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

RENESAS

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

SH7147群

瑞萨32位RISC单片机

SuperH™ RISC engine族

SH7147 R5F7147
SH7142 R5F7142

瑞萨电子

www.renesas.com

Rev.1.00 2009.03

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。

产品使用时的注意事项

本文对适用于单片机所有产品的 “使用时的注意事项”加以说明。有关个别使用时的注意事项请参阅正文。此外，如果记载与本手册的正文有差异之处，以正文为准。

1. 未使用的引脚的处理
   【注意】将未使用的引脚按照正文的 “未使用引脚的处理”进行处理。
   CMOS产品的输入引脚一般为高阻抗。如果于开路状态运行未使用的引脚，因感应现象，外加LSI周围的噪声，会在LSI内部产生穿透电流，有可能被误认为是输入信号而引起误动作。未使用的引脚，请按照正文的 “未使用引脚的处理”中的指示进行处理。

2. 通电时的处理
   【注意】通电时产品处于不定状态。
   通电时， LSI内部电路处于不确定状态，寄存器的设定和各引脚的状态不定。外部复位引脚对产品进行复位时，通电到复位有效之间，不能保证引脚的状态。
   与此相同，由内部上电复位功能对产品进行复位时，通电到复位产生规定电压的期间，不能保证引脚的状态。

3. 禁止存取保留地址 （保留区）
   【注意】禁止存取保留地址 （保留区）
   地址区域分配有用于功能扩展的保留地址 （保留区）。由于无法保证存取此地址时的运行，不能对保留地址 （保留区）进行存取。

4. 关于时钟
   【注意】复位时，待时钟稳定后可解除复位。
   在程序运行中切换时钟时，待目标时钟稳定后切换。复位时，使用外部振荡器 （或外部振荡电路）的时钟开始运行的系统，必须在时钟确保稳定后解除复位。另外，在程序运行中，切换为使用外部振荡器 （或者外部振荡电路）的时钟时，待目标时钟确保稳定后再切换。

5. 关于产品之间的差异
   【注意】更换不同型号的产品时，需事先检测，确保各型号产品的兼容性。
   即使是同一个群的单片机，如果产品型号不同，由于内部ROM、布线设计等不同，产品特性也会出现不同。因此，更换不同型号的产品时，必须对每一个型号的产品进行系统评估测试。

本手册的结构

本手册的内容结构如下：

1. 产品使用时的注意事项
2. 本手册的结构
3. 前言
4. 目录
5. 概要
6. 各功能模块的说明
   - CPU 及系统控制
   - 内部外围模块

     各模块功能说明的结构因模块而异，一般是由：
     ①特点、②输入 / 输出引脚、③寄存器说明、④运行说明、⑤使用时的注意事项等章节构成。

     采用本 LSI 设计应用系统时，首先需要充分确认注意事项。
     各章正文记载有关于说明的注意事项，各章的末尾记载使用时的注意事项。必须仔细阅读。
     （根据需要各章节记载使用时的注意事项）。

7. 寄存器一览表
8. 电特性
9. 附录
   - 产品型号、封装尺寸图等
10. 本版的修订部分 （仅适用于修订版）

    修订履历汇总了修正 / 追加前版内容的主要部分，并未记载所有修订内容，详情参阅本手册正文。

11. 索引

前言

SH7147 采用瑞萨科技独创的 RISC 结构 CPU 内核，集成了系统结构所需的外围功能，是一款单芯片精简指令结构的单片机。

对象　本手册以使用 SH-7147 设计应用系统的用户为对象。使用本手册的读者需具备电路、逻辑电路及单片机的相关基础知识。

目的　本手册以用户了解 SH-7147 的硬件功能和电特性为目的。
有关执行指令，详情参阅 “SH-1/SH-2/SH-DSP 软件手册”。

阅读方法

- 希望了解全部功能时
  → 请按照目录阅读。
  本手册大致以CPU、系统控制功能、外围功能、电特性的顺序构成。
- 希望了解CPU功能的详细内容时
  → 请参阅 “SH-1/SH-2/SH-DSP 软件手册”。
- 已知寄存器名称，希望了解详细功能时
  → 本手册后面附有 “索引”。请从索引查找页码。
  “第 24 章 寄存器一览表”汇总了地址、位、初始化等内容。

凡例

寄存器的表示方法	：	串行通信等相同或类似功能存在于多个通道时，使用以下表示方法。 XXX_N （XXX 表示基本寄存器名称， N 表示通道号）
位的表示方法	：	按左侧为高位、右侧为低位的顺序表示。
数字的表示方法	：	B'xxxx 表示 2 进制数、 H'xxxx 表示 16 进制数、 xxxx 表示 10 进制数。
信号的表示方法	：	低电平有效信号加上划线。 $\overline{\text{xxxx}}$

相关资料一览表　网站登载有最新资料。确认持有的资料是否为最新版本。
(http://www.renesas.com)

- SH-7147相关的用户手册

资料名称	资料编号
SH-7147 群 硬件手册	本手册
SH-1/SH-2/SH-DSP 软件手册	RCJ09B0065

- 开发工具相关的用户手册

资料名称	资料编号
SuperH RISC engine C/C++ Compiler,Assembler,Optimizing Linkage Editor Compiler Package V.9.00 User's Manual	REJ10B0152
SuperH RISC engine High-performance Embedded Workshop 3 User's Manual	REJ10B0025
SuperH RISC engine High-Performance Embedded Workshop3 Tutorial User's Manual	REJ10B0023

- 应用手册

资料名称	资料编号
SuperH RISC engine C/C++ Compiler Package Application Note	REJ05B0463

目　录

第 1 章　概要

1.1　特点

本 LSI 采用瑞萨科技独创的 RISC 结构 CPU 内核，集成了系统结构所需的外围功能，是一款单芯片精简指令结构的单片机。

本 LSI 的 CPU 有 RISC （Reduced Instruction Set Computer）结构的指令系统，基本指令以 1 条指令 1 个状态 （1 个系统时钟周期）运行，因此指令执行速度得以飞速提升，采用内部 32 位结构，加强了数据处理能力。通过本 LSI 的 CPU 可实现以往单片机未能实现的、需要高速实时控制的应用功能，并可组建成本更低、高性能 / 高功能的系统。

此外，本 LSI 内置大容量 ROM、 RAM、数据传送控制器 （DTC）、定时器、串行通信接口 （SCI）、同步串行通信单元 （SSU）、 A/D 转换器、中断控制器 （INTC）、 I/O 端口及控制器局域网 （RCAN-ET）等系统构成所需的外围功能。

同时，通过支持外部存储器存取功能，可与存储器或外围 LSI 直接连接。

由此，可大幅度降低系统成本。

内部 ROM 为内置闪存的 F-ZTAT^{TM} （Flexible Zero Turn Around Time）版 *。除了能够使用支持本 LSI 编程的编程器对闪存编写程序外，还能通过软件编程 / 擦除。因此，用户可在本 LSI 嵌入电路板的状态下改写程序。

本 LSI 的特点如表 1.1 所示。

【注】 * F-ZTAT^{TM} 是株式会社瑞萨科技的商标。

- 产品一览表

型号名称	ROM 容量	RAM 容量	CAN 通道数	工作环境温度 （Topr）	最大工作频率	电源电压		
						V_{CC}	PV_{CC}	AV_{CC}
R5F71474BJ80FPV	256KB	16KB	1ch	－40 ～ +85°C	80MHz	5.0V±0.5V	5.0V±0.5V	5.0V±0.5V
R5F71474BD80FPV								
R5F71474AK64FPV		12KB		－40 ～ +125°C	64MHz	3.3V±0.3V		
R5F71475BJ80FPV	384KB	16KB		－40 ～ +85°C	80MHz	5.0V±0.5V		
R5F71475AK64FPV				－40 ～ +125°C	64MHz	3.3V±0.3V		
R5F71476BJ80FPV	512KB			－40 ～ +85°C	80MHz	5.0V±0.5V		
R5F71476BD80FPV								
R5F71476AK64FPV				－40 ～ +125°C	64MHz	3.3V±0.3V		
R5F71424BJ80FPV	256KB		2ch	－40 ～ +85°C	80MHz	5.0V±0.5V		
R5F71424AK64FPV		12KB		－40 ～ +125°C	64MHz	3.3V±0.3V		
R5F71426BJ80FPV	512KB	16KB		－40 ～ +85°C	80MHz	5.0V±0.5V		
R5F71426BD80FPV								
R5F71426AK64FPV				－40 ～ +125°C	64MHz	3.3V±0.3V		

- 通用输入/输出端口

输入 / 输出端口	输入端口
57 个	16 个

- 封装

封装	封装代码	封装尺寸	引脚节距
LQFP-100	FP-100UV	14.0×14.0mm	0.5mm

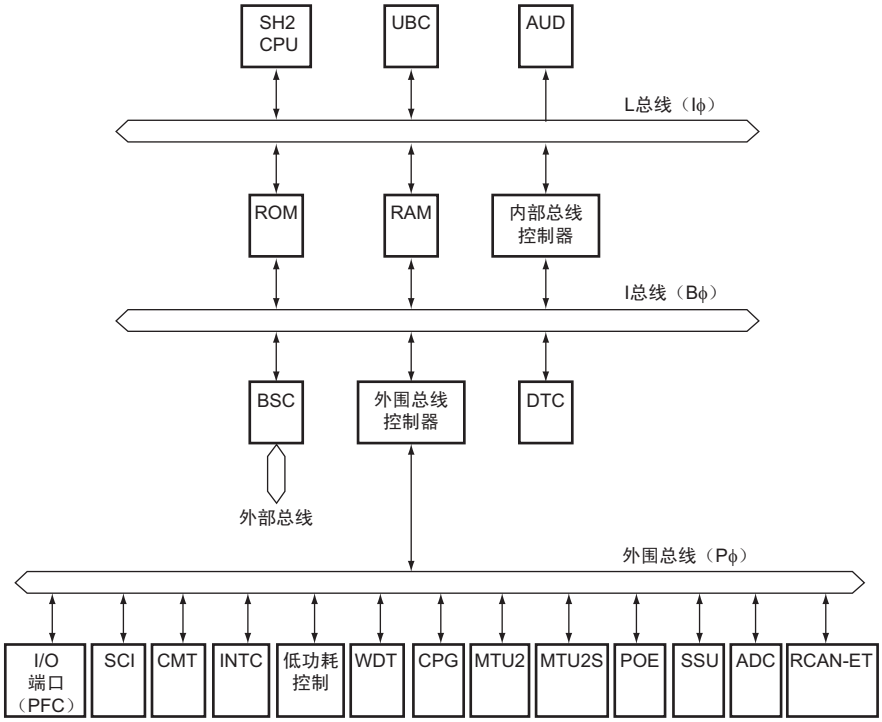
表 1.1 特点

分类	特 点
CPU	• 32 位 RISC （Reduced Instruction Set Computer）结构 CPU • 指令长度：固定为 16 位，以提高编码效率 • 加载存储结构 （在寄存器间执行基本运算） • 通用寄存器：32 位 ×16 个 • 流水线：5 段流水线方式 • 内置乘法器：以 2 ～ 5 个周期执行 32×32→64 乘法运算 • 基本指令：62 种。面向 C 语言的指令系统 【注】 请注意，槽非法指令的规格与以往的 SH-2 有所不同。详情参阅 “5.8.4 槽非法指令异常处理相关的注意事项”。
运行模式	• 运行模式 单芯片模式 扩展 ROM 有效模式 扩展 ROM 无效模式 • 处理状态 程序执行状态 异常处理状态 总线权释放状态 • 低功耗状态 睡眠模式 软件待机模式 深度软件待机模式 硬件待机模式 模块待机模式
用户断点控制器 （UBC）	• 地址、数据值、存取方式、数据长度均可设定为断点条件 • 支持顺次断点功能 • 2 个断点通道
内部 ROM	• 256K 字节 /384K 字节 /512K 字节 （参照产品一览表）
内部 RAM	• 12K 字节 /16K 字节 （参照产品一览表）
总线状态控制器 （BSC）	• 分别支持最大 64M 字节的 2 个区域 （CS0、 CS1）地址空间 • 外部总线 8 位 • 各区域可独立设定以下功能： 存取等待周期数 设定空闲等待周期 支持 SRAM • 对相应区域输出片选信号

分类	特　点
数据传送控制器 （DTC）	• 根据外围 I/O 中断请求，可与 CPU 独立传送数据 • 各中断源均可设定传送模式 （在存储器设定传送模式） • 对 1 个启动源，可传送多个数据 • 丰富的传送模式 可选择普通模式 / 重复模式 / 块传送模式 • 传送单位可设定为字节 / 字 / 长字 • 可对 CPU 请求启动 DTC 的中断 一次数据传送结束后，可对 CPU 产生中断 • 结束所有指定的数据传送后，可对 CPU 产生中断
中断控制器 （INTC）	• 5 个外部中断引脚 （NMI、 IRQ3 ～ IRQ0） • 内部外围中断：可对每个模块设定优先级 • 向量地址：每个中断源均有固定的向量地址
高级用户调试器 （AUD）	• RAM 监控模式 AUDCK 输入时钟：≤ 10MHz 且≤ 1/4 总线时钟 （Bϕ） 可读取 / 写入连接在内部 / 外部总线的模块 • 转移跟踪模式 AUD 工作频率：≤ 20MHz 通过内部时钟 （Iϕ）的时钟比支持 ×1、 ×1/2、 ×1/4、 ×1/8 • 输出转移地址
时钟振荡器 （CPG）	• 时钟模式：可从外部输入、晶体谐振器选择输入时钟 • 生成 5 种时钟 CPU 时钟：80MHz （Max.）（T_{opr}= – 40 ～ +85°C） CPU 时钟：64MHz （Max.）（T_{opr}= – 40 ～ +125°C） 总线时钟：40MHz （Max.）（T_{opr}= – 40 ～ +85°C） 总线时钟：32MHz （Max.）（T_{opr}= – 40 ～ +125°C） 外围时钟：40MHz （Max.）（T_{opr}= – 40 ～ +85°C） 外围时钟：32MHz （Max.）（T_{opr}= – 40 ～ +125°C） MTU2 专用时钟：40MHz （Max.）（T_{opr}= – 40 ～ +85°C） MTU2 专用时钟：32MHz （Max.）（T_{opr}= – 40 ～ +125°C） MTU2S 专用时钟：80MHz （Max.）（T_{opr}= – 40 ～ +85°C） MTU2S 专用时钟：64MHz （Max.）（T_{opr}= – 40 ～ +125°C）
看门狗定时器 （WDT）	• 1 个通道的看门狗定时器 • 可请求中断

分类	特 点
多功能定时器脉冲单元 2 （MTU2）	• 基于 5 个通道的 16 位定时器，最多可输入 / 输出 16 种脉冲 • 18 个输出比较兼输入捕捉寄存器 • 总计 18 个独立的比较器 • 可选择 8 种计数器输入时钟 • 输入捕捉功能 • 脉冲输出模式 单触发 / 交替 /PWM/ 互补 PWM/ 复位同步 PWM • 多个计数器同步功能 • 互补 PWM 输出模式 输出控制 6 相变频器的非重叠波形 自动设定空载时间 可在 0 ～ 100% 内任意设定 PWM 占空比 输出 OFF 功能 A/D 转换请求延迟功能 波峰 / 波谷中断跳过功能 • 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 可执行 2 相编码器计数处理
多功能定时器脉冲单元 2S （MTU2S）	• MTU2 通道 3、 4 的子集版 • 补偿与内部通道 5 （仅 MTU2S）联动的空载时间 • 最大工作频率为 80MHz （T_{opr}= – 40 ～ +85°C） • 最大工作频率为 64MHz （T_{opr}= – 40 ～ +125°C）
端口输出允许 （POE）	• 控制 MTU2/MTU2S 波形输出引脚的高阻抗
比较匹配定时器 （CMT）	• 16 位计数器 • 产生比较匹配中断 • 2 个通道
串行通信接口 （SCI）	• 时钟同步 / 异步模式 • 3 个通道
同步串行通信单元 （SSU）	• 可选择主模式与从属模式 • 可选择标准模式 / 双向模式 • 可选择发送 / 接收数据长度为 8/16/32 位 • 可同时发送 / 接收数据 （全双工） • 可连续执行串行通信 • 1 个通道
控制器局域网 （RCAN-ET）	• CAN 版本：支持 Bosch 2.0B active • 缓冲器容量：发送 / 接收 ×15 个；接收专用 ×1 个 • 1 个通道 /2 个通道 （参照产品一览表）
A/D 转换器 （ADC）	• 12 位 ×16 个通道 • 可通过外部触发或 MTU2/MTU2S 请求转换 • 内置 3 个单元 ×2 组的采样 & 保持功能 （1 组可同时采样 3 个通道）
I/O 端口	• 57 个通用输入 / 输出引脚 • 输入 / 输出复用端口可按位转换输入 / 输出
封装	• LQFP1414 -100 （0.5 节距）
电源电压	• V_{CC}：3.0 ～ 3.6V 或 4.5 ～ 5.5V、 PV_{CC}：4.5 ～ 5.5V、 AV_{CC}：4.5 ～ 5.5V

1.2 框图

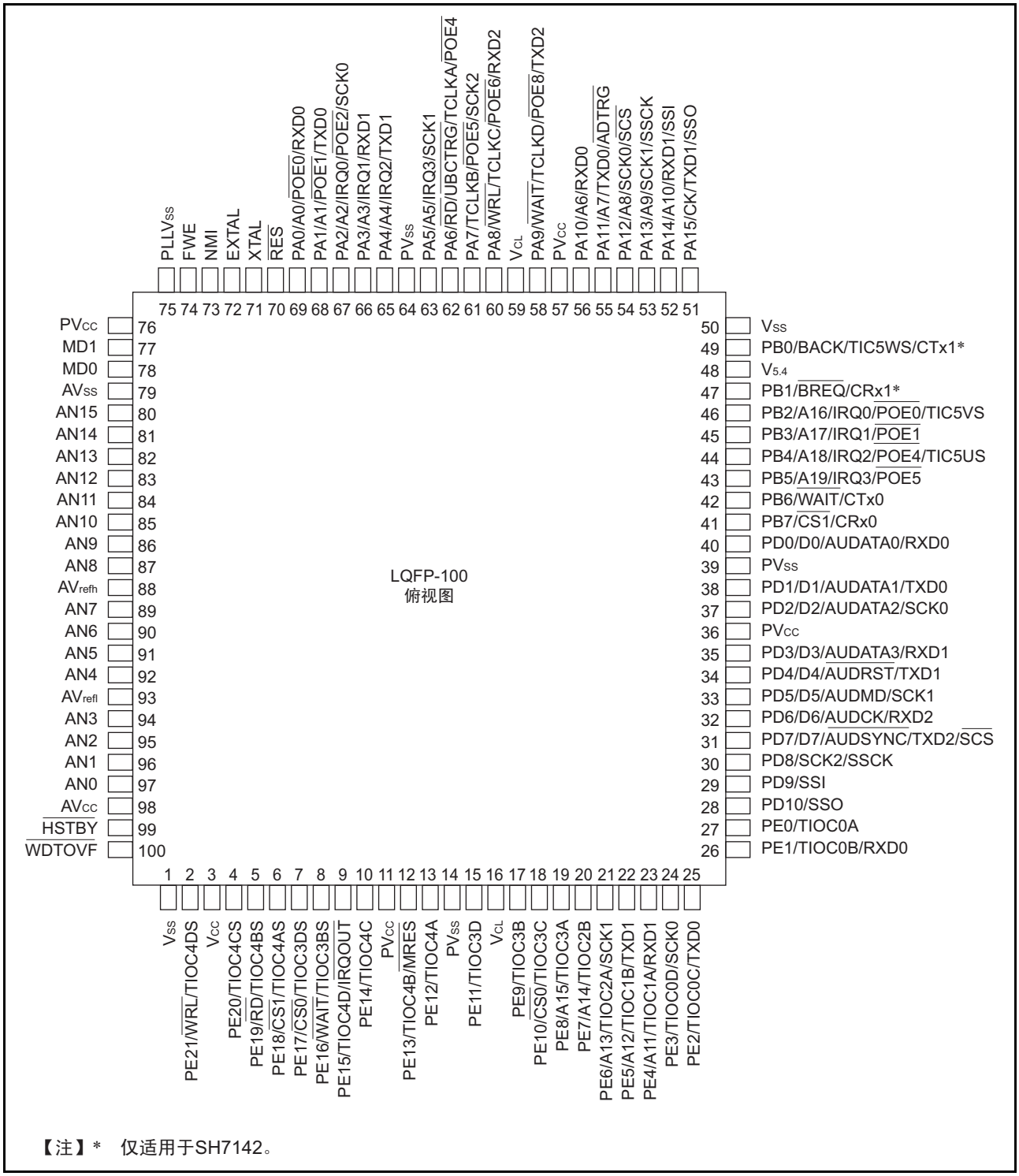


【符号说明】

ROM： 内部ROM
RAM： 内部RAM
UBC： 用户断点控制器
AUD： 高级用户调试器
INTC： 中断控制器
CPG： 时钟振荡器
WDT： 看门狗定时器
CPU： 中央处理器
BSC： 总线状态控制器
DTC： 数据传送控制器
PFC： 引脚功能控制器
MTU2： 多功能定时器脉冲单元2
MTU2S： 多功能定时器脉冲单元2（子集版）
POE： 端口输出允许
SCI： 串行通信接口
SSU： 同步串行通信单元
CMT： 比较匹配定时器
ADC： A/D转换器
RCAN-ET： 控制器局域网

图 1.1 框图

1.3 引脚排列图



【注】* 仅适用于SH7142。

图 1.2 引脚排列图

1.4 引脚功能

各引脚功能如表 1.2 所示。

表 1.2 引脚功能

分类	引脚名称	输入 / 输出	名称	功 能
电源	V_{CC}	输入	电源	电源引脚。必须将所有 V_{CC} 引脚连接至系统电源。如果有开路引脚，则不运行。 接通电源时，消耗电流增加。
	V_{SS}	输入	接地	接地引脚。必须将所有 V_{SS} 引脚连接至系统电源 （0V）。如果有开路引脚，则不运行。
	PV_{CC}	输入	电源	输入 / 输出引脚的电源引脚。必须将所有 PV_{CC} 引脚连接至系统电源。如果有开路引脚，则不运行。接通电源时，消耗电流增加。
	PV_{SS}	输入	接地	输入 / 输出引脚的接地引脚。必须将所有 PV_{SS} 引脚连接至系统电源 （0V）。如果有开路引脚，则不运行。
	V_{CL}	输出	内部降压电源	内部降压电源的外部电容引脚。必须将所有 V_{CL} 引脚通过 0.47μF 的电容器连接至 V_{SS} （配置于引脚附近）。
时钟	$PLLV_{SS}$	输入	PLL 的接地	内部 PLL 振荡器的接地引脚。
	EXTAL	输入	外部时钟	连接晶体谐振器。另外， EXTAL 引脚也可输入外部时钟。
	XTAL	输出	晶体	连接晶体谐振器。
	CK	输出	系统时钟	对外部器件提供系统时钟。
运行模式控制	MD1、 MD0	输入	模式设定	设定运行模式。不得在运行期间改变这些引脚。
	FWE	输入	闪存编程允许	闪存的引脚。可保护闪存的编程 / 擦除。
系统控制	$\overline{RES}$	输入	上电复位	如果此引脚为低电平，则为上电复位状态。
	$\overline{MRES}$	输入	手动复位	如果此引脚为低电平，则为手动复位状态。
	$\overline{HSTBY}$	输入	硬件待机	如果此引脚为低电平，则转移至硬件待机模式。无任何输入时，在内部上拉。 必须在 $\overline{HSTBY}$ 引脚的高电平状态，接通电源。
	$\overline{WDTOVF}$	输出	看门狗定时器上溢	WDT 的上溢输出信号。 解除深度待机模式后，在高阻抗状态期间产生上溢，因此必须上拉该引脚。
	$\overline{BREQ}$	输入	总线权请求	如果外部器件请求释放总线权，则设置为低电平。
	$\overline{BACK}$	输出	总线权请求应答	表示已向外部器件释放总线权。输出 $\overline{BREQ}$ 信号的器件接收 $\overline{BACK}$ 信号，即可得知已获取总线权。
中断	NMI	输入	非屏蔽中断	非屏蔽中断请求引脚。不使用时，必须固定为高电平或低电平。
	IRQ3 ～ IRQ0	输入	中断请求 3 ～ 0	可屏蔽中断请求引脚。 可选择电平输入或边沿输入。边沿输入时，可选择上升沿、下降沿或双边沿。
	$\overline{IRQOUT}$	输出	中断请求输出	表示已产生中断源。总线释放期间也可得知产生中断。

分类	引脚名称	输入 / 输出	名称	功 能
地址总线	A19 ～ A0	输出	地址总线	输出地址。
数据总线	D7 ～ D0	输入 / 输出	数据总线	双向总线。
总线控制	$\overline{CS1}$、$\overline{CS0}$	输出	片选 1、 0	外部存储器或器件的片选信号。
	$\overline{RD}$	输出	读取	表示从外部器件读取。
	$\overline{WRL}$	输出	低位写入	表示向外部数据的 bit7 ～ 0 写入。
	$\overline{WAIT}$	输入	等待	向存取外部空间时的总线周期插入等待周期的输入引脚。
多功能定时器脉冲单元 2（MTU2）	TCLKA、TCLKB、TCLKC、TCKLD	输入	MTU2 定时器时钟输入	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）	TGRA_0 ～ TGRD_0 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚。
	TIOC1A、TIOC1B	输入 / 输出	MTU2 输入捕捉 / 输出比较 （通道 1）	TGRA_1、 TGRB_1 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚。
	TIOC2A、TIOC2B	输入 / 输出	MTU2 输入捕捉 / 输出比较 （通道 2）	TGRA_2、 TGRB_2 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚。
	TIOC3A、TIOC3B、TIOC3C、TIOC3D	输入 / 输出	MTU2 输入捕捉 / 输出比较 （通道 3）	TGRA_3 ～ TGRD_3 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚。
	TIOC4A、TIOC4B、TIOC4C、TIOC4D	输入 / 输出	MTU2 输入捕捉 / 输出比较 （通道 4）	TGRA_4 ～ TGRD_4 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚。
多功能定时器脉冲单元 2S（MTU2S）	TIOC3BS、TIOC3DS	输入 / 输出	MTU2S 输入捕捉 / 输出比较 （通道 3）	TGRB_3S、 TGRD_3S 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚。
	TIOC4AS、TIOC4BS、TIOC4CS、TIOC4DS	输入 / 输出	MTU2S 输入捕捉 / 输出比较 （通道 4）	TGRA_4S ～ TGRD_4S 的输入捕捉输入 / 输出比较输出 / PWM 输出引脚。
	TIC5US、TIC5VS、TIC5WS	输入	MTU2S 输入捕捉 （通道 5）	TGRU_5S、 TGRV_5S、 TGRW_5S 的输入捕捉输入引脚。

分类	引脚名称	输入 / 输出	名称	功 能
端口输出允许（POE）	$\overline{\text{POE8}}$、$\overline{\text{POE6}}$ ～ $\overline{\text{POE4}}$、$\overline{\text{POE2}}$ ～ $\overline{\text{POE0}}$	输入	控制端口输出	将 MTU2/MTU2S 波形输出引脚置为高阻抗状态的请求信号输入引脚。
串行通信接口（SCI）	TXD2 ～ TXD0	输出	发送数据	发送数据的引脚。
	RXD2 ～ RXD0	输入	接收数据	接收数据的引脚。
	SCK2 ～ SCK0	输入 / 输出	串行时钟	时钟输入 / 输出引脚。
同步串行通信单元（SSU）	SSO	输入 / 输出	数据	数据输入 / 输出引脚。
	SSI	输入 / 输出	数据	数据输入 / 输出引脚。
	SSCK	输入 / 输出	时钟	时钟输入 / 输出引脚。
	$\overline{\text{SCS}}$	输入 / 输出	片选	片选输入 / 输出引脚。
控制器局域网（RCAN-ET）	CTx1*、 CTx0	输出	发送数据	CAN 总线发送引脚。
	CRx1*、 CRx0	输入	接收数据	CAN 总线接收引脚。
A/D 转换器（ADC）	AN15 ～ AN0	输入	模拟输入引脚	模拟输入引脚。
	$\overline{\text{ADTRG}}$	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。
	AV_{CC}	输入	模拟电源	A/D 转换器的电源引脚。不使用 A/D 转换器时，必须连接至系统电源 （PV_{CC}）。 必须将所有 AV_{CC} 引脚连接至系统电源 （PV_{CC}）。如果有开路引脚，则不运行。
	AV_{SS}	输入	模拟接地	A/D 转换器的接地引脚。必须连接至系统电源 （0V）。 必须将所有 AV_{SS} 引脚连接至系统电源 （0V）。如果有开路引脚，则不运行。
	AV_{refh}	输入	模拟基准电源（高电平）	模拟基准电源 （高电平）。
	AV_{refl}	输入	模拟基准电源（低电平）	模拟基准电源 （低电平）。
I/O 端口	PA15 ～ PA0	输入 / 输出	通用端口	通用输入 / 输出端口引脚。
	PB7 ～ PB0	输入 / 输出	通用端口	通用输入 / 输出端口引脚。
	PD10 ～ PD0	输入 / 输出	通用端口	通用输入 / 输出端口引脚。
	PE21 ～ PE0	输入 / 输出	通用端口	通用输入 / 输出端口引脚。
用户断点控制器（UBC）	$\overline{\text{UBCTRG}}$	输出	用户断点触发输出	UBC 条件匹配的触发输出引脚。

分类	引脚名称	输入 / 输出	名称	功　能
高级用户调试器（AUD）	AUDATA3 ～ AUDATA0	输入 / 输出	AUD 数据	转移跟踪模式： 转移目标 / 转移源地址输出引脚。 RAM 监控模式： 监控地址输入 / 数据输入或输出引脚。
	$\overline{\text{AUDRST}}$	输入	AUD 复位	复位信号输入引脚。
	AUDMD	输入	AUD 模式	模式选择信号输入引脚。 转移跟踪模式 （L） RAM 监控模式 （H）
	AUDCK	输入 / 输出	AUD 时钟	转移跟踪模式： 同步时钟输出引脚。 RAM 监控模式： 同步时钟输入引脚。
	$\overline{\text{AUDSYNC}}$	输入 / 输出	AUD 同步信号	转移跟踪模式： 数据起始位置识别信号输出引脚。 RAM 监控模式： 数据起始位置识别信号输入引脚。

【使用时的注意事项】

解除深度待机模式后，在高阻抗状态期间产生上溢，因此必须上拉 $\overline{\text{WDTOVF}}$ 引脚。

【注】 * 仅适用于 SH7142。

第 2 章 CPU

2.1 特点

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

2.2 寄存器结构

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

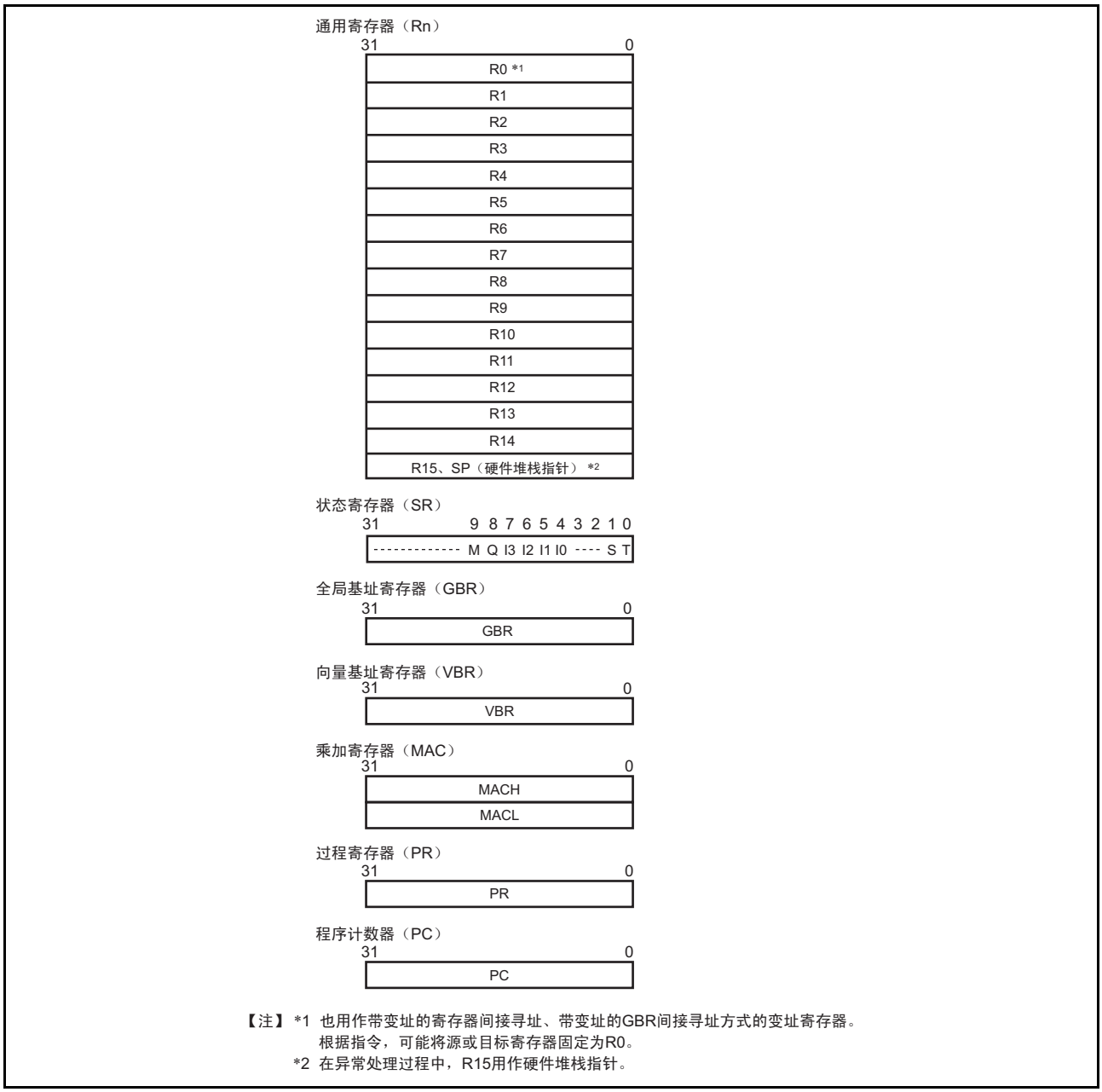


图 2.1 CPU 内部寄存器结构

2.2.1 通用寄存器 （Rn）

通用寄存器 （Rn）长度为 32 位，从 R0 到 R15 共有 16 个，用于数据处理与地址运算。 R0 也可用作变址寄存器。几个指令可使用的寄存器固定为 R0。 R15 可用作硬件堆栈指针 （SP）。通过使用 R15 参考堆栈，执行异常处理时状态寄存器 （SR）与程序计数器 （PC）的压栈和出栈。

2.2.2 控制寄存器

控制寄存器长度为 32 位，有状态寄存器 （SR）、全局基址寄存器 （GBR）及向量基址寄存器 （VBR）等 3 个。 SR 表示处理状态； GBR 用作 GBR 间接寻址方式的基址，用于内部外围模块寄存器的数据传送等；VBR 用作含中断的异常处理向量区域的基址。

(1) 状态寄存器 （SR）

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	M	Q	I[3:0]				-	-	S	T
初始值：	0	0	0	0	0	0	不定	不定	1	1	1	1	0	0	不定	不定
R/W：	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

位	位名称	初始值	R/W	说 明
31 ～ 10	—	均为 0	R	保留位 读取值、写入值总是为 0。
9	M	不定	R/W	DIV0U、 DIV0S、 DIV1 指令使用。
8	Q	不定	R/W	DIV0U、 DIV0S、 DIV1 指令使用。
7 ～ 4	I[3:0]	1111	R/W	中断屏蔽位
3、 2	—	均为 0	R	保留位 读取值、写入值总是为 0。
1	S	不定	R/W	S 位 乘加指令使用。
0	T	不定	R/W	T 位 下列指令时，表示真 (1)、假 (0)。 MOVT、 CMP/cond、 TAS、 TST、 BT (BT/S)、 BF(BF/S)、 SETT、 CLRT 下列指令时，表示进位、借位、上溢、下溢等。 ADDV、ADDC、SUBV、SUBC、NEGC、DIV0U、DIV0S、DIV1、SHAR、SHAL、SHLR、SHLL、ROTR、ROTL、ROTCR、ROTCL

(2) 全局基址寄存器 （GBR）

表示 GBR 间接寻址方式的基址。 GBR 间接寻址方式用于内部外围模块寄存器区域等的数据传送与逻辑运算。

(3) 向量基址寄存器 （VBR）

表示异常处理向量区域的基址。

2.2.3 系统寄存器

系统寄存器长度为 32 位，有乘加寄存器 （MACH、 MACL）、过程寄存器 （PR）及程序计数器 （PC）等 4 个。

(1) 乘加寄存器 （MACH、 MACL）

保存乘法、乘法累加运算的结果。

(2) 过程寄存器 （PR）

保存从子程序过程返回的地址。

(3) 程序计数器 （PC）

PC 表示当前执行指令的 4 个字节 （2 条指令）后的地址。

2.2.4 寄存器的初始值

复位的寄存器值如表 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）	向量地址表的 SP 值
控制寄存器	SR	I3 ～ I0 为 1111 （H'F），保留位为 0，其他不定
	GBR	不定
	VBR	H'00000000
系统寄存器	MACH、 MACL、 PR	不定
	PC	向量地址表的 PC 值

2.3 数据格式

2.3.1 寄存器的数据格式

寄存器操作数的数据长度总是为长字 （32 位）。存储器的数据加载至寄存器时，如果存储器操作数的数据长度为字节 （8 位）或字 （16 位），则符号扩展为长字，保存至寄存器。

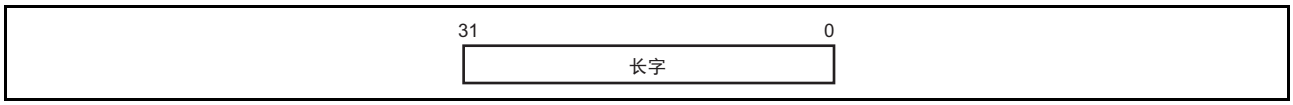


图 2.2 寄存器的数据格式

2.3.2 存储器的数据格式

存储器有字节、字、长字等 3 种数据格式。

将字节数据配置在任意地址，字数据必须从地址 2n 配置，长字数据必须从地址 4n 配置。如果不按此规则存取，则产生地址错误。此时，无法保证存取结果。注意：硬件堆栈指针 （SP、 R15）指向的堆栈以长字保存程序计数器 （PC）与状态寄存器 （SR），因此，硬件堆栈指针的值必须设定为 4n。

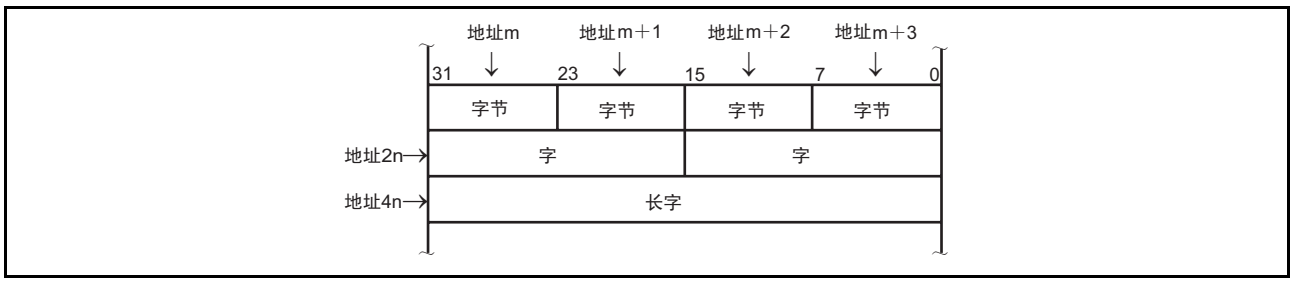


图 2.3 存储器的数据格式

2.3.3 立即数的数据格式

在操作码配置字节 （8 位）长度的立即数。

为 MOV、 ADD、 CMP/EQ 指令时，将立即数符号扩展，以长字运算；为 TST、 AND、 OR、 XOR 指令时，将立即数零扩展，以长字运算。因此，如果 AND 指令时使用立即数，则总是清除目标寄存器的高 24 位。

字与长字的立即数不配置在操作码，而配置在存储器表。使用带位移量的 PC 相对寻址方式的立即数数据传送指令 （MOV），参照存储器表。

2.4 指令特点

2.4.1 RISC 结构

指令为 RISC 结构，其特点如下：

(1) 16 位固定长度指令

指令长度均为 16 位固定长度，以提高程序的编码效率。

(2) 1 条指令 /1 个状态

采用流水线方式，能够以 1 条指令 /1 个状态执行基本指令。

(3) 数据长度

运算的基本数据长度为长字。存储器的存取长度可选择字节 / 字 / 长字。存储器的字节数据与字数据符号扩展后，以长字运算。立即数在算术运算时，符号扩展后以长字运算；而在逻辑运算时，零扩展后以长字运算。

表 2.2 符号扩展字数据

本 LSI 的 CPU	说 明	其他 CPU 的例子
MOV.W @(disp, PC), R1 ADD R1, R0 ..... .DATA.W H'1234	符号扩展为 32 位， R1 为 H'00001234。 由 ADD 指令执行运算。	ADD.W #H'1234, R0

【注】 通过 @(disp, PC) 参照立即数。

(4) 加载存储结构

在寄存器间执行基本运算。执行与存储器的运算时，需先将数据加载至寄存器 （加载存储结构）。但，AND 等位操作指令可直接对存储器执行。

(5) 延迟转移

无条件转移指令为延迟转移指令。延迟转移指令时，执行延迟转移指令的下一条指令后，再执行转移。由此，可减轻转移时的流水线混乱。带条件的转移指令，有延迟转移指令与通常转移指令两种。

表 2.3 延迟转移指令

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

(6) 乘法 / 乘法累加运算

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 的乘法累加运算，以 2 ～ 4 个状态执行 32×32→64 的乘法及 32×32+64→64 的乘法累加运算。

(7) T 位

比较结果反映在 SR 的 T 位，根据其真假执行条件转移。仅根据最少所需指令改变 T 位，以提高处理速度。

表 2.4 T 位

本 LSI 的 CPU	说 明	其他 CPU 的例子
CMP/GE R1, R0 BT TRGET0 BF TRGET1	R0 ≥ R1 时， T 位置位。 R0 ≥ R1 时，转移至 TRGET0。 R0 ＜ R1 时，转移至 TRGET1。	CMP.W R1, R0 BGE TRGET0 BLT TRGET1
ADD #–1, R0 CMP/EQ #0, R0 BT TRGET	ADD 时， T 位无变化。 R0=0 时， T 位置位。 R0=0 时，执行转移。	SUB.W #1, R0 BEQ TRGET

(8) 立即数

在操作码配置字节 （8 位）长度的立即数。字与长字的立即数不配置在操作码，而配置在存储器表。使用带位移量的 PC 相对寻址方式的立即数传送指令 （MOV），参照存储器表。

表 2.5 通过立即数参考

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

【注】 通过@ (disp, PC) 参考立即数。

(9) 绝对地址

通过绝对地址参考数据时，先将绝对地址的值配置在存储器表。执行指令时，通过加载立即数的方法将该值传送至寄存器，并通过寄存器间接寻址方式参考数据。

表 2.6 通过绝对地址参考

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

【注】 通过@ (disp, PC) 参考立即数。

(10)　16 位 /32 位位移

通过 16 位或 32 位位移参考数据时，预先将位移量的值配置在存储器表。执行指令时，通过加载立即数的方法将该值传送至寄存器，并通过带变址的寄存器间接寻址方式参考数据。

表 2.7　通过位移参考

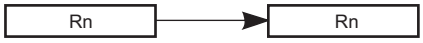
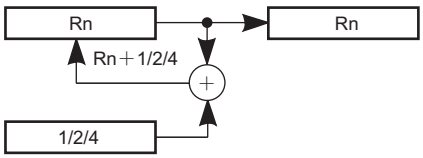
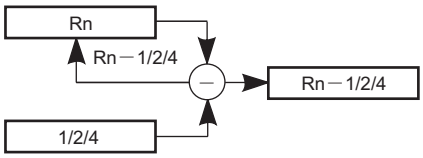
分　类	本 LSI 的 CPU	其他 CPU 的例子
16 位位移	MOV.W　@(disp,PC), R0 MOV.W　@(R0,R1), R2 ..... .DATA.W　H'1234	MOV.W　@(H'1234, R1), R2

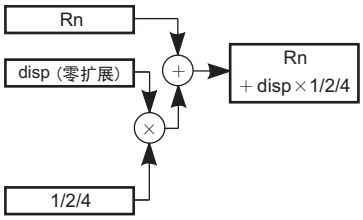
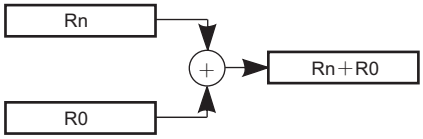
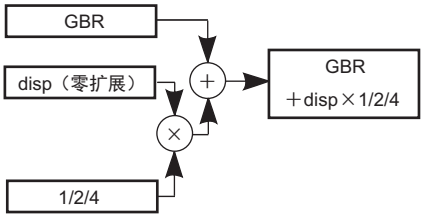
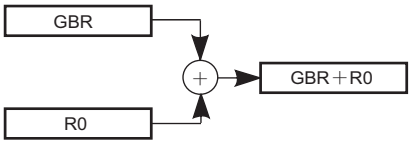
【注】　通过@ (disp，PC) 参考立即数。

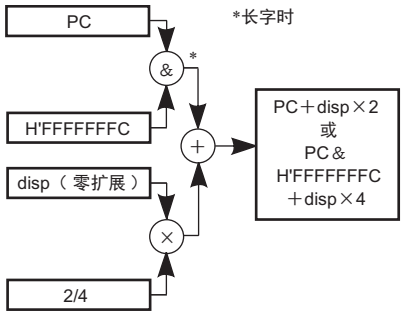
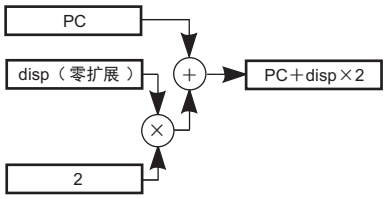
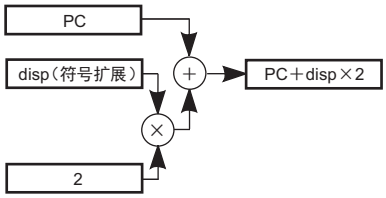
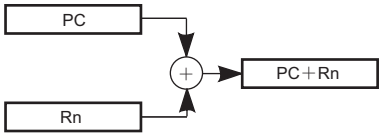
2.4.2　寻址方式

寻址方式与有效地址的计算方法如表 2.8 所示。

表 2.8　寻址方式与有效地址

寻址方式	指令格式	有效地址的计算方法	计算式
寄存器直接寻址	Rn	有效地址为寄存器 Rn。 （操作数为寄存器 Rn 的内容）	—
寄存器间接寻址	@Rn	有效地址为寄存器 Rn 的内容。 	Rn
后增寄存器间接寻址	@Rn+	有效地址为寄存器 Rn 的内容。执行指令后，对 Rn 加常数。操作数长度为字节时，常数是 1；为字时，长数是 2；为长字时，常数是 4。 	Rn 指令执行后 字节：Rn+1→Rn 字：　Rn+2→Rn 长字：　Rn+4→Rn
先减寄存器间接寻址	@ – Rn	有效地址为先减常数后寄存器 Rn 的内容。操作数长度为字节时，常数是 1；为字时常数是 2；为长字时，常数是 4。 	字节：Rn–1→Rn 字：　Rn–2→Rn 长字：Rn–4→Rn （用已计算的 Rn 执行指令）

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

寻址方式	指令格式	有效地址的计算方法	计算式
带位移量的 PC 相对寻址	@ (disp: 8, PC)	有效地址为寄存器 PC 加 8 位位移量 disp 的内容。 disp 零扩展后，操作数长度为字时是 2 倍，为长字时是 4 倍。且为长字时，屏蔽 PC 的低 2 位。  	字：PC+disp×2 长字： PC & H'FFFFFFFC +disp×4
PC 相对寻址	disp: 8	有效地址为 8 位位移量 disp 符号扩展后乘 2，再加寄存器 PC 的内容。  	PC+disp×2
	disp: 12	有效地址为 12 位位移量 disp 符号扩展后乘 2，再加寄存器 PC 的内容。  	PC+disp×2
	Rn	有效地址为寄存器 PC 加 Rn 的内容。  	PC+Rn
立即寻址	#imm:8	TST, AND, OR, XOR 指令的 8 位立即数 imm 零扩展。	—
	#imm:8	MOV, ADD, CMP/EQ 指令的 8 位立即数 imm 符号扩展。	—
	#imm:8	TRAPA 指令的 8 位立即数 imm 零扩展后乘 4。	—

2.4.3 指令格式

说明指令格式、源操作数及目标操作数的含义。操作数的含义取决于操作码，符号说明如下：

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	15 0 xxxx xxxx xxxx xxxx	—	—	NOP
n 格式	15 0 xxxx nnnn xxxx xxxx	—	nnnn：寄存器直接寻址	MOVT Rn
		控制寄存器或系统寄存器	nnnn：寄存器直接寻址	STS MACH, Rn
		控制寄存器或系统寄存器	nnnn：先减寄存器间接寻址	STC.L SR, @-Rn
m 格式	15 0 xxxx mmmm xxxx xxxx	mmmm：寄存器直接寻址	控制寄存器或系统寄存器	LDC Rm, SR
		mmmm：后增寄存器间接寻址	控制寄存器或系统寄存器	LDC.L @Rm+, SR
		mmmm：寄存器间接寻址	—	JMP @Rm
		mmmm：使用 Rm 的 PC 相对寻址	—	BRAF Rm
nm 格式	15 0 xxxx nnnn mmmm xxxx	mmmm：寄存器直接寻址	nnnn：寄存器直接寻址	ADD Rm, Rn
		mmmm：寄存器直接寻址	nnnn：寄存器间接寻址	MOV.L Rm, @Rn
		mmmm： 后增寄存器间接寻址 （乘法累加运算） nnnn：* 后增寄存器间接寻址 （乘法累加运算）	MACH, MACL	MAC.W @Rm+, @Rn+
		mmmm：后增寄存器间接寻址	nnnn：寄存器直接寻址	MOV.L @Rm+, Rn
		mmmm：寄存器直接寻址	nnnn：先减寄存器间接寻址	MOV.L Rm, @-Rn
		mmmm：寄存器直接寻址	nnnn： 带变址的寄存器间接寻址	MOV.L Rm,@(R0, Rn)
md 格式	15 0 xxxx xxxx mmmm dddd	mmmmdddd: 带位移量的寄存器间接寻址	R0 （寄存器直接寻址）	MOV.B @(disp,Rm),R0
nd4 格式	15 0 xxxx xxxx nnnn dddd	R0 （寄存器直接寻址）	nnnndddd： 带位移量的寄存器间接寻址	MOV.B R0,@(disp,Rn)

指令格式		源操作数	目标操作数	指令例子
nmd 格式	15 0 xxxx nnnn mmmm dddd	mmmm：寄存器直接寻址	nnnndddd： 带位移量的寄存器间接寻址	MOV.L Rm,@(disp,Rn)
		mmmmdddd： 带位移量的寄存器间接寻址	nnnn：寄存器直接寻址	MOV.L @(disp,Rm),Rn
d 格式	15 0 xxxx xxxx dddd dddd	dddddddd： 带位移量的 GBR 间接寻址	R0 （寄存器直接寻址）	MOV.L @(disp,GBR),R0
		R0 （寄存器直接寻址）	dddddddd： 带位移量的 GBR 间接寻址	MOV.L R0,@(disp,GBR)
		dddddddd： 带位移量的 PC 相对寻址	R0 （寄存器直接寻址）	MOVA @(disp,PC),R0
		—	dddddddd：PC 相对寻址	BF label
d12 格式	15 0 xxxx dddd dddd dddd	—	ddddddddddd： PC 相对寻址	BRA label (label=disp+pc)
nd8 格式	15 0 xxxx nnnn dddd dddd	dddddddd： 带位移量的 PC 相对寻址	nnnn：寄存器直接寻址	MOV.L @(disp,PC),Rn
i 格式	15 0 xxxx xxxx iiii iiii	iiiiiiii： 立即寻址	带变址的 GBR 间接寻址	AND.B #imm,@(R0,GBR)
		iiiiiiii： 立即寻址	R0 （寄存器直接寻址）	AND #imm, R0
		iiiiiiii： 立即寻址	—	TRAPA #imm
ni 格式	15 0 xxxx nnnn iiii iiii	iiiiiiii： 立即寻址	nnnn：寄存器直接寻址	ADD #imm, Rn

【注】 * 乘加指令时 nnnn 为源寄存器。

2.5 指令系统

2.5.1 指令系统分类

指令分类如表 2.10 所示。

表 2.10 指令分类

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

分类	指令的种类	操作码	功　能	指令数
逻辑运算指令	6	AND	逻辑 “与” 运算	14
		NOT	位取反	
		OR	逻辑 “或” 运算	
		TAS	存储器测试与置位	
		TST	逻辑 “与” 运算的 T 位置位	
		XOR	逻辑 “异或” 运算	
移位指令	10	ROTL	左循环 1 位	14
		ROTR	右循环 1 位	
		ROTCL	带 T 位向左循环 1 位	
		ROTCR	带 T 位向右循环 1 位	
		SHAL	算术左移 1 位	
		SHAR	算术右移 1 位	
		SHLL	逻辑左移 1 位	
		SHLLn	逻辑左移 n 位	
		SHLR	逻辑右移 1 位	
		SHLRn	逻辑右移 n 位	
转移指令	9	BF	条件转移、带延迟的条件转移 （T=0 时转移）	11
		BT	条件转移、带延迟的条件转移 （T=1 时转移）	
		BRA	无条件转移	
		BRAF	无条件转移	
		BSR	转移至子程序过程	
		BSRF	转移至子程序过程	
		JMP	无条件转移	
		JSR	转移至子程序过程	
		RTS	从子程序过程返回	
系统控制指令	11	CLRT	清除 T 位	31
		CLRMAC	清除 MAC 寄存器	
		LDC	加载到控制寄存器	
		LDS	加载到系统寄存器	
		NOP	空操作	
		RTE	从异常处理返回	
		SETT	T 位置位	
		SLEEP	转移至低功耗状态	
		STC	从控制寄存器存储	
		STS	从系统寄存器存储	
		TRAPA	陷阱异常处理	
	共计 62			142

指令的操作码、操作及执行状态，按下列格式分类说明。

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 OP.Sz SRC, DEST OP：操作码 Sz：长度 SRC：源 DEST：目标 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 i i i i：立即数 dddd：位移量	表示操作概况。 符号说明 →、←：传送方向 (xx)：存储器操作数 M/Q/T：SR 内的标志位 &：位“与” ｜：位“或” ∧：位“异或” ～：位“非” <<n：左移 n 位 >>n：右移 n 位	无等待时的值。 *1	表示执行指令后 T 位的值。 符号说明 —：不变化

【注】 *1 指令的执行状态

表所示的执行状态为最小值。实际根据以下条件，指令执行的状态数会增加。

（1）取指令与数据存取产生竞争

（2）加载指令 （存储器→寄存器）的目标寄存器与紧接着的指令所用的寄存器相同

*2 根据指令的操作数长度等被放大 （×1、 ×2、 ×4）。

详情参阅 “SH-1/SH-2/SH-DSP 软件手册”。

2.5.2 数据传送指令

表 2.11 数据传送指令

指令		操作码	操 作	执行状态	T 位
MOV	#imm, Rn	1110nnnniiiiiiii	#imm → 符号扩展 →Rn	1	—
MOV.W	@(disp, PC), Rn	1001nnnnddddddd	(disp×2+PC) → 符号扩展 → Rn	1	—
MOV.L	@(disp, PC), Rn	1101nnnnddddddd	(disp×4+PC) → Rn	1	—
MOV	Rm, Rn	0110nnnnmmmm0011	Rm→ Rn	1	—
MOV.B	Rm, @Rn	0010nnnnmmmm0000	Rm→(Rn)	1	—
MOV.W	Rm, @Rn	0010nnnnmmmm0001	Rm→(Rn)	1	—
MOV.L	Rm, @Rn	0010nnnnmmmm0010	Rm→(Rn)	1	—
MOV.B	@Rm, Rn	0110nnnnmmmm0000	(Rm)→ 符号扩展 →Rn	1	—
MOV.W	@Rm, Rn	0110nnnnmmmm0001	(Rm) → 符号扩展 →Rn	1	—
MOV.L	@Rm, Rn	0110nnnnmmmm0010	(Rm)→Rn	1	—
MOV.B	Rm, @- Rn	0010nnnnmmmm0100	Rn–1→Rn, Rm →(Rn)	1	—
MOV.W	Rm, @- Rn	0010nnnnmmmm0101	Rn–2→Rn, Rm →(Rn)	1	—
MOV.L	Rm, @- Rn	0010nnnnmmmm0110	Rn–4 →Rn, Rm →(Rn)	1	—
MOV.B	@Rm+, Rn	0110nnnnmmmm0100	(Rm) → 符号扩展 →Rn,Rm+1 → Rm	1	—
MOV.W	@Rm+, Rn	0110nnnnmmmm0101	(Rm)→ 符号扩展 →Rn, Rm+2 →Rm	1	—
MOV.L	@Rm+, Rn	0110nnnnmmmm0110	(Rm)→Rn, Rm+4→Rm	1	—
MOV.B	R0, @(disp, Rn)	10000000nnnnddddd	R0 →(disp+Rn)	1	—
MOV.W	R0, @(disp, Rn)	10000001nnnnddddd	R0 →(disp×2+Rn)	1	—
MOV.L	Rm, @(disp, Rn)	0001nnnnmmmmddddd	Rm→(disp×4+Rn)	1	—
MOV.B	@(disp, Rm), R0	10000100mmmmddddd	(disp+Rm)→ 符号扩展 →R0	1	—
MOV.W	@(disp, Rm), R0	10000101mmmmddddd	(disp×2+Rm) → 符号扩展 →R0	1	—
MOV.L	@(disp, Rm), Rn	0101nnnnmmmmddddd	(disp×4+Rm)→Rn	1	—
MOV.B	Rm, @(R0, Rn)	0000nnnnmmmm0100	Rm →(R0+Rn)	1	—
MOV.W	Rm, @(R0, Rn)	0000nnnnmmmm0101	Rm →(R0+Rn)	1	—
MOV.L	Rm, @(R0, Rn)	0000nnnnmmmm0110	Rm →(R0+Rn)	1	—
MOV.B	@(R0, Rm), Rn	0000nnnnmmmm1100	(R0+Rm)→ 符号扩展 →Rn	1	—
MOV.W	@(R0, Rm), Rn	0000nnnnmmmm1101	(R0+Rm) → 符号扩展 →Rn	1	—
MOV.L	@(R0, Rm), Rn	0000nnnnmmmm1110	(R0+Rm) →Rn	1	—
MOV.B	R0, @(disp, GBR)	11000000ddddddddd	R0 →(disp+GBR)	1	—
MOV.W	R0, @(disp, GBR)	11000001ddddddddd	R0→(disp×2+GBR)	1	—
MOV.L	R0, @(disp, GBR)	11000010ddddddddd	R0→ (disp×4+GBR)	1	—
MOV.B	@(disp, GBR), R0	11000100ddddddddd	(disp+GBR)→ 符号扩展 → R0	1	—
MOV.W	@(disp, GBR), R0	11000101ddddddddd	(disp×2+GBR) → 符号扩展 →R0	1	—
MOV.L	@(disp, GBR), R0	11000110ddddddddd	(disp×4+GBR)→R0	1	—
MOVA	@(disp, PC), R0	11000111ddddddddd	disp×4+PC →R0	1	—
MOVT	Rn	0000nnnn00101001	T →Rn	1	—
SWAP.B	Rm, Rn	0110nnnnmmmm1000	Rm→ 交换低位 2 个字节的高低字节 →Rn	1	—
SWAP.W	Rm, Rn	0110nnnnmmmm1001	Rm→ 交换高低字 →Rn	1	—
XTRCT	Rm, Rn	0010nnnnmmmm1101	Rm：Rn 的中间 32 位 →Rn	1	—

2.5.3 算术运算指令

表 2.12 算术运算指令

指令		操作码	操 作	执行状态	T 位
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	1	比较结果
CMP/EQ	Rm, Rn	0011nnnnmmmm0000	Rn=Rm 时 1→T	1	比较结果
CMP/HS	Rm, Rn	0011nnnnmmmm0010	无符号 Rn ≥ Rm 时 1→T	1	比较结果
CMP/GE	Rm, Rn	0011nnnnmmmm0011	有符号 Rn ≥ Rm 时 1→T	1	比较结果
CMP/HI	Rm, Rn	0011nnnnmmmm0110	无符号 Rn ＞ Rm 时 1→T	1	比较结果
CMP/GT	Rm, Rn	0011nnnnmmmm0111	有符号 Rn ＞ Rm 时 1→T	1	比较结果
CMP/PL	Rn	0100nnnn00010101	Rn ＞ 0 时 1→T	1	比较结果
CMP/PZ	Rn	0100nnnn00010001	Rn ≥ 0 时 1→T	1	比较结果
CMP/STR	Rm, Rn	0010nnnnmmmm1100	任意字节相等时 1→T	1	比较结果
DIV1	Rm, Rn	0011nnnnmmmm0100	单步除法 (Rn÷Rm)	1	计算结果
DIV0S	Rm, Rn	0010nnnnmmmm0111	Rn 的 MSB→Q， Rm 的 MSB→M、 M^Q→T	1	计算结果
DIV0U		0000000000011001	0→M/Q/T	1	0
DMULS.L	Rm, Rn	0011nnnnmmmm1101	带符号 Rn×Rm→MACH、 MACL 32×32→64 位	2 ～ 5*	—
DMULU.L	Rm, Rn	0011nnnnmmmm0101	无符号 Rn×Rm→MACH、 MACL 32×32→64 位	2 ～ 5*	—
DT	Rn	0100nnnn00010000	Rn–1→Rn，Rn 为 0 时 1→T Rn 为非 0 时 0→T	1	比较结果
EXTS.B	Rm, Rn	0110nnnnmmmm1110	Rm 从字节符号扩展 →Rn	1	—
EXTS.W	Rm, Rn	0110nnnnmmmm1111	Rm 从字符号扩展 →Rn	1	—
EXTU.B	Rm, Rn	0110nnnnmmmm1100	Rm 从字节零扩展 →Rn	1	—
EXTU.W	Rm, Rn	0110nnnnmmmm1101	Rm 从字零扩展 →Rn	1	—
MAC.L	@Rm+, @Rn+	0000nnnnmmmm1111	带符号 (Rn) × (Rm) + MAC →MAC 32×32+64→64 位	2 ～ 5*	—
MAC.W	@Rm+, @Rn+	0100nnnnmmmm1111	带符号 (Rn) × (Rm) + MAC→MAC 16×16+64→64 位	2 ～ 4*	—
MUL.L	Rm, Rn	0000nnnnmmmm0111	Rn × Rm → MACL 32×32→32 位	2 ～ 5*	—
MULS.W	Rm, Rn	0010nnnnmmmm1111	带符号 Rn×Rm →MACL 16×16→32 位	1 ～ 3*	—
MULU.W	Rm, Rn	0010nnnnmmmm1110	无符号 Rn×Rm →MACL 16×16→32 位	1 ～ 3*	—
NEG	Rm, Rn	0110nnnnmmmm1011	0 – Rm→ Rn	1	—
NEGC	Rm, Rn	0110nnnnmmmm1010	0 – m – T→Rn，借位 →T	1	借位
SUB	Rm, Rn	0011nnnnmmmm1000	Rn – Rm→Rn	1	—
SUBC	Rm, Rn	0011nnnnmmmm1010	Rn – Rm – T→Rn，借位 →T	1	借位
SUBV	Rm, Rn	0011nnnnmmmm1011	Rn – Rm→Rn，下溢 →T	1	下溢

【注】 * 表示普通执行状态。

2.5.4 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
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，1→MSB of (Rn)	4	测试结果
TST	Rm, Rn	0010nnnnmmmm1000	Rn & Rm，结果为 0 时 1→T	1	测试结果
TST	#imm, R0	11001000iiiiiiii	R0 & imm，结果为 0 时 1→T	1	测试结果
TST.B	#imm, @(R0, GBR)	11001100iiiiiiii	(R0+GBR) & imm，结果为 0 时 1→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	—

2.5.5 移位指令

表 2.14 移位指令

指令		操作码	操作	执行状态	T 位
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
SHAL	Rn	0100nnnn00100000	T←Rn← 0	1	MSB
SHAR	Rn	0100nnnn00100001	MSB→Rn→T	1	LSB
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	—

2.5.6 转移指令

表 2.15 转移指令

指令		操作码	操 作	执行状态	T 位
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	—
RTS		0000000000001011	延迟转移， PR→PC	2	—

【注】 * 不转移时，为 1 个状态。

2.5.7 系统控制指令

表 2.16 系统控制指令

指令		操作码	操 作	执行状态	T 位
CLRT		0000000000001000	0 →T	1	0
CLRMAC		0000000000101000	0 →MACH、 MACL	1	—
LDC	Rm, SR	0100mmmm00001110	Rm →SR	6	LSB
LDC	Rm, GBR	0100mmmm00011110	Rm →GBR	4	—
LDC	Rm, VBR	0100mmmm00101110	Rm →VBR	4	—
LDC.L	@Rm+, SR	0100mmmm00000111	(Rm) →SR、 Rm+4→Rm	8	LSB
LDC.L	@Rm+, GBR	0100mmmm00010111	(Rm) →GBR、 Rm+4 →Rm	4	—
LDC.L	@Rm+, VBR	0100mmmm00100111	(Rm) →VBR、 Rm+4→Rm	4	—
LDS	Rm, MACH	0100mmmm00001010	Rm→ MACH	1	—
LDS	Rm, MACL	0100mmmm00011010	Rm→MACL	1	—
LDS	Rm, PR	0100mmmm00101010	Rm→PR	1	—
LDS.L	@Rm+, MACH	0100mmmm00000110	(Rm) →MACH、 Rm+4→Rm	1	—
LDS.L	@Rm+, MACL	0100mmmm00010110	(Rm) →MACL、 Rm+4→Rm	1	—
LDS.L	@Rm+, PR	0100mmmm00100110	(Rm)→PR、 Rm+4→Rm	1	—
NOP		0000000000001001	空操作	1	—
RTE		0000000000101011	延迟转移，堆栈区 →PC/SR	5	—
SETT		0000000000011000	1→T	1	1
SLEEP		0000000000011011	睡眠	4*	—
STC	SR, Rn	0000nnnn00000010	SR→Rn	1	—
STC	GBR, Rn	0000nnnn00010010	GBR→Rn	1	—
STC	VBR, Rn	0000nnnn00100010	VBR→Rn	1	—
STC.L	SR, @ – Rn	0100nnnn00000011	Rn – 4→Rn、 SR→(Rn)	1	—
STC.L	GBR, @ – Rn	0100nnnn00010011	Rn – 4→Rn、 GBR→(Rn)	1	—
STC.L	VBR, @ – Rn	0100nnnn00100011	Rn – 4→Rn、 VBR→(Rn)	1	—
STS	MACH, Rn	0000nnnn00001010	MACH→Rn	1	—
STS	MACL, Rn	0000nnnn00011010	MACL→Rn	1	—
STS	PR, Rn	0000nnnn00101010	PR→Rn	1	—
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	8	—

【注】 * 转移至睡眠状态前的状态数。

指令的执行状态

表所示的执行状态为最小值。实际根据以下条件，指令执行的状态数会增加。

（1）取指令与数据存取产生竞争

（2）加载指令 （存储器→寄存器）的目标寄存器与紧接着的指令所用的寄存器相同。

2.6 处理状态

CPU 的处理状态有复位状态、异常处理状态、总线权释放状态、程序执行状态及低功耗状态等 5 种。各状态间的转移如图 2.4 所示。

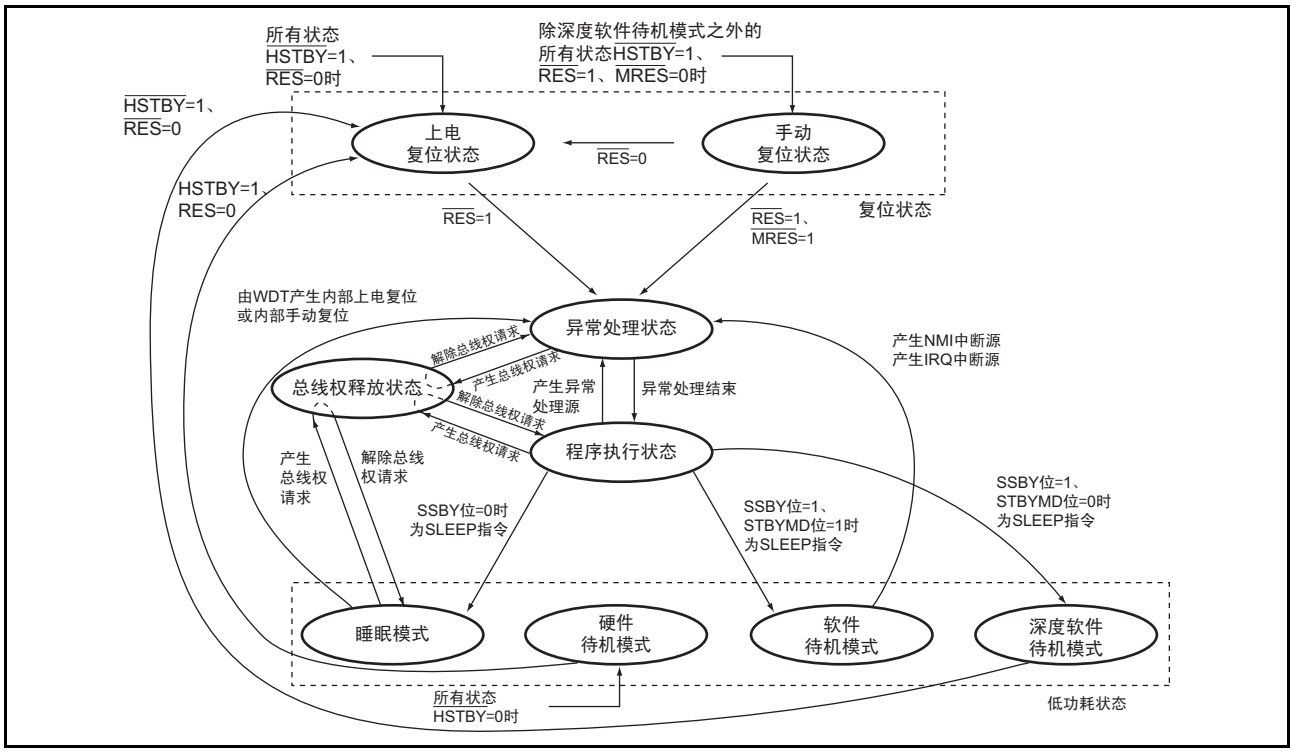


图 2.4 处理状态转移图

(1) 复位状态

CPU 复位的状态。$\overline{\text{HSTBY}}$ 引脚为高电平、$\overline{\text{RES}}$ 引脚为低电平时，为上电复位状态；$\overline{\text{HSTBY}}$ 引脚与 $\overline{\text{RES}}$ 引脚为高电平、$\overline{\text{MRES}}$ 引脚为低电平时，为手动复位状态。

(2) 异常处理状态

CPU 因复位或中断等异常处理源而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并分别保存执行起始地址（程序计数器（PC）的初始值）与堆栈指针（SP）的初始值，转移至起始地址后，开始执行程序。

产生中断等时，参照 SP，将 PC 与状态寄存器（SR）压栈。从异常处理向量表取出异常服务程序的起始地址，转移至该地址后，开始执行程序。

之后的处理状态为程序执行状态。

(3) 程序执行状态

CPU 依次执行程序的状态。

(4) 低功耗状态

CPU 停止运行、处于低功耗的状态。由睡眠指令转换为睡眠模式、软件待机模式或深度软件待机模式。另外，如果 $\overline{\text{HSTBY}}$ 引脚设置为低电平，则为硬件待机模式。

(5) 总线权释放状态

CPU 向请求总线权的器件释放总线的状态。

第 3 章 MCU 运行模式

3.1 运行模式的选择

本 LSI 有 3 种 MCU 运行模式和 3 种用于内部闪存编程的模式。
组合 FWE、 MD1 及 MD0 引脚，设定运行模式。
本 LSI 可设定的运行模式组合如表 3.1 所示，不得设定此表以外的组合。
接通系统电源时，必须执行上电复位处理。
MCU 运行模式有 MCU 扩展模式 0、 2 及单芯片模式。
用于内部闪存编程的模式有引导模式 （单板上编程模式）、用户引导模式及用户编程模式。

表 3.1 运行模式的选择

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

【注】 * 闪存的编程模式。

3.2 输入 / 输出引脚

运行模式相关的引脚结构如表 3.2 所示。

表 3.2 引脚结构

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

3.3 各运行模式的说明

3.3.1 模式 0 （MCU 扩展模式 0）

模式 0 中， CS0 空间的总线宽度为 8 位外部存储空间的宽度。

3.3.2 模式 2 （MCU 扩展模式 2）

模式 2 中，内部 ROM 有效，并可使用 CS0 空间。

3.3.3 模式 3 （单芯片模式）

单芯片模式中，可使用所有端口，但不可使用外部地址。

3.4　地址映射

各运行模式的地址映射如图 3.1 ～图 3.4 所示。

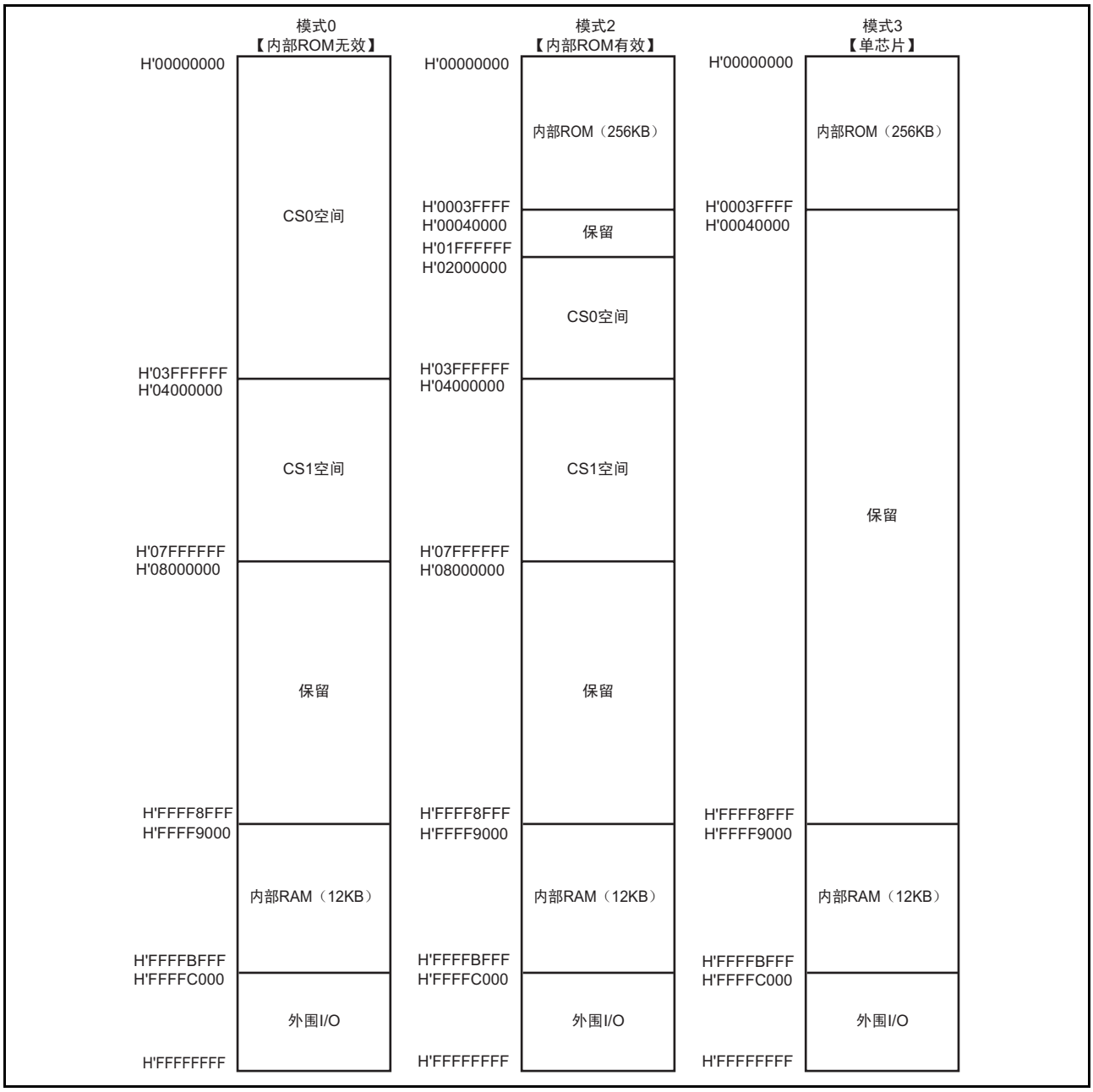


图 3.1　内部 ROM256KB/ 内部 RAM12KB 版各运行模式的地址映射

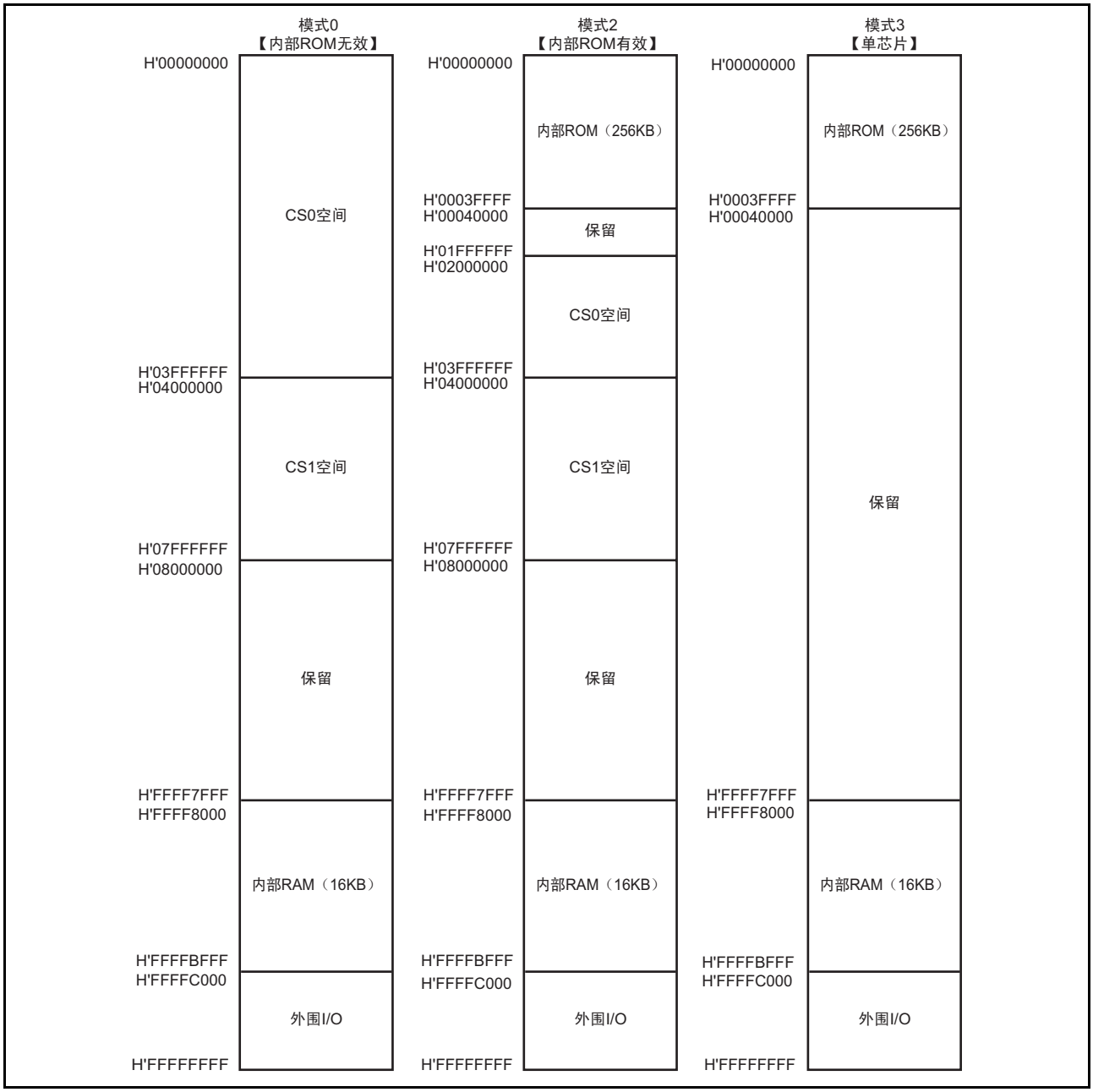


图 3.2 内部 ROM256KB/ 内部 RAM16KB 版各运行模式的地址映射

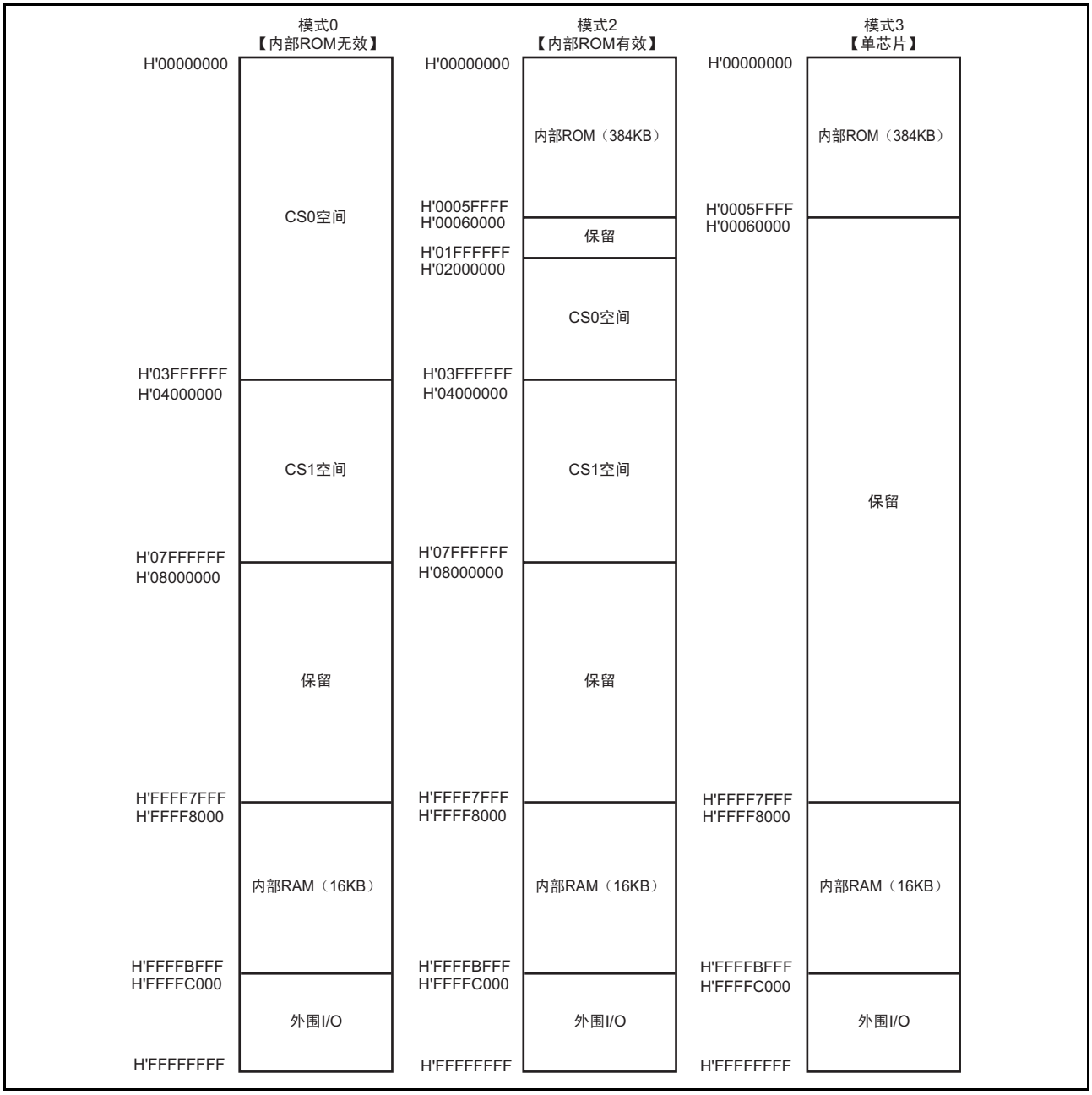


图 3.3 内部 ROM384KB/ 内部 RAM16KB 版各运行模式的地址映射

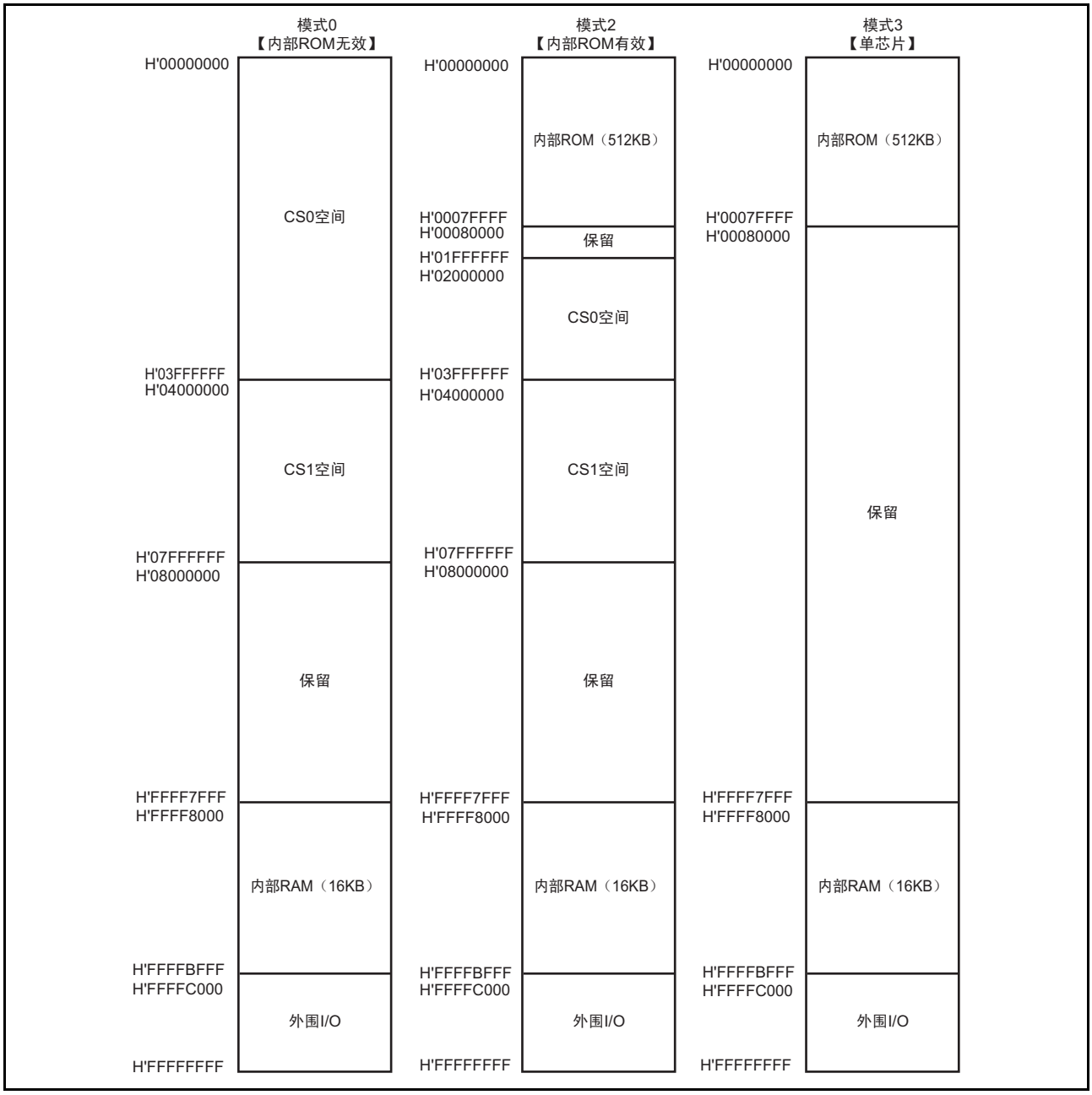


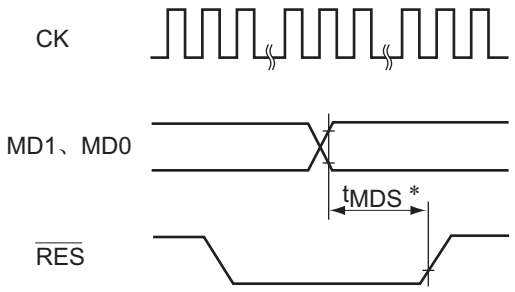
图 3.4 内部 ROM512KB/ 内部 RAM16KB 版各运行模式的地址映射

3.5　本 LSI 的初始状态

为降低功耗，本 LSI 在初始状态将部分内部模块设定为模块待机状态。因此，如果要运行这些模块，必须解除模块待机状态。详情参阅 “第 22 章 低功耗模式”。

3.6　更改运行模式时的注意事项

对本 LSI 外加电源时如果更改运行模式，必须在上电复位状态（对 $\overline{\text{RES}}$ 引脚外加低电平）进行。



【注】 *　请参阅“ 25.3.2　控制信号时序 ”。

图 3.5　更改运行模式时的复位输入时序

第 4 章 时钟振荡器 （CPG）

本 LSI 内置时钟振荡器 （CPG）。

CPG 可生成内部时钟 （Iϕ）、总线时钟 （Bϕ）、外围时钟 （Pϕ）及用于 MTU2S、 MTU2 模块的时钟（MIϕ、 MPϕ），并控制低功耗模式。

4.1 特点

- 可独立生成5种时钟
  可独立生成CPU使用的内部时钟 （Iϕ）、外围模块使用的外围时钟 （Pϕ）及外部总线接口使用的总线时钟 （Bϕ=CK）。
  另外，还可独立生成内部MTU2S模块使用的MTU2S时钟 （MIϕ）及MTU2模块使用的MTU2时钟（MPϕ）。
- 频率更改功能
  由CPG内部的分频电路，可独立更改内部时钟 （Iϕ）、总线时钟 （Bϕ）、外围时钟 （Pϕ）、 MTU2S时钟 （MIϕ）及MTU2时钟 （MPϕ）的频率。通过设定频率控制寄存器 （FRQCR），由软件更改频率。
- 低功耗模式的控制
  在睡眠模式或待机模式可停止时钟及停止模块待机功能中的特定模块。
- 振荡停止检测功能
  因某种原因停止从时钟输入引脚提供时钟时，可自动将定时器引脚设置为高阻抗状态。

时钟振荡器框图如图 4.1 所示。

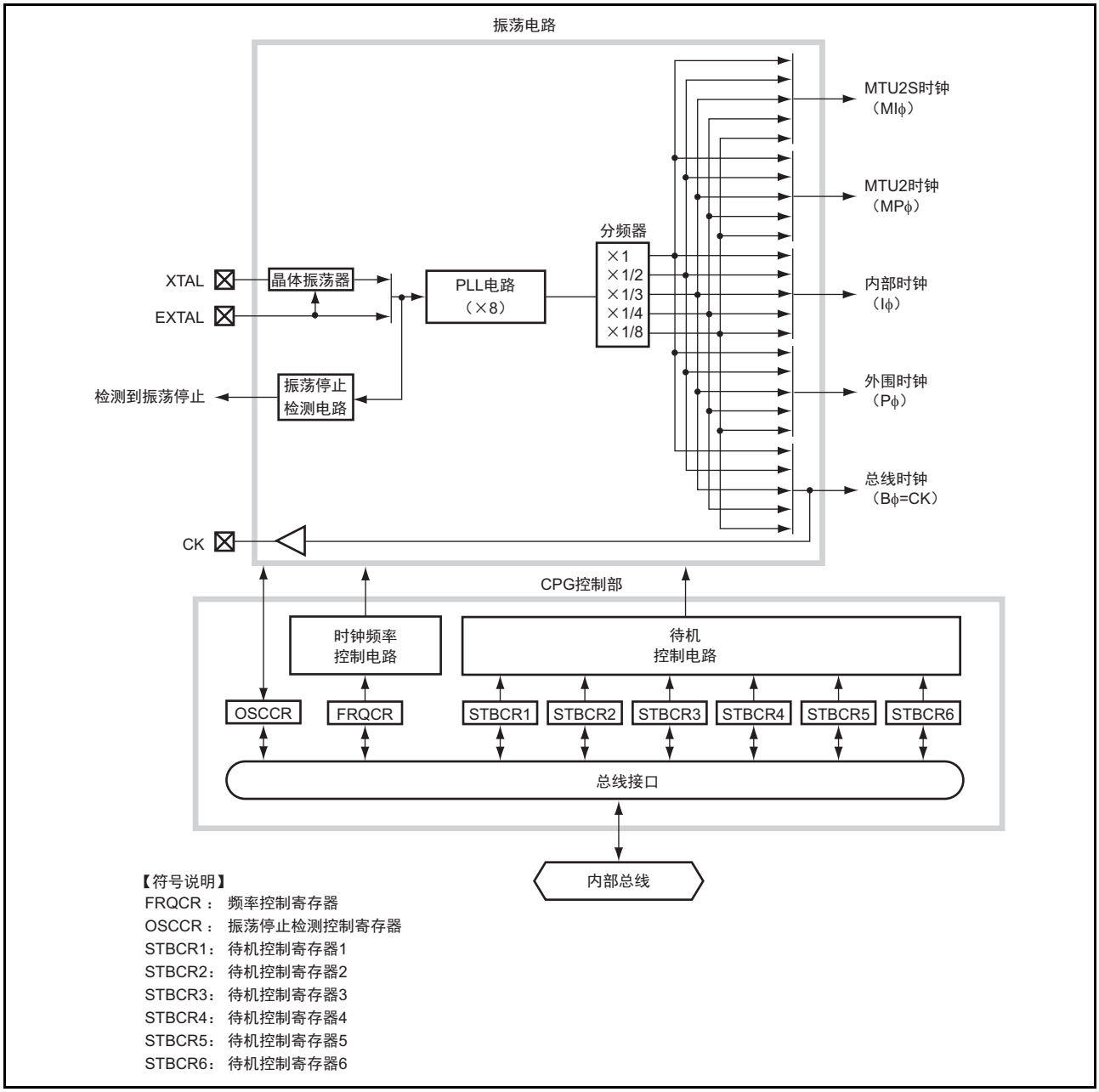


图 4.1　时钟振荡器框图

时钟振荡器各块的功能如下所示：

(1)　PLL 电路

PLL 电路可将晶体振荡器或 EXTAL 引脚的输入时钟频率增大 8 倍，倍频比总是固定为 8 倍。

(2)　晶体振荡器

将晶体谐振器连接到 XTAL、 EXTAL 引脚使用的振荡电路。

(3)　分频器

分频器可生成内部时钟 （Iϕ）、总线时钟 （Bϕ）、外围时钟 （Pϕ）、 MTU2S 时钟 （MIϕ）及 MTU2 时钟 （MPϕ）所使用的工作频率时钟。

工作频率可根据 PLL 电路的输出频率，选择 1 倍、 1/2 倍、 1/3 倍、 1/4 倍、 1/8 倍。

由频率控制寄存器 （FRQCR）设定分频比。

(4)　振荡停止检测电路

检测晶体振荡器异常的电路。

(5)　时钟频率控制电路

时钟频率控制电路由设定频率控制寄存器 （FRQCR），控制时钟频率。

(6)　待机控制电路

待机控制电路控制睡眠 / 待机模式的内部振荡电路及其他模块的状态。

(7)　频率控制寄存器 （FRQCR）

在频率控制寄存器 （FRQCR）分配内部时钟 （Iϕ）、总线时钟 （Bϕ）、外围时钟 （Pϕ）、 MTU2S 时钟 （MIϕ）及 MTU2 时钟 （MPϕ）频率分频比的各控制位。

(8)　振荡停止检测控制寄存器 （OSCCR）

在振荡停止检测控制寄存器 （OSCCR）分配振荡停止检测标志与外部引脚的标志输出选择位。

(9)　待机控制寄存器 1 ～ 6 （STBCR1 ～ STBCR6）

在待机控制寄存器 （STBCR）分配低功耗模式的各控制位。有关待机控制寄存器，详情参阅 “第 22 章 低功耗模式”。

各模块的运行时钟如表 4.1 所示。

表 4.1　各模块的运行时钟

运行时钟	相应模块
内部时钟 （Iϕ）	CPU UBC ROM RAM AUD
总线时钟 （Bϕ）	BSC DTC

运行时钟	相应模块
外围时钟 （Pϕ）	POE SCI SSU A/D CMT RCAN-ET WDT
MTU2 时钟 （MPϕ）	MTU2
MTU2S 时钟 （MIϕ）	MTU2S

4.2　输入 / 输出引脚

CPG 的引脚结构与功能如表 4.2 所示。

表 4.2　时钟振荡器的引脚结构与功能

名称	引脚名称	输入 / 输出	功能
晶体输入 / 输出引脚（时钟输入引脚）	XTAL	输出	连接晶体谐振器。
	EXTAL	输入	连接晶体谐振器或用作外部时钟输入引脚。
时钟输出引脚	CK	输出	用作外部时钟输出引脚。

【注】 使用时钟输出引脚 （CK）时，有时需要由引脚功能控制器 （PFC）设定引脚。详情参阅 “第 18 章 引脚功能控制器 （PFC）”。

4.3　时钟运行模式

本 LSI 的时钟运行模式如表 4.3 所示。

表 4.3　时钟运行模式

模式	时钟输入 / 输出		PLL 电路	向分频器输入
	提供源	输出		
1	EXTAL 输入 晶体谐振器	CK*	ON （×8）	×8

【注】 * 从 CK 引脚输出时钟时，必须设定 PFC。
有关PFC的设定，详情参阅 “第18章 引脚功能控制器 （PFC）”。

- 模式1

从 EXTAL 引脚输入外部时钟， PLL 电路将频率增大 8 倍后提供给本 LSI 内部，因此，亦可允许较低的外部生成时钟频率。可使用的输入时钟频率范围为 8MHz ～ 10MHz，内部时钟 （Iϕ）的频率范围为 10MHz ～ 80MHz。

最大工作频率：Iϕ=80MHz、 Bϕ=40MHz、 Pϕ=40MHz、 MIϕ=80MHz、 MPϕ=40MHz
（T_{opr}= – 40 ～ +85°C）

最大工作频率：Iϕ=64MHz、 Bϕ=32MHz、 Pϕ=32MHz、 MIϕ=64MHz、 MPϕ=32MHz
（T_{opr}= – 40 ～ +125°C）

FRQCR 可设定的分频比如表 4.4 所示。

表 4.4　FRQCR 可设定的分频比

PLL 倍频比	FRQCR 的分频比设定					时钟比					时钟频率 （MHz） *					
	Iϕ	Bϕ	Pϕ	MIϕ	MPϕ	Iϕ	Bϕ	Pϕ	MIϕ	MPϕ	输入时钟	Iϕ	Bϕ	Pϕ	MIϕ	MPϕ
×8	1/4	1/4	1/8	1/8	1/8	2	2	1	1	1	10	20	20	10	10	10
	1/4	1/4	1/8	1/4	1/8	2	2	1	2	1		20	20	10	20	10
	1/4	1/4	1/8	1/4	1/4	2	2	1	2	2		20	20	10	20	20
	1/4	1/4	1/4	1/4	1/4	2	2	2	2	2		20	20	20	20	20
	1/3	1/3	1/3	1/3	1/3	8/3	8/3	8/3	8/3	8/3		26	26	26	26	26
	1/2	1/4	1/8	1/8	1/8	4	2	1	1	1		40	20	10	10	10
	1/2	1/4	1/8	1/4	1/8	4	2	1	2	1		40	20	10	20	10
	1/2	1/4	1/8	1/4	1/4	4	2	1	2	2		40	20	10	20	20
	1/2	1/4	1/8	1/2	1/8	4	2	1	4	1		40	20	10	40	10
	1/2	1/4	1/8	1/2	1/4	4	2	1	4	2		40	20	10	40	20
	1/2	1/4	1/4	1/4	1/4	4	2	2	2	2		40	20	20	20	20
	1/2	1/4	1/4	1/2	1/4	4	2	2	4	2		40	20	20	40	20
	1/2	1/2	1/8	1/8	1/8	4	4	1	1	1		40	40	10	10	10
	1/2	1/2	1/8	1/4	1/8	4	4	1	2	1		40	40	10	20	10
	1/2	1/2	1/8	1/4	1/4	4	4	1	2	2		40	40	10	20	20
	1/2	1/2	1/8	1/2	1/8	4	4	1	4	1		40	40	10	40	10
	1/2	1/2	1/8	1/2	1/4	4	4	1	4	2		40	40	10	40	20
	1/2	1/2	1/8	1/2	1/2	4	4	1	4	4		40	40	10	40	40
	1/2	1/2	1/4	1/4	1/4	4	4	2	2	2		40	40	20	20	20
	1/2	1/2	1/4	1/2	1/4	4	4	2	4	2		40	40	20	40	20
	1/2	1/2	1/4	1/2	1/2	4	4	2	4	4		40	40	20	40	40
	1/2	1/2	1/2	1/2	1/2	4	4	4	4	4		40	40	40	40	40
	1/1	1/4	1/8	1/8	1/8	8	2	1	1	1		80	20	10	10	10
	1/1	1/4	1/8	1/4	1/8	8	2	1	2	1		80	20	10	20	10
	1/1	1/4	1/8	1/4	1/4	8	2	1	2	2		80	20	10	20	20
	1/1	1/4	1/8	1/2	1/8	8	2	1	4	1		80	20	10	40	10
	1/1	1/4	1/8	1/2	1/4	8	2	1	4	2		80	20	10	40	20
	1/1	1/4	1/8	1/1	1/8	8	2	1	8	1		80	20	10	80	10
	1/1	1/4	1/8	1/1	1/4	8	2	1	8	2		80	20	10	80	20
	1/1	1/4	1/4	1/4	1/4	8	2	2	2	2		80	20	20	20	20
	1/1	1/4	1/4	1/2	1/4	8	2	2	4	2		80	20	20	40	20
	1/1	1/4	1/4	1/1	1/4	8	2	2	8	2		80	20	20	80	20
	1/1	1/3	1/3	1/3	1/3	8	8/3	8/3	8/3	8/3		80	26	26	26	26
	1/1	1/3	1/3	1/1	1/3	8	8/3	8/3	8	8/3		80	26	26	80	26

PLL 倍频比	FRQCR 的分频比设定					时钟比					时钟频率 （MHz） *					
	Iϕ	Bϕ	Pϕ	MIϕ	MPϕ	Iϕ	Bϕ	Pϕ	MIϕ	MPϕ	输入时钟	Iϕ	Bϕ	Pϕ	MIϕ	MPϕ
×8	1/1	1/2	1/8	1/8	1/8	8	4	1	1	1	10	80	40	10	10	10
	1/1	1/2	1/8	1/4	1/8	8	4	1	2	1		80	40	10	20	10
	1/1	1/2	1/8	1/4	1/4	8	4	1	2	2		80	40	10	20	20
	1/1	1/2	1/8	1/2	1/8	8	4	1	4	1		80	40	10	40	10
	1/1	1/2	1/8	1/2	1/4	8	4	1	4	2		80	40	10	40	20
	1/1	1/2	1/8	1/2	1/2	8	4	1	4	4		80	40	10	40	40
	1/1	1/2	1/8	1/1	1/8	8	4	1	8	1		80	40	10	80	10
	1/1	1/2	1/8	1/1	1/4	8	4	1	8	2		80	40	10	80	20
	1/1	1/2	1/8	1/1	1/2	8	4	1	8	4		80	40	20	80	40
	1/1	1/2	1/4	1/4	1/4	8	4	2	2	2		80	40	20	20	20
	1/1	1/2	1/4	1/2	1/4	8	4	2	4	2		80	40	20	40	20
	1/1	1/2	1/4	1/2	1/2	8	4	2	4	4		80	40	20	40	40
	1/1	1/2	1/4	1/1	1/4	8	4	2	8	2		80	40	20	80	20
	1/1	1/2	1/4	1/1	1/2	8	4	2	8	4		80	40	20	80	40
	1/1	1/2	1/2	1/2	1/2	8	4	4	4	4		80	40	40	40	40
	1/1	1/2	1/2	1/1	1/2	8	4	4	8	4		80	40	40	80	40

【注】 * 时钟频率假设为输入时钟频率时的值。

1. PLL 电路的倍频比仅为 ×8。分频器的分频比可选择 ×1、 ×1/2、 ×1/3、 ×1/4、 ×1/8。
   可由频率控制寄存器对每个设定的时钟设定分频比。
2. PLL 电路的输出频率是晶体谐振器或 EXTAL 引脚的输入时钟频率乘 PLL 电路的 8 倍倍频比的频率。
3. 分频器的输入总是为 PLL 电路的输出。
4. 内部时钟 （Iϕ）的频率是晶体谐振器或 EXTAL 引脚的输入时钟频率乘 PLL 电路的 8 倍倍频比和分频器的分频比的频率。
   内部时钟 （Iϕ）的频率必须设定为小于等于最大工作频率 （80MHz）。
5. 总线时钟 （Bϕ）的频率是晶体谐振器或 EXTAL 引脚的输入时钟频率乘 PLL 电路的 8 倍倍频比和分频器的分频比的频率。
   总线时钟 （Bϕ）的频率必须设定为小于等于40MHz，并且小于等于内部时钟 （Iϕ）的频率。
6. 外围时钟 （Pϕ）的频率是晶体谐振器或 EXTAL 引脚的输入时钟频率乘 PLL 电路的 8 倍倍频比和分频器的分频比的频率。
   外围时钟 （Pϕ）的频率必须设定为小于等于40MHz，并且小于等于总线时钟 （Bϕ）的频率。
7. 使用 MTU2S 及 MTU2 时， MTU2S 时钟 （MIϕ）的频率必须设定为小于等于内部时钟 （Iϕ）的频率，并且大于等于 MTU2 时钟 （MPϕ）的频率。 MTU2 时钟 （MPϕ）的频率必须设定为小于等于 MTU2S 时钟 （MIϕ）及总线时钟 （Bϕ）的频率，并且大于等于外围时钟 （Pϕ）的频率。
   MTU2S时钟 （MIϕ）及MTU2时钟 （MPϕ）的频率是晶体谐振器或EXTAL引脚的输入时钟频率乘PLL电路的8倍倍频比和分频器的分频比的频率。
8. CK 引脚的频率总是与总线时钟 （Bϕ）的频率相等。

4.4　寄存器说明

CPG 有以下寄存器。有关寄存器的地址及各处理状态的寄存器状态，详情参阅 “第 24 章 寄存器一览表”。

表 4.5　寄存器结构

寄存器名称	简称	R/W	初始值	地址	存取长度
频率控制寄存器	FRQCR	R/W	H'36DB	H'FFFFE800	16
振荡停止检测控制寄存器	OSCCR	R/W	H'00	H'FFFFE814	8

4.4.1　频率控制寄存器 （FRQCR）

FRQCR 为可读取 / 写入的 16 位寄存器，可指定内部时钟 （Iϕ）、总线时钟 （Bϕ）、外围时钟 （Pϕ）、MTU2S 时钟 （MIϕ）及 MTU2 时钟 （MPϕ）的频率分频比。 FRQCR 仅可执行字存取。

仅在上电复位时 （WDT 上溢引起的上电复位除外），初始化本寄存器，初始值为 H'36DB。

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

位	位名称	初始值	R/W	说　明
15	—	0	R	保留位 读取值、写入值总是为 0。
14 ～ 12	IFC[2:0]	011	R/W	内部时钟 （Iϕ）的频率分频比 根据 PLL 电路的输出频率，指定内部时钟 （Iϕ）的分频比。如果设定禁止设定的值，则无法保证之后的运行。 000：×1 倍 001：×1/2 倍 010：×1/3 倍 011：×1/4 倍 100：×1/8 倍 上述以外：禁止设定
11 ～ 9	BFC[2:0]	011	R/W	总线时钟 （Bϕ）的频率分频比 根据 PLL 电路的输出频率，指定总线时钟 （Bϕ）的分频比。如果设定禁止设定的值，则无法保证之后的运行。 000：×1 倍 001：×1/2 倍 010：×1/3 倍 011：×1/4 倍 100：×1/8 倍 上述以外：禁止设定
8 ～ 6	PFC[2:0]	011	R/W	外围时钟 （Pϕ）的频率分频比 根据 PLL 电路的输出频率，指定外围时钟 （Pϕ）的分频比。如果设定禁止设定的值，则无法保证之后的运行。 000：×1 倍 001：×1/2 倍 010：×1/3 倍 011：×1/4 倍 100：×1/8 倍 上述以外：禁止设定

位	位名称	初始值	R/W	说　明
5 ～ 3	MIFC[2:0]	011	R/W	MTU2S 时钟 （MIϕ）的频率分频比 根据 PLL 电路的输出频率，指定 MTU2S 时钟 （MIϕ）的分频比。如果设定禁止设定的值，则无法保证之后的运行。 000：×1 倍 001：×1/2 倍 010：×1/3 倍 011：×1/4 倍 100：×1/8 倍 上述以外：禁止设定
2 ～ 0	MPFC[2:0]	011	R/W	MTU2 时钟 （MPϕ）的频率分频比 根据 PLL 电路的输出频率，指定 MTU2 时钟 （MPϕ）的分频比。如果设定禁止设定的值，则无法保证之后的运行。 000：×1 倍 001：×1/2 倍 010：×1/3 倍 011：×1/4 倍 100：×1/8 倍 上述以外：禁止设定

4.4.2　振荡停止检测控制寄存器 （OSCCR）

OSCCR 为可读取 / 写入的 8 位寄存器，选择振荡停止检测标志及外部引脚的标志输出。 OSCCR 仅可执行字节存取。

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

位	位名称	初始值	R/W	说　明
7 ～ 3	—	均为 0	R	保留位 读取值、写入值总是为 0。
2	OSCSTOP	0	R	振荡停止检测标志 ［置位条件］ • 普通运行过程中，检测到时钟输入停止时 • 转移至软件待机模式时 ［清除条件］ • $\overline{\text{RES}}$ 引脚的上电复位时 • 从软件待机模式返回时
1	—	0	R	保留位 读取值、写入值总是为 0。
0	OSCERS	0	R/W	振荡停止检测标志输出选择 选择是否从 $\overline{\text{WDTOVF}}$ 引脚输出振荡停止检测标志信号。 0：从 $\overline{\text{WDTOVF}}$ 引脚仅输出 WDT 上溢信号。 1：从 $\overline{\text{WDTOVF}}$ 引脚输出 WDT 上溢信号及振荡停止检测标志信号。

4.5　频率更改方法

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

1. 初始状态为IFC2～IFC0=H'011 （×1/4倍）、 BFC2～BFC0=H'011 （×1/4倍）、 PFC2～PFC0=H'011 （×1/4倍）、 MIFC2～MIFC0=H'011 （×1/4倍）、 MPFC2～MPFC0=H'011 （×1/4倍）。
2. 停止除CPU、内部ROM及内部RAM之外的模块。
3. 设定IFC2～IFC0、 BFC2～BFC0、 PFC2～PFC0、 MIFC2～MIFC0、 MPFC2～MPFC0位为目标值。PLL电路的倍频比固定为×8倍，因此频率仅取决于分频比的设定。此时，频率必须设定为内部时钟（Iϕ）≥总线时钟 （Bϕ）≥外围时钟 （Pϕ）。使用MTU2S时钟及MTU2时钟时，必须设定为内部时钟 （Iϕ）≥MTU2S时钟 （MIϕ）≥MTU2时钟 （MPϕ）≥外围时钟 （Pϕ），并且总线时钟 （Bϕ）≥MTU2时钟 （MPϕ）。
   必须在内部ROM或内部RAM执行FRQCR的改写处理。
4. 发行FRQCR改写指令，并在(1～24n)cyc+11Bϕ+7Pϕ后，执行切换。
   n：FRQCR的BFC位所设定的分频比 （1、 1/2、 1/3、 1/4、 1/8）
   cyc：通过PLL将 EXTAL8倍频后的时钟

【注】 (1 ～ 24n) 因内部状态不同而变化。

4.6　振荡器

提供时钟的方法有连接晶体谐振器与输入外部时钟两种。

4.6.1　连接晶体谐振器的方法

连接晶体谐振器时的例子如图 4.2 所示。必须使用如表 4.6 所示的阻尼电阻 R_d，且晶体谐振器的振荡频率必须为 8 ～ 10MHz。

有关晶体与 LSI 的兼容性，请咨询晶体制造商。

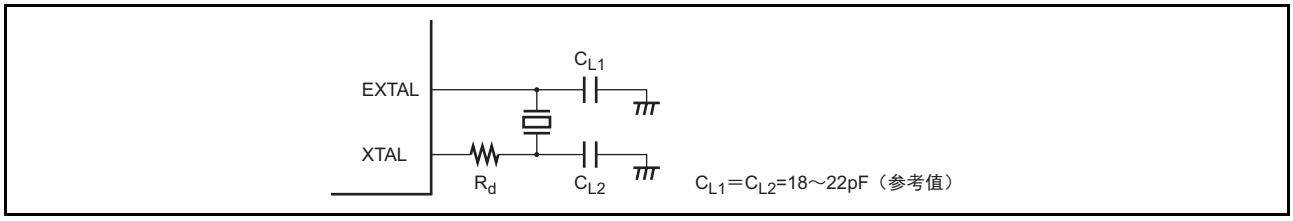


图 4.2　晶体谐振器的连接例

表 4.6　阻尼电阻值 （参考值）

频率 （MHz）	8	10
R_d （Ω）（参考值）	200	0

晶体谐振器的等效电路如图 4.3 所示。晶体谐振器必须使用表 4.7 所示的特性。

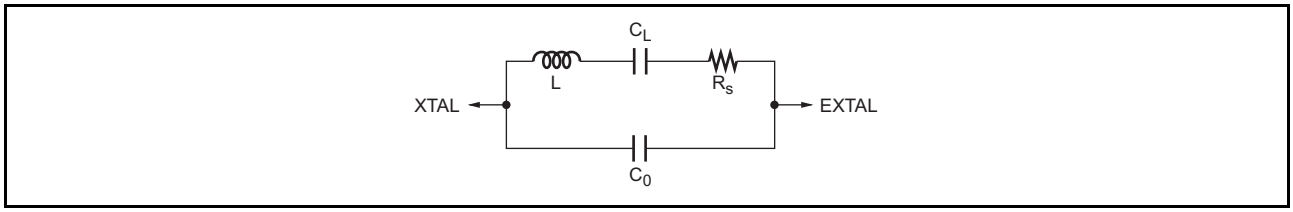


图 4.3　晶体谐振器的等效电路

表 4.7　晶体谐振器的特性

频率 （MHz）	8	10
R_s Max. （Ω）（参考值）	80	60
C_0 Max. （pF）（参考值）	7	

4.6.2　输入外部时钟的方法

输入外部时钟的连接例子如图 4.4 所示。如果在软件待机模式停止外部时钟，必须设置为高电平。运行时，外部输入时钟的频率必须设置为 8 ～ 10MHz。 XTAL 引脚的寄生电容必须≤ 10pF。

即使输入外部时钟，为确保 PLL 稳定时间，在通电或解除软件待机时，等待时间也必须大于等于振荡稳定时间。

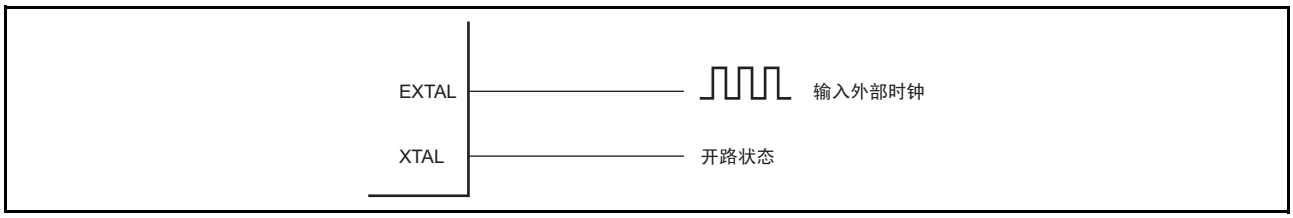


图 4.4　外部时钟的连接例

4.7　振荡停止检测功能

因某种系统异常引起振荡器停止时，本 CPG 可检测到时钟停止。

检测到 EXTAL 输入在一定周期内无变化时，将 OSCCR 寄存器的 OSCSTOP 位置 1，并在 $\overline{\text{RES}}$ 引脚的上电复位或解除软件待机模式前，保持此状态。此时，如果设定 OSCERS 位为 1，则从 $\overline{\text{WDTOVF}}$ 引脚输出振荡停止检测标志信号。与 PFC 的设定无关，可将大电流端口 （MTU2 的 TIOC3B、 TIOC3D、 TIOC4A ～ TIOC4D、 MTU2S 的 TIOC3BS、 TIOC3DS、 TIOC4AS ～ TIOC4DS 为多路复用引脚）设置为高阻抗。详情参阅 “附录 A. 引脚状态”。

即使在软件待机状态，上述引脚也可设置为高阻抗。详情参阅 “附录 A. 引脚状态”。解除软件待机状态后，返回普通运行。如果在软件待机状态之外，执行如停止振荡等异常运行，则其他 LSI 的运行会不稳定。此时，即使重新开始振荡，包含上述引脚在内 LSI 的运行仍不稳定。

即使 EXTAL 输入无变化，本 LSI 的 PLL 电路也继续在 100kHz ～ 10MHz 间 （因温度、工作电压而变化）振荡。

4.8　使用时的注意事项

4.8.1　谐振器的注意事项

谐振器的各种特性与用户的电路板设计密切相关，用户须参照本章所介绍的谐振器连接例子，在使用前进行详细评估。振荡电路的电路常数因谐振器、安装电路的寄生电容等不同而各异，因此，须要与谐振器制造商充分沟通后再确定。请注意：振荡引脚的外加电压不得超过最大额定电压。

4.8.2　电路板设计时的注意事项

本 LSI 已采取抗辐射噪声处理，但如果需进一步降低辐射噪声，推荐使用多层电路板及系统接地专用层。

使用晶体谐振器时，必须尽量将谐振器与负载电容配置在 XTAL、 EXTAL 引脚附近。如图 4.5 所示，振荡电路附近不得通过信号线，否则，可能因感应而无法正常振荡。

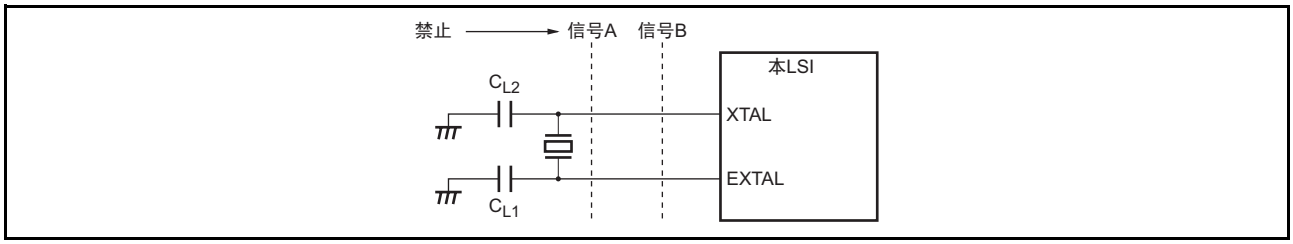


图 4.5　设计振荡电路部分的电路板时的注意事项

PLL 电路的外接推荐电路如图 4.6 所示。必须将 $PLLV_{SS}$ 、 V_{CC} 与 V_{SS} 从电源提供源分离，并在引脚附近插入旁路电容 CB 与 CPB。

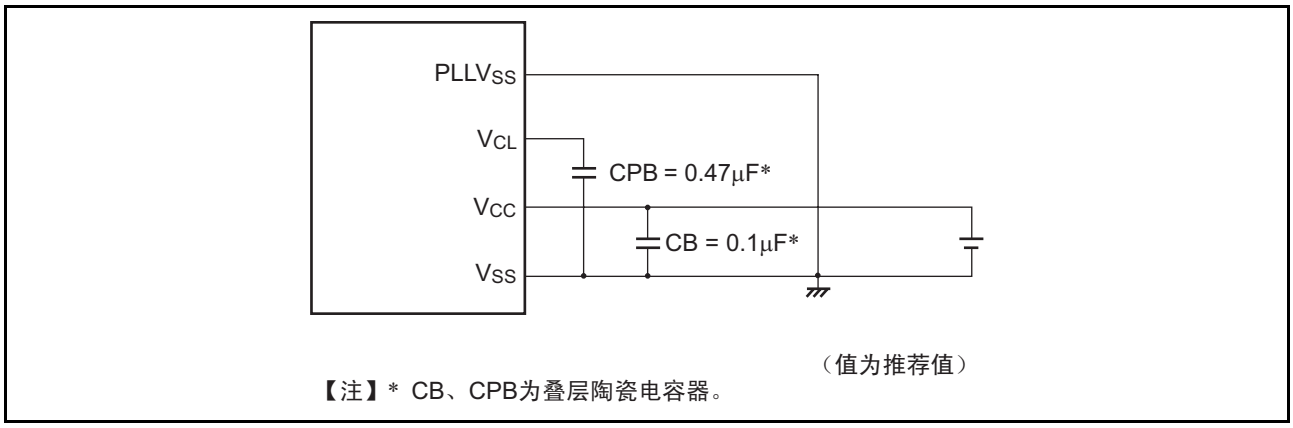


图 4.6　PLL 电路的外接推荐电路

第 5 章　异常处理

5.1　概要

5.1.1　异常处理的种类与优先顺序

如表 5.1 所示，由复位、地址错误、中断及指令等异常源启动异常处理。异常源设有优先顺序，如果同时产生多个异常源，则按表 5.1 所示优先顺序接受并处理。

表 5.1　异常源的种类与优先顺序

	异常处理	优先顺序
复位	上电复位	高
	手动复位	↑
中断	用户断点 （指令执行前的断点）	
地址错误	CPU 地址错误 （取指令）	
指令	一般非法指令 （未定义代码）	
	槽非法指令 （紧接着延迟转移指令 *1 之后的未定义代码或 PC 改写指令 *2）	
	陷阱指令 （TRAPA 指令）	
地址错误	CPU 地址错误 （数据存取）	
中断	用户断点 （执行指令后的断点或操作数断点）	
地址错误	DTC 地址错误 （数据存取）	
中断	NMI	
	IRQ	↓
	内部外围模块	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、LDC Rm,SR、 LDC.L@Rm+,SR

5.1.2 异常处理的运行

各异常源按表 5.2 所示的时序检测及处理。

表 5.2 异常源检测与异常处理开始的时序

异常处理		异常源检测及处理开始的时序
复位	上电复位	$\overline{\text{RES}}$ 引脚从低电平变为高电平或 WDT 上溢时，开始异常处理。
	手动复位	$\overline{\text{MRES}}$ 引脚从低电平变为高电平或 WDT 上溢时，开始异常处理。
地址错误		在指令解码阶段检测出地址错误，并在完成当前执行的指令之后，开始异常处理。
中断		
指令	陷阱指令	通过执行 TRAPA 指令，可开始异常处理。
	一般非法指令	解码除延迟转移指令 （延迟槽）之外的未定义代码时，开始异常处理。
	槽非法指令	解码配置于延迟转移指令 （延迟槽）后的未定义代码或 PC 改写指令时，开始异常处理。

启动异常处理时， CPU 运行如下：

(1) 由复位引起的异常处理

从异常处理向量表 （PC 与 SP 在上电复位时地址为 H'00000000、 H'00000004 ；手动复位时地址为 H'00000008、 H'0000000C）取出程序计数器 （PC）与堆栈指针 （SP）的初始值。有关异常处理向量表，详情参阅 “5.1.3 异常处理向量表”。将向量基址寄存器 （VBR）设置为 H'00000000、状态寄存器 （SR）的中断屏蔽位 （I3 ～ I0）设置为 H'F （B'1111）。从异常处理向量表取出的 PC 地址开始执行程序。

(2) 由地址错误、中断及指令引起的异常处理

将 SR 与 PC 保存至 R15 所示的堆栈。中断异常处理时，对 SR 的中断屏蔽位 （I3 ～ I0）写入中断优先级。由地址错误或指令引起的异常处理时， I3 ～ I0 位不受影响。从异常处理向量表取出起始地址，并从此地址开始执行程序。

5.1.3 异常处理向量表

执行异常处理前，需预先在存储器设定异常处理向量表。异常处理向量表保存异常服务程序的起始地址（复位异常处理表保存 PC 与 SP 的初始值）。

对各异常源分别分配不同的向量编号与向量表地址偏移量。由对应的向量编号和向量表地址偏移量计算向量表地址。异常处理时，从该向量表地址所示的异常处理向量表取出异常服务程序的起始地址。

向量编号与向量表地址偏移量如表 5.3 所示，向量表地址的计算方法如表 5.4 所示。

表 5.3　异常处理向量表

异常源		向量编号	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
DTC 地址错误		10	H'00000028 ～ H'0000002B
中断	NMI	11	H'0000002C ～ H'0000002F
	用户断点	12	H'00000030 ～ H'00000033
（系统保留）		13	H'00000034 ～ H'00000037
		…	…
		31	H'0000007C ～ H'0000007F
陷阱指令 （用户向量）		32	H'00000080 ～ H'00000083
		…	…
		63	H'000000FC ～ H'000000FF
中断	IRQ0	64	H'00000100 ～ H'00000103
	IRQ1	65	H'00000104 ～ H'00000107
	IRQ2	66	H'00000108 ～ H'0000010B
	IRQ3	67	H'0000010C ～ H'0000010F
（系统保留）		68	H'00000110 ～ H'00000113
		69	H'00000114 ～ H'00000117
		70	H'00000118 ～ H'0000011B
		71	H'0000011C ～ H'0000011F
内部外围模块 *		72	H'00000120 ～ H'00000123
		…	…
		255	H'000003FC ～ H'000003FF

【注】 *　各内部外围模块中断的向量编号与向量表偏移量，参照表 6.3。

表 5.4　异常处理向量表地址的计算方法

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

【注】 VBR：向量基址寄存器

向量表地址偏移量：参照表 5.3

向量编号：参照表 5.3

5.2 复位

5.2.1 复位的种类

复位是优先顺序最高的异常处理源，有上电复位与手动复位两种。如表 5.5 所示，在上电复位或手动复位时， CPU 的状态均被初始化。上电复位时，初始化内部外围模块的寄存器，而手动复位时，则不初始化。

表 5.5 复位状态

种类	转移至复位状态的条件			内部状态		
	$\overline{\text{RES}}$	WDT 上溢	$\overline{\text{MRES}}$	CPU/INTC	内部外围模块	POE、PFC、I/O 端口
上电复位	低	—	—	初始化	初始化	初始化
	高	上溢	高	初始化	初始化	初始化
手动复位	高	不上溢	低	初始化	不初始化	不初始化

5.2.2 上电复位

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

如果 $\overline{\text{RES}}$ 引脚设置为低电平，则本 LSI 变为上电复位状态。为确保本 LSI 复位，在接通电源或待机时（时钟停止运行时）， $\overline{\text{RES}}$ 引脚保持低电平的时间必须为振荡稳定时间；在时钟运行时，则必须大于等于 20tcyc。有关上电复位的状态， CPU 的内部状态与内部外围模块的寄存器均被初始化。有关上电复位状态的各引脚状态，详情参阅 “附录 A. 引脚状态”。

在上电复位状态，如果在 $\overline{\text{RES}}$ 引脚保持一定时间的低电平后，设置为高电平，则开始上电复位异常处理。此时， CPU 如下运行：

1. 从异常处理向量表取出程序计数器 （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，并将状态寄存器 （SR）的中断屏蔽位 （I3～I0）设置为H'F （B'1111）。
4. 将从异常处理向量表取出的值分别设定到PC与SP，并开始执行程序。

另外，必须在接通系统电源时，执行上电复位处理。

(2) 由 WDT 引起的上电复位

设定为在 WDT 看门狗定时器模式产生上电复位，如果 WDT 的 WTCNT 上溢，则变为上电复位状态。

此时，在由 WDT 引起的复位状态，不初始化时钟振荡器 （CPG）的频率控制寄存器 （FRQCR）及看门狗定时器 （WDT）的寄存器 （仅在由 $\overline{\text{RES}}$ 引脚引起的上电复位状态，此寄存器才被初始化）。

另外，如果同时产生由 $\overline{\text{RES}}$ 引脚输入信号引起的复位和由 WDT 上溢引起的复位，则优先执行由 $\overline{\text{RES}}$ 引脚引起的复位，并将 WTCSR 的 WOVF 位清 0。开始执行由 WDT 引起的上电复位处理时， CPU 如下运行：

1. 从异常处理向量表取出程序计数器 （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，并将状态寄存器 （SR）的中断屏蔽位 （I3～I0）设置为H'F （B'1111）。
4. 将从异常处理向量表取出的值分别设定到PC和SP，并开始执行程序。

5.2.3 手动复位

$\overline{\text{RES}}$ 引脚为高电平时，如果将 $\overline{\text{MRES}}$ 引脚设置为低电平，则本 LSI 变为手动复位状态。为确保本 LSI 复位，软件待机时（时钟停止时），$\overline{\text{MRES}}$ 引脚保持低电平的时间必须大于 WDT 设定的振荡稳定时间；时钟运行时，必须大于等于 20tcyc。在手动复位状态，初始化 CPU 的内部状态，但不初始化内部外围模块的各寄存器。如果在总线周期中途设定为手动复位状态，则等到总线周期结束，开始手动复位异常处理。因此，总线周期不会因手动复位而中途停止。但，如果 $\overline{\text{MRES}}$ 一旦设置为低电平，则必须保持低电平到总线周期结束，并进入手动复位状态为止（保持低电平的时间必须大于等于最长总线周期）。有关手动复位状态的各引脚状态，详情参阅 “附录 A. 引脚状态”。

在手动复位状态，如果在 $\overline{\text{MRES}}$ 引脚保持一定时间的低电平后，设置为高电平，则开始手动复位异常处理。此时，CPU 按照与上电复位异常处理相同的步骤运行。

5.3 地址错误

5.3.1 地址错误的产生源

如表 5.6 所示，取指令或读取 / 写入数据时，产生地址错误。

表 5.6 总线周期与地址错误

总线周期		总线周期的内容	地址错误的产生
种类	总线主控器		
取指令	CPU	从偶数地址取指令	无（正常）
		从奇数地址取指令	产生地址错误
		从内部外围模块空间以外取指令	无（正常）
		从内部外围模块空间取指令	产生地址错误
		单芯片模式时从外部存储器空间取指令	产生地址错误
读取 / 写入数据	CPU 或 DTC	从偶数地址存取字数据	无（正常）
		从奇数地址存取字数据	产生地址错误
		从长字边界存取长字数据	无（正常）
		从长字边界以外存取长字数据	产生地址错误
		在内部外围模块空间存取字数据和字节数据	无（正常）
		在 16 位内部外围模块空间存取长字数据	无（正常）
		在 8 位内部外围模块空间存取长字数据	无（正常）
		单芯片模式时存取外部存储器空间	产生地址错误

5.3.2 地址错误异常处理

产生地址错误时，引起地址错误的总线周期结束，并在当前执行指令结束后，开始地址错误异常处理。此时，CPU 如下运行：

1. 将状态寄存器（SR）压栈。
2. 将程序计数器（PC）压栈。保存的PC值为产生本异常的指令地址。但，如果将产生本异常的指令分配到延迟槽，则保存前一条延迟转移指令的地址。
3. 从产生地址错误所对应的异常处理向量表取出异常服务程序起始地址，并从该地址执行程序。此时的跳转不是延迟转移。

5.4　中断

5.4.1　中断源

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

表 5.7　中断源

种类	请求源	中断源个数
NMI	NMI 引脚 （从外部输入）	1
用户断点	用户断点控制器 （UBC）	1
IRQ	IRQ0 ～ IRQ3 引脚 （从外部输入）	4
内部外围模块	多功能定时器脉冲单元 2 （MTU2）	25
	多功能定时器脉冲单元 2S （MTU2S）	13
	数据传送控制器 （DTC）	1
	看门狗定时器 （WDT）	1
	A/D 转换器 （A/D_0、 A/D_1）	2
	比较匹配定时器 （CMT_0、 CMT_1）	2
	串行通信接口 （SCI_0、 SCI_1、 SCI_2）	12
	同步串行通信单元 （SSU）	3
	端口输出允许 （POE）	3
	控制器局域网 （RCAN-ET）	5/10*

【注】 *　仅适用于 SH7142

对各中断源分别分配不同的向量编号与向量表地址偏移量。向量编号与向量表地址偏移量参照表 6.3。

5.4.2　中断优先顺序

中断源预设有优先顺序，同时产生多个中断时 （多重中断），由中断控制器 （INTC）判断优先顺序，并按此判断结果启动异常处理。

用优先级 0 ～ 16 表示中断源的优先顺序，优先级 0 为最低级，优先级 16 为最高级。 NMI 中断为不可屏蔽的最优先中断 （优先级 16），可随时接受。用户断点中断为优先级 15。可由 INTC 的中断优先级寄存器 A、 D ～ F、 H ～ M （IPRA、 IPRD ～ IPRF、 IPRH ～ IPRM）自由设定 IRQ 中断与内部外围模块中断的优先级 （表 5.8）。可设定优先级 0 ～ 15，但不得设定优先级 16。有关 IPRA、 IPRD ～ IPRF、 IPRH ～ IPRM，参阅 “6.3.4 中断优先级寄存器 A、 D ～ F、 H ～ M （IPRA、 IPRD ～ IPRF、 IPRH ～ IPRM）”。

表 5.8　中断优先顺序

种类	优先级	备　注
NMI	16	固定优先级，不可屏蔽
用户断点	15	固定优先级
IRQ	0 ～ 15	可由中断优先级设定寄存器 A、 D ～ F、 H ～ M （IPRA、 IPRD ～ IPRF、 IPRH ～ IPRM）设定
内部外围模块		

5.4.3 中断异常处理

产生中断时，由中断控制器（INTC）判断优先顺序。可随时接受 NMI 中断，但为其他中断时，只可在优先级高于状态寄存器（SR）的中断屏蔽位（I3 ～ I0）设定的优先级时，才可被接受。

如果接受中断就开始中断异常处理。中断异常处理时，CPU 将 SR 和程序计数器（PC）压栈，并对 SR 的 I3 ～ I0 位写入所接受中断优先级的值。但 NMI 的中断优先级为 16，而 I3 ～ I0 位所设定的值为 H'F（优先级 15）。从已接受中断所对应的异常处理向量表取出异常服务程序起始地址，并跳转至该地址开始执行程序。有关中断异常处理，详情参阅 “6.6 运行说明”。

5.5 由指令引起的异常

5.5.1 由指令引起的异常的种类

如表 5.9 所示，启动异常处理的指令有陷阱指令、槽非法指令及一般非法指令。

表 5.9 由指令引起的异常的种类

种类	异常源指令	备 注
陷阱指令	TRAPA	
槽非法指令 *	配置于延迟转移指令之后（延迟槽）的未定义代码或 PC 改写指令	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、LDC Rm,SR、LDC.L@Rm+,SR
一般非法指令 *	延迟槽以外的未定义代码	

【注】 * 解码除 H'F000 ～ H'FFFF 外的未定义指令时，无法保证运行。

5.5.2 陷阱指令

如果执行 TRAPA 指令，就开始陷阱指令异常处理。此时，CPU 如下运行：

1. 将状态寄存器（SR）压栈。
2. 将程序计数器（PC）压栈。保存的PC值为TRAPA指令的下一条指令的起始地址。
3. 从TRAPA指令指定的向量编号所对应的异常处理向量表取出异常服务程序起始地址，并跳转至该地址开始执行程序。此时的跳转不是延迟转移。

5.5.3 槽非法指令

将配置于延迟转移指令之后的指令称为“配置于延迟槽的指令”。配置于延迟槽的指令为未定义代码时，如果解码此未定义代码，就开始槽非法指令异常处理；配置于延迟槽的指令为 PC 改写指令时，如果解码此 PC 改写指令，就开始槽非法指令异常处理。执行槽非法指令异常处理时，CPU 如下运行：

1. 将SR压栈。
2. 将PC压栈。保存的PC值为未定义代码或PC改写指令前的延迟转移指令的地址。
3. 从产生异常所对应的异常处理向量表取出异常服务程序起始地址，并跳转至该地址开始执行程序。此时的跳转不是延迟转移。

5.5.4 一般非法指令

如果解码未配置于延迟转移指令之后（延迟槽）的未定义代码，就开始一般非法指令异常处理。此时，CPU 的运行步骤与槽非法指令异常处理的步骤相同。但，保存的 PC 值与槽非法指令异常处理时不同，为该未定义代码的起始地址。

5.6　异常处理的接受

如表 5.10 所示，延迟槽或紧接着中断禁止指令后产生除复位之外的其他异常源时，有可能不会立即接受，而是保留。此时，当可接受异常的指令被解码时，则可接受异常处理。

表 5.10　延迟槽、紧接着中断禁止指令后产生异常源

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

【符号说明】○：可接受。
×：不可接受。
—：不可能的情况。

【注】 *1　中断禁止指令：LDC、 LDC.L、 STC、 STC.L、 LDS、 LDS.L、 STS、 STS.L。
*2　在执行延迟转移指令前可接受。但，如果在 RTE 指令的延迟槽产生地址错误或槽非法指令，则无法保证运行。
*3　延迟转移后 （延迟槽指令与延迟转移目标指令之间）可接受。
*4　执行完中断禁止指令的下一条指令后 （紧接着中断禁止指令后的指令与该条指令的下一条指令之间）可接受。

5.7　异常处理之后的堆栈状态

异常处理之后的堆栈状态如表 5.11 所示。

表 5.11　异常处理结束之后的堆栈状态

种类	堆栈状态	种类	堆栈状态
地址错误 （产生异常的指令在延迟槽时）	SP → 延迟转移指令地址 （32位） SR （32位）	地址错误 （上述情况以外）	SP → 异常源的指令地址 （32位） SR （32位）
中断	SP → 已执行指令的下一条指令地址 （32位） SR （32位）	陷阱指令	SP → TRAPA指令的下一条指令地址 （32位） SR （32位）
槽非法指令	SP → 延迟转移指令地址 （32位） SR （32位）	一般非法指令	SP → 一般非法指令地址 （32位） SR （32位）

5.8 使用时的注意事项

5.8.1 堆栈指针 （SP）的值

必须设定 SP 的值为 4 的倍数。如果 SP 的值非 4 的倍数，则在异常处理的堆栈存取时，产生地址错误。

5.8.2 向量基址寄存器 （VBR）的值

必须设定 VBR 的值为 4 的倍数。如果 VBR 的值非 4 的倍数，则在异常处理的堆栈存取时，产生地址错误。

5.8.3 地址错误异常处理的堆栈存取时产生的地址错误

如果 SP 的值非 4 的倍数，则在异常处理 （中断等）的堆栈存取时产生地址错误，并在此异常处理结束之后，转移至地址错误异常处理。地址错误异常处理的堆栈存取时，也会产生地址错误。但为避免因无休止的地址错误异常处理导致持续堆栈存取，而不接受此时的地址错误。由此，可将程序的控制转移至地址错误异常服务程序，并执行错误处理。

如果异常处理堆栈存取时产生地址错误，则执行堆栈存取的总线周期（写入）。SR 与 PC 的堆栈存取过程中， SP 分别减 4，因此即使堆栈存取结束之后， SP 的值也不是 4 的倍数。另外，堆栈存取的输出地址值为 SP 值的低 2 位舍入为 0 的值。此时，堆栈存取之后的写入数据不定。

5.8.4 槽非法指令异常处理相关的注意事项

本 LSI 的槽非法指令异常处理与以往的 SH-2 有以下不同：

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

对此，本公司有以下对应软件产品：

(1) 编译器

V.4 后版本的编译器不在延迟槽配置该指令。

(2) μITRON 规格实时 OS

(a) HI7000/4、 HI-SH7

OS 的延迟槽不存在该指令。

(b) HI7000

OS 的延迟槽存在该指令，因此本 LSI 将会产生槽非法指令。

(c) 其他

由汇编程序描述或导入对象的中间件，本 LSI 有可能产生槽非法指令。

本公司网站的产品信息页已登载该指令串的检查程序 （校验表）。如果有需要，可下载网站的校验表加以确认。

第 6 章　　中断控制器 （INTC）

中断控制器 （INTC）判断中断源的优先顺序，并控制向 CPU 发出的中断请求。

6.1　　特点

- 可设定16级中断优先顺序
- NMI噪声消除器功能
- 可对外部输出已产生中断的通知 （$\overline{\text{IRQOUT}}$引脚）

INTC 框图如图 6.1 所示。

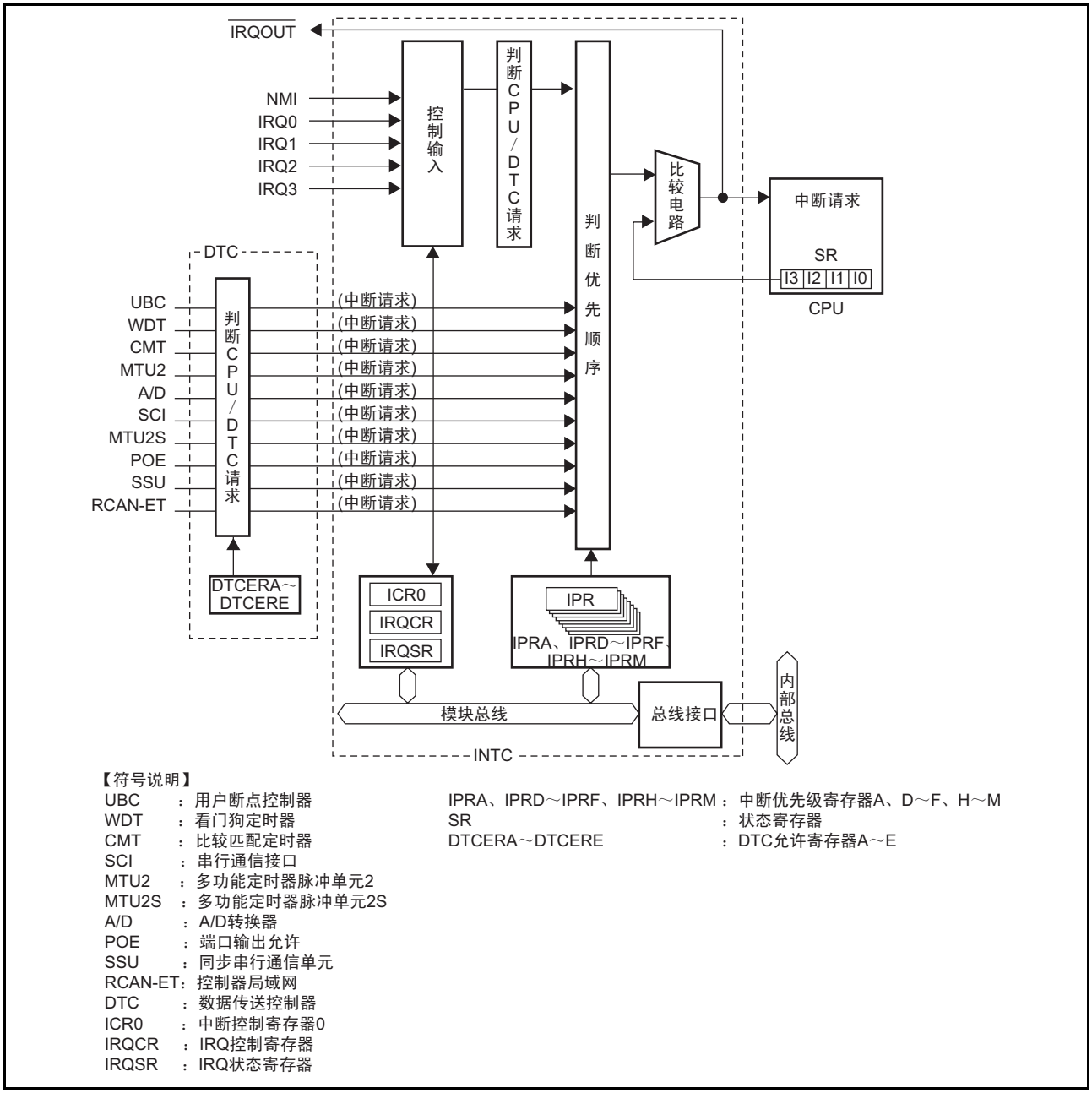


图 6.1　INTC 框图

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	引脚名称	输入 / 输出	功　能
非屏蔽中断输入引脚	NMI	输入	输入不可屏蔽的中断请求信号
中断请求输入引脚	IRQ0 ～ IRQ3	输入	输入可屏蔽的中断请求信号
中断请求输出引脚	$\overline{\text{IRQOUT}}$	输出	输出通知产生中断源的信号

6.3　寄存器说明

INTC 有以下寄存器。有关寄存器的地址及各处理状态的寄存器状态，详情参阅 “第 24 章 寄存器一览表”。

表 6.2　寄存器结构

寄存器名称	简称	R/W	初始值	地址	存取长度
中断控制寄存器 0	ICR0	R/W	H'x000	H'FFFFE900	8、16
IRQ 控制寄存器	IRQCR	R/W	H'0000	H'FFFFE902	8、16
IRQ 状态寄存器	IRQSR	R/W	H'Fx00	H'FFFFE904	8、16
中断优先级寄存器 A	IPRA	R/W	H'0000	H'FFFFE906	8、16
中断优先级寄存器 D	IPRD	R/W	H'0000	H'FFFFE982	16
中断优先级寄存器 E	IPRE	R/W	H'0000	H'FFFFE984	16
中断优先级寄存器 F	IPRF	R/W	H'0000	H'FFFFE986	16
中断优先级寄存器 H	IPRH	R/W	H'0000	H'FFFFE98A	16
中断优先级寄存器 I	IPRI	R/W	H'0000	H'FFFFE98C	16
中断优先级寄存器 J	IPRJ	R/W	H'0000	H'FFFFE98E	16
中断优先级寄存器 K	IPRK	R/W	H'0000	H'FFFFE990	16
中断优先级寄存器 L	IPRL	R/W	H'0000	H'FFFFE992	16
中断优先级寄存器 M	IPRM	R/W	H'0000	H'FFFFE994	16

6.3.1　中断控制寄存器 0（ICR0）

ICR0 为 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，并表示对 NMI 引脚的输入电平。

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

【注】* NMI引脚为高电平时，此位为1；低电平时，为0。

位	位名称	初始值	R/W	说　明
15	NMIL	*	R	NMI 输入电平 设定输入到 NMI 引脚的信号电平。通过读取本位，可得知 NMI 引脚的电平；写入无效。 0：对 NMI 引脚输入低电平 1：对 NMI 引脚输入高电平
14 ～ 9	—	均为 0	R	保留位 读取值、写入值总是为 0。
8	NMIE	0	R/W	选择 NMI 边沿 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	—	均为 0	R	保留位 读取值、写入值总是为 0。

6.3.2　IRQ 控制寄存器（IRQCR）

IRQCR 为 16 位寄存器，设定外部中断输入引脚 IRQ0 ～ IRQ3 的输入信号检测模式。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	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	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 读取值、写入值总是为 0。
7 6	IRQ31S IRQ30S	0 0	R/W R/W	选择 IRQ3 检测 选择在低电平、下降沿、上升沿或双边沿检测 IRQ3 引脚的中断信号。 00：在 IRQ3 输入的低电平检测中断请求 01：在 IRQ3 输入的下降沿检测中断请求 10：在 IRQ3 输入的上升沿检测中断请求 11：在 IRQ3 输入的双边沿检测中断请求
5 4	IRQ21S IRQ20S	0 0	R/W R/W	选择 IRQ2 检测 选择在低电平、下降沿、上升沿或双边沿检测 IRQ2 引脚的中断信号。 00：在 IRQ2 输入的低电平检测中断请求 01：在 IRQ2 输入的下降沿检测中断请求 10：在 IRQ2 输入的上升沿检测中断请求 11：在 IRQ2 输入的双边沿检测中断请求
3 2	IRQ11S IRQ10S	0 0	R/W R/W	选择 IRQ1 检测 选择在低电平、下降沿、上升沿或双边沿检测 IRQ1 引脚的中断信号。 00：在 IRQ1 输入的低电平检测中断请求 01：在 IRQ1 输入的下降沿检测中断请求 10：在 IRQ1 输入的上升沿检测中断请求 11：在 IRQ1 输入的双边沿检测中断请求
1 0	IRQ01S IRQ00S	0 0	R/W R/W	选择 IRQ0 检测 选择在低电平、下降沿、上升沿或双边沿检测 IRQ0 引脚的中断信号。 00：在 IRQ0 输入的低电平检测中断请求 01：在 IRQ0 输入的下降沿检测中断请求 10：在 IRQ0 输入的上升沿检测中断请求 11：在 IRQ0 输入的双边沿检测中断请求

6.3.3　IRQ 状态寄存器（IRQSR）

IRQSR 为 16 位寄存器，表示外部中断输入引脚 IRQ0 ～ IRQ3 的状态与中断请求的状态。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	IRQ3L	IRQ2L	IRQ1L	IRQ0L	-	-	-	-	IRQ3F	IRQ2F	IRQ1F	IRQ0F
初始值：	1	1	1	1	*	*	*	*	0	0	0	0	0	0	0	0
R/W：	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

【注】 * 对应的IRQ引脚为高电平时，此位为1；低电平时，为0。

位	位名称	初始值	R/W	说　明
15 ～ 12	—	均为 1	R	保留位 读取值、写入值总是为 1。
11	IRQ3L	*	R	表示 IRQ3 的引脚状态。 0：对 IRQ3 引脚输入低电平 1：对 IRQ3 引脚输入高电平
10	IRQ2L	*	R	表示 IRQ2 的引脚状态。 0：对 IRQ2 引脚输入低电平 1：对 IRQ2 引脚输入高电平
9	IRQ1L	*	R	表示 IRQ1 的引脚状态。 0：对 IRQ1 引脚输入低电平 1：对 IRQ1 引脚输入高电平
8	IRQ0L	*	R	表示 IRQ0 的引脚状态。 0：对 IRQ0 引脚输入低电平 1：对 IRQ0 引脚输入高电平
7 ～ 4	—	均为 0	R	保留位 读取值、写入值总是为 0。
3	IRQ3F	0	R/W	表示 IRQ3 中断请求状态。 设定电平检测时 0：无 IRQ3 中断请求 ［清除条件］ • 对 IRQ3 引脚输入高电平 1：有 IRQ3 中断请求 ［置位条件］ • 对 IRQ3 引脚输入低电平 设定边沿检测时 0：未检测到 IRQ3 中断请求 ［清除条件］ • 读取 IRQ3F=1 的状态后，写入 0 • 接受 IRQ3 中断 1：检测到 IRQ3 中断请求 ［置位条件］ • 检测 IRQ3 引脚的指定边沿

位	位名称	初始值	R/W	说　明
2	IRQ2F	0	R/W	表示 IRQ2 中断请求状态。 设定电平检测时 0：无 IRQ2 中断请求 ［清除条件］ • 对 IRQ2 引脚输入高电平 1：有 IRQ2 中断请求 ［置位条件］ • 对 IRQ2 引脚输入低电平 设定边沿检测时 0：未检测到 IRQ2 中断请求 ［清除条件］ • 读取 IRQ2F=1 的状态后，写入 0 • 接受 IRQ2 中断 1：检测到 IRQ2 中断请求 ［置位条件］ • 检测 IRQ2 引脚的指定边沿
1	IRQ1F	0	R/W	表示 IRQ1 中断请求状态。 设定电平检测时 0：无 IRQ1 中断请求 ［清除条件］ • 对 IRQ1 引脚输入高电平 1：有 IRQ1 中断请求 ［置位条件］ • 对 IRQ1 引脚输入低电平 设定边沿检测时 0：未检测到 IRQ1 中断请求 ［清除条件］ • 读取 IRQ1F=1 的状态后，写入 0 • 接受 IRQ1 中断 1：检测到 IRQ1 中断请求 ［置位条件］ • 检测 IRQ1 引脚的指定边沿
0	IRQ0F	0	R/W	表示 IRQ0 中断请求状态。 设定电平检测时 0：无 IRQ0 中断请求 ［清除条件］ • 对 IRQ0 引脚输入高电平 1：有 IRQ0 中断请求 ［置位条件］ • 对 IRQ0 引脚输入低电平 设定边沿检测时 0：未检测到 IRQ0 中断请求 ［清除条件］ • 读取 IRQ0F=1 的状态后，写入 0 • 接受 IRQ0 中断 1：检测到 IRQ0 中断请求 ［置位条件］ • 检测 IRQ0 引脚的指定边沿

【注】 * 对应的 IRQ 引脚为高电平时，此位为 1；低电平时，为 0。

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

IPR 为 10 个可读取 / 写入的 16 位寄存器，设定除 NMI 之外的中断源优先顺序（优先级 15 ～ 0）。各中断源与 IPR 的关系参照表 6.3。通过对 bit15 ～ 12、bit11 ～ 8、bit7 ～ 4 及 bit3 ～ 0 的各 4 位设定 H'0 到 H'F 范围的值，决定相应中断请求的优先顺序。未分配的保留位，必须设定 H'0（B'0000）。

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

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

【注】 位名称用普通名称表示，在寄存器一览表用模块名称表示。

6.4　中断源

6.4.1　外部中断源

中断源分 4 类，分别为用户断点、 NMI、 IRQ 及内部外围模块。用优先级的值（0 ～ 16）表示各中断的优先顺序， 0 为最低级， 16 为最高级。设定为 0 时，屏蔽该中断。

(1)　NMI 中断

NMI 中断为优先级 16 的中断，总是被接受。在边沿检测 NMI 引脚的输入，通过设定中断控制寄存器 0（ICR0）的 NMI 边沿选择位 （NMIE），可选择检测边沿为上升沿或下降沿。

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

(2)　IRQ3 ～ IRQ0 中断

由 IRQ0 ～ IRQ3 引脚的输入产生 IRQ 中断。通过设定 IRQ 控制寄存器 （IRQCR）的 IRQ 检测选择位（IRQ31S、 IRQ30S ～ IRQ01S、 IRQ00S），可对每个引脚选择低电平检测、下降沿检测、上升沿检测或双边沿检测。另外，通过中断优先级寄存器 A （IPRA），可在 0 ～ 15 范围内对每个引脚设定优先级。

IRQ 中断设定为低电平检测时， IRQ 引脚在低电平期间，对 INTC 发送中断请求信号； IRQ 引脚变为高电平时，不对 INTC 发送中断请求信号。可通过读取 IRQ 状态寄存器 （IRQSR）的 IRQ 标志 （IRQ3F ～ IRQ0F），确认有无中断请求。

IRQ 中断设定为下降沿 （上升沿、双边沿）检测时，可根据 IRQ 引脚从高电平变为低电平 （从低电平变为高电平、从低电平变为高电平或从高电平变为低电平）的变化，检测中断请求，并对 INTC 发送中断请求信号。保持 IRQ 中断请求的检测结果直到接受该中断请求为止。另外，可通过读取 IRQ 状态寄存器 （IRQSR）的 IRQ 标志 （IRQ3F ～ IRQ0F），确认是否检测到 IRQ 中断请求，并可通过读取 1 后写入 0，取消 IRQ 中断请求的检测结果。

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

IRQ3 ～ IRQ0 中断框图如图 6.2 所示。

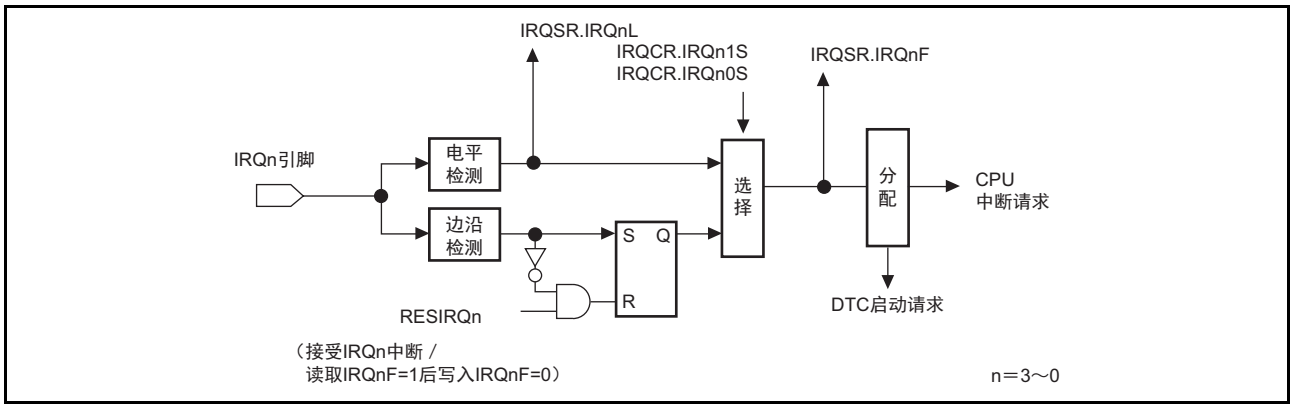


图 6.2　IRQ0 ～ IRQ3 中断控制

6.4.2　内部外围模块中断

内部外围模块中断为下述内部外围模块产生的中断。

对每个中断源分配不同的中断向量，因此无需通过异常服务程序判断中断源。通过中断优先级寄存器 D ～ F、 H ～ M （IPRD ～ IPRF、 IPRH ～ IPRM）可在优先级 0 ～ 15 的范围内对各模块设定优先顺序。内部外围模块中断异常处理时，将状态寄存器 （SR）的中断屏蔽位 （I3 ～ I0）设定为已接受的内部外围模块中断优先级的值。

6.4.3　用户断点中断

用户断点中断为用户断点控制器 （UBC）设定的断点条件成立时产生的中断，优先级为 15。在边沿检测用户断点中断请求，并保持到接受为止。用户断点异常处理时，将状态寄存器 （SR）的中断屏蔽位 （I3 ～ I0）设定为优先级 15。有关用户断点，详情参阅 “第 7 章 用户断点控制器 （UBC）”。

6.5　中断异常处理向量表

中断源、向量编号、向量表地址偏移量及中断优先顺序如表 6.3 所示。

分别对各中断源分配不同的向量编号与向量表地址偏移量。由该向量编号与向量表地址偏移量计算向量表地址。中断异常处理时，从该向量表地址所示的向量表中取出异常服务程序起始地址。向量表地址的计算方法参照表 5.4。

通过中断优先级寄存器 A、 D ～ F、 H ～ M （IPRA、 IPRD ～ IPRF、 IPRH ～ IPRM），可在优先级 0 ～ 15 范围内对各引脚或各模块任意设定 IRQ 中断与内部外围模块中断的优先顺序。但，相同 IPR 指定的多个中断源的优先顺序，则是向量编号越小所分配优先级越高，且不可更改。通过上电复位将 IRQ 中断与内部外围模块中断的优先顺序设定为优先级 0。设定多个中断源的优先顺序为相同优先级时，如果同时产生这些中断，则按表 6.3 所示的默认优先顺序执行处理。

表 6.3　中断源、向量地址及中断优先顺序一览表

中断产生源	名　称	向量编号	向量表起始地址	IPR	默认优先顺序
用户断点		12	H'00000030	—	高
外部引脚	NMI	11	H'0000002C	—	↑
	IRQ0	64	H'00000100	IPRA15 ～ IPRA12	
	IRQ1	65	H'00000104	IPRA11 ～ IPRA8	
	IRQ2	66	H'00000108	IPRA7 ～ IPRA4	
	IRQ3	67	H'0000010C	IPRA3 ～ IPRA0	
MTU2_0	TGIA_0	88	H'00000160	IPRD15 ～ IPRD12	
	TGIB_0	89	H'00000164		
	TGIC_0	90	H'00000168		
	TGID_0	91	H'0000016C		
	TCIV_0	92	H'00000170	IPRD11 ～ IPRD8	
	TGIE_0	93	H'00000174		
	TGIF_0	94	H'00000178		
MTU2_1	TGIA_1	96	H'00000180	IPRD7 ～ IPRD4	
	TGIB_1	97	H'00000184		
	TCIV_1	100	H'00000190	IPRD3 ～ IPRD0	
	TCIU_1	101	H'00000194		
MTU2_2	TGIA_2	104	H'000001A0	IPRE15 ～ IPRE12	
	TGIB_2	105	H'000001A4		
	TCIV_2	108	H'000001B0	IPRE11 ～ IPRE8	
	TCIU_2	109	H'000001B4		
MTU2_3	TGIA_3	112	H'000001C0	IPRE7 ～ IPRE4	
	TGIB_3	113	H'000001C4		
	TGIC_3	114	H'000001C8		
	TGID_3	115	H'000001CC		
	TCIV_3	116	H'000001D0	IPRE3 ～ IPRE0	
MTU2_4	TGIA_4	120	H'000001E0	IPRF15 ～ IPRF12	
	TGIB_4	121	H'000001E4		
	TGIC_4	122	H'000001E8		
	TGID_4	123	H'000001EC		
	TCIV_4	124	H'000001F0	IPRF11 ～ IPRF8	
POE （MTU2）	OEI1	132	H'00000210	IPRF3 ～ IPRF0	
	OEI3	133	H'00000214		
MTU2S_3	TGIA_3S	160	H'00000280	IPRH7 ～ IPRH4	
	TGIB_3S	161	H'00000284		
	TGIC_3S	162	H'00000288		
	TGID_3S	163	H'0000028C		↓
	TCIV_3S	164	H'00000290	IPRH3 ～ IPRH0	低

中断产生源	名　称	向量编号	向量表起始地址	IPR	默认优先顺序
MTU2S_4	TGIA_4S	168	H'000002A0	IPRI11 ～ IPRI8	高
	TGIB_4S	169	H'000002A4		↑
	TGIC_4S	170	H'000002A8		
	TGID_4S	171	H'000002AC		
	TCIV_4S	172	H'000002B0		
MTU2S_5	TGIU_5S	176	H'000002C0	IPRI7 ～ IPRI4	
	TGIV_5S	177	H'000002C4		
	TGIW_5S	178	H'000002C8		
POE （MTU2S）	OEI2	180	H'000002D0	IPRI3 ～ IPRI0	
CMT_0	CMI_0	184	H'000002E0	IPRJ15 ～ IPRJ12	
CMT_1	CMI_1	188	H'000002F0	IPRJ11 ～ IPRJ8	
WDT	ITI	196	H'00000310	IPRJ3 ～ IPRJ0	
A/D_0	ADI_3	208	H'00000340	IPRK7 ～ IPRK4	
A/D_1	ADI_4	212	H'00000350	IPRK3 ～ IPRK0	
SCI_0	ERI_0	216	H'00000360	IPRL15 ～ IPRL12	
	RXI_0	217	H'00000364		
	TXI_0	218	H'00000368		
	TEI_0	219	H'0000036C		
SCI_1	ERI_1	220	H'00000370	IPRL11 ～ IPRL8	
	RXI_1	221	H'00000374		
	TXI_1	222	H'00000378		
	TEI_1	223	H'0000037C		
SCI_2	ERI_2	224	H'00000380	IPRL7 ～ IPRL4	
	RXI_2	225	H'00000384		
	TXI_2	226	H'00000388		
	TEI_2	227	H'0000038C		
SSU	SSERI	232	H'000003A0	IPRM15 ～ IPRM12	
	SSRXI	233	H'000003A4		
	SSTXI	234	H'000003A8		
RCAN-ET_0	ERS_0	240	H'000003C0	IPRM7 ～ IPRM4	
	OVR_0	241	H'000003C4		
	RM0_0	242	H'000003C8		
	RM1_0				
	SLE_0	243	H'000003CC		
RCAN-ET_1*	ERS_1	244	H'000003D0	IPRM3 ～ IPRM0	
	OVR_1	245	H'000003D4		
	RM0_1	246	H'000003D8		
	RM1_1				
	SLE_1	247	H'000003DC		低

【注】 *　仅适用于 SH7142

6.6　运行说明

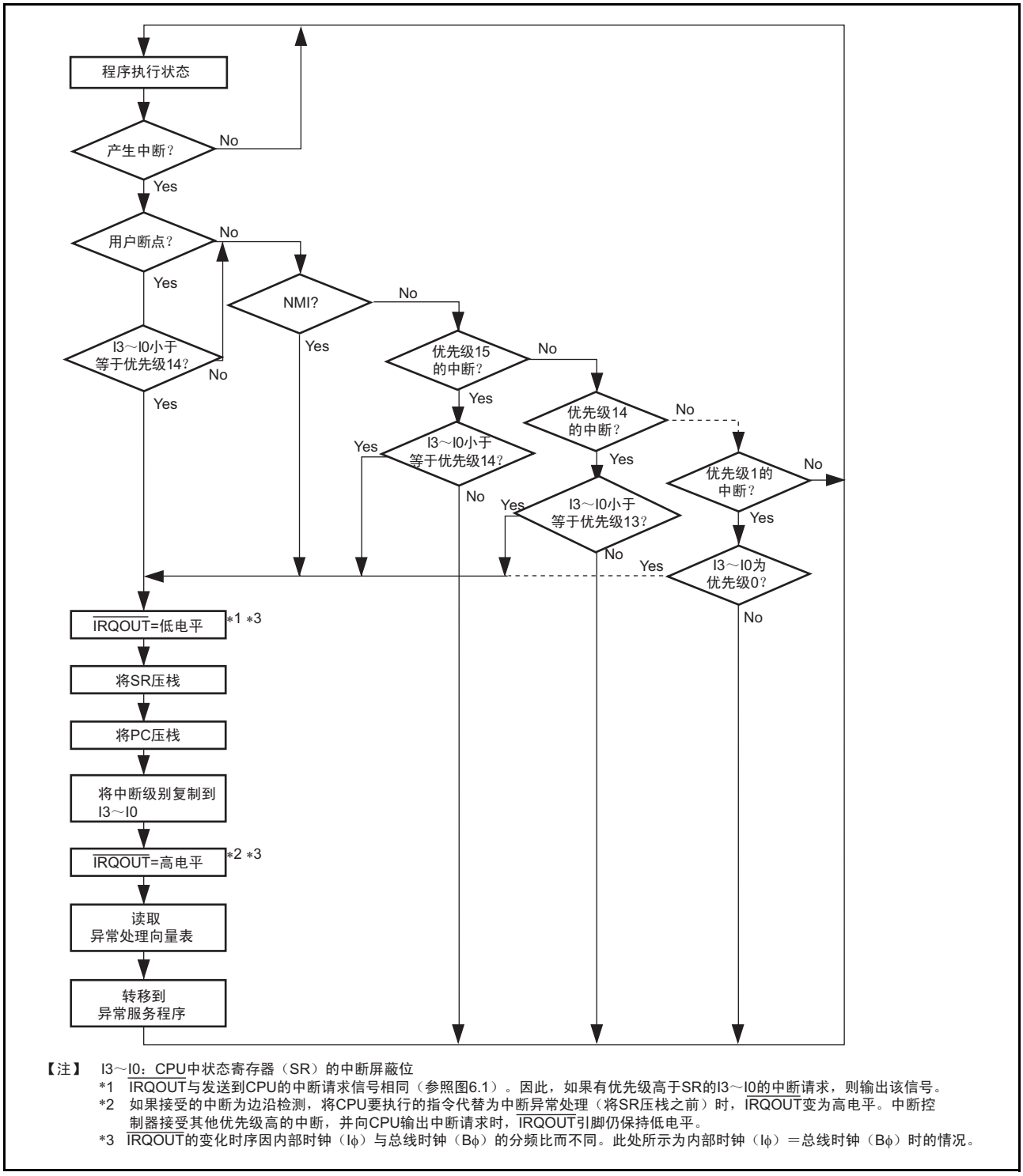
6.6.1　中断运行流程

产生中断时的运行流程说明如下，另外，运行流程如图 6.3 所示。

1. 从各中断请求源对中断控制器发送中断请求信号。
2. 中断控制器根据中断优先级寄存器A、D～F、H～M（IPRA、IPRD～IPRF、IPRH～IPRM），从送来的中断请求中选择优先顺序最高的中断，忽略*低于该优先顺序的中断。此时，如果同时产生多个设定为相同优先顺序的中断或相同模块内的中断，则按表6.3所示的默认优先顺序，选择优先顺序最高的中断。
3. 比较中断控制器所选中断的优先级和CPU状态寄存器 （SR）的中断屏蔽位 （I3～I0）。忽略优先级等于或低于I3～I0位所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，并对CPU发送中断请求信号。
4. 中断控制器接受中断时，从$\overline{\text{IRQOUT}}$引脚输出低电平。
5. 解码CPU要执行的指令时，检测中断控制器送来的中断请求，并由中断异常处理代替执行该指令。
6. 将SR与程序计数器 （PC）压栈。
7. 对SR的I3～I0位写入已接受中断的优先级。
8. 如果接受的中断为电平检测或内部外围模块产生的中断，则从$\overline{\text{IRQOUT}}$引脚输出高电平；如果接受中断为边沿检测，则在5.中由中断异常处理代替CPU要执行的指令时，从$\overline{\text{IRQOUT}}$引脚输出高电平。但，如果中断控制器接受其他中断 （其优先级高于正接受的中断），则$\overline{\text{IRQOUT}}$引脚保持低电平。
9. 从所接受中断对应的异常处理向量表取出异常服务程序起始地址，跳转至该地址后开始执行程序。此时的跳转不是延迟转移。

【注】 必须通过中断处理程序清除中断源标志。为了避免再次接受应清除的中断源，必须在清除后读取中断源标志，并确认已清除中断源标志后，执行 RTE 指令。

* 设定为边沿检测的中断请求保留到接受为止。如果为 IRQ 中断，则可通过存取 IRQ 状态寄存器 （IRQSR）取消。另外，通过上电复位或手动复位清除由边沿检测保留的中断。

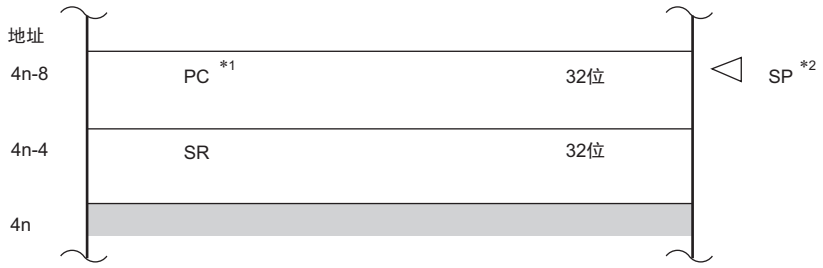


【注】 I3～I0：CPU中状态寄存器（SR）的中断屏蔽位
*1 IRQOUT与发送到CPU的中断请求信号相同（参照图6.1）。因此，如果有优先级高于SR的I3～I0的中断请求，则输出该信号。
*2 如果接受的中断为边沿检测，将CPU要执行的指令代替为中断异常处理（将SR压栈之前）时，IRQOUT变为高电平。中断控制器接受其他优先级高的中断，并向CPU输出中断请求时，IRQOUT引脚仍保持低电平。
*3 IRQOUT的变化时序因内部时钟（Iϕ）与总线时钟（Bϕ）的分频比而不同。此处所示为内部时钟（Iϕ）＝总线时钟（Bϕ）时的情况。

图 6.3　中断运行流程图

6.6.2　中断异常处理结束后的堆栈状态

中断异常处理结束后的堆栈状态如图 6.4 所示。



【注】 *1　PC：已执行指令的下一条指令（返回目标指令）的起始地址。
*2　SP必须设置为4的倍数。

图 6.4　中断异常处理结束后的堆栈状态

6.7　中断响应时间

从产生中断请求，到执行中断异常处理，并开始取异常服务程序起始指令为止的时间（中断响应时间）如表 6.4 所示。

表 6.4　中断响应时间

项　目		状态数			备　注
		NMI	IRQ	外围模块	
DTC 的启动判断		—	2×Bcyc	1×Pcyc	
优先顺序判断以及与 SR 屏蔽位的比较时间		1×Icyc+2×Pcyc	1×Icyc+1×Pcyc	1×Icyc+2×Pcyc	
到 CPU 正执行的顺序结束为止的等待时间		X（≥ 0）			最长顺序为中断异常处理或地址错误异常处理，X=7×Icyc+m1+m2+m3+m4。但继续执行屏蔽中断的指令时，可能会更长。
从开始中断异常处理，到开始取执行异常服务程序起始指令为止的时间		8×Icyc+m1+m2+m3			保存 SR、 PC 及取向量地址。
响应时间	合计	9×Icyc+2×Pcyc +m1+m2+m3+X	9×Icyc+1×Pcyc +2×Bcyc+m1+m2+m3+X	9×Icyc+3×Pcyc +m1+m2+m3+X	
	最短时 *	12×Icyc+2×Pcyc	12×Icyc+1×Pcyc+2×Bcyc	12×Icyc+3×Pcyc	SR、 PC 及向量表均在内部 RAM 时。
	最长时	16×Icyc+2×Pcyc +2(m1+m2+m3)+m4	16×Icyc+1×Pcyc +2×Bcyc+2(m1+m2+m3)+m4	16×Icyc+3×Pcyc +2(m1+m2+m3)+m4	

【注】 m1 ～ m4 为存取下列存储器时所需的状态数。

m1 ：保存 SR （长字写入）

m2 ：保存 PC （长字写入）

m3 ：读取向量地址 （长字读取）

m4 ：取中断服务程序起始指令

*　m1=m2=m3=m4=1×Icyc 时

6.8　通过中断请求信号传送数据

可通过中断请求信号，传送下列数据：

- 启动DTC， CPU中断取决于DTC的设定

相应的 DTCE 位为 1 时，中断源的 INTC 屏蔽 CPU 中断。 DTCE 清除条件与中断源标志清除条件如下所示：

DTCE 清除条件 =DTC 传送结束・DTCECLR

中断源标志清除条件 =DTC 传送结束・$\overline{\text{DTCECLR}}$

但， DTCECLR=DISEL+ 计数器 0

控制框图如图 6.5、图 6.6 所示。

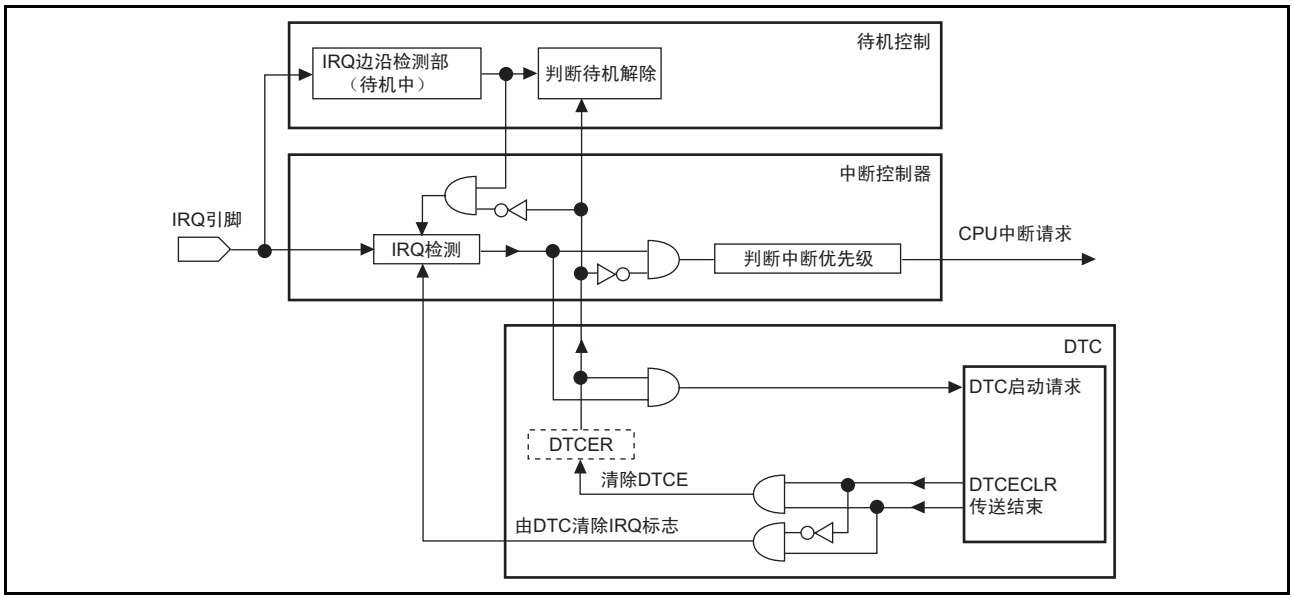


图 6.5　IRQ 中断控制框图

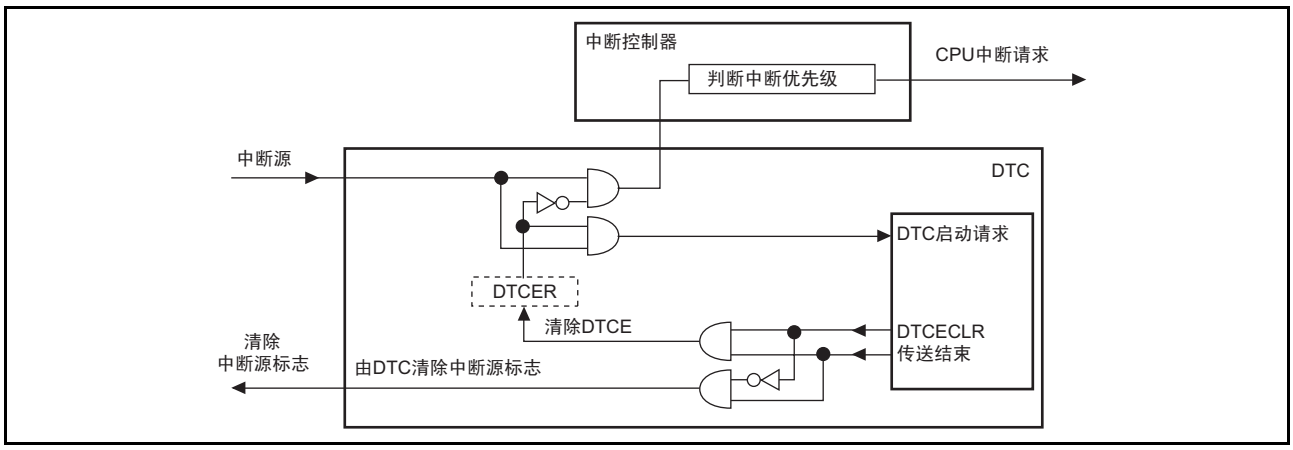


图 6.6　外围模块的中断控制框图

6.8.1 中断请求信号作为 DTC 启动源或 CPU 中断源时

1. 将对应DTC的DTCE位及DISEL位置1。
2. 产生中断时，对DTC提供启动源。
3. 传送数据时， DTC将DTCE位清0，并对CPU请求中断，不清除启动源。
4. CPU通过中断处理程序清除中断源，然后确认传送计数器的值。传送计数器的值≠0时，将DTCE位置1，并允许传送下一个数据。传送计数器的值=0时，则通过中断处理程序执行所需的结束处理。

6.8.2 中断请求信号作为 DTC 启动源，而非 CPU 中断源时

1. 将对应DTC的DTCE位置1， DISEL位清0。
2. 产生中断时，对DTC提供启动源。
3. 传送数据时， DTC清除启动源。保持DTCE位为1，因此不对CPU请求中断。
4. 但，传送计数器=0时，将DTCE位清0，并对CPU请求中断。
5. CPU通过中断处理程序执行所需的结束处理。

6.8.3 中断请求信号作为 CPU 中断源，而非 DTC 启动源时

1. 将对应DTC的DTCE位清0。
2. 产生中断时，对CPU请求中断。
3. CPU通过中断处理程序清除中断源，并执行所需处理。

6.9 使用时的注意事项

必须通过中断处理程序清除中断源标志。为了避免再次接受应清除的中断源，必须在清除后读取中断源标志，并确认已清除中断源标志后，执行 RTE 指令。

第 7 章 用户断点控制器 （UBC）

用户断点控制器 （UBC）提供简化程序调试的功能。使用此功能可容易作成自监控调试程序。即使不使用内部电路仿真器，也可通过本 LSI 调试程序。 UBC 可设定的断点条件有：取指令或读取 / 写入数据、数据长度、数据内容、地址值及取指令时的停止时序。

7.1 特点

1. 可设定下列断点比较条件：
   断点通道数：2个通道 （通道A和B）
   可将用户断点设定为通道A、 B独立或连续 （顺序）的条件。（顺序断点设定：通道A的断点条件匹配后，产生通道B断点条件匹配，并两者不在相同总线周期产生。）

- 地址
  可按位屏蔽地址32位的比较。
  可从L总线地址 （LAB）和I总线地址 （IAB）两者选一。
- 数据
  可屏蔽32位。
  可从L总线数据 （LDB）和I总线数据 （IDB）两者选一。
- 总线周期
  取指令或数据存取。
- 读取或写入
- 操作数长度
  支持字节、字及长字。

2. 可执行用户指定的用户断点中断异常处理程序。
3. 在取指令周期，可指定将用户断点设定在指令执行的前或后。
4. 作为断点条件 （仅对通道B），可指定的最大重复次数为 2^{12}–1。
5. 支持2组转移源/转移目标缓冲器。

UBC 框图如图 7.1 所示。

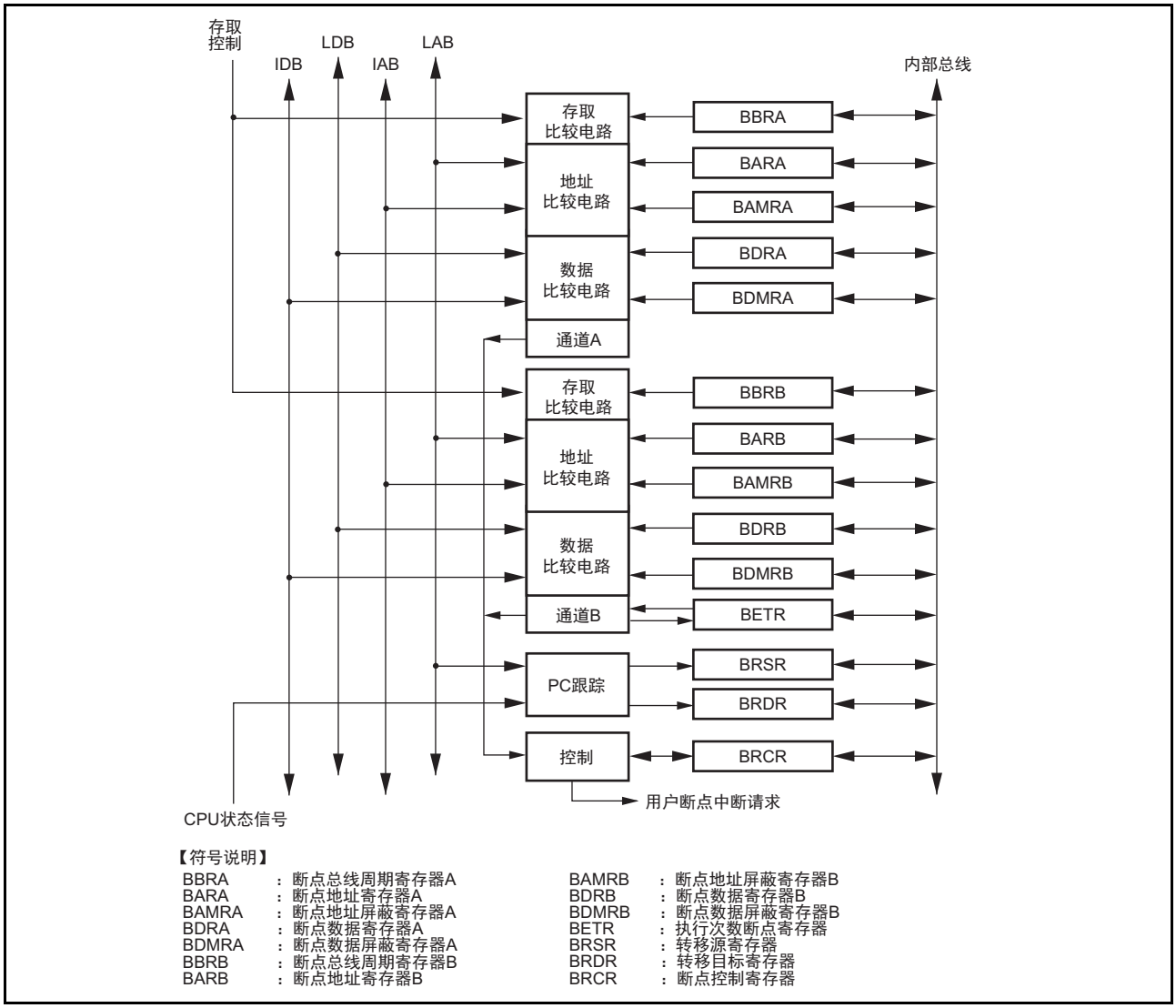


图 7.1　UBC 框图

7.2　输入 / 输出引脚

UBC 的引脚如表 7.1 所示。

表 7.1　引脚结构

名称	引脚名称	输入 / 输出	功　能
用户断点触发输出	$\overline{\text{UBCTRG}}$	输出	UBC 条件匹配的触发输出引脚。

7.3　寄存器说明

UBC 有以下寄存器。有关寄存器的地址及各处理状态的寄存器状态，详情参阅 “第 24 章 寄存器一览表”。

表 7.2　寄存器结构

寄存器名称	简称	R/W	初始值	地址	存取长度
断点地址寄存器 A	BARA	R/W	H'00000000	H'FFFFF300	32
断点地址屏蔽寄存器 A	BAMRA	R/W	H'00000000	H'FFFFF304	32
断点总线周期寄存器 A	BBRA	R/W	H'0000	H'FFFFF308	16
断点数据寄存器 A	BDRA	R/W	H'00000000	H'FFFFF310	32
断点数据屏蔽寄存器 A	BDMRA	R/W	H'00000000	H'FFFFF314	32
断点地址寄存器 B	BARB	R/W	H'00000000	H'FFFFF320	32
断点地址屏蔽寄存器 B	BAMRB	R/W	H'00000000	H'FFFFF324	32
断点总线周期寄存器 B	BBRB	R/W	H'0000	H'FFFFF328	16
断点数据寄存器 B	BDRB	R/W	H'00000000	H'FFFFF330	32
断点数据屏蔽寄存器 B	BDMRB	R/W	H'00000000	H'FFFFF334	32
断点控制寄存器	BRCR	R/W	H'00000000	H'FFFFF3C0	32
转移源寄存器	BRSR	R	H'0xxxxxxx	H'FFFFF3D0	32
转移目标寄存器	BRDR	R	H'0xxxxxxx	H'FFFFF3D4	32
执行次数断点寄存器	BETR	R/W	H'0000	H'FFFFF3DC	16

7.3.1　断点地址寄存器 A （BARA）

BARA 为可读取 / 写入的 32 位寄存器，指定作为通道 A 断点条件的地址。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAA31	BAA30	BAA29	BAA28	BAA27	BAA26	BAA25	BAA24	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

7.3.2　断点地址屏蔽寄存器 A （BAMRA）

BAMRA 为可读取 / 写入的 32 位寄存器，指定通过 BARA 指定的断点地址位中的屏蔽位。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAMA31	BAMA30	BAMA29	BAMA28	BAMA27	BAMA26	BAMA25	BAMA24	BAMA23	BAMA22	BAMA21	BAMA20	BAMA19	BAMA18	BAMA17	BAMA16
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAMA15	BAMA14	BAMA13	BAMA12	BAMA11	BAMA10	BAMA9	BAMA8	BAMA7	BAMA6	BAMA5	BAMA4	BAMA3	BAMA2	BAMA1	BAMA0
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说　明
31 ～ 0	BAMA31 ～ BAMA0	均为 0	R/W	断点地址屏蔽 A 指定通过 BARA （BAA31 ～ BAA0）指定的通道 A 断点地址位中的屏蔽位。 0：断点地址位 BAAn 包含在断点条件 1：断点地址位 BAAn 被屏蔽，不包含在断点条件 【注】　n=31 ～ 0

7.3.3　断点总线周期寄存器 A （BBRA）

BBRA 为可读取 / 写入的 16 位寄存器，可指定 （1） I 总线周期的总线主控器、（2） L 总线周期或 I 总线周期、（3）取指令或数据存取、（4）读取或写入、（5）操作数长度等作为通道 A 的断点条件。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	CPA[2:0]			CDA[1:0]		IDA[1:0]		RWA[1:0]		SZA[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/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 读取值、写入值总是为 0。
10 ～ 8	CPA[2:0]	000	R/W	I 总线的总线主控器选择 A 选择 I 总线作为通道 A 断点条件的总线周期时，选择总线主控器；选择 L 总线作为总线周期时，本位无效。 000：不执行条件比较 xx1：断点条件包含 CPU 周期 x1x：禁止设定 1xx：断点条件包含 DTC 周期
7、6	CDA[1:0]	00	R/W	L 总线周期 /I 总线周期选择 A 选择 L 总线周期或 I 总线周期作为通道 A 断点条件的总线周期。 00：不执行条件比较 01：断点条件为 L 总线周期 10：断点条件为 I 总线周期 11：断点条件为 L 总线周期
5、4	IDA[1:0]	00	R/W	取指令 / 数据存取选择 A 选择取指令周期或数据存取周期作为通道 A 断点条件的总线周期。 00：不执行条件比较 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数据存取周期
3、2	RWA[1:0]	00	R/W	读取 / 写入选择 A 选择读取周期或写入周期作为通道 A 断点条件的总线周期。 00：不执行条件比较 01：断点条件为读取周期 10：断点条件为写入周期 11：断点条件为读取周期或写入周期
1、0	SZA[1:0]	00	R/W	操作数长度选择 A 选择操作数长度作为通道 A 断点条件的总线周期。 00：断点条件不包含操作数长度 01：断点条件为字节存取 10：断点条件为字存取 11：断点条件为长字存取 【注】指定操作数长度时，必须使地址边界与操作数长度匹配。

【符号说明】 x ：Don't care

7.3.4　断点数据寄存器 A （BDRA）

BDRA 为可读取 / 写入的 32 位寄存器。作为断点条件 A 的对象数据总线有两种，通过断点总线周期寄存器 A （BBRA）的控制位 CDA1、 CDA0 选择。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BDA31	BDA30	BDA29	BDA28	BDA27	BDA26	BDA25	BDA24	BDA23	BDA22	BDA21	BDA20	BDA19	BDA18	BDA17	BDA16
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BDA15	BDA14	BDA13	BDA12	BDA11	BDA10	BDA9	BDA8	BDA7	BDA6	BDA5	BDA4	BDA3	BDA2	BDA1	BDA0
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说　明
31 ～ 0	BDA31 ～ BDA0	均为 0	R/W	断点数据位 A 保存指定通道 A 断点条件的数据。 通过 BBRA 选择 I 总线时，对 BDA31 ～ BDA0 指定 IDB 的断点数据。 通过 BBRA 选择 L 总线时，对 BDA31 ～ BDA0 指定 LDB 的断点数据。

【注】 1.　如果断点条件包含数据总线的值，则必须指定操作数长度。

2.　指定字节长度作为断点条件时，必须对 bit15 ～ 8 和 bit7 ～ 0 设置相同的字节数据，作为 BDRA 的断点数据。

7.3.5　断点数据屏蔽寄存器 A （BDMRA）

BDMRA 为可读取 / 写入的 32 位寄存器，指定通过 BDRA 指定的断点数据位中的屏蔽位。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BDMA31	BDMA30	BDMA29	BDMA28	BDMA27	BDMA26	BDMA25	BDMA24	BDMA23	BDMA22	BDMA21	BDMA20	BDMA19	BDMA18	BDMA17	BDMA16
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BDMA15	BDMA14	BDMA13	BDMA12	BDMA11	BDMA10	BDMA9	BDMA8	BDMA7	BDMA6	BDMA5	BDMA4	BDMA3	BDMA2	BDMA1	BDMA0
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说　明
31 ～ 0	BDMA31 ～ BDMA0	均为 0	R/W	断点数据屏蔽 A 指定通过 BARA （BDA31 ～ BDA0）指定的通道 A 断点地址位中的屏蔽位。 0：断点数据位 BDAn 包含在断点条件 1：断点数据位 BDAn 被屏蔽，不包含在断点条件 【注】　n=31 ～ 0

【注】 1.　如果断点条件包含数据总线的值，则必须指定操作数长度。

2.　指定字节长度作为断点条件时，必须对 bit15 ～ 8 和 bit7 ～ 0 设置相同的字节数据，作为 BDMRA 的断点屏蔽数据。

7.3.6　断点地址寄存器 B （BARB）

BARB 为可读取 / 写入的 32 位寄存器，指定作为通道 B 断点条件的地址。作为断点条件 B 的对象地址总线有两种，通过断点总线周期寄存器 B （BBRB）的控制位 CDB1、 CDB0 选择。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAB31	BAB30	BAB29	BAB28	BAB27	BAB26	BAB25	BAB24	BAB23	BAB22	BAB21	BAB20	BAB19	BAB18	BAB17	BAB16
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAB15	BAB14	BAB13	BAB12	BAB11	BAB10	BAB9	BAB8	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1	BAB0
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

7.3.7　断点地址屏蔽寄存器 B （BAMRB）

BAMRB 为可读取 / 写入的 32 位寄存器，指定通过 BARB 指定的断点地址位中的屏蔽位。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BAMB31	BAMB30	BAMB29	BAMB28	BAMB27	BAMB26	BAMB25	BAMB24	BAMB23	BAMB22	BAMB21	BAMB20	BAMB19	BAMB18	BAMB17	BAMB16
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAMB15	BAMB14	BAMB13	BAMB12	BAMB11	BAMB10	BAMB9	BAMB8	BAMB7	BAMB6	BAMB5	BAMB4	BAMB3	BAMB2	BAMB1	BAMB0
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说　明
31 ～ 0	BAMB31 ～ BAMB0	均为 0	R/W	断点地址屏蔽 B 指定通过 BARB （BAB31 ～ BAB0）指定的通道 B 断点地址位中的屏蔽位。 0：断点地址位 BABn 包含在断点条件 1：断点地址位 BABn 被屏蔽，并不包含在断点条件 【注】　n=31 ～ 0

7.3.8　断点数据寄存器 B （BDRB）

BDRB 为可读取 / 写入的 32 位寄存器。作为断点条件 B 的对象数据总线有两种，通过断点总线周期寄存器 B （BBRB）的控制位 CDB1、 CDB0 选择。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BDB31	BDB30	BDB29	BDB28	BDB27	BDB26	BDB25	BDB24	BDB23	BDB22	BDB21	BDB20	BDB19	BDB18	BDB17	BDB16
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BDB15	BDB14	BDB13	BDB12	BDB11	BDB10	BDB9	BDB8	BDB7	BDB6	BDB5	BDB4	BDB3	BDB2	BDB1	BDB0
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说　明
31 ～ 0	BDB31 ～ BDB0	均为 0	R/W	断点数据位 B 保存指定通道 B 断点条件的数据。 通过 BBRB 选择 I 总线时，对 BDB31 ～ BDB0 指定 IDB 的断点数据。 通过 BBRB 选择 L 总线时，对 BDB31 ～ BDB0 指定 LDB 的断点数据。

【注】 1.　如果断点条件包含数据总线的值，则必须指定操作数长度。

2.　指定字节长度作为断点条件时，必须对 bit15 ～ 8 和 bit7 ～ 0 设置相同的字节数据，作为 BDRB 的断点数据。

7.3.9　断点数据屏蔽寄存器 B （BDMRB）

BDMRB 为可读取 / 写入的 32 位寄存器，指定通过 BDRB 指定的断点数据位中的屏蔽位。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BDMB31	BDMB30	BDMB29	BDMB28	BDMB27	BDMB26	BDMB25	BDMB24	BDMB23	BDMB22	BDMB21	BDMB20	BDMB19	BDMB18	BDMB17	BDMB16
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BDMB15	BDMB14	BDMB13	BDMB12	BDMB11	BDMB10	BDMB9	BDMB8	BDMB7	BDMB6	BDMB5	BDMB4	BDMB3	BDMB2	BDMB1	BDMB0
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说　明
31 ～ 0	BDMB31 ～ BDMB0	均为 0	R/W	断点数据屏蔽 B 指定通过 BDRB （BDB31 ～ BDB0）指定的通道 B 断点数据位中的屏蔽位。 0：断点数据位 BDBn 包含在断点条件 1：断点数据位 BDBn 被屏蔽，并不包含在断点条件 【注】　n=31 ～ 0

【注】 1.　如果断点条件包含数据总线的值，则必须指定操作数长度。

2.　指定字节长度作为断点条件时，必须对 bit15 ～ 8 和 bit7 ～ 0 设置相同的字节数据，作为 BDMRB 的断点屏蔽数据。

7.3.10　断点总线周期寄存器 B （BBRB）

BBRB 为可读取 / 写入的 16 位寄存器，可指定 （1） I 总线周期的总线主控器、（2） L 总线周期或 I 总线周期、（3）取指令或数据存取、（4）读取或写入、（5）操作数长度等作为通道 B 的断点条件。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	CPB[2:0]			CDB[1:0]		IDB[1:0]		RWB[1:0]		SZB[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/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 读取值、写入值总是为 0。
10 ～ 8	CPB[2:0]	000	R/W	I 总线的总线主控器选择 B 选择 I 总线作为通道 B 断点条件的总线周期时，选择总线主控器。选择 L 总线作为总线周期时，本位无效。 000：不执行条件比较 xx1：断点条件包含 CPU 周期 x1x：禁止设定 1xx：断点条件包含 DTC 周期
7、6	CDB[1:0]	00	R/W	L 总线周期 /I 总线周期选择 B 选择 L 总线周期或 I 总线周期作为通道 B 断点条件的总线周期。 00：不执行条件比较 01：断点条件为 L 总线周期 10：断点条件为 I 总线周期 11：断点条件为 L 总线周期
5、4	IDB[1:0]	00	R/W	取指令 / 数据存取选择 B 选择取指令周期或数据存取周期作为通道 B 断点条件的总线周期。 00：不执行条件比较 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数据存取周期
3、2	RWB[1:0]	00	R/W	读取 / 写入选择 B 选择读取周期或写入周期作为通道 B 断点条件的总线周期。 00：不执行条件比较 01：断点条件为读取周期 10：断点条件为写入周期 11：断点条件为读取周期或写入周期
1、0	SZB[1:0]	00	R/W	操作数长度选择 B 选择操作数长度作为通道 B 断点条件的总线周期。 00：断点条件不包含操作数长度 01：断点条件为字节存取 10：断点条件为字存取 11：断点条件为长字存取 【注】指定操作数长度时，必须使地址边界与操作数长度匹配。

【符号说明】 x ：Don't care

7.3.11　断点控制寄存器 （BRCR）

BRCR 设定下列条件：

1. 指定通道A、 B作为2个独立的通道条件还是1个连续的条件使用。
2. 指定用户断点设定在指令执行前或指令执行后。
3. 指定通道B比较条件中是否包含执行次数。
4. 指定通道A、 B比较条件中是否包含数据总线的值。
5. 设置允许PC跟踪。
6. 选择$\overline{\text{UBCTRG}}$输出的脉宽。
7. 指定通道A、 B比较条件匹配时，是否请求用户断点中断。

BRCR 为可读取 / 写入的 32 位寄存器，有设置断点条件匹配标志和各种断点条件的位。

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

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCM FCA	SCM FCB	SCM FDA	SCM FDB	PCTE	PCBA	-	-	DBEA	PCBB	DBEB	-	SEQ	-	-	ETBE
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R	R/W	R	R	R/W

位	位名称	初始值	R/W	说　明
31 ～ 22	—	均为 0	R	保留位 读取值、写入值总是为 0。
21、 20	UTRGW[1:0]	00	R/W	选择 $\overline{\text{UBCTRG}}$ 输出脉宽 选择断点条件匹配时的 $\overline{\text{UBCTRG}}$ 输出脉宽。 00：禁止设定 01：将 $\overline{\text{UBCTRG}}$ 输出脉宽设置为 3 ～ $4t_{Bcyc}$ 10：将 $\overline{\text{UBCTRG}}$ 输出脉宽设置为 7 ～ $8t_{Bcyc}$ 11：将 $\overline{\text{UBCTRG}}$ 输出脉宽设置为 15 ～ $16t_{Bcyc}$ 【注】 t_{Bcyc} 表示外部总线时钟 （Bϕ=CK）的周期。
19	UBIDB	0	R/W	用户断点禁止 B 满足通道 B 断点条件时，选择是否禁止用户断点中断请求。 0：满足断点条件时，允许用户断点中断请求 1：满足断点条件时，禁止用户断点中断请求
18	—	0	R	保留位 读取值、写入值总是为 0。
17	UBIDA	0	R/W	用户断点禁止 A 满足通道 A 断点条件时，选择是否禁止用户断点中断请求。 0：满足断点条件时，允许用户断点中断请求 1：满足断点条件时，禁止用户断点中断请求
16	—	0	R	保留位 读取值、写入值总是为 0。
15	SCMFCA	0	R/W	L 总线周期条件匹配标志 A 满足通道 A 中设置的断点条件的 L 总线周期条件时，此标志置 1。对该位写入 0，清除此标志。 0：通道 A 的 L 总线周期条件不匹配 1：通道 A 的 L 总线周期条件匹配

位	位名称	初始值	R/W	说　明
14	SCMFCB	0	R/W	L 总线周期条件匹配标志 B 满足通道 B 中设置的断点条件的 L 总线周期条件时，将此标志置 1。对该位写入 0，清除此标志。 0：通道 B 的 L 总线周期条件不匹配 1：通道 B 的 L 总线周期条件匹配
13	SCMFDA	0	R/W	I 总线周期条件匹配标志 A 满足通道 A 中设置的断点条件的 I 总线周期条件时，此标志置 1。对该位写入 0，清除此标志。 0：通道 A 的 I 总线周期条件不匹配 1：通道 A 的 I 总线周期条件匹配
12	SCMFDB	0	R/W	I 总线周期条件匹配标志 B 满足通道 B 中设置的断点条件的 I 总线周期条件时，此标志置 1。对该位写入 0，清除此标志。 0：通道 B 的 I 总线周期条件不匹配 1：通道 B 的 I 总线周期条件匹配
11	PCTE	0	R/W	PC 跟踪允许 0：禁止 PC 跟踪 1：允许 PC 跟踪
10	PCBA	0	R/W	PC 断点选择 A 选择通道 A 取指令周期的断点时序在指令执行前或后。 0：通道 A 的 PC 断点设定在指令执行前 1：通道 A 的 PC 断点设定在指令执行后
9、8	—	均为 0	R	保留位 读取值、写入值总是为 0。
7	DBEA	0	R/W	数据断点允许 A 选择通道 A 断点条件中是否包含数据总线的值。 0：通道 A 断点条件中不包含数据总线的值 1：通道 A 断点条件中包含数据总线的值
6	PCBB	0	R/W	PC 断点选择 B 选择通道 B 取指令周期的断点时序在指令执行前或后。 0：通道 B 的 PC 断点设定在指令执行前 1：通道 B 的 PC 断点设定在指令执行后
5	DBEB	0	R/W	数据断点允许 B 选择通道 B 断点条件中是否包含数据总线的值。 0：通道 B 断点条件中不包含数据总线的值 1：通道 B 断点条件中包含数据总线的值
4	—	0	R	保留位 读取值、写入值总是为 0。
3	SEQ	0	R/W	选择顺序条件 选择通道 A 和 B 为 2 个独立条件或 1 个连续条件。 0：在独立条件下，比较通道 A 和通道 B 1：在连续条件下，比较通道 A 和通道 B （先通道 A、再通道 B）
2、1	—	均为 0	R	保留位 读取值、写入值总是为 0。
0	ETBE	0	R/W	执行次数断点允许 仅对通道 B 设置执行次数断点条件有效。该位为 1，且产生断点条件的次数与 BETR 寄存器指定的执行次数相等时，请求用户断点中断。 0：通道 B 的执行次数断点条件设置为无效 1：通道 B 的执行次数断点条件设置为有效

【注】 断点条件包含数据总线的值，并通过 RWA 位或 RWB 位指定写入周期为断点条件时，必须指定操作数长度。

7.3.12　执行次数断点寄存器 （BETR）

BETR 为可读取 / 写入的 16 位寄存器。通道 B 的执行次数断点条件设置为有效时，该寄存器指定执行断点的次数。执行次数最多为 2^{12} － 1 次。每满足一次断点条件， BETR 就递减 1。 BETR 变为 H'0001 后，如果满足断点条件，则请求用户断点中断。

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

位	位名称	初始值	R/W	说　明
15 ～ 12	—	均为 0	R	保留位 读取值、写入值总是为 0。
11 ～ 0	BET[11:0]	均为 0	R/W	执行次数

7.3.13　转移源寄存器 （BRSR）

BRSR 为 32 位只读寄存器，保存转移源指令地址的 bit27 ～ 0。BRSR 有转移时置 1 的标志位。读取 BRSR 时、从 PC 跟踪禁止状态设定为允许时、上电复位或手动复位时，此标志位清 0。通过复位，不初始化其他位。 2 个 BRSR 寄存器为队列结构，保存的寄存器每转移一次就产生移位。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SVF	-	-	-	BSA27	BSA26	BSA25	BSA24	BSA23	BSA22	BSA21	BSA20	BSA19	BSA18	BSA17	BSA16
初始值：	0	0	0	0	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W：	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BSA15	BSA14	BSA13	BSA12	BSA11	BSA10	BSA9	BSA8	BSA7	BSA6	BSA5	BSA4	BSA3	BSA2	BSA1	BSA0
初始值：	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W：	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名称	初始值	R/W	说　明
31	SVF	0	R	BRSR 有效标志 表示是否保存转移源地址。产生转移时，此标志置 1。读取 BRSR 时、从 PC 跟踪禁止状态设定为允许时或上电复位等任意一个条件下，此标志清 0。 0：BRSR 寄存器的值无效 1：BRSR 寄存器的值有效
30 ～ 28	—	均为 0	R	保留位 读取值、写入值总是为 0。
27 ～ 0	BSA27 ～ BSA0	不定	R	转移源地址 这些位保存转移源地址的 bit27 ～ 0。

7.3.14　转移目标寄存器 （BRDR）

BRDR 为 32 位只读寄存器，保存转移目标指令地址的 bit27 ～ 0。 BRDR 有转移时置 1 的标志位。读取 BRDR 时、从 PC 跟踪禁止状态设定为允许时、上电复位或手动复位时，此标志位清 0。通过复位，不初始化其他位。 2 个 BRDR 寄存器为队列结构，保存的寄存器每转移一次就产生移位。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DVF	-	-	-	BDA27	BDA26	BDA25	BDA24	BDA23	BDA22	BDA21	BDA20	BDA19	BDA18	BDA17	BDA16
初始值：	0	0	0	0	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W：	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BDA15	BDA14	BDA13	BDA12	BDA11	BDA10	BDA9	BDA8	BDA7	BDA6	BDA5	BDA4	BDA3	BDA2	BDA1	BDA0
初始值：	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W：	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名称	初始值	R/W	说　明
31	DVF	0	R	BRDR 有效标志 表示是否保存转移目标地址。产生转移时，此标志置 1。读取 BRDR 时、从 PC 跟踪禁止状态设定为允许时或上电复位等任意一个条件下，此标志位清 0。 0：BRDR 寄存器的值无效 1：BRDR 寄存器的值有效
30 ～ 28	—	均为 0	R	保留位 读取值、写入值总是为 0。
27 ～ 0	BDA27 ～ BDA0	不定	R	转移目标地址 这些位保存转移目标地址的 bit27 ～ 0。

7.4　运行说明

7.4.1　用户断点运行流程

从设定断点条件到用户断点异常处理的运行流程如下：

1. 在断点地址寄存器 （BARA或BARB）设置断点地址，在断点地址屏蔽寄存器 （BAMRA或BAMRB）设置要屏蔽的地址，在断点数据寄存器 （BDRA或BDRB）设置断点数据，在断点数据屏蔽寄存器 （BDMRA或BDMRB）设置要屏蔽的数据，在断点总线周期寄存器 （BBRA或BBRB）设置总线断点条件。 BBRA或BBRB的3对控制位，即L总线周期或I总线周期选择、取指令或数据存取选择、读取或写入选择3对控制位中的任意一对为B'00时，不产生用户断点。在BRCR的位设置断点控制。必须在设定其他所有断点相关寄存器后，设定BBRA或BBRB。
2. 满足断点条件时， UBC向CPU通知用户断点中断请求的同时，将各通道的L总线条件匹配标志（SCMFCA或SCMFCB）及I总线条件匹配标志 （SCMFDA或SCMFDB）分别置位。
3. 可使用相应条件匹配标志 （SCMFCA、 SCMFDA、 SCMFCB、 SCMFDB）检查设定条件是否匹配。条件匹配时，标志置位，但不能复位。为了能再次使用标志，必须先写入0，复位标志。
4. 设定在通道A和通道B的断点条件匹配可能同时产生。即使向CPU通知的用户断点中断请求只有1个，这2个条件匹配标志也可能都被置位。
5. 选择I总线作为断点条件时，必须注意下列事项：

- 将CPU与DTC连接至I总线。 UBC监视由BBRA寄存器的CPA2～CPA0位及BBRB寄存器的CPB2～CPB0位选择的总线主控器所生成的总线周期，并执行条件匹配比较。
- 将CPU的L总线上取指令引起的I总线周期 （包含读取填充周期）定义为I总线取指令周期，其他定义为数据存取周期。
- DTC发行的I总线周期仅为数据存取周期。
- 将I总线设定为断点条件时，即使在CPU执行的指令引起的I总线周期中条件匹配，也不能决定由哪条指令接受用户断点。

7.4.2　取指令周期的用户断点

1. 如果在断点总线周期寄存器 （BBRA或BBRB）设定 “不包含L总线/取指令/读取/字、长字或操作数长度”，则断点条件为L总线的取指令。可由相应通道的断点控制寄存器 （BRCR）的PCBA或PCBB位，选择在执行指令前或执行后产生用户断点。设定取指令周期作为断点条件时，必须将断点地址寄存器 （BARA或BARB）的LSB清0。如果此位置1，则不产生用户断点。
2. 将取指令产生的用户断点设定在指令执行前时，如果断点条件匹配，则取指令，并在确定要执行指令时，产生用户断点。因此，此功能不能用于因溢出 （转移或中断转移时被取出却没有被执行的指令）而取出的指令。如果对延迟转移指令的延迟槽设定此类断点条件，则在执行延迟转移指令前产生用户断点。

【注】 如果延迟转移指令未转移，则不将后续指令视为延迟槽。

3. 设定断点条件为在指令执行之后产生用户断点时，执行与断点条件匹配的指令，并在执行下一条指令之前，产生用户断点。和指令执行前的用户断点一样，此功能不用于因溢出而取出的指令。如果对延迟转移指令及其延迟槽设定此类断点条件，则不会在转移目标的第一条指令前产生用户断点。
4. 设定取指令周期时，忽略断点数据寄存器 （BDRA或BDRB）。因此，不可在取指令周期的用户断点设定断点数据。
5. 如果在取指令周期的用户断点设定I总线，则对I总线的取指令周期执行条件判断，详情参阅 “7.4.1 用户断点运行流程”的5.项。

7.4.3　数据存取周期的用户断点

1. 在数据存取断点，指定L总线作为断点条件时，对已执行指令的存取地址 （及数据）执行条件比较，并产生用户断点。指定I总线作为断点条件时，则对包含I总线上的CPU在内的所有总线主控器发行的数据存取周期地址 （及数据），执行条件比较，并产生用户断点。关于I总线上发行的CPU总线周期，详情参阅 “7.4.1 用户断点运行流程”的5. 项。
2. 数据存取周期地址及各操作数长度的比较条件关系如表7.3所示。

表 7.3　数据存取周期地址及操作数长度的比较条件

存取长度	比较地址
长字	比较断点地址寄存器的 bit31 ～ 2 和地址总线的 bit 31 ～ 2
字	比较断点地址寄存器的 bit31 ～ 1 和地址总线的 bit 31 ～ 1
字节	比较断点地址寄存器的 bit31 ～ 0 和地址总线的 bit 31 ～ 0

假设对断点地址寄存器 （BARA/BARB）设定地址H'00001003，此时，满足断点条件的总线周期（假设满足其他所有条件）包含下列含义：

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

3. 断点条件包含数据时
断点条件包含数据时，对断点总线周期寄存器 （BBRA或BBRB）指定长字、字或字节作为操作数长度。数据包含在断点条件时，如果地址条件与数据条件匹配，则产生用户断点。此时，为了指定字节数据，需对断点数据寄存器 （BDRA或BDRB）与断点数据屏蔽寄存器 （BDMRA或BDMRB）的bit15～8和bit7～0的2个字节设置相同的数据。选择字或字节作为操作数长度时，忽略BDRA或BDRB与BDMRA或BDMRB的bit31～16。
4. 选择L总线时，结束条件匹配指令的执行，并在执行下一条指令前产生用户断点。但，断点条件包含数据时，可能会在条件匹配指令的下一条指令执行结束后，产生用户断点。选择I总线时，不得指定产生用户断点的指令。另外，如果在延迟转移指令及其延迟槽产生此类用户断点，则不会在转移目标的第一条指令前产生用户断点。

7.4.4 顺序断点

1. 将BRCR的SEQ位置1时，在通道A断点条件匹配后，如果通道B断点条件匹配，则产生顺序断点。在通道A断点条件匹配前，通道B断点条件匹配时，不产生用户断点。另外，通道A与通道B的断点条件同时匹配时，也不产生顺序断点。指定顺序断点时，在通道A条件匹配，且通道B条件不匹配的情况下，欲清除通道A条件匹配时，必须将BRCR寄存器的SEQ位和通道A的条件匹配标志清0。
2. 指定顺序断点时，可选择L总线或I总线，也可指定执行次数断点条件。例如，指定执行次数断点条件时，如果在通道A断点条件匹配后，通道B断点条件为BETR=H'0001时匹配，则满足断点条件。

7.4.5 保存的程序计数器值

产生用户断点时，将需重新执行的指令地址压栈，并转移至异常处理状态。指定 L 总线作为断点条件时，可确定产生用户断点的指令 （断点条件包含数据的情况除外）；指定 I 总线作为断点条件时，不可确定产生用户断点的指令。

1. 指定取指令 （指令执行前）作为断点条件时
   将与断点条件匹配的指令地址压栈。不执行条件匹配指令，而在执行前产生用户断点。但，延迟槽指令的条件匹配时，将延迟转移指令的地址压栈。
2. 指定取指令 （指令执行后）作为断点条件时
   将与断点条件匹配的指令的下一条指令的地址压栈。执行条件匹配指令，并在执行该指令的下一条指令前，产生用户断点。延迟转移指令及其延迟槽匹配时，执行这些指令，且将转移目标的地址压栈。
3. 指定数据存取 （仅地址）作为断点条件时
   将与紧接着断点条件匹配指令后的指令的地址压栈。执行条件匹配指令，并在执行该指令的下一条指令前，产生用户断点。但，延迟槽指令条件匹配时，将转移目标的地址压栈。
4. 指定数据存取 （地址+数据）作为断点条件时
   如果将数据追加到断点条件，则将与断点条件匹配指令的下一条指令的地址或下下一条指令的地址压栈。无法正确确定产生用户断点的位置。
   延迟槽指令的条件匹配时，将转移目标地址压栈。另外，条件匹配指令的下一条指令为转移指令时，有时在转移指令与延迟槽执行结束后产生用户断点。此时，也将转移目标的地址压栈。

7.4.6 PC 跟踪

1. 将BRCR的PCTE置1，可允许PC跟踪。产生转移 （转移指令及中断异常）时，将转移源地址与转移目标地址分别保存至BRSR和BRDR。
2. 按转移种类，保存至BRSR、 BRDR中的值分别如下：

- 由转移指令产生转移时，将转移指令的地址保存至BRSR；将转移目标指令的地址保存至BRDR。
- 由中断和一般异常产生转移时，将产生异常时保存的堆栈值保存至BRSR；将异常处理程序的起始地址保存至BRDR。

3. BRSR和BRDR由2组队列构成。读取保存在PC跟踪寄存器的地址时，先读取队列的第一项。 BRSR和BRDR共用读取指针，必须按BRSR、 BRDR的顺序读取。仅在读取BRDR后队列移位。将BRCR的PCTE位从OFF切换为ON时，队列的值无效。
4. 与AUD共用2组队列，因此必须将STBCR5的MSTP25位清0，并在STBCR6的AUDSRST位置1后，将BRCR的PCTE位置1。
5. 由上电复位、手动复位或AUD软件复位初始化FIFO （8组队列）。由手动复位、 AUD软件复位初始化时，如果将BRCR的PCTE清0后置1，则可重新开始PC跟踪。

7.4.7　使用例

(1)　指定 L 总线取指令周期时产生断点的条件

（例 1-1）

- 指定寄存器

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

  指定条件：通道A/通道B为独立模式

  ＜通道A＞

  地址：H'00000404、地址屏蔽：H'00000000

  数据：H'00000000、数据屏蔽：H'00000000

  总线周期：L总线/取指令 （指令执行后） /读取 （断点条件不包含操作数长度）

  ＜通道B＞

  地址：H'00008010、地址屏蔽：H'00000006

  数据：H'00000000、数据屏蔽：H'00000000

  总线周期：L总线/取指令 （指令执行前） /读取 （断点条件中不包含操作数长度）

  执行地址H'00000404的指令后、或执行地址H'00008010～H'00008016的指令前，产生用户断点。

（例 1-2）

- 指定寄存器

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

  指定条件：通道A/通道B为顺序模式

  ＜通道A＞

  地址：H'00037226、地址屏蔽：H'00000000

  数据：H'00000000、数据屏蔽：H'00000000

  总线周期：L总线/取指令 （指令执行前） /读取/字

  ＜通道B＞

  地址：H'0003722E、地址屏蔽：H'00000000

  数据：H'00000000、数据屏蔽：H'00000000

  总线周期：L总线/取指令 （指令执行前） /读取/字

  执行地址H'00037226的指令后、或执行地址H'0003722E的指令前，产生用户断点。

（例 1-3）

- 指定寄存器

  BARA=H'00027128、 BAMRA=H'00000000、 BBRA=H'005A、 BDRA=H'00000000、
  BDMRA=H'00000000、 BARB=H'00031415、 BAMRB=H'00000000、 BBRB=H'0054、
  BDRB=H'00000000、 BDMRB=H'00000000、 BRCR=H'00000000
  指定条件：通道A/通道B为独立模式
  ＜通道A＞
  地址：H'00027128、地址屏蔽：H'00000000
  数据：H'00000000、数据屏蔽：H'00000000
  总线周期：L总线/取指令 （指令执行前） /写入/字
  ＜通道B＞
  地址：H'00031415、地址屏蔽：H'00000000
  数据：H'00000000、数据屏蔽：H'00000000
  总线周期：L总线/取指令 （指令执行前） /读取 （断点条件不包含操作数长度）

  通道A中，取指令不是写入周期，所以不产生用户断点。通道B中，对偶数地址执行取指令，所以不产生用户断点。

（例 1-4）

- 指定寄存器

  BARA=H'00037226、 BAMRA=H'00000000、 BBRA=H'005A、 BDRA=H'00000000、
  BDMRA=H'00000000、 BARB=H'0003722E、 BAMRB=H'00000000、 BBRB=H'0056、
  BDRB=H'00000000、 BDMRB=H'00000000、 BRCR=H'00000008
  指定条件：通道A/通道B为顺序模式
  ＜通道A＞
  地址：H'00037226、地址屏蔽：H'00000000
  数据：H'00000000、数据屏蔽：H'00000000
  总线周期：L总线/取指令 （指令执行前） /写入/字
  ＜通道B＞
  地址：H'0003722E、地址屏蔽：H'00000000
  数据：H'00000000、数据屏蔽：H'00000000
  总线周期：L总线/取指令 （指令执行前） /读取/字

  通道A中，取指令不是写入周期，所以不产生顺序条件匹配，因此，不产生用户断点。

（例 1-5）

- 指定寄存器

  BARA=H'00000500、 BAMRA=H'00000000、 BBRA=H'0057、 BDRA=H'00000000、
  BDMRA=H'00000000、 BARB=H'00001000、 BAMRB=H'00000000、 BBRB=H'0057、
  BDRB=H'00000000、 BDMRB=H'00000000、 BRCR=H'00000001、 BETR=H'0005
  指定条件：通道A/通道B为独立模式
  ＜通道A＞
  地址：H'00000500、地址屏蔽：H'00000000
  数据：H'00000000、数据屏蔽：H'00000000
  总线周期：L总线/取指令 （指令执行前） /读取/长字
  执行次数断点允许 （5次）
  ＜通道B＞
  地址：H'00001000、地址屏蔽：H'00000000
  数据：H'00000000、数据屏蔽：H'00000000
  总线周期：L总线/取指令 （指令执行前） /读取/长字

  通道A中，执行4次地址H'00000500的指令后，在第5次执行指令前，产生用户断点。通道B中，执行地址H'00001000的指令前，产生用户断点。

（例 1-6）

- 指定寄存器

  BARA=H'00008404、 BAMRA=H'00000FFF、 BBRA=H'0054、 BDRA=H'00000000、
  BDMRA=H'00000000、 BARB=H'00008010、 BAMRB=H'00000006、 BBRB=H'0054、
  BDRB=H'00000000、 BDMRB=H'00000000、 BRCR=H'00000400
  指定条件：通道A/通道B为独立模式
  ＜通道A＞
  地址：H'00008404、地址屏蔽：H'00000FFF
  数据：H'00000000、数据屏蔽：H'00000000
  总线周期：L总线/取指令 （指令执行后） /读取 （断点条件不包含操作数长度）
  ＜通道B＞
  地址：H'00008010、地址屏蔽：H'00000006
  数据：H'00000000、数据屏蔽：H'00000000
  总线周期：L总线/取指令 （指令执行前） /读取 （断点条件不包含操作数长度）

  执行地址H'00008000～H'00008FFE的指令后，或执行地址H'00008010～H'00008016的指令前，产生用户断点。

(2)　指定 L 总线数据存取周期时产生断点的条件

（例 2-1）

- 指定寄存器

  BARA=H'00123456、 BAMRA=H'00000000、 BBRA=H'0064、 BDRA=H'12345678、
  BDMRA=H'FFFFFFFF、 BARB=H'000ABCDE、 BAMRB=H'000000FF、 BBRB=H'006A、
  BDRB=H'0000A512、 BDMRB=H'00000000、 BRCR=H'00000080
  指定条件：通道A/通道B为独立模式
  ＜通道A＞
  地址：H'00123456、地址屏蔽：H'00000000
  数据：H'12345678、数据屏蔽：H'FFFFFFFF
  总线周期：L总线/数据存取/读取 （断点条件不包含操作数长度）
  ＜通道B＞
  地址：H'000ABCDE、地址屏蔽：H'000000FF
  数据：H'0000A512、数据屏蔽：H'00000000
  总线周期：L总线/数据存取/写入/字

  通道A中，长字读取地址H'00123454、字读取地址H'00123456或字节读取地址H'00123456时，产生用户断点。通道B中，对H'000ABC00～H'000ABCFE写入字H'A512时，产生用户断点。

(3)　指定 I 总线数据存取周期时产生断点的条件

（例 3-1）

- 指定寄存器

  BARA=H'00314154、 BAMRA=H'00000000、 BBRA=H'0194、 BDRA=H'12345678、
  BDMRA=H'FFFFFFFF、 BARB=H'00055555、 BAMRB=H'00000000、 BBRB=H'01A9、
  BDRB=H'00007878、 BDMRB=H'00000F0F、 BRCR=H'00000080
  指定条件：通道A/通道B为独立模式
  ＜通道A＞
  地址：H'00314154、地址屏蔽：H'00000000
  数据：H'12345678、数据屏蔽：H'FFFFFFFF
  总线周期：I总线 （CPU周期） /取指令/读取 （断点条件不包含操作数长度）
  ＜通道B＞
  地址：H'00055555、地址屏蔽：H'00000000
  数据：H'00000078、数据屏蔽； H'0000000F
  总线周期：I总线 （CPU周期） /数据存取/写入/字节

  通道A中，在对外部存储器空间的地址H'00314156取指令时，产生用户断点。通道B中，在CPU对外部存储器空间的地址H'00055555写入字节数据H'7x时，产生用户断点。

7.5 使用时的注意事项

1. 通过I总线读取或写入UBC寄存器，因此，从执行改写UBC寄存器的指令到实际反映该值期间，可能不会产生所希望的用户断点。为了得知更改UBC寄存器的时序，必须读取最后写入的寄存器，之后的指令对新写入的寄存器值有效。
2. UBC不可通过相同通道监视L总线周期和I总线周期。
3. 指定顺序断点的注意事项如下：
   设定顺序断点时，如果在产生A通道匹配之后，产生通道B匹配，则产生条件匹配。因此，即使设定同时产生通道A匹配和通道B匹配的总线周期，也不产生用户断点。
4. 由相同指令产生用户断点和其他异常时，按表5.1中所规定的优先顺序执行判断。产生较高优先级的异常时，不产生用户断点。

- 可优先于其他任何异常接受指令执行前的断点。
- 指令执行后的断点或数据存取断点与优先级较高的再执行型异常 （包含指令执行前的断点）同时产生时，接受再执行型异常，且表示条件匹配的标志不被置位 （但，有5.项所示的异常事项）。由异常处理解除再执行型异常源，在相同指令再次执行结束时，再次产生用户断点，并标志置位。
- 指令执行后断点或数据存取断点与优先级较高的结束型异常 （TRAPA）同时产生时，不产生用户断点，但表示条件匹配的标志置位。

5. 作为第4项异常事项，有下列注意事项：
   对于因数据存取产生CPU地址错误的指令，指令执行后的断点和数据存取断点成立时，在用户断点中断前，优先产生CPU地址错误，此时， UBC的条件匹配标志也置位。
6. 在延迟槽产生用户断点时，有下列注意事项：
   对RTE指令的延迟槽指令设定指令执行前的断点时，在执行RTE指令的转移目标前，不产生用户断点。
7. UBC模块待机时，不能使用用户断点功能。另外，模块待机中，不得读取/写入UBC寄存器，否则，无法保证该值。
8. 不得对SLEEP指令和延迟槽为SLEEP指令的转移指令设定执行指令后的断点。另外，不得对SLEEP指令及SLEEP指令的前1～2条指令设定数据存取断点。
9. 在MCU扩展模式使用UBC时，不得判断外部空间的断点。

第 8 章 数据传送控制器 （DTC）

本 LSI 内置数据传送控制器 （DTC）。 DTC 可由中断请求启动，用于传送数据。

8.1 特点

- 可进行任意通道数的传送
- 可链传送 （对一个启动源可传送多个数据）
  仅在指定次数的数据传送结束后，才可执行链传送 （计数器 = 0）
- 传送模式：3 种
  可选择普通传送模式、重复传送模式或块传送模式
  可选择传送源和传送目标地址为递增、递减或固定
- 以32位指定传送源和传送目标地址，并可直接指定4G字节的地址空间
- 传送的数据长度可设定为字节、字或长字
- 可向CPU请求启动DTC的中断
  一次数据传送结束后，可向CPU产生中断请求
  指定的数据传送结束后，可向CPU产生中断请求
- 可指定传送信息跳读
- 对选择为固定的传送源地址和传送目标地址执行跳过回写
- 可设定模块停止模式

DTC 框图如图 8.1 所示。可在数据区分配 DTC 的传送信息 *。

【注】 * 传送信息分配于内部 RAM 时， RAMCR 的 RAME 位必须置 1。

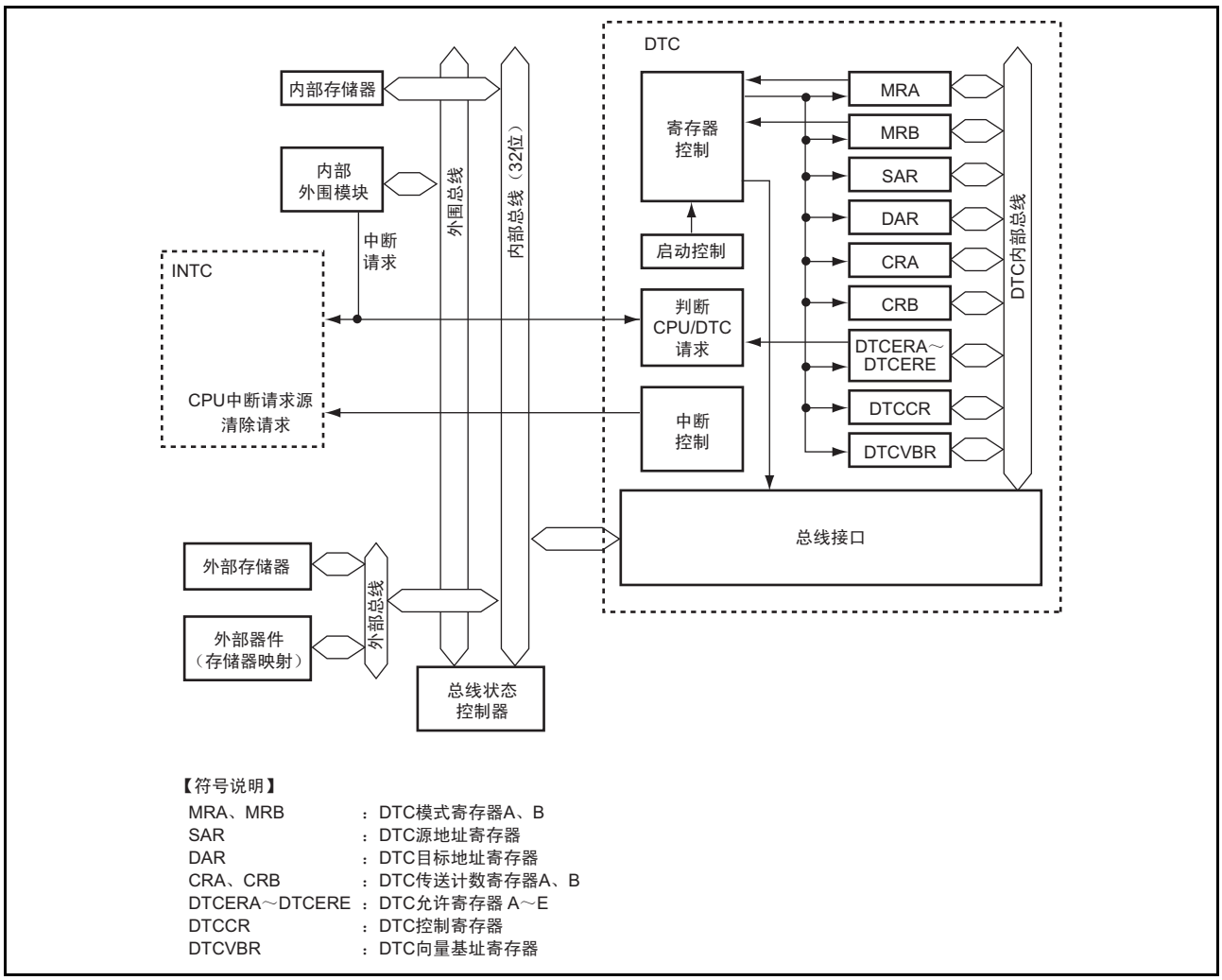


图 8.1　DTC 框图

8.2　寄存器说明

DTC 有以下寄存器。有关寄存器的地址及各处理状态的寄存器状态，详情参阅 “第 24 章 寄存器一览表”。

CPU 不可直接存取 MRA、 MRB、 SAR、 DAR、 CRA、 CRB 等 6 个寄存器。这些寄存器可作为传送信息分配到数据区域。产生 DTC 启动源时，根据各启动源所定的向量地址读取传送信息的起始地址，并将任意传送信息传送到 DTC 后执行数据传送。数据传送结束时，回写寄存器的内容。

此外， CPU 可直接存取 DTCERA ～ DTCERE、 DTCCR、 DTCVBR。

表 8.1　寄存器结构

寄存器名称	简称	R/W	初始值	地址	存取长度
DTC 允许寄存器 A	DTCERA	R/W	H'0000	H'FFFFCC80	8、16
DTC 允许寄存器 B	DTCERB	R/W	H'0000	H'FFFFCC82	8、16
DTC 允许寄存器 C	DTCERC	R/W	H'0000	H'FFFFCC84	8、16
DTC 允许寄存器 D	DTCERD	R/W	H'0000	H'FFFFCC86	8、16
DTC 允许寄存器 E	DTCERE	R/W	H'0000	H'FFFFCC88	8、16
DTC 控制寄存器	DTCCR	R/W	H'00	H'FFFFCC90	8
DTC 向量基址寄存器	DTCVBR	R/W	H'00000000	H'FFFFCC94	8、16、32
总线功能扩展寄存器	BSCEHR	R/W	H'0000	H'FFFFE89A	8、16

8.2.1　DTC 模式寄存器 A（MRA）

MRA 选择 DTC 的运行模式， CPU 不可直接存取 MRA。

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

位	位名称	初始值	R/W	说　明
7、6	MD[1:0]	不定	—	DTC 模式 1、0 指定 DTC 的传送模式。 00：普通传送模式 01：重复传送模式 10：块传送模式 11：禁止设定
5、4	Sz[1:0]	不定	—	DTC 数据传送长度 1、0 指定传送数据的长度。 00：字节传送 01：字传送 10：长字传送 11：禁止设定
3、2	SM[1:0]	不定	—	源地址模式 1、0 指定数据传送后 SAR 的运行。 0x：SAR 为固定 （跳过 SAR 的回写。） 10：传送后 SAR 递增 （Sz1、Sz0 为 B'00 时，加 1；为 B'01 时，加 2；为 B'10 时，加 4） 11：传送后 SAR 递减 （Sz1、Sz0 为 B'00 时，减 1；为 B'01 时，减 2；为 B'10 时，减 4）
1、0	—	不定	—	保留位 写入值总是为 0。

【注】 x：Don't care

8.2.2　DTC 模式寄存器 B（MRB）

MRB 选择 DTC 的运行模式。CPU 不可直接存取 MRB。

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

位	位名称	初始值	R/W	说　明
7	CHNE	不定	—	允许 DTC 链传送 指定链传送。链传送的详情参阅“8.5.6 链传送”。由 CHNS 位选择链传送条件。 0：禁止链传送 1：允许链传送
6	CHNS	不定	—	选择 DTC 链传送 选择链传送的条件。下一个传送为链传送时，不判断指定传送次数是否结束，也不清除启动源标志或 DTCER。 0：连续执行链传送 1：仅在传送计数器 =0 时，执行链传送
5	DISEL	不定	—	选择 DTC 中断 此位为 1 时，每传送 1 次数据或块数据，就向 CPU 请求中断，此位为 0 时，仅在指定次数的数据传送结束时，向 CPU 请求中断。
4	DTS	不定	—	选择 DTC 传送模式 重复传送模式或块传送模式时，指定源或目标的任意一个为重复区域或块区域。 0：目标为重复区域或块区域 1：源为重复区域或块区域
3、2	DM[1:0]	不定	—	目标地址模式 1、0 指定数据传送后 DAR 的运行。 0x：DAR 为固定 （跳过 DAR 的回写。） 10：传送后 DAR 递增 （Sz1、Sz0 为 B'00 时，加 1；为 B'01 时，加 2；为 B'10 时，加 4） 11：传送后 DAR 递减 （Sz1、Sz0 为 B'00 时，减 1；为 B'01 时，减 2；为 B'10 时，减 4）
1、0	—	不定	—	保留位 写入值总是为 0。

【注】 x：Don't care

8.2.3　DTC 源地址寄存器 （SAR）

SAR 为 32 位寄存器，指定 DTC 传送数据的传送源地址。

CPU 不可直接存取 SAR。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
初始值：	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W：	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

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

8.2.4　DTC 目标地址寄存器 （DAR）

DAR 为 32 位寄存器，指定 DTC 传送数据的传送目标地址。

CPU 不可直接存取 DAR。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
初始值：	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W：	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

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

8.2.5　DTC 传送计数寄存器 A （CRA）

CRA 为 16 位寄存器，指定 DTC 数据传送的次数。

在普通传送模式， CRA 整体用作 16 位传送计数器 （1 ～ 65536）。每传送 1 次数据，CRA 就递减 （–1）。计数器值为 H'0000 时，清除启动源对应的 DTCEn 位 （n=15 ～ 0）后，向 CPU 请求中断。设定值为 H'0001 时，传送 1 次；为 H'FFFF 时，传送 65535 次；为 H'0000 时，传送 65536 次。

在重复传送模式， CRA 分为高 8 位的 CRAH 和低 8 位的 CRAL。 CRAH 保存传送次数， CRAL 用作 8 位传送计数器 （1 ～ 256）。每传送 1 次数据， CRAL 就递减 （–1）。计数器值为 H'00 时，传送 CRAH 的内容。设定值为 CRAH=CRAL=H'01 时，传送 1 次；为 H'FF 时，传送 255 次；为 H'00 时，传送 256 次。

在块传送模式， CRA 分为高 8 位的 CRAH 和低 8 位的 CRAL。 CRAH 保存块长度， CRAL 用作 8 位块长度计数器 （1 ～ 256 个字节、 1 ～ 256 个字或 1 ～ 256 个长字）。每传送 1 次数据， CRAL 就递减 （–1） 1 个字节 （或 1 个字、 1 个长字）。计数值为 H'00 时，传送 CRAH 的内容。设定值为 CRAH=CRAL=H'01 时，块长度为 1 个字节 （或 1 个字、 1 个长字）；设定值为 H'FF 时，块长度为 255 个字节 （或 255 个字、 255 个长字）；设定值为 H'00 时，块长度为 256 个字节 （或 256 个字、 256 个长字）。

CPU 不可直接存取 CRA。

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

8.2.6　DTC 传送计数寄存器 B （CRB）

CRB 为 16 位寄存器，块传送模式时，指定 DTC 块数据传送的次数。 CRB 可用作 16 位的传送次数计数器（1 ～ 65536），每执行 1 次块数据传送，就递减 （–1）。计数值为 H'0000 时，清除启动源对应的 DTCEn 位（n=15 ～ 0）后，向 CPU 请求中断。设定值为 H'0001 时，传送 1 次；为 H'FFFF 时，传送 65535 次；为 H'0000 时，传送 65536 次。

在普通传送模式及重复传送模式，不使用 CRB。 CPU 不可直接存取 CRB。

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

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

DTCER 寄存器选择启动 DTC 的中断源有 DTCERA ～ DTCERE。各中断源与 DTCE 位的对应关系参照表 8.2。

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

位	位名称	初始值	R/W	说　明
15	DTCE15	0	R/W	DTC 启动允许 15 ～ 0
14	DTCE14	0	R/W	置 1 时，选择对应中断源作为 DTC 启动源。
13	DTCE13	0	R/W	［清除条件］
12	DTCE12	0	R/W	• 读取要清除的位为 1 的状态后，写入 0 时
11	DTCE11	0	R/W	• MRB 的 DISEL 位为 1，且 1 次数据传送结束时
10	DTCE10	0	R/W	• 指定次数的传送结束时
9	DTCE9	0	R/W	DISEL 位为 0，且指定次数的传送未结束时，不清除此位。
8	DTCE8	0	R/W	［置位条件］
7	DTCE7	0	R/W	• 读取要置位的位为 0 后，写入 1
6	DTCE6	0	R/W	
5	DTCE5	0	R/W	
4	DTCE4	0	R/W	
3	DTCE3	0	R/W	
2	DTCE2	0	R/W	
1	DTCE1	0	R/W	
0	DTCE0	0	R/W	

8.2.8　DTC 控制寄存器 （DTCCR）

DTCCR 设定传送信息跳读。

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

【注】* 为了清除标志，只能读取1后，再写入0。

位	位名称	初始值	R/W	说　明
7 ～ 5	—	均为 0	R	保留位 读取值、写入值总是为 0。
4	RRS	0	R/W	DTC 传送信息跳读允许 控制向量地址和传送信息的读取。比较 DTC 向量编号与上次启动的向量编号。向量编号值匹配并此位为 1 时，不读取向量地址和传送信息，而执行 DTC 数据传送。如果上次启动为链传送，则必须读取向量地址和传送信息。 0：不跳读传送信息 1：向量编号值匹配时，跳读传送信息
3	RCHNE	0	R/W	DTC 重复传送后链传送允许 重复传送过程中，禁止 / 允许传送计数器 =0 时的链传送。重复传送过程中，传送计数器 (CRAL)=0 时， CRAL 回写为 CRAH 指定的值，因此不产生传送计数器 =0 时的链传送。通过将此位置 1，允许传送计数器回写时的链传送。 0：禁止重复传送后的链传送 1：允许重复传送后的链传送
2、1	—	均为 0	R	保留位 仅读取有效，写入无效。
0	ERR	0	R/(W) *	传送停止标志 该标志表示产生 DTC 地址错误或 NMI 中断请求。如果在 DTC 启动时产生 DTC 地址错误或 NMI 中断请求，则在 DTC 总线权释放时释放总线权，之后执行 DTC 地址错误或 NMI 中断处理。数据传送后， DTC 在传送信息写入状态停止。 0：无中断请求 1：产生中断请求 ［清除条件］ • 读取 1 的状态后，写入 0 时

8.2.9　DTC 向量基址寄存器 （DTCVBR）

DTCVBR 为 32 位寄存器，设定计算向量表地址时的基址。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

位	位名称	初始值	R/W	说　明
31 ～ 12		均为 0	R/W	bit11 ～ 0 读取值、写入值总是为 0。
11 ～ 0	—	均为 0	R	

8.2.10　总线功能扩展寄存器 （BSCEHR）

BSCEHR 为 16 位寄存器，设定 DTC 总线权释放时序。详情参阅 “9.4.4 总线功能扩展寄存器（BSCEHR）”。

8.3　启动源

DTC 由中断请求启动，由 DTCER 选择启动中断源。对应位置 1 时，为 DTC 的启动源；清 0 时，为 CPU 的中断源。一次数据传送 （链传送时，连续的最后一个传送）结束时，清除作为启动源的中断标志或 DTCER 的对应位。

8.4　传送信息的分配和 DTC 向量表

在数据区域分配传送信息，传送信息的起始地址必须为地址 4n。如果未指定为地址 4n，则忽略低 2 位执行存取 （[1:0]=B'00）。数据区域的传送信息分配如图 8.2 所示。 DTC 按各启动源从向量表读取传送信息的起始地址，并从该起始地址读取传送信息。 DTC 向量表和传送信息的对应关系如图 8.3 所示。

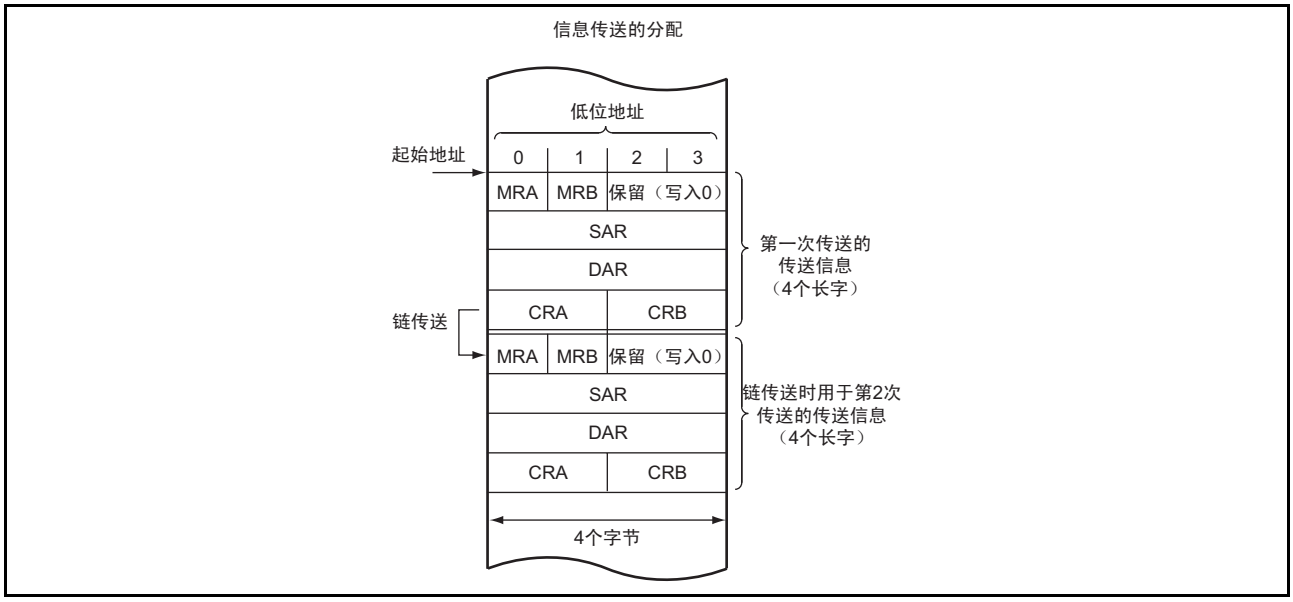


图 8.2　数据区域的传送信息分配

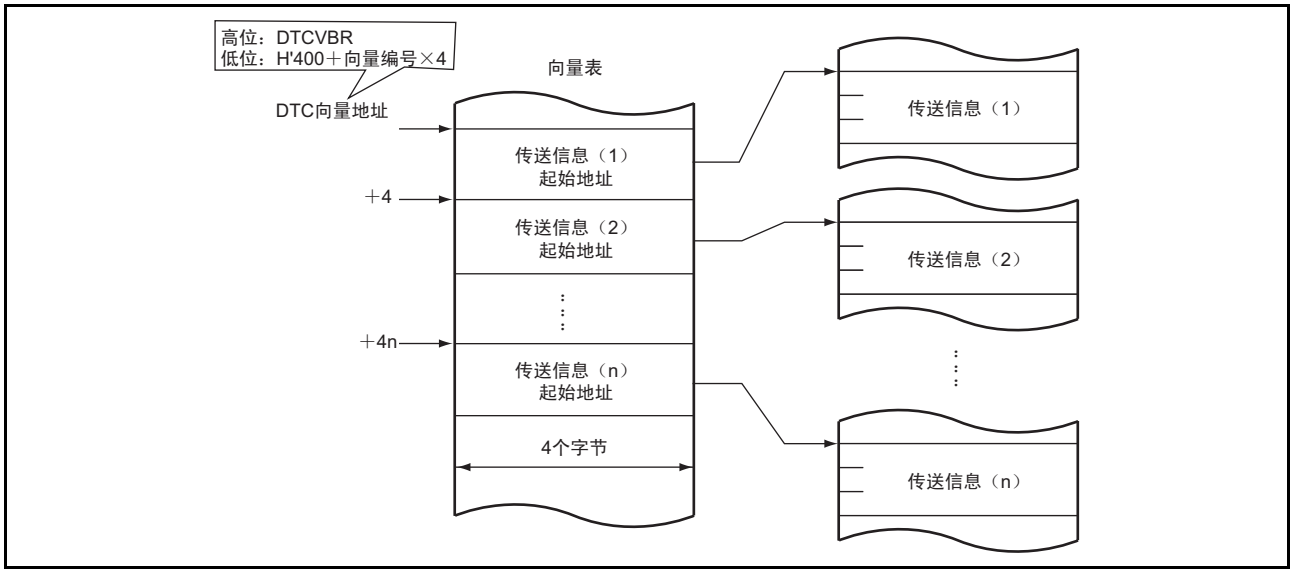


图 8.3　DTC 向量表和传送信息的对应关系

DTC 的启动源和向量地址的对应关系如表 8.2 所示。

表 8.2　中断源、 DTC 向量地址及对应的 DTCE

启动源的产生源	启动源	向量编号	DTC 向量地址偏移量	DTCE*1	传送源	传送目标	优先顺序
外部引脚	IRQ0	64	H'500	DTCERA15	任意 *2	任意 *2	高
	IRQ1	65	H'504	DTCERA14	任意 *2	任意 *2	↑
	IRQ2	66	H'508	DTCERA13	任意 *2	任意 *2	
	IRQ3	67	H'50C	DTCERA12	任意 *2	任意 *2	
MTU2_0	TGIA_0	88	H'560	DTCERB15	任意 *2	任意 *2	
	TGIB_0	89	H'564	DTCERB14	任意 *2	任意 *2	
	TGIC_0	90	H'568	DTCERB13	任意 *2	任意 *2	
	TGID_0	91	H'56C	DTCERB12	任意 *2	任意 *2	
MTU2_1	TGIA_1	96	H'580	DTCERB11	任意 *2	任意 *2	
	TGIB_1	97	H'584	DTCERB10	任意 *2	任意 *2	
MTU2_2	TGIA_2	104	H'5A0	DTCERB9	任意 *2	任意 *2	
	TGIB_2	105	H'5A4	DTCERB8	任意 *2	任意 *2	
MTU2_3	TGIA_3	112	H'5C0	DTCERB7	任意 *2	任意 *2	
	TGIB_3	113	H'5C4	DTCERB6	任意 *2	任意 *2	
	TGIC_3	114	H'5C8	DTCERB5	任意 *2	任意 *2	
	TGID_3	115	H'5CC	DTCERB4	任意 *2	任意 *2	
MTU2_4	TGIA_4	120	H'5E0	DTCERB3	任意 *2	任意 *2	
	TGIB_4	121	H'5E4	DTCERB2	任意 *2	任意 *2	
	TGIC_4	122	H'5E8	DTCERB1	任意 *2	任意 *2	
	TGID_4	123	H'5EC	DTCERB0	任意 *2	任意 *2	
	TCIV_4	124	H'5F0	DTCERC15	任意 *2	任意 *2	
MTU2S_3	TGIA_3S	160	H'680	DTCERC3	任意 *2	任意 *2	
	TGIB_3S	161	H'684	DTCERC2	任意 *2	任意 *2	
	TGIC_3S	162	H'688	DTCERC1	任意 *2	任意 *2	
	TGID_3S	163	H'68C	DTCERC0	任意 *2	任意 *2	
MTU2S_4	TGIA_4S	168	H'6A0	DTCERD15	任意 *2	任意 *2	
	TGIB_4S	169	H'6A4	DTCERD14	任意 *2	任意 *2	
	TGIC_4S	170	H'6A8	DTCERD13	任意 *2	任意 *2	
	TGID_4S	171	H'6AC	DTCERD12	任意 *2	任意 *2	
	TCIV_4S	172	H'6B0	DTCERD11	任意 *2	任意 *2	
MTU2S_5	TGIU_5S	176	H'6C0	DTCERD10	任意 *2	任意 *2	
	TGIV_5S	177	H'6C4	DTCERD9	任意 *2	任意 *2	↓
	TGIW_5S	178	H'6C8	DTCERD8	任意 *2	任意 *2	低

