

致尊敬的顾客

关于产品目录等资料中的旧公司名称

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

瑞萨电子公司网址：<http://www.renesas.com>

2010年4月1日
瑞萨电子公司

【发行】瑞萨电子公司（<http://www.renesas.com>）

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

Notice

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

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

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

SH7137 群

瑞萨 32 位 RISC 单片机

SuperH™ RISC engine 族

SH7136

R5F7136

SH7137

R5F7137

Notes regarding these materials

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

注意

本文只是参考译文，前页所载英文版“Cautions”具有正式效力。

关于利用本资料时的注意事项

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

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产品的输入引脚的阻抗一般为高阻抗。如果在开路的状态下运行未使用的引脚，由于感应现象，外加LSI周围的噪声，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。通过外部复位引脚对产品进行复位时，从通电到复位有效之前的期间，不能保证引脚的状态。

同样，使用内部上电复位功能对产品进行复位时，从通电到达到复位产生的一定电压的期间，不能保证引脚的状态。

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）

在地址区域中，有被分配将来用作功能扩展的保留地址（保留区）。因为无法保证存取这些地址时的运行，所以不能对保留地址（保留区）进行存取。

4. 关于时钟

【注意】复位时，请在时钟稳定后解除复位。

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

5. 关于产品间的差异

【注意】在变更不同型号的产品时，请对每一个产品型号进行系统评价测试。

即使是同一个群的单片机，如果产品型号不同，由于内部ROM、版本模式等不同，在电特性范围内有时特性值、动作容限、噪声耐量、噪声辐射量等不同。因此，在变更不同型号的产品时，请对每一个型号的产品进行系统评价测试。

本手册的构成

本手册由以下内容构成：

1. 产品使用时的注意事项
2. 本手册的构成
3. 前言
4. 目录
5. 概要
6. 各功能模块的说明

- CPU 和系统控制
- 内部外围模块

各模块功能说明的构成因各模块而不同，一般由①特点、②输入 / 输出引脚、③寄存器的说明、④运行说明、⑤使用时的注意事项等章节构成。

设计采用本 LSI 的应用系统时，请在充分确认了注意事项的基础上进行设计。

请务必阅读各章中有关说明的注意事项和各章最后的使用时的注意事项（使用时的注意事项根据需要进行记载）。

7. 寄存器一览
8. 电特性
9. 附录
 - 产品型号和封装尺寸图等
10. 本版中修订或者追加的部分（只适用于修订版）

修订记录汇总了对前版内容进行修改和追加的主要部分，并非全部的修订内容。请在本手册的正文中确认有关详细内容。

11. 索引

前言

SH7136/SH7137 是以瑞萨独创的 RISC 方式 CPU 为核心，聚集了系统结构所需外围功能的 RISC 单片机。

对 象 本手册是以“采用 SH7136/SH7137 的应用系统”的用户为对象。
使用本手册的读者需要具备电路、逻辑电路以及单片机的基础知识。

目 的 本手册是以“能让用户理解 SH7136/SH7137 的硬件功能和电特性”为目的。
执行指令的详细内容记述在《SH-1/ SH-2/SH-DSP 软件手册》，请对照阅读。

阅读方法

- 希望了解全部功能时
→ 请按照目录的顺序进行阅读。
本书大致按 CPU、系统控制功能、外围功能、电特性等的顺序进行分类。
- 希望了解 FPU 功能的详细内容时
→ 请参照《SH-1/ SH-2/SH-DSP 软件手册》。
- 知道寄存器名，希望了解详细功能时
→ 在本手册的后面附有“索引”，请从索引检索页号。
地址、位内容及其初始化汇总在“第25章 寄存器一览”。

范 例 寄存器的表示：如同或者类似于串行通信接口等的功能存在于多个通道时，使用以下的表示方法：

XXX_N (XXX 为基本寄存器名，N 为通道号)

位的表示 : 以左侧为高位、右侧为低位的顺序进行表示。

数字的表示 : 2 进制数为 B'XXXX，16 进制数为 H'XXXX，10 进制数为 XXXX。

信号的表示 : 对低电平有效的信号加上划线 (XXXX)。

相关资料一览 最新的资料刊登在网站上，请确认现有的资料是否为最新版 (<http://www.renesas.com>)。

- 有关SH7136/SH7137的用户手册

资料名	资料编号
SH7137 群硬件手册	本手册
SH-1/ SH-2/SH-DSP 软件手册	RCJ09B0065

- 有关开发工具的用户手册

资料名	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 Use's Manual	REJ10B0025
SuperH RISC engine High-performance Embedded Workshop 3 Tutorial	REJ10B0023

- 应用说明

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

所有的商标及注册商标分别归属于其所有者。

目 录

第 1 章	概要	1
1.1	SH7136/SH7137 的特点	1
1.2	框图	5
1.3	引脚排列图	6
1.4	引脚功能	8
第 2 章	CPU	12
2.1	特点	12
2.2	寄存器结构	13
2.2.1	通用寄存器 (Rn)	14
2.2.2	控制寄存器	14
2.2.3	系统寄存器	15
2.2.4	寄存器的初始值	15
2.3	数据格式	15
2.3.1	寄存器的数据格式	15
2.3.2	存储器的数据格式	16
2.3.3	立即数的数据格式	16
2.4	指令特点	16
2.4.1	RISC 方式	16
2.4.2	寻址方式	19
2.4.3	指令格式	22
2.5	指令系统	24
2.5.1	指令系统的分类	24
2.5.2	数据传送指令	26
2.5.3	算术运算指令	27
2.5.4	逻辑运算指令	29
2.5.5	移位指令	29
2.5.6	转移指令	30
2.5.7	系统控制指令	31
2.6	处理状态	32
第 3 章	MCU 运行模式	33
3.1	运行模式的选择	33
3.2	输入 / 输出引脚	33
3.3	各运行模式的说明	33
3.3.1	模式 0 (MCU 扩展模式 0)	33
3.3.2	模式 2 (MCU 扩展模式 2)	34
3.3.3	模式 3 (单芯片模式)	34
3.4	地址映像	34
3.5	本 LSI 的初始状态	36
3.6	更改运行模式时的注意事项	36
第 4 章	时钟振荡器 (CPG)	37
4.1	特点	37
4.2	输入 / 输出引脚	40
4.3	时钟工作模式	40
4.4	寄存器说明	43
4.4.1	频率控制寄存器 (FRQCR)	43
4.4.2	振荡停止检测控制寄存器 (OSCCR)	45
4.5	更改频率的方法	45

4.6	振荡器	46
4.6.1	连接晶体谐振器的方法	46
4.6.2	输入外部时钟的方法	46
4.7	振荡停止的检测功能	47
4.8	使用时的注意事项	47
4.8.1	有关谐振器的注意事项	47
4.8.2	电路板设计的注意事项	47
第 5 章	异常处理	49
5.1	概要	49
5.1.1	异常处理的种类和优先级	49
5.1.2	异常处理的运行	49
5.1.3	异常处理向量表	50
5.2	复位	51
5.2.1	复位的种类	51
5.2.2	上电复位	52
5.2.3	手动复位	52
5.3	地址错误	53
5.3.1	地址错误的发生源	53
5.3.2	地址错误的异常处理	53
5.4	中断	54
5.4.1	中断源	54
5.4.2	中断优先级	54
5.4.3	中断异常处理	55
5.5	指令的异常	55
5.5.1	指令异常的种类	55
5.5.2	陷阱指令	55
5.5.3	槽非法指令	55
5.5.4	一般非法指令	56
5.6	异常处理的接受	56
5.7	异常处理后的堆栈状态	57
5.8	使用时的注意事项	57
5.8.1	堆栈指针 (SP) 的值	57
5.8.2	向量基址寄存器 (VBR) 的值	57
5.8.3	地址错误异常处理的堆栈存取中发生的地址错误	57
5.8.4	有关槽非法指令异常处理的注意事项	58
第 6 章	中断控制器 (INTC)	59
6.1	特点	59
6.2	输入 / 输出引脚	60
6.3	寄存器说明	60
6.3.1	中断控制寄存器 0 (ICR0)	61
6.3.2	IRQ 控制寄存器 (IRQCR)	61
6.3.3	IRQ 状态寄存器 (IRQSR)	62
6.3.4	中断优先级寄存器 A、D ~ F、H ~ M (IPRA、IPRD ~ IPRF、IPRH ~ IPRM)	64
6.4	中断源	66
6.4.1	外部中断源	66
6.4.2	内部外围模块中断	66
6.4.3	用户断点中断	67
6.5	中断异常处理向量表	67
6.6	运行说明	69
6.6.1	中断运行的流程	69
6.6.2	中断异常处理结束后的堆栈状态	71

6.7	中断响应时间	71
6.8	通过中断请求信号进行的数据传送	72
6.8.1	中断请求信号作为 DTC 启动源和 CPU 中断源的情况	73
6.8.2	中断请求信号作为 DTC 启动源而不作为 CPU 中断源的情况	73
6.8.3	中断请求信号作为 CPU 中断源而不作为 DTC 启动源	73
6.9	使用时的注意事项	73
第 7 章	用户断点控制器 (UBC)	74
7.1	特点	74
7.2	输入 / 输出引脚	75
7.3	寄存器说明	76
7.3.1	断点地址寄存器 A (BARA)	76
7.3.2	断点地址屏蔽寄存器 A (BAMRA)	77
7.3.3	断点总线周期寄存器 A (BBRA)	77
7.3.4	断点数据寄存器 A (BDRA)	78
7.3.5	断点数据屏蔽寄存器 A (BDMRA)	79
7.3.6	断点地址寄存器 B (BARB)	79
7.3.7	断点地址屏蔽寄存器 B (BAMRB)	80
7.3.8	断点数据寄存器 B (BDRB)	80
7.3.9	断点数据屏蔽寄存器 B (BDMRB)	81
7.3.10	断点总线周期寄存器 B (BBRB)	81
7.3.11	断点控制寄存器 (BRCCR)	83
7.3.12	执行次数断点寄存器 (BETR)	85
7.3.13	转移源寄存器 (BRSR)	85
7.3.14	转移目标寄存器 (BRDR)	86
7.4	运行说明	87
7.4.1	用户断点运行的流程	87
7.4.2	取指令周期的用户断点	87
7.4.3	数据存取周期的用户断点	88
7.4.4	顺序断点	88
7.4.5	被保存的程序计数器值	89
7.4.6	PC 跟踪	89
7.4.7	使用例子	90
7.5	使用时的注意事项	93
第 8 章	数据传送控制器 (DTC)	94
8.1	特点	94
8.2	寄存器说明	96
8.2.1	DTC 模式寄存器 A (MRA)	96
8.2.2	DTC 模式寄存器 B (MRB)	97
8.2.3	DTC 源地址寄存器 (SAR)	98
8.2.4	DTC 目标地址寄存器 (DAR)	98
8.2.5	DTC 传送计数寄存器 A (CRA)	99
8.2.6	DTC 传送计数寄存器 B (CRB)	99
8.2.7	DTC 允许寄存器 A ~ E (DTCERA ~ DTCERE)	99
8.2.8	DTC 控制寄存器 (DTCCR)	100
8.2.9	DTC 向量基址寄存器 (DTCVBR)	101
8.2.10	总线功能扩展寄存器 (BSCEHR)	101
8.3	启动源	102
8.4	传送信息的分配和 DTC 向量表	102
8.5	运行说明	105
8.5.1	传送信息的跳读功能	108
8.5.2	传送信息的回写省略功能	108

8.5.3	正常传送模式	108
8.5.4	重复传送模式	109
8.5.5	块传送模式	110
8.5.6	链传送	111
8.5.7	运行时序	111
8.5.8	DTC 的执行状态	113
8.5.9	DTC 的总线权释放时序	114
8.5.10	DTC 启动优先级的设定	116
8.6	通过中断启动 DTC	117
8.7	DTC 使用例	118
8.7.1	正常传送	118
8.7.2	计数器 =0 的链传送	118
8.8	中断源	119
8.9	使用时的注意事项	119
8.9.1	模块待机模式的设定	119
8.9.2	内部 RAM	119
8.9.3	DTCE 位的设定	119
8.9.4	链传送	119
8.9.5	传送信息的起始地址 / 源地址 / 目标地址	120
8.9.6	通过 DTC 存取 DTC 寄存器	120
8.9.7	IRQ 中断作为 DTC 传送源时的注意事项	120
8.9.8	SCI 作为 DTC 启动源时的注意事项	120
8.9.9	中断源标志的清除	120
8.9.10	NMI 中断和 DTC 启动的竞争	120
8.9.11	中途取消 DTC 启动请求时的运行	120
第 9 章	总线状态控制器 (BSC)	121
9.1	特点	121
9.2	输入 / 输出引脚	122
9.3	区域概要	122
9.3.1	空间分割	122
9.3.2	地址映像	122
9.4	寄存器说明	123
9.4.1	共用控制寄存器 (CMNCR)	124
9.4.2	CSn 空间总线控制寄存器 (CSnBCR) (n=0、1)	124
9.4.3	CSn 空间等待控制寄存器 (CSnWCR) (n=0、1)	126
9.4.4	总线功能扩展寄存器 (BSCEHR)	127
9.5	运行说明	130
9.5.1	字节排序 / 存取长度和数据调整	130
9.5.2	通常空间接口	130
9.5.3	存取等待的控制	133
9.5.4	CSn 有效期间的扩展	134
9.5.5	存取周期之间的等待	134
9.5.6	总线仲裁	136
9.5.7	其他	138
9.5.8	CPU 存取内部 FLASH 和内部 RAM	138
9.5.9	CPU 存取内部外围 I/O 寄存器	138
9.5.10	CPU 存取外部存储器	139
第 10 章	多功能定时器脉冲单元 2 (MTU2)	142
10.1	特点	142
10.2	输入 / 输出引脚	146
10.3	寄存器说明	147

10.3.1	定时器的控制寄存器 (TCR)	150
10.3.2	定时器的模式寄存器 (TMDR)	153
10.3.3	定时器的 I/O 控制寄存器 (TIOR)	154
10.3.4	定时器的比较匹配清除寄存器 (TCNTCMPCLR)	167
10.3.5	定时器的中断允许寄存器 (TIER)	167
10.3.6	定时器的状态寄存器 (TSR)	170
10.3.7	定时器的缓冲运行传送模式寄存器 (TBTM)	174
10.3.8	定时器的输入捕捉控制寄存器 (TICCR)	175
10.3.9	定时器的同步清除寄存器 (TSYCR)	175
10.3.10	定时器的 A/D 转换开始请求的控制寄存器 (TADCR)	177
10.3.11	定时器的 A/D 转换开始请求的周期设定寄存器 (TADCORA/B_4)	179
10.3.12	定时器的 A/D 转换开始请求的周期设定缓冲寄存器 (TADCOBRA/B_4)	179
10.3.13	定时器的计数器 (TCNT)	179
10.3.14	定时器的通用寄存器 (TGR)	179
10.3.15	定时器的开始寄存器 (TSTR)	180
10.3.16	定时器的同步寄存器 (TSYR)	181
10.3.17	定时器的计数器同步开始寄存器 (TCSYSTR)	182
10.3.18	定时器的读写允许寄存器 (TRWER)	183
10.3.19	定时器的输出主控允许寄存器 (TOER)	184
10.3.20	定时器的输出控制寄存器 1 (TOCR1)	185
10.3.21	定时器的输出控制寄存器 2 (TOCR2)	186
10.3.22	定时器的输出电平缓冲寄存器 (TOLBR)	189
10.3.23	定时器的门控寄存器 (TGCR)	189
10.3.24	定时器的子计数器 (TCNTS)	191
10.3.25	定时器的空载时间数据寄存器 (TDDR)	191
10.3.26	定时器的周期数据寄存器 (TCDR)	191
10.3.27	定时器的周期缓冲寄存器 (TCBR)	191
10.3.28	定时器的中断减少设定寄存器 (TITCR)	192
10.3.29	定时器的中断减少次数计数器 (TITCNT)	193
10.3.30	定时器的缓冲传送设定寄存器 (TBTER)	194
10.3.31	定时器的空载时间允许寄存器 (TDER)	195
10.3.32	定时器的波形控制寄存器 (TWCR)	195
10.3.33	和总线主控的接口	196
10.4	运行说明	197
10.4.1	基本运行	197
10.4.2	同步运行	202
10.4.3	缓冲运行	203
10.4.4	级联运行	206
10.4.5	PWM 模式	210
10.4.6	相位计数模式	213
10.4.7	复位同步 PWM 模式	218
10.4.8	互补 PWM 模式	220
10.4.9	A/D 转换开始请求的延迟功能	250
10.4.10	MTU2-MTU2S 的同步运行	253
10.4.11	外部脉宽的测量功能	256
10.4.12	空载时间的补偿功能	257
10.4.13	互补 PWM 在“波峰/波谷”的 TCNT 捕捉运行	258
10.5	中断源	259
10.5.1	中断源和优先级	259
10.5.2	DTC 的启动	260
10.5.3	A/D 转换器的启动	260
10.6	运行时序	261
10.6.1	输入/输出时序	261

10.6.2	中断信号的时序	267
10.7	使用时的注意事项	271
10.7.1	模块待机模式的设定	271
10.7.2	输入时钟的限制事项	271
10.7.3	周期设定的注意事项	271
10.7.4	TCNT 的写操作和清除的竞争	272
10.7.5	TCNT 的写操作和递增计数的竞争	272
10.7.6	TGR 的写操作和比较匹配的竞争	273
10.7.7	缓冲寄存器的写操作和比较匹配的竞争	273
10.7.8	缓冲寄存器的写操作和清除 TCNT 的竞争	274
10.7.9	TGR 的读操作和输入捕捉的竞争	274
10.7.10	TGR 的写操作和输入捕捉的竞争	275
10.7.11	缓冲寄存器的写操作和输入捕捉的竞争	276
10.7.12	级联中 TCNT_2 的写操作和上溢 / 下溢的竞争	277
10.7.13	互补 PWM 模式停止时的计数器值	278
10.7.14	互补 PWM 模式的缓冲运行设定	278
10.7.15	复位同步 PWM 模式的缓冲运行和比较匹配标志	278
10.7.16	复位同步 PWM 模式的上溢标志	279
10.7.17	上溢 / 下溢和计数器清除的竞争	280
10.7.18	TCNT 的写操作和上溢 / 下溢的竞争	280
10.7.19	从正常模式或者 PWM 模式 1 转移到复位同步 PWM 模式时的注意事项	281
10.7.20	互补 PWM 模式或者复位同步 PWM 模式的输出电平	281
10.7.21	模块待机时的中断	281
10.7.22	级联中 TCNT_1 和 TCNT_2 的同时输入捕捉	281
10.8	MTU2 输出引脚的初始化方法	281
10.8.1	运行模式	281
10.8.2	复位开始时的运行	282
10.8.3	因运行中的异常等而重新设定时的运行	282
10.8.4	因运行中的异常等而初始化引脚的步骤和模式转移的概要	282
第 11 章	多功能定时器脉冲单元 2S (MTU2S)	300
11.1	输入 / 输出引脚	301
11.2	寄存器说明	302
第 12 章	端口输出的允许 (POE)	304
12.1	特点	304
12.2	输入 / 输出引脚	306
12.3	寄存器说明	307
12.3.1	输入电平控制 / 状态寄存器 1 (ICSR1)	307
12.3.2	输出电平控制 / 状态寄存器 1 (OCSR1)	309
12.3.3	输入电平控制 / 状态寄存器 2 (ICSR2)	310
12.3.4	输出电平控制 / 状态寄存器 2 (OCSR2)	312
12.3.5	输入电平控制 / 状态寄存器 3 (ICSR3)	313
12.3.6	软件端口的输出允许寄存器 (SPOER)	314
12.3.7	端口输出的允许控制寄存器 1 (POECR1)	315
12.3.8	端口的输出的允许控制寄存器 2 (POECR2)	315
12.4	运行说明	317
12.4.1	输入电平的检测	318
12.4.2	输出电平的比较	319
12.4.3	高阻抗状态的解除	319
12.5	中断	319
12.6	使用时的注意事项	320
12.6.1	从看门狗定时器产生上电复位时的引脚状态	320

第 13 章	看门狗定时器 (WDT)	321
13.1	特点	321
13.2	输入 / 输出引脚	322
13.3	寄存器说明	322
13.3.1	看门狗定时器的计数器 (WTCNT)	322
13.3.2	看门狗定时器的控制 / 状态寄存器 (WTCSR)	322
13.3.3	寄存器存取时的注意事项	324
13.4	运行说明	324
13.4.1	软件待机模式的解除步骤	324
13.4.2	看门狗定时器模式的用法	324
13.4.3	间隔定时器模式的用法	325
13.5	使用时的注意事项	325
13.5.1	WTCNT 的设定值	325
第 14 章	串行通信接口 (SCI)	326
14.1	特点	326
14.2	输入 / 输出引脚	327
14.3	寄存器说明	328
14.3.1	接收移位寄存器 (SCRSR)	328
14.3.2	接收数据寄存器 (SCRDR)	329
14.3.3	发送移位寄存器 (SCTSR)	329
14.3.4	发送数据寄存器 (SCTDR)	329
14.3.5	串行模式寄存器 (SCSMR)	329
14.3.6	串行控制寄存器 (SCSCR)	331
14.3.7	串行状态寄存器 (SCSSR)	333
14.3.8	串行端口寄存器 (SCSPTR)	335
14.3.9	串行方向控制寄存器 (SCSDCR)	336
14.3.10	位速率寄存器 (SCBRR)	337
14.4	运行说明	343
14.4.1	概要	343
14.4.2	异步模式的运行	344
14.4.3	时钟同步模式的运行	350
14.4.4	多处理器通信功能	357
14.4.5	多处理器串行数据的发送	358
14.4.6	多处理器串行数据的接收	359
14.5	SCI 的中断源和 DTC	361
14.6	串行端口寄存器 (SCSPTR) 和 SCI 引脚的关系	362
14.7	使用时的注意事项	363
14.7.1	有关 SCTDR 的写操作和 TDRE 标志的关系	363
14.7.2	有关同时发生多个接收错误时的运行	363
14.7.3	中止的检测和处理	363
14.7.4	中止的发送	363
14.7.5	异步模式的接收数据采样时序和接收容限	364
14.7.6	DTC 使用时的注意事项	364
14.7.7	时钟同步外部时钟模式的注意事项	365
14.7.8	模块待机模式的设定	365
第 15 章	同步串行通信单元 (SSU)	366
15.1	特点	366
15.2	输入 / 输出引脚	367
15.3	寄存器说明	368
15.3.1	SS 控制寄存器 H (SSCRH)	368
15.3.2	SS 控制寄存器 L (SSCRL)	369

15.3.3	SS 模式寄存器 (SSMR)	370
15.3.4	SS 允许寄存器 (SSER)	371
15.3.5	SS 状态寄存器 (SSSR)	371
15.3.6	SS 控制寄存器 2 (SSCR2)	373
15.3.7	SS 传送数据寄存器 0 ~ 3 (SSTD0 ~ SSTD3)	373
15.3.8	SS 接收数据寄存器 0 ~ 3 (SSRDR0 ~ SSRDR3)	374
15.3.9	SS 移位寄存器 (SSTRSR)	375
15.4	运行说明	375
15.4.1	传送时钟	375
15.4.2	时钟的相位、极性和数据的关系	375
15.4.3	数据输入 / 输出引脚和移位寄存器的关系	376
15.4.4	各通信模式和引脚功能	376
15.4.5	SSU 模式	378
15.4.6	SCS 引脚控制和冲突错误	383
15.4.7	时钟同步通信模式	384
15.5	SSU 的中断源和 DTC	389
15.6	使用时的注意事项	389
15.6.1	模块待机模式的设定	389
15.6.2	SSTD 和 SSRDR 寄存器的存取	389
15.6.3	SSU 从属模式中的连续发送和接收时的注意事项	389
第 16 章	I ² C 总线接口 2 (I ² C2)	390
16.1	特点	390
16.2	输入 / 输出引脚	392
16.3	寄存器说明	392
16.3.1	I ² C 总线控制寄存器 1 (ICCR1)	392
16.3.2	I ² C 总线控制寄存器 2 (ICCR2)	394
16.3.3	I ² C 总线模式寄存器 (ICMR)	395
16.3.4	I ² C 总线中断允许寄存器 (ICIER)	396
16.3.5	I ² C 总线状态寄存器 (ICSR)	397
16.3.6	从属地址寄存器 (SAR)	399
16.3.7	I ² C 总线发送数据寄存器 (ICDRT)	399
16.3.8	I ² C 总线接收数据寄存器 (ICDRR)	399
16.3.9	I ² C 总线移位寄存器 (ICDRS)	400
16.3.10	NF2CYC 寄存器 (NF2CYC)	400
16.4	运行说明	401
16.4.1	I ² C 总线格式	401
16.4.2	主控发送	402
16.4.3	主控接收	403
16.4.4	从属发送	405
16.4.5	从属接收	406
16.4.6	时钟同步串行格式	407
16.4.7	噪声消除电路	410
16.4.8	使用例	411
16.5	I ² C2 的中断源	415
16.6	DTC 的运行	415
16.7	位同步电路	416
16.8	使用时的注意事项	417
16.8.1	模块待机模式的设定	417
16.8.2	停止条件和开始条件 (重新传送) 的发行	417
16.8.3	开始条件与停止条件的连续发行	417
16.8.4	使用多主控时的设定	417
16.8.5	主控接收模式中的 ICDRR 的读操作	417

16.8.6	仿真器的支持	417
第 17 章	A/D 转换器 (ADC)	418
17.1	特点	418
17.2	输入 / 输出引脚	420
17.3	寄存器说明	421
17.3.1	A/D 控制寄存器 _0、_1 (ADCR_0、ADCR_1)	422
17.3.2	A/D 状态寄存器 _0、_1 (ADSR_0、ADSR_1)	423
17.3.3	A/D 开始触发选择寄存器 _0、_1 (ADSTRGR_0、ADSTRGR_1)	424
17.3.4	A/D 模拟输入通道选择寄存器 _0、_1 (ADANSR_0、ADANSR_1)	425
17.3.5	A/D 数据寄存器 0 ~ 15 (ADDR0 ~ ADDR15)	425
17.3.6	和 CPU 的接口	426
17.4	运行说明	426
17.4.1	单周期扫描模式	426
17.4.2	连续扫描模式	427
17.4.3	输入采样和 A/D 转换时间	428
17.4.4	通过 MTU2 和 MTU2S 启动 A/D 转换器	430
17.4.5	外部触发输入时序	430
17.4.6	ADDR 寄存器自动清除功能的使用例子	430
17.5	中断源和 DTC 传送请求	431
17.6	A/D 转换精度的定义	431
17.7	使用时的注意事项	432
17.7.1	模拟输入电压的设定范围	432
17.7.2	AV _{CC} 、AV _{SS} 与 V _{CC} 、V _{SS} 的关系	432
17.7.3	AV _{refh} 和 AV _{refl} 引脚的设定范围	432
17.7.4	电路板设计的注意事项	432
17.7.5	噪声对策的注意事项	433
17.7.6	寄存器设定时的注意事项	433
第 18 章	比较匹配定时器 (CMT)	434
18.1	特点	434
18.2	寄存器说明	435
18.2.1	比较匹配定时器的开始寄存器 (CMSTR)	435
18.2.2	比较匹配定时器的控制 / 状态寄存器 (CMCSR)	436
18.2.3	比较匹配计数器 (CMCNT)	437
18.2.4	比较匹配常数寄存器 (CMCOR)	437
18.3	运行说明	437
18.3.1	周期计数运行	437
18.3.2	CMCNT 的计数时序	437
18.4	中断	438
18.4.1	CMT 的中断源和 DTC	438
18.4.2	比较匹配标志的置位时序	438
18.4.3	比较匹配标志的清除时序	438
18.5	使用时的注意事项	439
18.5.1	模块待机模式的设定	439
18.5.2	CMCNT 的写操作和比较匹配的竞争	439
18.5.3	CMCNT 的字写和递增计数的竞争	439
18.5.4	CMCNT 的字节写和递增计数的竞争	440
18.5.5	CMCNT 和 CMCOR 的比较匹配	440
第 19 章	控制器局域网 (RCAN-ET)	441
19.1	特点	441
19.2	结构	442

19.2.1	框图	442
19.2.2	各块的功能	443
19.2.3	引脚结构	443
19.2.4	存储器映像	444
19.3	邮箱	444
19.3.1	邮箱的结构	444
19.3.2	信息控制字段	446
19.3.3	局部接收过滤屏蔽 (LAFM)	449
19.3.4	信息数据字段	449
19.4	RCAN-ET 的控制寄存器	450
19.4.1	主控控制寄存器 (MCR)	450
19.4.2	通用状态寄存器 (GSR)	454
19.4.3	位结构寄存器 0、1 (BCR0、BCR1)	456
19.4.4	中断请求寄存器 (IRR)	459
19.4.5	中断屏蔽寄存器 (IMR)	462
19.4.6	发送错误计数器 (TEC) / 接收错误计数器 (REC)	463
19.5	RCAN-ET 的邮箱寄存器	464
19.5.1	发送等待寄存器 1、0 (TXPR1、TXPR0)	465
19.5.2	发送取消寄存器 0 (TXCR0)	466
19.5.3	发送应答寄存器 0 (TXACK0)	467
19.5.4	中止应答寄存器 0 (ABACK0)	467
19.5.5	数据帧接收完成寄存器 0 (RXPR0)	468
19.5.6	远程帧接收完成寄存器 0 (RFPR0)	468
19.5.7	邮箱中断屏蔽寄存器 0 (MBIMR0)	469
19.5.8	未读信息状态寄存器 0 (UMSR0)	469
19.6	运行说明	470
19.6.1	RCAN-ET 的设定	470
19.6.2	测试模式的设定	473
19.6.3	信息的发送顺序	474
19.6.4	信息的接收顺序	476
19.6.5	邮箱的重新设定	477
19.7	中断源	478
19.8	DTC 接口	479
19.9	CAN 总线接口	480
19.10	使用时的注意事项	481
19.10.1	模块停止模式	481
19.10.2	复位	481
19.10.3	CAN 睡眠模式	481
19.10.4	寄存器的存取	481
19.10.5	中断	481
第 20 章	引脚功能控制器 (PFC)	482
20.1	寄存器说明	495
20.1.1	端口 A 的 IO 寄存器 L (PAIORL)	495
20.1.2	端口 A 的控制寄存器 L1 ~ L4 (PACRL1 ~ PACRL4)	496
20.1.3	端口 B 的 IO 寄存器 L (PBIORL)	504
20.1.4	端口 B 的控制寄存器 L1、L2 (PBCRL1、PBCRL2)	505
20.1.5	端口 D 的 IO 寄存器 L (PDIORL) (只限 SH7137)	509
20.1.6	端口 D 的控制寄存器 L1 ~ L3 (PDCRL1 ~ PDCRL3) (只限 SH7137)	509
20.1.7	端口 E 的 IO 寄存器 L、H (PEIORL 和 PEIORH)	512
20.1.8	端口 E 的控制寄存器 L1 ~ L4、H1、H2 (PECRL1 ~ PECRL4、PECRH1 和 PECRH2)	513
20.1.9	IRQOUT 功能的控制寄存器 (IFCR)	523
20.2	使用时的注意事项	524

第 21 章	I/O 端口	525
21.1	端口 A	525
21.1.1	寄存器说明	526
21.1.2	端口 A 的数据寄存器 L (PADRL)	526
21.1.3	端口 A 的端口寄存器 L (PAPRL)	527
21.2	端口 B	529
21.2.1	寄存器说明	529
21.2.2	端口 B 的数据寄存器 L (PBDRL)	529
21.2.3	端口 B 的端口寄存器 L (PBPRL)	531
21.3	端口 D (只限 SH7137)	533
21.3.1	寄存器说明	533
21.3.2	端口 D 的数据寄存器 L (PDDRL)	533
21.3.3	端口 D 的端口寄存器 L (PDPRL)	534
21.4	端口 E	535
21.4.1	寄存器说明	537
21.4.2	端口 E 的数据寄存器 H、L (PEDRH、PEDRL)	537
21.4.3	端口 E 的端口寄存器 H、L (PEPRH、PEPRL)	538
21.5	端口 F	540
21.5.1	寄存器说明	541
21.5.2	端口 F 的数据寄存器 L (PFDRL)	541
第 22 章	闪存	543
22.1	特点	543
22.2	概要	544
22.2.1	框图	544
22.2.2	运行模式	545
22.2.3	模式比较	546
22.2.4	闪存结构	546
22.2.5	块分割	547
22.2.6	编程 / 擦除的接口	548
22.3	输入 / 输出引脚	549
22.4	寄存器说明	550
22.4.1	寄存器一览表	550
22.4.2	编程 / 擦除接口寄存器	551
22.4.3	编程 / 擦除接口参数	555
22.4.4	RAM 仿真寄存器 (RAMER)	563
22.5	板上编程模式	563
22.5.1	引导模式	564
22.5.2	用户编程模式	566
22.5.3	用户引导模式	573
22.6	保护	576
22.6.1	硬件保护	576
22.6.2	软件保护	576
22.6.3	错误保护	577
22.7	通过 RAM 对闪存进行的仿真	578
22.8	使用时的注意事项	580
22.8.1	用户 MAT 和用户引导 MAT 的转换	580
22.8.2	编程 / 擦除过程中的中断	581
22.8.3	其他注意事项	582
22.9	附录	583
22.9.1	引导模式的标准串行通信接口规格	583
22.9.2	过程程序和编程数据的可保存区域	600
22.10	编程器模式	604

第 23 章	RAM	605
23.1	使用时的注意事项	605
23.1.1	模块待机模式的设定	605
23.1.2	地址错误	605
23.1.3	RAM 的初始值	605
第 24 章	低功耗模式	606
24.1	特点	606
24.1.1	低功耗模式的种类	606
24.2	输入 / 输出引脚	607
24.3	寄存器说明	607
24.3.1	待机控制寄存器 1 (STBCR1)	607
24.3.2	待机控制寄存器 2 (STBCR2)	608
24.3.3	待机控制寄存器 3 (STBCR3)	608
24.3.4	待机控制寄存器 4 (STBCR4)	609
24.3.5	待机控制寄存器 5 (STBCR5)	610
24.3.6	待机控制寄存器 6 (STBCR6)	611
24.3.7	RAM 控制寄存器 (RAMCR)	612
24.4	睡眠模式	613
24.4.1	向睡眠模式的转移	613
24.4.2	睡眠模式的解除	613
24.5	软件待机模式	613
24.5.1	向软件待机模式的转移	613
24.5.2	软件待机模式的解除	613
24.6	深度软件待机模式	614
24.6.1	向深度软件待机模式的转移	614
24.6.2	深度软件待机模式的解除	614
24.7	模块待机功能	614
24.7.1	向模块待机功能的转移	614
24.7.2	模块待机功能的解除	614
24.8	使用时的注意事项	615
24.8.1	振荡稳定待机中的消耗电流	615
24.8.2	深度软件待机模式	615
24.8.3	执行 SLEEP 指令时	615
第 25 章	寄存器一览	616
25.1	寄存器地址一览 (地址顺序)	617
25.2	寄存器位一览	632
25.3	各运行模式的寄存器状态	648
第 26 章	电特性	658
26.1	绝对最大额定值	658
26.2	DC 特性	659
26.3	AC 特性	663
26.3.1	时钟时序	663
26.3.2	控制信号的时序	665
26.3.3	AC 总线的时序规格	668
26.3.4	多功能定时器脉冲单元 2 (MTU2) 的时序	673
26.3.5	多功能定时器脉冲单元 2S (MTU2S) 的时序	674
26.3.6	I/O 端口的时序	675
26.3.7	看门狗定时器 (WDT) 的时序	675
26.3.8	串行通信接口 (SCI) 的时序	676
26.3.9	串行通信单元 (SSU) 的时序	678

26.3.10	控制器局域网 (RCAN-ET) 的时序	681
26.3.11	端口输出允许 (POE) 的时序	681
26.3.12	I ² C 总线接口 2 (I ² C2) 的时序	682
26.3.13	UBC 的触发时序	683
26.3.14	A/D 转换器的时序	683
26.3.15	AC 特性的测量条件	684
26.4	A/D 转换器特性	684
26.5	闪存特性	685
26.6	使用时的注意事项	685
26.6.1	V _{CL} 电容的连接方法	685
附录		686
附录 A.	引脚状态	686
附录 B.	有关总线信号的引脚状态	690
附录 C.	型号一览	691
附录 D.	封装尺寸图	692
索引		693

第 1 章 概要

1.1 SH7136/SH7137 的特点

本 LSI 是以瑞萨科技的 RISC 方式 CPU 为内核，集聚了系统构成所需外围功能的 RISC 单片机。

本 LSI 的 CPU 有 RISC（Reduced Instruction Set Computer）方式的指令系统，因为以 1 条指令 1 个状态（1 个系统时钟周期）执行基本指令，所以飞跃性地提高了指令的执行速度。另外，采用了内部 32 位系统结构，增强了数据处理能力。对于以往的单片机不能实现的高速性实时控制等应用系统，通过本 LSI 的 CPU 能构成低成本、高性能和高功能的系统。

本 LSI 内置了大容量 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、I²C 总线接口 2（I²C2）和控制器局域网（RCAN-ET）等系统构成所需的外围功能。

另外，本 LSI 通过外部存储器的存取支持功能，能直接连接存储器和外围 LSI（只限 SH7137），因此能大幅度地降低系统成本。

内部 ROM 是内置闪存的 F-ZTATTM（Flexible Zero Turn Around Time）版*。除了能使用支持本 LSI 的编程器对闪存进行编程外，还能通过软件进行编程和擦除，因此用户能在 LSI 安装于电路板上的状态下进行改写。

本 LSI 的特点如表 1.1 所示。

【注】 * F-ZTATTM 是（株）瑞萨科技的商标。

表 1.1 SH7136/SH7137 的特点

项目	特点
CPU	32 位 RISC (Reduced Instruction Set Computer) 方式 CPU - 指令长度: 通过固定为 16 位来提高编码效率 - 加载 / 存储体系结构 (在寄存器之间进行基本运算) - 通用寄存器: 32 位 × 16 个 - 流水线: 5 段流水线方式 - 内部乘法器: 以 2 ~ 5 个周期进行 32×32→64 的乘法运算 - 基本指令: 62 种面向 C 语言的指令系统 【注】 必须注意: 槽非法指令的规格和传统的 SH-2 有所不同。详细内容请参照 “5.8.4 有关槽非法指令异常处理的注意事项”。
运行模式	- 运行模式 - 单芯片模式 - 扩展 ROM 有效模式 (只限 SH7137) - 扩展 ROM 无效模式 (只限 SH7137) - 处理状态 - 程序执行状态 - 异常处理状态 - 总线权释放状态 (只限 SH7137) - 低功耗状态 - 睡眠模式 - 软件待机模式 - 深度软件待机模式 - 模块待机模式
用户断点控制器 (UBC)	- 能将地址、数值、存取类型和数据长度全部作为断点条件进行设定 - 支持连续断点功能 2 个断点通道
内部 ROM	256K 字节
内部 RAM	16K 字节
总线状态控制器 (BSC)	- 分别支持最大 1M 字节的 2 个区域 (CS0 和 CS1) 的地址空间 (只限 SH7137) 8 位外部总线 (只限 SH7137) - 各区域能独立设定以下功能: - 存取的等待周期数 - 空闲等待周期的设定 - 支持 SRAM - 将片选信号输出到对应的区域
数据传送控制器 (DTC)	- 能通过外围 I/O 的中断请求进行独立于 CPU 的数据传送 - 能按各中断源设定传送模式 (设定存储器中的传送模式) - 能对 1 个启动源进行多个数据的传送 - 丰富的传送模式 - 能选择正常模式 / 重复模式 / 块传送模式 - 能将传送单位设定为字节 / 字 / 长字 - 向 CPU 请求启动 DTC 的中断 - 能在 1 次数据传送结束后向 CPU 产生中断 - 能在指定的数据传送全部结束后向 CPU 产生中断

项目	特点
中断控制器 (INTC)	5 个外部中断引脚 (NMI、IRQ3 ~ IRQ0) - 内部外围中断: 设定各模块的优先级 - 向量地址: 各中断源固有的向量地址
用户调试接口 (H-UDI)	支持 E10A 仿真器
时钟振荡器 (CPG)	- 时钟模式: 能选择外部输入或者晶体振荡器作为输入时钟 - 生成 5 种时钟 - CPU 时钟: 80MHz (Max.) - 总线时钟: 40MHz (Max.) - 外围时钟: 40MHz (Max.) - MTU2 专用时钟: 40MHz (Max.) - MTU2S 专用时钟: 80MHz (Max.)
看门狗定时器 (WDT)	1 个通道的看门狗定时器 - 能请求中断
多功能定时器脉冲单元 2 (MTU2)	- 以 6 个通道的 16 位定时器为基础, 最多能输入 / 输出 16 种脉冲以及输入 3 个脉冲 21 个输出比较匹配寄存器兼输入捕捉寄存器 - 共有 21 个独立的比较器 - 能选择 8 种计数器输入时钟 - 输入捕捉功能 - 脉冲输出模式 - 交替 / PWM / 互补 PWM / 复位同步 PWM - 多个计数器的同步功能 - 互补 PWM 输出模式 - 输出用于控制 6 相反相器的非重叠波形 - 自动设定空载时间 - 能在 0 ~ 100% 范围内任意设定 PWM 占空比 - 输出 OFF 功能 - A/D 转换请求的延迟功能 - 空载时间的补偿功能 - 波峰 / 波谷的中断减少功能 - 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 - 能对 2 相编码器进行计数处理
多功能定时器脉冲单元 2S (MTU2S)	- 只限 MTU2 的通道 3、4、5 的子集版 - 最大工作频率为 80MHz
端口输出允许 (POE)	MTU2/MTU2S 波形输出引脚的高阻抗控制
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

项目	特点
同步串行通信单元 (SSU)	- 能选择主控模式 / 从属模式 - 能选择标准模式 / 双向模式 - 能从 8/16/32 位中选择接收和发送数据的长度 - 能同时接收和发送 (全双工) - 能连续进行串行通信 1 个通道
I ² C 总线接口 2 (I ² C2)	- 符合 Philips 公司提倡的 I²C 总线接口方式 - 内置主控模式 / 从属模式 - 能连续接收和发送 - 能选择 I²C 总线格式 / 时钟同步串行格式 1 个通道
控制器局域网 (RCAN-ET)	- CAN 版本: 对应 Bosch 2.0B active - 缓冲器个数: 接收 / 发送 ×15 个、接收专用 ×1 个 1 个通道
A/D 转换器 (ADC)	- SH7137: 12 位 ×16 个通道, SH7136: 12 位 ×12 个通道 - 能通过外部触发或者 MTU2/MTU2S 请求转换 - 内置 3 个单元 ×2 组的采样和保持功能 (1 组能同时进行 3 个通道的采样)
I/O 端口	- SH7137: 57 个通用输入 / 输出引脚和 16 个通用输入引脚 - SH7136: 44 个通用输入 / 输出引脚和 12 个通用输入引脚 - 能对输入 / 输出兼用端口按位进行输入 / 输出的转换
封装	- SH7137: LQFP1414-100 (0.5 节距) - SH7136: LQFP1414-80 (0.65 节距)
电源电压	V_{CC}: 3.0 ~ 3.6V 或者 4.0 ~ 5.5V, AV_{CC}: 4.5 ~ 5.5V

1.2 框图

SH7136/SH7137 的框图如图 1.1 所示。

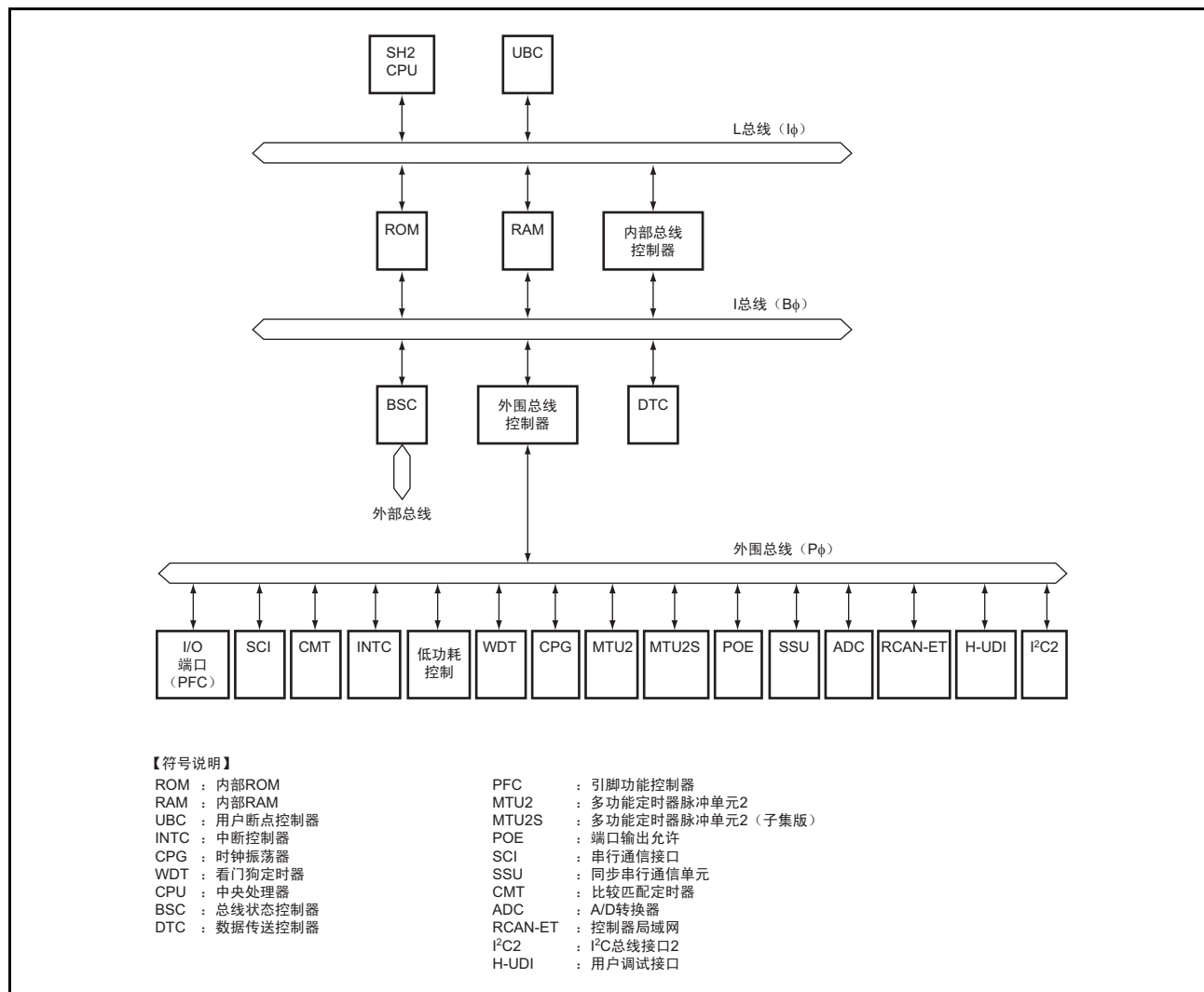


图 1.1 框图

1.3 引脚排列图

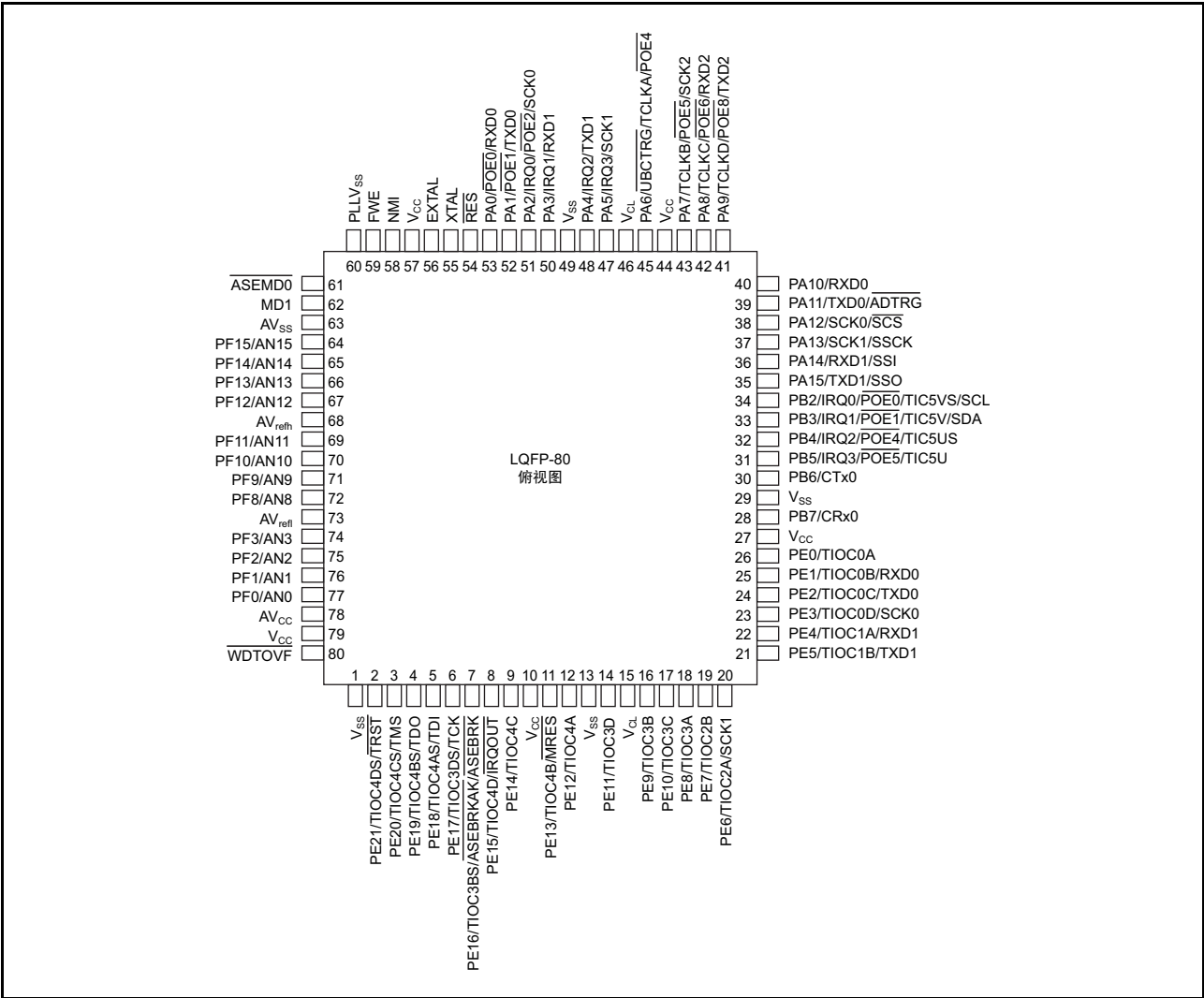


图 1.2 SH7136 的引脚排列图

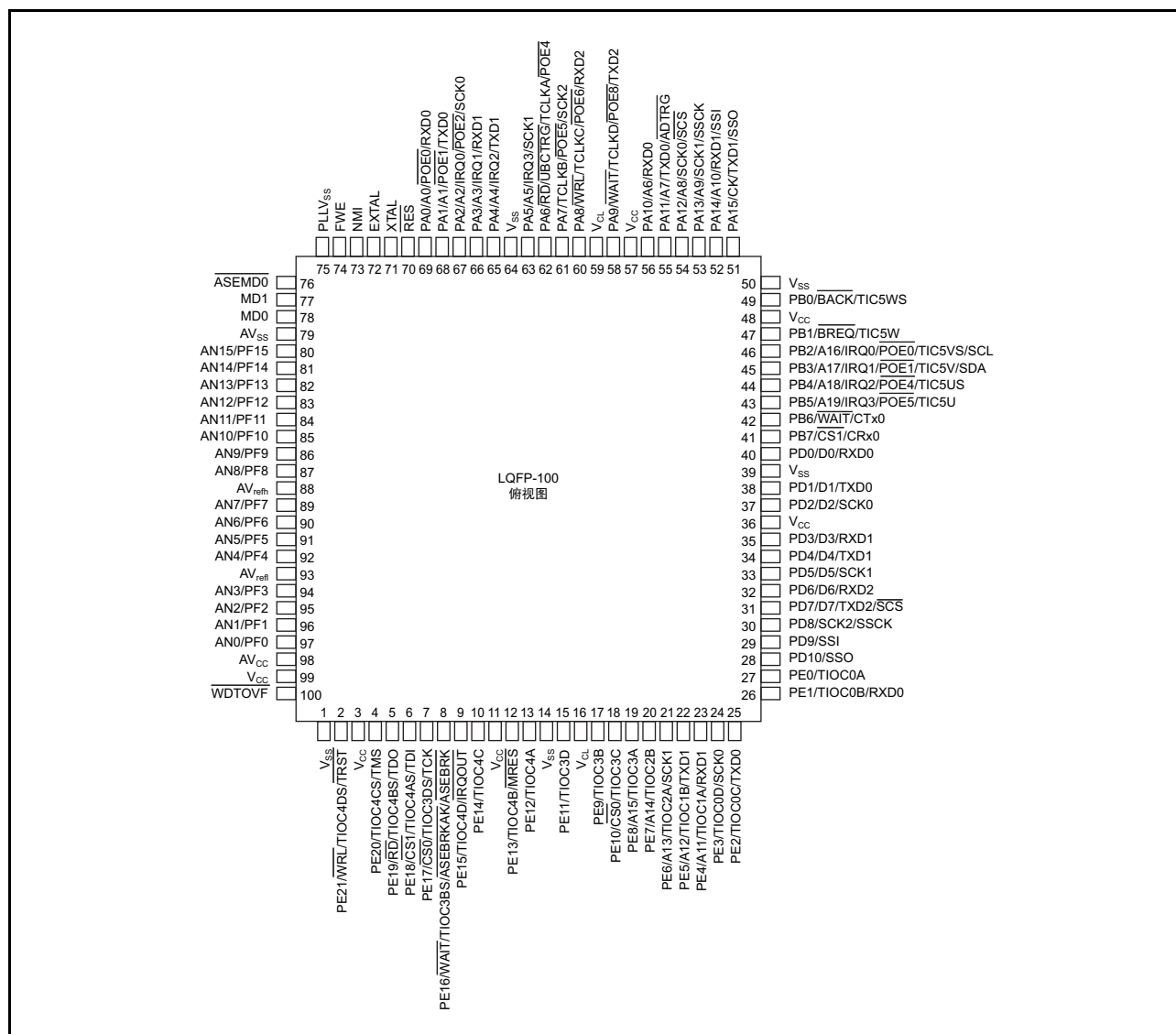


图 1.3 SH7137 的引脚排列图

1.4 引脚功能

各引脚的功能如表 1.2 所示。

表 1.2 引脚功能

分类	引脚名	输入 / 输出	名称	功能
电源	VCC	输入	电源	是电源引脚。必须将全部 VCC 引脚连接到系统电源。如果有开路引脚就不工作。
	VSS	输入	接地	是接地引脚。必须将全部 VSS 引脚连接到系统电源 (0V)。如果有开路引脚就不工作。
	VCL	输出	内部降压电源	是内部降压电源的外接电容引脚。必须通过 0.47μF 的电容器将全部 VCL 引脚连接到 VSS (靠近引脚)。
时钟	PLL $\overline{\text{VSS}}$	输入	PLL 接地	是内部 PLL 振荡器的接地引脚。
	EXTAL	输入	外部时钟	连接晶体振荡器, 也能输入外部时钟。
	XTAL	输出	晶体	连接晶体振荡器。
	CK	输出	系统时钟	给外部器件提供系统时钟。 SH7136 无此引脚。
运行模式控制	MD1、MD0	输入	模式设定	设定运行模式。在运行中不能改变这些引脚的信号。 SH7136 只有 MD1。
	FWE	输入	闪存的编程允许	是闪存的引脚。能保护闪存的编程 / 擦除。
系统控制	$\overline{\text{RES}}$	输入	上电复位	当此引脚为低电平时, 进入上电复位状态。
	$\overline{\text{MRES}}$	输入	手动复位	当此引脚为低电平时, 进入手动复位状态。
	$\overline{\text{WDTOVF}}$	输出	看门狗定时器的上溢	是 WDT 的上溢输出信号。 如果需要下拉, 就必须使用至少 1MΩ 的电阻。
	$\overline{\text{BREQ}}$	输入	总线权请求	在外部器件请求释放总线权时, 将此引脚置为低电平。此引脚只限于 SH7137。
	$\overline{\text{BACK}}$	输出	总线权请求的应答	表示已将总线权释放给外部器件。输出 $\overline{\text{BREQ}}$ 信号的器件能在接受到 $\overline{\text{BACK}}$ 信号后得知已获得总线权。此引脚只限于 SH7137。
中断	NMI	输入	非屏蔽中断	是非屏蔽中断的请求引脚。在不使用时必须固定为高电平或者低电平。
	IRQ3 ~ IRQ0	输入	中断请求 3 ~ 0	是可屏蔽中断的请求引脚。 能选择电平输入或者边沿输入。在选择边沿输入时, 能选择上升沿、下降沿或者双边沿。
	$\overline{\text{IRQOUT}}$	输出	中断请求的输出	表示已产生中断源。也能在总线释放时得知中断的发生。
地址总线	A19 ~ A0	输出	地址总线	输出地址。 此引脚只限于 SH7137。
数据总线	D7 ~ D0	输入 / 输出	数据总线	是 8 位双向总线。 此引脚只限于 SH7137。
总线控制	$\overline{\text{CS1}}$ 、 $\overline{\text{CS0}}$	输出	片选 1、0	是外部存储器或者器件的片选信号。 此引脚只限于 SH7137。
	$\overline{\text{RD}}$	输出	读	表示读外部器件。 此引脚只限于 SH7137。

分类	引脚名	输入 / 输出	名称	功能
总线控制	$\overline{WRL}$	输出	写	表示写外部数据的 bit7 ~ 0。 此引脚只限于 SH7137。
	$\overline{WAIT}$	输入	等待	是在存取外部空间时的总线周期中插入等待周期的输入信号。 此引脚只限于 SH7137。
多功能定时器 脉冲单元 2 (MTU2)	TCLKA、 TCLKB、 TCLKC、 TCKLD	输入	MTU2 定时器的 时钟输入	是定时器的外部时钟输入引脚。
	TIOC0A、 TIOC0B、 TIOC0C、 TIOC0D	输入 / 输出	MTU2 输入捕捉 / 输出比较 (通道 0)	是 TGRA_0 ~ TGRD_0 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。
	TIOC1A、 TIOC1B	输入 / 输出	MTU2 输入捕捉 / 输出比较 (通道 1)	是 TGRA_1 和 TGRB_1 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。
	TIOC2A、 TIOC2B	输入 / 输出	MTU2 输入捕捉 / 输出比较 (通道 2)	是 TGRA_2 和 TGRB_2 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。
	TIOC3A、 TIOC3B、 TIOC3C、 TIOC3D	输入 / 输出	MTU2 输入捕捉 / 输出比较 (通道 3)	是 TGRA_3 ~ TGRD_3 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。
	TIOC4A、 TIOC4B、 TIOC4C、 TIOC4D	输入 / 输出	MTU2 输入捕捉 / 输出比较 (通道 4)	是 TGRA_4 ~ TGRD_4 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。
	TIC5U、 TIC5V、 TIC5W	输入	MTU2 输入捕捉 (通道 5)	是 TGRU_5、TGRV_5 和 TGRW_5 的输入捕捉的输入引脚。(TIC5W 只限于 SH7137)。
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)	是 TGRB_3S 和 TGRD_3S 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。
	TIOC4AS、 TIOC4BS、 TIOC4CS、 TIOC4DS	输入 / 输出	MTU2S 输入捕捉 / 输出比较 (通道 4)	是 TGRA_4S ~ TGRD_4S 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚。
	TIC5US、 TIC5VS、 TIC5WS	输入	MTU2S 输入捕捉 (通道 5)	是 TGRU_5S、TGRV_5S 和 TGRW_5S 的输入捕捉的输入引脚 (TIC5WS 只限于 SH7137)。
端口输出允许 (POE)	$\overline{POE8}$ 、 $\overline{POE6} \sim \overline{POE4}$ 、 $\overline{POE2} \sim \overline{POE0}$	输入	端口的输出控制	是将 MTU2/MTU2S 波形输出引脚置为高阻抗状态的请求信号输入引脚。

