输出的情况下，如果给 PFDRH、PFDRL 写值，就从引脚输出该值；如果读取 PFDRH、PFDRL，就与引脚的状态无关而直接读出寄存器的值。

引脚功能为通用输入的情况下，如果读取 PFDRH、PFDRL，就不读寄存器的值而直接读引脚的状态。给 PFDRH、PFDRL 写值时，能对 PFDRH、PFDRL 写该值，但不影响引脚的状态。PFDRH、PFDRL 的读 / 写操作如表 30.12 所示。

（1） 端口 F 数据寄存器 H（PFDRH）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PF30DR	PF29DR	PF28DR	PF27DR	PF26DR	PF25DR	PF24DR	PF23DR	PF22DR	PF21DR	PF20DR	PF19DR	PF18DR	PF17DR	PF16DR
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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说明
15	—	0	R	保留位 读取时值不定。写入时总为 0
14	PF30DR	0	R/W	参考表 30.12
13	PF29DR	0	R/W	
12	PF28DR	0	R/W	
11	PF27DR	0	R/W	
10	PF26DR	0	R/W	
9	PF25DR	0	R/W	
8	PF24DR	0	R/W	
7	PF23DR	0	R/W	
6	PF22DR	0	R/W	
5	PF21DR	0	R/W	
4	PF20DR	0	R/W	
3	PF19DR	0	R/W	
2	PF18DR	0	R/W	
1	PF17DR	0	R/W	
0	PF16DR	0	R/W	

（2） 端口 F 数据寄存器 L （PFDRL）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF15DR	PF14DR	PF13DR	PF12DR	PF11DR	PF10DR	PF9DR	PF8DR	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR
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	位名称	初始值	R/W	说明
15	PF15DR	0	R/W	参考表 30.12
14	PF14DR	0	R/W	
13	PF13DR	0	R/W	
12	PF12DR	0	R/W	
11	PF11DR	0	R/W	
10	PF10DR	0	R/W	
9	PF9DR	0	R/W	
8	PF8DR	0	R/W	
7	PF7DR	0	R/W	
6	PF6DR	0	R/W	
5	PF5DR	0	R/W	
4	PF4DR	0	R/W	
3	PF3DR	0	R/W	
2	PF2DR	0	R/W	
1	PF1DR	0	R/W	
0	PF0DR	0	R/W	

表 30.12　端口 F 数据寄存器 H、 L （PFDRH、 PFDRL）的读 / 写操作

- PFDRH的bit14～0及PFDRL的bit15～0

PFIORH、 L	引脚功能	读	写
0	通用输入	引脚的状态	可以写 PFDRH、 L，但是不影响引脚的状态
	非通用输入	引脚的状态	可以写 PFDRH、 L，但是不影响引脚的状态
1	通用输出	PFDRH、 L 的值	从引脚输出写入值
	非通用输出	PFDRH、 L 的值	可以写 PFDRH、 L，但是不影响引脚的状态

30.7.3　端口 F 端口寄存器 H、L（PFPRH、PFPRL）

PFPRH、PFPRL 分别为 16 位只读寄存器，PF30PR ～ PF0PR 位分别对应 PF30/AUDIO_CLK ～ PF0/TCLKA/LCD_DATA0/SSCK0 引脚。与 PFC 的设定无关， PFPRH、 PFPRL 总能读出引脚的值。

（1） 端口 F 端口寄存器 H （PFPRH）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PF30PR	PF29PR	PF28PR	PF27PR	PF26PR	PF25PR	PF24PR	PF23PR	PF22PR	PF21PR	PF20PR	PF19PR	PF18PR	PF17PR	PF16PR
初始值：	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W：	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

【注】* 根据外部引脚的状态。

位	位名称	初始值	R/W	说明
15	—	0	R	保留位 读写的值总是 0。
14	PF30PR	引脚的状态	R	与 PFC 的设定无关，读取时，读出引脚的状态。写无效。
13	PF29PR	引脚的状态	R	
12	PF28PR	引脚的状态	R	
11	PF27PR	引脚的状态	R	
10	PF26PR	引脚的状态	R	
9	PF25PR	引脚的状态	R	
8	PF24PR	引脚的状态	R	
7	PF23PR	引脚的状态	R	
6	PF22PR	引脚的状态	R	
5	PF21PR	引脚的状态	R	
4	PF20PR	引脚的状态	R	
3	PF19PR	引脚的状态	R	
2	PF18PR	引脚的状态	R	
1	PF17PR	引脚的状态	R	
0	PF16PR	引脚的状态	R	

（2） 端口 F 端口寄存器 L （PFPRL）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF15PR	PF14PR	PF13PR	PF12PR	PF11PR	PF10PR	PF9PR	PF8PR	PF7PR	PF6PR	PF5PR	PF4PR	PF3PR	PF2PR	PF1PR	PF0PR
初始值：	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W：	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

【注】* 根据外部引脚的状态。

位	位名称	初始值	R/W	说明
15	PF15PR	引脚的状态	R	与 PFC 的设定无关，读取时，读出引脚的状态。写无效。
14	PF14PR	引脚的状态	R	
13	PF13PR	引脚的状态	R	
12	PF12PR	引脚的状态	R	
11	PF11PR	引脚的状态	R	
10	PF10PR	引脚的状态	R	
9	PF9PR	引脚的状态	R	
8	PF8PR	引脚的状态	R	
7	PF7PR	引脚的状态	R	
6	PF6PR	引脚的状态	R	
5	PF5PR	引脚的状态	R	
4	PF4PR	引脚的状态	R	
3	PF3PR	引脚的状态	R	
2	PF2PR	引脚的状态	R	
1	PF1PR	引脚的状态	R	
0	PF0PR	引脚的状态	R	

30.8 使用时的注意事项

通过 PFC 选择以下引脚功能时，不能通过存取数据寄存器或端口寄存器读出引脚状态。

- A25～A21、 A1、 A0 （地址总线）
- $\overline{\text{D31}}$～$\overline{\text{D16}}$ （数据总线）
- $\overline{\text{BS}}$
- $\overline{\text{CS7}}$、 $\overline{\text{CS4}}$～$\overline{\text{CS1}}$、 $\overline{\text{CS5}}$/$\overline{\text{CE1A}}$、 $\overline{\text{CS6}}$/$\overline{\text{CE1B}}$、 $\overline{\text{CE2A}}$、 $\overline{\text{CE2B}}$
- RDWR
- $\overline{\text{WE3}}$/DQMUU/AH/$\overline{\text{ICIOWR}}$、 $\overline{\text{WE2}}$/DQMUL/$\overline{\text{ICIORD}}$、 $\overline{\text{WE1}}$/DQMLU/$\overline{\text{WE}}$、 $\overline{\text{WE0}}$/DQMLL
- $\overline{\text{RASU}}$、 $\overline{\text{RASL}}$、 $\overline{\text{CASU}}$、 $\overline{\text{CASL}}$
- CKE
- $\overline{\text{FRAME}}$
- $\overline{\text{WAIT}}$
- $\overline{\text{BREQ}}$
- $\overline{\text{BACK}}$
- $\overline{\text{IOIS16}}$
- $\overline{\text{MRES}}$
- NAF7～0

第 31 章 内置 RAM

本 LSI 内置可进行高速存取的高速 RAM 以及即使在深度待机模式下也可保持数据的保持用 RAM。能保存指令或数据。

高速内置 RAM 通过 RAM 允许及写允许，能禁止存储器的运行及写存取。

保持用内置 RAM 能按页选择在深度待机模式下是否保持数据。

31.1 特点

- 存储器映像

  内置RAM配置在表31.1、表31.2的地址空间。

表 31.1 内置 RAM （高速）地址空间

页	地址
页 0	H'FFF80000 ～ H'FFF83FFF
页 1	H'FFF84000 ～ H'FFF87FFF
页 2	H'FFF88000 ～ H'FFF8BFFF
页 3	H'FFF8C000 ～ H'FFF8FFFF

表 31.2 内置 RAM （保持用）地址空间

页	地址
页 0	H'FFFF8000 ～ H'FFFF8FFF
页 1	H'FFFF9000 ～ H'FFFF9FFF
页 2	H'FFFFA000 ～ H'FFFFAFFF
页 3	H'FFFFB000 ～ H'FFFFBFFF

- 端口

  高速内置RAM的各页具有2个单独的读/写端口，连接内部DMA总线 （ID总线）、 CPU取指令总线（F总线）、 CPU存储器存取总线 （M总线）（但F总线只连接读出端口）。 F总线及M总线用于通过CPU进行的存取。 ID总线用于通过DMAC进行的存取。

  保持用内置RAM的各页具有1个读/写端口，连接外围总线。
- 优先顺序

  同时发生不同总线存取高速内置RAM相同页的请求时，按照优先顺序进行存取。优先顺序从高到低为ID总线、 M总线、 F总线。

31.2 使用时的注意事项

31.2.1 页竞争

同时发生不同总线存取高速内置 RAM 相同页的请求时，就发生页竞争。虽然各存取正常结束，但这样的竞争会导致存储器存取性能降低。因此，建议在软件上采取对策，尽量避免引起竞争。例如，如果每个总线存取不同的页，就不发生竞争。

31.2.2 关于 RAME 位、 RAMWE 位

对于高速内置 RAM，禁止设定 RAME 位及 RAMWE 位时，在设定 RAME 位及 RAMWE 位前必须对各页进行任意相同地址的读 / 写。不进行时，各页最后写入的数据有可能不能写入 RAM。

```
对//页0
 MOV.L  #H'FFF80000, R0
 MOV.L  @R0, R1
 MOV.L  R1, @R0

对//页1
 MOV.L  #H'FFF84000, R0
 MOV.L  @R0, R1
 MOV.L  R1, @R0

对//页2
 MOV.L  #H'FFF88000, R0
 MOV.L  @R0, R1
 MOV.L  R1, @R0

对//页3
 MOV.L  #H'FFF8C000, R0
 MOV.L  @R0, R1
 MOV.L  R1, @R0
```

图 31.1 读 / 写例子

31.2.3 指令配置禁止领域

从保持用内置 RAM 的最后地址开始 16 字节以内（地址 H'FFFFBFF0 ～ H'FFFFBFFF），请勿配置指令。配置时就会出现 CPU 从地址 H'FFFFC000 以后的内置外围模块空间取溢出，并发生地址错误的情况。

第 32 章　低功耗模式

本 LSI 支持的低功耗模式有：睡眠模式、软件待机模式、深度待机模式和模块待机功能。在低功耗模式中，通过停止 CPU、时钟、内置存储器、部分内置外围模块等的功能，或关闭电源，能降低功耗。通过复位或中断等解除低功耗模式。

32.1　特点

32.1.1　低功耗模式的种类

低功耗模式有以下的模式和功能：

1. 睡眠模式
2. 软件待机模式
3. 深度待机模式
4. 模块待机功能

从程序执行状态转移至各模式的条件、各模式中的 CPU 和外围模块等的状态以及各模式的解除方法如**表 32.1** 所示。

表 32.1　低功耗模式的状态

低功耗模式	转移条件	状态 *1									解除方法
		CPG	CPU	CPU 寄存器	内置 RAM（高速）缓冲存储器	内置 RAM（保持用）	内置外围模块	RTC	电源	外部存储器	
睡眠模式	在 STBCR 的 STBY 位为 0 的状态下执行 SLEEP 指令	运行	停止	保持	运行	运行	运行	运行 *2	运行	自动刷新	• 中断 • 手动复位 • 上电复位 • DMA 地址错误
软件待机模式	在 STBCR 的 STBY 位为 1、DEEP 位为 0 的状态下执行 SLEEP 指令	停止	停止	保持	停止（保持内容）	停止	停止	运行 *2	运行	自刷新	• NMI 中断 • IRQ 中断 • 手动复位 • 上电复位
深度待机模式	在 STBCR 的 STBY 位和 DEEP 位都为 1 的状态下执行 SLEEP 指令	停止	停止	停止	停止（不保持内容）	停止（保持内容 *3）	停止	运行 *2	停止	自刷新	• NMI 中断 *4 • IRQ 中断 *4 • 手动复位 *4 • 上电复位 *4
模块待机模式	将 STBCR2、STBCR3、STBCR4、STBCR5、STBCR6 的 MSTP 位置 1	运行	运行	保持	运行	运行	停止指定模块	停止	运行	自动刷新	• MSTP 位清 0 • 上电复位 （但仅限 H-UDI、UBC、 DMAC）

【注】 *1　引脚状态为保持或高阻抗。详细情况请参照 “附录 A. 引脚状态”。

*2　RCR2 寄存器的 START 位置 1 时， RTC 运行。详细情况请参照 “第 14 章 实时时钟 （RTC）”。

*3　如果 RAMKP 寄存器的 RAMKP3 ～ RAMKP0 位置 1，在转移至深度待机模式时，能保持内置 RAM 对象区域的内容。

*4　通过中断 （NMI、 IRQ）及复位 （手动复位、上电复位）解除深度待机模式。但通过 NMI 中断或 IRQ 中断解除深度待机模式时，不进行中断异常处理，而是进行上电复位异常处理。而且手动复位也是上电复位异常处理。

32.2　寄存器的说明

低功耗模式下使用的相关寄存器有以下寄存器。

表 32.2　寄存器结构

寄存器名	简称	R/W	初始值	地址	存取长度
待机控制寄存器	STBCR	R/W	H'00	H'FFFE0014	8
待机控制寄存器 2	STBCR2	R/W	H'00	H'FFFE0018	8
待机控制寄存器 3	STBCR3	R/W	H'7E	H'FFFE0408	8
待机控制寄存器 4	STBCR4	R/W	H'FF	H'FFFE040C	8
待机控制寄存器 5	STBCR5	R/W	H'FF	H'FFFE0410	8
待机控制寄存器 6	STBCR6	R/W	H'FF	H'FFFE0414	8
系统控制寄存器 1	SYSCR1	R/W	H'FF	H'FFFE0402	8
系统控制寄存器 2	SYSCR2	R/W	H'FF	H'FFFE0404	8
系统控制寄存器 3	SYSCR3	R/W	H'00	H'FFFE0418	8
深度待机控制寄存器	DSCTR	R/W	H'00	H'FFFF2800	8
深度待机控制寄存器 2	DSCTR2	R/W	H'00	H'FFFF2802	8
深度待机解除源选择寄存器	DSSSR	R/W	H'0000	H'FFFF2804	16
深度待机解除源标志寄存器	DSFR	R/W	H'0000	H'FFFF2808	16
保持用内置 RAM 调整寄存器	DSRTR	R/W	H'00	H'FFFF280C	8

32.2.1　待机控制寄存器 （STBCR）

STBCR 是 8 位可读 / 写寄存器，指定低功耗模式的状态。仅字节存取有效。

【注】 本寄存器进行写操作时，请参照 “32.4 使用时的注意事项”。

位：	7	6	5	4	3	2	1	0
	STBY	DEEP	-	-	-	-	-	-
初始值：	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R	R	R	R	R	R

位	位名	初始值	R/W	说　明
7	STBY	0	R/W	软件待机、深度待机
6	DEEP	0	R/W	指定向软件待机模式、深度待机模式的转移。 0x：通过执行 SLEEP 指令，转移至睡眠模式 10：通过执行 SLEEP 指令，转移至软件待机模式 11：通过执行 SLEEP 指令，转移至深度待机模式
5 ～ 0	—	均为 0	R	保留位 读写的值总是 0。

【符号说明】 x：Don't care

32.2.2　待机控制寄存器 2（STBCR2）

STBCR2 是 8 位可读 / 写寄存器，控制低功耗模式时各模块的运行。 STBCR2 仅字节存取时有效。

【注】 本寄存器进行写操作时，请参照 “32.4 使用时的注意事项”。

位：	7	6	5	4	3	2	1	0
	MSTP10	MSTP9	MSTP8	MSTP7	-	-	-	-
初始值：	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R	R	R	R

位	位名	初始值	R/W	说　明
7	MSTP10	0	R/W	模块停止 10 如果 MSTP10 位置 1，就停止向 H-UDI 提供时钟。 0：H-UDI 运行 1：停止向 H-UDI 提供时钟
6	MSTP9	0	R/W	模块停止 9 如果 MSTP9 位置 1，就停止向 UBC 提供时钟。 0：UBC 运行 1：停止向 UBC 提供时钟
5	MSTP8	0	R/W	模块停止 8 如果 MSTP8 位置 1，就停止向 DMAC 提供时钟。 0：DMAC 运行 1：停止向 DMAC 提供时钟
4	MSTP7	0	R/W	模块停止 7 如果 MSTP7 位置 1，就停止向 FPU 提供时钟。MSTP7 位置 1 后，不能通过写 0 清除。即：一旦通过 MSTP7 位置 1 停止向 FPU 提供时钟后，就不能通过将 MSTP7 位清 0 重新开始向 FPU 提供时钟。 停止向 FPU 提供时钟后，重新开始时，请将本 LSI 上电复位。 0：FPU 运行 1：停止向 FPU 提供时钟
3 ～ 0	—	均为 0	R	保留位 读写的值总是 0。

32.2.3　待机控制寄存器 3（STBCR3）

STBCR3 是 8 位可读 / 写寄存器，控制低功耗模式时各模块的运行。 STBCR3 仅在字节存取时有效。

【注】 本寄存器进行写操作时，请参照 “32.4 使用时的注意事项”。

位：	7	6	5	4	3	2	1	0
	HIZ	MSTP36	MSTP35	MSTP34	MSTP33	MSTP32	MSTP31	MSTP30
初始值：	0	1	1	1	1	1	1	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	HIZ	0	R/W	端口高阻抗 选择软件待机模式及深度待机模式时是保持特定的输出引脚状态还是高阻抗。控制哪个引脚请参照 “附录 A. 引脚状态”。 WDT 的 WTSCR 的 TME 位为 1 时，请勿设定该位。欲将输出引脚的状态设为高阻抗时，必须在 TME 位为 0 时，将 HIZ 位置位。 0：软件待机模式及深度待机模式时保持引脚状态 1：软件待机模式及深度待机模式时引脚状态为高阻抗
6	MSTP36	1	R/W	模块停止 36 如果 MSTP36 位置 1，停止向 IEB 提供时钟。 0：IEB 运行 1：停止向 IEB 提供时钟
5	MSTP35	1	R/W	模块停止 35 如果 MSTP35 位置 1，就停止向 MTU2 提供时钟。 0：MTU2 运行 1：停止向 MTU2 提供时钟
4	MSTP34	1	R/W	模块停止 34 如果 MSTP34 位置 1，就停止向 SDHI0 提供时钟。 0：SDHI0 运行 1：停止向 SDHI0 提供时钟
3	MSTP33	1	R/W	模块停止 33 如果 MSTP33 位置 1，就停止向 SDHI1 提供时钟。 0：SDHI1 运行 1：停止向 SDHI1 提供时钟
2	MSTP32	1	R/W	模块停止 32 如果 MSTP32 位置 1，就停止向 ADC 提供时钟。 0：ADC 运行 1：停止向 ADC 提供时钟
1	MSTP31	1	R/W	模块停止 31 如果 MSTP31 位置 1，就停止向 DAC 提供时钟。 0：DAC 运行 1：停止向 DAC 提供时钟
0	MSTP30	0	R/W	模块停止 30 如果 MSTP30 位置 1，就停止向 RTC 提供时钟。 0：RTC 运行 1：停止向 RTC 提供时钟

32.2.4　待机控制寄存器 4 （STBCR4）

STBCR4 是 8 位可读 / 写寄存器，控制低功耗模式时各模块的运行。 STBCR4 仅在字节存取时有效。

【注】 本寄存器进行写操作时，请参照 “32.4 使用时的注意事项”。

位：	7	6	5	4	3	2	1	0
	MSTP47	MSTP46	MSTP45	MSTP44	-	MSTP42	MSTP41	MSTP40
初始值：	1	1	1	1	1	1	1	1
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
7	MSTP47	1	R/W	模块停止 47 如果 MSTP47 位置 1，就停止向 SCIF0 提供时钟。 0：SCIF0 运行 1：停止向 SCIF0 提供时钟
6	MSTP46	1	R/W	模块停止 46 如果 MSTP46 位置 1，就停止向 SCIF1 提供时钟。 0：SCIF1 运行 1：停止向 SCIF1 提供时钟
5	MSTP45	1	R/W	模块停止 45 如果 MSTP45 位置 1，就停止向 SCIF2 提供时钟。 0：SCIF2 运行 1：停止向 SCIF2 提供时钟
4	MSTP44	1	R/W	模块停止 44 如果 MSTP44 位置 1，就停止向 SCIF3 提供时钟。 0：SCIF3 运行 1：停止向 SCIF3 提供时钟
3	—	1	R	保留位 读写的值总是 1。
2	MSTP42	1	R/W	模块停止 42 如果 MSTP42 位置 1，就停止向 CMT 提供时钟。 0：CMT 运行 1：停止向 CMT 提供时钟
1	MSTP41	1	R/W	模块停止 41 如果 MSTP41 位置 1，就停止向 LCDC 提供时钟。 0：LCDC 运行 1：停止向 LCDC 提供时钟
0	MSTP40	1	R/W	模块停止 40 如果 MSTP40 位置 1，就停止向 FLCTL 提供时钟。 0：FLCTL 运行 1：停止向 FLCTL 提供时钟

32.2.5　待机控制寄存器 5（STBCR5）

STBCR5 是 8 位可读 / 写寄存器，控制低功耗模式时各模块的运行。 STBCR5 仅在字节存取时有效。

【注】 本寄存器进行写操作时，请参照 “32.4 使用时的注意事项”。

位：	7	6	5	4	3	2	1	0
	MSTP57	MSTP56	MSTP55	MSTP54	MSTP53	MSTP52	MSTP51	MSTP50
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/W	R/W

位	位名	初始值	R/W	说　　明
7	MSTP57	1	R/W	模块停止 57 如果 MSTP57 位置 1，就停止向 IIC3-0 提供时钟。 0：IIC3-0 运行 1：停止向 IIC3-0 提供时钟
6	MSTP56	1	R/W	模块停止 56 如果 MSTP56 位置 1，就停止向 IIC3-1 提供时钟。 0：IIC3-1 运行 1：停止向 IIC3-1 提供时钟
5	MSTP55	1	R/W	模块停止 55 如果 MSTP55 位置 1，就停止向 IIC3-2 提供时钟。 0：IIC3-2 运行 1：停止向 IIC3-2 提供时钟
4	MSTP54	1	R/W	模块停止 54 如果 MSTP54 位置 1，就停止向 IIC3-3 提供时钟。 0：IIC3-3 运行 1：停止向 IIC3-3 提供时钟
3	MSTP53	1	R/W	模块停止 53 如果 MSTP53 位置 1，就停止向 RCAN0 提供时钟。 0：RCAN0 运行 1：停止向 RCAN0 提供时钟
2	MSTP52	1	R/W	模块停止 52 如果 MSTP52 位置 1，就停止向 RCAN1 提供时钟。 0：RCAN1 运行 1：停止向 RCAN1 提供时钟
1	MSTP51	1	R/W	模块停止 51 如果 MSTP51 位置 1，就停止向 SSU0 提供时钟。 0：SSU0 运行 1：停止向 SSU0 提供时钟
0	MSTP50	1	R/W	模块停止 50 如果 MSTP50 位置 1，就停止向 SSU1 提供时钟。 0：SSU1 运行 1：停止向 SSU1 提供时钟

32.2.6　待机控制寄存器 6（STBCR6）

STBCR6 是 8 位可读 / 写寄存器，控制低功耗模式时各模块的运行。 STBCR6 仅在字节存取时有效。

【注】 本寄存器进行写操作时，请参照 “32.4 使用时的注意事项”。

位：	7	6	5	4	3	2	1	0
	MSTP67	MSTP66	MSTP65	MSTP64	MSTP63	MSTP62	-	MSTP60
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/W

位	位名	初始值	R/W	说　明
7	MSTP67	1	R/W	模块停止 67 如果 MSTP67 位置 1，就停止向 SSI0 提供时钟。 0：SSI0 运行 1：停止向 SSI0 提供时钟
6	MSTP66	1	R/W	模块停止 66 如果 MSTP66 位置 1，就停止向 SSI1 提供时钟。 0：SSI1 运行 1：停止向 SSI1 提供时钟
5	MSTP65	1	R/W	模块停止 65 如果 MSTP65 位置 1，就停止向 SSI2 提供时钟。 0：SSI2 运行 1：停止向 SSI2 提供时钟
4	MSTP64	1	R/W	模块停止 64 如果 MSTP64 位置 1，就停止向 SSI3 提供时钟。 0：SSI3 运行 1：停止向 SSI3 提供时钟
3	MSTP63	1	R/W	模块停止 63 如果 MSTP63 位置 1，就停止向 ROM-DEC 提供时钟。 0：ROM-DEC 运行 1：停止向 ROM-DEC 提供时钟
2	MSTP62	1	R/W	模块停止 62 如果 MSTP62 位置 1，就停止向 SRC 提供时钟。 0：SRC 运行 1：停止向 SRC 提供时钟
1	—	1	R	保留位 读写的值总是 1。
0	MSTP60	1	R/W	模块停止 60 如果 MSTP60 位置 1，就停止向 USB 提供时钟。 0：USB 运行 1：停止向 USB 提供时钟

32.2.7　系统控制寄存器 1（SYSCR1）

SYSCR1 是 8 位可读 / 写寄存器，设定允许 / 禁止存取内置 RAM（高速）。SYSCR1 仅在字节存取时有效。

如果 RAME 位置 1，内置 RAM （高速）就有效。如果清 0，就不能存取内置 RAM （高速）。此时，从内置 RAM （高速）读数据或取指令时读出不定值，忽视内置 RAM （高速）的写入。初始值为 1。

禁止 RAME 位的设定时，在设定 RAME 位前必须执行对各页任意相同地址的读 / 写。如果不执行，最后写的数据就有可能不能写入内置 RAM （高速）。而且，在紧接 SYSCR1 的写指令后不能设置存取内置 RAM （高速）的指令。如果设置存取内置 RAM （高速）的指令，就不能保证正常存取。

通过将该位置 1 使内置 RAM （高速）有效时，请在紧接 SYSCR1 的写指令之后设置 SYSCR1 的读指令。如果在紧接 SYSCR1 的写指令之后设置内置 RAM （高速）存取指令时，就不能保证正常存取。

【注】 本寄存器进行写操作时，请参照 “32.4 使用时的注意事项”。

位：	7	6	5	4	3	2	1	0
	-	-	-	-	RAME3	RAME2	RAME1	RAME0
初始值：	1	1	1	1	1	1	1	1
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	—	均为 1	R	保留位 读写的值总是 1。
3	RAME3	1	R/W	RAM 允许 3 （对象：内置 RAM （高速）的页 3*） 0：内置 RAM （高速）无效 1：内置 RAM （高速）有效
2	RAME2	1	R/W	RAM 允许 2 （对象：内置 RAM （高速）的页 2*） 0：内置 RAM （高速）无效 1：内置 RAM （高速）有效
1	RAME1	1	R/W	RAM 允许 1 （对象：内置 RAM （高速）的页 1*） 0：内置 RAM （高速）无效 1：内置 RAM （高速）有效
0	RAME0	1	R/W	RAM 允许 0 （对象：内置 RAM （高速）的页 0*） 0：内置 RAM （高速）无效 1：内置 RAM （高速）有效

【注】 * 关于各页的地址请参照 “第 31 章 内置 RAM”。

32.2.8　系统控制寄存器 2（SYSCR2）

SYSCR2 是 8 位可读 / 写寄存器，设定允许 / 禁止向内置 RAM（高速）进行写入。SYSCR2 仅在字节存取时有效。

如果 RAMWE 位置 1，向内置 RAM（高速）的写入就有效。如果清 0，就不能向内置 RAM（高速）进行写入。此时，忽视向内置 RAM（高速）的写入。初始值为 1。

禁止 RAMWE 位的设定时，在设定 RAMWE 位前必须执行对各页任意相同地址的读 / 写。不执行时，最后写的数据就有可能不能写入内置 RAM（高速）。而且，在紧接 SYSCR2 的写指令之后不能设置存取内置 RAM（高速）的指令。如果设置存取内置 RAM（高速）的指令，就不能保证正常存取。

通过将该位置 1 使向内置 RAM（高速）的写入有效时，请在紧接 SYSCR2 的写指令之后设置 SYSCR2 的读指令。如果在紧接 SYSCR2 的写指令之后设置存取内置 RAM（高速）的指令时，就不能保证正常存取。

【注】 本寄存器进行写操作时，请参照“32.4 使用时的注意事项”。

位：	7	6	5	4	3	2	1	0
	-	-	-	-	RAMWE3	RAMWE2	RAMWE1	RAMWE0
初始值：	1	1	1	1	1	1	1	1
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	—	均为 1	R	保留位 读写的值总是 1。
3	RAMWE3	1	R/W	RAM 写允许 3（对象：内置 RAM（高速）的页 3*） 0：内置 RAM（高速）的写无效 1：内置 RAM（高速）的写有效
2	RAMWE2	1	R/W	RAM 写允许 2（对象：内置 RAM（高速）的页 2*） 0：内置 RAM（高速）的写无效 1：内置 RAM（高速）的写有效
1	RAMWE1	1	R/W	RAM 写允许 1（对象：内置 RAM（高速）的页 1*） 0：内置 RAM（高速）的写无效 1：内置 RAM（高速）的写有效
0	RAMWE0	1	R/W	RAM 写允许 0（对象：内置 RAM（高速）的页 0*） 0：内置 RAM（高速）的写无效 1：内置 RAM（高速）的写有效

【注】 * 关于各页的地址请参照“第 31 章 内置 RAM”。

32.2.9　系统控制寄存器 3（SYSCR3）

SYSCR3 是 8 位可读 / 写寄存器，执行 SSI0 ～ 3 和 IEB 的软件复位控制。 SYSCR3 仅在字节存取时有效。

【注】 本寄存器进行写操作时，请参照 “32.4 使用时的注意事项”。

位：	7	6	5	4	3	2	1	0
	AXTALE	-	-	IEBSRST	SSI3SRST	SSI2SRST	SSI1SRST	SSI0SRST
初始值：	0	0	0	0	0	0	0	0
R/W：	R/W	R	R	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说　　明
7	AXTALE	0	R/W	音频用晶体谐振器允许 指定音频用晶体谐振器的功能有效 / 无效。 0：音频用晶体谐振器的功能有效 1：音频用晶体谐振器的功能无效
6、5	—	均为 0	R	保留位 读写的值总是 0。
4	IEBSRST	0	R/W	IEB 软件复位 通过软件控制 IEB 的复位。 0：解除 IEB 的复位 1：IEB 为复位状态
3	SSI3SRST	0	R/W	SSI3 软件复位 通过软件控制 SSI3 的复位。 0：解除 SSI3 的复位 1：SSI3 为复位状态
2	SSI2SRST	0	R/W	SSI2 软件复位 通过软件控制 SSI2 的复位。 0：解除 SSI2 的复位 1：SSI2 为复位状态
1	SSI1SRST	0	R/W	SSI1 软件复位 通过软件控制 SSI1 的复位。 0：解除 SSI1 的复位 1：SSI1 为复位状态
0	SSI0SRST	0	R/W	SSI0 软件复位 通过软件控制 SSI0 的复位。 0：解除 SSI0 的复位 1：SSI0 为复位状态

32.2.10　深度待机控制寄存器 （DSCTR）

DSCTR 是 8 位可读 / 写寄存器，设定在深度待机模式时是否保持相应内置 RAM （保持用）区域的内容。DSCTR 仅在字节存取时有效。

如果 RRAMKP3 ～ 0 位置 1，在深度待机模式时保持相应内置 RAM （保持用）区域的内容。如果清 0，在深度待机模式时不保持相应内置 RAM （保持用）区域的内容。

【注】 本寄存器进行写操作时，请参照 “32.4 使用时的注意事项”。

位：	7	6	5	4	3	2	1	0
	-	-	-	-	RRAMKP3	RRAMKP2	RRAMKP1	RRAMKP0
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	—	均为 0	R	保留位 读写的值总是 0。
3	RRAMKP3	0	R/W	RRAM 保持区域 3 （对象：内置 RAM （保持用）的页 3*） 0：深度待机模式时，不保持内置 RAM （保持用） 1：深度待机模式时，保持内置 RAM （保持用）
2	RRAMKP2	0	R/W	RRAM 保持区域 2 （对象：内置 RAM （保持用）的页 2*） 0：深度待机模式时，不保持内置 RAM （保持用） 1：深度待机模式时，保持内置 RAM （保持用）
1	RRAMKP1	0	R/W	RRAM 保持区域 1 （对象：内置 RAM （保持用）的页 1*） 0：深度待机模式时，不保持内置 RAM （保持用） 1：深度待机模式时，保持内置 RAM （保持用）
0	RRAMKP0	0	R/W	RRAM 保持区域 0 （对象：内置 RAM （保持用）的页 0*） 0：深度待机模式时，不保持内置 RAM （保持用） 1：深度待机模式时，保持内置 RAM （保持用）

【注】 * 关于各页的地址请参照 “第 31 章 内置 RAM”。

32.2.11　深度待机控制寄存器 2 （DSCTR2）

DSCTR2 是 8 位可读 / 写寄存器，控制从深度待机模式返回时，外部总线控制引脚的状态控制和返回时的启动方法。 DSCTR2 仅在字节存取时有效。

【注】 本寄存器进行写操作时，请参照 “32.4 使用时的注意事项”。

位：	7	6	5	4	3	2	1	0
	CS0 KEEPE	RAM BOOT	-	-	-	-	-	-
初始值：	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R	R	R	R	R	R

位	位名	初始值	R/W	说　　明
7	CS0KEEPE	0	R/W	外部总线控制引脚状态保持 0：从深度待机模式返回时，不保持外部总线控制引脚的状态 1：从深度待机模式返回时，保持外部总线控制引脚的状态
6	RAMBOOT	0	R/W	选择深度待机返回方法 在通过 $\overline{\text{MRES}}$、 NMI、 IRQ 解除深度待机时的上电复位异常处理中，从以下地址取出程序计数器 （PC）和堆栈指针 （SP） 0：H'00000000 地址、 H'00000004 地址 1：H'FFFF8000 地址、 H'FFFF8004 地址
5 ～ 0	—	均为 0	R	保留位 读写的值总是 0。

32.2.12　深度待机解除源选择寄存器 （DSSSR）

DSSSR 是 16 位可读 / 写寄存器，由选择通过哪种中断解除深度待机模式的位构成。只有在引脚分配到 PE4 ～ PE11， IRQ0 ～ 7 才有效。仅在字节存取时有效。

【注】 本寄存器进行写操作时，请参照 “32.4 使用时的注意事项”。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	MRES	IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
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/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 ～ 9	—	均为 0	R	保留位 读写的值总是 0。
8	MRES	0	R/W	通过手动复位返回 0：通过手动复位，不返回 1：通过手动复位，返回
7	IRQ7	0	R/W	通过 IRQ7 返回 （PE11） 0：通过 IRQ7，不返回 1：通过 IRQ7，返回
6	IRQ6	0	R/W	通过 IRQ6 返回 （PE10） 0：通过 IRQ6，不返回 1：通过 IRQ6，返回
5	IRQ5	0	R/W	通过 IRQ5 返回 （PE9） 0：通过 IRQ5，不返回 1：通过 IRQ5，返回
4	IRQ4	0	R/W	通过 IRQ4 返回 （PE8） 0：通过 IRQ4，不返回 1：通过 IRQ4，返回
3	IRQ3	0	R/W	通过 IRQ3 返回 （PE7） 0：通过 IRQ3，不返回 1：通过 IRQ3，返回
2	IRQ2	0	R/W	通过 IRQ2 返回 （PE6） 0：通过 IRQ2，不返回 1：通过 IRQ2，返回
1	IRQ1	0	R/W	通过 IRQ1 返回 （PE5） 0：通过 IRQ1，不返回 1：通过 IRQ1，返回
0	IRQ0	0	R/W	通过 IRQ0 返回 （PE4） 0：通过 IRQ0，不返回 1：通过 IRQ0，返回

32.2.13　深度待机解除源标志寄存器 （DSFR）

DSFR 是 16 位可读 / 写寄存器，由确认通过哪种中断解除深度待机模式的标志以及解除深度待机模式后执行引脚状态解除的位所构成。通过中断 （NMI、 IRQ）及手动复位解除深度待机模式时， DSFR 执行上电复位异常处理，但本寄存器保持之前的值。通过上电复位解除深度待机模式时，本寄存器初始化为 H'0000。仅在字节存取时有效。

转移至深度待机模式前，需要清除所有的标志。

【注】 本寄存器进行写操作时，请参照 “32.4 使用时的注意事项”。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IOKEEP	-	-	-	-	-	MRESF	NMIF	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	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)*

【注】* 为清除标志，只能在读1后写0。

位	位名	初始值	R/W	说　　明
15	IOKEEP	0	R/(W)*	引脚状态保持解除 解除深度待机模式时，解除引脚状态保持的位。 0：不保持引脚状态 ［清除条件］ • 读 1 后写 0 1：保持引脚状态 ［置位条件］ • 转移为深度待机模式时
14 ～ 10	—	均为 0	R	保留位 读写的值总是 0。
9	MRESF	0	R/(W)*	MRES 标志 0：在 $\overline{\text{MRES}}$ 引脚不中断 1：在 $\overline{\text{MRES}}$ 引脚中断
8	NMIF	0	R/(W)*	NMI 标志 0：在 NMI 引脚不中断 1：在 NMI 引脚中断
7	IRQ7F	0	R/(W)*	IRQ7 标志 0：在 IRQ7 （PE11）引脚不中断 1：在 IRQ7 （PE11）引脚中断
6	IRQ6F	0	R/(W)*	IRQ6 标志 0：在 IRQ6 （PE10）引脚不中断 1：在 IRQ6 （PE10）引脚中断
5	IRQ5F	0	R/(W)*	IRQ5 标志 0：在 IRQ5 （PE9）引脚不中断 1：在 IRQ5 （PE9）引脚中断
4	IRQ4F	0	R/(W)*	IRQ4 标志 0：在 IRQ4 （PE8）引脚不中断 1：在 IRQ4 （PE8）引脚中断

位	位名	初始值	R/W	说　　明
3	IRQ3F	0	R/(W)*	IRQ3 标志 0：在 IRQ3 （PE7）引脚不中断 1：在 IRQ3 （PE7）引脚中断
2	IRQ2F	0	R/(W)*	IRQ2 标志 0：在 IRQ2 （PE6）引脚不中断 1：在 IRQ2 （PE6）引脚中断
1	IRQ1F	0	R/(W)*	IRQ1 标志 0：在 IRQ1 （PE5）引脚不中断 1：在 IRQ1 （PE5）引脚中断
0	IRQ0F	0	R/(W)*	IRQ0 标志 0：在 IRQ0 （PE4）引脚不中断 1：在 IRQ0 （PE4）引脚中断

【注】 *　为了清除标志，只能在读 1 后写 0。

32.2.14　保持用内置 RAM 调整寄存器 （DSRTR）

DSRTR 是 8 位可读 / 写寄存器。

DSRTR 是在深度待机模式时，调整保持用内置 RAM 的待机电流的寄存器。DSRTR 仅在字节存取时有效。在深度待机模式保持内置 RAM（保持用）的数据时，必须在向本寄存器写入 H'09 后转移至深度待机模式。$\overline{\text{RES}}$ 引脚为断言或解除深度待机模式后，初始化本寄存器。

【注】 本寄存器进行写操作时，请参照 “32.4 使用时的注意事项”。

位：	7	6	5	4	3	2	1	0
	-	TRMD[6:0]						
初始值：	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

位	位名	初始值	R/W	说　　明
7	—	0	R	保留位 读写的值总是 0。
6 ～ 0	TRMD[6:0]	均为 0	R/W	保持用内置 RAM 调整数据 深度待机模式时，可调整保持用内置 RAM 的待机电流。

32.3 运行说明

32.3.1 睡眠模式

（1） 向睡眠模式转移

如果在 STBCR 的 STBY 位为 0 时执行 SLEEP 指令，就从程序执行状态转移至睡眠模式。虽然执行 SLEEP 指令后，CPU 停止运行，但保持 CPU 寄存器内容。内置外围模块继续运行。在时钟模式 0、1、3 时，能设定继续从 CKIO 引脚输出时钟。

（2） 解除睡眠模式

通过中断 （NMI、 IRQ、内置外围）、 DMA 地址错误及复位 （手动复位、上电复位）解除睡眠模式。

- 通过中断解除
  如果产生NMI、 IRQ及内置外围的各中断，就解除睡眠模式并执行中断异常处理。产生中断的优先级低于CPU的状态寄存器（SR）设定的中断屏蔽级时以及在模块通过内置外围模块禁止中断时，不接受中断请求而且也不解除睡眠模式。
- 通过DMA地址错误解除
  如果发生DMA地址错误，就解除睡眠模式并执行DMA地址错误异常处理。
- 通过复位解除
  通过上电复位及手动复位解除睡眠模式。

32.3.2 软件待机模式

（1） 向软件待机模式转移

在 STBCR 的 STBY 位为 1、DEEP 位为 0 时，如果执行 SLEEP 指令，就从程序执行状态转移至软件待机模式。在软件待机模式中不仅是 CPU，时钟和内置外围模块也停止运行。在时钟模式 0、 1、 3 时，停止从 CKIO 引脚输出时钟。

保持 CPU 及高速缓存的寄存器内容。内置外围模块的寄存器有的被初始化。软件待机模式时外围模块的寄存器状态请参照 **“34.3 各运行模式下的寄存器状态一览”**。

CPU 在 1 个周期结束向 STBCR 的写入后，执行下一个指令处理。但实际的写入会超过 1 个周期。因此，为了使从 CPU 向 STBCR 写入的值准确反映在 SLEEP 指令，请在读出 STBCR 后执行 SLEEP 指令。

向软件待机模式转移的顺序如下所示。

1. 将WDT的定时控制寄存器 （WTCSR）的TME位清0，停止WDT运行。
2. 将WDT的定时器计数器 （WTCNT）置0，设定WTCSR寄存器CKS[2:0]位的值超过振荡稳定时间。
3. 将STBCR寄存器的STBY位设定为1、DEEP位设定0后，读取STBCR寄存器。之后执行SLEEP指令。

（2） 解除软件待机模式

通过中断（NMI、IRQ）及复位（手动复位、上电复位）解除软件待机模式。在时钟模式 0、1、3 时，开始从 CKIO 引脚输出时钟。

- 通过中断解除

  如果检测出 NMI 引脚的下降沿或上升沿（通过中断控制器（INTC）的中断控制寄存器 0（ICR0）的 NMI 边沿选择位（NMIE）选择）、IRQ 引脚（IRQ7 ～ IRQ0）的下降沿或上升沿（通过中断控制器（INTC）的中断控制寄存器 1（ICR1）的 IRQn 检测选择位（IRQn1S ～ IRQn0S）选择），就开始时钟振荡。该时钟只提供给计算振荡稳定时间的振荡稳定计数器（WDT）。

  转移至软件待机模式前，如果经过 WDT 的看门狗定时器控制 / 状态寄存器（WTCSR）的时钟选择位（CKS[2:0]）设定的时间，就发生 WDT 溢出。通过发生溢出判断时钟稳定时，就向本 LSI 整体提供时钟。这样就解除软件待机模式，并执行 NMI 中断异常处理（IRQ 时，为 IRQ 中断异常处理）。

  通过 NMI 中断或 IRQ 中断解除软件待机模式时，设定 CKS[2:0] 位以便 WDT 的溢出周期超过振荡稳定时间。

  从检测出中断到解除软件待机模式为止，CKIO 引脚时钟输出的相位不稳定。而且，如果通过 NMI 引脚的下降沿解除软件待机模式，在进入软件待机模式时（时钟停止时）的 NMI 引脚应为高电平，并且返回软件待机模式时（启动振荡稳定后的时钟时）NMI 引脚应为低电平。而且，如果通过 NMI 引脚的上升沿解除软件待机模式，进入软件待机模式时（时钟停止时）的 NMI 引脚应为低电平，并且返回软件待机模式时（启动振荡稳定后的时钟时）NMI 引脚应为高电平（与 IRQ 引脚的情况相同）。

- 通过复位解除

  如果 $\overline{\text{RES}}$ 引脚为低电平，就解除软件待机模式，并转移至上电复位之后，如果 $\overline{\text{RES}}$ 引脚为高电平，就开始上电复位异常处理。

  如果 $\overline{\text{MRES}}$ 引脚为低电平，就解除软件待机模式，并转移至手动复位。之后，如果 $\overline{\text{MRES}}$ 引脚为高电平，就开始手动复位异常处理。

  时钟振荡稳定前，$\overline{\text{RES}}$ 引脚或 $\overline{\text{MRES}}$ 引脚请保持为低电平。

  时钟模式 0、1、3 时，继续从 CKIO 引脚输出内部时钟。

（3） 软件待机模式转移时的注意事项

通过中断（NMI、IRQ）及复位（手动复位、上电复位）解除软件待机模式，但如果同时产生 SLEEP 指令和 NMI、IRQ 以外的中断，就会出现由于接收中断，而从软件待机模式返回的情况。

向软件待机模式转移时，请在执行 SLEEP 指令前设定不产生中断。

32.3.3 软件待机模式的应用例子

对在 NMI 信号的下降沿转移至软件待机模式，以及在 NMI 信号的上升沿进行解除的例子进行说明。该例子的时序如图 32.1 所示。

在中断控制寄存器 0 （ICR0）的 NMI 边沿选择位 （NMIE）为 0 （下降沿检测）的状态下，如果 NMI 引脚从高电平变为低电平，就接收 NMI 中断。通过 NMI 异常服务程序将 NMIE 位置 1（上升沿检测），在 STBCR 的 STBY 位和 DEEP 位分别为 1、0 的状态下，如果执行 SLEEP 指令，就转移至软件待机模式。之后如果 NMI 引脚从低电平变为高电平，就解除软件待机模式。

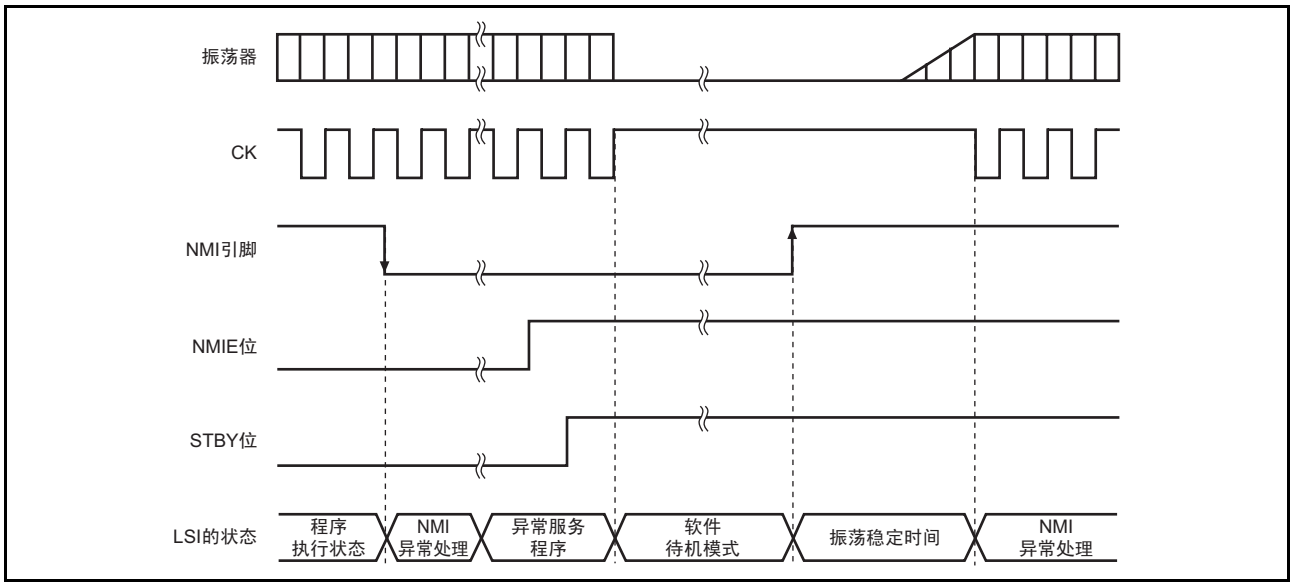


图 32.1 软件待机模式时的 NMI 时序 （应用例子）

32.3.4　深度待机模式

（1） 向深度待机模式转移

在 STBCR 的 STBY 位和 DEEP 位置 1 的状态下，如果执行 SLEEP 指令，就从程序执行状态转移至深度待机模式。在深度待机模式中，不仅 CPU、时钟及内置外围模块停止运行，通过 DSCTR 寄存器的 RRAMKP3 ～ RRAMKP0 位的设定，除保持的内置 RAM （保持用）区域以及 RTC 外，关闭电源，这样可大幅降低功耗。因此不保存 CPU、高速缓存的寄存器内容及内置外围模块的寄存器的值。但可保持向深度待机模式转移前引脚状态的值。

CPU 在 1 个周期结束向 DSCTR 的写入后，执行下一个指令处理。但实际的写入会超过 1 个周期。因此，为了将从 CPU 向 DSCTR 写入的值准确反映在 SLEEP 指令，请在读出 DSCTR 后执行 SLEEP 指令。

向深度待机模式转移的顺序如下所示。另外，该流程如**图 32.2** 所示。

1. 深度待机模式时，如果在内置RAM （保持用）区域保持数据，就在DSRTR寄存器设定H'09。
2. 对所需保持的内置RAM （保持用）区域，设定DSCTR寄存器的RRAMKP3 ～ RRAMKP0位。将需要保持的程序传送到设定的内置RAM （保持用）区域。
3. 通过中断从深度待机模式返回时，设定DSSSR的相应位，选择用于返回的引脚。此时设定返回引脚的输入信号检测模式（通过中断控制器 （INTC）的中断控制寄存器0、 1（ICR0、 ICR1）设定）。在深度待机模式时，只有上升沿或下降沿的设定才有效 （IRQ的低电平及双边沿检测无效）。
4. 对保持的内置RAM （保持用）的各页，执行任意相同地址的读/写。不执行时，最后写入的数据有可能不能写入内置RAM（保持用）。之后有向内置RAM（保持用）的写入时，在最后向内置RAM（保持用）进行写入后，执行该处理。
5. STBCR寄存器的STBY位和DEEP位设定为1。
6. 清除DSFR寄存器的标志后，读取DSFR寄存器。之后执行SLEEP指令。

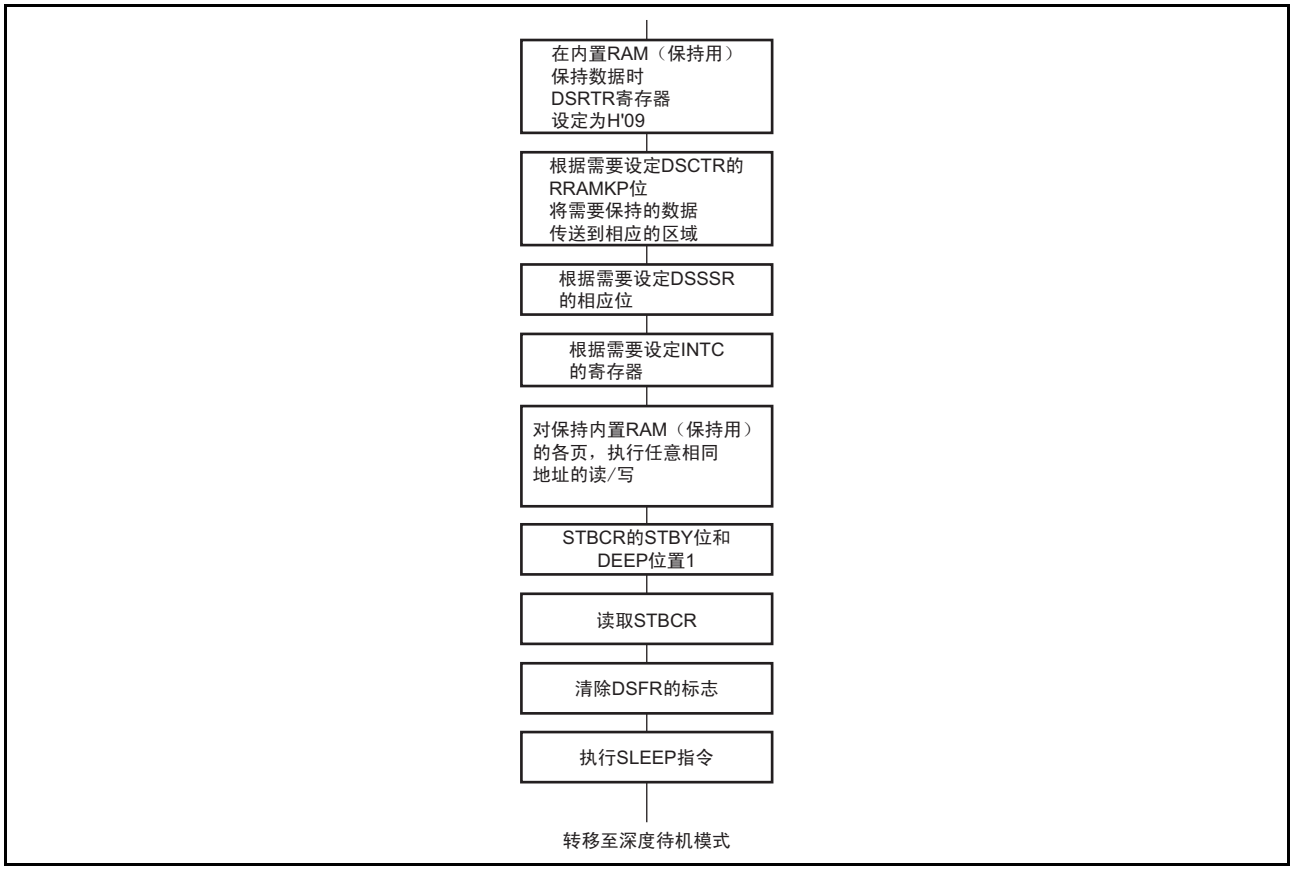


图 32.2　深度待机模式转移流程

（2） 解除深度待机模式

通过中断 （NMI、分配到 PE11 ～ PE4 的 IRQ）及复位 （手动复位、上电复位）解除深度待机模式。通过 NMI 中断或 IRQ 中断解除时，不执行中断异常处理而执行上电复位异常处理。同样手动复位也为上电复位异常处理。深度待机模式解除的流程如图 32.3 所示。

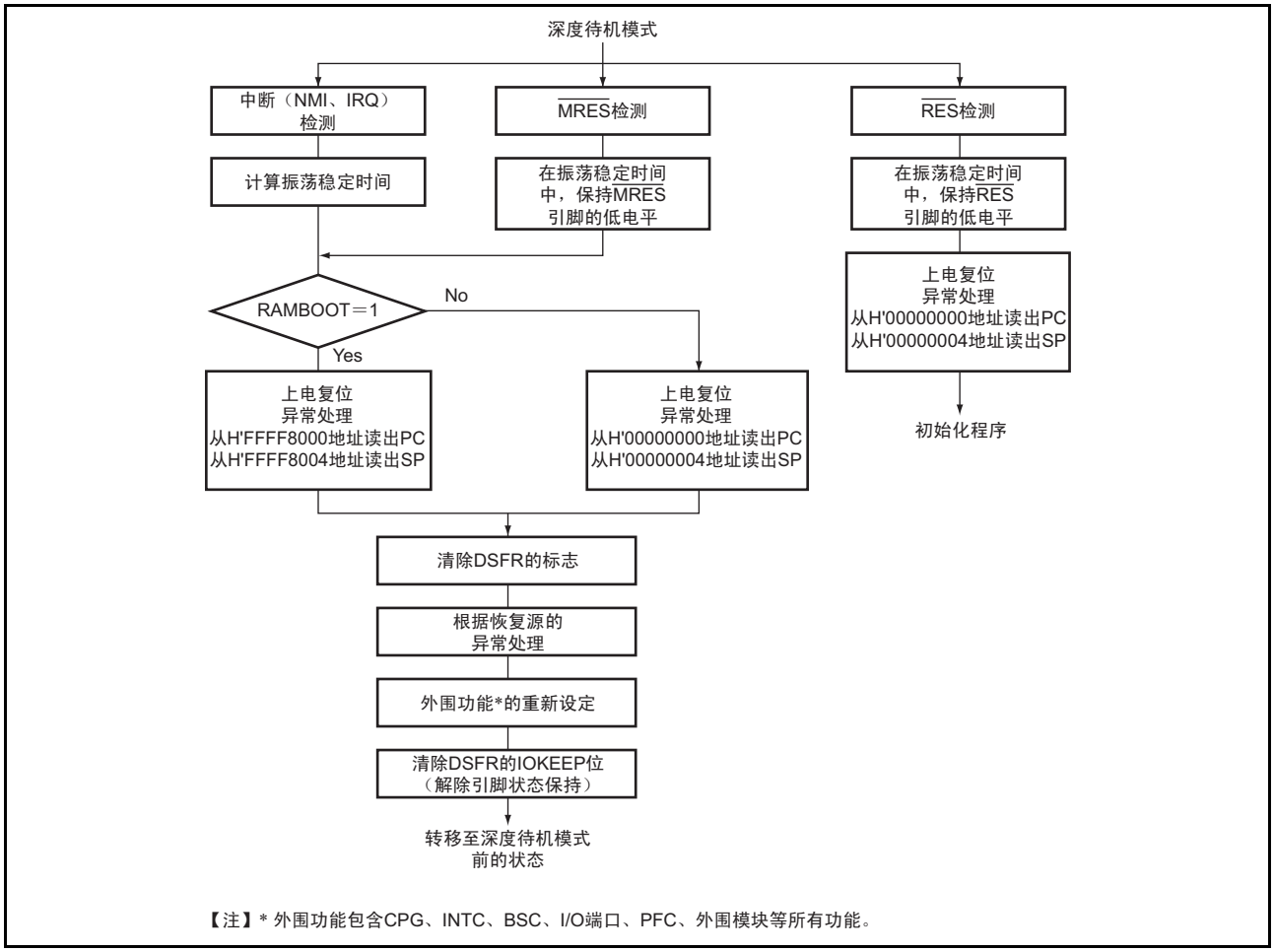


图 32.3　深度待机模式解除流程

- 通过中断解除

  如果检测出 NMI 引脚的下降沿或上升沿 （通过中断控制器 （INTC）的中断控制寄存器 0 （ICR0）的 NMI 边沿选择位 （NMIE）选择）、 IRQ 引脚 （IRQ7 ～ IRQ0 分配到 PE11 ～ PE4）的下降沿或上升沿（通过中断控制器（INTC）的中断控制寄存器 1（ICR1）的 IRQn 检测选择位（IRQn1S ～ IRQn0S）选择），在电源稳定等待时间后就开始时钟振荡。在经过振荡稳定时间后，解除深度待机模式，并执行上电复位异常处理。

  从检测出中断到解除深度待机模式为止，CKIO 引脚时钟输出的相位有不稳定的情况。而且，如果通过 NMI 引脚的下降沿解除深度待机模式，进入深度待机模式时 （时钟停止时）的 NMI 引脚为高电平，返回深度待机模式时（启动振荡稳定后的时钟时） NMI 引脚为低电平。而且，通过 NMI 引脚的上升沿解除深度待机模式时，进入深度待机模式时 （时钟停止时） NMI 引脚为低电平，返回深度待机模式时（启动振荡稳定后的时钟时） NMI 引脚为高电平 （与 IRQ 引脚时相同）。

- 通过复位解除
  如果$\overline{\text{RES}}$引脚为低电平，就解除深度待机模式，并转移至上电复位。之后如果$\overline{\text{RES}}$引脚为高电平，就开始上电复位异常处理。在时钟模式0、1、3时，如果$\overline{\text{RES}}$引脚为低电平，就从CKIO引脚开始输出内部时钟。
  如果$\overline{\text{MRES}}$引脚为低电平，就解除深度待机模式，并转移至上电复位。之后如果$\overline{\text{MRES}}$引脚为高电平，就开始上电复位异常处理。在时钟模式0、1、3时，如果$\overline{\text{MRES}}$引脚为高电平，就从CKIO引脚开始输出内部时钟。
  时钟振荡稳定前，$\overline{\text{RES}}$引脚或$\overline{\text{MRES}}$引脚保持为低电平。

（3） 解除深度待机模式后的运行

通过 DSCTR2 寄存器的 RAMBOOT 位的设定，能选择启动外部总线或启动内置 RAM （保持用）。通过 CS0KEEPE 位的设定，即使解除深度待机模式后也能保持外部总线控制引脚的状态。通过各位解除深度待机模式后的引脚状态如**表 32.3** 所示。外部总线控制引脚一览如**表 32.4** 所示。

表 32.3 通过 DSCTR2 寄存器设定解除深度待机后的引脚状态和启动方法

CS0KEEPE 位	RAMBOOT 位	启动方法	解除深度待机后的引脚状态
0	0	外部总线	不保持外部总线控制引脚。 IOKEEP 位清除时，解除其他引脚的保持状态。
0	1	内置 RAM （保持用）	不保持外部总线控制引脚。 解除深度待机后，解除外部总线控制引脚的保持状态。 IOKEEP 位清除时，解除其他引脚的保持状态。
1	0	—	设定禁止
1	1	内置 RAM （保持用）	保持外部总线控制引脚。 IOKEEP 位清除时，解除外部总线控制引脚及其他引脚的保持状态。

表 32.4 外部总线控制引脚一览

运行模式 0 （启动外部 16 位总线）	运行模式 1 （启动外部 32 位总线）
A[20:1] D[15:0] $\overline{\text{CS0}}$、$\overline{\text{RD}}$、 CKIO	A[20:2] D[31:0] $\overline{\text{CS0}}$、$\overline{\text{RD}}$、 CKIO

通过中断（NMI、IRQ）及手动复位解除深度待机模式时，可通过深度待机解除源标志寄存器（DSFR）确认通过哪种中断进行解除。

向深度待机模式转移时，引脚保持之前的状态。但在外部总线启动模式时，解除深度待机模式后，解除保持外部总线控制引脚的状态，因此可取出程序。解除深度待机模式后，从 DSFR 寄存器的 IOKEEP 位读出 1 到写 0 前，继续保持其他引脚的状态。在内置 RAM （保持用）启动模式，解除深度待机模式后，从 DSFR 寄存器的 IOKEEP 位读出 1 到写 0 前，继续保持外部总线控制引脚和其他引脚的状态。返回深度待机模式转移前的状态需要重新设定外围功能。外围功能包含 CPG、INTC、BSC、I/O 端口、PFC、外围模块等所有功能。重新设定后，通过 IOKEEP 位读 1 后写 0，可解除保持引脚的状态，并成为转移为深度待机模式前的状态。

（4） 深度待机模式转移时的注意事项

设定深度待机模式后，屏蔽除通过深度待机解除源设定的中断。即：只接收通过深度待机解除源设定的中断，但设定多个解除源，并输入多个解除源时，多个解除源标志被置位。

如果同时发生深度待机模式转移的 SLEEP 指令和 NMI、IRQ 的中断及手动复位，就有可能出现接受中断，并从深度待机模式返回的情况。

32.3.5 模块待机功能

（1） 向模块待机功能转移

通过将待机控制寄存器的各 MSTP 位置 1，可停止向各位对应的内置外围模块提供时钟。通过使用该功能，可降低程序执行状态及睡眠模式时的功耗。在转移至模块待机状态前必须禁止该模块。另外，不能对模块待机状态下的模块进行寄存器的存取。

寄存器的状态请参照 “34.3 各运行模式下的寄存器状态一览”。

（2） 解除模块待机功能

通过将各 MSTP 位清 0 或者上电复位 （仅限 RTC、 H-UDI、 UBC、 DMAC）解除模块待机功能。通过将各 MSTP 位清 0，解除模块待机功能时，请读出相应位并确认已经清 0。

32.4 使用时的注意事项

对与低功耗模式相关的寄存器进行写操作时，请注意以下情况。

从 CPU 写入与低功耗模式相关的寄存器时，在 CPU 执行写指令后，不等待实际寄存器的写入结束，而执行后续的指令。

执行后续指令时，如果需要通过向寄存器的写入反映改变，请在寄存器写指令和后续指令之间对相同寄存器进行虚读。

第 33 章　用户调试接口 （H-UDI）

为了支持仿真器，本 LSI 内置用户调试接口 （H-UDI）。

33.1　特点

用户调试接口 （H-UDI），具备有复位及中断请求功能。
本 LSI 的 H-UDI 用于连接仿真器。
与仿真器的连接方法，请参照仿真器手册。
H-UDI 的框图如图 33.1 所示。

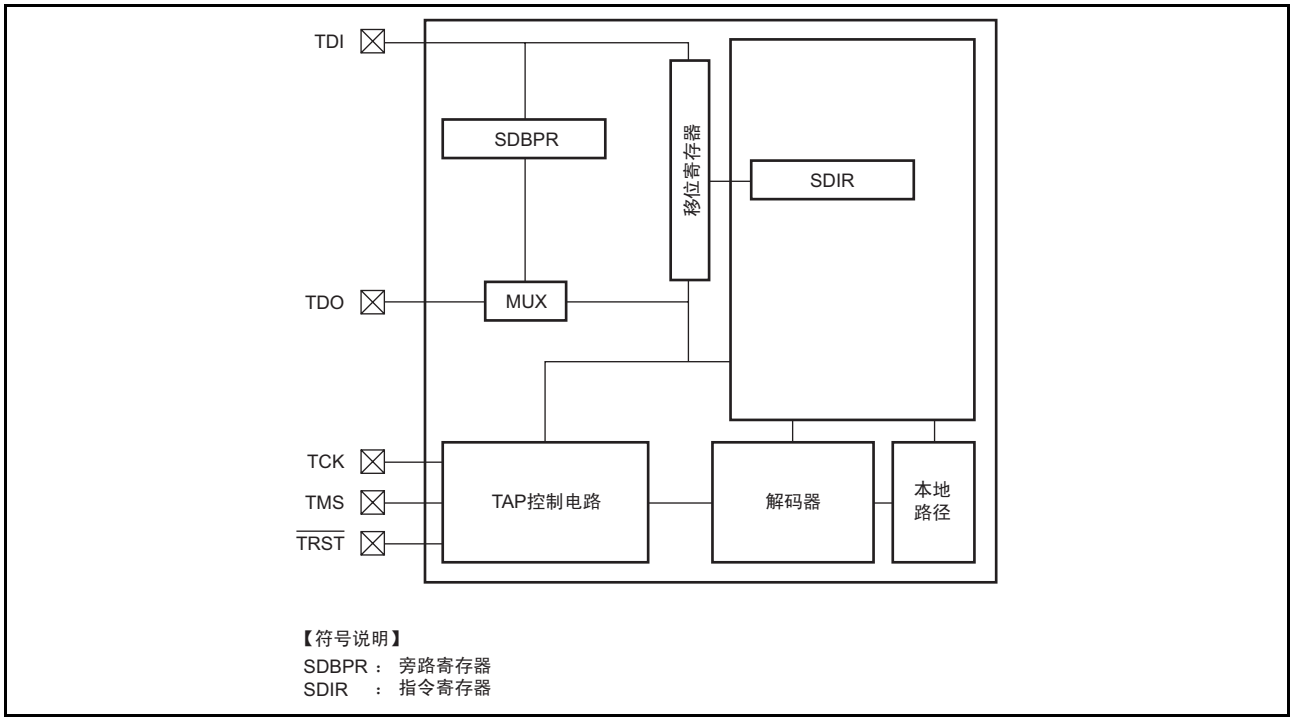


图 33.1　H-UDI 的框图

33.2　输入 / 输出引脚

表 33.1　引脚结构

名称	引脚名	输入 / 输出	功能
H-UDI 串行数据输入用时钟引脚	TCK	输入	与该时钟同步，从数据输入引脚 （TDI）向 H-UDI 串行提供数据，从数据输出引脚 （TDO）输出数据。
模式选择输入引脚	TMS	输入	通过与 TCK 同步改变该信号，决定 TAP 控制电路的状态。协议请参照图 33.2。
H-UDI 复位输入引脚	$\overline{\text{TRST}}$	输入	与 TCK 异步，接收输入，在低电平时复位 H-UDI。与是否利用 H-UDI 功能无关，必须在通电时使 $\overline{\text{TRST}}$ 保持一定时间的低电平。关于复位结构的详细内容，请参照 “33.4.2 复位结构”。
H-UDI 串行数据输入引脚	TDI	输入	通过与 TCK 同步改变该引脚，向 H-UDI 发送数据。
H-UDI 串行数据输出引脚	TDO	输出	通过与 TCK 同步读出该引脚，从 H-UDI 读取数据。数据输出时序的初始值为下降沿同步，但是能通过对 SDIR 输入 “TDO 变化时序切换”命令来变更为上升沿同步。详细请参照“33.4.3 TDO 输出时序”。
ASE 模式选择引脚	$\overline{\text{ASEMD}}$*	输入	在 $\overline{\text{RES}}$ 引脚断言期间，如果对 $\overline{\text{ASEMD}}$ 引脚输入低电平，就变为 ASE 模式；输入高电平，就变为产品芯片模式。在 ASE 模式中，能使用仿真器专用的功能。向 $\overline{\text{ASEMD}}$ 引脚输入的电平，请在使 $\overline{\text{RES}}$ 引脚取反后，保持最低 1 个周期。

【注】 *　不使用仿真器时，请固定为高电平。

33.3　寄存器说明

H-UDI 有以下寄存器。

表 33.2　寄存器结构

寄存器名	简称	R/W	初始值	地址	存取长度
旁路寄存器	SDBPR	—	—	—	—
指令寄存器	SDIR	R	H'EFFD	H'FFFE2000	16

33.3.1　旁路寄存器 （SDBPR）

SDBPR 是 CPU 不能存取的 1 位寄存器。将 SDIR 设定为 BYPASS 模式时，SDBPR 就被连接到 H-UDI 引脚的 TDI 和 TDO 之间。初始值不定。