启动源的产生源	启动源	向量编号	DTC 向量地址偏移量	DTCE*1	传送源	传送目标	优先顺序
CMT_0	CMI_0	184	H'6E0	DTCERD7	任意 *2	任意 *2	高
CMT_1	CMI_1	188	H'6F0	DTCERD6	任意 *2	任意 *2	↑
A/D_0	ADI_3	208	H'740	DTCERD2	ADDR0 ～ ADDR7	任意 *2	
A/D_1	ADI_4	212	H'750	DTCERD1	ADDR8 ～ ADDR15	任意 *2	
SCI_0	RXI_0	217	H'764	DTCERE15	SCRDR_0	任意 *2	
	TXI_0	218	H'768	DTCERE14	任意 *2	SCTDR_0	
SCI_1	RXI_1	221	H'774	DTCERE13	SCRDR_1	任意 *2	
	TXI_1	222	H'778	DTCERE12	任意 *2	SCTDR_1	
SCI_2	RXI_2	225	H'784	DTCERE11	SCRDR_2	任意 *2	
	TXI_2	226	H'788	DTCERE10	任意 *2	SCTDR_2	
SSU	SSRXI	233	H'7A4	DTCERE7	SSRDR0 ～ SSRDR3	任意 *2	
	SSTXI	234	H'7A8	DTCERE6	任意 *2	SSTDR0 ～ SSTDR3	
RCAN-ET_0	RM0_0	242	H'7C8	DTCERE3	CONTROL0H ～ CONTROL1L*3	任意 *2	↓
RCAN-ET_1*4	RM0_1	246	H'7D8	DTCERE2	CONTROL0H ～ CONTROL1L*3	任意 *2	低

【注】 *1　无相应中断源的 DTCE 位为保留位，必须写入 0。
通过中断解除软件待机状态时，必须对相应的DTCE位写入0。

*2　外部存储器、存储器映射外部器件、内部存储器、内部外围模块 （DTC、 BSC、 UBC、 AUD、 FLASH 等除外）。
但必须至少将传送源或传送目标设定为内部外围模块。
不可执行外部存储器、存储器映射外部器件、内部存储器间的传送。

*3　必须利用块传送模式等读取到邮箱 0 的信息控制字段 1 （CONTROL1）。

*4　仅适用于 SH7142。

8.5　运行说明

传送模式有普通传送模式、重复传送模式及块传送模式。将传送信息保存至数据区域，可执行任意通道数的数据传送。启动 DTC 时，从数据区域读取传送信息后传送数据，并回写数据传送后的传送信息。

DTC 由 SAR 指定传送源地址；由 DAR 指定传送目标地址。传送后， SAR 和 DAR 分别递增、递减或固定。

DTC 的传送模式如表 8.3 所示。

表 8.3　DTC 的传送模式

传送模式	1 次传送请求可传送的数据长度	存储器地址的增减	传送次数
普通传送模式	1 个字节 / 字 / 长字	增减 1、 2、 4 或固定	1 ～ 65536 次
重复传送模式 *1	1 个字节 / 字 / 长字	增减 1、 2、 4 或固定	1 ～ 256 次 *3
块传送模式 *2	CRAH 指定的块长度 （1 ～ 256 个字节 / 字 / 长字）	增减 1、 2、 4 或固定	1 ～ 65536 次 *4

【注】 *1　将源或目标设定为重复区域

*2　将源或目标设定为块区域

*3　指定次数的传送结束后，恢复初始状态并继续运行

*4　1 次表示 1 块的长度

另外，预先将 MRB 的 CHNE 位置 1，用一个启动源可执行多个传送 （链传送）；也可设定 MRB 的 CHNS 位，设定在传送计数器 = 0 时执行链传送。

DTC 的运行流程如图 8.4 所示。 DTC 传送条件 （含链传送）如表 8.4 所示 （省略了第 2 次到第 3 次传送的组合）。

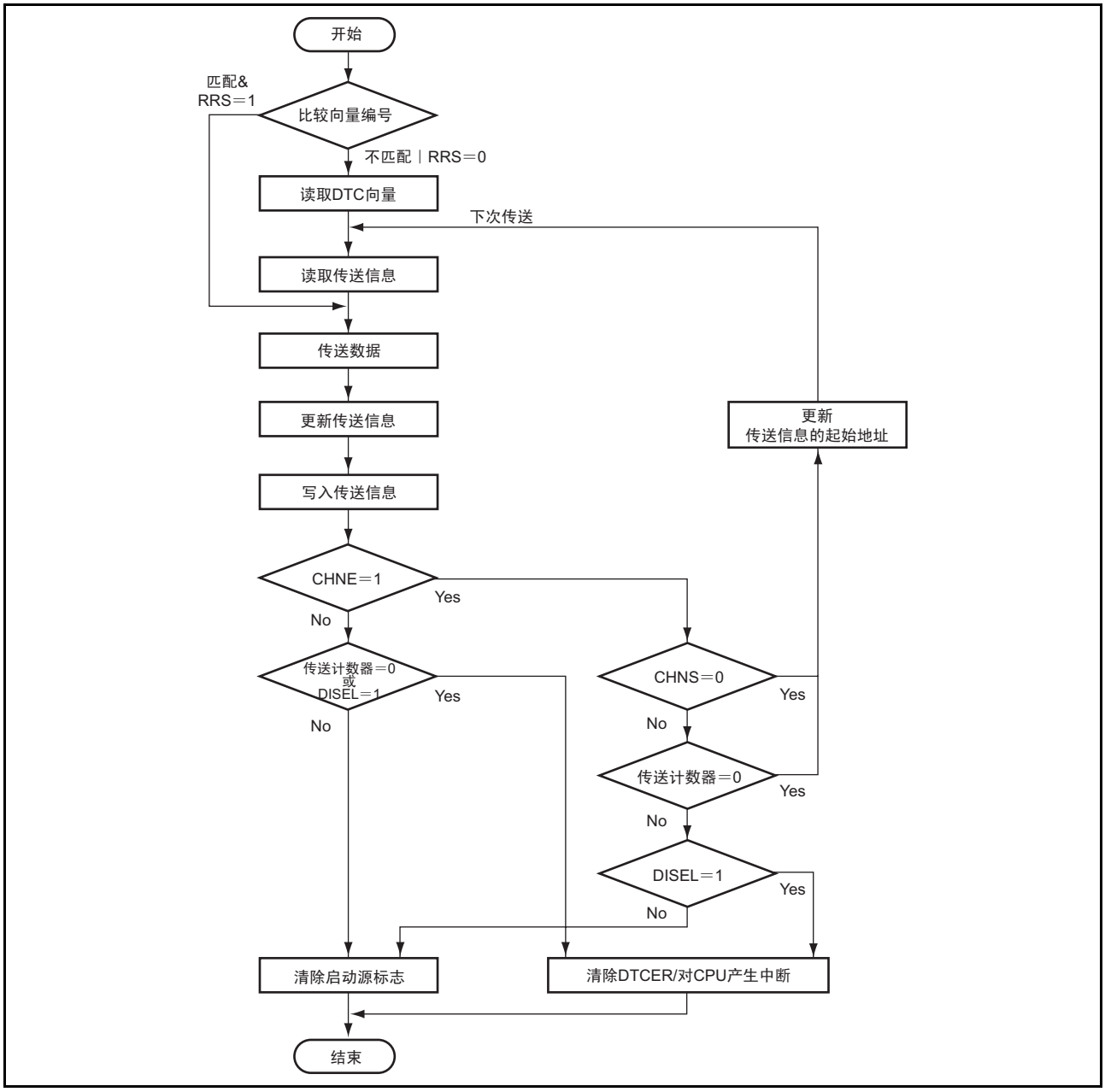


图 8.4　DTC 的运行流程图

表 8.4　DTC 传送条件（含链传送）

传送模式	第 1 次传送					第 2 次传送					DTC 传送
	CHNE	CHNS	RCHNE	DISEL	传送计数器 *1	CHNE	CHNS	RCHNE	DISEL	传送计数器 *1	
普通	0	—	—	0	不为 0	—	—	—	—	—	第 1 次传送后结束
	0	—	—	0	0	—	—	—	—	—	第 1 次传送后结束 向 CPU 请求中断
	0	—	—	1	—	—	—	—	—	—	
	1	0	—	—	—	0	—	—	0	不为 0	第 2 次传送后结束
						0	—	—	0	0	第 2 次传送后结束 向 CPU 请求中断
						0	—	—	—	—	
	1	1	—	0	不为 0	—	—	—	—	—	第 1 次传送后结束
	1	1	—	1	不为 0	—	—	—	—	—	第 1 次传送后结束 向 CPU 请求中断
	1	1	—	—	0	0	—	—	0	不为 0	第 2 次传送后结束
						0	—	—	0	0	第 2 次传送后结束 向 CPU 请求中断
						0	—	—	1	—	
重复	0	—	—	0	—	—	—	—	—	—	第 1 次传送后结束
	0	—	—	1	—	—	—	—	—	—	第 1 次传送后结束 向 CPU 请求中断
	1	0	—	—	—	0	—	—	0	—	第 2 次传送后结束
						0	—	—	1	—	第 2 次传送后结束 向 CPU 请求中断
	1	1	—	0	不为 0	—	—	—	—	—	第 1 次传送后结束
	1	1	—	1	不为 0	—	—	—	—	—	第 1 次传送后结束 向 CPU 请求中断
	1	1	0	0	0*2	—	—	—	—	—	第 1 次传送后结束
	1	1	0	1	0*2	—	—	—	—	—	第 1 次传送后结束 向 CPU 请求中断
	1	1	1	—	0*2	0	—	—	0	—	第 2 次传送后结束
						0	—	—	1	—	第 2 次传送后结束 向 CPU 请求中断
块	0	—	—	0	不为 0	—	—	—	—	—	第 1 次传送后结束
	0	—	—	0	0	—	—	—	—	—	第 1 次传送后结束 向 CPU 请求中断
	0	—	—	1	—	—	—	—	—	—	
	1	0	—	—	—	0	—	—	0	不为 0	第 2 次传送后结束
						0	—	—	0	0	第 2 次传送后结束 向 CPU 请求中断
						0	—	—	1	—	
	1	1	—	0	—	—	—	—	—	—	第 1 次传送后结束
	1	1	—	1	不为 0	—	—	—	—	—	第 1 次传送后结束 向 CPU 请求中断
	1	1	—	1	0	0	—	—	0	不为 0	第 2 次传送后结束
						0	—	—	0	0	第 2 次传送后结束 向 CPU 请求中断
						0	—	—	1	—	

【注】 *1　普通传送模式：CRA；重复传送模式：CRAL；块传送模式：CRB。

*2　表示 CRAL 的内容改写为 CRAH 的内容时。

8.5.1　传送信息跳读功能

设定 DTCCR 的 RRS 位，可跳读向量地址和传送信息。随时比较 DTC 向量编号与上次启动的向量编号，比较结果匹配且 RRS=1 时，不读取向量地址和传送信息，而执行 DTC 数据传送。上次启动为链传送时，必须读取向量地址和传送信息。传送信息跳读时序如图 8.5 所示。

更新向量表和传送信息时，必须设定 RRS=0，并在更新向量表和传送信息后，再设定 RRS 位。 RRS=0 时，舍弃所保持的向量编号，在下次启动时读取更新的向量表及传送信息。

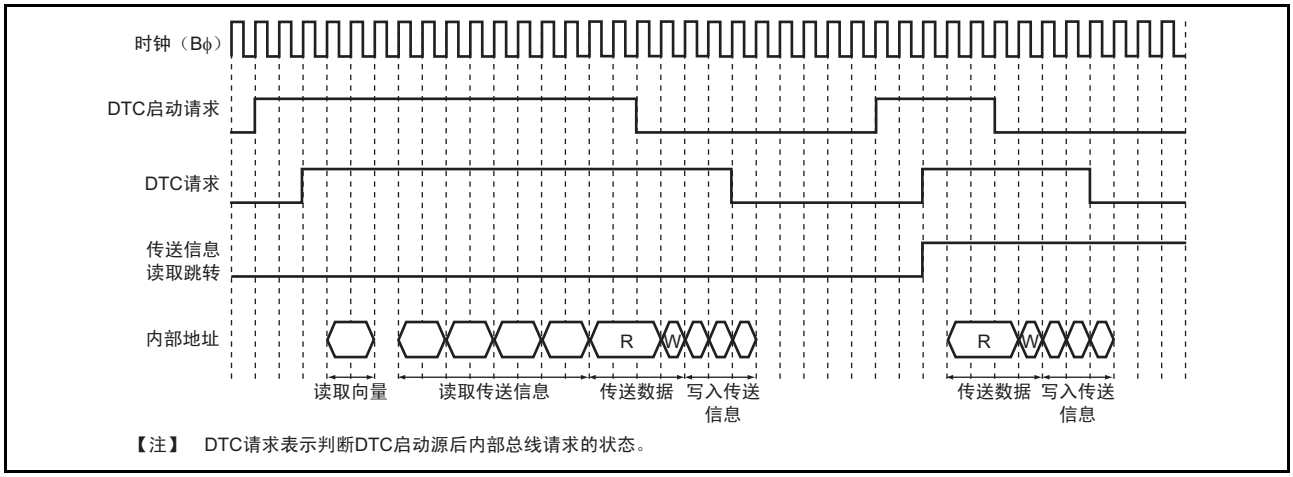


图 8.5　传送信息跳读时序图
（由内部外围模块启动， Iϕ:Bϕ:Pϕ=1:1/2:1/2、
从内部外围模块向内部 RAM 传送数据，传送信息的写入为 3 个状态时）

8.5.2　传送信息跳过回写功能

设定 MRA 的 SM1 位和 MRB 的 DM1 位为固定地址时，一部分传送信息不可回写。传送信息跳过回写的条件和跳过回写的寄存器如表 8.5 所示。 CRA、 CRB 必须回写，而 MRA、 MRB 必须跳过回写。

表 8.5　传送信息跳过回写的条件和跳过回写的寄存器

SM1	DM1	SAR	DAR
0	0	跳过	跳过
0	1	跳过	回写
1	0	回写	跳过
1	1	回写	回写

8.5.3　普通传送模式

用 1 个启动源，执行 1 个字节、1 个字或 1 个长字的数据传送。传送次数为 1 ～ 65536 次。传送源地址和传送目标地址可分别设定为递增、递减或固定。指定次数的传送结束后，可向 CPU 请求中断。

普通传送模式的寄存器功能和存储器映射分别如表 8.6 和图 8.6 所示。

表 8.6　普通传送模式的寄存器功能

寄存器	功能	写入传送信息时回写的值
SAR	传送源地址	递增 / 递减 / 固定 *
DAR	传送目标地址	递增 / 递减 / 固定 *
CRA	传送计数 A	CRA-1
CRB	传送计数 B	不更新

【注】 *　跳过传送信息的回写。

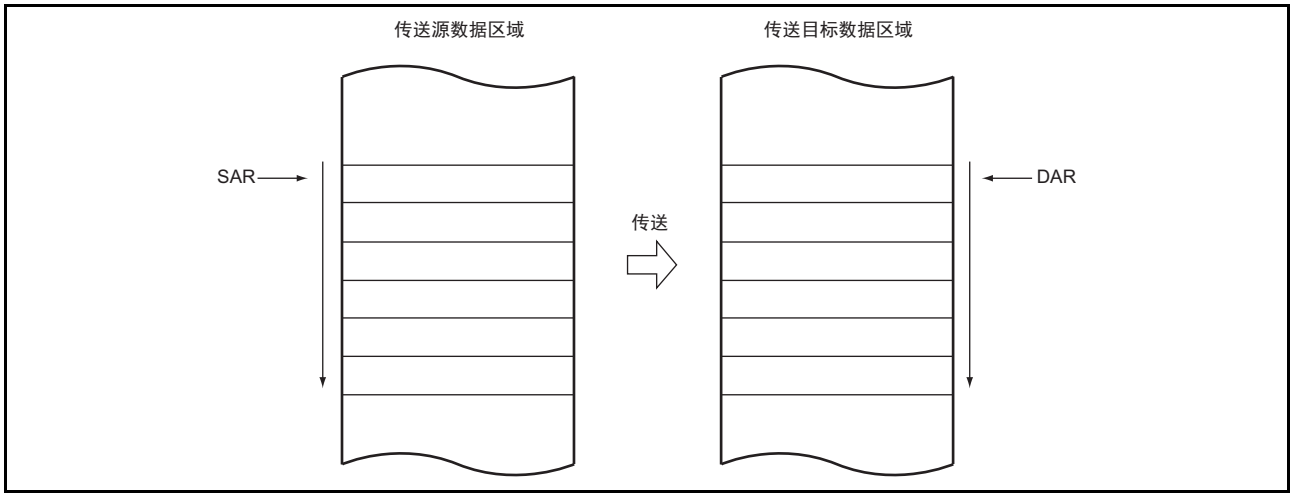


图 8.6　普通传送模式的存储器映射

8.5.4　重复传送模式

用 1 个启动源，执行 1 个字节、 1 个字或 1 个长字数据传送。由 MRB 的 DTS 位指定传送源或传送目标为重复区域。传送次数为 1 ～ 256 次，指定次数的传送结束时，传送计数器及被指定为重复区域的地址寄存器恢复为初始状态，并重复传送。其他地址寄存器连续递增、递减或固定。在重复传送模式，传送计数器（CRAL）为 H'00 时， CRAL 更新为 CRAH 设定的值，因此传送计数器不会为 H'00， DISEL=0 时，不向 CPU 请求中断。

重复传送模式的寄存器功能和存储器映射分别如表 8.7 和图 8.7 所示。

表 8.7　重复传送模式的寄存器功能

寄存器	功能	写入传送信息时回写的值	
		CRAL 不为 1 时	CRAL 为 1 时
SAR	传送源地址	递增 / 递减 / 固定 *	（DTS=0）递增 / 递减 / 固定 * （DTS=1） SAR 的初始值
DAR	传送目标地址	递增 / 递减 / 固定 *	（DTS=0） DAR 的初始值 （DTS=1）递增 / 递减 / 固定 *
CRAH	保存传送计数	CRAH	CRAH
CRAL	传送计数器 A	CRAL-1	CRAH
CRB	传送计数器 B	不更新	不更新

【注】 *　跳过传送信息的回写。

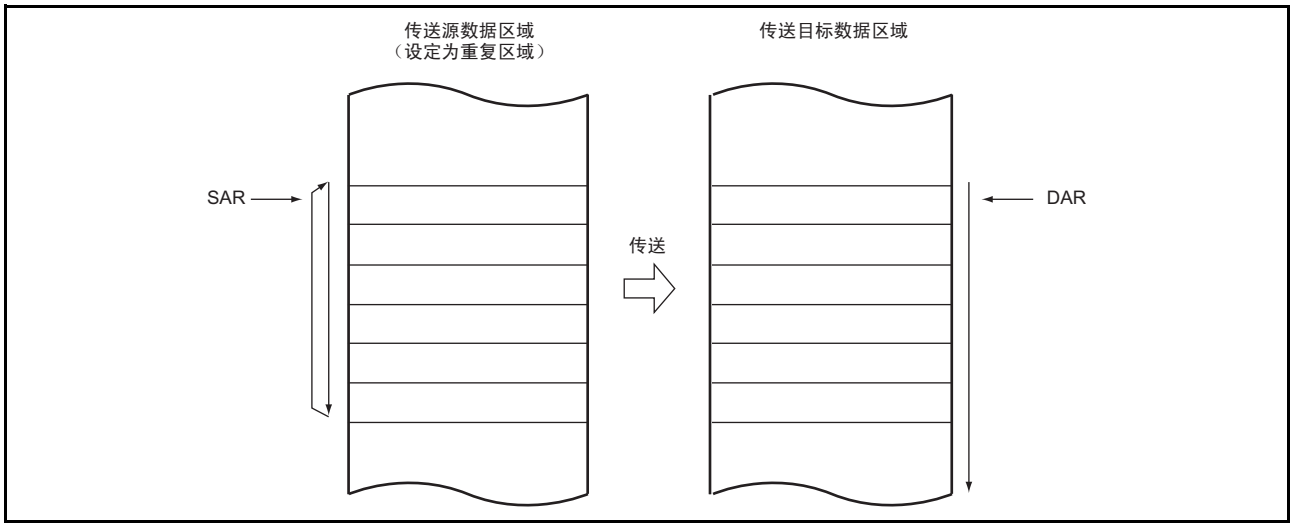


图 8.7　重复传送模式的存储器映射 （指定传送源为重复区域）

8.5.5　块传送模式

用 1 个启动源，执行 1 个块的块数据传送。由 MRB 的 DTS 位指定传送源或传送目标为块区域。块长度为 1 ～ 256 个字节 （或 1 ～ 256 个字、 1 ～ 256 个长字）。一个块的块数据传送结束时，块长度计数器 （CRAL）和被指定为块区域的地址寄存器 （DTS=1 时为 SAR ； DTS=0 时为 DAR）恢复为初始状态。其他地址寄存器连续递增、递减或固定。传送次数为 1 ～ 65536 次。指定次数的块传送结束后，可向 CPU 请求中断。

块传送模式的寄存器功能和存储器映射分别如表 8.8 和图 8.8 所示。

表 8.8　块传送模式的寄存器功能

寄存器	功能	写入传送信息时回写的值
SAR	传送源地址	（DTS=0）递增 / 递减 / 固定 * （DTS=1） SAR 的初始值
DAR	传送目标地址	（DTS=0） DAR 的初始值 （DTS=1）递增 / 递减 / 固定 *
CRAH	保存块长度	CRAH
CRAL	块长度计数器	CRAH
CRB	块传送次数计数器	CRB-1

【注】 *　跳过传送信息的回写。

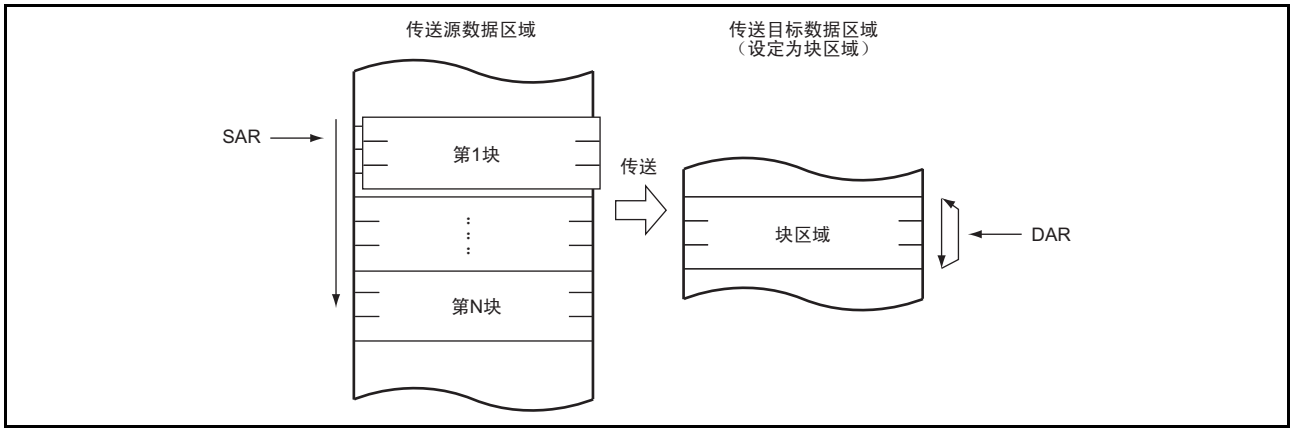


图 8.8　块传送模式的存储器映射 （指定传送目标为块区域）

8.5.6　　链传送

MRB 的 CHNE 位置 1 时，用一个启动源可连续传送多个数据。MRB 的 CHNE 位、CHNS 位分别置 1 时，仅在传送计数器 =0 时执行链传送。可分别独立设定定义数据传送的 SAR、 DAR、 CRA、 CRB 及 MRA、MRB。链传送的运行如图 8.9 所示。

设定 CHNE=1 的数据传送过程中，指定传送次数结束或 DISEL=1 时，不向 CPU 请求中断。此外，CHNE=1 的传送不影响作为启动源的中断源标志及 DTCER。

在重复传送模式， DTCCR 的 RCHNE 位、 MRB 的 CHNE、 CHNS 位分别置 1 时，传送计数器 =1 的传送结束后，可执行链传送。

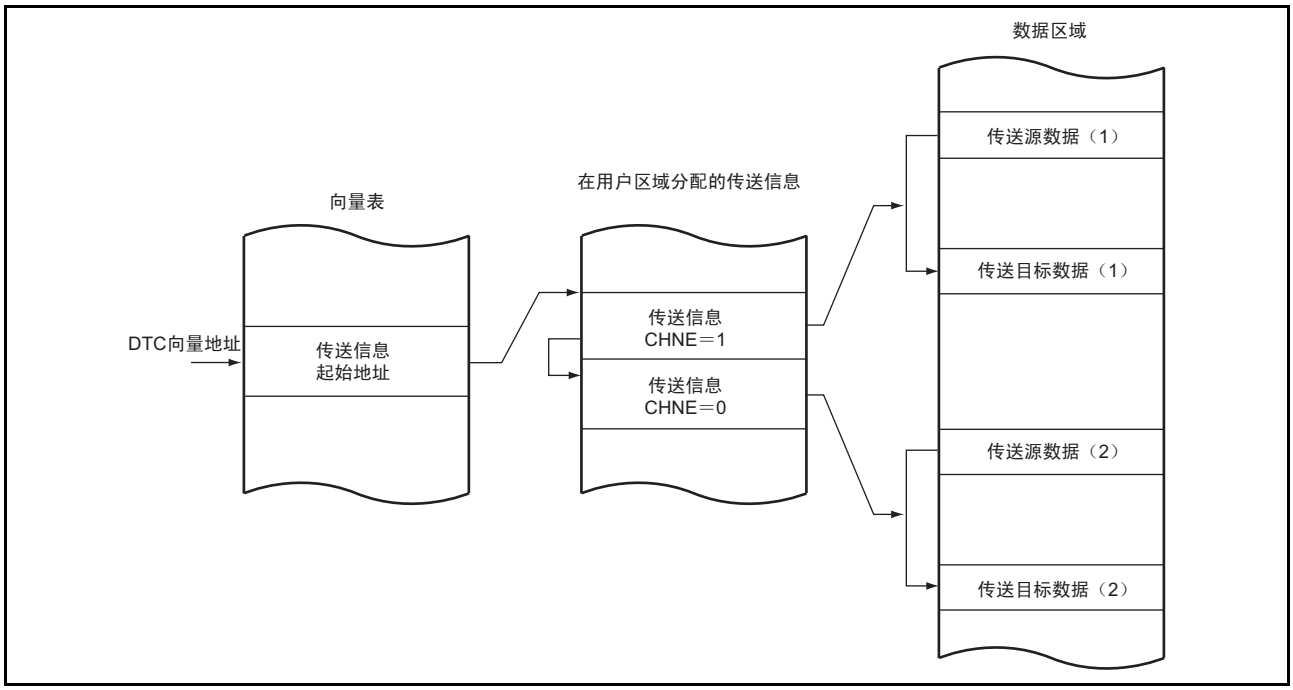


图 8.9　链传送的运行

8.5.7　运行时序

DTC 的运行时序如图 8.10 ～图 8.12 所示。

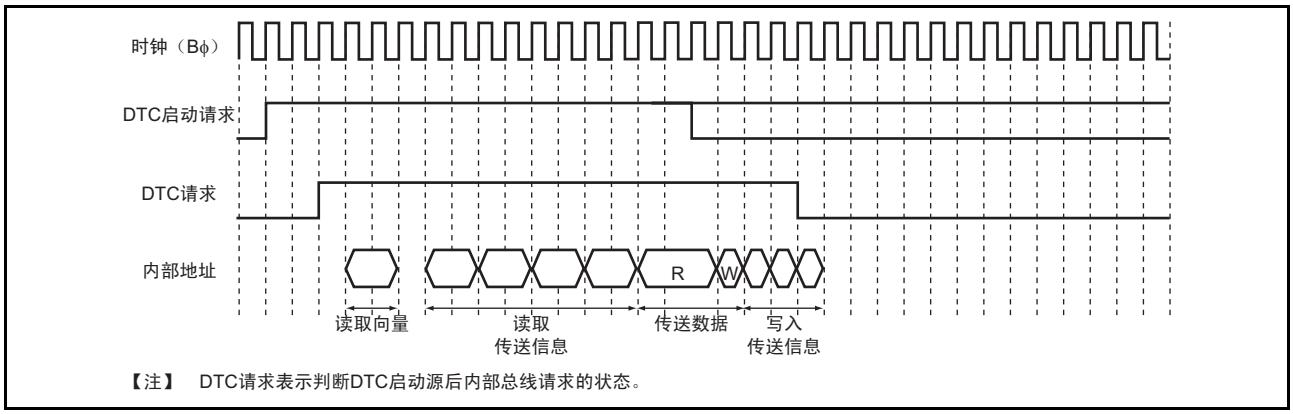


图 8.10　DTC 的运行时序例【普通传送、重复传送】
（由内部外围模块启动， Iϕ:Bϕ:Pϕ=1:1/2:1/2、
从内部外围模块向内部 RAM 传送数据，传送信息的写入为 3 个状态时）

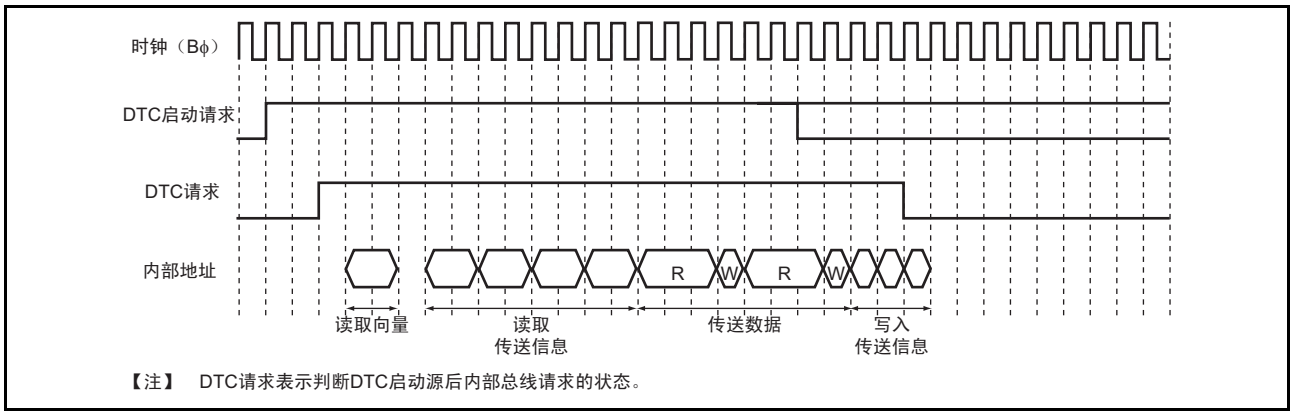


图 8.11　DTC 的运行时序例【块传送、块长度 =2】
（由内部外围模块启动， Iϕ:Bϕ:Pϕ=1:1/2:1/2、
从内部外围模块向内部 RAM 传送数据，传送信息的写入为 3 个状态时）

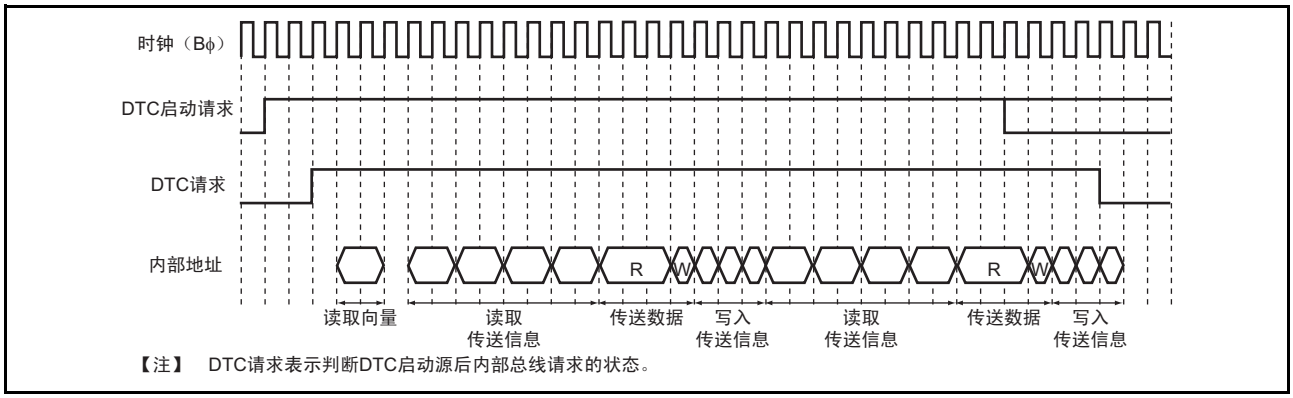


图 8.12　DTC 的运行时序例【链传送】
（由内部外围模块启动， Iϕ:Bϕ:Pϕ=1:1/2:1/2、
从内部外围模块向内部 RAM 传送数据，传送信息的写入为 3 个状态时）

8.5.8　DTC 的执行状态

DTC 传送 1 次数据的执行状态如表 8.9 所示，执行状态所需的状态数如表 8.10 所示。

表 8.9　DTC 的执行状态

模式	读取向量 I		读取传送信息 J		写入传送信息 K			读取数据 L	写入数据 M	内部运行 N	
普通	1	0*1	4	0*1	3	2*2	1*3	1	1	1	0*1
重复	1	0*1	4	0*1	3	2*2	1*3	1	1	1	0*1
块	1	0*1	4	0*1	3	2*2	1*3	1・P	1・P	1	0*1

【注】 *1　跳读传送信息时

*2　SAR 或 DAR 为固定模式时

*3　SAR 和 DAR 为固定模式时

P：块长度 （CRAH、 CRAL 的初始设定）

表 8.10　执行状态所需的状态数

存取对象		内部 RAM*1/ROM*2	内部 I/O 寄存器	外部设备 *4
总线宽度		32 位	16 位	8 位
存取状态		1Bϕ ～ 3Bϕ*1*2	2Pϕ	2Bϕ
执行状态	读取向量 S_I	1Bϕ ～ 3Bϕ*1*2	—	9Bϕ
	读取传送信息 S_J	1Bϕ ～ 3Bϕ*1	—	9Bϕ
	写入传送信息 S_K	1Bϕ ～ 3Bϕ*1	—	2Bϕ*5
	读取字节数据 S_L	1Bϕ ～ 3Bϕ*1	1Bϕ+2Pϕ*3	3Bϕ
	读取字数据 S_L	1Bϕ ～ 3Bϕ*1	1Bϕ+2Pϕ*3	5Bϕ
	读取长字数据 S_L	1Bϕ ～ 3Bϕ*1	1Bϕ+4Pϕ*3	9Bϕ
	写入字节数据 S_M	1Bϕ ～ 3Bϕ*1	1Bϕ+2Pϕ*3	2Bϕ*5
	写入字数据 S_M	1Bϕ ～ 3Bϕ*1	1Bϕ+2Pϕ*3	2Bϕ*5
	写入长字数据 S_M	1Bϕ ～ 3Bϕ*1	1Bϕ+4Pϕ*3	2Bϕ*5
	内部运行 S_N	1		

【注】 *1　内部 RAM 为对象。周期数取决于 Iϕ:Bϕ 比。

	读取	写入
Iϕ:Bϕ=1:1 时	3Bϕ	3Bϕ
Iϕ:Bϕ=1:1/2 时	2Bϕ	1Bϕ
Iϕ:Bϕ=1:1/3 时	2Bϕ	1Bϕ
Iϕ:Bϕ ≤ 1:1/4 时	1Bϕ	1Bϕ

*2　内部 ROM 为对象。周期数取决于 Iϕ:Bϕ 比，周期数与内部 RAM 相等。
只可读取向量。

*3　表中为最快的情况。因内部总线的状态而变慢时， 1Bϕ 为 1Pϕ。

*4　取决于 BSC 寄存器设定。表中是等待 =0 且 CSnWCR 的 WM 位 =1 时的例子。

*5　取决于总线状态。
块传送时频繁写入使外部等待时间变长，或使用外部总线时，不能有效使用写入缓冲器，因而增加周期数。
有关写入缓冲器，详情参阅 “9.5.7(2) LSI 内部总线主控器的存取”。

按以下公式计算执行状态数。
Σ 表示用 1 个启动源所传送次数 （CHNE 位置 1 的次数 +1）的总和。

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

8.5.9　DTC 的总线权释放时序

产生启动请求时，DTC 对总线仲裁请求总线权。DTC 释放总线权的时序为：读取向量后、读取向量后发行 NOP 时、读取传送信息后、传送 1 次数据后、传送信息回写后。读取传送信息过程中、传送 1 次数据后和传送信息回写后不释放总线权。

通过总线功能扩展寄存器（BSCEHR）可设定总线权释放时序。详情参阅“9.4.4 总线功能扩展寄存器（BSCEHR）”。各设定对应的总线权释放时序如表 8.11 所示。但，不得在 DTC 启动时更改设定。

时序如图 8.13 所示。

表 8.11　DTC 的总线权释放时序

设定	总线功能扩展寄存器（BSCEHR）的设定	总线权释放时序（○：释放总线权；△：根据 CPU 的执行状态释放总线权；×：不释放总线权）					
	bit15	读取向量后	发行 NOP 时 *	读取传送信息后	1 次传送后	写入传送信息后	
						通常时	连续传送时
设定 1	1	○	○	△	△	○	○
设定 2	0	×	○	×	×	○	○

【注】 *　仅对读取向量后的 CPU 外部存取请求释放总线权。

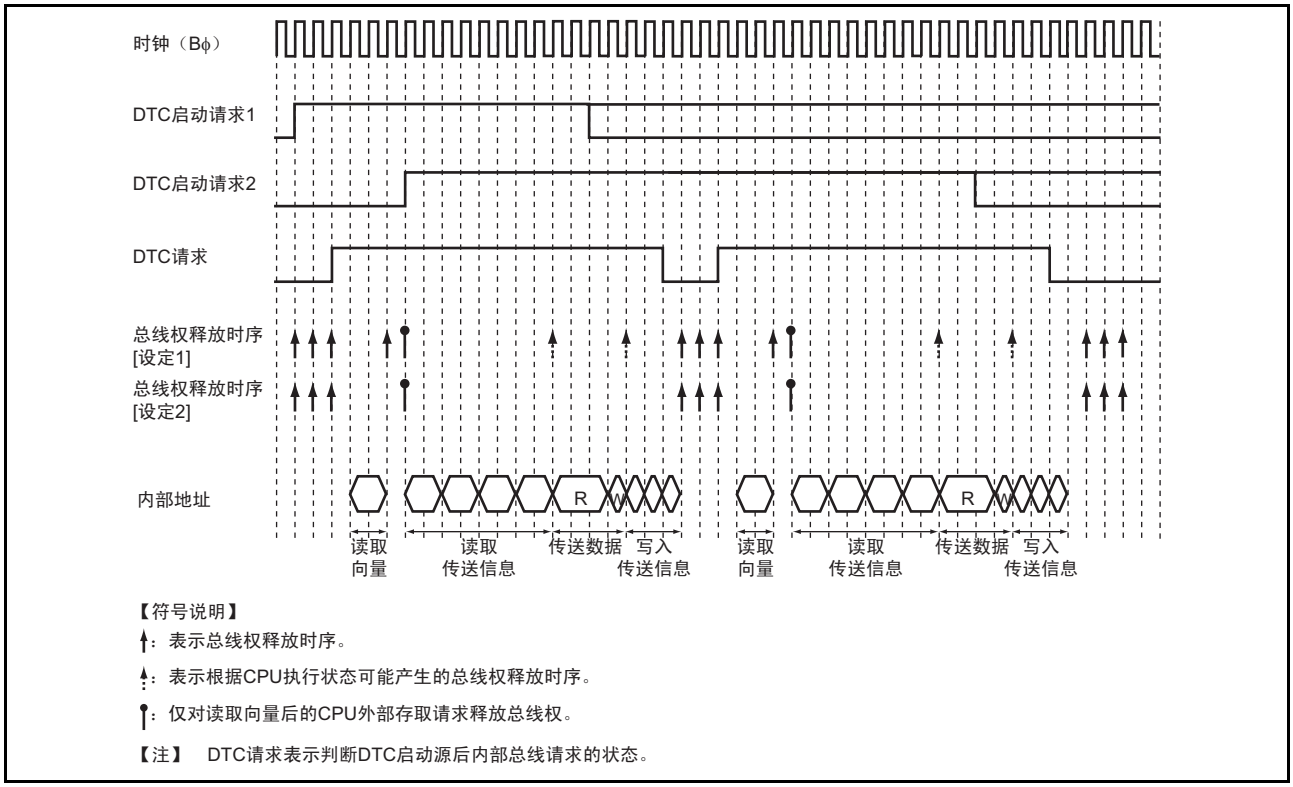


图 8.13　DTC 的运行时序例【普通传送过程中 2 个源竞争时】
（由内部外围模块启动，Iϕ:Bϕ:Pϕ=1:1/2:1/2、
从内部外围模块向内部 RAM 传送数据，传送信息的写入为 3 个状态时）

8.5.10　DTC 启动的优先顺序

在 DTC 启动前产生多个 DTC 启动请求时，从最初发出请求的启动源开始传送。但在 DTC 启动中产生多个 DTC 启动请求时，根据 DTC 启动的优先顺序执行下一个传送。DTC 启动优先顺序的运行例子如图 8.14 所示。

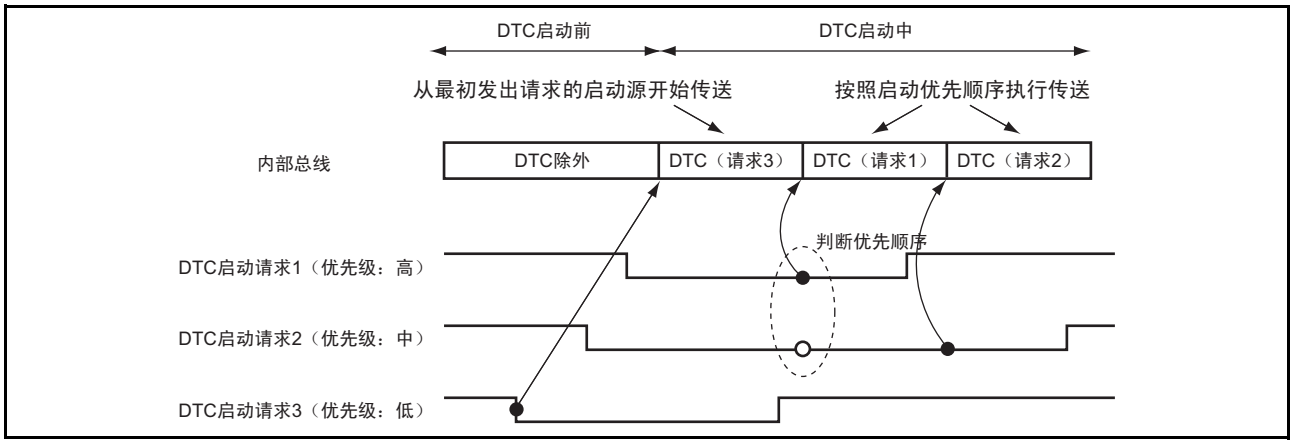
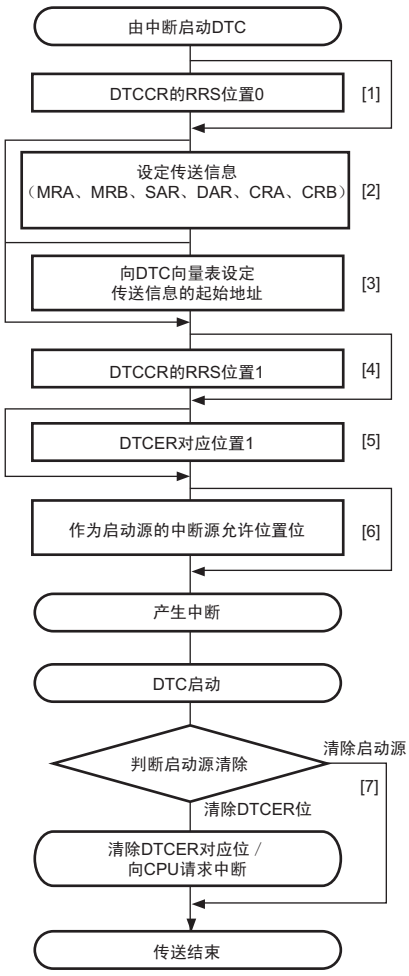


图 8.14　DTC 的优先顺序

8.6　由中断启动 DTC

由中断启动 DTC 的步骤如图 8.15 所示。



[1]　通过将DTCCR的RRS位置0，可复位传送信息跳读标志。本设定后的DTC启动时，不跳读传送信息。更新传送信息时，必须执行本设定。

[2]　必须在数据区域分配传送信息（MRA、MRB、SAR、DAR、CRA、CRB）。传送信息的设定方法参阅“8.2　寄存器说明”。另外，传送信息的分配方法参阅“8.4　传送信息的分配和DTC向量表”。

[3]　必须向DTC向量表设定传送信息的起始地址。DTC向量表的设定方法参阅“8.4　传送信息的的分配和DTC向量表”。

[4]　通过将DTCCR的RRS位置1，可在执行相同中断源引起的连续DTC启动时，跳过第2次以后的传送信息读取周期。RRS位在任何时候都可写入1。但，DTC传送过程中的设定从下一次传送开始有效。

[5]　必须将启动DTC的中断对应的DTCER对应位置1。有关中断及DTCER的关系，参照表8.2。第2次以后的DTC传送时，有时DTCER的对应位会置1。无需重新设定。

[6]　将作为启动源的中断源允许位置位。产生中断源时，启动DTC。有关中断源允许位的设定，参阅作为启动源的模块的设定方法。

[7]　1次数据传送结束后，DTC清除启动源标志或DTCER的对应位并对CPU产生中断。传送后的运行取决于传送信息的设定。详情参阅“8.2　寄存器说明”及图8.4。

图 8.15　由中断启动 DTC 的方法

8.7 DTC 使用例

8.7.1 普通传送

以通过 SCI 接收 128 个字节的数据为例子，说明 DTC 的使用。

1. MRA 设定源地址固定 （SM1=SM0=0）、目标地址递增 （DM1=1、 DM0=0）、普通传送模式 （MD1=MD0=0）及字节长度 （Sz1=Sz0=0）。 DTS 位可设置为任意值。 MRB 通过 1 次中断执行 1 次数据传送 （CHNE=0、 DISEL=0）。 SAR 设定 SCI 的 RDR 的地址， DAR 设定保存数据的 RAM 的起始地址， CRA 设定 128 （H'0080）。 CRB 可设定为任意值。
2. 向 DTC 向量表设定用于 RXI 中断的传送信息的起始地址。
3. DTCER 的对应位置 1。
4. SCI 设定为规定的接收模式。 SCR 的 RIE 位置 1，允许接收结束 （RXI）中断。另外，如果 SCI 接收过程中产生接收错误，则不可执行之后的接收，因此，必须将 CPU 设定为可接受接收错误中断。
5. 每当 SCI 的 1 个字节数据接收结束时， SSR 的 RDRF 标志置 1，并产生 RXI 中断，启动 DTC。通过 DTC 将接收数据从 RDR 传送至 RAM，且 DAR 递增， CRA 递减， RDRF 标志自动清 0。
6. 128 次数据传送结束后， CRA 变为 0， RDRF 标志仍保持 1， DTCE 位清 0，并向 CPU 请求 RXI 中断。必须通过中断处理程序结束处理。

8.7.2 计数器 =0 时的链传送

仅在计数器为 0 时执行第 2 数据传送，并通过重新设定第 1 数据传送，即可执行超过 256 次的重复传送。

举例说明 128K 字节输入缓冲器结构，该输入缓冲器设定为从低位地址 H'0000 开始。计数器 =0 时的链传送如图 8.16 所示。

1. 作为第 1 数据传送，设定用于输入数据的普通传送模式。必须设定传送源地址为固定、 CRA=H'0000 （65,536 次）、 CHNE=1、 CHNS=1、 DISEL=0。
2. 必须在其他区域 （ROM 等）准备第 1 数据的传送目标起始地址 （以 65,536 次传送单位）的高 8 位地址。例如，设定输入缓冲器为 H'200000 ～ H'21FFFF 时，准备 H'21、 H'20。
3. 作为第 2 数据传送，设定用于重新设定第 1 数据的传送目标地址的重复传送模式 （源为重复区域）。传送目标作为第 1 传送信息区域 DAR 的高 8 位。此时必须设定为 CHNE=DISEL=0。如果上述输入缓冲器为 H'200000 ～ H'21FFFF，则必须设定传送计数器 =2。
4. 通过中断执行 65,536 次第 1 数据传送。第 1 数据传送的传送计数器为 0 时，启动第 2 数据传送。将第 1 数据传送的传送源地址高 8 位设定为 H'21。第 1 数据传送的传送目标地址低 16 位的传送计数器为 H'0000。
5. 继续通过中断执行第 1 数据传送指定的 65,536 次第 1 数据传送。第 1 数据传送的传送计数器为 0 时，启动第 2 数据传送。将第 1 数据传送的传送源地址高 8 位设定为 H'20。第 1 数据传送的传送目标地址低 16 位的传送计数器为 H'0000。
6. 无限重复上述的 4. 和 5.。第 2 数据传送为重复传送模式，因此不向 CPU 请求中断。

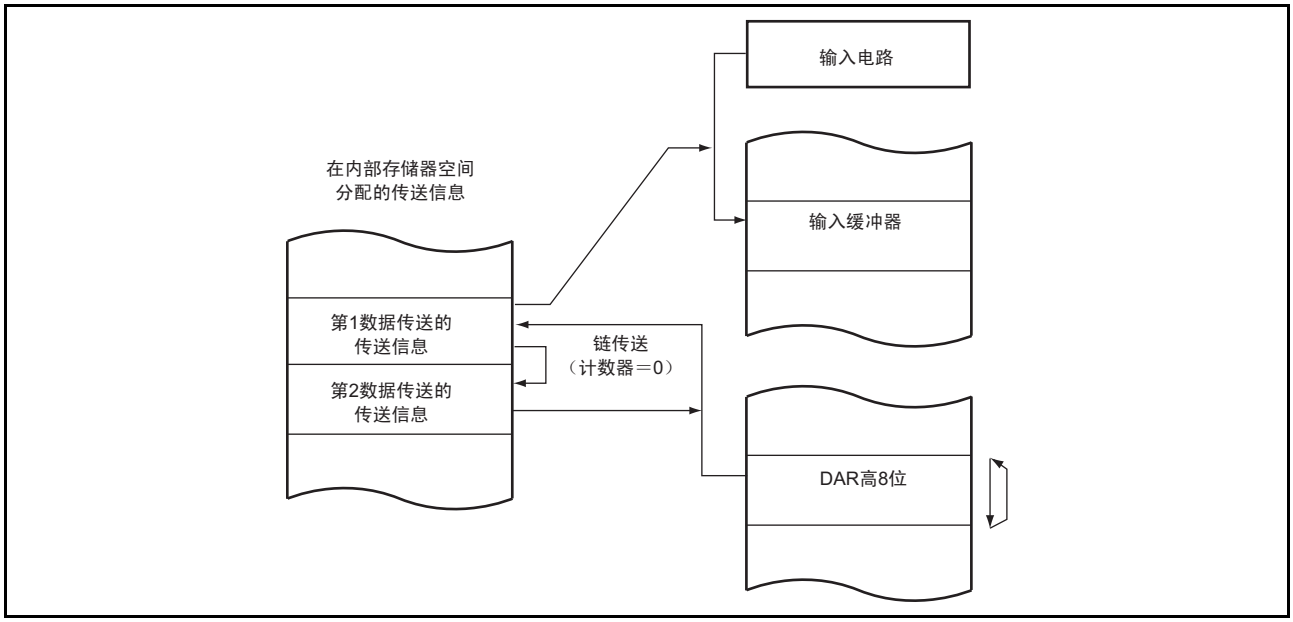


图 8.16　计数器 =0 时的链传送

8.8　中断源

DTC 结束指定次数的数据传送时、 DISEL 位置 1 的 1 次数据传送或 1 次块数据传送结束时，向 CPU 请求中断。启动中断时，产生设定为启动源的中断。向 CPU 产生的中断由 CPU 的屏蔽级和中断控制器的优先级控制。详情参阅 “6.8 通过中断请求信号传送数据”。

8.9　使用时的注意事项

8.9.1　模块待机模式的设定

待机控制寄存器可设定禁止 / 允许 DTC 运行。初始值时， DTC 为禁止运行状态。通过设定为模块待机模式，可禁止 DTC 运行，但不禁止寄存器存取，不得在 DTC 启动时设定模块待机模式。转移至软件待机模式及模块待机模式时，必须清除所有的 DTCER 寄存器。详情参阅 “第 22 章 低功耗模式”。

8.9.2　内部 RAM

传送信息可分配于 RAM。此时，不得将 RAMCR 的 RAME 位清 0。

8.9.3　DTCE 位的设定

通过禁止中断并对该寄存器读取 0 后写入 1，可设定 DTCE 位。不得在 DTC 传送过程中更改 DTCE 位。

8.9.4　链传送

执行链传送时，在传送最后的链接数据时清除启动源或 DTCER。另一方面，读取 / 写入规定的寄存器时，清除 SCI、 SSU、 RCAN-ET 及 A/D 转换器的中断 / 启动源。

8.9.5 传送信息起始地址 / 源地址 / 目标地址

在向量表设定的传送信息起始地址必须为地址 4n。

另外，传送信息必须分配于内部 RAM 或外部存储器空间。

8.9.6 由 DTC 存取 DTC 寄存器

不得使用 DTC 存取 DTC 寄存器。

8.9.7 IRQ 中断作为 DTC 传送源时的注意事项

- 不得由该IRQ中断解除软件待机模式。
- 不得在软件待机时产生的IRQ边沿执行DTC传送。
- 检测出IRQ为低电平时，如果因DTC传送结束向CPU产生中断 （传送计数器=0、或DISEL=1），则在CPU接收中断前，必须保持IRQ引脚为低电平。

8.9.8 SCI 作为 DTC 启动源时的注意事项

由 SCI 的 TXI 中断启动 DTC 时，不得将 SCI 的 TEND 标志用作传送结束标志。

8.9.9 中断源标志的清除

DTC 传送结束后产生的中断，与普通中断相同，必须在中断处理程序过程中清除中断源标志。详情参阅“6.9 使用时的注意事项”。

8.9.10 NMI 中断与 DTC 启动的竞争

NMI 中断与 DTC 启动竞争时，优先执行 NMI 中断，因此， ERR 位置 1，但不启动 DTC。

另外，判断由 NMI 引起的 DTC 停止需要 1×Bcyc+3×Pcyc ；判断由 IRQ 引起的 DTC 启动需要 2×Bcyc ；判断外围模块中断引起的 DTC 启动需要 1×Pcyc。

8.9.11 DTC 启动请求的运行

DTC 接受启动请求后，在回写结束前的一连串 DTC 处理完成前，不接受下一个启动请求。

第 9 章　总线状态控制器 （BSC）

外部总线控制器 （BSC）向连接到外部地址空间的各种存储器和外部器件输出控制信号，因此可直接连接 SRAM 等各种存储器和外部器件。

9.1　特点

1. 外部地址空间
- CS0、 CS1的各空间分别最大支持64M字节
- 每一空间均可指定数据总线宽度 （8位）
- 每一空间均可控制等待状态的插入
- 每次读取/写入存取均可控制等待状态的插入
- 在连续存取为读取-写入 （相同空间或不同空间）、读取-读取 （相同空间或不同空间）及起始周期为写入等5种情况下，可独立设定空闲周期
2. 普通空间接口
- 支持可直接连接SRAM的接口

BSC 框图如图 9.1 所示。

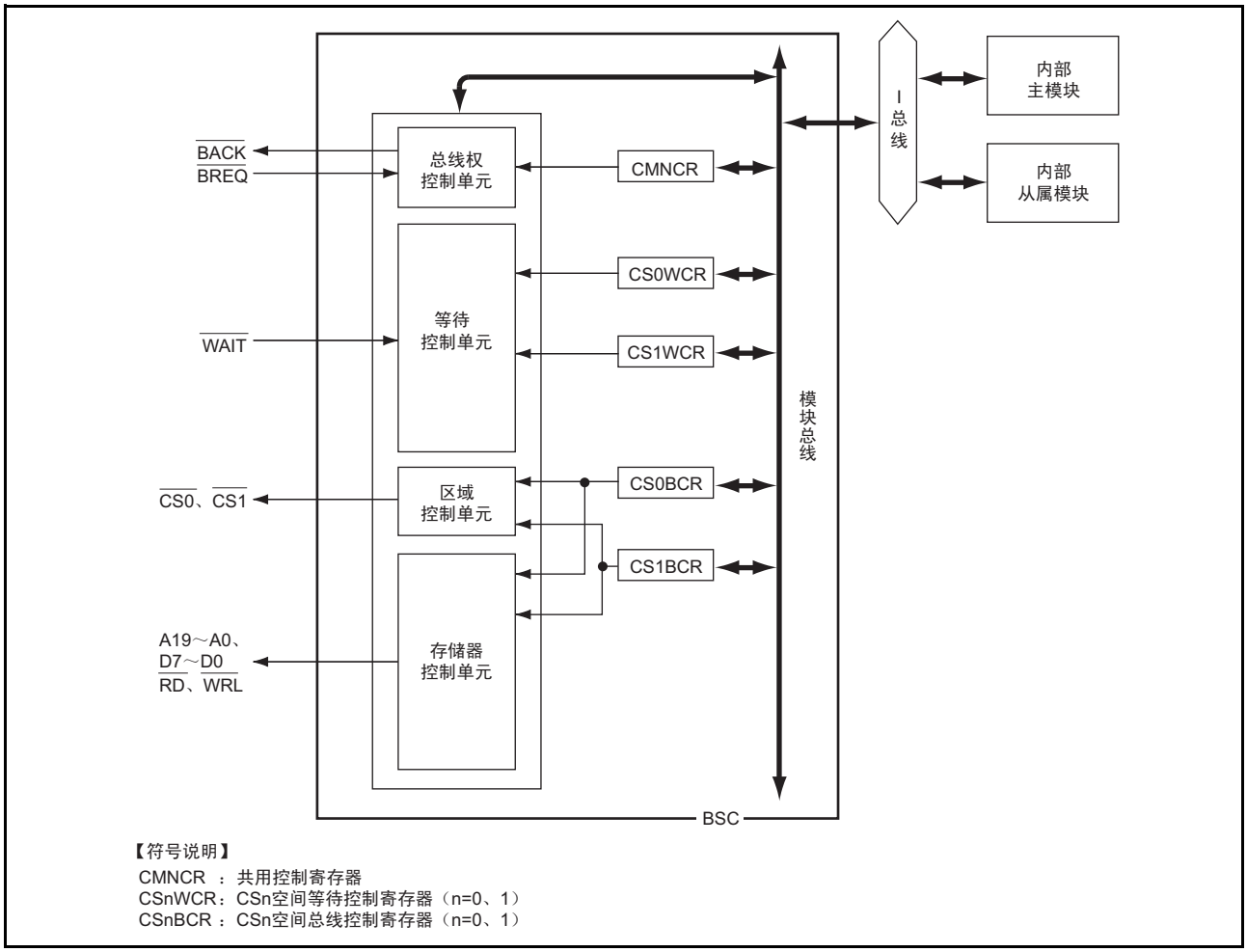


图 9.1　BSC 框图

9.2　输入 / 输出引脚

BSC 引脚结构如表 9.1 所示。

表 9.1　引脚结构

引脚名称	输入 / 输出	功　能
A19 ～ A0	输出	地址总线
D7 ～ D0	输入 / 输出	数据总线
$\overline{CS0}$、$\overline{CS1}$	输出	片选
$\overline{RD}$	输出	读取脉冲信号 （读取数据输出允许信号）
$\overline{WRL}$	输出	D7 ～ D0 相应字节的写入指示
$\overline{WAIT}$	输入	输入外部等待
$\overline{BREQ}$	输入	输入总线权请求
$\overline{BACK}$	输出	输出总线使用允许

9.3　区域概要

9.3.1　空间分割

本 LSI 有 32 位地址空间体系结构。

如表 9.2 ～表 9.5 所示，本 LSI 可分别将各种存储器连接到 2 个空间，并输出相应的片选信号 （$\overline{CS0}$、$\overline{CS1}$）。$\overline{CS0}$ 在存取区域 0 时有效。

9.3.2　地址映射

外部地址空间共有 128M 字节，分为 2 个空间使用，对各空间分别指定连接存储器的种类及数据总线宽度。所有空间的地址映射如表 9.2 ～表 9.5 所示。

表 9.2　(1) 地址映射 （内部 ROM256KB/ 内部 RAM12KB 版、内部 ROM 有效模式时）

地址	空间	存储器种类	长度	总线宽度
H'00000000 ～ H'0003FFFF	内部 ROM		256K 字节	32 位
H'00040000 ～ H'01FFFFFF	保留			
H'02000000 ～ H'03FFFFFF	CS0 空间	普通空间	32M 字节	8 位
H'04000000 ～ H'07FFFFFF	CS1 空间	普通空间	64M 字节	8 位
H'08000000 ～ H'FFFF8FFF	保留			
H'FFFF9000 ～ H'FFFFBFFF	内部 RAM		12K 字节	32 位
H'FFFFC000 ～ H'FFFFFFFF	内部外围模块		16K 字节	8/16 位

【注】 不得存取保留空间，如果存取，则无法保证正常运行。在单芯片模式，不可使用内部 ROM、内部 RAM、内部外围模块以外的空间。

表 9.2　(2) 地址映射 （内部 ROM256KB/ 内部 RAM12KB 版、内部 ROM 无效模式时）

地址	空间	存储器种类	长度	总线宽度
H'00000000 ～ H'03FFFFFF	CS0 空间	普通空间	64M 字节	8 位
H'04000000 ～ H'07FFFFFF	CS1 空间	普通空间	64M 字节	8 位
H'08000000 ～ H'FFFF8FFF	保留			
H'FFFF9000 ～ H'FFFFBFFF	内部 RAM		12K 字节	32 位
H'FFFFC000 ～ H'FFFFFFFF	内部外围模块		16K 字节	8/16 位

【注】 不得存取保留空间，如果存取，则无法保证正常运行。

表 9.3　(1) 地址映射 （内部 ROM256KB/ 内部 RAM16KB 版、内部 ROM 有效模式时）

地址	空间	存储器种类	长度	总线宽度
H'00000000 ～ H'0003FFFF	内部 ROM		256K 字节	32 位
H'00040000 ～ H'01FFFFFF	保留			
H'02000000 ～ H'03FFFFFF	CS0 空间	普通空间	32M 字节	8 位
H'04000000 ～ H'07FFFFFF	CS1 空间	普通空间	64M 字节	8 位
H'08000000 ～ H'FFFF7FFF	保留			
H'FFFF8000 ～ H'FFFFBFFF	内部 RAM		16K 字节	32 位
H'FFFFC000 ～ H'FFFFFFFF	内部外围模块		16K 字节	8/16 位

【注】 不得存取保留空间，如果存取，则无法保证正常运行。在单芯片模式，不可使用内部 ROM、内部 RAM、内部外围模块以外的空间。

表 9.3　(2) 地址映射 （内部 ROM256KB/ 内部 RAM16KB 版、内部 ROM 无效模式时）

地址	空间	存储器种类	长度	总线宽度
H'00000000 ～ H'03FFFFFF	CS0 空间	普通空间	64M 字节	8 位
H'04000000 ～ H'07FFFFFF	CS1 空间	普通空间	64M 字节	8 位
H'08000000 ～ H'FFFF7FFF	保留			
H'FFFF8000 ～ H'FFFFBFFF	内部 RAM		16K 字节	32 位
H'FFFFC000 ～ H'FFFFFFFF	内部外围模块		16K 字节	8/16 位

【注】 不得存取保留空间，如果存取，则无法保证正常运行。

表 9.4　(1) 地址映射 （内部 ROM384KB/ 内部 RAM16KB 版、内部 ROM 有效模式时）

地址	空间	存储器种类	长度	总线宽度
H'00000000 ～ H'0005FFFF	内部 ROM		384K 字节	32 位
H'00060000 ～ H'01FFFFFF	保留			
H'02000000 ～ H'03FFFFFF	CS0 空间	普通空间	32M 字节	8 位
H'04000000 ～ H'07FFFFFF	CS1 空间	普通空间	64M 字节	8 位
H'08000000 ～ H'FFFF7FFF	保留			
H'FFFF8000 ～ H'FFFFBFFF	内部 RAM		16K 字节	32 位
H'FFFFC000 ～ H'FFFFFFFF	内部外围模块		16K 字节	8/16 位

【注】 不得存取保留空间，如果存取，则无法保证正常运行。在单芯片模式，不可使用内部 ROM、内部 RAM、内部外围模块以外的空间。

表 9.4　(2) 地址映射 （内部 ROM384KB/ 内部 RAM16KB 版、内部 ROM 无效模式时）

地址	空间	存储器种类	长度	总线宽度
H'00000000 ～ H'03FFFFFF	CS0 空间	普通空间	64M 字节	8 位
H'04000000 ～ H'07FFFFFF	CS1 空间	普通空间	64M 字节	8 位
H'08000000 ～ H'FFFF7FFF	保留			
H'FFFF8000 ～ H'FFFFBFFF	内部 RAM		16K 字节	32 位
H'FFFFC000 ～ H'FFFFFFFF	内部外围模块		16K 字节	8/16 位

【注】　不得存取保留空间，如果存取，则无法保证正常运行。

表 9.5　(1) 地址映射 （内部 ROM512KB/ 内部 RAM16KB 版、内部 ROM 有效模式时）

地址	空间	存储器种类	长度	总线宽度
H'00000000 ～ H'0007FFFF	内部 ROM		512K 字节	32 位
H'00080000 ～ H'01FFFFFF	保留			
H'02000000 ～ H'03FFFFFF	CS0 空间	普通空间	32M 字节	8 位
H'04000000 ～ H'07FFFFFF	CS1 空间	普通空间	64M 字节	8 位
H'08000000 ～ H'FFFF7FFF	保留			
H'FFFF8000 ～ H'FFFFBFFF	内部 RAM		16K 字节	32 位
H'FFFFC000 ～ H'FFFFFFFF	内部外围模块		16K 字节	8/16 位

【注】　不得存取保留空间，如果存取，则无法保证正常运行。在单芯片模式，不可使用内部 ROM、内部 RAM、内部外围模块以外的空间。

表 9.5　(2) 地址映射 （内部 ROM512KB/ 内部 RAM16KB 版、内部 ROM 无效模式时）

地址	空间	存储器种类	长度	总线宽度
H'00000000 ～ H'03FFFFFF	CS0 空间	普通空间	64M 字节	8 位
H'04000000 ～ H'07FFFFFF	CS1 空间	普通空间	64M 字节	8 位
H'08000000 ～ H'FFFF7FFF	保留			
H'FFFF8000 ～ H'FFFFBFFF	内部 RAM		16K 字节	32 位
H'FFFFC000 ～ H'FFFFFFFF	内部外围模块		16K 字节	8/16 位

【注】　不得存取保留空间，如果存取，则无法保证正常运行。

9.4　寄存器说明

BSC 有以下寄存器。有关寄存器的地址及各处理状态的寄存器状态，详情参阅 “第 24 章 寄存器一览表”。

与存储器的接口设定结束前，不得存取 CS0 以外的空间。

表 9.6　寄存器结构

寄存器名称	简称	R/W	初始值	地址	存取长度
共用控制寄存器	CMNCR	R/W	H'00001010	H'FFFFF000	32
CS0 空间总线控制寄存器	CS0BCR	R/W	H'36DB0600	H'FFFFF004	32
CS1 空间总线控制寄存器	CS1BCR	R/W	H'36DB0600	H'FFFFF008	32
CS0 空间等待控制寄存器	CS0WCR	R/W	H'00000500	H'FFFFF028	32
CS1 空间等待控制寄存器	CS1WCR	R/W	H'00000500	H'FFFFF02C	32
总线功能扩展寄存器	BSCEHR	R/W	H'0000	H'FFFFE89A	8、16

9.4.1　共用控制寄存器 （CMNCR）

CMNCR 为执行各区域通用控制的 32 位寄存器。

寄存器初始设定结束前，不得存取区域 0 以外的外部存储器。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

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

位	位名称	初始值	R/W	说　明
31 ～ 13	—	均为 0	R	保留位 读取值、写入值总是为 0。
12	—	1	R	保留位 读取值、写入值总是为 1。
11 ～ 5	—	均为 0	R	保留位 读取值、写入值总是为 0。
4	—	1	R	保留位 读取值、写入值总是为 1。
3、2	—	均为 0	R	保留位 读取值、写入值总是为 0。
1	HIZMEM	0	R/W	Hi-Z 存储器控制 本位指定 A19 ～ A0、$\overline{\text{CSn}}$、$\overline{\text{WRL}}$、$\overline{\text{RD}}$ 在软件待机模式的引脚状态。释放总线时，与本位无关，上述引脚为高阻抗。 0：在软件待机模式时为高阻抗状态 1：在软件待机模式时为驱动状态
0	—	0	R	保留位 读取值、写入值总是为 0。

9.4.2　CSn 空间总线控制寄存器 （CSnBCR）（n=0、 1）

CSnBCR 设定各空间的数据总线宽度及存取周期之间的等待数。

寄存器初始设定结束前，不得存取区域 0 以外的外部存储器。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	IWW[1:0]		-	IWRWD[1:0]		-	IWRWS[1:0]		-	IWRRD[1:0]		-	IWRRS[1:0]	
初始值：	0	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1
R/W：	R	R	R/W	R/W	R	R/W	R/W	R	R/W	R/W	R	R/W	R/W	R	R/W	R/W

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

位	位名称	初始值	R/W	说　明
31、 30	—	均为 0	R	保留位 读取值、写入值总是为 0。
29、 28	IWW[1:0]	11	R/W	指定写入－读取 / 写入－写入周期间的空闲周期 存取连接到空间的存储器后需插入的空闲周期数，由本位指定。对象周期为写入－读取周期和写入－写入周期。 00：无空闲周期 01：插入 1 个空闲周期 10：插入 2 个空闲周期 11：插入 4 个空闲周期
27	—	0	R	保留位 读取值、写入值总是为 0。
26、 25	IWRWD[1:0]	11	R/W	指定不同空间读取－写入周期间的空闲周期 存取连接到空间的存储器后需插入的空闲周期数，由本位指定。对象周期为不同空间的连续读取－写入周期。 00：无空闲周期 01：插入 1 个空闲周期 10：插入 2 个空闲周期 11：插入 4 个空闲周期
24	—	0	R	保留位 读取值、写入值总是为 0。
23、 22	IWRWS[1:0]	11	R/W	指定相同空间读取－写入周期间的空闲周期 存取连接到空间的存储器后需插入的空闲周期数，由本位指定。对象周期为相同空间的连续读取－写入周期。 00：无空闲周期 01：插入 1 个空闲周期 10：插入 2 个空闲周期 11：插入 4 个空闲周期
21	—	0	R	保留位 读取值、写入值总是为 0。
20、 19	IWRRD[1:0]	11	R/W	指定不同空间读取－读取周期间的空闲周期 存取连接到空间的存储器后需插入的空闲周期数，由本位指定。对象周期为不同空间的连续读取－读取周期。 00：无空闲周期 01：插入 1 个空闲周期 10：插入 2 个空闲周期 11：插入 4 个空闲周期

位	位名称	初始值	R/W	说　明
18	—	0	R	保留位 读取值、写入值总是为 0。
17、16	IWRRS[1:0]	11	R/W	指定相同空间读取－读取周期间的空闲周期 存取连接到空间的存储器后需插入的空闲周期数，由本位指定。对象周期为相同空间的连续读取－读取周期。 00：无空闲周期 01：插入 1 个空闲周期 10：插入 2 个空闲周期 11：插入 4 个空闲周期
15 ～ 11	—	均为 0	R	保留位 读取值、写入值总是为 0。
10、9	BSZ[1:0]	11	R/W	指定数据总线宽度 本位指定空间的数据总线宽度。内部 ROM 有效时，为了将数据总线宽度指定为 8 位，存取 CSn 空间前必须向本位写入 B'01。 【注】内部 ROM 无效时，由外部输入引脚设定区域 0 的数据总线宽度。忽略 CS0BCR 的 BSZ1、 0 位的设定。
8 ～ 0	—	均为 0	R	保留位 读取值、写入值总是为 0。

9.4.3　CSn 空间等待控制寄存器 （CSnWCR）（n=0、 1）

CSnWCR 设定有关存储器存取的各种等待周期，必须在存取对象区域前设定。另外，在设定 CSnBCR 寄存器后，再设定 CSnWCR 寄存器。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	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	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	位名称	初始值	R/W	说　明
31 ～ 19	—	均为 0	R	保留位 读取值、写入值总是为 0。
18 ～ 16	WW[2:0]	000	R/W	写入存取等待周期数 本位指定写入存取所需的周期数。 000：与 WR3 ～ WR0 设定 （读取存取等待）的周期相同 001：0 个周期 010：1 个周期 011：2 个周期 100：3 个周期 101：4 个周期 110：5 个周期 111：6 个周期
15 ～ 13	—	均为 0	R	保留位 读取值、写入值总是为 0。
12、 11	SW[1:0]	00	R/W	地址、 $\overline{\text{CSn}}$ 有效 →$\overline{\text{RD}}$、 $\overline{\text{WRL}}$ 有效延迟周期数 本位指定地址、 $\overline{\text{CSn}}$ 有效到 $\overline{\text{RD}}$、 $\overline{\text{WRL}}$ 有效的延迟周期数。 00：0.5 个周期 01：1.5 个周期 10：2.5 个周期 11：3.5 个周期
10 ～ 7	WR[3:0]	1010	R/W	读取存取等待周期数 本位指定读取存取所需的等待周期数。 0000：0 个周期 0001：1 个周期 0010：2 个周期 0011：3 个周期 0100：4 个周期 0101：5 个周期 0110：6 个周期 0111：8 个周期 1000：10 个周期 1001：12 个周期 1010：14 个周期 1011：18 个周期 1100：24 个周期 1101：保留 （不可设定） 1110：保留 （不可设定） 1111：保留 （不可设定）
6	WM	0	R/W	指定外部等待屏蔽 本位指定外部等待输入有效还是忽略。即使存取等待周期数为 0，本位设定仍有效。 0：外部等待输入有效 1：忽略外部等待输入
5 ～ 2	—	均为 0	R	保留位 读取值、写入值总是为 0。
1、 0	HW[1:0]	00	R/W	$\overline{\text{RD}}$、 $\overline{\text{WRL}}$ 无效 → 地址、 $\overline{\text{CSn}}$ 无效延迟周期数 本位指定 $\overline{\text{RD}}$、 $\overline{\text{WRL}}$ 无效到地址、 $\overline{\text{CSn}}$ 无效的延迟周期数。 00：0.5 个周期 01：1.5 个周期 10：2.5 个周期 11：3.5 个周期

9.4.4　总线功能扩展寄存器 （BSCEHR）

BSCEHR 为 16 位寄存器，设定 DTC 的总线权释放时序。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTLOCK	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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	R	R	R

位	位名称	初始值	R/W	说　明
15	DTLOCK	0	R/W	DTC 锁定允许 选择 DTC 释放总线权的时序。 0：读取向量后发行 NOP 时及传送信息回写后，释放总线权。 1：读取向量后、读取向量后发行 NOP 时、读取传送信息后、1 次数据传送后及传送信息回写后，释放总线权。
14 ～ 0	—	均为 0	R	保留位 读取值、写入值总是为 0。

9.5　运行说明

9.5.1　字节序 / 存取长度和数据对齐

本 LSI 支持高位字节 （MSB）为地址 0 的大端法字节数据排列。

数据总线宽度为 8 位。根据各器件的数据总线宽度去对齐数据。因此，要从 8 位宽度的器件读取长字数据，必须读取 4 次。在本 LSI 的各接口之间，自动对齐数据并变换数据长度。

器件的数据长度与存取单位的关系如表 9.7 所示。

表 9.7　8 位外部器件的存取和数据对齐

操作		数据总线	选通信号
		D7 ～ D0	$\overline{\text{WRL}}$
字节存取地址 0		数据 7 ～ 0	有效
字节存取地址 1		数据 7 ～ 0	有效
字节存取地址 2		数据 7 ～ 0	有效
字节存取地址 3		数据 7 ～ 0	有效
字存取地址 0	第 1 次 （地址 0）	数据 15 ～ 8	有效
	第 2 次 （地址 1）	数据 7 ～ 0	有效
字存取地址 2	第 1 次 （地址 2）	数据 15 ～ 8	有效
	第 2 次 （地址 3）	数据 7 ～ 0	有效
长字存取地址 0	第 1 次 （地址 0）	数据 31 ～ 24	有效
	第 2 次 （地址 1）	数据 23 ～ 16	有效
	第 3 次 （地址 2）	数据 15 ～ 8	有效
	第 4 次 （地址 3）	数据 7 ～ 0	有效

9.5.2　普通空间接口

(1)　基本时序

普通空间存取时，主要考虑与无字节选择引脚的 SRAM 直接连接，而输出选通信号。普通空间存取的基本时序如图 9.2 所示。无等待的普通存取以 2 个周期结束。

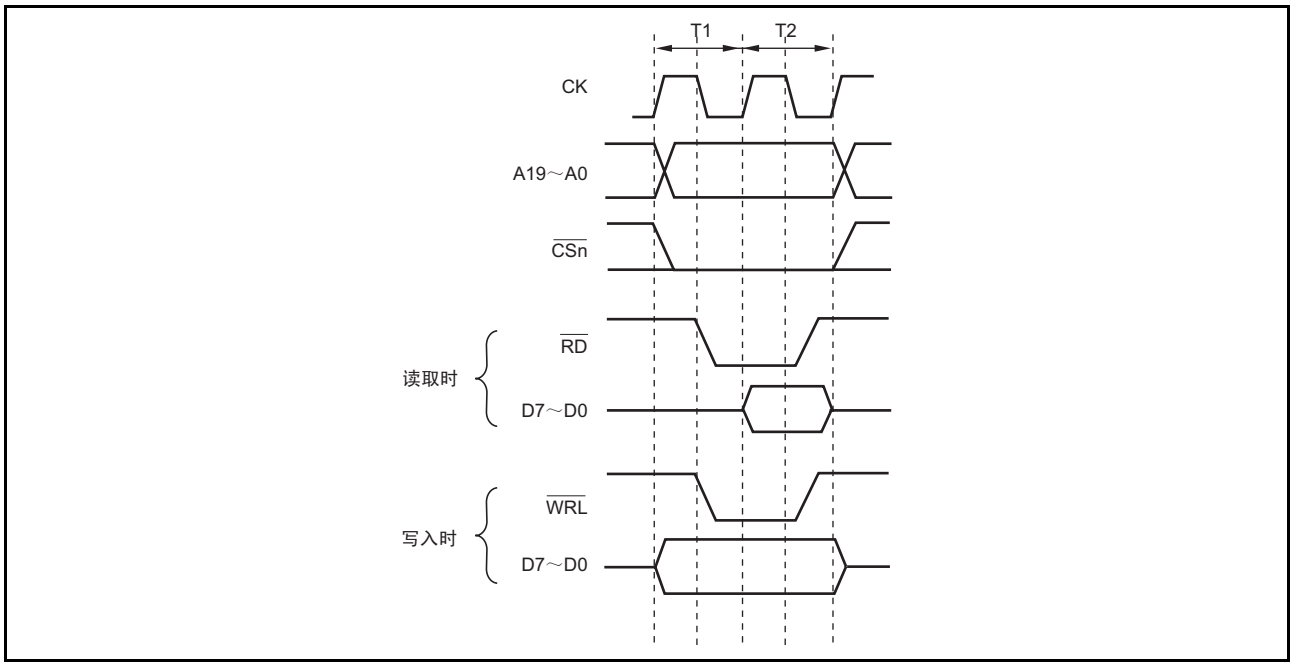


图 9.2　普通空间基本存取 （存取等待 0）

读取时，不指定外部总线的存取长度。向地址的最低位输出正确的存取起始地址，但因未指定存取长度，因此总是读取 8 位。写入时，仅执行写入字节的 $\overline{\text{WRL}}$ 信号有效。

在数据总线设置缓冲器时，必须使用 $\overline{\text{RD}}$ 执行读取数据的输出控制。

普通空间连续存取的例子如图 9.3、图 9.4 所示。 CSnWCR.WM 位设定为 0 时，为了评估外部等待而插入 1 个周期 Tnop （图 9.3）；但 CSnWCR.WM 位设定为 1 时，可忽略外部等待并禁止插入 Tnop 周期 （图 9.4）。

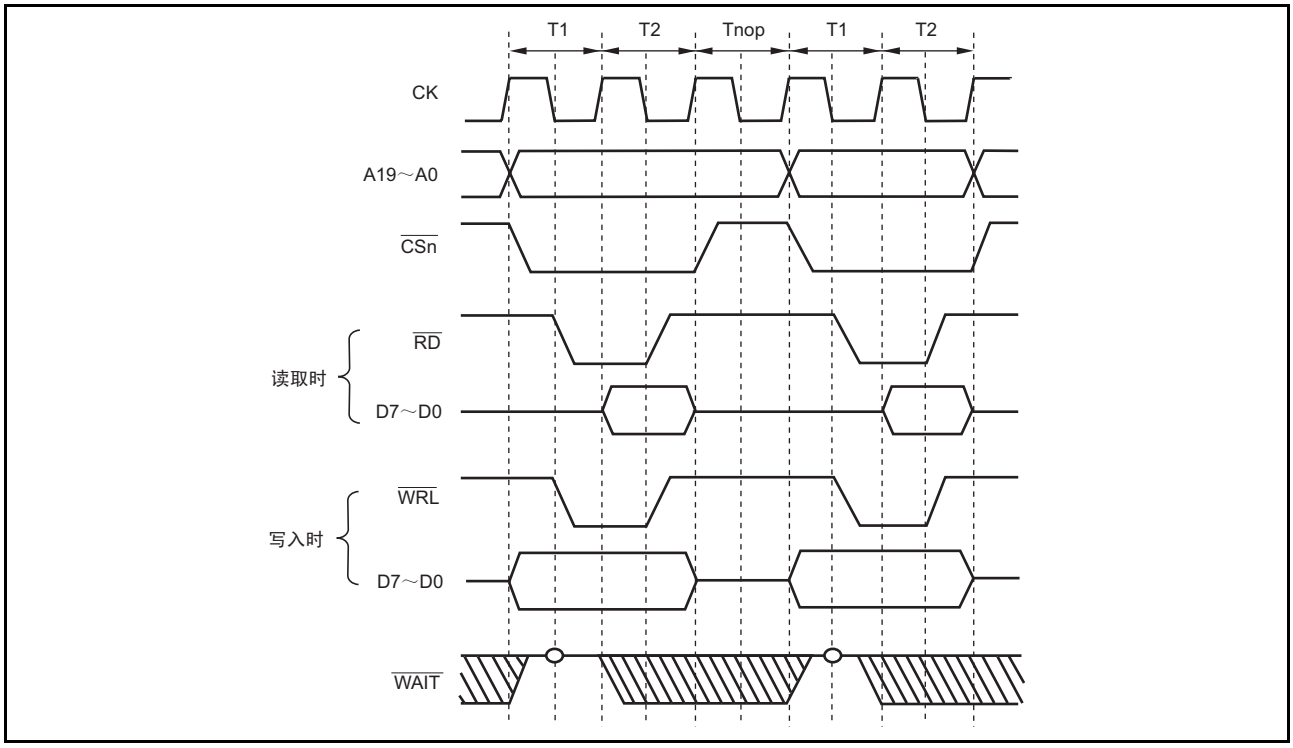


图 9.3 普通空间连续存取例 1
总线宽度 8 位、长字存取、 CSnWCR.WM 位 =0 （存取等待 0、周期间的等待 0）

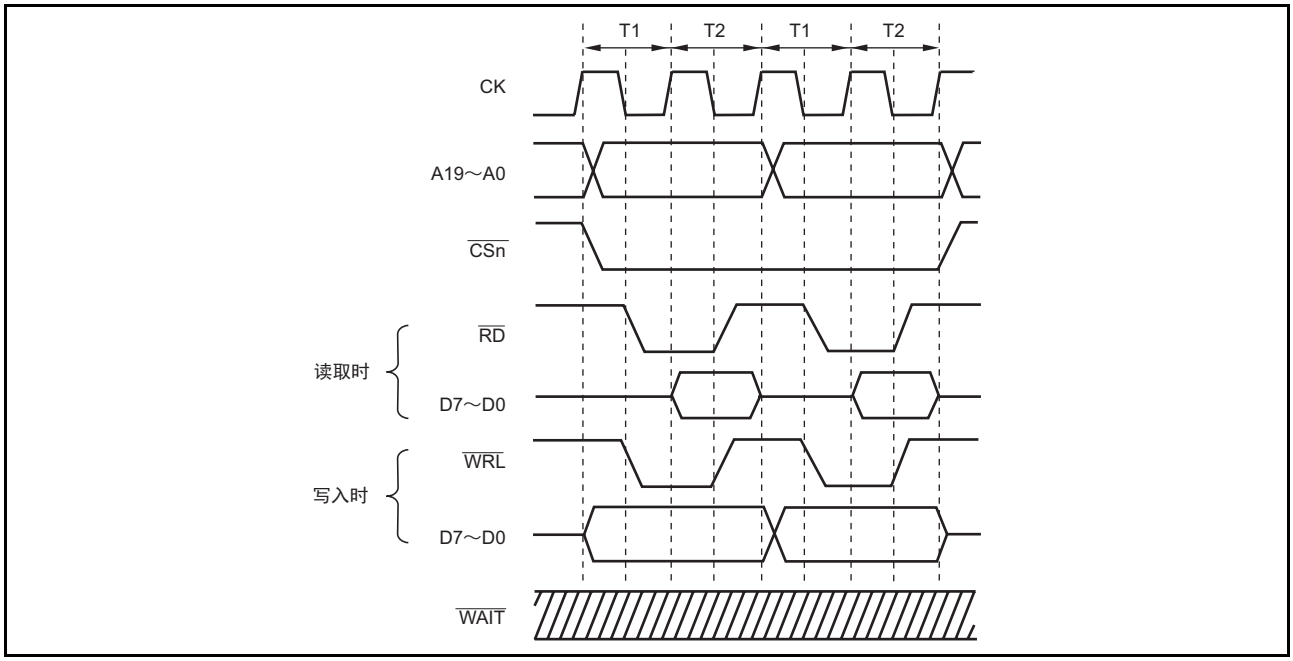


图 9.4 普通空间连续存取例 2
总线宽度 8 位、长字存取、 CSnWCR.WM 位 =1 （存取等待 0、周期间的等待 0）

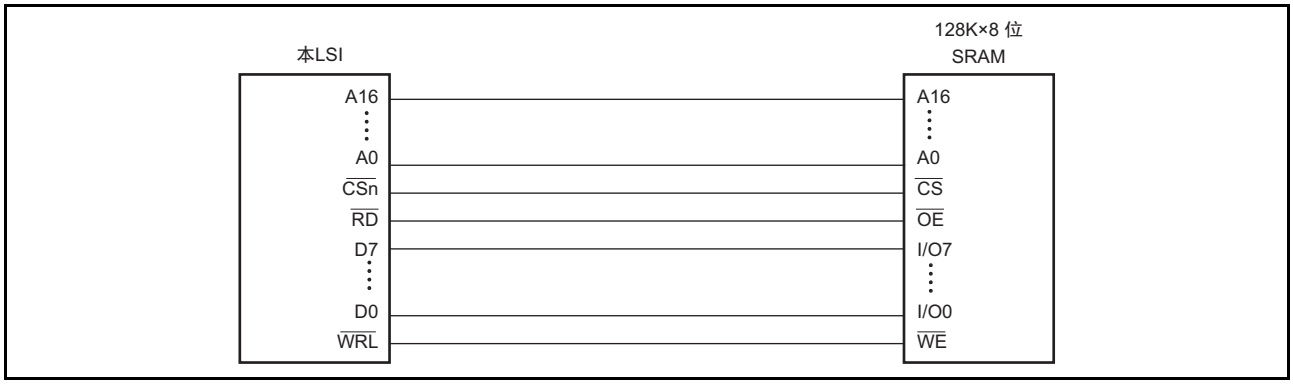


图 9.5　8 位数据长度 SRAM 连接例

9.5.3　　存取等待控制

通过设定 CSnWCR 的 WR3、 WR2、 WR1 及 WR0 位，可控制插入普通空间存取的等待周期。读取存取与写入存取过程中，可独立插入等待周期。如图 9.6 所示，普通空间存取过程中， Tw 周期仅插入作为等待周期指定的周期数。

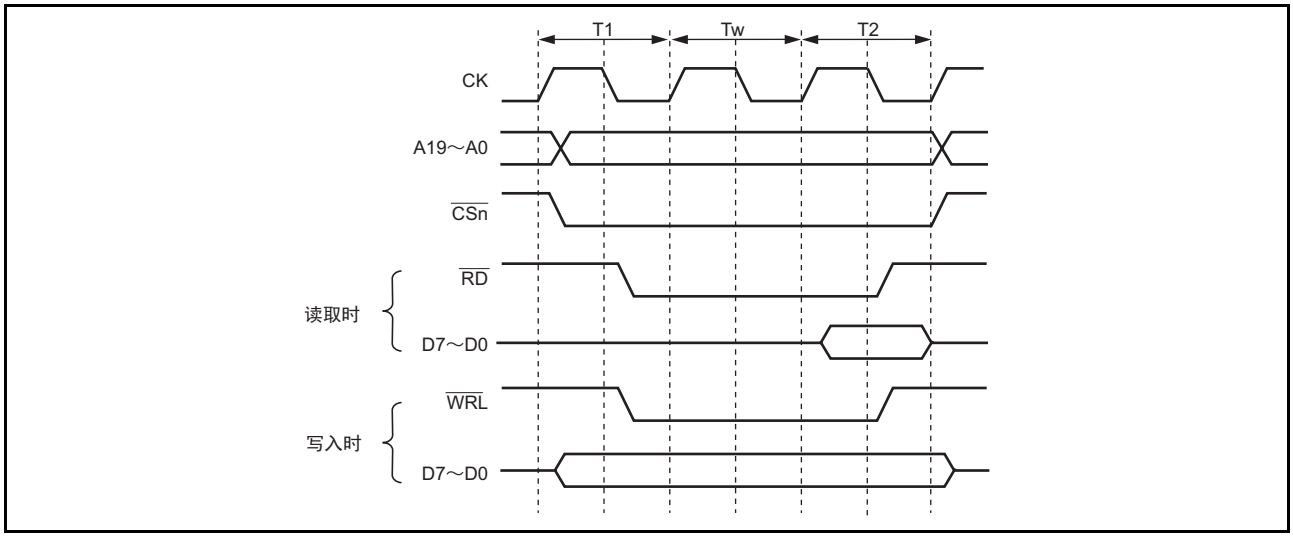


图 9.6　普通空间存取的等待时序 （仅限软件等待）

CSnWCR 的 WM 位为 0 时，也对外部的等待输入 $\overline{\text{WAIT}}$ 信号进行采样。 $\overline{\text{WAIT}}$ 信号的采样如图 9.7 所示。指定 2 个周期的等待作为软件等待。 $\overline{\text{WAIT}}$ 信号从 T1 或 Tw 周期转移至 T2 周期时，在 CK 的下降沿采样。

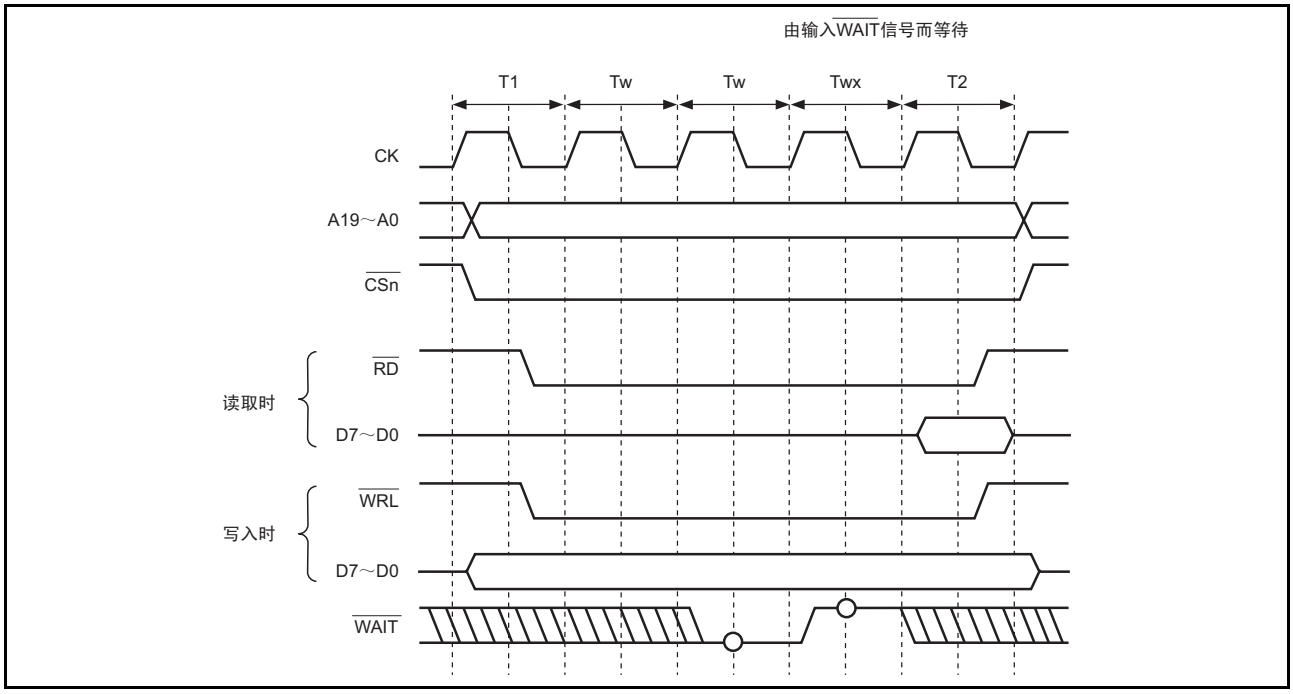


图 9.7　普通空间存取的等待时序 （由 WAIT 信号插入等待）

9.5.4　CSn 有效周期扩展

通过设定 CSnWCR 的 SW1 和 SW0 位，可指定从 CSn 有效到 RD 和 WRL 有效的周期数。通过设定 HW1 和 HW0 位，可指定从 RD 和 WRL 无效到 CSn 无效的周期数。由此，外部器件可与各种接口对应。如图 9.8 所示，将 Th 和 Tf 周期分别追加到通常的周期前后。这些周期中，除 RD 和 WRL 无效外，其他均有效。另外，数据输出延长到 Tf 周期，因此对写入运行较慢的器件等有效。

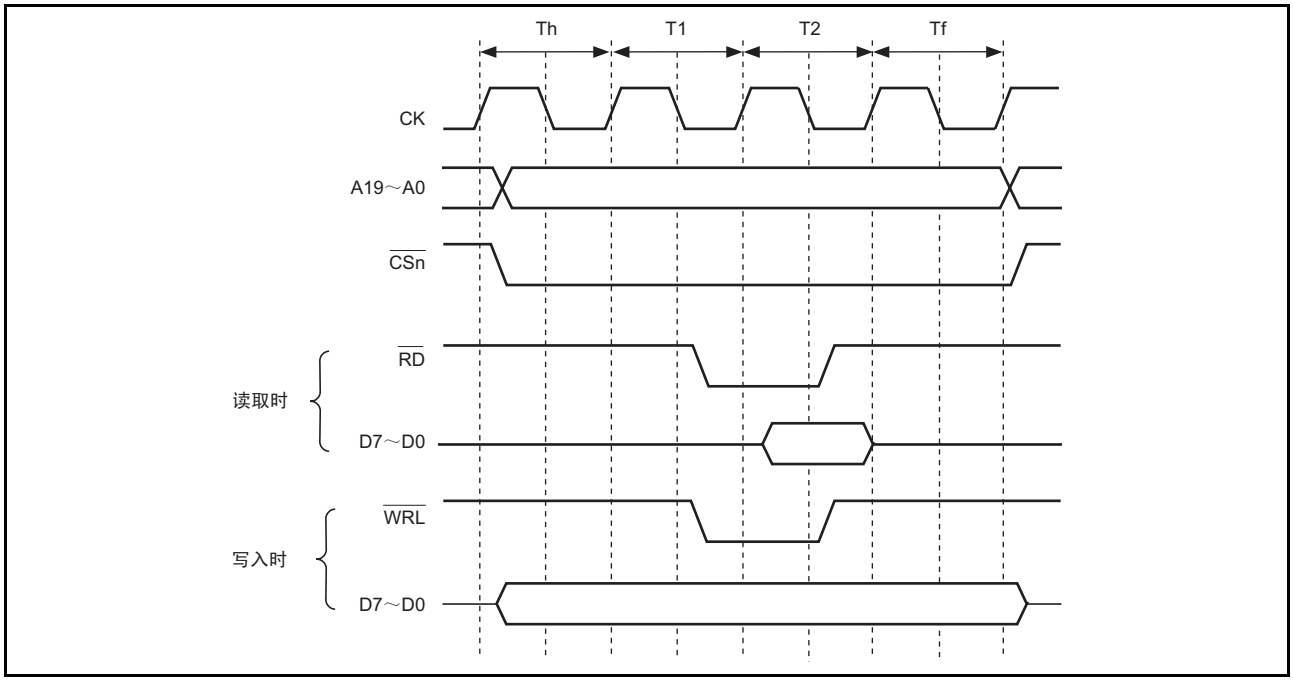


图 9.8　CSn 有效期间扩展

9.5.5 存取周期间的等待

由于 LSI 的工作频率提高，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，导致与下一个器件的数据输出发生冲突，从而降低器件的可靠性或引起误动作。为了防止发生这样的情况，设置了在连续存取之间插入存取周期间的等待，避免数据冲突的功能。

由 CSnBCR 的 IWW[1:0] 位、IWRWD[1:0] 位、IWRWS[1:0] 位、IWRRD[1:0] 位、IWRRS[1:0] 位指定存取周期间的等待周期数。下列条件时，可插入存取周期间的等待 （空闲周期）：

1. 连续存取为写入－读取、写入－写入时
2. 连续存取为不同空间且读取－写入时
3. 连续存取为相同空间且读取－写入时
4. 连续存取为不同空间且读取－读取时
5. 连续存取为相同空间且读取－读取时

除上述所示存取周期间的等待空闲周期外，为确保与内部总线的接口和多路复用引脚 （$\overline{\text{WRL}}$）的最小脉宽，有时需插入空闲周期。

6. 伴随内部总线接口的外部总线的空闲周期
   (a) 在外部总线空闲或读取周期后的写入存取前，插入1个空闲周期。
   (b) 在外部总线的读取周期结束时，为了将读取数据传送至内部总线，插入1个空闲周期。
   在读取周期后紧接着的写入周期，插入包含上述 （a）的2～3个空闲周期。

插入的最小空闲周期数如表 9.8、表 9.9 所示。表中的 “CSnBCR 的空闲设定”表示由 IWW、 IWRWD、IWRWS、 IWRRD、 IWRRS 设定的空闲周期数。

表 9.8　普通空间接口 CPU 存取之间的最小空闲周期数

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

【注】 表中的最小空闲周期数依次表示 Iϕ:Bϕ 为 4:1、 3:1、 2:1、 1:1 时的情况。

*1　总线宽度为 8 位，执行 16 位存取时，地址 0 字节存取和地址 1 字节存取之间的最小空闲周期数；总线宽度为 8 位，执行 32 位存取时，地址 0 字节存取、地址 1 字节存取、地址 2 字节存取及地址 3 字节存取之间的最小空闲周期数

*2　上述以外

表 9.9　普通空间接口 DTC 传送时的存取之间的最小空闲周期数

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

【注】 DTC 为 Bϕ 运行，因此与时钟比的更改无关。

*1　8 位总线宽度时的 16 位存取过程中，地址 0 字节存取和地址 1 字节存取之间的最小空闲周期数； 8 位总线宽度时的 32 位存取过程中，地址 0 字节存取、地址 1 字节存取、地址 2 字节存取及地址 3 字节存取之间的最小空闲周期数

*2　上述以外

9.5.6　总线仲裁

本 LSI 在通常状态占有总线权，在接收外部器件的总线权请求后释放总线。另外，本 LSI 内部也有 CPU、AUD 及 DTC 等 3 个总线主控器，对总线主控器转让总线权的优先顺序如下：

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

但，在 DTC 或 AUD 请求总线权的状态， CPU 不会连续获得总线权。

CPU 存取外部空间的请求处理如下：

CPU 存取外部空间的过程中，按 DTC、 AUD 的顺序产生启动请求时，先执行 DTC 传送。 CPU 存取外部空间的过程中，产生 AUD/DTC 竞争时的总线仲裁如图 9.9 所示。

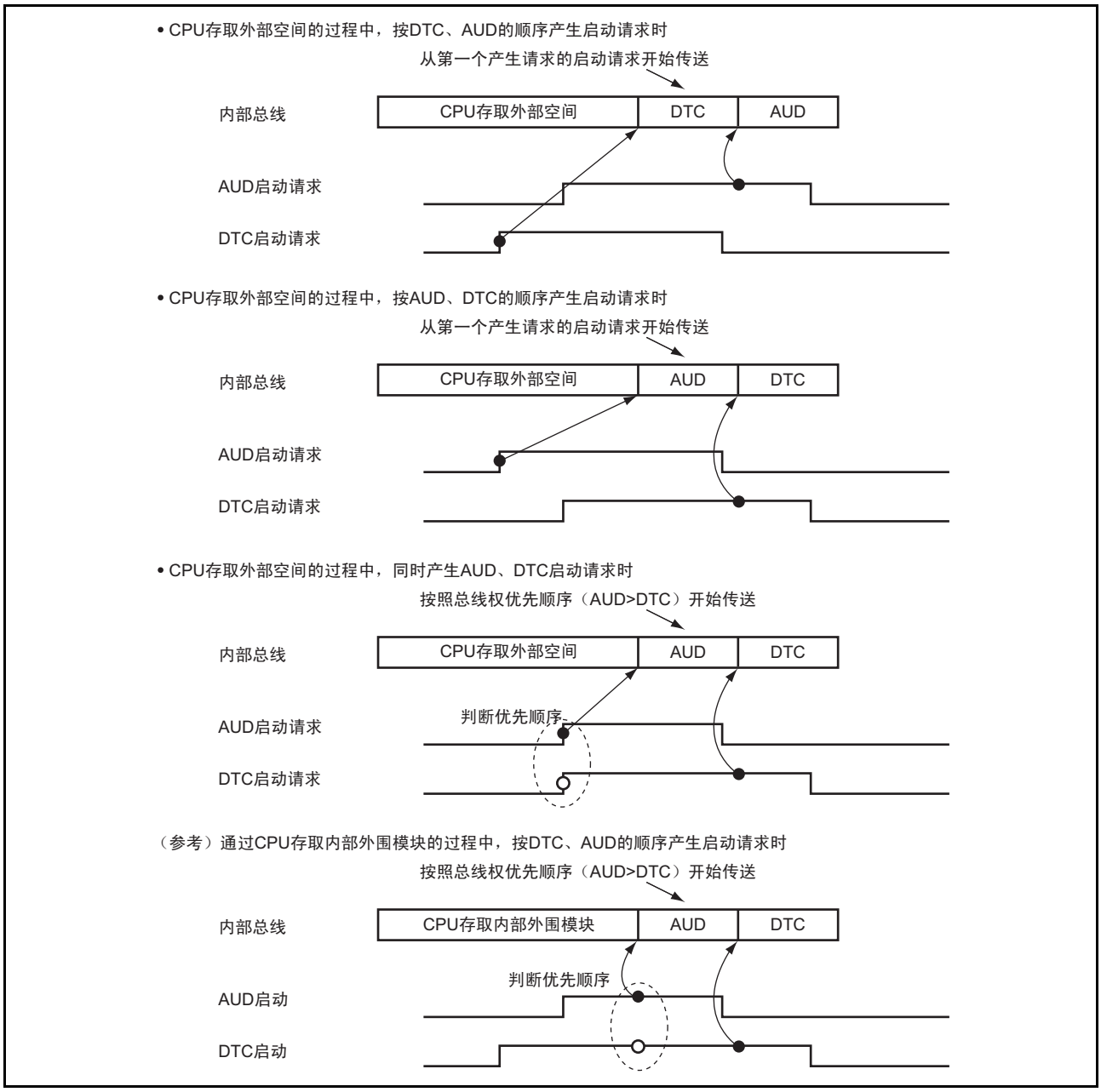


图 9.9　CPU 存取外部空间过程中产生 DTC/AUD 竞争时的总线仲裁

另外，“9.5.7(2) LSI 内部总线主控器的存取”中描述的写入缓冲器运行过程中， CPU 读取和写入外部空间时， CPU 和 AUD/DTC 的仲裁不同。 CPU 存取外部空间的过程中，产生 AUD/DTC 启动请求时的总线仲裁如图 9.10 所示。

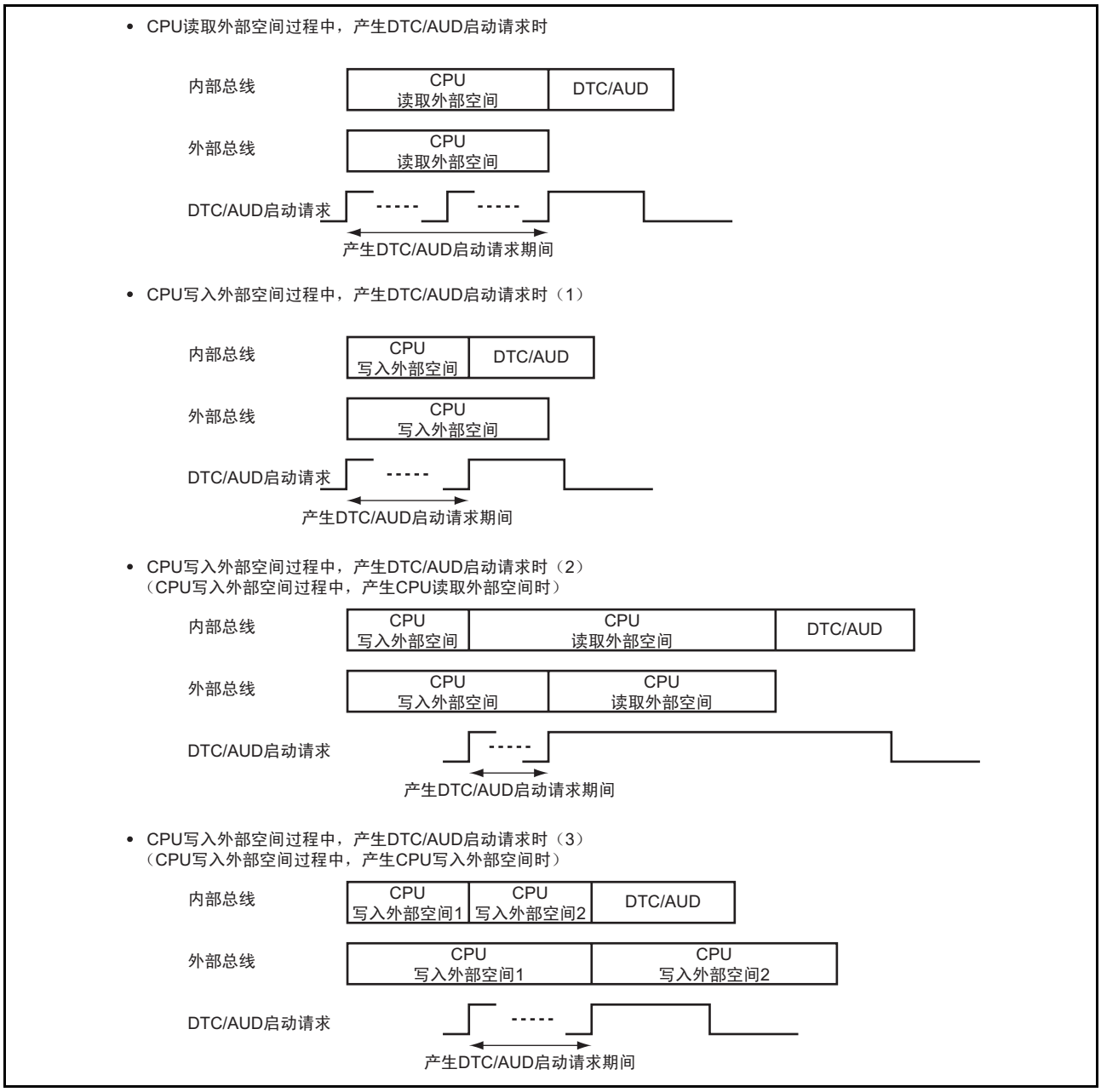


图 9.10　CPU 存取外部空间的过程中产生 DTC/AUD 启动请求时的总线仲裁

不执行总线仲裁的状态如下所示：

1. TAS指令的读取周期和写入周期之间
2. 因数据总线宽度小于存取长度而产生多个总线周期 （例如，对8位数据总线宽度的存储器执行长字存取时的总线周期之间）

向外部器件转让总线权时，为了防止连接的器件产生误动作，在释放总线前将所有总线控制信号设置为无效状态。接受总线权时，总线控制信号也从无效状态开始驱动。在总线控制信号的高阻抗状态，为了防止因外来噪声引起误动作，需要给这些控制信号配置上拉电阻。

在总线周期的边界与外部器件进行总线权转让。请求总线权时，如果未执行总线周期，则立即释放总线权；如果正在执行总线周期，则等总线周期结束后再释放总线权。即使从 LSI 外部看来未执行总线周期，也可能存在由于存取周期之间插入等待等，内部已开始执行总线周期的情况，因此，不可在看到 $\overline{\text{CSn}}$ 信号及其他总线控制信号后，立即判断是否释放总线。

由 $\overline{\text{BREQ}}$ 和 $\overline{\text{BACK}}$ 的信号交换释放外部总线时，会有很多开销。系统必须设计为从属模块的任务较多时，通过获取一次总线权便可执行多次总线周期。减少总线权转让所需的周期，可简化系统设计。

只要不接受外部器件的总线权请求，就继续占有总线权。接受外部总线权请求 $\overline{\text{BREQ}}$ 有效 （低电平），正在执行的总线周期结束后立即释放总线，并将总线使用允许 $\overline{\text{BACK}}$ 设置为有效 （低电平）。接受表示外部器件已释放总线的 $\overline{\text{BREQ}}$ 无效 （高电平）后，设置 $\overline{\text{BACK}}$ 为无效 （高电平），重新开始使用总线。

另外，在释放总线期间，不可转移至睡眠模式、软件待机模式和深度软件待机模式。

具体的总线释放顺序如下：

首先，与 CK 的上升沿同步，将地址总线和数据总线设置为高阻抗。 0.5 个周期后，与 CK 的下降沿同步，将总线使用允许信号设置为有效。在此后的 CK 上升沿，设置 $\overline{\text{CSn}}$ 等总线控制信号为高阻抗。这些总线控制信号，最迟在设置为高阻抗的 1 个周期前被置为高电平。在 CK 的下降沿采样总线权请求信号。

从外部器件再次获得总线权的顺序如下所示：

在 CK 的下降沿检测出 $\overline{\text{BREQ}}$ 无效时， 1.5 个周期后由高电平开始驱动总线控制信号。在此后的 CK 下降沿，设置总线使用允许信号为无效。然后在紧接着的上升沿开始驱动地址总线和数据总线。设置总线控制信号为有效后到实际开始总线周期最早是在与驱动地址和数据信号相同的 CK 上升沿，总线仲裁时序如图 9.11 所示。

总线权请求 （$\overline{\text{BREQ}}$ 的低电平有效）后的总线权释放 （$\overline{\text{BREQ}}$ 的高电平无效），必须在允许总线使用（$\overline{\text{BACK}}$ 的低电平有效）后执行。在 $\overline{\text{BACK}}$ 有效前设置 $\overline{\text{BREQ}}$ 为无效时，根据 $\overline{\text{BREQ}}$ 无效时序， $\overline{\text{BACK}}$ 仅 1 个周期有效，而使外部器件和本 LSI 之间可能产生总线冲突。

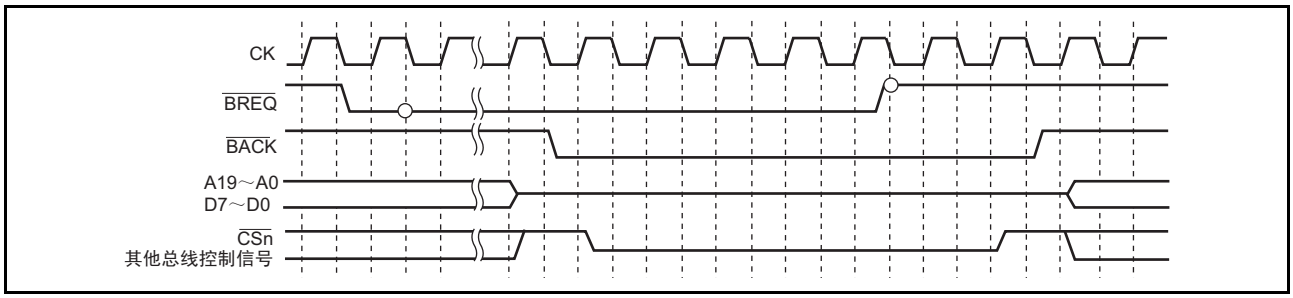


图 9.11　总线仲裁

9.5.7　其他

(1)　复位

总线状态控制器仅在上电复位时完全初始化。上电复位时，不管是否正在执行总线周期，都立即设置所有信号为无效，关闭输出缓冲器，初始化所有控制寄存器。在待机、睡眠及手动复位状态，一概不初始化总线状态控制器。手动复位时，正在执行的总线周期结束后，变为存取等待状态。手动复位信号有效时，不接受 $\overline{\text{BREQ}}$ 产生的仲裁请求。

(2)　LSI 内部总线主控器的存取

本 LSI 内部总线分为 L 总线、 I 总线及外围总线等 3 种。 CPU 连接到 L 总线； DTC 和总线状态控制器连接到 I 总线；低速外围模块分别连接到外围总线。另外，内部存储器连接到 L 总线和 I 总线。

存取外部空间和内部外围模块时，通过 I 总线启动存取周期。因此， CPU 存取内部存储器时，不和 CPU 产生总线仲裁，就可启动 DTC。

总线状态控制器有一段写入缓冲器，因此即使在写入周期，芯片外部的总线周期未结束， I 总线也可用于其他存取。写入芯片外部的低速存储器后，读取或写入内部外围模块时，不必等低速存储器写入结束即可存取内部外围模块。

CPU 需等待至运行结束才可开始读取，因此在确认数据已写入到实际器件后想要继续进行处理时，只需连续对相同地址进行虚读，即可确认写入结束。

DTC 存取时，总线状态控制器的写入缓冲器也同样运行。

另外，如果在写入缓冲器运行过程中更改 BSC 寄存器，则不可正确存取，因此，不得在写入存取后立即更改 BSC 寄存器。如果需要在写入存取后立即更改 BSC 寄存器，则必须先执行写入数据的虚读再确认写入存取结束。

9.5.8　CPU 对内部 FLASH、内部 RAM 的存取

与 Iϕ 时钟同步，以 1 个时钟周期读取内部 FLASH。有关写入、擦除，详情参阅 “第 20 章　闪存”； 与 Iϕ 时钟同步，以 1 个时钟周期读取 / 写入内部 RAM，详情参阅 “第 21 章　RAM”。

9.5.9　CPU 对内部外围 I/O 寄存器的存取

CPU 存取内部外围 I/O 寄存器时的存取周期数如表 9.10 所示。

表 9.10　内部外围 I/O 寄存器的存取周期数

	存取周期数
写入	(3+n)×Iϕ+(1+m)×Bϕ+2×Pϕ
读取	(3+n)×Iϕ+(1+m)×Bϕ+2×Pϕ+2×Iϕ

【注】 1.　Iϕ:Bϕ=8:1 时， n=0 ～ 7
Iϕ:Bϕ=4:1 时， n=0～3
Bϕ:Pϕ=4:1 时， m=0～3
Iϕ:Bϕ=3:1 时， n=0～2
Bϕ:Pϕ=3:1 时， m=0～2
Iϕ:Bϕ=2:1 时， n=0～1
Bϕ:Pϕ=2:1 时， m=0～1
Iϕ:Bϕ=1:1 时， n=0
Bϕ:Pϕ=1:1 时， m=0
n、 m 取决于内部的执行状态。

2.　MIϕ、 MPϕ 的时钟比不影响存取周期数。

本产品采用同步逻辑，总线结构为分层总线结构。 L 总线、 I 总线、外围总线的数据输入 / 输出，分别与 Iϕ 时钟、 Bϕ 时钟、 Pϕ 时钟的上升沿同步。对 Iϕ:B ϕ:Pϕ=4:2:2、外围总线连接宽度为 16 位、以 2 个 Pϕ 周期存取的寄存器写入字节长度数据的时序例子如图 9.12 所示。 CPU 存取内部外围 I/O 寄存器时，对 L 总线输出数据后，向 I 总线传送数据时需要 3 个 Iϕ 周期的准备时间。 3 个周期后，即可与 Bϕ 的上升沿同步向 I 总线传送数据，但 Iϕ:Bϕ=4:2 时， 1 个 Bϕ 时钟周期有 2 个 Iϕ 时钟周期。因此 Iϕ:Bϕ=4:2 时，在 (3+n)×Iϕ、 n=0 ～ 1 的期间，从 L 总线向 I 总线传送数据 （图中例子为 3×Iϕ）。数据载入 L 总线的时序与 Bϕ 上升沿的时序关系取决于程序执行状态。图中例子为 n=0、 m=0，因此存取时间为 3×Iϕ+1×Bϕ+2×Pϕ。

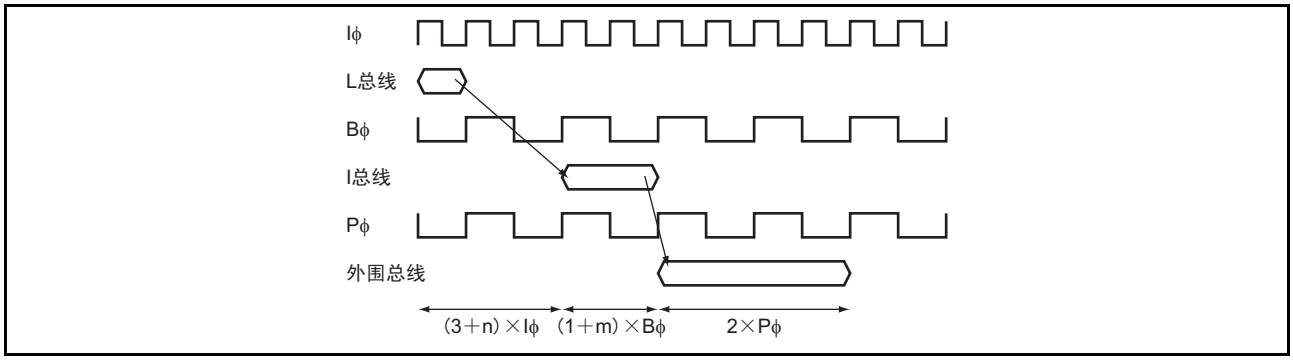


图 9.12　Iϕ:Bϕ:Pϕ=4:2:2 时内部外围 I/O 寄存器的写入存取时序

Iϕ:Bϕ:Pϕ=4:2:1 时，外围总线的读取存取时序如图 9.13 所示。从 L 总线到外围总线的传送与写入时相同，但读取时需要向 CPU 传送在外围总线读取的值。从外围总线到 I 总线和从 I 总线到 L 总线的传送也与各总线时钟的上升沿同步，但 Iϕ ≥ Bϕ ≥ Pϕ，因此需要 2×Iϕ 的时间。下图例子中 n=0、 m=1，因此存取时间为 3×Iϕ+2×Bϕ+2×Pϕ+2×Iϕ。

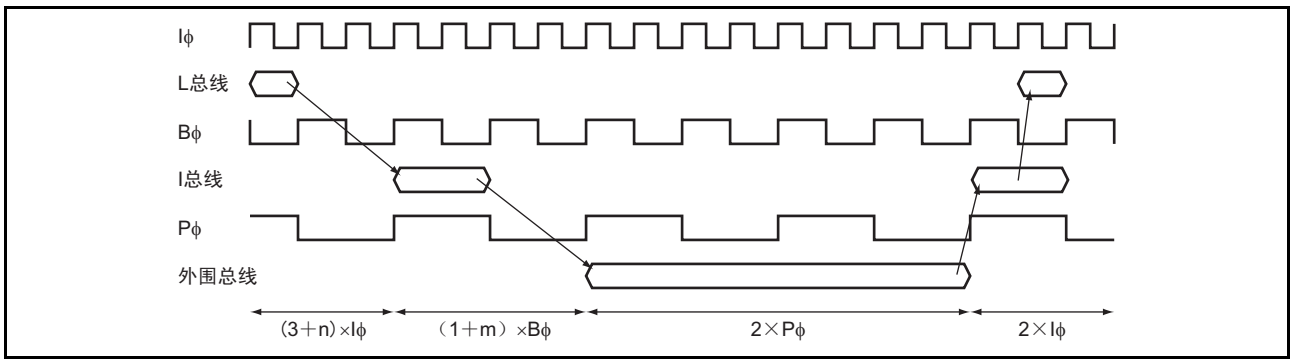


图 9.13　Iϕ:Bϕ:Pϕ=4:2:1 时内部外围 I/O 寄存器的读取存取时序

第 10 章　多功能定时器脉冲单元 2 （MTU2）、 2S （MTU2S）

本 LSI 内置由 5 个通道 （CH0 ～ 4）的 16 位定时器构成的多功能定时器脉冲单元 2 （MTU2）和由 3 个通道 （CH3 ～ 5）的 16 位定时器构成的多功能定时器脉冲单元 2S （MTU2S）。仅 MTU2S 的互补 PWM 模式输出可在≤ 80MHz 下运行， MTU2S 的其他功能及 MTU2 则可在≤ 40MHz 下运行。本章为了区分 MTU2 和 MTU2S，在 MTU2S 的输入 / 输出引脚名称及寄存器名称末尾追加 “S”，例如：TIOC3A 写为 TIOC3AS，TGRA_3 则写为 TGRA_3S 等。 MTU2 和 MTU2S 的 CH3 和 CH4 功能相同，因此从 10.3.1 节开始统一写为 MTU2，不加 “S”（CH5 也不加 “S”）。

10.1　特点

- MTU2最多可执行16个脉冲输入/输出， MTU2S最多可执行6个脉冲输入/输出和3个脉冲输入
- 各通道可选择6～8种计数器输入时钟 （CH5为4种）
- MTU2的CH0～4可设定以下运行：比较匹配的输出波形、输入捕捉功能、清除计数器、同时写入多个定时器计数器 （TCNT）、比较匹配/输入捕捉的同步清除、计数器的同步运行的各寄存器同步输入/输出、与同步运行组合最多输出12相PWM （组合CH3、 4的功能， MTU2S最多输出6相PWM）
- CH0、 3、 4可设定缓冲运行
- CH1、 2可分别独立设定相位计数模式
- 级联运行
- 通过内部16位总线高速存取
- MTU2有25种中断源， MTU2S有13种中断源
- 可自动传送寄存器数据
- 可生成A/D转换器的转换开始触发
- 可设定模块待机模式
- 由CH3、 4联动，可设定互补PWM、复位PWM3相的正、负共6相波形输出
- 由CH0、 3、 4联动，可设定使用互补PWM、复位PWM的AC同步马达 （无刷DC马达）驱动模式，并可选择两种 （斩波、电平）波形输出 （仅限MTU2 ）
- 由CH5可实现空载时间补偿计数器功能 （仅限MTU2S）
- 在互补PWM模式，可跳过计数器波峰/波谷处的中断及A/D转换器转换开始触发

表 10.1　MTU2 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		MPϕ/1 MPϕ/4 MPϕ/16 MPϕ/64 TCLKA TCLKB TCLKC TCLKD	MPϕ/1 MPϕ/4 MPϕ/16 MPϕ/64 MPϕ/256 TCLKA TCLKB	MPϕ/1 MPϕ/4 MPϕ/16 MPϕ/64 MPϕ/1024 TCLKA TCLKB TCLKC	MPϕ/1 MPϕ/4 MPϕ/16 MPϕ/64 MPϕ/256 MPϕ/1024 TCLKA TCLKB	MPϕ/1 MPϕ/4 MPϕ/16 MPϕ/64 MPϕ/256 MPϕ/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
计数器清除功能		TGR 的 比较匹配 / 输入捕捉	TGR 的 比较匹配 / 输入捕捉	TGR 的 比较匹配 / 输入捕捉	TGR 的 比较匹配 / 输入捕捉	TGR 的 比较匹配 / 输入捕捉
比较 匹配输出	输出 0	○	○	○	○	○
	输出 1	○	○	○	○	○
	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		○	—	—	○	○
空载时间补偿计数器功能		—	—	—	—	—
DTC 的启动		TGR 的 比较匹配 / 输入捕捉	TGR 的 比较匹配 / 输入捕捉	TGR 的 比较匹配 / 输入捕捉	TGR 的 比较匹配 / 输入捕捉	TGR 的比较匹配 / 输入捕捉和 TCNT 上溢 / 下溢
A/D 转换开始触发		TGRA_0 的 比较匹配 / 输入捕捉 TGRE_0 的 比较匹配	TGRA_1 的 比较匹配 / 输入捕捉	TGRA_2 的 比较匹配 / 输入捕捉	TGRA_3 的 比较匹配 / 输入捕捉	TGRA_4 的 比较匹配 / 输入捕捉 在互补 PWM 模式， TCNT_4 的下溢 （波谷）

项　目	通道 0	通道 1	通道 2	通道 3	通道 4
中断源	7 个 • 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 • 比较匹配 / 输入捕捉 1A • 比较匹配 / 输入捕捉 1B • 上溢 • 下溢	4 个 • 比较匹配 / 输入捕捉 2A • 比较匹配 / 输入捕捉 2B • 上溢 • 下溢	5 个 • 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 • 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
中断跳过功能	—	—	—	• 跳过 TGRA_3 的比较匹配中断	• 跳过TCIV_4 中断

【符号说明】

○：可

—：不可

表 10.2　MTU2S 功能一览表

项　目		通道 3	通道 4	通道 5
计数时钟		MIϕ/1 MIϕ/4 MIϕ/16 MIϕ/64 MIϕ/256 MIϕ/1024	MIϕ/1 MIϕ/4 MIϕ/16 MIϕ/64 MIϕ/256 MIϕ/1024	MIϕ/1 MIϕ/4 MIϕ/16 MIϕ/64
通用寄存器 （TGR）		TGRA_3S TGRB_3S	TGRA_4S TGRB_4S	TGRU_5S TGRV_5S TGRW_5S
通用寄存器 / 缓冲寄存器		TGRC_3S TGRD_3S	TGRC_4S TGRD_4S	—
输入 / 输出引脚		TIOC3BS TIOC3DS	TIOC4AS TIOC4BS TIOC4CS TIOC4DS	输入引脚 TIC5US TIC5VS TIC5WS
计数器清除功能		TGR 的比较匹配 / 输入捕捉	TGR 的比较匹配 / 输入捕捉	TGR 的比较匹配 / 输入捕捉
比较 匹配输出	输出 0	○	○	—
	输出 1	○	○	—
	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		○	○	—
空载时间补偿计数器功能		—	—	○
DTC 的启动		TGR 的比较匹配 / 输入捕捉	TGR 的比较匹配 / 输入捕捉和 TCNT 上溢 / 下溢	TGR 的比较匹配 / 输入捕捉
A/D 转换开始触发		TGRA_3S 的比较匹配 / 输入捕捉	TGRA_4S 的比较匹配 / 输入捕捉 在互补 PWM 模式， TCNT_4S 下溢（波谷）	—
中断源		5 个 • 比较匹配 / 输入捕捉 3AS • 比较匹配 / 输入捕捉 3BS • 比较匹配 / 输入捕捉 3CS • 比较匹配 / 输入捕捉 3DS • 上溢	5 个 • 比较匹配 / 输入捕捉 4AS • 比较匹配 / 输入捕捉 4BS • 比较匹配 / 输入捕捉 4CS • 比较匹配 / 输入捕捉 4DS • 上溢 / 下溢	3 个 • 比较匹配 / 输入捕捉 5US • 比较匹配 / 输入捕捉 5VS • 比较匹配 / 输入捕捉 5WS
A/D 转换开始请求延迟功能		—	• TADCORA_4S 和 TCNT_4S 匹配时，请求开始 A/D 转换 • TADCORB_4S 和 TCNT_4S 匹配时，请求开始 A/D 转换	—
中断跳过功能		• 跳过 TGRA_3S 的比较匹配中断	• 跳过 TCIV_4S 中断	—

【符号说明】

○：可

—：不可

MTU2 框图如图 10.1 所示。

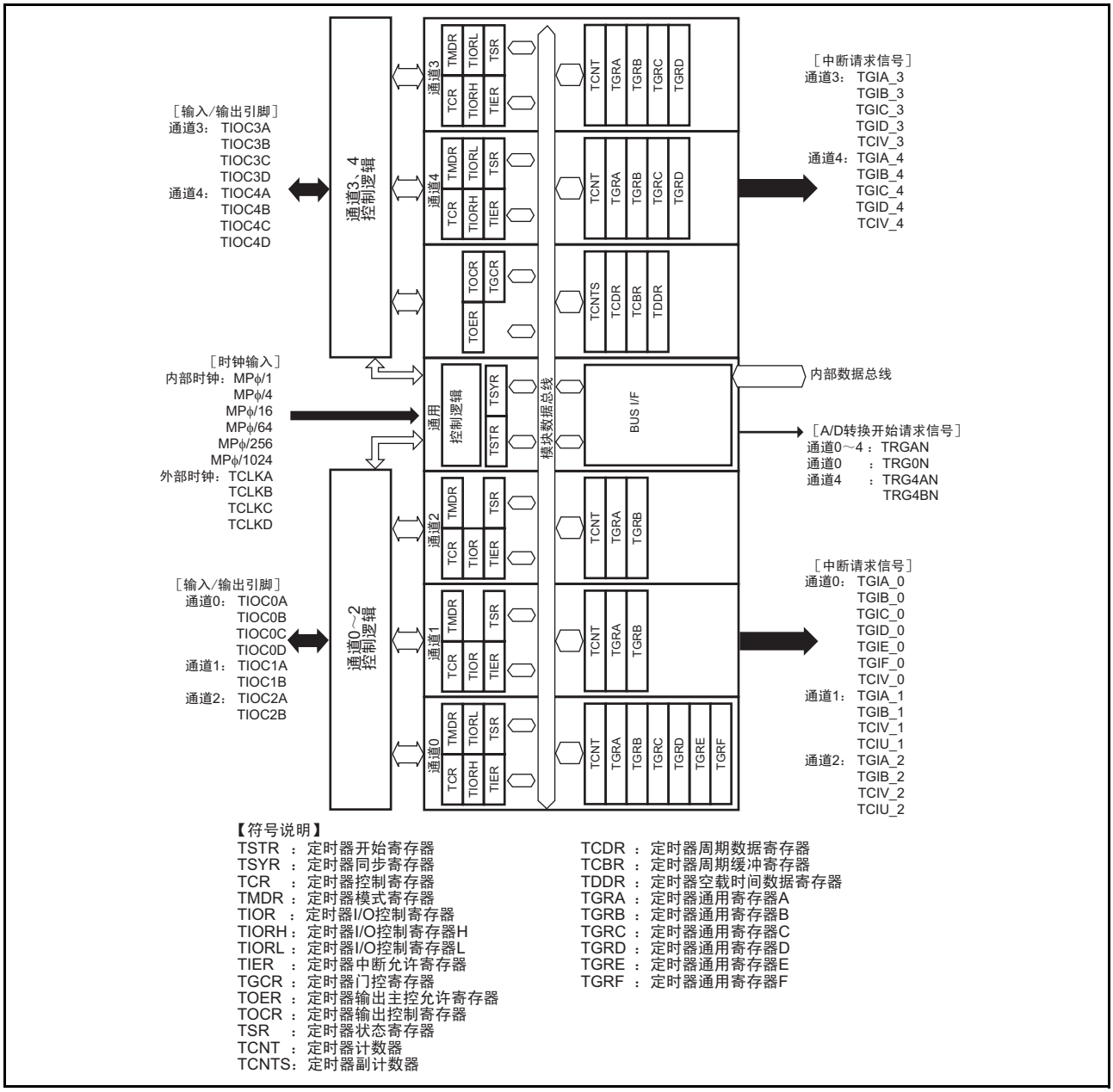
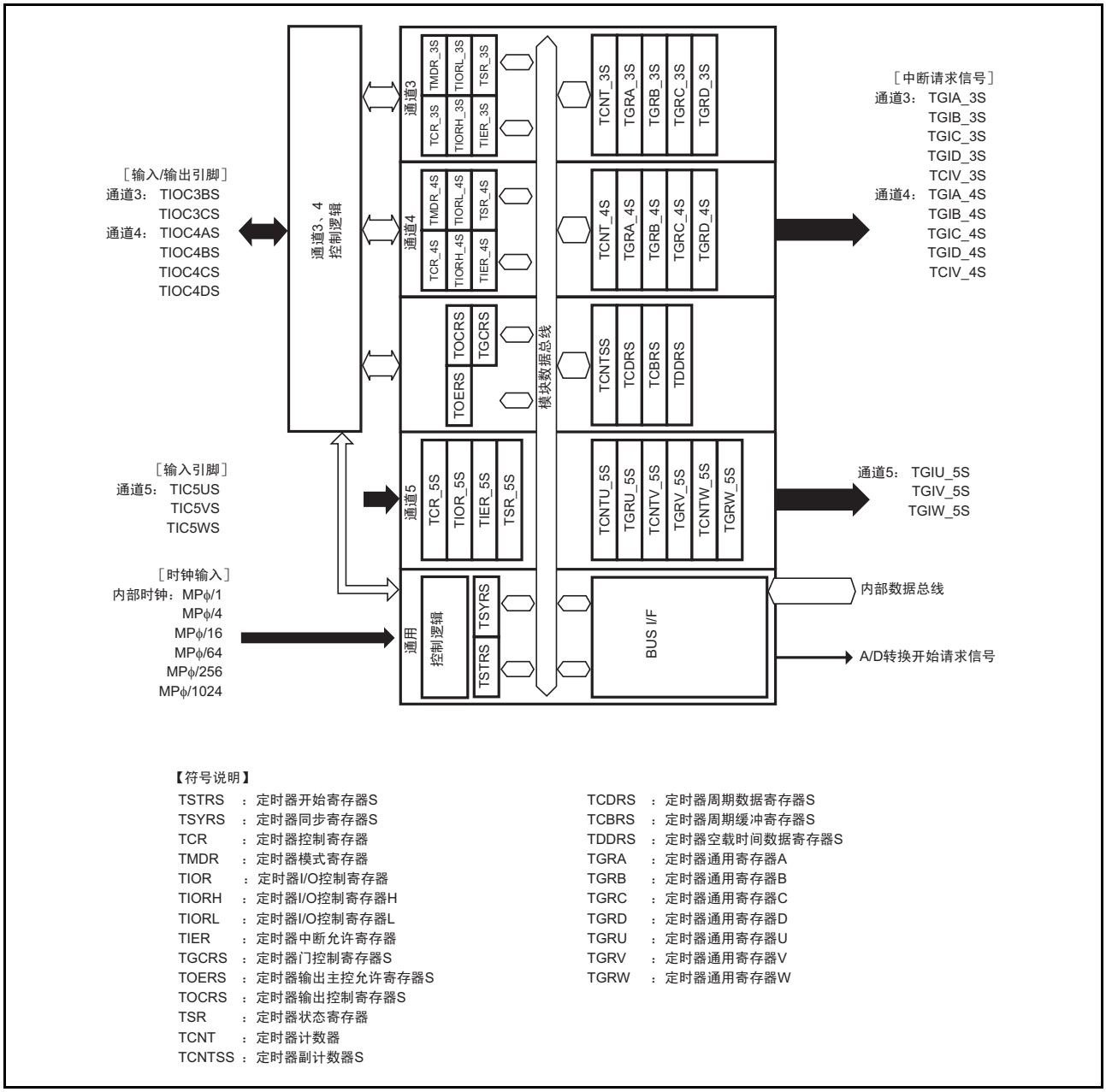


图 10.1　MTU2 框图

MTU2S 框图如图 10.2 所示。



【符号说明】

TSTRS ：定时器开始寄存器S
TSYRS ：定时器同步寄存器S
TCR ：定时器控制寄存器
TMDR ：定时器模式寄存器
TIOR ：定时器I/O控制寄存器
TIORH ：定时器I/O控制寄存器H
TIORL ：定时器I/O控制寄存器L
TIER ：定时器中断允许寄存器
TGCRS ：定时器门控制寄存器S
TOERS ：定时器输出主控允许寄存器S
TOCRS ：定时器输出控制寄存器S
TSR ：定时器状态寄存器
TCNT ：定时器计数器
TCNTSS ：定时器副计数器S
TCDRS ：定时器周期数据寄存器S
TCBRS ：定时器周期缓冲寄存器S
TDDRS ：定时器空载时间数据寄存器S
TGRA ：定时器通用寄存器A
TGRB ：定时器通用寄存器B
TGRC ：定时器通用寄存器C
TGRD ：定时器通用寄存器D
TGRU ：定时器通用寄存器U
TGRV ：定时器通用寄存器V
TGRW ：定时器通用寄存器W

图 10.2　MTU2S 框图

10.2　输入 / 输出引脚

表 10.3　MTU2 的引脚结构

通道	引脚名称	输入 / 输出	功　能
通用	TCLKA	输入	外部时钟 A 输入引脚 （通道 1 的相位计数模式 A 相输入）
	TCLKB	输入	外部时钟 B 输入引脚 （通道 1 的相位计数模式 B 相输入）
	TCLKC	输入	外部时钟 C 输入引脚 （通道 2 的相位计数模式 A 相输入）
	TCLKD	输入	外部时钟 D 输入引脚 （通道 2 的相位计数模式 B 相输入）
0	TIOC0A	输入 / 输出	TGRA_0 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚
	TIOC0B	输入 / 输出	TGRB_0 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚
	TIOC0C	输入 / 输出	TGRC_0 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚
	TIOC0D	输入 / 输出	TGRD_0 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚
1	TIOC1A	输入 / 输出	TGRA_1 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚
	TIOC1B	输入 / 输出	TGRB_1 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚
2	TIOC2A	输入 / 输出	TGRA_2 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚
	TIOC2B	输入 / 输出	TGRB_2 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚
3	TIOC3A	输入 / 输出	TGRA_3 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚
	TIOC3B	输入 / 输出	TGRB_3 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚
	TIOC3C	输入 / 输出	TGRC_3 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚
	TIOC3D	输入 / 输出	TGRD_3 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚
4	TIOC4A	输入 / 输出	TGRA_4 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚
	TIOC4B	输入 / 输出	TGRB_4 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚
	TIOC4C	输入 / 输出	TGRC_4 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚
	TIOC4D	输入 / 输出	TGRD_4 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚

表 10.4　MTU2S 的引脚结构

通道	引脚名称	输入 / 输出	功　能
3	TIOC3BS	输入 / 输出	TGRB_3S 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚
	TIOC3DS	输入 / 输出	TGRD_3S 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚
4	TIOC4AS	输入 / 输出	TGRA_4S 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚
	TIOC4BS	输入 / 输出	TGRB_4S 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚
	TIOC4CS	输入 / 输出	TGRC_4S 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚
	TIOC4DS	输入 / 输出	TGRD_4S 的输入捕捉输入 / 输出比较输出 /PWM 输出引脚
5	TIC5US	输入	TGRU_5S 的输入捕捉输入 / 外部脉冲输入引脚
	TIC5VS	输入	TGRV_5S 的输入捕捉输入 / 外部脉冲输入引脚
	TIC5WS	输入	TGRW_5S 的输入捕捉输入 / 外部脉冲输入引脚

10.3　寄存器说明

MTU2 和 MTU2S 的各通道有下列寄存器。有关寄存器的地址及各处理状态 的寄存器状态，详情参阅“第 24 章　寄存器一览表”。各通道寄存器名称的表示：如 MTU2 通道 0 的 TCR，写为 TCR_0。

表 10.5　MTU2 的寄存器结构

寄存器名称	简称	R/W	初始值	地址	存取长度
定时器控制寄存器 _3	TCR_3	R/W	H'00	H'FFFFC200	8、16、32
定时器控制寄存器 _4	TCR_4	R/W	H'00	H'FFFFC201	8
定时器模式寄存器 _3	TMDR_3	R/W	H'00	H'FFFFC202	8、16
定时器模式寄存器 _4	TMDR_4	R/W	H'00	H'FFFFC203	8
定时器 I/O 控制寄存器 H_3	TIORH_3	R/W	H'00	H'FFFFC204	8、16、32
定时器 I/O 控制寄存器 L_3	TIORL_3	R/W	H'00	H'FFFFC205	8
定时器 I/O 控制寄存器 H_4	TIORH_4	R/W	H'00	H'FFFFC206	8、16
定时器 I/O 控制寄存器 L_4	TIORL_4	R/W	H'00	H'FFFFC207	8
定时器中断允许寄存器 _3	TIER_3	R/W	H'00	H'FFFFC208	8、16
定时器中断允许寄存器 _4	TIER_4	R/W	H'00	H'FFFFC209	8
定时器输出主控允许寄存器	TOER	R/W	H'C0	H'FFFFC20A	8
定时器门控寄存器	TGCR	R/W	H'80	H'FFFFC20D	8
定时器输出控制寄存器 1	TOCR1	R/W	H'00	H'FFFFC20E	8、16
定时器输出控制寄存器 2	TOCR2	R/W	H'00	H'FFFFC20F	8
定时器计数器 _3	TCNT_3	R/W	H'0000	H'FFFFC210	16、32
定时器计数器 _4	TCNT_4	R/W	H'0000	H'FFFFC212	16
定时器周期数据寄存器	TCDR	R/W	H'FFFF	H'FFFFC214	16、32
定时器空载时间数据寄存器	TDDR	R/W	H'FFFF	H'FFFFC216	16
定时器通用寄存器 A_3	TGRA_3	R/W	H'FFFF	H'FFFFC218	16、32
定时器通用寄存器 B_3	TGRB_3	R/W	H'FFFF	H'FFFFC21A	16
定时器通用寄存器 A_4	TGRA_4	R/W	H'FFFF	H'FFFFC21C	16、32
定时器通用寄存器 B_4	TGRB_4	R/W	H'FFFF	H'FFFFC21E	16
定时器副计数器	TCNTS	R	H'0000	H'FFFFC220	16、32
定时器周期缓冲寄存器	TCBR	R/W	H'FFFF	H'FFFFC222	16
定时器通用寄存器 C_3	TGRC_3	R/W	H'FFFF	H'FFFFC224	16、32
定时器通用寄存器 D_3	TGRD_3	R/W	H'FFFF	H'FFFFC226	16
定时器通用寄存器 C_4	TGRC_4	R/W	H'FFFF	H'FFFFC228	16、32
定时器通用寄存器 D_4	TGRD_4	R/W	H'FFFF	H'FFFFC22A	16
定时器状态寄存器 _3	TSR_3	R/W	H'C0	H'FFFFC22C	8、16
定时器状态寄存器 _4	TSR_4	R/W	H'C0	H'FFFFC22D	8

寄存器名称	简称	R/W	初始值	地址	存取长度
定时器中断跳过设定寄存器	TITCR	R/W	H'00	H'FFFFC230	8、16
定时器中断跳过次数计数器	TITCNT	R	H'00	H'FFFFC231	8
定时器缓冲传送设定寄存器	TBTER	R/W	H'00	H'FFFFC232	8
定时器空载时间允许寄存器	TDER	R/W	H'01	H'FFFFC234	8
定时器输出电平缓冲寄存器	TOLBR	R/W	H'00	H'FFFFC236	8
定时器缓冲运行传送模式寄存器 _3	TBTM_3	R/W	H'00	H'FFFFC238	8、16
定时器缓冲运行传送模式寄存器 _4	TBTM_4	R/W	H'00	H'FFFFC239	8
定时器 A/D 转换开始请求控制寄存器	TADCR	R/W	H'0000	H'FFFFC240	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'FFFFC244	16、32
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'FFFFC246	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'FFFFC248	16、32
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'FFFFC24A	16
定时器波形控制寄存器	TWCR	R/W	H'00	H'FFFFC260	8
定时器开始寄存器	TSTR	R/W	H'00	H'FFFFC280	8、16
定时器同步寄存器	TSYR	R/W	H'00	H'FFFFC281	8
定时器计数器同步开始寄存器	TCSYSTR	R/W	H'00	H'FFFFC282	8
定时器读取写入允许寄存器	TRWER	R/W	H'01	H'FFFFC284	8
定时器控制寄存器 _0	TCR_0	R/W	H'00	H'FFFFC300	8、16、32
定时器模式寄存器 _0	TMDR_0	R/W	H'00	H'FFFFC301	8
定时器 I/O 控制寄存器 H_0	TIORH_0	R/W	H'00	H'FFFFC302	8、16
定时器 I/O 控制寄存器 L_0	TIORL_0	R/W	H'00	H'FFFFC303	8
定时器中断允许寄存器 _0	TIER_0	R/W	H'00	H'FFFFC304	8、16、32
定时器状态寄存器 _0	TSR_0	R/W	H'C0	H'FFFFC305	8
定时器计数器 _0	TCNT_0	R/W	H'0000	H'FFFFC306	16
定时器通用寄存器 A_0	TGRA_0	R/W	H'FFFF	H'FFFFC308	16、32
定时器通用寄存器 B_0	TGRB_0	R/W	H'FFFF	H'FFFFC30A	16
定时器通用寄存器 C_0	TGRC_0	R/W	H'FFFF	H'FFFFC30C	16、32
定时器通用寄存器 D_0	TGRD_0	R/W	H'FFFF	H'FFFFC30E	16
定时器通用寄存器 E_0	TGRE_0	R/W	H'FFFF	H'FFFFC320	16、32
定时器通用寄存器 F_0	TGRF_0	R/W	H'FFFF	H'FFFFC322	16

寄存器名称	简称	R/W	初始值	地址	存取长度
定时器中断允许寄存器 2_0	TIER2_0	R/W	H'00	H'FFFFC324	8、16
定时器状态寄存器 2_0	TSR2_0	R/W	H'C0	H'FFFFC325	8
定时器缓冲运行传送模式寄存器 _0	TBTM_0	R/W	H'00	H'FFFFC326	8
定时器控制寄存器 _1	TCR_1	R/W	H'00	H'FFFFC380	8、16
定时器模式寄存器 _1	TMDR_1	R/W	H'00	H'FFFFC381	8
定时器 I/O 控制寄存器 _1	TIOR_1	R/W	H'00	H'FFFFC382	8
定时器中断允许寄存器 _1	TIER_1	R/W	H'00	H'FFFFC384	8、16、32
定时器状态寄存器 _1	TSR_1	R/W	H'C0	H'FFFFC385	8
定时器计数器 _1	TCNT_1	R/W	H'0000	H'FFFFC386	16
定时器通用寄存器 A_1	TGRA_1	R/W	H'FFFF	H'FFFFC388	16、32
定时器通用寄存器 B_1	TGRB_1	R/W	H'FFFF	H'FFFFC38A	16
定时器输入捕捉控制寄存器	TICCR	R/W	H'00	H'FFFFC390	8
定时器控制寄存器 _2	TCR_2	R/W	H'00	H'FFFFC400	8、16
定时器模式寄存器 _2	TMDR_2	R/W	H'00	H'FFFFC401	8
定时器 I/O 控制寄存器 _2	TIOR_2	R/W	H'00	H'FFFFC402	8
定时器中断允许寄存器 _2	TIER_2	R/W	H'00	H'FFFFC404	8、16、32
定时器状态寄存器 _2	TSR_2	R/W	H'C0	H'FFFFC405	8
定时器计数器 _2	TCNT_2	R/W	H'0000	H'FFFFC406	16
定时器通用寄存器 A_2	TGRA_2	R/W	H'FFFF	H'FFFFC408	16、32
定时器通用寄存器 B_2	TGRB_2	R/W	H'FFFF	H'FFFFC40A	16

表 10.6　MTU2S 的寄存器结构

寄存器名称	简称	R/W	初始值	地址	存取长度
定时器控制寄存器 _3S	TCR_3S	R/W	H'00	H'FFFFC600	8、16、32
定时器控制寄存器 _4S	TCR_4S	R/W	H'00	H'FFFFC601	8
定时器模式寄存器 _3S	TMDR_3S	R/W	H'00	H'FFFFC602	8、16
定时器模式寄存器 _4S	TMDR_4S	R/W	H'00	H'FFFFC603	8
定时器 I/O 控制寄存器 H_3S	TIORH_3S	R/W	H'00	H'FFFFC604	8、16、32
定时器 I/O 控制寄存器 L_3S	TIORL_3S	R/W	H'00	H'FFFFC605	8
定时器 I/O 控制寄存器 H_4S	TIORH_4S	R/W	H'00	H'FFFFC606	8、16
定时器 I/O 控制寄存器 L_4S	TIORL_4S	R/W	H'00	H'FFFFC607	8
定时器中断允许寄存器 _3S	TIER_3S	R/W	H'00	H'FFFFC608	8、16
定时器中断允许寄存器 _4S	TIER_4S	R/W	H'00	H'FFFFC609	8
定时器输出主控允许寄存器 S	TOERS	R/W	H'C0	H'FFFFC60A	8
定时器门控寄存器 S	TGCRS	R/W	H'80	H'FFFFC60D	8
定时器输出控制寄存器 1S	TOCR1S	R/W	H'00	H'FFFFC60E	8、16
定时器输出控制寄存器 2S	TOCR2S	R/W	H'00	H'FFFFC60F	8
定时器计数器 _3S	TCNT_3S	R/W	H'0000	H'FFFFC610	16、32
定时器计数器 _4S	TCNT_4S	R/W	H'0000	H'FFFFC612	16
定时器周期数据寄存器 S	TCDRS	R/W	H'FFFF	H'FFFFC614	16、32
定时器空载时间数据寄存器 S	TDDRS	R/W	H'FFFF	H'FFFFC616	16
定时器通用寄存器 A_3S	TGRA_3S	R/W	H'FFFF	H'FFFFC618	16、32
定时器通用寄存器 B_3S	TGRB_3S	R/W	H'FFFF	H'FFFFC61A	16
定时器通用寄存器 A_4S	TGRA_4S	R/W	H'FFFF	H'FFFFC61C	16、32
定时器通用寄存器 B_4S	TGRB_4S	R/W	H'FFFF	H'FFFFC61E	16
定时器副计数器 S	TCNTSS	R	H'0000	H'FFFFC620	16、32
定时器周期缓冲寄存器 S	TCBRS	R/W	H'FFFF	H'FFFFC622	16
定时器通用寄存器 C_3S	TGRC_3S	R/W	H'FFFF	H'FFFFC624	16、32
定时器通用寄存器 D_3S	TGRD_3S	R/W	H'FFFF	H'FFFFC626	16
定时器通用寄存器 C_4S	TGRC_4S	R/W	H'FFFF	H'FFFFC628	16、32
定时器通用寄存器 D_4S	TGRD_4S	R/W	H'FFFF	H'FFFFC62A	16
定时器状态寄存器 _3S	TSR_3S	R/W	H'C0	H'FFFFC62C	8、16
定时器状态寄存器 _4S	TSR_4S	R/W	H'C0	H'FFFFC62D	8
定时器中断跳过设定寄存器 S	TITCRS	R/W	H'00	H'FFFFC630	8、16
定时器中断跳过次数计数器 S	TITCNTS	R	H'00	H'FFFFC631	8
定时器缓冲传送设定寄存器 S	TBTERS	R/W	H'00	H'FFFFC632	8
定时器空载时间允许寄存器 S	TDERS	R/W	H'01	H'FFFFC634	8

寄存器名称	简称	R/W	初始值	地址	存取长度
定时器输出电平缓冲寄存器 S	TOLBRS	R/W	H'00	H'FFFFC636	8
定时器缓冲运行传送模式寄存器 _3S	TBTM_3S	R/W	H'00	H'FFFFC638	8、16
定时器缓冲运行传送模式寄存器 _4S	TBTM_4S	R/W	H'00	H'FFFFC639	8
定时器 A/D 转换开始请求控制寄存器 S	TADCRS	R/W	H'0000	H'FFFFC640	16
定时器 A/D 转换开始请求周期设定寄存器 A_4S	TADCORA_4S	R/W	H'FFFF	H'FFFFC644	16、32
定时器 A/D 转换开始请求周期设定寄存器 B_4S	TADCORB_4S	R/W	H'FFFF	H'FFFFC646	16
定时器 A/D 转换开始请求周期设定缓冲寄存器 A_4S	TADCOBRA_4S	R/W	H'FFFF	H'FFFFC648	16、32
定时器 A/D 转换开始请求周期设定缓冲寄存器 B_4S	TADCOBRB_4S	R/W	H'FFFF	H'FFFFC64A	16
定时器同步清除寄存器 S	TSYCRS	R/W	H'00	H'FFFFC650	8
定时器波形控制寄存器 S	TWCRS	R/W	H'00	H'FFFFC660	8
定时器开始寄存器 S	TSTRS	R/W	H'00	H'FFFFC680	8、16
定时器同步寄存器 S	TSYRS	R/W	H'00	H'FFFFC681	8
定时器读取 / 写入允许寄存器 S	TRWERS	R/W	H'01	H'FFFFC684	8
定时器计数器 U_5S	TCNTU_5S	R/W	H'0000	H'FFFFC880	16、32
定时器通用寄存器 U_5S	TGRU_5S	R/W	H'FFFF	H'FFFFC882	16
定时器控制寄存器 U_5S	TCRU_5S	R/W	H'00	H'FFFFC884	8
定时器 I/O 控制寄存器 U_5S	TIORU_5S	R/W	H'00	H'FFFFC886	8
定时器计数器 V_5S	TCNTV_5S	R/W	H'0000	H'FFFFC890	16、32
定时器通用寄存器 V_5S	TGRV_5S	R/W	H'FFFF	H'FFFFC892	16
定时器控制寄存器 V_5S	TCRV_5S	R/W	H'00	H'FFFFC894	8
定时器 I/O 控制寄存器 V_5S	TIORV_5S	R/W	H'00	H'FFFFC896	8
定时器计数器 W_5S	TCNTW_5S	R/W	H'0000	H'FFFFC8A0	16、32
定时器通用寄存器 W_5S	TGRW_5S	R/W	H'FFFF	H'FFFFC8A2	16
定时器控制寄存器 W_5S	TCRW_5S	R/W	H'00	H'FFFFC8A4	8
定时器 I/O 控制寄存器 W_5S	TIORW_5S	R/W	H'00	H'FFFFC8A6	8
定时器状态寄存器 _5S	TSR_5S	R/W	H'00	H'FFFFC8B0	8
定时器中断允许寄存器 _5S	TIER_5S	R/W	H'00	H'FFFFC8B2	8
定时器开始寄存器 _5S	TSTR_5S	R/W	H'00	H'FFFFC8B4	8
定时器比较匹配清除寄存器 S	TCNTCMPCLRS	R/W	H'00	H'FFFFC8B6	8

10.3.1　定时器控制寄存器 （TCR）

TCR 为控制各通道 TCNT 的 8 位可读取 / 写入寄存器。 MTU2 共有 8 个 TCR，通道 0 ～ 4 各有一个，通道 5 有 3 个即 TCRU/V/W_5。必须在 TCNT 运行停止状态设定 TCR。

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	位名称	初始值	R/W	说　明
7 ～ 5	CCLR[2:0]	000	R/W	计数器清除 2、 1、 0 选择 TCNT 的计数器清除源。详情参照表 10.7、表 10.8。
4、 3	CKEG[1:0]	00	R/W	时钟边沿 1、 0 选择输入时钟的边沿。在双边沿计数内部时钟时，输入时钟的周期为 1/2（例如：MPϕ/4 的双边沿 =MPϕ/2 的上升沿）。在通道 1、 2 使用相位计数模式时，忽略本设定，优先相位计数模式的设定。在输入时钟等于或迟于 MPϕ/4 时，内部时钟的边沿选择有效。选择 MPϕ/1 或其他通道的上溢 / 下溢为输入时钟时，可写入值，但运行为初始值。 00：在上升沿计数 01：在下降沿计数 1x：在双边沿计数
2 ～ 0	TPSC[2:0]	000	R/W	定时器预分频器 2、 1、 0 选择 TCNT 计数器时钟，各通道可独立选择时钟源。详情参照表 10.9 ～表 10.12。

【符号说明】 x：Don't care

表 10.7　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	由 TGRA 的比较匹配 / 输入捕捉清除 TCNT
	0	1	0	由 TGRB 的比较匹配 / 输入捕捉清除 TCNT
	0	1	1	通过清除正在同步清除 / 同步运行的其他通道的计数器清除 TCNT*1
	1	0	0	禁止清除 TCNT
	1	0	1	由 TGRC 的比较匹配 / 输入捕捉清除 TCNT*2
	1	1	0	由 TGRD 的比较匹配 / 输入捕捉清除 TCNT*2
	1	1	1	通过清除正在同步清除 / 同步运行的其他通道的计数器清除 TCNT*1

【注】 *1　将 TSYR 的 SYNC 位置 1，由此设定同步运行。

*2　TGRC 或 TGRD 用作缓冲寄存器时，优先设定缓冲寄存器，不产生比较匹配 / 输入捕捉，因此不可清除 TCNT。

表 10.8　CCLR2 ～ CCLR0 （通道 1、 2）

通道	bit7	bit6	bit5	说　明
	保留位 *2	CCLR1	CCLR0	
1、 2	0	0	0	禁止清除 TCNT
	0	0	1	由 TGRA 的比较匹配 / 输入捕捉清除 TCNT
	0	1	0	由 TGRB 的比较匹配 / 输入捕捉清除 TCNT
	0	1	1	通过清除正在同步清除 / 同步运行的其他通道的计数器清除 TCNT*1

【注】　*1　将 TSYR 的 SYNC 位置 1，由此设定同步运行。

*2　通道 1、 2 的 bit7 为保留位。读取值总是为 0，写入无效。

表 10.9　TPSC2 ～ TPSC0 （通道 0）

通道	bit2	bit1	bit0	说　明
	TPSC2	TPSC1	TPSC0	
0	0	0	0	内部时钟：以 MPϕ/1 计数
	0	0	1	内部时钟：以 MPϕ/4 计数
	0	1	0	内部时钟：以 MPϕ/16 计数
	0	1	1	内部时钟：以 MPϕ/64 计数
	1	0	0	外部时钟：由 TCLKA 引脚输入计数
	1	0	1	外部时钟：由 TCLKB 引脚输入计数
	1	1	0	外部时钟：由 TCLKC 引脚输入计数
	1	1	1	外部时钟：由 TCLKD 引脚输入计数

表 10.10　TPSC2 ～ TPSC0 （通道 1）

通道	bit2	bit1	bit0	说　明
	TPSC2	TPSC1	TPSC0	
1	0	0	0	内部时钟：以 MPϕ/1 计数
	0	0	1	内部时钟：以 MPϕ/4 计数
	0	1	0	内部时钟：以 MPϕ/16 计数
	0	1	1	内部时钟：以 MPϕ/64 计数
	1	0	0	外部时钟：由 TCLKA 引脚输入计数
	1	0	1	外部时钟：由 TCLKB 引脚输入计数
	1	1	0	内部时钟：以 MPϕ/256 计数
	1	1	1	通过 TCNT_2 的上溢 / 下溢计数

【注】　通道 1 为相位计数模式时，此设定无效。

表 10.11　TPSC2 ～ TPSC0 （通道 2）

通道	bit2	bit1	bit0	说　明
	TPSC2	TPSC1	TPSC0	
2	0	0	0	内部时钟：以 MPϕ/1 计数
	0	0	1	内部时钟：以 MPϕ/4 计数
	0	1	0	内部时钟：以 MPϕ/16 计数
	0	1	1	内部时钟：以 MPϕ/64 计数
	1	0	0	外部时钟：由 TCLKA 引脚输入计数
	1	0	1	外部时钟：由 TCLKB 引脚输入计数
	1	1	0	外部时钟：由 TCLKC 引脚输入计数
	1	1	1	内部时钟：以 MPϕ/1024 计数

【注】 通道 2 为相位计数模式时，此设定无效。

表 10.12　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：以 MPϕ/1 计数
	0	0	1	内部时钟：以 MPϕ/4 计数
	0	1	0	内部时钟：以 MPϕ/16 计数
	0	1	1	内部时钟：以 MPϕ/64 计数
	1	0	0	内部时钟：以 MPϕ/256 计数
	1	0	1	内部时钟：以 MPϕ/1024 计数
	1	1	0	外部时钟：由 TCLKA 引脚输入计数
	1	1	1	外部时钟：由 TCLKB 引脚输入计数

表 10.13　TPSC1、 TPSC0 （通道 5）

通道	bit1	bit0	说　明
	TPSC1	TPSC0	
5	0	0	内部时钟：以 MPϕ/1 计数
	0	1	内部时钟：以 MPϕ/4 计数
	1	0	内部时钟：以 MPϕ/16 计数
	1	1	内部时钟：以 MPϕ/64 计数

【注】 通道 5 的 bit7 ～ 2 为保留位。读取值、写入值总是为 0。

10.3.2　定时器模式寄存器 （TMDR）

TMDR 为 8 位可读取 / 写入寄存器，可设定各通道的运行模式。 MTU2 共有 5 个 TMDR，通道 0 ～ 4 各有 1 个。必须在 TCNT 运行停止状态设定 TMDR。

位：	7	6	5	4	3	2	1	0
	—	BFE	BFB	BFA	MD[3:0]			
初始值：	0	0	0	0	0	0	0	0
R/W：	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说　明
7	—	0	—	保留位 读取值、写入值总是为 0。
6	BFE	0	R/W	缓冲运行 E 选择 TGRE_0 和 TGRF_0 为普通运行或缓冲运行。 TGRF 用作缓冲寄存器时，也产生 TGRF 的比较匹配。 在通道 1、 2、 3、 4 为保留位。读取值、写入值总是为 0。 0：TGRE_0 和 TGRF_0 为普通运行 1：TGRE_0 和 TGRF_0 为缓冲运行
5	BFB	0	R/W	缓冲运行 B 设定 TGRB 为普通运行或与 TGRD 组合后缓冲运行。 TGRD 用作缓冲寄存器时，除互补 PWM 模式之外，不产生 TGRD 的输入捕捉 / 输出比较，但在互补 PWM 模式产生 TGRD 的比较匹配。另外，比较匹配产生在互补 PWM 模式的 Tb 区间时， TGFD 置位，因此必须将定时器中断允许寄存器 _3/4 （TIER_3/4）的 TGIED 位置 0。 在无 TGRD 的通道 1、 2，此位为保留位。读取值、写入值总是为 0。 0：TGRB 和 TGRD 为普通运行 1：TGRB 和 TGRD 为缓冲运行
4	BFA	0	R/W	缓冲运行 A 设定 TGRA 为普通运行或与 TGRC 组合后缓冲运行。 TGRC 用作缓冲寄存器时，除互补 PWM 模式之外，不产生 TGRC 的输入捕捉 / 输出比较，但在互补 PWM 模式产生 TGRC 的比较匹配。通道 4 的比较匹配产生在互补 PWM 模式的 Tb 区间时， TGFC 置位，因此必须将定时器中断允许寄存器 _4 （TIER_4）的 TGIEC 位置 0。 在无 TGRC 的通道 1、 2，此位为保留位。读取值、写入值总是为 0。 0：TGRA 和 TGRC 为普通运行 1：TGRA 和 TGRC 为缓冲运行
3 ～ 0	MD[3:0]	0000	R/W	模式 3 ～ 0 MD3 ～ MD0 设定定时器的运行模式。 详情参照表 10.14。

表 10.14　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	普通运行
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（在波峰传送）*3
1	1	1	0	互补 PWM 模式 2（在波谷传送）*3
1	1	1	1	互补 PWM 模式 3（在波峰 / 波谷传送）*3

【符号说明】x：Don't care

【注】 *1　在通道 3、 4 不可设定 PWM 模式 2。

*2　在通道 0、 3、 4 不可设定相位计数模式。

*3　仅可在通道 3 设定复位同步 PWM 模式或互补 PWM 模式。

设定通道 3 为复位同步 PWM 模式或互补 PWM 模式时，通道 4 的设定无效，自动遵从通道 3 的设定。但，不得在通道 4 设定复位同步 PWM 模式或互补 PWM 模式。

在通道0、 1、 2不可设定复位同步PWM模式或互补PWM模式。

10.3.3　定时器 I/O 控制寄存器 （TIOR）

TIOR 为控制 TGR 的 8 位可读取 / 写入寄存器。 MTU2 共有 11 个 TIOR，通道 0、 3、 4 各有 2 个，通道 1、 2 各有 1 个，通道 5 有 3 个即 TIORU/V/W_5。

TMDR 为普通运行、 PWM 模式、相位计数模式时，可设定 TIOR。

由 TIOR 指定的初始输出在计数器停止 （TSTR 的 CST 位清 0 后）状态有效。在 PWM 模式 2，指定计数器清 0 时的输出。

设定 TGRC 或 TGRD 为缓冲运行时，本设定无效，作为缓冲寄存器运行。

- TIORH_0、 TIOR_1、 TIOR_2、 TIORH_3、 TIORH_4

位：	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	位名称	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。 请参照下列各表： TIORH_0：表 10.15　TIOR_1：表 10.17　TIOR_2：表 10.18 TIORH_3：表 10.19　TIORH_4：表 10.21
3 ～ 0	IOA[3:0]	0000	R/W	I/O 控制 A3 ～ A0 IOA3 ～ IOA0 设定 TGRA 的功能。 请参照下列各表： TIORH_0：表 10.23　TIOR_1：表 10.25　TIOR_2：表 10.26 TIORH_3：表 10.27　TIORH_4：表 10.29

- 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	位名称	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。 请参照下列各表： TIORL_0：表 10.16　TIORL_3：表 10.20　TIORL_4：表 10.22
3 ～ 0	IOC[3:0]	0000	R/W	I/O 控制 C3 ～ C0 IOC3 ～ IOC0 位设定 TGRC 的功能。 请参照下列各表： TIORL_0：表 10.24　TIORL_3：表 10.28　TIORL_4：表 10.30

- TIORU_5、 TIORV_5、 TIORW_5

位：	7	6	5	4	3	2	1	0
	—	—	—	IOC[4:0]				
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

位	位名称	初始值	R/W	说　明
7 ～ 5	—	均为 0	R	保留位 读取值、写入值总是为 0。
4 ～ 0	IOC[4:0]	00000	R/W	I/O 控制 C4 ～ C0 IOC4 ～ IOC0 位设定 TGRU/V/W_5 的功能。 详情参照表 10.31。

表 10.15　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		初始输出为 0 比较匹配时输出 0
0	0	1	0		初始输出为 0 比较匹配时输出 1
0	0	1	1		初始输出为 0 比较匹配时交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出为 1 比较匹配时输出 0
0	1	1	0		初始输出为 1 比较匹配时输出 1
0	1	1	1		初始输出为 1 比较匹配时交替输出
1	0	0	0	输入捕捉寄存器	在上升沿输入捕捉
1	0	0	1		在下降沿输入捕捉
1	0	1	x		在双边沿输入捕捉
1	1	x	x		捕捉输入源为通道 1/ 计数时钟 由 TCNT_1 的递增 / 递减计数输入捕捉

【符号说明】x：Don't care

【注】 *　上电复位后到设定 TIOR 前，输出 0。

表 10.16　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		初始输出为 0 比较匹配时输出 0
0	0	1	0		初始输出为 0 比较匹配时输出 1
0	0	1	1		初始输出为 0 比较匹配时交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出为 1 比较匹配时输出 0
0	1	1	0		初始输出为 1 比较匹配时输出 1
0	1	1	1		初始输出为 1 比较匹配时交替输出
1	0	0	0	输入捕捉寄存器 *2	在上升沿输入捕捉
1	0	0	1		在下降沿输入捕捉
1	0	1	x		在双边沿输入捕捉
1	1	x	x		捕捉输入源为通道 1/ 计数时钟 由 TCNT_1 的递增 / 递减计数输入捕捉

【符号说明】 x：Don't care

【注】 *1　上电复位后到设定 TIOR 前，输出 0。

*2　TMDR_0 的 BFB 位置 1，并且 TGRD_0 用作缓冲寄存器时，本设定无效，不产生输入捕捉 / 输出比较。

表 10.17　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		初始输出为 0 比较匹配时输出 0
0	0	1	0		初始输出为 0 比较匹配时输出 1
0	0	1	1		初始输出为 0 比较匹配时交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出为 1 比较匹配时输出 0
0	1	1	0		初始输出为 1 比较匹配时输出 1
0	1	1	1		初始输出为 1 比较匹配时交替输出
1	0	0	0	输入捕捉寄存器	在上升沿输入捕捉
1	0	0	1		在下降沿输入捕捉
1	0	1	x		在双边沿输入捕捉
1	1	x	x		产生 TGRC_0 的比较匹配 / 输入捕捉时输入捕捉

【符号说明】 x：Don't care

【注】 *　上电复位后到设定 TIOR 前，输出 0。

表 10.18　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		初始输出为 0 比较匹配时输出 0
0	0	1	0		初始输出为 0 比较匹配时输出 1
0	0	1	1		初始输出为 0 比较匹配时交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出为 1 比较匹配时输出 0
0	1	1	0		初始输出为 1 比较匹配时输出 1
0	1	1	1		初始输出为 1 比较匹配时交替输出
1	x	0	0	输入捕捉寄存器	在上升沿输入捕捉
1	x	0	1		在下降沿输入捕捉
1	x	1	x		在双边沿输入捕捉

【符号说明】 x：Don't care

【注】 *　上电复位后到设定 TIOR 前，输出 0。

表 10.19　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		初始输出为 0 比较匹配时输出 0
0	0	1	0		初始输出为 0 比较匹配时输出 1
0	0	1	1		初始输出为 0 比较匹配时交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出为 1 比较匹配时输出 0
0	1	1	0		初始输出为 1 比较匹配时输出 1
0	1	1	1		初始输出为 1 比较匹配时交替输出
1	x	0	0	输入捕捉寄存器	在上升沿输入捕捉
1	x	0	1		在下降沿输入捕捉
1	x	1	x		在双边沿输入捕捉

【符号说明】 x：Don't care

【注】 *　上电复位后到设定 TIOR 前，输出 0。

表 10.20　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		初始输出为 0 比较匹配时输出 0
0	0	1	0		初始输出为 0 比较匹配时输出 1
0	0	1	1		初始输出为 0 比较匹配时交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出为 1 比较匹配时输出 0
0	1	1	0		初始输出为 1 比较匹配时输出 1
0	1	1	1		初始输出为 1 比较匹配时交替输出
1	x	0	0	输入捕捉寄存器 *2	在上升沿输入捕捉
1	x	0	1		在下降沿输入捕捉
1	x	1	x		在双边沿输入捕捉

【符号说明】 x：Don't care

【注】 *1　上电复位后到设定 TIOR 前，输出 0。

*2　TMDR_3 的 BFB 位置 1，并且 TGRD_3 用作缓冲寄存器时，本设定无效，不产生输入捕捉 / 输出比较。

表 10.21　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		初始输出为 0 比较匹配时输出 0
0	0	1	0		初始输出为 0 比较匹配时输出 1
0	0	1	1		初始输出为 0 比较匹配时交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出为 1 比较匹配时输出 0
0	1	1	0		初始输出为 1 比较匹配时输出 1
0	1	1	1		初始输出为 1 比较匹配时交替输出
1	x	0	0	输入捕捉寄存器	在上升沿输入捕捉
1	x	0	1		在下降沿输入捕捉
1	x	1	x		在双边沿输入捕捉

【符号说明】 x：Don't care

【注】 *　上电复位后到设定 TIOR 前，输出 0。

表 10.22　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		初始输出为 0 比较匹配时输出 0
0	0	1	0		初始输出为 0 比较匹配时输出 1
0	0	1	1		初始输出为 0 比较匹配时交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出为 1 比较匹配时输出 0
0	1	1	0		初始输出为 1 比较匹配时输出 1
0	1	1	1		初始输出为 1 比较匹配时交替输出
1	x	0	0	输入捕捉寄存器 *2	在上升沿输入捕捉
1	x	0	1		在下降沿输入捕捉
1	x	1	x		在双边沿输入捕捉

【符号说明】 x：Don't care

【注】 *1　上电复位后到设定 TIOR 前，输出 0。

*2　TMDR_4 的 BFB 位置 1，并且 TGRD_4 用作缓冲寄存器时，本设定无效，不产生输入捕捉 / 输出比较。

表 10.23　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		初始输出为 0 比较匹配时输出 0
0	0	1	0		初始输出为 0 比较匹配时输出 1
0	0	1	1		初始输出为 0 比较匹配时交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出为 1 比较匹配时输出 0
0	1	1	0		初始输出为 1 比较匹配时输出 1
0	1	1	1		初始输出为 1 比较匹配时交替输出
1	0	0	0	输入捕捉寄存器	在上升沿输入捕捉
1	0	0	1		在下降沿输入捕捉
1	0	1	x		在双边沿输入捕捉
1	1	x	x		捕捉输入源为通道 1 / 计数时钟 由 TCNT_1 的递增 / 递减计数输入捕捉

【符号说明】 x：Don't care

【注】 *　上电复位后到设定 TIOR 前，输出 0。

表 10.24　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		初始输出为 0 比较匹配时输出 0
0	0	1	0		初始输出为 0 比较匹配时输出 1
0	0	1	1		初始输出为 0 比较匹配时交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出为 1 比较匹配时输出 0
0	1	1	0		初始输出为 1 比较匹配时输出 1
0	1	1	1		初始输出为 1 比较匹配时交替输出
1	0	0	0	输入捕捉寄存器 *2	在上升沿输入捕捉
1	0	0	1		在下降沿输入捕捉
1	0	1	x		在双边沿输入捕捉
1	1	x	x		捕捉输入源为通道 1/ 计数时钟 由 TCNT_1 的递增 / 递减计数输入捕捉

【符号说明】 x：Don't care

【注】 *1　上电复位后到设定 TIOR 前，输出 0。

*2　TMDR_0 的 BFA 位置 1，并且 TGRC_0 用作缓冲寄存器时，本设定无效，不产生输入捕捉 / 输出比较。

表 10.25　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		初始输出为 0 比较匹配时输出 0
0	0	1	0		初始输出为 0 比较匹配时输出 1
0	0	1	1		初始输出为 0 比较匹配时交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出为 1 比较匹配时输出 0
0	1	1	0		初始输出为 1 比较匹配时输出 1
0	1	1	1		初始输出为 1 比较匹配时交替输出
1	0	0	0	输入捕捉寄存器	在上升沿输入捕捉
1	0	0	1		在下降沿输入捕捉
1	0	1	x		在双边沿输入捕捉
1	1	x	x		产生 TGRA_0 的比较匹配 / 输入捕捉时执行输入捕捉

【符号说明】 x：Don't care

【注】 *　上电复位后到设定 TIOR 前，输出 0。

表 10.26　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		初始输出为 0 比较匹配时输出 0
0	0	1	0		初始输出为 0 比较匹配时输出 1
0	0	1	1		初始输出为 0 比较匹配时交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出为 1 比较匹配时输出 0
0	1	1	0		初始输出为 1 比较匹配时输出 1
0	1	1	1		初始输出为 1 比较匹配时交替输出
1	x	0	0	输入捕捉寄存器	在上升沿输入捕捉
1	x	0	1		在下降沿输入捕捉
1	x	1	x		在双边沿输入捕捉

【符号说明】 x：Don't care

【注】 *　上电复位后到设定 TIOR 前，输出 0。

表 10.27　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		初始输出为 0 比较匹配时输出 0
0	0	1	0		初始输出为 0 比较匹配时输出 1
0	0	1	1		初始输出为 0 比较匹配时交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出为 1 比较匹配时输出 0
0	1	1	0		初始输出为 1 比较匹配时输出 1
0	1	1	1		初始输出为 1 比较匹配时交替输出
1	x	0	0	输入捕捉寄存器	在上升沿输入捕捉
1	x	0	1		在下降沿输入捕捉
1	x	1	x		在双边沿输入捕捉

【符号说明】 x：Don't care

【注】 *　上电复位后到设定 TIOR 前，输出 0。

表 10.28　TIORL_3（通道 3）

bit3	bit2	bit1	bit0	说　明	
IOC3	IOC2	IOC1	IOC0	TGRC_3 的功能	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		初始输出为 0 比较匹配时输出 0
0	0	1	0		初始输出为 0 比较匹配时输出 1
0	0	1	1		初始输出为 0 比较匹配时交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出为 1 比较匹配时输出 0
0	1	1	0		初始输出为 1 比较匹配时输出 1
0	1	1	1		初始输出为 1 比较匹配时交替输出
1	x	0	0	输入捕捉寄存器 *2	在上升沿输入捕捉
1	x	0	1		在下降沿输入捕捉
1	x	1	x		在双边沿输入捕捉

【符号说明】 x：Don't care

【注】 *1　上电复位后到设定 TIOR 前，输出 0。

*2　TMDR_3 的 BFA 位置 1，并且 TGRC_3 用作缓冲寄存器时，本设定无效，不产生输入捕捉 / 输出比较。

表 10.29　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		初始输出为 0 比较匹配时输出 0
0	0	1	0		初始输出为 0 比较匹配时输出 1
0	0	1	1		初始输出为 0 比较匹配时交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出为 1 比较匹配时输出 0
0	1	1	0		初始输出为 1 比较匹配时输出 1
0	1	1	1		初始输出为 1 比较匹配时交替输出
1	x	0	0	输入捕捉寄存器	在上升沿输入捕捉
1	x	0	1		在下降沿输入捕捉
1	x	1	x		在双边沿输入捕捉

【符号说明】 x：Don't care

【注】 *　上电复位后到设定 TIOR 前，输出 0。

表 10.30　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		初始输出为 0 比较匹配时输出 0
0	0	1	0		初始输出为 0 比较匹配时输出 1
0	0	1	1		初始输出为 0 比较匹配时交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出为 1 比较匹配时输出 0
0	1	1	0		初始输出为 1 比较匹配时输出 1
0	1	1	1		初始输出为 1 比较匹配时交替输出
1	x	0	0	输入捕捉寄存器 *2	在上升沿输入捕捉
1	x	0	1		在下降沿输入捕捉
1	x	1	x		在双边沿输入捕捉

【符号说明】 x：Don't care

【注】 *1　上电复位后到设定 TIOR 前，输出 0。

*2　TMDR_4 的 BFA 位置 1，并且 TGRC_4 用作缓冲寄存器时，本设定无效，不产生输入捕捉 / 输出比较。

表 10.31　TIORU_5、 TIORV_5、 TIORW_5 （通道 5）

bit4	bit3	bit2	bit1	bit0	说　明	
IOC4	IOC3	IOC2	IOC1	IOC0	TGRU_5、 TGRV_5、 TGRW_5 的功能	TIC5U、 TIC5V、 TIC5W 引脚的功能
0	0	0	0	0	比较匹配寄存器	比较匹配
0	0	0	0	1		禁止设定
0	0	0	1	x		禁止设定
0	0	1	x	x		禁止设定
0	1	x	x	x		禁止设定
1	0	0	0	0	输入捕捉寄存器	禁止设定
1	0	0	0	1		在上升沿输入捕捉
1	0	0	1	0		在下降沿输入捕捉
1	0	0	1	1		在双边沿输入捕捉
1	0	1	x	x		禁止设定
1	1	0	0	0		禁止设定
1	1	0	0	1		用于测量外部输入信号的低电平脉宽 在互补 PWM 模式的波谷捕捉
1	1	0	1	0		用于测量外部输入信号的低电平脉宽 在互补 PWM 模式的波峰捕捉
1	1	0	1	1		用于测量外部输入信号的低电平脉宽 在互补 PWM 模式的波峰和波谷捕捉
1	1	1	0	0		禁止设定
1	1	1	0	1		用于测量外部输入信号的高电平脉宽 在互补 PWM 模式的波谷捕捉
1	1	1	1	0		用于测量外部输入信号的高电平脉宽 在互补 PWM 模式的波峰捕捉
1	1	1	1	1		用于测量外部输入信号的高电平脉宽 在互补 PWM 模式的波峰和波谷捕捉

【符号说明】 x：Don't care

10.3.4　定时器比较匹配清除寄存器 （TCNTCMPCLR）

TCNTCMPCLR 为 8 位可读取 / 写入的寄存器，可设定 TCNTU_5、 TCNTV_5、 TCNTW_5 的清除请求。MTU2 的通道 5 有 1 个 TCNTCMPCLR。

位：	7	6	5	4	3	2	1	0
	—	—	—	—	—	CMPCLR5U	CMPCLR5V	CMPCLR5W
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	位名称	初始值	R/W	说　明
7 ～ 3	—	均为 0	R	保留位 读取值、写入值总是为 0。
2	CMPCLR5U	0	R/W	TCNT 比较清除 5U 通过 TGRU_5 的比较匹配 / 输入捕捉，允许或禁止 TCNTU_5 的清除请求。 0：通过 TCNTU_5 和 TGRU_5 的比较匹配 / 输入捕捉，禁止 TCNTU_5 清除为 H'0000 1：通过 TCNTU_5 和 TGRU_5 的比较匹配 / 输入捕捉，允许 TCNTU_5 清除为 H'0000
1	CMPCLR5V	0	R/W	TCNT 比较清除 5V 通过 TGRV_5 的比较匹配 / 输入捕捉，允许或禁止 TCNTV_5 的清除请求。 0：通过 TCNTV_5 和 TGRV_5 的比较匹配 / 输入捕捉，禁止 TCNTV_5 清除为 H'0000 1：通过 TCNTV_5 和 TGRV_5 的比较匹配 / 输入捕捉，允许 TCNTV_5 清除为 H'0000
0	CMPCLR5W	0	R/W	TCNT 比较清除 5W 通过 TCNTW_5 和 TGRW_5 的比较匹配 / 输入捕捉，允许或禁止 TCNTW_5 的清除请求。 0：通过根据 TCNTW_5 和 TGRW_5 的比较匹配 / 输入捕捉，禁止 TCNTW_5 清除为 H'0000 1：通过 TCNTW_5 和 TGRW_5 的比较匹配 / 输入捕捉，允许 TCNTW_5 清除为 H'0000

10.3.5　定时器中断允许寄存器 （TIER）

TIER 为 8 位可读取 / 写入的寄存器，控制允许或禁止各通道的中断请求。 MTU2 共有 7 个 TIER，通道 0 有 2 个，通道 1 ～ 5 各有 1 个。

- 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	位名称	初始值	R/W	说　明
7	TTGE	0	R/W	A/D 转换开始请求允许 通过 TGRA 的输入捕捉 / 比较匹配，允许或禁止产生 A/D 转换开始请求。 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 转换开始请求允许 2 在互补 PWM 模式，允许或禁止由 TCNT_4 的下溢 （波谷）产生 A/D 转换请求。 在通道 0 ～ 3 为保留位。读取值、写入值总是为 0。 0：禁止由 TCNT_4 的下溢 （波谷）产生 A/D 转换请求 1：允许由 TCNT_4 的下溢 （波谷）产生 A/D 转换请求
5	TCIEU	0	R/W	下溢中断允许 在通道 1、 2， TSR 的 TCFU 标志置 1 时，允许或禁止由 TCFU 标志产生中断请求 （TCIU）。 在通道 0、 3、 4 为保留位。读取值、写入值总是为 0。 0：禁止由 TCFU 产生中断请求 （TCIU） 1：允许由 TCFU 产生中断请求 （TCIU）
4	TCIEV	0	R/W	上溢中断允许 TSR 的 TCFV 标志置 1 时，允许或禁止由 TCFV 标志产生中断请求 （TCIV）。 0：禁止由 TCFV 产生中断请求 （TCIV） 1：允许由 TCFV 产生中断请求 （TCIV）

位	位名称	初始值	R/W	说　明
3	TGIED	0	R/W	TGR 中断允许 D 在通道 0、3、4，TSR 的 TGFD 位置 1 时，允许或禁止由 TGFD 位产生中断请求 （TGID）。 在通道 1、 2 为保留位，读取值、写入值总是为 0。 0：禁止由 TGFD 位产生中断请求 （TGID） 1：允许由 TGFD 位产生中断请求 （TGID）
2	TGIEC	0	R/W	TGR 中断允许 C 在通道 0、3、4，TSR 的 TGFC 位置 1 时，允许或禁止由 TGFC 位产生中断请求 （TGIC）。 在通道 1、 2 为保留位，读取值、写入值总是为 0。 0：禁止由 TGFC 位产生中断请求 （TGIC） 1：允许由 TGFC 位产生中断请求 （TGIC）
1	TGIEB	0	R/W	TGR 中断允许 B TSR 的 TGFB 位置 1 时，允许或禁止由 TGFB 位产生中断请求 （TGIB）。 0：禁止由 TGFB 位产生中断请求 （TGIB） 1：允许由 TGFB 位产生中断请求 （TGIB）
0	TGIEA	0	R/W	TGR 中断允许 A TSR 的 TGFA 位置 1 时，允许或禁止由 TGFA 位产生中断请求 （TGIA）。 0：禁止由 TGFA 位产生中断请求 （TGIA） 1：允许由 TGFA 位产生中断请求 （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	位名称	初始值	R/W	说　明
7	TTGE2	0	R/W	A/D 转换开始请求允许 2 允许或禁止由 TCNT_0 和 TGRE_0 的比较匹配产生 A/D 转换开始请求。 0：禁止由 TCNT_0 和 TGRE_0 的比较匹配产生 A/D 转换开始请求 1：允许由 TCNT_0 和 TGRE_0 的比较匹配产生 A/D 转换开始请求
6 ～ 2	—	均为 0	R	保留位 读取值、写入值总是为 0。
1	TGIEF	0	R/W	TGR 中断允许 F 允许或禁止由 TCNT_0 和 TGRF_0 的比较匹配产生中断请求。 0：禁止由 TGFE 位产生中断请求 （TGIF） 1：允许由 TGFE 位产生中断请求 （TGIF）
0	TGIEE	0	R/W	TGR 中断允许 E 允许或禁止由 TCNT_0 和 TGRE_0 的比较匹配产生中断请求。 0：禁止由 TGEE 位产生中断请求 （TGIE） 1：允许由 TGEE 位产生中断请求 （TGIE）

- TIER_5

位：	7	6	5	4	3	2	1	0
	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W
初始值：	0	0	0	0	0	0	0	0
R/W：	R	R	R	R	R	R/W	R/W	R/W

位	位名称	初始值	R/W	说　明
7 ～ 3	—	均为 0	R	保留位 读取值、写入值总是为 0。
2	TGIE5U	0	R/W	TGR 中断允许 5U TSR_5 的 CMFU5 位置 1 时，允许或禁止由 CMFU5 位产生中断请求（TGIU_5）。 0：禁止 TGIU_5 中断请求 1：允许 TGIU_5 中断请求
1	TGIE5V	0	R/W	TGR 中断允许 5V TSR_5 的 CMFV5 位置 1 时，允许或禁止由 CMFV5 位产生中断请求（TGIV_5）。 0：禁止 TGIV_5 中断请求 1：允许 TGIV_5 中断请求
0	TGIE5W	0	R/W	TGR 中断允许 5W TSR_5 的 CMFW5 位置 1 时，允许或禁止由 CMFW5 位产生中断请求（TGIW_5）。 0：禁止 TGIW_5 中断请求 1：允许 TGIW_5 中断请求

10.3.6　定时器状态寄存器 （TSR）

TSR 为 8 位可读取 / 写入的寄存器，表示各通道的状态。 MTU2 共有 7 个 TSR，通道 0 有 2 个，通道 1 ～ 5 各有 1 个。

- TSR_0、 TSR_1、 TSR_2、 TSR_3、 TSR_4

位：	7	6	5	4	3	2	1	0
	TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA
初始值：	1	1	0	0	0	0	0	0
R/W：	R	R	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1

【注】*1　为了清除标志，仅可在读取1后写入0。

位	位名称	初始值	R/W	说　明
7	TCFD	1	R	计数方向标志 表示通道 1 ～ 4 的 TCNT 计数方向的状态。 在通道 0 为保留位，读取值、写入值总是为 1。 0：TCNT 为递减计数 1：TCNT 为递增计数
6	—	1	R	保留位 读取值、写入值总是为 1。
5	TCFU	0	R/(W)*1	下溢标志 通道 1、 2 为相位计数模式时，表示 TCNT 产生下溢的状态。为了清除标志，仅可写入 0。 在通道 0、 3、 4 为保留位，读取值、写入值总是为 0。 ［置位条件］ • TCNT 的值下溢 （H'0000 → H'FFFF）时 ［清除条件］ • 在 TCFU=1 的状态，读取 TCFU 后，对 TCFU 写入 0 时 *2
4	TCFV	0	R/(W)*1	上溢标志 表示 TCNT 产生上溢的状态。为了清除标志，仅可写入 0。 ［置位条件］ • TCNT 的值上溢 （H'FFFF → H'0000）时 对于通道 4，在互补 PWM 模式 TCNT_4 的值下溢 （H'0001 → H'0000）时，此标志也置位。 ［清除条件］ • 在 TCFV=1 的状态，读取 TCFV 后，对 TCFV 写入 0 时 *2 对于通道 4，由 TCIV 中断启动 DTC，DTC 的 MRB 的 DISEL 位为 0 时，此位也被清除。
3	TGFD	0	R/(W)*1	输入捕捉 / 输出比较标志 D 表示通道 0、 3、 4 产生 TGRD 输入捕捉或比较匹配的状态。为了清除标志，仅可写入 0。 在通道 1、 2 为保留位。读取值、写入值总是为 0。 ［置位条件］ • TGRD 用作输出比较寄存器的情况下， TCNT=TGRD 时 • TGRD 用作输入捕捉的情况下，通过输入捕捉信号向 TGRD 传送 TCNT 值时 ［清除条件］ • 由 TGID 中断启动 DTC， DTC 的 MRB 的 DISEL 位为 0 时 • TGFD=1 时，读取 TGFD 后，对 TGFD 写入 0 时 *2

位	位名称	初始值	R/W	说　明
2	TGFC	0	R/(W)*1	输入捕捉 / 输出比较标志 C 表示通道 0、 3、 4 产生 TGRC 输入捕捉或比较匹配的状态。为了清除标志，仅可写入 0。 在通道 1、 2 为保留位。读取值、写入值总是为 0。 ［置位条件］ • TGRC 用作输出比较寄存器的情况下， TCNT=TGRC 时 • TGRC 用作输入捕捉的情况下，通过输入捕捉信号向 TGRC 传送 TCNT 值时 ［清除条件］ • 由 TGIC 中断启动 DTC， DTC 的 MRB 的 DISEL 位为 0 时 • TGFC=1 时，读取 TGFC 后，对 TGFC 写入 0 时 *2
1	TGFB	0	R/(W)*1	输入捕捉 / 输出比较标志 B 表示产生 TGRB 输入捕捉或输出比较的状态。为了清除标志，仅可写入 0。 ［置位条件］ • TGRB 用作输出比较寄存器的情况下， TCNT=TGRB 时 • TGRB 用作输入捕捉的情况下，由输入捕捉信号向 TGRB 传送 TCNT 值时 ［清除条件］ • 由 TGIB 中断启动 DTC， DTC 的 MRB 的 DISEL 位为 0 时 • TGFB=1 时，读取 TGFB 后，对 TGFB 写入 0 时 *2
0	TGFA	0	R/(W)*1	输入捕捉 / 输出比较标志 A 表示产生 TGRA 输入捕捉或比较匹配的状态。为了清除标志，仅可写入 0。 ［置位条件］ • TGRA 用作输出比较寄存器的情况下， TCNT=TGRA 时 • TGRA 用作输入捕捉的情况下，由输入捕捉信号向 TGRA 传送 TCNT 值时 ［清除条件］ • 由 TGIA 中断启动 DTC，DTC 的 MRB 的 DISEL 位为 0 时 • TGFA=1 时，读取 TGFA 后，对 TGFA 写入 0 时 *2

【注】 *1　为了清除标志，仅可在读取 1 后写入 0。

*2　如果读取 1 后写入 0 前置位下一个标志，即使写入 0 也不可清除标志，因此必须再次读取 1 后写入 0。

- TSR2_0

位：	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TGFF	TGFE
初始值：	1	1	0	0	0	0	0	0
R/W：	R	R	R	R	R	R	R/(W)*1	R/(W)*1

【注】*1　为了清除标志，仅可在读取1后写入0。

位	位名称	初始值	R/W	说　明
7、6	—	均为 1	R	保留位 读取值、写入值总是为 1。
5 ～ 2	—	均为 0	R	保留位 读取值、写入值总是为 0。
1	TGFF	0	R/(W)*1	比较匹配标志 F 表示产生 TCNT_0 和 TGRF_0 比较匹配的状态。 ［置位条件］ • TGRF_0 用作比较寄存器的情况下， TCNT_0=TGRF_0 时 ［清除条件］ • TGFF=1 时，读取 TGFF 后，对 TGFF 写入 0 时 *2
0	TGFE	0	R/(W)*1	比较匹配标志 E 表示产生 TCNT_0 和 TGRE_0 比较匹配的状态。 ［置位条件］ • TGRE_0 用作比较寄存器的情况下， TCNT_0=TGRE_0 时 ［清除条件］ • TGFE=1 时，读取 TGFE 后，对 TGFE 写入 0 时 *2

【注】　*1　为了清除标志，仅可读取 1 后再写入 0。

*2　如果读取 1 后写入 0 前置位下一个比较匹配的标志，即使写入 0 也不可清除标志，因此必须再次读取 1 后写入 0。

- TSR_5

位：	7	6	5	4	3	2	1	0
	—	—	—	—	—	CMFU5	CMFV5	CMFW5
初始值：	0	0	0	0	0	0	0	0
R/W：	R	R	R	R	R	R/(W)*1	R/(W)*1	R/(W)*1

【注】*1　为了清除标志，仅可在读取1后写入0。

位	位名称	初始值	R/W	说　明
7 ～ 3	—	均为 0	R	保留位 读取值、写入值总是为 0。
2	CMFU5	0	R/(W)*1	比较匹配 / 输入捕捉标志 U5 表示产生 TGRU_5 输入捕捉或比较匹配的状态。为了清除标志，仅可写入 0。 ［置位条件］ • TGRU_5 用作比较匹配寄存器的情况下， TCNTU_5=TGRU_5 时 • TGRU_5 用作输入捕捉的情况下，由输入捕捉信号向 TGRU_5 传送 TCNTU_5 的值时 • TGRU_5 用于测量外部输入信号脉宽的情况下，向 TGRU_5 传送 TCNTU_5 的值时 *2 ［清除条件］ • 由 TGIU_5 中断启动 DTC， DTC 的 MRB 的 DISEL 位为 0 时 • CMFU5=1 时，读取 CMFU5 后，对 CMFU5 写入 0 时
1	CMFV5	0	R/(W)*1	比较匹配 / 输入捕捉标志 V5 表示产生 TGRV_5 输入捕捉或比较匹配的状态。为了清除标志，仅可写入 0。 ［置位条件］ • TGRV_5 用作比较匹配寄存器的情况下， TCNTV_5=TGRV_5 时 • TGRV_5 用作输入捕捉的情况下，由输入捕捉信号向 TGRV_5 传送 TCNTV_5 的值时 • TGRV_5 用于测量外部输入信号脉宽的情况下，向 TGRV_5 传送 TCNTV_5 的值时 *2 ［清除条件］ • 由 TGIV_5 中断启动 DTC， DTC 的 MRB 的 DISEL 位为 0 时 • CMFV5=1 时，读取 CMFV5 后，对 CMFV5 写入 0 时
0	CMFW5	0	R/(W)*1	比较匹配 / 输入捕捉标志 W5 表示产生 TGRW_5 输入捕捉或比较匹配的状态。 ［置位条件］ • TGRW_5 用作比较匹配寄存器的情况下， TCNTW_5=TGRW_5 时 • TGRW_5 用作输入捕捉的情况下，由输入捕捉信号向 TGRW_5 传送 TCNTW_5 的值时 • TGRW_5 用于测量外部输入信号的脉宽的情况下，向 TGRW_5 传送 TCNTW_5 的值时 *2 ［清除条件］ • 由 TGIW_5 中断启动 DTC， DTC 的 MRB 的 DISEL 位为 0 时 • CMFW5=1 时，读取 CMFW5 后，对 CMFW5 写入 0 时

【注】　*1　为了清除标志，仅可读取 1 后再写入 0。

*2　通过定时器 I/O 控制寄存器 U_5/V_5/W_5 （TIORU_5/V_5/W_5）的 IOC 位设定传送时序。

10.3.7　定时器缓冲运行传送模式寄存器 （TBTM）

TBTM 为 8 位可读取 / 写入的寄存器，设定从 PWM 模式的缓冲寄存器向定时器通用寄存器传送的时序。MTU2 共有 3 个 TBTM，通道 0、 3、 4 各有 1 个。

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	位名称	初始值	R/W	说　明
7 ～ 3	—	均为 0	R	保留位 读取值、写入值总是为 0。
2	TTSE	0	R/W	时序选择 E 设定缓冲运行时从 TGRF_0 传送到 TGRE_0 的时序。在通道 3、 4 为保留位。读取值、写入值总是为 0。在 PWM 模式以外使用通道 0 时，本位不得设定为 1。 0：产生通道 0 的比较匹配 E 时 1：清除 TCNT_0 时
1	TTSB	0	R/W	时序选择 B 设定各通道缓冲运行时从 TGRD 传送到 TGRB 的时序。在 PWM 模式以外使用的通道，本位不得设定为 1。 0：产生各通道的比较匹配 B 时 1：清除各通道的 TCNT 时
0	TTSA	0	R/W	时序选择 A 设定各通道缓冲运行时从 TGRC 传送到 TGRA 的时序。在 PWM 模式以外使用的通道，本位不得设定为 1。 0：产生各通道的比较匹配 A 时 1：清除各通道的 TCNT 时

10.3.8　定时器输入捕捉控制寄存器 （TICCR）

TICCR 为 8 位可读取 / 写入的寄存器，控制 TCNT_1 和 TCNT_2 级联时的输入捕捉条件。MTU2 的通道 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	位名称	初始值	R/W	说　明
7 ～ 4	—	均为 0	R	保留位 读取值、写入值总是为 0。
3	I2BE	0	R/W	输入捕捉允许 选择是否将 TIOC2B 引脚追加到 TGRB_1 的输入捕捉条件。 0：TIOC2B 引脚未追加到 TGRB_1 的输入捕捉条件 1：TIOC2B 引脚追加到 TGRB_1 的输入捕捉条件
2	I2AE	0	R/W	输入捕捉允许 选择是否将 TIOC2A 引脚追加到 TGRA_1 的输入捕捉条件 0：TIOC2A 引脚未追加到 TGRA_1 的输入捕捉条件 1：TIOC2A 引脚追加到 TGRA_1 的输入捕捉条件
1	I1BE	0	R/W	输入捕捉允许 选择是否将 TIOC1B 引脚追加到 TGRB_2 的输入捕捉条件 0：TIOC1B 引脚未追加到 TGRB_2 的输入捕捉条件 1：TIOC1B 引脚追加到 TGRB_2 的输入捕捉条件
0	I1AE	0	R/W	输入捕捉允许 选择是否将 TIOC1A 引脚追加到 TGRA_2 的输入捕捉条件 0：TIOC1A 引脚未追加到 TGRA_2 的输入捕捉条件 1：TIOC1A 引脚追加到 TGRA_2 的输入捕捉条件

10.3.9　定时器同步清除寄存器 （TSYCR）

TSYCR 为 8 位可读取 / 写入的寄存器，设定来自 MTU2 的 MTU2S 的 TCNT_3、 TCNT_4 同步清除条件。MTU2S 的通道 3 有 1 个 TSYCR， MTU2 无 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	位名称	初始值	R/W	说　明
7	CE0A	0	R/W	清除允许 0A 设定禁止 / 允许通过 MTU2 的 TSR_0 的 TGFA 标志置位清除。 0：禁止通过 TSR_0 的 TGFA 标志置位清除 1：允许通过 TSR_0 的 TGFA 标志置位清除
6	CE0B	0	R/W	清除允许 0B 设定禁止 / 允许通过 MTU2 的 TSR_0 的 TGFB 标志置位清除。 0：禁止通过 TSR_0 的 TGFB 标志置位清除 1：允许通过 TSR_0 的 TGFB 标志置位清除
5	CE0C	0	R/W	清除的允许 0C 设定禁止 / 允许通过 MTU2 中 TSR_0 的 TGFC 标志置位清除。 0：禁止通过 TSR_0 的 TGFC 标志置位清除 1：允许通过 TSR_0 的 TGFC 标志置位清除
4	CE0D	0	R/W	清除允许 0D 设定禁止 / 允许通过 MTU2 的 TSR_0 的 TGFD 标志置位清除。 0：禁止通过 TSR_0 的 TGFD 标志置位清除 1：允许通过 TSR_0 的 TGFD 标志置位清除
3	CE1A	0	R/W	清除允许 1A 设定禁止 / 允许通过设定 MTU2 的 TSR_1 的 TGFA 标志置位清除。 0：禁止通过 TSR_1 的 TGFA 标志置位清除 1：允许通过 TSR_1 的 TGFA 标志置位清除
2	CE1B	0	R/W	清除允许 1B 设定禁止 / 允许通过 MTU2 中 TSR_1 的 TGFB 标志置位清除。 0：禁止通过 TSR_1 的 TGFB 标志置位清除 1：允许通过 TSR_1 的 TGFB 标志置位清除
1	CE2A	0	R/W	清除允许 2A 设定禁止 / 允许通过 MTU2 的 TSR_2 的 TGFA 标志置位清除。 0：禁止通过 TSR_2 的 TGFA 标志置位清除 1：允许通过 TSR_2 的 TGFA 标志置位清除
0	CE2B	0	R/W	清除允许 2B 设定禁止 / 允许通过 MTU2 的 TSR_2 的 TGFB 标志置位清除。 0：禁止通过 TSR_2 的 TGFB 标志置位清除 1：允许通过 TSR_2 的 TGFB 标志置位清除

10.3.10　定时器 A/D 转换开始请求控制寄存器 （TADCR）

TADCR 为 16 位可读取 / 写入的寄存器，设定允许 / 禁止 A/D 转换开始请求及是否联动中断跳过和 A/D 转换开始请求。 MTU2 的通道 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】* 除互补PWM模式外，不可设定为1。

位	位名称	初始值	R/W	说　明
15、 14	BF[1:0]	00	R/W	TADCOBRA/B_4 传送时序选择 选择从 TADCOBRA/B_4 传送到 TADCORA/B_4 的时序。 详情参照表 10.32。
13 ～ 8	—	均为 0	R	保留位 读取值、写入值总是为 0。
7	UT4AE	0	R/W	递增计数 TRG4AN 允许 TCNT_4 递增计数时，设定允许 / 禁止 A/D 转换开始请求 （TRG4AN）。 0：TCNT_4 递增计数时，禁止 A/D 转换的开始请求 （TRG4AN） 1：TCNT_4 递增计数时，允许 A/D 转换的开始请求 （TRG4AN）
6	DT4AE	0*	R/W	递减计数 TRG4AN 允许 TCNT_4 递减计数时，设定允许 / 禁止 A/D 转换开始请求 （TRG4AN）。 0：TCNT_4 递减计数时，禁止 A/D 转换的开始请求 （TRG4AN） 1：TCNT_4 递减计数时，允许 A/D 转换的开始请求 （TRG4AN）
5	UT4BE	0	R/W	递增计数 TRG4BN 允许 TCNT_4 递增计数时，设定允许 / 禁止 A/D 转换开始请求 （TRG4BN）。 0：TCNT_4 递增计数时，禁止 A/D 转换的开始请求 （TRG4BN） 1：TCNT_4 递增计数时，允许 A/D 转换的开始请求 （TRG4BN）
4	DT4BE	0*	R/W	递减计数 TRG4BN 允许 TCNT_4 递减计数时，设定允许 / 禁止 A/D 转换开始请求 （TRG4BN）。 0：TCNT_4 递减计数时，禁止 A/D 转换的开始请求 （TRG4BN） 1：TCNT_4 递减计数时，允许 A/D 转换的开始请求 （TRG4BN）
3	ITA3AE	0*	R/W	TGIA_3 中断跳过联动允许 选择 A/D 转换的开始请求 （TRG4AN）是否与 TGIA_3 中断跳过功能联动。 0：不与 TGIA_3 中断跳过功能联动 1：与 TGIA_3 中断跳过功能联动
2	ITA4VE	0*	R/W	TCIV_4 中断跳过联动允许 选择 A/D 转换的开始请求 （TRG4AN）是否与 TCIV_4 中断跳过功能联动。 0：不与 TCIV_4 中断跳过功能联动 1：与 TCIV_4 中断跳过功能联动

位	位名称	初始值	R/W	说　明
1	ITB3AE	0*	R/W	TGIA_3 中断跳过联动允许 选择 A/D 转换的开始请求 （TRG4BN）是否与 TGIA_3 中断跳过功能联动。 0：不与 TGIA_3 中断跳过功能联动 1：与 TGIA_3 中断跳过功能联动
0	ITB4VE	0*	R/W	TCIV_4 中断跳过联动允许 选择 A/D 转换的开始请求 （TRG4BN）是否与 TCIV_4 中断跳过功能联动。 0：不与 TCIV_4 中断跳过功能联动 1：与 TCIV_4 中断跳过功能联动

【注】 1. 禁止以 8 位为单位存取 TADCR，必须以 16 位为单位存取。

2. 禁止中断跳过时 （将定时器中断跳过设定寄存器 （TITCR）的 T3AEN、 T4VEN 位设定为 0，或将 TITCR 的跳过次数设定位 （3ACOR、 4VCOR）设定为 0 时），不可设定为与中断跳过功能联动 （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. 禁止中断跳过时，如果已设定为与中断跳过联动，则不执行 A/D 转换开始请求。

* 除互补 PWM 模式之外，不可设定为 1。

表 10.32　由 BF1、 BF0 位设定传送时序

bit7	bit6	说　明
BF1	BF0	
0	0	不从周期设定缓冲寄存器向周期设定寄存器传送
0	1	在 TCNT_4 的波峰，从周期设定缓冲寄存器向周期设定寄存器传送 *1
1	0	在 TCNT_4 的波谷，从周期设定缓冲寄存器向周期设定寄存器传送 *2
1	1	在 TCNT_4 的波峰和波谷，从周期设定缓冲寄存器向周期设定寄存器传送 *2

【注】 *1 在互补 PWM 模式的 TCNT_4 的波峰及复位同步 PWM 模式的 TCNT_3 与 TGRA_3 比较匹配时，或 PWM 模式 1/ 普通运行模式的 TCNT_4 与 TGRA_4 比较匹配时，从周期设定缓冲寄存器向周期设定寄存器传送。

*2 除互补 PWM 模式之外，禁止设定。

10.3.11　定时器 A/D 转换开始请求周期设定寄存器 （TADCORA/B_4）

TADCORA/B_4 为 16 位可读取 / 写入的寄存器。与 TCNT_4 匹配时，产生相应的 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，总是以16位为单位存取。

10.3.12　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 为 16 位可读取 / 写入的寄存器。在波峰或波谷从 TADCORA/B_4 的缓冲寄存器向 TADCORA/B_4 传送。

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，总是以16位为单位存取。

10.3.13　定时器计数器 （TCNT）

TCNT 为 16 位可读取 / 写入的计数器，共有 8 个，通道 0 ～ 4 各有 1 个，通道 5 有 3 个即 TCNTU/V/W_5。

复位时， TCNT 初始化为 H'0000。

禁止以 8 位为单位存取 TCNT，总是以 16 位为单位存取。

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

【注】　禁止以8位为单位存取TCNT，总是以16位为单位存取。

10.3.14　定时器通用寄存器 （TGR）

TGR 为 16 位可读取 / 写入的寄存器，共有 21 个，通道 0 有 6 个，通道 1、 2 各有 2 个，通道 3、 4 各有 4 个，通道 5 有 3 个。

TGRA、TGRB、TGRC、TGRD 为输出比较 / 输入捕捉复用的寄存器。通道 0、3、4 的 TGRC 和 TGRD 可设定作为缓冲寄存器。 TGR 和缓冲寄存器的组合为 TGRA － TGRC、 TGRB － TGRD。

TGRE_0、 TGRF_0 用作比较寄存器。 TCNT_0 和 TGRE_0 匹配时，可产生 A/D 转换开始请求。可设定 TGRF 作为缓冲寄存器。 TGR 和缓冲寄存器的组合为 TGRE － TGRF。

TGRU_5、 TGRV_5、 TGRW_5 为比较匹配 / 输入捕捉 / 测量外部脉宽的复用寄存器。

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，总是以16位为单位存取。TGR的初始值为H'FFFF。

10.3.15　定时器开始寄存器 （TSTR）

TSTR 为 8 位可读取 / 写入的寄存器，选择通道 0 ～ 4 的 TCNT 运行 / 停止。

TSTR_5 为 8 位可读取 / 写入的寄存器，选择通道 5 的 TCNTU/V/W_5 运行 / 停止。

必须在停止 TCNT 计数器运行后，设定 TMDR 运行模式或对 TCR 设定 TCNT 的计数时钟。

- TSTR

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	位名称	初始值	R/W	说　明
7	CST4	0	R/W	计数器开始 4、 3 选择 TCNT 运行或停止。 在 TIOC 引脚为输出状态时，对 CST 位写入 0，计数器停止运行，但保持 TIOC 引脚的输出比较输出电平。如果在 CST 位为 0 的状态写入 TIOR，引脚的输出电平则更改为已设定的初始输出值。 0：TCNT_4、 TCNT_3 的计数停止 1：TCNT_4、 TCNT_3 计数
6	CST3	0	R/W	
5 ～ 3	—	均为 0	R	保留位 读取值、写入值总是为 0。
2	CST2	0	R/W	计数器开始 2 ～ 0 选择 TCNT 运行或停止。 在 TIOC 引脚为输出状态时，对 CST 位写入 0，计数器停止运行，但保持 TIOC 引脚的输出比较输出电平。如果在 CST 位为 0 的状态写入 TIOR，引脚的输出电平更改为已设定的初始输出值。 0：TCNT_2 ～ TCNT_0 的计数停止 1：TCNT_2 ～ TCNT_0 计数
1	CST1	0	R/W	
0	CST0	0	R/W	

- TSTR_5

位：	7	6	5	4	3	2	1	0
	—	—	—	—	—	CSTU5	CSTV5	CSTW5
初始值：	0	0	0	0	0	0	0	0
R/W：	R	R	R	R	R	R/W	R/W	R/W

位	位名称	初始值	R/W	说　明
7 ～ 3	—	均为 0	R	保留位 读取值、写入值总是为 0。
2	CSTU5	0	R/W	计数器开始 U5 选择 TCNTU_5 运行或停止。 0：TCNTU_5 的计数停止 1：TCNTU_5 计数
1	CSTV5	0	R/W	计数器开始 V5 选择 TCNTV_5 运行或停止。 0：TCNTV_5 的计数停止 1：TCNTV_5 计数
0	CSTW5	0	R/W	计数器开始 W5 选择 TCNTW_5 运行或停止。 0：TCNTW_5 的计数停止 1：TCNTW_5 计数

10.3.16　定时器同步寄存器 （TSYR）

TSYR 为 8 位可读取 / 写入的寄存器，选择通道 0 ～ 4 的 TCNT 独立运行或同步运行。对相应的位置 1 后的通道执行同步运行。

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	位名称	初始值	R/W	说　明
7	SYNC4	0	R/W	定时器同步 4、 3 选择独立运行或与其他通道同步运行。 选择同步运行时，可同步预置多个 TCNT，并且可通过清除其他通道的计数器执行同步清除。 设定同步运行时，至少需要将 2 个通道的 SYNC 位置 1。设定同步清除时，除 SYNC 位之外，还需要由 TCR 的 CCLR2 ～ CCLR0 位设定 TCNT 的清除源。 0：TCNT_4、 TCNT_3 独立运行 （TCNT 的预置 / 清除与其他通道无关） 1：TCNT_4、 TCNT_3 同步运行 可同步预置 / 同步清除 TCNT
6	SYNC3	0	R/W	
5 ～ 3	—	均为 0	R	保留位 读取值、写入值总是为 0。
2	SYNC2	0	R/W	定时器同步 2 ～ 0 选择独立运行或与其他通道同步运行。 选择同步运行时，可同步预置多个 TCNT，并可通过清除其他通道的计数器执行同步清除。 设定同步运行时，至少需要将 2 个通道的 SYNC 位置 1，除 SYNC 位之外，还需要由 TCR 的 CCLR2 ～ CCLR0 位设定 TCNT 的清除源。 0：TCNT_2 ～ TCNT_0 独立运行 （TCNT 的预置 / 清除和其他通道无关） 1：TCNT_2 ～ TCNT_0 同步运行 可同步预置 / 同步清除 TCNT
1	SYNC1	0	R/W	
0	SYNC0	0	R/W	

10.3.17　定时器计数器同步开始寄存器 （TCSYSTR）

TCSYSTR 为 8 位可读取 / 写入的寄存器，可同步开始 MTU2 和 MTU2S 的计数器。但 MTU2S 无 TCSYSTR。

位：	7	6	5	4	3	2	1	0
	SCH0	SCH1	SCH2	SCH3	SCH4	—	SCH3S	SCH4S
初始值：	0	0	0	0	0	0	0	0
R/W：	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R/(W)*	R/(W)*

【注】* 为了设置寄存器，仅可写入1。

位	位名称	初始值	R/W	说　明
7	SCH0	0	R/(W)*	同步开始 控制 MTU2 的 TCNT_0 的同步开始。 0：不同步开始 MTU2 的 TCNT_0 1：同步开始 MTU2 的 TCNT_0 ［清除条件］ • 在 SCH0=1 的状态下， MTU2 的 TSTR 的 CST0 位置 1 时
6	SCH1	0	R/(W)*	同步开始 控制 MTU2 的 TCNT_1 的同步开始。 0：不同步开始 MTU2 的 TCNT_1 1：同步开始 MTU2 的 TCNT_1 ［清除条件］ • 在 SCH1=1 的状态下，MTU2 的 TSTR 的 CST1 位置 1 时
5	SCH2	0	R/(W)*	同步开始 控制 MTU2 的 TCNT_2 的同步开始。 0：不同步开始 MTU2 的 TCNT_2 1：同步开始 MTU2 的 TCNT_2 ［清除条件］ • 在 SCH2=1 的状态下， MTU2 的 TSTR 的 CST2 位置 1 时
4	SCH3	0	R/(W)*	同步开始 控制 MTU2 的 TCNT_3 的同步开始。 0：不同步开始 MTU2 的 TCNT_3 1：同步开始 MTU2 的 TCNT_3 ［清除条件］ • 在 SCH3=1 的状态下，MTU2 的 TSTR 的 CST3 位置 1 时
3	SCH4	0	R/(W)*	同步开始 控制 MTU2 的 TCNT_4 的同步开始。 0：不同步开始 MTU2 的 TCNT_4 1：同步开始 MTU2 的 TCNT_4 ［清除条件］ • 在 SCH4=1 的状态下，MTU2 的 TSTR 的 CST4 位置 1 时
2	—	0	R	保留位 读取值、写入值总是为 0。

位	位名称	初始值	R/W	说　明
1	SCH3S	0	R/(W)*	同步开始 控制 MTU2S 的 TCNT_3S 的同步开始。 0：不同步开始 MTU2S 的 TCNT_3S 1：同步开始 MTU2S 的 TCNT_3S ［清除条件］ • 在 SCH3S=1 的状态下，MTU2S 的 TSTRS 的 CST3 位置 1 时
0	SCH4S	0	R/(W)*	同步开始 控制 MTU2S 的 TCNT_4S 的同步开始。 0：不同步开始 MTU2S 的 TCNT_4S 1：同步开始 MTU2S 的 TCNT_4S ［清除条件］ • 在 SCH4S=1 的状态下，MTU2S 的 TSTRS 的 CST4 位置 1 时

【注】 * 为了设置寄存器，仅可写入 1。

10.3.18 定时器读取 / 写入允许寄存器 （TRWER）

TRWER 为 8 位可读取 / 写入的寄存器，设定允许或禁止存取通道 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	位名称	初始值	R/W	说　明
7 ～ 1	—	均为 0	R	保留位 读取值、写入值总是为 0。
0	RWE	1	R/W	读取 / 写入允许 设定允许 / 禁止对误写防止寄存器的读取 / 写入。 0：禁止寄存器的读取 / 写入 1：允许寄存器的读取 / 写入 ［清除条件］ • 在 RWE=1 的状态下，读取 RWE 后，对 RWE 写入 0 时

• 误写防止的对象寄存器/计数器

TCR_3、 4、 TMDR_3、 4、 TIORH_3、 4、 TIORL_3、 4、 TIER_3、 4、 TGRA_3、 4、 TGRB_3、 4、 TOER、 TOCR1、 TOCR2、 TGCR、 TCDR、 TDDR 和 TCNT_3、 4，共计 22 个寄存器。

10.3.19　定时器输出主控允许寄存器 （TOER）

TOER 为 8 位可读取 / 写入的寄存器，允许 / 禁止执行输出引脚 TIOC4D、 TIOC4C、 TIOC3D、 TIOC4B、 TIOC4A、 TIOC3B 的输出设定。如果未设定 TOER 的各位，则这些引脚无法正确输出。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	位名称	初始值	R/W	说　明
7、6	—	均为 1	R	保留位 读取值、写入值总是为 1。
5	OE4D	0	R/W	主控允许 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。详情参阅 “10.3.20 定时器输出控制寄存器 1 （TOCR1）”、“10.3.21 定时器输出控制寄存器 2 （TOCR2）”。如果在除互补 PWM 模式 / 复位同步 PWM 模式之外执行 MTU2 输出，本位必须设定为 1，设定为 0 时，输出低电平。

10.3.20　定时器输出控制寄存器 1　（TOCR1）

TOCR1 为 8 位可读取 / 写入的寄存器，允许或禁止与互补 PWM 模式 / 复位同步 PWM 模式的 PWM 周期同步的交替输出，并控制 PWM 的输出电平取反。

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】*　上电复位后，仅可写入1次。写入1后，不可写入0。

位	位名称	初始值	R/W	说　明
7	—	0	R	保留位 读取值、写入值总是为 0。
6	PSYE	0	R/W	PWM 同步输出允许 设定允许 / 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 读取值、写入值总是为 0。
3	TOCL	0	R/(W)*1	TOC 寄存器写入禁止位 *1 设定禁止 / 允许写入 TOCR1 寄存器的 TOCS 位、 OLSN 位、 OLSP 位。 0：允许写入 TOCS 位、 OLSN 位、 OLSP 位 1：禁止写入 TOCS 位、 OLSN 位、 OLSP 位
2	TOCS	0	R/W	TOC 选择位 选择通过设定互补 PWM 模式 / 复位同步 PWM 模式的输出电平是使 TOCR1 的设定有效，还是使 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	输出电平选择 N*3 复位同步 PWM 模式 / 互补 PWM 模式时，选择反相输出电平。请参照表 10.33。
0	OLSP	0	R/W	输出电平选择 P*3 复位同步 PWM 模式 / 互补 PWM 模式时，选择正相输出电平。请参照表 10.34。

【注】 *1　上电复位后，仅可写入一次 1。写入 1 后，不可写入 0。

*2　TOCL 位置 1，可防止 CPU 失控时的误写入。

*3　通过将 TOCS 位置 0，使本设定有效。

表 10.33　输出电平选择功能

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，在计数开始并经过空载时间后，变为有效电平。

表 10.34　输出电平选择功能

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 模式的输出例子 （1 个相位）如图 10.3 所示。

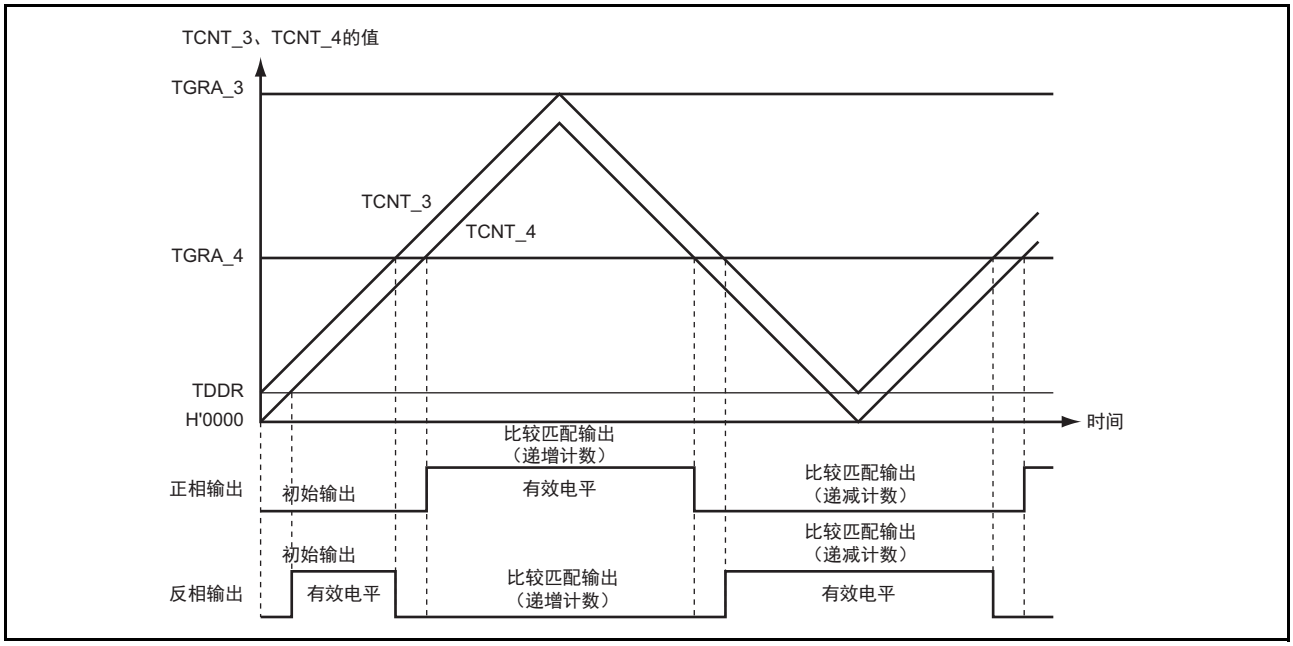


图 10.3　互补 PWM 模式的输出电平例子

10.3.21　定时器输出控制寄存器 2 （TOCR2）

TOCR2 为 8 位可读取 / 写入的寄存器，控制在互补 PWM 模式 / 复位同步 PWM 模式的 PWM 输出的输出电平取反。

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	位名称	初始值	R/W	说　明
7、6	BF[1:0]	00	R/W	选择 TOLBR 缓冲传送时序 选择从 TOLBR 向 TOCR2 的缓冲传送时序。详情参照表 10.35。
5	OLS3N	0	R/W	选择输出电平 3N* 复位同步 PWM 模式 / 互补 PWM 模式时，选择 TIOC4D 的输出电平。请参照表 10.36。
4	OLS3P	0	R/W	选择输出电平 3P* 复位同步 PWM 模式 / 互补 PWM 模式时，选择 TIOC4B 的输出电平。请参照表 10.37。
3	OLS2N	0	R/W	选择输出电平 2N* 复位同步 PWM 模式 / 互补 PWM 模式时，选择 TIOC4C 的输出电平。请参照表 10.38。
2	OLS2P	0	R/W	选择输出电平 2P* 复位同步 PWM 模式 / 互补 PWM 模式时，选择 TIOC4A 的输出电平。请参照表 10.39。
1	OLS1N	0	R/W	选择输出电平 1N* 复位同步 PWM 模式 / 互补 PWM 模式时，选择 TIOC3D 的输出电平。请参照表 10.40。
0	OLS1P	0	R/W	选择输出电平 1P* 复位同步 PWM 模式 / 互补 PWM 模式时，选择 TIOC3B 的输出电平。请参照表 10.41。

【注】 *　通过将 TOCR1 的 TOCS 位置 1，使本设定有效。

表 10.35　BF1、 BF0 位的设定

bit7	bit6	说　明	
BF1	BF0	互补 PWM 模式	复位 PWM 模式
0	0	不从缓冲寄存器 （TOLBR）向 TOCR2 传送	不从缓冲寄存器 （TOLBR）向 TOCR2 传送
0	1	在 TCNT_4 的波峰，从缓冲寄存器 （TOLBR）向 TOCR2 传送	清除 TCNT_3/4 计数器时，从缓冲寄存器 （TOLBR）向 TOCR2 传送
1	0	在 TCNT_4 的波谷，从缓冲寄存器 （TOLBR）向 TOCR2 传送	禁止设定
1	1	在 TCNT_4 的波峰和波谷，从缓冲寄存器 （TOLBR）向 TOCR2 传送	禁止设定

表 10.36　TIOC4D 输出电平选择功能

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，在计数开始并经过空载时间后，变为有效电平。

表 10.37　TIOC4B 输出电平选择功能

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 10.38　TIOC4C 输出电平选择功能

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，在计数开始并经过空载时间后，变为有效电平。

表 10.39　TIOC4A 输出电平选择功能

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 10.40　TIOC3D 输出电平选择功能

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】 反相波形的初始初始值，在计数开始并经过空载时间后，变为有效电平。

表 10.41　TIOC3B 输出电平选择功能

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

10.3.22　定时器输出电平缓冲寄存器 （TOLBR）

TOLBR 为 TOCR2 的缓冲寄存器，设定互补 PWM 模式 / 复位同步 PWM 模式的 PWM 输出电平。TOLBR 寄存器为 8 位可读取 / 写入的寄存器。

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	位名称	初始值	R/W	说　明
7、6	—	均为 0	R	保留位 读取值、写入值总是为 0。
5	OLS3N	0	R/W	必须对 TOCR2 的 OLS3N 位设定缓冲传送的值。
4	OLS3P	0	R/W	必须对 TOCR2 的 OLS3P 位设定缓冲传送的值。
3	OLS2N	0	R/W	必须对 TOCR2 的 OLS2N 位设定缓冲传送的值。
2	OLS2P	0	R/W	必须对 TOCR2 的 OLS2P 位设定缓冲传送的值。
1	OLS1N	0	R/W	必须对 TOCR2 的 OLS1N 位设定缓冲传送的值。
0	OLS1P	0	R/W	必须对 TOCR2 的 OLS1P 位设定缓冲传送的值。

缓冲运行过程中设定 PWM 输出电平的步骤如图 10.4 所示。

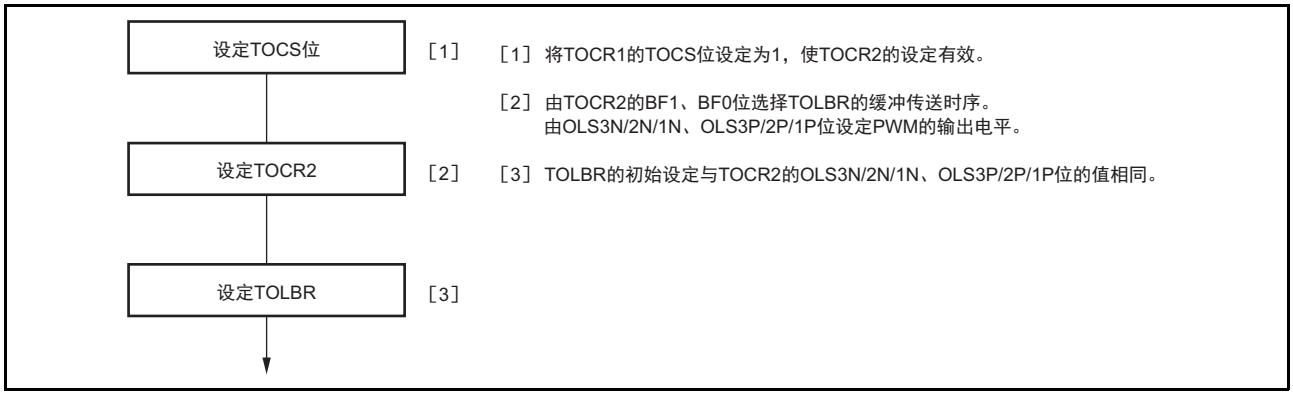


图 10.4　缓冲运行过程中设定 PWM 输出电平的步骤

10.3.23　定时器门控寄存器 （TGCR）

TGCR 在复位同步 PWM 模式 / 互补 PWM 模式，可对无刷 DC 马达控制执行必要的波形输出控制。 TGCR 寄存器为 8 位可读取 / 写入的寄存器。在除互补 PWM 模式 / 复位同步 PWM 模式之外，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	位名称	初始值	R/W	说　明
7	—	1	R	保留位 读取值、写入值总是为 1。
6	BDC	0	R/W	无刷 DC 马达 选择设置本寄存器的功能有效或无效。 0：普通输出 1：本寄存器功能有效
5	N	0	R/W	反相输出 （N）控制 输出反相引脚 （TIOC3D 引脚、 TIOC4C 引脚、 TIOC4D 引脚）时，选择电平输出或复位同步 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 输出正相引脚 （TIOC3B 引脚、 TIOC4A 引脚、 TIOC4B 引脚）时，选择电平输出或复位同步 PWM/ 互补 PWM 输出。 0：电平输出 1：复位同步 PWM/ 互补 PWM 输出
3	FB*	0	R/W	外部反馈信号允许 选择由 MTU2/ 通道 0 的 TGRA、 TGRB、 TGRC 的输入捕捉信号自动转换正相 / 反相输出，或通过对 TGCR 的 bit2 ～ 0 写入 0 或 1 转换正相 / 反相输出。 0：由外部输入 （输入源为通道 0 的 TGRA、 TGRB、 TGRC 的输入捕捉信号）执行输出转换 1：由软件执行输出转换 （TGCR 的 UF、 VF、 WF 的设定值）
2	WF	0	R/W	输出相转换 2 ～ 0 设定正 / 反输出相的 ON、 OFF。这些位的设定仅在寄存器的 FB 位为 1 时有效。此时， bit2 ～ 0 的设定取代外部输入。请参照表 10.42
1	VF	0	R/W	
0	UF	0	R/W	

【注】 *　在 MTU2S 设定 BDC 位为 1 时，不得将 FB 位设定为 0。

表 10.42　输出电平选择功能

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

10.3.24　定时器副计数器（TCNTS）

TCNTS 为 16 位只读计数器，仅可在互补 PWM 模式使用，初始值为 H'0000。

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

【注】　禁止以8位为单位存取TCNTS，总是以16位为单位存取。

10.3.25　定时器空载时间数据寄存器（TDDR）

TDDR 为 16 位寄存器，仅可在互补 PWM 模式使用，互补 PWM 模式时设定 TCNT_3 和 TCNT_4 计数器的偏移值。在互补 PWM 模式，清除 TCNT_3、 TCNT_4 计数器后重新开始计数时，将 TDDR 寄存器的值加载到 TCNT_3 计数器后开始计数。 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，总是以16位为单位存取。

10.3.26　定时器周期数据寄存器 （TCDR）

TCDR 为 16 位寄存器，仅可在互补 PWM 模式使用， TCDR 寄存器的值设定为 PWM 载波周期的 1/2。在互补 PWM 模式， TCDR 寄存器可随时和 TCNTS 计数器比较，如果匹配， TCNTS 计数器则转换计数方向（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，总是以16位为单位存取。

10.3.27　定时器周期缓冲寄存器 （TCBR）

TCBR 为 16 位寄存器，仅可在互补 PWM 模式使用，并可作为 TCDR 寄存器的缓冲寄存器。按照在 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，总是以16位为单位存取。

10.3.28　定时器中断跳过设定寄存器 （TITCR）

TITCR 为 8 位可读取 / 写入的寄存器，禁止 / 允许中断跳过，控制中断跳过次数的设定。 MTU2 有 1 个 TITCR。

位：	7	6	5	4	3	2	1	0
	T3AEN	3ACOR[2:0]			T4VEN	4VCOR[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	位名称	初始值	R/W	说　明
7	T3AEN	0	R/W	T3AEN 设定禁止 / 允许跳过 TGIA_3 中断的跳过。 0：禁止 TGIA_3 中断跳过 1：允许 TGIA_3 中断跳过
6 ～ 4	3ACOR[2:0]	000	R/W	设定 TGIA_3 中断跳过次数在 0 ～ 7 次内。 * 详情参照表 10.43。
3	T4VEN	0	R/W	T4VEN 设定禁止 / 允许 TCIV_4 中断的跳过。 0：禁止 TCIV_4 中断跳过 1：允许 TCIV_4 中断跳过
2 ～ 0	4VCOR[2:0]	000	R/W	设定 TCIV_4 中断跳过次数在 0 ～ 7 次内。 * 详情参照表 10.44。

【注】 *　中断跳过次数设定为 0 时，不执行跳过。更改中断跳过次数前，必须将 T3AEN、 T4VEN 位设定为 0，再清除跳过次数计数器 （TITCNT）。

表 10.43　由 3ACOR2 ～ 3ACOR0 位设定中断跳过次数

bit6	bit5	bit4	说　　明
3ACOR2	3ACOR1	3ACOR0	
0	0	0	不执行 TGIA_3 的中断跳过
0	0	1	设定 TGIA_3 的中断跳过次数为 1 次
0	1	0	设定 TGIA_3 的中断跳过次数为 2 次
0	1	1	设定 TGIA_3 的中断跳过次数为 3 次
1	0	0	设定 TGIA_3 的中断跳过次数为 4 次
1	0	1	设定 TGIA_3 的中断跳过次数为 5 次
1	1	0	设定 TGIA_3 的中断跳过次数为 6 次
1	1	1	设定 TGIA_3 的中断跳过次数为 7 次

表 10.44　由 4VCOR2 ～ 4VCOR0 位设定中断跳过次数

bit2	bit1	bit0	说　　明
4VCOR2	4VCOR1	4VCOR0	
0	0	0	不执行 TCIV_4 的中断跳过
0	0	1	设定 TCIV_4 的中断跳过次数为 1 次
0	1	0	设定 TCIV_4 的中断跳过次数为 2 次
0	1	1	设定 TCIV_4 的中断跳过次数为 3 次
1	0	0	设定 TCIV_4 的中断跳过次数为 4 次
1	0	1	设定 TCIV_4 的中断跳过次数为 5 次
1	1	0	设定 TCIV_4 的中断跳过次数为 6 次
1	1	1	设定 TCIV_4 的中断跳过次数为 7 次

10.3.29　定时器中断跳过次数计数器 （TITCNT）

TITCNT 为 8 位可读取计数器， MTU2 有 1 个 TITCNT。 TCNT_3 及 TCNT_4 的计数停止后， TITCNT 仍保持其值。

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	位名称	初始值	R/W	说　明
7	—	0	R	保留位 读取值总是为 0。
6 ～ 4	3ACNT[2:0]	000	R	TGIA_3 中断计数器 TITCR 的 T3AEN 位置 1，产生 TGIA_3 中断源时，递增 1。 ［清除条件］ • TITCR 的 3ACOR2 ～ 3ACOR0 和 TITCNT 的 3ACNT2 ～ 3ACNT0 匹配 • TITCR 的 T3AEN 位为 0 • TITCR 的 3ACOR2 ～ 3ACOR0 为 0
3	—	0	R	保留位 读取值总是为 0。
2 ～ 0	4VCNT[2:0]	000	R	TCIV_4 中断计数器 TITCR 的 T4VEN 位置 1，产生 TCIV_4 中断源时，递增 1。 ［清除条件］ • TITCR 的 4VCOR2 ～ 4VCOR0 和 TITCNT 的 4VCNT2 ～ 4VCNT0 匹配 • TITCR 的 T4VEN 位为 0 • TITCR 的 4VCOR2 ～ 4VCOR0 为 0

【注】 清除 TITCNT 的值时，必须将 TITCR 的 T3AEN 位和 T4VEN 位清 0。

10.3.30　定时器缓冲传送设定寄存器 （TBTER）

TBTER 为 8 位可读取 / 写入的寄存器，可设定是否抑制从互补 PWM 模式使用的缓冲寄存器 * 向暂存器传送，或设定是否与中断跳过功能联动。 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	位名称	初始值	R/W	说　明
7 ～ 2	—	均为 0	R	保留位 读取值、写入值总是为 0。
1、 0	BTE[1:0]	00	R/W	设定是否抑制从互补 PWM 模式使用的缓冲寄存器 * 向暂存器传送，或设定是否与中断跳过功能联动。详情参照表 10.45。

【注】 *　对象缓冲寄存器
TGRC_3、 TGRD_3、 TGRC_4、 TGRD_4、 TCBR

表 10.45　BTE1、 BTE0 位的设定

bit1	bit0	说　　明
BTE1	BTE0	
0	0	不抑制从缓冲寄存器向暂存器传送 *1 不与中断跳过功能联动
0	1	抑制从缓冲寄存器向暂存器传送
1	0	从缓冲寄存器向暂存器传送与中断跳过功能联动 *2
1	1	禁止设定

【注】 *1　按照 TMDR 的 MD3 ～ MD0 的设定传送。详情参阅 “10.4.8　互补 PWM 模式”。

*2　禁止中断跳过时 （定时器中断跳过设定寄存器 （TITCR）的 T3AEN、 T4VEN 位置 0，或 TITCR 的中断跳过次数设定位 （3ACOR、 4VCOR）置 0），必须设定为缓冲传送不与中断跳过联动 （定时器缓冲传送寄存器 （TBTER）的 BTE1 设定为 0）。

禁止中断跳过时，如果设定为缓冲传送与中断跳过联动，则不执行缓冲传送。

10.3.31　定时器空载时间允许寄存器 （TDER）

TDER 为 8 位可读取 / 写入的寄存器。通道 3 有 1 个 TDER，可控制互补 PWM 模式空载时间的生成。MTU2 有 1 个 TDER，必须在 TCNT 运行停止状态设定 TDER。

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	位名称	初始值	R/W	说　　明
7 ～ 1	—	均为 0	R	保留位 读取值、写入值总是为 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］ • 在 TDER=1 的状态，读取 TDER 后，对 TDER 写入 0

【注】 *　必须设定为 TDDR ≥ 1。

10.3.32　定时器波形控制寄存器 （TWCR）

TWCR 为 8 位可读取 / 写入的寄存器。控制在互补 PWM 模式产生 TNCT_3、 TNCT_4 计数器同步清除时的输出波形，并设定是否由 TGRA_3 的比较匹配清除计数器。必须在 TCNT 运行停止状态设定 TWCR 的 CCE 位、 WRE 位。

位：	7	6	5	4	3	2	1	0
	CCE	—	—	—	—	—	SCC	WRE
初始值：	0*	0	0	0	0	0	0	0
R/W：	R/(W)	R	R	R	R	R	R/(W)	R/(W)

【注】* 除互补PWM模式1之外，不可设定为1。

位	位名称	初始值	R/W	说　明
7	CCE	0*	R/(W)	比较匹配清除允许 在互补 PWM 模式，设定是否由 TGRA_3 的比较匹配清除计数器 0：不由 TGRA_3 的比较匹配清除计数器 1：由 TGRA_3 的比较匹配清除计数器 ［置位条件］ • 在 CCE=0 的状态，读取 CCE 后，对 CCE 写入 1
6 ～ 2	—	均为 0	R	保留位 读取值、写入值总是为 0。
1	SCC	0	R/(W)	控制同步清除 在互补 PWM 模式产生 MTU2 － MTU2S 计数器同步清除时，设定是否清除 MTU2S 的 TCNT_3、 TCNT_4。 使用本功能时，必须设置 MTU2S 为互补 PWM 模式。 在计数器运行过程改写 SCC 位时，不得更改 CCE 位、 WRE 位的值。 只有在波谷的 Tb 区间以外产生同步清除时，才可设定 SCC 位使 MTU2 的同步清除无效。如果在包含 TCNT_3、 TCNT_4 开始后的波谷 Tb 区间产生同步清除，则清除 MTU2S 的 TCNT_3、 TCNT_4。 关于互补 PWM 模式波谷的 Tb 区间，参照图 10.41。 在 MTU2 为保留位。读取值、写入值总是为 0。 0：通过 MTU2 － MTU2S 同步清除功能，使清除 MTU2S 的 TCNT_3、TCNT_4 有效 1：通过 MTU2 － MTU2S 同步清除功能，使清除 MTU2S 的 TCNT_3、TCNT_4 无效 ［置位条件］ • 在 SCC=0 的状态，读取 SCC 后，对 SCC 写入 1
0	WRE	0	R/(W)	波形保持允许 选择在互补 PWM 模式产生计数器同步清除时的输出波形。 只有在互补 PWM 模式的波谷的 Tb 区间产生同步清除时，可由此功能保持波形。在此之外产生同步清除时，与 WRE 位的设定无关，输出在 TOCR 寄存器设定的初始值。在 TCNT_3、 TCNT_4 开始后的波谷的 Tb 区间产生同步清除时，也输出由 TOCR 寄存器设定的初始值。 关于互补 PWM 模式波谷的 Tb 区间，参照图 10.41。 0：输出由 TOCR 寄存器设定的初始值 1：保持同步清除前的波形 ［置位条件］ • 在 WRE=0 的状态，读取 WRE 后，对 WRE 写入 1

【注】 * 除互补 PWM 模式 1 之外，不得设定为 1。

10.3.33　与总线主控器的接口

定时器计数器 （TCNT）、通用寄存器 （TGR）、定时器副计数器 （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）及定时器 A/D 转换开始请求周期设定缓冲寄存器 （TADCOBR）均为 16 位寄存器。与总线主控器之间的数据总线宽度为 16 位，因此能以 16 位为单位读取 / 写入，但不能以 8 位为单位读取 / 写入，总是以 16 位为单位存取。

除上述之外，其余均为 8 位寄存器。与 CPU 之间的数据总线宽度为 16 位，因此能以 16 位为单位读取 / 写入。同时，也能以 8 位为单位读取 / 写入。

10.4　运行说明

10.4.1　基本运行

在各通道有 TCNT 和 TGR。 TCNT 可递增计数、自由运行计数器、周期计数器或外部事件计器可运行。

TGR 可分别用作输入捕捉寄存器或输出比较寄存器。

必须由引脚功能控制器 （PFC）设定 MTU2 的外部引脚功能。

(1)　计数器的运行

TSTR 的 CST0 ～ CST4 位及 TSTR_5 的 CSTU5、 CSTV5、 CSTW5 位置 1 时，相应通道的 TCNT 开始计数，自由运行计数器、周期计数器等可运行。

(a)　计数器运行的设定步骤例子

计数器运行的设定步骤例子如图 10.5 所示。

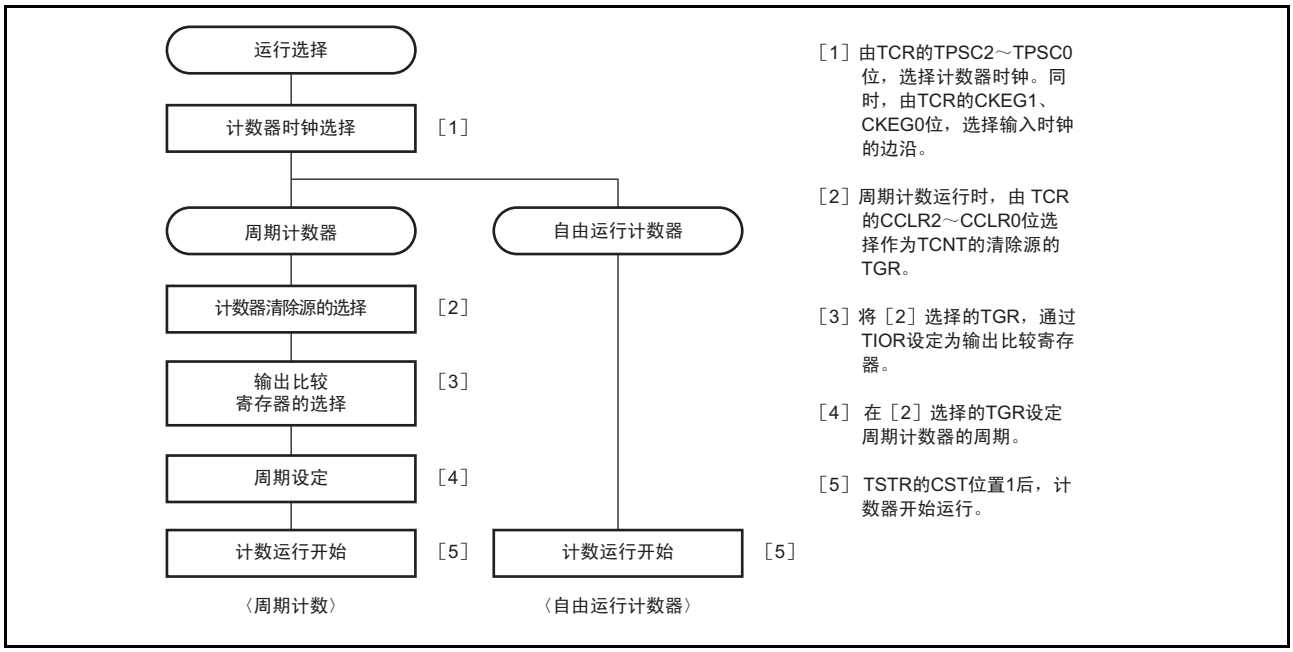


图 10.5　计数器运行设定步骤例子

(b)　自由运行计数器和周期计数器的运行

MTU2 的 TCNT 在刚复位后均为自由运行计数器的设定，如果 TSTR 的对应位置 1，则以自由运行计数器开始递增计数。 TCNT 上溢 （H'FFFF→H'0000）时，TSR 的 TCFV 位置 1。此时，如果对应的 TIER 的 TCIEV 位为 1，MTU2 则请求中断。 TCNT 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 10.6 所示。