分类	引脚名	输入 / 输出	名称	功能
串行通信接口 (SCI)	TXD2 ~ TXD0	输出	发送数据	是发送数据的引脚。
	RXD2 ~ RXD0	输入	接收数据	是接收数据的引脚。
	SCK2 ~ SCK0	输入 / 输出	串行时钟	是时钟的输入 / 输出引脚。
同步串行通信 单元 (SSU)	SSO	输入 / 输出	数据	是数据的输入 / 输出引脚。
	SSI	输入 / 输出	数据	是数据的输入 / 输出引脚。
	SCK	输入 / 输出	时钟	是时钟的输入 / 输出引脚。
	SCS	输入 / 输出	片选	是片选信号的输入 / 输出引脚。
控制器局域网 (RCAN-ET)	CTx0	输出	发送数据	是 CAN 总线的发送引脚。
	CRx0	输入	接收数据	是 CAN 总线的接收引脚。
I ² C 总线接口 2 (I ² C2)	SCL	输入 / 输出	I ² C 时钟的输入 / 输出	是 I ² C 总线的时钟输入 / 输出引脚。
	SDA	输入 / 输出	I ² C 数据的输入 / 输出	是 I ² C 总线的数据输入 / 输出引脚。
A/D 转换器 (ADC)	AN15 ~ AN0	输入	模拟输入引脚	是模拟输入引脚。 SH7136 的引脚为 AN15 ~ AN8、AN3 ~ AN0。
	ADTRG	输入	A/D 转换的触发输入	是 A/D 转换开始的外部触发输入引脚。
	AV _{CC}	输入	模拟电源	是 A/D 转换器的电源引脚。在不使用 A/D 转换器时, 必须连接到系统电源 (V _{CC})。必须将全部 AV _{CC} 引脚连接到系统电源 (V _{CC})。如果有开路引脚就不工作。
	AV _{SS}	输入	模拟接地	是 A/D 转换器的接地引脚, 必须连接到系统电源 (0V)。必须将全部 AV _{SS} 引脚连接到系统电源 (0V)。如果有开路引脚就不工作。
	AV _{refh}	输入	模拟基准电源 (高电平)	是模拟基准电源 (高电平)。
	AV _{refl}	输入	模拟基准电源 (低电平)	是模拟基准电源 (低电平)。
I/O 端口	PA15 ~ PA0	输入 / 输出	通用端口	是 16 位通用输入 / 输出端口的引脚。
	PB7 ~ PB0	输入 / 输出	通用端口	是 8 位通用输入 / 输出端口的引脚。 SH7136 的引脚为 PB7 ~ PB2。
	PD10 ~ PD0	输入 / 输出	通用端口	是 11 位通用输入 / 输出端口的引脚。 SH7136 无此引脚。
	PE21 ~ PE0	输入 / 输出	通用端口	是 22 位通用输入 / 输出端口的引脚。
	PF15 ~ PF0	输入	通用端口	是 16 位通用输入端口的引脚。 SH7136 的引脚为 PF15 ~ PF8、PF3 ~ PF0。
用户断点控制 器 (UBC)	UBCTRG	输出	用户断点的触发输出	是 UBC 条件一致的触发输出引脚。
用户调试接口 (H-UDI)	TCK	输入	测试时钟	是测试时钟的输入引脚。
	TMS	输入	测试模式的选择	是测试模式选择信号的输入引脚。
	TDI	输入	测试数据的输入	是指令和数据的串行输入引脚。
	TDO	输出	测试数据的输出	是指令和数据的串行输出引脚。
	TRST	输入	测试的复位	是初始化信号的输入引脚。

分类	引脚名	输入 / 输出	名称	功能
E10A 接口	ASEMD0	输入	ASE 模式	设定 ASE 模式。 当给此引脚输入低电平时，为 ASE 模式；当输入高电平时，为正常模式。在 ASE 模式中，能使用仿真器专用功能。如果没有给引脚输入任何信号，就在内部进行上拉。
	ASEBRK	输入	断点请求	是 E10A 仿真器的断点输入引脚。
	ASEBRKAK	输出	断点模式的应答	表示 E10A 仿真器已进入断点模式。

【使用时的注意事项】

不要下拉 WDTOVF 引脚。如果需要下拉，就必须使用至少 1MΩ 的电阻。

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 (R_n)
 - 寄存器间接 ($@R_n$)
 - 后增寄存器间接 ($@R_n+$)
 - 先减寄存器间接 ($@-R_n$)
 - 带位移量的寄存器间接 ($@disp:4,R_n$)
 - 带变址的寄存器间接 ($@R0,R_n$)
 - 带位移量的 GBR 间接 ($@disp:8,GBR$)
 - 带变址的 GBR 间接 ($@R0,GBR$)
 - 带位移量的 PC 相对 ($@disp:8,PC$)
 - PC 相对 ($disp:8/disp:12/R_n$)
 - 立即数 ($\#imm:8$)

2.2 寄存器结构

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

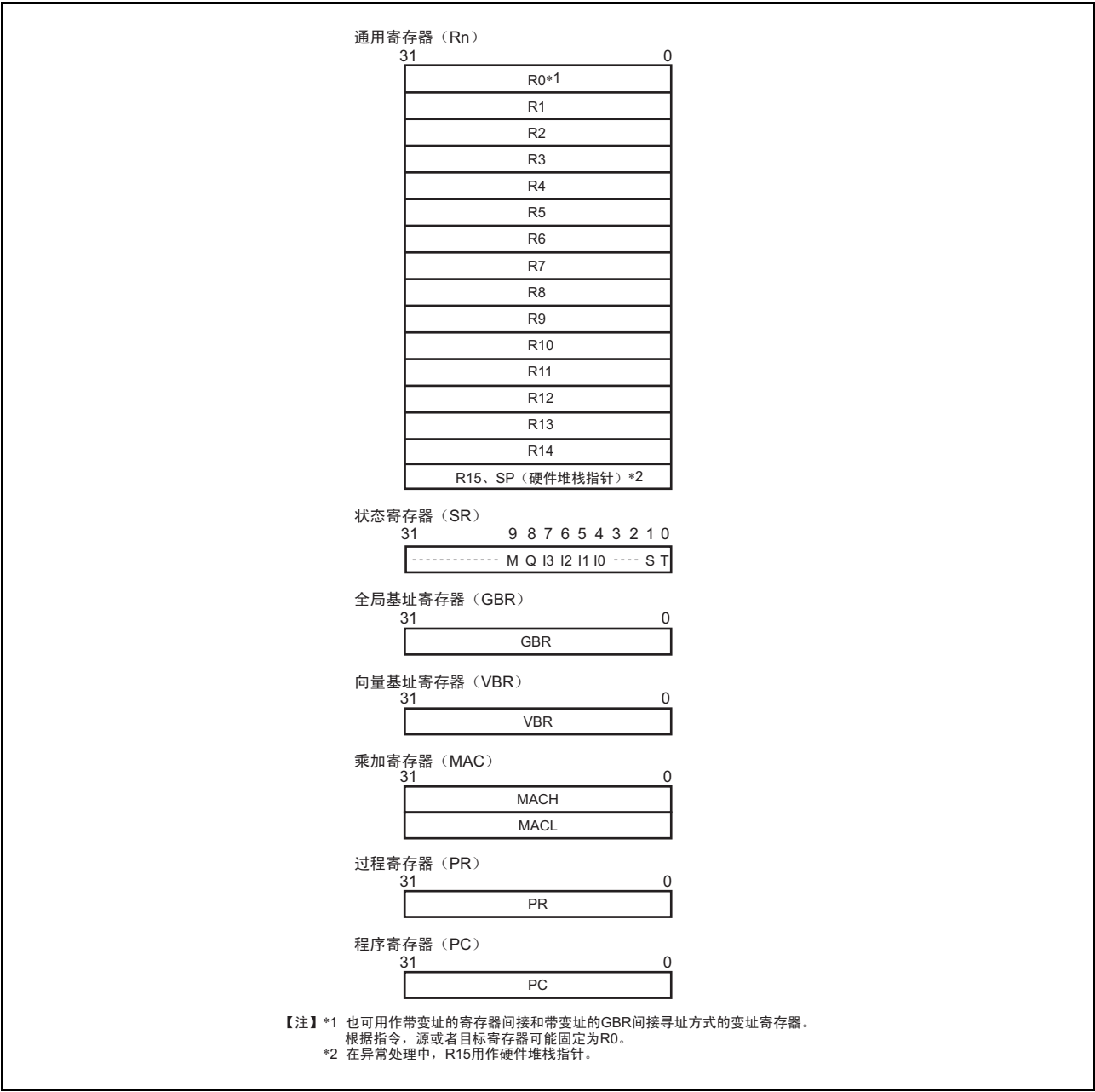


图 2.1 CPU 内部寄存器的结构

2.2.1 通用寄存器（Rn）

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

2.2.2 控制寄存器

控制寄存器为 32 位长，有状态寄存器（SR）、全局基址寄存器（GBR）和向量基址寄存器（VBR）共 3 个。SR 表示处理的状态。GBR 用作 GBR 间接寻址方式的基址，用于内部外围模块的寄存器数据传送等。VBR 用作异常处理（包括中断）向量区的基址。

(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	初始值	R/W	说明
31 ~ 10	—	全 0	R	保留位 读写值总是 0。
9	M	不定值	R/W	DIV0U、DIV0S 和 DIV1 指令使用此位。
8	Q	不定值	R/W	DIV0U、DIV0S 和 DIV1 指令使用此位。
7 ~ 4	I[3:0]	1111	R/W	中断屏蔽位
3、2	—	全 0	R	保留位 读写值总是 0。
1	S	不定值	R/W	S 位 乘加指令使用此位。
0	T	不定值	R/W	T 位 以下指令表示真（1）或者伪（0）： MOVT、CMP/cond、TAS、TST、BT(BT/S)、BF(BF/S)、SETT、CLRT 以下指令表示进位、借位、上溢和下溢等： 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的寄存器区等的数据传送和逻辑运算。

(3) 向量基址寄存器 (VBR)

表示异常处理向量区的基址。

2.2.3 系统寄存器

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

(1) 乘加寄存器 (MACH 和 MACL)

此寄存器保存乘法运算和乘加运算的结果。

(2) 过程寄存器 (PR)

此寄存器保存子程序过程的返回地址。

(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

复位后的寄存器值如表 2.1 所示。

表 2.1 寄存器的初始值

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

2.3 数据格式

2.3.1 寄存器的数据格式

寄存器的操作数长度总是长字 (32 位)。在将存储器的数据加载到寄存器时，如果存储器的操作数长度为字节 (8 位) 或者字 (16 位)，就将数据扩展 (符号扩展) 为长字，保存到寄存器。

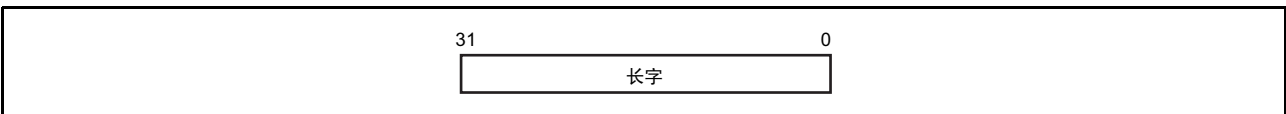


图 2.2 寄存器的数据格式

2.3.2 存储器的数据格式

数据格式有字节、字和长字。

字节数据可分配到任意地址，字数据和长字数据必须分别从地址 $2n$ 和地址 $4n$ 开始分配。如果不按此规定进行存取，就会发生地址错误，此时不保证存取的结果。尤其对于硬件堆栈指针（SP 和 R15）指向的堆栈，因为是用长字保存程序计数器（PC）和状态寄存器（SR），所以必须将硬件堆栈指针的值设定为 $4n$ 。

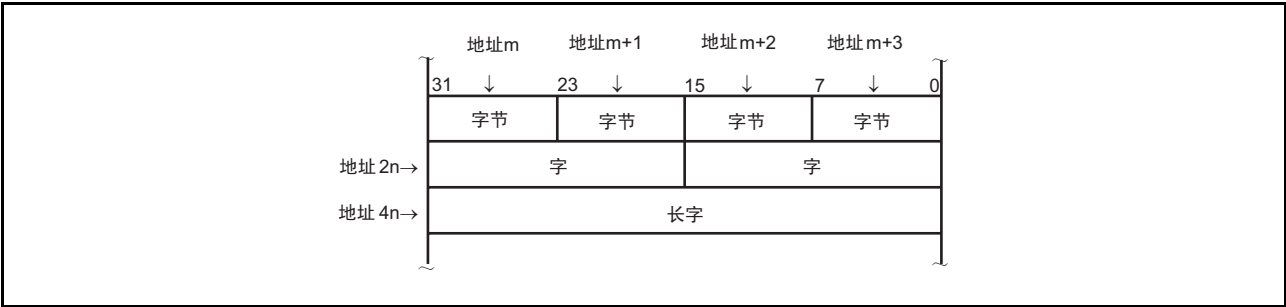


图 2.3 存储器的数据格式

2.3.3 立即数的数据格式

字节（8 位）的立即数分配在指令码中。

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

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

2.4 指令特点

2.4.1 RISC 方式

指令为 RISC 方式，特点如下：

(1) 16 位固定长指令

所有指令长度全部为 16 位固定长，因此能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，存储器的存取长度能选择字节 / 字 / 长字。对于存储器的字节数据和字数据，在进行符号扩展后用长字进行运算；对于立即数，在算术运算时进行符号扩展而在逻辑运算时进行零扩展后，用长字进行运算。

表 2.2 字数据的符号扩展

本 LSI 的 CPU	说明	其他 CPU 的例子
MOV.W @(disp, PC), R1 ADD R1, R0DATA.W H'1234	将数据扩展（符号扩展）为 32 位，R1 为 H'00001234。 然后用 ADD 指令进行运算。	ADD.W #H'1234, R0

【注】 用 @(disp, PC) 参照立即数。

(4) 加载 / 存储体系结构

在寄存器之间进行基本运算。将数据加载到寄存器后，进行和存储器之间的运算（加载 / 存储体系结构）。但是，AND 等位操作指令直接对存储器进行运算。

(5) 延迟转移

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

表 2.3 延迟转移指令

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

(6) 乘法 / 乘加运算

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

(7) T 位

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

表 2.4 T 位

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

(8) 立即数

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

表 2.5 立即数的参照

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), R0DATA.W H'1234	MOV.W #H'1234, R0
32 位立即数	MOV.L @(disp, PC), R0DATA.L H'12345678	MOV.L #H'12345678, R0

【注】 用 @(disp, PC) 参照立即数。

(9) 绝对地址

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

表 2.6 绝对地址的参照

分类	本 LSI 的 CPU	其他 CPU 的例子
绝对地址	MOV.L @(disp, PC), R1 MOV.B @ R1, R0DATA.L H'12345678	MOV.B @ H'12345678, R0

【注】 用 @(disp, PC) 参照立即数。

(10) 16 位 /32 位的位移量

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

表 2.7 位移量的参照

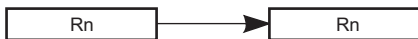
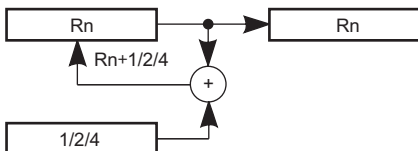
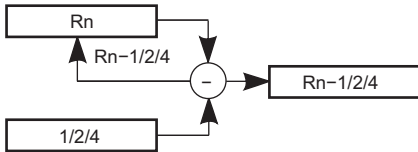
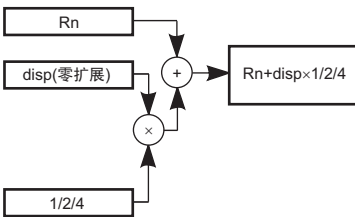
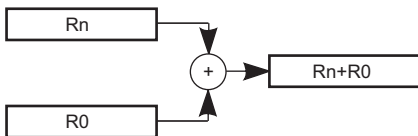
分类	本 LSI 的 CPU	其他 CPU 的例子
16 位的位移量	MOV.W @(disp, PC), R0 MOV.W @(R0, R1), R2DATA.W H'1234	MOV.W @(H'1234, R1), R2

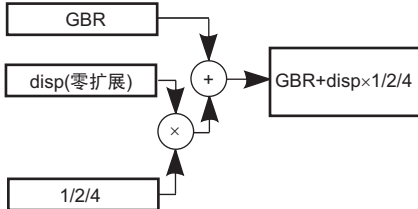
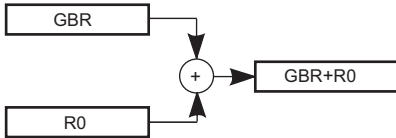
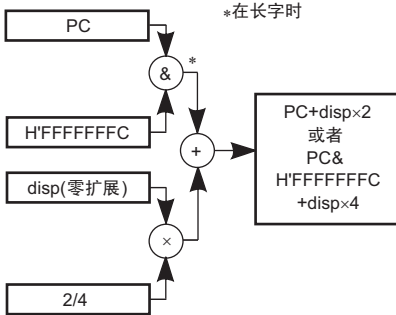
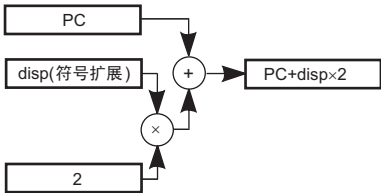
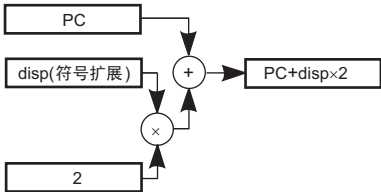
【注】 用 @(disp, PC) 参照立即数。

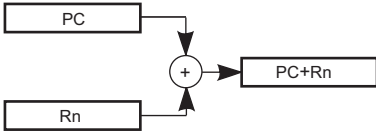
2.4.2 寻址方式

寻址方式和有效地址的计算方法如表 2.8 所示。

表 2.8 寻址方式和有效地址

寻址方式	指令格式	有效地址的计算方法	计算式
寄存器直接	Rn	有效地址为寄存器 Rn。 (操作数为寄存器 Rn 的内容)	—
寄存器间接	@Rn	有效地址为寄存器 Rn 的内容。 	Rn
后增寄存器间接	@Rn+	有效地址为寄存器 Rn 的内容。在执行指令后给 Rn 加上常数。常数在操作数长度为字节时是 1, 为字时是 2, 为长字时是 4。 	执行 Rn 指令后 字节: $Rn+1 \rightarrow Rn$ 字: $Rn+2 \rightarrow Rn$ 长字: $Rn+4 \rightarrow Rn$
先减寄存器间接	@-Rn	有效地址为寄存器 Rn 减去常数后的内容。常数在操作数长度为字节时是 1, 为字时是 2, 为长字时是 4。 	字节: $Rn-1 \rightarrow Rn$ 字: $Rn-2 \rightarrow Rn$ 长字: $Rn-4 \rightarrow Rn$ (用计算后的 Rn 执行指令)
带位移量的寄存器间接	@(disp:4, Rn)	有效地址为寄存器 Rn 加上 4 位位移量 disp 后的内容。disp 在进行零扩展后, 操作数长度为字节时乘 1, 为字时乘 2, 为长字时乘 4。 	字节: $Rn+disp$ 字: $Rn+disp \times 2$ 长字: $Rn+disp \times 4$
带变址的寄存器间接	@(R0, Rn)	有效地址为寄存器 Rn 加上 R0 后的内容。 	$Rn+R0$

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 GBR 间接	@(disp: 8, GBR)	有效地址为寄存器 GBR 加上 8 位位移量 disp 后的内容。disp 在进行零扩展后，操作数长度为字节时乘 1，为字时乘 2，为长字时乘 4。 	字节: $GBR+disp$ 字: $GBR+disp \times 2$ 长字: $GBR+disp \times 4$
带变址的 GBR 间接	@(R0, GBR)	有效地址为寄存器 GBR 加上 R0 后的内容。 	$GBR+R0$
带位移量的 PC 相对	@(disp: 8, PC)	有效地址为寄存器 PC 加上 8 位位移量 disp 后的内容。disp 在进行零扩展后，操作数长度为字时乘 2，为长字时乘 4。另外，操作数长度为长字时，屏蔽 PC 的低 2 位。 	字: $PC+disp \times 2$ 长字: $PC \& H'FFFFFFC + disp \times 4$
PC 相对	disp: 8	有效地址为寄存器 PC 加上 8 位位移量 disp（符号扩展后的 disp）乘 2 后的内容。 	$PC+disp \times 2$
	disp: 12	有效地址为寄存器 PC 加上 12 位位移量 disp（符号扩展后的 disp）乘 2 后的内容。 	$PC+disp \times 2$

寻址方式	指令格式	有效地址的计算方法	计算式
PC 相对	Rn	有效地址为寄存器 PC 加上 Rn 后的内容。 	PC+Rn
立即数	#imm:8	TST、AND、OR, 和 XOR 指令的 8 位立即数 imm 进行零扩展。	—
	#imm:8	MOV、ADD 和 CMP/EQ 指令的 8 位立即数 imm 进行符号扩展。	—
	#imm:8	TRAPA 指令的 8 位立即数 imm 进行零扩展后乘 4。	—

2.4.3 指令格式

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

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

表 2.9 指令格式

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

指令格式	源操作数	目标操作数	指令例子
d 格式	15 xxxx xxxx dddd dddd 0	dddddddd: 带位移量的 GBR 间接	R0 (寄存器直接) MOV.L @(disp, GBR), R0
		R0 (寄存器直接)	dddddddd: 带位移量的 GBR 间接 MOV.L R0, @(disp, GBR)
		dddddddd: 带位移量的 PC 相对	MOVA @(disp, PC), R0
		—	dddddddd: PC 相对 BF label
d12 格式	15 xxxx dddd dddd dddd 0	—	dddddddddddd: PC 相对 BRA label (label=disp+pc)
nd8 格式	15 xxxx nnnn dddd dddd 0	dddddddd: 带位移量的 PC 相对	nnnn: 寄存器直接 MOV.L @(disp, PC), Rn
i 格式	15 xxxx xxxx iiii iiii 0	iiiiiii: 立即数	带变址的 GBR 间接 AND.B #imm, @(R0, GBR)
		iiiiiii: 立即数	R0 (寄存器直接) AND #imm, R0
		iiiiiii: 立即数	— TRAPA #imm
ni 格式	15 xxxx nnnn iiii iiii 0	iiiiiii: 立即数	nnnn: 寄存器直接 ADD #imm, Rn

【注】* 在乘加指令中，nnnn 为源寄存器。

2.5 指令系统

2.5.1 指令系统的分类

指令的分类如表 2.10 所示。

表 2.10 指令的分类

分类	指令的种类	操作码	功能	指令数
数据传送指令	5	MOV	传送数据 传送立即数 传送外围模块数据 传送结构体数据	39
		MOVA	传送有效地址	
		MOVT	传送 T 位	
		SWAP	交换高位和低位	
		XTRCT	抽出连接寄存器的中间部分	
算术运算指令	21	ADD	2 进制加法	33
		ADDC	带进位的 2 进制加法	
		ADDV	带上溢的 2 进制加法	
		CMP/cond	比较	
		DIV1	除法	
		DIV0S	带符号的除法初始化	
		DIV0U	无符号的除法初始化	
		DMULS	带符号的双精度乘法	
		DMULU	无符号的双精度乘法	
		DT	递减和测试	
		EXTS	符号扩展	
		EXTU	零扩展	
		MAC	乘加运算、双精度乘加运算	
		MUL	双精度乘法	
		MULS	带符号乘法	
		MULU	无符号乘法	
		NEG	符号取反	
		NEGC	带借位的符号取反	
		SUB	2 进制减法	
		SUBC	带借位的 2 进制减法	
		SUBV	带下溢的 2 进制减法	
逻辑运算指令	6	AND	逻辑与运算	14
		NOT	位取反	
		OR	逻辑或运算	
		TAS	存储器测试和置位	
		TST	逻辑与运算的 T 位置位	
		XOR	逻辑异或运算	

分类	指令的种类	操作码	功能	指令数
移位指令	10	ROTL	左循环 1 位	14
		ROTR	右循环 1 位	
		ROTCL	带 T 位的左循环 1 位	
		ROTCR	带 T 位的右循环 1 位	
		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	转移到子程序过程	
		BSRF	转移到子程序过程	
		JMP	无条件转移	
		JSR	转移到子程序过程	
		RTS	从子程序过程返回	
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	无操作	
		RTE	从异常处理返回	
		SETT	T 位置位	
		SLEEP	转移到低功耗状态	
		STC	保存控制寄存器的内容	
		STS	保存系统寄存器的内容	
		TRAPA	陷阱异常处理	
	合计 62			142

用以下形式，按照分类顺序说明指令的指令码、操作和执行状态。

指令	指令码	操作概要	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 iiii: 立即数 dddd: 位移量	表示操作的概要。 符号说明 →、←: 传送方向 (xx): 存储器操作数 M/Q/T: SR 内的标志位 &: 每位的逻辑与 : 每位的逻辑或 ^: 每位的逻辑异或 ~: 每位的逻辑非 <<n: 左移 n 位 >>n: 右移 n 位	是无等待时的值。*1	表示执行指令后 T 位的值。 符号说明 —: 不变

【注】 *1 有关指令的执行状态

表中所示的执行状态为最小值。实际上，在以下条件下，指令执行状态数会增加：

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

*2 根据指令的操作数长度等进行倍增（×1、×2、×4）。

详细内容请参照《SH-1/SH-2/SH-DSP 软件手册》。

2.5.2 数据传送指令

表 2.11 数据传送指令

指令	指令码	操作	执行状态	T 位
MOV #imm, Rn	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	—

指令	指令码	操作	执行状态	T 位
MOV.B R0, @(disp, Rn)	10000000nnnnndddd	R0→(disp+Rn)	1	—
MOV.W R0, @(disp, Rn)	10000001nnnnndddd	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)	11000000ddddd	R0→(disp+GBR)	1	—
MOV.W R0, @(disp, GBR)	11000001ddddd	R0→(disp×2+GBR)	1	—
MOV.L R0, @(disp, GBR)	11000010ddddd	R0→(disp×4+GBR)	1	—
MOV.B @(disp, GBR), R0	11000100ddddd	(disp+GBR)→ 符号扩展 →R0	1	—
MOV.W @(disp, GBR), R0	11000101ddddd	(disp×2+GBR)→ 符号扩展 →R0	1	—
MOV.L @(disp, GBR), R0	11000110ddddd	(disp×4+GBR)→R0	1	—
MOVA @(disp, PC), R0	11000111ddddd	disp×4+PC→R0	1	—
MOVT Rn	0000nnnn00101001	T→Rn	1	—
SWAP.B Rm, Rn	0110nnnnmmmm1000	Rm→ 交换低位 2 字节的高低字节 →Rn	1	—
SWAP.W Rm, Rn	0110nnnnmmmm1001	Rm→ 交换高低字 →Rn	1	—
XTRCT Rm, Rn	0010nnnnmmmm1101	Rm: Rn 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	指令码	操作	执行状态	T 位
ADD Rm, Rn	0011nnnnmmmm1100	Rn+Rm→Rn	1	—
ADD #imm, Rn	0111nnnniiiiiii	Rn+imm→Rn	1	—
ADDC Rm, Rn	0011nnnnmmmm1110	Rn+Rm+T→Rn, 进位 →T	1	进位
ADDV Rm, Rn	0011nnnnmmmm1111	Rn+Rm→Rn, 上溢 →T	1	上溢
CMP/EQ #imm, R0	10001000iiiiiii	R0=imm, 1→T	1	比较结果
CMP/EQ Rm, Rn	0011nnnnmmmm0000	Rn=Rm, 1→T	1	比较结果
CMP/HS Rm, Rn	0011nnnnmmmm0010	当无符号 Rn ≥ Rm 时, 1→T	1	比较结果
CMP/GE Rm, Rn	0011nnnnmmmm0011	当带符号 Rn ≥ Rm 时, 1→T	1	比较结果
CMP/HI Rm, Rn	0011nnnnmmmm0110	当无符号 Rn > Rm 时, 1→T	1	比较结果
CMP/GT Rm, Rn	0011nnnnmmmm0111	当带符号 Rn > Rm 时, 1→T	1	比较结果
CMP/PL Rn	0100nnnn00010101	当 Rn > 0 时, 1→T	1	比较结果
CMP/PZ Rn	0100nnnn00010001	当 Rn ≥ 0 时, 1→T	1	比较结果
CMP/STR Rm, Rn	0010nnnnmmmm1100	当任意字节相等时, 1→T	1	比较结果

指令	指令码	操作	执行状态	T 位
DIV1 Rm, Rn	0011nnnnmmmm0100	单步除法 ($R_n \div R_m$)	1	计算结果
DIV0S Rm, Rn	0010nnnnmmmm0111	R_n 的 MSB $\rightarrow Q$, R_m 的 MSB $\rightarrow M$, $M^*Q \rightarrow T$	1	计算结果
DIV0U	0000000000011001	$0 \rightarrow M/Q/T$	1	0
DMULS.L Rm, Rn	0011nnnnmmmm1101	带符号 $R_n \times R_m \rightarrow MACH$ 和 $MACL$ $32 \times 32 \rightarrow 64$ 位	2 ~ 5*	—
DMULU.L Rm, Rn	0011nnnnmmmm0101	无符号 $R_n \times R_m \rightarrow MACH$ 和 $MACL$ $32 \times 32 \rightarrow 64$ 位	2 ~ 5*	—
DT Rn	0100nnnn00010000	$R_n - 1 \rightarrow R_n$ 当 R_n 为 0 时, $1 \rightarrow T$ 当 R_n 不为 0 时, $0 \rightarrow T$	1	比较结果
EXTS.B Rm, Rn	0110nnnnmmmm1110	将 R_m 的字节进行符号扩展 $\rightarrow R_n$	1	—
EXTS.W Rm, Rn	0110nnnnmmmm1111	将 R_m 的字进行符号扩展 $\rightarrow R_n$	1	—
EXTU.B Rm, Rn	0110nnnnmmmm1100	将 R_m 的字节进行零扩展 $\rightarrow R_n$	1	—
EXTU.W Rm, Rn	0110nnnnmmmm1101	将 R_m 的字进行零扩展 $\rightarrow R_n$	1	—
MAC.L @Rm+, @Rn+	0000nnnnmmmm1111	带符号 $(R_n) \times (R_m) + MAC \rightarrow MAC$ $32 \times 32 + 64 \rightarrow 64$ 位	2 ~ 5*	—
MAC.W @Rm+, @Rn+	0100nnnnmmmm1111	带符号 $(R_n) \times (R_m) + MAC \rightarrow MAC$ $16 \times 16 + 64 \rightarrow 64$ 位	2 ~ 4*	—
MUL.L Rm, Rn	0000nnnnmmmm0111	$R_n \times R_m \rightarrow MACL$ $32 \times 32 \rightarrow 32$ 位	2 ~ 5*	—
MULS.W Rm, Rn	0010nnnnmmmm1111	带符号 $R_n \times R_m \rightarrow MACL$ $16 \times 16 \rightarrow 32$ 位	1 ~ 3*	—
MULU.W Rm, Rn	0010nnnnmmmm1110	无符号 $R_n \times R_m \rightarrow MACL$ $16 \times 16 \rightarrow 32$ 位	1 ~ 3*	—
NEG Rm, Rn	0110nnnnmmmm1011	$0 - R_m \rightarrow R_n$	1	—
NEGC Rm, Rn	0110nnnnmmmm1010	$0 - R_m - T \rightarrow R_n$, 借位 $\rightarrow T$	1	借位
SUB Rm, Rn	0011nnnnmmmm1000	$R_n - R_m \rightarrow R_n$	1	—
SUBC Rm, Rn	0011nnnnmmmm1010	$R_n - R_m - T \rightarrow R_n$, 借位 $\rightarrow T$	1	借位
SUBV Rm, Rn	0011nnnnmmmm1011	$R_n - R_m \rightarrow R_n$, 下溢 $\rightarrow T$	1	上溢

【注】 * 表示通常的执行状态。

2.5.4 逻辑运算指令

表 2.13 逻辑运算指令

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

2.5.5 移位指令

表 2.14 移位指令

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

2.5.6 转移指令

表 2.15 转移指令

指令	指令码	操作	执行状态	T 位
BF label	10001011dddddddd	当 T=0 时, $\text{disp} \times 2 + \text{PC} \rightarrow \text{PC}$ 当 T=1 时, nop	3/1*	—
BF/S label	10001111dddddddd	延迟转移 当 T=0 时, $\text{disp} \times 2 + \text{PC} \rightarrow \text{PC}$ 当 T=1 时, nop	2/1*	—
BT label	10001001dddddddd	当 T=1 时, $\text{disp} \times 2 + \text{PC} \rightarrow \text{PC}$ 当 T=0 时, nop	3/1*	—
BT/S label	10001101dddddddd	延迟转移 当 T=1 时, $\text{disp} \times 2 + \text{PC} \rightarrow \text{PC}$ 当 T=0 时, nop	2/1*	—
BRA label	1010dddddddddddd	延迟转移 $\text{disp} \times 2 + \text{PC} \rightarrow \text{PC}$	2	—
BRAF Rm	0000mmmm00100011	延迟转移 $\text{Rm} + \text{PC} \rightarrow \text{PC}$	2	—
BSR label	1011dddddddddddd	延迟转移 $\text{PC} \rightarrow \text{PR}$ $\text{disp} \times 2 + \text{PC} \rightarrow \text{PC}$	2	—
BSRF Rm	0000mmmm00000011	延迟转移 $\text{PC} \rightarrow \text{PR}$, $\text{Rm} + \text{PC} \rightarrow \text{PC}$	2	—
JMP @Rm	0100mmmm00101011	延迟转移 $\text{Rm} \rightarrow \text{PC}$	2	—
JSR @Rm	0100mmmm00001011	延迟转移 $\text{PC} \rightarrow \text{PR}$, $\text{Rm} \rightarrow \text{PC}$	2	—
RTS	0000000000001011	延迟转移 $\text{PR} \rightarrow \text{PC}$	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	指令码	操作	执行状态	T 位
CLRT	0000000000001000	0→T	1	0
CLRMACH	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	无操作	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】 * 表示转移到睡眠状态前的状态数。

有关指令的执行状态

表中所示的执行状态为最小值。实际上，在以下条件下，指令执行状态数会增加：

- (1) 取指令和数据存取发生竞争
- (2) 加载指令（存储器→寄存器）的目标寄存器和紧接者的指令所使用的寄存器相同

2.6 处理状态

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

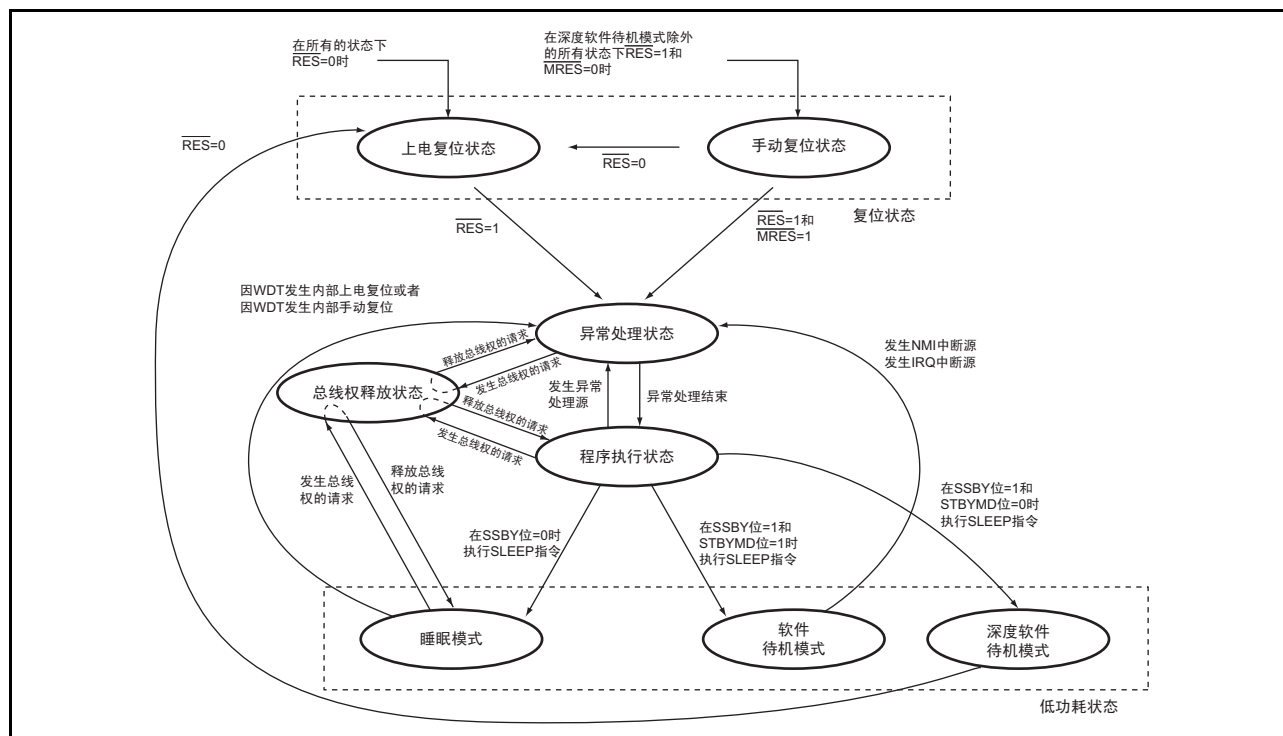


图 2.4 处理状态的转移图

(1) 复位状态

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

(2) 异常处理状态

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

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

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

以后的处理状态为程序执行状态。

(3) 程序执行状态

这是 CPU 依次执行程序的状态。

(4) 低功耗状态

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

(5) 总线权释放状态

这是 CPU 对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 4 种 MCU 运行模式和 3 种内部闪存的编程模式。

通过 FWE 引脚、MD1 引脚和 MD0 引脚的组合进行运行模式的设定。

本 LSI 能设定的运行模式组合如表 3.1 所示，不能设定此表中没有的组合。

在接通系统电源时，必须进行上电复位处理。

MCU 运行模式有 MCU 扩展模式 0、MCU 扩展模式 2 和单芯片模式。

内部闪存的编程模式有引导模式（板上编程模式）、用户引导模式和用户编程模式。

表 3.1 运行模式的选择

MCU 运行模式	引脚的设定			模式	内部 ROM	CS0 空间的总线宽度	
	FWE	MD1	MD0*1			SH7136	SH7137
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	—	8
模式 3	0	1	1	单芯片模式	有效	—	
模式 4*2	1	0	0	引导模式	有效	—	
模式 5*2	1	0	1	用户引导模式	有效	—	8
模式 6*2	1	1	0	用户编程模式	有效	—	8
模式 7*2	1	1	1		有效	—	

【注】 *1 SH7136 无 MD0 引脚。FWE 引脚和 MD1 引脚的组合只支持以下运行模式：

单芯片模式 : FWE 引脚=0、MD1 引脚=1

引导模式 : FWE 引脚=1、MD1 引脚=0

用户编程模式 : FWE 引脚=1、MD1 引脚=1

*2 闪存编程模式。

3.2 输入 / 输出引脚

有关运行模式的引脚结构如表 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	是用于内部闪存编程 / 擦除的硬件允许引脚。

3.3 各运行模式的说明

3.3.1 模式 0（MCU 扩展模式 0）

在模式 0 中，CS0 空间的总线宽度为外部存储器空间的 8 位总线（SH7137）。

3.3.2 模式 2（MCU 扩展模式 2）

在模式 2 中，内部 ROM 有效，能使用 CS 空间。

3.3.3 模式 3（单芯片模式）

在单芯片模式中，能使用全部端口，但是不能使用外部地址。

3.4 地址映像

各运行模式的地址映像如图 3.1 和图 3.2 所示。

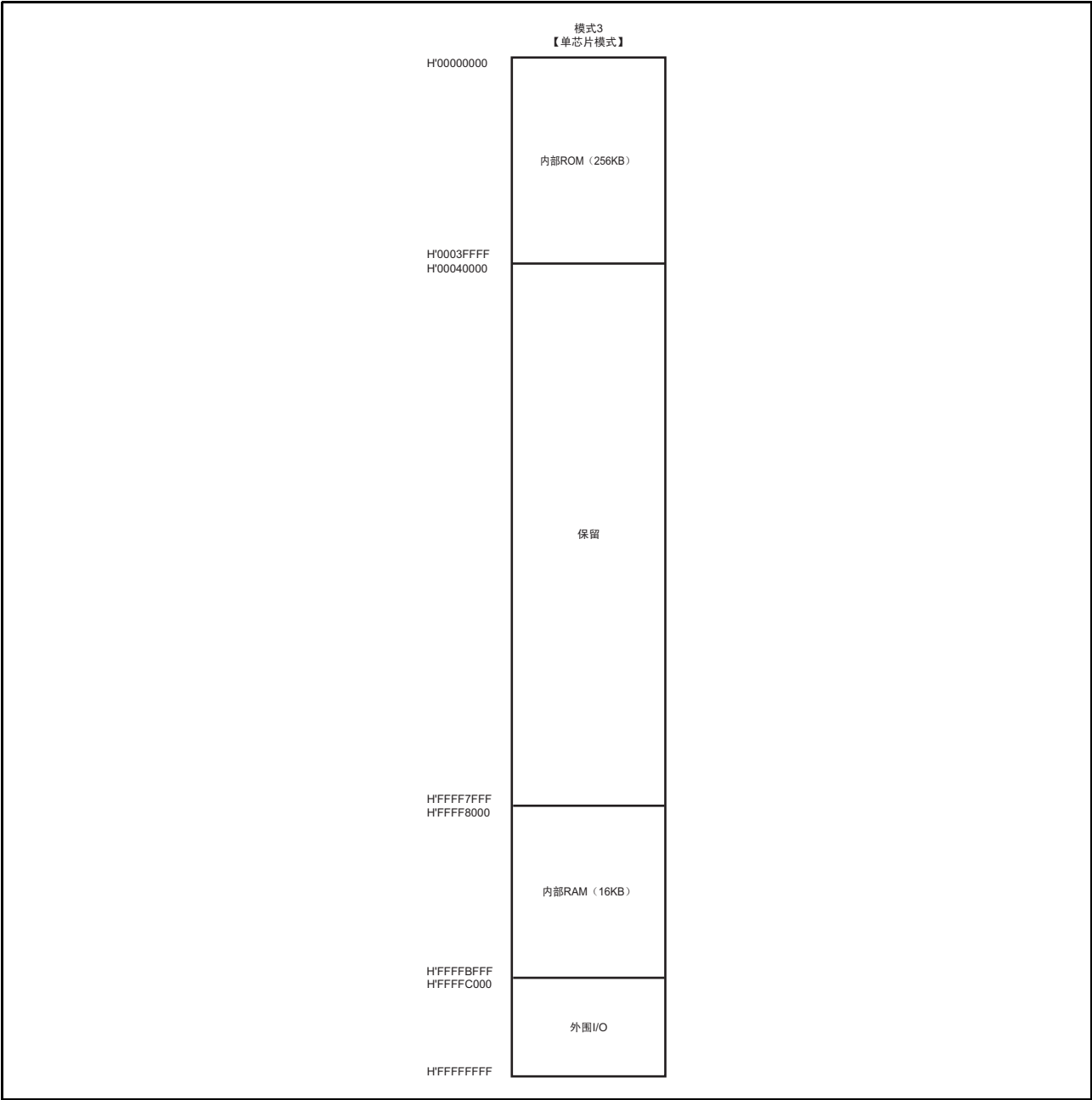


图 3.1 SH7136 各运行模式的地址映像

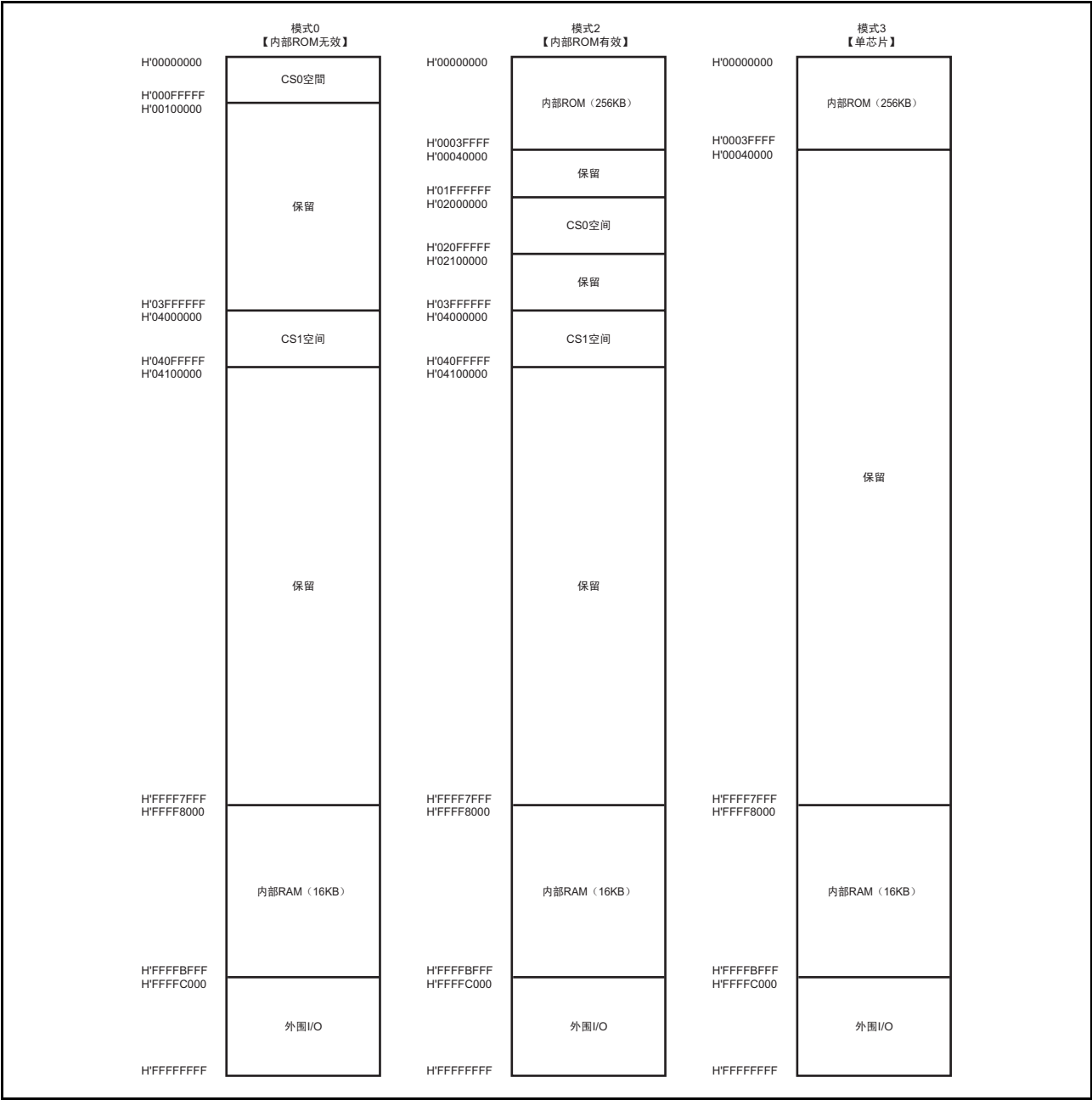


图 3.2 SH7137 各运行模式的地址映像

3.5 本 LSI 的初始状态

为了降低功耗，本 LSI 在初始状态下将一部分内部模块设定为模块待机状态。因此，在要使这些模块运行时，必须解除模块待机状态。详细内容请参照“第 24 章 低功耗模式”。

3.6 更改运行模式时的注意事项

如果要在给 LSI 外加电源时更改运行模式，就必须在上电复位的状态（给 $\overline{\text{RES}}$ 引脚外加低电平）下进行。

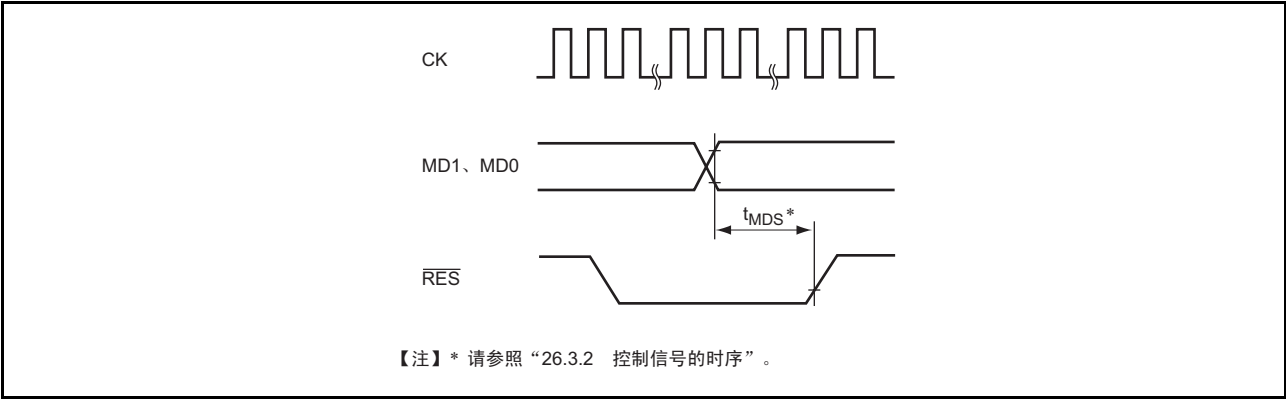


图 3.3 更改运行模式时的复位输入时序

第 4 章 时钟振荡器 (CPG)

本 LSI 内置时钟振荡器 (CPG)。

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

4.1 特点

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

时钟振荡器的框图如图 4.1 所示。

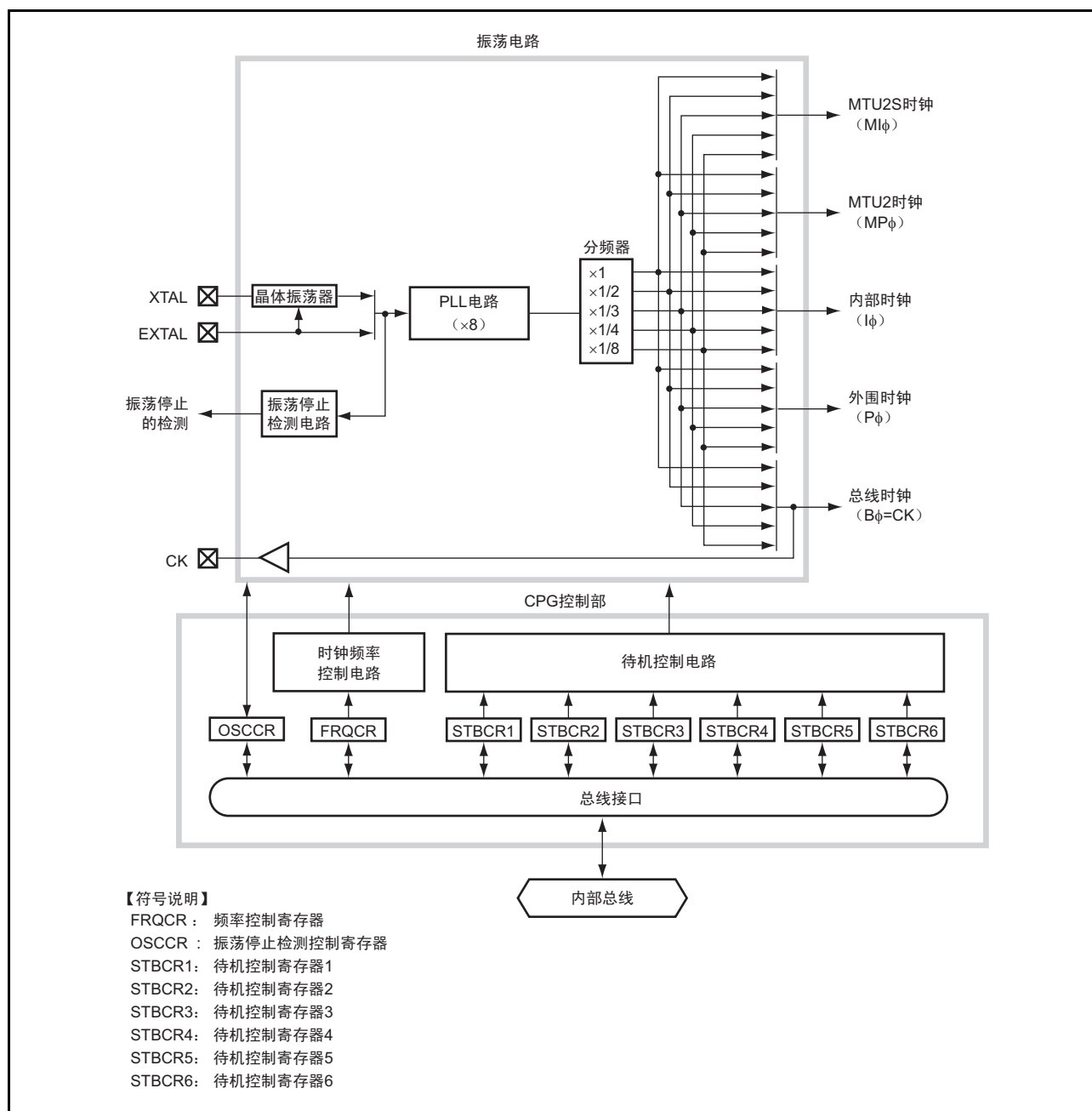


图 4.1 时钟振荡器的框图

时钟振荡器各模块的功能如下:

(1) PLL 电路

PLL 电路具有将晶体谐振器或者 EXTAL 引脚的输入时钟频率递增 8 倍的功能。倍增率总是固定为 8 倍。

(2) 晶体振荡器

这是将晶体谐振器连接到 XTAL 引脚和 EXTAL 引脚时的振荡电路。

(3) 分频器

分频器能生成内部时钟 ($I\phi$)、总线时钟 ($B\phi$)、外围时钟 ($P\phi$)、MTU2S 时钟 ($MI\phi$) 和 MTU2 时钟 ($MP\phi$) 使用的工作频率时钟。

工作频率可选择 PLL 电路输出频率的 1 倍、1/2 倍、1/3 倍、1/4 倍或者 1/8 倍。

通过频率控制寄存器 (FRQCR) 设定分频比。

(4) 振荡停止的检测电路

这是检测晶体谐振器异常的电路。

(5) 时钟频率的控制电路

时钟频率控制电路通过设定频率控制寄存器 (FRQCR) 进行时钟频率的控制。

(6) 待机控制电路

待机控制电路对睡眠 / 待机模式中的内部振荡电路和其他模块的状态进行控制。

(7) 频率控制寄存器 (FRQCR)

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

(8) 振荡停止检测控制寄存器 (OSCCR)

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

(9) 待机控制寄存器 1 ~ 6 (STBCR1 ~ STBCR6)

在待机控制寄存器 (STBCR) 中分配了低功耗模式的各控制位。有关待机控制寄存器请参照“第 24 章 低功耗模式”。

各模块的运行时钟如表 4.1 所示。

表 4.1 各模块的运行时钟

运行时钟	对应模块	运行时钟	对应模块
内部时钟 (I ϕ)	CPU UBC ROM RAM	外围时钟 (P ϕ)	POE SCI SSU RCAN-ET I ² C A/D CMT WDT
总线时钟 (B ϕ)	BSC DTC	MTU2 时钟 (MP ϕ)	MTU2
		MTU2S 时钟 (MI ϕ)	MTU2S

4.2 输入 / 输出引脚

CPG 的引脚结构和功能如表 4.2 所示。

表 4.2 时钟振荡器的引脚结构和功能

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

【注】 在使用时钟输出引脚 (CK) 时, 需要通过引脚功能控制器 (PFC) 设定引脚。详细内容请参照 “第 20 章 引脚功能控制器 (PFC)”。

4.3 时钟工作模式

本 LSI 的时钟工作模式如表 4.3 所示。

表 4.3 时钟工作模式

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

【注】 * 在从 CK 引脚输出时钟时, 需要设定 PFC。有关 PFC 设定的详细内容请参照 “第 20 章 引脚功能控制器 (PFC)”。

• 模式 1

因为从 EXTAL 引脚输入外部时钟, 并通过 PLL 电路将其频率递增 8 倍后提供给本 LSI 内部, 所以能降低外部生成的时钟频率。能使用的输入时钟频率为 5MHz ~ 12.5MHz, 内部时钟 (I ϕ) 的频率范围为 10MHz ~ 80MHz。

最大工作频率: I ϕ =80MHz、B ϕ =40MHz、P ϕ =40MHz、MI ϕ =80MHz、MP ϕ =40MHz、

能通过 FRQCR 设定的分频比如表 4.4 所示。

表 4.4 FRQCR 设定的分频比

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

PLL 倍增率	FRQCR 的分频比设定					时钟比					时钟频率 (MHz) *					
	l ϕ	B ϕ	P ϕ	MI ϕ	MP ϕ	l ϕ	B ϕ	P ϕ	MI ϕ	MP ϕ	输入时钟	l ϕ	B ϕ	P ϕ	MI ϕ	MP ϕ
×8	1/1	1/4	1/8	1/1	1/8	8	2	1	8	1	10	80	20	10	80	10
	1/1	1/4	1/8	1/1	1/4	8	2	1	8	2		80	20	10	80	20
	1/1	1/4	1/4	1/4	1/4	8	2	2	2	2		80	20	20	20	20
	1/1	1/4	1/4	1/2	1/4	8	2	2	4	2		80	20	20	40	20
	1/1	1/4	1/4	1/1	1/4	8	2	2	8	2		80	20	20	80	20
	1/1	1/3	1/3	1/3	1/3	8	8/3	8/3	8/3	8/3		80	26	26	26	26
	1/1	1/3	1/3	1/1	1/3	8	8/3	8/3	8	8/3		80	26	26	80	26
	1/1	1/2	1/8	1/8	1/8	8	4	1	1	1		80	40	10	10	10
	1/1	1/2	1/8	1/4	1/8	8	4	1	2	1		80	40	10	20	10
	1/1	1/2	1/8	1/4	1/4	8	4	1	2	2		80	40	10	20	20
	1/1	1/2	1/8	1/2	1/8	8	4	1	4	1		80	40	10	40	10
	1/1	1/2	1/8	1/2	1/4	8	4	1	4	2		80	40	10	40	20
	1/1	1/2	1/8	1/2	1/2	8	4	1	4	4		80	40	10	40	40
	1/1	1/2	1/8	1/1	1/8	8	4	1	8	1		80	40	10	80	10
	1/1	1/2	1/8	1/1	1/4	8	4	1	8	2		80	40	10	80	20
	1/1	1/2	1/8	1/1	1/2	8	4	1	8	4		80	40	10	80	40
	1/1	1/2	1/4	1/4	1/4	8	4	2	2	2		80	40	20	20	20
	1/1	1/2	1/4	1/2	1/4	8	4	2	4	2		80	40	20	40	20
	1/1	1/2	1/4	1/2	1/2	8	4	2	4	4		80	40	20	40	40
	1/1	1/2	1/4	1/1	1/4	8	4	2	8	2		80	40	20	80	20
	1/1	1/2	1/4	1/1	1/2	8	4	2	8	4		80	40	20	80	40
	1/1	1/2	1/2	1/2	1/2	8	4	4	4	4		80	40	40	40	40
	1/1	1/2	1/2	1/1	1/2	8	4	4	8	4		80	40	40	80	40
	1/1	1/1	1/4	1/4	1/4	8	8	2	2	2	5	40	40	10	10	10
	1/1	1/1	1/4	1/2	1/4	8	8	2	4	2		40	40	10	20	10
	1/1	1/1	1/4	1/2	1/2	8	8	2	4	4		40	40	10	20	20
	1/1	1/1	1/4	1/1	1/4	8	8	2	8	2		40	40	10	40	10
	1/1	1/1	1/4	1/1	1/2	8	8	2	8	4		40	40	10	40	20
	1/1	1/1	1/4	1/1	1/1	8	8	2	8	8		40	40	10	40	40
	1/1	1/1	1/3	1/3	1/3	8	8	8/3	8/3	8/3		40	40	13	13	13
	1/1	1/1	1/3	1/1	1/3	8	8	8/3	8	8/3		40	40	13	40	13
	1/1	1/1	1/3	1/1	1/1	8	8	8/3	8	8		40	40	13	40	40
	1/1	1/1	1/2	1/2	1/2	8	8	4	4	4		40	40	20	20	20
	1/1	1/1	1/2	1/1	1/2	8	8	4	8	4		40	40	20	40	20
	1/1	1/1	1/2	1/1	1/1	8	8	4	8	8		40	40	20	40	40
	1/1	1/1	1/1	1/1	1/1	8	8	8	8	8		40	40	40	40	40