33.3.2　指令寄存器 （SDIR）

SDIR 是 16 位只读寄存器。在 $\overline{\text{TRST}}$ 断言或者 TAP 的 Test-Logic-Reset 状态时被初始化。同时，与 CPU 的模式无关，能从 H-UDI 进行写入。向该寄存器设定保留命令时，不能保证运行。初始值为 H'EFFD。

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

【注】* TI[7:0]的初始值为保留值,但是设定命令时必须设定非保留值。

位	位名	初始值	R/W	说明
15 ～ 8	TI[7:0]	11101111*	R	测试指令 通过 TDI 的串行输入， H-UDI 的指令被传送到 SDIR。 命令请参照表 33.3。
7 ～ 2	—	均为 1	R	保留位 读取时总是读出 1。
1	—	0	R	保留位 读取时总是读出 0。
0	—	1	R	保留位 读取时总是读出 1。

表 33.3　H-UDI 命令

bit15 ～ 8								说明
TI7	TI6	TI5	TI4	TI3	TI2	TI1	TI0	
0	1	1	0	—	—	—	—	H-UDI 复位取反
0	1	1	1	—	—	—	—	H-UDI 复位断言
1	0	0	1	1	1	0	0	TDO 变化时序切换
1	0	1	1	—	—	—	—	H-UDI 中断
1	1	1	1	—	—	—	—	BYPASS 模式
上述以外								保留

33.4　运行说明

33.4.1　TAP 控制器

TAP 控制器内部状态如图 33.2 所示。

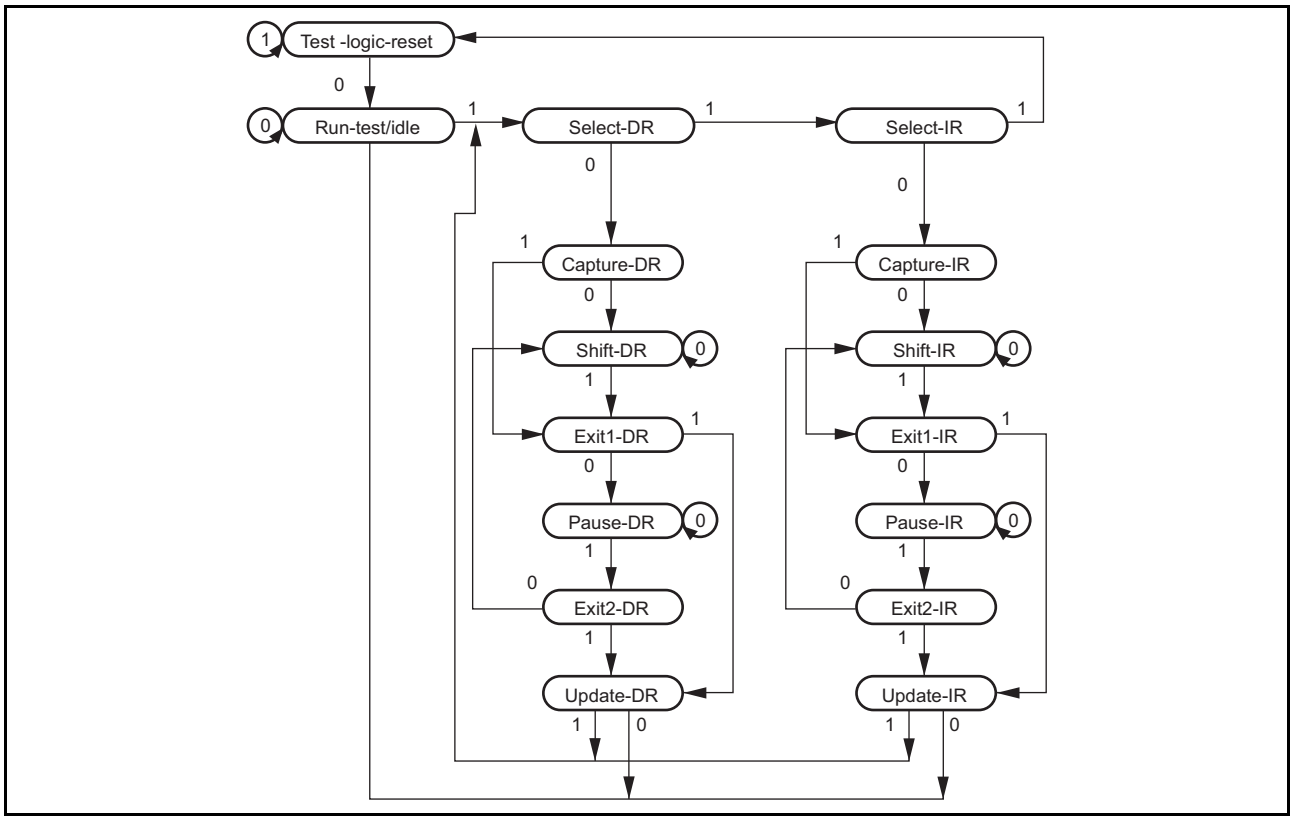


图 33.2　TAP 控制器状态转移图

【注】 转移条件为 TCK 上升沿的 TMS 值。 TDI 值在 TCK 的上升沿进行采样，在 TCK 的下降沿发生移位。关于 TDO 值的变化时序，请参照 “33.4.3 TDO 输出时序”。 TDO 不是 Shift-DR、 Shift-IR 状态时为高阻抗状态。通过 $\overline{\text{TRST}}$ 断言，转移至 Test-Logic-Reset 状态与 TCK 异步。

33.4.2　复位结构

表 33.4　复位结构

$\overline{\text{ASEMD}}$*1	$\overline{\text{RES}}$	$\overline{\text{TRST}}$	芯片状态
H	L	L	上电复位及 H-UDI 的复位
		H	上电复位
	H	L	只有 H-UDI 复位
		H	正常运行
L	L	L	复位保持 *2
		H	上电复位
	H	L	只有 H-UDI 复位
		H	正常运行

【注】　*1　选择产品芯片模式和 ASE 模式的设定。
$\overline{\text{ASEMD}}$=H、产品芯片模式
$\overline{\text{ASEMD}}$=L、 ASE模式

*2　ASE 模式下， $\overline{\text{RES}}$ 取反时，如果 $\overline{\text{TRST}}$ 引脚为低电平，就成为复位保持状态。该状态下不启动 CPU。此后，如果将 $\overline{\text{TRST}}$ 设定为高电平， H-UDI 运行就为有效，但是不启动 CPU。通过上电复位，解除复位保持状态。

33.4.3　TDO 输出时序

在 TCK 的下降沿同步输出 TDO 变化时序的初始值。但是通过从 H-UDI 引脚对 SDIR 设定 “TDO 变化时序切换”命令，并通过 Update-IR，使 TDO 的变化时序与 TCK 的上升沿同步。之后，在 TCK 的下降沿同步输出 TDO 的变化时序时，$\overline{\text{TRST}}$ 引脚需要与上电复位同时断言。通过 $\overline{\text{RES}}$ 引脚上电复位时，在 $\overline{\text{RES}}$ 引脚取反后，同步复位也需要在芯片内部运行一定时间。因此，如果在 $\overline{\text{RES}}$ 引脚取反后马上使 $\overline{\text{TRST}}$ 引脚断言时，“TDO 变化时序切换”命令就被清除，并在 TCK 的下降沿同步输出 TDO 的变化时序。为了防止该情况的发生， $\overline{\text{RES}}$ 引脚和 $\overline{\text{TRST}}$ 引脚之间的信号变化必须设置大于等于 20tcyc 的间隔。

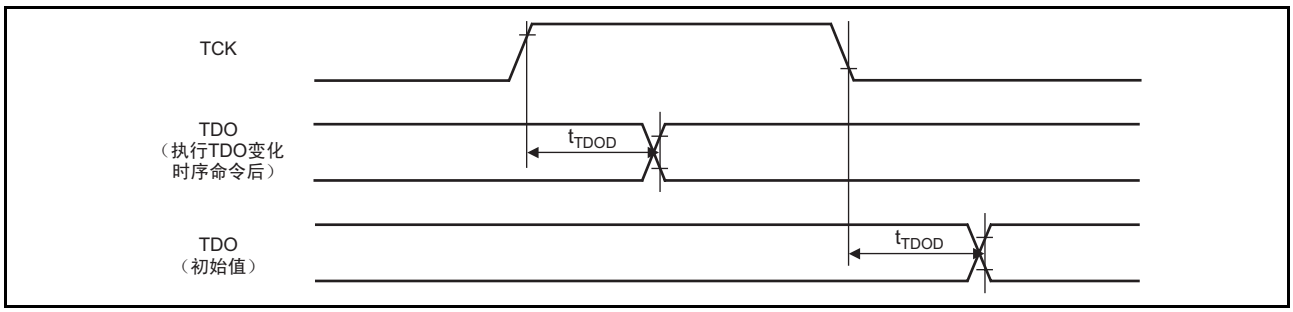


图 33.3　H-UDI 数据传送时序

33.4.4　H-UDI 复位

通过对 SDIR 设定 H-UDI 复位断言命令，产生 H-UDI 复位。 H-UDI 复位是与上电复位相同的复位。通过设定 H-UDI 复位取反命令，解除 H-UDI 复位。 H-UDI 复位断言命令与 H-UDI 复位取反命令之间所需的时间与 $\overline{\text{RES}}$ 引脚保持低电平的时间 （为了请求上电复位）相同。

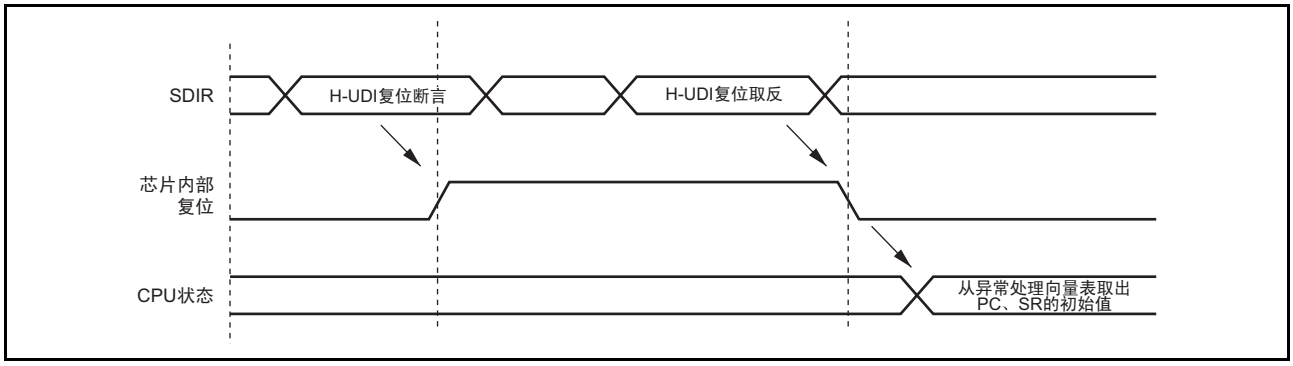


图 33.4　H-UDI 复位

33.4.5　H-UDI 中断

H-UDI 中断功能为通过对 SDIR 设定来自 H-UDI 的命令来发生中断。 H-UDI 中断是普通异常 / 中断运行，从异常处理向量表取出异常服务程序开始地址，转移到该地址并执行。该中断请求有固定优先级 15。

在睡眠模式中也能接受 H-UDI 中断，但是在软件待机模式中不能接受 H-UDI 中断。

33.5　使用时的注意事项

1. 除非其他命令从H-UDI再次被设定，否则H-UDI命令一旦被设定就不会改变。如果连续设定相同命令，必须在设定对芯片运行没有影响的命令 （BYPASS模式等）后再次设定命令。
2. 软件待机模式、H-UDI模块待机状态中，H-UDI的全部功能均不能使用。同时，为了在待机模式前后保持TAP的状态，需要在待机模式转移时预先设置TCK为高电平。
3. 与是否使用H-UDI无关，在通电时，如果通过$\overline{\text{RES}}$引脚断言解除深度待机，就必须将$\overline{\text{TRST}}$引脚设置为低电平，并初期化为H-UDI。
4. $\overline{\text{RES}}$引脚与$\overline{\text{TRST}}$引脚间的信号变化必须设置大于等于20tcyc的间隔。
5. $\overline{\text{TRST}}$引脚取反后，运行TAP控制器时，必须设置大于等于200ns的间隔。

第 34 章 寄存器一览

寄存器一览汇总了有关本 LSI 的内部 I/O 寄存器信息，通过以下方式进行说明。

（1） 寄存器地址一览 （按功能模块、手册章节编号顺序）

- 按功能模块、手册的章节编号顺序记载寄存器。
- 请勿存取寄存器地址一览中未记载的保留地址。
- 在地址为16位或者32位时，以大端法为前提，记载MSB侧的地址。

（2） 寄存器位一览

- 按 “寄存器地址一览 （按功能模块、手册章节编号顺序）” 的顺序，记载各寄存器的位结构。
- 在位名栏用 “–” 表示保留位。
- 表中位名栏为空白的部分，表示该寄存器全部分配给计数器或者数据。

（3） 各运行模式下的寄存器状态一览

- 按 “寄存器地址一览 （按功能模块、手册章节编号顺序）” 的顺序，记载寄存器的状态。
- 初始化的各位的状态，请参照对应章节的寄存器说明。
- 表示基本运行模式时的寄存器状态。有内部外围模块固有的复位时，请参照内部外围模块的章节。

（4） 内部外围模块的寄存器写入时的注意事项

从内部总线存取内部外围模块的寄存器时，需要 2 周期以上的外围模块时钟（Pϕ）。另一方面，从 CPU 向内部外围寄存器进行写入时，在寄存器写入完成前， CPU 就执行后续的指令。

以降低功耗时向软件待机模式的转移为例子进行说明。为了进行转移，需要在将 STBCR 寄存器的 STBY 位置 1 后执行 SLEEP 指令。但是必须在执行 SLEEP 指令前虚读 STBCR 寄存器。如果不进行虚读，由于 STBY 位被置 1 之前 CPU 执行 SLEEP 指令，因此不转移至需要的软件待机模式而是转移至睡眠模式。为了等待对 STBY 位进行写入，需要对 STBCR 寄存器进行虚读。

如此例所示，在执行后续指令时需要通过内部外围寄存器反映变更时，请在寄存器写指令后进行相同寄存器的虚读，之后再执行需要的后续指令。

34.1 寄存器地址一览 （按功能模块、手册章节编号顺序）

模块名	寄存器名	简称	位数	地址	存取长度
CPG	频率控制寄存器	FRQCR	16	H'FFFE0010	16
INTC	中断控制寄存器 0	ICR0	16	H'FFFE0800	16、32
	中断控制寄存器 1	ICR1	16	H'FFFE0802	16、32
	中断控制寄存器 2	ICR2	16	H'FFFE0804	16、32
	IRQ 中断请求寄存器	IRQRR	16	H'FFFE0806	16、32
	PINT 中断允许寄存器	PINTER	16	H'FFFE0808	16、32
	PINT 中断请求寄存器	PIRR	16	H'FFFE080A	16、32
	存储体控制寄存器	IBCR	16	H'FFFE080C	16、32
	存储体编号寄存器	IBNR	16	H'FFFE080E	16、32
	中断优先级设定寄存器 01	IPR01	16	H'FFFE0818	16、32
	中断优先级设定寄存器 02	IPR02	16	H'FFFE081A	16、32
	中断优先级设定寄存器 05	IPR05	16	H'FFFE0820	16、32
	中断优先级设定寄存器 06	IPR06	16	H'FFFE0C00	16、32
	中断优先级设定寄存器 07	IPR07	16	H'FFFE0C02	16、32
	中断优先级设定寄存器 08	IPR08	16	H'FFFE0C04	16、32
	中断优先级设定寄存器 09	IPR09	16	H'FFFE0C06	16、32
	中断优先级设定寄存器 10	IPR10	16	H'FFFE0C08	16、32
	中断优先级设定寄存器 11	IPR11	16	H'FFFE0C0A	16、32
	中断优先级设定寄存器 12	IPR12	16	H'FFFE0C0C	16、32
	中断优先级设定寄存器 13	IPR13	16	H'FFFE0C0E	16、32
	中断优先级设定寄存器 14	IPR14	16	H'FFFE0C10	16、32
	中断优先级设定寄存器 15	IPR15	16	H'FFFE0C12	16、32
	中断优先级设定寄存器 16	IPR16	16	H'FFFE0C14	16、32
	中断优先级设定寄存器 17	IPR17	16	H'FFFE0C16	16、32
UBC	断点地址寄存器 _0	BAR_0	32	H'FFFC0400	32
	断点地址屏蔽寄存器 _0	BAMR_0	32	H'FFFC0404	32
	断点数据寄存器 _0	BDR_0	32	H'FFFC0408	32
	断点数据屏蔽寄存器 _0	BDMR_0	32	H'FFFC040C	32
	断点地址寄存器 _1	BAR_1	32	H'FFFC0410	32
	断点地址屏蔽寄存器 _1	BAMR_1	32	H'FFFC0414	32
	断点数据寄存器 _1	BDR_1	32	H'FFFC0418	32
	断点数据屏蔽寄存器 _1	BDMR_1	32	H'FFFC041C	32
	断点总线周期寄存器 _0	BBR_0	16	H'FFFC04A0	16

模块名	寄存器名	简称	位数	地址	存取长度
UBC	断点总线周期寄存器 _1	BBR_1	16	H'FFFC04B0	16
	断点控制寄存器	BRCR	32	H'FFFC04C0	32
高速缓存	高速缓存控制寄存器 1	CCR1	32	H'FFFC1000	32
	高速缓存控制寄存器 2	CCR2	32	H'FFFC1004	32
BSC	通用控制寄存器	CMNCR	32	H'FFFC0000	32
	CS0 空间总线控制寄存器	CS0BCR	32	H'FFFC0004	32
	CS1 空间总线控制寄存器	CS1BCR	32	H'FFFC0008	32
	CS2 空间总线控制寄存器	CS2BCR	32	H'FFFC000C	32
	CS3 空间总线控制寄存器	CS3BCR	32	H'FFFC0010	32
	CS4 空间总线控制寄存器	CS4BCR	32	H'FFFC0014	32
	CS5 空间总线控制寄存器	CS5BCR	32	H'FFFC0018	32
	CS6 空间总线控制寄存器	CS6BCR	32	H'FFFC001C	32
	CS7 空间总线控制寄存器	CS7BCR	32	H'FFFC0020	32
	CS0 空间等待控制寄存器	CS0WCR	32	H'FFFC0028	32
	CS1 空间等待控制寄存器	CS1WCR	32	H'FFFC002C	32
	CS2 空间等待控制寄存器	CS2WCR	32	H'FFFC0030	32
	CS3 空间等待控制寄存器	CS3WCR	32	H'FFFC0034	32
	CS4 空间等待控制寄存器	CS4WCR	32	H'FFFC0038	32
	CS5 空间等待控制寄存器	CS5WCR	32	H'FFFC003C	32
	CS6 空间等待控制寄存器	CS6WCR	32	H'FFFC0040	32
	CS7 空间等待控制寄存器	CS7WCR	32	H'FFFC0044	32
	SDRAM 控制寄存器	SDCR	32	H'FFFC004C	32
	刷新定时器控制 / 状态寄存器	RTCSR	32	H'FFFC0050	32
	刷新定时器计数器	RTCNT	32	H'FFFC0054	32
	刷新时间常数寄存器	RTCOR	32	H'FFFC0058	32
DMAC	DMA 源地址寄存器 _0	SAR0	32	H'FFFE1000	16、32
	DMA 目标地址寄存器 _0	DAR0	32	H'FFFE1004	16、32
	DMA 传送计数寄存器 _0	DMATCR0	32	H'FFFE1008	16、32
	DMA 沟道控制寄存器 _0	CHCR0	32	H'FFFE100C	8、16、32
	DMA 再装入源地址寄存器 _0	RSAR0	32	H'FFFE1100	16、32
	DMA 再装入目标地址寄存器 _0	RDAR0	32	H'FFFE1104	16、32

模块名	寄存器名	简称	位数	地址	存取长度
DMAC	DMA 再装入传送计数寄存器 _0	RDMATCR0	32	H'FFFE1108	16、32
	DMA 源地址寄存器 _1	SAR1	32	H'FFFE1010	16、32
	DMA 目标地址寄存器 _1	DAR1	32	H'FFFE1014	16、32
	DMA 传送计数寄存器 _1	DMATCR1	32	H'FFFE1018	16、32
	DMA 通道控制寄存器 _1	CHCR1	32	H'FFFE101C	8、16、32
	DMA 再装入源地址寄存器 _1	RSAR1	32	H'FFFE1110	16、32
	DMA 再装入目标地址寄存器 _1	RDAR1	32	H'FFFE1114	16、32
	DMA 再装入传送计数寄存器 _1	RDMATCR1	32	H'FFFE1118	16、32
	DMA 源地址寄存器 _2	SAR2	32	H'FFFE1020	16、32
	DMA 目标地址寄存器 _2	DAR2	32	H'FFFE1024	16、32
	DMA 传送计数寄存器 _2	DMATCR2	32	H'FFFE1028	16、32
	DMA 通道控制寄存器 _2	CHCR2	32	H'FFFE102C	8、16、32
	DMA 再装入源地址寄存器 _2	RSAR2	32	H'FFFE1120	16、32
	DMA 再装入目标地址寄存器 2	RDAR2	32	H'FFFE1124	16、32
	DMA 再装入传送计数寄存器 _2	RDMATCR2	32	H'FFFE1128	16、32
	DMA 源地址寄存器 _3	SAR3	32	H'FFFE1030	16、32
	DMA 目标地址寄存器 _3	DAR3	32	H'FFFE1034	16、32
	DMA 传送计数寄存器 _3	DMATCR3	32	H'FFFE1038	16、32
	DMA 通道控制寄存器 _3	CHCR3	32	H'FFFE103C	8、16、32
	DMA 再装入源地址寄存器 _3	RSAR3	32	H'FFFE1130	16、32
	DMA 再装入目标地址寄存器 _3	RDAR3	32	H'FFFE1134	16、32
	DMA 再装入传送计数寄存器 _3	RDMATCR3	32	H'FFFE1138	16、32
	DMA 源地址寄存器 _4	SAR4	32	H'FFFE1040	16、32
	DMA 目标地址寄存器 _4	DAR4	32	H'FFFE1044	16、32
	DMA 传送计数寄存器 _4	DMATCR4	32	H'FFFE1048	16、32
	DMA 通道控制寄存器 _4	CHCR4	32	H'FFFE104C	8、16、32
	DMA 再装入源地址寄存器 _4	RSAR4	32	H'FFFE1140	16、32

模块名	寄存器名	简称	位数	地址	存取长度
DMAC	DMA 再装入目标地址寄存器 _4	RDAR4	32	H'FFFE1144	16、32
	DMA 再装入传送计数寄存器 _4	RDMATCR4	32	H'FFFE1148	16、32
	DMA 源地址寄存器 _5	SAR5	32	H'FFFE1050	16、32
	DMA 目标地址寄存器 _5	DAR5	32	H'FFFE1054	16、32
	DMA 传送计数寄存器 _5	DMATCR5	32	H'FFFE1058	16、32
	DMA 通道控制寄存器 _5	CHCR5	32	H'FFFE105C	8、16、32
	DMA 再装入源地址寄存器 _5	RSAR5	32	H'FFFE1150	16、32
	DMA 再装入目标地址寄存器 _5	RDAR5	32	H'FFFE1154	16、32
	DMA 再装入传送计数寄存器 _5	RDMATCR5	32	H'FFFE1158	16、32
	DMA 源地址寄存器 _6	SAR6	32	H'FFFE1060	16、32
	DMA 目标地址寄存器 _6	DAR6	32	H'FFFE1064	16、32
	DMA 传送计数寄存器 _6	DMATCR6	32	H'FFFE1068	16、32
	DMA 通道控制寄存器 _6	CHCR6	32	H'FFFE106C	8、16、32
	DMA 再装入源地址寄存器 _6	RSAR6	32	H'FFFE1160	16、32
	DMA 再装入目标地址寄存器 _6	RDAR6	32	H'FFFE1164	16、32
	DMA 再装入传送计数寄存器 _6	RDMATCR6	32	H'FFFE1168	16、32
	DMA 源地址寄存器 _7	SAR7	32	H'FFFE1070	16、32
	DMA 目标地址寄存器 _7	DAR7	32	H'FFFE1074	16、32
	DMA 传送计数寄存器 _7	DMATCR7	32	H'FFFE1078	16、32
	DMA 通道控制寄存器 _7	CHCR7	32	H'FFFE107C	8、16、32
	DMA 再装入源地址寄存器 _7	RSAR7	32	H'FFFE1170	16、32
	DMA 再装入目标地址寄存器 _7	RDAR7	32	H'FFFE1174	16、32
	DMA 再装入传送计数寄存器 _7	RDMATCR7	32	H'FFFE1178	16、32
	DMA 操作寄存器	DMAOR	16	H'FFFE1200	8、16
	DMA 扩展资源选择器 0	DMARS0	16	H'FFFE1300	16
	DMA 扩展资源选择器 1	DMARS1	16	H'FFFE1304	16
	DMA 扩展资源选择器 2	DMARS2	16	H'FFFE1308	16
	DMA 扩展资源选择器 3	DMARS3	16	H'FFFE130C	16

模块名	寄存器名	简称	位数	地址	存取长度
MTU2	定时器控制寄存器 _0	TCR_0	8	H'FFFE4300	8
	定时器模式寄存器 _0	TMDR_0	8	H'FFFE4301	8
	定时器 I/O 控制寄存器 H_0	TIORH_0	8	H'FFFE4302	8
	定时器 I/O 控制寄存器 L_0	TIORL_0	8	H'FFFE4303	8
	定时器中断允许寄存器 _0	TIER_0	8	H'FFFE4304	8
	定时器状态寄存器 _0	TSR_0	8	H'FFFE4305	8
	定时器计数器 _0	TCNT_0	16	H'FFFE4306	16
	定时器通用寄存器 A_0	TGRA_0	16	H'FFFE4308	16
	定时器通用寄存器 B_0	TGRB_0	16	H'FFFE430A	16
	定时器通用寄存器 C_0	TGRC_0	16	H'FFFE430C	16
	定时器通用寄存器 D_0	TGRD_0	16	H'FFFE430E	16
	定时器通用寄存器 E_0	TGRE_0	16	H'FFFE4320	16
	定时器通用寄存器 F_0	TGRF_0	16	H'FFFE4322	16
	定时器中断允许寄存器 2_0	TIER2_0	8	H'FFFE4324	8
	定时器状态寄存器 2_0	TSR2_0	8	H'FFFE4325	8
	定时器缓冲器运行传送模式寄存器 _0	TBTM_0	8	H'FFFE4326	8
	定时器控制寄存器 _1	TCR_1	8	H'FFFE4380	8
	定时器模式寄存器 _1	TMDR_1	8	H'FFFE4381	8
	定时器 I/O 控制寄存器 _1	TIOR_1	8	H'FFFE4382	8
	定时器中断允许寄存器 _1	TIER_1	8	H'FFFE4384	8
	定时器状态寄存器 _1	TSR_1	8	H'FFFE4385	8
	定时器计数器 _1	TCNT_1	16	H'FFFE4386	16
	定时器通用寄存器 A_1	TGRA_1	16	H'FFFE4388	16
	定时器通用寄存器 B_1	TGRB_1	16	H'FFFE438A	16
	定时器输入捕捉控制寄存器	TICCR	8	H'FFFE4390	8
	定时器控制寄存器 _2	TCR_2	8	H'FFFE4000	8
	定时器模式寄存器 _2	TMDR_2	8	H'FFFE4001	8
	定时器 I/O 控制寄存器 _2	TIOR_2	8	H'FFFE4002	8
	定时器中断允许寄存器 _2	TIER_2	8	H'FFFE4004	8
	定时器状态寄存器 _2	TSR_2	8	H'FFFE4005	8
	定时器计数器 _2	TCNT_2	16	H'FFFE4006	16
	定时器通用寄存器 A_2	TGRA_2	16	H'FFFE4008	16
	定时器通用寄存器 B_2	TGRB_2	16	H'FFFE400A	16
	定时器控制寄存器 _3	TCR_3	8	H'FFFE4200	8
	定时器模式寄存器 _3	TMDR_3	8	H'FFFE4202	8

模块名	寄存器名	简称	位数	地址	存取长度
MTU2	定时器 I/O 控制寄存器 H_3	TIORH_3	8	H'FFFE4204	8
	定时器 I/O 控制寄存器 L_3	TIORL_3	8	H'FFFE4205	8
	定时器中断允许寄存器 _3	TIER_3	8	H'FFFE4208	8
	定时器状态寄存器 _3	TSR_3	8	H'FFFE422C	8
	定时器计数器 _3	TCNT_3	16	H'FFFE4210	16
	定时器通用寄存器 A_3	TGRA_3	16	H'FFFE4218	16
	定时器通用寄存器 B_3	TGRB_3	16	H'FFFE421A	16
	定时器通用寄存器 C_3	TGRC_3	16	H'FFFE4224	16
	定时器通用寄存器 D_3	TGRD_3	16	H'FFFE4226	16
	定时器缓冲器运行传送模式寄存器 _3	TBTM_3	8	H'FFFE4238	8
	定时器控制寄存器 _4	TCR_4	8	H'FFFE4201	8
	定时器模式寄存器 _4	TMDR_4	8	H'FFFE4203	8
	定时器 I/O 控制寄存器 H_4	TIORH_4	8	H'FFFE4206	8
	定时器 I/O 控制寄存器 L_4	TIORL_4	8	H'FFFE4207	8
	定时器中断允许寄存器 _4	TIER_4	8	H'FFFE4209	8
	定时器状态寄存器 _4	TSR_4	8	H'FFFE422D	8
	定时器计数器 _4	TCNT_4	16	H'FFFE4212	16
	定时器通用寄存器 A_4	TGRA_4	16	H'FFFE421C	16
	定时器通用寄存器 B_4	TGRB_4	16	H'FFFE421E	16
	定时器通用寄存器 C_4	TGRC_4	16	H'FFFE4228	16
	定时器通用寄存器 D_4	TGRD_4	16	H'FFFE422A	16
	定时器缓冲器运行传送模式寄存器 _4	TBTM_4	8	H'FFFE4239	8
	定时器 A/D 转换开始请求控制寄存器	TADCR	16	H'FFFE4240	16
	定时器 A/D 转换开始请求周期设定寄存器 A_4	TADCORA_4	16	H'FFFE4242	16
	定时器 A/D 转换开始请求周期设定寄存器 B_4	TADCORB_4	16	H'FFFE4244	16
	定时器 A/D 转换开始请求周期设定缓冲寄存器 A_4	TADCOBRA_4	16	H'FFFE4246	16
	定时器 A/D 转换开始请求周期设定缓冲寄存器 B_4	TADCOBRB_4	16	H'FFFE4248	16
	定时器启动寄存器	TSTR	8	H'FFFE4280	8
	定时器同步寄存器	TSYR	8	H'FFFE4281	8
	定时器读 / 写允许寄存器	TRWER	8	H'FFFE4284	8
	定时器输出主控器允许寄存器	TOER	8	H'FFFE420A	8

模块名	寄存器名	简称	位数	地址	存取长度
MTU2	定时器输出控制寄存器 1	TOCR1	8	H'FFFE420E	8
	定时器输出控制寄存器 2	TOCR2	8	H'FFFE420F	8
	定时器门控制寄存器	TGCR	8	H'FFFE420D	8
	定时器周期数据寄存器	TCDR	16	H'FFFE4214	16
	定时器空载时间数据寄存器	TDDR	16	H'FFFE4216	16
	定时器子计数器	TCNTS	16	H'FFFE4220	16
	定时器周期缓冲寄存器	TCBR	16	H'FFFE4222	16
	定时器跳过中断设定寄存器	TITCR	8	H'FFFE4230	8
	定时器跳过中断次数计数器	TITCNT	8	H'FFFE4231	8
	定时器缓冲器传送设定寄存器	TBTER	8	H'FFFE4232	8
	定时器空载时间允许寄存器	TDER	8	H'FFFE4234	8
	定时器同步清除寄存器	TSYCR	8	H'FFFE4250	8
	定时器波形控制寄存器	TWCR	8	H'FFFE4260	8
	定时器输出电平缓冲器寄存器	TOLBR	8	H'FFFE4236	8
CMT	比较匹配定时器启动寄存器	CMSTR	16	H'FFFEC000	16
	比较匹配定时器控制 / 状态寄存器 _0	CMCSR0	16	H'FFFEC002	16
	比较匹配计数器 _0	CMCNT0	16	H'FFFEC004	16
	比较匹配常数寄存器 _0	CMCOR0	16	H'FFFEC006	16
	比较匹配定时器控制 / 状态寄存器 _1	CMCSR1	16	H'FFFEC008	16
	比较匹配计数器 _1	CMCNT1	16	H'FFFEC00A	16
	比较匹配常数寄存器 _1	CMCOR1	16	H'FFFEC00C	16
WDT	看门狗定时器控制 / 状态寄存器	WTCNT	8	H'FFFE0002	8、16
	看门狗定时器的计数器	WTCSR	8	H'FFFE0000	8、16
	看门狗复位控制 / 状态寄存器	WRCSR	8	H'FFFE0004	8、16
RTC	64Hz 计数器	R64CNT	8	H'FFFF2000	8
	秒计数器	RSECCNT	8	H'FFFF2002	8
	分计数器	RMINCNT	8	H'FFFF2004	8
	小时计数器	RHRCNT	8	H'FFFF2006	8
	星期计数器	RWKCNT	8	H'FFFF2008	8
	日计数器	RDAYCNT	8	H'FFFF200A	8
	月计数器	RMONCNT	8	H'FFFF200C	8

模块名	寄存器名	简称	位名	地址	存取长度
RTC	年计数器	RYRCNT	16	H'FFFF200E	16
	秒报警寄存器	RSECAR	8	H'FFFF2010	8
	分报警寄存器	RMINAR	8	H'FFFF2012	8
	小时报警寄存器	RHRAR	8	H'FFFF2014	8
	星期报警寄存器	RWKAR	8	H'FFFF2016	8
	日报警寄存器	RDAYAR	8	H'FFFF2018	8
	月报警寄存器	RMONAR	8	H'FFFF201A	8
	年报警寄存器	RYRAR	16	H'FFFF2020	16
	RTC 控制寄存器 1	RCR1	8	H'FFFF201C	8
	RTC 控制寄存器 2	RCR2	8	H'FFFF201E	8
	RTC 控制寄存器 3	RCR3	8	H'FFFF2024	8
SCIF	串行模式寄存器 _0	SCSMR_0	16	H'FFFE8000	16
	位速率寄存器 _0	SCBRR_0	8	H'FFFE8004	8
	串行控制寄存器 _0	SCSCR_0	16	H'FFFE8008	16
	发送 FIFO 数据寄存器 _0	SCFTDR_0	8	H'FFFE800C	8
	串行状态寄存器 _0	SCFSR_0	16	H'FFFE8010	16
	接收 FIFO 数据寄存器 _0	SCFRDR_0	8	H'FFFE8014	8
	FIFO 控制寄存器 _0	SCFCR_0	16	H'FFFE8018	16
	FIFO 数据计数置位寄存器 _0	SCFDR_0	16	H'FFFE801C	16
	串行端口寄存器 _0	SCSPTR_0	16	H'FFFE8020	16
	线路状态寄存器 _0	SCLSR_0	16	H'FFFE8024	16
	串行扩展模式寄存器 _0	SCEMR_0	16	H'FFFE8028	16
	串行模式寄存器 _1	SCSMR_1	16	H'FFFE8800	16
	位速率寄存器 _1	SCBRR_1	8	H'FFFE8804	8
	串行控制寄存器 _1	SCSCR_1	16	H'FFFE8808	16
	发送 FIFO 数据寄存器 _1	SCFTDR_1	8	H'FFFE880C	8
	串行状态寄存器 _1	SCFSR_1	16	H'FFFE8810	16
	接收 FIFO 数据寄存器 _1	SCFRDR_1	8	H'FFFE8814	8
	FIFO 控制寄存器 _1	SCFCR_1	16	H'FFFE8818	16
	FIFO 数据计数置位寄存器 _1	SCFDR_1	16	H'FFFE881C	16
	串行端口寄存器 _1	SCSPTR_1	16	H'FFFE8820	16
	线路状态寄存器 _1	SCLSR_1	16	H'FFFE8824	16
	串行扩展模式寄存器 _1	SCEMR_1	16	H'FFFE8828	16
	串行模式寄存器 _2	SCSMR_2	16	H'FFFE9000	16
	位速率寄存器 _2	SCBRR_2	8	H'FFFE9004	8
	串行控制寄存器 _2	SCSCR_2	16	H'FFFE9008	16

模块名	寄存器名	简称	位数	地址	存取长度
SCIF	发送 FIFO 数据寄存器 _2	SCFTDR_2	8	H'FFFE900C	8
	串行状态寄存器 _2	SCFSR_2	16	H'FFFE9010	16
	接收 FIFO 数据寄存器 _2	SCFRDR_2	8	H'FFFE9014	8
	FIFO 控制寄存器 _2	SCFCR_2	16	H'FFFE9018	16
	FIFO 数据计数置位寄存器 _2	SCFDR_2	16	H'FFFE901C	16
	串行端口寄存器 _2	SCSPTR_2	16	H'FFFE9020	16
	线路状态寄存器 _2	SCLSR_2	16	H'FFFE9024	16
	串行扩展模式寄存器 _2	SCEMR_2	16	H'FFFE9028	16
	串行模式寄存器 _3	SCSMR_3	16	H'FFFE9800	16
	位速率寄存器 _3	SCBRR_3	8	H'FFFE9804	8
	串行控制寄存器 _3	SCSCR_3	16	H'FFFE9808	16
	发送 FIFO 数据寄存器 _3	SCFTDR_3	8	H'FFFE980C	8
	串行状态寄存器 _3	SCFSR_3	16	H'FFFE9810	16
	接收 FIFO 数据寄存器 _3	SCFRDR_3	8	H'FFFE9814	8
	FIFO 控制寄存器 _3	SCFCR_3	16	H'FFFE9818	16
	FIFO 数据计数置位寄存器 _3	SCFDR_3	16	H'FFFE981C	16
	串行端口寄存器 _3	SCSPTR_3	16	H'FFFE9820	16
	线路状态寄存器 _3	SCLSR_3	16	H'FFFE9824	16
	串行扩展模式寄存器 _3	SCEMR_3	16	H'FFFE9828	16
SSU	SS 控制寄存器 H_0	SSCRH_0	8	H'FFFE7000	8、16
	SS 控制寄存器 L_0	SSCRL_0	8	H'FFFE7001	8
	SS 模式寄存器 _0	SSMR_0	8	H'FFFE7002	8、16
	SS 允许寄存器 _0	SSER_0	8	H'FFFE7003	8
	SS 状态寄存器 _0	SSSR_0	8	H'FFFE7004	8、16
	SS 控制寄存器 2_0	SSCR2_0	8	H'FFFE7005	8
	SS 发送数据寄存器 0_0	SSTDR0_0	8	H'FFFE7006	8、16
	SS 发送数据寄存器 1_0	SSTDR1_0	8	H'FFFE7007	8
	SS 发送数据寄存器 2_0	SSTDR2_0	8	H'FFFE7008	8、16
	SS 发送数据寄存器 3_0	SSTDR3_0	8	H'FFFE7009	8
	SS 接收数据寄存器 0_0	SSRDR0_0	8	H'FFFE700A	8、16
	SS 接收数据寄存器 1_0	SSRDR1_0	8	H'FFFE700B	8
	SS 接收数据寄存器 2_0	SSRDR2_0	8	H'FFFE700C	8、16
	SS 接收数据寄存器 3_0	SSRDR3_0	8	H'FFFE700D	8
	SS 控制寄存器 H_1	SSCRH_1	8	H'FFFE7800	8、16
	SS 控制寄存器 L_1	SSCRL_1	8	H'FFFE7801	8
	SS 模式寄存器 _1	SSMR_1	8	H'FFFE7802	8、16

模块名	寄存器名	简称	位数	地址	存取长度
SSU	SS 允许寄存器 _1	SSER_1	8	H'FFFE7803	8
	SS 状态寄存器 _1	SSSR_1	8	H'FFFE7804	8、16
	SS 控制寄存器 2_1	SSCR2_1	8	H'FFFE7805	8
	SS 发送数据寄存器 0_1	SSTDR0_1	8	H'FFFE7806	8、16
	SS 发送数据寄存器 1_1	SSTDR1_1	8	H'FFFE7807	8
	SS 发送数据寄存器 2_1	SSTDR2_1	8	H'FFFE7808	8、16
	SS 发送数据寄存器 3_1	SSTDR3_1	8	H'FFFE7809	8
	SS 接收数据寄存器 0_1	SSRDR0_1	8	H'FFFE780A	8、16
	SS 接收数据寄存器 1_1	SSRDR1_1	8	H'FFFE780B	8
	SS 接收数据寄存器 2_1	SSRDR2_1	8	H'FFFE780C	8、16
	SS 接收数据寄存器 3_1	SSRDR3_1	8	H'FFFE780D	8
IIC3	I2C 总线控制寄存器 1_0	ICCR1_0	8	H'FFFEE000	8
	I2C 总线控制寄存器 2_0	ICCR2_0	8	H'FFFEE001	8
	I2C 总线模式寄存器 _0	ICMR_0	8	H'FFFEE002	8
	I2C 总线中断允许寄存器 _0	ICIER_0	8	H'FFFEE003	8
	I2C 总线状态寄存器 _0	ICSR_0	8	H'FFFEE004	8
	从属地址寄存器 _0	SAR_0	8	H'FFFEE005	8
	I2C 总线发送数据寄存器 _0	ICDRT_0	8	H'FFFEE006	8
	I2C 总线接收数据寄存器 _0	ICDRR_0	8	H'FFFEE007	8
	NF2CYC 寄存器 _0	NF2CYC_0	8	H'FFFEE008	8
	I2C 总线控制寄存器 1_1	ICCR1_1	8	H'FFFEE400	8
	I2C 总线控制寄存器 2_1	ICCR2_1	8	H'FFFEE401	8
	I2C 总线模式寄存器 _1	ICMR_1	8	H'FFFEE402	8
	I2C 总线中断允许寄存器 _1	ICIER_1	8	H'FFFEE403	8
	I2C 总线状态寄存器 _1	ICSR_1	8	H'FFFEE404	8
	从属地址寄存器 _1	SAR_1	8	H'FFFEE405	8
	I2C 总线发送数据寄存器 _1	ICDRT_1	8	H'FFFEE406	8
	I2C 总线接收数据寄存器 _1	ICDRR_1	8	H'FFFEE407	8
	NF2CYC 寄存器 _1	NF2CYC_1	8	H'FFFEE408	8
	I2C 总线控制寄存器 1_2	ICCR1_2	8	H'FFFEE800	8
	I2C 总线控制寄存器 2_2	ICCR2_2	8	H'FFFEE801	8
	I2C 总线模式寄存器 _2	ICMR_2	8	H'FFFEE802	8
	I2C 总线中断允许寄存器 _2	ICIER_2	8	H'FFFEE803	8
	I2C 总线状态寄存器 _2	ICSR_2	8	H'FFFEE804	8

模块名	寄存器名	简称	位数	地址	存取长度
IIC3	从属地址寄存器 _2	SAR_2	8	H'FFFEE805	8
	I2C 总线发送数据寄存器 _2	ICDRT_2	8	H'FFFEE806	8
	I2C 总线接收数据寄存器 _2	ICDRR_2	8	H'FFFEE807	8
	NF2CYC 寄存器 _2	NF2CYC_2	8	H'FFFEE808	8
	I2C 总线控制寄存器 1_3	ICCR1_3	8	H'FFFEEC00	8
	I2C 总线控制寄存器 2_3	ICCR2_3	8	H'FFFEEC01	8
	I2C 总线模式寄存器 _3	ICMR_3	8	H'FFFEEC02	8
	I2C 总线中断允许寄存器 _3	ICIER_3	8	H'FFFEEC03	8
	I2C 总线状态寄存器 _3	ICSR_3	8	H'FFFEEC04	8
	从属地址寄存器 _3	SAR_3	8	H'FFFEEC05	8
	I2C 总线发送数据寄存器 _3	ICDRT_3	8	H'FFFEEC06	8
	I2C 总线接收数据寄存器 _3	ICDRR_3	8	H'FFFEEC07	8
	NF2CYC 寄存器 _3	NF2CYC_3	8	H'FFFEEC08	8
SSI	控制寄存器 _0	SSICR_0	32	H'FFFFC000	32
	状态寄存器 _0	SSISR_0	32	H'FFFFC004	32
	发送数据寄存器 _0	SSITDR_0	32	H'FFFFC008	32
	接收数据寄存器 _0	SSIRDR_0	32	H'FFFFC00C	32
	控制寄存器 _1	SSICR_1	32	H'FFFFC800	32
	状态寄存器 _1	SSISR_1	32	H'FFFFC804	32
	发送数据寄存器 _1	SSITDR_1	32	H'FFFFC808	32
	接收数据寄存器 _1	SSIRDR_1	32	H'FFFFC80C	32
	控制寄存器 _2	SSICR_2	32	H'FFFFD000	32
	状态寄存器 _2	SSISR_2	32	H'FFFFD004	32
	发送数据寄存器 _2	SSITDR_2	32	H'FFFFD008	32
	接收数据寄存器 _2	SSIRDR_2	32	H'FFFFD00C	32
	控制寄存器 _3	SSICR_3	32	H'FFFFD800	32
	状态寄存器 _3	SSISR_3	32	H'FFFFD804	32
	发送数据寄存器 _3	SSITDR_3	32	H'FFFFD808	32
	接收数据寄存器 _3	SSIRDR_3	32	H'FFFFD80C	32
RCAN-TL1	主控制寄存器 _0	MCR_0	16	H'FFFF0000	16
	通用状态寄存器 _0	GSR_0	16	H'FFFF0002	16
	位结构寄存器 1_0	BCR1_0	16	H'FFFF0004	16
	位结构寄存器 0_0	BCR0_0	16	H'FFFF0006	16
	中断请求寄存器 _0	IRR_0	16	H'FFFF0008	16

模块名	寄存器名	简称	位数	地址	存取长度
RCAN-TL1	中断屏蔽寄存器 _0	IMR_0	16	H'FFFF000A	16
	发送错误计数器 / 接收错误计数器 _0	TEC_REC_0	16	H'FFFF000C	8、16
	发送等待寄存器 1_0	TXPR1_0	16	H'FFFF0020	32
	发送等待寄存器 0_0	TXPR0_0	16	H'FFFF0022	16
	发送取消寄存器 1_0	TXCR1_0	16	H'FFFF0028	16
	发送取消寄存器 0_0	TXCR0_0	16	H'FFFF002A	16
	发送应答寄存器 1_0	TXACK1_0	16	H'FFFF0030	16
	发送应答寄存器 0_0	TXACK0_0	16	H'FFFF0032	16
	中断应答寄存器 1_0	ABACK1_0	16	H'FFFF0038	16
	中断应答寄存器 0_0	ABACK0_0	16	H'FFFF003A	16
	数据帧接收完成寄存器 1_0	RXPR1_0	16	H'FFFF0040	16
	数据帧接收完成寄存器 0_0	RXPR0_0	16	H'FFFF0042	16
	远程帧接收完成寄存器 1_0	RFPR1_0	16	H'FFFF0048	16
	远程帧接收完成寄存器 0_0	RFPR0_0	16	H'FFFF004A	16
	邮箱中断屏蔽寄存器 1_0	MBIMR1_0	16	H'FFFF0050	16
	邮箱中断屏蔽寄存器 0_0	MBIMR0_0	16	H'FFFF0052	16
	未读信息状态寄存器 1_0	UMSR1_0	16	H'FFFF0058	16
	未读信息状态寄存器 0_0	UMSR0_0	16	H'FFFF005A	16
	时间触发控制寄存器 _0	TTCR0_0	16	H'FFFF0080	16
	最大周期 /Tx_Enable_Window 寄存器 _0	CMAX_TEW_0	16	H'FFFF0084	16
	基准触发偏移寄存器 _0	RFTROFF_0	16	H'FFFF0086	16
	定时器状态寄存器 _0	TSR_0	16	H'FFFF0088	16
	周期计数器寄存器 _0	CCR_0	16	H'FFFF008A	16
	定时器计数器寄存器 _0	TCNTR_0	16	H'FFFF008C	16
	周期时间寄存器 _0	CYCTR_0	16	H'FFFF0090	16
	基准标记寄存器 _0	RFMK_0	16	H'FFFF0094	16
	定时器比较匹配寄存器 0_0	TCMR0_0	16	H'FFFF0098	16
	定时器比较匹配寄存器 1_0	TCMR1_0	16	H'FFFF009C	16
	定时器比较匹配寄存器 2_0	TCMR2_0	16	H'FFFF00A0	16
	发送触发时间选择寄存器 _0	TTTSEL_0	16	H'FFFF00A4	16
	邮箱 n 控制 0H_0 （n=0 ～ 31）	MBn_ CONTROL 0_H_0 （n=0 ～ 31）	16	H'FFFF0100 +n×32	16、32

模块名	寄存器名	简称	位数	地址	存取长度
RCAN-TL1	邮箱 n 控制 0L_0（n=0 ～ 31）	MBn_CONTROL0_L_0（n=0 ～ 31）	16	H'FFFF0102+n×32	16
	邮箱 n 本地接收过滤器屏蔽 0_0 （n=0 ～ 31）	MBn_LAFM0_0（n=0 ～ 31）	16	H'FFFF0104+n×32	16、 32
	邮箱 n 本地接收过滤器屏蔽 1_0 （n=0 ～ 31）	MBn_LAFM1_0（n=0 ～ 31）	16	H'FFFF0106+n×32	16
	邮箱 n 数据 01_0（n=0 ～ 31）	MBn_DATA_01_0（n=0 ～ 31）	16	H'FFFF0108+n×32	8、 16、 32
	邮箱 n 数据 23_0（n=0 ～ 31）	MBn_DATA_23_0（n=0 ～ 31）	16	H'FFFF010A+n×32	8、 16
	邮箱 n 数据 45_0（n=0 ～ 31）	MBn_DATA_45_0（n=0 ～ 31）	16	H'FFFF010C+n×32	8、 16、 32
	邮箱 n 数据 67_0（n=0 ～ 31）	MBn_DATA_67_0（n=0 ～ 31）	16	H'FFFF010E+n×32	8、 16
	邮箱 n 控制 1_0（n=0 ～ 31）	MBn_CONTROL1_0（n=0 ～ 31）	16	H'FFFF0110+n×32	8、 16
	邮箱 n 时戳 _0（n=0 ～ 15、 30、 31）	MBn_TIMESTAMP_0（n=0 ～ 15、 30、 31）	16	H'FFFF0112+n×32	16
	邮箱 n 触发时间 _0（n=24 ～ 30）	MBn_TTT_0（n=24 ～ 30）	16	H'FFFF0114+n×32	16
	邮箱 nTT 控制 _0（n=24 ～ 29）	MBn_TTCONTROL_0（n=24 ～ 29）	16	H'FFFF0116+n×32	16
	主控制寄存器 _1	MCR_1	16	H'FFFF0800	16
	通用状态寄存器 _1	GSR_1	16	H'FFFF0802	16
	位结构寄存器 1_1	BCR1_1	16	H'FFFF0804	16
	位结构寄存器 0_1	BCR0_1	16	H'FFFF0806	16
	中断请求寄存器 _1	IRR_1	16	H'FFFF0808	16
	中断屏蔽寄存器 _1	IMR_1	16	H'FFFF080A	16
	发送错误计数器 / 接收错误计数器 _1	TEC_REC_1	16	H'FFFF080C	8、 16
	发送等待寄存器 1_1	TXPR1_1	16	H'FFFF0820	32
	发送等待寄存器 0_1	TXPR0_1	16	H'FFFF0822	16
	发送取消寄存器 1_1	TXCR1_1	16	H'FFFF0828	16
	发送取消寄存器 0_1	TXCR0_1	16	H'FFFF082A	16
	发送应答寄存器 1_1	TXACK1_1	16	H'FFFF0830	16
	发送应答寄存器 0_1	TXACK0_1	16	H'FFFF0832	16

模块名	寄存器名	简称	位数	地址	存取长度
RCAN-TL1	中断应答寄存器 1_1	ABACK1_1	16	H'FFFF0838	16
	中断应答寄存器 0_1	ABACK0_1	16	H'FFFF083A	16
	数据帧接收完成寄存器 1_1	RXPR1_1	16	H'FFFF0840	16
	数据帧接收完成寄存器 0_1	RXPR0_1	16	H'FFFF0842	16
	远程帧接收完成寄存器 1_1	RFPR1_1	16	H'FFFF0848	16
	远程帧接收完成寄存器 0_1	RFPR0_1	16	H'FFFF084A	16
	邮箱中断屏蔽寄存器 1_1	MBIMR1_1	16	H'FFFF0850	16
	邮箱中断屏蔽寄存器 0_1	MBIMR0_1	16	H'FFFF0852	16
	未读信息状态寄存器 1_1	UMSR1_1	16	H'FFFF0858	16
	未读信息状态寄存器 0_1	UMSR0_1	16	H'FFFF085A	16
	时间触发控制寄存器 _1	TTCR0_1	16	H'FFFF0880	16
	最大周期 /Tx_Enable_Window 寄存器 _1	CMAX_TEW_1	16	H'FFFF0884	16
	基准触发偏移寄存器 _1	RFTROFF_1	16	H'FFFF0886	16
	定时器状态寄存器 _1	TSR_1	16	H'FFFF0888	16
	周期计数器寄存器 _1	CCR_1	16	H'FFFF088A	16
	定时器计数器寄存器 _1	TCNTR_1	16	H'FFFF088C	16
	周期时间寄存器 _1	CYCTR_1	16	H'FFFF0890	16
	基准标记寄存器 _1	RFMK_1	16	H'FFFF0894	16
	时间比较匹配寄存器 0_1	TCMR0_1	16	H'FFFF0898	16
	时间比较匹配寄存器 1_1	TCMR1_1	16	H'FFFF089C	16
	时间比较匹配寄存器 2_1	TCMR2_1	16	H'FFFF08A0	16
	发送触发时间选择寄存器 _1	TTTSEL_1	16	H'FFFF08A4	16
	邮箱 n 控制 0H_1 （n=0 ～ 31）	MBn_CONTROL0_H_1 （n=0 ～ 31）	16	H'FFFF0900+n×32	16、 32
	邮箱 n 控制 0L_1 （n=0 ～ 31）	MBn_CONTROL0_L_1 （n=0 ～ 31）	16	H'FFFF0902+n×32	16
	邮箱 n 本地接收过滤器屏蔽 0_1 （n=0 ～ 31）	MBn_LAFM0_1 （n=0 ～ 31）	16	H'FFFF0904+n×32	16、 32
	邮箱 n 本地接收过滤器屏蔽 1_1 （n=0 ～ 31）	MBn_LAFM1_1 （n=0 ～ 31）	16	H'FFFF0906+n×32	16
	邮箱 n 数据 01_1 （n=0 ～ 31）	MBn_DATA_01_1 （n=0 ～ 31）	16	H'FFFF0908+n×32	8、 16、 32
	邮箱 n 数据 23_1 （n=0 ～ 31）	MBn_DATA_23_1 （n=0 ～ 31）	16	H'FFFF090A+n×32	8、 16
	邮箱 n 数据 45_1 （n=0 ～ 31）	MBn_DATA_45_1 （n=0 ～ 31）	16	H'FFFF090C+n×32	8、 16、 32

模块名	寄存器名	简称	位数	地址	存取长度
RCAN-TL1	邮箱 n 数据 67_1 （n=0 ～ 31）	MBn_DATA_67_1 （n=0 ～ 31）	16	H'FFFF090E+n×32	8、 16
	邮箱 n 控制 1_1 （n=0 ～ 31）	MBn_CONTROL1_1 （n=0 ～ 31）	16	H'FFFF0910+n×32	8、 16
	邮箱 n 时戳 _1 （n=0 ～ 15、 30、 31）	MBn_TIMESTAMP_1 （n=0 ～ 15、 30、 31）	16	H'FFFF0912+n×32	16
	邮箱 n 触发时间 _1 （n=24 ～ 30）	MBn_TTT_1 （n=24 ～ 30）	16	H'FFFF0914+n×32	16
	邮箱 nTT 控制 _1 （n=24 ～ 29）	MBn_TTCONTROL_1 （n=24 ～ 29）	16	H'FFFF0916+n×32	16
IEB	IEBus 控制寄存器	IECTR	8	H'FFFEF000	8
	IEBus 指令寄存器	IECMR	8	H'FFFEF001	8
	IEBus 主控制寄存器	IEMCR	8	H'FFFEF002	8
	IEBus 主单元地址寄存器 1	IEAR1	8	H'FFFEF003	8
	IEBus 主单元地址寄存器 2	IEAR2	8	H'FFFEF004	8
	IEBus 从属地址设定寄存器 1	IESA1	8	H'FFFEF005	8
	IEBus 从属地址设定寄存器 2	IESA2	8	H'FFFEF006	8
	IEBus 发送信息长度寄存器	IETBFL	8	H'FFFEF007	8
	IEBus 接收基本地址寄存器 1	IEMA1	8	H'FFFEF009	8
	IEBus 接收基本地址寄存器 2	IEMA2	8	H'FFFEF00A	8
	IEBus 接收控制字段寄存器	IERCTL	8	H'FFFEF00B	8
	IEBus 接收信息长度寄存器	IERBFL	8	H'FFFEF00C	8
	IEBus 锁定地址寄存器 1	IELA1	8	H'FFFEF00E	8
	IEBus 锁定地址寄存器 2	IELA2	8	H'FFFEF00F	8
	IEBus 通用标志寄存器	IEFLG	8	H'FFFEF010	8
	IEBus 发送状态寄存器	IETSR	8	H'FFFEF011	8
	IEBus 发送中断允许寄存器	IEIET	8	H'FFFEF012	8
	IEBus 接收状态寄存器	IERSR	8	H'FFFEF014	8
	IEBus 接收中断允许寄存器	IEIER	8	H'FFFEF015	8
	IEBus 时钟选择寄存器	IECKSR	8	H'FFFEF018	8
	IEBus 发送数据缓冲器 001 ～ 128	IETB001 ～ IETB128	8	H'FFFEF100 ～ H'FFFEF17F	8
	IEBus 接收数据缓冲器 001 ～ 128	IERB001 ～ IERB128	8	H'FFFEF200 ～ H'FFFEF27F	8

模块名	寄存器名	简称	位数	地址	存取长度
ROM-DEC	ROMDEC 允许控制寄存器	CROMEN	8	H'FFFC2000	8
	同步码的同步控制寄存器	CROMSY0	8	H'FFFC2001	8
	解码模式控制寄存器	CROMCTL0	8	H'FFFC2002	8
	EDC、 ECC 检查控制寄存器	CROMCTL1	8	H'FFFC2003	8
	解码处理自动停止控制寄存器	CROMCTL3	8	H'FFFC2005	8
	解码选项设定控制寄存器	CROMCTL4	8	H'FFFC2006	8
	HEAD20 ～ 22 显示控制寄存器	CROMCTL5	8	H'FFFC2007	8
	同步码状态寄存器	CROMST0	8	H'FFFC2008	8
	ECC 后的扇区头对应的错误状态寄存器	CROMST1	8	H'FFFC2009	8
	ECC 后子头对应的错误状态寄存器	CROMST3	8	H'FFFC200B	8
	扇区头、子头数据妥当性判定状态寄存器	CROMST4	8	H'FFFC200C	8
	模式判定结果和连接扇区检测状态寄存器	CROMST5	8	H'FFFC200D	8
	ECC、 EDC 错误状态寄存器	CROMST6	8	H'FFFC200E	8
	缓冲器状态寄存器	CBUFST0	8	H'FFFC2014	8
	解码中止源状态寄存器	CBUFST1	8	H'FFFC2015	8
	缓冲溢出状态寄存器	CBUFST2	8	H'FFFC2016	8
	ECC 校正前扇区头 -MINUTES 数据寄存器	HEAD00	8	H'FFFC2018	8
	ECC 校正前扇区头 -SECONDS 数据寄存器	HEAD01	8	H'FFFC2019	8
	ECC 校正前扇区头 -FRAMES（1/75 秒）数据寄存器	HEAD02	8	H'FFFC201A	8
	ECC 校正前扇区头 -MODE 数据寄存器	HEAD03	8	H'FFFC201B	8
	ECC 校正前子头 - 文件号码（BYTE-16）数据寄存器	SHEAD00	8	H'FFFC201C	8
	ECC 校正前子头 - 通道号码（BYTE-17）数据寄存器	SHEAD01	8	H'FFFC201D	8
	ECC 校正前子头 - 子模式（BYTE-18）数据寄存器	SHEAD02	8	H'FFFC201E	8
	ECC 校正前子头 - 数据类型（BYTE-19）数据寄存器	SHEAD03	8	H'FFFC201F	8
	ECC 校正前子头 - 文件号码（BYTE-20）数据寄存器	SHEAD04	8	H'FFFC2020	8
	ECC 校正前子头 - 通道号码（BYTE-21）数据寄存器	SHEAD05	8	H'FFFC2021	8
	ECC 校正前子头 - 子模式（BYTE-22）数据寄存器	SHEAD06	8	H'FFFC2022	8

模块名	寄存器名	简称	位数	地址	存取长度
ROM-DEC	ECC 校正前子头 - 数据类型（BYTE-23）数据寄存器	SHEAD07	8	H'FFFC2023	8
	ECC 校正后扇区头 -MINUTES 数据寄存器	HEAD20	8	H'FFFC2024	8
	ECC 校正后扇区头 -SECONDS 数据寄存器	HEAD21	8	H'FFFC2025	8
	ECC 校正后扇区头 -FRAMES（1/75 秒）数据寄存器	HEAD22	8	H'FFFC2026	8
	ECC 校正后扇区头 -MODE 数据寄存器	HEAD23	8	H'FFFC2027	8
	ECC 校正后子头 - 文件号码（BYTE-16）数据寄存器	SHEAD20	8	H'FFFC2028	8
	ECC 校正后子头 - 通道号码（BYTE-17）数据寄存器	SHEAD21	8	H'FFFC2029	8
	ECC 校正后子头 - 子模式（BYTE-18）数据寄存器	SHEAD22	8	H'FFFC202A	8
	ECCC 校正后子头 - 数据类型（BYTE-19）数据寄存器	SHEAD23	8	H'FFFC202B	8
	ECC 校正后子头 - 文件号码（BYTE-20）数据寄存器	SHEAD24	8	H'FFFC202C	8
	ECC 校正后子头 - 通道号码（BYTE-21）数据寄存器	SHEAD25	8	H'FFFC202D	8
	ECC 校正后子头 - 子模式（BYTE-22）数据寄存器	SHEAD26	8	H'FFFC202E	8
	ECC 校正后子头 - 数据类型（BYTE-23）数据寄存器	SHEAD27	8	H'FFFC202F	8
	自动缓冲设定控制寄存器	CBUFCTL0	8	H'FFFC2040	8
	自动缓冲开始扇区设定 -MINUTES 控制寄存器	CBUFCTL1	8	H'FFFC2041	8
	自动缓冲开始扇区设定 -SECONDS 控制寄存器	CBUFCTL2	8	H'FFFC2042	8
	自动缓冲开始扇区设定 -FRAMES 控制寄存器	CBUFCTL3	8	H'FFFC2043	8
	ISY 中断源屏蔽控制寄存器	CROMST0M	8	H'FFFC2045	8
	CD-ROM 解码器模块复位控制寄存器	ROMDECRST	8	H'FFFC2100	8
	CD-ROM 解码器模块复位状态寄存器	RSTSTAT	8	H'FFFC2101	8
	SSI 数据控制寄存器	SSI	8	H'FFFC2102	8
	中断标志寄存器	INTHOLD	8	H'FFFC2108	8
	中断源屏蔽控制寄存器	INHINT	8	H'FFFC2109	8
	CD-ROM 解码器数据流输入寄存器	STRMDIN0	16	H'FFFC2200	16、32
	CD-ROM 解码器数据流输入寄存器	STRMDIN2	16	H'FFFC2202	16
	CD-ROM 解码器数据流输出寄存器	STRMDOUT0	16	H'FFFC2204	16、32

模块名	寄存器名	简称	位数	地址	存取长度
ADC	A/D 数据寄存器 A	ADDRA	16	H'FFFE5800	16
	A/D 数据寄存器 B	ADDRB	16	H'FFFE5802	16
	A/D 数据寄存器 C	ADDRC	16	H'FFFE5804	16
	A/D 数据寄存器 D	ADDRD	16	H'FFFE5806	16
	A/D 数据寄存器 E	ADDRE	16	H'FFFE5808	16
	A/D 数据寄存器 F	ADDRF	16	H'FFFE580A	16
	A/D 数据寄存器 G	ADDRG	16	H'FFFE580C	16
	A/D 数据寄存器 H	ADDRH	16	H'FFFE580E	16
	A/D 控制 / 状态寄存器	ADCSR	16	H'FFFE5820	16
DAC	D/A 数据寄存器 0	DADR0	8	H'FFFE6800	8、16
	D/A 数据寄存器 1	DADR1	8	H'FFFE6801	8、16
	D/A 控制寄存器	DACR	8	H'FFFE6802	8、16
FLCTL	通用控制寄存器	FLCMNCR	32	H'FFFFF000	32
	命令控制寄存器	FLCMDCR	32	H'FFFFF004	32
	命令码寄存器	FLCMCDR	32	H'FFFFF008	32
	地址寄存器	FLADR	32	H'FFFFF00C	32
	地址寄存器 2	FLADR2	32	H'FFFFF03C	32
	数据寄存器	FLDATAR	32	H'FFFFF010	32
	数据计数器寄存器	FLDTCNTR	32	H'FFFFF014	32
	中断 DMA 控制寄存器	FLINTDMACR	32	H'FFFFF018	32
	就绪忙超时设定寄存器	FLBSYTMR	32	H'FFFFF01C	32
	就绪忙超时计数器	FLBSYCNT	32	H'FFFFF020	32
	数据 FIFO 寄存器	FLDTFIFO	32	H'FFFFF050	32
	管理码 FIFO 寄存器	FLECFIFO	32	H'FFFFF060	32
	传送控制寄存器	FLTRCR	8	H'FFFFF02C	8
USB	系统结构控制寄存器	SYSCFG	16	H'FFFC1C00	16
	系统结构状态寄存器	SYSSTS	16	H'FFFC1C02	16
	器件状态控制寄存器	DVSTCTR	16	H'FFFC1C04	16
	测试模式寄存器	TESTMODE	16	H'FFFC1C06	16
	CPU-FIFO 总线结构寄存器	CFBCFG	16	H'FFFC1C0A	16
	DMA0-FIFO 总线结构寄存器	D0FBCFG	16	H'FFFC1C0C	16
	DMA1-FIFO 总线结构寄存器	D1FBCFG	16	H'FFFC1C0E	16
	CFIFO 端口寄存器	CFIFO	32	H'FFFC1C10	8、16、32
	D0FIFO 端口寄存器	D0FIFO	32	H'FFFC1C14	8、16、32
	D1FIFO 端口寄存器	D1FIFO	32	H'FFFC1C18	8、16、32
	CFIFO 端口选择寄存器	CFIFOSEL	16	H'FFFC1C1E	16

模块名	寄存器名	简称	位数	地址	存取长度
USB	CFIFO 端口控制寄存器	CFIFOCTR	16	H'FFFC1C20	16
	CFIFO 端口 SIE 寄存器	CFIFOSIE	16	H'FFFC1C22	16
	D0FIFO 端口选择寄存器	D0FIFOSEL	16	H'FFFC1C24	16
	D0FIFO 端口控制寄存器	D0FIFOCTR	16	H'FFFC1C26	16
	D0 处理计数器寄存器	D0FIFOTRN	16	H'FFFC1C28	16
	D1FIFO 端口选择寄存器	D1FIFOSEL	16	H'FFFC1C2A	16
	D1FIFO 端口控制寄存器	D1FIFOCTR	16	H'FFFC1C2C	16
	D1 处理计数器寄存器	D1FIFOTRN	16	H'FFFC1C2E	16
	中断允许寄存器 0	INTENB0	16	H'FFFC1C30	16
	中断允许寄存器 1	INTENB1	16	H'FFFC1C32	16
	BRDY 中断允许寄存器	BRDYENB	16	H'FFFC1C36	16
	NRDY 中断允许寄存器	NRDYENB	16	H'FFFC1C38	16
	BEMP 中断允许寄存器	BEMPENB	16	H'FFFC1C3A	16
	中断状态寄存器 0	INTSTS0	16	H'FFFC1C40	16
	中断状态寄存器 1	INTSTS1	16	H'FFFC1C42	16
	BRDY 中断状态寄存器	BRDYSTS	16	H'FFFC1C46	16
	NRDY 中断状态寄存器	NRDYSTS	16	H'FFFC1C48	16
	BEMP 中断状态寄存器	BEMPSTS	16	H'FFFC1C4A	16
	帧号寄存器	FRMNUM	16	H'FFFC1C4C	16
	μ 帧号寄存器	UFRMNUM	16	H'FFFC1C4E	16
	USB 地址寄存器	USBADDR	16	H'FFFC1C50	16
	USB 请求类型寄存器	USBREQ	16	H'FFFC1C54	16
	USB 请求值寄存器	USBVAL	16	H'FFFC1C56	16
	USB 请求变址寄存器	USBINDX	16	H'FFFC1C58	16
	USB 请求长度寄存器	USBLENG	16	H'FFFC1C5A	16
	DCP 结构寄存器	DCPCFG	16	H'FFFC1C5C	16
	DCP Max 包大小寄存器	DCPMAXP	16	H'FFFC1C5E	16
	DCP 控制寄存器	DCPCTR	16	H'FFFC1C60	16
	管道窗口选择寄存器	PIPESEL	16	H'FFFC1C64	16
	管道结构寄存器	PIPECFG	16	H'FFFC1C66	16
	管道缓冲器指定寄存器	PIPEBUF	16	H'FFFC1C68	16
	管道 Max 包大小寄存器	PIPEMAXP	16	H'FFFC1C6A	16
	管道周期控制寄存器	PIPEPERI	16	H'FFFC1C6C	16
	管道 1 控制寄存器	PIPE1CTR	16	H'FFFC1C70	16
	管道 2 控制寄存器	PIPE2CTR	16	H'FFFC1C72	16