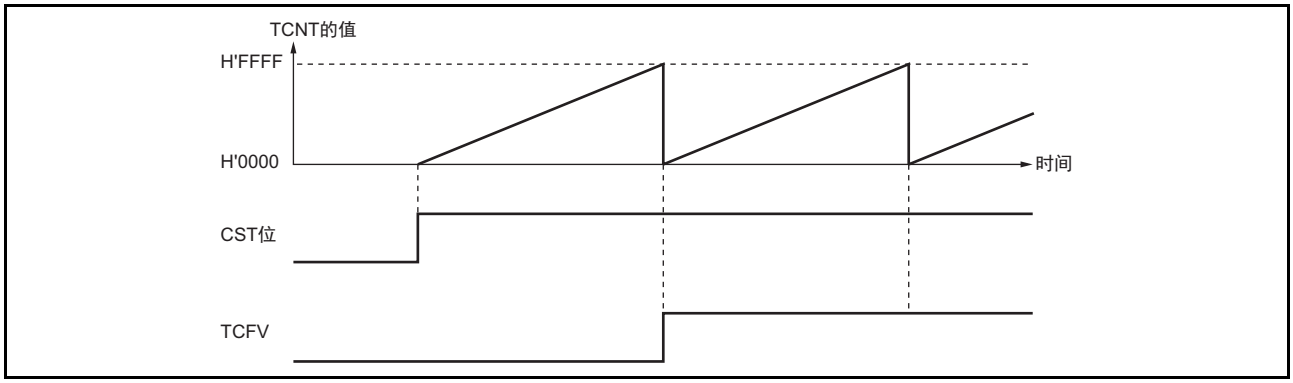


图 10.6　自由运行计数器的运行

选择比较匹配为 TCNT 的清除源时，对应通道的 TCNT 执行周期计数。将用于周期设定的 TGR 设定为输出比较寄存器，由 TCR 的 CCLR2 ～ CCLR0 位选择比较匹配产生的计数器清除。设定后，如果 TSTR 的对应位置 1，则作为周期计数器开始递增计数。计数值和 TGR 的值匹配时，TSR 的 TGF 位置 1， TCNT 清除为 H'0000。

此时如果对应 TIER 的 TGIE 位为 1， MTU2 请求中断。 TCNT 在比较匹配后，从 H'0000 开始继续递增计数。

周期计数器的运行如图 10.7 所示。

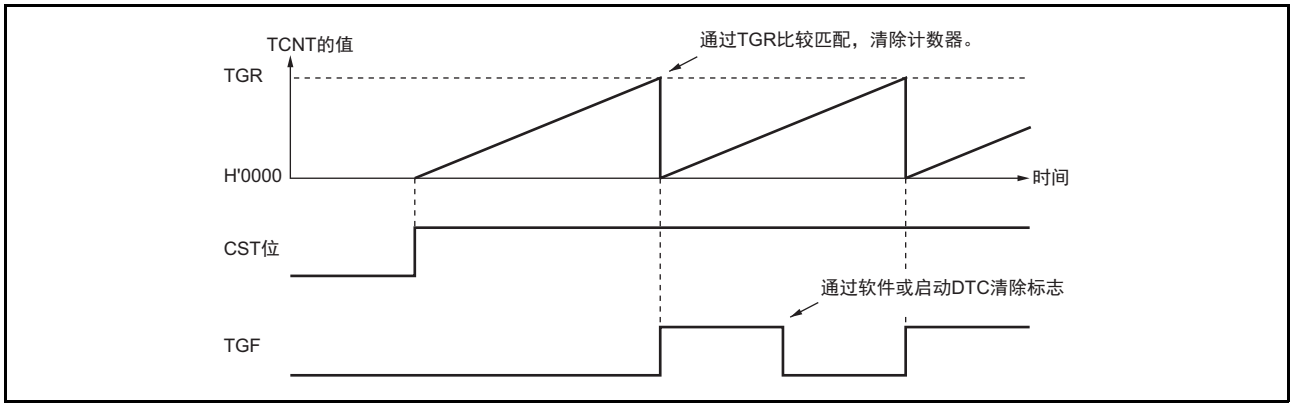


图 10.7　周期计数器的运行

(2)　通过比较匹配执行波形输出功能

MTU2 可通过比较匹配从对应的输出引脚输出 0/ 输出 1/ 交替输出。

(a)　通过比较匹配执行波形输出的设定步骤例子

通过比较匹配执行波形输出的设定步骤例子如图 10.8 所示。

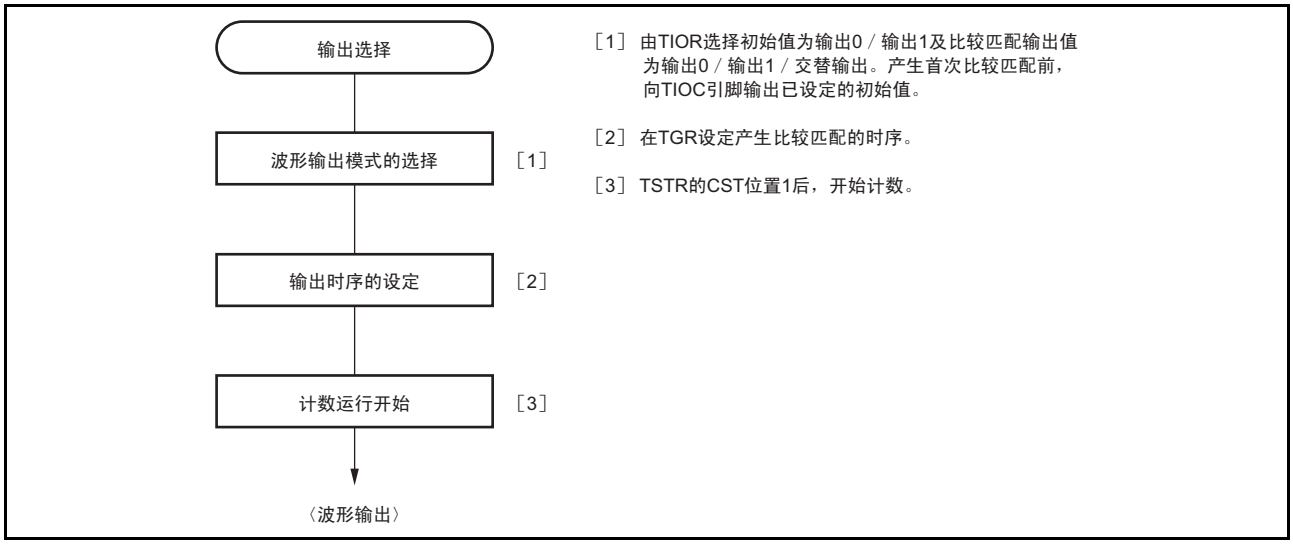


图 10.8　通过比较匹配执行波形输出的设定步骤例子

(b)　波形输出运行例子

输出 0/ 输出 1 的例子如图 10.9 所示。

在此，设定 TCNT 为自由运行计数、通过比较匹配 A 输出 1、比较匹配 B 输出 0。设定的电平与引脚电平一致时，引脚电平不变。

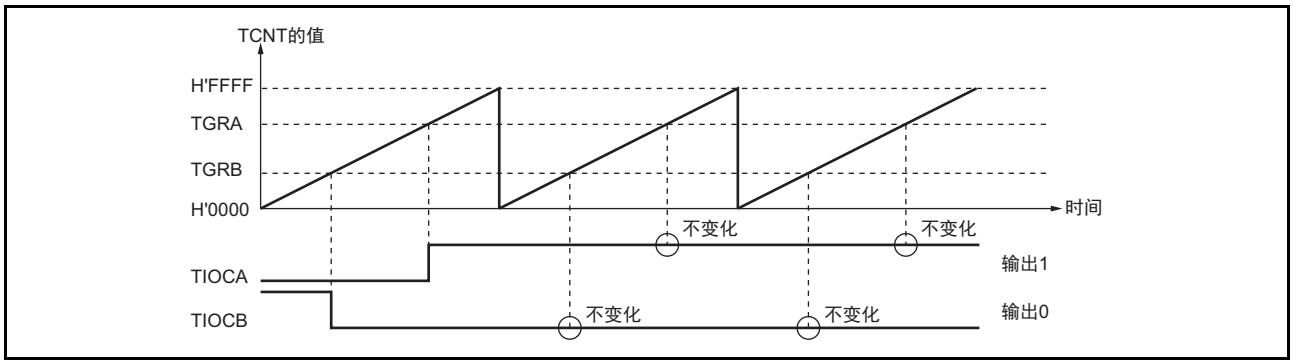


图 10.9　输出 0/ 输出 1 的运行例子

交替输出例子如图 10.10 所示。

在此，设定 TCNT 为周期计数 （通过比较匹配 B 清除计数器），比较匹配 A、 B 均设定为交替输出。

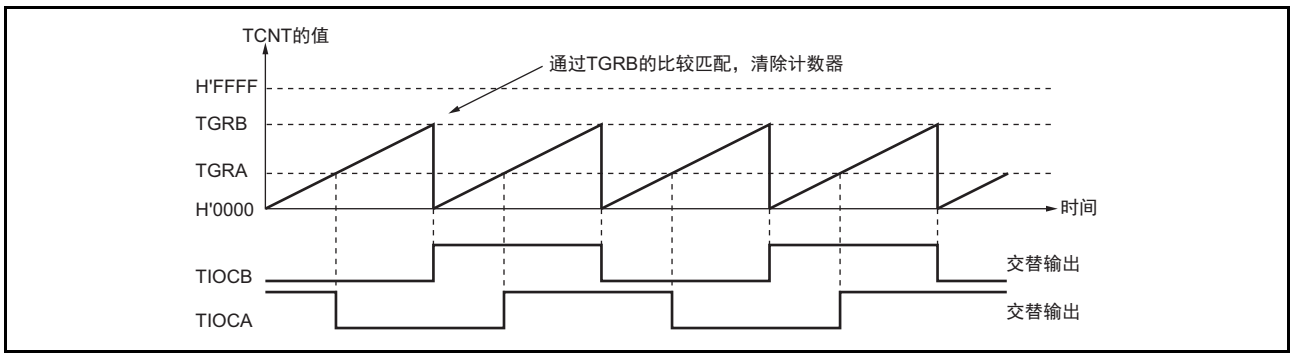


图 10.10　交替输出运行例子

(3)　输入捕捉功能

本功能可检测 TIOC 引脚的输入边沿，并传送 TCNT 的值到 TGR。

检测边沿可选择上升沿 / 下降沿 / 双边沿。通道 0、 1 可将其他通道的计数器输入时钟和比较匹配信号作为输入捕捉源。

【注】 在通道 0、 1 用其他通道的计数器输入时钟作为输入捕捉时，用作输入捕捉的计数器输入时钟不可选择 MPϕ/1。如果

选择 MPϕ/1，则不产生输入捕捉。

(a)　输入捕捉的设定步骤例子

输入捕捉的设定步骤例子如图 10.11 所示。

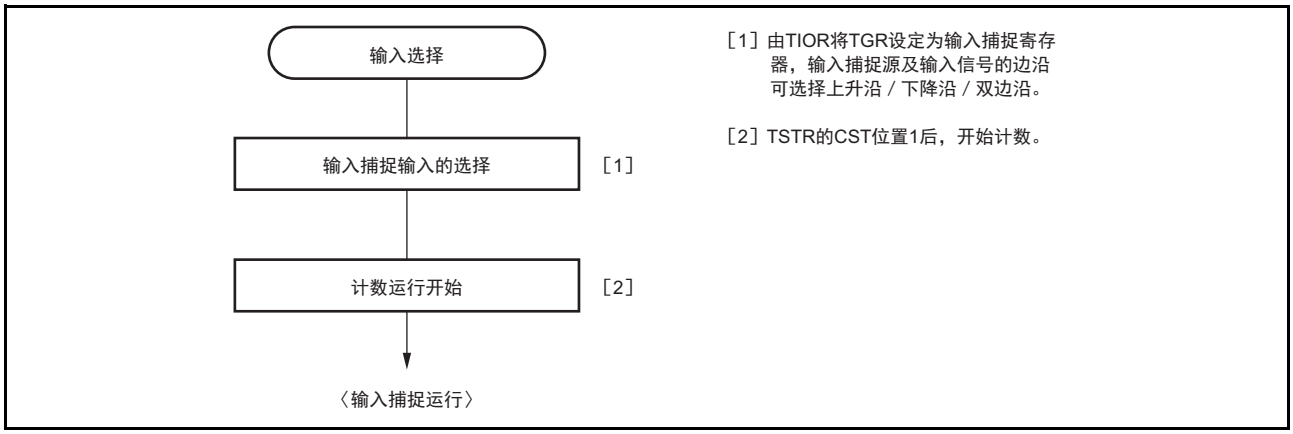


图 10.11　输入捕捉的设定例子

(b)　输入捕捉运行例子

输入捕捉运行例子如图 10.12 所示。

本例选择上升 / 下降双边沿为 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，并将 TCNT 设定为由 TGRB 的输入捕捉清除计数器。

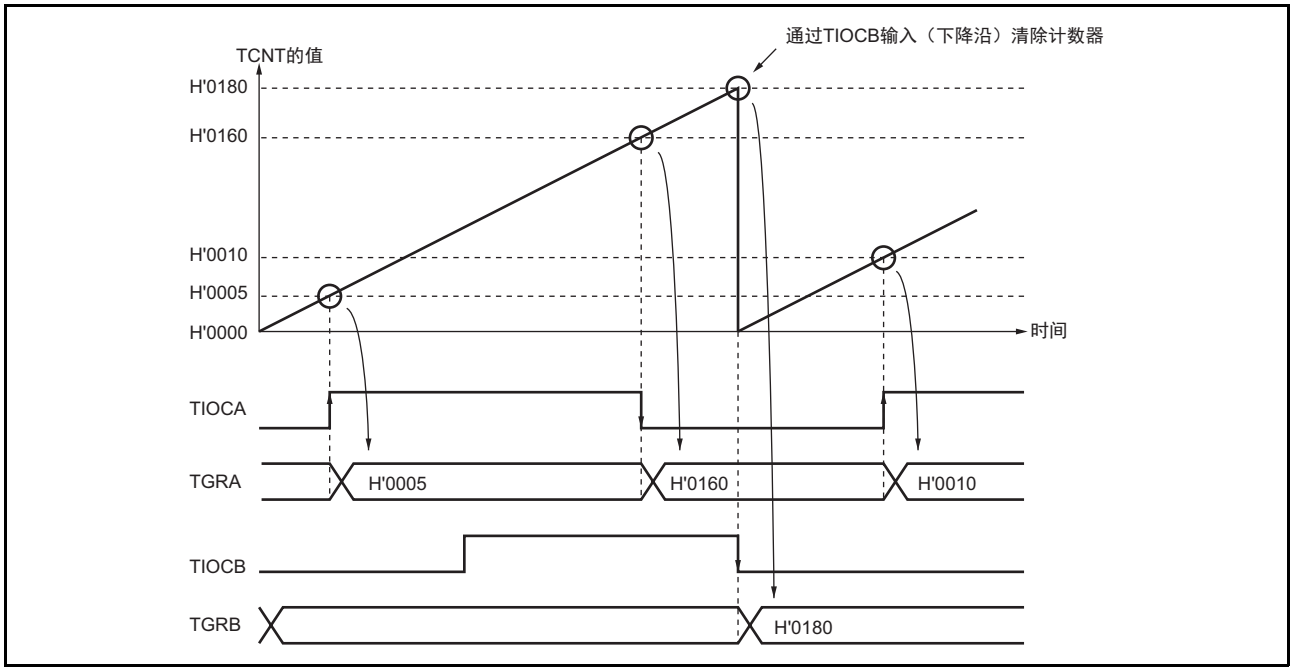


图 10.12　输入捕捉运行例子

10.4.2　同步运行

同步运行可同时改写多个 TCNT 的值（同步预置），还可通过设定 TCR 同时清除多个 TCNT（同步清除）。

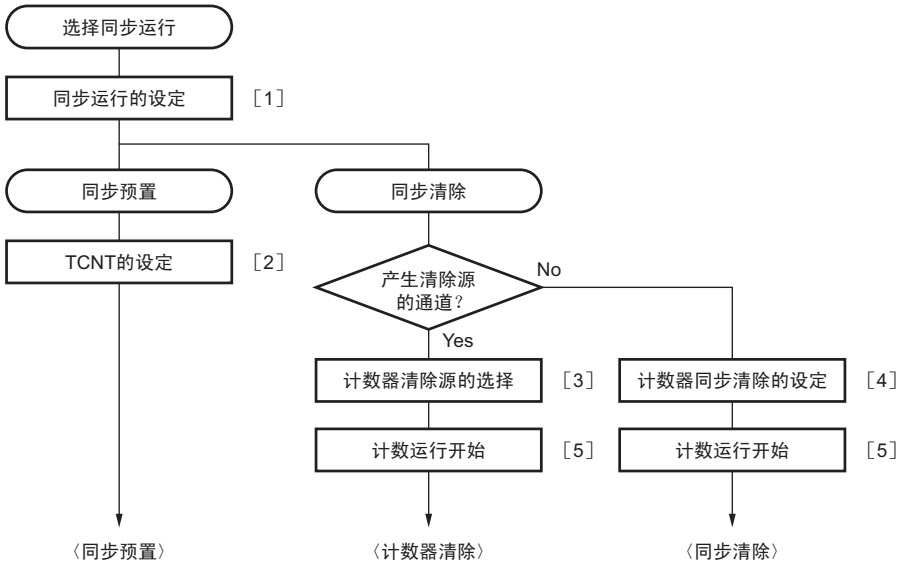
同步运行可增加 1 个时基运行的 TGR 个数。

通道 0 ～ 4 均可设定为同步运行。

通道 5 不可同步运行。

(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 10.13 所示。



[1] 与设定为同步运行的通道相对应的TSYR的SYNC位置1。

[2] 如果对设定为同步运行的通道的某个TCNT进行写入操作，则其他TCNT同时也被写入相同值。

[3] 通过TCR的CCLR2～CCLR0位，设定为由输入捕捉 / 输出比较清除TCNT。

[4] 通过TCR的CCLR2～CCLR0位，设定同步清除计数器清除源。

[5] 与对象通道相对应的TSTR的CST位置1后，开始计数。

图 10.13　同步运行的设定步骤例子

(2)　同步运行例子

同步运行例子如图 10.14 所示。

本例设定通道 0 ～ 2 为同步运行且为 PWM 模式 1，设定 TGRB_0 的比较匹配为通道 0 的计数器清除源，同步清除为通道 1、 2 计数器清除源。

从 TIOC0A、 TIOC1A 和 TIOC2A 引脚输出 3 相 PWM 波形。此时，通道 0 ～ 2 的 TCNT 同步预置，通过 TGRB_0 的比较匹配进行同步清除，在 TGRB_0 设定的数据为 PWM 周期。

有关 PWM 模式，详情参阅 “10.4.5 PWM 模式”。

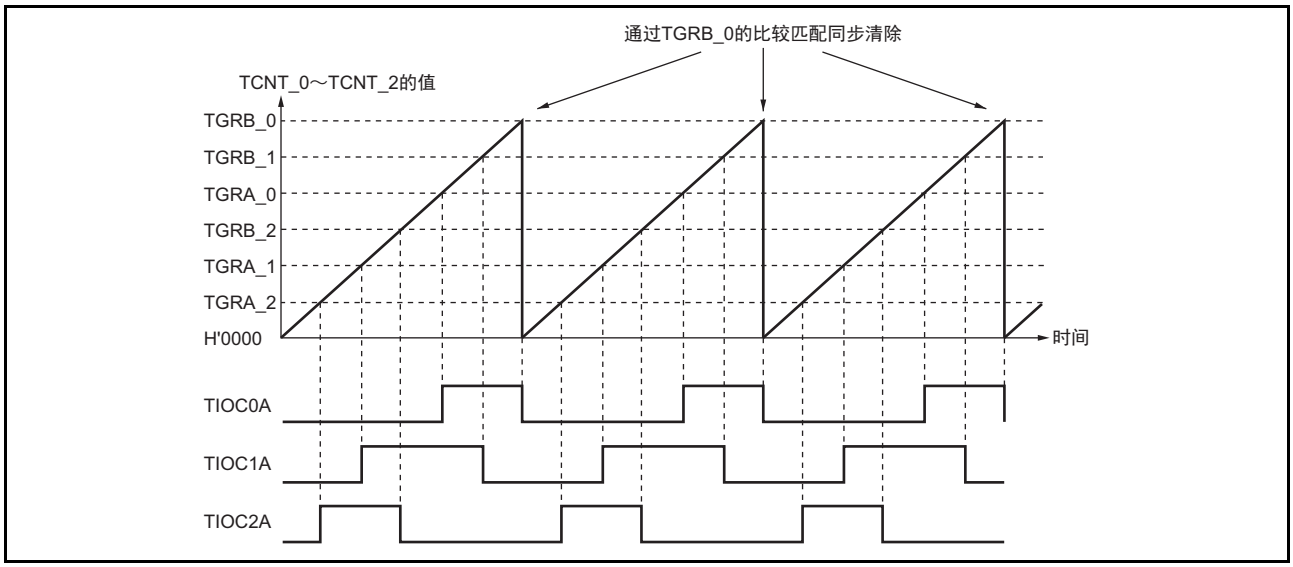


图 10.14　同步运行的例子

10.4.3　缓冲运行

通道 0、3、4 具有缓冲运行功能。TGRC 和 TGRD 可用作缓冲寄存器，通道 0 的 TGRF 也可用作缓冲寄存器。

TGR 设定为输入捕捉寄存器和设定为比较匹配寄存器时的缓冲运行各不相同。

【注】 TGRE_0 不可设定为输入捕捉寄存器，仅可作为比较匹配寄存器运行。

缓冲运行时的寄存器组合如表 10.46 所示。

表 10.46　寄存器组合

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

产生比较匹配时，相应通道的缓冲寄存器值传送到定时器通用寄存器。如图 10.15 所示。

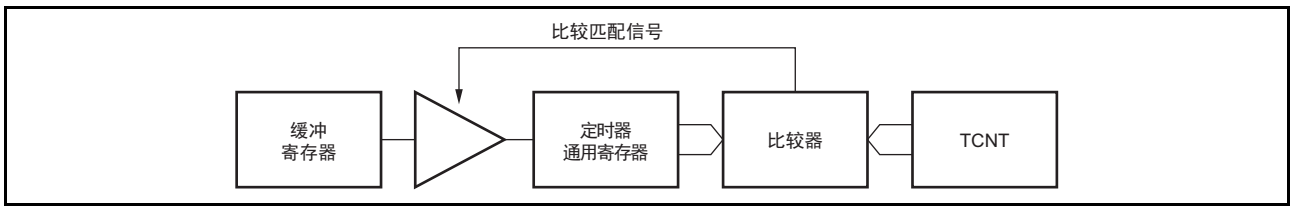


图 10.15　比较匹配缓冲运行

- TGR 为输入捕捉寄存器时

产生输入捕捉时， TCNT 的值传送至 TGR 的同时，将之前保存在 TGR 的值传送至缓冲寄存器。此运行如图 10.16 所示。

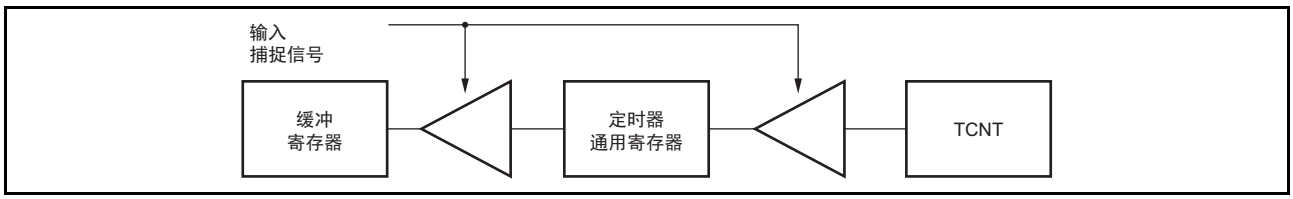


图 10.16　输入捕捉缓冲运行

(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 10.17 所示。

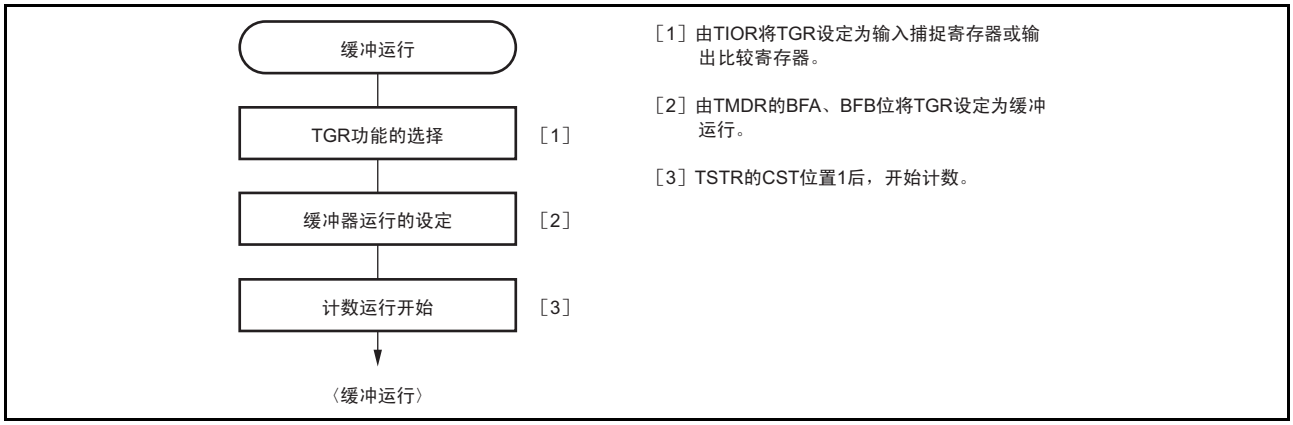


图 10.17　缓冲运行的设定步骤例子

(2)　缓冲运行例子

(a)　TGR 为输出比较寄存器时

通道 0 设定为 PWM 模式 1， TGRA 和 TGRC 设定为缓冲运行的例子如图 10.18 所示。设定由比较匹配 B 清除 TCNT，由比较匹配 A 输出 1、比较匹配 B 输出 0。此例中， TBTM 的 TTSA 位设定为 0。

因为设定为缓冲运行，产生比较匹配 A 后，输出发生变化的同时，缓冲寄存器 TGRC 的值传送到定时器通用寄存器 TGRA。每产生比较匹配 A，就重复此运行。

有关 PWM 模式，详情参阅 “10.4.5 PWM 模式”。

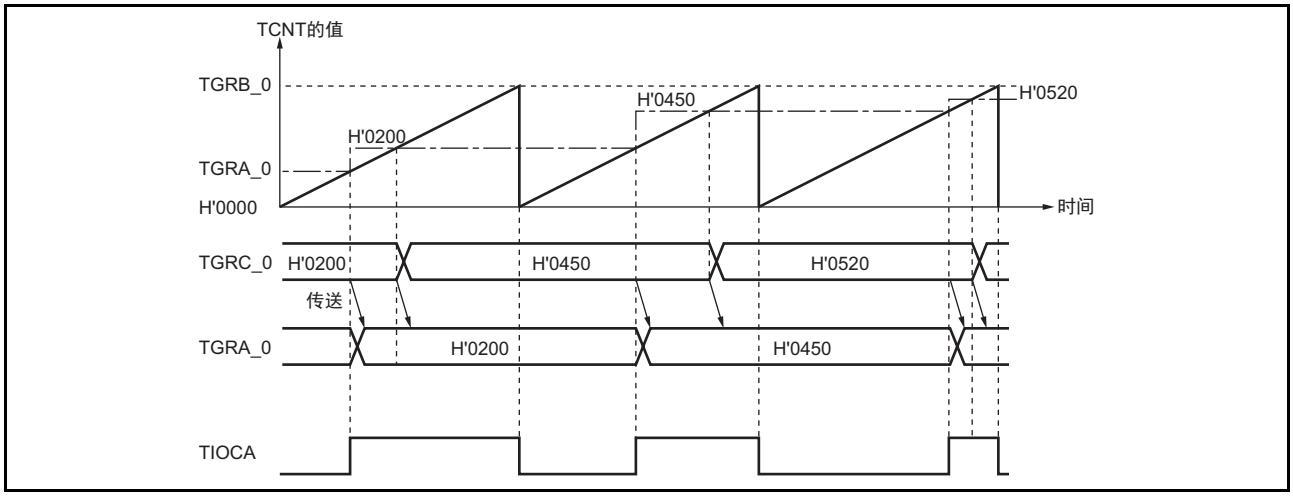


图 10.18　缓冲运行例子 （1）

(b)　TGR 为输入捕捉寄存器时

设定 TGRA 为输入捕捉寄存器、 TGRA 和 TGRC 为缓冲运行时的例子如图 10.19 所示。

TCNT 由 TGRA 的输入捕捉清除计数器， TIOCA 引脚的输入捕捉输入边沿可选择上升 / 下降的双边沿。

因为设定为缓冲运行，通过输入捕捉 A 将 TCNT 的值保存到 TGRA 的同时，将之前保存在 TGRA 的值传送到 TGRC。

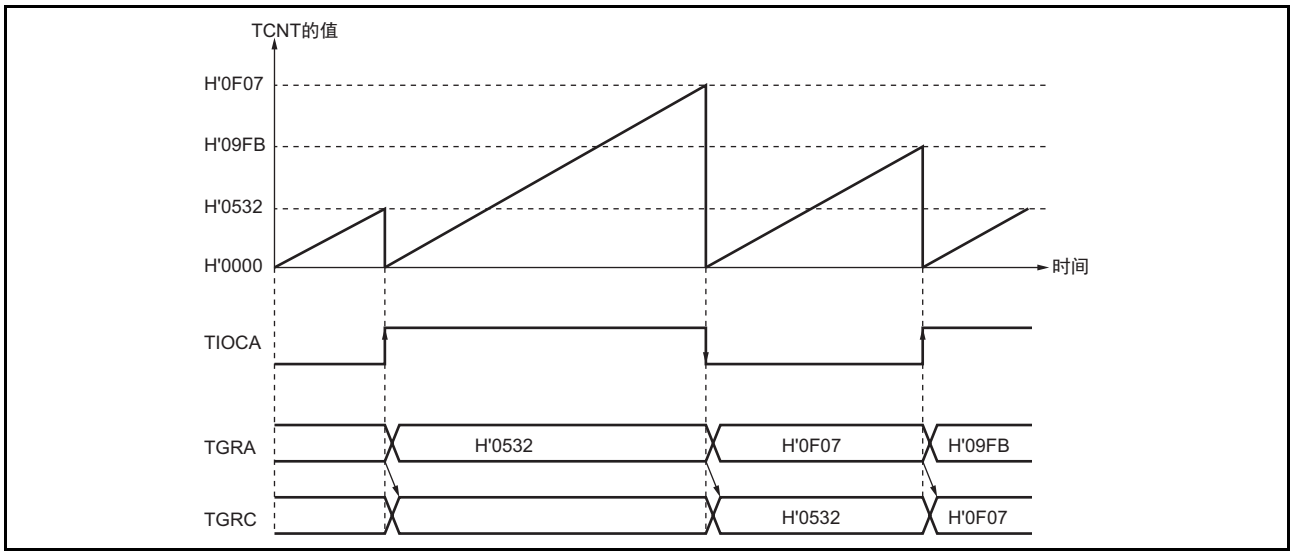


图 10.19　缓冲运行例子 （2）

(3)　选择从缓冲运行时的缓冲寄存器向定时器通用寄存器传送的时序

通过设定缓冲运行传送模式寄存器（TBTM_0、TBTM_3、TBTM_4），可选择在通道 0 为 PWM 模式 1、2 以及通道 3、4 为 PWM 模式 1 时，从缓冲寄存器向定时器通用寄存器传送的时序。可选择的缓冲传送时序为产生比较匹配时（初始值）或清除 TCNT 时的时序。以下任一条件成立时，可清除 TCNT。

- TCNT 上溢时（H'FFFF→H'0000）
- 计数器运行中，对 TCNT 写入 H'0000 时
- 通过 TCR 的 CCLR2～CCLR0 位设定的清除源，将 TCNT 清除为 H'0000 时

【注】 必须在 TCNT 停止的状态设定 TBTM。

设定通道 0 为 PWM 模式 1， TGRA_0 和 TGRC_0 为缓冲运行的例子如图 10.20 所示。设定由比较匹配 B 清除 TCNT_0，产生比较匹配 A 输出 1、比较匹配 B 输出 0，并设定 TBTM_0 的 TTSA 位为 1。

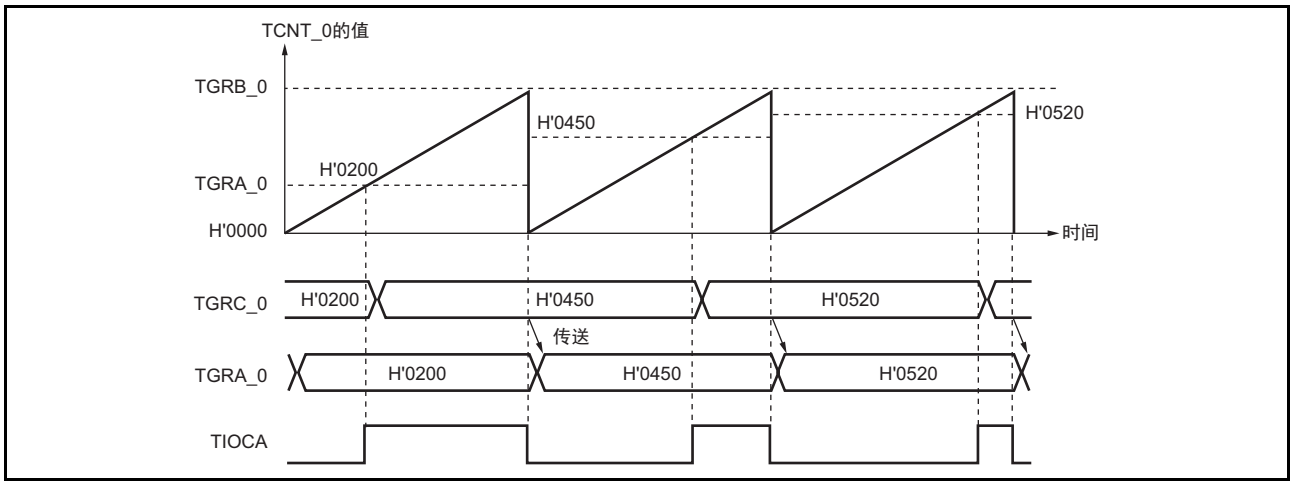


图 10.20　选择清除 TCNT_0 时从 TGRC_0 向 TGRA_0 的缓冲传送时序的例子

10.4.4　级联运行

级联运行是连接 2 个通道的 16 位计数器作为 32 位计数器运行的功能。

通过在 TCR 的 TPSC2 ～ TPSC0 位设定通道 1 的计数器时钟为通过 TCNT_2 的上溢 / 下溢计数，运行此功能。

仅在低 16 位的 TCNT 为相位计数模式时产生下溢。

级联组合如表 10.47 所示。

【注】 设定通道 1 为相位计数模式时，计数器时钟的设定无效，在相位计数模式独立运行。

表 10.47　级联组合

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，同时输入捕捉 TCNT_1 和 TCNT_2，通过设定输入捕捉控制寄存器 （TICCR），可追加输入引脚为输入捕捉条件。有关级联时的输入捕捉，详情参阅 “10.7.22 级联中 TCNT_1 和 TCNT_2 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 10.48 所示。

表 10.48　TICCR 设定值和输入捕捉输入引脚的对应表

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 10.21 所示。

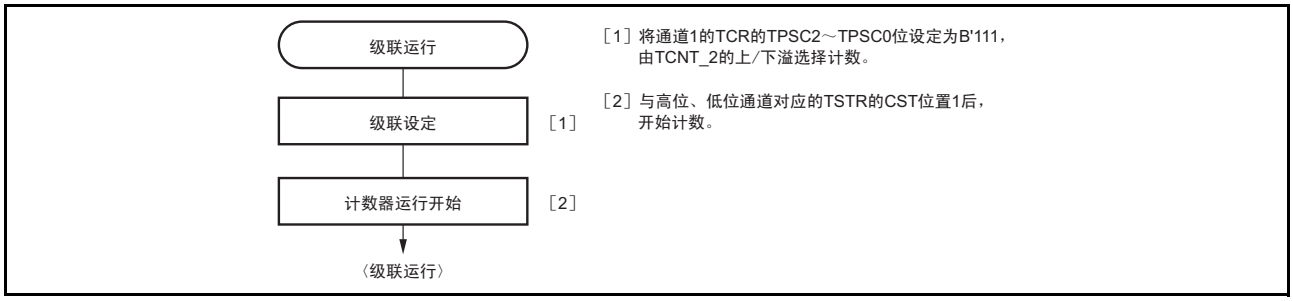


图 10.21　级联运行的设定步骤例子

(2)　级联运行例子 （a）

TCNT_1 通过 TCNT_2 的上溢 / 下溢执行计数，通道 2 设定为相位计数模式时的运行如图 10.22 所示。通过 TCNT_2 的上溢，TCNT_1 递增计数；通过 TCNT_2 的下溢，TCNT_1 递减计数。

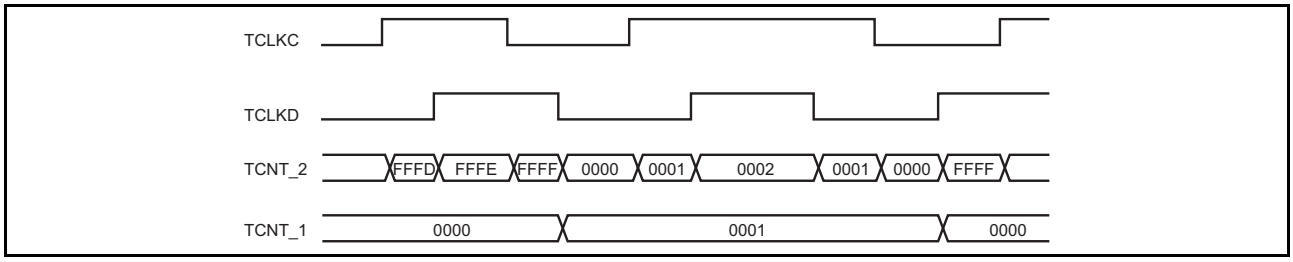


图 10.22　级联运行例子 （a）

(3)　级联运行例子 （b）

级联 TCNT_1 和 TCNT_2， TICCR 的 I2AE 位置 1 后，将 TIOC2A 引脚追加到 TGRA_1 输入捕捉条件时的运行如图 10.23 所示。在此，设定 TIOR_1 的 IOA0 ～ IOA3 在 （TIOC1A 的）上升沿执行输入捕捉， TIOR2 的 IOA0 ～ IOA3 在 （TIOC2A 的）上升沿执行输入捕捉。

此时，设定 TIOC1A 和 TIOC2A 的上升沿均为 TGRA_1 输入捕捉条件， TIOC2A 的上升沿为 TGRA_2 的输入捕捉条件。

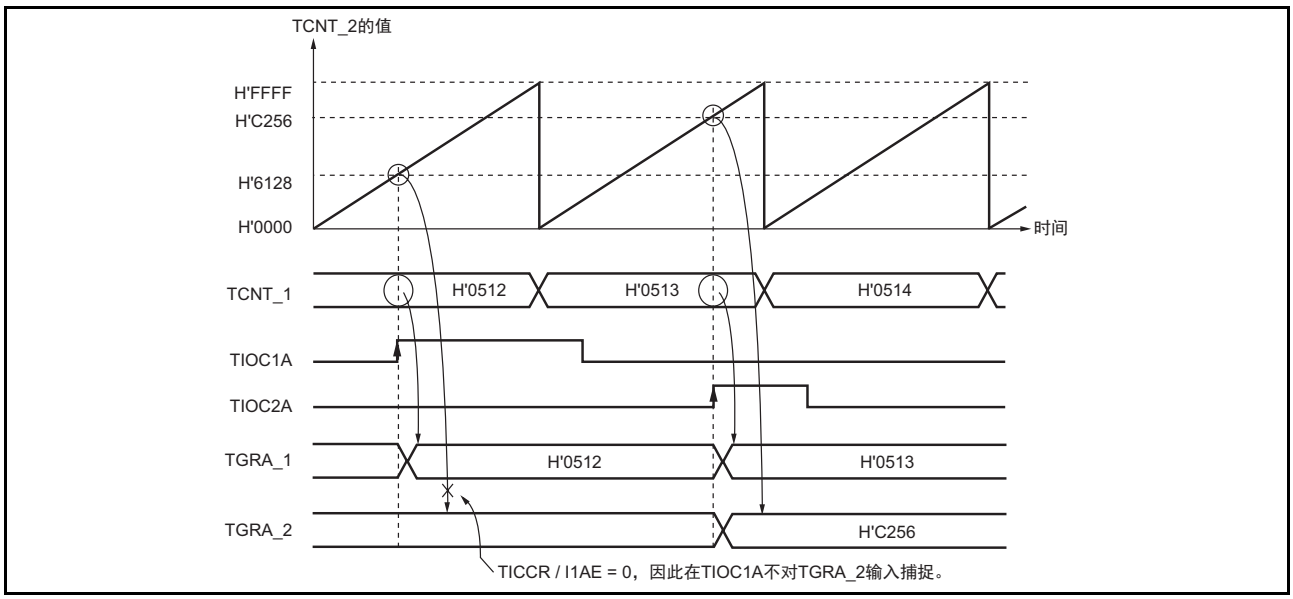


图 10.23　级联运行例子 （b）

(4)　级联运行例子 （c）

级联 TCNT_1 和 TCNT_2，TICCR 的 I2AE 位和 I1AE 位置 1 后，将 TIOC2A、 TIOC1A 引脚分别追加到 TGRA_1 和 TGRA_2 输入捕捉条件时的运行如图 10.24 所示。在此设定 TIOR_1、 TIOR_2 的 IOA0 ～ IOA3 均可在双边沿执行输入捕捉。此时， TIOC1A 和 TIOC2A 输入的 OR 为 TGRA1 和 TGRA2 的输入捕捉条件。

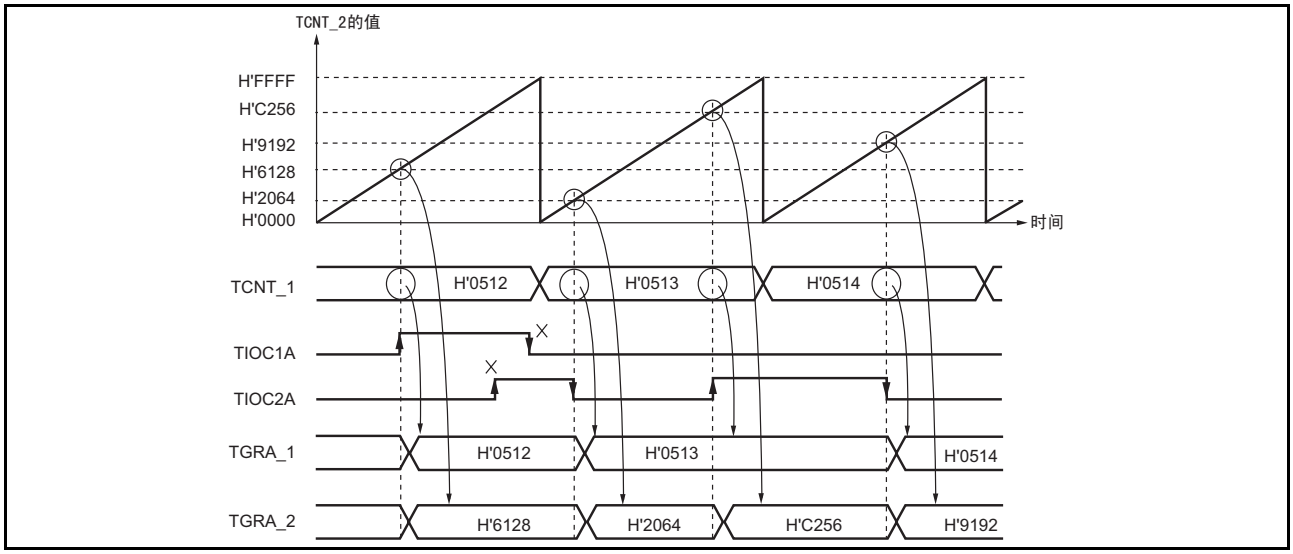


图 10.24　级联运行例子 （c）

(5)　级联运行例子 （d）

级联 TCNT_1 和 TCNT_2， TICCR 的 I2AE 位置 1 后，将 TIOC2A 引脚追加到 TGRA_1 的输入捕捉条件时的运行如图 10.25 所示。在此，通过产生 TGRA_0 的比较匹配 / 输入捕捉设定 TIOR_1 的 IOA0 ～ IOA3 为输入捕捉， TIOR_2 的 IOA0 ～ IOA3 在 （TIOC2A 的）上升沿执行输入捕捉。

此时，TIOR_1 设定为通过产生 TGRA_0 的比较匹配 / 输入捕捉执行输入捕捉，因此即使设定 TICCR 的 I2AE 位为 1，TIOC2A 的边沿也不可作为 TGRA_1 的输入捕捉条件。

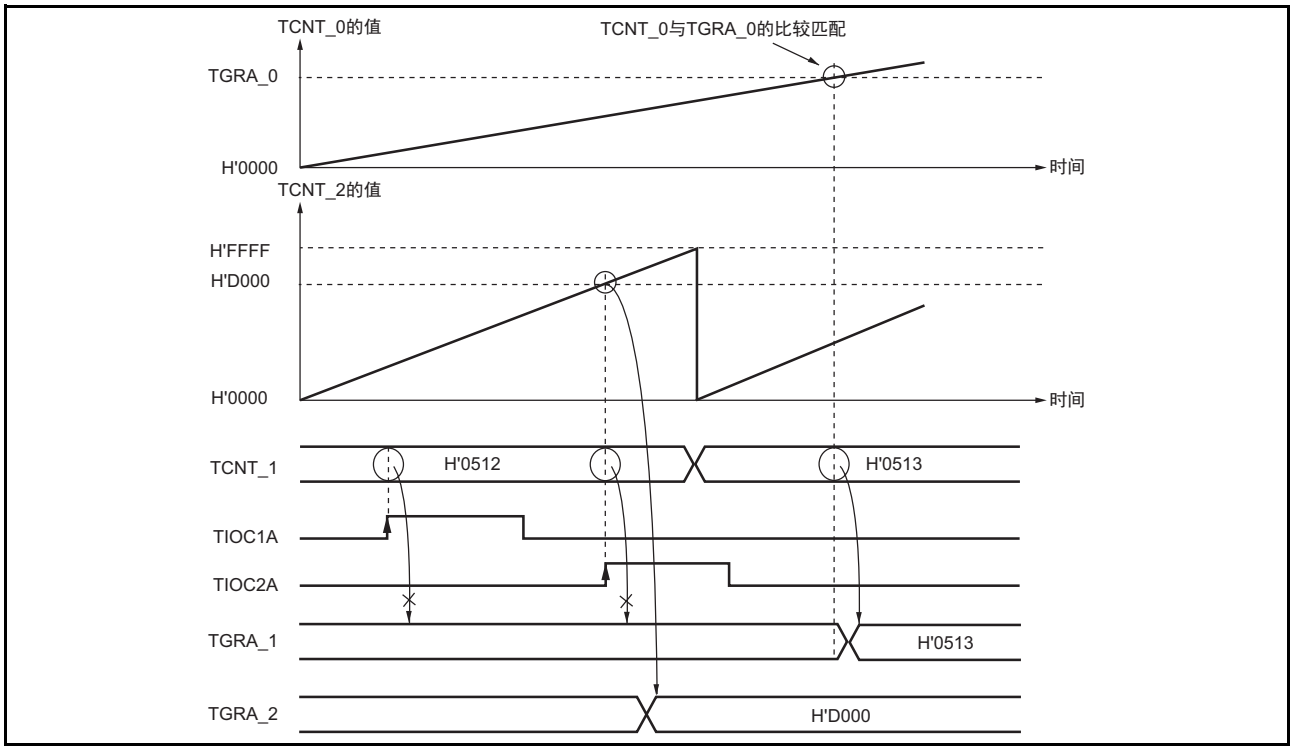


图 10.25　级联运行例子 （d）

10.4.5　PWM 模式

PWM 模式为可从输出引脚输出各 PWM 波形的模式。各 TGR 的比较匹配输出电平可选择输出 0/ 输出 1/ 交替输出。

根据各 TGR 的设定，可输出占空比为 0 ～ 100% 的 PWM 波形。

由于设置 TGR 的比较匹配为计数器清除源，因此在该寄存器可设定周期。所有通道均可独立设定为 PWM 模式，也可同步运行。

PWM 模式有以下两种：

(a)　PWM 模式 1

配对使用 TGRA 和 TGRB、 TGRC 和 TGRD，从 TIOCA 和 TIOCC 引脚产生 PWM 输出。由比较匹配 A、C，从 TIOCA、 TIOCC 引脚可执行 TIOR 的 IOA3 ～ IOA0、 IOC3 ～ IOC0 位指定的输出，由比较匹配 B、 D 执行 TIOR 的 IOB3 ～ IOB0、 IOD3 ～ IOD0 位指定的输出。初始输出值为在 TGRA、 TGRC 设定的值。配对使用的 TGR 设定值相同时，即使产生比较匹配，输出值也不变。

在 PWM 模式 1，最多可输出 8 相 PWM。

(b)　PWM 模式 2

将 1 个 TGR 用作周期寄存器，其他 TGR 用作占空比寄存器，产生 PWM 输出。由比较匹配，执行 TIOR 指定的输出。并且，通过由同步寄存器的比较匹配清除计数器，各引脚输出 TIOR 设定的初始值。周期寄存器和占空比寄存器的设定值相同时，即使产生比较匹配，输出值也不变。

在 PWM 模式 2，与同步运行并用，最多可输出 8 相 PWM。

PWM 输出引脚和寄存器的对应关系如表 10.49 所示。

表 10.49　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	不可设定
	TGRB_3		
	TGRC_3	TIOC3C	
	TGRD_3		
4	TGRA_4	TIOC4A	
	TGRB_4		
	TGRC_4	TIOC4C	
	TGRD_4		

【注】 PWM 模式 2 时，已设定周期的 TGR 不可执行 PWM 输出。

(1)　PWM 模式的设定步骤

PWM 模式的设定步骤如图 10.26 所示。

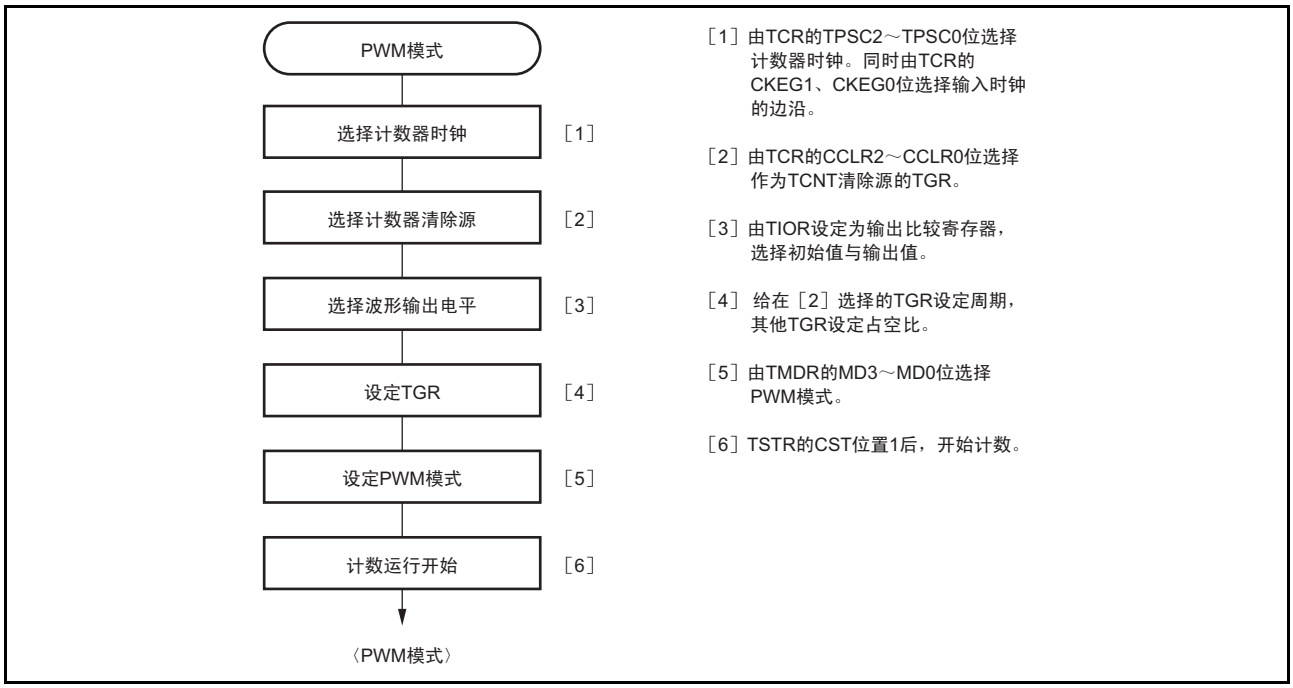


图 10.26　PWM 模式的设定步骤例子

(2)　PWM 模式的运行例子

PWM 模式 1 的运行例子如图 10.27 所示。

此图中，设定 TCNT 的清除源为 TGRA 的比较匹配，设定 TGRA 的初始输出值和输出值为 0， TGRB 的输出值为 1。

此时，设定在 TGRA 的值为周期，设定在 TGRB 的值为占空比。

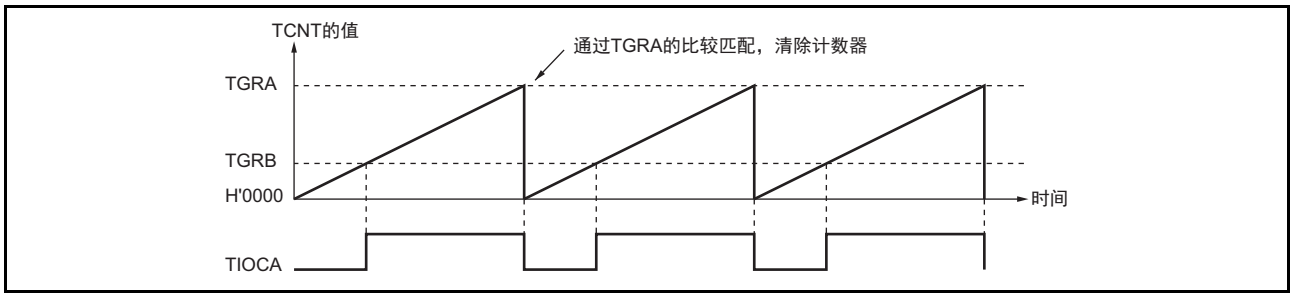


图 10.27　PWM 模式的运行例子

PWM 模式 2 运行例子如图 10.28 所示。

此图中，通道 0 和 1 同步运行，设定 TCNT 的清除源为 TGRB_1 的比较匹配，设定其他 TGR （TGRA_0 ～ TGRD_0、 TGRA_1）的初始输出值为 0、输出值为 1，可输出 5 相 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。

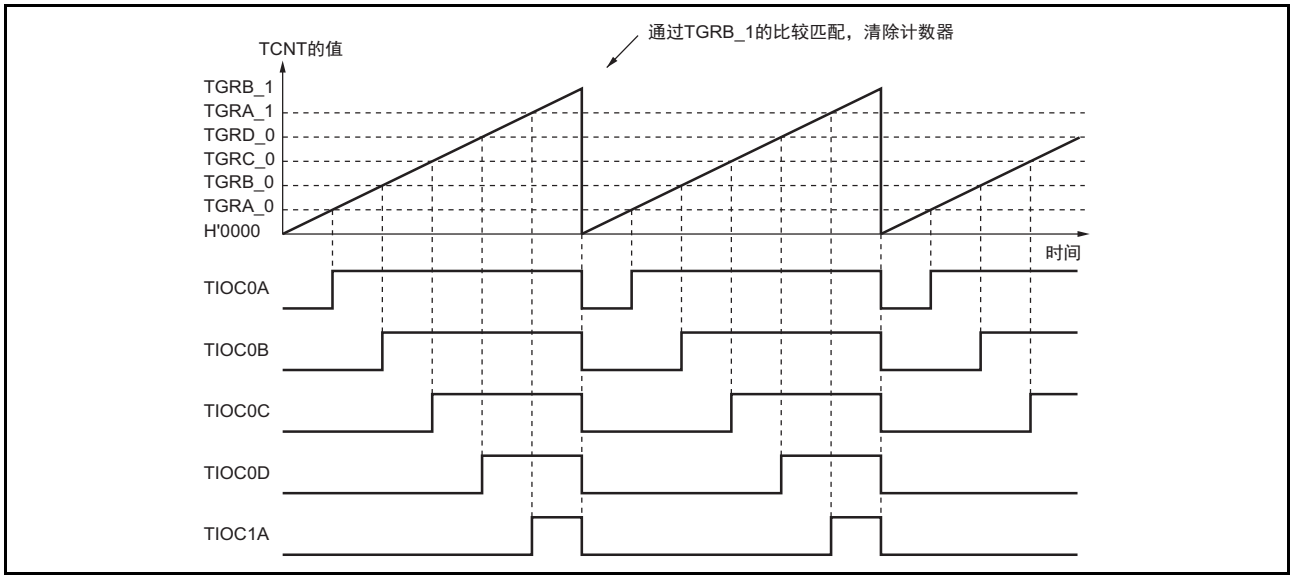


图 10.28　PWM 模式的运行例子

在 PWM 模式，输出占空比 0%和占空比 100%的 PWM 波形例子如图 10.29 所示。

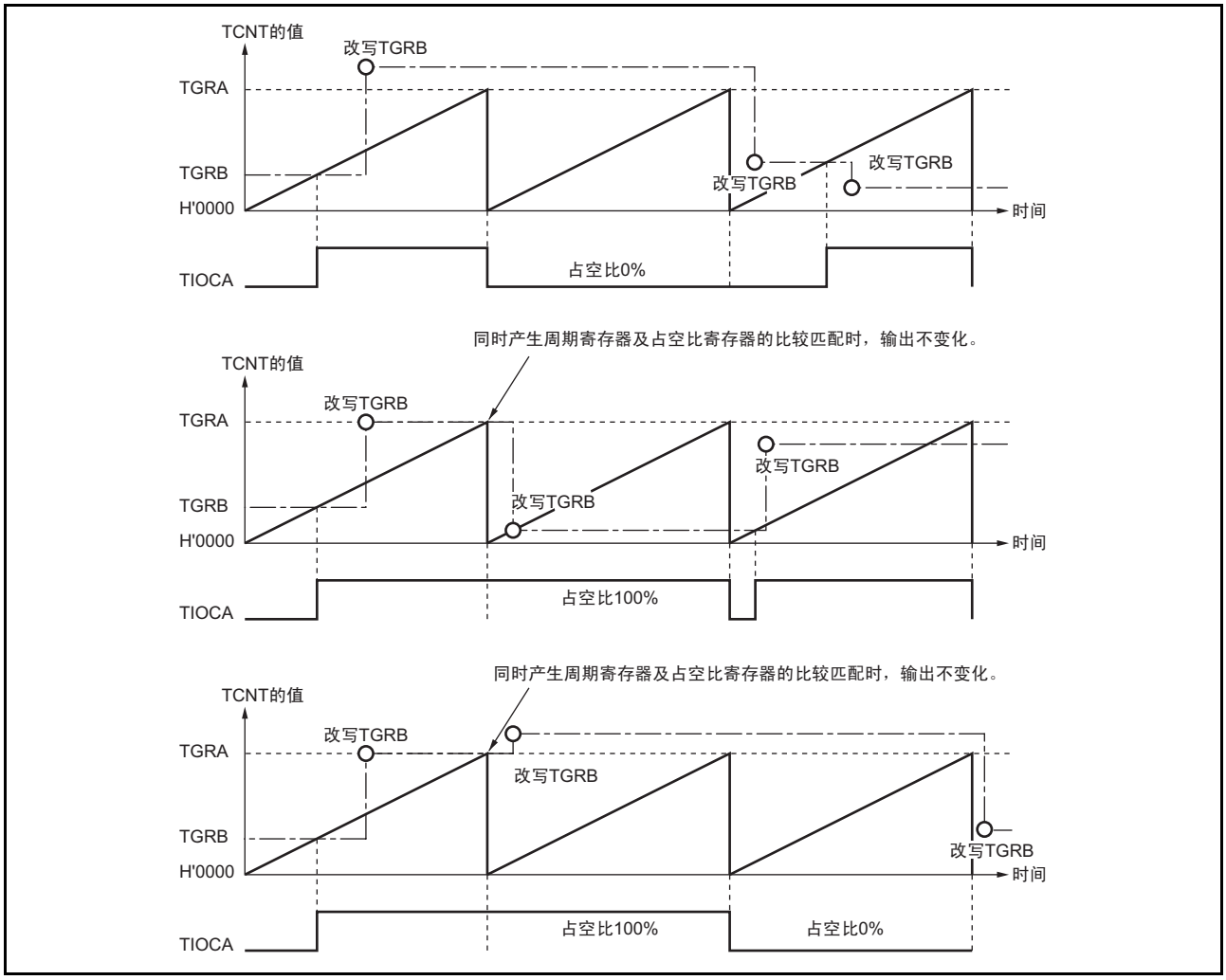


图 10.29　PWM 模式运行例子

10.4.6　相位计数模式

相位计数模式可通过通道 1、 2 的设定，检测 2 个外部时钟输入的相位差，并递增 / 递减计数 TCNT。

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。 TCR 的 CCLR1、 CCLR0 位、 TIOR、 TIER、 TGR 的功能有效，因此可使用输入捕捉 / 比较匹配功能和中断功能。

本模式可用于 2 相编码器脉冲的输入。

TCNT 递增计数时，如果产生上溢，则 TSR 的 TCFV 标志置位；递减计数时，如果产生下溢，则 TCFU 标志置位。

TSR 的 TCFD 位为计数方向标志。读取 TCFD 标志，可确认 TCNT 为递增计数或递减计数。

外部时钟引脚和通道的对应关系如表 10.50 所示。

表 10.50　相位计数模式时钟输入引脚

通道	外部时钟引脚	
	A 相	B 相
通道 1 设定为相位计数模式时	TCLKA	TCLKB
通道 2 设定为相位计数模式时	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 10.30 所示。

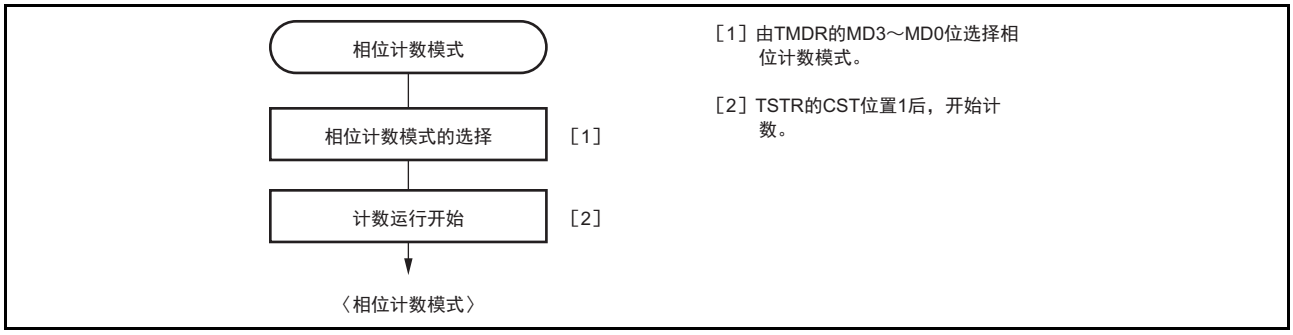


图 10.30　相位计数模式的设定步骤例子

(2)　相位计数模式运行例子

在相位计数模式，通过 2 个外部时钟的相位差， TCNT 递增 / 递减计数。按计数条件分类，有 4 种模式。

(a)　相位计数模式 1

相位计数模式 1 运行例子如图 10.31 所示， TCNT 的递增 / 递减计数条件如表 10.51 所示。

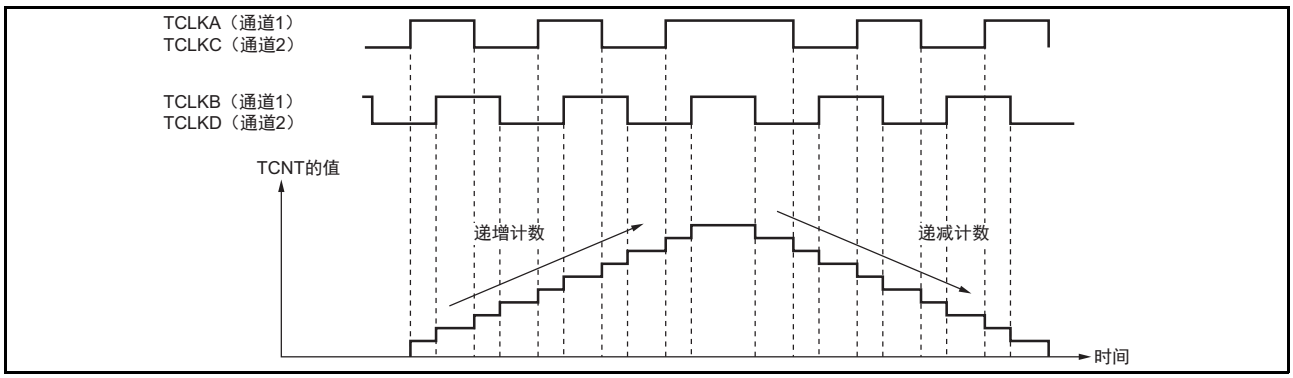


图 10.31　相位计数模式 1 的运行例子

表 10.51　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
高电平	↑	递增计数
低电平	↓	
↑	低电平	
↓	高电平	
高电平	↓	递减计数
低电平	↑	
↑	高电平	
↓	低电平	

【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 运行例子如图 10.32 所示， TCNT 的递增 / 递减计数条件如表 10.52 所示。

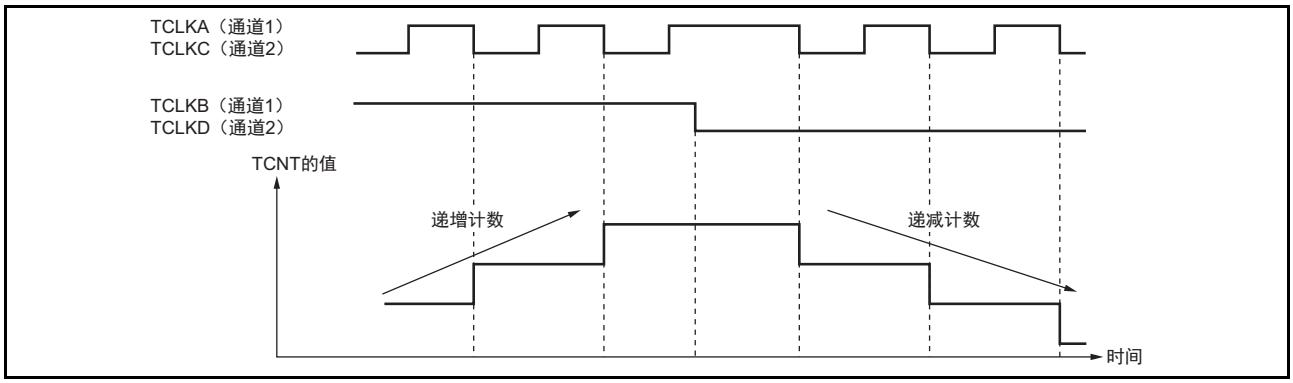


图 10.32　相位计数模式 2 的运行例子

表 10.52　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
高电平	↑	不计数（Don't care）
低电平	↓	不计数（Don't care）
↑	低电平	不计数（Don't care）
↓	高电平	递增计数
高电平	↓	不计数（Don't care）
低电平	↑	不计数（Don't care）
↑	高电平	不计数（Don't care）
↓	低电平	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 运行例子如图 10.33 所示， TCNT 的递增 / 递减计数条件如表 10.53 所示。

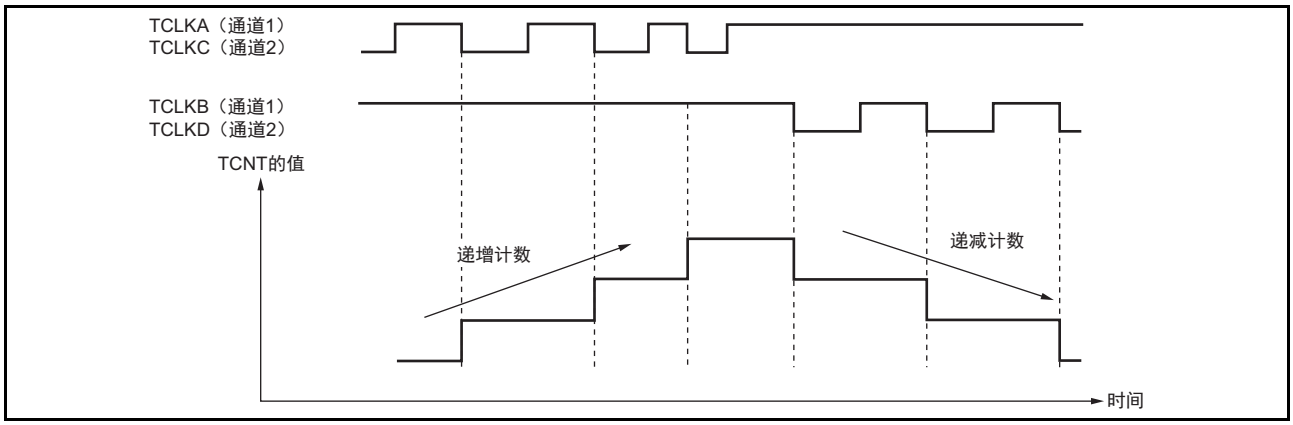


图 10.33　相位计数模式 3 的运行例子

表 10.53　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
高电平	↑	不计数 （Don't care）
低电平	↓	不计数 （Don't care）
↑	低电平	不计数 （Don't care）
↓	高电平	递增计数
高电平	↓	递减计数
低电平	↑	不计数 （Don't care）
↑	高电平	不计数 （Don't care）
↓	低电平	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 运行例子如图 10.34 所示， TCNT 的递增 / 递减计数条件如表 10.54 所示。

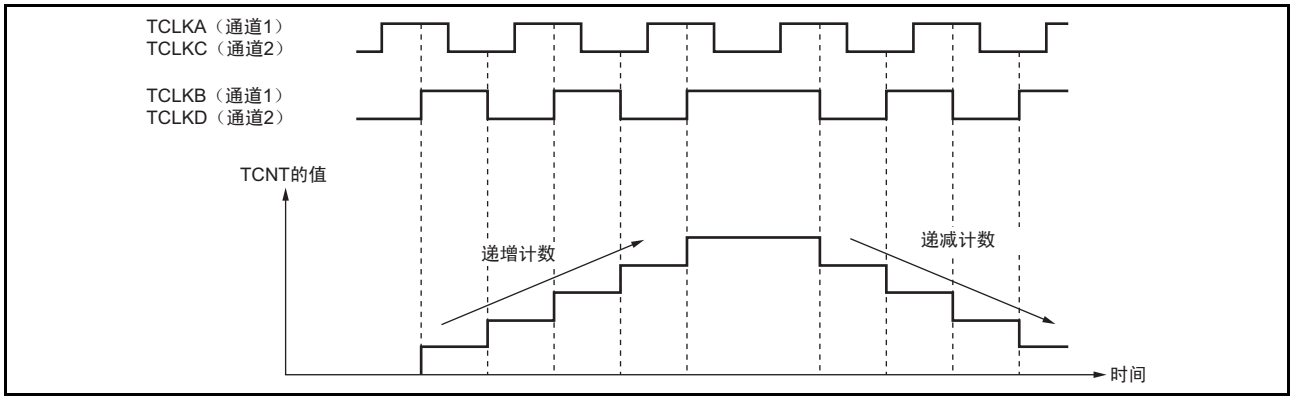


图 10.34　相位计数模式 4 的运行例子

表 10.54　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
高电平	↑	递增计数
低电平	↓	
↑	低电平	不计数 （Don't care）
↓	高电平	
高电平	↓	递减计数
低电平	↑	
↑	高电平	不计数 （Don't care）
↓	低电平	

【符号说明】

↑ ：上升沿

↓ ：下降沿

(3)　相位计数模式的应用例子

设定通道 1 为相位计数模式，并与通道 0 联合输入伺服马达的 2 相编码器脉冲后，检测位置或速度的例子如图 10.35 所示。

设定通道 1 为相位计数模式 1，将编码器脉冲的 A 相和 B 相输入至 TCLKA 和 TCLKB。

通道 0 可由 TGRC_0 的比较匹配清除 TCNT 计数器， TGRA_0 和 TGRC_0 用作比较匹配，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，使 TGRB_0 和 TGRD_0 缓冲运行。 TGRB_0 的输入捕捉源设置为通道 1 的计数器输入时钟，检测 2 相编码器的 4 倍频脉宽。

通道 1 的 TGRA_1 和 TGRB_1 设定为输入捕捉功能，输入捕捉源可选择通道 0 的 TGRA_0 和 TGRC_0 的比较匹配，并保存各自控制周期的递增 / 递减值。

由此可检测正确的位置 / 速度。

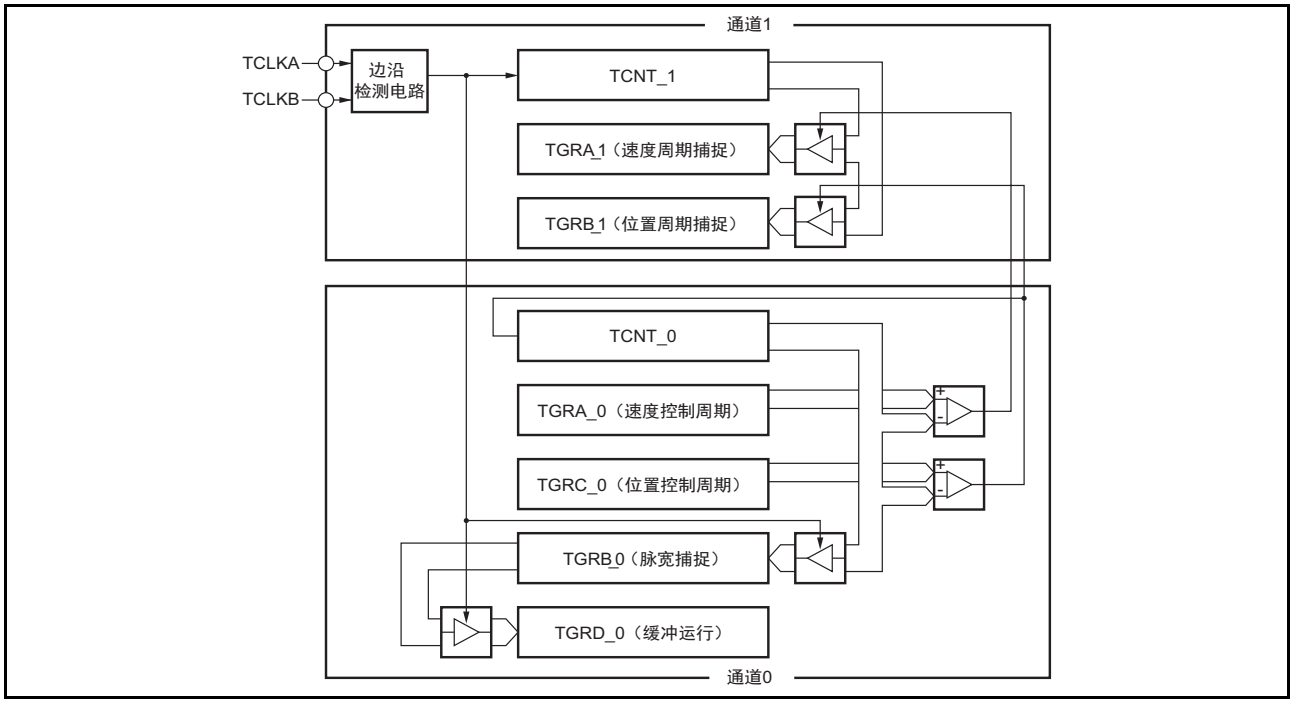


图 10.35　相位计数模式应用例子

10.4.7　复位同步 PWM 模式

在复位同步 PWM 模式，通过组合通道 3、 4，可 3 相输一方波形变化点为共同关系的 PWM 波形 （正相 / 反相）。

如果设定为复位同步 PWM 模式，则 TIOC3B、TIOC3D、TIOC4A、TIOC4C、TIOC4B 及 TIOC4D 引脚为 PWM 输出引脚，定时器计数器 3 （TCNT_3）用作递增计数器。

使用的 PWM 输出引脚和寄存器的设定分别如表 10.55 和表 10.56 所示。

表 10.55　复位同步 PWM 模式时的输出引脚

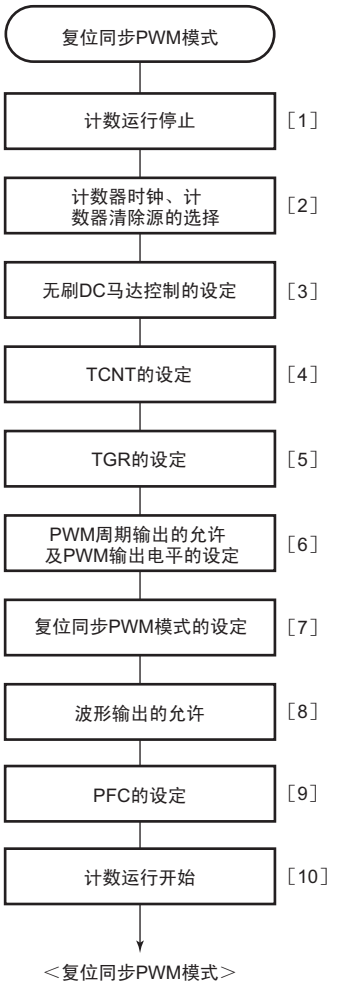
通道	输出引脚	说　　明
3	TIOC3B	PWM 输出引脚 1
	TIOC3D	PWM 输出引脚 1' （PWM 输出 1 的反相波形）
4	TIOC4A	PWM 输出引脚 2
	TIOC4C	PWM 输出引脚 2' （PWM 输出 2 的反相波形）
	TIOC4B	PWM 输出引脚 3
	TIOC4D	PWM 输出引脚 3' （PWM 输出 3 的反相波形）

表 10.56　复位同步 PWM 模式时的寄存器设定

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 PWM 波形的变化点
TGRA_4	设定从 TIOC4A、 TIOC4C 引脚输出 PWM 波形的变化点
TGRB_4	设定从 TIOC4B、 TIOC4D 引脚输出 PWM 波形的变化点

(1)　复位同步 PWM 模式的设定步骤例子

复位同步 PWM 模式的设定步骤例子如图 10.36 所示。



［1］ 定时器开始寄存器（TSTR）的CST3、CST4位清0，停止定时器计数器（TCNT）的计数。在TCNT_3、TCNT_4停止状态设定复位同步PWM模式。

［2］ 由定时器控制寄存器3（TCR_3）的TPSC2～TPSC0位及CKEG1、CKEG0位设定通道3的计数器时钟及时钟边沿。由CCLR2～CCLR0位设定计数器清除源为TGRA的比较匹配。

［3］ 控制无刷DC马达时，将定时器门控制寄存器（TGCR）的BDC位置位，并将反馈信号的输入源及输出的斩波信号或门信号设定为直接输出。

［4］ TCNT_3、TCNT_4为H'0000。

［5］ TGRA_3为周期寄存器，在TGRA_3设定周期。在TGRB_3、TGRA_4及TGRB_4设定PWM输出波形变化时序。但设定值必须在与TCNT_3比较匹配的范围内。

X≤TGRA_3　　（X：设定值）

［6］ 由定时器输出控制寄存器（TOCR1）的PSYE位选择允许/禁止与PWM周期同步的交替输出，由OLSP、OLSN位设定PWM输出电平。在缓冲运行中由TOCR2、TOLBR寄存器设定PWM输出电平时，参照图10.4。

［7］ 将定时器模式寄存器3（TMDR_3）的MD3～MD0位设定为B'1000后，再设定复位同步PWM模式。不得设定TMDR_4。

［8］ 由定时器输出主控允许寄存器（TOER）设定允许/禁止PWM波形输出引脚的输出。

［9］ 设定端口控制寄存器及端口I/O寄存器。

［10］ TSTR的CST3位置1后，开始计数。

【注】 X=TGRA_3即：设定周期＝占空比时，输出波形在TCNT_3=TGRA_3=X的点交替运行。

图 10.36　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 10.37 所示。

在复位同步 PWM 模式， TCNT_3 和 TCNT_4 作为递增计数器运行。 TCNT_3 和 TGRA_3 比较匹配时，清除计数器，并从 H'0000 重新开始递增计数。每当 TGRB_3、 TGRA_4 及 TGRB_4 分别产生比较匹配和计数器清除时， PWM 输出引脚交替输出。

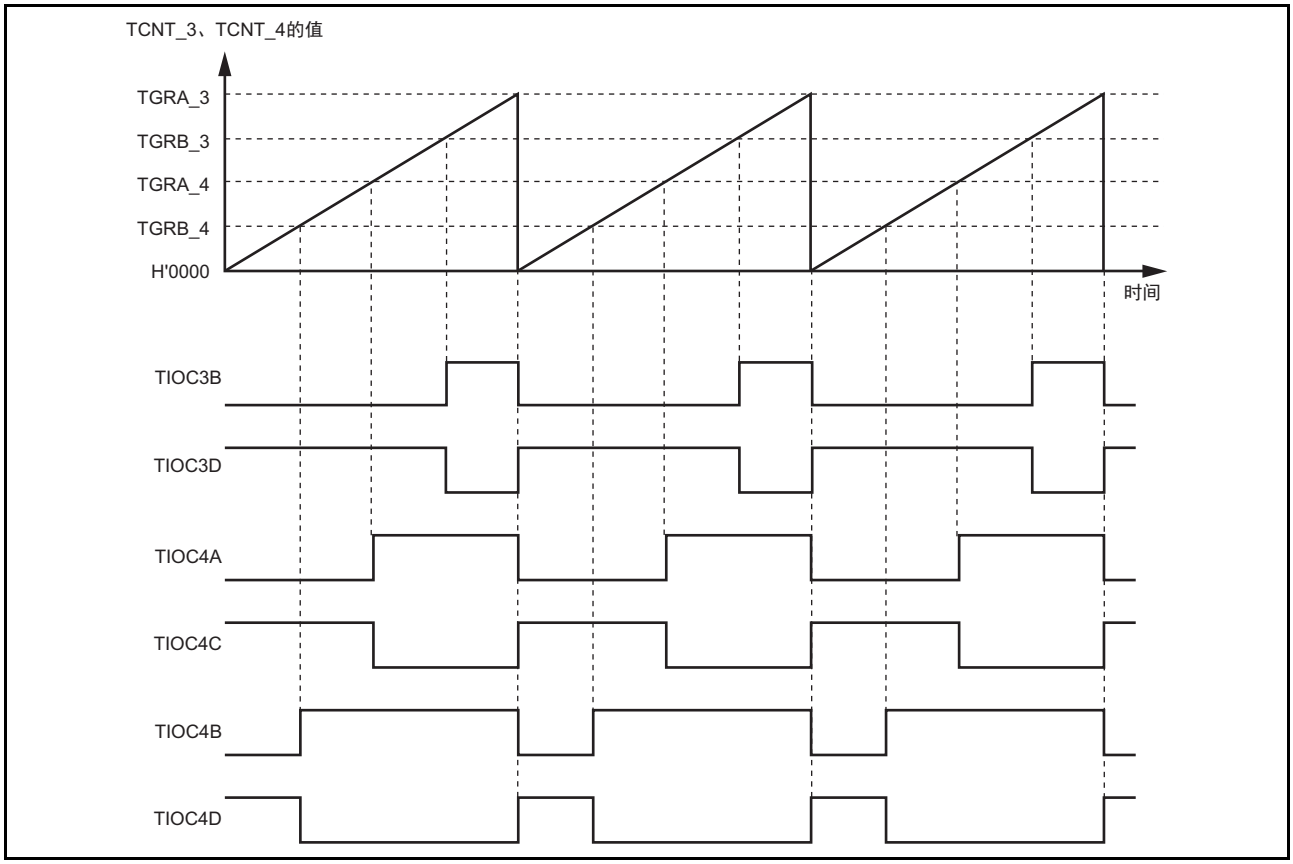


图 10.37　复位同步 PWM 模式运行例子 （设定 TOCR 的 OLSN=1、 OLSP=1 时）

10.4.8　互补 PWM 模式

在互补 PWM 模式，通过组合通道 3、 4，可 3 相输出正、反相为非重叠关系的 PWM 波形。也可设定为无非重叠时间。

如果设定为互补 PWM 模式，则 TIOC3B、 TIOC3D、 TIOC4A、 TIOC4B、 TIOC4C 及 TIOC4D 引脚为 PWM 输出引脚， TIOC3A 引脚可设定为与 PWM 周期同步的交替输出。

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 10.57 和表 10.58 所示。

端口功能支持由外部信号直接关闭 PWM 输出。

表 10.57　互补 PWM 模式时的输出引脚

通道	输出引脚	说　　明
3	TIOC3A	与 PWM 周期同步的交替输出 （或输入 / 输出端口）
	TIOC3B	PWM 输出引脚 1
	TIOC3C	输入 / 输出端口 *
	TIOC3D	PWM 输出引脚 1' （PWM 输出 1 和非重叠关系的反相波形。也可设定为无非重叠时间）
4	TIOC4A	PWM 输出引脚 2
	TIOC4C	PWM 输出引脚 2' （PWM 输出 2 和非重叠关系的反相波形。也可设定为无非重叠时间）
	TIOC4B	PWM 输出引脚 3
	TIOC4D	PWM 输出引脚 3' （PWM 输出 3 和非重叠关系的反相波形。也可设定为无非重叠时间）

【注】　*　TIOC3C 引脚在互补 PWM 模式时，不可设定为定时器输入 / 输出引脚。

表 10.58　互补 PWM 模式时的寄存器设定

通道	计数器 / 寄存器	说　　明	CPU 的读取 / 写入
3	TCNT_3	从空载时间寄存器设定的值开始递增计数	通过设定 TRWER* 可屏蔽
	TGRA_3	设定 TCNT_3 的上限值 （1/2 载波周期＋空载时间）	通过设定 TRWER* 可屏蔽
	TGRB_3	PWM 输出 1 的比较寄存器	通过设定 TRWER* 可屏蔽
	TGRC_3	TGRA_3 的缓冲寄存器	总是可读取 / 写入
	TGRD_3	PWM 输出 1/TGRB_3 的缓冲寄存器	总是可读取 / 写入
4	TCNT_4	初始设定 H'0000，并开始递增计数	通过设定 TRWER* 可屏蔽
	TGRA_4	PWM 输出 2 的比较寄存器	通过设定 TRWER* 可屏蔽
	TGRB_4	PWM 输出 3 的比较寄存器	通过设定 TRWER* 可屏蔽
	TGRC_4	PWM 输出 2/TGRA_4 的缓冲寄存器	总是可读取 / 写入
	TGRD_4	PWM 输出 3/TGRB_4 的缓冲寄存器	总是可读取 / 写入
定时器空载时间数据寄存器 （TDDR）		设定 TCNT_4 和 TCNT_3 的偏移值 （空载时间值）	通过设定 TRWER* 可屏蔽
定时器周期数据寄存器 （TCDR）		设定 TCNT_4 的上限值 （1/2 载波周期）	通过设定 TRWER* 可屏蔽
定时器周期缓冲寄存器 （TCBR）		TCDR 的缓冲寄存器	总是可读取 / 写入
副计数器 （TCNTS）		生成空载时间的副计数器	仅可读取
暂存器 1 （TEMP1）		PWM 输出 1/TGRB_3 的暂存器	不可读取 / 写入
暂存器 2 （TEMP2）		PWM 输出 2/TGRA_4 的暂存器	不可读取 / 写入
暂存器 3 （TEMP3）		PWM 输出 3/TGRB_4 的暂存器	不可读取 / 写入

【注】　*　通过 TRWER （定时器的读取写入允许寄存器）的设定，可允许 / 禁止存取。

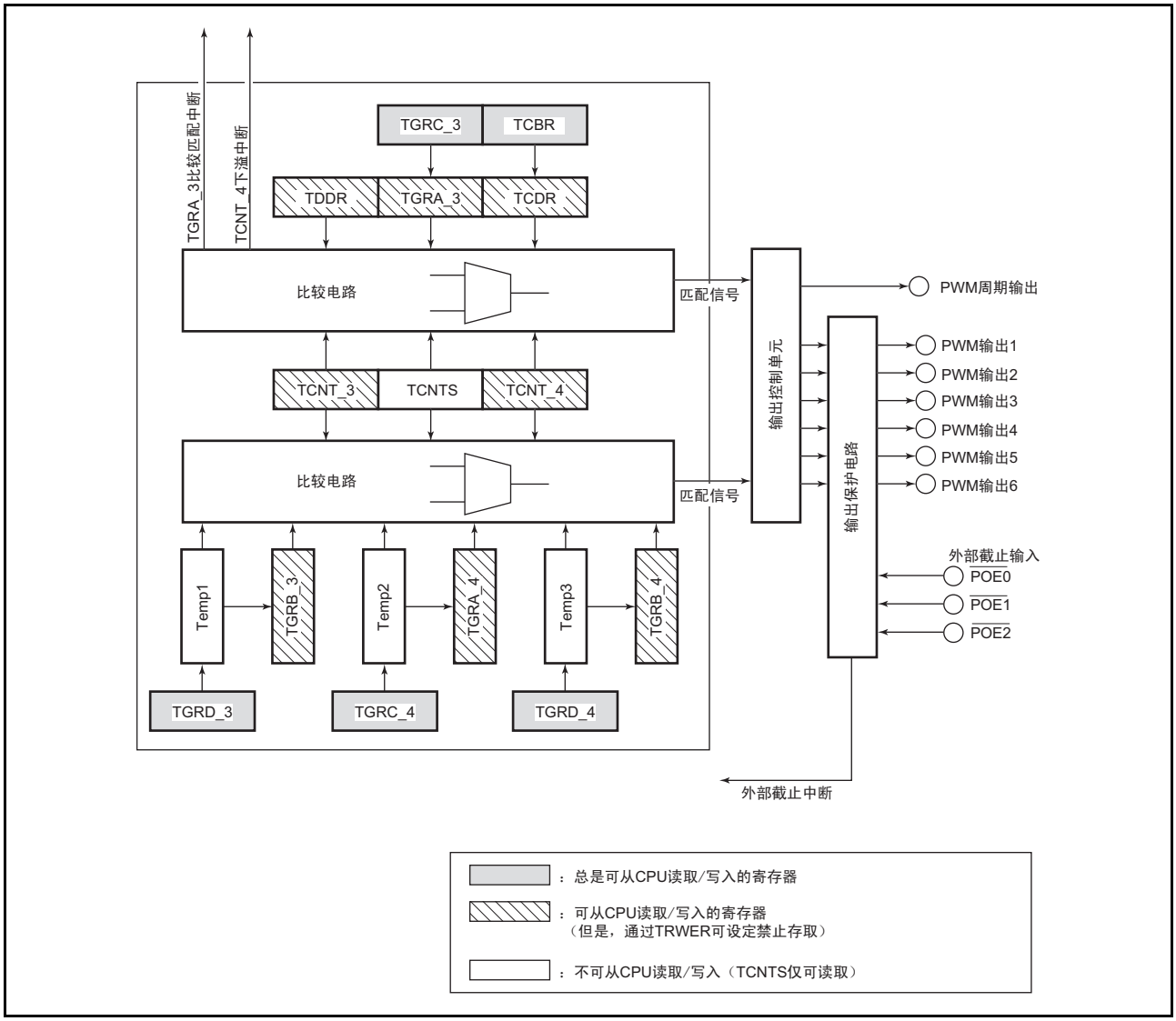


图 10.38　互补 PWM 模式时通道 3、 4 的框图

(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 10.39 所示。

图 10.39　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，可实现 6 相 PWM 输出。互补 PWM 模式的计数器和互补 PWM 模式的运行，分别如图 10.40 和图 10.41 所示。

(a)　计数器的运行

在互补 PWM 模式， TCNT_3、 TCNT_4 及 TCNTS 等 3 个计数器递增计数。

TCNT_3 设定为互补 PWM 模式及 TSTR 的 CST 位为 0 时，设定在 TDDR 的值自动设定为初始值。

CST 位设定为 1 时，递增计数直至 TGRA_3 的设定值，如果与 TGRA_3 匹配，则转换为递减计数。此后，如果与 TDDR 匹配，则转换为递增计数，重复此运行。

TCNT_4 的初始值设定为 H'0000。

CST 位设定为 1，与 TCNT_3 同步递增计数，如果与 TCDR 匹配，则转换为递减计数。之后如果与 H'0000 匹配，则转换为递增计数，重复此运行。

TCNTS 为只读计数器，无需设定初始值。

TCNT_3、 4 递增 / 递减计数时，如果 TCNT_3 和 TCDR 匹配，则开始递减计数，如果 TCNTS 与 TCDR 匹配，则转换为递增计数。如果与 TGRA_3 匹配，则被清除为 H'0000。

TCNT_3、 TCNT_4 递减计数时，如果 TCNT_4 与 TDDR 匹配，则开始递增计数，如果 TCNTS 与 TDDR 匹配，则转换为递减计数。如果与 H'0000 匹配，则设定 TCNTS 为 TGRA_3 的值。

TCNTS 仅在计数运行期间与已设定 PWM 占空比的比较寄存器及暂存器比较。

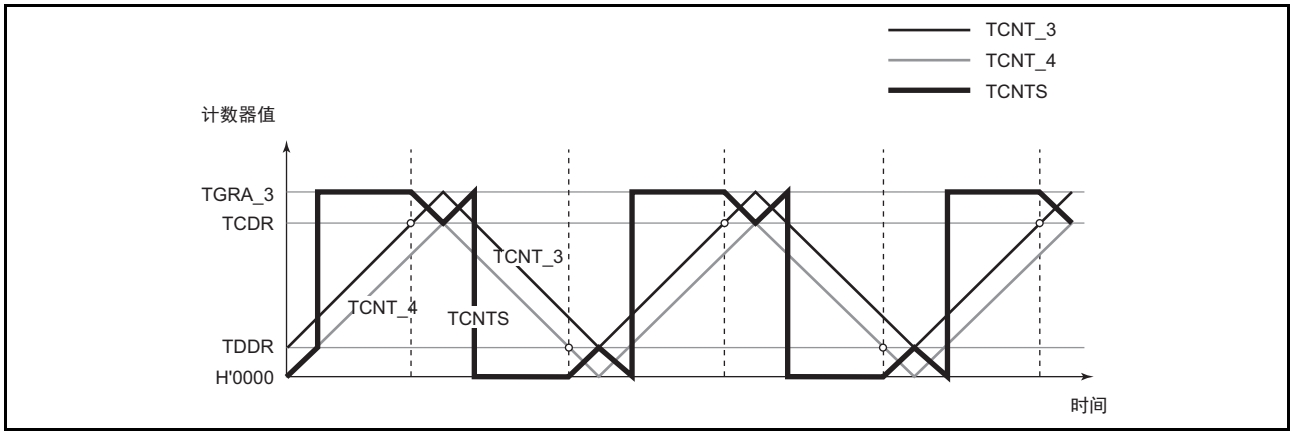


图 10.40　互补 PWM 模式的计数器运行

(b)　寄存器运行

在互补 PWM 模式，使用比较寄存器、缓冲寄存器及暂存器等 9 个寄存器。互补 PWM 模式运行例子如图 10.41 所示。

为了输出 PWM，总是与计数器比较的寄存器有：TGRB_3、 TGRA_4 及 TGRB_4。如果这些寄存器和计数器匹配，则输出由定时器输出控制寄存器 （TOCR）的 OLSN、 OLSP 位设定的值。

比较寄存器的缓冲寄存器为 TGRD_3、 TGRC_4 及 TGRD_4。

在缓冲寄存器和比较寄存器之间有暂存器，但不可从 CPU 存取暂存器。

为了更改比较寄存器的数据，必须在相应的缓冲寄存器写入更改数据。缓冲寄存器随时可读取 / 写入。

写入缓冲寄存器的数据，在 Ta 区间随时可传送到暂存器，但在 Tb 区间不可传送至暂存器。 Tb 区间结束后，将在此区间写入到缓冲寄存器的数据传送至暂存器。

如果 Tb 区间结束， TCNTS 递增计数时与 TGRA_3 匹配或递减计数时与 H'0000 匹配，将传送到暂存器的值传送到比较寄存器。可由定时器模式寄存器 （TMDR）的 MD3 ～ MD0 位，选择从暂存器到比较寄存器的传送时序。在波谷选择更改模式的例子如图 10.41 所示。

在不向暂存器传送数据的 Tb （图 10.41 中的 Tb1）区间，暂存器具有和比较寄存器相同的功能，可与计数器比较。因此，在此区间， 2 个比较匹配寄存器输出 1 相，比较寄存器为更改前的数据，暂存器为新更改的数据。在此区间， TCNT_3、 4 和 TCNTS （3 个计数器）与比较寄存器、暂存器这 2 个寄存器比较，可控制 PWM 输出。

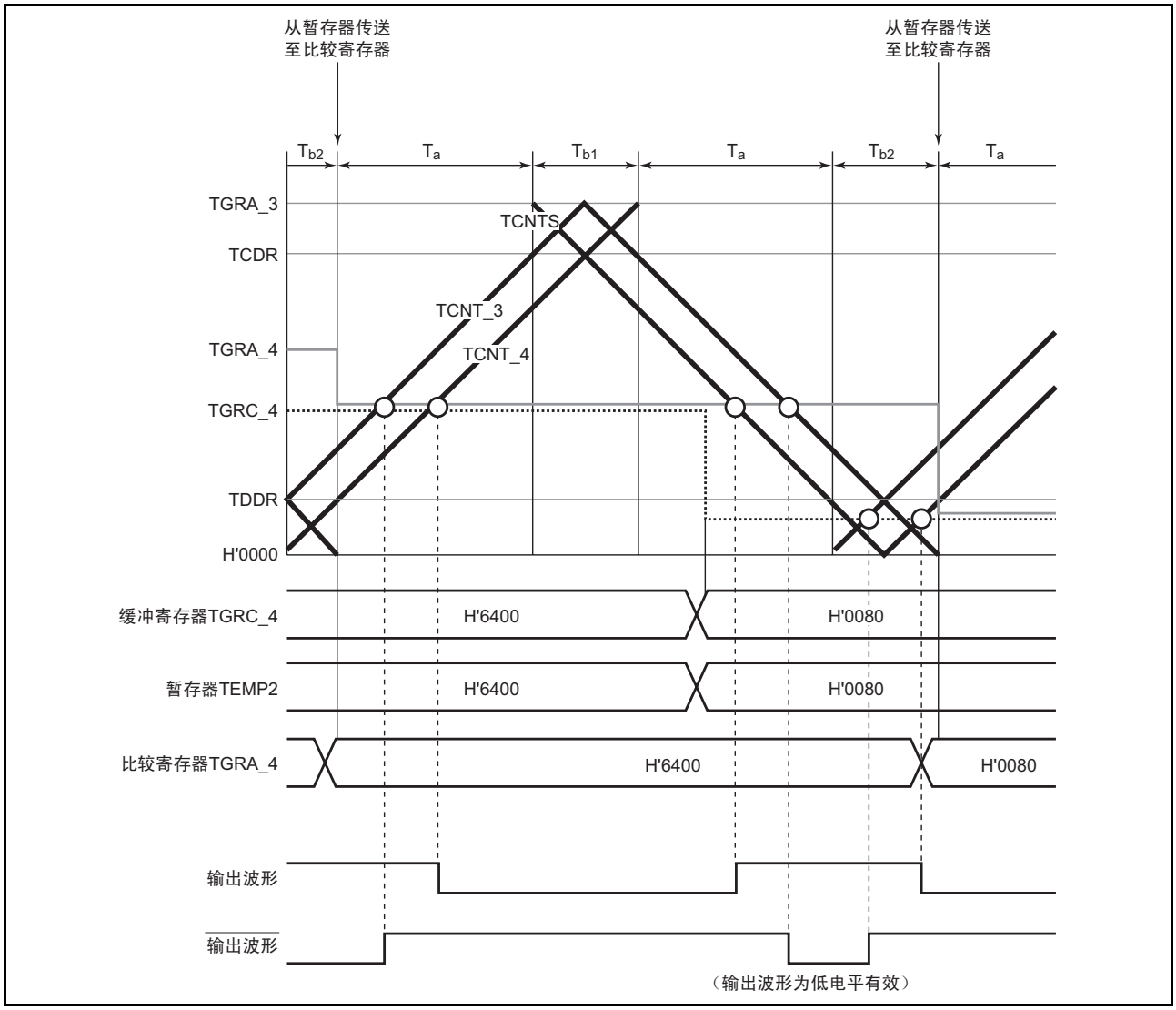


图 10.41　互补 PWM 模式运行例子

(c)　初始设定

在互补 PWM 模式，有 6 个寄存器需要初始设定，有 1 个寄存器需要设定是否生成空载时间 （仅在不生成空载时间时设定）。

通过定时器模式寄存器 （TMDR）的 MD3 ～ MD0 位设定为互补 PWM 模式前，必须设定以下寄存器的初始值：

TGRC_3 作为 TGRA_3 的缓冲寄存器运行，设定 1/2PWM 载波周期 + 空载时间 Td。定时器周期缓冲寄存器 （TCBR）作为定时器周期数据寄存器 （TCDR）的缓冲寄存器运行，设定 1/2PWM 载波周期。并且，在定时器空载时间数据寄存器 （TDDR）设定空载时间 Td。

不生成空载时间时，将定时器空载时间允许寄存器 （TDER）的 TDER 位设定为 0，在 TGRC_3、TGRA_3 设定 1/2PWM 空载周期 +1，在 TDDR 设定 1。

在 3 个缓冲寄存器 TGRD_3、 TGRC_4 及 TGRD_4，可分别设定 PWM 占空比的初始值。

设定为互补 PWM 模式的同时，将在 TDDR 之外的 5 个缓冲寄存器设定的值传送至各对应的比较寄存器。

设定为互补 PWM 模式前，必须将 TCNT_4 设置为 H'0000。

表 10.59　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	1/2PWM 载波周期 + 空载时间 Td （通过 TDER 设定为不生成空载时间时，为 1/2PWM 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，为 1）
TCBR	1/2PWM 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 TCBR 设定的 1/2 PWM 载波周期值加上 TDDR 设定的空载时间值之和。如果通过 TDER 设
定为不生成空载时间时，必须设置为 1/2PWM 载波周期 +1。

(d)　PWM 输出电平的设定

在互补 PWM 模式，可通过定时器输出控制寄存器 1 （TOCR1）的 OLSN、 OLSP 位，或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。

6 相输出电平可设定为正反各 3 相。

必须在解除互补 PWM 模式的状态，设定 / 更改输出电平。

(e)　空载时间的设定

在互补 PWM 模式，可输出正反相为非重叠关系的 PWM 脉冲。将此非重叠时间称为空载时间。

在定时器空载时间数据寄存器 （TDDR）可设定非重叠时间。设定在 TDDR 的值为 TCNT_3 的计数器开始值，生成 TCNT_3 和 TCNT_4 的非重叠时间。必须在解除互补 PWM 模式的状态，更改 TDDR 的内容。

(f)　设定不生成空载时间

设定不生成空载时间时，可将定时器空载时间允许寄存器 （TDER）的 TDER 位设定为 0。仅可在 TDER=1 的状态，读取 TDER 后对 TDER 写入 0 时，将 TDER 置 0。

对 TGRA_3、 TGRC_3 设定 1/2PWM 载波周期 +1，定时器空载时间数据寄存器 （TDDR）设定 1。

如果设定不生成空载时间，则输出无空载时间的 PWM 波形，不生成空载时间的运行例子如图 10.42 所示。

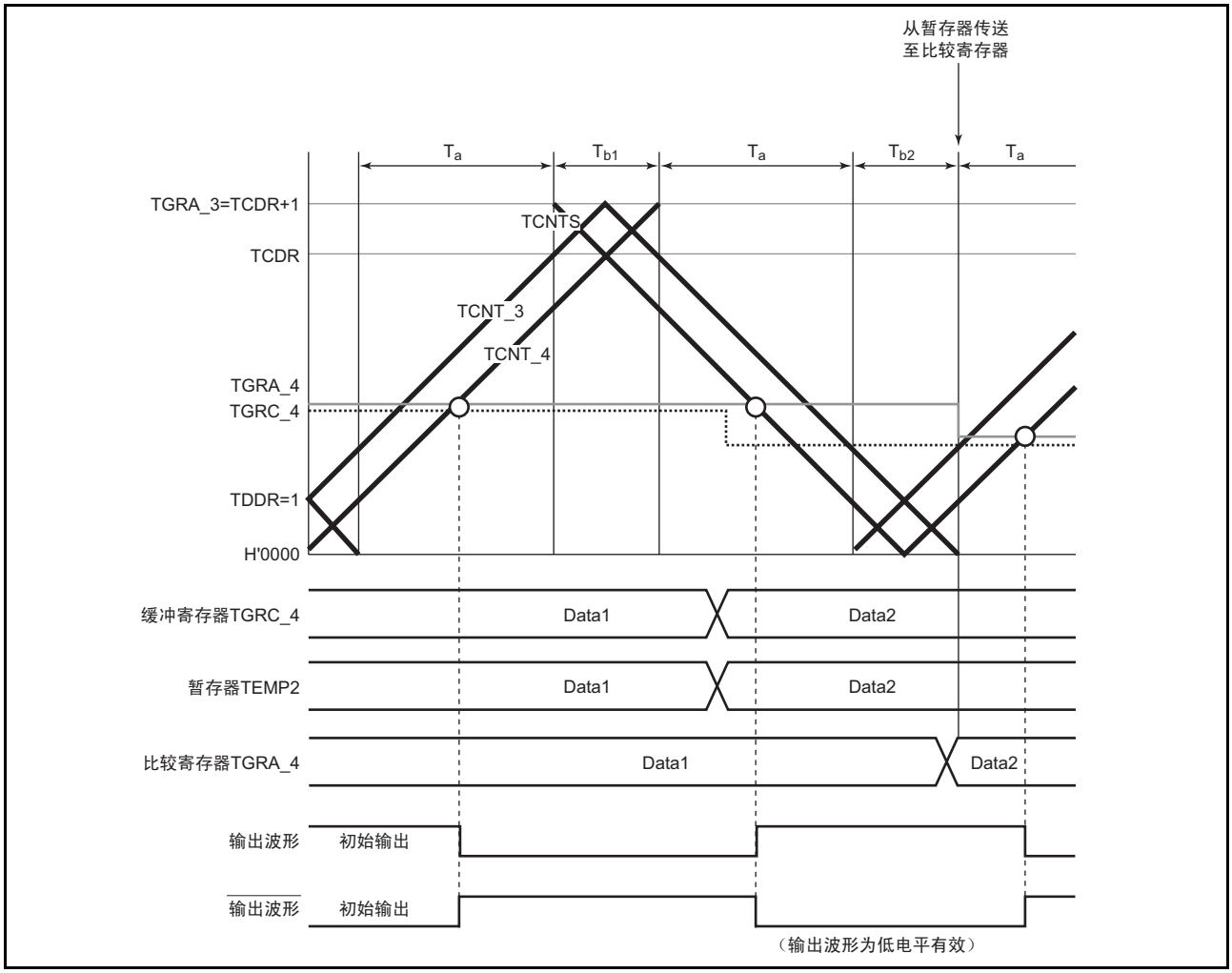


图 10.42　不生成空载时间的运行例子

(g)　PWM 周期的设定

在互补 PWM 模式，可在 TGRA_3 和 TCDR 这 2 个寄存器将 PWM 脉冲周期设定为 TCNT3 的上限值和 TCNT_4 的上限值。

2 个寄存器的关系如下设定：

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

必须通过给缓冲寄存器 TGRC_3、 TCBR 设定值进行 TGRA_3、 TCDR 的设定。根据定时器模式寄存器（TMDR）的 MD3 ～ MD0 选择的传送时序，将在 TGRC_3、 TCBR 设定的值同时传送至 TGRA_3、 TCDR。

在波谷更新数据时，从下一周期反映更改的 PWM 周期；在波峰更新数据时，从该周期反映更改的 PWM 周期。在波峰更改 PWM 周期的运行例子如图 10.43 所示。

另外，有关各缓冲寄存器的数据更新方法，详情参阅以下 “(h) 寄存器数据的更新”。

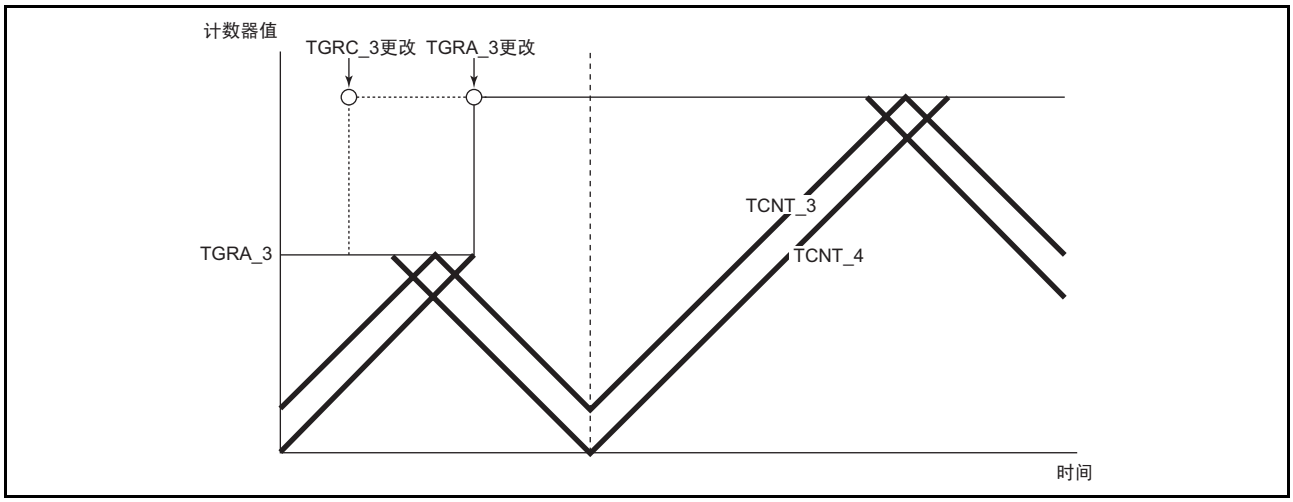


图 10.43　PWM 周期的更改例子

(h)　寄存器数据的更新

在互补 PWM 模式，更新比较寄存器的数据时使用缓冲寄存器，更新数据可随时写入缓冲寄存器。带有缓冲寄存器的运行过程中， 5 个用于 PWM 占空比和载波周期的寄存器可更新。

这些寄存器与缓冲寄存器之间有各自的暂存器。副计数器 TCNTS 停止计数期间，更新缓冲寄存器数据时，暂存器值也随之改写。 TCNTS 计数时，不从缓冲寄存器向暂存器传送。 TCNTS 停止后，可传送写入缓冲寄存器的值。

根据定时器模式寄存器 （TMDR）的 MD3 ～ MD0 位设定的数据更新时序，向比较寄存器传送暂存器值。互补 PWM 模式的数据更新例子如图 10.44 所示，此图为在计数器波峰与波谷更新数据的模式。

改写缓冲寄存器数据时，最后必须写入 TGRD_4。写入 TGRD_4 后， 5 个寄存器同时从缓冲寄存器向暂存器传送数据。

另外，不更新 5 个寄存器或不更新 TGRD_4 数据时，在写入要更新的寄存器数据后，必须写入 TGRD_4，此时写入 TGRD_4 的数据应与写入前相同。

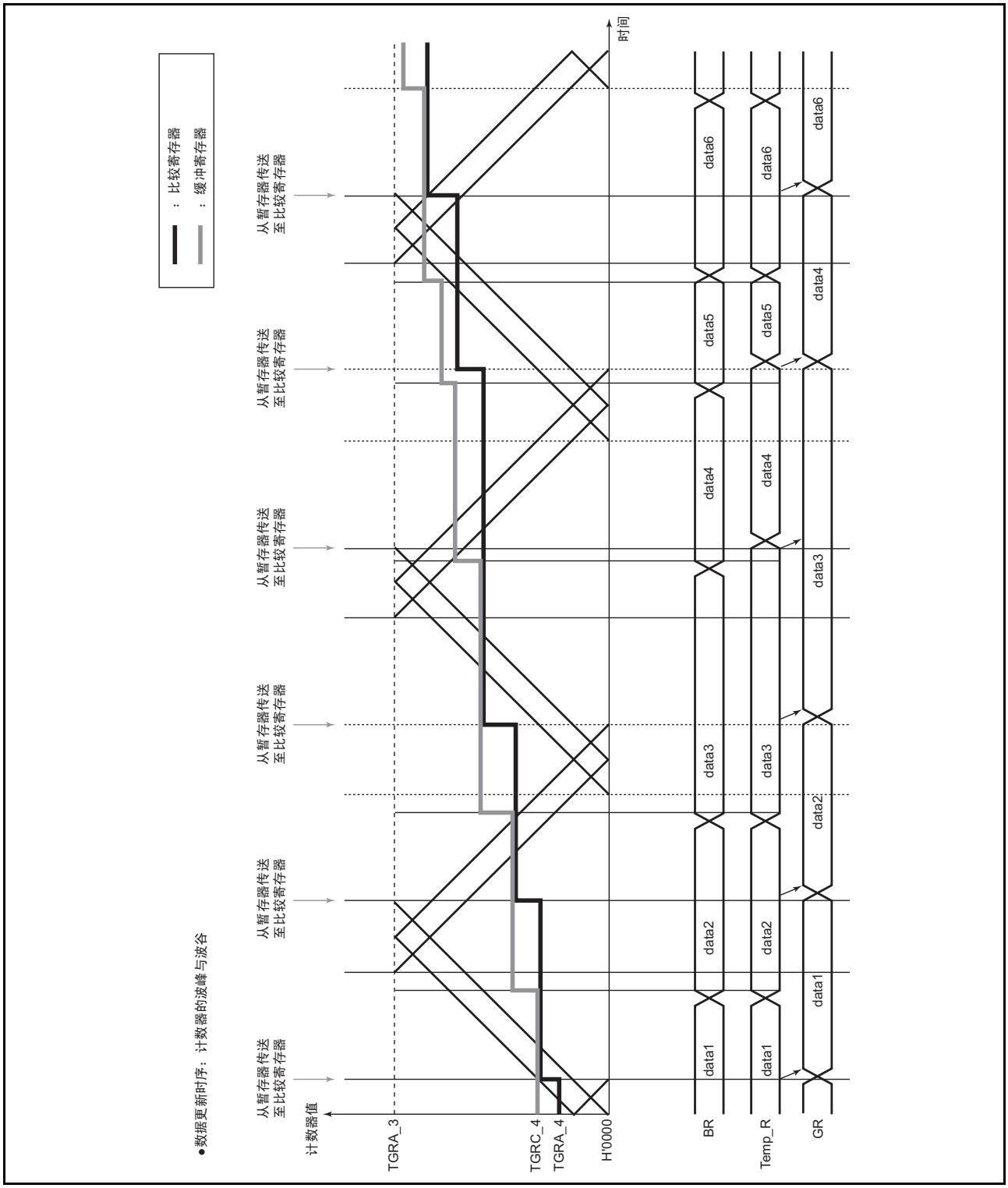


图 10.44　互补 PWM 模式的数据更新例子

(i)　互补 PWM 模式的初始输出

在互补 PWM 模式，由定时器输出控制寄存器 1 （TOCR1）中 OLSN、 OLSP 位的设定或者定时器输出控制寄存器 2 （TOCR2）中 OLS1N ～ OLS3N、 OLS1P ～ OLS3P 位的设定决定初始输出。

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 大于空载定时器寄存器 （TDDR）设定的值为止，输出此初始输出。互补 PWM 模式的初始输出例子如图 10.45 所示。

PWM 占空比的初始值小于 TDDR 值时的波形例子如图 10.46 所示。

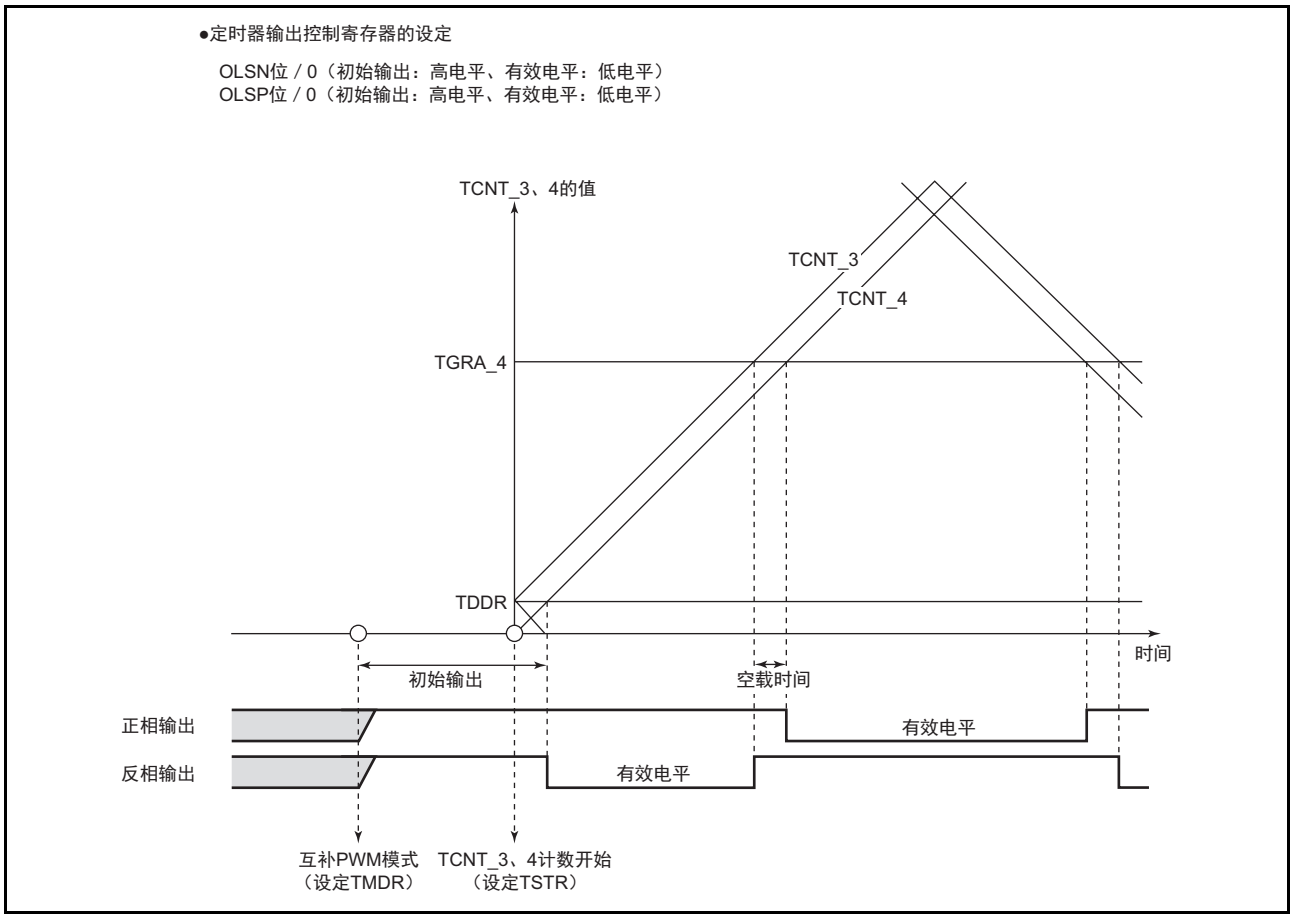


图 10.45　互补 PWM 模式的初始输出例子 （1）

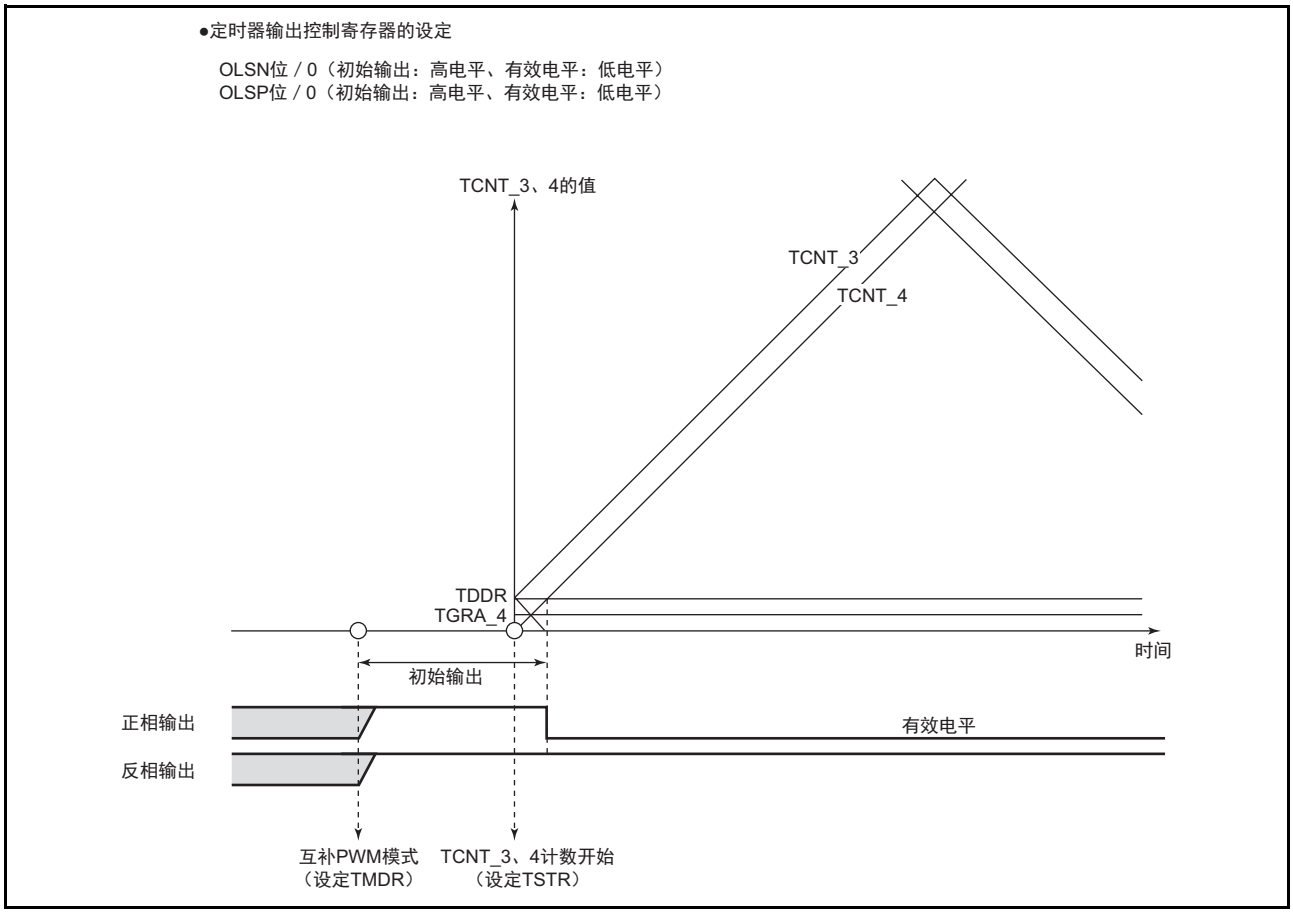


图 10.46　互补 PWM 模式的初始输出例子 （2）

(j)　互补 PWM 模式的 PWM 输出生成方法

在互补 PWM 模式， 3 相输出带有非重叠时间的正 / 反相 PWM 波形，将此非重叠波形时间称为空载时间。

计数器与数据寄存器产生比较匹配时，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。TCNTS 计数期间，为了生成 0 ～ 100% 的连续 PWM 脉冲，可同时比较数据寄存器值与暂存器值，此时，产生 ON、 OFF 的比较匹配的时序可能会颠倒，为了确保空载时间，并避免重叠正 / 反相的 ON 时间，必须使 OFF 各项的比较匹配优先。互补 PWM 模式的波形生成例子如图 10.47 ～图 10.49 所示。

通过与实线计数器的比较匹配生成正 / 反相的 OFF 时序，通过与虚线计数器 （从实线计数器开始延迟运行，其延迟时间为空载时间）比较匹配后生成 ON 时序。在 T1 期间，优先 OFF 反相的 a 的比较匹配，忽略比 a 先产生的比较匹配；在 T2 期间，优先 OFF 正相的 c 的比较匹配，忽略比 c 先产生的比较匹配。

如图 10.47 所示，通常的比较匹配按 a→b→c→d （或 c→d→a'→b'）的顺序产生。

比较匹配不符合 a→b→c→d 的顺序时，反相 OFF 的时间小于 2 倍的空载时间，因此正相不为 ON ；不符合 c→d→a'→b' 的顺序时，正相 OFF 的时间小于 2 倍的空载时间，因此反相不为 ON。

如图 10.48 所示，在 a 的比较匹配之后先产生 c 的比较匹配时，可忽略 b 的比较匹配并由 d 的比较匹配使反相 OFF。这是由于正相的 OFF 时序即 c 的比较匹配的产生先于正相 ON 时序即 b 的比较匹配，因此可优先 OFF 正相 （正相从 OFF 到 OFF，因此波形不变化）。

同样，如图 10.49 所示，在 c 的比较匹配前产生与暂存器的新数据比较匹配的 a'，但在产生 OFF 正相的 c 前，可忽略其他比较匹配，因此，反相不为 ON。

在互补 PWM 模式，可优先进行 OFF 时序的比较匹配， ON 时序的比较匹配即使产生在 OFF 前也被忽略。

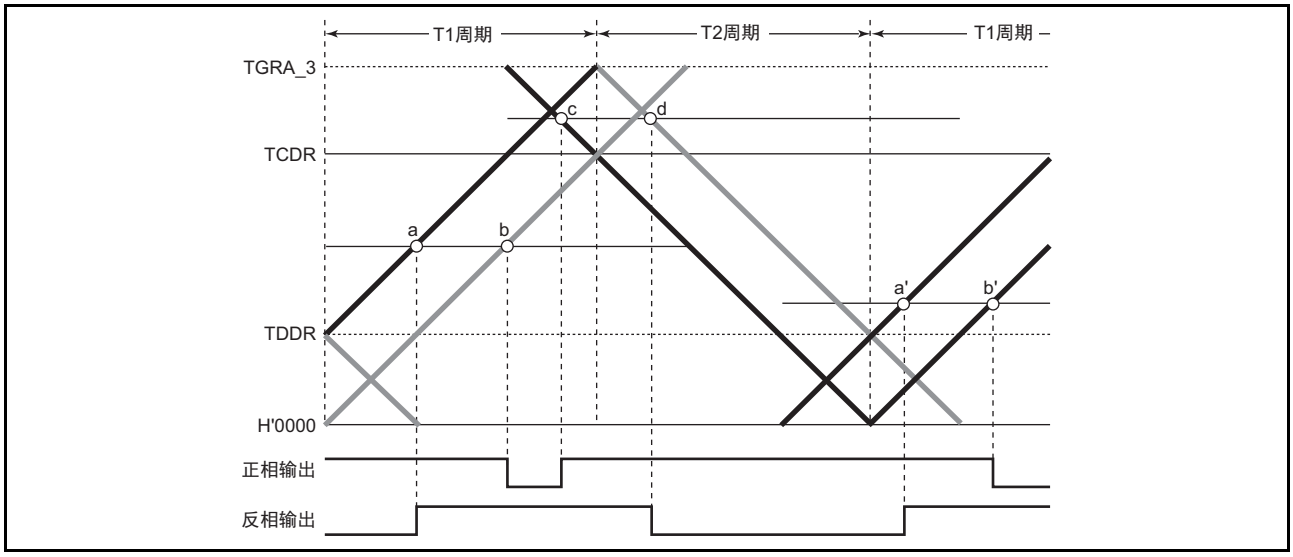


图 10.47　互补 PWM 模式波形输出例子 （1）

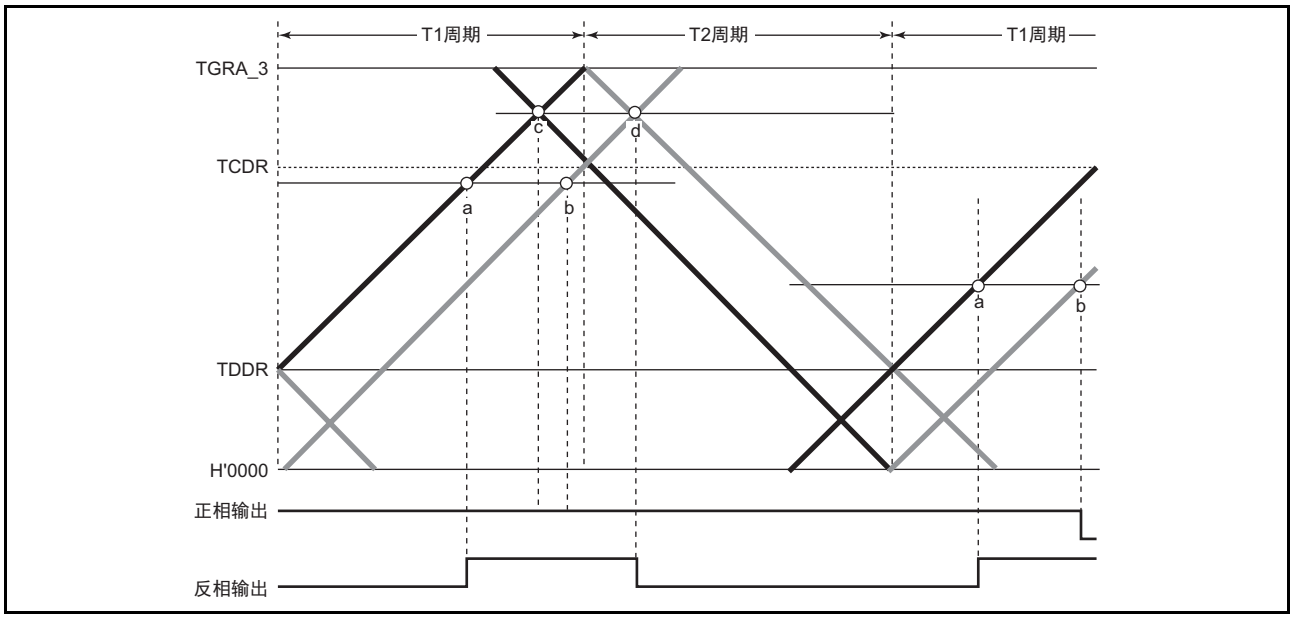


图 10.48　互补 PWM 模式波形输出例子 （2）

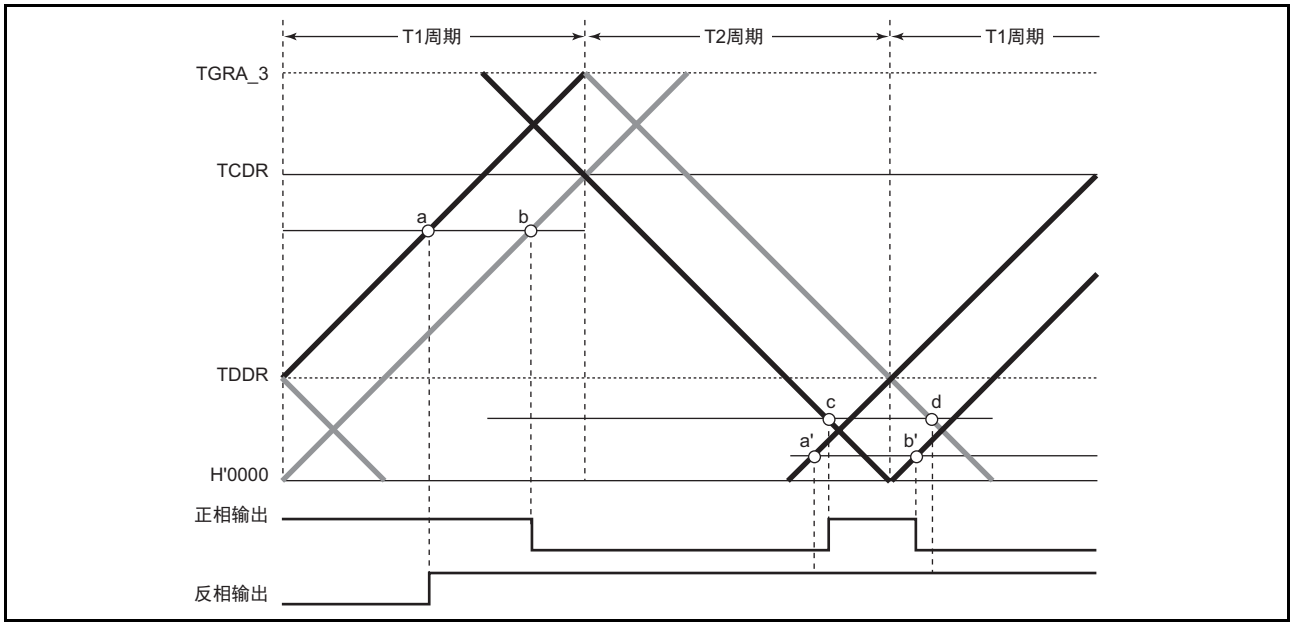


图 10.49　互补 PMW 模式波形输出例子 （3）

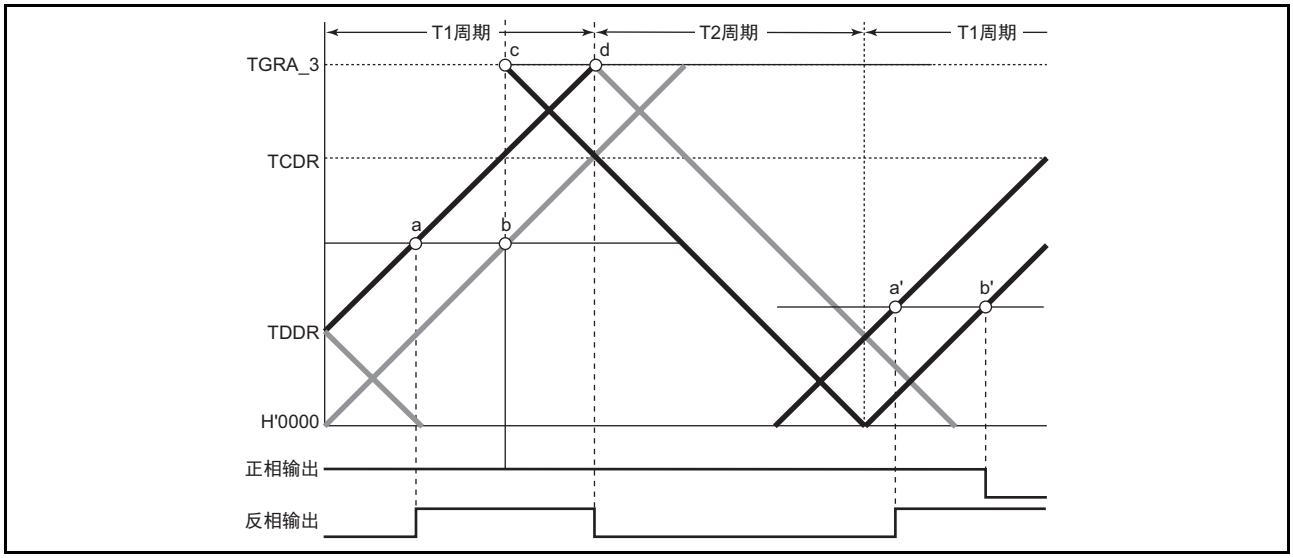


图 10.50　互补 PWM 模式 0%、 100% 波形输出例子 （1）

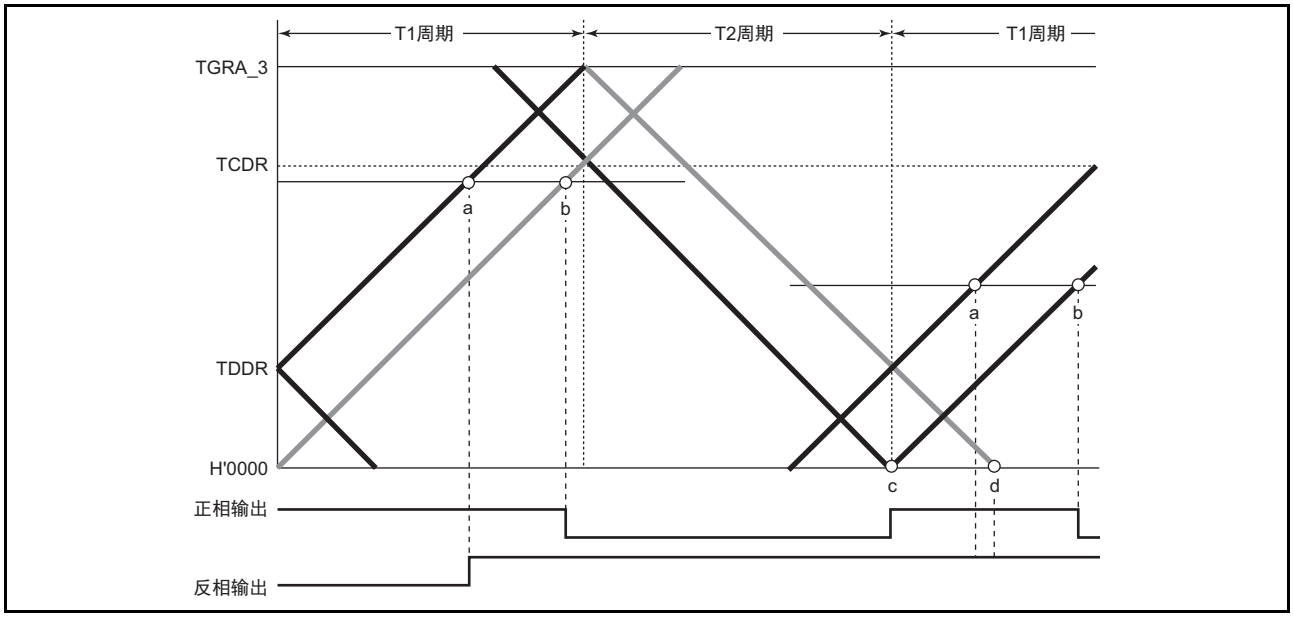


图 10.51　互补 PWM 模式 0%、 100% 波形输出例子 （2）

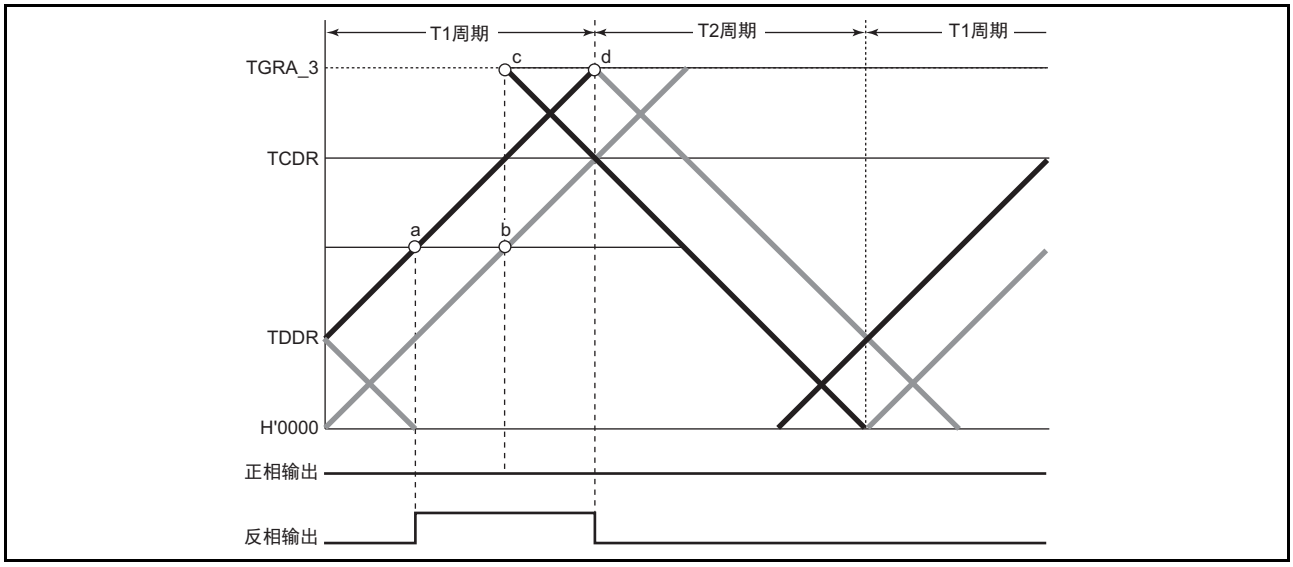


图 10.52　互补 PWM 模式 0%、 100% 波形输出例子 （3）

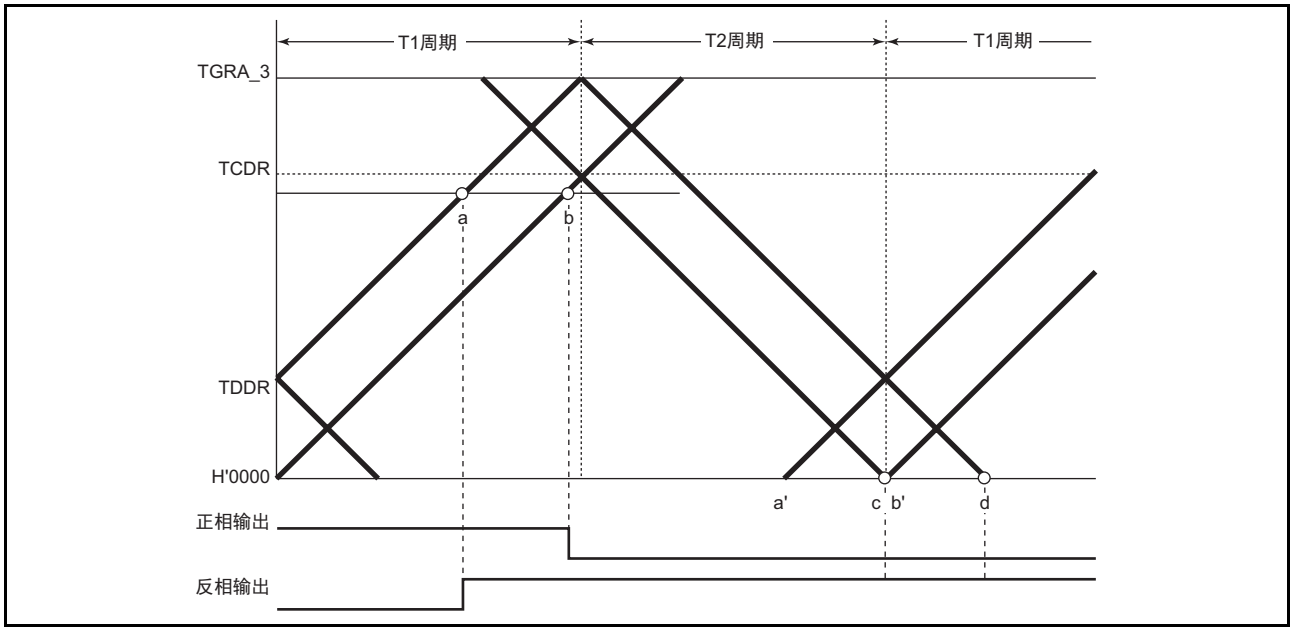


图 10.53　互补 PWM 模式 0%、 100% 波形输出例子 （4）

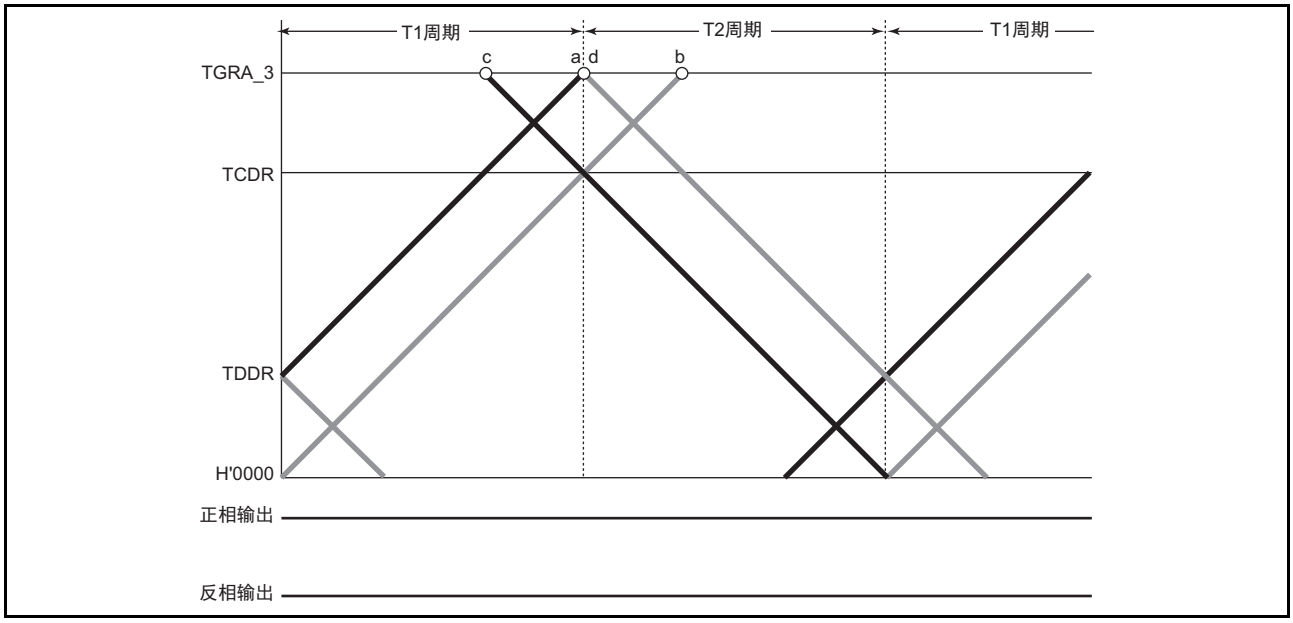


图 10.54　互补 PWM 模式 0%、 100% 波形输出例子 （5）

(k)　互补 PWM 模式的占空比 0%、 100% 的输出

在互补 PWM 模式，可任意输出占空比 0%、 100%，如图 10.50 ～图 10.54 所示。

将数据寄存器的值设定为 H'0000 时，输出占空比 100%，此时的波形为正相 100% ON 状态的波形；如果将数据寄存器的值设定为与 TGRA_3 相同的值，则输出占空比 0%，此时的波形为正相 100% OFF 状态的波形。

此时，同时产生 ON 和 OFF 的比较匹配，但是，如果同时产生同相的 ON 的比较匹配和 OFF 的比较匹配，则忽略双方的比较匹配，且波形不变。

(l)　与 PWM 周期同步的交替输出

在互补 PWM 模式，通过将定时器输出控制寄存器 （TOCR）的 PSYE 位置 1，可与 PWM 载波周期同步交替输出，交替输出的波形如图 10.55 所示。

由 TCNT_3 与 TGRA_3 的比较匹配和 TCNT4 与 H'0000 的比较匹配执行交替输出。

此交替输出引脚为 TIOC3A 引脚，初始输出为 1。

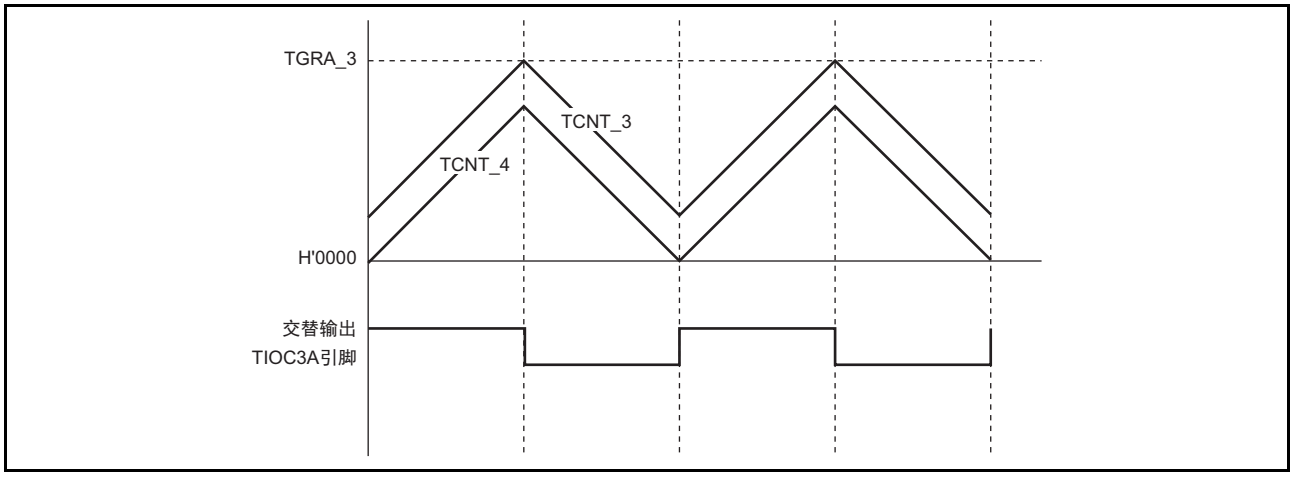


图 10.55　与 PWM 输出同步的交替输出波形例子

(m)　通过其他通道清除计数器

在互补 PWM 模式，由定时器同步寄存器 （TSYR）设定为与其他通道的同步模式，并由定时器控制寄存器 （TCR）的 CCLR2 ～ CCLR0 选择同步清除，即可通过其他通道清除 TCNT_3、 TCNT_4 及 TCNTS。

运行例子如图 10.56 所示。

使用此功能，可通过外部信号清除计数器及重新开始计数。

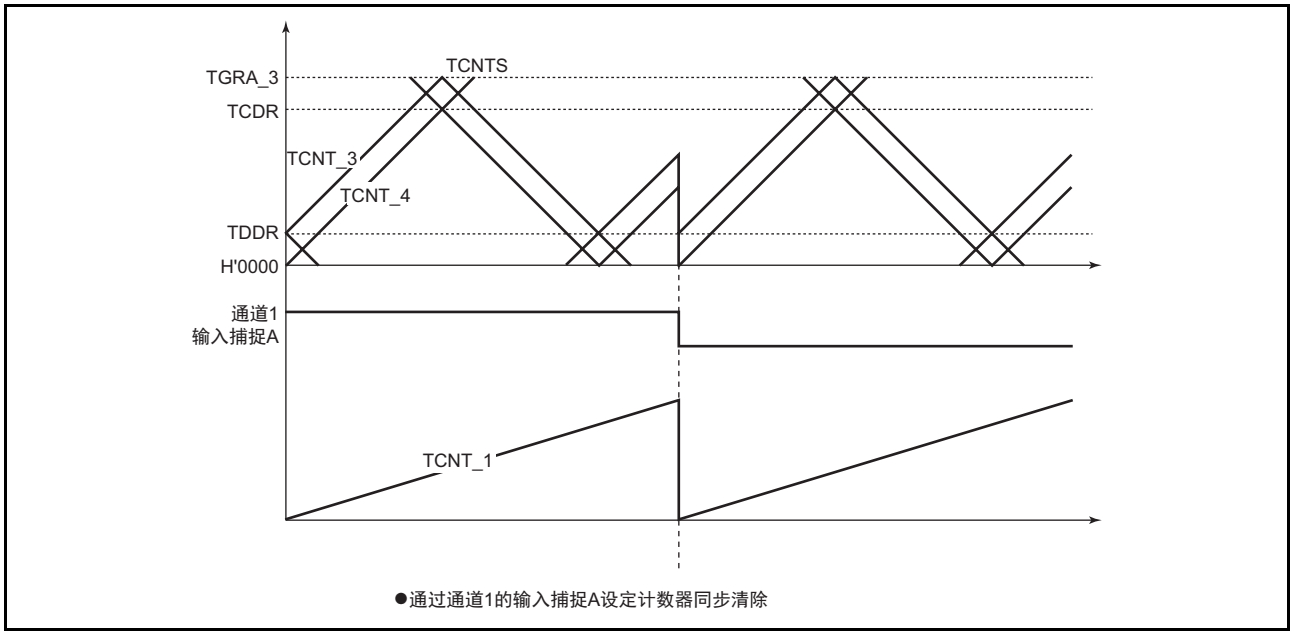


图 10.56　清除与其他通道同步的计数器

(n)　在互补 PWM 模式同步清除计数器时输出波形的控制

将 TWCR 寄存器的 WRE 位设定为 1，可抑制在互补 PWM 模式波谷的 Tb 区间同步清除计数器时的初始输出。由此，可抑制同步清除计数器时占空比的急剧变化。

仅在图 10.57 的 （10）、 (11) 所示的波谷 Tb 区间执行同步清除时，才可通过将 WRE 位设定为 1，抑制初始输出。在其他时序产生同步清除时，可输出由 TOCR 寄存器的 OLS 位设定的初始值。另外，即使在波谷 Tb 区间，如果在图 10.57 的 (1) 所示的计数器刚开始的初始输出期间产生同步清除，也不抑制初始输出。

MTU2 或 MTU2S 均可使用本功能。 MTU2、 MTU2S 的计数器清除源分别为 MTU2 通道 0 ～ 2 的同步清除和 MTU2 通道 0 ～ 2 的标志置位 （比较匹配 / 输入捕捉）。

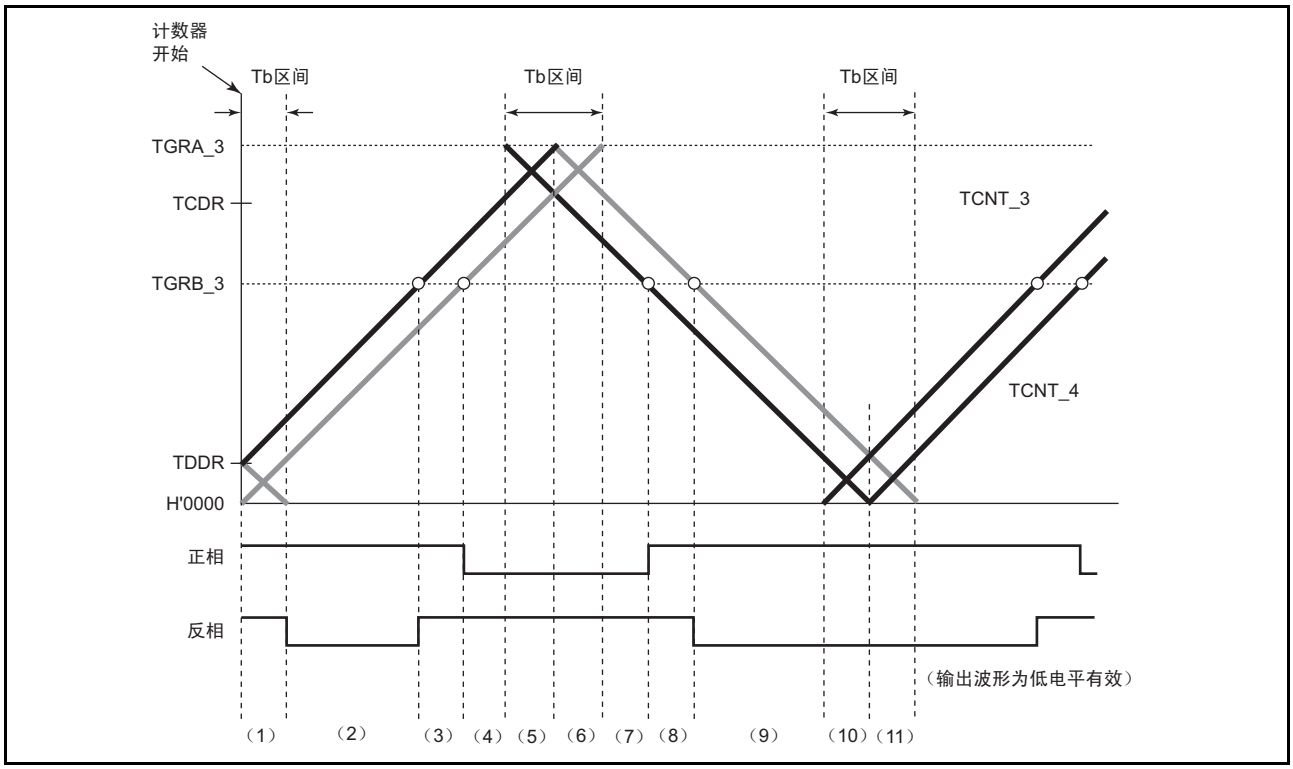


图 10.57　同步清除计数器的时序

• 在互补PWM模式同步清除计数器时输出波形控制的设定步骤例子

在互补 PWM 模式同步清除计数器时输出波形控制的设定步骤例子如图 10.58 所示。

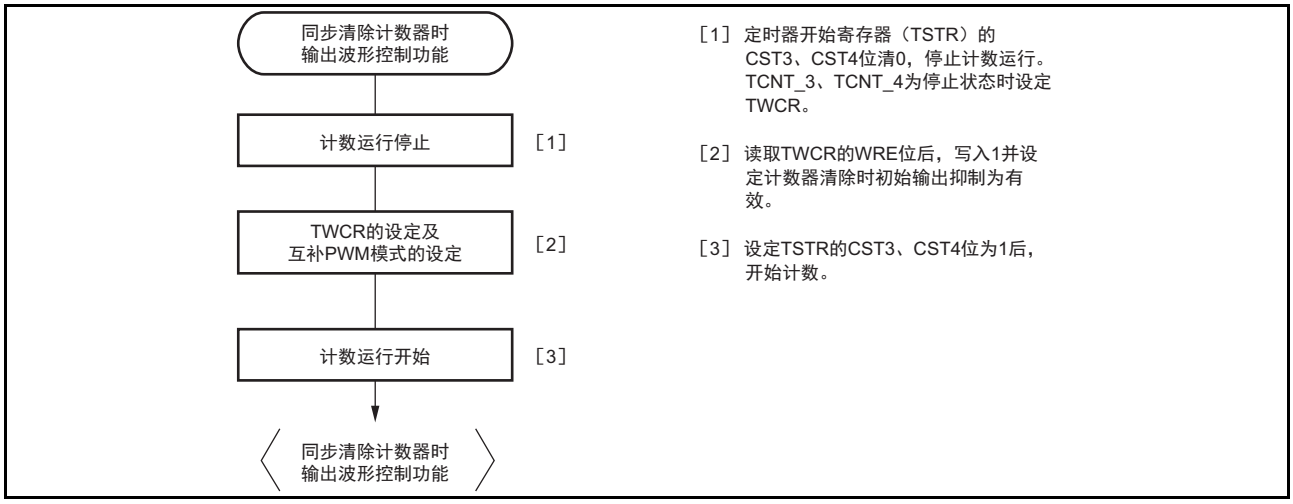


图 10.58　在互补 PWM 模式同步清除计数器时输出波形控制的设定步骤例子

• 在互补PWM模式同步清除计数器时输出波形控制的运行例子

在 TWCR 的 WRE 位设定为 1 的状态使 MTU2 互补 PWM 运行，并同步清除计数器时的运行例子如图 10.59 ～图 10.62 所示。在此，图 10.59 ～图 10.62 的计数器同步清除时序分别为图 10.57 的 （3）、（6）、（8）、（11）所示的时序。

此例相当于在 MTU2S 的 TWCR 的 SCC 位设定为 0 和 WRE 位设定为 1 的状态，使 MTU2S 互补 PWM 运行，并同步清除计数器的情况。

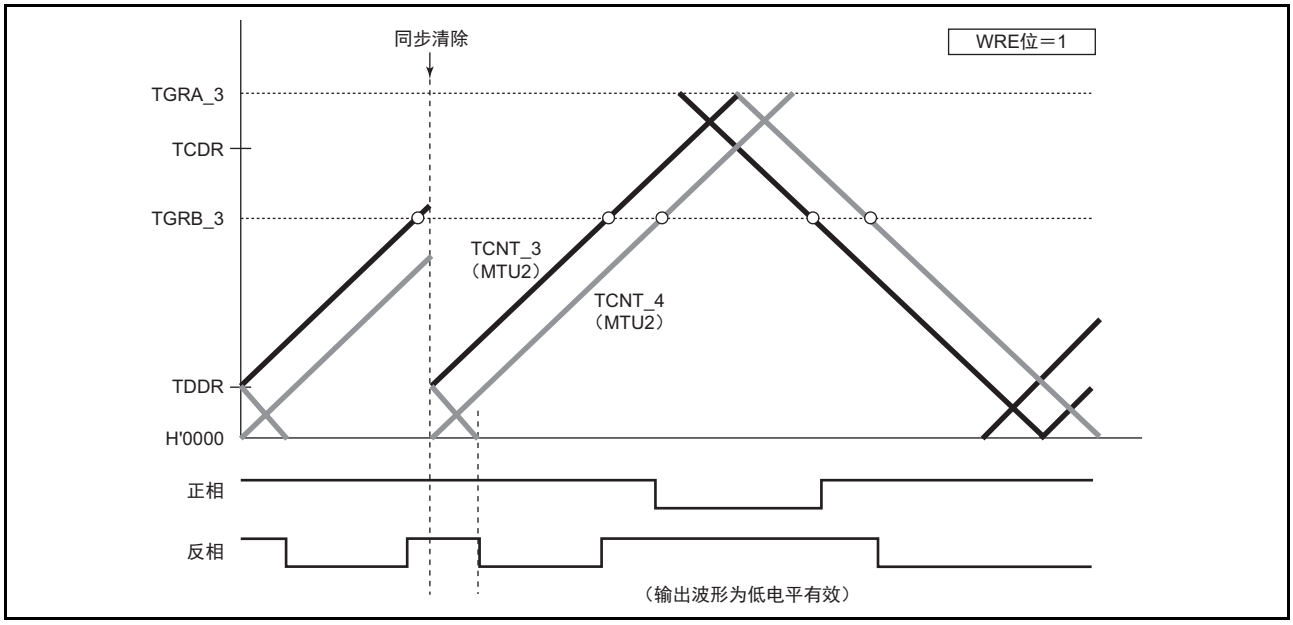


图 10.59　在递增计数的空载时间产生同步清除时
（图 10.57 的时序 （3）、 MTU2 的 TWCR 寄存器 WRE 位 =1）

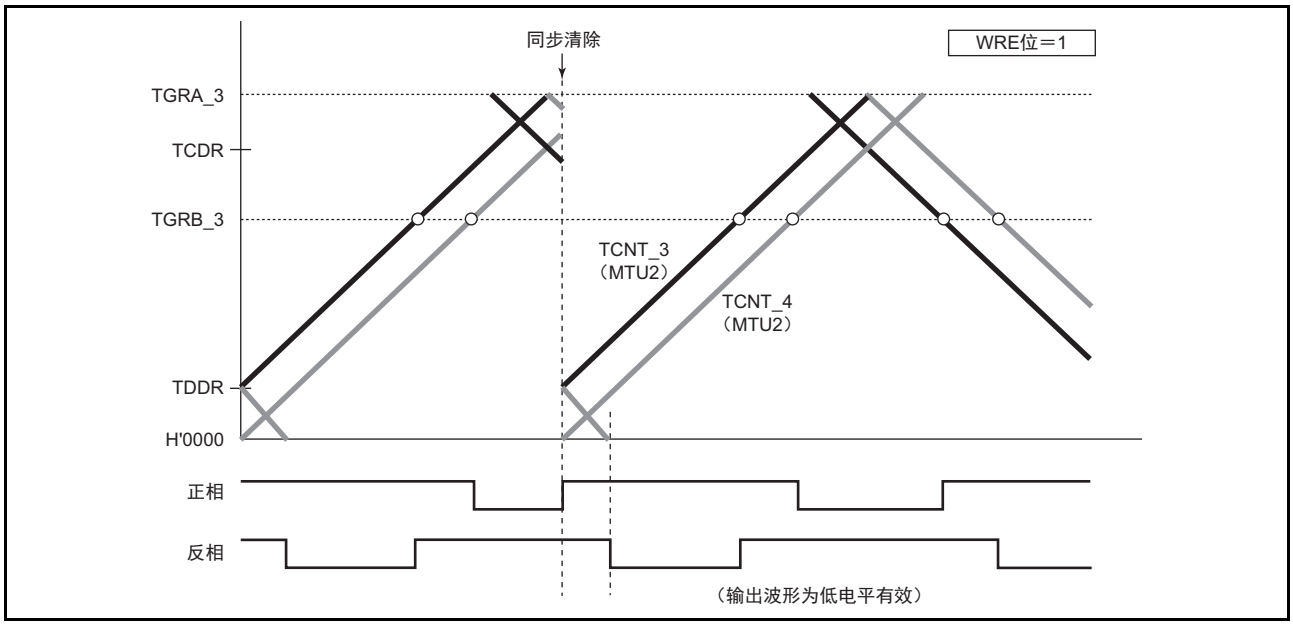


图 10.60　在波峰的 Tb 区间产生同步清除时
（图 10.57 的时序 （6）、 MTU2 的 TWCR 寄存器的 WRE 位 =1）

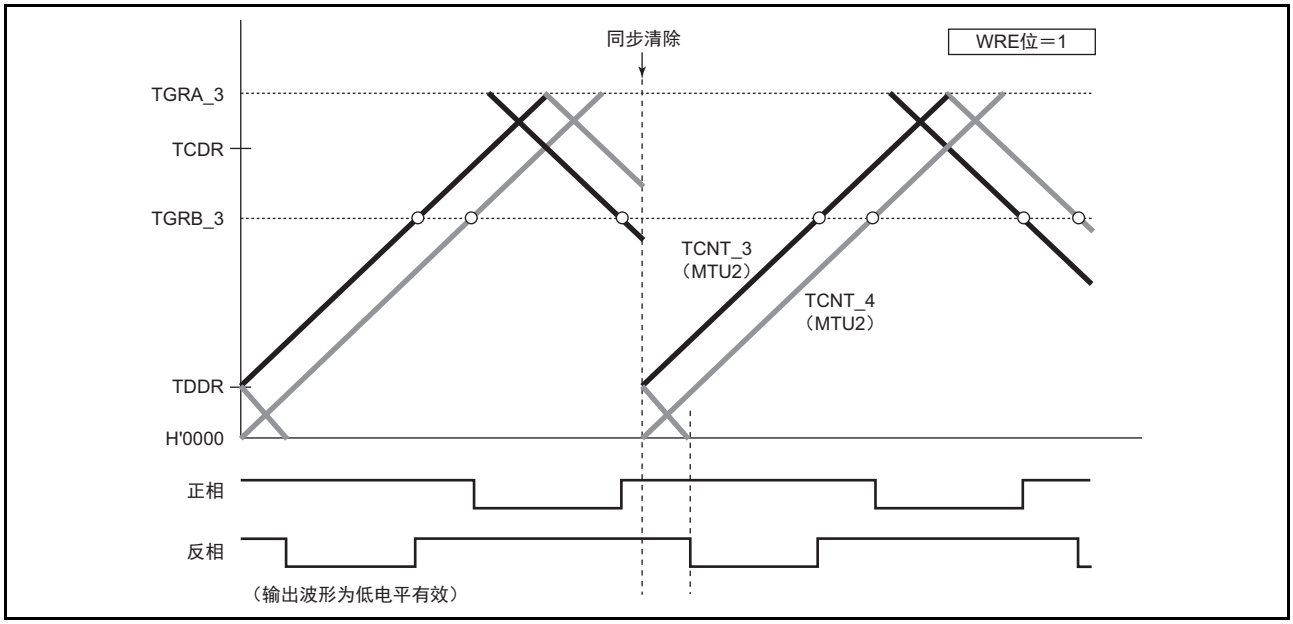


图 10.61　在递减计数的空载时间产生同步清除时
（图 10.57 的时序 （8）、 TWCR 寄存器的 WRE 位 =1）

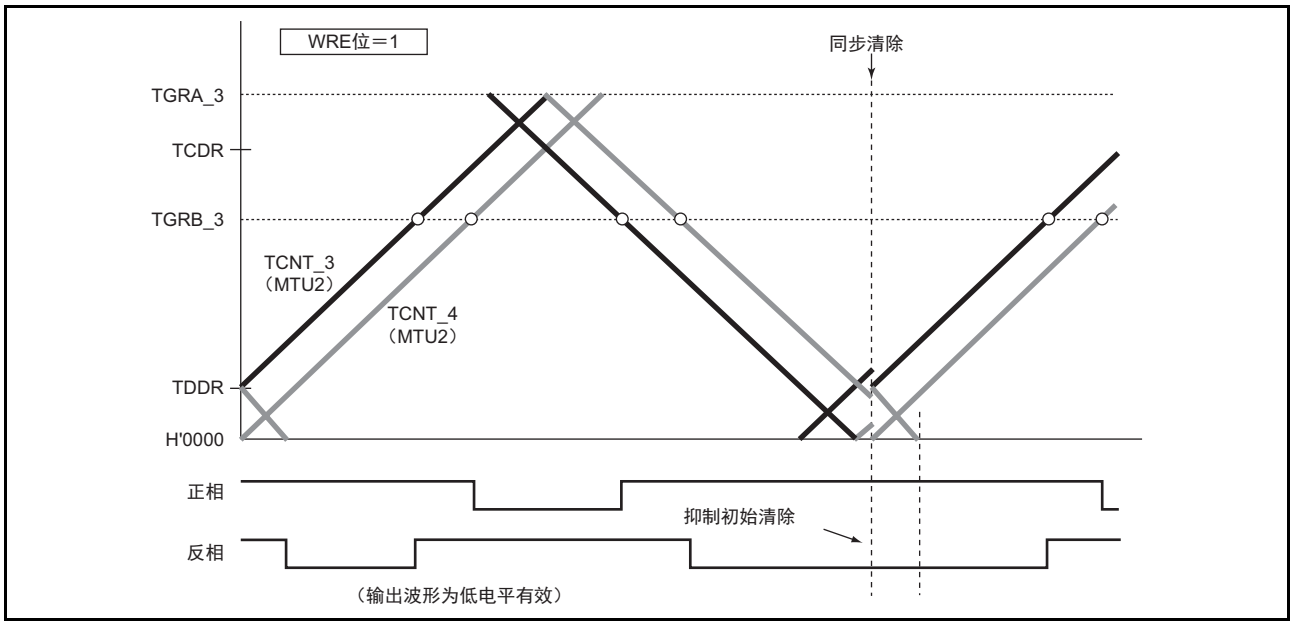


图 10.62　在波谷的 Tb 区间产生同步清除时
（图 10.57 的时序 (11)、 TWCR 寄存器的 WRE 位 =1）

(o)　MTU2 － MTU2S 计数器同步清除的抑制功能

MTU2S 中，设定 TWCR 寄存器的 SCC 位为 1，可抑制 MTU2 的同步清除。

通过设定 SCC 位，可抑制 MTU2 同步清除的区域如图 10.63 所示。

使用本功能时，必须设定 MTU2S 为互补 PWM 模式。

有关 MTU2 同步清除，详情参阅 “10.4.10(2) 使用 MTU2 标志置位源清除 MTU2S 计数器 （MTU2 － MTU2S 计数器同步清除）”。

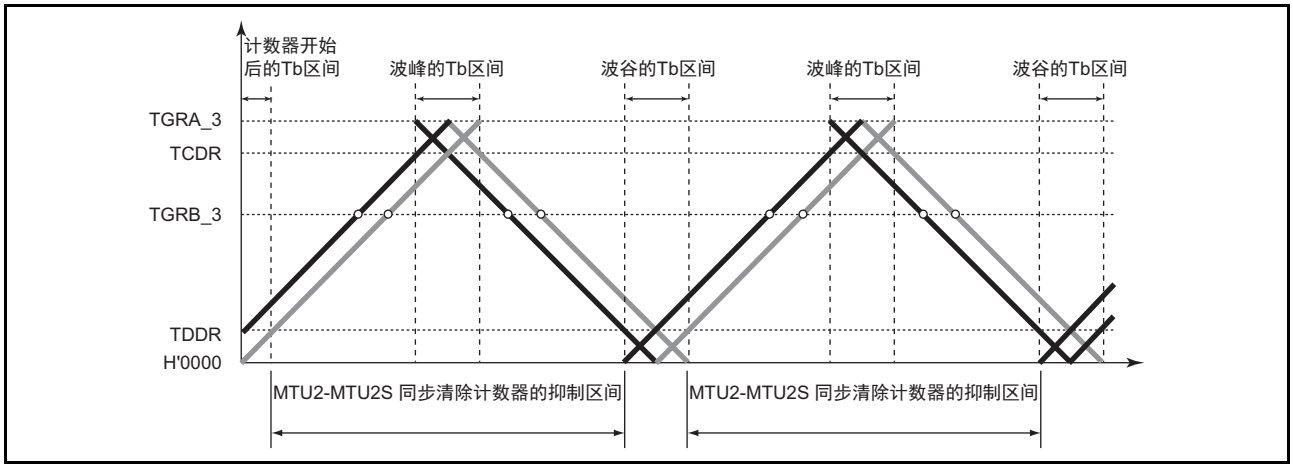


图 10.63　通过 TWCR 的 SCC 位置位，抑制 MTU2 － MTU2S 同步清除的区间

• MTU2－MTU2S计数器同步清除抑制功能的设定步骤例子

MTU2 － MTU2S 计数器同步清除抑制功能的设定步骤例子如图 10.64 所示。

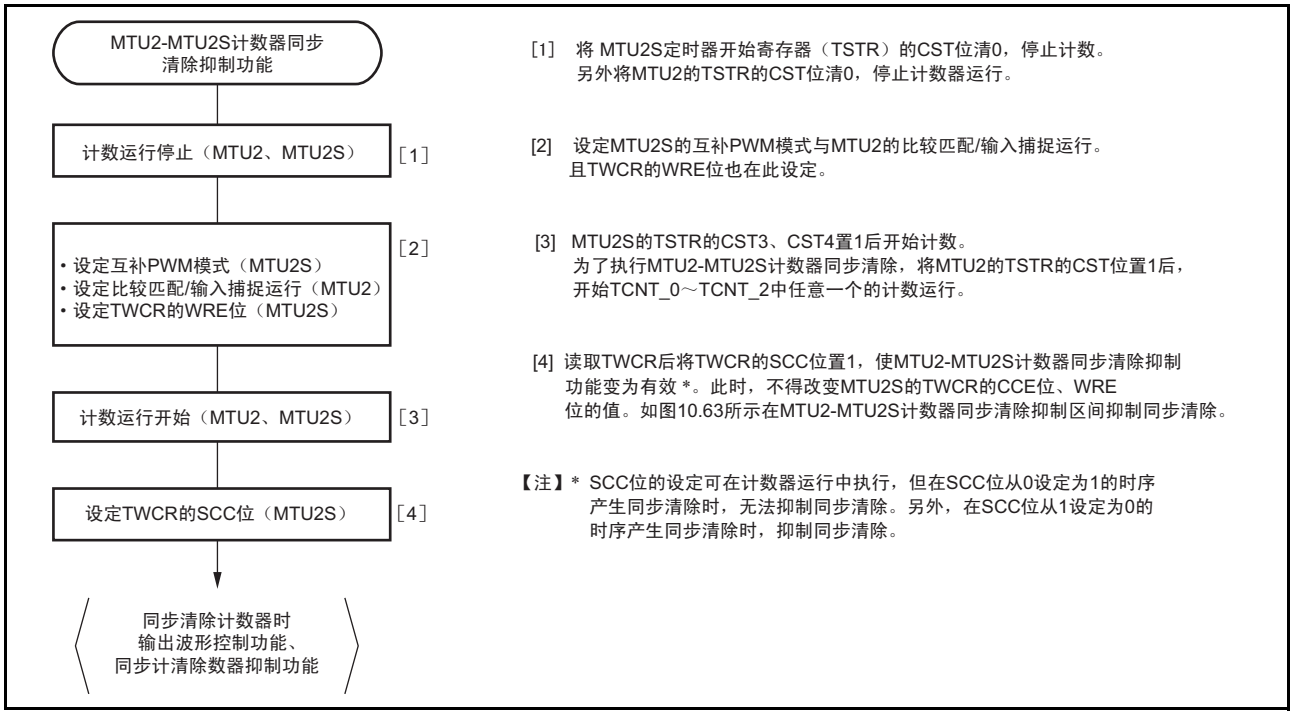


图 10.64　MTU2 － MTU2S 计数器同步清除抑制功能的设定步骤例子

• MTU2－MTU2S 计数器同步清除抑制功能的运行例子

MTU2S 的 TWCR 的 SCC 位设定为 1，并使 MTU2S 互补 PWM 运行， MTU2 － MTU2S 计数器同步清除抑制功能为有效时的运行例子如图 10.65 ～图 10.68 所示。在此，图 10.65 ～图 10.68 的计数器同步清除的时序分别如图 10.57 的 （3）、（6）、（8）、 (11) 所示。此例中，将 MTU2S 的 TWCR 的 WRE 位设定为 1。

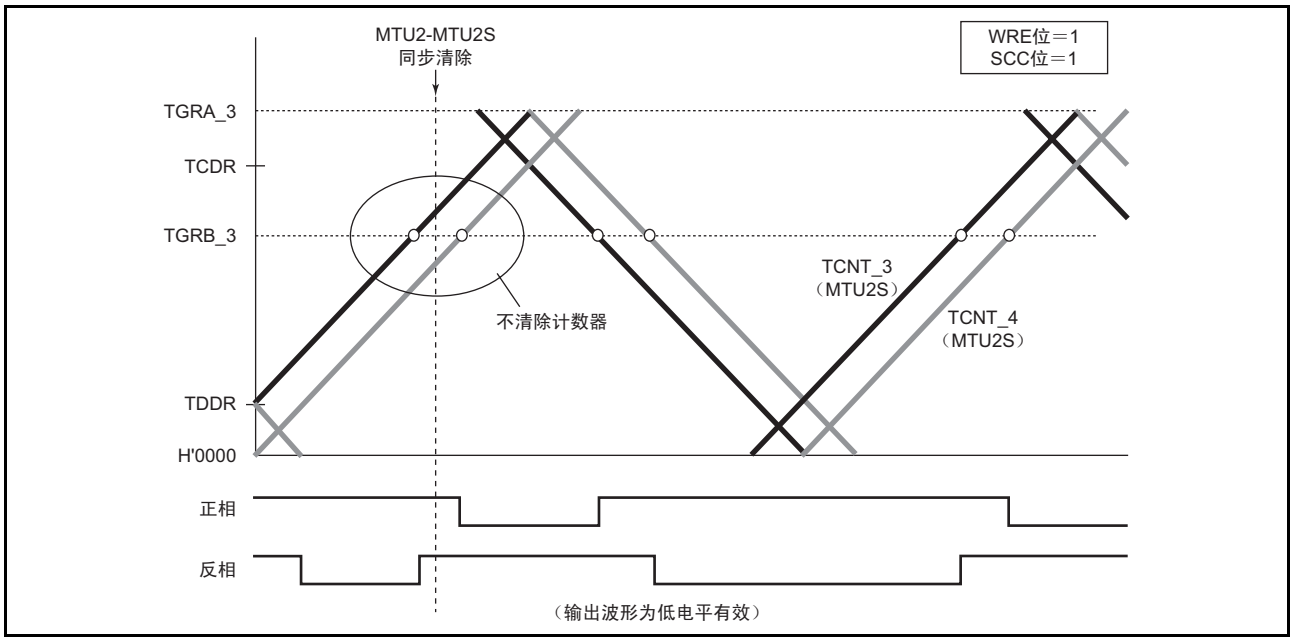


图 10.65　在递增计数的空载时间产生同步清除时
（图 10.57 的时序 （3）、 MTU2S 中 TWCR 寄存器的 WRE 位 =1、 SCC 位 =1）

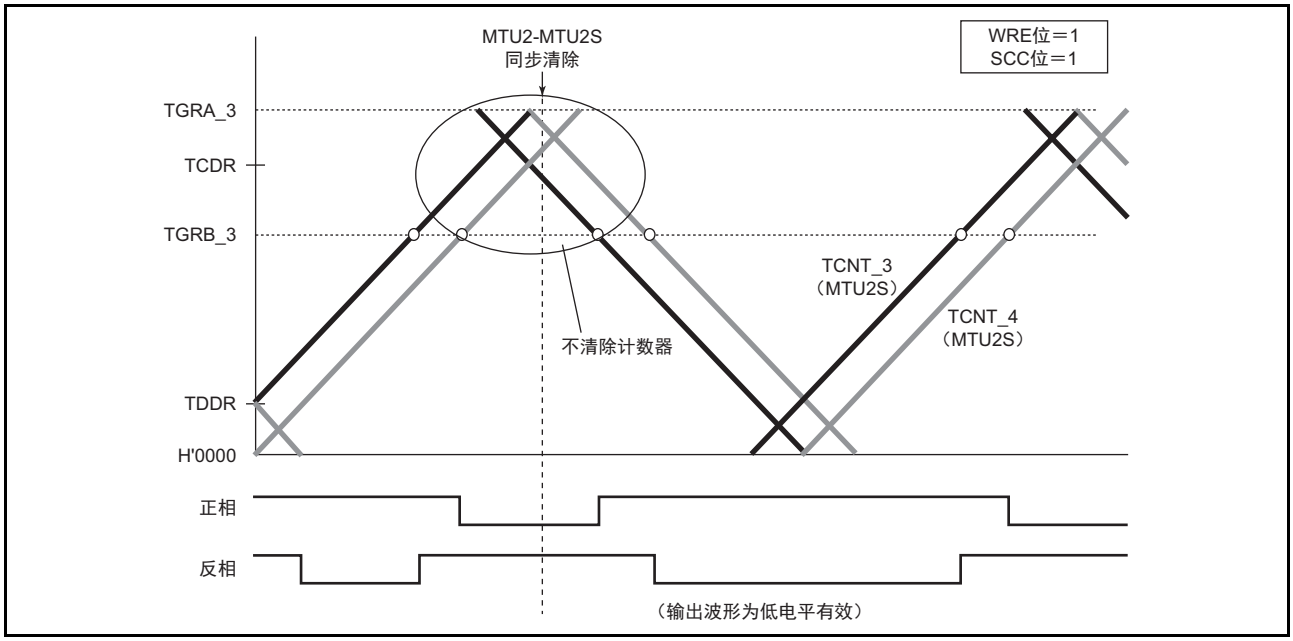


图 10.66　在波峰的 Tb 区间产生同步清除时
（图 10.57 的时序 （6）、 MTU2S 中 TWCR 寄存器的 WRE 位 =1、 SCC 位 =1）

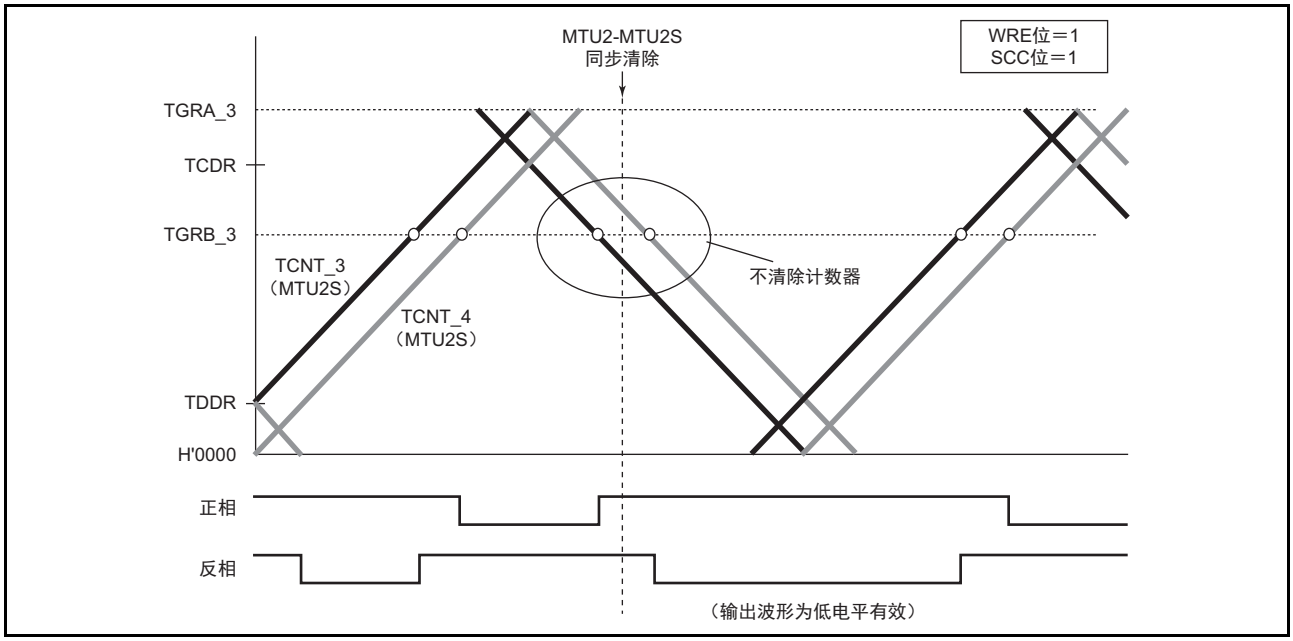


图 10.67　在递减计数的空载时间产生同步清除时
（图 10.57 的时序 （8）、 MTU2S 中 TWCR 寄存器的 WRE 位 =1、 SCC 位 =1）

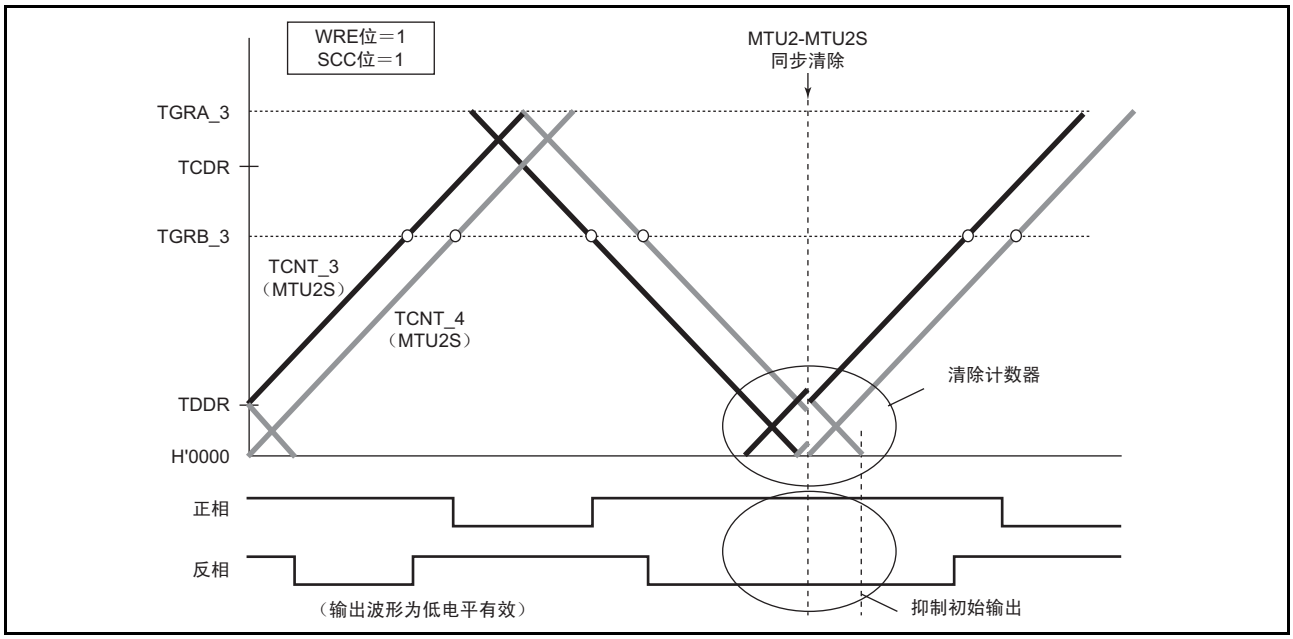


图 10.68　在波谷的 Tb 区间产生同步清除时
（图 10.57 的时序 （11）、 MTU2S 中 TWCR 寄存器的 WRE 位 =1、 SCC 位 =1）

(p)　通过 TGRA_3 的比较匹配清除计数器

在互补 PWM 模式，通过设定定时器波形控制寄存器 （TWCR）的 CCE 位，可由 TGRA_3 的比较匹配清除 TCNT_3、 TCNT_4 及 TCNTS。

其运行如图 10.69 所示。

【注】 1. 仅可在互补 PWM 模式 1 （在波峰传送）使用。
2. 不可设定为与其他通道同步清除的功能 （不可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 占空比不可设定为 H'0000。
4. 不可将定时器输出控制寄存器 1 （TOCR1）的 PSYE 位设定为 1。

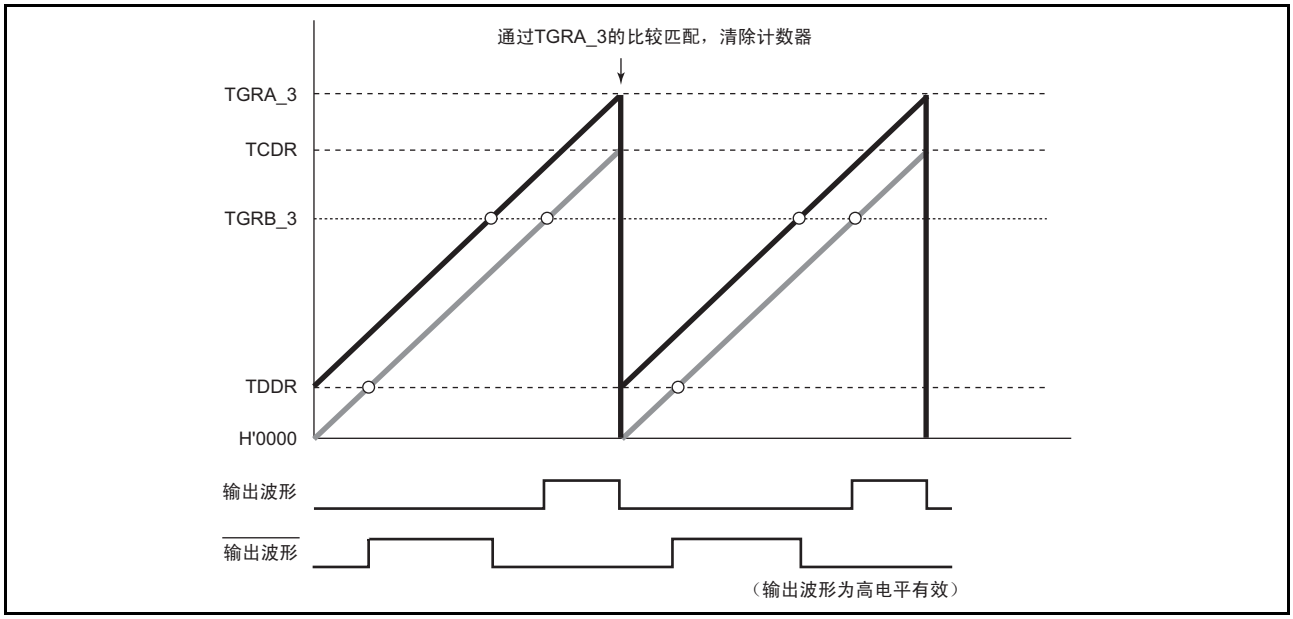


图 10.69　通过 TGRA_3 的比较匹配清除计数器的运行例子

(q)　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，使用定时器门控寄存器 （TGCR），可轻易地控制无刷 DC 马达。使用 TGCR 的无刷 DC 马达的驱动波形输出例子如图 10.70 ～图 10.73 所示。

通过使用霍尔元件等检测到的外部信号，转换 3 相无刷 DC 马达的输出相时，将 TGCR 的 FB 位设定为 0。此时，将表示磁极位置的外部信号输入到通道 0 的定时器输入引脚 TIOC0A、 TIOC0B、 TIOC0C （必须由 PFC 设定）。如果 TIOC0A、 TIOC0B、 TIOC0C 等 3 个引脚产生边沿，输出的 ON/OFF 则自动转换。

FB 位为 1 时，如果设定 TGCR 的 UF、 VF、 WF 各位为 0 或 1，则转换输出的 ON/OFF。

从互补 PWM 模式的 6 相输出引脚输出驱动波形。

将 N 位或 P 位设定为 1，在 ON 输出时，此 6 相输出使用互补 PWM 模式可输出斩波。 N 位或 P 位为 0 时，为电平输出。

6 相输出的有效电平 （ON 输出时的电平）与 N 位及 P 位的设定无关，可由定时器输出控制寄存器 （TOCR）的 OLSN 位、 OLSP 位设定。

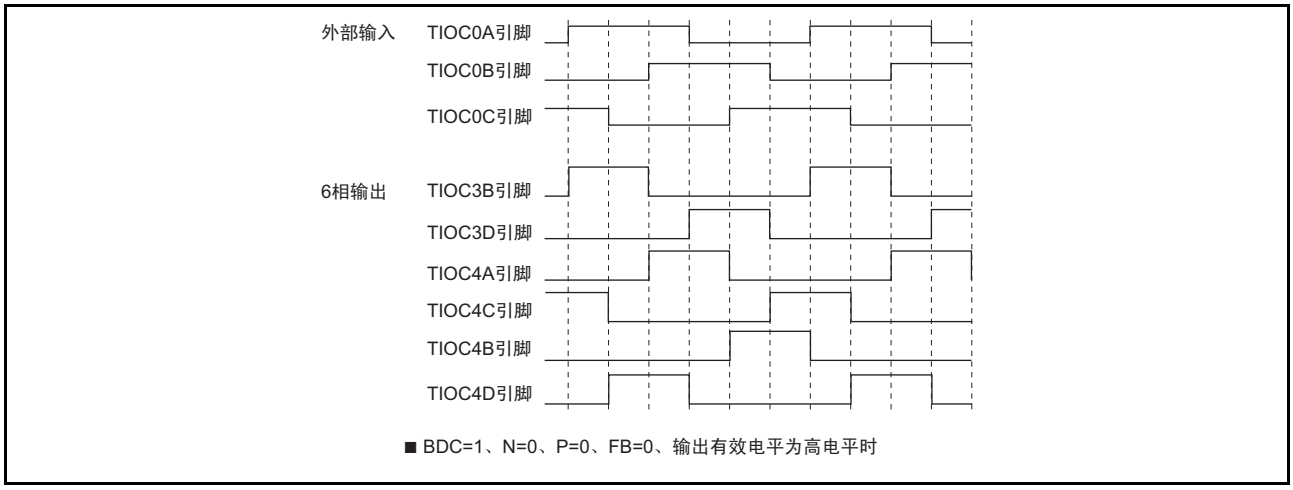


图 10.70　由外部输入转换输出相的例子 （1）

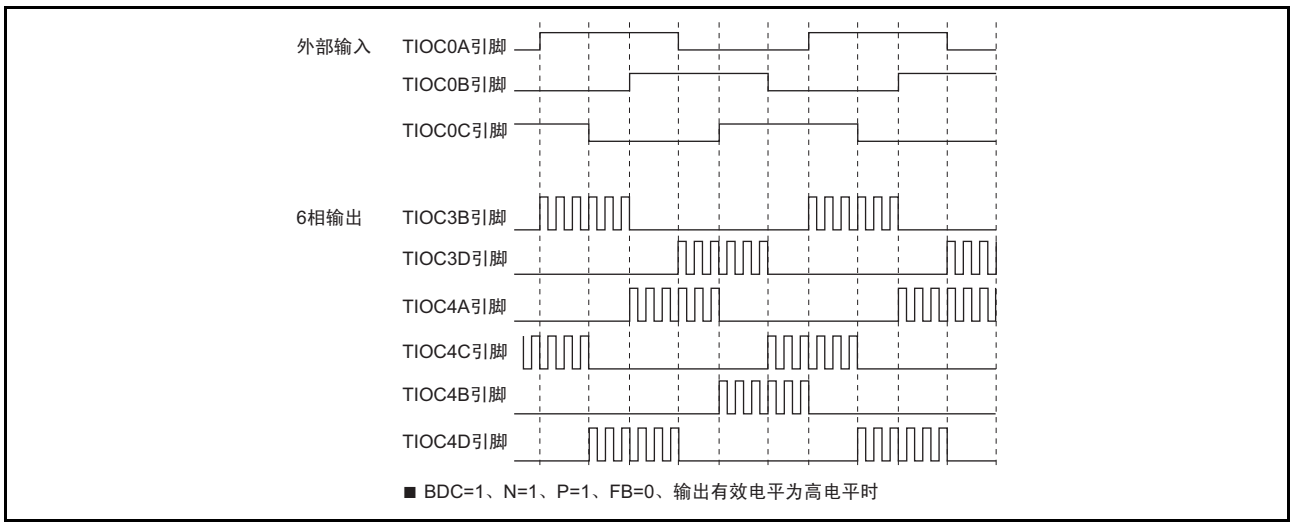


图 10.71　由外部输入转换输出相的例子 （2）

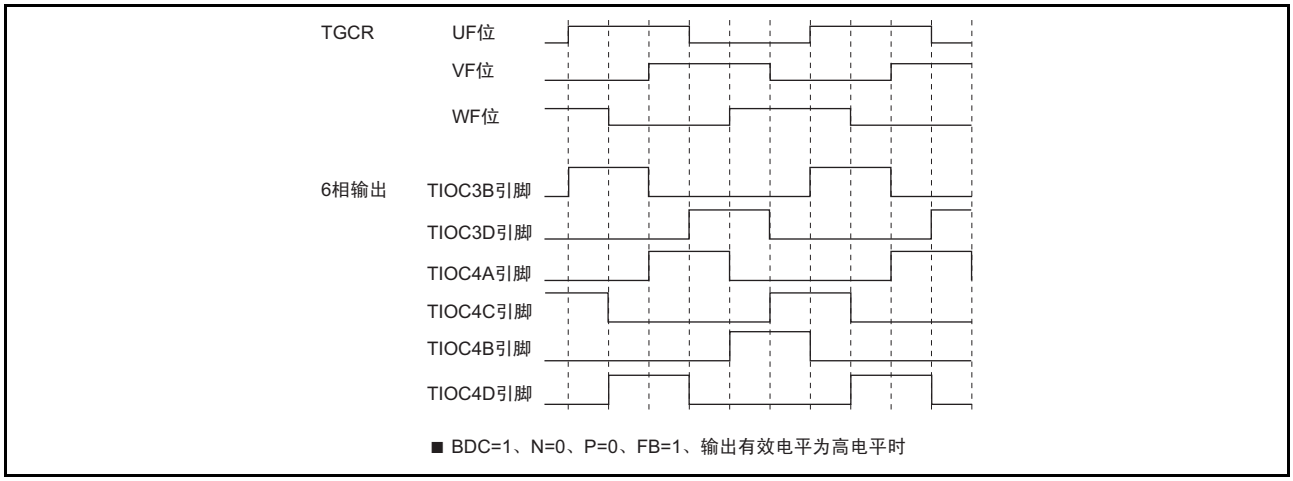


图 10.72　通过设定 UF、 VF、 WF 位转换输出相的例子 （1）

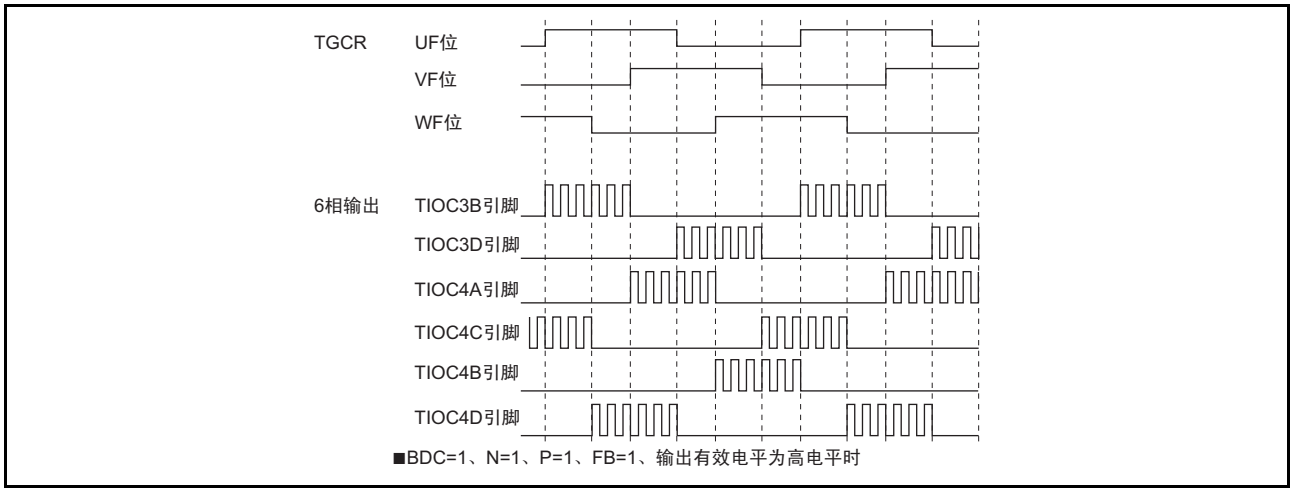


图 10.73　通过设定 UF、 VF、 WF 位转换输出相的例子 （2）

(r)　A/D 转换开始请求的设定

互补 PWM 模式时，可使用 TGRA_3 的比较匹配、 TCNT_4 的下溢 （波谷）、除通道 3、 4 之外的通道比较匹配，执行 A/D 转换开始请求。

使用 TGRA_3 的比较匹配设定开始请求时，可在 TCNT_3 的波峰开始 A/D 转换。

定时器中断允许寄存器 （TIER）的 TTGE 位置 1，可设定 A/D 转换的开始请求； TIER_4 的 TTGE2 位置 1，可设定 TCNT_4 的下溢 （波谷）的 A/D 转换开始请求。

(3)　互补 PWM 模式的中断跳过功能

通过设定定时器中断跳过设定寄存器 （TITCR），通道 3 和 4 的 TGIA_3 （波峰的中断）及 TCIV_4 （波谷的中断）最多可跳过 7 次中断。

设定定时器缓冲传送寄存器 （TBTER），可联动从缓冲寄存器到暂存器 / 比较寄存器的传送，并跳过中断。与缓冲寄存器的联动，详情参阅 “(c) 与中断跳过联动的缓冲传送控制”。

设定定时器 A/D 转换请求控制寄存器 （TADCR），可与 A/D 转换开始请求延迟功能的 A/D 转换开始请求联动，并跳过中断。与 A/D 转换开始请求延迟功能的联动，详情参阅 “10.4.9 A/D 转换开始请求延迟功能”。

必须通过设定 TIER_3、 TIER_4 寄存器，在禁止 TGIA_3 和 TCIV_4 中断请求的状态，以及在不通过比较匹配产生 TGFA_3、 TCFV_4 标志置位的状态，设定定时器中断跳过设定寄存器 （TITCR）。另外，更改跳过次数前，必须将 T3AEN 和 T4VEN 位设定为 0，清除跳过计数器。

(a)　中断跳过功能的设定步骤

中断跳过功能的设定步骤例子和中断跳过次数的可更改期间分别如图 10.74 和图 10.75 所示。

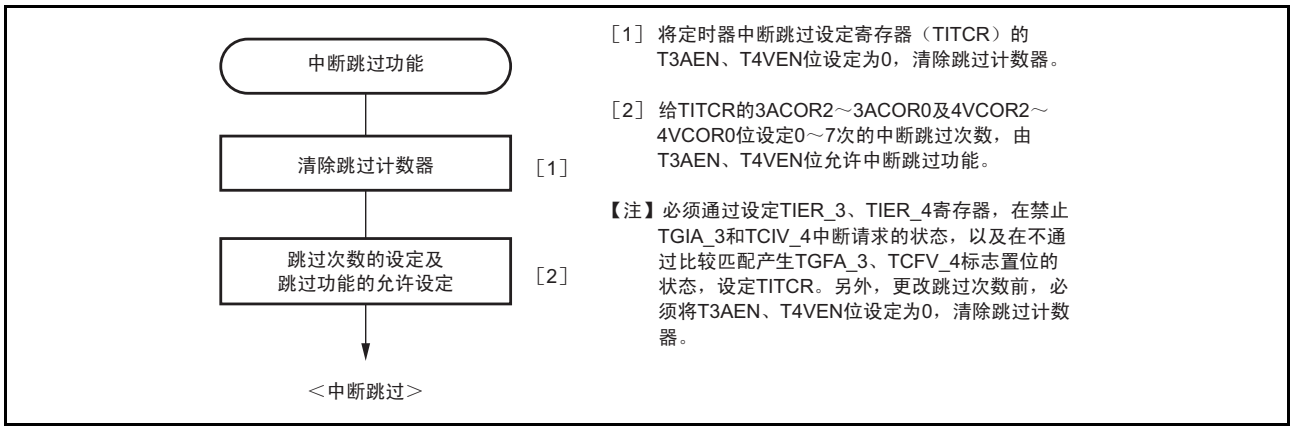


图 10.74　中断跳过功能的设定步骤例子

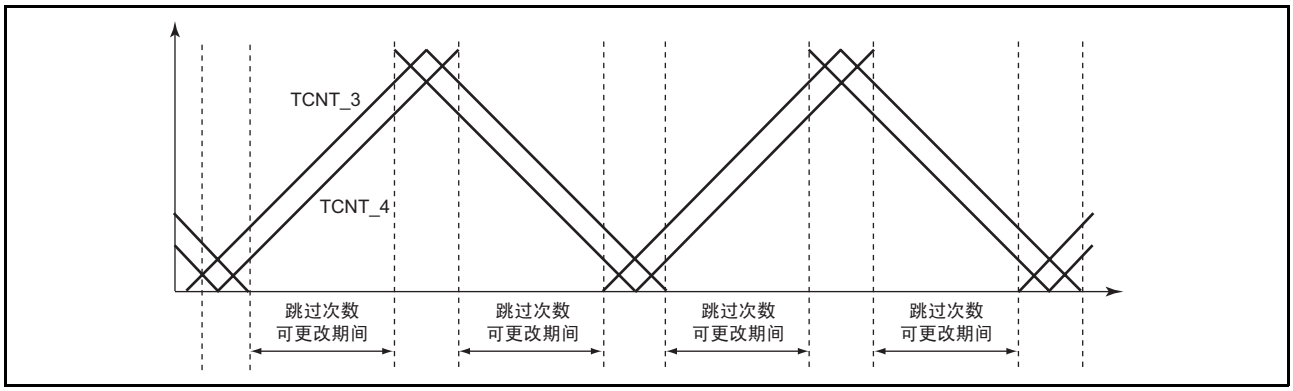


图 10.75　中断跳过次数的可更改期间

(b)　中断跳过功能的运行例子

通过定时器中断跳过设定寄存器 （TITCR）的 3ACOR 位将中断跳过次数设定为 3 次，并将 T3AEN 位设定为 1 时的 TGIA_3 的中断跳过功能运行例子如图 10.76 所示。

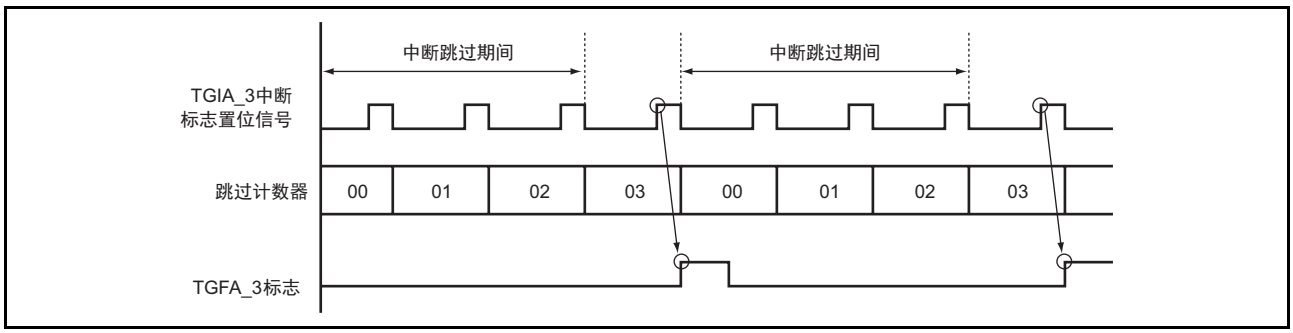


图 10.76　中断跳过功能运行例子

(c)　与中断跳过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 模式是否从缓冲寄存器向暂存器缓冲传送，或是否与中断跳过联动。

设定为抑制缓冲传送 （BTE1=0、 BTE0=1）的运行例子如图 10.77 所示。在此设定期间，不从缓冲寄存器向暂存器传送。

设定为缓冲传送与中断跳过功能 （BTE1=1、 BTE0=0）联动的运行例子如图 10.78 所示。此设定在缓冲传送允许期间之外，不从缓冲寄存器向暂存器传送。

另外，如果将定时器中断跳过设定寄存器 （TITCR）的 T3AEN 位、 T4VEN 位或 T3AEN/T4VEN 位置 1，各缓冲传送允许期间也不同。 TITCR 的 T3AEN、 T4VEN 位的设定与缓冲允许期间的关系如图 10.79 所示。

【注】 本功能必须与中断跳过功能组合使用。

禁止中断跳过 （定时器中断跳过设定寄存器 （TITCR）的 T3AEN、 T4VEN 位设为 0，或 TITCR 的跳过次数设定位 （3ACOR、 4VCOR）设为 0）时，必须设定为缓冲传送不与中断跳过联动 （将定时器缓冲传送设定寄存器（TBTER）的 BTE1 设定为 0）。

禁止中断跳过时，如果设定缓冲传送与中断跳过联动，则不缓冲传送。

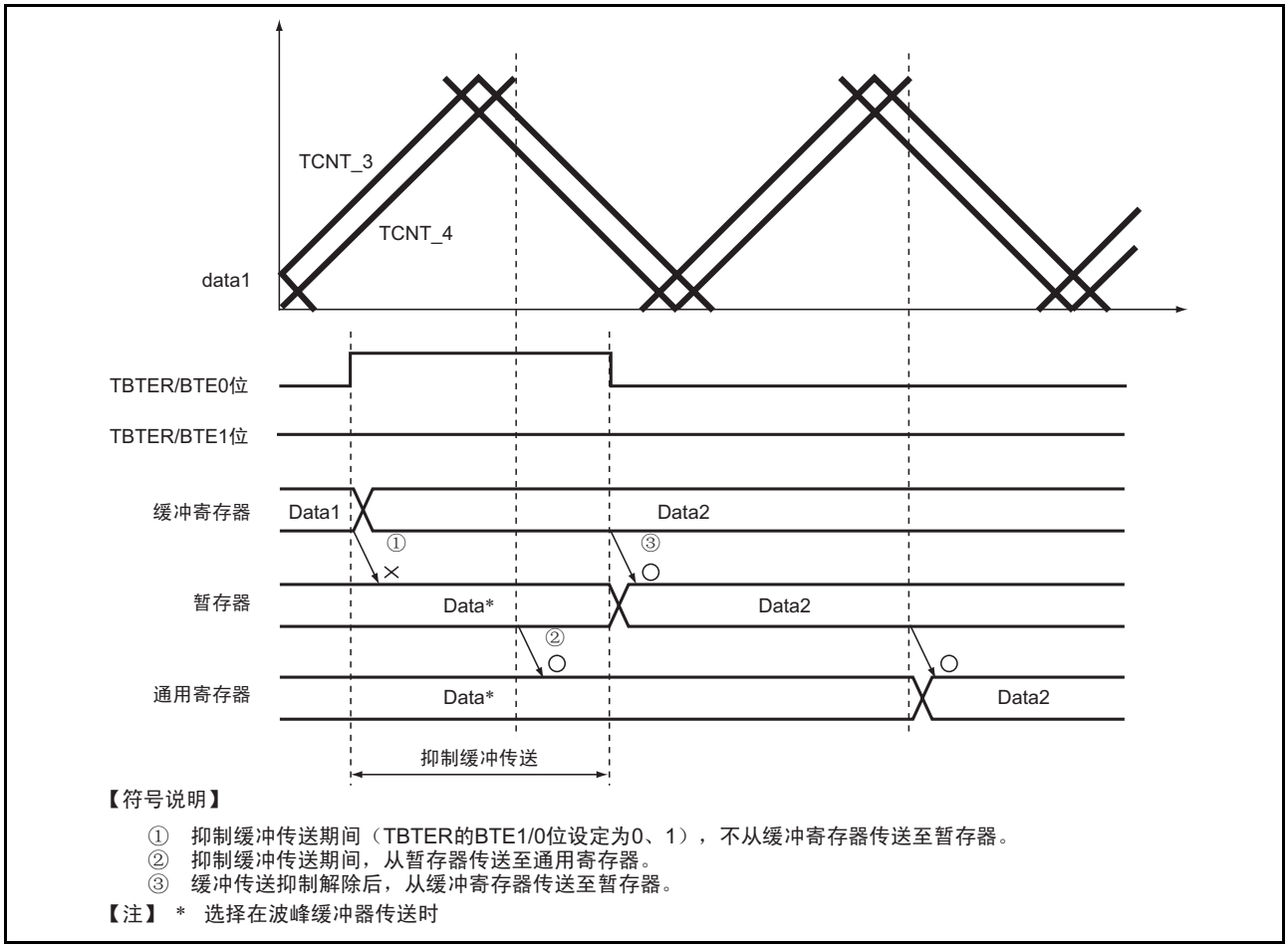


图 10.77　设定为抑制缓冲传送时的运行例子 （BTE1=0、 BTE0=1）

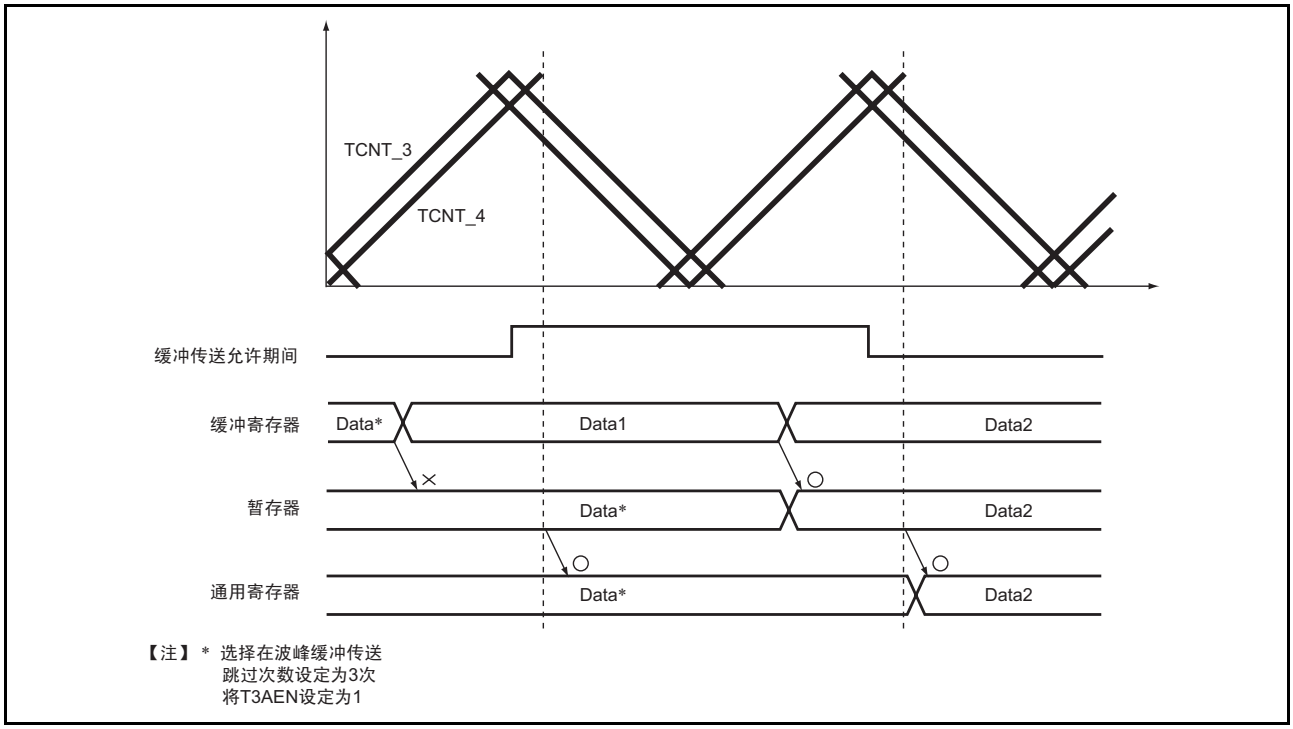


图 10.78　设定为缓冲传送与中断跳过联动时的运行例子 （BTE1=1、 BTE0=0）

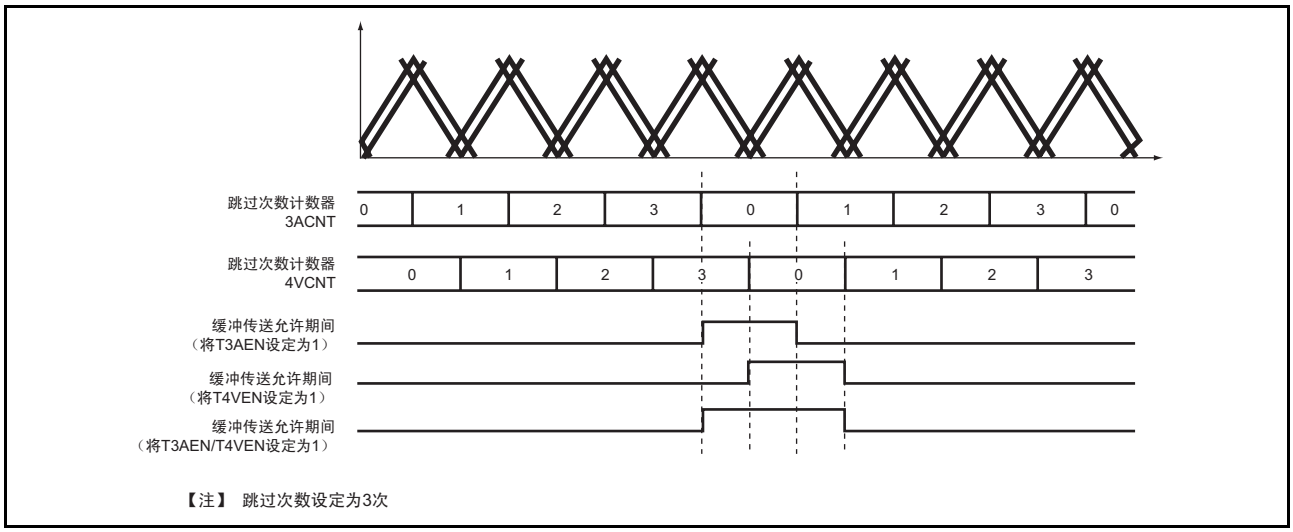


图 10.79　定时器中断跳过设定寄存器 （TITCR）的 T3AEN、 T4VEN 位的设定和缓冲传送允许期间的关系

(4)　互补 PWM 模式的输出保护功能

互补 PWM 模式的输出具有下列保护功能：

(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、计数器中，除总是改写的缓冲寄存器之外，模式寄存器、控制寄存器、比较寄存器及计数器，均可通过定时器读取写入允许寄存器 （TRWER）的 RWE 位的设定，选择允许 / 禁止从 CPU 存取。其对象为通道 3 和 4 的一部分寄存器，如下所示：

TCR_3 和 TCR_4、 TMDR_3 和 TMDR_4、 TIORH_3 和 TIORH_4、 TIORL_3 和 TIORL_4、
TIER_3 和 TIER_4、 TCNT_3 和 TCNT_4、 TGRA_3 和 TGRA_4、 TGRB_3 和 TGRB_4、
TOER、 TOCR、 TGCR、 TCDR、 TDDR　共计 21 个寄存器

此功能通过将模式寄存器、控制寄存器和计数器设定为禁止从 CPU 存取，可防止因 CPU 失控产生的误写。在禁止存取状态，对象寄存器的读取值不定且写入无效。

(b)　由外部信号停止 PWM 输出的功能

输入指定的外部信号，可将 6 相 PWM 输出引脚自动设定为高阻抗状态。

详情参阅 “第 11 章　端口输出允许 （POE）”。

(c)　振荡停止时的 PWM 输出停止功能

检测出输入到本 LSI 的时钟停止时， 6 相 PWM 输出引脚自动变为高阻抗状态。但是，如果时钟重新开始振荡，则无法保证引脚的状态。

详情参阅 “4.7　振荡停止检测功能”。

10.4.9　A/D 转换开始请求延迟功能

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 启动请求的周期寄存器（TADCORA_4、 TADCORB_4）和周期缓冲寄存器 （TADCOBRA_4、 TADCOBRB_4），可请求开始 A/D 转换。

A/D 转换开始请求延迟功能是比较 TCNT_4 与 TADCORA_4、 TADCORB_4，如果匹配时，执行各自的 A/D 转换开始请求 （TRG4AN、 TRG4BN）。

另外，设定 TADCR 的 ITA3AE、 ITA4VE、 ITB3AE、 ITB4VE 位，可与中断跳过功能联动，跳过 A/D 转换的开始请求 （TRG4AN、 TRG4BN）。

(a)　A/D 转换开始请求延迟功能的设定步骤例子

A/D 转换开始请求延迟功能的设定步骤例子如图 10.80 所示。

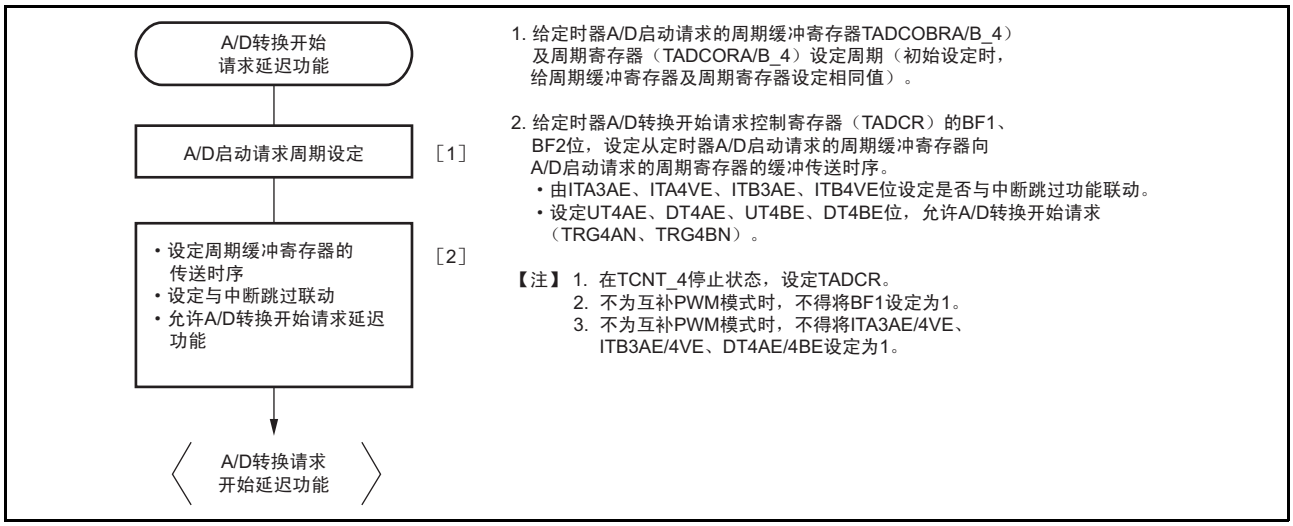


图 10.80　A/D 转换开始请求延迟功能的设定步骤例子

(b)　A/D 转换开始请求延迟功能的基本运行例子

在 TCNT_4 的波谷设定缓冲传送时序，且设定为在 TCNT_4 递减计数时输出 A/D 转换开始请求信号（TRG4AN）时， A/D 转换开始请求信号 （TRG4AN）的基本运行例子如图 10.81 所示。

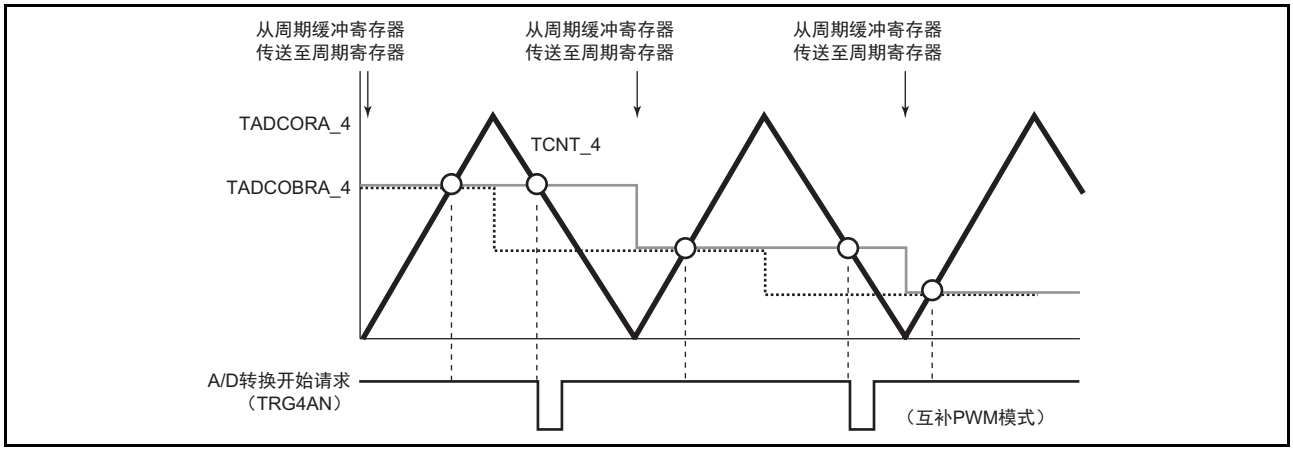


图 10.81　A/D 转换开始请求信号 （TRG4AN）的基本运行例子

(c)　缓冲传送

将数据写入定时器 A/D 启动请求的周期设定缓冲寄存器 （TADCOBRA/B_4），可更新定时器 A/D 启动请求的周期设定寄存器 （TADCORA/B_4）的数据。设定定时器 A/D 转换开始请求控制寄存器 （TADCR_4）的 BF1、 BF0 位，可选择从定时器 A/D 启动请求的周期设定缓冲寄存器向定时器 A/D 启动请求的周期设定寄存器传送的时序。

(d)　与中断跳过功能联动的 A/D 转换开始请求延迟功能

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、ITB4VE 位，可与中断跳过功能联动，执行 A/D 转换开始请求（TRG4AN、TRG4BN）。设定在 TCNT_4 递增计数时及递减计数时允许 TRG4AN 输出，与中断跳过功能联动时的 A/D 转换开始请求信号（TRG4AN）的运行例子如图 10.82 所示。

设定在 TCNT_4 递增计数时允许 TRG4AN 输出，与中断跳过功能联动时的 A/D 转换开始请求信号（TRG4AN）的运行例子如图 10.83 所示。

【注】 本功能必须和中断跳过功能组合使用。
禁止中断跳过时（将定时器中断跳过设定寄存器（TITCR）的 T3AEN、T4VEN 位设为 0 或将 TITCR 的跳过次数设定位（3ACOR、4VCOR）设为 0 时），必须设定为不与中断跳过功能联动（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）。

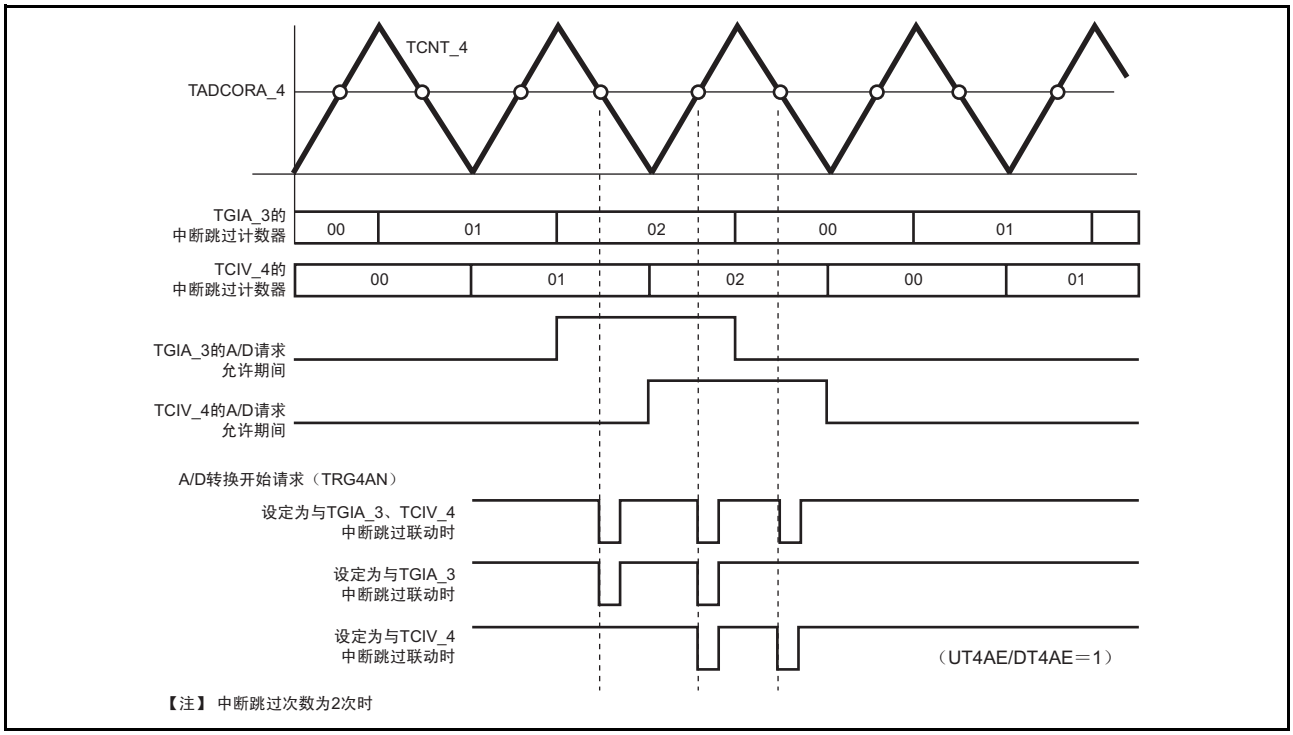


图 10.82　与中断跳过功能联动时的 A/D 转换开始请求信号（TRG4AN）的运行例子

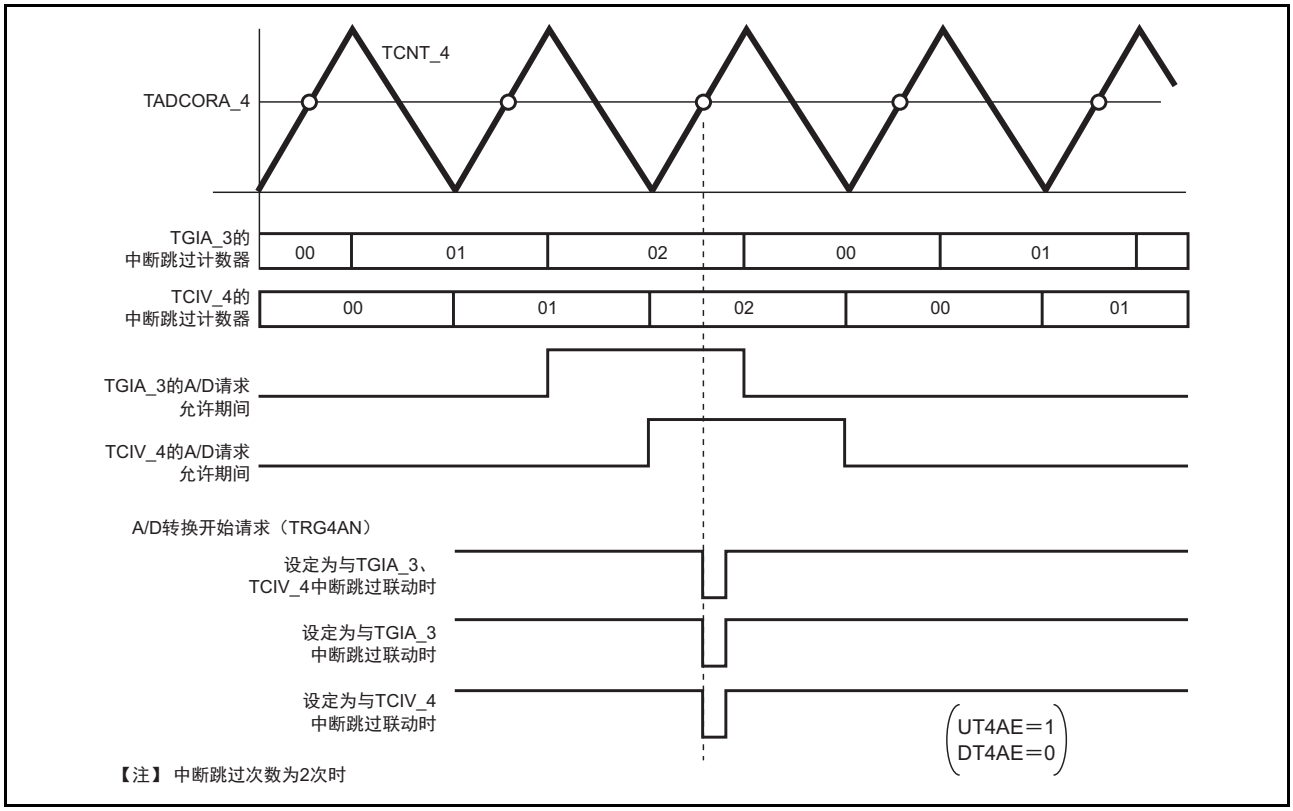


图 10.83　与中断跳过功能联动时的 A/D 转换开始请求信号 （TRG4AN）的运行例子

10.4.10　MTU2 － MTU2S 的同步运行

(1)　MTU2 － MTU2S 计数器同步开始

设定 MTU2 的 TCSYSTR 寄存器，可同步开始以不同时钟运行的 MTU2 和 MTU2S 的计数器。

(a)　MTU2 － MTU2S 计数器同步开始的设定步骤例子

计数器同步开始的设定步骤例子如图 10.84 所示。

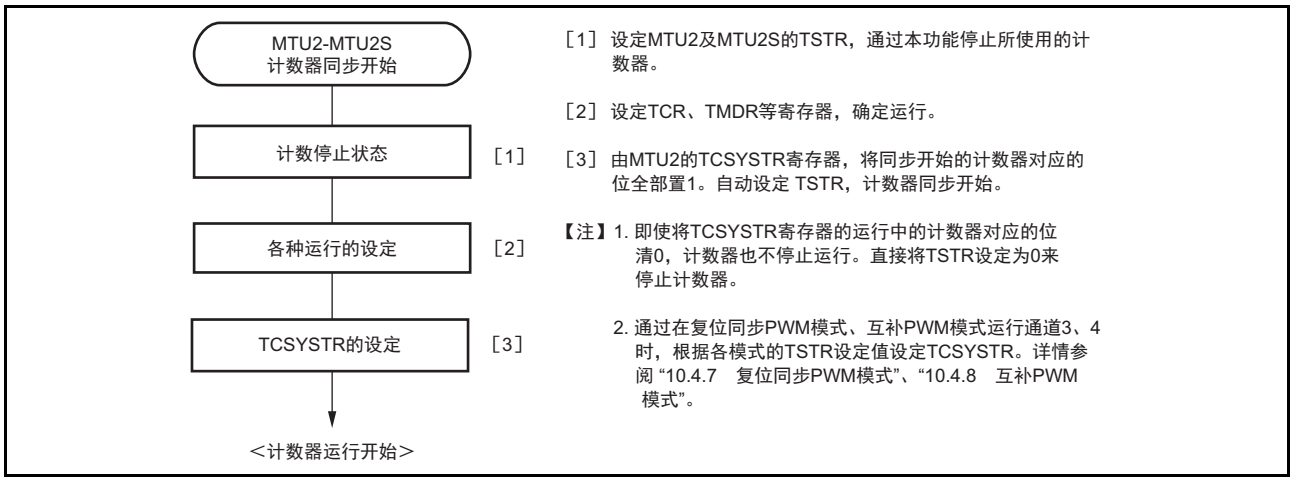


图 10.84　计数器同步开始设定步骤例子

(b)　计数器同步开始运行例子

MTU2 和 MTU2S 的时钟频率比为 1:1、 1:2、 1:3、 1:4 时的计数器同步开始运行例子分别如图 10.85(1)、图 10.85(2)、图 10.85(3)、图 10.85(4) 所示。