【注】* 时钟频率是假定了输入时钟频率时的值。

1. PLL 电路的倍增率只能为 ×8。分频器的分频比能选择 ×1、×1/2、×1/3、×1/4 或者 ×1/8。通过频率控制寄存器，给要设定的各个时钟设定分频比。
2. PLL 电路的输出频率为晶体谐振器或者 EXTAL 引脚的输入时钟频率乘 PLL 电路的倍增率（8 倍）。
3. 分频器的输入总是为 PLL 电路的输出。

4. 内部时钟（ $I\phi$ ）的频率为晶体谐振器或者 EXTAL 引脚的输入时钟频率乘 PLL 电路的倍增率（8 倍）和分频器的分频比。
内部时钟（ $I\phi$ ）的频率不能超过最大工作频率（80MHz）。
5. 总线时钟（ $B\phi$ ）的频率为晶体谐振器或者 EXTAL 引脚的输入时钟频率乘 PLL 电路的倍增率（8 倍）和分频器的分频比。
总线时钟（ $B\phi$ ）的频率不能超过 40MHz 和内部时钟（ $I\phi$ ）的频率。
6. 外围时钟（ $P\phi$ ）的频率为晶体谐振器或者 EXTAL 引脚的输入时钟频率乘 PLL 电路的倍增率（8 倍）和分频器的分频比。
外围时钟（ $P\phi$ ）的频率不能超过 40MHz 和总线时钟（ $B\phi$ ）的频率。
7. 在使用 MTU2S 和 MTU2 时，MTU2S 时钟（ $MI\phi$ ）的频率不能超过内部时钟（ $I\phi$ ）的频率，并且不能低于 MTU2 时钟（ $MP\phi$ ）的频率；MTU2 时钟（ $MP\phi$ ）的频率不能超过 MTU2S 时钟（ $MI\phi$ ）的频率和总线时钟（ $B\phi$ ）的频率，并且不能低于外围时钟（ $P\phi$ ）的频率。
MTU2S 时钟（ $MI\phi$ ）和 MTU2 时钟（ $MP\phi$ ）的频率为晶体谐振器或者 EXTAL 引脚的输入时钟频率乘 PLL 电路的倍增率（8 倍）和分频器的分频比。
8. CK 引脚的频率总是和总线时钟（ $B\phi$ ）的频率相等。

4.4 寄存器说明

CPG 有以下寄存器。有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 25 章 寄存器一览”。

表 4.5 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
频率控制寄存器	FRQCR	R/W	H'36DB	H'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 是 16 位可读写寄存器，能指定内部时钟（ $I\phi$ ）、总线时钟（ $B\phi$ ）、外围时钟（ $P\phi$ ）、MTU2S 时钟（ $MI\phi$ ）和 MTU2 时钟（ $MP\phi$ ）的分频比。FRQCR 只能进行字存取。

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

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

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
14 ~ 12	IFC[2:0]	011	R/W	内部时钟 (Iϕ) 频率的分频比 对于 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, 就不保证以后的运行。 000: $\times 1$ 倍 001: $\times 1/2$ 倍 010: $\times 1/3$ 倍 011: $\times 1/4$ 倍 100: $\times 1/8$ 倍 上述以外: 禁止设定
11 ~ 9	BFC[2:0]	011	R/W	总线时钟 (Bϕ) 频率的分频比 对于 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, 就不保证以后的运行。 000: $\times 1$ 倍 001: $\times 1/2$ 倍 010: $\times 1/3$ 倍 011: $\times 1/4$ 倍 100: $\times 1/8$ 倍 上述以外: 禁止设定
8 ~ 6	PFC[2:0]	011	R/W	外围时钟 (Pϕ) 频率的分频比 对于 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, 就不保证以后的运行。 000: $\times 1$ 倍 001: $\times 1/2$ 倍 010: $\times 1/3$ 倍 011: $\times 1/4$ 倍 100: $\times 1/8$ 倍 上述以外: 禁止设定
5 ~ 3	MIFC[2:0]	011	R/W	MTU2S 时钟 (MIϕ) 频率的分频比 对于 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”的值时, 就不保证以后的运行。 000: $\times 1$ 倍 001: $\times 1/2$ 倍 010: $\times 1/3$ 倍 011: $\times 1/4$ 倍 100: $\times 1/8$ 倍 上述以外: 禁止设定
2 ~ 0	MPFC[2:0]	011	R/W	MTU2 时钟 (MPϕ) 频率的分频比 对于 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, 就不保证以后的运行。 000: $\times 1$ 倍 001: $\times 1/2$ 倍 010: $\times 1/3$ 倍 011: $\times 1/4$ 倍 100: $\times 1/8$ 倍 上述以外: 禁止设定

4.4.2 振荡停止检测控制寄存器 (OSCCR)

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

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

位	位名	初始值	R/W	说明
7 ~ 3	—	全 0	R	保留位 读写值总是 0。
2	OSCSTOP	0	R	振荡停止检测标志 [置位条件] - 在正常运行中检测到时钟停止输入时 - 已转移到软件待机模式时 [清除条件] - 从 $\overline{\text{RES}}$ 引脚进行上电复位时 - 从软件待机模式返回时
1	—	0	R	保留位 读写值总是 0。
0	OSCERS	0	R/W	选择振荡停止检测标志的输出 选择是否从 $\overline{\text{WDTOVF}}$ 引脚输出振荡停止检测标志信号。 0: 从 $\overline{\text{WDTOVF}}$ 引脚输出 WDT 上溢信号。 1: 从 $\overline{\text{WDTOVF}}$ 引脚输出 WDT 上溢信号和振荡停止检测标志信号。

4.5 更改频率的方法

能通过改变分频器的分频比，更改内部时钟、总线时钟、外围时钟、MTU2S 时钟和 MTU2 时钟的频率。在软件中通过频率控制寄存器 (FRQCR) 对这些频率进行控制的方法如下：

1. 在初始状态下， $\text{IFC2} \sim \text{IFC0} = \text{H}'011$ ($\times 1/4$ 倍)、 $\text{BFC2} \sim \text{BFC0} = \text{H}'011$ ($\times 1/4$ 倍)、 $\text{PFC2} \sim \text{PFC0} = \text{H}'011$ ($\times 1/4$ 倍)、 $\text{MIFC2} \sim \text{MIFC0} = \text{H}'011$ ($\times 1/4$ 倍)、 $\text{MPFC2} \sim \text{MPFC0} = \text{H}'011$ ($\times 1/4$ 倍)。
2. 停止除 CPU、内部 ROM 和内部 RAM 以外的模块。
3. 将 $\text{IFC2} \sim \text{IFC0}$ 、 $\text{BFC2} \sim \text{BFC0}$ 、 $\text{PFC2} \sim \text{PFC0}$ 、 $\text{MIFC2} \sim \text{MIFC0}$ 和 $\text{MPFC2} \sim \text{MPFC0}$ 位设定为目标值。因为 PLL 电路的频率倍增率固定为 $\times 8$ 倍，所以只能通过设定分频比来决定频率。此时的频率必须设定为内部时钟 ($\text{I}\phi$) $\geq$ 总线时钟 ($\text{B}\phi$) $\geq$ 外围时钟 ($\text{P}\phi$)。另外，在使用 MTU2S 时钟和 MTU2 时钟时，必须设定内部时钟 ($\text{I}\phi$) $\geq$ MTU2S 时钟 ($\text{MI}\phi$) $\geq$ MTU2 时钟 ($\text{MP}\phi$) $\geq$ 外围时钟 ($\text{P}\phi$)，并且总线时钟 ($\text{B}\phi$) $\geq$ MTU2 时钟 ($\text{MP}\phi$)。
必须在内部 ROM 或者内部 RAM 内执行 FRQCR 的改写处理程序。
4. 发行 FRQCR 改写指令后，在经过 $(1 \sim 24n) \text{ cyc} + 11\text{B}\phi + 7\text{P}\phi$ 后进行切换。
 n: FRQCR 的 BFC 位设定的分频比 (1、1/2、1/3、1/4、1/8)
 cyc: 通过 PLL 将 EXTAL 引脚的输入时钟频率递增 8 倍后的时钟

【注】 (1 ~ 24n) 取决于内部状态。

4.6 振荡器

提供时钟的方法有连接晶体谐振器和输入外部时钟 2 种方法。

4.6.1 连接晶体谐振器的方法

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

有关晶体谐振器和 LSI 的相容性，请与晶体谐振器厂家商讨。

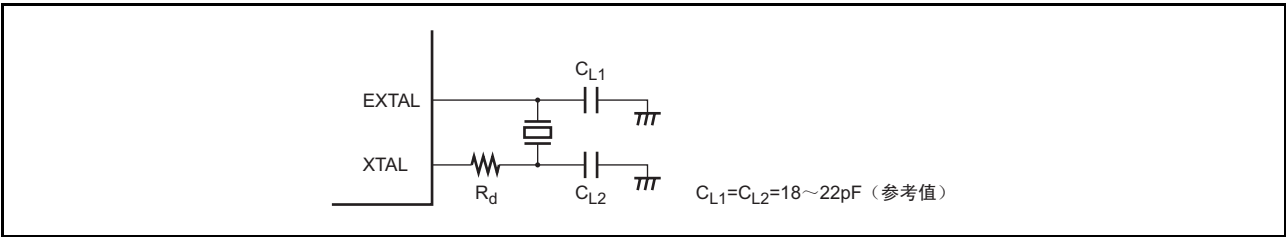


图 4.2 晶体谐振器的连接例子

表 4.6 阻尼电阻值（参考值）

频率（MHz）	5	8	10	12.5
R_d （ Ω ）（参考值）	500	200	0	0

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

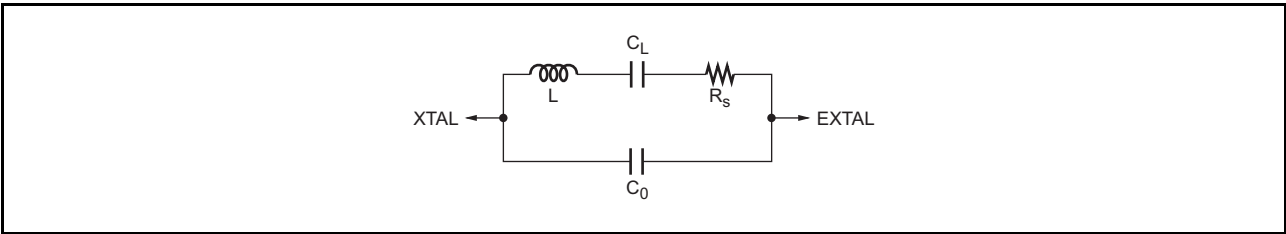


图 4.3 晶体谐振器的等效电路

表 4.7 晶体谐振器的特性

频率（MHz）	5	8	10	12.5
R_s Max.（ Ω ）（参考值）	120	80	60	50
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中停止外部时钟，就必须输入高电平。在工作时，外部输入时钟的频率必须在 5 ~ 12.5MHz 之间，XTAL 引脚的寄生电容不能大于 10pF。

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



图 4.4 外部时钟的连接例子

4.7 振荡停止的检测功能

为了防止因某种系统异常而停止振荡器的振荡，本 CPG 内置了检测时钟停止的功能。

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

即使在软件待机状态下，也能将上述引脚置为高阻抗。详细内容请参照“附录 A. 引脚状态”。在解除软件待机状态后，进入正常运行状态。另外，如果在非软件待机状态下出现振荡停止等异常情况，其他的 LSI 运行就不稳定。此时，即使重新开始振荡，包括上述引脚的 LSI 运行也不稳定。

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

4.8 使用时的注意事项

4.8.1 有关谐振器的注意事项

因为谐振器的各种特性与用户的电路板设计密切相关，所以请用户参照本节介绍的谐振器的连接例子，在给予充分的评估后再使用。因为振荡电路的电路常数因谐振器、安装电路的寄生电容而不同，所以必须在和振荡器厂家商讨后再作决定。外加在振荡引脚的电压不能超过最大额定值。

4.8.2 电路板设计的注意事项

本 LSI 虽然采取了防止辐射噪声的措施，但是，如果需要进一步降低辐射噪声，建议使用多层电路板和系统接地专用层。

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

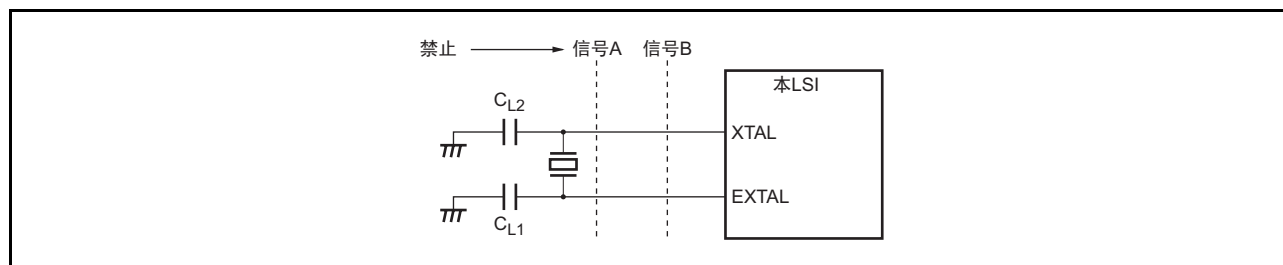


图 4.5 有关振荡电路电路板设计的注意事项

PLL 电路的外接推荐电路如图 4.6 所示。必须将 PLLV_{SS}、V_{CC} 和 V_{SS} 远离电路板的电源供给源，并在引脚的附近插入旁路电容 CB 和 CPB。

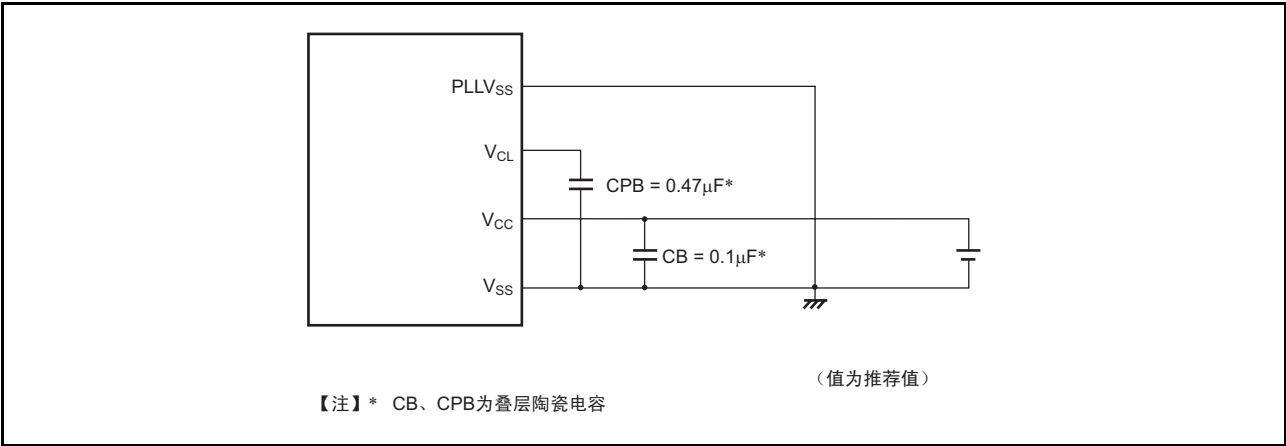


图 4.6 PLL 电路的外接推荐电路

第 5 章 异常处理

5.1 概要

5.1.1 异常处理的种类和优先级

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

表 5.1 异常源的种类和优先级

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

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

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

5.1.2 异常处理的运行

按照表 5.2 所示的时序，在检测到各异常源后开始异常处理。

表 5.2 检测异常源和开始异常处理的时序

异常处理		检测异常源和开始异常处理的时序
复位	上电复位	在 $\overline{\text{RES}}$ 引脚从低电平变为高电平或者 WDT 上溢时开始异常处理。
	手动复位	在 $\overline{\text{MRES}}$ 引脚从低电平变为高电平或者 WDT 上溢时开始异常处理。
地址错误		在指令的解码阶段检测到地址错误或者中断，在执行中的指令结束后开始异常处理。
中断		
指令	陷阱指令	通过执行 TRAPA 指令开始异常处理。
	一般非法指令	在对延迟转移指令（延迟槽）以外的未定义代码进行解码时开始异常处理。
	槽非法指令	当对紧接在延迟转移指令（延迟槽）之后的未定义代码或者 PC 改写指令进行解码时开始异常处理

在启动异常处理时，CPU 的运行如下：

(1) 复位的异常处理

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

(2) 地址错误、中断和指令的异常处理

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

5.1.3 异常处理向量表

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

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

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

表 5.3 异常处理向量表

异常源		向量号	向量表地址偏移量
上电复位	PC	0	H'00000000 ~ H'00000003
	SP	1	H'00000004 ~ H'00000007
手动复位	PC	2	H'00000008 ~ H'0000000B
	SP	3	H'0000000C ~ H'0000000F
一般非法指令		4	H'00000010 ~ H'00000013
（系统保留）		5	H'00000014 ~ H'00000017
槽非法指令		6	H'00000018 ~ H'0000001B
（系统保留）		7	H'0000001C ~ H'0000001F
		8	H'00000020 ~ H'00000023
CPU 地址错误		9	H'00000024 ~ H'00000027
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

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

表 5.4 异常处理向量表地址的计算方法

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

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

5.2 复位

5.2.1 复位的种类

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

表 5.5 复位状态

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

5.2.2 上电复位

(1) $\overline{\text{RES}}$ 引脚的上电复位

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

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

1. 从异常处理向量表中取出程序计数器（PC）的初始值（执行的起始地址）。
 2. 从异常处理向量表中取出堆栈指针（SP）的初始值。
 3. 将向量基址寄存器（VBR）清除为 H'00000000，状态寄存器（SR）的中断屏蔽位（I3～I0）置 H'F（B'1111）。
 4. 分别给 PC 和 SP 设定从异常处理向量表中取出的值，然后开始执行程序。
- 必须在接通系统电源时进行上电复位处理。

(2) WDT 的上电复位

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

在 WDT 引起的复位时，时钟振荡器（CPG）的频率控制寄存器（FRQCR）和看门狗定时器（WDT）的寄存器被初始化（只有在 $\overline{\text{RES}}$ 引脚的上电复位时，这些寄存器才被初始化）。

如果 $\overline{\text{RES}}$ 引脚输入信号引起的复位和 WDT 上溢引起的复位同时发生，就优先进行 $\overline{\text{RES}}$ 引脚引起的复位，并将 WTCSR 的 WOVF 位清 0。在开始进行 WDT 的上电复位处理时，CPU 的运行如下：

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

5.2.3 手动复位

如果在 $\overline{\text{RES}}$ 引脚为高电平时将 $\overline{\text{MRES}}$ 引脚置为低电平，本 LSI 就进入手动复位状态。为了确保本 LSI 的复位， $\overline{\text{MRES}}$ 引脚的低电平在软件待机时（时钟停止运行时），必须至少保持 WDT 设定的振荡稳定时间；在时钟运行时，必须至少保持 20tcyc。在手动复位的状态下，CPU 的内部状态被初始化，而内部外围模块的各寄存器不被初始化。如果在总线周期中进入手动复位状态，就在等待总线周期结束后开始手动复位的异常处理。因此，总线周期不会因手动复位而中途停止。但是，一旦将 $\overline{\text{MRES}}$ 置为低电平，就必须将低电平保持到总线周期结束并进入手动复位状态为止（低电平必须至少保持最长的总线周期）。有关手动复位状态下的各引脚状态，请参照“附录 A. 引脚状态”。在手动复位的状态下，如果 $\overline{\text{MRES}}$ 引脚在保持一定期间的低电平后变为高电平，就开始手动复位的异常处理。此时，CPU 的运行步骤和上电复位的异常处理相同。

5.3 地址错误

5.3.1 地址错误的发生源

如表 5.6 所示，在取指令或者读写数据时发生地址错误。

表 5.6 总线周期和地址错误

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

5.3.2 地址错误的异常处理

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

1. 将状态寄存器（SR）压栈。
2. 将程序计数器（PC）压栈。被压栈的 PC 值是发生此异常处理的指令地址。但是，如果发生此异常处理的指令已分配到延迟槽，就将前一条延迟转移指令的地址压栈。
3. 从对应地址错误的异常处理向量表中取出异常服务程序的起始地址，并从该地址开始执行程序。此时的转移不是延迟转移。

5.4 中断

5.4.1 中断源

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

表 5.7 中断源

种类	请求源	源数
NMI	NMI 引脚（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）	28
	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 和 A/D_1）	2
	比较匹配定时器（CMT_0 和 CMT_1）	2
	串行通信接口（SCI_0、SCI_1 和 SCI_2）	12
	端口输出允许（POE）	3
	同步串行通信单元（SSU）	3
	I ² C 总线接口 2（I ² C2）	5
	控制器局域网（RCAN-ET）	5

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

5.4.2 中断优先级

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

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

表 5.8 中断优先级

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

5.4.3 中断异常处理

如果发生中断，就通过中断控制器（INTC）判断优先级。NMI 随时被接受，而其他中断只有在其优先级高于状态寄存器（SR）的中断屏蔽位（I3 ~ I0）设定的优先级时才被接受。

如果接受中断，就开始中断异常处理。在中断异常处理中，CPU 将 SR 和程序计数器（PC）压栈，并将接受的中断优先级的值写到 SR 的 I3 ~ I0 位。但是在 NMI 的情况下，优先级为 16，而 I3 ~ I0 位的设定值为 H'F（优先级 15）。从接受的中断所对应的异常处理向量表中取出异常服务程序的起始地址，并在转移到该地址后开始执行程序。有关中断异常处理，请参照“6.6 运行说明”。

5.5 指令的异常

5.5.1 指令异常的种类

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

表 5.9 指令异常的种类

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

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

5.5.2 陷阱指令

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

1. 将状态寄存器（SR）压栈。
2. 将程序计数器（PC）压栈。被压栈的 PC 值为 TRAPA 指令的下一条指令的起始地址。
3. 从对应 TRAPA 指令指定向量号的异常处理向量表中取出异常服务程序的起始地址，并在转移到该地址后开始执行程序。此时的转移不是延迟转移。

5.5.3 槽非法指令

紧接在延迟转移指令之后的指令称为“分配到延迟槽的指令”。当分配到延迟槽的指令为未定义代码时，如果对此未定义的代码进行解码，就开始槽非法指令的异常处理；当分配到延迟槽的指令为 PC 改写指令时，如果对 PC 改写指令进行解码，就开始槽非法指令的异常处理。在进行槽非法指令的异常处理时，CPU 的运行如下：

1. 将 SR 压栈。
2. 将 PC 压栈。被压栈的 PC 值为未定义代码或者 PC 改写指令的前一条延迟转移指令的地址。
3. 从发生的异常所对应的异常处理向量表中取出异常服务程序的起始地址，并在转移到该地址后开始执行程序。此时的转移不是延迟转移。

5.5.4 一般非法指令

如果对紧接在延迟转移指令（延迟槽）之后的未定义代码进行解码，就开始一般非法指令的异常处理。此时，CPU 的运行步骤和槽非法指令的异常处理相同。但是，不同于槽非法指令的异常处理是：被压栈的 PC 值是此未定义代码的起始地址。

5.6 异常处理的接受

如表 5.10 所示，如果复位以外的异常源发生在紧接着延迟槽或者中断禁止指令之后，就有可能不被立即接受而被保留。此时，在解码完能接受异常的指令后才接受这些异常源。

表 5.10 发生在紧接着延迟槽和中断禁止指令之后的异常源

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

【符号说明】 ○：能接受。

×：不能接受。

—：不可能的情况。

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

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

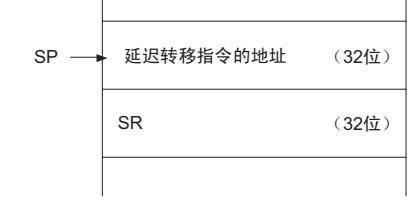
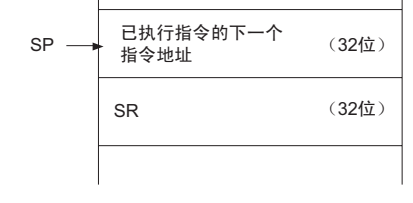
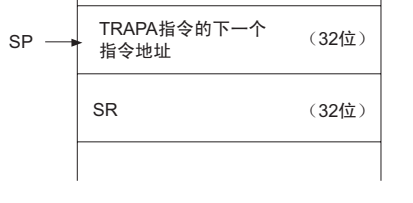
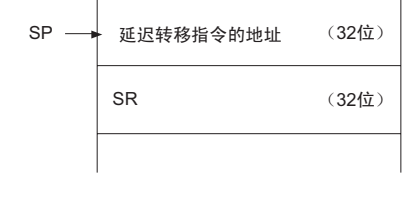
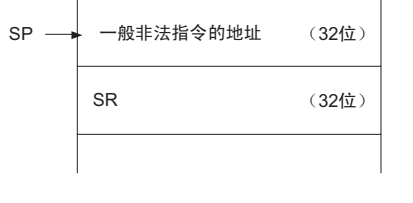
*3 在延迟转移后（在延迟槽指令和延迟转移目标指令之间）被接受。

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

5.7 异常处理后的堆栈状态

异常处理结束后的堆栈状态如表 5.11 所示。

表 5.11 异常处理结束后的堆栈状态

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

5.8 使用时的注意事项

5.8.1 堆栈指针 (SP) 的值

SP 值必须是 4 的倍数。否则，就会在异常处理的堆栈存取时发生地址错误。

5.8.2 向量基址寄存器 (VBR) 的值

VBR 值必须是 4 的倍数。否则，就会在异常处理的堆栈存取时发生地址错误。

5.8.3 地址错误异常处理的堆栈存取中发生的地址错误

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

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

5.8.4 有关槽非法指令异常处理的注意事项

本 LSI 的槽非法指令异常处理的规格和以往的 SH-2 有以下不同点：

- 以往的 SH-2 : LDC Rm,SR、LDC.L @Rm+,SR 不是槽非法指令。
- 本 LSI : LDC Rm,SR、LDC.L @Rm+,SR 是槽非法指令。

与此有关的本公司软件产品的对应情况如下：

(1) 编译程序

V.4 以后版本的编译程序没有将该指令分配到延迟槽。

(2) μ ITRON 规格实时 OS

(a) HI7000/4、HI-SH7

在 OS 内，延迟槽中没有该指令。

(b) HI7000

在 OS 内，延迟槽中有该指令，所以本 LSI 发生槽非法指令。

(c) 其他

在通过汇编程序进行记述或者导入中间件时，本 LSI 有可能发生槽非法指令。

上述指令串的检查程序登载于本公司网站的产品信息页上，如果需要，请从主页下载该检查程序进行确认。

第 6 章 中断控制器 (INTC)

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

6.1 特点

- 能将中断优先级设定为 16 级
- NMI 噪声消除功能
- 能将中断的发生信号输出到外部 ($\overline{\text{IRQOUT}}$ 引脚)

INTC 的框图如图 6.1 所示。

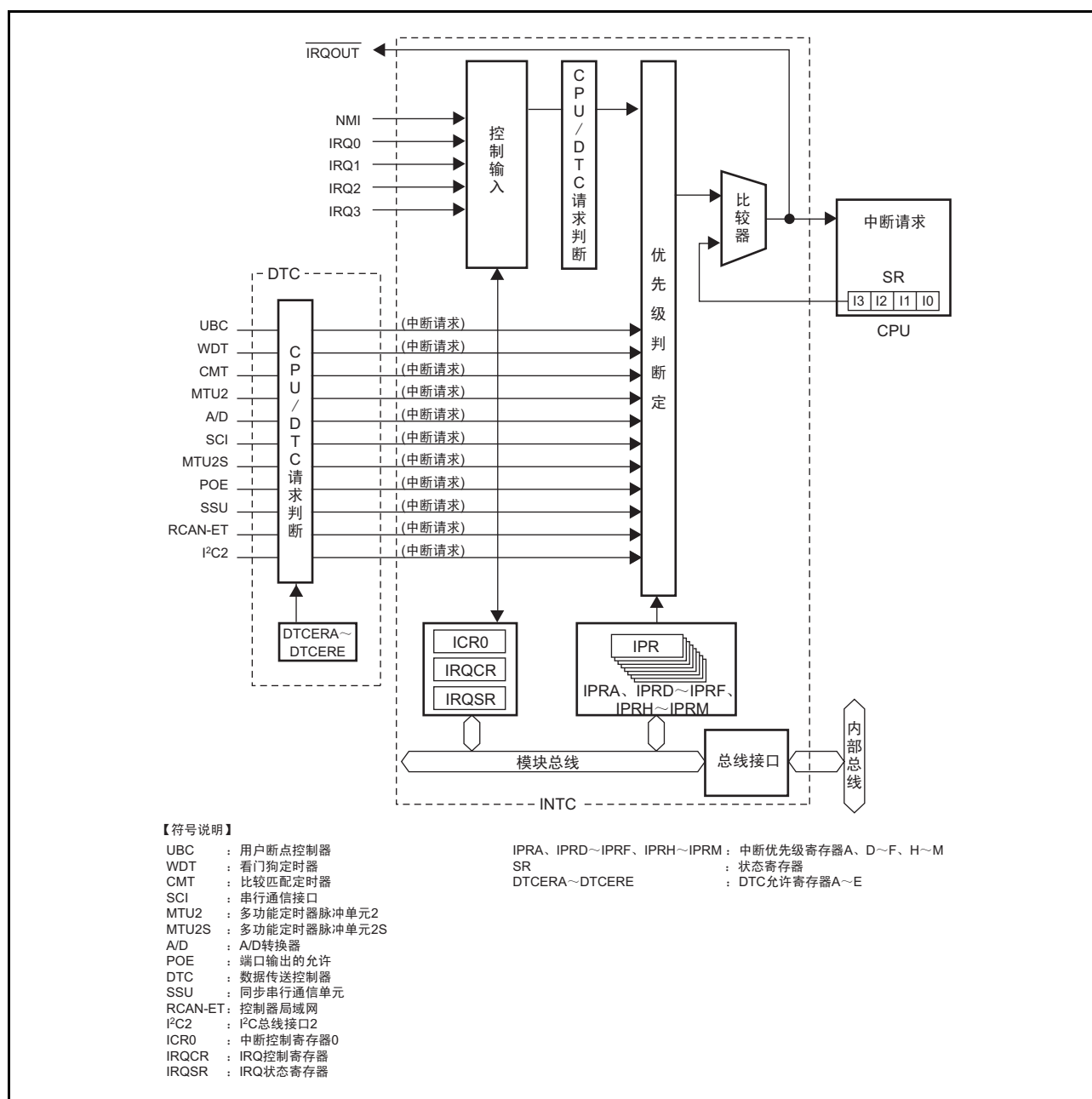


图 6.1 INTC 的框图

6.2 输入 / 输出引脚

INTC 的引脚如表 6.1 所示。

表 6.1 引脚结构

名称	引脚名	输入 / 输出	功能
非屏蔽中断输入引脚	NMI	输入	输入不可屏蔽的中断请求信号。
中断请求输入引脚	IRQ0 ~ IRQ3	输入	输入可屏蔽的中断请求信号。
中断请求输出引脚	IRQOUT	输出	输出通知发生中断源的信号。

6.3 寄存器说明

INTC 有以下寄存器。有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 25 章 寄存器一览”。

表 6.2 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
中断控制寄存器 0	ICR0	R/W	H'x000	H'FFFFE900	8、16
IRQ 控制寄存器	IRQCR	R/W	H'0000	H'FFFFE902	8、16
IRQ 状态寄存器	IRQSR	R/W	H'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 是 16 位寄存器，设定外部中断输入引脚 NMI 的输入信号检测模式以及表示 NMI 引脚的输入电平。

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

【注】* NMI引脚为高电平时此位是1，为低电平时是0。

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

6.3.2 IRQ 控制寄存器 (IRQCR)

IRQCR 是 16 位寄存器，设定外部中断输入引脚 IRQ0 ~ 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	初始值	R/W	说明
15 ~ 8	—	全为 0	R	保留位 读写值总是 0。
7 6	IRQ31S IRQ30S	0 0	R/W R/W	IRQ3 检测的选择 选择是在低电平、下降沿、上升沿还是在双边沿检测 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

位	位名	初始值	R/W	说明
5 4	IRQ21S IRQ20S	0 0	R/W R/W	IRQ2 检测的选择 选择是在低电平、下降沿、上升沿还是在双边沿检测 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	IRQ1 检测的选择 选择是在低电平、下降沿、上升沿还是在双边沿检测 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	IRQ0 检测的选择 选择是在低电平、下降沿、上升沿还是在双边沿检测 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 是 16 位寄存器，表示外部中断输入引脚 IRQ0 ~ IRQ3 的状态和中断请求状态。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	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是1，为低电平时是0。

位	位名	初始值	R/W	说明
15 ~ 12	—	全 1	R	保留位 读写值总是 1。
11	IRQ3L	*	R	表示 IRQ3 的引脚状态。 0: IRQ3 引脚为低电平 1: IRQ3 引脚为高电平
10	IRQ2L	*	R	表示 IRQ2 的引脚状态。 0: IRQ2 引脚为低电平 1: IRQ2 引脚为高电平
9	IRQ1L	*	R	表示 IRQ1 的引脚状态。 0: IRQ1 引脚为低电平 1: IRQ1 引脚为高电平

位	位名	初始值	R/W	说明
8	IRQ0L	*	R	表示 IRQ0 的引脚状态。 0: IRQ0 引脚为低电平 1: IRQ0 引脚为高电平
7 ~ 4	—	全 0	R	保留位 读写值总是 0。
3	IRQ3F	0	R/W	表示 IRQ3 中断请求的状态。 在设定了电平检测时 0: 没有 IRQ3 中断请求 [清除条件] • 给 IRQ3 引脚输入高电平 1: 有 IRQ3 中断请求 [置位条件] • 给 IRQ3 引脚输入低电平 在设定了边沿检测时 0: 未检测到 IRQ3 中断请求 [清除条件] • 在读 IRQ3F=1 的状态后写 0 • 接受 IRQ3 中断 1: 检测到 IRQ3 中断请求 [置位条件] • 检测到 IRQ3 引脚的指定边沿
2	IRQ2F	0	R/W	表示 IRQ2 中断请求的状态。 在设定了电平检测时 0: 没有 IRQ2 中断请求 [清除条件] • 给 IRQ2 引脚输入高电平 1: 有 IRQ2 中断请求 [置位条件] • 给 IRQ2 引脚输入低电平 在设定了边沿检测时 0: 未检测到 IRQ2 中断请求 [清除条件] • 在读 IRQ2F=1 的状态后写 0 • 接受 IRQ2 中断 1: 检测到 IRQ2 中断请求 [置位条件] • 检测到 IRQ2 引脚的指定边沿

位	位名	初始值	R/W	说明
1	IRQ1F	0	R/W	表示 IRQ1 中断请求的状态。 在设定了电平检测时 0: 没有 IRQ1 中断请求 [清除条件] - 给 IRQ1 引脚输入高电平 1: 有 IRQ1 中断请求 [置位条件] - 给 IRQ1 引脚输入低电平 在设定了边沿检测时 0: 未检测到 IRQ1 中断请求 [清除条件] - 在读 IRQ1F=1 的状态后写 0 - 接受 IRQ1 中断 1: 检测到 IRQ1 中断请求 [置位条件] - 检测到 IRQ1 引脚的指定边沿
0	IRQ0F	0	R/W	表示 IRQ0 中断请求的状态。 在设定了电平检测时 0: 没有 IRQ0 中断请求 [清除条件] - 给 IRQ0 引脚输入高电平 1: 有 IRQ0 中断请求 [置位条件] - 给 IRQ0 引脚输入低电平 在设定了边沿检测时 0: 未检测到 IRQ0 中断请求 [清除条件] - 在读 IRQ0F=1 的状态后写 0 - 接受 IRQ0 中断 1: 检测到 IRQ0 中断请求 [置位条件] - 检测到 IRQ0 引脚的指定边沿

【注】 * 此位的初始值在对应的 IRQ 引脚为高电平时是 1，为低电平时是 0。

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

IPR 是 10 个 16 位可读写寄存器，设定除 NMI 以外的中断源优先级（优先级 15 ~ 0）。有关各中断源和 IPR 的对应，请参照表 6.3。通过给 bit15 ~ 12、bit11 ~ 8、bit7 ~ 4、bit3 ~ 0 的各 4 位设定 H'0 ~ H'F 范围内的值，决定对应中断请求的优先级。对于没有分配的保留位，必须设定 H'0 (B'0000)。

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

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

【注】 在此表中位名用一般名称表示，而在寄存器一览表中用模块名表示。

6.4 中断源

6.4.1 外部中断源

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

(1) NMI 中断

NMI 中断的优先级为 16, 并且随时被接受。检测 NMI 引脚的输入边沿, 能通过设定中断控制寄存器 0 (ICR0) 的 NMI 边沿选择位 (NMIE), 将检测的边沿选择为上升沿或者下降沿。

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

(2) IRQ3 ~ IRQ0 中断

IRQ 中断是由 IRQ0 ~ IRQ3 引脚输入产生的中断。能通过设定 IRQ 控制寄存器 (IRQCR) 的 IRQ 检测选择位 (IRQ31S、IRQ30S ~ IRQ01S 和 IRQ00S), 给各引脚选择低电平检测、下降沿检测、上升沿检测或者双边沿检测; 能通过中断优先级寄存器 A (IPRA), 给各引脚设定 0 ~ 15 的优先级。

如果将 IRQ 中断设定为低电平检测, 就在 IRQ 引脚为低电平期间将中断请求信号送给 INTC。当 IRQ 引脚变为高电平时, 就停止中断请求信号的传送。能通过读 IRQ 状态寄存器 (IRQSR) 的 IRQ 标志 (IRQ3F ~ IRQ0F) 确认是否有中断请求。

如果将 IRQ 中断设定为下降沿 (上升沿或者双边沿) 检测, 就在 IRQ 引脚从高电平变为低电平 (从低电平变为高电平或者从高电平变为低电平) 时检测到中断请求, 并将中断请求信号送给 INTC。IRQ 中断请求的检测结果一直保持到该中断请求被接受为止。另外, 能通过读 IRQ 状态寄存器 (IRQSR) 的 IRQ 标志 (IRQ3F ~ IRQ0F) 确认是否检测到 IRQ 中断请求, 能通过读 1 后写 0 来取消 IRQ 中断请求的检测结果。

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

IRQ3 ~ IRQ0 中断的框图如图 6.2 所示。

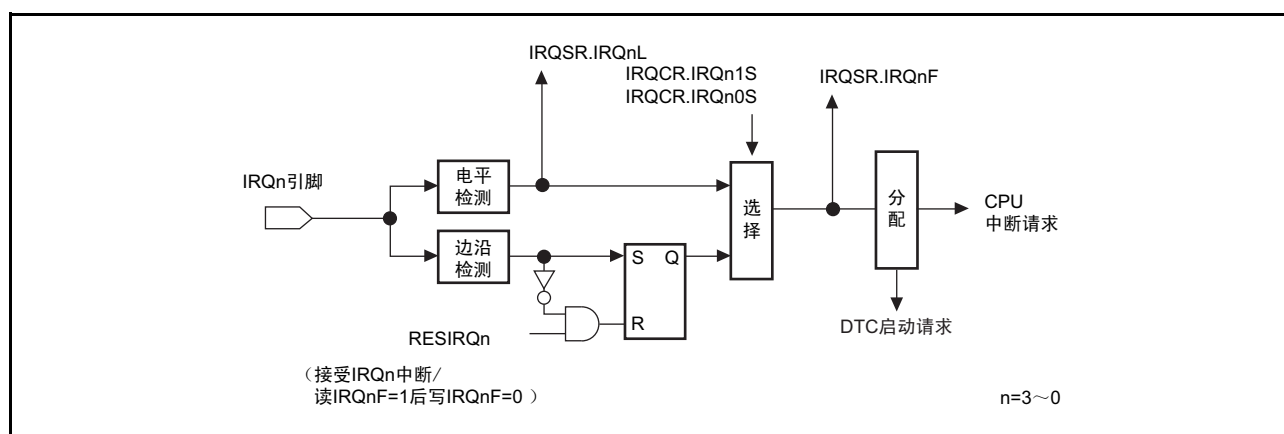


图 6.2 IRQ0 ~ IRQ3 中断控制

6.4.2 内部外围模块中断

内部外围模块中断是以下内部外围模块产生的中断。

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

6.4.3 用户断点中断

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

6.5 中断异常处理向量表

中断源、向量号、向量表地址偏移量和中断优先级如表 6.3 所示。

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

通过中断优先级寄存器 A、D ~ F、H ~ M (IPRA、IPRD ~ IPRF、IPRH ~ IPRM), 能按各引脚和模块将 IRQ 中断和内部外围模块中断的优先级任意设定为 0 ~ 15。但是, 因为小向量号的中断源分配了高优先级, 因此不能更改同一 IPR 指定的多个中断源的优先级。通过上电复位, 将 IRQ 中断和内部外围模块中断的优先级设定为 0。在将多个中断源设定为相同优先级的情况下, 如果同时发生中断, 就根据表 6.3 的默认优先级进行处理。

表 6.3 中断源、向量地址和中断优先级一览表

中断发生源	名称	向量号	向量表起始地址	IPR	默认优先级
用户断点		12	H'00000030	—	高 ↑ ↓ 低
外部引脚	NMI	11	H'0000002C	—	
	IRQ0	64	H'00000100	IPRA15 ~ IPRA12	
	IRQ1	65	H'00000104	IPRA11 ~ IPRA8	
	IRQ2	66	H'00000108	IPRA7 ~ IPRA4	
	IRQ3	67	H'0000010C	IPRA3 ~ IPRA0	
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

中断发生源	名称	向量号	向量表起始地址	IPR	默认优先级
MTU2_4	TGIA_4	120	H'000001E0	IPRF15 ~ IPRF12	高 ↑

中断发生源	名称	向量号	向量表起始地址	IPR	默认优先级
SSU	SSERI	232	H'000003A0	IPRM15 ~ IPRM12	高 ↑ ↓ 低
	SSRXI	233	H'000003A4		
	SSTXI	234	H'000003A8		
I ² C2*	IITEI	236	H'000003B0	IPRM11 ~ IPRM8	
	IISTPI	237	H'000003B4		
	IITXI	238	H'000003B8		
	IIRXI	239	H'000003BC		
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		

【注】 * I²C2 的中断向量地址有可能因中断源而被分开。

6.6 运行说明

6.6.1 中断运行的流程

以下说明发生中断时的运行流程，运行流程如图 6.3 所示。

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

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

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

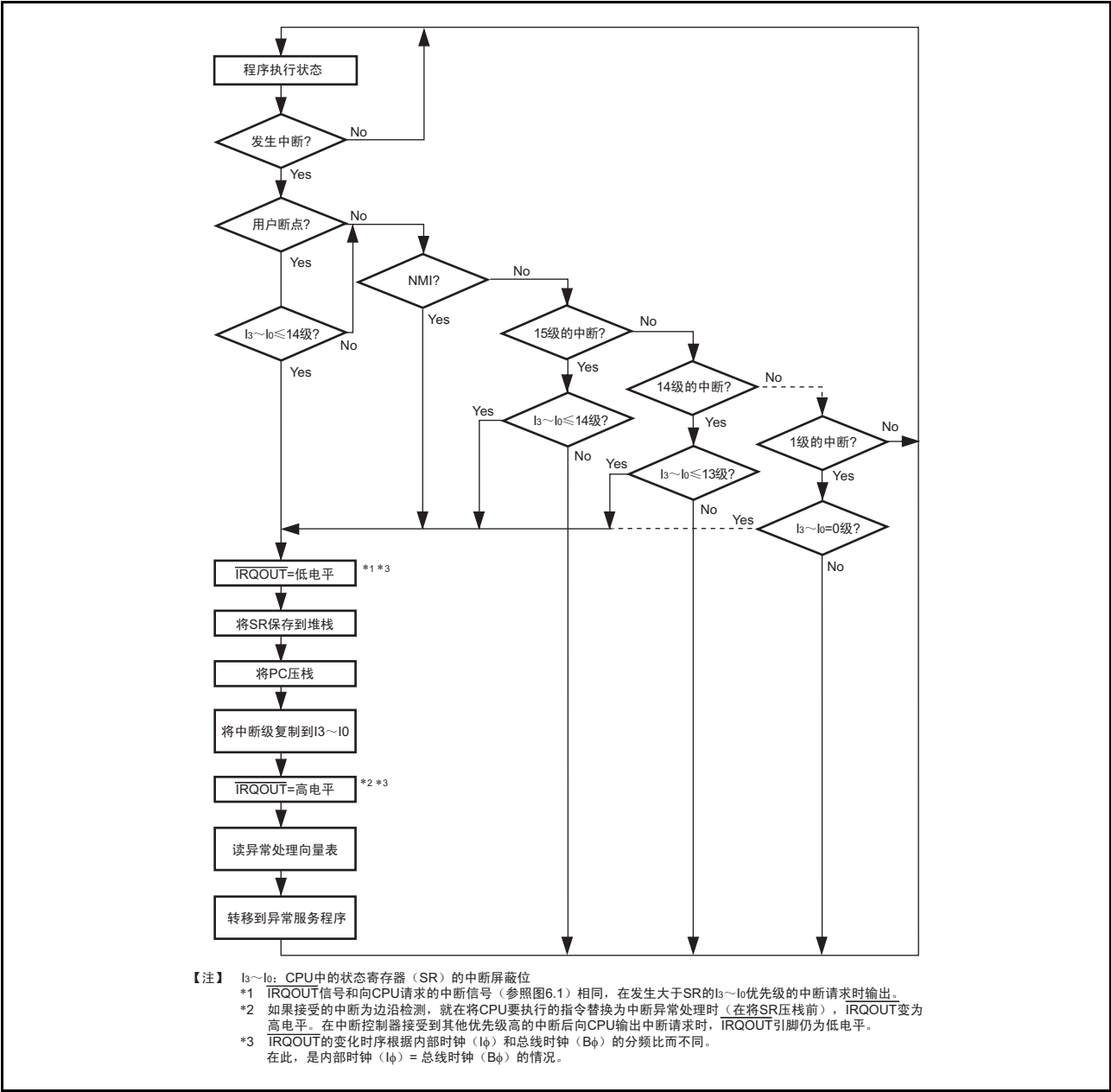


图 6.3 中断运行流程

6.6.2 中断异常处理结束后的堆栈状态

中断异常处理结束后的堆栈状态如图 6.4 所示。

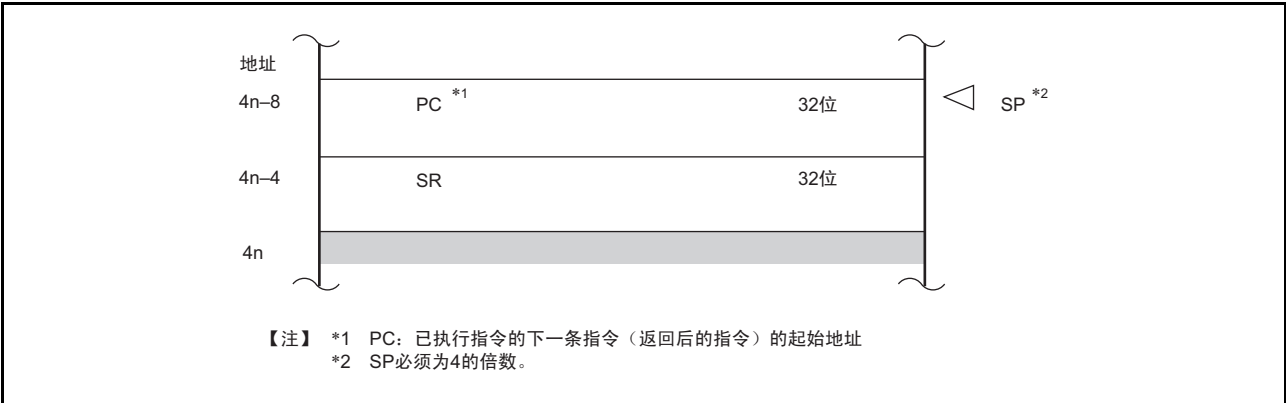


图 6.4 中断异常处理结束后的堆栈状态

6.7 中断响应时间

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

表 6.4 中断响应时间

项目		状态数			备注
		NMI	IRQ	外围模块	
DTC 的启动判断		—	$2 \times \text{Bcyc}$	$1 \times \text{Pcyc}$	
优先级的判断以及与 SR 屏蔽位进行比较的时间		$1 \times \text{lcyc} + 2 \times \text{Pcyc}$	$1 \times \text{lcyc} + 1 \times \text{Pcyc}$	$1 \times \text{lcyc} + 2 \times \text{Pcyc}$	
CPU 执行中的顺序结束前的等待时间		$X (\geq 0)$			最长的顺序是中断异常处理和地址错误异常处理, 为 $X = 7 \times \text{lcyc} + m1 + m2 + m3 + m4$ 。但是, 在中断的后续指令为屏蔽指令时, 时间可能更长。
从开始中断异常处理到开始取异常服务程序起始指令的时间		$8 \times \text{lcyc} + m1 + m2 + m3$			保存 SR、PC 以及取向量地址。
响应时间	合计	$9 \times \text{lcyc} + 2 \times \text{Pcyc} + m1 + m2 + m3 + X$	$9 \times \text{lcyc} + 1 \times \text{Pcyc} + 2 \times \text{Bcyc} + m1 + m2 + m3 + X$	$9 \times \text{lcyc} + 3 \times \text{Pcyc} + m1 + m2 + m3 + X$	
	最小时 *	$12 \times \text{lcyc} + 2 \times \text{Pcyc}$	$12 \times \text{lcyc} + 1 \times \text{Pcyc} + 2 \times \text{Bcyc}$	$12 \times \text{lcyc} + 3 \times \text{Pcyc}$	是 SR、PC 和向量表全部在内部 RAM 的情况。
	最大时	$16 \times \text{lcyc} + 2 \times \text{Pcyc} + 2(m1 + m2 + m3) + m4$	$16 \times \text{lcyc} + 1 \times \text{Pcyc} + 2 \times \text{Bcyc} + 2(m1 + m2 + m3) + m4$	$16 \times \text{lcyc} + 3 \times \text{Pcyc} + 2(m1 + m2 + m3) + m4$	

【注】 m1 ~ m4 为存取下述存储器时所需的状态数:
 m1: 保存 SR (写长字)
 m2: 保存 PC (写长字)
 m3: 读向量地址 (读长字)
 m4: 取中断服务程序的起始指令
 * 当 $m1 = m2 = m3 = m4 = 1 \times \text{lcyc}$ 时

6.8 通过中断请求信号进行的数据传送

能通过中断请求信号进行以下的数据传送：

- 启动DTC，CPU中断取决于DTC的设定

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

DTCE清除条件=DTC传送结束 · DTCECLR

中断源标志清除条件=DTC传送结束 · DTCECLR

其中，DTCECLR=DISEL+计数器0。

控制框图如图 6.5 和图 6.6 所示。

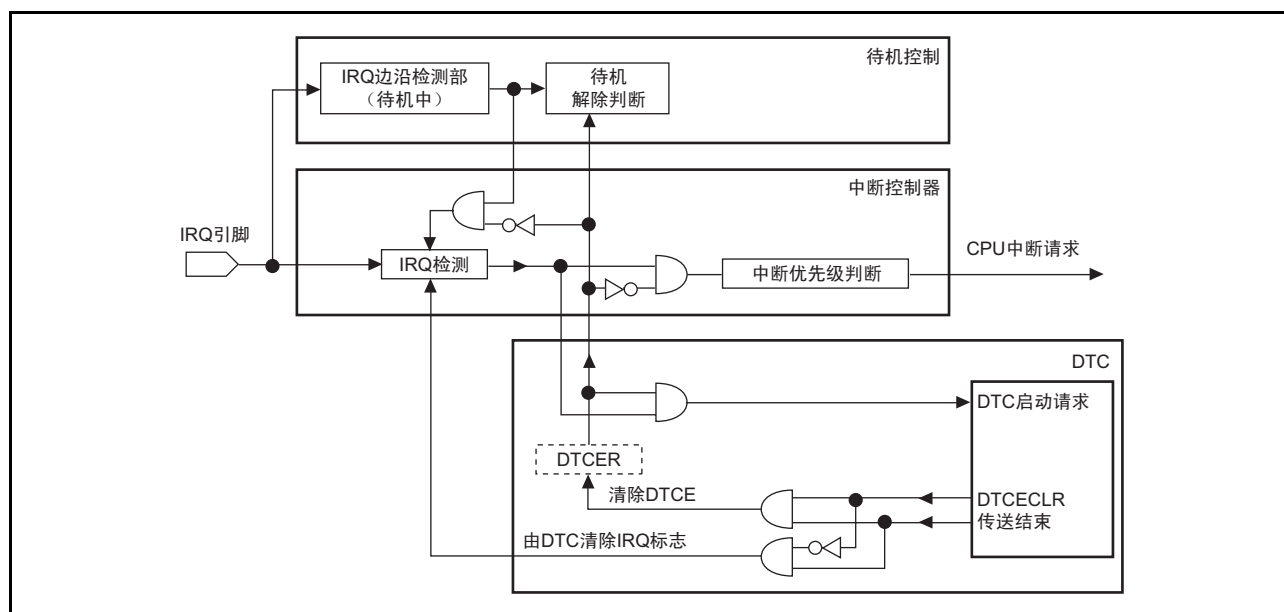


图 6.5 IRQ 中断控制框图

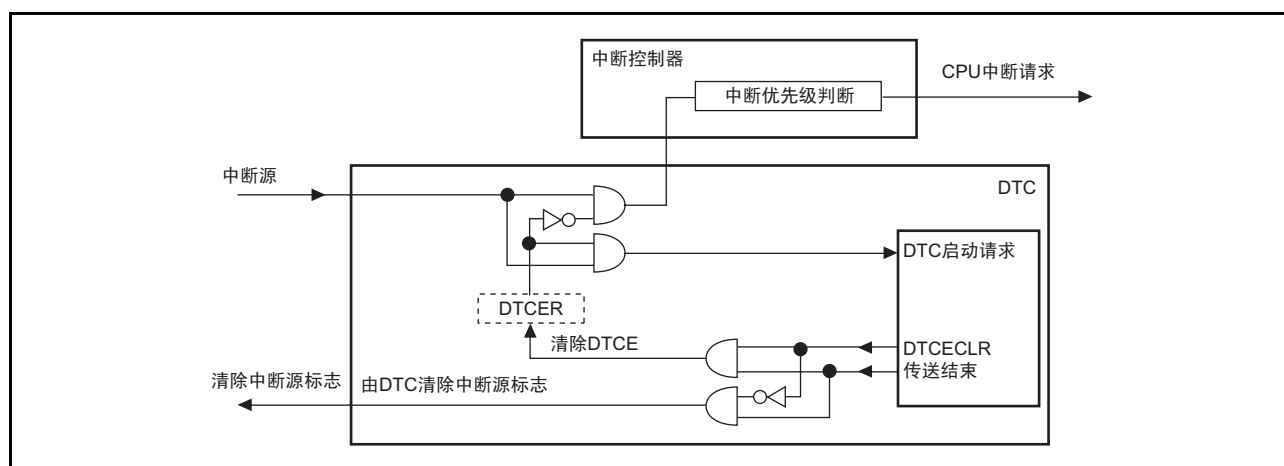


图 6.6 外围模块的中断控制框图

6.8.1 中断请求信号作为 DTC 启动源和 CPU 中断源的情况

1. 将DTC对应的DTCE位和DISEL位置1。
2. 在发生中断时，将启动源送给DTC。
3. 如果进行数据传送，DTC就将DTCE位清0，并向CPU请求中断，但是不清除启动源。
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。当传送计数器的值 $\neq 0$ 时，将DTCE位置1，允许下一次的数据传送；当传送计数器的值 $=0$ 时，通过中断处理程序进行所需的结束处理。

6.8.2 中断请求信号作为 DTC 启动源而不作为 CPU 中断源的情况

1. 将DTC对应的DTCE位置1，DISEL位清0。
2. 在发生中断时，将启动源送给DTC。
3. 如果进行数据传送，DTC就清除启动源。因为DTCE位保持为1，所以不向CPU请求中断。
4. 当传送计数器的值 $=0$ 时，将DTCE位清0，并向CPU请求中断。
5. CPU通过中断处理程序进行所需的结束处理。

6.8.3 中断请求信号作为 CPU 中断源而不作为 DTC 启动源

1. 将DTC对应的DTCE位清0。
2. 在发生中断时，向CPU请求中断。
3. CPU通过中断处理程序清除中断源，并进行所需的结束处理。

6.9 使用时的注意事项

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

第 7 章 用户断点控制器 (UBC)

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

7.1 特点

1. 能设定以下的断点比较条件:

断点通道数: 2个通道 (通道A和通道B)

能将用户断点设定为通道A和通道B独立或者连续 (顺序) 的条件 (顺序断点的设定是指: 通道A的断点条件成立后通道B的断点条件成立, 并且两者不发生在同一个总线周期。)

- 地址
能按位对32位地址的比较进行屏蔽。
能选择L总线地址 (LAB) 或者I总线地址 (IAB)。
- 数据
能进行32位的屏蔽。
能选择L总线数据 (LDB) 或者I总线数据 (IDB)。
- 总线周期
取指令周期或者数据存取周期。
- 读周期或者写周期
- 操作数长度
支持字节、字和长字。

2. 能执行用户指定的用户断点的中断异常处理程序。
3. 在取指令周期中, 能将用户断点设定在指令执行前或者在指令执行后。
4. 作为断点条件 (只对通道B), 最多能指定 $2^{12}-1$ 次的重复次数。
5. 支持4组转移源/转移目标缓冲器。