模块名	寄存器名	简称	位数	地址	存取长度
USB	管道 3 控制寄存器	PIPE3CTR	16	H'FFFC1C74	16
	管道 4 控制寄存器	PIPE4CTR	16	H'FFFC1C76	16
	管道 5 控制寄存器	PIPE5CTR	16	H'FFFC1C78	16
	管道 6 控制寄存器	PIPE6CTR	16	H'FFFC1C7A	16
	管道 7 控制寄存器	PIPE7CTR	16	H'FFFC1C7C	16
	USB AC 特性转换寄存器	USBACSWR	32	H'FFFC1C84	32
LCDC	LCDC 输入时钟寄存器	LDICKR	16	H'FFFFFC00	16
	LCDC 模块类型寄存器	LDMRT	16	H'FFFFFC02	16
	LCDC 数据格式寄存器	LDDFR	16	H'FFFFFC04	16
	LCDC 扫描模式寄存器	LDSMR	16	H'FFFFFC06	16
	LCDC 上部显示面板用取数据开始地址寄存器	LDSARU	32	H'FFFFFC08	32
	LCDC 下部显示面板用取数据开始地址寄存器	LDSARL	32	H'FFFFFC0C	32
	LCDC 显示面板用取数据行地址偏移寄存器	LDLAOR	16	H'FFFFFC10	16
	LCDC 调色板控制寄存器	LDPALCR	16	H'FFFFFC12	16
	调色板数据寄存器 00 ～ FF	LDPR00 ～ FF	32	H'FFFFF800 ～ H'FFFFFBFC	32
	LCDC 水平字符数寄存器	LDHCNR	16	H'FFFFFC14	16
	LCDC 水平同步信号寄存器	LDHSYNR	16	H'FFFFFC16	16
	LCDC 垂直显示行数寄存器	LDVDLNR	16	H'FFFFFC18	16
	LCDC 垂直总行数寄存器	LDVTLNR	16	H'FFFFFC1A	16
	LCDC 垂直同步信号寄存器	LDVSYNR	16	H'FFFFFC1C	16
	LCDC AC 转变信号交替行数寄存器	LDACLNR	16	H'FFFFFC1E	16
	LCDC 中断控制寄存器	LDINTR	16	H'FFFFFC20	16
	LCDC 电源管理模式寄存器	LDPMMR	16	H'FFFFFC24	16
	LCDC 通电顺序期间寄存器	LDPSPR	16	H'FFFFFC26	16
	LCDC 控制寄存器	LDCNTR	16	H'FFFFFC28	16
	LCDC 用户指定中断控制寄存器	LDUINTR	16	H'FFFFFC34	16
	LCDC 用户指定中断行数寄存器	LDUINTLNR	16	H'FFFFFC36	16
	LCDC 存储器存取间隔数寄存器	LDLIRNR	16	H'FFFFFC40	16
SRC	SRC 输入数据寄存器	SRCID	32	H'FFFF4000	16、32
	SRC 输出数据寄存器	SRCOD	32	H'FFFF4004	16、32
	SRC 输入数据控制寄存器	SRCIDCTRL	16	H'FFFF4008	16
	SRC 输出数据控制寄存器	SRCODCTRL	16	H'FFFF400A	16
	SRC 控制寄存器	SRCCTRL	16	H'FFFF400C	16
	SRC 状态寄存器	SRCSTAT	16	H'FFFF400E	16

模块名	寄存器名	简称	位数	地址	存取长度
PFC	端口 B/IO 寄存器 L	PBIORL	16	H'FFFE3886	8、16
	端口 B 控制寄存器 L4	PBCRL4	16	H'FFFE3890	16、32
	端口 B 控制寄存器 L3	PBCRL3	16	H'FFFE3892	8、16
	端口 B 控制寄存器 L2	PBCRL2	16	H'FFFE3894	8、16、32
	端口 B 控制寄存器 L1	PBCRL1	16	H'FFFE3896	8、16
	IRQOUT 功能控制寄存器	IFCR	16	H'FFFE38A2	8、16
	端口 C/IO 寄存器 L	PCIORL	16	H'FFFE3906	8、16
	端口 C 控制寄存器 L4	PCCRL4	16	H'FFFE3910	8、16、32
	端口 C 控制寄存器 L3	PCCRL3	16	H'FFFE3912	8、16
	端口 C 控制寄存器 L2	PCCRL2	16	H'FFFE3914	8、16、32
	端口 C 控制寄存器 L1	PCCRL1	16	H'FFFE3916	8、16
	端口 D/IO 寄存器 L	PDIORL	16	H'FFFE3986	8、16
	端口 D 控制寄存器 L4	PDCRL4	16	H'FFFE3990	8、16、32
	端口 D 控制寄存器 L3	PDCRL3	16	H'FFFE3992	8、16
	端口 D 控制寄存器 L2	PDCRL2	16	H'FFFE3994	8、16、32
	端口 D 控制寄存器 L1	PDCRL1	16	H'FFFE3996	8、16
	端口 E/IO 寄存器 L	PEIORL	16	H'FFFE3A06	8、16
	端口 E 控制寄存器 L4	PECRL4	16	H'FFFE3A10	8、16、32
	端口 E 控制寄存器 L3	PECRL3	16	H'FFFE3A12	8、16
	端口 E 控制寄存器 L2	PECRL2	16	H'FFFE3A14	8、16、32
	端口 E 控制寄存器 L1	PECRL1	16	H'FFFE3A16	8、16
	端口 F/IO 寄存器 H	PFIORH	16	H'FFFE3A84	8、16、32
	端口 F/IO 寄存器 L	PFIORL	16	H'FFFE3A86	8、16
	端口 F 控制寄存器 H4	PFCRH4	16	H'FFFE3A88	8、16、32
	端口 F 控制寄存器 H3	PFCRH3	16	H'FFFE3A8A	8、16
	端口 F 控制寄存器 H2	PFCRH2	16	H'FFFE3A8C	8、16、32
	端口 F 控制寄存器 H1	PFCRH1	16	H'FFFE3A8E	8、16
	端口 F 控制寄存器 L4	PFCRL4	16	H'FFFE3A90	8、16、32
	端口 F 控制寄存器 L3	PFCRL3	16	H'FFFE3A92	8、16
	端口 F 控制寄存器 L2	PFCRL2	16	H'FFFE3A94	8、16、32
	端口 F 控制寄存器 L1	PFCRL1	16	H'FFFE3A96	8、16
	SSI 时钟选择寄存器	SCSR	16	H'FFFE3AA2	8、16

模块名	寄存器名	简称	位数	地址	存取长度
I/O 端口	端口 A 数据寄存器 L	PADRL	16	H'FFFE3802	8、16
	端口 B 数据寄存器 L	PBDRL	16	H'FFFE3882	8、16
	端口 B 端口寄存器 L	PBPRL	16	H'FFFE389E	8、16
	端口 C 数据寄存器 L	PCDRL	16	H'FFFE3902	8、16
	端口 C 端口寄存器 L	PCPRL	16	H'FFFE391E	8、16
	端口 D 数据寄存器 L	PDDRL	16	H'FFFE3982	8、16
	端口 D 端口寄存器 L	PDPRL	16	H'FFFE399E	8、16
	端口 E 数据寄存器 L	PEDRL	16	H'FFFE3A02	8、16
	端口 E 端口寄存器 L	PEPRL	16	H'FFFE3A1E	8、16
	端口 F 数据寄存器 H	PFDRH	16	H'FFFE3A80	8、16、32
	端口 F 数据寄存器 L	PFDRL	16	H'FFFE3A82	8、16
	端口 F 端口寄存器 H	PFPRH	16	H'FFFE3A9C	8、16、32
	端口 F 端口寄存器 L	PFPRL	16	H'FFFE3A9E	8、16
低功耗模式	待机控制寄存器	STBCR	8	H'FFFE0014	8
	待机控制寄存器 2	STBCR2	8	H'FFFE0018	8
	待机控制寄存器 3	STBCR3	8	H'FFFE0408	8
	待机控制寄存器 4	STBCR4	8	H'FFFE040C	8
	待机控制寄存器 5	STBCR5	8	H'FFFE0410	8
	待机控制寄存器 6	STBCR6	8	H'FFFE0414	8
	系统控制寄存器 1	SYSCR1	8	H'FFFE0402	8
	系统控制寄存器 2	SYSCR2	8	H'FFFE0404	8
	系统控制寄存器 3	SYSCR3	8	H'FFFE0418	8
	深度待机控制寄存器	DSCTR	8	H'FFFF2800	8
	深度待机控制寄存器 2	DSCTR2	8	H'FFFF2802	8
	深度待机解除源选择寄存器	DSSSR	16	H'FFFF2804	16
	深度待机解除源标志寄存器	DSFR	16	H'FFFF2808	16
	保持用内置 RAM 调整寄存器	DSRTR	8	H'FFFF280C	8
H-UDI	指令寄存器	SDIR	16	H'FFFE2000	16

34.2　寄存器位一览

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
CPG	FRQCR	—	CKOEN2	CKOEN[1]	CKOEN[0]	—	—	STC[1]	STC[0]
		—	—	—	IFC	—	PFC[2]	PFC[1]	PFC[0]
INTC	ICR0	NMIL	—	—	—	—	—	—	NMIE
		—	—	—	—	—	—	—	—
	ICR1	IRQ71S	IRQ70S	IRQ61S	IRQ60S	IRQ51S	IRQ50S	IRQ41S	IRQ40S
		IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S
	ICR2	—	—	—	—	—	—	—	—
		PINT7S	PINT6S	PINT5S	PINT4S	PINT3S	PINT2S	PINT1S	PINT0S
	IRQRR	—	—	—	—	—	—	—	—
		IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
	PINTER	—	—	—	—	—	—	—	—
		PINT7E	PINT6E	PINT5E	PINT4E	PINT3E	PINT2E	PINT1E	PINT0E
	PIRR	—	—	—	—	—	—	—	—
		PINT7R	PINT6R	PINT5R	PINT4R	PINT3R	PINT2R	PINT1R	PINT0R
	IBCR	E15	E14	E13	E12	E11	E10	E9	E8
		E7	E6	E5	E4	E3	E2	E1	—
	IBNR	BE[1]	BE[0]	BOVE	—	—	—	—	—
		—	—	—	—	BN[3]	BN[2]	BN[1]	BN[0]
	IPR01								
	IPR02								
	IPR05								
	IPR06								
	IPR07								
	IPR08								
	IPR09								
	IPR10								

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
INTC	IPR11								
	IPR12								
	IPR13								
	IPR14								
	IPR15								
	IPR16								
	IPR17								
UBC	BAR_0	BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24
		BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
		BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8
		BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0
	BAMR_0	BAM31	BAM30	BAM29	BAM28	BAM27	BAM26	BAM25	BAM24
		BAM23	BAM22	BAM21	BAM20	BAM19	BAM18	BAM17	BAM16
		BAM15	BAM14	BAM13	BAM12	BAM11	BAM10	BAM9	BAM8
		BAM7	BAM6	BAM5	BAM4	BAM3	BAM2	BAM1	BAM0
	BBR_0	—	—	UBID	DBE	—	—	CP[1]	CP[0]
		CD[1]	CD[0]	ID[1]	ID[0]	RW[1]	RW[0]	SZ[1]	SZ[0]
	BDR_0	BD31	BD30	BD29	BD28	BD27	BD26	BD25	BD24
		BD23	BD22	BD21	BD20	BD19	BD18	BD17	BD16
		BD15	BD14	BD13	BD12	BD11	BD10	BD9	BD8
		BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0
	BDMR_0	BDM31	BDM30	BDM29	BDM28	BDM27	BDM26	BDM25	BDM24
		BDM23	BDM22	BDM21	BDM20	BDM19	BDM18	BDM17	BDM16
		BDM15	BDM14	BDM13	BDM12	BDM11	BDM10	BDM9	BDM8
		BDM7	BDM6	BDM5	BDM4	BDM3	BDM2	BDM1	BDM0
	BAR_1	BA31	BA30	BA29	BA28	BA27	BA26	BA25	BA24
		BA23	BA22	BA21	BA20	BA19	BA18	BA17	BA16
		BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8
		BA7	BA6	BA5	BA4	BA3	BA2	BA1	BA0

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
UBC	BAMR_1	BAM31	BAM30	BAM29	BAM28	BAM27	BAM26	BAM25	BAM24
		BAM23	BAM22	BAM21	BAM20	BAM19	BAM18	BAM17	BAM16
		BAM15	BAM14	BAM13	BAM12	BAM11	BAM10	BAM9	BAM8
		BAM7	BAM6	BAM5	BAM4	BAM3	BAM2	BAM1	BAM0
	BBR_1	—	—	UBID	DBE	—	—	CP[1]	CP[0]
		CD[1]	CD[0]	ID[1]	ID[0]	RW[1]	RW[0]	SZ[1]	SZ[0]
	BDR_1	BD31	BD30	BD29	BD28	BD27	BD26	BD25	BD24
		BD23	BD22	BD21	BD20	BD19	BD18	BD17	BD16
		BD15	BD14	BD13	BD12	BD11	BD10	BD9	BD8
		BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0
	BDMR_1	BDM31	BDM30	BDM29	BDM28	BDM27	BDM26	BDM25	BDM24
		BDM23	BDM22	BDM21	BDM20	BDM19	BDM18	BDM17	BDM16
		BDM15	BDM14	BDM13	BDM12	BDM11	BDM10	BDM9	BDM8
		BDM7	BDM6	BDM5	BDM4	BDM3	BDM2	BDM1	BDM0
	BRCR	—	—	—	—	—	—	—	—
		—	—	—	—	UTOD1	UTOD0	CKS[1]	CKS[0]
		SCMFC0	SCMFC1	SCMFD0	SCMFD1	—	—	—	—
		—	PCB1	PCB0	—	—	—	—	—
高速缓存	CCR1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	ICF	—	—	ICE
		—	—	—	—	OCF	—	WT	OCE
	CCR2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	LE
		—	—	—	—	—	—	W3LOAD	W3LOCK
		—	—	—	—	—	—	W2LOAD	W2LOCK
BSC	CMNCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	BLOCK	DPRTY[1]	DPRTY[0]	DMAIW[2]
		DMAIW[1]	DMAIW[0]	DMAIWA	—	—	—	HIZMEM	HIZCNT
	CS0BCR	—	IWW[2]	IWW[1]	IWW[0]	IWRWD[2]	IWRWD[1]	IWRWD[0]	IWRWS[2]
		IWRWS[1]	IWRWS[0]	IWRRD[2]	IWRRD[1]	IWRRD[0]	IWRRS[2]	IWRRS[1]	IWRRS[0]
		—	TYPE[2]	TYPE[1]	TYPE[0]	ENDIAN	BSZ[1]	BSZ[0]	—
		—	—	—	—	—	—	—	—

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
BSC	CS1BCR	—	IWW[2]	IWW[1]	IWW[0]	IWRWD[2]	IWRWD[1]	IWRWD[0]	IWRWS[2]
		IWRWS[1]	IWRWS[0]	IWRRD[2]	IWRRD[1]	IWRRD[0]	IWRRS[2]	IWRRS[1]	IWRRS[0]
		—	TYPE[2]	TYPE[1]	TYPE[0]	ENDIAN	BSZ[1]	BSZ[0]	—
		—	—	—	—	—	—	—	—
	CS2BCR	—	IWW[2]	IWW[1]	IWW[0]	IWRWD[2]	IWRWD[1]	IWRWD[0]	IWRWS[2]
		IWRWS[1]	IWRWS[0]	IWRRD[2]	IWRRD[1]	IWRRD[0]	IWRRS[2]	IWRRS[1]	IWRRS[0]
		—	TYPE[2]	TYPE[1]	TYPE[0]	ENDIAN	BSZ[1]	BSZ[0]	—
		—	—	—	—	—	—	—	—
	CS3BCR	—	IWW[2]	IWW[1]	IWW[0]	IWRWD[2]	IWRWD[1]	IWRWD[0]	IWRWS[2]
		IWRWS[1]	IWRWS[0]	IWRRD[2]	IWRRD[1]	IWRRD[0]	IWRRS[2]	IWRRS[1]	IWRRS[0]
		—	TYPE[2]	TYPE[1]	TYPE[0]	ENDIAN	BSZ[1]	BSZ[0]	—
		—	—	—	—	—	—	—	—
	CS4BCR	—	IWW[2]	IWW[1]	IWW[0]	IWRWD[2]	IWRWD[1]	IWRWD[0]	IWRWS[2]
		IWRWS[1]	IWRWS[0]	IWRRD[2]	IWRRD[1]	IWRRD[0]	IWRRS[2]	IWRRS[1]	IWRRS[0]
		—	TYPE[2]	TYPE[1]	TYPE[0]	ENDIAN	BSZ[1]	BSZ[0]	—
		—	—	—	—	—	—	—	—
	CS5BCR	—	IWW[2]	IWW[1]	IWW[0]	IWRWD[2]	IWRWD[1]	IWRWD[0]	IWRWS[2]
		IWRWS[1]	IWRWS[0]	IWRRD[2]	IWRRD[1]	IWRRD[0]	IWRRS[2]	IWRRS[1]	IWRRS[0]
		—	TYPE[2]	TYPE[1]	TYPE[0]	ENDIAN	BSZ[1]	BSZ[0]	—
		—	—	—	—	—	—	—	—
	CS6BCR	—	IWW[2]	IWW[1]	IWW[0]	IWRWD[2]	IWRWD[1]	IWRWD[0]	IWRWS[2]
		IWRWS[1]	IWRWS[0]	IWRRD[2]	IWRRD[1]	IWRRD[0]	IWRRS[2]	IWRRS[1]	IWRRS[0]
		—	TYPE[2]	TYPE[1]	TYPE[0]	ENDIAN	BSZ[1]	BSZ[0]	—
		—	—	—	—	—	—	—	—
	CS7BCR	—	IWW[2]	IWW[1]	IWW[0]	IWRWD[2]	IWRWD[1]	IWRWD[0]	IWRWS[2]
		IWRWS[1]	IWRWS[0]	IWRRD[2]	IWRRD[1]	IWRRD[0]	IWRRS[2]	IWRRS[1]	IWRRS[0]
		—	TYPE[2]	TYPE[1]	TYPE[0]	ENDIAN	BSZ[1]	BSZ[0]	—
		—	—	—	—	—	—	—	—
	CS0WCR*1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	SW[1]	SW[0]	WR[3]	WR[2]	WR[1]
		WR[0]	WM	—	—	—	—	HW[1]	HW[0]
	CS0WCR*2	—	—	—	—	—	—	—	—
		—	—	BST[1]	BST[0]	—	—	BW[1]	BW[0]
		—	—	—	—	—	W[3]	W[2]	W[1]
		W[0]	WM	—	—	—	—	—	—

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
BSC	CS0WCR*3	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	BW[1]	BW[0]
		—	—	—	—	—	W[3]	W[2]	W[1]
		W[0]	WM	—	—	—	—	—	—
	CS1WCR*4	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	WW[2]	WW[1]	WW[0]
		—	—	—	SW[1]	SW[0]	WR[3]	WR[2]	WR[1]
		WR[0]	WM	—	—	—	—	HW[1]	HW[0]
	CS2WCR*1	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	—	—	—
		—	—	—	—	—	WR[3]	WR[2]	WR[1]
		WR[0]	WM	—	—	—	—	—	—
	CS2WCR*2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	A2CL1
		A2CL0	—	—	—	—	—	—	—
	CS3WCR*1	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	—	—	—
		—	—	—	—	—	WR[3]	WR[2]	WR[1]
		WR[0]	WM	—	—	—	—	—	—
	CS3WCR*5	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	WTRP[1]	WTRP[0]	—	WTRCD[1]	WTRCD[0]	—	A3CL1
		A3CL0	—	—	TRWL[1]	TRWL[0]	—	WTRC[1]	WTRC[0]
	CS4WCR*1	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	WW[2]	WW[1]	WW[0]
		—	—	—	SW[1]	SW[0]	WR[3]	WR[2]	WR[1]
		WR[0]	WM	—	—	—	—	HW[1]	HW[0]
	CS4WCR*2	—	—	—	—	—	—	—	—
		—	—	BST[1]	BST[0]	—	—	BW[1]	BW[0]
		—	—	—	SW[1]	SW[0]	W[3]	W[2]	W[1]
		W[0]	WM	—	—	—	—	HW[1]	HW[0]
	CS5WCR*1	—	—	—	—	—	—	—	—
		—	—	SZSEL	MPXW/BAS	—	WW[2]	WW[1]	WW[0]
		—	—	—	SW[1]	SW[0]	WR[3]	WR[2]	WR[1]
		WR[0]	WM	—	—	—	—	HW[1]	HW[0]

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
BSC	CS5WCR*6	—	—	—	—	—	—	—	—
		—	—	SA[1]	SA[0]	—	—	—	—
		—	TED[3]	TED[2]	TED[1]	TED[0]	PCW[1]	PCW[0]	PCW[1]
		PCW[0]	WM	—	—	TEH[3]	TEH[2]	TEH[1]	TEH[0]
	CS6WCR*1	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	—	—	—
		—	—	—	SW[1]	SW[0]	WR[3]	WR[2]	WR[1]
		WR[0]	WM	—	—	—	—	HW[1]	HW[0]
	CS6WCR*7	—	—	—	—	—	—	—	—
		—	—	MPXAW[1]	MPXAW[0]	MPXMD	—	BW[1]	BW[0]
		—	—	—	—	—	W[3]	W[2]	W[1]
		W[0]	WM	—	—	—	—	—	—
	CS6WCR*6	—	—	—	—	—	—	—	—
		—	—	SA[1]	SA[0]	—	—	—	—
		—	TED[3]	TED[2]	TED[1]	TED[0]	PCW[1]	PCW[0]	PCW[1]
		PCW[0]	WM	—	—	TEH[3]	TEH[2]	TEH[1]	TEH[0]
	CS7WCR*4	—	—	—	—	—	—	—	—
		—	—	—	BAS	—	WW[2]	WW[1]	WW[0]
		—	—	—	SW[1]	SW[0]	WR[3]	WR[2]	WR[1]
		WR[0]	WM	—	—	—	—	HW[1]	HW[0]
	SDCR	—	—	—	—	—	—	—	—
		—	—	—	A2ROW[1]	A2ROW[0]	—	A2COL[1]	A2COL[0]
		—	—	DEEP	SLOW	RFSH	RMODE	PDOWN	BACTV
		—	—	—	A3ROW[1]	A3ROW[0]	—	A3COL[1]	A3COL[0]
	RTCSR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		CMF	CMIE	CKS[2]	CKS[1]	CKS[0]	RRC[2]	RRC[1]	RRC[0]
	RTCNT	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
	RTCOR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
DMAC	SAR0								
	DAR0								
	DMATCR0	—	—	—	—	—	—	—	—
	CHCR0	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		DO	TL	—	TEMASK	HE	HIE	AM	AL
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		DL	DS	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR0								
	RDAR0								
	RDMATCR0	—	—	—	—	—	—	—	—
	SAR1								
	DAR1								

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
DMAC	DMATCR1	—	—	—	—	—	—	—	—
	CHCR1	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		DO	TL	—	TEMASK	HE	HIE	AM	AL
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		DL	DS	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR1								
	RDAR1								
	RDMATCR1	—	—	—	—	—	—	—	—
	SAR2								
	DAR2								
	DMATCR2	—	—	—	—	—	—	—	—
	CHCR2	TC	—	RLDSAR	RLDDAR	—	—	—	—
		DO	—	—	TEMASK	HE	HIE	AM	AL
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		DL	DS	TB	TS[1]	TS[0]	IE	TE	DE

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
DMAC	RSAR2								
	RDAR2								
	RDMATCR2	—	—	—	—	—	—	—	—
	SAR3								
	DAR3								
	DMATCR3	—	—	—	—	—	—	—	—
	CHCR3	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		DO	—	—	TEMASK	HE	HIE	AM	AL
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		DL	DS	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR3								
	RDAR3								

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
DMAC	RDMATCR3	—	—	—	—	—	—	—	—
	SAR4								
	DAR4								
	DMATCR4	—	—	—	—	—	—	—	—
	CHCR4	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR4								
	RDAR4								
	RDMATCR4	—	—	—	—	—	—	—	—
	SAR5								

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
DMAC	DAR5								
	DMATCR5	—	—	—	—	—	—	—	—
	CHCR5	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR5								
	RDAR5								
	RDMATCR5	—	—	—	—	—	—	—	—
	SAR6								
	DAR6								
	DMATCR6	—	—	—	—	—	—	—	—

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
DMAC	CHCR6	TC	—	RLDSAR	RLDDAR	—	—	—	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR6								
	RDAR6								
	RDMATCR6	—	—	—	—	—	—	—	—
	SAR7								
	DAR7								
	DMATCR7	—	—	—	—	—	—	—	—
	CHCR7	TC	—	RLDSAR	RLDDAR	—	DAF	SAF	—
		—	—	—	TEMASK	HE	HIE	—	—
		DM[1]	DM[0]	SM[1]	SM[0]	RS[3]	RS[2]	RS[1]	RS[0]
		—	—	TB	TS[1]	TS[0]	IE	TE	DE
	RSAR7								

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
DMAC	RDAR7								
	RDMATCR7	—	—	—	—	—	—	—	—
	DMAOR	—	—	CMS[1]	CMS[0]	—	—	PR[1]	PR[0]
		—	—	—	—	—	AE	NMIF	DME
	DMARS0	CH1MID[5]	CH1MID[4]	CH1MID[3]	CH1MID[2]	CH1MID[1]	CH1MID[0]	CH1RID[1]	CH1RID[0]
		CH0MID[5]	CH0MID[4]	CH0MID[3]	CH0MID[2]	CH0MID[1]	CH0MID[0]	CH0RID[1]	CH0RID[0]
	DMARS1	CH3MID[5]	CH3MID[4]	CH3MID[3]	CH3MID[2]	CH3MID[1]	CH3MID[0]	CH3RID[1]	CH3RID[0]
		CH2MID[5]	CH2MID[4]	CH2MID[3]	CH2MID[2]	CH2MID[1]	CH2MID[0]	CH2RID[1]	CH2RID[0]
	DMARS2	CH5MID[5]	CH5MID[4]	CH5MID[3]	CH5MID[2]	CH5MID[1]	CH5MID[0]	CH5RID[1]	CH5RID[0]
		CH4MID[5]	CH4MID[4]	CH4MID[3]	CH4MID[2]	CH4MID[1]	CH4MID[0]	CH4RID[1]	CH4RID[0]
	DMARS3	CH7MID[5]	CH7MID[4]	CH7MID[3]	CH7MID[2]	CH7MID[1]	CH7MID[0]	CH7RID[1]	CH7RID[0]
		CH6MID[5]	CH6MID[4]	CH6MID[3]	CH6MID[2]	CH6MID[1]	CH6MID[0]	CH6RID[1]	CH6RID[0]
MTU2	TCR_0	CCLR[2]	CCLR[1]	CCLR[0]	CKEG[1]	CKEG[0]	TPSC[2]	TPSC[1]	TPSC[0]
	TMDR_0	—	BFE	BFB	BFA	MD[3]	MD[2]	MD[1]	MD[0]
	TIORH_0	IOB[3]	IOB[2]	IOB[1]	IOB[0]	IOA[3]	IOA[2]	IOA[1]	IOA[0]
	TIORL_0	IOD[3]	IOD[2]	IOD[1]	IOD[0]	IOC[3]	IOC[2]	IOC[1]	IOC[0]
	TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
	TSR_0	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_0								
	TGRA_0								
	TGRB_0								
	TGRC_0								
	TGRD_0								
	TGRE_0								

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
MTU2	TGRF_0								
	TIER2_0	TTGE2	—	—	—	—	—	TGIEF	TGIEE
	TSR2_0	—	—	—	—	—	—	TGFF	TGFE
	TBTM_0	—	—	—	—	—	TTSE	TTSB	TTSA
	TCR_1	—	CCLR[1]	CCLR[0]	CKEG[1]	CKEG[0]	TPSC[2]	TPSC[1]	TPSC[0]
	TMDR_1	—	—	—	—	MD[3]	MD[2]	MD[1]	MD[0]
	TIOR_1	IOB[3]	IOB[2]	IOB[1]	IOB[0]	IOA[3]	IOA[2]	IOA[1]	IOA[0]
	TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
	TSR_1	TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_1								
	TGRA_1								
	TGRB_1								
	TICCR	—	—	—	—	I2BE	I2AE	I1BE	I1AE
	TCR_2	—	CCLR[1]	CCLR[0]	CKEG[1]	CKEG[0]	TPSC[2]	TPSC[1]	TPSC[0]
	TMDR_2	—	—	—	—	MD[3]	MD[2]	MD[1]	MD[0]
	TIOR_2	IOB[3]	IOB[2]	IOB[1]	IOB[0]	IOA[3]	IOA[2]	IOA[1]	IOA[0]
	TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
	TSR_2	TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_2								
	TGRA_2								
	TGRB_2								
	TCR_3	CCLR[2]	CCLR[1]	CCLR[0]	CKEG[1]	CKEG[0]	TPSC[2]	TPSC[1]	TPSC[0]
	TMDR_3	—	—	BFB	BFA	MD[3]	MD[2]	MD[1]	MD[0]
	TIORH_3	IOB[3]	IOB[2]	IOB[1]	IOB[0]	IOA[3]	IOA[2]	IOA[1]	IOA[0]
	TIORL_3	IOD[3]	IOD[2]	IOD[1]	IOD[0]	IOC[3]	IOC[2]	IOC[1]	IOC[0]
	TIER_3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
	TSR_3	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_3								

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
MTU2	TGRA_3								
	TGRB_3								
	TGRC_3								
	TGRD_3								
	TBTM_3	—	—	—	—	—	—	TTSB	TTSA
	TCR_4	CCLR[2]	CCLR[1]	CCLR[0]	CKEG[1]	CKEG[0]	TPSC[2]	TPSC[1]	TPSC[0]
	TMDR_4	—	—	BFB	BFA	MD[3]	MD[2]	MD[1]	MD[0]
	TIORH_4	IOB[3]	IOB[2]	IOB[1]	IOB[0]	IOA[3]	IOA[2]	IOA[1]	IOA[0]
	TIORL_4	IOD[3]	IOD[2]	IOD[1]	IOD[0]	IOC[3]	IOC[2]	IOC[1]	IOC[0]
	TIER_4	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
	TSR_4	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
	TCNT_4								
	TGRA_4								
	TGRB_4								
	TGRC_4								
	TGRD_4								
	TBTM_4	—	—	—	—	—	—	TTSB	TTSA
	TADCR	BF[1]	BF[0]	—	—	—	—	—	—
		UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
	TADCORA_4								
	TADCORB_4								
	TADCOBRA_4								
	TADCOBRB_4								

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
MTU2	TSTR	CST4	CST3	—	—	—	CST2	CST1	CST0
	TSYR	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0
	TRWER	—	—	—	—	—	—	—	RWE
	TOER	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
	TOCR1	—	PSYE	—	—	TOCL	TOCS	OLSN	PLSP
	TOCR2	BF[1]	BF[0]	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
	TGCR	—	BDC	N	P	FB	WF	VF	UF
	TCDR								
	TDDR								
	TCNTS								
	TCBR								
	TITCR	T3AEN	3ACOR[2]	3ACOR[1]	3ACOR[0]	T4VEN	4VCOR[2]	4VCOR[1]	4VCOR[0]
	TITCNT	—	3ACNT[2]	3ACNT[1]	3ACNT[0]	—	4VCNT[2]	4VCNT[1]	4VCNT[0]
	TBTER	—	—	—	—	—	—	BTE[1]	BTE[0]
	TDER	—	—	—	—	—	—	—	TDER
	TSYCR	CE0A	CE0B	CE0C	CE0D	CE1A	CE1B	CE2A	CE2B
	TWCR	CCE	—	—	—	—	—	—	WRE
	TOLBR	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
CMT	CMSTR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	STR1	STR0
	CMCSR0	—	—	—	—	—	—	—	—
		CMF	CMIE	—	—	—	—	CKS[1]	CKS[0]
	CMCNT0								
	CMCOR0								
	CMCSR1	—	—	—	—	—	—	—	—
		CMF	CMIE	—	—	—	—	CKS[1]	CKS[0]
	CMCNT1								
	CMCOR1								

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
WDT	WTCNT								
	WTCSR	IOVF	WT/IT	TME	—	—	CKS[2]	CKS[1]	CKS[0]
	WRCSR	WOVF	RSTE	RSTS	—	—	—	—	—
RTC	R64CNT	—	1Hz	2Hz	4Hz	8Hz	16Hz	32Hz	64Hz
	RSECCNT	—	10 秒 [2]	10 秒 [1]	10 秒 [0]	1 秒 [3]	1 秒 [2]	1 秒 [1]	1 秒 [0]
	RMINCNT	—	10 分 [2]	10 分 [1]	10 分 [0]	1 分 [3]	1 分 [2]	1 分 [1]	1 分 [0]
	RHRCNT	—	—	10 小时 [1]	10 小时 [0]	1 小时 [3]	1 小时 [2]	1 小时 [1]	1 小时 [0]
	RWKCNT	—	—	—	—	—	星期 [2]	星期 [1]	星期 [0]
	RDAYCNT	—	—	10 日 [1]	10 日 [0]	1 日 [3]	1 日 [2]	1 日 [1]	1 日 [0]
	RMONCNT	—	—	—	10 月	1 月 [3]	1 月 [2]	1 月 [1]	1 月 [0]
	RYRCNT	1000 年 [3]	1000 年 [2]	1000 年 [1]	1000 年 [0]	100 年 [3]	100 年 [2]	100 年 [1]	100 年 [0]
		10 年 [3]	10 年 [2]	10 年 [1]	10 年 [0]	1 年 [3]	1 年 [2]	1 年 [1]	1 年 [0]
	RSECAR	ENB	10 秒 [2]	10 秒 [1]	10 秒 [0]	1 秒 [3]	1 秒 [2]	1 秒 [1]	1 秒 [0]
	RMINAR	ENB	10 分 [2]	10 分 [1]	10 分 [0]	1 分 [3]	1 分 [2]	1 分 [1]	1 分 [0]
	RHRAR	ENB	—	10 小时 [1]	10 小时 [0]	1 小时 [3]	1 小时 [2]	1 小时 [1]	1 小时 [0]
	RWKAR	ENB	—	—	—	—	星期 [2]	星期 [1]	星期 [0]
	RDAYAR	ENB	—	10 日 [1]	10 日 [0]	1 日 [3]	1 日 [2]	1 日 [1]	1 日 [0]
	RMONAR	ENB	—	—	10 月	1 月 [3]	1 月 [2]	1 月 [1]	1 月 [0]
	RYRAR	1000 年 [3]	1000 年 [2]	1000 年 [1]	1000 年 [0]	100 年 [3]	100 年 [2]	100 年 [1]	100 年 [0]
		10 年 [3]	10 年 [2]	10 年 [1]	10 年 [0]	1 年 [3]	1 年 [2]	1 年 [1]	1 年 [0]
	RCR1	CF	—	—	CIE	AIE	—	—	AF
	RCR2	PEF	PES[2]	PES[1]	PES[0]	RTCEN	ADJ	RESET	START
	RCR3	ENB	—	—	—	—	—	—	—
SCIF	SCSMR_0	—	—	—	—	—	—	—	—
		C/A	CHR	PE	O/E	STOP	—	CKS[1]	CKS[0]
	SCBRR_0								
	SCSCR_0	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE[1]	CKE[0]
	SCFTDR_0								
	SCFSR_0	PER[3]	PER[2]	PER[1]	PER[0]	FER[3]	FER[2]	FER[1]	FER[0]
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_0								
	SCFCR_0	—	—	—	—	—	RSTRG[2]	RSTRG[1]	RSTRG[0]
		RTRG[1]	RTRG[0]	TTRG[1]	TTRG[0]	MCE	TFRST	RFRST	LOOP
	SCFDR_0	—	—	—	T[4]	T[3]	T[2]	T[1]	T[0]
		—	—	—	R[4]	R[3]	R[2]	R[1]	R[0]

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
SCIF	SCSPTR_0	—	—	—	—	—	—	—	—
		—	—	—	—	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_0	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCEMR_0	—	—	—	—	—	—	—	—
		BGDM	—	—	—	—	—	—	ABCS
	SCSMR_1	—	—	—	—	—	—	—	—
		C/A	CHR	PE	O/E	STOP	—	CKS[1]	CKS[0]
	SCBRR_1								
	SCSCR_1	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE[1]	CKE[0]
	SCFTDR_1								
	SCFSR_1	PER[3]	PER[2]	PER[1]	PER[0]	FER[3]	FER[2]	FER[1]	FER[0]
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_1								
	SCFCR_1	—	—	—	—	—	RSTRG[2]	RSTRG[1]	RSTRG[0]
		RTRG[1]	RTRG[0]	TTRG[1]	TTRG[0]	MCE	TFRST	RFRST	LOOP
	SCFDR_1	—	—	—	T[4]	T[3]	T[2]	T[1]	T[0]
		—	—	—	R[4]	R[3]	R[2]	R[1]	R[0]
	SCSPTR_1	—	—	—	—	—	—	—	—
		—	—	—	—	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_1	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCEMR_1	—	—	—	—	—	—	—	—
		BGDM	—	—	—	—	—	—	ABCS
	SCSMR_2	—	—	—	—	—	—	—	—
		C/A	CHR	PE	O/E	STOP	—	CKS[1]	CKS[0]
	SCBRR_2								
	SCSCR_2	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE[1]	CKE[0]
	SCFTDR_2								
	SCFSR_2	PER[3]	PER[2]	PER[1]	PER[0]	FER[3]	FER[2]	FER[1]	FER[0]
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_2								
	SCFCR_2	—	—	—	—	—	RSTRG[2]	RSTRG[1]	RSTRG[0]
		RTRG[1]	RTRG[0]	TTRG[1]	TTRG[0]	MCE	TFRST	RFRST	LOOP

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
SCIF	SCFDR_2	—	—	—	T[4]	T[3]	T[2]	T[1]	T[0]
		—	—	—	R[4]	R[3]	R[2]	R[1]	R[0]
	SCSPTR_2	—	—	—	—	—	—	—	—
		—	—	—	—	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCEMR_2	—	—	—	—	—	—	—	—
		BGDM	—	—	—	—	—	—	ABCS
	SCSMR_3	—	—	—	—	—	—	—	—
		C/A	CHR	PE	O/E	STOP	—	CKS[1]	CKS[0]
	SCBRR_3								
	SCSCR_3	—	—	—	—	—	—	—	—
		TIE	RIE	TE	RE	REIE	—	CKE[1]	CKE[0]
	SCFTDR_3								
	SCFSR_3	PER[3]	PER[2]	PER[1]	PER[0]	FER[3]	FER[2]	FER[1]	FER[0]
		ER	TEND	TDFE	BRK	FER	PER	RDF	DR
	SCFRDR_3								
	SCFCR_3	—	—	—	—	—	RSTRG[2]	RSTRG[1]	RSTRG[0]
		RTRG[1]	RTRG[0]	TTRG[1]	TTRG[0]	MCE	TFRST	RFRST	LOOP
	SCFDR_3	—	—	—	T[4]	T[3]	T[2]	T[1]	T[0]
		—	—	—	R[4]	R[3]	R[2]	R[1]	R[0]
	SCSPTR_3	—	—	—	—	—	—	—	—
		RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT
	SCLSR_3	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	ORER
	SCEMR_3	—	—	—	—	—	—	—	—
		BGDM	—	—	—	—	—	—	ABCS
SSU	SSCRH_0	MSS	BIDE	—	SOL	SOLP	—	CSS[1]	CSS[0]
	SSCRL_0	FCLRM	SSUMS	SRES	—	—	—	DATS[1]	DATS[0]
	SSMR_0	MLS	CPOS	CPHS	—	—	CKS[2]	CKS[1]	CKS[0]
	SSER_0	TE	RE	—	—	TEIE	TIE	RIE	CEIE
	SSSR_0	—	ORER	—	—	TEND	TDRE	RDRF	CE
	SSCR2_0	—	—	—	TENDSTS	SCSATS	SSODTS	—	—
	SSTDR0_0								
	SSTDR1_0								
	SSTDR2_0								
	SSTDR3_0								

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
SSU	SSRDR0_0								
	SSRDR1_0								
	SSRDR2_0								
	SSRDR3_0								
	SSCRH_1	MSS	BIDE	—	SOL	SOLP	—	CSS[1]	CSS[0]
	SSCRL_1	FCLRM	SSUMS	SRES	—	—	—	DATS[1]	DATS[0]
	SSMR_1	MLS	CPOS	CPHS	—	—	CKS[2]	CKS[1]	CKS[0]
	SSER_1	TE	RE	—	—	TEIE	TIE	RIE	CEIE
	SSSR_1	—	ORER	—	—	TEND	TDRE	RDRF	CE
	SSCR2_1	—	—	—	TENDSTS	SCSATS	SSODTS	—	—
	SSTDR0_1								
	SSTDR1_1								
	SSTDR2_1								
	SSTDR3_1								
	SSRDR0_1								
	SSRDR1_1								
	SSRDR2_1								
	SSRDR3_1								
IIC3	ICCR1_0	ICE	RCVD	MST	TRS	CKS[3]	CKS[2]	CKS[1]	CKS[0]
	ICCR2_0	BBSY	SCP	SDAO	SDAOP	SCL	—	IICRST	—
	ICMR_0	MLS	—	—	—	BCWP	BC[2]	BC[1]	BC[0]
	ICIER_0	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
	ICSR_0	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
	SAR_0	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA[0]	FS
	ICDRT_0								
	ICDRR_0								
	NF2CYC_0	—	—	—	—	—	—	PRS	NF2CYC
	ICCR1_1	ICE	RCVD	MST	TRS	CKS[3]	CKS[2]	CKS[1]	CKS[0]
	ICCR2_1	BBSY	SCP	SDAO	SDAOP	SCL	—	IICRST	—
	ICMR_1	MLS	—	—	—	BCWP	BC[2]	BC[1]	BC[0]
	ICIER_1	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
	ICSR_1	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
	SAR_1	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA[0]	FS
	ICDRT_1								
	ICDRR_1								
	NF2CYC_1	—	—	—	—	—	—	PRS	NF2CYC

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
IIC3	ICCR1_2	ICE	RCVD	MST	TRS	CKS[3]	CKS[2]	CKS[1]	CKS[0]
	ICCR2_2	BBSY	SCP	SDAO	SDAOP	SCL	—	IICRST	—
	ICMR_2	MLS	—	—	—	BCWP	BC[2]	BC[1]	BC[0]
	ICIER_2	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
	ICSR_2	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
	SAR_2	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA[0]	FS
	ICDRT_2								
	ICDRR_2								
	NF2CYC_2	—	—	—	—	—	—	PRS	NF2CYC
	ICCR1_3	ICE	RCVD	MST	TRS	CKS[3]	CKS[2]	CKS[1]	CKS[0]
	ICCR2_3	BBSY	SCP	SDAO	SDAOP	SCL	—	IICRST	—
	ICMR_3	MLS	—	—	—	BCWP	BC[2]	BC[1]	BC[0]
	ICIER_3	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
	ICSR_3	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
	SAR_3	SVA[6]	SVA[5]	SVA[4]	SVA[3]	SVA[2]	SVA[1]	SVA[0]	FS
	ICDRT_3								
	ICDRR_3								
	NF2CYC_3	—	—	—	—	—	—	PRS	NF2CYC
SSI	SSICR_0	—	—	—	DMEN	UIEN	OIEN	IIEN	DIEN
		CHNL[1]	CHNL[0]	DWL[2]	DWL[1]	DWL[0]	SWL[2]	SWL[1]	SWL[0]
		SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL
		BREN	CKDV[2]	CKDV[1]	CKDV[0]	MUEN	CPEN	TRMD	EN
	SSISR_0	—	—	—	DMRQ	UIRQ	OIRQ	IIRQ	DIRQ
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	CHNO1	CHNO0	SWNO	IDST
	SSITDR_0								
	SSIRDR_0								

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
SSI	SSICR_1	—	—	—	DMEN	UIEN	OIEN	IIEN	DIEN
		CHNL[1]	CHNL[0]	DWL[2]	DWL[1]	DWL[0]	SWL[2]	SWL[1]	SWL[0]
		SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL
		BREN	CKDV[2]	CKDV[1]	CKDV[0]	MUEN	CPEN	TRMD	EN
	SSISR_1	—	—	—	DMRQ	UIRQ	OIRQ	IIRQ	DIRQ
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	CHNO1	CHNO0	SWNO	IDST
	SSITDR_1								
	SSIRDR_1								
	SSICR_2	—	—	—	DMEN	UIEN	OIEN	IIEN	DIEN
		CHNL[1]	CHNL[0]	DWL[2]	DWL[1]	DWL[0]	SWL[2]	SWL[1]	SWL[0]
		SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL
		BREN	CKDV[2]	CKDV[1]	CKDV[0]	MUEN	CPEN	TRMD	EN
	SSISR_2	—	—	—	DMRQ	UIRQ	OIRQ	IIRQ	DIRQ
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	CHNO1	CHNO0	SWNO	IDST
	SSITDR_2								
	SSIRDR_2								
	SSICR_3	—	—	—	DMEN	UIEN	OIEN	IIEN	DIEN
		CHNL[1]	CHNL[0]	DWL[2]	DWL[1]	DWL[0]	SWL[2]	SWL[1]	SWL[0]
		SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL
		BREN	CKDV[2]	CKDV[1]	CKDV[0]	MUEN	CPEN	TRMD	EN

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
SSI	SSISR_3	—	—	—	DMRQ	UIRQ	OIRQ	IIRQ	DIRQ
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	CHNO1	CHNO0	SWNO	IDST
	SSITDR_3								
	SSIRDR_3								
RCAN-TL1	MCR_0	MCR15	MCR14	—	—	—	TST[2]	TST[1]	TST[0]
		MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0
	GSR_0	—	—	—	—	—	—	—	—
		—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
	BCR1_0	TSEG1[3]	TSEG1[2]	TSEG1[1]	TSEG1[0]	—	TSEG2[2]	TSEG2[1]	TSEG2[0]
		—	—	SJW[1]	SJW[0]	—	—	—	BSP
	BCR0_0	—	—	—	—	—	—	—	—
		BRP[7]	BRP[6]	BRP[5]	BRP[4]	BRP[3]	BRP[2]	BRP[1]	BRP[0]
	IRR_0	IRR15	IRR14	IRR13	IRR12	IRR11	IRR10	IRR9	IRR8
		IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
	IMR_0	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8
		IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
	TEC_REC_0	TEC[7]	TEC[6]	TEC[5]	TEC[4]	TEC[3]	TEC[2]	TEC[1]	TEC[0]
		REC[7]	REC[6]	REC[5]	REC[4]	REC[3]	REC[2]	REC[1]	REC[0]
	TXPR1_0	TXPR1[15]	TXPR1[14]	TXPR1[13]	TXPR1[12]	TXPR1[11]	TXPR1[10]	TXPR1[9]	TXPR1[8]
		TXPR1[7]	TXPR1[6]	TXPR1[5]	TXPR1[4]	TXPR1[3]	TXPR1[2]	TXPR1[1]	TXPR1[0]
	TXPR0_0	TXPR0[15]	TXPR0[14]	TXPR0[13]	TXPR0[12]	TXPR0[11]	TXPR0[10]	TXPR0[9]	TXPR0[8]
		TXPR0[7]	TXPR0[6]	TXPR0[5]	TXPR0[4]	TXPR0[3]	TXPR0[2]	TXPR0[1]	—
	TXCR1_0	TXCR1[15]	TXCR1[14]	TXCR1[13]	TXCR1[12]	TXCR1[11]	TXCR1[10]	TXCR1[9]	TXCR1[8]
		TXCR1[7]	TXCR1[6]	TXCR1[5]	TXCR1[4]	TXCR1[3]	TXCR1[2]	TXCR1[1]	TXCR1[0]
	TXCR0_0	TXCR0[15]	TXCR0[14]	TXCR0[13]	TXCR0[12]	TXCR0[11]	TXCR0[10]	TXCR0[9]	TXCR0[8]
		TXCR0[7]	TXCR0[6]	TXCR0[5]	TXCR0[4]	TXCR0[3]	TXCR0[2]	TXCR0[1]	—
	TXACK1_0	TXACK1[15]	TXACK1[14]	TXACK1[13]	TXACK1[12]	TXACK1[11]	TXACK1[10]	TXACK1[9]	TXACK1[8]
		TXACK1[7]	TXACK1[6]	TXACK1[5]	TXACK1[4]	TXACK1[3]	TXACK1[2]	TXACK1[1]	TXACK1[0]

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
RCAN-TL1	TXACK0_0	TXACK0 [15]	TXACK0 [14]	TXACK0 [13]	TXACK0 [12]	TXACK0 [11]	TXACK0 [10]	TXACK0 [9]	TXACK0 [8]
		TXACK0 [7]	TXACK0 [6]	TXACK0 [5]	TXACK0 [4]	TXACK0 [3]	TXACK0 [2]	TXACK0 [1]	—
	ABACK1_0	ABACK1 [15]	ABACK1 [14]	ABACK1 [13]	ABACK1 [12]	ABACK1 [11]	ABACK1 [10]	ABACK1 [9]	ABACK1 [8]
		ABACK1 [7]	ABACK1 [6]	ABACK1 [5]	ABACK1 [4]	ABACK1 [3]	ABACK1 [2]	ABACK1 [1]	ABACK1 [0]
	ABACK0_0	ABACK0 [15]	ABACK0 [14]	ABACK0 [13]	ABACK0 [12]	ABACK0 [11]	ABACK0 [10]	ABACK0 [9]	ABACK0 [8]
		ABACK0 [7]	ABACK0 [6]	ABACK0 [5]	ABACK0 [4]	ABACK0 [3]	ABACK0 [2]	ABACK0 [1]	—
	RXPR1_0	RXPR1 [15]	RXPR1 [14]	RXPR1 [13]	RXPR1 [12]	RXPR1 [11]	RXPR1 [10]	RXPR1 [9]	RXPR1 [8]
		RXPR1 [7]	RXPR1 [6]	RXPR1 [5]	RXPR1 [4]	RXPR1 [3]	RXPR1 [2]	RXPR1 [1]	RXPR1 [0]
	RXPR0_0	RXPR0 [15]	RXPR0 [14]	RXPR0 [13]	RXPR0 [12]	RXPR0 [11]	RXPR0 [10]	RXPR0 [9]	RXPR [8]
		RXPR0 [7]	RXPR0 [6]	RXPR0 [5]	RXPR0 [4]	RXPR0 [3]	RXPR0 [2]	RXPR0 [1]	RXPR0 [0]
	RFPR1_0	RFPR1 [15]	RFPR1 [14]	RFPR1 [13]	RFPR1 [12]	RFPR1 [11]	RFPR1 [10]	RFPR1 [9]	RFPR1 [8]
		RFPR1 [7]	RFPR1 [6]	RFPR1 [5]	RFPR1 [4]	RFPR1 [3]	RFPR1 [2]	RFPR1 [1]	RFPR1 [0]
	RFPR0_0	RFPR0 [15]	RFPR0 [14]	RFPR0 [13]	RFPR0 [12]	RFPR0 [11]	RFPR0 [10]	RFPR0 [9]	RFPR0 [8]
		RFPR0 [7]	RFPR0 [6]	RFPR0 [5]	RFPR0 [4]	RFPR0 [3]	RFPR0 [2]	RFPR0 [1]	RFPR0 [0]
	MBIMR1_0	MBIMR1 [15]	MBIMR1 [14]	MBIMR1 [13]	MBIMR1 [12]	MBIMR1 [11]	MBIMR1 [10]	MBIMR1 [9]	MBIMR1 [8]
		MBIMR1 [7]	MBIMR1 [6]	MBIMR1 [5]	MBIMR1 [4]	MBIMR1 [3]	MBIMR1 [2]	MBIMR1 [1]	MBIMR1 [0]
	MBIMR0_0	MBIMR0 [15]	MBIMR0 [14]	MBIMR0 [13]	MBIMR0 [12]	MBIMR0 [11]	MBIMR0 [10]	MBIMR0 [9]	MBIMR0 [8]
		MBIMR0 [7]	MBIMR0 [6]	MBIMR0 [5]	MBIMR0 [4]	MBIMR0 [3]	MBIMR0 [2]	MBIMR0 [1]	MBIMR0 [0]
	UMSR1_0	UMSR1 [15]	UMSR1 [14]	UMSR1 [13]	UMSR1 [12]	UMSR1 [11]	UMSR1 [10]	UMSR1 [9]	UMSR1 [8]
		UMSR1 [7]	UMSR1 [6]	UMSR1 [5]	UMSR1 [4]	UMSR1 [3]	UMSR1 [2]	UMSR1 [1]	UMSR1 [0]

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
RCAN-TL1	UMSR0_0	UMSR0 [15]	UMSR0 [14]	UMSR0 [13]	UMSR0 [12]	UMSR0 [11]	UMSR0 [10]	UMSR0 [9]	UMSR0 [8]
		UMSR0 [7]	UMSR0 [6]	UMSR0 [5]	UMSR0 [4]	UMSR0 [3]	UMSR0 [2]	UMSR0 [1]	UMSR0 [0]
	TTCR0_0	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	—	—
		—	TCR6	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0
	CMAX_TEW_0	—	—	—	—	—	CMAX[2]	CMAX[1]	CMAX[0]
		—	—	—	TEW[4]	TEW[3]	TEW[2]	TEW[1]	TEW[0]
	RFTROFF_0	RTROFF [7]	RTROFF [6]	RTROFF [5]	RTROFF [4]	RTROFF [3]	RTROFF [2]	RTROFF [1]	RTROFF [0]
		—	—	—	—	—	—	—	—
	TSR_0	—	—	—	—	—	—	—	—
		—	—	—	TSR4	TSR3	TSR2	TSR1	TSR0
	CCR_0	—	—	—	—	—	—	—	—
		—	—	CCR[5]	CCR[4]	CCR[3]	CCR[2]	CCR[1]	CCR[0]
	TCNTR_0	TCNTR [15]	TCNTR [14]	TCNTR [13]	TCNTR [12]	TCNTR [11]	TCNTR [10]	TCNTR [9]	TCNTR [8]
		TCNTR [7]	TCNTR [6]	TCNTR [5]	TCNTR [4]	TCNTR [3]	TCNTR [2]	TCNTR [1]	TCNTR [0]
	CYCTR_0	CYCTR [15]	CYCTR [14]	CYCTR [13]	CYCTR [12]	CYCTR [11]	CYCTR [10]	CYCTR [9]	CYCTR [8]
		CYCTR [7]	CYCTR [6]	CYCTR [5]	CYCTR [4]	CYCTR [3]	CYCTR [2]	CYCTR [1]	CYCTR [0]
	RFMK_0	RFMK [15]	RFMK [14]	RFMK [13]	RFMK [12]	RFMK [11]	RFMK [10]	RFMK [9]	RFMK [8]
		RFMK[7]	RFMK[6]	RFMK[5]	RFMK[4]	RFMK[3]	RFMK[2]	RFMK[1]	RFMK[0]
	TCMR0_0	TCMR0 [15]	TCMR0 [14]	TCMR0 [13]	TCMR0 [12]	TCMR0 [11]	TCMR0 [10]	TCMR0 [9]	TCMR0 [8]
		TCMR0 [7]	TCMR0 [6]	TCMR0 [5]	TCMR0 [4]	TCMR0 [3]	TCMR0 [2]	TCMR0 [1]	TCMR0 [0]
	TCMR1_0	TCMR1 [15]	TCMR1 [14]	TCMR1 [13]	TCMR1 [12]	TCMR1 [11]	TCMR1 [10]	TCMR1 [9]	TCMR1 [8]
		TCMR1 [7]	TCMR1 [6]	TCMR1 [5]	TCMR1 [4]	TCMR1 [3]	TCMR1 [2]	TCMR1 [1]	TCMR1 [0]
	TCMR2_0	TCMR2 [15]	TCMR2 [14]	TCMR2 [13]	TCMR2 [12]	TCMR2 [11]	TCMR2 [10]	TCMR2 [9]	TCMR2 [8]
		TCMR2 [7]	TCMR2 [6]	TCMR2 [5]	TCMR2 [4]	TCMR2 [3]	TCMR2 [2]	TCMR2 [1]	TCMR2 [0]

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
RCAN-TL1	TTTSEL_0	—	TTTSEL[14]	TTTSEL[13]	TTTSEL[12]	TTTSEL[11]	TTTSEL[10]	TTTSEL[9]	TTTSEL[8]
		—	—	—	—	—	—	—	—
	MBn_CONTROL0_H_0 (n=0 ～ 31)*8	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]
		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]
	MBn_CONTROL0_H_0 (n=0 ～ 31)*9	IDE	RTR	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]
		STDID[5]	STDID[4]	STDID[3]	STDID[2]	STDID[1]	STDID[0]	EXTID[17]	EXTID[16]
	MBn_CONTROL0_L_0 (n=0 ～ 31)	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]
		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]
	MBn_LAFM0_0 (n=0 ～ 31)*8	—	STDID_LAFM[10]	STDID_LAFM[9]	STDID_LAFM[8]	STDID_LAFM[7]	STDID_LAFM[6]	STDID_LAFM[5]	STDID_LAFM[4]
		STDID_LAFM[3]	STDID_LAFM[2]	STDID_LAFM[1]	STDID_LAFM[0]	—	IDE	EXTID_LAFM[17]	EXTID_LAFM[16]
	MBn_LAFM0_0 (n=0 ～ 31)*9	IDE	—	—	STDID_LAFM[10]	STDID_LAFM[9]	STDID_LAFM[8]	STDID_LAFM[7]	STDID_LAFM[6]
		STDID_LAFM[5]	STDID_LAFM[4]	STDID_LAFM[3]	STDID_LAFM[2]	STDID_LAFM[1]	STDID_LAFM[0]	EXTID_LAFM[17]	EXTID_LAFM[16]
	MBn_LAFM1_0 (n=0 ～ 31)	EXTID_LAFM[15]	EXTID_LAFM[14]	EXTID_LAFM[13]	EXTID_LAFM[12]	EXTID_LAFM[11]	EXTID_LAFM[10]	EXTID_LAFM[9]	EXTID_LAFM[8]
		EXTID_LAFM[7]	EXTID_LAFM[6]	EXTID_LAFM[5]	EXTID_LAFM[4]	EXTID_LAFM[3]	EXTID_LAFM[2]	EXTID_LAFM[1]	EXTID_LAFM[0]
	MBn_DATA_01_0(n=0 ～ 31)	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0
		MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1
	MBn_DATA_23_0(n=0 ～ 31)	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2
		MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3
	MBn_DATA_45_0(n=0 ～ 31)	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4
		MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
RCAN-TL1	MBn_DATA_67_0(n=0 ～ 31)	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6
		MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7
	MBn_CONTROL1_0(n=0)	—	—	NMC	—	—	MBC[2]	MBC[1]	MBC[0]
		—	—	—	—	DLD[3]	DLD[2]	DLD[1]	DLD[0]
	MBn_CONTROL1_0 (n=1 ～ 31)	—	—	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]
		—	—	—	—	DLD[3]	DLD[2]	DLD[1]	DLD[0]
	MBn_TIMESTAMP_0 (n=0 ～ 15、30、 31)	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8
		TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
	MBn_TTT_0 (n=24 ～ 30)	TTT15	TTT14	TTT13	TTT12	TTT11	TTT10	TTT9	TTT8
		TTT7	TTT6	TTT5	TTT4	TTT3	TTT2	TTT1	TTT0
	MBn_TTCONTROL_0 (n=24 ～ 29)	TTW[1]	TTW[0]	OFFSET[5]	OFFSET[4]	OFFSET[3]	OFFSET[2]	OFFSET[1]	OFFSET[0]
		—	—	—	—	—	REP_FACTOR[2]	REP_FACTOR[1]	REP_FACTOR[0]
	MCR_1	MCR15	MCR14	—	—	—	TST2	TST1	TST0
		MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0
	GSR_1	—	—	—	—	—	—	—	—
		—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
	BCR1_1	TSEG13	TSEG12	TSEG11	TSEG10	—	TSEG22	TSEG21	TSEG20
		—	—	SJW1	SJW0	—	—	—	BSP
	BCR0_1	—	—	—	—	—	—	—	—
		BRP7	BRP6	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
	IRR_1	IRR15	IRR14	IRR13	IRR12	IRR11	IRR10	IRR9	IRR8
		IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
	IMR_1	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8
		IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
	TEC_REC_1	TEC[7]	TEC[6]	TEC[5]	TEC[4]	TEC[3]	TEC[2]	TEC[1]	TEC[0]
		REC[7]	REC[6]	REC[5]	REC[4]	REC[3]	REC[2]	REC[1]	REC[0]
	TXPR1_1	TXPR1[15]	TXPR1[14]	TXPR1[13]	TXPR1[12]	TXPR1[11]	TXPR1[10]	TXPR1[9]	TXPR1[8]
		TXPR1[7]	TXPR1[6]	TXPR1[5]	TXPR1[4]	TXPR1[3]	TXPR1[2]	TXPR1[1]	TXPR1[0]

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
RCAN-TL1	TXPR0_1	TXPR0[15]	TXPR0[14]	TXPR0[13]	TXPR0[12]	TXPR0[11]	TXPR0[10]	TXPR0[9]	TXPR0[8]
		TXPR0[7]	TXPR0[6]	TXPR0[5]	TXPR0[4]	TXPR0[3]	TXPR0[2]	TXPR0[1]	—
	TXCR1_1	TXCR1[15]	TXCR1[14]	TXCR1[13]	TXCR1[12]	TXCR1[11]	TXCR1[10]	TXCR1[9]	TXCR1[8]
		TXCR1[7]	TXCR1[6]	TXCR1[5]	TXCR1[4]	TXCR1[3]	TXCR1[2]	TXCR1[1]	TXCR1[0]
	TXCR0_1	TXCR0[15]	TXCR0[14]	TXCR0[13]	TXCR0[12]	TXCR0[11]	TXCR0[10]	TXCR0[9]	TXCR0[8]
		TXCR0[7]	TXCR0[6]	TXCR0[5]	TXCR0[4]	TXCR0[3]	TXCR0[2]	TXCR0[1]	—
	TXACK1_1	TXACK1 [15]	TXACK1 [14]	TXACK1 [13]	TXACK1 [12]	TXACK1 [11]	TXACK1 [10]	TXACK1[9]	TXACK1[8]
		TXACK1 [7]	TXACK1 [6]	TXACK1 [5]	TXACK1 [4]	TXACK1 [3]	TXACK1 [2]	TXACK1 [1]	TXACK1 [0]
	TXACK0_1	TXACK0 [15]	TXACK0 [14]	TXACK0 [13]	TXACK0 [12]	TXACK0 [11]	TXACK0 [10]	TXACK0 [9]	TXACK0 [8]
		TXACK0 [7]	TXACK0 [6]	TXACK0 [5]	TXACK0 [4]	TXACK0 [3]	TXACK0 [2]	TXACK0 [1]	—
	ABACK1_1	ABACK1 [15]	ABACK1 [14]	ABACK1 [13]	ABACK1 [12]	ABACK1 [11]	ABACK1 [10]	ABACK1 [9]	ABACK1 [8]
		ABACK1 [7]	ABACK1 [6]	ABACK1 [5]	ABACK1 [4]	ABACK1 [3]	ABACK1 [2]	ABACK1 [1]	ABACK1 [0]
	ABACK0_1	ABACK0 [15]	ABACK0 [14]	ABACK0 [13]	ABACK0 [12]	ABACK0 [11]	ABACK0 [10]	ABACK0 [9]	ABACK0 [8]
		ABACK0 [7]	ABACK0 [6]	ABACK0 [5]	ABACK0 [4]	ABACK0 [3]	ABACK0 [2]	ABACK0 [1]	—
	RXPR1_1	RXPR1[15]	RXPR1[14]	RXPR1[13]	RXPR1[12]	RXPR1[11]	RXPR1[10]	RXPR1[9]	RXPR1[8]
		RXPR1[7]	RXPR1[6]	RXPR1[5]	RXPR1[4]	RXPR1[3]	RXPR1[2]	RXPR1[1]	RXPR1[0]
	RXPR0_1	RXPR0[15]	RXPR0[14]	RXPR0[13]	RXPR0[12]	RXPR0[11]	RXPR0[10]	RXPR0[9]	RXPR0[8]
		RXPR0[7]	RXPR0[6]	RXPR0[5]	RXPR0[4]	RXPR0[3]	RXPR0[2]	RXPR0[1]	RXPR0[0]
	RFPR1_1	RFPR1[15]	RFPR1[14]	RFPR1[13]	RFPR1[12]	RFPR1[11]	RFPR1[10]	RFPR1[9]	RFPR1[8]
		RFPR1[7]	RFPR1[6]	RFPR1[5]	RFPR1[4]	RFPR1[3]	RFPR1[2]	RFPR1[1]	RFPR1[0]
	RFPR0_1	RFPR0 [15]	RFPR0 [14]	RFPR0 [13]	RFPR0 [12]	RFPR0 [11]	RFPR0 [10]	RFPR0 [9]	RFPR0 [8]
		RFPR0 [7]	RFPR0 [6]	RFPR0 [5]	RFPR0 [4]	RFPR0 [3]	RFPR0 [2]	RFPR0 [1]	RFPR0 [0]
	MBIMR1_1	MBIMR1 [15]	MBIMR1 [14]	MBIMR1 [13]	MBIMR1 [12]	MBIMR1 [11]	MBIMR1 [10]	MBIMR1[9]	MBIMR1[8]
		MBIMR1 [7]	MBIMR1 [6]	MBIMR1 [5]	MBIMR1 [4]	MBIMR1 [3]	MBIMR1 [2]	MBIMR1 [1]	MBIMR1 [0]

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
RCAN-TL1	MBIMR0_1	MBIMR0 [15]	MBIMR0 [14]	MBIMR0 [13]	MBIMR0 [12]	MBIMR0 [11]	MBIMR0 [10]	MBIMR0 [9]	MBIMR0 [8]
		MBIMR0 [7]	MBIMR0 [6]	MBIMR0 [5]	MBIMR0 [4]	MBIMR0 [3]	MBIMR0 [2]	MBIMR0 [1]	MBIMR0 [0]
	UMSR1_1	UMSR1 [15]	UMSR1 [14]	UMSR1 [13]	UMSR1 [12]	UMSR1 [11]	UMSR1 [10]	UMSR1 [9]	UMSR1 [8]
		UMSR1 [7]	UMSR1 [6]	UMSR1 [5]	UMSR1 [4]	UMSR1 [3]	UMSR1 [2]	UMSR1 [1]	UMSR1 [0]
	UMSR0_1	UMSR0 [15]	UMSR0 [14]	UMSR0 [13]	UMSR0 [12]	UMSR0 [11]	UMSR0 [10]	UMSR0 [9]	UMSR0 [8]
		UMSR0 [7]	UMSR0 [6]	UMSR0 [5]	UMSR0 [4]	UMSR0 [3]	UMSR0 [2]	UMSR0 [1]	UMSR0 [0]
	TTCR0_1	TCR[15]	TCR[14]	TCR[13]	TCR[12]	TCR[11]	TCR[10]	—	—
		—	TCR[6]	TPSC[5]	TPSC[4]	TPSC[3]	TPSC[2]	TPSC[1]	TPSC[0]
	CMAX_TEW_1	—	—	—	—	—	CMAX[2]	CMAX[1]	CMAX[0]
		—	—	—	TEW[4]	TEW[3]	TEW[2]	TEW[1]	TEW[0]
	RFTROFF_1	RTROFF[7]	RTROFF[6]	RTROFF[5]	RTROFF[4]	RTROFF[3]	RTROFF[2]	RTROFF[1]	RTROFF[0]
		—	—	—	—	—	—	—	—
	TSR_1	—	—	—	—	—	—	—	—
		—	—	—	TSR[4]	TSR[3]	TSR[2]	TSR[1]	TSR[0]
	CCR_1	—	—	—	—	—	—	—	—
		—	—	CCR[5]	CCR[4]	CCR[3]	CCR[2]	CCR[1]	CCR[0]
	TCNTR_1	TCNTR [15]	TCNTR [14]	TCNTR [13]	TCNTR [12]	TCNTR [11]	TCNTR [10]	TCNTR[9]	TCNTR[8]
		TCNTR [7]	TCNTR[6]	TCNTR [5]	TCNTR[4]	TCNTR [3]	TCNTR [2]	TCNTR[1]	TCNTR[0]
	CYCTR_1	CYCTR [15]	CYCTR [14]	CYCTR [13]	CYCTR [12]	CYCTR [11]	CYCTR [10]	CYCTR [9]	CYCTR [8]
		CYCTR [7]	CYCTR [6]	CYCTR [5]	CYCTR [4]	CYCTR [3]	CYCTR [2]	CYCTR [1]	CYCTR [0]
	RFMK_1	RFMK [15]	RFMK [14]	RFMK [13]	RFMK [12]	RFMK [11]	RFMK [10]	RFMK [9]	RFMK [8]
		RFMK[7]	RFMK[6]	RFMK[5]	RFMK[4]	RFMK[3]	RFMK[2]	RFMK[1]	RFMK[0]
	TCMR0_1	TCMR0 [15]	TCMR0 [14]	TCMR0 [13]	TCMR0 [12]	TCMR0 [11]	TCMR0 [10]	TCMR0 [9]	TCMR0 [8]
		TCMR0[7]	TCMR0[6]	TCMR0[5]	TCMR0[4]	TCMR0[3]	TCMR0[2]	TCMR0[1]	TCMR0[0]