这些例子中，计数器时钟设定为 MPϕ/1。

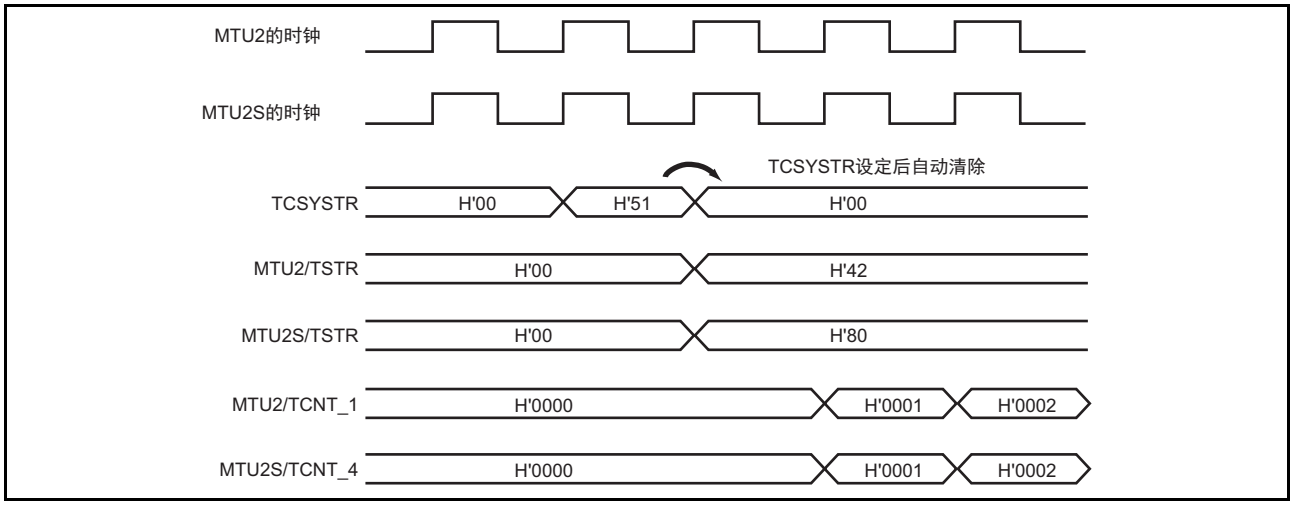


图 10.85(1)　计数器同步开始运行例子 （MTU2 和 MTU2S 的时钟频率比为 1:1）

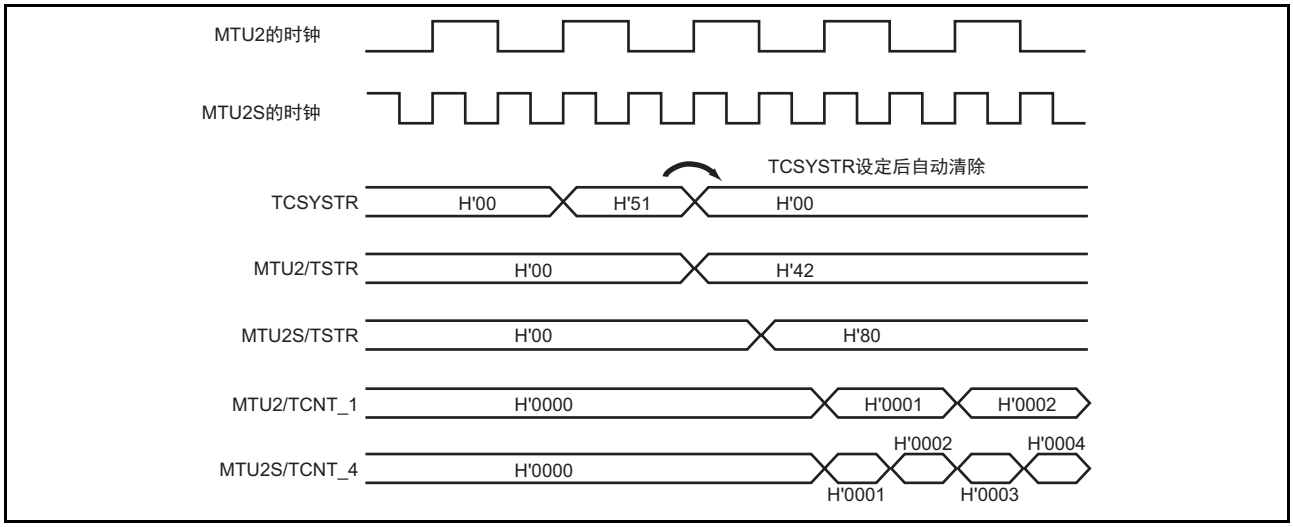


图 10.85(2)　计数器同步开始运行例子 （MTU2 和 MTU2S 的时钟频率比为 1:2）

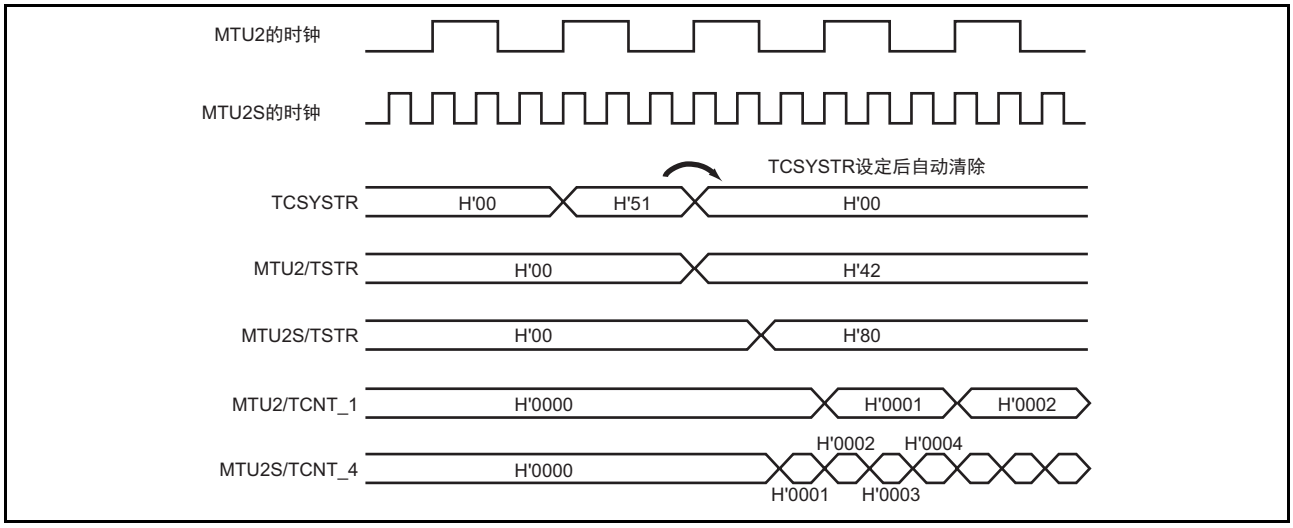


图 10.85(3)　计数器同步开始运行 （MTU2 和 MTU2S 的时钟频率比为 1:3）

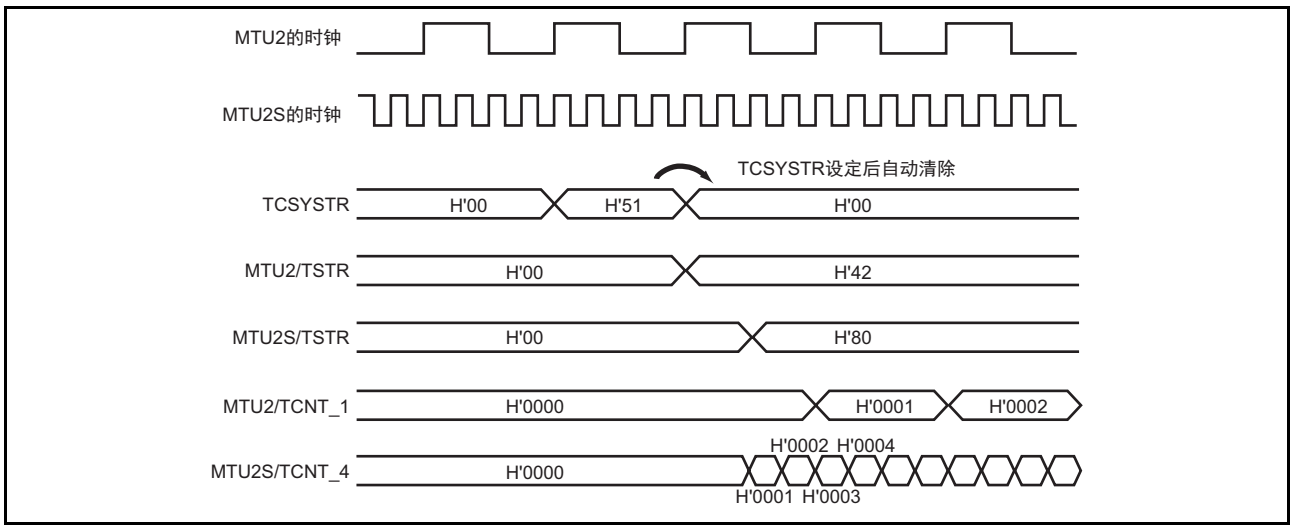


图 10.85(4)　计数器同步开始运行 （MTU2 和 MTU2S 的时钟频率比为 1:4）

(2)　使用 MTU2 标志置位源清除 MTU2S 计数器 （MTU2 － MTU2S 计数器同步清除）

通过设定 TSYCR_3 寄存器，使用 MTU2 的 TSR_0 ～ TSR_2 的标志置位源， MTU2S 可清除计数器。

(a)　使用 MTU2 标志置位源清除 MTU2S 计数器的设定步骤例子

使用 MTU2 标志置位源清除 MTU2S 计数器的设定步骤例子如图 10.86 所示。

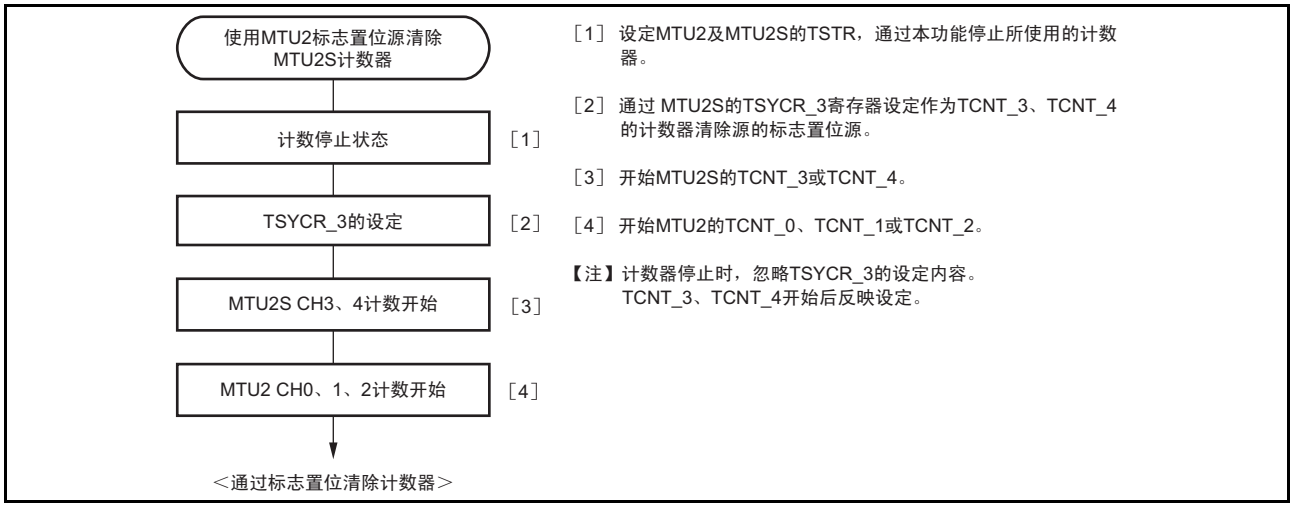


图 10.86　使用 MTU2 标志置位源清除 MTU2S 计数器的设定步骤例子

(b)　使用 MTU2 标志置位源清除 MTU2S 计数器

使用 MTU2 标志置位源清除 MTU2S 计数器的运行例子如图 10.87(1) 和图 10.87(2) 所示。

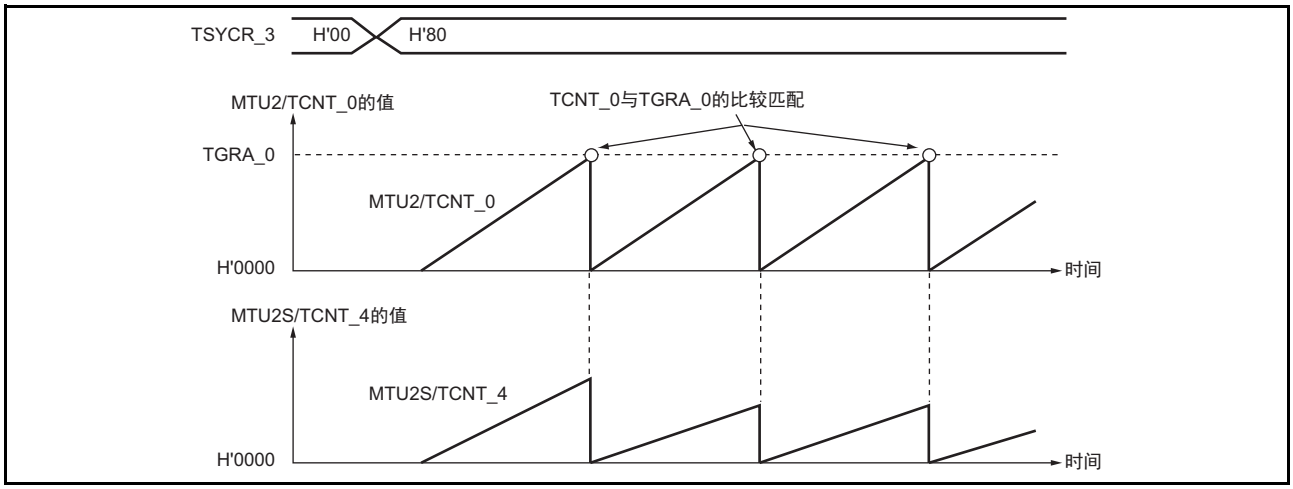


图 10.87(1)　使用 MTU2 标志置位源清除 MTU2S 计数器的运行例子 （1）

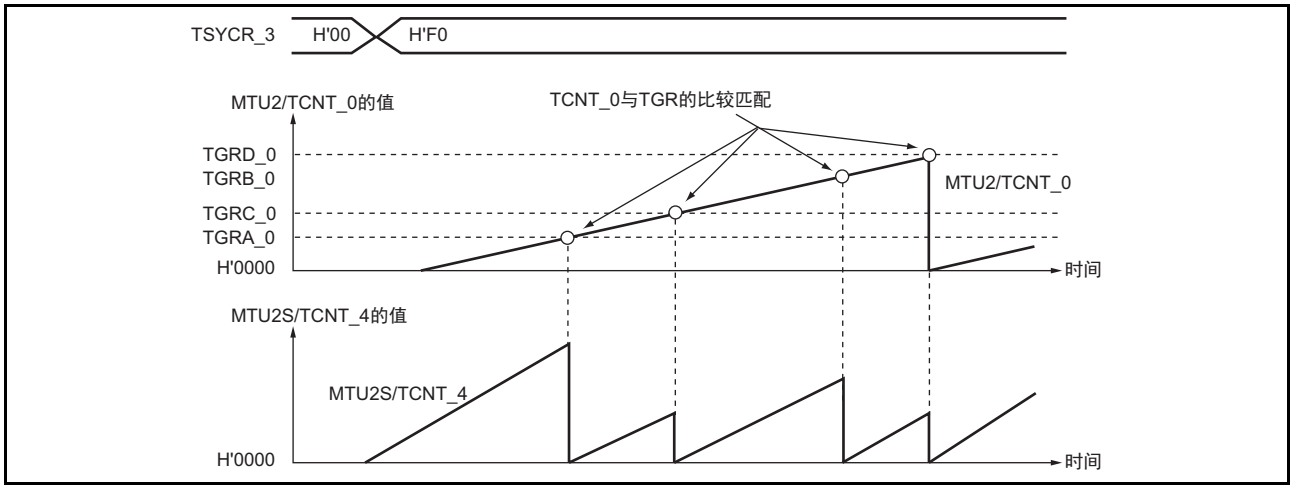


图 10.87(2)　使用 MTU2 标志置位源清除 MTU2S 计数器的运行例子 （2）

10.4.11　外部脉宽测量功能

通道 5 最多可测量 3 个外部脉宽。

(1)　测量外部脉宽的设定步骤例子

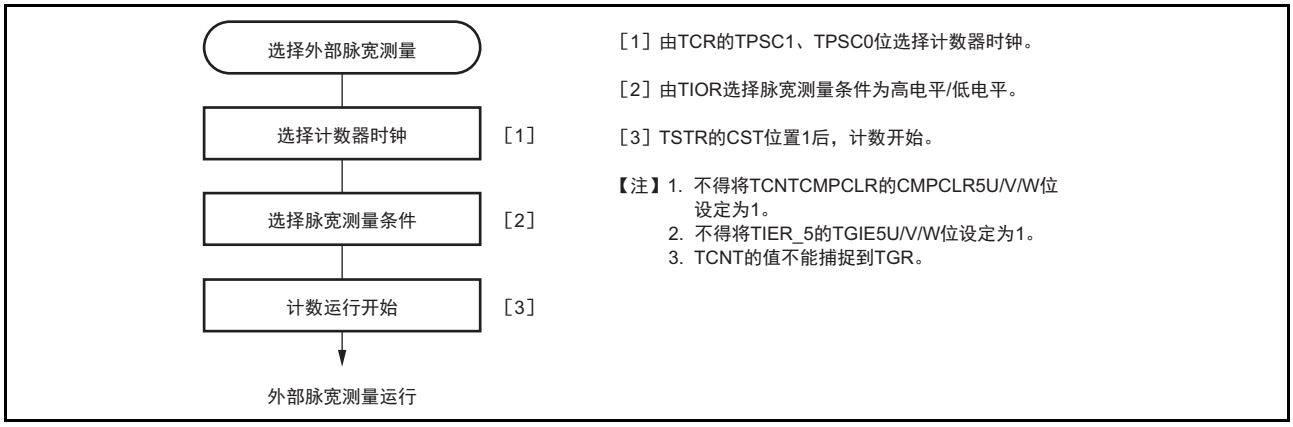


图 10.88　测量外部脉宽的设定步骤例子

(2)　测量外部脉宽的运行例子

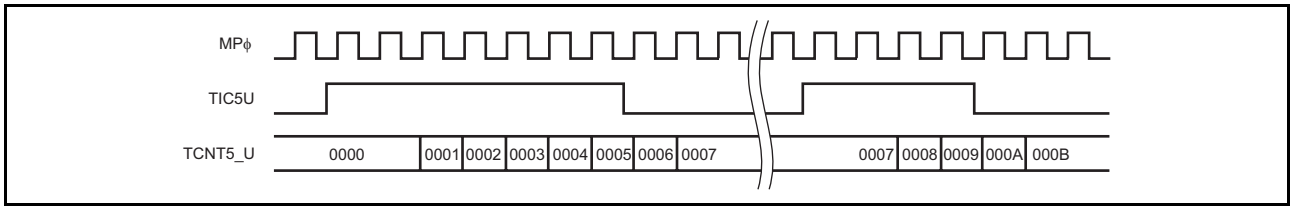


图 10.89　测量外部脉宽的运行例子 （测量高电平脉宽）

10.4.12　空载时间补偿功能

测量输出波形的延迟并反映到占空比，可将外部脉宽测量功能作为互补 PWM 运行时对 PWM 输出波形补偿空载时间的功能。

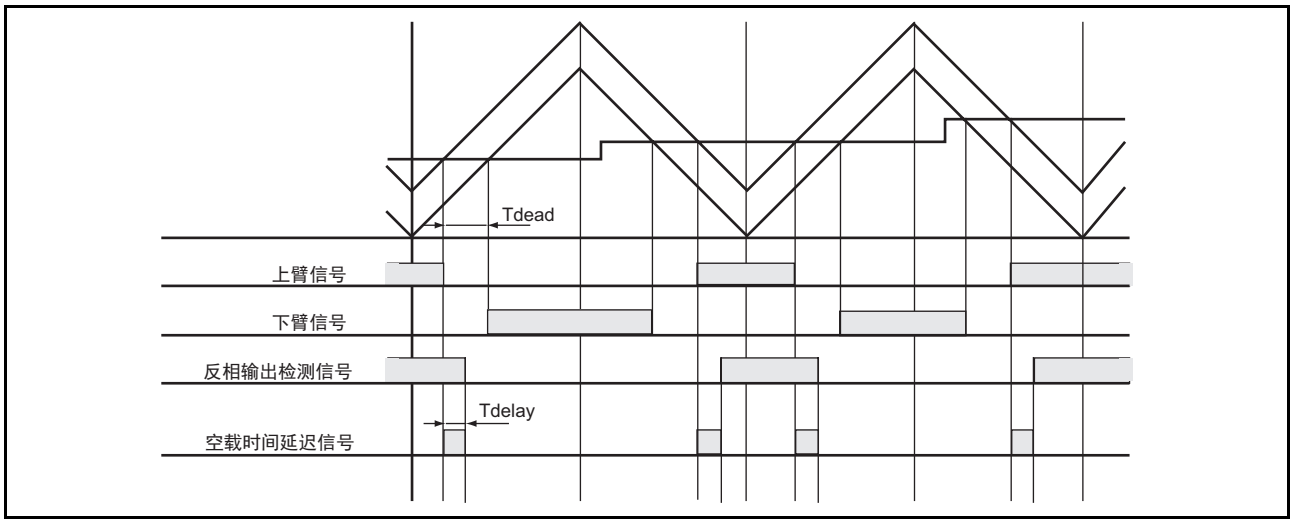


图 10.90　互补 PWM 运行时的空载时间延迟

(1)　空载时间补偿功能的设定步骤例子

使用通道 5 的 3 个计数器的空载时间补偿功能的设定步骤例子如图 10.91 所示。

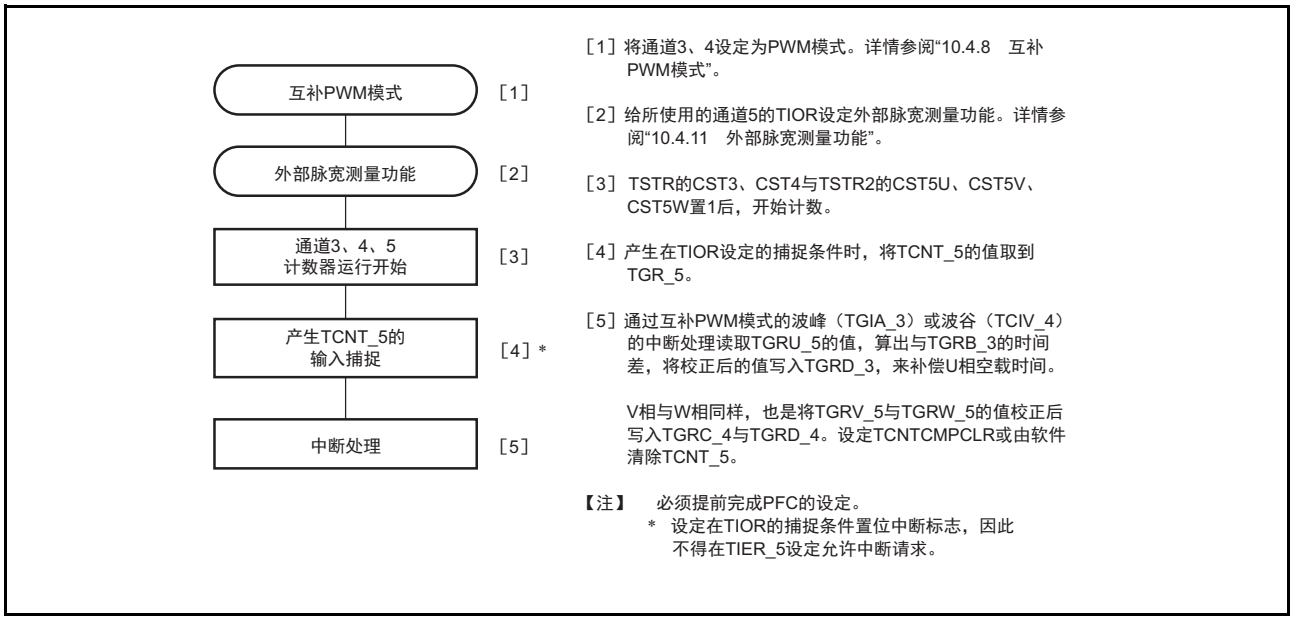


图 10.91　空载时间补偿功能的设定步骤例子

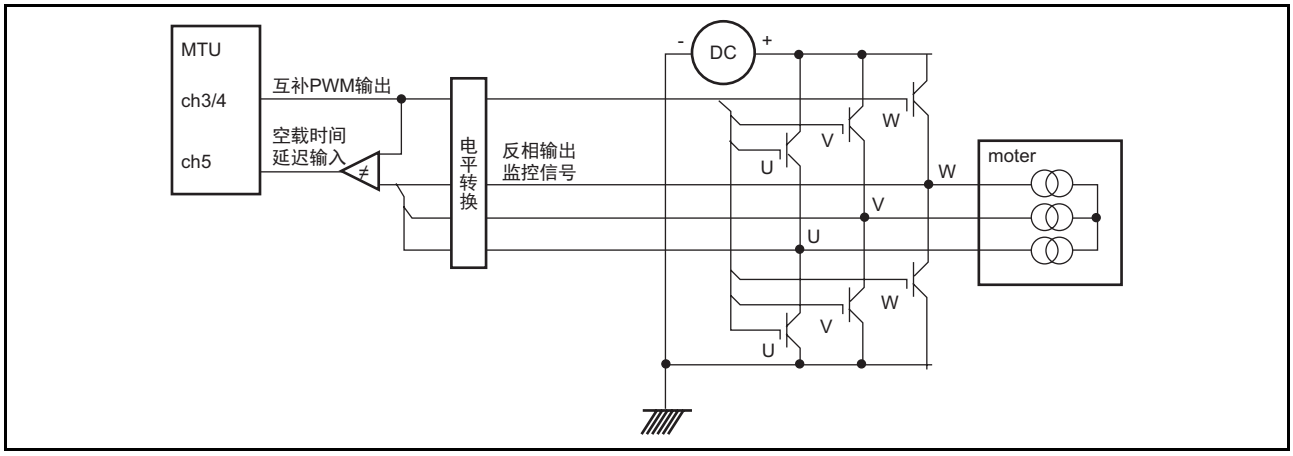


图 10.92　马达控制电路结构例子

10.4.13　在互补 PWM 的 “波峰 / 波谷” 的 TCNT 捕捉运行

互补 PWM 运行时，在 “波峰、波谷、波峰和波谷” 将 TCNT 的值保存到 TGR。由 TIOR 选择取至 TGR 的时序转换。

TCNT 自由运行，且使用时无需清除，在已设定的 “波峰、波谷” 捕捉到 TGR 的运行例子如图 10.93 所示。

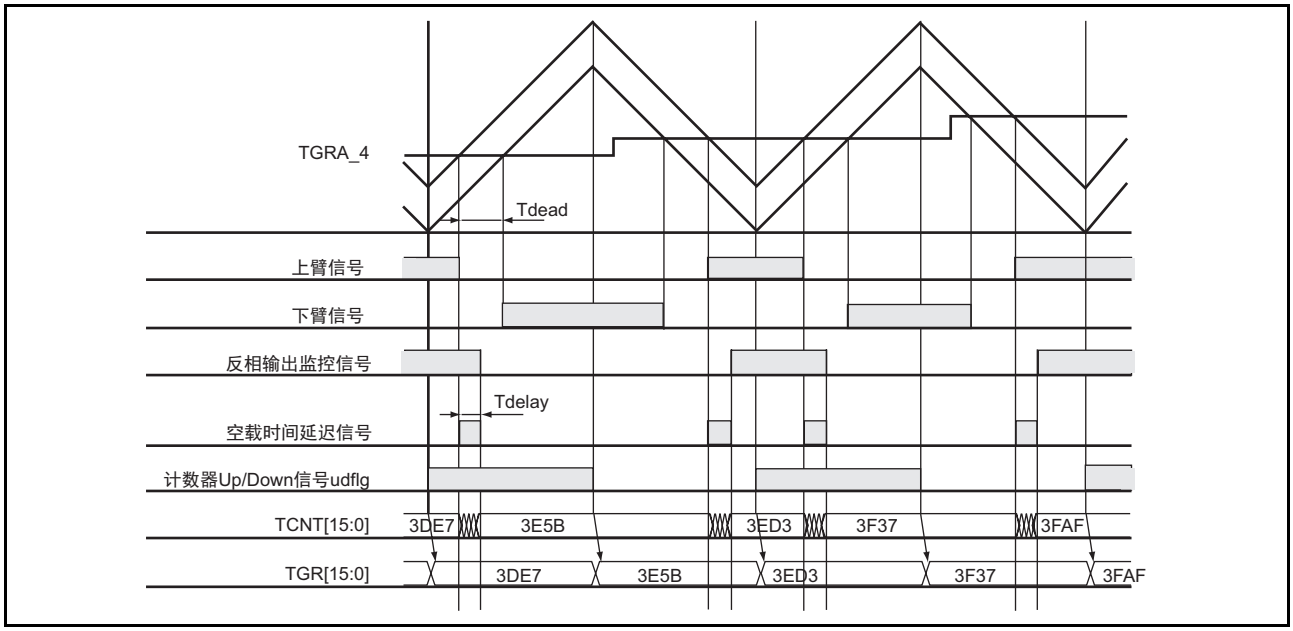


图 10.93　在互补 PWM 的 “波峰 / 波谷” 的 TCNT 捕捉运行

10.5　中断源

10.5.1　中断源与优先顺序

MTU2 的中断源有 TGR 的输入捕捉 / 比较匹配、 TCNT 的上溢和下溢等 3 种。各中断源具有各自专用的状态标志和允许 / 禁止位，因此可独立允许或禁止中断请求信号的产生。

如果产生中断源， TSR 对应的状态标志则置 1。此时如果 TIER 对应的允许 / 禁止位置 1，则请求中断。将状态标志清 0 可解除中断请求。

通道间的优先顺序可通过中断控制器更改，通道内的优先顺序是固定的。详情参阅 “第 6 章　中断控制器（INTC）”。

MTU2 的中断源如表 10.60 所示。

表 10.60　MTU2 中断源

通道	名称	中断源	中断标志	DTC 的启动	优先顺序
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	可	↑
	TGIC_0	TGRC_0 的输入捕捉 / 比较匹配	TGFC_0	可	
	TGID_0	TGRD_0 的输入捕捉 / 比较匹配	TGFD_0	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	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	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	可	
	TGIC_3	TGRC_3 的输入捕捉 / 比较匹配	TGFC_3	可	
	TGID_3	TGRD_3 的输入捕捉 / 比较匹配	TGFD_3	可	
	TCIV_3	TCNT_3 的上溢	TCFV_3	不可	
4	TGIA_4	TGRA_4 的输入捕捉 / 比较匹配	TGFA_4	可	
	TGIB_4	TGRB_4 的输入捕捉 / 比较匹配	TGFB_4	可	
	TGIC_4	TGRC_4 的输入捕捉 / 比较匹配	TGFC_4	可	
	TGID_4	TGRD_4 的输入捕捉 / 比较匹配	TGFD_4	可	
	TCIV_4	TCNT_4 的上溢 / 下溢	TCFV_4	可	
5	TGIU_5	TGRU_5 的输入捕捉 / 比较匹配	TGFU_5	可	
	TGIV_5	TGRV_5 的输入捕捉 / 比较匹配	TGFV_5	可	↓
	TGIW_5	TGRW_5 的输入捕捉 / 比较匹配	TGFW_5	可	低

【注】 表示刚复位的初始状态。可通过中断控制器更改通道之间的优先顺序。

(1)　输入捕捉 / 比较匹配中断

通过各通道产生的 TGR 输入捕捉 / 比较匹配， TSR 的 TGF 标志置 1 时，如果将 TIER 的 TGIE 位置 1，则请求中断。 TGF 标志清 0 可解除中断请求。 MTU2 共有 21 个输入捕捉 / 比较匹配中断，通道 0 有 6 个，通道 3、 4 各有 4 个，通道 1、 2 各有 2 个，通道 5 有 3 个。通道 0 的 TGFE_0、 TGFF_0 标志不能通过输入捕捉置位。

(2)　上溢中断

通过各通道产生的 TCNT 上溢， TSR 的 TCFV 标志置 1 时，如果将 TIER 的 TCIEV 位置 1，则请求中断。 TCFV 标志清 0 可解除中断请求。 MTU2 共有 5 个上溢中断，各通道有 1 个。

(3)　下溢中断

通过各通道产生的 TCNT 下溢， TSR 的 TCFU 标志置 1 时，如果将 TIER 的 TCIEU 位置 1，则请求中断。 TCFU 标志清 0 可解除中断请求。在 MTU2 共有 2 个下溢中断，通道 1、 2 各有 1 个。

10.5.2　DTC 的启动

通过各通道 TGR 的输入捕捉 / 比较匹配中断和通道 4 的上溢中断，可启动 DTC。详情参阅 “第 8 章 数据传送控制器 （DTC）”。

MTU2 共有 20 个输入捕捉 / 比较匹配中断和上溢中断，通道 0、 3 各有 4 个、通道 1、 2 各有 2 个、通道 4 有 5 个、通道 5 有 3 个，可作为 DTC 的启动源。

10.5.3　A/D 转换器的启动

在 MTU2 可由以下 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 10.61 所示。

(1)　TGRA 的输入捕捉 / 比较匹配及在互补 PWM 模式的 TCNT_4 波谷启动 A/D 转换器

可由各通道 TGRA 的输入捕捉 / 比较匹配启动 A/D 转换器。在 TIER_4 的 TTGE2 位置 1 的状态，如果运行互补 PWM， TCNT_4 在波谷 （TCNT_4=H'0000）时，也可启动 A/D 转换器。

在以下条件，对 A/D 转换器产生 A/D 转换开始请求 TRGAN。

- 通过各通道产生TGRA输入捕捉/比较匹配， TSR的TGFA标志置1，且TIER的TTGE位也置1时
- 在TIER_4的TTGE2位置1的状态，运行互补PWM并且TCNT_4在波谷 （TCNT_4=H'0000）时

上述情况中，如果在 A/D 转换器选择 MTU2 的转换开始触发 TRGAN，则开始 A/D 转换。

(2)　由 TCNT_0 和 TGRE_0 的比较匹配启动 A/D 转换器

由通道 0 的 TCNT_0 与 TGRE_0 的比较匹配，产生 A/D 转换开始请求 TRG0N，可启动 A/D 转换器。

通过通道 0 的 TCNT_0 与 TGRE_0 的比较匹配， TSR2_0 的 TGFE 标志置 1 时，如果将 TIER2_0 的 TTGE2 位置 1，则对 A/D 转换器产生 A/D 转换开始请求 TRG0N。此时，如果在 A/D 转换器选择 MTU2 的转换开始触发 TRG0N，则开始 A/D 转换。

(3)　由 A/D 转换开始请求延迟功能启动 A/D 转换器

A/D 转换请求控制寄存器 （TADCR）的 TAD4AE、 TAD4BE 位置 1 时，通过 TADCORA、 TADCORB 与 TCNT_4 匹配，产生 TRG4AN 和 TRG4BN，即可启动 A/D 转换器。详情参阅 “10.4.9 A/D 转换开始请求延迟功能”。

产生 TRG4AN 时，如果在 A/D 转换器选择 MTU2 的转换开始触发 TRG4AN，则开始 A/D 转换。另外，产生 TRG4BN 时，如果在 A/D 转换器选择 MTU2 的转换开始触发 TRG4BN，则开始 A/D 转换。

表 10.61　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 转换开始请求
TGRA_0 和 TCNT_0	输入捕捉 / 比较匹配	TRGAN
TGRA_1 和 TCNT_1		
TGRA_2 和 TCNT_2		
TGRA_3 和 TCNT_3		
TGRA_4 和 TCNT_4		
TCNT_4	互补 PWM 模式时 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

10.6　运行时序

10.6.1　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 10.94 和图 10.95 所示。另外，外部时钟运行（普通模式）时的 TCNT 计数时序如图 10.96、外部时钟运行（相位计数模式）时的 TCNT 计数时序如图 10.97 所示。

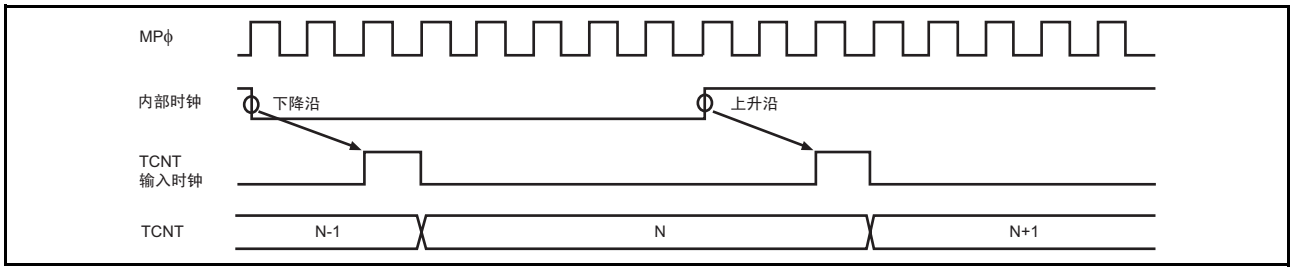


图 10.94　内部时钟运行时的计数时序（通道 0 ～ 4）

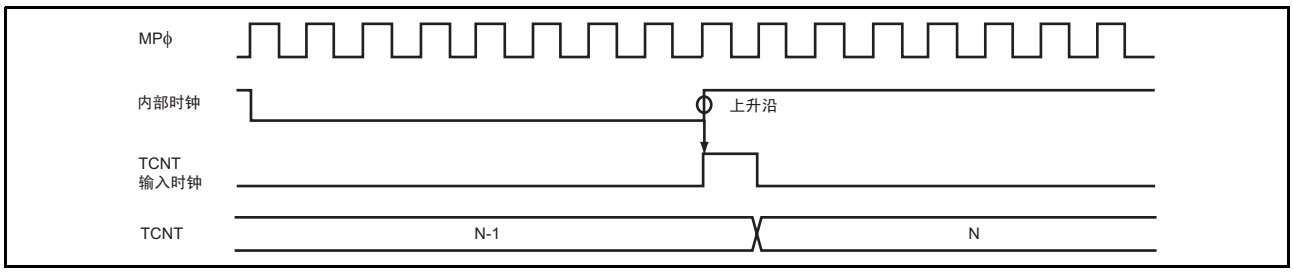


图 10.95　内部时钟运行时的计数时序（通道 5）

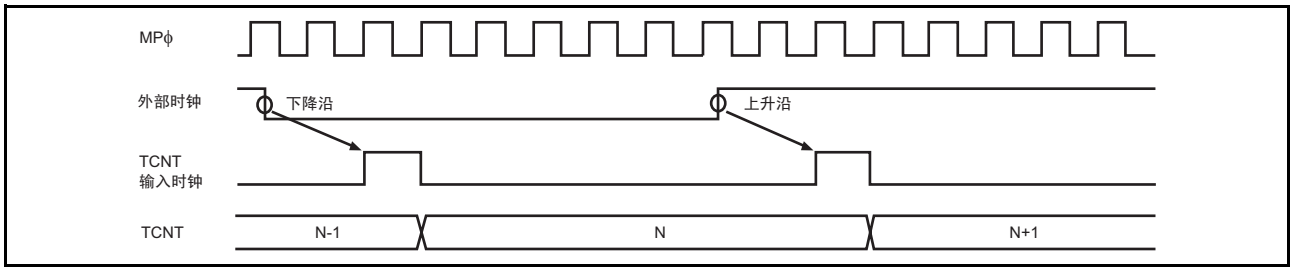


图 10.96　外部时钟运行时的计数时序 （通道 0 ～ 4）

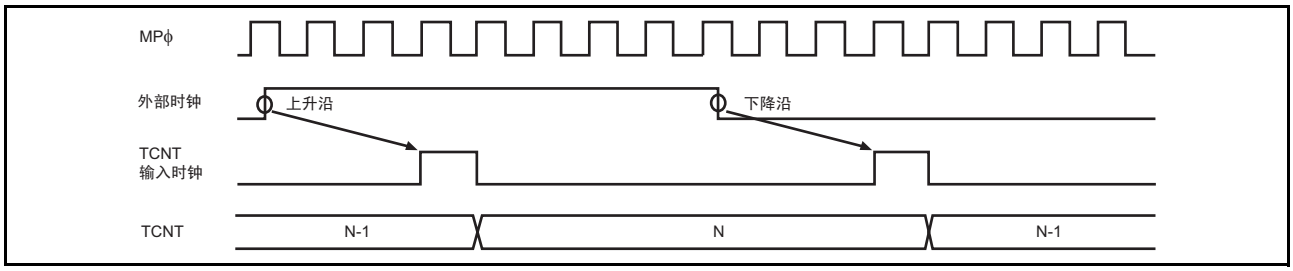


图 10.97　外部时钟运行时的计数时序 （相位计数模式）

(2)　输出比较的输出时序

在 TCNT 和 TGR 匹配的最后状态 （更新与 TCNT 匹配后的计数值的时序），产生比较匹配信号。此时，向输出比较输出引脚 （TIOC 引脚）输出 TIOR 设定的输出值。 TCNT 和 TGR 匹配后，到产生 TCNT 输入时钟前，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 10.98 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 10.99 所示。

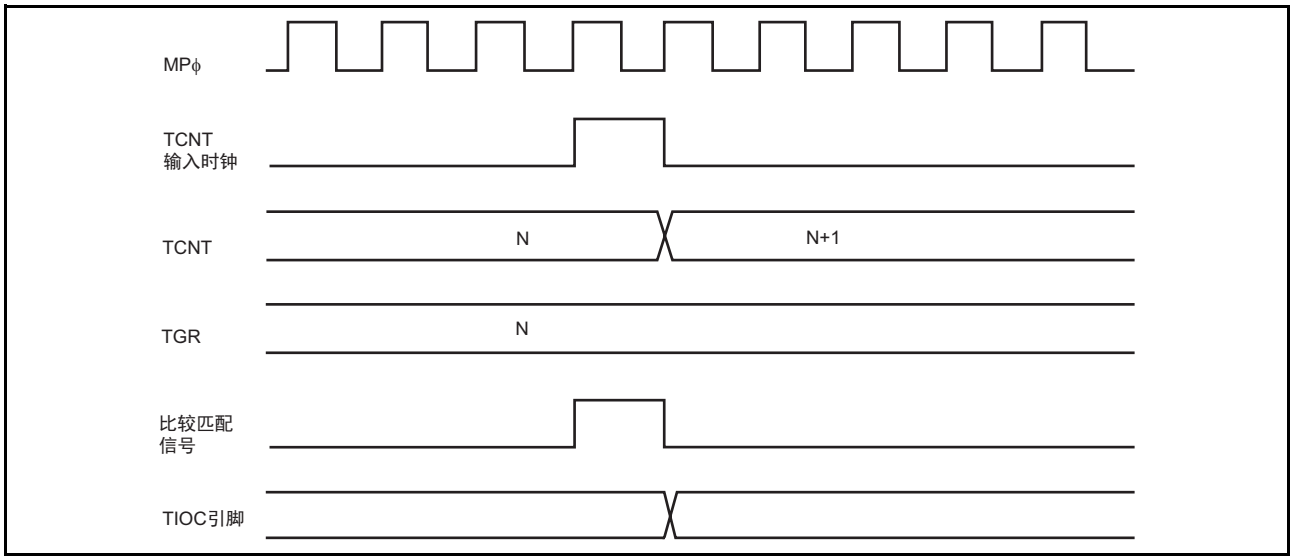


图 10.98　输出比较的输出时序 （普通模式、 PWM 模式）

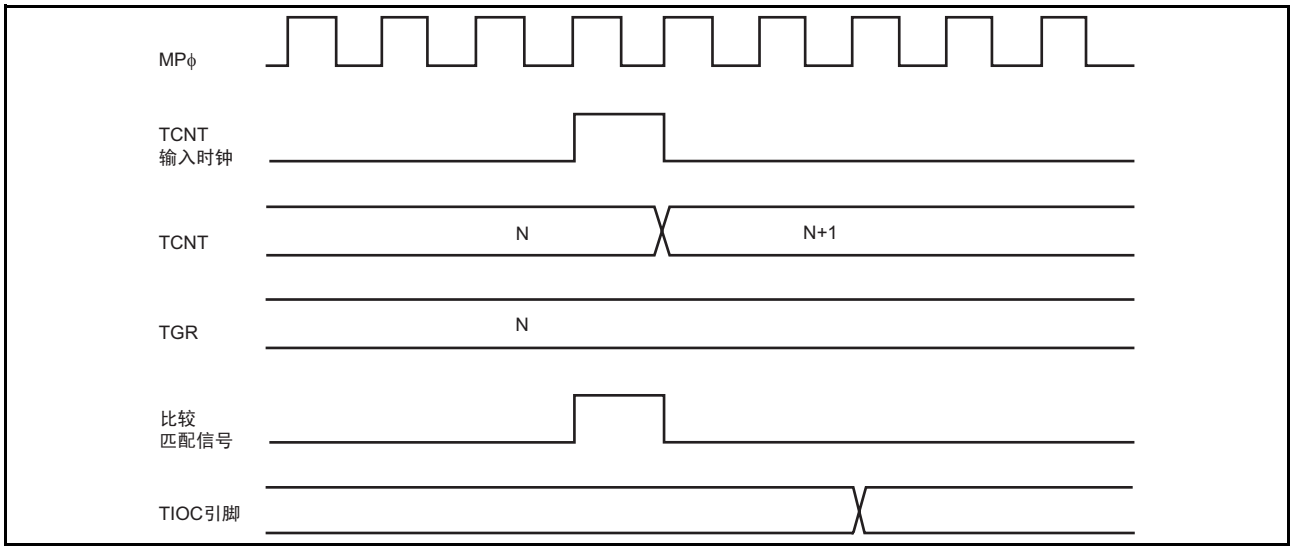


图 10.99　输出比较的输出时序 （互补 PWM 模式、复位同步 PWM 模式）

(3)　输入捕捉信号时序

输入捕捉时序如图 10.100 所示。

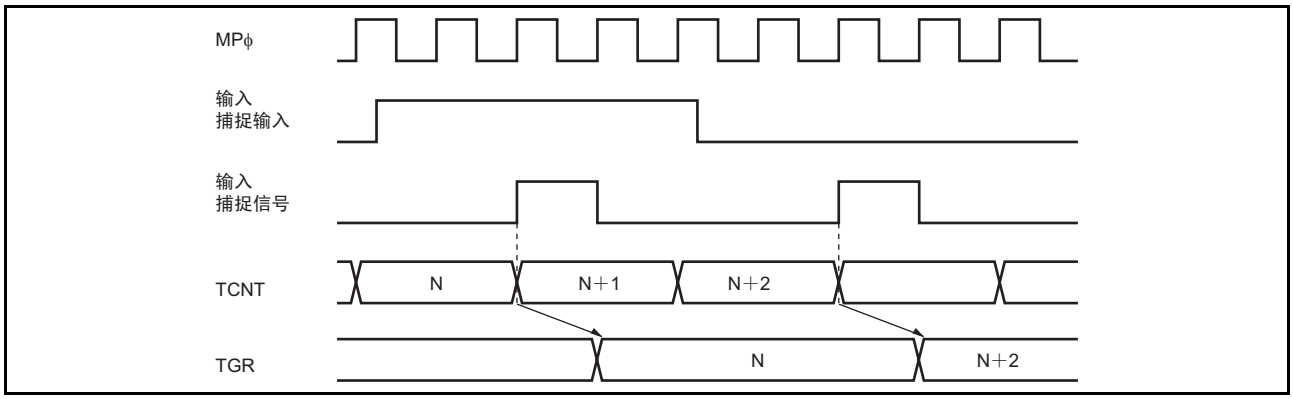


图 10.100　输入捕捉的输入信号时序

(4)　由比较匹配 / 输入捕捉清除计数器的时序

由产生的比较匹配，指定计数器清除的时序，如图 10.101、图 10.102 所示。
由产生的输入捕捉，指定计数器清除的时序，如图 10.103 所示。

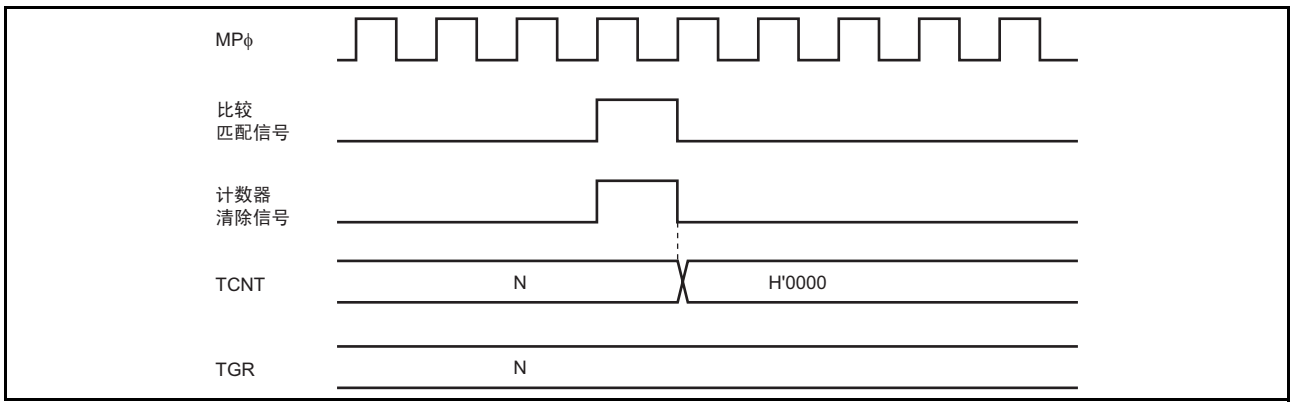


图 10.101　计数器清除时序 （比较匹配）（通道 0 ～ 4）

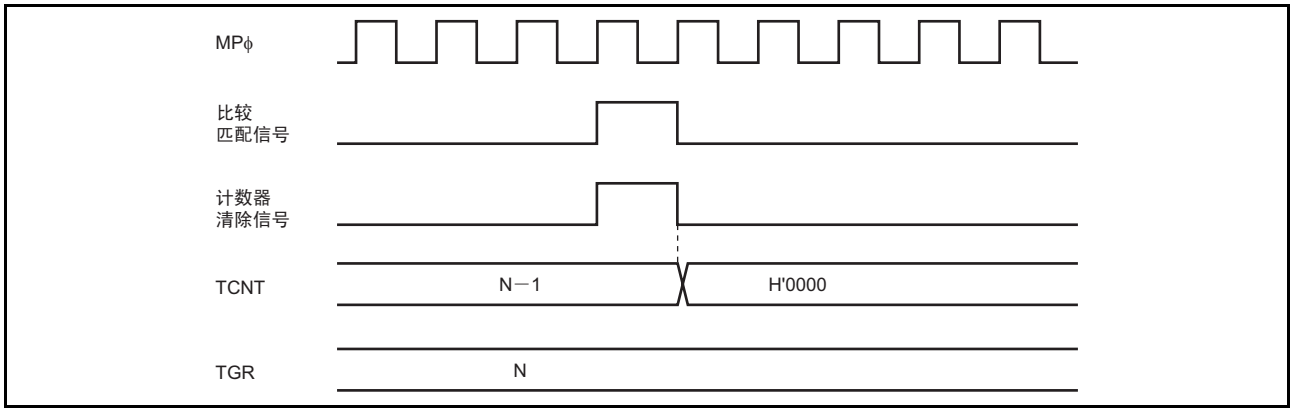


图 10.102　计数器清除时序 （比较匹配）（通道 5）

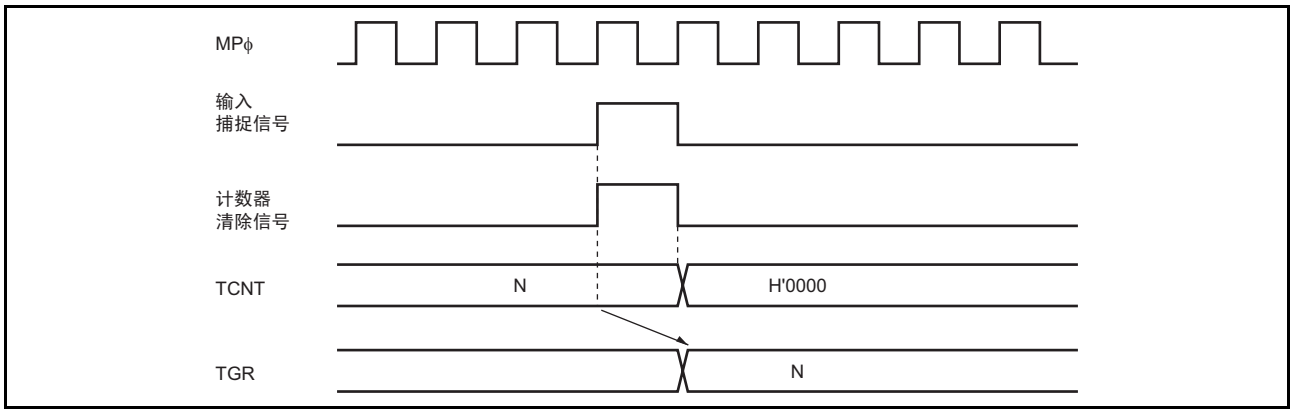


图 10.103　计数器清除时序 （输入捕捉）（通道 0 ～ 5）

(5)　缓冲运行时序

缓冲运行时序如图 10.104 ～图 10.106 所示。

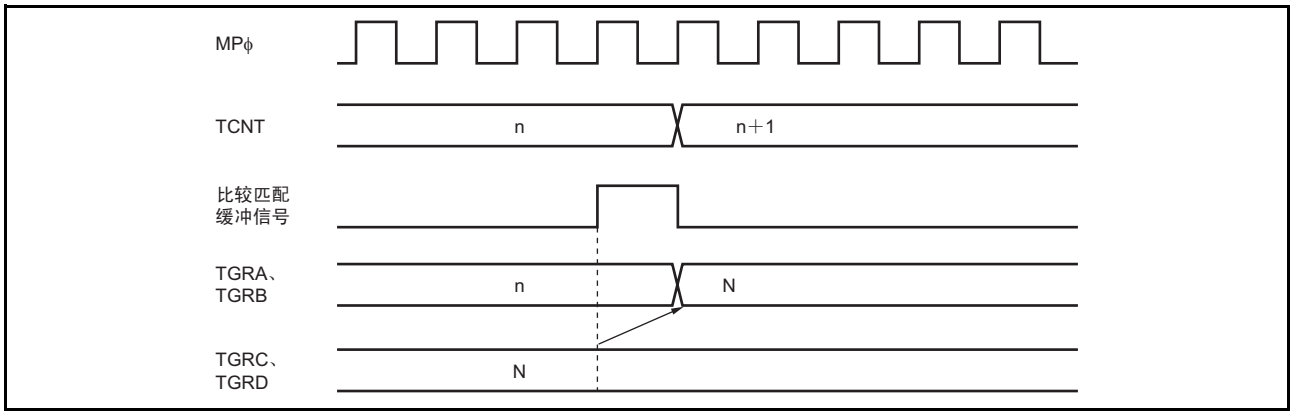


图 10.104　缓冲运行时序 （比较匹配）

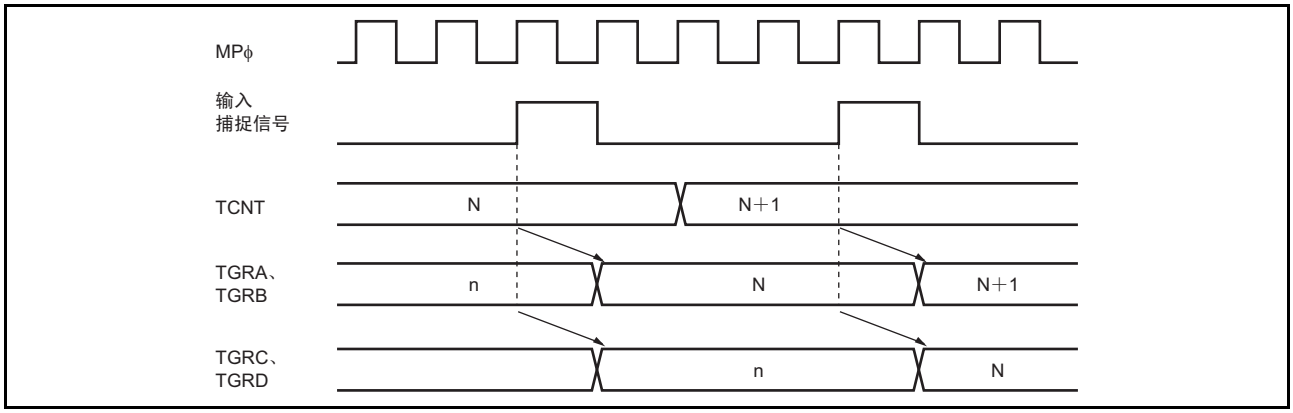


图 10.105　缓冲运行时序 （输入捕捉）

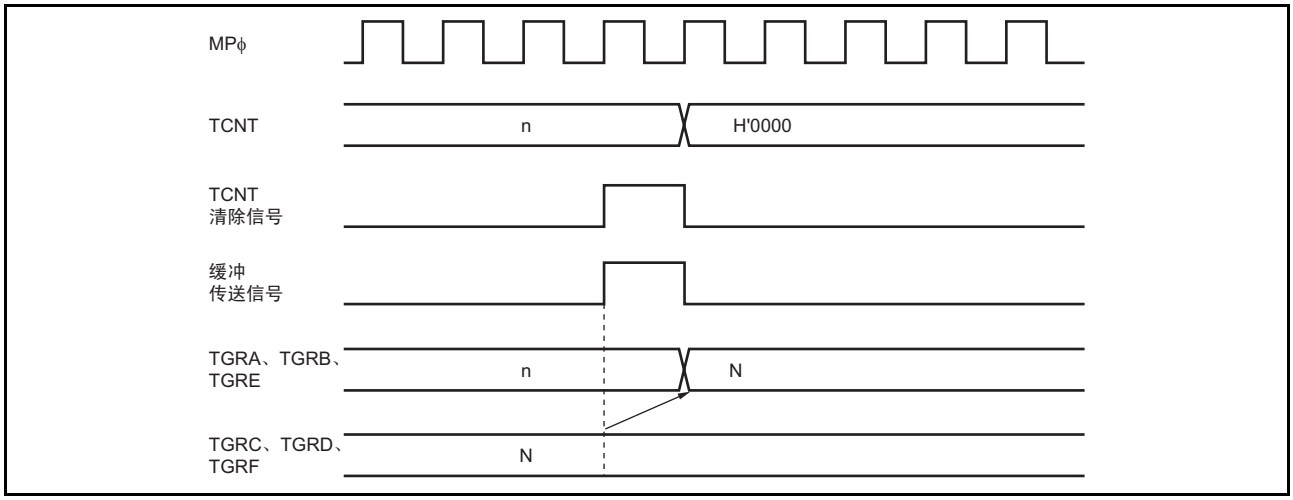


图 10.106　缓冲运行时序 （清除 TCNT 时）

(6)　缓冲传送时序 （互补 PWM 模式时）

互补 PWM 模式的缓冲传送时序如图 10.107 ～图 10.109 所示。

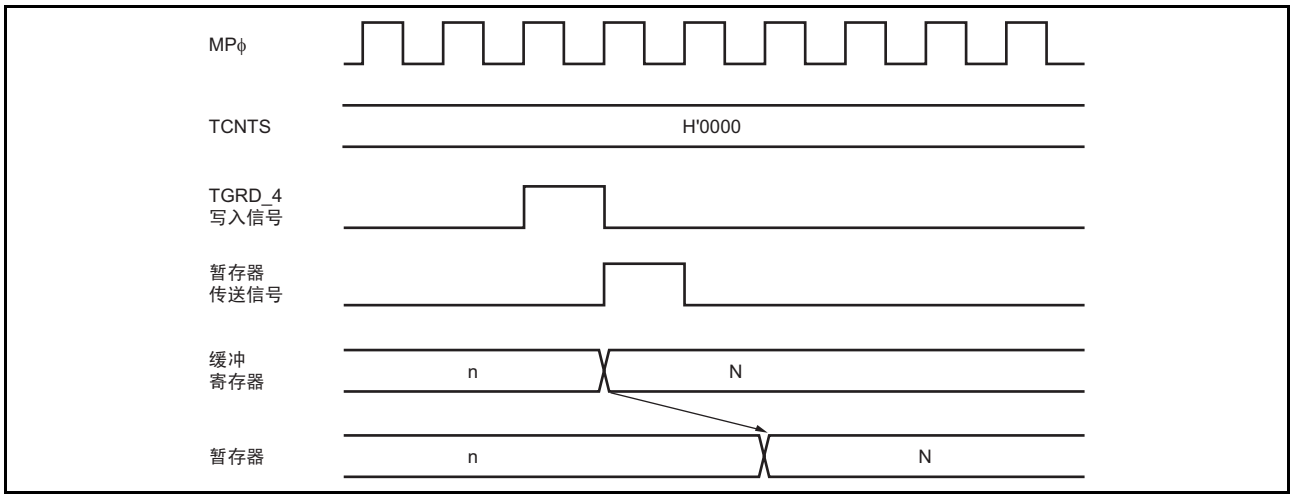


图 10.107　缓冲寄存器到暂存器的传送时序 （TCNTS 停止时）

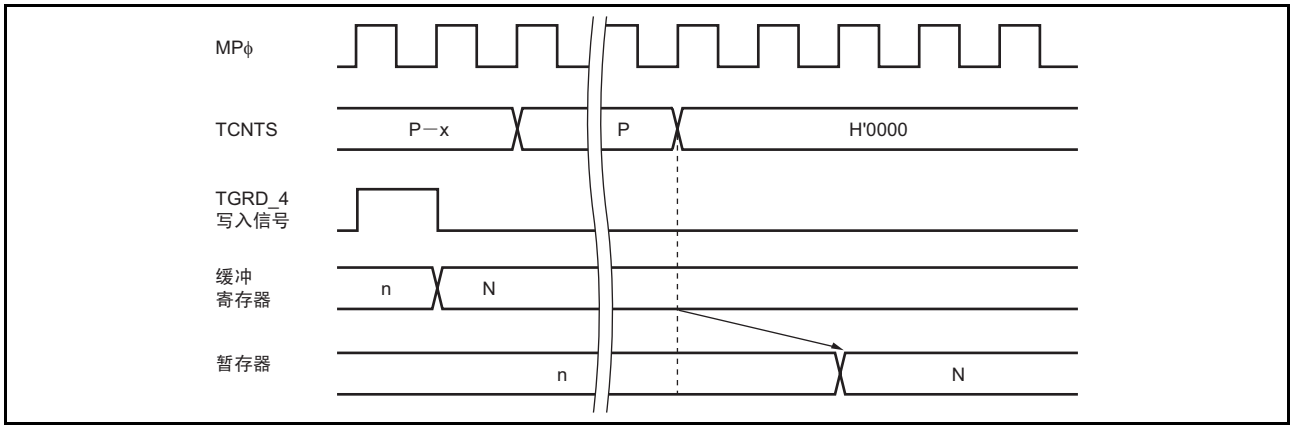


图 10.108　缓冲寄存器到暂存器的传送时序 （TCNTS 运行时）

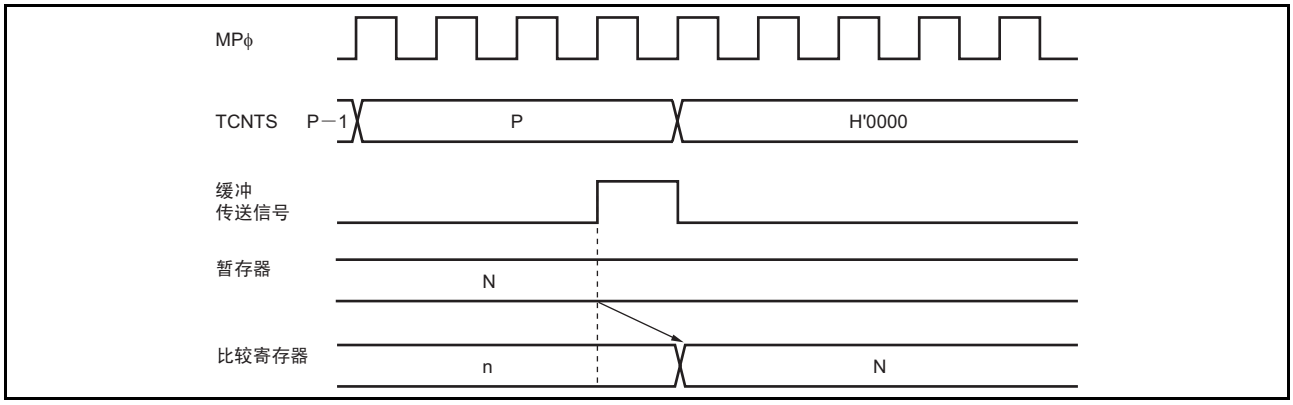


图 10.109　暂存器到比较寄存器的传送时序

10.6.2　中断信号时序

(1)　比较匹配时的 TGF 标志的置位时序

通过产生比较匹配置位 TSR 的 TGF 标志的时序和 TGI 中断请求信号的时序分别如图 10.110 和图 10.111 所示。

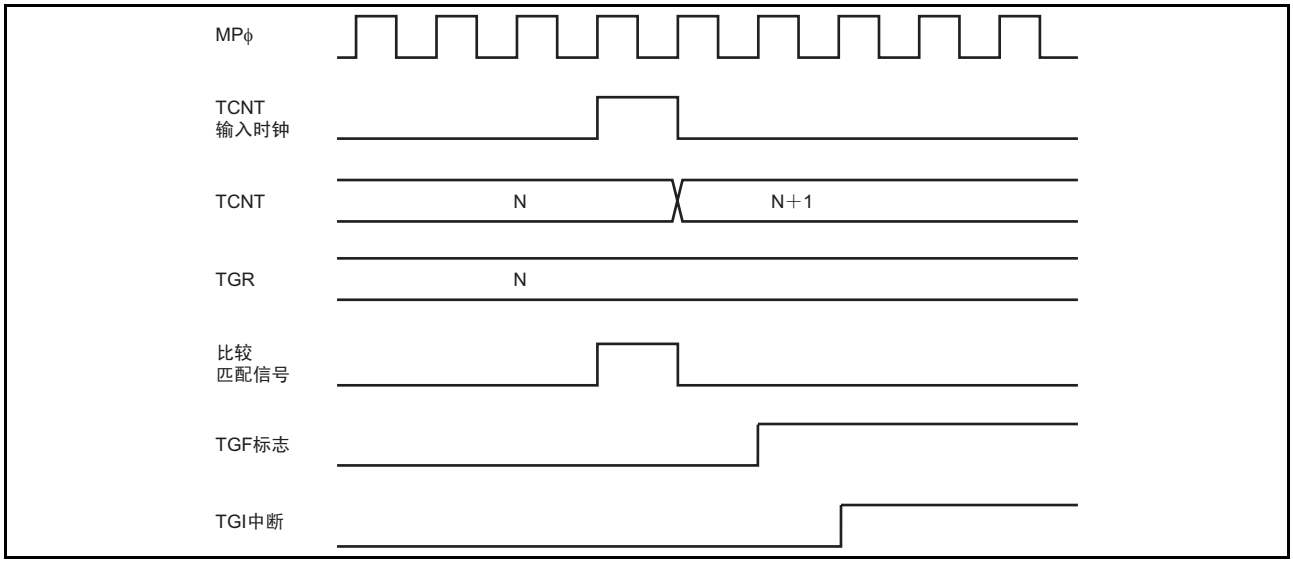


图 10.110　TGI 中断时序 （比较匹配）（通道 0 ～ 4）

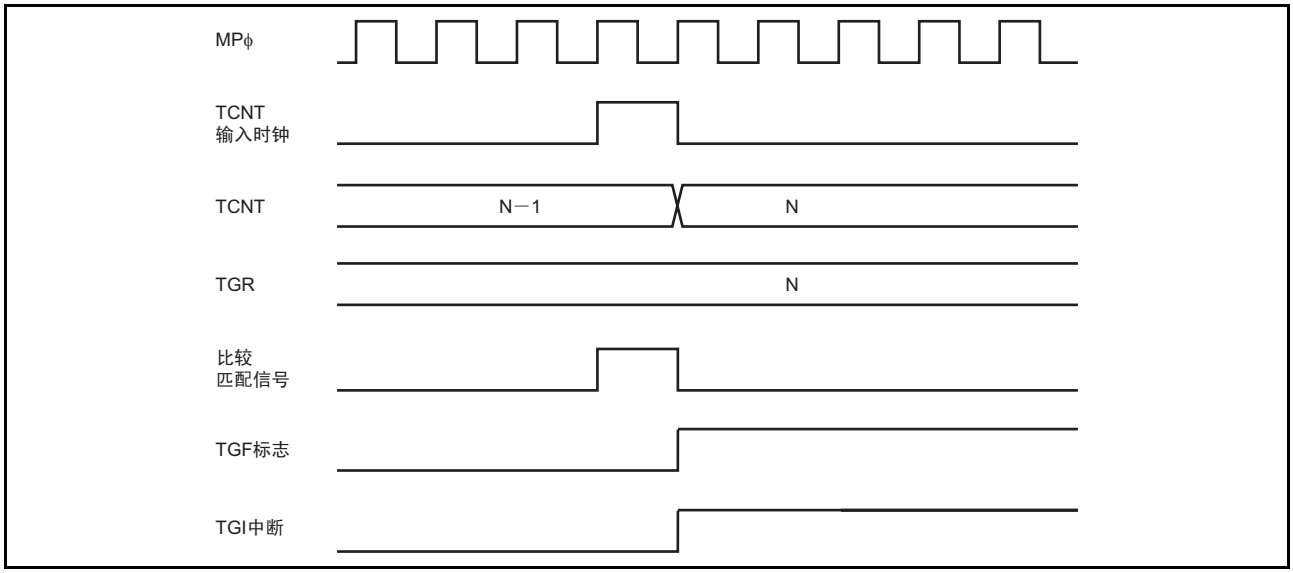


图 10.111　TGI 中断时序 （比较匹配）（通道 5）

(2)　输入捕捉时的 TGF 标志的置位时序

通过产生输入捕捉置位 TSR 的 TGF 标志的时序和 TGI 中断请求信号的时序分别如图 10.112 和图 10.113 所示。

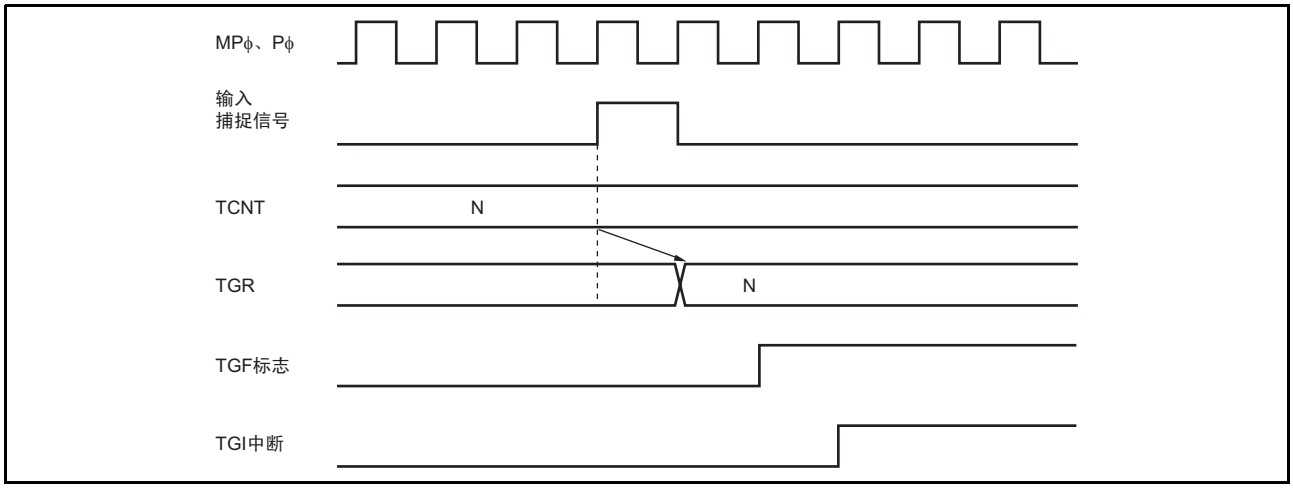


图 10.112　TGI 中断时序 （输入捕捉）（通道 0 ～ 4）

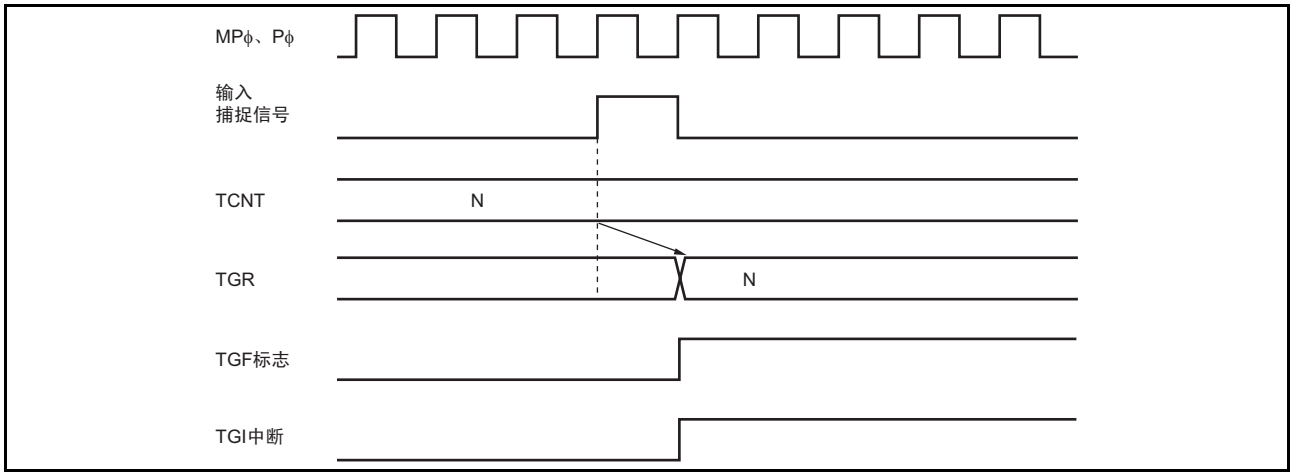


图 10.113　TGI 中断时序 （输入捕捉）（通道 5）

(3)　TCFV 标志 /TCFU 标志的置位时序

通过产生上溢置位 TSR 的 TCFV 标志的时序和 TCIV 中断请求信号的时序如图 10.114 所示。
通过产生下溢置位 TSR 的 TCFU 标志的时序和 TCIU 中断请求信号的时序如图 10.115 所示。

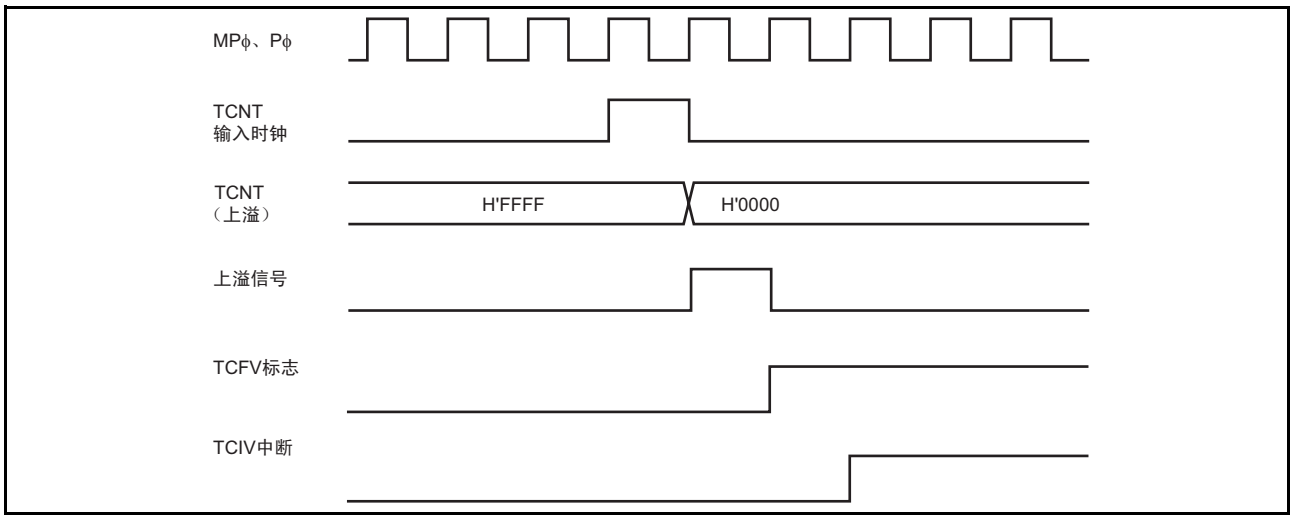


图 10.114　TCIV 中断的置位时序

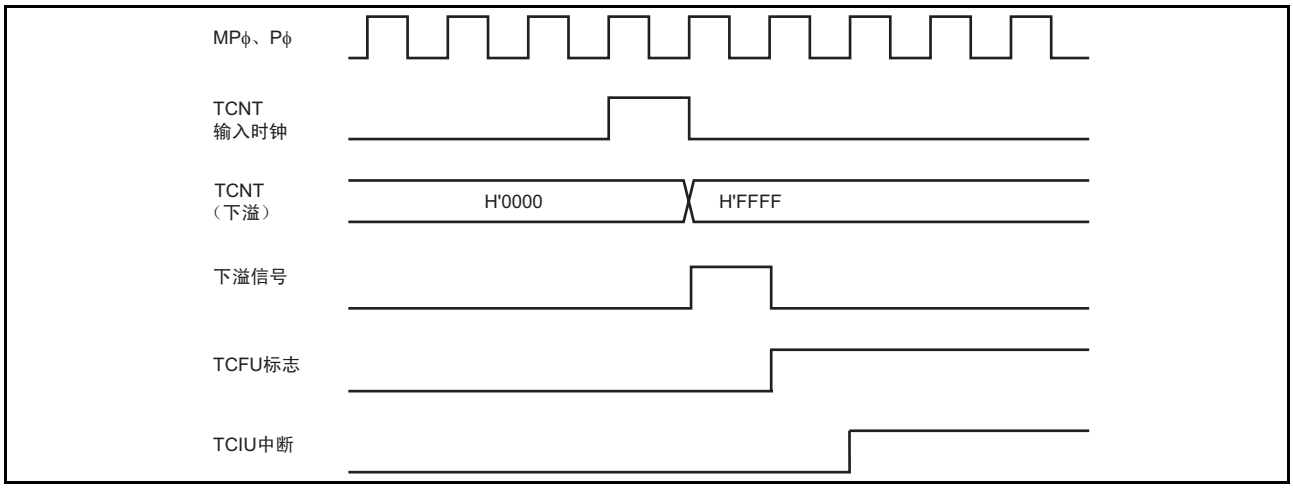


图 10.115　TCIU 中断的置位时序

(4)　状态标志的清除时序

CPU 读取 1 再写入 0，可清除状态标志。启动 DTC 时，也可自动清除。 CPU 清除状态标志的时序如图 10.116、图 10.117 所示； DTC 清除状态标志的时序如图 10.118、图 10.119 所示。

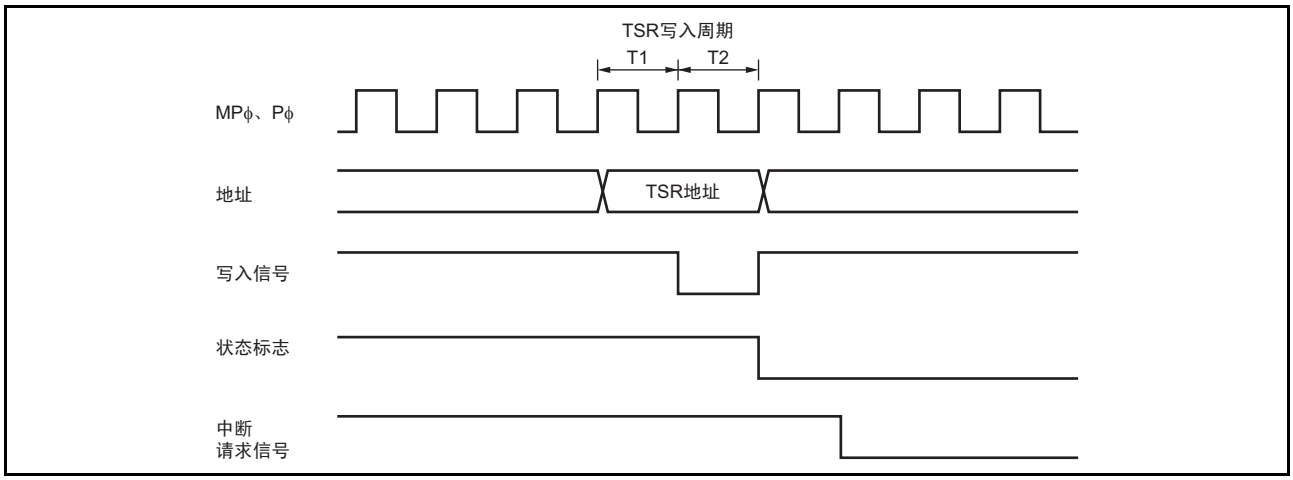


图 10.116　CPU 清除状态标志的时序 （通道 0 ～ 4）

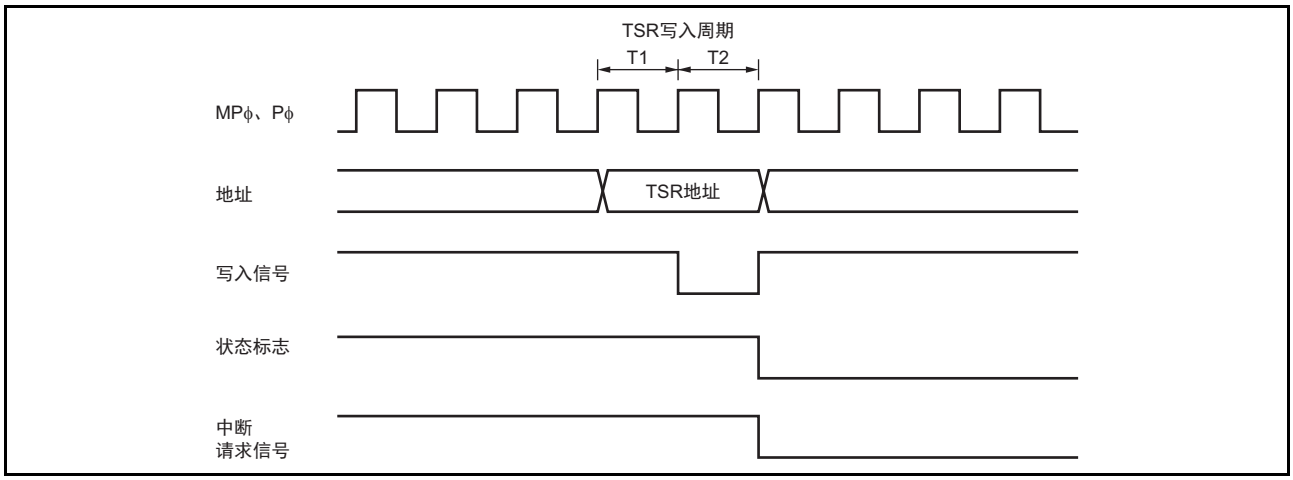


图 10.117　CPU 清除状态标志的时序 （通道 5）

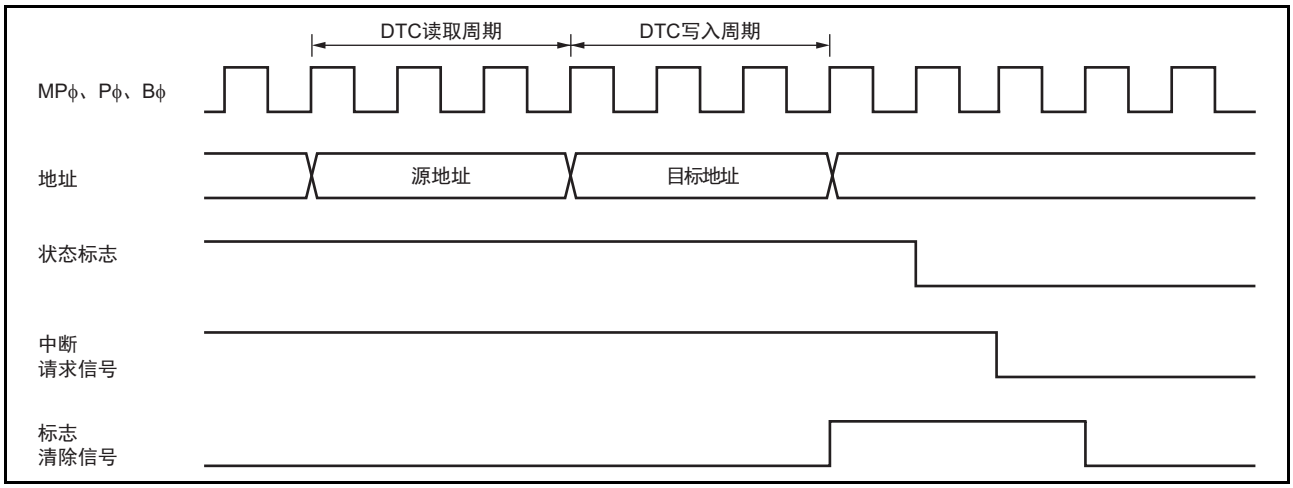


图 10.118　启动 DTC 清除状态标志的时序 （通道 0 ～ 4）

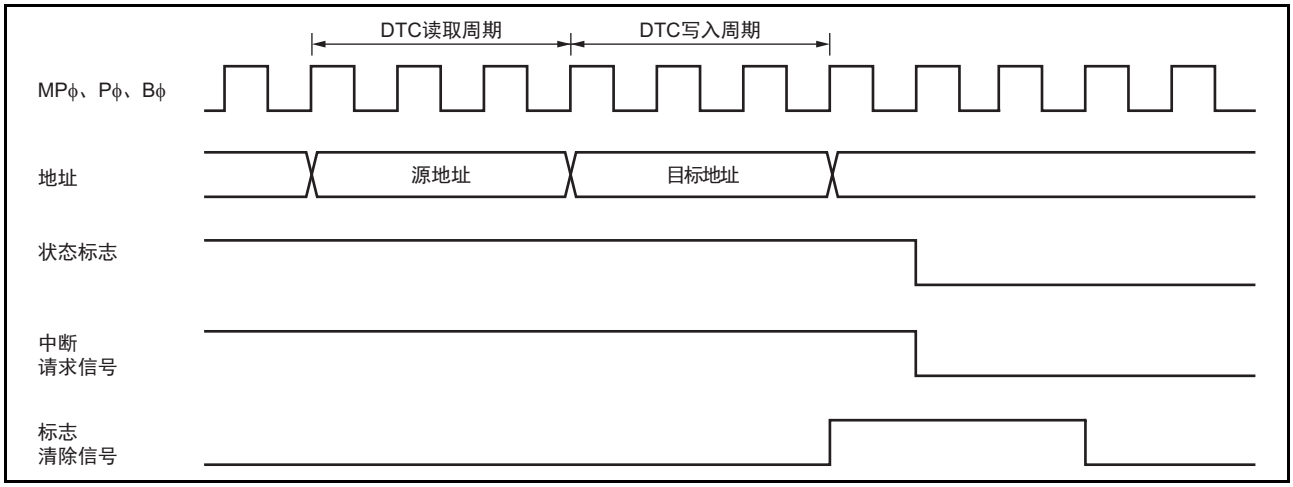


图 10.119　启动 DTC 清除状态标志的时序 （通道 5）

10.7　使用时的注意事项

10.7.1　模块待机模式的设定

MTU2 可由待机控制寄存器设定禁止 / 允许本模块运行。为初始值时， MTU2 停止运行。解除模块待机模式，可存取寄存器。详情参阅 “第 22 章　低功耗模式”。

10.7.2　输入时钟的限制事项

输入时钟的脉宽为单边沿时至少需要 1.5 个状态时钟，为双边沿时至少需要 2.5 个状态时钟。注意：如果没有达到此脉宽则无法正常运行。

在相位计数模式， 2 个输入时钟的相位差及重叠至少需要 1.5 个状态时钟，并且脉宽至少是 2.5 个状态时钟。相位计数模式的输入时钟条件如图 10.120 所示。

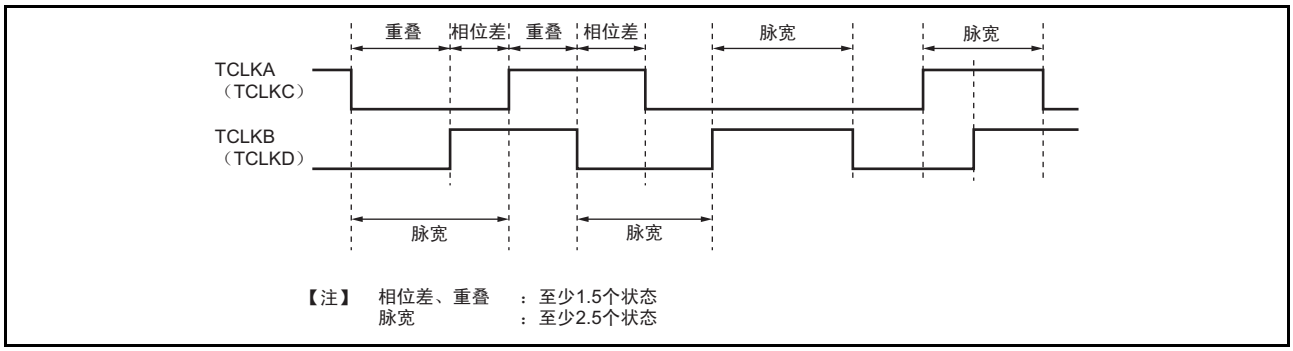


图 10.120　相位计数模式的相位差、重叠及脉宽

10.7.3　周期设定时的注意事项

如果设定为由比较匹配清除计数器，则在 TCNT 和 TGR 的值匹配的最后状态 （更新 TCNT 匹配后的计数值的时序）清除。因此，实际计算计数器的频率公式如下。

- 通道0～4时

$$f = \frac{MP\phi}{(N+1)}$$

- 通道5时

$$f = \frac{MP\phi}{N}$$

f　　：计数器频率
MPϕ　：MTU2 时钟工作频率
N　　：TGR 的设定值

10.7.4　TCNT 的写入与清除的竞争

如果在 TCNT 写入周期的 T2 状态产生计数器清除信号，则不写入 TCNT，而优先清除 TCNT。
此时序如图 10.121 所示。

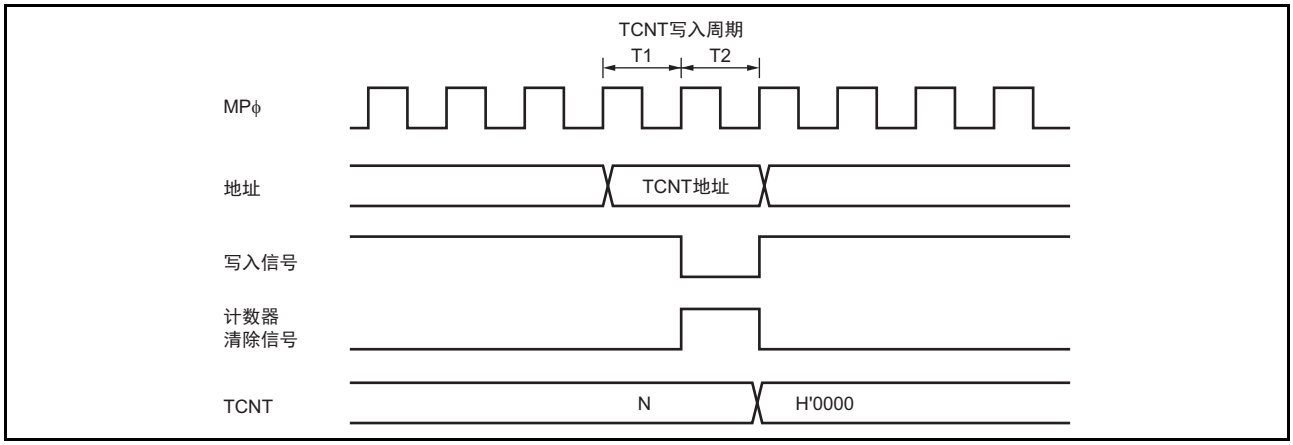


图 10.121　TCNT 的写入与清除的竞争

10.7.5　TCNT 的写入与递增计数的竞争

即使在 TCNT 写入周期的 T2 状态产生递增计数，也不递增计数，而优先写入 TCNT。
此时序如图 10.122 所示。

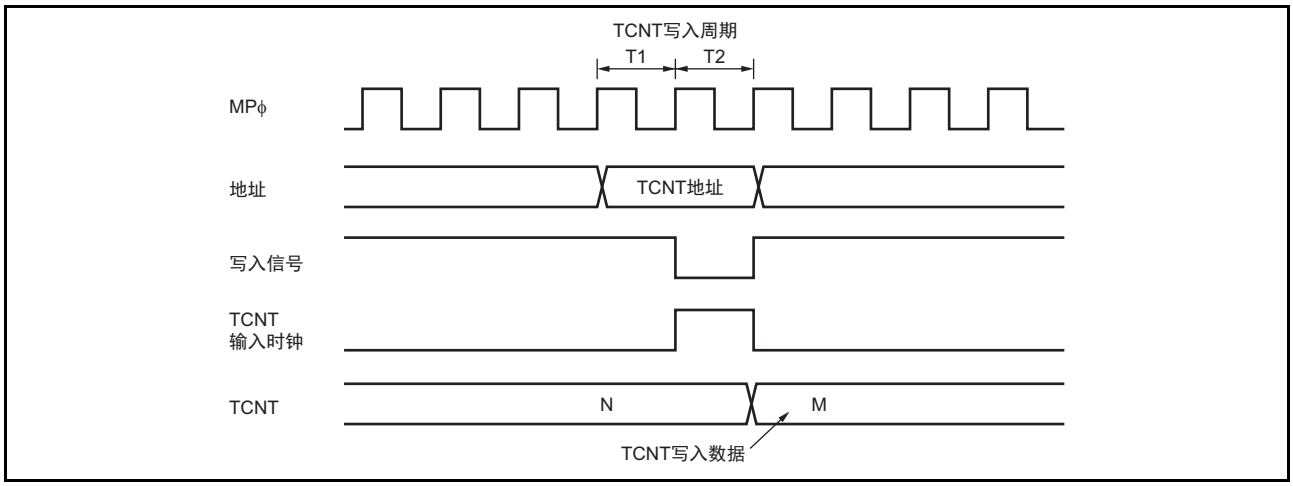


图 10.122　TCNT 的写入与递增计数的竞争

10.7.6　TGR 的写入与比较匹配的竞争

在 TGR 写入周期的 T2 状态产生比较匹配时，写入 TGR 并且也产生比较匹配信号。
此时序如图 10.123 所示。

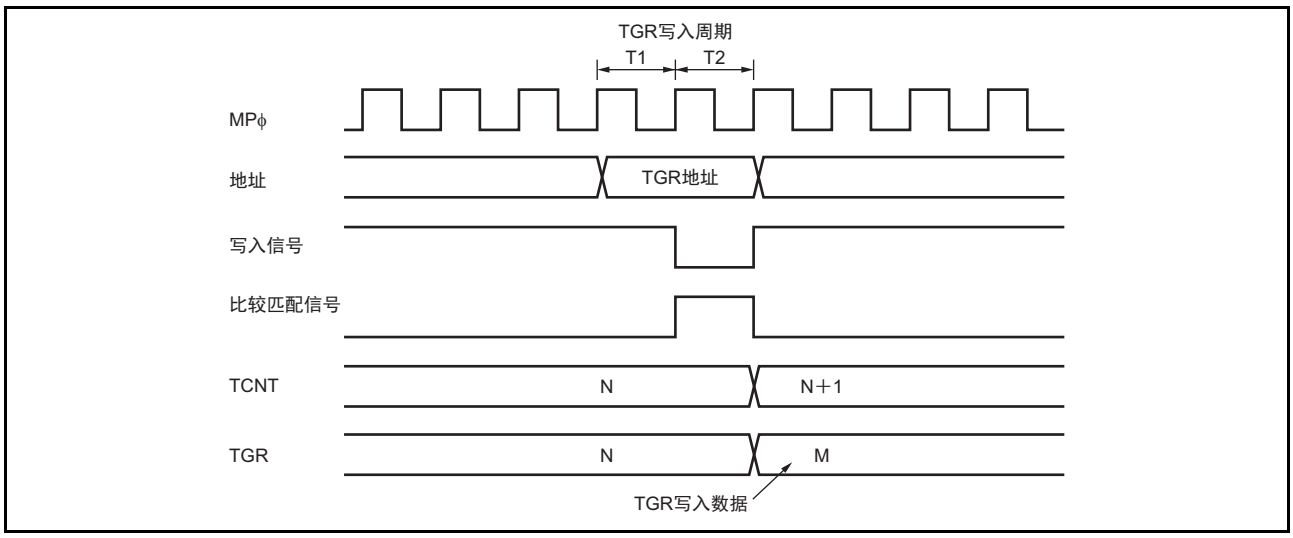


图 10.123　TGR 的写入与比较匹配的竞争

10.7.7　缓冲寄存器的写入与比较匹配的竞争

在 TGR 写入周期的 T2 状态产生比较匹配时，通过缓冲运行传送到 TGR 的数据为写入前的数据。此时序如图 10.124 所示。

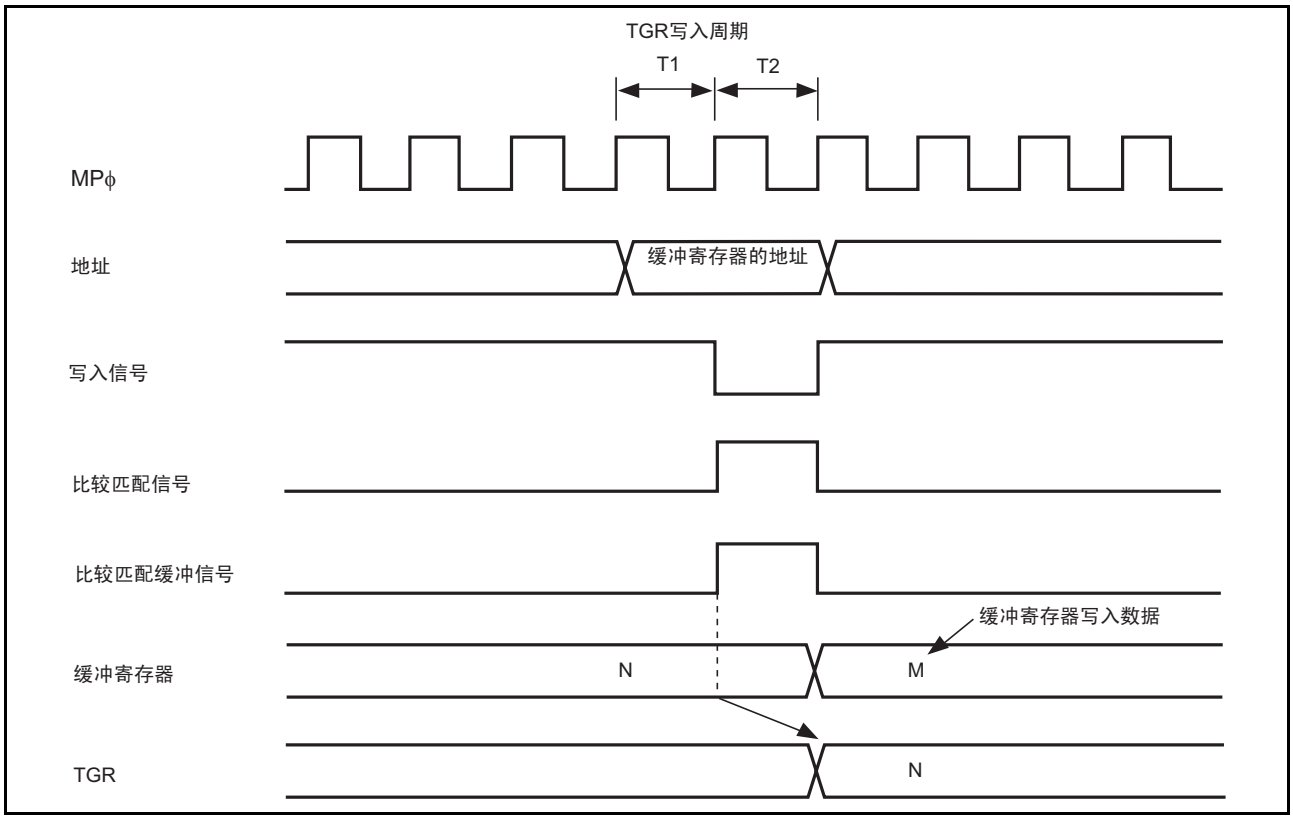


图 10.124　缓冲寄存器的写入与比较匹配的竞争

10.7.8　缓冲寄存器的写入与 TCNT 清除的竞争

由缓冲传送模式寄存器（TBTM）将缓冲传送时序设定为 TCNT 清除时，如果在 TGR 写入周期的 T2 状态产生 TCNT 清除，则通过缓冲运行传送到 TGR 的数据为写入前的数据。

此时序如图 10.125 所示。

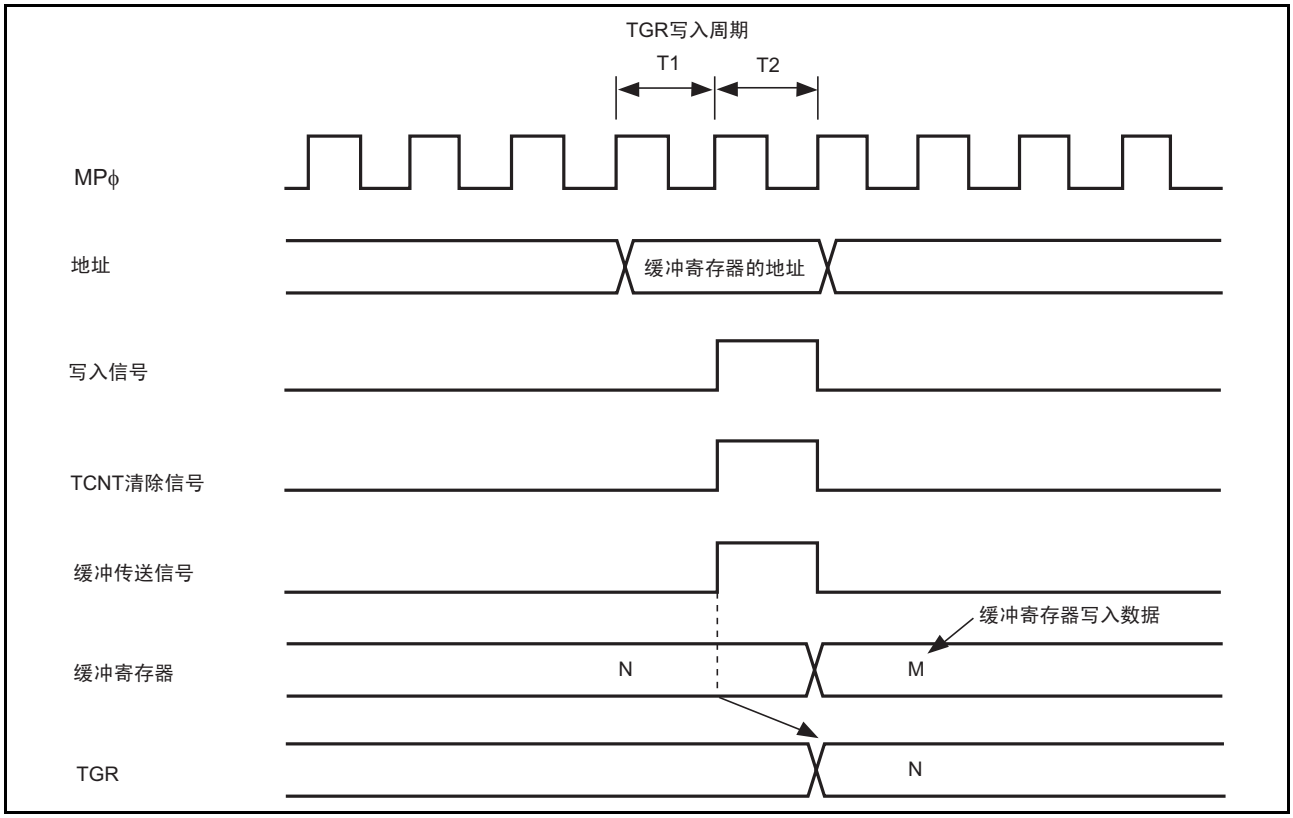


图 10.125　缓冲寄存器的写入与清除 TCNT 的竞争

10.7.9　TGR 的读取与输入捕捉的竞争

在 TGR 读取周期的 T1 状态产生输入捕捉信号时，读取的数据在通道 0 ～ 4 为输入捕捉传送前的数据，在通道 5 为输入捕捉传送后的数据。

此时序如图 10.126、图 10.127 所示。

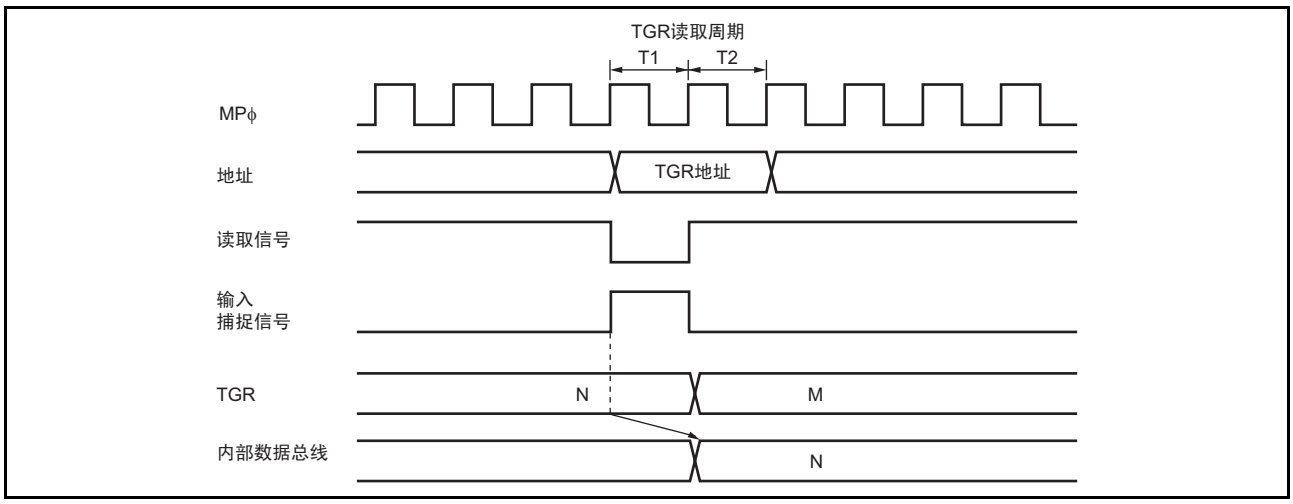


图 10.126　TGR 的读取与输入捕捉的竞争（通道 0 ～ 4）

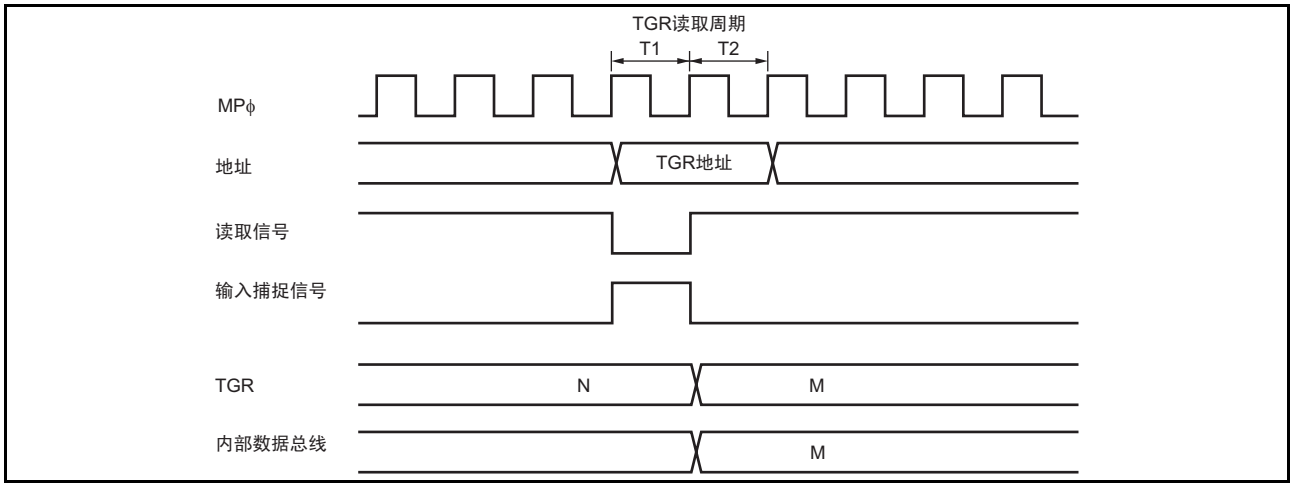


图 10.127　TGR 的读取与输入捕捉的竞争（通道 5）

10.7.10　TGR 的写入与输入捕捉的竞争

在 TGR 写入周期的 T2 状态产生输入捕捉信号时，在通道 0 ～ 4 不写入 TGR 而优先输入捕捉，在通道 5 写入 TGR 并且也产生输入捕捉信号。

此时序如图 10.128、图 10.129 所示。

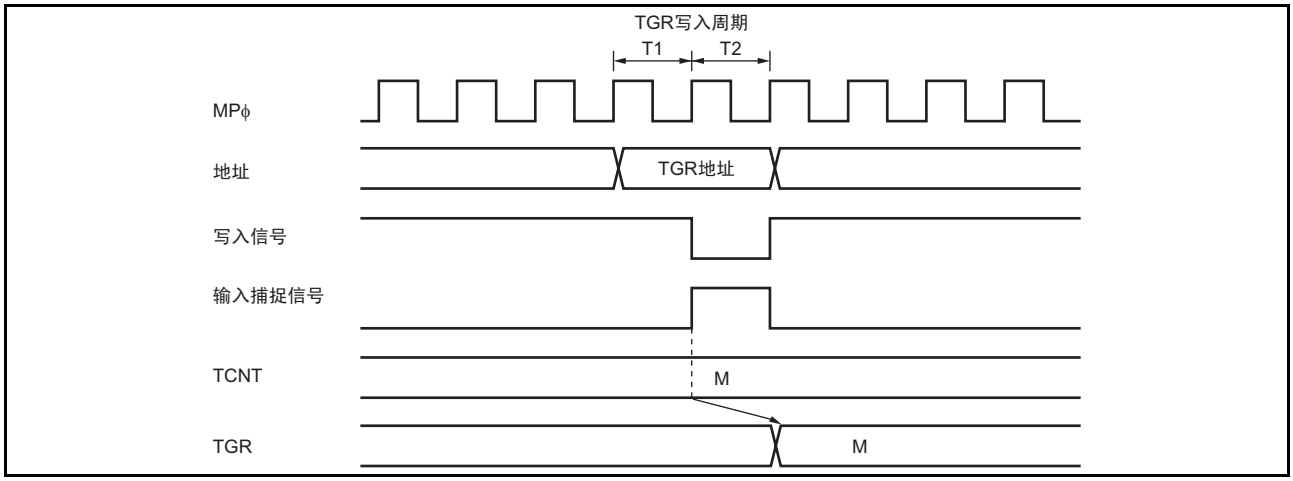


图 10.128　TGR 的写入与输入捕捉的竞争（通道 0 ～ 4）

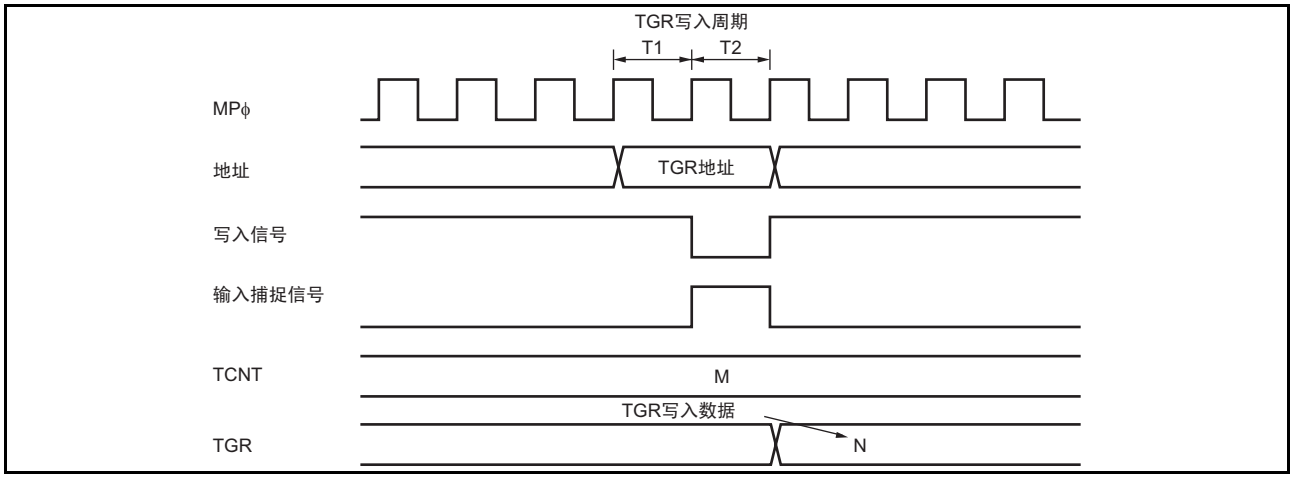


图 10.129　TGR 的写入与输入捕捉的竞争（通道 5）

10.7.11　缓冲寄存器的写入与输入捕捉的竞争

在缓冲写入周期的 T2 状态产生输入捕捉信号时，不写入缓冲寄存器，而优先缓冲运行。

此时序如图 10.130 所示。

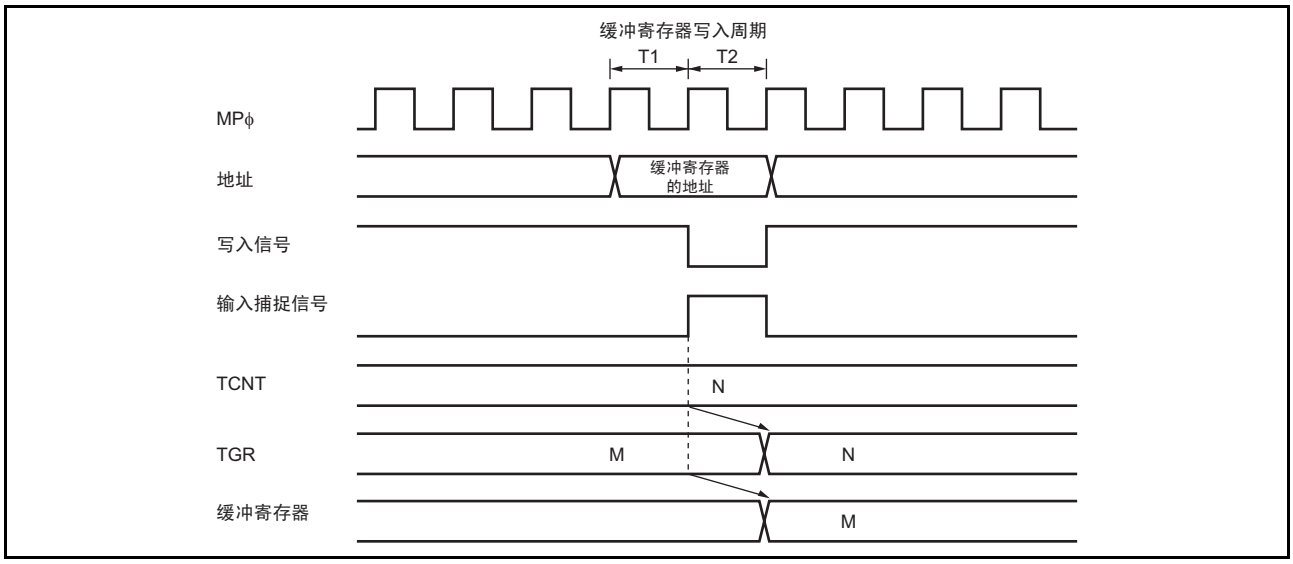


图 10.130　缓冲寄存器的写入与输入捕捉的竞争

10.7.12　级联中 TCNT_2 的写入与上溢 / 下溢的竞争

级联定时器计数器 （TCNT_1 和 TCNT_2），如果 TCNT_1 在计数的瞬间 （TCNT_2 上溢 / 下溢的瞬间）与 TCNT_2 的写入周期的 T2 状态产生竞争，则写入 TCNT_2，禁止 TCNT_1 的计数信号。此时， TGRA_1 作为比较匹配寄存器运行，与 TCNT_1 的值匹配时，产生比较匹配信号。

选择 TCNT_1 计数时钟为通道 0 的输入捕捉源时， TGRA_0 ～ D_0 执行输入捕捉。并且选择 TGRC_0 的比较匹配 / 输入捕捉为 TGRB_1 的输入捕捉源时， TGRB_1 执行输入捕捉。

此时序如图 10.131 所示。

另外，级联运行中设定清除 TCNT 时，需同步设定通道 1 和通道 2。

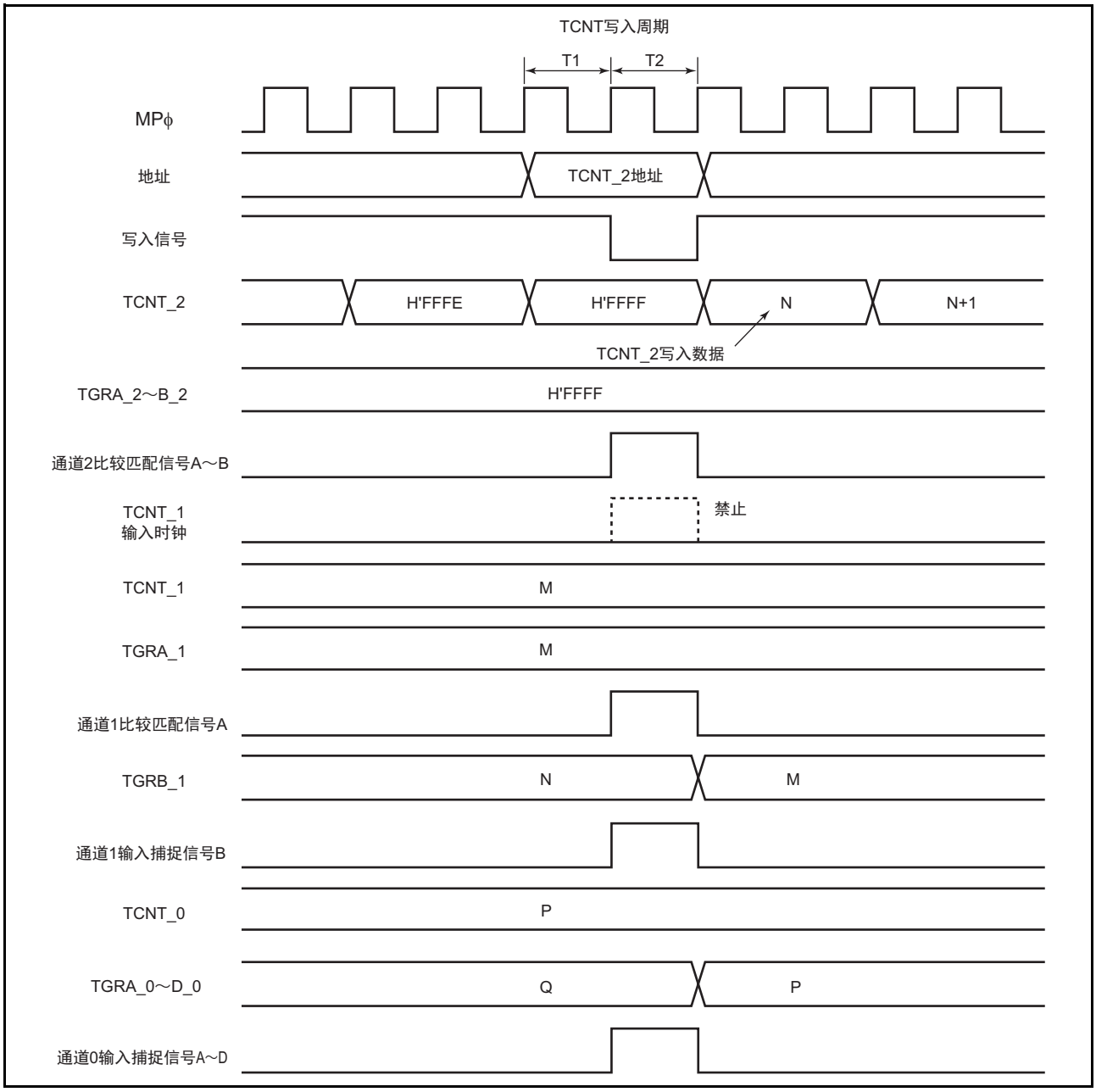


图 10.131　级联中 TCNT_2 的写入与上溢 / 下溢的竞争

10.7.13　互补 PWM 模式停止时的计数器值

在互补 PWM 模式，如果 TCNT_3 和 TCNT_4 在运行过程中停止计数， TCNT_3 则为定时器空载时间寄存器 （TDDR）的值， TCNT_4 为 H'0000。

重新开始互补 PWM 时，自动从初始状态开始计数。

此说明如图 10.132 所示。

在其他运行模式开始计数时，需给 TCNT_3、 TCNT_4 设定计数初始值。

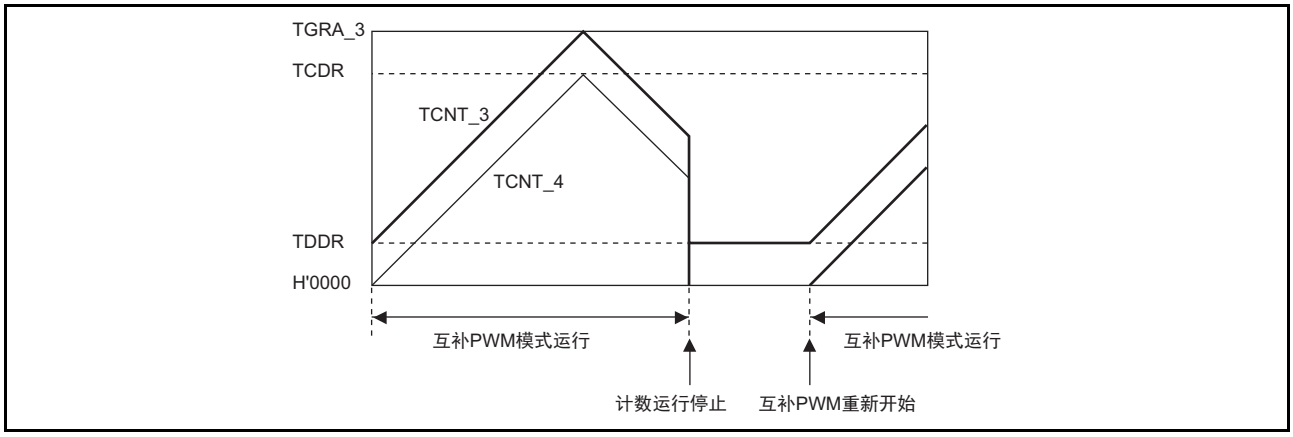


图 10.132　互补 PWM 模式停止时的计数器值

10.7.14　互补 PWM 模式中缓冲运行的设定

在互补 PWM 模式，必须通过缓冲运行改写 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）。

互补 PWM 模式的通道 3 和 4 按照 TMDR_3 的 BFA、 BFB 位的设定缓冲运行。 TMDR_3 的 BFA 位置 1 时， TGRC_3 作为 TGRA_3 缓冲寄存器运行，同时， TGRC_4 作为 TGRA_4 缓冲寄存器运行， TCBR 作为 TCDR 缓冲寄存器运行。

10.7.15　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 模式设定缓冲运行时，需将 TMDR_4 的 BFA、 BFB 位设定为 0。如果 TMDR_4 的 BFA 位置 1，则不可输出 TIOC4C 引脚的波形。

复位同步 PWM 模式中通道 3 和 4 按照 TMDR_3 的 BFA、 BFB 位的设定缓冲运行。例如，TMDR_3 的 BFA 位置 1 时， TGRC_3 作为 TGRA_3 缓冲寄存器运行，同时， TGRC_4 作为 TGRA_4 缓冲寄存器运行。

在 TGRC_3、 TGRD_3 作为缓冲寄存器运行时，TSR_3、 TSR_4 的 TGFC 位及 TGFD 位不置位。

TMDR_3 的 BFA、BFB 位置 1，TMDR_4 的 BFA、BFB 位置 0 时，TGR_3、TGR_4、TIOC3 及 TIOC4 的运行例子如图 10.133 所示。

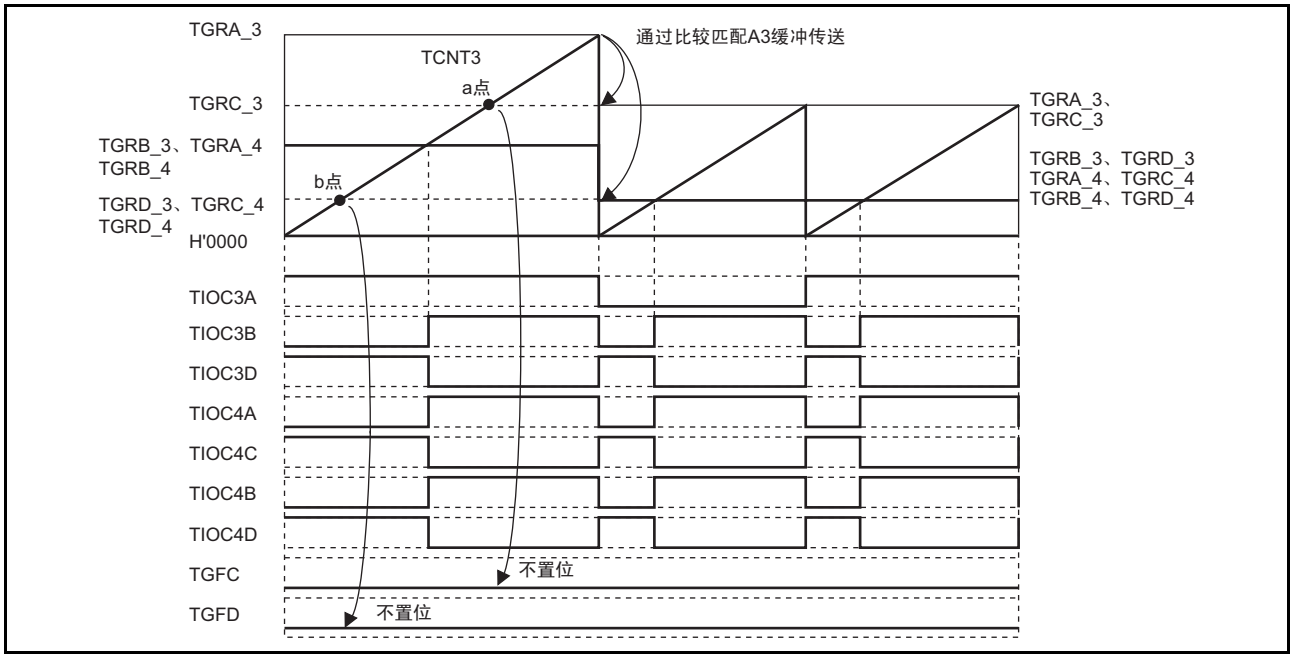


图 10.133　复位同步 PWM 模式的缓冲运行和比较匹配标志

10.7.16　复位同步 PWM 模式的上溢标志

设定复位同步 PWM 模式，如果 TSTR 的 CST3 位设定为 1，则 TCNT_3 和 TCNT_4 开始计数。此时，TCNT_4 的计数时钟源和计数边沿取决于 TCR_3 的设定。

在复位同步 PWM 模式，周期寄存器 TGRA_3 的设定值为 H'FFFF，并指定 TGRA_3 的比较匹配为计数器清除源时，TCNT_3、 TCNT_4 递增计数至 H'FFFF 时，产生与 TGRA_3 的比较匹配， TCNT_3 和 TCNT_4 同时清除计数。此时，不可置位 TSR 的上溢标志 TCFV 位。

在复位同步 PWM 模式，周期寄存器 TGRA_3 的设定值为 H'FFFF，指定 TGRA_3 的比较匹配为计数器清除源且没有设定同步时， TCFV 位的运行例子如图 10.134 所示。

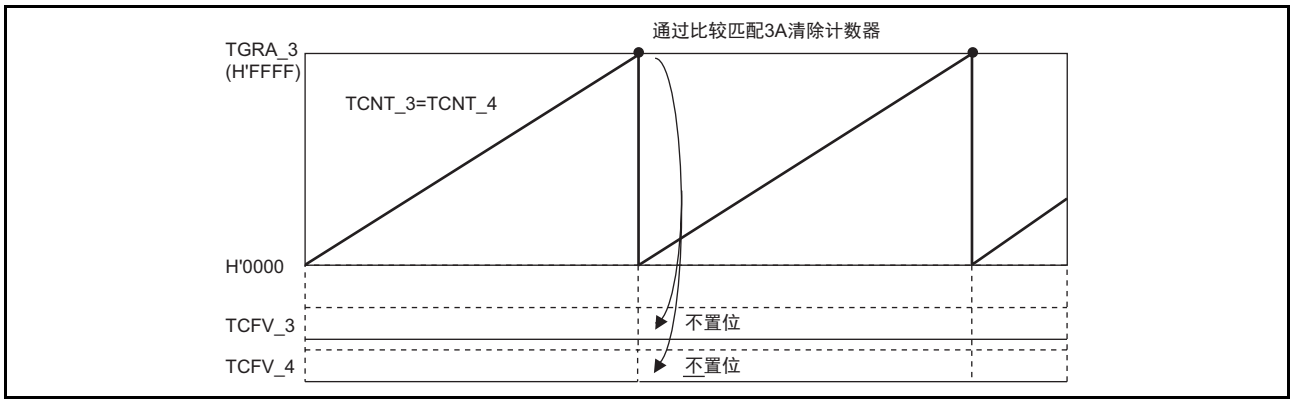


图 10.134　复位同步 PWM 模式的上溢标志

10.7.17　上溢 / 下溢与计数器清除的竞争

如果同时产生上溢 / 下溢与清除计数器， TSR 的 TCFV/TCFU 标志不置位，而优先清除 TCNT。

将 TGR 的比较匹配作为清除源，将 TGR 设定为 H'FFFF 时的运行时序如图 10.135 所示。

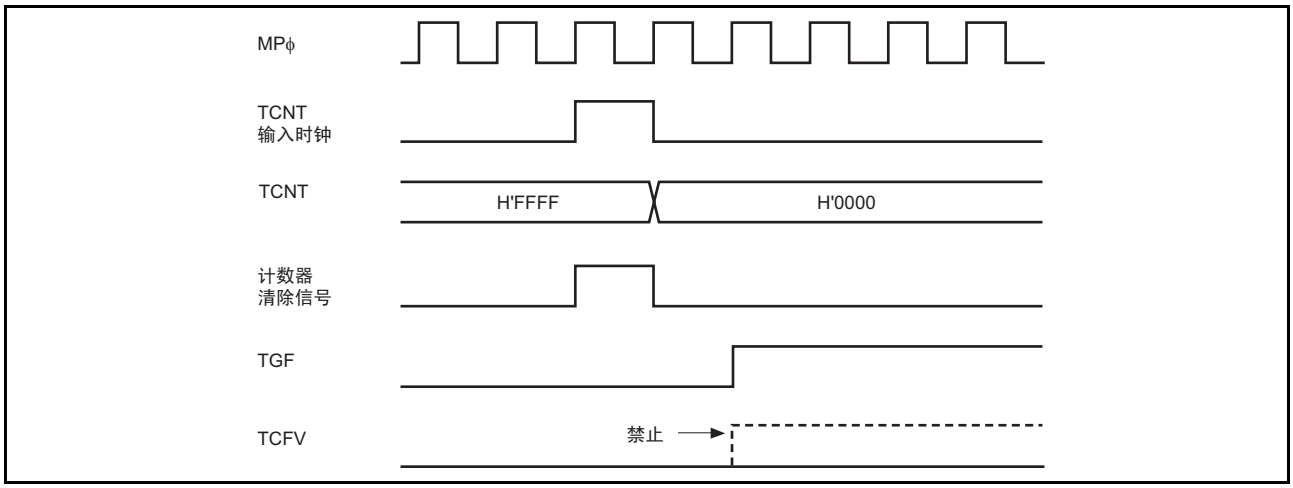


图 10.135　上溢与计数器清除的竞争

10.7.18　TCNT 的写入与上溢 / 下溢的竞争

即使在 TCNT 写入周期的 T2 状态产生递增计数 / 递减计数和上溢 / 下溢，也优先写入 TCNT， TSR 的 TCFV/TCFU 标志不置位。

TCNT 的写入与上溢竞争时的运行时序如图 10.136 所示。

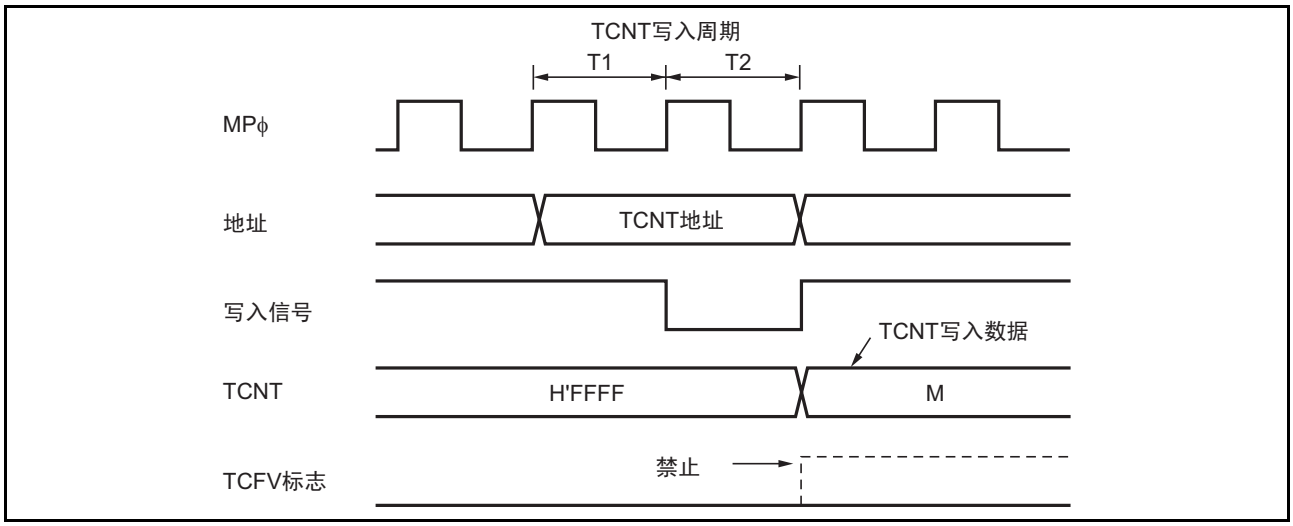


图 10.136　TCNT 的写入与上溢的竞争

10.7.19　从普通运行或 PWM 模式 1 向复位同步 PWM 模式转移时的注意事项

从通道 3、 4 的普通运行或 PWM 模式 1 向复位同步 PWM 模式转移时，输出引脚 （TIOC3B、 TIOC3D、TIOC4A、TIOC4C、TIOC4B、TIOC4D）在保持高电平的状态，停止计数器，必须注意：如果转移至复位同步 PWM 模式后使其运行，则不可正常输出引脚的初始输出。

从普通运行向复位同步 PWM 模式转移时，必须给 TIORH_3、 TIORL_3、 TIORH_4 及 TIORL_4 寄存器写入 H'11 并将输出引脚初始化为低电平后，设定寄存器的初始值 H'00 后再转移。

从 PWM 模式 1 向复位同步 PWM 模式转移时，暂且转移至普通运行，并将输出引脚初始化为低电平后，设定寄存器的初始值 H'00 后再转移至复位同步 PWM 模式。

10.7.20　互补 PWM 模式、复位同步 PWM 模式的输出电平

通道 3、 4 为互补 PWM 模式或复位同步 PWM 模式时，通过定时器输出控制寄存器 （TOCR）的 OLSP、OLSN 位设定 PWM 波形的输出电平。在互补 PWM 模式或复位同步 PWM 模式，需将 TIOR 设定为 H'00。

10.7.21　模块待机时的中断

在请求中断的状态进入模块待机时，不可清除 CPU 的中断源或 DTC 的启动源。

需预先禁止中断后，再设置模块待机模式。

10.7.22　级联中 TCNT_1 和 TCNT_2 同时输入捕捉

级联定时器计数器 1、 2 （TCNT_1 和 TCNT_2），并作为 32 位计数器运行时，即使 TIOC1A 与 TIOC2A 或 TIOC1B 与 TIOC2B 同时执行输入捕捉输入，将输入到 TCNT_1、 TCNT_2 的外部输入捕捉信号与内部时钟同步，并取入内部时，产生 TIOC1A、 TIOC2A、或 TIOC1B 与 TIOC2B 取入时序偏差，可能无法正常捕捉级联计数器值。

例如， TCNT_1 （高 16 位计数器）本应由 TCNT_2 （低 16 位计数器）的上溢捕捉递增计数值，却捕捉到递增计数前的计数值，此时，正确的是应将 TCNT_1=H'FFF1、 TCNT_2=H'0000 的值传送到 TGRA_1 与 TGRA_2 或 TGRB_1 与 TGRB_2，却误传 TCNT_1=H'FFF0、 TCNT_2=H'0000 的值。

MTU2 增加由 1 个输入捕捉输入可同时捕捉 TCNT_1 与 TCNT_2 的功能。使用本功能，不会有 TCNT_1 与 TCNT_2 的捕捉时序的偏差，就可取入 32 位计数器。详情参阅 “10.3.8 定时器输入捕捉控制寄存器（TICCR）”。

10.7.23　互补 PWM 模式中缓冲寄存器的标志位

在 MTU2、 MTU2S 的互补 PWM 模式 1 ～ 3，设定为缓冲运行的定时器通用寄存器 （TGR）产生比较匹配时，状态寄存器 （TSR）的对应位 （缓冲运行 A 时的 TGFC，缓冲运行 B 时的 TGFD）也置 1。

为抑制设定为缓冲运行的寄存器比较匹配中断请求，需将已设定为缓冲运行的 TGR 的定时器中断允许寄存器 （TIER）对应位置 0。

10.8　MTU2 输出引脚的初始化方法

10.8.1　运行模式

MTU2 有以下 6 种运行模式，任意一个模式都可输出波形。

- 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 MTU2 输出引脚的初始化方法。

10.8.2　复位开始时的运行

MTU2 的输出引脚 （TIOC*）在复位或待机模式时初始化为低电平。由引脚功能控制器 （PFC）选择 MTU2 的引脚功能，所以在设定 PFC 的同时，向端口输出 MTU2 的引脚状态。复位后立即由 PFC 选择 MTU2 的输出时，端口输出 MTU2 输出的初始状态低电平。有效电平为低电平时，会使系统运行，所以需在 MTU2 输出引脚的初始设定结束后设定 PFC。

【注】 * 通道编号 + 端口符号。

10.8.3　因运行过程中的异常等重新设定时的运行

在 MTU2 的运行过程中产生异常时，需由系统截断 MTU2 的输出。由 PFC 将引脚的输出转换为端口输出，并通过输出取反有效电平，执行截断。另外，大电流引脚可使用端口输出允许 （POE），硬性截断输出。因运行过程中的异常等重新设定引脚的初始化步骤及重新设定后在其他运行模式重新开始的步骤如下所示。

MTU2 具有之前所述的 6 种运行模式。模式转移的组合有 36 种，但也存在通道和模式的组合中不存在的转移。其一览表如**表 10.62** 所示。

使用下述符号表示：

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 10.62　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）

10.8.4　因运行过程中的异常等初始化引脚的步骤、模式转移的概要

- 根据定时器I/O控制寄存器（TIOR）的设定，转移至选择引脚输出电平的模式（Normal、PWM1、PWM2、PCM）时，通过TIOR的设定初始化引脚。
- 在PWM模式1，不向TIOC*B（TIOC*D）引脚输出波形，因此即使设定TIOR也不初始化引脚。如果要初始化，在普通模式初始化后，转移至PWM模式1。
- 在PWM模式2，不向周期寄存器的引脚输出波形，因此即使设定TIOR，也不初始化引脚。如果要初始化，在普通模式初始化后，转移至PWM模式2。
- 在普通模式或PWM模式2，TGRC、TGRD作为缓冲寄存器运行时，即使设定TIOR也不初始化缓冲寄存器的引脚。如果要初始化，解除缓冲模式并且初始化后，重新设定缓冲模式。
- 在PWM模式1，TGRC或TGRD作为缓冲寄存器运行时，即使设定TIOR也不初始化TGRC引脚。如果要初始化TGRC引脚，解除缓冲模式并且初始化后，重新设定缓冲模式。
- 根据定时器输出控制寄存器（TOCR）的设定，转移至选择引脚输出电平的模式（CPWM、RPWM）时，先转移至普通模式并通过TIOR初始化，TIOR返回初始值后，必须由定时器输出主控允许寄存器（TOER），将通道3、4设置为暂时禁止输出。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】 本项说明中的 * 包含通道编号。

按照表 10.62 的组合 No. 初始化引脚的步骤如下所示。另外，有效电平为低电平。

(1)　在普通模式运行过程中产生异常并在普通模式重新开始时的运行

在普通模式产生异常，重新设定后，在普通模式重新开始时的说明如图 10.137 所示。

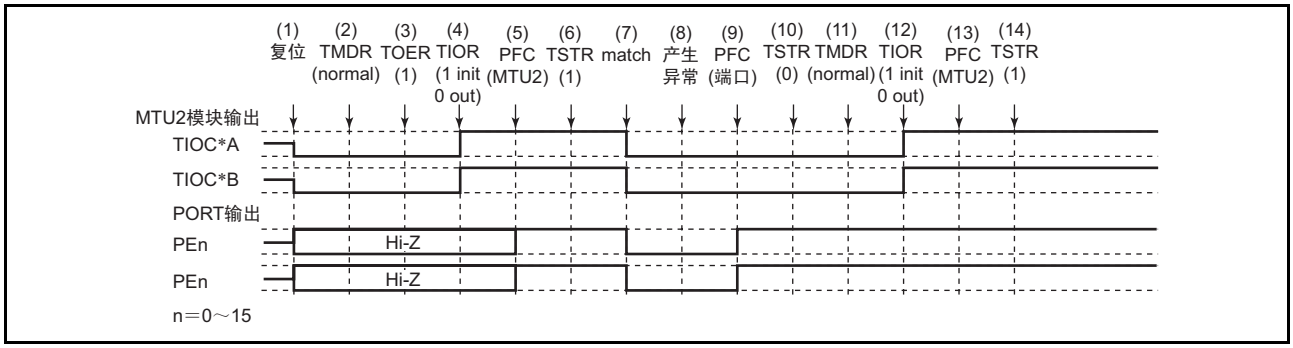


图 10.137　在普通模式产生异常并在普通模式恢复时

（1）由于复位，MTU2 输出为低电平，端口为高阻抗。
（2）由于复位，TMDR 为普通模式。
（3）在通道 3、4 由 TIOR 初始化引脚前，允许由 TOER 输出。
（4）由 TIOR 初始化引脚（例子中，初始输出为高电平，通过比较匹配输出低电平）。
（5）由 PFC 设定为 MTU2 输出。
（6）由 TSTR 开始计数。
（7）通过产生比较匹配输出低电平。
（8）产生异常。
（9）由 PFC 设定为端口输出，输出取反的有效电平。
（10）由 TSTR 停止计数。
（11）在普通模式重新开始时无需此步骤。
（12）由 TIOR 初始化引脚。
（13）由 PFC 设定为 MTU2 输出。
（14）由 TSTR 重新开始。

(2)　在普通模式运行过程中产生异常并在 PWM 模式 1 重新开始时的运行

在普通模式产生异常，重新设定后，在 PWM 模式 1 重新开始时的说明如图 10.138 所示。

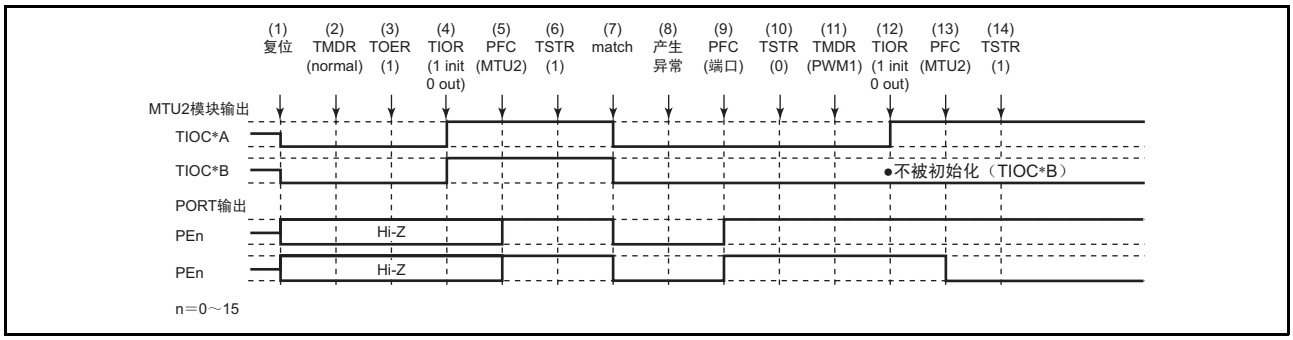


图 10.138　在普通模式产生异常并在 PWM 模式 1 恢复时

（1）～（10）和图 10.137 相同。

（11）设定 PWM 模式 1。

（12）由 TIOR 初始化引脚（在 PWM 模式 1，不初始化 TIOC*B，如果要初始化，在普通模式初始化后，转移至 PWM 模式 1）。

（13）由 PFC 设定为 MTU2 输出。

（14）由 TSTR 重新开始。

(3)　在普通模式运行过程中产生异常并在 PWM 模式 2 重新开始时的运行

在普通模式产生异常，重新设定后，在 PWM 模式 2 重新开始时的说明如图 10.139 所示。

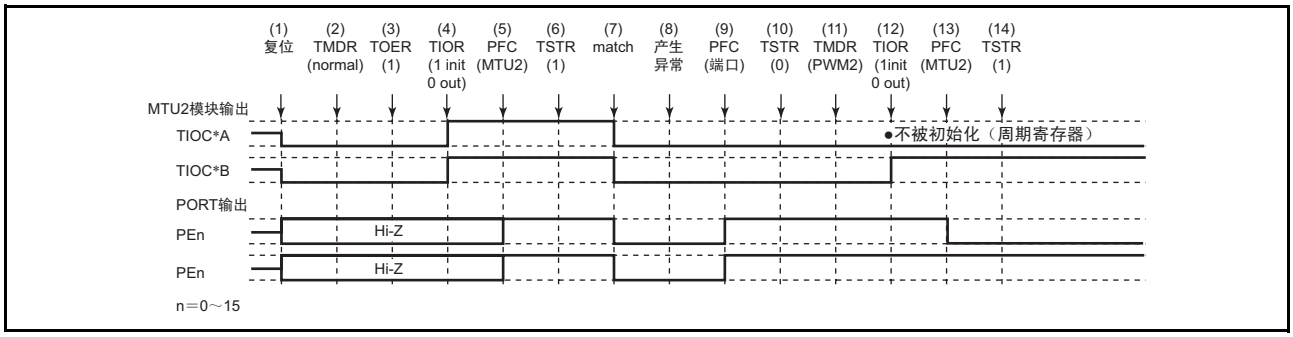


图 10.139　在普通模式产生异常并在 PWM 模式 2 恢复时

（1）～（10）和图 10.137 相同。

（11）设定 PWM 模式 2。

（12）由 TIOR 初始化引脚（在 PWM 模式 2，不初始化周期寄存器的引脚，如果要初始化时，在普通模式

初始化后，转移至 PWM 模式 2）。

（13）由 PFC 设定为 MTU2 输出。

（14）由 TSTR 重新开始。

【注】 仅可在通道 0 ～ 2 中设定 PWM 模式 2，因此无需设定 TOER。

(4)　在普通模式运行过程中产生异常并在相位计数模式重新开始时的运行

在普通模式产生异常，重新设定后，在相位计数模式重新开始时的说明如图 10.140 所示。

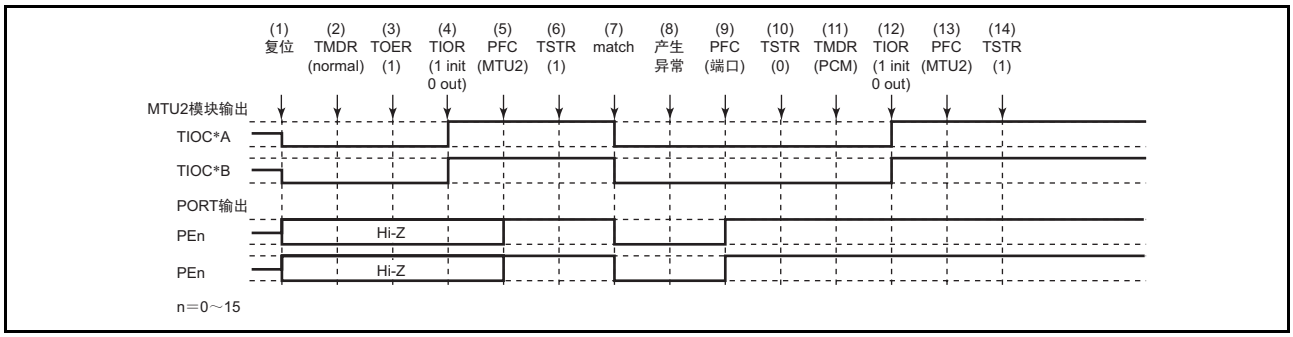


图 10.140　在普通模式产生异常并在相位计数模式恢复时

（1）～（10）和图 10.137 相同。
（11）设定相位计数模式。
（12）由 TIOR 初始化引脚。
（13）由 PFC 设定为 MTU2 输出。
（14）由 TSTR 重新开始。

【注】 仅可在通道 1、2 设定相位计数模式，因此无需设定 TOER。

(5)　在普通模式产生异常并在互补 PWM 模式重新开始时的运行

在普通模式产生异常，重新设定后，在互补 PWM 模式重新开始时的说明如图 10.141 所示。

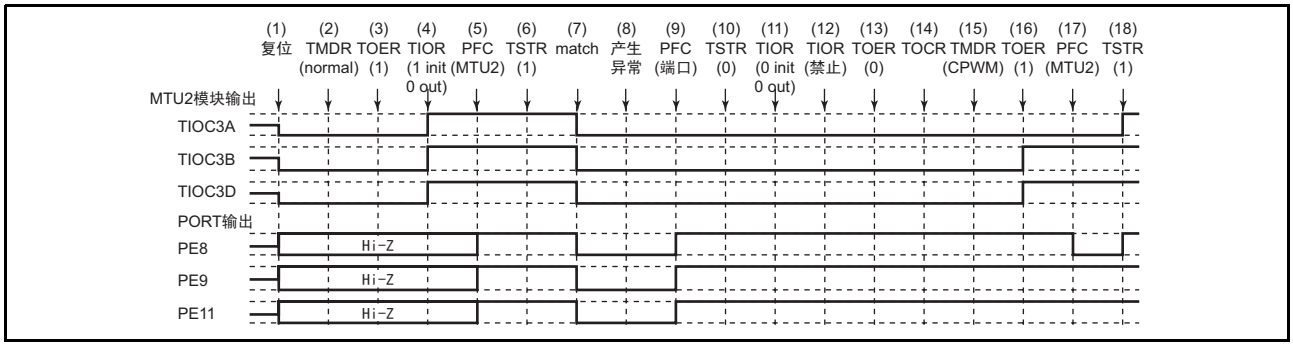


图 10.141　在普通模式产生异常并在互补 PWM 模式恢复时

（1）～（10）和图 10.137 相同。
（11）由 TIOR 初始化普通模式的波形生成部分。
（12）由 TIOR 禁止普通模式的波形生成部分的运行。
（13）由 TOER 禁止通道 3、4 的输出。
（14）由 TOCR 选择互补 PWM 的输出电平和周期输出的允许或禁止。
（15）设定互补 PWM。
（16）由 TOER 允许通道 3、4 的输出。
（17）由 PFC 设定为 MTU2 输出。
（18）由 TSTR 重新开始。

(6)　在普通模式运行过程中产生异常并在复位同步 PWM 模式重新开始时的运行

在普通模式产生异常，重新设定后，在复位同步 PWM 模式重新开始时的说明如图 10.142 所示。

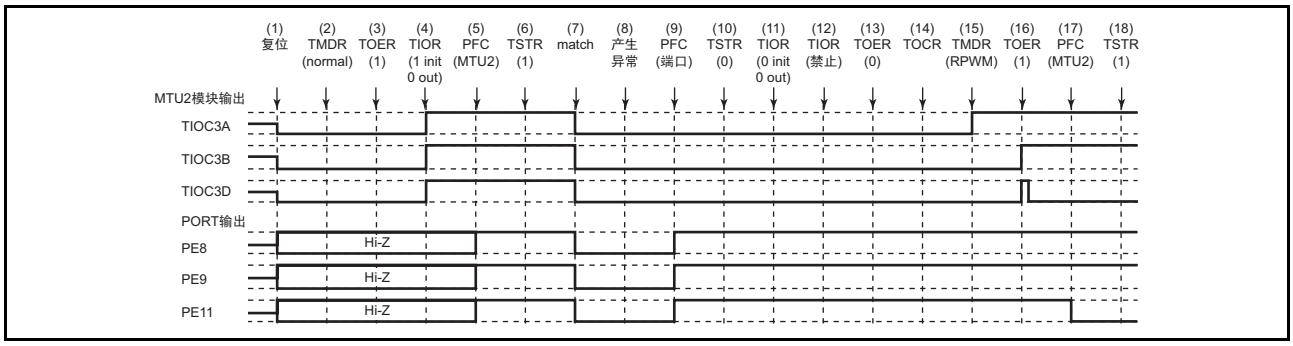


图 10.142　在普通模式产生异常并在复位同步 PWM 模式恢复时

（1）～（13）和图 10.137 相同。
（14）由 TOCR 选择复位同步 PWM 的输出电平和周期输出的允许或禁止。
（15）设定复位同步 PWM。
（16）由 TOER 允许通道 3、 4 的输出。
（17）由 PFC 设定为 MTU2 输出。
（18）由 TSTR 重新开始。

(7)　在 PWM 模式 1 运行过程中产生异常并在普通模式重新开始时的运行

在 PWM 模式 1 产生异常，重新设定后，在普通模式重新开始时的说明如图 10.143 所示。

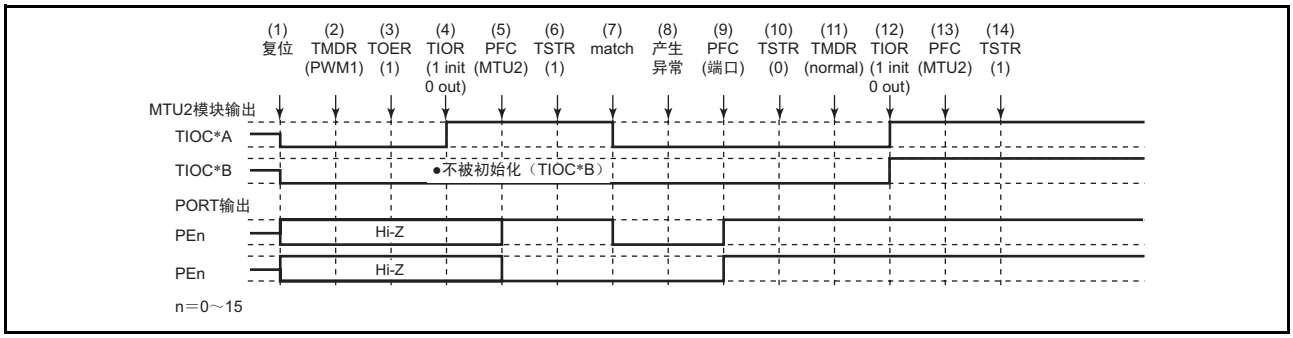


图 10.143　在 PWM 模式 1 产生异常并在普通模式恢复时

（1）由于复位， MTU2 输出为低电平，端口为高阻抗。
（2）设定 PWM 模式 1。
（3）在通道 3、 4 由 TIOR 初始化引脚前，允许由 TOER 输出。
（4）由 TIOR 初始化引脚 （例子中，初始输出为高电平，通过比较匹配输出低电平。在 PWM 模式 1 不初始化 TIOC*B）。
（5）由 PFC 设定为 MTU2 输出。
（6）由 TSTR 开始计数。
（7）通过产生比较匹配输出低电平。
（8）产生异常。
（9）由 PFC 设定为端口输出，输出取反的有效电平。
（10）由 TSTR 停止计数。
（11）设定普通模式。
（12）由 TIOR 初始化引脚。
（13）由 PFC 设定为 MTU2 输出。
（14）由 TSTR 重新开始。

(8)　在 PWM 模式 1 运行过程中产生异常并在 PWM 模式 1 重新开始时的运行

在 PWM 模式 1 产生异常，重新设定后，在 PWM 模式 1 重新开始时的说明如图 10.144 所示。

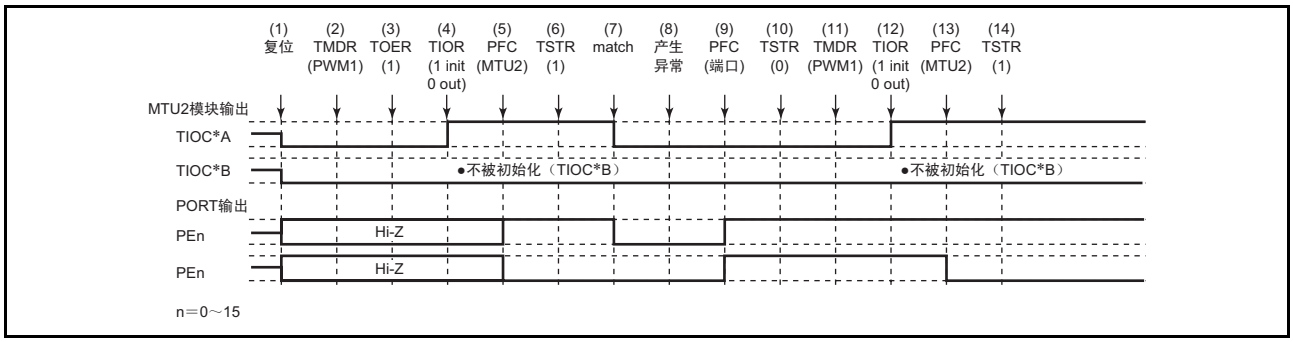


图 10.144　在 PWM 模式 1 产生异常并在 PWM 模式 1 恢复时

（1）～（10）和图 10.143 相同。
（11）在 PWM 模式 1 重新开始时无需此步骤。
（12）由 TIOR 初始化引脚（在 PWM 模式 1 不初始化 TIOC*B）。
（13）由 PFC 设定为 MTU2 输出。
（14）由 TSTR 重新开始。

(9)　在 PWM 模式 1 运行过程中产生异常并在 PWM 模式 2 重新开始时的运行

在 PWM 模式 1 产生异常，重新设定后，在 PWM 模式 2 重新开始时的说明如图 10.145 所示。

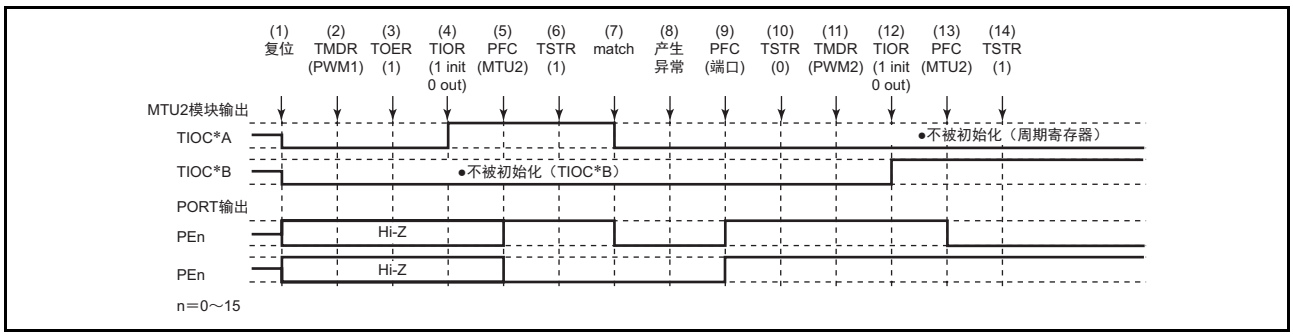


图 10.145　在 PWM 模式 1 产生异常并在 PWM 模式 2 恢复时

（1）～（10）和图 10.143 相同。
（11）设定 PWM 模式 2。
（12）由 TIOR 初始化引脚（在 PWM 模式 2 不初始化周期寄存器的引脚）。
（13）由 PFC 设定为 MTU2 输出。
（14）由 TSTR 重新开始。

【注】 仅在通道 0 ～ 2 设定 PWM 模式 2，因此无需设定 TOER。

(10) 在 PWM 模式 1 运行过程中产生异常并在相位计数模式重新开始时的运行

在 PWM 模式 1 产生异常，重新设定后，在相位计数模式重新开始时的说明如图 10.146 所示。

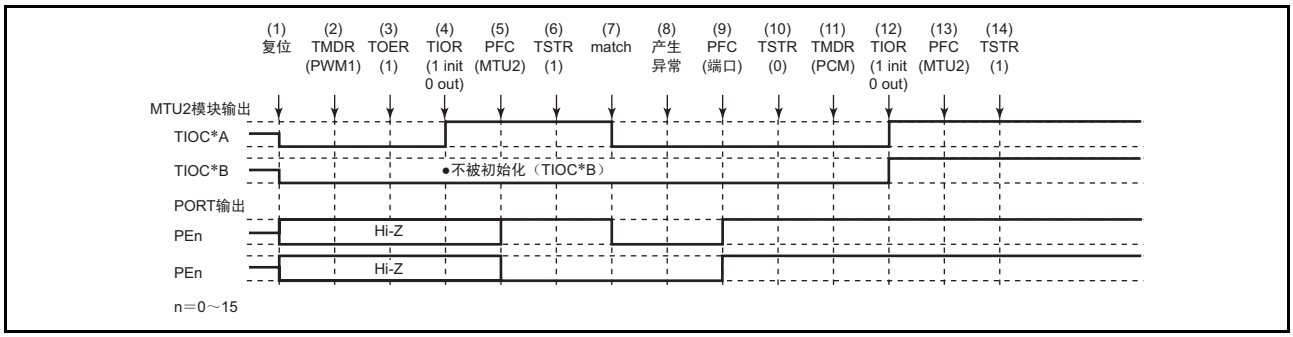


图 10.146　在 PWM 模式 1 产生异常并在相位计数模式恢复时

（1）～（10）和图 10.143 相同。
（11）设定相位计数模式。
（12）由 TIOR 初始化引脚。
（13）由 PFC 设定为 MTU2 输出。
（14）由 TSTR 重新开始。

【注】 仅可在通道 1、2 设定相位计数模式，因此无需设定 TOER。

(11) 在 PWM 模式 1 运行过程中产生异常并在互补 PWM 模式重新开始时的运行

在 PWM 模式 1 产生异常，重新设定后，以互补 PWM 模式重新开始时的说明如图 10.147 所示。

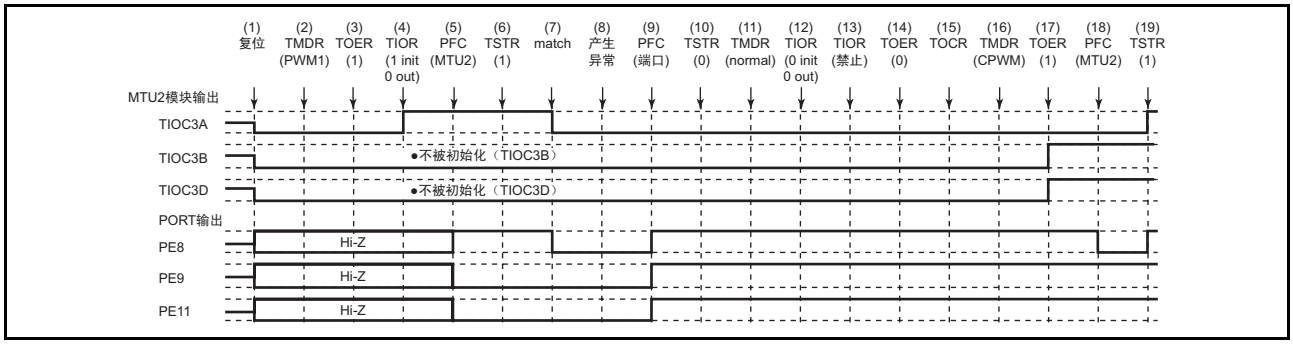


图 10.147　在 PWM 模式 1 产生异常并在互补 PWM 模式恢复时

（1）～（10）和图 10.143 相同。
（11）为了初始化波形生成部分，需设定普通模式。
（12）由 TIOR 初始化 PWM 模式 1 的波形生成部分。
（13）由 TIOR 禁止 PWM 模式 1 的波形生成部分的运行
（14）由 TOER 禁止通道 3、4 的输出。
（15）由 TOCR 选择互补 PWM 的输出电平及周期输出的允许或禁止。
（16）设定互补 PWM。
（17）由 TOER 允许通道 3、4 的输出。
（18）由 PFC 设定为 MTU2 输出。
（19）由 TSTR 重新开始。

(12) 在 PWM 模式 1 运行过程中产生异常并在复位同步 PWM 模式重新开始时的运行

在 PWM 模式 1 产生异常，重新设定后，在复位同步 PWM 模式重新开始时的说明如图 10.148 所示。

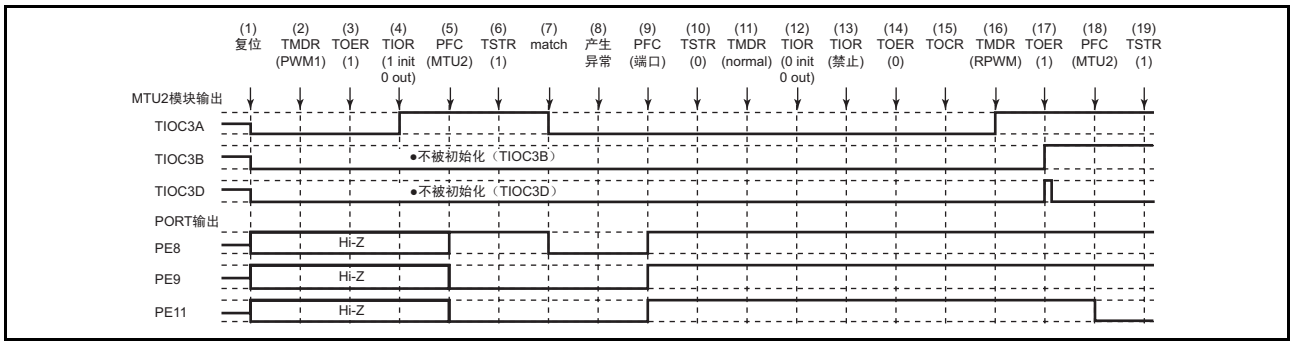


图 10.148　在 PWM 模式 1 产生异常并在复位同步 PWM 模式恢复时

（1）～（14）和图 10.147 相同。

（15）由 TOCR 选择复位同步 PWM 的输出电平和周期输出的允许或禁止。

（16）设定复位同步 PWM。

（17）由 TOER 允许通道 3、 4 的输出。

（18）由 PFC 设定为 MTU2 输出。

（19）由 TSTR 重新开始。

(13) 在 PWM 模式 2 运行过程中产生异常并在普通模式重新开始时的运行

在 PWM 模式 2 产生异常，重新设定后，在普通模式重新开始时的说明如图 10.149 所示。

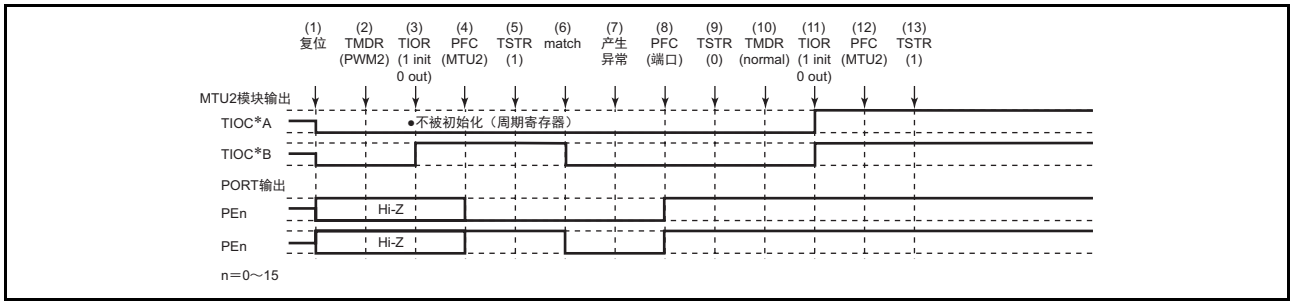


图 10.149　在 PWM 模式 2 产生异常并在普通模式恢复时

（1）由于复位， MTU2 输出为低电平，端口为高阻抗。

（2）设定 PWM 模式 2。

（3）由 TIOR 初始化引脚 （例子中，初始输出为高电平，通过比较匹配输出低电平。在 PWM 模式 2 不初始化周期寄存器的引脚。此例子中 TIOC*A 为周期寄存器）。

（4）由 PFC 设定为 MTU2 输出。

（5）由 TSTR 开始计数。

（6）通过产生比较匹配输出低电平。

（7）产生异常。

（8）由 PFC 设定为端口输出，输出取反的有效电平。

（9）由 TSTR 停止计数。

（10）设定普通模式。

（11）由 TIOR 初始化引脚。

（12）由 PFC 设定为 MTU2 输出。

（13）由 TSTR 重新开始。

(14) 在 PWM 模式 2 运行过程中产生异常并在 PWM 模式 1 重新开始时的运行

在 PWM 模式 2 产生异常，重新设定后，在 PWM 模式 1 重新开始时的说明如图 10.150 所示。

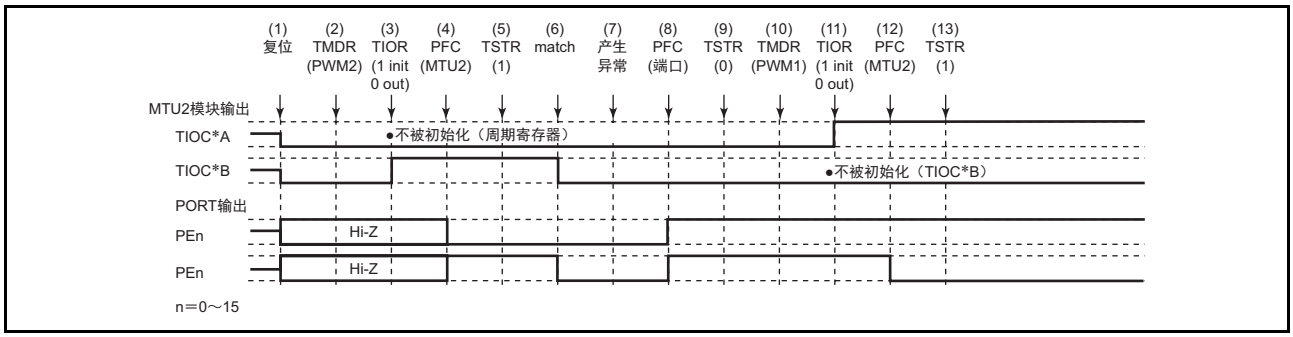


图 10.150　在 PWM 模式 2 产生异常并在 PWM 模式 1 恢复时

（1）～（9）和图 10.149 相同。

（10）设定 PWM 模式 1。

（11）由 TIOR 初始化引脚（在 PWM 模式 1 不初始化 TIOC*B）。

（12）由 PFC 设定为 MTU2 输出。

（13）由 TSTR 重新开始。

(15) 在 PWM 模式 2 运行过程中产生异常并在 PWM 模式 2 重新开始时的运行

在 PWM 模式 2 产生异常，重新设定后，在 PWM 模式 2 重新开始时的说明如图 10.151 所示。

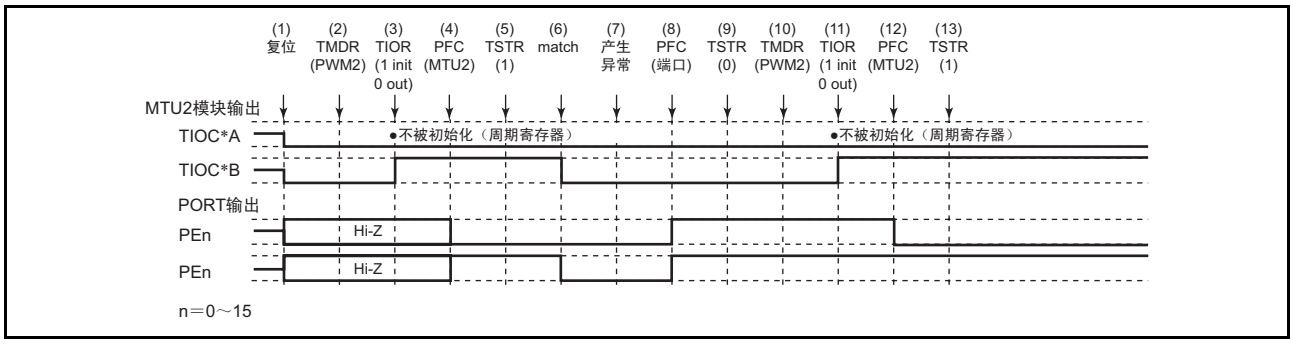


图 10.151　在 PWM 模式 2 产生异常并在 PWM 模式 2 恢复时

（1）～（9）和图 10.149 相同。

（10）在 PWM 模式 2 重新开始时无需此步骤。

（11）由 TIOR 初始化引脚（在 PWM 模式 2 不初始化周期寄存器的引脚）。

（12）由 PFC 设定为 MTU2 输出。

（13）由 TSTR 重新开始。

(16) 在 PWM 模式 2 运行过程中产生异常并在相位计数模式重新开始时的运行

在 PWM 模式 2 产生异常，重新设定后，在相位计数模式重新开始时的说明如图 10.152 所示。

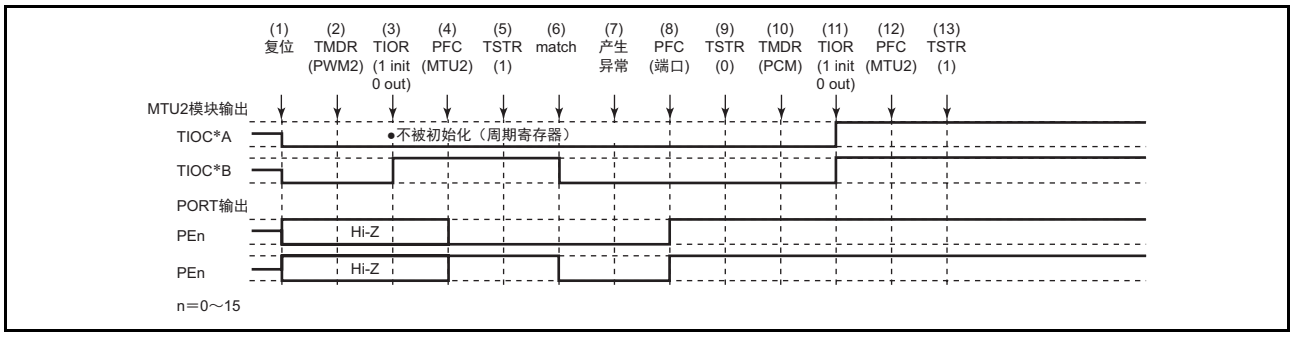


图 10.152　在 PWM 模式 2 产生异常并在相位计数模式恢复时

（1）～（9）和图 10.149 相同。
（10）设定相位计数模式。
（11）由 TIOR 初始化引脚。
（12）由 PFC 设定为 MTU2 输出。
（13）由 TSTR 重新开始。

(17) 在相位计数模式运行过程中产生异常并在普通模式重新开始时的运行

在相位计数模式产生异常，重新设定后，在普通模式重新开始时的说明如图 10.153 所示。

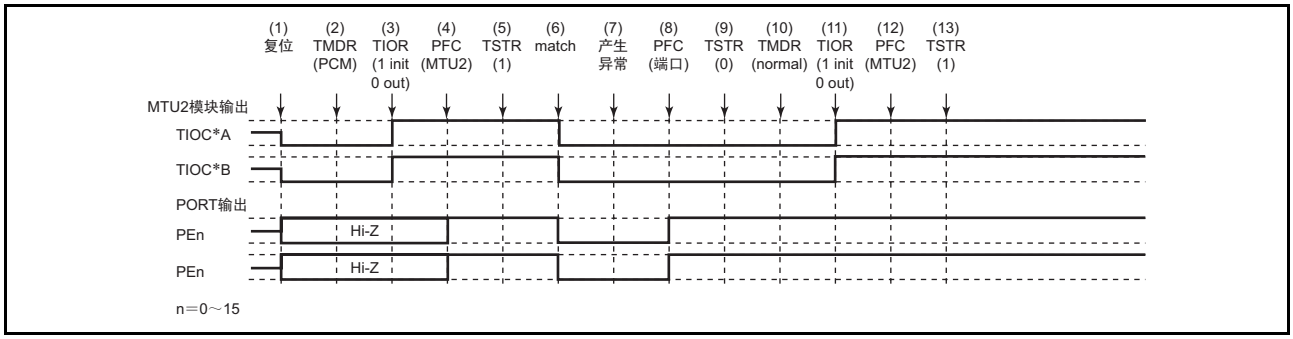


图 10.153　在相位计数模式产生异常并在普通模式恢复时

（1）由于复位， MTU2 输出为低电平，端口为高阻抗。
（2）设定相位计数模式。
（3）由 TIOR 初始化引脚 （例子中，初始输出为高电平，通过比较匹配输出低电平）。
（4）由 PFC 设定为 MTU2 输出。
（5）由 TSTR 开始计数。
（6）通过产生比较匹配输出低电平。
（7）产生异常。
（8）由 PFC 设定为端口输出，输出取反的有效电平。
（9）由 TSTR 停止计数。
（10）在普通模式设定。
（11）由 TIOR 初始化引脚。
（12）由 PFC 设定为 MTU2 输出。
（13）由 TSTR 重新开始。

(18) 在相位计数模式运行过程中产生异常并在 PWM 模式 1 重新开始时的运行

在相位计数模式产生异常，重新设定后，在 PWM 模式 1 重新开始时的说明如图 10.154 所示。

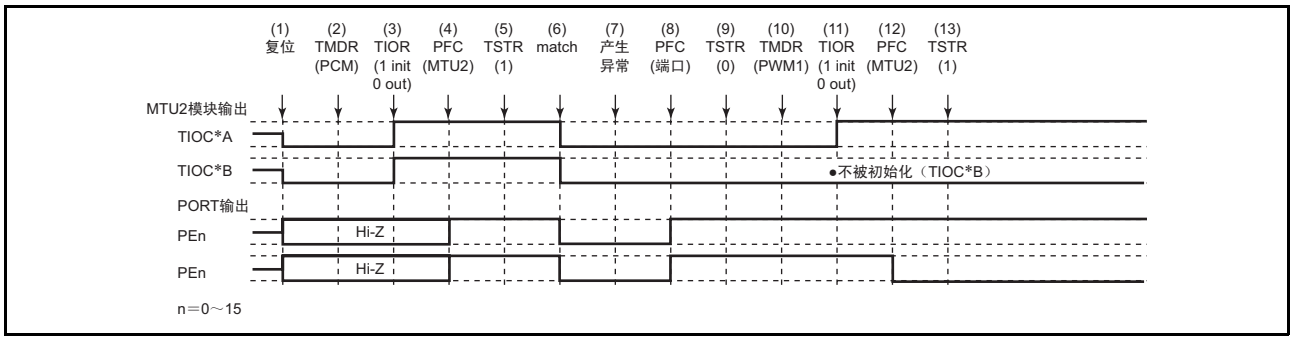


图 10.154　在相位计数模式产生异常并在 PWM 模式 1 恢复时

（1）～（9）和图 10.153 相同。
（10）设定 PWM 模式 1。
（11）由 TIOR 初始化引脚（在 PWM 模式 1 不初始化 TIOC*B）。
（12）由 PFC 设定为 MTU2 输出。
（13）由 TSTR 重新开始。

(19) 在相位计数模式运行过程中产生异常并在 PWM 模式 2 重新开始时的运行

在相位计数模式中产生异常，重新设定后，在 PWM2 模式重新开始时的说明如图 10.155 所示。

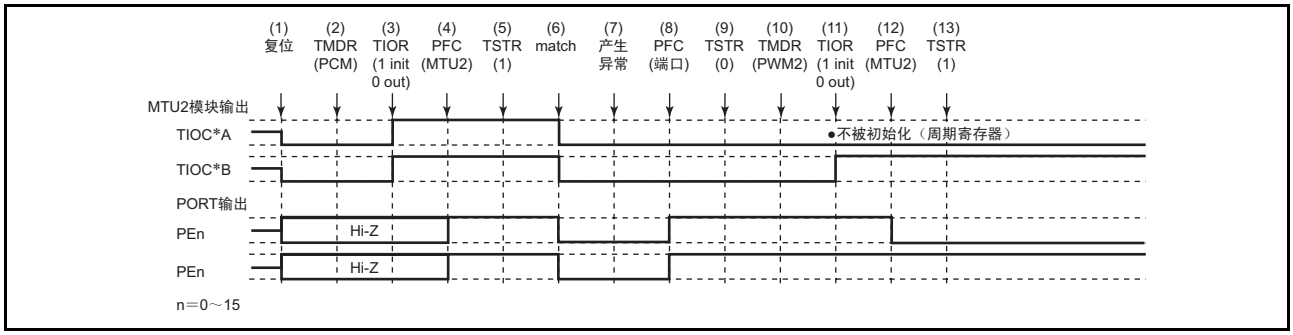


图 10.155　在相位计数模式产生异常并在 PWM 模式 2 恢复时

（1）～（9）和图 10.153 相同。
（10）设定 PWM 模式 2。
（11）由 TIOR 初始化引脚（在 PWM 模式 2 不初始化周期寄存器的引脚）。
（12）由 PFC 设定为 MTU2 输出。
（13）由 TSTR 重新开始。

(20) 在相位计数模式运行过程中产生异常并在相位计数模式重新开始时的运行

在相位计数模式产生异常，重新设定后，在相位计数模式重新开始时的说明如图 10.156 所示。

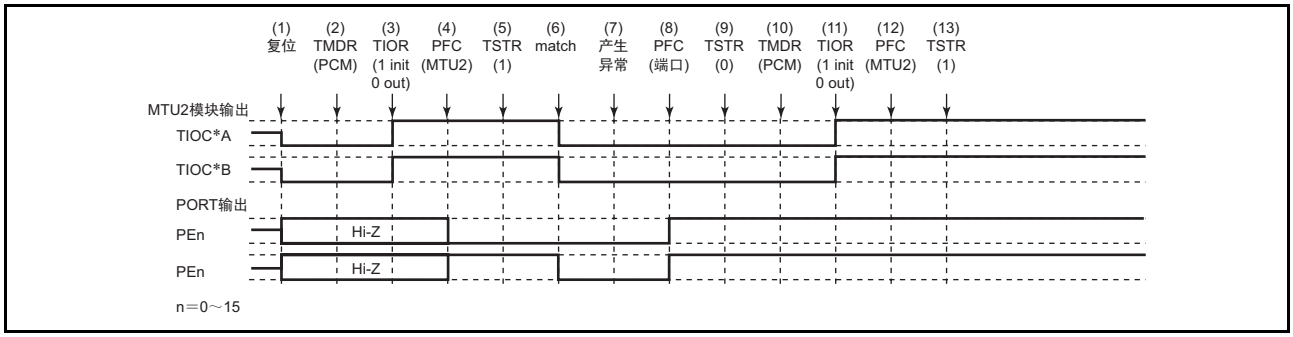


图 10.156　在相位计数模式产生异常并在相位计数模式恢复时

（1）～（9）和图 10.153 相同。
（10）在相位计数模式重新开始时，无需此步骤。
（11）由 TIOR 初始化引脚。
（12）由 PFC 设定为 MTU2 输出。
（13）由 TSTR 重新开始。

(21) 在互补 PWM 模式运行过程中产生异常并在普通模式重新开始时的运行

在互补 PWM 模式产生异常，重新设定后，在普通模式重新开始时的说明如图 10.157 所示。

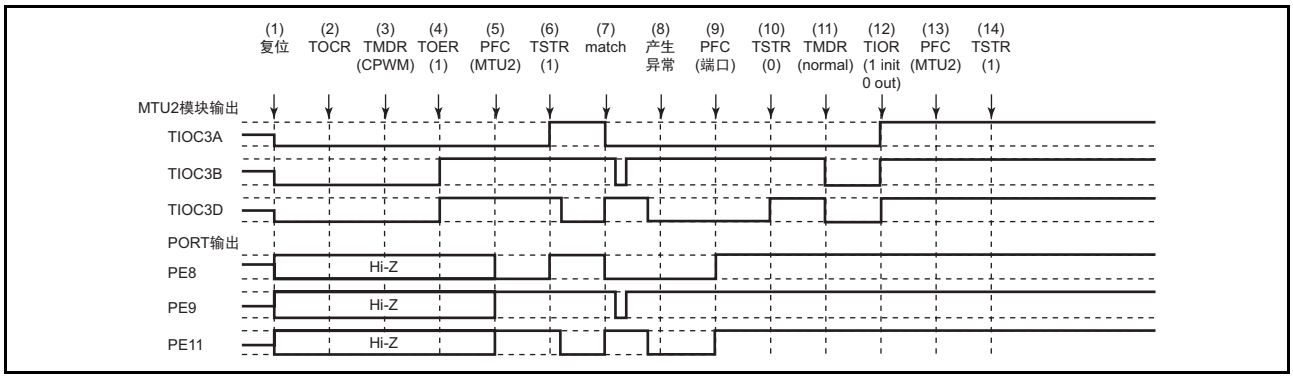


图 10.157　在互补 PWM 模式产生异常并在普通模式恢复时

（1）由于复位， MTU2 输出为低电平，端口为高阻抗。
（2）由 TOCR 选择互补 PWM 的输出电平及周期输出的允许或禁止。
（3）设定互补 PWM。
（4）由 TOER 允许通道 3、 4 的输出。
（5）由 PFC 设定为 MTU2 输出。
（6）由 TSTR 开始计数。
（7）通过产生比较匹配输出互补 PWM 波形。
（8）产生异常。
（9）由 PFC 设定为端口输出，输出取反的有效电平。
（10）由 TSTR 停止计数 （MTU2 输出为互补 PWM 输出初始值）。
（11）设定普通模式 （MTU2 输出为低电平）。
（12）由 TIOR 初始化引脚。
（13）由 PFC 设定为 MTU2 输出。
（14）由 TSTR 重新开始。

(22) 在互补 PWM 模式运行过程中产生异常并在 PWM 模式 1 重新开始时的运行

在互补 PWM 模式产生异常，重新设定后，在 PWM 模式 1 重新开始时的说明如图 10.158 所示。

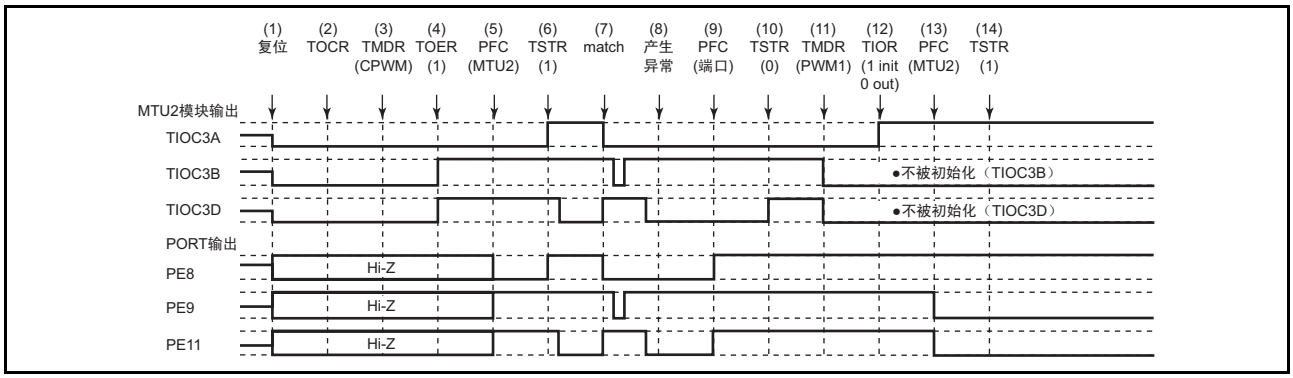


图 10.158　在互补 PWM 模式产生异常并在 PWM 模式 1 恢复时

（1）～（10）和图 10.157 相同。

（11）设定 PWM 模式 1（MTU2 输出为低电平）。

（12）由 TIOR 初始化引脚（在 PWM 模式 1 不初始化 TIOC*B）。

（13）由 PFC 设定为 MTU2 输出。

（14）由 TSTR 重新开始。

(23) 在互补 PWM 模式运行过程中产生异常并在互补 PWM 模式重新开始时的运行

在互补 PWM 模式产生异常，重新设定后，在互补 PWM 模式重新开始时的说明如图 10.159 所示（重新开始的周期、占空比设定与计数器停止时的值相同时的例子）。

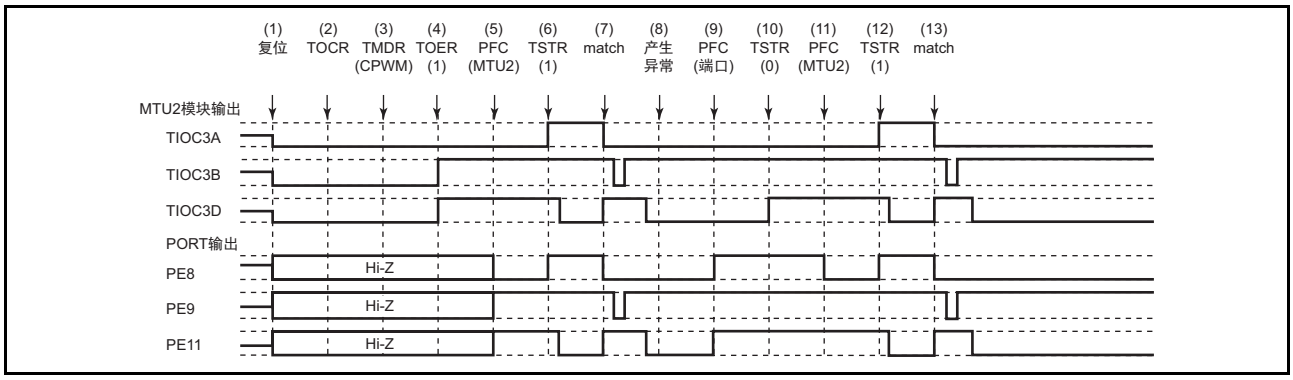


图 10.159　在互补 PWM 模式产生异常并在互补 PWM 模式恢复时

（1）～（10）和图 10.157 相同。

（11）由 PFC 设定为 MTU2 输出。

（12）由 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 模式运行过程中产生异常并在互补 PWM 模式重新开始时的运行

在互补 PWM 模式产生异常，重新设定后，在互补 PWM 模式重新开始时的说明如图 10.160 所示（用新的周期、占空比设定值重新开始）。

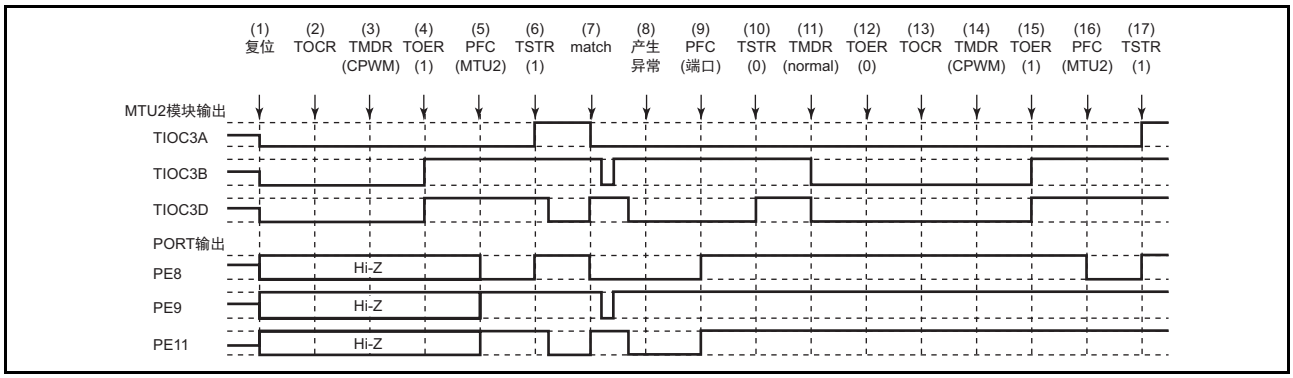


图 10.160　在互补 PWM 模式产生异常并在互补 PWM 模式恢复时

（1）～（10）和图 10.157 相同。
（11）设定普通模式和新的设定值（MTU2 输出为低电平）。
（12）由 TOER 禁止通道 3、4 的输出。
（13）由 TOCR 选择互补 PWM 模式的输出电平和周期输出的允许或禁止。
（14）设定互补 PWM。
（15）由 TOER 允许通道 3、4 的输出。
（16）由 PFC 设定为 MTU2 输出。
（17）由 TSTR 重新开始。

(25) 在互补 PWM 模式运行过程中产生异常并在复位同步 PWM 模式重新开始时的运行

在互补 PWM 模式产生异常，重新设定后，在复位同步 PWM 模式重新开始时的说明如图 10.161 所示。

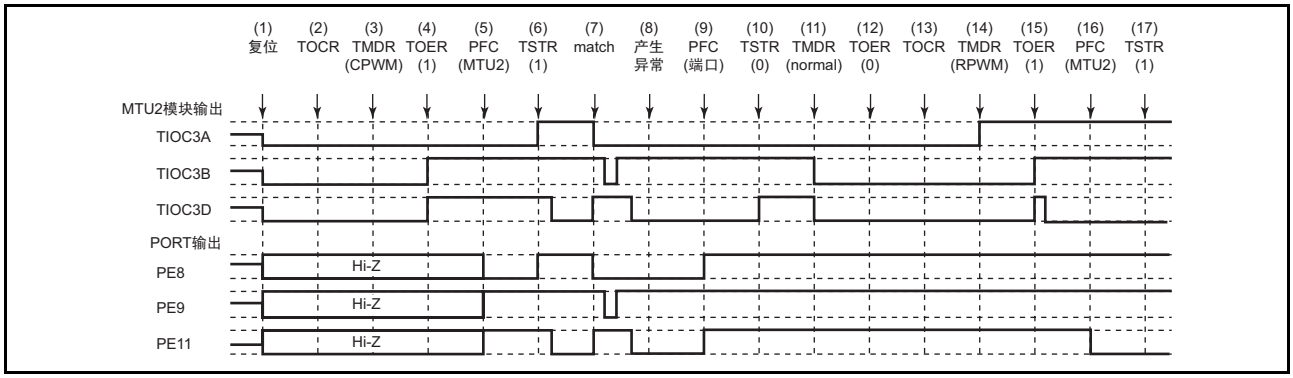


图 10.161　在互补 PWM 模式产生异常并在复位同步 PWM 模式恢复时

（1）～（10）和图 10.157 相同。
（11）设定普通模式（MTU2 输出为低电平）。
（12）由 TOER 禁止通道 3、4 的输出。
（13）由 TOCR 选择复位同步 PWM 模式的输出电平和周期输出的允许或禁止。
（14）设定复位同步 PWM。
（15）由 TOER 允许通道 3、4 的输出。
（16）由 PFC 设定为 MTU2 输出。
（17）由 TSTR 重新开始。

(26) 在复位同步 PWM 模式运行过程中产生异常并在普通模式重新开始时的运行

在复位同步 PWM 模式产生异常，重新设定后，在普通模式重新开始时的说明如图 10.162 所示。

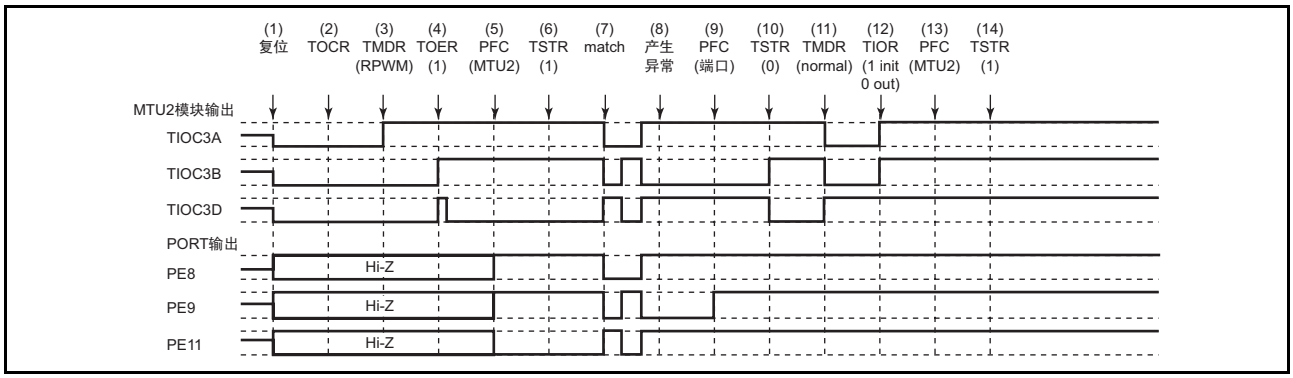


图 10.162　在复位同步 PWM 模式产生异常并在普通模式恢复时

（1）由于复位， MTU2 输出为低电平，端口为高阻抗。
（2）由 TOCR 选择复位同步 PWM 的输出电平和周期输出的允许或禁止。
（3）设定复位同步 PWM。
（4）由 TOER 允许通道 3、 4 的输出。
（5）由 PFC 设定为 MTU2 输出。
（6）由 TSTR 开始计数。
（7）通过产生比较匹配输出复位同步 PWM 波形。
（8）产生异常。
（9）由 PFC 设定为端口输出，输出取反的有效电平。
（10）由 TSTR 停止计数 （MTU2 输出为复位同步 PWM 输出初始值）。
（11）设定普通模式 （MTU2 输出的正相为低电平，反相为高电平）。
（12）由 TIOR 初始化引脚。
（13）由 PFC 设定为 MTU2 输出。
（14）由 TSTR 重新开始。

(27) 在复位同步 PWM 模式运行过程中产生异常并在 PWM 模式 1 重新开始时的运行

在复位同步 PWM 模式产生异常，重新设定后，在 PWM 模式 1 重新开始时的说明如图 10.163 所示。

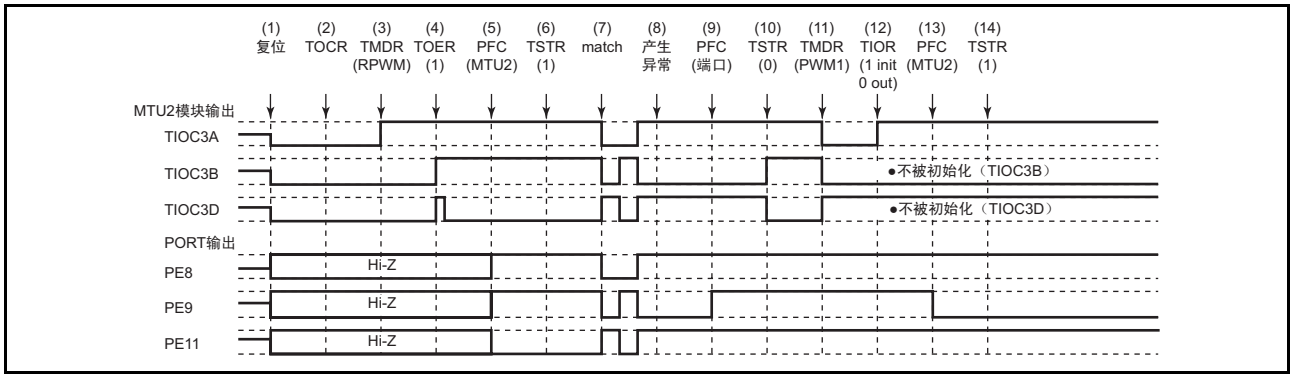


图 10.163　在复位同步 PWM 模式产生异常并在 PWM 模式 1 恢复时

（1）～（10）和图 10.162 相同。
（11）设定 PWM 模式 1 （MTU2 输出的正相为低电平，反相为高电平）。
（12）由 TIOR 初始化引脚 （在 PWM 模式 1 不初始化 TIOC*B）。
（13）由 PFC 设定为 MTU2 输出。
（14）由 TSTR 重新开始。

(28) 在复位同步 PWM 模式运行过程中产生异常并在互补 PWM 模式重新开始时的运行

在复位同步 PWM 模式产生异常，重新设定后，在互补 PWM 模式重新开始时的说明如图 10.164 所示。

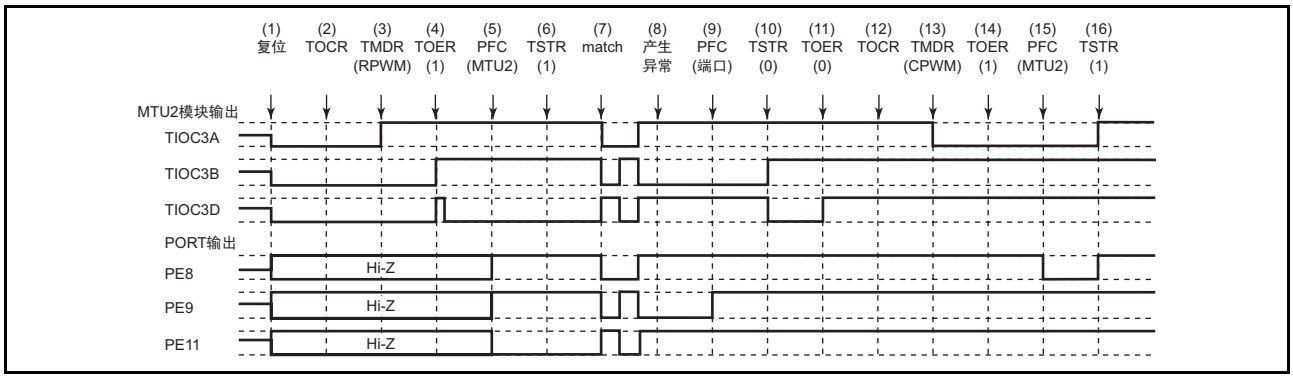


图 10.164　在复位同步 PWM 模式产生异常并在互补 PWM 模式恢复时

（1）～（10）和图 10.162 相同。
（11）由 TOER 禁止通道 3、 4 的输出。
（12）由 TOCR 选择互补 PWM 的输出电平和周期输出的允许或禁止。
（13）设定互补 PWM （MTU2 的周期输出引脚为低电平）。
（14）由 TOER 允许通道 3、 4 的输出。
（15）由 PFC 设定为 MTU2 输出。
（16）由 TSTR 重新开始。

(29) 在复位同步 PWM 模式运行过程中产生异常并在复位同步 PWM 模式重新开始时的运行

在复位同步 PWM 模式产生异常，重新设定后，在复位同步 PWM 模式重新开始时的说明如图 10.165 所示。

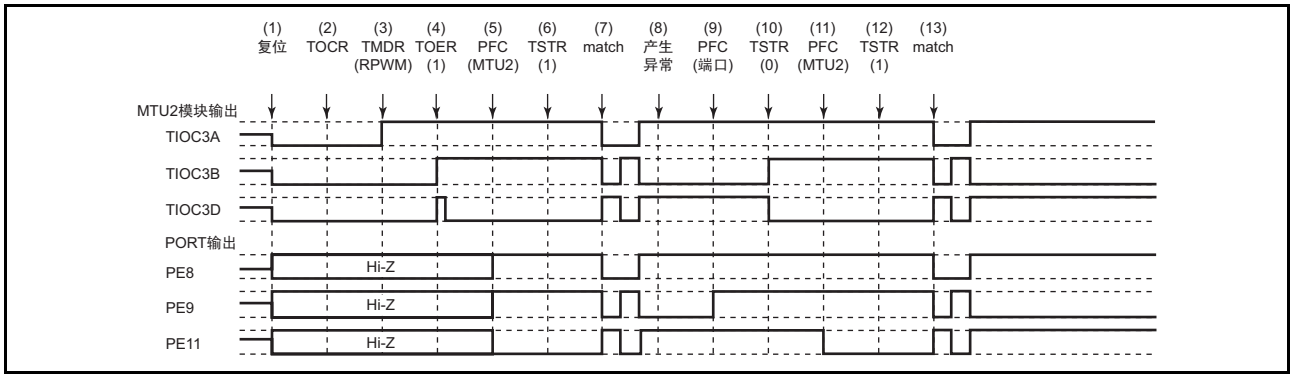


图 10.165　在复位同步 PWM 模式产生异常并在复位同步 PWM 模式恢复时

（1）～（10）和图 10.162 相同。
（11）由 PFC 设定为 MTU2 输出。
（12）由 TSTR 重新开始。
（13）通过产生比较匹配输出复位同步 PWM 波形。

第 11 章 端口输出允许 （POE）

端口输出允许 （POE）根据 $\overline{\text{POE0}}$ ～ $\overline{\text{POE2}}$、 $\overline{\text{POE4}}$ ～ $\overline{\text{POE6}}$、 $\overline{\text{POE8}}$ 引脚的输入变化、 MTU2、 2S 特殊引脚 （MTU2 的 TIOC3B、 TIOC3D、 TIOC4A、 TIOC4B、 TIOC4C、 TIOC4D 及 MTU2S 的 TIOC3BS、 TIOC3DS、 TIOC4AS、 TIOC4BS、 TIOC4CS、 TIOC4DS 为多路复用引脚）的输出状态或寄存器的设定，可将 MTU2、 2S 特殊引脚及 MTU2 的 CH0 引脚 （TIOC0A、 TIOC0B、 TIOC0C、 TIOC0D 为多路复用引脚）设置为高阻抗状态，并可同时发行中断请求。

11.1 特点

- 可给 $\overline{\text{POE0}}$～$\overline{\text{POE2}}$、 $\overline{\text{POE4}}$～$\overline{\text{POE6}}$、 $\overline{\text{POE8}}$ 等各输入引脚设定下降沿和 Pϕ/8×16 次、 Pϕ/16×16 次、 Pϕ/128×16 次的低电平采样。
- 通过 $\overline{\text{POE0}}$～$\overline{\text{POE2}}$、 $\overline{\text{POE4}}$～$\overline{\text{POE6}}$、 $\overline{\text{POE8}}$ 引脚的下降沿或低电平采样，可将 MTU2、 2S 特殊引脚 MTU2 的 CH0 引脚设置为高阻抗状态。
- 比较 MTU2、 2S 特殊引脚的输出电平，同时如果有效电平输出持续 1 个 Pϕ 周期以上，则可将 MTU2、 2S
  特殊引脚设置为高阻抗状态。
- 通过写入 POE 的寄存器，可将 MTU2、 2S 特殊引脚及 MTU2 的 CH0 引脚设置为高阻抗状态。
- 根据输入电平的采样或输出电平的比较结果，可分别产生中断。

如图 11.1 的框图所示， POE 由输入电平检测电路、输出电平比较电路及高阻抗请求 / 中断请求生成电路构成。

除此之外，即使在振荡器停止和软件待机状态，也可将 MTU2、 2S 特殊引脚设置为高阻抗状态。详情参阅 “附录 A. 引脚状态”。

POE 框图如图 11.1 所示。

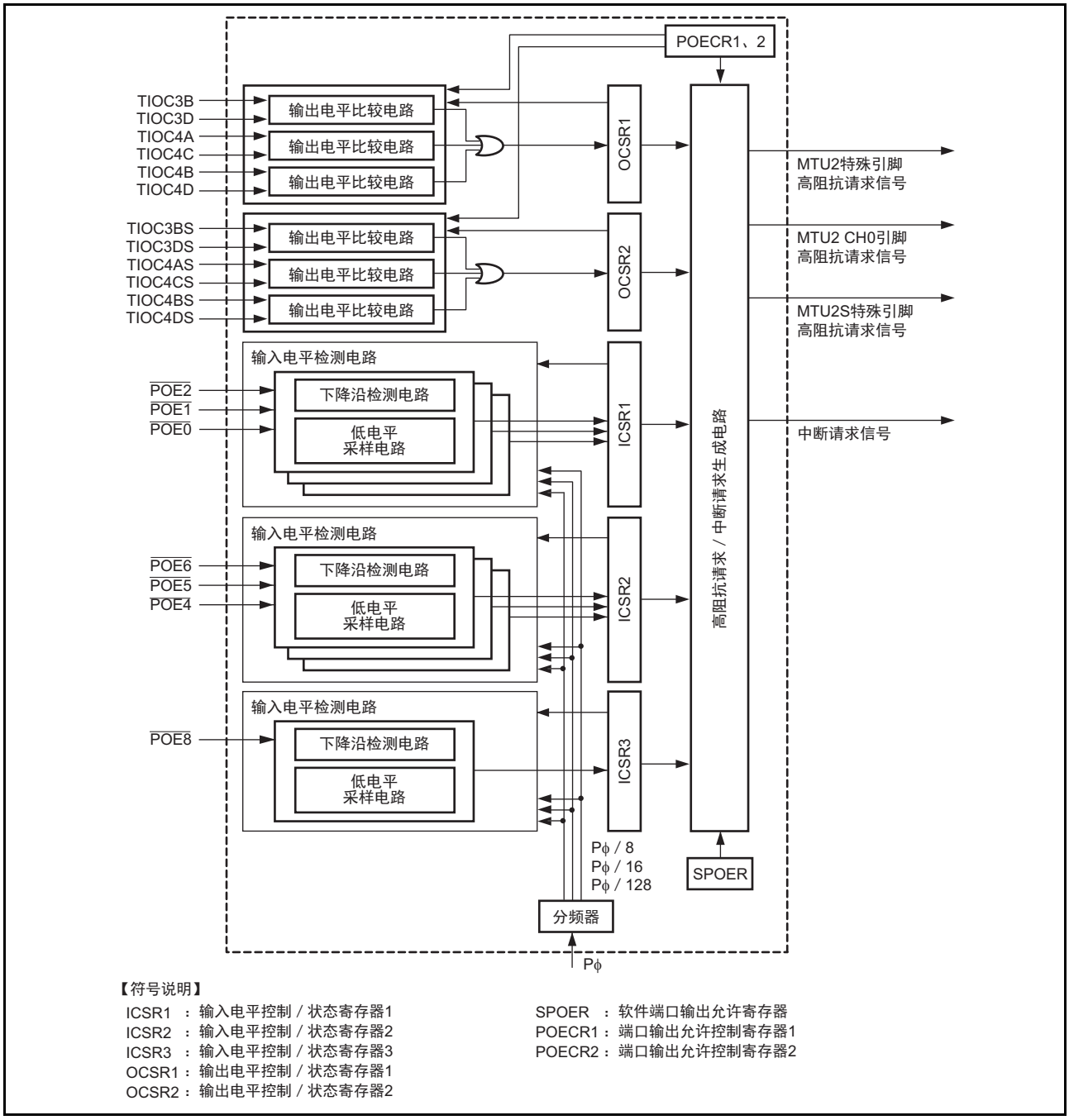


图 11.1　POE 框图

11.2　输入 / 输出引脚

表 11.1　引脚结构

名称	引脚名称	输入 / 输出	功能
端口输出允许 的输入引脚 0 ～ 2	$\overline{\text{POE0}}$ ～ $\overline{\text{POE2}}$	输入	输入将 MTU2 特殊引脚设置为高阻抗状态的请求信号
端口输出允许 的输入引脚 4 ～ 6	$\overline{\text{POE4}}$ ～ $\overline{\text{POE6}}$	输入	输入将 MTU2S 特殊引脚设置为高阻抗状态的请求信号
端口输出允许的输入引脚 8	$\overline{\text{POE8}}$	输入	输入将 MTU2 的 CH0 引脚设置为高阻抗状态的请求信号

按表 11.2 所示的引脚组合比较输出电平。

表 11.2　引脚组合

引脚组合	输入 / 输出	功能
PE9/TIOC3B 与 PE11/TIOC3D	输出	有效电平输出 （MTU2 的定时器输出控制寄存器 （TOCR）的输出电平选择 P（OLSP）位为 0 时，输出低电平；为 1 时，输出高电平）持续 1 个外围时钟（Pϕ）周期以上时，将 MTU2 特殊引脚设置为高阻抗状态。 通过设定引脚功能控制器，设定为 MTU2 的输出功能或通用输出功能时，比较上述有效电平。设定为其他功能时，则不比较。 通过 POE 的寄存器，可设定对哪一组合执行输出比较及高阻抗控制。
PE12/TIOC4A 与 PE14/TIOC4C	输出	
PE13/TIOC4B 与 PE15/TIOC4D	输出	
PE16/TIOC3BS 与 PE17/TIOC3DS	输出	有效电平输出 （MTU2S 的定时器输出控制寄存器 （TOCR）的输出电平选择 P（OLSP）位为 0 时，输出低电平；为 1 时，输出高电平）持续 1 个外围时钟（Pϕ）周期以上时，将 MTU2S 特殊引脚设置为高阻抗状态。 通过设定引脚功能控制器，设定为 MTU2S 的输出功能或通用输出功能时，比较上述有效电平；设定为其他功能时，则不比较。 通过 POE 的寄存器，可设定对哪一组合执行输出比较及高阻抗控制。
PE18/TIOC4AS 与 PE20/TIOC4CS	输出	
PE19/TIOC4BS 与 PE21/TIOC4DS	输出	

11.3　寄存器说明

POE 有以下寄存器。有关寄存器的地址及各处理状态的寄存器状态，详情参阅 “第 24 章 寄存器一览表”。

表 11.3　寄存器结构

寄存器名称	简称	R/W	初始值	地址	存取长度
输入电平控制 / 状态寄存器 1	ICSR1	R/W	H'0000	H'FFFFD000	8、16、32
输出电平控制 / 状态寄存器 1	OCSR1	R/W	H'0000	H'FFFFD002	8、16
输入电平控制 / 状态寄存器 2	ICSR2	R/W	H'0000	H'FFFFD004	8、16、32
输出电平控制 / 状态寄存器 2	OCSR2	R/W	H'0000	H'FFFFD006	8、16
输入电平控制 / 状态寄存器 3	ICSR3	R/W	H'0000	H'FFFFD008	8、16
软件端口输出允许寄存器	SPOER	R/W	H'00	H'FFFFD00A	8
端口输出允许控制寄存器 1	POECR1	R/W	H'00	H'FFFFD00B	8
端口输出允许控制寄存器 2	POECR2	R/W	H'7700	H'FFFFD00C	8、16

11.3.1　输入电平控制 / 状态寄存器 1（ICSR1）

ICSR1 为可读取 / 写入的 16 位寄存器，选择 $\overline{\text{POE0}}$ ～ $\overline{\text{POE2}}$ 引脚的输入模式、控制中断的允许 / 禁止并显示状态。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	POE2F	POE1F	POE0F	-	-	-	PIE1	-	-	POE2M[1:0]		POE1M[1:0]		POE0M[1:0]	
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R	R/(W)*1	R/(W)*1	R/(W)*1	R	R	R	R/W	R	R	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

【注】*1　为了清除标志，仅可在读取1后写入0。
*2　上电复位后，仅可写入一次。

位	位名称	初始值	R/W	说明
15	—	0	R	保留位 读取值、写入值总是为 0。
14	POE2F	0	R/(W)*1	POE2 标志位 该标志表示已向 $\overline{\text{POE2}}$ 引脚输入高阻抗请求 ［清除条件］ • 读取 POE2F=1 的状态后，对 POE2F 写入 0 时（通过 ICSR1 的 bit5、4 设定为下降沿时） • 使用 Pϕ/8、16、128 时钟采样 POE2 输入的高电平，读取 POE2F=1 的状态后，对 POE2F 写入 0 时（通过 ICSR1 的 bit5、4 设定为低电平采样时） ［置位条件］ • $\overline{\text{POE2}}$ 引脚产生 ICSR1 的 bit5、4 所设定的输入时
13	POE1F	0	R/(W)*1	POE1 标志位 该标志表示已向 $\overline{\text{POE1}}$ 引脚输入高阻抗请求 ［清除条件］ • 读取 POE1F=1 的状态后，对 POE1F 写入 0 时（通过 ICSR1 的 bit3、2 设定为下降沿时） • 使用 Pϕ/8、16、128 时钟采样 POE1 输入的高电平，读取 POE1F=1 的状态后，对 POE1F 写入 0 时（通过 ICSR1 的 bit3、2 设定为低电平采样时） ［置位条件］ • $\overline{\text{POE1}}$ 引脚产生 ICSR1 的 bit3、2 所设定的输入时
12	POE0F	0	R/(W)*1	POE0 标志位 该标志表示已向 $\overline{\text{POE0}}$ 引脚输入高阻抗请求 ［清除条件］ • 读取 POE0F=1 的状态后，对 POE0F 写入 0 时（通过 ICSR1 的 bit1、0 设定为下降沿时） • 使用 Pϕ/8、16、128 时钟采样 POE0 输入的高电平，读取 POE0F=1 的状态后，对 POE0F 写入 0 时（通过 ICSR1 的 bit1、0 设定为低电平采样时） ［置位条件］ • $\overline{\text{POE0}}$ 引脚产生 ICSR1 的 bit1、0 所设定的输入时
11 ～ 9	—	均为 0	R	保留位 读取值、写入值总是为 0。

位	位名称	初始值	R/W	说明
8	PIE1	0	R/W	端口中断允许位 1 指定在 ICSR1 的 POE0F ～ POE2F 位中有 1 位置 1 时，是否请求中断。 0：禁止中断请求 1：允许中断请求
7、6	—	均为 0	R	保留位 读取值、写入值总是为 0。
5、4	POE2M[1:0]	00	R/W*2	POE2 模式位 1、 0 选择 $\overline{\text{POE2}}$ 引脚的输入模式。 00：在 POE2 输入的下降沿接受请求 01：每 Pϕ/8 时钟周期对 POE2 输入的低电平进行 16 次采样，结果均为低电平时，接受请求。 10：每 Pϕ/16 时钟周期对 POE2 输入的低电平进行 16 次采样，结果均为低电平时，接受请求。 11：每 Pϕ/128 时钟周期对 POE2 输入的低电平进行 16 次采样，结果均为低电平时，接受请求。
3、2	POE1M[1:0]	00	R/W*2	POE1 模式位 1、 0 选择 $\overline{\text{POE1}}$ 引脚的输入模式。 00：在 POE1 输入的下降沿接受请求 01：每 Pϕ/8 时钟周期对 POE1 输入的低电平进行 16 次采样，结果均为低电平时，接受请求。 10：每 Pϕ/16 时钟周期对 POE1 输入的低电平进行 16 次采样，结果均为低电平时，接受请求。 11：每 Pϕ/128 时钟周期对 POE1 输入的低电平进行 16 次采样，结果均为低电平时，接受请求。
1、0	POE0M[1:0]	00	R/W*2	POE0 模式位 1、 0 选择 $\overline{\text{POE0}}$ 引脚的输入模式。 00：在 POE0 输入的下降沿接受请求 01：每 Pϕ/8 时钟周期对 POE0 输入的低电平进行 16 次采样，结果均为低电平时，接受请求。 10：每 Pϕ/16 时钟周期对 POE0 输入的低电平进行 16 次采样，结果均为低电平时，接受请求。 11：每 Pϕ/128 时钟周期对 POE0 输入的低电平进行 16 次采样，结果均为低电平时，接受请求。

【注】 *1　为了清除标志，仅可在读取 1 后写入 0。

*2　上电复位后，仅可写入一次。

11.3.2　输出电平控制 / 状态寄存器 1（OCSR1）

OCSR1 为可读取 / 写入的 16 位寄存器，允许 / 禁止输出电平的比较、控制中断的允许 / 禁止并显示状态。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSF1	-	-	-	-	-	OCE1	OIE1	-	-	-	-	-	-	-	-
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/(W)*1	R	R	R	R	R	R/W*2	R/W	R	R	R	R	R	R	R	R

【注】*1　为了清除标志，仅可在读取1后写入0。
*2　上电复位后，仅可写入一次。

位	位名称	初始值	R/W	说明
15	OSF1	0	R/(W)*1	输出短路标志位 1 该标志表示在比较用于 MTU2 的 3 组 2 相输出中，至少有 1 组同时为有效电平。 ［清除条件］ • 读取 OSF1=1 的状态后，对 OSF1 写入 0 时 ［置位条件］ • 3 组 2 相输出中，至少有 1 组同时为有效电平时
14 ～ 10	—	均为 0	R	保留位 读取值、写入值总是为 0。
9	OCE1	0	R/W*2	输出短路高阻抗允许位 1 指定在 OCSR1 的 OSF1 位置位时，是否将引脚设置为高阻抗。 0：不将引脚设置为高阻抗 1：将引脚设置为高阻抗
8	OIE1	0	R/W	输出短路中断允许位 1 指定 OCSR1 的 OSF1 位置位时，是否请求中断。 0：禁止中断请求 1：允许中断请求
7 ～ 0	—	均为 0	R	保留位 读取值、写入值总是为 0。

11.3.3　输入电平控制 / 状态寄存器 2（ICSR2）

ICSR2 为可读取 / 写入的 16 位寄存器，选择 $\overline{\text{POE4}}$ ～ $\overline{\text{POE6}}$ 引脚的输入模式、控制中断的允许 / 禁止并显示状态。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	POE6F	POE5F	POE4F	-	-	-	PIE2	-	-	POE6M[1:0]		POE5M[1:0]		POE4M[1:0]	
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R	R/(W)*1	R/(W)*1	R/(W)*1	R	R	R	R/W	R	R	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

【注】*1　为了清除标志，仅可在读取1后写入0。
*2　上电复位后，仅可写入一次。

位	位名称	初始值	R/W	说明
15	—	0	R	保留位 读取值、写入值总是为 0。
14	POE6F	0	R/(W)*1	POE6 标志位 该标志表示已向 $\overline{\text{POE6}}$ 引脚输入高阻抗请求 ［清除条件］ • 读取 POE6F=1 的状态后，对 POE6F 写入 0 时（通过 ICSR2 的 bit5、4 设定为下降沿时） • 使用 Pϕ/8、16、128 时钟采样 POE6 输入的高电平，读取 POE6F=1 的状态后，对 POE6F 写入 0 时（通过 ICSR2 的 bit5、4 设定为低电平采样时） ［置位条件］ • $\overline{\text{POE6}}$ 引脚产生 ICSR2 的 bit5、4 所设定的输入时
13	POE5F	0	R/(W)*1	POE5 标志位 该标志表示已向 $\overline{\text{POE5}}$ 引脚输入高阻抗请求 ［清除条件］ • 读取 POE5F=1 的状态后，对 POE5F 写入 0 时（通过 ICSR2 的 bit3、2 设定为下降沿时） • 使用 Pϕ/8、16、128 时钟采样 POE5 输入的高电平，读取 POE5F=1 的状态后，对 POE5F 写入 0 时（通过 ICSR2 的 bit3、2 设定为低电平采样时） ［置位条件］ • $\overline{\text{POE5}}$ 引脚产生 ICSR2 的 bit3、2 所设定的输入时
12	POE4F	0	R/(W)*1	POE4 标志位 该标志表示已向 $\overline{\text{POE4}}$ 引脚输入高阻抗请求 ［清除条件］ • 读取 POE4F=1 的状态后，对 POE4F 写入 0 时（通过 ICSR2 的 bit1、0 设定为下降沿时） • 使用 Pϕ/8、16、128 时钟采样 POE4 输入的高电平，读取 POE4F=1 的状态后，对 POE4F 写入 0 时（通过 ICSR2 的 bit1、0 设定为低电平采样时） ［置位条件］ • $\overline{\text{POE4}}$ 引脚产生 ICSR2 的 bit1、0 所设定的输入时
11 ～ 9	—	均为 0	R	保留位 读取值、写入值总是为 0。
8	PIE2	0	R/W	端口中断允许位 2 指定在 ICSR2 的 POE4F ～ POE7F 位中有 1 位置 1 时，是否请求中断。 0：禁止中断请求 1：允许中断请求

位	位名称	初始值	R/W	说明
7、6	—	均为 0	R	保留位 读取值、写入值总是为 0。
5、4	POE6M[1:0]	00	R/W*2	POE6 模式位 1、 0 选择 $\overline{\text{POE6}}$ 引脚的输入模式。 00：在 POE6 输入的下降沿接受请求 01：每 Pϕ/8 时钟周期对 POE6 输入的低电平进行 16 次采样，结果均为低电平时，接受请求。 10：每 Pϕ/16 时钟周期对 POE6 输入的低电平进行 16 次采样，结果均为低电平时，接受请求。 11：每 Pϕ/128 时钟周期对 POE6 输入的低电平进行 16 次采样，结果均为低电平时，接受请求。
3、2	POE5M[1:0]	00	R/W*2	POE5 模式位 1、 0 选择 $\overline{\text{POE5}}$ 引脚的输入模式。 00：在 POE5 输入的下降沿接受请求 01：每 Pϕ/8 时钟周期对 POE5 输入的低电平进行 16 次采样，结果均为低电平时，接受请求。 10：每 Pϕ/16 时钟周期对 POE5 输入的低电平进行 16 次采样，结果均为低电平时，接受请求。 11：每 Pϕ/128 时钟周期对 POE5 输入的低电平进行 16 次采样，结果均为低电平时，接受请求。
1、0	POE4M[1:0]	00	R/W*2	POE4 模式位 1、 0 选择 $\overline{\text{POE4}}$ 引脚的输入模式。 00：在 POE4 输入的下降沿接受请求 01：每 Pϕ/8 时钟周期对 POE4 输入的低电平进行 16 次采样，结果均为低电平时，接受请求。 10：每 Pϕ/16 时钟周期对 POE4 输入的低电平进行 16 次采样，结果均为低电平时，接受请求。 11：每 Pϕ/128 时钟周期对 POE4 输入的低电平进行 16 次采样，结果均为低电平时，接受请求。

【注】 *1　为了清除标志，仅可读取 1 后写入 0。

*2　上电复位后，仅可写入一次。

11.3.4　输出电平控制 / 状态寄存器 2（OCSR2）

OCSR2 为可读取 / 写入的 16 位寄存器，禁止 / 允许输出电平的比较、控制中断的允许 / 禁止并显示状态。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSF2	-	-	-	-	-	OCE2	OIE2	-	-	-	-	-	-	-	-
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/(W)*1	R	R	R	R	R	R/W*2	R/W	R	R	R	R	R	R	R	R

【注】*1　为了清除标志，仅可在读取1后写入0。
*2　上电复位后，仅可写入一次。

位	位名称	初始值	R/W	说明
15	OSF2	0	R/(W)*1	输出短路标志位 2 该标志表示在比较用于 MTU2S 的 3 组 2 相输出中，至少有 1 组同时为有效电平。 ［清除条件］ • 读取 OSF2=1 的状态后，对 OSF2 写入 0 时 ［置位条件］ • 3 组 2 相输出中，至少有 1 组同时为有效电平时
14 ～ 10	—	均为 0	R	保留位 读取值、写入值总是为 0。
9	OCE2	0	R/W*2	输出短路高阻抗允许位 2 指定 OCSR2 的 OSF2 位置位时，是否将引脚设置为高阻抗。 0：不将引脚设置为高阻抗 1：将引脚设置为高阻抗
8	OIE2	0	R/W	输出短路中断允许位 2 指定 OCSR2 的 OSF2 位置位时，是否请求中断。 0：禁止中断请求 1：允许中断请求
7 ～ 0	—	均为 0	R	保留位 读取值、写入值总是为 0。

【注】　*1　为了清除标志，仅可读取 1 后写入 0。
*2　上电复位后，仅可写入一次。

11.3.5　输入电平控制 / 状态寄存器 3（ICSR3）

ICSR3 为可读取 / 写入的 16 位寄存器，选择 $\overline{\text{POE8}}$ 引脚的输入模式、控制中断的允许 / 禁止并显示状态。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	POE8F	-	-	POE8E	PIE3	-	-	-	-	-	-	POE8M[1:0]	
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R	R	R	R/(W)*1	R	R	R/W*2	R/W	R	R	R	R	R	R	R/W*2	R/W*2

【注】*1　为了清除标志，仅可在读取1后写入0。
*2　上电复位后，仅可写入一次。

位	位名称	初始值	R/W	说明
15 ～ 13	—	均为 0	R	保留位 读取值、写入值总是为 0。
12	POE8F	0	R/(W)*1	POE8 标志位 该标志表示已向 $\overline{\text{POE8}}$ 引脚输入高阻抗请求 ［清除条件］ • 读取 POE8F=1 的状态后，对 POE8F 写入 0 时（通过 ICSR3 的 bit1、0 设定为下降沿时） • 使用 Pϕ/8、16、128 时钟采样 POE8 输入的高电平，读取 POE8F=1 的状态后，对 POE8F 写入 0 时（通过 ICSR3 的 bit1、0 设定为低电平采样时） ［置位条件］ • $\overline{\text{POE8}}$ 引脚产生 ICSR3 的 bit1、0 所设定的输入时
11、10	—	均为 0	R	保留位 读取值、写入值总是为 0。
9	POE8E	0	R/W*2	POE8 高阻抗允许位置位 指定 ICSR3 的 POE8F 位置位时，是否将引脚设置为高阻抗。 0：不将引脚设置为高阻抗 1：将引脚设置为高阻抗
8	PIE3	0	R/W	端口中断允许位 3 指定 ICSR3 的 POE8 位置 1 时，是否请求中断。 0：禁止中断请求 1：允许中断请求
7 ～ 2	—	均为 0	R	保留位 读取值、写入值总是为 0。
1、0	POE8M[1:0]	00	R/W*2	POE8 模式位 1、0 选择 $\overline{\text{POE8}}$ 引脚的输入模式。 00：在 POE8 输入的下降沿接受请求 01：每 Pϕ/8 时钟周期对 POE8 输入的低电平进行 16 次采样，结果均为低电平时，接受请求。 10：每 Pϕ/16 时钟周期对 POE8 输入的低电平进行 16 次采样，结果均为低电平时，接受请求。 11：每 Pϕ/128 时钟周期对 POE8 输入的低电平进行 16 次采样，结果均为低电平时，接受请求。

【注】　*1　为了清除标志，仅可读取 1 后写入 0。
*2　上电复位后，仅可写入一次。

11.3.6　软件端口输出允许寄存器（SPOER）

SPOER 为可读取 / 写入的 8 位寄存器，控制引脚的高阻抗。

位：	7	6	5	4	3	2	1	0
	-	-	-	-	-	MTU2S HIZ	MTU2 CH0HIZ	MTU2 CH34HIZ
初始值：	0	0	0	0	0	0	0	0
R/W：	R	R	R	R	R	R/W	R/W	R/W

位	位名称	初始值	R/W	说明
7 ～ 3	—	均为 0	R	保留位 读取值、写入值总是为 0。
2	MTU2SHIZ	0	R/W	MTU2S 输出高阻抗 控制是否将 MTU2S 特殊引脚设置为高阻抗。 0：不设置为高阻抗状态。 ［清除条件］ • 上电复位 • 读取 MTU2SHIZ=1 的状态后，对 MTU2SHIZ 写入 0 时 1：设置为高阻抗状态。 ［置位条件］ • 对 MTU2SHIZ 写入 1 时
1	MTU2CH0HIZ	0	R/W	MTU2CH0 输出高阻抗 控制是否将 MTU2 的 CH0 引脚设置为高阻抗。 0：不设置为高阻抗状态。 ［清除条件］ • 上电复位 • 读取 MTU2CH0HIZ=1 的状态后，对 MTU2CH0HIZ 写入 0 时 1：设置为高阻抗状态。 ［置位条件］ • 对 MTU2CH0HIZ 写入 1 时
0	MTU2CH34HIZ	0	R/W	MTU2CH3、4 输出高阻抗 控制是否将 MTU2 特殊引脚设置为高阻抗。 0：不设置为高阻抗状态。 ［清除条件］ • 上电复位 • 读取 MTU2CH34HIZ=1 的状态后，对 MTU2CH34HIZ 写入 0 时 1：设置为高阻抗状态。 ［置位条件］ • 对 MTU2CH34HIZ 写入 1 时

11.3.7　端口输出允许控制寄存器 1（POECR1）

POECR1 为可读取 / 写入的 8 位寄存器，控制引脚的高阻抗。

位：	7	6	5	4	3	2	1	0
	-	-	-	-	MTU2 PE3ZE	MTU2 PE2ZE	MTU2 PE1ZE	MTU2 PE0ZE
初始值：	0	0	0	0	0	0	0	0
R/W：	R	R	R	R	R/W*	R/W*	R/W*	R/W*

【注】* 上电复位后，仅可写入一次。

位	位名称	初始值	R/W	说明
7 ～ 4	—	均为 0	R	保留位 读取值、写入值总是为 0。
3	MTU2PE3ZE	0	R/W*	MTU2 PE3 高阻抗允许 设定在 POE8F、MTU2CH0HIZ 位中至少有 1 位置位时，是否将 MTU2 的 CH0 引脚的 PE3/TIOC0D 设置为高阻抗状态。 0：不设置为高阻抗。 1：设置为高阻抗。
2	MTU2PE2ZE	0	R/W*	MTU2 PE2 高阻抗允许 设定在 POE8F、MTU2CH0HIZ 位中至少有 1 位置位时，是否将 MTU2 的 CH0 引脚的 PE2/TIOC0C 设置为高阻抗状态。 0：不设置为高阻抗。 1：设置为高阻抗。
1	MTU2PE1ZE	0	R/W*	MTU2 PE1 高阻抗允许 设定在 POE8F、MTU2CH0HIZ 位中至少有 1 位置位时，是否将 MTU2 的 CH0 引脚的 PE1/TIOC0B 设置为高阻抗状态。 0：不设置为高阻抗。 1：设置为高阻抗。
0	MTU2PE0ZE	0	R/W*	MTU2 PE0 高阻抗允许 设定在 POE8F、MTU2CH0HIZ 位中至少有 1 位置位时，是否将 MTU2 的 CH0 引脚的 PE0/TIOC0A 设置为高阻抗状态。 0：不设置为高阻抗。 1：设置为高阻抗。

11.3.8　端口输出允许控制寄存器 2（POECR2）

POECR2 为可读取 / 写入的 16 位寄存器，控制引脚的高阻抗。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	MTU2 P1CZE	MTU2 P2CZE	MTU2 P3CZE	-	MTU2S P1CZE	MTU2S P2CZE	MTU2S P3CZE	-	-	-	-	-	-	-	-
初始值：	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0
R/W：	R	R/W*	R/W*	R/W*	R	R/W*	R/W*	R/W*	R	R	R	R	R	R	R	R

【注】* 上电复位后，仅可写入一次。

位	位名称	初始值	R/W	说明
15	—	0	R	保留位 读取值、写入值总是为 0。
14	MTU2P1CZE	1	R/W*	MTU2 端口 1 输出比较 / 高阻抗允许 设定是否对 MTU2 特殊引脚 PE9/TIOC3B 和 PE11/TIOC3D 比较输出电平。 另外， OCE1 位为 1 时，如果 OSF1 位置位，或 POE0F、 POE1F、 POE2F、 MTU2CH34HIZ 位中有 1 位置位时，设定是否设置为高阻抗状态。 0：不比较输出电平，不设置为高阻抗。 1：比较输出电平，设置为高阻抗。
13	MTU2P2CZE	1	R/W*	MTU2 端口 2 输出比较 / 高阻抗允许 设定是否对 MTU2 特殊引脚 PE12/TIOC4A 和 PE14/TIOC4C 比较输出电平。 另外， OCE1 位为 1 时，如果 OSF1 位置位，或 POE0F、 POE1F、 POE2F、 MTU2CH34HIZ 位中有 1 位置位时，设定是否设置为高阻抗状态。 0：不比较输出电平，不设置为高阻抗。 1：比较输出电平，设置为高阻抗。
12	MTU2P3CZE	1	R/W*	MTU2 端口 3 输出比较 / 高阻抗允许 设定是否对 MTU2 特殊引脚 PE13/TIOC4B 和 PE15/TIOC4D 比较输出电平。 另外， OCE1 位为 1 时，如果 OSF1 位置位，或 POE0F、 POE1F、 POE2F、 MTU2CH34HIZ 位中有 1 位置位时，设定是否设置为高阻抗状态。 0：不比较输出电平，不设置为高阻抗。 1：比较输出电平，设置为高阻抗。
11	—	0	R	保留位 读取值、写入值总是为 0。
10	MTU2SP1CZE	1	R/W*	MTU2S 端口 1 输出比较 / 高阻抗允许 设定是否对 MTU2S 特殊引脚 PE16/TIOC3BS 和 PE17/TIOC3DS 比较输出电平。 另外， OCE2 位为 1 时，如果 OSF2 位置位，或 POE4F、 POE5F、 POE6F、 POE7F、 MTU2SHIZ 位中有 1 位置位时，设定是否设置为高阻抗状态。 0：不比较输出电平，不设置为高阻抗。 1：比较输出电平，设置为高阻抗。
9	MTU2SP2CZE	1	R/W*	MTU2S 端口 2 输出比较 / 高阻抗允许 设定是否对 MTU2S 特殊引脚 PE18/TIOC4AS 和 PE20/TIOC4CS 比较输出电平。 另外， OCE2 位为 1 时，如果 OSF2 位置位，或 POE4F、 POE5F、 POE6F、 POE7F、 MTU2SHIZ 位中有 1 位置位时，设定是否设置为高阻抗状态。 0：不比较输出电平，不设置为高阻抗。 1：比较输出电平，设置为高阻抗。
8	MTU2SP3CZE	1	R/W*	MTU2S 端口 3 输出比较 / 高阻抗允许 设定是否对 MTU2S 特殊引脚 PE19/TIOC4BS 和 PE21/TIOC4DS 比较输出电平。 另外， OCE2 位为 1 时，如果 OSF2 位置位，或 POE4F、 POE5F、 POE6F、 POE7F、 MTU2SHIZ 位中有 1 位置位时，设定是否设置为高阻抗状态。 0：不比较输出电平，不设置为高阻抗。 1：比较输出电平，设置为高阻抗。
7 ～ 0	—	均为 0	R	保留位 读取值、写入值总是为 0。

【注】　*　上电复位后，仅可写入一次。

11.4　运行说明

高阻抗控制的对象引脚及成为高阻抗的条件如表 11.4 所示。

表 11.4　高阻抗控制的对象及条件

引脚	条件	详细条件
MTU2 特殊引脚 （PE9/TIOC3B、 PE11/TIOC3D）	输入电平检测运行、 输出电平比较运行或 设定 SPOER 寄存器	MTU2P1CZE・((POE2F+POE1F+POE0F)+ (OSF1・OCE1)+(MTU2CH34HIZ))
MTU2 特殊引脚 （PE12/TIOC4A、 PE14/TIOC4C）	输入电平检测运行、 输出电平比较运行或 设定 SPOER 寄存器	MTU2P2CZE・((POE2F+POE1F+POE0F)+ (OSF1・OCE1)+(MTU2CH34HIZ))
MTU2 特殊引脚 （PE13/TIOC4B、 PE15/TIOC4D）	输入电平检测运行、 输出电平比较运行或 设定 SPOER 寄存器	MTU2P3CZE・((POE2F+POE1F+POE0F)+ (OSF1・OCE1)+(MTU2CH34HIZ))
MTU2S 特殊引脚 （PE16/TIOC3BS、 PE17/TIOC3DS）	输入电平检测运行、 输出电平比较运行或 设定 SPOER 寄存器	MTU2SP1CZE・((POE4F+POE5F+POE6F)+ (OSF2・OCE2)+(MTU2SHIZ))
MTU2S 特殊引脚 （PE18/TIOC4AS、 PE20/TIOC4CS）	输入电平检测运行、 输出电平比较运行或 设定 SPOER 寄存器	MTU2SP2CZE・((POE4F+POE5F+POE6F)+ (OSF2・OCE2)+(MTU2SHIZ))
MTU2S 特殊引脚 （PE19/TIOC4BS、 PE21/TIOC4DS）	输入电平检测运行、 输出电平比较运行或 设定 SPOER 寄存器	MTU2SP3CZE・((POE4F+POE5F+POE6F)+ (OSF2・OCE2)+(MTU2SHIZ))
MTU2 用的 CH0 引脚 （PE0/TIOC0A）	输入电平检测运行或 设定 SPOER 寄存器	MTU2PE0ZE ((POE8F・POE8E)+(MTU2CH0HIZ))
MTU2 用的 CH0 引脚 （PE1/TIOC0B）	输入电平检测运行或 设定 SPOER 寄存器	MTU2PE1ZE ((POE8F・POE8E)+(MTU2CH0HIZ))
MTU2 用的 CH0 引脚 （PE2/TIOC0C）	输入电平检测运行或 设定 SPOER 寄存器	MTU2PE2ZE ((POE8F・POE8E)+(MTU2CH0HIZ))
MTU2 用的 CH0 引脚 （PE3/TIOC0D）	输入电平检测运行或 设定 SPOER 寄存器	MTU2PE3ZE ((POE8F・POE8E)+(MTU2CH0HIZ))

11.4.1　输入电平检测运行

$\overline{\text{POE0}}$ ～ $\overline{\text{POE2}}$、$\overline{\text{POE4}}$ ～ $\overline{\text{POE6}}$、$\overline{\text{POE8}}$ 引脚产生 ICSR1 ～ ICSR3 所设定的输入条件时， MTU2、 2S 特殊引脚及 MTU2 的 CH0 引脚设置为高阻抗状态。但，仅在选择通用输入 / 输出功能或 MTU2、 MTU2S 功能时，MTU2、 2S 特殊引脚及 MTU2 的 CH0 引脚才为高阻抗。

(1)　下降沿检测

$\overline{\text{POE0}}$ ～ $\overline{\text{POE2}}$、 $\overline{\text{POE4}}$ ～ $\overline{\text{POE6}}$、 $\overline{\text{POE8}}$ 引脚的输入电平从高电平变为低电平时， MTU2、 2S 特殊引脚及 MTU2 的 CH0 引脚设置为高阻抗状态。

从输入 $\overline{\text{POE0}}$ ～ $\overline{\text{POE2}}$、 $\overline{\text{POE4}}$ ～ $\overline{\text{POE6}}$、 $\overline{\text{POE8}}$ 引脚到引脚高阻抗为止的时序例子如图 11.2 所示。

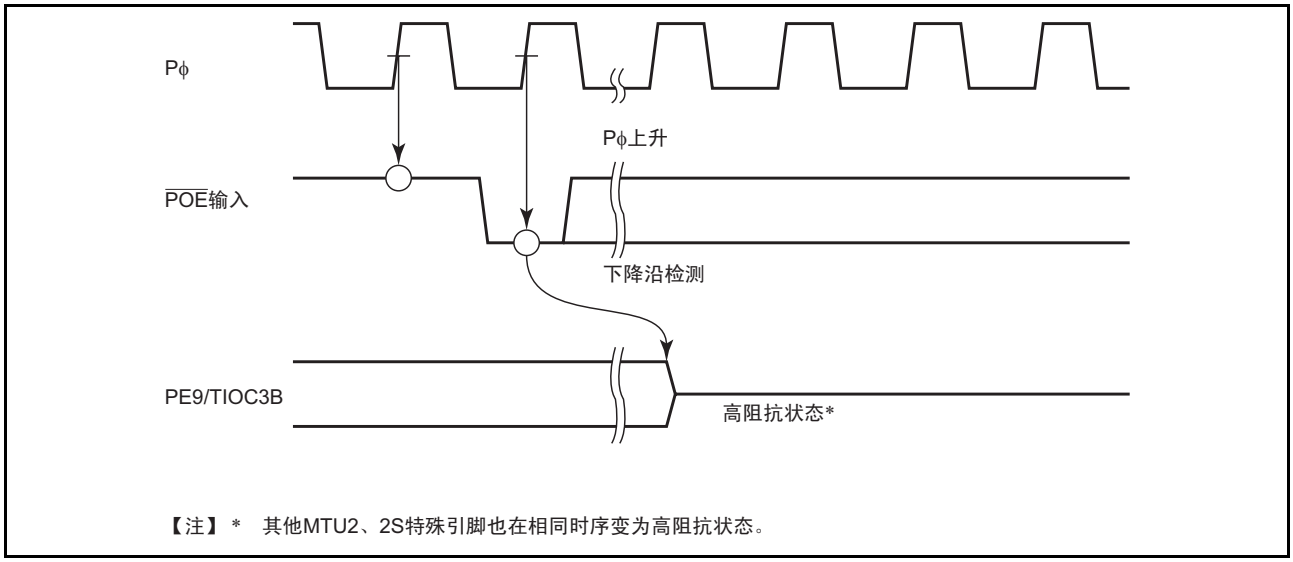


图 11.2　下降沿检测运行

(2)　低电平检测

低电平检测运行如图 11.3 所示。使用 ICSR1 ～ ICSR3 设定的采样时钟连续执行 16 次低电平采样。此时，只要检测到 1 次高电平，就不接受。

另外，从采样时钟到 MTU2、 2S 特殊引脚变为高阻抗状态的时序为下降沿检测与低电平检测相同。

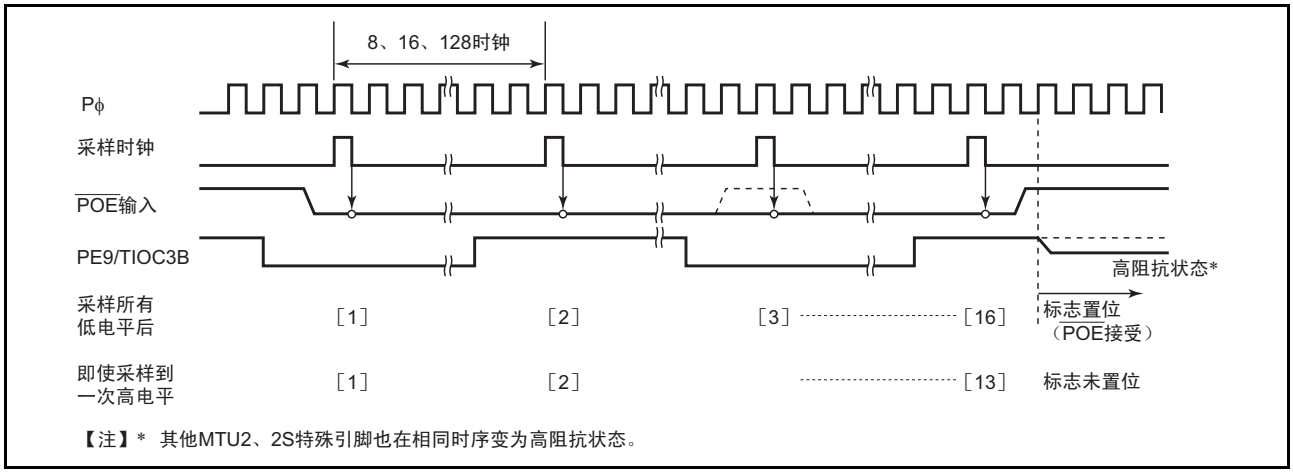


图 11.3　低电平检测运行

11.4.2　输出电平比较运行

以 TIOC3B 与 TIOC3D 的组合为例，输出电平比较运行如图 11.4 所示。其他引脚组合也相同。

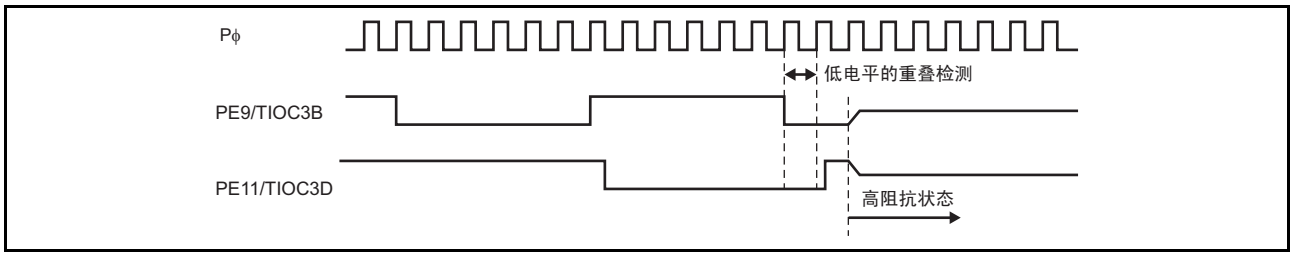


图 11.4　输出电平比较运行

11.4.3　高阻抗状态的解除

经输入电平检测变为高阻抗状态的 MTU2、2S 特殊引脚，通过上电复位恢复为初始状态，或通过清除 ICSR1 ～ ICSR3 的 bit12 ～ 15（POE0F ～ POE2F、POE4F ～ POE6F、POE8F）的标志解除高阻抗状态。但通过 ICSR1 ～ ICSR3 的 bit0 ～ 7 设定为低电平采样时，必须先从 POE 引脚输入高电平并采样，否则，即使对标志写入 0 也无效，不能清除标志。

经输出电平检测变为高阻抗状态的 MTU2、2S 特殊引脚，通过上电复位恢复为初始状态，或通过清除 OCSR1、OCSR2 的 bit15（OCF1、2）的标志解除高阻抗状态。但必须先从 MTU2、2S 特殊引脚输出非有效电平，否则，即使对标志写入 0 也无效，不能清除标志。可通过设定 MTU2、MTU2S 内的寄存器，输出非有效电平。

11.5　中断

输入电平检测运行或输出电平比较运行过程中条件匹配时，POE 可发出中断请求并产生中断。中断的种类与发出中断请求的条件如表 11.5 所示。

表 11.5　中断请求的种类与条件

名称	中断源	中断标志	条件
OEI1	输出允许中断 1	POE2F、POE1F、POE0F、OSF1	PIE1・(POE2F+POE1F+POE0F)+OIE1・OSF1
OEI3	输出允许中断 3	POE8F	PIE3・POE8F
OEI2	输出允许中断 2	POE4F、POE5F、POE6F、OSF2	PIE2・(POE4F+POE5F+POE6F)+OIE2・OSF2

11.6　使用时的注意事项

11.6.1　从看门狗定时器发行上电复位时的引脚状态

从看门狗定时器（WDT）发行上电复位时，引脚功能控制器（PFC）初始化，I/O 端口变为通用输入（初始值）。但，端口输出允许（POE）引起的引脚高阻抗处理过程中，从 WDT 发行上电复位时，在转换为通用输入前的 1 个外围时钟（Pϕ）周期内，引脚为输出状态。

MTU2、MTU2S 的短路检测引起的高阻抗处理过程中，从 WDT 发行上电复位时的状态同上。

选择定时器输出时的 POE 输入引起的高阻抗处理过程中，从 WDT 发行上电复位时的状态如图 11.5 所示。

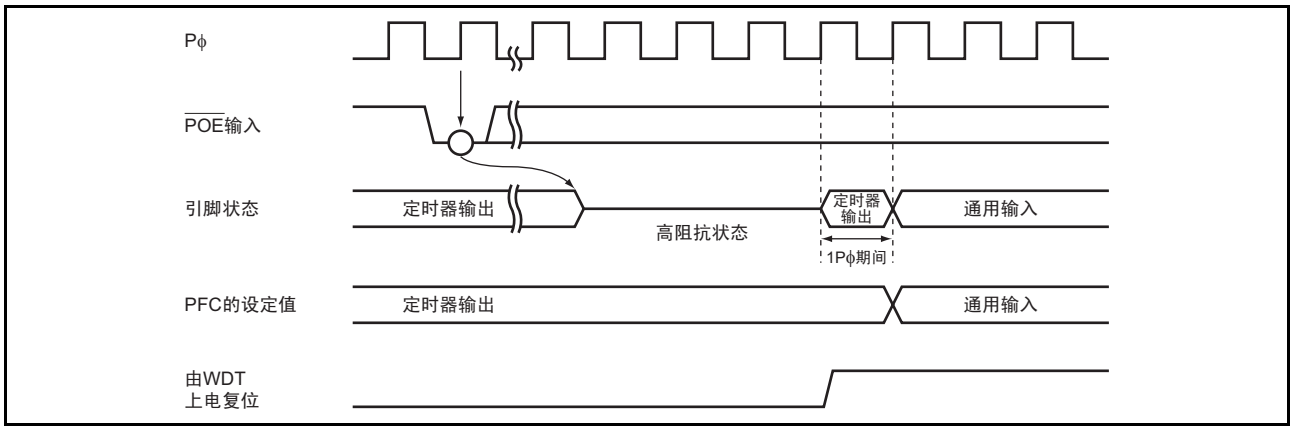


图 11.5　从看门狗定时器发行上电复位时的引脚状态

第 12 章　看门狗定时器 （WDT）

本 LSI 内置看门狗定时器 （WDT），在系统失控等导致计数器的值不可改写而上溢时，可复位本 LSI 内部。

WDT 为 1 个通道的定时器，以外围时钟作为输入时钟，解除软件待机模式时，可用于计数时钟稳定时间。也可用作间隔定时器。

12.1　特点

- 可确保时钟稳定时间
  解除软件待机模式时使用。
- 可切换看门狗定时器模式和间隔定时器模式
- 看门狗定时器模式时，产生内部复位
  因计数器上溢而复位内部。
- 间隔定时器模式时，产生中断
  因计数器上溢而产生间隔定时器中断。
- 可选择8种计数器输入时钟
  可从已分频外围时钟的8种时钟 （×1～×1/4096）中选择。
- 复位种类可选择上电复位和手动复位。

WDT 框图如图 12.1 所示。

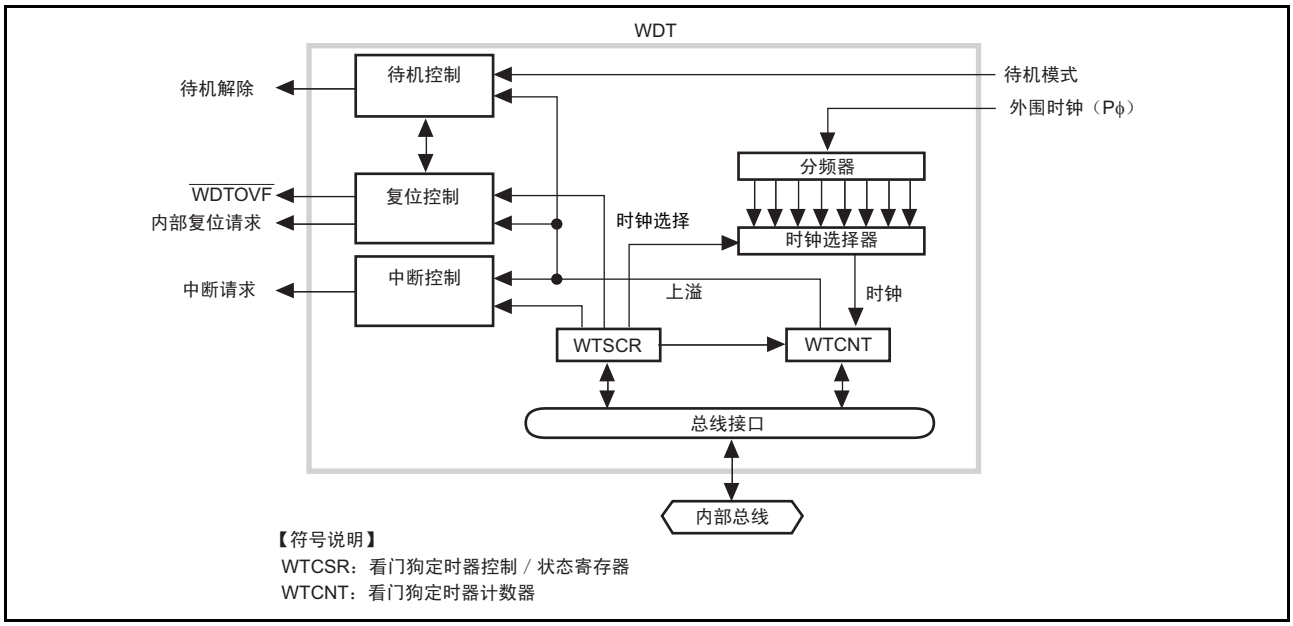


图 12.1　WDT 框图

12.2　输入 / 输出引脚

WDT 引脚功能如表 12.1 所示。

表 12.1　WDT 引脚结构

名称	引脚名称	输入 / 输出	功　　能
看门狗定时器上溢	$\overline{\text{WDTOVF}}$	输出	因在看门狗定时器模式产生上溢，而产生内部复位，在 WTCSR 的 CKS2 ～ CKS0 设定的 1 个时钟周期期间，从本引脚输出低电平。

12.3　寄存器说明

WDT 有以下寄存器。有关寄存器的地址及各处理状态的寄存器状态，详情参阅 “第 24 章 寄存器一览表”。

表 12.2　寄存器结构

寄存器名称	简称	R/W	初始值	地址	存取长度
看门狗定时器计数器	WTCNT	R/W	H'00	H'FFFFE810	8、16
看门狗定时器控制 / 状态寄存器	WTCSR	R/W	H'00	H'FFFFE812	8、16

12.3.1　看门狗定时器计数器 （WTCNT）

WTCNT 为可读取 / 写入的 8 位寄存器，是由所选时钟累加计数的计数器。上溢时，在看门狗定时器模式产生复位；在间隔定时器模式产生中断。因 WDT 上溢而内部复位时， WTCNT 计数器不被初始化，仅在 $\overline{\text{RES}}$ 引脚引起上电复位时，初始化为 H'00。

写入 WTCNT 计数器时，必须将高位字节设置为 H'5A，并以字长度写入，以字节长度读取。

【注】 为了防止错误改写，本寄存器的写入方法与一般的寄存器不同。详情参阅 “12.3.3 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

12.3.2　看门狗定时器控制 / 状态寄存器 （WTCSR）

WTCSR 为可读取 / 写入的 8 位寄存器，由选择计数时钟的位、上溢标志及允许位构成。

因 WDT 上溢而内部复位时， WTCSR 寄存器保持其值。仅在 $\overline{\text{RES}}$ 引脚引起上电复位时，初始化为 H'00。计数软件待机解除时的时钟稳定时间时，计数器上溢后保持其值。

写入 WTCSR 寄存器时，必须将高位字节设置为 H'A5，并以字长度写入，以字节长度读取。

【注】 为了防止错误改写，本寄存器的写入方法与一般的寄存器不同。详情参阅 “12.3.3 寄存器存取时的注意事项”。

位：	7	6	5	4	3	2	1	0
	TME	WT/IT	RSTS	WOVF	IOVF	CKS[2:0]		
初始值：	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说　明
7	TME	0	R/W	定时器允许 设定开始或停止定时器运行。如果解除软件待机模式时使用 WDT，则该位必须清 0。 0：禁止定时器。停止累加计数，保持 WTCNT 的值 1：允许定时器
6	WT/IT	0	R/W	定时器模式选择 指定用作看门狗定时器或间隔定时器。 0：间隔定时器模式 1：看门狗定时器模式 【注】在 WDT 运行过程中改写 WT/IT 时，可能无法正确累加计数。
5	RSTS	0	R/W	复位选择 在看门狗定时器模式，指定 WTCNT 上溢时的复位类型；在间隔定时器模式，忽略设定值。 0：上电复位 1：手动复位
4	WOVF	0	R/W	看门狗定时器上溢 表示在看门狗定时器模式 WTCNT 上溢，在间隔定时器模式不被置位。 0：无上溢 1：在看门狗定时器模式 WTCNT 上溢
3	IOVF	0	R/W	间隔定时器上溢 表示在间隔定时器模式 WTCNT 上溢，在看门狗定时器模式不被置位。 0：无上溢 1：在间隔定时器模式 WTCNT 上溢
2 ～ 0	CKS[2:0]	000	R/W	时钟选择 2 ～ 0 从分频外围时钟 （Pϕ）后得到的 8 种时钟中，选择计数 WTCNT 的时钟。括号内的上溢周期为外围时钟 Pϕ=40MHz 时的值。 000：Pϕ　（6.4μs） 001：Pϕ/4　（25.6μs） 010：Pϕ/16　（102.4μs） 011：Pϕ/32　（204.8μs） 100：Pϕ/64　（409.6μs） 101：Pϕ/256　（1.64 ms） 110：Pϕ/1024　（6.55 ms） 111：Pϕ/4096　（26.21 ms） 【注】 WDT 运行过程中改写 CKS2 ～ CKS0 位时，可能无法正确累加计数。改写 CKS2 ～ CKS0 位时，必须停止 WDT。