UBC 的框图如图 7.1 所示。

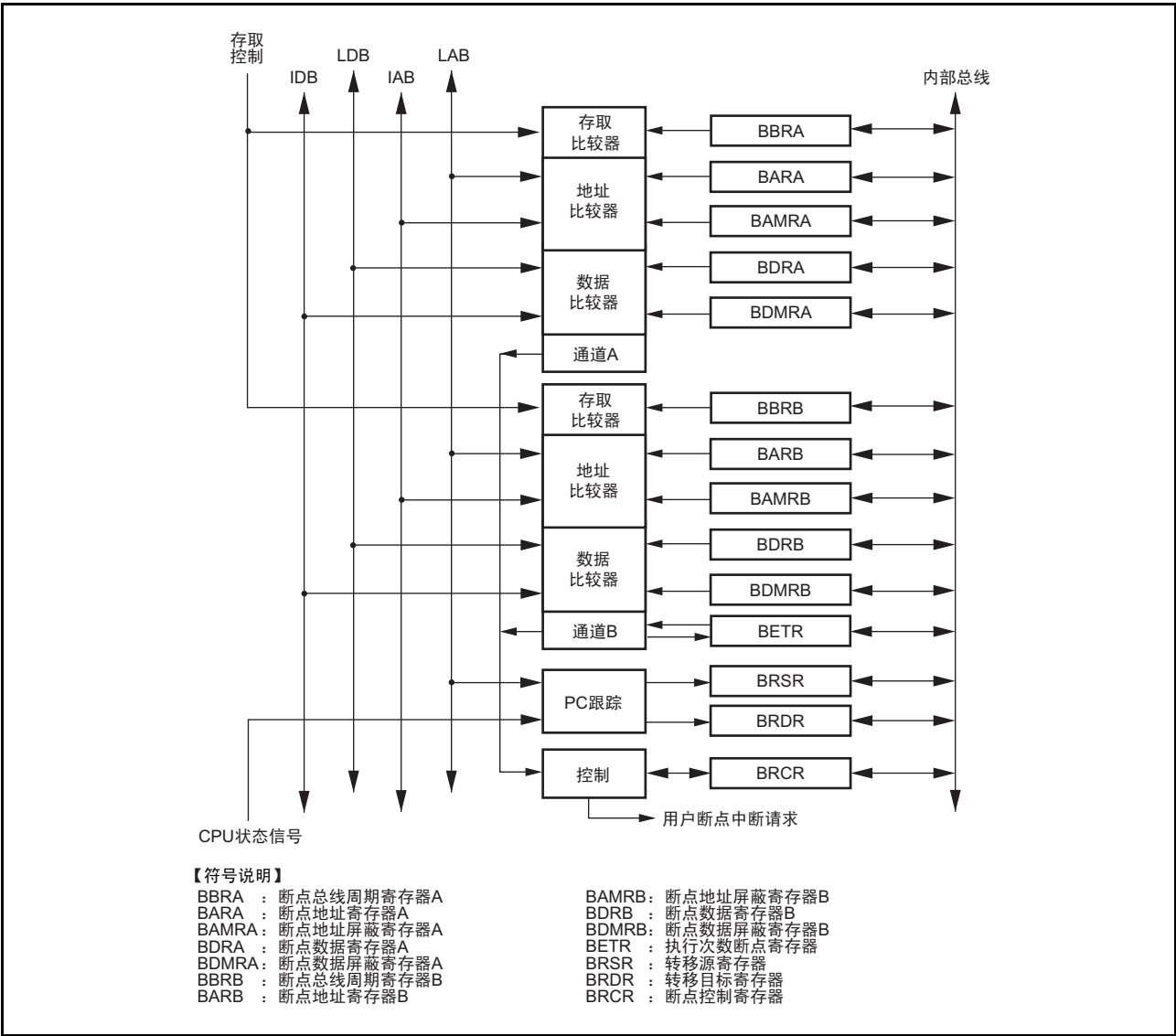


图 7.1 UBC 的框图

7.2 输入 / 输出引脚

UBC 的引脚如表 7.1 所示。

表 7.1 引脚结构

名称	引脚名	输入 / 输出	功能
用户断点触发输出	UBCTR $\overline{G}$	输出	是 UBC 条件成立时的触发输出引脚。

7.3 寄存器说明

UBC 有以下寄存器。有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 25 章 寄存器一览”。

表 7.2 寄存器结构

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

7.3.1 断点地址寄存器 A（BARA）

BARA 是 32 位可读写寄存器，指定作为通道 A 断点条件的地址。

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

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

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

7.3.2 断点地址屏蔽寄存器 A (BAMRA)

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

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

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

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

7.3.3 断点总线周期寄存器 A (BBRA)

BBRA 是 16 位可读写寄存器，将 (1) I 总线周期的总线主控 (2) L 总线周期或者 I 总线周期 (3) 取指令周期或者数据存取周期 (4) 读周期或者写周期 (5) 操作数长度指定为通道 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	初始值	R/W	说明
15 ~ 11	—	全 0	R	保留位 读写值总是 0。
10 ~ 8	CPA[2:0]	000	R/W	I 总线的总线主控选择 A 当选择 I 总线作为通道 A 断点条件的总线周期时，选择总线主控；当选择 L 总线作为总线周期时，此位无效。 000: 不进行条件比较 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 总线周期的选择 A 选择 L 总线周期或者 I 总线周期作为通道 A 断点条件的总线周期。 00: 不进行条件比较 01: 断点条件为 L 总线周期 10: 断点条件为 I 总线周期 11: 断点条件为 L 总线周期

位	位名	初始值	R/W	说明
5、4	IDA[1:0]	00	R/W	取指令 / 数据存取周期的选择 A 选择取指令周期或者数据存取周期作为通道 A 断点条件的总线周期。 00: 不进行条件比较 01: 断点条件为取指令周期 10: 断点条件为数据存取周期 11: 断点条件为取指令周期或者数据存取周期
3、2	RWA[1:0]	00	R/W	读写周期的选择 A 选择读周期或者写周期作为通道 A 断点条件的总线周期。 00: 不进行条件比较 01: 断点条件为读周期 10: 断点条件为写周期 11: 断点条件为读周期或者写周期
1、0	SZA[1:0]	00	R/W	操作数长度的选择 A 选择总线周期的操作数长度作为通道 A 的断点条件。 00: 断点条件不包含操作数长度 01: 断点条件为字节存取 10: 断点条件为字存取 11: 断点条件为长字存取 【注】 在指定操作数长度时, 地址边界和操作数长度必须一致。

【符号说明】 x: Don't care

7.3.4 断点数据寄存器 A (BDRA)

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

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

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

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

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

7.3.5 断点数据屏蔽寄存器 A (BDMRA)

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

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

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

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

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

7.3.6 断点地址寄存器 B (BARB)

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

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

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

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

7.3.7 断点地址屏蔽寄存器 B（BAMRB）

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

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

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

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

7.3.8 断点数据寄存器 B（BDRB）

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

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

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

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

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

7.3.9 断点数据屏蔽寄存器 B (BDMRB)

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

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

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

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

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

7.3.10 断点总线周期寄存器 B (BBRB)

BBRB 是 16 位可读写寄存器，将 (1) I 总线周期的总线主控 (2) L 总线周期或者 I 总线周期 (3) 取指令周期或者数据存取周期 (4) 读周期或者写周期 (5) 操作数长度指定为通道 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	初始值	R/W	说明
15 ~ 11	—	全 0	R	保留位 读写值总是 0。
10 ~ 8	CPB[2:0]	000	R/W	I 总线的总线主控选择 B 当选择 I 总线作为通道 B 断点条件的总线周期时，选择总线主控；当选择 L 总线作为总线周期时，此位无效。 000: 不进行条件比较 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

位	位名	初始值	R/W	说明
7、6	CDB[1:0]	00	R/W	L 总线周期 / I 总线周期的选择 B 选择 L 总线周期或者 I 总线周期作为通道 B 断点条件的总线周期。 00: 不进行条件比较 01: 断点条件为 L 总线周期 10: 断点条件为 I 总线周期 11: 断点条件为 L 总线周期
5、4	IDB[1:0]	00	R/W	取指令 / 数据存取周期的选择 B 选择取指令周期或者数据存取周期作为通道 B 断点条件的总线周期。 00: 不进行条件比较 01: 断点条件为取指令周期 10: 断点条件为数据存取周期 11: 断点条件为取指令周期或者数据存取周期
3、2	RWB[1:0]	00	R/W	读写周期的选择 B 选择读周期或者写周期作为通道 B 断点条件的总线周期。 00: 不进行条件比较 01: 断点条件为读周期 10: 断点条件为写周期 11: 断点条件为读周期或者写周期
1、0	SZB[1:0]	00	R/W	操作数长度的选择 B 选择总线周期的操作数长度作为通道 B 的断点条件。 00: 断点条件不包含操作数长度 01: 断点条件为字节存取 10: 断点条件为字存取 11: 断点条件为长字存取 【注】 在指定操作数长度时, 地址边界和操作数长度必须一致。

【符号说明】 x: Don't care

7.3.11 断点控制寄存器 (BRCR)

BRCR 设定以下条件:

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

BRCR 是 32 位可读写寄存器，有断点条件成立标志和设定各种断点条件的位。

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

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

位	位名	初始值	R/W	说明
31 ~ 22	—	全 0	R	保留位 读写值总是 0。
21、20	UTRGW[1:0]	00	R/W	UBCTRG 输出脉宽的选择 选择断点条件成立时的 UBCTRG 输出脉宽。 00: 禁止设定 01: UBCTRG 输出的脉宽为 3 ~ 4t _{Bcyc} 10: UBCTRG 输出的脉宽为 7 ~ 8t _{Bcyc} 11: UBCTRG 输出的脉宽为 15 ~ 16t _{Bcyc} 【注】 t _{Bcyc} 表示外部总线时钟 (Bφ=CK) 的周期。
19	UBIDB	0	R/W	用户断点的禁止 B 选择在通道 B 的断点条件成立时，是否禁止用户断点的中断请求。 0: 当断点条件成立时，允许用户断点的中断请求 1: 当断点条件成立时，禁止用户断点的中断请求
18	—	0	R	保留位 读写值总是 0。
17	UBIDA	0	R/W	用户断点的禁止 A 选择在通道 A 的断点条件成立时，是否禁止用户断点的中断请求。 0: 当断点条件成立时，允许用户断点的中断请求 1: 当断点条件成立时，禁止用户断点的中断请求
16	—	0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
15	SCMFCA	0	R/W	L 总线周期条件的成立标志 A 当给通道 A 设定的断点条件的 L 总线周期条件成立时, 就将此标志置 1。通过给此位写 0 清除此标志。 0: 通道 A 的 L 总线周期条件不成立 1: 通道 A 的 L 总线周期条件成立
14	SCMFCB	0	R/W	L 总线周期条件的成立标志 B 当给通道 B 设定的断点条件的 L 总线周期条件成立时, 就将此标志置 1。通过给此位写 0 清除此标志。 0: 通道 B 的 L 总线周期条件不成立 1: 通道 B 的 L 总线周期条件成立
13	SCMFDA	0	R/W	I 总线周期条件的成立标志 A 当给通道 A 设定的断点条件的 I 总线周期条件成立时, 就将此标志置 1。通过给此位写 0 清除此标志。 0: 通道 A 的 I 总线周期条件不成立 1: 通道 A 的 I 总线周期条件成立
12	SCMFDB	0	R/W	I 总线周期条件的成立标志 B 当给通道 B 设定的断点条件的 I 总线周期条件成立时, 就将此标志置 1。通过给此位写 0 清除此标志。 0: 通道 B 的 I 总线周期条件不成立 1: 通道 B 的 I 总线周期条件成立
11	PCTE	0	R/W	PC 跟踪的允许 0: 禁止 PC 跟踪 1: 允许 PC 跟踪
10	PCBA	0	R/W	PC 断点的选择 A 选择通道 A 取指令周期的断点时序是在指令执行前还是在指令执行后。 0: 将通道 A 的 PC 断点设定在指令执行前 1: 将通道 A 的 PC 断点设定在指令执行后
9、8	—	全 0	R	保留位 读写值总是 0。
7	DBEA	0	R/W	数据断点的允许 A 选择通道 A 的断点条件是否包含数据总线条件。 0: 通道 A 的断点条件不包含数据总线条件 1: 通道 A 的断点条件包含数据总线条件
6	PCBB	0	R/W	PC 断点的选择 B 选择通道 B 取指令周期的断点时序是在指令执行前还是在指令执行后。 0: 将通道 B 的 PC 断点设定在指令执行前 1: 将通道 B 的 PC 断点设定在指令执行后
5	DBEB	0	R/W	数据断点的允许 B 选择通道 B 的断点条件是否包含数据总线条件。 0: 通道 B 的断点条件不包含数据总线条件 1: 通道 B 的断点条件包含数据总线条件
4	—	0	R	保留位 读写值总是 0。

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

7.3.12 执行次数断点寄存器 (BETR)

BETR 是 16 位可读写寄存器。如果将通道 B 的执行次数断点条件置为有效, 此寄存器就指定执行断点的次数, 最大值为 $2^{12}-1$ 。每一次断点条件成立时, BETR 就减 1。如果 BETR 为 H'0001 并且断点条件成立, 就请求用户断点中断。

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

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

7.3.13 转移源寄存器 (BRSR)

BRSR 是 32 位只读寄存器, 保存转移源指令地址的 bit27 ~ 0。BRSR 有发生转移时被置 1 的标志位, 此标志位在读 BRSR 时或者将 PC 跟踪从禁止状态设定为允许时或者上电复位时被清 0, 而其他位在复位时不被初始化。4 个 BRSR 寄存器为队列结构, 被保存的寄存器每转移一次就移位一次。

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

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

位	位名	初始值	R/W	说明
31	SVF	0	R	BRSR 有效标志 表示转移源的地址是否已被保存。此标志在发生转移时置 1；在读 BRSR 时或者将 PC 跟踪从禁止状态设定为允许时或者上电复位时被清 0。 0: BRSR 寄存器的值无效 1: BRSR 寄存器的值有效
30 ~ 28	—	全 0	R	保留位 读写值总是 0。
27 ~ 0	BSA27 ~ BSA0	不定值	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 是 32 位只读寄存器，保存转移目标指令地址的 bit27 ~ 0。BRDR 有发生转移时被置 1 的标志位，此标志位在读 BRDR 时或者将 PC 跟踪从禁止状态设定为允许时或者上电复位时被清 0，而其他位在复位时不被初始化。4 个 BRDR 寄存器为队列结构，被保存的寄存器每转移一次就移位一次。

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

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

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

7.4 运行说明

7.4.1 用户断点运行的流程

从设定断点条件到用户断点异常处理的运行流程如下：

1. 将断点地址和要屏蔽的地址分别设定在断点地址寄存器（BARA 或者 BARB）和断点地址屏蔽寄存器（BAMRA 或者 BAMRB）；将断点数据和要屏蔽的数据分别设定在断点数据寄存器（BDRA 或者 BDRB）和断点数据屏蔽寄存器（BDMRA 或者 BDMRB）；将总线断点条件设定在断点总线周期寄存器（BBRA 或者 BBRB）。如果 BBRA 或者 BBRB 的 3 对控制位（即 L 总线周期/I 总线周期的选择位、取指令/数据存取的选择位、读写选择位）中的任意 1 对是 B'00，就不发生用户断点的暂停。将断点控制设定在 BRCCR 的位。必须在设定其他全部的断点相关寄存器后对 BBRA 或者 BBRB 进行设定。
2. 当断点条件成立时，UBC 在将用户断点的中断请求通知给 CPU 的同时，将各通道的 L 总线条件成立标志（SCMFCA 或者 SCMFCE）和 I 总线条件成立标志（SCMFDA 或者 SCMFDE）置位。
3. 能通过对应的条件成立标志（SCMFCA、SCMFDA、SCMFCE 和 SCMFDE）检查设定条件是否成立。在条件成立时，将标志置位但不能复位。为了能再次使用标志，必须先写 0，然后将标志复位。
4. 通道 A 和通道 B 设定的断点条件有可能几乎同时成立。此时，即使通知 CPU 的用户断点中断请求只有 1 个，也有可能设定 2 个条件成立标志。
5. 当选择 I 总线作为断点条件时，必须注意以下事项：
 - 将 CPU 和 DTC 连接到 I 总线。UBC 监视通过 BBRA 寄存器的 CPA2～CPA0 位和 BBRB 寄存器的 CPB2～CPB0 位选择的总线主控所生成的总线周期，并进行条件一致的比较。
 - 将 CPU 的 L 总线上的取指令产生的 I 总线周期定义为 I 总线上的取指令周期，其他定义为数据存取周期。
 - DTC 发行的 I 总线周期只为数据存取周期。
 - 当给 I 总线设定断点条件时，即使在由 CPU 执行指令引起的 I 总线周期中条件成立，也不能决定是哪条指令接受用户断点。

7.4.2 取指令周期的用户断点

1. 如果给断点总线周期寄存器（BBRA 或者 BBRB）设定“不包含 L 总线/取指令/读/字、长字或操作数长度”，断点条件就为 L 总线的取指令周期。能通过对应该通道的断点控制寄存器（BRCCR）的 PCBA 或者 PCBB 位，选择是在指令执行前还是在指令执行后发生用户断点的暂停。在将取指令周期设定为断点条件时，必须将断点地址寄存器（BARA 或者 BARB）的 LSB 清 0。如果此位被置 1，就不发生用户断点的暂停。
2. 在将取指令的用户断点设定在该指令执行前的状态下，如果断点条件成立，就会在取指令后执行指令时发生用户断点的暂停。因此，此功能不能用于因预取（在转移或者中断转移中取出的没被执行的指令）而取出的指令。如果给延迟转移指令的延迟槽设定此类断点条件，就会在执行延迟转移指令前发生用户断点的暂停。

【注】如果延迟转移指令不发生转移，就不将后续的指令视为延迟槽。
3. 如果将断点条件的用户断点设定在指令执行后，就在断点条件成立的指令执行后并且在下一条指令执行前发生用户断点的暂停。和执行前发生用户断点暂停的情况相同，此功能不能用于因预取而取出的指令。如果给延迟转移指令的延迟槽设定此种类断点条件，就不会在转移目标的第一条指令前发生用户断点的暂停。
4. 如果设定取指令周期，就忽视断点数据寄存器（BDRA 或者 BDRB）。因此，不能给取指令周期的用户断点设定断点数据。
5. 如果给取指令周期的用户断点条件设定 I 总线，就不对 I 总线上的取指令周期进行条件判断。详细内容请参照“7.4.1 用户断点运行的流程”的第 5 项。

7.4.3 数据存取周期的用户断点

1. 对数据存取的断点，如果将L总线指定为断点条件，就对执行指令的存取地址（和数据）进行条件比较并发生用户断点的暂停；如果将I总线指定为断点条件，就对I总线上包含CPU的所有总线主控所发行的数据存取周期的地址（和数据）进行条件比较并发生用户断点的暂停。有关发行给I总线上的CPU总线周期，请参照“7.4.1 用户断点运行的流程”的第5项。
2. 有关数据存取周期地址和各操作数长度的比较条件关系如表7.3所示。

表 7.3 数据存取周期地址和操作数长度的比较条件

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

例如，当给断点地址寄存器（BARA/BARB）设定地址H'00001003时，断点条件成立时的总线周期（假设其他全部条件都成立）包含以下含义：

对H'00001000进行长字存取

对H'00001002进行字存取

对H'00001003进行字节存取

3. 断点条件包含数值的情况
当断点条件包含数值时，给断点总线周期寄存器（BBRA/BBRB）指定长字、字或者字节的操作数长度。如果断点条件包含数值，就在地址条件和数据条件成立时发生用户断点的暂停。此时，为了指定字节数据，需要给断点数据寄存器（BDRA/BDRB）和断点数据屏蔽寄存器（BDMRA/BDMRB）的bit15 ~ 8、bit7 ~ 0的2个字节设定相同的数据。如果选择字或者字节的操作数长度，就忽视BDRA或者BDRB、BDMRA或者BDMRB的bit31 ~ 16。
4. 如果选择L总线，就在条件成立的指令执行后并且在执行下一条指令前发生用户断点的暂停。但是，在条件包含数值时，也可能在条件成立指令的下一条指令执行结束后暂停。如果选择I总线，就不能特定发生用户断点暂停的指令。另外，如果在延迟转移指令或者该延迟槽中发生此类用户断点，就不会在转移地址的第一条指令前发生用户断点的暂停。

7.4.4 顺序断点

1. 如果将BRCR的SEQ位置1，就在通道A断点条件成立后通道B断点条件成立时发生顺序断点的暂停。如果在通道A断点条件成立前通道B断点条件成立，就不发生用户断点的暂停。如果通道A和通道B的断点条件同时成立，也不发生顺序断点的暂停。在指定顺序断点时，如果想在通道A条件成立而通道B条件不成立时清除通道A成立标志，就必须给BRCR寄存器的SEQ位写0并将通道A的条件成立标志清0。
2. 在指定顺序断点时，能选择L总线或者I总线，也能指定执行次数的断点条件。例如：在指定执行次数断点条件的情况下，如果在通道A断点条件成立后并且BETR=H'0001时通道B断点条件成立，断点条件就成立。

7.4.5 被保存的程序计数器值

在发生用户断点的暂停时，将应该继续执行的指令地址压栈，并转移到异常处理状态。如果将 L 总线指定为断点条件，就能决定发生用户断点暂停的指令（断点条件包含数据的情况除外）；如果将 I 总线指定为断点条件，就不能决定发生用户断点暂停的指令。

1. 将取指令（执行指令前）指定为断点条件

将断点条件成立的指令地址压栈。不执行条件成立的指令而在执行前发生用户断点的暂停。但是，如果延迟槽指令条件成立，就将延迟转移指令的地址压栈。

2. 将取指令（执行指令后）指定为断点条件

将断点条件成立的指令的下一条指令地址压栈。执行条件成立的指令并且在下一条指令执行前发生用户断点的暂停。如果延迟转移指令或者延迟槽条件成立，就执行这些指令，并将转移目标地址压栈。

3. 将数据存取（只限地址）指定为断点条件

将断点条件成立指令的下一条指令地址压栈。执行条件成立的指令，并且在下一条指令执行前发生用户断点的暂停。如果延迟槽条件成立，就将转移目标地址压栈。

4. 将数据存取（地址+数据）指定为断点条件

如果给断点条件追加数值，就将断点条件成立指令的下一条指令地址或者下下一条指令地址压栈。无法准确地确定用户断点暂停的位置。

如果延迟槽指令条件成立，就将转移目标地址压栈。如果条件成立的指令的下一条指令为转移指令，就有可能在转移指令或者延迟槽指令执行结束后发生用户断点的暂停。此时，也将转移目标地址压栈。

7.4.6 PC 跟踪

1. 通过将 BRCR 的 PCTE 置 1，允许 PC 跟踪。如果发生转移（转移指令和中断异常），就将转移源地址和转移目标地址分别保存到 BRSR 和 BRDR。
2. 按照转移的种类，保存到 BRSR 和 BRDR 的值分别如下：
 - 当因转移指令而发生转移时，将转移指令的地址和转移目标指令的地址分别保存到 BRSR 和 BRDR。
 - 当因中断或者一般异常而发生转移时，将因异常而被保存的堆栈值和异常处理程序的起始地址分别保存到 BRSR 和 BRDR。
3. BRSR 和 BRDR 为 4 组队列结构。在读被保存在 PC 跟踪寄存器的地址时，先读队列的第一项。BRSR 和 BRDR 共享读指针，按照 BRSR、BRDR 的顺序读取，队列只在读 BRDR 后进行移位。如果 BRCR 的 PCTE 位从 OFF 切换到 ON，队列的值就无效。
4. 因为 4 组队列和 AUD 共享，所以必须在将 STBCR5 的 MSTP25 位置 0、STBCR6 的 AUDSRST 位置 1 后，将 BRCR 的 PCTE 位置 1。虽然本产品没有 AUD 功能，但是也需要进行此设定。
5. 在上电复位、手动复位和 AUD 软件复位时，对 FIFO（4 组队列）进行初始化。在通过手动复位或者 AUD 软件复位进行初始化后，一旦将 BRCR 的 PCTE 置 0 后再置 1，就能重新开始 PC 跟踪。

7.4.7 使用例子

(1) 指定为 L 总线取指令周期的断点条件

(例 1-1)

- 寄存器指定

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

指定条件: 通道 A/通道 B 的独立模式

<通道 A>

地址: H'00000404, 地址屏蔽: H'00000000

数据: H'00000000, 数据屏蔽: H'00000000

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

<通道 B>

地址: H'00008010, 地址屏蔽: H'00000006

数据: H'00000000, 数据屏蔽: H'00000000

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

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

(例 1-2)

- 寄存器指定

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

指定条件: 通道 A/通道 B 的顺序模式

<通道 A>

地址: H'00037226, 地址屏蔽: H'00000000

数据: H'00000000, 数据屏蔽: H'00000000

总线周期: L 总线/取指令 (执行指令前) /读/字

<通道 B>

地址: H'0003722E, 地址屏蔽: H'00000000

数据: H'00000000, 数据屏蔽: H'00000000

总线周期: L 总线/取指令 (执行指令前) /读/字

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

(例 1-3)

- 寄存器指定

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

指定条件: 通道 A/通道 B 的独立模式

<通道 A>

地址: H'00027128, 地址屏蔽: H'00000000

数据: H'00000000, 数据屏蔽: H'00000000

总线周期: L 总线/取指令 (执行指令前) /写/字

<通道 B>

地址: H'00031415, 地址屏蔽: H'00000000

数据: H'00000000, 数据屏蔽: H'00000000

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

因为通道 A 的取指令不是写周期, 所以不发生用户断点的暂停; 因为通道 B 对偶数地址进行取指令, 所以也不发生用户断点的暂停。

(例 1-4)

• 寄存器指定

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

指定条件: 通道 A/通道 B 的顺序模式

<通道 A>

地址: H'00037226, 地址屏蔽: H'00000000

数据: H'00000000, 数据屏蔽: H'00000000

总线周期: L 总线/取指令 (执行指令前) /写/字

<通道 B>

地址: H'0003722E, 地址屏蔽: H'00000000

数据: H'00000000, 数据屏蔽: H'00000000

总线周期: L 总线/取指令 (执行指令前) /读/字

因为通道 A 的取指令不是写周期, 所以顺序条件不成立, 也不发生用户断点的暂停。

(例 1-5)

• 寄存器指定

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

指定条件: 通道 A/通道 B 的独立模式

<通道 A>

地址: H'00000500, 地址屏蔽: H'00000000

数据: H'00000000, 数据屏蔽: H'00000000

总线周期: L 总线/取指令 (执行指令前) /读/长字

允许执行次数断点 (5次)

<通道 B>

地址: H'00001000, 地址屏蔽: H'00000000

数据: H'00000000, 数据屏蔽: H'00000000

总线周期: L 总线/取指令 (执行指令前) /读/长字

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

(例 1-6)

• 寄存器指定

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

指定条件: 通道 A/通道 B 的独立模式

<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 (执行指令后) /读 (断点条件不包含操作数长度)

<通道 B>

地址: H'00008010, 地址屏蔽: H'00000006

数据: H'00000000, 数据屏蔽: H'00000000

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

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

(2) 指定为 L 总线数据存取周期的断点条件

(例 2-1)

- 寄存器指定

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

指定条件: 通道 A/通道 B 的独立模式

<通道 A>

地址: H'00123456, 地址屏蔽: H'00000000

数据: H'12345678, 数据屏蔽: H'FFFFFFFF

总线周期: L 总线/数据存取/读 (断点条件不包含操作数长度)

<通道 B>

地址: H'000ABCDE, 地址屏蔽: H'000000FF

数据: H'0000A512, 数据屏蔽: H'00000000

总线周期: L 总线/数据存取/写/字

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

(3) 指定为 I 总线数据存取周期的断点条件

(例 3-1)

- 寄存器指定

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、BR CR=H'00000080

指定条件: 通道 A/通道 B 的独立模式

<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 (断点条件不包含操作数长度)

<通道 B>

地址: H'00055555, 地址屏蔽: H'00000000

数据: H'00000078, 数据屏蔽: H'0000000F

总线周期: I 总线 (CPU 周期) / 数据存取 / 写 / 字节

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

7.5 使用时的注意事项

1. 通过 I 总线读写 UBC 寄存器。因此, 在执行 UBC 寄存器的改写指令后到实际反映该值的期间, 有可能不发生需要的用户断点。为了得知更改 UBC 寄存器的时序, 必须读取最后写的寄存器。此后的指令对新写的寄存器值有效。
2. UBC 不能用相同的通道监视 L 总线周期和 I 总线周期。
3. 有关顺序断点指定的注意事项如下:
如果设定顺序断点, 就在 A 通道条件成立后 B 通道条件成立时, 产生条件成立。因此, 即使设定通道 A 和通道 B 同时成立的总线周期, 也不发生用户断点的暂停。
4. 如果在同一条指令中发生用户断点异常和其他异常, 就按照“第 5 章 异常处理”的表 5.1 规定的优先级进行判断。如果发生更高优先级的异常, 就不发生用户断点的暂停。
 - 比其他任何异常优先接受指令执行前的断点。
 - 如果指令执行后的断点或者数据存取的断点与优先级更高的再执行型异常 (包括指令执行前的断点) 同时产生, 就接受再执行型异常处理, 并且不将条件成立标志置位 (但是, 有第 5 项的例外情况)。在通过异常处理解除再执行型的异常源并结束同一指令的重新执行时, 发生用户断点的暂停, 并且将标志置位。
 - 如果指令执行后的断点或者数据存取的断点与优先级更高的完成型异常 (TRAPA) 同时产生, 就不发生用户断点的暂停, 但是将条件成立标志置位。
5. 作为第 4 项的例外, 有以下注意事项:
对于因数据存取而发生 CPU 地址错误的指令, 如果指令执行后的断点或者数据存取的断点条件成立, 就优先于用户断点中断而发生 CPU 地址错误。此时, 将 UBC 的条件成立标志置位。
6. 延迟槽中发生用户断点的暂停时的注意事项:
如果对 RTE 指令的延迟槽指令设定指令执行前的断点, 就在 RTE 指令的转移目标执行前不发生用户断点的暂停。
7. 在 UBC 模块待机模式时, 不能使用用户断点功能, 也不能读写 UBC 寄存器。否则, 就不保证该值。
8. 对 SLEEP 指令和延迟槽为 SLEEP 指令的转移指令, 不能设定指令执行后的断点。另外, 在 SLEEP 指令和 SLEEP 指令的前 1~2 条指令, 不能设定数据存取的断点。
9. 在 DTC 运行中, UBC 在 I 总线上不能正确判断 CPU 的外部空间存取。如果通过上述条件在 I 总线上判断外部空间存取, 就必须选择全部的总线主控。此时, 不能判断特定总线主控的条件。但是, 如果能从数值推定总线主控, 就能通过包含数据的条件推定总线主控。

第 8 章 数据传送控制器 (DTC)

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

8.1 特点

- 能进行任意通道数的传送能进行链传送 (对一个启动源进行多个数据的传送)
只能在指定次数的数据传送后进行链传送 (当计数器=0时)
- 传送模式: 3种
可选择正常传送模式、重复传送模式或者块传送模式。
可选择传送源地址和传送目标地址的增减或者固定。
- 能用32位指定传送源和传送目标的地址, 并且能直接指定4G字节的地址空间
- 能将传送数据的长度设定为字节、字或者长字
- 能向CPU请求启动DTC的中断
能在一次数据传送结束后向CPU请求中断。
能在指定的数据传送结束后向CPU请求中断。
- 能指定传送信息的跳读
- 对选择固定的传送源地址或者传送目标地址, 执行回写省略功能
- 能设定模块停止模式
- 能设定短地址模式
- 能从5种时序中选择总线权的释放时序
- 能从2种优先级中选择DTC启动时的优先级

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

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

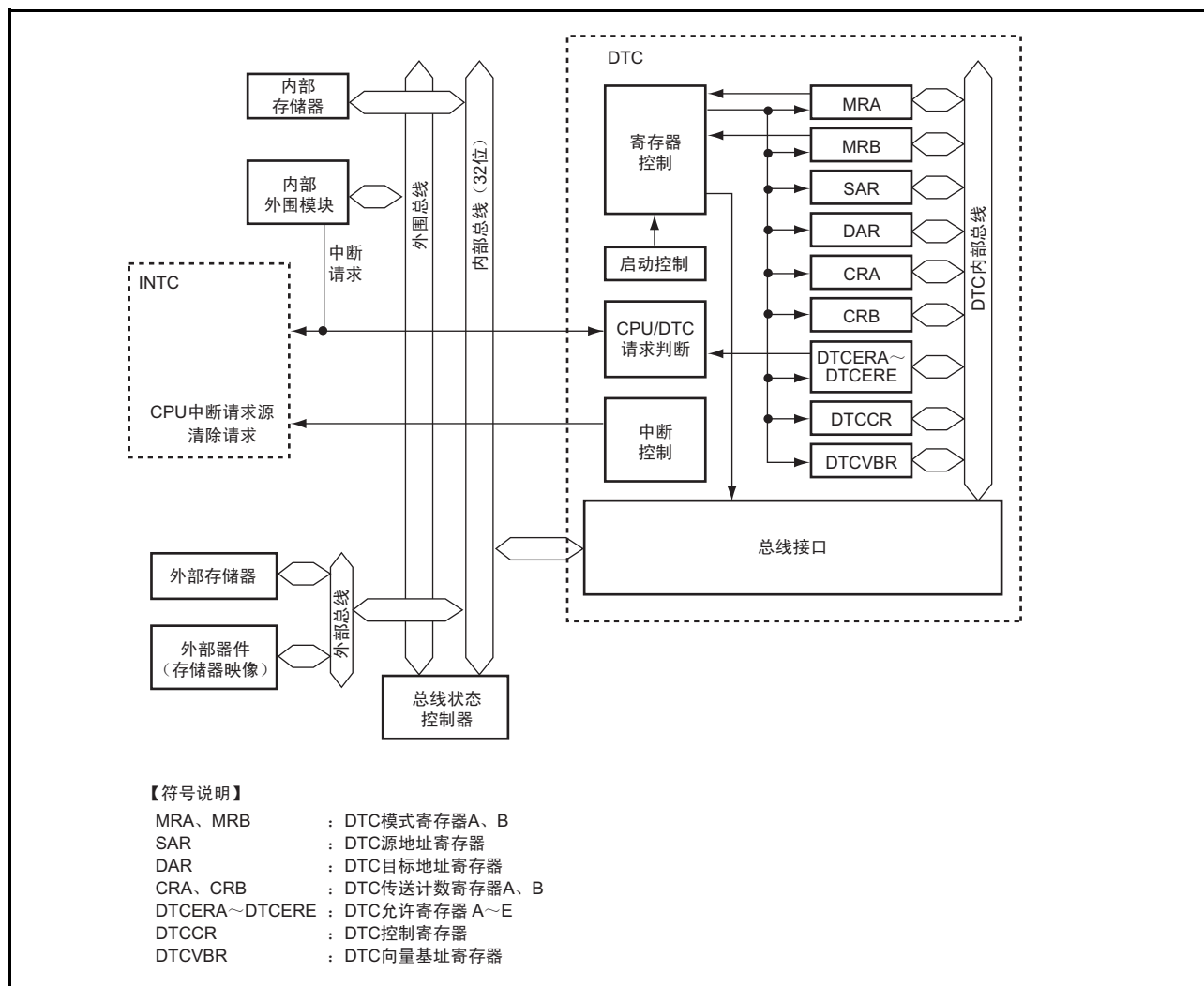


图 8.1 DTC 的框图

8.2 寄存器说明

DTC 有以下寄存器。有关这些寄存器的地址和各处理模式的寄存器状态, 请参照“第 25 章 寄存器一览”。

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

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

表 8.1 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
DTC 允许寄存器 A	DTCERA	R/W	H'0000	H'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 不能直接存取 MRA。

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

位	位名	初始值	R/W	说明
7、6	MD[1:0]	不定值	—	DTC 模式 1、0 指定 DTC 的传送模式。 00: 正常传送模式 01: 重复传送模式 10: 块传送模式 11: 禁止设定
5、4	Sz[1:0]	不定值	—	DTC 数据的传送长度 1、0 指定传送数据的长度。 00: 字节传送 01: 字传送 10: 长字传送 11: 禁止设定

位	位名	初始值	R/W	说明
3、2	SM[1:0]	不定值	—	源地址模式 1、0 指定数据传送后的 SAR 运行。 0x: SAR 固定 (不进行 SAR 的回写。) 10: 传送后 SAR 递增 (Sz1、Sz0 为 B'00 时 SAR+1, 为 B'01 时 SAR+2, 为 B'10 时 SAR+4) 11: 传送后 SAR 递减 (Sz1、Sz0 为 B'00 时 SAR-1, 为 B'01 时 SAR-2, 为 B'10 时 SAR-4)
1、0	—	不定值	—	保留位 只能写 0。

【注】 x: Don't care

8.2.2 DTC 模式寄存器 B (MRB)

MRB 选择 DTC 的运行模式。CPU 不能直接存取 MRB。

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

位	位名	初始值	R/W	说明
7	CHNE	不定值	—	DTC 链传送的允许 指定链传送。链传送的详细内容请参照“8.5.6 链传送”。通过 CHNS 位选择链传送的条件。 0: 禁止链传送 1: 允许链传送
6	CHNS	不定值	—	DTC 链传送的选择 选择链传送的条件。如果下一次传送是链传送, 就不进行指定传送次数的结束判断, 并且不清除启动源标志或者 DTCER。 0: 连续进行链传送 1: 只在传送计数器 =0 时进行链传送
5	DISEL	不定值	—	DTC 中断的选择 当此位是 1 时, 每进行 1 次数据传送或者 1 次块数据传送就向 CPU 请求中断; 当此位是 0 时, 只在指定次数的数据传送结束时向 CPU 请求中断。
4	DTS	不定值	—	DTC 传送模式的选择 在重复传送模式或者块传送模式中, 将源或者目标指定为重复区或者块区域。 0: 目标为重复区域或者块区域 1: 源为重复区域或者块区域

位	位名	初始值	R/W	说明
3、2	DM[1:0]	不定值	—	目标地址模式 1、0 指定数据传送后的 DAR 运行。 0x: DAR 固定 (不进行 DAR 的回写。) 10: 传送后 DAR 递增 (Sz1、Sz0 为 B'00 时 DAR+1, 为 B'01 时 DAR+2, 为 B'10 时 DAR+4) 11: 传送后 DAR 递减 (Sz1、Sz0 为 B'00 时 DAR-1, 为 B'01 时 DAR-2, 为 B'10 时 DAR-4)
1、0	—	不定值	—	保留位 只能写 0。

【注】 x: Don't care

8.2.3 DTC 源地址寄存器 (SAR)

SAR 是 32 位寄存器, 指定 DTC 数据传送的传送源地址。

CPU 不能直接存取 SAR。

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

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

8.2.4 DTC 目标地址寄存器 (DAR)

DAR 是 32 位寄存器, 指定 DTC 数据传送的传送目标地址。

CPU 不能直接存取 DAR。

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

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

8.2.5 DTC 传送计数寄存器 A (CRA)

CRA 是 16 位寄存器，指定 DTC 数据传送的传送次数。

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

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

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

CPU 不能直接存取 CRA。

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

8.2.6 DTC 传送计数寄存器 B (CRB)

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

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

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

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

DTCE_R 选择启动 DTC 的中断源，有 DTCERA ~ DTCERE。有关各中断源和 DTCE 位的对应关系，请参照表 8.2。

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

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

8.2.8 DTC 控制寄存器 (DTCCR)

DTCCR 设定传送信息的跳读。

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

【注】* 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说明
7 ~ 5	—	全 0	R	保留位 读写值总是 0。
4	RRS	0	R/W	DTC 传送信息跳读的允许 控制向量地址和传送信息的读取。DTC 向量号随时和上次启动的向量号进行比较，如果向量号的值相同并且此位为 1，就不读向量地址和传送信息而进行 DTC 的数据传送。如果上次的启动是链传送，就必须读向量地址和传送信息。 但是，如果将总线功能扩展寄存器 (BSCEHR) 的 DTPR 位置 1，就与此位的设定无关，不跳读传送信息。 0: 不跳读传送信息 1: 在向量号的值相同时跳读传送信息

位	位名	初始值	R/W	说明
3	RCHNE	0	R/W	DTC 重复传送后的链传送允许 允许或者禁止重复传送时的链传送（当传送计数器 =0 时）。如果在进行重复传送时传送计数器（CRAL）为 0，因为 CRAH 指定的值被回写到 CRAL，所以在传送计数器 =0 时不发生链传送。通过将此位置 1，允许在回写传送计数器时进行链传送。 0：禁止重复传送后的链传送 1：允许重复传送后的链传送
2、1	—	全 0	R	保留位 这些位可读不可写。
0	ERR	0	R/(W)*	传送停止标志 表示发生 DTC 地址错误或者 NMI 中断请求。如果在启动 DTC 时发生 DTC 地址错误或者 NMI 中断请求，就在释放 DTC 的总线权后进行 DTC 地址错误或者 NMI 中断的处理。在数据传送后，DTC 以传送信息的写状态停止。 0：没有中断请求 1：发生中断请求 [清除条件] • 当读到 1 的状态后写 0 时

8.2.9 DTC 向量基址寄存器 (DTCVBR)

DTCVBR 是 32 位寄存器，设定用于计算向量表地址的基址。

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

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

位	位名	初始值	R/W	说明
31 ~ 12		全 0	R/W	bit11 ~ 0 的读写值总是 0。
11 ~ 0	—	全 0	R	

8.2.10 总线功能扩展寄存器 (BSCEHR)

BSCEHR 是 16 位寄存器，设定 DTC 总线权的释放时序等。在优先进行 DTC 传送以及减少 DTC 启动周期数时，BSCEHR 能设定有效功能。详细内容请参照“9.4.4 总线功能扩展寄存器 (BSCEHR)”。

8.3 启动源

通过中断请求启动 DTC。由 DTCER 选择要启动 DTC 的中断源，当 DTCER 的对应位被置 1 时，对应的中断源就为 DTC 的启动源；当对应位被清 0 时，对应的中断源就为 CPU 的中断源。在 1 次数据传送（在链传送时，连续的最后传送）结束时，清除启动源中断标志或者 DTCER 的对应位。

8.4 传送信息的分配和 DTC 向量表

将传送信息分配到数据区。传送信息的起始地址必须为地址 4n，否则就忽视低 2 位进行存取 ([1:0]=B'00)。数据区中的传送信息分配如图 8.2 所示。只有在全部 DTC 传送的传送源 / 传送目标都为内部 RAM 和内部外围模块时，才能通过将“9.4.4 总线功能扩展寄存器 (BSCEHR)”中的 DTSA 位置 1 进行短地址模式的选择。

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

DTC 按照启动源从向量表中读取传送信息的起始地址，然后从该起始地址读传送信息。DTC 向量表和传送信息的对应关系如图 8.3 所示。

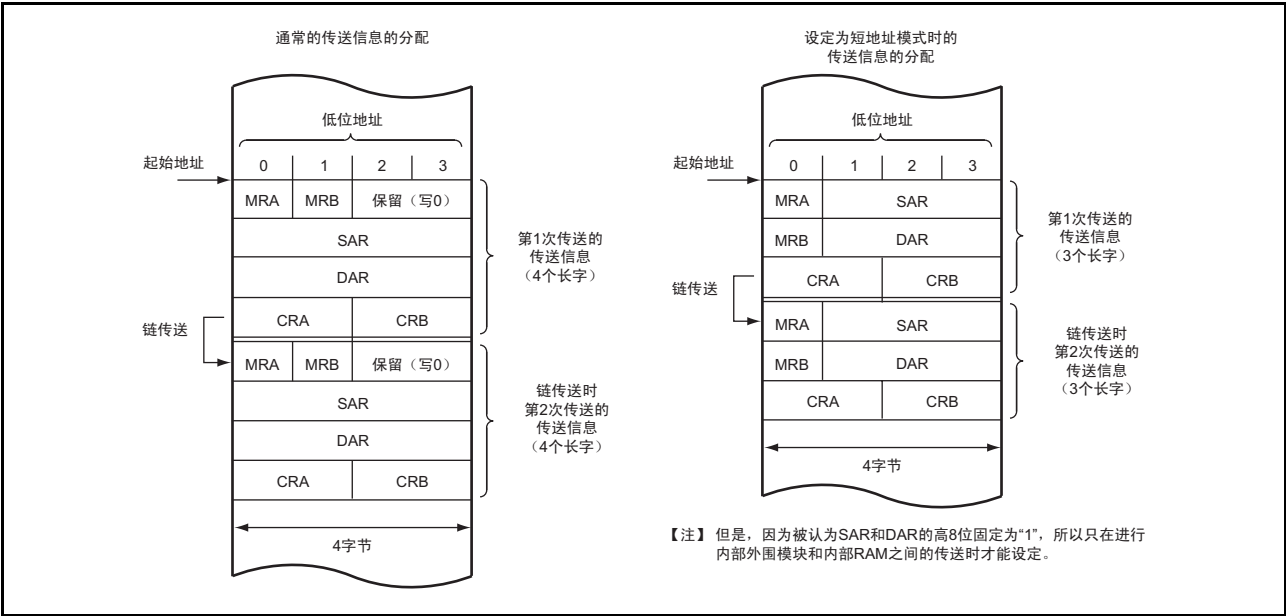


图 8.2 数据区中的传送信息的分配

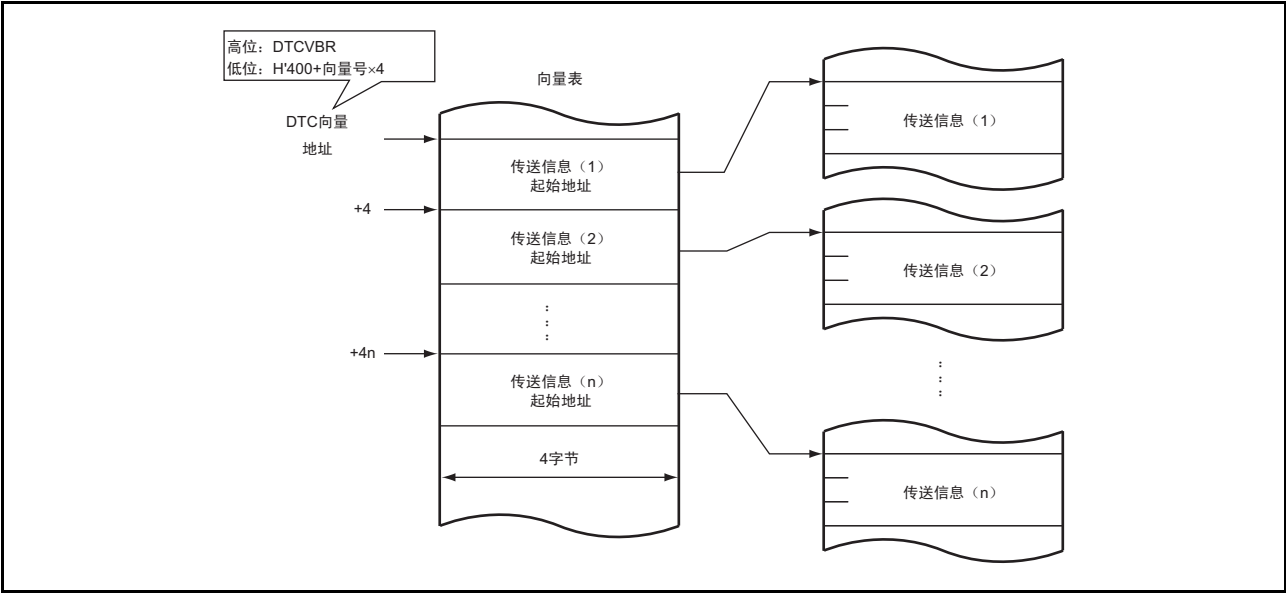


图 8.3 DTC 向量表和传送信息的对应关系

DTC 的启动源和向量地址的对应关系如表 8.2 所示。

表 8.2 中断源和 DTC 向量地址以及对应的 DTCE

启动源的发生源	启动源	向量号	DTC 向量地址的偏移量	DTCE*1	传送源	传送目标	优先级
外部引脚	IRQ0	64	H'500	DTCERA15	任意 *2	任意 *2	高
	IRQ1	65	H'504	DTCERA14	任意 *2	任意 *2	
	IRQ2	66	H'508	DTCERA13	任意 *2	任意 *2	
	IRQ3	67	H'50C	DTCERA12	任意 *2	任意 *2	
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	低
	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	

启动源的发生源	启动源	向量号	DTC 向量地址的偏移量	DTCE*1	传送源	传送目标	优先级
MTU2_5	TGIU_5	128	H'600	DTCERC14	任意 *2	任意 *2	高 ↑ ↓ 低
	TGIV_5	129	H'604	DTCERC13	任意 *2	任意 *2	
	TGIW_5	130	H'608	DTCERC12	任意 *2	任意 *2	
MTU2S_3	TGIA_3S	160	H'680	DTCERC3	任意 *2	任意 *2	
	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	
CMT_0	CMI_0	184	H'6E0	DTCERD7	任意 *2	任意 *2	
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	SSTD0 ~ SSTDR3	
I ² C2	IITXI	238	H'7B8	DTCERE5	任意 *2	ICDRT	
	IIRXI	239	H'7BC	DTCERE4	ICDRR	任意 *2	
RCAN-ET_0	RM0_0	242	H'7C8	DTCERE3	CONTROL0H ~ CONTROL1L*3	任意 *2	

【注】 *1 没有对应中断源的 DTCE 位是保留位，必须写 0。如果通过中断解除软件待机状态，就必须给对应的 DTCE 位写 0。

*2 外部存储器、存储器映像外部器件、内部存储器、内部外围模块（DTC、BSC、UBC 和 FLASH 除外）。但是，至少必须将传送源或者传送目标设定为内部外围模块。

不能在外部存储器、存储器映像外部器件以及内部存储器之间进行传送。

*3 必须使用块传送模式等，读到邮箱 0 的信息控制字段 1（CONTROL1）为止。

8.5 运行说明

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

DTC 用 SAR 指定传送源地址，用 DAR 指定传送目标地址。在传送后，SAR 和 DAR 分别被增减或者固定。DTC 的传送模式如表 8.3 所示。

表 8.3 DTC 的传送模式

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

- 【注】 *1 将源或者目标设定为重复区域。
 *2 将源或者目标设定为块区域。
 *3 在进行指定次数的传送后，恢复初始状态继续运行。
 *4 1 次表示 1 块。

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

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

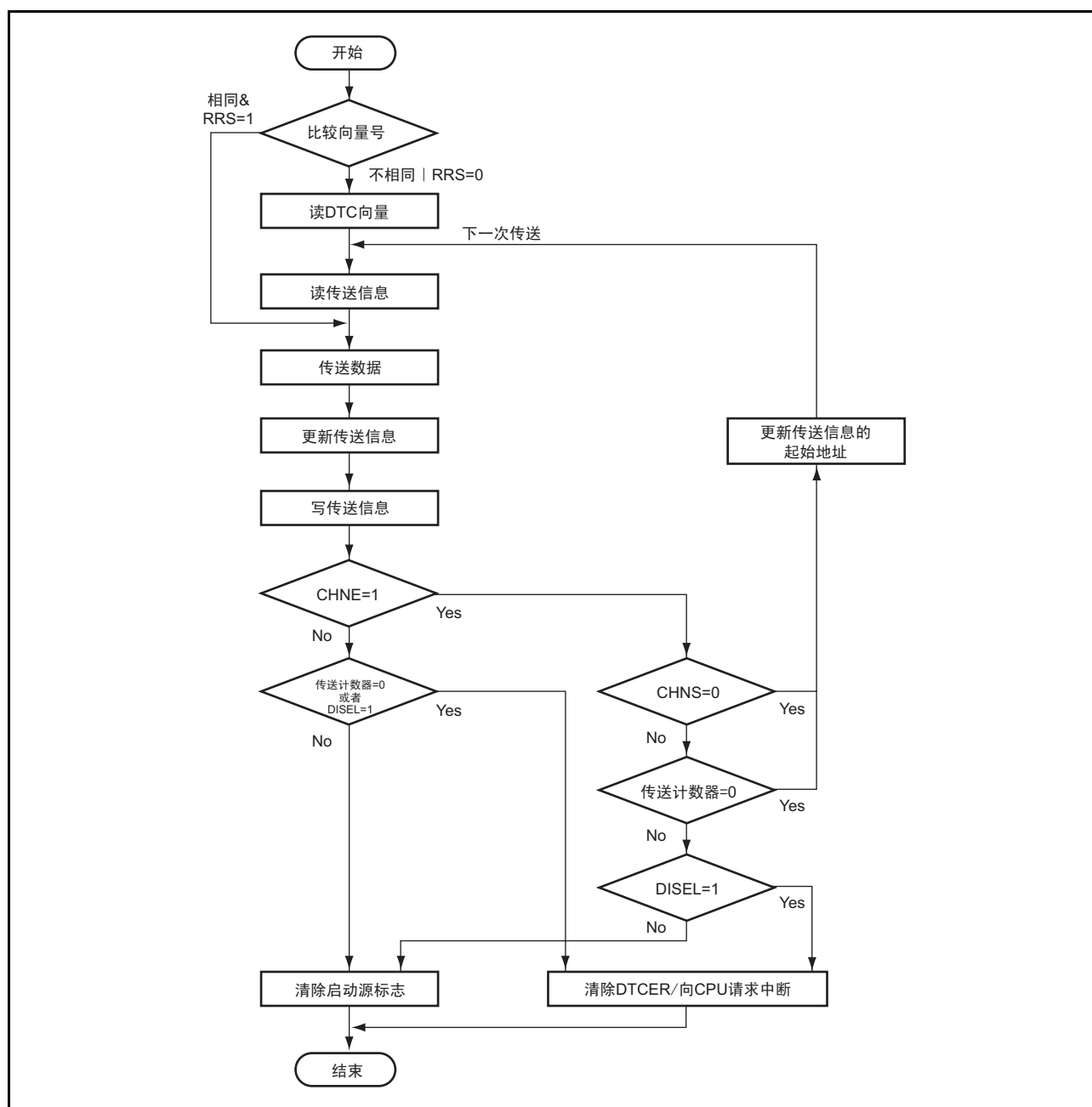


图 8.4 DTC 的运行流程图

表 8.4 DTC 的传送条件 (包括链传送)

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

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

*2 表示 CRAL 的内容被改写为 CRAH 的内容的情况。

8.5.1 传送信息的跳读功能

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

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

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

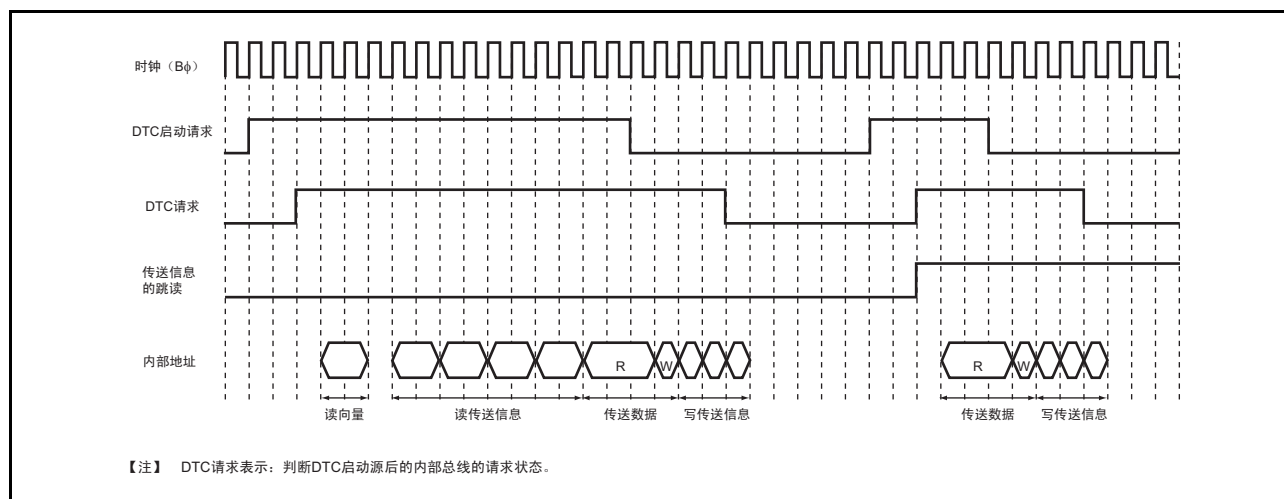


图 8.5 传送信息的跳读时序图

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

8.5.2 传送信息的回写省略功能

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

表 8.5 传送信息的回写省略条件和回写省略的寄存器

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

8.5.3 正常传送模式

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

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

表 8.6 正常传送模式的寄存器功能

寄存器	功能	写传送信息时的回写值
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】 * 不回写传送信息。

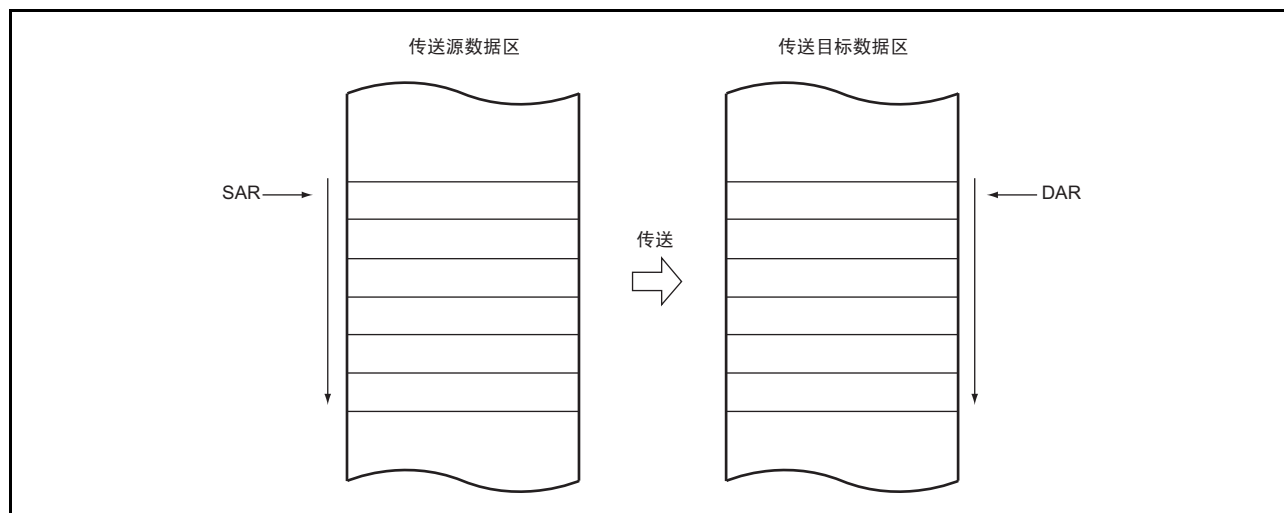


图 8.6 正常传送模式的存储器映像

8.5.4 重复传送模式

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

重复传送模式的寄存器功能和存储器映像分别如表 8.7 和图 8.7 所示。

表 8.7 重复传送模式的寄存器功能

寄存器	功能	写传送信息时的回写值	
		CRAL≠1 时	CRAL=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 A	CRAL-1	CRAH
CRB	传送计数 B	不更新	不更新

【注】 * 不回写传送信息。

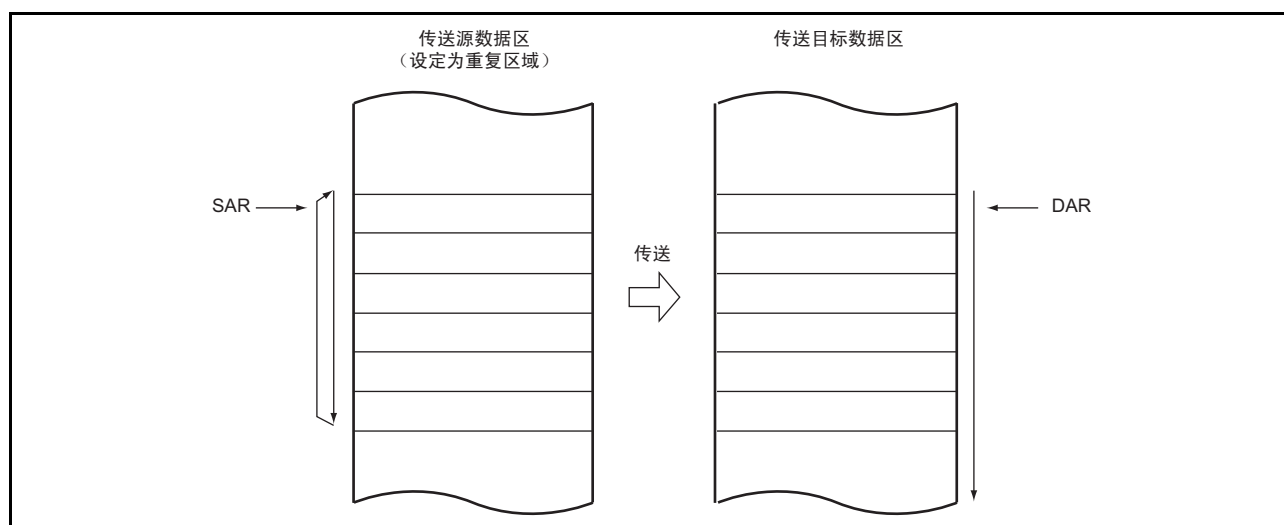


图 8.7 重复传送模式的存储器映像（将传送源指定为重复区域）

8.5.5 块传送模式

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

块传送模式的寄存器功能和存储器映像分别如表 8.8 和图 8.8 所示。

表 8.8 块传送模式的寄存器功能

寄存器	功能	写传送信息时的回写值
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	保存块大小	CRAH
CRAL	块大小的计数器	CRAH
CRB	块传送次数的计数器	CRB-1