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
RCAN-TL1	TCMR1_1	TCMR1[15]	TCMR1[14]	TCMR1[13]	TCMR1[12]	TCMR1[11]	TCMR1[10]	TCMR1[9]	TCMR1[8]
		TCMR1[7]	TCMR1[6]	TCMR1[5]	TCMR1[4]	TCMR1[3]	TCMR1[2]	TCMR1[1]	TCMR1[0]
	TCMR2_1	TCMR2[15]	TCMR2[14]	TCMR2[13]	TCMR2[12]	TCMR2[11]	TCMR2[10]	TCMR2[9]	TCMR2[8]
		TCMR2[7]	TCMR2[6]	TCMR2[5]	TCMR2[4]	TCMR2[3]	TCMR2[2]	TCMR2[1]	TCMR2[0]
	TTTSEL_1	—	TTTSEL[14]	TTTSEL[13]	TTTSEL[12]	TTTSEL[11]	TTTSEL[10]	TTTSEL[9]	TTTSEL[8]
		—	—	—	—	—	—	—	—
	MBn_CONTROL0_H_1 (n=0 ～ 31)*8	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]
		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]
	MBn_CONTROL0_H_1 (n=0 ～ 31)*9	IDE	RTR	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]
		STDID[5]	STDID[4]	STDID[3]	STDID[2]	STDID[1]	STDID[0]	EXTID[17]	EXTID[16]
	MBn_CONTROL0_L_1 (n=0 ～ 31)	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]
		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]
	MBn_LAFM0_1 (n=0 ～ 31)*8	—	STDID_LAFM[10]	STDID_LAFM[9]	STDID_LAFM[8]	STDID_LAFM[7]	STDID_LAFM[6]	STDID_LAFM[5]	STDID_LAFM[4]
		STDID_LAFM[3]	STDID_LAFM[2]	STDID_LAFM[1]	STDID_LAFM[0]	—	IDE	EXTID_LAFM[17]	EXTID_LAFM[16]
	MBn_LAFM0_1 (n=0 ～ 31)*9	IDE	—	—	STDID_LAFM[10]	STDID_LAFM[9]	STDID_LAFM[8]	STDID_LAFM[7]	STDID_LAFM[6]
		STDID_LAFM[5]	STDID_LAFM[4]	STDID_LAFM[3]	STDID_LAFM[2]	STDID_LAFM[1]	STDID_LAFM[0]	EXTID_LAFM[17]	EXTID_LAFM[16]
	MBn_LAFM1_1 (n=0 ～ 31)	EXTID_LAFM[15]	EXTID_LAFM[14]	EXTID_LAFM[13]	EXTID_LAFM[12]	EXTID_LAFM[11]	EXTID_LAFM[10]	EXTID_LAFM[9]	EXTID_LAFM[8]
		EXTID_LAFM[7]	EXTID_LAFM[6]	EXTID_LAFM[5]	EXTID_LAFM[4]	EXTID_LAFM[3]	EXTID_LAFM[2]	EXTID_LAFM[1]	EXTID_LAFM[0]
	MBn_DATA_01_1(n=0 ～ 31)	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0	MSG_DATA0
		MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1	MSG_DATA1
	MBn_DATA_23_1(n=0 ～ 31)	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2	MSG_DATA2
		MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3	MSG_DATA3

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
RCAN-TL1	MBn_DATA_45_1(n=0 ～ 31)	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4	MSG_DATA4
		MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5	MSG_DATA5
	MBn_DATA_67_1(n=0 ～ 31)	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6	MSG_DATA6
		MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7	MSG_DATA7
	MBn_CONTROL1_1(n=0)	—	—	NMC	—	—	MBC[2]	MBC[1]	MBC[0]
		—	—	—	—	DLD[3]	DLD[2]	DLD[1]	DLD[0]
	MBn_CONTROL1_1(n=1 ～ 31)	—	—	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]
		—	—	—	—	DLD[3]	DLD[2]	DLD[1]	DLD[0]
	MBn_TIMESTAMP_1(n=0 ～ 15、 30、 31)	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8
		TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
	MBn_TTT_1(n=24 ～ 30)	TTT15	TTT14	TTT13	TTT12	TTT11	TTT10	TTT9	TTT8
		TTT7	TTT6	TTT5	TTT4	TTT3	TTT2	TTT1	TTT0
	MBn_TTCONTROL_1(n=24 ～ 29)	TTW[1]	TTW[0]	OFFSET[5]	OFFSET[4]	OFFSET[3]	OFFSET[2]	OFFSET[1]	OFFSET[0]
		—	—	—	—	—	REP_FACTOR[2]	REP_FACTOR[1]	REP_FACTOR[0]
IEB	IECTR	—	IOL	DEE	—	RE	—	—	—
	IECMR	—	—	—	—	—	CMD[2]	CMD[1]	CMD[0]
	IEMCR	SS	RN[2]	RN[1]	RN[0]	CTL[3]	CTL[2]	CTL[1]	CTL[0]
	IEAR1	IARL4[3]	IARL4[2]	IARL4[1]	IARL4[0]	IMD[1]	IMD[0]	—	STE
	IEAR2	IARU8[7]	IARU8[6]	IARU8[5]	IARU8[4]	IARU8[3]	IARU8[2]	IARU8[1]	IARU8[0]
	IESA1	ISAL4[3]	ISAL4[2]	ISAL4[1]	ISAL4[0]	—	—	—	—
	IESA2	ISAU8[7]	ISAU8[6]	ISAU8[5]	ISAU8[4]	ISAU8[3]	ISAU8[2]	ISAU8[1]	ISAU8[0]
	IETBFL	IBFL[7]	IBFL[6]	IBFL[5]	IBFL[4]	IBFL[3]	IBFL[2]	IBFL[1]	IBFL[0]
	IEMA1	IMAL4[3]	IMAL4[2]	IMAL4[1]	IMAL4[0]	—	—	—	—
	IEMA2	IMAU8[7]	IMAU8[6]	IMAU8[5]	IMAU8[4]	IMAU8[3]	IMAU8[2]	IMAU8[1]	IMAU8[0]
	IERCTL	—	—	—	—	RCTL[3]	RCTL[2]	RCTL[1]	RCTL[0]
	IERBFL	RBFL[7]	RBFL[6]	RBFL[5]	RBFL[4]	RBFL[3]	RBFL[2]	RBFL[1]	RBFL[0]
	IELA1	ILAL8[7]	ILAL8[6]	ILAL8[5]	ILAL8[4]	ILAL8[3]	ILAL8[2]	ILAL8[1]	ILAL8[0]
	IELA2	—	—	—	—	ILAU4[3]	ILAU4[2]	ILAU4[1]	ILAU4[0]
	IEFLG	CMX	MRQ	SRQ	SRE	LCK	—	RSS	GG

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
IEB	IETSR	—	TXS	TXF	—	TXEAL	TXETTME	TXERO	TXEACK
	IEIET	—	TXSE	TXFE	—	TXEALE	TXETTMEE	TXEROE	TXEACKE
	IERSR	RXBSY	RXS	RXF	RXEDE	RXEOVE	RXERTME	RXEDLE	RXEPE
	IEIER	RXBSYE	RXSE	RXFE	RXEDEE	RXEOVEE	RXERTMEE	RXEDLEE	RXEPEE
	IECKSR	—	—	—	CKS3	—	CKS[2]	CKS[1]	CKS[0]
	IETB001 ～ IETB128								
	IERB001 ～ IERB128								
ROM-DEC	CROMEN	SUBC_EN	CROM_EN	CROM_STP	—	—	—	—	—
	CROMSY0	SY_AUT	SY_IEN	SY_DEN	—	—	—	—	—
	CROMCTL0	MD_DESC	—	MD_AUTO	MD_AUTOS1	MD_AUTOS2	MD_SEC[2]	MD_SEC[1]	MD_SEC[0]
	CROMCTL1	M2F2EDC	MD_DEC[2]	MD_DEC[1]	MD_DEC[0]	—	—	MD_PQREP[1]	MD_PQREP[0]
	CROMCTL3	STP_ECC	STP_EDC	—	STP_MD	STP_MIN	—	—	—
	CROMCTL4	LINKOFF	LINK2	—	EROSEL	NO_ECC	—	—	—
	CROMCTL5	—	—	—	—	—	—		MSF_LBA_SEL
	CROMST0	—	—	ST_SYIL	ST_SYNO	ST_BLKS	ST_BLKL	ST_SECS	ST_SECL
	CROMST1	—	—	—	—	ER2_HEAD0	ER2_HEAD1	ER2_HEAD2	ER2_HEAD3
	CROMST3	ER2_SHEAD0	ER2_SHEAD1	ER2_SHEAD2	ER2_SHEAD3	ER2_SHEAD4	ER2_SHEAD5	ER2_SHEAD6	ER2_SHEAD7
	CROMST4	NG_MD	NG_MDCMP1	NG_MDCMP2	NG_MDCMP3	NG_MDCMP4	NG_MDDEF	NG_MDTIM1	NG_MDTIM2
	CROMST5	ST_AMD[2]	ST_AMD[1]	ST_AMD[0]	ST_MDX	LINK_ON	LINK_DET	LINK_SDET	LINK_OUT1
	CROMST6	ST_ERR	—	ST_ECCABT	ST_ECCNG	ST_ECCP	ST_ECCQ	ST_EDC1	ST_EDC2
	CBUFST0	BUF_REF	BUF_ACT	—	—	—	—	—	—
	CBUFST1	BUF_ECC	BUF_EDC	—	BUF_MD	BUF_MIN	—	—	—
	CBUFST2	BUF_NG	—	—	—	—	—	—	—
	HEAD00	HEAD00[7]	HEAD00[6]	HEAD00[5]	HEAD00[4]	HEAD00[3]	HEAD00[2]	HEAD00[1]	HEAD00[0]
	HEAD01	HEAD01[7]	HEAD01[6]	HEAD01[5]	HEAD01[4]	HEAD01[3]	HEAD01[2]	HEAD01[1]	HEAD01[0]

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
ROM-DEC	HEAD02	HEAD02 [7]	HEAD02 [6]	HEAD02 [5]	HEAD02 [4]	HEAD02 [3]	HEAD02 [2]	HEAD02 [1]	HEAD02 [0]
	HEAD03	HEAD03 [7]	HEAD03 [6]	HEAD03 [5]	HEAD03 [4]	HEAD03 [3]	HEAD03 [2]	HEAD03 [1]	HEAD03 [0]
	SHEAD00	SHEAD00 [7]	SHEAD00 [6]	SHEAD00 [5]	SHEAD00 [4]	SHEAD00 [3]	SHEAD00 [2]	SHEAD00 [1]	SHEAD00 [0]
	SHEAD01	SHEAD01 [7]	SHEAD01 [6]	SHEAD01 [5]	SHEAD01 [4]	SHEAD01 [3]	SHEAD01 [2]	SHEAD01 [1]	SHEAD01 [0]
	SHEAD02	SHEAD02 [7]	SHEAD02 [6]	SHEAD02 [5]	SHEAD02 [4]	SHEAD02 [3]	SHEAD02 [2]	SHEAD02 [1]	SHEAD02 [0]
	SHEAD03	SHEAD03 [7]	SHEAD03 [6]	SHEAD03 [5]	SHEAD03 [4]	SHEAD03 [3]	SHEAD03 [2]	SHEAD03 [1]	SHEAD03 [0]
	SHEAD04	SHEAD04 [7]	SHEAD04 [6]	SHEAD04 [5]	SHEAD04 [4]	SHEAD04 [3]	SHEAD04 [2]	SHEAD04 [1]	SHEAD04 [0]
	SHEAD05	SHEAD05 [7]	SHEAD05 [6]	SHEAD05 [5]	SHEAD05 [4]	SHEAD05 [3]	SHEAD05 [2]	SHEAD05 [1]	SHEAD05 [0]
	SHEAD06	SHEAD06 [7]	SHEAD06 [6]	SHEAD06 [5]	SHEAD06 [4]	SHEAD06 [3]	SHEAD06 [2]	SHEAD06 [1]	SHEAD06 [0]
	SHEAD07	SHEAD07 [7]	SHEAD07 [6]	SHEAD07 [5]	SHEAD07 [4]	SHEAD07 [3]	SHEAD07 [2]	SHEAD07 [1]	SHEAD07 [0]
	HEAD20	HEAD20 [7]	HEAD20 [6]	HEAD20 [5]	HEAD20 [4]	HEAD20 [3]	HEAD20[2]	HEAD20 [1]	HEAD20 [0]
	HEAD21	HEAD21 [7]	HEAD21 [6]	HEAD21 [5]	HEAD21 [4]	HEAD21 [3]	HEAD21[2]	HEAD21 [1]	HEAD21 [0]
	HEAD22	HEAD22 [7]	HEAD22 [6]	HEAD22 [5]	HEAD22 [4]	HEAD22 [3]	HEAD22[2]	HEAD22 [1]	HEAD22 [0]
	HEAD23	HEAD23 [7]	HEAD23 [6]	HEAD23 [5]	HEAD23 [4]	HEAD23 [3]	HEAD23[2]	HEAD23 [1]	HEAD23 [0]
	SHEAD20	SHEAD20 [7]	SHEAD20 [6]	SHEAD20 [5]	SHEAD20 [4]	SHEAD20 [3]	SHEAD20 [2]	SHEAD20 [1]	SHEAD20 [0]
	SHEAD21	SHEAD21 [7]	SHEAD21 [6]	SHEAD21 [5]	SHEAD21 [4]	SHEAD21 [3]	SHEAD21 [2]	SHEAD21 [1]	SHEAD21 [0]
	SHEAD22	SHEAD22 [7]	SHEAD22 [6]	SHEAD22 [5]	SHEAD22 [4]	SHEAD22 [3]	SHEAD22 [2]	SHEAD22 [1]	SHEAD22 [0]
	SHEAD23	SHEAD23 [7]	SHEAD23 [6]	SHEAD23 [5]	SHEAD23 [4]	SHEAD23 [3]	SHEAD23 [2]	SHEAD23 [1]	SHEAD23 [0]
	SHEAD24	SHEAD24 [7]	SHEAD24 [6]	SHEAD24 [5]	SHEAD24 [4]	SHEAD24 [3]	SHEAD24 [2]	SHEAD24 [1]	SHEAD24 [0]
	SHEAD25	SHEAD25 [7]	SHEAD25 [6]	SHEAD25 [5]	SHEAD25 [4]	SHEAD25 [3]	SHEAD25 [2]	SHEAD25 [1]	SHEAD25 [0]

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
ROM-DEC	SHEAD26	SHEAD26[7]	SHEAD26[6]	SHEAD26[5]	SHEAD26[4]	SHEAD26[3]	SHEAD26[2]	SHEAD26[1]	SHEAD26[0]
	SHEAD27	SHEAD27[7]	SHEAD27[6]	SHEAD27[5]	SHEAD27[4]	SHEAD27[3]	SHEAD27[2]	SHEAD27[1]	SHEAD27[0]
	CBUFCTL0	CBUF_AUT	CBUF_EN	CBUF_LINK	CBUF_MD[1]	CBUF_MD[0]	CBUF_TS	CBUF_Q	—
	CBUFCTL1	BS_MIN[7]	BS_MIN[6]	BS_MIN[5]	BS_MIN[4]	BS_MIN[3]	BS_MIN[2]	BS_MIN[1]	BS_MIN[0]
	CBUFCTL2	BS_SEC[7]	BS_SEC[6]	BS_SEC[5]	BS_SEC[4]	BS_SEC[3]	BS_SEC[2]	BS_SEC[1]	BS_SEC[0]
	CBUFCTL3	BS_FRM[7]	BS_FRM[6]	BS_FRM[5]	BS_FRM[4]	BS_FRM[3]	BS_FRM[2]	BS_FRM[1]	BS_FRM[0]
	CROMST0M	—	—	ST_SYILM	ST_SYNOM	ST_BLKSM	ST_BLKLM	ST_SECSM	ST_SECLM
	ROMDECRST	LOGICRST	RAMRST	—	—	—	—	—	—
	RSTSTAT	RAMCLRST	—	—	—	—	—	—	—
	SSI	BYTEND	BITEND	BUFEND0[1]	BUFEND0[0]	BUFEND1[1]	BUFEND1[0]	—	—
	INTHOLD	ISEC	ITARG	ISY	IERR	IBUF	IREADY	—	—
	INHINT	INHISEC	INHITARG	INHISY	INHIERR	INHIBUF	INHI READY	PREINH REQDM	PREINHI READY
	STRMDIN0	STRMDIN[31]	STRMDIN[30]	STRMDIN[29]	STRMDIN[28]	STRMDIN[27]	STRMDIN[26]	STRMDIN[25]	STRMDIN[24]
		STRMDIN[23]	STRMDIN[22]	STRMDIN[21]	STRMDIN[20]	STRMDIN[19]	STRMDIN[18]	STRMDIN[17]	STRMDIN[16]
	STRMDIN2	STRMDIN[15]	STRMDIN[14]	STRMDIN[13]	STRMDIN[12]	STRMDIN[11]	STRMDIN[10]	STRMDIN[9]	STRMDIN[8]
		STRMDIN[7]	STRMDIN[6]	STRMDIN[5]	STRMDIN[4]	STRMDIN[3]	STRMDIN[2]	STRMDIN[1]	STRMDIN[0]
	STRMDOUT0	STRMDOUT[15]	STRMDOUT[14]	STRMDOUT[13]	STRMDOUT[12]	STRMDOUT[11]	STRMDOUT[10]	STRMDOUT[9]	STRMDOUT[8]
		STRMDOUT[7]	STRMDOUT[6]	STRMDOUT[5]	STRMDOUT[4]	STRMDOUT[3]	STRMDOUT[2]	STRMDOUT[1]	STRMDOUT[0]

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
ADC	ADDRA								
				—	—	—	—	—	—
	ADDRB								
				—	—	—	—	—	—
	ADDRC								
				—	—	—	—	—	—
	ADDRD								
				—	—	—	—	—	—
	ADDRE								
				—	—	—	—	—	—
	ADDRF								
				—	—	—	—	—	—
	ADDRG								
				—	—	—	—	—	—
	ADDRH								
				—	—	—	—	—	—
	ADCSR	ADF	ADIE	ADST	—	TRGS[3]	TRGS[2]	TRGS[1]	TRGS[0]
		CKS[1]	CKS[0]	MDS[2]	MDS[1]	MDS[0]	CH[2]	CH[1]	CH[0]
DAC	DADR0								
	DADR1								
	DACR	DAOE1	DAOE0	DAE	—	—	—	—	—
FLCTL	FLCMNCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	SNAND	QTSEL	—
		FCKSEL	—	ECCPOS[1]	ECCPOS[0]	ACM[1]	ACM[0]	NANDWF	—
		—	—	—	—	CE	—	—	TYPESEL
	FLCMDCR	ADRCNT2	SCTCNT[19]	SCTCNT[18]	SCTCNT[17]	SCTCNT[16]	ADRMD	CDSRC	DOSR
		—	—	SELRW	DOADR	ADRCNT[1]	ADRCNT[0]	DOCMD2	DOCMD1
		SCTCNT[15]	SCTCNT[14]	SCTCNT[13]	SCTCNT[12]	SCTCNT[11]	SCTCNT[10]	SCTCNT[9]	SCTCNT[8]
		SCTCNT[7]	SCTCNT[6]	SCTCNT[5]	SCTCNT[4]	SCTCNT[3]	SCTCNT[2]	SCTCNT[1]	SCTCNT[0]

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
FLCTL	FLCMCDR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		CMD2[7]	CMD2[6]	CMD2[5]	CMD2[4]	CMD2[3]	CMD2[2]	CMD2[1]	CMD2[0]
		CMD1[7]	CMD1[6]	CMD1[5]	CMD1[4]	CMD1[3]	CMD1[2]	CMD1[1]	CMD1[0]
	FLADR*10	ADR4[7]	ADR4[6]	ADR4[5]	ADR4[4]	ADR4[3]	ADR4[2]	ADR4[1]	ADR4[0]
		ADR3[7]	ADR3[6]	ADR3[5]	ADR3[4]	ADR3[3]	ADR3[2]	ADR3[1]	ADR3[0]
		ADR2[7]	ADR2[6]	ADR2[5]	ADR2[4]	ADR2[3]	ADR2[2]	ADR2[1]	ADR2[0]
		ADR1[7]	ADR1[6]	ADR1[5]	ADR1[4]	ADR1[3]	ADR1[2]	ADR1[1]	ADR1[0]
	FLADR*11	—	—	—	—	—	—	ADR[25]	ADR[24]
		ADR[23]	ADR[22]	ADR[21]	ADR[20]	ADR[19]	ADR[18]	ADR[17]	ADR[16]
		ADR[15]	ADR[14]	ADR[13]	ADR[12]	ADR[11]	ADR[10]	ADR[9]	ADR[8]
		ADR[7]	ADR[6]	ADR[5]	ADR[4]	ADR[3]	ADR[2]	ADR[1]	ADR[0]
	FLADR2	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		ADR5[7]	ADR5[6]	ADR5[5]	ADR5[4]	ADR5[3]	ADR5[2]	ADR5[1]	ADR5[0]
	FLDTCNTR	ECFLW[7]	ECFLW[6]	ECFLW[5]	ECFLW[4]	ECFLW[3]	ECFLW[2]	ECFLW[1]	ECFLW[0]
		DTFLW[7]	DTFLW[6]	DTFLW[5]	DTFLW[4]	DTFLW[3]	DTFLW[2]	DTFLW[1]	DTFLW[0]
		—	—	—	—	DTCNT[11]	DTCNT[10]	DTCNT[9]	DTCNT[8]
		DTCNT[7]	DTCNT[6]	DTCNT[5]	DTCNT[4]	DTCNT[3]	DTCNT[2]	DTCNT[1]	DTCNT[0]
	FLDATAR	DT4[7]	DT4[6]	DT4[5]	DT4[4]	DT4[3]	DT4[2]	DT4[1]	DT4[0]
		DT3[7]	DT3[6]	DT3[5]	DT3[4]	DT3[3]	DT3[2]	DT3[1]	DT3[0]
		DT2[7]	DT2[6]	DT2[5]	DT2[4]	DT2[3]	DT2[2]	DT2[1]	DT2[0]
		DT1[7]	DT1[6]	DT1[5]	DT1[4]	DT1[3]	DT1[2]	DT1[1]	DT1[0]
	FLINTDMACR	—	—	—	—	—	—	—	ECERINTE
		—	—	FIFOTRG[1]	FIFOTRG[0]	AC1CLR	AC0CLR	DREQ1EN	DREQ0EN
		—	—	—	—	—	—	ECERB	STERB
		BTOERB	TRREQF1	TRREQF0	STERINTE	RBERINTE	TEINTE	TRINTE1	TRINTE0
	FLBSYTMR	—	—	—	—	—	—	—	—
		—	—	—	—	RBTMOUT[19]	RBTMOUT[18]	RBTMOUT[17]	RBTMOUT[16]
		RBTMOUT[15]	RBTMOUT[14]	RBTMOUT[13]	RBTMOUT[12]	RBTMOUT[11]	RBTMOUT[10]	RBTMOUT[9]	RBTMOUT[8]
		RBTMOUT[7]	RBTMOUT[6]	RBTMOUT[5]	RBTMOUT[4]	RBTMOUT[3]	RBTMOUT[2]	RBTMOUT[1]	RBTMOUT[0]

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
FLCTL	FLBSYCNT	STAT[7]	STAT[6]	STAT[5]	STAT[4]	STAT[3]	STAT[2]	STAT[1]	STAT[0]
		—	—	—	—	RBTIMCNT[19]	RBTIMCNT[18]	RBTIMCNT[17]	RBTIMCNT[16]
		RBTIMCNT[15]	RBTIMCNT[14]	RBTIMCNT[13]	RBTIMCNT[12]	RBTIMCNT[11]	RBTIMCNT[10]	RBTIMCNT[9]	RBTIMCNT[8]
		RBTIMCNT[7]	RBTIMCNT[6]	RBTIMCNT[5]	RBTIMCNT[4]	RBTIMCNT[3]	RBTIMCNT[2]	RBTIMCNT[1]	RBTIMCNT[0]
	FLDTFIFO	DTFO[31]	DTFO[30]	DTFO[29]	DTFO[28]	DTFO[27]	DTFO[26]	DTFO[25]	DTFO[24]
		DTFO[23]	DTFO[22]	DTFO[21]	DTFO[20]	DTFO[19]	DTFO[18]	DTFO[17]	DTFO[16]
		DTFO[15]	DTFO[14]	DTFO[13]	DTFO[12]	DTFO[11]	DTFO[10]	DTFO[9]	DTFO[8]
		DTFO[7]	DTFO[6]	DTFO[5]	DTFO[4]	DTFO[3]	DTFO[2]	DTFO[1]	DTFO[0]
	FLECFIFO	ECFO[31]	ECFO[30]	ECFO[29]	ECFO[28]	ECFO[27]	ECFO[26]	ECFO[25]	ECFO[24]
		ECFO[23]	ECFO[22]	ECFO[21]	ECFO[20]	ECFO[19]	ECFO[18]	ECFO[17]	ECFO[16]
		ECFO[15]	ECFO[14]	ECFO[13]	ECFO[12]	ECFO[11]	ECFO[10]	ECFO[9]	ECFO[8]
		ECFO[7]	ECFO[6]	ECFO[5]	ECFO[4]	ECFO[3]	ECFO[2]	ECFO[1]	ECFO[0]
	FLTRCR	—	—	—	—	—	—	TREND	TRSTRT
USB	SYSCFG	—	—	—	—	—	—	—	—
		HSE	DCFM	DMRPD	DPRPU	—	FSRPC	—	USBE
	SYSSTS	—	—	—	—	—	—	—	—
		—	—	SOFEN	—	—	—	LNST[1]	LNST[0]
	DVSTCTR	UACKEY0	—	—	UACKEY1	—	—	—	WKUP
		RWUPE	USBRST	RESUME	UACT	—	—	RHST[1]	RHST[0]
	TESTMODE	HOSTPCC	—	—	—	—	—	—	—
		—	—	—	—	UTST[3]	UTST[2]	UTST[1]	UTST[0]
	CFBCFG	—	—	—	—	—	—	—	—
		—	—	—	—	FWAIT[3]	FWAIT[2]	FWAIT[1]	FWAIT[0]
	D0FBCFG	—	—	—	—	—	—	TENDE	FEND
		—	—	—	—	FWAIT[3]	FWAIT[2]	FWAIT[1]	FWAIT[0]
	D1FBCFG	—	—	—	—	—	—	TENDE	FEND
		—	—	—	—	FWAIT[3]	FWAIT[2]	FWAIT[1]	FWAIT[0]

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
USB	CFIFO	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D0FIFO	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	D1FIFO	FIFOPORT[31]	FIFOPORT[30]	FIFOPORT[29]	FIFOPORT[28]	FIFOPORT[27]	FIFOPORT[26]	FIFOPORT[25]	FIFOPORT[24]
		FIFOPORT[23]	FIFOPORT[22]	FIFOPORT[21]	FIFOPORT[20]	FIFOPORT[19]	FIFOPORT[18]	FIFOPORT[17]	FIFOPORT[16]
		FIFOPORT[15]	FIFOPORT[14]	FIFOPORT[13]	FIFOPORT[12]	FIFOPORT[11]	FIFOPORT[10]	FIFOPORT[9]	FIFOPORT[8]
		FIFOPORT[7]	FIFOPORT[6]	FIFOPORT[5]	FIFOPORT[4]	FIFOPORT[3]	FIFOPORT[2]	FIFOPORT[1]	FIFOPORT[0]
	CFIFOSEL	RCNT	REW	—	—	MBW[1]	MBW[0]	—	—
		—	—	ISEL	—	—	CURPIPE[2]	CURPIPE[1]	CURPIPE[0]
	CFIFOCTR	BVAL	BCLR	FRDY	—	DTLN[11]	DTLN[10]	DTLN[9]	DTLN[8]
		DTLN[7]	DTLN[6]	DTLN[5]	DTLN[4]	DTLN[3]	DTLN[2]	DTLN[1]	DTLN[0]
	CFIFOSIE	TGL	SCLR	SBUSY	—	—	—	—	—
		—	—	—	—	—	—	—	—
	D0FIFOSEL	RCNT	REW	DCLRM	DREQE	MBW[1]	MBW[0]	TRENB	TRCLR
		DEZPM	—	—	—	—	CURPIPE[2]	CURPIPE[1]	CURPIPE[0]
	D0FIFOCTR	BVAL	BCLR	FRDY	—	DTLN[11]	DTLN[10]	DTLN[9]	DTLN[8]
		DTLN[7]	DTLN[6]	DTLN[5]	DTLN[4]	DTLN[3]	DTLN[2]	DTLN[1]	DTLN[0]

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
USB	D0FIFOTRN	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	D1FIFOSEL	RCNT	REW	DCLRM	DREQE	MBW[1]	MBW[0]	TRENB	TRCLR
		DEZPM	—	—	—	—	CURPIPE[2]	CURPIPE[1]	CURPIPE[0]
	D1FIFOCTR	BVAL	BCLR	FRDY	—	DTLN[11]	DTLN[10]	DTLN[9]	DTLN[8]
		DTLN[7]	DTLN[6]	DTLN[5]	DTLN[4]	DTLN[3]	DTLN[2]	DTLN[1]	DTLN[0]
	D1FIFOTRN	TRNCNT[15]	TRNCNT[14]	TRNCNT[13]	TRNCNT[12]	TRNCNT[11]	TRNCNT[10]	TRNCNT[9]	TRNCNT[8]
		TRNCNT[7]	TRNCNT[6]	TRNCNT[5]	TRNCNT[4]	TRNCNT[3]	TRNCNT[2]	TRNCNT[1]	TRNCNT[0]
	INTENB0	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE
		URST	SADR	SCFG	SUSP	WDST	RDST	CMPL	SERR
	INTENB1	—	BCHGE	—	DTCHE	—	—	—	—
		—	—	SIGNE	SACKE	—	BRDYM	—	—
	BRDYENB	—	—	—	—	—	—	—	—
		PIPE7BRDYE	PIPE6BRDYE	PIPE5BRDYE	PIPE4BRDYE	PIPE3BRDYE	PIPE2BRDYE	PIPE1BRDYE	PIPE0BRDYE
	NRDYENB	—	—	—	—	—	—	—	—
		PIPE7NRDYE	PIPE6NRDYE	PIPE5NRDYE	PIPE4NRDYE	PIPE3NRDYE	PIPE2NRDYE	PIPE1NRDYE	PIPE0NRDYE
	BEMPENB	—	—	—	—	—	—	—	—
		PIPE7BEMPE	PIPE6BEMPE	PIPE5BEMPE	PIPE4BEMPE	PIPE3BEMPE	PIPE2BEMPE	PIPE1BEMPE	PIPE0BEMPE
	INTSTS0	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY
		VBSTS	DVSQ[2]	DVSQ[1]	DVSQ[0]	VALID	CTSQ[2]	CTSQ[1]	CTSQ[0]
	INTSTS1	—	BCHG	SOFR	DTCH	—	BEMP	NRDY	BRDY
		—	—	SIGN	SACK	—	—	—	—

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
USB	BRDYSTS	—	—	—	—	—	—	—	—
		PIPE7 BRDY	PIPE6 BRDY	PIPE5 BRDY	PIPE4 BRDY	PIPE3 BRDY	PIPE2 BRDY	PIPE1 BRDY	PIPE0 BRDY
	NRDYSTS	—	—	—	—	—	—	—	—
		PIPE7 NRDY	PIPE6 NRDY	PIPE5 NRDY	PIPE4 NRDY	PIPE3 NRDY	PIPE2 NRDY	PIPE1 NRDY	PIPE0 NRDY
	BEMPSTS	—	—	—	—	—	—	—	—
		PIPE7 BEMP	PIPE6 BEMP	PIPE5 BEMP	PIPE4 BEMP	PIPE3 BEMP	PIPE2 BEMP	PIPE1 BEMP	PIPE0 BEMP
	FRMNUM	OVRN	CRCE	—	—	SOFRM	FRNM[10]	FRNM[9]	FRNM[8]
		FRNM[7]	FRNM[6]	FRNM[5]	FRNM[4]	FRNM[3]	FRNM[2]	FRNM[1]	FRNM[0]
	UFRMNUM	—	—	—	—	—	—	—	—
		—	—	—	—	—	UFRNM[2]	UFRNM[1]	UFRNM[0]
	USBADDR	—	—	—	—	—	—	—	—
		—	USBADDR [6]	USBADDR [5]	USBADDR [4]	USBADDR [3]	USBADDR [2]	USBADDR [1]	USBADDR [0]
	USBREQ	BREQUEST [7]	BREQUEST [6]	BREQUEST [5]	BREQUEST [4]	BREQUEST [3]	BREQUEST [2]	BREQUEST [1]	BREQUEST [0]
		BMREQUESTTYPE[7]	BMREQUESTTYPE[6]	BMREQUESTTYPE[5]	BMREQUESTTYPE[4]	BMREQUESTTYPE[3]	BMREQUESTTYPE[2]	BMREQUESTTYPE[1]	BMREQUESTTYPE[0]
	USBVAL	WVALUE [15]	WVALUE [14]	WVALUE[13]	WVALUE [12]	WVALUE [11]	WVALUE [10]	WVALUE [9]	WVALUE [8]
		WVALUE [7]	WVALUE [6]	WVALUE [5]	WVALUE [4]	WVALUE [3]	WVALUE [2]	WVALUE [1]	WVALUE [0]
	USBINDX	WINDEX [15]	WINDEX [14]	WINDEX [13]	WINDEX [12]	WINDEX [11]	WINDEX [10]	WINDEX [9]	WINDEX [8]
		WINDEX [7]	WINDEX [6]	WINDEX [5]	WINDEX [4]	WINDEX [3]	WINDEX [2]	WINDEX [1]	WINDEX [0]
	USBLENG	WLENGTH [15]	WLENGTH [14]	WLENGTH [13]	WLENGTH [12]	WLENGTH [11]	WLENGTH [10]	WLENGTH [9]	WLENGTH [8]
		WLENGTH [7]	WLENGTH [6]	WLENGTH [5]	WLENGTH [4]	WLENGTH [3]	WLENGTH [2]	WLENGTH [1]	WLENGTH [0]
	DCPCFG	—	—	—	—	—	—	—	CNTMD
		SHTNAK	—	—	DIR	—	—	—	—
	DCPMAXP	DEVSEL[1]	DEVSEL[0]	—	—	—	—	—	—
		—	MXPS[6]	MXPS[5]	MXPS[4]	MXPS[3]	MXPS[2]	MXPS[1]	MXPS[0]
	DCPCTR	BSTS	SUREQ	—	—	—	—	—	SQCLR
		SQSET	SQMON	—	—	—	CCPL	PID[1]	PID[0]

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
USB	PIPESEL	—	—	—	—	—	—	—	—
		—	—	—	—	—	PIPESEL[2]	PIPESEL[1]	PIPESEL[0]
	PIPECFG	TYPE[1]	TYPE[0]	—	—	—	BFRE	DBLB	CNTMD
		SHTNAK	—	—	DIR	EPNUM[3]	EPNUM[2]	EPNUM[1]	EPNUM[0]
	PIPEBUF	—	BUFSIZE[4]	BUFSIZE[3]	BUFSIZE[2]	BUFSIZE[1]	BUFSIZE[0]	—	—
		—	BUFNMB[6]	BUFNMB[5]	BUFNMB[4]	BUFNMB[3]	BUFNMB[2]	BUFNMB[1]	BUFNMB[0]
	PIPEMAXP	DEVSEL[1]	DEVSEL[0]	—	—	—	MXPS[10]	MXPS[9]	MXPS[8]
		MXPS[7]	MXPS[6]	MXPS[5]	MXPS[4]	MXPS[3]	MXPS[2]	MXPS[1]	MXPS[0]
	PIPEPERI	—	—	—	IFIS	—	—	—	—
		—	—	—	—	—	IITV[2]	IITV[1]	IITV[0]
	PIPE1CTR	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	—	—	—	—	PID[1]	PID[0]
	PIPE2CTR	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	—	—	—	—	PID[1]	PID[0]
	PIPE3CTR	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	—	—	—	—	PID[1]	PID[0]
	PIPE4CTR	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	—	—	—	—	PID[1]	PID[0]
	PIPE5CTR	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR
		SQSET	SQMON	—	—	—	—	PID[1]	PID[0]
	PIPE6CTR	BSTS	INBUFM	—	—	—	—	ACLRM	SQCLR
		SQSET	SQMON	—	—	—	—	PID[1]	PID[0]
	PIPE7CTR	BSTS	INBUFM	—	—	—	—	ACLRM	SQCLR
		SQSET	SQMON	—	—	—	—	PID[1]	PID[0]
	USBACSWR	—	—	—	—	—	—	—	—
		UACS23	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—
		—	—	—	—	—	—	—	—

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
LCDC	LDICKR	—	—	ICKSEL1	ICKSEL0	—	—	—	—
		—	—	DCDR5	DCDR4	DCRD3	DCRD2	DCRD1	DCDR0
	LDMTR	FLMPOL	CL1POL	DISPPOL	DPOL	—	MCNT	CL1CNT	CL2CNT
		—	—	MIFTYP5	MIFTYP4	MIFTYP3	MIFTYP2	MIFTYP1	MIFTYP0
	LDDFR	—	—	—	—	—	—	—	PABD
		—	DSPCOLOR6	DSPCOLOR5	DSPCOLOR4	DSPCOLOR3	DSPCOLOR2	DSPCOLOR1	DSPCOLOR0
	LDSMR	—	—	ROT	—	—	—	AU1	AU0
		—	—	—	—	—	—	—	—
	LDSARU	—	—	—	—	—	—	SAU25	SAU24
		SAU23	SAU22	SAU21	SAU20	SAU19	SAU18	SAU17	SAU16
		SAU15	SAU14	SAU13	SAU12	SAU11	SAU10	SAU9	SAU8
		SAU7	SAU6	SAU5	SAU4	—	—	—	—
	LDSARL	—	—	—	—	—	—	SAL25	SAL24
		SAL23	SAL22	SAL21	SAL20	SAL19	SAL18	SAL17	SAL16
		SAL15	SAL14	SAL13	SAL12	SAL11	SAL10	SAL9	SAL8
		SAL7	SAL6	SAL5	SAL4	—	—	—	—
	LDLAOR	LAO15	LAO14	LAO13	LAO12	LAO11	LAO10	LAO9	LAO8
		LAO7	LAO6	LAO5	LAO4	LAO3	LAO2	LAO1	LAO0
	LDPALCR	—	—	—	—	—	—	—	—
		—	—	—	PALS	—	—	—	PALEN
	LDPRnn（nn=00 ～ FF）	—	—	—	—	—	—	—	—
		PALDnn23	PALDnn22	PALDnn21	PALDnn20	PALDnn19	PALDnn18	PALDnn17	PALDnn16
		PALDnn15	PALDnn14	PALDnn13	PALDnn12	PALDnn11	PALDnn10	PALDnn9	PALDnn8
		PALDnn7	PALDnn6	PALDnn5	PALDnn4	PALDnn3	PALDnn2	PALDnn1	PALDnn0
	LDHCNR	HDCN7	HDCN6	HDCN5	HDCN4	HDCN3	HDCN2	HDCN1	HDCN0
		HTCN7	HTCN6	HTCN5	HTCN4	HTCN3	HTCN2	HTCN1	HTCN0
	LDHSYNR	HSYNW3	HSYNW2	HSYNW1	HSYNW0	—	—	—	—
		HSYNP7	HSYNP6	HSYNP5	HSYNP4	HSYNP3	HSYNP2	HSYNP1	HSYNP0
	LDVDLNR	—	—	—	—	—	VDLN10	VDLN9	VDLN8
		VDLN7	VDLN6	VDLN5	VDLN4	VDLN3	VDLN2	VDLN1	VDLN0
	LDVTLNR	—	—	—	—	—	VTLN10	VTLN9	VTLN8
		VTLN7	VTLN6	VTLN5	VTLN4	VTLN3	VTLN2	VTLN1	VTLN0
	LDVSYNR	VSYNW3	VSYNW2	VSYNW1	VSYNW0	—	VSYNP10	VSYNP9	VSYNP8
		VSYNP7	VSYNP6	VSYNP5	VSYNP4	VSYNP3	VSYNP2	VSYNP1	VSYNP0

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
LCDC	LDACLNR	—	—	—	—	—	—	—	—
		—	—	—	ACLN4	ACLN3	ACLN2	ACLN1	ACLN0
	LDINTR	MINTEN	FINTEN	VSINTEN	VEINTEN	MINTS	FINTS	VSINTS	VEINTS
		—	—	—	—	—	—	—	—
	LDPMMR	OCN3	OCN2	OCN1	OCN0	OFFD3	OFFD2	OFFD1	OFFD0
		—	VCPE	VEPE	DONE	—	—	LPS1	LPS0
	LDPSPR	ONA3	ONA2	ONA1	ONA0	ONB3	ONB2	ONB1	ONB0
		OFFE3	OFFE2	OFFE1	OFFE0	OFFF3	OFFF2	OFFF1	OFFF0
	LDCNTR	—	—	—	—	—	—	—	—
		—	—	—	DON2	—	—	—	DON
	LDUINTR	—	—	—	—	—	—	—	UINTEN
		—	—	—	—	—	—	—	UINTS
	LDUINTLNR	—	—	—	—	—	UINTLN10	UINTLN9	UINTLN8
		UINTLN7	UINTLN6	UINTLN5	UINTLN4	UINTLN3	UINTLN2	UINTLN1	UINTLN0
	LDLIRNR	—	—	—	—	—	—	—	—
		LIRN7	LIRN6	LIRN5	LIRN4	LIRN3	LIRN2	LIRN17	LIRN0
SRC	SRCID								
	SRCOD								
	SRCIDCTRL	—	—	—	—	—	—	IED	IEN
		—	—	—	—	—	—	IFTRG[1]	IFTRG[0]
	SRCODCTRL	—	—	—	—	—	OCH	OED	OEN
		—	—	—	—	—	—	OFTRG[1]	OFTRG[0]
	SRCCTRL	—	—	—	SRCEN	—	EEN	FL	CL
		IFS[3]	IFS[2]	IFS[1]	IFS[0]	—	—	—	OFS
	SRCSTAT	OFDN[3]	OFDN[2]	OFDN[1]	OFDN[0]	IFDN[4]	IFDN[3]	IFDN[2]	IFDN[1]
		IFDN[0]	—	—	FLF	—	OVF	IINT	OINT

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
PFC	PBIORL	—	—	—	—	PB11IOR	PB10IOR	PB9IOR	PB8IOR
		—	—	—	—	—	—	—	—
	PBCRL4	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	PB12MD[1]	PB12MD[0]
	PBCRL3	—	—	PB11MD[1]	PB11MD[0]	—	—	PB10MD[1]	PB10MD[0]
		—	—	PB9MD[1]	PB9MD[0]	—	—	PB8MD[1]	PB8MD[0]
	PBCRL2	—	—	PB7MD[1]	PB7MD[0]	—	—	PB6MD[1]	PB6MD[0]
		—	—	PB5MD[1]	PB5MD[0]	—	—	PB4MD[1]	PB4MD[0]
	PBCRL1	—	—	PB3MD[1]	PB3MD[0]	—	—	PB2MD[1]	PB2MD[0]
		—	—	PB1MD[1]	PB1MD[0]	—	—	PB0MD[1]	PB0MD[0]
	IFCR	—	—	—	—	—	—	—	—
		—	—	—	—	—	—	PB12IRQ1	PB12IRQ0
	PCIORL	—	PC14IOR	PC13IOR	PC12IOR	PC11IOR	PC10IOR	PC9IOR	PC8IOR
		PC7IOR	PC6IOR	PC5IOR	PC4IOR	PC3IOR	PC2IOR	PC1IOR	PC0IOR
	PCCRL4	—	—	—	—	—	—	—	PC14MD[0]
		—	—	—	PC13MD[0]	—	—	—	PC12MD[0]
	PCCRL3	—	—	PC11MD[1]	PC11MD[0]	—	—	PC10MD[1]	PC10MD[0]
		—	—	—	PC9MD[0]	—	—	—	PC8MD[0]
	PCCRL2	—	—	—	PC7MD[0]	—	—	—	PC6MD[0]
		—	—	—	PC5MD[0]	—	—	—	PC4MD[0]
	PCCRL1	—	—	—	PC3MD[0]	—	—	—	PC2MD[0]
		—	—	—	PC1MD[0]	—	—	PC0MD[1]	PC0MD[0]
	PDIORL	PD15IOR	PD14IOR	PD13IOR	PD12IOR	PD11IOR	PD10IOR	PD9IOR	PD8IOR
		PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR
	PDCRL4	—	PD15MD[2]	PD15MD[1]	PD15MD[0]	—	PD14MD[2]	PD14MD[1]	PD14MD[0]
		—	PD13MD[2]	PD13MD[1]	PD13MD[0]	—	PD12MD[2]	PD12MD[1]	PD12MD[0]
	PDCRL3	—	PD11MD[2]	PD11MD[1]	PD11MD[0]	—	PD10MD[2]	PD10MD[1]	PD10MD[0]
		—	PD9MD[2]	PD9MD[1]	PD9MD[0]	—	PD8MD[2]	PD8MD[1]	PD8MD[0]
	PDCRL2	—	PD7MD[2]	PD7MD[1]	PD7MD[0]	—	PD6MD[2]	PD6MD[1]	PD6MD[0]
		—	PD5MD[2]	PD5MD[1]	PD5MD[0]	—	PD4MD[2]	PD4MD[1]	PD4MD[0]
	PDCRL1	—	PD3MD[2]	PD3MD[1]	PD3MD[0]	—	PD2MD[2]	PD2MD[1]	PD2MD[0]
		—	PD1MD[2]	PD1MD[1]	PD1MD[0]	—	PD0MD[2]	PD0MD[1]	PD0MD[0]
	PEIORL	PE15IOR	PE14IOR	PE13IOR	PE12IOR	PE11IOR	PE10IOR	PE9IOR	PE8IOR
		PE7IOR	PE6IOR	PE5IOR	PE4IOR	PE3IOR	PE2IOR	PE1IOR	PE0IOR

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
PFC	PECRL4	—	—	PE15MD[1]	PE15MD[0]	—	—	PE14MD[1]	PE14MD[0]
		—	—	PE13MD[1]	PE13MD[0]	—	—	PE12MD[1]	PE12MD[0]
	PECRL3	—	PE11MD[2]	PE11MD[1]	PE11MD[0]	—	PE10MD[2]	PE10MD[1]	PE10MD[0]
		—	—	PE9MD[1]	PE9MD[0]	—	—	PE8MD[1]	PE8MD[0]
	PECRL2	—	PE7MD[2]	PE7MD[1]	PE7MD[0]	—	PE6MD[2]	PE6MD[1]	PE6MD[0]
		—	PE5MD[2]	PE5MD[1]	PE5MD[0]	—	PE4MD[2]	PE4MD[1]	PE4MD[0]
	PECRL1	—	—	PE3MD[1]	PE3MD[0]	—	—	PE2MD[1]	PE2MD[0]
		—	—	PE1MD[1]	PE1MD[0]	—	PE0MD[2]	PE0MD[1]	PE0MD[0]
	PFIORH	—	PF30IOR	PF29IOR	PF28IOR	PF27IOR	PF26IOR	PF25IOR	PF24IOR
		PF23IOR	PF22IOR	PF21IOR	PF20IOR	PF19IOR	PF18IOR	PF17IOR	PF16IOR
	PFIORL	PF15IOR	PF14IOR	PF13IOR	PF12IOR	PF11IOR	PF10IOR	PF9IOR	PF8IOR
		PF7IOR	PF6IOR	PF5IOR	PF4IOR	PF3IOR	PF2IOR	PF1IOR	PF0IOR
	PFCRH4	—	—	—	—	—	—	—	PF30MD[0]
		—	—	—	PF29MD[0]	—	—	—	PF28MD[0]
	PFCRH3	—	—	—	PF27MD[0]	—	—	—	PF26MD[0]
		—	—	—	PF25MD[0]	—	—	—	PF24MD[0]
	PFCRH2	—	—	PF23MD[1]	PF23MD[0]	—	—	PF22MD[1]	PF22MD[0]
		—	—	PF21MD[1]	PF21MD[0]	—	—	PF20MD[1]	PF20MD[0]
	PFCRH1	—	—	PF19MD[1]	PF19MD[0]	—	—	PF18MD[1]	PF18MD[0]
		—	—	PF17MD[1]	PF17MD[0]	—	—	PF16MD[1]	PF16MD[0]
	PFCRL4	—	—	PF15MD[1]	PF15MD[0]	—	—	PF14MD[1]	PF14MD[0]
		—	—	PF13MD[1]	PF13MD[0]	—	—	PF12MD[1]	PF12MD[0]
	PFCRL3	—	—	PF11MD[1]	PF11MD[0]	—	—	PF10MD[1]	PF10MD[0]
		—	—	PF9MD[1]	PF9MD[0]	—	—	PF8MD[1]	PF8MD[0]
	PFCRL2	—	—	PF7MD[1]	PF7MD[0]	—	—	PF6MD[1]	PF6MD[0]
		—	—	PF5MD[1]	PF5MD[0]	—	—	PF4MD[1]	PF4MD[0]
	PFCRL1	—	—	PF3MD[1]	PF3MD[0]	—	—	PF2MD[1]	PF2MD[0]
		—	—	PF1MD[1]	PF1MD[0]	—	—	PF0MD[1]	PF0MD[0]
	SCSR	—	S3CKS2	S3CKS1	S3CKS0	—	S2CKS2	S2CKS1	S2CKS0
		—	S1CKS2	S1CKS1	S1CKS0	—	S0CKS2	S0CKS1	S0CKS0
I/O 端口	PADRL	—	—	—	—	—	—	—	—
		PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR
	PBDRL	—	—	—	PB12DR	PB11DR	PB10DR	PB9DR	PB8DR
		PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
I/O 端口	PBPRL	—	—	—	—	PB11PR	PB10PR	PB9PR	PB8PR
		PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	PB0PR
	PCDRL	—	PC14DR	PC13DR	PC12DR	PC11DR	PC10DR	PC9DR	PC8DR
		PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
	PCPRL	—	PC14PR	PC13PR	PC12PR	PC11PR	PC10PR	PC9PR	PC8PR
		PC7PR	PC6PR	PC5PR	PC4PR	PC3PR	PC2PR	PC1PR	PC0PR
	PDDRL	PD15DR	PD14DR	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR	PD8DR
		PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
	PDPRL	PD15PR	PD14PR	PD13PR	PD12PR	PD11PR	PD10PR	PD9PR	PD8PR
		PD7PR	PD6PR	PD5PR	PD4PR	PD3PR	PD2PR	PD1PR	PD0PR
	PEDRL	PE15DR	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR
		PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
	PEPRL	PE15PR	PE14PR	PE13PR	PE12PR	PE11PR	PE10PR	PE9PR	PE8PR
		PE7PR	PE6PR	PE5PR	PE4PR	PE3PR	PE2PR	PE1PR	PE0PR
	PFDRH	—	PF30DR	PF29DR	PF28DR	PF27DR	PF26DR	PF25DR	PF24DR
		PF23DR	PF22DR	PF21DR	PF20DR	PF19DR	PF18DR	PF17DR	PF16DR
	PFDRL	PF15DR	PF14DR	PF13DR	PF12DR	PF11DR	PF10DR	PF9DR	PF8DR
		PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR
	PFPRH	—	PF30PR	PF29PR	PF28PR	PF27PR	PF26PR	PF25PR	PF24PR
		PF23PR	PF22PR	PF21PR	PF20PR	PF19PR	PF18PR	PF17PR	PF16PR
	PFPRL	PF15PR	PF14PR	PF13PR	PF12PR	PF11PR	PF10PR	PF9PR	PF8PR
		PF7PR	PF6PR	PF5PR	PF4PR	PF3PR	PF2PR	PF1PR	PF0PR
低功耗模式	STBCR	STBY	DEEP	—	—	—	—	—	—
	STBCR2	MSTP10	MSTP9	MSTP8	MSTP7	—	—	—	—
	STBCR3	HIZ	MSTP36	MSTP35	MSTP34	MSTP33	MSTP32	MSTP31	MSTP30
	STBCR4	MSTP47	MSTP46	MSTP45	MSTP44	MSTP43	MSTP42	MSTP41	MSTP40
	STBCR5	MSTP57	MSTP56	MSTP55	MSTP54	MSTP53	MSTP52	MSTP51	MSTP50
	STBCR6	MSTP67	MSTP66	MSTP65	MSTP64	MSTP63	MSTP62	—	MSTP60
	SYSCR1	—	—	—	—	RAME3	RAME2	RAME1	RAME0
	SYSCR2	—	—	—	—	RAMWE3	RAMWE2	RAMWE1	RAMWE0
	SYSCR3	AXTALE	—	—	IEBSRST	SSI3SRST	SSI2SRST	SSI1SRST	SSI0SRST
	DSCTR	—	—	—	—	RAMKP3	RAMKP2	RAMKP1	RAMKP0
	DSCTR2	CS0KEEPE	RAMBOOT	—	—	—	—	—	—

模块名	寄存器简称	bit31/23/15/7	bit30/22/14/6	bit29/21/13/5	bit28/20/12/4	bit27/19/11/3	bit26/18/10/2	bit25/17/9/1	bit24/16/8/0
低功耗模式	DSSSR	—	—	—	—	—	—	—	MRES
		IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
	DSFR	IOKEEP	—	—	—	—	—	MRESF	NMIF
		IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
	DSRTR	—	TRMD[6]	TRMD[5]	TRMD[4]	TRMD[3]	TRMD[2]	TRMD[1]	TRMD[0]
H-UDI	SDIR	T1[7]	T1[6]	T1[5]	T1[4]	T1[3]	T1[2]	T1[1]	T1[0]
		—	—	—	—	—	—	—	—

【注】 *1　存储器种类指定为通常空间时，带字节选择的 SRAM、地址 / 数据多路复用 I/O （MPX-IO）时。

*2　存储器种类指定为突发 ROM （时钟异步）时。

*3　存储器种类指定为突发 ROM （时钟同步）时。

*4　存储器种类指定为通常空间、带字节选择的 SRAM 时。

*5　存储器种类指定为 SDRAM 时。

*6　存储器种类指定为 PCMCIA 时。

*7　存储器种类指定为突发 MPX-IO 时。

*8　MCR15=0 时。

*9　MCR15=1 时。

*10 命令存取模式时

*11 扇区存取模式时

34.3　各运行模式下的寄存器状态一览

模块	寄存器名	上电复位	手动复位	深度待机	软件待机	模块待机	睡眠
CPG	FRQCR	初始化 *1	保持	初始化	保持	—	保持
INTC	IBNR	初始化	保持 *2	初始化	保持	—	保持
	上述以外的全部寄存器	初始化	保持	初始化	保持	—	保持
UBC	全部寄存器	初始化	保持	初始化	保持	保持	保持
高速缓存	全部寄存器	初始化	保持	初始化	保持	—	保持
BSC	RTCSR	初始化	保持 *3	初始化	保持	—	保持 *3
	RTCNT	初始化	保持 *4	初始化	保持	—	保持 *4
	上述以外的全部寄存器	初始化	保持	初始化	保持	—	保持
DMAC	全部寄存器	初始化	保持	初始化	保持	保持	保持 *5
MTU2	全部寄存器	初始化	保持	初始化	初始化	初始化	保持
CMT	全寄存器	初始化	保持	初始化	初始化	保持	保持
WDT	WRCSR	初始化 *1	保持	初始化	保持	—	保持
	上述以外的全部寄存器	初始化	保持	初始化	保持	—	保持
RTC	R64CNT	保持 *4	保持 *4	保持 *4	保持 *4	保持	保持 *4
	RSECCNT						
	RMINCNT						
	RHRCNT						
	RWKCNT						
	RDAYCNT						
	RMONCNT						
	RYRCNT						
	RSECAR	初始化	保持	初始化	保持	保持	保持
	RMINAR						
	RHRAR						
	RWKAR						
	RDAYAR						
	RMONAR						
	RYRAR						
	RCR1	初始化	初始化	初始化	保持	保持	保持
	RCR2	初始化	初始化 *6	初始化	保持	保持	保持
	RCR3	初始化	保持	初始化	保持	保持	保持
	全部寄存器	初始化	保持	初始化	保持	保持	保持
SCIF	全部寄存器	初始化	保持	初始化	保持	保持	保持
SSU	全部寄存器	初始化	保持	初始化	初始化	初始化	保持

模块	寄存器名	上电复位	手动复位	深度待机	软件待机	模块待机	睡眠
IIC3	ICMR_0、 1、 2、 3	初始化	保持	初始化	保持 *7	保持 *7	保持
	上述以外的全部寄存器	初始化	保持	初始化	保持	保持	保持
SSI	全部寄存器	初始化	保持	初始化	保持	保持	保持
RCAN-TL1	全部寄存器	初始化	保持	初始化	保持	保持	保持
IEB	全部寄存器	初始化	保持	初始化	保持	保持	保持
ROM-DEC	全部寄存器	初始化	保持	初始化	保持	保持	保持
ADC	全部寄存器	初始化	保持	初始化	初始化	初始化	保持
DAC	全部寄存器	初始化	保持	初始化	保持	初始化	保持
FLCTL	全部寄存器	初始化	保持	初始化	保持	保持	保持
USB	全部寄存器	初始化	保持	初始化	保持	保持	保持
LCDC	全部寄存器	初始化	保持	初始化	保持	保持	保持
PFC	全部寄存器	初始化	保持	初始化	保持	—	保持
I/O 端口	全部寄存器 *8	初始化	保持	初始化	保持	—	保持
低功耗模式	STBCR	初始化	保持	初始化	保持	—	保持
	STBCR2	初始化	保持	初始化	保持	—	保持
	SYSCR1	初始化	保持	初始化	保持	—	保持
	SYSCR2	初始化	保持	初始化	保持	—	保持
	SYSCR3	初始化	保持	初始化	保持	—	保持
	STBCR3	初始化	保持	初始化	保持	—	保持
	STBCR4	初始化	保持	初始化	保持	—	保持
	STBCR5	初始化	保持	初始化	保持	—	保持
	STBCR6	初始化	保持	初始化	保持	—	保持
	DSCTR	初始化	保持	初始化	保持	—	保持
	DSCTR2	初始化	保持	保持	保持	—	保持
	DSSSR	初始化	保持	初始化	保持	—	保持
	DSFR	初始化	保持	保持	保持	—	保持
	DSRTR	初始化 *10	保持	初始化	保持	—	保持
H-UDI*9	SDIR	保持	保持	初始化	保持	保持	保持
SRC	全部寄存器	初始化	保持	初始化	保持	保持	保持

【注】 *1　在通过 WDT 进行内部上电复位后，保持之前的值。

*2　初始化 BN[3:0] 位。

*3　继续执行标志处理。

*4　继续执行累加计数。

*5　能继续传送运行。

*6　保持 RTCEN、 START 位。

*7　初始化 BC[3:0] 位。

*8　由于 PADRL 及端口寄存器用于读出引脚状态，所以既不初始化也不保持之前的值。

*9　$\overline{\text{TRST}}$ 断言或者 TAP 的 Test-Logic-Reset 状态时初始化。

*10 通过 $\overline{\text{RES}}$ 断言进行初始化，并且在通过 H-UDI 复位断言命令或者 WDT 进行内部上电复位后，保持之前的值。

第 35 章　电特性

35.1　绝对最大额定值

表 35.1　绝对最大额定值

项目		符号	额定值	单位
电源电压 （I/O）		PV_{CC}	–0.3 ～ 4.6	V
电源电压 （内部）		V_{CC}	–0.3 ～ 1.7	V
PLL 电源电压		$PLLV_{CC}$	–0.3 ～ 1.7	V
模拟电源电压		AV_{CC}	–0.3 ～ 4.6	V
模拟基准电压		AV_{ref}	–0.3 ～ AV_{CC}+0.3	V
USB 收发器部分模拟电源电压 （I/O）		$USBAPV_{CC}$	–0.3 ～ 4.6	V
USB 收发器部分数字电源电压 （I/O）		$USBDPV_{CC}$	–0.3 ～ 4.6	V
USB 收发器部分模拟电源电压 （内部）		$USBAV_{CC}$	–0.3 ～ 1.7	V
USB 收发器部分数字电源电压 （内部）		$USBDV_{CC}$	–0.3 ～ 1.7	V
输入电压	模拟输入引脚	V_{AN}	–0.3 ～ AV_{CC}+0.3	V
	其他输入引脚	V_{in}	–0.3 ～ PV_{CC}+0.3	V
工作温度		T_{opr}	–40 ～ 85	°C
保存温度		T_{stg}	–55 ～ 125	°C

【使用时的注意事项】

超过绝对最大额定值使用 LSI 时，会导致 LSI 的永久性破坏。

35.2　通电、断电顺序

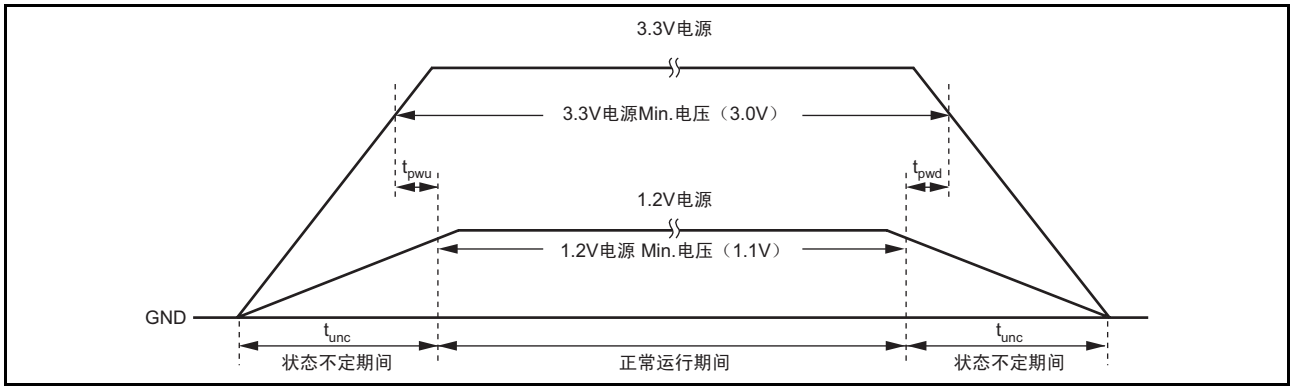


图 35.1　通电、断电顺序

表 35.2　通电、断电时间

项目	符号	Min.	Max.	单位
3.3V–1.2V 通电时间差 *1	t_{pwu}	0	—	ms
1.2V–3.3V 断电时间差 *1	t_{pwd}	0	—	ms
状态不定时间 *2	t_{unc}	—	100	ms

【注】　*1　请在接通 1.2V 电源（V_{CC}、$PLLV_{CC}$、$USBAV_{CC}$、$USBDV_{CC}$）之前先接通 3.3V 电源（PV_{CC}、AV_{CC}、$USBAPV_{CC}$、$USBDPV_{CC}$）。同样，请在切断 1.2V 电源之后再切断 3.3V 电源。

*2　为推荐值，并非要求严格设定的值。 1.2V 电源超过 Min. 电压之前以及低于 Min. 电压之后，为状态不定期间，在此期间引脚状态及内部状态不定。进行系统设计时，请避免由于此状态而使整个系统引起误操作。

35.3　DC 特性

表 35.3　DC 特性 （1）【通用项目】

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、 AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目		符号	Min.	Typ.	Max.	单位	测量条件
电源电压		PV_{CC}	3.0	3.3	3.6	V	
		V_{CC}	1.1	1.2	1.3	V	
PLL 电源电压		$PLLV_{CC}$	1.1	1.2	1.3	V	
模拟电源电压		AV_{CC}	3.0	3.3	3.6	V	
USB 电源电压		$USBAPV_{CC}$ $USBDPV_{CC}$	3.0	3.3	3.6	V	
		$USBAV_{CC}$ $USBDV_{CC}$	1.1	1.2	1.3	V	
消耗电流 *1	正常操作时	I_{CC}*2	—	240	400	mA	V_{CC}=1.2V Iϕ=200.00MHz
	睡眠模式时	I_{sleep}*2	—	180	360	mA	Bϕ=66.66MHz Pϕ=33.33MHz
	软件待机模式时	I_{sstby}*2	—	12	120	mA	T_a ＞ 50°C V_{CC}=1.2V
			—	4	40	mA	T_a ≤ 50°C V_{CC}=1.2V
	深度待机模式时	I_{dstby}	—	5	30	μA	T_a ＞ 50°C 1.2V 电源 *3=1.2V RAM 0KB 保持
			—	23	130	μA	T_a ＞ 50°C 1.2V 电源 *3=1.2V RAM 4KB 保持
			—	41	230	μA	T_a ＞ 50°C 1.2V 电源 *3=1.2V RAM 8KB 保持
			—	59	330	μA	T_a ＞ 50°C 1.2V 电源 *3=1.2V RAM 12KB 保持
			—	77	430	μA	T_a ＞ 50°C 1.2V 电源 *3=1.2V RAM 16KB 保持
			—	9	58	μA	T_a ＞ 50°C 3.3V 电源 *4=3.3V
			—	11	12	μA	T_a ＞ 50°C VBUS=5.0V

项目		符号	Min.	Typ.	Max.	单位	测量条件
消耗电流 *1	深度待机模式时	I_{dstby}	—	2	10	μA	$T_a \leqslant 50°C$ 1.2V 电源 *3=1.2V RAM 0KB 保持
			—	12	24	μA	$T_a \leqslant 50°C$ 1.2V 电源 *3=1.2V RAM 4KB 保持
			—	22	38	μA	$T_a \leqslant 50°C$ 1.2V 电源 *3=1.2V RAM 8KB 保持
			—	32	52	μA	$T_a \leqslant 50°C$ 1.2V 电源 *3=1.2V RAM 12KB 保持
			—	42	66	μA	$T_a \leqslant 50°C$ 1.2V 电源 *3=1.2V RAM 16KB 保持
			—	5	20	μA	$T_a \leqslant 50°C$ 3.3V 电源 *4=3.3V
			—	11	12	μA	$T_a \leqslant$～ 50°CVBUS=5.0V
输入泄漏电流	全部输入引脚（除 PB7 ～ PB0）	$\|I_{in}\|$	—	—	1.0	μA	V_{in}=0.5 ～ PV_{CC}–0.5V
	PB7 ～ PB0		—	—	10	μA	
三态泄漏电流	全部输入、输出引脚（除 PB7 ～ PB0 及弱保持引脚）（断路状态）	$\|I_{STI}\|$	—	—	1.0	μA	V_{in}=0.5 ～ PV_{CC}–0.5V
	PB7 ～ PB0		—	—	10	μA	
引脚电容	全部引脚	C_{in}	—	—	20	pF	
模拟电源电流	A/D、D/A 转换中	AI_{CC}	—	2	4	mA	
	A/D、D/A 转换待机时		—	1	10	μA	
模拟基准电压电流		AI_{ref}	—	2	4	mA	
USB 电源电流	$USBAV_{CC}$+ $USBDV_{CC}$	I_{USBCC}	—	15	20	mA	$USBAV_{CC}$= $USBDV_{CC}$=1.2V
	$USBAPV_{CC}$+ $USBDPV_{CC}$	I_{USBPCC}	—	40	50	mA	$USBAPV_{CC}$= $USBDPV_{CC}$=3.3V

【使用时的注意事项】在未使用 A/D 转换器及 D/A 转换器时，请勿将 AV_{CC}、AV_{SS} 引脚置为开路。

【注】 *1　消费电流为将全部输出引脚及带上拉的引脚设置为无负载状态时的值。

*2　I_{CC}、I_{sleep}、I_{sstby} 为 V_{CC}、$PLLV_{CC}$ 系统中消耗的电流的合计值。

*3　I_{dstby} 的 1.2V 电流为 V_{CC}、$PLLV_{CC}$、$USBAV_{CC}$、$USBDV_{CC}$ 中消耗的电流的合计值。

*4　I_{dstby} 的 3.3V 电流为 PV_{CC}、AV_{CC}、$USBAPV_{CC}$、$USBDPV_{CC}$ 中消耗的电流的合计值。

表 35.3　DC 特性（2）【除 I²C、USB 相关引脚】

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、AV_{CC}=3.0 ～ 3.6V、$USBAV_{CC}$=1.1 ～ 1.3V、$USBAPV_{CC}$=3.0 ～ 3.6V、V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、T_a=–40 ～ 85°C