12.3.3　寄存器存取时的注意事项

为了防止轻易改写，看门狗定时器计数器 （WTCNT）、看门狗定时器控制 / 状态寄存器 （WTCSR）的写入方法与一般的寄存器不同，必须按照以下方法写入。

- 写入WTCNT、 WTCSR时，必须使用字传送指令，字节传送、长字传送指令不可用于写入。

如图 12.2 所示，写入 WTCNT 时，必须将高位字节设置为 H'5A，将低位字节作为写入数据传送；写入 WTCSR 时，必须将高位字节设置为 H'A5，将低位字节作为写入数据传送。照此传送，低位字节的数据即可写入 WTCNT 或 WTCSR。

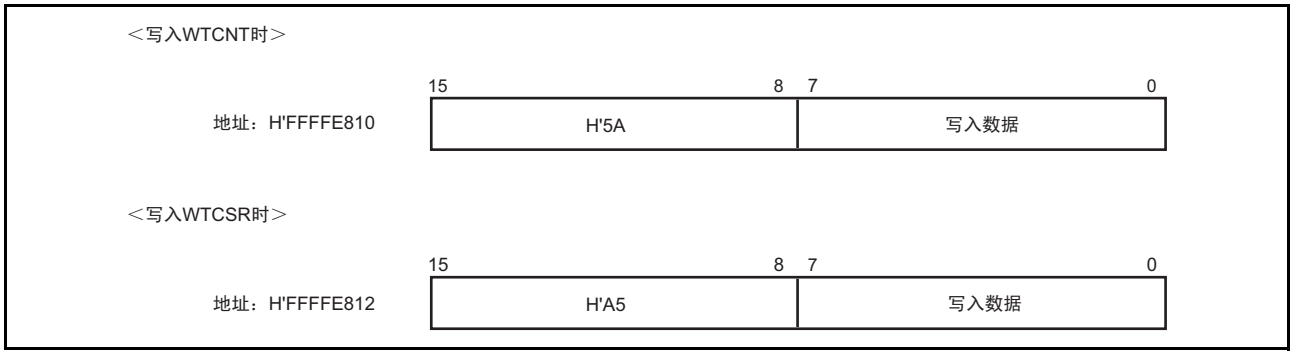


图 12.2　写入 WTCNT、 WTCSR

12.4　运行说明

12.4.1　解除软件待机的步骤

WDT 用于由 NMI 中断或外部中断 （IRQ）解除软件待机模式，步骤如下：（由复位解除时 WDT 不运行，因此，在时钟稳定前，必须保持 $\overline{\text{RES}}$ 引脚的低电平状态。）

1. 转移至软件待机模式前，必须将WTCSR寄存器的TME位设定为0。如果设定为1时，可能在计数上溢时错误复位或产生间隔定时器中断。
2. 预先将使用的计数时钟的种类和时钟的初始值分别设定在WTCSR寄存器的CKS2～CKS0位和WTCNT计数器。设定这些值时，必须使计数上溢前的时间大于等于时钟振荡稳定时间。
3. 执行SLEEP指令，转移至软件待机模式，时钟停止。
4. 向NMI引脚或IRQ引脚输入的电平有变时，通过边沿检测， WDT开始计数。
5. WDT计数上溢时， CPG开始提供时钟，本LSI重新开始运行。此时， WTCSR寄存器的WOVF标志不被置位。

12.4.2　看门狗定时器模式的使用方法

看门狗定时器模式运行过程中，计数器每次上溢时，产生 WTCSR 寄存器 RSTS 位所选种类的内部复位，$\overline{\text{WDTOVF}}$ 引脚有效。

1. WTCSR寄存器的WT/IT位设定为1，并在RSTS位设定复位类型，在CKS2～CKS0设定计数时钟种类，在WTCNT计数器设定计数器的初始值。
2. WTCSR寄存器的TME位置1时，以看门狗定时器模式开始计数。
3. 看门狗定时器模式运行过程中，为了防止计数器上溢，必须定期改写计数器。
4. 计数器上溢时， WDT将WTCSR寄存器的WOVF标志置1，在CKS2～CKS0位设定的计数时钟的1个周期期间， $\overline{\text{WDTOVF}}$引脚有效，并产生RSTS位指定类型的复位。此后计数器继续计数。

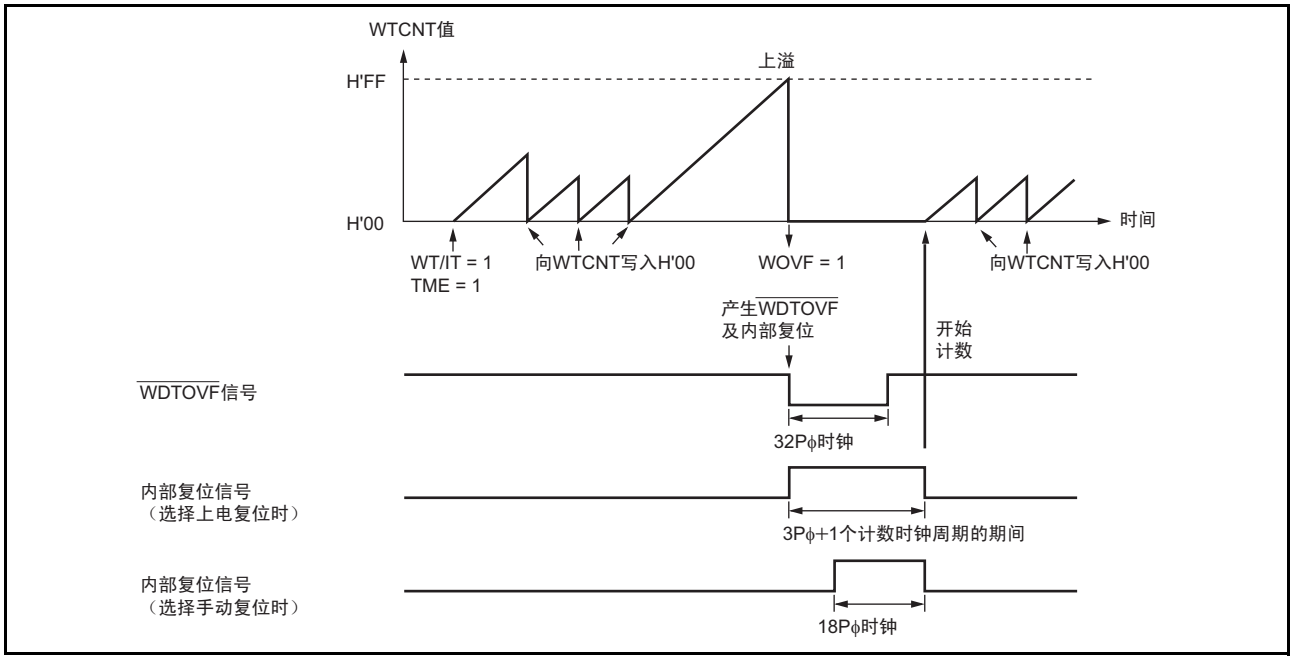


图 12.3　看门狗定时器模式时的运行
（通过 CKS0 ～ CKS2 设定 WTCNT 的计数时钟为 Pϕ/32 时）

12.4.3　间隔定时器模式的使用方法

间隔定时器模式运行过程中，计数器每次上溢时，产生间隔定时器中断。因此，可定期产生中断。

1. WTCSR寄存器的WT/IT位置0，并在CKS2～CKS0位设定计数时钟的种类，在WTCNT计数器设定计数器的初始值。
2. WTCSR寄存器的TME位置1时，在间隔定时器模式开始计数。
3. 计数器上溢时， WDT将WTCSR寄存器的IOVF标志置1，并向INTC发送间隔定时器中断请求。计数器继续计数。

12.5　使用时的注意事项

12.5.1　WTCNT 的设定值

在间隔定时器模式给 WTCNT 设定 H'FF 时， 1 个计数时钟后的 H'FF→H'00 时不产生上溢，而 257 个计数时钟后的 H'FF→H'00 时产生上溢。

在看门狗定时器模式给 WTCNT 设定 H'FF 时， 1 个计数时钟后的 H'FF→H'00 时产生上溢。

第 13 章 串行通信接口 （SCI）

本 LSI 具有 3 个通道的独立串行通信接口 （SCI：Serial Communication Interface）。 SCI 能以异步及时钟同步两种方式执行串行通信。异步方式时，可与 Universal Asynchronous Receiver/Transmitter （UART）或 Asynchronous Communication Interface Adapter （ACIA）等标准异步通信 LSI 执行串行通信。另外，异步模式还具有多个处理器之间的串行通信功能 （多处理器通信功能）。

13.1 特点

- 可从异步模式或时钟同步模式选择串行通信模式
- 异步模式
  通过以字符为单位取得同步的异步方式执行串行数据通信。可与Universal Asynchronous Receiver/Transmitter （UART）或Asynchronous Communication Interface Adapter （ACIA）等标准异步通信的LSI执行串行数据通信。
  可从12种格式中选择串行数据通信格式。
  数据长度：7位或8位
  停止位长度：1位或2位
  奇偶校验：偶校验、奇校验或无奇偶校验
  可执行多处理器间的通信
  检测接收错误：检测奇偶校验错误、溢出错误、帧错误
  检测中断：产生帧错误时，可通过直接读取RXD引脚的电平检测中断。
- 时钟同步模式
  与时钟同步执行串行数据通信。可与具有时钟同步通信功能的其他LSI执行串行数据通信。
  串行数据通信格式为1种。
  数据长度：8位
  检测接收错误：检测溢出错误
- 可执行全双工通信
  具有独立的发送部与接收部，因此可同时发送与接收。
  另外，发送部与接收部均为双缓冲结构，因此可连续发送/接收串行数据。
- 可由内部波特率发生器选择任意位速率
- 可从波特率发生器的内部时钟或SCK引脚的外部时钟选择发送/接收时钟源
- 可选择LSB first/MSB first （除异步7位数据之外）
- 4种中断源
  中断源有发送数据空、发送结束、接收数据满及接收错误等4种，可分别独立请求中断。另外，可由发送数据空请求和接收数据满请求，启动数据传送控制器 （DTC），执行数据传送。
- 可设定模块待机模式

SCI 框图如图 13.1 所示。

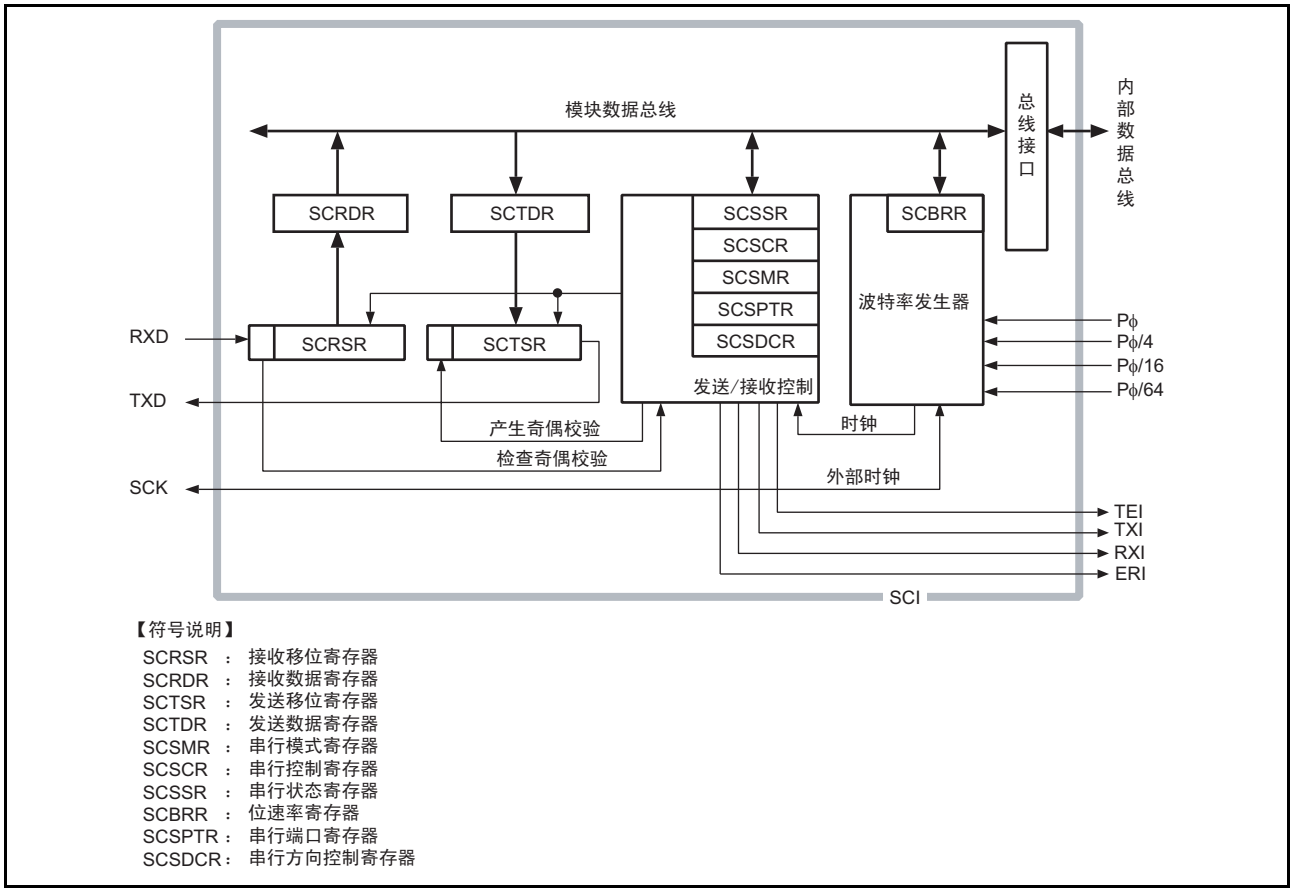


图 13.1　SCI 框图

13.2　输入 / 输出引脚

SCI 的输入 / 输出引脚如表 13.1 所示。

表 13.1　引脚结构

通道	引脚名称 *	输入 / 输出	功　能
0	SCK0	输入 / 输出	通道 0 的时钟输入 / 输出引脚
	RXD0	输入	通道 0 的接收数据输入引脚
	TXD0	输出	通道 0 的发送数据输出引脚
1	SCK1	输入 / 输出	通道 1 的时钟输入 / 输出引脚
	RXD1	输入	通道 1 的接收数据输入引脚
	TXD1	输出	通道 1 的发送数据输出引脚
2	SCK2	输入 / 输出	通道 2 的时钟输入 / 输出引脚
	RXD2	输入	通道 2 的接收数据输入引脚
	TXD2	输出	通道 2 的发送数据输出引脚

【注】 *　本章省去 “通道”，分别简称为 SCK、 RXD、 TXD。

13.3　寄存器说明

SCI 的各通道有以下寄存器。有关寄存器的地址及各处理状态的寄存器状态，详情参阅“第 24 章 寄存器一览表”。

表 13.2　寄存器结构

通道	寄存器名称	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'00	H'FFFFC000	8
	位速率寄存器 _0	SCBRR_0	R/W	H'FF	H'FFFFC002	8
	串行控制寄存器 _0	SCSCR_0	R/W	H'00	H'FFFFC004	8
	发送数据寄存器 _0	SCTDR_0	—	—	H'FFFFC006	8
	串行状态寄存器 _0	SCSSR_0	R/W	H'84	H'FFFFC008	8
	接收数据寄存器 _0	SCRDR_0	—	—	H'FFFFC00A	8
	串行方向控制寄存器 _0	SCSDCR_0	R/W	H'F2	H'FFFFC00C	8
	串行端口寄存器 _0	SCSPTR_0	R/W	H'0x	H'FFFFC00E	8
1	串行模式寄存器 _1	SCSMR_1	R/W	H'00	H'FFFFC080	8
	位速率寄存器 _1	SCBRR_1	R/W	H'FF	H'FFFFC082	8
	串行控制寄存器 _1	SCSCR_1	R/W	H'00	H'FFFFC084	8
	发送数据寄存器 _1	SCTDR_1	—	—	H'FFFFC086	8
	串行状态寄存器 _1	SCSSR_1	R/W	H'84	H'FFFFC088	8
	接收数据寄存器 _1	SCRDR_1	—	—	H'FFFFC08A	8
	串行方向控制寄存器 _1	SCSDCR_1	R/W	H'F2	H'FFFFC08C	8
	串行端口寄存器 _1	SCSPTR_1	R/W	H'0x	H'FFFFC08E	8
2	串行模式寄存器 _2	SCSMR_2	R/W	H'00	H'FFFFC100	8
	位速率寄存器 _2	SCBRR_2	R/W	H'FF	H'FFFFC102	8
	串行控制寄存器 _2	SCSCR_2	R/W	H'00	H'FFFFC104	8
	发送数据寄存器 _2	SCTDR_2	—	—	H'FFFFC106	8
	串行状态寄存器 _2	SCSSR_2	R/W	H'84	H'FFFFC108	8
	接收数据寄存器 _2	SCRDR_2	—	—	H'FFFFC10A	8
	串行方向控制寄存器 _2	SCSDCR_2	R/W	H'F2	H'FFFFC10C	8
	串行端口寄存器 _2	SCSPTR_2	R/W	H'0x	H'FFFFC10E	8

13.3.1　接收移位寄存器 （SCRSR）

SCRSR 为接收串行数据的寄存器。

SCI 按从 LSB （bit0）接收的顺序，将从 RXD 引脚输入的串行数据设置到 SCRSR，并转换为并行数据。接收完 1 个字节的数据时，数据自动向 SCRDR 传送。

不可由 CPU 直接读取 / 写入 SCRSR。

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

13.3.2　接收数据寄存器 （SCRDR）

SCRDR 为保存已接收串行数据的寄存器。

接收完 1 个字节的串行数据时， SCI 将接收的串行数据从接收移位寄存器 （SCRSR）传送至 SCRDR 并保存，结束接收运行。此后， SCRSR 变为可接收状态。

SCRSR 与 SCRDR 为双缓冲器，因此可连续接收运行。

SCRDR 为只读寄存器，不可由 CPU 写入。

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

13.3.3　发送移位寄存器 （SCTSR）

SCTSR 为发送串行数据的寄存器。

SCI 从发送数据寄存器 （SCTDR）暂且向 SCTSR 传送发送数据，并通过从 LSB （bit0）依次发送至 TXD 引脚执行串行数据传送。

发送完 1 个字节的数据时，自动从 SCTDR 向 SCTSR 传送下一个发送数据后，开始发送。但，串行状态寄存器 （SCSSR）的 TDRE 标志置 1 时，不执行从 SCTDR 到 SCTSR 的数据传送。

不可由 CPU 直接读取 / 写入 SCTSR。

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

13.3.4　发送数据寄存器 （SCTDR）

SCTDR 为保存串行发送数据的 8 位寄存器。

检测出发送移位寄存器 （SCTSR）为空时， SCI 将写入 SCTDR 的发送数据传送至 SCTSR 后，开始串行发送。如果在 SCTSR 串行数据发送过程中，对 SCTDR 写入下一个发送数据，则可连续串行发送。

总是可由 CPU 读取 / 写入 SCTDR。

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

13.3.5　串行模式寄存器（SCSMR）

SCSMR 为 8 位寄存器，设定 SCI 的串行通信格式及选择波特率发生器的时钟源。
总是可由 CPU 读取 / 写入 SCSMR。

位：	7	6	5	4	3	2	1	0
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS[1:0]	
初始值：	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说　明
7	C/$\overline{A}$	0	R/W	通信模式 从异步模式和时钟同步模式选择 SCI 的运行模式。 0：异步模式 1：时钟同步模式
6	CHR	0	R/W	字符长度 异步模式的数据长度可选择 7 位 /8 位数据。时钟同步模式时，与 CHR 的设定无关，数据长度固定为 8 位数据。选择 7 位数据时，发送数据寄存器（SCTDR）的 MSB（bit7）不可发送。 0：8 位数据 1：7 位数据
5	PE	0	R/W	奇偶校验允许 异步模式时，选择发送时是否附加奇偶校验位，接收时是否检查奇偶校验位。时钟同步模式时，与 PE 位的设定无关，不附加及检查奇偶校验位。 0：禁止附加及检查奇偶校验位 1：允许附加及检查奇偶校验位 * 【注】 * 如果 PE 位置 1，则发送时，对发送数据附加 O/$\overline{E}$ 位指定的偶校验或奇校验后发送；接收时，检查已接收的奇偶校验位是否为 O/$\overline{E}$ 位指定的偶校验或奇校验。
4	O/$\overline{E}$	0	R/W	奇偶性模式 选择通过奇校验或偶校验附加或检查奇偶校验位。仅在异步模式将 PE 位置 1，并允许附加或检查奇偶校验位时， O/$\overline{E}$ 位的设定才有效。时钟同步模式或异步模式中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：奇校验 如果设定为偶校验，则发送时附加奇偶校验位，对照使奇偶校验位与发送字符中 1 的个数总计为偶数，然后发送；接收时检查奇偶校验位与接收字符中 1 的个数总计是否为偶数。 如果设定为奇校验，则发送时附加奇偶校验位，对照使奇偶校验位与发送字符中 1 的个数总计为奇数，然后发送；接收时检查奇偶校验位与接收字符中 1 的个数总计是否为奇数。

位	位名称	初始值	R/W	说　明
3	STOP	0	R/W	停止位长度 选择异步模式时的停止位长度为 1 位 /2 位。STOP 位的设定仅在异步模式时有效。时钟同步模式时，不附加停止位，因此，本位设定无效。 0：1 位停止位 *1 1：2 位停止位 *2 接收时，与 STOP 位的设定无关，仅检查已接收停止位的第 1 位。停止位的第 2 位为 1 时，作为停止位处理；为 0 时，作为下一个发送字符的开始位处理。 【注】 *1 发送时，在发送字符的最末尾附加 1 位的 1 （停止位）后再发送。 *2 发送时，在发送字符的最末尾附加 2 位的 1 （停止位）后再发送。
2	MP	0	R/W	多处理器模式 （仅异步模式有效） 允许 / 禁止多处理器功能。多处理器模式时， PE、 O/$\overline{E}$ 位的设定无效。 0：禁止多处理器模式 1：允许多处理器模式
1、 0	CKS[1:0]	00	R/W	时钟选择 1、 0 选择内部波特率发生器的时钟源。通过设定 CKS1、 CKS0 位，可从 Pϕ、 Pϕ/4、 Pϕ/16、 Pϕ/64 等 4 种时钟中选择时钟源。 时钟源、位速率寄存器的设定值及波特率的关系，参阅 “13.3.10 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

13.3.6　串行控制寄存器 （SCSCR）

SCSCR 执行 SCI 的发送 / 接收运行、输出异步模式时的串行时钟、允许 / 禁止中断请求并选择发送 / 接收时钟源。

总是可由 CPU 读取 / 写入 SCSCR。

位：	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
初始值：	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说　明
7	TIE	0	R/W	发送中断允许 从发送数据寄存器 （SCTDR）向发送移位寄存器 （SCTSR）传送串行发送数据，串行状态寄存器 （SCSSR）的 TDRE 标志置 1 时，允许 / 禁止产生发送数据空中断 （TXI）请求。 可通过读取 TDRE 标志为 1 后清 0 或将 TIE 清 0，解除 TXI。 0：禁止发送数据空中断 （TXI）请求 1：允许发送数据空中断 （TXI）请求
6	RIE	0	R/W	接收中断允许 从接收移位寄存器 （SCRSR）向接收数据寄存器 （SCRDR）传送串行接收数据， SCSSR 的 RDRF 标志置 1 时，允许 / 禁止产生接收数据满中断 （RXI）请求及接收错误中断 （ERI）请求。 可通过读取 RDRF 标志、或 FER、 PER、 ORER 标志为 1 后清 0，或将 RIE 位清 0，解除 RXI 及 ERI 中断请求。 0：禁止接收数据满中断 （RXI）请求及接收错误 （ERI）请求 1：允许接收数据满中断 （RXI）请求及接收错误 （ERI）请求
5	TE	0	R/W	发送允许 允许 / 禁止开始 SCI 的串行发送运行。 0：禁止发送运行 *1 1：允许发送运行 *2 【注】 *1 SCSSR 的 TDRE 标志固定为 1。 *2 此状态下，如果对 SCTDR 写入发送数据，并将 SCSSR 的 TDRE 标志清 0，则开始串行发送。 另外， TE位置1前，必须设定串行模式寄存器 （SCSMR），确定接收格式。
4	RE	0	R/W	接收允许 允许 / 禁止开始 SCI 的串行接收运行。 0：禁止接收运行 *1 1：允许接收运行 *2 【注】 *1 必须注意：即使 RE 位清 0， RDRF、 FER、 PER、 ORER 的各标志也不受影响，仍保持状态。 *2 此状态下，如果在异步模式检测到开始位、在时钟同步模式检测到同步时钟输入，则开始串行接收。 另外， RE位置1前，必须设定SCSMR，确定接收格式。
3	MPIE	0	R/W	多处理器中断允许 （在异步模式 SCSMR 的 MP=1 时有效） 如果此位置 1，则跳读多处理器位为 0 的数据，禁止置位 SCSSR 的 RDRF、 FER、 ORER 等各状态标志。如果接收多处理器位为 1 的数据，则自动清除此位，返回至通常的接收运行。详情参阅 “13.4.4 多处理器通信功能”。
2	TEIE	0	R/W	发送结束中断允许 发送 MSB 数据时 SCTDR 中无有效发送数据时，允许 / 禁止产生发送结束中断 （TEI）请求。 可通过读取 SCSSR 的 TDRE 标志为 1 后将其与 TEND 标志清 0，或将 TEIE 位清 0，解除 TEI。 0：禁止发送结束中断 （TEI）请求 1：允许发送结束中断 （TEI）请求

位	位名称	初始值	R/W	说　明
1、0	CKE[1:0]	00	R/W	时钟允许 1、0 选择 SCI 的时钟源及设定允许 / 禁止从 SCK 引脚输出时钟。根据 CKE1 位与 CKE0 位的组合，决定将 SCK 引脚设定为串行时钟输出引脚或串行时钟输入引脚。 在时钟同步模式设定为同步时钟输出时，必须将 SCSMR 的 C/$\overline{A}$ 位设定为 1 后再设定 CKE1、 CKE0 位。有关 SCI 时钟源的选择，参照表 13.14。 异步模式 00：内部时钟 /SCK 引脚为输入引脚 （忽略输入信号） 01：内部时钟 /SCK 引脚为时钟输出 *1 10：外部时钟 /SCK 引脚为时钟输入 *2 11：外部时钟 /SCK 引脚为时钟输入 *2 时钟同步模式 00：内部时钟 /SCK 引脚为同步时钟输出 01：内部时钟 /SCK 引脚为同步时钟输出 10：外部时钟 /SCK 引脚为同步时钟输入 11：外部时钟 /SCK 引脚为同步时钟输入 【注】 *1　时钟输出频率为 16 倍位速率 *2　时钟输入频率为 16 倍位速率

13.3.7　串行状态寄存器 （SCSSR）

SCSSR 为 8 位寄存器，内置表示 SCI 运行状态的状态标志。

总是可由 CPU 读取 / 写入 SCSSR。但不可对 TDRE、 RDRF、 ORER、 PER 及 FER 等各标志写入 1。另外，为了将这些位清 0，必须先读取 1。 TEND 标志为只读标志，不可写入。

位：	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
初始值：	1	0	0	0	0	1	0	0
R/W：	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

【注】* 为了清除标志，仅可在读取1后写入0。

位	位名称	初始值	R/W	说　明
7	TDRE	1	R/(W)*	发送数据寄存器空 表示可从发送数据寄存器 （SCTDR）向发送移位寄存器 （SCTSR）传送数据，并可对 SCTDR 写入下一个串行发送数据。 0：表示对 SCTDR 写入了有效发送数据 ［清除条件］ • 读取 TDRE=1 的状态后写入 0 时 • 由 TXI 中断启动 DTC，并且 DTC 的 MRB 的 DISEL 位为 0 时，向 SCTDR 传送发送数据时 1：表示 SCTDR 中无有效发送数据 ［置位条件］ • 上电复位、待机模式时 • SCSCR 的 TE 位为 0 时 • 可从 SCTDR 向 SCTSR 传送数据，并可对 SCTDR 写入数据时

位	位名称	初始值	R/W	说　明
6	RDRF	0	R/(W)*	接收数据寄存器满 表示接收数据已保存至接收数据寄存器（SCRDR）。 0：表示 SCRDR 中未保存有效接收数据 ［清除条件］ • 上电复位、待机模式时 • 读取 RDRF=1 的状态后，写入 0 时 • 由 RXI 中断启动 DTC，并且 DTC 的 MRB 的 DISEL 位为 0 时，从 SCRDR 传送数据时 1：表示 SCRDR 中已保存有效接收数据 ［置位条件］ • 串行接收正常结束，从 SCRSR 向 SCRDR 传送接收数据时 【注】 接收时检测出错误及串行控制寄存器（SCSCR）的 RE 位清 0 时， SCRDR 及 RDRF 标志不受影响，仍保持以前的状态。必须注意：如果在RDRF标志置1的状态下，接收完下一个数据，则会产生溢出错误，并丢失接收数据。
5	ORER	0	R/(W)*	溢出错误 表示接收时产生溢出错误，并异常结束。 0：表示正在接收或正常完成接收 *1 ［清除条件］ • 上电复位、待机模式时 • 读取 ORER=1 的状态后，写入 0 时 1：表示接收时产生溢出错误 *2 ［置位条件］ • 表示在 RDRF=1 的状态，完成下一个串行接收时 【注】 *1 SCSCR 的 RE 位清 0 时， ORER 标志不受影响，仍保持以前的状态。 *2 SCRDR 保持产生溢出错误以前的接收数据，而丢失之后的接收数据。在 ORER=1 的状态，不可继续执行以后的串行接收。
4	FER	0	R/(W)*	帧错误 表示在异步模式接收时产生帧错误，并异常结束。 0：表示正在接收或正常完成接收 *1 ［清除条件］ • 上电复位、待机模式时 • 读取 FER=1 的状态后，写入 0 时 1：表示接收时产生帧错误 ［置位条件］ • 接收结束时， SCI 检查接收数据最末尾的停止位是否为 1，且停止位为 0 时 *2 【注】 *1 将 SCSCR 的 RE 位清 0 时， FER 不受影响，仍保持以前的状态。 *2 2 位停止位模式时，仅判断第 1 位停止位是否为 1，不检查第 2 位停止位。产生帧错误时的接收数据传送至 SCRDR，但 RDRF 标志不置位。在 FER 标志置 1 的状态，不能继续执行以后的串行接收。

位	位名称	初始值	R/W	说　明
3	PER	0	R/(W)*	奇偶校验错误 表示在异步模式接收附加奇偶校验位的数据时，产生奇偶校验错误，并异常结束。 0：表示正在接收或正常完成接收 *1 ［清除条件］ • 上电复位、待机模式时 • 读取 PER=1 的状态后，写入 0 时 1：表示接收时产生奇偶校验错误 *2 ［置位条件］ • 接收时的接收数据和奇偶校验位为 1 的个数合计与串行模式寄存器（SCSMR）的 O/$\overline{E}$ 位指定的偶校验 / 奇校验设定不匹配时 【注】 *1 将 SCSCR 的 RE 位清 0 时， PER 标志不受影响，仍保持以前的状态。 *2 产生奇偶校验错误时的接收数据传送至 SCRDR，但 RDRF 标志不置位。在 PER 标志置 1 的状态，不能继续执行以后的串行接收。
2	TEND	1	R	发送结束 表示发送字符的最末尾位时， SCTDR 中无有效数据，结束发送。 TEND 标志为只读标志，不可写入。 0：表示正在发送 ［清除条件］ • 读取 TDRE=1 的状态后，对 TDRE 标志写入 0 时 1：表示发送结束 ［置位条件］ • 上电复位、待机模式时 • SCSCR 的 TE 位为 0 时 • 发送 1 个字节串行发送字符的最末尾位时， TDRE=1 的情况下 【注】 由 TXI 中断启动 DTC，向 SCTDR 写入数据时， TEND 标志为不定，因此，不得将 TEND 标志用作发送结束标志。
1	MPB	0	R	多处理器位 保存接收帧过程中的多处理器位的值。 SCSCR 的 RE 为 0 时，无变化。
0	MPBT	0	R/W	传送多处理器位 设定附加到发送帧的多处理器位的值。

【注】 * 为了清除标志，仅可在读取 1 后写入 0。

13.3.8　串行端口寄存器 （SCSPTR）

SCSPTR 控制串行通信接口 （SCI）引脚的多路复用端口的输入 / 输出及数据。可对 TXD 引脚写入输出数据，控制串行发送 / 接收的中断，还可通过 bit3 及 bit2 对 SCK 引脚读取数据并写入输出数据。 bit7 控制允许 / 禁止 RXI 中断。 SCSPTR 为 8 位寄存器，总是可由 CPU 读取 / 写入。读取 SCI 引脚的值时，必须使用端口寄存器。详情参阅 “第 19 章 I/O 端口”。

位：	7	6	5	4	3	2	1	0
	EIO	-	-	-	SPB1IO	SPB1DT	SPB0IO	SPB0DT
初始值：	0	0	0	0	0	不定	0	不定
R/W：	R/W	-	-	-	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说　明
7	EIO	0	R/W	仅错误中断 EIO 位为 1 时，即使 RIE 位置 1，也不向 CPU 请求 RXI 中断。此位设定允许 / 禁止 RXI 中断。 0：RIE 位允许 / 禁止 RXI 和 ERI 中断 RIE 位为 1 时，向 INTC 发送 RXI 和 ERI 中断 1：RIE 位为 1 时，仅向 INTC 发送 ERI 中断
6 ～ 4	—	均为 0	—	保留位 读取值、写入值总是为 0。
3	SPB1IO	0	R/W	串行端口的时钟端口输入 / 输出 指定串行端口 SCK 引脚的输入 / 输出。实际将 SCK 引脚作为端口输出引脚输出 SPB1DT 位设定的值时，必须将 SCSMR 的 C/$\overline{A}$ 位与 SCSCR 的 CKE1、 CKE0 位设定为 0。 0：不向 SCK 引脚输出 SPB1DT 位的值 1：向 SCK 引脚输出 SPB1DT 位的值
2	SPB1DT	不定	R/W	串行端口的时钟端口数据 指定串行端口 SCK 引脚的输出数据。由 SPB1IO 位指定输出是否有效 （详情参阅 SPB1IO 位的说明）。输出时，将 SPB1DT 位的值输出到 SCK 引脚。 0：输出数据为低电平 1：输出数据为高电平
1	SPB0IO	0	R/W	串行端口中断输出 与 SPB0DT 位及 SCSCR 的 TE 位一起控制 TXD 引脚。
0	SPB0DT	不定	R/W	串行端口中断数据 与 SPB0IO 位及 SCSCR 的 TE 位一起控制 TXD 引脚。但是，必须先通过引脚功能控制器 （PFC）选择 TXD 引脚功能。

SCSCR 的 TE 位设定值	SPB0IO 位设定值	SPB0DT 位设定值	TXD 引脚状态
0	0	*	SPB0DT 输出无效状态 （初始状态）
0	1	0	低电平输出
0	1	1	高电平输出
1	*	*	根据串行核心逻辑，输出发送数据

【注】　*　Don't care

13.3.9　串行方向控制寄存器 （SCSDCR）

SCSDCR 通过 DIR 位选择 LSB first/MSB first。与串行通信模式无关，仅在 8 位长度时，可选择 LSB first/MSB first。

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

位	位名称	初始值	R/W	说　明
7 ～ 4	—	均为 1	R	保留位 读取值、写入值总是为 1。
3	DIR	0	R/W	数据传送方向 选择串行 / 并行转换格式。发送 / 接收格式为 8 位时有效。 0：以 LSB first 发送 SCTDR 的内容 接收数据按 LSB first 保存至 SCRDR 1：以 MSB first 发送 SCTDR 的内容 接收数据按 MSB first 保存至 SCRDR
2	—	0	R	保留位 读取值、写入值总是为 0。
1	—	1	R	保留位 读取值、写入值总是为 1。
0	—	0	R	保留位 读取值、写入值总是为 0。

13.3.10　位速率寄存器 （SCBRR）

SCBRR 为 8 位寄存器，根据串行模式寄存器 （SCSMR）的 CKS1、 CKS0 位所选择的波特率发生器的运行时钟，设定串行发送 / 接收的位速率。

总是可由 CPU 读取 / 写入 SCBRR。

SCBRR 的设定值通过以下公式计算。

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

〔异步模式〕

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

〔时钟同步模式〕

$$N = \frac{P\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B ：位速率 （bit/s）

N ：波特率发生器的 SCBRR 设定值 （0 ≤ N ≤ 255）
（必须设定满足电特性的值）

Pϕ：外围模块的工作频率 （MHz）

n ：波特率发生器输入时钟 （n=0、 1、 2、 3）
（n 与时钟的关系参照表 13.3）

表 13.3　SCSMR 的设定值

n	时钟	SCSMR 的设定值	
		CKS1	CKS0
0	Pϕ	0	0
1	Pϕ/4	0	1
2	Pϕ/16	1	0
3	Pϕ/64	1	1

异步模式的位速率误差通过以下公式计算。

$$\text{误差（\%）} = \left\{ \frac{P\phi \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

异步模式的 SCBRR 设定例子如表 13.4 ～表 13.6 所示；时钟同步模式的 SCBRR 设定例子如表 13.7 ～表 13.9 所示。

表 13.4　位速率的 SCBRR 设定例子 （异步模式）（1）

位速率（bit/s）	Pϕ（MHz）																	
	10			12			14			16			18			20		
	n	N	误差（%）	n	N	误差（%）	n	N	误差（%）	n	N	误差（%）	n	N	误差（%）	n	N	误差（%）
110	2	177	–0.25	2	212	0.03	2	248	–0.17	3	70	0.03	3	79	–0.12	3	88	–0.25
150	2	129	0.16	2	155	0.16	2	181	0.16	2	207	0.16	2	233	0.16	3	64	0.16
300	2	64	0.16	2	77	0.16	2	90	0.16	2	103	0.16	2	116	0.16	2	129	0.16
600	1	129	0.16	1	155	0.16	1	181	0.16	1	207	0.16	1	233	0.16	2	64	0.16
1200	1	64	0.16	1	77	0.16	1	90	0.16	1	103	0.16	1	116	0.16	1	129	0.16
2400	0	129	0.16	0	155	0.16	0	181	0.16	0	207	0.16	0	233	0.16	1	64	0.16
4800	0	64	0.16	0	77	0.16	0	90	0.16	0	103	0.16	0	116	0.16	0	129	0.16
9600	0	32	–1.36	0	38	0.16	0	45	–0.93	0	51	0.16	0	58	–0.69	0	64	0.16
14400	0	21	–1.36	0	25	0.16	0	29	1.27	0	34	–0.79	0	38	0.16	0	42	0.94
19200	0	15	1.73	0	19	–2.34	0	22	–0.93	0	25	0.16	0	28	1.02	0	32	–1.36
28800	0	10	–1.36	0	12	0.16	0	14	1.27	0	16	2.12	0	19	–2.34	0	21	–1.36
31250	0	9	0.00	0	11	0.00	0	13	0.00	0	15	0.00	0	17	0.00	0	19	0.00
38400	0	7	1.73	0	9	–2.34	0	10	3.57	0	12	0.16	0	14	–2.34	0	15	1.73

表 13.5　位速率的 SCBRR 设定例子（异步模式）（2）

位速率（bit/s）	Pϕ（MHz）																	
	22			24			26			28			30			32		
	n	N	误差（%）	n	N	误差（%）	n	N	误差（%）	n	N	误差（%）	n	N	误差（%）	n	N	误差（%）
110	3	97	–0.35	3	106	–0.44	3	114	0.36	3	123	0.23	3	132	0.13	3	141	0.03
150	3	71	–0.54	3	77	0.16	3	84	–0.43	3	90	0.16	3	97	–0.35	3	103	0.16
300	2	142	0.16	2	155	0.16	2	168	0.16	2	181	0.16	2	194	0.16	2	207	0.16
600	2	71	–0.54	2	77	0.16	2	84	–0.43	2	90	0.16	2	97	–0.35	2	103	0.16
1200	1	142	0.16	1	155	0.16	1	168	0.16	1	181	0.16	1	194	0.16	1	207	0.16
2400	1	71	–0.54	1	77	0.16	1	84	–0.43	1	90	0.16	1	97	–0.35	1	103	0.16
4800	0	142	0.16	0	155	0.16	0	168	0.16	0	181	0.16	0	194	0.16	0	207	0.16
9600	0	71	–0.54	0	77	0.16	0	84	–0.43	0	90	0.16	0	97	–0.35	0	103	0.16
14400	0	47	–0.54	0	51	0.16	0	55	0.76	0	60	–0.39	0	64	0.16	0	68	0.64
19200	0	35	–0.54	0	38	0.16	0	41	0.76	0	45	–0.93	0	48	–0.35	0	51	0.16
28800	0	23	–0.54	0	25	0.16	0	27	0.76	0	29	1.27	0	32	–1.36	0	34	–0.79
31250	0	21	0.00	0	23	0.00	0	25	0.00	0	27	0.00	0	29	0.00	0	31	0.00
38400	0	17	–0.54	0	19	–2.34	0	20	0.76	0	22	–0.93	0	23	1.73	0	25	0.16

表 13.6　位速率的 SCBRR 设定例子（异步模式）（3）

位速率（bit/s）	Pϕ（MHz）											
	34			36			38			40		
	n	N	误差（%）	n	N	误差（%）	n	N	误差（%）	n	N	误差（%）
110	3	150	–0.05	3	159	–0.12	3	168	–0.19	3	177	–0.25
150	3	110	–0.29	3	116	0.16	3	123	–0.24	3	129	0.16
300	2	220	0.16	2	233	0.16	2	246	0.16	3	64	0.16
600	2	110	–0.29	2	116	0.16	2	123	–0.24	2	129	0.16
1200	1	220	0.16	1	233	0.16	1	246	0.16	2	64	0.16
2400	1	110	–0.29	1	116	0.16	1	123	–0.24	1	129	0.16
4800	0	220	0.16	0	233	0.16	0	246	0.16	1	64	0.16
9600	0	110	–0.29	0	116	0.16	0	123	–0.24	0	129	0.16
14400	0	73	–0.29	0	77	0.16	0	81	0.57	0	86	–0.22
19200	0	54	0.62	0	58	–0.69	0	61	–0.24	0	64	0.16
28800	0	36	–0.29	0	38	0.16	0	40	0.57	0	42	0.94
31250	0	33	0.00	0	35	0.00	0	37	0.00	0	39	0.00
38400	0	27	–1.18	0	28	1.02	0	30	–0.24	0	32	–1.36

表 13.7　位速率的 SCBRR 设定例子（时钟同步模式）（1）

位速率（bit/s）	Pϕ（MHz）											
	10		12		14		16		18		20	
	n	N	n	N	n	N	n	N	n	N	n	N
250	3	155	3	187	3	218	3	249				
500	3	77	3	93	3	108	3	124	3	140	3	155
1000	2	155	2	187	2	218	2	249	3	69	3	77
2500	1	249	2	74	2	87	2	99	2	112	2	124
5000	1	124	1	149	1	174	1	199	1	224	1	249
10000	0	249	1	74	1	87	1	99	1	112	1	124
25000	0	99	0	119	0	139	0	159	0	179	0	199
50000	0	49	0	59	0	69	0	79	0	89	0	99
100000	0	24	0	29	0	34	0	39	0	44	0	49
250000	0	9	0	11	0	13	0	15	0	17	0	19
500000	0	4	0	5	0	6	0	7	0	8	0	9
1000000	—	—	0	2	—	—	0	3	—	—	0	4
2500000	0	0*	—	—	—	—	—	—	—	—	0	1
5000000			—	—	—	—	—	—	—	—	0	0*

表 13.8　位速率的 SCBRR 设定例子（时钟同步模式）（2）

位速率（bit/s）	Pϕ（MHz）											
	22		24		26		28		30		32	
	n	N	n	N	n	N	n	N	n	N	n	N
250												
500	3	171	3	187	3	202	3	218	3	233	3	249
1000	3	85	3	93	3	101	3	108	3	116	3	124
2500	2	137	2	149	2	162	2	174	2	187	2	199
5000	2	68	2	74	2	80	2	87	2	93	2	99
10000	1	137	1	149	1	162	1	174	1	187	1	199
25000	0	219	0	239	1	64	1	69	1	74	1	79
50000	0	109	0	119	0	129	0	139	0	149	0	159
100000	0	54	0	59	0	64	0	69	0	74	0	79
250000	0	21	0	23	0	25	0	27	0	29	0	31
500000	0	10	0	11	0	12	0	13	0	14	0	15
1000000	—	—	0	5	—	—	0	6	—	—	0	7
2500000	—	—	—	—	—	—	—	—	0	2	—	—
5000000	—	—	—	—	—	—	—	—	—	—	—	—

表 13.9　位速率的 SCBRR 设定例子（时钟同步模式）（3）

位速率（bit/s）	Pϕ（MHz）							
	34		36		38		40	
	n	N	n	N	n	N	n	N
250								
500								
1000	3	132	3	140	3	147	3	155
2500	2	212	2	224	2	237	2	249
5000	2	105	2	112	2	118	2	124
10000	1	212	1	224	1	237	1	249
25000	1	84	1	89	1	94	1	99
50000	0	169	0	179	0	189	0	199
100000	0	84	0	89	0	94	0	99
250000	0	33	0	35	0	37	0	39
500000	0	16	0	17	0	18	0	19
1000000	—	—	0	8	—	—	0	9
2500000	—	—	—	—	—	—	0	3
5000000	—	—	—	—	—	—	0	1

【注】　必须尽量将误差设定在 1％以内。

【符号说明】

空栏　：不可设定。

—　：可设定，但有误差。

*　：不可连续发送 / 接收。

使用波特率发生器时，异步模式下各频率的最大位速率如表 13.10 所示。外部时钟输入时的最大位速率如表 13.11 和表 13.12 所示。

表 13.10　使用波特率发生器时的各频率的最大位速率（异步模式）

Pϕ（MHz）	最大位速率（bit/s）	设定值	
		n	N
10	312500	0	0
12	375000	0	0
14	437500	0	0
16	500000	0	0
18	562500	0	0
20	625000	0	0
22	687500	0	0
24	750000	0	0
26	812500	0	0
28	875000	0	0
30	937500	0	0
32	1000000	0	0
34	1062500	0	0
36	1125000	0	0
38	1187500	0	0
40	1250000	0	0

表 13.11　外部时钟输入时的最大位速率（异步模式）

Pϕ（MHz）	外部输入时钟（MHz）	最大位速率（bit/s）
10	2.5000	156250
12	3.0000	187500
14	3.5000	218750
16	4.0000	250000
18	4.5000	281250
20	5.0000	312500
22	5.5000	343750
24	6.0000	375000
26	6.5000	406250
28	7.0000	437500
30	7.5000	468750
32	8.0000	500000
34	8.5000	531250
36	9.0000	562500
38	9.5000	593750
40	10.0000	625000

表 13.12　外部时钟输入时的最大位速率（时钟同步模式）

Pϕ（MHz）	外部输入时钟（MHz）	最大位速率（bit/s）
10	1.6667	1666666.7
12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0
20	3.3333	3333333.3
22	3.6667	3666666.7
24	4.0000	4000000.0
26	4.3333	4333333.3
28	4.6667	4666666.7
30	5.0000	5000000.0
32	5.3333	5333333.3
34	5.6667	5666666.7
36	6.0000	6000000.0
38	6.3333	6333333.3
40	6.6667	6666666.7

13.4　运行说明

13.4.1　概要

SCI 可通过异步模式（以字符为单位取得同步并通信）和时钟同步模式（通过时钟脉冲取得同步并通信）两种方式执行串行通信。

如表 13.13 所示，由串行模式寄存器（SCSMR）选择异步模式、时钟同步模式及发送格式。如表 13.14 所示，SCI 的时钟源由 SCSMR 的 C/$\overline{A}$ 位及串行控制寄存器（SCSCR）的 CKE1、CKE0 位的组合决定。

(1)　异步模式

- 数据长度：可选择7位/8位
- 可选择附加奇偶校验位及1位或2位停止位（由这些组合决定发送/接收格式及字符长度）
- 接收时，可检测帧错误、奇偶校验错误、溢出错误及中断
- SCI的时钟源：可选择内部时钟/外部时钟
  选择内部时钟时：SCI可通过波特率发生器的时钟运行，并输出频率为16倍位速率的时钟
  选择外部时钟时：需输入频率为16倍位速率的时钟（不使用内部波特率发生器）

(2)　时钟同步模式

- 发送/接收格式：固定为8位数据
- 接收时，可检测溢出错误
- SCI的时钟源：可选择内部时钟/外部时钟
  选择内部时钟时：SCI通过波特率发生器的时钟运行，并向外部输出同步时钟
  选择外部时钟时：不使用内部波特率发生器，通过输入的同步时钟运行

表 13.13　SCSMR 的设定值与串行发送 / 接收格式

SCSMR 的设定值				模式	SCI 的发送 / 接收格式		
bit7	bit6	bit5	bit3		数据长度	奇偶校验位	停止位长度
C/$\overline{A}$	CHR	PE	STOP				
1	0	0	0	异步模式	8 位数据	无	1 位
			1				2 位
		1	0			有	1 位
			1				2 位
	1	0	0		7 位数据	无	1 位
			1				2 位
		1	0			有	1 位
			1				2 位
1	x	x	x	时钟同步模式	8 位数据	无	无

【符号说明】 x：Don't care

表 13.14　SCSMR、 SCSCR 的设定与 SCI 时钟源的选择

SCSMR	SCSCR 的设定值		模式	时钟源	SCK 引脚功能
bit7	bit1	bit0			
C/$\overline{A}$	CKE1	CKE0			
0	0	0	异步模式	内部	SCI 不使用 SCK 引脚
		1			输出频率为 16 倍位速率的时钟
	1	0		外部	输入频率为 16 倍位速率的时钟
		1			
1	0	0	时钟同步模式	内部	输出同步时钟
		1			
	1	0		外部	输入同步时钟
		1			

13.4.2　异步模式时的运行

异步模式可发送 / 接收数据后附加开始位 （表示通信开始）与结束位 （表示通信结束）的字符，并且以 1 个字符为单位取得同步并执行串行通信。

SCI 内部具有独立的发送部与接收部，因此可执行全双工通信。另外，发送部与接收部均为双缓冲结构，在发送 / 接收过程中可读取 / 写入数据，因此可连续发送 / 接收。

异步串行通信的一般格式如图 13.2 所示。

异步串行通信时，通信线路通常保持标记状态 （高电平）。 SCI 监视通信线路，将空格 （低电平）处视为开始位，开始串行通信。

按照从开始位 （低电平）开始，数据 （LSB first：最低位起）、奇偶校验位 （高 / 低电平）、最后到停止位 （高电平）的顺序构成串行通信的 1 个字符。

异步模式接收时，SCI 在开始位的下降沿执行同步化，SCI 通过 1 位间隔为 16 倍频率的第 8 个时钟执行数据采样，因此，在各位的中部获取通信数据。

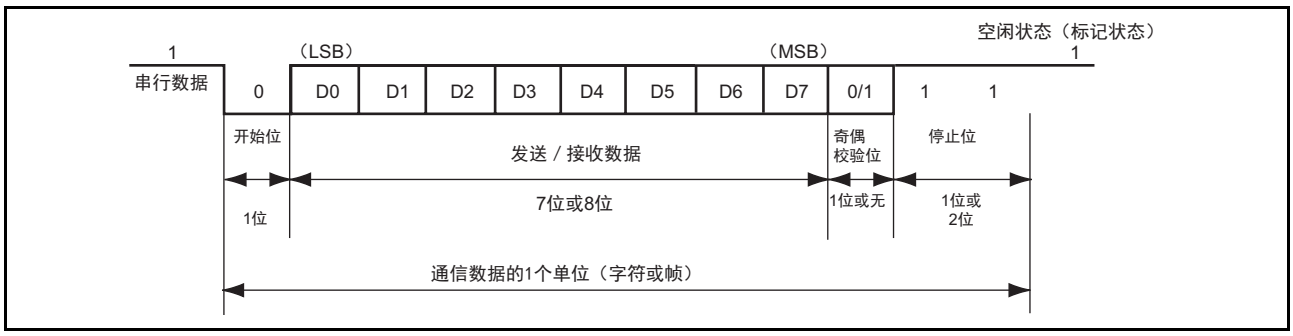


图 13.2　异步通信的数据格式
（8 位数据 / 有奇偶校验位 /2 位停止位的例子）

(1)　发送 / 接收格式

在异步模式可设定的发送 / 接收格式如表 13.15 所示。
发送 / 接收格式有 12 种，可通过设定串行模式寄存器 （SCSMR）选择。

表 13.15　串行发送 / 接收格式 （异步模式）

SCSMR的设定				串行发送/接受格式及帧长度											
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	S	8位数据								STOP		
0	0	0	1	S	8位数据								STOP	STOP	
0	1	0	0	S	8位数据								P	STOP	
0	1	0	1	S	8位数据								P	STOP	STOP
1	0	0	0	S	7位数据							STOP			
1	0	0	1	S	7位数据							STOP	STOP		
1	1	0	0	S	7位数据							P	STOP		
1	1	0	1	S	7位数据							P	STOP	STOP	
0	x	1	0	S	8位数据								MPB	STOP	
0	x	1	1	S	8位数据								MPB	STOP	STOP
1	x	1	0	S	7位数据							MPB	STOP		
1	x	1	1	S	7位数据							MPB	STOP	STOP	

【符号说明】

S　　：开始位
STOP ：停止位
P　　：奇偶校验位
MPB　：多处理器位
x　　：Don't care

(2)　时钟

通过设定 SCSMR 的 C/$\overline{A}$ 位与串行控制寄存器 （SCSCR）的 CKE1、 CKE0 位，可从内部波特率发生器生成的内部时钟或从 SCK 引脚输入的外部时钟中，选择 SCI 的发送 / 接收时钟。 SCI 时钟源的选择参照表 13.14。

向 SCK 引脚输入外部时钟时，必须输入频率为 16 倍位速率的时钟。

以内部时钟运行时，可从 SCK 引脚输出时钟，此时，输出时钟的频率为位速率的 16 倍。

(3)　数据的发送 / 接收运行

- SCI的初始化 （异步模式）

发送 / 接收数据前，必须先将 SCSCR 的 TE 位及 RE 位清 0，然后按以下顺序初始化 SCI。

更改运行模式、通信格式时，必须先将 TE 位及 RE 位清 0，然后按以下顺序更改。 TE 位清 0 时， TDRE 标志置 1，初始化发送移位寄存器 （SCTSR）。必须注意：即使 RE 位清 0，仍保持 RDRF、 PER、 FER、 ORER 各标志及接收数据寄存器 （SCRDR）的内容。

使用外部时钟时运行不稳定，因此不得在包括初始化的运行过程中停止时钟。

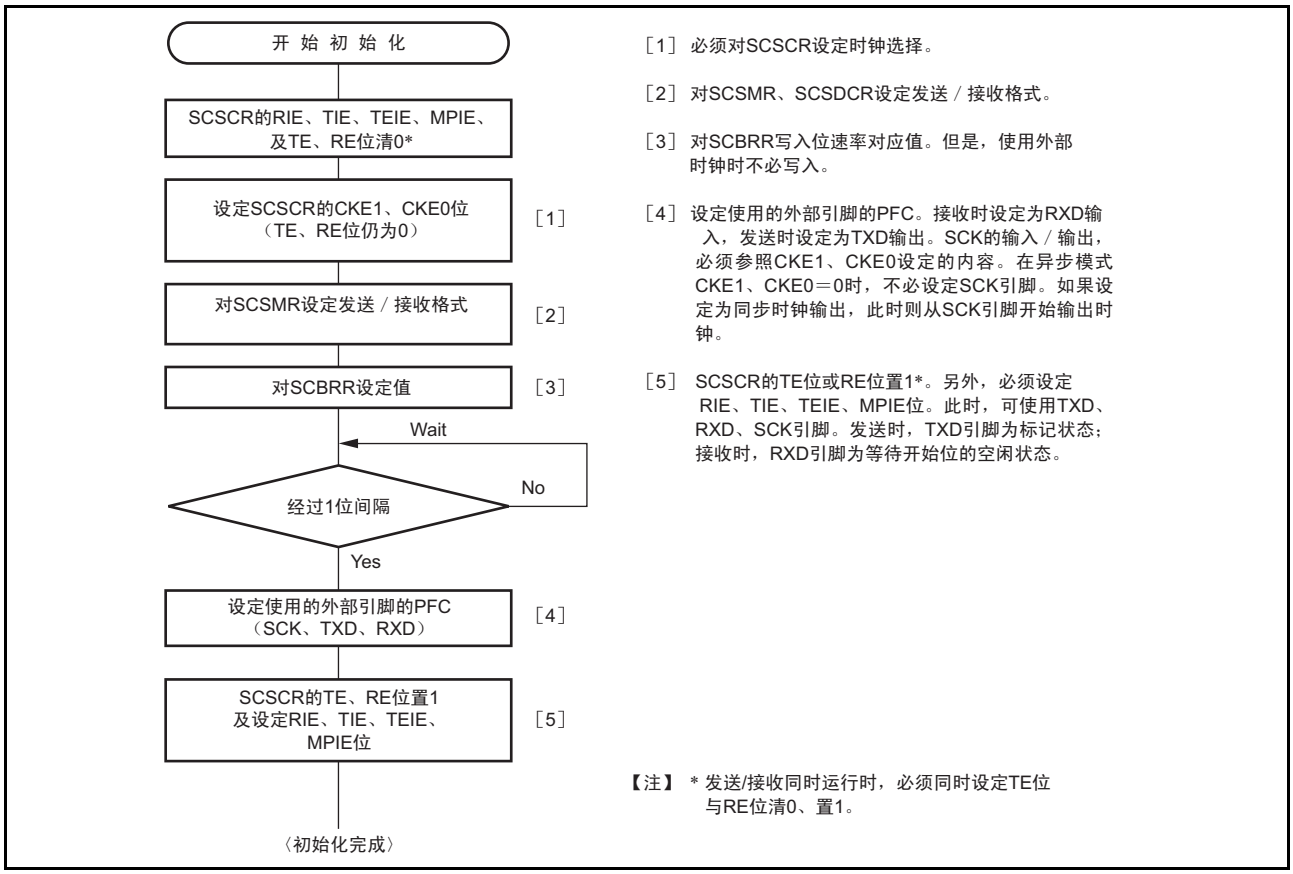


图 13.3　SCI 的初始化流程图例子

• 串行数据发送（异步模式）

串行发送的流程图例子如图 13.4 所示。

SCI 设定为可发送运行状态后，必须按以下顺序执行串行数据发送。

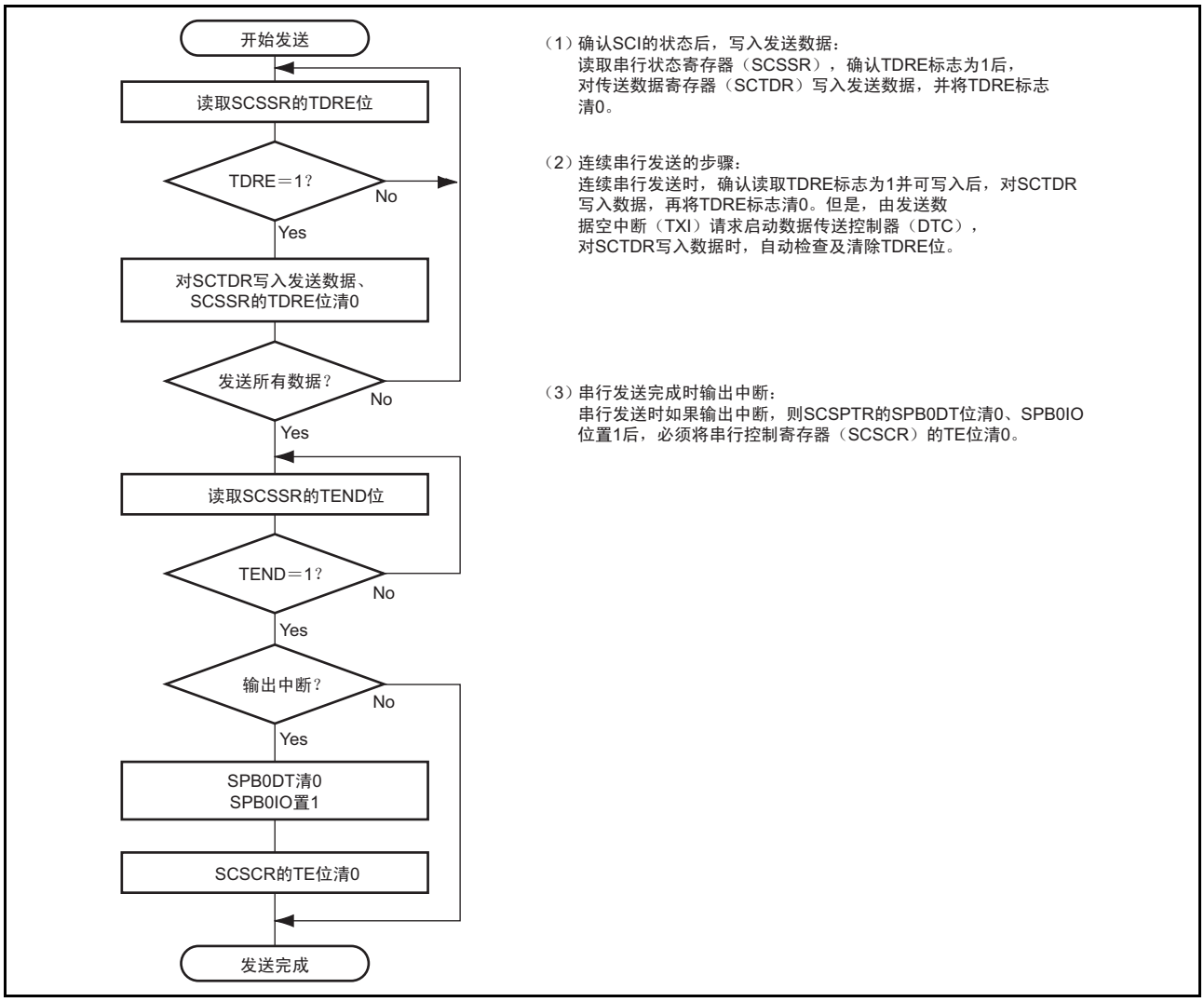


图 13.4　串行发送的流程图例子

串行发送时的 SCI 运行如下：

1. SCI监视串行状态寄存器 （SCSSR）的TDRE标志，为0时，识别为已对发送数据寄存器 （SCTDR）写入数据，并从SCTDR向发送移位寄存器 （SCTSR）传送数据。
2. 从SCTDR向SCTSR传送数据后， TDRE标志置1，开始发送。
   此时，如果串行控制寄存器 （SCSCR）的TIE位置1，则产生发送数据空中断 （TXI）请求。

按以下顺序从 TXD 引脚发送串行发送数据。

(a) 开始位：输出1位的0。
(b) 发送数据：从LSB依次输出8位或7位数据。
(c) 奇偶校验位或多处理器位：输出1位奇偶校验位 （偶校验或奇校验）或1位多处理器位。
也可选择不输出奇偶校验位或多处理器位的格式。
(d) 停止位：输出1位或2位的1 （停止位）。
(e) 标记状态：持续输出1，直至送出下一次发送的开始位。

3. SCI在发送停止位的时序检查TDRE标志。
   TDRE标志为0时，从SCTDR向SCTSR传送数据，送出停止位后，开始下一帧的串行发送。
   TDRE标志为1时，串行状态寄存器 （SCSSR）的TEND标志置1，送出停止位后，变为输出1的标记状态。此时，如果SCSCR的TEIE位置1，则产生TEI请求。

异步模式发送时的运行例子如图 13.5 所示。

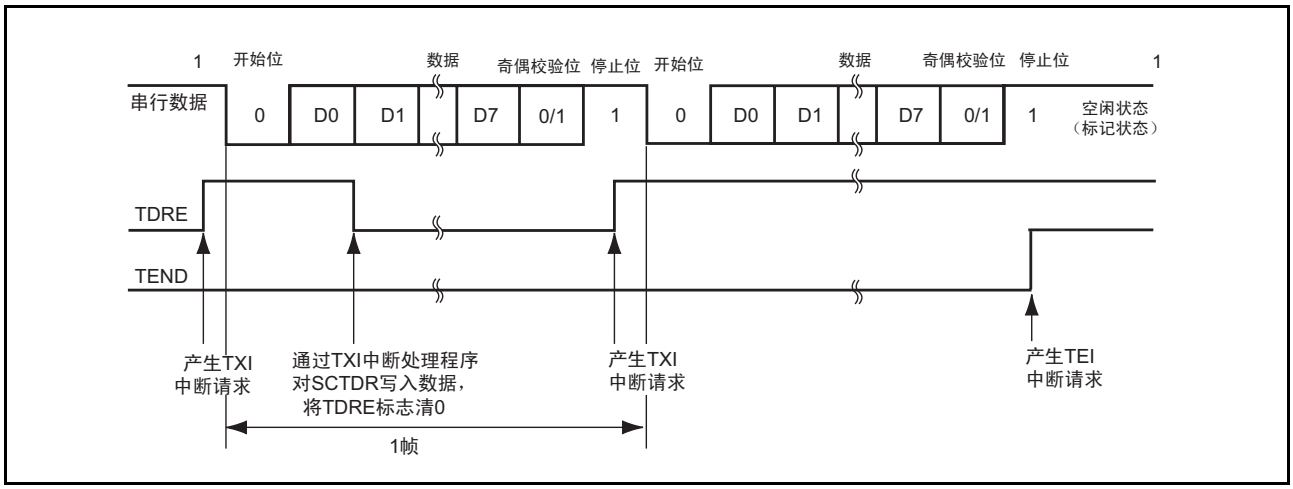


图 13.5　异步模式发送时的运行例子
（8 位数据 / 有奇偶校验位 /1 位停止位的例子）

• 串行数据接收（异步模式）

串行接收的流程图例子如图 13.6 所示。

SCI 设定为可接收运行状态后，必须按以下顺序执行串行数据接收。

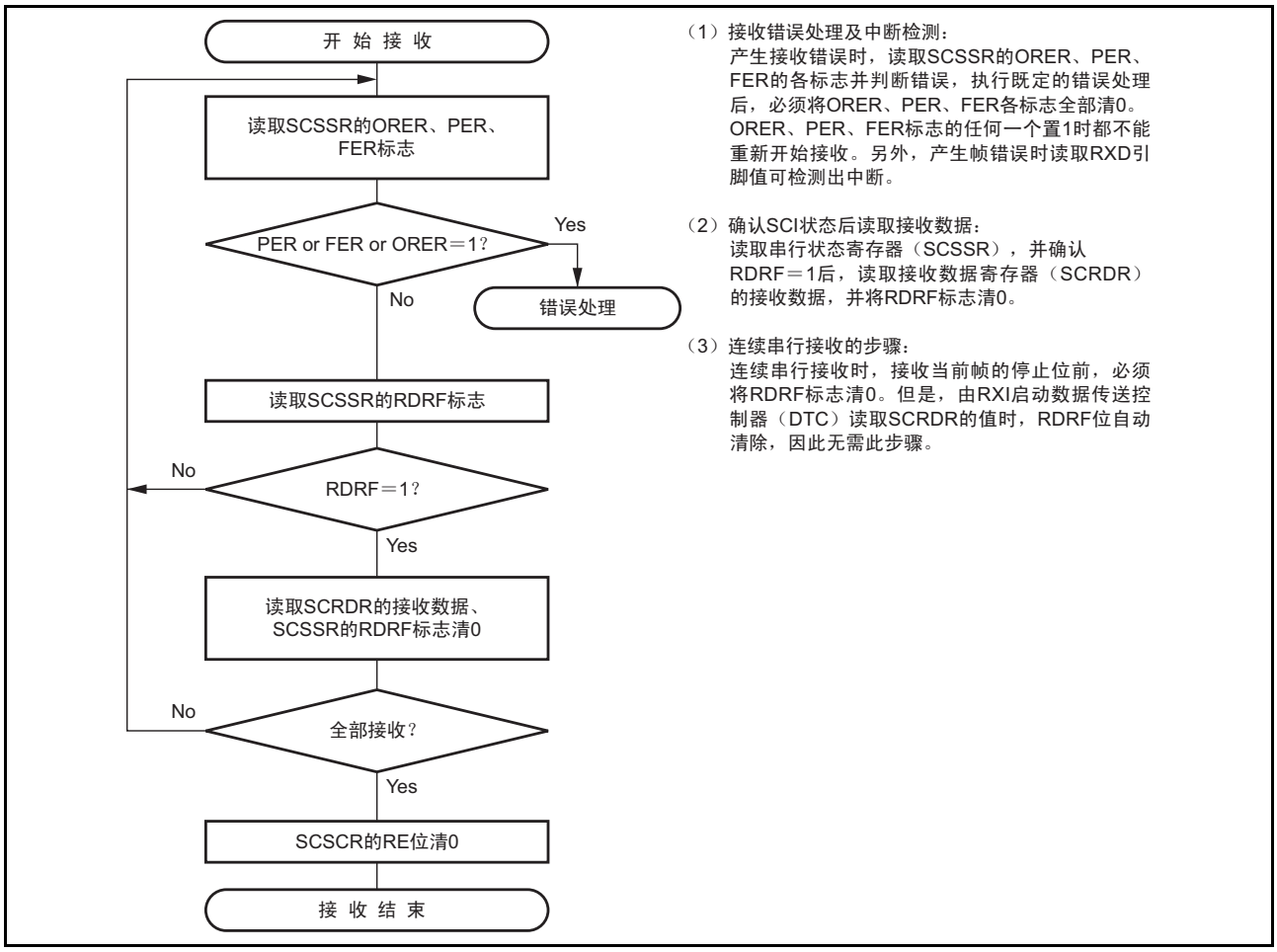


图 13.6　串行接收的流程图例子（1）

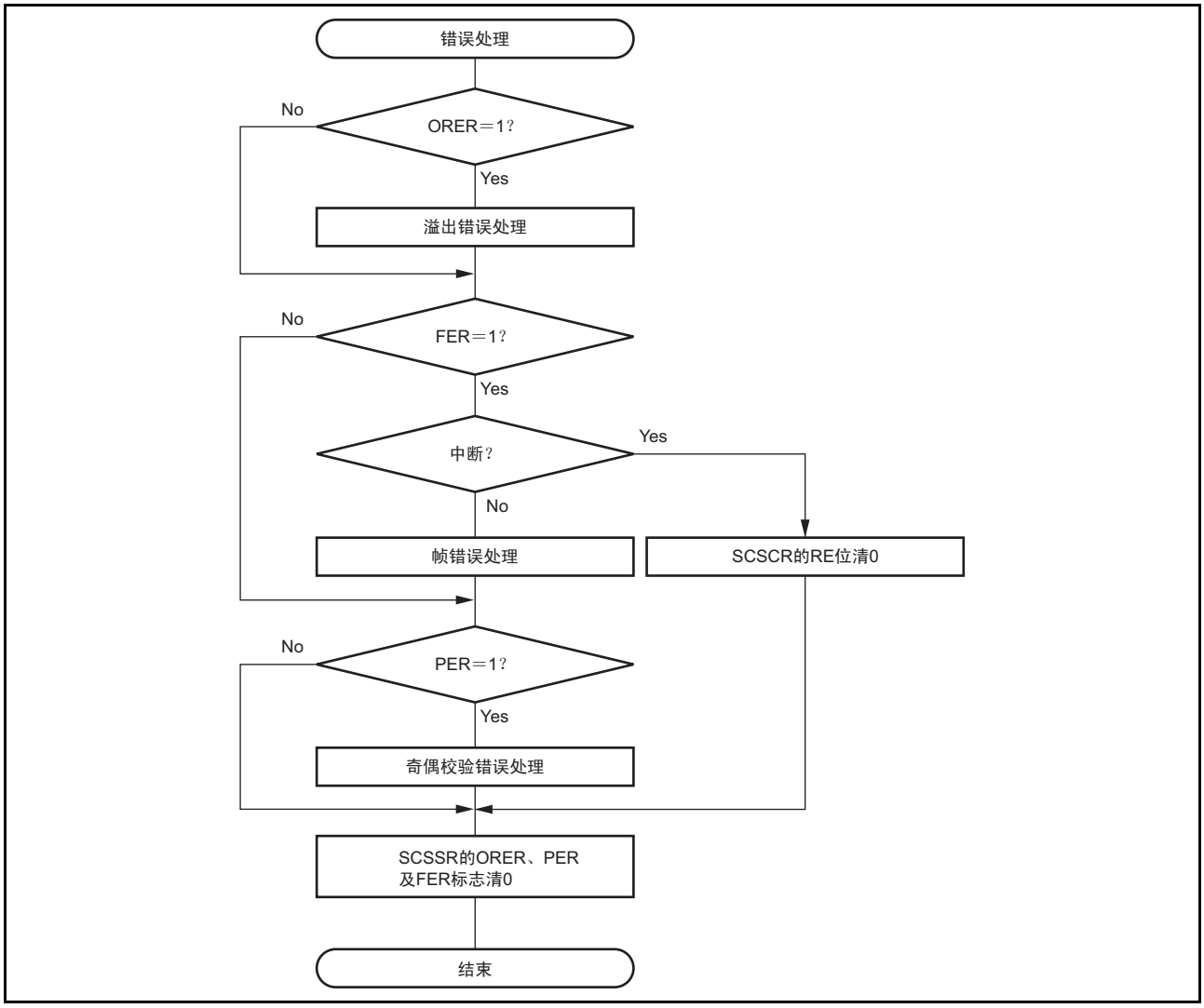


图 13.6 串行接收的流程图例子（2）

接收时的 SCI 运行如下：

1. SCI 监视通信线路，如果检测出开始位的 0，则使内部同步，并开始接收。
2. 按照从 SCRSR 的 LSB 到 MSB 的顺序保存接收的数据。
3. 接收奇偶校验位及停止位。

接收后， SCI 执行如下检查：

(a) 检查奇偶校验：检查接收数据中 1 的个数，并检查其是否为串行模式寄存器（SCSMR）的 O/$\overline{E}$ 位设定的偶/奇校验。

(b) 检查停止位：检查停止位是否为 1。
但是， 2 位停止位时，只检查第 1 位的停止位。

(c) 检查状态：检查 RDRF 标志是否为 0、是否可从接收移位寄存器（SCRSR）向 SCRDR 传送接收数据。

上述检查均通过时， RDRF 标志置 1，接收数据保存至 SCRDR。
如果由错误检查产生接收错误，则执行表 13.16 所示的运行。

【注】 在产生接收错误的状态，不可执行以后的接收运行。
接收时， RDRF 标志不置 1，因此，必须将错误标志清 0。

4. RDRF 标志为 1 时，如果 SCSPTR 的 EIO 位清 0、 SCSCR 的 RIE 位置 1，则产生接收数据满中断（RXI）请求。
另外， ORER、 PER 及 FER 标志的任意一个位为 1 时，如果 SCSCR 的 RIE 位置 1，则产生接收错误中断（ERI）请求。

表 13.16　接收错误与产生条件

接收错误名称	简称	产生条件	数据传送
溢出错误	ORER	在 SCSSR 的 RDRF 标志置 1 的情况下，完成下一个数据接收时	不从 SCRSR 向 SCRDR 传送接收数据
帧错误	FER	停止位为 0 时	从 SCRSR 向 SCRDR 传送接收数据
奇偶校验错误	PER	SCSMR 设定的偶 / 奇校验的设定与接收数据不同时	从 SCRSR 向 SCRDR 传送接收数据

异步模式接收时的运行例子如图 13.7 所示。

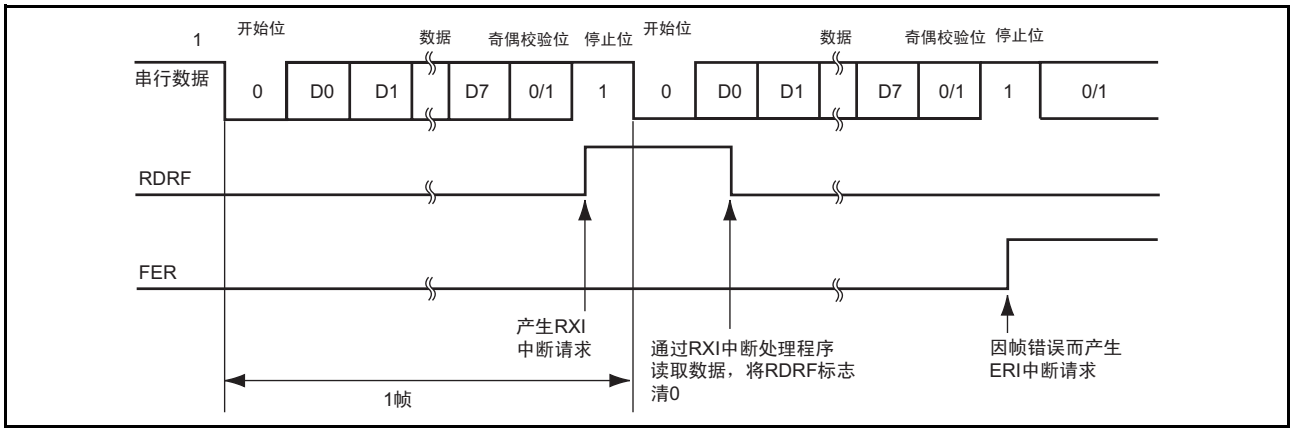


图 13.7　SCI 接收时的运行例子
（8 位数据 / 有奇偶校验位 /1 位停止位的例子）

13.4.3　时钟同步模式时的运行

时钟同步模式，是与时钟脉冲同步发送 / 接收数据的模式，适用于高速串行通信。

在 SCI 内部具有独立的发送部与接收部，因此，通过共用时钟可执行全双工通信。

另外，发送部与接收部均为双缓冲结构，可在发送 / 接收过程中读取 / 写入数据，可连续发送 / 接收。

时钟同步串行通信的一般格式如图 13.8 所示。

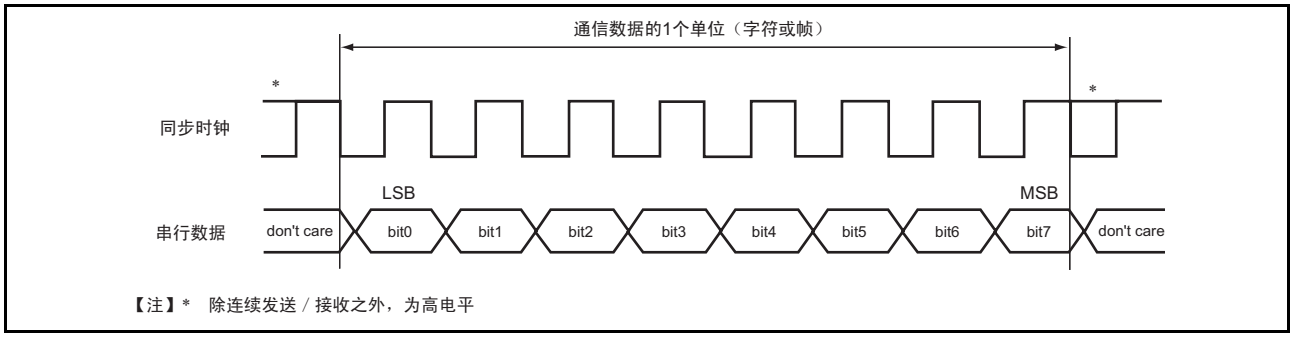


图 13.8　时钟同步通信的数据格式

时钟同步串行通信时，从同步时钟的下降沿到下一个下降沿期间输出通信线路的数据。另外，在同步时钟的上升沿保证数据有效。

按从 LSB 到 MSB 的顺序，输出串行通信的 1 个字符。 MSB 输出后的通信线路状态保持 MSB 的状态。

时钟同步模式时， SCI 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。

不可附加奇偶校验位。

(2)　时钟

通过设定 SCSMR 的 C/$\overline{A}$ 位与 SCSCR 的 CKE1、 CKE0 位，可选择内部波特率发生器生成的内部时钟或从 SCK 引脚输入的外部同步时钟。有关 SCI 时钟源的选择，参照表 13.14。

通过内部时钟运行时，从 SCK 引脚输出同步时钟。

发送 / 接收 1 个字符时，同步时钟输出 8 个脉冲；不发送 / 接收时，固定为高电平。但是，只接收时，输出同步时钟直至产生溢出或 RE 位清 0。要执行 n 个字符的接收运行时，必须将时钟源设置为外部时钟。使用内部时钟时，必须先设置 RE=1 并且 TE=1，然后在发送 n 个字符的虚拟数据的同时接收 n 个字符。

(3)　数据的发送 / 接收运行

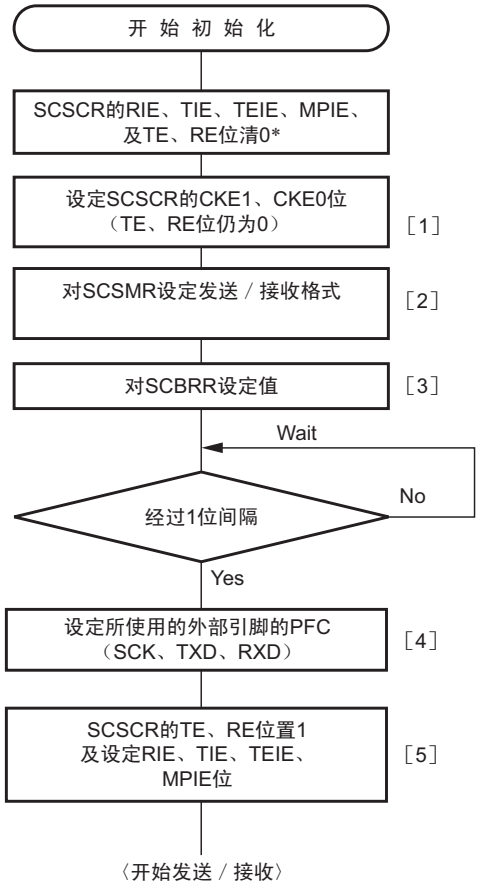
- SCI的初始化 （时钟同步模式）

发送 / 接收数据前，必须将串行控制寄存器 （SCSCR）的 TE 及 RE 位清 0 后，按以下顺序初始化 SCI。

更改模式及通信格式时，必须将 TE 及 RE 位清 0 后，按以下顺序执行。如果 TE 位清 0，则 TDRE 标志置 1，初期化发送移位寄存器 （SCTSR）。

必须注意：即使 RE 位清 0，仍保持 RDRF、 PER、 FER、 ORER 各标志及接收数据寄存器 （SCRDR）的内容。

SCI 初始化的流程图例子如图 13.9 所示。



[1] 必须对SCSCR设定选择时钟。

[2] 对SCSMR设定发送／接收格式。

[3] 对SCBRR写入位速率对应的值。但是，使用外部时钟时，无需执行此操作。

[4] 设定所使用的外部引脚的PFC。接收时，设定为RXD输入；发送时，设定为TXD输出。另外，SCK的输入／输出，必须参照CKE1、CKE0设定的内容而设定。

[5] SCSCR的TE位或RE位置1*。另外，必须设定RIE、TIE、TEIE、MPIE位。此时TXD、RXD、SCK引脚为可使用状态。发送时，TXD引脚为标记状态。在时钟同步模式接收且设定为同步时钟输出（时钟主控器）时，此时从SCK引脚开始输出时钟。

【注】* 发送／接收同时运行时，必须同时设定TE位与RE位的清0、置1。

图 13.9　SCI 初始化的流程图例子

• 串行数据发送 （时钟同步模式）

串行发送的流程图例子如图 13.10 所示。

SCI 设定为可发送运行状态后，必须按以下顺序执行串行数据发送。

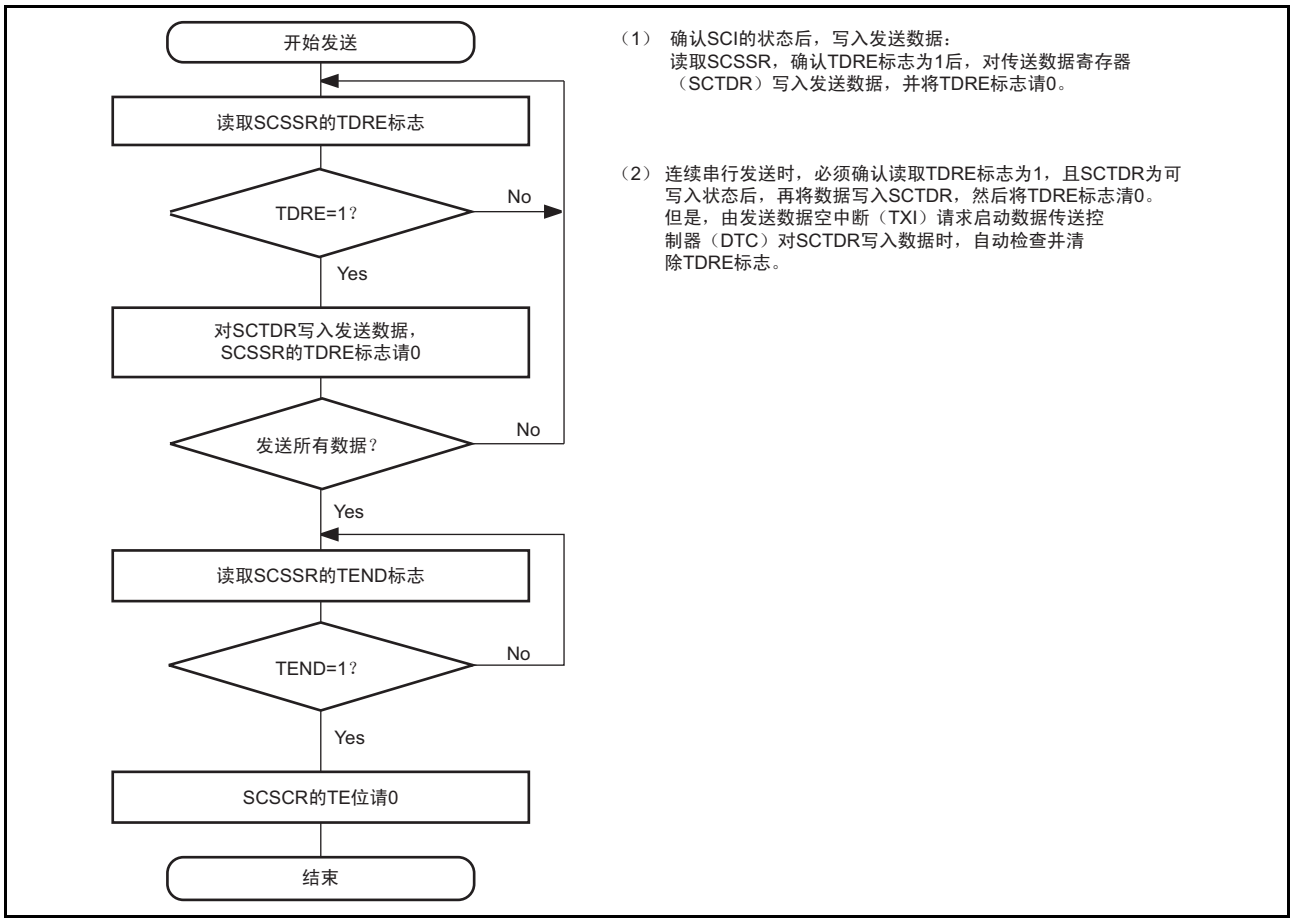


图 13.10　串行发送的流程图例子

串行发送时的 SCI 运行如下：

1. SCI监视串行状态寄存器 （SCSSR）的TDRE标志，为0时，识别已对发送数据寄存器 （SCTDR）写入数据，并从SCTDR向发送移位寄存器 （SCTSR）传送数据。
2. 从SCTDR向SCTSR传送数据后， TDRE标志置1，开始发送。
   此时，如果串行控制寄存器 （SCSCR）的发送数据空中断允许位(TIE)置1，则产生发送数据空中断（TXI）请求。
   设定为时钟输出模式时， SCI输出8个同步时钟脉冲。
   设定为外部时钟时，与输入时钟同步输出数据。
   按LSB （bit0）～MSB （bit7）的顺序，从TXD引脚发送串行发送数据。
3. SCI在发送MSB （bit7）的时序检查TDRE标志。
   TDRE标志为0时，从SCTDR向SCTSR传送数据，并开始串行发送下一帧。
   TDRE标志为1时，串行状态寄存器 （SCSSR）的TEND标志置1，发送MSB （bit7）后，发送数据引脚 （TXD引脚）保持该状态。
   此时，如果SCSCR的发送结束中断允许位 （TEIE）置1，则产生发送结束中断请求 （TEI）。
4. 串行发送结束后， SCK引脚固定为高电平。

SCI 发送时的运行例子如图 13.11 所示。

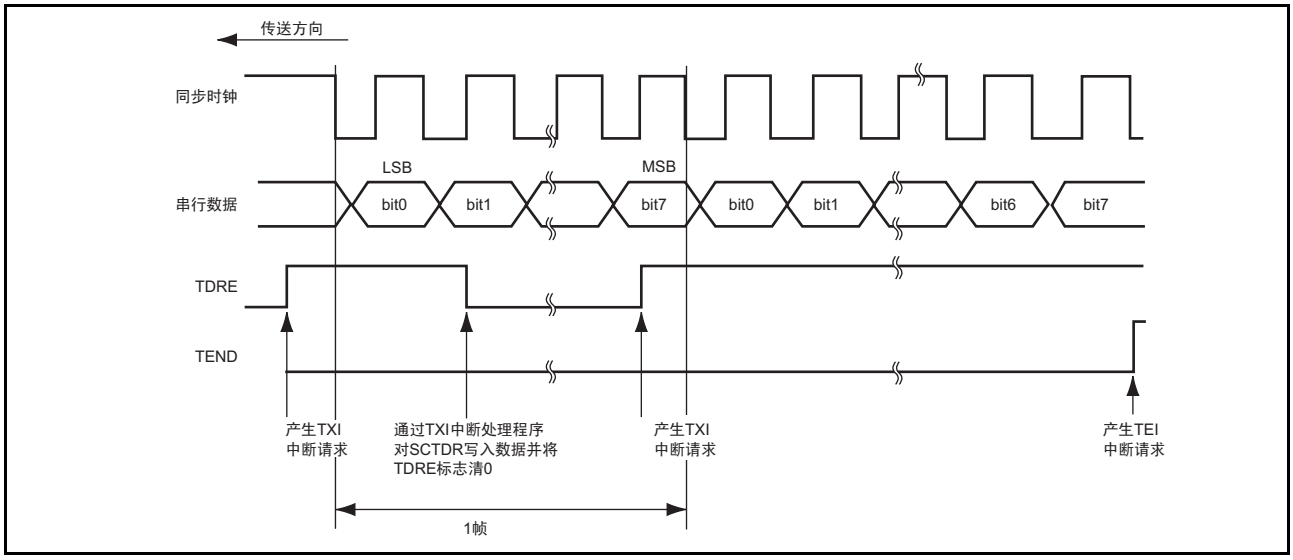


图 13.11　SCI 发送时的运行例子

- 串行数据接收（时钟同步模式）

串行接收的流程图例子如图 13.12 所示。

SCI 设定为可接收运行状态后，必须按以下顺序执行串行数据接收。

运行模式从异步模式转换到时钟同步模式时，必须确认 ORER、 PER 及 FER 各标志是否清 0。如果 FER 与 PER 标志置 1，则 RDRF 标志不被置位。另外，不能执行接收运行。

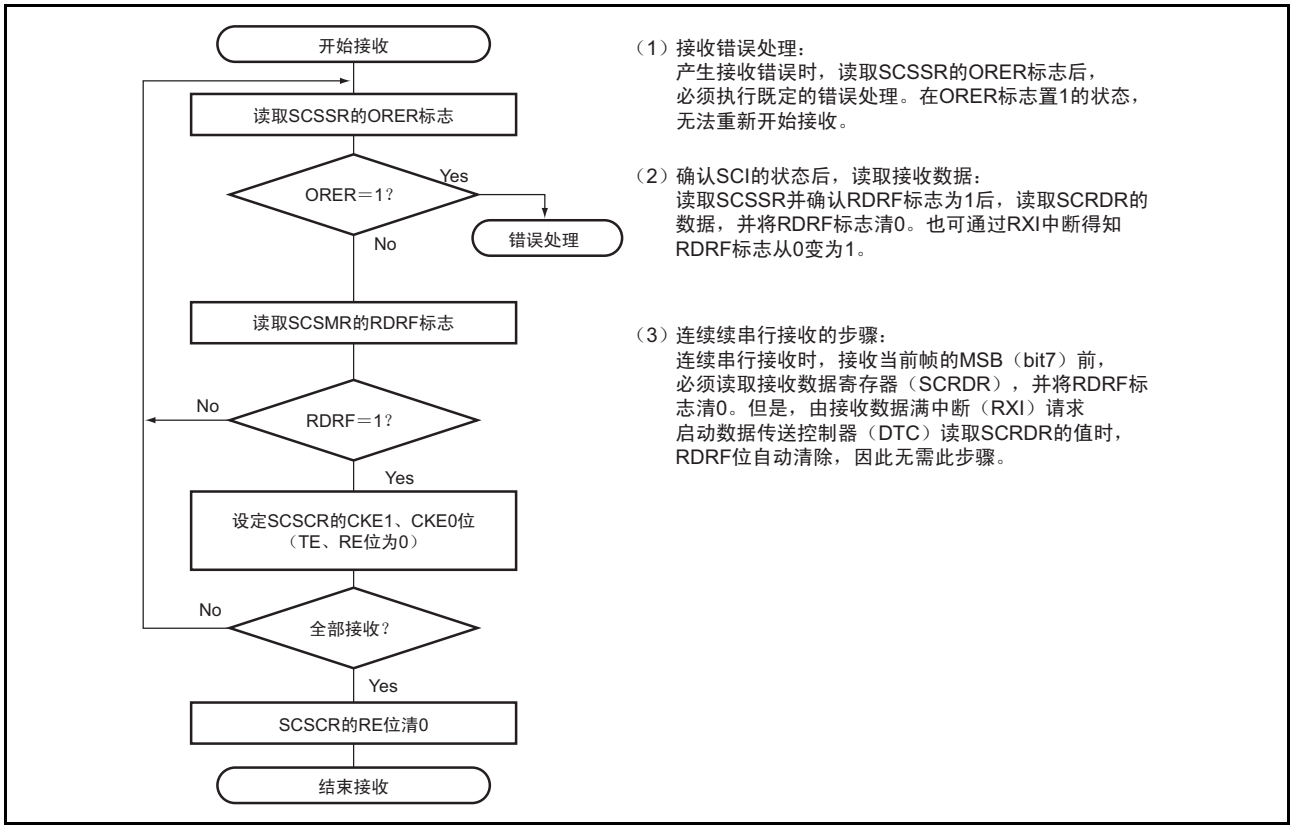


图 13.12　串行数据的接收流程图例子（1）

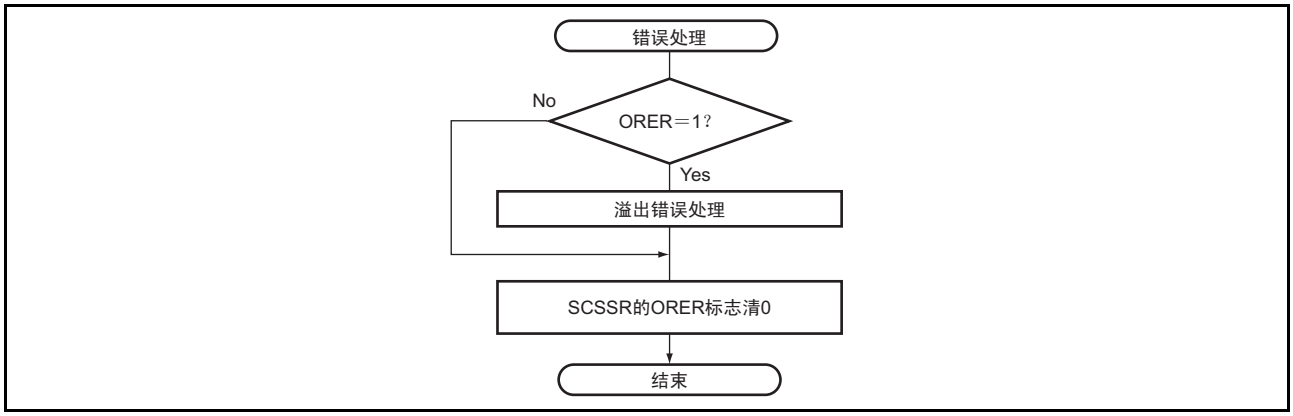


图 13.12 串行数据接收的流程图例子 （2）

接收时的 SCI 运行如下：

1. SCI 与同步时钟的输入或输出同步开始接收。
2. 按照从接收移位寄存器 （SCRSR）的LSB到MSB的顺序，保存接收的数据。
   接收后， SCI检查RDRF标志是否为0、是否为可从SCRSR向接收数据寄存器 （SCRDR）传送接收数据的状态。
   检查通过后， RDRF标志置1，并将接收数据保存至SCRDR。
   如果由错误检查产生接收错误，则如表13.16所示运行，在此状态，不可执行以后的发送/接收。
   另外，接收时RDRF标志不置1，因此必须将标志清0。
3. RDRF标志为1时，如果串行控制寄存器 （SCSCR）的RIE位置1，则产生接收数据满中断 （RXI）请求。
   另外， ORER标志为1时，如果SCSCR的RIE位置1，则产生接收错误中断 （ERI）请求。

SCI 接收时的运行例子如图 13.13 所示。

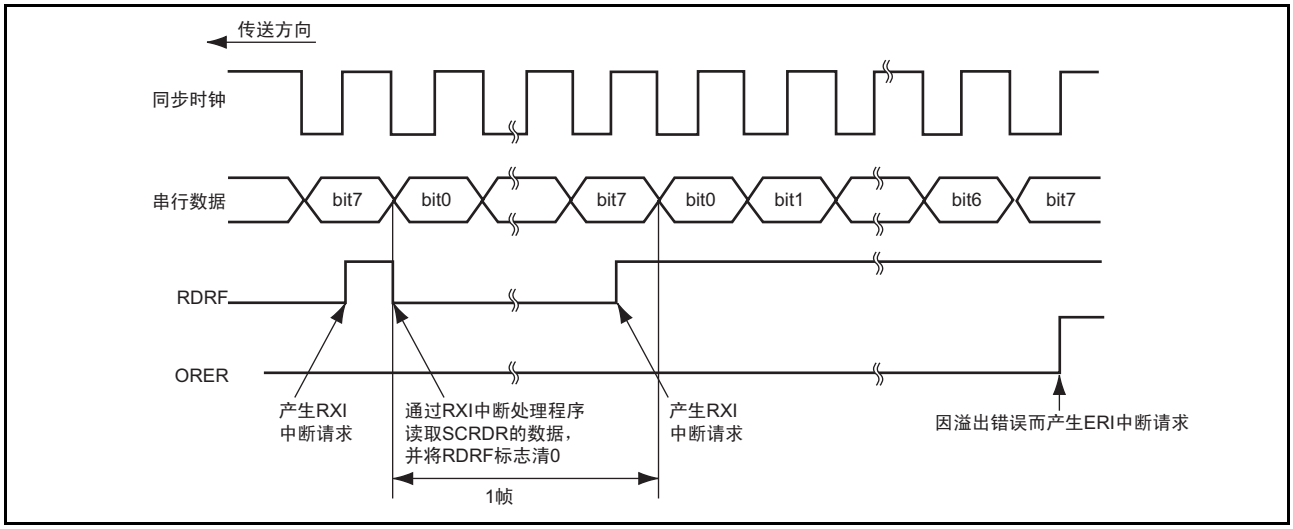
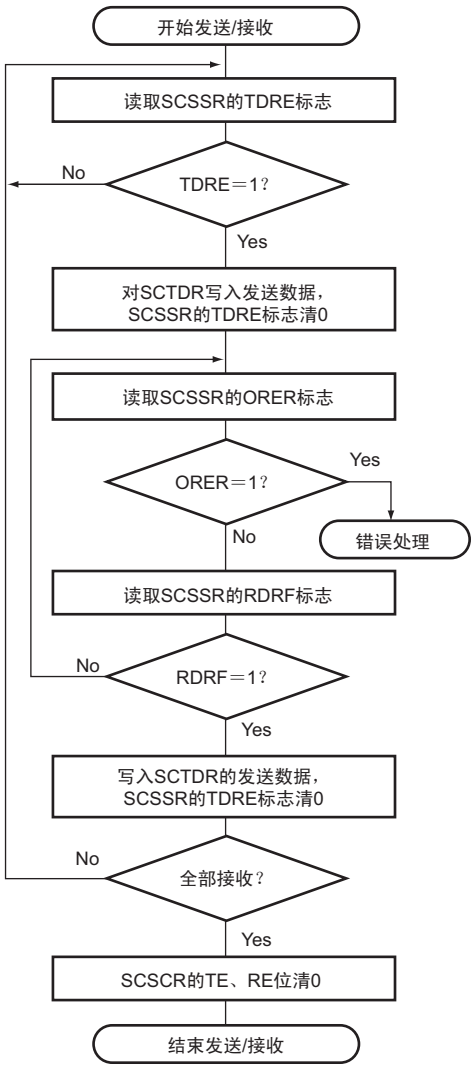


图 13.13　SCI 接收时的运行例子

• 串行数据发送/接收同时运行（时钟同步模式）

串行发送 / 接收同时运行的流程图例子如图 13.14 所示。

SCI 设定为可发送 / 接收状态后，必须按以下顺序执行串行数据发送 / 接收同时运行。



（1） 确认SCI的状态后，写入接收数据：
读取SCSSR，确认TDRE为1后，对SCTDR写入数据，并将TDRE标志清0。也可通过TXI中断得知TDRE标志从0变为1。

（2） 接收错误处理：
产生接收错误时，读取SCSSR的ORER标志，并执行既定的错误处理后，必须将ORER标志清0。ORER标志为1时，不可重新开始接收。

（3） 确认SCI的状态后读取接收数据：
读取SCSSR，确认RDRF标志为1后，读取SCRDR的接收数据，并将RDRF标志清0。也可通过RXI中断得知RDRF标志从0变为1。

（4） 连续串行发送/接收的步骤：
连续串行发送/接收时，接收当前帧的MSB（bit7）之前，必须读取RDRF标志及SCRDR，并将RDRF标志清0。另外，发送当前帧的MSB（bit7）之前，必须确认读取TDRE标志为1且为可写入状态，还必须对SCTDR写入数据，并将TDRE标志清0。但是，由发送数据空中断（TXI）请求启动数据传送控制器（DTC）对SCTDR写入数据时，自动执行TDRE位的校验及清除。另外，由接收数据满中断（RXI）请求启动数据传送控制器（DTC）读取SCRDR的值时，自动清除RDRF位。

【注】 从发送或接收运行转换到同时发送/接收时，必须在TE位及RE位清0后同时将TE位及RE位置1。

图 13.14　串行数据发送 / 接收的流程图例子

13.4.4　多处理器通信功能

使用多处理器通信功能时，可通过附加多处理器位的异步串行通信，在多个处理器之间共享通信线路，发送 / 接收数据。多处理器通信时，对接收站分配各自固有的 ID 码。串行通信周期由指定接收站的 ID 发送周期和指定接收站的数据发送周期构成。使用多处理器位区分 ID 发送周期与数据发送周期。多处理器位为 1 时，是 ID 发送周期；为 0 时，是数据发送周期。使用多处理器格式的处理器间通信例子如图 13.15 所示。发送站首先在接收站 ID 码后附加为 1 的多处理器位，作为通信数据发送，之后，在发送数据后附加为 0 的多处理器位，作为通信数据发送。接收站接收多处理器位为 1 的通信数据后，与本站的 ID 比较，如果匹配，则继续接收被发送的通信数据；如果不匹配，则跳读通信数据，直到再次接收多处理器位为 1 的通信数据。

SCI 支持此功能，因此在 SCSCR 设有 MPIE 位。如果 MPIE 置 1，则在接收多处理器位为 1 的数据前，禁止从 SCRSR 向 SCRDR 传送接收数据、检测接收错误及置位 SCSSR 的 RDRF、 FER、 OER 的各状态标志。如果接收多处理器位为 1 的接收字符，则 SCSSR 的 MPBR 置 1，同时，在自动清除 MPIE 后返回通常的接收运行。此时，如果 SCSCR 的 RIE 置位，则产生 RXI 中断。

指定多处理器格式时，奇偶校验位的指定无效。其他均与通常的异步模式相同。执行多处理器通信时的时钟也与通常的异步模式相同。

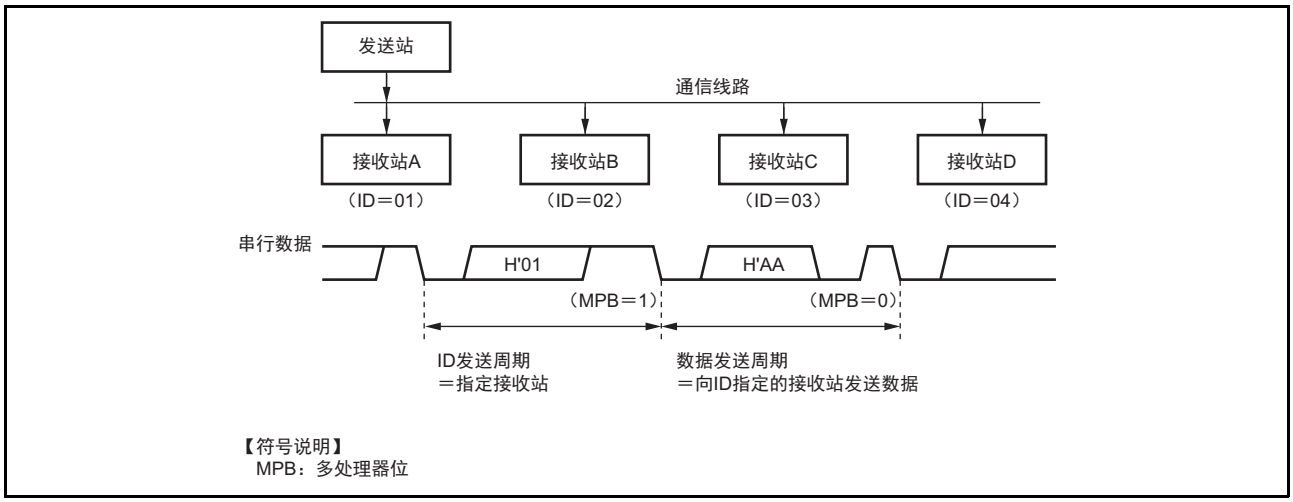


图 13.15　使用多处理器格式的通信例子（向接收站 A 发送数据 H'AA 的例子）

13.4.5　多处理器串行数据的发送

多处理器数据处理的流程图例子如图 13.16 所示。ID 发送周期时，必须将 SCSSR 的 MPBT 置 1 后发送；数据发送周期时，必须将 SCSSR 的 MPBT 清 0 后发送。其他运行与异步模式相同。

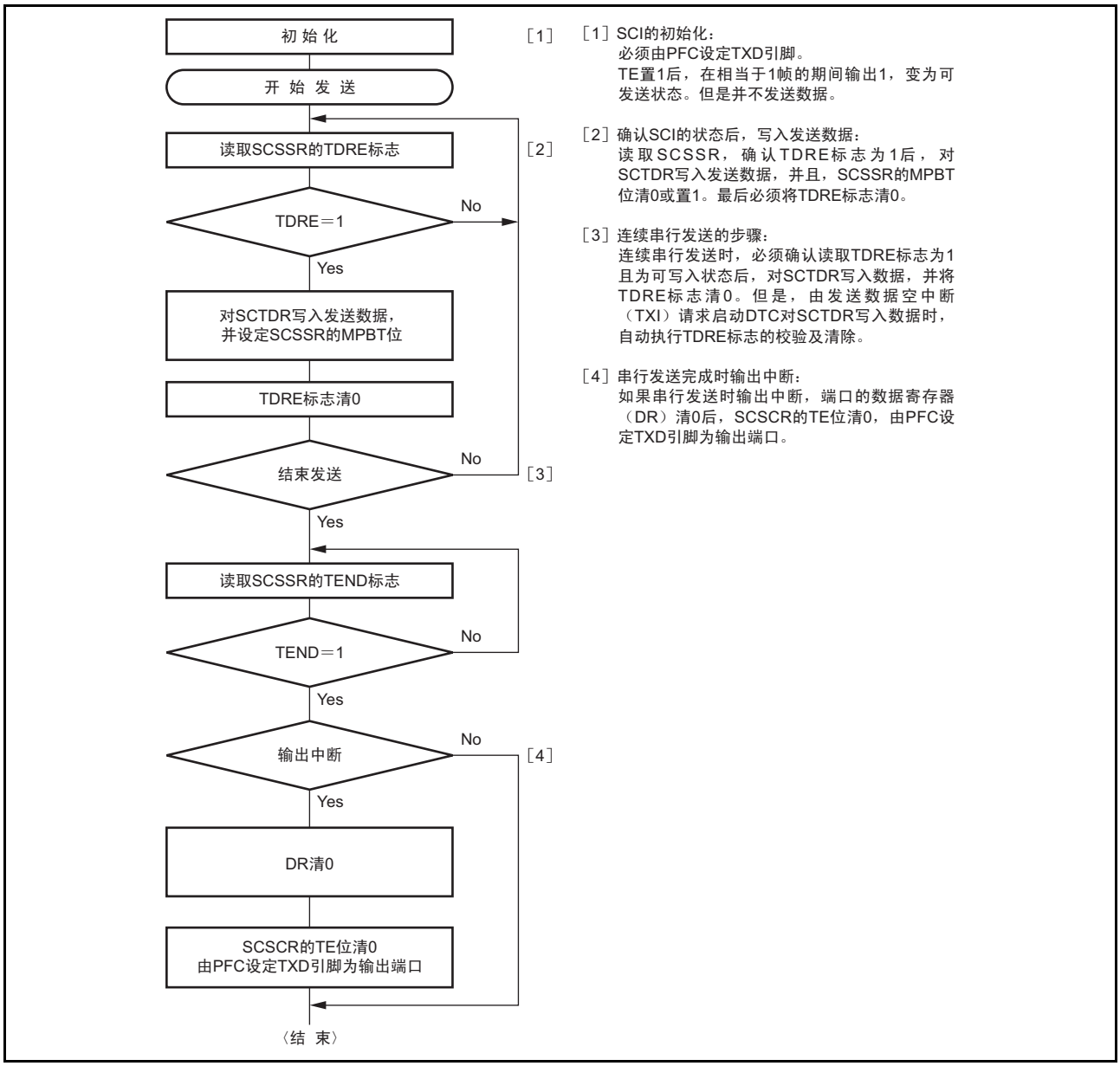


图 13.16　多处理器串行发送的流程图例子

13.4.6　多处理器串行数据的接收

多处理器数据接收的流程图例子如图 13.18 所示。如果 SCSCR 的 MPIE 置 1，则跳读通信数据，直至接收多处理器位为 1 的通信数据。接收多处理器位为 1 的通信数据后，向 SCRDR 传送接收数据。此时，产生 RXI 中断请求。其他运行与异步模式相同。接收时的运行例子如图 13.17 所示。

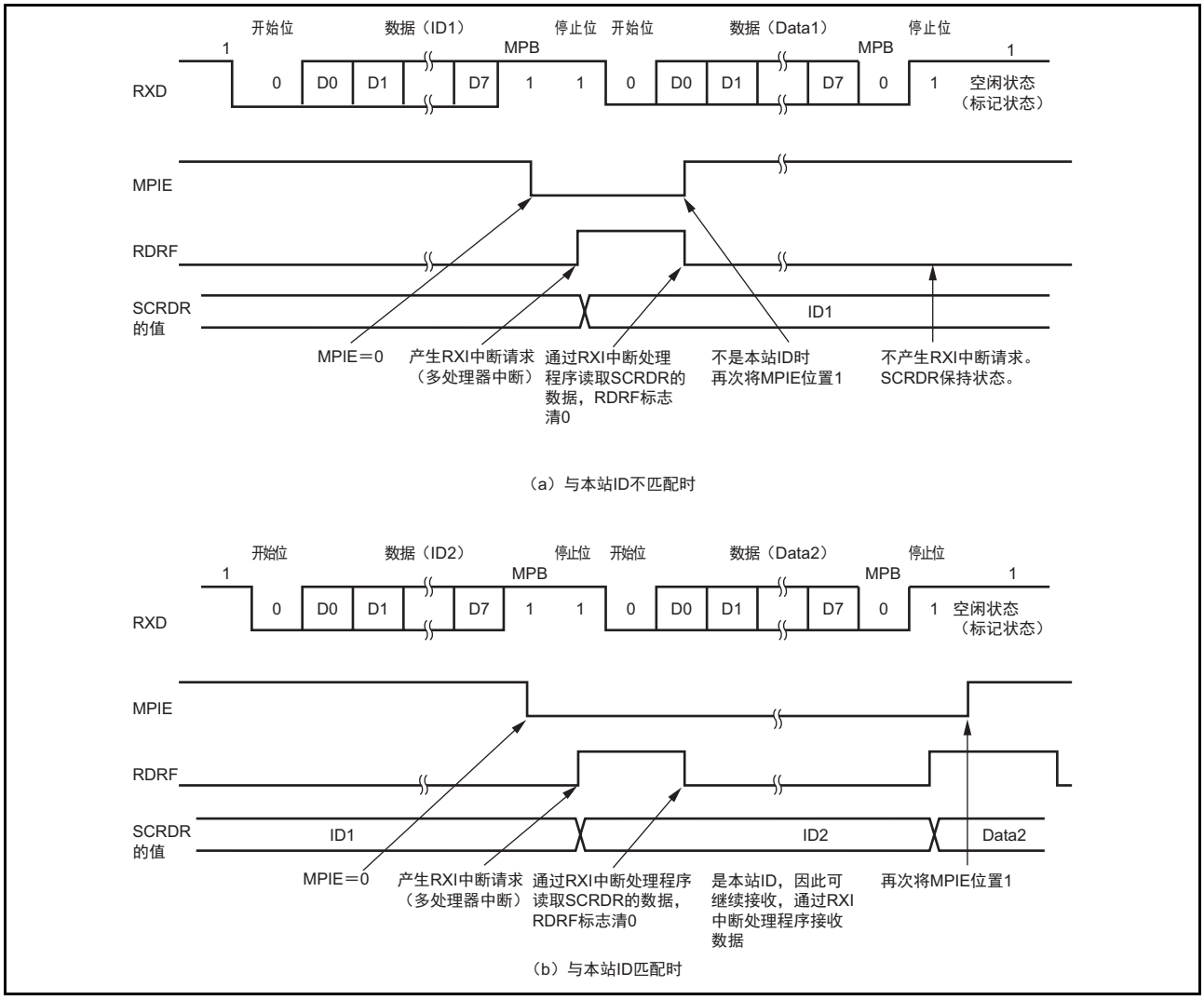


图 13.17　SCI 接收时的运行例子（8 位数据 / 有多处理器位 /1 位停止位的例子）

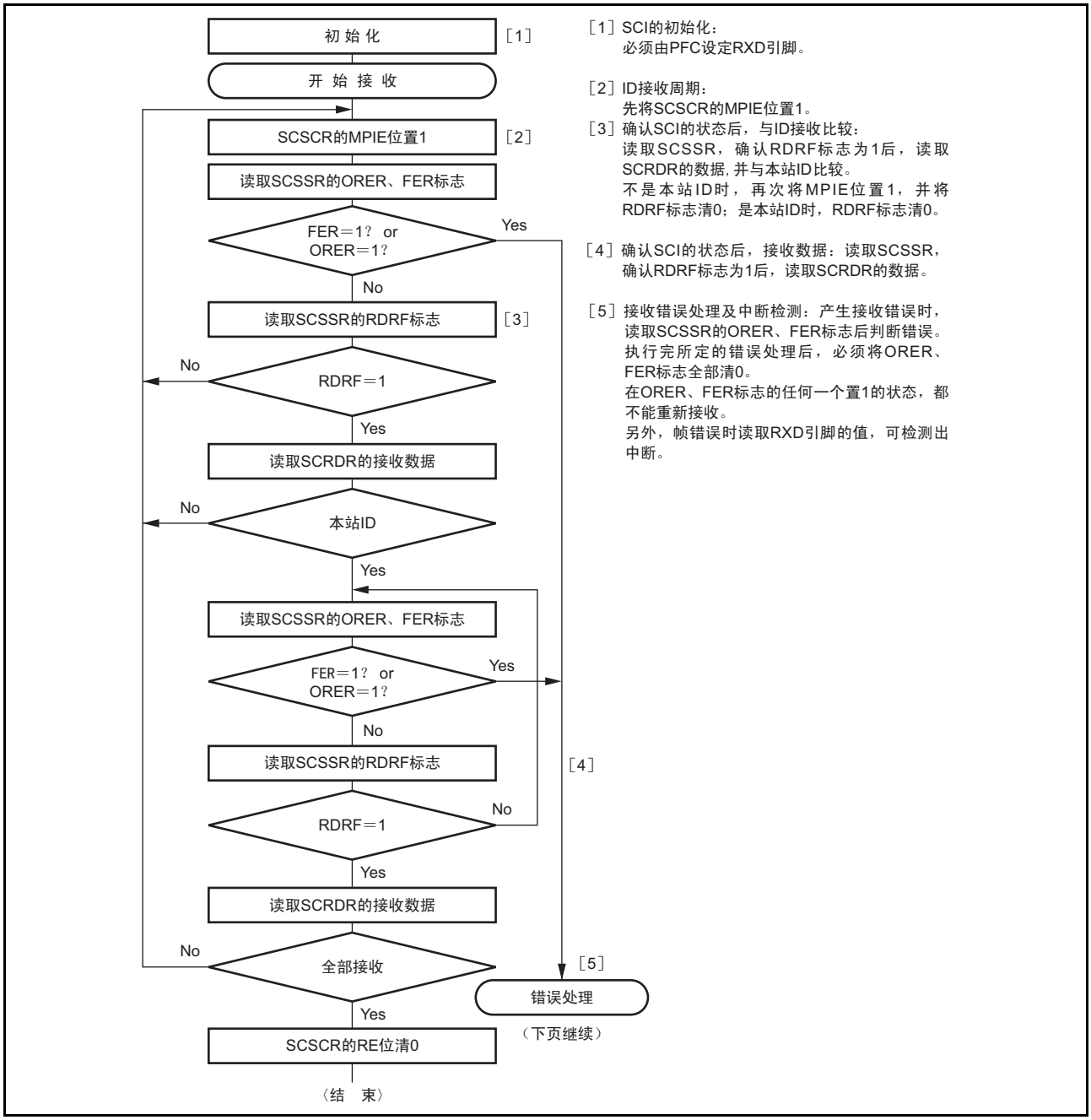


图 13.18　多处理器串行接收的流程图例子（1）

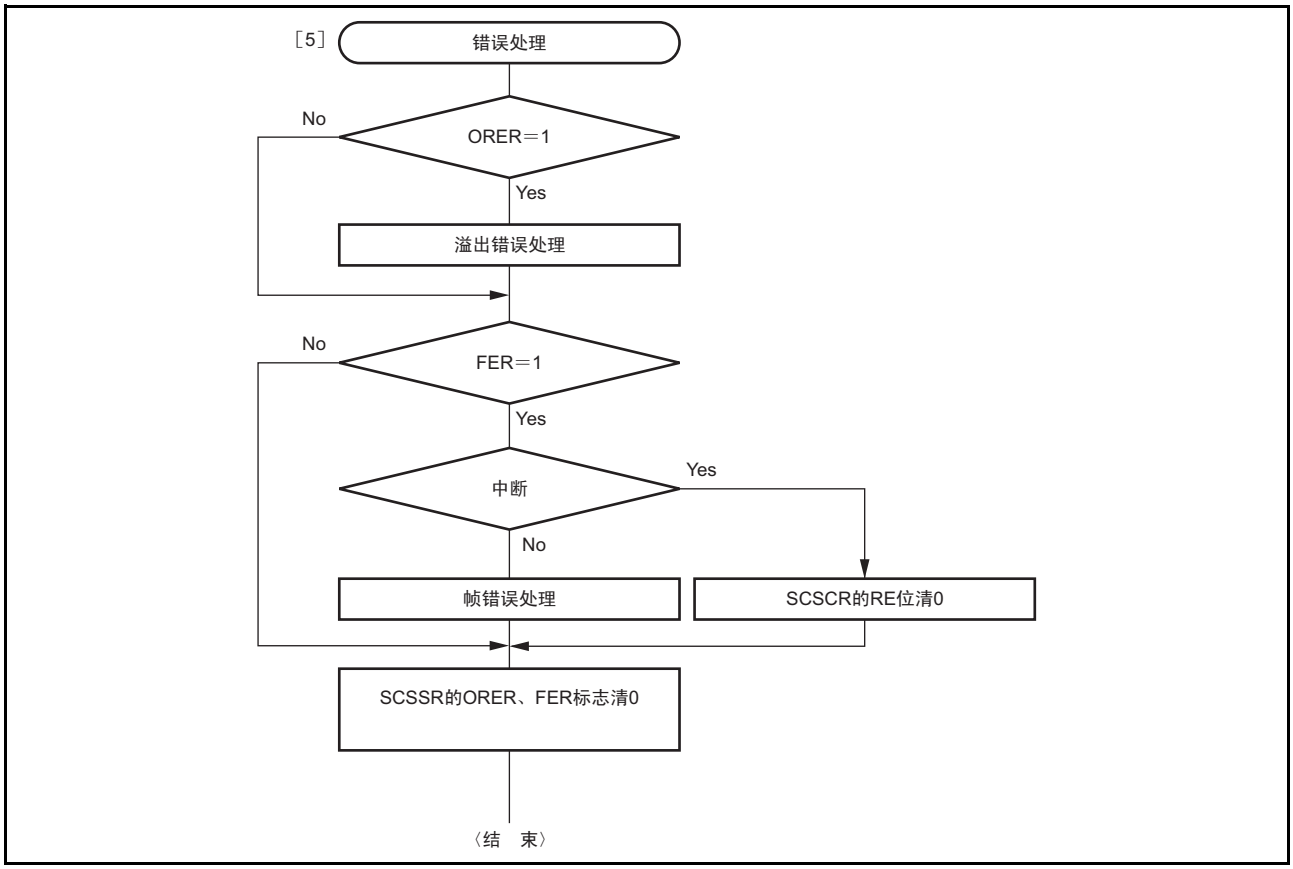


图 13.18　多处理器串行接收的流程图例子（2）

13.5　SCI 的中断源与 DTC

SCI 有发送结束中断 （TEI）请求、接收错误中断 （ERI）请求、接收数据满中断 （RXI）请求、发送数据空中断 （TXI）请求等 4 种中断源。

各中断源与优先顺序如表 13.17 所示。可通过 SCSCR 的 TIE、 RIE、 TEIE 位及 SCSPTR 的 EIO 位，允许或禁止各中断源。各中断请求可分别独立发送至中断控制器。

串行状态寄存器 （SCSSR）的 TDRE 标志置 1 时，产生 TDR 空请求。通过 TDR 空请求，可启动数据传送控制器 （DTC）执行数据传送。由 DTC 向发送数据寄存器 （SCTDR）写入时， TDRE 标志自动清 0。

SCSSR 的 RDRF 标志置 1 时，产生 RDR 满请求。可由 RDR 满请求启动 DTC 执行数据传送。

通过 DTC 读取接收数据寄存器 （SCRDR）时， RDRF 标志自动清 0。

另外， SCSSR 的 ORER 及 FER 标志或 PER 置 1 时，产生 ERI 中断请求。不可由此 ERI 中断请求启动 DTC。由 DTC 处理接收数据及通过向 CPU 请求中断执行接收错误处理时，必须将 RIE 位设定为 1，同时将 SCSPTR 的 EIO 位置 1，并设置为仅在接收错误时产生中断错误。将 EIO 位设定为 0 时，正常执行数据接收，也向 CPU 产生中断。

SCSSR 的 TEND 标志置 1 时，产生 TEI 中断请求。不可由此 TEI 中断请求启动 DTC。

此外， TXI 中断表示可写入发送数据； TEI 中断表示发送运行结束。

表 13.17　SCI 中断源

中断源	内　　容	启动 DTC
ERI	接收错误 （ORER、 FER、 PER）引起的中断	不可
RXI	接收数据满 （RDRF）引起的中断	可
TXI	发送数据空 （TDRE）引起的中断	可
TEI	发送结束 （TEND）引起的中断	不可

13.6　串行端口寄存器 （SCSPTR）与 SCI 引脚的关系

SCSPTR 与 SCI 引脚的关系如图 13.19、图 13.20 所示。

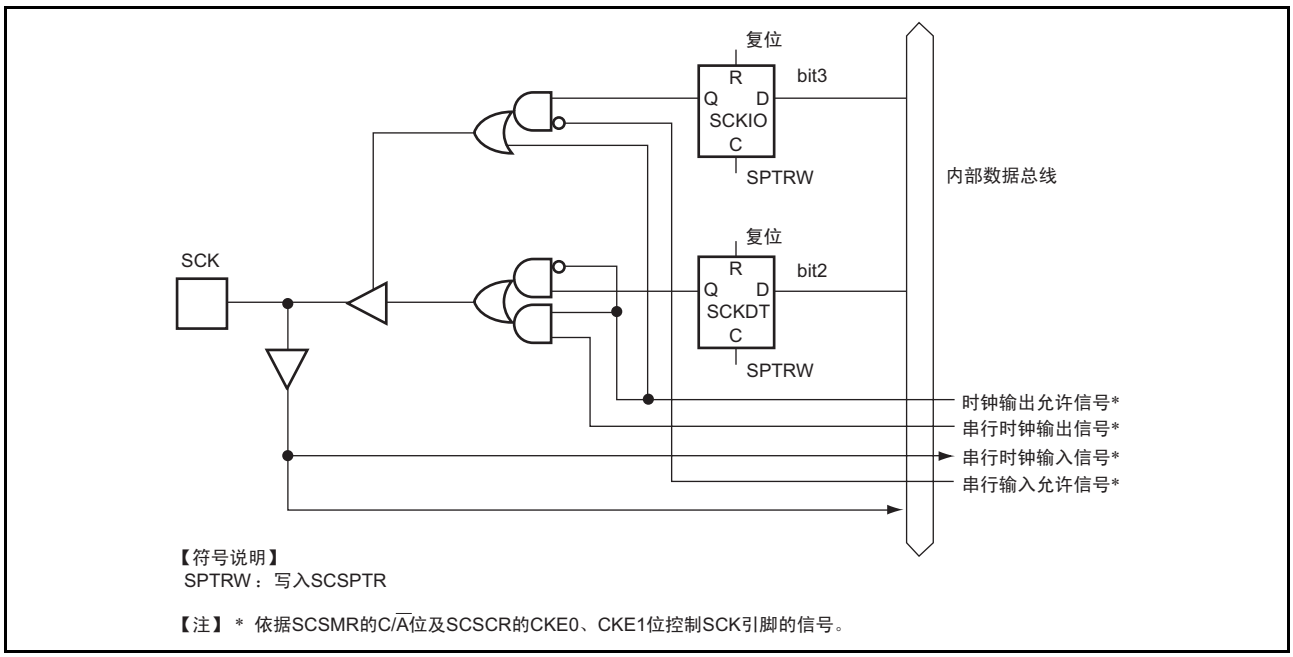


图 13.19　SCKIO 位、 SCKDT 位与 SCK 引脚的关系

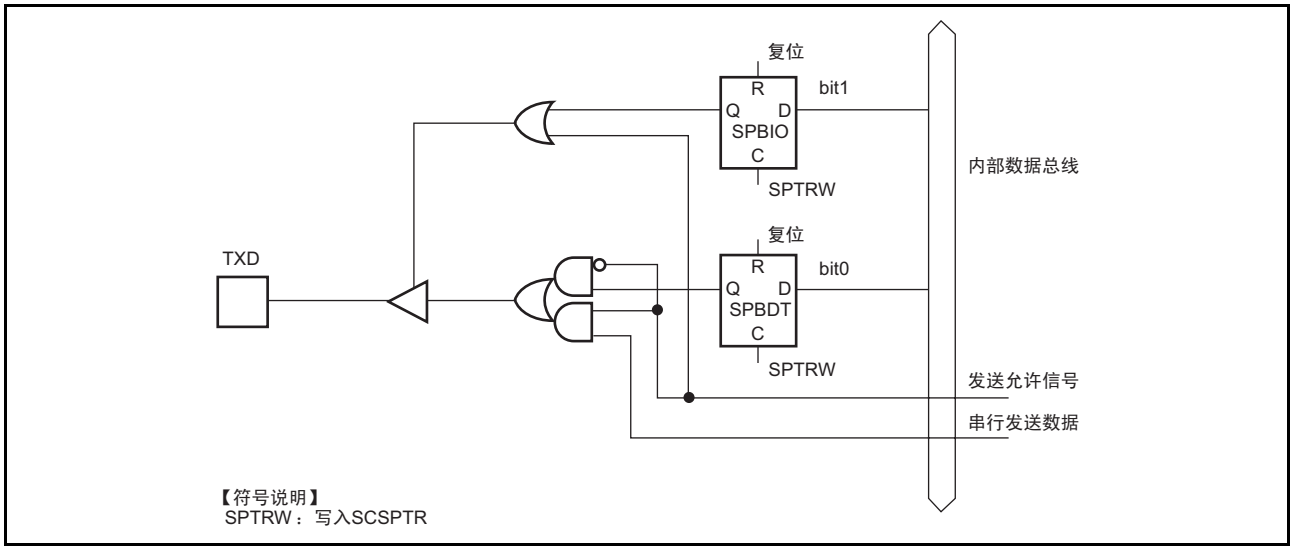


图 13.20　SPBIO 位、 SPBDT 位与 TXD 引脚的关系

13.7　使用时的注意事项

13.7.1　向 SCTDR 的写入与 TDRE 标志的关系

串行状态寄存器（SCSSR）的 TDRE 标志表示已从发送数据寄存器（SCTDR）向发送移位寄存器（SCTSR）传送发送数据。如果 SCI 从 SCTDR 向 SCTSR 传送数据，则 TDRE 标志置 1。

与 TDRE 标志的状态无关，可向 SCTDR 写入数据。如果在 TDRE 标志为 0 的状态，向 SCTDR 写入新数据，则在 SCTDR 保存的数据因尚未传送至 SCTSR 而丢失。因此，向 SCTDR 写入发送数据时，必须先确认 TDRE 标志置 1。

13.7.2　同时产生多个接收错误时的运行

同时产生多个接收错误时， SCSSR 各状态标志的状态如表 13.18 所示。产生溢出错误时，不从接收移位寄存器（SCRSR）向接收数据寄存器（SCRDR）传送数据，而接收数据则丢失。

表 13.18　SCSSR 状态标志的状态与接收数据的传送

接收错误的状态	SCSSR 的状态标志				传送接收数据 SCRSR→SCRDR
	RDRF	ORER	FER	PER	
溢出错误	1	1	0	0	×
帧错误	0	0	1	0	○
奇偶校验错误	0	0	0	1	○
溢出错误 + 帧错误	1	1	1	0	×
溢出错误 + 奇偶校验错误	1	1	0	1	×
帧错误 + 奇偶校验错误	0	0	1	1	○
溢出错误 + 帧错误 + 奇偶校验错误	1	1	1	1	×

【符号说明】

○：从 SCRSR 向 SCRDR 传送接收数据。

×：不从 SCRSR 向 SCRDR 传送接收数据。

13.7.3　中断检测与处理

检测帧错误 （FER）时直接读取 RXD 引脚的值，可检测中断。中断时，从 RXD 引脚的输入均为 0，因此，FER 标志置位，并且奇偶校验错误 （PER）有时也置位。

SCI 接收中断后，停止向 SCRDR 传送接收数据，但继续接收运行。

13.7.4　中断的发送

通过串行端口寄存器 （SCSPTR）的 SPB0IO、 SPB0DT 位，可确定 TXD 引脚的输入 / 输出条件及电平。使用该特点可发送中断。

从串行发送的初始化到 TE 位置 1 （可发送）前，不可用作 TXD 引脚。在此期间， SPB0DT 位的值代替标记状态。因此，开始时，预先将 SPB0IO 及 SPB0DT 位设定为 1 （输出、高电平）。

要在串行发送时发送中断，将 SPB0DT 位清 0 （低电平）后，将 TE 位清 0 （发送停止）。如果 TE 位清 0，则与当前的发送状态无关，初始化发送部，从 TXD 引脚输出 0。

13.7.5　异步模式的接收数据采样时序与接收容限

异步模式时， SCI 通过频率为 16 倍传送速率的基本时钟运行。

接收时， SCI 使用基本时钟采样开始位的下降沿，并进行内部同步。另外，在基本时钟的第 8 个上升沿向内部获取接收数据。

如图 13.21 所示。

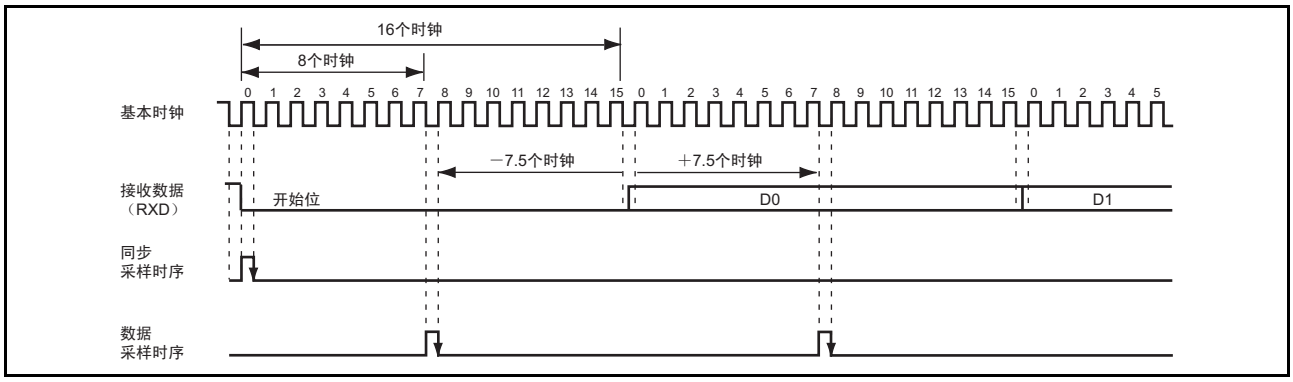


图 13.21　异步模式的接收数据采样时序

因此，异步模式时的接收容限公式 （1）如下所示：

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L-0.5)F - \frac{|D-0.5|}{N}(1+F) \right| \times 100[\%] \quad \text{…公式（1）}$$

M：接收容限 （%）

N：时钟的位速率比 （N=16）

D：时钟占空比 （D=0 ～ 1.0）

L：帧长度 （L=9 ～ 12）

F：时钟频率偏差的绝对值

公式 （1）中，假设 F=0、 D=0.5，根据公式 （2），得接收容限为 46.875%。

D =0.5、 F=0 时

M =(0.5 － 1/(2×16))×100%

=46.875%　　…公式 （2）

但是，此值仅为计算值，系统设计时，还必须保证 20 ～ 30%的余地。

13.7.6　使用 DTC 时的注意事项

1. 使用外部时钟源作为同步时钟时，由DTC更新SCTDR后，必须等待大于等于5个外围运行时钟周期的时间，然后再输入外部时钟。如果在SCTDR更新后的4个周期内输入发送时钟，则可能出现误动作（参照图13.22）。

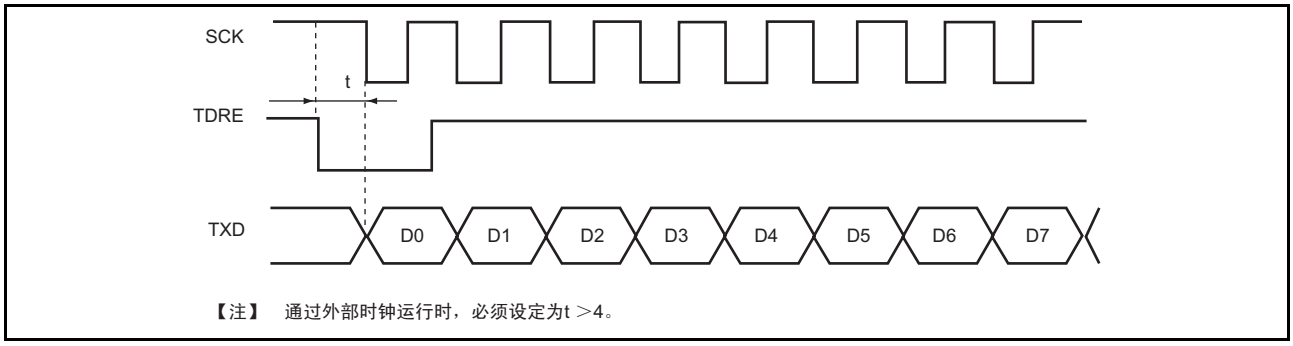


图 13.22　通过 DTC 执行同步时钟传送的例子

2. 由TXI中断启动DTC向SCTDR写入数据时， TEND标志不定，因此，不得将TEND标志用作发送结束标志。

13.7.7　时钟同步外部时钟模式时的注意事项

外部时钟 SCK 从 0 设置为 1 后，必须等待大于等于 4 个外围运行时钟周期的时间，然后再设定 TE=1、RE=1。必须在外部时钟 SCK 为 1 时，设定 TE=RE=1。

13.7.8　模块待机模式的设定

SCI 可通过待机控制寄存器设定禁止 / 允许本模块运行。为初始值时， SCI 运行停止。解除模块待机模式，可存取寄存器。详情参阅 “第 22 章 低功耗模式”。

第 14 章 同步串行通信单元 （SSU）

本 LSI 具有 1 个通道的同步串行通信单元 （SSU：Synchronous Serial communication Unit）。 SSU 有以本 LSI 为主器件向外部输出时钟并执行同步串行通信的主模式，及输入来自外部器件的时钟并执行同步串行通信的从属模式。可在时钟极性与时钟相位不同的器件之间执行同步串行通信。

14.1 特点

- 可选择SSU模式与时钟同步通信模式
- 可选择主模式与从属模式
- 可选择标准模式与双向模式
- 可在时钟相位和时钟极性不同的其他器件之间执行同步串行通信
- 可选择发送/接收数据长度为8位/16位/32位
- 可执行全双工通信
  配备可同时执行发送与接收的移位寄存器
- 可连续串行通信
- 可选择LSB first方式/MSB first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）与外部时钟作为时钟源
- 中断源：5种
  即：发送结束、发送数据空、接收数据满、溢出错误及竞争错误
  通过发送数据空请求、接收数据满请求启动数据传送控制器 （DTC），可传送数据。
- 可设定模块待机模式

SSU 框图如图 14.1 所示。

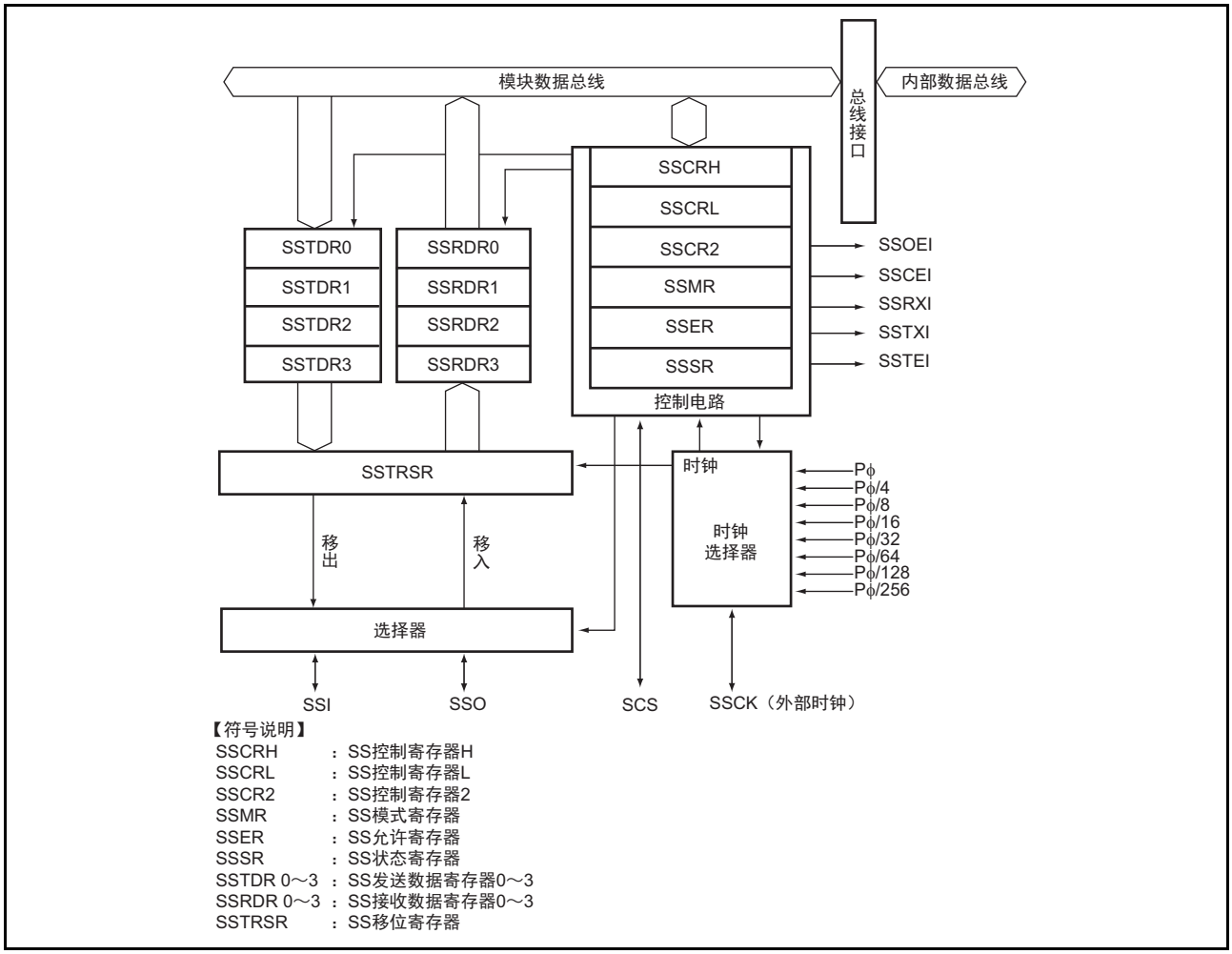


图 14.1　SSU 框图

14.2　输入 / 输出引脚

SSU 具有表 14.1 所示的输入 / 输出引脚。

表 14.1　引脚结构

引脚名称	输入 / 输出	功　能
SSCK	输入 / 输出	SSU 时钟输入 / 输出引脚
SSI	输入 / 输出	SSU 数据输入 / 输出引脚
SSO	输入 / 输出	SSU 数据输入 / 输出引脚
$\overline{\text{SCS}}$	输入 / 输出	SSU 片选输入 / 输出引脚

14.3　寄存器说明

SSU 有以下寄存器。有关寄存器的地址及各处理状态的寄存器状态，详情参阅 “第 24 章 寄存器一览表”。

表 14.2　寄存器结构

寄存器名称	简称	R/W	初始值	地址	存取长度
SS 控制寄存器 H	SSCRH	R/W	H'0D	H'FFFFCD00	8、16
SS 控制寄存器 L	SSCRL	R/W	H'00	H'FFFFCD01	8
SS 模式寄存器	SSMR	R/W	H'00	H'FFFFCD02	8、16
SS 允许寄存器	SSER	R/W	H'00	H'FFFFCD03	8
SS 状态寄存器	SSSR	R/W	H'04	H'FFFFCD04	8、16
SS 控制寄存器 2	SSCR2	R/W	H'00	H'FFFFCD05	8
SS 发送数据寄存器 0	SSTDR0	R/W	H'00	H'FFFFCD06	8、16
SS 发送数据寄存器 1	SSTDR1	R/W	H'00	H'FFFFCD07	8
SS 发送数据寄存器 2	SSTDR2	R/W	H'00	H'FFFFCD08	8、16
SS 发送数据寄存器 3	SSTDR3	R/W	H'00	H'FFFFCD09	8
SS 接收数据寄存器 0	SSRDR0	R	H'00	H'FFFFCD0A	8、16
SS 接收数据寄存器 1	SSRDR1	R	H'00	H'FFFFCD0B	8
SS 接收数据寄存器 2	SSRDR2	R	H'00	H'FFFFCD0C	8、16
SS 接收数据寄存器 3	SSRDR3	R	H'00	H'FFFFCD0D	8

14.3.1　SS 控制寄存器 H （SSCRH）

SSCRH 可设定主 / 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	位名称	初始值	R/W	说　明
7	MSS	0	R/W	选择主 / 从属器件 选择 SSU 用作主模式或从属模式。用作主模式时，从 SSCK 引脚输出传送时钟。 SSSR 的 CE 位置位时，此位自动清除。 0：选择从属模式 1：选择主模式
6	BIDE	0	R/W	双向模式允许 选择使用串行数据输入及输出 2 个引脚或其中 1 个引脚。但选择双向模式时，不可同时发送 / 接收。详情参阅 “14.4.3 数据输入 / 输出引脚和移位寄存器的关系”。 0：标准模式 （使用数据输入及输出 2 个引脚执行通信） 1：双向模式 （仅使用数据输入或输出的 1 个引脚执行通信）
5	—	0	R	保留位 读取值、写入值总是为 0。
4	SOL	0	R/W	选择串行数据输出值 发送结束后的串行数据输出保持发送数据最末位的值，但发送前或发送后可更改串行数据的输出电平。更改输出电平时，必须由 MOV 指令将 SOLP 位清 0。数据传送过程中写入此位会产生误动作，因此发送过程中不可写入此位。 0：串行数据的输出电平更改为低电平 1：串行数据的输出电平更改为高电平
3	SOLP	1	R/W	SOL 位写入保护 更改串行数据的输出电平时，必须由 MOV 指令设定 SOL=1 且 SOLP=0，或 SOL=0 且 SOLP=0。 0：可根据 SOL 的值更改输出电平 1：不可根据 SOL 的值更改输出电平 读取值总为 1。
2	—	1	R	保留位 读取值、写入值总是为 1。
1、 0	CSS[1:0]	01	R/W	$\overline{\text{SCS}}$ 引脚选择 选择将 $\overline{\text{SCS}}$ 引脚用作 $\overline{\text{SCS}}$ 输入或 $\overline{\text{SCS}}$ 输出。 00：禁止设定 01：$\overline{\text{SCS}}$ 输入功能 10：$\overline{\text{SCS}}$ 自动输入 / 输出功能 （传送前、传送后为 $\overline{\text{SCS}}$ 输入，传送过程中为低电平输出） 11：$\overline{\text{SCS}}$ 自动输出功能 （传送前、传送后为高电平输出，传送过程中为低电平输出）

14.3.2　SS 控制寄存器 L（SSCRL）

SSCRL 可选择运行模式、软件复位及发送 / 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	位名称	初始值	R/W	说　明
7	FCLRM	0	R/W	标志清除模式 选择写入 SSTDR 时或读取 SSRDR 和 DTC 传送结束时清除 SSRXI、SSTXI 中断标志。使用 DTC 时，本位必须设定为 0。 0：DTC 传送结束时 1：寄存器存取时
6	SSUMS	0	R/W	选择 SSU 模式与时钟同步通信模式。 0：SSU 模式 1：时钟同步通信模式
5	SRES	0	R/W	软件复位 如果本位置 1，则强行复位 SSU 内部定序器。此后，自动清除本位，并初始化 SSSR 的 ORER、 TEND、 TDRE、 RDRF、 CE 各位及 SSER 的 TE、 RE 位，且保持其他 SSU 内部寄存器的值。 欲在中途中断传送，必须对本位写入 1，并复位内部定序器。
4 ～ 2	—	均为 0	R	保留位 读取值、写入值总是为 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

14.3.3　SS 模式寄存器 （SSMR）

SSMR 可选择 MSB first/LSB first、时钟极性、时钟相位及传送时钟速率。

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	位名称	初始值	R/W	说　明
7	MLS	0	R/W	选择 MSB first/LSB first 选择以 MSB first 或 LSB first 传送串行数据。 0：LSB first 1：MSB first
6	CPOS	0	R/W	选择时钟极性 选择 SSCK 时钟的极性。 0：空闲时为高电平输出，有效时为低电平输出 1：空闲时为低电平输出，有效时为高电平输出
5	CPHS	0	R/W	选择时钟相位 （仅限 SSU 模式时有效） 选择 SSCK 时钟的相位。 0：在第一个边沿数据变化 1：在第一个边沿锁存数据
4、3	—	均为 0	R	保留位 读取值、写入值总是为 0。
2 ～ 0	CKS[2:0]	000	R/W	选择传送时钟速率 选择内部时钟时的传送时钟速率 （预分频器分频比）。 000：保留 001：Pϕ/4 010：Pϕ/8 011：Pϕ/16 100：Pϕ/32 101：Pϕ/64 110：Pϕ/128 111：Pϕ/256

14.3.4　SS 允许寄存器（SSER）

SSER 可设定发送允许、接收允许及中断请求允许。

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	位名称	初始值	R/W	说　明
7	TE	0	R/W	发送允许 如果此位置 1，则可发送。
6	RE	0	R/W	接收允许 如果此位置 1，则可接收。
5、4	—	均为 0	R	保留位 读取值、写入值总是为 0。
3	TEIE	0	R/W	发送结束中断允许 如果此位置 1，则允许 SSTEI 中断请求。
2	TIE	0	R/W	发送中断允许 如果此位置 1，则允许 SSTXI 中断请求。
1	RIE	0	R/W	接收中断允许 如果此位置 1，则允许 SSRXI 中断及 SSOEI 中断请求。
0	CEIE	0	R/W	竞争错误中断允许 如果此位置 1，则允许 SSCEI 中断请求。

14.3.5　SS 状态寄存器（SSSR）

SSSR 为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	位名称	初始值	R/W	说　明
7	—	0	R	保留位 读取值、写入值总是为 0。
6	ORER	0	R/W	溢出错误 表示在 RDRF=1 的状态，如果接收下一个数据，则产生溢出错误，并异常结束。 SSRDR 保持产生溢出错误前的 1 帧的接收数据，之后接收到的数据丢失。且在 ORER=1 的状态不可继续此后的串行接收及发送。 ［置位条件］ • 在 RDRF=1 的状态，完成下一个串行接收的 1 字节时 ［清除条件］ • 读取 1 的状态后，写入 0 时
5、4	—	均为 0	R	保留位 读取值、写入值总是为 0。
3	TEND	0	R/W	发送结束 ［置位条件］ • SSCR2 的 TENDSTS 为 0 时，在 TDRE=1 的状态，发送数据的最末位时 • SSCR2 的 TENDSTS 为 1 时，在 TDRE=1 的状态，发送数据的最末位后 ［清除条件］ • 读取 TEND=1 状态后，对 TEND 标志写入 0 时 • 对 SSTDR 写入数据时

位	位名称	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 为 0 时 • 从 SSTDR 向 SSTRSR 传送数据，并可对 SSTDR 写入数据时 ［清除条件］ • 读取 TDRE=1 的状态后，对 TDRE 标志写入 0 时 • 在 TE=1 的状态，对 SSTDR 写入数据时 • 由 SSTXI 中断启动 DTC，在 DTC 的 MRB 的 DISEL 位为 0 时对 SSTDR 写入传送数据的情况
1	RDRF	0	R/W	接收数据寄存器满 表示 SSRDR 内有无数据。 ［置位条件］ • 串行接收正常结束，并从 SSTRSR 向 SSRDR 传送接收数据时 ［清除条件］ • 读取 RDRF=1 状态后，对 RDRF 标志写入 0 时 • 从 SSRDR 读取接收数据时 • 由 SSRXI 中断启动 DTC，在 DTC 的 MRB 的 DISEL 位为 0 时对 SSRDR 读取接收数据的情况
0	CE	0	R/W	竞争错误 / 未完成错误 表示在 SSUMS=0 （SSU 模式）、 MSS=1 （主器件）的状态，由外部向 $\overline{\text{SCS}}$ 输入 0 时，产生竞争错误。在 SSUMS=0 （SSU 模式）、MSS=0 （从属器件）的状态， $\overline{\text{SCS}}$ 引脚为 1 时，判断主器件传送停止，并产生未完成错误。在设定 CE=1 的状态，不可继续此后的串行接收及发送。开始重新传送前，必须将 SSCRL 的 SRES 置 1，并复位内部定序器。 ［置位条件］ • 在主器件 （SSCRH 的 MSS=1）的状态下向 $\overline{\text{SCS}}$ 引脚输入低电平时 • 在从属器件 （SSCRH 的 MSS=0）的状态下在传送过程中 $\overline{\text{SCS}}$ 引脚为 1 时 ［清除条件］ • 读取 1 的状态后，写入 0 时

14.3.6　SS 控制寄存器 2（SSCR2）

SSCR2 可设定 $\overline{\text{SCS}}$ 引脚的有效时序、SSO 引脚的数据输出时序及 TEND 位的置位时序。

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	位名称	初始值	R/W	说　明
7 ～ 5	—	均为 0	R	保留位 读取值、写入值总是为 0。
4	TENDSTS	0	R/W	选择 TEND 位的置位时序（仅限设定 SSU 模式及主模式时有效） 0：发送最末位的过程中置位 TEND 位 1：发送最末位后置位 TEND 位
3	SCSATS	0	R/W	选择 $\overline{\text{SCS}}$ 引脚的有效时序（仅限设定 SSU 模式及主模式时有效） 0：t_{LEAD}、t_{LAG} 输出期间的 Min. 为 $1/2\times t_{SUcyc}$ 1：t_{LEAD}、t_{LAG} 输出期间的 Min. 为 $3/2\times t_{SUcyc}$
2	SSODTS	0	R/W	选择 SSO 引脚的数据输出时序（仅限设定 SSU 模式及主模式时有效） 0：BIDE=0、MSS=1、TE=1，或 BIDE=1、TE=1、RE=0 时 SSO 引脚输出数据 1：BIDE=0、MSS=1、TE=1，或 BIDE=1、TE=1、RE=0 时，且 $\overline{\text{SCS}}$ 引脚为低电平期间，SSO 引脚输出数据
1、0	—	均为 0	R	保留位 读取值、写入值总是为 0。

14.3.7　SS 发送数据寄存器 0 ～ 3 （SSTDR0 ～ SSTDR3）

SSTDR 为保存发送数据的 8 位寄存器。根据 SSCRL 的 DATS1、 DATS0 位的设定，选择 8 位数据长度时 SSTDR0 有效；选择 16 位数据长度时 SSTDR0、 SSTDR1 有效；选择 32 位数据长度时 SSTDR0、 SSTDR1、 SSTDR2、 SSTDR3 有效。不得存取无效的 SSTDR。

如果检测出 SSTRSR 为空，则 SSU 将写入 SSTDR 的发送数据传送至 SSTRSR，并开始串行发送。SSTRSR 串行数据发送过程中，如果预先将下一个数据写入 SSTDR，则可连续串行发送。

SSTDR 总是可由 CPU 及 DTC 读取 / 写入，但为了确实执行串行通信，必须在确认 SSSR 的 TDRE 置 1 后写入 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	位名称	初始值	R/W	说　明
7 ～ 0		均为 0	R/W	串行发送数据

表 14.3　SSCRL 的 DATS 位的设定与 SSTDR 的对应关系

	DATS[1:0] 的设定			
	00	01	10	11 （设定无效）
SSTDR0	有效	有效	有效	无效
SSTDR1	无效	有效	有效	无效
SSTDR2	无效	无效	有效	无效
SSTDR3	无效	无效	有效	无效

14.3.8　SS 接收数据寄存器 0 ～ 3 （SSRDR0 ～ SSRDR3）

SSRDR 为保存接收数据的 8 位寄存器。根据 SSCRL 的 DATS1、 DATS0 位的设定，选择 8 位数据长度时 SSRDR0 有效；选择 16 位数据长度时 SSRDR0、 SSRDR1 有效；选择 32 位数据长度时 SSRDR0、 SSRDR1、 SSRDR2、 SSRDR3 有效。不得存取无效的 SSRDR。

如果接收完成 1 字节的数据，则 SSU 从 SSTRSR 向 SSRDR 传送接收的串行数据，并保存。此后，SSTRSR 为可接收状态。 SSTRSR 与 SSRDR 为双缓冲结构，因此可连续接收。

必须在确认 SSSR 寄存器的 RDRF 位置 1 后读取 SSRDR。

SSRDR 为只读寄存器，不可由 CPU 写入。

位：	7	6	5	4	3	2	1	0
初始值：	0	0	0	0	0	0	0	0
R/W：	R	R	R	R	R	R	R	R

位	位名称	初始值	R/W	说　明
7 ～ 0		均为 0	R	串行接收数据

表 14.4　SSCRL 的 DATS 位的设定与 SSRDR 的对应关系

	DATS[1:0] 的设定			
	00	01	10	11 （设定无效）
SSRDR0	有效	有效	有效	无效
SSRDR1	无效	有效	有效	无效
SSRDR2	无效	无效	有效	无效
SSRDR3	无效	无效	有效	无效

14.3.9　SS 移位寄存器 （SSTRSR）

SSTRSR 为发送 / 接收串行数据的移位寄存器。

从 SSTDR 向 SSTRSR 传送发送数据时，如果 SSMR 的 MLS=0，则将 SSTDR 的 bit0 传送至 SSTRSR 的 bit0 （LSB first 通信）；如果 MLS=1，则将 SSTDR 的 bit7 传送至 SSTRSR 的 bit7 （MSB first 通信）。此后，通过从 SSTRSR 的 LSB （bit0）开始按顺序将数据传送至 SSO 引脚，发送串行数据。

接收时，从 LSB （bit0）开始按接收顺序给 SSTRSR 设定从 SSI 引脚输入的串行数据。如果接受完成 1 字节的数据，则自动将 SSTRSR 的数据传送至 SSRDR。 SSTRSR 不可由 CPU 直接存取。

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

14.4　运行说明

14.4.1　传送时钟

可从 7 种内部时钟与外部时钟选择传送时钟。首先，使用本模块时，需预先通过 PFC 设置 SSCK 引脚有效。 SSCRH 的 MSS=1 时，选择内部时钟，且 SSCK 引脚为输出引脚。如果开始传送，则从 SSCK 引脚输出由 SSMR 的 CKS2 ～ CKS0 设定的传送速率的时钟。 MSS=0 时，选择外部时钟，且 SSCK 引脚为输入引脚。

14.4.2　时钟的相位、极性和数据的关系

SSCRL 的 SSUMS=0 时，因组合 SSMR 的 CPOS 与 CPHS，改变时钟的相位、极性及传送数据的关系，如图 14.2 所示。 SSUMS=1 时， CPOS 的设定有效，但 CPHS 的设定无效。

通过设定 SSMR 的 MLS，可选择以 MSB first 或 LSB first 传送。 MLS=0 时，按照从 LSB 到 MSB 的顺序传送； MLS=1 时，按照从 MSB 到 LSB 的顺序传送。

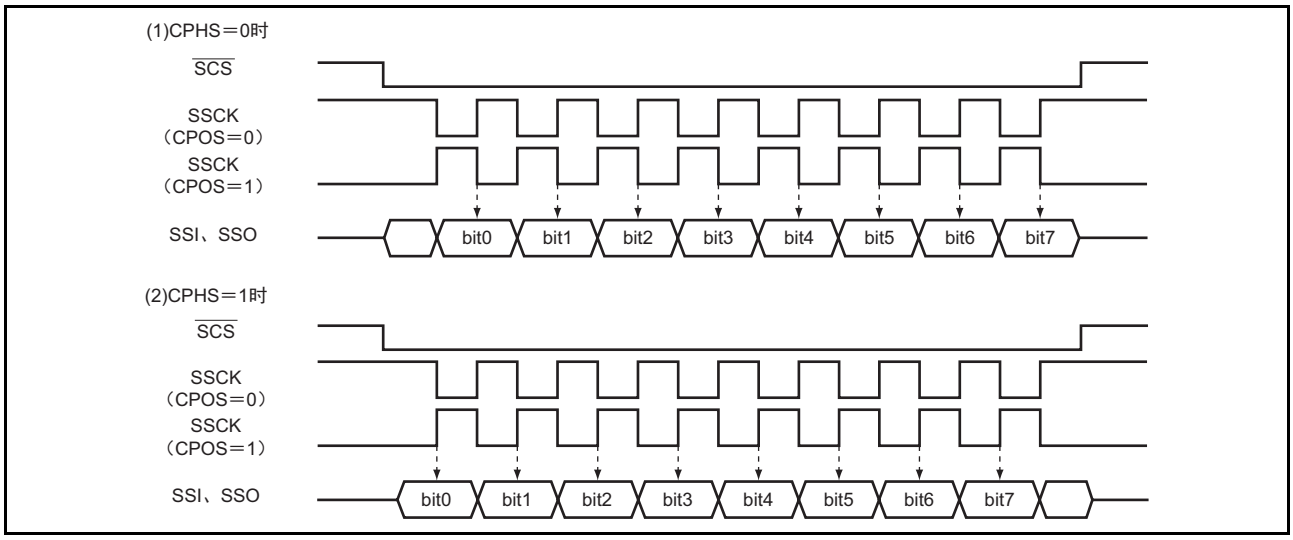


图 14.2　时钟的相位、极性和数据的关系

14.4.3　数据输入 / 输出引脚和移位寄存器的关系

因组合 SSCRH 的 MSS、 BIDE 与 SSCRL 的 SSUMS，改变数据输入 / 输出引脚与 SS 移位寄存器（SSTRSR）的连接关系，如图 14.3 所示。

SSU 在 BIDE=0、 MSS=1 （标准、主模式）的状态运行时，从 SSO 引脚发送串行数据，并从 SSI 引脚接收串行数据 （图 14.3 （1））；在 BIDE=0、 MSS=0 （标准、从属模式）的状态运行时，从 SSI 引脚发送串行数据，并从 SSO 引脚接收串行数据 （图 14.3 （2））。

在 BIDE=1 （双向模式）的状态，与主模式及从属模式无关，从 SSO 引脚发送或接收串行数据 （图 14.3 （3）、图 14.3 （4））。

但不可同时执行 TE 与 RE 同时置 1 时的发送 / 接收，必须选择 TE 或 RE 中的 1 个。

在 SSUMS=1 的状态运行时，从 SSO 引脚发送串行数据，从 SSI 引脚接收串行数据。 MSS=1 时，从 SSCK 引脚输出内部时钟； MSS=0 时， SSCK 引脚为输入引脚 （图 14.3 （5）、图 14.3 （6））。

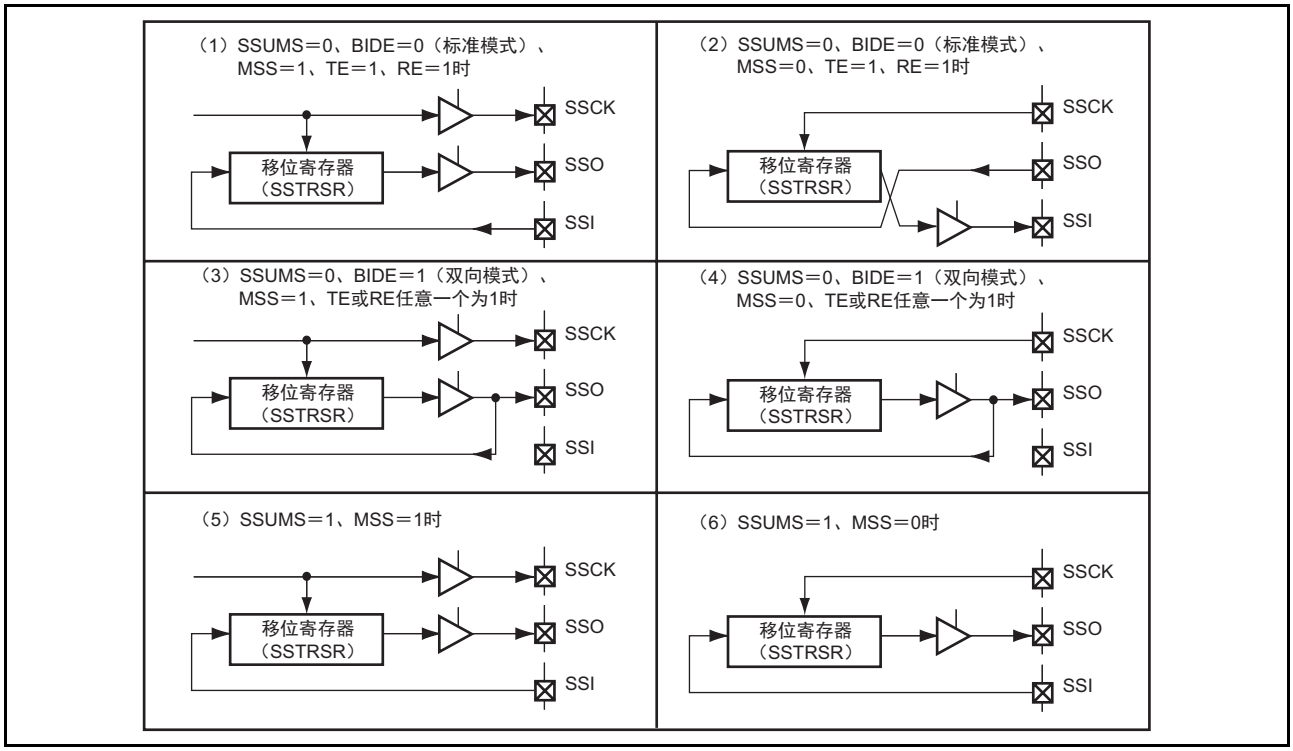


图 14.3　数据输入 / 输出引脚和移位寄存器的关系

14.4.4　各通信模式与引脚功能

SSU 通过设定各通信模式与寄存器转换输入 / 输出引脚（SSI、SSO、SSCK、$\overline{\text{SCS}}$）的功能，必须在端口 I/O 寄存器设定引脚输入 / 输出的转换。各通信模式与输入 / 输出引脚的关系如表 14.5 ～表 14.7 所示。

表 14.5　各通信模式与 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

表 14.6　各通信模式与 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

表 14.7　各通信模式与 $\overline{\text{SCS}}$ 引脚的状态

通信模式	寄存器状态				引脚状态
	SSUMS	MSS	CSS1	CSS0	$\overline{\text{SCS}}$
SSU 通信模式	0	0	x	x	输入
		1	0	0	—
			0	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

14.4.5　SSU 模式

在 SSU 模式使用时钟线（SSCK）、数据输入线（SSI 或 SSO）、数据输出线（SSI 或 SSO）及片选（$\overline{\text{SCS}}$）等 4 条总线执行数据通信。

也支持通过 1 个引脚执行数据输入线与数据输出线的双向模式。

(1)　SSU 模式的初始设定

SSU 模式的初始设定例子如图 14.4 所示。发送 / 接收数据前，必须在 SSER 的 TE 及 RE 清 0 后执行初始设定。

【注】 更改运行模式及通信格式时，必须将 TE 及 RE 清 0，还须注意：如果 TE 清 0，则 TDRE 置 1，但即使 RE 清 0，仍保持 RDRF、 ORER 的各标志及 SSRDR 的内容。

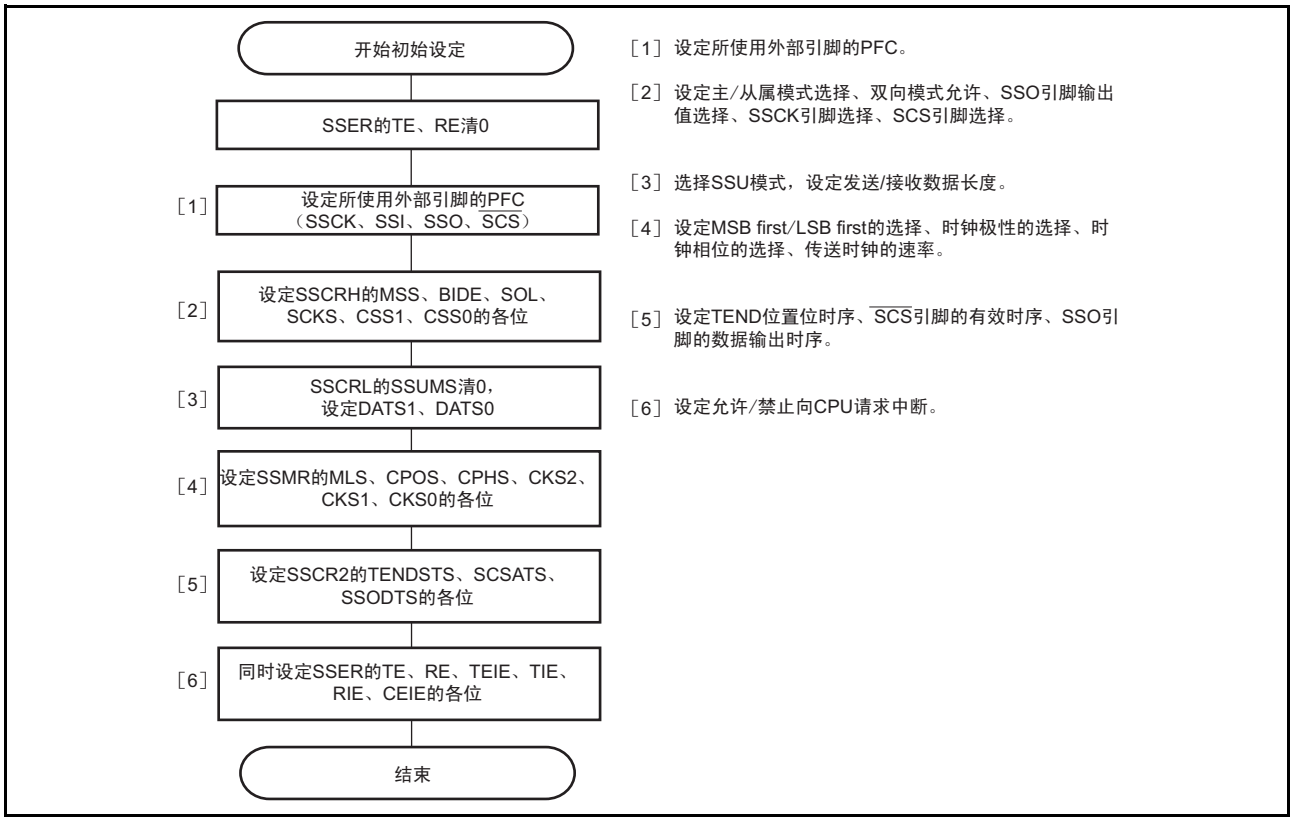


图 14.4　SSU 模式的初始设定例子

(2)　发送数据

发送时的运行例子如图 14.5 所示，发送数据的流程图例子如图 14.6 所示。

发送数据时 SSU 如下运行：

如果将 SSU 设定为主器件，则输出传送时钟及数据；如果设定为从属器件，则对 $\overline{\text{SCS}}$ 引脚输入低电平，从 SSCK 引脚输入传送时钟时，与该传送时钟同步输出数据。

SSU 在 SSER 的 TE 置 1 后，如果对 SSTDR 写入发送数据，则 SSSR 的 TDRE 自动清 0，并从 SSTDR 向 SSTRSR 传送数据，之后， TDRE 置 1，并开始发送。此时，如果 SSER 的 TIE 置 1，则产生 TXI 中断请求。

在 TDRE=0 的状态，如果结束 1 帧的数据传送，则从 SSTDR 向 SSTRSR 传送数据，并开始发送下一帧。在 TDRE=1 的状态，如果发送第 8 位，则 SSSR 的 TEND 置 1，并保持状态。此时，如果 SSER 的 TEIE 置 1，则产生 TEI 中断。发送结束后， SSCK 引脚在 SSMR 的 CPOS=0 时固定为高电平， CPOS=1 时固定为低电平。

在 SSSR 的 ORER 置 1 的状态不可发送，发送前必须确认 ORER 是否清 0。

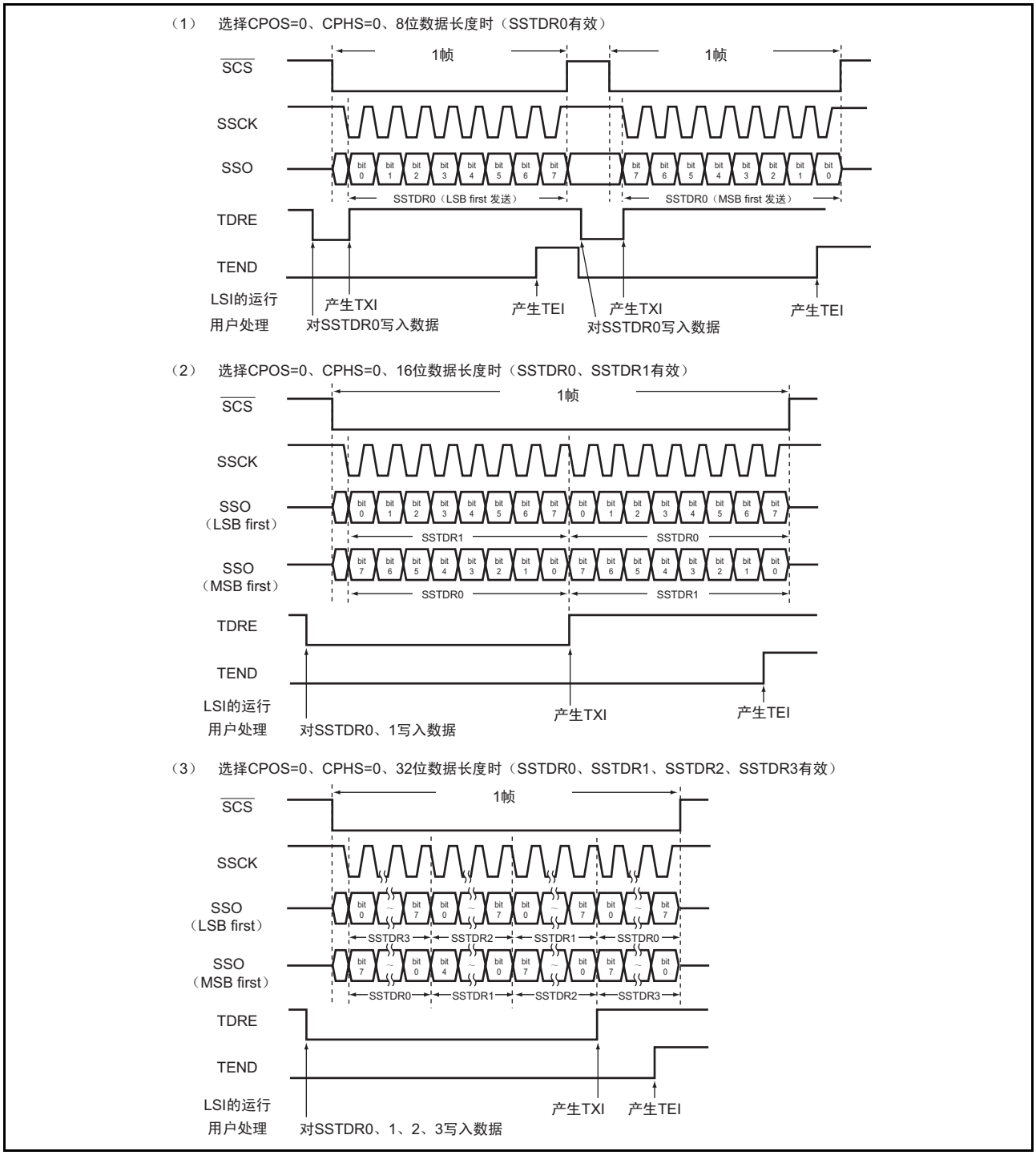


图 14.5　发送时的运行例子 （SSU 模式）

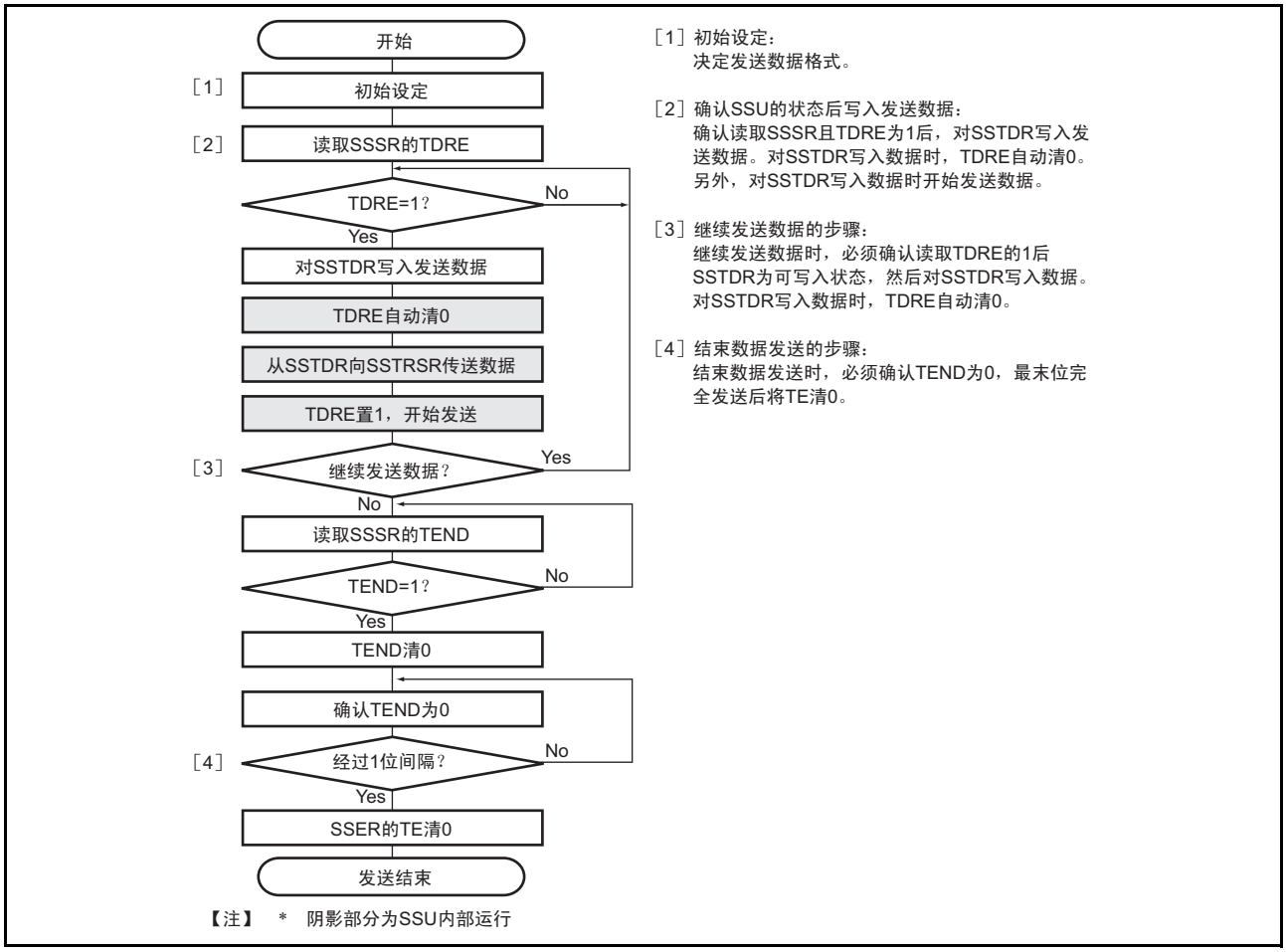


图 14.6　发送数据的流程图例子 （SSU 模式）

(3)　接收数据

接收时的运行例子如图 14.7 所示，接收数据的流程图例子如图 14.8 所示。接收数据时， SSU 如下运行：

通过将 SSER 的 RE 置 1，并虚读 SSRDR， SSU 开始接收。

如果将 SSU 设定为主器件，则输出传送时钟，并输入接收数据；如果设定为从属器件，则对 $\overline{\text{SCS}}$ 引脚输入低电平，从 SSCK 引脚输入传送时钟时，与该传送时钟同步输入接收数据。

接收 1 帧的数据后， SSSR 的 RDRF 置 1，并在 SSRDR 保存接收数据。此时，如果 SSER 的 RIE 置 1，则产生 RXI 中断请求。如果读取 SSRDR，则 RDRF 自动清 0。

在 RDRF=1 的状态如果第 8 个时钟上升，则 SSSR 的 ORER 置 1，产生溢出错误 （OEI），并停止接收。在 ORER=1 的状态不可接收，因此重新开始接收时 ORER 必须清 0。

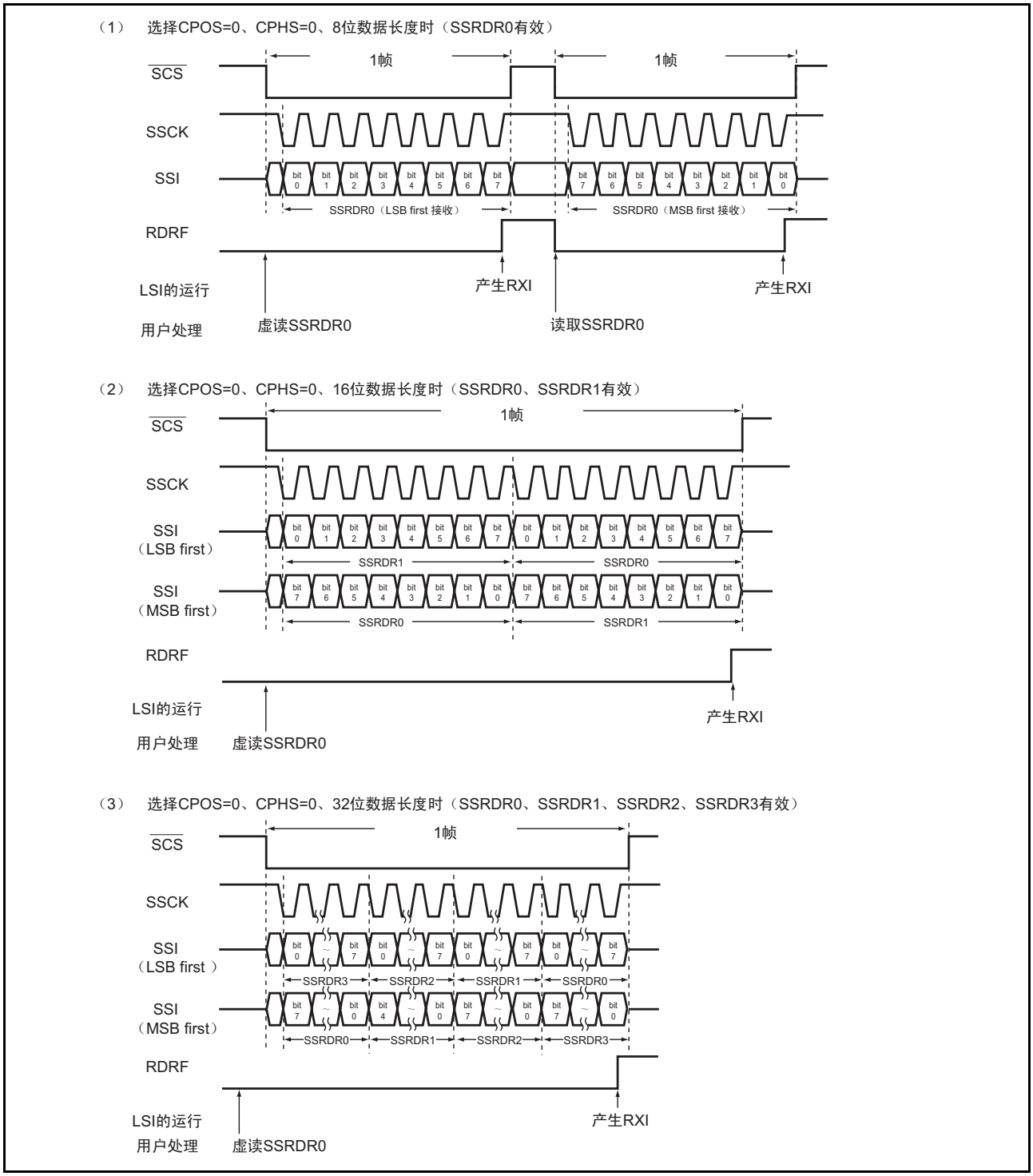
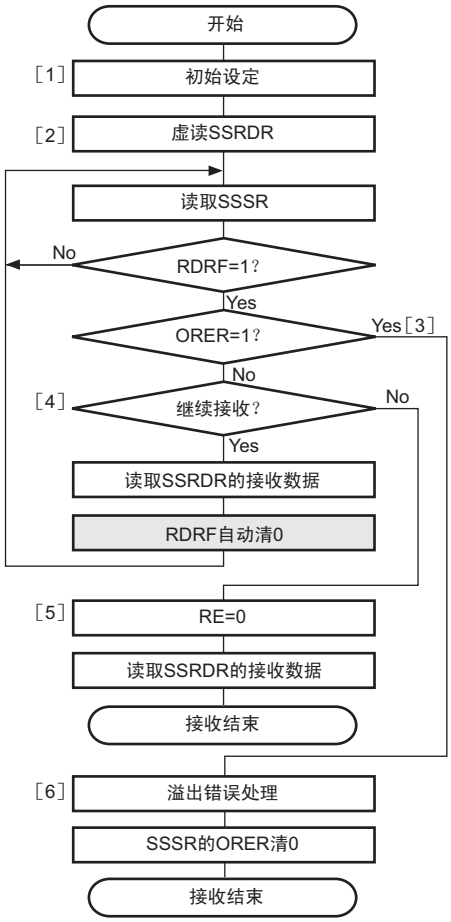


图 14.7　接收时的运行例子 （SSU 模式）



【注】 阴影部分为SSU内部运行

[1] 初始设定：
决定接收数据格式。

[2] 开始接收运行：
在RE=1的状态虚读SSRDR时，开始接收运行。

[3] [6] 接收错误处理：
产生接收错误时，读取SSSR的ORER标志，并执行指定的错误处理后，必须将ORER标志清0。在ORER标志置1的状态不可重新开始发送 / 接收。

[4] 继续单接收运行：
继续单接收运行时，在RDRF标志置1的状态，必须在等待t_{SUcyc}后读取SSRDR的接收数据。读取SSRDR的接收数据后，开始下一个单接收运行。

[5] 结束接收运行：
要结束接收运行，必须在RE清0后读取接收数据。如果在RE未清0时读取SSRDR，则不能再次开始接收运行。

图 14.8　接收数据的流程图例子 （SSU 模式）

(4)　发送 / 接收数据

发送 / 接收数据同时运行的流程图例子如图 14.9 所示。发送 / 接收数据为发送与接收数据的混合运行。在 TE=RE=1 的状态，通过对 SSTDR 写入发送数据开始发送 / 接收数据。

从发送模式 （TE=1）或接收模式 （RE=1）转换为发送 / 接收模式 （TE=RE=1）时，必须先将 TE、 RE 清 0 一次后再转换，确认 TEND、 RDRF 及 ORER 清 0 后，将 TE 及 RE 置 1。

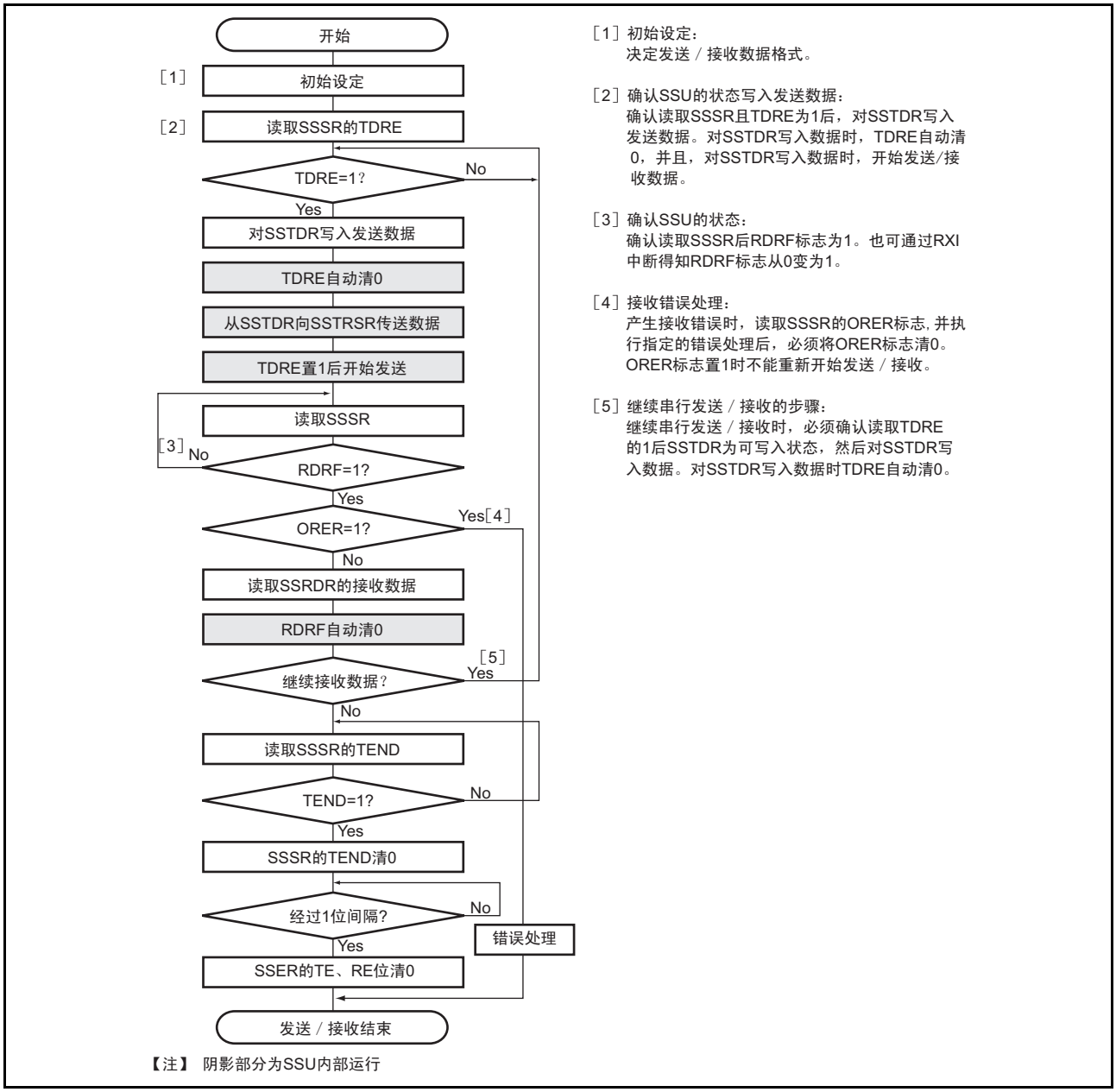


图 14.9　发送 / 接收数据同时运行的流程图例子 （SSU 模式）

14.4.6　$\overline{\text{SCS}}$ 引脚控制与竞争错误

设定 SSCRH 的 CSS1、 CSS0=10、 SSCRL 的 SSUMS=0 时，在 SSCRH 的 MSS 置 1 后开始串行传送前及传送结束后， $\overline{\text{SCS}}$ 引脚为输入 （Hi-Z）引脚，并检测竞争错误。在此期间，如果从 $\overline{\text{SCS}}$ 引脚输入低电平，则为竞争错误， SSSR 的 CE 位置位，并清除 MSS。

【注】 在竞争错误位置位的状态，不可执行之后的发送 / 接收。开始发送 / 接收前， CE 必须清 0。

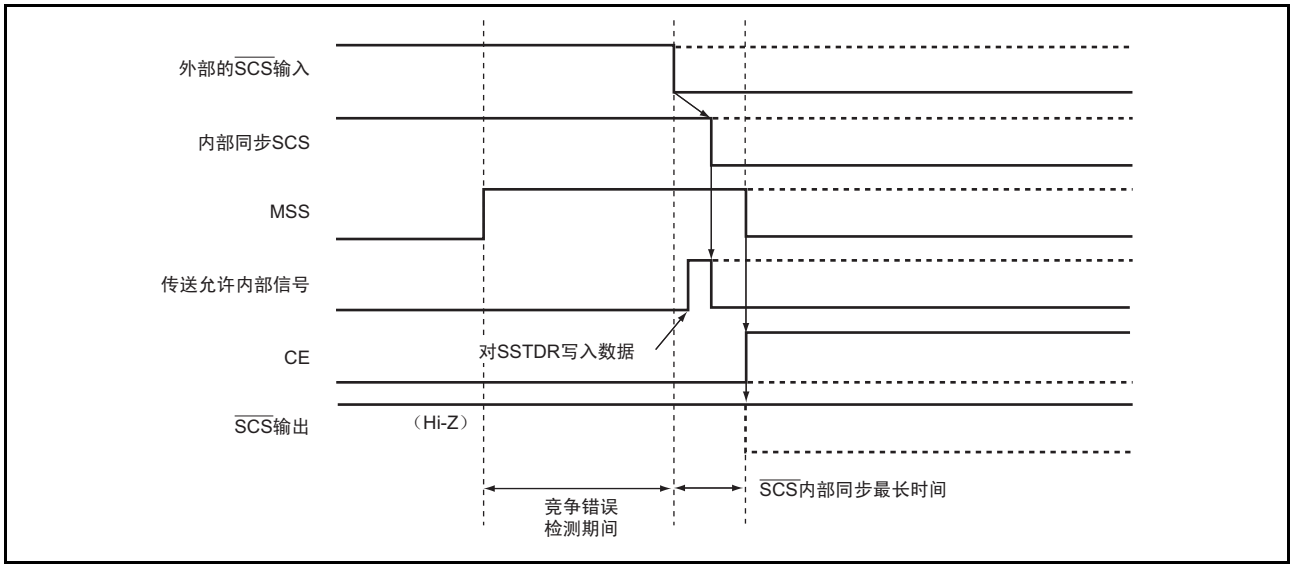


图 14.10　竞争错误检测时序 （传送前）

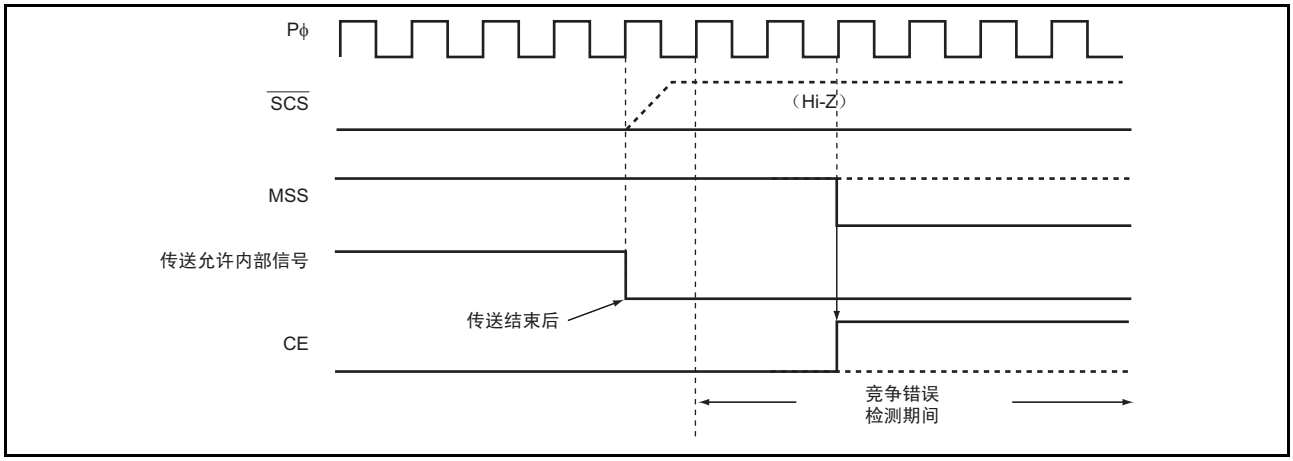


图 14.11　竞争错误检测时序 （传送结束后）

14.4.7　时钟同步通信模式

在时钟同步通信模式使用时钟线 （SSCK）、数据输入线 （SSI）及数据输出线 （SSO）等 3 条总线执行数据通信。

(1)　时钟同步通信模式的初始设定

时钟同步通信模式的初始设定例子如图 14.12 所示。发送 / 接收数据前，必须先将 SSER 的 TE 及 RE 清 0 后执行初始设定。

【注】 更改运行模式及通信格式时，必须将 TE 及 RE 清 0，还须注意：如果 TE 清 0，则 TDRE 置 1，但即使 RE 清 0，仍保持 RDRF、 ORER 的各标志及 SSRDR 的内容。

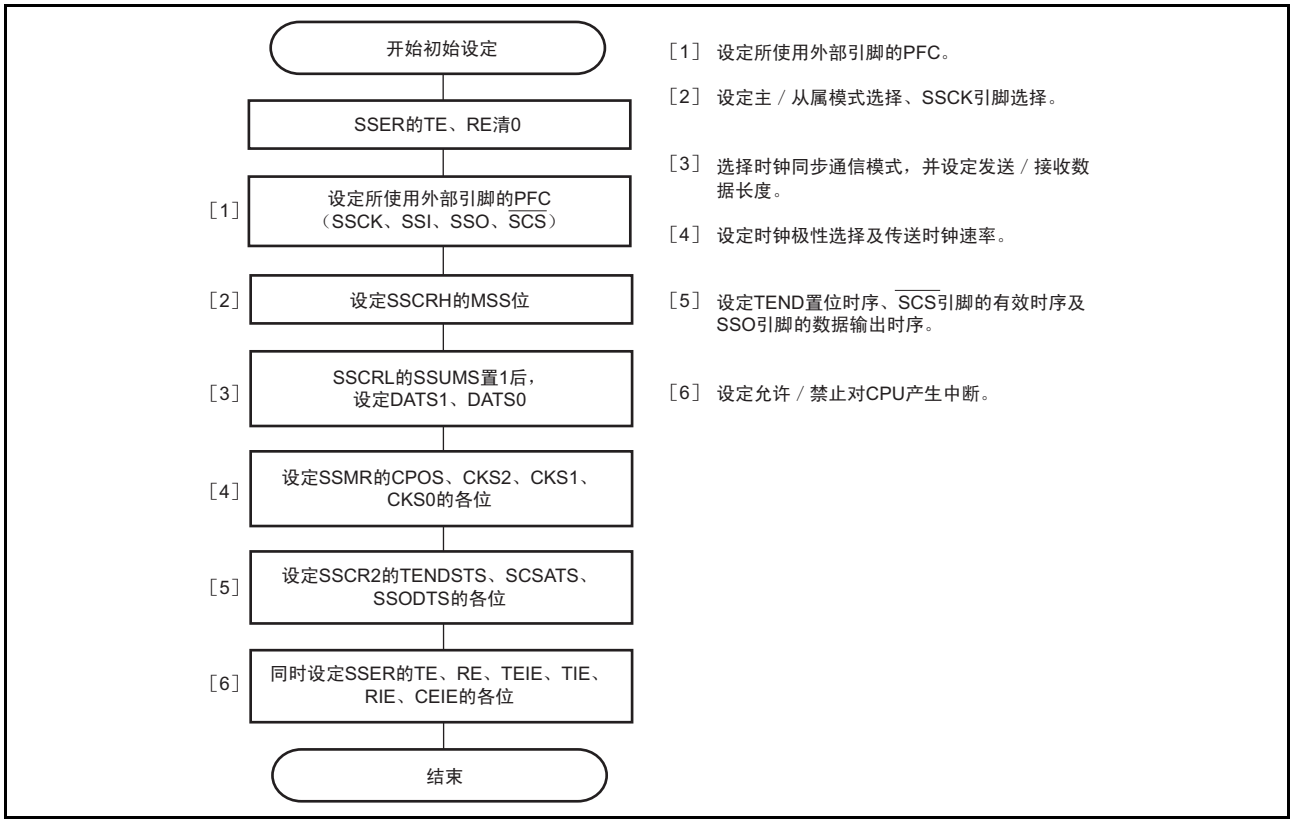


图 14.12　时钟同步通信模式的初始设定例子

(2)　发送数据

发送时的运行例子如图 14.13 所示，发送数据的流程图例子如图 14.14 所示。发送数据时，时钟同步通信模式如下运行：

如果将 SSU 设定为主器件，则输出传送时钟与数据；如果设定为从属器件，则从 SSCK 引脚输入传送时钟时，与该传送时钟同步输出数据。

SSU 在 SSER 的 TE 置 1 后，如果对 SSTDR 写入发送数据，则 SSSR 的 TDRE 自动清 0，并从 SSTDR 向 SSTRSR 传送数据，此后， TDRE 置 1，并开始发送。此时，如果 SSER 的 TIE 置 1，则产生 TXI 中断请求。

在 TDRE=0 的状态，如果结束 1 帧的数据传送，则从 SSTDR 向 SSTRSR 传送数据，并开始发送下一帧。在 TDRE=1 的状态，如果发送第 8 位，则 SSSR 的 TEND 置 1，并保持状态。此时，如果 SSER 的 TEIE 置 1，则产生 TEI 中断请求。

在 SSSR 的 ORER 置 1 的状态不可发送，发送前必须确认 ORER 是否清 0。

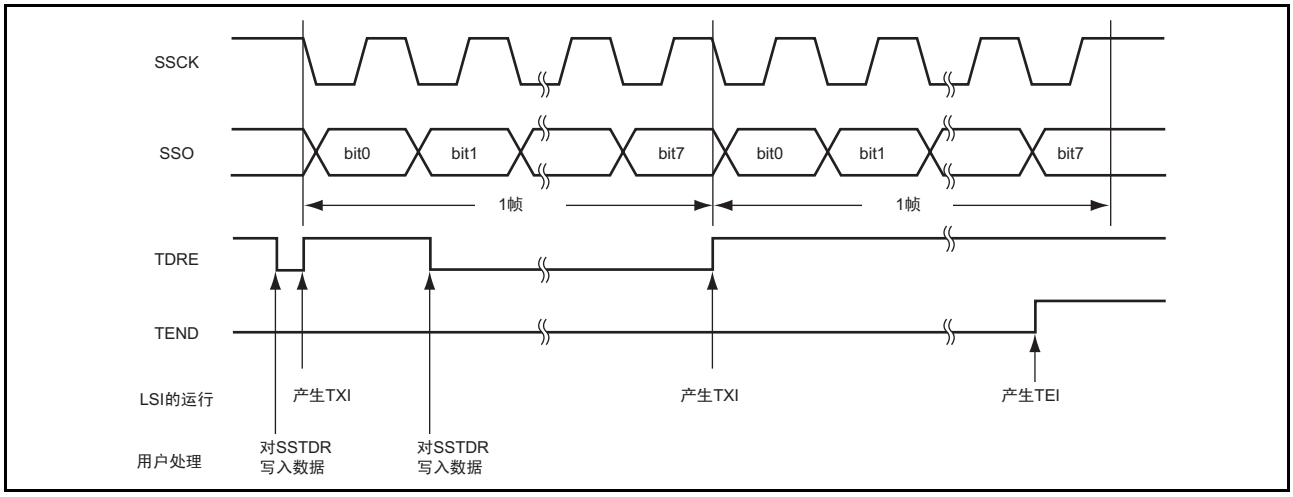


图 14.13　发送时的运行例子 （时钟同步通信模式）

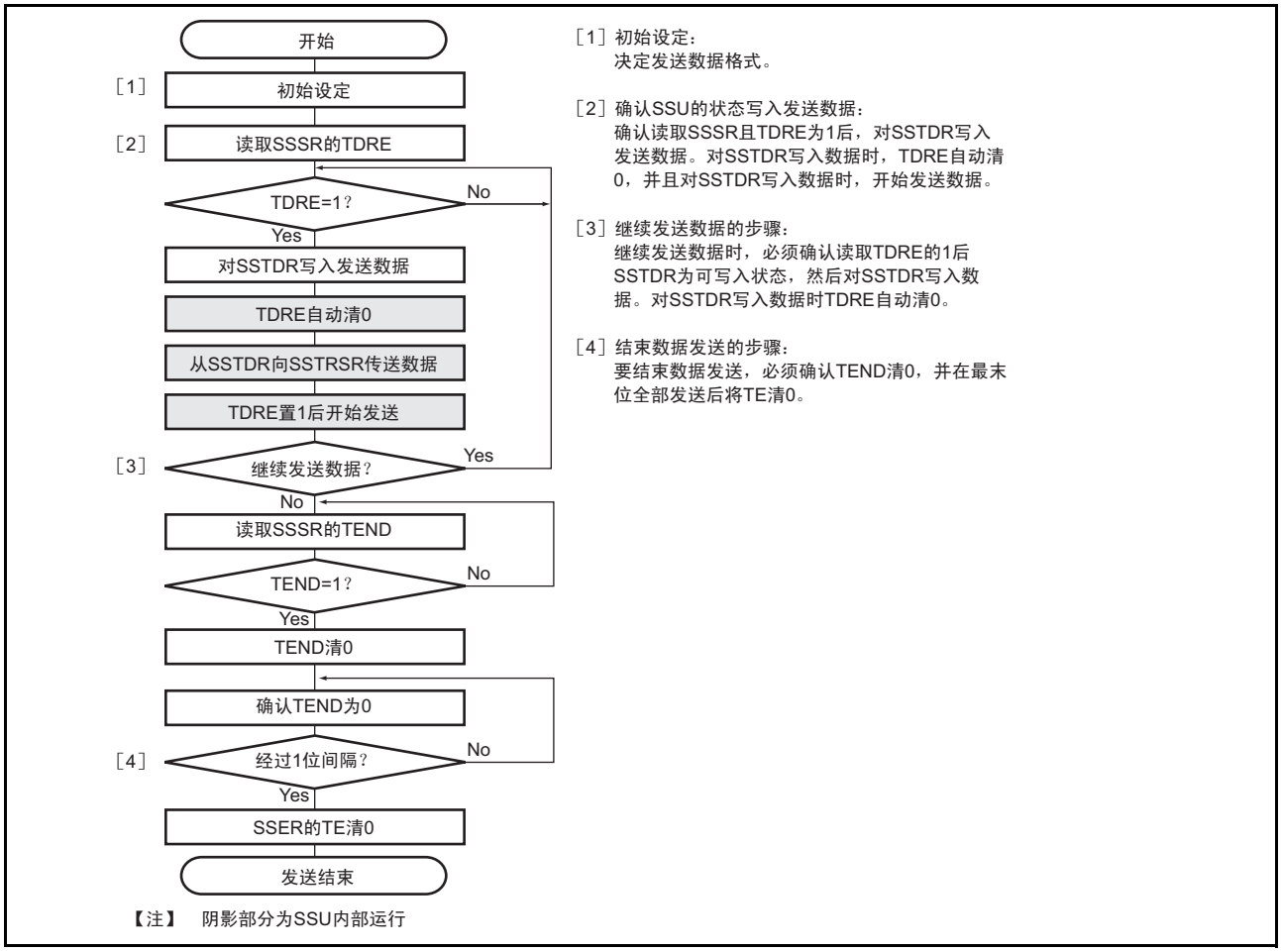


图 14.14　发送数据的流程图例子 （时钟同步通信模式）

(3)　接收数据

接收时的运行例子如图 14.15 所示，接收数据的流程图例子如图 14.16 所示。接收数据时 SSU 如下运行：

如果 SSER 的 RE 置 1，则 SSU 开始接收。

如果将 SSU 设定为主器件，则输出传送时钟，并输入接收数据；如果设定为从属器件，则从 SSCK 引脚输入传送时钟时，与该传送时钟同步输入接收数据。

接收 1 帧的数据后， SSSR 的 RDRF 置 1，并在 SSRDR 保存接收数据。此时，如果 SSER 的 RIE 置 1，则产生 RXI 中断请求；如果读取 SSRDR，则 RDRF 自动清 0。

在 RDRF=1 的状态如果第 8 个时钟上升，则 SSSR 的 ORER 置 1，产生溢出错误 （OEI），并停止接收。在 ORER=1 的状态不可接收，因此重新开始接收时 ORER 必须清 0。

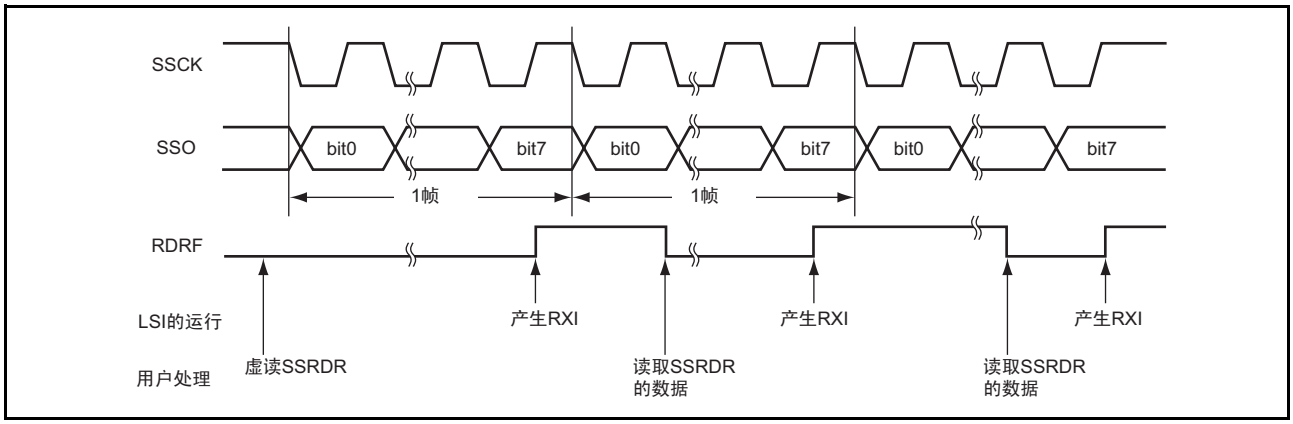


图 14.15　接收时的运行例子 （时钟同步通信模式）

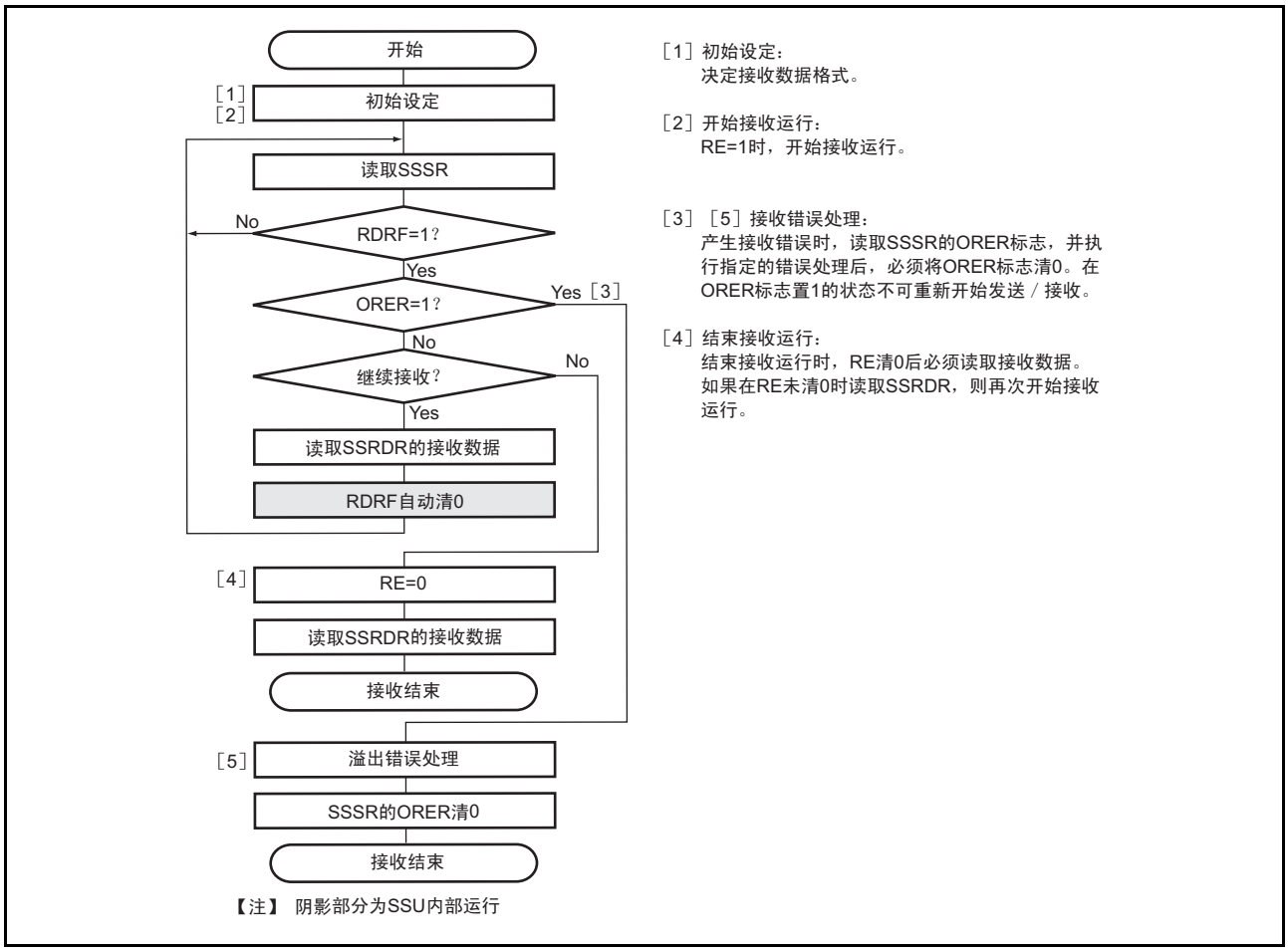


图 14.16　接收数据的流程图例子 （时钟同步通信模式）

(4)　发送 / 接收数据

发送 / 接收数据同时运行的流程图例子如图 14.17 所示。发送 / 接收数据为接收与发送数据的混合运行。在 TE=RE=1 的状态，通过对 SSTDR 写入发送数据开始发送 / 接收数据。

从发送模式 （TE=1）或接收模式 （RE=1）转换为发送 / 接收模式 （TE=RE=1）时，必须先将 TE、 RE 清 0 一次后再转换，确认 TEND、 RDRF、 ORER 清 0 后，将 TE 及 RE 置 1。

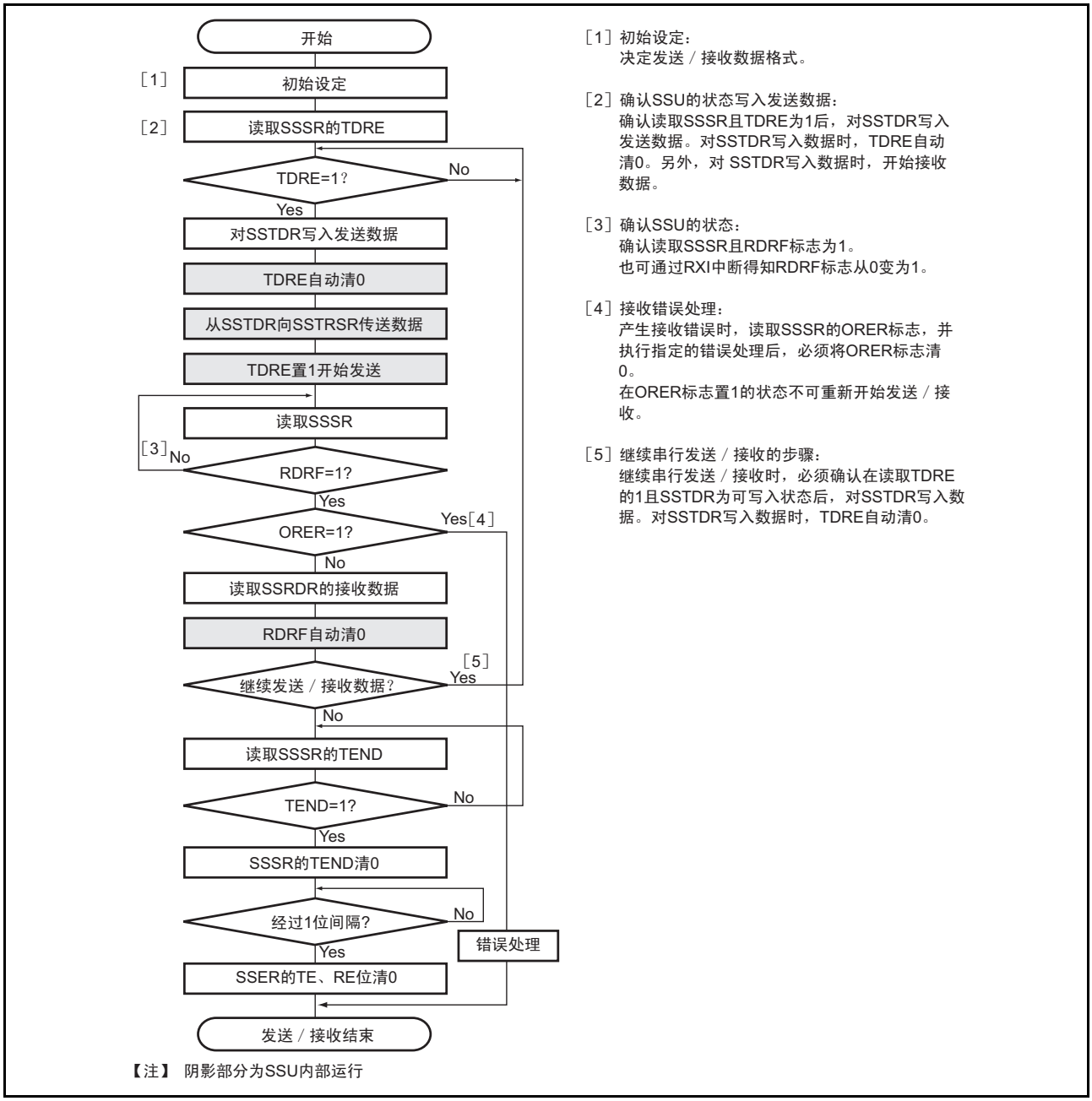


图 14.17　发送 / 接收数据同时运行的流程图例子 （时钟同步通信模式）

14.5　SSU 的中断源与 DTC

SSU 的中断请求有溢出错误、竞争错误、接收数据满、发送数据空及发送结束中断，可通过接收数据满、发送数据空的中断请求启动 DTC 后执行数据传送。

溢出错误、竞争错误的中断请求分配至 SSERI 的向量地址，发送数据空、发送结束的中断请求分配至 SSTXI 的向量地址，因此需根据标志判断中断源。中断源如表 14.8 所示。

如果表 14.8 的中断条件成立，则产生中断请求，必须通过 CPU 或 DTC 的数据传送清除中断源。

表 14.8　SSU 中断源

名称	中断源	简称	中断条件	DTC 的启动
SSERI	溢出错误	SSOEI	（RIE=1）・（ORER=1）	—
	竞争错误	SSCEI	（CEIE=1）・（CE=1）	—
SSRXI	接收数据满	SSRXI	（RIE=1）・（RDRF=1）	○
SSTXI	发送数据空	SSTXI	（TIE=1）・（TDRE=1）	○
	发送结束	SSTEI	（TEIE=1）・（TEND=1）	—

14.6　使用时的注意事项

14.6.1　模块待机模式的设定

SSU 可通过待机控制寄存器设定禁止 / 允许本模块的运行。如果为初始值，则 SSU 停止运行。解除模块待机模式，可存取寄存器。详情参阅 “第 22 章 低功耗模式”。

14.6.2　SSTDR、SSRDR 寄存器的存取

不可存取 SSCRL 寄存器的 DATS 位所设定的无效 SSTDR、SSRDR，否则可能无法正常执行之后的发送及接收。

14.6.3　在 SSU 从属模式连续发送 / 接收时的注意事项

在 SSU 从属模式连续发送 / 接收时，必须将每一帧的 $\overline{\text{SCS}}$ 引脚设置为无效 （高电平）。如果有超过 1 帧的 $\overline{\text{SCS}}$ 引脚设置为有效 （低电平），则无法正确发送 / 接收。

第 15 章 A/D 转换器 （ADC）

本 LSI 内置逐次逼近方式的 12 位 A/D 转换器。

15.1 特点

- 分辨率：12位
- 输入通道数：16个通道 （内置2个独立的A/D转换模块）
- 运行模式：两种
  单周期扫描模式：在1～8通道连续A/D转换
  连续扫描模式：在1～8通道重复A/D转换
- 16个12位A/D数据寄存器
  A/D_0与A/D_1分别有8个，共计16个16位A/D数据寄存器 （ADDR）。 A/D转换结果保存在对应各输入通道的A/D数据寄存器 （ADDR）。
- 采样＆保持功能
  本LSI的A/D转换器内置采样＆保持电路，因此可简单构成外部模拟输入电路。通道0～2、 8～10备有各通道专用的采样＆保持电路，因此可同时采样多个通道。下列组合可同时采样：
  组A （GrA）：同时采样从通道0、 1、 2中选择的模拟输入引脚
  组B （GrB）：同时采样从通道8、 9、 10中选择的模拟输入引脚
- A/D转换开始请求：3种
  软件：设定ADCR的ADST位
  定时器：MTU2的TRGAN、TRG0N、TRG4AN、TRG4BN、MTU2S的TRGAN、TRG4AN、TRG4BN
  外部触发：$\overline{\text{ADTRG}}$ （LSI引脚）
- 模拟输入通道选择功能
  通过设定A/D模拟输入通道选择寄存器 （ADANSR），可选择任意通道执行A/D转换。
- 支持A/D转换结束中断及DTC传送功能
  A/D转换结束时，可对CPU产生A/D转换结束中断 （ADI_3、 ADI_4），并可由ADI_3、 ADI_4启动DTC。

A/D 转换器框图如图 15.1 所示。

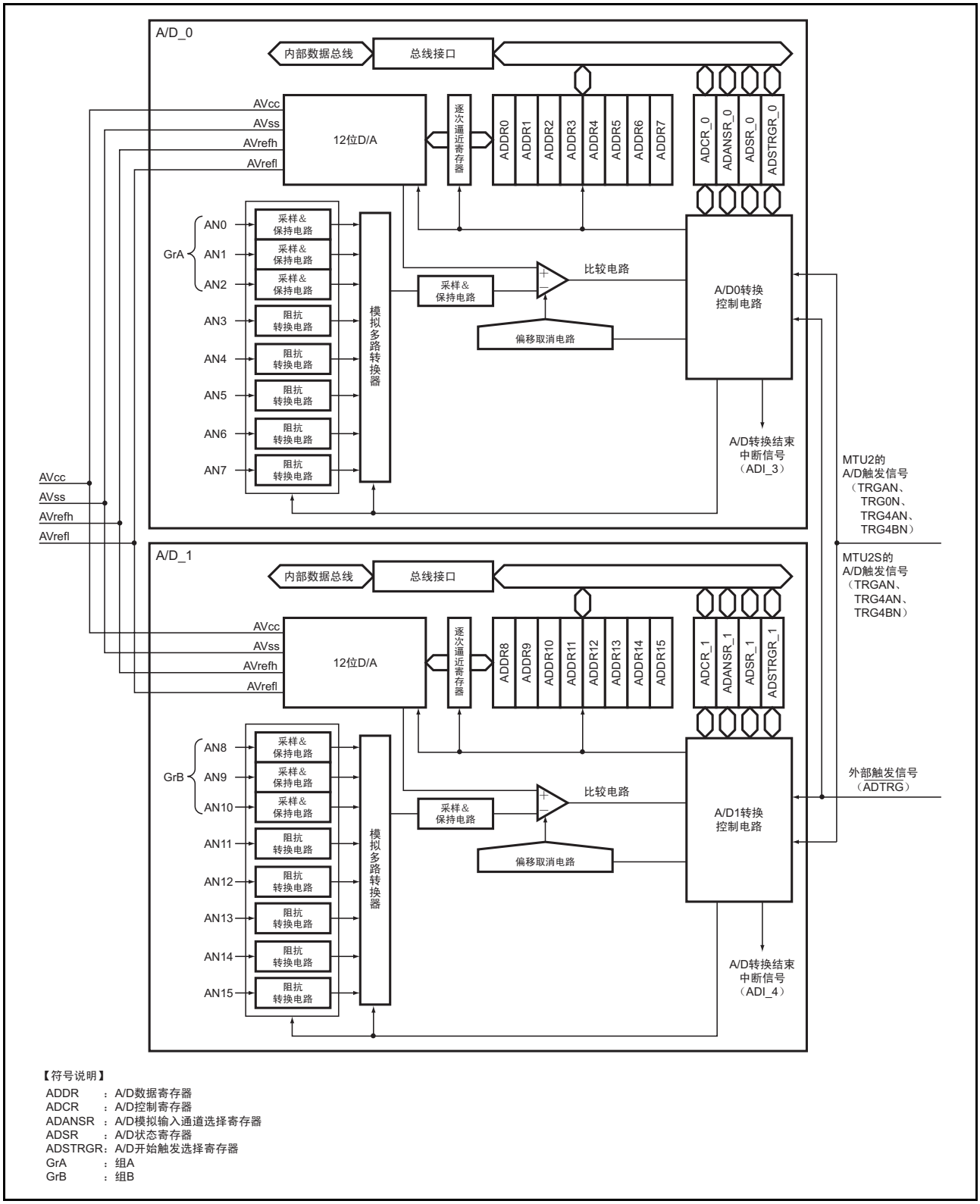


图 15.1　A/D 转换器框图

15.2　输入 / 输出引脚

用于 A/D 转换器的引脚如表 15.1 所示。有关使用引脚的注意事项，详情参阅 “15.7 使用时的注意事项”。

表 15.1　引脚结构

模块分类	引脚名称	输入 / 输出	功　能
通用	AV_{CC}	输入	模拟部的电源引脚
	AV_{SS}	输入	模拟部的接地引脚
	AV_{refh}	输入	模拟部的基准电源引脚 （高电平侧）（$AV_{refl} < AV_{refh}$）
	AV_{refl}	输入	模拟部的基准电源引脚 （低电平侧）（$AV_{refl} < AV_{refh}$）
	$\overline{ADTRG}$	输入	A/D 外部触发输入引脚
A/D 模块 0 （A/D_0）	AN0	输入	模拟输入引脚 0 （组 A）
	AN1	输入	模拟输入引脚 1 （组 A）
	AN2	输入	模拟输入引脚 2 （组 A）
	AN3	输入	模拟输入引脚 3
	AN4	输入	模拟输入引脚 4
	AN5	输入	模拟输入引脚 5
	AN6	输入	模拟输入引脚 6
	AN7	输入	模拟输入引脚 7
A/D 模块 1 （A/D_1）	AN8	输入	模拟输入引脚 8 （组 B）
	AN9	输入	模拟输入引脚 9 （组 B）
	AN10	输入	模拟输入引脚 10 （组 B）
	AN11	输入	模拟输入引脚 11
	AN12	输入	模拟输入引脚 12
	AN13	输入	模拟输入引脚 13
	AN14	输入	模拟输入引脚 14
	AN15	输入	模拟输入引脚 15

15.3 寄存器说明

A/D 转换器有以下寄存器。

表 15.2 寄存器结构

寄存器名称	简称	R/W	初始值	地址	存取长度
A/D 控制寄存器 _0	ADCR_0	R/W	H'00	H'FFFFD400	8
A/D 状态寄存器 _0	ADSR_0	R/W	H'00	H'FFFFD402	8
A/D 开始触发选择寄存器 _0	ADSTRGR_0	R/W	H'00	H'FFFFD41C	8
A/D 模拟输入通道选择寄存器 _0	ADANSR_0	R/W	H'00	H'FFFFD420	8
A/D 数据寄存器 0	ADDR0	R	H'0000	H'FFFFD440	16
A/D 数据寄存器 1	ADDR1	R	H'0000	H'FFFFD442	16
A/D 数据寄存器 2	ADDR2	R	H'0000	H'FFFFD444	16
A/D 数据寄存器 3	ADDR3	R	H'0000	H'FFFFD446	16
A/D 数据寄存器 4	ADDR4	R	H'0000	H'FFFFD448	16
A/D 数据寄存器 5	ADDR5	R	H'0000	H'FFFFD44A	16
A/D 数据寄存器 6	ADDR6	R	H'0000	H'FFFFD44C	16
A/D 数据寄存器 7	ADDR7	R	H'0000	H'FFFFD44E	16
A/D 控制寄存器 _1	ADCR_1	R/W	H'00	H'FFFFD600	8
A/D 状态寄存器 _1	ADSR_1	R/W	H'00	H'FFFFD602	8
A/D 开始触发选择寄存器 _1	ADSTRGR_1	R/W	H'00	H'FFFFD61C	8
A/D 模拟输入通道选择寄存器 _1	ADANSR_1	R/W	H'00	H'FFFFD620	8
A/D 数据寄存器 8	ADDR8	R	H'0000	H'FFFFD640	16
A/D 数据寄存器 9	ADDR9	R	H'0000	H'FFFFD642	16
A/D 数据寄存器 10	ADDR10	R	H'0000	H'FFFFD644	16
A/D 数据寄存器 11	ADDR11	R	H'0000	H'FFFFD646	16
A/D 数据寄存器 12	ADDR12	R	H'0000	H'FFFFD648	16
A/D 数据寄存器 13	ADDR13	R	H'0000	H'FFFFD64A	16
A/D 数据寄存器 14	ADDR14	R	H'0000	H'FFFFD64C	16
A/D 数据寄存器 15	ADDR15	R	H'0000	H'FFFFD64E	16

15.3.1　A/D 控制寄存器 _0、 _1 （ADCR_0、 ADCR_1）

ADCR 为可读取 / 写入的 8 位寄存器，用于选择 A/D_0、 A/D_1 的转换模式等。

位：	7	6	5	4	3	2	1	0
	ADST	ADCS	ACE	ADIE	-	-	TRGE	EXTRG
初始值：	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R	R	R/W	R/W

位	位名称	初始值	R/W	说　明
7	ADST	0	R/W	A/D 转换开始 如果清 0，则中止 A/D 转换，进入待机状态；如果置 1，则开始 A/D 转换。在单周期扫描模式，如果所选通道的 A/D 转换结束，则自动清除。在连续扫描模式，由软件、复位、软件待机模式、硬件待机模式或模块待机模式清除前，依次连续转换所选通道。
6	ADCS	0	R/W	A/D 连续扫描 本位可选择扫描模式时的单周期扫描 / 连续扫描，仅在扫描模式时有效。 0：单周期扫描 1：连续扫描 在 ADST 为 0 的状态转换运行模式。
5	ACE	0	R/W	自动清除允许 由 CPU、 DTC 读取 ADDR 寄存器后，允许 / 禁止自动清除 ADDR 寄存器。本位置 1 时，由 CPU 及 DTC 读取 ADDR 寄存器后，自动将 ADDR 寄存器清除为 H'0000。可通过此功能检测 ADDR 寄存器的未更新故障。 0：读取 ADDR 后，禁止自动清除 ADDR 1：读取 ADDR 后，允许自动清除 ADDR
4	ADIE	0	R/W	A/D 中断允许 允许 / 禁止对 CPU 产生 A/D 转换结束中断 （ADI_3、 ADI_4）。为了防止误动作，必须在 ADST 为 0 的状态转换运行模式。 A/D 转换结束后， ADSR 寄存器的 ADF 位置 1 时，如果本位置 1，则对 CPU 产生 ADI_3、ADI_4。可由 ADF 或 ADIE 清 0，清除 ADI_3、 ADI_4。 0：禁止产生 A/D 转换结束中断 1：允许产生 A/D 转换结束中断
3、 2	—	均为 0	R	保留位 读取值、写入值总是为 0。
1	TRGE	0	R/W	触发允许 允许 / 禁止由外部触发输入 （$\overline{\text{ADTRG}}$）或 MTU2/MTU2S 的 A/D 转换开始触发 （MTU2 的 TRGAN、TRG0N、TRG4AN、TRG4BN、MTU2S 的 TRGAN、TRG4AN、TRG4BN）开始 A/D 转换。 参照 EXTRG 位的说明，选择外部触发与 MTU2、 MTU2S 的 A/D 转换开始触发。 0：禁止由外部触发或 MTU2/MTU2S 的 A/D 转换开始触发开始 A/D 转换 1：允许由外部触发或 MTU2/MTU2S 的 A/D 转换开始触发开始 A/D 转换
0	EXTRG	0	R/W	选择触发 选择外部触发 （$\overline{\text{ADTRG}}$），或 MTU2、 MTU2S 的 A/D 转换开始触发作为 A/D 转换开始触发。 设定为外部触发 （EXTRG=1）时， TRGE 置 1 后，如果对 $\overline{\text{ADTRG}}$ 引脚输入低电平脉冲，则 A/D 转换检测脉冲的下降沿，且 ADCR 寄存器的 ADST 位置 1。此后，执行与通过软件对 ADST 位写入 1 时相同的运行。但由外部触发输入开始 A/D 转换的功能仅在 ADST 位清 0 时有效。 使用外部触发开始功能时，对 $\overline{\text{ADTRG}}$ 引脚输入的低电平脉宽必须大于等于 1.5Pϕ 时钟。 0：由 MTU2/MTU2S 的 A/D 转换开始触发启动 A/D 转换器 1：由外部引脚 （$\overline{\text{ADTRG}}$）启动 A/D 转换器

15.3.2　A/D 状态寄存器 _0、 _1 （ADSR_0、 ADSR_1）

ADSR 为可读取 / 写入的 8 位寄存器，表示 A/D 转换器的状态。

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

【注】* 为了清除标志，仅可在读取1后写入0。
标志为0时，不得重写0。

位	位名称	初始值	R/W	说　明
7 ～ 1	—	均为 0	R	保留位 读取值、写入值总是为 0。
0	ADF	0	R/(W)*	A/D 结束标志 本位是表示 A/D 转换结束的状态标志。 ［置位条件］ • 在扫描模式选择的所有通道的 A/D 转换结束时 ［清除条件］ • 读取 1 的状态后写入 0 时 • 由 ADI 中断启动 DTC 并读取 ADDR 时

【注】 * 为了清除标志，仅可在读取 1 后写入 0。标志为 0 时，不得重写 0。

15.3.3　A/D 开始触发选择寄存器 _0、 _1 （ADSTRGR_0、 ADSTRGR_1）

ADCR 寄存器的 TRGE 位设定为 1 且 ADCR 寄存器的 EXTRG 位设定为 0 时， ADSTRGR 可选择 MTU2 或 MTU2S 的 A/D 转换开始触发作为 A/D 转换开始源。

位：	7	6	5	4	3	2	1	0
	-	STR6	STR5	STR4	STR3	STR2	STR1	STR0
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	位名称	初始值	R/W	说　明
7	—	0	R	保留位 读取值、写入值总是为 0。
6	STR6	0	R/W	开始触发 6 设定允许 / 禁止从 MTU2S 输入的 A/D 转换开始请求。 0：禁止由 TRGAN 触发 （MTU2S）开始 A/D 转换 1：允许由 TRGAN 触发 （MTU2S）开始 A/D 转换
5	STR5	0	R/W	开始触发 5 设定允许 / 禁止从 MTU2S 输入的 A/D 转换开始请求。 0：禁止由 TRG4AN 触发 （MTU2S）开始 A/D 转换 1：允许由 TRG4AN 触发 （MTU2S）开始 A/D 转换
4	STR4	0	R/W	开始触发 4 设定允许 / 禁止从 MTU2S 输入的 A/D 转换开始请求。 0：禁止由 TRG4BN 触发 （MTU2S）开始 A/D 转换 1：允许由 TRG4BN 触发 （MTU2S）开始 A/D 转换
3	STR3	0	R/W	开始触发 3 设定允许 / 禁止从 MTU2 输入的 A/D 转换开始请求。 0：禁止由 TRG0N 触发 （MTU2）开始 A/D 转换 1：允许由 TRG0N 触发 （MTU2）开始 A/D 转换
2	STR2	0	R/W	开始触发 2 设定允许 / 禁止从 MTU2 输入的 A/D 转换开始请求。 0：禁止由 TRGAN 触发 （MTU2）开始 A/D 转换 1：允许由 TRGAN 触发 （MTU2）开始 A/D 转换
1	STR1	0	R/W	开始触发 1 设定允许 / 禁止从 MTU2 输入的 A/D 转换开始请求。 0：禁止由 TRG4AN 触发 （MTU2）开始 A/D 转换 1：允许由 TRG4AN 触发 （MTU2）开始 A/D 转换
0	STR0	0	R/W	开始触发 0 设定允许 / 禁止从 MTU2 输入的 A/D 转换开始请求。 0：禁止由 TRG4BN 触发 （MTU2）开始 A/D 转换 1：允许由 TRG4BN 触发 （MTU2）开始 A/D 转换

15.3.4　A/D 模拟输入通道选择寄存器 _0、_1 （ADANSR_0、ADANSR_1）

ADANSR 为可读取 / 写入的 8 位寄存器，用于选择模拟输入通道。

位：	7	6	5	4	3	2	1	0
	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
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	位名称	初始值	R/W	说　明
7	ANS7	0	R/W	通过将 A/D 模拟输入通道选择寄存器的位设定为 1，选择与设定位对应的通道。模拟输入引脚及位的对应关系参照表 15.3。 为了防止误动作，必须在 ADCR 寄存器的 ADST 位为 0 的状态转换模拟输入通道。
6	ANS6	0	R/W	
5	ANS5	0	R/W	
4	ANS4	0	R/W	
3	ANS3	0	R/W	
2	ANS2	0	R/W	
1	ANS1	0	R/W	
0	ANS0	0	R/W	

表 15.3　通道选择一览表

位名称	模拟输入通道	
	A/D_0	A/D_1
ANS0	AN0	AN8
ANS1	AN1	AN9
ANS2	AN2	AN10
ANS3	AN3	AN11
ANS4	AN4	AN12
ANS5	AN5	AN13
ANS6	AN6	AN14
ANS7	AN7	AN15

15.3.5　A/D 数据寄存器 0 ～ 15 （ADDR0 ～ ADDR15）

ADDR 为保存 A/D 转换结果的 16 位只读寄存器。各模拟输入通道的转换结果保存在相应编号的 ADDR（参照表 15.4）。

12 位的转换数据保存在 ADDR 的 bit11 ～ bit0。

ADDR 的初始值为 H'0000。

通过将 ADCR 寄存器的 ACE 位置 1，读取 ADDR 寄存器后，可自动将 ADDR 寄存器清除为 H'0000。

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

位	位名称	初始值	R/W	说　明
15 ～ 12	—	均为 0	R	保留位
11 ～ 0	ADD[11:0]	均为 0	R	位数据 （12 位）

表 15.4　模拟输入通道与 ADDR0 ～ ADDR15 寄存器的对应关系

A/D_0 转换器		A/D_1 转换器	
模拟输入通道	A/D 数据寄存器	模拟输入通道	A/D 数据寄存器
AN0	ADDR0	AN8	ADDR8
AN1	ADDR1	AN9	ADDR9
AN2	ADDR2	AN10	ADDR10
AN3	ADDR3	AN11	ADDR11
AN4	ADDR4	AN12	ADDR12
AN5	ADDR5	AN13	ADDR13
AN6	ADDR6	AN14	ADDR14
AN7	ADDR7	AN15	ADDR15

15.3.6 与 CPU 的接口

连接 CPU 的芯片内总线宽度为 16 位，因此无需个别读取高位字节 / 低位字节的数据。

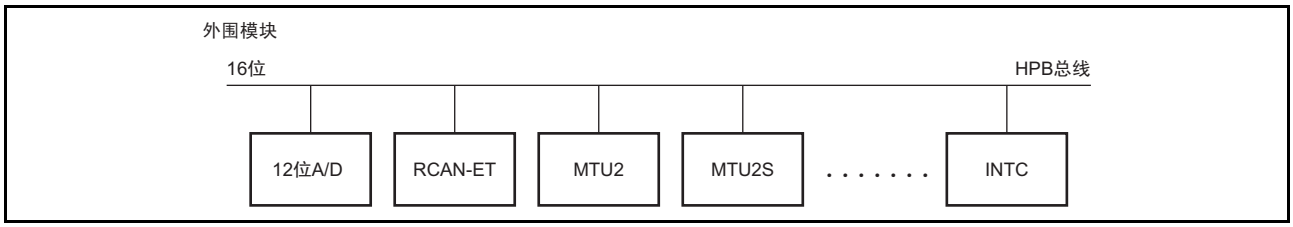


图 15.2 CPU 与 12 位 A/D 转换器的接口

15.4 运行说明

A/D 转换的运行模式有单周期扫描模式与连续扫描模式两种。单周期扫描模式为执行 1 次不少于 1 个通道的 A/D 转换后就结束的模式，连续扫描模式为 ADST 位清 0 前重复执行指定的不少于 1 个通道的 A/D 转换的模式。

由 A/D 控制寄存器 （ADCR）的 ADCS 位选择运行模式。设定为单周期扫描模式时， ADCS 位清 0 ；设定为连续扫描模式时， ADCS 位置 1。如果两个模式同时开始转换，则按照 A/D 模拟输入通道选择寄存器（ADANSR）选择的模拟输入通道编号由小到大的顺序 （A/D_0：AN0→ AN7， A/D_1：AN8 →AN15）执行 A/D 转换。

在单周期扫描模式，如果结束 1 次所有设定通道的转换，则 ADSR 的 ADF 位置 1， ADST 位自动清 0。在连续扫描模式，如果结束所有设定通道的转换，则 ADSR 的 ADF 位置 1。停止 A/D 转换时，必须对 ADST 位写入 0。 ADF 位置 1 时，如果 ADCR 的 ADIE 位置 1，则产生 A/D 转换结束中断 （ADI）； ADF 位清 0 时，必须在读取 ADF 位的 1 后，写入 0。但由 ADI 中断启动 DTC 时， ADF 位自动清 0。

15.4.1 单周期扫描模式

选择模拟输入通道 0 ～ 3 （AN0 ～ AN3），在 4 个通道单周期扫描 A/D_0 转换的运行例子如下所示：该运行同样适用于 A/D_1 的 A/D 转换。

1. A/D控制寄存器_0 （ADCR_0）的ADCS位清0。
2. A/D模拟输入通道选择寄存器_0 （ADANSR_0）的ANS0～ANS3位均置1。
3. A/D控制寄存器_0 （ADCR_0）的ADST位置1后开始A/D转换。
4. 同时采样通道0～通道2 （GrA）后，执行偏移取消处理 （OFC）。此后，开始通道0的A/D转换A/D转换结束后，将A/D转换结果传送至ADDR0寄存器。之后转换通道1， A/D转换结束后，将A/D转换结果传送至ADDR1寄存器。此后，同样转换通道2，将A/D转换结果传送至ADDR2寄存器。然后开始通道3的A/D转换， A/D转换结束后，将A/D转换结果传送至ADDR3寄存器。
5. 结束所有设定通道 （AN0～AN3）的转换后，则ADF位置1， ADST位自动清0，并停止A/D转换。此时，如果ADIE位置1，则在A/D转换结束后，产生ADI_3中断。

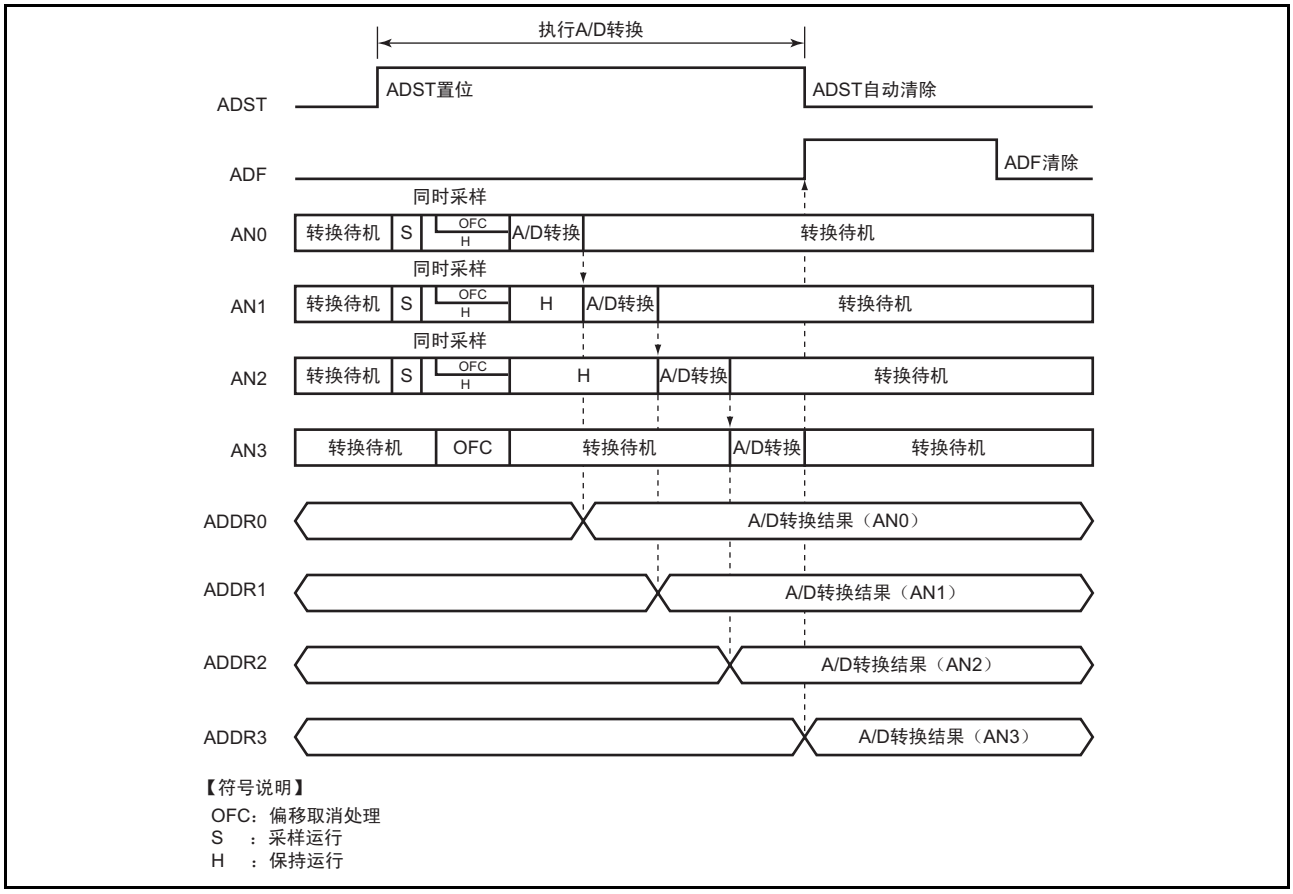


图 15.3　A/D_0 转换器的运行例子 （单周期扫描模式）

15.4.2　连续扫描模式

选择模拟输入通道 0、 2、 3 （AN0、 AN2、 AN3），在 3 个通道连续扫描 A/D_0 转换的运行例子如下所示：此运行同样适用于 A/D_1 的 A/D 转换。

1. A/D控制寄存器_0 （ADCR_0）的ADCS位置1。
2. A/D模拟输入通道选择寄存器_0 （ADANSR_0）的ANS0、 ANS2、 ANS3位均置1。
3. A/D控制寄存器_0 （ADCR_0）的ADST位置1后开始A/D转换。
4. 同时采样通道0、 2 （GrA）。此时， ADANSR_0寄存器的ANS1位设定为0，因此不采样通道1。执行偏移取消处理 （OFC），开始通道0的A/D转换， A/D转换结束后，将A/D转换结果传送至ADDR0寄存器。然后转换通道2， A/D转换结束后，将A/D转换结果传送至ADDR2寄存器。通道1不执行A/D转换。
5. 此后，开始通道3的A/D转换， A/D转换结束后，则将A/D转换结果传送至ADDR3寄存器。
6. 结束所有设定通道 （AN0、 AN2、 AN3）的转换后 ， ADF位置1。此时，如果ADIE位置1，则在A/D转换结束后，产生ADI_3中断。
7. ADST位置1期间，重复执行4.～6.。如果ADST位清0，则停止A/D转换，此后，如果ADST位置1，则再次开始A/D转换，并重复执行4.～6.。