【注】 * 不回写传送信息。

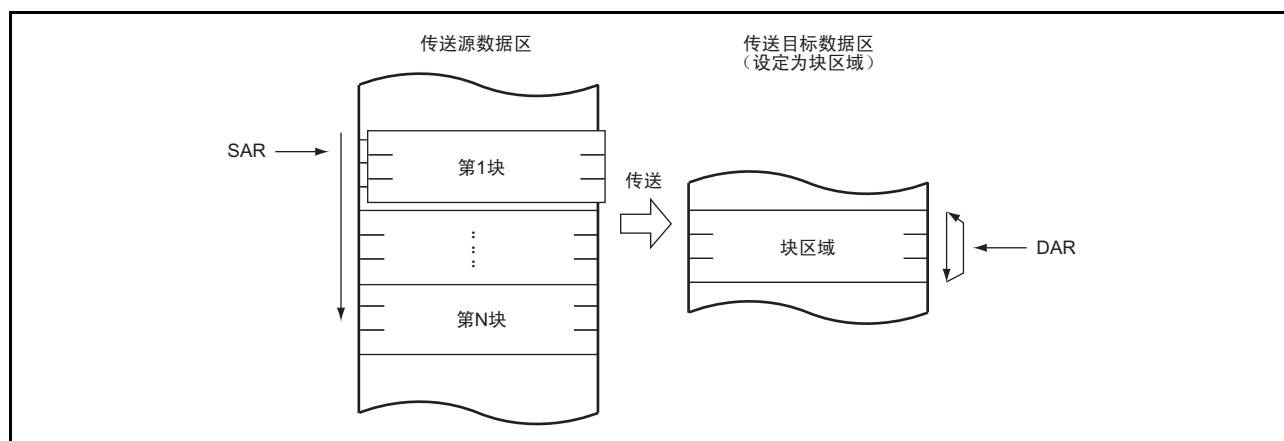


图 8.8 块传送模式的存储器映像（将传送目标指定为块区域）

8.5.6 链传送

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

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

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

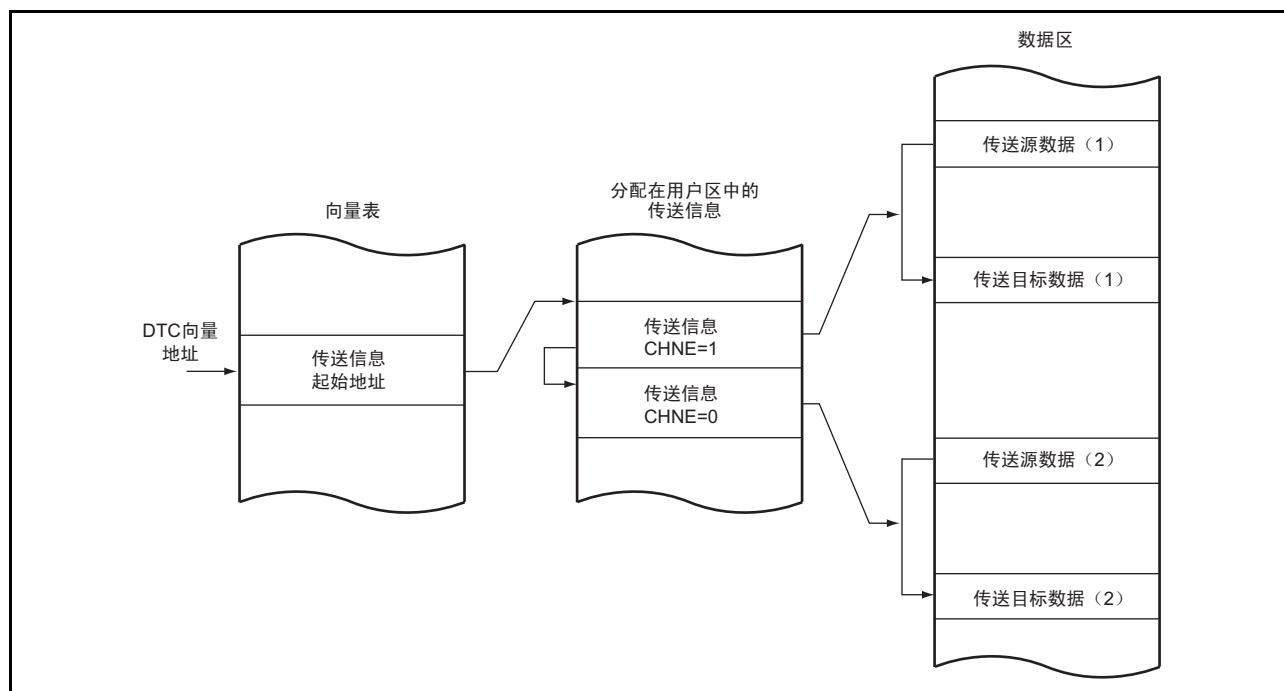


图 8.9 链传送的运行

8.5.7 运行时序

DTC 的运行时序如图 8.10 ~ 图 8.15 所示。

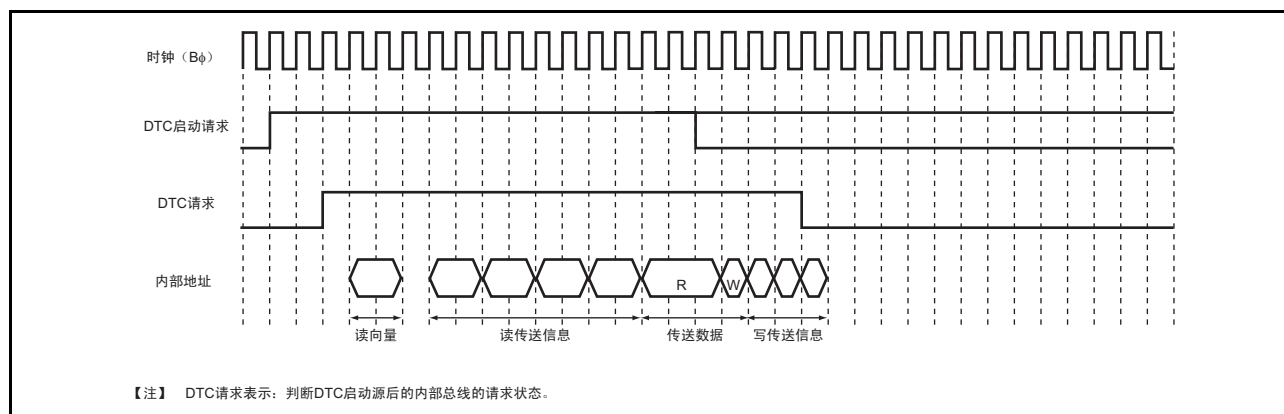


图 8.10 DTC 的运行时序例【正常传送或者重复传送】

(从内部外围模块启动，Iφ:Bφ:Pφ=1:1/2:1/2，
从内部外围模块将数据传送到内部 RAM，传送信息的写操作为 3 个状态)

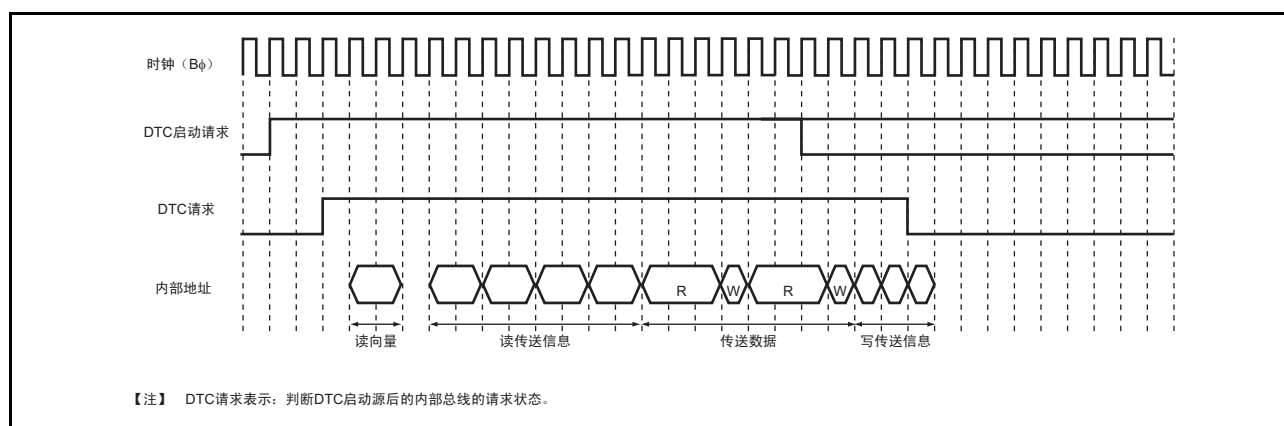


图 8.11 DTC 的运行时序例【块传送, 块大小 = 2】

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

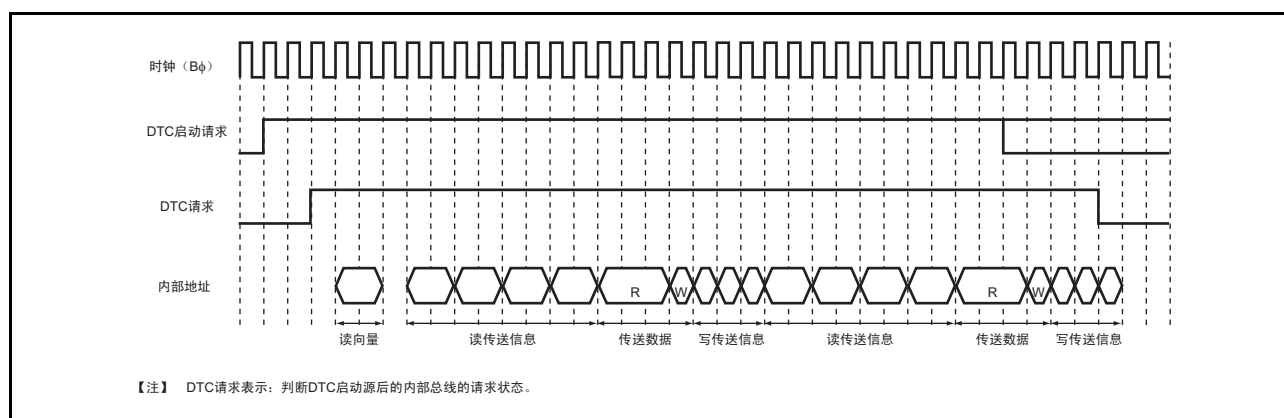


图 8.12 DTC 的运行时序例【链传送】

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

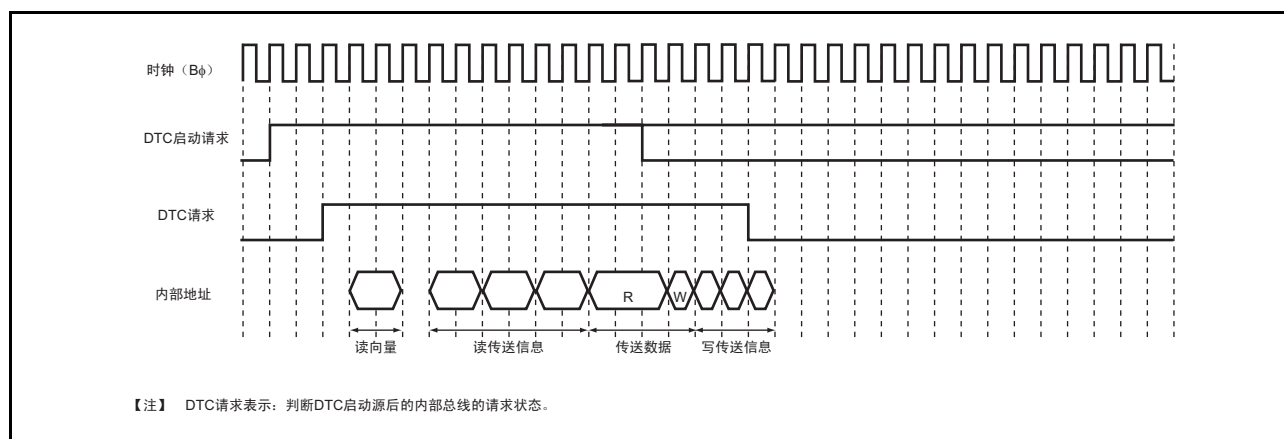


图 8.13 DTC 的运行时序例【短地址模式: 正常传送或者重复传送】

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

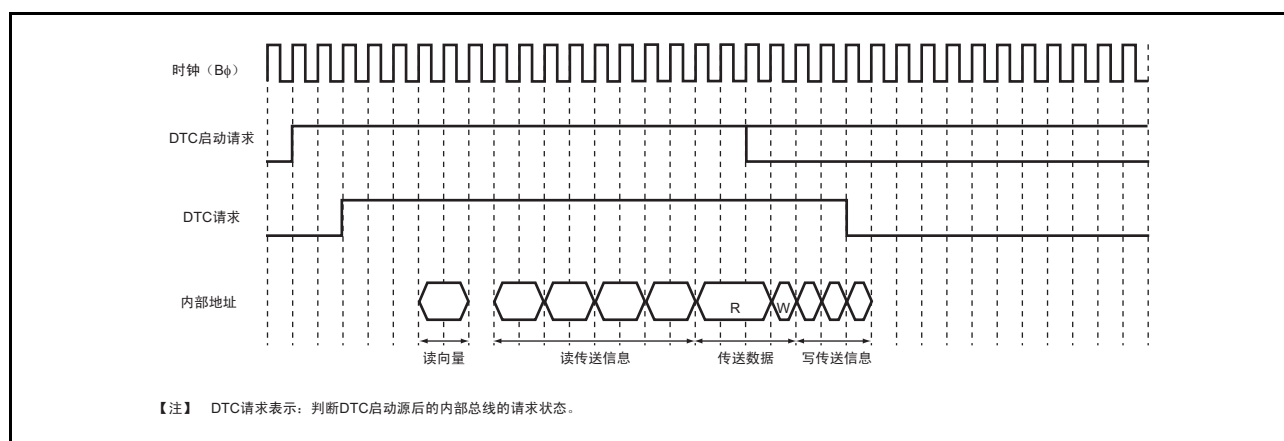


图 8.14 DTC 的运行时序例【正常传送或者重复传送，DTPR=1】
 (从内部外围模块启动， $I\phi:B\phi:P\phi=1:1/2:1/2$ ，
 从内部外围模块将数据传送到内部 RAM，传送信息的写操作为 3 个状态)

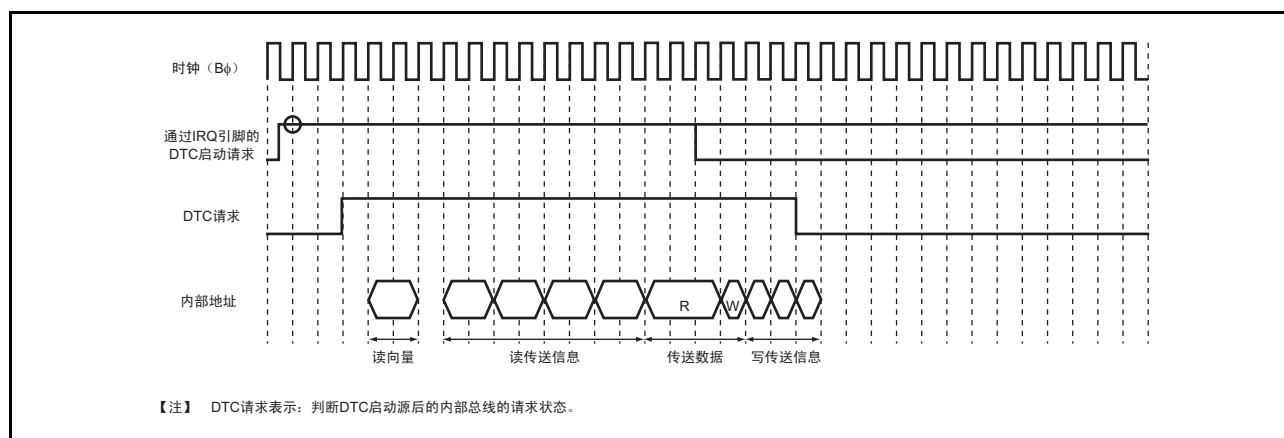


图 8.15 DTC 的运行时序例【正常传送或者重复传送】
 (从 IRQ 启动， $I\phi:B\phi:P\phi=1:1/2:1/2$ ，
 从内部外围模块将数据传送到内部 RAM，传送信息的写操作为 3 个状态)

8.5.8 DTC 的执行状态

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

表 8.9 DTC 的执行状态

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

【注】 *1 在跳读传送信息时

*2 在 SAR 或者 DAR 为固定模式时

*3 在 SAR 和 DAR 都为固定模式时

*4 在短地址模式中

P: 块大小 (CRAH和CRAL的初始设定)

表 8.10 执行状态所需的状态数

存取对象		内部 RAM*1/ROM*2	内部 I/O 寄存器	外部器件 *4	
总线宽度		32 位	16 位	8 位	16 位
存取状态		$1B\phi \sim 3B\phi^{*1*2}$	$2P\phi$	$2B\phi$	$2B\phi$
执行 状态	读向量 S_I	$1B\phi \sim 3B\phi^{*1*2}$	—	$9B\phi$	$5B\phi$
	读传送信息 S_J	$1B\phi \sim 3B\phi^{*1}$	—	$9B\phi$	$5B\phi$
	写传送信息 S_K	$1B\phi \sim 3B\phi^{*1}$	—	$2B\phi^{*5}$	$2B\phi^{*5}$
	读字节数据 S_L	$1B\phi \sim 3B\phi^{*1}$	$1B\phi+2P\phi^{*3}$	$3B\phi$	$3B\phi$
	读字数据 S_L	$1B\phi \sim 3B\phi^{*1}$	$1B\phi+2P\phi^{*3}$	$5B\phi$	$3B\phi$
	读长字数据 S_L	$1B\phi \sim 3B\phi^{*1}$	$1B\phi+4P\phi^{*3}$	$9B\phi$	$5B\phi$
	写字节数据 S_M	$1B\phi \sim 3B\phi^{*1}$	$1B\phi+2P\phi^{*3}$	$2B\phi^{*5}$	$2B\phi^{*5}$
	写字数据 S_M	$1B\phi \sim 3B\phi^{*1}$	$1B\phi+2P\phi^{*3}$	$2B\phi^{*5}$	$2B\phi^{*5}$
	写长字数据 S_M	$1B\phi \sim 3B\phi^{*1}$	$1B\phi+4P\phi^{*3}$	$2B\phi^{*5}$	$2B\phi^{*5}$
	内部运行 S_N	1			

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

	读	写
当 $l\phi:B\phi=1:1$ 时	$3B\phi$	$3B\phi$
当 $l\phi:B\phi=1:1/2$ 时	$2B\phi$	$1B\phi$
当 $l\phi:B\phi=1:1/3$ 时	$2B\phi$	$1B\phi$
当 $l\phi:B\phi \leq 1:1/4$ 时	$1B\phi$	$1B\phi$

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

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

*4 因 BSC 寄存器的设定而不同。表中的值是等待 =0 并且 CSnWCR 的 WM 位 =1 的例子。

*5 因总线的状态而不同。

如果正在使用外部总线或者因经常发生写操作（如块传送）而需要长时间的外部等待，就不能有效利用写缓冲区而增加周期数。

有关写缓冲区的详细内容请参照“9.5.7(2) LSI 内部总线主控的存取”。

用以下计算式计算执行状态数：

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

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

8.5.9 DTC 的总线权释放时序

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

能通过总线功能扩展寄存器（BSCEHR）设定总线权的释放时序，详细内容请参照“9.4.4 总线功能扩展寄存器（BSCEHR）”。根据设定，不同的总线权释放时序如表 8.11 所示。但是，不能进行设定 1～5 以外的设定，并且不能在 DTC 启动时更改设定。

时序图如图 8.16 所示。

表 8.11 DTC 的总线权释放时序

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

【注】 *1 只在读向量后 CPU 请求外部存取时，释放总线权。

*2 设定 4 有以下限制：

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

*3 Don't care

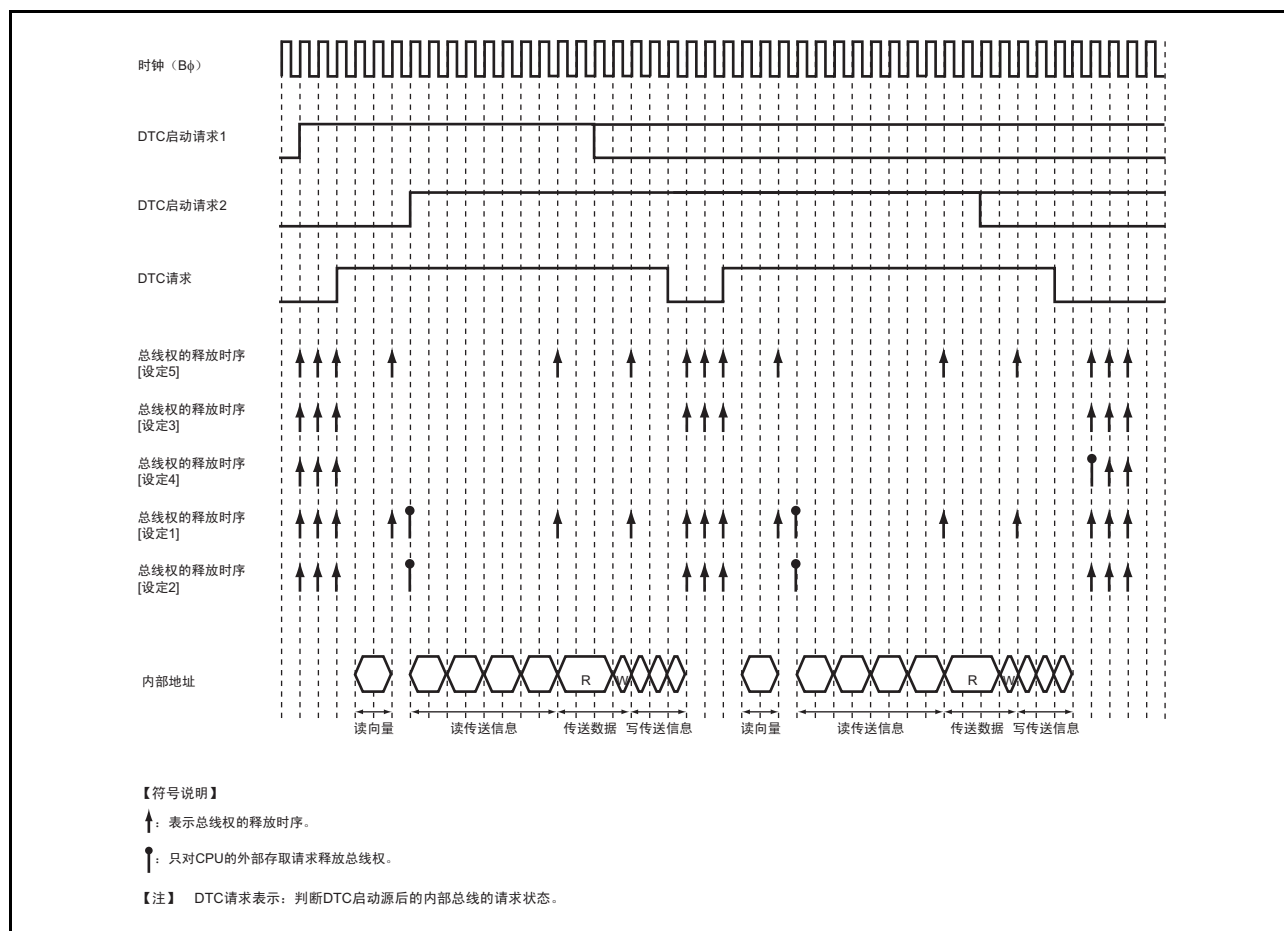


图 8.16 DTC 的运行时序例【正常传送中 2 个源竞争时】
(从内部外围模块启动, $l\phi:B\phi:P\phi=1:1/2:1/2$,
从内部外围模块将数据传送到内部 RAM, 传送信息的写操作为 3 个状态)

8.5.10 DTC 启动优先级的设定

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

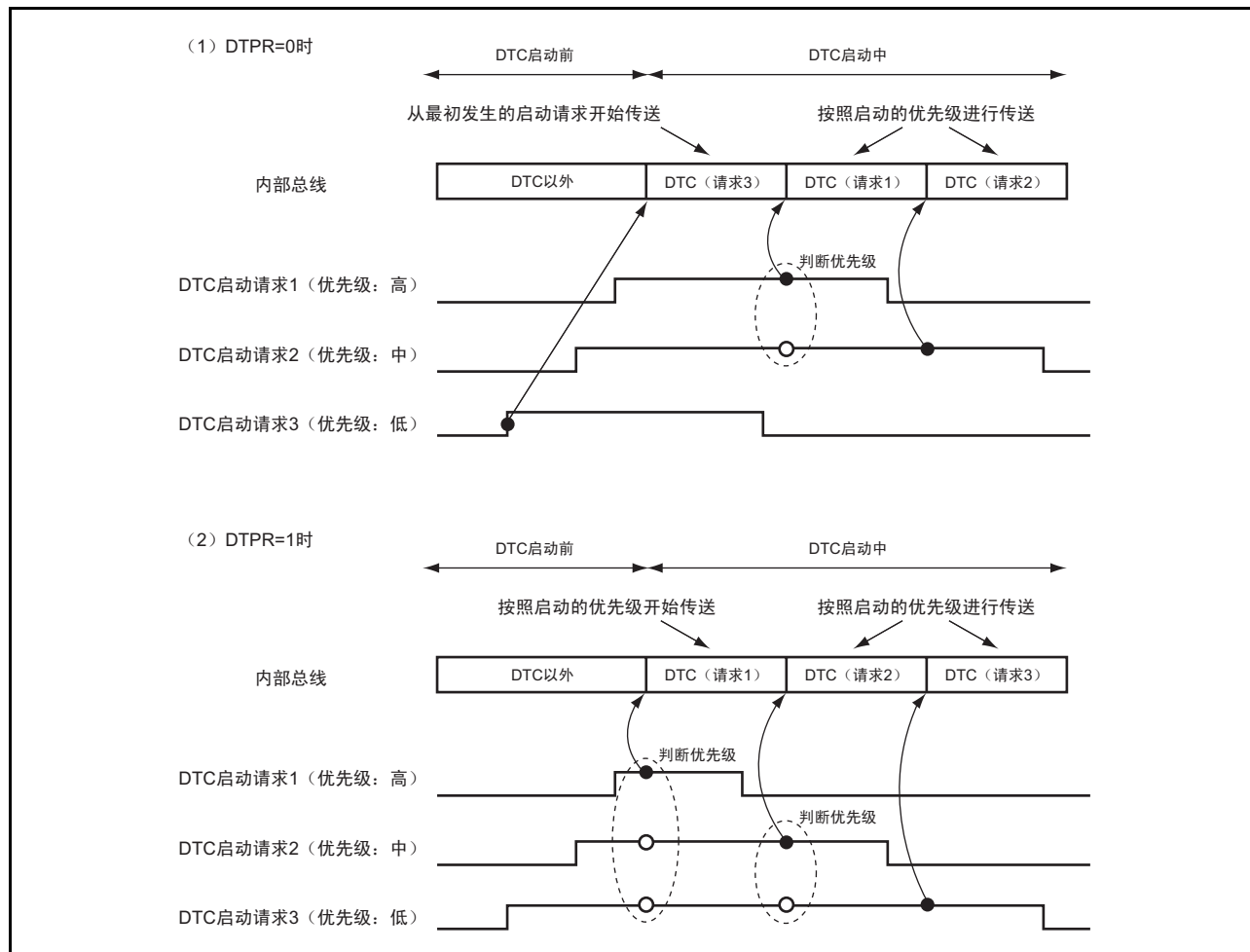


图 8.17 DTC 启动优先级的运行例子

8.6 通过中断启动 DTC

通过中断启动 DTC 的步骤如图 8.18 所示。

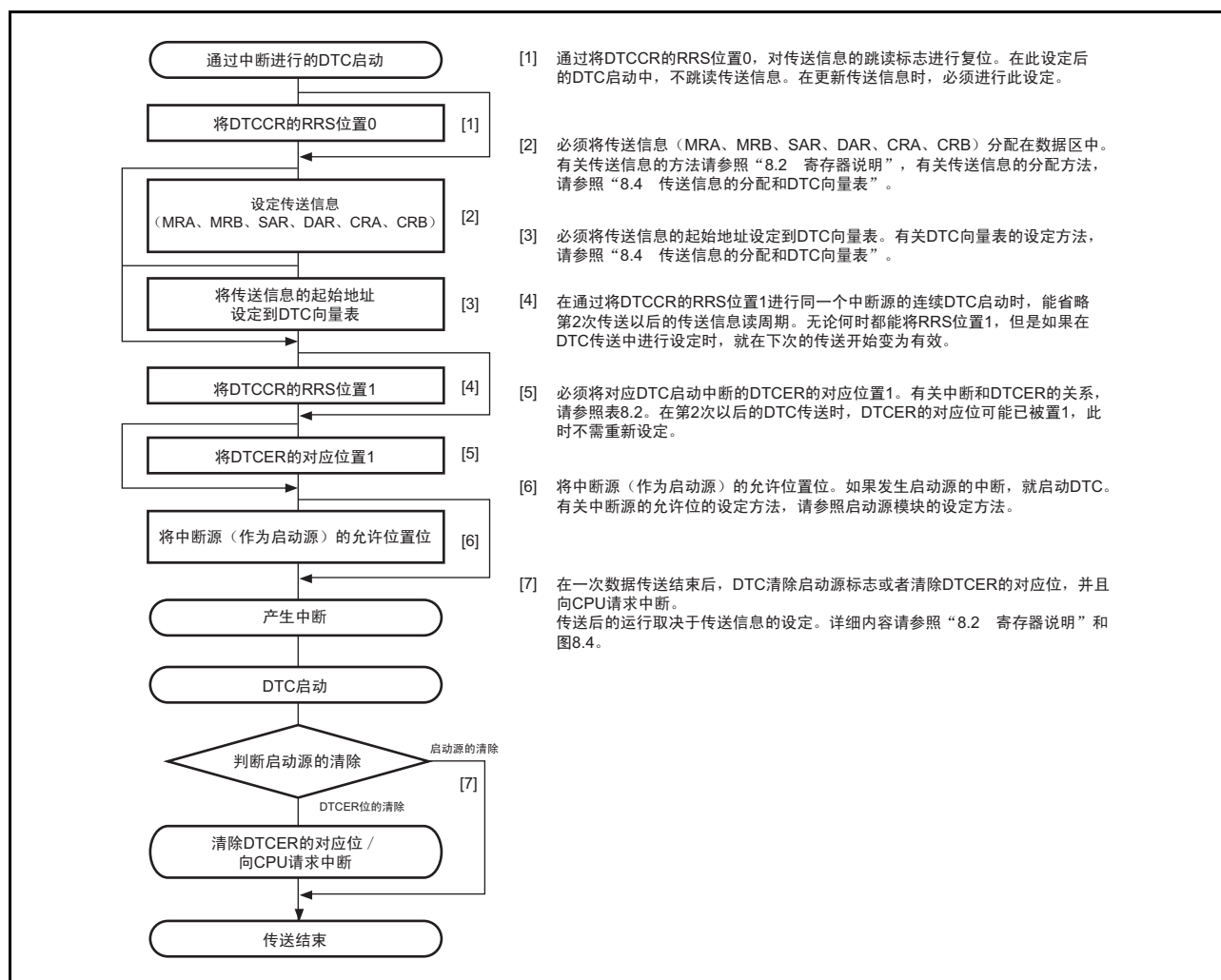


图 8.18 通过中断启动 DTC 的方法

8.7 DTC 使用例

8.7.1 正常传送

以通过 SCI 进行 128 个字节的数据接收为例说明 DTC 的使用例：

1. 通过 MRA 设定源地址固定 (SM1=SM0=0)、目标地址递增 (DM1=1、DM0=0)、正常传送模式 (MD1=MD0=0) 和字节长度 (Sz1=Sz0=0)，DTS 位可为任意值；通过 MRB 设定由 1 次中断进行 1 次数据传送 (CHNE=0、DISEL=0)；通过 SAR 设定 SCI 的 RDR 地址，DAR 设定保存数据的 RAM 起始地址，CRA 设定 128 (H'0080)，CRB 可为任意值。
2. 将用于 RXI 中断的传送信息的起始地址设定到 DTC 向量表。
3. 将 DTCER 的对应位置 1。
4. 将 SCI 设定为规定的接收模式。将 SCR 的 RIE 位置 1，允许接收结束 (RXI) 中断。如果在 SCI 的接收运行时发生接收错误，就不进行以后的接收，因此必须设定为 CPU 能响应接收错误中断。
5. 每当 SCI 的 1 个字节数据接收结束时，SSR 的 RDRF 标志就被置 1，产生 RXI 中断，启动 DTC。通过 DTC 将接收数据从 RDR 传送到 RAM，进行 DAR 的递增和 CRA 的递减。RDRF 标志自动清 0。
6. 如果在 128 次的数据传送结束后 CRA 变为 0，RDRF 标志就保持 1，而 DTCE 位被清 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 的链传送

只有在计数器变为 0 时进行第 2 数据的传送并且重新设定第 1 数据的传送，才能进行传送次数 ≥ 256 次的重复传送。

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

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

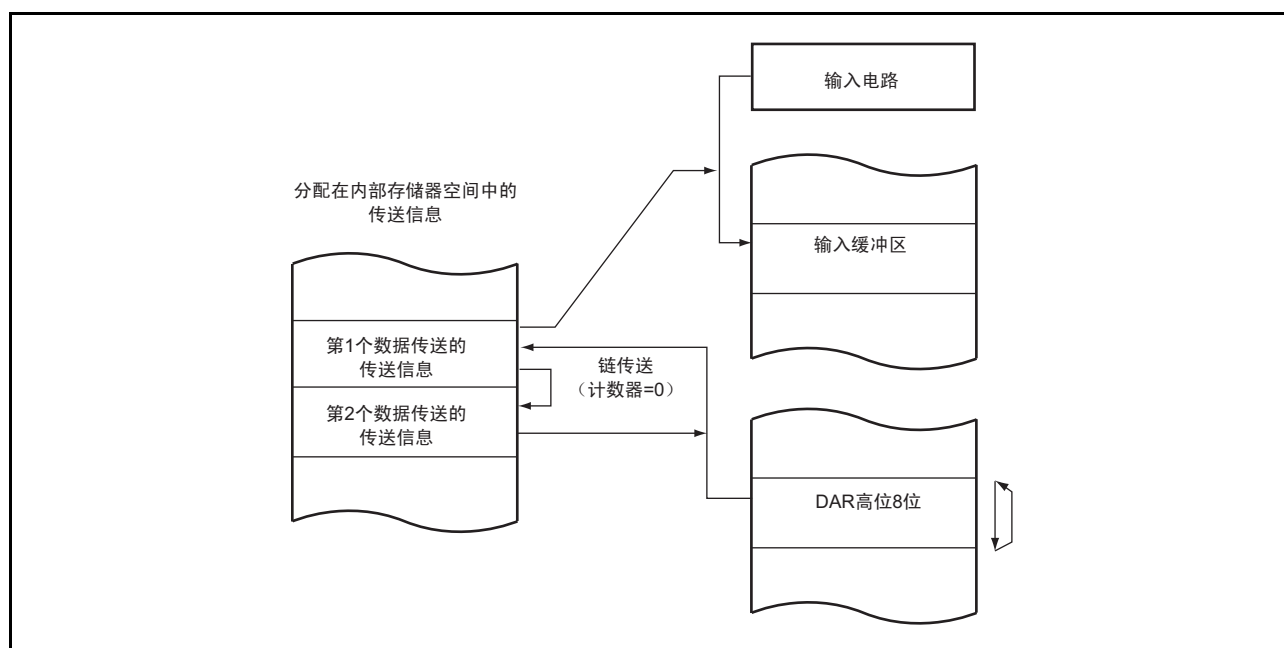


图 8.19 计数器 =0 的链传送

8.8 中断源

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

8.9 使用时的注意事项

8.9.1 模块待机模式的设定

能通过待机控制寄存器设定禁止或者允许 DTC 的运行，初始值为禁止 DTC 的运行。在设定为模块待机模式时，禁止 DTC 的运行，但是不禁止寄存器的存取。不能在启动 DTC 时设定模块待机模式。如果要转换到软件待机模式或者模块待机模式，就必须清除全部 DTCE 寄存器。详细内容请参照“第 24 章 低功耗模式”。

8.9.2 内部 RAM

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

8.9.3 DTCE 位的设定

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

8.9.4 链传送

在进行链传送时，在传送最后连接的数据时清除启动源或者 DTCE。另一方面，在读写规定的寄存器时，SCI 和 A/D 转换器的中断 / 启动源被清除。

8.9.5 传送信息的起始地址 / 源地址 / 目标地址

必须将指向向量表的传送信息的起始地址指定为地址 4n。

另外, 必须将传送信息分配到内部 RAM 或者外部存储空间。

8.9.6 通过 DTC 存取 DTC 寄存器

不能使用 DTC 存取 DTC 的寄存器。

8.9.7 IRQ 中断作为 DTC 传送源时的注意事项

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

8.9.8 SCI 作为 DTC 启动源时的注意事项

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

8.9.9 中断源标志的清除

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

8.9.10 NMI 中断和 DTC 启动的竞争

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

另外, 在判断由 NMI 引起的 DTC 停止时, 需要 $1 \times Bcyc + 3 \times Pcyc$ 周期; 在判断由 IRQ 引起的 DTC 启动时, 需要 $2 \times Bcyc$ 周期; 在判断由外围模块引起的 DTC 启动时, 需要 $1 \times Pcyc$ 周期。

8.9.11 中途取消 DTC 启动请求时的运行

DTC 在接受启动请求后到回写结束前的一系列 DTC 处理期间, 不接受下一个启动请求。

第 9 章 总线状态控制器 (BSC)

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

9.1 特点

1. 外部地址空间

- CS0 和 CS1 各空间最大支持 1M 字节
- 每个空间的数据总线宽度固定为 8 位
- 能控制每个空间的等待状态的插入
- 能控制每次读写存取的等待状态插入
- 在连续存取为读-写 (同一空间或者不同空间)、读-读 (同一空间或者不同空间) 或者起始周期为写周期的 5 种情况下, 能独立设定空闲周期。

2. 通常空间接口

- 支持能直接连接 SRAM 的接口

BSC 的框图如图 9.1 所示。

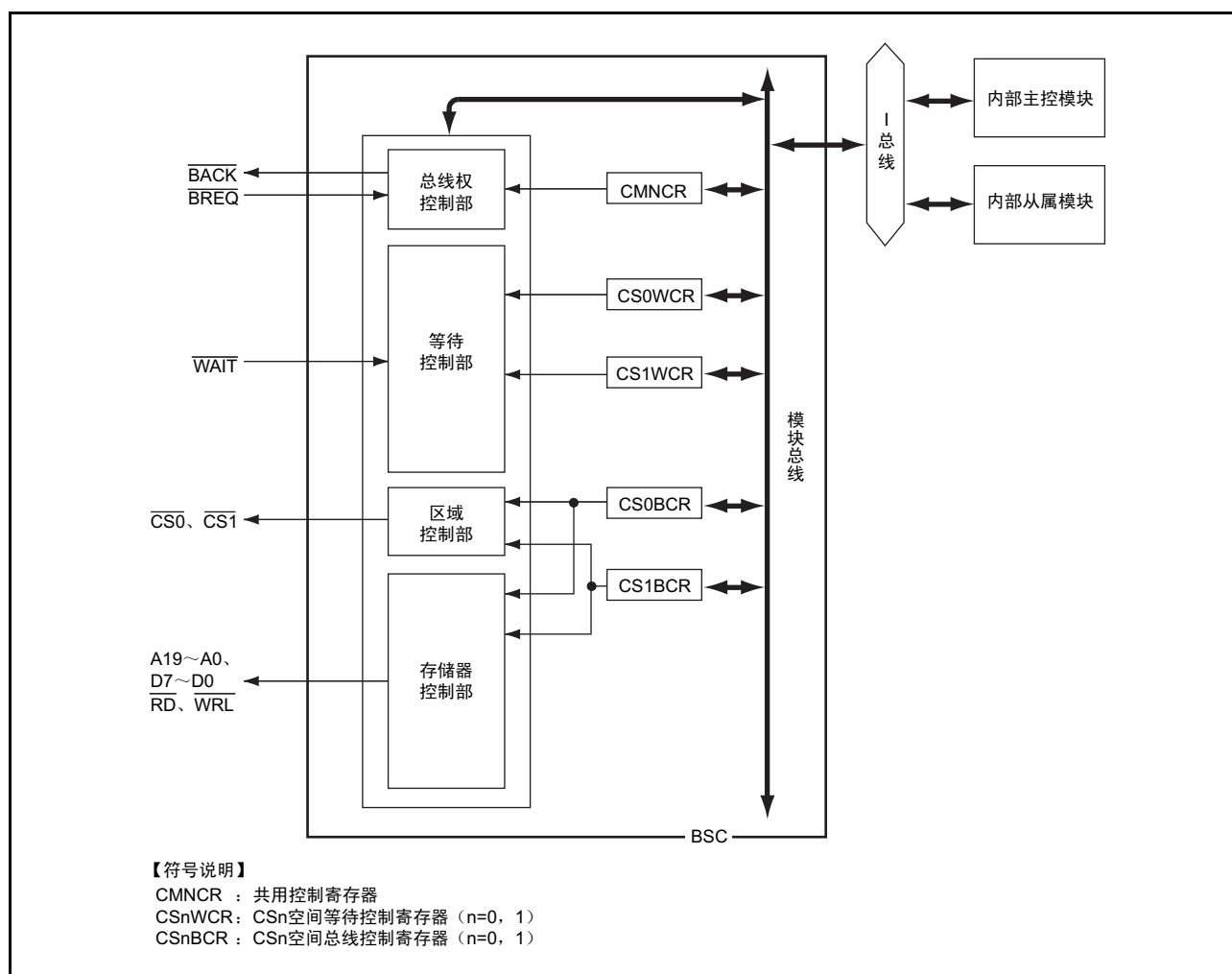


图 9.1 BSC 的框图

9.2 输入 / 输出引脚

BSC 的引脚结构如表 9.1 所示。

表 9.1 引脚结构

引脚名	输入 / 输出	功能
A19 ~ A0	输出	地址总线
D7 ~ D0	输入 / 输出	数据总线
$\overline{\text{CS0}}$ 、 $\overline{\text{CS1}}$	输出	片选
$\overline{\text{RD}}$	输出	读脉冲信号（读数据的输出允许信号）
$\overline{\text{WRL}}$	输出	对应 D7 ~ D0 的字节写指示
$\overline{\text{WAIT}}$	输入	外部等待的输入
$\overline{\text{BREQ}}$	输入	总线权请求的输入
$\overline{\text{BACK}}$	输出	总线使用的允许输出

9.3 区域概要

9.3.1 空间分割

本 LSI 为 32 位地址空间的体系结构。

如表 9.2 ~ 表 9.4 所示，本 LSI 能分别将各种存储器连接到 2 个空间，输出对应各存储器的片选信号（ $\overline{\text{CS0}}$ 和 $\overline{\text{CS1}}$ ）。 $\overline{\text{CS0}}$ 在存取区域 0 时有效。

9.3.2 地址映像

外部地址空间共有 2M 字节，被分割为 2 个空间使用。按空间指定被连接的存储器种类和数据总线宽度。全空间的地址映像如表 9.2 ~ 表 9.4 所示。

表 9.2 地址映像（SH7136、SH7137 单芯片模式）

地址	空间	存储器种类	容量	总线宽度
H'00000000 ~ H'0003FFFF	内部 ROM		256K 字节	32 位
H'00040000 ~ H'FFFF7FFF	保留			
H'FFFF8000 ~ H'FFFBFFFF	内部 RAM		16K 字节	32 位
H'FFFC000 ~ H'FFFFFFF	内部外围模块		16K 字节	8/16 位

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

表 9.3 地址映像（SH7137、内部 ROM 有效模式）

地址	空间	存储器种类	容量	总线宽度
H'00000000 ~ H'0003FFFF	内部 ROM		256K 字节	32 位
H'00040000 ~ H'01FFFFFF	保留			
H'02000000 ~ H'020FFFFFFF	CS0 空间	通常空间	1M 字节	8 位
H'02100000 ~ H'03FFFFFF	保留			
H'04000000 ~ H'040FFFFFFF	CS1 空间	通常空间	1M 字节	8 位
H'04100000 ~ 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】 不能存取保留空间，否则就不能保证运行。在单芯片模式中，不能使用内部 ROM、内部 RAM 和内部外围模块以外的空间。

表 9.4 地址映像（SH7137、内部 ROM 无效模式）

地址	空间	存储器种类	容量	总线宽度
H'00000000 ~ H'000FFFFFFF	CS0 空间	通常空间	1M 字节	8 位
H'00100000 ~ H'03FFFFFF	保留			
H'04000000 ~ H'040FFFFFFF	CS1 空间	通常空间	1M 字节	8 位
H'04100000 ~ 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】 不能存取保留空间，否则就不能保证运行。

9.4 寄存器说明

BSC 有以下寄存器。有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 25 章 寄存器一览”。

表 9.5 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
共用控制寄存器	CMNCR	R/W	H'00001010	H'FFFFFF000	32
CS0 空间总线控制寄存器	CS0BCR	R/W	H'36DB0600	H'FFFFFF004	32
CS1 空间总线控制寄存器	CS1BCR	R/W	H'36DB0600	H'FFFFFF008	32
CS0 空间等待控制寄存器	CS0WCR	R/W	H'00000500	H'FFFFFF028	32
CS1 空间等待控制寄存器	CS1WCR	R/W	H'00000500	H'FFFFFF02C	32
总线功能扩展寄存器	BSCEHR	R/W	H'0000	H'FFFE89A	8、16

9.4.1 共用控制寄存器 (CMNCR)

CMNCR 是共同控制各区域的 32 位寄存器。

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

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	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	初始值	R/W	说明
31 ~ 13	—	全 0	R	保留位 读写值总是 0。
12	—	1	R	保留位 读写值总是 1。
11 ~ 5	—	全 0	R	保留位 读写值总是 0。
4	—	1	R	保留位 读写值总是 1。
3、2	—	全 0	R	保留位 读写值总是 0。
1	HIZMEM	0	R/W	High-Z 存储器控制 此位指定软件待机模式时的 A19 ~ A0、 $\overline{\text{CSn}}$ 、 $\overline{\text{WRL}}$ 和 $\overline{\text{RD}}$ 引脚的状态。在释放总线时，与此位无关，这些引脚为高阻抗状态。 0: 在软件待机模式时，引脚为高阻抗状态 1: 在软件待机模式时，引脚为驱动状态
0	—	0	R	保留位 读写值总是 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和存取周期之间的等待数。

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

【注】* 在内部ROM无效时此位为0。

位	位名	初始值	R/W	说明
31、30	—	全 0	R	保留位 读写值总是 0。
29、28	IWW[1:0]	11	R/W	写 - 读 / 写 - 写周期之间的空闲指定 此位指定在存取已连接空间的存储器后要插入的空闲周期数。写 - 读周期和写 - 写周期为对象周期。 00: 不插入空闲周期 01: 插入 1 个空闲周期 10: 插入 2 个空闲周期 11: 插入 4 个空闲周期
27	—	0	R	保留位 读写值总是 0。
26、25	IWRWD[1:0]	11	R/W	不同空间的读 - 写周期之间的空闲指定 此位指定在存取已连接空间的存储器后要插入的空闲周期数。对象周期是在不同空间的连续读 - 写周期。 00: 不插入空闲周期 01: 插入 1 个空闲周期 10: 插入 2 个空闲周期 11: 插入 4 个空闲周期
24	—	0	R	保留位 读写值总是 0。
23、22	IWRWS[1:0]	11	R/W	同一空间的读 - 写周期之间的空闲指定 此位指定在存取已连接空间的存储器后要插入的空闲周期数。对象周期是在同一空间的连续读 - 写周期。 00: 不插入空闲周期 01: 插入 1 个空闲周期 10: 插入 2 个空闲周期 11: 插入 4 个空闲周期
21	—	0	R	保留位 读写值总是 0。
20、19	IWRRD[1:0]	11	R/W	不同空间的读 - 读周期之间的空闲指定 此位指定在存取已连接空间的存储器后要插入的空闲周期数。对象周期是在不同空间的连续读 - 读周期。 00: 不插入空闲周期 01: 插入 1 个空闲周期 10: 插入 2 个空闲周期 11: 插入 4 个空闲周期
18	—	0	R	保留位 读写值总是 0。
17、16	IWRRS[1:0]	11	R/W	同一空间的读 - 读周期之间的空闲指定 此位指定在存取已连接空间的存储器后要插入的空闲周期数。对象周期是在同一空间的连续读 - 读周期。 00: 不插入空闲周期 01: 插入 1 个空闲周期 10: 插入 2 个空闲周期 11: 插入 4 个空闲周期

位	位名	初始值	R/W	说明
15 ~ 11	—	全 0	R	保留位 读写值总是 0。
10、9	BSZ[1:0]	01/11*	R/W	数据总线宽度的指定 此位指定空间的数据总线宽度。在内部 ROM 有效时，为了在 CSn 空间存取前将数据总线宽度指定为 8 位，必须写 B'01。 【注】 在内部 ROM 无效时，与 CS0BCR 的 BSZ1 和 BSZ0 位的设定无关，区域 0 的数据总线宽度为 8 位。
8 ~ 0	—	全 0	R	保留位 读写值总是 0。

【注】 * 在内部 ROM 无效时为 01。

9.4.3 CSn 空间等待控制寄存器 (CSnWCR) (n=0、1)

CSnWCR 设定与存储器存取有关的各种等待周期，必须在存取对象区域前并且在设定 CSnBCR 寄存器后设定 CSnWCR 寄存器。

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	初始值	R/W	说明
31 ~ 19	—	全 0	R	保留位 读写值总是 0。
18 ~ 16	WW[2:0]	000	R/W	写存取的等待周期数 此位指定写存取所需的周期数。 000: 与 WR3 ~ WR0 设定 (读存取等待) 的周期相同 001: 0 个周期 010: 1 个周期 011: 2 个周期 100: 3 个周期 101: 4 个周期 110: 5 个周期 111: 6 个周期
15 ~ 13	—	全 0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
12、11	SW[1:0]	00	R/W	地址和 $\overline{\text{CSn}}$ 有效 $\rightarrow \overline{\text{RD}}$ 和 $\overline{\text{WRL}}$ 有效的延迟周期数 此位指定从地址和 $\overline{\text{CSn}}$ 有效到 $\overline{\text{RD}}$ 和 $\overline{\text{WRL}}$ 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	读存取的等待周期数 此位指定读存取所需的等待周期数。 0000: 0 个周期 0001: 1 个周期 0010: 2 个周期 0011: 3 个周期 0100: 4 个周期 0101: 5 个周期 0110: 6 个周期 0111: 8 个周期 1000: 10 个周期 1001: 12 个周期 1010: 14 个周期 1011: 18 个周期 1100: 24 个周期 1101: 保留 (不能设定) 1110: 保留 (不能设定) 1111: 保留 (不能设定)
6	WM	0	R/W	外部等待屏蔽的指定 此位指定是将外部等待输入设定为有效还是忽视外部等待输入。即使 存等待周期数为 0, 此位的设定也有效。 0: 外部等待输入有效 1: 忽视外部等待输入
5 ~ 2	—	全 0	R	保留位 读写值总是 0。
1、0	HW[1:0]	00	R/W	$\overline{\text{RD}}$ 和 $\overline{\text{WRL}}$ 无效 $\rightarrow$ 地址和 $\overline{\text{CSn}}$ 无效的延迟周期数 此位指定从 $\overline{\text{RD}}$ 和 $\overline{\text{WRL}}$ 无效到地址和 $\overline{\text{CSn}}$ 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 是 16 位寄存器, 设定 DTC 的总线权释放时序等。在优先进行 DTC 传送以及减少 DTC 启动周期数时, BSCEHR 能设定有效功能。

有关 DTLOCK 位、CSSTP1 位和 DTBST 位组合产生的 DTC 运行的不同点, 请参照“8.5.9 DTC 的总线权释放时序”。

通过设定 CSSTP2 位, 能提高 DTLOCK 位为 0 时的 DTC 传送性能。还能通过设定 CSSTP3 位, 选择 DTC 传送和 CPU 存取外部空间的优先级。

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

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

另外，不能在启动 DTC 时更改此寄存器。

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

位	位名	初始值	R/W	说明
15	DTLOCK	0	R/W	DTC 锁定的允许 此位选择 DTC 的总线权释放时序。 0: 在发行读向量后的 NOP 时或者回写传送信息后，释放总线权。 1: 在读向量后、或者在发行读向量后的 NOP 时、或者读传送信息后、或者传送 1 次数据后或者回写传送信息后，释放总线权。
14	CSSTP1	0	R/W	发行 DTC 的 NOP 时的总线权释放的选择 此位选择是否在启动 DTC 期间发行读向量后的 NOP 时，对 CPU 存取外部空间的请求释放总线权。 但是，当 CSSTP2 位是 1 时，与此位的设定无关，始终不释放总线权。 0: 在发行 DTC 的 NOP 时释放总线权 1: 在发行 DTC 的 NOP 时不释放总线权
13	—	0	R	保留位 读写值总是 0。
12	CSSTP2	0	R/W	DTC 传送中的总线权释放的选择 此位在 DTLOCK 位为 0 时的 DTC 传送中，对于 CPU 存取外部空间的请求，选择是否在每传送 1 次数据后释放总线权。 0: 当 DTLOCK 位和 CSSTP1 位都为 0 时，在发行读向量后的 NOP 时释放总线权；当 DTLOCK 位为 0 而 CSSTP1 位为 1 时，每传送 1 次数据后释放总线权 1: 在全部数据传送结束后释放总线权
11	DTBST	0	R/W	DTC 突发的允许 当 DTC 产生多个启动请求时，不释放总线权而连续启动 DTC。 0: 每当 1 个 DTC 启动源结束时，释放总线权。 1: 在 DTC 启动源全部结束前，不释放总线权而连续启动 DTC。 【注】 当将此位置 1 时，有以下限制： 1. 必须通过频率控制寄存器（FRQCR）将时钟设定为 $I\phi:B\phi:P\phi:MI\phi:MP\phi=8:4:4:4:4$ 、 $4:2:2:2:2$ 或者 $2:1:1:1:1$ 。 2. 必须将向量信息分配到内部 ROM 或者内部 RAM。 3. 必须将传送信息分配到内部 RAM。 4. 必须将传送源和传送目标设定在内部 RAM 和内部外围模块之间、或者外部存储器和内部外围模块之间。

位	位名	初始值	R/W	说明
10	DTSA	0	R/W	DTC 短地址模式 是用 3 个长字读 DTC 传送信息的模式。 0: 用 4 个长字读传送信息。传送信息的分配为图 8.2 的正常模式时的分配。 1: 用 3 个长字读传送信息。传送信息的分配为图 8.2 的短地址模式时的分配。 【注】 因为将 SAR 和 DAR 的高 8 位视为 1，所以只能在内部外围模块和内部 RAM 之间传送时使用。
9	CSSTP3	0	R/W	CPU 存取外部存储器的优先选择 此位选择在 DTC 传送时是否优先 CPU 存取外部空间。 0: 优先 DTC 传送 1: 优先 CPU 存取外部空间 【注】 此位为 0 时，在 CPU 存取内部 I/O 及其后发行的 CPU 存取外部空间之间，插入 1 个 Bϕ 的 NOP。
8	DTPR	0	R/W	DTC 启动优先级的指定 在 DTC 启动前发生多个 DTC 启动请求时，此位选择是从最先发生的启动请求开始传送还是根据 DTC 的启动优先级开始传送。 但是，在启动 DTC 时发生多个 DTC 启动请求的情况下，要根据 DTC 的启动优先级进行下一次传送。 0: 从最先发生的 DTC 启动请求开始传送 1: 根据 DTC 的启动优先级开始传送 【注】 当将此位置 1 时，有以下限制： 1. 必须将向量信息分配到内部 ROM 或者内部 RAM。 2. 必须将传送信息分配到内部 RAM。 3. 传送信息的跳读取功能总是无效。
7 ~ 0	—	全 0	R	保留位 读写值总是 0。

9.5 运行说明

9.5.1 字节排序 / 存取长度和数据调整

本 LSI 支持高位字节（MSB）为地址 0 的大端法字节数据的排列方法。

数据总线宽度为 8 位。根据各器件的数据总线宽度进行数据调整，因此，如果要从 8 位器件中读长字数据，就必须读 4 次。在本 LSI 的各接口之间自动进行数据调整和数据长度的转换。