项目		符号	Min.	Typ.	Max.	单位	测量条件
输入高电平电压	$\overline{RES}$、$\overline{MRES}$、NMI、MD、MD_CLK1、MD_CLK0、$\overline{ASEMD}$、$\overline{TRST}$、EXTAL、CKIO、AUDIO_X1、RTC_X1	V_{IH}	PV_{CC}–0.5	—	PV_{CC}+0.3	V	
	PA7 ～ PA0		2.2	—	AV_{CC}+0.3	V	
	其他输入引脚（除施密特引脚）		2.2	—	PV_{CC}+0.3	V	
输入低电平电压	$\overline{RES}$、$\overline{MRES}$、NMI、MD、MD_CLK1、MD_CLK0、$\overline{ASEMD}$、$\overline{TRST}$、EXTAL、CKIO、AUDIO_X1、RTC_X1	V_{IL}	–0.3	—	0.5	V	
	其他输入引脚（除施密特引脚）		–0.3	—	0.8	V	
施密特触发输入特性	IRQ7 ～ IRQ0、PINT7 ～ PINT0、$\overline{IOIS16}$、DREQ3 ～ DREQ0、TIOC0A ～ TIOC0D、TIOC1A、TIOC1B、TIOC2A、TIOC2B、TIOC3A ～ TIOC3D、TIOC4A ～ TIOC4D、TCLKA ～ TCLKD、SCK3 ～ SCK0、RxD3 ～ RxD0、$\overline{CTS3}$、$\overline{RTS3}$、SSCK1、SSCK0、SSI1、SSI0、SSO1、SSO0、$\overline{SCS1}$、$\overline{SCS0}$、$\overline{ADTRG}$、PE15 ～ PE0、PF7 ～ PF0	V_T^+	PV_{CC}–0.5	—	—	V	
		V_T^-	—	—	0.5	V	
		V_T^+–V_T^-	0.2	—	—	V	
输出高电平电压		V_{OH}	PV_{CC}–0.5	—	—	V	I_{OH}=–200μA
输出低电平电压		V_{OL}	—	—	0.4	V	I_{OL}=1.6mA
RAM 待机电压	软件待机模式时	V_{RAMS}	0.75	—	—	V	将 V_{CC}（=$PLLV_{CC}$）作为参数进行测量
	深度待机模式时（仅为保持用内置 RAM）	V_{RAMD}	1.1	—	—	V	

表 35.3　DC 特性 （3）【I²C 相关引脚 *】

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、 AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目	符号	Min.	Typ.	Max.	单位	测量条件
输入高电平电压	V_{IH}	PV_{CC}×0.7	—	PV_{CC}+0.3	V	
输入低电平电压	V_{IL}	–0.3	—	PV_{CC}×0.3	V	
施密特触发输入特性	V_{IH}–V_{IL}	PV_{CC}×0.05	—	—	V	
输出低电平电压	V_{OL}	—	—	0.4	V	I_{OL}=3.0mA

【注】 *　PB7/SDA3/PINT7/IRQ7 ～ PB0/SCL0/PINT0/IRQ0 引脚 （漏极开路引脚）

表 35.3　DC 特性 （4）【USB 相关引脚 *】

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、 AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目	符号	Min.	Typ.	Max.	单位	测量条件
基准电阻	R_{REF}	5.6kΩ ± 1%				
输入高电平电压 （VBUS）	V_{IH}	4.0	—	5.5	V	
输入低电平电压 （VBUS）	V_{IL}	–0.3	—	1.0	V	
输入高电平电压 （USB_X1）	V_{IH}	PV_{CC}–0.5	—	PV_{CC}+0.3	V	
输入低平电压 （USB_X1）	V_{IL}	–0.3	—	0.5	V	

【注】 *　REFRIN、 VBUS、 USB_X1、 USB_X2 引脚

表 35.3　DC 特性 （5）【USB 相关引脚 * （全速 / 高速通用项目）】

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、 AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目	符号	Min.	Typ.	Max.	单位	测量条件
DP 上拉电阻 （选择功能时）	R_{pu}	0.900	—	1.575	kΩ	空闲时
		1.425	—	3.090	kΩ	发送 / 接收时
DP、 DM 下拉电阻 （选择主机功能时）	R_{pd}	14.25	—	24.80	kΩ	

【注】 *　DP、 DM 引脚

表 35.3 DC 特性 （6）【USB 相关引脚 *（全速时）】

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、 AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目	符号	Min.	Typ.	Max.	单位	测量条件
输入高电平电压	V_{IH}	2.0	—	—	V	
输入低电平电压	V_{IL}	—	—	0.8	V	
差动输入灵敏度	V_{DI}	0.2	—	—	V	|（DP）–（DM）|
差动共模范围	V_{CM}	0.8	—	2.5	V	
输出高电平电压	V_{OH}	2.8	—	3.6	V	I_{OH}=–200μA
输出低电平电压	V_{OL}	0.0	—	0.3	V	I_{OL}=2mA
单端的接收器阈值电压	V_{SE}	0.8	—	2.0	V	
输出信号交叉电压	V_{ORS}	1.3	—	2.0	V	C_L=50pF

【注】 * DP、 DM 引脚

表 35.3 DC 特性 （7）【USB 相关引脚 *（高速时）】

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、 AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目	符号	Min.	Typ.	Max.	单位	测量条件
噪声抑制电路检测阈值电压 （差动电压）	V_{HSSQ}	100	—	150	mV	
共模电压范围	V_{HSCM}	–50	—	500	mV	
空闲状态	V_{HSOI}	–10.0	—	10.0	mV	
输出高电平电压	V_{HSOH}	360	—	440	mV	
输出低电平电压	V_{HSOL}	–10.0	—	10.0	mV	
Chirp J 输出电压 （差分）	V_{CHIRPJ}	700	—	1100	mV	
Chirp K 输出电压 （差分）	V_{CHIRPK}	–900	—	–500	mV	

【注】 * DP、 DM 引脚

表 35.4　输出容许电流值

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、 AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目		符号	Min.	Typ.	Max.	单位
输出低电平容许电流（每个引脚）	PB7 ～ PB0	I_{OL}	—	—	10	mA
	其他输出引脚				2	mA
输出低电平容许电流 （总和）		ΣI_{OL}	—	—	150	mA
输出高电平容许电流 （每个引脚）		$-I_{OH}$	—	—	2	mA
输出高电平容许电流 （总和）		$\Sigma -I_{OH}$	—	—	50	mA

【使用时的注意事项】

为了确保 LSI 的可靠性，输出电流值请勿超过表 35.4 所列的值。

35.4　AC 特性

本 LSI 的输入，原则上为时钟同步输入。只要无特殊理由，就必须遵守各输入信号的设置 / 保持时间。

表 35.5　最大工作频率

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、 AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目		符号	Min.	Max.	单位	备注
工作频率	CPU （Iϕ）	f	80.00	200.00	MHz	
	内部总线、外部总线 （Iϕ）		40.00	66.66	MHz	
	外围模块 （Pϕ）		6.66	33.33	MHz	

35.4.1 时钟时序

表 35.6 时钟时序

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、 AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目	符号	Min.	Max.	单位	参考图
EXTAL 时钟输入频率	f_{EX}	10.00	33.33	MHz	35.2
EXTAL 时钟输入周期时间	t_{EXcyc}	30	100	ns	
AUDIO_X1、 AUDIO_CLK 时钟输入频率	f_{EX}	10	40	MHz	
AUDIO_X1、 AUDIO_CLK 时钟输入周期时间	t_{EXcyc}	25	100	ns	
USB_X1 时钟输入频率	f_{EX}	48MHz ± 100ppm			
EXTAL、 AUDIO_X1、 AUDIO_CLK 时钟输入低电平脉冲宽度	t_{EXL}	0.4	0.6	t_{EXcyc}	
EXTAL、 AUDIO_X1、 AUDIO_CLK 时钟输入高电平脉冲宽度	t_{EXH}	0.4	0.6	t_{EXcyc}	
EXTAL、 AUDIO_X1、 AUDIO_CLK、 USB_X1 时钟输入上升时间	t_{EXr}	—	4	ns	
EXTAL、 AUDIO_X1、 AUDIO_CLK、 USB_X1 时钟输入下降时间	t_{EXf}	—	4	ns	
CKIO 时钟输入频率	f_{CK}	40.00	66.66	MHz	35.3
CKIO 时钟输入周期时间	t_{CKIcyc}	15	25	ns	
CKIO 时钟输入低电平脉冲宽度	t_{CKIL}	0.4	0.6	t_{CKIcyc}	
CKIO 时钟输入高电平脉冲宽度	t_{CKIH}	0.4	0.6	t_{CKIcyc}	
CKIO 时钟输入上升时间	t_{CKIr}	—	3	ns	
CKIO 时钟输入下降时间	t_{CKIf}	—	3	ns	
CKIO 时钟输出频率	f_{OP}	40.00	66.66	MHz	35.4
CKIO 时钟输出周期时间	t_{cyc}	15	25	ns	
CKIO 时钟输出低电平脉冲宽度	t_{CKOL}	0.4	0.6	t_{cyc}	
CKIO 时钟输出高电平脉冲宽度	t_{CKOH}	0.4	0.6	t_{cyc}	
CKIO 时钟输出上升时间	t_{CKOr}	—	3	ns	
CKIO 时钟输出下降时间	t_{CKOf}	—	3	ns	
上电振荡稳定时间	t_{OSC1}	10	—	ms	35.5
待机返回振荡稳定时间 1	t_{OSC2}	10	—	ms	35.6
待机返回振荡稳定时间 2	t_{OSC3}	10	—	ms	35.7
RTC 时钟振荡稳定时间	t_{ROSC}	3	—	s	35.8

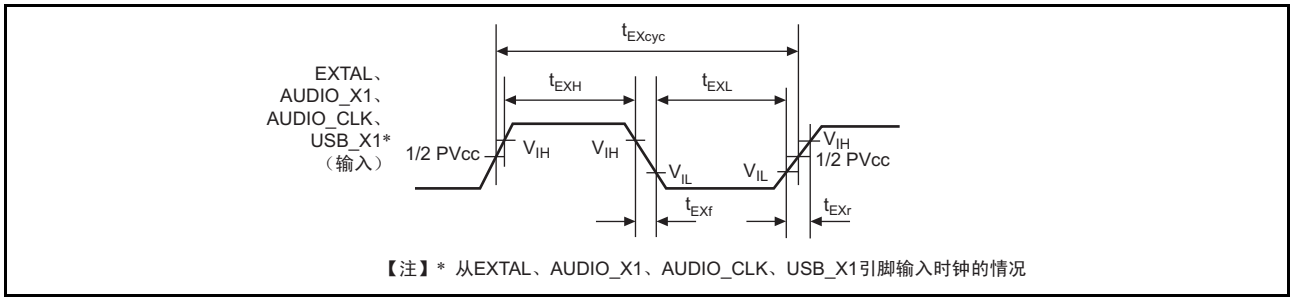


图 35.2 EXTAL、 AUDIO_X1、 AUDIO_CLK、 USB_X1 时钟输入时序

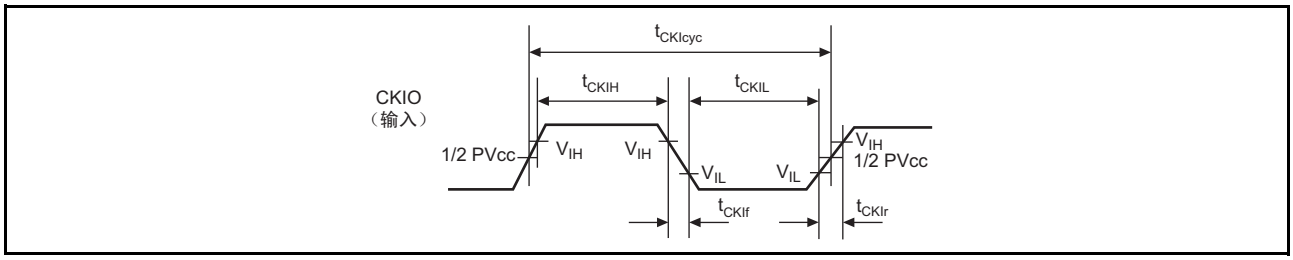


图 35.3 CKIO 时钟输入时序

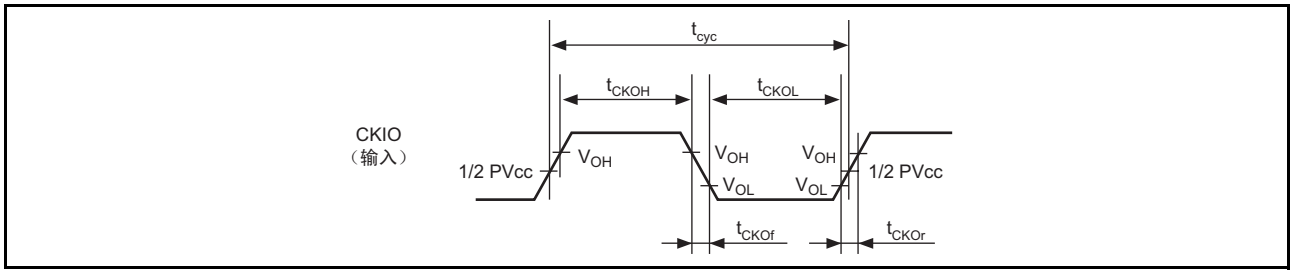


图 35.4 CKIO 时钟输出时序

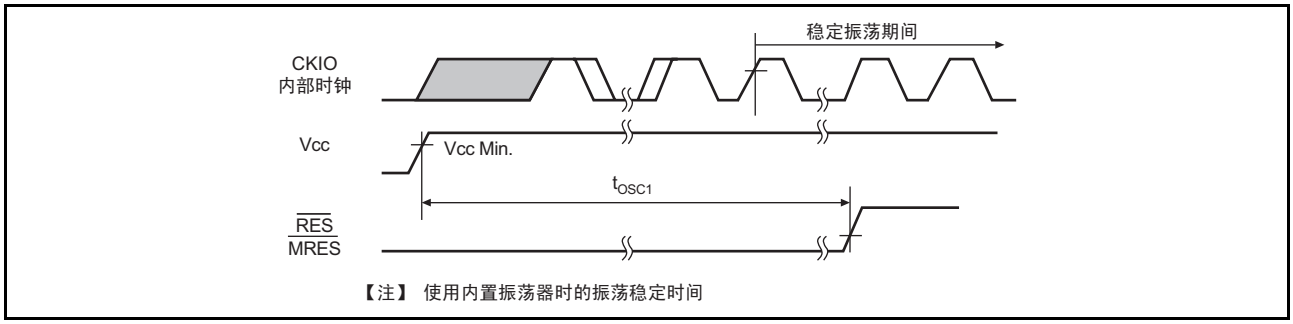


图 35.5 上电振荡稳定时间

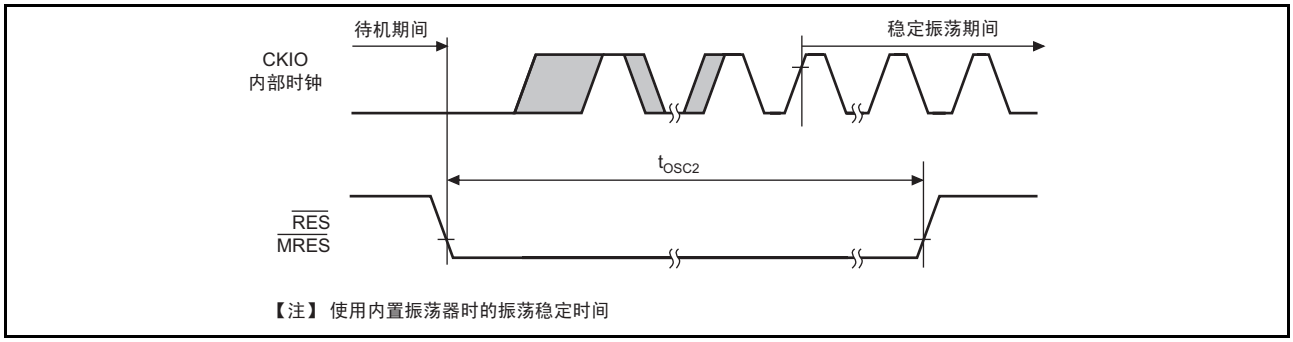


图 35.6 待机返回时的振荡稳定时间 （通过复位产生的返回）

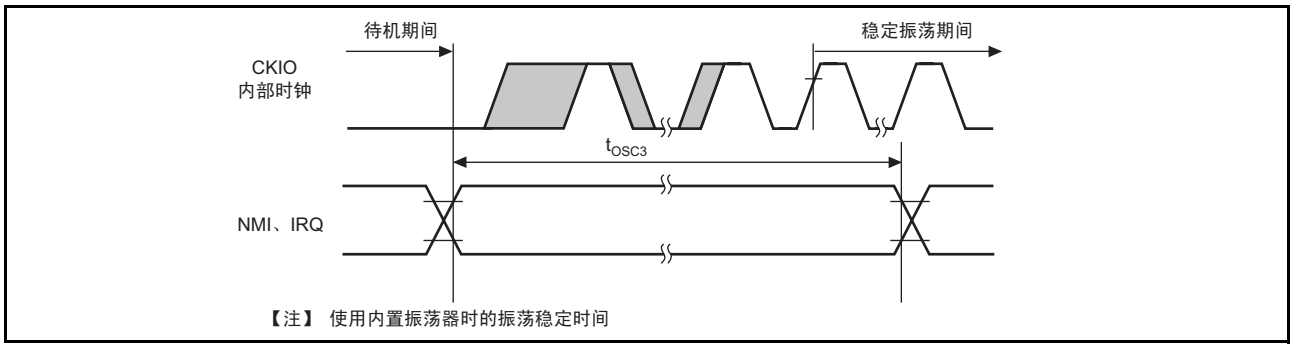


图 35.7 待机返回时的振荡稳定时间 （通过 NMI、 IRQ 产生的返回）

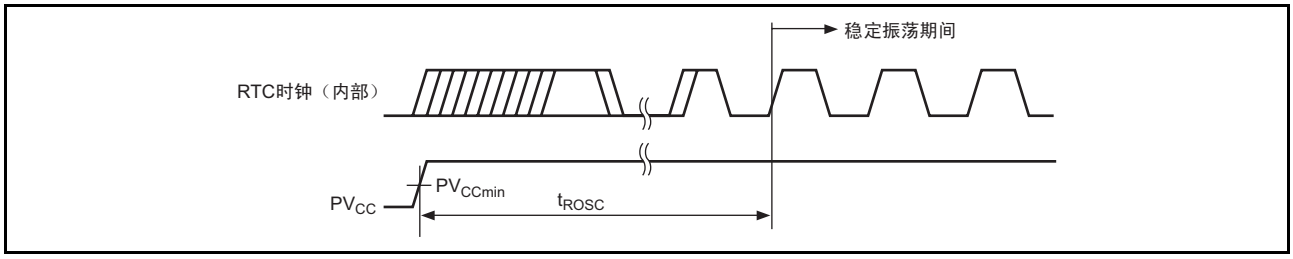


图 35.8 RTC 时钟振荡稳定时间

35.4.2 控制信号时序

表 35.7 控制信号时序

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、 AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目	符号	Bϕ=66.66MHz		单位	参考图
		Min.	Max.		
$\overline{\text{RES}}$ 脉冲宽度	t_{RESW}	20*1	—	t_{cyc}	35.9
$\overline{\text{MRES}}$ 脉冲宽度	t_{MRESW}	20*2	—	t_{cyc}	
NM 脉冲宽度	t_{NMIW}	20*3	—	t_{cyc}	35.10
IRQ 脉冲宽度	t_{IRQW}	20*3	—	t_{cyc}	
PINT 脉冲宽度	t_{PINTW}	20	—	t_{cyc}	
$\overline{\text{IRQOUT}}$/$\overline{\text{REFOUT}}$ 输出延迟时间	t_{IRQOD}	—	100	ns	35.11
$\overline{\text{BREQ}}$ 设置时间	t_{BREQS}	$1/2t_{cyc}$+7	—	ns	35.12
$\overline{\text{BREQ}}$ 保持时间	t_{BREQH}	$1/2t_{cyc}$+2	—	ns	
$\overline{\text{BACK}}$ 延迟时间	t_{BACKD}	—	$1/2t_{cyc}$+13	ns	
总线三态延迟时间 1	t_{BOFF1}	0	15	ns	
总线三态延迟时间 2	t_{BOFF2}	0	15	ns	
总线缓冲器保持时间 1	t_{BON1}	0	15	ns	
总线缓冲器保持时间 2	t_{BON2}	0	15	ns	

【注】 *1 待机模式时或时钟倍频率发生变化时，为 t_{RESW}=t_{OSC2} （10ms）。

*2 待机模式时，为 t_{MRESW}=t_{OSC2} （10ms）。

*3 待机模式时，为 t_{NMIW}/t_{IRQW}=t_{OSC3} （10ms）。

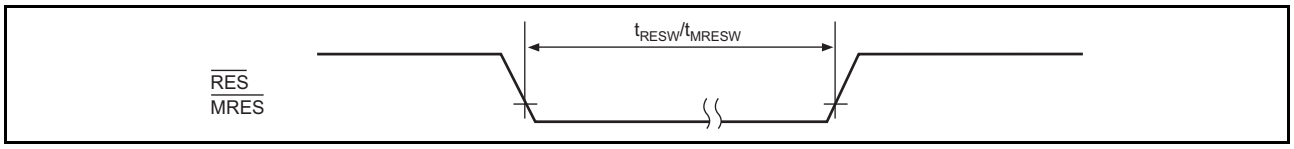


图 35.9 复位输入时序

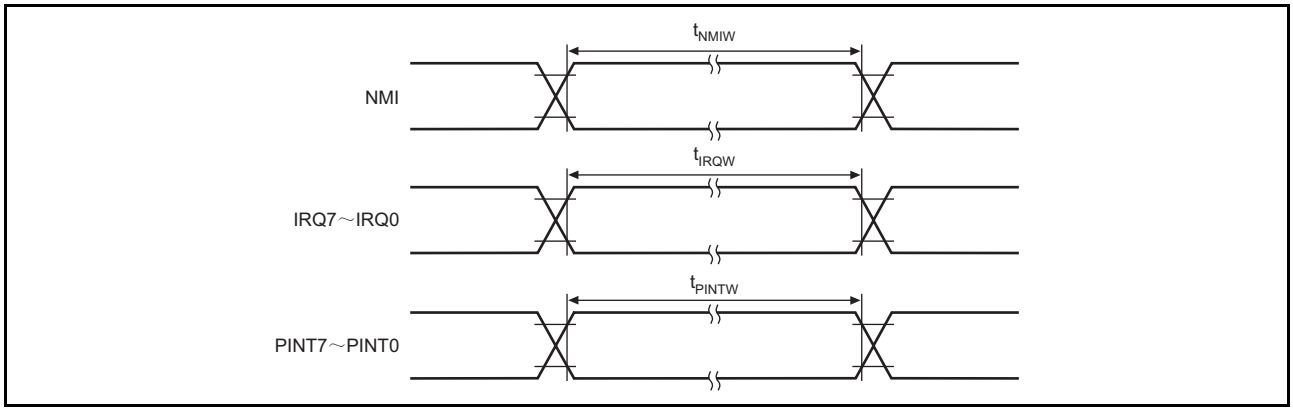


图 35.10 中断信号输入时序

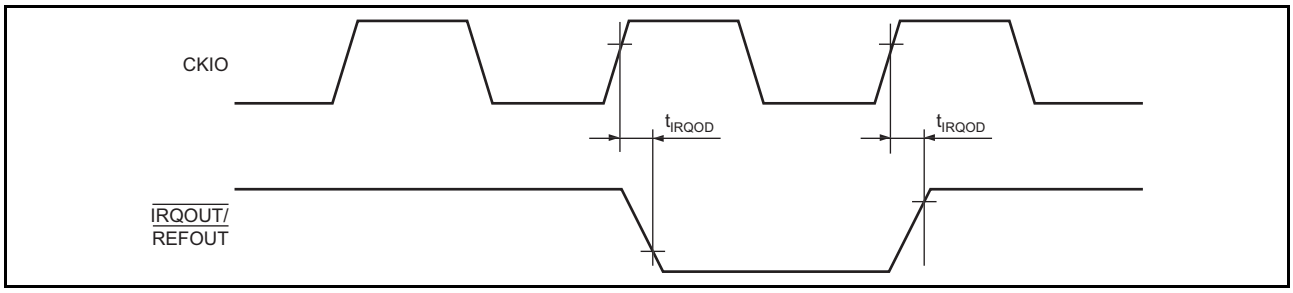


图 35.11 中断信号输出时序

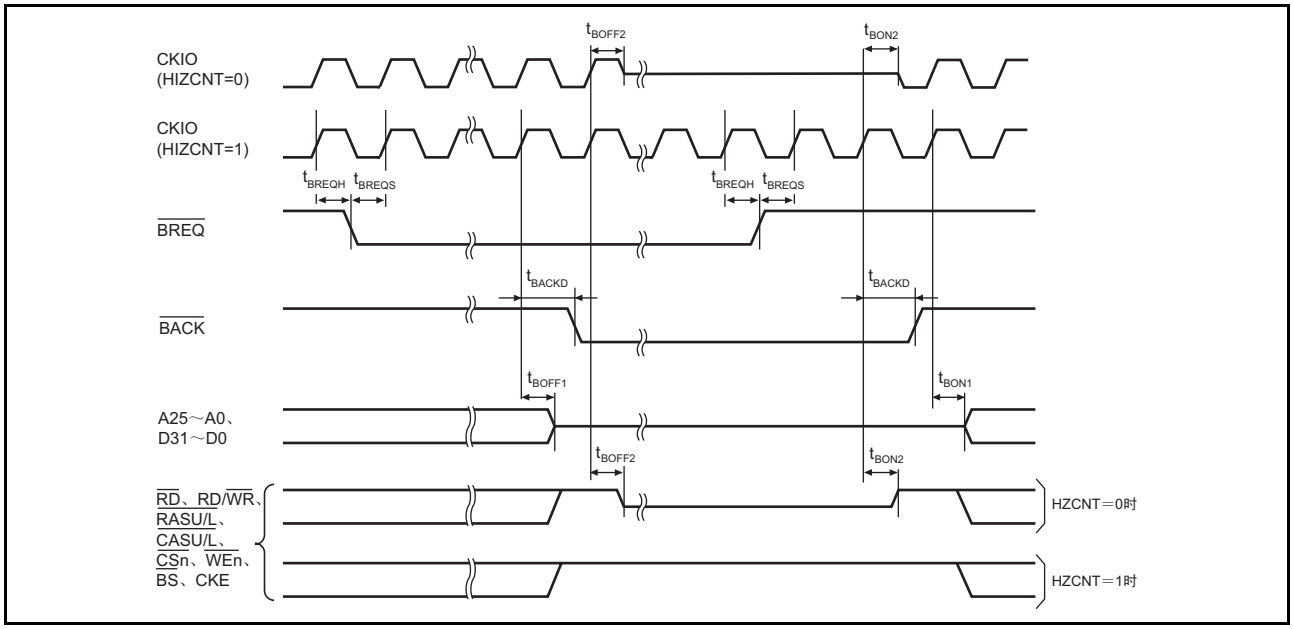


图 35.12 总线释放时序

35.4.3 总线时序

表 35.8 总线时序

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、 AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

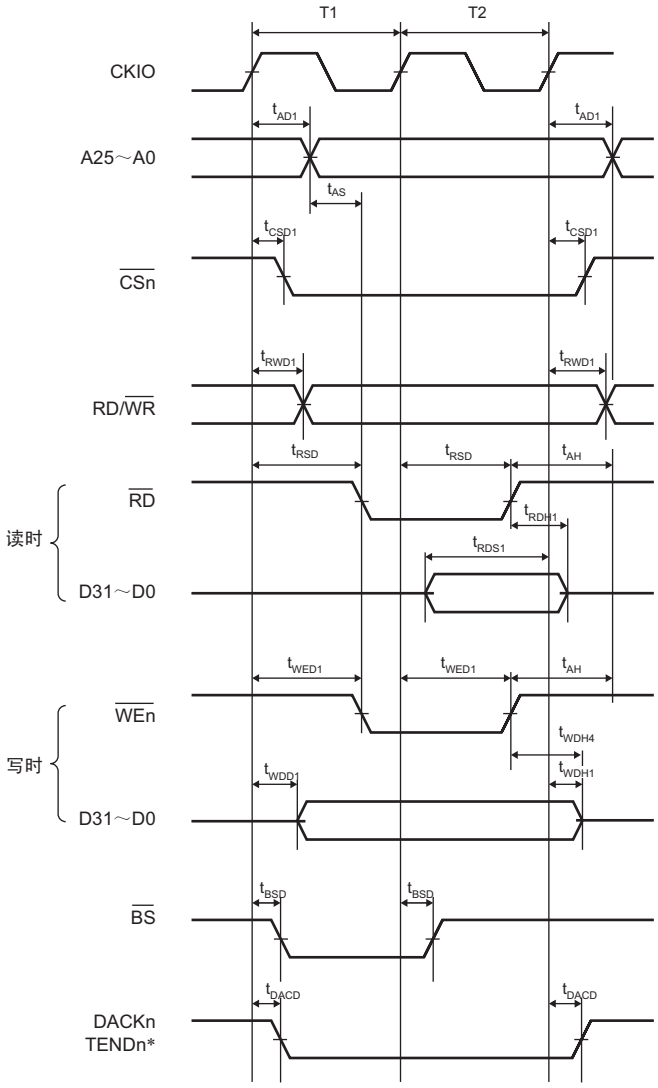
项目	符号	Bϕ=66.66MHz*1,*3		单位	参考图
		Min.	Max.		
地址延迟时间 1	t_{AD1}	0/1*2	13	ns	35.13 ～ 35.38、 35.40 ～ 35.44
地址延迟时间 2	t_{AD2}	$1/2t_{cyc}$	$1/2t_{cyc}$+13	ns	35.21
地址延迟时间 3	t_{AD3}	$1/2t_{cyc}$	$1/2t_{cyc}$+13	ns	35.39、 35.40
地址设置时间	t_{AS}	0	—	ns	35.13 ～ 35.16、 35.21
地址保持时间	t_{AH}	0	—	ns	35.13 ～ 35.16
$\overline{BS}$ 延迟时间	t_{BSD}	—	13	ns	35.13 ～ 35.35、 35.39、 35.41 ～ 35.44
$\overline{CS}$ 延迟时间 1	t_{CSD1}	0/1*2	13	ns	35.13 ～ 35.38、 35.41 ～ 35.44
$\overline{CS}$ 延迟时间 2	t_{CSD2}	$1/2t_{cyc}$	$1/2t_{cyc}$+13	ns	35.39、 35.40
读写延迟时间 1	t_{RWD1}	0/1*2	13	ns	35.13 ～ 35.38、 35.41 ～ 35.44
读写延迟时间 2	t_{RWD2}	$1/2t_{cyc}$	$1/2t_{cyc}$+13	ns	35.39、 35.40
读选通延迟时间	t_{RSD}	$1/2t_{cyc}$	$1/2t_{cyc}$+13	ns	35.13 ～ 35.17、 35.19 ～ 35.21、 35.41、 35.42
读数据设置时间 1	t_{RDS1}	$1/2t_{cyc}$+13	—	ns	35.13 ～ 35.17、 35.19、 35.20、 35.41 ～ 35.44
读数据设置时间 2	t_{RDS2}	8	—	ns	35.18、 35.22 ～ 35.25、 35.30 ～ 35.32
读数据设置时间 3	t_{RDS3}	1/2tcyc+13	—	ns	35.21
读数据设置时间 4	t_{RDS4}	1/2tcyc+13	—	ns	35.39
读数据保持时间 1	t_{RDH1}	0	—	ns	35.13 ～ 35.17、 35.19、 35.20、 35.41 ～ 35.44
读数据保持时间 2	t_{RDH2}	2	—	ns	35.18、 35.22 ～ 35.25、 35.30 ～ 35.32
读数据保持时间 3	t_{RDH3}	0	—	ns	35.21
读数据保持时间 4	t_{RDH4}	$1/2t_{cyc}$+6	—	ns	35.39
写允许延迟时间 1	t_{WED1}	$1/2t_{cyc}$	$1/2t_{cyc}$+13	ns	35.13 ～ 35.17、 35.18、 35.41、 35.42
写允许延迟时间 2	t_{WED2}	—	13	ns	35.20
写数据延迟时间 1	t_{WDD1}	—	13	ns	35.13 ～ 35.20、 35.41 ～ 35.44
写数据延迟时间 2	t_{WDD2}	—	13	ns	35.26 ～ 35.29、 35.33 ～ 35.35
写数据延迟时间 3	t_{WDD3}	—	$1/2t_{cyc}$+13	ns	35.39
写数据保持时间 1	t_{WDH1}	1	—	ns	35.13 ～ 35.20、 35.41 ～ 35.44
写数据保持时间 2	t_{WDH2}	1	—	ns	35.26 ～ 35.29、 35.33 ～ 35.35
写数据保持时间 3	t_{WDH3}	$1/2t_{cyc}$	—	ns	35.39
写数据保持时间 4	t_{WDH4}	0	—	ns	35.13、 35.17、 35.41、 35.43
$\overline{WAIT}$ 设置时间	t_{WTS}	$1/2t_{cyc}$+8	—	ns	35.14 ～ 35.21、 35.42、 35.44
$\overline{WAIT}$ 保持时间	t_{WTH}	$1/2t_{cyc}$+5	—	ns	35.14 ～ 35.21、 35.42、 35.44

项目	符号	Bϕ=66.66MHz*1,*3		单位	参考图
		Min.	Max.		
$\overline{\text{IOIS16}}$ 设置时间	t_{IO16S}	$1/2t_{cyc}$+8	—	ns	35.44
$\overline{\text{IOIS16}}$ 保持时间	t_{IO16H}	$1/2t_{cyc}$+5	—	ns	35.44
$\overline{\text{RAS}}$ 延迟时间 1	t_{RASD1}	1	13	ns	35.22 ～ 35.38
$\overline{\text{RAS}}$ 延迟时间 2	t_{RASD2}	$1/2t_{cyc}$	$1/2t_{cyc}$+13	ns	35.39、 35.40
$\overline{\text{CAS}}$ 延迟时间 1	t_{CASD1}	1	13	ns	35.22 ～ 35.38
$\overline{\text{CAS}}$ 延迟时间 2	t_{CASD2}	$1/2t_{cyc}$	$1/2t_{cyc}$+13	ns	35.39、 35.40
DQM 延迟时间 1	t_{DQMD1}	1	13	ns	35.22 ～ 35.35
DQM 延迟时间 2	t_{DQMD2}	$1/2t_{cyc}$	$1/2t_{cyc}$+13	ns	35.39、 35.40
CKE 延迟时间 1	t_{CKED1}	1	13	ns	35.37
CKE 延迟时间 2	t_{CKED2}	$1/2t_{cyc}$	$1/2t_{cyc}$+13	ns	35.40
$\overline{\text{AH}}$ 延迟时间	t_{AHD}	$1/2t_{cyc}$	$1/2t_{cyc}$+13	ns	35.17
多路复用地址延迟时间	t_{MAD}	—	13	ns	35.17
多路复用地址保持时间	t_{MAH}	1	—	ns	35.17
DACK、 TEND 延迟时间	t_{DACD}	—	参考 DMAC 时序	ns	35.13 ～ 35.35、 35.39、 35.41 ～ 35.44
$\overline{\text{FRAME}}$ 延迟时间	t_{FMD}	0	13	ns	35.18
$\overline{\text{ICIORD}}$ 延迟时间	t_{ICRSD}	—	$1/2t_{cyc}$+13	ns	35.43、 35.44
$\overline{\text{ICIOWR}}$ 延迟时间	t_{ICWSD}	—	$1/2t_{cyc}$+13	ns	35.43、 35.44

【注】 *1　Bϕ（外部总线时钟）的 f_{max}，请根据所使用的系统配置，对照等待数进行研究。

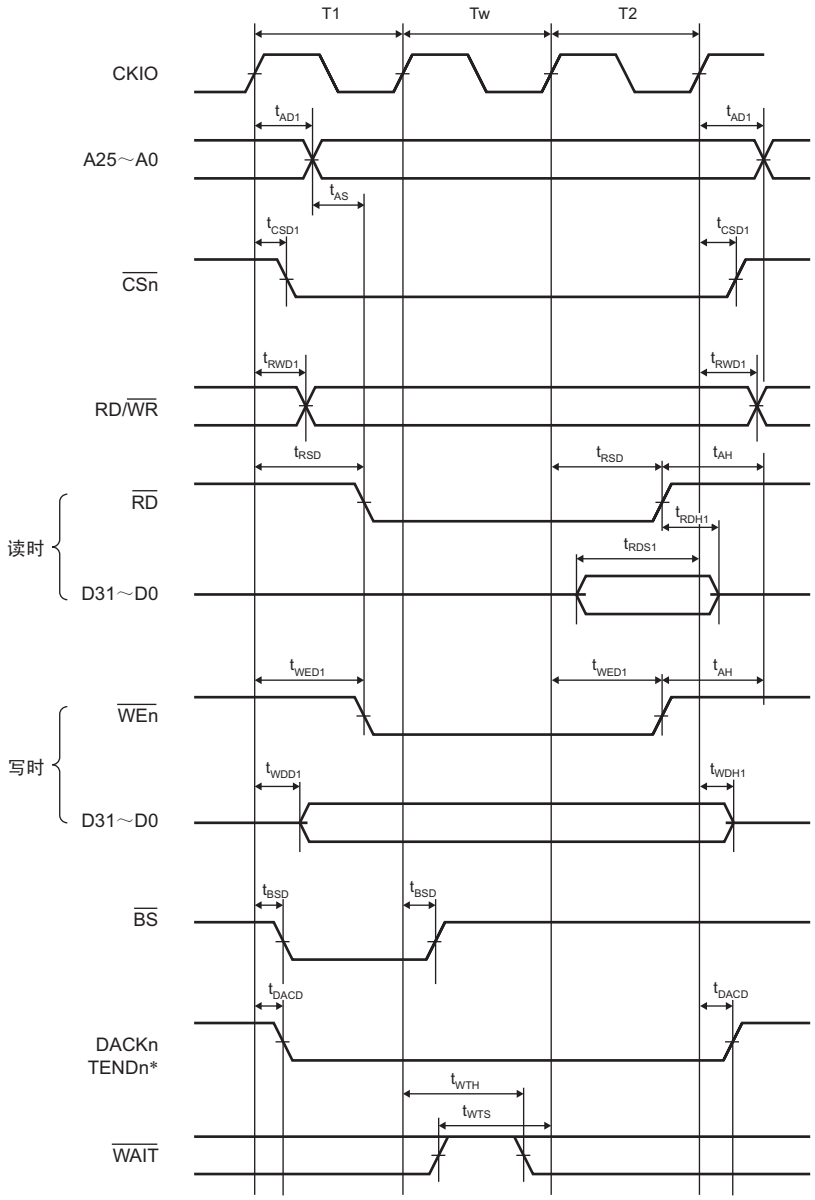
*2　为使用 SDRAM 时的值。

*3　延迟时间与设置、保持时间的 $1/2t_{cyc}$ 的记述，表示从时钟上升开始 1/2 周期，即：为时钟的下降基准。



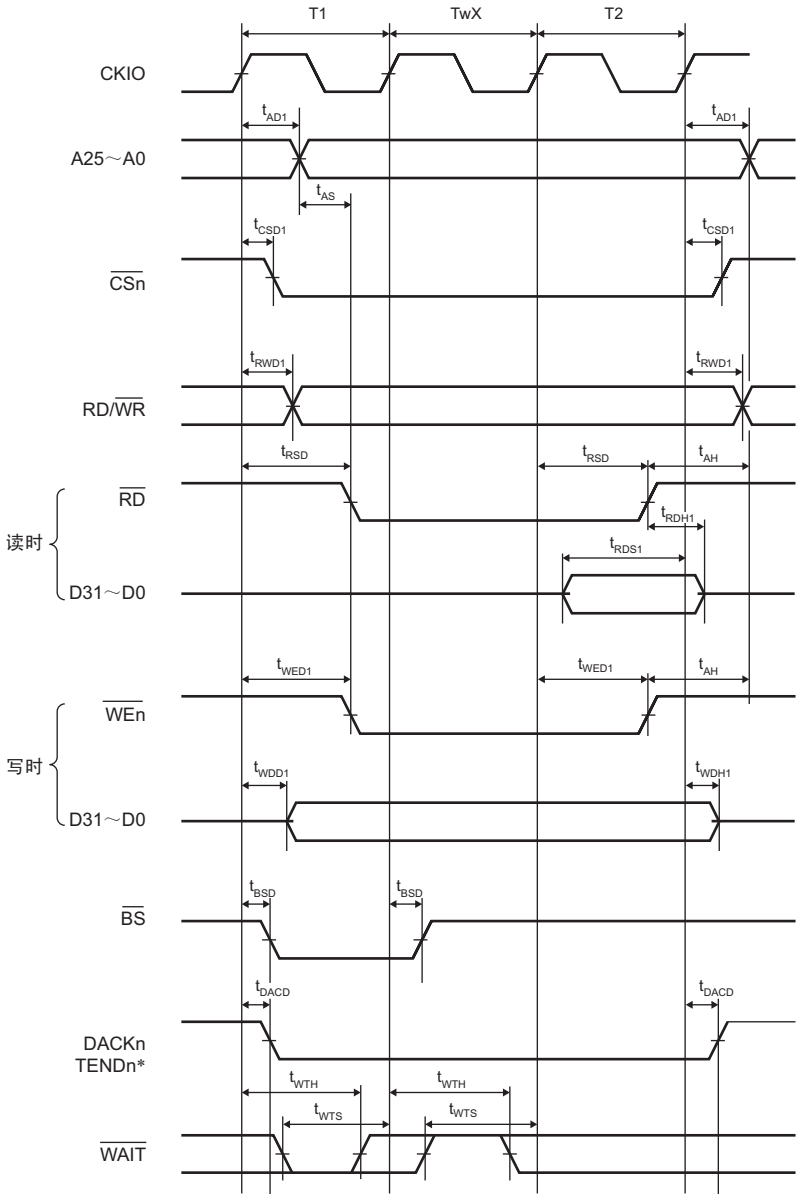
【注】* DACKn、TENDn为指定低电平有效时的波形

图 35.13 正常空间基本总线周期 （无等待）



【注】* DACKn、TENDn为指定低电平有效时的波形

图 35.14 正常空间基本总线周期 （1 个软件等待）



【注】* DACKn、TENDn为指定低电平有效时的波形

图 35.15 正常空间基本总线周期 （插入 1 个外部等待）

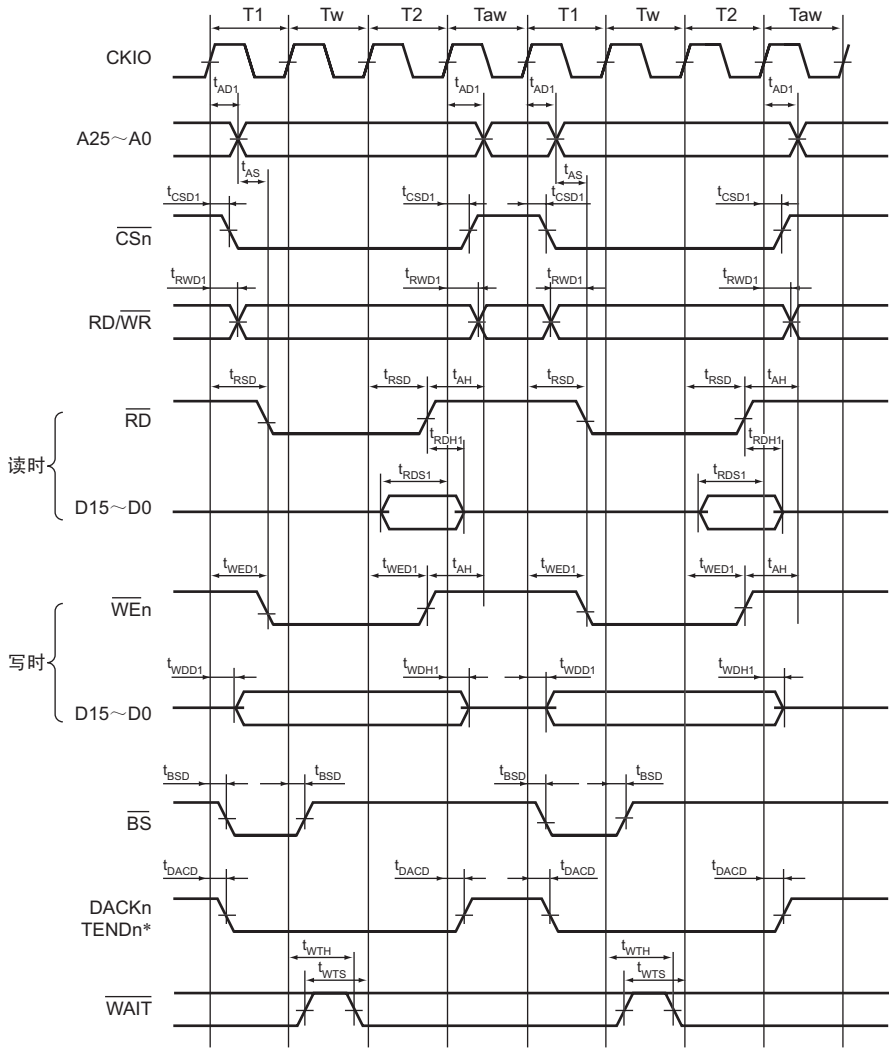


图 35.16 正常空间基本总线周期
（1 个软件等待、外部等待有效 （WM 位 =0）、无空闲周期）

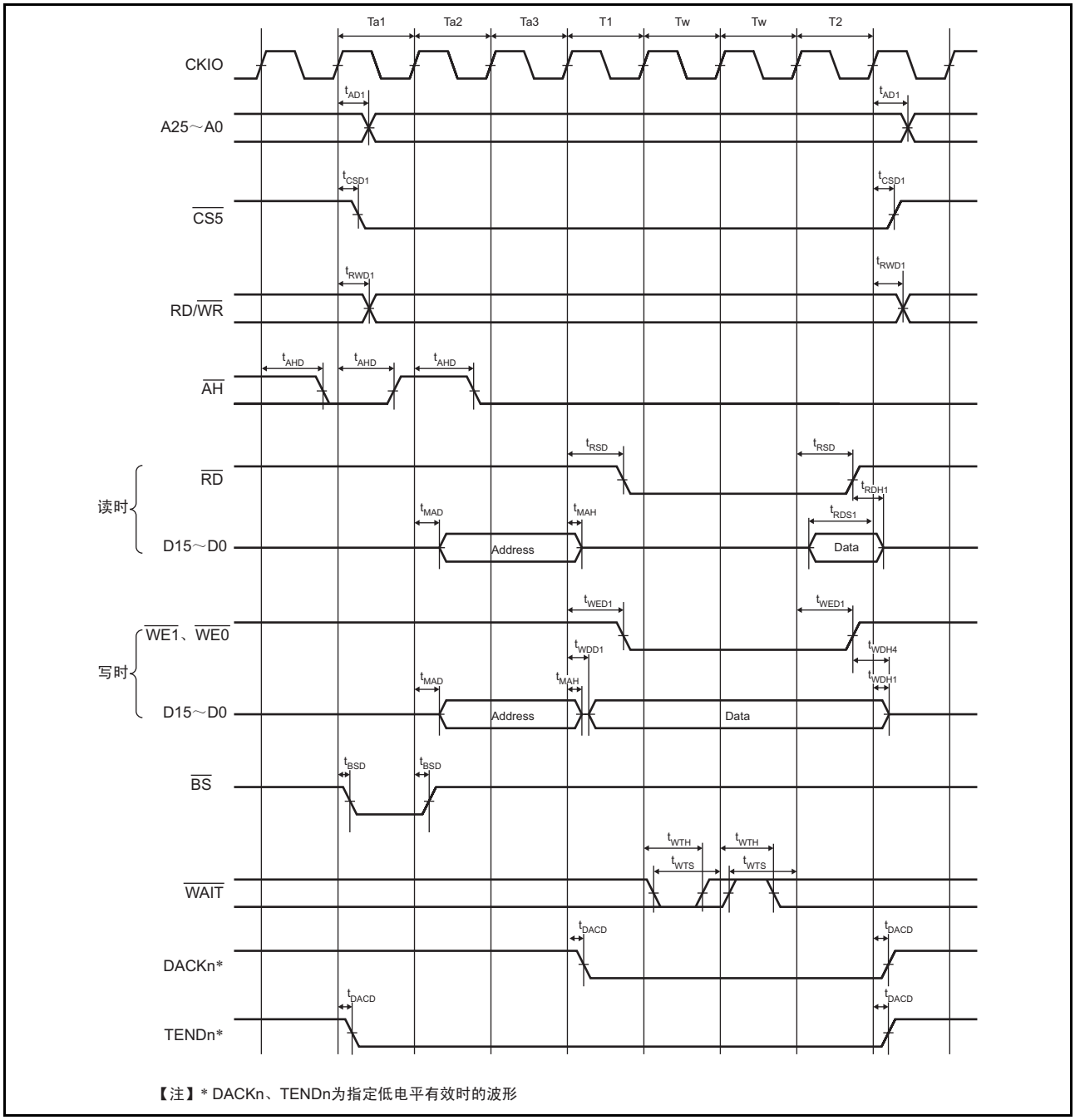
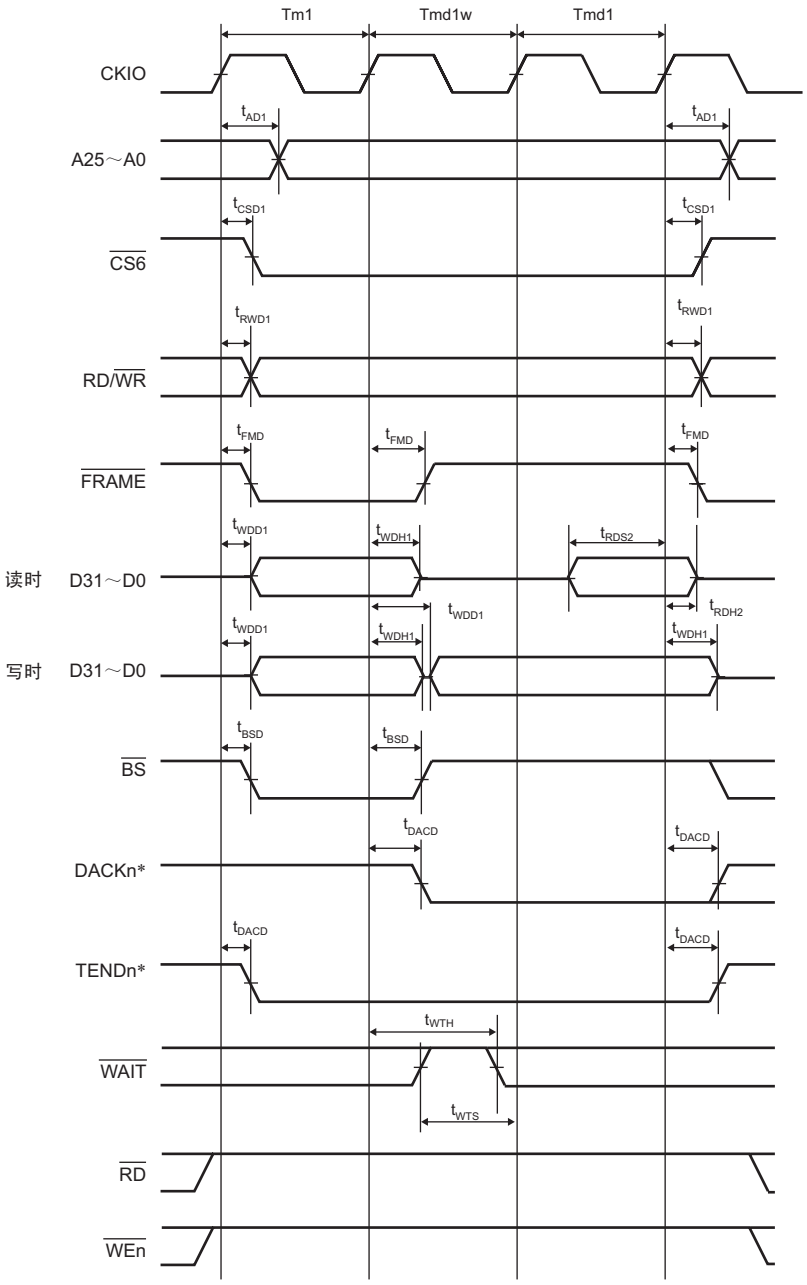