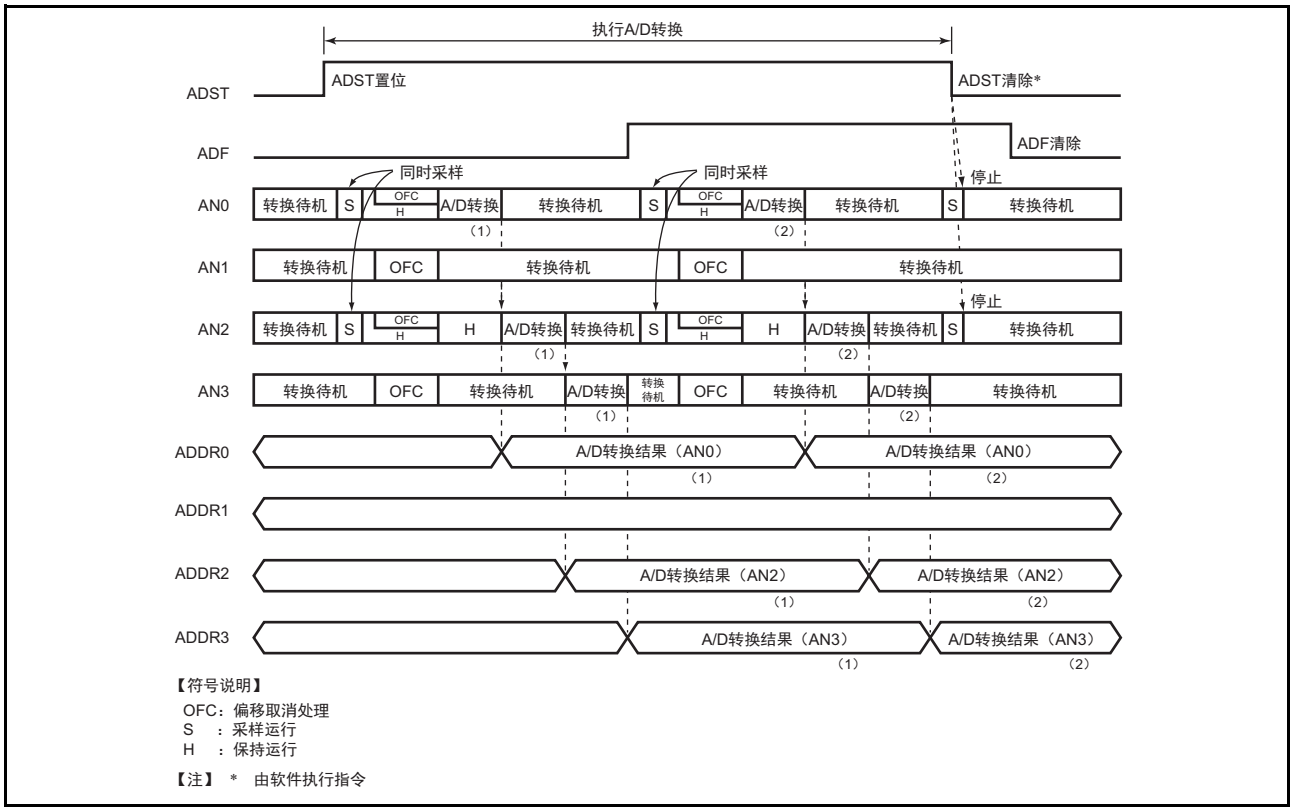


图 15.4　A/D_0 转换器的运行例子 （连续扫描模式）

15.4.3　输入采样与 A/D 转换时间

A/D_0 内置各通道通用的采样 & 保持电路。 A/D_0 的通道 0 ～ 2 内置各通道专用的采样 & 保持电路，可作为一组同时采样，该组称为组 A （GrA）（表 15.5）。即使在组内通过 ADANSR 寄存器仅选择 1 个通道，仍执行使用专用采样 & 保持电路的采样 & 保持运行；通过 ADANSR 寄存器设定不具有专用采样 & 保持电路的通道，仍要经过与使用专用采样 & 保持电路相同的时间。

上述情况也适用于 A/D_1。

产生 CPU 写入、 MTU2 与 MTU2S 引起的 A/D 启动请求及外部触发等 ADST 置位源时，经过 A/D 转换开始延迟时间 （t_D）后，通过各通道专用的采样 & 保持电路采样模拟输入，然后执行偏移取消处理 （OFC）。此后，通过各通道通用的采样 & 保持电路采样模拟输入后，开始 A/D 转换。此时 A/D 转换的时序如图 15.5 所示。 A/D 转换时间 （t_{CONV}）包含 t_D 与偏移取消处理时间 （t_{OFC}）、各通道专用采样 & 保持电路的模拟输入采样时间 （t_{SPLSH}）及各通道通用采样 & 保持电路的模拟输入采样时间 （t_{SPL}），另外 t_{SPLSH} 不因同时采样的通道数而改变。

在连续扫描模式，表 15.6 所示的 A/D 转换时间 （t_{CONV}）相当于第 1 个周期的转换时间。第 2 个周期以后，用 （$t_{CONV} - t_D + 6$）表示所有通道转换结束的转换时间。

表 15.5　模拟输入通道与可同时采样的组的对应关系

A/D_0 转换器		A/D_1 转换器	
模拟输入通道	组	模拟输入通道	组
AN0	GrA	AN8	GrB
AN1		AN9	
AN2		AN10	
AN3	—	AN11	—
AN4	—	AN12	—
AN5	—	AN13	—
AN6	—	AN14	—
AN7	—	AN15	—

表 15.6　A/D 转换时间

项　目	符号	需要状态数		
		Min.	Typ.	Max.
A/D 转换开始延迟时间	t_D	11*1	—	15*2
GrA/GrB 专用采样＆保持电路的模拟输入采样时间	t_{SPLSH}	—	30	—
偏移取消处理时间	t_{OFC}	—	50	—
各通道通用采样＆保持电路的模拟输入采样时间	t_{SPL}	—	20	—
A/D 转换时间	t_{CONV}	50n+95*3	—	50n+99*3

【注】 *1　由 MTU2、 MTU2S 触发信号启动 A/D 时

*2　由外部触发信号启动 A/D 时

*3　n 为转换通道数 （n=1 ～ 8）

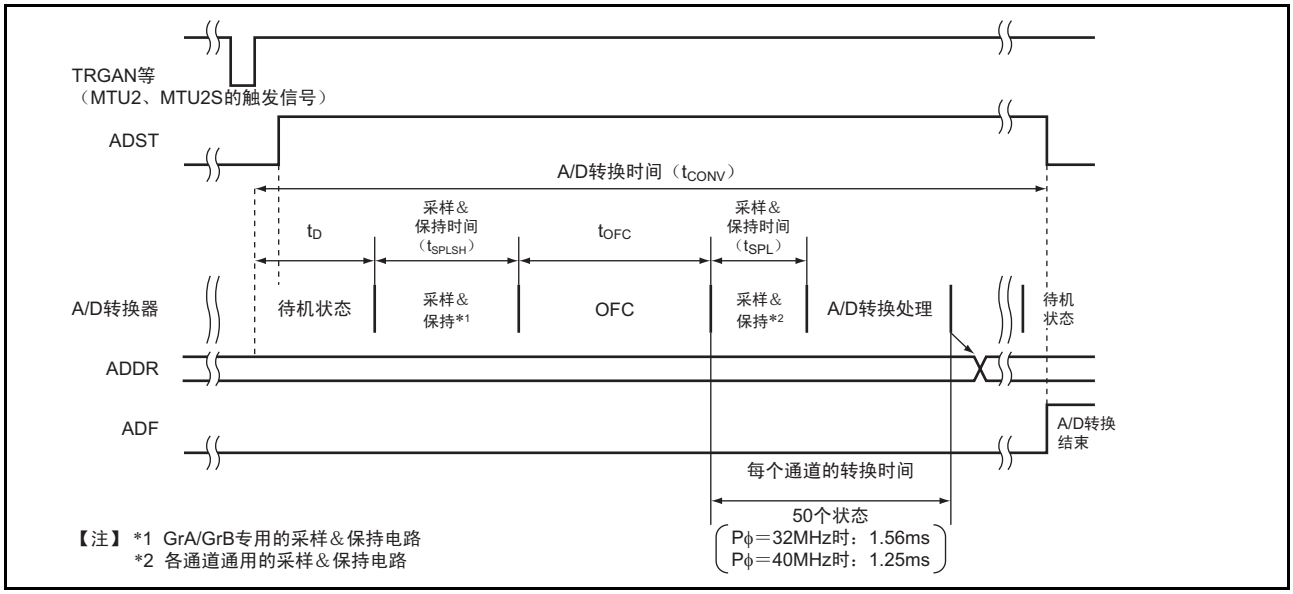


图 15.5　A/D 转换时序 （单周期扫描模式）

15.4.4　由 MTU2、 MTU2S 启动 A/D 转换器

可由 MTU2 的 A/D 转换开始触发（TRGAN、 TRG0N、 TRG4AN、 TRG4BN）、 MTU2S 的 A/D 转换开始触发（TRGAN、 TRG4AN、 TRG4BN）启动 A/D 转换。要使此功能有效时，必须将 ADCR 寄存器的 TRGE 位置 1， EXTRG 位清 0。此状态时，如果产生 ADSTRGR 寄存器设定的 MTU2、 MTU2S 的 A/D 转换开始触发，则 ADST 位置 1。从 ADST 位置 1 到 A/D 转换开始的时序与 A/D 转换启动源无关，均相同。

必须在设定 ADCR 寄存器、 ADSTRGR 寄存器、 ADANSR 寄存器后，输入 A/D 转换开始触发。

15.4.5　外部触发输入时序

可由外部触发输入开始 A/D 转换。为了执行外部触发输入，必须通过引脚功能控制器 （PFC）将引脚功能设定为 $\overline{\text{ADTRG}}$，并对 $\overline{\text{ADTRG}}$ 引脚输入高电平，在 ADCR 寄存器的 TRGE 位置 1、 EXTRG 位置 1 的状态，从 $\overline{\text{ADTRG}}$ 引脚输入低电平。在 $\overline{\text{ADTRG}}$ 的下降沿， ADCR 的 ADST 位置 1，并开始 A/D 转换。其他运行与 A/D 转换启动源无关，均相同。该时序如图 15.6 所示。

ADST 位置 1 时的时序为 A/D 转换器检测出 $\overline{\text{ADTRG}}$ 引脚的下降沿后的 5 个状态。

必须在设定 ADCR 寄存器、 ADSTRGR 寄存器、 ADANSR 寄存器后，对 $\overline{\text{ADTRG}}$ 引脚输入低电平。

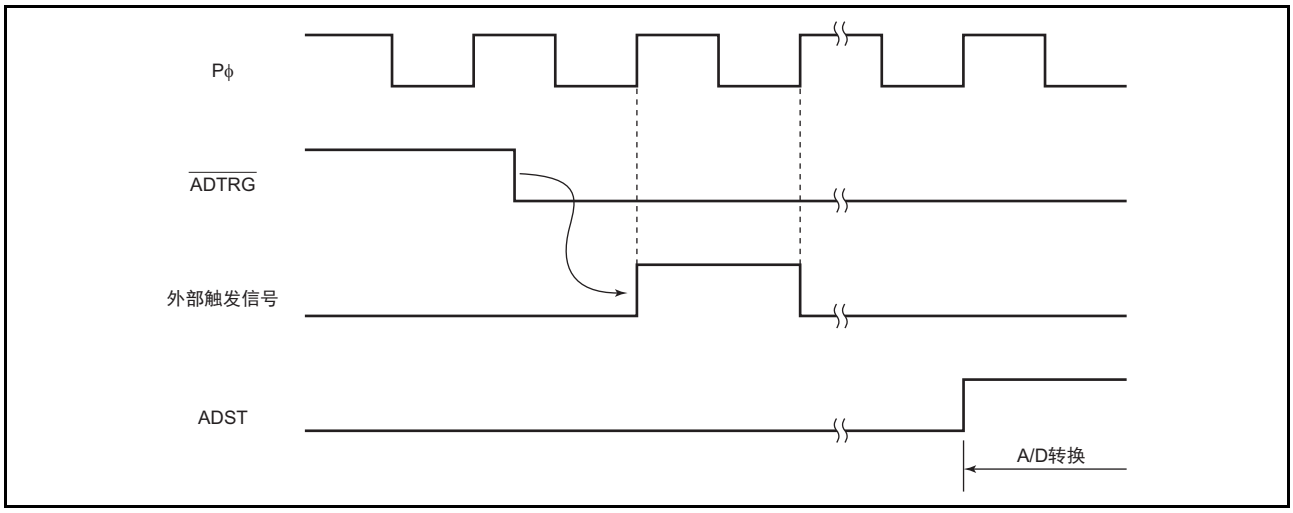


图 15.6　外部触发输入时序

15.4.6　ADDR 寄存器自动清除功能的使用例子

通过将 ADCR 寄存器的 ACE 位置 1，由 CPU 及 DTC 读取 A/D 数据寄存器 （ADDR）时，可自动将 ADDR 寄存器清除为 H'0000。使用此功能，可检测 ADDR 寄存器的未更新故障。

ADDR 寄存器自动清除功能无效 （通常状态） / 有效时的例子如图 15.7 所示。

ACE 位为 0 （初始值）时，如果 A/D 转换结果 （H'0222）因某种原因未写入 ADDR 寄存器，则 ADDR 寄存器的值仍为原来的数据 （H'0111）。而且，使用 A/D 转换结束中断，向通用寄存器等读取该 ADDR 寄存器的值时，将原来的数据 （H'0111）保存至通用寄存器。执行未更新的检查时，需将原来的数据逐一保存至 RAM 及通用寄存器并检查。

ACE 位为 1 时，由 CPU/DTC 读取 ADDR=H'0111， ADDR 寄存器自动清除为 H'0000。此后， A/D 转换结果即 H'0222 因某种原因未传送至 ADDR 寄存器时，清除的数据 （H'0000）作为 ADDR 寄存器的值保留。在此，使用 A/D 转换结束中断，向通用寄存器等读取该 ADDR 寄存器的值时，将 H'0000 保存至通用寄存器。仅检查读取数据的值是否为 H'0000，即可判断有无 ADDR 寄存器的未更新故障。

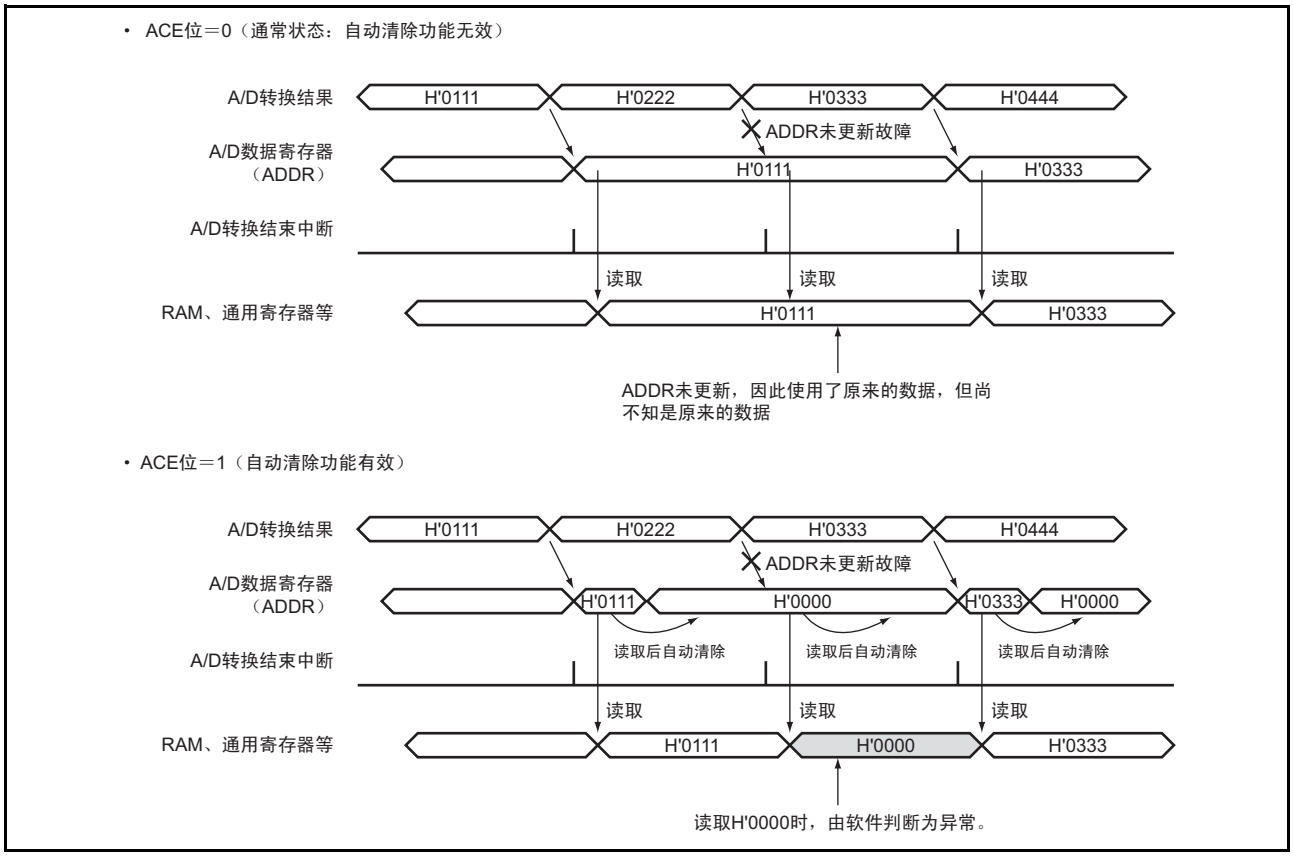


图 15.7　ADDR 寄存器自动清除功能无效 （通常状态） / 有效时的例子

15.5　中断源与 DTC 传送请求

A/D 转换器可产生 A/D 转换结束中断 （ADI_3、 ADI_4）。 ADCR_0 寄存器的 ADIE 位置 1，允许产生 ADI_3 中断； ADCR_1 寄存器的 ADIE 位置 1，允许产生 ADI_4 中断。另一方面， ADCR_0 寄存器的 ADIE 位清 0，禁止产生 ADI_3 中断； ADCR_1 寄存器的 ADIE 位清 0，禁止产生 ADI_4 中断。根据 DTC 的设定，可在产生 ADI_3 中断或 ADI_4 中断时启动 DTC。由 ADI_3 中断或 ADI_4 中断启动 DTC 且由 DTC 传送数据时，ADSR_0、 ADSR_1 的 ADF 位自动清除。

15.6　A/D 转换精度的定义

本 LSI 的 A/D 转换精度的定义如下：

- 分辨率
  A/D转换器的数字转换输出代码数。
- 偏移误差
  数字输出值从最小值 （零电压） B'000000000000变为B'000000000001时的实际A/D转换特性与理想A/D转换特性的偏差，但不含量化误差 （图15.8）。
- 满刻度误差
  数字输出值从B'111111111110变为最大值 （满刻度电压） B'111111111111时的实际A/D转换特性与理想A/D转换特性的偏差，但不含量化误差 （图15.8）。
- 量化误差
  A/D转换器固有的误差，误差的值为1/2LSB （图15.8）。
- 非线性误差
  零电压到满刻度电压之间的实际A/D转换特性与理想A/D转换特性的误差，但不含偏移误差、满刻度误差及量化误差 （图15.8）。
- 绝对精度
  数字值与模拟输入值的偏差，包含偏移误差、满刻度误差、量化误差及非线性误差。

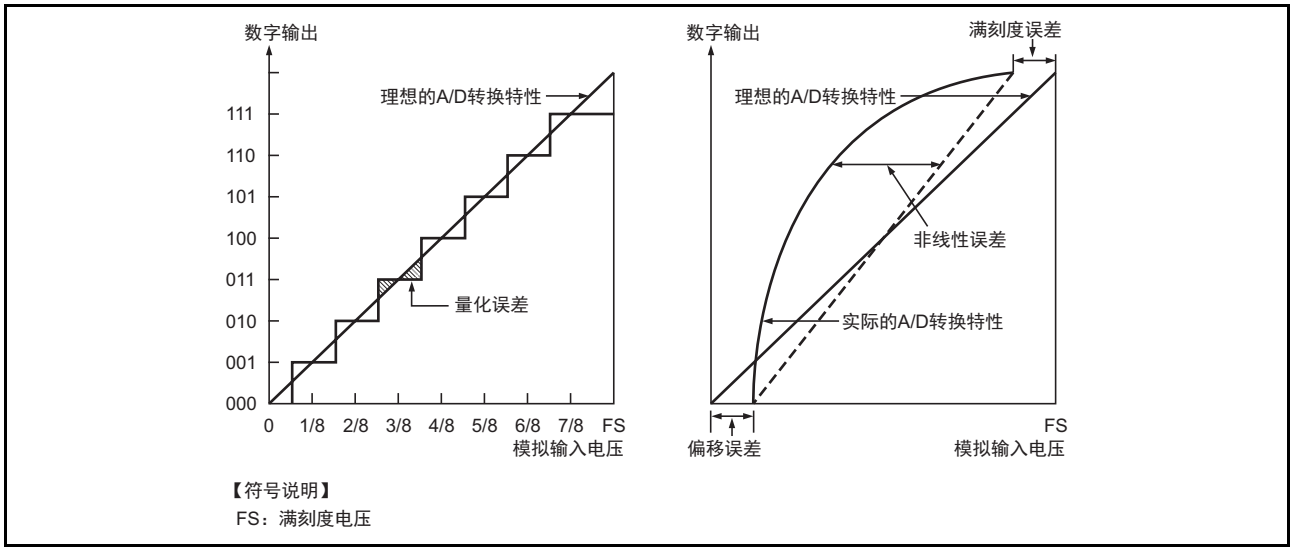


图 15.8　A/D 转换精度的定义

15.7　使用时的注意事项

15.7.1　模拟输入电压的设定范围

A/D 转换过程中，对模拟输入引脚 （ANn）外加的电压必须在 AV_{refl} ≤ ANn （n=0 ～ 15）≤ AV_{refh} 范围内。

15.7.2　AV_{CC}、AV_{SS} 与 V_{CC}、V_{SS} 的关系

使用 A/D 转换器时，必须设定 AV_{CC}=5.0V ± 0.5V、 AV_{SS}=V_{SS}；不使用 A/D 转换器时，必须设定 AV_{SS}=V_{SS}， AV_{CC} 引脚不得设置为开路。

15.7.3　AV_{refh}、AV_{refl} 引脚的设定范围

使用 A/D 转换器时，必须设定 AV_{refh}=4.5 ～ AV_{CC}；不使用 A/D 转换器时，必须设定 AV_{refh} ≤ AV_{CC}。不遵守以上规定时，会对 LSI 的可靠性造成不良影响。对于 AV_{refl}，必须设定 AV_{refl}=AV_{SS}=V_{SS}。

15.7.4　电路板设计时的注意事项

设计电路板时，必须尽量分离数字电路与模拟电路后再布线，布线时必须避免数字电路的信号配线与模拟电路的信号配线交叉及接近，信号线间的感应等会导致模拟电路的误动作，并影响 A/D 转换值。

模拟输入信号 （AN0 ～ AN15）、模拟基准电压 （AV_{refh}、 AV_{refl}）、模拟电源 （AV_{CC}）及模拟接地（AV_{SS}）必须与数字电路分离，而且 AV_{SS} 必须单点连接至电路板上稳定的数字接地 （V_{SS}）。

15.7.5　噪声对策的注意事项

为了防止过大电涌等异常电压破坏模拟输入引脚 （AN0 ～ AN15）及模拟基准电压 （AV_{refh}、 AV_{refl}），如图 15.9 所示，必须在 AV_{CC} – AV_{SS} 之间连接保护电路。与 AV_{refh}、 AV_{refl} 连接的旁路电容及与 ANn 连接的滤波器电容必须连接至 AV_{SS}。如图 15.9 所示，连接滤波器的电容时，平均模拟输入引脚 （ANn）的输入电流，因此可能会产生误差。所以必须充分研讨后再确定电路常数。

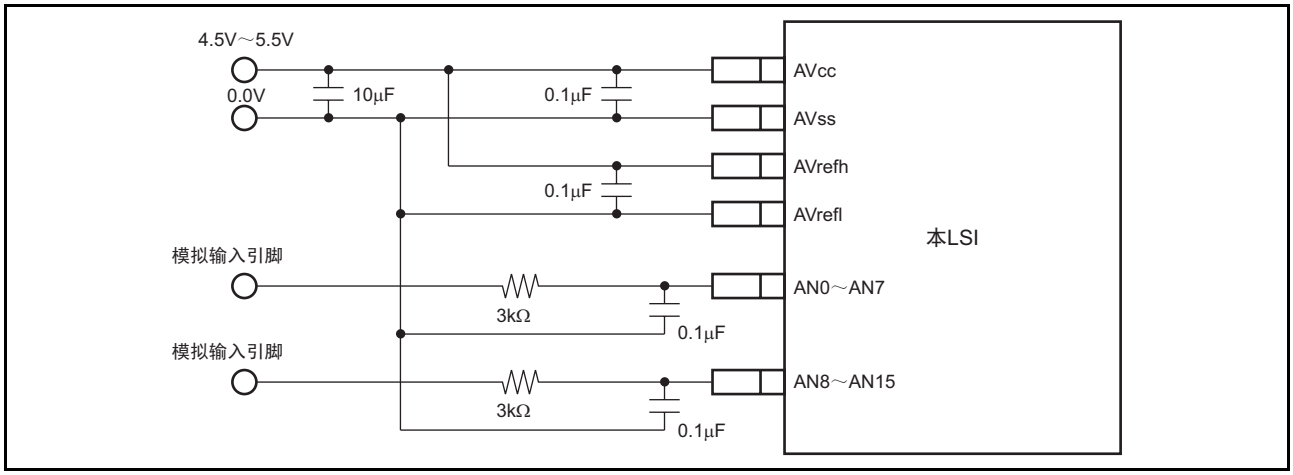


图 15.9　模拟输入引脚的保护电路例子

15.7.6　寄存器设定时的注意事项

- 完成A/D开始触发选择寄存器 （ADSTRGR）、 A/D模拟输入通道选择寄存器 （ADANSR）的设定后，A/D控制寄存器 （ADCR）的ADST位置位。在ADCR的ADST位为1的状态，不得更改ADCR的ADCS位、 ACE位、 ADIE位、 TRGE位、及EXTRG位的设定。
- 不得在A/D模拟输入通道选择寄存器 （ADANSR）的ANS7～ANS0位均为0时开始A/D转换。

第 16 章　比较匹配定时器 （CMT）

本 LSI 内置由 2 个通道的 16 位定时器构成的比较匹配定时器 （CMT）。 CMT 有 16 位计数器，在每个设定的周期可产生中断。

16.1　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）
- 比较匹配时可请求中断
- 可设定模块待机模式

CMT 框图如图 16.1 所示。

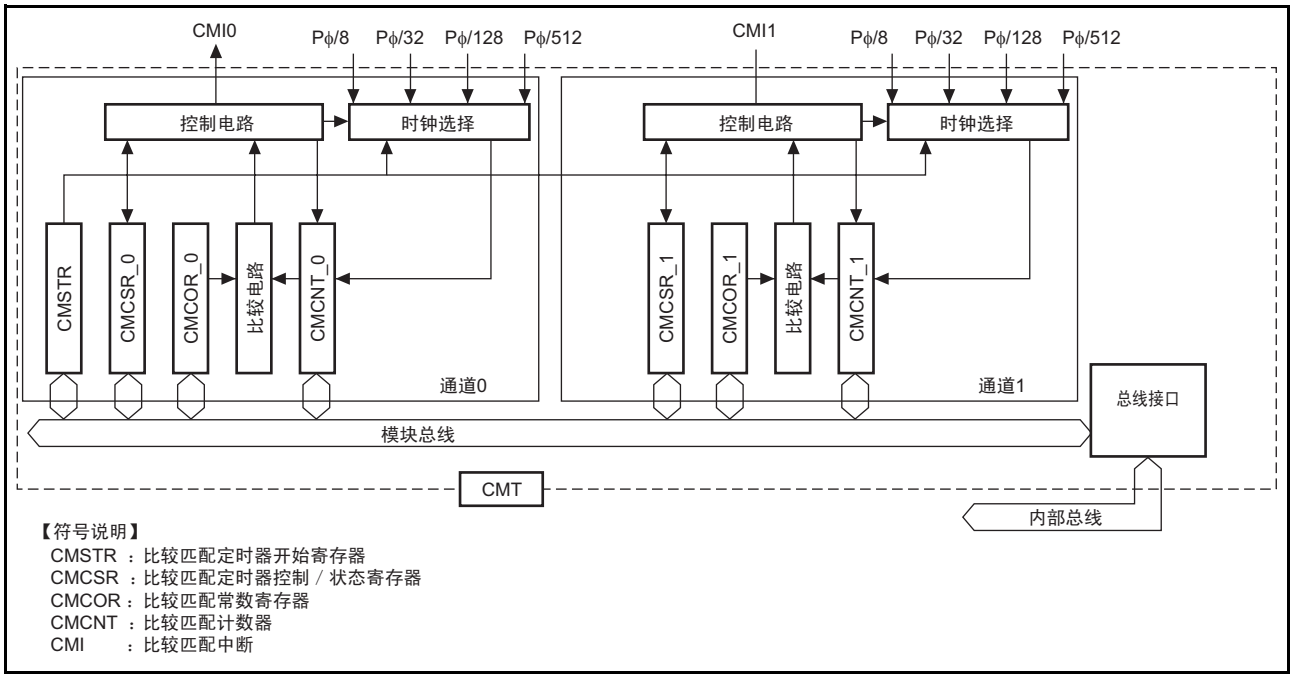


图 16.1　CMT 框图

16.2　寄存器说明

CMT 有以下寄存器。有关寄存器的地址及各处理状态的寄存器状态，详情参阅 “第 24 章 寄存器一览表”。本章省略了通道数。

表 16.1　寄存器结构

寄存器名称	简称	R/W	初始值	地址	存取长度
比较匹配定时器开始寄存器	CMSTR	R/W	H'0000	H'FFFFCE00	8、16、32
比较匹配定时器控制 / 状态寄存器 _0	CMCSR_0	R/W	H'0000	H'FFFFCE02	8、16
比较匹配计数器 _0	CMCNT_0	R/W	H'0000	H'FFFFCE04	8、16、32
比较匹配常数寄存器 _0	CMCOR_0	R/W	H'FFFF	H'FFFFCE06	8、16
比较匹配定时器控制 / 状态寄存器 _1	CMCSR_1	R/W	H'0000	H'FFFFCE08	8、16、32
比较匹配计数器 _1	CMCNT_1	R/W	H'0000	H'FFFFCE0A	8、16
比较匹配常数寄存器 _1	CMCOR_1	R/W	H'FFFF	H'FFFFCE0C	8、16、32

16.2.1　比较匹配定时器开始寄存器 （CMSTR）

CMSTR 为 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	位名称	初始值	R/W	说　　明
15 ～ 2	—	均为 0	R	保留位 读取值、写入值总是为 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

16.2.2　比较匹配定时器控制 / 状态寄存器 （CMCSR）

CMCSR 为 16 位寄存器，表示是否产生比较匹配，并设定中断及计数器输入时钟。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	CMF	CMIE	-	-	-	-	CKS[1:0]	
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R	R	R	R	R	R	R	R	(R/W)*1	R/W	R	R	R	R	R/W	R/W

【注】 *1　为了清除标志，仅可在读取1后写入0。

位	位名称	初始值	R/W	说　　明
15 ～ 8	—	均为 0	R	保留位 读取值、写入值总是为 0。
7	CMF	0	(R/W)*1	比较匹配标志 该标志表示 CMCNT 与 CMCOR 的值是否匹配。 0：CMCNT 与 CMCOR 的值不匹配 ［清除条件］ • 对 CMF 写入 0 时 *2 • 由 CMI 中断启动 DTC， DTC 的 MRB 的 DISEL 位为 0，可存取 CMT 寄存器时 ［置位条件］ 1：CMCNT 与 CMCOR 的值匹配
6	CMIE	0	R/W	比较匹配中断允许 CMCNT 和 CMCOR 的值匹配时 （CMF=1），选择允许或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 读取值、写入值总是为 0。
1、 0	CKS[1:0]	00	R/W	时钟选择 1、 0 从分频外围运行时钟 （Pϕ）的 4 种内部时钟，选择输入 CMCNT 的时钟。CMSTR 的 STR 位置 1 时， CMCNT 用 CKS1、CKS0 位选择的时钟开始计数。 00：Pϕ/8 01：Pϕ/32 10：Pϕ/128 11：Pϕ/512

【注】 *1　为了清除标志，仅可在读取 1 后写入 0。

*2　读取 1 后写入 0 前，产生由下一个比较匹配引起的标志置位时，即使写入 0 也不清除标志，因此，必须再次读取 1 后写入 0。

16.2.3　比较匹配计数器 （CMCNT）

CMCNT 为 16 位寄存器，用作递增计数器。由 CMCSR 的 CKS1、CKS0 位选择计数器输入时钟。 CMSTR 的 STR 位置 1 时， CMCNT 用所选的时钟开始计数。 CMCNT 的值与比较匹配常数寄存器 （CMCOR）的值匹配时， CMCNT 清除为 H'0000， CMCSR 的 CMF 标志置 1。

初始值为 H'0000。

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

16.2.4　比较匹配常数寄存器 （CMCOR）

CMCOR 为 16 位寄存器，设定与 CMCNT 比较匹配前的时间。

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

16.3　运行说明

16.3.1　周期计数运行

由 CMCSR 的 CKS1、CKS0 位选择内部时钟。CMSTR 的 STR 位置 1 时，CMCNT 用所选的时钟开始递增计数。 CMCNT 的值与 CMCOR 的值匹配时， CMCNT 清除为 H'0000， CMCSR 的 CMF 标志置 1。此时，如果 CMCSR 寄存器的 CMIE 位设定为 1，则请求比较匹配中断 （CMI）。 CMCNT 从 H'0000 重新开始递增计数。

比较匹配计数器运行如图 16.2 所示。

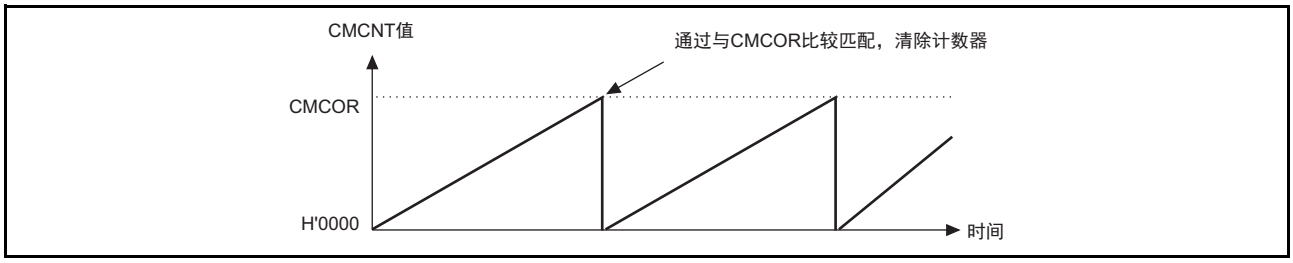


图 16.2　计数器运行

16.3.2　CMCNT 计数时序

可由 CMCSR 的 CKS1、CKS0 位从分频时钟 （Pϕ）得到的 4 种时钟 （Pϕ/8、Pϕ/32、Pϕ/128、Pϕ/512）中选择 1 个时钟。其时序如图 16.3 所示。

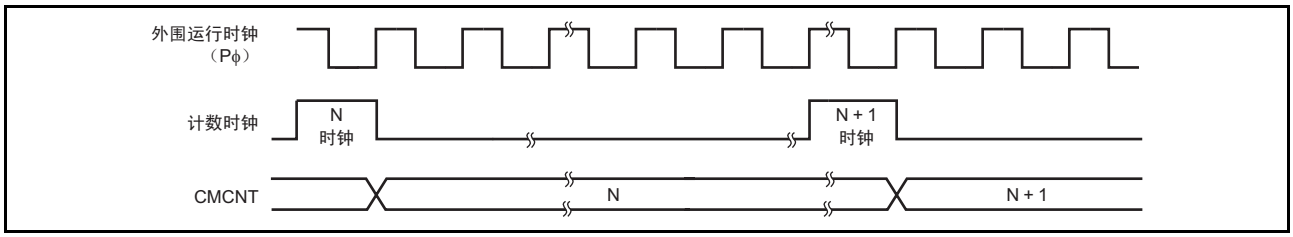


图 16.3　计数时序

16.4　中断

16.4.1　CMT 的中断源和 DTC

CMT 的各通道都有比较匹配中断，分别分配独立的向量地址。中断请求标志 （CMF）置 1 且中断允许位（CMIE）也置 1 时，输出相应中断请求。由中断请求启动 CPU 中断时，可通过设定中断控制器更改通道间的优先顺序。详情参阅 “第 6 章 中断控制器 （INTC）”。

另外，也可将中断请求用作数据传送控制器 （DTC）的启动源。此时，通道间的优先顺序固定。详情参阅 “第 8 章 数据传送控制器 （DTC）”。

16.4.2　比较匹配标志的置位时序

CMCOR 与 CMCNT 匹配时，产生比较匹配信号， CMCSR 的 CMF 位置 1。在匹配的最终状态 （CMCNT 的值更新为 H'0000 时）产生比较匹配信号。即， CMCOR 与 CMCNT 匹配后，如果未输入用于 CMCNT 计数器的时钟，则不产生比较匹配信号。 CMF 位的置位时序如图 16.4 所示。

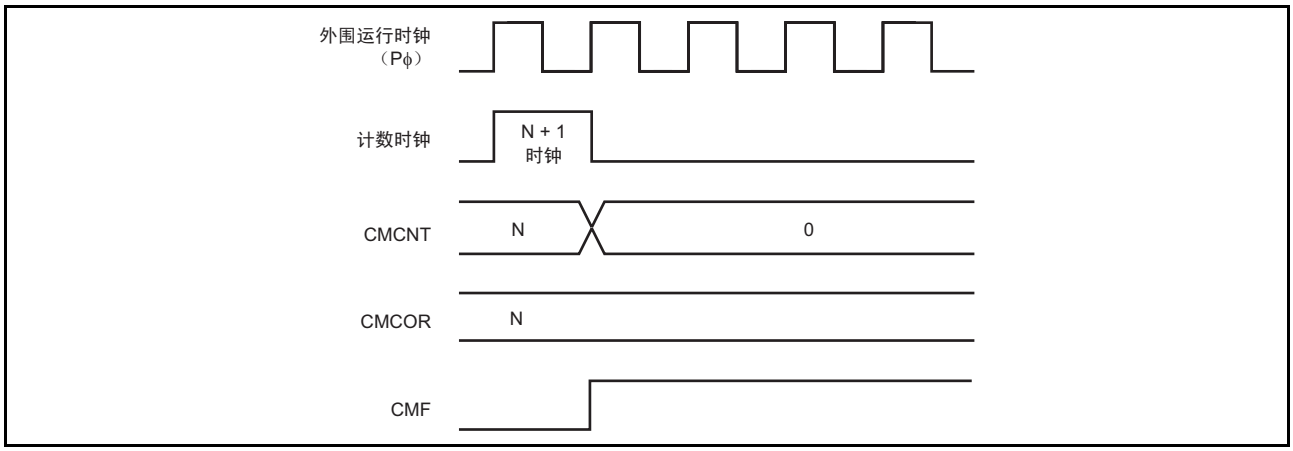


图 16.4　CMF 置位时序

16.4.3　比较匹配标志的清除时序

读取 CMF=1 后写入 0，即可清除 CMCSR 的 CMF 位。

16.5　使用时的注意事项

16.5.1　模块待机模式的设定

CMT 可通过待机控制寄存器设定禁止 / 允许本模块运行。为初始值时， CMT 停止运行。解除模块待机模式，即可存取寄存器。详情参阅 “第 22 章 低功耗模式”。

16.5.2　CMCNT 的写入与比较匹配的竞争

在 CMCNT 计数器写入周期的 T2 状态产生比较匹配信号时，不写入 CMCNT 计数器，而优先清除 CMCNT 计数器，该时序如图 16.5 所示。

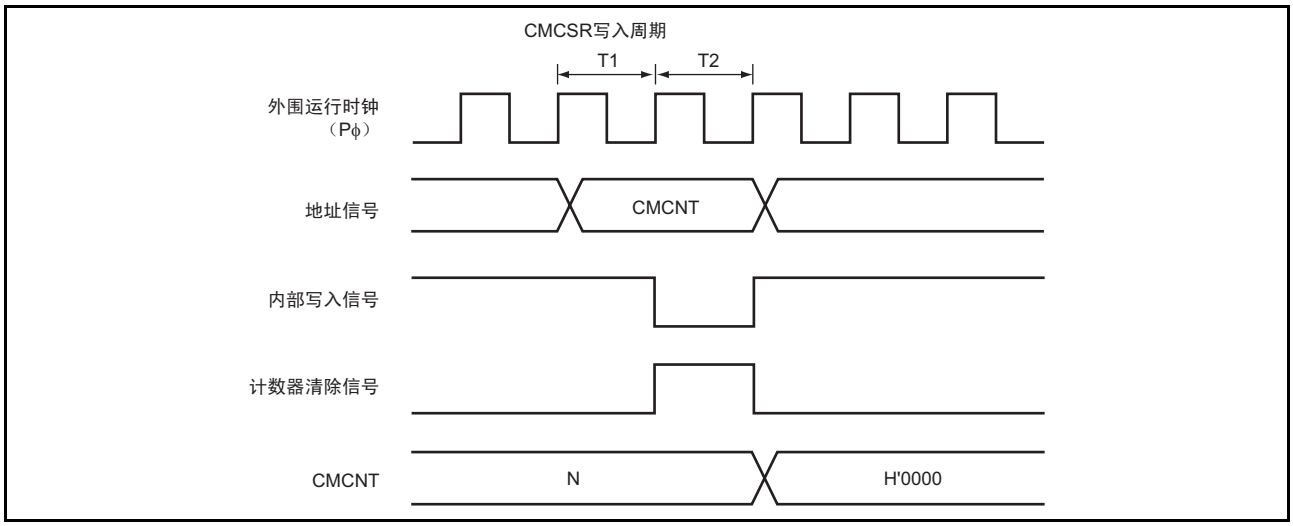


图 16.5　CMCNT 的写入与比较匹配的竞争

16.5.3　CMCNT 的字写入与递增计数的竞争

在 CMCNT 计数器的字写入周期的 T2 状态，即使产生递增计数，也不递增计数，而优先写入计数器，该时序如图 16.6 所示。

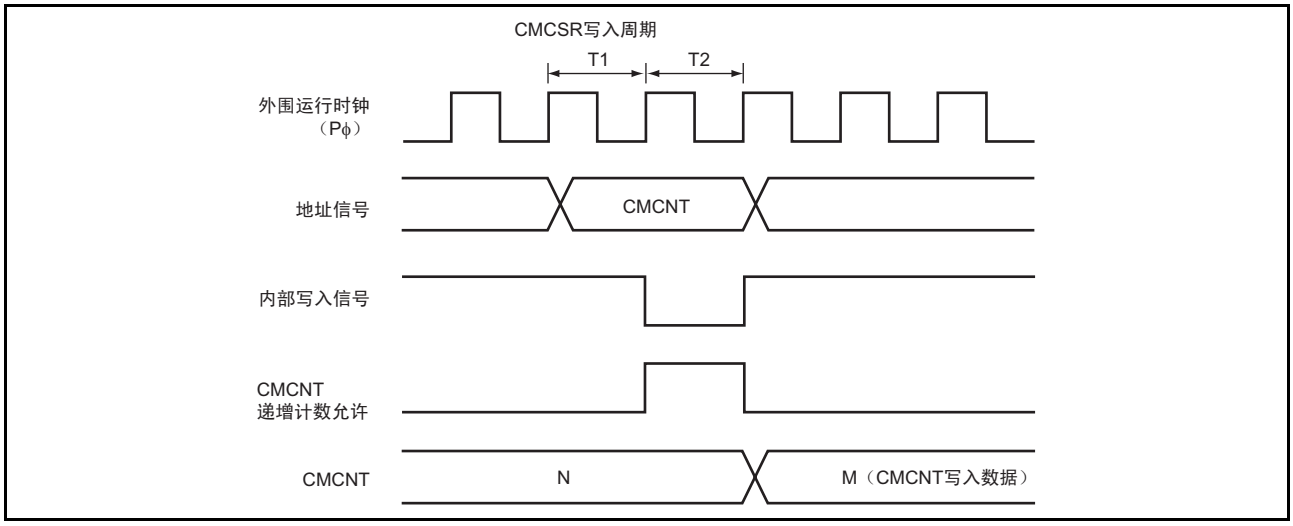


图 16.6　CMCNT 的字写入与递增计数的竞争

16.5.4　CMCNT 的字节写入与递增计数的竞争

在 CMCNT 的字节写入周期的 T2 状态，即使产生递增计数，已写入的数据也不递增计数，而优先写入计数器。未写入的字节数据也不递增计数，仍为写入前的内容。

在 CMCNTH 的写入周期的 T2 状态产生递增计数的时序如图 16.7 所示。

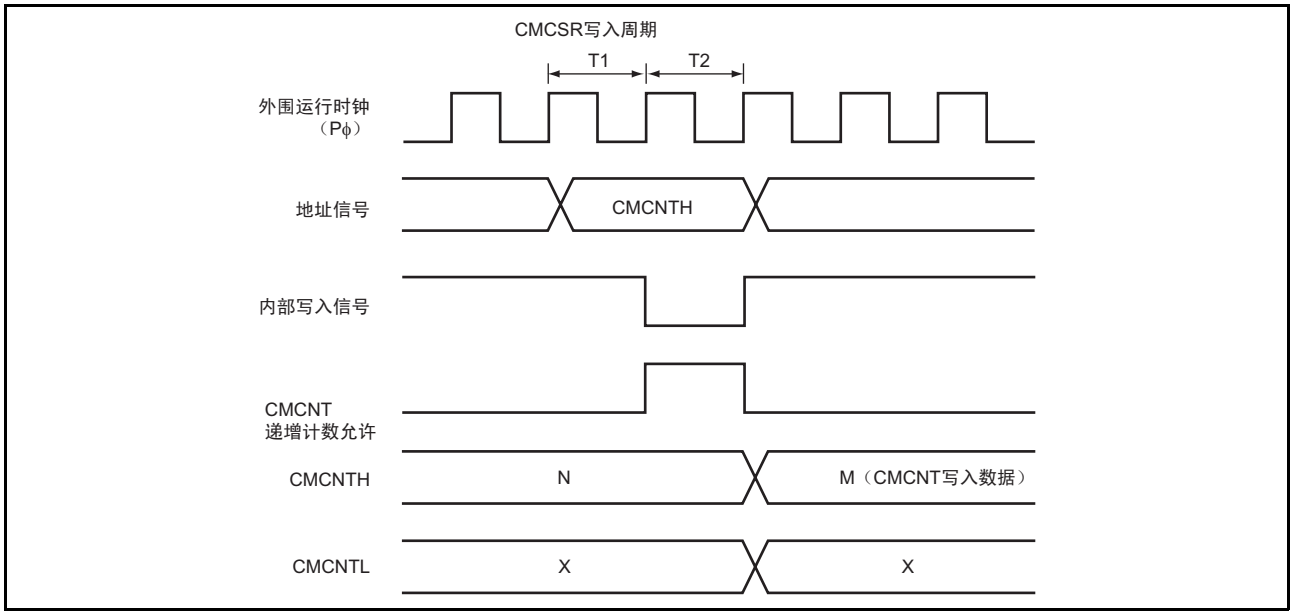


图 16.7　CMCNT 的字节写入与递增计数的竞争

16.5.5　CMCNT 与 CMCOR 的比较匹配

在 CMCNT 计数运行停止状态， CMCNT 和 CMCOR 不得设定为相同值。

在 CMCNT 计数运行停止状态， CMCNT 和 CMCOR 设定为相同值时， CMCSR 的 CMF 位置 1， CMCNT 清除为 H'0000。

第 17 章 控制器局域网 （RCAN-ET）

控制器局域网 （RCAN-ET）是控制以汽车及工业设备系统等的实时通信为目标的 CAN （Controller Area Network）的模块。

本章说明 RCAN-ET 的编程接口的相关内容。

CAN 数据链路控制器的功能参阅以下参考文献：

［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-CD-11898-1, 2002）

17.1 特点

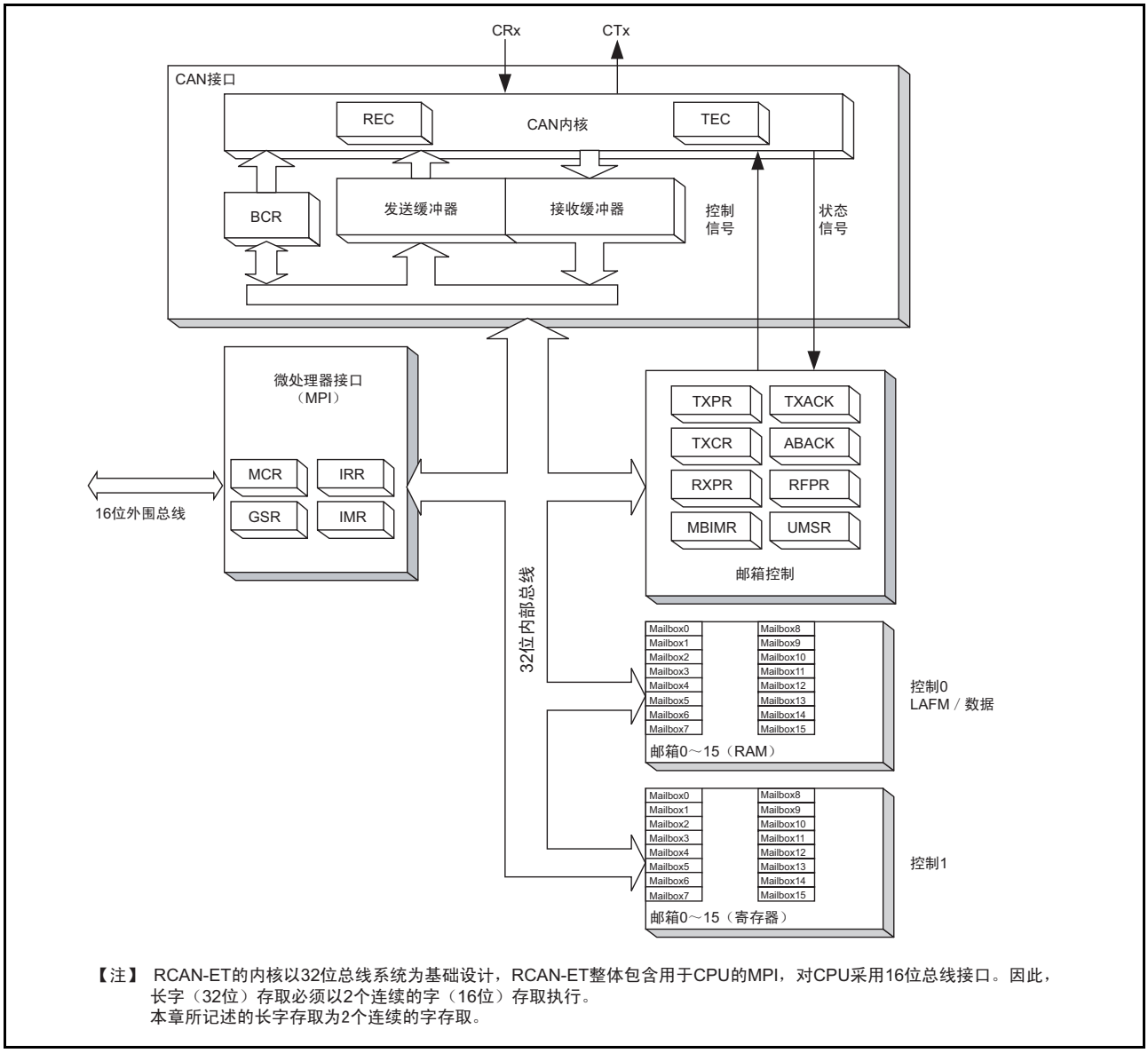
- 支持CAN2.0B标准
- 位时序依据ISO-11898标准
- 16个邮箱
- 时钟频率：16～40 MHz
- 15个可编程的发送/接收邮箱及1个接收邮箱
- CAN低功耗睡眠模式及检测到CAN总线激活后自动解除CAN睡眠模式
- 支持所有邮箱的可编程接收滤波器屏蔽 （标准ID及扩展ID）
- 最大1Mbps的可编程CAN数据速率
- 为了防止实时应用程序引起的优先顺序颠倒，具备可在内部区分优先顺序的传送信息队列
- 丰富的中断源
- 内置测试功能 （只听模式、错误被动模式）

17.2　结构

17.2.1　　框图

RCAN-ET 提供自由且精炼的方法，用于构成支持 CAN2.0B Active 及 ISO-11898 的 CAN 帧，从功能上讲， RCAN-ET 由微处理器接口 （MPI）、邮箱、邮箱控制及 CAN 接口等 4 种功能块构成。

RCAN-ET 框图如图 17.1 所示。



【注】 RCAN-ET的内核以32位总线系统为基础设计，RCAN-ET整体包含用于CPU的MPI，对CPU采用16位总线接口。因此，长字（32位）存取必须以2个连续的字（16位）存取执行。
本章所记述的长字存取为2个连续的字存取。

图 17.1　RCAN-ET 框图

17.2.2　各块的功能

(1)　微处理器接口 （MPI）

MPI 可执行 CPU 与 RCAN-ET 的寄存器与邮箱之间的通信，并控制存储器接口。 MPI 检测 CAN 总线激活，并具有将 CAN 总线激活情况通知给 MPI 及 RCAN-ET 的其他模块的唤醒控制逻辑，因此 RCAN-ET 可自动解除 CAN 睡眠模式。 MPI 的寄存器有 MCR、 IRR、 GSR 及 IMR。

(2)　邮箱

邮箱可作为信息缓冲器排列在 RAM 及寄存器。
RAM 及寄存器中分别有 16 个邮箱，保存以下信息：

［RAM］
- CAN信息控制 （ID、 RTR、 IDE等）
- CAN信息数据 （用于CAN数据帧）
- 用于接收的局部接收滤波器屏蔽 （LAFM）

［寄存器］
- CAN信息控制 （DLC）
- 3位邮箱结构、自动重新发送无效位、用于远程请求的自动发送位及新信息控制位

(3)　邮箱控制

邮箱控制有以下功能：
- 接收信息时，比较ID，生成将来自CAN接口的信息保存至邮箱的RAM地址与数据，并置位/清除相应的寄存器。
- 发送信息时， RCAN-ET执行内部仲裁，选择优先顺序正确的信息，将信息从邮箱加载至CAN接口的发送缓冲器。此后，置位/清除相应的寄存器。
- 执行CPU与邮箱控制之间的邮箱存取的仲裁。
- 寄存器有TXPR、 TXCR、 TXACK、 ABACK、 RXPR、 RFPR、 MBIMR及UMSR。

(4)　CAN 接口

本块支持参考文献 [2]、 [4] 的 CAN 总线数据链路控制器规格，满足 OSI 模型所规定的数据链路控制器的所有功能，还对 CAN 总线提供特殊的寄存器及逻辑。具体有接收错误计数器、发送错误计数器、位结构寄存器及各种测试模式等，还有保存 CAN 数据链路控制器发送 / 接收的功能。

17.2.3　引脚结构

RCAN-ET 的引脚结构如表 17.1 所示。

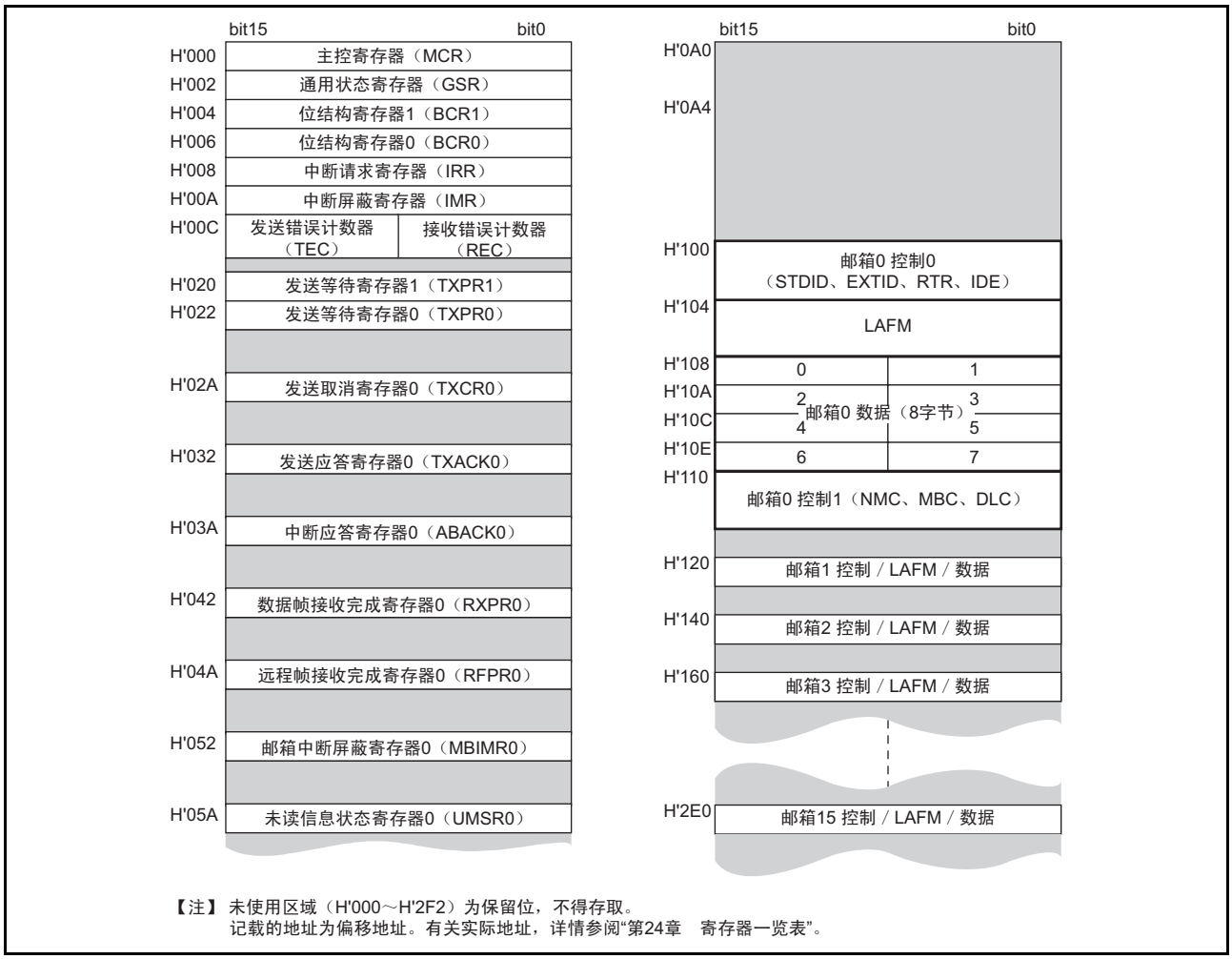
表 17.1　引脚结构

通道	名称	引脚名称	输入 / 输出	功　能
0	发送数据引脚	CTx0	输出	CAN 总线发送引脚。
	接收数据引脚	CRx0	输入	CAN 总线接收引脚。
1*	发送数据引脚	CTx1	输出	CAN 总线发送引脚。
	接收数据引脚	CRx1	输入	CAN 总线接收引脚。

【注】 *　仅适用于 SH7142

17.2.4　存储器映射

RCAN-ET 的存储器映射如图 17.2 所示。



【注】 未使用区域（H'000～H'2F2）为保留位，不得存取。
记载的地址为偏移地址。有关实际地址，详情参阅“第24章　寄存器一览表”。

图 17.2　RCAN-ET 的存储器映射

17.3　邮箱

17.3.1　邮箱结构

邮箱可作为发送 / 接收 CAN 帧的信息缓冲器运行。各邮箱由信息控制、局部接收滤波器屏蔽 （LAFM）及信息数据等 3 个存储字段构成。

各邮箱的信息控制、 LAFM、信息数据的地址映射如表 17.2 所示。

表 17.2　各邮箱的地址映射

邮箱	地址			
	控制 0	LAFM	数据	控制 1
	4 字节	4 字节	8 字节	2 字节
0 （仅接收）	H'100 － H'103	H'104 － H'107	H'108 － H'10F	H'110 － H'111
1	H'120 － H'123	H'124 － H'127	H'128 － H'12F	H'130 － H'131
2	H'140 － H'143	H'144 － H'147	H'148 － H'14F	H'150 － H'151
3	H'160 － H'163	H'164 － H'167	H'168 － H'16F	H'170 － H'171
4	H'180 － H'183	H'184 － H'187	H'188 － H'18F	H'190 － H'191
5	H'1A0 － H'1A3	H'1A4 － H'1A7	H'1A8 － H'1AF	H'1B0 － H'1B1
6	H'1C0 － H'1C3	H'1C4 － H'1C7	H'1C8 － H'1CF	H'1D0 － H'1D1
7	H'1E0 － H'1E3	H'1E4 － H'1E7	H'1E8 － H'1EF	H'1F0 － H'1F1
8	H'200 － H'203	H'204 － H'207	H'208 － H'20F	H'210 － H'211
9	H'220 － H'223	H'224 － H'227	H'228 － H'22F	H'230 － H'231
10	H'240 － H'243	H'244 － H'247	H'248 － H'24F	H'250 － H'251
11	H'260 － H'263	H'264 － H'267	H'268 － H'26F	H'270 － H'271
12	H'280 － H'283	H'284 － H'287	H'288 － H'28F	H'290 － H'291
13	H'2A0 － H'2A3	H'2A4 － H'2A7	H'2A8 － H'2AF	H'2B0 － H'2B1
14	H'2C0 － H'2C3	H'2C4 － H'2C7	H'2C8 － H'2CF	H'2D0 － H'2D1
15	H'2E0 － H'2E3	H'2E4 － H'2E7	H'2E8 － H'2EF	H'2F0 － H'2F1

邮箱 0 为接收专用邮箱，邮箱 1 ～ 15 可通过设定信息控制的 MBC （邮箱结构）位执行发送及接收。邮箱结构的详情如图 17.3 所示。

• 邮箱0（用于接收的邮箱）

寄存器名称	地址	数据总线																存取长度	字段名称
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
MB[0].CONTROL0H	H'100	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		16/32位	控制0
MB[0].CONTROL0L	H'102	EXTID[15:0]																16位	
MB[0].LAFMH	H'104	IDE LAFM	0	0	STDID_LAFM[10:0]											EXTID LAFM[17:16]		16/32位	LAFM
MB[0].LAFML	H'106	EXTID_LAFM[15:0]																16位	
MB[0].MSG_DATA[0][1]	H'108	MSG_DATA_0 （第一个Rx 字节）								MSG_DATA_1								8/16/32位	数据
MB[0].MSG_DATA[2][3]	H'10A	MSG_DATA_2								MSG_DATA_3								8/16位	
MB[0].MSG_DATA[4][5]	H'10C	MSG_DATA_4								MSG_DATA_5								8/16/32位	
MB[0].MSG_DATA[6][7]	H'10E	MSG_DATA_6								MSG_DATA_7								8/16位	
MB[0].CONTROL1H、L	H'110	0	0	NMC	0	0	MBC[2:0]			0	0	0	0	DLC[3:0]				8/16位	控制1

• 邮箱1～15（用于发送/接收的邮箱）

寄存器名称	地址	数据总线																存取长度	字段名称
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
MB[n].CONTROL0H	H'100+n*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		16/32位	控制0
MB[n].CONTROL0L	H'102+n*32	EXTID[15:0]																16位	
MB[n].LAFMH	H'104+n*32	IDE LAFM	0	0	STDID_LAFM[10:0]											EXTID LAFM[17:16]		16/32位	LAFM
MB[n].LAFML	H'106+n*32	EXTID_LAFM[15:0]																16位	
MB[n].MSG_DATA[0][1]	H'108+n*32	MSG_DATA_0 （第一个Rx/Tx 字节）								MSG_DATA_1								8/16/32位	数据
MB[n].MSG_DATA[2][3]	H'10A+n*32	MSG_DATA_2								MSG_DATA_3								8/16位	
MB[n].MSG_DATA[4][5]	H'10C+n*32	MSG_DATA_4								MSG_DATA_5								8/16/32位	
MB[n].MSG_DATA[6][7]	H'10E+n*32	MSG_DATA_6								MSG_DATA_7								8/16位	
MB[n].CONTROL1H、L	H'110+n*32	0	0	NMC	ATX	DART	MBC[2:0]			0	0	0	0	DLC[3:0]				8/16位	控制1

【注】 1. 用灰色表示的位为保留位。写入值必须为0。读取值不一定为0，且不能保证。
2. 邮箱0的MBC1位的值总为1。
3. 邮箱0不支持ATX和DART，且邮箱0的MBC的设定值受限制。
4. MCR15位为1时，信息控制和LAFM的STDID、RTR、IDE、EXTID的顺序与HCAN2的顺序不同。
5. n：0～15（邮箱编号）

图 17.3　邮箱 N 的结构

17.3.2　信息控制字段

寄存器名称	地址	位	位名称	说　明
MB[n]. CONTROL0H	H'100+n*32	15	IDE	ID 扩展 区分 CAN 数据帧与远程帧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。通过数据帧或远程帧，接收 CAN 帧可改写此位。 【重点】MBC=B'001 时，如果数据帧自动发送 （ATX）位置位，则 RTR 不可置位。如果接收远程帧，则可通过对应的 RFPR 位或 IRR2 （远程帧请求中断）位通知 CPU，但 RCAN-ET 需将当前的信息作为数据帧发送，因此 RTR 位无变化。 【重点】MBC=B'001 且 ATX=1 时，为了自动应答远程帧，必须将 RTR 标志清 0 以便允许传送数据帧。 【注意】将邮箱设定为发送远程帧时，用于传送的 DLC 为保存在邮箱的内容。 0：数据帧 1：远程帧
		13	—	保留位 写入值总是为 0，不保证读取值。
		12 ～ 2	STDID [10:0]	标准 ID 本位可设定数据帧与远程帧的 ID （标准 ID）。
		1、0	EXTID [17:16]	扩展 ID 本位可设定数据帧与远程帧的 ID （扩展 ID）。
MB[n]. CONTROL0L	H'102+n*32	15 ～ 0	EXTID [15:0]	

【注】 n：0 ～ 15 （邮箱编号）

- 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/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

【注】　MBC1的值总是为1。

- 邮箱1～15

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	位名称	说　明
MB[n]. CONTROL1H	H'110+n*32	15、 14	—	保留位 写入值总是为 0，不保证读取值。
		13	NMC	新信息控制 如果此位置 0，则 RXPR 或 RFPR 已被置位的邮箱不保存新信息，而保存之前的信息，并置位 UMSR 的对应位；如果此位置 1，则 RXPR 或 RFPR 已被置位的邮箱重写新信息，并置位 UMSR 的对应位。 【重要】以数据帧改写远程帧，或执行与之相反的运行时， RXPR 及 RFPR 标志可能在 （与 USMR 一起）相同邮箱置位。此时，也重写存在于邮箱控制字段的 RTR 位。 0：溢出模式 1：重写模式
		12	ATX	自动发送数据帧 如果本位置 1 且在邮箱接收远程帧，则保存 DLC。通过置位 TXPR，使用当前信息数据及改写后的 DLC，可从相同邮箱自动传送数据帧。如信息发送优先位 （MCR2）构成的那样，根据 ID 优先顺序或邮箱优先顺序控制发送时序。使用本功能时，MBC[2:0] 必须设定为 B'001。使用此功能执行发送时，所使用的数据长度代码 （DLC）使用接收的 DLC。应用时，远程帧的 DLC 需对应所请求数据帧的 DLC。 【重要】使用 ATX 且 MBC 为 B'001 时，远程帧的 ID 与响应信息时相同，必须与数据帧的 ID 完全相同，因此不可使用 IDE 位的滤波器。 【重要】使用本功能时，与接收远程帧无关， RTR 位不置位。如果接收 远程帧，则可通过已置位的 RFPR 通知 CPU，但 RCAN-ET 需将当前信息作为数据帧发送，不可更改 RTR 位。 【重要】在溢出状态 （设定为 NMC=0 的邮箱的 UMSR 置位），放弃接收信息。设定为 ATX=1 的邮箱通过远程帧进入溢出状态时，有时会接受旧信息的自动发送请求。 0：自动发送数据帧无效 1：自动发送数据帧有效 【注】 本位仅存在于邮箱 1 ～ 15，在邮箱 0 为保留位。
		11	DART	自动重新发送无效 如果此位置位，则在产生 CAN 总线错误事件或 CAN 总线仲裁失败时，信息的自动重新发送无效。实际使用本功能时，在发送最初，对应的 TXCR 位自动置位。如果此位置 0，则 RCAN-ET 在发送正常结束前或通过 TXCR 取消前，只发送请求的次数。 0：重新发送有效 1：重新发送无效 【注】 本位仅存在于邮箱 1 ～ 15，在邮箱 0 为保留位。
		10 ～ 8	MBC[2:0]	邮箱结构 如表 17.3 所示，这些位可设定各邮箱的功能。 MBC 为 B'111 时，邮箱无效。即：与 TXPR 或其他设定无关，不发送 / 接收信息。 禁止将 MBC 设定为 B'100、 B'101 及 B'110，如果 MBC 设定为除此之外的值，则可使用 LAFM 字段。 设置 MBC 为接收时，不得设定 TXPR。 无硬件保护功能， TXPR 仍为置位状态。邮箱 0 的 MBC[1] 为接收专用，因此由硬件固定为 1。

寄存器名称	地址	位	位名称	说　明
MB[n]. CONTROL1L	H'111+n*32	7 ～ 4	—	保留位 写入值总是为 0，不保证读取值。
		3 ～ 0	DLC[3:0]	数据长度代码 编码以数据帧发送的数据字节数 （0 ～ 8）。发送远程帧请求时，所使用 DLC 的值必须与请求数据帧的 DLC 的值相同。 0000：数据长度为 0 字节 0001：数据长度为 1 字节 0010：数据长度为 2 字节 0011：数据长度为 3 字节 0100：数据长度为 4 字节 0101：数据长度为 5 字节 0110：数据长度为 6 字节 0111：数据长度为 7 字节 1xxx：数据长度为 8 字节 【注】　x：Don't care

【注】 *　n：0 ～ 15 （邮箱编号）

表 17.3　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	• 禁止使用邮箱 0
0	0	1	可	可	不可	可	• 可用于 ATX* • 禁止使用邮箱 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】 *　为了支持自动重新发送，在 MBC 为 B'001 且 ATX 为 1 时， RTR 必须清 0。设定 ATX 为 1 时，不得使用 IDE 的滤波器。

17.3.3　局部接收滤波器屏蔽 （LAFM）

MBC 为 B'001、 B'010 及 B'011 的值时，该字段可用作接收的 LAFM。 LAFM 允许邮箱接受多个接收 ID。如图 17.4 所示， LAFM 由 2 个 16 位可读取 / 写入的区域构成。

寄存器名称	地址	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	存取长度	字段名称
MB[n].LAFMH	H'104+n*32	IDE LAFM	0	0	STDID_LAFM[10:0]											EXTID LAFM[17:16]		16/32位	LAFM
MB[n].LAFML	H'106+n*32	EXTID_LAFM[15:0]																16位	

【注】n：0～15（邮箱编号）

图 17.4　局部接收滤波器屏蔽 （LAFM）

1 个位在 LAFM 置位时，如果 RCAN-ET 用匹配的 CAN-ID 搜索邮箱，则忽略已接收 CAN 的 ID 对应位。清除位时，已接收 CAN 的 ID 对应位必须与保存的邮箱设定的 STDID/IDE/EXTID 匹配。 LAFM 的结构与邮箱的信息控制相同，不使用此功能时，需填充 0。

【注】 1. RCAN-ET 从邮箱 15 到邮箱 0，开始搜索匹配的 ID。如果 RCAN-ET 检测出匹配的 ID，则立即结束搜索，与 NMC 或 RXPR/RFPR 标志无关，保存该信息。这表示即使使用 LAFM，接收信息也仅保存在 1 个邮箱。
2. 如果接收 1 条信息并发现匹配邮箱，则将信息全部保存至邮箱。使用 LAFM 时， STDID、 RTR、 IDE、 EXTID 更新为已接收信息的 STDID、 RTR、 IDE、 EXTID，因此与接收前的设定不同。

寄存器名称	地址	位	位名称	说　明
MB[n]. LAFMH	H'104+n*32	15	IDE_LAFM	IDE 位的滤波器屏蔽位 0：对应的 IDE 位有效 1：对应的 IDE 位无效
		14、 13	—	保留位 写入值总是为 0，不保证读取值。
		12 ～ 2	STDID_LAFM [10:0]	STDID[10:0] 的滤波器屏蔽位 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] 的滤波器屏蔽位 0：对应的 EXTID 位有效 1：对应的 EXTID 位无效
MB[n]. LAFML	H'106+n*32	15 ～ 0	EXTID_LAFM [15:0]	

【注】 n：0 ～ 15 （邮箱编号）

17.3.4　信息数据字段

保存发送 / 接收的 CAN 信息。 MSG_DATA_0 对应发送 / 接收的最初数据字节， CAN 总线的位排列顺序为 bit7 到 bit0。

17.4　RCAN-ET 的控制寄存器

说明 RCAN-ET 的控制寄存器的相关内容。 RCAN-ET 控制寄存器仅可通过字长度 （16 位）存取。

RCAN-ET 的控制寄存器如表 17.4 所示。

表 17.4　RCAN-ET 的控制寄存器结构

寄存器名称	简称	存取长度 （位）
主控寄存器	MCR	字
通用状态寄存器	GSR	字
位结构寄存器 1	BCR1	字
位结构寄存器 0	BCR0	字
中断请求寄存器	IRR	字
中断屏蔽寄存器	IMR	字
发送错误计数器 / 接收错误计数器	TEC/REC	字

17.4.1　主控寄存器 （MCR）

MCR 为可读取 / 写入的 16 位寄存器，控制 RCAN-ET。

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	ID 排序 本位可设定与 HCAN2 兼容的信息控制及 LAFM 的 STDID、 RTR、 IDE、 EXTID 的顺序。仅可在复位模式更改本位。有关 ID 排序顺序，详情参照图 17.5。 0：RCAN-ET 与 HCAN2 顺序相同 1：RCAN-ET 与 HCAN2 顺序不同
14	MCR14	0	R/W	总线断开自动停机 同时设定本位与 MCR6 时，如果 RCAN-ET 转移至总线断开状态，则 MCR1 立即自动置位。仅可在复位模式更改本位。 0：按照通常的返回顺序 （128×11 隐性位）维持 RCAN-ET 总线断开状态 1：如果 MCR6 置位，则 RCAN-ET 在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 读取值、写入值总是为 0。
10 ～ 8	TST[2:0]	000	R/W	测试模式 本位设置测试模式有效 / 无效。必须注意，启动测试模式前，需将 RCAN-ET 设置为停机模式或复位模式，这是为了避免向测试模式的转移影响正在执行的发送 / 接收。详情参阅 “17.6.2 测试模式的设定”。 测试模式仅用于诊断及测试，因此不可在 RCAN-ET 普通运行时使用。 000：普通模式 001：只听模式 （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 本位设置 CAN 睡眠模式的自动唤醒模式有效 / 无效。如果设定本位，则 RCAN-ET 检测出 CAN 总线激活 （显性位）后，自动解除 CAN 睡眠模式 （MCR5）；如果清除本位，则 RCAN-ET 不自动解除 CAN 睡眠模式。 RCAN-ET 不可保存唤醒 RCAN-ET 的信息。 0：由 CAN 总线激活使自动唤醒模式无效 1：由 CAN 总线激活使自动唤醒模式有效 【注】在 CAN 睡眠模式不可更改 MCR7 位。

位	位名称	初始值	R/W	说　明
6	MCR6	0	R/W	总线断开时停机 本位设定在总线断开时如果 MCR1 置位，则立即转移至停机模式的操作有效或无效。仅可在复位或停机模式更改本位。必须注意，总线断开且转移至停机模式时， CAN 控制器也立即返回错误主动模式。 0：即使 MCR1 置位，在总线断开时，也不转移至停机模式，而等待返回顺序结束。 1：总线断开时，如果 MCR1 或 MCR14 有效，则立即转移至停机模式。
5	MCR5	0	R/W	CAN 睡眠模式 本位将向 CAN 睡眠模式的转移设置为有效 / 无效。RCAN-ET 在停机模式时，如果本位置位，则向 CAN 睡眠模式的转移有效，转移至停机模式后允许设定此位。 2 个错误计数器 （REC、 TEC）在 CAN 睡眠模式期间无变化。解除 CAN 睡眠模式的方法有两种： • 对 MCR5 位写入 0 • MCR7 有效时，检测 CAN 总线的显性位 如果自动唤醒模式无效，则在结束 CAN 睡眠模式前， RCAN-ET 忽略所有的 CAN 总线运行。解除 CAN 睡眠模式时， RCAN-ET 在转移至 CAN 总线运行前，通过检查 11 个隐性位与 CAN 总线取得同步。这意味着使用第两种方法时， RCAN-ET 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，因此需采用此方法设计软件。 在 CAN 睡眠模式，仅可存取 MCR、 GSR、 IRR 及 IMR 寄存器。 0：解除 CAN 睡眠模式 1：向 CAN 睡眠模式转移有效 【注】建议 RCAN-ET 最初设定为停机模式，再转移至 CAN 睡眠模式。由此，在转移至 CAN 睡眠模式前， CPU 可清除等待状态的所有中断。清除所有中断的同时， RCAN-ET 必须从停机模式转移至 CAN 睡眠模式 （具体来说，就是对 MCR5 写入 1 的同时对 MCR1 写入 0）。
4、3	—	均为 0	R	保留位 读取值、写入值总是为 0。
2	MCR2	0	R/W	信息发送优先级 本位可选择等待过程中发送数据的发送顺序。本位置 1 时，发送数据可根据发送等待寄存器 （TXPR）的位顺序发送。最先从邮箱 15 开始发送，持续发送到邮箱 1 为止 （邮箱设定为用于发送时）。如果清除本位，则所有发送信息 （通过运行内部仲裁）按照优先顺序入队列。最优先的信息具有最小数的仲裁字段 (STDID + IDE 位 + EXTID(IDE=1 时)+ RTR 位)，并最先发送。内部仲裁包含 RTR 位与 IDE 位 （通过与 2 个 CAN 节点之间的 CAN 总线的仲裁同时开始转移的相同方法运行内部仲裁）。仅可在复位或停机模式更改本位。 0：按信息 ID 优先顺序发送 1：按邮箱编号顺序 （邮箱 15 → 邮箱 1）发送

位	位名称	初始值	R/W	说　明
1	MCR1	0	R/W	停机请求 如果本位置位，则 CAN 控制器结束当前运行后转移至停机模式 （在此，从 CAN 总线分离）。 RCAN-ET 在清除本位前保持停机模式。在停机模式， CAN 接口与 CAN 总线运行无关，不保存或发送信息。除了用于自身通知停机状态的 IRR0 与 GSR4，还保持用户寄存器 （包含邮箱内容及 TEC/REC）的内容。 CAN 总线在空闲或间歇状态时，与 MCR6 无关， RCAN-ET 在 1 位间隔内为停机模式。如果 MCR6 置位，则总线断开时的停机请求也在 1 位间隔内运行。除此之外，事先结束总线断开返回顺序。如果为停机模式，则通过 IRR0 与 GSR4 通知。 MCR14 与 MCR6 置位时，如果 RCAN-ET 为总线断开状态，则本位立即自动置位。 在停机模式， RCAN-ET 与总线运行无关，因此除了设定位时序，还可更改 RCAN-ET 的设定。再次加入 CAN 总线运行时，需将本位清 0。清除后， RCAN-ET 等待检测出 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. 发行停机请求后， CPU 在完成向停机模式的转移前 （由 IRR0 及 GSR4 通知），不可存取 TXPR 与 TXCR，也不可清除本位。 MCR1 置位后，仅可在转移至停机模式后，或由 （软件或硬件）复位解除此状态。 2. 仅可在 BCR1 与 BCR0 寄存器设定为适当的波特率时，向停机模式转移或从停机模式转移。
0	MCR0	1	R/W	复位请求 本位可控制 RCAN-ET 模块的复位。本位从 0 变为 1 时， RCAN-ET 控制器进入复位程序，再次初始化内部逻辑，为了通知复位模式，置位 GSR3 及 IRR0。初始化所有用户寄存器。 本位置位期间， RCAN-ET 可重新设定 （结构模式）。为了加入 CAN 总线，本位需写入 0 后清除，清除后， RCAN-ET 等待检测出 11 个隐性位后，加入 CAN 总线。 为了采样 CAN 总线的值，需设定适当的波特率。上电复位后，此位与 GSR3 总是置位，这表示可请求复位，并需要设定 RCAN-ET。 复位请求相当于上电复位，由软件控制。 0：CAN 接口普通模式 （MCR0=0 且 GSR3=0） ［清除条件］复位 RCAN-ET 后写入 0 时 1：CAN 接口的复位模式转移请求

• MCR15（ID排序）＝0

地址	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	存取长度	字段名称
H'100+n*32	0	STDID[10:0]											RTR	IDE	EXTID[17:16]		16/32位	控制0
H'102+n*32	EXTID[15:0]																16位	
H'104+n*32	0	STDID_LAFM[10:0]											0	IDE_LAFM	EXTID_LAFM [17:16]		16/32位	LAFM字段
H'106+n*32	EXTID_LAFM[15:0]																16位	

• MCR15（ID排序）＝1

地址	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	存取长度	字段名称
H'100+n*32	IDE	RTR	0	STDID[10:0]											EXTID[17:16]		16/32位	控制0
H'102+n*32	EXTID[15:0]																16位	
H'104+n*32	IDE_LAFM	0	0	STDID_LAFM[10:0]											EXTID_LAFM [17:16]		16/32位	LAFM字段
H'106+n*32	EXTID_LAFM[15:0]																16位	

【注】 n：0～15（邮箱编号）

图 17.5　ID 排序

17.4.2　通用状态寄存器 （GSR）

GSR 为 16 位只读寄存器，可表示 RCAN-ET 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 读取值、写入值总是为 0。
5	GSR5	0	R	错误被动状态 表示 CAN 接口是否处于错误被动状态。如果 RCAN-ET 处于错误被动状态，则本位立即置 1，如果再次处于错误主动状态，则清除此位。这意味着在错误被动及总线断开状态， GSR5 保持为 1。因此，需要得知正确状态时，必须检查 GSR5 与 GSR0。 0：RCAN-ET 不处于错误被动或总线断开状态 ［清除条件］ RCAN-ET 处于错误主动状态期间 1：RCAN-ET 处于错误被动 （但 GSR0=0 时）或总线断开状态 （但 GSR0=1 时） ［置位条件］ TEC ≥ 128、 REC ≥ 128 或选择错误被动测试模式时
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。必须注意，此标志的清除时间与 IRR12 的置位时间不同。此标志反映 CAN 控制器的状态，但不能充分反映 RCAN-ET 的状态。结束 CAN 睡眠模式，如果清除 MCR5，则可存取 RCAN-ET。经过传送时钟 2 个位后结束 CAN 控制器的 CAN 睡眠模式。 0：RCAN-ET 不为停机模式也不为 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 置位且 CAN 总线为间歇 / 空闲状态时；或 MCR5 置位且 RCAN-ET 为停机模式时；或 MCR14 及 MCR6 均置位且 RCAN-ET 转移至总线断开时
3	GSR3	1	R	复位状态 表示 RCAN-ET 是否处于复位状态。 0：RCAN-ET 不处于复位状态 1：RCAN-ET 处于复位状态 ［置位条件］ RCAN-ET 的软件或硬件复位后
2	GSR2	1	R	信息发送进行标志 该标志向 CPU 显示 RCAN-ET 是否处于总线断开状态、是否在发送信息过程中或是否正在发送因在发送过程中检测出错误而引起的错误 / 超载标志。TXACK 的置位时序与 GSR2 的清除时序不同。 TXACK 在帧结束的第 7 个位置位。无发送等待信息时， GSR2 可在间歇状态的第 3 个位置位，也可通过仲裁失败、总线空闲、接收、复位或停机转移置位。 0：RCAN-ET 处于总线断开状态或在发送信息过程中 1：［置位条件］不处于总线断开状态，或不处于发送信息过程中
1	GSR1	0	R	发送 / 接收警告标志 该标志表示错误警告。 0：［清除条件］ TEC ＜ 96 且 REC ＜ 96 或总线断开时 1：［置位条件］ 96 ≤ TEC ＜ 256 或 96 ≤ REC ＜ 256 时 【注】　REC 计数总线断开返回顺序所需要 11 个隐性位的重复次数，因此在总线断开状态递增，但本位在总线断开状态不置位。
0	GSR0	0	R	总线断开标志 表示 RCAN-ET 处于总线断开状态。 0：［清除条件］从总线断开状态返回或硬件 / 软件复位后 1：［置位条件］ TEC ≥ 256 （总线断开状态）

17.4.3　位结构寄存器 0、 1 （BCR0、 BCR1）

BCR0、 BCR1 分别为可读取 / 写入的 16 位寄存器，设定 CAN 位时序参数与 CAN 接口的波特率预分频器。

时间量定义如下：

时间量 =2×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

有关 TSEG1 与 TSEG2 的设定，详情参照表 17.5。

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

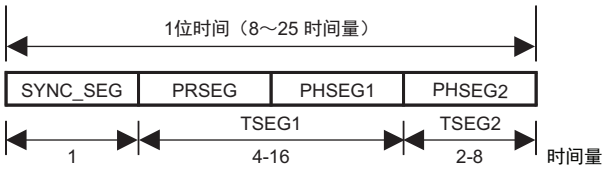
位	位名称	初始值	R/W	说　明
15 ～ 12	TSG1[3:0]	0000	R/W	时间段 1 以正相位误差补偿 CAN 总线的边沿，因此这些位用于设定段 TSEG1(=PRSEG + PHSEG1)，可设定 4 ～ 16 个时间量。 0000：禁止设定 0001：禁止设定 0010：禁止设定 0011：PRSEG + PHSEG1=4 个时间量 0100：PRSEG + PHSEG1=5 个时间量 ：　　： 1111：PRSEG + PHSEG1=16 个时间量
11	—	0	R	保留位 读取值、写入值总是为 0。
10 ～ 8	TSG2[2:0]	000	R/W	时间段 2 以负相位误差补偿 CAN 总线的边沿，因此这些位用于设定段 TSEG2(=PRSEG2)，可设定 2 ～ 8 个时间量。 000：禁止设定 001：PHSEG2=2 个时间量 （根据条件，禁止设定。详情参照表 17.5） 010：PHSEG2=3 个时间量 011：PHSEG2=4 个时间量 100：PHSEG2=5 个时间量 101：PHSEG2=6 个时间量 110：PHSEG2=7 个时间量 111：PHSEG2=8 个时间量
7、 6	—	均为 0	R	保留位 读取值、写入值总是为 0。
5、 4	SJW[1:0]	00	R/W	再次同步跳转宽度 设定同步跳转宽度。 00：同步跳转宽度 =1 个时间量 01：同步跳转宽度 =2 个时间量 10：同步跳转宽度 =3 个时间量 11：同步跳转宽度 =4 个时间量
3 ～ 1	—	均为 0	R	保留位 读取值、写入值总是为 0。
0	BSP	0	R/W	位采样点 设定采样数据的点。 0：在 1 处执行位采样 （时间段 1 的末尾） 1：在 3 处执行位采样 （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 读取值、写入值总是为 0。
7 ～ 0	BRP[7:0]	0000000	R/W	波特率预分频 这些位设定对应 1 个时间量的外围总线时钟数。 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×(寄存器值 +1)× 外围总线时钟 1111111：512× 外围总线时钟

- 位结构寄存器



SYNC_SEG：使 CAN 总线节点同步的段 （在此段产生通常的位边沿转移）
PRSEG　：调整网络间物理性延迟的段
PHSEG1　：相位漂移 （正向）的缓冲段 （同步或再次同步时扩展）
PHSEG2　：相位漂移 （负向）的缓冲段 （同步或再次同步时缩小）
TSEG1　：TSG1+1
TSEG2　：TSG2+1

RCAN-ET 位速率如下计算：

位速率 $=f_{clk} / (2 \times (BRP + 1) \times (TSEG1 + TSEG2 + 1))$

在此将寄存器值 +1 赋予 BRP， TSEG1 及 TSEG2 为后述表中计算出的值。上述位速率公式的时间段之所以为 “+1”，是根据 SYNC_SEC 的一个时间量得出。

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 个时间量 （不可为 TSEG1+TSEG2+1=7）
TSEG2 ≥ 2

如果位结构寄存器的 TSEG1 及 TSEG2 值在表 17.5 所示的设定值范围内，则满足上述限制事项。表 17.5 的值为可设定的 SJW 的值，显示 “不可” 的位置无法取得 TSEG1 与 TSEG2 的组合。

表 17.5　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：f_{clk} 为 40MHz 且位速率为 500kbps 时， BRP =3、TSEG1 = 6、TSEG2 = 3 满足条件。此时，对 BCR1 写入 H'5200，对 BCR0 写入 H'0003。

例 2：f_{clk} 为 35MHz 且位速率为 250kbps 时， BRP = 4、TSEG1 = 8、TSEG2 = 5 满足条件。此时，对 BCR1 写入 H'7400，对 BCR0 写入 H'0004。

例 3：f_{clk} 为 32MHz 且位速率为 500kbps 时， BRP = 1、 TSEG1 = 11、 TSEG2 = 4 满足条件。此时，对 BCR1 写入 H'A300，对 BCR0 写入 H'0001。

例 4：f_{clk} 为 20MHz 且位速率为 500kbps 时， BRP = 1、TSEG1 = 6、TSEG2 = 3 满足条件。此时，对 BCR1 写入 H'5200，对 BCR0 写入 H'0001。

17.4.4　中断请求寄存器 （IRR）

IRR 为可读取 / 写入的 16 位寄存器，由各种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
	—	—	IRR13	IRR12	—	—	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	R	R/W	R/W	R	R	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、14	—	均为 0	R	保留位
13	IRR13	0	R/W	信息错误中断 此中断表示在测试模式产生信息错误。但在测试模式即使产生信息超载条件，本位也不置位。除测试模式之外此功能无效。 0：在测试模式不产生信息错误 ［清除条件］写入 1 1：［置位条件］在测试模式产生信息错误
12	IRR12	0	R/W	CAN 睡眠模式时总线激活 本位表示 CAN 总线激活的存在。 RCAN-ET 为 CAN 睡眠模式时，如果在 CAN 总线检测出显性位，则此位置位。对此位写入 1，清除中断；写入 0 忽略。不使用自动唤醒功能，也不请求此中断时，必须通过对应的屏蔽位将此中断设置为无效；不使用自动唤醒功能，且请求此中断时，必须从 CAN 睡眠模式返回后清除此位。这是为了防止通过接收线的新下降沿再次设置中断。必须注意，此中断的设置时间与 GSR4 的清除时间不同。 0：总线空闲状态 ［清除条件］写入 1 1：在 CAN 睡眠模式检测出 CAN 总线激活 ［置位条件］ CAN 睡眠模式时，在 CRx 检测出显性位的状态变化
11、10	—	均为 0	R	保留位
9	IRR9	0	R	信息溢出 / 重写中断标志 本位表示已接收信息，但对应的 RXPR 或 RFPR 已置 1，因此不可读取邮箱中的信息，根据 NMC 位的设定舍弃 （溢出）或重写未通过 CPU 清除而新接收的信息。如果对 UMSR 寄存器的对应位写入 1，则清除此位，也可通过对 MBIMR 对应的所有位写入 1 清除此位，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 的所有位或所有已置位的 UMSR 对应的 MBIMR 置位 1：产生溢出，因此舍弃接收信息或重写信息 ［置位条件］对应的 PXPR 或 RFPR=1 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 如果一条设定为发送的信息正常发送 （对应的 TXACK 标志置位）或中止发送 （被取消发送的信息对应的 ABACK 标志置位），则此位置位。 此时清除对应的 TXPR，邮箱可接受下一条发送的信息数据。实际可通过未被 MBIMR 标志屏蔽的 TXACK 与 ABACK 位的 OR 信号置位本位，因此，如果清除所有 TXACK 及 ABACK 位，则此位自动清除。即使对 MBIMR 的所有对应位写入 1，也可清除此位，对此位的写入无效。 0：发送或发送取消的信息不在处理过程中 ［清除条件］ 清除所有 TXACK 及 ABACK 位，或置位所有已置位的 TXACK 及 ABACK 位对应的 MBIMR 1：发送信息或中止发送 （取消发送），可保存下一条信息 ［置位条件］ TXACK 或 ABACK 位置位时 （MBIMR=0 时）
7	IRR7	0	R/W	超载帧 本标志表示 RCAN-ET 检测出发送超载帧。对 IRR7 写入 1，清除该位；写入 0，忽略。 0：［清除条件］写入 1 1：［置位条件］检测出超载条件
6	IRR6	0	R/W	总线断开中断标志 RCAN-ET 为总线断开状态，或从总线断开状态返回错误主动状态时，本位置位。因此，节点的 TEC ≥ 256，总线断开返回顺序结束 （连续接收 128 次 11 个隐性位）或从总线断开状态向停止状态的转移 （自动或手动）为中断源。 RCAN-ET 为总线断开解除状态后本位仍为置位状态，因此需通过软件清除。必须通过软件读取 GSR0，并判断 RCAN-ET 处于总线断开状态或错误主动状态。即使节点仍处于总线断开状态，本位写入 1 时仍清除，写入 0 时仍忽略。 0：［清除条件］写入 1 1：因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进入错误被动状态或通过测试模式强行进入错误被动状态。对本位写入 1，清除；写入 0，忽略。有时即使清除此位，节点仍处于错误被动状态，需通过软件检查 GSR0 及 GSR5，并判断 RCAN-ET 处于错误被动状态或总线断开状态。 0：［清除条件］写入 1 1：发送 / 接收错误引起的错误被动状态 ［置位条件］ TEC ≥ 128、 REC ≥ 128 或使用错误被动测试模式
4	IRR4	0	R/W	接收错误计数器警告中断标志 RCAN-ET 不处于总线断开状态时，如果接收错误计数器 （REC）超过 95，则置位。对本位写入 1，清除中断；写入 0，忽略。 0：［清除条件］写入 1 1：接收错误引起的错误警告状态 ［置位条件］ RCAN-ET 处于总线断开状态以外且 REC ≥ 96

位	位名称	初始值	R/W	说　明
3	IRR3	0	R/W	发送错误计数器警告中断标志 如果发送错误计数器 （TEC）超过 95，则置位。对本位写入 1，清除中断；写入 0，忽略。 0：［清除条件］写入 1 （使用中断或通过 CPU 清除时，必须在写入 1 后读取标志） 1：发送错误引起的错误警告状态 ［置位条件］ TEC ≥ 96
2	IRR2	0	R	远程帧请求中断标志 表示邮箱接收远程帧。如果至少 1 个未置位的对应 MBIMR 的邮箱保存远程帧发送请求，则本位置位。如果清除远程帧接收完成寄存器 （RFPR）的所有位，则清除此位。对 MBIMR 的所有对应位写入 1，清除此位，对此位的写入无效。 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 表示存在处理等待状态的接收数据帧。本位为 1 时，至少在 1 个邮箱保存未处理的信息。如果清除数据帧接收完成寄存器 （RXPR）的所有位 （任何接收邮箱均无未处理的信息），则清除此位。设定 MBIMR 不为 0 的各接收邮箱 RXPR 标志的逻辑 “或”。 对 MBIMR 的所有对应位写入 1，清除此位，对此位的写入无效。 0：［清除条件］清除 RXPR 的所有位 1：接收数据帧，并保存在邮箱 ［置位条件］对应的 MBIMR=0 时，接收数据
0	IRR0	1	R/W	复位 / 停机 / 睡眠中断标志 通过下述 3 个中断源置位： • 软件复位 （MCR0）或硬件复位后，转移至复位模式 • 请求停机 （MCR1）后，转移至停机模式 • 在停机模式发行睡眠请求 （MCR5）后，转移至 CAN 睡眠模式 本位置位后，读取 GSR，可得知 RCAN-ET 的状态。 【注】 需发行睡眠请求时，必须事先使用停机模式，详情参阅 MCR5 的说明及图 17.8。 GSR3或GSR4从0变为1时，或从停机模式转移至CAN睡眠模式时，IRR0置位。因此， RCAN-ET在停机模式解除后立即再次转移至停机模式时，不清除GSR4， IRR0不置位。同样，从CAN睡眠模式直接转移至停机模式时， IRR0也不置位。从停机模式/CAN睡眠模式转移至发送/接收运行时，清除GSR4前需要 [1位时间 － SEG2]～[1位时间×2 － TSEG2]的时间。 在复位模式， IRR0置位，但通过初始化IMR0可自动置位，因此向CPU请求的中断无效。 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］ 在请求复位 （MCR0 或硬件）、停机模式 （MCR1）或 CAN 睡眠模式 （MCR5）后，完成向复位 / 停机 /CAN 睡眠模式的转移。

17.4.5　中断屏蔽寄存器 （IMR）

IMR 为可读取 / 写入的 16 位寄存器，屏蔽生成中断请求寄存器 （IRR）各位所对应中断的 IRQ 输出信号。如果将位设定为 1，则屏蔽对应的中断请求。 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] 各位对应的中断源 如果本位置位，即使 IRR 的对应位置位，也不生成此中断信号。 0：不屏蔽对应的 IRR （如果产生中断源，则生成 IRQ） 1：屏蔽 IRR 的对应中断

17.4.6　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，也为显示 CAN 接口的发送 / 接收信息错误数的计数器。计数值在上述参考文献的 [1]、[2]、[3]、[4] 有所规定。除写错误计数器测试模式之外，本寄存器为只读寄存器，仅可通过 CAN 接口改写。通过转移至复位请求 （MCR0）或总线断开状态，清除本寄存器。

在写错误计数器测试模式 （TST[2:0]=B'100），可向本寄存器写入。仅可对 TEC 及 REC 写入相同值。在 TEC 及 REC 设定写入 TEC 的值。写入本寄存器时，需将 RCAN-ET 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】 * 仅可在通过 MCR 的 TST[2:0]=B'100 进入测试模式时写入。在总线断开状态， REC 计数总线断开返回顺序所需的 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 发送过程中，如果检测出基于 CAN 规格的错误，则本寄存器的值递增。
7 ～ 0	REC[7:0]	均为 0	R/W*	接收错误计数器 接收过程中，如果检测出基于 CAN 规格的错误，则本寄存器的值递增。

17.5　RCAN-ET 邮箱寄存器

说明 RCAN-ET 邮箱寄存器的相关内容。 RCAN-ET 邮箱寄存器可控制各邮箱，表示邮箱状态。

RCAN-ET 邮箱寄存器如表 17.6 所示。

【重要】　长字存取相当于 2 个连续的字存取。

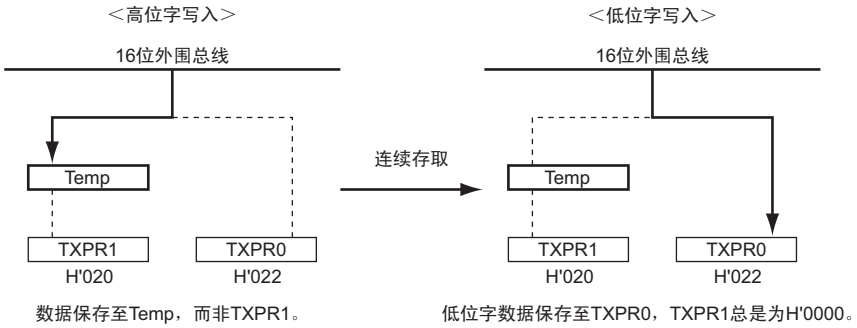
表 17.6　RCAN-ET 邮箱寄存器

寄存器名称	简称	地址	存取长度 （位）
发送等待寄存器 1	TXPR1	H'020	长字
发送等待寄存器 0	TXPR0	H'022	—
		H'024	
		H'026	
		H'028	
发送取消寄存器 0	TXCR0	H'02A	字
		H'02C	
		H'02E	
		H'030	
发送应答寄存器 0	TXACK0	H'032	字
		H'034	
		H'036	
		H'038	
中止应答寄存器 0	ABACK0	H'03A	字
		H'03C	
		H'03E	
		H'040	
数据帧接收完成寄存器 0	RXPR0	H'042	字
		H'044	
		H'046	
		H'048	
远程帧接收完成寄存器 0	RFPR0	H'04A	字
		H'04C	
		H'04E	
		H'050	
邮箱中断屏蔽寄存器 0	MBIMR0	H'052	字
		H'054	
		H'056	
		H'058	
未读信息状态寄存器 0	UMSR0	H'05A	字
		H'05C	
		H'05E	

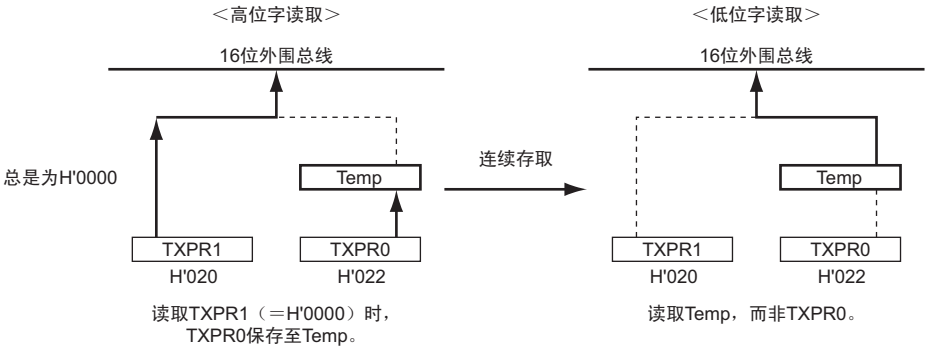
17.5.1　发送等待寄存器 1、 0 （TXPR1、 TXPR0）

连接 TXPR1 与 TXPR0，构成保存 CAN 模块发送等待标志的 32 位寄存器。为 16 位总线接口时，长字存取相当于 2 个连续的字存取。

＜长字写入操作＞



＜长字读取操作＞



TXPR1 寄存器总是固定为 0，且对 TXPR1 寄存器的写入无效。 TXPR0 寄存器控制邮箱 15 ～ 1。通过对位写入 1， CPU 可操作对应的发送信息。写入 0 无效，清除 TXPR 时，必须置位 TXCR 的对应位。通过读 TXPR， CPU 可得知发送等待的邮箱及发送过程中的邮箱。实际除邮箱 0 之外的所有邮箱均有发送等待位。禁止对未设定为发送邮箱的位写入 1。

正常发送信息或中止 TXCR 的发送后， RCAN-ET 清除对应的发送等待标志。因 CAN 节点仲裁失败或在 CAN 总线产生错误，不发送信息时，不清除 TXPR 的标志，此时，如果对应邮箱信息控制字段的 DART （自动重新发送无效）位未置位，则 RCAN-ET 尝试自动重新发送；如果 DART 置位，则清除发送，并通过邮箱空中断标志 （IRR8）及中断应答寄存器 （ABACK）的对应位通知 CPU。

TXPR 状态发生变化时，即使总线仲裁失败或在 CAN 总线产生错误，仍总是发送 ID 所示的优先顺序（MCR2=0）最高的信息。详情参阅 “17.6 运行说明”。

RCAN-ET 将 TXPR 的位状态更改为 0 时，有时会生成邮箱空中断标志 （IRR8），这表示邮箱的发送正常结束或中止。信息发送正常结束时，反映在 TXACK ；信息发送中止时，反映在 ABACK。通过检查这些寄存器，可改写对应邮箱的信息数据字段的内容，用于下次发送。

(1)　TXPR1

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXPR1[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】 * 总是读取 H'0000。必须以长字存取的方式读取 / 写入 TXPR1/TXPR0，对 TXPR1 的写入无效。

(2)　TXPR0

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXPR0[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】 * 仅可对设定为发送的邮箱的对应位写入 1。
总是读取 H'0000。必须以长字存取的方式读取 / 写入 TXPR1/TXPR0，对 TXPR0 的 bit0 的写入无效。

位	位名称	初始值	R/W	说　明
15 ～ 1	TXPR0[15:1]	H'0000	R/W	表示在对应的邮箱产生 CAN 帧发送请求。 bit15 ～ 1 对应邮箱 15 ～ 1。多个位置位时，通过设定 MCR2 位，按照信息 ID 优先顺序或邮箱编号顺序发送。 0：对应的邮箱为发送信息空闲状态 ［清除条件］信息发送结束或信息发送中止 （自动清除） 1：在对应的邮箱产生发送请求
0	—	0	R	保留位 为接收专用邮箱，因此本位总是为 0。写入 1 时忽略，读取值为 0。

17.5.2　发送取消寄存器 0 （TXCR0）

TXCR0 为可读取 / 带条件写入的 16 位寄存器，控制邮箱 15 ～ 1。 CPU 使用 TXCR0，取消 TXPR 的发送请求。清除 TXPR 的位时，必须由 CPU 向对应的 TXCR 的位写入 1，写入 0 时忽略。

如果正常执行中止，则 CAN 控制器清除对应的 TXPR 与 TXCR 的位，并置位 ABACK 的位。但邮箱开始发送后，不可通过本寄存器的位取消。此时，如果发送正常结束，则 CAN 控制器清除 TXPR 与 TXCR 的位，并置位 TXACK 的位。但仲裁失败或在总线产生错误而不能发送时， CAN 控制器清除对应的 TXPR 与 TXCR 的位，并置位对应的 ABACK 的位。即使 CPU 要清除非发送等待状态的邮箱的发送，仍被忽略。此时， CPU 不可置位 TXCR 的标志。

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】 * 仅可对已接受发送请求的发送邮箱的对应位写入 1。

位	位名称	初始值	R/W	说　明
15 ～ 1	TXCR0[15:1]	H'0000	R/W	对进入发送队列的邮箱，请求取消发送。 bit15 ～ 1 对应邮箱 15 ～ 1 （及 TXPR0[15:1]）。 0：对应的邮箱为取消发送信息的空闲状态 ［清除条件］取消发送信息结束 （自动清除） 1：向对应的邮箱请求取消发送
0	—	0	R	保留位 为接收专用邮箱，因此本位总是为 0。写入 1 时忽略，读取值总是为 0。

17.5.3　发送应答寄存器 0 （TXACK0）

TXACK0 为可读取 / 带条件写入的 16 位寄存器，用于向 CPU 通知邮箱已正常发送。如果正常发送，则 RCAN-ET 置位 TXACK0 寄存器的对应位。通过写入 1， CPU 可清除 TXACK0 的位，写入 0 时忽略。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXACK0[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】 * 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	通知向对应邮箱请求的发送已正常执行。 bit15 ～ 1 对应邮箱 15 ～ 1。 0：［清除条件］写入 1 1：正常发送对应邮箱的信息 （数据帧或远程帧） ［置位条件］对应邮箱的信息发送结束
0	—	0	R	保留位 为接收专用邮箱，因此本位总是为 0。写入 1 时忽略，读取值总是为 0。

17.5.4　中止应答寄存器 0 （ABACK0）

ABACK0 为可读取 / 带条件写入的 16 位寄存器，用于向 CPU 通知根据请求中止邮箱发送。如果执行中止，则 RCAN-ET 置位 ABACK0 寄存器的对应位。通过写入 1， CPU 可清除 ABACK0 的位，写入 0 时忽略。RCAN-ET 通过置位 ABACK0 的位，确认已通过对应的 TXCR 位清除 TXPR 位。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ABACK0[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】 * 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	通知向对应邮箱请求的发送取消已正常执行。 bit15 ～ 1 对应邮箱 15 ～ 1。 0：［清除条件］写入 1 1：取消发送对应邮箱的信息 （数据帧或远程帧） ［置位条件］取消发送对应邮箱的信息
0	—	0	R	保留位 为接收专用邮箱，因此本位总是为 0。写入 1 时忽略，读取值总是为 0。

17.5.5　数据帧接收完成寄存器 0 （RXPR0）

RXPR0 为可读取 / 带条件写入的 16 位寄存器，由表示设定为接收的邮箱已接收数据帧的标志构成。如果 CAN 数据帧正常保存在接收邮箱，则 RXPR 的对应位置位。如果写入 1，则清除对应位；写入 0，则无效。但仅在通过 MBC （邮箱结构）将邮箱设定为可接收数据帧时才置位。如果 RXPR 的位置位，即使对应的 MBIMR 不置位， IRR1 （数据帧接收中断标志）也置位， IMR1 不置位时，生成中断信号。本寄存器的位仅在接收数据帧时置位，接收远程帧时不置位。

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 对应设定为接收的邮箱 15 ～ 0。 0：［清除条件］写入 1 1：对应邮箱接收 CAN 数据帧 ［置位条件］对应邮箱的数据帧接收结束

17.5.6　远程帧接收完成寄存器 0 （RFPR0）

RFPR0 为可读取 / 带条件写入的 16 位寄存器，由表示设定为接收的邮箱已接收远程帧的标志构成。如果 CAN 远程帧正常保存在接收邮箱，则 RFPR 的对应位置位。如果写入 1，则清除对应位；写入 0，则无效。所有邮箱均具有位，但仅在通过 MBC （邮箱结构）将邮箱设定为可接收远程帧时才置位。如果 RFPR 的位置位，即使对应的 MBIMR 不置位， IRR2 （远程帧请求中断标志）也置位， IMR2 不置位时，生成中断信号。本寄存器的位仅在接收远程帧时置位，接收数据帧时不置位。

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	该标志为邮箱 15 ～ 0 的远程请求接收标志。 0：［清除条件］写入 1 1：对应邮箱接收 CAN 远程帧 ［置位条件］对应邮箱的远程帧接收结束

17.5.7　邮箱中断屏蔽寄存器 0 （MBIMR0）

MBIMR0 为可读取 / 写入的 16 位寄存器。 MBIMR0 可屏蔽与邮箱运行相关的 IRR （IRR1：数据帧接收中断， IRR2：远程帧请求中断， IRR8：邮箱空中断， IRR9：信息溢出中断）。邮箱设定为接收时，屏蔽接收中断标志（IRR1、IRR2、IRR9）引起的中断，不影响 RXPR、RFPR、UMSR 位的置位。邮箱设定为发送时，屏蔽发送或发送中止 （IRR8）引起的中断或邮箱空中断，不影响发送引起的 TXPR/TXCR 位的清除及 TXACK 位的置位，也不影响发送中止引起的 TXPR/TXCR 位的清除及 ABACK 位的置位。

通过对屏蔽的邮箱所对应的位写入 1 来设定屏蔽。复位时屏蔽所有邮箱中断。

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

17.5.8　未读信息状态寄存器 0 （UMSR0）

UMSR0 为可读取 / 带条件写入的 16 位寄存器，接收新信息时，由 CPU 记录未存取内容的接收邮箱。接收邮箱的新信息时，如果 CPU 未清除 RXPR0 或 RFPR0 的对应位，则 UMSR0 的位置位。如果写入 1，则清除；写入 0，则无效。

邮箱设定为发送时，对应 UMSR0 的位不置位。

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 的未读接收信息被重写或产生溢出。 0：［清除条件］写入 1 1：未读接收信息被重写新信息或产生溢出 ［置位条件］清除 RXPR0 或 RFPR0 前，接收新信息

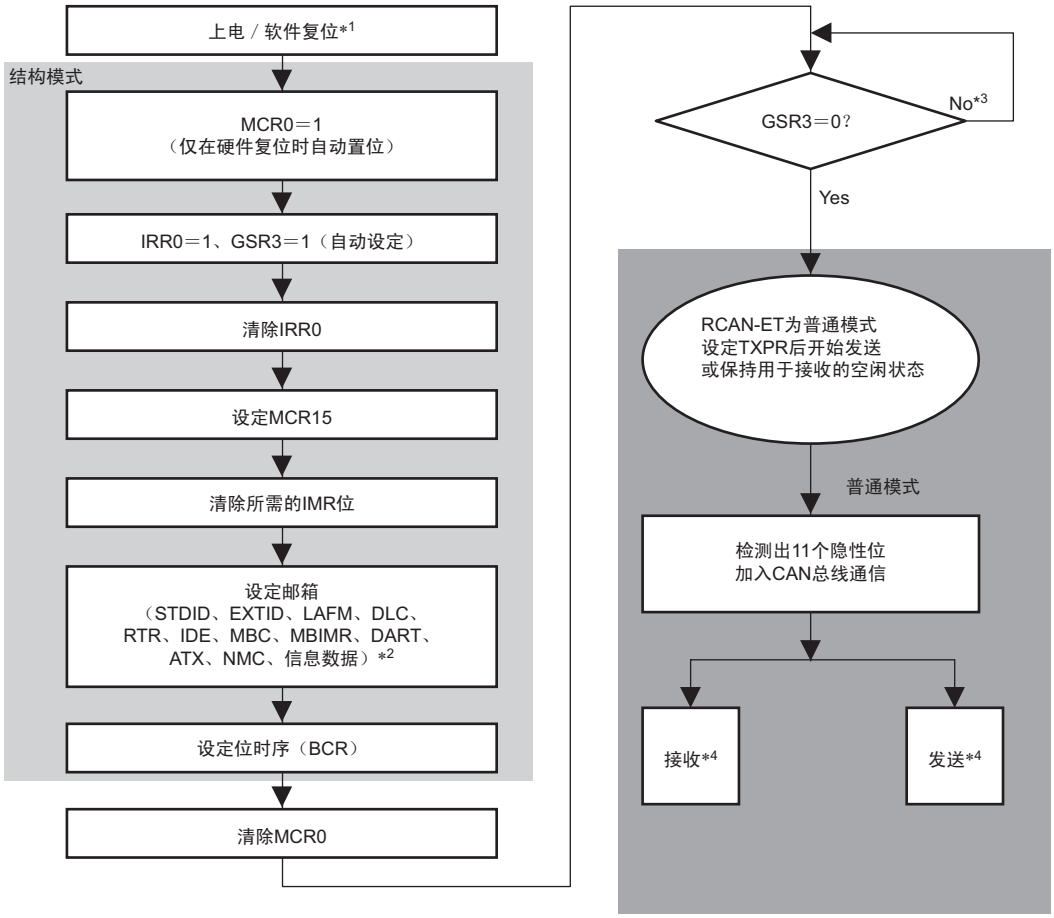
17.6　运行说明

17.6.1　RCAN-ET 的设定

说明硬件复位 （上电复位）或软件复位 （MCR0）后的结构模式及停机模式下的 RCAN-ET 的设定。在任何情况， RCAN-ET 均不可加入 CAN 总线激活。更改 RCAN-ET 的设定不影响 CAN 总线通信。

(1)　复位顺序

软件复位或硬件复位后 RCAN-ET 的设定顺序如图 17.6 所示。初始化复位后的所有寄存器，因此，加入 CAN 总线激活前需设定 RCAN-ET，详情参照图中注释。



【注】 *1　设定MCR0＝1，可随时执行软件复位。
*2　邮箱由RAM构成，因此必须将即使没有使用，但已通过MBC设置为有效的所有邮箱初始化。
*3　该判断到为0之前大约需要25个外围总线周期。
*4　如果TXPR0位均未置位，则RCAN-ET接收下一条信息。
如果TXPR0被设定，则RCAN-ET开始发送信息，通过CAN总线执行仲裁。如果仲裁失败，则接收信息。

图 17.6　复位顺序

(2)　停机模式

RCAN-ET 在停机模式时不可加入 CAN 总线激活，因此，不影响 CAN 总线通信，用户可更改所需寄存器的设定。重要的是：更改寄存器前需等到 RCAN-ET 成为停机模式，但未必立即转移至停机模式 （在 CAN 总线空闲或停机状态时转移）。如果 RCAN-ET 转移至停机模式，则 GSR4 位置位。

设定结束后需解除停机请求。 RCAN-ET 在 CAN 总线检测出 11 个隐性位后加入 CAN 总线激活。

(3)　CAN 睡眠模式

在 CAN 睡眠模式，为了降低消耗电流，停止 RCAN-ET 主要模块的时钟，仅可存取 MCR、 GSR、 IRR 及 IMR 寄存器。在 CAN 睡眠模式不可清除与发送 / 接收相关的中断，因此必须预先清除。

RCAN-ET 的 CAN 睡眠模式流程图如图 17.7 所示。

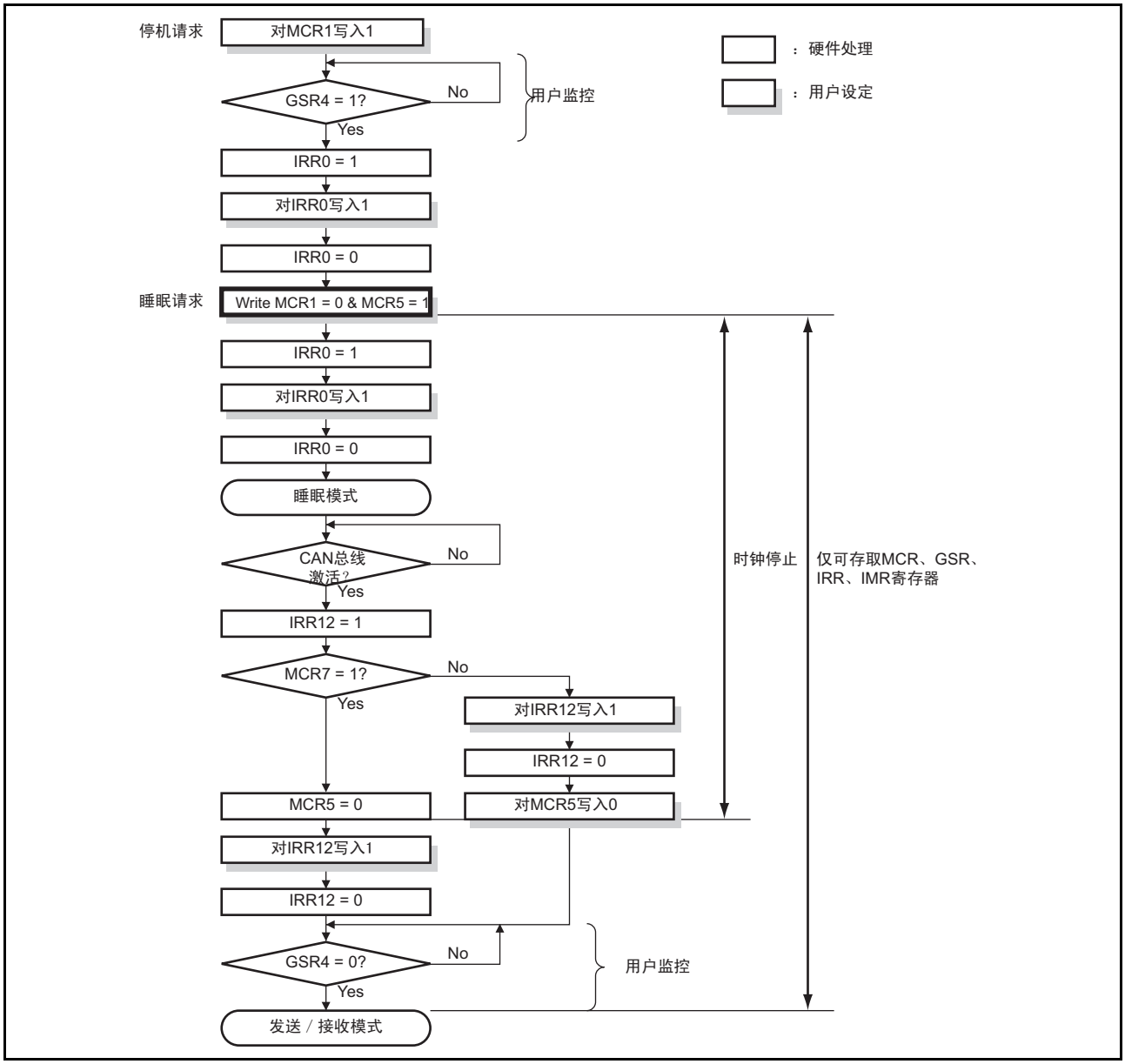


图 17.7　CAN 睡眠模式流程图

可能的状态转移如图 17.8 所示。

除停机模式之外的模式不得置位 MCR5 （CAN 睡眠模式）位。

置位 MCR1 位后，清除 MCR1 前，必须确认 GSR4 置位，且 RCAN-ET 为停机模式。

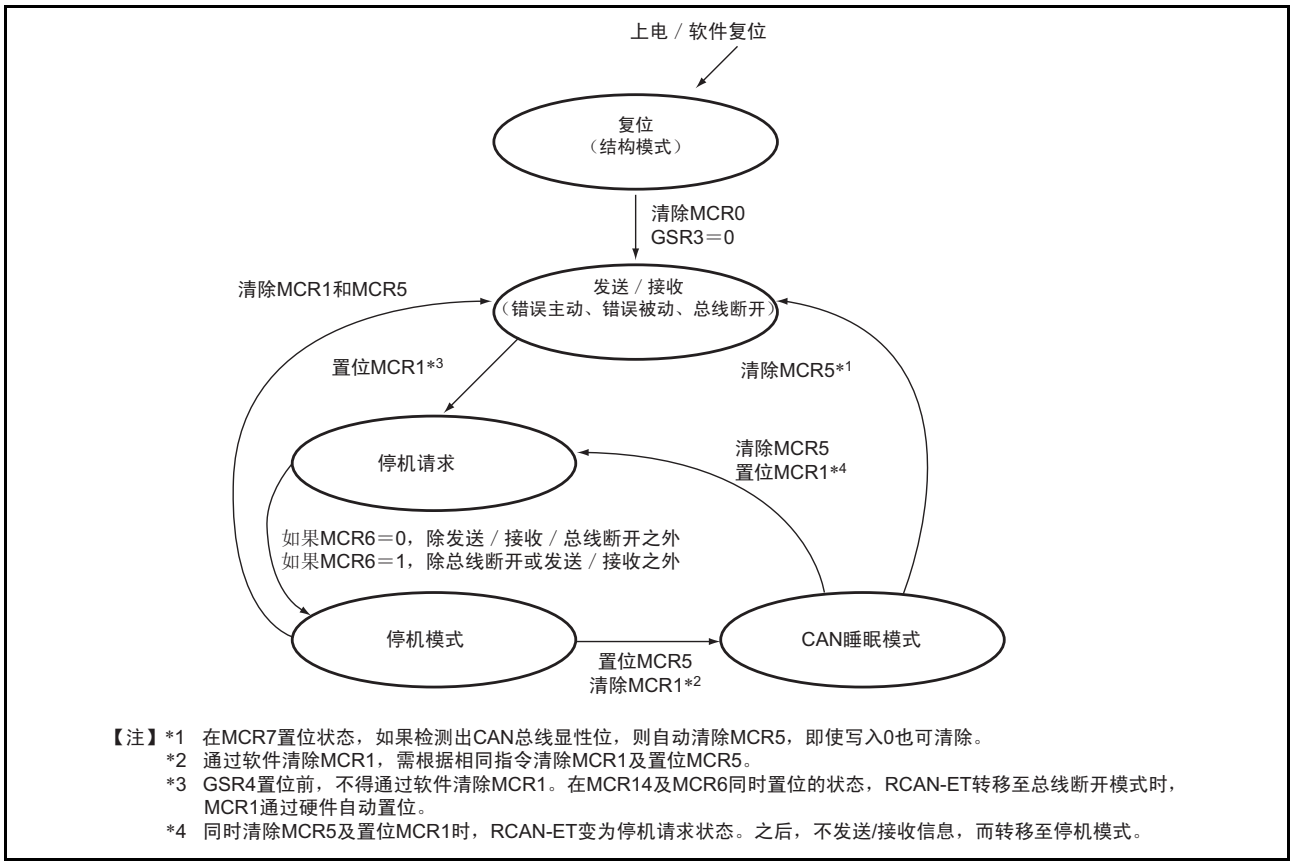


图 17.8　状态转移图

允许在各模式存取的条件如表 17.7 所示。

表 17.7　可存取的寄存器

状态模式	寄存器									
	MCR、GSR	IRR、IMR	BCR	MBIMR	标志寄存器	邮箱（控制 0、LAFM）		邮箱（数据）	邮箱（控制 1）	
复位	Yes	Yes	Yes	Yes	Yes	Yes		Yes	Yes	
发送 / 接收	Yes	Yes	No	Yes	Yes	No	Yes*	Yes*	No	Yes*
停机请求	Yes	Yes	No	Yes	Yes	No	Yes*	Yes*	No	Yes*
停机模式	Yes	Yes	No	Yes	Yes	Yes		Yes	Yes	
CAN 睡眠模式	Yes	Yes	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 未置位时

17.6.2　测试模式的设定

RCAN-ET 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-ET 在普通模式以默认（初始值）方式运行。

测试模式的设定如表 17.8 所示。

仅可在结构模式选择测试模式。执行选择的测试模式时，必须在选择测试模式后解除结构模式 （确认已设定 BCR0/BCR1）。

表 17.8　测试模式的设定

TST2	TST1	TST0	模式
0	0	0	普通模式
0	0	1	只听模式 （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-ET普通运行。
- 只听模式
  该模式为ISO-11898标准要求的检测波特率的模式。错误计数器在清除后禁止运行，且TEC/REC的值不增加。禁止CTx输出，抑制由RCAN-ET生成错误帧或应答位。如果产生信息错误，则IRR13置位。
- 自测试模式1
  RCAN-ET自身生成应答位，需要时将信息保存至RCAN-ET的接收邮箱。 CRx/CTx引脚必须与CAN总线连接。
- 自测试模式2
  RCAN-ET自身生成应答位，需要时将信息保存至RCAN-ET的接收邮箱。内部CTx被循环反馈至内部CRx，因此无需将CRx/CTx引脚连接至CAN总线的其他外部器件。 CTx引脚仅输出隐性位， CRx引脚无效。
- 写错误计数器
  TEC/REC可在此模式写入。通过对错误计数器写入大于127的值，可强行将RCAN-ET设置为错误被动模式。写入TEC的值被写入REC，因此TEC与REC总是设定为相同值。同样，通过写入大于95的值，可强行将RCAN-ET设置为错误警告模式。
  写入TEC/REC时， RCAN-ET必须为停机模式 （错误计数器写入时， MCR1=1）。而且按照CAN规格，需在解除停机模式前解除此测试模式，使TEC/REC可以再次运行。
- 错误被动模式
  可强行将RCAN-ET设置为错误被动模式。
  执行错误被动模式，不改变REC的值，但是一旦在错误被动模式运行，接收错误时， REC的值照常增加。在此模式，如果TEC的值达到256，则RCAN-ET为总线断开状态，但使用此模式时， RCAN-ET不可为错误主动状态。因此， RCAN-ET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。

产生信息错误时，在所有测试模式 IRR13 位均被置位。

17.6.3 信息发送顺序

(1) 信息发送请求

在总线发送 CAN 帧的顺序如图 17.9 所示。

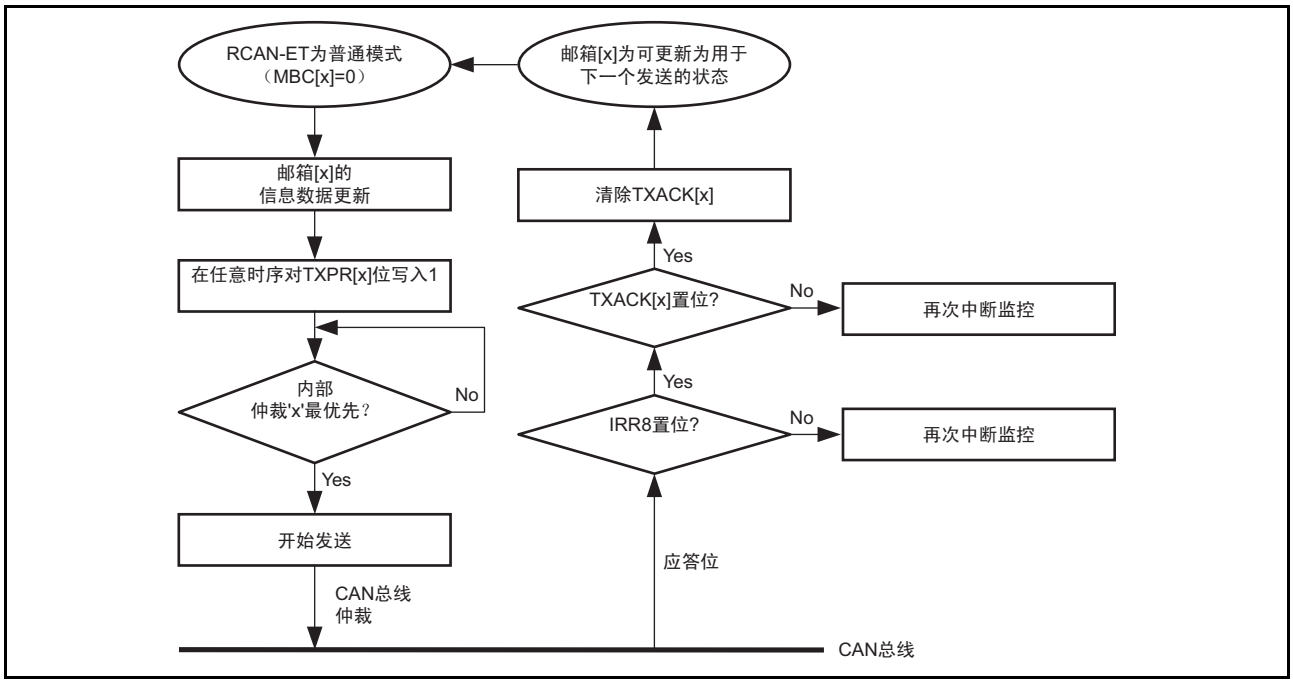


图 17.9 信息发送请求

注意：根据在寄存器章节的说明，如果 TXACK 或 ABACK 位的任意一个位置位，则 IRR8 置位。这意味着任意一个邮箱结束发送或发送中止，并可更新为用于下一个发送的状态。另一方面， GSR2 表示未产生当前发送请求 （所有 TXPR 标志均未置位）。

(2)　用于发送的内部仲裁

图 17.10 说明 RCAN-ET 如何按照信息 ID 的顺序调度已请求发送的信息。通过内部仲裁获取优先级最高的已请求发送的信息。

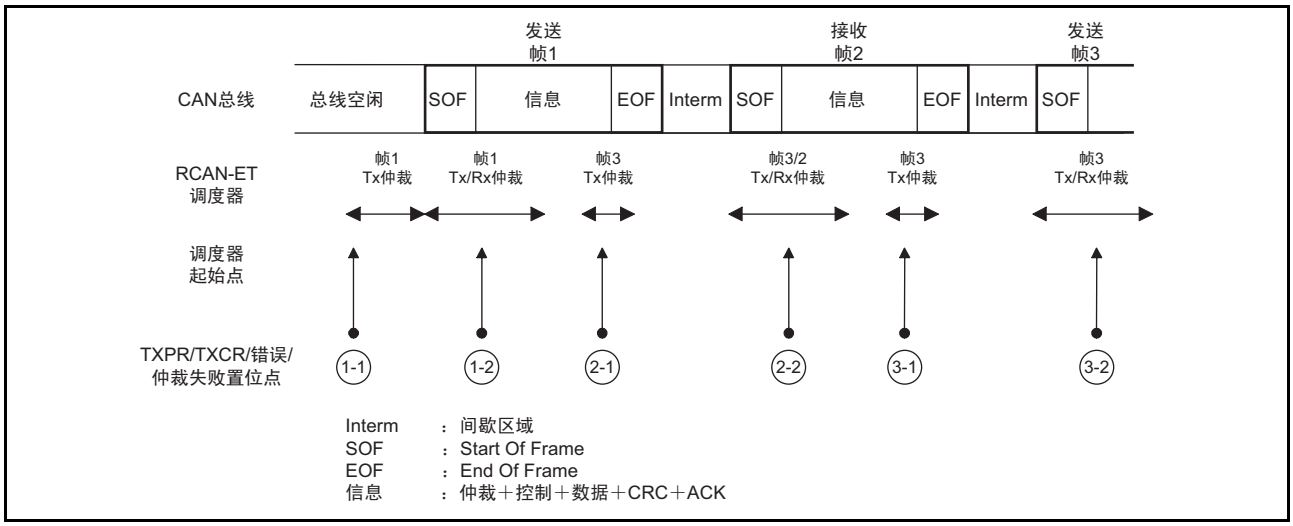


图 17.10　用于发送的内部仲裁

1-1 ：CAN 总线为空闲状态时，如果 TXPR 位置位，则立即执行内部仲裁，并开始发送。
1-2 ：通过 SOF 开始发送与接收。不为接收帧，因此 RCAN-ET 执行发送。
2-1 ：在 CRC 的定界符开始内部仲裁，并搜索下一条发送信息。
2-2 ：通过 SOF 开始发送与接收。为高优先级的接收帧时， RCAN-ET 执行接收，因此不发送帧 3 而接收。
3-1 ：在 CRC 的定界符开始内部仲裁，并搜索下一条发送信息。
3-2 ：通过 SOF 开始发送与接收。发送帧的优先级比接收帧高，因此 RCAN-ET 执行发送。

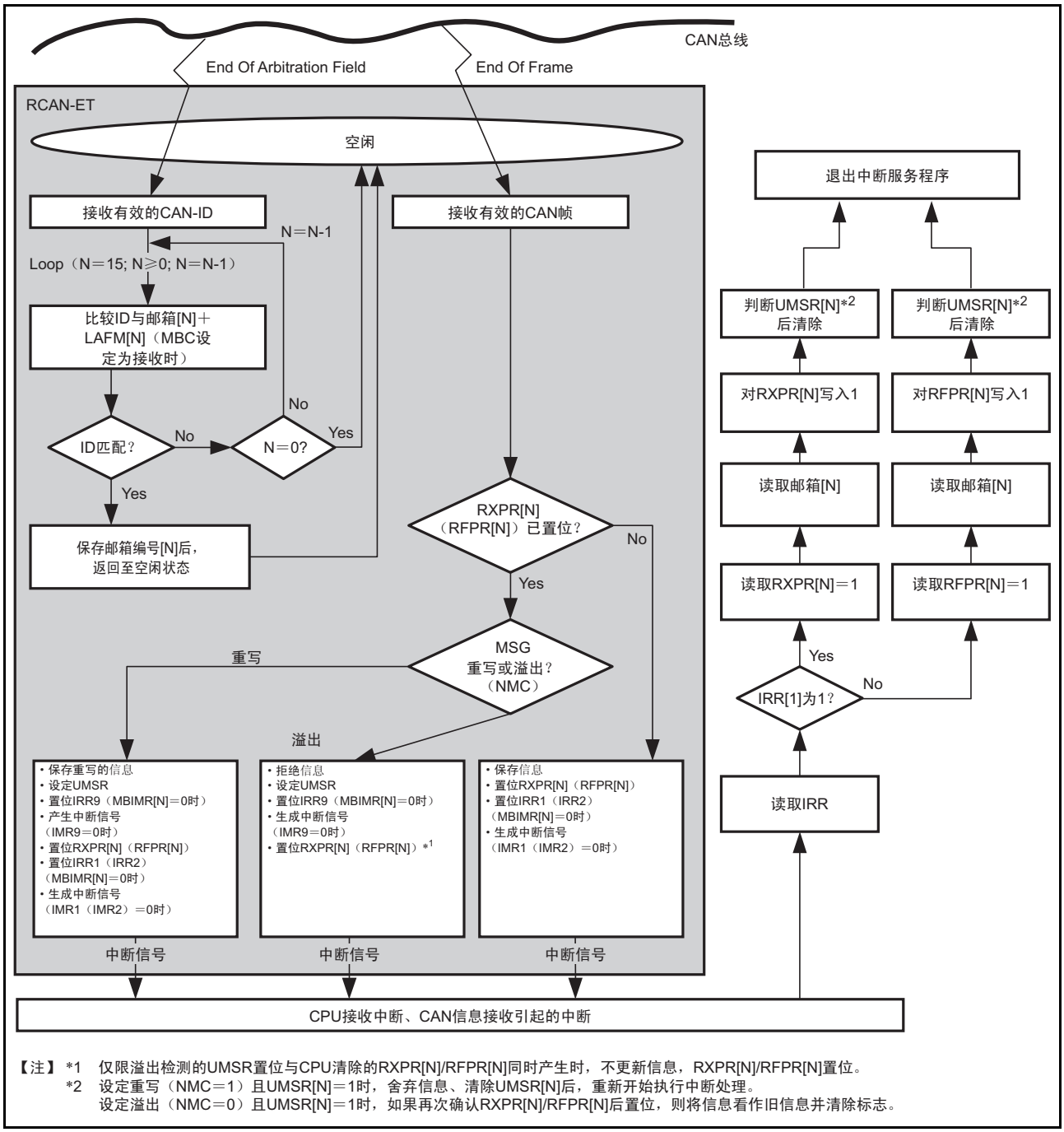
检测出 CAN 总线的错误时，也可在各错误定界符的起始点执行下一个用于发送的内部仲裁，还可在超载帧后的错误定界符的起始点执行内部仲裁。

可在 CRC 的定界符执行用于发送的内部仲裁，因此， ATX=1 的邮箱接收远程帧请求时，根据该请求，发送的信息加入用于发送的内部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 置位后，在最多延迟 1 个 CAN 帧后执行对应信息的中止处理。

17.6.4　信息接收顺序

信息接收顺序如图 17.11 所示。



【注】 *1　仅限溢出检测的UMSR置位与CPU清除的RXPR[N]/RFPR[N]同时产生时，不更新信息，RXPR[N]/RFPR[N]置位。
*2　设定重写（NMC＝1）且UMSR[N]＝1时，舍弃信息、清除UMSR[N]后，重新开始执行中断处理。
设定溢出（NMC＝0）且UMSR[N]＝1时，如果再次确认RXPR[N]/RFPR[N]后置位，则将信息看作旧信息并清除标志。

图 17.11　信息接收顺序

接收信息过程中，如果 RCAN-ET 识别出仲裁字段的末尾，则开始比较接收的 ID 与邮箱设定的 ID。比较顺序为从邮箱 15 到邮箱 0。首先，检查 MBC，检查邮箱是否设定为接收。此后读取 LAFM，接着读取邮箱 15（设定为接收时）的 ID，与接收到的 ID 比较。不匹配时，对邮箱 14 （设定为接收时）执行相同检查，如果发现匹配的 ID，则 RCAN-ET 将此邮箱编号 （N）保存在内部缓冲器，停止检索后返回空闲状态，并等待 EOF（End Of Frame）。如果通知 EOF 的第 6 位，则根据 NMC 位的设定写入或舍弃接收信息。

通信过程中，不可更改 RCAN-ET 的信息 ID 及 LAFM 的设定。更改设定的方法，有停机模式及结构模式。将信息写入对应的邮箱时，由于写入包含信息 ID，因此根据使用的 LAFM，有可能用接收信息不同的 CAN-ID 重写 CAN-ID。这意味着接收信息的 ID 与多个邮箱的 ID+LAFM 匹配时，接收信息总是保存在邮箱编号最大的邮箱，而不保存在编号小的邮箱。因此，需注意仔细选择 ID 与 LAFM 的设定值。

如图 17.11 所示，接收数据及远程帧时，读取 IRR 后清除 UMSR 标志，这是为了在执行中断服务程序过程中，检测用保存在相同邮箱的新信息重写信息的情况。在 UMSR 的最后检查过程中，检测出重写时，需舍弃信息，再次读取。

UMSR 置位的邮箱设定为溢出模式 （NMC=0）时，不重写而保留信息，但不是 CAN 总线监控的最新信息。

清除相应邮箱的 RXPR/RFPR 标志前，必须读取该邮箱的所有内容 （邮箱 [N] 的区域）。

以数据帧重写接收到的远程帧时，必须注意：远程帧请求中断 （IRR2）及数据帧接收中断 （IRR1）置位，接收标志 （RXPR、 RFPR）也置位。同样，以远程帧重写数据帧时， IRR2 及 IRR1 也置位。

在溢出模式 （NMC=0），这些标志仅通过最初的邮箱置位。因此，最初如果接收数据帧，则 RXPR 及 IRR1 有效。此后如果在读取数据帧前接收远程帧，则 RFPR 及 IRR2 不置位。此时，对应邮箱的 UMSR 置位。

17.6.5　邮箱的重新设定

需重新设定邮箱时必须按照下述顺序操作：

(1)　更改发送邮箱的设定

有下述两种情况：

- 更改ID、 RTR、 IDE、 LAFM、数据、 DLC、 NMC、 ATX、 DART及MBC
  仅可在MBC=B'000时更改。必须确认对应的TXPR位未置位，可随时更改设定 （除MBC之外）。
- 从发送邮箱更改为接收邮箱
  必须确认对应的TXPR位未置位，仅可在停机模式或复位状态更改。RCAN-ET在接收或发送信息过程中，需要一定时间才能转移至停机状态 （因为需等待发送/接收结束后再转移至停机状态）。必须注意：不可在停机状态发送/接收信息。
  RCAN-ET为总线断开状态时，依据MCR寄存器的bit6及bit14的设定转移至停机状态。

(2)　更改接收邮箱的 ID、RTR、IDE、LAFM、数据、DLC、NMC、ATX、DART、MBC 的设定及从接收邮箱更改为发送邮箱

仅可在停机模式更改设定。信息在 CAN 总线且 RCAN-ET 为接收模式时，不会丢失此信息。 RCAN-ET 完成当前执行的接收后转移至停机模式。 RCAN-ET 在接收或发送信息过程中，需要一定时间才能转移至停机状态 （因为需等待发送 / 接收结束后转移至停机状态）。必须注意：不可在停机状态发送 / 接收信息。

RCAN-ET 为总线断开状态时，依据 MCR 寄存器的 bit6 及 bit14 的设定转移至停机状态。

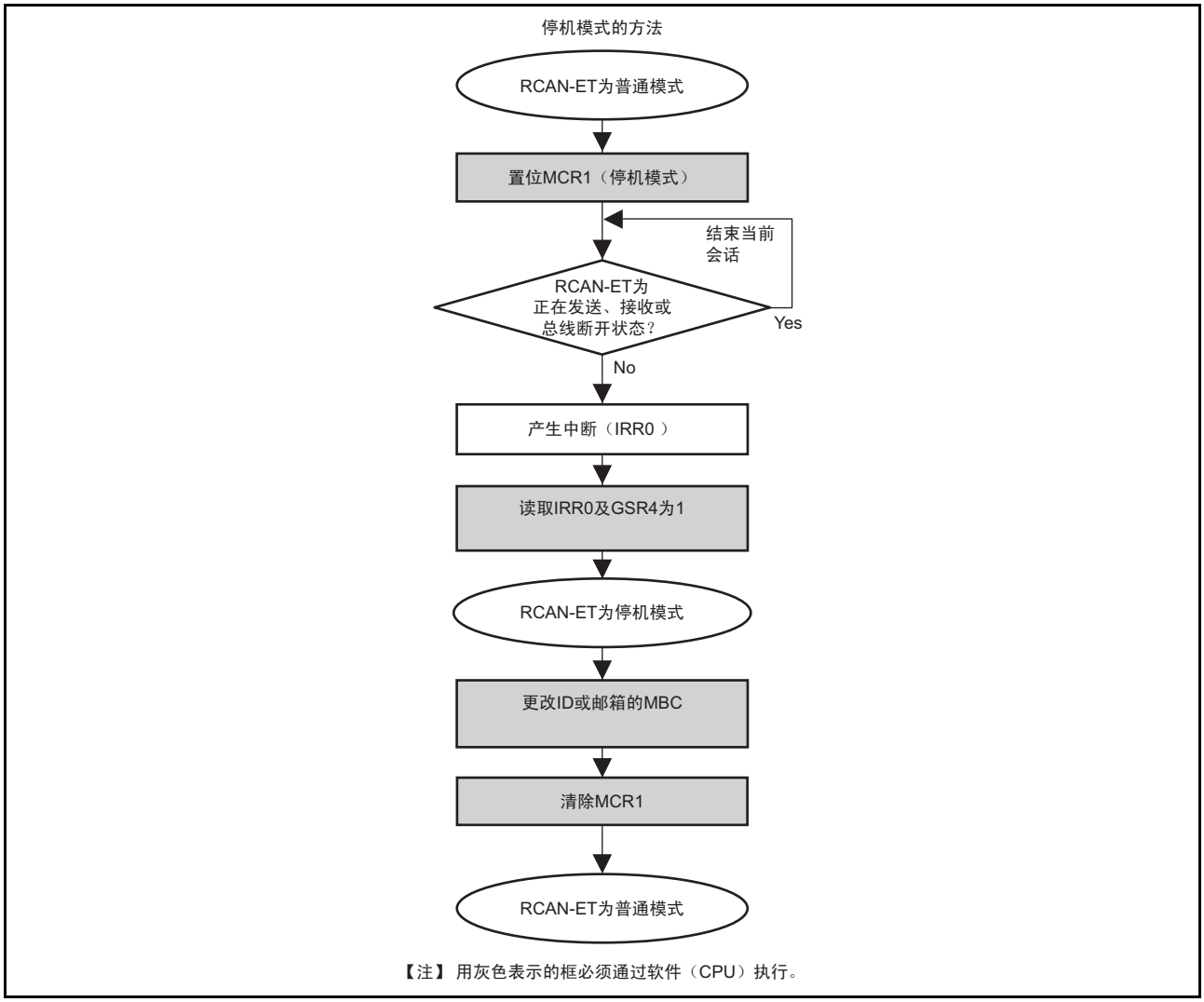


图 17.12　更改接收邮箱的 ID/ 从接收邮箱更改为发送邮箱

17.7　中断源

RCAN-ET 具有表 17.9 所示的中断源。除上电复位引起的复位处理中断 （IRR0）之外，可屏蔽所有中断源。屏蔽时使用邮箱中断屏蔽寄存器 0 （MBIMR0）及中断屏蔽寄存器 （IMR）。有关各中断请求的中断向量，详情参阅 “第 6 章 中断控制器 （INTC）”。

表 17.9　RCAN-ET 的中断源

通道	名称	中断源	中断标志	启动 DTC
0	ERS_0	错误被动 （TEC ≥ 128 或 REC ≥ 128）	IRR5	不可
		总线断开 （TEC ≥ 256） / 从总线断开返回	IRR6	
		错误警告 （TEC ≥ 96）	IRR3	
		错误警告 （REC ≥ 96）	IRR4	
	OVR_0	检测信息错误	IRR13*1	
		转移至复位 / 停机 /CAN 睡眠模式	IRR0	
		发送超载帧	IRR7	
		重写未读信息 （溢出）	IRR9	
		检测 CAN 睡眠模式的 CAN 总线运行	IRR12	
	RM0_0*2 RM1_0*2	接收数据帧	IRR1*3	可 *4
		接收远程帧	IRR2*3	
	SLE_0	发送信息 / 取消发送 （槽为空）	IRR8	不可
1*5	ERS_1	错误被动 （TEC ≥ 128 或 REC ≥ 128）	IRR5	不可
		总线断开 （TEC ≥ 256） / 从总线断开返回	IRR6	
		错误警告 （TEC ≥ 96）	IRR3	
		错误警告 （REC ≥ 96）	IRR4	
	OVR_1	检测信息错误	IRR13*1	
		转移至复位 / 停机 /CAN 睡眠模式	IRR0	
		发送超载帧	IRR7	
		重写未读信息 （溢出）	IRR9	
		检测 CAN 睡眠的 CAN 总线运行	IRR12	
	RM0_1*2 RM1_1*2	接收数据帧	IRR1*3	可 *4
		接收远程帧	IRR2*3	
	SLE_1	发送信息 / 取消发送 （槽为空）	IRR8	不可

【注】 *1　仅在测试模式有效。

*2　RM0_0 为邮箱 0 的远程帧接收标志 （RFPR0[0]）或数据帧接收标志 （RXPR0[0]）引起的中断， RM1_0 为邮箱 n （n=1 ～ 15）的远程帧接收标志 （RFPR0[n]）或数据帧接收标志 （RXPR0[n]）引起的中断。

*3　IRR1 为邮箱 0 ～ 15 的数据帧接收标志， IRR2 为邮箱 0 ～ 15 的远程帧接收标志。

*4　RM0_0、 RM0_1 中断仅可启动 DTC。

*5　仅适用于 SH7142

17.8　DTC 接口

如果在 RCAN-ET 的邮箱 0 接收信息，则可启动 DTC。设定 DTC 启动，如果结束 DTC 传送，则可自动清除 RXPR0 及 RFPR0 寄存器的标志。此时， RCAN-ET 的接收中断不会对 CPU 产生中断。 DTC 的传送流程图如图 17.13 所示。

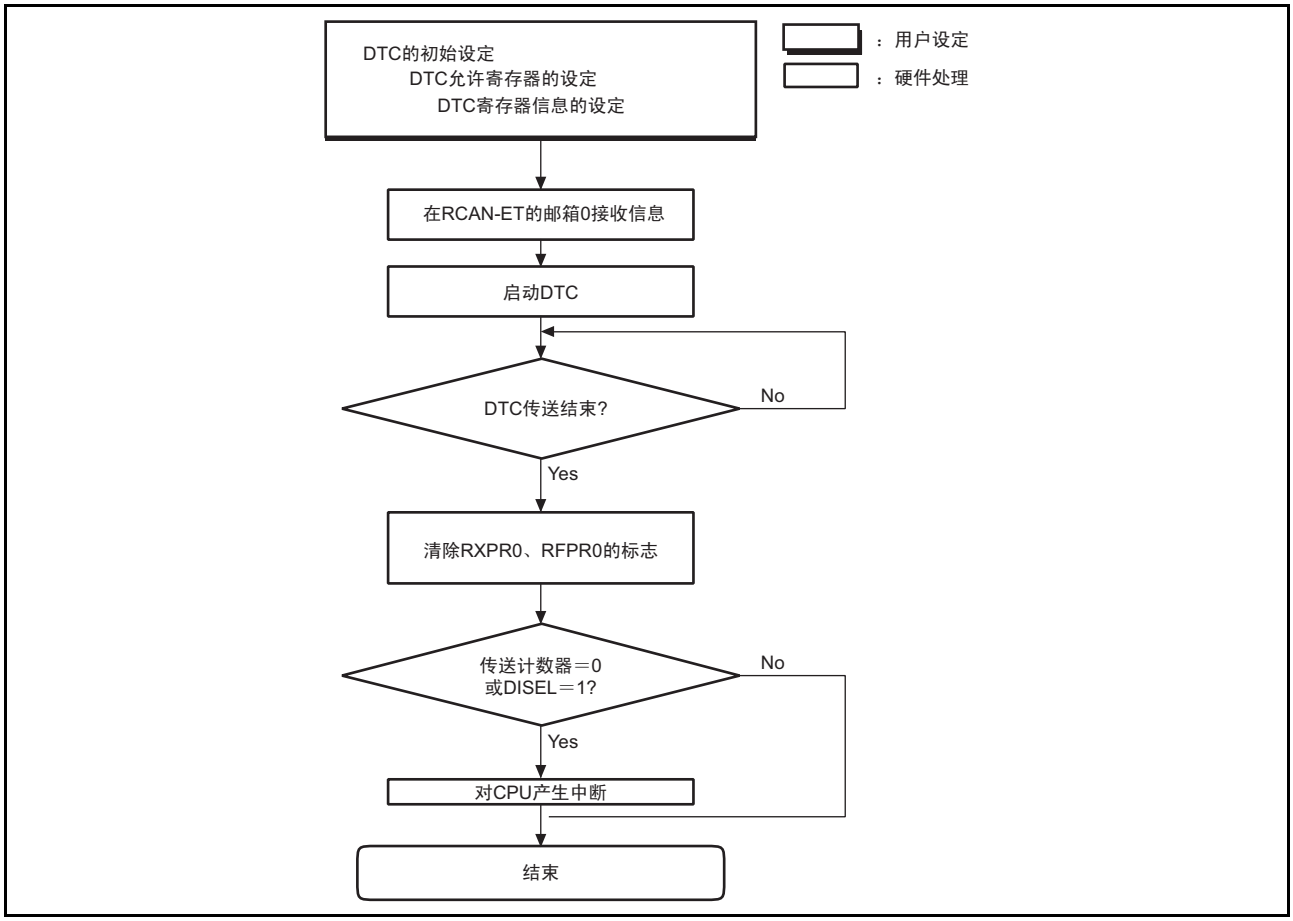


图 17.13　DTC 的传送流程图

17.9　CAN 总线接口

为了连接本 LSI 与 CAN 总线，需要总线收发器 IC。收发器 IC 推荐使用瑞萨 HA13721，使用除 HA13721 之外的产品时，必须使用与 HA13721 完全兼容的产品。连接例子如图 17.14 所示。

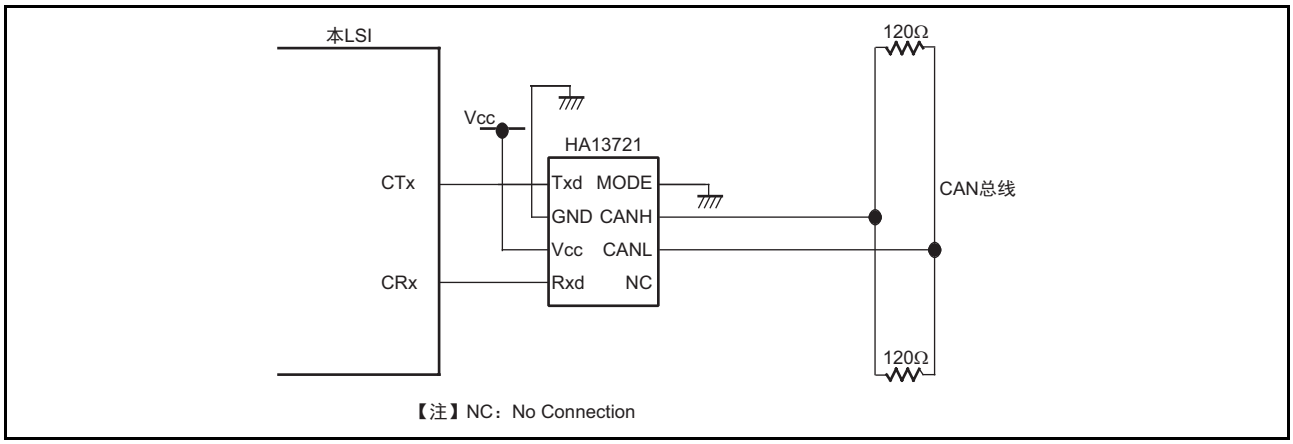


图 17.14　使用 HA13721 的高速 CAN 接口

17.10　使用时的注意事项

17.10.1　模块停止模式

通过待机控制寄存器 3 （STBCR3），可设定运行 / 停止 RCAN-ET 时钟。为初始值时， RCAN-ET 时钟停止，解除模块停止模式后， RCAN-ET 时钟运行。

17.10.2　复位

RCAN-ET 的复位有硬件复位及软件复位。

- 硬件复位

通过上电复位、硬件待机、模块停止、软件待机初始化 RCAN-ET。

- 软件复位

通过主控寄存器 （MCR）的 MCR0 位，初始化除 MCR0 位之外的寄存器及 CAN 通信功能。

复位时，中断请求寄存器 （IRR）的 IRR0 位通过初始化置位，因此必须在复位顺序所示的结构模式清除此位。

除邮箱的信息控制字段 1 （CONTROL1）之外的区域均由 RAM 构成，因此不可通过复位执行初始化。上电复位后，必须在复位顺序所示的结构模式初始设定所有邮箱。

17.10.3　CAN 睡眠模式

CAN 睡眠模式下，在模块内部停止主要部分的时钟，因此，在 CAN 睡眠模式，不得存取除 MCR、 GSR、 IRR、 IMR 之外的寄存器。

17.10.4　寄存器存取

使用 RCAN-ET 内部的 CAN 通信功能，将 CAN 总线接收帧保存至邮箱时，如果存取邮箱区域，则要等待 0 ～ 5 个外围总线周期。

17.10.5　中断

如表 17.9 所示，邮箱 0 接收中断可启动 DTC。将邮箱 0 接收中断作为启动源，如果选择清除 DTC 传送时的中断源，则必须使用块传送模式等，读取到邮箱 0 的信息控制字段 1 （CONTROL1）为止。

第 18 章　引脚功能控制器 （PFC）

引脚功能控制器 （PFC）由选择多路复用引脚功能及其输入 / 输出方向的寄存器构成。本 LSI 的多路复用引脚如表 18.1 ～表 18.4 所示。

各运行模式的引脚功能一览表如表 18.5 所示。

表 18.1　多路复用引脚一览表 （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）	功能 4 （相关模块）	功能 5 （相关模块）
A	PA0 输入 / 输出 （端口）	A0 输出 （BSC）	$\overline{\text{POE0}}$ 输入 （POE）	RXD0 输入 （SCI）	—
	PA1 输入 / 输出 （端口）	A1 输出 （BSC）	$\overline{\text{POE1}}$ 输入 （POE）	TXD0 输出 （SCI）	—
	PA2 输入 / 输出 （端口）	A2 输出 （BSC）	IRQ0 输入 （INTC）	$\overline{\text{POE2}}$ 输入 （POE）	SCK0 输入 / 输出 （SCI）
	PA3 输入 / 输出 （端口）	A3 输出 （BSC）	IRQ1 输入 （INTC）	RXD1 输入 （SCI）	—
	PA4 输入 / 输出 （端口）	A4 输出 （BSC）	IRQ2 输入 （INTC）	TXD1 输出 （SCI）	—
	PA5 输入 / 输出 （端口）	A5 输出 （BSC）	IRQ3 输入 （INTC）	SCK1 输入 / 输出 （SCI）	
	PA6 输入 / 输出 （端口）	$\overline{\text{RD}}$ 输出 （BSC）	$\overline{\text{UBCTRG}}$ 输出 （UBC）	TCLKA 输入 （MTU2）	$\overline{\text{POE4}}$ 输入 （POE）
	PA7 输入 / 输出 （端口）	TCLKB 输入 （MTU2）	$\overline{\text{POE5}}$ 输入 （POE）	SCK2 输入 / 输出 （SCI）	—
	PA8 输入 / 输出 （端口）	$\overline{\text{WRL}}$ 输出 （BSC）	TCLKC 输入 （MTU2）	$\overline{\text{POE6}}$ 输入 （POE）	RXD2 输入 （SCI）
	PA9 输入 / 输出 （端口）	$\overline{\text{WAIT}}$ 输入 （BSC）	TCLKD 输入 （MTU2）	$\overline{\text{POE8}}$ 输入 （POE）	TXD2 输出 （SCI）
	PA10 输入 / 输出 （端口）	A6 输出 （BSC）	RXD0 输入 （SCI）	—	—
	PA11 输入 / 输出 （端口）	A7 输出 （BSC）	TXD0 输出 （SCI）	$\overline{\text{ADTRG}}$ 输入 （A/D）	—
	PA12 输入 / 输出 （端口）	A8 输出 （BSC）	SCK0 输入 / 输出 （SCI）	$\overline{\text{SCS}}$ 输入 / 输出 （SSU）	—
	PA13 输入 / 输出 （端口）	A9 输出 （BSC）	SCK1 输入 / 输出 （SCI）	SSCK 输入 / 输出 （SSU）	—
	PA14 输入 / 输出 （端口）	A10 输出 （BSC）	RXD1 输入 （SCI）	SSI 输入 / 输出 （SSU）	—
	PA15 输入 / 输出 （端口）	CK 输出 （CPG）	TXD1 输出 （SCI）	SSO 输入 / 输出 （SSU）	—

表 18.2　多路复用引脚一览表 （端口 B）

端口	功能 1 （相关模块）	功能 2 （相关模块）	功能 3 （相关模块）	功能 4 （相关模块）	功能 5 （相关模块）
B	PB0 输入 / 输出 （端口）	$\overline{\text{BACK}}$ 输出 （BSC）	TIC5WS 输入 （MTU2S）	CTx1 输出 （RCAN-ET） *	—
	PB1 输入 / 输出 （端口）	$\overline{\text{BREQ}}$ 输入 （BSC）	CRx1 输入 （RCAN-ET） *	—	—
	PB2 输入 / 输出 （端口）	A16 输出 （BSC）	IRQ0 输入 （INTC）	$\overline{\text{POE0}}$ 输入 （POE）	TIC5VS 输入 （MTU2S）
	PB3 输入 / 输出 （端口）	A17 输出 （BSC）	IRQ1 输入 （INTC）	$\overline{\text{POE1}}$ 输入 （POE）	—
	PB4 输入 / 输出 （端口）	A18 输出 （BSC）	IRQ2 输入 （INTC）	$\overline{\text{POE4}}$ 输入 （POE）	TIC5US 输入 （MTU2S）
	PB5 输入 / 输出 （端口）	A19 输出 （BSC）	IRQ3 输入 （INTC）	$\overline{\text{POE5}}$ 输入 （POE）	—
	PB6 输入 / 输出 （端口）	$\overline{\text{WAIT}}$ 输入 （BSC）	CTx0 输出 （RCAN-ET）	—	—
	PB7 输入 / 输出 （端口）	$\overline{\text{CS1}}$ 输出 （BSC）	CRx0 输入 （RCAN-ET）	—	—

【注】 *　仅适用于 SH7142

表 18.3　多路复用引脚一览表 （端口 D）

端口	功能 1 （相关模块）	功能 2 （相关模块）	功能 3 （相关模块）	功能 4 （相关模块）	功能 5 （相关模块）
D	PD0 输入 / 输出 （端口）	D0 输入 / 输出 （BSC）	RXD0 输入 （SCI）	AUDATA0 输出 （AUD）	—
	PD1 输入 / 输出 （端口）	D1 输入 / 输出 （BSC）	TXD0 输出 （SCI）	AUDATA1 输出 （AUD）	—
	PD2 输入 / 输出 （端口）	D2 输入 / 输出 （BSC）	SCK0 输入 / 输出 （SCI）	AUDATA2 输出 （AUD）	—
	PD3 输入 / 输出 （端口）	D3 输入 / 输出 （BSC）	RXD1 输入 （SCI）	AUDATA3 输出 （AUD	—
	PD4 输入 / 输出 （端口）	D4 输入 / 输出 （BSC）	TXD1 输出 （SCI）	$\overline{\text{AUDRST}}$ 输入 （AUD）	—
	PD5 输入 / 输出 （端口）	D5 输入 / 输出 （BSC）	SCK1 输入 / 输出 （SCI）	AUDMD 输入 （AUD）	—
	PD6 输入 / 输出 （端口）	D6 输入 / 输出 （BSC）	RXD2 输入 （SCI）	AUDCK 输出 （AUD）	—
	PD7 输入 / 输出 （端口）	D7 输入 / 输出 （BSC）	TXD2 输出 （SCI）	$\overline{\text{SCS}}$ 输入 / 输出 （SSU）	$\overline{\text{AUDSYNC}}$ 输出 （AUD）
	PD8 输入 / 输出 （端口）	SCK2 输入 / 输出 （SCI）	SSCK 输入 / 输出 （SSU）	—	—
	PD9 输入 / 输出 （端口）	SSI 输入 / 输出 （SSU）	—	—	—
	PD10 输入 / 输出 （端口）	SSO 输入 / 输出 （SSU）	—	—	—

表 18.4　多路复用引脚一览表 （端口 E）

端口	功能 1 （相关模块）	功能 2 （相关模块）	功能 3 （相关模块）	功能 4 （相关模块）	功能 5 （相关模块）
E	PE0 输入 / 输出 （端口）	TIOC0A 输入 / 输出 （MTU2）	—	—	—
	PE1 输入 / 输出 （端口）	TIOC0B 输入 / 输出 （MTU2）	RXD0 输入 （SCI）	—	—
	PE2 输入 / 输出 （端口）	TIOC0C 输入 / 输出 （MTU2）	TXD0 输出 （SCI）	—	—
	PE3 输入 / 输出 （端口）	TIOC0D 输入 / 输出 （MTU2）	SCK0 输入 / 输出 （SCI）	—	—
	PE4 输入 / 输出 （端口）	A11 输出 （BSC）	TIOC1A 输入 / 输出 （MTU2）	RXD1 输入 （SCI）	—
	PE5 输入 / 输出 （端口）	A12 输出 （BSC）	TIOC1B 输入 / 输出 （MTU2）	TXD1 输出 （SCI）	—
	PE6 输入 / 输出 （端口）	A13 输出 （BSC）	TIOC2A 输入 / 输出 （MTU2）	SCK1 输入 / 输出 （SCI）	—
	PE7 输入 / 输出 （端口）	A14 输出 （BSC）	TIOC2B 输入 / 输出 （MTU2）	—	—
	PE8 输入 / 输出 （端口）	A15 输出 （BSC）	TIOC3A 输入 / 输出 （MTU2）	—	—
	PE9 输入 / 输出 （端口）	TIOC3B 输入 / 输出 （MTU2）	—	—	—
	PE10 输入 / 输出 （端口）	$\overline{\text{CS0}}$ 输出 （BSC）	TIOC3C 输入 / 输出 （MTU2）	—	—
	PE11 输入 / 输出 （端口）	TIOC3D 输入 / 输出 （MTU2）	—	—	—
	PE12 输入 / 输出 （端口）	TIOC4A 输入 / 输出 （MTU2）	—	—	—
	PE13 输入 / 输出 （端口）	TIOC4B 输入 / 输出 （MTU2）	$\overline{\text{MRES}}$ 输入 （INTC）	—	—
	PE14 输入 / 输出 （端口）	TIOC4C 输入 / 输出 （MTU2）	—	—	—
	PE15 输入 / 输出 （端口）	TIOC4D 输入 / 输出 （MTU2）	$\overline{\text{IRQOUT}}$ 输出 （INTC）	—	—
	PE16 输入 / 输出 （端口）	$\overline{\text{WAIT}}$ 输入 （BSC）	TIOC3BS 输入 / 输出 （MTU2S）	—	—
	PE17 输入 / 输出 （端口）	$\overline{\text{CS0}}$ 输出 （BSC）	TIOC3DS 输入 / 输出 （MTU2S）	—	—
	PE18 输入 / 输出 （端口）	$\overline{\text{CS1}}$ 输出 （BSC）	TIOC4AS 输入 / 输出 （MTU2S）	—	—
	PE19 输入 / 输出 （端口）	$\overline{\text{RD}}$ 输出 （BSC）	TIOC4BS 输入 / 输出 （MTU2S）	—	—
	PE20 输入 / 输出 （端口）	TIOC4CS 输入 / 输出 （MTU2S）	—	—	—
	PE21 输入 / 输出 （端口）	$\overline{\text{WRL}}$ 输出 （BSC）	TIOC4DS 输入 / 输出 （MTU2S）	—	—

表 18.5　各运行模式的引脚功能一览表 （1）

引脚编号	引脚名称	
	内部 ROM 无效 （MCU 模式 0）	
	初始功能	PFC 可设定的功能
48、 3	Vcc	Vcc
50、 1	Vss	Vss
11、 36、 57、 76	PVcc	PVcc
14、 39、 64	PVss	PVss
16、 59	V_{CL}	V_{CL}
98	AVcc	AVcc
79	AVss	AVss
88	AV_{refh}	AV_{refh}
93	AV_{refl}	AV_{refl}
75	PLLVss	PLLVss
72	EXTAL	EXTAL
71	XTAL	XTAL
78	MD0	MD0
77	MD1	MD1
74	FWE	FWE
70	$\overline{RES}$	$\overline{RES}$
100	$\overline{WDTOVF}$	$\overline{WDTOVF}$
73	NMI	NMI
99	$\overline{HSTBY}$	$\overline{HSTBY}$
69	A0	PA0/A0/$\overline{POE0}$/RXD0
68	A1	PA1/A1/$\overline{POE1}$/TXD0
67	A2	PA2/A2/IRQ0/$\overline{POE2}$/SCK0
66	A3	PA3/A3/IRQ1/RXD1
65	A4	PA4/A4/IRQ2/TXD1
63	A5	PA5/A5/IRQ3/SCK1
62	$\overline{RD}$	PA6/$\overline{RD}$/$\overline{UBCTRG}$/TCLKA/$\overline{POE4}$
61	PA7	PA7/TCLKB/$\overline{POE5}$/SCK2
60	$\overline{WRL}$	PA8/$\overline{WRL}$/TCLKC/$\overline{POE6}$/RXD2
58	PA9	PA9/$\overline{WAIT}$/TCLKD/$\overline{POE8}$/TXD2
56	A6	PA10/A6/RXD0
55	A7	PA11/A7/TXD0/$\overline{ADTRG}$
54	A8	PA12/A8/SCK0/$\overline{SCS}$
53	A9	PA13/A9/SCK1/SSCK
52	A10	PA14/A10/RXD1/SSI
51	CK	PA15/CK/TXD1/SSO

引脚编号	引脚名称	
	内部 ROM 无效 （MCU 模式 0）	
	初始功能	PFC 可设定的功能
49	PB0	PB0/$\overline{\text{BACK}}$/TIC5WS/CTx1*
47	PB1	PB1/$\overline{\text{BREQ}}$/CRx1*
46	A16	PB2/A16/IRQ0/$\overline{\text{POE0}}$/TIC5VS
45	A17	PB3/A17/IRQ1/$\overline{\text{POE1}}$
44	PB4	PB4/A18/IRQ2/$\overline{\text{POE4}}$/TIC5US
43	PB5	PB5/A19/IRQ3/$\overline{\text{POE5}}$
42	PB6	PB6/$\overline{\text{WAIT}}$/CTx0
41	PB7	PB7/$\overline{\text{CS1}}$/CRx0
40	D0	PD0/D0/RXD0/AUDATA0
38	D1	PD1/D1/TXD0/AUDATA1
37	D2	PD2/D2/SCK0/AUDATA2
35	D3	PD3/D3/RXD1/AUDATA3
34	D4	PD4/D4/TXD1/$\overline{\text{AUDRST}}$
33	D5	PD5/D5/SCK1/AUDMD
32	D6	PD6/D6/RXD2/AUDCK
31	D7	PD7/D7/TXD2/$\overline{\text{SCS}}$/$\overline{\text{AUDSYNC}}$
30	PD8	PD8/SCK2/SSCK
29	PD9	PD9/SSI
28	PD10	PD10/SSO
27	PE0	PE0/TIOC0A
26	PE1	PE1/TIOC0B/RXD0
25	PE2	PE2/TIOC0C/TXD0
24	PE3	PE3/TIOC0D/SCK0
23	A11	PE4/A11/TIOC1A/RXD1
22	A12	PE5/A12/TIOC1B/TXD1
21	A13	PE6/A13/TIOC2A/SCK1
20	A14	PE7/A14/TIOC2B
19	A15	PE8/A15/TIOC3A
17	PE9	PE9/TIOC3B
18	$\overline{\text{CS0}}$	PE10/$\overline{\text{CS0}}$/TIOC3C
15	PE11	PE11/TIOC3D
13	PE12	PE12/TIOC4A
12	PE13	PE13/TIOC4B/$\overline{\text{MRES}}$
10	PE14	PE14/TIOC4C
9	PE15	PE15/TIOC4D/$\overline{\text{IRQOUT}}$

引脚编号	引脚名称	
	内部 ROM 无效 （MCU 模式 0）	
	初始功能	PFC 可设定的功能
8	PE16	PE16/$\overline{\text{WAIT}}$/TIOC3BS
7	PE17	PE17/$\overline{\text{CS0}}$/TIOC3DS
6	PE18	PE18/$\overline{\text{CS1}}$/TIOC4AS
5	PE19	PE19/$\overline{\text{RD}}$/TIOC4BS
4	PE20	PE20/TIOC4CS
2	PE21	PE21/$\overline{\text{WRL}}$/TIOC4DS
97	AN0	AN0
96	AN1	AN1
95	AN2	AN2
94	AN3	AN3
92	AN4	AN4
91	AN5	AN5
90	AN6	AN6
89	AN7	AN7
87	AN8	AN8
86	AN9	AN9
85	AN10	AN10
84	AN11	AN11
83	AN12	AN12
82	AN13	AN13
81	AN14	AN14
80	AN15	AN15

【注】 *　仅适用于 SH7142

表 18.5　各运行模式的引脚功能一览表 （2）

引脚编号	引脚名称			
	内部 ROM 有效 （MCU 模式 2）		单芯片模式 （MCU 模式 3）	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
48、3	Vcc	Vcc	Vcc	Vcc
50、1	Vss	Vss	Vss	Vss
11、36、57、76	PVcc	PVcc	PVcc	PVcc
14、39、64	PVss	PVss	PVss	PVss
16、59	V_{CL}	V_{CL}	V_{CL}	V_{CL}
98	AVcc	AVcc	AVcc	AVcc
79	AVss	AVss	AVss	AVss
88	AV_{refh}	AV_{refh}	AV_{refh}	AV_{refh}
93	AV_{refl}	AV_{refl}	AV_{refl}	AV_{refl}
75	PLLVss	PLLVss	PLLVss	PLLVss
72	EXTAL	EXTAL	EXTAL	EXTAL
71	XTAL	XTAL	XTAL	XTAL
78	MD0	MD0	MD0	MD0
77	MD1	MD1	MD1	MD1
74	FWE	FWE	FWE	FWE
70	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$
100	$\overline{WDTOVF}$	$\overline{WDTOVF}$	$\overline{WDTOVF}$	$\overline{WDTOVF}$
73	NMI	NMI	NMI	NMI
99	$\overline{HSLBY}$	$\overline{HSLBY}$	$\overline{HSLBY}$	$\overline{HSLBY}$
69	PA0	PA0/A0/$\overline{POE0}$/RXD0	PA0	PA0/A0/$\overline{POE0}$/RXD0
68	PA1	PA1/A1/$\overline{POE1}$/TXD0	PA1	PA1/A1/$\overline{POE1}$/TXD0
67	PA2	PA2/A2/IRQ0/$\overline{POE2}$/SCK0	PA2	PA2/A2/IRQ0/$\overline{POE2}$/SCK0
66	PA3	PA3/A3/IRQ1/RXD1	PA3	PA3/A3/IRQ1/RXD1
65	PA4	PA4/A4/IRQ2/TXD1	PA4	PA4/A4/IRQ2/TXD1
63	PA5	PA5/A5/IRQ3/SCK1	PA5	PA5/A5/IRQ3/SCK1
62	PA6	PA6/$\overline{RD}$/$\overline{UBCTRG}$/TCLKA/$\overline{POE4}$	PA6	PA6/$\overline{RD}$/$\overline{UBCTRG}$/TCLKA/$\overline{POE4}$
61	PA7	PA7/TCLKB/$\overline{POE5}$/SCK2	PA7	PA7/TCLKB/$\overline{POE5}$/SCK2
60	PA8	PA8/$\overline{WRL}$/TCLKC/$\overline{POE6}$/RXD2	PA8	PA8/$\overline{WRL}$/TCLKC/$\overline{POE6}$/RXD2
58	PA9	PA9/$\overline{WAIT}$/TCLKD/$\overline{POE8}$/TXD2	PA9	PA9/$\overline{WAIT}$/TCLKD/$\overline{POE8}$/TXD2
56	PA10	PA10/A6/RXD0	PA10	PA10/A6/RXD0
55	PA11	PA11/A7/TXD0/$\overline{ADTRG}$	PA11	PA11/A7/TXD0/$\overline{ADTRG}$
54	PA12	PA12/A8/SCK0/$\overline{SCS}$	PA12	PA12/A8/SCK0/$\overline{SCS}$
53	PA13	PA13/A9/SCK1/SSCK	PA13	PA13/A9/SCK1/SSCK

引脚编号	引脚名称			
	内部 ROM 有效 （MCU 模式 2）		单芯片模式 （MCU 模式 3）	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
52	PA14	PA14/A10/RXD1/SSI	PA14	PA14/A10/RXD1/SSI
51	CK	PA15/CK/TXD1/SSO	PA15	PA15/CK/TXD1/SSO
49	PB0	PB0/$\overline{\text{BACK}}$/TIC5WS/CTx1*	PB0	PB0/$\overline{\text{BACK}}$/TIC5WS/CTx1*
47	PB1	PB1/$\overline{\text{BREQ}}$/CRx1*	PB1	PB1/$\overline{\text{BREQ}}$/CRx1*
46	PB2	PB2/A16/IRQ0/$\overline{\text{POE0}}$/TIC5VS	PB2	PB2/A16/IRQ0/$\overline{\text{POE0}}$/TIC5VS
45	PB3	PB3/A17/IRQ1/$\overline{\text{POE1}}$	PB3	PB3/A17/IRQ1/$\overline{\text{POE1}}$
44	PB4	PB4/A18/IRQ2/$\overline{\text{POE4}}$/TIC5US	PB4	PB4/A18/IRQ2/$\overline{\text{POE4}}$/TIC5US
43	PB5	PB5/A19/IRQ3/$\overline{\text{POE5}}$	PB5	PB5/A19/IRQ3/$\overline{\text{POE5}}$
42	PB6	PB6/$\overline{\text{WAIT}}$/CTx0	PB6	PB6/$\overline{\text{WAIT}}$/CTx0
41	PB7	PB7/$\overline{\text{CS1}}$/CRx0	PB7	PB7/$\overline{\text{CS1}}$/CRx0
40	PD0	PD0/D0/RXD0/AUDATA0	PD0	PD0/D0/RXD0/AUDATA0
38	PD1	PD1/D1/TXD0/AUDATA1	PD1	PD1/D1/TXD0/AUDATA1
37	PD2	PD2/D2/SCK0/AUDATA2	PD2	PD2/D2/SCK0/AUDATA2
35	PD3	PD3/D3/RXD1/AUDATA3	PD3	PD3/D3/RXD1/AUDATA3
34	PD4	PD4/D4/TXD1/$\overline{\text{AUDRST}}$	PD4	PD4/D4/TXD1/$\overline{\text{AUDRST}}$
33	PD5	PD5/D5/SCK1/AUDMD	PD5	PD5/D5/SCK1/AUDMD
32	PD6	PD6/D6/RXD2/AUDCK	PD6	PD6/D6/RXD2/AUDCK
31	PD7	PD7/D7/TXD2/$\overline{\text{SCS}}$/$\overline{\text{AUDSYNC}}$	PD7	PD7/D7/TXD2/$\overline{\text{SCS}}$/$\overline{\text{AUDSYNC}}$
30	PD8	PD8/SCK2/SSCK	PD8	PD8/SCK2/SSCK
29	PD9	PD9/SSI	PD9	PD9/SSI
28	PD10	PD10/SSO	PD10	PD10/SSO
27	PE0	PE0/TIOC0A	PE0	PE0/TIOC0A
26	PE1	PE1/TIOC0B/RXD0	PE1	PE1/TIOC0B/RXD0
25	PE2	PE2/TIOC0C/TXD0	PE2	PE2/TIOC0C/TXD0
24	PE3	PE3/TIOC0D/SCK0	PE3	PE3/TIOC0D/SCK0
23	PE4	PE4/A11/TIOC1A/RXD1	PE4	PE4/A11/TIOC1A/RXD1
22	PE5	PE5/A12/TIOC1B/TXD1	PE5	PE5/A12/TIOC1B/TXD1
21	PE6	PE6/A13/TIOC2A/SCK1	PE6	PE6/A13/TIOC2A/SCK1
20	PE7	PE7/A14/TIOC2B	PE7	PE7/A14/TIOC2B
19	PE8	PE8/A15/TIOC3A	PE8	PE8/A15/TIOC3A
17	PE9	PE9/TIOC3B	PE9	PE9/TIOC3B
18	PE10	PE10/$\overline{\text{CS0}}$/TIOC3C	PE10	PE10/$\overline{\text{CS0}}$/TIOC3C
15	PE11	PE11/TIOC3D	PE11	PE11/TIOC3D
13	PE12	PE12/TIOC4A	PE12	PE12/TIOC4A
12	PE13	PE13/TIOC4B/$\overline{\text{MRES}}$	PE13	PE13/TIOC4B/$\overline{\text{MRES}}$

引脚编号	引脚名称			
	内部 ROM 有效 （MCU 模式 2）		单芯片模式 （MCU 模式 3）	
	初始功能	PFC 可设定的功能	初始功能	PFC 可设定的功能
10	PE14	PE14/TIOC4C	PE14	PE14/TIOC4C
9	PE15	PE15/TIOC4D/$\overline{\text{IRQOUT}}$	PE15	PE15/TIOC4D/$\overline{\text{IRQOUT}}$
8	PE16	PE16/$\overline{\text{WAIT}}$/TIOC3BS	PE16	PE16/$\overline{\text{WAIT}}$/TIOC3BS
7	PE17	PE17/$\overline{\text{CS0}}$/TIOC3DS	PE17	PE17/$\overline{\text{CS0}}$/TIOC3DS
6	PE18	PE18/$\overline{\text{CS1}}$/TIOC4AS	PE18	PE18/$\overline{\text{CS1}}$/TIOC4AS
5	PE19	PE19/$\overline{\text{RD}}$/TIOC4BS	PE19	PE19/$\overline{\text{RD}}$/TIOC4BS
4	PE20	PE20/TIOC4CS	PE20	PE20/TIOC4CS
2	PE21	PE21/$\overline{\text{WRL}}$/TIOC4DS	PE21	PE21/$\overline{\text{WRL}}$/TIOC4DS
97	AN0	AN0	AN0	AN0
96	AN1	AN1	AN1	AN1
95	AN2	AN2	AN2	AN2
94	AN3	AN3	AN3	AN3
92	AN4	AN4	AN4	AN4
91	AN5	AN5	AN5	AN5
90	AN6	AN6	AN6	AN6
89	AN7	AN7	AN7	AN7
87	AN8	AN8	AN8	AN8
86	AN9	AN9	AN9	AN9
85	AN10	AN10	AN10	AN10
84	AN11	AN11	AN11	AN11
83	AN12	AN12	AN12	AN12
82	AN13	AN13	AN13	AN13
81	AN14	AN14	AN14	AN14
80	AN15	AN15	AN15	AN15

【注】 *　仅适用于 SH7142

18.1　寄存器说明

PFC 具有以下寄存器。有关寄存器的地址及各处理状态的寄存器状态，详情参阅 “第 24 章 寄存器一览表”。

表 18.6　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/IO 寄存器 L	PAIORL	R/W	H'0000	H'FFFFD106	8、16
端口 A 控制寄存器 L4	PACRL4	R/W	H'0000*	H'FFFFD110	8、16、32
端口 A 控制寄存器 L3	PACRL3	R/W	H'0000*	H'FFFFD112	8、16
端口 A 控制寄存器 L2	PACRL2	R/W	H'0000*	H'FFFFD114	8、16、32
端口 A 控制寄存器 L1	PACRL1	R/W	H'0000*	H'FFFFD116	8、16
端口 B/IO 寄存器 L	PBIORL	R/W	H'0000	H'FFFFD186	8、16
端口 B 控制寄存器 L2	PBCRL2	R/W	H'0000	H'FFFFD194	8、16、32
端口 B 控制寄存器 L1	PBCRL1	R/W	H'0000*	H'FFFFD196	8、16
端口 D/IO 寄存器 L	PDIORL	R/W	H'0000	H'FFFFD286	8、16
端口 D 控制寄存器 L3	PDCRL3	R/W	H'0000	H'FFFFD292	8、16
端口 D 控制寄存器 L2	PDCRL2	R/W	H'0000*	H'FFFFD294	8、16、32
端口 D 控制寄存器 L1	PDCRL1	R/W	H'0000*	H'FFFFD296	8、16
端口 E/IO 寄存器 H	PEIORH	R/W	H'0000	H'FFFFD304	8、16、32
端口 E/IO 寄存器 L	PEIORL	R/W	H'0000	H'FFFFD306	8、16
端口 E 控制寄存器 H2	PECRH2	R/W	H'0000	H'FFFFD30C	8、16、32
端口 E 控制寄存器 H1	PECRH1	R/W	H'0000	H'FFFFD30E	8、16
端口 E 控制寄存器 L4	PECRL4	R/W	H'0000	H'FFFFD310	8、16、32
端口 E 控制寄存器 L3	PECRL3	R/W	H'0000*	H'FFFFD312	8、16
端口 E 控制寄存器 L2	PECRL2	R/W	H'0000*	H'FFFFD314	8、16、32
端口 E 控制寄存器 L1	PECRL1	R/W	H'0000	H'FFFFD316	8、16

【注】 * 在内部 ROM 有效 / 无效外部扩展模式或内部 ROM 无效外部扩展模式，初始值不同。详情参阅本章的各寄存器说明。

18.1.1　端口 A/IO 寄存器 L （PAIORL）

PAIORL 为可读取 / 写入的 16 位寄存器，选择端口 A 的引脚输入 / 输出方向。PA15IOR ～ PA0IOR 位分别对应 PA15 ～ PA0 引脚 （省略引脚名称中除端口之外的多路复用引脚名称）。PAIORL 在端口 A 的引脚功能为通用输入 / 输出 （PA15 ～ PA0）时有效，除此之外无效。

如果 PAIORL 的位置 1，则对应的引脚为输出引脚，置 0 则为输入引脚。

PAIORL 的初始值为 H'0000。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15 IOR	PA14 IOR	PA13 IOR	PA12 IOR	PA11 IOR	PA10 IOR	PA9 IOR	PA8 IOR	PA7 IOR	PA6 IOR	PA5 IOR	PA4 IOR	PA3 IOR	PA2 IOR	PA1 IOR	PA0 IOR
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

18.1.2　端口 A 控制寄存器 L1 ～ L4 （PACRL1 ～ PACRL4）

PACRL1 ～ PACRL4 分别为可读取 / 写入的 16 位寄存器，选择端口 A 的多路复用引脚功能。

(1)　端口 A 控制寄存器 L4 （PACRL4）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA15 MD2	PA15 MD1	PA15 MD0	-	PA14 MD2	PA14 MD1	PA14 MD0	-	PA13 MD2	PA13 MD1	PA13 MD0	-	PA12 MD2	PA12 MD1	PA12 MD0
初始值：	0	0	0	0*1	0	0*2	0	0	0	0*2	0	0	0	0*2	0	0
R/W：	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

【注】*1 在内部ROM有效 / 无效外部扩展模式，初始值为1。
*2 在内部ROM无效外部扩展模式，初始值为1。

位	位名称	初始值	R/W	说　明
15	—	0	R	保留位 读取值、写入值总是为 0。
14 13 12	PA15MD2 PA15MD1 PA15MD0	0 0 0*1	R/W R/W R/W	PA15 模式位 选择 PA15/CK/TXD1/SSO 引脚的功能。 000：PA15 输入 / 输出 （端口） 001：CK 输出 （CPG） *3 101：SSO 输入 / 输出 （SSU） 110：TXD1 输出 （SCI） 上述以外：禁止设定
11	—	0	R	保留位 读取值、写入值总是为 0。

位	位名称	初始值	R/W	说　明
10 9 8	PA14MD2 PA14MD1 PA14MD0	0*2 0 0	R/W R/W R/W	PA14 模式位 选择 PA14/A10/RXD1/SSI 引脚的功能。 000：PA14 输入 / 输出 （端口） 100：A10 输出 （BSC） *3 101：SSI 输入 / 输出 （SSU） 110：RXD1 输入 （SCI） 上述以外：禁止设定
7	—	0	R	保留位 读取值、写入值总是为 0。
6 5 4	PA13MD2 PA13MD1 PA13MD0	0*2 0 0	R/W R/W R/W	PA13 模式位 选择 PA13/A9/SCK1/SSCK 引脚的功能。 000：PA13 输入 / 输出 （端口） 100：A9 输出 （BSC） *3 101：SSCK 输入 / 输出 （SSU） 110：SCK1 输入 / 输出 （SCI） 上述以外：禁止设定
3	—	0	R	保留位 读取值、写入值总是为 0。
2 1 0	PA12MD2 PA12MD1 PA12MD0	0*2 0 0	R/W R/W R/W	PA12 模式位 选择 PA12/A8/SCK0/$\overline{\text{SCS}}$ 引脚的功能。 000：PA12 输入 / 输出 （端口） 100：A8 输出 （BSC） *3 101：$\overline{\text{SCS}}$ 输入 / 输出 （SSU） 110：SCK0 输入 / 输出 （SCI） 上述以外：禁止设定

【注】 *1　在内部 ROM 有效 / 无效外部扩展模式，初始值为 1。

*2　在内部 ROM 无效外部扩展模式，初始值为 1。

*3　仅在内部 ROM 有效 / 无效外部扩展模式有效，不得在单芯片模式设定。

(2)　端口 A 控制寄存器 L3 （PACRL3）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA11 MD2	PA11 MD1	PA11 MD0	-	PA10 MD2	PA10 MD1	PA10 MD0	-	PA9 MD2	PA9 MD1	PA9 MD0	-	PA8 MD2	PA8 MD1	PA8 MD0
初始值：	0	0*1	0	0	0	0*1	0	0	0	0	0	0	0	0*1	0	0
R/W：	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

【注】*1 在内部ROM无效外部扩展模式，初始值为1。

位	位名称	初始值	R/W	说　明
15	—	0	R	保留位 读取值、写入值总是为 0。
14 13 12	PA11MD2 PA11MD1 PA11MD0	0*1 0 0	R/W R/W R/W	PA11 模式位 选择 PA11/A7/TXD0/$\overline{\text{ADTRG}}$ 引脚的功能。 000：PA11 输入 / 输出 （端口） 010：$\overline{\text{ADTRG}}$ 输入 （A/D） 100：A7 输出 （BSC） *2 110：TXD0 输出 （SCI） 上述以外：禁止设定
11	—	0	R	保留位 读取值、写入值总是为 0。
10 9 8	PA10MD2 PA10MD1 PA10MD0	0*1 0 0	R/W R/W R/W	PA10 模式位 选择 PA10/A6/RXD0 引脚的功能。 000：PA10 输入 / 输出 （端口） 100：A6 输出 （BSC） *2 110：RXD0 输入 （SCI） 上述以外：禁止设定
7	—	0	R	保留位 读取值、写入值总是为 0。
6 5 4	PA9MD2 PA9MD1 PA9MD0	0 0 0	R/W R/W R/W	PA9 模式位 选择 PA9/$\overline{\text{WAIT}}$/TCLKD/$\overline{\text{POE8}}$/TXD2 引脚的功能。 000：PA9 输入 / 输出 （端口） 001：TCLKD 输入 （MTU2） 100：$\overline{\text{WAIT}}$ 输入 （BSC） *2 110：TXD2 输出 （SCI） 111：$\overline{\text{POE8}}$ 输入 （POE） 上述以外：禁止设定
3	—	0	R	保留位 读取值、写入值总是为 0。
2 1 0	PA8MD2 PA8MD1 PA8MD0	0*1 0 0	R/W R/W R/W	PA8 模式位 选择 PA8/$\overline{\text{WRL}}$/TCLKC/$\overline{\text{POE6}}$/RXD2 引脚的功能。 000：PA8 输入 / 输出 （端口） 001：TCLKC 输入 （MTU2） 100：$\overline{\text{WRL}}$ 输出 （BSC） *2 110：RXD2 输入 （SCI） 111：$\overline{\text{POE6}}$ 输入 （POE） 上述以外：禁止设定

【注】 *1　在内部 ROM 无效外部扩展模式，初始值为 1。

*2　仅在内部 ROM 有效 / 无效外部扩展模式有效，不得在单芯片模式设定。

(3)　端口 A 控制寄存器 L2 （PACRL2）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA7 MD2	PA7 MD1	PA7 MD0	-	PA6 MD2	PA6 MD1	PA6 MD0	-	PA5 MD2	PA5 MD1	PA5 MD0	-	PA4 MD2	PA4 MD1	PA4 MD0
初始值：	0	0	0	0	0	0	0*1	0*1	0	0*1	0	0	0	0*1	0	0
R/W：	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

【注】 *1 在内部ROM无效外部扩展模式，初始值为1。

位	位名称	初始值	R/W	说　明
15	—	0	R	保留位 读取值、写入值总是为 0。
14 13 12	PA7MD2 PA7MD1 PA7MD0	0 0 0	R/W R/W R/W	PA7 模式位 选择 PA7/TCLKB/$\overline{\text{POE5}}$/SCK2 引脚的功能。 000：PA7 输入 / 输出 （端口） 001：TCLKB 输入 （MTU2） 110：SCK2 输入 / 输出 （SCI） 111：$\overline{\text{POE5}}$ 输入 （POE） 上述以外：禁止设定
11		0	R	保留位 读取值、写入值总是为 0。
10 9 8	PA6MD2 PA6MD1 PA6MD0	0 0*1 0*1	R/W R/W R/W	PA6 模式位 选择 PA6/$\overline{\text{RD}}$/$\overline{\text{UBCTRG}}$/TCLKA/$\overline{\text{POE4}}$ 引脚的功能。 000：PA6 输入 / 输出 （端口） 001：TCLKA 输入 （MTU2） 011：$\overline{\text{RD}}$ 输出 （BSC） *2 101：$\overline{\text{UBCTRG}}$ 输出 （UBC） 111：$\overline{\text{POE4}}$ 输入 （POE） 上述以外：禁止设定
7	—	0	R	保留位 读取值、写入值总是为 0。
6 5 4	PA5MD2 PA5MD1 PA5MD0	0*1 0 0	R/W R/W R/W	PA5 模式位 选择 PA5/A5/IRQ3/SCK1 引脚的功能。 000：PA5 输入 / 输出 （端口） 001：SCK1 输入 / 输出 （SCI） 100：A5 输出 （BSC） *2 111：IRQ3 输入 （INTC） 上述以外：禁止设定
3	—	0	R	保留位 读取值、写入值总是为 0。
2 1 0	PA4MD2 PA4MD1 PA4MD0	0*1 0 0	R/W R/W R/W	PA4 模式位 选择 PA4/A4/IRQ2/TXD1 引脚的功能。 000：PA4 输入 / 输出 （端口） 001：TXD1 输出 （SCI） 100：A4 输出 （BSC） *2 111：IRQ2 输入 （INTC） 上述以外：禁止设定

【注】 *1　在内部 ROM 无效外部扩展模式，初始值为 1。

*2　仅在内部 ROM 有效 / 无效外部扩展模式有效，不得在单芯片模式设定。

(4)　端口 A 控制寄存器 L1 （PACRL1）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PA3 MD2	PA3 MD1	PA3 MD0	-	PA2 MD2	PA2 MD1	PA2 MD0	-	PA1 MD2	PA1 MD1	PA1 MD0	-	PA0 MD2	PA0 MD1	PA0 MD0
初始值：	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0
R/W：	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

【注】*1 在内部ROM无效外部扩展模式，初始值为1。

位	位名称	初始值	R/W	说　明
15	—	0	R	保留位 读取值、写入值总是为 0。
14 13 12	PA3MD2 PA3MD1 PA3MD0	0*1 0 0	R/W R/W R/W	PA3 模式位 选择 PA3/A3/IRQ1/RXD1 引脚的功能。 000：PA3 输入 / 输出 （端口） 001：RXD1 输入 （SCI） 100：A3 输出 （BSC） *2 111：IRQ1 输入 （INTC） 上述以外：禁止设定
11	—	0	R	保留位 读取值、写入值总是为 0。
10 9 8	PA2MD2 PA2MD1 PA2MD0	0*1 0 0	R/W R/W R/W	PA2 模式位 选择 PA2/A2/IRQ0/$\overline{\text{POE2}}$/SCK0 引脚的功能。 000：PA2 输入 / 输出 （端口） 001：SCK0 输入 / 输出 （SCI） 011：IRQ0 输入 （INTC） 100：A2 输出 （BSC） *2 111：$\overline{\text{POE2}}$ 输入 （POE） 上述以外：禁止设定
7	—	0	R	保留位 读取值、写入值总是为 0。
6 5 4	PA1MD2 PA1MD1 PA1MD0	0*1 0 0	R/W R/W R/W	PA1 模式位 选择 PA1/A1/$\overline{\text{POE1}}$/TXD0 引脚的功能。 000：PA1 输入 / 输出 （端口） 001：TXD0 输出 （SCI） 100：A1 输出 （BSC） *2 111：$\overline{\text{POE1}}$ 输入 （POE） 上述以外：禁止设定
3	—	0	R	保留位 读取值、写入值总是为 0。
2 1 0	PA0MD2 PA0MD1 PA0MD0	0*1 0 0	R/W R/W R/W	PA0 模式位 选择 PA0/A0/$\overline{\text{POE0}}$/RXD0 引脚功能。 000：PA0 输入 / 输出 （端口） 001：RXD0 输入 （SCI） 100：A0 输出 （BSC） *2 111：$\overline{\text{POE0}}$ 输入 （POE） 上述以外：禁止设定

【注】 *1　在内部 ROM 无效外部扩展模式，初始值为 1。

*2　仅在内部 ROM 有效 / 无效外部扩展模式有效，不得在单芯片模式设定。

18.1.3　端口 B/IO 寄存器 L （PBIORL）

PBIORL 为可读取 / 写入的 16 位寄存器，选择端口 B 的引脚输入 / 输出方向。 PB7IOR ～ PB0IOR 位对应 PB7 引脚～ PB0 引脚 （省略引脚名称中除端口之外的多路复用引脚名称）。 PBIORL 在端口 B 的引脚功能为通用输入 / 输出 （PB7 ～ PB0）时有效，除此之外无效。

如果 PBIORL 的位置 1，则对应的引脚为输出引脚，置 0 则为输入引脚。

PBIORL 的 bit15 ～ 8 为保留位，读取值、写入值总是为 0。

PBIORL 的初始值为 H'0000。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PB7 IOR	PB6 IOR	PB5 IOR	PB4 IOR	PB3 IOR	PB2 IOR	PB1 IOR	PB0 IOR
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

18.1.4　端口 B 控制寄存器 L1、 L2 （PBCRL1、 PBCRL2）

PBCRL1、 PBCRL2 分别为可读取 / 写入的 16 位寄存器，选择端口 B 的多路复用引脚功能。

(1)　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
	-	PB7 MD2	PB7 MD1	PB7 MD0	-	PB6 MD2	PB6 MD1	PB6 MD0	-	PB5 MD2	PB5 MD1	PB5 MD0	-	PB4 MD2	PB4 MD1	PB4 MD0
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名称	初始值	R/W	说　明
15	—	0	R	保留位 读取值、写入值总是为 0。
14 13 12	PB7MD2 PB7MD1 PB7MD0	0 0 0	R/W R/W R/W	PB7 模式位 选择 PB7/$\overline{\text{CS1}}$/CRx0 引脚的功能。 000：PB7 输入 / 输出 （端口） 101：$\overline{\text{CS1}}$ 输出 （BSC） * 110：CRx0 输入 （RCAN-ET） 上述以外：禁止设定
11	—	0	R	保留位 读取值、写入值总是为 0。
10 9 8	PB6MD2 PB6MD1 PB6MD0	0 0 0	R/W R/W R/W	PB6 模式位 选择 PB6/$\overline{\text{WAIT}}$/CTx0 引脚的功能。 000：PB6 输入 / 输出 （端口） 101：$\overline{\text{WAIT}}$ 输入 （BSC） * 110：CTx0 输出 （RCAN-ET） 上述以外：禁止设定

位	位名称	初始值	R/W	说　明
7	—	0	R	保留位 读取值、写入值总是为 0。
6 5 4	PB5MD2 PB5MD1 PB5MD0	0 0 0	R/W R/W R/W	PB5 模式位 选择 PB5/A19/IRQ3/$\overline{\text{POE5}}$ 引脚的功能。 000：PB5 输入 / 输出 （端口） 001：IRQ3 输入 （INTC） 101：A19 输出 （BSC） * 111：$\overline{\text{POE5}}$ 输入 （POE） 上述以外：禁止设定
3	—	0	R	保留位 读取值、写入值总是为 0。
2 1 0	PB4MD2 PB4MD1 PB4MD0	0 0 0	R/W R/W R/W	PB4 模式位 选择 PB4/A18/IRQ2/$\overline{\text{POE4}}$/TIC5US 引脚的功能。 000：PB4 输入 / 输出 （端口） 001：IRQ2 输入 （INTC） 011：TIC5US 输入 （MTU2S） 101：A18 输出 （BSC） * 111：$\overline{\text{POE4}}$ 输入 （POE） 上述以外：禁止设定

【注】 *　仅在内部 ROM 有效 / 无效外部扩展模式有效，不得在单芯片模式设定。

(2)　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
	-	PB3 MD2	PB3 MD1	PB3 MD0	-	PB2 MD2	PB2 MD1	PB2 MD0	-	PB1 MD2	PB1 MD1	PB1 MD0	-	PB0 MD2	PB0 MD1	PB0 MD0
初始值：	0	0*1	0	0*1	0	0*1	0	0*1	0	0	0	0	0	0	0	0
R/W：	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

【注】*1 在内部ROM无效外部扩展模式，初始值为1。

位	位名称	初始值	R/W	说　明
15	—	0	R	保留位 读取值、写入值总是为 0。
14 13 12	PB3MD2 PB3MD1 PB3MD0	0*1 0 0*1	R/W R/W R/W	PB3 模式位 选择 PB3/A17/IRQ1/$\overline{\text{POE1}}$ 引脚的功能。 000：PB3 输入 / 输出 （端口） 001：IRQ1 输入 （INTC） 010：$\overline{\text{POE1}}$ 输入 （POE） 101：A17 输出 （BSC） *3 上述以外：禁止设定
11	—	0	R	保留位 读取值、写入值总是为 0。
10 9 8	PB2MD2 PB2MD1 PB2MD0	0*1 0 0*1	R/W R/W R/W	PB2 模式位 选择 PB2/A16/IRQ0/$\overline{\text{POE0}}$/TIC5VS 引脚的功能。 000：PB2 输入 / 输出 （端口） 001：IRQ0 输入 （INTC） 010：$\overline{\text{POE0}}$ 输入 （POE） 011：TIC5VS 输入 （MTU2S） 101：A16 输出 （BSC） *3 上述以外：禁止设定
7	—	0	R	保留位 读取值、写入值总是为 0。
6 5 4	PB1MD2 PB1MD1 PB1MD0	0 0 0	R/W R/W R/W	PB1 模式位 选择 PB1/$\overline{\text{BREQ}}$ 引脚的功能。 000：PB1 输入 / 输出 （端口） 101：$\overline{\text{BREQ}}$ 输入 （BSC） *3 110：CRx1 输入 （RCAN-ET） *2 上述以外：禁止设定
3	—	0	R	保留位 读取值、写入值总是为 0。
2 1 0	PB0MD2 PB0MD1 PB0MD0	0 0 0	R/W R/W R/W	PB0 模式位 选择 PB0/$\overline{\text{BACK}}$/TIC5WS 引脚的功能。 000：PB0 输入 / 输出 （端口） 011：TIC5WS 输入 （MTU2S） 101：$\overline{\text{BACK}}$ 输出 （BSC） *3 110：CTx1 输出 （RCAN-ET） *2 上述以外：禁止设定

【注】 *1　在内部 ROM 无效外部扩展模式，初始值为 1。

*2　仅在 SH7142 才可设定。

*3　仅在内部 ROM 有效 / 无效外部扩展模式有效，不得在单芯片模式设定。

18.1.5　端口 D/IO 寄存器 L （PDIORL）

PDIORL 为可读取 / 写入的 16 位寄存器，选择端口 D 的引脚输入 / 输出方向。 PD10IOR ～ PD0IOR 位分别对应 PD10 ～ PD0 引脚 （省略引脚名称中除端口之外的多路复用引脚名称）。 PDIORL 在端口 D 的引脚功能为通用输入 / 输出 （PD10 ～ PD0）时有效，除此之外无效。

如果 PDIORL 的位置 1，则对应的引脚为输出引脚，置 0 则为输入引脚。

但 PDIORL 的 bit15 ～ 11 无效。

PDIORL 的初始值为 H'0000。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	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	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

18.1.6　端口 D 控制寄存器 L1 ～ L3 （PDCRL1 ～ PDCRL3）

PDCRL1 ～ PDCRL3 分别为可读取 / 写入的 16 位寄存器，选择端口 D 的多路复用引脚功能。

(1)　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
	-	-	-	-	-	PD10 MD2	PD10 MD1	PD10 MD0	-	PD9 MD2	PD9 MD1	PD9 MD0	-	PD8 MD2	PD8 MD1	PD8 MD0
初始值：	0	0	0	0	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/W	R/W	R/W	R	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 读取值、写入值总是为 0。
10 9 8	PD10MD2 PD10MD1 PD10MD0	0 0 0	R/W R/W R/W	PD10 模式位 选择 PD10/SSO 引脚的功能。 000：PD10 输入 / 输出 （端口） 101：SSO 输入 / 输出 （SSU） 上述以外：禁止设定
7	—	0	R	保留位 读取值、写入值总是为 0。
6 5 4	PD9MD2 PD9MD1 PD9MD0	0 0 0	R/W R/W R/W	PD9 模式位 选择 PD9/SSI 引脚的功能。 000：PD9 输入 / 输出 （端口） 101：SSI 输入 / 输出 （SSU） 上述以外：禁止设定
3	—	0	R	保留位 读取值、写入值总是为 0
2 1 0	PD8MD2 PD8MD1 PD8MD0	0 0 0	R/W R/W R/W	PD8 模式位 选择 PD8/SCK2/SSCK 引脚的功能。 000：PD8 输入 / 输出 （端口） 101：SSCK 输入 / 输出 （SSU） 110：SCK2 输入 / 输出 （SCI） 上述以外：禁止设定

(2)　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
	-	PD7 MD2	PD7 MD1	PD7 MD0	-	PD6 MD2	PD6 MD1	PD6 MD0	-	PD5 MD2	PD5 MD1	PD5 MD0	-	PD4 MD2	PD4 MD1	PD4 MD0
初始值：	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W：	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

【注】*1 在内部ROM无效外部扩展模式，初始值为1。

位	位名称	初始值	R/W	说　明
15	—	0	R	保留位 读取值、写入值总是为 0。
14 13 12	PD7MD2 PD7MD1 PD7MD0	0 0 0*1	R/W R/W R/W	PD7 模式位 选择 PD7/D7/TXD2/$\overline{\text{SCS}}$/$\overline{\text{AUDSYNC}}$ 引脚的功能。 000：PD7 输入 / 输出 （端口） 001：D7 输入 / 输出 （BSC） *2 011：$\overline{\text{AUDSYNC}}$ 输出 （AUD） 101：$\overline{\text{SCS}}$ 输入 / 输出 （SSU） 110：TXD2 输出 （SCI） 上述以外：禁止设定
11	—	0	R	保留位 读取值、写入值总是为 0。
10 9 8	PD6MD2 PD6MD1 PD6MD0	0 0 0*1	R/W R/W R/W	PD6 模式位 选择 PD6/D6/RXD2/AUDCK 引脚的功能。 000：PD6 输入 / 输出 （端口） 001：D6 输入 / 输出 （BSC） *2 011：AUDCK 输出 （AUD） 110：RXD2 输入 （SCI） 上述以外：禁止设定
7	—	0	R	保留位 读取值、写入值总是为 0。
6 5 4	PD5MD2 PD5MD1 PD5MD0	0 0 0*1	R/W R/W R/W	PD5 模式位 选择 PD5/D5/SCK1/AUDMD 引脚的功能。 000：PD5 输入 / 输出 （端口） 001：D5 输入 / 输出 （BSC） *2 011：AUDMD 输入 （AUD） 110：SCK1 输入 / 输出 （SCI） 上述以外：禁止设定
3	—	0	R	保留位 读取值、写入值总是为 0。
2 1 0	PD4MD2 PD4MD1 PD4MD0	0 0 0*1	R/W R/W R/W	PD4 模式位 选择 PD4/D4/TXD1/$\overline{\text{AUDRST}}$ 引脚的功能。 000：PD4 输入 / 输出 （端口） 001：D4 输入 / 输出 （BSC） *2 011：$\overline{\text{AUDRST}}$ 输入 （AUD） 110：TXD1 输出 （SCI） 上述以外：禁止设定

【注】 *1　在内部 ROM 无效外部扩展模式，初始值为 1。

*2　仅在内部 ROM 有效 / 无效外部扩展模式有效，不得在单芯片模式设定。

(3)　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
	-	PD3 MD2	PD3 MD1	PD3 MD0	-	PD2 MD2	PD2 MD1	PD2 MD0	-	PD1 MD2	PD1 MD1	PD1 MD0	-	PD0 MD2	PD0 MD1	PD0 MD0
初始值：	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W：	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

【注】*1 在内部ROM无效外部扩展模式，初始值为1。

位	位名称	初始值	R/W	说　明
15	—	0	R	保留位 读取值、写入值总是为 0。
14 13 12	PD3MD2 PD3MD1 PD3MD0	0 0 0*1	R/W R/W R/W	PD3 模式位 选择 PD3/D3/RXD1/AUDATA3 引脚的功能。 000：PD3 输入 / 输出 （端口） 001：D3 输入 / 输出 （BSC） *2 011：AUDATA3 输出 （AUD） 110：RXD1 输入 （SCI） 上述以外：禁止设定
11	—	0	R	保留位 读取值、写入值总是为 0。
10 9 8	PD2MD2 PD2MD1 PD2MD0	0 0 0*1	R/W R/W R/W	PD2 模式位 选择 PD2/D2/SCK0/AUDATA2 引脚的功能。 000：PD2 输入 / 输出 （端口） 001：D2 输入 / 输出 （BSC） *2 011：AUDATA2 输出 （AUD） 110：SCK0 输入 / 输出 （SCI） 上述以外：禁止设定
7	—	0	R	保留位 读取值、写入值总是为 0。
6 5 4	PD1MD2 PD1MD1 PD1MD0	0 0 0*1	R/W R/W R/W	PD1 模式位 选择 PD1/D1/TXD0/AUDATA1 引脚的功能。 000：PD1 输入 / 输出 （端口） 001：D1 输入 / 输出 （BSC） *2 011：AUDATA1 输出 （AUD） 110：TXD0 输出 （SCI） 上述以外：禁止设定
3	—	0	R	保留位 读取值、写入值总是为 0。
2 1 0	PD0MD2 PD0MD1 PD0MD0	0 0 0*1	R/W R/W R/W	PD0 模式位 选择 PD0/D0/RXD0/AUDATA0 引脚的功能。 000：PD0 输入 / 输出 （端口） 001：D0 输入 / 输出 （BSC） *2 011：AUDATA0 输出 （AUD） 110：RXD0 输入 （SCI） 上述以外：禁止设定

【注】 *1　在内部 ROM 无效外部扩展模式，初始值为 1。

*2　仅在内部 ROM 有效 / 无效外部扩展模式有效，不得在单芯片模式设定。

18.1.7　端口 E/IO 寄存器 L、 H （PEIORL、 PEIORH）

PEIORL、 PEIORH 分别为可读取 / 写入的 16 位寄存器，选择端口 E 的引脚输入 / 输出方向。 PE21IOR ～ PE0IOR 位分别对应 PE21 引脚～ PE0 引脚 （省略引脚名称中除端口之外的多路复用引脚名称）。 PEIORL 在端口 E 的引脚功能为通用输入 / 输出 （PE15 ～ PE0）及 MTU2 的 TIOC 输入 / 输出时有效，除此之外无效。 PEIORH 在端口 E 的引脚功能为通用输入 / 输出 （PE21 ～ PE16）及 MTU2S 的 TIOC 输入 / 输出时有效，除此之外无效。

如果 PEIORL 及 PEIORH 的位置 1，则对应的引脚为输出引脚，置 0 则为输入引脚。

PEIORH 的 bit15 ～ 6 为保留位，读取值、写入值总是为 0。

PEIORL、 PEIORH 的初始值均为 H'0000。

(1)　端口 E/IO 寄存器 H （PEIORH）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PE21 IOR	PE20 IOR	PE19 IOR	PE18 IOR	PE17 IOR	PE16 IOR
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

(2)　端口 E/IO 寄存器 L （PEIORL）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15 IOR	PE14 IOR	PE13 IOR	PE12 IOR	PE11 IOR	PE10 IOR	PE9 IOR	PE8 IOR	PE7 IOR	PE6 IOR	PE5 IOR	PE4 IOR	PE3 IOR	PE2 IOR	PE1 IOR	PE0 IOR
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

18.1.8　端口 E 控制寄存器 L1 ～ L4、 H1、 H2 （PECRL1 ～ PECRL4、 PECRH1、 PECRH2）

PECRL1 ～ PECRL4、 PECRH1、 PECRH2 分别为可读取 / 写入的 16 位寄存器，选择端口 E 的多路复用引脚功能。

(1)　端口 E 控制寄存器 H2 （PECRH2）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PE21 MD1	PE21 MD0	-	-	PE20 MD1	PE20 MD0
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

位	位名称	初始值	R/W	说　明
15 ～ 6	—	均为 0	R	保留位 读取值、写入值总是为 0。
5 4	PE21MD1 PE21MD0	0 0	R/W R/W	PE21 模式位 选择 PE21/$\overline{\text{WRL}}$/TIOC4DS 引脚的功能。 00：PE21 输入 / 输出 （端口） 01：TIOC4DS 输入 / 输出 （MTU2S） 10：$\overline{\text{WRL}}$ 输出 （BSC） * 上述以外：禁止设定
3、 2	—	均为 0	R	保留位 读取值、写入值总是为 0。
1 0	PE20MD1 PE20MD0	0 0	R/W R/W	PE20 模式位 选择 PE20/TIOC4CS 引脚的功能。 00：PE20 输入 / 输出 （端口） 01：TIOC4CS 输入 / 输出 （MTU2S） 上述以外：禁止设定

【注】 *　仅在内部 ROM 有效 / 无效外部扩展模式有效，不得在单芯片模式设定。

(2)　端口 E 控制寄存器 H1 （PECRH1）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	PE19 MD1	PE19 MD0	-	-	PE18 MD1	PE18 MD0	-	-	PE17 MD1	PE17 MD0	-	PE16 MD2	PE16 MD1	PE16 MD0
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

位	位名称	初始值	R/W	说　明
15、14	—	均为 0	R	保留位 读取值、写入值总是为 0。
13 12	PE19MD1 PE19MD0	0 0	R/W R/W	PE19 模式位 选择 PE19/$\overline{\text{RD}}$/TIOC4BS 引脚的功能。 00：PE19 输入 / 输出 （端口） 01：TIOC4BS 输入 / 输出 （MTU2S） 10：$\overline{\text{RD}}$ 输出 （BSC） * 上述以外：禁止设定
11、10	—	均为 0	R	保留位 读取值、写入值总是为 0。
9 8	PE18MD1 PE18MD0	0 0	R/W R/W	PE18 模式位 选择 PE18/$\overline{\text{CS1}}$/TIOC4AS 引脚的功能。 00：PE18 输入 / 输出 （端口） 01：TIOC4AS 输入 / 输出 （MTU2S） 10：$\overline{\text{CS1}}$ 输出 （BSC） * 上述以外：禁止设定
7、6	—	均为 0	R	保留位 读取值、写入值总是为 0。
5 4	PE17MD1 PE17MD0	0 0	R/W R/W	PE17 模式位 选择 PE17/$\overline{\text{CS0}}$/TIOC3DS 引脚的功能。 00：PE17 输入 / 输出 （端口） 01：TIOC3DS 输入 / 输出 （MTU2S） 10：$\overline{\text{CS0}}$ 输出 （BSC） * 上述以外：禁止设定
3	—	0	R	保留位 读取值、写入值总是为 0。
2 1 0	PE16MD2 PE16MD1 PE16MD0	0 0 0	R/W R/W R/W	PE16 模式位 选择 PE16/$\overline{\text{WAIT}}$/TIOC3BS 引脚的功能。 000：PE16 输入 / 输出 （端口） 001：TIOC3BS 输入 / 输出 （MTU2S） 010：$\overline{\text{WAIT}}$ 输入 （BSC） * 上述以外：禁止设定

【注】 *　仅在内部 ROM 有效 / 无效外部扩展模式有效，不得在单芯片模式设定。

(3)　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
	-	PE15 MD2	PE15 MD1	PE15 MD0	-	PE14 MD2	PE14 MD1	PE14 MD0	-	-	PE13 MD1	PE13 MD0	-	PE12 MD2	PE12 MD1	PE12 MD0
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

位	位名称	初始值	R/W	说　明
15	—	0	R	保留位 读取值、写入值总是为 0。
14 13 12	PE15MD2 PE15MD1 PE15MD0	0 0 0	R/W R/W R/W	PE15 模式位 选择 PE15/TIOC4D/$\overline{\text{IRQOUT}}$ 引脚的功能。 000：PE15 输入 / 输出 （端口） 001：TIOC4D 输入 / 输出 （MTU2） 011：$\overline{\text{IRQOUT}}$ 输出 （INTC） 上述以外：禁止设定
11	—	0	R	保留位 读取值、写入值总是为 0。
10 9 8	PE14MD2 PE14MD1 PE14MD0	0 0 0	R/W R/W R/W	PE14 模式位 选择 PE14/TIOC4C 引脚的功能。 000：PE14 输入 / 输出 （端口） 001：TIOC4C 输入 / 输出 （MTU2） 上述以外：禁止设定
7、6	—	均为 0	R	保留位 读取值、写入值总是为 0。
5 4	PE13MD1 PE13MD0	0 0	R/W R/W	PE13 模式位 选择 PE13/TIOC4B/$\overline{\text{MRES}}$ 引脚的功能。 00：PE13 输入 / 输出 （端口） 01：TIOC4B 输入 / 输出 （MTU2） 10：$\overline{\text{MRES}}$ 输入 （INTC） 上述以外：禁止设定
3	—	0	R	保留位 读取值、写入值总是为 0。
2 1 0	PE12MD2 PE12MD1 PE12MD0	0 0 0	R/W R/W R/W	PE12 模式位 选择 PE12/TIOC4A 引脚的功能。 000：PE12 输入 / 输出 （端口） 001：TIOC4A 输入 / 输出 （MTU2） 上述以外：禁止设定

(4)　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
	-	PE11 MD2	PE11 MD1	PE11 MD0	-	PE10 MD2	PE10 MD1	PE10 MD0	-	PE9 MD2	PE9 MD1	PE9 MD0	-	PE8 MD2	PE8 MD1	PE8 MD0
初始值：	0	0	0	0	0	0*1	0	0	0	0	0	0	0	0*1	0	0
R/W：	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

【注】*1 在内部ROM无效外部扩展模式，初始值为1。

位	位名称	初始值	R/W	说　明
15	—	0	R	保留位 读取值、写入值总是为 0。
14 13 12	PE11MD2 PE11MD1 PE11MD0	0 0 0	R/W R/W R/W	PE11 模式位 选择 PE11/TIOC3D 引脚的功能。 000：PE11 输入 / 输出 （端口） 001：TIOC3D 输入 / 输出 （MTU2） 上述以外：禁止设定
11	—	0	R	保留位 读取值、写入值总是为 0。
10 9 8	PE10MD2 PE10MD1 PE10MD0	0*1 0 0	R/W R/W R/W	PE10 模式位 选择 PE10/$\overline{\text{CS0}}$/TIOC3C 引脚的功能。 000：PE10 输入 / 输出 （端口） 001：TIOC3C 输入 / 输出 （MTU2） 100：$\overline{\text{CS0}}$ 输出 （BSC） *2 上述以外：禁止设定
7	—	0	R	保留位 读取值、写入值总是为 0。
6 5 4	PE9MD2 PE9MD1 PE9MD0	0 0 0	R/W R/W R/W	PE9 模式位 选择 PE9/TIOC3B 引脚的功能。 000：PE9 输入 / 输出 （端口） 001：TIOC3B 输入 / 输出 （MTU2） 上述以外：禁止设定
3	—	0	R	保留位 读取值、写入值总是为 0。
2 1 0	PE8MD2 PE8MD1 PE8MD0	0*1 0 0	R/W R/W R/W	PE8 模式位 选择 PE8/A15/TIOC3A 引脚的功能。 000：PE8 输入 / 输出 （端口） 001：TIOC3A 输入 / 输出 （MTU2） 100：A15 输出 （BSC） *2 上述以外：禁止设定

【注】 *1　在内部 ROM 无效外部扩展模式，初始值为 1。

*2　仅在内部 ROM 有效 / 无效外部扩展模式有效，不得在单芯片模式设定。

(5)　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
	-	PE7 MD2	PE7 MD1	PE7 MD0	-	PE6 MD2	PE6 MD1	PE6 MD0	-	PE5 MD2	PE5 MD1	PE5 MD0	-	PE4 MD2	PE4 MD1	PE4 MD0
初始值：	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0
R/W：	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

【注】*1 在内部ROM无效外部扩展模式，初始值为1。

位	位名称	初始值	R/W	说　明
15	—	0	R	保留位 读取值、写入值总是为 0。
14 13 12	PE7MD2 PE7MD1 PE7MD0	0*1 0 0	R/W R/W R/W	PE7 模式位 选择 PE7/A14/TIOC2B 引脚的功能。 000：PE7 输入 / 输出 （端口） 001：TIOC2B 输入 / 输出 （MTU2） 100：A14 输出 （BSC） *2 上述以外：禁止设定
11	—	0	R	保留位 读取值、写入值总是为 0。
10 9 8	PE6MD2 PE6MD1 PE6MD0	0*1 0 0	R/W R/W R/W	PE6 模式位 选择 PE6/A13/TIOC2A/SCK1 引脚的功能。 000：PE6 输入 / 输出 （端口） 001：TIOC2A 输入 / 输出 （MTU2） 100：A13 输出 （BSC） *2 110：SCK1 输入 / 输出 （SCI） 上述以外：禁止设定
7	—	0	R	保留位 读取值、写入值总是为 0。
6 5 4	PE5MD2 PE5MD1 PE5MD0	0*1 0 0	R/W R/W R/W	PE5 模式位 选择 PE5/A12/TIOC1B/TXD1 引脚的功能。 000：PE5 输入 / 输出 （端口） 001：TIOC1B 输入 / 输出 （MTU2） 100：A12 输出 （BSC） *2 110：TXD1 输出 （SCI） 上述以外：禁止设定
3	—	0	R	保留位 读取值、写入值总是为 0。
2 1 0	PE4MD2 PE4MD1 PE4MD0	0*1 0 0	R/W R/W R/W	PE4 模式位 选择 PE4/A11/TIOC1A/RXD1 引脚的功能。 000：PE4 输入 / 输出 （端口） 001：TIOC1A 输入 / 输出 （MTU2） 100：A11 输出 （BSC） *2 110：RXD1 输入 （SCI） 上述以外：禁止设定

【注】 *1　在内部 ROM 无效外部扩展模式，初始值为 1。

*2　仅在内部 ROM 有效 / 无效外部扩展模式有效，不得在单芯片模式设定。

(6)　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
	-	PE3 MD2	PE3 MD1	PE3 MD0	-	PE2 MD2	PE2 MD1	PE2 MD0	-	PE1 MD2	PE1 MD1	PE1 MD0	-	-	PE0 MD1	PE0 MD0
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W

位	位名称	初始值	R/W	说　明
15	—	0	R	保留位 读取值、写入值总是为 0。
14 13 12	PE3MD2 PE3MD1 PE3MD0	0 0 0	R/W R/W R/W	PE3 模式位 选择 PE3/TIOC0D/SCK0 引脚的功能。 000：PE3 输入 / 输出 （端口） 001：TIOC0D 输入 / 输出 （MTU2） 110：SCK0 输入 / 输出 （SCI） 上述以外：禁止设定
11	—	0	R	保留位 读取值、写入值总是为 0。
10 9 8	PE2MD2 PE2MD1 PE2MD0	0 0 0	R/W R/W R/W	PE2 模式位 选择 PE2/TIOC0C/TXD0 引脚的功能。 000：PE2 输入 / 输出 （端口） 001：TIOC0C 输入 / 输出 （MTU2） 110：TXD0 输出 （SCI） 上述以外：禁止设定
7	—	0	R	保留位 读取值、写入值总是为 0。
6 5 4	PE1MD2 PE1MD1 PE1MD0	0 0 0	R/W R/W R/W	PE1 模式位 选择 PE1/TIOC0B/RXD0 引脚的功能。 000：PE1 输入 / 输出 （端口） 001：TIOC0B 输入 / 输出 （MTU2） 110：RXD0 输入 （SCI） 上述以外：禁止设定
3、2	—	均为 0	R	保留位 读取值、写入值总是为 0。
1 0	PE0MD1 PE0MD0	0 0	R/W R/W	PE0 模式位 选择 PE0/TIOC0A 引脚的功能。 00：PE0 输入 / 输出 （端口） 01：TIOC0A 输入 / 输出 （MTU2） 上述以外：禁止设定

18.2　使用时的注意事项

1. 在本LSI，相同功能作为多路复用功能分配至多个引脚，这是为了在提高引脚功能选择自由度的同时，更易于设定电路板，但在2个以上的引脚使用1个功能时，必须注意以下几点：

- 引脚功能为输入功能时
  从多个引脚输入的信号，通过OR或AND逻辑形成1个信号，传送至LSI内部，因此，根据其他相同功能引脚的输入状态，有时会将与输入信号不同的信号传送至LSI内部。分配至多个引脚的输入功能的传送形式如表18.7所示。在2个以上的引脚使用以下任意一个功能时，必须考虑传送形式，并注意信号的极性。

表 18.7　分配至多个引脚的输入功能的传送形式

OR 型	AND 型
SCK0 ～ SCK2、 RXD0 ～ RXD2、 SSO、 SSI、 SSCK	IRQ0 ～ IRQ3、 $\overline{\text{WAIT}}$、 $\overline{\text{SCS}}$、 $\overline{\text{POE0}}$、 $\overline{\text{POE1}}$、 $\overline{\text{POE4}}$、 $\overline{\text{POE5}}$

  OR型：从多个引脚输入的信号，通过OR逻辑形成1个信号，传送至LSI内部。
  AND型：从多个引脚输入的信号，通过AND逻辑形成1个信号，传送至LSI内部。

- 引脚功能为输出功能时
  可从选择的所有引脚输出相同功能。

2. 输入/输出端口与IRQ为多路复用的引脚，端口输入从低电平状态转换为IRQ边沿检测时，检测此边沿。
3. 仅可设定表18.5中PFC可设定的功能，否则无法保证运行。
4. 在单芯片模式 （MCU运行模式3） PFC的设定
   在单芯片模式， PFC不可选择地址总线、数据总线、总线控制信号、 $\overline{\text{BREQ}}$、 $\overline{\text{BACK}}$及CK，如果选择，地址总线变为高电平或低电平输出，数据总线变为高阻抗输出，其他输出信号变为高电平输出。
   $\overline{\text{BREQ}}$与$\overline{\text{WAIT}}$为输入引脚，因此不得设置为开路，但总线权请求输入与外部等待无效。

第 19 章　I/O 端口

有 A、 B、 D、 E 四个端口。 A、 B、 D、 E 分别为 16 位、 8 位、 11 位、 22 位输入 / 输出端口。

各端口的引脚均为兼具其他功能的多路复用引脚。可通过引脚功能控制器 （PFC）选择多路复用引脚的功能。

各端口均具有保存引脚数据的数据寄存器。

19.1　端口 A

如图 19.1 所示，端口 A 为具有 16 个引脚的输入 / 输出端口。

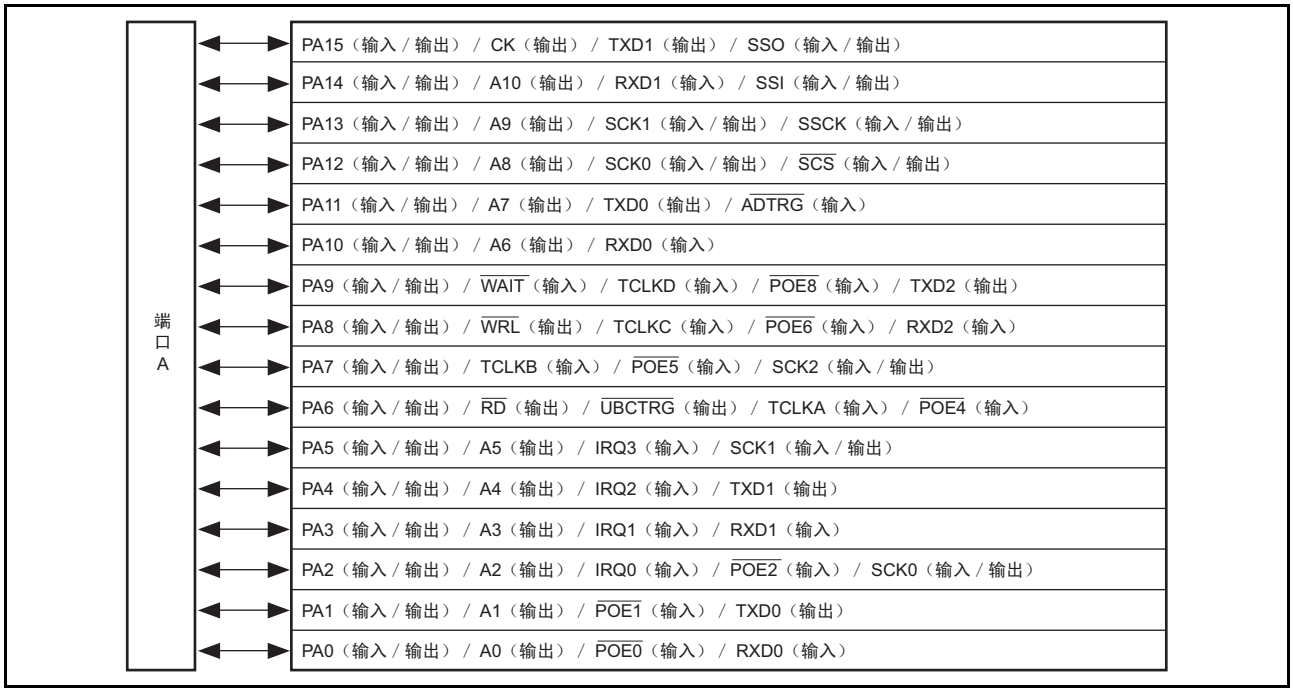


图 19.1　端口 A

19.1.1　寄存器说明

端口 A 为 16 位输入 / 输出端口，并具有以下寄存器。有关寄存器的地址及各处理状态的寄存器状态，详情参阅 “第 24 章 寄存器一览表”。

表 19.1　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/W	H'0000	H'FFFFD102	8、16
端口 A 端口寄存器 L	PAPRL	R	H'xxxx	H'FFFFD11E	8、16

19.1.2　端口 A 数据寄存器 L （PADRL）

PADRL 为可读取 / 写入的 16 位寄存器，用于保存端口 A 的数据。 PA15DR ～ PA0DR 位分别对应 PA15 ～ PA0 引脚 （省略复用功能的说明）。

引脚功能为通用输出时，如果对 PADRL 写入值，则从引脚输出该值；如果读取 PADRL，则与引脚状态无关，直接读取寄存器的值。

引脚功能为通用输入时，如果读取 PADRL，则直接读取引脚状态而非寄存器的值。可对 PADRL 写入该值，但不影响引脚状态。端口 A 数据寄存器的读取 / 写入如表 19.2 所示。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15 DR	PA14 DR	PA13 DR	PA12 DR	PA11 DR	PA10 DR	PA9 DR	PA8 DR	PA7 DR	PA6 DR	PA5 DR	PA4 DR	PA3 DR	PA2 DR	PA1 DR	PA0 DR
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说　明
15	PA15DR	0	R/W	参照表 19.2
14	PA14DR	0	R/W	
13	PA13DR	0	R/W	
12	PA12DR	0	R/W	
11	PA11DR	0	R/W	
10	PA10DR	0	R/W	
9	PA9DR	0	R/W	
8	PA8DR	0	R/W	
7	PA7DR	0	R/W	
6	PA6DR	0	R/W	
5	PA5DR	0	R/W	
4	PA4DR	0	R/W	
3	PA3DR	0	R/W	
2	PA2DR	0	R/W	
1	PA1DR	0	R/W	
0	PA0DR	0	R/W	

表 19.2　端口 A 数据寄存器 L （PADRL）的读取 / 写入

- PADRL 的 bit15～0

PAIOR	引脚功能	读取	写入
0	通用输入	引脚状态	可写入 PADRL，但不影响引脚状态
	除通用输入之外	引脚状态	可写入 PADRL，但不影响引脚状态
1	通用输出	PADRL 的值	从引脚输出写入值
	除通用输出之外	PADRL 的值	可写入 PADRL，但不影响引脚状态

19.1.3　端口 A 端口寄存器 L （PAPRL）

PAPRL 为 16 位只读寄存器，与 PFC 的设定无关，总是可读取引脚状态。 PA15PR ～ PA0PR 位分别对应 PA15 ～ PA0 引脚 （省略复用功能的说明）。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15 PR	PA14 PR	PA13 PR	PA12 PR	PA11 PR	PA10 PR	PA9 PR	PA8 PR	PA7 PR	PA6 PR	PA5 PR	PA4 PR	PA3 PR	PA2 PR	PA1 PR	PA0 PR
初始值：	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W：	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名称	初始值	R/W	说　明
15	PA15PR	引脚状态	R	与 PFC 的设定无关，可读取引脚状态，写入无效。
14	PA14PR	引脚状态	R	
13	PA13PR	引脚状态	R	
12	PA12PR	引脚状态	R	
11	PA11PR	引脚状态	R	
10	PA10PR	引脚状态	R	
9	PA9PR	引脚状态	R	
8	PA8PR	引脚状态	R	
7	PA7PR	引脚状态	R	
6	PA6PR	引脚状态	R	
5	PA5PR	引脚状态	R	
4	PA4PR	引脚状态	R	
3	PA3PR	引脚状态	R	
2	PA2PR	引脚状态	R	
1	PA1PR	引脚状态	R	
0	PA0PR	引脚状态	R	

19.2　端口 B

如图 19.2 所示，端口 B 为具有 8 个引脚的输入 / 输出端口。

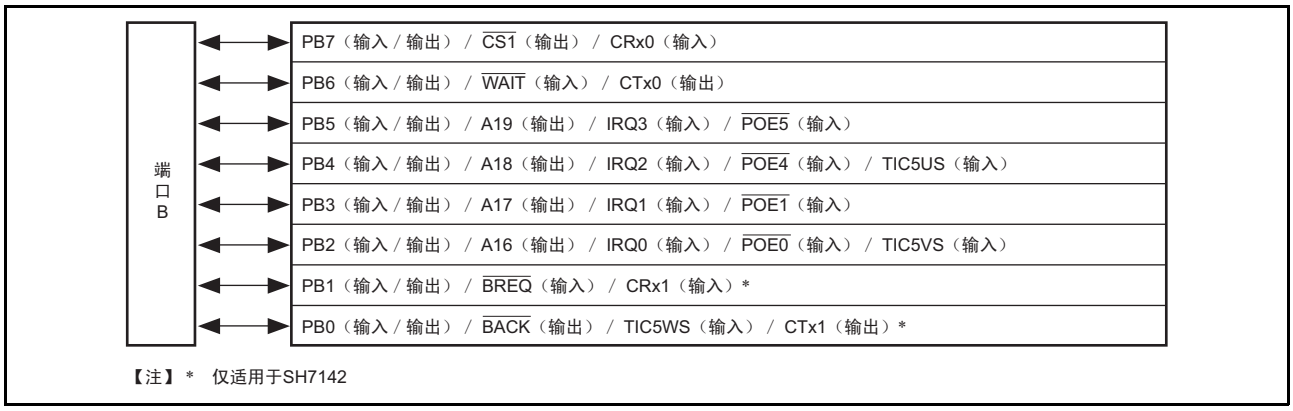


图 19.2　端口 B

19.2.1　寄存器说明

端口 B 为 8 位输入 / 输出端口，并具有以下寄存器。有关寄存器的地址及各处理状态的寄存器状态，详情参阅 “第 24 章 寄存器一览表”。

表 19.3　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'0000	H'FFFFD182	8、16
端口 B 端口寄存器 L	PBPRL	R	H'00xx	H'FFFFD19E	8、16

19.2.2　端口 B 数据寄存器 L （PBDRL）

PBDRL 为可读取 / 写入的 16 位寄存器，用于保存端口 B 的数据。 PB7DR ～ PB0DR 位分别对应 PB7 ～ PB0 引脚 （省略复用功能的说明）。

引脚功能为通用输出时，如果对 PBDRL 写入值，则从引脚输出该值；如果读取 PBDRL，则与引脚状态无关，直接读取寄存器的值。

引脚功能为通用输入时，如果读取 PBDRL，则直接读取引脚状态而非寄存器的值。可对 PBDRL 写入该值，但不影响引脚状态。端口 B 数据寄存器的读取 / 写入如表 19.4 所示。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
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 读取值、写入值总是为 0。
7	PB7DR	0	R/W	参照表 19.4
6	PB6DR	0	R/W	
5	PB5DR	0	R/W	
4	PB4DR	0	R/W	
3	PB3DR	0	R/W	
2	PB2DR	0	R/W	
1	PB1DR	0	R/W	
0	PB0DR	0	R/W	

表 19.4　端口 B 数据寄存器 L （PBDRL）的读取 / 写入

- PBDRL 的 bit7 ～ 0

PBIOR	引脚功能	读取	写入
0	通用输入	引脚状态	可写入 PBDRL，但不影响引脚状态
	除通用输入之外	引脚状态	可写入 PBDRL，但不影响引脚状态
1	通用输出	PBDRL 的值	从引脚输出写入值
	除通用输出之外	PBDRL 的值	可写入 PBDRL，但不影响引脚状态

19.2.3　端口 B 端口寄存器 L （PBPRL）

PBPRL 为 16 位只读寄存器，与 PFC 的设定无关，总是可读取引脚状态。 PB7PR ～ PB0PR 位分别对应 PB7 ～ PB0 引脚 （省略复用功能的说明）。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	PB0PR
初始值：	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	*
R/W：	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名称	初始值	R/W	说　明
15 ～ 8	—	均为 0	R	保留位 读取值、写入值总是为 0。
7	PB7PR	引脚状态	R	与 PFC 的设定无关，可读取引脚状态，写入无效。
6	PB6PR	引脚状态	R	
5	PB5PR	引脚状态	R	
4	PB4PR	引脚状态	R	
3	PB3PR	引脚状态	R	
2	PB2PR	引脚状态	R	
1	PB1PR	引脚状态	R	
0	PB0PR	引脚状态	R	

19.3　端口 D

如图 19.3 所示，端口 D 为具有 11 个引脚的输入 / 输出端口。

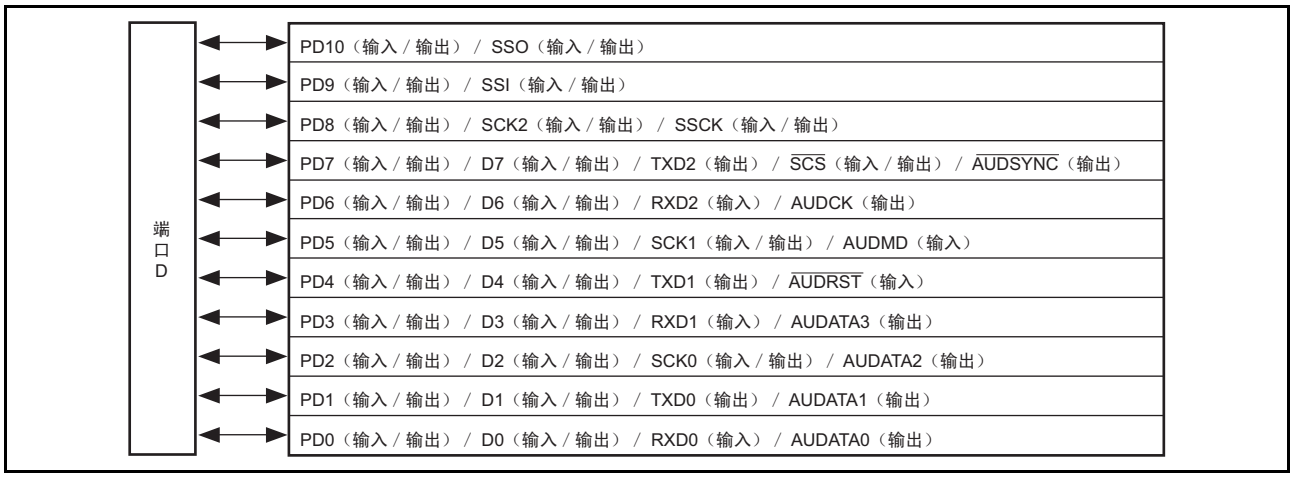


图 19.3　端口 D

19.3.1　寄存器说明

端口 D 为 11 位输入 / 输出端口，并具有以下寄存器。有关寄存器的地址及各处理状态的寄存器状态，详情参阅 “第 24 章　寄存器一览表”。

表 19.5　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 D 数据寄存器 L	PDDRL	R/W	H'0000	H'FFFFD282	8、16
端口 D 端口寄存器 L	PDPRL	R	Hxxxx	H'FFFFD29E	8、16

19.3.2　端口 D 数据寄存器 L （PDDRL）

PDDRL 为可读取 / 写入的 16 位寄存器，用于保存端口 D 的数据。 PD10DR ～ PD0DR 位分别对应 PD10 ～ PD0 引脚 （省略复用功能的说明）。

引脚功能为通用输出时，如果对 PDDRL 写入值，则从引脚输出该值；如果读取 PDDRL，则与引脚状态无关，直接读取寄存器的值。

引脚功能为通用输入时，如果读取 PDDRL，则直接读取引脚状态而非寄存器的值。可对 PDDRL 写入该值，但不影响引脚状态。端口 D 数据寄存器 L 的读取 / 写入如表 19.6 所示。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	PD10 DR	PD9 DR	PD8 DR	PD7 DR	PD6 DR	PD5 DR	PD4 DR	PD3 DR	PD2 DR	PD1 DR	PD0 DR
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R	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 读取值、写入值总是为 0。
10	PD10DR	0	R/W	参照表 19.6
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 19.6　端口 D 数据寄存器 L （PDDRL）的读取 / 写入

- PDDRL 的 bit15 ～ 0

PDIOR	引脚功能	读取	写入
0	通用输入	引脚状态	可写入 PDDRL，但不影响引脚状态
	通用输入以外	引脚状态	可写入 PDDRL，但不影响引脚状态
1	通用输出	PDDRL 的值	从引脚输出写入值
	通用输出以外	PDDRL 的值	可写入 PDDRL，但不影响引脚状态

19.3.3　端口 D 端口寄存器 L （PDPRL）

PDPRL 为 16 位只读寄存器，与 PFC 的设定无关，总是可读取引脚状态。 PD10PR ～ PD0PR 位分别对应 PD10 ～ PD0 引脚 （省略复用功能的说明）。

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	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：	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 ～ 11	—	均为 0	R	保留位 读取值、写入值总是为 0。
10	PD10PR	引脚状态	R	与 PFC 的设定无关，可读取引脚状态，写入无效。
9	PD9PR	引脚状态	R	
8	PD8PR	引脚状态	R	
7	PD7PR	引脚状态	R	
6	PD6PR	引脚状态	R	
5	PD5PR	引脚状态	R	
4	PD4PR	引脚状态	R	
3	PD3PR	引脚状态	R	
2	PD2PR	引脚状态	R	
1	PD1PR	引脚状态	R	
0	PD0PR	引脚状态	R	

19.4　端口 E

如图 19.4 所示，端口 E 为具有 22 个引脚的输入 / 输出端口。

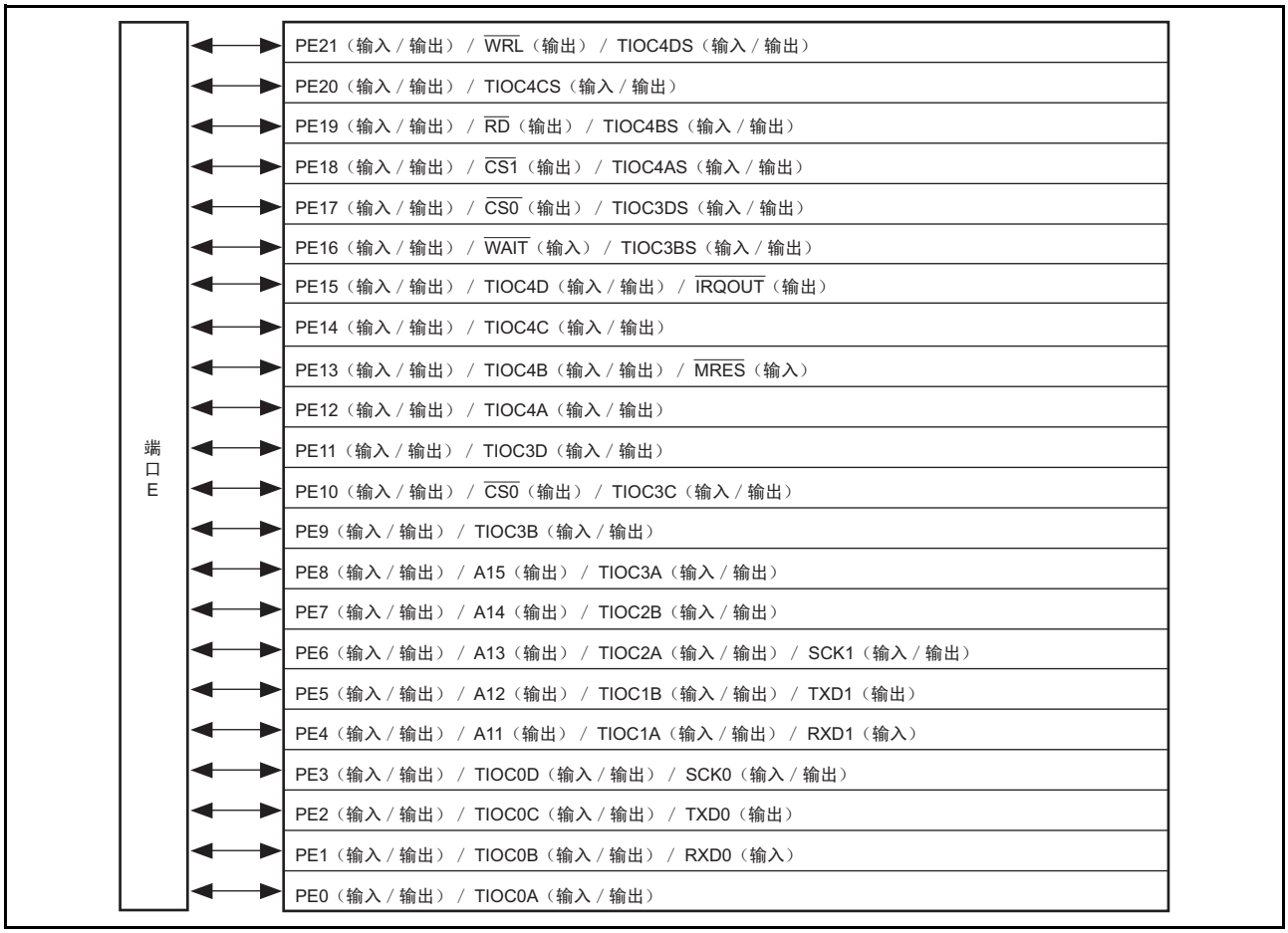


图 19.4　端口 E

19.4.1　寄存器说明

端口 E 为 22 位输入 / 输出端口，并具有以下寄存器。有关寄存器的地址及各处理状态的寄存器，详情参阅 “第 24 章 寄存器一览表”。

表 19.7　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 E 数据寄存器 H	PEDRH	R/W	H'0000	H'FFFFD300	8、16、32
端口 E 数据寄存器 L	PEDRL	R/W	H'0000	H'FFFFD302	8、16
端口 E 端口寄存器 H	PEPRH	R	H'00xx	H'FFFFD31C	8、16、32
端口 E 端口寄存器 L	PEPRL	R	H'xxxx	H'FFFFD31E	8、16

19.4.2　端口 E 数据寄存器 H、L（PEDRH、PEDRL）

PEDRH 及 PEDRL 为可读取 / 写入的 16 位寄存器，用于保存端口 E 的数据。 PE21DR ～ PE0DR 位分别对应 PE21 ～ PE0 引脚（省略复用功能的说明）。

引脚功能为通用输出时，如果对 PEDRH 或 PEDRL 写入值，则从引脚输出该值；如果读取 PEDRH 或 PEDRL，则与引脚状态无关，直接读取寄存器的值。

引脚功能为通用输入时，如果读取 PEDRH 或 PEDRL，则直接读取引脚状态而非寄存器的值。可对 PEDRH 或 PEDRL 写入该值，但不影响引脚状态。端口 E 数据寄存器的读取 / 写入如表 19.8 所示。

(1)　端口 E 数据寄存器 H（PEDRH）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PE21DR	PE20DR	PE19DR	PE18DR	PE17DR	PE16DR
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 读取值、写入值总是为 0。
5	PE21DR	0	R/W	参照表 19.8
4	PE20DR	0	R/W	
3	PE19DR	0	R/W	
2	PE18DR	0	R/W	
1	PE17DR	0	R/W	
0	PE16DR	0	R/W	

(2) 端口 E 数据寄存器 L （PEDRL）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15 DR	PE14 DR	PE13 DR	PE12 DR	PE11 DR	PE10 DR	PE9 DR	PE8 DR	PE7 DR	PE6 DR	PE5 DR	PE4 DR	PE3 DR	PE2 DR	PE1 DR	PE0 DR
初始值：	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说 明
15	PE15DR	0	R/W	参照表 19.8
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 19.8 端口 E 数据寄存器 （PEDR）的读取 / 写入

- PEDRH 的 bit5～0 及 PEDRL 的 bit15～0

PEIOR	引脚功能	读取	写入
0	通用输入	引脚状态	可写入 PEDRH、L，但不影响引脚状态
	除通用输入之外	引脚状态	可写入 PEDRH、L，但不影响引脚状态
1	通用输出	PEDRH、L 的值	从引脚输出写入值
	除通用输出之外	PEDRH、L 的值	可写入 PEDRH、L，但不影响引脚状态

19.4.3 端口 E 端口寄存器 H、L（PEPRH、PEPRL）

PEPRH、PEPRL 分别为 16 位只读寄存器，与 PFC 的设定无关，总是可读取引脚状态。PE21PR ～ PE0PR 位分别对应 PE21 ～ PE0 引脚（省略复用功能的说明）。

(1) 端口 E 端口寄存器 H（PEPRH）

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PE21PR	PE20PR	PE19PR	PE18PR	PE17PR	PE16PR
初始值：	0	0	0	0	0	0	0	0	0	0	*	*	*	*	*	*
R/W：	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名称	初始值	R/W	说　明
15 ～ 6	—	均为 0	R	保留位 读取值、写入值总是为 0。
5	PE21PR	引脚状态	R	与 PFC 的设定无关，可读取引脚状态，写入无效。
4	PE20PR	引脚状态	R	
3	PE19PR	引脚状态	R	
2	PE18PR	引脚状态	R	
1	PE17PR	引脚状态	R	
0	PE16PR	引脚状态	R	