器件的数据长度和存取单位的关系如表 9.6 所示。

表 9.6 8 位外部器件的存取和数据调整

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

9.5.2 通常空间接口

(1) 基本时序

通常空间存取主要针对直接连接没有字节选择引脚的 SRAM 输出选通信号。通常空间存取的基本时序如图 9.2 所示。没有等待的通常存取以 2 个周期结束。

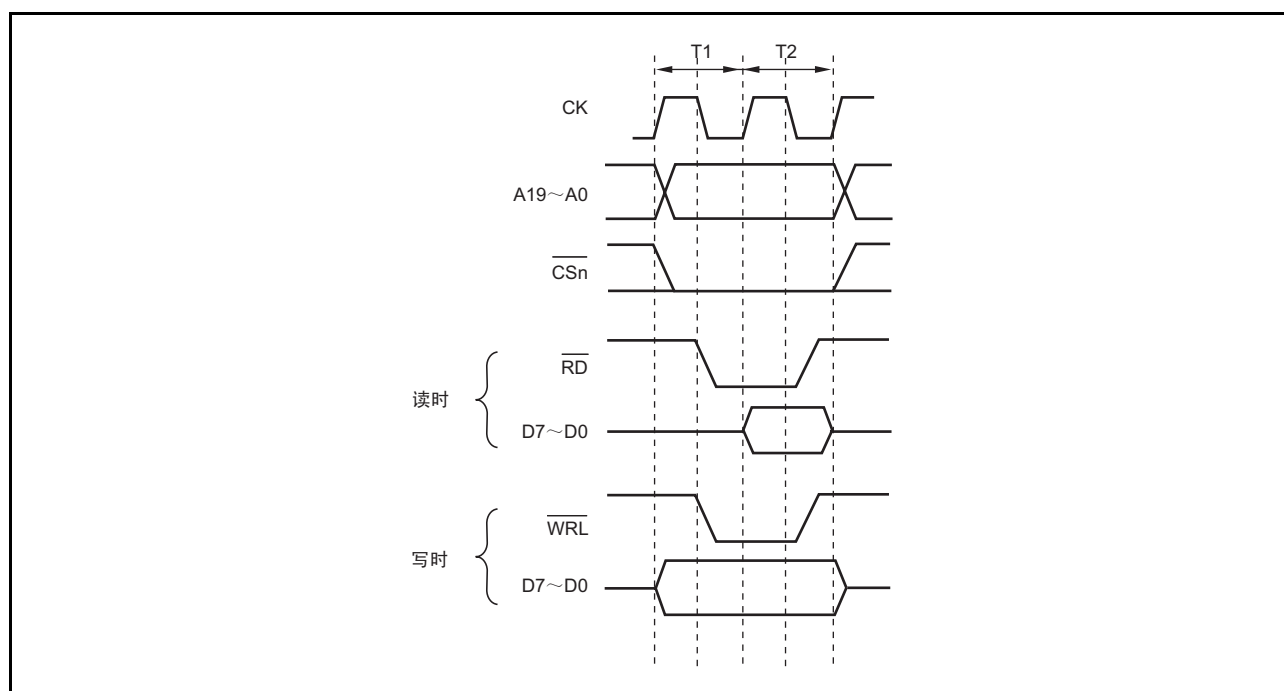


图 9.2 通常空间的基本存取（存取等待 = 0）

在给数据总线设置缓冲器时，需要使用 $\overline{RD}$ 进行读数据的输出控制。

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

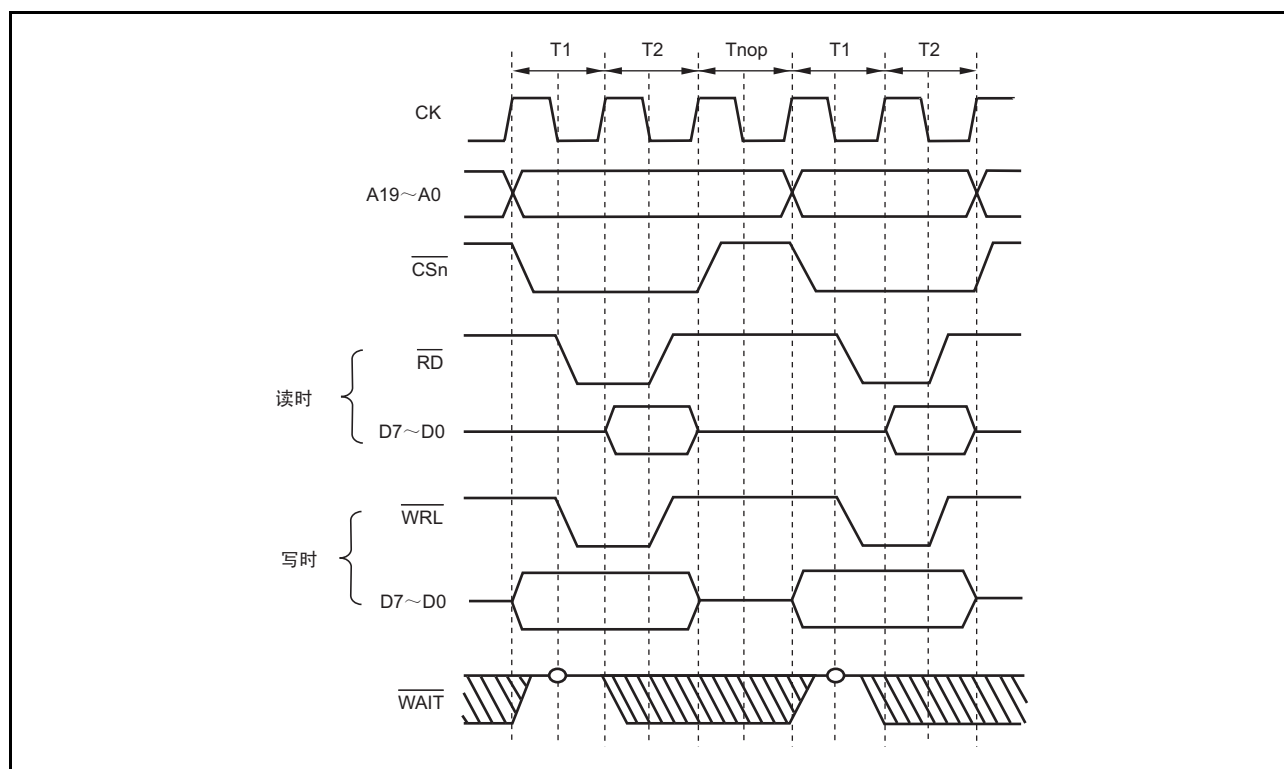


图 9.3 通常空间的连续存取例 1

8 位总线宽度、字存取、 $CSnWCR.WM$ 位 = 0（存取等待 = 0、周期之间的等待 = 0）

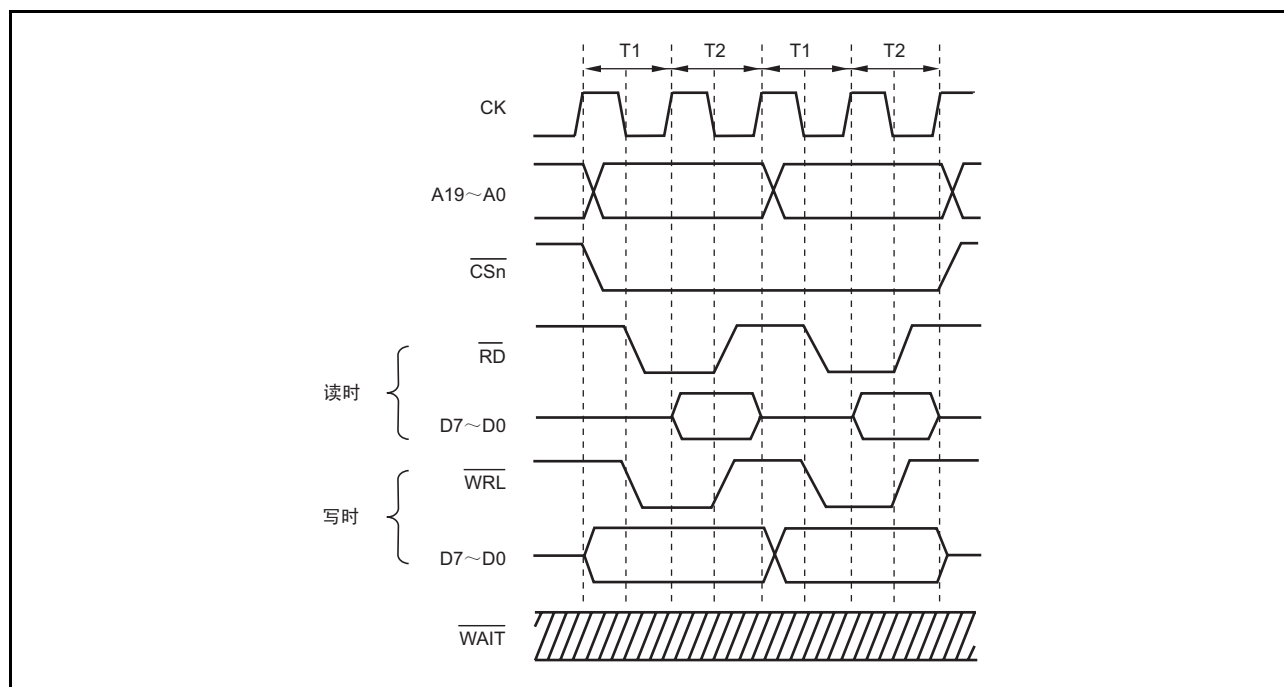


图 9.4 通常空间的连续存取例 2

8 位总线宽度、字存取、CSnWCR.WM 位 = 1 (存取等待 = 0、周期之间的等待 = 0)

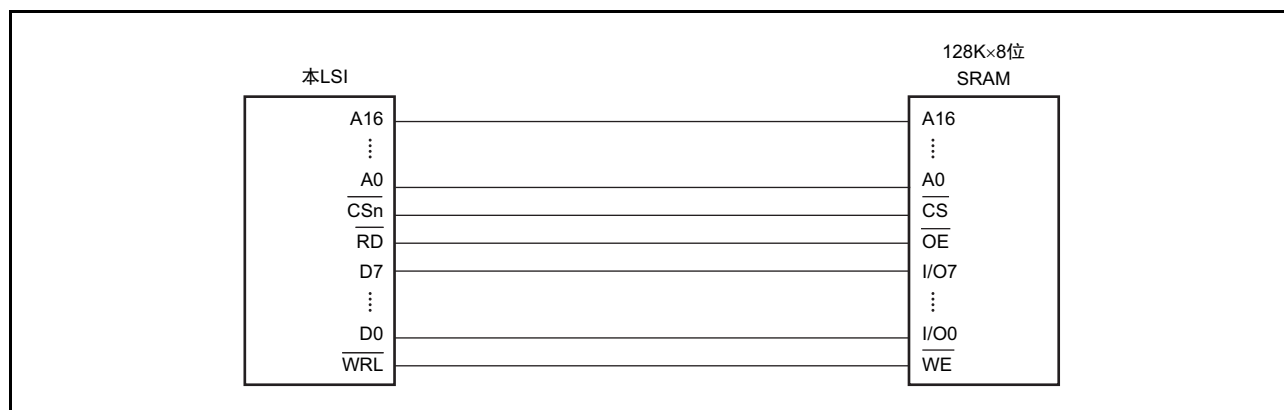


图 9.5 8 位数据宽度的 SRAM 连接例子

9.5.3 存取等待的控制

通过设定 CSnWCR 的 WR3 位、WR2 位、WR1 位和 WR0 位，能控制通常空间存取的等待周期的插入，还能在读写存取中独立插入等待周期。如图 9.6 所示，在通常空间存取中只插入被指定为等待周期的 Tw 周期数。

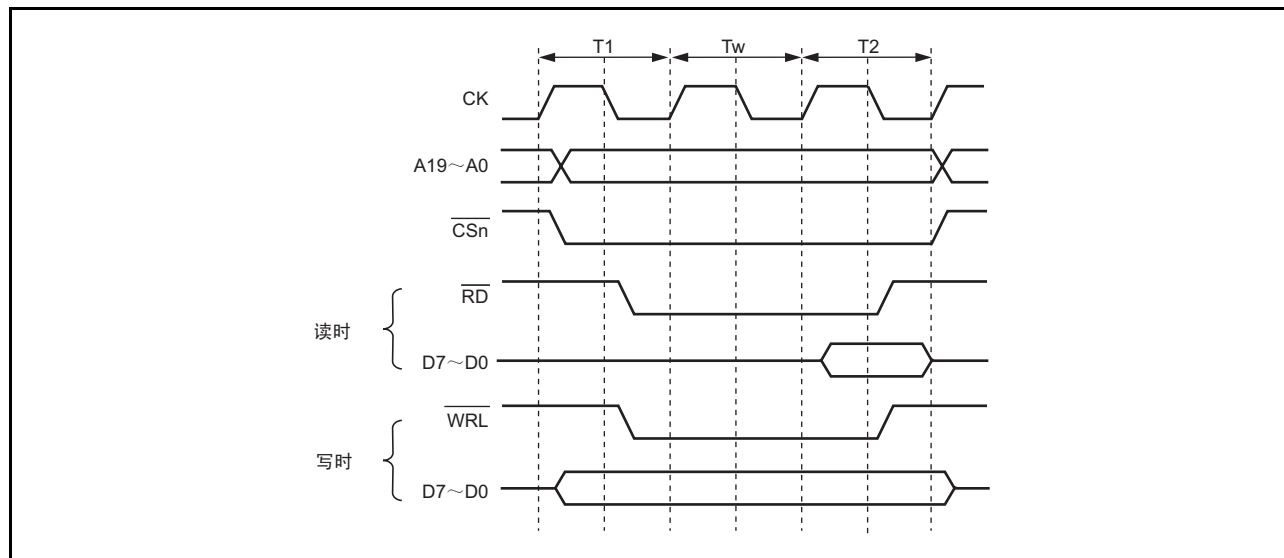


图 9.6 通常空间存取的等待时序（只限软件等待）

在将 CSnWCR 的 WM 位置 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。将 2 个周期的等待指定为软件等待，在从 T1 周期或者 Tw 周期转移到 T2 周期时，在 CK 的下降沿对 $\overline{\text{WAIT}}$ 信号进行采样。

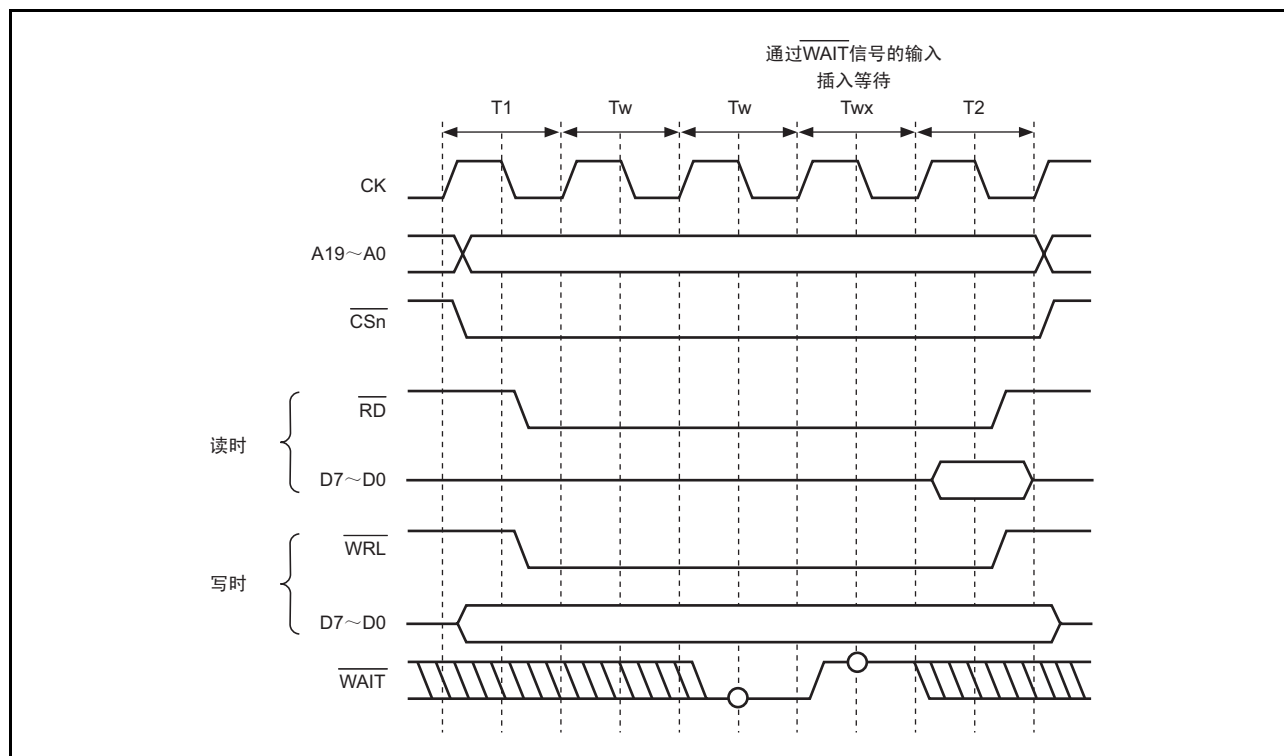


图 9.7 通常空间存取的等待时序（通过 $\overline{\text{WAIT}}$ 信号插入等待）

9.5.4 $\overline{\text{CSn}}$ 有效期间的扩展

通过设定 CSnWCR 的 SW1 位和 SW0 位, 能指定从 $\overline{\text{CSn}}$ 有效到 $\overline{\text{RD}}$ 和 $\overline{\text{WRL}}$ 有效的周期数。还能通过设定 HW1 位和 HW0 位, 指定从 $\overline{\text{RD}}$ 和 $\overline{\text{WRL}}$ 无效到 $\overline{\text{CSn}}$ 无效的周期数。因此, 能对应各种外部器件的接口。在如图 9.8 所示的例子中, 将 T_h 周期和 T_f 周期分别附加在通常周期的前后。在这些周期中, 除了 $\overline{\text{RD}}$ 和 $\overline{\text{WRL}}$ 无效以外, 其他信号都有效。另外, 因数据被延长输出到 T_f 周期为止, 所以对写操作较慢的器件有效。

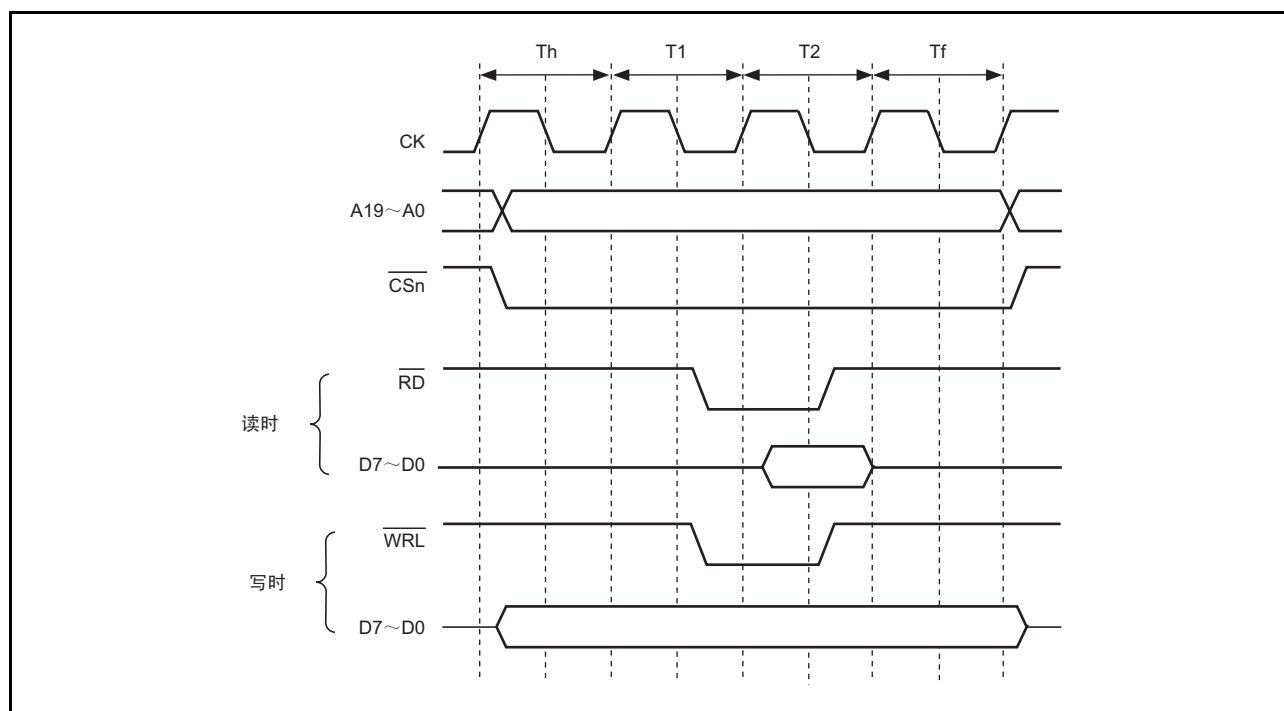


图 9.8 $\overline{\text{CSn}}$ 有效期间的扩展

9.5.5 存取周期之间的等待

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

通过 CSnBCR 的 $\text{IWW}[1:0]$ 位、 $\text{IWRWD}[1:0]$ 位、 $\text{IWRWS}[1:0]$ 位、 $\text{IWRRD}[1:0]$ 位和 $\text{IWRRS}[1:0]$ 位指定存取周期之间的等待周期数。能在存取周期之间插入等待 (空闲周期) 的条件如下:

1. 连续存取为写-读或者写-写
2. 连续存取为不同空间的读-写
3. 连续存取为相同空间的读-写
4. 连续存取为不同空间的读-读
5. 连续存取为相同空间的读-读

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

6. 与内部总线的接口有关的外部总线的空闲周期
 - a. 在外部总线空闲周期或者读周期后的写存取前, 插入 1 个空闲周期。
 - b. 为了在外部总线的读周期结束时将读数据传送到内部总线, 插入 1 个空闲周期。
在紧接着读周期后的写周期中, 包含上述 a 插入 2~3 个周期的空闲周期。

插入的最小空闲周期数如表 9.7 和表 9.8 所示。表中的“CSnBCR 的空闲设定”表示由 IWW、IWRWD、IWRWS、IWRRD 或者 IWRRS 设定的空闲周期数。

表 9.7 通常空间接口的 CPU 存取之间的最小空闲周期数

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

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

*1 在 8 位总线宽度的 16 位存取中，表示地址 0 的字节存取和地址 1 的字节存取之间的最小空闲周期数；在 8 位总线宽度的 32 位存取中，表示地址 0 的字节存取、地址 1 的字节存取、地址 2 的字节存取和地址 3 的字节存取之间的最小空闲周期数。

*2 上述以外

表 9.8 通常空间接口的 DTC 传送时的存取之间的最小空闲周期数

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

【注】 因为 DTC 是通过 $B\phi$ 运行，所以与时钟比无关。

*1 在 8 位总线宽度的 16 位存取中，表示地址 0 的字节存取和地址 1 的字节存取之间的最小空闲周期数；在 8 位总线宽度的 32 位存取中，表示地址 0 的字节存取、地址 1 的字节存取、地址 2 的字节存取和地址 3 的字节存取之间的最小空闲周期数。

*2 上述以外

9.5.6 总线仲裁

本 LSI 在通常状态下占有总线权，在接受外部器件的总线权请求后释放总线。另外，本 LSI 内部有 CPU 和 DTC 共 2 个总线主控，释放给这些总线主控的总线权的优先级如下：

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

但是，在 DTC 请求总线权的状态下，CPU 不能连续获得总线权。

当总线功能扩展寄存器 (BSCEHR) 的 CSSTP2 位为 1 时，CPU 存取外部空间的请求优先级低于 DTC 传送请求 (总线功能扩展寄存器 (BSCEHR) 的 DTLOCK 位为 0 时)。

另外，对于“9.5.7 (2) LSI 内部总线主控的存取”中记述的写缓冲器操作，在 CPU 读写外部空间时，CPU 和 DTC 的仲裁不同。在 CPU 存取外部空间期间发生 DTC 启动请求时的总线仲裁如图 9.9 所示。

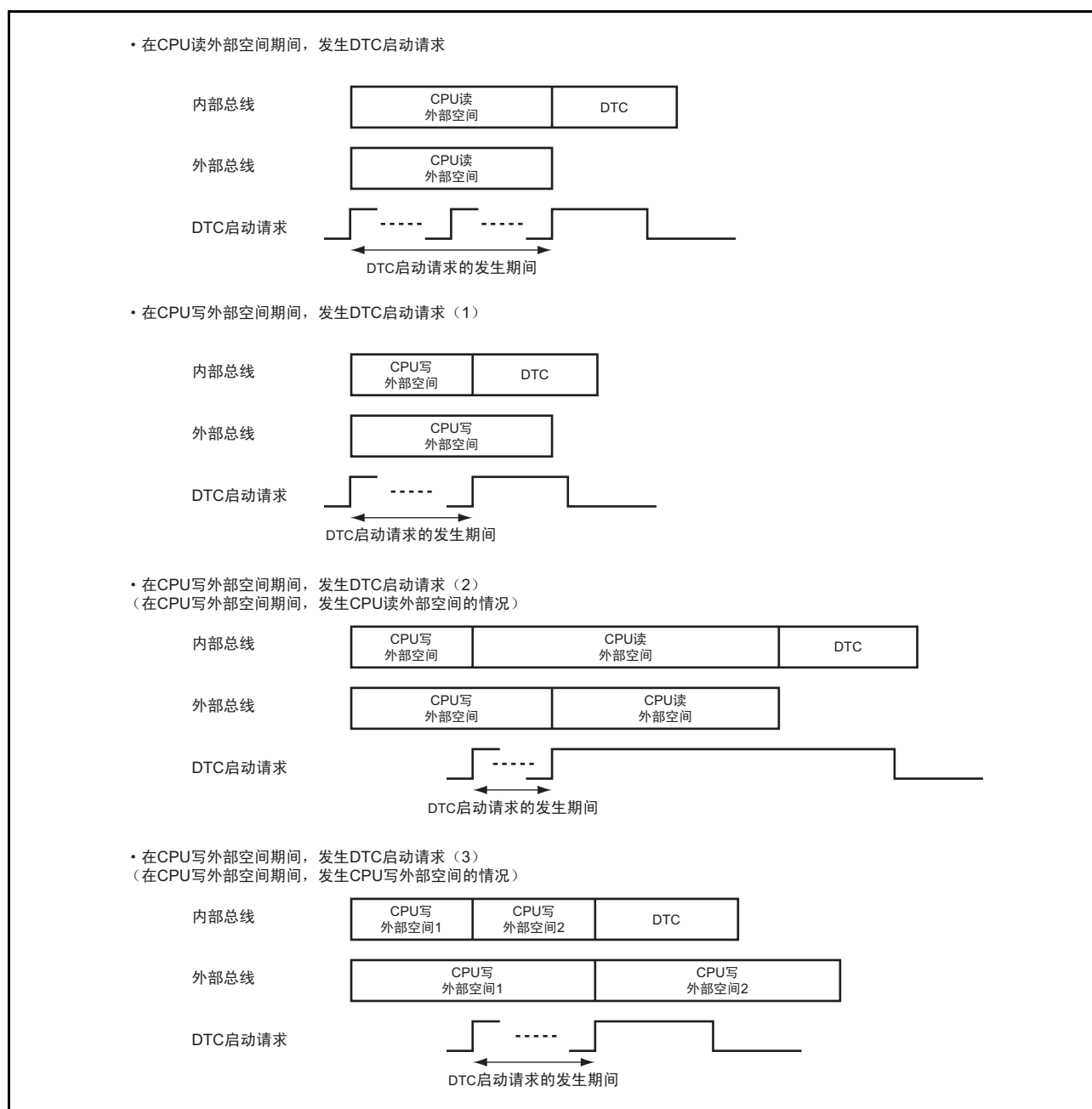


图 9.9 在 CPU 存取外部空间期间发生 DTC 启动请求时的总线仲裁

不进行总线仲裁的状态如下所示：

1. 在TAS指令的读周期和写周期之间
2. 因数据总线宽度小于存取长度而产生的多个总线周期（例如，在对8位数据总线宽度的存储器进行长字存取时的总线周期之间）

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

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

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

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

另外，即使在将总线释放给外部器件期间，只要不存取外部器件，本 LSI 就继续处理。然后，在存取外部器件时，进入总线权返回等待状态。

在释放总线期间，不能转移到睡眠模式、软件待机模式和深度软件待机模式。

具体的总线释放顺序如下：

首先与 CK 的上升沿同步，将地址总线和数据总线设定为高阻抗，在此 0.5 个周期后与 CK 的下降沿同步，将总线使用允许信号置为有效，然后在 CK 的下一个上升沿，将 $\overline{\text{CSn}}$ 等总线控制信号设定为高阻抗。这些总线控制信号至少在变为高阻抗的 1 个周期前被设定为高电平。在 CK 的下降沿对总线权请求信号进行采样。

从外部器件重新获得总线权的顺序如下：

如果在 CK 的下降沿检测到 $\overline{\text{BREQ}}$ 无效，就在 1.5 个周期后用高电平开始驱动总线控制信号。接着在 CK 的下一个下降沿，将总线使用的允许信号设定为无效。然后在 CK 的上升沿，开始驱动地址总线和数据总线。在总线控制信号置为有效后到实际开始总线周期，最快的情况是在 CK 上升沿（与驱动地址和数据信号相同）。总线仲裁时序如图 9.10 所示。

如果发生了总线权请求（ $\overline{\text{BREQ}}$ 为低电平有效），就必须在允许使用总线（ $\overline{\text{BACK}}$ 为低电平有效）后，释放总线权（ $\overline{\text{BREQ}}$ 为高电平无效）。如果在 $\overline{\text{BACK}}$ 有效前将 $\overline{\text{BREQ}}$ 置为无效，根据 $\overline{\text{BREQ}}$ 无效时序， $\overline{\text{BACK}}$ 只有 1 个周期有效，可能在本 LSI 之间发生总线冲突。

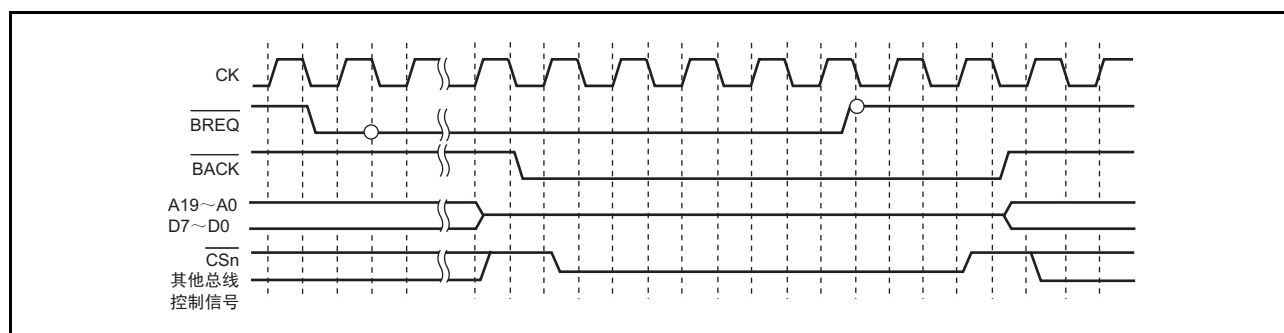


图 9.10 总线仲裁

在对 DTC 进行总线仲裁时，不在中途插入 NOP 等，而连续进行总线存取处理。

9.5.7 其他

(1) 复位

只在上电复位时才对总线状态控制器进行完全的初始化。在上电复位时，与是否正在执行总线周期无关，立即将全部信号置为无效，关闭输出缓冲器，并对全部控制寄存器进行初始化。在待机模式、睡眠模式和手动复位时，不对总线状态控制器的控制寄存器进行初始化。如果进行手动复位，就只在当前正在执行的总线周期结束前执行该总线周期，然后进入存取等待状态。在手动复位信号有效时，不接受 $\overline{\text{BREQ}}$ 产生的仲裁请求。

(2) LSI 内部总线主控的存取

本 LSI 内部总线分为 L 总线、I 总线和外部总线共 3 种。CPU 连接 L 总线，DTC 和总线状态控制器连接 I 总线，低速外围模块连接外部总线，内部存储器连接 L 总线和 I 总线。

在存取外部空间和内部外围模块时，通过 I 总线启动存取周期。因此，在 CPU 存取内部存储器时，能在不发生和 CPU 总线仲裁的情况下启动 DTC。

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

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

对于 DTC 的存取，总线状态控制器的写缓冲器也同样起作用。

如果在写缓冲器时更改 BSC 的寄存器，就不能正确存取，因此不能在写存取后立即更改 BSC 的寄存器。如果需要在写存取后立即更改 BSC 的寄存器，就必须虚读写数据，并在确认写存取结束后更改 BSC 的寄存器。

9.5.8 CPU 存取内部 FLASH 和内部 RAM

和 $I\phi$ 时钟同步，以 1 个时钟周期进行内部 FLASH 的读存取。有关编程和擦除请参照“第 22 章 闪存”。和 $I\phi$ 时钟同步，以 1 个时钟周期进行内部 RAM 的读写存取，详细内容请参照“第 23 章 RAM”。

9.5.9 CPU 存取内部外围 I/O 寄存器

CPU 存取内部外围 I/O 寄存器时的存取周期数如表 9.9 所示。

表 9.9 内部外围 I/O 寄存器的存取周期数

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

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

本 LSI 采用了同步逻辑和分层总线结构。L 总线和 $I\phi$ 时钟的上升沿同步，I 总线和 $B\phi$ 时钟的上升沿同步，外部总线和 $P\phi$ 时钟的上升沿同步，进行数据的输入 / 输出。当 $I\phi:B\phi:P\phi=4:2:2$ 时，对 16 位外部总线宽度的 2 个 $P\phi$ 周期存取的寄存器，进行字节数据写存取的时序例子如图 9.11 所示。在存取内部外围 I/O 寄存器时，CPU 在将数据输出到 L 总线后需要 3 个 $I\phi$ 周期准备 I 总线的数据传送。在此 3 个周期后，能和 $B\phi$ 的上升沿同步将数据传送到 I 总线。当 $I\phi:B\phi=4:2$ 时，在 1 个 $B\phi$ 时钟期间有 2 个 $I\phi$ 时钟。因此，当 $I\phi:B\phi=2:1$ 时，在 $(3+n)\times I\phi$ 、 $n=0\sim 1$ 期间，将数据从 L 总线传送到 I 总线（图中为 $3\times I\phi$ ）。将数据传送到 L 总线的时序和 $B\phi$ 上升沿时序的关系取决于程序的执行状态。在图的例子中，因为 $n=0$ 、 $m=0$ ，所以存取时间为 $3\times I\phi+1\times B\phi+2\times P\phi$ 。

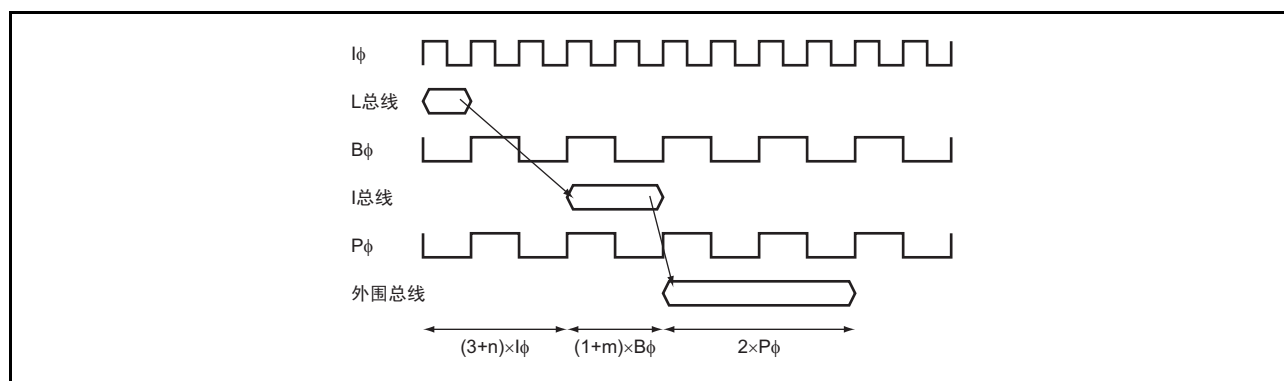


图 9.11 $I\phi:B\phi:P\phi=4:2:2$ 时的内部外围 I/O 寄存器的写存取时序

当 $I\phi:B\phi:P\phi=4:2:1$ 时，外部总线读存取的时序例子如图 9.12 所示。从 L 总线到外部总线的传送与写操作相同，但是需要将外部总线读取的值传送到 CPU。从外部总线到 I 总线的传送以及从 I 总线到 L 总线的传送都与各总线时钟的上升沿同步进行，由于 $I\phi \geq B\phi \geq P\phi$ ，所以实际需要 $2\times I\phi$ 周期。在图的例子中，因为 $n=0$ 、 $m=1$ ，所以存取时间为 $3\times I\phi+2\times B\phi+2\times P\phi+2\times I\phi$ 。

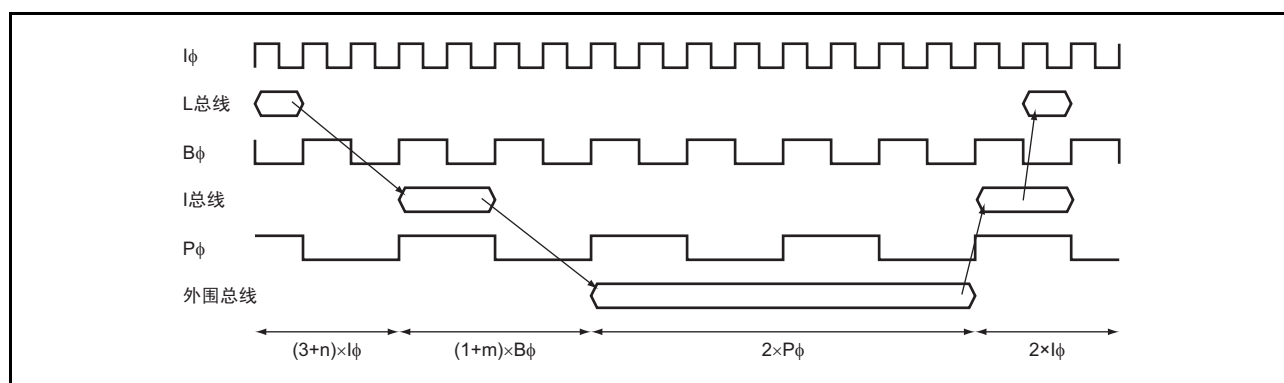


图 9.12 $I\phi:B\phi:P\phi=4:2:1$ 时的内部外围 I/O 寄存器的读存取时序

9.5.10 CPU 存取外部存储器

CPU 存取外部存储器时的存取周期数如表 9.10 所示。除表 9.10 所示的时钟比、存取长度、LSI 的外部总线宽度以外，外部存取周期还因插入等待的设定等发生变化。有关插入等待的设定等详细内容，请参照“9.4 寄存器说明”。

表 9.10 外部存取周期数

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

【注】 n : 当 $I\phi:B\phi=8:1$ 时, $n=0 \sim 7$

当 $I\phi:B\phi=4:1$ 时, $n=0 \sim 3$

当 $I\phi:B\phi=3:1$ 时, $n=0 \sim 2$

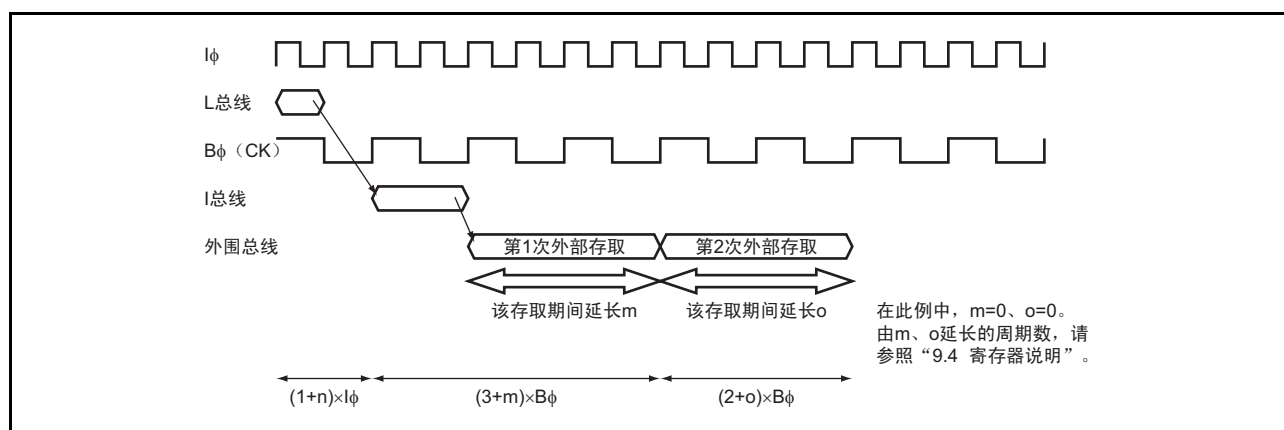
当 $I\phi:B\phi=2:1$ 时, $n=0 \sim 1$

当 $I\phi:B\phi=1:1$ 时, $n=0$

m、o : m: 等待设定, o: 等待设定+空闲设定等

详细内容请参照“9.4 寄存器说明”。

本 LSI 采用了同步逻辑和分层总线结构。L 总线和 $I\phi$ 时钟的上升沿同步, I 总线、外部总线和 $B\phi$ 时钟的上升沿同步, 进行数据的输入 / 输出。当 $I\phi:B\phi=2:1$ 时, 对 8 位外部总线宽度进行字数据写存取的时序例子如图 9.13 所示。CPU 在将数据输出到 L 总线后, 和 $B\phi$ 的上升沿同步将数据传送到 I 总线。当 $I\phi:B\phi=2:1$ 时, 在 1 个 $B\phi$ 时钟期间有 2 个 $I\phi$ 时钟。因此, 当 $I\phi:B\phi=2:1$ 时, 在 $(1+n) \times I\phi$ 、 $n=0、1$ 期间, 将数据从 L 总线传送到 I 总线 (图中为 2 个 $I\phi$)。将数据传送到 L 总线的时序和 $B\phi$ 上升沿时序的关系取决于程序的执行状态。传送到 I 总线的的数据在 1 个 $B\phi$ 周期后被传送到外部总线。1 次外部存取最快需要 2 个周期, 但是通过 BSC 寄存器的设定能延长存取周期 (存取周期计算式的 m、o)。在图 9.13 的例子中, 为了对 8 位总线宽度进行字数据的写存取, 需要 2 次外部存取。在图的例子中, 因为 $n=1$ 、 $m=0$ 、 $o=0$, 所以存取时间为 $2 \times I\phi + 3 \times B\phi + 2 \times B\phi$ 。

图 9.13 $I\phi:B\phi=2:1$ 时, 对 8 位外部总线宽度的外部存储器进行字数据写存取的时序

当 $I\phi:B\phi=4:1$ 时, 在外部总线宽度 $\geq$ 数据宽度的条件下, 读存取的时序例子如图 9.14 所示。从 L 总线到外部总线的传送与写操作相同, 但是需要将外部总线读取的值传送到 CPU。从外部总线到 I 总线的传送以及从 I 总线到 L 总线的传送和各总线时钟的上升沿同步进行, 在实际运行中, 从外部总线到 L 总线的传送在 1 个 ϕ 期间内完成。在图的例子中, 因为 $n=2$ 、 $m=0$ 、 $o=0$, 所以存取时间为 $3 \times I\phi + 3 \times B\phi + 1 \times I\phi$ 。

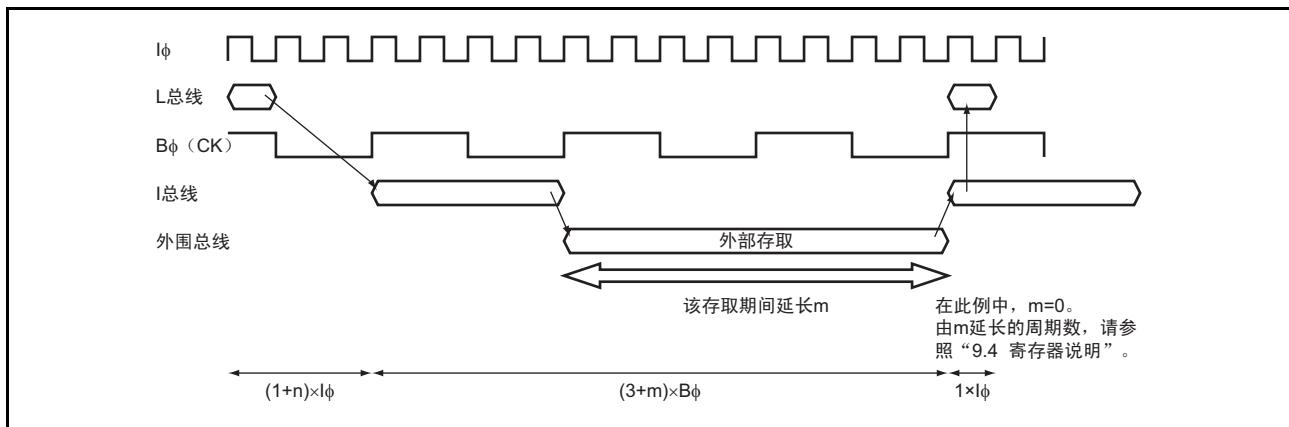


图 9.14 当 $I\phi:B\phi=4:1$ 时，在外部总线宽度 $\geq$ 数据宽度条件下的读存取时序

由 DTC 进行的存取周期数为 CPU 存取周期数减去 L 总线存取所需的 $I\phi$ 周期数。

第 10 章 多功能定时器脉冲单元 2 (MTU2)

本 LSI 内置由 6 个通道的 16 位定时器构成的多功能定时器脉冲单元 2 (MTU2)。

10.1 特点

- 最多能进行 16 个脉冲输入/输出和 3 个脉冲输入
- 每个通道能选择 8 种计数器输入时钟 (通道 5 能选择 4 种)
- 通道 0~4 能设定以下的运行: 比较匹配的波形输出、输入捕捉功能、计数器的清除、多个定时器计数器 (TCNT) 的同时写、比较匹配/输入捕捉的同时清除、计数器同步运行的各寄存器同步输入/输出、和同步运行组合进行最多 12 相的 PWM 输出。
- 通道 0、3、4 能设定缓冲运行
- 通道 1、2 能各自独立设定相位计数模式
- 级联运行
- 内部 16 位总线的高速存取
- 28 种中断源
- 能自动传送寄存器数据
- 能生成 A/D 转换器的转换开始触发
- 能设定模块待机模式
- 能通过 CH3、4 联动运行设定互补 PWM 或者复位 PWM 的正负 3 相共 6 相的波形输出
- 能通过 CH0、3、4 联动运行设定使用互补 PWM 或者复位 PWM 的 AC 同步马达 (无刷 DC 马达) 驱动模式, 并能选择 2 种 (斩波和电平) 波形输出
- CH5 具有用于空载时间补偿的计数器功能
- 在互补 PWM 模式中, 能减少在计数器波峰/波谷的中断和 A/D 转换器的转换开始触发

表 10.1 MTU2 的功能一览表

项目	通道 0	通道 1	通道 2	通道 3	通道 4	通道 5
计数时钟	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	MPφ/1 MPφ/4 MPφ/16 MPφ/64
通用寄存器 (TGR)	TGRA_0 TGRB_0 TGRE_0	TGRA_1 TGRB_1	TGRA_2 TGRB_2	TGRA_3 TGRB_3	TGRA_4 TGRB_4	TGRU_5 TGRV_5 TGRW_5
通用寄存器 / 缓冲寄存器	TGRC_0 TGRD_0 TGRF_0	—	—	TGRC_3 TGRD_3	TGRC_4 TGRD_4	—
输入 / 输出引脚	TIOC0A TIOC0B TIOC0C TIOC0D	TIOC1A TIOC1B	TIOC2A TIOC2B	TIOC3A TIOC3B TIOC3C TIOC3D	TIOC4A TIOC4B TIOC4C TIOC4D	输入引脚 TIC5U TIC5V TIC5W
计数器的清除功能	TGR 的比较匹 配或者输入捕捉	TGR 的比较匹 配或者输入捕捉	TGR 的比较匹 配或者输入捕捉	TGR 的比较匹 配或者输入捕捉	TGR 的比较匹 配或者输入捕捉	TGR 的比较匹 配或者输入捕捉
比较匹 配输出	0 输出	○	○	○	○	—
	1 输出	○	○	○	○	—
	交替输出	○	○	○	○	—
输入捕捉功能	○	○	○	○	○	○
同步运行	○	○	○	○	○	—
PWM 模式 1	○	○	○	○	○	—
PWM 模式 2	○	○	○	—	—	—
互补 PWM 模式	—	—	—	○	○	—
复位 PWM 模式	—	—	—	○	○	—
AC 同步马达 驱动模式	○	—	—	○	○	—
相位计数模式	—	○	○	—	—	—
缓冲运行	○	—	—	○	○	—
用于空载时间补偿 的计数器功能	—	—	—	—	—	○
DTC 的启动	TGR 的比较匹 配或者输入捕捉	TGR 的比较匹 配或者输入捕捉	TGR 的比较匹 配或者输入捕捉	TGR 的比较匹 配或者输入捕捉	TGR 的比较匹 配或者输入捕捉 以及 TCNT 的 上溢 / 下溢	TGR 的比较匹 配或者输入捕捉

项目	通道 0	通道 1	通道 2	通道 3	通道 4	通道 5
A/D 转换的开始触发	TGRA_0 的比较匹配或者输入捕捉 TGRE_0 的比较匹配	TGRA_1 的比较匹配或者输入捕捉	TGRA_2 的比较匹配或者输入捕捉	TGRA_3 的比较匹配或者输入捕捉	TGRA_4 的比较匹配或者输入捕捉 互补 PWM 模式中的 TCNT_4 下溢 (波谷)	—
中断源	7 个源 • 比较匹配 / 输入捕捉 0A • 比较匹配 / 输入捕捉 0B • 比较匹配 / 输入捕捉 0C • 比较匹配 / 输入捕捉 0D • 比较匹配 0E • 比较匹配 0F • 上溢	4 个源 • 比较匹配 / 输入捕捉 1A • 比较匹配 / 输入捕捉 1B • 上溢 • 下溢	4 个源 • 比较匹配 / 输入捕捉 2A • 比较匹配 / 输入捕捉 2B • 上溢 • 下溢	5 个源 • 比较匹配 / 输入捕捉 3A • 比较匹配 / 输入捕捉 3B • 比较匹配 / 输入捕捉 3C • 比较匹配 / 输入捕捉 3D • 上溢	5 个源 • 比较匹配 / 输入捕捉 4A • 比较匹配 / 输入捕捉 4B • 比较匹配 / 输入捕捉 4C • 比较匹配 / 输入捕捉 4D • 上溢 / 下溢	3 个源 • 比较匹配 / 输入捕捉 5U • 比较匹配 / 输入捕捉 5V • 比较匹配 / 输入捕捉 5W
A/D 转换的开始请求延迟功能	—	—	—	—	• TADCORA_4 和 TCNT_4 相同时, 请求 A/D 转换的开始 • TADCORB_4 和 TCNT_4 相同时, 请求 A/D 转换的开始	—
中断减少功能	—	—	—	• 减少 TGRA_3 的比较匹配中断	• 减少 TCIV_4 的中断	—

【符号说明】

○: 能

—: 不能

MTU2 的框图如图 10.1 所示。

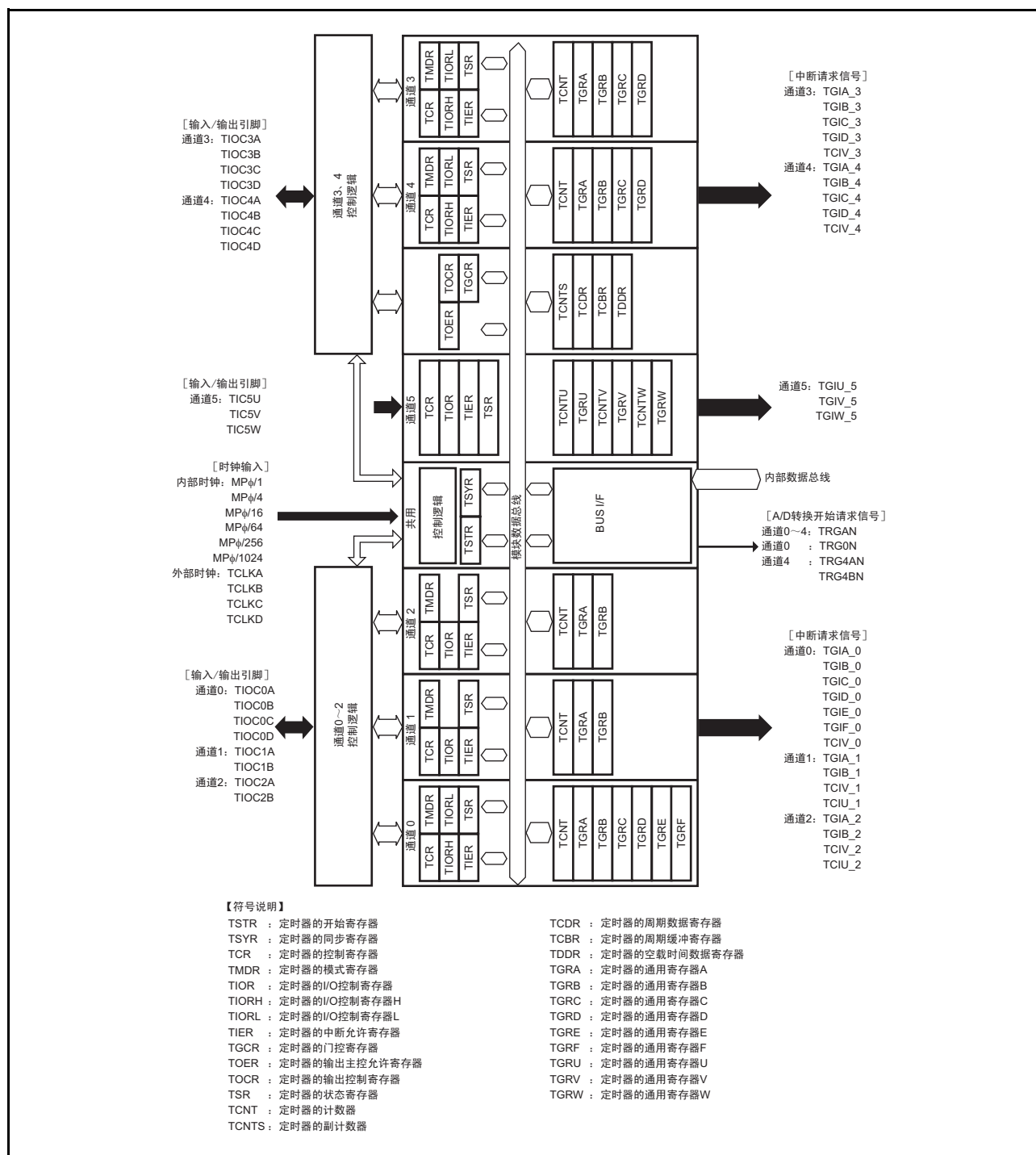


图 10.1 MTU2 的框图

10.2 输入 / 输出引脚

表 10.2 引脚结构

通道	引脚名	输入 / 输出	功能
通用	TCLKA	输入	是外部时钟 A 的输入引脚 (通道 1 的相位计数模式的 A 相输入)。
	TCLKB	输入	是外部时钟 B 输入引脚 (通道 1 的相位计数模式的 B 相输入)。
	TCLKC	输入	是外部时钟 C 的输入引脚 (通道 2 的相位计数模式的 A 相输入)。
	TCLKD	输入	是外部时钟 D 的输入引脚 (通道 2 的相位计数模式的 B 相输入)。
0	TIOC0A	输入 / 输出	是 TGRA_0 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚。
	TIOC0B	输入 / 输出	是 TGRB_0 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚。
	TIOC0C	输入 / 输出	是 TGRC_0 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚。
	TIOC0D	输入 / 输出	是 TGRD_0 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚。
1	TIOC1A	输入 / 输出	是 TGRA_1 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚。
	TIOC1B	输入 / 输出	是 TGRB_1 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚。
2	TIOC2A	输入 / 输出	是 TGRA_2 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚。
	TIOC2B	输入 / 输出	是 TGRB_2 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚。
3	TIOC3A	输入 / 输出	是 TGRA_3 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚。
	TIOC3B	输入 / 输出	是 TGRB_3 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚。
	TIOC3C	输入 / 输出	是 TGRC_3 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚。
	TIOC3D	输入 / 输出	是 TGRD_3 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚。
4	TIOC4A	输入 / 输出	是 TGRA_4 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚。
	TIOC4B	输入 / 输出	是 TGRB_4 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚。
	TIOC4C	输入 / 输出	是 TGRC_4 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚。
	TIOC4D	输入 / 输出	是 TGRD_4 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 输出引脚。
5	TIC5U	输入	是 TGRU_5 的输入捕捉的输入引脚 / 外部脉冲输入引脚。
	TIC5V	输入	是 TGRV_5 的输入捕捉的输入引脚 / 外部脉冲输入引脚。
	TIC5W	输入	是 TGRW_5 的输入捕捉的输入引脚 / 外部脉冲输入引脚。

10.3 寄存器说明

MTU2 的各通道有以下寄存器。有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 25 章 寄存器一览”。对于各通道的寄存器名，将通道 0 的 TCR 记载为 TCR_0。

表 10.3 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
定时器的控制寄存器_3	TCR_3	R/W	H'00	H'FFFFC200	8、16、32
定时器的控制寄存器_4	TCR_4	R/W	H'00	H'FFFFC201	8
定时器的模式寄存器_3	TMDR_3	R/W	H'00	H'FFFFC202	8、16
定时器的模式寄存器_4	TMDR_4	R/W	H'00	H'FFFFC203	8
定时器的 I/O 控制寄存器 H_3	TIORH_3	R/W	H'00	H'FFFFC204	8、16、32
定时器的 I/O 控制寄存器 L_3	TIORL_3	R/W	H'00	H'FFFFC205	8
定时器的 I/O 控制寄存器 H_4	TIORH_4	R/W	H'00	H'FFFFC206	8、16
定时器的 I/O 控制寄存器 L_4	TIORL_4	R/W	H'00	H'FFFFC207	8
定时器的中断允许寄存器_3	TIER_3	R/W	H'00	H'FFFFC208	8、16
定时器的中断允许寄存器_4	TIER_4	R/W	H'00	H'FFFFC209	8
定时器的输出主控允许寄存器	TOER	R/W	H'C0	H'FFFFC20A	8
定时器的门控寄存器	TGCR	R/W	H'80	H'FFFFC20D	8
定时器的输出控制寄存器 1	TOCR1	R/W	H'00	H'FFFFC20E	8、16
定时器的输出控制寄存器 2	TOCR2	R/W	H'00	H'FFFFC20F	8
定时器的计数器_3	TCNT_3	R/W	H'0000	H'FFFFC210	16、32
定时器的计数器_4	TCNT_4	R/W	H'0000	H'FFFFC212	16
定时器的周期数据寄存器	TCDR	R/W	H'FFFF	H'FFFFC214	16、32
定时器的空载时间数据寄存器	TDDR	R/W	H'FFFF	H'FFFFC216	16
定时器的通用寄存器 A_3	TGRA_3	R/W	H'FFFF	H'FFFFC218	16、32
定时器的通用寄存器 B_3	TGRB_3	R/W	H'FFFF	H'FFFFC21A	16
定时器的通用寄存器 A_4	TGRA_4	R/W	H'FFFF	H'FFFFC21C	16、32
定时器的通用寄存器 B_4	TGRB_4	R/W	H'FFFF	H'FFFFC21E	16
定时器的子计数器	TCNTS	R	H'0000	H'FFFFC220	16、32
定时器的周期缓冲寄存器	TCBR	R/W	H'FFFF	H'FFFFC222	16
定时器的通用寄存器 C_3	TGRC_3	R/W	H'FFFF	H'FFFFC224	16、32
定时器的通用寄存器 D_3	TGRD_3	R/W	H'FFFF	H'FFFFC226	16
定时器的通用寄存器 C_4	TGRC_4	R/W	H'FFFF	H'FFFFC228	16、32
定时器的通用寄存器 D_4	TGRD_4	R/W	H'FFFF	H'FFFFC22A	16
定时器的状态寄存器_3	TSR_3	R/W	H'C0	H'FFFFC22C	8、16
定时器的状态寄存器_4	TSR_4	R/W	H'C0	H'FFFFC22D	8
定时器的中断减少设定寄存器	TITCR	R/W	H'00	H'FFFFC230	8、16
定时器的中断减少次数计数器	TITCNT	R	H'00	H'FFFFC231	8
定时器的缓冲传送设定寄存器	TBTER	R/W	H'00	H'FFFFC232	8
定时器的空载时间允许寄存器	TDER	R/W	H'01	H'FFFFC234	8
定时器的输出电平缓冲寄存器	TOLBR	R/W	H'00	H'FFFFC236	8
定时器的缓冲运行传送模式寄存器_3	TBTM_3	R/W	H'00	H'FFFFC238	8、16