图 35.17 MPX-I/O 接口总线周期
（插入 3 个地址周期、 1 个软件等待、 1 个外部等待）



【注】* DACKn、TENDn为指定低电平有效时的波形

图 35.18 突发 MPX-I/O 接口总线周期单一读写
（1 个地址周期、 1 个软件等待）

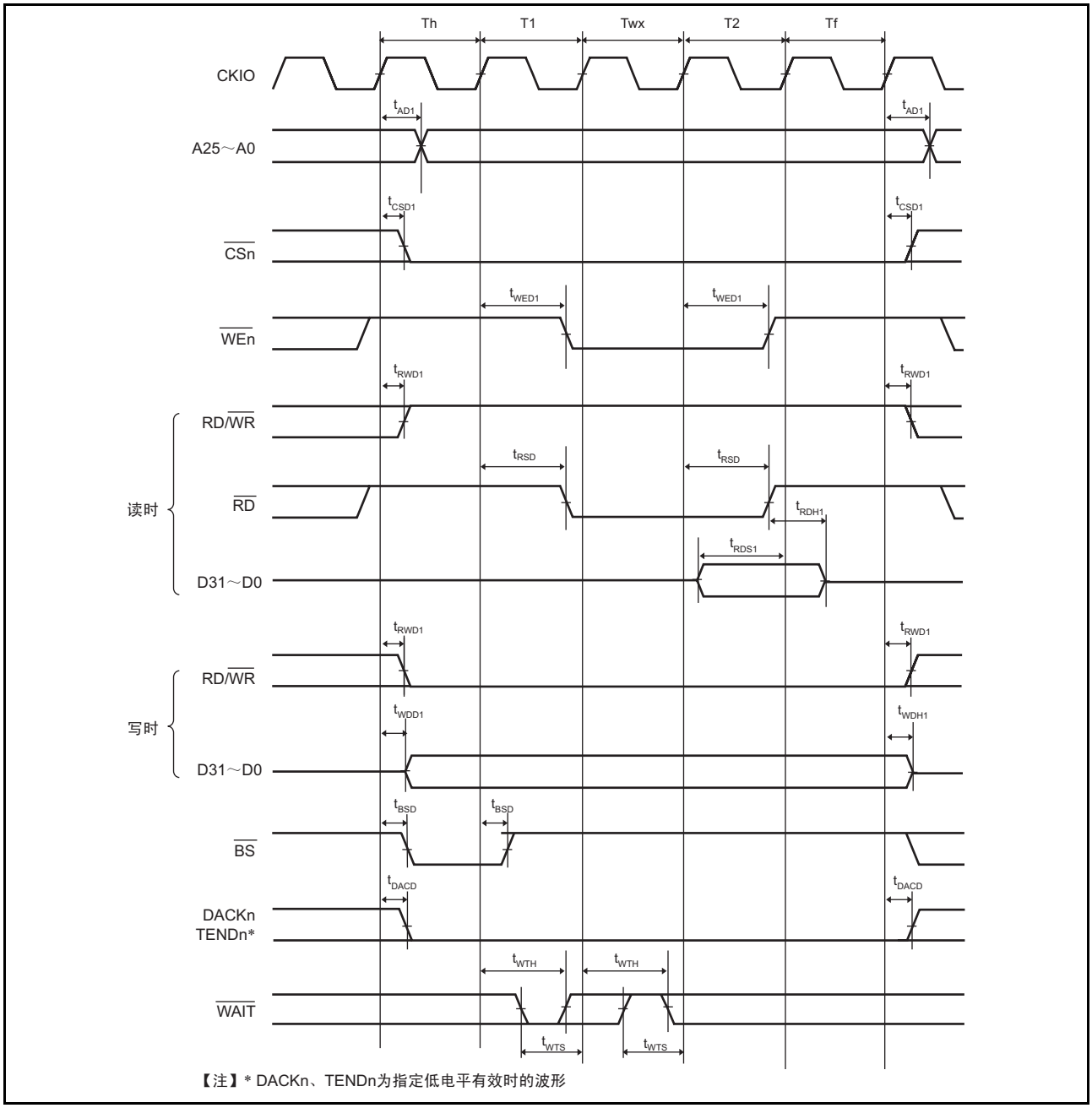


图 35.19 带字节选择的 SRAM 总线周期
（SW=1 个周期、 HW=1 个周期、插入 1 个异步外部等待、 BAS=0 （写周期 UB/LB 控制））

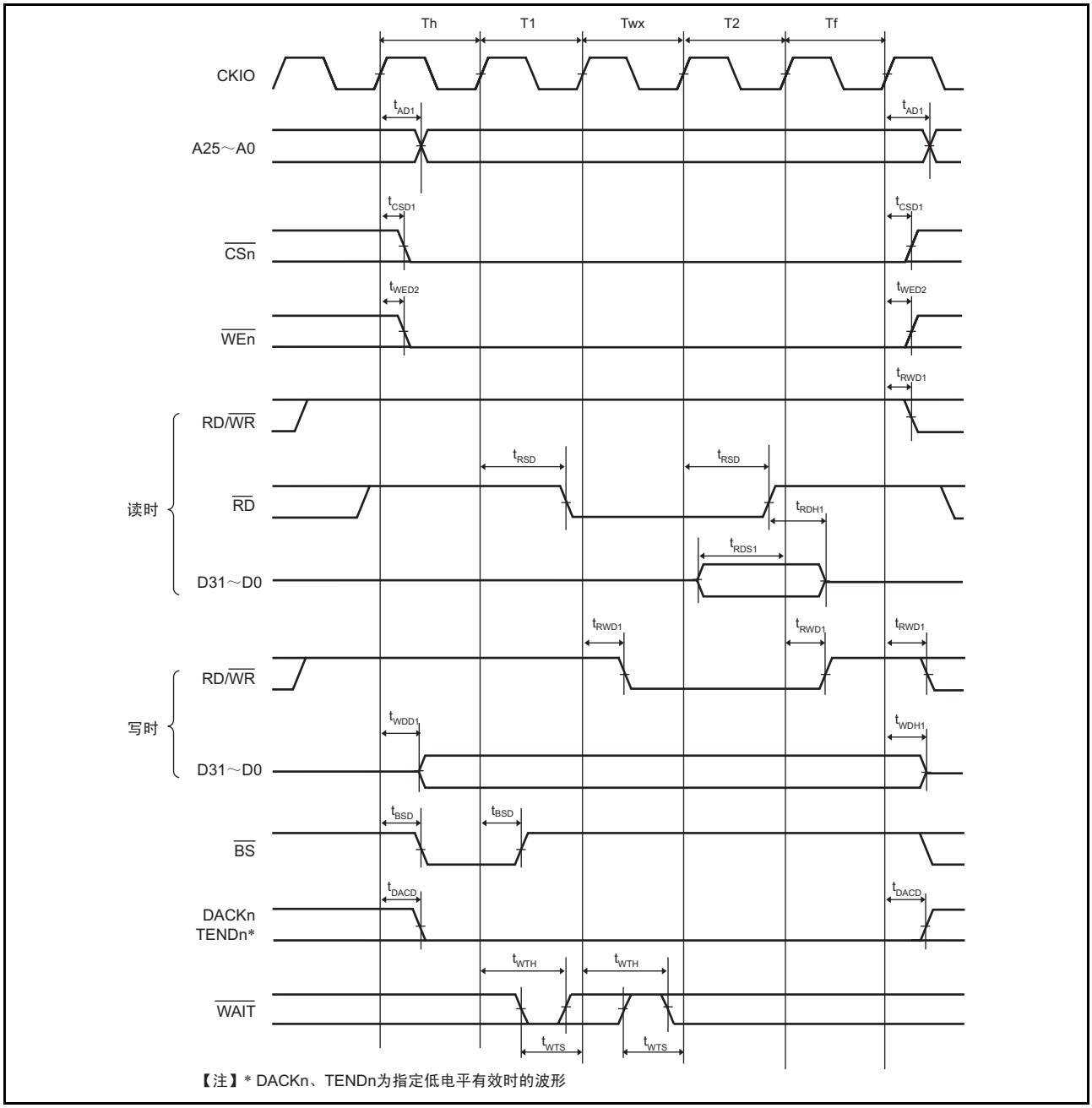


图 35.20 带字节选择的 SRAM 总线周期
（SW=1 个周期、 HW=1 个周期、插入 1 个异步外部等待、 BAS=1 （写周期 WE 控制））

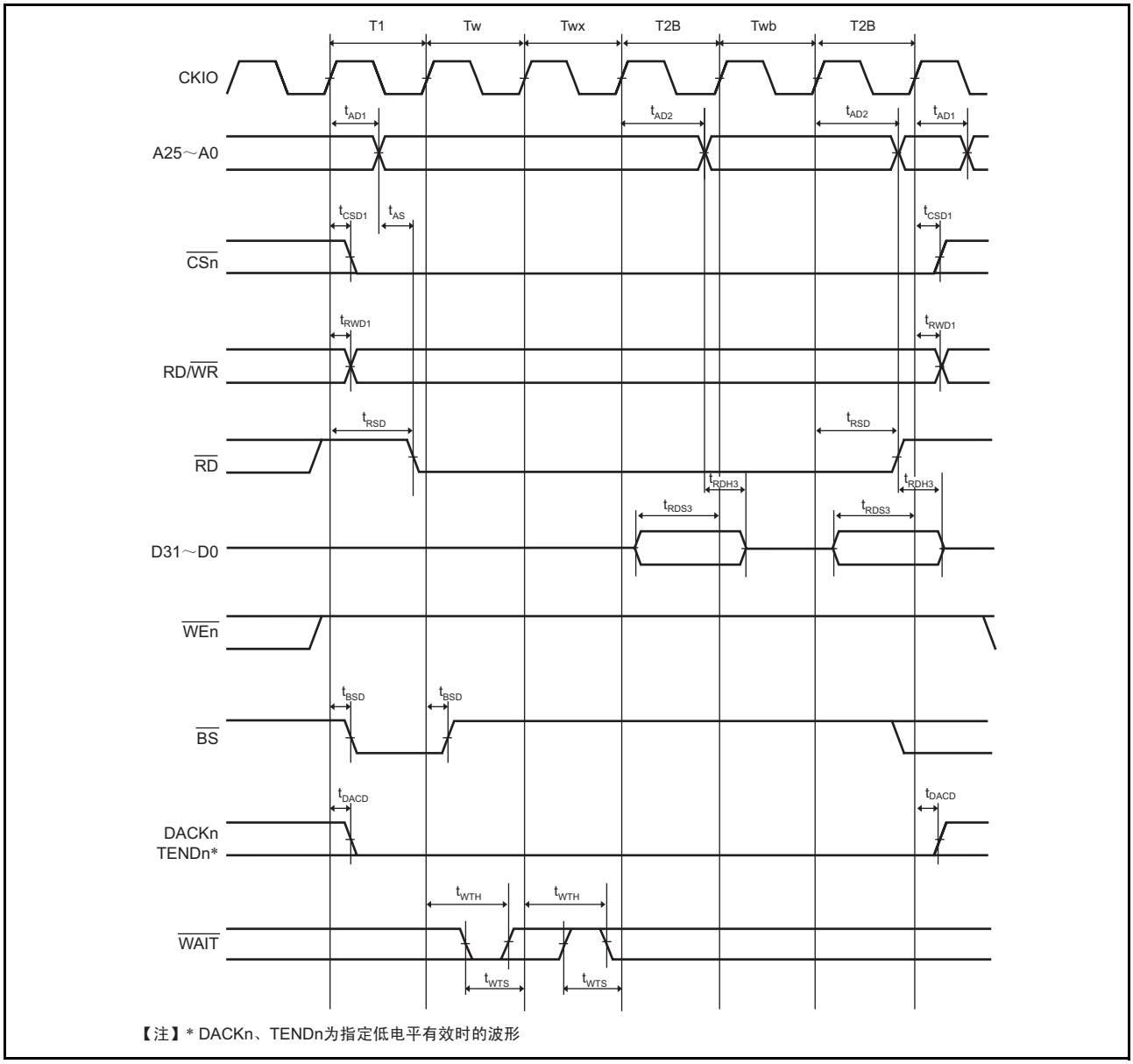


图 35.21 突发 ROM 读周期
（1 个软件等待、插入 1 个异步外部等待、 1 个突发等待、 2 个突发）

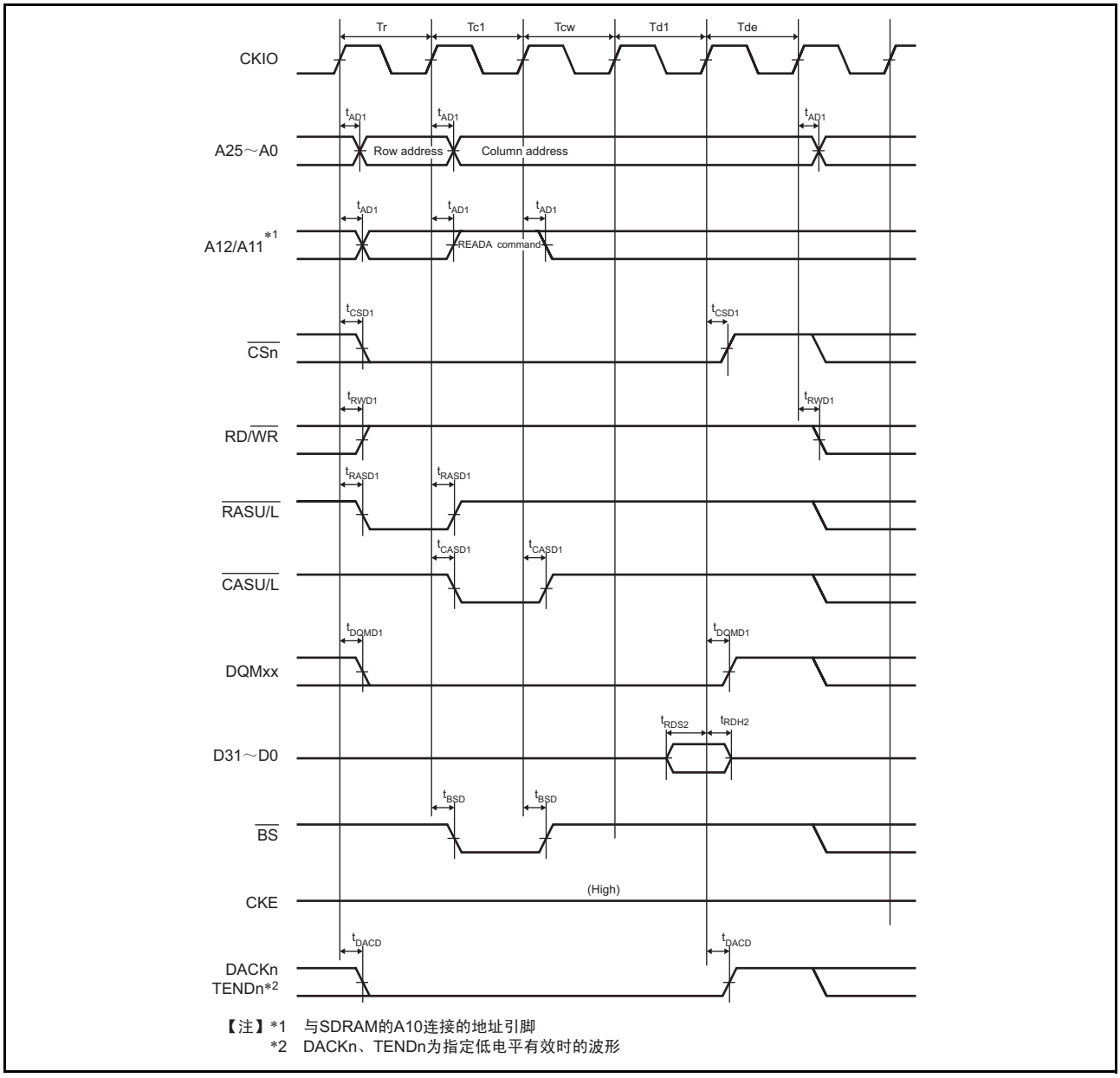


图 35.22 同步 DRAM 单一读总线周期
（有自动预充电、 CAS 等待时间 2、 WTRCD=0 周期、 WTRP=0 周期）

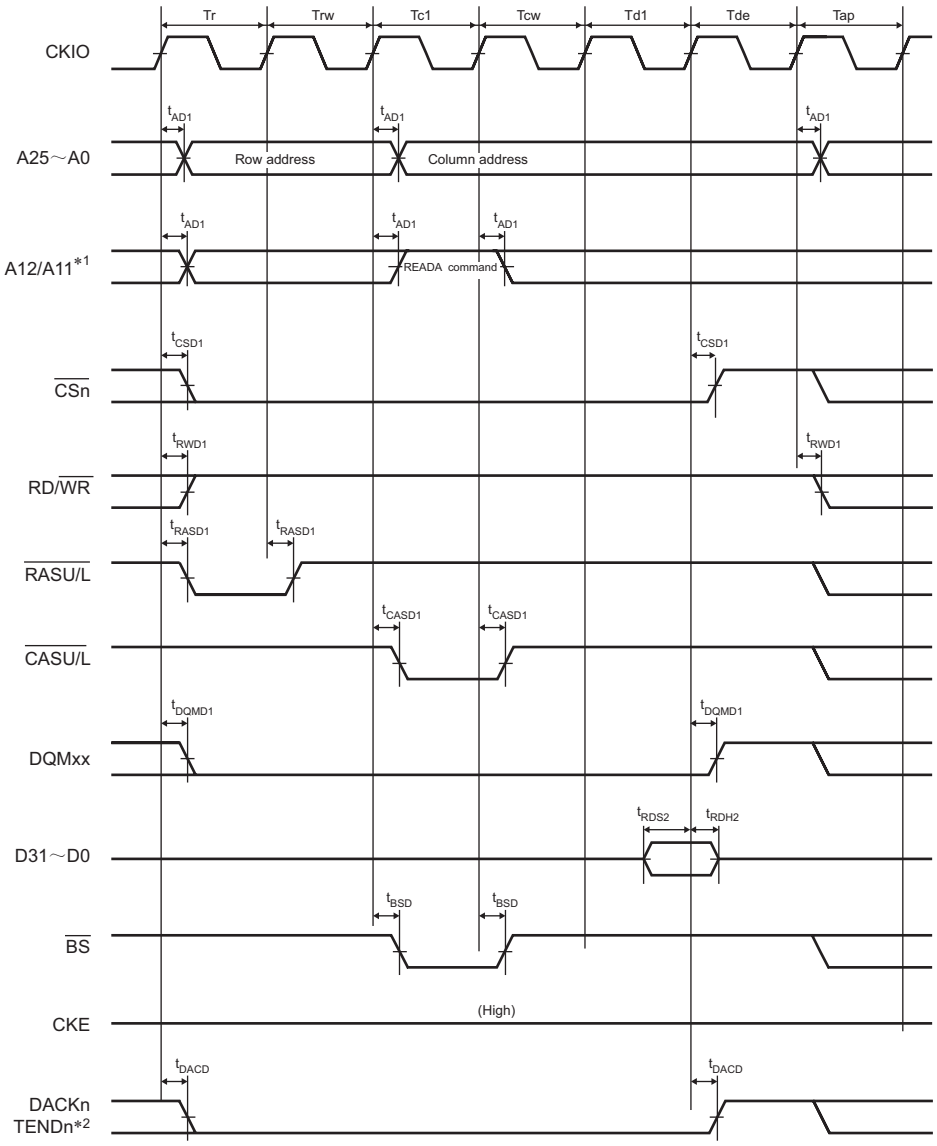


图 35.23 同步 DRAM 单一读总线周期
（有自动预充电、 CAS 等待时间 2、 WTRCD=1 个周期、 WTRP=1 个周期）

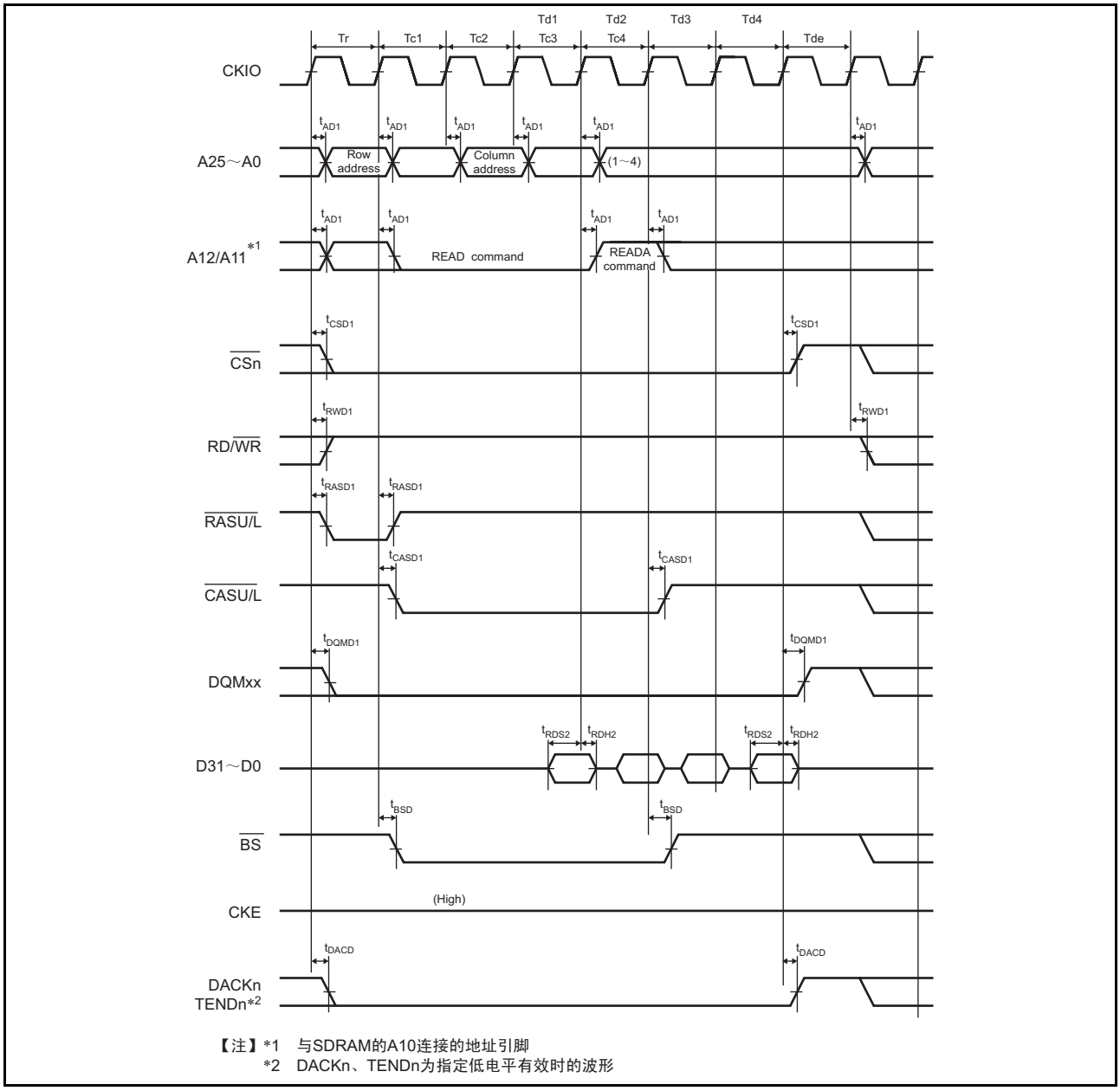
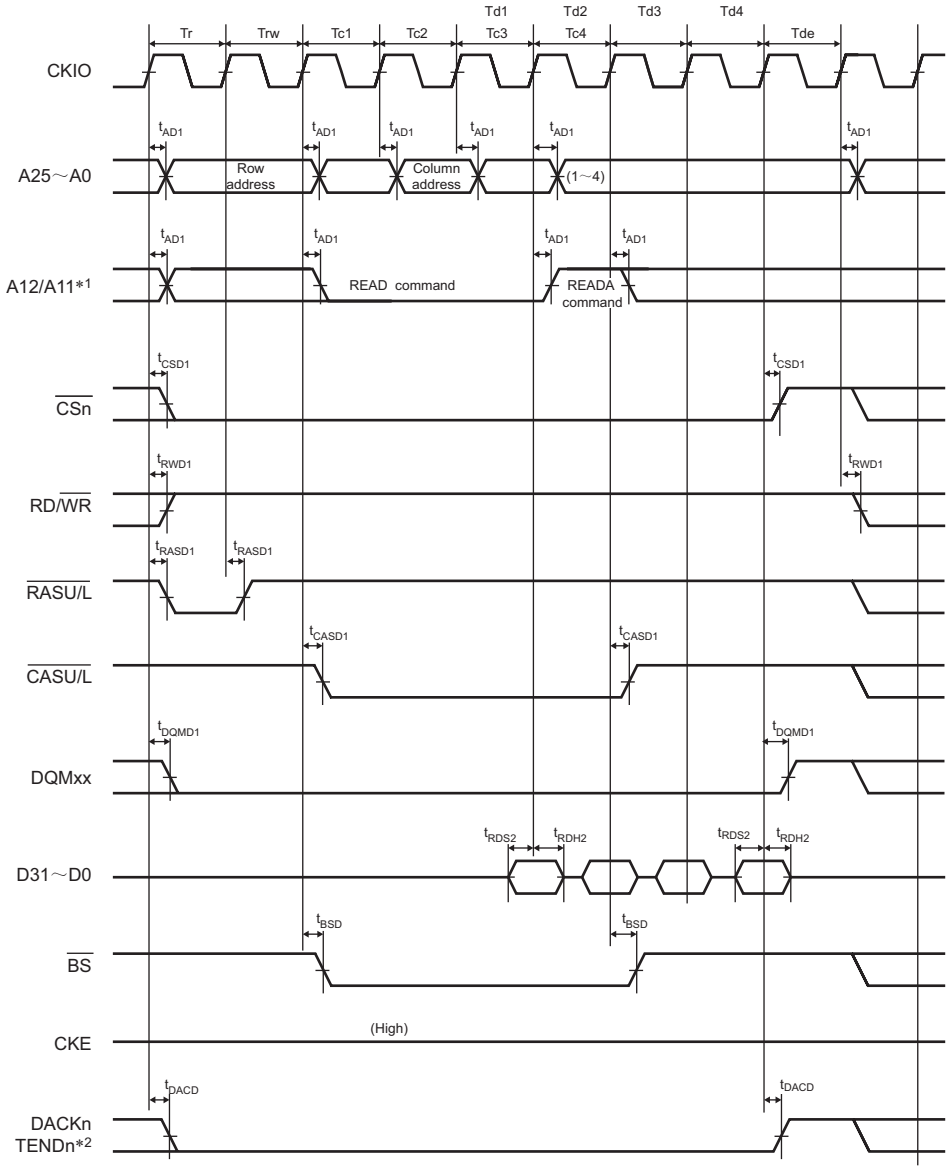
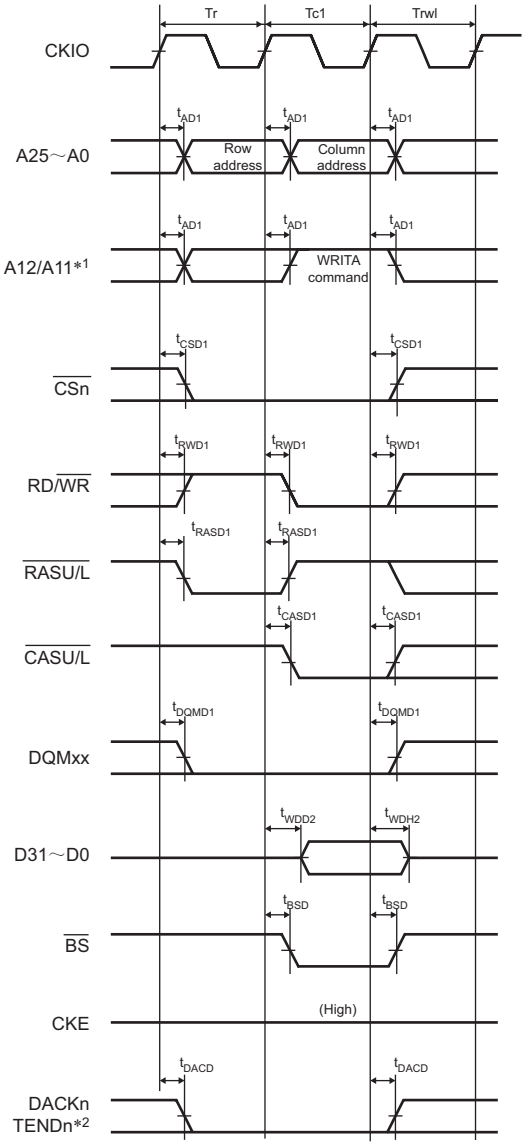


图 35.24 同步 DRAM 突发读总线周期 （读 4 个周期）
（有自动预充电、 CAS 等待时间 2、 WTRCD=0 周期、 WTRP=1 个周期）



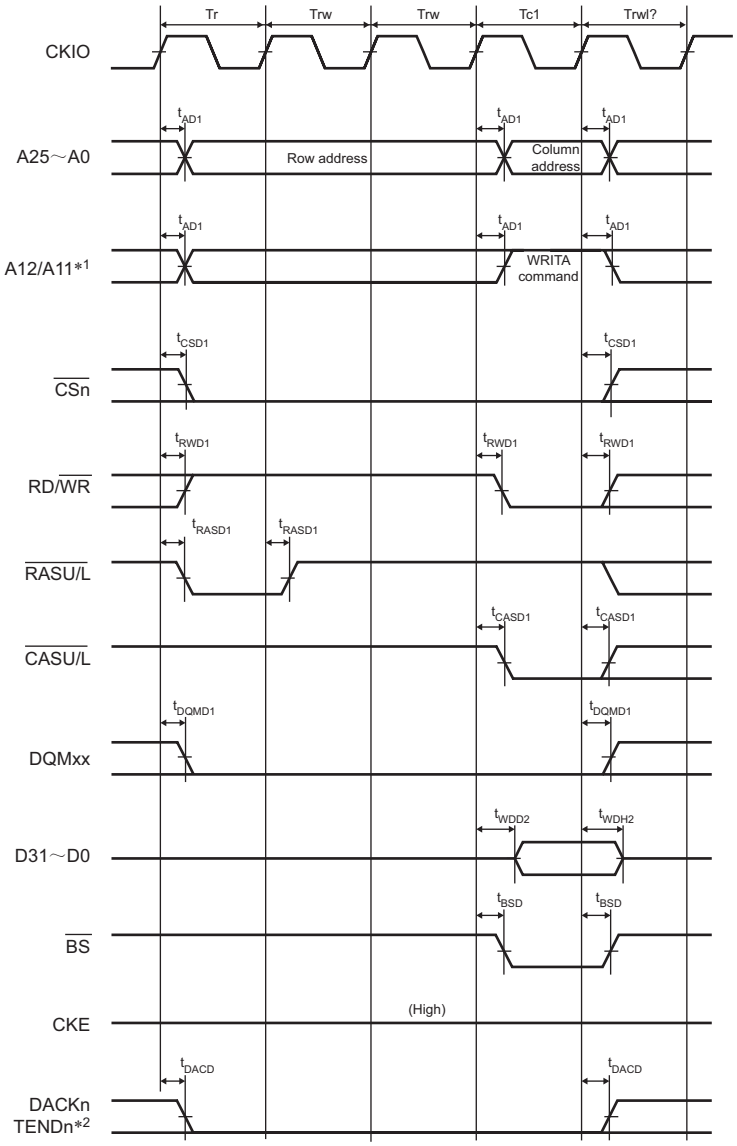
【注】*1 与SDRAM的A10连接的地址引脚
*2 DACKn、TENDn为指定低电平有效时的波形

图 35.25 同步 DRAM 突发读总线周期 （读 4 个周期）
（有自动预充电、 CAS 等待时间 2、 WTRCD=1 个周期、 WTRP=0 周期）



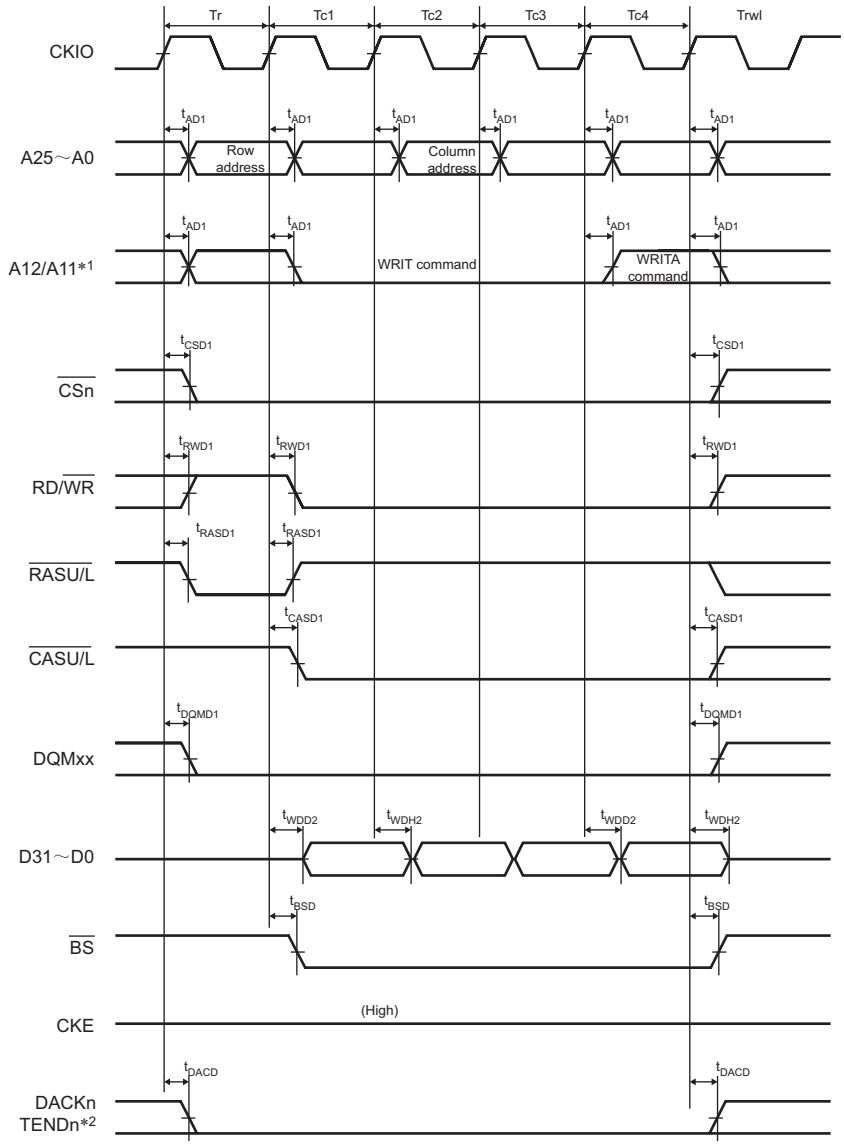
【注】*1 与SDRAM的A10连接的地址引脚
*2 DACKn、TENDn为指定低电平有效时的波形

图 35.26 同步 DRAM 单一写总线周期
（有自动预充电、 TRWL=1 个周期）



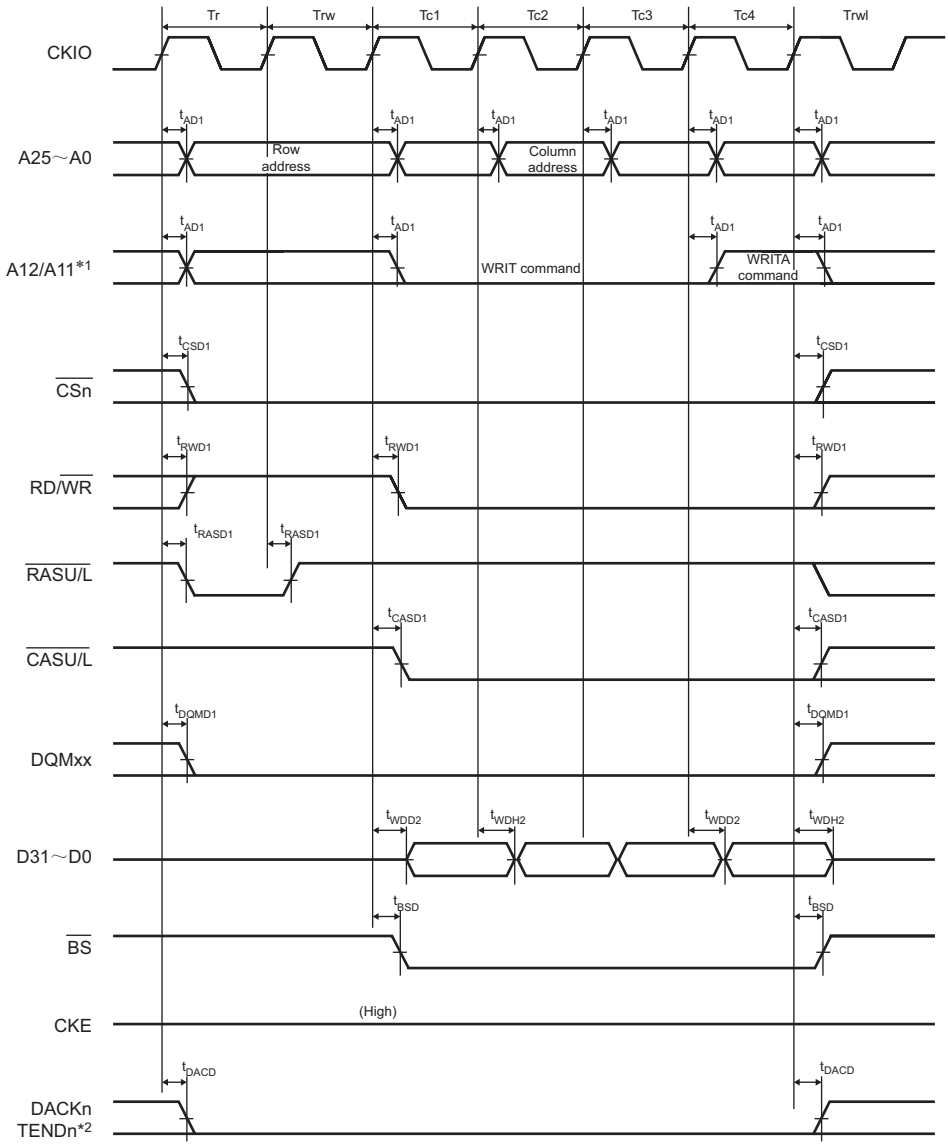
【注】*1 与SDRAM的A10连接的地址引脚
*2 DACKn、TENDn为指定低电平有效时的波形

图 35.27 同步 DRAM 单一写总线周期
（有自动预充电、 WTRCD=2 个周期、 TRWL=1 个周期）



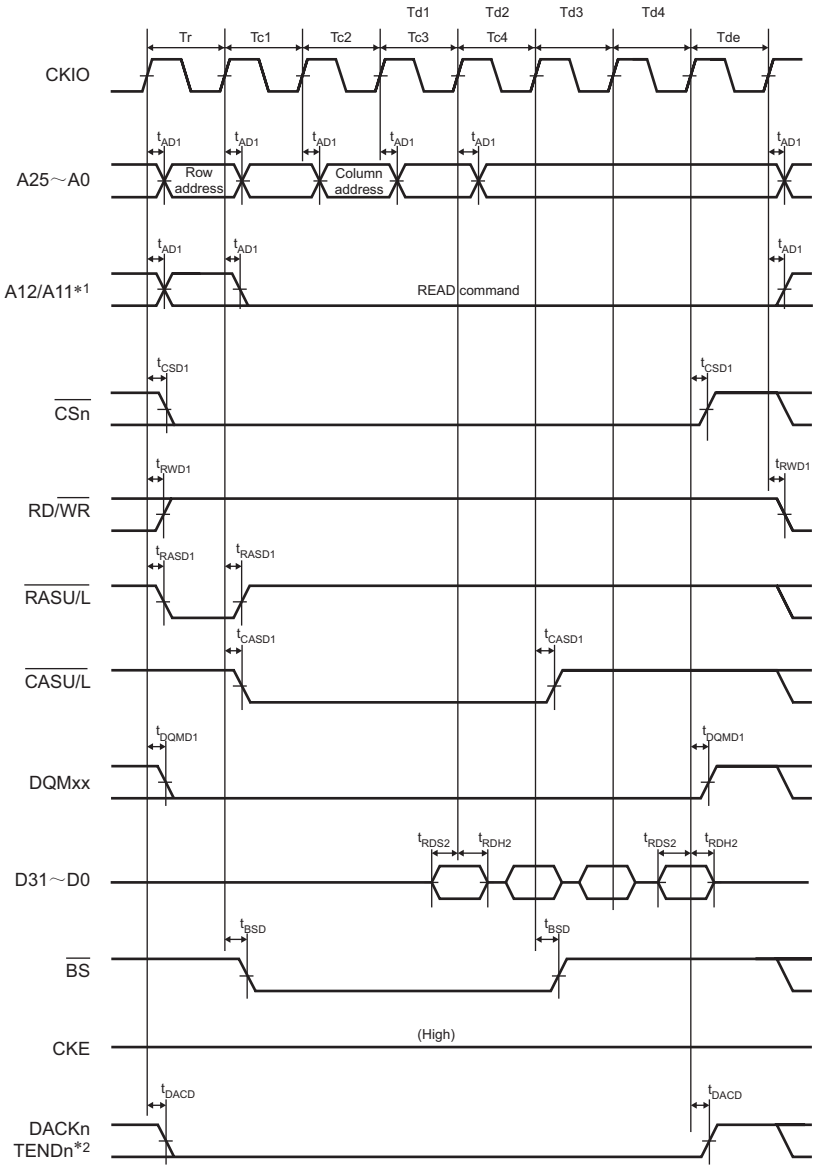
【注】*1 与SDRAM的A10连接的地址引脚
*2 DACKn、TENDn为指定低电平有效时的波形

图 35.28 同步 DRAM 单一写总线周期 （写 4 个周期）
（有自动预充电、 WTRCD=0 周期、 TRWL=1 个周期）



【注】*1 与SDRAM的A10连接的地址引脚
*2 DACKn、TENDn为指定低电平有效时的波形

图 35.29 同步 DRAM 突发写总线周期 （写 4 个周期）
（有自动预充电、 WTRCD=1 个周期、 TRWL=1 个周期）



【注】*1 与SDRAM的A10连接的地址引脚
*2 DACKn、TENDn为指定低电平有效时的波形

图 35.30 同步 DRAM 突发写总线周期 （写 4 个周期）
（存储体有效模式：ACT+READ 命令、 CAS 等待时间 2、 WTRCD=0 周期）

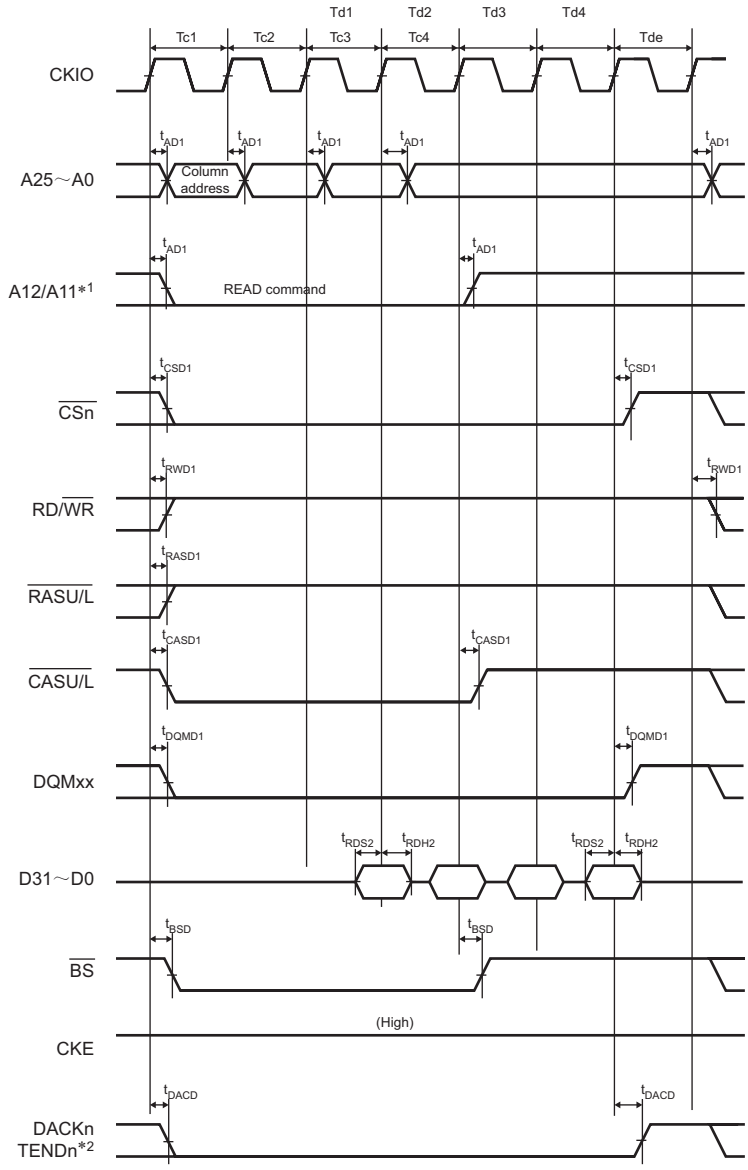
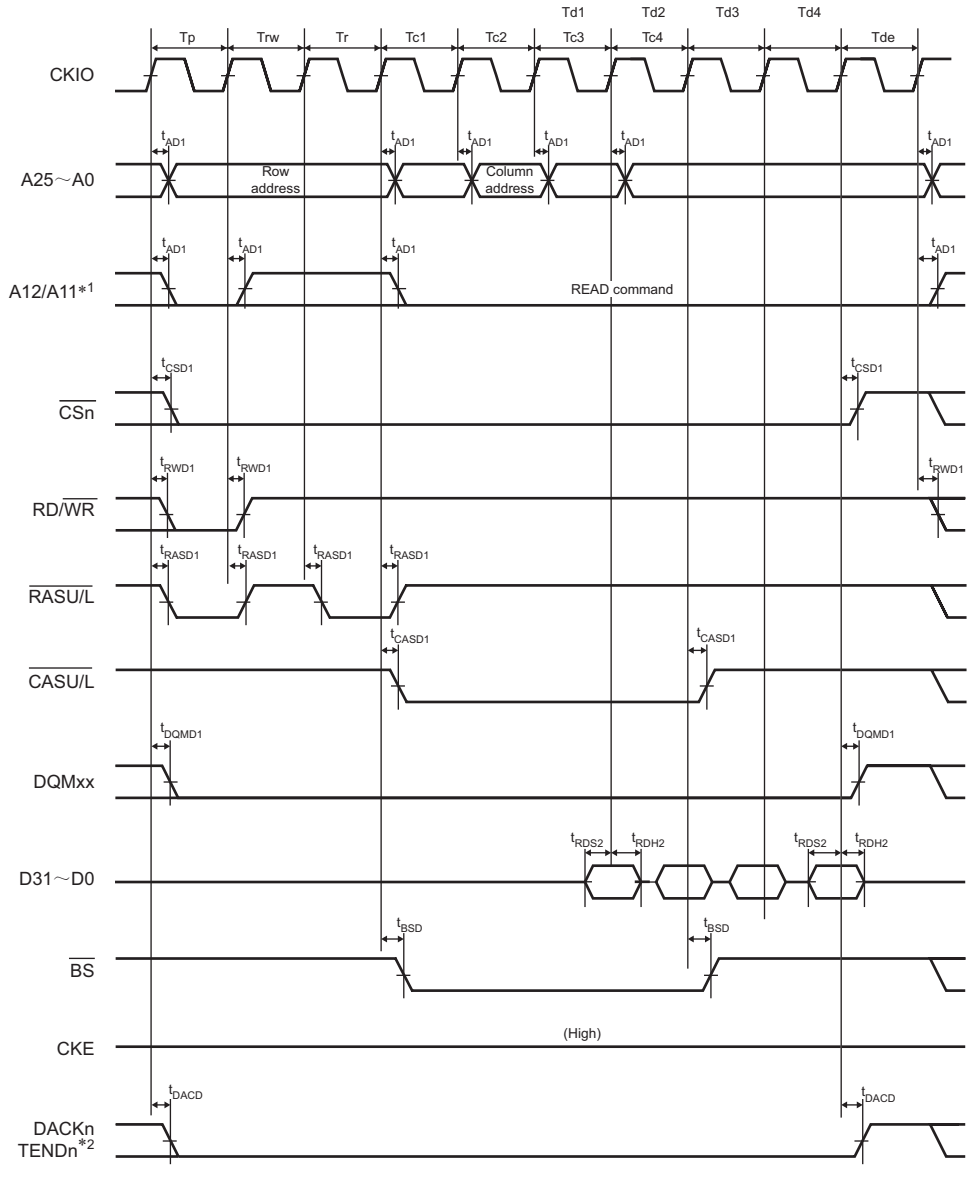
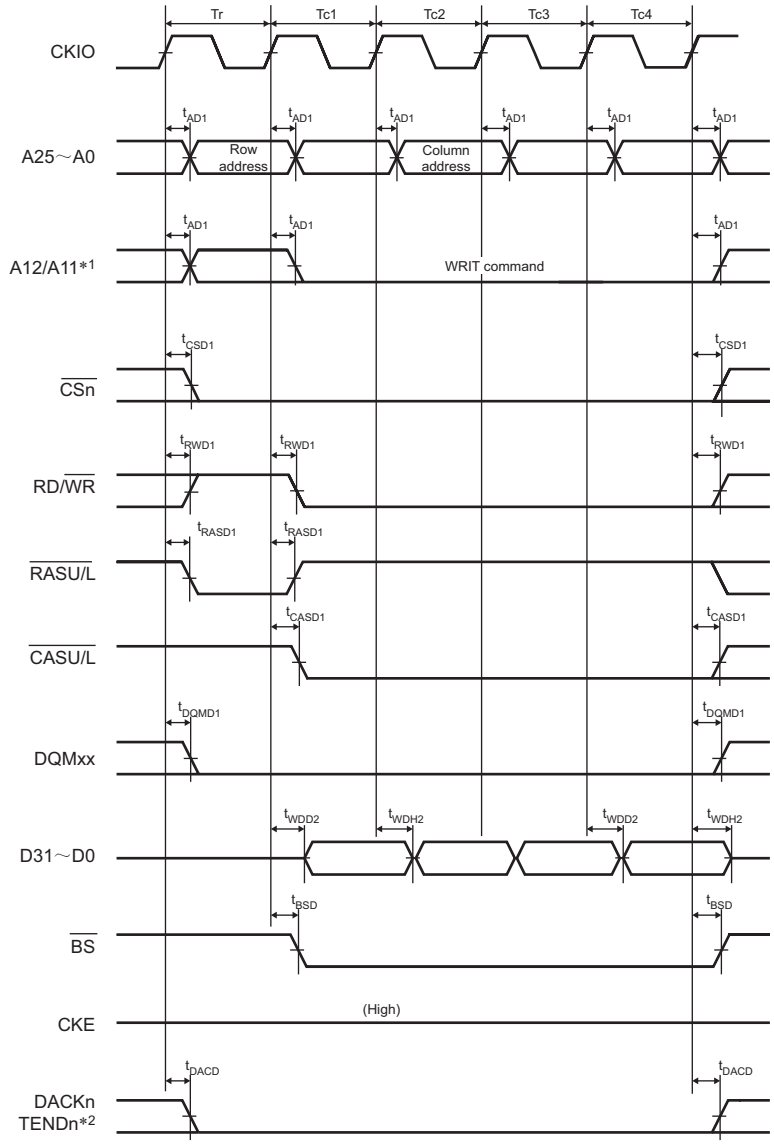


图 35.31 同步 DRAM 突发写总线周期 （写 4 个周期）
（存储体有效模式：READ 命令、相同的行地址、 CAS 等待时间 2、 WTRCD=0 周期）



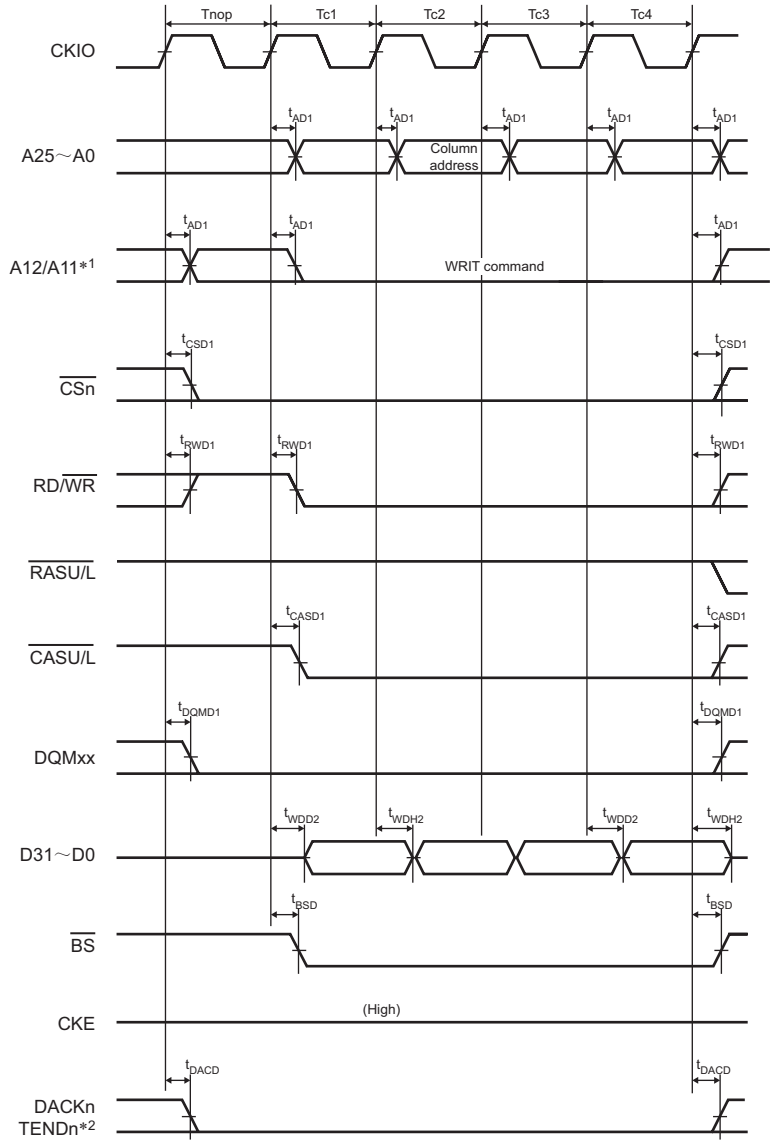
【注】*1 与SDRAM的A10连接的地址引脚
*2 DACKn、TENDn为指定低电平有效时的波形

图 35.32 同步 DRAM 突发写总线周期 （写 4 个周期）
（存储体有效模式：PRE+ACT+READ 命令、不同的行地址、 CAS 等待时间 2、 WTRCD=0 周期）



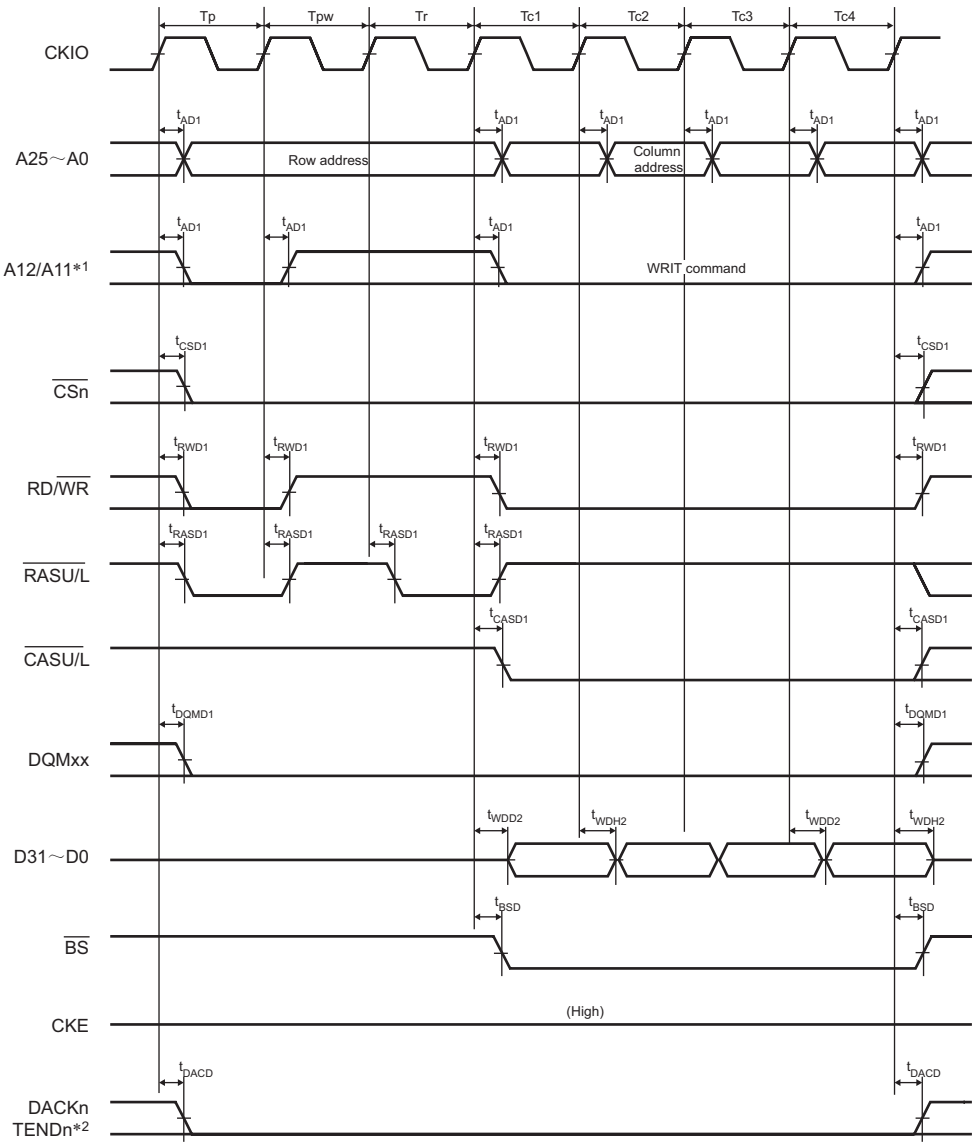
【注】*1 与SDRAM的A10连接的地址引脚
*2 DACKn、TENDn为指定低电平有效时的波形

图 35.33 同步 DRAM 突发写总线周期 （写 4 个周期）
（存储体有效模式：ACT+WRITE 命令、 WTRCD=0 周期、 TRWL=0 周期）



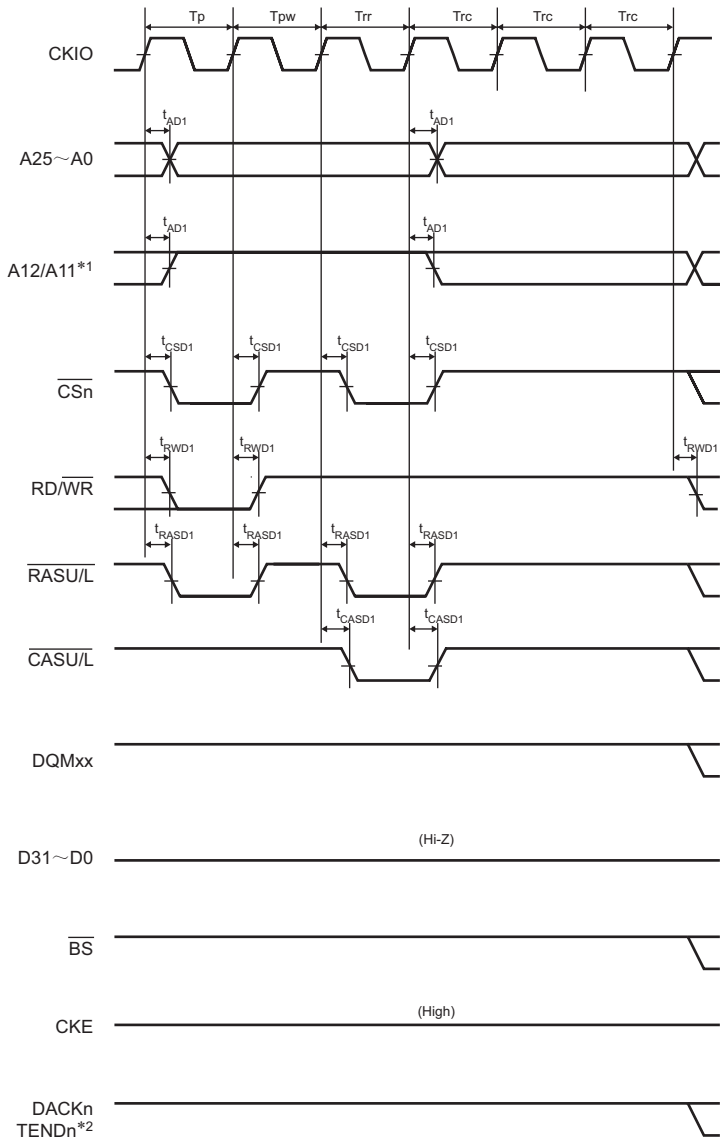
【注】*1 与SDRAM的A10连接的地址引脚
*2 DACKn、TENDn为指定低电平有效时的波形

图 35.34 同步 DRAM 突发写总线周期 （写 4 个周期）
（存储体有效模式：WRITE 命令、相同的行地址、 WTRCD=0 周期、 TRWL=0 周期）



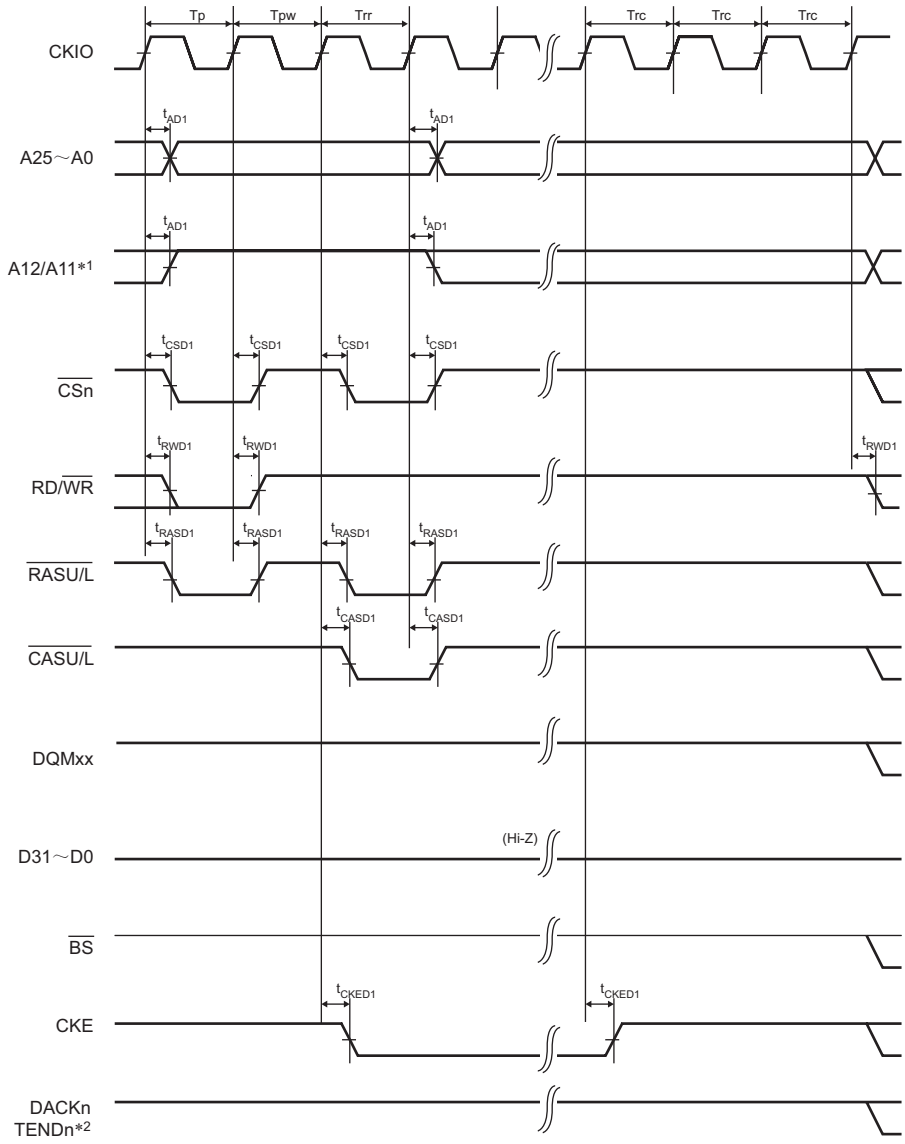
【注】*1 与SDRAM的A10连接的地址引脚
*2 DACKn、TENDn为指定低电平有效时的波形

图 35.35 同步 DRAM 突发写总线周期 （写 4 个周期）
（存储体有效模式：PRE+ACT+WRITE 命令、不同的行地址、 WTRCD=0 周期、 TRWL=0 周期）



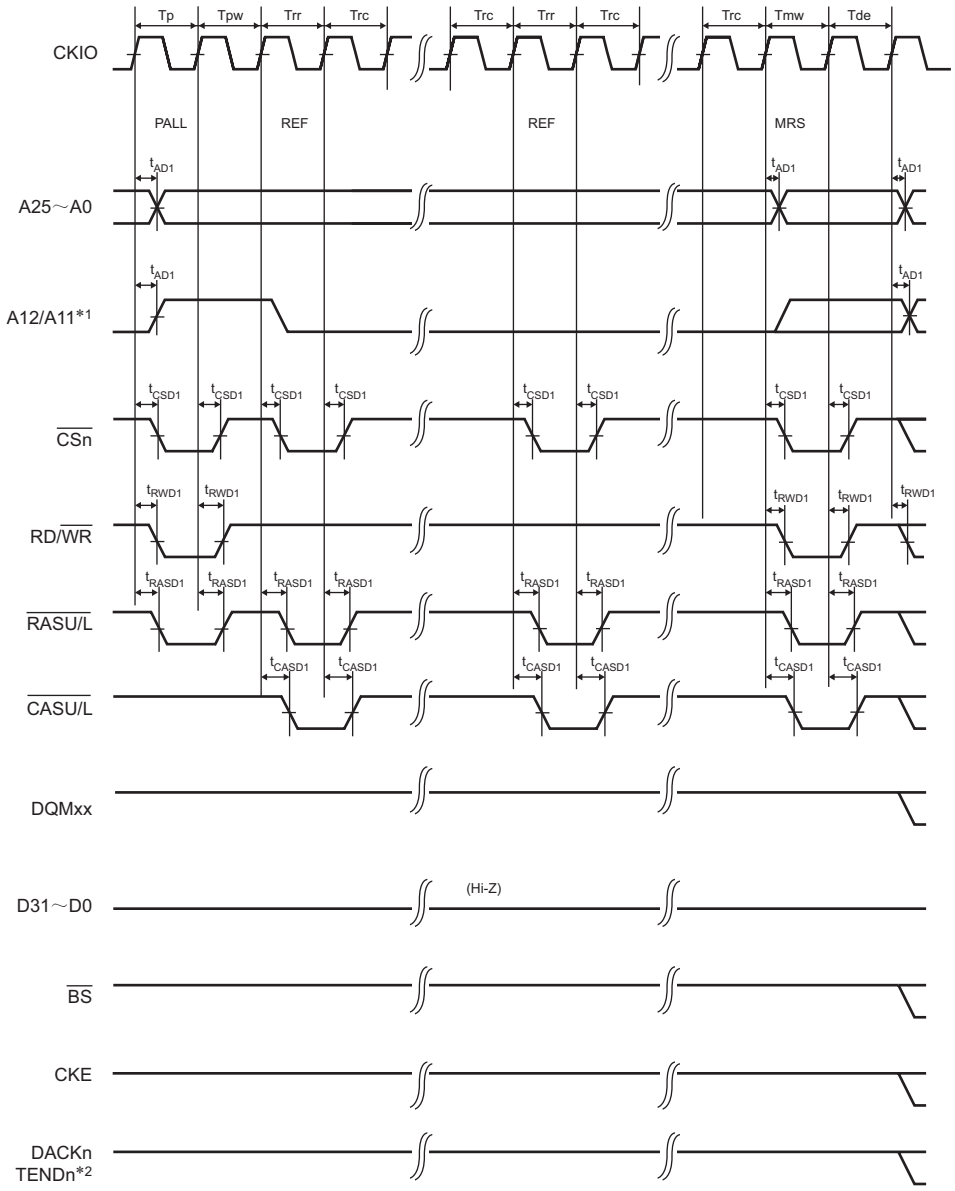
【注】*1 与SDRAM的A10连接的地址引脚
*2 DACKn、TENDn为指定低电平有效时的波形

图 35.36 同步 DRAM 自动刷新时序 （WTRP=1 个周期、 WTRC=3 个周期）



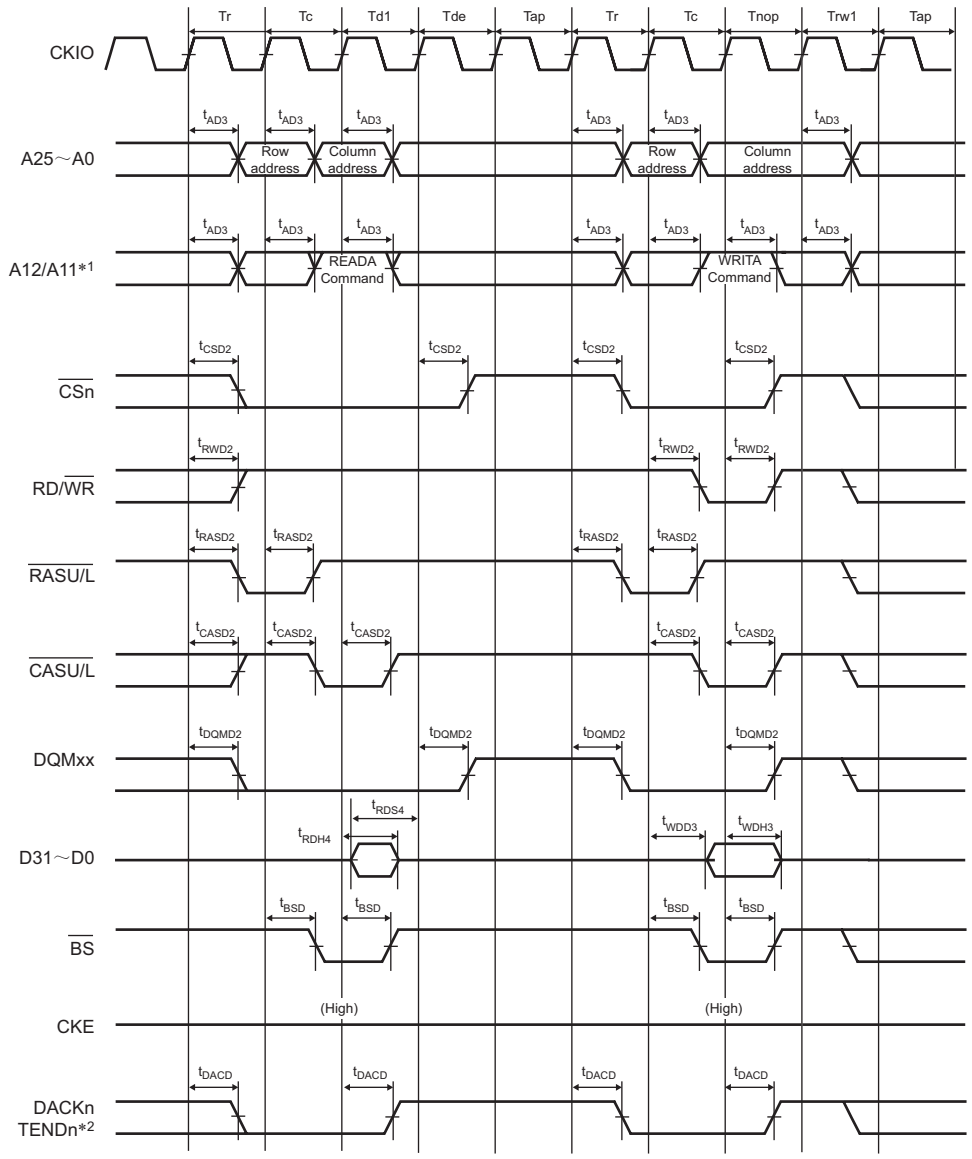
【注】*1 与SDRAM的A10连接的地址引脚
*2 DACKn、TENDn为指定低电平有效时的波形

图 35.37 同步 DRAM 自刷新时序 （WTRP=1 个周期）



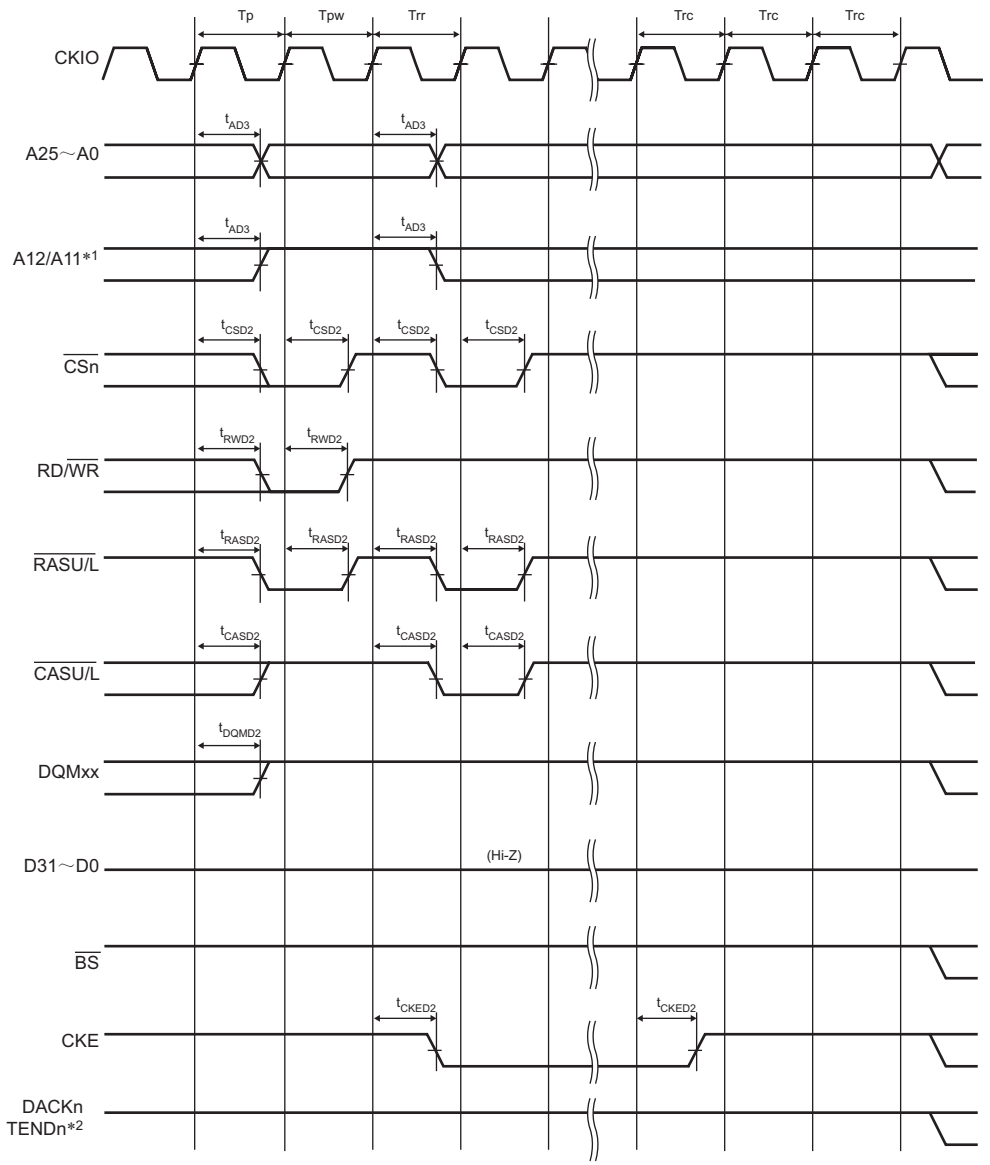
【注】*1 与SDRAM的A10连接的地址引脚
*2 DACKn、TENDn为指定低电平有效时的波形

图 35.38 同步 DRAM 模式寄存器写时序 （WTRP=1 个周期）



【注】*1 与SDRAM的A10连接的地址引脚
*2 DACKn、TENDn为指定低电平有效时的波形

图 35.39 同步 DRAM 低频模式下的存取时序
（自动预充电模式、 TRWL=2 个周期）



【注】*1 与SDRAM的A10连接的地址引脚
*2 DACKn、TENDn为指定低电平有效时的波形

图 35.40 同步 DRAM 低频模式自刷新时序 （WTRP=2 个周期）

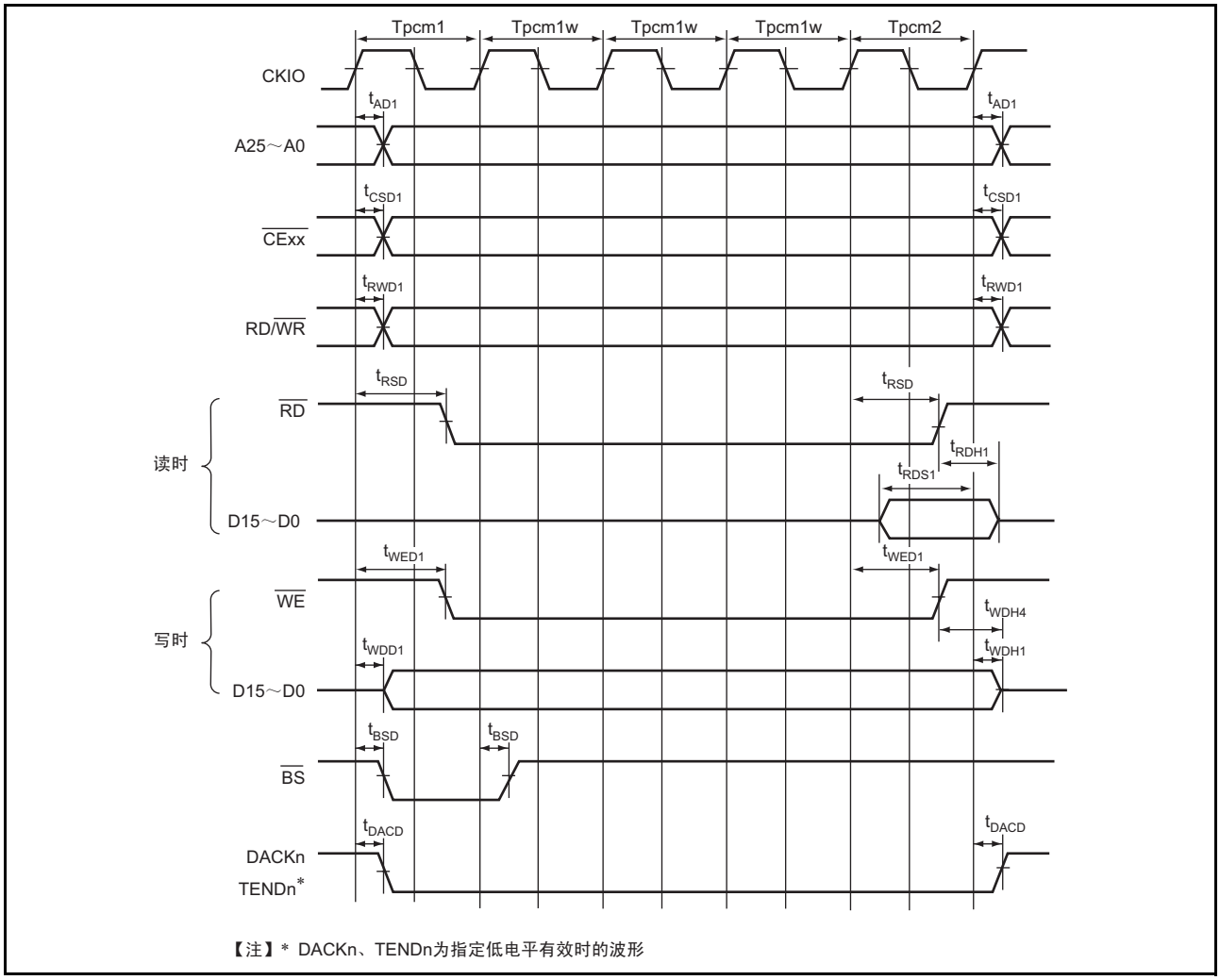


图 35.41 PCMCIA 存储卡总线周期 （TED=0 周期、 TEH=0 周期、无等待）

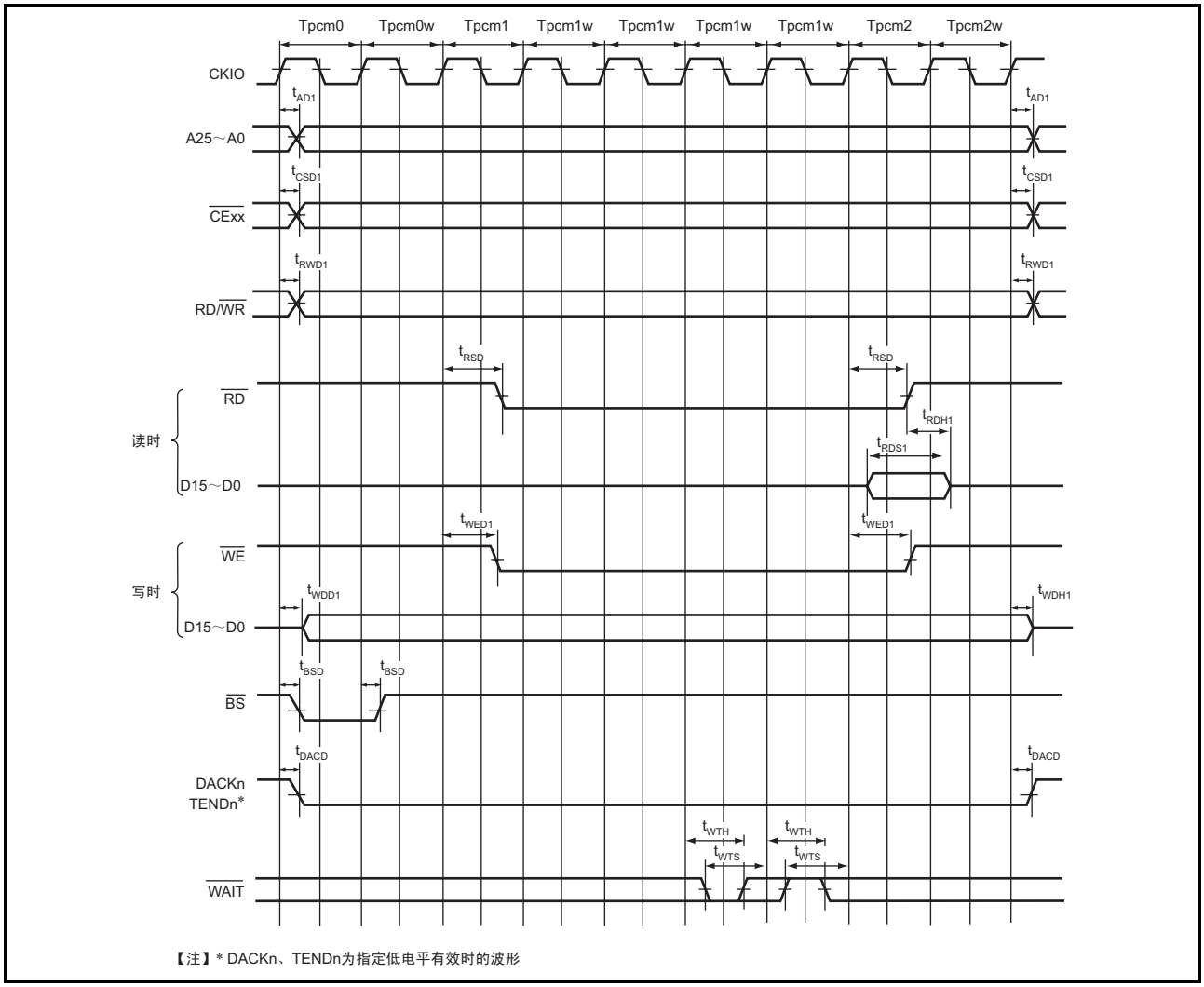


图 35.42 PCMCIA 存储卡总线周期
（TED=2 个周期、 TEH=1 个周期、软件等待 0、硬件等待 1）

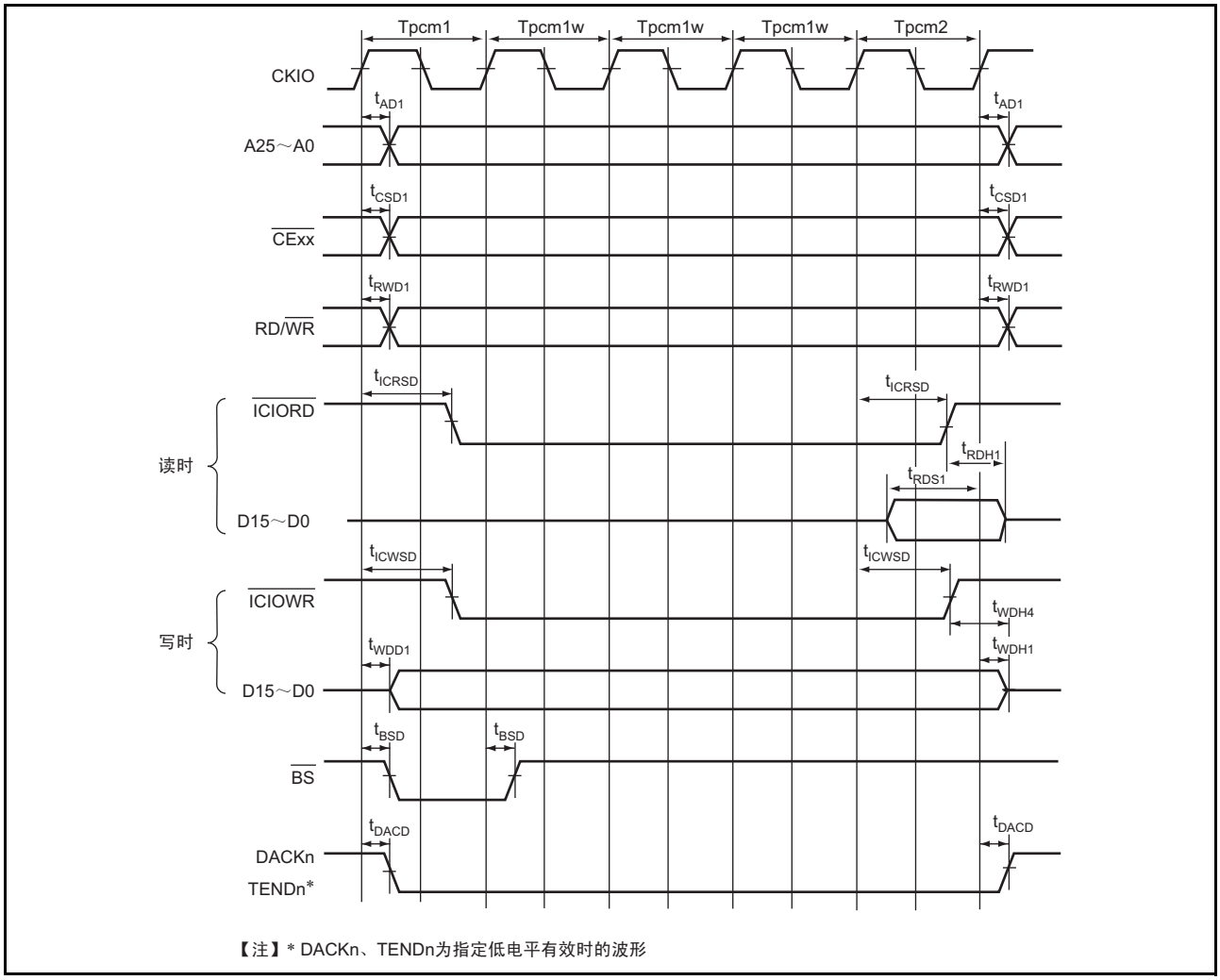


图 35.43 PCMCIA I/O 卡总线周期 （TED=0 周期、 TEH=0 周期、无等待）

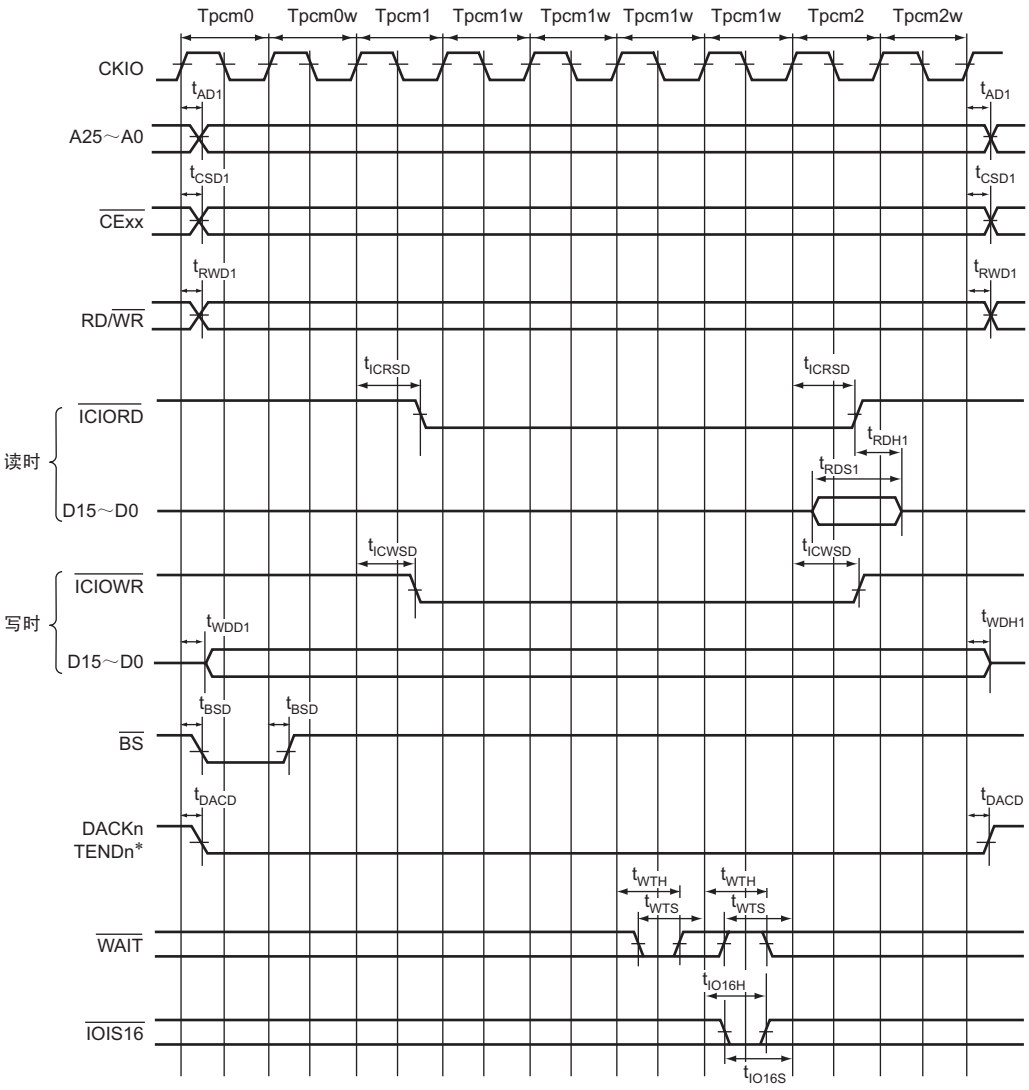


图 35.44 PCMCIA I/O 卡总线周期
（TED=2 个周期、 TEH=1 个周期、软件等待 0、硬件等待 1）

35.4.4 UBC 时序

表 35.9 UBC 时序

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、
AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=
$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目	符号	Min.	Max.	单位	参考图
$\overline{\text{UBCTRG}}$ 延迟时间	t_{UBCTGD}	—	14	ns	图 35.45