(2) 端口 E 端口寄存器 L（PEPRL）

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

位	位名称	初始值	R/W	说　明
15	PE15PR	引脚状态	R	与 PFC 的设定无关，可读取引脚状态，写入无效。
14	PE14PR	引脚状态	R	
13	PE13PR	引脚状态	R	
12	PE12PR	引脚状态	R	
11	PE11PR	引脚状态	R	
10	PE10PR	引脚状态	R	
9	PE9PR	引脚状态	R	
8	PE8PR	引脚状态	R	
7	PE7PR	引脚状态	R	
6	PE6PR	引脚状态	R	
5	PE5PR	引脚状态	R	
4	PE4PR	引脚状态	R	
3	PE3PR	引脚状态	R	
2	PE2PR	引脚状态	R	
1	PE1PR	引脚状态	R	
0	PE0PR	引脚状态	R	

第 20 章 闪存

本 LSI 内置 512KB/384KB/256KB* 的闪存，特点如下：

20.1 特点

- 配合LSI启动模式的两种闪存MAT
  内部闪存有分配在相同地址空间的两种存储器空间（以下称为存储器MAT），根据启动时设定的模式，可选择从哪个存储器MAT开始启动。启动后还可通过存储体转换方式转换MAT。
  在用户模式，上电复位时启动的用户MAT：512KB/384KB/256KB*
  在用户引导模式，上电复位时启动的用户引导MAT：12KB
- 3种板上编程模式与1种板外编程模式
  板上编程模式：
  引导模式
  为使用内部SCI接口的编程模式，可改写用户MAT和用户引导MAT。在该模式，可自动调整主机与本LSI之间的位速率。
  用户编程模式
  可通过任意接口改写用户MAT。
  用户引导模式
  可编写任意接口的用户引导程序，可改写用户MAT。

  板外编程模式：
  编程器模式
  为使用PROM编程器的编程器模式，可改写用户MAT与用户引导MAT。

【注】 * 参阅“1.1 特点”的产品一览表。

- 由下载内部程序产生的编程/擦除接口
  本LSI内置专用的编程/擦除程序。将此程序下载到内部RAM后，仅设定自变量参数就可编程或擦除，并支持用户分支。
  用户分支：
  编程处理以128字节为单位执行，由外加写入脉冲、读取验证等几个步骤构成；擦除处理以1个分割块为单位执行，由如果干处理步骤构成。在此步骤之间可设定执行用户处理程序，此设定称为带用户分支。
- 由内部RAM产生的闪存仿真功能
  闪存与内部RAM部分重叠，因此可对闪存的改写实时仿真。
- 保护模式
  有通过设定寄存器实现的软件保护模式与通过FWE引脚实现的硬件保护模式等两种，可设定编程/擦除闪存的保护状态。
  检测出编程/擦除失控等异常时，具有转移至错误保护状态，中断编程/擦除处理的功能。
- 编程/擦除时间
  128字节同时编程时，闪存的编程时间为$t_P$ms（Typ.），换算成每个字节为t_P/128ms，擦除时间为每个块为t_Es（Typ.）。
- 改写次数
  闪存可改写N_{WEC}次。
- 编程/擦除时的工作频率
  编程/擦除时的工作频率最大为40MHz（Pϕ）。

20.2 概要

20.2.1 框图

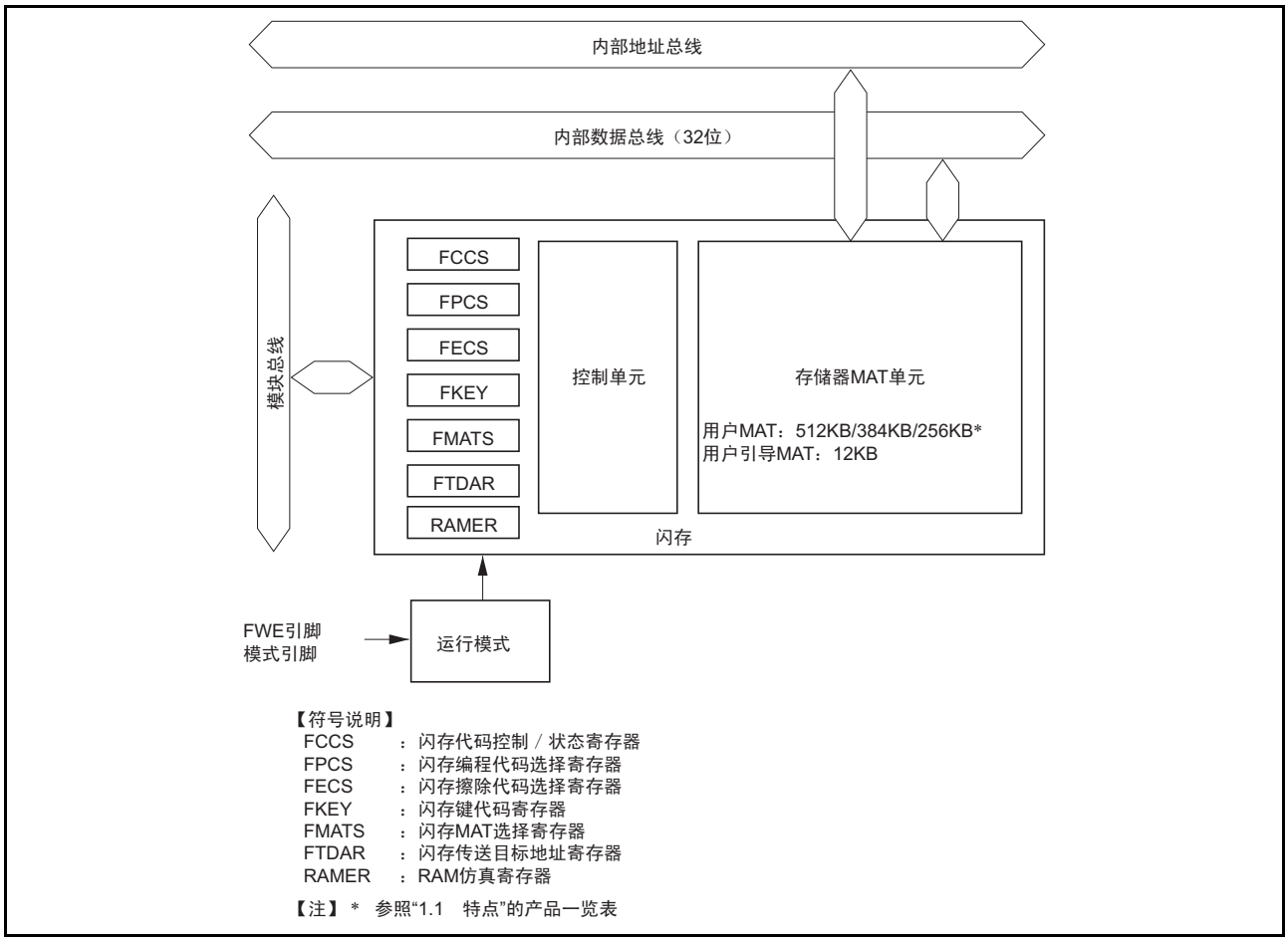


图 20.1 闪存框图

20.2.2 运行模式

在复位状态设定各模式引脚与 FWE 引脚，解除复位时，单片机如图 20.2 所示转移至各运行模式。各模式引脚和 FWE 引脚的设定，详情参照表 20.1。

1. 在ROM无效模式，不可读取/编程/擦除闪存。
   不可写入编程/擦除接口寄存器，读取值总是为H'00。
2. 在用户模式，可读取闪存，但不可编程/擦除。
3. 用户编程模式、用户引导模式、引导模式时，可在板上读取/编程/擦除闪存。
4. 在编程器模式，使用PROM编程器，读取/编程/擦除闪存。

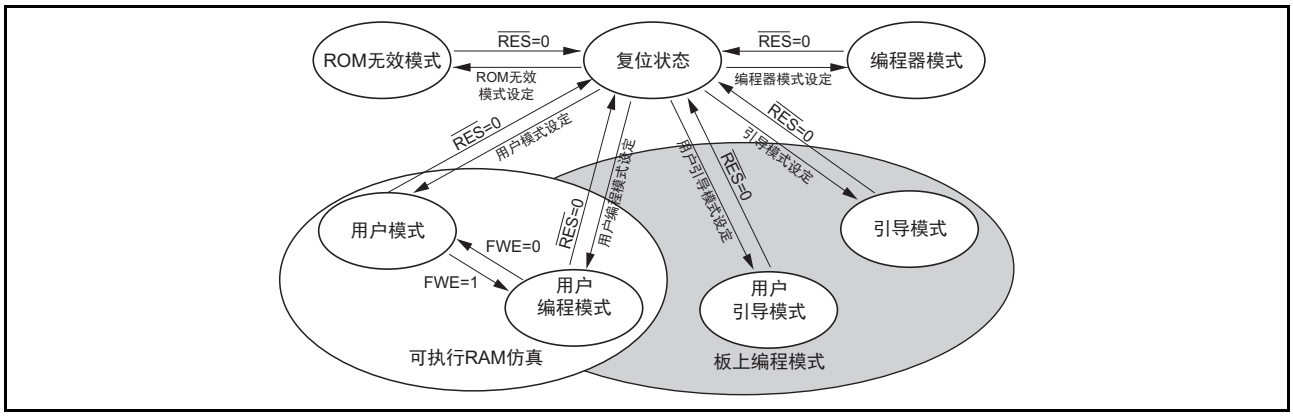


图 20.2 闪存的模式转移图

表 20.1 FWE 引脚、 MD 引脚设定和运行模式

引脚	复位状态	ROM 无效模式	用户模式	用户编程模式	用户引导模式	引导模式	编程器模式
$\overline{\text{RES}}$	0	1	1	1	1	1	依据专用的 PROM 编程器的条件
FWE	0/1	0	0	1	1	1	
MD0	0/1	0/1*1	0/1*2	0/1*2	1	0	
MD1	0/1	0	1	1	0	0	

【注】 *1 MD0=0：8 位宽度的外部总线， MD0=1：不可设定

*2 MD0=0：可使用外部总线， MD0=1：单芯片模式 （不可使用外部总线）

20.2.3 模式比较

引导模式、用户编程模式、用户引导模式及编程器模式的编程 / 擦除相关项目的比较如表 20.2 所示。

表 20.2 编程模式的比较

	引导模式	用户编程模式	用户引导模式	编程器模式
编程 / 擦除环境	板上编程			板外编程
可编程 / 擦除 MAT	用户 MAT 用户引导 MAT	用户 MAT	用户 MAT	用户 MAT 用户引导 MAT
控制编程 / 擦除	命令方式	编程 / 擦除接口	编程 / 擦除接口	—
全面擦除	○（自动）	○	○	○（自动）
块分割擦除	○ *1	○	○	×
传送编程数据	从主机经过 SCI	从任意器件经过 RAM	从任意器件经过 RAM	经过编程器
用户分支功能	×	○	○	×
RAM 仿真	×	○	×	×
复位开始时启动 MAT	嵌入式程序保存 MAT	用户 MAT	用户引导 MAT*2	嵌入式程序保存 MAT
转移至用户模式	更改 & 复位模式设定	更改 FWE 设定	更改 & 复位模式设定	—

【注】 *1 先执行全面擦除，之后，可擦除特定块。

*2 先从嵌入式程序保存 MAT 启动，检查闪存相关寄存器后，从用户引导 MAT 的复位向量启动。

- 仅可在引导模式与编程器模式编程/擦除用户引导MAT。
- 在引导模式，先全面擦除用户MAT与用户引导MAT，之后，可用命令方式编程用户MAT或用户引导MAT，进入此状态前，不可读取MAT的内容。
- 只可编程用户引导MAT，还可在用户引导模式改写用户MAT或不使用用户引导模式只可改写用户MAT等。
- 在用户引导模式，通过设定不同于用户编程模式的模式引脚，可实现任意接口的引导运行。

20.2.4 闪存结构

本 LSI 的闪存由 512KB/384KB/256KB* 的用户 MAT 和 12KB 的用户引导 MAT 构成。

用户 MAT 与用户引导 MAT 分配在起始地址相同的地址，因此在 2 个 MAT 间执行程序或跨越数据存取时，必须由 FMATS 寄存器转换 MAT。

如果为 ROM 有效模式，则在任何模式均可读取用户 MAT/ 用户引导 MAT，但仅可在引导模式与编程器模式改写用户引导 MAT。

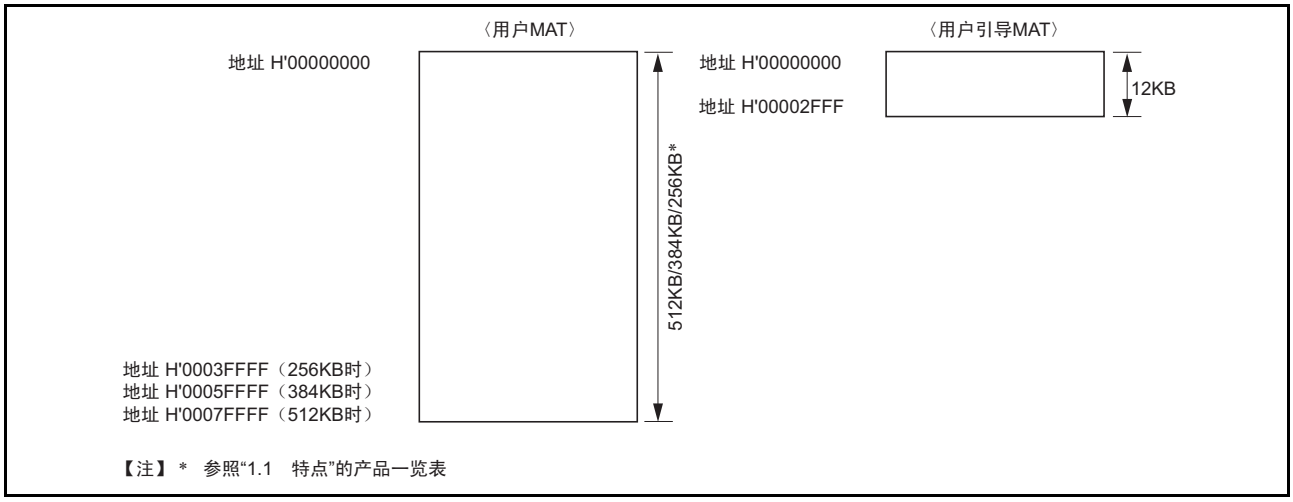


图 20.3 闪存结构图

用户 MAT 和用户引导 MAT 的存储器容量不同。不得存取大于等于 12KB 空间的用户引导 MAT，读取超过 12KB 的用户引导 MAT 时，读取值不定。

20.2.5 块分割

如图 20.4 所示，用户 MAT 分割为 64KB （512KB 产品中有 7 块、 384KB 产品中有 5 块、 256KB 产品中有 3 块）、 32KB （1 块）及 4KB （8 块）。以此分割块为单位擦除时，由 EB0 ～ EB11 的擦除块编号指定。

4KB 分割的 8 块为可执行 RAM 仿真的区域。

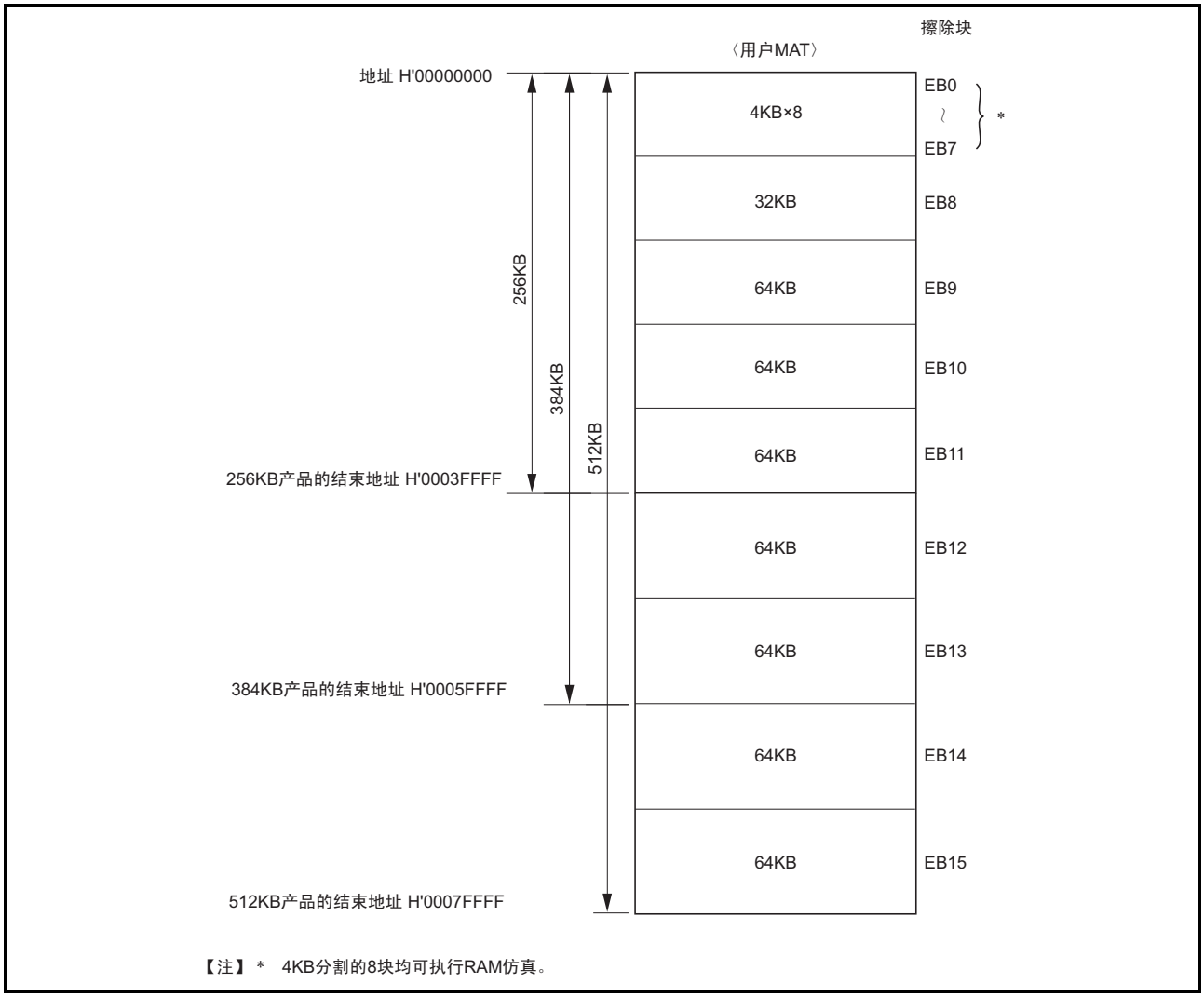


图 20.4 用户 MAT 的块分割

20.2.6 编程 / 擦除接口

将内部程序下载至内部 RAM，由接口寄存器 / 参数指定编程地址 / 数据及擦除块等后，执行编程 / 擦除。

在用户编程模式 / 用户引导模式，由用户编写一系列过程程序。步骤概要如下，详情参阅 “20.5.2 用户编程模式”。

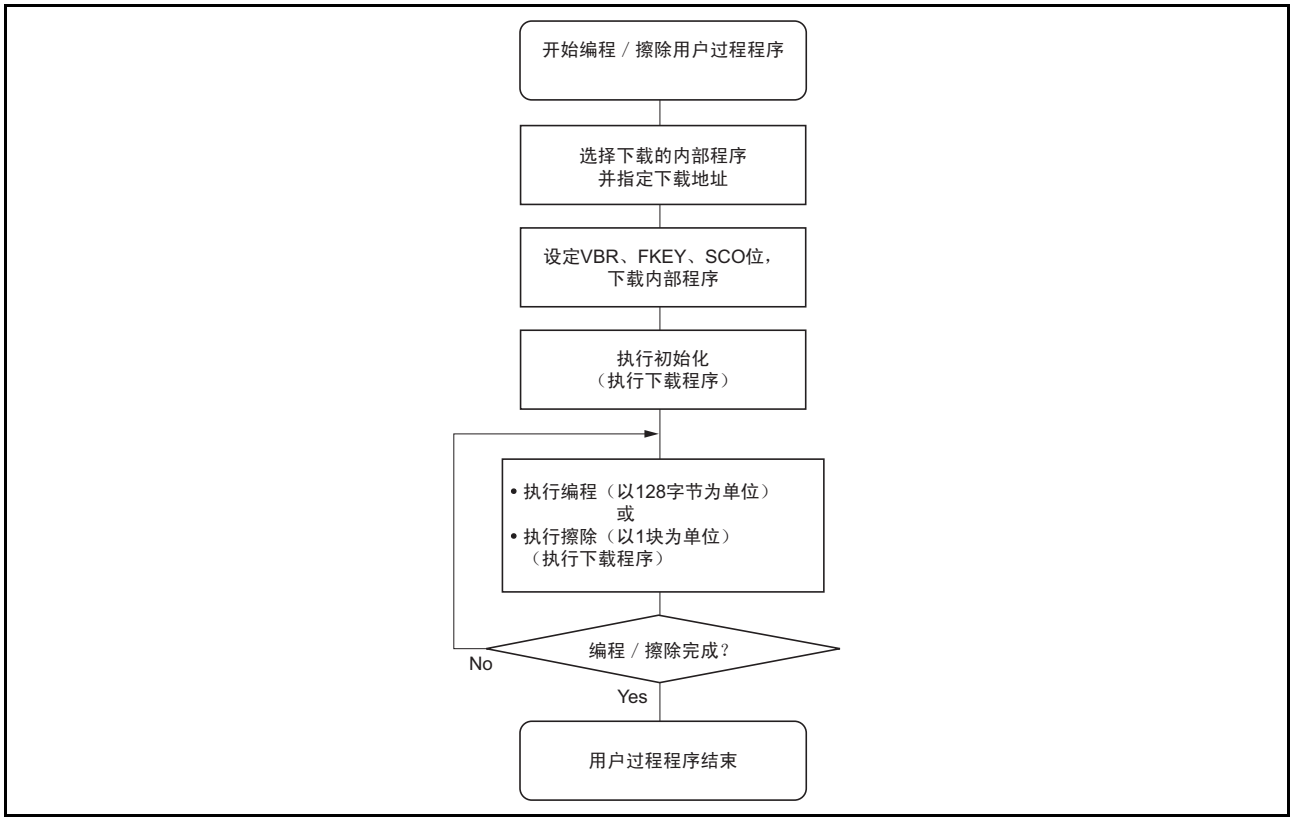


图 20.5 用户过程程序的概要

(1) 下载的内部程序的选择与下载目标的指定

本 LSI 内置编程 / 擦除相关的程序，可下载至内部 RAM，置位编程 / 擦除接口寄存器的对应位，选择要下载的内部程序，并可通过 FTDAR 寄存器指定下载目标。

(2) 内部程序的下载

将 CPU 的 VBR 寄存器设定为 H'84000000 后，通过设定编程 / 擦除接口寄存器的闪存键代码寄存器（FKEY）和闪存代码控制 / 状态寄存器 （FCCS）的 SCO 位，自动下载内部程序。

下载过程中，闪存 MAT 与嵌入式程序保存区域调换。编程 / 擦除时，不可读取闪存，因此必须在闪存以外 （内部 RAM 等）执行下载后到编程 / 擦除结束前的一系列过程程序。

下载结果可返回到编程 / 擦除接口参数，因此可确认是否正常下载。

下载结束后，可更改 VBR。

(3) 编程 / 擦除的初始化

编程 / 擦除前，设定工作频率与用户分支。用户分支地址必须设定在内部闪存区域以外且在下载的内部程序区域以外，并通过编程 / 擦除接口参数设定。

(4) 编程 / 擦除

为了编程 / 擦除，需将 FWE 引脚设定为高电平，并将模式设定为用户编程模式。

编程时，以 128 字节为单位指定编程数据 / 编程地址。

擦除时，以 1 个擦除块为单位指定擦除块。

由编程 / 擦除接口参数设定这些指定，并启动内部程序。由 JSR 指令或 BSR 指令调用 （子程序调用）内部 RAM 的特定地址，执行内部程序。执行结果可返回到编程 / 擦除接口参数。

在闪存编程时，需事先擦除对象区域。

编程 / 擦除处理过程中的中断处理，有限制 / 注意事项。详情参阅 “20.8.2 执行编程 / 擦除过程中的中断”。

(5) 连续执行编程 / 擦除时

编程 128 字节或擦除 1 块仍不可结束处理时，需更新编程地址 / 数据及擦除块编号，并连续编程 / 擦除。

下载的内部程序在处理结束后仍留在内部 RAM，因此连续执行相同处理时，无需下载及初始化。

20.3 输入 / 输出引脚

由表 20.3 所示的引脚控制闪存。

表 20.3 引脚结构

名称	引脚名称	输入 / 输出	功 能
上电复位	$\overline{\text{RES}}$	输入	复位
闪存编程允许	FWE	输入	闪存改写的硬件保护
模式 1	MD1	输入	设定本 LSI 的运行模式
模式 0	MD0	输入	设定本 LSI 的运行模式
发送数据	TXD1 （PA4）	输出	输出串行发送数据 （在引导模式使用）
接收数据	RXD1 （PA3）	输入	输入串行接收数据 （在引导模式使用）

20.4 寄存器说明

20.4.1 寄存器一览表

控制内部闪存有效时的闪存的寄存器 / 参数如表 20.4 所示。

闪存的存取模式有读取 / 写入模式等几种。存储器 MAT 也有用户 MAT 与用户引导 MAT，通过选择各运行模式及 MAT，分配专用的寄存器 / 参数。运行模式与使用寄存器 / 参数的对应关系如表 20.5 所示。

表 20.4 (1) 寄存器结构

寄存器名称	简称 *4	R/W	初始值	地址	存取长度
闪存代码控制 / 状态寄存器	FCCS	R、W*1	H'00*2 H'80*2	H'FFFFCC00	8
闪存编程代码选择寄存器	FPCS	R/W	H'00	H'FFFFCC01	8
闪存擦除代码选择寄存器	FECS	R/W	H'00	H'FFFFCC02	8
闪存键代码寄存器	FKEY	R/W	H'00	H'FFFFCC04	8
闪存 MAT 选择寄存器	FMATS	R/W	H'00*3 H'AA*3	H'FFFFCC05	8
闪存传送目标地址寄存器	FTDAR	R/W	H'00	H'FFFFCC06	8
RAM 仿真寄存器	RAMER	R/W	H'0000	H'FFFFF108	16

【注】 *1 除 SCO 位之外，均为只读位， SCO 位为只写位 （读取值总为 0）。

*2 向 FWE 引脚输入低电平时的 FWE 位初始值为 0。
向FWE引脚输入高电平时的FWE位初始值为1。

*3 在用户模式、用户编程模式启动时的初始值为 H'00。
在用户引导模式启动时的初始值为H'AA。

*4 除 RAMER 寄存器之外的寄存器，仅字节存取有效。
RAMER寄存器可字节/字存取。

表 20.4 (2) 参数结构

参数名称	简称	R/W	初始值	分配	存取长度
下载成功 / 失败结果	DPFR	R/W	不定	内部 RAM*	8、16、32
闪存成功 / 失败结果	FPFR	R/W	不定	CPU 的 R0	8、16、32
闪存多用途地址区域	FMPAR	R/W	不定	CPU 的 R5	8、16、32
闪存多用途数据目标区域	FMPDR	R/W	不定	CPU 的 R4	8、16、32
闪存擦除块选择	FEBS	R/W	不定	CPU 的 R4	8、16、32
闪存编程擦除频率控制	FPEFEQ	R/W	不定	CPU 的 R4	8、16、32
闪存用户分支地址设置	FUBRA	R/W	不定	CPU 的 R5	8、16、32

【注】 * FTDAR 寄存器指定的内部 RAM 区域起始地址的 1 字节有效。

表 20.5　使用的寄存器 / 参数与对象模式

		下载	初始化	编程	擦除	读取	RAM 仿真
编程 / 擦除接口寄存器	FCCS	○	—	—	—	—	—
	FPCS	○	—	—	—	—	—
	FECS	○	—	—	—	—	—
	FKEY	○	—	○	○	—	—
	FMATS	—	—	○ (*1)	○ (*1)	○ (*2)	—
	FTDAR	○	—	—	—	—	—
编程 / 擦除接口参数	DPFR	○	—	—	—	—	—
	FPFR	—	○	○	○	—	—
	FPEFEQ	—	○	—	—	—	—
	FUBRA	—	○	—	—	—	—
	FMPAR	—	—	○	—	—	—
	FMPDR	—	—	○	—	—	—
	FEBS	—	—	—	○	—	—
RAM 仿真	RAMER	—	—	—	—	—	○

【注】 *1　在用户引导模式，编程 / 擦除用户 MAT 时需设定。

*2　根据启动模式与读取对象 MAT 组合，有时需设定。

20.4.2　编程 / 擦除接口寄存器

编程 / 擦除接口寄存器均为 8 位寄存器并只可字节存取。

(1)　闪存代码控制 / 状态寄存器 （FCCS）

FCCS 由 FWE 引脚状态监控位、闪存编程 / 擦除过程中的错误产生监控位及下载内部程序的请求位构成。

位：	7	6	5	4	3	2	1	0
	FWE	MAT	-	FLER	-	-	-	SCO
初始值：	1/0	1/0	0	0	0	0	0	0
R/W：	R	R	R	R	R	R	R	(R)/W

位	位名称	初始值	R/W	说　　明
7	FWE	1/0	R	闪存编程允许位 FWE 位可监控输入到硬件保护闪存编程 / 擦除的 FWE 引脚的电平。根据 FWE 引脚状态，初始值为 0 或 1。 0：向 FWE 引脚输入低电平时 （硬件保护状态） 1：向 FWE 引脚输入高电平时
6	MAT	1/0	R	MAT 位 表示选择用户 MAT 或用户引导 MAT。 0：选择用户 MAT 时 1：选择用户引导 MAT 时
5	—	0	R	保留位 读取值、写入值总是为 0。
4	FLER	0	R	闪存错误 表示编程 / 擦除闪存过程中产生错误。 FLER=1 时，闪存转移至错误保护状态。 FLER=1 时，对闪存内部外加高电压，因此为了降低闪存的损伤，通常在大于 100 μs 的复位输入周期后，解除复位。 0：闪存正常运行。对闪存的编程 / 擦除保护 （错误保护）无效 ［清除条件］ 上电复位或硬件待机模式时 1：表示编程 / 擦除闪存过程中产生错误。对闪存的编程 / 擦除保护（错误保护）有效。 ［置位条件］ 参阅 “20.6.3 错误保护”。
3 ～ 1	—	均为 0	R	保留位 读取值、写入值总是为 0。
0	SCO	0	(R)/W	源程序复制操作 请求将内部改写 / 擦除程序下载至内部 RAM。如果对本位写入 1，则 FPCS/FECS 寄存器选择的内部程序自动下载至 FTDAR 寄存器指定的内部 RAM 区域。为了对本位写入 1，需解除 RAM 仿真状态，并对 FKEY 寄存器写入 H'A5 及在内部 RAM 执行。对本位写入 1 后，必须立即执行 4 条 NOP 指令。 有关下载过程中的中断，详情参阅 “20.8.2 执行编程 / 擦除过程中的中断”，下载时间参阅 “20.8.3 其他注意事项”。下载结束时本位清 0，因此不可读取本位为 1 的状态。 通过 SCO 位下载时，执行伴随内部程序保存区域存储体转换的特殊中断处理，因此请求下载 （SCO=1）前，必须将 VBR 的值设定为 H'84000000。如果不设定 VBR，则会失控。确认下载结束后，可更改 VBR。使用 SCO 功能时，必须使用 FWE 引脚为高电平的模式。 0：不将内部编程 / 擦除程序下载至内部 RAM。 ［清除条件］ 下载结束时清除。 1：产生将内部编程 / 擦除程序下载至内部 RAM 的请求。 ［置位条件］ 在完全满足以下条件的状态写入 1 时 • 对 FKEY 寄存器写入 H'A5 • 在内部 RAM 执行 • 非 RAM 仿真模式 （RAMER 的 RAMS=0）

(2) 闪存编程代码选择寄存器 （FPCS）

FPCS 为选择下载与编程相关的内部程序的寄存器。

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

位	位名称	初始值	R/W	说　　明
7 ～ 1	—	均为 0	R	保留位 读取值、写入值总是为 0。
0	PPVS	0	R/W	单脉冲编程 选择编程程序。 0：不选择内部编程程序。 ［清除条件］ 传送结束时清除。 1：选择内部编程程序。

(3) 闪存擦除代码选择寄存器 （FECS）

FECS 为选择下载与擦除相关的内部程序的寄存器。

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

位	位名称	初始值	R/W	说　　明
7 ～ 1	—	均为 0	R	保留位 读取值、写入值总是为 0。
0	EPVB	0	R/W	擦除脉冲验证块 选择擦除程序。 0：不选择内部擦除程序。 ［清除条件］ 传送结束时清除。 1：选择内部擦除程序。

(4) 闪存键代码寄存器 （FKEY）

FKEY 为允许下载内部程序及编程 / 擦除闪存的软件保护的寄存器。为了下载内部程序，对 SCO 位写入 1 前或执行已下载的编程 / 擦除程序前，如果未写入键代码，则无法执行各处理。

位：	7	6	5	4	3	2	1	0
	K[7:0]							
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	位名称	初始值	R/W	说　　明
7 ～ 0	K[7:0]	均为 0	R/W	键代码 仅在写入 H'A5 时，对 SCO 位的写入才有效。对 FKEY 寄存器写入 H'A5 以外的值时，不可对 SCO 位写入 1，因此不可下载至内部 RAM。 仅在写入 H'5A 时，可编程 / 擦除闪存。即使执行内部编程 / 擦除程序，对 FKEY 寄存器写入除 H'5A 之外的值时，也不可编程 / 擦除闪存。 H'A5：允许写入 SCO 位 （写入除 H'A5 之外的值时，SCO 位不可置位）。 H'5A：允许编程 / 擦除 （写入除 H'5A 之外的值时，为软件保护状态）。 H'00：初始值

(5) 闪存 MAT 选择寄存器 （FMATS）

FMATS 为指定选择用户 MAT 或用户引导 MAT 的寄存器。

位：	7	6	5	4	3	2	1	0
	MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0
初始值：	0/1	0	0/1	0	0/1	0	0/1	0
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说　　明
7	MS7	0/1	R/W	MAT 选择 写入除 H'AA 之外的值时，为用户 MAT 选择状态，写入 H'AA 的状态时为用户引导 MAT 选择状态。通过内部 RAM 的指令对 FMATS 写入值，执行 MAT 转换。 必须依据 “20.8.1 用户 MAT 与用户引导 MAT 的转换” 执行 MAT 转换（即使由 FMATS 选择用户引导 MAT，也不可在用户编程模式改写用户引导 MAT。必须在引导模式或编程器模式改写用户引导 MAT）。 H'AA：选择用户引导 MAT （写入除 H'AA 之外的值时，为用户 MAT 选择状态）且为用户引导模式启动时的初始值。 H'00：为非用户引导模式启动时的初始值 （用户 MAT 选择状态）。 【注】［可编程条件］在内部 RAM 的执行状态
6	MS6	0	R/W	
5	MS5	0/1	R/W	
4	MS4	0	R/W	
3	MS3	0/1	R/W	
2	MS2	0	R/W	
1	MS1	0/1	R/W	
0	MS0	0	R/W	

(6) 闪存传送目标地址寄存器 （FTDAR）

FTDAR 为指定内部程序下载目标的内部 RAM 地址的寄存器。

对 FCCS 寄存器的 SCO 位写入 1 前，必须设定本寄存器。初始值为 H'00，表示内部 RAM 的地址（H'FFFF9000）。

位：	7	6	5	4	3	2	1	0
	TDER	TDA[6:0]						
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	位名称	初始值	R/W	说　　明
7	TDER	0	R/W	传送目标地址设定错误 bit6 ～ 0 （TDA6 ～ TDA0）指定的下载起始地址指定错误时，此位置 1。地址指定错误的判断方法：FCCS 寄存器的 SCO 位置 1，执行下载时，判断 TDA6 ～ TDA0 的值是否在 H'00 ～ H'04 范围内。 SCO 位置 1 前，必须将 FTDAR 寄存器的值设定在 H'00 ～ H'04 范围内，包括将本位的值置 0。 0：TDA6 ～ TDA0 的设定值正常。 1：TDER、 TDA6 ～ TDA0 的设定值为 H'05 ～ H'FF，表示下载中断。
6 ～ 0	TDA[6:0]	均为 0	R/W	传送目标地址 指定下载起始地址。可设定的值为 H'00 ～ H'04，能以 2KB 为单位指定内部 RAM 的下载起始地址。 不得设定 H'05 ～ H'7F 的值。设定该值时，在下载过程中，本寄存器的 bit7：TDER 设定为 1，且不可下载内部程序。 H'00：下载起始地址设定为 H'FFFF9000 H'01：下载起始地址设定为 H'FFFF9800 H'02：下载起始地址设定为 H'FFFFA000 H'03：下载起始地址设定为 H'FFFFA800 H'04：下载起始地址设定为 H'FFFFB000 H'05 ～ H'7F：不得设定。设定时，下载的 bit7：TDER 为 1，并中断下载处理。

20.4.3 编程 / 擦除接口参数

编程 / 擦除接口参数对已下载的内部程序指定工作频率、用户分支目标地址、编程数据保存位置、编程目标地址及擦除块等，并交换处理结果。该参数使用 CPU 的通用寄存器 （R4、 R5 与 R0）与内部 RAM 区域。初始值不定。

下载时，可保存所有的 CPU 寄存器，初始化及执行内部程序时，可保存除 R0 之外的 CPU 寄存器，并在 R0 写入处理结果的返回值。堆栈区用作寄存器的保存或工作区域，处理开始时，需确保堆栈区 （使用的堆栈区最大为 128 字节）。

编程 / 擦除接口参数用于以下 4 项：

1. 下载控制
2. 编程/擦除前的初始化
3. 编程
4. 擦除

各项使用的参数不同，对应关系如表 20.6 所示。

此处，在初始化处理、编程处理及擦除处理时， FPFR 参数返回处理结果，但根据处理内容，位的含义不同。详情参阅各处理的 FPFR 说明。

表 20.6　使用参数与对象模式

参数名称	简称	下载	初始化	编程	擦除	R/W	初始值	分配
下载成功 / 失败结果	DPFR	○	—	—	—	R/W	不定	内部 RAM*
闪存成功 / 失败结果	FPFR	—	○	○	○	R/W	不定	CPU 的 R0
闪存编程擦除频率控制	FPEFEQ	—	○	—	—	R/W	不定	CPU 的 R4
闪存用户分支地址设置	FUBRA	—	○	—	—	R/W	不定	CPU 的 R5
闪存多用途地址区域	FMPAR	—	—	○	—	R/W	不定	CPU 的 R5
闪存多用途数据目标区域	FMPDR	—	—	○	—	R/W	不定	CPU 的 R4
闪存擦除块选择	FEBS	—	—	—	○	R/W	不定	CPU 的 R4

【注】 *　FTDAR 寄存器所指定的下载目标起始地址的 1 字节

(1)　下载控制

SCO 位置 1，可自动下载内部程序。下载的内部 RAM 区域为从 FTDAR 寄存器所指定的起始地址开始的 3KB 的区域。有关内部 RAM 的地址映射，详情参照图 20.10。

通过之前所述的编程 / 擦除接口寄存器设定下载控制，由 DPFR 参数传递返回值。

- 下载成功/失败结果参数 （DPFR：FTDAR寄存器所指定的内部RAM的起始地址的1字节）
  为下载结果的返回值。必须由本参数的值判断下载是否成功。由于无法确认SCO位是否置1，为了可准确判断，必须在下载开始前 （SCO位置1前），将FTDAR寄存器所指定的内部RAM起始地址的1字节设定为下载返回值以外的值 （H'FF等）。有关下载结果的检查方法，详情参阅 “20.5.2(2.5)”。

位：	7	6	5	4	3	2	1	0
	-	-	-	-	-	SS	FK	SF
初始值：	不定	不定	不定	不定	不定	不定	不定	不定
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说　　明
7 ～ 3	—	不定	R/W	未使用位 返回 0。
2	SS	不定	R/W	源选择错误检测位 1 次操作仅可指定 1 种可下载的内部程序，选择两种以上、不选择或选择未映射时，产生错误。 0：下载程序的选择正常 1：产生下载错误 （多种选择或选择未映射）
1	FK	不定	R/W	闪存键寄存器错误检测位 返回检查 FKEY 寄存器的值是否为 H'A5 的结果。 0：FKEY 寄存器的设定值正常 （FKEY=H'A5） 1：FKEY 寄存器的设定值错误 （FKEY 为除 H'A5 之外的值）
0	SF	不定	R/W	成功 / 失败位 返回下载是否正常结束的结果。 0：下载正常结束 （无错误） 1：下载异常结束 （产生错误）

(2) 编程 / 擦除的初始化

可下载的编程 / 擦除内部程序也包含初始化程序。

通过 CPU 指令构成在编程 / 擦除程序过程中所定时间的等待循环，因此需设定 CPU 的工作频率。并支持用户分支功能，因此需设定用户分支目标地址。

初始化程序将这些设定作为已下载的编程 / 擦除程序的参数。

- 闪存程序擦除频率控制参数 （FPEFEQ：CPU的通用寄存器R4）
  为设定CPU工作频率的参数。
  有关本LSI的工作频率范围，详情参阅 “25.3.1 时钟时序”。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值：	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	F15	F14	F13	F12	F11	F10	F9	F8	F7	F6	F5	F4	F3	F2	F1	F0
初始值：	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说　　明
31 ～ 16	—	不定	R/W	未使用位 必须设定为 0。
15 ～ 0	F15 ～ F0	不定	R/W	频率设定位 设定 CPU 的工作频率，如下计算设定值： 1. 将以 MHz 为单位表示的工作频率取到小数点后第 2 位，四舍五入小数点后第 3 位。 2. 将扩大 100 倍的值转换为 2 进制数，写入 FPEFEQ 参数 （通用寄存器 R4）。 CPU 的工作频率为 28.882MHz 时，具体例子如下： 3. 28.882 的小数点后第 3 位四舍五入为 28.88。 4. 将 28.88×100=2888 转换为 2 进制数，并在 R4 设定 B'0000、B'1011、 B'0100、 B'1000 （H'0B48）。

- 闪存用户分支地址设置参数 （FUBRA：CPU的通用寄存器R5）
  为设定用户分支目标地址的参数。可根据编程/擦除时某个确定的处理单位，执行设定的用户程序。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	UA31	UA30	UA29	UA28	UA27	UA26	UA25	UA24	UA23	UA22	UA21	UA20	UA19	UA18	UA17	UA16
初始值：	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UA15	UA14	UA13	UA12	UA11	UA10	UA9	UA8	UA7	UA6	UA5	UA4	UA3	UA2	UA1	UA0
初始值：	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说　　明
31 ～ 0	UA31 ～ UA0	不定	R/W	用户分支目标地址 无需用户分支时，设定 H'00000000。必须将用户分支目标设定在内部闪存以外或传送内部程序的 RAM 区域以外或外部总线空间。注意：为避免失控，不得转移至无执行码的区域，不得损坏下载内部程序的区域及堆栈区。失控及下载区 / 堆栈区损坏时，无法保证闪存的值。用户分支目标处理时，不得启动内部程序的下载、初始化及编程 / 擦除程序，无法保证从用户分支目标返回时的编程 / 擦除。不得改写准备好的编程数据。 必须保存通用寄存器 R8 ～ R15，通用寄存器 R0 ～ R7 无需保存，可直接使用。不得在用户分支目标的处理过程改写编程 / 擦除接口寄存器及转移至 RAM 仿真模式。 用户分支处理结束后，必须通过 RTS 指令返回编程 / 擦除程序。 有关用户分支处理的执行间隔，详情参阅 “20.8.3(2) 用户分支处理的间隔”。

- 闪存成功/失败结果参数 （FPFR：CPU的通用寄存器R0）
  在此说明初始化处理结果返回值FPFR。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值：	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

位	位名称	初始值	R/W	说　　明
31 ～ 3	—	不定	R/W	未使用位 返回 0。
2	BR	不定	R/W	用户分支错误检测位 返回检查指定的用户分支目标地址是否在下载的编程 / 擦除相关程序的保存区域以外的结果。 0：用户分支地址设定值正常 1：用户分支地址设定值异常
1	FQ	不定	R/W	频率错误检测位 返回检查指定的 CPU 工作频率是否在支持的工作频率范围内的结果。 0：工作频率的设定值正常 1：工作频率的设定值异常
0	SF	不定	R/W	成功 / 失败位 返回初始化是否正常结束的结果。 0：初始化正常结束 （无错误） 1：初始化异常结束 （产生错误）

(3)　编程

对闪存执行编程时，需将用户 MAT 的编程目标地址与编程数据传递给已下载的编程程序。

1. 必须在通用寄存器R5设定用户MAT的编程目标起始地址。该参数称为FMPAR （闪存多用途地址区域参数）。
   编程数据总是以128字节为单位，用户MAT编程起始地址的边界即地址的低8位 （A7～A0）必须设定为H'00或H'80。
2. 必须在连续区域准备用户MAT的编程数据，通过CPU的MOV.B指令将编程数据设置在可存取的连续空间，并在内部闪存空间以外。
   即使编程数据未满128字节，也必须填写虚码 （H'FF）构成128字节的编程数据。
   在通用寄存器R4设定保存准备好的编程数据的区域起始地址。该参数称为FMPDR （闪存多用途数据目标区域参数）。

有关编程处理的过程，详情参阅 “20.5.2 用户编程模式”。

- 闪存多用途地址区域参数 （FMPAR：CPU的通用寄存器R5）
  设定用户MAT的编程目标起始地址。
  设定闪存空间以外的地址时，出现错误。
  编程目标起始地址必须在128字节边界，不符合此边界条件时，也会出现错误。此类错误反映在FPFR参数的bit 1：WA位。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MOA31	MOA30	MOA29	MOA28	MOA27	MOA26	MOA25	MOA24	MOA23	MOA22	MOA21	MOA20	MOA19	MOA18	MOA17	MOA16
初始值：	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MOA15	MOA14	MOA13	MOA12	MOA11	MOA10	MOA9	MOA8	MOA7	MOA6	MOA5	MOA4	MOA3	MOA2	MOA1	MOA0
初始值：	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说　　明
31 ～ 0	MOA31 ～ MOA0	不定	R/W	MOA31 ～ MOA0 保存用户 MAT 编程目标起始地址。从此处指定的用户 MAT 起始地址写入连续的 128 字节。因此，指定的编程目标起始地址为 128 字节边界，MOA6 ～ MOA0 总是为 0。

- 闪存多用途数据目标区域参数 （FMPDR：CPU的通用寄存器R4）
  在用户MAT设定保存编程数据的区域的起始地址。编程数据的保存目标在闪存时，出现错误，并反映在FPFR参数的bit 2：WD位。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MOD31	MOD30	MOD29	MOD28	MOD27	MOD26	MOD25	MOD24	MOD23	MOD22	MOD21	MOD20	MOD19	MOD18	MOD17	MOD16
初始值：	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位：	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MOD15	MOD14	MOD13	MOD12	MOD11	MOD10	MOD9	MOD8	MOD7	MOD6	MOD5	MOD4	MOD3	MOD2	MOD1	MOD0
初始值：	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说　　明
31 ～ 0	MOD31 ～ MOD0	不定	R/W	MOD31 ～ MOD0 保存用于存储用户 MAT 编程数据的区域的起始地址。从此处指定的起始地址开始，将连续的 128 字节数据写入用户 MAT。

- 闪存成功/失败结果参数 （FPFR：CPU的通用寄存器R0）
  在此说明编程处理结果返回值FPFR。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值：	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

位	位名称	初始值	R/W	说　　明
31 ～ 7	—	不定	R/W	未使用位 返回 0。
6	MD	不定	R/W	编程模式相关设定错误检测位 返回检查 FWE 引脚输入值为高电平且不在错误保护状态的结果。 FWE 引脚为低电平并在错误保护状态时，写入 1。可由 FCCS 寄存器的 bit7：FWE 及 bit4：FLER 的各位确认这些状态。有关转移至错误保护状态的条件，详情参阅 “20.6.3 错误保护”。 0：FWE、 FLER 状态正常 （FWE=1、 FLER=0） 1：FWE=0 或 FLER=1，为不可编程状态
5	EE	不定	R/W	编程时错误检测位 为了使用户 MAT 不被擦除，不可编程指定数据；或从用户分支处理返回时可改写部分闪存相关的寄存器时，本位返回 1。因此，本位为 1 时，中途可改写用户 MAT 的可能性较高，消除产生错误的原因后，必须从擦除开始重新执行。另外， FMATS 寄存器的值为 H'AA 且为用户引导 MAT 选择状态时，即使编程，也会出现错误。此时，均不可改写用户 MAT 与用户引导 MAT。 必须在引导模式或编程器模式执行用户引导 MAT 的编程。 0：编程处理正常结束 1：编程处理异常结束 （无法保证编程结果）
4	FK	不定	R/W	闪存键寄存器错误检测位 返回编程处理开始前检查 FKEY 寄存器值的结果。 0：FKEY 寄存器的设定值正常 （FKEY=H'5A） 1：FKEY 寄存器的设定值错误 （FKEY 为除 H'5A 之外的值）
3	—	不定	R/W	未使用位 返回 0。
2	WD	不定	R/W	编程数据地址错误检测位 指定闪存区域的地址作为编程数据保存目标起始地址时，出现错误。 0：编程数据地址的设定值正常 1：编程数据地址的设定值异常
1	WA	不定	R/W	编程地址错误检测位 指定以下内容为编程目标起始地址时，出现错误。 • 指定闪存区域以外作为编程目标地址时 • 指定的地址不在 128 字节边界 （A6 ～ A0 不为 0）时 0：编程目标地址的设定值正常 1：编程目标地址的设定值异常
0	SF	不定	R/W	成功 / 失败位 返回编程处理是否正常结束的结果。 0：正常结束 （无错误） 1：异常结束 （产生错误）

(4) 擦除

擦除闪存时，需将用户 MAT 擦除块编号传递至已下载的擦除程序，并设定到 FEBS 参数（通用寄存器 R4）。

从 0 ～ 15 的块编号指定 1 块。

有关擦除处理的过程，详情参阅 “20.5.2 用户编程模式”。

- 闪存擦除块选择参数（FEBS：CPU的通用寄存器R4）
  指定擦除块编号，不可指定多个块编号。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值：	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

位	位名称	初始值	R/W	说　　明
31 ～ 8	—	不定	R/W	未使用位 必须设定为 0。
7 ～ 0	EBS[7:0]	不定	R/W	• 闪存为 512KB 时 在 0 ～ 15 的范围内设定擦除块编号。 0 对应 EB0 块， 15 对应 EB15 块。如果设定为 0 ～ 15 （H'00 ～ H'0F）以外，则出现错误。 • 闪存为 384KB 时 在 0 ～ 13 的范围内设定擦除块编号。 0 对应 EB0 块， 13 对应 EB13 块。如果设定为 0 ～ 13 （H'00 ～ H'0D）以外，则出现错误。 • 闪存为 256KB 时 在 0 ～ 11 的范围内设定擦除块编号。 0 对应 EB0 块， 11 对应 EB11 块。如果设定为 0 ～ 11 （H'00 ～ H'0B）以外，则出现错误。

- 闪存成功/失败结果参数（FPFR：CPU的通用寄存器R0）
  在此说明擦除处理结果返回值FPFR。

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值：	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W：	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

位	位名称	初始值	R/W	说　　明
31 ～ 7	—	不定	R/W	未使用位 返回 0。
6	MD	不定	R/W	擦除模式相关设定错误检测位 返回检查 FWE 引脚的输入值为高电平且不在错误保护状态的结果。 FWE 引脚为低电平并在错误保护状态时，写入 1。可由 FCCS 寄存器的 bit7：FWE 及 bit4：FLER 的各位确认这些状态。有关转移至错误保护状态的条件，详情参阅 “20.6.3 错误保护”。 0：FWE、 FLER 状态正常 （FWE=1、 FLER=0） 1：FWE=0 或 FLER=1，为不可擦除状态
5	EE	不定	R/W	擦除时错误检测位 不可擦除用户 MAT 或从用户分支处理返回时可改写部分闪存相关的寄存器时，本位返回 1。因此本位为 1 时，中途可擦除用户 MAT 的可能性较高，消除产生错误的原因后，必须从擦除开始重新执行。 FMATS 寄存器的值为 H'AA 并在用户引导 MAT 选择状态时，即使擦除，也出现错误。此时，用户 MAT 与用户引导 MAT 均不可擦除。必须在引导模式或编程器模式擦除用户引导 MAT。 0：擦除处理正常结束 1：擦除处理异常结束 （无法保证擦除结果）
4	FK	不定	R/W	闪存键寄存器错误检测位 返回擦除处理开始前检查 FKEY 寄存器值的结果。 0：FKEY 寄存器的设定值正常 （FKEY=H'5A） 1：FKEY 寄存器的设定值错误 （FKEY 为除 H'5A 之外的值）
3	EB	不定	R/W	擦除块选择错误检测位 返回检查指定的擦除块编号是否在用户 MAT 的块范围内的结果。 0：擦除块编号的设定值正常 1：擦除块编号的设定值异常
2、 1	—	不定	R/W	未使用位 返回 0。
0	SF	不定	R/W	成功 / 失败位 返回擦除处理是否正常结束的结果。 0：正常结束 （无错误） 1：异常结束 （产生错误）

20.4.4 RAM 仿真寄存器 （RAMER）

RAMER 为仿真用户 MAT 的实时改写时，设定与内部 RAM 部分重叠的用户 MAT 区域的寄存器。必须在用户模式及用户编程模式执行 RAM 仿真。

有关用户 MAT 区域的分割方法，详情参照表 20.7。为了确保仿真功能准确运行，改写本寄存器后，不得立即存取 RAM 仿真的对象 MAT，否则，无法保证正常存取。

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

位	位名称	初始值	R/W	说　　明
15 ～ 4	—	均为 0	R	保留位 读取值、写入值总是为 0。
3	RAMS	0	R/W	RAM 选择 设定是否选择通过 RAM 仿真用户 MAT。 RAMS=1 时，用户 MAT 所有块均为编程 / 擦除保护状态。 0：不选择仿真 用户 MAT 所有块的编程 / 擦除保护无效 1：选择仿真 用户 MAT 所有块的编程 / 擦除保护有效
2 ～ 0	RAM[2:0]	000	R/W	选择用户 MAT 区域 与 bit3 一起使用，选择与内部 RAM 重叠的用户 MAT 区域。（参照表 20.7）

表 20.7 RAM 区域与用户 MAT 区域的重叠

RAM 区域	块名称	RAMS	RAM2	RAM1	RAM0
H'FFFFA000 ～ H'FFFFAFFF	RAM 区域 （4KB）	0	x	x	x
H'00000000 ～ H'00000FFF	EB0 （4KB）	1	0	0	0
H'00001000 ～ H'00001FFF	EB1 （4KB）	1	0	0	1
H'00002000 ～ H'00002FFF	EB2 （4KB）	1	0	1	0
H'00003000 ～ H'00003FFF	EB3 （4KB）	1	0	1	1
H'00004000 ～ H'00004FFF	EB4 （4KB）	1	1	0	0
H'00005000 ～ H'00005FFF	EB5 （4KB）	1	1	0	1
H'00006000 ～ H'00006FFF	EB6 （4KB）	1	1	1	0
H'00007000 ～ H'00007FFF	EB7 （4KB）	1	1	1	1

【注】 x：Don't care

20.5 板上编程模式

在板上编程模式设定引脚并开始复位时，转移至可编程 / 擦除内部闪存的板上编程状态。板上编程模式有用户编程模式、用户引导模式及引导模式等 3 种。

有关转移至各模式的引脚设定方法，详情参照表 20.1。有关闪存的各模式状态转移，详情参照图 20.2。

20.5.1 引导模式

引导模式是通过使用内部 SCI 并从主机发送控制命令与编程数据的方式，编程 / 擦除用户 MAT 与用户引导 MAT 的模式。需预先在主机准备发送控制命令的工具与编程数据，并设定使用的 SCI 通信模式为异步模式。如果本 LSI 的引脚设定为引导模式后开始复位，则启动预先嵌入单片机内部的引导程序，SCI 位速率自动调整后，通过控制命令方式与主机进行通信。

引导模式时的系统结构如图 20.6 所示。引导模式的引脚设定参照表 20.1。虽然可忽略引导模式的 NMI 及其他中断，但还是要避免产生此中断。必须注意：引导模式运行过程中不可使用 AUD。

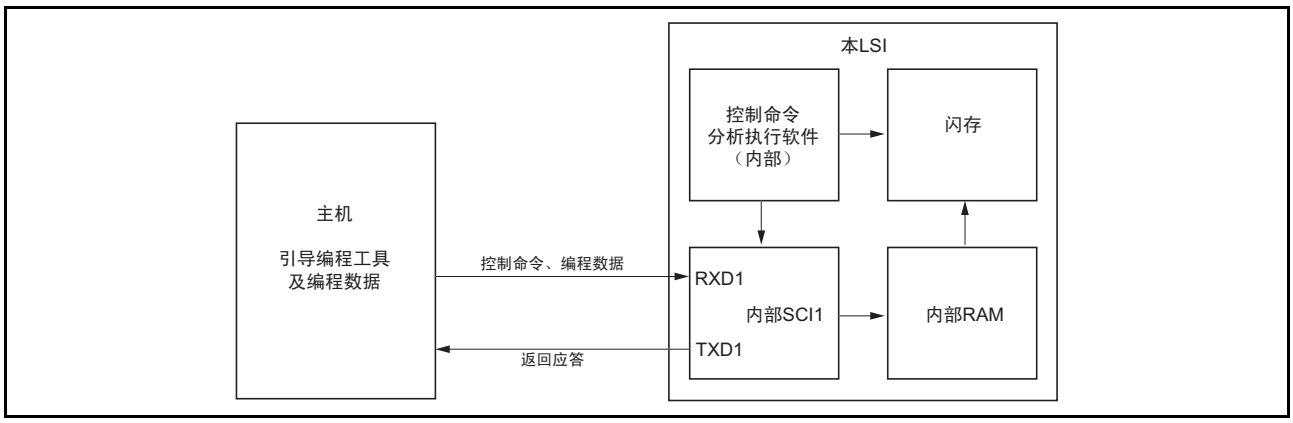


图 20.6 引导模式时的系统结构图

(1) 主机的 SCI 接口设定

启动引导模式时，本 LSI 测量从主机连续发送的异步 SCI 通信数据 （H'00）的低电平期间。必须将此时的 SCI 发送 / 接收格式设定为 “8 位数据、 1 个停止位、无奇偶校验”。本 LSI 根据测量的低电平期间计算主机发送的位速率，并向主机发送位调整结束信号 （H'00 为 1 字节）。主机必须确认已正常接收该调整结束信号（H'00），并向本 LSI 发送 1 字节的 H'55。不可正常接收时，必须再次启动引导模式 （复位），执行上述操作。根据主机发送的位速率及本 LSI 的系统时钟频率，主机与本 LSI 的位速率会产生误差。为了使 SCI 正常运行，必须将主机的传送位速率设定为 9,600bps 或 19,200bps。

主机的传送位速率与可自动调整本 LSI 位速率的系统时钟频率如表 20.8 所示。必须在该系统时钟范围内启动引导模式，在引导模式，不支持各内部时钟的分频比为 ×1/3 倍的设定。

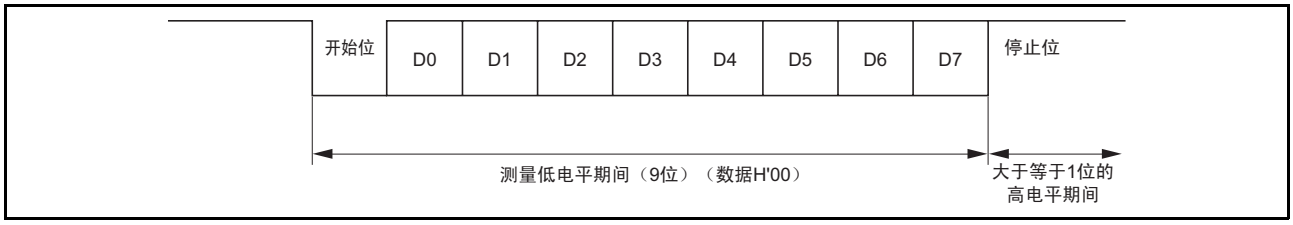


图 20.7 SCI 位速率的自动调整

表 20.8 可自动调整本 LSI 的系统时钟频率

主机的位速率	可自动调整本 LSI 位速率的外围时钟（Pϕ）频率
9,600bps	10 ～ 40MHz（T_{opr}= –40 ～ +85°C） 10 ～ 32MHz（T_{opr}= –40 ～ +125°C）
19,200bps	10 ～ 40MHz（T_{opr}= –40 ～ +85°C） 10 ～ 32MHz（T_{opr}= –40 ～ +125°C）

【注】 在引导模式，不支持各内部时钟的分频比为 ×1/3 倍的设定。

(2) 状态转移图

启动引导模式后的状态转移概要如图 20.8 所示。有关引导模式，详情参阅 “20.9.1 引导模式的标准串行通信接口规格”。

1. 位速率调整
   启动引导模式后，调整主机与SCI接口的位速率。
2. 等待查询选择命令
   将用户MAT容量、用户MAT结构、 MAT起始地址及支持信息等查询所需的信息发送至主机。
3. 所有用户MAT及用户引导MAT的自动擦除
   查询选择结束并发送编程擦除状态转移命令时，自动擦除所有用户MAT与用户引导MAT。
4. 等待编程/擦除命令

- 接收 “编程选择命令”时，转移至编程数据等待状态。必须向编程命令连续发送编程起始地址及编程数据。编程结束时，必须将编程起始地址设定为H'FFFFFFFF并发送。由此，从编程数据的等待状态返回编程/擦除命令的等待状态。
- 接收 “擦除选择命令”时，转移至擦除块数据等待状态。必须向擦除命令连续发送擦除块编号。擦除结束时，必须将擦除块编号设定为H'FF并发送。由此，从擦除块数据的等待状态返回编程/擦除命令等待状态。在引导模式编程后不启动复位，只改写特定块时执行擦除。 1次操作就可编程时，在转移至编程/擦除/其他命令等待状态之前擦除所有块，因此无需执行本次擦除操作。
- 除编程/擦除之外，还有用户MAT/用户引导MAT的和校验、用户MAT/用户引导MAT的空白检查（擦除检查）、用户MAT/用户引导MAT的存储器读取及获取当前状态信息的命令。

必须注意：读取用户 MAT/ 用户引导 MAT 的存储器时，在自动擦除所有用户 MAT/ 用户引导 MAT 后，只可读取已编程的数据。

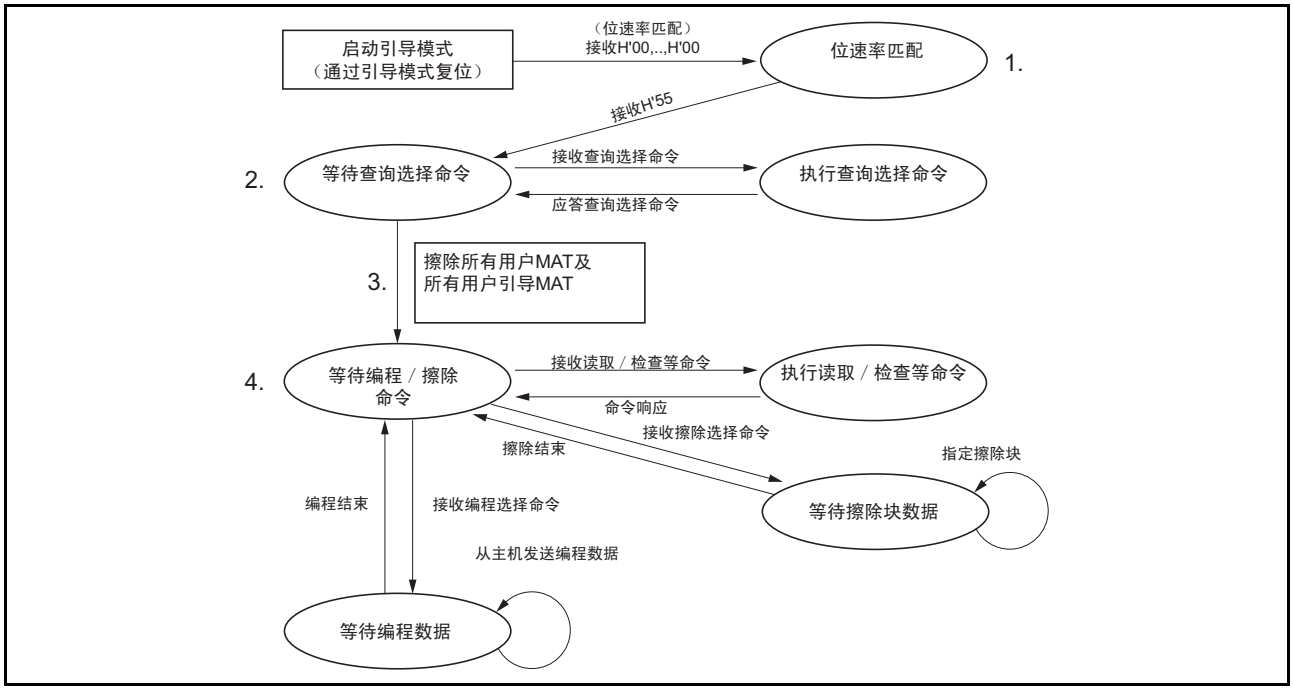


图 20.8 引导模式的状态转移概要图

20.5.2 用户编程模式

在用户编程模式，可编程 / 擦除用户 MAT （不可编程 / 擦除用户引导 MAT）。

预先下载单片机的内部程序后编程 / 擦除。

概要流程如图 20.9 所示。

编程 / 擦除处理过程中对闪存内部外加高电压，因此不得转移至复位及硬件待机，否则可能损坏闪存。错误复位时，必须在长于正常复位输入期间即 100 μs 后解除复位。

编程步骤参阅后述的 “(2) 用户编程模式的编程步骤”，擦除步骤参阅 “(3) 在用户编程模式擦除的步骤”。

使用 FTDAR 寄存器，将编程 / 擦除程序下载至各自的内部 RAM 区域，有关重复执行擦除与编程处理的概要，详情参阅 “(4) 用户编程模式的擦除 / 编程步骤”。

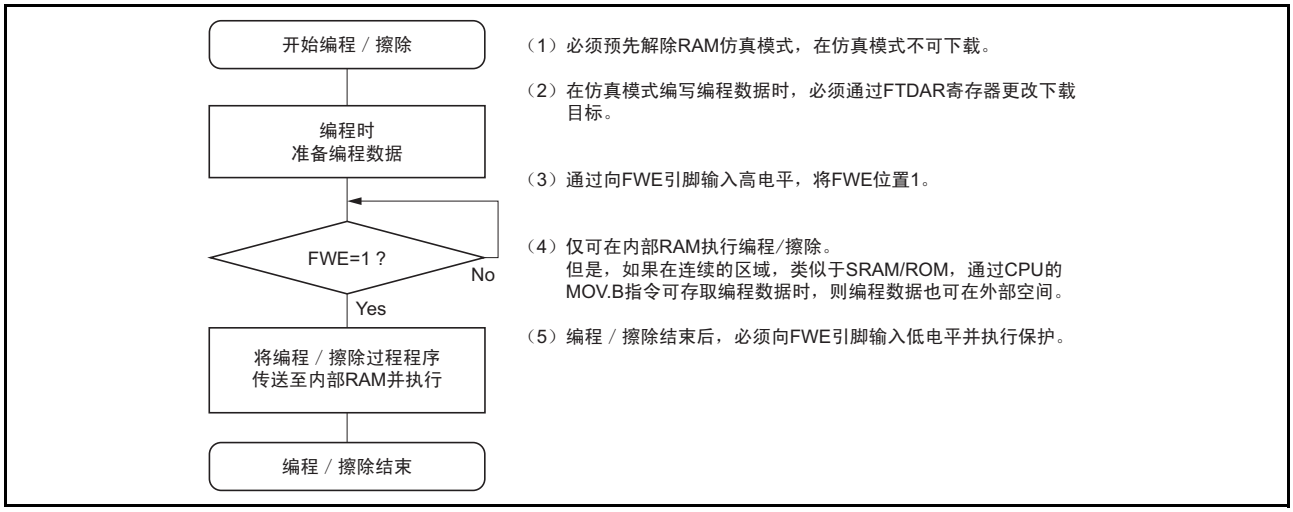


图 20.9 编程 / 擦除概要流程

(1) 编程 / 擦除时的内部 RAM 地址映射

必须在内部 RAM 执行由用户编写的下载请求、编程 / 擦除步骤及结果判断等过程程序的一部分。下载的内部程序全部存在于内部 RAM。为了避免重复，必须注意内部 RAM 区的管理。

下载的程序区域如图 20.10 所示。

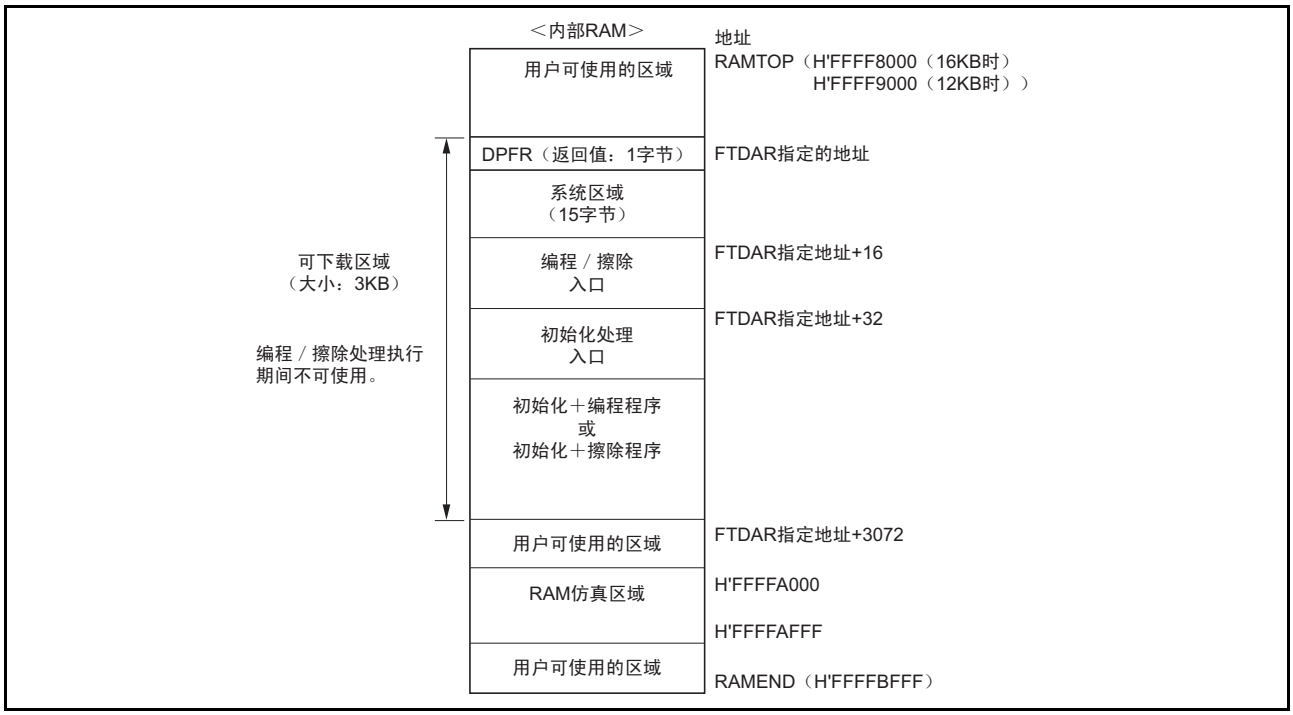


图 20.10 下载后的内部 RAM 映射

(2) 用户编程模式的编程步骤

下载、初始化及编程步骤如图 20.11 所示。

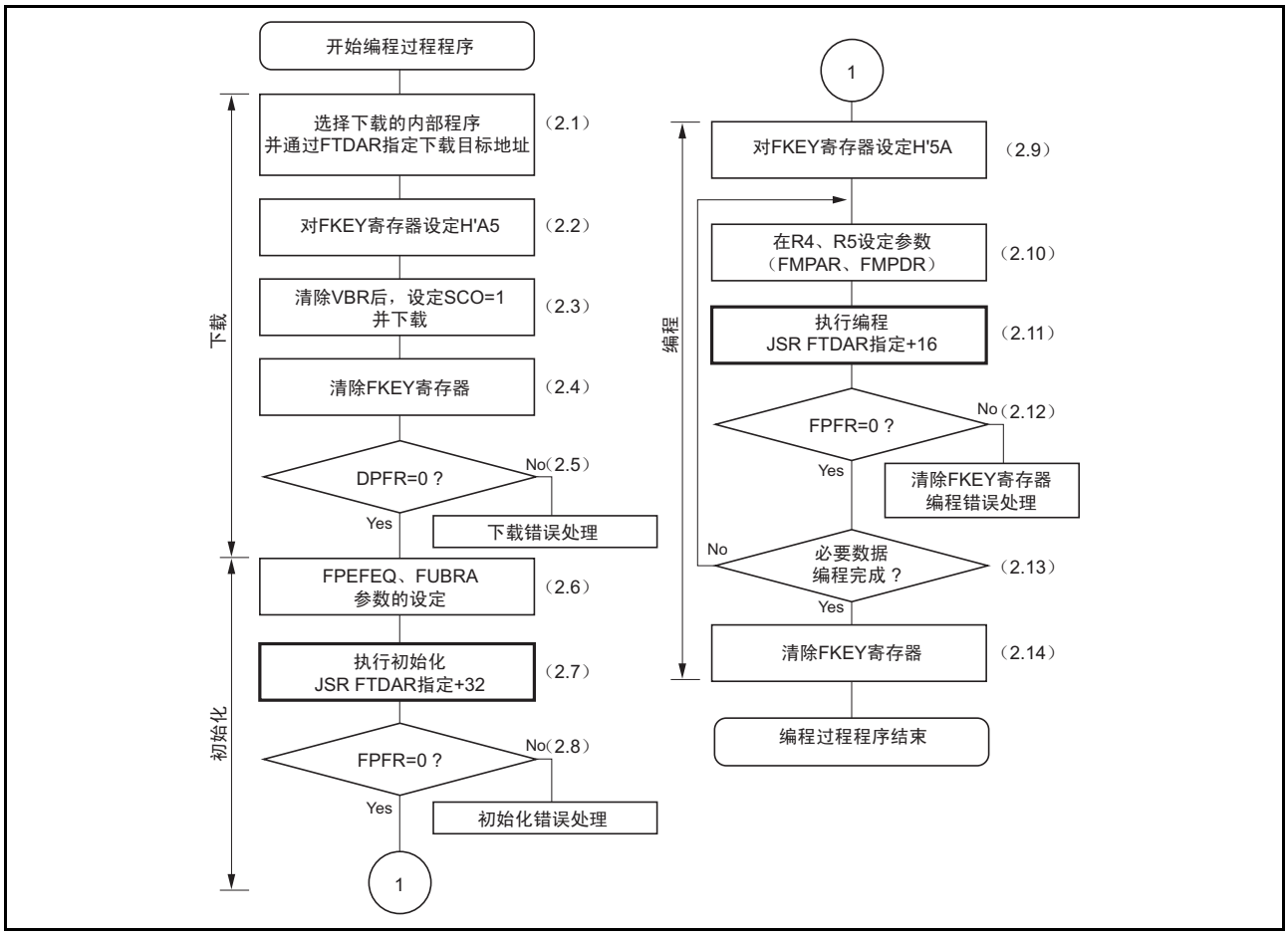


图 20.11 编程步骤

以下详细说明编程步骤。必须在编程对象的闪存以外区域执行过程程序。特别是为了下载而将 FCCS 寄存器的 SCO 位设定为 1 的部分，必须在内部 RAM 执行。且频率控制寄存器 （FRQCR）设定的内部时钟（Iϕ）、总线时钟 （Bϕ）、外围时钟 （Pϕ）的分频比必须全部设定为 1/4 （初始值）。

编程 / 擦除程序的下载结束及 SCO 位清 0 后，可更改频率控制寄存器 （FRQCR）的设定值为任意值。

可单步执行用户过程程序的区域 （内部 RAM、用户 MAT、外部空间等），详情参阅 “20.9.2 可保存过程程序或编程数据的区域”。

以下说明的前提是可擦除用户 MAT 的编程对象区域并在连续区域准备好编程数据。未擦除时，必须在编程前擦除。

1 次编程处理可编程 128 字节，超过 128 字节时，以 128 字节为单位更新编程目标地址 / 编程数据的参数并重复编程。

编程未满 128 字节时，需填写无效数据凑齐 128 字节。填写的无效数据为 H'FF 时，可缩短编程处理时间。

（2.1）选择下载的内部程序并指定下载目标。

FPCS 寄存器的 PPVS 位设定为 1 时，选择编程程序。

不可选择多个编程 / 擦除程序。设定为多个时，不可下载，向 DPFR 参数的源选择检测位 （SS）报告下载错误。

由 FTDAR 寄存器指定下载目标的起始地址。

（2.2）对 FKEY 寄存器写入 H'A5。

为了实现保护功能而不对 FKEY 寄存器写入 H'A5 时，不可对请求下载的 SCO 位写入 1。

（2.3）通过设定 VBR 寄存器及对 FCCS 寄存器的 SCO 位写入 1，执行下载。

SCO 位置位前，必须将 VBR 寄存器设定为 H'84000000。

为了对 SCO 位写入 1，需满足以下所有条件：

1. 解除RAM仿真模式。
2. 对FKEY寄存器写入H'A5。
3. 在内部RAM写入SCO位。

如果 SCO 位为 1，则开始自动下载，返回用户过程程序时， SCO 位清 0，因此在用户过程程序不可确认 SCO=1。

仅通过 DPFR 参数的返回值确认下载结果， SCO=1 前，必须将作为 DPFR 参数的 FTDAR 所指定的内部 RAM 起始的 1 字节设定为返回值以外 （H'FF 等）的值，以防产生误判断。

下载时，如下所示，将伴随存储体转换的特殊中断处理作为单片机内部处理并执行，因此需将 VBR 设定为 H'84000000。设定 SCO=1 的指令后，必须立即执行 4 条 NOP 指令。

1. 用户MAT空间转换为内部程序保存区域。
2. 检查下载程序选择条件与FTDAR指定的地址后，传送至FTDAR指定的内部RAM。
3. FPCS寄存器、 FECS寄存器及FCCS寄存器的SCO位清0。
4. 返回值设定为DPFR参数。
5. 内部程序保存区域返回至用户MAT空间后，返回用户过程程序。
   下载结束并返回用户过程程序后，可重新设定VBR。

下载时的注意事项如下所述：

下载时，可保存 CPU 通用寄存器的值。

下载过程中，不得产生中断。有关下载与中断，详情参阅 “20.8.2 执行编程 / 擦除过程中的中断”。

最大可使用 128 字节的堆栈区，因此 SCO=1 前，必须确保堆栈区大于等于 128 字节。

下载过程中，通过 DTC 存取闪存时，无法保证运行，因此必须注意不得通过 DTC 进行存取。

（2.4）为了实现保护功能需将 FKEY 寄存器清除为 H'00。

（2.5）检查 DPFR 参数的值并确认下载结果。

推荐使用以下方法确认下载结果：

1. 检查DPFR参数 （FTDAR指定的下载目标的起始地址的1字节）值。如果为H'00，则可正常下载；为H'00以外的值时，可按以下步骤调查不可下载的原因。
2. DPFR参数的值与下载前的设定值 （H'FF等）相同时，考虑到FTDAR的下载目标地址设定异常，因此必须确认FTDAR的bit7：TDER位。
3. DPFR参数的值与下载前的设定值不同时，必须由DPFR参数的bit2：SS位及bit1：FK位确认下载程序的选择及FKEY寄存器的设定是否正常。

（2.6）为了初始化，将工作频率与用户分支目标设定给 FPEFEQ 与 FUBRA 参数。

1. 将当前的CPU时钟频率设定给FPEFEQ参数（通用寄存器：R4）。有关FPEFEQ参数的可设定范围，详情参阅“25.3.1 时钟时序”。
   设定此范围以外的频率时，向初始化程序的FPFR参数报告错误，不执行初始化。有关频率的设定方法，详情参阅“20.4.3 编程/擦除接口参数”的“• 闪存程序擦除频率控制参数（FPEFEQ：CPU的通用寄存器R4）”。
2. 将用户分支目标的起始地址设定给FUBRA参数（通用寄存器：R5）。
   无需用户分支处理时，必须将FUBRA设定为0。
   执行用户分支时，分支目标必须在编程对象的闪存以外。并且不可设定在下载的内部程序区域。
   必须由RTS指令从用户分支处理返回至编程处理。
   详情参阅“20.4.3 编程/擦除接口参数”的“• 闪存用户分支地址设置参数（FUBRA：CPU的通用寄存器R5）”。

（2.7）初始化的执行

下载编程程序时，与初始化程序一起下载至内部 RAM。在 FTDAR 设定的下载起始地址 +32 字节的区域有初始化程序的入口点，因此必须根据以下方法调用子程序并执行初始化。

```
MOV.L    #DLTOP+32,R1    ;    将入口地址设定至 R1
JSR      @R1             ;    调用初始化程序
NOP
```

1. 在初始化程序保存除R0之外的通用寄存器。
2. R0的值为FPFR参数的返回值。
3. 在初始化程序使用堆栈区，因此必须预先确保RAM有大于等于128字节的堆栈区域。
4. 可接受初始化程序执行过程中的中断，但不得破坏内部RAM的程序保存区域、堆栈区域及寄存器的值。

（2.8）判断初始化程序的返回值 FPFR（通用寄存器 R0）。

（2.9）为了可编程用户 MAT，必须对 FKEY 寄存器设定 H'5A。

（2.10）设定编程所需的参数。

将用户 MAT 编程目标的起始地址（FMPAR）设定至通用寄存器 R5，编程数据保存区的起始地址（FMPDR）设定至通用寄存器 R4。

1. 设定FMPAR
   FMPAR指定编程起始地址，因此指定用户MAT区域以外的地址时，即使执行编程程序也不可编程，向返回值参数FPFR报告错误。并且，编程时以128字节为单位，因此低8位（MOA7～MOA0）需在H'00或H'80的128字节边界。
2. 设定FMPDR
   编程数据的保存目标在闪存时，即使执行编程程序也不可编程，向FPFR参数报告错误。此时，先传送至内部RAM后再执行编程。

（2.11）编程处理

在FTDAR指定的下载目标的起始地址+16字节的区域有编程程序的入口点，因此必须根据以下方法调用子程序并执行编程处理。

```
MOV.L        #DLTOP+16,R1     ;        将入口地址设定至 R1
JSR          @R1              ;        调用编程程序
NOP
```

1. 在编程程序保存除R0之外的通用寄存器。
2. R0的值为FPFR参数的返回值。
3. 在编程程序使用堆栈区，因此必须预先确保RAM有大于等于128字节的堆栈区。

（2.12）判断编程程序的返回值 FPFR （通用寄存器 R0）。

（2.13）判断必要的数据编程是否完成。

编程超过 128 字节的数据时，以 128 字节为单位，更新 FMPAR、 FMPDR 的设定并重复上述 （2.10）～（2.13）的处理。必须正确执行编程目标地址的 128 字节的递增和编程数据指针的更新。对已编程地址重复编程时，不仅会引起编程错误，还会对闪存造成损伤。

（2.14）必须在编程结束后清除 FKEY 寄存器并设置软件保护。

对用户 MAT 完成编程后，立即由上电复位重新启动时，必须设置长于普通复位的执行期间，即大于等于 100μs （$\overline{\text{RES}}$=0 的期间）。

(3) 在用户编程模式擦除的步骤

下载、初始化及擦除步骤如图 20.12 所示。

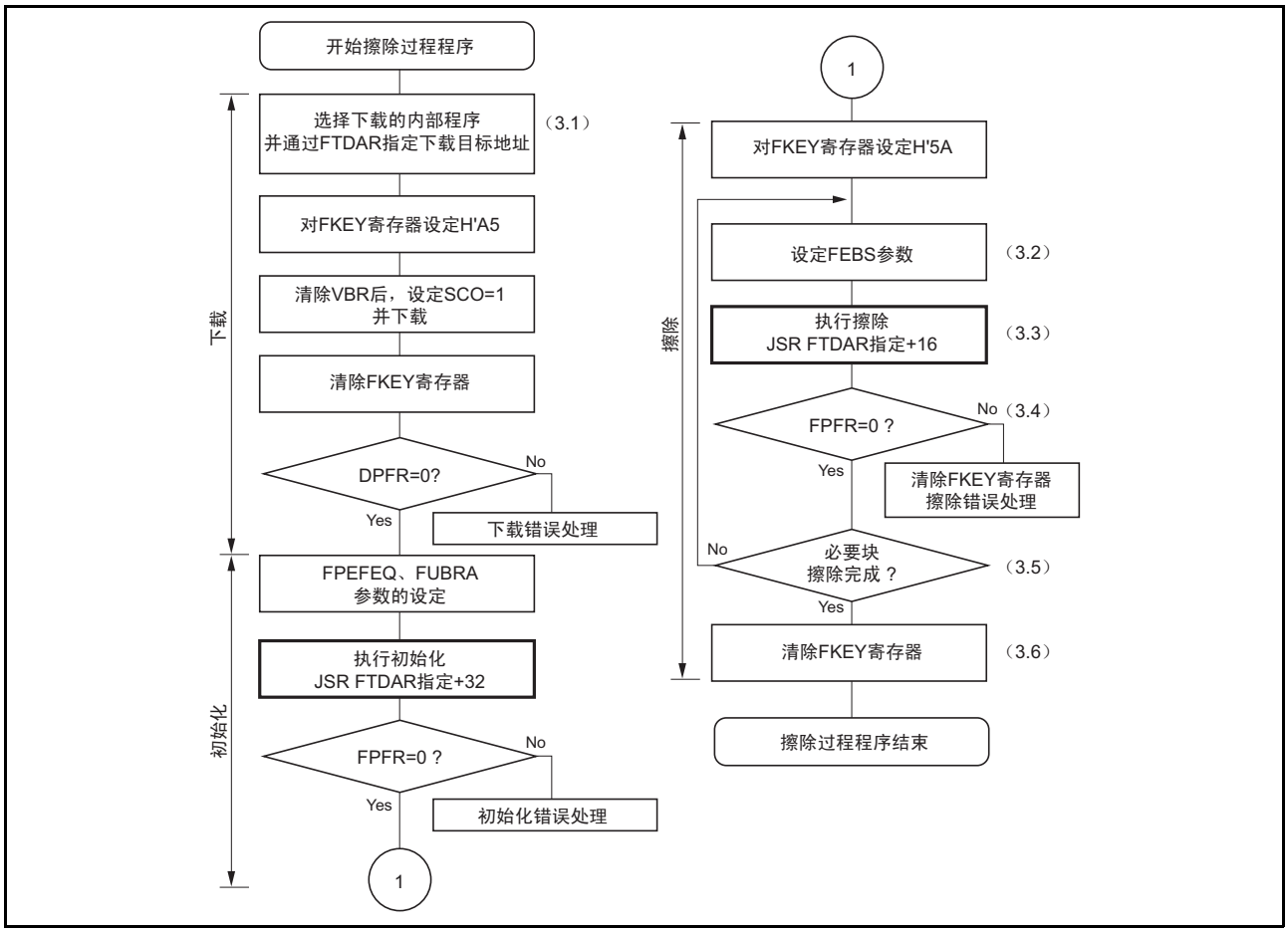


图 20.12 擦除步骤

以下详细说明擦除步骤。必须在擦除对象的闪存以外执行步骤程序。

特别是为了下载将 FCCS 寄存器的 SCO 位设定为 1 的部分，必须在内部 RAM 执行，且频率控制寄存器（FRQCR）的内部时钟 （Iϕ）、总线时钟 （Bϕ）、外围时钟 （Pϕ）的分频比必须全部设定为 ×1/4 （初始值）。

编程 / 擦除程序的下载结束及 SCO 位清 0 后，可更改频率控制寄存器 （FRQCR）的设定值为任意值。

可单步执行用户过程程序的区域 （内部 RAM、用户 MAT、外部空间等），详情参阅 “20.9.2 可保存过程程序或编程数据的区域”。

有关下载的内部程序区域，详情参照图 20.10 的下载后的内部 RAM 映射。

1 次擦除处理可擦除 1 个分割块。有关块分割，详情参照图 20.4。擦除不少于 2 个的分割块时，更新擦除块编号并重复擦除。

（3.1）选择下载的内部程序并指定下载目标地址。

将 FECS 寄存器的 EPVB 位设定为 1。

不可选择多个编程 / 擦除程序，设定为多个时，不可下载，向 DPFR 参数的源选择错误检测位 （SS）报告下载错误。

由 FTDAR 寄存器指定下载目标的起始地址。

FKEY 寄存器设定后的下载、初始化等步骤与编程步骤相同，详情参阅 “20.5.2(2) 用户编程模式的编程步骤”。

设定擦除程序参数之后的步骤如下。

（3.2）设定擦除所需的 FEBS 参数。

将用户 MAT 的擦除块编号设定给闪存擦除块选择参数 FEBS （通用寄存器 R4）。设定为用户 MAT 分割块编号以外的值时，即使执行擦除处理程序也不擦除，并向返回值参数 FPFR 报告错误。

（3.3）擦除处理

与编程相同，在 FTDAR 指定的下载目标的起始地址 +16 字节的区域有擦除程序的入口点，因此必须根据以下方法调用子程序并执行擦除。

```
MOV.L    #DLTOP+16,R1    ;    将设定入口地址至 R1
JSR       @R1            ;    调用擦除程序
NOP
```

1. 在擦除程序保存除R0之外的通用寄存器。
2. R0的值为FPFR参数的返回值。
3. 在擦除程序使用堆栈区，因此必须预先确保RAM有大于等于128字节的堆栈区。

（3.4）判断擦除程序的返回值 FPFR （通用寄存器 R0）。

（3.5）判断必要的块擦除是否完成。

擦除多个块时，更新设定 FEBS 参数并重复上述 （3.2）～ （3.5）的处理，可擦除已擦除完的块。

（3.6）必须在擦除结束后清除 FKEY 寄存器并设置软件保护。

擦除用户 MAT 后，立即由上电复位重新启动时，必须设置长于普通复位执行的期间，即大于等于 100μs （$\overline{\text{RES}}$=0 的期间）。

(4) 用户编程模式的擦除 / 编程步骤

通过 FTDAR 寄存器更改下载目标的内部 RAM 地址，可将擦除程序与编程程序下载至各自的内部 RAM 区域。

重复执行 RAM 仿真、擦除及编程的使用例子如图 20.13 所示。

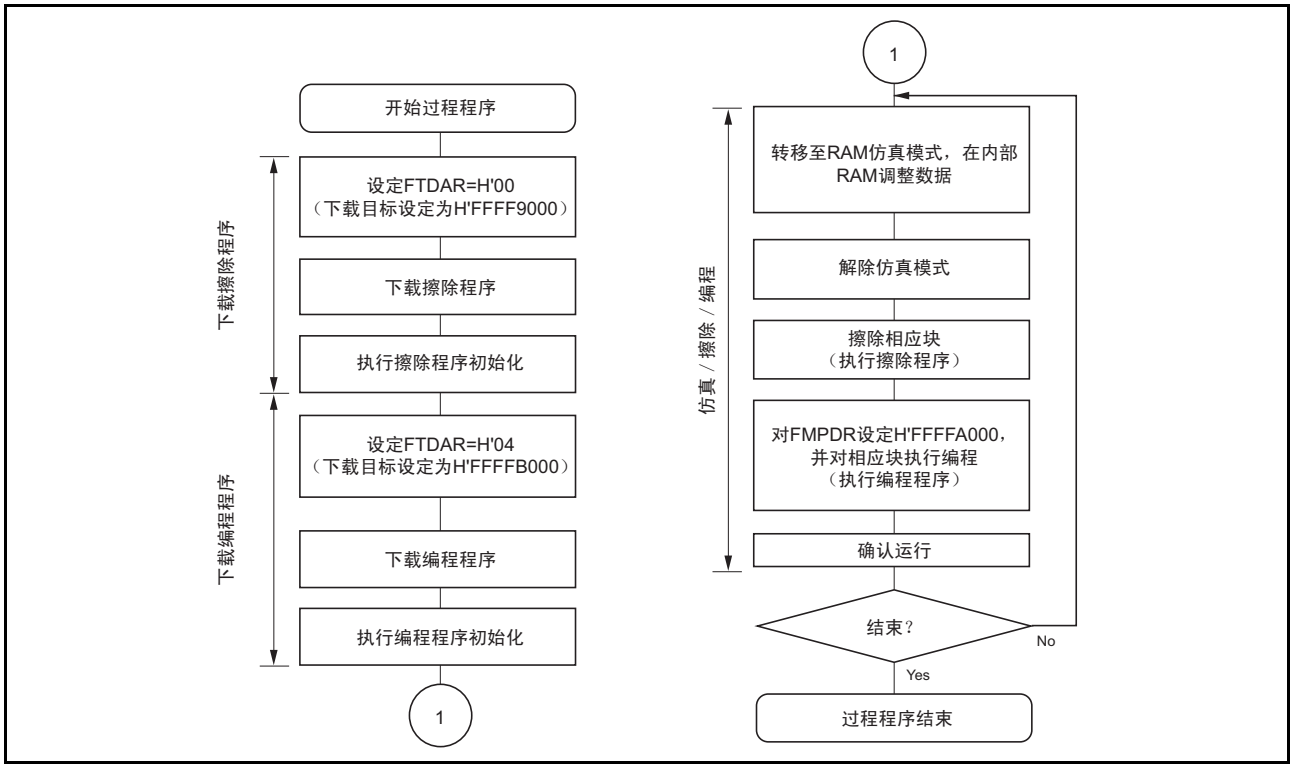


图 20.13 重复执行 RAM 仿真、擦除及编程的例子 （概要）

本例中，为了执行 RAM 仿真，需避开 H'FFFFA000 ～ H'FFFFAFFF，下载擦除 / 编程程序。

下载与初始化只在最初执行 1 次。

执行类似本例的步骤时，必须注意以下内容：

1. 注意内部RAM区域的重复破坏。
   除RAM仿真区域、擦除程序区域、编程程序区域之外，用户编写的过程程序、工作区域及堆栈区域等也存在于内部RAM，因此不得破坏这些区域。
2. 擦除程序及编程程序的初始化。
   必须在擦除程序及编程程序执行设定FPEFEQ参数及FUBRA参数的初始化。初始化的入口地址为擦除程序的下载起始地址+32字节 （在本例为H'FFFF9020）及编程程序的下载起始地址+32字节 （在本例为H'FFFFB020）。

20.5.3 用户引导模式

本 LSI 具有通过不同于用户编程模式、引导模式的模式引脚设定启动的用户引导模式，可实现与使用内部 SCI 的引导模式不同的用户任意引导模式。

在用户引导模式可编程 / 擦除的 MAT 仅限用户 MAT。必须在引导模式或编程器模式编程 / 擦除用户引导 MAT。

(1) 用户引导模式的启动

启动用户引导模式的模式引脚设定，详情参照表 20.1。

如果在用户引导模式开始复位，则执行闪存相关寄存器的检查程序。该程序使用的 RAM 容量约为从地址 H'FFFF9800 开始的 1.2KB 与用作堆栈的从地址 H'FFFFAFFC 开始的 4 字节。此期间，不可接受 NMI 及其他中断，也不可使用 AUD。以 40MHz 的内部频率运行时，该期间约为 100 ms。

之后，从用户引导 MAT 的复位向量开始执行起始地址处理。此时，执行 MAT 为用户引导 MAT，因此在闪存 MAT 选择寄存器 FMATS 设定 H'AA。

(2) 在用户引导模式编程用户 MAT

在用户引导模式编程用户 MAT 时，需追加以下步骤：通过 FMATS 寄存器从用户引导 MAT 选择状态转换至用户 MAT 选择状态；编程结束后，再从用户 MAT 选择状态返回至用户引导 MAT 选择状态。

在用户引导模式编程用户 MAT 的步骤如图 20.14 所示。

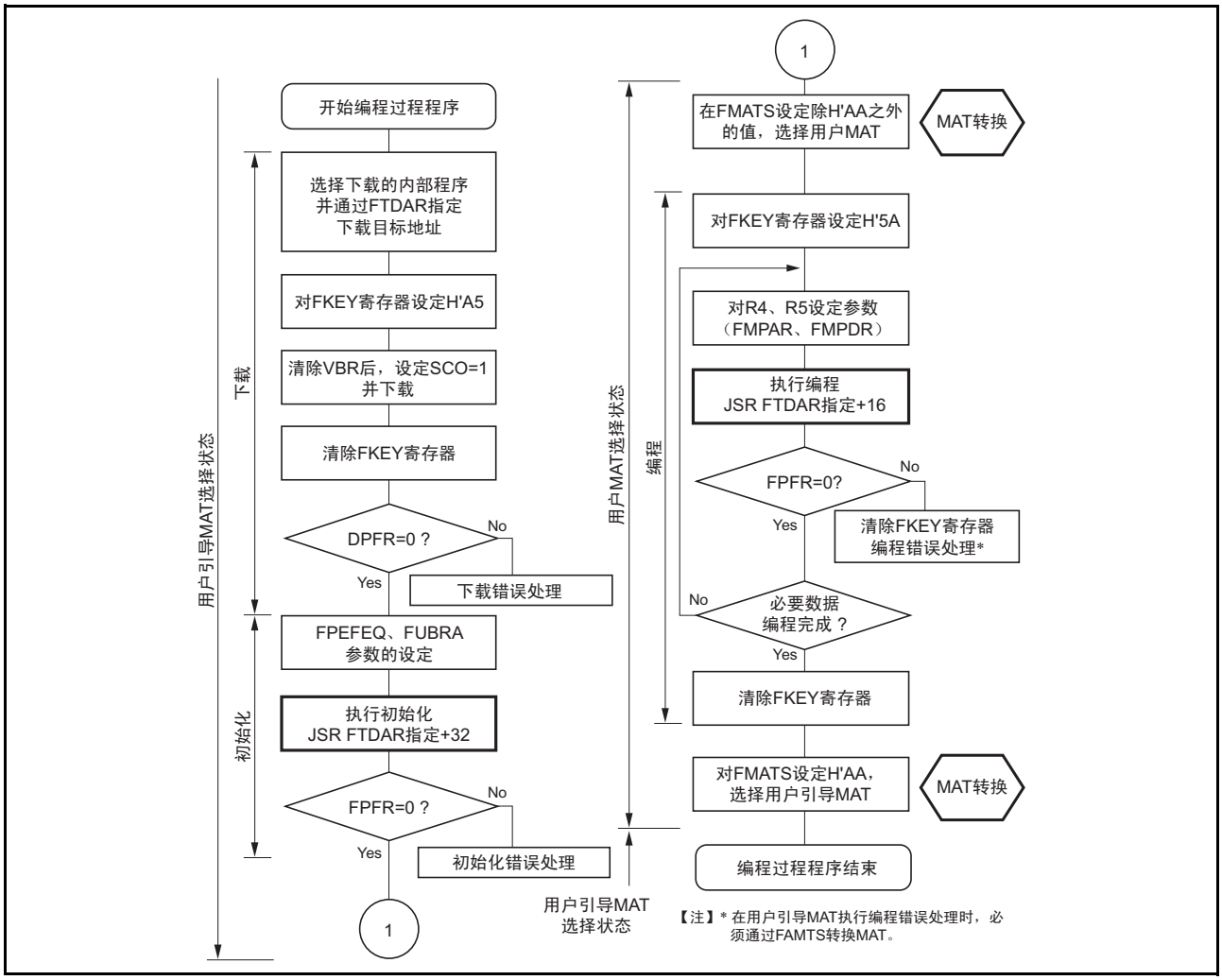


图 20.14 在用户引导模式编程用户 MAT 的步骤

如图 20.14 所示，用户编程模式与用户引导模式的编程步骤的不同在于是否转换 MAT。

在用户引导模式，可在闪存空间看到用户引导 MAT，用户 MAT 为隐藏在 “背面” 的状态。仅在编程用户 MAT 期间，转换用户 MAT 与用户引导 MAT。编程处理过程中，隐藏用户引导 MAT，且用户 MAT 为编程状态，因此需在闪存以外的区域执行过程程序。编程处理结束后，为了返回最初状态需重新转换 MAT。

通过对 FMATS 寄存器写入规定值可实现 MAT 转换，但在 MAT 转换完全结束前不可存取。产生中断时，存在从哪个 MAT 读取中断向量的不稳定状态。有关 MAT 转换，详情参阅 “20.8.1 用户 MAT 与用户引导 MAT 的转换”。

MAT 转换之外的编程步骤与用户编程模式的步骤相同。

可单步执行用户过程程序的区域（内部 RAM、用户 MAT 及外部空间等），详情参阅 “20.9.2 可保存过程程序或编程数据的区域”。

(3)　在用户引导模式擦除用户 MAT

在用户引导模式擦除用户 MAT 时，需追加以下步骤：通过 FMATS 寄存器从用户引导 MAT 选择状态转换至用户 MAT 选择状态；擦除结束后，再从用户 MAT 选择状态返回至用户引导 MAT 选择状态。

在用户引导模式擦除用户 MAT 的步骤如图 20.15 所示。

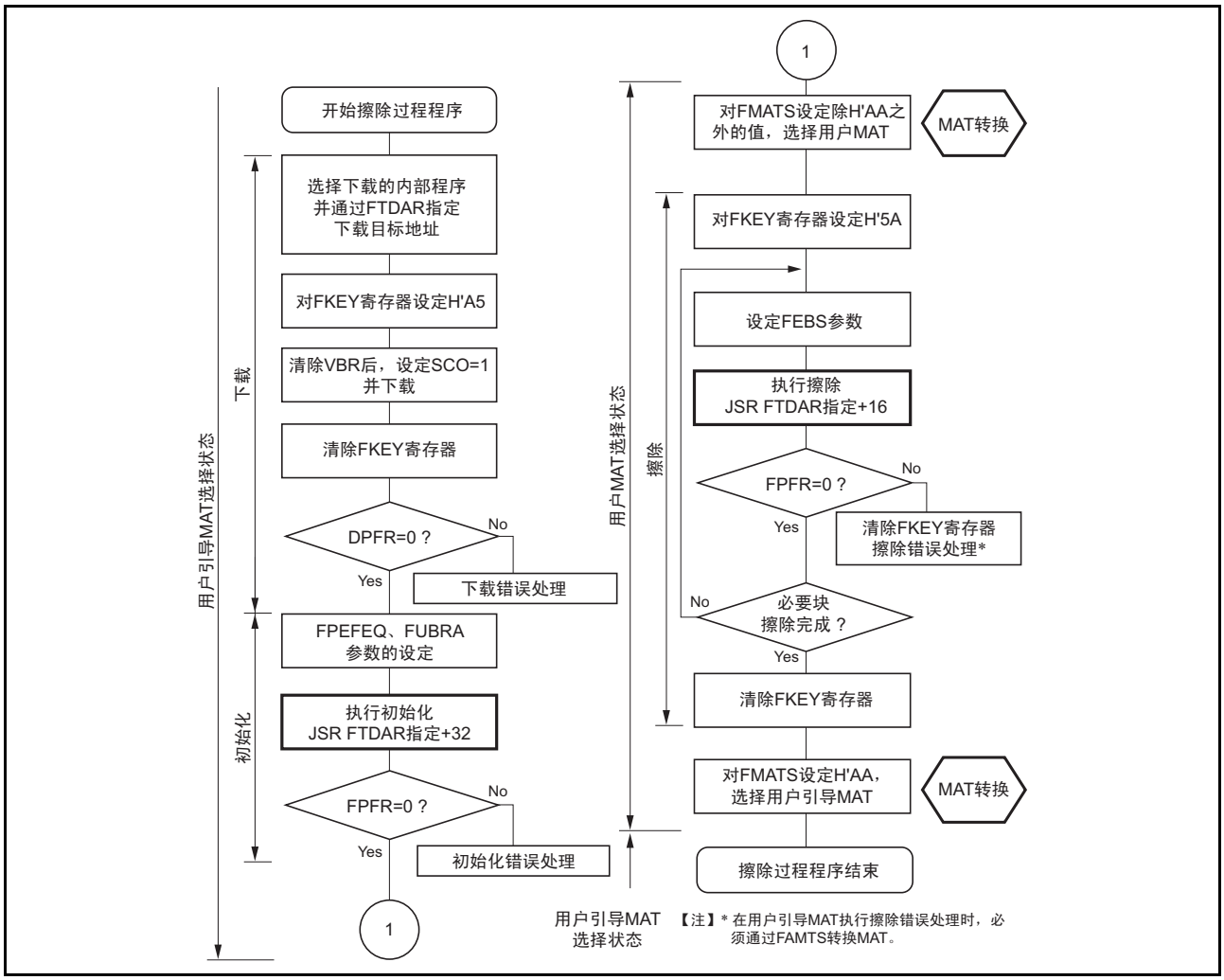


图 20.15　在用户引导模式擦除用户 MAT 的步骤

如图 20.14 所示，用户编程模式与用户引导模式擦除步骤的不同在于是否转换 MAT。

通过对 FMATS 寄存器写入规定值，可实现 MAT 转换，但在 MAT 转换完全结束前不可存取。产生中断时，存在从哪个 MAT 读取中断向量等不稳定状态。有关 MAT 转换，详情参照 “20.8.1 用户 MAT 与用户引导 MAT 的转换”。

除 MAT 转换之外的擦除步骤与用户编程模式的步骤相同。

可单步执行用户过程程序的区域（内部 RAM、用户 MAT 及外部空间等），详情参照 “20.9.2 可保存过程程序或编程数据的区域”。

20.6 保护

闪存的编程 / 擦除保护有硬件保护、软件保护及错误保护等 3 种。

20.6.1 硬件保护

硬件保护为强制禁止或中断对闪存执行编程 / 擦除的状态。可下载内部程序及执行初始化，即使启动编程 / 擦除程序，也不可编程 / 擦除用户 MAT，通过 FPFR 参数报告编程 / 擦除错误。

表 20.9 硬件保护

项目	说　明	保护功能有效	
		下载	编程与擦除
FWE 引脚保护	• 向 FWE 引脚输入低电平时，清除 FCCS 寄存器的 FWE 位，进入编程 / 擦除保护状态。	—	○
复位、待机保护	• 上电复位（包括由 WDT 产生的上电复位）及待机时，初始化编程 / 擦除接口寄存器，进入编程 / 擦除保护状态。 • 通过 $\overline{\text{RES}}$ 引脚复位，通电后到振荡稳定前，如果 $\overline{\text{RES}}$ 引脚不保持低电平，则不进入复位状态。正在运行的复位必须在 AC 特性规定的 $\overline{\text{RES}}$ 脉宽之间将 $\overline{\text{RES}}$ 引脚保持为低电平。无法保证编程 / 擦除过程中闪存的值。此时，必须在擦除后重新编程。	○	○

20.6.2 软件保护

软件保护有从内部编程 / 擦除程序下载的保护、通过键代码的保护及通过 RAM 仿真寄存器的保护。

表 20.10 软件保护

项目	说　明	保护功能有效	
		下载	编程与擦除
SCO 位保护	• 将 FCCS 寄存器的 SCO 位清 0，不可下载编程 / 擦除程序，因此进入编程 / 擦除保护状态。	○	○
FKEY 寄存器保护	• 如果不对 FKEY 寄存器写入键代码，则不可下载及编程 / 擦除，要下载及编程 / 擦除，需设定不同的键代码。	○	○
仿真保护	• 将 RAM 仿真寄存器（RAMER）的 RAMS 位置 1，进入编程 / 擦除保护状态。	○	○

20.6.3 错误保护

错误保护可检测编程 / 擦除闪存过程中，单片机失控或未按规定的编程 / 擦除步骤运行时产生的异常，并强制中断编程 / 擦除。通过中断编程 / 擦除，可防止因过度编程 / 擦除对闪存的损伤。

编程 / 擦除闪存过程中，如果单片机运行异常，则 FCCS 寄存器的 FLER 位置 1，转移至错误保护状态，并中断编程 / 擦除。

FLER 位的置位条件如下：

1. 编程/擦除过程中读取闪存的该存储体区域时 （包含读取向量及取指令）
2. 编程/擦除过程中执行SLEEP指令时 （包含软件待机）

仅在上电复位或硬件待机时解除错误保护 （清除 FLER 位）。

此时，必须在长于普通复位输入期间即 100μs 后解除复位。编程 / 擦除闪存过程中外加高电压，因此转移至错误保护状态时，可能无法完全去除外加电压。因此需延长复位期间及去除外加电压，以减少对闪存的损伤。

向错误保护状态的转移如图 20.16 所示。

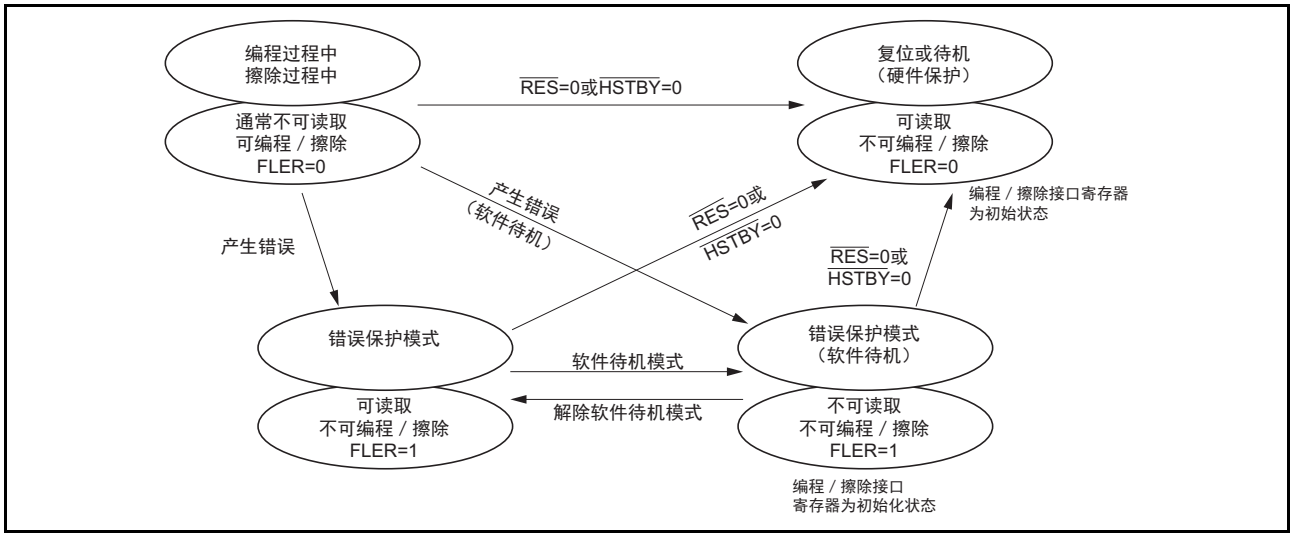


图 20.16 向错误保护状态的转移图

20.7　通过 RAM 仿真闪存

为了通过 RAM 实时仿真闪存改写的数据，可将 RAM 仿真寄存器 （RAMER）设定的闪存 （用户 MAT）区域与 RAM 的一部分重叠使用。设定 RAMER 后，可从用户 MAT 区域及与其重叠的 RAM 区域这 2 个区域存取。可仿真的模式为用户模式及用户编程模式。

仿真用户 MAT 的实时改写例子如图 20.17 所示。

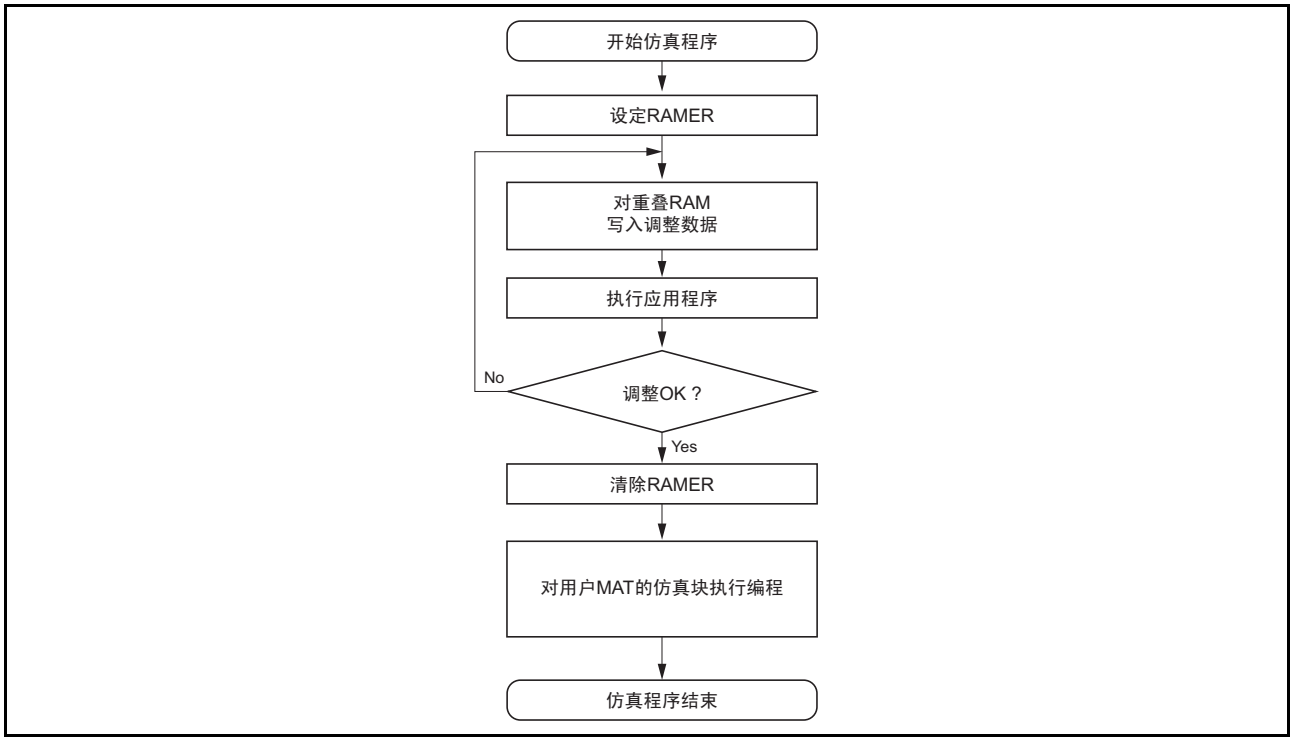


图 20.17　通过 RAM 仿真

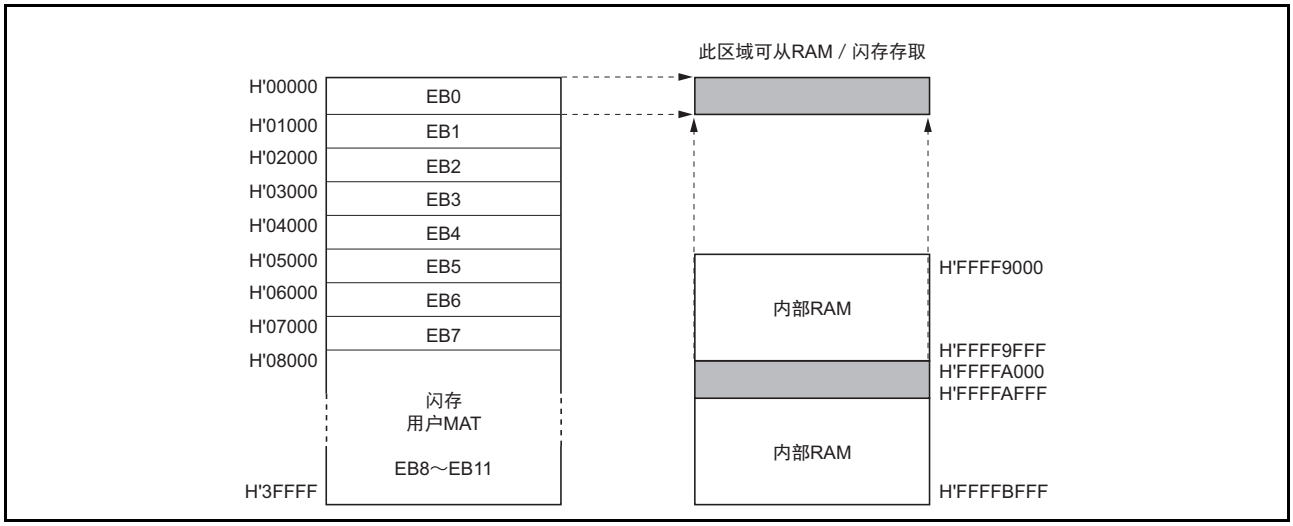


图 20.18　RAM 的重叠运行例子 （闪存 256KB 版时）

使闪存的块区域 EB0 重叠的例子如图 20.18 所示。

可仿真的闪存区域为通过 RAMER 寄存器的 RAM2 ～ RAM0 位从用户 MAT 的 EB0 ～ EB7 的 8 个区域选择的 1 个区域。

1. 为了使需实时改写的区域EB0与RAM的一部分重叠，必须将RAMER的RAMS位设定为1，并将RAM2～RAM0位均设定为0。
2. 使用重叠的RAM实时改写。

编程 / 擦除用户 MAT 时，需执行包含下载内部程序的一系列过程程序。此时，为了避免重叠的 RAM 区域与下载的内部程序区域重合，必须使用 FTDAR 寄存器并设定下载区域。

对用户 MAT 的 EB0 区域写入仿真完成后的数据例子如图 20.19 所示。

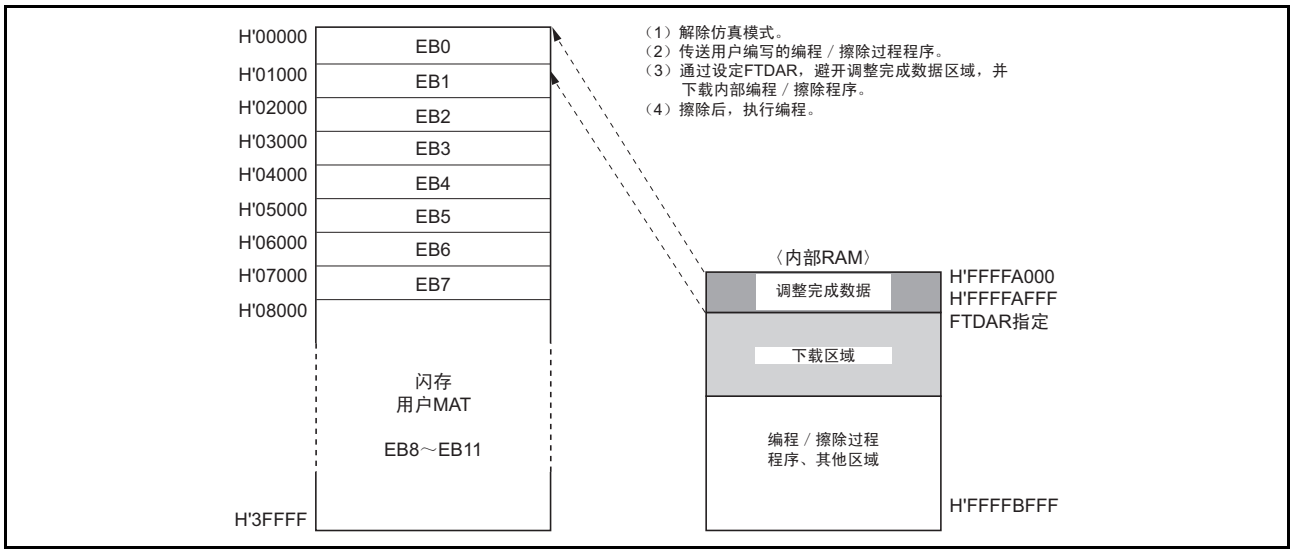


图 20.19　调整完成数据的编程例子 （闪存 256KB 版时）

1. 确定改写数据后，清除RAMS位，解除RAM的重叠，并解除仿真模式及仿真保护。
2. 将用户编写的编程/擦除过程程序传送至RAM。
3. 启动RAM的编程/擦除过程程序，将单片机内部的编程/擦除程序下载至RAM。
   此时，为了避免调整完成数据区域与下载区域重合，必须通过设定FTDAR寄存器指定下载起始址。
4. 未擦除用户MAT的EB0区域时，必须擦除后再执行编程。对编程处理的参数FMPAR、 FMPDR指定已调整的数据，并执行编程处理。

【注】 RAMS 位置 1 时，与 RAM2 ～ RAM0 的值无关，闪存的所有块为编程 / 擦除保护状态 （仿真保护）。实际编程 / 擦除时，必须清除 RAMS 位。
即使选择用户引导MAT，也可执行RAM仿真，但仅在引导模式或编程器模式擦除/编程用户引导MAT。

20.8 使用时的注意事项

20.8.1 用户 MAT 与用户引导 MAT 的转换

可转换用户 MAT 与用户引导 MAT，分配至相同的地址 0，因此需执行以下步骤：

（不可在转换为用户引导 MAT 状态执行编程 / 擦除，必须在引导模式或编程器模式改写用户引导 MAT。）

1. 必须在内部RAM由FMATS寄存器转换MAT。
   在SH单片机需预取执行指令，例如，在用户MAT执行程序过程中转换MAT时，需预取用户MAT的操作码或预取转换后的用户引导MAT的指令，因此运行不稳定。
2. 为了保证存取转换后的MAT，必须在内部RAM改写FMATS寄存器后，立即在相同的内部RAM执行4条NOP指令 （因为在转换过程中不存取闪存）。
3. 转换过程中产生中断时，无法保证存取哪个存储器MAT。
   MAT转换前，必须屏蔽可屏蔽的中断，并在MAT转换过程中，设置不产生NMI中断的系统。
4. 必须注意：MAT转换结束后，也转换各种中断的向量表区域。
   MAT转换前后，执行相同中断处理或不可禁止产生中断时，预先将中断处理程序传送至内部RAM，并通过设定VBR寄存器，将中断向量表设定至内部RAM。此时，要注意VBR寄存器的更改与中断之间的竞争。
5. 用户MAT与用户引导MAT的存储器容量不同。不得存取大于等于12KB空间的用户引导MAT，存取大于等于12KB的空间时，读取值不定。

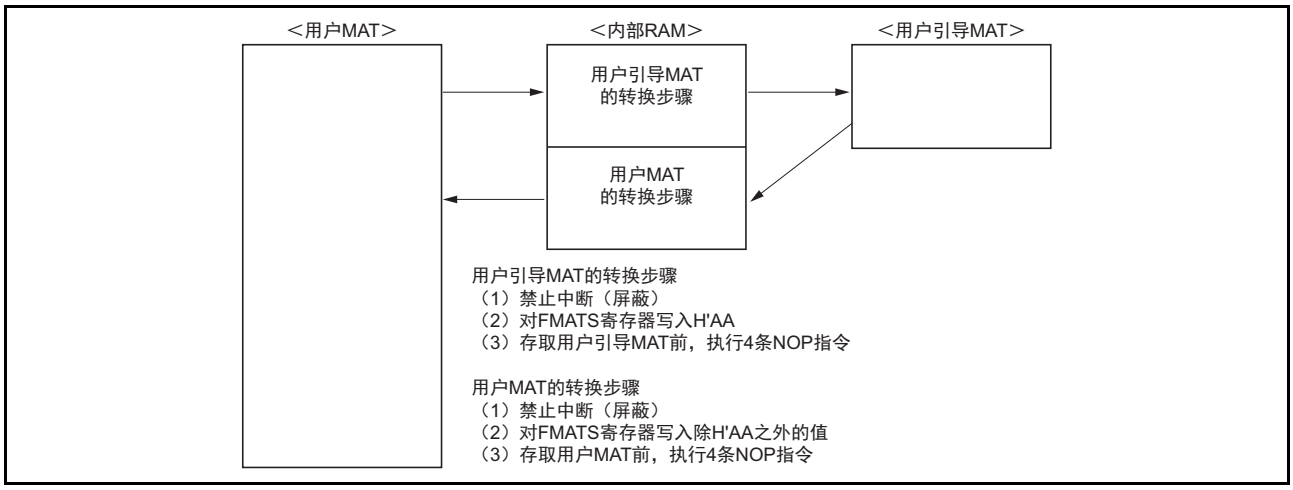


图 20.20 用户 MAT/ 用户引导 MAT 的转换

20.8.2 执行编程 / 擦除过程中的中断

(1) 内部程序的下载

（1.1） VBR 的更改

下载内部程序前，必须将 VBR 寄存器设置为 H'84000000。否则使用 VBR 时，如果将 VBR 设定为 H'84000000，则中断向量表为用户 MAT （FMATS≠H'AA 时）或用户引导 MAT （FMATS=H'AA 时）。

VBR 的更改与中断竞争时，可能产生参照 VBR 更改前后哪个向量表的问题。

可能与中断产生竞争时，必须在用户 MAT 或用户引导 MAT 的起始部分准备 VBR=H'00000000 （初始值）时参照的向量表。

（1.2） SCO 下载请求与中断请求

FCCS 寄存器的 SCO 位置 1 时，下载编程 / 擦除程序，会产生伴随 MAT 转换的特殊中断。以下说明 SCO 下载请求与中断请求竞争时的运行：

1. SCO下载请求与中断请求的竞争

执行设定 FCCS 寄存器的 SCO 位为 1 的指令与接受中断的竞争时序如图 20.21 所示。

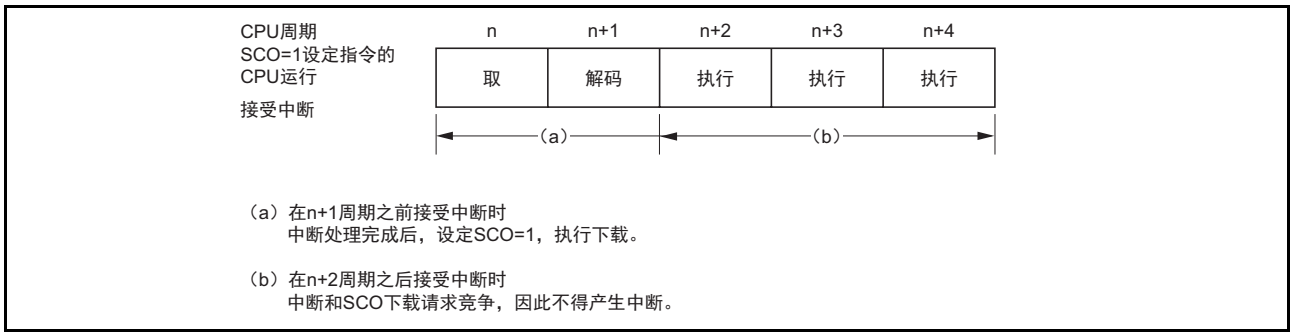


图 20.21 SCO 下载请求与中断请求的竞争时序

2. 下载过程中产生的中断请求

SCO下载过程中，不得产生中断。

(2) 编程 / 擦除处理过程中的中断

禁止在编程 / 擦除下载的内部程序时产生中断及通过除 CPU 之外的总线主控器获取总线权。

20.8.3 其他注意事项

(1) 内部程序的下载时

包含初始化程序的编程程序或擦除程序的代码长度分别小于等于 3KB。因此， CPU 时钟频率为 20MHz 时，各自的下载时间最长约为 10ms。

(2) 用户分支处理的间隔

执行用户分支处理的间隔因编程 / 擦除各异，也因处理相位不同而各异。 CPU 时钟频率为 64 ～ 80MHz 时的最大启动间隔如表 20.11 所示。

表 20.11 用户分支处理的启动间隔

	最大间隔
编程处理	约 2ms
擦除处理	约 15ms

CPU 时钟以 64 ～ 80MHz 运行时到最初用户分支处理为止的最长时间如表 20.12 所示。

表 20.12 用户分支处理的时间

	最长
编程处理	约 2ms
擦除处理	约 15ms

(3) 通过 DTC 写入闪存相关寄存器

在内部 RAM 执行指令时，从 DTC 也可写入请求下载的 FCCS 寄存器的 SCO 位或执行 MAT 转换的 FMATS 寄存器。必须注意：不慎写入此寄存器，会执行下载、破坏 RAM、产生 MAT 转换且导致失控等。

(4) 中断忽略状态

在以下模式或期间，即使产生中断也可忽略，不执行也不保持中断源：

- 引导模式运行过程中
- 编程器模式运行过程中

(5) 用户 MAT 在 384KB 或 256KB 产品编程时的注意事项

必须注意：在 384KB 的产品，对用户 MAT 执行大于等于 384KB 的编程时，无法保证第 384KB 以后的编程内容。

必须注意：在 256KB 的产品，对用户 MAT 执行大于等于 256KB 的编程时，无法保证第 256KB 以后的编程内容。

(6) 与以往的 F-ZTAT SH 单片机编程 / 擦除程序的兼容性

不支持由 SCO 传送请求执行的内部程序下载方式。在本 LSI，以往的 F-ZTAT SH 单片机使用的闪存编程 / 擦除程序不运行。

在本 LSI 编程 / 擦除闪存时，必须下载内部程序。

(7) 监控因 WDT 引起的失控等

与以往的 F-ZTAT SH 单片机不同，通过下载的内部程序执行编程 / 擦除的过程中，不对因 WDT 引起的失控等采取措施。

根据需要，必须对考虑了编程 / 擦除时间的 WDT 采取措施 （使用用户分支程序及定期的定时器中断等）。

(8) 堆栈地址

编程 / 擦除过程中，堆栈的起始地址必须在内部 RAM。

(9) 使用 RAM 仿真功能时的存取禁止区域

使用 RAM 仿真功能时，禁止存取表 20.13 所示的区域。

表 20.13 使用 RAM 仿真功能时的禁止存取的区域

	选择 EB0 时	选择 EB1 时	选择 EB2 时	选择 EB3 时	选择 EB4 时	选择 EB5 时	选择 EB6 时	选择 EB7 时
存取禁止区域	H'00008000 ～ H'00008FFF	H'00009000 ～ H'00009FFF	H'0000A000 ～ H'0000AFFF	H'0000B000 ～ H'0000BFFF	H'0000C000 ～ H'0000CFFF	H'0000D000 ～ H'0000DFFF	H'0000E000 ～ H'0000EFFF	H'0000F000 ～ H'0000FFFF
	H'00010000 ～ H'00010FFF	H'00011000 ～ H'00011FFF	H'00012000 ～ H'00012FFF	H'00013000 ～ H'00013FFF	H'00014000 ～ H'00014FFF	H'00015000 ～ H'00015FFF	H'00016000 ～ H'00016FFF	H'00017000 ～ H'00017FFF
	H'00018000 ～ H'00018FFF	H'00019000 ～ H'00019FFF	H'0001A000 ～ H'0001AFFF	H'0001B000 ～ H'0001BFFF	H'0001C000 ～ H'0001CFFF	H'0001D000 ～ H'0001DFFF	H'0001E000 ～ H'0001EFFF	H'0001F000 ～ H'0001FFFF
	H'00020000 ～ H'00020FFF	H'00021000 ～ H'00021FFF	H'00022000 ～ H'00022FFF	H'00023000 ～ H'00023FFF	H'00024000 ～ H'00024FFF	H'00025000 ～ H'00025FFF	H'00026000 ～ H'00026FFF	H'00027000 ～ H'00027FFF
	H'00028000 ～ H'00028FFF	H'00029000 ～ H'00029FFF	H'0002A000 ～ H'0002AFFF	H'0002B000 ～ H'0002BFFF	H'0002C000 ～ H'0002CFFF	H'0002D000 ～ H'0002DFFF	H'0002E000 ～ H'0002EFFF	H'0002F000 ～ H'0002FFFF
	H'00030000 ～ H'00030FFF	H'00031000 ～ H'00031FFF	H'00032000 ～ H'00032FFF	H'00033000 ～ H'00033FFF	H'00034000 ～ H'00034FFF	H'00035000 ～ H'00035FFF	H'00036000 ～ H'00036FFF	H'00037000 ～ H'00037FFF
	H'00038000 ～ H'00038FFF	H'00039000 ～ H'00039FFF	H'0003A000 ～ H'0003AFFF	H'0003B000 ～ H'0003BFFF	H'0003C000 ～ H'0003CFFF	H'0003D000 ～ H'0003DFFF	H'0003E000 ～ H'0003EFFF	H'0003F000 ～ H'0003FFFF
	H'00040000 ～ H'00040FFF	H'00041000 ～ H'00041FFF	H'00042000 ～ H'00042FFF	H'00043000 ～ H'00043FFF	H'00044000 ～ H'00044FFF	H'00045000 ～ H'00045FFF	H'00046000 ～ H'00046FFF	H'00047000 ～ H'00047FFF
	H'00048000 ～ H'00048FFF	H'00049000 ～ H'00049FFF	H'0004A000 ～ H'0004AFFF	H'0004B000 ～ H'0004BFFF	H'0004C000 ～ H'0004CFFF	H'0004D000 ～ H'0004DFFF	H'0004E000 ～ H'0004EFFF	H'0004F000 ～ H'0004FFFF
	H'00050000 ～ H'00050FFF	H'00051000 ～ H'00051FFF	H'00052000 ～ H'00052FFF	H'00053000 ～ H'00053FFF	H'00054000 ～ H'00054FFF	H'00055000 ～ H'00055FFF	H'00056000 ～ H'00056FFF	H'00057000 ～ H'00057FFF
	H'00058000 ～ H'00058FFF	H'00059000 ～ H'00059FFF	H'0005A000 ～ H'0005AFFF	H'0005B000 ～ H'0005BFFF	H'0005C000 ～ H'0005CFFF	H'0005D000 ～ H'0005DFFF	H'0005E000 ～ H'0005EFFF	H'0005F000 ～ H'0005FFFF
	H'00060000 ～ H'00060FFF	H'00061000 ～ H'00061FFF	H'00062000 ～ H'00062FFF	H'00063000 ～ H'00063FFF	H'00064000 ～ H'00064FFF	H'00065000 ～ H'00065FFF	H'00066000 ～ H'00066FFF	H'00067000 ～ H'00067FFF
	H'00068000 ～ H'00068FFF	H'00069000 ～ H'00069FFF	H'0006A000 ～ H'0006AFFF	H'0006B000 ～ H'0006BFFF	H'0006C000 ～ H'0006CFFF	H'0006D000 ～ H'0006DFFF	H'0006E000 ～ H'0006EFFF	H'0006F000 ～ H'0006FFFF
	H'00070000 ～ H'00070FFF	H'00071000 ～ H'00071FFF	H'00072000 ～ H'00072FFF	H'00073000 ～ H'00073FFF	H'00074000 ～ H'00074FFF	H'00075000 ～ H'00075FFF	H'00076000 ～ H'00076FFF	H'00077000 ～ H'00077FFF
	H'00078000 ～ H'00078FFF	H'00079000 ～ H'00079FFF	H'0007A000 ～ H'0007AFFF	H'0007B000 ～ H'0007BFFF	H'0007C000 ～ H'0007CFFF	H'0007D000 ～ H'0007DFFF	H'0007E000 ～ H'0007EFFF	H'0007F000 ～ H'0007FFFF

20.9 附录

20.9.1 引导模式的标准串行通信接口规格

使用 LSI 内部的 SCI 执行主机与在引导模式启动的引导程序的发送 / 接收。主机与引导程序的串行通信接口规格如下：

- 状态

引导程序有 3 种状态。

1. 位速率匹配状态
   使主机与发送/接收位速率匹配的状态。如果在引导模式启动，则启动引导程序，并进入位速率匹配状态，接收来自主机的命令，并使位速率匹配。匹配结束时，转移至查询选择状态。
2. 查询选择状态
   响应主机查询命令的状态。在此状态选择器件、时钟模式及位速率。选择结束时，通过编程/擦除状态转移命令转移至编程/擦除状态。转移至编程/擦除状态前，引导程序将擦除程序传送至RAM，并擦除用户MAT与用户引导MAT。
3. 编程/擦除状态
   执行编程/擦除的状态。根据主机的命令，将编程/擦除程序传送至RAM，并执行编程/擦除。通过命令执行和校验及空白检查。

引导程序的处理流程如图 20.22 所示。

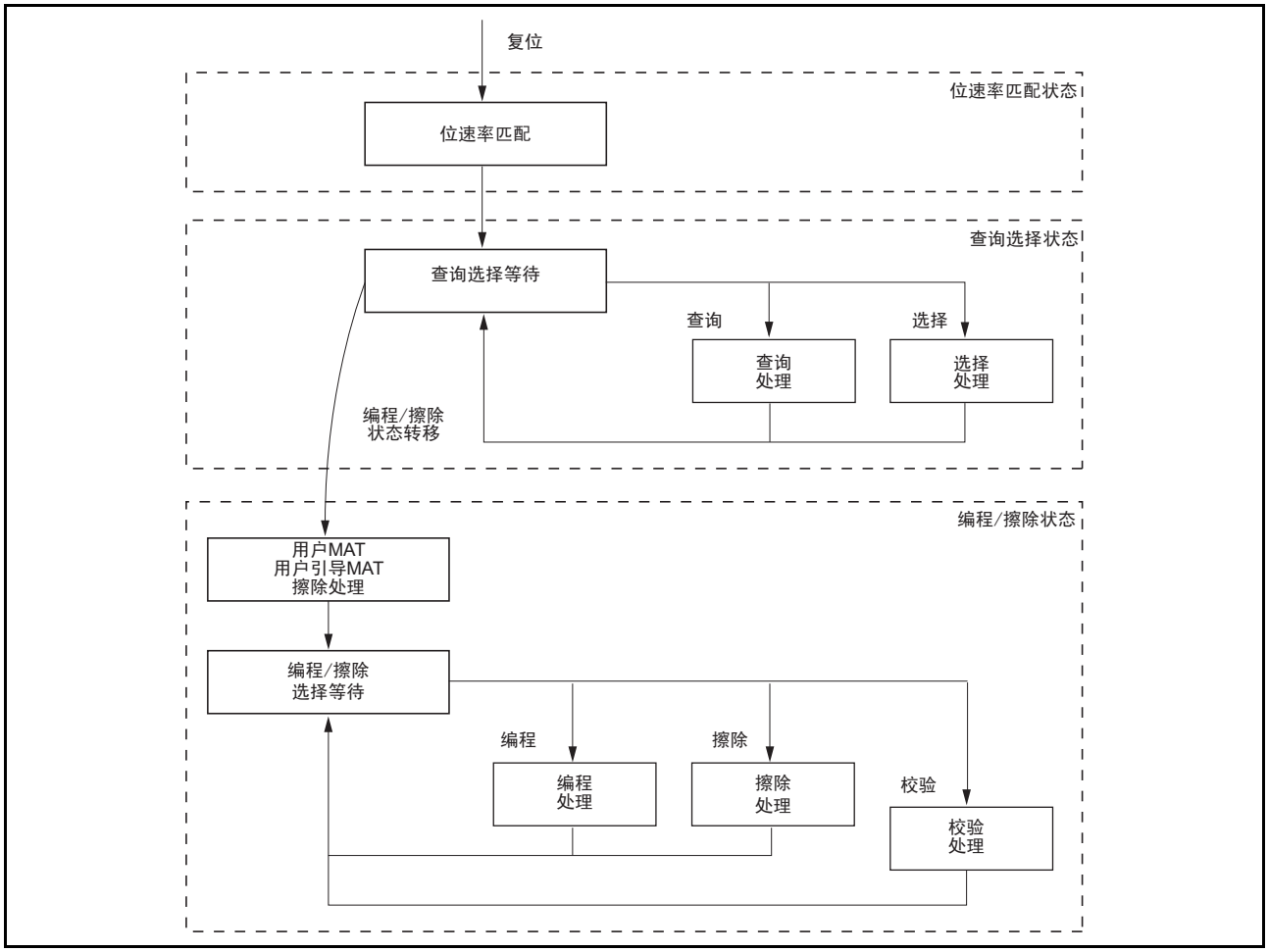


图 20.22 引导程序的处理流程

• 位速率匹配状态

位速率匹配可测量从主机发送的 H'00 低电平区间，并计算位速率。可通过新位速率选择命令更改该位速率。位速率匹配结束时，引导程序转移至查询选择状态。位速率匹配的顺序如图 20.23 所示。

图 20.23 位速率匹配的顺序

• 通信协议

位速率匹配结束后的主机与引导程序之间的串行通信协议如下：

1. 1个字符命令或1个字符应答
   命令或应答仅为1个字符，因此具有查询及正常结束的ACK。
2. n个字符命令或n个字符应答
   命令及应答需n字节的数据，因此具有对应选择命令与查询的应答。
   另外规定编程数据的长度，所以在此省略。
3. 错误应答
   对命令的错误应答，有错误应答与错误代码2个字节。
4. 128字节的编程
   为无长度的命令，可通过编程长度查询的应答得知数据长度。
5. 存储器读取的应答
   为包含4字节长度信息的应答。

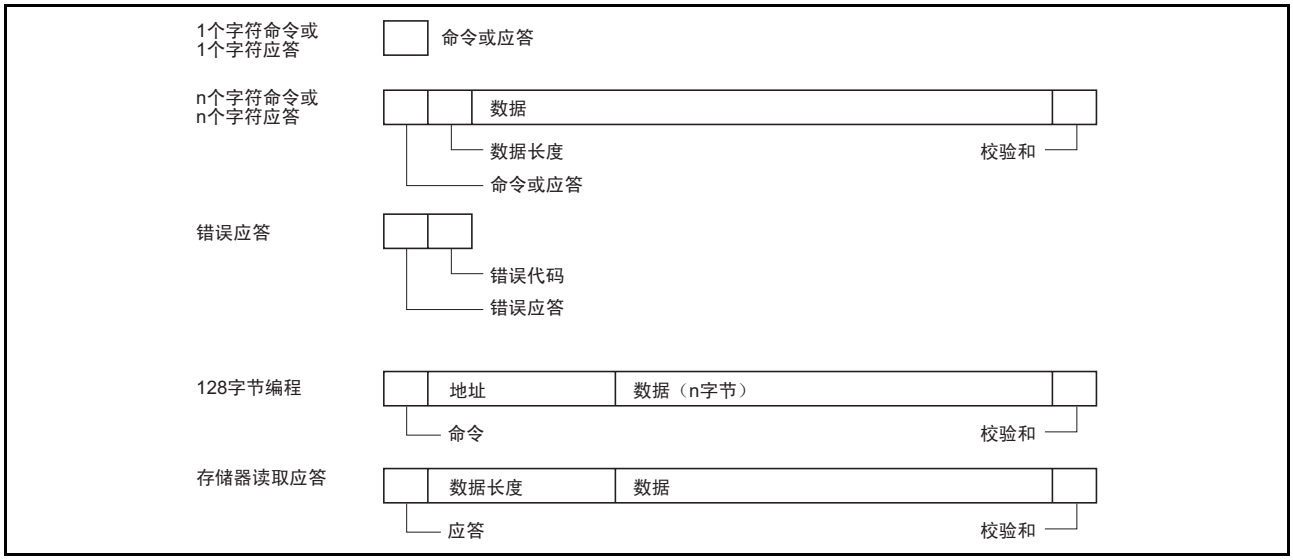


图 20.24 通信协议格式

- 命令 （1字节）：查询、选择、编程、擦除及检查等命令
- 应答 （1字节）：对查询的响应
- 长度 （1字节或2字节）：除命令/应答、长度、校验和之外的发送/接收数据长度
- 数据 （n字节）：命令、应答的详细数据
- 校验和 （1字节）：命令到校验和相加，并设定使低位1字节为H'00
- 错误应答 （1字节）：对命令的错误应答
- 错误代码 （1字节）：产生的错误种类
- 地址 （4字节）：编程地址
- 数据 （n字节）：编程数据。可从编程长度查询命令的应答得知n
- 数据长度 （4字节）：存储器读取的应答为4字节长度

- **查询选择状态**

在查询选择状态，引导程序对主机的查询命令响应闪存 ROM 的信息，对选择命令选择器件、时钟模式及位速率。

查询选择命令一览表如**表 20.14** 所示。

表 20.14　查询选择命令一览表

命令	命令名称	功　　能
H'20	支持器件查询	查询器件代码与产品名称
H'10	器件选择	选择器件代码
H'21	时钟模式查询	查询时钟模式数及各自的值
H'11	时钟模式选择	通知选择的时钟模式
H'22	倍频比查询	查询倍频比或分频比的种类数与各自的个数及其值
H'23	工作频率查询	查询主时钟与外围时钟的最小值与最大值
H'24	用户引导 MAT 信息查询	查询用户引导 MAT 的个数与各自的起始地址及结束地址
H'25	用户 MAT 信息查询	查询用户 MAT 的个数与各自的起始地址及结束地址
H'26	擦除块信息查询	查询块数与各自的起始地址及结束地址
H'27	编程长度查询	查询编程时的数据长度
H'3F	新位速率选择	选择新位速率
H'40	编程 / 擦除状态转移	擦除用户 MAT 及用户引导 MAT，并转移至编程 / 擦除状态
H'4F	引导程序状态查询	查询引导的处理状态

必须按照器件选择 （H'10）、时钟模式选择 （H'11）、新位速率选择 （H'3F）的顺序，从主机发送选择命令，而且这些命令都必须要。相同的选择命令发送 2 次以上时，后发送的选择命令有效。

除引导程序状态查询 （H'4F）之外，这些命令在接受编程 / 擦除状态转移 （H'40）前均有效。主机可从上述命令选择所需命令并查询。引导程序状态查询 （H'4F）在接受编程 / 擦除状态转移 （H'40）后也有效。

(1)　支持器件查询

对于支持器件查询，引导程序响应可支持器件的器件代码与产品名称。

命令

H'20

- 命令 [H'20] （1字节）：支持器件查询

应答

H'30	长度	器件数
字符数	器件代码	产品名称
…		
SUM		

- 应答 [H'30] （1字节）：对支持器件查询的响应
- 长度 （1字节）：除命令、长度及校验和之外的发送/接收数据长度。此处为器件数、字符数、器件代码及产品名称的总长度
- 器件数 （1字节）：单片机内的引导程序支持的品种数
- 字符数 （1字节）：器件代码与引导程序产品名称的字符数
- 器件代码 （4字节）：所支持产品名称的ASCII代码
- 产品名称 （n字节）：引导程序型号 （ASCII代码）
- SUM （1字节）：校验和
  命令到SUM相加，使结果为H'00

(2) 器件选择

对于器件选择，引导程序将支持器件设定为指定的支持器件。对于以后的查询，响应所选择器件的信息。

命令

H'10	长度	器件代码	SUM

- 命令[H'10] （1字节）：器件选择
- 长度 （1字节）：器件代码的字符数 （固定值为2）
- 器件代码 （4字节）：由支持器件查询响应的器件代码 （ASCII代码）
- SUM （1字节）：校验和

应答

H'06

- 应答[H'06] （1字节）：对器件选择的响应
  指定的器件代码与支持器件匹配时，返回ACK

错误

应答

H'90	ERROR

- 错误应答[H'90] （1字节）：对器件选择的错误响应
- ERROR：（1字节）：错误代码
  H'11：和校验错误
  H'21：器件代码不匹配错误

(3) 时钟模式查询

对于时钟模式查询，引导程序响应可选择的时钟模式。

命令

H'21

- 命令[H'21] （1字节）：时钟模式查询

应答

H'31	长度	模式	…	SUM

- 应答[H'31] （1字节）：对时钟模式查询的响应
- 长度 （1字节）：模式数、模式的总长度
- 模式 （1字节）：可选择的时钟模式 （例：H'01 时钟模式1）
- SUM （1字节）：校验和

(4) 时钟模式选择

对于时钟模式选择，引导程序将时钟模式设定为指定的时钟模式。对于以后的查询，响应所选择时钟模式的信息。

必须在发送器件选择命令后，发送时钟模式选择命令。

命令

H'11	长度	模式	SUM

- 命令[H'11] （1字节）：时钟模式选择
- 长度 （1字节）：模式的字符数 （固定值为1）
- 模式 （1字节）：时钟模式查询响应的时钟模式
- SUM （1字节）：校验和

应答

H'06

- 应答[H'06] （1字节）：对时钟模式选择的响应

指定的时钟模式与可选择的时钟模式匹配时，返回 ACK

错误

应答

H'91	ERROR

- 错误应答[H'91] （1字节）：对时钟模式选择的错误响应
- ERROR：（1字节）：错误代码

  H'11：和校验错误

  H'22：时钟模式不匹配错误

(5) 倍频比查询

对于倍频比查询，引导程序响应可选择的倍频比或分频比。

命令

H'22

- 命令[H'22] （1字节）：倍频比查询

应答

H'32	长度	频率种类					
倍频比数	倍频比	…					
…							
SUM							

- 应答[H'32] （1字节）：对倍频比查询的响应
- 长度 （1字节）：频率的种类数、倍频比数、倍频比的总长度
- 频率的种类数 （1字节）：器件可选择的倍频比种类数

  （如果为主工作频率与外围模块工作频率两种，则为H'02）
- 倍频比数 （1字节）：各工作频率可选择的倍频比数

  主模块及外围模块可选择的倍频比数
- 倍频比 （1字节）

  倍频比：倍频的数值 （例 4倍频：H'04）

  分频比：分频的数值、负数 （例 2分频：H'FE[-2]）

  倍频比重复的次数仅为倍频比数，倍频比数与倍频比的组合重复的次数仅为频率种类数。
- SUM （1字节）：校验和

(6) 工作频率查询

对于工作频率查询，引导程序响应工作频率数及其最小值、最大值。

命令

H'23

- 命令[H'23] （1字节）：工作频率查询

应答

H'33	长度	频率种类
工作频率最小值		工作频率最大值
…		
SUM		

- 应答[H'33] （1字节）：对工作频率查询的响应
- 长度 （1字节）：频率的种类数、工作频率最小值及最大值的总长度
- 频率的种类数 （1字节）：器件所需的工作频率种类数
  例如，主工作频率与外围模块工作频率时，为2
- 工作频率最小值 （2字节）：倍频或分频时钟的最小值
  工作频率最小值、最大值为将频率 （MHz）的小数点后2位的值扩大100倍的值 （例如：20.00MHz时，扩大100倍为2000，设定为H'07D0）
- 工作频率最大值 （2字节）：倍频或分频时钟的最大值
  工作频率最小值及最大值的数据重复的次数为频率种类数
- SUM （1字节）：校验和

(7) 用户引导 MAT 信息查询

对于用户引导 MAT 信息查询，引导程序响应用户引导 MAT 的区域数与地址。

命令

H'24

- 命令[H'24] （1字节）：用户引导MAT信息查询

应答

H'34	长度	区域数	
区域起始地址			区域结束地址
…			
SUM			

- 应答[H'34] （1字节）：对用户引导MAT信息查询的响应
- 长度 （1字节）：区域数、区域起始地址及区域结束地址的总长度
- 区域数 （1字节）：用户引导MAT的区域数
  用户引导MAT的区域连续时，为H'01
- 区域起始地址 （4字节）：区域的起始地址
- 区域结束地址 （4字节）：区域的结束地址
  区域起始地址及区域结束地址的数据重复的次数为区域数
- SUM （1字节）：校验和

(8) 用户 MAT 信息查询

对于用户 MAT 信息查询，引导程序响应用户 MAT 的区域数与地址。

命令

H'25

- 命令[H'25] （1字节）：用户MAT信息查询

应答

H'35	长度	区域数

区域起始地址	区域结束地址
…	

SUM

- 应答[H'35] （1字节）：对用户MAT信息查询的响应
- 长度 （1字节）：区域数、区域起始地址及区域结束地址的总长度
- 区域数 （1字节）：用户MAT的区域数
  用户MAT的区域连续时，为H'01
- 区域起始地址 （4字节）：区域的起始地址
- 区域结束地址 （4字节）：区域的结束地址
  区域起始地址及区域结束地址的数据重复的次数为区域数
- SUM （1字节）：校验和

(9) 擦除块信息查询

对于擦除块信息查询，引导程序响应用户 MAT 擦除块的块数及其地址。

命令

H'26

- 命令[H'26] （1字节）：擦除块信息查询

应答

H'36	长度	块数

块起始地址	块结束地址
…	

SUM

- 应答[H'36] （1字节）：对擦除块信息查询的响应
- 长度 （2字节）：块数、块起始地址及块结束地址的总长度
- 块数 （1字节）：闪存擦除块数
- 块起始地址 （4字节）：块的起始地址
- 块结束地址 （4字节）：块的结束地址
  块起始地址及块结束地址的数据重复的次数为块数
- SUM （1字节）：校验和

(10) 编程长度查询

对于编程长度查询，引导程序响应编程数据的编程单位。

命令

H'27

- 命令[H'27] （1字节）：编程长度查询

应答

H'37	长度	编程长度	SUM

- 应答[H'37] （1字节）：对编程长度查询的响应
- 长度 （1字节）：编程单位长度的字符数 （固定值为2）
- 编程长度 （2字节）：编程单位的长度
  用该长度接受编程数据
- SUM （1字节）：校验和

(11) 新位速率选择

对于新位速率选择，引导程序选择更改为指定的位速率，并通过新位速率响应来自主机的确认。
必须在发送时钟模式选择命令后，发送新位速率选择命令。

命令

H'3F	长度	位速率	输入频率
倍频比数	倍频比 1	倍频比 2	
SUM			

- 命令[H'3F] （1字节）：新位速率选择
- 长度 （1字节）：位速率、输入频率、倍频比数及倍频比的总长度
- 位速率 （2字节）：新位速率
  为1/100的值 （例如：19200bps时为192，设置为H'00C0）
- 输入频率 （2字节）：输入引导程序的时钟频率
  为频率 （MHz）的小数点后2位的值 （例如：28.882MHz时，小数点后2位扩大100倍为2888，设置为H'0B48）
- 倍频比数 （1字节）：器件可选择的倍频比数
  通常为主工作频率与外围模块工作频率，为2
- 倍频比1 （1字节）：主工作频率的倍频比或分频比
  倍频比：倍频的数值 （例 4倍频：H'04）
  分频比：分频的数值，负数 （例 2分频：H'FE[-2]）
- 倍频比2 （1字节）：外围工作频率的倍频比或分频比
  倍频比：倍频的数值 （例 4倍频：H'04）
  分频比：分频的数值，负数 （例 2分频：H'FE[-2]）
- SUM （1字节）：校验和

应答

H'06

- 应答[H'06] （1字节）：对选择新位速率的响应
  选择指定的位速率时，作为ACK发送

错误

应答

H'BF	ERROR

- 错误应答[H'BF]（1字节）：对选择新位速率的错误响应
- ERROR：（1字节）：错误代码

H'11：和校验错误

H'24：不可选择位速率的错误
不可选择指定的位速率

H'25：输入频率错误
输入频率不在最小值与最大值的范围内

H'26：倍频比错误
倍频比不匹配

H'27：工作频率错误
工作频率不在最小值与最大值的范围内

已接收数据的检查方法如下：

1. 输入频率
   检查接收的输入频率值是否在输入频率（对于已选择器件的时钟模式）的最小值与最大值的范围内。如果不在范围内，则为输入频率错误。
2. 倍频比
   检查接收的倍频比或分频比的值与倍频比或分频比（对于已选择器件的时钟模式）是否匹配。如果不匹配，则为倍频比错误。
3. 工作频率
   由已接收的输入频率与倍频比或分频比计算工作频率。输入频率为提供给LSI的频率；工作频率为LSI实际运行的频率。公式如下：
   工作频率＝输入频率×倍频比
   或工作频率＝输入频率÷分频比
   检查计算出的工作频率是否在工作频率（对于已选择器件的时钟模式）的最小值与最大值的范围内如果不在范围内，则为工作频率错误。
4. 位速率
   从外围工作频率（Pϕ）与位速率（B）得出串行模式寄存器（SCSMR）的时钟选择（CKS）值（n）与位速率寄存器（SCBRR）的值（N），并计算误差，检查误差是否小于4%，如果误差大于等于4%，则为位速率选择错误。误差计算如下：

$$\text{误差（\%）} = \left\{\left[\frac{P\phi\times10^{6}}{(N+1)\times B\times64\times2^{2n-1}}\right]-1\right\}\times100$$

可选择新位速率时，在响应 ACK 后，给寄存器选择新位速率的值。主机以新位速率发送 ACK，引导程序以新位速率响应。

确认

H'06

- 确认[H'06]（1字节）：确认新位速率

应答

H'06

- 应答[H'06]（1字节）：对确认新位速率的响应

选择新位速率的顺序如图 20.25 所示。

图 20.25 选择新位速率的顺序

(12) 编程 / 擦除状态转移

对于编程 / 擦除状态转移，引导程序传送擦除程序，并按照用户 MAT、用户引导 MAT 的顺序擦除数据。如果擦除结束，则响应 ACK，并转移至编程 / 擦除状态。

发送编程选择命令与编程数据前，主机必须通过器件选择命令、时钟模式选择命令及新位速率选择命令选择 LSI 的器件、时钟模式及新位速率，并将编程 / 擦除状态转移命令发送至引导程序。

命令

H'40

- 命令 [H'40] （1 字节）：编程/擦除状态转移

应答

H'06

- 应答 [H'06] （1 字节）：对编程/擦除状态转移的响应。传送擦除程序后，用户引导 MAT 及用户 MAT 正常擦除数据时，返回 ACK

错误

应答

H'C0	H'51

- 错误应答 [H'C0] （1 字节）：对编程/擦除状态转移的错误响应
- 错误代码 [H'51] （1 字节）：擦除错误
  产生错误且不可擦除

- **命令错误**

未定义命令、命令顺序不正确或不可接受命令时，均为命令错误。例如：器件选择前的时钟模式选择命令、编程 / 擦除状态转移命令后的查询命令为命令错误。

错误

应答	H'80	H'xx

- 错误应答[H'80] （1字节）：命令错误
- 命令[H'xx] （1字节）：已接收的命令

- **命令的顺序**

在查询选择状态的命令顺序例子如下所示：

1. 通过支持器件查询 （H'20）命令，查询支持器件。
2. 根据响应器件信息选择器件，并执行器件选择 （H'10）。
3. 通过时钟模式查询 （H'21）命令查询时钟模式。
4. 根据响应时钟模式选择时钟模式，并执行时钟模式选择 （H'11）。
5. 器件选择及时钟模式选择结束后，由倍频比查询 （H'22）及工作频率查询 （H'23）命令查询选择新位速率所需的信息。
6. 根据倍频比及工作频率的信息，选择新位速率 （H'3F）。
7. 器件选择及时钟模式选择结束后，通过用户引导MAT信息查询 （H'24）、用户MAT信息查询 （H'25）、擦除块信息查询 （H'26）及编程长度查询 （H'27）命令查询用户引导MAT及用户MAT的编程/擦除信息。
8. 查询与选择新位速率结束后，执行编程/擦除状态转移 （H'40），转移至编程/擦除状态。

- **编程/擦除状态**

在编程 / 擦除状态，引导程序通过编程选择命令选择编程方法，通过 128 字节编程命令执行数据编程，并通过擦除选择命令与块擦除命令执行块擦除。编程 / 擦除命令一览表如表 20.15 所示。

表 20.15 编程 / 擦除命令一览表

命令	命令名称	功 能
H'42	用户引导 MAT 编程选择	选择用户引导 MAT 编程程序
H'43	用户 MAT 编程选择	选择用户 MAT 编程程序
H'50	128 字节编程	128 字节编程
H'48	擦除选择	选择擦除程序
H'58	块擦除	擦除块数据
H'52	存储器读取	读取存储器
H'4A	用户引导 MAT 的和校验	对用户引导 MAT 执行和校验
H'4B	用户 MAT 的和校验	对用户 MAT 执行和校验
H'4C	用户引导 MAT 的空白检查	对用户引导 MAT 执行空白检查
H'4D	用户 MAT 的空白检查	对用户 MAT 执行空白检查
H'4F	引导程序状态查询	查询引导程序的处理状态

- 编程

通过编程选择命令与 128 字节编程命令执行编程。

首先，主机发送编程选择命令，并选择编程方式与编程 MAT。根据编程区域与编程方式，编程选择命令有以下两种：

1. 用户引导MAT编程选择
2. 用户MAT编程选择

其次，发送 128 字节编程命令。选择命令后续的 128 字节编程命令分别为选择命令所指定编程方式的编程数据。编程超过 128 字节的数据时，重复执行 128 字节编程命令。要结束编程时，必须从主机发送地址为 H'FFFFFFFF 的 128 字节编程命令。编程结束时，进入编程 / 擦除选择等待状态。

从编程选择命令开始继续执行其他方式及其他 MAT 的编程。

编程选择命令与 128 字节编程命令的顺序如图 20.26 所示。

图 20.26 编程顺序

(1) 用户引导 MAT 编程选择

对于用户引导 MAT 编程选择，引导程序传送编程程序，并由传送的编程程序对用户引导 MAT 执行编程。

命令

H'42

- 命令[H'42]（1字节）：用户引导MAT编程选择

应答

H'06

- 应答[H'06]（1字节）：对用户引导MAT编程选择的响应。传送编程程序后，返回ACK。

错误

应答

H'C2	ERROR

- 错误应答[H'C2]（1字节）：对用户引导MAT编程选择的错误响应
- ERROR：（1字节）：错误代码

  H'54：选择处理错误（产生传送错误，且处理未完成）

(2) 用户 MAT 编程选择

对于用户 MAT 编程选择，引导程序传送编程程序，并由已传送的编程程序对用户 MAT 执行编程。

命令

H'43

- 命令[H'43] （1字节）：用户MAT编程选择

应答

H'06

- 应答[H'06] （1字节）：对用户MAT编程选择的响应。传送编程程序后，返回ACK

错误

应答

H'C3	ERROR

- 错误应答[H'C3] （1字节）：对用户MAT编程选择的错误响应
- ERROR：（1字节）：错误代码
  H'54：选择处理错误 （产生传送错误，且处理未完成）

(3) 128 字节编程

对于 128 字节编程，引导程序通过编程选择传送的编程程序，对用户引导 MAT 或用户 MAT 执行编程。

命令

H'50	编程地址						
数据	…						
…							
SUM							

- 命令[H'50] （1字节）：128字节编程
- 编程地址 （4字节）：编程起始地址
  必须指定128字节边界的地址
  例） H'00、 H'01、 H'00、 H'00：H'00010000
- 编程数据 （n字节）：编程数据
  编程数据的长度为[编程长度查询]响应的长度
- SUM （1字节）：校验和

应答

H'06

- 应答[H'06] （1字节）：对128字节编程的响应
  编程结束时，返回ACK

错误

应答

H'D0	ERROR

- 错误应答[H'D0] （1字节）：对128字节编程的错误响应
- ERROR：（1字节）：错误代码
  H'11：和校验错误
  H'2A：地址错误 （地址不在指定的MAT范围内）
  H'53：编程错误 （产生编程错误，且无法编程）

按照数据编程长度，必须指定边界地址。例如：数据编程长度为 128 字节时，必须将地址的低位字节设置为 H'00 或 H'80。

主机必须将 128 字节中无编程数据的部分填入 H'FF，并发送。

编程处理结束时，必须发送地址 H'FFFFFFFF 的 128 字节编程命令。对于地址 H'FFFFFFFF 的 128 字节编程命令，引导程序判断数据已结束后，进入编程 / 擦除选择命令等待状态。

命令

H'50	编程地址	SUM

- 命令[H'50] （1字节）：128字节编程
- 编程地址 （4字节）：结束代码 （H'FF、 H'FF、 H'FF、 H'FF）
- SUM （1字节）：校验和

应答

H'06

- 应答[H'06] （1字节）：对128字节编程的响应
  编程处理结束时，返回ACK

错误

应答

H'D0	ERROR

- 错误应答[H'D0] （1字节）：对128字节编程的错误响应
- ERROR：（1字节）：错误代码
  H'11：和校验错误
  H'53：编程错误

- 擦除

由擦除选择命令与块擦除命令执行擦除。

首先，由擦除选择命令选择擦除，其次，由块擦除命令擦除指定的块。有多个擦除块时，重复块擦除命令。擦除结束时，必须从主机发送块编号 H'FF 的块擦除命令，并进入编程 / 擦除选择等待状态。

擦除选择命令与块擦除命令的顺序如图 20.27 所示。

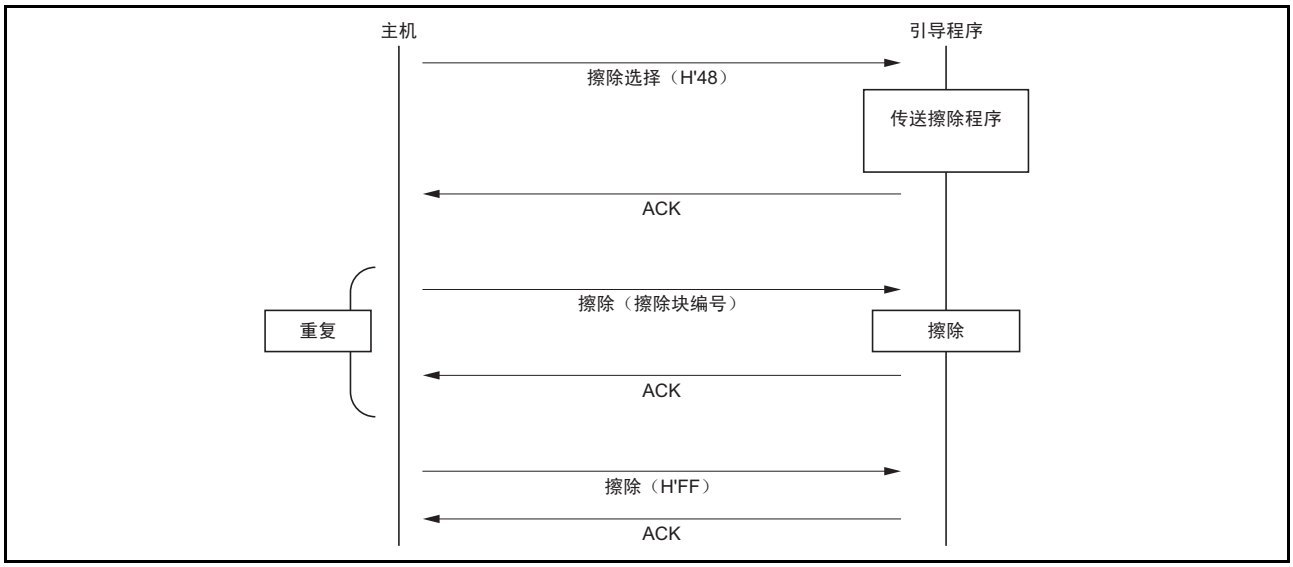


图 20.27 擦除顺序

(1) 擦除选择

对于擦除选择，引导程序传送擦除程序。由已传送的擦除程序擦除用户 MAT 数据。

命令

H'48

- 命令[H'48] （1字节）：擦除选择

应答

H'06

- 应答[H'06] （1字节）：对擦除选择的响应
  传送擦除程序后，返回ACK

错误

应答

H'C8	ERROR

- 错误应答[H'C8] （1字节）：对擦除选择的错误响应
- ERROR：（1字节）：错误代码
  H'54：选择处理错误 （产生传送错误，且处理未完成）

(2) 块擦除

对于擦除，引导程序擦除所指定用户 MAT 的块。

命令

H'58	长度	块编号	SUM

- 命令[H'58] （1字节）：擦除
- 长度 （1字节）：擦除块编号的字符数 （固定值为1）
- 块编号 （1字节）：擦除数据的擦除块编号
- SUM （1字节）：校验和

应答

H'06

- 应答[H'06] （1字节）：对擦除的响应
  擦除结束时，返回ACK

错误

应答

H'D8	ERROR

- 错误应答[H'D8] （1字节）：对擦除的错误响应
- ERROR：（1字节）：错误代码
  H'11：和校验错误
  H'29：块编号错误
  块编号不正确
  H'51：擦除错误
  擦除过程中产生错误

块编号为 H'FF，引导程序结束擦除处理后，进入选择命令等待状态。

命令

H'58	长度	块编号	SUM

- 命令[H'58] （1字节）：擦除
- 长度 （1字节）：擦除块编号的字符数 （固定值为1）
- 块编号 （1字节）：H'FF （擦除处理的结束代码）
- SUM （1字节）：校验和

应答

H'06

- 应答[H'06] （1字节）：对擦除结束的响应ACK

由 H'FF 指定块编号后，重新擦除时，从擦除选择开始执行。

- **存储器读取**

对于存储器读取，引导程序响应所指定地址的数据。

命令

H'52	长度	区域	读取起始地址
读取长度			SUM

- 命令[H'52] （1字节）：存储器读取
- 长度 （1字节）：区域、读取地址及读取长度的总长度 （固定值为9）
- 区域 （1字节）

  H'00：用户引导MAT

  H'01：用户MAT

  区域的指定不正确时，为地址错误
- 读取起始地址 （4字节）：读取起始地址
- 读取长度 （4字节）：读取数据的长度
- SUM （1字节）：校验和

应答

H'52	读取长度						
数据	…						
SUM							

- 应答[H'52] （1字节）：对存储器读取的响应
- 读取长度 （4字节）：读取数据的长度
- 数据 （n字节）：从读取地址读取长度为读取长度的数据
- SUM （1字节）：校验和

错误

应答

H'D2	ERROR

- 错误应答[H'D2] （1字节）：对存储器读取的错误响应
- ERROR：（1字节）：错误代码

  H'11：和校验错误

  H'2A：地址错误

  读取起始地址不在MAT的范围内

  H'2B：长度错误

  读取长度超出MAT的范围，或由读取起始地址与读取长度计算的读取结束地址不在MAT的范围内，或读取长度为0

- 用户引导MAT的和校验

对于用户引导 MAT 的和校验，引导程序将用户引导 MAT 的数据相加并响应其结果。

命令

H'4A

- 命令[H'4A] （1字节）：用户引导MAT的和校验

应答

H'5A	长度	MAT 的校验和	SUM

- 应答[H'5A] （1字节）：对用户引导MAT和校验的响应
- 长度 （1字节）：MAT校验和的字符数 （固定值为4）
- MAT的校验和 （4字节）：以字节为单位，将用户引导MAT的和校验值相加
- SUM （1字节）：校验和 （发送数据的）

- 用户MAT的和校验

对于用户 MAT 的和校验，引导程序将用户 MAT 的数据相加并响应其结果。

命令

H'4B

- 命令[H'4B] （1字节）：用户MAT的和校验

应答

H'5B	长度	MAT 的校验和	SUM

- 应答[H'5B] （1字节）：对用户MAT和校验的响应
- 长度 （1字节）：MAT校验和的字符数 （固定值为4）
- MAT的校验和 （4字节）：用户MAT的和校验值
  以字节为单位相加
- SUM （1字节）：校验和 （发送数据的）

- 用户引导MAT的空白检查

对于用户引导 MAT 的空白检查，引导程序检查用户引导 MAT 全部为空白并响应其结果。

命令

H'4C

- 命令[H'4C] （1字节）：用户引导MAT的空白检查

应答

H'06

- 应答[H'06] （1字节）：对用户引导MAT空白检查的响应。区域全部为空白 （H'FF）时，返回ACK

错误

应答

H'CC	H'52

- 错误应答[H'CC] （1字节）：对用户引导MAT空白检查的错误响应
- 错误代码[H'52] （1字节）：未擦除错误

- 用户MAT的空白检查

对于用户 MAT 的空白检查，引导程序检查用户 MAT 全部为空白并响应其结果。

命令

H'4D

- 命令[H'4D] （1字节）：用户MAT的空白检查

应答

H'06

- 应答[H'06] （1字节）：对用户MAT空白检查的响应。区域全部为空白 （H'FF）时，返回ACK

错误

应答

H'CD	H'52

- 错误应答[H'CD] （1字节）：对用户MAT空白检查的错误响应
- 错误代码[H'52] （1字节）：未擦除错误

- 引导程序状态查询

对于引导程序状态查询，引导程序响应当前的状态与错误状态。该查询在查询选择状态及编程 / 擦除状态均有效。

命令

H'4F

- 命令[H'4F] （1字节）：引导程序状态查询

应答

H'5F	长度	STATUS	ERROR	SUM

- 应答[H'5F] （1字节）：对引导程序状态查询的响应
- 长度 （1字节）：数据的字符数 （固定值为2）
- STATUS （1字节）：标准引导程序的状态
  详情参照表20.16。
- ERROR （1字节）：错误状态
  ERROR＝0时，正常
  ERROR≠0时，异常
  详情参照表20.17。
- SUM （1字节）：校验和

表 20.16 状态代码

代码	内 容
H'11	器件选择等待
H'12	时钟模式选择等待
H'13	位速率选择等待
H'1F	编程 / 擦除状态转移等待 （位速率选择完成）
H'31	用户 MAT、用户引导 MAT 擦除过程中
H'3F	编程 / 擦除选择等待 （擦除完成）
H'4F	编程数据接收等待 （编程完成）
H'5F	擦除块指定等待 （擦除完成）

表 20.17 错误代码

代码	内 容
H'00	无错误
H'11	和校验错误
H'21	器件代码不匹配错误
H'22	时钟模式不匹配错误
H'24	不可选择位速率的错误
H'25	输入频率错误
H'26	倍频比错误
H'27	工作频率错误
H'29	块编号错误
H'2A	地址错误
H'2B	数据长度错误 （长度错误）
H'51	擦除错误
H'52	未擦除错误
H'53	编程错误
H'54	选择处理错误
H'80	命令错误
H'FF	位速率匹配确认错误

20.9.2 可保存过程程序或编程数据的区域

本章正文已举例说明在内部 RAM 准备可保存编程 / 擦除过程程序及编程数据的区域，如果遵守以下条件，则可在其他区域 （外部空间区域等）执行。

1. 从FTDAR寄存器指定的内部RAM的地址下载并执行内部编程/擦除程序，因此不可使用该区域。
2. 内部编程/擦除程序使用大于等于128字节的堆栈区，因此必须确保该区域。
3. 将SCO位置1，执行下载请求的处理中，会产生MAT转换，因此必须在内部RAM执行。
4. 开始编程/擦除前 （判断下载结果前），可存取闪存。在单芯片模式等不可存取外部空间的模式，必须在此之前，将所需的过程程序、中断向量、中断处理程序及用户分支处理程序等传送至内部RAM。
5. 编程/擦除过程中不可存取闪存，因此通过下载至内部RAM的程序执行。在闪存以外的内部RAM或外部总线空间也必须有启动该程序的过程程序、编程/擦除过程中用户分支目标的用户程序、中断向量表及中断处理程序的执行区域。
6. 从编程/擦除结束后到清除FKEY寄存器的期间，禁止存取闪存。
   编程/擦除结束后，立即更改LSI模式并开始复位时，必须设定大于等于100 μs的复位期间 （$\overline{\text{RES}}$=0的期间）。
   另外，禁止转移至编程/擦除过程中的复位状态及硬件待机状态，但错误复位时，必须在长于普通复位期间即100μs后，解除复位。
7. 在用户引导模式编程/擦除用户MAT时，需通过FMATS转换MAT。必须在内部RAM执行MAT转换。（详情参阅 “20.8.1 用户MAT与用户引导MAT的转换”）
   转换MAT时，必须注意当前选择了哪个MAT。
8. 编程处理的参数FMPDR所表示的编程数据保存区域在闪存时，判断为错误，因此必须先传送至内部RAM，并设定FMPDR表示的地址在闪存空间以外。

基于这些条件，根据各运行模式 / 处理内容的组合，编程数据保存区域及可执行区域如下表所示：

表 20.18　可执行的 MAT

处理	启动模式	
	用户编程模式	用户引导模式 *
编程	表 20.19（1）	表 20.19（3）
擦除	表 20.19（2）	表 20.19（4）

【注】 *　可编程 / 擦除用户 MAT。

表 20.19　(1)　在用户编程模式的编程处理可使用的区域

项目	可保存 / 执行的区域			选择的 MAT	
	内部 RAM	用户 MAT	外部空间	用户 MAT	嵌入式程序保存 MAT
编程数据的保存区域	○	× *	○	—	—
选择下载的内部程序	○	○	○	○	
对键寄存器写入 H'A5	○	○	○	○	
设定 FCCS 的 SCO=1（下载）	○	×	×		○
清除键寄存器	○	○	○	○	
判断下载结果	○	○	○	○	
处理下载错误	○	○	○	○	
设定初始化参数	○	○	○	○	
执行初始化	○	×	×	○	
判断初始化结果	○	○	○	○	
处理初始化错误	○	○	○	○	
中断处理程序	○	×	○	○	
对键寄存器写入 H'5A	○	○	○	○	
设定编程参数	○	×	○	○	
执行编程	○	×	×	○	
判断编程结果	○	×	○	○	
处理编程错误	○	×	○	○	
清除键寄存器	○	×	○	○	

编程顺序 ↓

【注】 *　预先传送至内部 RAM 即可保存 / 执行。

表 20.19 (2) 在用户编程模式的擦除处理可使用的区域

项目	可保存 / 执行的区域			选择的 MAT	
	内部 RAM	用户 MAT	外部空间	用户 MAT	嵌入式程序保存 MAT
选择下载的内部程序	○	○	○	○	
对键寄存器写入 H'A5	○	○	○	○	
设定 FCCS 的 SCO=1 （下载）	○	×	×		○
清除键寄存器	○	○	○	○	
判断下载结果	○	○	○	○	
处理下载错误	○	○	○	○	
设定初始化参数	○	○	○	○	
执行初始化	○	×	×	○	
判断初始化结果	○	○	○	○	
处理初始化错误	○	○	○	○	
中断处理程序	○	×	○	○	
对键寄存器写入 H'5A	○	○	○	○	
设定擦除参数	○	×	○	○	
执行擦除	○	×	×	○	
判断擦除结果	○	×	○	○	
处理擦除错误	○	×	○	○	
清除键寄存器	○	×	○	○	

擦除顺序

表 20.19　(3)　在用户引导模式的编程处理可使用的区域

项目	可保存 / 执行的区域			选择的 MAT		
	内部 RAM	用户引导 MAT	外部空间	用户 MAT	用户引导 MAT	嵌入式程序保存 MAT
编程数据的保存区域	○	× *1	○	—	—	—
选择下载的内部程序	○	○	○		○	
对键寄存器写入 H'A5	○	○	○		○	
设定 FCCS 的 SCO=1 （下载）	○	×	×			○
清除键寄存器	○	○	○		○	
判断下载结果	○	○	○		○	
处理下载错误	○	○	○		○	
设定初始化参数	○	○	○		○	
执行初始化	○	×	×		○	
判断初始化结果	○	○	○		○	
处理初始化错误	○	○	○		○	
中断处理程序	○	×	○		○	
由 FMATS 转换 MAT	○	×	×	○		
对键寄存器写入 H'5A	○	×	○	○		
设定编程参数	○	×	○	○		
执行编程	○	×	×	○		
判断编程结果	○	×	○	○		
编程错误处理	○	× *2	○	○		
清除键寄存器	○	×	○	○		
由 FMATS 转换 MAT	○	×	×		○	

编程顺序 ↓

【注】　*1　预先传送至内部 RAM 即可保存 / 执行。

*2　在内部 RAM 转换 FMATS 后即可保存 / 执行。

表 20.19　(4)　在用户引导模式的擦除处理可使用的区域

项目	可保存 / 执行的区域			选择的 MAT		
	内部 RAM	用户引导 MAT	外部空间	用户 MAT	用户引导 MAT	嵌入式程序保存 MAT
选择下载的内部程序	○	○	○		○	
对键寄存器写入 H'A5	○	○	○		○	
设定 FCCS 的 SCO=1 （下载）	○	×	×			○
清除键寄存器	○	○	○		○	
判断下载结果	○	○	○		○	
处理下载错误	○	○	○		○	
设定初始化参数	○	○	○		○	
执行初始化	○	×	×		○	
判断初始化结果	○	○	○		○	
处理初始化错误	○	○	○		○	
中断处理程序	○	×	○		○	
由 FMATS 转换 MAT	○	×	×		○	
对键寄存器写入 H'5A	○	×	○	○		
设定擦除参数	○	×	○	○		
执行擦除	○	×	×	○		
判断擦除结果	○	×	○	○		
处理擦除错误	○	× *	○	○		
清除键寄存器	○	×	○	○		
由 FMATS 转换 MAT	○	×	×		○	

擦除顺序 ↓

【注】　*　在内部 RAM 转换 FMATS 后即可保存 / 执行。

20.10　编程器模式

在编程器模式，与单一闪存相同，可通过插座适配器由 PROM 编程器执行编程 / 擦除。必须使用支持瑞萨 512KB/384KB/256KB 闪存内部单片机器件类型 （F-ZTATxxxx）的 PROM 编程器。

第 21 章 RAM

本 LSI 内置高速静态 RAM。内部 RAM 通过 32 位宽的数据总线 （L 总线）连接 CPU，通过 32 位宽的数据总线 （I 总线）连接数据传送控制器 （DTC），能以 8、 16 或 32 位宽存取内部 RAM。内部 RAM 分配至图 21.1、图 21.2 所示地址，根据不同的地址，分为页 0、页 1。 RAM 可由 CPU （通过 L 总线）、 DTC （通过 I 总线）存取。从不同总线同时对相同页请求存取时，优先顺序为 I 总线 （DTC）＞ L 总线 （CPU）。此竞争会降低 RAM 存取性能，因此推荐采取软件措施，尽量避免产生竞争。例如，各总线存取不同页时，不产生竞争。只要不产生页竞争， L 总线 （CPU）的存取为单周期存取。 I 总线 （DTC）的存取因内部时钟 （Iϕ）与总线时钟 （Bϕ）的比或 DTC 运行状态等不同而变化。在睡眠模式、软件待机模式、上电复位、手动复位，可保持内部 RAM 的内容。但在深度软件待机模式及硬件待机模式时，不保持内部 RAM 的内容。

RAM 控制寄存器 （RAMCR）的 RAME 位可控制 RAM 有效或无效。详情参阅 “22.3.7 RAM 控制寄存器 （RAMCR）”。

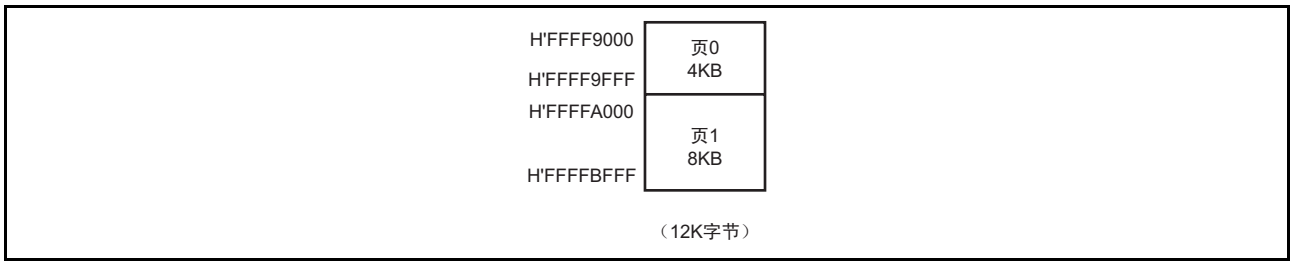


图 21.1 内部 RAM12KB 版的地址

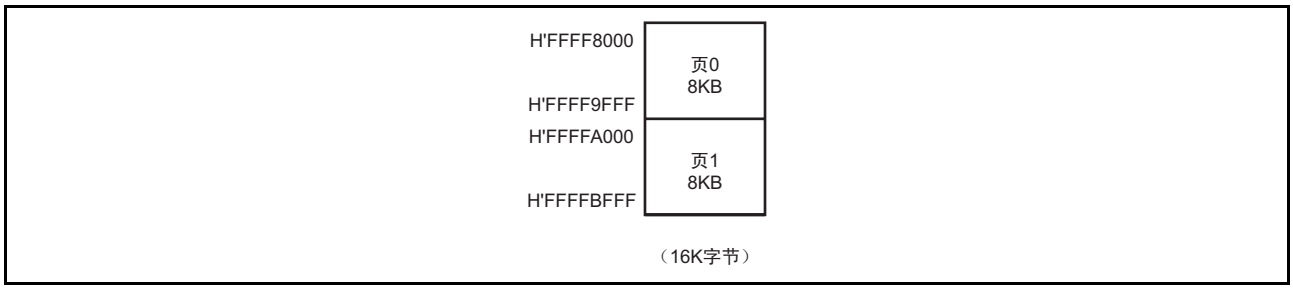


图 21.2 内部 RAM16KB 版的地址

21.1 使用时的注意事项

21.1.1 模块待机模式的设定

待机控制寄存器可设定禁止 / 允许 RAM 存取。为初始值时，允许 RAM 存取。通过设定模块待机模式，可禁止 RAM 存取。详情参阅 “第 22 章 低功耗模式”。

21.1.2 地址错误

对 RAM 执行导致地址错误的写入时，可能破坏 RAM 的内容。

21.1.3 RAM 的初始值

通电后，写入 RAM 前， RAM 的初始值不定。

第 22 章　低功耗模式

本 LSI 支持睡眠模式、软件待机模式、深度软件待机模式、硬件待机模式及模块待机功能等低功耗模式。

22.1　特点

- 支持睡眠模式/软件待机模式/模块待机模式/深度软件待机模式/硬件待机模式。

22.1.1　低功耗模式的种类

低功耗模式具有以下模式及功能：

1. 睡眠模式
2. 软件待机模式
3. 深度软件待机模式
4. 模块待机功能
5. 硬件待机模式

从程序执行状态转移至各模式的方法、各模式的 CPU 及外围模块等状态及各模式的解除方法如表 22.1 所示。

表 22.1　低功耗模式的状态

低功耗模式	转移方法	状　态					解除方法
		CPG	CPU	CPU 寄存器	内部存储器	内部外围模块	
睡眠模式	在 STBCR1 的 STBY 位为 0 的状态，执行 SLEEP 指令	运行	停止	保持	运行	运行	复位
软件待机模式	在 STBCR1 的 STBY 位为 1 且 STBCR6 的 STBYMD 位为 1 的状态，执行 SLEEP 指令	停止	停止	保持	停止（保持内容）	停止	(1)NMI、IRQ 引起的中断 (2) $\overline{\text{RES}}$ 引脚引起的上电复位
深度软件待机模式	在 STBCR1 的 STBY 位为 1 且 STBCR6 的 STBYMD 位为 0 的状态，执行 SLEEP 指令	停止	停止	不定	停止（内容不定）	停止	$\overline{\text{RES}}$ 引脚引起的上电复位
模块待机功能	将 STBCR2 ～ 5 的 MSTP 位置 1	运行	运行	保持	指定模块停止（保持内容）	指定模块停止	(1) MSTP 位清 0 (2) 上电复位（MSTP 位初始值为 0 的模块）
硬件待机模式	将 $\overline{\text{HSTBY}}$ 引脚设置为低电平	停止	停止	不定	停止（内容不定）	停止	$\overline{\text{RES}}$ 引脚引起的上电复位

【注】 有关各模式的内部外围模块的寄存器状态及引脚状态，详情分别参阅“24.3 各运行模式的寄存器状态”与“附录 A. 引脚状态”。

22.2　输入 / 输出引脚

低功耗模式相关的引脚结构如表 22.2 所示。

表 22.2　引脚结构

名称	引脚名称	输入 / 输出	功　能
上电复位	$\overline{\text{RES}}$	输入	为上电复位输入信号，低电平时上电复位。
手动复位	$\overline{\text{MRES}}$	输入	为手动复位输入信号，低电平时手动复位。
硬件待机	$\overline{\text{HSTBY}}$	输入	为硬件待机输入信号，低电平时硬件待机。无任何输入时，在内部上拉。

22.3　寄存器说明

低功耗模式相关的寄存器如下。有关寄存器的地址及各处理状态的寄存器状态，详情参阅“第 24 章　寄存器一览表”。

表 22.3　寄存器结构

寄存器名称	简称	R/W	初始值	地址	存取长度
待机控制寄存器 1	STBCR1	R/W	H'00	H'FFFFE802	8
待机控制寄存器 2	STBCR2	R/W	H'38	H'FFFFE804	8
待机控制寄存器 3	STBCR3	R/W	H'FF	H'FFFFE806	8
待机控制寄存器 4	STBCR4	R/W	H'FF	H'FFFFE808	8
待机控制寄存器 5	STBCR5	R/W	H'03	H'FFFFE80A	8
待机控制寄存器 6	STBCR6	R/W	H'00	H'FFFFE80C	8
RAM 控制寄存器	RAMCR	R/W	H'10	H'FFFFE880	8

22.3.1　待机控制寄存器 1（STBCR1）

STBCR1 为可读取 / 写入的 8 位寄存器，指定低功耗模式的状态。

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

位	位名称	初始值	R/W	说　明
7	STBY	0	R/W	待机 指定向软件待机模式转移。 0：通过执行 SLEEP 指令，转移至睡眠模式 1：通过执行 SLEEP 指令，转移至软件待机模式 / 深度软件待机模式
6 ～ 0	—	均为 0	R	保留位 读取值、写入值总是为 0。

22.3.2 待机控制寄存器 2（STBCR2）

STBCR2 为可读取 / 写入的 8 位寄存器，控制低功耗时各模块的运行。

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

位	位名称	初始值	R/W	说　明
7	MSTP7	0	R/W	模块停止位 7 本位置 1 时，停止向 RAM 提供时钟。 0：RAM 运行 1：停止向 RAM 提供时钟
6	MSTP6	0	R/W	模块停止位 6 本位置 1 时，停止向 ROM 提供时钟。 0：ROM 运行 1：停止向 ROM 提供时钟
5	—	1	R	保留位 读取值、写入值总是为 1。
4	MSTP4	1	R/W	模块停止位 4 本位置 1 时，停止向 DTC 提供时钟。 0：DTC 运行 1：停止向 DTC 提供时钟
3	—	1	R	保留位 读取值、写入值总是为 1。
2 ～ 0	—	均为 0	R	保留位 读取值、写入值总是为 0。

22.3.3　待机控制寄存器 3（STBCR3）

STBCR3 为可读取 / 写入的 8 位寄存器，控制低功耗时各模块的运行。

位：	7	6	5	4	3	2	1	0
	-	-	MSTP 13	MSTP 12	MSTP 11	MSTP 10	MSTP 9*	MSTP 8
初始值：	1	1	1	1	1	1	1	1
R/W：	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
7、6	—	均为 1	R	保留位 读取值、写入值总是为 1。
5	MSTP13	1	R/W	模块停止位 13 本位置 1 时，停止向 SCI_2 提供时钟。 0：SCI_2 运行 1：停止向 SCI_2 提供时钟
4	MSTP12	1	R/W	模块停止位 12 本位置 1 时，停止向 SCI_1 提供时钟。 0：SCI_1 运行 1：停止向 SCI_1 提供时钟
3	MSTP11	1	R/W	模块停止位 11 本位置 1 时，停止向 SCI_0 提供时钟。 0：SCI_0 运行 1：停止向 SCI_0 提供时钟
2	MSTP10	1	R/W	模块停止位 10 本位置 1 时，停止向 SSU 提供时钟。 0：SSU 运行 1：停止向 SSU 提供时钟
1	MSTP9*	1	R/W	模块停止位 9（仅适用于 SH7142） 本位置 1 时，停止向 RCAN-ET_1 提供时钟。 0：RCAN-ET_1 运行 1：停止向 RCAN-ET_1 提供时钟
0	MSTP8	1	R/W	模块停止位 8 本位置 1 时，停止向 RCAN-ET_0 提供时钟。 0：RCAN-ET_0 运行 1：停止向 RCAN-ET_0 提供时钟

【注】 *　在 SH7147，为保留位，且读取值、写入值总是为 1。

22.3.4　待机控制寄存器 4 （STBCR4）

STBCR4 为可读取 / 写入的 8 位寄存器，控制低功耗时各模块的运行。

位：	7	6	5	4	3	2	1	0
	MSTP 23	MSTP 22	MSTP 21	MSTP 20	MSTP 19	-	-	-
初始值：	1	1	1	1	1	1	1	1
R/W：	R/W	R/W	R/W	R/W	R/W	R	R	R

位	位名称	初始值	R/W	说　明
7	MSTP23	1	R/W	模块停止位 23 本位置 1 时，停止向 MTU2S 提供时钟。 0：MTU2S 运行 1：停止向 MTU2S 提供时钟
6	MSTP22	1	R/W	模块停止位 22 本位置 1 时，停止向 MTU2 提供时钟。 0：MTU2 运行 1：停止向 MTU2 提供时钟
5	MSTP21	1	R/W	模块停止位 21 本位置 1 时，停止向 CMT 提供时钟。 0：CMT 运行 1：停止向 CMT 提供时钟
4	MSTP20	1	R/W	模块停止位 20 本位置 1 时，停止向 A/D_1 提供时钟。 0：A/D_1 运行 1：停止向 A/D_1 提供时钟
3	MSTP19	1	R/W	模块停止位 19 本位置 1 时，停止向 A/D_0 提供时钟。 0：A/D_0 运行 1：停止向 A/D_0 提供时钟
2 ～ 0	—	均为 1	R	保留位 读取值、写入值总是为 1。

22.3.5 待机控制寄存器 5（STBCR5）

STBCR5 为可读取 / 写入的 8 位寄存器，指定低功耗模式的状态。

位：	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	MSTP 25	MSTP 24
初始值：	0	0	0	0	0	0	1	1
R/W：	R	R	R	R	R	R	R/W	R/W

位	位名称	初始值	R/W	说　明
7 ～ 2	—	均为 0	R	保留位 读取值、写入值总是为 0。
1	MSTP25	1	R/W	模块停止位 25 本位置 1 时，停止向 AUD 提供时钟。 0：AUD 运行 1：停止向 AUD 提供时钟
0	MSTP24	1	R/W	模块停止位 24 本位置 1 时，停止向 UBC 提供时钟。 0：UBC 运行 1：停止向 UBC 提供时钟

22.3.6 待机控制寄存器 6（STBCR6）

STBCR6 为可读取 / 写入的 8 位寄存器，指定低功耗模式的状态。

位：	7	6	5	4	3	2	1	0
	AUD SRST	HIZ	-	-	-	-	STBY MD	-
初始值：	0	0	0	0	0	0	0	0
R/W：	R/W	R/W	R	R	R	R	R/W	R

位	位名称	初始值	R/W	说　明
7	AUDSRST	0	R/W	AUD 软件复位 由软件控制 AUD 复位。 对本位写入 0 时， AUD 模块为上电复位状态。 0：设置为 AUD 复位状态 1：解除 AUD 复位 本位置 1 时， STBCR5 的 MSTP25 位必须为 0。
6	HIZ	0	R/W	端口高阻抗状态 在软件待机模式，选择保持引脚状态或将引脚设置为高阻抗状态 0：在软件待机模式，保持引脚状态 1：在软件待机模式，将引脚设置为高阻抗状态
5 ～ 2	—	均为 0	R	保留位 读取值、写入值总是为 0。
1	STBYMD	0	R/W	选择软件待机模式 在 STBCR1 的 STBY 位为 1 的状态执行 SLEEP 指令时，选择转移至软件待机模式或深度软件待机模式。 0：转移至深度软件待机模式 1：转移至软件待机模式
0	—	0	R	保留位 读取值、写入值总是为 0。

22.3.7 RAM 控制寄存器 （RAMCR）

RAMCR 为可读取 / 写入的 8 位寄存器，指定允许或禁止存取内部 RAM。

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

位	位名称	初始值	R/W	说 明
7 ～ 5	—	均为 0	R	保留位 读取值、写入值总是为 0。
4	RAME	1	R/W	RAM 允许 选择内部 RAM 有效或无效。 0：内部 RAM 无效 1：内部 RAM 有效 本位清 0 时，不可存取内部 RAM。此时，从内部 RAM 读取的值及取指令的值不定，可忽略对内部 RAM 的写入。 本位清 0 并设置内部 RAM 无效时，不得紧接着 RAMCR 的写入指令设置内部 RAM 存取指令，如果设置内部 RAM 存取指令，则无法保证正常存取。 本位置 1 并设置内部 RAM 有效时，必须紧接着 RAMCR 的写入指令设置 RAMCR 读取指令，如果紧接着 RAMCR 的读取指令设置内部 RAM 存取指令，则无法保证正常存取。
3 ～ 0	—	均为 0	R	保留位 读取值、写入值总是为 0。

22.4 睡眠模式

22.4.1 向睡眠模式的转移

在 STBCR1 的 STBY 位为 0 的状态，如果执行 SLEEP 指令，则从程序执行状态转移至睡眠模式。但在释放总线 （向 $\overline{\text{BREQ}}$ 引脚输入低电平）期间，不可转移至睡眠模式。 CPU 在执行 SLEEP 指令后停止运行，但保持 CPU 的寄存器内容，且内部外围模块继续运行。

22.4.2 睡眠模式的解除

由复位解除睡眠模式。

不得由中断解除睡眠模式。

(1) 由复位解除

由 $\overline{\text{RES}}$ 引脚引起的上电复位、 $\overline{\text{MRES}}$ 引脚引起的手动复位、 WDT 引起的内部上电复位 / 手动复位，解除睡眠模式。

22.5 软件待机模式

22.5.1 向软件待机模式的转移

在 STBCR1 的 STBY 位且 STBCR6 的 STBYMD 位均为 1 的状态，如果执行 SLEEP 指令，则从程序执行状态转移至软件待机模式。但在释放总线（向 $\overline{\text{BREQ}}$ 引脚输入低电平）期间，不可转移至软件待机模式。另外，必须在 DTC 停止后执行 SLEEP 指令。在软件待机模式，不仅 CPU 停止，时钟及内部外围模块也停止。

保持 CPU 的寄存器内容与内部 RAM 的数据，某些内部外围模块的寄存器会被初始化。有关软件待机模式的外围模块的寄存器状态及引脚状态，详情分别参阅 **“24.3 各运行模式的寄存器状态”**与 **“附录 A. 引脚状态”**。

向软件待机模式转移的步骤如下所示：

1. 将WDT的定时器控制寄存器（WTCSR）的TME位置0，使WDT停止。
2. 将WDT的定时器计数器（WTCNT）置0，并给WTCSR寄存器的CKS2～CKS0位设定与指定的振荡稳定时间对应的值。
3. DTC运行时，停止DTC。
4. 释放总线（向$\overline{\text{BREQ}}$引脚输入低电平）时，可获取总线（向$\overline{\text{BREQ}}$引脚输入高电平）。
5. 将STBCR1的STBY位及STBCR6的STBYMD位设定为1后，执行SLEEP指令。
6. 进入软件待机模式， LSI内部的时钟停止。

22.5.2 软件待机模式的解除

由中断（NMI、 IRQ）及复位，解除软件待机模式。

(1) 由中断解除

可由内部 WDT 热启动。检测 NMI、 IRQ 中断（边沿检测）时，经过 WDT 的定时器控制 / 状态寄存器设定的时间后，对整个 LSI 提供时钟，并解除软件待机模式。此后，执行中断异常处理。

但 IRQ 的中断优先级低于对 CPU 的状态寄存器（SR）设定的中断屏蔽级时，不接受中断请求，也不解除软件待机模式。

NMI 引脚设定为下降沿检测时，必须在转移至软件待机模式前，将 NMI 引脚设置为高电平； NMI 引脚设定为上升沿检测时，必须在转移至软件待机模式前，将 NMI 引脚设置为低电平。

同样，IRQ 引脚设定为下降沿检测时，必须在转移至软件待机模式前，将 IRQ 引脚设置为高电平； IRQ 引脚设定为上升沿检测时，必须在转移至软件待机模式前，将 IRQ 引脚设置为低电平。

(2) 由上电复位解除

由 $\overline{\text{RES}}$ 引脚引起的上电复位，解除软件待机模式。 $\overline{\text{RES}}$ 引脚必须在时钟振荡稳定前保持低电平。

22.6 深度软件待机模式

22.6.1 向深度软件待机模式的转移

在 STBCR1 的 STBY 位为 1 且 STBCR6 的 STBYMD 位为 0 的状态，如果执行 SLEEP 指令，则从程序执行状态转移至深度软件待机模式。但在释放总线（向 $\overline{\text{BREQ}}$ 引脚输入低电平）期间，不可转移至深度软件待机模式。并且必须在 DTC 停止运行后，执行 SLEEP 指令。在深度软件待机模式，不仅 CPU 停止，时钟及内部外围模块也停止，并切断本 LSI 的内部电源。

CPU 的寄存器内容与内部 RAM 的数据不定，内部外围模块的寄存器也被初始化。有关深度软件待机模式的引脚状态，详情参阅 **“附录 A. 引脚状态”**。

向深度软件待机模式转移的步骤如下所示：

1. 将WDT的定时器控制寄存器（WTCSR）的TME位置0，使WDT停止。
2. DTC运行时，停止DTC。
3. 释放总线（向$\overline{\text{BREQ}}$引脚输入低电平）时，可获取总线（向$\overline{\text{BREQ}}$引脚输入高电平）。
4. 将STBCR1的STBY位设定为1、STBCR6的STBYMD位设定为0后，执行SLEEP指令。
5. 进入深度软件待机模式，LSI内部的时钟停止，并切断本LSI的内部电源。

22.6.2 深度软件待机模式的解除

由 $\overline{\text{RES}}$ 引脚引起的上电复位，解除深度软件待机模式。$\overline{\text{RES}}$ 引脚必须在时钟振荡稳定前保持低电平。

22.7 模块待机功能

22.7.1 向模块待机功能的转移

通过将待机控制寄存器 2 ～ 5（STBCR2 ～ 5）的各 MSTP 位置 1，可停止对各自对应的内部外围模块提供时钟，且使用该功能，可降低普通模式的功耗。

不得存取已设定为模块待机模式的外围模块寄存器。有关模块待机模式的外围模块寄存器状态，详情参阅 **“24.3 各运行模式的寄存器状态”**。

22.7.2 模块待机功能的解除

将 STBCR2 ～ 5 的各 MSTP 位清 0，可解除模块待机功能，还可由上电复位解除 MSTP 位初始值为 0 的模块。

22.8 硬件待机模式

22.8.1 向硬件待机模式的转移

如果向 $\overline{\text{HSTBY}}$ 引脚输入低电平，则转移至硬件待机模式。在硬件待机模式，不仅 CPU 停止，时钟及内部外围模块也停止，并切断本 LSI 的内部电源。CPU 的寄存器内容及内部 RAM 的数据不定，内部外围模块的寄存器也被初始化。通过外部引脚输入转移至硬件待机模式，因此与当前的 LSI 状态无关，异步转移至该模式。有关硬件待机模式的引脚状态，详情参阅 **“附录 A. 引脚状态”**。在硬件待机模式，不得改变模式引脚（MD1、MD0）的状态。

22.8.2 硬件待机模式的解除

由 $\overline{\text{HSTBY}}$ 引脚及 $\overline{\text{RES}}$ 引脚解除硬件待机模式。

在 $\overline{\text{RES}}$ 引脚为低电平的状态，如果将 $\overline{\text{HSTBY}}$ 引脚设置为高电平，则内部时钟开始振荡。此时，$\overline{\text{RES}}$ 引脚必须在内部时钟振荡稳定之前保持低电平。之后，如果 $\overline{\text{RES}}$ 引脚为高电平，则经过上电复位异常处理，转移至程序执行状态。

22.8.3 硬件待机模式的时序

硬件待机模式的时序例子如图 22.1 所示。

在 $\overline{\text{RES}}$ 引脚为低电平的状态，如果 $\overline{\text{HSTBY}}$ 引脚为低电平，则转移至硬件待机模式。将 $\overline{\text{HSTBY}}$ 设置为高电平且经过时钟的振荡稳定时间后，通过将 $\overline{\text{RES}}$ 引脚从低电平设置为高电平，解除硬件待机模式。

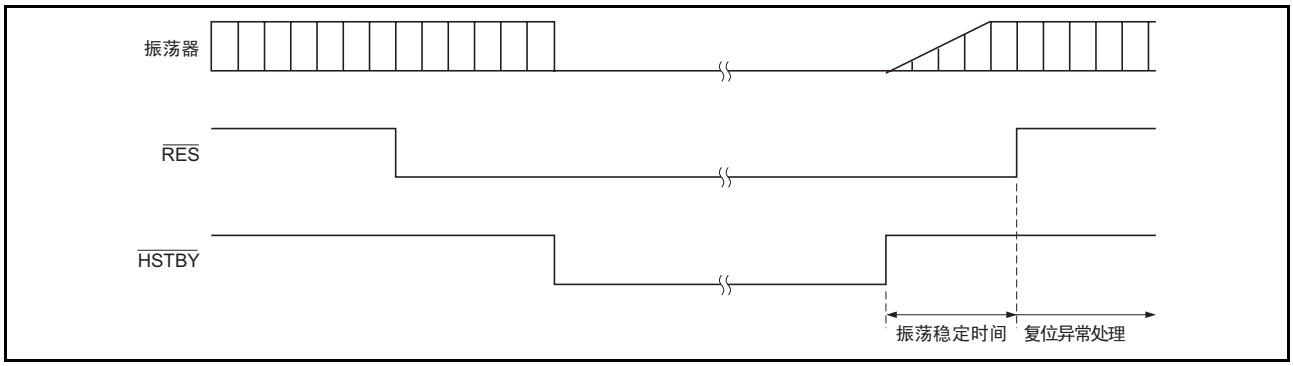


图 22.1 硬件待机模式的时序

22.9 使用时的注意事项

22.9.1 振荡稳定待机时的消耗电流

振荡稳定待机时，消耗电流会增加。

22.9.2 SLEEP 指令执行时

通过执行 SLEEP 指令，转移至睡眠模式或软件待机模式时，必须执行以下任意一项措施：

措施 A. 执行 SLEEP 指令前，必须停止 DTC 运行及不产生内部外围模块的中断、 IRQ 中断、 NMI 中断。

措施 B. 执行 SLEEP 指令前，必须将 FRQCR 的值改写为初始值 H'36DB，并虚读 2 次 FRQCR。

第 23 章　高级用户调试器 （AUD）

AUD 提供在产品芯片安装状态支持用户程序调试的功能。使用 AUD，可组成获取转移跟踪数据和监控 / 协调内部 RAM 数据等的简易仿真器。

23.1　特点

- 共8个输入/输出引脚

通过转换 AUDMD， AUD 可使用以下两种模式。

- 转移跟踪模式
- RAM监控模式

(1)　转移跟踪模式

- 转移跟踪功能：可选择跟踪转移源和转移目标，或跟踪转移源、转移目标的其中一个
- 窗口数据跟踪功能：支持窗口A和窗口B的2个通道窗口。跟踪窗口内产生存储器存取的地址及其数据。
- 支持8/16/32位的数据长度
- 跟踪对象总线：L总线、 I总线
- 输出地址信息时，以少量位数仅输出信息中有差异的地址信息。
- 内置8级FIFO
- 全跟踪功能：即使未来得及输出跟踪数据，也在停止CPU的同时输出所有跟踪数据
- 实时跟踪功能：只要CPU不停止，即输出允许输出的跟踪数据
- AUD工作频率：支持CPU内核时钟比×1、 ×1/2、 ×1/4、 ×1/8。但设定工作频率时，不能超过20MHz。

(2)　RAM 监控模式

- 读取/写入连接内部/外部总线的模块
- 从外部将地址写入AUDATA时，输出该地址对应的数据。
- 从外部将地址、数据写入AUDATA时，向该地址传送数据。
  AUDCK输入时钟必须设置小于等于10MHz及1/4Pϕ。

使用 AUD 时，需要将待机控制寄存器 6 （STBCR6）的 AUD 软件复位位 （AUDSRST）的值清 0。有关 AUDSRST 位，详情参阅 “22.3.6 待机控制寄存器 6 （STBCR6）”。

AUD 框图如图 23.1 所示。

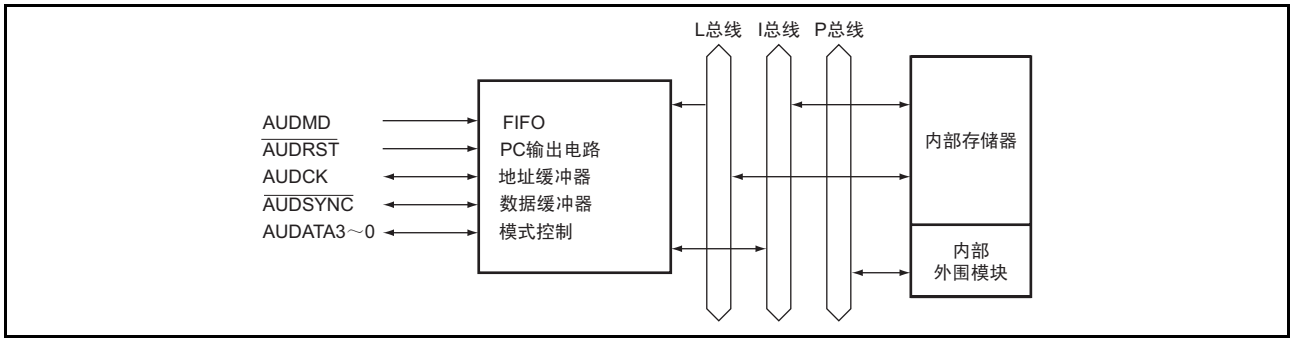


图 23.1　AUD 框图

23.2　输入 / 输出引脚

表 23.1　引脚结构

名称	引脚名称	功　　能	
		转移跟踪模式	RAM 监控模式
AUD 复位	$\overline{\text{AUDRST}}$	输入 AUD 复位	输入 AUD 复位
AUD 同步信号	$\overline{\text{AUDSYNC}}$	输出数据起始位置识别信号	输入数据起始位置识别信号
AUD 时钟	AUDCK	输出同步时钟	输入外部时钟
AUD 模式	AUDMD	输入模式选择 （L）	输入模式选择 （H）
AUD 数据	AUDATA3 ～ 0	输出转移目标 / 转移源地址	输入监控地址或输入 / 输出数据

23.2.1　通用引脚说明

表 23.2　通用引脚

引脚名称	说　　明
AUDMD	转换本引脚的输入电平，可选择模式。 L：转移跟踪模式 H：RAM 监控模式 必须在 $\overline{\text{AUDRST}}$ 为低电平的状态转换本引脚的输入电平。
$\overline{\text{AUDRST}}$	对本引脚输入低电平，可初始化 AUD 内的缓冲器及逻辑。 输入低电平时， AUD 为复位状态， AUD 内的缓冲器及处理状态复位。确定 AUDMD 的电平后，如果返回高电平，则以选定的模式运行。

23.2.2　转移跟踪模式的引脚说明

表 23.3　转移跟踪模式的引脚说明

引脚名称	说　　明
AUDCK	AUD 工作频率：输出 CPU 内核时钟比 ×1、 ×1/2、 ×1/4、 ×1/8。 与 AUDATA 取得同步的时钟。 但设定时不可超过 20MHz。
$\overline{\text{AUDSYNC}}$	本引脚表示 AUDATA 的输出是否有效。 H：未输出有效数据时 L：输出有效数据时
AUDATA3 ～ 0	分时输出以下信息。 • AUD 总线命令 • 转移目标 / 转移源地址

23.2.3　RAM 监控模式的引脚说明

表 23.4　RAM 监控模式的引脚说明

引脚名称	说　　明
AUDCK	本引脚为外部时钟输入引脚，必须输入用于调试的时钟，可输入的频率必须小于等于 10MHz 及 1/4Pϕ。
$\overline{\text{AUDSYNC}}$	从外部向 AUDATA 输入命令，准备好必要的数据前，本引脚不可设置为有效。详情参阅后述协议。
AUDATA3 ～ 0	从外部输入命令时，发送 Ready 后输出数据，$\overline{\text{AUDSYNC}}$ 设置为无效后，开始输出。详情参阅后述协议。

23.3　转移跟踪模式

23.3.1　寄存器说明

表 23.5　寄存器结构

寄存器名称	简称	R/W	初始值	地址	存取长度
AUD 控制寄存器	AUCSR	R/W	H'00000000	H'FFFFF400	32
AUD 窗口 A 起始地址寄存器	AUWASR	R/W	不定	H'FFFFF404	32
AUD 窗口 A 结束地址寄存器	AUWAER	R/W	不定	H'FFFFF408	32
AUD 窗口 B 起始地址寄存器	AUWBSR	R/W	不定	H'FFFFF40C	32
AUD 窗口 B 结束地址寄存器	AUWBER	R/W	不定	H'FFFFF410	32
AUD 扩展控制寄存器	AUECSR	R/W	H'00000001	H'FFFFF414	32

23.3.2　AUD 控制寄存器 （AUCSR）

AUCSR 为可读取 / 写入的 32 位寄存器。通过上电复位、手动复位、$\overline{\text{AUDRST}}$、AUD 软件复位、硬件待机等初始化。

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
	CLK[1:0]		-	-	BRE	OC[1:0]		BR	WA[1:0]		WB[1:0]		-	-	TM	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	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

位	位名称	初始值	R/W	说　　明
31 ～ 16	—	均为 0	R	保留位 读取值、写入值总是为 0。
15、 14	CLK[1:0]	00	R/W	选择 AUD 时钟 通过 CPU 时钟比设定 AUD 内部运行时钟。 00：选择 CPU 时钟比 1/8 01：选择 CPU 时钟比 1/4 10：选择 CPU 时钟比 1/2 11：选择 CPU 时钟比 1/1 【注】必须在 AUCSR 的 EN 位为 0 的状态更改 CLK 位。
13、 12	—	均为 0	R	保留位 读取值、写入值总是为 0。
11 8	BRE BR	0 0	R/W R/W	转移跟踪功能 通过设定本位， AUD 可跟踪转移目标地址、转移源地址。 00：转移跟踪无效 01：转移跟踪有效、同时输出转移源 / 转移目标 10：转移跟踪有效、仅输出转移源 11：转移跟踪有效、仅输出转移目标 该寄存器的设定，从输出设定后执行的跟踪开始有效。设定后的输出格式不反映设定前读入 FIFO 的转移跟踪数据。
10、 9	OC[1:0]	00	R/W	输出计数器模式 因转移跟踪等输出的跟踪信息，通常仅输出地址有变的低位部分，但在某个固定周期，会输出全部 32 位的地址。 OC 位指定输出全部 32 位地址信息的周期。 AUCSR 的 OC 位为 B'11 时，除了复位后第 1 次输出 AUDATA 的跟踪地址外，总是输出地址的差异部分； OC 位为 B'00 时，每输出 128 个跟踪信息，就输出 32 位地址。 00：每输出 128 个跟踪信息就输出全部地址 01：保留 10：保留 11：总是仅输出地址有变的低位部分
7、 6	WA[1:0]	00	R/W	窗口 A 数据跟踪功能 通过设定本位， AUD 可跟踪窗口 A 指定区域的存储器存取。可指定读取存取、写入存取、读取 / 写入存取为跟踪条件。 并可指定跟踪 L 总线或 I 总线。详情参阅 “23.3.7 AUD 扩展控制寄存器 （AUECSR）”。 00：窗口 A 数据跟踪功能无效 01：仅写入存取为跟踪对象 10：仅读取存取为跟踪对象 11：读取 / 写入均为跟踪对象
5、 4	WB[1:0]	00	R/W	窗口 B 数据跟踪功能 通过设定本位， AUD 可跟踪窗口 B 指定区域的存储器存取。可指定读取存取、写入存取、读取 / 写入存取为跟踪条件。 并可指定跟踪 L 总线或 I 总线。详情参阅 “23.3.7 AUD 扩展控制寄存器 （AUECSR）”。 00：窗口 B 数据跟踪功能无效 01：仅写入存取为跟踪对象 10：仅读取存取为跟踪对象 11：读取 / 写入均为跟踪对象

位	位名称	初始值	R/W	说　　明
3、2	—	均为 0	R	保留位 读取值、写入值总是为 0。
1	TM	0	R/W	跟踪模式 保存 AUD 各种跟踪信息的 FIFO 缓冲器已满时，指定 CPU 的运行。 0：全跟踪模式 （输出所有已产生跟踪的模式） 1：实时跟踪模式 （不停止 CPU，实时输出跟踪信息的模式）
0	EN	0	R/W	AUD 允许 本位置 1 时， AUD 选择 BRE、 BR、 WA0、 WA1、 WB0、 WB1 位指定的跟踪功能。 本位为 0 时，不执行由 BRE、BR、WA0、WA1、WB0、WB1 位指定的跟踪。 0：AUD 跟踪功能无效 1：AUD 跟踪功能有效

23.3.3　AUD 窗口 A 起始地址寄存器 （AUWASR）

AUWASR 为可读取 / 写入的 32 位寄存器。本寄存器指定跟踪对象窗口 A 的起始地址， AUD 窗口 A 结束地址寄存器 （AUWBER）指定结束地址。窗口 A 定义为满足 （AUWASR ≤窗口 A ≤ AUWAER）条件的区域。通过上电复位、手动复位、 $\overline{\text{AUDRST}}$、 AUD 软件复位、硬件待机等初始化。

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

23.3.4　AUD 窗口 A 结束地址寄存器 （AUWAER）

AUWAER 为可读取 / 写入的 32 位寄存器，与 AUWASR 同时指定窗口 A 区域。通过上电复位、手动复位、 $\overline{\text{AUDRST}}$、 AUD 软件复位、硬件待机等初始化。

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

23.3.5　AUD 窗口 B 起始地址寄存器 （AUWBSR）

AUWBSR 为可读取 / 写入的 32 位寄存器，本寄存器指定跟踪对象窗口 B 的起始地址。 AUD 窗口 B 结束地址寄存器 （AUWBER）指定结束地址。窗口 B 定义为满足 （AUWBSR ≤窗口 B ≤ AUWBER）条件的区域。通过上电复位、手动复位、 $\overline{\text{AUDRST}}$、 AUD 软件复位、硬件待机等初始化。

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

23.3.6　AUD 窗口 B 结束地址寄存器 （AUWBER）

AUWBER 为可读取 / 写入的 32 位寄存器，与 AUWBSR 同时指定窗口 B 区域。通过上电复位、手动复位、 $\overline{\text{AUDRST}}$、 AUD 软件复位、硬件待机等初始化。

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

23.3.7　AUD 扩展控制寄存器 （AUECSR）

AUECSR 为可读取 / 写入的 32 位寄存器。通过上电复位、手动复位、 $\overline{\text{AUDRST}}$、 AUD 软件复位、硬件待机等初始化。

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
	-	-	-	-	-	-	WAOB	-	-	WBOB	-	-	TREX	TRSB	TRGN	-
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/W	R	R	R/W	R	R	R/W	R/W	R/W	R

位	位名称	初始值	R/W	说　明
31 ～ 10	—	均为 0	R	保留位 读取值、写入值总是为 0。
9	WAOB	0	R/W	选择窗口 A 的跟踪总线 指定在窗口 A 执行哪个内部总线的数据跟踪。 0：指定 L 总线 1：指定 I 总线 【注】必须在 AUCSR 寄存器的 EN 位为 0 的状态更改本位。
8、7	—	均为 0	R	保留位 读取值、写入值总是为 0。
6	WBOB	0	R/W	选择窗口 B 的跟踪总线 指定在窗口 B 执行哪个内部总线的数据跟踪。 0：指定 L 总线 1：指定 I 总线 【注】必须在 AUCSR 寄存器的 EN 位为 0 的状态更改本位。
5、4	—	均为 0	R	保留位 读取值、写入值总是为 0。
3	TREX	0	R/W	选择异常转移跟踪 转移跟踪时，指定是否跟踪异常转移 （异常、中断、 RTE 指令）。 0：跟踪异常转移 1：不跟踪异常转移
2	TRSB	0	R/W	选择子程序转移跟踪 转移跟踪时，指定是否跟踪子程序转移 （BSR、 BSRF、 JSR、 RTS 指令）。 0：跟踪子程序转移 1：不跟踪子程序转移
1	TRGN	0	R/W	选择普通转移跟踪 转移跟踪时，指定是否跟踪普通转移（BF、BT、BF/S、BT/S、BRA、BRAF、 JMP 指令）。 0：跟踪普通转移 1：不跟踪普通转移
0	—	1	R	保留位 读取值、写入值总是为 1。

23.3.8　运行说明

转移跟踪条件为必须给 AUCSR、AUECSR、AUWASR、AUWAER、AUWBSR、AUWBER 设定任意值，但 AUCSR 的 EN 位必须置 1。

AUD 支持转移跟踪功能、窗口数据跟踪功能，这两种功能可独立运行。此外，可选择实时跟踪模式、全跟综模式作为向外部输出由本功能得到的跟踪数据的模式。

(1)　AUD 总线命令

由 AUD 获得的跟踪信息，通过 AUDATA[3:0]、 $\overline{\text{AUDSYNC}}$ 引脚，与 AUDCK 同步以信息包格式输出。信息包由命令部和 0 ～ 17 个数据部构成，从 AUDATA[3:0] 引脚连续输出。通常，命令部由表示信息包类型的 CMD1 和表示数据部信息包长度的 CMD2 构成。此外还有仅由表示 AUD 状态的 CMD1 构成的特殊信息包和带有追加信息命令部的信息包。 $\overline{\text{AUDSYNC}}$ 的电平，在空闲时及 CMD1 从 AUDATA[3:0] 引脚输出时为高电平；在 CMD1E 和 CMD2 及输出数据时为低电平。命令详情如表 23.6 所示。

表 23.6　AUD 总线命令一览表

命令	CMD1	CMD1E	CMD2	说　　明
STDBY	B'0000	—	—	表示待机状态。 没有持续输出的数据部。
LOST	B'0001	—	—	在实时模式，表示输出数据丢失。 在全跟踪模式，表示 AUD 的 FIFO 是否已满，以及多个跟踪同时产生， CPU 暂时停止运行。
BPC	B'0010	—	(sda)(ssa)	输出转移跟踪信息。在本命令之后，按照转移目标地址、转移源地址、地址校正值 （=PID）的顺序输出数据部 （地址校正值的计算参阅 “23.3.8(3) 计算转移跟踪时的地址”）。 sda：表示转移目标地址的长度。 ssa：表示转移源地址的长度。 sda/ssa=B'00：地址的低 4 位 sda/ssa=B'01：地址的低 8 位 sda/ssa=B'10：地址的低 16 位 sda/ssa=B'11：32 位全地址
WDWM	B'1001	(pt)(bt)	(sa)(sd)	输出窗口数据跟踪的写入存取信息。 在本命令之后，按照写入地址、写入数据的顺序输出数据部。 pt：已获取信息的处理器内核的 1 位标识符，本 LSI 固定为 0。 bt：已获取信息的总线的 3 位标识符。 bt=B'000：表示 L 总线跟踪 bt=B'100：表示 I 总线跟踪 bt=B'001 ～ B'011：保留 bt=B'101 ～ B'111：保留 sa：表示写入地址的长度。 sa=B'00：地址的低 4 位 sa=B'01：地址的低 8 位 sa=B'10：地址的低 16 位 sa=B'11：32 位全地址 sd：表示写入数据的长度。 sd=B'01：字节数据 （8 位） sd=B'10：字数据 （16 位） sd=B'11：长字数据 （32 位）
WDRM	B'1101	(pt)(bt)	(sa)(sd)	输出窗口数据跟踪的读取存取信息。 在本命令之后，按照读取地址、读取数据的顺序输出数据部。 pt：已获取信息的处理器内核标识符。与 WDWM 的 pt 相同。 bt：已获取信息的总线的标识符。与 WDWM 的 bt 相同。 sa：表示读取地址的长度。与 WDWM 的 sa 相同。 sd：表示读取数据的长度。与 WDWM 的 sd 相同。

(2)　转移跟踪

转移跟踪的功能为：根据转移指令的执行和中断的产生，捕捉 PC 变化，并向外部输出转移源地址、转移目标地址。通过设定 AUCSR 的 BRE、BR、EN 位及 AUECSR 的 TREX、TRSB、TRGN 位，跟踪用户程序的转移信息。

转移跟踪的输出例子如图 23.2 ～图 23.4 所示。

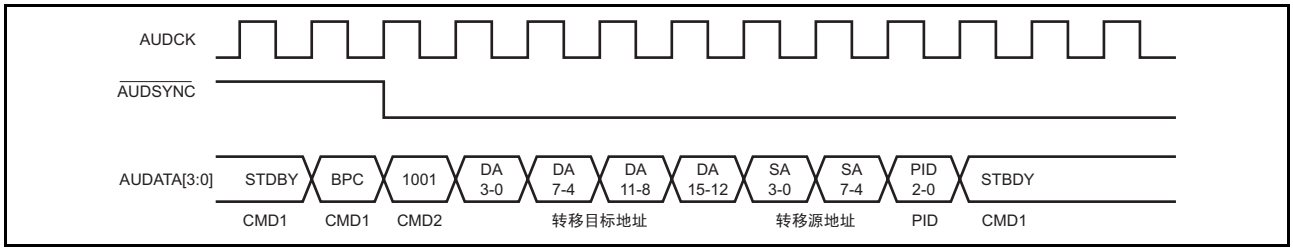


图 23.2　BPC 命令的转移跟踪 （转移目标 / 转移源）

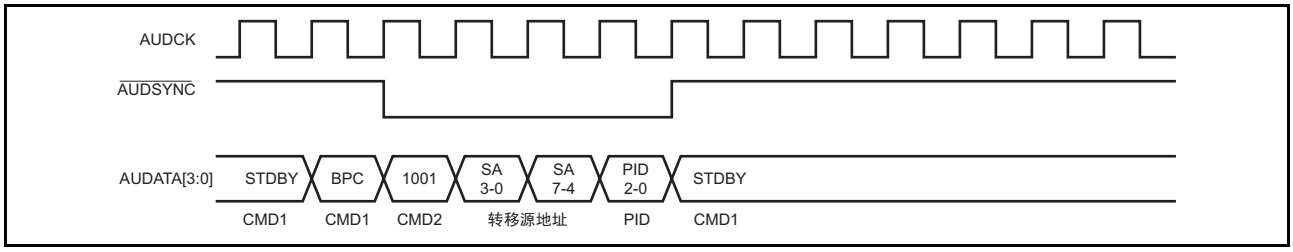


图 23.3　BPC 命令的转移跟踪 （仅转移源）

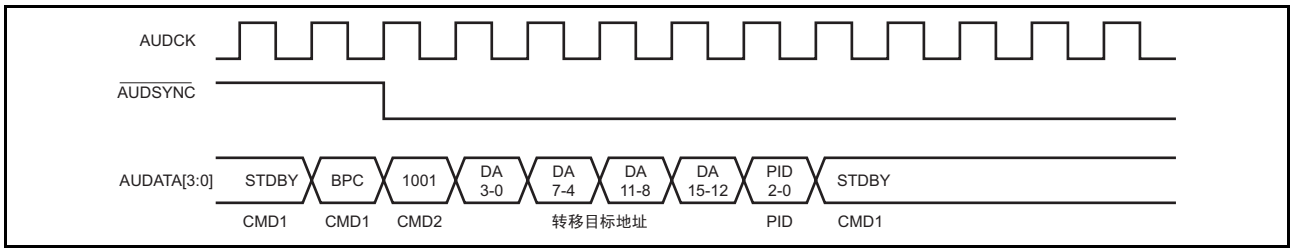


图 23.4　BPC 命令的转移跟踪 （仅转移目标）

不论哪种输出方法， CMD1、 CMD2、 PID 的输出都相同。 CMD1 输出 BPC （B'0010）， CMD2(ssa)/(sda) 也总是输出 BPC （B'0010）。

仅输出转移目标时，上一次转移的转移目标地址为地址差异部分的对比，而非上一次转移的转移源地址。

在用户程序，未产生转移事件 （转移指令的执行及中断等）期间，不输出转移信息；产生转移事件时，在命令之后，按照转移目标地址、转移源地址、 PID 的顺序从 AUDATA[3:0] 输出跟踪信息。以 PFBA 的值（上一次输出的转移目标地址）为基础，通过比较匹配两个地址的高位确定输出转移目标、转移源的地址长度（4/8/16/32 位）。

本算法如图 23.5 所示。

```
定义：PFBA     上次输出的转移目标地址
      CDA      此次输出的转移目标地址
      CSA      此次输出的转移源地址
      PID      生成正确转移源地址的校正值
      TSA      正确转移源地址

*****************************************************************************************
/* PFBA_enable = 0: initial value*/
if(PFBA_enable == 0) {output32bit(CDA[31:0]);}
else if(PFBA[31:4] == CDA[31:4]) {output4bit(CDA[3:0]);}
else if(PFBA[31:8] == CDA[31:8]) {output8bit(CDA[7:0]);}
else if(PFBA[31:16] == CDA[31:16]) {output16bit(CDA[15:0]);}
else {output32bit(CDA[31:0]);}

if(PFBA_enable == 0) {output32bit(CSA[31:0]);}
else if(PFBA[31:4] == CSA[31:4]) {output4bit(CSA[3:0]);}
else if(PFBA[31:8] == CSA[31:8]) {output8bit(CSA[7:0]);}
else if(PFBA[31:16] == CSA[31:16]) {output16bit(CSA[15:0]);}
else {output32bit(CSA[31:0]);}

PFBA_enable = 1;
PFBA = CDA;  /* update PFBA */
*****************************************************************************************

实际转移源地址必须用以下公式计算。
TSA = CSA.(2*PID)
```

图 23.5　转移跟踪的算法

使用本算法可大幅削减应输出的跟踪数据量。只有在 FIFO 有空间时才可保存转移跟踪信息， FIFO 已满时， CPU 的运行因 AUCSR 的 TM 位设定值而不同。

(3)　计算转移跟踪时的地址

(a)　执行转移指令前产生中断时

通过 TSA=CSA，可计算出转移源指令的地址。在其他处理器需要用 TSA=CSA– (2× PID) 计算，为使两个计算式均可对应， PID 的输出值总是为 0。

(b)　计算转移源地址

转移源地址 （TSA）指向哪里，取决于转移的种类。

- 转移指令：
  TSA 指定转移指令的地址。
- 中断：
  TSA 指定中断前已执行指令的地址。
  CDA 可输出中断起始指令地址。
- 异常：
  产生异常时的指令为完成型指令时， TSA 指定下一条指令的地址。
  产生异常时的指令为再执行型指令时， TSA 指定其指令的地址。
  CDA 可输出异常程序的起始指令地址。

(4)　窗口数据跟踪

窗口数据跟踪的功能为：向外部输出 2 个地址指针所指区域 （称为窗口）产生的存储器存取信息。 AUD 支持 A、 B2 个通道的窗口。窗口 A 的范围由 AUWASR 和 AUWAER 指定，窗口 B 的范围由 AUWBSR 和 AUWBER 指定。设定 AUCSR 的 WA0、 WA1、 WB0、 WB1 位后，通过将 EN 位置 1，跟踪用户程序的存储器存取信息。

窗口数据跟踪例子如图 23.6 所示。

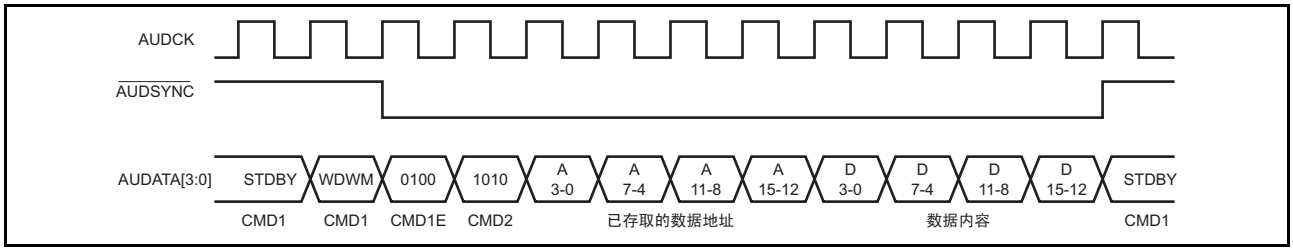


图 23.6　WDWM 命令的窗口数据跟踪

在用户程序，存取窗口区域的数据时，在命令之后，按照存取的地址、数据内容的顺序，从 AUDATA[3:0] 输出跟踪信息。与转移跟踪相同，以 PFDA 的值 （上一次输出的转移目标地址）为基础，通过比较匹配两个地址的高位确定输出地址长度 （4/8/16/32 位）。根据存取长度 （8/16/32 位）可输出数据内容。本算法如图 23.7 所示。

```
定义：PFDA    上次以32位长度输出的数据地址
      CDA     此次输出的数据地址

/* PFDA_enable = 0: initial value*/
if(PFDA_enable == 0) {output32bit(CDA[31:0]);}
else if(PFDA[31:4] == CDA[31:4]) {output4bit(CDA[3:0]);}
else if(PFDA[31:8] == CDA[31:8]) {output8bit(CDA[7:0]);}
else if(PFDA[31:16] == CDA[31:16]) {output16bit(CDA[15:0]);}
else {output32bit(CDA[31:0]);}

PFDA_enable = 1;
PFDA = CDA;  /* update PFDA */
```

图 23.7　窗口数据跟踪的算法

使用本算法可大幅削减应输出的跟踪数据量。只有在 FIFO 有空间时才保存数据跟踪信息。 FIFO 已满时的运行，因 AUCSR 的 TM 位设定值而不同。

(5)　实时跟踪模式

在实时跟踪模式，经过 AUDATA[3:0]，实时向外部输出由转移跟踪、窗口数据跟踪得到的各种跟踪数据。AUCSR 的 TM 位置 1，可选择实时跟踪模式。

在实时跟踪模式，即使正在输出跟踪数据，CPU 也与未使用跟踪功能时一样继续运行。已产生的各种跟踪事件，仅在 AUD 内部 FIFO 有空间时才可依次保存，并向外部输出。 FIFO 无空间时，忽略已产生的跟踪事件，不向外部输出。但通过 LOST 命令向外部通知产生漏掉跟踪事件。输出 LOST 命令时，总是在 LOST 命令输出的下一个周期 $\overline{\text{AUDSYNC}}$=L。

同时产生至少 2 个事件时，按照转移跟踪、窗口数据跟踪的顺序判断优先级，仅选择 1 个事件保存到 FIFO。此时，由 LOST 命令向外部通知有未能保存到 FIFO 的事件。

图 23.8 为输出的 2 个转移跟踪事件期间，丢失了某些跟踪信息 （至少 1 个）的例子。图 23.9 为 LOST 命令后成为 STDBY 的例子。

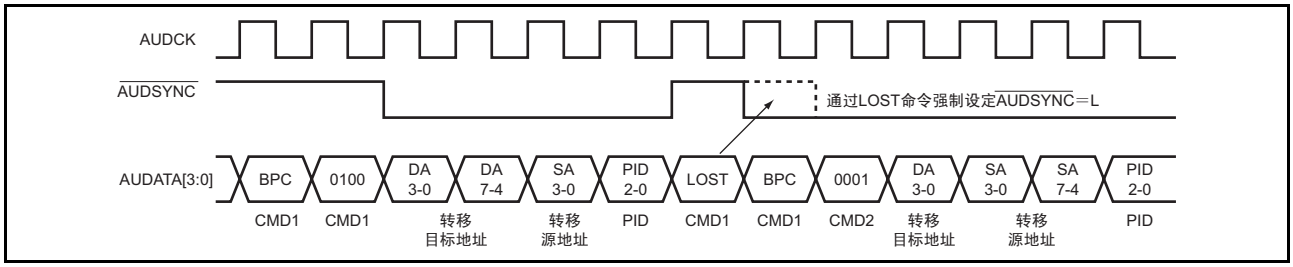


图 23.8　实时跟踪时跟踪数据丢失例子

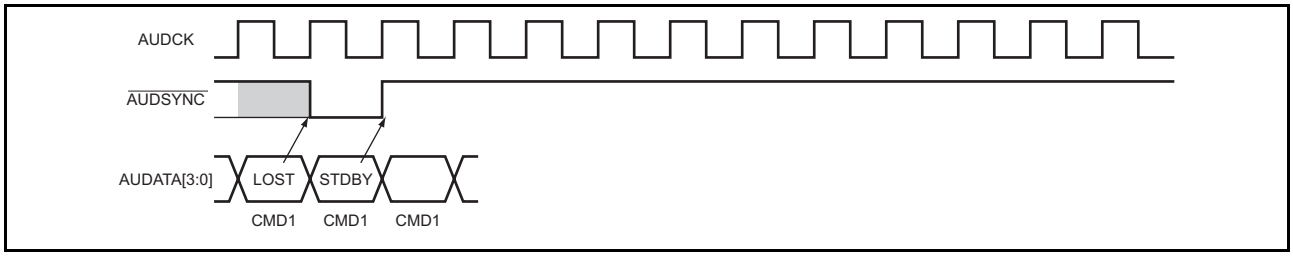


图 23.9　LOST 命令后成为 STDBY 的例子

(6)　全跟踪模式

在全跟踪模式，所有的跟踪数据均无丢失，全部向外部输出。 AUCSR 的 TM 位置 0，可选择全跟踪模式。在全跟踪模式产生的各种跟踪信息，经过 AUD 的内部 FIFO 向外部输出。 FIFO 已满时， CPU 在向外部输出 FIFO 内的跟踪数据前停止运行。

此时，不会像实时跟踪模式那样丢失跟踪数据，但在 FIFO 已满、 CPU 暂时停止运行时，输出表示 CPU 暂时停止运行的 LOST 命令。此时与实时跟踪模式时相同，在 LOST 命令输出的下一个周期总是为 $\overline{\text{AUDSYNC}}$=L。

(7)　地址比较与输出其地址的差异部位

为了有效利用 AUD 总线较窄的带宽，以内部寄存器的 PFBA 及 PFDA 中保存的上一次输出地址为基础，仅选择必要的低位部分作为跟踪信息地址部输出。

以 PFBA 值为基础，通过比较匹配转移跟踪时输出的转移目标、转移源两个地址的高位，确定这两个地址或软件跟踪的地址信息部分的长度 （4/8/16/32 位）。并且，以 PFDA 值为基础，通过比较匹配两个地址的高位，确定窗口数据跟踪输出的地址。每次输出比较对象的跟踪信息，都更新用于比较的 PFBA、 PFDA。

AUCSR 的 EN 位从 1 更新为 0、或输出计数器溢出时， PFBA、 PFDA 无效。此时，以 32 位长度输出跟踪信息的地址部。每次产生跟踪事件时，输出计数器递增计数，每输出 128 个跟踪数据， PFBA、 PFDA 无效。并且， AUCSR 的 EN 位从 1 更新为 0 时，输出计数器也清 0。输出计数器的功能为：恢复因某些原因被破坏的 AUD 输出跟踪信息。地址的比较匹配例子如图 23.10 所示。

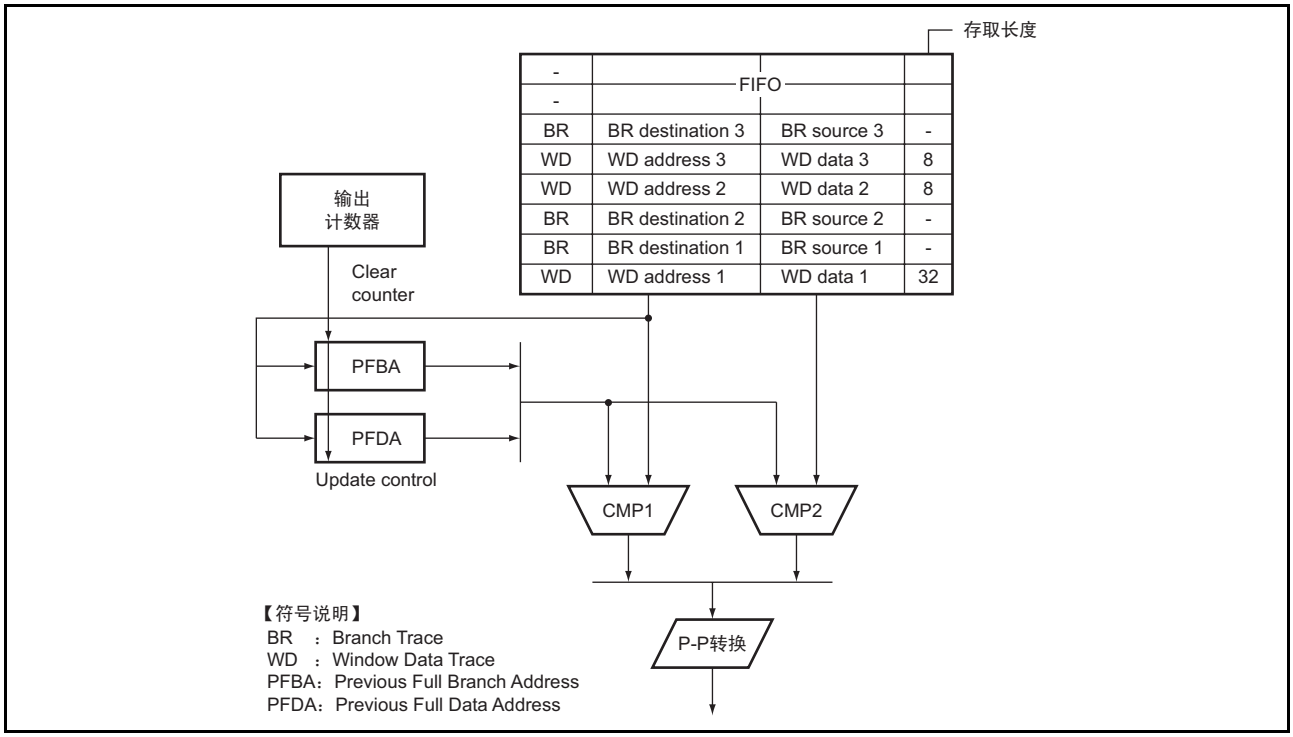


图 23.10　地址的比较匹配例子

1. 首先， WD address 1和PFDA通过CMP1比较匹配，根据比较结果，输出WD address 1的低位部分 （4/8/16/32位）。然后输出32位长度的WD data 1，之后， PFDA更新为WD address 1。
2. 其次， BR destination 1和PFBA通过CMP1、 BR source 1和PFBA通过CMP2进行比较。根据CMP1的比较结果，输出BR destination 1的低位部分 （4/8/16/32位），然后根据CMP2的比较结果，输出BR source 1的低位部分 （4/8/16/32位）。之后， PFBA更新为BR destination 1。
3. BR destination 2和PFBA通过CMP1、 BR source 2和PFBA通过CMP2进行比较。 PFBA保持上述2.所保存的值，即BR destination 1。根据CMP1的比较结果，输出BR destination 2的低位部分 （4/8/16/32位）。然后根据CMP2的比较结果输出BR source 2的低位部分 （4/8/16/32位）。之后， PFBA更新为BR destination 2。
4. WD address 2和PFDA通过CMP1进行比较。 PFDA保持上述1.所保存的值，即WD address 1。根据比较结果输出WD address 2的低位部分 （4/8/16/32位）。然后输出8位长度的WD data 1。之后， PFDA更新为WD address 2。
5. 输出计数器溢出时， PFBA、 PFDA的值无效。此时，跟踪信息中的转移目标、转移源、数据地址部总以32位长度输出。

23.3.9　使用时的注意事项 （转移跟踪模式）

(1)　转移跟踪模式初始化的相关规定

在以下条件时初始化本调试器内置的缓冲器及处理状态。

- 上电复位
- 手动复位
- 硬件待机
- 向$\overline{\text{AUDRST}}$引脚输入低电平
- 软件复位 （STBCR6的AUDSRST位清0时）
- AUCSR的EN位从0转换为1时
  清除AUD内部的FIFO
  PFBA、 PFDA值无效。 EN位的值转换为1后，以32位长度输出最初输出的各跟踪事件的地址部。

(2)　AUDCK 的规定

- AUCSR的EN位为1时，从AUDCK输出时钟； EN位为0时，不输出时钟。
- 待机时， AUDCK停止。在此期间，保持AUD输出的所有值及AUD的内部状态。从待机返回的同时重新开始已中断的处理。
- 必须在AUCSR的EN位为0的状态，更改AUCSR中的CLK0位和CLK1位。
- 设定AUDCK的频率时，不可超过20MHz。

(3)　写入 AUD 寄存器

- 经过I总线写入AUD寄存器的值。因此，在改写AUD寄存器后的周期，因条件改变可能不产生跟踪。为了获知AUD寄存器条件更改的时序，必须读取1次最后改写的寄存器。其后的跟踪对新写入的寄存器值有效。
- 必须在AUCSR的EN位为0的状态，更改除AUCSR之外的AUD寄存器的设定。

(4)　其他限制事项

- 通过位移量为0的BT、 BF指令也可输出转移信息。相应的指令为操作码、 H'8900、 H'8B00。
- AUD内的FIFO存在有效数据时，即使跟踪功能无效，也输出FIFO中剩余的数据，之后变为内部无效状态。输出该FIFO中剩余数据时，跟踪功能不可设置为有效。必须确认AUD不可输出数据后， AUD的跟踪功能才能设置为有效。
- AUD的FIFO与UBC的PC跟踪FIFO通用。因此，禁止同时使用AUD和UBC的PC跟踪。
- 由引脚功能控制器 （PFC）选择AUD引脚功能时，不可将AUD转换为模块待机。

23.4　RAM 监控模式

RAM 监控模式的功能为读取 / 写入连接内部 / 外部总线的所有模块，本功能可监控 / 协调 RAM。

23.4.1　通信协议

$\overline{\text{AUDSYNC}}$ 有效时， AUD 获得 AUDATA。必须用以下格式输入 AUDATA。

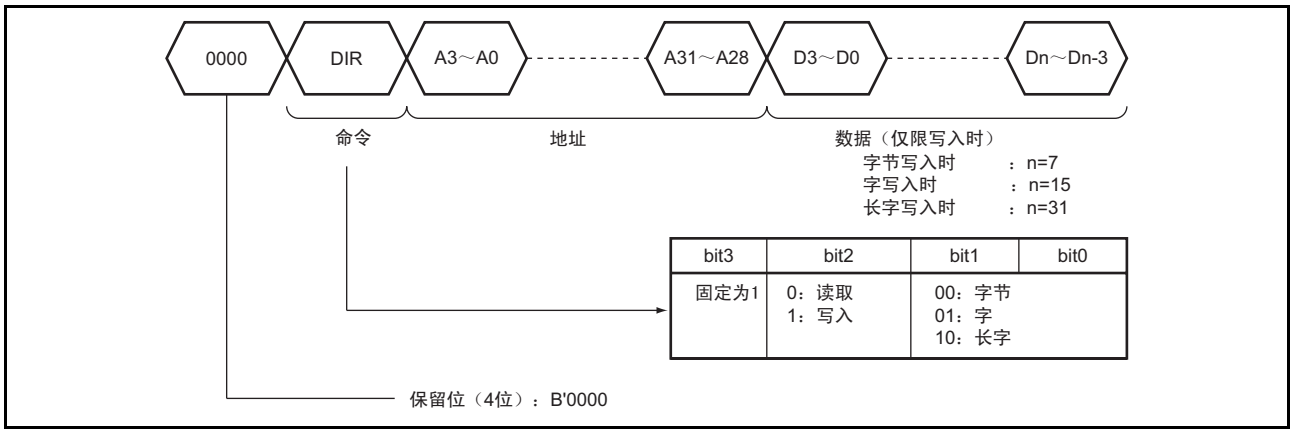


图 23.11　AUDATA 输入格式

23.4.2　运行说明

在 $\overline{\text{AUDRST}}$ 有效状态， AUDMD 设置为高电平，使 $\overline{\text{AUDRST}}$ 无效时，以 RAM 监控模式运行。读取运行例子如图 23.12、写入运行例子如图 23.13 所示。

$\overline{\text{AUDSYNC}}$ 有效时，从 AUDATA 开始输入。以图 23.11 所示格式输入命令、地址、数据 （仅写入时）时，开始读取 / 写入指定地址。内部执行过程中， AUD 返回 Not Ready （B'0000）。执行完成时，返回 Ready 标志 （B'0001）（图 23.12、图 23.13）。 Ready 标志的格式如表 23.7 所示。

读取时，检测出该标志后，使 $\overline{\text{AUDSYNC}}$ 无效时，输出指定长度的数据 （图 23.12）。向 DIR 输入除上述之外的命令时， AUD 将其作为命令错误，使处理无效，并将 Ready 标志内的 bit1 置 1。因 DIR 内指定命令的读取 / 写入运行导致总线错误时，使处理无效，并将 Ready 标志内 bit2 置 1 （图 23.14）。