寄存器名	略称	R/W	初始值	地址	存取长度
定时器的缓冲运行传送模式寄存器 _4	TBTM_4	R/W	H'00	H'FFFFC239	8
定时器的 A/D 转换开始请求的控制寄存器	TADCR	R/W	H'0000	H'FFFFC240	16
定时器的 A/D 转换开始请求的周期设定寄存器 A_4	TADCORA_4	R/W	H'FFFF	H'FFFFC244	16、32
定时器的 A/D 转换开始请求的周期设定寄存器 B_4	TADCORB_4	R/W	H'FFFF	H'FFFFC246	16
定时器的 A/D 转换开始请求的周期设定缓冲寄存器 A_4	TADCOBRA_4	R/W	H'FFFF	H'FFFFC248	16、32
定时器的 A/D 转换开始请求的周期设定缓冲寄存器 B_4	TADCOBRB_4	R/W	H'FFFF	H'FFFFC24A	16
定时器的波形控制寄存器	TWCR	R/W	H'00	H'FFFFC260	8
定时器的开始寄存器	TSTR	R/W	H'00	H'FFFFC280	8、16
定时器的同步寄存器	TSYR	R/W	H'00	H'FFFFC281	8
定时器的计数器同步开始寄存器	TCSYSTR	R/W	H'00	H'FFFFC282	8
定时器的读写允许寄存器	TRWER	R/W	H'01	H'FFFFC284	8
定时器的控制寄存器 _0	TCR_0	R/W	H'00	H'FFFFC300	8、16、32
定时器的模式寄存器 _0	TMDR_0	R/W	H'00	H'FFFFC301	8
定时器的 I/O 控制寄存器 H_0	TIORH_0	R/W	H'00	H'FFFFC302	8、16
定时器的 I/O 控制寄存器 L_0	TIORL_0	R/W	H'00	H'FFFFC303	8
定时器的中断允许寄存器 _0	TIER_0	R/W	H'00	H'FFFFC304	8、16、32
定时器的状态寄存器 _0	TSR_0	R/W	H'C0	H'FFFFC305	8
定时器的计数器 _0	TCNT_0	R/W	H'0000	H'FFFFC306	16
定时器的通用寄存器 A_0	TGRA_0	R/W	H'FFFF	H'FFFFC308	16、32
定时器的通用寄存器 B_0	TGRB_0	R/W	H'FFFF	H'FFFFC30A	16
定时器的通用寄存器 C_0	TGRC_0	R/W	H'FFFF	H'FFFFC30C	16、32
定时器的通用寄存器 D_0	TGRD_0	R/W	H'FFFF	H'FFFFC30E	16
定时器的通用寄存器 E_0	TGRE_0	R/W	H'FFFF	H'FFFFC320	16、32
定时器的通用寄存器 F_0	TGRF_0	R/W	H'FFFF	H'FFFFC322	16
定时器的中断允许寄存器 2_0	TIER2_0	R/W	H'00	H'FFFFC324	8、16
定时器的状态寄存器 2_0	TSR2_0	R/W	H'C0	H'FFFFC325	8
定时器的缓冲运行传送模式寄存器 _0	TBTM_0	R/W	H'00	H'FFFFC326	8
定时器的控制寄存器 _1	TCR_1	R/W	H'00	H'FFFFC380	8、16
定时器的模式寄存器 _1	TMDR_1	R/W	H'00	H'FFFFC381	8
定时器的 I/O 控制寄存器 _1	TIOR_1	R/W	H'00	H'FFFFC382	8
定时器的中断允许寄存器 _1	TIER_1	R/W	H'00	H'FFFFC384	8、16、32
定时器的状态寄存器 _1	TSR_1	R/W	H'C0	H'FFFFC385	8
定时器的计数器 _1	TCNT_1	R/W	H'0000	H'FFFFC386	16
定时器的通用寄存器 A_1	TGRA_1	R/W	H'FFFF	H'FFFFC388	16、32
定时器的通用寄存器 B_1	TGRB_1	R/W	H'FFFF	H'FFFFC38A	16
定时器的输入捕捉控制寄存器	TICCR	R/W	H'00	H'FFFFC390	8
定时器的控制寄存器 _2	TCR_2	R/W	H'00	H'FFFFC400	8、16

寄存器名	略称	R/W	初始值	地址	存取长度
定时器的模式寄存器 _2	TMDR_2	R/W	H'00	H'FFFFC401	8
定时器的 I/O 控制寄存器 _2	TIOR_2	R/W	H'00	H'FFFFC402	8
定时器的中断允许寄存器 _2	TIER_2	R/W	H'00	H'FFFFC404	8、16、32
定时器的状态寄存器 _2	TSR_2	R/W	H'C0	H'FFFFC405	8
定时器的计数器 _2	TCNT_2	R/W	H'0000	H'FFFFC406	16
定时器的通用寄存器 A_2	TGRA_2	R/W	H'FFFF	H'FFFFC408	16、32
定时器的通用寄存器 B_2	TGRB_2	R/W	H'FFFF	H'FFFFC40A	16
定时器的计数器 U_5	TCNTU_5	R/W	H'0000	H'FFFFC480	16、32
定时器的通用寄存器 U_5	TGRU_5	R/W	H'FFFF	H'FFFFC482	16
定时器的控制寄存器 U_5	TCRU_5	R/W	H'00	H'FFFFC484	8
定时器的 I/O 控制寄存器 U_5	TIORU_5	R/W	H'00	H'FFFFC486	8
定时器的计数器 V_5	TCNTV_5	R/W	H'0000	H'FFFFC490	16、32
定时器的通用寄存器 V_5	TGRV_5	R/W	H'FFFF	H'FFFFC492	16
定时器的控制寄存器 V_5	TCRV_5	R/W	H'00	H'FFFFC494	8
定时器的 I/O 控制寄存器 V_5	TIORV_5	R/W	H'00	H'FFFFC496	8
定时器的计数器 W_5	TCNTW_5	R/W	H'0000	H'FFFFC4A0	16、32
定时器的通用寄存器 W_5	TGRW_5	R/W	H'FFFF	H'FFFFC4A2	16
定时器的控制寄存器 W_5	TCRW_5	R/W	H'00	H'FFFFC4A4	8
定时器的 I/O 控制寄存器 W_5	TIORW_5	R/W	H'00	H'FFFFC4A6	8
定时器的状态寄存器 _5	TSR_5	R/W	H'00	H'FFFFC4B0	8
定时器的中断允许寄存器 _5	TIER_5	R/W	H'00	H'FFFFC4B2	8
定时器的开始寄存器 _5	TSTR_5	R/W	H'00	H'FFFFC4B4	8
定时器的比较匹配清除寄存器	TCNTCMPCLR	R/W	H'00	H'FFFFC4B6	8

10.3.1 定时器的控制寄存器 (TCR)

TCR 是控制各通道 TCNT 的 8 位可读写寄存器。MTU2 的通道 0 ~ 4 各有 1 个，通道 5 有 3 个 TCRU/V/W_5，共计 8 个 TCR。必须在 TCNT 运行停止的状态下进行 TCR 的设定。

位:	7	6	5	4	3	2	1	0
	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7 ~ 5	CCLR[2:0]	000	R/W	计数器的清除 2、1、0 选择 TCNT 的计数器清除源。详细内容请参照表 10.4 和表 10.5。
4、3	CKEG[1:0]	00	R/W	时钟边沿 1、0 选择输入时钟的边沿。如果在双边沿对内部时钟进行计数，输入时钟的周期就为 1/2（例如：MP ϕ /4 的双边沿 = MP ϕ /2 的上升沿）。在通道 1、2 使用相位计数模式时，忽略此设定，而优先设定相位计数模式。内部时钟的边沿选择在输入时钟为 MP ϕ /4 或者慢于 MP ϕ /4 时有效。当选择 MP ϕ /1 或者其他通道的上溢 / 下溢作为输入时钟时，能写值，但是在运行时为初始值。 00: 在上升沿进行计数 01: 在下降沿进行计数 1x: 在双边沿进行计数
2 ~ 0	TPSC[2:0]	000	R/W	定时器的预分频器 2、1、0 选择 TCNT 的计数器时钟，各通道能独立选择时钟源。详细内容请参照表 10.6 ~ 表 10.10。

【符号说明】 x: Don't care

表 10.4 CCLR2 ~ CCLR0 (通道 0、3、4)

通道	bit7	bit6	bit5	说明
	CCLR2	CCLR1	CCLR0	
0、3、4	0	0	0	禁止 TCNT 的清除。
	0	0	1	通过 TGRA 的比较匹配 / 输入捕捉进行 TCNT 的清除。
	0	1	0	通过 TGRB 的比较匹配 / 输入捕捉进行 TCNT 的清除。
	0	1	1	通过同步清除 / 同步运行的其他通道的计数器清除进行 TCNT 的清除 *1。
	1	0	0	禁止 TCNT 的清除。
	1	0	1	通过 TGRC 的比较匹配 / 输入捕捉进行 TCNT 的清除 *2。
	1	1	0	通过 TGRD 的比较匹配 / 输入捕捉进行 TCNT 的清除 *2。
	1	1	1	通过同步清除 / 同步运行的其他通道的计数器清除进行 TCNT 的清除 *1。

【注】 *1 通过将 TSYR 的 SYNC 位置 1 进行同步运行的设定。

*2 在将 TGRC 或者 TGRD 用作缓冲寄存器时，优先对缓冲寄存器进行设定。此时，因为不发生比较匹配 / 输入捕捉，所以不清除 TCNT。

表 10.5 CCLR2 ~ CCLR0 (通道 1、2)

通道	bit7	bit6	bit5	说明
	保留 *2	CCLR1	CCLR0	
1、2	0	0	0	禁止 TCNT 的清除。
	0	0	1	通过 TGRA 的比较匹配 / 输入捕捉进行 TCNT 的清除。
	0	1	0	通过 TGRB 的比较匹配 / 输入捕捉进行 TCNT 的清除。
	0	1	1	通过同步清除 / 同步运行的其他通道的计数器清除进行 TCNT 的清除 *1。

【注】 *1 通过将 TSYR 的 SYNC 位置 1 进行同步运行的设定。

*2 在通道 1、2 中，bit7 为保留位。读取值总是 0，写操作无效。

表 10.6 TPSC2 ~ TPSC0 (通道 0)

通道	bit2	bit1	bit0	说明
	TPSC2	TPSC1	TPSC0	
0	0	0	0	内部时钟：通过 MP ϕ /1 进行计数。
	0	0	1	内部时钟：通过 MP ϕ /4 进行计数。
	0	1	0	内部时钟：通过 MP ϕ /16 进行计数。
	0	1	1	内部时钟：通过 MP ϕ /64 进行计数。
	1	0	0	外部时钟：通过 TCLKA 引脚的输入进行计数。
	1	0	1	外部时钟：通过 TCLKB 引脚的输入进行计数。
	1	1	0	外部时钟：通过 TCLKC 引脚的输入进行计数。
	1	1	1	外部时钟：通过 TCLKD 引脚的输入进行计数。

表 10.7 TPSC2 ~ TPSC0 (通道 1)

通道	bit2	bit1	bit0	说明
	TPSC2	TPSC1	TPSC0	
1	0	0	0	内部时钟：通过 MP ϕ /1 进行计数。
	0	0	1	内部时钟：通过 MP ϕ /4 进行计数。
	0	1	0	内部时钟：通过 MP ϕ /16 进行计数。
	0	1	1	内部时钟：通过 MP ϕ /64 进行计数。
	1	0	0	外部时钟：通过 TCLKA 引脚的输入进行计数。
	1	0	1	外部时钟：通过 TCLKB 引脚的输入进行计数。
	1	1	0	内部时钟：通过 MP ϕ /256 进行计数。
	1	1	1	通过 TCNT_2 的上溢 / 下溢进行计数。

【注】 在通道 1 为相位计数模式时，此设定无效。

表 10.8 TPSC2 ~ TPSC0 (通道 2)

通道	bit2	bit1	bit0	说明
	TPSC2	TPSC1	TPSC0	
2	0	0	0	内部时钟: 通过 MP ϕ /1 进行计数。
	0	0	1	内部时钟: 通过 MP ϕ /4 进行计数。
	0	1	0	内部时钟: 通过 MP ϕ /16 进行计数。
	0	1	1	内部时钟: 通过 MP ϕ /64 进行计数。
	1	0	0	外部时钟: 通过 TCLKA 引脚的输入进行计数。
	1	0	1	外部时钟: 通过 TCLKB 引脚的输入进行计数。
	1	1	0	外部时钟: 通过 TCLKC 引脚的输入进行计数。
	1	1	1	内部时钟: 通过 MP ϕ /1024 进行计数。

【注】 在通道 2 为相位计数模式时, 此设定无效。

表 10.9 TPSC2 ~ TPSC0 (通道 3、4)

通道	bit2	bit1	bit0	说明
	TPSC2	TPSC1	TPSC0	
3、4	0	0	0	内部时钟: 通过 MP ϕ /1 进行计数。
	0	0	1	内部时钟: 通过 MP ϕ /4 进行计数。
	0	1	0	内部时钟: 通过 MP ϕ /16 进行计数。
	0	1	1	内部时钟: 通过 MP ϕ /64 进行计数。
	1	0	0	内部时钟: 通过 MP ϕ /256 进行计数。
	1	0	1	内部时钟: 通过 MP ϕ /1024 进行计数。
	1	1	0	外部时钟: 通过 TCLKA 引脚的输入进行计数。
	1	1	1	外部时钟: 通过 TCLKB 引脚的输入进行计数。

表 10.10 TPSC1、TPSC0 (通道 5)

通道	bit1	bit0	说明
	TPSC1	TPSC0	
5	0	0	内部时钟: 通过 MP ϕ /1 进行计数。
	0	1	内部时钟: 通过 MP ϕ /4 进行计数。
	1	0	内部时钟: 通过 MP ϕ /16 进行计数。
	1	1	内部时钟: 通过 MP ϕ /64 进行计数。

【注】 在通道 5 中, bit7 ~ 2 为保留位。读写值总是 0。

10.3.2 定时器的模式寄存器（TMDR）

TMDR 是 8 位可读写寄存器，设定各通道的运行模式。MTU2 的通道 0～4 各有 1 个，共计 5 个 TMDR。必须在 TCNT 运行停止的状态下进行 TMDR 的设定。

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

位	位名	初始值	R/W	说明
7	—	0	—	保留位 读写值总是 0。
6	BFE	0	R/W	缓冲运行 E 选择是使 TGRE_0 和 TGRF_0 进行正常运行还是进行缓冲运行。即使将 TGRF 用作缓冲寄存器，也会发生 TGRF 的比较匹配。 在通道 1、2、3、4 中，此位为保留位，读写值总是 0。 0: TGRE_0 和 TGRF_0 进行正常运行 1: TGRE_0 和 TGRF_0 进行缓冲运行
5	BFB	0	R/W	缓冲运行 B 设定是使 TGRB 进行正常运行还是和 TGRD 组合进行缓冲运行。如果将 TGRD 用作缓冲寄存器，除了互补 PWM 模式以外，不发生 TGRD 的输入捕捉 / 输出比较。但是，在互补 PWM 模式中会发生 TGRD 的比较匹配。另外，如果在互补 PWM 模式的 Tb 区间发生比较匹配，TGRD 就被置位，因此必须将定时器的中断允许寄存器 _3/4（TIER_3/4）的 TGIED 位置 0。 在没有 TGRD 的通道 1、2 中，此位为保留位，读写值总是 0。 0: TGRB 和 TGRD 进行正常运行 1: TGRB 和 TGRD 进行缓冲运行
4	BFA	0	R/W	缓冲运行 A 设定是使 TGRA 进行正常运行还是和 TGRC 组合进行缓冲运行。如果将 TGRC 用作缓冲寄存器，除了互补 PWM 模式以外，不发生 TGRC 的输入捕捉 / 输出比较。但是，在互补 PWM 模式中会发生 TGRC 的比较匹配。另外，如果在互补 PWM 模式的 Tb 区间发生通道 4 的比较匹配，TGFC 就被置位，因此必须将定时器的中断允许寄存器 _4（TIER_4）的 TGIEC 位置 0。 在没有 TGRC 的通道 1、2 中，此位为保留位，读写值总是 0。 0: TGRA 和 TGRC 进行正常运行 1: TGRA 和 TGRC 进行缓冲运行
3～0	MD[3:0]	0000	R/W	模式 3～0 MD3～MD0 设定定时器的运行模式。 详细内容请参照表 10.11。

表 10.11 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	正常运行
0	0	0	1	禁止设定
0	0	1	0	PWM 模式 1
0	0	1	1	PWM 模式 2*1
0	1	0	0	相位计数模式 1*2
0	1	0	1	相位计数模式 2*2
0	1	1	0	相位计数模式 3*2
0	1	1	1	相位计数模式 4*2
1	0	0	0	复位同步 PWM 模式 *3
1	0	0	1	禁止设定
1	0	1	x	禁止设定
1	1	0	0	禁止设定
1	1	0	1	互补 PWM 模式 1 (在波峰进行传送) *3
1	1	1	0	互补 PWM 模式 2 (在波谷进行传送) *3
1	1	1	1	互补 PWM 模式 3 (在波峰和波谷进行传送) *3

【符号说明】 x: Don't care

【注】 *1 不能给通道 3、4 设定 PWM 模式 2。

*2 不能给通道 0、3、4 设定相位计数模式。

*3 只有通道 3 能设定复位同步 PWM 模式或者互补 PWM 模式。

如果给通道 3 设定复位同步 PWM 模式或者互补 PWM 模式，通道 4 的设定就无效，自动服从通道 3 的设定。但是，不能给通道 4 设定复位同步 PWM 模式或者互补 PWM 模式。

不能给通道 0、1、2 设定复位同步 PWM 模式或者互补 PWM 模式。

10.3.3 定时器的 I/O 控制寄存器 (TIOR)

TIOR 是控制 TGR 的 8 位可读写寄存器。MTU2 的通道 0、3、4 各有 2 个，通道 1、2 各有 1 个，通道 5 有 3 个 TIORU/V/W_5，共计 11 个 TIOR。

在 TMDR 被设定为正常运行模式、PWM 模式或者相位计数模式时，进行 TIOR 的设定。

TIOR 指定的初始输出在计数器停止 (将 TSTR 的 CST 位清 0) 的状态下有效。另外，在 PWM 模式 2 的情况下，指定计数器清 0 时的输出。

如果将 TGRC 或者 TGRD 设定为缓冲运行，此设定无效，用作缓冲寄存器。

- TIORH_0、TIOR_1、TIOR_2、TIORH_3、TIORH_4

位:	7	6	5	4	3	2	1	0
	IOB[3:0]				IOA[3:0]			
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7 ~ 4	IOB[3:0]	0000	R/W	I/O 控制 B3 ~ B0 IOB3 ~ IOB0 位设定 TGRB 的功能。 请参照下表： TIORH_0: 表 10.12 TIOR_1: 表 10.14 TIOR_2: 表 10.15 TIORH_3: 表 10.16 TIORH_4: 表 10.18
3 ~ 0	IOA[3:0]	0000	R/W	I/O 控制 A3 ~ A0 IOA3 ~ IOA0 位设定 TGRA 的功能。 请参照下表： TIORH_0: 表 10.20 TIOR_1: 表 10.22 TIOR_2: 表 10.23 TIORH_3: 表 10.24 TIORH_4: 表 10.26

- TIORL_0、TIORL_3、TIORL_4

位:	7	6	5	4	3	2	1	0
	IOD[3:0]				IOC[3:0]			
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7 ~ 4	IOD[3:0]	0000	R/W	I/O 控制 D3 ~ D0 IOD3 ~ IOD0 位设定 TGRD 的功能。 请参照下表： TIORL_0: 表 10.13 TIORL_3: 表 10.17 TIORL_4: 表 10.19
3 ~ 0	IOC[3:0]	0000	R/W	I/O 控制 C3 ~ C0 IOC3 ~ IOC0 位设定 TGRC 的功能。 请参照下表： TIORL_0: 表 10.21 TIORL_3: 表 10.25 TIORL_4: 表 10.27

- 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	初始值	R/W	说明
7 ~ 5	—	全 0	R	保留位 读写值总是 0。
4 ~ 0	IOC[4:0]	00000	R/W	I/O 控制 C4 ~ C0 IOC4 ~ IOC0 位设定 TGRU/V/W_5 的功能。 详细内容请参照表 10.28。

表 10.12 TIORH_0 (通道 0)

bit7	bit6	bit5	bit4	说明	
IOB3	IOB2	IOB1	IOB0	TGRB_0 的功能	TIOC0B 引脚的功能
0	0	0	0	输出比较寄存器	保持输出 *
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	0	0	0	输入捕捉寄存器	在上升沿进行输入捕捉。
1	0	0	1		在下降沿进行输入捕捉。
1	0	1	x		在双边沿进行输入捕捉。
1	1	x	x		捕捉输入源为通道 1/ 计数时钟, 通过 TCNT_1 的递增计数 / 递减计数进行输入捕捉。

【符号说明】 x: Don't care

【注】 * 从上电复位后到设定 TIOR 为止, 输出 0。

表 10.13 TIORL_0 (通道 0)

bit7	bit6	bit5	bit4	说明	
IOD3	IOD2	IOD1	IOD0	TGRD_0 的功能	TIOC0D 引脚的功能
0	0	0	0	输出比较寄存器 *2	保持输出 *1
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出

bit7	bit6	bit5	bit4	说明	
IOD3	IOD2	IOD1	IOD0	TGRD_0 的功能	TIOC0D 引脚的功能
1	0	0	0	输入捕捉寄存器 *2	在上升沿进行输入捕捉。
1	0	0	1		在下降沿进行输入捕捉。
1	0	1	x		在双边沿进行输入捕捉。
1	1	x	x		捕捉输入源为通道 1/ 计数时钟, 通过 TCNT_1 的递增计数 / 递减计数进行输入捕捉。

【符号说明】 x: Don't care

【注】 *1 从上电复位后到设定 TIOR 为止, 输出 0。

*2 如果将 TMDR_0 的 BFB 位置 1, TGRD_0 用作缓冲寄存器, 此设定就无效并且不发生输入捕捉 / 输出比较。

表 10.14 TIOR_1 (通道 1)

bit7	bit6	bit5	bit4	说明	
IOB3	IOB2	IOB1	IOB0	TGRB_1 的功能	TIOC1B 引脚的功能
0	0	0	0	输出比较寄存器	保持输出 *
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	0	0	0	输入捕捉寄存器	在上升沿进行输入捕捉。
1	0	0	1		在下降沿进行输入捕捉。
1	0	1	x		在双边沿进行输入捕捉。
1	1	x	x		通过 TGRC_0 的比较匹配 / 输入捕捉进行输入捕捉。

【符号说明】 x: Don't care

【注】 * 从上电复位后到设定 TIOR 为止, 输出 0。

表 10.15 TIOR_2 (通道 2)

bit7	bit6	bit5	bit4	说明	
IOB3	IOB2	IOB1	IOB0	TGRB_2 的功能	TIOC2B 引脚的功能
0	0	0	0	输出比较寄存器	保持输出 *
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	x	0	0	输入捕捉寄存器	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

【符号说明】 x: Don't care

【注】 * 从上电复位后到设定 TIOR 为止, 输出 0。

表 10.16 TIORH_3 (通道 3)

bit7	bit6	bit5	bit4	说明	
IOB3	IOB2	IOB1	IOB0	TGRB_3 的功能	TIOC3B 引脚的功能
0	0	0	0	输出比较寄存器	保持输出 *
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	x	0	0	输入捕捉寄存器	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

【符号说明】 x: Don't care

【注】 * 从上电复位后到设定 TIOR 为止, 输出 0。

表 10.17 TIORL_3 (通道 3)

bit7	bit6	bit5	bit4	说明	
IOD3	IOD2	IOD1	IOD0	TGRD_3 的功能	TIOC3D 引脚的功能
0	0	0	0	输出比较寄存器 *2	保持输出 *1
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	x	0	0	输入捕捉寄存器 *2	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

【符号说明】 x: Don't care

【注】 *1 从上电复位后到设定 TIOR 为止, 输出 0。

*2 如果将 TMDR_3 的 BFB 位置 1, TGRD_3 用作缓冲寄存器, 此设定就无效并且不发生输入捕捉 / 输出比较。

表 10.18 TIORH_4 (通道 4)

bit7	bit6	bit5	bit4	说明	
IOB3	IOB2	IOB1	IOB0	TGRB_4 的功能	TIOC4B 引脚的功能
0	0	0	0	输出比较寄存器	保持输出 *
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出

bit7	bit6	bit5	bit4	说明	
IOB3	IOB2	IOB1	IOB0	TGRB_4 的功能	TIOC4B 引脚的功能
1	x	0	0	输入捕捉寄存器	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

【符号说明】 x: Don't care

【注】 * 从上电复位后到设定 TIOR 为止，输出 0。

表 10.19 TIORL_4 (通道 4)

bit7	bit6	bit5	bit4	说明	
IOD3	IOD2	IOD1	IOD0	TGRD_4 的功能	TIOC4D 引脚的功能
0	0	0	0	输出比较寄存器 *2	保持输出 *1
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	x	0	0	输入捕捉寄存器 *2	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

【符号说明】 x: Don't care

【注】 *1 从上电复位后到设定 TIOR 为止，输出 0。

*2 如果将 TMDR_4 的 BFB 位置 1，TGRD_4 用作缓冲寄存器，此设定就无效并且不发生输入捕捉 / 输出比较。

表 10.20 TIORH_0 (通道 0)

bit3	bit2	bit1	bit0	说明	
IOA3	IOA2	IOA1	IOA0	TGRA_0 的功能	TIOC0A 引脚的功能
0	0	0	0	输出比较寄存器	保持输出 *
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	0	0	0	输入捕捉寄存器	在上升沿进行输入捕捉。
1	0	0	1		在下降沿进行输入捕捉。
1	0	1	x		在双边沿进行输入捕捉。
1	1	x	x		捕捉输入源为通道 1/ 计数时钟, 通过 TCNT_1 的递增计数 / 递减计数进行输入捕捉。

【符号说明】 x: Don't care

【注】 * 从上电复位后到设定 TIOR 为止, 输出 0。

表 10.21 TIORL_0 (通道 0)

bit3	bit2	bit1	bit0	说明	
IOC3	IOC2	IOC1	IOC0	TGRC_0 的功能	TIOC0C 引脚的功能
0	0	0	0	输出比较寄存器 *2	保持输出 *1
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出

bit3	bit2	bit1	bit0	说明	
IOC3	IOC2	IOC1	IOC0	TGRC_0 的功能	TIOC0C 引脚的功能
1	0	0	0	输入捕捉寄存器 *2	在上升沿进行输入捕捉。
1	0	0	1		在下降沿进行输入捕捉。
1	0	1	x		在双边沿进行输入捕捉。
1	1	x	x		捕捉输入源为通道 1/ 计数时钟, 通过 TCNT_1 的递增计数 / 递减计数进行输入捕捉。

【符号说明】 x: Don't care

【注】 *1 从上电复位后到设定 TIOR 为止, 输出 0。

*2 如果将 TMDR_0 的 BFA 位置 1, TGRC_0 用作缓冲寄存器, 此设定就无效并且不发生输入捕捉 / 输出比较。

表 10.22 TIOR_1 (通道 1)

bit3	bit2	bit1	bit0	说明	
IOA3	IOA2	IOA1	IOA0	TGRA_1 的功能	TIOC1A 引脚的功能
0	0	0	0	输出比较寄存器	保持输出 *
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	0	0	0	输入捕捉寄存器	在上升沿进行输入捕捉。
1	0	0	1		在下降沿进行输入捕捉。
1	0	1	x		在双边沿进行输入捕捉。
1	1	x	x		通过 TGRA_0 的比较匹配 / 输入捕捉进行输入捕捉。

【符号说明】 x: Don't care

【注】 * 从上电复位后到设定 TIOR 为止, 输出 0。

表 10.23 TIOR_2 (通道 2)

bit3	bit2	bit1	bit0	说明	
IOA3	IOA2	IOA1	IOA0	TGRA_2 的功能	TIOC2A 引脚的功能
0	0	0	0	输出比较寄存器	保持输出 *
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	x	0	0	输入捕捉寄存器	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

【符号说明】 x: Don't care

【注】 * 从上电复位后到设定 TIOR 为止, 输出 0。

表 10.24 TIORH_3 (通道 3)

bit3	bit2	bit1	bit0	说明	
IOA3	IOA2	IOA1	IOA0	TGRA_3 的功能	TIOC3A 引脚的功能
0	0	0	0	输出比较寄存器	保持输出 *
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	x	0	0	输入捕捉寄存器	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

【符号说明】 x: Don't care

【注】 * 从上电复位后到设定 TIOR 为止, 输出 0。

表 10.25 TIORL_3 (通道 3)

bit3	bit2	bit1	bit0	说明	
IOC3	IOC2	IOC1	IOC0	TGRC_3 的引脚	TIOC3C 引脚的功能
0	0	0	0	输出比较寄存器 *2	保持输出 *1
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	x	0	0	输入捕捉寄存器 *2	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

【符号说明】 x: Don't care

【注】 *1 从上电复位后到设定 TIOR 为止, 输出 0。

*2 如果将 TMDR_3 的 BFA 位置 1, TGRC_3 用作缓冲寄存器, 此设定就无效并且不发生输入捕捉 / 输出比较。

表 10.26 TIORH_4 (通道 4)

bit3	bit2	bit1	bit0	说明	
IOA3	IOA2	IOA1	IOA0	TGRA_4 的功能	TIOC4A 引脚的功能
0	0	0	0	输出比较寄存器	保持输出 *
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出

bit3	bit2	bit1	bit0	说明	
IOA3	IOA2	IOA1	IOA0	TGRA_4 的功能	TIOC4A 引脚的功能
1	x	0	0	输入捕捉寄存器	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

【符号说明】 x: Don't care

【注】 * 从上电复位后到设定 TIOR 为止，输出 0。

表 10.27 TIORL_4 (通道 4)

bit3	bit2	bit1	bit0	说明	
IOC3	IOC2	IOC1	IOC0	TGRC_4 的功能	TIOC4C 引脚的功能
0	0	0	0	输出比较寄存器 *2	保持输出 *1
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	1		初始输出: 0 输出 比较匹配: 交替输出
1	x	0	0	输入捕捉寄存器 *2	在上升沿进行输入捕捉。
1	x	0	1		在下降沿进行输入捕捉。
1	x	1	x		在双边沿进行输入捕捉。

【符号说明】 x: Don't care

【注】 *1 从上电复位后到设定 TIOR 为止，输出 0。

*2 如果将 TMDR_4 的 BFA 位置 1，TGRC_4 用作缓冲寄存器，此设定就无效并且不发生输入捕捉 / 输出比较。

表 10.28 TIORU_5、TIO RV_5、TIO RW_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	输出比较寄存器	比较匹配
0	0	0	0	1		禁止设定
0	0	0	1	x		禁止设定
0	0	1	x	x		禁止设定
0	1	x	x	x		禁止设定
1	0	0	0	0	输入捕捉寄存器	禁止设定
1	0	0	0	1		在上升沿进行输入捕捉。
1	0	0	1	0		在下降沿进行输入捕捉。
1	0	0	1	1		在双边沿进行输入捕捉。
1	0	1	x	x		禁止设定
1	1	0	0	0		禁止设定
1	1	0	0	1		用于测量外部输入信号的低电平脉宽，在互补 PWM 模式的波谷进行捕捉。
1	1	0	1	0		用于测量外部输入信号的低电平脉宽，在互补 PWM 模式的波峰进行捕捉。
1	1	0	1	1		用于测量外部输入信号的低电平脉宽，在互补 PWM 模式的波峰和波谷进行捕捉。
1	1	1	0	0		禁止设定
1	1	1	0	1		用于测量外部输入信号的高电平脉宽，在互补 PWM 模式的波谷进行捕捉。
1	1	1	1	0		用于测量外部输入信号的高电平脉宽，在互补 PWM 模式的波峰进行捕捉。
1	1	1	1	1		用于测量外部输入信号的高电平脉宽，在互补 PWM 模式的波峰和波谷进行捕捉。

【符号说明】 x: Don't care

10.3.4 定时器的比较匹配清除寄存器（TCNTCMPCLR）

TCNTCMPCLR 是 8 位可读写寄存器，能设定 TCNTU_5、TCNTV_5 和 TCNTW_5 的清除请求。MTU2 的通道 5 有 1 个 TCNTCMPCLR。

位:	7	6	5	4	3	2	1	0
	—	—	—	—	—	CMP CLR5U	CMP CLR5V	CMP CLR5W
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7 ~ 3	—	全 0	R	保留位 读写值总是 0。
2	CMPCLR5U	0	R/W	TCNT 比较清除 5U 允许或者禁止通过 TCNTU_5 和 TGRU_5 的比较匹配 / 输入捕捉产生的 TCNTU_5 的清除请求。 0: 禁止通过 TCNTU_5 和 TGRU_5 的比较匹配 / 输入捕捉将 TCNTU_5 清 0 1: 允许通过 TCNTU_5 和 TGRU_5 的比较匹配 / 输入捕捉将 TCNTU_5 清 0
1	CMPCLR5V	0	R/W	TCNT 比较清除 5V 允许或者禁止通过 TCNTV_5 和 TGRV_5 的比较匹配 / 输入捕捉产生的 TCNTV_5 的清除请求。 0: 禁止通过 TCNTV_5 和 TGRV_5 的比较匹配 / 输入捕捉将 TCNTV_5 清 0 1: 允许通过 TCNTV_5 和 TGRV_5 的比较匹配 / 输入捕捉将 TCNTV_5 清 0
0	CMPCLR5W	0	R/W	TCNT 比较清除 5W 允许或者禁止通过 TCNTW_5 和 TGRW_5 的比较匹配 / 输入捕捉产生的 TCNTW_5 的清除请求。 0: 禁止通过 TCNTW_5 和 TGRW_5 的比较匹配 / 输入捕捉将 TCNTW_5 清 0 1: 允许通过 TCNTW_5 和 TGRW_5 的比较匹配 / 输入捕捉将 TCNTW_5 清 0

10.3.5 定时器的中断允许寄存器（TIER）

TIER 是 8 位可读写寄存器，允许或者禁止各通道的中断请求。MTU2 的通道 0 有 2 个，通道 1 ~ 5 各有 1 个，共计 7 个 TIER。

- TIER_0、TIER_1、TIER_2、TIER_3、TIER_4

位:	7	6	5	4	3	2	1	0
	TTGE	TTGE2	TCIEU	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7	TTGE	0	R/W	A/D 转换开始请求的允许 允许或者禁止通过 TGRA 的输入捕捉 / 比较匹配产生的 A/D 转换器开始请求。 0: 禁止 A/D 转换开始请求的产生 1: 允许 A/D 转换开始请求的产生
6	TTGE2	0	R/W	A/D 转换开始请求的允许 2 在互补 PWM 模式中, 允许或者禁止通过 TCNT_4 的下溢 (波谷) 产生的 A/D 转换请求。 在通道 0 ~ 3 中, 此位为保留位, 读写值总是 0。 0: 禁止通过 TCNT_4 的下溢 (波谷) 产生的 A/D 转换请求 1: 允许通过 TCNT_4 的下溢 (波谷) 产生的 A/D 转换请求
5	TCIEU	0	R/W	下溢的中断允许 在通道 1、2 中, 当 TSR 的 TCFU 标志被置 1 时, 允许或者禁止 TCFU 标志的中断请求 (TCIU)。 在通道 0、3、4 中, 此位为保留位, 读写值总是 0。 0: 禁止 TCFU 的中断请求 (TCIU) 1: 允许 TCFU 的中断请求 (TCIU)
4	TCIEV	0	R/W	上溢的中断允许 当 TSR 的 TCFV 标志被置 1 时, 允许或者禁止 TCFV 标志的中断请求 (TCIV)。 0: 禁止 TCFV 的中断请求 (TCIV) 1: 允许 TCFV 的中断请求 (TCIV)
3	TGIED	0	R/W	TGR 的中断允许 D 在通道 0、3、4 中, 当 TSR 的 TGFD 位被置 1 时, 允许或者禁止 TGFD 位的中断请求 (TGID)。 在通道 1、2 中, 此位为保留位, 读写值总是 0。 0: 禁止 TGFD 位的中断请求 (TGID) 1: 允许 TGFD 位的中断请求 (TGID)
2	TGIEC	0	R/W	TGR 的中断允许 C 在通道 0、3、4 中, 当 TSR 的 TGFC 位被置 1 时, 允许或者禁止 TGFC 位的中断请求 (TGIC)。 在通道 1、2 中, 此位为保留位, 读写值总是 0。 0: 禁止 TGFC 位的中断请求 (TGIC) 1: 允许 TGFC 位的中断请求 (TGIC)
1	TGIEB	0	R/W	TGR 的中断允许 B 当 TSR 的 TGFB 位被置 1 时, 允许或者禁止 TGFB 位的中断请求 (TGIB)。 0: 禁止 TGFB 位的中断请求 (TGIB) 1: 允许 TGFB 位的中断请求 (TGIB)
0	TGIEA	0	R/W	TGR 的中断允许 A 当 TSR 的 TGFA 位被置 1 时, 允许或者禁止 TGFA 位的中断请求 (TGIA)。 0: 禁止 TGFA 位的中断请求 (TGIA) 1: 允许 TGFA 位的中断请求 (TGIA)

• TIER2_0

位:	7	6	5	4	3	2	1	0
	TTGE2	—	—	—	—	—	TGIEF	TGIEE
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R/W	R/W

位	位名	初始值	R/W	说明
7	TTGE2	0	R/W	A/D 转换开始请求的允许 2 允许或者禁止通过 TCNT_0 和 TGRE_0 的比较匹配产生的 A/D 转换开始请求。 0: 禁止通过 TCNT_0 和 TGRE_0 的比较匹配产生的 A/D 转换开始请求 1: 允许通过 TCNT_0 和 TGRE_0 的比较匹配产生的 A/D 转换开始请求
6 ~ 2	—	全 0	R	保留位 读写值总是 0。
1	TGIEF	0	R/W	TGR 中断的允许 F 允许或者禁止通过 TCNT_0 和 TGRF_0 的比较匹配产生的中断请求。 0: 禁止 TGFE 位的中断请求 (TGIF) 1: 允许 TGFE 位的中断请求 (TGIF)
0	TGIEE	0	R/W	TGR 中断的允许 E 允许或者禁止通过 TCNT_0 和 TGRE_0 的比较匹配产生的中断请求。 0: 禁止 TGEE 位的中断请求 (TGIE) 1: 允许 TGEE 位的中断请求 (TGIE)

• TIER_5

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

位	位名	初始值	R/W	说明
7 ~ 3	—	全 0	R	保留位 读写值总是 0。
2	TGIE5U	0	R/W	TGR 中断的允许 5U 当 TSR_5 的 CMFU5 位被置 1 时, 允许或者禁止通过 CMFU5 位产生的 (TGIU_5) 中断请求。 0: 禁止 TGIU_5 的中断请求 1: 允许 TGIU_5 的中断请求
1	TGIE5V	0	R/W	TGR 中断的允许 5V 当 TSR_5 的 CMFV5 位被置 1 时, 允许或者禁止通过 CMFV5 位产生的 (TGIV_5) 中断请求。 0: 禁止 TGIV_5 的中断请求 1: 允许 TGIV_5 的中断请求

位	位名	初始值	R/W	说明
0	TGIE5W	0	R/W	TGR 中断的允许 5W 当 TSR_5 的 CMFW5 位被置 1 时，允许或者禁止通过 CMFW5 位产生的 (TGIW_5) 中断请求。 0: 禁止 TGIW_5 的中断请求 1: 允许 TGIW_5 的中断请求

10.3.6 定时器的状态寄存器 (TSR)

TSR 是 8 位可读写寄存器，表示各通道的状态。MTU2 的通道 0 有 2 个，通道 1～5 各有 1 个，共计 7 个 TSR。

- TSR_0、TSR_1、TSR_2、TSR_3、TSR_4

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

【注】*1 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说明
7	TCFD	1	R	计数方向标志 是表示通道 1～4 的 TCNT 计数方向的状态标志。 在通道 0 中，此位为保留位，读写值总是 1。 0: TCNT 进行递减计数 1: TCNT 进行递增计数
6	—	1	R	保留位 读写值总是 1。
5	TCFU	0	R(W)*1	下溢标志 是表示通道 1、2 为相位计数模式时的 TCNT 下溢的状态标志。为了清除标志，只能写 0。 在通道 0、3、4 中，此位为保留位，读写值总是 0。 [置位条件] • 当 TCNT 的值发生下溢 (H'0000→H'FFFF) 时 [清除条件] • 在 TCFU=1 的状态下，当读 TCFU 后给 TCFU 写 0 时 *2
4	TCFV	0	R(W)*1	上溢标志 是表示 TCNT 上溢的状态标志。为了清除标志，只能写 0。 [置位条件] • 当 TCNT 的值发生上溢 (H'FFFF→H'0000) 时 对于通道 4，在互补 PWM 模式中，当 TCNT_4 的值发生下溢 (H'0001→H'0000) 时，此标志也被置位。 [清除条件] • 在 TCFV=1 的状态下，当读 TCFV 后给 TCFV 写 0 时 *2 对于通道 4，在通过 TCIV 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位为 0 时，清除此标志。

位	位名	初始值	R/W	说明
3	TGFD	0	R/(W)*1	输入捕捉 / 输出比较标志 D 是表示通道 0、3、4 的 TGRD 输入捕捉或者比较匹配的状态标志。为了清除标志，只能写 0。在通道 1、2 中，此位为保留位，读写值总是 0。 [置位条件] - 在 TGRD 用作输出比较寄存器的情况下，当 TCNT=TGRD 时 - 在 TGRD 用作输入捕捉的情况下，当通过输入捕捉信号将 TCNT 的值传送到 TGRD 时 [清除条件] - 在通过 TGID 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位为 0 时 - 在 TGFD=1 的状态下，当读 TGFD 后给 TGFD 写 0 时 *2
2	TGFC	0	R/(W)*1	输入捕捉 / 输出比较标志 C 是表示通道 0、3、4 的 TGRC 输入捕捉或者比较匹配的状态标志。为了清除标志，只能写 0。在通道 1、2 中，此位为保留位，读写值总是 0。 [置位条件] - 在 TGRC 用作输出比较寄存器的情况下，当 TCNT=TGRC 时 - 在 TGRC 用作输入捕捉的情况下，当通过输入捕捉信号将 TCNT 的值传送到 TGRC 时 [清除条件] - 在通过 TGIC 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位为 0 时 - 在 TGFC=1 的状态下，当读 TGFC 后给 TGFC 写 0 时 *2
1	TGFB	0	R/(W)*1	输入捕捉 / 输出比较标志 B 是表示 TGRB 的输入捕捉或者比较匹配的状态标志。为了清除标志，只能写 0。 [置位条件] - 在 TGRB 用作输出比较寄存器的情况下，当 TCNT=TGRB 时 - 在 TGRB 用作输入捕捉的情况下，当通过输入捕捉信号将 TCNT 的值传送到 TGRB 时 [清除条件] - 在通过 TGIB 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位为 0 时 - 在 TGFB=1 的状态下，当读 TGFB 后给 TGFB 写 0 时 *2
0	TGFA	0	R/(W)*1	输入捕捉 / 输出比较标志 A 是表示 TGRA 的输入捕捉或者比较匹配的状态标志。为了清除标志，只能写 0。 [置位条件] - 在 TGRA 用作输出比较寄存器的情况下，当 TCNT=TGRA 时 - 在 TGRA 用作输入捕捉的情况下，当通过输入捕捉信号将 TCNT 的值传送到 TGRA [清除条件] - 在通过 TGIA 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位为 0 时 - 在 TGFA=1 的状态下，当读 TGFA 后给 TGFA 写 0 时 *2

【注】 *1 为了清除标志，只能在读 1 后写 0。

*2 如果在读 1 后写 0 前下一个标志被置位，即使写 0 也不能清除标志，所以必须重新读 1 后再写 0。

• TSR2_0

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

【注】*1 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说明
7、6	—	全 1	R	保留位 读写值总是 1。
5 ~ 2	—	全 0	R	保留位 读写值总是 0。
1	TGFF	0	R/(W)*1	比较匹配标志 F 是表示 TCNT_0 和 TGRF_0 的比较匹配的状态标志。 [置位条件] • 在 TGRF_0 用作比较寄存器的情况下，当 TCNT_0=TGRF_0 时 [清除条件] • 在 TGFF=1 的状态下，当读 TGFF 后给 TGFF 写 0 时 *2
0	TGFE	0	R/(W)*1	比较匹配标志 E 是表示 TCNT_0 和 TGRE_0 的比较匹配的状态标志。 [置位条件] • 在 TGRE_0 用作比较寄存器的情况下，当 TCNT_0=TGRE_0 时 [清除条件] • 在 TGFE=1 的状态下，当读 TGFE 后给 TGFE 写 0 时 *2

【注】 *1 为了清除标志，只能在读 1 后写 0。

*2 如果在读 1 后写 0 前下一个比较匹配标志被置位，即使写 0 也不能清除标志，所以必须重新读 1 后再写 0。

• TSR_5

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

【注】*1 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说明
7 ~ 3	—	全 0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
2	CMFU5	0	R/(W)*1	比较匹配 / 输入捕捉标志 U5 是表示 TGRU_5 的输入捕捉或者比较匹配的状态标志。为了清除标志，只能写 0。 [置位条件] - 在 TGRU_5 用作比较匹配寄存器的情况下，当 TCNTU_5 = TGRU_5 时 - 在 TGRU_5 用作输入捕捉的情况下，当通过输入捕捉信号将 TCNTU_5 的值传送到 TGRU_5 时 - 在 TGRU_5 用于测量外部输入信号脉宽的情况下，当将 TCNTU_5 的值传送到 TGRU_5 时 *2 [清除条件] - 在通过 TGIU_5 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位为 0 时 - 在 CMFU5=1 的状态下，当读 CMFU5 后给 CMFU5 写 0 时
1	CMFV5	0	R/(W)*1	比较匹配 / 输入捕捉标志 V5 是表示 TGRV_5 的输入捕捉或者比较匹配的状态标志。为了清除标志，只能写 0。 [置位条件] - 在 TGRV_5 用作比较匹配寄存器的情况下，当 TCNTV_5 = TGRV_5 时 - 在 TGRV_5 用作输入捕捉的情况下，当通过输入捕捉信号将 TCNTV_5 的值传送到 TGRV_5 时 - 在 TGRV_5 用于测量外部输入信号脉宽的情况下，当将 TCNTV_5 的值传送到 TGRV_5 时 *2 [清除条件] - 在通过 TGIV_5 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位为 0 时 - 在 CMFV5=1 的状态下，当读 CMFV5 后给 CMFV5 写 0 时
0	CMFW5	0	R/(W)*1	比较匹配 / 输入捕捉标志 W5 是表示 TGRW_5 的输入捕捉或者比较匹配的状态标志。 [置位条件] - 在 TGRW_5 用作比较匹配寄存器的情况下，当 TCNTW_5 = TGRW_5 时 - 在 TGRW_5 用作输入捕捉的情况下，当通过输入捕捉信号将 TCNTW_5 的值传送到 TGRW_5 时 - 在 TGRW_5 用于测量外部输入信号脉宽的情况下，将 TCNTW_5 的值传送到 TGRW_5 时 *2 [清除条件] - 在通过 TGIW_5 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位为 0 时 - 在 CMFW5=1 的状态下，当读 CMFW5 后给 CMFW5 写 0 时

【注】 *1 为了清除标志，只能在读 1 后写 0。

*2 通过定时器的 I/O 控制寄存器 U_5/V_5/W_5 (TIOU_5/V_5/W_5) 的 IOC 位设定传送时序。

10.3.7 定时器的缓冲运行传送模式寄存器 (TBTM)

TBTM 是 8 位可读写寄存器，设定在 PWM 模式中缓冲寄存器到定时器通用寄存器的传送时序。MTU2 的通道 0、3、4 各有 1 个，共计 3 个 TBTM。

位:	7	6	5	4	3	2	1	0
	—	—	—	—	—	TTSE	TTSB	TTSA
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7 ~ 3	—	全 0	R	保留位 读写值总是 0。
2	TTSE	0	R/W	时序的选择 E 设定在缓冲运行时 TGRF_0 到 TGRE_0 的传送时序。在通道 3、4 中，此位为保留位，读写值总是 0。另外，如果在 PWM 以外的模式中使用通道 0，就不能将此位置 1。 0: 当通道 0 发生比较匹配 E 时 1: 当清除 TCNT_0 时
1	TTSB	0	R/W	时序的选择 B 设定在各通道缓冲运行时 TGRD 到 TGRB 的传送时序。另外，如果在 PWM 以外的模式中使用通道 0，就不能将此位置 1。 0: 当各通道发生比较匹配 B 时 1: 当清除各通道的 TCNT 时
0	TTSA	0	R/W	时序的选择 A 设定在各通道缓冲运行时 TGRC 到 TGRA 的传送时序。另外，如果在 PWM 以外的模式中使用通道 0，就不能将此位置 1。 0: 当各通道发生比较匹配 A 时 1: 当清除各通道的 TCNT 时

10.3.8 定时器的输入捕捉控制寄存器 (TICCR)

TICCR 是 8 位可读写寄存器，控制 TCNT_1 和 TCNT_2 级联时的输入捕捉条件。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	初始值	R/W	说明
7 ~ 4	—	全 0	R	保留位 读写值总是 0。
3	I2BE	0	R/W	输入捕捉的允许 选择是否将 TIOC2B 引脚追加到 TGRB_1 的输入捕捉条件。 0: 不将 TIOC2B 引脚追加到 TGRB_1 的输入捕捉条件 1: 将 TIOC2B 引脚追加到 TGRB_1 的输入捕捉条件
2	I2AE	0	R/W	输入捕捉的允许 选择是否将 TIOC2A 引脚追加到 TGRA_1 的输入捕捉条件。 0: 不将 TIOC2A 引脚追加到 TGRA_1 的输入捕捉条件 1: 将 TIOC2A 引脚追加到 TGRA_1 的输入捕捉条件
1	I1BE	0	R/W	输入捕捉的允许 选择是否将 TIOC1B 引脚追加到 TGRB_2 的输入捕捉条件。 0: 不将 TIOC1B 引脚追加到 TGRB_2 的输入捕捉条件 1: 将 TIOC1B 引脚追加到 TGRB_2 的输入捕捉条件
0	I1AE	0	R/W	输入捕捉的允许 选择是否将 TIOC1A 引脚追加到 TGRA_2 的输入捕捉条件。 0: 不将 TIOC1A 引脚追加到 TGRA_2 的输入捕捉条件 1: 将 TIOC1A 引脚追加到 TGRA_2 的输入捕捉条件

10.3.9 定时器的同步清除寄存器 (TSYCR)

TSYCR 是 8 位可读写寄存器，设定从 MTU2 进行 MTU2S 的 TCNT_3 和 TCNT_4 的同步清除条件。MTU2S 的通道 3 有 1 个 TSYCR，但是 MTU2 没有 TSYCR。

位:	7	6	5	4	3	2	1	0
	CE0A	CE0B	CE0C	CE0D	CE1A	CE1B	CE2A	CE2B
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7	CE0A	0	R/W	清除的允许 0A 设定是禁止还是允许通过 MTU2 的 TSR_0 的 TGFA 标志置位进行清除。 0: 禁止通过 TSR_0 的 TGFA 标志置位进行清除 1: 允许通过 TSR_0 的 TGFA 标志置位进行清除

位	位名	初始值	R/W	说明
6	CE0B	0	R/W	清除的允许 0B 设定是禁止还是允许通过 MTU2 的 TSR_0 的 TGFB 标志置位进行清除。 0: 禁止通过 TSR_0 的 TGFB 标志置位进行清除 1: 允许通过 TSR_0 的 TGFB 标志置位进行清除
5	CE0C	0	R/W	清除的允许 0C 设定是禁止还是允许通过 MTU2 的 TSR_0 的 TGFC 标志置位进行清除。 0: 禁止通过 TSR_0 的 TGFC 标志置位进行清除 1: 允许通过 TSR_0 的 TGFC 标志置位进行清除
4	CE0D	0	R/W	清除的允许 0D 设定是禁止还是允许通过 MTU2 的 TSR_0 的 TGFD 标志置位进行清除。 0: 禁止通过 TSR_0 的 TGFD 标志置位进行清除 1: 允许通过 TSR_0 的 TGFD 标志置位进行清除
3	CE1A	0	R/W	清除的允许 1A 设定是禁止还是允许通过 MTU2 的 TSR_1 的 TGFA 标志置位进行清除。 0: 禁止通过 TSR_1 的 TGFA 标志置位进行清除 1: 允许通过 TSR_1 的 TGFA 标志置位进行清除
2	CE1B	0	R/W	清除的允许 1B 设定是禁止还是允许通过 MTU2 的 TSR_1 的 TGFB 标志置位进行清除。 0: 禁止通过 TSR_1 的 TGFB 标志置位进行清除 1: 允许通过 TSR_1 的 TGFB 标志置位进行清除
1	CE2A	0	R/W	清除的允许 2A 设定是禁止还是允许通过 MTU2 的 TSR_2 的 TGFA 标志置位进行清除。 0: 禁止通过 TSR_2 的 TGFA 标志置位进行清除 1: 允许通过 TSR_2 的 TGFA 标志置位进行清除
0	CE2B	0	R/W	清除的允许 2B 设定是禁止还是允许通过将 MTU2 的 TSR_2 的 TGFB 标志置位进行清除。 0: 禁止通过将 TSR_2 的 TGFB 标志置位进行清除 1: 允许通过将 TSR_2 的 TGFB 标志置位进行清除

10.3.10 定时器的 A/D 转换开始请求的控制寄存器 (TADCR)

TADCR 是 16 位可读写寄存器，允许或者禁止 A/D 转换开始请求以及设定 A/D 转换开始请求是否联动中断减少功能。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】* 只能在互补PWM模式时置1。

位	位名	初始值	R/W	说明
15、14	BF[1:0]	00	R/W	TADCOBRA/B_4 传送时序的选择 选择 TADCOBRA/B_4 到 TADCORA/B_4 的传送时序。 详细内容请参照表 10.29。
13 ~ 8	—	全 0	R	保留位 读写值总是 0。
7	UT4AE	0	R/W	递增计数 TRG4AN 的允许 在 TCNT_4 进行递增计数时，允许或者禁止 A/D 转换的开始请求 (TRG4AN)。 0: 在 TCNT_4 进行递增计数时，禁止 A/D 转换的开始请求 (TRG4AN) 1: 在 TCNT_4 进行递增计数时，允许 A/D 转换的开始请求 (TRG4AN)
6	DT4AE	0*	R/W	递减计数 TRG4AN 的允许 在 TCNT_4 进行递减计数时，允许或者禁止 A/D 转换的开始请求 (TRG4AN)。 0: 在 TCNT_4 进行递减计数时，禁止 A/D 转换的开始请求 (TRG4AN) 1: 在 TCNT_4 进行递减计数时，允许 A/D 转换的开始请求 (TRG4AN)
5	UT4BE	0	R/W	递增计数 TRG4BN 的允许 在 TCNT_4 进行递增计数时，允许或者禁止 A/D 转换的开始请求 (TRG4BN)。 0: 在 TCNT_4 进行递增计数时，禁止 A/D 转换的开始请求 (TRG4BN) 1: 在 TCNT_4 进行递增计数时，允许 A/D 转换的开始请求 (TRG4BN)
4	DT4BE	0*	R/W	递减计数 TRG4BN 的允许 在 TCNT_4 进行递减计数时，允许或者禁止 A/D 转换的开始请求 (TRG4BN)。 0: 在 TCNT_4 进行递减计数时，禁止 A/D 转换的开始请求 (TRG4BN) 1: 在 TCNT_4 进行递减计数时，允许 A/D 转换的开始请求 (TRG4BN)
3	ITA3AE	0*	R/W	TGIA_3 中断减少功能的联动允许 选择 A/D 转换的开始请求 (TRG4AN) 是否联动 TGIA_3 中断减少功能。 0: 不联动 TGIA_3 中断减少功能 1: 联动 TGIA_3 中断减少功能
2	ITA4VE	0*	R/W	TCIV_4 中断减少功能的联动允许 选择 A/D 转换的开始请求 (TRG4AN) 是否联动 TCIV_4 中断减少功能。 0: 不联动 TCIV_4 中断减少功能 1: 联动 TCIV_4 中断减少功能

位	位名	初始值	R/W	说明
1	ITB3AE	0*	R/W	TGIA_3 中断减少功能的联动允许 选择 A/D 转换的开始请求 (TRG4BN) 是否联动 TGIA_3 中断减少功能。 0: 不联动 TGIA_3 中断减少功能 1: 联动 TGIA_3 中断减少功能
0	ITB4VE	0*	R/W	TCIV_4 中断减少功能的联动允许 选择 A/D 转换的开始请求 (TRG4BN) 是否联动 TCIV_4 中断减少功能。 0: 不联动 TCIV_4 中断减少功能 1: 联动 TCIV_4 中断减少功能

- 【注】 1. 禁止以 8 位为单位存取 TADCR，必须以 16 位为单位进行存取。
2. 在禁止中断减少功能时（将定时器的中断减少设定寄存器（TITCR）的 T3AEN 位、T4VEN 位或者 TITCR 位的减少次数设定位（3ACOR 和 4VCOR）置 0 时），必须设定为不联动中断减少功能（将定时器的 A/D 转换开始请求控制寄存器（TADCR）的 ITA3AE 位、ITA4VE 位、ITB3AE 位和 ITB4VE 位置 0）。
3. 在禁止中断减少功能时，如果设定为联动中断减少功能，就不进行 A/D 转换的开始请求。
- * 除了互补 PWM 模式以外，不能置 1。

表 10.29 通过 BF1 位和 BF0 位设定的传送时序

bit7	bit6	说明
BF1	BF0	
0	0	不将周期设定缓冲寄存器的值传送到周期设定寄存器。
0	1	在 TCNT_4 的波峰，将周期设定缓冲寄存器的值传送到周期设定寄存器 *1。
1	0	在 TCNT_4 的波谷，将周期设定缓冲寄存器的值传送到周期设定寄存器 *2。
1	1	在 TCNT_4 的波峰和波谷，将周期设定缓冲寄存器的值传送到周期设定寄存器 *2。

- 【注】 *1 在互补 PWM 模式中 TCNT_4 处于波谷时，或者在复位同步 PWM 模式中 TCNT_3 和 TGRA_3 比较匹配时，或者在 PWM 模式 1/ 正常运行模式中 TCNT_4 和 TGRA_4 比较匹配时，将周期设定缓冲寄存器的值传送到周期设定寄存器。
- *2 除了互补 PWM 模式以外，禁止设定。

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

TADCORA/B_4 是 16 位可读写寄存器。当 TADCORA/B_4 和 TCNT_4 相同时, 就发生对应的 A/D 转换开始请求。

TADCORA/B_4 的初始值为 H'FFFF。

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

【注】 禁止以8位为单位存取TADCORA/B_4, 必须以16位为单位进行存取。

10.3.12 定时器的 A/D 转换开始请求的周期设定缓冲寄存器 (TADCOBRA/B_4)

TADCOBRA/B_4 是 16 位可读写寄存器。在波峰或者波谷将 TADCORA/B_4 的缓冲寄存器的值传送到 TADCORA/B_4。

TADCOBRA/B_4 的初始值为 H'FFFF。

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, 必须以16位为单位进行存取。

10.3.13 定时器的计数器 (TCNT)

TCNT 是 16 位可读写计数器。通道 0 ~ 4 各有 1 个, 通道 5 有 3 个 TCNTU/V/W_5, 共计 8 个 TCNT。TCNT 在复位时被初始化为 H'0000。

禁止以 8 位为单位存取 TCNT, 必须以 16 位为单位进行存取。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
初始值:	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, 必须以16位为单位进行存取。

10.3.14 定时器的通用寄存器 (TGR)

TGR 是 16 位可读写寄存器。通道 0 有 6 个, 通道 1、2 各有 2 个, 通道 3、4 各有 4 个, 通道 5 有 3 个, 共计 21 个通用寄存器。

TGRA、TGRB、TGRC 和 TGRD 是输出比较 / 输入捕捉兼用的寄存器。能将通道 0、3、4 的 TGRC 和 TGRD 用作缓冲寄存器。TGR 和缓冲寄存器的组合为 TGRA-TGRC 和 TGRB-TGRD

TGRE_0 和 TGRF_0 用作比较寄存器, 当 TCNT_0 和 TGRE_0 相同时, 就发生 A/D 转换的开始请求。能将 TGRF 用作缓冲寄存器。TGR 和缓冲寄存器的组合为 TGRE-TGRF。

TGRU_5、TGRV_5 和 TGRW_5 是比较匹配 / 输入捕捉 / 测量外部脉宽兼用的寄存器。

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，必须以16位为单位进行存取。TGR的初始值为H'FFFF。

10.3.15 定时器的开始寄存器 (TSTR)