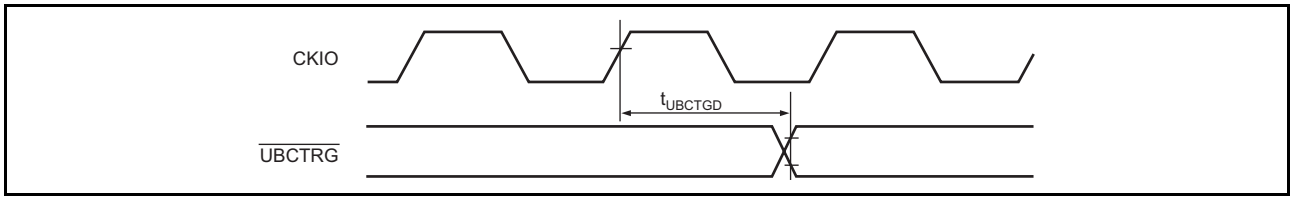


图 35.45 UBC 触发时序

35.4.5 DMAC 时序

表 35.10 DMAC 时序

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、
AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=
$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目	符号	Min.	Max.	单位	参考图
DREQ 设置时间	t_{DRQS}	15	—	ns	35.46
DREQ 保持时间	t_{DRQH}	15	—		
DACK、 TEND 延迟时间	t_{DACD}	—	15		35.47

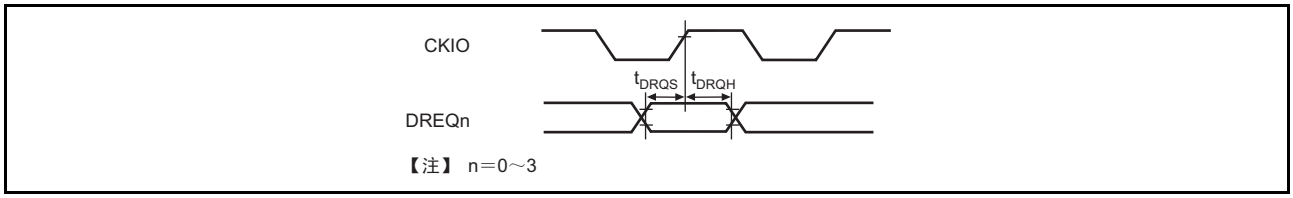


图 35.46 DREQ 输入时序

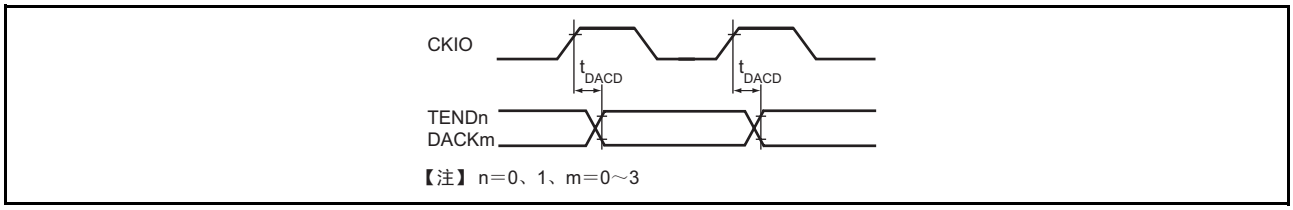


图 35.47 DACK、 TEND 输出时序

35.4.6　MTU2 时序

表 35.11　MTU2 时序

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、
AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=
$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目	符号	Min.	Max.	单位	参考图
输出比较的输出延迟时间	t_{TOCD}	—	100	ns	35.48
输入捕捉的输入设置时间	t_{TICS}	t_{cyc}/2+20	—	ns	
定时器输入设置时间	t_{TCKS}	t_{cyc}+20	—	ns	35.49
定时器时钟脉冲宽度 （指定单边沿）	$t_{TCKWH/L}$	1.5	—	t_{pcyc}	
定时器时钟脉冲宽度 （指定双边沿）	$t_{TCKWH/L}$	2.5	—	t_{pcyc}	
定时器时钟脉冲宽度 （相位计数模式）	$t_{TCKWH/L}$	2.5	—	t_{pcyc}	

【注】 t_{pcyc} 表示外围时钟 （Pϕ）的周期。

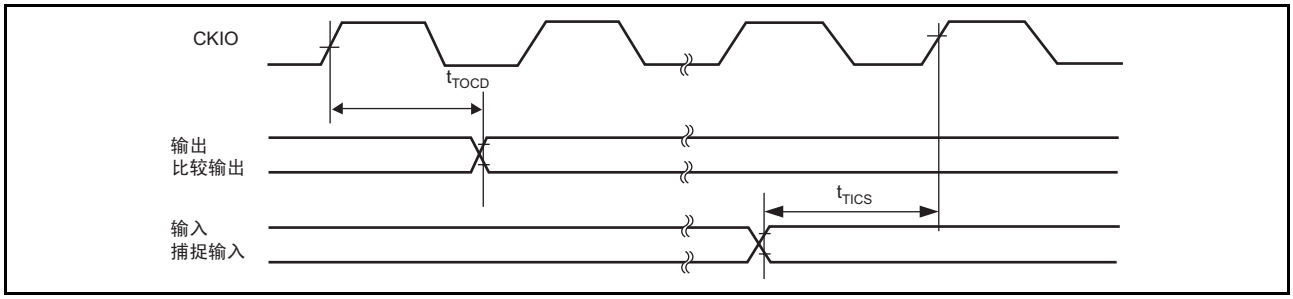


图 35.48　MTU2 输入 / 输出时序

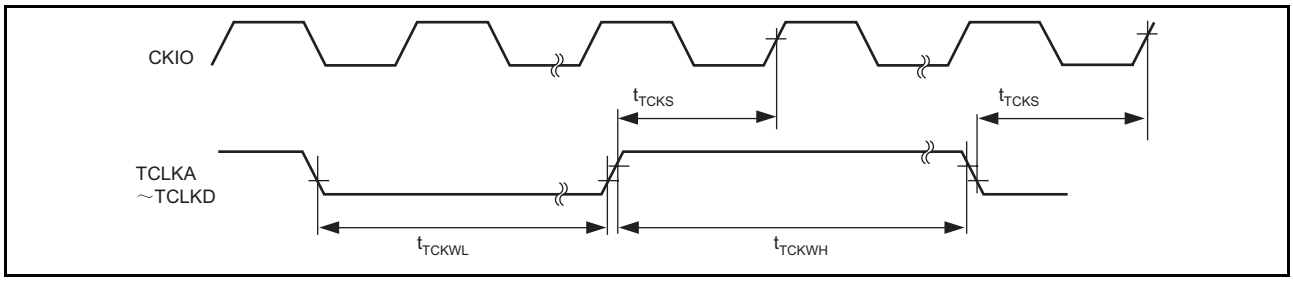


图 35.49　MTU2 时钟输入时序

35.4.7　WDT 时序

表 35.12　WDT 时序

条件：V_{CC}=PLLV_{CC}=USBDV_{CC}=1.1 ～ 1.3V、 PV_{CC}=USBDPV_{CC}=3.0 ～ 3.6V、
AV_{CC}=3.0 ～ 3.6V、 USBAV_{CC}=1.1 ～ 1.3V、 USBAPV_{CC}=3.0 ～ 3.6V、 V_{SS}=PLLV_{SS}=PV_{SS}=AV_{SS}=USBDV_{SS}=
USBAV_{SS}=USBDPV_{SS}=USBAPV_{SS}=0V、 T_a=–40 ～ 85°C

项目	符号	Min.	Max.	单位	参考图
$\overline{\text{WDTOVF}}$ 延迟时间	t_{WOVD}	—	100	ns	35.50

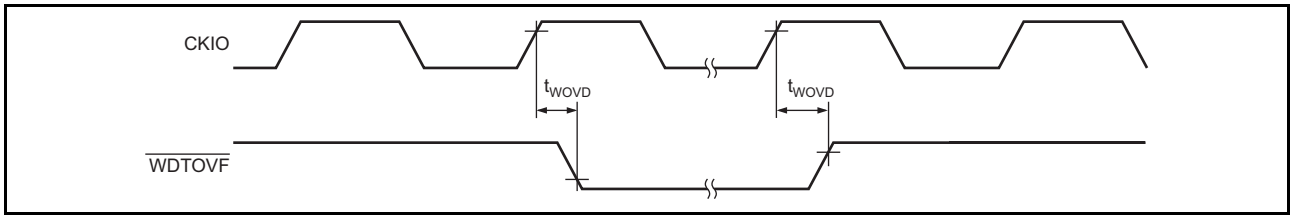


图 35.50　WDT 时序

35.4.8　SCIF 时序

表 35.13　SCIF 时序

条件：V_{CC}=PLLV_{CC}=USBDV_{CC}=1.1 ～ 1.3V、 PV_{CC}=USBDPV_{CC}=3.0 ～ 3.6V、
AV_{CC}=3.0 ～ 3.6V、 USBAV_{CC}=1.1 ～ 1.3V、 USBAPV_{CC}=3.0 ～ 3.6V、 V_{SS}=PLLV_{SS}=PV_{SS}=AV_{SS}=USBDV_{SS}=
USBAV_{SS}=USBDPV_{SS}=USBAPV_{SS}=0V、 T_a=–40 ～ 85°C

项目		符号	Min.	Max.	单位	参考图
输入时钟周期	时钟同步	t_{Scyc}	12	—	t_{pcyc}	35.51
	异步		4	—	t_{pcyc}	35.51
输入时钟上升时间		t_{SCKr}	—	1.5	t_{pcyc}	35.51
输入时钟下降时间		t_{SCKf}	—	1.5	t_{pcyc}	35.51
输入时钟宽度		t_{SCKW}	0.4	0.6	t_{Scyc}	35.51
发送数据延迟时间 （时钟同步）		t_{TXD}	—	$3t_{pcyc}$+15	ns	35.52
接收数据设置时间 （时钟同步）		t_{RXS}	$4t_{pcyc}$+15	—	ns	35.52
接收数据保持时间 （时钟同步）		t_{RXH}	$1t_{pcyc}$+15	—	ns	35.52

【注】 t_{pcyc} 表示外围时钟 （Pϕ）的周期。

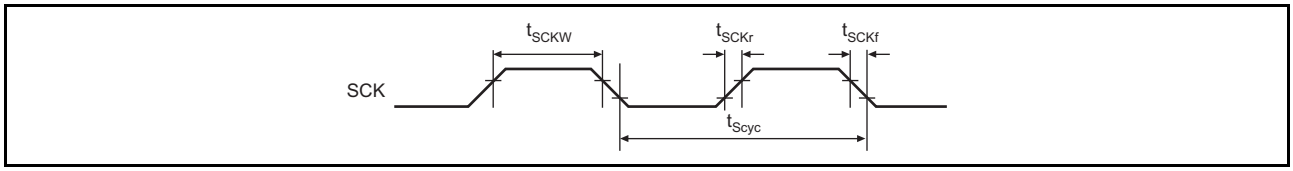


图 35.51　SCK 输入时钟时序

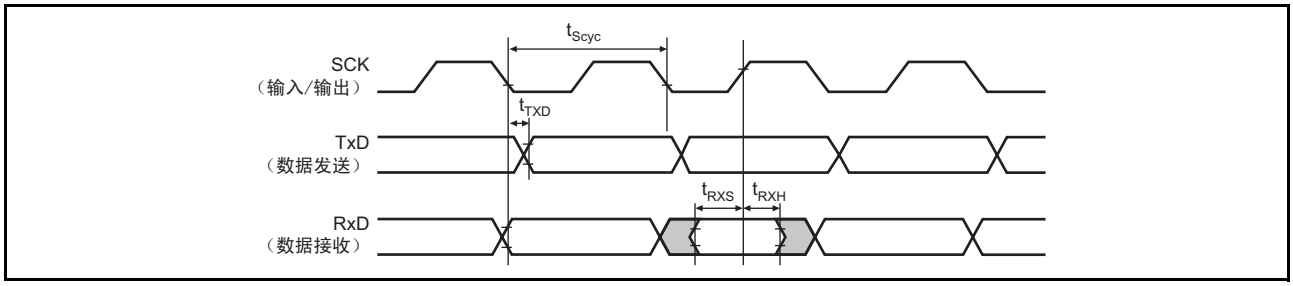


图 35.52　时钟同步模式时的 SCIF 输入 / 输出时序

35.4.9　SSU 时序

表 35.14　SSU 时序

条件：V_{CC}=PLLV_{CC}=USBDV_{CC}=1.1 ～ 1.3V、 PV_{CC}=USBDPV_{CC}=3.0 ～ 3.6V、
AV$_{CC}$=3.0 ～ 3.6V、 USBAV$_{CC}$=1.1 ～ 1.3V、 USBAPV_{CC}=3.0 ～ 3.6V、 V_{SS}=PLLV_{SS}=PV_{SS}=AV$_{SS}$=USBDV_{SS}=
USBAV$_{SS}$=USBDPV_{SS}=USBAPV_{SS}=0V、 T_a=–40 ～ 85°C

项目		符号	Min.	Max.	单位	参考图
时钟周期	主	t_{SUcyc}	4	256	t_{pcyc}	35.53 35.54 35.55 35.56
	从		4	256		
时钟高电平脉冲宽度	主	t_{HI}	48	—	ns	
	从		48	—		
时钟低电平脉冲宽度	主	t_{LO}	48	—	ns	
	从		48	—		
时钟上升时间		t_{RISE}	—	12	ns	
时钟下降时间		t_{FALL}	—	12	ns	
数据输入设置时间	主	t_{SU}	30	—	ns	
	从		20	—		
数据输入保持时间	主	t_H	0	—	ns	
	从		20	—		
$\overline{SCS}$ 设置时间	主	t_{LEAD}	1.5	—	t_{pcyc}	
	从		1.5	—		
$\overline{SCS}$ 保持时间	主	t_{LAG}	1.5	—	t_{pcyc}	
	从		1.5	—		
数据输出延迟时间	主	t_{OD}	—	50	ns	
	从		—	50		
数据输出保持时间	主	t_{OH}	0	—	ns	
	从		0	—		
连续发送延迟时间	主	t_{TD}	1.5	—	t_{pcyc}	
	从		1.5	—		

项目	符号	Min.	Max.	单位	参考图
从属存取时间	t_{SA}	—	1	t_{pcyc}	35.55
从属输出释放时间	t_{REL}	—	1	t_{pcyc}	35.56

【注】 t_{pcyc} 表示外围时钟 （Pϕ）的周期。

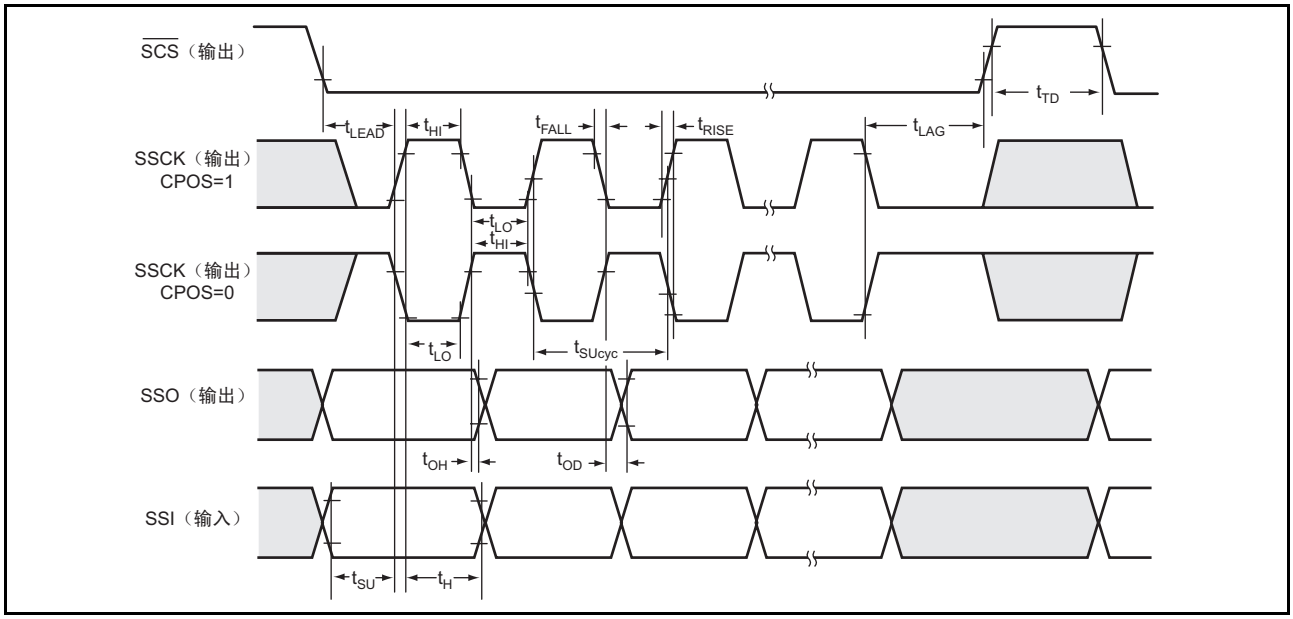


图 35.53 SSU 时序 （主、 CPHS=1）

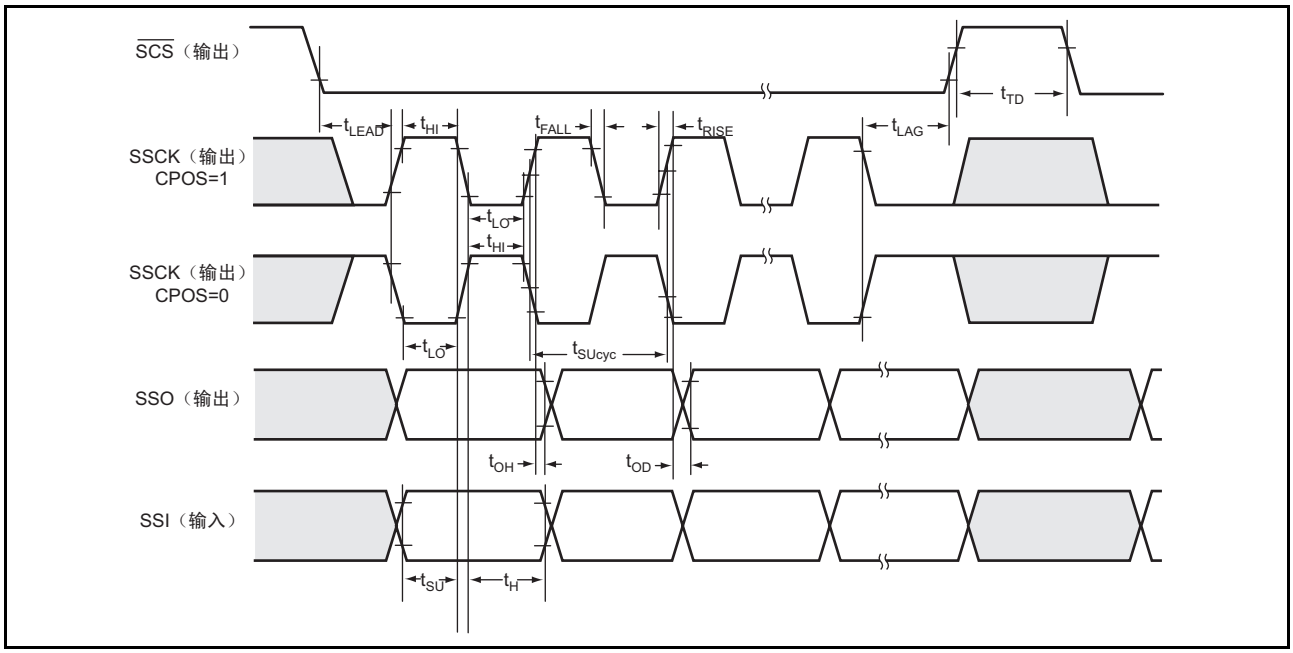


图 35.54 SSU 时序 （主、 CPHS=0）

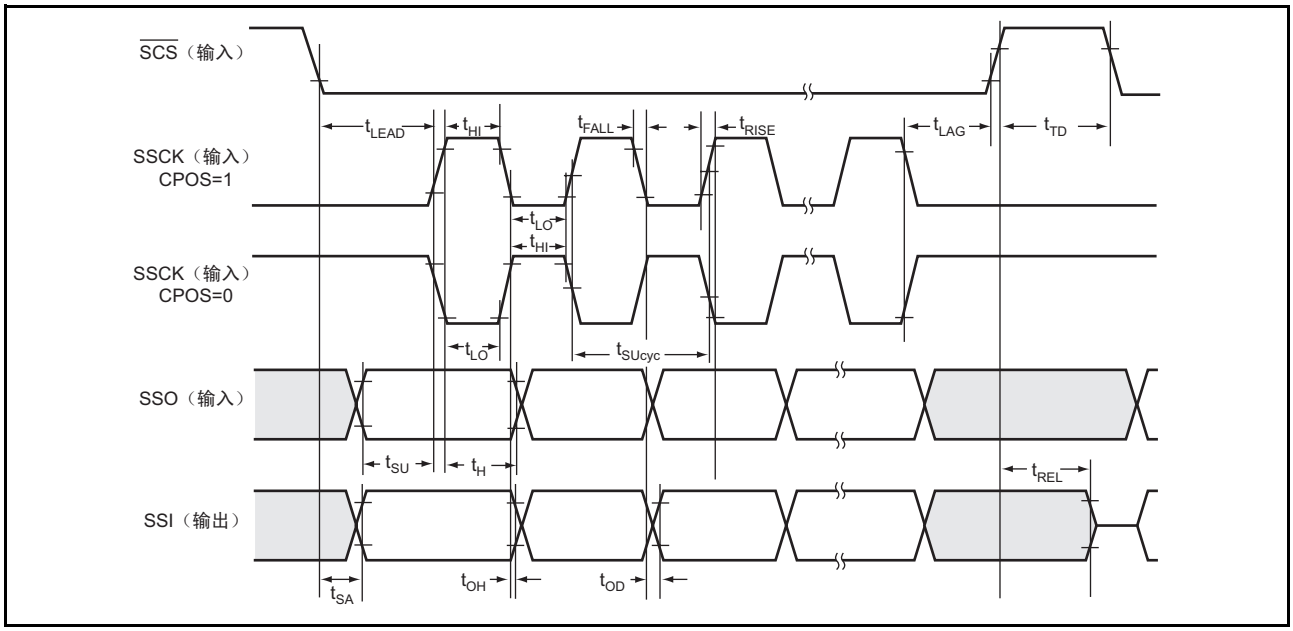


图 35.55 SSU 时序 （从、 CPHS=1）

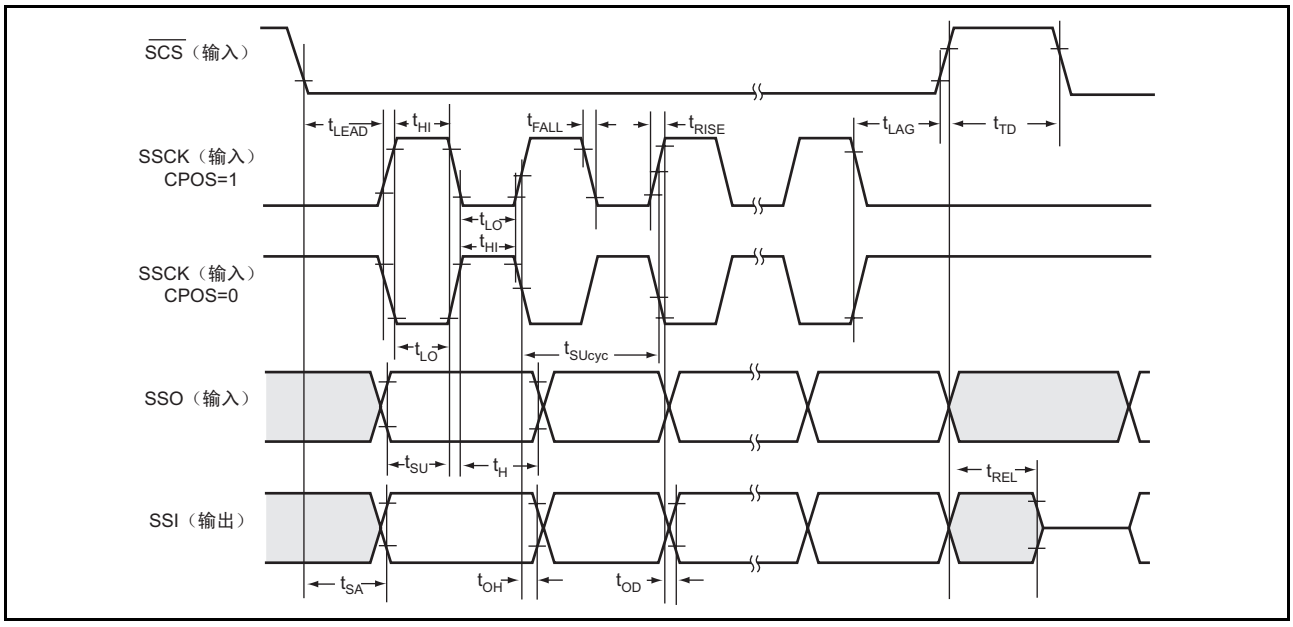


图 35.56 SSU 时序 （从、 CPHS=0）

35.4.10　IIC3 时序

表 35.15　IIC3 时序

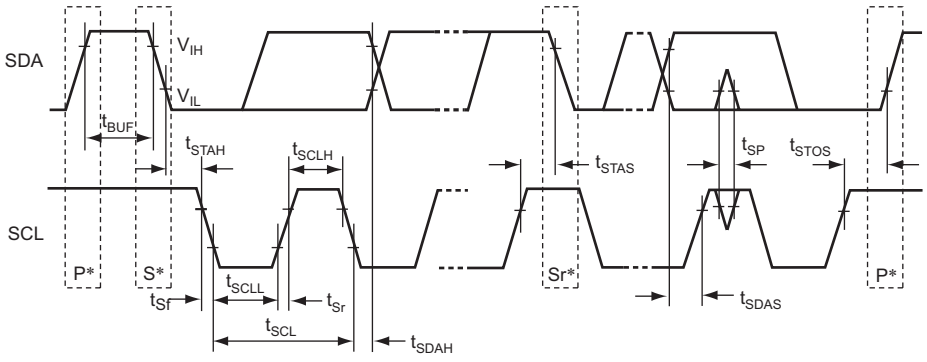
条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目	符号	Min.	Max.	单位	参考图
SCL 输入周期时间	t_{SCL}	$12t_{pcyc}$*1+600	—	ns	35.57
SCL 输入高电平脉冲宽度	t_{SCLH}	$3t_{pcyc}$*1+300	—	ns	
SCL 输入低电平脉冲宽度	t_{SCLL}	$5t_{pcyc}$*1+300	—	ns	
SCL、 SDA 输入上升时间	t_{Sr}	—	300	ns	
SCL、 SDA 输入下降时间	t_{Sf}	—	300	ns	
SCL、 SDA 输入尖峰脉冲消除时间 *2	t_{SP}	—	1、 2	t_{pcyc}*1	
SDA 输入总线自由时间	t_{BUF}	5	—	t_{pcyc}*1	
开始条件输入保持时间	t_{STAH}	3	—	t_{pcyc}*1	
再发送开始条件输入设置时间	t_{STAS}	3	—	t_{pcyc}*1	
停止条件输入设置时间	t_{STOS}	3	—	t_{pcyc}*1	
数据输入设置时间	t_{SDAS}	$1t_{pcyc}$*1+20	—	ns	
数据输入保持时间	t_{SDAH}	0	—	ns	
SCL、 SDA 的电容负载	Cb	0	400	pF	
SCL、 SDA 输出的下降时间 *3	t_{Sf}	—	250	ns	

【注】 *1　t_{pcyc} 表示外围时钟 （Pϕ）的周期。

*2　依存于 NF2CYC 寄存器的值。

*3　表示 I/O 缓冲器的特性。



* S、P、Sr 各自表示如下。
S：开始条件
P：停止条件
Sr：再次发送“开始”条件

图 35.57　IIC3 输入 / 输出时序

35.4.11 SSI 时序

表 35.16 SSI 时序

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、
AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=
$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目	符号	Min.	Max.	单位	备注	参考图
输出时钟周期	t_O	160	3364	ns	输出	35.58
输入时钟周期	t_I	80	3364	ns	输入	
时钟高电平	t_{HC}	32	—	ns	双向	
时钟低电平	t_{LC}	32	—	ns		
时钟上升时间	t_{RC}	—	20	ns	输出 （100pF）	
延迟	t_{DTR}	–5	25	ns	发送	35.59、 35.60
设置时间	t_{SR}	25	—	ns	接收	
保持时间	t_{HTR}	5	—	ns	接收、发送	
AUDIO_CLK 输入频率	f_{AUDIO}	1	40	MHz		35.61

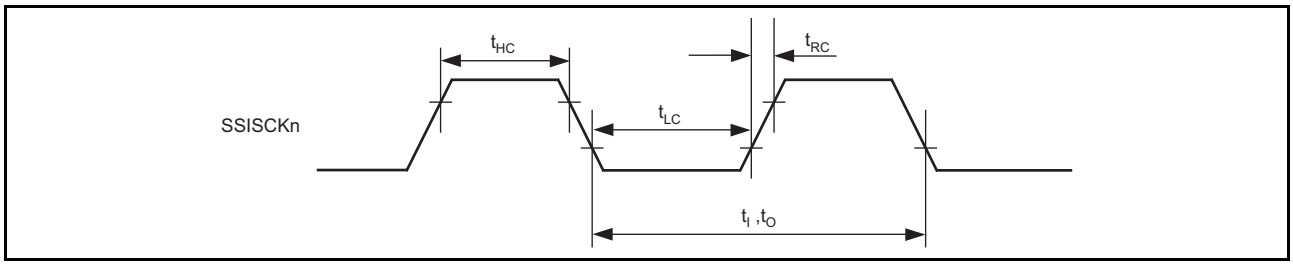


图 35.58 时钟输入 / 输出时序

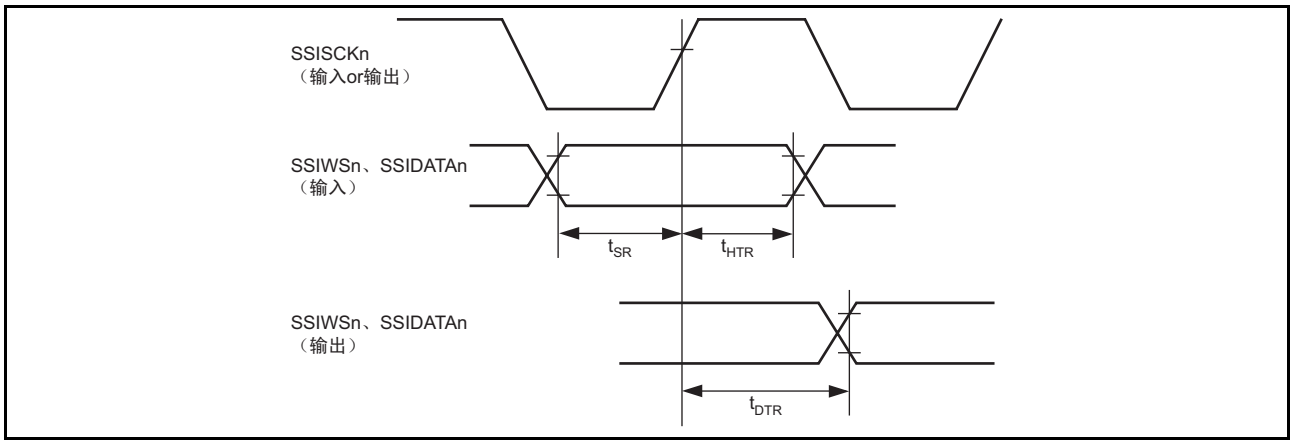


图 35.59 SSI 发送 / 接收时序 （SSISCKn 上升沿同步）

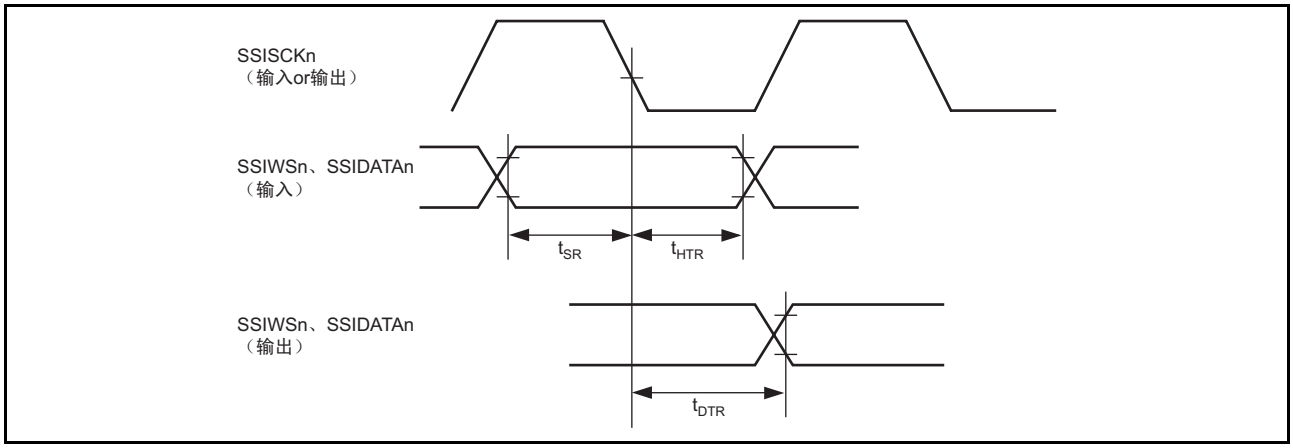


图 35.60 SSI 发送 / 接收时序 （SSISCKn 下降沿同步）

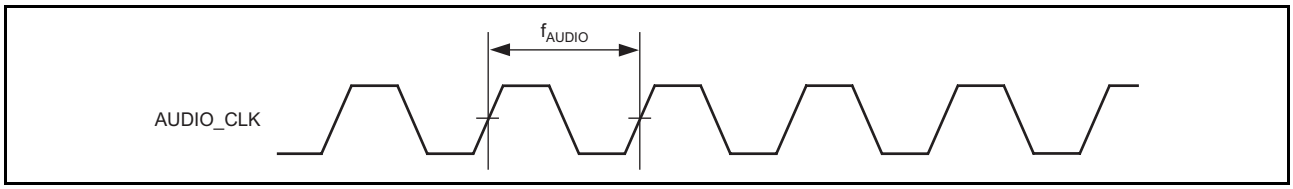


图 35.61 AUDIO_CLK 输入时序

35.4.12 RCAN-TL1 时序

表 35.17 RCAN-TL1 时序

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、
AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=
$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目	符号	Min.	Max.	单位	参考图
发送数据延迟时间	t_{CTXD}	—	100	ns	35.62
接收数据设置时间	t_{CRXS}	100	—		
接收数据保持时间	t_{CRXH}	100	—		

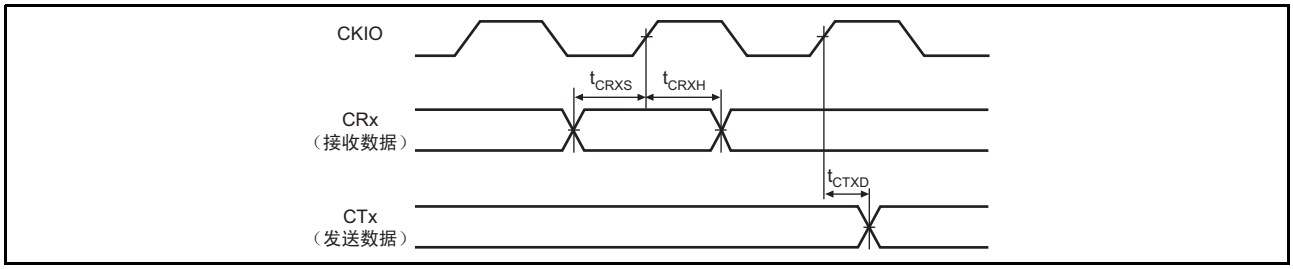


图 35.62 RCAN-TL1 输入 / 输出时序

35.4.13 ADC 时序

表 35.18 ADC 时序

条件：V_{CC}=PLLV_{CC}=USBDV_{CC}=1.1 ～ 1.3V、 PV_{CC}=USBDPV_{CC}=3.0 ～ 3.6V、
AV_{CC}=3.0 ～ 3.6V、 USBAV_{CC}=1.1 ～ 1.3V、 USBAPV_{CC}=3.0 ～ 3.6V、 V_{SS}=PLLV_{SS}=PV_{SS}=AV_{SS}=USBDV_{SS}=
USBAV_{SS}=USBDPV_{SS}=USBAPV_{SS}=0V、 T_a=–40 ～ 85°C

模块	项目		符号	Min.	Max.	单位	参考图
A/D 转换器	触发输入设置时间	B：P 时钟比 =1：1	t_{TRGS}	17	—	ns	35.63
		B：P 时钟比 =2：1		t_{cyc}+17	—		
		B：P 时钟比 =4：1		3×t_{cyc}+17	—		

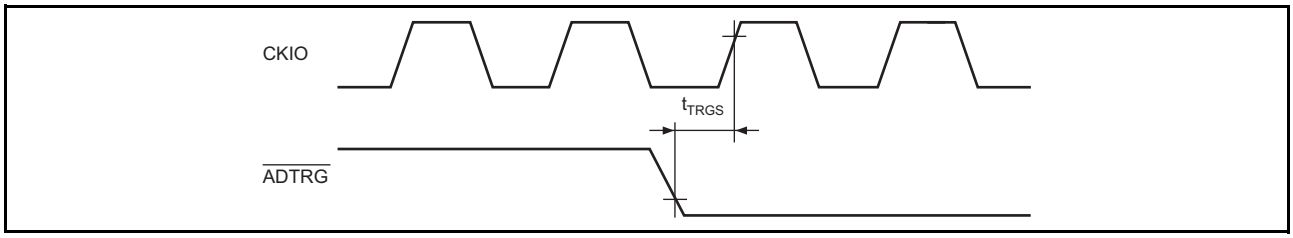


图 35.63 A/D 转换器外部触发输入时序

35.4.14 FLCTL 时序

表 35.19 AND 型闪存接口时序

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目	符号	Min.	Max.	单位	参考图
命令发送设置时间	t_{ACDS}	$2\times t_{fcyc}$–10	—	ns	35.64、 35.68
命令发送保持时间	t_{ACDH}	$2\times t_{fcyc}$–10	—	ns	
数据输出设置时间	t_{ADOS}	t_{fcyc}–10	—	ns	35.64、 35.65、 35.68
数据输出保持时间	t_{ADOH}	t_{fcyc}–10	—	ns	
数据输出设置时间 2	t_{ADOS2}	$0.5\times t_{fcyc}$–10	—	ns	35.67
数据输出保持时间 2	t_{ADOH2}	$0.5\times t_{fcyc}$–10	—	ns	
$\overline{FWE}$ 周期时间	t_{ACWC}	$2\times t_{fcyc}$–5	—	ns	35.65
$\overline{FWE}$ 低脉冲宽度	t_{AWP}	t_{fcyc}–5	—	ns	35.64、 35.65、 35.68
$\overline{FWE}$ 高脉冲宽度	t_{AWPH}	t_{fcyc}–5	—	ns	35.65
命令 - 地址转换时间	t_{ACAS}	$4\times t_{fcyc}$	—	ns	
地址 - 数据读转换时间	t_{AADDR}	$32\times t_{pcyc}$	—	ns	35.66
地址 - 就绪 / 忙转换时间	t_{AADRB}	—	$35\times t_{pcyc}$	ns	
就绪 / 忙 - 数据读转换时间	t_{ARBDR}	$3\times t_{fcyc}$	—	ns	
数据读设置时间	t_{ADRS}	t_{fcyc}–10	—	ns	35.66
FSC 周期时间	t_{ASCC}	t_{fcyc}–5	—	ns	35.66、 35.67
FSC 高脉冲宽度	t_{ASP}	$0.5\times t_{fcyc}$–5	—	ns	
FSC 低脉冲宽度	t_{ASPL}	$0.5\times t_{fcyc}$–5	—	ns	
读数据设置时间	t_{ARDS}	24	—	ns	35.66、 35.68
读数据保持时间	t_{ARDH}	5	—	ns	
状态读数据设置时间	t_{ASRDS}	$2\times t_{fcyc}$+24	—	ns	35.68
地址 - 数据写转换时间	t_{AADDW}	$4\times t_{PCYC}$	—	ns	35.67
数据写的设置时间	t_{ADWS}	$50\times t_{pcyc}$	—	ns	
FSC-FOE 保持时间	t_{ASOH}	$2\times t_{fcyc}$–10	—	ns	35.66

【注】 t_{fcyc} 表示 FLCTL 时钟的 1 个周期时间。

t_{PCYC} 表示外围时钟 （Pϕ）的 1 个周期时间。

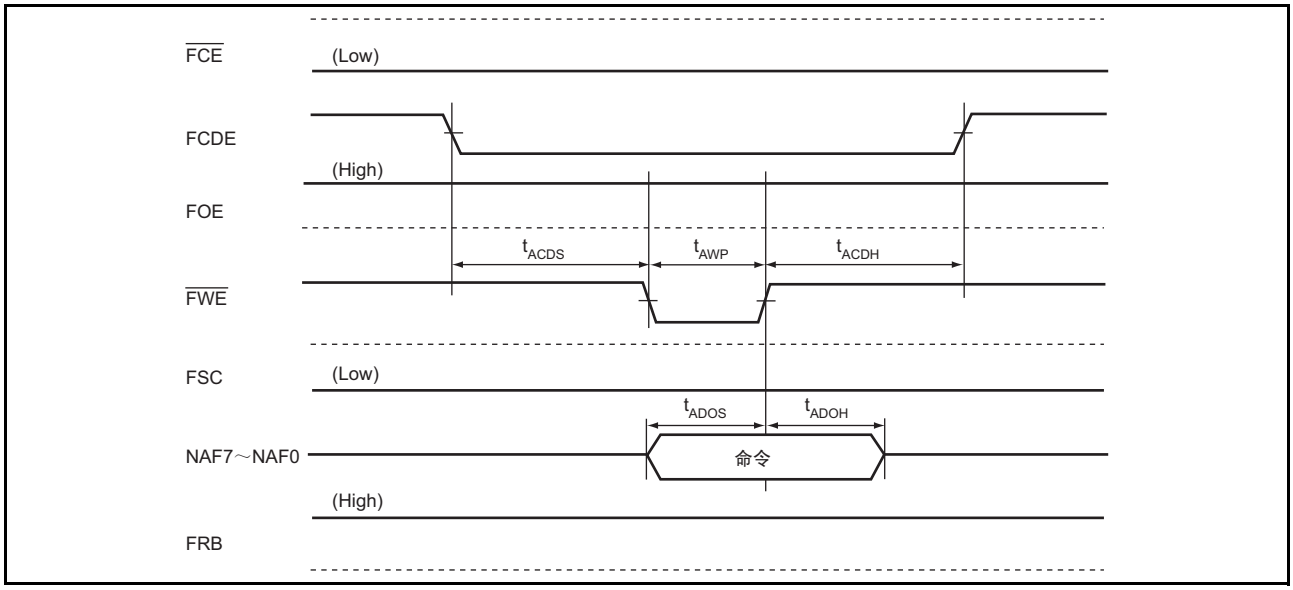


图 35.64 AND 型闪存的命令发送时序

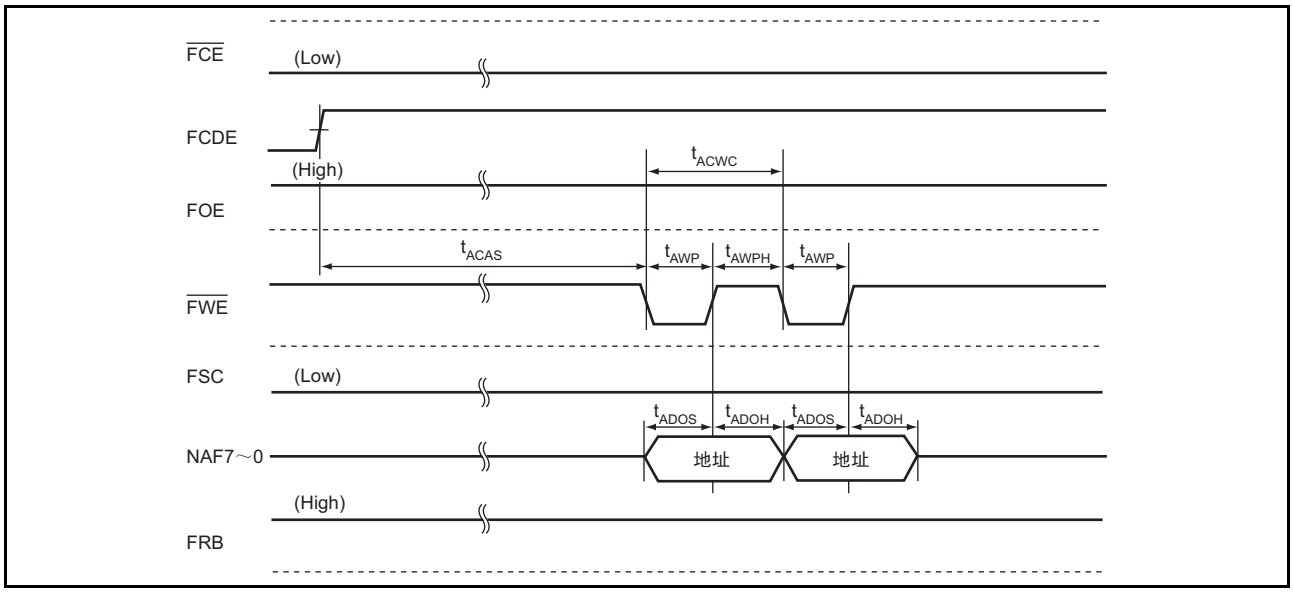


图 35.65 AND 型闪存的地址发送时序

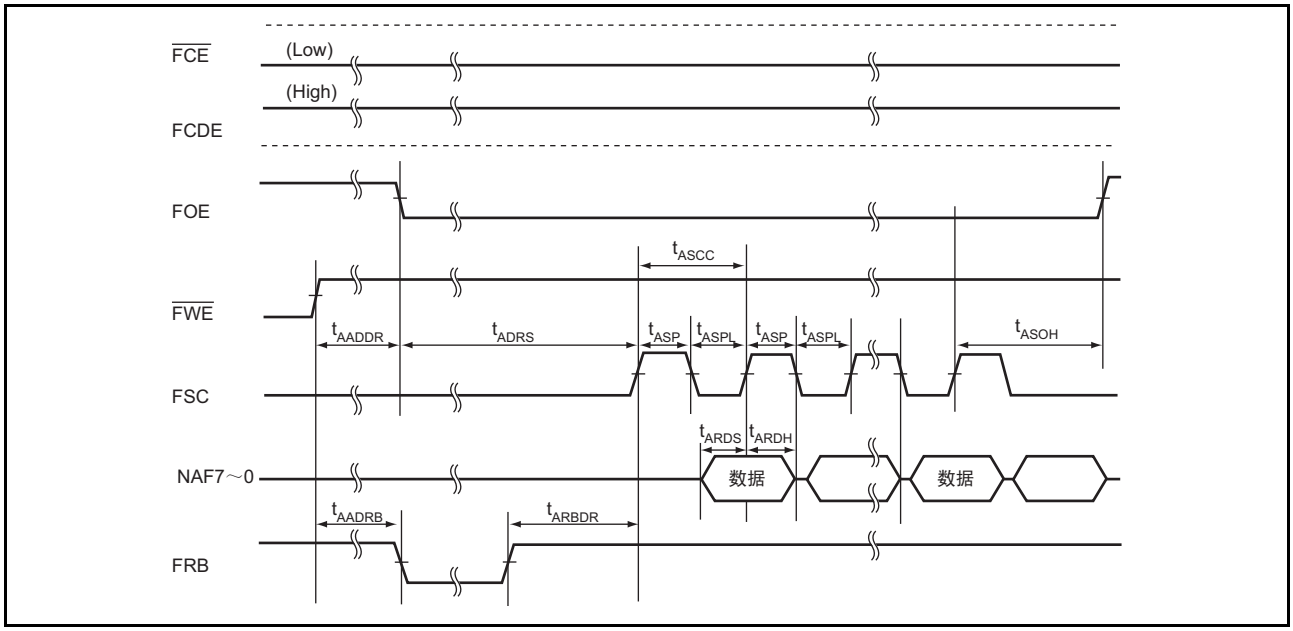


图 35.66 AND 型闪存的数据读时序

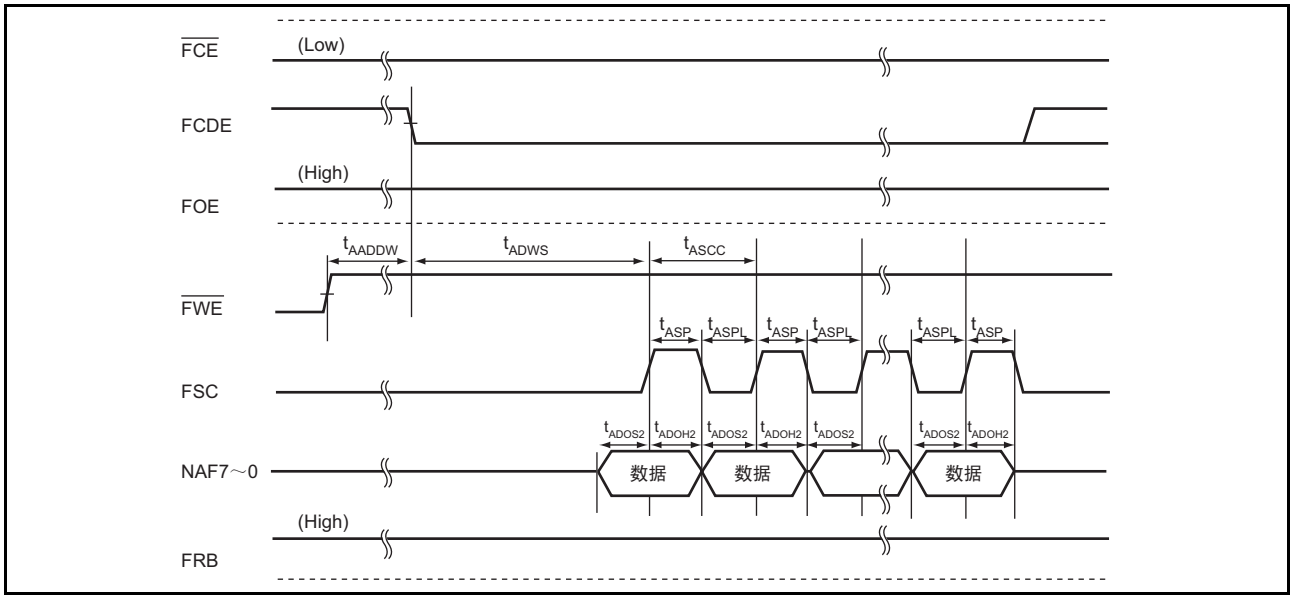


图 35.67 AND 型闪存的数据写时序

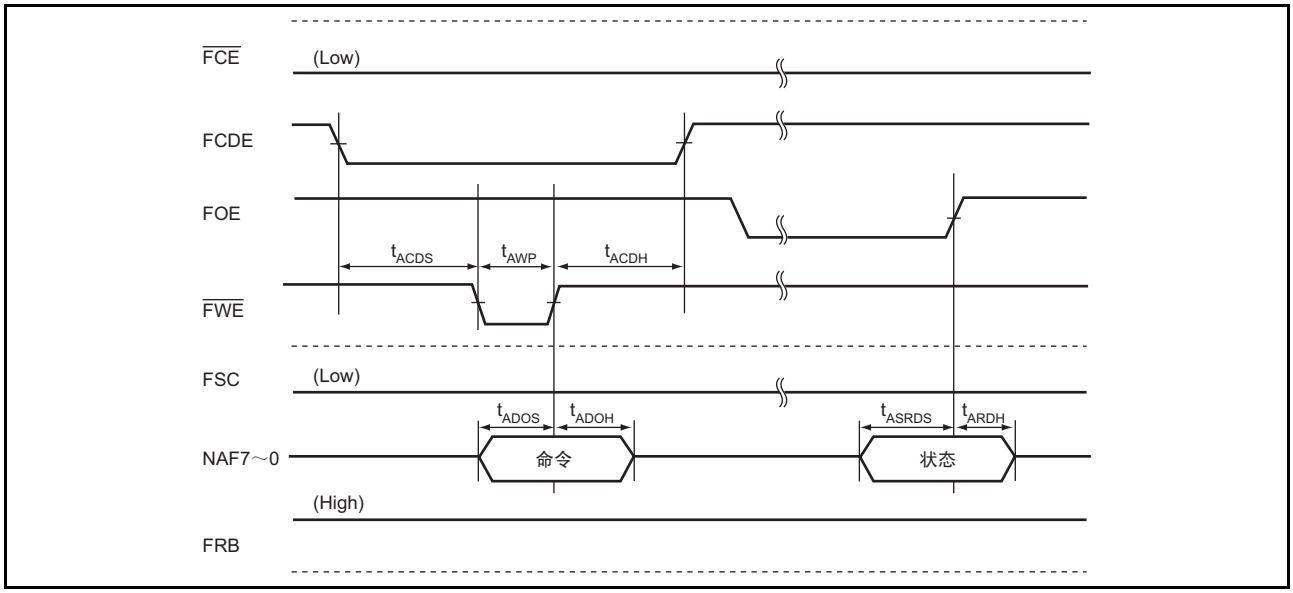


图 35.68 AND 型闪存的状态读时序

表 35.20　NAND 型闪存接口时序

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、
AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=
$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目	符号	Min.	Max.	单位	参考图
命令输出设置时间	t_{NCDS}	$2\times t_{fcyc}-10$	—	ns	35.69、 35.73
命令输出保持时间	t_{NCDH}	$1.5\times t_{fcyc}-5$	—	ns	
数据输出设置时间	t_{NDOS}	$0.5\times t_{wfcyc}-5$	—	ns	35.69、 35.70、 35.72、 35.73
数据输出保持时间	t_{NDOH}	$0.5\times t_{wfcyc}-10$	—	ns	
命令 - 地址转换时间 1	t_{NCDAD1}	$1.5\times t_{fcyc}-10$	—	ns	35.69、 35.70
命令 - 地址转换时间 2	t_{NCDAD2}	$2\times t_{fcyc}-10$	—	ns	35.70
$\overline{FWE}$ 周期时间	t_{NWC}	$t_{wfcyc}-5$	—	ns	35.70、 35.72
$\overline{FWE}$ 低脉冲宽度	t_{NWP}	$0.5\times t_{wfcyc}-5$	—	ns	35.69、 35.70、 35.72、 35.73
$\overline{FWE}$ 高脉冲宽度	t_{NWH}	$0.5\times t_{wfcyc}-5$	—	ns	35.70、 35.72
地址 - 就绪 / 忙转换时间	t_{NADRB}	—	$32\times t_{pcyc}$	ns	35.70、 35.71
命令 - 就绪 / 忙转换时间	t_{NCDRB}	—	$10\times t_{pcyc}$	ns	35.70、 35.71
就绪 / 忙 - 数据读换时间 1	t_{NRBDR1}	$1.5\times t_{fcyc}$	—	ns	35.71
就绪 / 忙 - 数据读换时间 2	t_{NRBDR2}	$32\times t_{pcyc}$	—	ns	
FSC 周期时间	t_{NSCC}	$t_{wfcyc}-5$	—	ns	
FSC 低脉冲宽度	t_{NSP}	$0.5\times t_{wfcyc}-5$	—	ns	35.71、 35.73
FSC 高脉冲宽度	t_{NSPH}	$0.5\times t_{wfcyc}-5$	—	ns	35.71
读数据设置时间	t_{NRDS}	24	—	ns	35.71、 35.73
读数据保持时间	t_{NRDH}	5	—	ns	35.71、 35.73
数据写设置时间	t_{NDWS}	$32\times t_{pcyc}$	—	ns	35.72
命令 - 状态读转换时间	t_{NCDSR}	$4\times t_{fcyc}$	—	ns	35.73
命令输出关断 - 状态读转换时间	t_{NCDFSR}	$3.5\times t_{fcyc}$	—	ns	
状态读设置时间	t_{NSTS}	$2.5\times t_{fcyc}$	—	ns	

【注】 t_{fcyc} 为 FLCTL 时钟的 1 个周期时间。

在 NANDWF 位为 0 时，t_{wfcyc} 为 FLCTL 时钟的 1 个周期时间；在 NANDWF 位为 1 时，t_{wfcyc} 为 FLCTL 时钟的 2 个周期时间。

t_{pcyc} 表示外围时钟 （Pϕ）的 1 个周期时间。

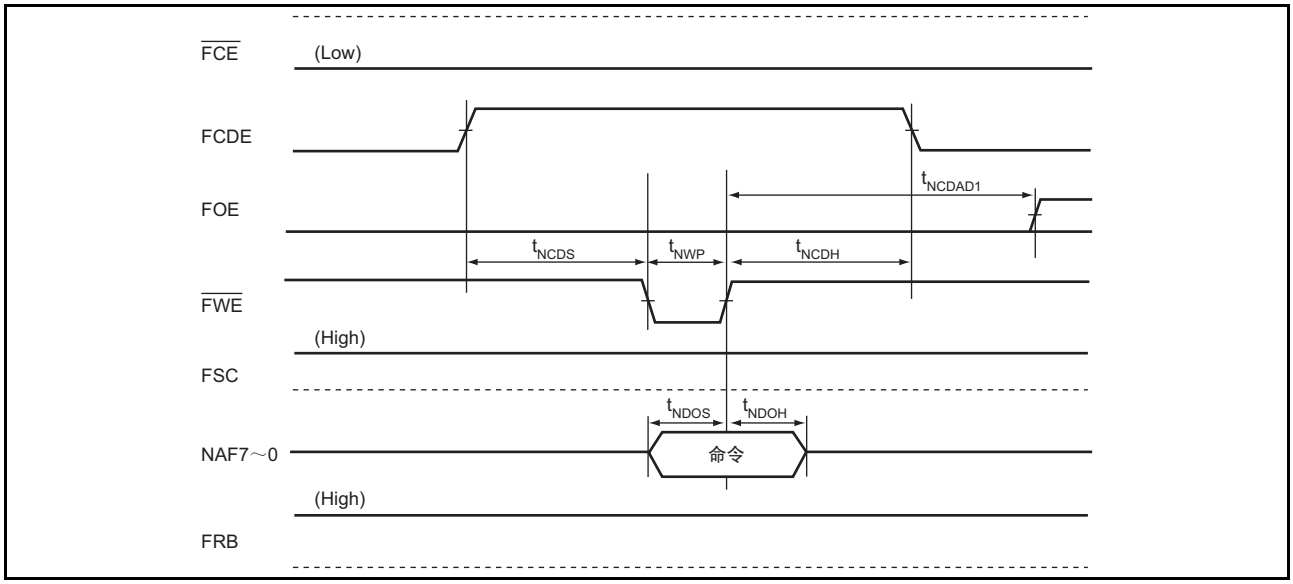


图 35.69 NAND 型闪存的命令发送时序

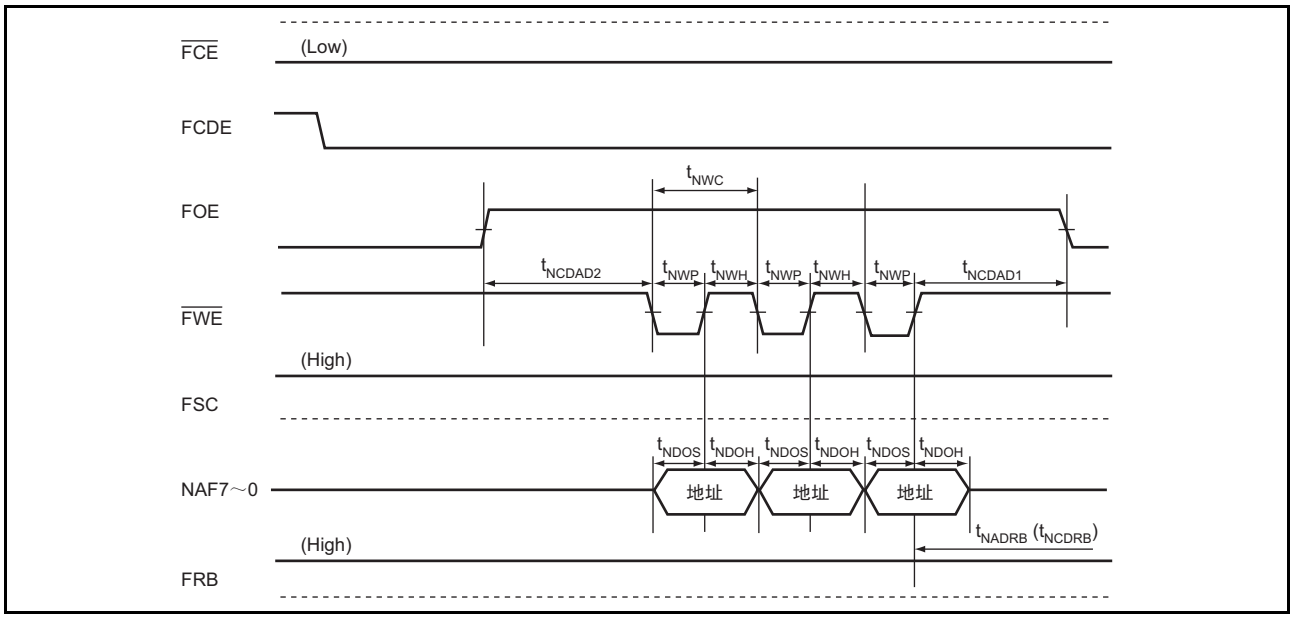


图 35.70 NAND 型闪存的地址发送时序

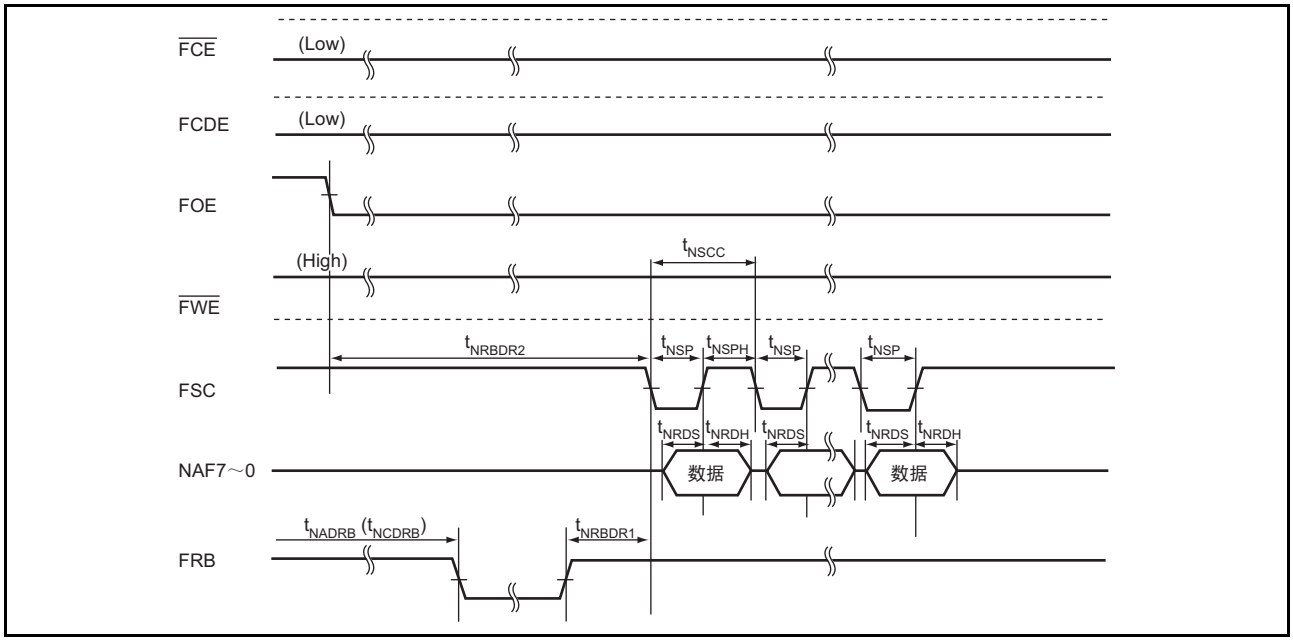


图 35.71 NAND 型闪存的数据读时序

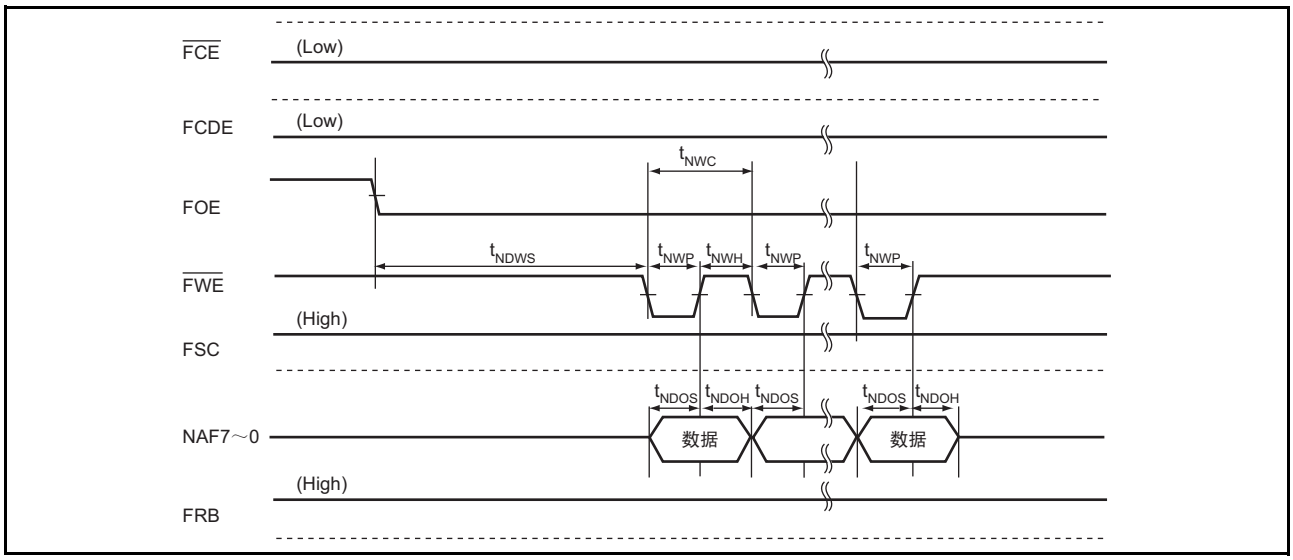


图 35.72 NAND 型闪存的数据写时序

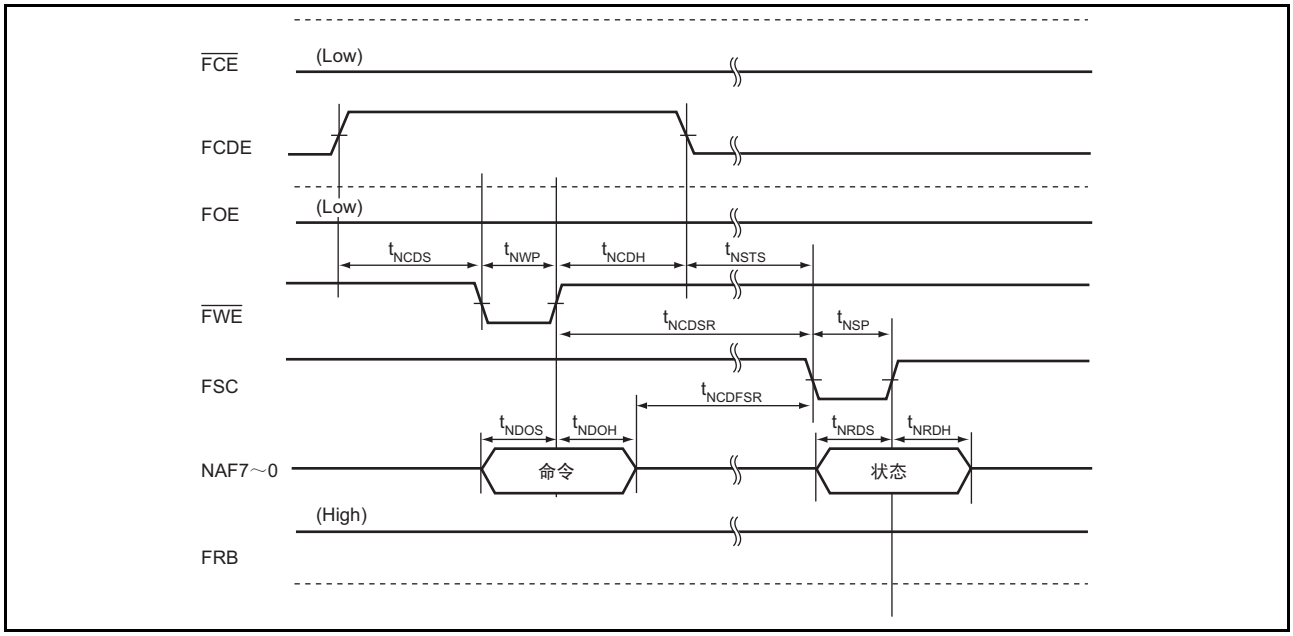


图 35.73 NAND 型闪存的状态读时序

35.4.15 USB 时序

表 35.21 USB 收发器时序 （全速时）

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、
AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$$PV_{SS}$=$AV_{SS}$=$USBDV_{SS}$=
$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目	符号	Min.	Typ.	Max.	单位	参考图
上升时间	t_{FR}	4	—	20	ns	35.74
下降时间	t_{FF}	4	—	20	ns	
上升时间 / 下降时间比	t_{FR}/t_{FF}	90	—	111.11	%	