总线错误条件如下：

1. 在地址4n+1、 4n+3执行字存取
2. 在地址4n+1、 4n+2、 4n+3执行长字存取
3. 在单芯片模式存取外部空间

表 23.7　Ready 标志格式

bit 3	bit2	bit1	bit0
固定为 0	0：正常状态 1：产生总线错误	0：正常状态 1：命令错误	0：Not Ready 1：Ready

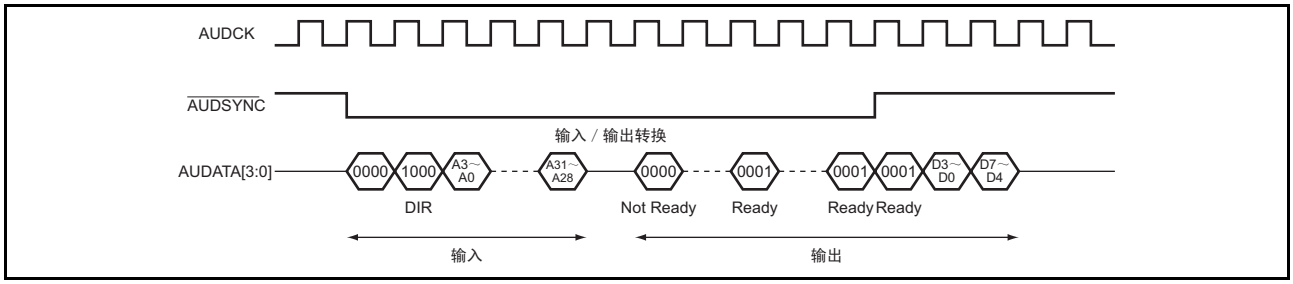


图 23.12　读取运行例子 （字节读取）

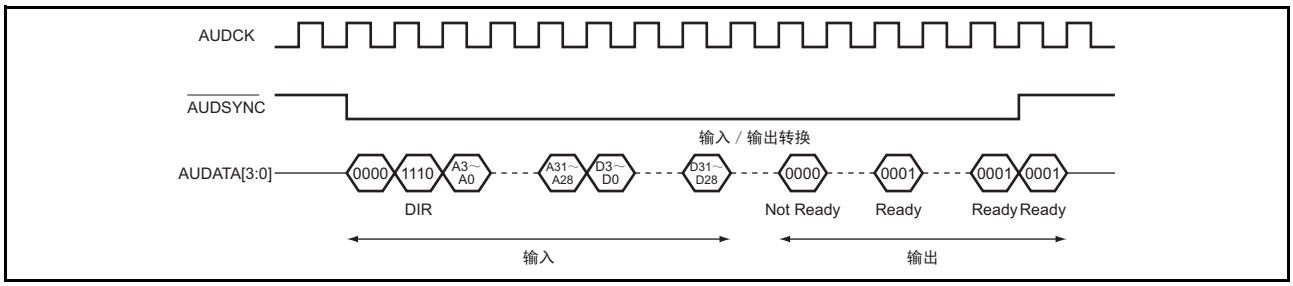


图 23.13　写入运行例子 （长字写入）

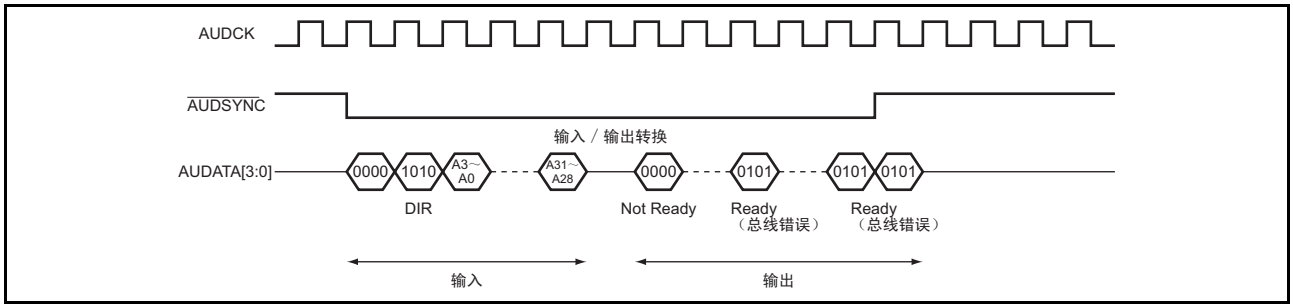


图 23.14　产生错误例子 （长字读取）

23.4.3　使用时的注意事项 （RAM 监控模式）

(1)　RAM 监控模式初始化的相关规定

在以下条件时初始化本 AUD 内置的缓冲器及处理状态。

- 上电复位
- 手动复位
- 硬件待机
- 向 $\overline{\text{AUDRST}}$ 引脚外加低电平
- 软件复位 （STBCR6 的 AUDSRST 位清 0 时）

(2)　AUDCK 的规定

AUDCK 为外部时钟输入，必须输入用于调试的时钟。
AUDCK 必须设置为小于等于 10MHz 及 1/4Pϕ。

(3)　其他限制事项

- 向 AUDATA 输入命令，并准备好必要的数据前，不可将 $\overline{\text{AUDSYNC}}$ 设置为有效。
- 使用 RAM 监控功能时，不可转换至软件待机。
- 由引脚功能控制器选择 AUD 引脚功能时，不可将 AUD 转换为模块待机。

第 24 章 寄存器一览表

寄存器一览表汇总了内部寄存器地址、位结构及各运行模式状态的相关信息，表示方法如下：

1. 寄存器地址一览表 （地址顺序）

- 按寄存器分配地址从小到大的顺序记载。
- 在寄存器名称栏用 “－” 表示保留地址。
  不得存取保留地址。
- 地址为16位或32位时，记载MSB侧的地址。
- 根据模块名称分类。
- 表示存取长度。

2. 位结构一览表

- 按照 “寄存器地址一览表 （地址顺序）” 的顺序记载位结构。
- 在位名称栏用 “－” 表示保留位。
- 位名称栏为空白，表示该寄存器全部分配至计数器和数据。
- 16位或32位寄存器时，从MSB侧的位开始记载。

3. 各运行模式的寄存器状态

- 按照 “寄存器地址一览表 （地址顺序）” 的顺序记录寄存器状态。
- 表示基本运行模式的寄存器状态，有内部模块固有的复位时，参阅内部模块的章节。

24.1　寄存器地址一览表（地址顺序）

存取长度表示位数。

存取状态数表示指定的基准时钟状态数，但 B 为 8 位存取时的值； W 为 16 位存取时的值； L 为 32 位存取时的值。

【注】 1. 禁止存取未定义 / 保留地址。不可保证存取此寄存器时的运行及后续运行，因此不得存取。

2. 存取 RCAN-ET 的邮箱区域，可能产生 0 ～ 5Pϕ 的等待期间。

寄存器名称	简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
串行模式寄存器 _0	SCSMR_0	8	H'FFFFC000	SCI（通道 0）	8	Pϕ 基准 B:2	16 位
位速率寄存器 _0	SCBRR_0	8	H'FFFFC002		8		
串行控制寄存器 _0	SCSCR_0	8	H'FFFFC004		8		
发送数据寄存器 _0	SCTDR_0	8	H'FFFFC006		8		
串行状态寄存器 _0	SCSSR_0	8	H'FFFFC008		8		
接收数据寄存器 _0	SCRDR_0	8	H'FFFFC00A		8		
串行方向控制寄存器 _0	SCSDCR_0	8	H'FFFFC00C		8		
串行端口寄存器 _0	SCSPTR_0	8	H'FFFFC00E		8		
串行模式寄存器 _1	SCSMR_1	8	H'FFFFC080	SCI（通道 1）	8	Pϕ 基准 B:2	16 位
位速率寄存器 _1	SCBRR_1	8	H'FFFFC082		8		
串行控制寄存器 _1	SCSCR_1	8	H'FFFFC084		8		
发送数据寄存器 _1	SCTDR_1	8	H'FFFFC086		8		
串行状态寄存器 _1	SCSSR_1	8	H'FFFFC088		8		
接收数据寄存器 _1	SCRDR_1	8	H'FFFFC08A		8		
串行方向控制寄存器 _1	SCSDCR_1	8	H'FFFFC08C		8		
串行端口寄存器 _1	SCSPTR_1	8	H'FFFFC08E		8		
串行模式寄存器 _2	SCSMR_2	8	H'FFFFC100	SCI（通道 2）	8	Pϕ 基准 B:2	16 位
位速率寄存器 _2	SCBRR_2	8	H'FFFFC102		8		
串行控制寄存器 _2	SCSCR_2	8	H'FFFFC104		8		
发送数据寄存器 _2	SCTDR_2	8	H'FFFFC106		8		
串行状态寄存器 _2	SCSSR_2	8	H'FFFFC108		8		
接收数据寄存器 _2	SCRDR_2	8	H'FFFFC10A		8		
串行方向控制寄存器 _2	SCSDCR_2	8	H'FFFFC10C		8		
串行端口寄存器 _2	SCSPTR_2	8	H'FFFFC10E		8		
定时器控制寄存器 _3	TCR_3	8	H'FFFFC200	MTU2	8、16、32	MPϕ 基准 B:2、W:2、L:4	16 位
定时器控制寄存器 _4	TCR_4	8	H'FFFFC201		8		
定时器模式寄存器 _3	TMDR_3	8	H'FFFFC202		8、16		
定时器模式寄存器 _4	TMDR_4	8	H'FFFFC203		8		

寄存器名称	简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
定时器 I/O 控制寄存器 H_3	TIORH_3	8	H'FFFFC204	MTU2	8、16、32	MPφ 基准 B:2、W:2、L:4	16 位
定时器 I/O 控制寄存器 L_3	TIORL_3	8	H'FFFFC205		8		
定时器 I/O 控制寄存器 H_4	TIORH_4	8	H'FFFFC206		8、16		
定时器 I/O 控制寄存器 L_4	TIORL_4	8	H'FFFFC207		8		
定时器中断允许寄存器 _3	TIER_3	8	H'FFFFC208		8、16		
定时器中断允许寄存器 _4	TIER_4	8	H'FFFFC209		8		
定时器输出主控允许寄存器	TOER	8	H'FFFFC20A		8		
定时器门控寄存器	TGCR	8	H'FFFFC20D		8		
定时器输出控制寄存器 1	TOCR1	8	H'FFFFC20E		8、16		
定时器输出控制寄存器 2	TOCR2	8	H'FFFFC20F		8		
定时器计数器 _3	TCNT_3	16	H'FFFFC210		16、32		
定时器计数器 _4	TCNT_4	16	H'FFFFC212		16		
定时器周期数据寄存器	TCDR	16	H'FFFFC214		16、32		
定时器空载时间数据寄存器	TDDR	16	H'FFFFC216		16		
定时器通用寄存器 A_3	TGRA_3	16	H'FFFFC218		16、32		
定时器通用寄存器 B_3	TGRB_3	16	H'FFFFC21A		16		
定时器通用寄存器 A_4	TGRA_4	16	H'FFFFC21C		16、32		
定时器通用寄存器 B_4	TGRB_4	16	H'FFFFC21E		16		
定时器副计数器	TCNTS	16	H'FFFFC220		16、32		
定时器周期缓冲寄存器	TCBR	16	H'FFFFC222		16		
定时器通用寄存器 C_3	TGRC_3	16	H'FFFFC224		16、32		
定时器通用寄存器 D_3	TGRD_3	16	H'FFFFC226		16		
定时器通用寄存器 C_4	TGRC_4	16	H'FFFFC228		16、32		
定时器通用寄存器 D_4	TGRD_4	16	H'FFFFC22A		16		
定时器状态寄存器 _3	TSR_3	8	H'FFFFC22C		8、16		
定时器状态寄存器 _4	TSR_4	8	H'FFFFC22D		8		
定时器中断跳过设定寄存器	TITCR	8	H'FFFFC230		8、16		
定时器中断跳过次数寄存器	TITCNT	8	H'FFFFC231		8		
定时器缓冲传送设定寄存器	TBTER	8	H'FFFFC232		8		
定时器空载时间允许寄存器	TDER	8	H'FFFFC234		8		
定时器输出电平缓冲寄存器	TOLBR	8	H'FFFFC236		8		
定时器缓冲运行传送模式寄存器 _3	TBTM_3	8	H'FFFFC238		8、16		
定时器缓冲运行传送模式寄存器 _4	TBTM_4	8	H'FFFFC239		8		

寄存器名称	简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
定时器 A/D 转换开始请求控制寄存器	TADCR	16	H'FFFFC240	MTU2	16	MPφ 基准 B:2、 W:2、 L:4	16 位
定时器 A/D 转换开始请求周期设定寄存器 A_4	TADCORA_4	16	H'FFFFC244		16、 32		
定时器 A/D 转换开始请求周期设定寄存器 B_4	TADCORB_4	16	H'FFFFC246		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'FFFFC248		16、 32		
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'FFFFC24A		16		
定时器波形控制寄存器	TWCR	8	H'FFFFC260		8		
定时器开始寄存器	TSTR	8	H'FFFFC280		8、 16		
定时器同步寄存器	TSYR	8	H'FFFFC281		8		
定时器计数器同步开始寄存器	TCSYSTR	8	H'FFFFC282		8		
定时器读取 / 写入允许寄存器	TRWER	8	H'FFFFC284		8		
定时器控制寄存器 _0	TCR_0	8	H'FFFFC300		8、 16、 32		
定时器模式寄存器 _0	TMDR_0	8	H'FFFFC301		8		
定时器 I/O 控制寄存器 H_0	TIORH_0	8	H'FFFFC302		8、 16		
定时器 I/O 控制寄存器 L_0	TIORL_0	8	H'FFFFC303		8		
定时器中断允许寄存器 _0	TIER_0	8	H'FFFFC304		8、 16、 32		
定时器状态寄存器 _0	TSR_0	8	H'FFFFC305		8		
定时器计数器 _0	TCNT_0	16	H'FFFFC306		16		
定时器通用寄存器 A_0	TGRA_0	16	H'FFFFC308		16、 32		
定时器通用寄存器 B_0	TGRB_0	16	H'FFFFC30A		16		
定时器通用寄存器 C_0	TGRC_0	16	H'FFFFC30C		16、 32		
定时器通用寄存器 D_0	TGRD_0	16	H'FFFFC30E		16		
定时器通用寄存器 E_0	TGRE_0	16	H'FFFFC320		16、 32		
定时器通用寄存器 F_0	TGRF_0	16	H'FFFFC322		16		
定时器中断允许寄存器 2_0	TIER2_0	8	H'FFFFC324		8、 16		
定时器状态寄存器 2_0	TSR2_0	8	H'FFFFC325		8		
定时器缓冲运行传送模式寄存器 _0	TBTM_0	8	H'FFFFC326		8		
定时器控制寄存器 _1	TCR_1	8	H'FFFFC380		8、 16		
定时器模式寄存器 _1	TMDR_1	8	H'FFFFC381		8		
定时器 I/O 控制寄存器 _1	TIOR_1	8	H'FFFFC382		8		
定时器中断允许寄存器 _1	TIER_1	8	H'FFFFC384		8、 16、 32		
定时器状态寄存器 _1	TSR_1	8	H'FFFFC385		8		
定时器计数器 _1	TCNT_1	16	H'FFFFC386		16		

寄存器名称	简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
定时器通用寄存器 A_1	TGRA_1	16	H'FFFFC388	MTU2	16、32	MPϕ 基准 B:2、W:2、L:4	16 位
定时器通用寄存器 B_1	TGRB_1	16	H'FFFFC38A		16		
定时器输入捕捉控制寄存器	TICCR	8	H'FFFFC390		8		
定时器控制寄存器 _2	TCR_2	8	H'FFFFC400		8、16		
定时器模式寄存器 _2	TMDR_2	8	H'FFFFC401		8		
定时器 I/O 控制寄存器 _2	TIOR_2	8	H'FFFFC402		8		
定时器中断允许寄存器 _2	TIER_2	8	H'FFFFC404		8、16、32		
定时器状态寄存器 _2	TSR_2	8	H'FFFFC405		8		
定时器计数器 _2	TCNT_2	16	H'FFFFC406		16		
定时器通用寄存器 A_2	TGRA_2	16	H'FFFFC408		16、32		
定时器通用寄存器 B_2	TGRB_2	16	H'FFFFC40A		16		
定时器控制寄存器 _3S	TCR_3S	8	H'FFFFC600	MTU2S	8、16、32	MIϕ 基准 B:2、W:2、L:4	16 位
定时器控制寄存器 _4S	TCR_4S	8	H'FFFFC601		8		
定时器模式寄存器 _3S	TMDR_3S	8	H'FFFFC602		8、16		
定时器模式寄存器 _4S	TMDR_4S	8	H'FFFFC603		8		
定时器 I/O 控制寄存器 H_3S	TIORH_3S	8	H'FFFFC604		8、16、32		
定时器 I/O 控制寄存器 L_3S	TIORL_3S	8	H'FFFFC605		8		
定时器 I/O 控制寄存器 H_4S	TIORH_4S	8	H'FFFFC606		8、16		
定时器 I/O 控制寄存器 L_4S	TIORL_4S	8	H'FFFFC607		8		
定时器中断允许寄存器 _3S	TIER_3S	8	H'FFFFC608		8、16		
定时器中断允许寄存器 _4S	TIER_4S	8	H'FFFFC609		8		
定时器输出主控允许寄存器 S	TOERS	8	H'FFFFC60A		8		
定时器门控寄存器 S	TGCRS	8	H'FFFFC60D		8		
定时器输出控制寄存器 1S	TOCR1S	8	H'FFFFC60E		8、16		
定时器输出控制寄存器 2S	TOCR2S	8	H'FFFFC60F		8		
定时器计数器 _3S	TCNT_3S	16	H'FFFFC610		16、32		
定时器计数器 _4S	TCNT_4S	16	H'FFFFC612		16		
定时器周期数据寄存器 S	TCDRS	16	H'FFFFC614		16、32		
定时器空载时间数据寄存器 S	TDDRS	16	H'FFFFC616		16		
定时器通用寄存器 A_3S	TGRA_3S	16	H'FFFFC618		16、32		
定时器通用寄存器 B_3S	TGRB_3S	16	H'FFFFC61A		16		
定时器通用寄存器 A_4S	TGRA_4S	16	H'FFFFC61C		16、32		
定时器通用寄存器 B_4S	TGRB_4S	16	H'FFFFC61E		16		

寄存器名称	简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
定时器副计数器 S	TCNTSS	16	H'FFFFC620	MTU2S	16、32	MIϕ 基准 B:2、W:2、L:4	16 位
定时器周期缓冲寄存器 S	TCBRS	16	H'FFFFC622		16		
定时器通用寄存器 C_3S	TGRC_3S	16	H'FFFFC624		16、32		
定时器通用寄存器 D_3S	TGRD_3S	16	H'FFFFC626		16		
定时器通用寄存器 C_4S	TGRC_4S	16	H'FFFFC628		16、32		
定时器通用寄存器 D_4S	TGRD_4S	16	H'FFFFC62A		16		
定时器状态寄存器 _3S	TSR_3S	8	H'FFFFC62C		8、16		
定时器状态寄存器 _4S	TSR_4S	8	H'FFFFC62D		8		
定时器中断跳过设定寄存器 S	TITCRS	8	H'FFFFC630		8、16		
定时器中断跳过次数寄存器 S	TITCNTS	8	H'FFFFC631		8		
定时器缓冲传送设定寄存器 S	TBTERS	8	H'FFFFC632		8		
定时器空载时间允许寄存器 S	TDERS	8	H'FFFFC634		8		
定时器输出电平缓冲寄存器 S	TOLBRS	8	H'FFFFC636		8		
定时器缓冲运行传送模式寄存器 _3S	TBTM_3S	8	H'FFFFC638		8、16		
定时器缓冲运行传送模式寄存器 _4S	TBTM_4S	8	H'FFFFC639		8		
定时器 A/D 转换开始请求控制寄存器 S	TADCRS	16	H'FFFFC640		16		
定时器 A/D 转换开始请求周期设定寄存器 A_4S	TADCORA_4S	16	H'FFFFC644		16、32		
定时器 A/D 转换开始请求周期设定寄存器 B_4S	TADCORB_4S	16	H'FFFFC646		16		
定时器 A/D 转换开始请求周期设定缓冲寄存器 A_4S	TADCOBRA_4S	16	H'FFFFC648		16、32		
定时器 A/D 转换开始请求周期设定缓冲寄存器 B_4S	TADCOBRB_4S	16	H'FFFFC64A		16		
定时器同步清除寄存器 S	TSYCRS	8	H'FFFFC650		8		
定时器波形控制寄存器 S	TWCRS	8	H'FFFFC660		8		
定时器开始寄存器 S	TSTRS	8	H'FFFFC680		8、16		
定时器同步寄存器 S	TSYRS	8	H'FFFFC681		8		
定时器读取 / 写入允许寄存器 S	TRWERS	8	H'FFFFC684		8		
定时器计数器 U_5S	TCNTU_5S	16	H'FFFFC880		16、32		
定时器通用寄存器 U_5S	TGRU_5S	16	H'FFFFC882		16		
定时器控制寄存器 U_5S	TCRU_5S	8	H'FFFFC884		8		
定时器 I/O 控制寄存器 U_5S	TIORU_5S	8	H'FFFFC886		8		
定时器计数器 V_5S	TCNTV_5S	16	H'FFFFC890		16、32		

寄存器名称	简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
定时器通用寄存器 V_5S	TGRV_5S	16	H'FFFFC892	MTU2S	16	MIϕ 基准 B:2、 W:2、 L:4	16 位
定时器控制寄存器 V_5S	TCRV_5S	8	H'FFFFC894		8		
定时器 I/O 控制寄存器 V_5S	TIORV_5S	8	H'FFFFC896		8		
定时器计数器 W_5S	TCNTW_5S	16	H'FFFFC8A0		16、 32		
定时器通用寄存器 W_5S	TGRW_5S	16	H'FFFFC8A2		16		
定时器控制寄存器 W_5S	TCRW_5S	8	H'FFFFC8A4		8		
定时器 I/O 控制寄存器 W_5S	TIORW_5S	8	H'FFFFC8A6		8		
定时器状态寄存器 _5S	TSR_5S	8	H'FFFFC8B0		8		
定时器中断允许寄存器 _5S	TIER_5S	8	H'FFFFC8B2		8		
定时器开始寄存器 _5S	TSTR_5S	8	H'FFFFC8B4		8		
定时器比较匹配清除寄存器 S	TCNTCMPCLRS	8	H'FFFFC8B6		8		
闪存代码控制 / 状态寄存器	FCCS	8	H'FFFFCC00	FLASH	8	Pϕ 基准 B:5	16 位
闪存编程代码选择寄存器	FPCS	8	H'FFFFCC01		8		
闪存擦除代码选择寄存器	FECS	8	H'FFFFCC02		8		
闪存键代码寄存器	FKEY	8	H'FFFFCC04		8		
闪存 MAT 选择寄存器	FMATS	8	H'FFFFCC05		8		
闪存传送目标地址寄存器	FTDAR	8	H'FFFFCC06		8		
DTC 允许寄存器 A	DTCERA	16	H'FFFFCC80	DTC	8、 16	Pϕ 基准 B:2、 W:2、 L:4	16 位
DTC 允许寄存器 B	DTCERB	16	H'FFFFCC82		8、 16		
DTC 允许寄存器 C	DTCERC	16	H'FFFFCC84		8、 16		
DTC 允许寄存器 D	DTCERD	16	H'FFFFCC86		8、 16		
DTC 允许寄存器 E	DTCERE	16	H'FFFFCC88		8、 16		
DTC 控制寄存器	DTCCR	8	H'FFFFCC90		8		
DTC 向量基址寄存器	DTCVBR	32	H'FFFFCC94		8、 16、 32		
SS 控制寄存器 H	SSCRH	8	H'FFFFCD00	SSU	8、 16	Pϕ 基准 B:2、 W:2	16 位
SS 控制寄存器 L	SSCRL	8	H'FFFFCD01		8		
SS 模式寄存器	SSMR	8	H'FFFFCD02		8、 16		
SS 允许寄存器	SSER	8	H'FFFFCD03		8		
SS 状态寄存器	SSSR	8	H'FFFFCD04		8、 16		
SS 控制寄存器 2	SSCR2	8	H'FFFFCD05		8		
SS 发送数据寄存器 0	SSTDR0	8	H'FFFFCD06		8、 16		

寄存器名称	简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
SS 发送数据寄存器 1	SSTDR1	8	H'FFFFCD07	SSU	8	Pϕ 基准 B:2、 W:2	16 位
SS 发送数据寄存器 2	SSTDR2	8	H'FFFFCD08		8、 16		
SS 发送数据寄存器 3	SSTDR3	8	H'FFFFCD09		8		
SS 接收数据寄存器 0	SSRDR0	8	H'FFFFCD0A		8、 16		
SS 接收数据寄存器 1	SSRDR1	8	H'FFFFCD0B		8		
SS 接收数据寄存器 2	SSRDR2	8	H'FFFFCD0C		8、 16		
SS 接收数据寄存器 3	SSRDR3	8	H'FFFFCD0D		8		
比较匹配定时器开始寄存器	CMSTR	16	H'FFFFCE00	CMT	8、 16、 32	Pϕ 基准 B:2、 W:2、 L:4	16 位
比较匹配定时器控制 / 状态寄存器 _0	CMCSR_0	16	H'FFFFCE02		8、 16		
比较匹配计数器 _0	CMCNT_0	16	H'FFFFCE04		8、 16、 32		
比较匹配常数寄存器 _0	CMCOR_0	16	H'FFFFCE06		8、 16		
比较匹配定时器控制 / 状态寄存器 _1	CMCSR_1	16	H'FFFFCE08		8、 16、 32		
比较匹配计数器 _1	CMCNT_1	16	H'FFFFCE0A		8、 16		
比较匹配常数寄存器 _1	CMCOR_1	16	H'FFFFCE0C		8、 16、 32		
输入电平控制 / 状态寄存器 1	ICSR1	16	H'FFFFD000	POE	8、 16、 32	Pϕ 基准 B:2、 W:2、 L:4	16 位
输出电平控制 / 状态寄存器 1	OCSR1	16	H'FFFFD002		8、 16		
输入电平控制 / 状态寄存器 2	ICSR2	16	H'FFFFD004		8、 16、 32		
输出电平控制 / 状态寄存器 2	OCSR2	16	H'FFFFD006		8、 16		
输入电平控制 / 状态寄存器 3	ICSR3	16	H'FFFFD008		8、 16		
软件端口输出允许寄存器	SPOER	8	H'FFFFD00A		8		
端口输出允许控制寄存器 1	POECR1	8	H'FFFFD00B		8		
端口输出允许控制寄存器 2	POECR2	16	H'FFFFD00C		8、 16		
端口 A 数据寄存器 L	PADRL	16	H'FFFFD102	I/O	8、 16	Pϕ 基准 B:2、 W:2、 L:4	16 位
端口 A/IO 寄存器 L	PAIORL	16	H'FFFFD106	PFC	8、 16		
端口 A 控制寄存器 L4	PACRL4	16	H'FFFFD110		8、 16、 32		
端口 A 控制寄存器 L3	PACRL3	16	H'FFFFD112		8、 16		
端口 A 控制寄存器 L2	PACRL2	16	H'FFFFD114		8、 16、 32		
端口 A 控制寄存器 L1	PACRL1	16	H'FFFFD116		8、 16		
端口 A 端口寄存器 L	PAPRL	16	H'FFFFD11E	I/O	8、 16		
端口 B 数据寄存器 L	PBDRL	16	H'FFFFD182		8、 16		

寄存器名称	简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
端口 B/IO 寄存器 L	PBIORL	16	H'FFFFD186	PFC	8、16	Pϕ 基准 B:2、W:2、L:4	16 位
端口 B 控制寄存器 L2	PBCRL2	16	H'FFFFD194		8、16、32		
端口 B 控制寄存器 L1	PBCRL1	16	H'FFFFD196		8、16		
端口 B 端口寄存器 L	PBPRL	16	H'FFFFD19E	I/O	8、16		
端口 D 数据寄存器 L	PDDRL	16	H'FFFFD282		8、16		
端口 D/IO 寄存器 L	PDIORL	16	H'FFFFD286	PFC	8、16		
端口 D 控制寄存器 L3	PDCRL3	16	H'FFFFD292		8、16		
端口 D 控制寄存器 L2	PDCRL2	16	H'FFFFD294		8、16、32		
端口 D 控制寄存器 L1	PDCRL1	16	H'FFFFD296		8、16		
端口 D 端口寄存器 L	PDPRL	16	H'FFFFD29E	I/O	8、16		
端口 E 数据寄存器 H	PEDRH	16	H'FFFFD300		8、16、32		
端口 E 数据寄存器 L	PEDRL	16	H'FFFFD302		8、16		
端口 E/IO 寄存器 H	PEIORH	16	H'FFFFD304	PFC	8、16、32		
端口 E/IO 寄存器 L	PEIORL	16	H'FFFFD306		8、16		
端口 E 控制寄存器 H2	PECRH2	16	H'FFFFD30C		8、16、32		
端口 E 控制寄存器 H1	PECRH1	16	H'FFFFD30E		8、16		
端口 E 控制寄存器 L4	PECRL4	16	H'FFFFD310		8、16、32		
端口 E 控制寄存器 L3	PECRL3	16	H'FFFFD312		8、16		
端口 E 控制寄存器 L2	PECRL2	16	H'FFFFD314		8、16、32		
端口 E 控制寄存器 L1	PECRL1	16	H'FFFFD316		8、16		
端口 E 端口寄存器 H	PEPRH	16	H'FFFFD31C	I/O	8、16、32		
端口 E 端口寄存器 L	PEPRL	16	H'FFFFD31E		8、16		
A/D 控制寄存器 _0	ADCR_0	8	H'FFFFD400	A/D（通道 0）	8	Pϕ 基准 B:2、W:2	16 位
A/D 状态寄存器 _0	ADSR_0	8	H'FFFFD402		8		
A/D 开始触发选择寄存器 _0	ADSTRGR_0	8	H'FFFFD41C		8		
A/D 模拟输入通道选择寄存器 _0	ADANSR_0	8	H'FFFFD420		8		
A/D 数据寄存器 0	ADDR0	16	H'FFFFD440		16		
A/D 数据寄存器 1	ADDR1	16	H'FFFFD442		16		
A/D 数据寄存器 2	ADDR2	16	H'FFFFD444		16		
A/D 数据寄存器 3	ADDR3	16	H'FFFFD446		16		
A/D 数据寄存器 4	ADDR4	16	H'FFFFD448		16		
A/D 数据寄存器 5	ADDR5	16	H'FFFFD44A		16		
A/D 数据寄存器 6	ADDR6	16	H'FFFFD44C		16		
A/D 数据寄存器 7	ADDR7	16	H'FFFFD44E		16		

寄存器名称	简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
A/D 控制寄存器 _1	ADCR_1	8	H'FFFFD600	A/D（通道 1）	8	Pϕ 基准 B:2、 W:2	16 位
A/D 状态寄存器 _1	ADSR_1	8	H'FFFFD602		8		
A/D 开始触发选择寄存器 _1	ADSTRGR_1	8	H'FFFFD61C		8		
A/D 模拟输入通道选择寄存器 _1	ADANSR_1	8	H'FFFFD620		8		
A/D 数据寄存器 8	ADDR8	16	H'FFFFD640		16		
A/D 数据寄存器 9	ADDR9	16	H'FFFFD642		16		
A/D 数据寄存器 10	ADDR10	16	H'FFFFD644		16		
A/D 数据寄存器 11	ADDR11	16	H'FFFFD646		16		
A/D 数据寄存器 12	ADDR12	16	H'FFFFD648		16		
A/D 数据寄存器 13	ADDR13	16	H'FFFFD64A		16		
A/D 数据寄存器 14	ADDR14	16	H'FFFFD64C		16		
A/D 数据寄存器 15	ADDR15	16	H'FFFFD64E		16		
主控制寄存器 _0	MCR_0	16	H'FFFFD800	RCAN-ET（通道 0）	16	Pϕ 基准 B:2、 W:2、 L:4	16 位
通用状态寄存器 _0	GSR_0	16	H'FFFFD802		16		
位结构寄存器 1_0	BCR1_0	16	H'FFFFD804		16		
位结构寄存器 0_0	BCR0_0	16	H'FFFFD806		16		
中断请求寄存器 _0	IRR_0	16	H'FFFFD808		16		
中断屏蔽寄存器 _0	IMR_0	16	H'FFFFD80A		16		
发送错误计数器 / 接收错误计数器 _0	TEC/REC_0	16	H'FFFFD80C		16		
发送等待寄存器 1_0、 0_0	TXPR1_0、 0_0	32	H'FFFFD820		32		
发送取消寄存器 0_0	TXCR0_0	16	H'FFFFD82A		16		
发送应答寄存器 0_0	TXACK0_0	16	H'FFFFD832		16		
中止应答寄存器 0_0	ABACK0_0	16	H'FFFFD83A		16		
数据帧接收完成寄存器 0_0	RXPR0_0	16	H'FFFFD842		16		
远程帧接收完成寄存器 0_0	RFPR0_0	16	H'FFFFD84A		16		
邮箱中断屏蔽寄存器 0_0	MBIMR0_0	16	H'FFFFD852		16		
未读信息状态寄存器 0_0	UMSR0_0	16	H'FFFFD85A		16		
MB[0]. CONTROL0H	—	16	H'FFFFD900		16、 32		
CONTROL0L	—	16	H'FFFFD902		16		
LAFMH	—	16	H'FFFFD904		16、 32		
LAFML	—	16	H'FFFFD906		16		
MSG_DATA[0]	—	8	H'FFFFD908		8、 16、 32		
MSG_DATA[1]	—	8	H'FFFFD909		8		
MSG_DATA[2]	—	8	H'FFFFD90A		8、 16		

寄存器名称		简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
MB[0].	MSG_DATA[3]	—	8	H'FFFFD90B	RCAN-ET（通道 0）	8	Pϕ 基准 B:2、 W:2、 L:4	16 位
	MSG_DATA[4]	—	8	H'FFFFD90C		8、 16、 32		
	MSG_DATA[5]	—	8	H'FFFFD90D		8		
	MSG_DATA[6]	—	8	H'FFFFD90E		8、 16		
	MSG_DATA[7]	—	8	H'FFFFD90F		8		
	CONTROL1H	—	8	H'FFFFD910		8、 16		
	CONTROL1L	—	8	H'FFFFD911		8		
MB[1].	CONTROL0H	—	16	H'FFFFD920		16、 32		
	CONTROL0L	—	16	H'FFFFD922		16		
	LAFMH	—	16	H'FFFFD924		16、 32		
	LAFML	—	16	H'FFFFD926		16		
	MSG_DATA[0]	—	8	H'FFFFD928		8、 16、 32		
	MSG_DATA[1]	—	8	H'FFFFD929		8		
	MSG_DATA[2]	—	8	H'FFFFD92A		8、 16		
	MSG_DATA[3]	—	8	H'FFFFD92B		8		
	MSG_DATA[4]	—	8	H'FFFFD92C		8、 16、 32		
	MSG_DATA[5]	—	8	H'FFFFD92D		8		
	MSG_DATA[6]	—	8	H'FFFFD92E		8、 16		
	MSG_DATA[7]	—	8	H'FFFFD92F		8		
	CONTROL1H	—	8	H'FFFFD930		8、 16		
	CONTROL1L	—	8	H'FFFFD931		8		
MB[2].	CONTROL0H	—	16	H'FFFFD940		16、 32		
	CONTROL0L	—	16	H'FFFFD942		16		
	LAFMH	—	16	H'FFFFD944		16、 32		
	LAFML	—	16	H'FFFFD946		16		
	MSG_DATA[0]	—	8	H'FFFFD948		8、 16、 32		
	MSG_DATA[1]	—	8	H'FFFFD949		8		
	MSG_DATA[2]	—	8	H'FFFFD94A		8、 16		
	MSG_DATA[3]	—	8	H'FFFFD94B		8		
	MSG_DATA[4]	—	8	H'FFFFD94C		8、 16、 32		
	MSG_DATA[5]	—	8	H'FFFFD94D		8		
	MSG_DATA[6]	—	8	H'FFFFD94E		8、 16		
	MSG_DATA[7]	—	8	H'FFFFD94F		8		
	CONTROL1H	—	8	H'FFFFD950		8、 16		
	CONTROL1L	—	8	H'FFFFD951		8		

寄存器名称		简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
MB[3].	CONTROL0H	—	16	H'FFFFD960	RCAN-ET（通道 0）	16、32	Pϕ 基准 B:2、W:2、L:4	16 位
	CONTROL0L	—	16	H'FFFFD962		16		
	LAFMH	—	16	H'FFFFD964		16、32		
	LAFML	—	16	H'FFFFD966		16		
	MSG_DATA[0]	—	8	H'FFFFD968		8、16、32		
	MSG_DATA[1]	—	8	H'FFFFD969		8		
	MSG_DATA[2]	—	8	H'FFFFD96A		8、16		
	MSG_DATA[3]	—	8	H'FFFFD96B		8		
	MSG_DATA[4]	—	8	H'FFFFD96C		8、16、32		
	MSG_DATA[5]	—	8	H'FFFFD96D		8		
	MSG_DATA[6]	—	8	H'FFFFD96E		8、16		
	MSG_DATA[7]	—	8	H'FFFFD96F		8		
	CONTROL1H	—	8	H'FFFFD970		8、16		
	CONTROL1L	—	8	H'FFFFD971		8		
MB[4].	CONTROL0H	—	16	H'FFFFD980		16、32		
	CONTROL0L	—	16	H'FFFFD982		16		
	LAFMH	—	16	H'FFFFD984		16、32		
	LAFML	—	16	H'FFFFD986		16		
	MSG_DATA[0]	—	8	H'FFFFD988		8、16、32		
	MSG_DATA[1]	—	8	H'FFFFD989		8		
	MSG_DATA[2]	—	8	H'FFFFD98A		8、16		
	MSG_DATA[3]	—	8	H'FFFFD98B		8		
	MSG_DATA[4]	—	8	H'FFFFD98C		8、16、32		
	MSG_DATA[5]	—	8	H'FFFFD98D		8		
	MSG_DATA[6]	—	8	H'FFFFD98E		8、16		
	MSG_DATA[7]	—	8	H'FFFFD98F		8		
	CONTROL1H	—	8	H'FFFFD990		8、16		
	CONTROL1L	—	8	H'FFFFD991		8		
MB[5].	CONTROL0H	—	16	H'FFFFD9A0		16、32		
	CONTROL0L	—	16	H'FFFFD9A2		16		
	LAFMH	—	16	H'FFFFD9A4		16、32		
	LAFML	—	16	H'FFFFD9A6		16		
	MSG_DATA[0]	—	8	H'FFFFD9A8		8、16、32		
	MSG_DATA[1]	—	8	H'FFFFD9A9		8		
	MSG_DATA[2]	—	8	H'FFFFD9AA		8、16		

寄存器名称		简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
MB[5].	MSG_DATA[3]	—	8	H'FFFFD9AB	RCAN-ET（通道 0）	8	Pϕ 基准 B:2、 W:2、 L:4	16 位
	MSG_DATA[4]	—	8	H'FFFFD9AC		8、 16、 32		
	MSG_DATA[5]	—	8	H'FFFFD9AD		8		
	MSG_DATA[6]	—	8	H'FFFFD9AE		8、 16		
	MSG_DATA[7]	—	8	H'FFFFD9AF		8		
	CONTROL1H	—	8	H'FFFFD9B0		8、 16		
	CONTROL1L	—	8	H'FFFFD9B1		8		
MB[6].	CONTROL0H	—	16	H'FFFFD9C0		16、 32		
	CONTROL0L	—	16	H'FFFFD9C2		16		
	LAFMH	—	16	H'FFFFD9C4		16、 32		
	LAFML	—	16	H'FFFFD9C6		16		
	MSG_DATA[0]	—	8	H'FFFFD9C8		8、 16、 32		
	MSG_DATA[1]	—	8	H'FFFFD9C9		8		
	MSG_DATA[2]	—	8	H'FFFFD9CA		8、 16		
	MSG_DATA[3]	—	8	H'FFFFD9CB		8		
	MSG_DATA[4]	—	8	H'FFFFD9CC		8、 16、 32		
	MSG_DATA[5]	—	8	H'FFFFD9CD		8		
	MSG_DATA[6]	—	8	H'FFFFD9CE		8、 16		
	MSG_DATA[7]	—	8	H'FFFFD9CF		8		
	CONTROL1H	—	8	H'FFFFD9D0		8、 16		
	CONTROL1L	—	8	H'FFFFD9D1		8		
MB[7].	CONTROL0H	—	16	H'FFFFD9E0		16、 32		
	CONTROL0L	—	16	H'FFFFD9E2		16		
	LAFMH	—	16	H'FFFFD9E4		16、 32		
	LAFML	—	16	H'FFFFD9E6		16		
	MSG_DATA[0]	—	8	H'FFFFD9E8		8、 16、 32		
	MSG_DATA[1]	—	8	H'FFFFD9E9		8		
	MSG_DATA[2]	—	8	H'FFFFD9EA		8、 16		
	MSG_DATA[3]	—	8	H'FFFFD9EB		8		
	MSG_DATA[4]	—	8	H'FFFFD9EC		8、 16、 32		
	MSG_DATA[5]	—	8	H'FFFFD9ED		8		
	MSG_DATA[6]	—	8	H'FFFFD9EE		8、 16		
	MSG_DATA[7]	—	8	H'FFFFD9EF		8		
	CONTROL1H	—	8	H'FFFFD9F0		8、 16		
	CONTROL1L	—	8	H'FFFFD9F1		8		

寄存器名称		简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
MB[8].	CONTROL0H	—	16	H'FFFFDA00	RCAN-ET（通道 0）	16、32	Pϕ 基准 B:2、W:2、L:4	16 位
	CONTROL0L	—	16	H'FFFFDA02		16		
	LAFMH	—	16	H'FFFFDA04		16、32		
	LAFML	—	16	H'FFFFDA06		16		
	MSG_DATA[0]	—	8	H'FFFFDA08		8、16、32		
	MSG_DATA[1]	—	8	H'FFFFDA09		8		
	MSG_DATA[2]	—	8	H'FFFFDA0A		8、16		
	MSG_DATA[3]	—	8	H'FFFFDA0B		8		
	MSG_DATA[4]	—	8	H'FFFFDA0C		8、16、32		
	MSG_DATA[5]	—	8	H'FFFFDA0D		8		
	MSG_DATA[6]	—	8	H'FFFFDA0E		8、16		
	MSG_DATA[7]	—	8	H'FFFFDA0F		8		
	CONTROL1H	—	8	H'FFFFDA10		8、16		
	CONTROL1L	—	8	H'FFFFDA11		8		
MB[9].	CONTROL0H	—	16	H'FFFFDA20		16、32		
	CONTROL0L	—	16	H'FFFFDA22		16		
	LAFMH	—	16	H'FFFFDA24		16、32		
	LAFML	—	16	H'FFFFDA26		16		
	MSG_DATA[0]	—	8	H'FFFFDA28		8、16、32		
	MSG_DATA[1]	—	8	H'FFFFDA29		8		
	MSG_DATA[2]	—	8	H'FFFFDA2A		8、16		
	MSG_DATA[3]	—	8	H'FFFFDA2B		8		
	MSG_DATA[4]	—	8	H'FFFFDA2C		8、16、32		
	MSG_DATA[5]	—	8	H'FFFFDA2D		8		
	MSG_DATA[6]	—	8	H'FFFFDA2E		8、16		
	MSG_DATA[7]	—	8	H'FFFFDA2F		8		
	CONTROL1H	—	8	H'FFFFDA30		8、16		
	CONTROL1L	—	8	H'FFFFDA31		8		
MB[10].	CONTROL0H	—	16	H'FFFFDA40		16、32		
	CONTROL0L	—	16	H'FFFFDA42		16		
	LAFMH	—	16	H'FFFFDA44		16、32		
	LAFML	—	16	H'FFFFDA46		16		
	MSG_DATA[0]	—	8	H'FFFFDA48		8、16、32		
	MSG_DATA[1]	—	8	H'FFFFDA49		8		
	MSG_DATA[2]	—	8	H'FFFFDA4A		8、16		

寄存器名称		简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
MB[10].	MSG_DATA[3]	—	8	H'FFFFDA4B	RCAN-ET（通道 0）	8	Pϕ 基准 B:2、 W:2、 L:4	16 位
	MSG_DATA[4]	—	8	H'FFFFDA4C		8、 16、 32		
	MSG_DATA[5]	—	8	H'FFFFDA4D		8		
	MSG_DATA[6]	—	8	H'FFFFDA4E		8、 16		
	MSG_DATA[7]	—	8	H'FFFFDA4F		8		
	CONTROL1H	—	8	H'FFFFDA50		8、 16		
	CONTROL1L	—	8	H'FFFFDA51		8		
MB[11].	CONTROL0H	—	16	H'FFFFDA60		16、 32		
	CONTROL0L	—	16	H'FFFFDA62		16		
	LAFMH	—	16	H'FFFFDA64		16、 32		
	LAFML	—	16	H'FFFFDA66		16		
	MSG_DATA[0]	—	8	H'FFFFDA68		8、 16、 32		
	MSG_DATA[1]	—	8	H'FFFFDA69		8		
	MSG_DATA[2]	—	8	H'FFFFDA6A		8、 16		
	MSG_DATA[3]	—	8	H'FFFFDA6B		8		
	MSG_DATA[4]	—	8	H'FFFFDA6C		8、 16、 32		
	MSG_DATA[5]	—	8	H'FFFFDA6D		8		
	MSG_DATA[6]	—	8	H'FFFFDA6E		8、 16		
	MSG_DATA[7]	—	8	H'FFFFDA6F		8		
	CONTROL1H	—	8	H'FFFFDA70		8、 16		
	CONTROL1L	—	8	H'FFFFDA71		8		
MB[12].	CONTROL0H	—	16	H'FFFFDA80		16、 32		
	CONTROL0L	—	16	H'FFFFDA82		16		
	LAFMH	—	16	H'FFFFDA84		16、 32		
	LAFML	—	16	H'FFFFDA86		16		
	MSG_DATA[0]	—	8	H'FFFFDA88		8、 16、 32		
	MSG_DATA[1]	—	8	H'FFFFDA89		8		
	MSG_DATA[2]	—	8	H'FFFFDA8A		8、 16		
	MSG_DATA[3]	—	8	H'FFFFDA8B		8		
	MSG_DATA[4]	—	8	H'FFFFDA8C		8、 16、 32		
	MSG_DATA[5]	—	8	H'FFFFDA8D		8		
	MSG_DATA[6]	—	8	H'FFFFDA8E		8、 16		
	MSG_DATA[7]	—	8	H'FFFFDA8F		8		
	CONTROL1H	—	8	H'FFFFDA90		8、 16		
	CONTROL1L	—	8	H'FFFFDA91		8		

寄存器名称		简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
MB[13].	CONTROL0H	—	16	H'FFFFDAA0	RCAN-ET（通道 0）	16、32	Pϕ 基准 B:2、W:2、L:4	16 位
	CONTROL0L	—	16	H'FFFFDAA2		16		
	LAFMH	—	16	H'FFFFDAA4		16、32		
	LAFML	—	16	H'FFFFDAA6		16		
	MSG_DATA[0]	—	8	H'FFFFDAA8		8、16、32		
	MSG_DATA[1]	—	8	H'FFFFDAA9		8		
	MSG_DATA[2]	—	8	H'FFFFDAAA		8、16		
	MSG_DATA[3]	—	8	H'FFFFDAAB		8		
	MSG_DATA[4]	—	8	H'FFFFDAAC		8、16、32		
	MSG_DATA[5]	—	8	H'FFFFDAAD		8		
	MSG_DATA[6]	—	8	H'FFFFDAAE		8、16		
	MSG_DATA[7]	—	8	H'FFFFDAAF		8		
	CONTROL1H	—	8	H'FFFFDAB0		8、16		
	CONTROL1L	—	8	H'FFFFDAB1		8		
MB[14].	CONTROL0H	—	16	H'FFFFDAC0		16、32		
	CONTROL0L	—	16	H'FFFFDAC2		16		
	LAFMH	—	16	H'FFFFDAC4		16、32		
	LAFML	—	16	H'FFFFDAC6		16		
	MSG_DATA[0]	—	8	H'FFFFDAC8		8、16、32		
	MSG_DATA[1]	—	8	H'FFFFDAC9		8		
	MSG_DATA[2]	—	8	H'FFFFDACA		8、16		
	MSG_DATA[3]	—	8	H'FFFFDACB		8		
	MSG_DATA[4]	—	8	H'FFFFDACC		8、16、32		
	MSG_DATA[5]	—	8	H'FFFFDACD		8		
	MSG_DATA[6]	—	8	H'FFFFDACE		8、16		
	MSG_DATA[7]	—	8	H'FFFFDACF		8		
	CONTROL1H	—	8	H'FFFFDAD0		8、16		
	CONTROL1L	—	8	H'FFFFDAD1		8		
MB[15].	CONTROL0H	—	16	H'FFFFDAE0		16、32		
	CONTROL0L	—	16	H'FFFFDAE2		16		
	LAFMH	—	16	H'FFFFDAE4		16、32		
	LAFML	—	16	H'FFFFDAE6		16		
	MSG_DATA[0]	—	8	H'FFFFDAE8		8、16、32		
	MSG_DATA[1]	—	8	H'FFFFDAE9		8		
	MSG_DATA[2]	—	8	H'FFFFDAEA		8、16		

寄存器名称		简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
MB[15].	MSG_DATA[3]	—	8	H'FFFFDAEB	RCAN-ET（通道 0）	8	Pϕ 基准 B:2、 W:2、 L:4	16 位
	MSG_DATA[4]	—	8	H'FFFFDAEC		8、 16、 32		
	MSG_DATA[5]	—	8	H'FFFFDAED		8		
	MSG_DATA[6]	—	8	H'FFFFDAEE		8、 16		
	MSG_DATA[7]	—	8	H'FFFFDAEF		8		
	CONTROL1H	—	8	H'FFFFDAF0		8、 16		
	CONTROL1L	—	8	H'FFFFDAF1		8		
主控制寄存器 _1		MCR_1	16	H'FFFFE000	RCAN-ET（通道 1）*1	16	Pϕ 基准 B:2、 W:2、 L:4	16 位
通用状态寄存器 _1		GSR_1	16	H'FFFFE002		16		
位结构寄存器 1_1		BCR1_1	16	H'FFFFE004		16		
位结构寄存器 0_1		BCR0_1	16	H'FFFFE006		16		
中断请求寄存器 _1		IRR_1	16	H'FFFFE008		16		
中断屏蔽寄存器 _1		IMR_1	16	H'FFFFE00A		16		
发送错误计数器 / 接收错误计数器 _1		TEC/REC_1	16	H'FFFFE00C		16		
发送等待寄存器 1_1、 0_1		TXPR1_1、 0_1	32	H'FFFFE020		32		
发送取消寄存器 0_1		TXCR0_1	16	H'FFFFE02A		16		
发送应答寄存器 0_1		TXACK0_1	16	H'FFFFE032		16		
中止应答寄存器 0_1		ABACK0_1	16	H'FFFFE03A		16		
数据帧接收完成寄存器 0_1		RXPR0_1	16	H'FFFFE042		16		
远程帧接收完成寄存器 0_1		RFPR0_1	16	H'FFFFE04A		16		
邮箱中断屏蔽寄存器 0_1		MBIMR0_1	16	H'FFFFE052		16		
未读信息状态寄存器 0_1		UMSR0_1	16	H'FFFFE05A		16		
MB[0].	CONTROL0H	—	16	H'FFFFE100		16、 32		
	CONTROL0L	—	16	H'FFFFE102		16		
	LAFMH	—	16	H'FFFFE104		16、 32		
	LAFML	—	16	H'FFFFE106		16		
	MSG_DATA[0]	—	8	H'FFFFE108		8、 16、 32		
	MSG_DATA[1]	—	8	H'FFFFE109		8		
	MSG_DATA[2]	—	8	H'FFFFE10A		8、 16		
	MSG_DATA[3]	—	8	H'FFFFE10B		8		
	MSG_DATA[4]	—	8	H'FFFFE10C		8、 16、 32		
	MSG_DATA[5]	—	8	H'FFFFE10D		8		
	MSG_DATA[6]	—	8	H'FFFFE10E		8、 16		
	MSG_DATA[7]	—	8	H'FFFFE10F		8		
	CONTROL1H	—	8	H'FFFFE110		8、 16		
	CONTROL1L	—	8	H'FFFFE111		8		

寄存器名称		简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
MB[1].	CONTROL0H	—	16	H'FFFFE120	RCAN-ET（通道 1）*1	16、32	Pϕ 基准 B:2、W:2、L:4	16 位
	CONTROL0L	—	16	H'FFFFE122		16		
	LAFMH	—	16	H'FFFFE124		16、32		
	LAFML	—	16	H'FFFFE126		16		
	MSG_DATA[0]	—	8	H'FFFFE128		8、16、32		
	MSG_DATA[1]	—	8	H'FFFFE129		8		
	MSG_DATA[2]	—	8	H'FFFFE12A		8、16		
	MSG_DATA[3]	—	8	H'FFFFE12B		8		
	MSG_DATA[4]	—	8	H'FFFFE12C		8、16、32		
	MSG_DATA[5]	—	8	H'FFFFE12D		8		
	MSG_DATA[6]	—	8	H'FFFFE12E		8、16		
	MSG_DATA[7]	—	8	H'FFFFE12F		8		
	CONTROL1H	—	8	H'FFFFE130		8、16		
	CONTROL1L	—	8	H'FFFFE131		8		
MB[2].	CONTROL0H	—	16	H'FFFFE140		16、32		
	CONTROL0L	—	16	H'FFFFE142		16		
	LAFMH	—	16	H'FFFFE144		16、32		
	LAFML	—	16	H'FFFFE146		16		
	MSG_DATA[0]	—	8	H'FFFFE148		8、16、32		
	MSG_DATA[1]	—	8	H'FFFFE149		8		
	MSG_DATA[2]	—	8	H'FFFFE14A		8、16		
	MSG_DATA[3]	—	8	H'FFFFE14B		8		
	MSG_DATA[4]	—	8	H'FFFFE14C		8、16、32		
	MSG_DATA[5]	—	8	H'FFFFE14D		8		
	MSG_DATA[6]	—	8	H'FFFFE14E		8、16		
	MSG_DATA[7]	—	8	H'FFFFE14F		8		
	CONTROL1H	—	8	H'FFFFE150		8、16		
	CONTROL1L	—	8	H'FFFFE151		8		
MB[3].	CONTROL0H	—	16	H'FFFFE160		16、32		
	CONTROL0L	—	16	H'FFFFE162		16		
	LAFMH	—	16	H'FFFFE164		16、32		
	LAFML	—	16	H'FFFFE166		16		
	MSG_DATA[0]	—	8	H'FFFFE168		8、16、32		
	MSG_DATA[1]	—	8	H'FFFFE169		8		
	MSG_DATA[2]	—	8	H'FFFFE16A		8、16		

寄存器名称		简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
MB[3].	MSG_DATA[3]	—	8	H'FFFFE16B	RCAN-ET（通道 1）*1	8	Pϕ 基准 B:2、 W:2、 L:4	16 位
	MSG_DATA[4]	—	8	H'FFFFE16C		8、 16、 32		
	MSG_DATA[5]	—	8	H'FFFFE16D		8		
	MSG_DATA[6]	—	8	H'FFFFE16E		8、 16		
	MSG_DATA[7]	—	8	H'FFFFE16F		8		
	CONTROL1H	—	8	H'FFFFE170		8、 16		
	CONTROL1L	—	8	H'FFFFE171		8		
MB[4].	CONTROL0H	—	16	H'FFFFE180		16、 32		
	CONTROL0L	—	16	H'FFFFE182		16		
	LAFMH	—	16	H'FFFFE184		16、 32		
	LAFML	—	16	H'FFFFE186		16		
	MSG_DATA[0]	—	8	H'FFFFE188		8、 16、 32		
	MSG_DATA[1]	—	8	H'FFFFE189		8		
	MSG_DATA[2]	—	8	H'FFFFE18A		8、 16		
	MSG_DATA[3]	—	8	H'FFFFE18B		8		
	MSG_DATA[4]	—	8	H'FFFFE18C		8、 16、 32		
	MSG_DATA[5]	—	8	H'FFFFE18D		8		
	MSG_DATA[6]	—	8	H'FFFFE18E		8、 16		
	MSG_DATA[7]	—	8	H'FFFFE18F		8		
	CONTROL1H	—	8	H'FFFFE190		8、 16		
	CONTROL1L	—	8	H'FFFFE191		8		
MB[5].	CONTROL0H	—	16	H'FFFFE1A0		16、 32		
	CONTROL0L	—	16	H'FFFFE1A2		16		
	LAFMH	—	16	H'FFFFE1A4		16、 32		
	LAFML	—	16	H'FFFFE1A6		16		
	MSG_DATA[0]	—	8	H'FFFFE1A8		8、 16、 32		
	MSG_DATA[1]	—	8	H'FFFFE1A9		8		
	MSG_DATA[2]	—	8	H'FFFFE1AA		8、 16		
	MSG_DATA[3]	—	8	H'FFFFE1AB		8		
	MSG_DATA[4]	—	8	H'FFFFE1AC		8、 16、 32		
	MSG_DATA[5]	—	8	H'FFFFE1AD		8		
	MSG_DATA[6]	—	8	H'FFFFE1AE		8、 16		
	MSG_DATA[7]	—	8	H'FFFFE1AF		8		
	CONTROL1H	—	8	H'FFFFE1B0		8、 16		
	CONTROL1L	—	8	H'FFFFE1B1		8		

寄存器名称		简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
MB[6].	CONTROL0H	—	16	H'FFFFE1C0	RCAN-ET（通道 1）*1	16、32	Pϕ 基准 B:2、W:2、L:4	16 位
	CONTROL0L	—	16	H'FFFFE1C2		16		
	LAFMH	—	16	H'FFFFE1C4		16、32		
	LAFML	—	16	H'FFFFE1C6		16		
	MSG_DATA[0]	—	8	H'FFFFE1C8		8、16、32		
	MSG_DATA[1]	—	8	H'FFFFE1C9		8		
	MSG_DATA[2]	—	8	H'FFFFE1CA		8、16		
	MSG_DATA[3]	—	8	H'FFFFE1CB		8		
	MSG_DATA[4]	—	8	H'FFFFE1CC		8、16、32		
	MSG_DATA[5]	—	8	H'FFFFE1CD		8		
	MSG_DATA[6]	—	8	H'FFFFE1CE		8、16		
	MSG_DATA[7]	—	8	H'FFFFE1CF		8		
	CONTROL1H	—	8	H'FFFFE1D0		8、16		
	CONTROL1L	—	8	H'FFFFE1D1		8		
MB[7].	CONTROL0H	—	16	H'FFFFE1E0		16、32		
	CONTROL0L	—	16	H'FFFFE1E2		16		
	LAFMH	—	16	H'FFFFE1E4		16、32		
	LAFML	—	16	H'FFFFE1E6		16		
	MSG_DATA[0]	—	8	H'FFFFE1E8		8、16、32		
	MSG_DATA[1]	—	8	H'FFFFE1E9		8		
	MSG_DATA[2]	—	8	H'FFFFE1EA		8、16		
	MSG_DATA[3]	—	8	H'FFFFE1EB		8		
	MSG_DATA[4]	—	8	H'FFFFE1EC		8、16、32		
	MSG_DATA[5]	—	8	H'FFFFE1ED		8		
	MSG_DATA[6]	—	8	H'FFFFE1EE		8、16		
	MSG_DATA[7]	—	8	H'FFFFE1EF		8		
	CONTROL1H	—	8	H'FFFFE1F0		8、16		
	CONTROL1L	—	8	H'FFFFE1F1		8		
MB[8].	CONTROL0H	—	16	H'FFFFE200		16、32		
	CONTROL0L	—	16	H'FFFFE202		16		
	LAFMH	—	16	H'FFFFE204		16、32		
	LAFML	—	16	H'FFFFE206		16		
	MSG_DATA[0]	—	8	H'FFFFE208		8、16、32		
	MSG_DATA[1]	—	8	H'FFFFE209		8		
	MSG_DATA[2]	—	8	H'FFFFE20A		8、16		

寄存器名称		简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
MB[8].	MSG_DATA[3]	—	8	H'FFFFE20B	RCAN-ET（通道 1）*1	8	Pϕ 基准 B:2、W:2、L:4	16 位
	MSG_DATA[4]	—	8	H'FFFFE20C		8、16、32		
	MSG_DATA[5]	—	8	H'FFFFE20D		8		
	MSG_DATA[6]	—	8	H'FFFFE20E		8、16		
	MSG_DATA[7]	—	8	H'FFFFE20F		8		
	CONTROL1H	—	8	H'FFFFE210		8、16		
	CONTROL1L	—	8	H'FFFFE211		8		
MB[9].	CONTROL0H	—	16	H'FFFFE220		16、32		
	CONTROL0L	—	16	H'FFFFE222		16		
	LAFMH	—	16	H'FFFFE224		16、32		
	LAFML	—	16	H'FFFFE226		16		
	MSG_DATA[0]	—	8	H'FFFFE228		8、16、32		
	MSG_DATA[1]	—	8	H'FFFFE229		8		
	MSG_DATA[2]	—	8	H'FFFFE22A		8、16		
	MSG_DATA[3]	—	8	H'FFFFE22B		8		
	MSG_DATA[4]	—	8	H'FFFFE22C		8、16、32		
	MSG_DATA[5]	—	8	H'FFFFE22D		8		
	MSG_DATA[6]	—	8	H'FFFFE22E		8、16		
	MSG_DATA[7]	—	8	H'FFFFE22F		8		
	CONTROL1H	—	8	H'FFFFE230		8、16		
	CONTROL1L	—	8	H'FFFFE231		8		
MB[10].	CONTROL0H	—	16	H'FFFFE240		16、32		
	CONTROL0L	—	16	H'FFFFE242		16		
	LAFMH	—	16	H'FFFFE244		16、32		
	LAFML	—	16	H'FFFFE246		16		
	MSG_DATA[0]	—	8	H'FFFFE248		8、16、32		
	MSG_DATA[1]	—	8	H'FFFFE249		8		
	MSG_DATA[2]	—	8	H'FFFFE24A		8、16		
	MSG_DATA[3]	—	8	H'FFFFE24B		8		
	MSG_DATA[4]	—	8	H'FFFFE24C		8、16、32		
	MSG_DATA[5]	—	8	H'FFFFE24D		8		
	MSG_DATA[6]	—	8	H'FFFFE24E		8、16		
	MSG_DATA[7]	—	8	H'FFFFE24F		8		
	CONTROL1H	—	8	H'FFFFE250		8、16		
	CONTROL1L	—	8	H'FFFFE251		8		

寄存器名称		简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
MB[11].	CONTROL0H	—	16	H'FFFFE260	RCAN-ET（通道 1）*1	16、32	Pϕ 基准 B:2、W:2、L:4	16 位
	CONTROL0L	—	16	H'FFFFE262		16		
	LAFMH	—	16	H'FFFFE264		16、32		
	LAFML	—	16	H'FFFFE266		16		
	MSG_DATA[0]	—	8	H'FFFFE268		8、16、32		
	MSG_DATA[1]	—	8	H'FFFFE269		8		
	MSG_DATA[2]	—	8	H'FFFFE26A		8、16		
	MSG_DATA[3]	—	8	H'FFFFE26B		8		
	MSG_DATA[4]	—	8	H'FFFFE26C		8、16、32		
	MSG_DATA[5]	—	8	H'FFFFE26D		8		
	MSG_DATA[6]	—	8	H'FFFFE26E		8、16		
	MSG_DATA[7]	—	8	H'FFFFE26F		8		
	CONTROL1H	—	8	H'FFFFE270		8、16		
	CONTROL1L	—	8	H'FFFFE271		8		
MB[12].	CONTROL0H	—	16	H'FFFFE280		16、32		
	CONTROL0L	—	16	H'FFFFE282		16		
	LAFMH	—	16	H'FFFFE284		16、32		
	LAFML	—	16	H'FFFFE286		16		
	MSG_DATA[0]	—	8	H'FFFFE288		8、16、32		
	MSG_DATA[1]	—	8	H'FFFFE289		8		
	MSG_DATA[2]	—	8	H'FFFFE28A		8、16		
	MSG_DATA[3]	—	8	H'FFFFE28B		8		
	MSG_DATA[4]	—	8	H'FFFFE28C		8、16、32		
	MSG_DATA[5]	—	8	H'FFFFE28D		8		
	MSG_DATA[6]	—	8	H'FFFFE28E		8、16		
	MSG_DATA[7]	—	8	H'FFFFE28F		8		
	CONTROL1H	—	8	H'FFFFE290		8、16		
	CONTROL1L	—	8	H'FFFFE291		8		
MB[13].	CONTROL0H	—	16	H'FFFFE2A0		16、32		
	CONTROL0L	—	16	H'FFFFE2A2		16		
	LAFMH	—	16	H'FFFFE2A4		16、32		
	LAFML	—	16	H'FFFFE2A6		16		
	MSG_DATA[0]	—	8	H'FFFFE2A8		8、16、32		
	MSG_DATA[1]	—	8	H'FFFFE2A9		8		
	MSG_DATA[2]	—	8	H'FFFFE2AA		8、16		

寄存器名称		简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
MB[13].	MSG_DATA[3]	—	8	H'FFFFE2AB	RCAN-ET（通道 1）*1	8	Pϕ 基准 B:2、 W:2、 L:4	16 位
	MSG_DATA[4]	—	8	H'FFFFE2AC		8、 16、 32		
	MSG_DATA[5]	—	8	H'FFFFE2AD		8		
	MSG_DATA[6]	—	8	H'FFFFE2AE		8、 16		
	MSG_DATA[7]	—	8	H'FFFFE2AF		8		
	CONTROL1H	—	8	H'FFFFE2B0		8、 16		
	CONTROL1L	—	8	H'FFFFE2B1		8		
MB[14].	CONTROL0H	—	16	H'FFFFE2C0		16、 32		
	CONTROL0L	—	16	H'FFFFE2C2		16		
	LAFMH	—	16	H'FFFFE2C4		16、 32		
	LAFML	—	16	H'FFFFE2C6		16		
	MSG_DATA[0]	—	8	H'FFFFE2C8		8、 16、 32		
	MSG_DATA[1]	—	8	H'FFFFE2C9		8		
	MSG_DATA[2]	—	8	H'FFFFE2CA		8、 16		
	MSG_DATA[3]	—	8	H'FFFFE2CB		8		
	MSG_DATA[4]	—	8	H'FFFFE2CC		8、 16、 32		
	MSG_DATA[5]	—	8	H'FFFFE2CD		8		
	MSG_DATA[6]	—	8	H'FFFFE2CE		8、 16		
	MSG_DATA[7]	—	8	H'FFFFE2CF		8		
	CONTROL1H	—	8	H'FFFFE2D0		8、 16		
	CONTROL1L	—	8	H'FFFFE2D1		8		
MB[15].	CONTROL0H	—	16	H'FFFFE2E0		16、 32		
	CONTROL0L	—	16	H'FFFFE2E2		16		
	LAFMH	—	16	H'FFFFE2E4		16、 32		
	LAFML	—	16	H'FFFFE2E6		16		
	MSG_DATA[0]	—	8	H'FFFFE2E8		8、 16、 32		
	MSG_DATA[1]	—	8	H'FFFFE2E9		8		
	MSG_DATA[2]	—	8	H'FFFFE2EA		8、 16		
	MSG_DATA[3]	—	8	H'FFFFE2EB		8		
	MSG_DATA[4]	—	8	H'FFFFE2EC		8、 16、 32		
	MSG_DATA[5]	—	8	H'FFFFE2ED		8		
	MSG_DATA[6]	—	8	H'FFFFE2EE		8、 16		
	MSG_DATA[7]	—	8	H'FFFFE2EF		8		
	CONTROL1H	—	8	H'FFFFE2F0		8、 16		
	CONTROL1L	—	8	H'FFFFE2F1		8		

寄存器名称	简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
频率控制寄存器	FRQCR	16	H'FFFFE800	CPG	16	Pϕ 基准 W:2	16 位
待机控制寄存器 1	STBCR1	8	H'FFFFE802	低功耗	8	Pϕ 基准 B:2	16 位
待机控制寄存器 2	STBCR2	8	H'FFFFE804		8		
待机控制寄存器 3	STBCR3	8	H'FFFFE806		8		
待机控制寄存器 4	STBCR4	8	H'FFFFE808		8		
待机控制寄存器 5	STBCR5	8	H'FFFFE80A		8		
待机控制寄存器 6	STBCR6	8	H'FFFFE80C		8		
看门狗定时器计数器	WTCNT	8	H'FFFFE810	WDT	8*2、16*3	Pϕ 基准 B:2*2、 W:2*3	16 位
看门狗定时器控制 / 状态寄存器	WTCSR	8	H'FFFFE812		8*2、16*3		
振荡停止检测控制寄存器	OSCCR	8	H'FFFFE814	CPG	8	Pϕ 基准 B:2	16 位
RAM 控制寄存器	RAMCR	8	H'FFFFE880	低功耗	8	Pϕ 基准 B:2	16 位
总线功能扩展寄存器	BSCEHR	16	H'FFFFE89A	BSC	8、16	Pϕ 基准 B:2、W:2	16 位
中断控制寄存器 0	ICR0	16	H'FFFFE900	INTC	8、16	Pϕ 基准 B:2、W:2	16 位
IRQ 控制寄存器	IRQCR	16	H'FFFFE902		8、16		
IRQ 状态寄存器	IRQSR	16	H'FFFFE904		8、16		
中断优先级寄存器 A	IPRA	16	H'FFFFE906		8、16		
中断优先级寄存器 D	IPRD	16	H'FFFFE982		16		
中断优先级寄存器 E	IPRE	16	H'FFFFE984		16		
中断优先级寄存器 F	IPRF	16	H'FFFFE986		16		
中断优先级寄存器 H	IPRH	16	H'FFFFE98A		16		
中断优先级寄存器 I	IPRI	16	H'FFFFE98C		16		
中断优先级寄存器 J	IPRJ	16	H'FFFFE98E		16		
中断优先级寄存器 K	IPRK	16	H'FFFFE990		16		
中断优先级寄存器 L	IPRL	16	H'FFFFE992		16		
中断优先级寄存器 M	IPRM	16	H'FFFFE994		16		
共用控制寄存器	CMNCR	32	H'FFFFF000	BSC	32	Bϕ 基准 L:2	16 位
CS0 空间总线控制寄存器	CS0BCR	32	H'FFFFF004		32		
CS1 空间总线控制寄存器	CS1BCR	32	H'FFFFF008		32		
CS0 空间等待控制寄存器	CS0WCR	32	H'FFFFF028		32		
CS1 空间等待控制寄存器	CS1WCR	32	H'FFFFF02C		32		
RAM 仿真寄存器	RAMER	16	H'FFFFF108	FLASH	16	Bϕ 基准 W:2	16 位

寄存器名称	简称	位数	地址	模块	存取长度	存取状态数	连接总线宽度
断点地址寄存器 A	BARA	32	H'FFFFF300	UBC	32	Bϕ 基准 B:2、 W:2、 L:2	16 位
断点地址屏蔽寄存器 A	BAMRA	32	H'FFFFF304		32		
断点总线周期寄存器 A	BBRA	16	H'FFFFF308		16		
断点数据寄存器 A	BDRA	32	H'FFFFF310		32		
断点数据屏蔽寄存器 A	BDMRA	32	H'FFFFF314		32		
断点地址寄存器 B	BARB	32	H'FFFFF320		32		
断点地址屏蔽寄存器 B	BAMRB	32	H'FFFFF324		32		
断点总线周期寄存器 B	BBRB	16	H'FFFFF328		16		
断点数据寄存器 B	BDRB	32	H'FFFFF330		32		
断点数据屏蔽寄存器 B	BDMRB	32	H'FFFFF334		32		
断点控制寄存器	BRCR	32	H'FFFFF3C0		32		
转移源寄存器	BRSR	32	H'FFFFF3D0		32		
转移目标寄存器	BRDR	32	H'FFFFF3D4		32		
执行次数断点寄存器	BETR	16	H'FFFFF3DC		16		
AUD 控制寄存器	AUCSR	32	H'FFFFF400	AUD	32	Bϕ 基准 L:2	16 位
AUD 窗口 A 起始地址寄存器	AUWASR	32	H'FFFFF404		32		
AUD 窗口 A 结束地址寄存器	AUWAER	32	H'FFFFF408		32		
AUD 窗口 B 起始地址寄存器	AUWBSR	32	H'FFFFF40C		32		
AUD 窗口 B 结束地址寄存器	AUWBER	32	H'FFFFF410		32		
AUD 扩展控制寄存器	AUECSR	32	H'FFFFF414		32		

【注】 *1　仅适用于 SH7142

*2　读取时

*3　写入时

24.2　寄存器位一览表

内部外围模块寄存器的地址和位名称如下所示。

16 位、 32 位寄存器每 8 位分 2 段或 4 段表示。

寄存器简称	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0	模块
SCSMR_0	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS[1:0]		SCI（通道 0）
SCBRR_0									
SCSCR_0	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]		
SCTDR_0									
SCSSR_0	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
SCRDR_0									
SCSDCR_0	—	—	—	—	DIR	—	—	—	
SCSPTR_0	EIO	—	—	—	SPB1IO	SPB1DT	SPB0IO	SPB0DT	
SCSMR_1	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS[1:0]		SCI（通道 1）
SCBRR_1									
SCSCR_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]		
SCTDR_1									
SCSSR_1	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
SCRDR_1									
SCSDCR_1	—	—	—	—	DIR	—	—	—	
SCSPTR_1	EIO	—	—	—	SPB1IO	SPB1DT	SPB0IO	SPB0DT	
SCSMR_2	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS[1:0]		SCI（通道 2）
SCBRR_2									
SCSCR_2	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]		
SCTDR_2									
SCSSR_2	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
SCRDR_2									
SCSDCR_2	—	—	—	—	DIR	—	—	—	
SCSPTR_2	EIO	—	—	—	SPB1IO	SPB1DT	SPB0IO	SPB0DT	
TCR_3	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]			MTU2
TCR_4	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]			
TMDR_3	—	—	BFB	BFA	MD[3:0]				
TMDR_4	—	—	BFB	BFA	MD[3:0]				
TIORH_3	IOB[3:0]				IOA[3:0]				
TIORL_3	IOD[3:0]				IOC[3:0]				
TIORH_4	IOB[3:0]				IOA[3:0]				

寄存器简称	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0	模块
TIORL_4	IOD[3:0]				IOC[3:0]				MTU2
TIER_3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TIER_4	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TOER	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B	
TGCR	—	BDC	N	P	FB	WF	VF	UF	
TOCR1	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP	
TOCR2	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P	
TCNT_3									
TCNT_4									
TCDR									
TDDR									
TGRA_3									
TGRB_3									
TGRA_4									
TGRB_4									
TCNTS									
TCBR									
TGRC_3									
TGRD_3									
TGRC_4									
TGRD_4									

寄存器简称	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0	模块
TSR_3	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	MTU2
TSR_4	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TITCR	T3AEN	3ACOR[2:0]			T4VEN	4VCOR[2:0]			
TITCNT	—	3ACNT[2:0]			—	4VCNT[2:0]			
TBTER	—	—	—	—	—	—	BTE[1:0]		
TDER	—	—	—	—	—	—	—	TDER	
TOLBR	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P	
TBTM_3	—	—	—	—	—	—	TTSB	TTSA	
TBTM_4	—	—	—	—	—	—	TTSB	TTSA	
TADCR	BF[1:0]		—	—	—	—	—	—	
	UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE	
TADCORA_4									
TADCORB_4									
TADCOBRA_4									
TADCOBRB_4									
TWCR	CCE	—	—	—	—	—	—	WRE	
TSTR	CST4	CST3	—	—	—	CST2	CST1	CST0	
TSYR	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0	
TCSYSTR	SCH0	SCH1	SCH2	SCH3	SCH4	—	SCH3S	SCH4S	
TRWER	—	—	—	—	—	—	—	RWE	
TCR_0	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]			
TMDR_0	—	BFE	BFB	BFA	MD[3:0]				
TIORH_0	IOB[3:0]				IOA[3:0]				
TIORL_0	IOD[3:0]				IOC[3:0]				
TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_0									
TGRA_0									

寄存器简称	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0	模块
TGRB_0									MTU2
TGRC_0									
TGRD_0									
TGRE_0									
TGRF_0									
TIER2_0	TTGE2	—	—	—	—	—	TGIEF	TGIEE	
TSR2_0	—	—	—	—	—	—	TGFF	TGFE	
TBTM_0	—	—	—	—	—	TTSE	TTSB	TTSA	
TCR_1	—	CCLR[1:0]		CKEG[1:0]		TPSC[2:0]			
TMDR_1	—	—	—	—	MD[3:0]				
TIOR_1	IOB[3:0]				IOA[3:0]				
TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_1									
TGRA_1									
TGRB_1									
TICCR	—	—	—	—	I2BE	I2AE	I1BE	I1AE	
TCR_2	—	CCLR[1:0]		CKEG[1:0]		TPSC[2:0]			
TMDR_2	—	—	—	—	MD[3:0]				
TIOR_2	IOB[3:0]				IOA[3:0]				
TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_2									
TGRA_2									
TGRB_2									

寄存器简称	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0	模块
TCR_3S	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]			MTU2S
TCR_4S	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]			
TMDR_3S	—	—	BFB	BFA	MD[3:0]				
TMDR_4S	—	—	BFB	BFA	MD[3:0]				
TIORH_3S	IOB[3:0]				IOA[3:0]				
TIORL_3S	IOD[3:0]				IOC[3:0]				
TIORH_4S	IOB[3:0]				IOA[3:0]				
TIORL_4S	IOD[3:0]				IOC[3:0]				
TIER_3S	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TIER_4S	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TOERS	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B	
TGCRS	—	BDC	N	P	FB	WF	VF	UF	
TOCR1S	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP	
TOCR2S	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P	
TCNT_3S									
TCNT_4S									
TCDRS									
TDDRS									
TGRA_3S									
TGRB_3S									
TGRA_4S									
TGRB_4S									
TCNTSS									
TCBRS									

寄存器简称	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0	模块
TGRC_3S									MTU2S
TGRD_3S									
TGRC_4S									
TGRD_4S									
TSR_3S	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TSR_4S	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TITCRS	T3AEN	3ACOR[2:0]			T4VEN	4VCOR[2:0]			
TITCNTS	—	3ACNT[2:0]			—	4VCNT[2:0]			
TBTERS	—	—	—	—	—	—	BTE[1:0]		
TDERS	—	—	—	—	—	—	—	TDER	
TOLBRS	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P	
TBTM_3S	—	—	—	—	—	—	TTSB	TTSA	
TBTM_4S	—	—	—	—	—	—	TTSB	TTSA	
TADCRS	BF[1:0]		—	—	—	—	—	—	
	UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE	
TADCORA_4S									
TADCORB_4S									
TADCOBRA_4S									
TADCOBRB_4S									
TSYCRS	CE0A	CE0B	CE0C	CE0D	CE1A	CE1B	CE2A	CE2B	
TWCRS	CCE	—	—	—	—	—	SCC	WRE	
TSTRS	CST4	CST3	—	—	—	CST2	CST1	CST0	
TSYRS	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0	
TRWERS	—	—	—	—	—	—	—	RWE	
TCNTU_5S									

寄存器简称	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0	模块
TGRU_5S									MTU2S
TCRU_5S	—	—	—	—	—	—	TPSC[1:0]		
TIORU_5S	—	—	—	IOC[4:0]					
TCNTV_5S									
TGRV_5S									
TCRV_5S	—	—	—	—	—	—	TPSC[1:0]		
TIORV_5S	—	—	—	IOC[4:0]					
TCNTW_5S									
TGRW_5S									
TCRW_5S	—	—	—	—	—	—	TPSC[1:0]		
TIORW_5S	—	—	—	IOC[4:0]					
TSR_5S	—	—	—	—	—	CMFU5	CMFV5	CMFW5	
TIER_5S	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W	
TSTR_5S	—	—	—	—	—	CSTU5	CSTV5	CSTW5	
TCNTCMPCLRS	—	—	—	—	—	CMPCLR5U	CMPCLR5V	CMPCLR5W	
FCCS	FWE	MAT	—	FLER	—	—	—	SCO	FLASH
FPCS	—	—	—	—	—	—	—	PPVS	
FECS	—	—	—	—	—	—	—	EPVB	
FKEY	K[7:0]								
FMATS	MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0	
FTDAR	TDER	TDA[6:0]							
DTCERA	DTCERA15	DTCERA14	DTCERA13	DTCERA12	—	—	—	—	DTC
	—	—	—	—	—	—	—	—	
DTCERB	DTCERB15	DTCERB14	DTCERB13	DTCERB12	DTCERB11	DTCERB10	DTCERB9	DTCERB8	
	DTCERB7	DTCERB6	DTCERB5	DTCERB4	DTCERB3	DTCERB2	DTCERB1	DTCERB0	
DTCERC	DTCERC15	—	—	—	—	—	—	—	
	—	—	—	—	DTCERC3	DTCERC2	DTCERC1	DTCERC0	

寄存器简称	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0	模块
DTCERD	DTCERD15	DTCERD14	DTCERD13	DTCERD12	DTCERD11	DTCERD10	DTCERD9	DTCERD8	DTC
	DTCERD7	DTCERD6	—	—	—	DTCERD2	DTCERD1	—	
DTCERE	DTCERE15	DTCERE14	DTCERE13	DTCERE12	DTCERE11	DTCERE10	—	—	
	DTCERE7	DTCERE6	—	—	DTCERE3	—	—	—	
DTCCR	—	—	—	RRS	RCHNE	—	—	ERR	
DTCVBR									
					—	—	—	—	
	—	—	—	—	—	—	—	—	
SSCRH	MSS	BIDE	—	SOL	SOLP	—	CSS[1:0]		SSU
SSCRL	FCLRM	SSUMS	SRES	—	—	—	DATS[1:0]		
SSMR	MLS	CPOS	CPHS	—	—	CKS[2:0]			
SSER	TE	RE	—	—	TEIE	TIE	RIE	CEIE	
SSSR	—	ORER	—	—	TEND	TDRE	RDRF	CE	
SSCR2	—	—	—	TENDSTS	SCSATS	SSODTS	—	—	
SSTDR0									
SSTDR1									
SSTDR2									
SSTDR3									
SSRDR0									
SSRDR1									
SSRDR2									
SSRDR3									
CMSTR	—	—	—	—	—	—	—	—	CMT
	—	—	—	—	—	—	STR1	STR0	
CMCSR_0	—	—	—	—	—	—	—	—	
	CMF	CMIE	—	—	—	—	CKS[1:0]		
CMCNT_0									
CMCOR_0									
CMCSR_1	—	—	—	—	—	—	—	—	
	CMF	CMIE	—	—	—	—	CKS[1:0]		
CMCNT_1									

寄存器简称	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0	模块
CMCOR_1									CMT
ICSR1	—	POE2F	POE1F	POE0F	—	—	—	PIE1	POE
	—	—	POE2M[1:0]		POE1M[1:0]		POE0M[1:0]		
OCSR1	OSF1	—	—	—	—	—	OCE1	OIE1	
	—	—	—	—	—	—	—	—	
ICSR2	—	POE6F	POE5F	POE4F	—	—	—	PIE2	
	—	—	POE6M[1:0]		POE5M[1:0]		POE4M[1:0]		
OCSR2	OSF2	—	—	—	—	—	OCE2	OIE2	
	—	—	—	—	—	—	—	—	
ICSR3	—	—	—	POE8F	—	—	POE8E	PIE3	
	—	—	—	—	—	—	POE8M[1:0]		
SPOER	—	—	—	—	—	MTU2SHIZ	MTU2CH0 HIZ	MTU2CH34 HIZ	
POECR1	—	—	—	—	MTU2PE3 ZE	MTU2PE2 ZE	MTU2PE1 ZE	MTU2PE0Z E	
POECR2	—	MTU2P1C ZE	MTU2P2C ZE	MTU2P3C ZE	—	MTU2SP1 CZE	MTU2SP2 CZE	MTU2SP3C ZE	
	—	—	—	—	—	—	—	—	
PADRL	PA15DR	PA14DR	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR	PA8DR	I/O
	PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR	
PAIORL	PA15IOR	PA14IOR	PA13IOR	PA12IOR	PA11IOR	PA10IOR	PA9IOR	PA8IOR	PFC
	PA7IOR	PA6IOR	PA5IOR	PA4IOR	PA3IOR	PA2IOR	PA1IOR	PA0IOR	
PACRL4	—	PA15MD2	PA15MD1	PA15MD0	—	PA14MD2	PA14MD1	PA14MD0	
	—	PA13MD2	PA13MD1	PA13MD0	—	PA12MD2	PA12MD1	PA12MD0	
PACRL3	—	PA11MD2	PA11MD1	PA11MD0	—	PA10MD2	PA10MD1	PA10MD0	
	—	PA9MD2	PA9MD1	PA9MD0	—	PA8MD2	PA8MD1	PA8MD0	
PACRL2	—	PA7MD2	PA7MD1	PA7MD0	—	PA6MD2	PA6MD1	PA6MD0	
	—	PA5MD2	PA5MD1	PA5MD0	—	PA4MD2	PA4MD1	PA4MD0	
PACRL1	—	PA3MD2	PA3MD1	PA3MD0	—	PA2MD2	PA2MD1	PA2MD0	
	—	PA1MD2	PA1MD1	PA1MD0	—	PA0MD2	PA0MD1	PA0MD0	
PAPRL	PA15PR	PA14PR	PA13PR	PA12PR	PA11PR	PA10PR	PA9PR	PA8PR	I/O
	PA7PR	PA6PR	PA5PR	PA4PR	PA3PR	PA2PR	PA1PR	PA0PR	
PBDRL	—	—	—	—	—	—	—	—	
	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR	

寄存器简称	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0	模块
PBIORL	—	—	—	—	—	—	—	—	PFC
	PB7IOR	PB6IOR	PB5IOR	PB4IOR	PB3IOR	PB2IOR	PB1IOR	PB0IOR	
PBCRL2	—	PB7MD2	PB7MD1	PB7MD0	—	PB6MD2	PB6MD1	PB6MD0	
	—	PB5MD2	PB5MD1	PB5MD0	—	PB4MD2	PB4MD1	PB4MD0	
PBCRL1	—	PB3MD2	PB3MD1	PB3MD0	—	PB2MD2	PB2MD1	PB2MD0	
	—	PB1MD2	PB1MD1	PB1MD0	—	PB0MD2	PB0MD1	PB0MD0	
PBPRL	—	—	—	—	—	—	—	—	I/O
	PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	PB0PR	
PDDRL	—	—	—	—	—	PD10DR	PD9DR	PD8DR	
	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR	
PDIORL	—	—	—	—	—	PD10IOR	PD9IOR	PD8IOR	PFC
	PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR	
PDCRL3	—	—	—	—	—	PD10MD2	PD10MD1	PD10MD0	
	—	PD9MD2	PD9MD1	PD9MD0	—	PD8MD2	PD8MD1	PD8MD0	
PDCRL2	—	PD7MD2	PD7MD1	PD7MD0	—	PD6MD2	PD6MD1	PD6MD0	
	—	PD5MD2	PD5MD1	PD5MD0	—	PD4MD2	PD4MD1	PD4MD0	
PDCRL1	—	PD3MD2	PD3MD1	PD3MD0	—	PD2MD2	PD2MD1	PD2MD0	
	—	PD1MD2	PD1MD1	PD1MD0	—	PD0MD2	PD0MD1	PD0MD0	
PDPRL	—	—	—	—	—	PD10PR	PD9PR	PD8PR	I/O
	PD7PR	PD6PR	PD5PR	PD4PR	PD3PR	PD2PR	PD1PR	PD0PR	
PEDRH	—	—	—	—	—	—	—	—	
	—	—	PE21DR	PE20DR	PE19DR	PE18DR	PE17DR	PE16DR	
PEDRL	PE15DR	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR	
	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR	
PEIORH	—	—	—	—	—	—	—	—	PFC
	—	—	PE21IOR	PE20IOR	PE19IOR	PE18IOR	PE17IOR	PE16IOR	
PEIORL	PE15IOR	PE14IOR	PE13IOR	PE12IOR	PE11IOR	PE10IOR	PE9IOR	PE8IOR	
	PE7IOR	PE6IOR	PE5IOR	PE4IOR	PE3IOR	PE2IOR	PE1IOR	PE0IOR	
PECRH2	—	—	—	—	—	—	—	—	
	—	—	PE21MD1	PE21MD0	—	—	PE20MD1	PE20MD0	
PECRH1	—	—	PE19MD1	PE19MD0	—	—	PE18MD1	PE18MD0	
	—	—	PE17MD1	PE17MD0	—	PE16MD2	PE16MD1	PE16MD0	
PECRL4	—	PE15MD2	PE15MD1	PE15MD0	—	PE14MD2	PE14MD1	PE14MD0	
	—	—	PE13MD1	PE13MD0	—	PE12MD2	PE12MD1	PE12MD0	
PECRL3	—	PE11MD2	PE11MD1	PE11MD0	—	PE10MD2	PE10MD1	PE10MD0	
	—	PE9MD2	PE9MD1	PE9MD0	—	PE8MD2	PE8MD1	PE8MD0	

寄存器简称	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0	模块
PECRL2	—	PE7MD2	PE7MD1	PE7MD0	—	PE6MD2	PE6MD1	PE6MD0	PFC
	—	PE5MD2	PE5MD1	PE5MD0	—	PE4MD2	PE4MD1	PE4MD0	
PECRL1	—	PE3MD2	PE3MD1	PE3MD0	—	PE2MD2	PE2MD1	PE2MD0	
	—	PE1MD2	PE1MD1	PE1MD0	—	—	PE0MD1	PE0MD0	
PEPRH	—	—	—	—	—	—	—	—	I/O
	—	—	PE21PR	PE20PR	PE19PR	PE18PR	PE17PR	PE16PR	
PEPRL	PE15PR	PE14PR	PE13PR	PE12PR	PE11PR	PE10PR	PE9PR	PE8PR	
	PE7PR	PE6PR	PE5PR	PE4PR	PE3PR	PE2PR	PE1PR	PE0PR	
ADCR_0	ADST	ADCS	ACE	ADIE	—	—	TRGE	EXTRG	A/D（通道 0）
ADSR_0	—	—	—	—	—	—	—	ADF	
ADSTRGR_0	—	STR6	STR5	STR4	STR3	STR2	STR1	STR0	
ADANSR_0	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	
ADDR0	—	—	—	—	ADD[11:8]				
	ADD[7:0]								
ADDR1	—	—	—	—	ADD[11:8]				
	ADD[7:0]								
ADDR2	—	—	—	—	ADD[11:8]				
	ADD[7:0]								
ADDR3	—	—	—	—	ADD[11:8]				
	ADD[7:0]								
ADDR4	—	—	—	—	ADD[11:8]				
	ADD[7:0]								
ADDR5	—	—	—	—	ADD[11:8]				
	ADD[7:0]								
ADDR6	—	—	—	—	ADD[11:8]				
	ADD[7:0]								
ADDR7	—	—	—	—	ADD[11:8]				
	ADD[7:0]								
ADCR_1	ADST	ADCS	ACE	ADIE	—	—	TRGE	EXTRG	A/D（通道 1）
ADSR_1	—	—	—	—	—	—	—	ADF	
ADSTRGR_1	—	STR6	STR5	STR4	STR3	STR2	STR1	STR0	
ADANSR_1	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	
ADDR8	—	—	—	—	ADD[11:8]				
	ADD[7:0]								

寄存器简称	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0	模块
ADDR9	—	—	—	—	ADD[11:8]				A/D（通道 1）
	ADD[7:0]								
ADDR10	—	—	—	—	ADD[11:8]				
	ADD[7:0]								
ADDR11	—	—	—	—	ADD[11:8]				
	ADD[7:0]								
ADDR12	—	—	—	—	ADD[11:8]				
	ADD[7:0]								
ADDR13	—	—	—	—	ADD[11:8]				
	ADD[7:0]								
ADDR14	—	—	—	—	ADD[11:8]				
	ADD[7:0]								
ADDR15	—	—	—	—	ADD[11:8]				
	ADD[7:0]								
MCR_0	MCR15	MCR14	—	—	—	TST[2:0]			RCAN-ET（通道 0）
	MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0	
GSR_0	—	—	—	—	—	—	—	—	
	—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0	
BCR1_0	TSG1[3:0]				—	TSG2[2:0]			
	—	—	SJW[1:0]		—	—	—	BSP	
BCR0_0	—	—	—	—	—	—	—	—	
	BRP[7:0]								
IRR_0	—	—	IRR13	IRR12	—	—	IRR9	IRR8	
	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0	
IMR_0	IMR15	IMR14	MR13	IMR12	IMR11	IMR10	IMR9	IMR8	
	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0	
TEC/REC_0	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	
	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0	
TXPR1_0、0_0	TXPR1[15:8]								
	TXPR1[7:0]								
	TXPR0[15:8]								
	TXPR0[7:1]							—	
TXCR0_0	TXCR0[15:8]								
	TXCR0[7:1]							—	
TXACK0_0	TXACK0[15:8]								
	TXACK0[7:1]							—	

寄存器简称	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0	模块
ABACK0_0	ABACK0[15:8]								RCAN-ET（通道 0）
	ABACK0[7:1]							—	
RXPR0_0	RXPR0[15:8]								
	RXPR0[7:0]								
RFPR0_0	RFPR0[15:8]								
	RFPR0[7:0]								
MBIMR0_0	MBIMR0[15:8]								
	MBIMR0[7:0]								
UMSR0_0	UMSR0[15:8]								
	UMSR0[7:0]								
MB[0]. CONTROL0H	IDE	RTR	—	STDID[10:6]					RCAN-ET（MCR15 = 1）
	STDID[5:0]						EXTID[17:16]		
MB[0]. CONTROL0H	—	STDID[10:4]							RCAN-ET（MCR15 = 0）
	STDID[3:0]				RTR	IDE	EXTID[17:16]		
MB[0]. CONTROL0L	EXTID[15:8]								RCAN-ET（通道 0）
	EXTID[7:0]								
MB[0]. LAFMH	IDE_LAFM	—	—	STDID_LAFM[10:6]					RCAN-ET（MCR15 = 1）（通道 0）
	STDID_LAFM[5:0]						EXTID_LAFM[17:16]		
MB[0]. LAFMH	—	STDID_LAFM[10:4]							RCAN-ET（MCR15 = 0）（通道 0）
	STDID_LAFM[3:0]				—	IDE_LAFM	EXTID_LAFM[17:16]		
MB[0]. LAFML	EXTID_LAFM[15:8]								RCAN-ET（通道 0）
	EXTID_LAFM[7:0]								
MB[0]. MSG_DATA[0]	MSG_DATA_0								
MB[0]. MSG_DATA[1]	MSG_DATA_1								
MB[0]. MSG_DATA[2]	MSG_DATA_2								
MB[0]. MSG_DATA[3]	MSG_DATA_3								
MB[0]. MSG_DATA[4]	MSG_DATA_4								
MB[0]. MSG_DATA[5]	MSG_DATA_5								
MB[0]. MSG_DATA[6]	MSG_DATA_6								

寄存器简称	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0	模块
MB[0]. MSG_DATA[7]	MSG_DATA_7								RCAN-ET（通道 0）
MB[0]. CONTROL1H	—	—	NMC	—	—	MBC[2:0]			
MB[0]. CONTROL1L	—	—	—	—	DLC[3:0]				
MB[1]	与 MB[0] 的位结构相同								
MB[2]	与 MB[0] 的位结构相同								
MB[3]	与 MB[0] 的位结构相同								
↓	（重复）								
MB[13]	与 MB[0] 的位结构相同								
MB[14]	与 MB[0] 的位结构相同								
MB[15]	与 MB[0] 的位结构相同								
MCR_1	MCR15	MCR14	—	—	—	TST[2:0]			RCAN-ET（通道 1）*
	MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0	
GSR_1	—	—	—	—	—	—	—	—	
	—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0	
BCR1_1	TSG1[3:0]				—	TSG2[2:0]			
	—	—	SJW[1:0]		—	—	—	BSP	
BCR0_1	—	—	—	—	—	—	—	—	
	BRP[7:0]								
IRR_1	—	—	IRR13	IRR12	—	—	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	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	
	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0	
TXPR1_1、0_1	TXPR1[15:8]								
	TXPR1[7:0]								
	TXPR0[15:8]								
	TXPR0[7:1]							—	
TXCR0_1	TXCR0[15:8]								
	TXCR0[7:1]							—	
TXACK0_1	TXACK0[15:8]								
	TXACK0[7:1]							—	
ABACK0_1	ABACK0[15:8]								
	ABACK0[7:1]							—	

寄存器简称	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0	模块
RXPR0_1	RXPR0[15:8]								RCAN-ET（通道 1）*
	RXPR0[7:0]								
RFPR0_1	RFPR0[15:8]								
	RFPR0[7:0]								
MBIMR0_1	MBIMR0[15:8]								
	MBIMR0[7:0]								
UMSR0_1	UMSR0[15:8]								
	UMSR0[7:0]								
MB[0]. CONTROL0H	IDE	RTR	—	STDID[10:6]					RCAN-ET（MCR15 = 1）
	STDID[5:0]						EXTID[17:16]		
MB[0]. CONTROL0H	—	STDID[10:4]							RCAN-ET（MCR15 = 0）
	STDID[3:0]				RTR	IDE	EXTID[17:16]		
MB[0]. CONTROL0L	EXTID[15:8]								RCAN-ET（通道 1）*
	EXTID[7:0]								
MB[0]. LAFMH	IDE_LAFM	—	—	STDID_LAFM[10:6]					RCAN-ET（MCR15 = 1）（通道 1）*
	STDID_LAFM[5:0]						EXTID_LAFM[17:16]		
MB[0]. LAFMH	—	STDID_LAFM[10:4]							RCAN-ET（MCR15 = 0）（通道 1）*
	STDID_LAFM[3:0]				—	IDE_LAFM	EXTID_LAFM[17:16]		
MB[0]. LAFML	EXTID_LAFM[15:8]								RCAN-ET（通道 1）*
	EXTID_LAFM[7:0]								
MB[0]. MSG_DATA[0]	MSG_DATA_0								
MB[0]. MSG_DATA[1]	MSG_DATA_1								
MB[0]. MSG_DATA[2]	MSG_DATA_2								
MB[0]. MSG_DATA[3]	MSG_DATA_3								
MB[0]. MSG_DATA[4]	MSG_DATA_4								
MB[0]. MSG_DATA[5]	MSG_DATA_5								
MB[0]. MSG_DATA[6]	MSG_DATA_6								
MB[0]. MSG_DATA[7]	MSG_DATA_7								

寄存器简称	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0	模块
MB[0]. CONTROL1H	—	—	NMC	—	—	MBC[2:0]			RCAN-ET（通道 1）*
MB[0]. CONTROL1L	—	—	—	—	DLC[3:0]				
MB[1]	与 MB[0] 的位结构相同								
MB[2]	与 MB[0] 的位结构相同								
MB[3]	与 MB[0] 的位结构相同								
↓	（重复）								
MB[13]	与 MB[0] 的位结构相同								
MB[14]	与 MB[0] 的位结构相同								
MB[15]	与 MB[0] 的位结构相同								
FRQCR	—	IFC[2:0]			BFC[2:0]			PFC[2]	CPG
	PFC[1:0]		MIFC[2:0]			MPFC[2:0]			
STBCR1	STBY	—	—	—	—	—	—	—	低功耗
STBCR2	MSTP7	MSTP6	—	MSTP4	—	—	—	—	
STBCR3	—	—	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9*	MSTP8	
STBCR4	MSTP23	MSTP22	MSTP21	MSTP20	MSTP19	—	—	—	
STBCR5	—	—	—	—	—	—	MSTP25	MSTP24	
STBCR6	AUDSRST	HIZ	—	—	—	—	STBYMD	—	
WTCNT									WDT
WTCSR	TME	WT/IT	RSTS	WOVF	IOVF	CKS[2:0]			
OSCCR	—	—	—	—	—	OSCSTOP	—	OSCERS	CPG
RAMCR	—	—	—	RAME	—	—	—	—	低功耗
BSCEHR	DTLOCK	—	—	—	—	—	—	—	BSC
	—	—	—	—	—	—	—	—	
ICR0	NMIL	—	—	—	—	—	—	NMIE	INTC
	—	—	—	—	—	—	—	—	
IRQCR	—	—	—	—	—	—	—	—	
	IRQ31S	IRQ30S	IRQ21S	IRQ20S	IRQ11S	IRQ10S	IRQ01S	IRQ00S	
IRQSR	—	—	—	—	IRQ3L	IRQ2L	IRQ1L	IRQ0L	
	—	—	—	—	IRQ3F	IRQ2F	IRQ1F	IRQ0F	
IPRA	IRQ0	IRQ0	IRQ0	IRQ0	IRQ1	IRQ1	IRQ1	IRQ1	
	IRQ2	IRQ2	IRQ2	IRQ2	IRQ3	IRQ3	IRQ3	IRQ3	
IPRD	MTU2_0	MTU2_0	MTU2_0	MTU2_0	MTU2_0	MTU2_0	MTU2_0	MTU2_0	
	MTU2_1	MTU2_1	MTU2_1	MTU2_1	MTU2_1	MTU2_1	MTU2_1	MTU2_1	

寄存器简称	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0	模块
IPRE	MTU2_2	MTU2_2	MTU2_2	MTU2_2	MTU2_2	MTU2_2	MTU2_2	MTU2_2	INTC
	MTU2_3	MTU2_3	MTU2_3	MTU2_3	MTU2_3	MTU2_3	MTU2_3	MTU2_3	
IPRF	MTU2_4	MTU2_4	MTU2_4	MTU2_4	MTU2_4	MTU2_4	MTU2_4	MTU2_4	
	—	—	—	—	POE(MTU2)	POE(MTU2)	POE(MTU2)	POE(MTU2)	
IPRH	—	—	—	—	—	—	—	—	
	MTU2S_3	MTU2S_3	MTU2S_3	MTU2S_3	MTU2S_3	MTU2S_3	MTU2S_3	MTU2S_3	
IPRI	MTU2S_4	MTU2S_4	MTU2S_4	MTU2S_4	MTU2S_4	MTU2S_4	MTU2S_4	MTU2S_4	
	MTU2S_5	MTU2S_5	MTU2S_5	MTU2S_5	POE(MTU2S)	POE(MTU2S)	POE(MTU2S)	POE(MTU2S)	
IPRJ	CMT_0	CMT_0	CMT_0	CMT_0	CMT_1	CMT_1	CMT_1	CMT_1	
	—	—	—	—	WDT	WDT	WDT	WDT	
IPRK	—	—	—	—	—	—	—	—	
	A/D_0	A/D_0	A/D_0	A/D_0	A/D_1	A/D_1	A/D_1	A/D_1	
IPRL	SCI_0	SCI_0	SCI_0	SCI_0	SCI_1	SCI_1	SCI_1	SCI_1	
	SCI_2	SCI_2	SCI_2	SCI_2	—	—	—	—	
IPRM	SSU	SSU	SSU	SSU	—	—	—	—	
	RCAN-ET_0	RCAN-ET_0	RCAN-ET_0	RCAN-ET_0	RCAN-ET_1*	RCAN-ET_1*	RCAN-ET_1*	RCAN-ET_1*	
CMNCR	—	—	—	—	—	—	—	—	BSC
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	HIZMEM	—	
CS0BCR	—	—	IWW[1:0]		—	IWRWD[1:0]		—	
	IWRWS[1:0]		—	IWRRD[1:0]		—	IWRRS[1:0]		
	—	—	—	—	—	BSZ[1:0]		—	
	—	—	—	—	—	—	—	—	
CS1BCR	—	—	IWW[1:0]		—	IWRWD[1:0]		—	
	IWRWS[1:0]		—	IWRRD[1:0]		—	IWRRS[1:0]		
	—	—	—	—	—	BSZ[1:0]		—	
	—	—	—	—	—	—	—	—	
CS0WCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	WW[2:0]			
	—	—	—	SW[1:0]		WR[3:1]			
	WR[0]	WM	—	—	—	—	HW[1:0]		

寄存器简称	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0	模块
CS1WCR	—	—	—	—	—	—	—	—	BSC
	—	—	—	—	—	WW[2:0]			
	—	—	—	SW[1:0]		WR[3:1]			
	WR[0]	WM	—	—	—	—	HW[1:0]		
RAMER	—	—	—	—	—	—	—	—	FLASH
	—	—	—	—	RAMS	RAM[2:0]			
BARA	BAA31	BAA30	BAA29	BAA28	BAA27	BAA26	BAA25	BAA24	UBC
	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16	
	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8	
	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0	
BAMRA	BAMA31	BAMA30	BAMA29	BAMA28	BAMA27	BAMA26	BAMA25	BAMA24	
	BAMA23	BAMA22	BAMA21	BAMA20	BAMA19	BAMA18	BAMA17	BAMA16	
	BAMA15	BAMA14	BAMA13	BAMA12	BAMA11	BAMA10	BAMA9	BAMA8	
	BAMA7	BAMA6	BAMA5	BAMA4	BAMA3	BAMA2	BAMA1	BAMA0	
BBRA	—	—	—	—	—	CPA[2:0]			
	CDA[1:0]		IDA[1:0]		RWA[1:0]		SZA[1:0]		
BDRA	BDA31	BDA30	BDA29	BDA28	BDA27	BDA26	BDA25	BDA24	
	BDA23	BDA22	BDA21	BDA20	BDA19	BDA18	BDA17	BDA16	
	BDA15	BDA14	BDA13	BDA12	BDA11	BDA10	BDA9	BDA8	
	BDA7	BDA6	BDA5	BDA4	BDA3	BDA2	BDA1	BDA0	
BDMRA	BDMA31	BDMA30	BDMA29	BDMA28	BDMA27	BDMA26	BDMA25	BDMA24	
	BDMA23	BDMA22	BDMA21	BDMA20	BDMA19	BDMA18	BDMA17	BDMA16	
	BDMA15	BDMA14	BDMA13	BDMA12	BDMA11	BDMA10	BDMA9	BDMA8	
	BDMA7	BDMA6	BDMA5	BDMA4	BDMA3	BDMA2	BDMA1	BDMA0	
BARB	BAB31	BAB30	BAB29	BAB28	BAB27	BAB26	BAB25	BAB24	
	BAB23	BAB22	BAB21	BAB20	BAB19	BAB18	BAB17	BAB16	
	BAB15	BAB14	BAB13	BAB12	BAB11	BAB10	BAB9	BAB8	
	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1	BAB0	
BAMRB	BAMB31	BAMB30	BAMB29	BAMB28	BAMB27	BAMB26	BAMB25	BAMB24	
	BAMB23	BAMB22	BAMB21	BAMB20	BAMB19	BAMB18	BAMB17	BAMB16	
	BAMB15	BAMB14	BAMB13	BAMB12	BAMB11	BAMB10	BAMB9	BAMB8	
	BAMB7	BAMB6	BAMB5	BAMB4	BAMB3	BAMB2	BAMB1	BAMB0	
BBRB	—	—	—	—	—	CPB[2:0]			
	CDB[1:0]		IDB[1:0]		RWB[1:0]		SZB[1:0]		

寄存器简称	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0	模块
BDRB	BDB31	BDB30	BDB29	BDB28	BDB27	BDB26	BDB25	BDB24	UBC
	BDB23	BDB22	BDB21	BDB20	BDB19	BDB18	BDB17	BDB16	
	BDB15	BDB14	BDB13	BDB12	BDB11	BDB10	BDB9	BDB8	
	BDB7	BDB6	BDB5	BDB4	BDB3	BDB2	BDB1	BDB0	
BDMRB	BDMB31	BDMB30	BDMB29	BDMB28	BDMB27	BDMB26	BDMB25	BDMB24	
	BDMB23	BDMB22	BDMB21	BDMB20	BDMB19	BDMB18	BDMB17	BDMB16	
	BDMB15	BDMB14	BDMB13	BDMB12	BDMB11	BDMB10	BDMB9	BDMB8	
	BDMB7	BDMB6	BDMB5	BDMB4	BDMB3	BDMB2	BDMB1	BDMB0	
BRCR	—	—	—	—	—	—	—	—	
	—	—	UTRGW[1:0]		UBIDB	—	UBIDA	—	
	SCMFCA	SCMFCB	SCMFDA	SCMFDB	PCTE	PCBA	—	—	
	DBEA	PCBB	DBEB	—	SEQ	—	—	ETBE	
BRSR	SVF	—	—	—	BSA27	BSA26	BSA25	BSA24	
	BSA23	BSA22	BSA21	BSA20	BSA19	BSA18	BSA17	BSA16	
	BSA15	BSA14	BSA13	BSA12	BSA11	BSA10	BSA9	BSA8	
	BSA7	BSA6	BSA5	BSA4	BSA3	BSA2	BSA1	BSA0	
BRDR	DVF	—	—	—	BDA27	BDA26	BDA25	BDA24	
	BDA23	BDA22	BDA21	BDA20	BDA19	BDA18	BDA17	BDA16	
	BDA15	BDA14	BDA13	BDA12	BDA11	BDA10	BDA9	BDA8	
	BDA7	BDA6	BDA5	BDA4	BDA3	BDA2	BDA1	BDA0	
BETR	—	—	—	—	BET[11:8]				
	BET[7:0]								
AUCSR	—	—	—	—	—	—	—	—	AUD
	—	—	—	—	—	—	—	—	
	CLK[1:0]		—	—	BRE	OC[1:0]		BR	
	WA[1:0]		WB[1:0]		—	—	TM	EN	
AUWASR									
AUWAER									

寄存器简称	bit 31/23/15/7	bit 30/22/14/6	bit 29/21/13/5	bit 28/20/12/4	bit 27/19/11/3	bit 26/18/10/2	bit 25/17/9/1	bit 24/16/8/0	模块
AUWBSR									AUD
AUWBER									
AUECSR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	WAOB	—	
	—	WBOB	—	—	TREX	TRSB	TRGN	—	

【注】 * 仅适用于 SH7142

24.3　各运行模式的寄存器状态

寄存器简称	上电复位	手动复位	软件待机	深度软件待机	模块待机	睡眠	硬件待机	模块
SCSMR_0	初始化	保持	初始化	初始化	初始化	保持	初始化	SCI（通道 0）
SCBRR_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
SCSCR_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
SCTDR_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
SCSSR_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
SCRDR_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
SCSDCR_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
SCSPTR_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
SCSMR_1	初始化	保持	初始化	初始化	初始化	保持	初始化	SCI（通道 1）
SCBRR_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
SCSCR_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
SCTDR_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
SCSSR_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
SCRDR_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
SCSDCR_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
SCSPTR_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
SCSMR_2	初始化	保持	初始化	初始化	初始化	保持	初始化	SCI（通道 2）
SCBRR_2	初始化	保持	初始化	初始化	初始化	保持	初始化	
SCSCR_2	初始化	保持	初始化	初始化	初始化	保持	初始化	
SCTDR_2	初始化	保持	初始化	初始化	初始化	保持	初始化	
SCSSR_2	初始化	保持	初始化	初始化	初始化	保持	初始化	
SCRDR_2	初始化	保持	初始化	初始化	初始化	保持	初始化	
SCSDCR_2	初始化	保持	初始化	初始化	初始化	保持	初始化	
SCSPTR_2	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCR_3	初始化	保持	初始化	初始化	初始化	保持	初始化	MTU2
TCR_4	初始化	保持	初始化	初始化	初始化	保持	初始化	
TMDR_3	初始化	保持	初始化	初始化	初始化	保持	初始化	
TMDR_4	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIORH_3	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIORL_3	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIORH_4	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIORL_4	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIER_3	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIER_4	初始化	保持	初始化	初始化	初始化	保持	初始化	

寄存器简称	上电复位	手动复位	软件待机	深度软件待机	模块待机	睡眠	硬件待机	模块
TOER	初始化	保持	初始化	初始化	初始化	保持	初始化	MTU2
TGCR	初始化	保持	初始化	初始化	初始化	保持	初始化	
TOCR1	初始化	保持	初始化	初始化	初始化	保持	初始化	
TOCR2	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCNT_3	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCNT_4	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCDR	初始化	保持	初始化	初始化	初始化	保持	初始化	
TDDR	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRA_3	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRB_3	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRA_4	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRB_4	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCNTS	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCBR	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRC_3	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRD_3	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRC_4	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRD_4	初始化	保持	初始化	初始化	初始化	保持	初始化	
TSR_3	初始化	保持	初始化	初始化	初始化	保持	初始化	
TSR_4	初始化	保持	初始化	初始化	初始化	保持	初始化	
TITCR	初始化	保持	初始化	初始化	初始化	保持	初始化	
TITCNT	初始化	保持	初始化	初始化	初始化	保持	初始化	
TBTER	初始化	保持	初始化	初始化	初始化	保持	初始化	
TDER	初始化	保持	初始化	初始化	初始化	保持	初始化	
TOLBR	初始化	保持	初始化	初始化	初始化	保持	初始化	
TBTM_3	初始化	保持	初始化	初始化	初始化	保持	初始化	
TBTM_4	初始化	保持	初始化	初始化	初始化	保持	初始化	
TADCR	初始化	保持	初始化	初始化	初始化	保持	初始化	
TADCORA_4	初始化	保持	初始化	初始化	初始化	保持	初始化	
TADCORB_4	初始化	保持	初始化	初始化	初始化	保持	初始化	
TADCOBRA_4	初始化	保持	初始化	初始化	初始化	保持	初始化	
TADCOBRB_4	初始化	保持	初始化	初始化	初始化	保持	初始化	
TWCR	初始化	保持	初始化	初始化	初始化	保持	初始化	
TSTR	初始化	保持	初始化	初始化	初始化	保持	初始化	
TSYR	初始化	保持	初始化	初始化	初始化	保持	初始化	

寄存器简称	上电复位	手动复位	软件待机	深度软件待机	模块待机	睡眠	硬件待机	模块
TCSYSTR	初始化	保持	初始化	初始化	初始化	保持	初始化	MTU2
TRWER	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCR_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
TMDR_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIORH_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIORL_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIER_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
TSR_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCNT_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRA_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRB_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRC_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRD_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRE_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRF_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIER2_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
TSR2_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
TBTM_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCR_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
TMDR_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIOR_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIER_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
TSR_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCNT_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRA_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRB_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
TICCR	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCR_2	初始化	保持	初始化	初始化	初始化	保持	初始化	
TMDR_2	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIOR_2	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIER_2	初始化	保持	初始化	初始化	初始化	保持	初始化	
TSR_2	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCNT_2	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRA_2	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRB_2	初始化	保持	初始化	初始化	初始化	保持	初始化	

寄存器简称	上电复位	手动复位	软件待机	深度软件待机	模块待机	睡眠	硬件待机	模块
TCR_3S	初始化	保持	初始化	初始化	初始化	保持	初始化	MTU2S
TCR_4S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TMDR_3S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TMDR_4S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIORH_3S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIORL_3S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIORH_4S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIORL_4S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIER_3S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIER_4S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TOERS	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGCRS	初始化	保持	初始化	初始化	初始化	保持	初始化	
TOCR1S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TOCR2S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCNT_3S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCNT_4S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCDRS	初始化	保持	初始化	初始化	初始化	保持	初始化	
TDDRS	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRA_3S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRB_3S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRA_4S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRB_4S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCNTSS	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCBRS	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRC_3S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRD_3S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRC_4S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRD_4S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TSR_3S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TSR_4S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TITCRS	初始化	保持	初始化	初始化	初始化	保持	初始化	
TITCNTS	初始化	保持	初始化	初始化	初始化	保持	初始化	
TBTERS	初始化	保持	初始化	初始化	初始化	保持	初始化	
TDERS	初始化	保持	初始化	初始化	初始化	保持	初始化	
TOLBRS	初始化	保持	初始化	初始化	初始化	保持	初始化	

寄存器简称	上电复位	手动复位	软件待机	深度软件待机	模块待机	睡眠	硬件待机	模块
TBTM_3S	初始化	保持	初始化	初始化	初始化	保持	初始化	MTU2S
TBTM_4S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TADCRS	初始化	保持	初始化	初始化	初始化	保持	初始化	
TADCORA_4S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TADCORB_4S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TADCOBRA_4S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TADCOBRB_4S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TSYCRS	初始化	保持	初始化	初始化	初始化	保持	初始化	
TWCRS	初始化	保持	初始化	初始化	初始化	保持	初始化	
TSTRS	初始化	保持	初始化	初始化	初始化	保持	初始化	
TSYRS	初始化	保持	初始化	初始化	初始化	保持	初始化	
TRWERS	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCNTU_5S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRU_5S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCRU_5S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIORU_5S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCNTV_5S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRV_5S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCRV_5S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIORV_5S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCNTW_5S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TGRW_5S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCRW_5S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIORW_5S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TSR_5S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TIER_5S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TSTR_5S	初始化	保持	初始化	初始化	初始化	保持	初始化	
TCNTCMPCLRS	初始化	保持	初始化	初始化	初始化	保持	初始化	
FCCS	初始化	保持	初始化	初始化	初始化	保持	初始化	FLASH
FPCS	初始化	保持	初始化	初始化	初始化	保持	初始化	
FECS	初始化	保持	初始化	初始化	初始化	保持	初始化	
FKEY	初始化	保持	初始化	初始化	初始化	保持	初始化	
FMATS	初始化	保持	初始化	初始化	初始化	保持	初始化	
FTDAR	初始化	保持	初始化	初始化	初始化	保持	初始化	

寄存器简称	上电复位	手动复位	软件待机	深度软件待机	模块待机	睡眠	硬件待机	模块
DTCERA	初始化	保持	保持	初始化	保持	保持	初始化	DTC
DTCERB	初始化	保持	保持	初始化	保持	保持	初始化	
DTCERC	初始化	保持	保持	初始化	保持	保持	初始化	
DTCERD	初始化	保持	保持	初始化	保持	保持	初始化	
DTCERE	初始化	保持	保持	初始化	保持	保持	初始化	
DTCCR	初始化	保持	保持	初始化	保持	保持	初始化	
DTCVBR	初始化	保持	保持	初始化	保持	保持	初始化	
SSCRH	初始化	保持	初始化	初始化	初始化	保持	初始化	SSU
SSCRL	初始化	保持	初始化	初始化	初始化	保持	初始化	
SSMR	初始化	保持	初始化	初始化	初始化	保持	初始化	
SSER	初始化	保持	初始化	初始化	初始化	保持	初始化	
SSSR	初始化	保持	初始化	初始化	初始化	保持	初始化	
SSCR2	初始化	保持	初始化	初始化	初始化	保持	初始化	
SSTDR0	初始化	保持	初始化	初始化	初始化	保持	初始化	
SSTDR1	初始化	保持	初始化	初始化	初始化	保持	初始化	
SSTDR2	初始化	保持	初始化	初始化	初始化	保持	初始化	
SSTDR3	初始化	保持	初始化	初始化	初始化	保持	初始化	
SSRDR0	初始化	保持	初始化	初始化	初始化	保持	初始化	
SSRDR1	初始化	保持	初始化	初始化	初始化	保持	初始化	
SSRDR2	初始化	保持	初始化	初始化	初始化	保持	初始化	
SSRDR3	初始化	保持	初始化	初始化	初始化	保持	初始化	
CMSTR	初始化	保持	初始化	初始化	初始化	保持	初始化	CMT
CMCSR_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
CMCNT_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
CMCOR_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
CMCSR_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
CMCNT_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
CMCOR_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
ICSR1	初始化	保持	保持	初始化	—	保持	初始化	POE
OCSR1	初始化	保持	保持	初始化	—	保持	初始化	
ICSR2	初始化	保持	保持	初始化	—	保持	初始化	
OCSR2	初始化	保持	保持	初始化	—	保持	初始化	
ICSR3	初始化	保持	保持	初始化	—	保持	初始化	
SPOER	初始化	保持	保持	初始化	—	保持	初始化	
POECR1	初始化	保持	保持	初始化	—	保持	初始化	
POECR2	初始化	保持	保持	初始化	—	保持	初始化	

寄存器简称	上电复位	手动复位	软件待机	深度软件待机	模块待机	睡眠	硬件待机	模块
PADRL	初始化	保持	保持	初始化	—	保持	初始化	I/O
PAIORL	初始化	保持	保持	初始化	—	保持	初始化	PFC
PACRL4	初始化	保持	保持	初始化	—	保持	初始化	
PACRL3	初始化	保持	保持	初始化	—	保持	初始化	
PACRL2	初始化	保持	保持	初始化	—	保持	初始化	
PACRL1	初始化	保持	保持	初始化	—	保持	初始化	
PAPRL	初始化	保持	保持	初始化	—	保持	初始化	I/O
PBDRL	初始化	保持	保持	初始化	—	保持	初始化	
PBIORL	初始化	保持	保持	初始化	—	保持	初始化	PFC
PBCRL2	初始化	保持	保持	初始化	—	保持	初始化	
PBCRL1	初始化	保持	保持	初始化	—	保持	初始化	
PBPRL	初始化	保持	保持	初始化	—	保持	初始化	I/O
PDDRL	初始化	保持	保持	初始化	—	保持	初始化	
PDIORL	初始化	保持	保持	初始化	—	保持	初始化	PFC
PDCRL3	初始化	保持	保持	初始化	—	保持	初始化	
PDCRL2	初始化	保持	保持	初始化	—	保持	初始化	
PDCRL1	初始化	保持	保持	初始化	—	保持	初始化	
PDPRL	初始化	保持	保持	初始化	—	保持	初始化	I/O
PEDRH	初始化	保持	保持	初始化	—	保持	初始化	
PEDRL	初始化	保持	保持	初始化	—	保持	初始化	
PEIORH	初始化	保持	保持	初始化	—	保持	初始化	PFC
PEIORL	初始化	保持	保持	初始化	—	保持	初始化	
PECRH2	初始化	保持	保持	初始化	—	保持	初始化	
PECRH1	初始化	保持	保持	初始化	—	保持	初始化	
PECRL4	初始化	保持	保持	初始化	—	保持	初始化	
PECRL3	初始化	保持	保持	初始化	—	保持	初始化	
PECRL2	初始化	保持	保持	初始化	—	保持	初始化	
PECRL1	初始化	保持	保持	初始化	—	保持	初始化	
PEPRH	初始化	保持	保持	初始化	—	保持	初始化	I/O
PEPRL	初始化	保持	保持	初始化	—	保持	初始化	
ADCR_0	初始化	保持	初始化	初始化	初始化	保持	初始化	A/D（通道 0）
ADSR_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
ADSTRGR_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
ADANSR_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
ADDR0	初始化	保持	初始化	初始化	初始化	保持	初始化	

寄存器简称	上电复位	手动复位	软件待机	深度软件待机	模块待机	睡眠	硬件待机	模块
ADDR1	初始化	保持	初始化	初始化	初始化	保持	初始化	A/D（通道 0）
ADDR2	初始化	保持	初始化	初始化	初始化	保持	初始化	
ADDR3	初始化	保持	初始化	初始化	初始化	保持	初始化	
ADDR4	初始化	保持	初始化	初始化	初始化	保持	初始化	
ADDR5	初始化	保持	初始化	初始化	初始化	保持	初始化	
ADDR6	初始化	保持	初始化	初始化	初始化	保持	初始化	
ADDR7	初始化	保持	初始化	初始化	初始化	保持	初始化	
ADCR_1	初始化	保持	初始化	初始化	初始化	保持	初始化	A/D（通道 1）
ADSR_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
ADSTRGR_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
ADANSR_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
ADDR8	初始化	保持	初始化	初始化	初始化	保持	初始化	
ADDR9	初始化	保持	初始化	初始化	初始化	保持	初始化	
ADDR10	初始化	保持	初始化	初始化	初始化	保持	初始化	
ADDR11	初始化	保持	初始化	初始化	初始化	保持	初始化	
ADDR12	初始化	保持	初始化	初始化	初始化	保持	初始化	
ADDR13	初始化	保持	初始化	初始化	初始化	保持	初始化	
ADDR14	初始化	保持	初始化	初始化	初始化	保持	初始化	
ADDR15	初始化	保持	初始化	初始化	初始化	保持	初始化	
MCR_0	初始化	保持	初始化	初始化	初始化	保持	初始化	RCAN-ET（通道 0）
GSR_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
BCR1_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
BCR0_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
IRR_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
IMR_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
TEC/REC_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
TXPR1_0、0_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
TXCR0_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
TXACK0_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
ABACK0_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
RXPR0_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
RFPR0_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
MBIMR0_0	初始化	保持	初始化	初始化	初始化	保持	初始化	
UMSR0_0	初始化	保持	初始化	初始化	初始化	保持	初始化	

寄存器简称	上电复位	手动复位	软件待机	深度软件待机	模块待机	睡眠	硬件待机	模块
MB[0]. CONTROL0H	—	保持	—	—	—	保持	—	RCAN-ET（通道 0）
MB[0]. CONTROL0L	—	保持	—	—	—	保持	—	
MB[0]. LAFMH	—	保持	—	—	—	保持	—	
MB[0]. LAFML	—	保持	—	—	—	保持	—	
MB[0]. MSG_DATA[0]	—	保持	—	—	—	保持	—	
MB[0]. MSG_DATA[1]	—	保持	—	—	—	保持	—	
MB[0]. MSG_DATA[2]	—	保持	—	—	—	保持	—	
MB[0]. MSG_DATA[3]	—	保持	—	—	—	保持	—	
MB[0]. MSG_DATA[4]	—	保持	—	—	—	保持	—	
MB[0]. MSG_DATA[5]	—	保持	—	—	—	保持	—	
MB[0]. MSG_DATA[6]	—	保持	—	—	—	保持	—	
MB[0]. MSG_DATA[7]	—	保持	—	—	—	保持	—	
MB[0]. CONTROL1H	初始化	保持	初始化	初始化	初始化	保持	初始化	
MB[0]. CONTROL1L	初始化	保持	初始化	初始化	初始化	保持	初始化	
MB[1]	与 MB[0] 相同							
MB[2]	与 MB[0] 相同							
MB[3]	与 MB[0] 相同							
↓	（重复）							
MB[13]	与 MB[0] 相同							
MB[14]	与 MB[0] 相同							
MB[15]	与 MB[0] 相同							
MCR_1	初始化	保持	初始化	初始化	初始化	保持	初始化	RCAN-ET（通道 1） *1
GSR_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
BCR1_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
BCR0_1	初始化	保持	初始化	初始化	初始化	保持	初始化	

寄存器简称	上电复位	手动复位	软件待机	深度软件待机	模块待机	睡眠	硬件待机	模块
IRR_1	初始化	保持	初始化	初始化	初始化	保持	初始化	RCAN-ET（通道 1）*1
IMR_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
TEC/REC_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
TXPR1_1、0_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
TXCR0_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
TXACK0_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
ABACK0_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
RXPR0_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
RFPR0_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
MBIMR0_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
UMSR0_1	初始化	保持	初始化	初始化	初始化	保持	初始化	
MB[0]. CONTROL0H	—	保持	—	—	—	保持	—	
MB[0]. CONTROL0L	—	保持	—	—	—	保持	—	
MB[0]. LAFMH	—	保持	—	—	—	保持	—	
MB[0]. LAFML	—	保持	—	—	—	保持	—	
MB[0]. MSG_DATA[0]	—	保持	—	—	—	保持	—	
MB[0]. MSG_DATA[1]	—	保持	—	—	—	保持	—	
MB[0]. MSG_DATA[2]	—	保持	—	—	—	保持	—	
MB[0]. MSG_DATA[3]	—	保持	—	—	—	保持	—	
MB[0]. MSG_DATA[4]	—	保持	—	—	—	保持	—	
MB[0]. MSG_DATA[5]	—	保持	—	—	—	保持	—	
MB[0]. MSG_DATA[6]	—	保持	—	—	—	保持	—	
MB[0]. MSG_DATA[7]	—	保持	—	—	—	保持	—	
MB[0]. CONTROL1H	初始化	保持	初始化	初始化	初始化	保持	初始化	
MB[0]. CONTROL1L	初始化	保持	初始化	初始化	初始化	保持	初始化	

寄存器简称	上电复位	手动复位	软件待机	深度软件待机	模块待机	睡眠	硬件待机	模块
MB[1]	与 MB[0] 相同							RCAN-ET（通道 1）*1
MB[2]	与 MB[0] 相同							
MB[3]	与 MB[0] 相同							
↓	（重复）							
MB[13]	与 MB[0] 相同							
MB[14]	与 MB[0] 相同							
MB[15]	与 MB[0] 相同							
FRQCR	初始化 *2	保持	保持	初始化	—	保持	初始化	CPG
STBCR1	初始化	保持	保持	初始化	—	保持	初始化	低功耗
STBCR2	初始化	保持	保持	初始化	—	保持	初始化	
STBCR3	初始化	保持	保持	初始化	—	保持	初始化	
STBCR4	初始化	保持	保持	初始化	—	保持	初始化	
STBCR5	初始化	保持	保持	初始化	—	保持	初始化	
STBCR6	初始化	保持	保持	初始化	—	保持	初始化	
WTCNT	初始化 *2	保持	保持	初始化	—	保持	初始化	WDT
WTCSR	初始化 *2	保持	保持	初始化	—	保持	初始化	
OSCCR	初始化 *3	保持	保持 *4	初始化	—	保持	初始化	CPG
RAMCR	初始化	保持	保持	初始化	—	保持	初始化	低功耗
BSCEHR	初始化	保持	保持	初始化	—	保持	初始化	BSC
ICR0	初始化	初始化	保持	初始化	—	保持	初始化	INTC
IRQCR	初始化	初始化	保持	初始化	—	保持	初始化	
IRQSR	初始化	初始化	保持	初始化	—	保持	初始化	
IPRA	初始化	初始化	保持	初始化	—	保持	初始化	
IPRD	初始化	初始化	保持	初始化	—	保持	初始化	
IPRE	初始化	初始化	保持	初始化	—	保持	初始化	
IPRF	初始化	初始化	保持	初始化	—	保持	初始化	
IPRH	初始化	初始化	保持	初始化	—	保持	初始化	
IPRI	初始化	初始化	保持	初始化	—	保持	初始化	
IPRJ	初始化	初始化	保持	初始化	—	保持	初始化	
IPRK	初始化	初始化	保持	初始化	—	保持	初始化	
IPRL	初始化	初始化	保持	初始化	—	保持	初始化	
IPRM	初始化	初始化	保持	初始化	—	保持	初始化	
CMNCR	初始化	保持	保持	初始化	—	保持	初始化	BSC
CS0BCR	初始化	保持	保持	初始化	—	保持	初始化	
CS1BCR	初始化	保持	保持	初始化	—	保持	初始化	

寄存器简称	上电复位	手动复位	软件待机	深度软件待机	模块待机	睡眠	硬件待机	模块
CS0WCR	初始化	保持	保持	初始化	—	保持	初始化	BSC
CS1WCR	初始化	保持	保持	初始化	—	保持	初始化	
RAMER	初始化	初始化	保持	初始化	保持	保持	初始化	FLASH
BARA	初始化	保持	保持	初始化	初始化	保持	初始化	UBC
BAMRA	初始化	保持	保持	初始化	初始化	保持	初始化	
BBRA	初始化	保持	保持	初始化	初始化	保持	初始化	
BDRA	初始化	保持	保持	初始化	初始化	保持	初始化	
BDMRA	初始化	保持	保持	初始化	初始化	保持	初始化	
BARB	初始化	保持	保持	初始化	初始化	保持	初始化	
BAMRB	初始化	保持	保持	初始化	初始化	保持	初始化	
BBRB	初始化	保持	保持	初始化	初始化	保持	初始化	
BDRB	初始化	保持	保持	初始化	初始化	保持	初始化	
BDMRB	初始化	保持	保持	初始化	初始化	保持	初始化	
BRCR	初始化	保持	保持	初始化	初始化	保持	初始化	
BRSR	初始化	初始化	保持	初始化	初始化	保持	初始化	
BRDR	初始化	初始化	保持	初始化	初始化	保持	初始化	
BETR	初始化	保持	保持	初始化	初始化	保持	初始化	
AUCSR	初始化	初始化	保持	初始化	保持	保持	初始化	AUD
AUWASR	初始化	初始化	保持	初始化	保持	保持	初始化	
AUWAER	初始化	初始化	保持	初始化	保持	保持	初始化	
AUWBSR	初始化	初始化	保持	初始化	保持	保持	初始化	
AUWBER	初始化	初始化	保持	初始化	保持	保持	初始化	
AUECSR	初始化	初始化	保持	初始化	保持	保持	初始化	

【注】 *1 仅适用于 SH7142。

*2 通过 WDT 上电复位时不可初始化。

*3 通过 WDT 上电复位时，不可初始化 OSCSTOP 位。

*4 初始化 OSCSTOP 位。

第 25 章　电特性

25.1　绝对最大额定值

绝对最大额定值如表 25.1 所示。

表 25.1　绝对最大额定值

项　目		符号	额定值	单位
电源电压（内部）	标准产品	V_{CC}	–0.3 ～ +7.0	V
	宽温度范围产品		–0.3 ～ +4.3	V
电源电压（I/O）		PV_{CC}	–0.3 ～ +7.0	V
输入电压（除模拟输入引脚之外）		V_{in}	–0.3 ～ PV_{CC}+0.3	V
模拟电源电压		AV_{CC}	–0.3 ～ +7.0	V
模拟基准电压		AV_{refh}	–0.3 ～ AV_{CC}+0.3	V
模拟输入电压		V_{an}	–0.3 ～ AV_{CC}+0.3	V
工作温度 （除闪存的 W/E 之外）	标准产品	T_{opr}	–40 ～ +85	°C
	宽温度范围产品		–40 ～ +125	°C
工作温度（闪存的 W/E）		TWE_{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，可能对LSI造成永久性破坏。

25.2　DC 特性

DC 特性如表 25.2、表 25.3 所示。

表 25.2　DC 特性（标准产品）

条件（标准产品）：V_{CC}=4.5V ～ 5.5V、PV_{CC}=4.5V ～ 5.5V、AV_{CC}=4.5V ～ 5.5V、AV_{refh}=4.5V ～ AV_{CC}、V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、T_a=–40 ～ +85°C

项　目		符号	Min.	Typ.	Max.	单位	测定条件
输入高电平电压（除施密特触发输入引脚之外）	$\overline{RES}$、$\overline{HSTBY}$、NMI、FWE、MD1、MD0、EXTAL	V_{IH}	PV_{CC}–0.6	—	PV_{CC}+0.3	V	
	模拟用引脚		2.2	—	AV_{CC}+0.3	V	
	其他输入引脚		2.2	—	PV_{CC}+0.3	V	
输入低电平电压（除施密特触发输入引脚之外）	$\overline{RES}$、$\overline{HSTBY}$、NMI、FWE、MD1、MD0、EXTAL	V_{IL}	–0.3	—	0.4	V	
	其他输入引脚		–0.3	—	0.8	V	

项 目		符号	Min.	Typ.	Max.	单位	测定条件
施密特触发输入电压	IRQ3 ～ IRQ0、 $\overline{\text{POE0}}$ ～ $\overline{\text{POE2}}$、 $\overline{\text{POE4}}$ ～ $\overline{\text{POE6}}$、 $\overline{\text{POE8}}$、 TCLKA ～ TCLKD、 TIOC0A ～ TIOC0D、 TIOC1A、 TIOC1B、 TIOC2A、 TIOC2B、 TIOC3A ～ TIOC3D、 TIOC4A ～ TIOC4D、 TIOC3BS、 TIOC3DS、 TIOC4AS ～ TIOC4DS、 TIC5US、 TIC5VS、 TIC5WS、 SCK0 ～ SCK2、 RXD0 ～ RXD2、 SSCK、 $\overline{\text{SCS}}$、 SSI、 SSO	V_T^+	PV_{CC}– 0.5	—	—	V	
		V_T^-	—	—	1.0	V	
		$V_T^+ - V_T^-$	0.4	—	—	V	
输入泄漏电流	所有输入引脚 （除 $\overline{\text{HSTBY}}$ 之外）	$\lvert I_{in} \rvert$	—	—	1.0	μA	
输入上拉 MOS 电流	$\overline{\text{HSTBY}}$	$-I_{pu}$	—	—	800	μA	V_{in}=0V
三态泄漏电流 （断态）	端口 A、 B、 D、 E	$\lvert I_{tsi} \rvert$	—	—	1.0	μA	
输出高电平电压	所有输出引脚	V_{OH}	PV_{CC} – 0.8	—	—	V	I_{OH}= –2mA
			PV_{CC}– 0.5	—	—	V	I_{OH}= –200μA
输出低电平电压	所有输出引脚	V_{OL}	—	—	0.5	V	I_{OL}=2mA
			—	—	0.4	V	I_{OL}=1.6mA
输入电容	所有输入引脚	C_{in}	—	—	20	pF	V_{in}=0V、 f=1MHz、 T_a=25°C
消耗电流	普通运行时	I_{CC}	—	100	135	mA	Iϕ=80MHz Bϕ=40MHz Pϕ=40MHz MPϕ=40MHz MIϕ=80MHz
	睡眠时	I_{CC}	—	80	120	mA	Bϕ=40MHz Pϕ=40MHz MPϕ=40MHz MIϕ=80MHz
	软件待机时	I_{CC}	—	40	60	mA	$T_a \leqslant$ 50°C
		I_{CC}	—	—	70	mA	50°C ＜ T_a
	深度软件待机时及硬件待机时	I_{CC}	—	6	30	μA	$T_a \leqslant$ 50°C
		I_{CC}	—	—	100	μA	50°C ＜ T_a
模拟电源电流	A/D 转换过程中	AI_{CC}	—	9	15	mA	包含 AI_{refh}
	A/D 转换待机时		—	—	3	mA	
	待机时		—	—	150	μA	

【使用时的注意事项】

1. 不使用 A/D 转换器时，不得将 AV_{CC}、 AV_{SS}、 AV_{refh}、 AV_{refl} 引脚设置为开路。
2. V_{IH}(Min.)=PV_{CC} –0.5V、 V_{IL}(Max.)=0.5V 的条件下，消耗电流值是所有输出引脚均为无负载状态时的值。

表 25.3 DC 特性 （宽温度范围产品）

条件 （宽温度范围产品）：V_{CC}=3.0V ～ 3.6V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、
V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +125°C

项 目		符号	Min.	Typ.	Max.	单位	测定条件
输入高电平电压（除施密特触发输入引脚之外）	$\overline{RES}$、$\overline{HSTBY}$、NMI、FWE、MD1、MD0、EXTAL	V_{IH}	PV_{CC}–0.6	—	PV_{CC}+0.3	V	
	模拟用引脚		2.2	—	AV_{CC}+0.3	V	
	其他输入引脚		2.2	—	PV_{CC}+0.3	V	
输入低电平电压（除施密特触发输入引脚之外）	$\overline{RES}$、$\overline{HSTBY}$、NMI、FWE、MD1、MD0、EXTAL	V_{IL}	–0.3	—	0.4	V	
	其他输入引脚		–0.3	—	0.8	V	
施密特触发输入电压	IRQ3 ～ IRQ0、$\overline{POE0}$ ～ $\overline{POE2}$、$\overline{POE4}$ ～ $\overline{POE6}$、$\overline{POE8}$、TCLKA ～ TCLKD、TIOC0A ～ TIOC0D、TIOC1A、TIOC1B、TIOC2A、TIOC2B、TIOC3A ～ TIOC3D、TIOC4A ～ TIOC4D、TIOC3BS、TIOC3DS、TIOC4AS ～ TIOC4DS、TIC5US、TIC5VS、TIC5WS、SCK0 ～ SCK2、RXD0 ～ RXD2、SSCK、$\overline{SCS}$、SSI、SSO	V_T^+	PV_{CC}–0.5	—	—	V	
		V_T^-	—	—	1.0	V	
		$V_T^+ - V_T^-$	0.4	—	—	V	
输入泄漏电流	所有输入引脚（除 $\overline{HSTBY}$ 之外）	$\lvert I_{in} \rvert$	—	—	1.0	μA	
输入上拉 MOS 电流	$\overline{HSTBY}$	$-I_{pu}$	—	—	800	μA	V_{in}=0V
三态泄漏电流（断态）	端口 A、B、D、E	$\lvert I_{tsi} \rvert$	—	—	1.0	μA	
输出高电平电压	所有输出引脚	V_{OH}	PV_{CC} – 0.8	—	—	V	I_{OH}= –2mA
			PV_{CC} – 0.5	—	—	V	I_{OH}= –200μA
输出低电平电压	所有输出引脚	V_{OL}	—	—	0.5	V	I_{OL}=2mA
			—	—	0.4	V	I_{OL}=1.6mA
输入电容	所有输入引脚	C_{in}	—	—	20	pF	V_{in}=0V、f=1MHz、T_a=25°C

项 目		符号	Min.	Typ.	Max.	单位	测定条件
消耗电流	普通运行时	I_{CC}	—	83	116	mA	Iϕ=64MHz Bϕ=32MHz Pϕ=32MHz MPϕ=32MHz MIϕ=64MHz
	睡眠时	I_{CC}	—	65	105	mA	Bϕ=32MHz Pϕ=32MHz MPϕ=32MHz MIϕ=64MHz
	软件待机时	I_{CC}	—	40	60	mA	$T_a \leqslant 50°C$
		I_{CC}	—	—	70	mA	$50℃ < T_a$
	深度软件待机及硬件待机时	I_{CC}	—	6	30	μA	$T_a \leqslant 50°C$
		I_{CC}	—	—	100	μA	$50°C < T_a$
模拟电源电流	A/D 转换过程中	AI_{CC}	—	8	13	mA	包含 AI_{refh}
	A/D 转换待机时		—	—	3	mA	
	待机时		—	—	150	μA	

【使用时的注意事项】

1. 不使用 A/D 转换器时，不得将 AV_{CC}、AV_{SS}、AV_{refh}、AV_{refl} 引脚设置为开路。
2. V_{IH}(Min.)=PV_{CC}–0.5V、V_{IL}(Max.)=0.5V 的条件下，消耗电流值是所有输出引脚均为无负载状态时的值。
3. 为了提高本 LSI 的散热性，必须使用 4 层以上的电路板。

表 25.4 输出容许电流值

条件 （标准产品）：V_{CC}=4.5V ～ 5.5V、PV_{CC}=4.5V ～ 5.5V、AV_{CC}=4.5V ～ 5.5V、AV_{refh}=4.5V ～ AV_{CC}、V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、T_a=–40 ～ +85°C

条件 （宽温度范围产品）：V_{CC}=3.0V ～ 3.6V、PV_{CC}=4.5V ～ 5.5V、AV_{CC}=4.5V ～ 5.5V、AV_{refh}=4.5V ～ AV_{CC}、V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、T_a=–40 ～ +125°C

项 目	符号	Min.	Typ.	Max.	单位
输出低电平容许电流 （每个引脚）	I_{OL}	—	—	2.0	mA
输出低电平容许电流 （总合）	ΣI_{OL}	—	—	80	mA
输出高电平容许电流 （每个引脚）	$-I_{OH}$	—	—	2.0	mA
输出高电平容许电流 （总合）	$\Sigma -I_{OH}$	—	—	25	mA

【使用时的注意事项】为了确保 LSI 的可靠性，输出电流值不得超过表 25.4 的值。

25.3 AC 特性

原则上本 LSI 的输入为时钟同步输入，除非有特别说明，否则必须遵守各输入信号的准备 / 保持时间。

表 25.5 最大工作频率

条件 （标准产品）：V_{CC}=4.5V ～ 5.5V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、 V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +85°C

条件 （宽温度范围产品）：V_{CC}=3.0V ～ 3.6V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、 V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +125°C

项 目			符号	Min.	Typ.	Max.	单位	备注
工作频率	CPU （Iϕ）	标准产品	f	10	—	80	MHz	
		宽温度范围产品		10	—	64		
	外部总线 （Bϕ）	标准产品		10	—	40		
		宽温度范围产品		10	—	32		
	外围模块 （Pϕ）	标准产品		10	—	40		
		宽温度范围产品		10	—	32		
	MTU2 （MPϕ）	标准产品		10	—	40		
		宽温度范围产品		10	—	32		
	MTU2S （MIϕ）	标准产品		10	—	80		
		宽温度范围产品		10	—	64		

25.3.1 时钟时序

表 25.6 时钟时序

条件 （标准产品）：V_{CC}=4.5V ～ 5.5V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、 V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +85°C

条件 （宽温度范围产品）：V_{CC}=3.0V ～ 3.6V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、 V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +125°C

项 目		符号	Min.	Max.	单位	参照图
EXTAL 时钟输入频率		f_{EX}	8	10	MHz	图 25.1
EXTAL 时钟输入周期时间		t_{EXcyc}	100	125	ns	
EXTAL 时钟输入低电平脉宽		t_{EXL}	30	—	ns	
EXTAL 时钟输入高电平脉宽		t_{EXH}	30	—	ns	
EXTAL 时钟输入上升时间		t_{EXr}	—	5	ns	
EXTAL 时钟输入下降时间		t_{EXf}	—	5	ns	
CK 时钟输出频率	标准产品	f_{OP}	16	40	MHz	图 25.2
	宽温度范围产品		16	32	MHz	
CK 时钟输出周期时间	标准产品	t_{cyc}	25	62.5	ns	
	宽温度范围产品		31.25	62.5	ns	
CK 时钟输出低电平脉宽		t_{CKL}	1/2 t_{cyc}–7.5	—	ns	
CK 时钟输出高电平脉宽		t_{CKH}	1/2 t_{cyc}–7.5	—	ns	
CK 时钟输出上升时间		t_{CKr}	—	5	ns	
CK 时钟输出下降时间		t_{CKf}	—	5	ns	
上电振荡稳定时间		t_{OSC1}	10	—	ms	图 25.3
待机恢复振荡稳定时间 1		t_{OSC2}	10	—	ms	图 25.4
待机恢复振荡稳定时间 2		t_{OSC3}	10	—	ms	图 25.5

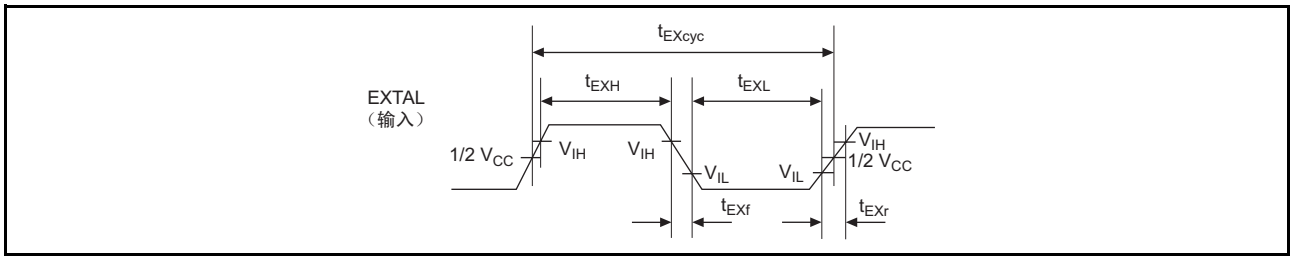


图 25.1 EXTAL 时钟输入时序

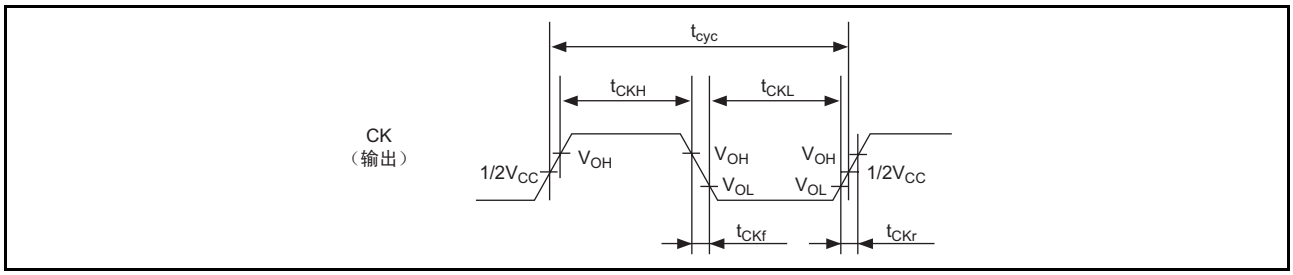


图 25.2 CK 时钟输出时序

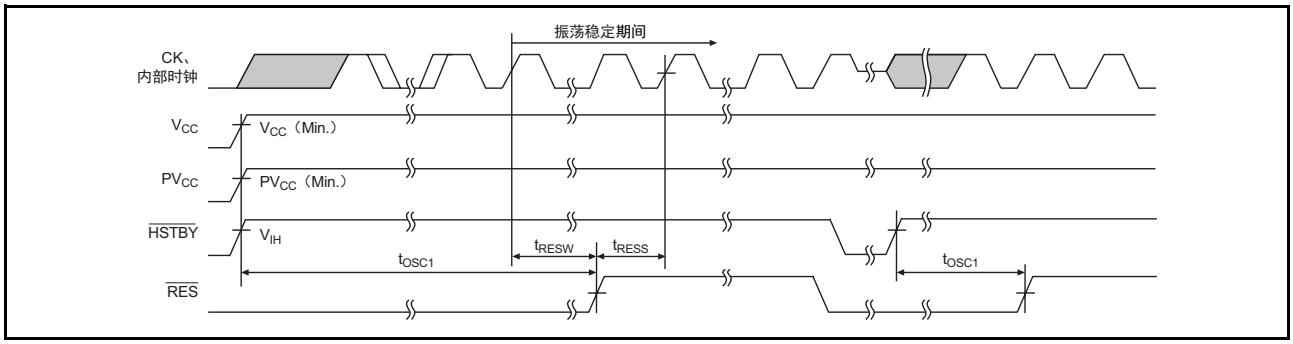


图 25.3 上电振荡稳定时间

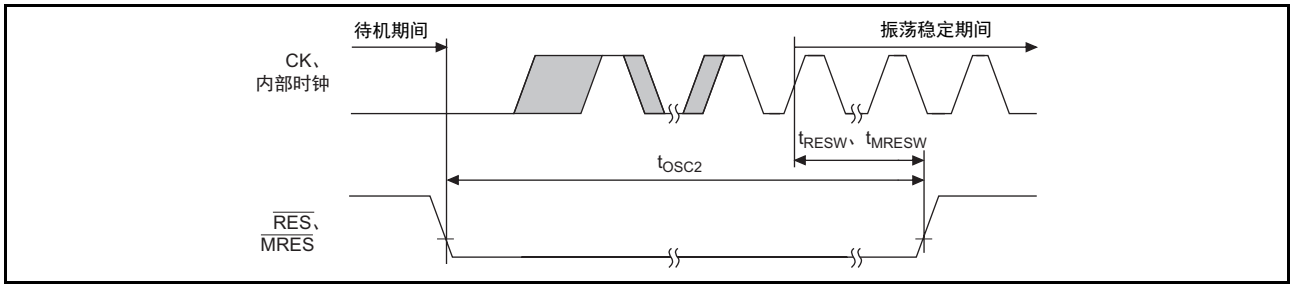


图 25.4 待机恢复时振荡稳定时间 （通过复位恢复）

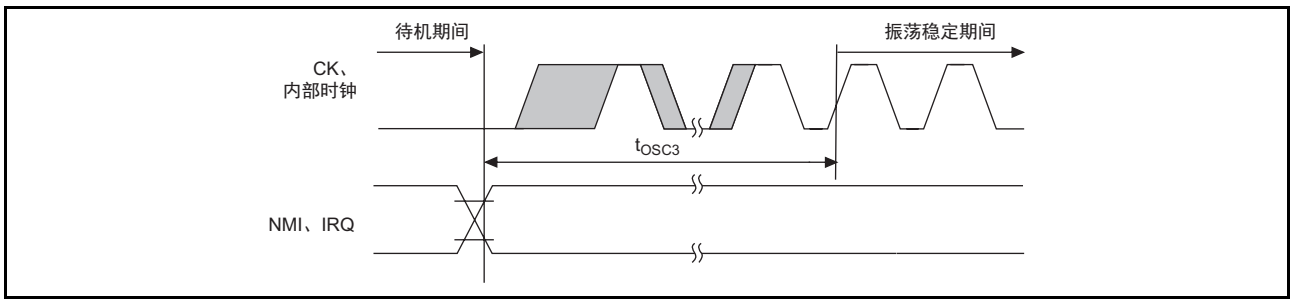


图 25.5 待机恢复时振荡稳定时间 （通过 NMI、 IRQ 恢复）

25.3.2　控制信号时序

表 25.7　控制信号时序

条件 （标准产品）：V_{CC}=4.5V ～ 5.5V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、 V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +85°C

条件 （宽温度范围产品）：V_{CC}=3.0V ～ 3.6V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、 V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +125°C

项　　目	符号	Min.	Max.	单位	参照图
$\overline{\text{RES}}$ 脉宽	t_{RESW}	20*2	—	t_{Bcyc}*4	图 25.3 图 25.4 图 25.6 图 25.7
$\overline{\text{RES}}$ 准备时间 *1	t_{RESS}	65	—	ns	
$\overline{\text{RES}}$ 保持时间	t_{RESH}	15	—	ns	
$\overline{\text{MRES}}$ 脉宽	t_{MRESW}	20*3	—	t_{Bcyc}*4	
$\overline{\text{MRES}}$ 准备时间 *1	t_{MRESS}	25	—	ns	
$\overline{\text{MRES}}$ 保持时间	t_{MRESH}	15	—	ns	
MD1、 MD0、 FWE 准备时间	t_{MDS}	20	—	t_{Bcyc}*4	图 25.6
$\overline{\text{BREQ}}$ 准备时间	t_{BREQS}	$1/2t_{Bcyc}$+15	—	ns	图 25.9
$\overline{\text{BREQ}}$ 保持时间	t_{BREQH}	$1/2t_{Bcyc}$+10	—	ns	
NMI 准备时间 *1	t_{NMIS}	60	—	ns	图 25.7
NMI 保持时间	t_{NMIH}	10	—	ns	
IRQ3 ～ IRQ0 准备时间 *1	t_{IRQS}	35	—	ns	
IRQ3 ～ IRQ0 保持时间	t_{IRQH}	35	—	ns	
$\overline{\text{IRQOUT}}$ 输出延迟时间	t_{IRQOD}	—	100	ns	图 25.8
$\overline{\text{BACK}}$ 延迟时间	t_{BACKD}	—	$1/2t_{Bcyc}$+20	ns	图 25.9 图 25.10
总线三态延迟时间	t_{BOFF}	0	100	ns	
总线缓冲器 ON 时间	t_{BON}	0	100	ns	

【注】 *1 $\overline{\text{RES}}$、 $\overline{\text{MRES}}$、 NMI、 $\overline{\text{BREQ}}$ 及 IRQ3 ～ IRQ0 为异步信号。遵守上述准备时间，可在时钟上升时检测出变化。如未遵守准备时间，可能延迟到下一个时钟上升沿检测出变化。

*2 待机模式时， t_{RESW}=t_{OSC2} （10ms）。

*3 待机模式时， t_{MRESW}=t_{OSC2} （10ms）。

*4 t_{Bcyc} 表示外部总线时钟 （Bϕ=CK）的周期。

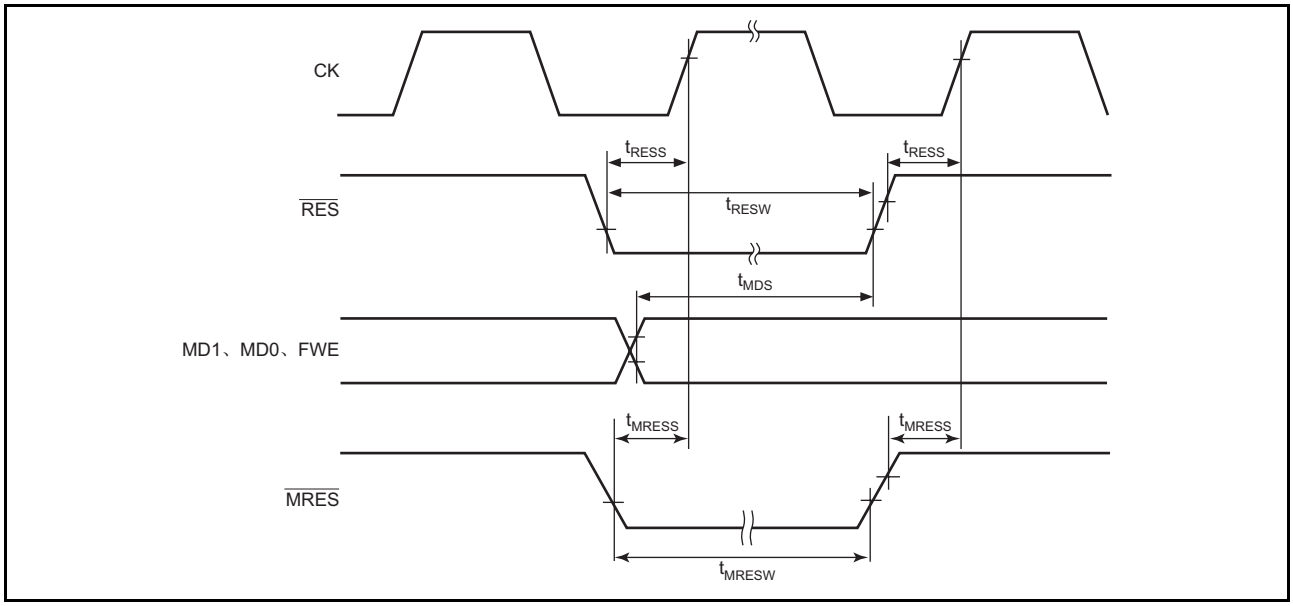


图 25.6 复位输入时序

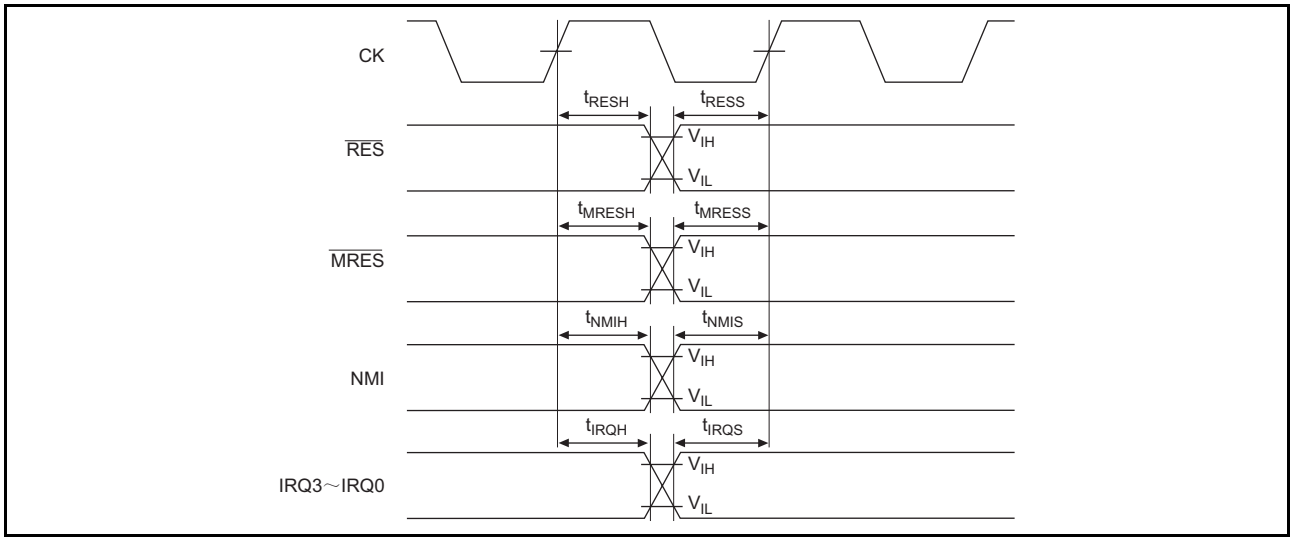


图 25.7 中断信号输入时序

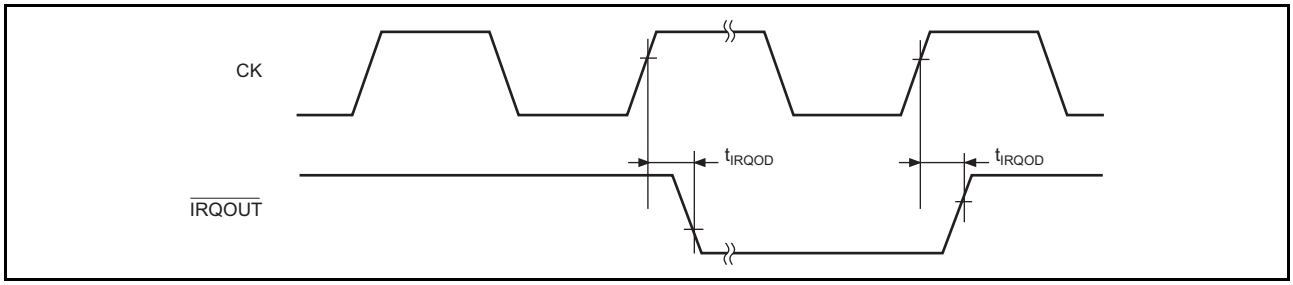


图 25.8 中断信号输出时序

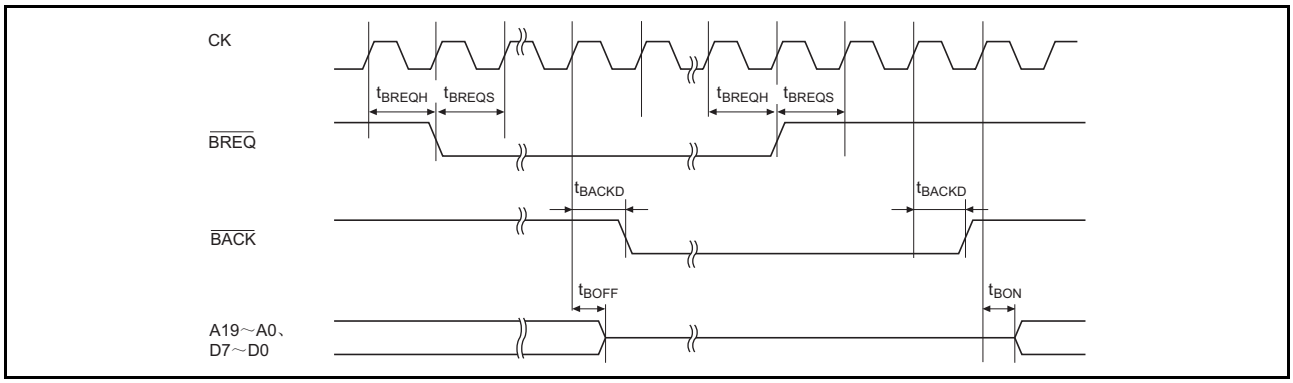


图 25.9 总线权释放时序

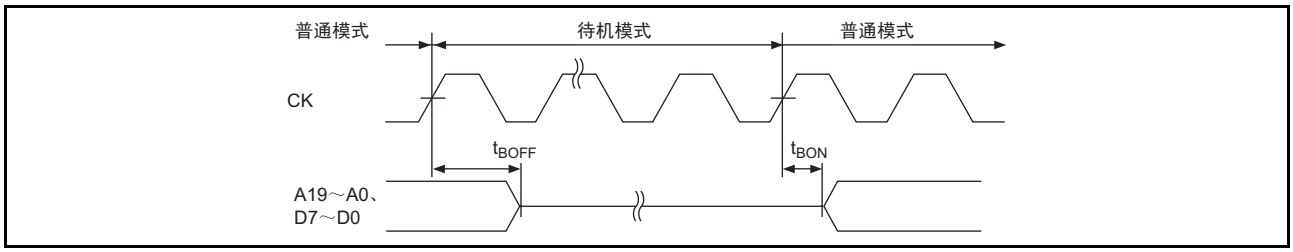


图 25.10 待机时引脚驱动时序

25.3.3　AC 总线时序规格

表 25.8　总线时序

条件 （标准产品）：V_{CC}=4.5V ～ 5.5V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +85°C

条件 （宽温度范围产品）：V_{CC}=3.0V ～ 3.6V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +125°C

项　　目	符号	Min.	Max.	单位	参照图
地址延迟时间 1	t_{AD1}	1	30	ns	图 25.11 ～ 25.15
地址准备时间	t_{AS}	0	—	ns	图 25.11 ～ 25.14
地址保持时间	t_{AH}	0	—	ns	图 25.11 ～ 25.14
$\overline{CS}$ 延迟时间	t_{CSD}	1	30	ns	图 25.11 ～ 25.15
读取 / 写入延迟时间	t_{RWD}	1	30	ns	图 25.11 ～ 25.15
读取选通延迟时间	t_{RSD}	$1/2t_{Bcyc}$+1	$1/2t_{Bcyc}$+30	ns	图 25.11 ～ 25.15
读取数据准备时间 1	t_{RDS1}	$1/2t_{Bcyc}$+30	—	ns	图 25.11 ～ 25.15
读取数据保持时间 1	t_{RDH1}	0	—	ns	图 25.11 ～ 25.15
读取数据存取时间	t_{ACC}*2	t_{Bcyc} ×(n+1.5)–35*1	—	ns	图 25.11 ～ 25.15
读取选通的存取时间	t_{OE}*2	t_{Bcyc} ×(n+1)–35*1	—	ns	图 25.11 ～ 25.15
写入选通延迟时间 1	t_{WSD1}	$1/2t_{Bcyc}$+1	$1/2t_{Bcyc}$+30	ns	图 25.11 ～ 25.15
写入数据延迟时间 1	t_{WDD1}	—	30	ns	图 25.11 ～ 25.15
写入数据保持时间 1	t_{WDH1}	1	—	ns	图 25.11 ～ 25.15
写入数据保持时间	t_{WRH}	0	—	ns	图 25.11 ～ 25.14
$\overline{WAIT}$ 准备时间	t_{WTS}	$1/2t_{Bcyc}$+20	—	ns	图 25.12 ～ 25.15
$\overline{WAIT}$ 保持时间	t_{WTH}	$1/2t_{Bcyc}$+20	—	ns	图 25.12 ～ 25.15

【注】 t_{Bcyc} 表示外部总线时钟 （Bϕ=CK）的周期。

*1　n 为等待数

*2　满足存取时间，则不必满足 t_{RDS1}。

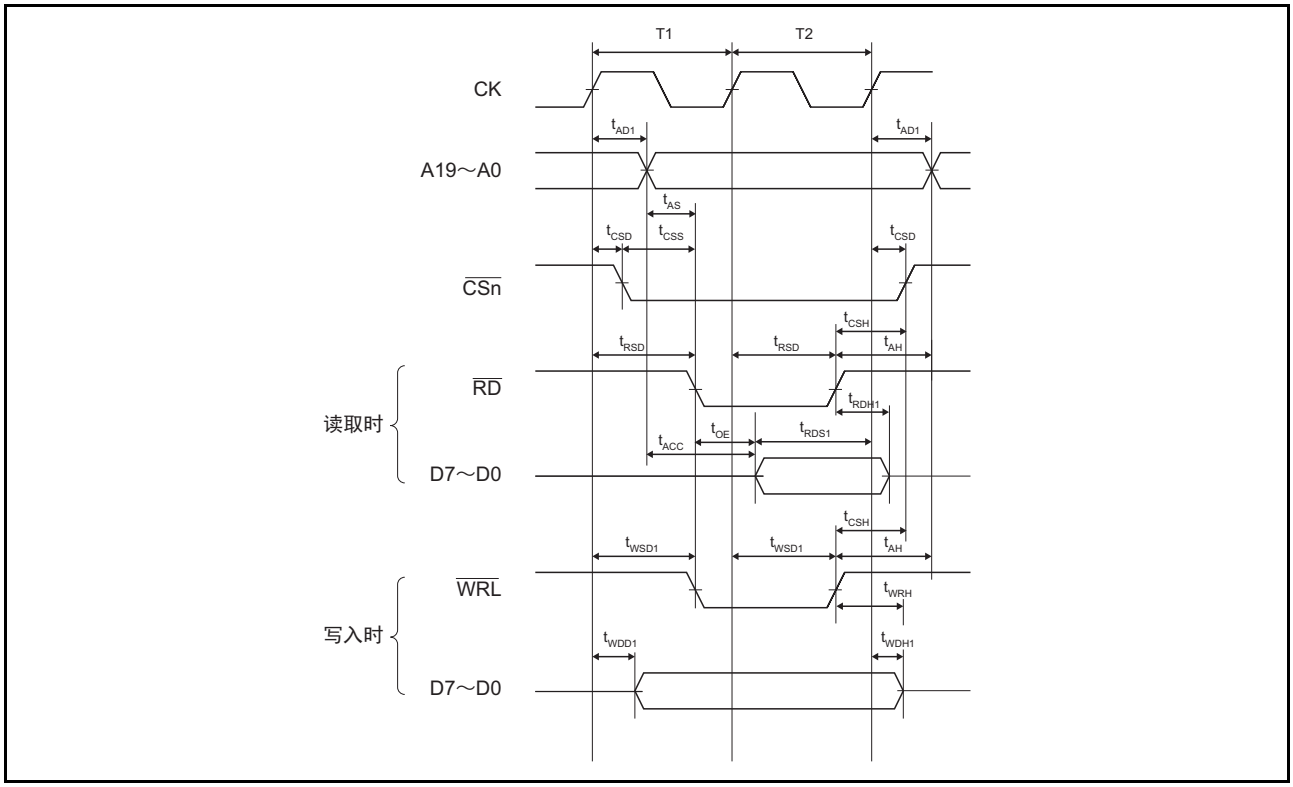


图 25.11 普通空间基本总线周期 （无等待）

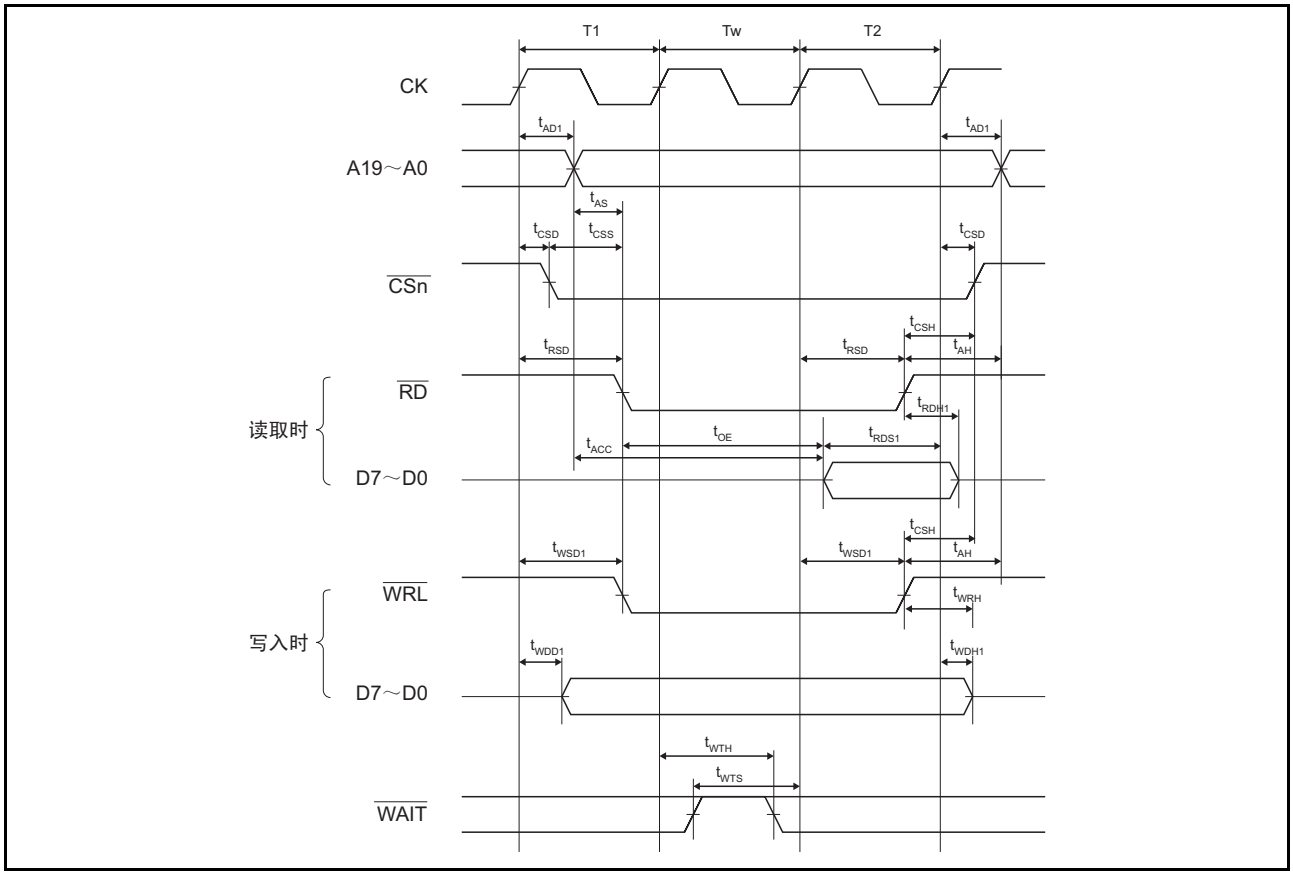


图 25.12 普通空间基本总线周期 （1 个软件等待）

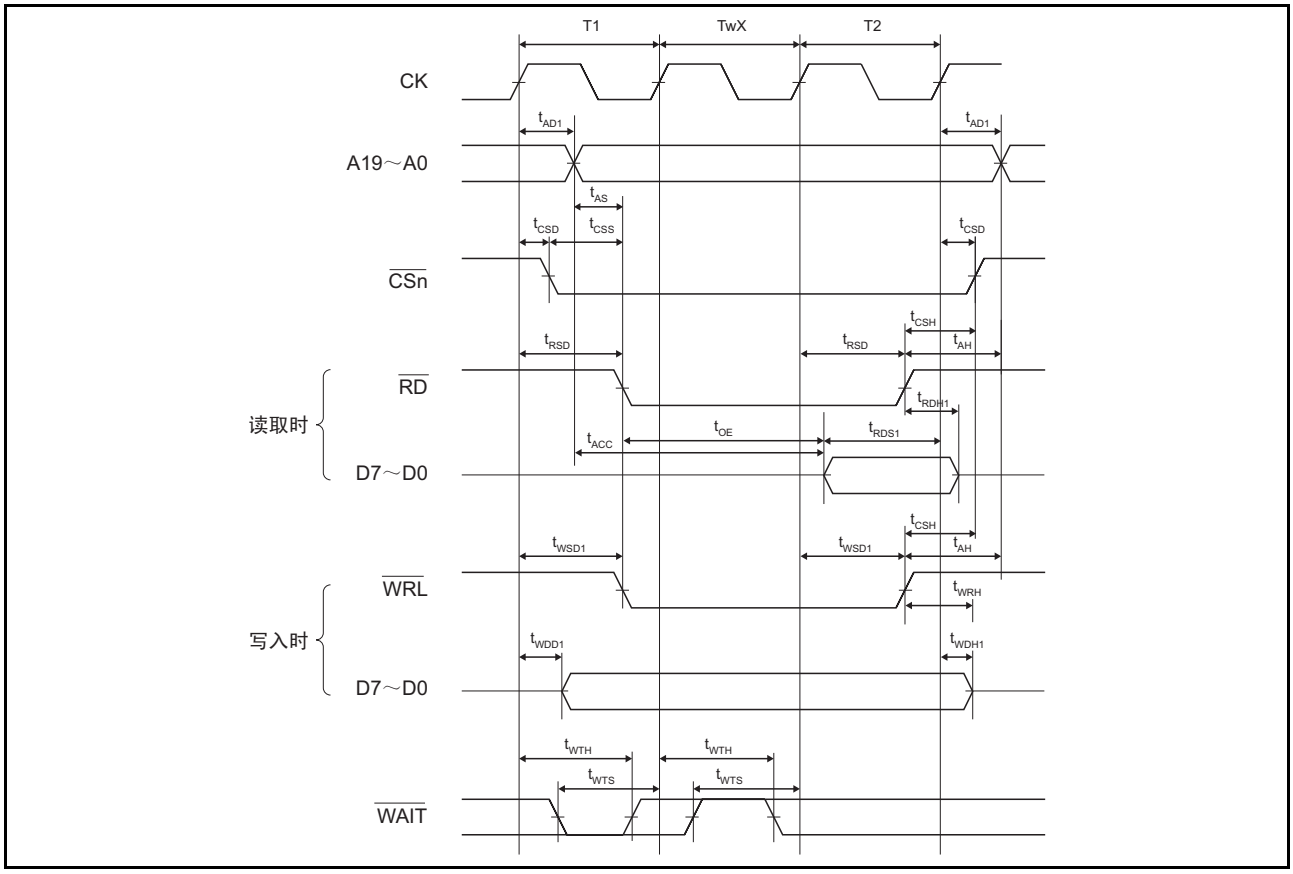


图 25.13 普通空间基本总线周期 （插入 1 个外部等待）

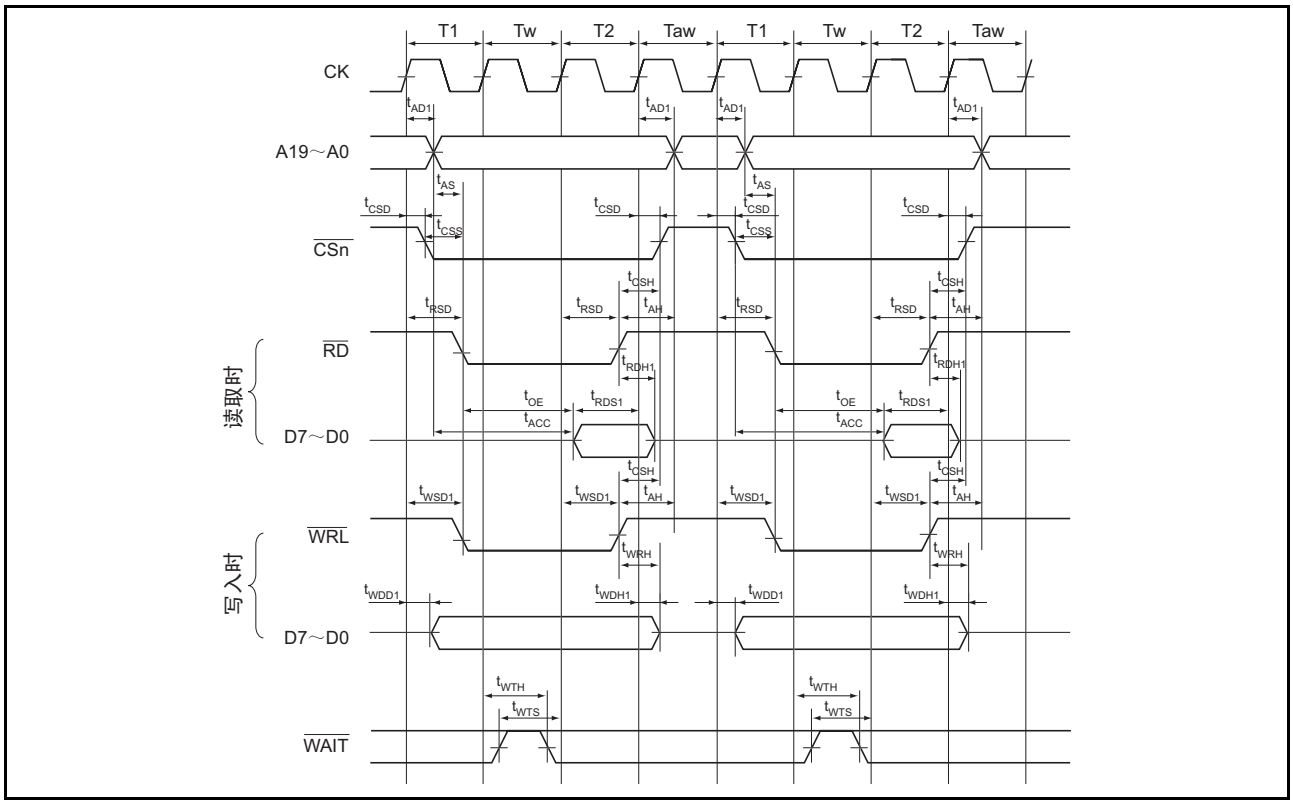


图 25.14 普通空间基本总线周期 （1 个软件等待、外部等待有效 （WM 位 =0）、无空闲周期）

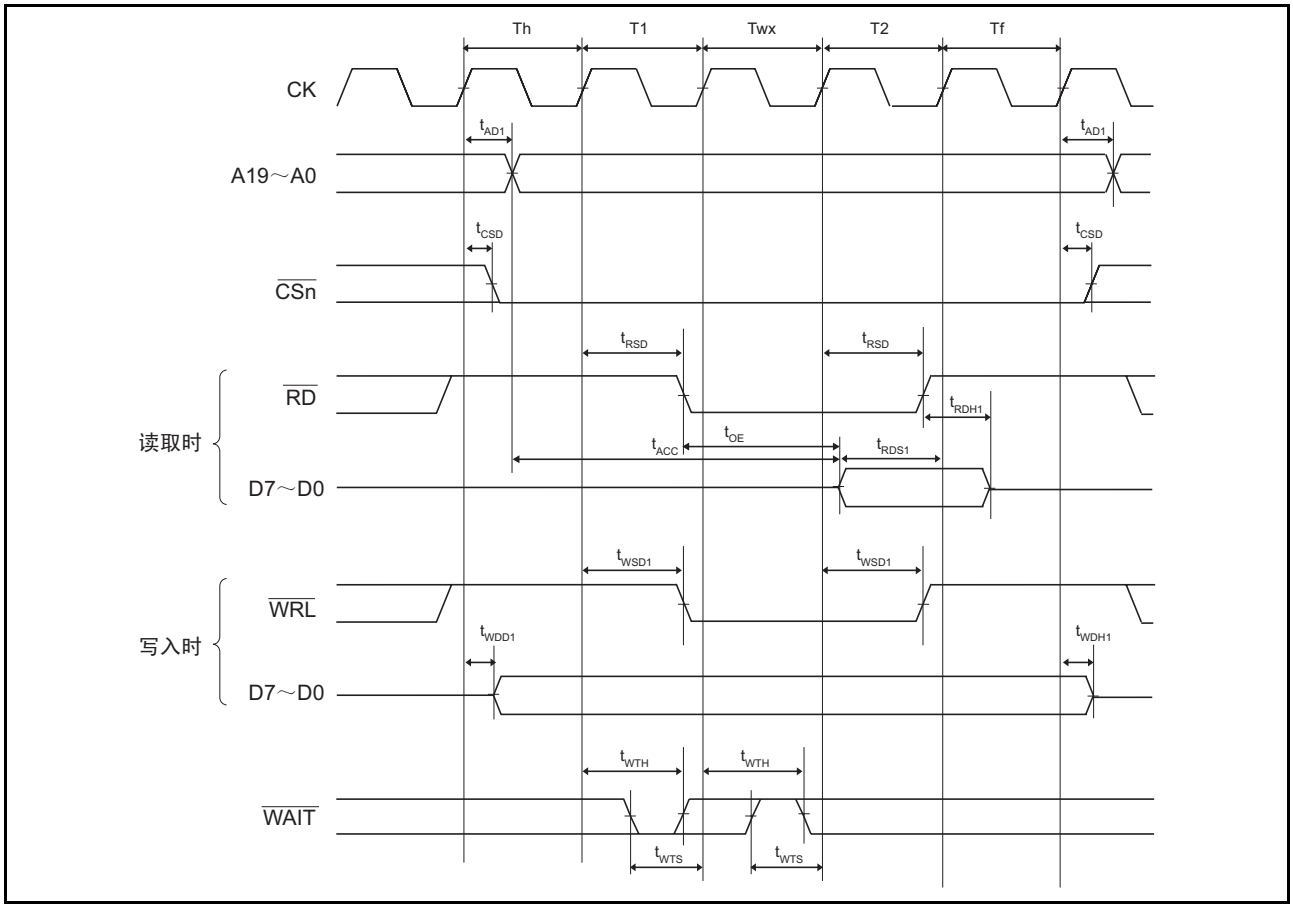


图 25.15 普通空间 CS 扩展总线周期 （SW=1 个周期、 HW=1 个周期、插入 1 个外部等待）

25.3.4 多功能定时器脉冲单元 2 （MTU2）时序

表 25.9 多功能定时器脉冲单元 2 （MTU2）时序

条件 （标准产品）：V_{CC}=4.5V ～ 5.5V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、 V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +85°C

条件 （宽温度范围产品）：V_{CC}=3.0V ～ 3.6V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、 V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +125°C

项 目	符号	Min.	Max.	单位	参照图
输出比较输出延迟时间	t_{TOCD}	—	50	ns	图 25.16
输入捕捉输入准备时间	t_{TICS}	20	—	ns	
输入捕捉输入脉宽 （指定单边沿）	t_{TICW}	1.5	—	t_{MPcyc}	
输入捕捉输入脉宽 （指定双边沿）	t_{TICW}	2.5	—	t_{MPcyc}	
定时器输入准备时间	t_{TCKS}	20	—	ns	图 25.17
定时器时钟脉宽 （指定单边沿）	$t_{TCKWH/L}$	1.5	—	t_{MPcyc}	
定时器时钟脉宽 （指定双边沿）	$t_{TCKWH/L}$	2.5	—	t_{MPcyc}	
定时器时钟脉宽 （相位计数模式）	$t_{TCKWH/L}$	2.5	—	t_{MPcyc}	

【注】 t_{MPcyc} 表示 MTU2 时钟 （MPϕ）的周期。

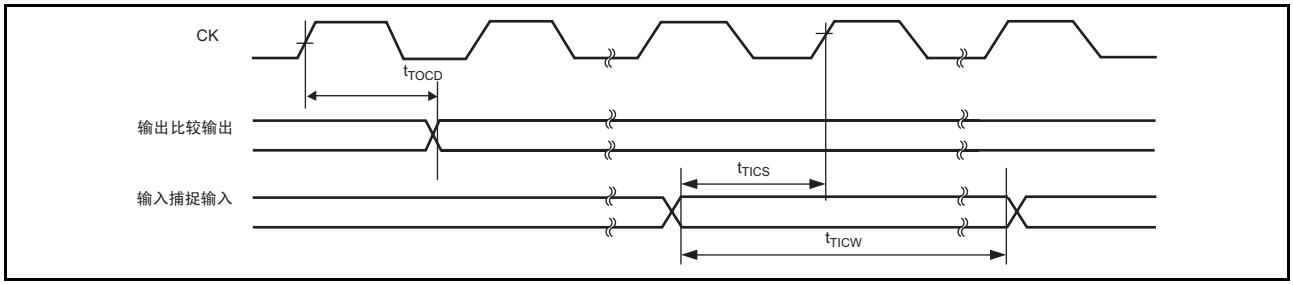


图 25.16　MTU2 输入 / 输出时序

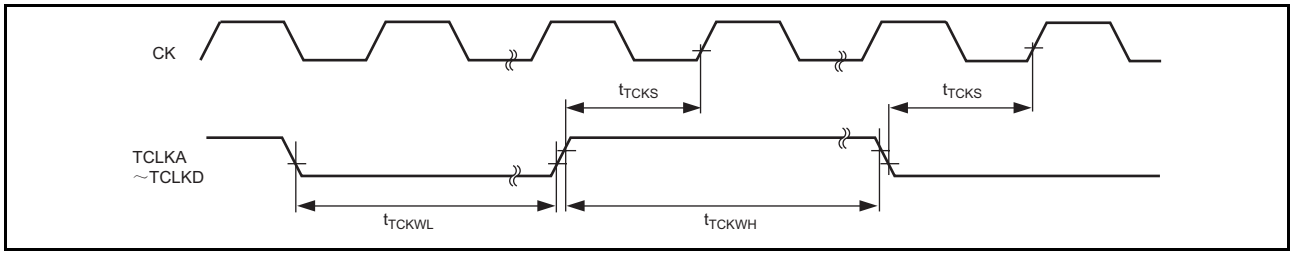


图 25.17　MTU2 时钟输入时序

25.3.5　多功能定时器脉冲单元 2S （MTU2S）时序

表 25.10　多功能定时器脉冲单元 2S （MTU2S）时序

条件 （标准产品）：V_{CC}=4.5V ～ 5.5V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、 V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +85°C

条件 （宽温度范围产品）：V_{CC}=3.0V ～ 3.6V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、 V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +125°C

项　　目	符号	Min.	Max.	单位	参照图
输出比较输出延迟时间	t_{TOCD}	—	50	ns	图 25.18
输入捕捉输入准备时间	t_{TICS}	20	—	ns	
输入捕捉输入脉宽 （指定单边沿）	t_{TICW}	1.5	—	t_{MIcyc}	
输入捕捉输入脉宽 （指定双边沿）	t_{TICW}	2.5	—	t_{MIcyc}	

【注】 t_{MIcyc} 表示 MTU2S 时钟 （MIϕ）的周期。

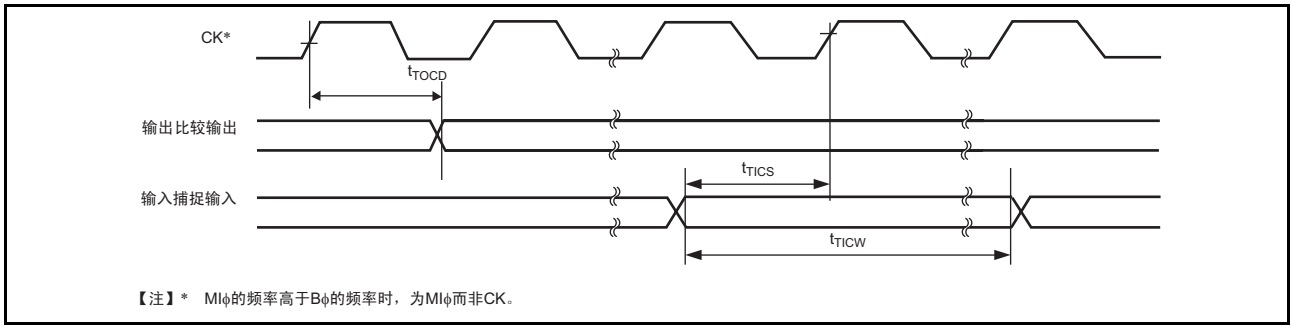


图 25.18　MTU2S 输入 / 输出时序

25.3.6　I/O 端口时序

表 25.11　I/O 端口时序

条件 （标准产品）：V_{CC}=4.5V ～ 5.5V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、 V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +85°C

条件 （宽温度范围产品）：V_{CC}=3.0V ～ 3.6V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、 V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +125°C

项　目	符号	Min.	Max.	单位	参照图
端口输出数据延迟时间	t_{PWD}	—	50	ns	图 25.19
端口输入保持时间	t_{PRH}	20	—	ns	
端口输入准备时间	t_{PRS}	20	—	ns	

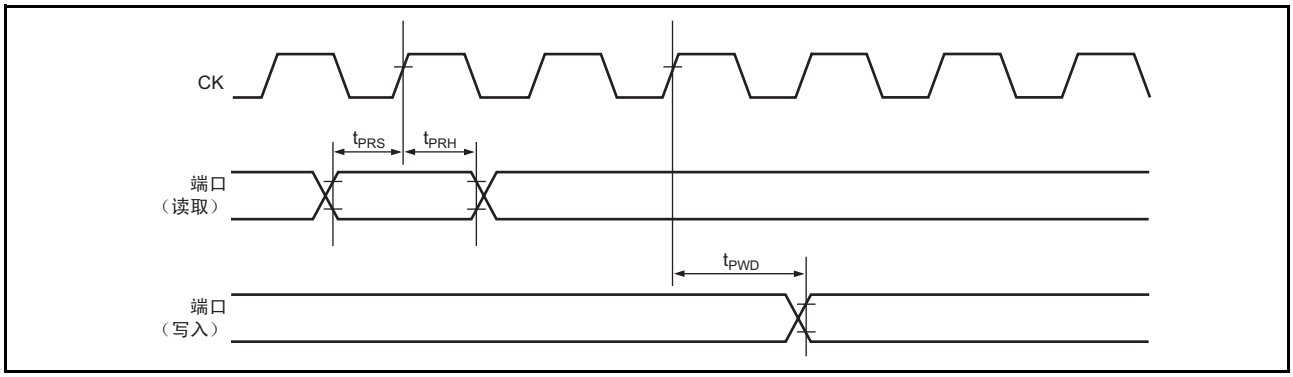


图 25.19　I/O 端口输入 / 输出时序

25.3.7　看门狗定时器 （WDT）时序

表 25.12　看门狗定时器 （WDT）时序

条件 （标准产品）：V_{CC}=4.5V ～ 5.5V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、 V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +85°C

条件 （宽温度范围产品）：V_{CC}=3.0V ～ 3.6V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、 V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +125°C

项　目	符号	Min.	Max.	单位	参照图
$\overline{WDTOVF}$ 延迟时间	t_{WOVD}	—	50	ns	图 25.20

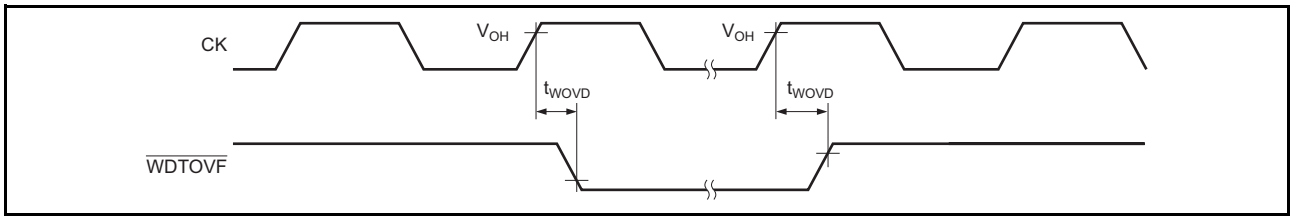


图 25.20　看门狗定时器时序

25.3.8　串行通信接口（SCI）时序

表 25.13　串行通信接口（SCI）时序

条件（标准产品）：V_{CC}=4.5V ～ 5.5V、PV_{CC}=4.5V ～ 5.5V、AV_{CC}=4.5V ～ 5.5V、AV_{refh}=4.5V ～ AV_{CC}、V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、T_a=–40 ～ +85°C

条件（宽温度范围产品）：V_{CC}=3.0V ～ 3.6V、PV_{CC}=4.5V ～ 5.5V、AV_{CC}=4.5V ～ 5.5V、AV_{refh}=4.5V ～ AV_{CC}、V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、T_a=–40 ～ +125°C

项　　目		符号	Min.	Max.	单位	参照图
输入时钟周期（异步）		t_{scyc}	4	—	t_{pcyc}	图 25.21
输入时钟周期（时钟同步）		t_{scyc}	6	—	t_{pcyc}	
输入时钟脉宽		t_{sckw}	0.4	0.6	t_{scyc}	
输入时钟上升时间		t_{sckr}	—	1.5	t_{pcyc}	
输入时钟下降时间		t_{sckf}	—	1.5	t_{pcyc}	
发送数据延迟时间	异步	t_{TXD}	—	$4t_{pcyc}$+10	ns	图 25.22
接收数据准备时间		t_{RXS}	$4t_{pcyc}$	—	ns	
接收数据保持时间		t_{RXH}	$4t_{pcyc}$	—	ns	
发送数据延迟时间	时钟同步	t_{TXD}	—	$3t_{pcyc}$+10	ns	
接收数据准备时间		t_{RXS}	$3t_{pcyc}$+50	—	ns	
接收数据保持时间		t_{RXH}	$3t_{pcyc}$+50	—	ns	

【注】 t_{pcyc} 表示外围时钟（Pϕ）的周期。

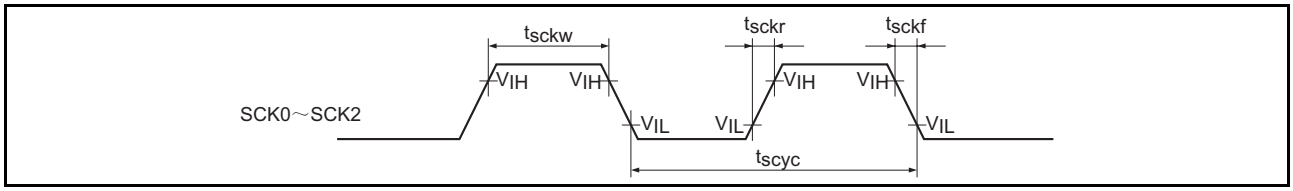


图 25.21　输入时钟时序

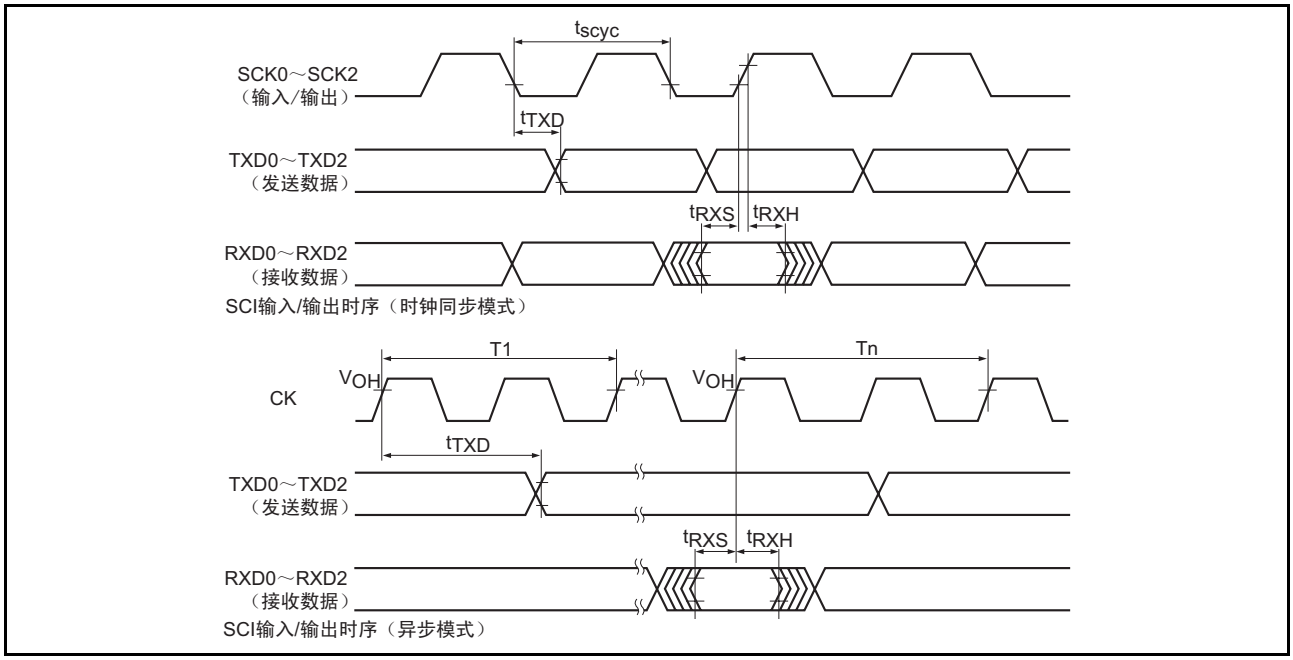


图 25.22　SCI 输入 / 输出时序

25.3.9　同步串行通信单元（SSU）时序

表 25.14　同步串行通信单元（SSU）时序

条件（标准产品）：V_{CC}=4.5V ～ 5.5V、PV_{CC}=4.5V ～ 5.5V、AV_{CC}=4.5V ～ 5.5V、AV_{refh}=4.5V ～ AV_{CC}、V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、T_a=–40 ～ +85°C

条件（宽温度范围产品）：V_{CC}=3.0V ～ 3.6V、PV_{CC}=4.5V ～ 5.5V、AV_{CC}=4.5V ～ 5.5V、AV_{refh}=4.5V ～ AV_{CC}、V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、T_a=–40 ～ +125°C

项　目		符号	Min.	Max.	单位	参照图
时钟周期	主模式	t_{SUcyc}	4	256	t_{pcyc}	图 25.23 图 25.24 图 25.25 图 25.26
	从属模式		4	256		
时钟高电平脉宽	主模式	t_{HI}	60	—	ns	
	从属模式		60	—		
时钟低电平脉宽	主模式	t_{LO}	60	—	ns	
	从属模式		60	—		
时钟上升时间		t_{RISE}	—	20	ns	
时钟下降时间		t_{FALL}	—	20	ns	
数据输入准备时间	主模式	t_{SU}	25	—	ns	
	从属模式		30	—		
数据输入保持时间	主模式	t_H	10	—	ns	
	从属模式		10	—		
$\overline{SCS}$ 准备时间	主模式	t_{LEAD}	1.5	—	t_{pcyc}	
	从属模式		1.5	—		
$\overline{SCS}$ 保持时间	主模式	t_{LAG}	1.5	—	t_{pcyc}	
	从属模式		1.5	—		
数据输出延迟时间	主模式	t_{OD}	—	40	ns	
	从属模式		—	40		
数据输出保持时间	主模式	t_{OH}	30	—	ns	
	从属模式		30	—		
连续发送延迟时间	主模式	t_{TD}	1.5	—	t_{pcyc}	
	从属模式		1.5	—		
从属模式存取时间		t_{SA}	—	1	t_{pcyc}	图 25.25 图 25.26
从属模式输出释放时间		t_{REL}	—	1	t_{pcyc}	

【注】 t_{pcyc} 表示外围时钟（Pϕ）的周期。

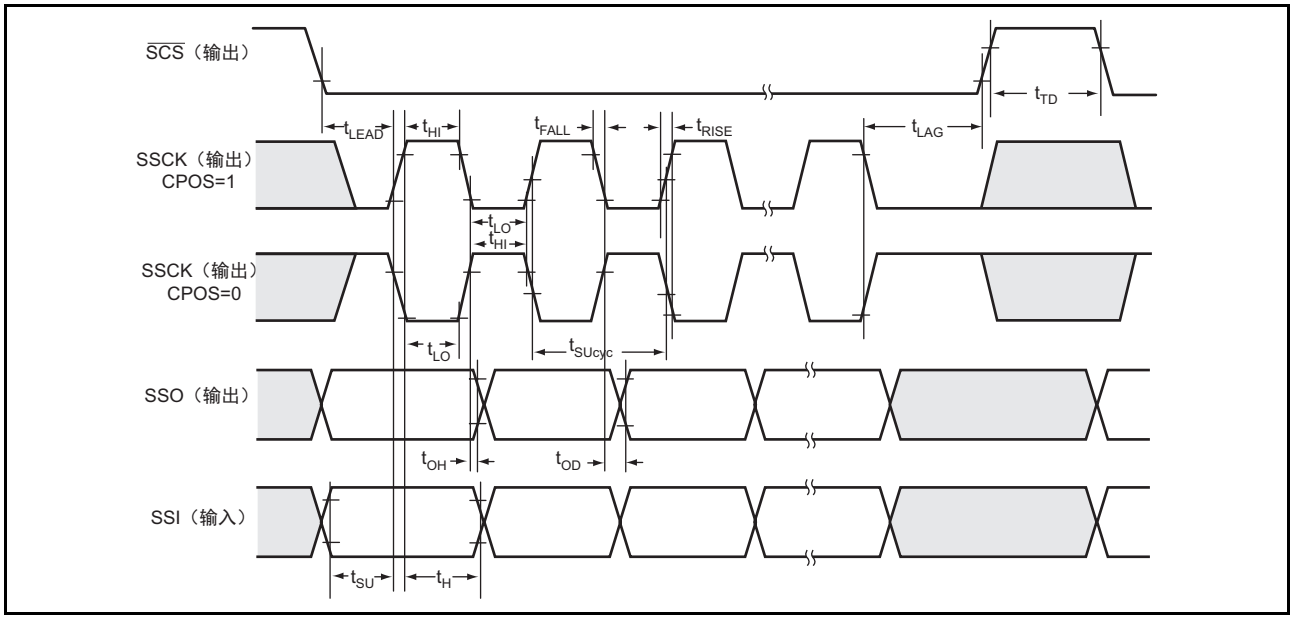


图 25.23 SSU 时序 （主模式、 CPHS=1）

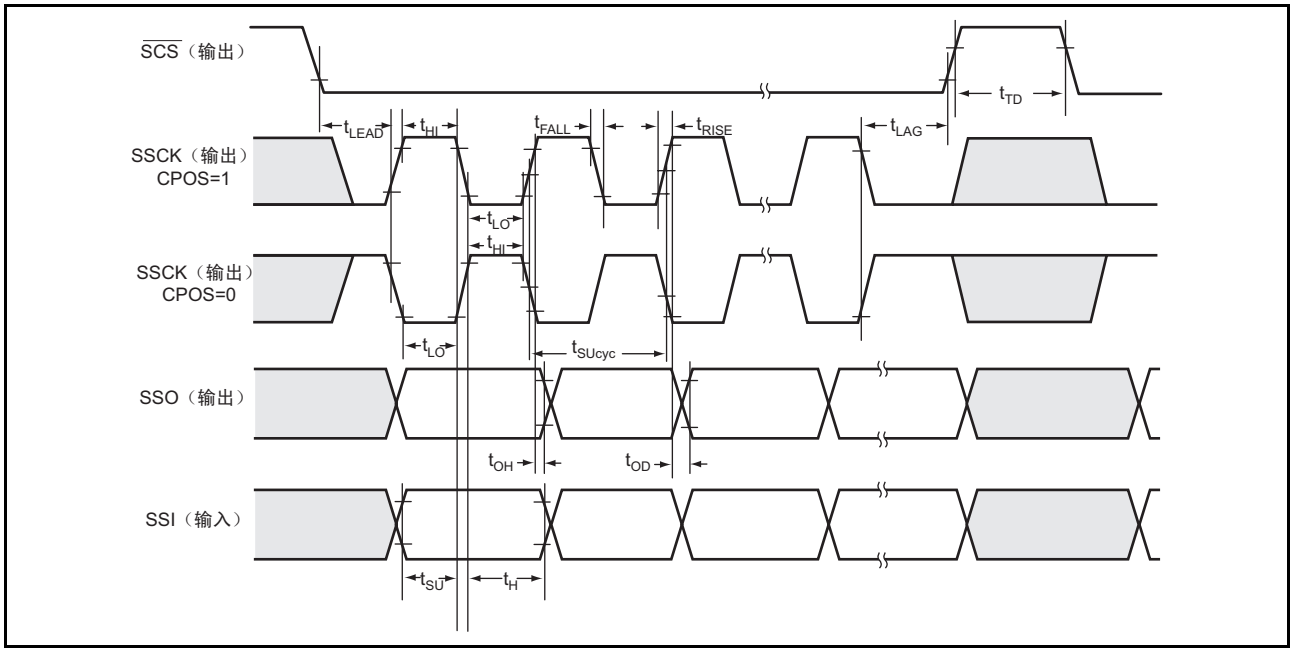


图 25.24 SSU 时序 （主模式、 CPHS=0）

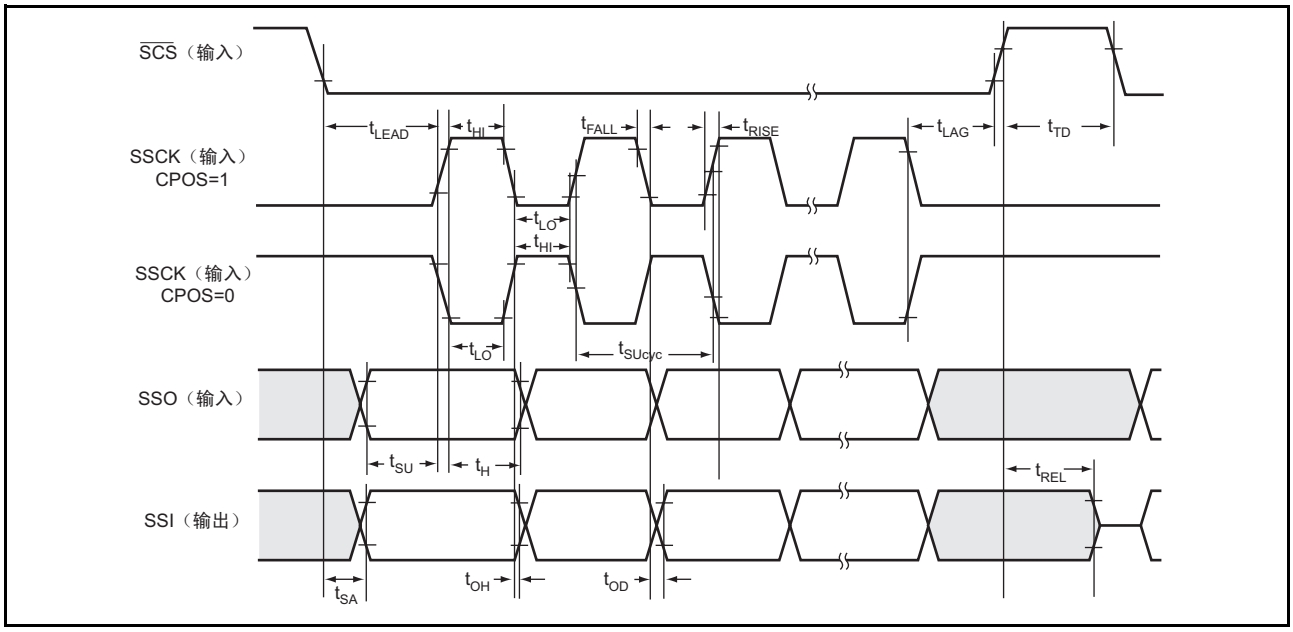


图 25.25 SSU 时序 （从属模式、 CPHS=1）

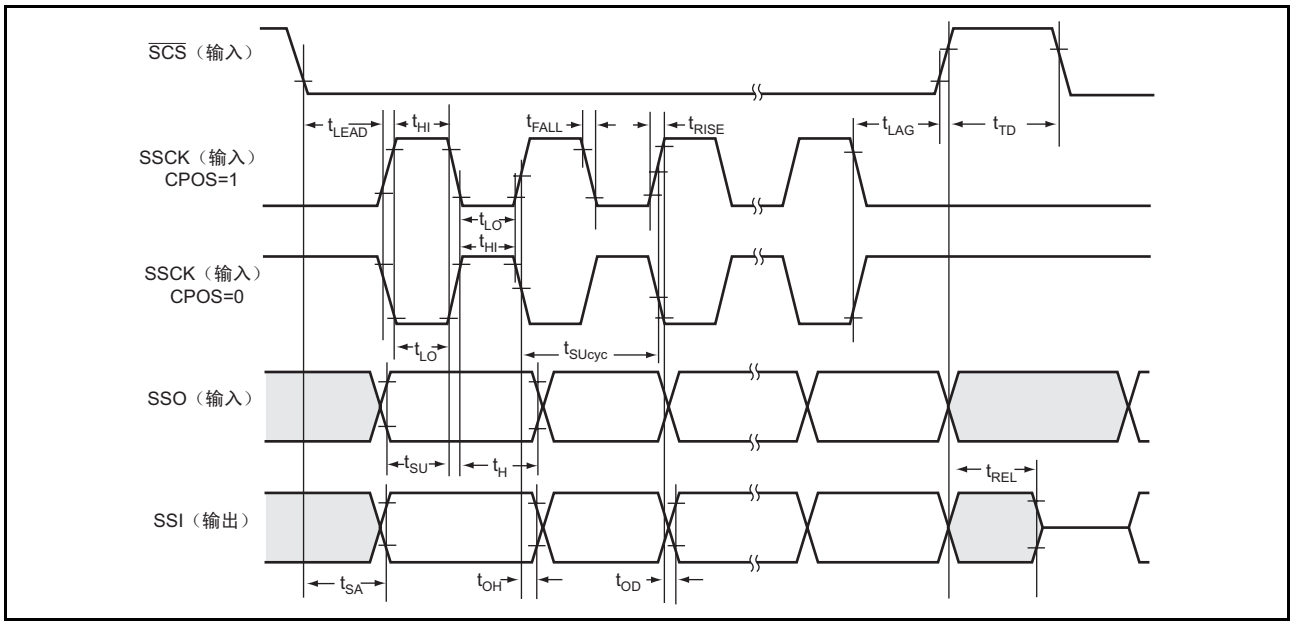


图 25.26 SSU 时序 （从属模式、 CPHS=0）

25.3.10　控制器局域网（RCAN-ET）时序

RCAN-ET 时序如表 25.15 所示。

表 25.15　控制器局域网（RCAN-ET）时序

条件（标准产品）：V_{CC}=4.5V ～ 5.5V、PV_{CC}=4.5V ～ 5.5V、AV_{CC}=4.5V ～ 5.5V、AV_{refh}=4.5V ～ AV_{CC}、V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、T_a=–40 ～ +85°C

条件（宽温度范围产品）：V_{CC}=3.0V ～ 3.6V、PV_{CC}=4.5V ～ 5.5V、AV_{CC}=4.5V ～ 5.5V、AV_{refh}=4.5V ～ AV_{CC}、V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、T_a=–40 ～ +125°C

项　目	符号	Min.	Max.	单位	参照图
发送数据延迟时间	t_{CTxD}	—	100	ns	图 25.27
接收数据准备时间	t_{CRxS}	100	—	ns	
接收数据保持时间	t_{CRxH}	100	—	ns	

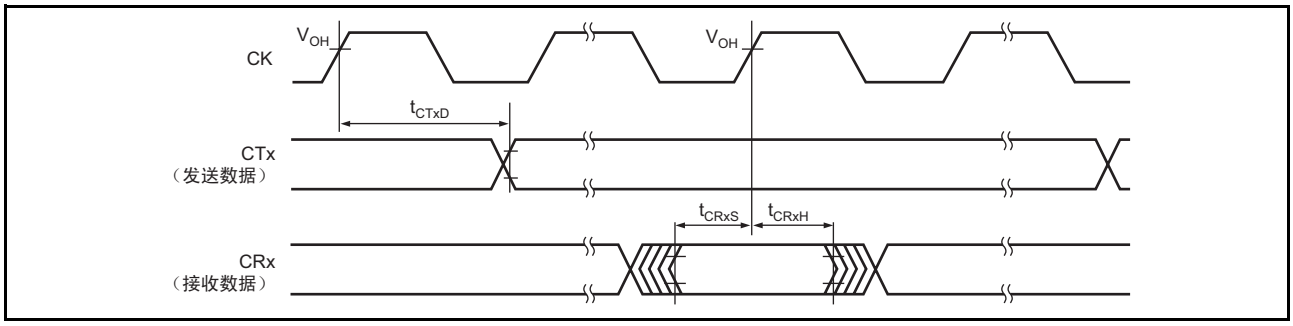


图 25.27　RCAN-ET 输入 / 输出时序

25.3.11　端口输出允许（POE）时序

表 25.16　端口输出允许（POE）时序

条件（标准产品）：V_{CC}=4.5V ～ 5.5V、PV_{CC}=4.5V ～ 5.5V、AV_{CC}=4.5V ～ 5.5V、AV_{refh}=4.5V ～ AV_{CC}、V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、T_a=–40 ～ +85°C

条件（宽温度范围产品）：V_{CC}=3.0V ～ 3.6V、PV_{CC}=4.5V ～ 5.5V、AV_{CC}=4.5V ～ 5.5V、AV_{refh}=4.5V ～ AV_{CC}、V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、T_a=–40 ～ +125°C

项　目	符号	Min.	Max.	单位	参照图
$\overline{POE}$ 输入准备时间	t_{POES}	50	—	ns	图 25.28
$\overline{POE}$ 输入脉宽	t_{POEW}	1.5	—	t_{pcyc}	

【注】 t_{pcyc} 表示外围时钟（Pϕ）的周期。

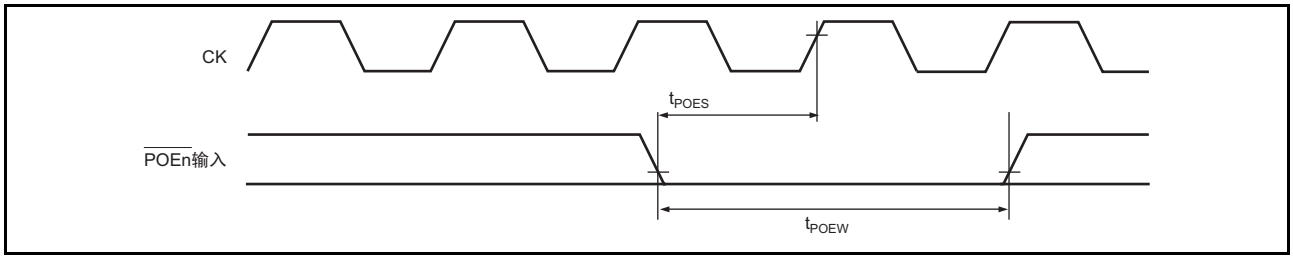


图 25.28　$\overline{POE}$ 输入时序

25.3.12 UBC 触发时序

表 25.17 UBC 触发时序

条件 （标准产品）：V_{CC}=4.5V ～ 5.5V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、 V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +85°C

条件 （宽温度范围产品）：V_{CC}=3.0V ～ 3.6V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、 V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +125°C

项 目	符号	Min.	Max.	单位	参照图
$\overline{\text{UBCTRG}}$ 延迟时间	t_{UBCTGD}	—	150	ns	图 25.29

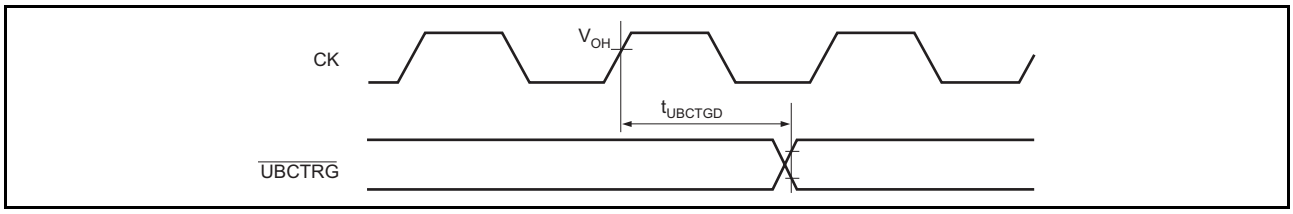


图 25.29 UBC 触发时序

25.3.13 A/D 转换器时序

表 25.18 A/D 转换器时序

条件 （标准产品）：V_{CC}=4.5V ～ 5.5V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、 V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +85°C

条件 （宽温度范围产品）：V_{CC}=3.0V ～ 3.6V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、 V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +125°C

项 目	符号	Min.	Typ.	Max.	单位	参照图
外部触发输入开始延迟时间	t_{TRGS}	25	—	—	ns	图 25.30

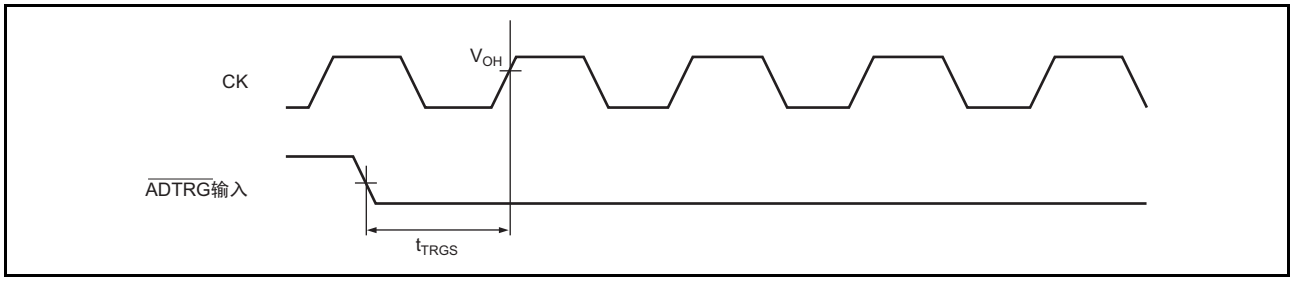


图 25.30 外部触发输入时序

25.3.14　AUD 时序

AUD 时序如表 25.19 所示。

表 25.19　AUD 时序

条件 （标准产品）：V_{CC}=4.5V ～ 5.5V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、 V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +85°C

条件 （宽温度范围产品）：V_{CC}=3.0V ～ 3.6V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、 V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +125°C

项　　目	符号	Min.	Max.	单位	参照图
$\overline{\text{AUDRST}}$ 脉宽 （转移跟踪时）	$t_{AUDRSTW}$	20	—	t_{Bcyc}	图 25.31
$\overline{\text{AUDRST}}$ 脉宽 （RAM 监控时）	$t_{AUDRSTW}$	5	—	t_{RMCYC}	
AUDMD 准备时间 （转移跟踪时）	t_{AUDMDS}	20	—	t_{Bcyc}	
AUDMD 准备时间 （RAM 监控时）	t_{AUDMDS}	5	—	t_{RMCYC}	
转移跟踪时钟周期	t_{BTCYC}	2	2	ns	图 25.32
转移跟踪时钟占空比	t_{BTCKW}	40	60	%	
转移跟踪数据延迟时间	t_{BTDD}	—	50	ns	
转移跟踪数据保持时间	t_{BTDH}	50	—	ns	
转移跟踪 SYNC 延迟时间	t_{BTSD}	—	50	ns	
转移跟踪 SYNC 保持时间	t_{BTSH}	50	—	ns	
RAM 监控时钟周期	t_{RMCYC}	100	—	ns	图 25.33
RAM 监控时钟低电平脉宽	t_{RMCKW}	42	—	ns	
RAM 监控输出数据延迟时间	t_{RMDD}	5	t_{RMCYC}–20	ns	
RAM 监控输出数据保持时间	t_{RMDHD}	50	—	ns	
RAM 监控输入数据准备时间	t_{RMDS}	50	—	ns	
RAM 监控输入数据保持时间	t_{RMDH}	50	—	ns	
RAM 监控 SYNC 准备时间	t_{RMSS}	50	—	ns	
RAM 监控 SYNC 保持时间	t_{RMSH}	50	—	ns	

负载条件：

AUDCK （输出时）：CL=30pF
$\overline{\text{AUDSYNC}}$　　：CL=30pF
AUDATA3 ～ 0　　：CL=30pF

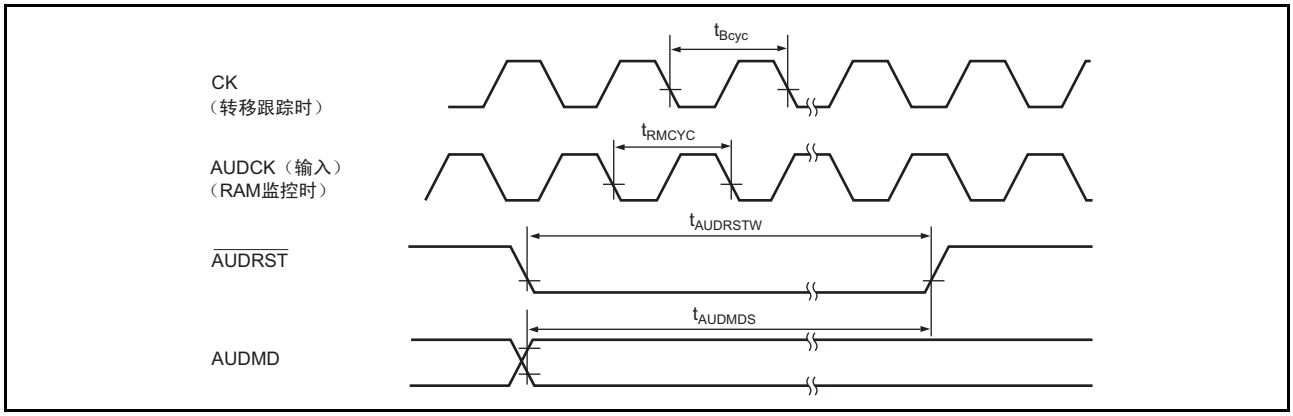


图 25.31 AUD 复位时序

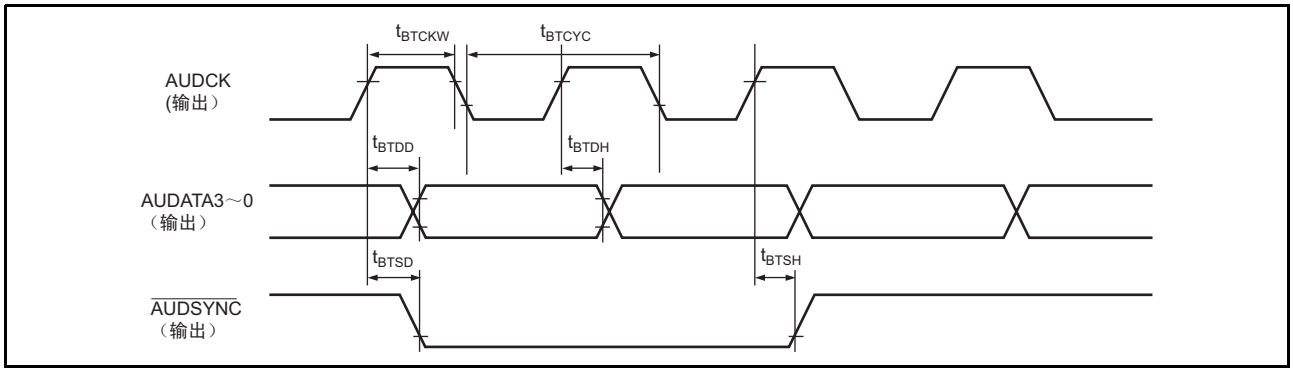


图 25.32 转移跟踪时序

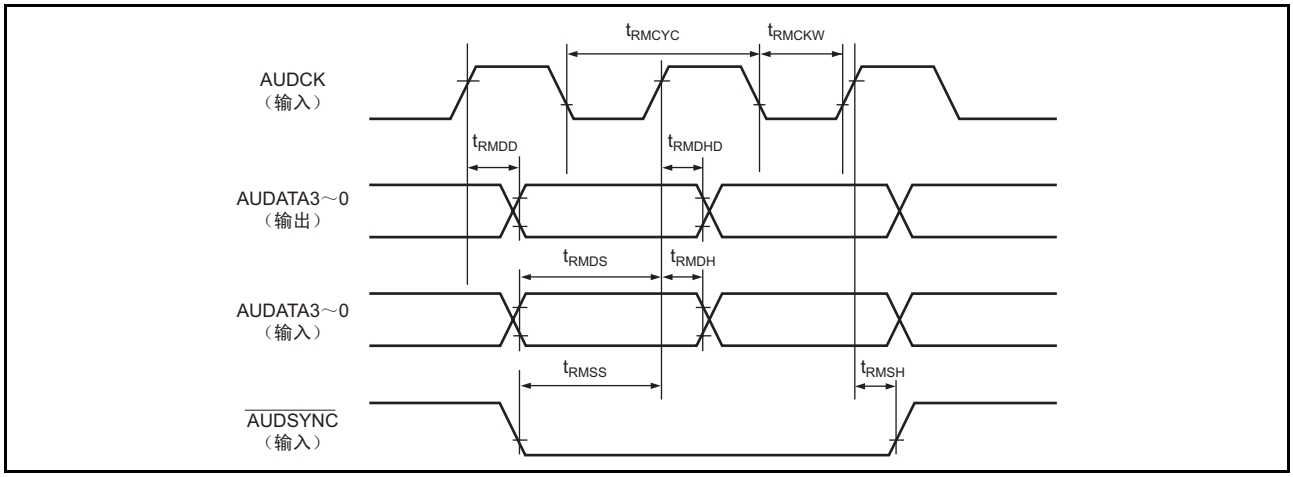
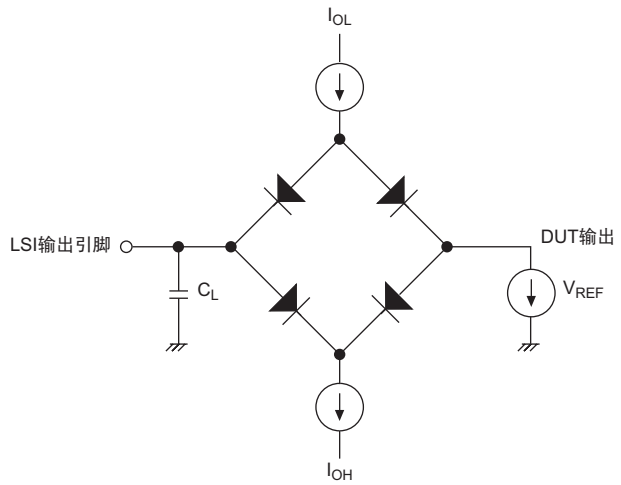


图 25.33 RAM 监控时序

25.3.15　AC 特性测定条件

- 输入信号电平：V_{IL}（Max.）/V_{IH}（Min.）
- 输出信号参考电平：高电平：2.0V、低电平：0.8V



【注】1. C_L为包含测量装置容量的总值。各引脚如下设定。
20pF：CK
30pF：上述以外的所有输出引脚
2. I_{OL}、I_{OH}的条件为I_{OL}=1.6mA、I_{OH}=-200μA。

图 25.34　输出负载电路

25.4　A/D 转换器特性

表 25.20　A/D 转换器特性

条件（标准产品）：V_{CC}=4.5V ～ 5.5V、PV_{CC}=4.5V ～ 5.5V、AV_{CC}=4.5V ～ 5.5V、AV_{refh}=4.5V ～ AV_{CC}、V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、T_a=–40 ～ +85°C

条件（宽温度范围产品）：V_{CC}=3.0V ～ 3.6V、PV_{CC}=4.5V ～ 5.5V、AV_{CC}=4.5V ～ 5.5V、AV_{refh}=4.5V ～ AV_{CC}、V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、T_a=–40 ～ +125°C

项　目		Min.	Typ.	Max.	单位
分辨率		12	12	12	位
转换时间	标准产品	—	—	1.25*1	μs
	宽温度范围产品			1.56*2	
模拟输入电容		—	—	5	pF
容许信号源阻抗		—	—	3	kΩ
非线性误差（积分误差）		—	—	±4*3	LSB
绝对精度 *4		—	—	±8	LSB

【注】 *1　未使用采样 & 保持电路、A/D 时钟以 40MHz 运行时，每个通道的转换时间。
*2　未使用采样 & 保持电路、A/D 时钟以 32MHz 运行时，每个通道的转换时间。
*3　非线性误差为参考值。
*4　绝对精度的保证范围为 AV_{refl}+0.25V ～ AV_{refh}–0.25V。

25.5　闪存特性

表 25.21　闪存特性

条件 （标准产品）：V_{CC}=4.5V ～ 5.5V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、 V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +85°C

条件 （宽温度范围产品）：V_{CC}=3.0V ～ 3.6V、 PV_{CC}=4.5V ～ 5.5V、 AV_{CC}=4.5V ～ 5.5V、 AV_{refh}=4.5V ～ AV_{CC}、 V_{SS}=$PLLV_{SS}$=AV_{SS}=AV_{refl}=0V、 T_a=–40 ～ +85°C*6

项　　目	符号	Min.	Typ.	Max.	单位
编程时间 *1*2*4	t_P	—	1	10	ms/128 字节
擦除时间 *1*3*5	t_E	—	8	25	ms/ 千字节
改写次数	N_{WEC}	—	—	100	次

【注】 *1　编程 / 擦除时必须使用内部的编程 / 擦除程序。

*2　全部写入 0 时。

*3　64KB 块时。

*4　总改写时间 （编程时间 + 擦除时间）如下：
256K字节版：60s （Typ.）、参考值90s、 120s （Max.）
384K字节版：90s （Typ.）、参考值120s、 180s （Max.）
512K字节版：120s （Typ.）、参考值180s、 240s （Max.）
但只有90％符合参考值。

*5　t_E 分布于 Typ. 值的附近。

*6　仅编程 / 擦除时为 85°C，普通运行时为 125°C。

25.6　使用时的注意事项

25.6.1　V_{CL} 电容连接方法

本 LSI 内置内部降压电路，可将单片机内部的电源电压自动降压到最佳电平。内部降压电源 （V_{CL} 引脚）与 V_{SS} 引脚之间需要连接可稳定内部电压的电容 （0.47μF）。连接外部电容的方法如图 25.35 所示。外部电容必须配置在引脚附近。不可对 V_{CL} 引脚外加电源电压。

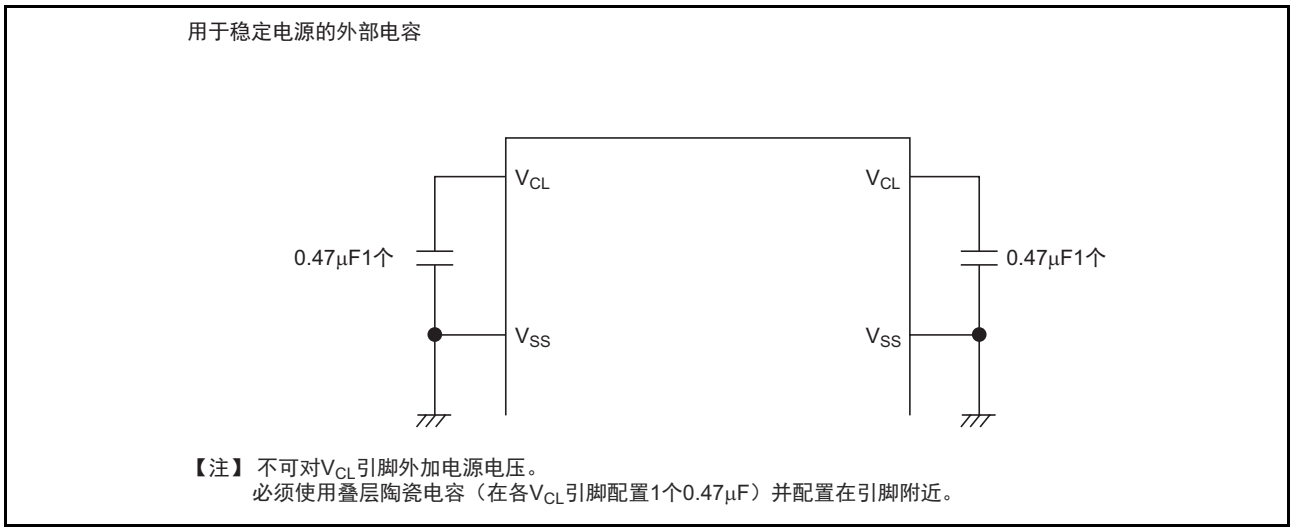


图 25.35　连接 V_{CL} 电容的方法

附录

附录 A. 引脚状态

引脚的初始值因 MCU 运行模式而不同，详情参照“第 18 章 引脚功能控制器 （PFC）”。

表 A.1 引脚状态

引脚功能		引脚状态										
分类	引脚名称	复位状态				低功耗状态				总线权释放状态	检测到振荡停止时	使用 POE 功能时
		上电			手动	硬件待机	深度软件待机	软件待机	睡眠			
		无 ROM 扩展	有 ROM 扩展	单芯片								
时钟	CK	O		Z	O	Z	Z	H*1	O	O	O	O
	XTAL	O			O	L	L	L	O	O	O	O
	EXTAL	I			I	Z	Z	I	I	I	I	I
系统控制	$\overline{\text{RES}}$	I			I	I	I	I	I	I	I	I
	$\overline{\text{MRES}}$	Z			I	Z	Z	Z	I	I	Z	I
	$\overline{\text{WDTOVF}}$	O*3			O	O	O	O	O	O	O	O
	$\overline{\text{BREQ}}$	Z			I	Z	Z	Z	I	I	I	I
	$\overline{\text{BACK}}$	Z			O	Z	Z	Z	O	L	O	O
运行模式控制	MD0、 MD1	I			I	I	I	I	I	I	I	I
	$\overline{\text{HSTBY}}$	I*4			I*4	I	I	I*4	I*4	I*4	I*4	I*4
	FWE	I			I	I	I	I	I	I	I	I
中断	NMI	I			I	Z	I	I	I	I	I	I
	IRQ0 ～ IRQ3	Z			I	Z	Z	I	I	I	I	I
	$\overline{\text{IRQOUT}}$	Z			O	Z	Z	Z	O	O	Z	O
地址总线	A0 ～ A17	O	Z		O	Z	Z	Z*2	O	Z	O	O
	A18、 A19	Z			O	Z	Z	Z*2	O	Z	O	O
数据总线	D0 ～ D7	Z			I/O	Z	Z	Z	I/O	Z	I/O	I/O
总线控制	$\overline{\text{WAIT}}$	Z			I	Z	Z	Z	I	Z	I	I
	$\overline{\text{CS0}}$ （PE10）	H	Z		O	Z	Z	Z*2	O	Z	O	O
	$\overline{\text{CS0}}$ （PE17）、$\overline{\text{CS1}}$	Z			O	Z	Z	Z*2	O	Z	O	O
	$\overline{\text{RD}}$ （PA6）	H	Z		O	Z	Z	Z*2	O	Z	O	O
	$\overline{\text{RD}}$ （PE19）	Z			O	Z	Z	Z*2	O	Z	O	O
	$\overline{\text{WRL}}$ （PA8）	H	Z		O	Z	Z	Z*2	O	Z	O	O
	$\overline{\text{WRL}}$ （PE21）	Z			O	Z	Z	Z*2	O	Z	O	O

引脚功能		引脚状态										
分类	引脚名称	复位状态				低功耗状态				总线权释放状态	检测到振荡停止时	使用 POE 功能时
		上电			手动	硬件待机	深度软件待机	软件待机	睡眠			
		无 ROM 扩展	有 ROM 扩展	单芯片								
MTU2	TCLKA ～ TCLKD	Z			I	Z	Z	Z	I	I	I	I
	TIOC0A ～ TIOC0D	Z			I/O	Z	Z	K*[1]	I/O	I/O	I/O	Z
	TIOC1A、TIOC1B	Z			I/O	Z	Z	K*[1]	I/O	I/O	I/O	I/O
	TIOC2A、TIOC2B	Z			I/O	Z	Z	K*[1]	I/O	I/O	I/O	I/O
	TIOC3A、TIOC3C	Z			I/O	Z	Z	K*[1]	I/O	I/O	I/O	I/O
	TIOC3B、TIOC3D	Z			I/O	Z	Z	Z	I/O	I/O	Z	Z
	TIOC4A ～ TIOC4D	Z			I/O	Z	Z	Z	I/O	I/O	Z	Z
MTU2S	TIOC3BS、TIOC3DS	Z			I/O	Z	Z	Z	I/O	I/O	Z	Z
	TIOC4AS ～ TIOC4DS	Z			I/O	Z	Z	Z	I/O	I/O	Z	Z
	TIC5US、TIC5VS、TIC5WS	Z			I	Z	Z	Z	I	I	I	I
POE	$\overline{\text{POE0}}$ ～ $\overline{\text{POE2}}$、$\overline{\text{POE4}}$ ～ $\overline{\text{POE6}}$、$\overline{\text{POE8}}$	Z			I	Z	Z	Z	I	I	I	I
SCI	SCK0 ～ SCK2	Z			I/O	Z	Z	Z	I/O	I/O	I/O	I/O
	RXD0 ～ RXD2	Z			I	Z	Z	Z	I	I	I	I
	TXD0 ～ TXD2	Z			O	Z	Z	O*[1]	O	O	O	O
SSU	SSCK	Z			I/O	Z	Z	Z	I/O	I/O	I/O	I/O
	$\overline{\text{SCS}}$	Z			I/O	Z	Z	Z	I/O	I/O	Z	I/O
	SSI	Z			I/O	Z	Z	Z	I/O	I/O	I/O	I/O
	SSO	Z			I/O	Z	Z	Z	I/O	I/O	I/O	I/O
UBC	$\overline{\text{UBCTRG}}$	Z			O	Z	Z	O*[1]	O	O	O	O
RCAN-ET	CTx0	Z			O	Z	Z	O*[1]	O	O	O	O
	CRx0	Z			I	Z	Z	Z	I	I	I	I
	CTx1	Z			O	Z	Z	O*[1]	O	O	O	O
	CRx1	Z			I	Z	Z	Z	I	I	I	I

引脚功能		引脚状态										
分类	引脚名称	复位状态				低功耗状态				总线权释放状态	检测到振荡停止时	使用 POE 功能时
		上电			手动	硬件待机	深度软件待机	软件待机	睡眠			
		无 ROM 扩展	有 ROM 扩展	单芯片								
A/D 转换器	AN0 ～ AN15	Z			I	Z	Z	Z	I	I	I	I
	$\overline{\text{ADTRG}}$	Z			I	Z	Z	Z	I	I	I	I
I/O 端口	PA0 ～ PA15	Z			I/O	Z	Z	K*1	I/O	I/O	I/O	I/O
	PB0 ～ PB7	Z			I/O	Z	Z	K*1	I/O	I/O	I/O	I/O
	PD0 ～ PD10	Z			I/O	Z	Z	K*1	I/O	I/O	I/O	I/O
	PE0 ～ PE3	Z			I/O	Z	Z	K*1	I/O	I/O	I/O	Z
	PE4 ～ PE8、PE10	Z			I/O	Z	Z	K*1	I/O	I/O	I/O	I/O
	PE9、PE11 ～ PE15	Z			I/O	Z	Z	Z	I/O	I/O	Z	Z
	PE16 ～ PE21	Z			I/O	Z	Z	Z	I/O	I/O	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，输出引脚保持其状态

【注】 *1 将待机控制寄存器 6 （STBCR6）的 HIZ 位置 1 时，输出引脚为高阻抗。

*2 将共用控制寄存器 （CMNCR）的 HIZMEM 位置 1 时，为输出。

*3 上电复位时为输入。为了防止误动作，必须上拉。另外，需下拉时，必须以大于等于 1MΩ 的电阻值执行下拉。

*4 无任何输入时，在内部上拉。

表 A.2 AUD 的引脚状态

引脚功能		引脚状态								
分类	引脚名称	复位状态				低功耗状态				
		上电				硬件待机	AUD 模块待机	深度软件待机	软件待机	睡眠
		无 ROM 扩展	有 ROM 扩展	单芯片	AUD 复位					
AUD	$\overline{\text{AUDRST}}$	H 输入	H 输入	H 输入	L 输入	Z	Z	Z	Z	H 输入
	AUDMD	I	I	I	I	Z	Z	Z	Z	I
	AUDATA3 ～ AUDATA0	AUDMD=H 时：I/O AUDMD=L 时：O	AUDMD=H 时：I/O AUDMD=L 时：O	AUDMD=H 时：I/O AUDMD=L 时：O	AUDMD=H 时：I AUDMD=L 时：O	Z	Z	Z	Z	AUDMD=H 时：I/O AUDMD=L 时：O
	AUDCK	AUDMD=H 时：I AUDMD=L 时：O	AUDMD=H 时：I AUDMD=L 时：O	AUDMD=H 时：I AUDMD=L 时：O	AUDMD=H 时：I AUDMD=L 时：O	Z	Z	Z	Z	AUDMD=H 时：I AUDMD=L 时：O
	$\overline{\text{AUDSYNC}}$	AUDMD=H 时：I AUDMD=L 时：O	AUDMD=H 时：I AUDMD=L 时：O	AUDMD=H 时：I AUDMD=L 时：O	AUDMD=H 时：I AUDMD=L 时：O	Z	Z	Z	Z	AUDMD=H 时：I AUDMD=L 时：O

【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

附录 B. 总线相关信号的引脚状态

表 B.1 总线相关信号的引脚状态 （1）

引脚名称		内部 ROM 空间	内部 RAM 空间	内部外围模块空间
$\overline{\text{CS0}}$、$\overline{\text{CS1}}$		H	H	H
$\overline{\text{RD}}$	R	H	H	H
	W	—	H	H
$\overline{\text{WRL}}$	R	H	H	H
	W	—	H	H
A19 ～ A0		地址 *	地址 *	地址 *
D7 ～ D0		Hi-Z	Hi-Z	Hi-Z

【符号说明】

R：读取

W：写入

【注】 * 以前存取的外部空间的地址值

表 B.2 总线相关信号的引脚状态 （2）

引脚名称		外部空间 （通常空间）
		8 位空间
$\overline{CS0}$、$\overline{CS1}$		有效
$\overline{RD}$	R	L
	W	H
$\overline{WRL}$	R	H
	W	L
A19 ～ A0		地址
D7 ～ D0		数据

【符号说明】

R：读取

W：写入

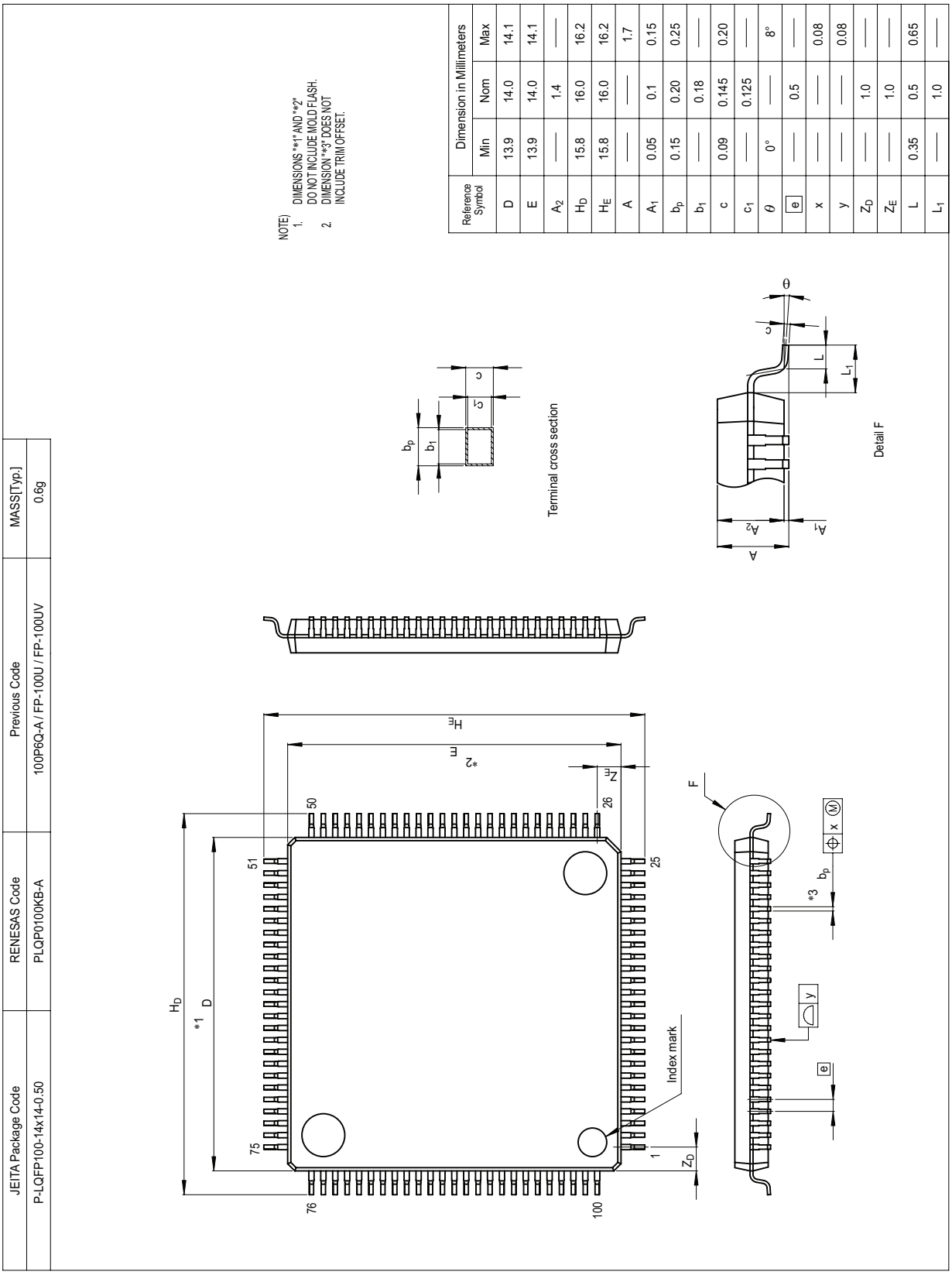
有效：存取区域所对应的片选信号 =L，其他片选信号 =H。

附录 C. 型号一览表

表 C.1 型号一览表

产品分类					产品型号	封装 （封装代码）
产品名称	分类	ROM 容量	RAM 容量	工作温度		
SH7147	F-ZTAT 版	256KB	16KB	– 40 ～ +85°C	R5F71474BJ80FPV	P-LQFP100 （FP-100UV）
					R5F71474BD80FPV	
			12KB	– 40 ～ +125°C	R5F71474AK64FPV	
		384KB	16KB	– 40 ～ +85°C	R5F71475BJ80FPV	
				– 40 ～ +125°C	R5F71475AK64FPV	
		512KB		– 40 ～ +85°C	R5F71476BJ80FPV	
					R5F71476BD80FPV	
				– 40 ～ +125°C	R5F71476AK64FPV	
SH7142	F-ZTAT 版	256KB	16KB	– 40 ～ +85°C	R5F71424BJ80FPV	
			12KB	– 40 ～ +125°C	R5F71424AK64FPV	
		512KB	16KB	– 40 ～ +85°C	R5F71426BJ80FPV	
					R5F71426BD80FPV	
				– 40 ～ +125°C	R5F71426AK64FPV	

附录 D. 封装尺寸图



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	13.9	14.0	14.1
E	13.9	14.0	14.1
A_2	—	1.4	—
H_D	15.8	16.0	16.2
H_E	15.8	16.0	16.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.15	0.20	0.25
b_1	—	0.18	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Z_D	—	1.0	—
Z_E	—	1.0	—
L	0.35	0.5	0.65
L_1	—	1.0	—

图 D.1 封装尺寸图

索引

A

B

C

D

F

G

H

I

J

K

L

M

N

P

Q

R

S

T

V

W

X

Y

Z

修订记录	SH-7147 群硬件手册

Rev.	发行日	修订内容	
		页	修订处
1.00	2009.03.13	—	初版发行

瑞萨 32 位 RISC 单片机
硬件手册
SH7147 群

Publication Date: Rev.1.00, Mar. 13, 2009
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7858/7898

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

Colophon 6.2

SH7147群

RENESAS

瑞萨电子株式会社

RCJ09B0066-0100