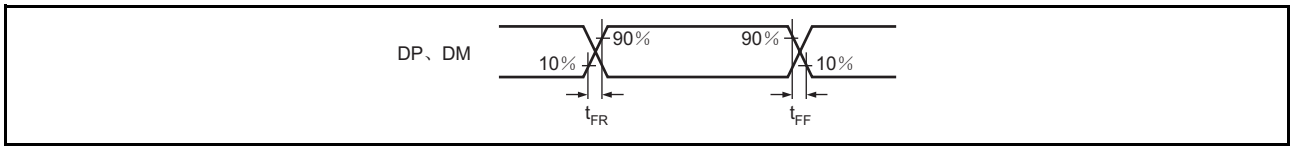


图 35.74 DP、 DM 输出时序 （全速时）

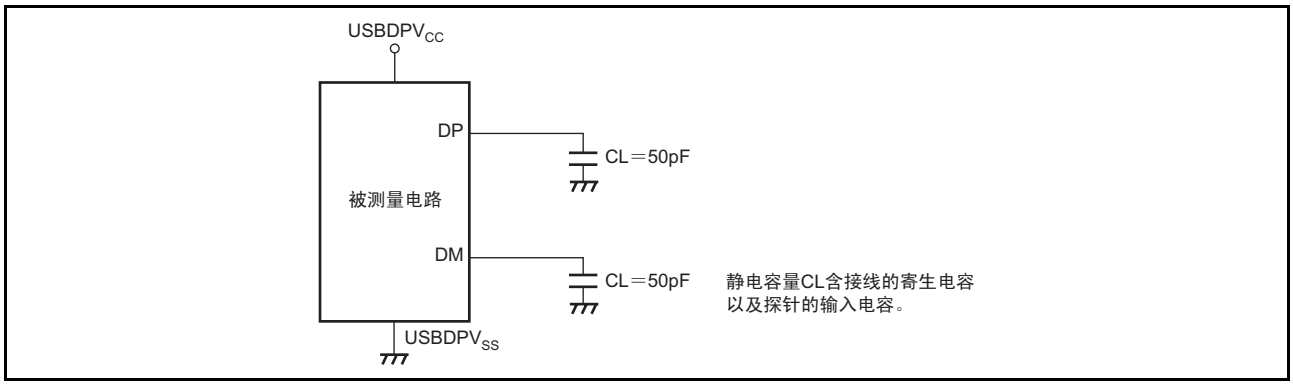


图 35.75 测量电路 （全速时）

表 35.22　USB 收发器时序 （高速时）

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、
AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=
$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目	符号	Min.	Typ.	Max.	单位	参考图
上升时间	t_{HSR}	500	—	—	ps	35.76
下降时间	t_{HSF}	500	—	—	ps	
输出驱动器电阻	Z_{HSDRV}	40.5	—	49.5	Ω	

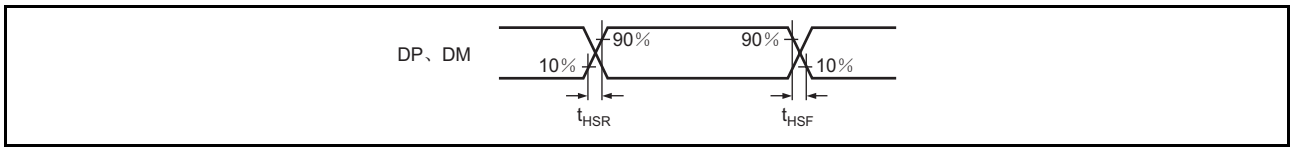


图 35.76　DP、 DM 输出时序 （高速时）

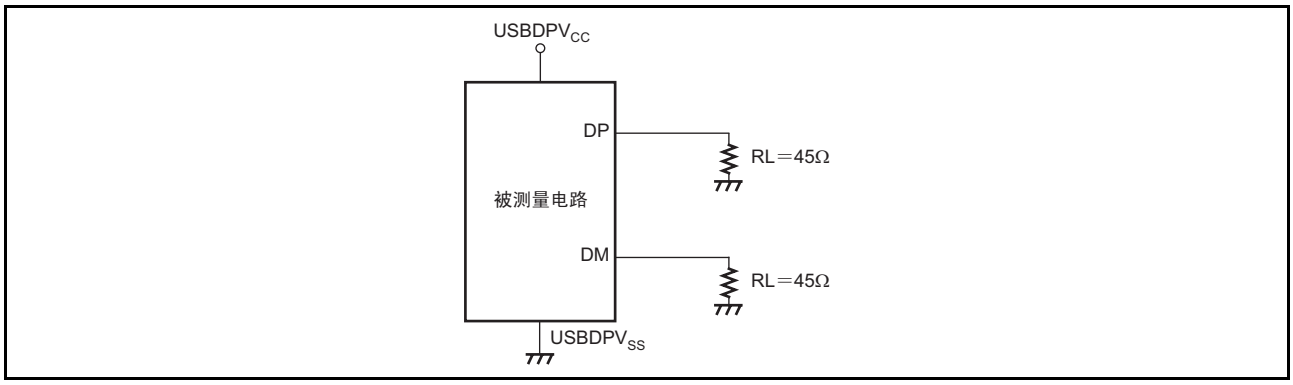


图 35.77　测量电路 （高速时）

35.4.16 LCDC 时序

表 35.23 LCDC 时序

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、
AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=
$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目	符号	Min.	Max.	单位	参考图
LCD_CLK 输入时钟频率	t_{FREQ}	—	66.66	MHz	
LCD_CLK 输入时钟上升时间	t_r	—	3	ns	
LCD_CLK 输入时钟下降时间	t_f	—	3	ns	
LCD_CLK 输入时钟占空比	t_{DUTY}	90	110	%	
时钟 （LCD_CL2）周期时间	t_{CC}	25	—	ns	35.78
时钟 （LCD_CL2）高电平脉冲宽度	t_{CHW}	7	—	ns	
时钟 （LCD_CL2）低电平脉冲宽度	t_{CLW}	7	—	ns	
时钟 （LCD_CL2）转换时间 （上升 / 下降）	t_{CT}	—	3	ns	
数据 （LCD_DATA）延迟时间	t_{DD}	–3.5	3	ns	
表示允许 （LCD_M_DISP）延迟时间	t_{ID}	–3.5	3	ns	
水平同步信号 （LCD_CL1）延迟时间	t_{HD}	–3.5	3	ns	
垂直同步信号 （LCD_FLM）延迟时间	t_{VD}	–3.5	3	ns	

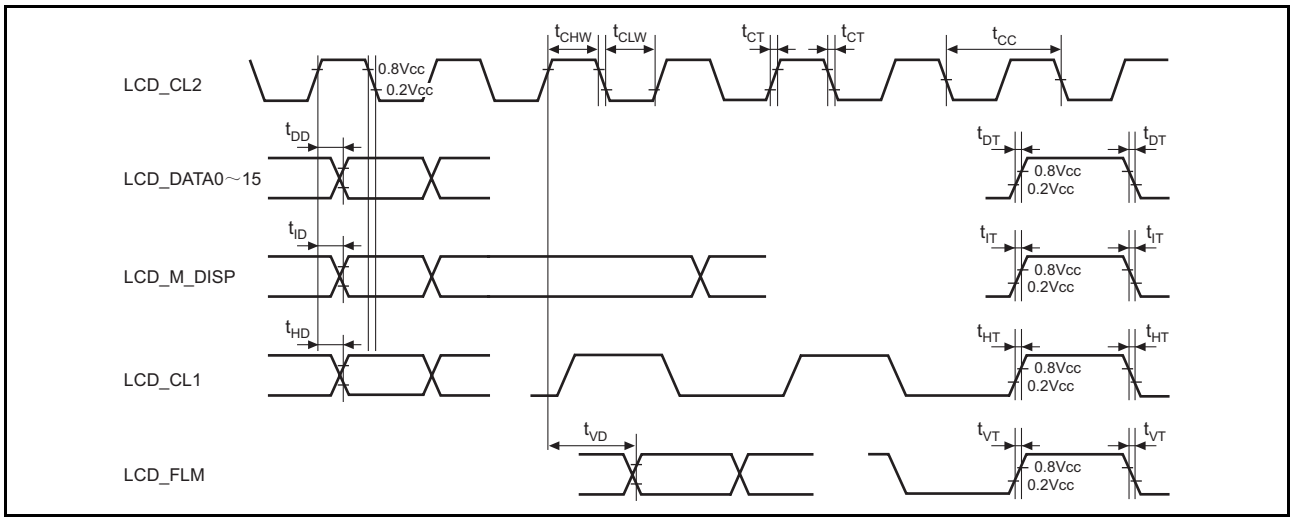


図 35.78 LCDC 模块时序

35.4.17 SDHI 时序

表 35.24 SDHI 时序

条件：V_{CC}=PLLV_{CC}=USBDV_{CC}=1.1 ～ 1.3V、 PV_{CC}=USBDPV_{CC}=3.0 ～ 3.6V、
AV$_{CC}$=3.0 ～ 3.6V、 USBAV$_{CC}$=1.1 ～ 1.3V、 USBAPV_{CC}=3.0 ～ 3.6V、 V_{SS}=PLLV_{SS}=PV_{SS}=AV$_{SS}$=USBDV_{SS}=
USBAV$_{SS}$=USBDPV_{SS}=USBAPV_{SS}=0V、 T_a=–40 ～ 85°C

项目	符号	Min.	Max.	单位	参考图
SD_CLK 时钟周期	t_{SDPP}	2×t_{pcyc}	—	ns	35.79
SD_CLK 时钟高电平宽度	t_{SDWH}	0.4×t_{SDPP}	—	ns	
SD_CLK 时钟低电平宽度	t_{SDWL}	0.4×t_{SDPP}	—	ns	
SD_CMD、SD_D3 ～ SD_D0 输出数据延迟（数据传送模式）	t_{SDODLY}	—	14	ns	
SD_CMD、 SD_D3 ～ SD_D0 输入数据设置	t_{SDISU}	5	—	ns	
SD_CMD、 SD_D3 ～ SD_D0 输入数据保持	t_{SDIH}	5	—	ns	

【注】 t_{pcyc} 表示外围时钟 （Pϕ）的 1 个周期时间。

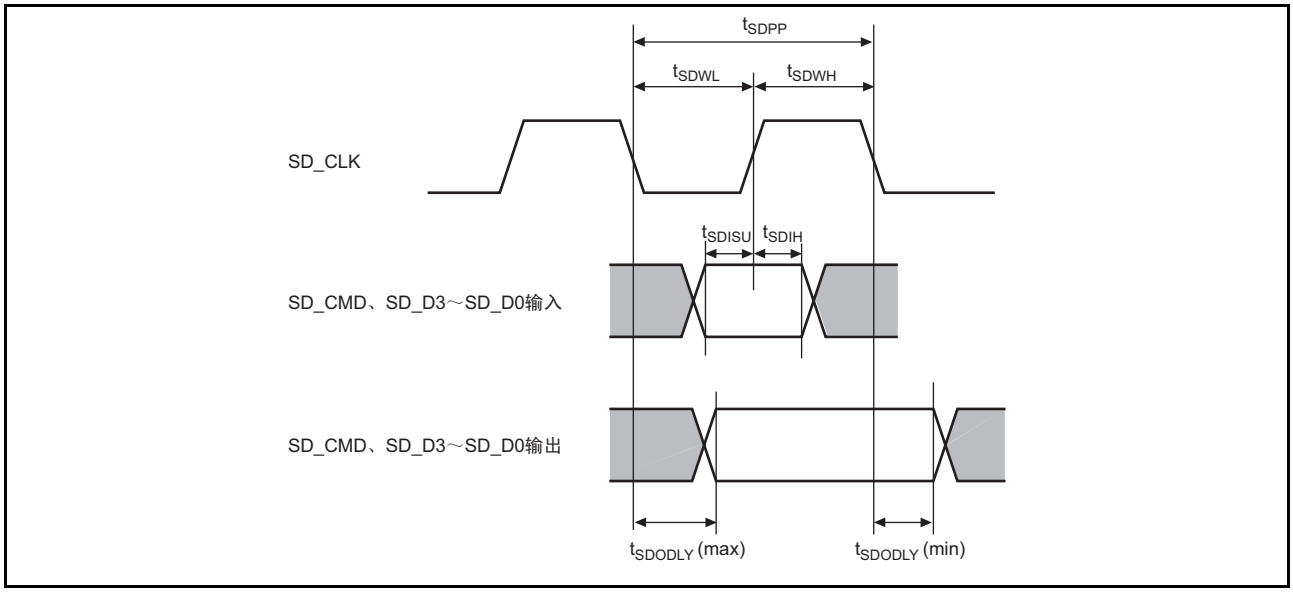


图 35.79 SD 卡接口

35.4.18 I/O 端口时序

表 35.25 I/O 端口时序

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、
AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=
$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目	符号	Min.	Max.	单位	参考图
输出数据延迟时间	t_{PORTD}	—	100	ns	35.80
输入数据设置时间	t_{PORTS}	100	—		
输入数据保持时间	t_{PORTH}	100	—		

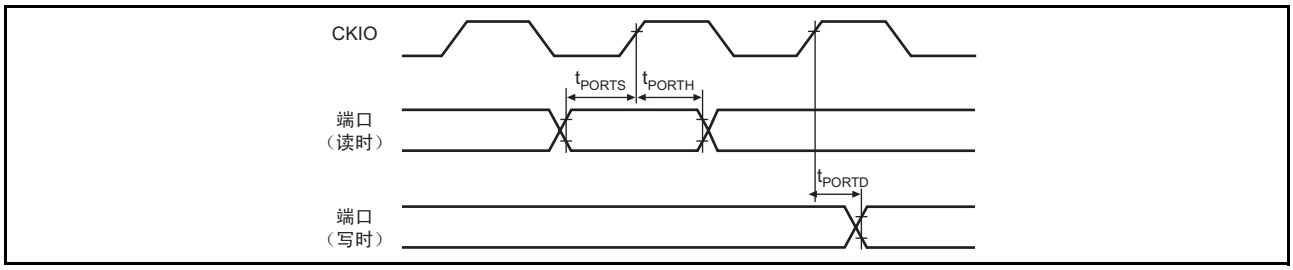


图 35.80 I/O 端口时序

35.4.19　H-UDI 时序

表 35.26　H-UDI 时序

条件：V_{CC}=PLLV_{CC}=USBDV_{CC}=1.1 ～ 1.3V、 PV_{CC}=USBDPV_{CC}=3.0 ～ 3.6V、
AV_{CC}=3.0 ～ 3.6V、 USBAV_{CC}=1.1 ～ 1.3V、 USBAPV_{CC}=3.0 ～ 3.6V、 V_{SS}=PLLV_{SS}=PV_{SS}=AV_{SS}=USBDV_{SS}=
USBAV_{SS}=USBDPV_{SS}=USBAPV_{SS}=0V、 T_a=–40 ～ 85°C

项目	符号	Min.	Max.	单位	参考图
TCK 周期时间	t_{TCKcyc}	50*	—	ns	35.81
TCK 高电平脉冲宽度	t_{TCKH}	0.4	0.6	t_{TCKcyc}	
TCK 低电平脉冲宽度	t_{TCKL}	0.4	0.6	t_{TCKcyc}	
TDI 设置时间	t_{TDIS}	10	—	ns	35.82
TDI 保持时间	t_{TDIH}	10	—	ns	
TMS 设置时间	t_{TMSS}	10	—	ns	
TMS 保持时间	t_{TMSH}	10	—	ns	
TDO 延迟时间	t_{TDOD}	—	16	ns	

【注】 *　请设置为大于外围时钟 （Pϕ）周期时间。

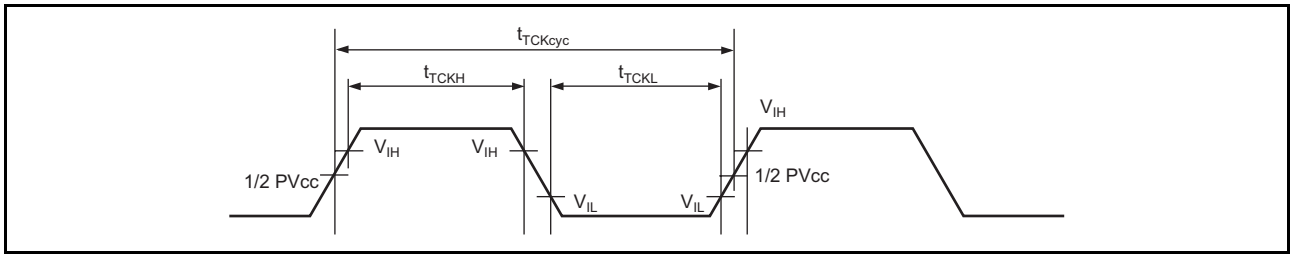


图 35.81　TCK 输入时序

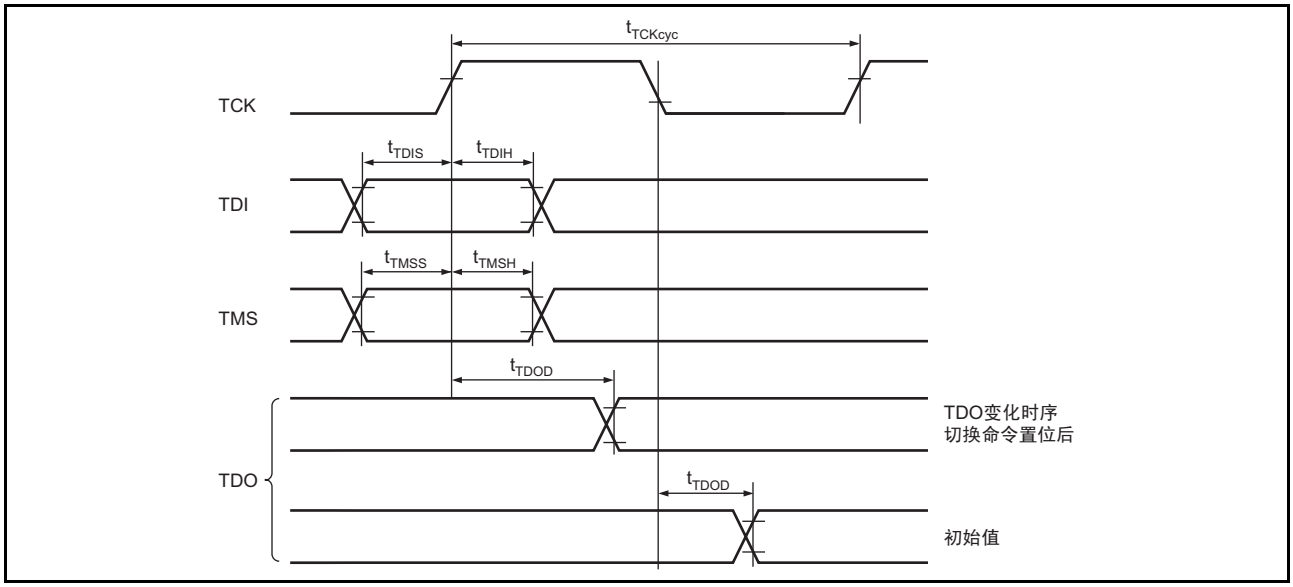
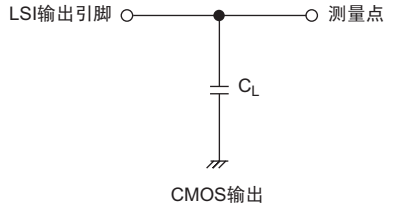


图 35.82　H-UDI 数据传送时序

35.4.20　AC 特性测量条件

- 输入/输出信号参考电平：$PV_{CC}/2$（PV_{CC}=3.0～3.6V、 V_{CC}=1.1～1.3V）
- 输入脉冲电平：PV_{SS}～3.0V（但$\overline{RES}$、$\overline{MRES}$、NMI、MD、MD_CLK1、MD_CLK0、$\overline{ASEMD}$、$\overline{TRST}$及施密特触发器输入引脚为PV_{SS}～PV_{CC}）
- 输入上升、下降时间：1ns



【注】 1. C_L是包含测试设备电容的合计值。设定各引脚如下。
30pF：CKIO、$\overline{RASU}$、$\overline{RASL}$、$\overline{CASU}$、$\overline{CASL}$、$\overline{CS0}$～$\overline{CS7}$、$\overline{BACK}$
50pF：上述以外的引脚
2. I_{OL}、I_{OH}是表35.3所示的值。

图 35.83　输出补充电路

35.5　A/D 转换器特性

表 35.27　A/D 转换器特性

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、
AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=
$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目	Min.	Typ.	Max.	单位
分辨率	10	10	10	位
转换时间	3.9	—	—	μs
模拟输入电容	—	—	20	pF
容许信号源阻抗	—	—	5	kΩ
非线性误差	—	—	± 3.0*	LSB
偏移误差	—	—	± 2.0*	LSB
满刻度误差	—	—	± 2.0*	LSB
量化误差	—	—	± 0.5*	LSB
绝对精度	—	—	± 4.0	LSB

【注】 *　参考值

35.6　D/A 转换器特性

表 35.28　D/A 转换器特性

条件：V_{CC}=$PLLV_{CC}$=$USBDV_{CC}$=1.1 ～ 1.3V、 PV_{CC}=$USBDPV_{CC}$=3.0 ～ 3.6V、AV_{CC}=3.0 ～ 3.6V、 $USBAV_{CC}$=1.1 ～ 1.3V、 $USBAPV_{CC}$=3.0 ～ 3.6V、 V_{SS}=$PLLV_{SS}$=PV_{SS}=AV_{SS}=$USBDV_{SS}$=$USBAV_{SS}$=$USBDPV_{SS}$=$USBAPV_{SS}$=0V、 T_a=–40 ～ 85°C

项目	Min.	Typ.	Max.	单位	测量条件
分辨率	8	8	8	位	
转换时间	10	—	—	μs	负载电容 20pF
绝对精度	—	± 2.0	± 3.0	LSB	负载电阻 2MΩ
	—	—	± 2.5	LSB	负载电阻 4MΩ

35.7　使用时的注意事项

请将多层陶瓷电容作为旁路电容连接至每对电源引脚。旁路电容请尽可能安装在 LSI 的电源引脚附近。电容的电容值请使用 0.1μF ～ 0.33μF （推荐值）。关于水晶振荡相关电容的详细内容请参考“4.6 电路板设计的注意事项”。

外接电容配置例如图 35.84 所示。

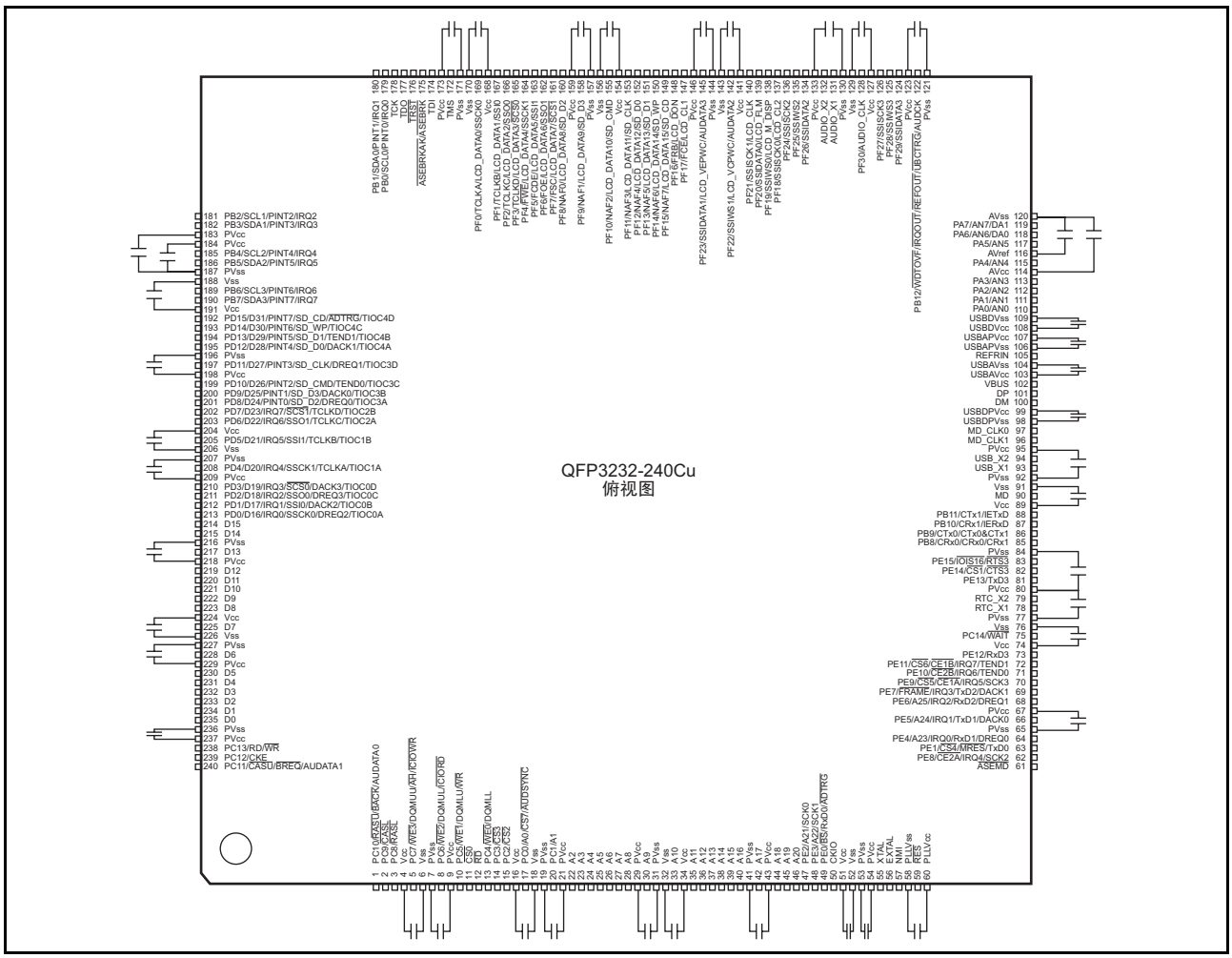


图 35.84　外接电容配置例

附录

附录 A. 引脚状态

表 A.1 引脚状态

引脚功能				引脚状态					
分类	引脚名			通常状态（右边情况除外）	复位状态		低功耗状态		总线权释放状态
					上电复位 *1	引脚状态保持 *2	深度待机模式 *3	软件待机模式	
时钟	EXTAL*4	时钟运行模式	0、1	I	I	I	Z	I	I
			2、3	Z	Z	Z	Z	Z	Z
	XTAL*4			O	O	O	L	L	O
	CKIO	时钟运行模式	0、1、3	O	O	O/Z*6*13	O/Z*6	O/Z*6	O/Z*6
			2	I	I	I	Z	I	I
系统控制	$\overline{\text{RES}}$			I	I	I	I	I	I
	$\overline{\text{MRES}}$			I	—	I/Z*10	I/Z*10	I	I
	$\overline{\text{WDTOVF}}$			O	H	H	H	H	O
	$\overline{\text{BREQ}}$			I	—	Z	Z	Z	I
	$\overline{\text{BACK}}$			O	—	Z	Z	Z	L
运行模式控制	MD			I	I	I	I	I	I
	MD_CLK1、MD_CLK0			I	I	I	I	I	I
	$\overline{\text{ASEMD}}$			I	I	I	I	I	I
中断	NMI			I	I	I	I	I	I
	IRQ7 ～ IRQ0（PB7 ～ PB0）			I	—	I	I	I	I
	IRQ7 ～ IRQ0（PD7 ～ PD0）			I	—	Z	Z	I	I
	IRQ7 ～ IRQ0（PE11 ～ PE4）			I	—	I/Z*10	I/Z*10	I	I
	PINT7 ～ PINT0（PB7 ～ PB0）			I	—	I	I	I	I
	PINT7 ～ PINT0（PD7 ～ PD0）			I	—	Z	Z	Z	I
	$\overline{\text{IRQOUT}}$			O	—	H/Z*7	H/Z*7	H/Z*7	O
UBC	$\overline{\text{UBCTRG}}$			O	—	O/Z*7	O/Z*7	O/Z*7	O
地址总线	A25 ～ A21、A0			O	—	O/Z*8	O/Z*8	O/Z*8	Z
	A20 ～ A2			O	O	O/Z*8*13	O/Z*8	O/Z*8	Z
	A1			O	O*5	O/Z*8*13	O/Z*8	O/Z*8	Z

引脚功能		引脚状态					
分类	引脚名	通常状态（右边情况除外）	复位状态		低功耗状态		总线权释放状态
			上电复位 *1	引脚状态保持 *2	深度待机模式 *3	软件待机模式	
数据总线	D31 ～ D16	I/O/Z	Z*5	Z	Z	Z	Z
	D15 ～ D0	I/O/Z	Z	Z	Z	Z	Z
总线控制	$\overline{CS0}$	O	H	H/Z*8*13	H/Z*8	H/Z*8	Z
	$\overline{CS7}$ ～ $\overline{CS1}$ 、 $\overline{CE1A}$ 、 $\overline{CE1B}$ 、 $\overline{CE2A}$ 、 $\overline{CE2B}$	O	—	H/Z*8	H/Z*8	H/Z*8	Z
	$\overline{RD}$	O	H	H/Z*8*13	H/Z*8	H/Z*8	Z
	RD/$\overline{WR}$	O	—	H/Z*8	H/Z*8	H/Z*8	Z
	$\overline{BS}$	O	—	H/Z*8	H/Z*8	H/Z*8	Z
	$\overline{FRAME}$	O	—	H/Z*8	H/Z*8	H/Z*8	Z
	$\overline{WAIT}$	I	—	Z	Z	Z	Z
	$\overline{WE3}$/DQMUU/$\overline{ICIOWR}$/$\overline{AH}$、$\overline{WE2}$/DQMUL/$\overline{ICIORD}$、$\overline{WE1}$/DQMLU/$\overline{WE}$、 $\overline{WE0}$/DQMLL	O	—	H/Z*8	H/Z*8	H/Z*8	Z
	$\overline{RASU}$、 $\overline{RASL}$、 $\overline{CASU}$、 $\overline{CASL}$	O	—	O/Z*9	O/Z*9	O/Z*9	O/Z*9
	CKE	O	—	O/Z*9	O/Z*9	O/Z*9	O/Z*9
	IOIS16	I	—	Z	Z	Z	I
	$\overline{REFOUT}$	O	—	H/Z*7	H/Z*7	H/Z*7	O
DMAC	DREQ3 ～ DREQ0	I	—	Z	Z	Z	I
	DACK3 ～ DACK0	O	—	O/Z*7	O/Z*7	O/Z*7	O
	TEND1、 TEND0	O	—	O/Z*7	O/Z*7	O/Z*7	O
MTU2	TCLKA、 TCLKB、 TCLKC、 TCLKD	I	—	Z	Z	Z	I
	TIOC0A、TIOC0B、TIOC0C、TIOC0D	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
	TIOC1A、 TIOC1B	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
	TIOC2A、 TIOC2B	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
	TIOC3A、TIOC3B、TIOC3C、TIOC3D	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
	TIOC4A、TIOC4B、TIOC4C、TIOC4D	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
RTC	RTC_X1*4	I/Z*11	I	I	I	I/Z*11	I/Z*11
	RTC_X2*4	O/H*11	O	O	O	O/H*11	O/H*11
SCIF	TxD3 ～ TxD0	O/Z	—	O/Z*7	O/Z*7	O/Z*7	O/Z
	RxD3 ～ RxD0	I	—	Z	Z	Z	I
	SCK3 ～ SCK0	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
	$\overline{RTS3}$	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
	$\overline{CTS3}$	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O

引脚功能		引脚状态					
分类	引脚名	通常状态（右边情况除外）	复位状态		低功耗状态		总线权释放状态
			上电复位 *1	引脚状态保持 *2	深度待机模式 *3	软件待机模式	
SSU	SSO1、 SSO0	I/O	—	Z	Z	Z	I/O
	SSI1、 SSI0	I/O	—	Z	Z	Z	I/O
	SSCK1、 SSCK0	I/O	—	Z	Z	Z	I/O
	$\overline{\text{SCS1}}$、 $\overline{\text{SCS0}}$	I/O	—	Z	Z	Z	I/O
IIC3	SCL3 ～ SCL0	I/O	—	I	I	I	I/O
	SDA3 ～ SDA0	I/O	—	I	I	I	I/O
SSI	SSIDATA3 ～ SSIDATA0	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
	SSISCK3 ～ SSISCK0	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
	SSIWS3 ～ SSIWS0	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
	AUDIO_CLK	I	—	Z	Z	Z	I
	AUDIO_X1*4	I/Z*12	I	I	Z	Z	I/Z*12
	AUDIO_X2*4	O/L*12	O	O	L	L	O/L*12
RCAN-TL1	CTx1、 CTx0	O	—	O/Z*7	O/Z*7	O/Z*7	O
	CRx1、 CRx0	I	—	Z	Z	Z	I
IEB	IETxD	O	—	O/Z*7	O/Z*7	O/Z*7	O
	IERxD	I	—	Z	Z	Z	I
ADC	AN7 ～ AN0	I	—	Z	Z	Z	I
	$\overline{\text{ADTRG}}$	I	—	Z	Z	Z	I
DAC	DA1、 DA0	O	—	Z	Z	O	O
FLCTL	FOE	O	—	O/Z*7	O/Z*7	O/Z*7	O
	FSC	O	—	O/Z*7	O/Z*7	O/Z*7	O
	$\overline{\text{FCE}}$	O	—	O/Z*7	O/Z*7	O/Z*7	O
	FCDE	O	—	O/Z*7	O/Z*7	O/Z*7	O
	FRB	I	—	Z	Z	Z	I
	$\overline{\text{FWE}}$	O	—	O/Z*7	O/Z*7	O/Z*7	O
	NAF7 ～ NAF0	I/O/Z	—	K/Z*7	K/Z*7	K/Z*7	I/O/Z
USB	DP、 DM	I/O	I/O	I/O	I/O	I/O	I/O
	VBUS	I	I	I	I	I	I
	REFRIN	I	I	I	I	I	I
	USB_X1*4	I	I	I	Z	Z	I
	USB_X2*4	O	O	O	L	L	O

引脚功能		引脚状态					
分类	引脚名	通常状态（右边情况除外）	复位状态		低功耗状态		总线权释放状态
			上电复位 *1	引脚状态保持 *2	深度待机模式 *3	软件待机模式	
LCDC	LCD_DATA15 ～ LCD_DATA0	O	—	O/Z*7	O/Z*7	O/Z*7	O
	LCD_DON	O	—	O/Z*7	O/Z*7	O/Z*7	O
	LCD_CL1、 LCD_CL2	O	—	O/Z*7	O/Z*7	O/Z*7	O
	LCD_M_DISP	O	—	O/Z*7	O/Z*7	O/Z*7	O
	LCD_FLM	O	—	O/Z*7	O/Z*7	O/Z*7	O
	LCD_VCPWC、 LCD_VEPWC	O	—	O/Z*7	O/Z*7	O/Z*7	O
	LCD_CLK	I	—	Z	Z	Z	I
SDHI	SD_CLK	O	—	O/Z*7	O/Z*7	O/Z*7	O
	SD_CMD	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
	SD_D3 ～ SD_D0	I/O	—	K/Z*7	K/Z*7	K/Z*7	I/O
	SD_CD	I	—	Z	Z	Z	I
	SD_WP	I	—	Z	Z	Z	I
I/O 端口	PA7 ～ PA0	I	Z	Z	Z	Z	I
	PB12	O	—	O/Z*7	O/Z*7	O/Z*7	O
	PB11 ～ PB8	I/O	Z	K/Z*7	K/Z*7	K/Z*7	I/O
	PB7 ～ PB0	I	I	I	I	I	I
	PC14 ～ PC2、 PC0	I/O	Z	K/Z*7	K/Z*7	K/Z*7	I/O
	PC1	I/O	Z*5	K/Z*7	K/Z*7	K/Z*7	I/O
	PD15 ～ PD0	I/O	Z*5	K/Z*7	K/Z*7	K/Z*7	I/O
	PE15 ～ PE0	I/O	Z	K/Z*7	K/Z*7	K/Z*7	I/O
	PF30 ～ PF0	I/O	Z	K/Z*7	K/Z*7	K/Z*7	I/O
H-UDI	$\overline{\text{TRST}}$	I	I	I	Z	I	I
	TCK	I	I	I	Z	I	I
	TDI	I	I	I	Z	I	I
	TDO	O/Z*14	O/Z14	O/Z14	O/Z14	O/Z14	O/Z14
	TMS	I	I	I	Z	I	I
仿真器 *15	$\overline{\text{AUDSYNC}}$	—	—	—	—	—	—
	AUDCK	—	—	—	—	—	—
	AUDATA3 ～ AUDATA0	—	—	—	—	—	—
	$\overline{\text{ASEBRKAK}}$/$\overline{\text{ASEBRK}}$	Z	Z	Z	Z	Z	Z

【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，输出引脚保持原来的状态

【注】 *1 通过给 $\overline{\text{RES}}$ 引脚输入低电平指定上电复位。通过 H-UDI 复位断言命令及 WDT 溢出进行上电复位时，各引脚的状态与通常运行时的初始引脚状态相同 （参照 “第 29 章 引脚功能控制器 （PFC）”）。

*2 通过 NMI、 $\overline{\text{MRES}}$ 、 IRQ7 ～ IRQ0 的各引脚输入从深度待机模式返回后，保持清除深度待机解除源标志寄存器 （DSFR）的 IOKEEP 位之前的状态 （参照 “第 32 章 低功耗模式”）。

*3 关闭内置于 I/O 引脚的弱保持电路。

*4 不使用连接晶体振荡器的引脚时，请上拉输入引脚 （EXTAL、 RTC_X1、 AUDIO_X1、 USB_X1），并将输出引脚 （XTAL、 RTC_X2、 AUDIO_X2、 USB_X2）置为开路。

*5 根据区域 0 的数据总线宽度，初始功能会有所不同 （参照 “第 29 章 引脚功能控制器 （PFC）”）

*6 取决于 CPG 的频率控制寄存器 （FRQCR）的 CKOEN 位的设定 （参照 “第 4 章 时钟脉冲振荡器 （CPG）”）。

*7 取决于待机控制寄存器 3 （STBCR3）的 HIZ 位的设定 （参照 “第 32 章 低功耗模式”）。

*8 取决于 BSC 的通用控制寄存器 （CMNCR）的 HIZMEM 位的设定 （参照 “第 9 章 总线状态控制器 （BSC）”）。

*9 按取决于 BSC 的通用控制寄存器 （CMNCR）的 HIZCNT 位的设定 （参照 “第 9 章 总线状态控制器 （BSC））。

*10 取决于深度待机解除源选择寄存器 （DSSSR）各位的设定 （参照 “第 32 章 低功耗模式”）。

*11 取决于 RTC 的 RTC 控制寄存器 （RCR2）的 RTCEN 位的设定 （“第 14 章 实时时钟 （RTC）”）。

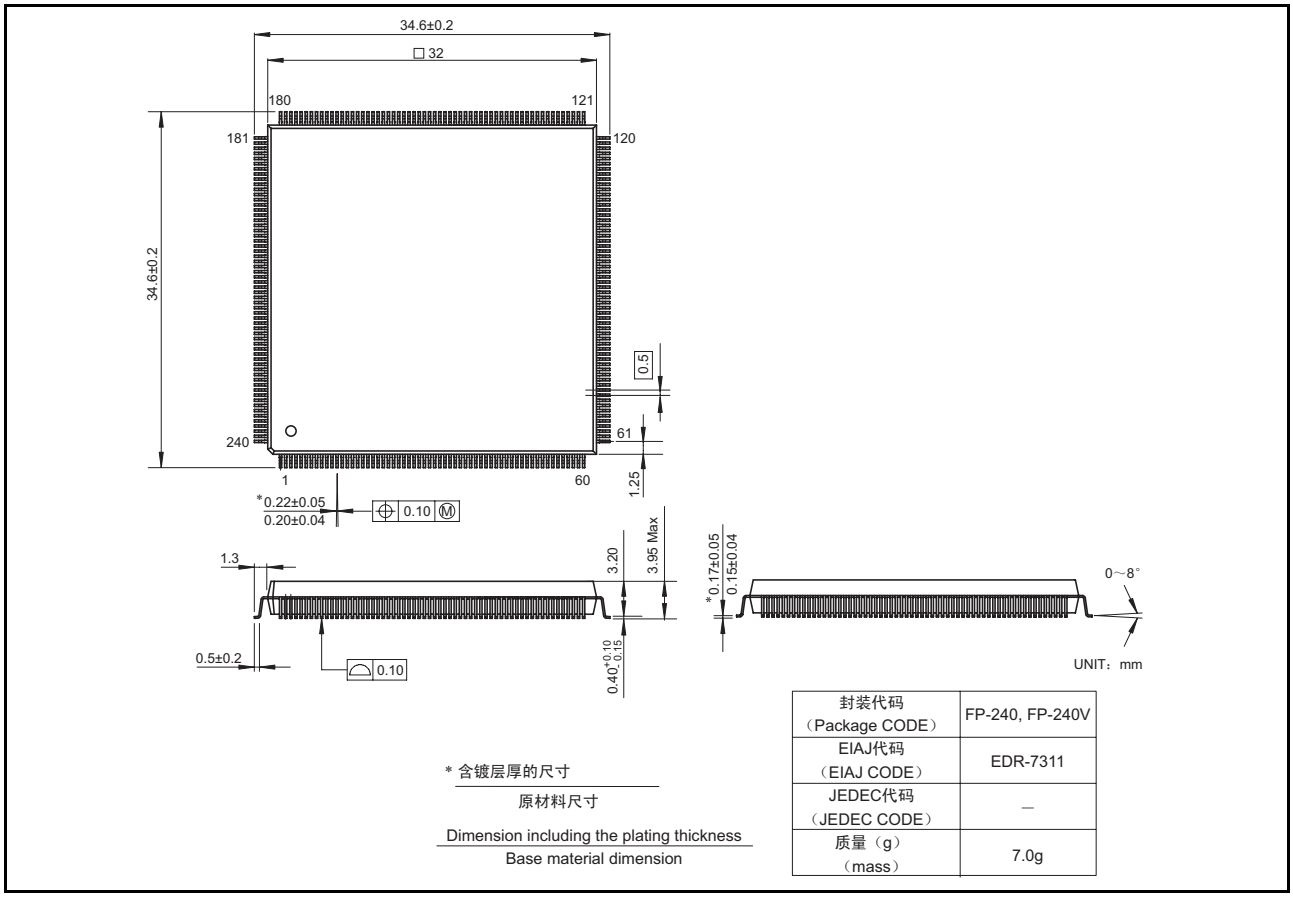
*12 取决于待机控制寄存器 （STBCR）的 AXTALE 位的设定 （参照 “第 32 章 低功耗模式”）。

*13 深度待机控制寄存器 2 （DSCTR2）的 CS0KEEPE 位为 1 时，保持深度待机模式中的引脚状态。 CS0KEEPE 位为 0 时，为上电复位的引脚状态 （参照 “第 32 章 低功耗模式”）。

*14 H-UDI 的 TAP 控制器不是 Shift-DR、 Shift-IR 的状态时为 Z。

*15 产品芯片模式时 （$\overline{\text{ASEMD}}$ =H）的引脚状态。关于 ASE 模式时 （$\overline{\text{ASEMD}}$ =L）的引脚状态，请参照 《仿真手册》。

附录 B. 外形尺寸图



封装代码（Package CODE）	FP-240, FP-240V
EIAJ代码（EIAJ CODE）	EDR-7311
JEDEC代码（JEDEC CODE）	—
质量（g）（mass）	7.0g

图 B.1 外形尺寸图

索引

数字

16 位 /32 位的位移量 37

A

A/D 转换精度的定义 815
A/D 转换开始请求的延迟功能 414
A/D 转换器的启动 421
A/D 转换器特性 1238
A/D 转换器 （ADC） 801
A/D 转换时间 （单通道模式） 813
A/D 转换时间 （多通道模式 / 扫描模式） 813
A/D 转换时序 812
AC 特性 1174
AC 特性测量条件 1238
ADC 时序 1223
AND/NAND 闪存控制器 （FLCTL） 823

B

BCHG 中断 915
BEMP 中断 911
BRDY 中断 907
伴随式计算功能 792
报警功能的使用方法 498
倍频率的变更 83
比较匹配定时器 （CMT） 468
变更 ID 排列 648

C

CAN 接口 632
CAN 睡眠模式 685
CAN 总线接口 704
CD-DSP 停止 / 再次开始时的注意点 800
CD-ROM 解码器 （ROM-DEC） 753
CMCNT 的计数时序 472
CMCNT 的写入和比较匹配的竞争 473
CMCNT 的字节写入和递增计数的竞争 474
CMCNT 的字写入和递增计数的竞争 473
CPU 30
$\overline{\text{CS}}$n 断言期间的扩展 210
采样频率转换器 （SRC） 995
槽非法指令 98
朝 0 舍入 72
测试模式的设定 688
乘法 / 乘加运算 36
乘加低位寄存器 （MACL） 33
乘加高位寄存器 （MACH） 33
程序计数器 （PC） 33
程序执行状态 64
冲突错误 567
处理产生条件 942
传输协议 709
传送率 579
传送时钟 557
串行位时钟控制 628
串行音频接口 （SSI） 604
从堆栈的返回 131
从时间 693
从属发送运行 592
从属接收运行 593
存储器的数据格式 34
存储器分配的高速缓存结构 157
存储体的对象寄存器和输入 / 输出方式 129
存储体的返回 130
存储体激活 229
存取长度和数据对齐 199
存取长度和突发数的关系 223
存取等待的控制 208
存取周期之间的空闲 255

D

D/A 转换器特性 1239
D/A 转换器 （DAC） 819
DC 特性 1169
DMAC 的启动 421
DMAC 接口 703
DMAC 时序 1214
DMAC 使用时的注意事项 543
DMA 传送流程图 279
DREQ 引脚的采样时序 294
DTCH 中断 915
带 FIFO 的串行通信接口 （SCIF） 501
待机控制电路 76
带字节选择的 SRAM 接口 242
单次读 226
单次写 228
单地址模式 290
单精度浮点格式 67
单通道模式 806
低功耗模式 1067
低功耗状态 64
低功率 SDRAM 238
低频率模式 235
地址错误 92
地址的多路复用 216
地址映射 164
地址阵列 149, 157
电特性 1167
电源控制顺序处理 980
掉电模式 236
读取地址阵列 157
读取数据阵列 158
断点的发送 543
断点的检测和处理 543
端法 199

端口 A 1046
端口 B 1048
端口 C 1050
端口 D 1053
端口 E 1056
端口 F 1059
对绝对精度的影响 818
多功能定时器脉冲单元 2 （MTU2） 298
多路复用一览表 （端口 A） 1008
多路复用一览表 （端口 B） 1008
多路复用一览表 （端口 C） 1009
多路复用一览表 （端口 D） 1010
多路复用一览表 （端口 E） 1011
多路复用一览表 （端口 F） 1012
多通道模式 808

E

ECC 错误校验 851
ECC 码 851
ECC 校正功能 792
EDC 校验功能 792

F

FLCTL 的中断请求 853
FLCTL 时序 1224
FPU 异常处理 73
FPU 异常源 73
发生紧接延迟转移指令之后的异常源 99
发送触发控制字段 645
发送触发时间 （TTT） 645
发送用内部仲裁 689
非规格化数 69
非数 （NaN） 69
非线性误差 815
非压缩模式 616
分频比的变更 83
分频器 1 76
分频器 2 76
浮点的范围 68
浮点寄存器 70
浮点异常 73
浮点运算单元 （FPU） 66
浮点运算指令 60, 99
复位顺序 684
复位同步 PWM 模式 384
复位状态 64

G

高速缓存 148
高速缓存的搜索 154
高速缓存与外部存储器匹配 156
高速缓存运行汇总 156
各运行模式下的寄存器状态一览 1165
固定模式 284
管道安排 942
管道控制 916
关于端口 A 的功能切换 1044
过程寄存器 （PR） 33

H

H-UDI 复位 1094
H-UDI 命令 1091
H-UDI 时序 1237
H-UDI 中断 115, 1094
互补 PWM 模式 387
缓冲存储器 921
缓冲格式 793
恢复中断 915
回写缓冲器 （仅限操作数高速缓存） 156

I

I/O 端口 1046
I/O 端口时序 1236
I^2C 总线格式 588
I^2C 总线接口 3 （IIC3） 574
IBUF 中断 798
IEBus 的位格式 716
IEBus™ 控制器 （IEB） 706
IEBus 通信协议 707
IERR 中断 797
IIC3 时序 1220
IREADY 中断 798
IRQ 中断 115
ISEC 中断 797
ISY 中断 797
ITARG 中断 797

J

寄存器
ABACK0 668
ABACK1 668
ADCSR 804
ADDRA ～ ADDRH 803
BAMR 138
BAR 137
BBR 140
BCR0 655
BCR1 653
BDMR 139
BDR 138
BEMPENB 879
BEMPSTS 886
BRCR 141
BRDYENB 877
BRDYSTS 884
CBUFCTL0 780

CBUFCTL1 781
CBUFCTL2 781
CBUFCTL3 781
CBUFST0 771
CBUFST1 771
CBUFST2 772
CCR 680
CCR1 150
CCR2 151
CFBCFG 865
CFIFO 867
CFIFOCTR 872
CFIFOSEL 868
CFIFOSIE 873
CHCR 268
CMAX_TEW 676
CMCNT 471
CMCOR 471
CMCSR 470
CMNCR 167
CMSTR 469
CROMCTL0 762
CROMCTL1 763
CROMCTL3 764
CROMCTL4 765
CROMCTL5 766
CROMEN 760
CROMST0 766
CROMST0M 782
CROMST1 767
CROMST3 767
CROMST4 768
CROMST5 769
CROMST6 770
CROMSY0 761
CS0WCR 171, 182, 193
CS1WCR 173
CS2WCR 175, 186
CS3WCR 175, 187
CS4WCR 176, 184
CS5WCR 178, 189
CS6WCR 180, 189, 191
CS7WCR 173
CSnBCR （n = 0 ～ 7） 168
CYCTR 682
D0FBCFG 865
D0FIFO 867
D0FIFOCTR 872
D0FIFOSEL 870
D0FIFOTRN 874
D1FBCFG 865
D1FIFO 867
D1FIFOCTR 872
D1FIFOSEL 870
D1FIFOTRN 874
DACR 820
DADR0 820
DADR1 820
DAR 267
DCPCFG 891
DCPCTR 893
DCPMAXP 892
DMAOR 273
DMARS0 ～ DMARS3 276
DMATCR 267
DSCTR 1077
DSCTR2 1078
DSFR 1080
DSRTR 1081
DSSSR 1079
DVSTCTR 862
FLADR 832
FLADR2 833
FLBSYCNT 840
FLBSYTMR 839
FLCMCDR 831
FLCMDCR 830
FLCMNCR 828
FLDATAR 835
FLDTCNTR 834
FLDTFIFO 841
FLECFIFO 841
FLINTDMACR 836
FLTRCR 842
FPSCR 71
FPUL 72
FRMNUM 887
FRQCR 81
GSR 652
HEAD00 772
HEAD01 772
HEAD02 773
HEAD03 773
HEAD20 776
HEAD21 776
HEAD22 777
HEAD23 777
IBCR 113
IBNR 114
ICCR1 578
ICCR2 580
ICDRR 586
ICDRS 587
ICDRT 586
ICIER 582
ICMR 581
ICR0 107
ICR1 108
ICR2 109
ICSR 584
IEAR1 723
IEAR2 723
IECKSR 737
IECMR 721

IECTR 720
IEFLG 728
IEIER 736
IEIET 732
IELA1 727
IELA2 727
IEMA1 725
IEMA2 726
IEMCR 722
IERB 738
IERBFL 727
IERCTL 726
IERSR 733
IESA1 724
IESA2 724
IETB 738
IETBFL 725
IETSR 730
IFCR 1043
IMR 661
INHINT 786
INTENB0 874
INTENB1 876
INTHOLD 785
INTSTS0 880
INTSTS1 882
IPR01、 IPR02、 IPR05 ～ IPR17 106
IRQRR 110
IRR 657
LDACLNR 961
LDCNTR 965
LDDFR 951
LDHCNR 956
LDHSYNR 957
LDICKR 948
LDINTR 961
LDLAOR 954
LDLIRNR 968
LDMTR 949
LDPALCR 955
LDPMMR 963
LDPR 955
LDPSPR 964
LDSARL 954
LDSARU 953
LDSMR 952
LDVDLNR 958
LDUINTLNR 967
LDUINTR 966
LDVSYNR 960
LDVTLNR 959
MBIMR0 671
MBIMR1 671
MCR 648
NF2CYC 587
NRDYENB 878
NRDYSTS 885
PADRL 1047
PBCRL1 1018
PBCRL2 1017
PBCRL3 1016
PBCRL4 1015
PBDRL 1048
PBIORL 1015
PBPRL 1050
PCCRL1 1022
PCCRL2 1021
PCCRL3 1020
PCCRL4 1019
PCDRL 1051
PCIORL 1019
PCPRL 1052
PDCRL1 1029
PDCRL2 1027
PDCRL3 1025
PDCRL4 1023
PDDRL 1054
PDIORL 1023
PDPRL 1055
PECRL1 1034
PECRL2 1033
PECRL3 1032
PECRL4 1031
PEDRL 1057
PEIORL 1030
PEPRL 1058
PFCRH1 1038
PFCRH2 1037
PFCRH3 1036
PFCRH4 1035
PFCRL1 1042
PFCRL2 1041
PFCRL3 1040
PFCRL4 1039
PFDRH 1060
PFDRL 1061
PFIORH 1035
PFIORL 1035
PFPRH 1062
PFPRL 1063
PINTER 111
PIPEBUF 898
PIPECFG 896
PIPEMAXP 899
PIPEnCTR （n = 1 ～ 7） 901
PIPEPERI 900
PIPESEL 895
PIRR 112
R64CNT 486
RCR1 494
RCR2 495
RCR3 496
RDAR 272
RDAYAR 492

RDAYCNT 488
RDMATCR 272
REC 662
RFMK 682
RFPR0 670
RFPR1 670
RFTROFF 677
RHRAR 491
RHRCNT 487
RMINAR 490
RMINCNT 487
RMONAR 492
RMONCNT 489
ROMDECRST 782
RSAR 272
RSECAR 490
RSECCNT 486
RSTSTAT 783
RTCNT 197
RTCOR 198
RTCSR 196
RWKAR 491
RWKCNT 488
RXPR0 669
RXPR1 669
RYRAR 493
RYRCNT 489
SAR （DMAC） 267
SAR （IIC3） 586
SCBRR 513
SCEMR 526
SCFCR 521
SCFDR 523
SCFRDR 504
SCFSR 509
SCFTDR 505
SCLSR 525
SCRSR 504
SCSCR 507
SCSMR 505
SCSPTR 523
SCSR 1043
SCTSR 505
SDBPR 1090
SDCR 194
SDIR 1091
SHEAD00 773
SHEAD01 774
SHEAD02 774
SHEAD03 774
SHEAD04 775
SHEAD05 775
SHEAD06 775
SHEAD07 776
SHEAD20 777
SHEAD21 778
SHEAD22 778
SHEAD23 778
SHEAD24 779
SHEAD25 779
SHEAD26 779
SHEAD27 780
SRCCTRL 1000
SRCID 996
SRCIDCTRL 998
SRCOD 997
SRCODCTRL 999
SRCSTAT 1002
SSCR2 554
SSCRH 548
SSCRL 549
SSER 551
SSI 784
SSICR 608
SSIRDR 615
SSISR 612
SSITDR 615
SSMR 550
SSRDR0 ～ SSRDR3 556
SSSR 552
SSTDR0 ～ SSTDR3 555
SSTRSR 557
STBCR 1068
STBCR2 1069
STBCR3 1070
STBCR4 1071
STBCR5 1072
STBCR6 1073
STRMDIN0 787
STRMDIN2 787
STRMDOUT0 787
SYSCFG 859
SYSCR1 1074
SYSCR2 1075
SYSCR3 1076
SYSSTS 860
TADCOBRA_4 339
TADCOBRB_4 339
TADCORA_4 339
TADCORB_4 339
TADCR 337
TBTER 356
TBTM 334
TCBR 353
TCDR 353
TCMR0 683
TCMR1 683
TCMR2 683
TCNT 340
TCNTR 681
TCNTS 352
TCR 306
TDDR 352
TDER 357

TEC 662
TESTMODE 864
TGCR 351
TGR 340
TICCR 335
TIER 329
TIOR 312
TITCNT 355
TITCR 354
TMDR 310
TOCR1 345
TOCR2 347
TOER 344
TOLBR 350
TRWER 343
TSR 331, 678
TSTR 341
TSYCR 336
TSYR 342
TTCR0 674
TTTSEL 684
TWCR 358
TXACK0 667
TXACK1 667
TXCR0 666
TXCR1 666
TXPR0 665
TXPR1 665
UFRMNUM 888
UMSR0 672
UMSR1 672
USBACSWR 902
USBADDR 889
USBINDX 890
USBLENG 890
USBREQ 889
USBVAL 890
WRCSR 478
WTCNT 476
WTCSR 477
寄存器的数据格式 34
寄存器地址一览
（按功能模块、手册章节编号顺序） 1096
寄存器位一览 1118
寄存器组 33, 129
寄存器组错误 93
寄存器组错误异常处理 93, 131
寄存器组异常 131
级联运行 369
间隔定时器模式的用法 482
晶体振荡器 76
晶体振荡器使用时的注意事项 84
就近舍入 72
局部接收过滤器屏蔽 （LAFM） 642
决定空闲周期数的项目 255
绝对地址 37
绝对地址的参照 37
绝对最大额定值 1167

K

看门狗定时器模式的用法 481
看门狗定时器 （WDT） 475
控制寄存器的初始值 33
控制器局域网 （RCAN-TL1） 630
控制信号时序 1178
控制传送阶段转移中断 913

L

LCDC 时序 1234
LCD 控制器 （LCDC） 945
LRU 149
立即数 36
立即数的参照 36
立即数的数据格式 34
量化误差 815
逻辑运算指令 55

M

MPX-I/O 接口 211
MTU2 的功能一览表 299
MTU2 时序 1215
MTU2 输出引脚的初始化方法 444
MTU2 中断源 419
满刻度误差 815
命令存取模式 845
模块待机功能 1088
模块待机模式的设定 573
模拟电压的设定 816, 822
模拟输入引脚的处理 816
模拟输入引脚的规格 817
目标扇区缓冲功能 795

N

NMI 中断 115
NRDY 中断 909
NYET 信号交换响应表 934
内置 RAM （保持用）地址空间 1065
内置 RAM （高速）地址空间 1065
内置外围模块请求模式 281
内置外围模块中断 116
能支持的 DMA 传送 287

P

PCMCIA 接口 247
PINT 中断 116
PLL 电路 76
PLL 振荡电路使用时的注意事项 85
PWM 模式 373

批量传送 933
偏移误差 815
频率变更的步骤 480
频率的变更方法 83

Q

器件状态转移中断 911
清除中断源的时序 133
取指令周期的断点 143
全局基址寄存器 （GBR） 32

R

RCAN-TL1 的存储器映像 634
RCAN-TL1 的定时器寄存器 673
RCAN-TL1 的控制寄存器 648
RCAN-TL1 的设定 684
RCAN-TL1 的中断源 702
RCAN-TL1 时序 1222
RCAN-TL1 引脚端口设定 704
RCAN-TL1 邮箱寄存器 663
RISC 方式 35
RTC 用晶体振荡电路 500
容许信号源阻抗 818
软件待机模式 1082
软件待机模式解除的步骤 （WDT） 480
软件待机模式时的 D/A 输出保持功能 822

S

SACK 中断 915
SCIF 时序 1216
SCIF 中断源 542
SDHI 时序 1235
SDRAM 接口 213
SD 主机接口 （SDHI） 1007
SIGN 中断 915
SOF 插值功能 941
SSI 模块的总线格式 616
SSI 时序 1221
SSU 模式 560
SSU 时序 1217
SSU 中断源 573
扫描模式 810
扇区存取模式 849
上电复位 90
上电顺序 236
舍入 72
深度待机模式 1085
深度掉电模式 240
时戳 644
时间触发发送 690
时间触发控制 （TT 控制） 645
时间触发系统例 697
实时时钟 （RTC） 484
使用过采样时钟时的注意事项 629
时钟脉冲振荡器 （CPG） 74
时钟频率的控制电路 76
时钟时序 1175
时钟同步串行格式 595
时钟同步模式中的运行 535
时钟运行模式 78
手动复位 91
输出补充电路 1238
数据存取周期的断点 144
数据 PID 顺序位 919
数据阵列 149, 158
数据传送指令 50
输入数据流字节序变换功能 788
刷新请求和总线周期的关系 235
双地址模式 288
双精度浮点格式 67
睡眠模式 1082
算术运算指令 53

T

TAP 控制器 1092
TDO 输出时序 1093
TSG 与 TSEG 的设定 656
TTW[1:0] （时间触发窗口） 645
T 位 36
调色板数据格式 976
停机模式 685
同步串行通信单元 （SSU） 545
同步码保护功能 789
同步传送 936
通常空间接口 203
通电、断电顺序 1168
通过以下公式求 SCBRR 的设定值 513
通过中断请求信号的数据传送 132
通用寄存器 30
通用寄存器的初始值 33
突发读 223
突发 MPX-I/O 接口 251
突发模式 292
突发 ROM （时钟同步）接口 254
突发 ROM （时钟异步）接口 241
突发写 226

U

UBC 时序 1214
USB2.0 主机 / 功能模块 （USB） 854
USB 时序 1232
USB 数据总线电阻控制 904
USB 通信允许的状态下为设定禁止的寄存器 919

V

VBUS 中断 915

W

WDT 时序 1216
外部触发输入时序 813
外部请求模式 280
外部时钟输入时的注意事项 84
位操作指令 63
微处理单元接口 （MPI） 632
位同步电路 602
位移量的参照 37
无延迟槽的无条件转移指令 36

X

系统寄存器的初始值 33
系统矩阵 647
系统控制指令 58
陷阱指令 97
显示分辨率的设定 980
显示 OFF 模式 （LCDC 停止）的注意事项 984
向存储体的保存 130
向堆栈的保存 131
向量基址寄存器 （VBR） 32
相位计数模式 378
写入地址阵列 （没有关联） 157
写入地址阵列 （有关联） 157
写入数据阵列 158
信息发送请求 689, 696
信息接收顺序 699
信息控制字段 639
信息数据字段 643
选择功能控制器功能时的控制传送 932
选择主机控制器功能时的控制传送 931
循环模式 284
寻址方式 38

Y

延迟转移指令 35
液晶模块电源状态 983
页竞争 1066
一般非法指令 98
异步模式的接收数据采样时序和接收容限 544
异步模式中的运行 528
异常处理 86
异常处理的种类和优先级 86
异常处理后堆栈的状态 100
异常处理向量表 88, 117
异常处理向量表地址的计算方法 89
异常处理状态 64
移位指令 56
引脚功能控制器 （PFC） 1008
引脚状态 1240
硬件旋转运行 984
用户调试接口 （H-UDI） 1089
用户断点控制器 （UBC） 134
用户断点运行的流程 143
用户断点中断 115
有关 FPU 的 CPU 指令 62
有关振荡器的注意事项 84
邮箱 632, 635
邮箱的再设定 700
邮箱的作用 636
邮箱功能的设定 641
邮箱控制 632
有效地址的计算方法 38
由指令引起的异常 97
预取操作 （仅限操作数高速缓存） 155

Z

噪声消除电路 597
朝 0 舍入 72
帧更新中断 914
整数除法指令 98
直接存储器存取控制器 （DMAC） 262
指令格式 42
指令特点 35
指令系统 45
中断控制器 （INTC） 102
中断响应时间 124
中断异常处理 96
中断异常处理结束后的堆栈状态 123
中断优先级 95
中断传送 935
周期挪用模式 291
周期设定的注意事项 431
主发送运行 589
主接收运行 590
转移表基址寄存器 （TBR） 32
转移指令 57
装入 - 存储结构 35
状态寄存器 （SR） 31
自动解码停止功能 793
自动请求模式 280
自动刷新 233
字数据的符号扩展 35
自刷新 234
总线权释放状态 64
总线时序 1180
总线仲裁 259
总线状态控制器 （BSC） 160

修订记录	SH7263 群数据表

Rev.	发行日	修订内容	
		页	修订处
1.00	2008.09.12	—	初版发行

瑞萨 32 位 RISC 单片机
硬件手册
SH7263 群

Publication Date: Rev1.00, Sep.12, 2008
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

©2008. Renesas Technology Corp., All rights reserved. Printed in Japan.

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) 6887-7858/7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2377-3473

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

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, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

SH7263群

瑞萨电子株式会社

RCJ09B0062-0100