TSTR 是 8 位可读写寄存器，选择通道 0 ~ 4 的 TCNT 运行或者停止。
TSTR_5 是 8 位可读写寄存器，选择通道 5 的 TCNTU/V/W_5 运行或者停止。
在给 TMDR 设定运行模式时或者给 TCR 设定 TCNT 的计数时钟时，必须停止 TCNT 的计数器。

• 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	初始值	R/W	说明
7	CST4	0	R/W	计数器的开始 4、3 选择 TCNT 的运行或者停止。 在 TIOC 引脚为输出的状态下，如果在运行中给 CST 位写 0，就停止计数器并保持 TIOC 引脚的输出比较的输出电平。如果在 CST 位为 0 的状态下写 TIOR，引脚的输出电平就被更新为已设定的初始输出值。 0: TCNT_4 和 TCNT_3 停止计数 1: TCNT_4 和 TCNT_3 计数
6	CST3	0	R/W	
5 ~ 3	—	全 0	R	保留位 读写值总是 0。
2	CST2	0	R/W	计数器的开始 2 ~ 0 选择 TCNT 的运行或者停止。 在 TIOC 引脚为输出的状态下，如果在运行中给 CST 位写 0，就停止计数器并保持 TIOC 引脚的输出比较的输出电平。如果在 CST 位为 0 的状态下写 TIOR，引脚的输出电平就被更新为已设定的初始输出值。 0: TCNT_2 ~ TCNT_0 停止计数 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	初始值	R/W	说明
7 ~ 3	—	全 0	R	保留位 读写值总是 0。
2	CSTU5	0	R/W	计数器的开始 U5 选择 TCNTU_5 的运行或者停止。 0: TCNTU_5 停止计数 1: TCNTU_5 计数
1	CSTV5	0	R/W	计数器的开始 V5 选择 TCNTV_5 的运行或者停止。 0: TCNTV_5 停止计数 1: TCNTV_5 计数
0	CSTW5	0	R/W	计数器的开始 W5 选择 TCNTW_5 的运行或者停止。 0: TCNTW_5 停止计数 1: TCNTW_5 计数

10.3.16 定时器的同步寄存器 (TSYR)

TSYR 是 8 位可读写寄存器，选择通道 0 ~ 4 的 TCNT 独立运行或者同步运行。对应位置 1 的通道进行同步运行。

位:	7	6	5	4	3	2	1	0
	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7	SYNC4	0	R/W	定时器的同步 4、3
6	SYNC3	0	R/W	选择和其他通道同步运行或者独立运行。 在选择同步运行时，能对多个 TCNT 进行同步预置，并且能通过其他通道的计数器清除进行同步清除。 要设定同步运行时，至少需要将 2 个通道的 SYNC 位置 1。要设定同步清除时，除了 SYNC 位以外，还需要通过 TCR 的 CCLR2 ~ CCLR0 位设定 TCNT 的清除源。 0: TCNT_4 和 TCNT_3 独立运行 (TCNT 的预置 / 清除与其他通道无关) 1: TCNT_4 和 TCNT_3 同步运行 (能进行 TCNT 的同步预置 / 同步清除)
5 ~ 3	—	全 0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
2	SYNC2	0	R/W	定时器的同步 2 ~ 0 选择和其他通道同步运行或者独立运行。 在选择同步运行时，能对多个 TCNT 进行同步预置，并能通过其他通道的计数器清除进行同步清除。 要设定同步运行时，至少需要将 2 个通道的 SYNC 位置 1。要设定同步清除时，除了 SYNC 位以外，还需要通过 TCR 的 CCLR2 ~ CCLR0 位设定 TCNT 的清除源。 0: TCNT_2 ~ TCNT_0 独立运行 (TCNT 的预置 / 清除与其他通道无关) 1: TCNT_2 ~ TCNT_0 同步运行 (能进行 TCNT 的同步预置 / 同步清除)
1	SYNC1	0	R/W	
0	SYNC0	0	R/W	

10.3.17 定时器的计数器同步开始寄存器 (TCSYSTR)

TCSYSTR 是 8 位可读写寄存器，控制 MTU2 和 MTU2S 计数器的同步开始，但是 MTU2S 没有 TCSYSTR。

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

【注】* 为了给寄存器置位，只能写1。

位	位名	初始值	R/W	说明
7	SCH0	0	R/(W)*	同步开始 控制 MTU2 的 TCNT_0 的同步开始。 0: MTU2 的 TCNT_0 不同步开始 1: MTU2 的 TCNT_0 同步开始 [清除条件] • 在 SCH0=1 的状态下，当将 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: MTU2 的 TCNT_1 不同步开始 1: MTU2 的 TCNT_1 同步开始 [清除条件] • 在 SCH1=1 的状态下，当将 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: MTU2 的 TCNT_2 不同步开始 1: MTU2 的 TCNT_2 同步开始 [清除条件] • 在 SCH2=1 的状态下，当将 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: MTU2 的 TCNT_3 不同步开始 1: MTU2 的 TCNT_3 同步开始 [清除条件] • 在 SCH3=1 的状态下，当将 MTU2 的 TSTR 的 CST3 位置 1 时

位	位名	初始值	R/W	说明
3	SCH4	0	R/(W)*	同步开始 控制 MTU2 的 TCNT_4 的同步开始。 0: MTU2 的 TCNT_4 不同步开始 1: MTU2 的 TCNT_4 同步开始 [清除条件] • 在 SCH4=1 的状态下, 当将 MTU2 的 TSTR 的 CST4 位置 1 时
2	—	0	R	保留位 读写值总是 0。
1	SCH3S	0	R/(W)*	同步开始 控制 MTU2S 的 TCNT_3S 的同步开始。 0: MTU2S 的 TCNT_3S 不同步开始 1: MTU2S 的 TCNT_3S 同步开始 [清除条件] • 在 SCH3S=1 的状态下, 当将 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: MTU2S 的 TCNT_4S 不同步开始 1: MTU2S 的 TCNT_4S 同步开始 [清除条件] • 在 SCH4S=1 的状态下, 当将 MTU2S 的 TSTRS 的 CST4 位置 1 时

【注】 * 为了将寄存器置位, 只能写 1。

10.3.18 定时器的读写允许寄存器 (TRWER)

TRWER 是 8 位可读写寄存器, 允许或者禁止存取通道 3、4 的误写防止对象寄存器 / 计数器。

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

位	位名	初始值	R/W	说明
7 ~ 1	—	全 0	R	保留位 读写值总是 0。
0	RWE	1	R/W	允许读写 允许或者禁止读写误写防止对象寄存器。 0: 禁止读写寄存器 1: 允许读写寄存器 [清除条件] • 在 RWE=1 的状态下, 当读 RWE 后给 RWE 写 0 时

- 误写防止的对象寄存器 / 计数器

TCR_3、4、TMDR_3、4、TIOH_3、4、TIORL_3、4、TIER_3、4、TGRA_3、4、TGRB_3、4、TOER、TOCR1、TOCR2、TGCR、TCDR、TDDR 与 TCNT_3、4, 共计 22 个寄存器。

10.3.19 定时器的输出主控允许寄存器（TOER）

TOER 是 8 位可读写寄存器，允许或者禁止输出引脚 TIOC4D、TIOC4C、TIOC3D、TIOC4B、TIOC4A、TIOC3B 的输出设定。如果不设定 TOER 的各位，这些引脚就不能正确输出。对于通道 3、4，必须在进行通道 3、4 的 TIOR 设定前设定 TOER 的值。

位:	7	6	5	4	3	2	1	0
	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
初始值:	1	1	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7、6	—	全 1	R	保留位 读写值总是 1。
5	OE4D	0	R/W	主控的允许 TIOC4D 允许或者禁止 TIOC4D 引脚的 MTU2 输出。 0: 禁止 MTU2 输出（无效电平）* 1: 允许 MTU2 输出
4	OE4C	0	R/W	主控的允许 TIOC4C 允许或者禁止 TIOC4C 引脚的 MTU2 输出。 0: 禁止 MTU2 输出（无效电平）* 1: 允许 MTU2 输出
3	OE3D	0	R/W	主控的允许 TIOC3D 允许或者禁止 TIOC3D 引脚的 MTU2 输出。 0: 禁止 MTU2 输出（无效电平）* 1: 允许 MTU2 输出
2	OE4B	0	R/W	主控的允许 TIOC4B 允许或者禁止 TIOC4B 引脚的 MTU2 输出。 0: 禁止 MTU2 输出（无效电平）* 1: 允许 MTU2 输出
1	OE4A	0	R/W	主控的允许 TIOC4A 允许或者禁止 TIOC4A 引脚的 MTU2 输出。 0: 禁止 MTU2 输出（无效电平）* 1: 允许 MTU2 输出
0	OE3B	0	R/W	主控的允许 TIOC3B 允许或者禁止 TIOC3B 引脚的 MTU2 输出。 0: 禁止 MTU2 输出（无效电平）* 1: 允许 MTU2 输出

【注】 * 无效电平取决于定时器的输出控制寄存器 1/2（TOCR1/2）的设定。详细内容请参照“10.3.20 定时器的输出控制寄存器 1（TOCR1）”和“10.3.21 定时器的输出控制寄存器 2（TOCR2）”。另外，除了互补 PWM 模式 / 复位同步 PWM 模式以外，在进行 MTU2 输出时必须置 1，如果置 0 就输出低电平。

10.3.20 定时器的输出控制寄存器 1 (TOCR1)

TOCR1 是 8 位可读写寄存器，允许或者禁止与互补 PWM 模式 / 复位同步 PWM 模式的 PWM 周期同步的交替输出，进行 PWM 输出电平的反相控制。

位:	7	6	5	4	3	2	1	0
	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R/(W)*	R/W	R/W	R/W

【注】* 在上电复位后，只能进行1次的写1。写1后不能写0。

位	位名	初始值	R/W	说明
7	—	0	R	保留位 读写值总是 0。
6	PSYE	0	R/W	PWM 同步输出的允许 允许或者禁止与 PWM 周期同步的交替输出。 0: 禁止交替输出 1: 允许交替输出
5、4	—	全 0	R	保留位 读写值总是 0。
3	TOCL	0	R/(W)*	TOC 寄存器的写禁止位 *1 禁止或者允许写 TOCR1 寄存器的 TOCS 位、OLSN 位和 OLSP 位。 0: 允许写 TOCS 位、OLSN 位和 OLSP 位 1: 禁止写 TOCS 位、OLSN 位和 OLSP 位
2	TOCS	0	R/W	TOC 的选择位 对于互补 PWM 模式 / 复位同步 PWM 模式的输出电平的设定，是选择 TOCR1 的设定有效还是选择 TOCR2 的设定有效。 0: TOCR1 的设定有效 1: TOCR2 的设定有效
1	OLSN	0	R/W	输出电平的选择 N*2 选择复位同步 PWM 模式 / 互补 PWM 模式的反相输出电平。详细内容请参照表 10.30。
0	OLSP	0	R/W	输出电平的选择 P*2 选择复位同步 PWM 模式 / 互补 PWM 模式的正相输出电平。详细内容请参照表 10.31。

【注】*1 能通过将 TOCL 位置 1，防止 CPU 失控时的误写。

*2 通过将 TOCS 位置 0，使此设定有效。

表 10.30 输出电平的选择功能

bit1	功能			
OLSN	初始输出	有效电平	比较匹配输出	
			递增计数	递减计数
0	高电平	低电平	高电平	低电平
1	低电平	高电平	低电平	高电平

【注】反相波形的初始输出值在开始计数并经过空载时间后变为有效电平。

表 10.31 输出电平的选择功能

bit0	功能			
OLSP	初始输出	有效电平	比较匹配输出	
			递增计数	递减计数
0	高电平	低电平	低电平	高电平
1	低电平	高电平	高电平	低电平

OLSN=1、OLSP=1 时的互补 PWM 模式的输出例子（1 相）如图 10.2 所示。

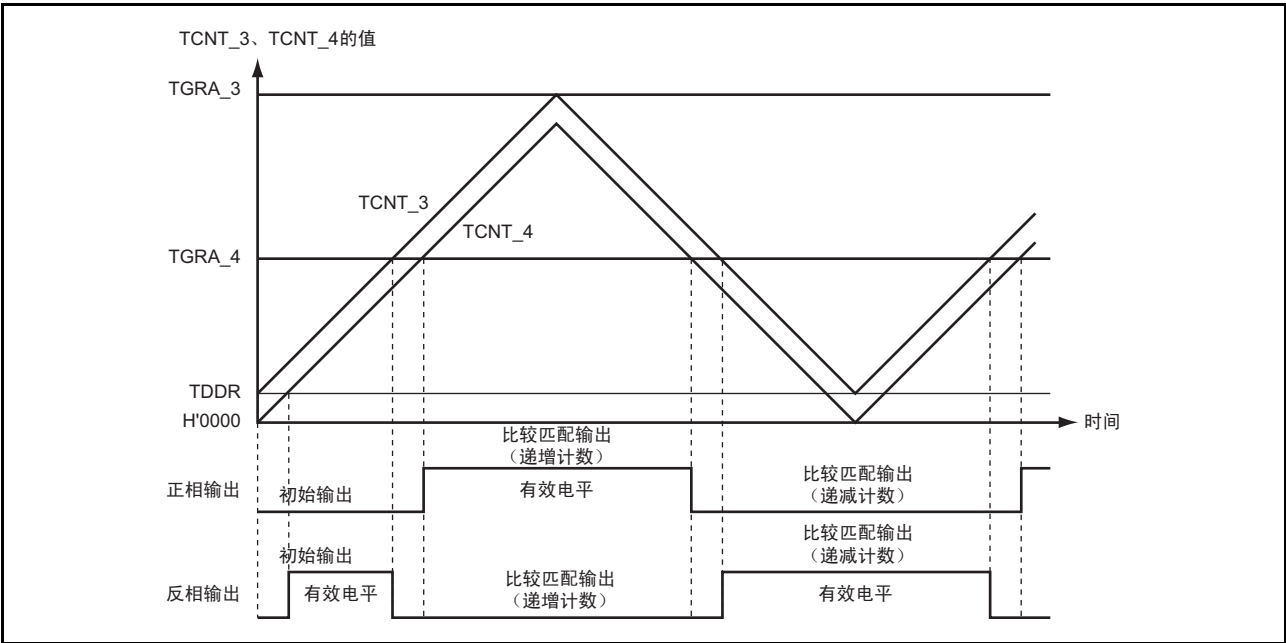


图 10.2 互补 PWM 模式的输出电平的例子

10.3.21 定时器的输出控制寄存器 2（TOCR2）

TOCR2 是 8 位可读写寄存器，在互补 PWM 模式 / 复位同步 PWM 模式中进行 PWM 输出电平的反相控制。

位:	7	6	5	4	3	2	1	0
	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7、6	BF[1:0]	00	R/W	TOLBR 缓冲传送时序的选择 选择 TOLBR 到 TOCR2 的缓冲传送时序。 详细内容请参照表 10.32。
5	OLS3N	0	R/W	输出电平的选择 3N* 选择复位同步 PWM 模式 / 互补 PWM 模式的 TIOC4D 的输出电平。详细内容请参照表 10.33。

位	位名	初始值	R/W	说明
4	OLS3P	0	R/W	输出电平的选择 3P* 选择复位同步 PWM 模式 / 互补 PWM 模式的 TIOC4B 的输出电平。详细内容请参照表 10.34。
3	OLS2N	0	R/W	输出电平选择 2N* 选择复位同步 PWM 模式 / 互补 PWM 模式的 TIOC4C 的输出电平。详细内容请参照表 10.35。
2	OLS2P	0	R/W	输出电平的选择 2P* 选择复位同步 PWM 模式 / 互补 PWM 模式的 TIOC4A 的输出电平，详细内容请参照表 10.36。
1	OLS1N	0	R/W	输出电平的选择 1N* 选择复位同步 PWM 模式 / 互补 PWM 模式的 TIOC3D 的输出电平。详细内容请参照表 10.37。
0	OLS1P	0	R/W	输出电平的选择 1P* 选择复位同步 PWM 模式 / 互补 PWM 模式的 TIOC3B 的输出电平。详细内容请参照表 10.38。

【注】 * 通过将 TOCR1 的 TOCS 位置 1，使此设定有效。

表 10.32 BF1 位和 BF0 位的设定

bit7	bit6	说明	
BF1	BF0	互补 PWM 模式	复位 PWM 模式
0	0	不将缓冲寄存器 (TOLBR) 的值传送到 TOCR2。	不将缓冲寄存器 (TOLBR) 的值传送到 TOCR2。
0	1	在 TCNT_4 的波峰，将缓冲寄存器 (TOLBR) 的值传送到 TOCR2。	在清除 TCNT_3/4 计数器时，将缓冲寄存器 (TOLBR) 的值传送到 TOCR2。
1	0	在 TCNT_4 的波谷，将缓冲寄存器 (TOLBR) 的值传送到 TOCR2。	禁止设定。
1	1	在 TCNT_4 的波峰和波谷，将缓冲寄存器 (TOLBR) 的值传送到 TOCR2。	禁止设定。

表 10.33 TIOC4D 输出电平的选择功能

bit5	功能			
OLS3N	初始输出	有效电平	比较匹配输出	
			递增计数	递减计数
0	高电平	低电平	高电平	低电平
1	低电平	高电平	低电平	高电平

【注】 反相波形的初始输出值在开始计数并经过空载时间后变为有效电平。

表 10.34 TIOC4B 输出电平的选择功能

bit4	功能			
OLS3P	初始输出	有效电平	比较匹配输出	
			递增计数	递减计数
0	高电平	低电平	低电平	高电平
1	低电平	高电平	高电平	低电平

表 10.35 TIOC4C 输出电平的选择功能

bit3	功能			
OLS2N	初始输出	有效电平	比较匹配输出	
			递增计数	递减计数
0	高电平	低电平	高电平	低电平
1	低电平	高电平	低电平	高电平

【注】 反相波形的初始输出值在开始计数并经过空载时间后变为有效电平。

表 10.36 TIOC4A 输出电平的选择功能

bit2	功能			
OLS2P	初始输出	有效电平	比较匹配输出	
			递增计数	递减计数
0	高电平	低电平	低电平	高电平
1	低电平	高电平	高电平	低电平

表 10.37 TIOC3D 输出电平的选择功能

bit1	功能			
OLS1N	初始输出	有效电平	比较匹配输出	
			递增计数	递减计数
0	高电平	低电平	高电平	低电平
1	低电平	高电平	低电平	高电平

【注】 反相波形的初始输出值在开始计数并经过空载时间后变为有效电平。

表 10.38 TIOC4B 输出电平的选择功能

bit0	功能			
OLS1P	初始输出	有效电平	比较匹配输出	
			递增计数	递减计数
0	高电平	低电平	低电平	高电平
1	低电平	高电平	高电平	低电平

10.3.22 定时器的输出电平缓冲寄存器 (TOLBR)

TOLBR 是 TOCR2 的 8 位可读写缓冲寄存器，设定互补 PWM 模式 / 复位同步 PWM 模式的 PWM 输出电平。

位:	7	6	5	4	3	2	1	0
	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7、6	—	全 0	R	保留位 读写值总是 0。
5	OLS3N	0	R/W	必须给 TOCR2 的 OLS3N 位设定缓冲传送值。
4	OLS3P	0	R/W	必须给 TOCR2 的 OLS3P 位设定缓冲传送值。
3	OLS2N	0	R/W	必须给 TOCR2 的 OLS2N 位设定缓冲传送值。
2	OLS2P	0	R/W	必须给 TOCR2 的 OLS2P 位设定缓冲传送值。
1	OLS1N	0	R/W	必须给 TOCR2 的 OLS1N 位设定缓冲传送值。
0	OLS1P	0	R/W	必须给 TOCR2 的 OLS1P 位设定缓冲传送值。

缓冲运行中 PWM 输出电平的设定步骤例子如图 10.3 所示。

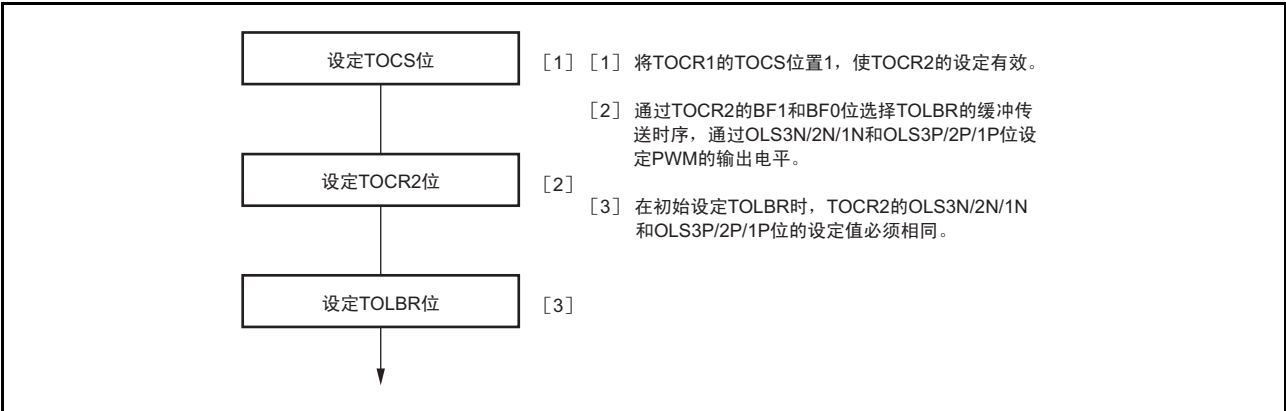


图 10.3 缓冲运行中 PWM 输出电平的设定步骤例子

10.3.23 定时器的门控寄存器 (TGCR)

TGCR 是 8 位可读写寄存器，在复位同步 PWM 模式 / 互补 PWM 模式中进行无刷 DC 马达控制所需波形的输出控制。除了互补 PWM 模式 / 复位同步 PWM 模式以外，此寄存器的设定无效。

位:	7	6	5	4	3	2	1	0
	—	BDC	N	P	FB*	WF	VF	UF
初始值:	1	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7	—	1	R	保留位 读写值总是 1。
6	BDC	0	R/W	无刷 DC 马达 选择是将此寄存器的功能设定为有效还是无效。 0: 正常输出 1: 此寄存器的功能有效
5	N	0	R/W	反相输出 (N) 的控制 在输出反相引脚 (TIOC3D 引脚、TIOC4C 引脚和 TIOC4D 引脚) 时, 选择电平输出或者复位同步 PWM/ 互补 PWM 输出。 0: 电平输出 1: 复位同步 PWM/ 互补 PWM 输出
4	P	0	R/W	正相输出 (P) 的控制 在输出正相引脚 (TIOC3B 引脚、TIOC4A 引脚和 TIOC4B 引脚) 时, 选择电平输出或者复位同步 PWM/ 互补 PWM 输出。 0: 电平输出 1: 复位同步 PWM/ 互补 PWM 输出
3	FB*	0	R/W	外部反馈信号的允许 选择是通过 MTU2/ 通道 0 的 TGRA、TGRB、TGRC 的输入捕捉信号还是通过给 TGCR 的 bit2 ~ 0 写 0 或 1 自动进行正相 / 反相的输出转换。 0: 通过外部输入进行输出的转换 (输入源为通道 0 的 TGRA、TGRB 和 TGRC 的输入捕捉信号) 1: 通过软件进行输出的转换 (TGCR 的 UF、VF、WF 的设定值)
2	WF	0	R/W	输出相的转换 2 ~ 0 设定正相 / 反相的输出相 ON/OFF。这些位的设定只在此寄存器的 FB 位为 1 时有效。此时, bit2 ~ 0 的设定取代外部输入。详细内容请参照表 10.39。
1	VF	0	R/W	
0	UF	0	R/W	

【注】 * 在 MTU2S 的 BDC 位被置 1 时, 不能将 FB 位置 0。

表 10.39 输出电平的选择功能

bit2	bit1	bit0	功能					
WF	VF	UF	TIOC3B	TIOC4A	TIOC4B	TIOC3D	TIOC4C	TIOC4D
			U 相	V 相	W 相	U 相	V 相	W 相
0	0	0	OFF	OFF	OFF	OFF	OFF	OFF
0	0	1	ON	OFF	OFF	OFF	OFF	ON
0	1	0	OFF	ON	OFF	ON	OFF	OFF
0	1	1	OFF	ON	OFF	OFF	OFF	ON
1	0	0	OFF	OFF	ON	OFF	ON	OFF
1	0	1	ON	OFF	OFF	OFF	ON	OFF
1	1	0	OFF	OFF	ON	ON	OFF	OFF
1	1	1	OFF	OFF	OFF	OFF	OFF	OFF

10.3.24 定时器的子计数器 (TCNTS)

TCNTS 是只在互补 PWM 模式中使用的 16 位只读计数器，初始值为 H'0000。

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，必须以16位为单位进行存取。

10.3.25 定时器的空载时间数据寄存器 (TDDR)

TDDR 是只在互补 PWM 模式中使用的 16 位寄存器，设定互补 PWM 模式的 TCNT_3 和 TCNT_4 计数器的偏移值。在互补 PWM 模式中清除 TCNT_3 和 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，必须以16位为单位进行存取。

10.3.26 定时器的周期数据寄存器 (TCDR)

TCDR 是只在互补 PWM 模式中使用的 16 位寄存器。TCDR 寄存器的值必须为 PWM 载波周期的 1/2。此寄存器在互补 PWM 模式中随时和 TCNTS 计数器比较，如果相同，TCNTS 计数器就转换计数方向（递减计数 → 递增计数）。TCDR 的初始值为 H'FFFF。

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

【注】 禁止以8位为单位存取TCDR，必须以16位为单位进行存取。

10.3.27 定时器的周期缓冲寄存器 (TCBR)

TCBR 是只在互补 PWM 模式中使用的 16 位寄存器，用作 TCDR 寄存器的缓冲寄存器。在 TMDR 寄存器设定的传送时序中将 TCBR 寄存器的值传送到 TCDR 寄存器。TCBR 的初始值为 H'FFFF。

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，必须以16位为单位进行存取。

10.3.28 定时器的中断减少设定寄存器（TITCR）

TITCR 是 8 位可读写寄存器，禁止或者允许中断减少功能以及设定中断减少次数。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	初始值	R/W	说明
7	T3AEN	0	R/W	T3AEN 禁止或者允许 TGIA_3 的中断减少功能。 0: 禁止 TGIA_3 的中断减少功能 1: 允许 TGIA_3 的中断减少功能
6 ~ 4	3ACOR[2:0]	000	R/W	以 0 ~ 7 次设定 TGIA_3 的中断减少次数。* 详细内容请参照表 10.40。
3	T4VEN	0	R/W	T4VEN 禁止或者允许 TCIV_4 的中断减少功能。 0: 禁止 TCIV_4 的中断减少功能 1: 允许 TCIV_4 的中断减少功能
2 ~ 0	4VCOR[2:0]	000	R/W	以 0 ~ 7 次设定 TCIV_4 的中断减少次数。* 详细内容请参照表 10.41。

【注】 * 如果将中断减少次数置 0，就不减少中断。
另外，在设定中断减少次数前，必须在将 T3AEN 位和 T4VEN 位置 0 后清除中断减少次数计数器（TITCNT）。

表 10.40 通过 3ACOR2 ~ 3ACOR0 位设定中断减少次数

bit6	bit5	bit4	说明
3ACOR2	3ACOR1	3ACOR0	
0	0	0	不减少 TGIA_3 的中断。
0	0	1	将 TGIA_3 中断减少次数设定为 1 次。
0	1	0	将 TGIA_3 中断减少次数设定为 2 次。
0	1	1	将 TGIA_3 中断减少次数设定为 3 次。
1	0	0	将 TGIA_3 中断减少次数设定为 4 次。
1	0	1	将 TGIA_3 中断减少次数设定为 5 次。
1	1	0	将 TGIA_3 中断减少次数设定为 6 次。
1	1	1	将 TGIA_3 中断减少次数设定为 7 次。

表 10.41 通过 4VCOR2 ~ 4VCOR0 位设定中断减少次数

bit2	bit1	bit0	说明
4VCOR2	4VCOR1	4VCOR0	
0	0	0	不减少 TCIV_4 的中断。
0	0	1	将 TCIV_4 中断减少次数设定为 1 次。
0	1	0	将 TCIV_4 中断减少次数设定为 2 次。
0	1	1	将 TCIV_4 中断减少次数设定为 3 次。
1	0	0	将 TCIV_4 中断减少次数设定为 4 次。
1	0	1	将 TCIV_4 中断减少次数设定为 5 次。
1	1	0	将 TCIV_4 中断减少次数设定为 6 次。
1	1	1	将 TCIV_4 中断减少次数设定为 7 次。

10.3.29 定时器的中断减少次数计数器 (TITCNT)

TITCNT 是 8 位可读计数器。MTU2 有 1 个 TITCNT，在 TCNT_3 和 TCNT_4 停止计数后，保持 TITCNT 的值。

位:	7	6	5	4	3	2	1	0
	—	3ACNT[2:0]			—	4VCNT[2:0]		
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
7	—	0	R	保留位 读取值总是 0。
6 ~ 4	3ACNT [2:0]	000	R	TGIA_3 中断计数器 如果将 TITCR 的 T3AEN 位置 1，就在产生 TGIA_3 中断源时进行递增计数（加 1）。 [清除条件] - 当 TITCR 的 3ACOR2 ~ 3ACOR0 和 TITCNT 的 3ACNT2 ~ 3ACNT0 相同时 - 当 TITCR 的 T3AEN 位为 0 时 - 当 TITCR 的 3ACOR2 ~ 3ACOR0 为 0 时
3	—	0	R	保留位 读取值总是 0。
2 ~ 0	4VCNT [2:0]	000	R	TCIV_4 中断计数器 如果将 TITCR 的 T4VEN 位置 1，就在产生 TCIV_4 中断源时进行递增计数（加 1）。 [清除条件] - 当 TITCR 的 4VCOR2 ~ 4VCOR0 和 TITCNT 的 4VCNT2 ~ 4VCNT0 相同时 - 当 TITCR 的 T4VEN 位为 0 时 - 当 TITCR 的 4VCOR2 ~ 4VCOR0 为 0 时

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

10.3.30 定时器的缓冲传送设定寄存器（TBTER）

TBTER 是 8 位可读写寄存器，设定是否抑制缓冲寄存器 *（在互补 PWM 模式中使用的缓冲寄存器）到暂存器的传送，或者设定是否联动中断减少功能。MTU2 有 1 个 TBTER。

位:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	BTE[1:0]	
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

位	位名	初始值	R/W	说明
7 ~ 2	—	全 0	R	保留位 读写值总是 0。
1、0	BTE[1:0]	00	R/W	设定是否抑制缓冲寄存器 *（在互补 PWM 模式中使用的缓冲寄存器）到暂存器的传送，或者设定是否联动中断减少功能。详细内容请参照表 10.42。

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

表 10.42 BTE1 位和、BTE0 位的设定

bit1	bit0	说明
BTE1	BTE0	
0	0	不抑制缓冲寄存器到暂存器的传送 *1，并且不联动中断减少功能。
0	1	抑制缓冲寄存器到暂存器的传送。
1	0	将缓冲寄存器到暂存器的传送与中断减少功能联动 *2。
1	1	禁止设定。

【注】 *1 按照 TMDR 的 MD3 ~ MD0 的设定进行传送。详细内容请参照“10.4.8 互补 PWM 模式”。
*2 在禁止中断减少功能时（在将定时器的中断减少设定寄存器（TITCR）的 T3AEN 位、T4VEN 位或者 TITCR 位的减少间隔次数设定位（3ACOR 和 4VCOR）置 0 时），必须设定为缓冲传送不联动中断减少功能（将定时器的缓冲传送寄存器（TBTER）的 BTE1 置 0）。
如果设定为缓冲传送联动中断减少功能，就在禁止中断减少功能时不进行缓冲传送。

10.3.31 定时器的空载时间允许寄存器 (TDER)

TDER 是 8 位可读写寄存器。通道 3 有 1 个 TDER，能控制互补 PWM 模式的空载时间的生成。MTU2 有 1 个 TDER。必须在 TCNT 停止运行的状态下进行 TDER 的设置。

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

位	位名	初始值	R/W	说明
7 ~ 1	—	全 0	R	保留位 读写值总是 0。
0	TDER	1	R(W)	空载时间允许寄存器 设定是否生成空载时间。 0: 不生成空载时间 1: 生成空载时间 * [清除条件] • 在 TDER=1 的状态下, 当读 TDER 后给 TDER 写 0 时

【注】 * 必须设定为 $TDDR \geq 1$ 。

10.3.32 定时器的波形控制寄存器 (TWCR)

TWCR 是 8 位可读写寄存器, 设定在互补 PWM 模式中发生 TNCT_3、TNCT_4 同步计数器清除时的输出波形控制以及设定是否进行通过 TGRA_3 比较匹配产生的计数器清除。必须在 TCNT 运行停止的状态下进行 TWCR 的 CCE 位和 WRE 位的设定。

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

【注】 * 只能在互补 PWM 模式 1 时置 1。

位	位名	初始值	R/W	说明
7	CCE	0*	R(W)	比较匹配清除的允许 设定是否在互补 PWM 模式中进行通过 TGRA_3 比较匹配产生的计数器清除。 0: 不进行通过 TGRA_3 比较匹配产生的计数器清除 1: 进行通过 TGRA_3 比较匹配产生的计数器清除 [置位条件] • 在 CCE=0 的状态下, 当读 CCE 后给 CCE 写 1 时
6 ~ 2	—	全 0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
1	SCC	0	R/(W)	同步清除控制 设定在互补 PWM 模式中发生 MTU2-MTU2S 计数器同步清除时是否清除 MTU2S 的 TCNT_3 和 TCNT_4。 在使用此功能时，必须将 MTU2S 设定为互补 PWM 模式。另外，在计数器运行中改写 SCC 位时，不能更改 CCE 位和 WRE 位的值。 只有在波谷的 Tb 区间以外的区间发生同步清除时，才能通过设定 SCC 位使 MTU2 的同步清除无效。如果在包含 TCNT_3 和 TCNT_4 刚开始计数后的波谷的 Tb 区间发生同步清除，就清除 MTU2S 的 TCNT_3 和 TCNT_4。 有关互补 PWM 模式的波谷 Tb 区间，请参照图 10.40。 对于 MTU2，此位为保留位，读写值总是 0。 0: 通过 MTU2-MTU2S 同步清除功能进行的 MTU2S 的 TCNT_3 和 TCNT_4 清除有效 1: 通过 MTU2-MTU2S 同步清除功能进行的 MTU2S 的 TCNT_3 和 TCNT_4 清除无效 [置位条件] 在 SCC=0 的状态下，当读 SCC 后给 SCC 写 1 时
0	WRE	0	R/(W)	波形保持的允许 选择在互补 PWM 模式中发生同步计数器清除时的输出波形。此功能只在互补 PWM 模式的波谷 Tb 区间发生同步清除时保持波形。如果在其他区间发生同步清除，就与 WRE 位的设定无关，输出 TOCR 寄存器设定的初始值。另外，如果在 TCNT_3 和 TCNT_4 刚开始的波谷 Tb 区间发生同步清除，也输出 TOCR 寄存器设定的初始值。 有关互补 PWM 模式的波谷 Tb 区间，请参照图 10.40。 0: 输出 TOCR 寄存器设定的初始输出值 1: 保持同步清除前的波形 [置位条件] 在 WRE=0 的状态下，当读 WRE 后给 WRE 写 1 时

【注】 * 只能在互补 PWM 模式 1 时置 1。

10.3.33 和总线主控的接口

定时器的计数器 (TCNT)、通用寄存器 (TGR)、定时器的子计数器 (TCNTS)、定时器的周期缓冲寄存器 (TCBR)、定时器的空载时间数据寄存器 (TDDR)、定时器的周期数据寄存器 (TCDR)、定时器的 A/D 转换开始请求的控制寄存器 (TADCR)、定时器的 A/D 转换开始请求的周期设定寄存器 (TADCOR) 以及定时器的 A/D 转换开始请求的周期设定缓冲寄存器 (TADCOBR) 是 16 位寄存器。因为和总线主控之间的数据总线为 16 位宽，所以能以 16 位为单位而不能以 8 位为单位进行读写。必须以 16 位为单位进行存取。

上述以外的寄存器是 8 位寄存器，因为和 CPU 之间的数据总线为 16 位宽，所以既能以 16 位为单位也能以 8 位为单位进行读写。

10.4 运行说明

10.4.1 基本运行

各通道有 TCNT 和 TGR。TCNT 能进行递增计数、自由计数、周期计数和外部事件计数。
TGR 能分别用作输入捕捉寄存器和输出比较寄存器。
必须通过引脚功能控制器 (PFC) 设定 MTU2 的外部引脚功能。

(1) 计数器的运行

如果将 TSTR 的 CST0 ~ CST4 位、TSTR_5 的 CSTU5 位、CSTV5 位和 CSTW5 位置 1，对应通道的 TCNT 就开始计数，能进行自由运行计数和周期计数等。

(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 10.4 所示。

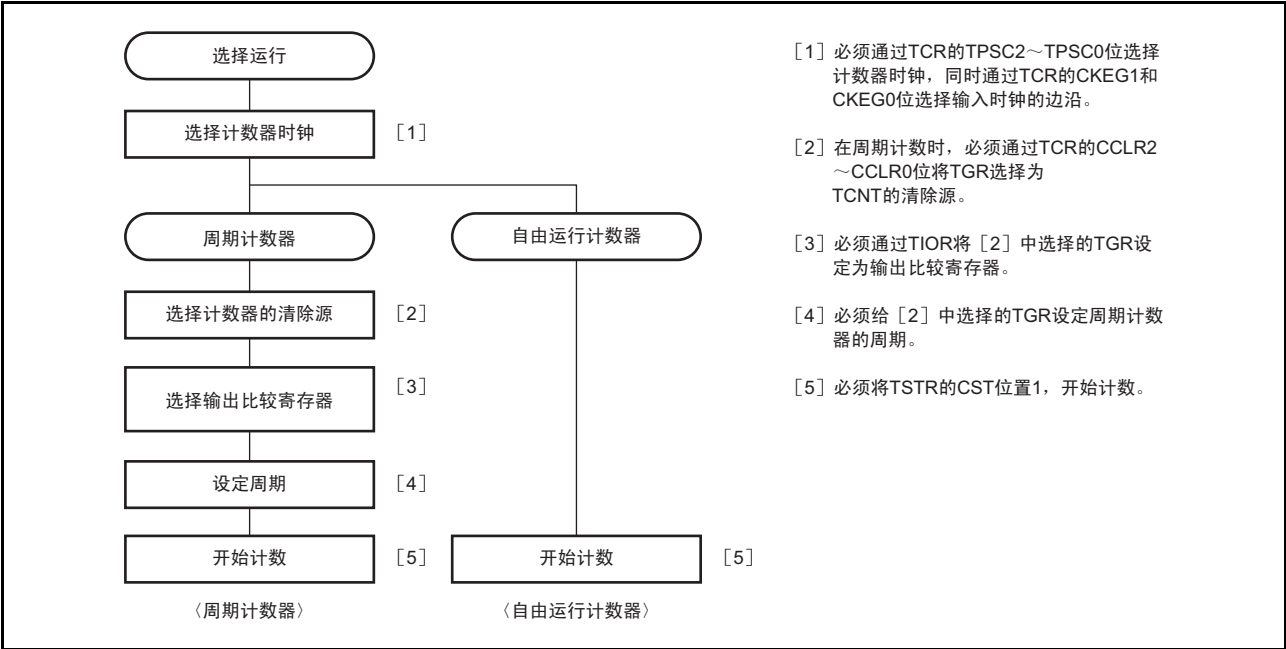


图 10.4 计数器运行的设定步骤例子

(b) 自由运行计数和周期计数

MTU2 的 TCNT 在刚复位后全部设定为自由运行计数器，如果将 TSTR 的对应位置 1，就作为自由运行计数器开始递增计数。当 TCNT 发生上溢 (H'FFFF→H'0000) 时，TSR 的 TCFV 位就被置 1。此时，如果对应 TIER 的 TCIEV 位为 1，MTU2 就请求中断。在 TCNT 发生上溢后，从 H'0000 开始继续递增计数。

自由运行计数器的运行如图 10.5 所示。

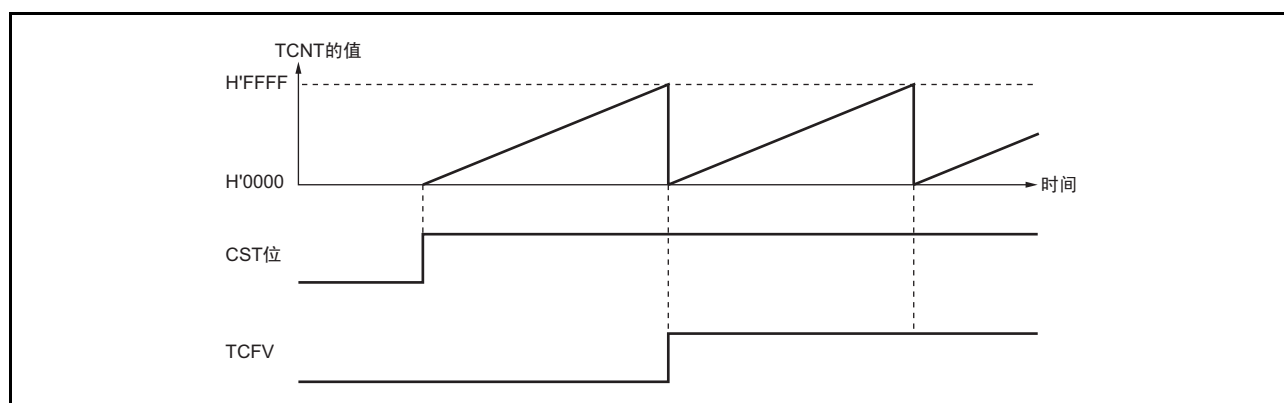


图 10.5 自由运行计数器的运行

当选择比较匹配作为 TCNT 的清除源时，对应通道的 TCNT 就进行周期计数。将用于周期设定的 TGR 设定为输出比较寄存器，并用 TCR 的 CCLR2 ~ CCLR0 位选择通过比较匹配进行计数器清除。在设定后，如果将 TSTR 的对应位置 1，周期计数器就开始递增计数。当计数值和 TGR 的值相同时，TSR 的 TGF 位被置 1，TCNT 被清 0。

此时，如果对应 TIER 的 TGIE 位为 1，MTU2 就请求中断。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 10.6 所示。

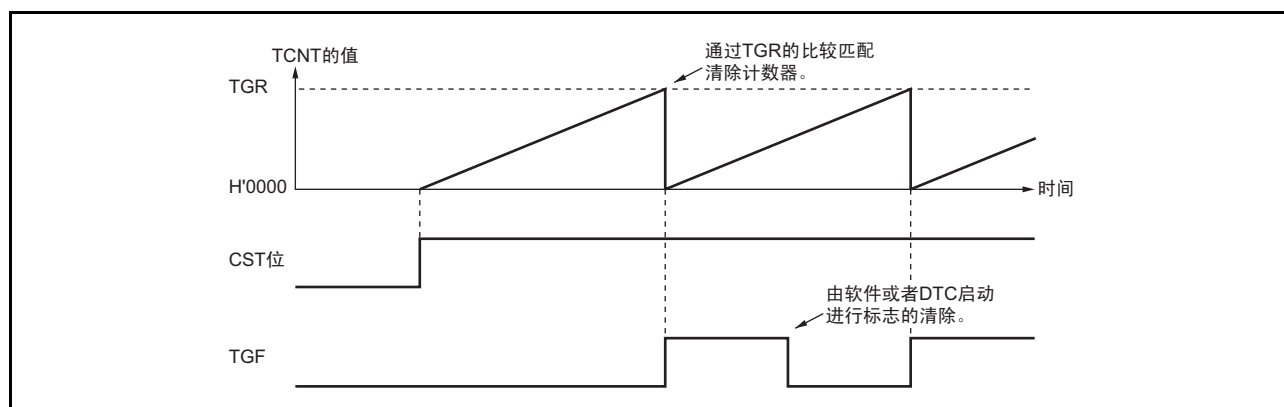


图 10.6 周期计数器的运行

(2) 通过比较匹配进行波形输出的功能

MTU2 能通过比较匹配从对应的输出引脚进行 0 输出 / 1 输出 / 交替输出。

(a) 通过比较匹配进行波形输出的设定步骤例子

通过比较匹配进行波形输出的设定步骤例子如图 10.7 所示。

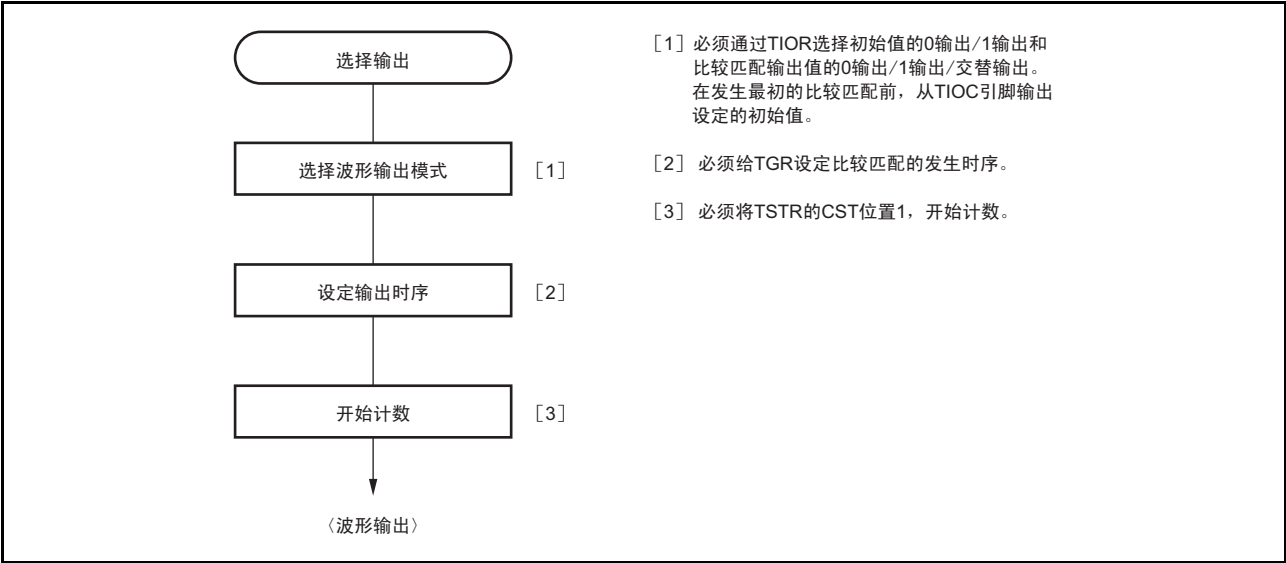


图 10.7 通过比较匹配进行波形输出的运行例子

(b) 波形输出的例子

0 输出 /1 输出的例子如图 10.8 所示。

在此例中，将 TCNT 作为自由运行计数器运行，并设定为通过比较匹配 A 进行 1 输出、通过比较匹配 B 进行 0 输出。当设定的电平和引脚的电平相同时，引脚的电平不变。

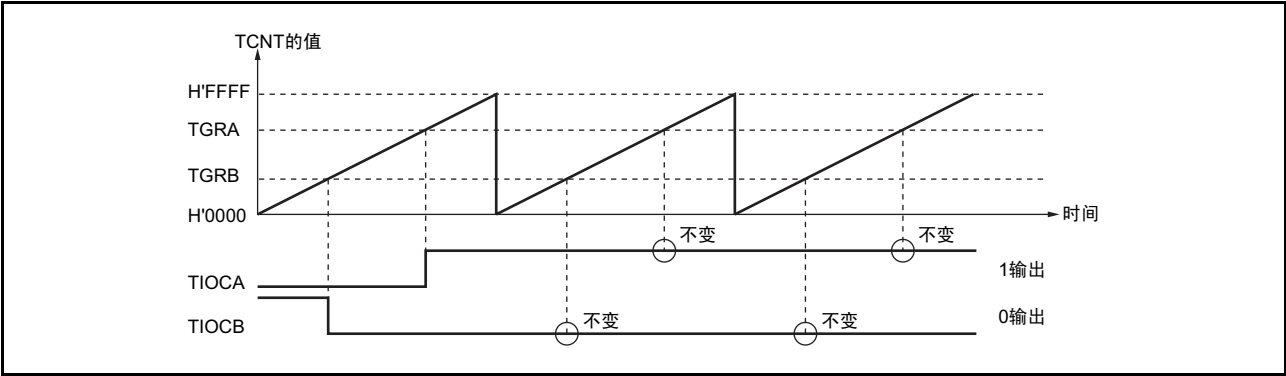


图 10.8 0 输出 /1 输出的运行例子

交替输出的例子如图 10.9 所示。

在此例中，将 TCNT 作为周期计数器运行（通过比较匹配 B 进行计数器清除），并将比较匹配 A 和 B 都设定为交替输出。

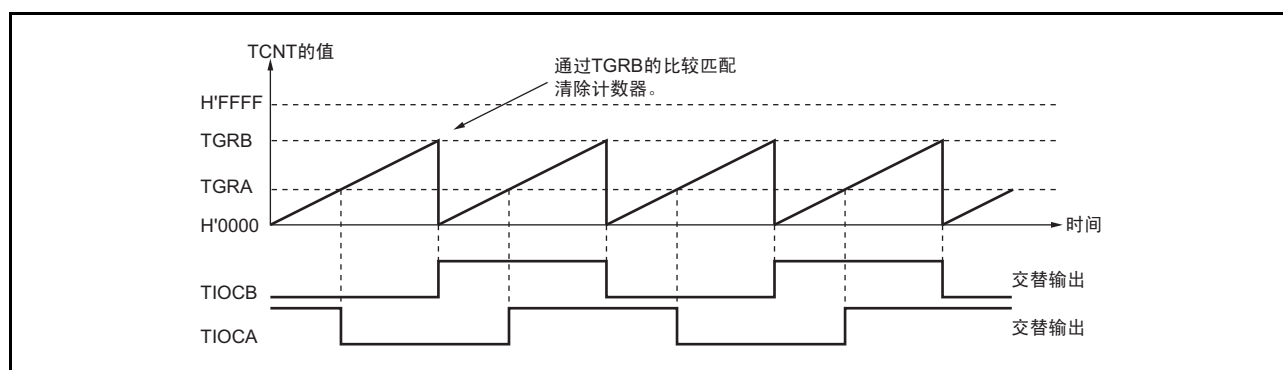


图 10.9 交替输出的运行例子

(3) 输入捕捉功能

能检测 TIOC 引脚的输入边沿，将 TCNT 的值传送到 TGR。

能从上升沿 / 下降沿 / 双边沿中选择检测边沿。通道 0、1 也能将其他通道的计数器输入时钟或者比较匹配信号作为输入捕捉源。

【注】 如果在通道 0、1 中将其他通道的计数器输入时钟作为输入捕捉的输入，就不能选择 MP ϕ /1 作为输入捕捉的计数器输入时钟。如果选择 MP ϕ /1，就不发生输入捕捉。

(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 10.10 所示。

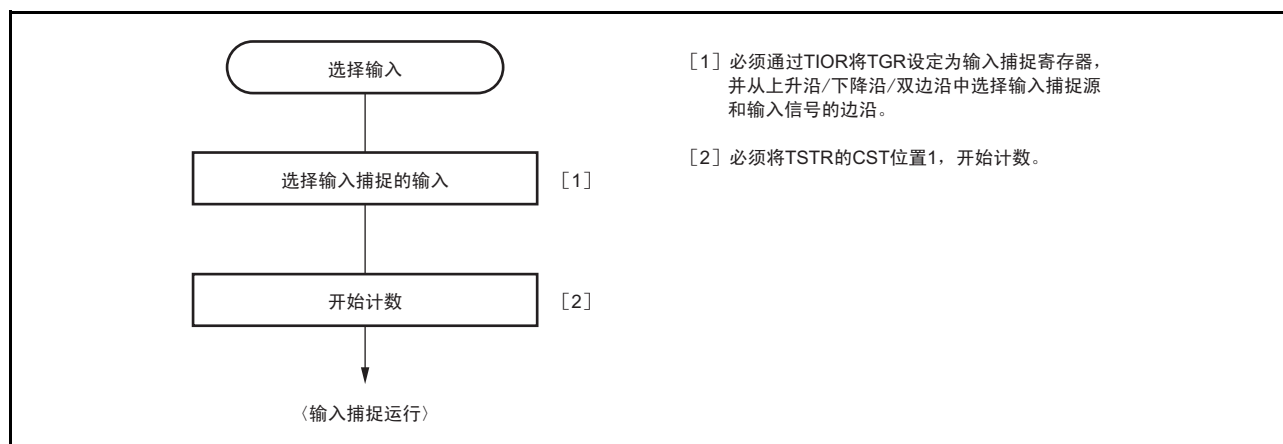


图 10.10 输入捕捉的设定例子

(b) 输入捕捉的例子

输入捕捉的运行例子如图 10.11 所示。

在此例中，选择上升 / 下降的双边沿作为 TIOCA 引脚输入捕捉的输入边沿，选择下降沿作为 TIOCB 引脚输入捕捉的输入边沿，并将 TCNT 设定为通过 TGRB 的输入捕捉进行计数器清除。

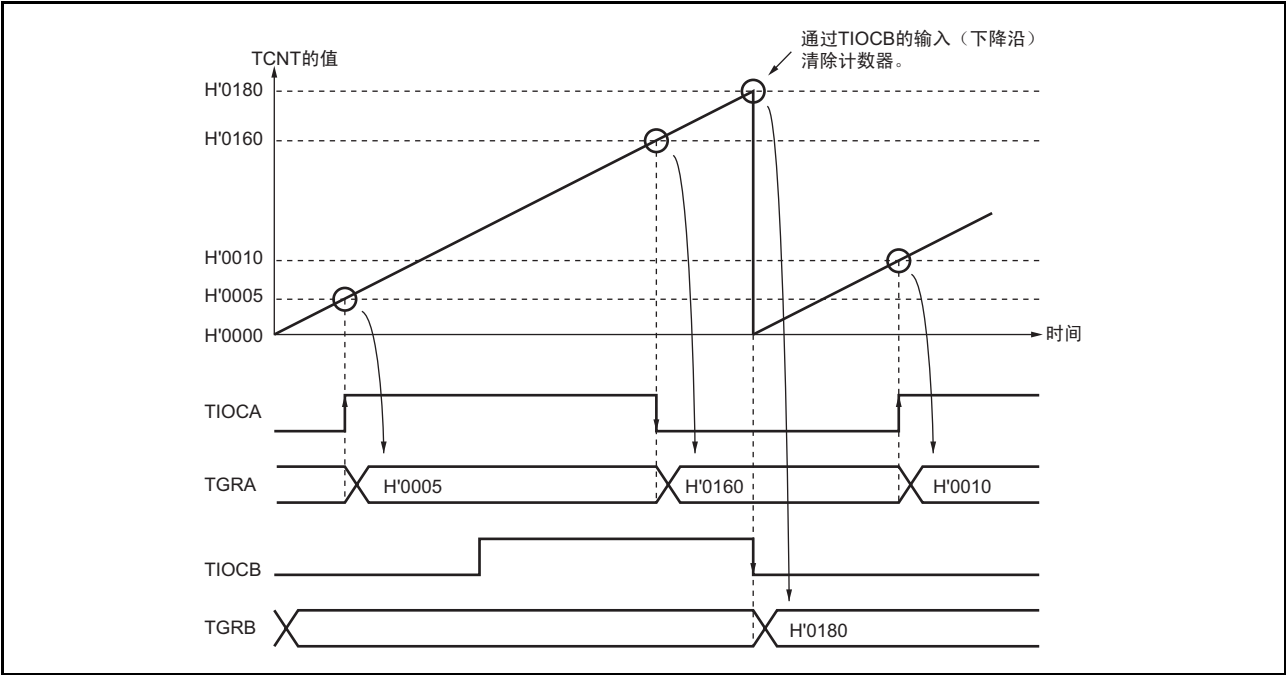


图 10.11 输入捕捉的运行例子

10.4.2 同步运行

同步运行能同时改写多个 TCNT 的值（同步预置），还能通过 TCR 的设定同时清除多个 TCNT（同步清除）。

能通过同步运行增加 1 个时基要运行的 TGR 个数。

通道 0 ~ 4 都能设定为同步运行。

通道 5 不能进行同步运行。

(1) 同步运行的设定步骤例子

同步运行的设定步骤例子如图 10.12 所示。

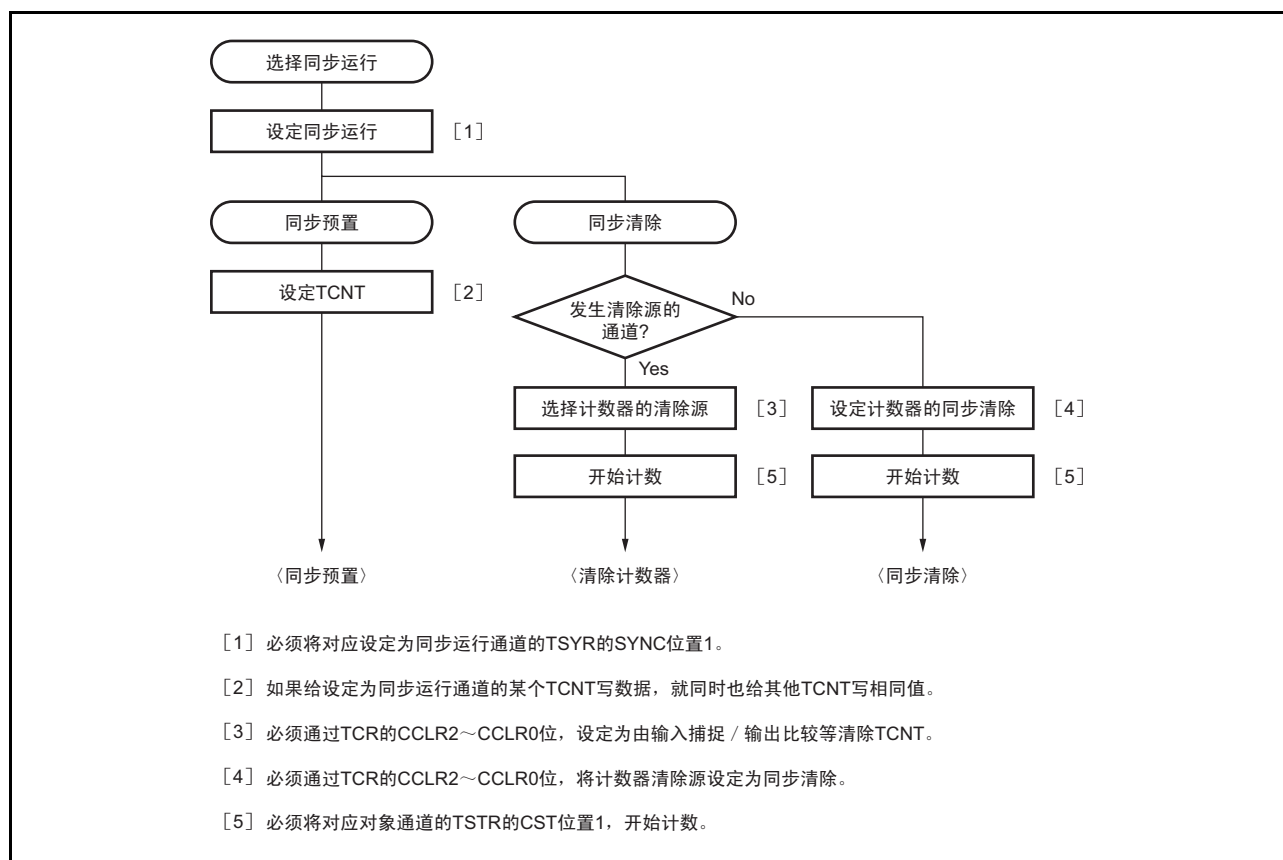


图 10.12 同步运行的设定步骤例子

(2) 同步运行的例子

同步运行的例子如图 10.13 所示。

在此例中，设定通道 0 ~ 2 为同步运行和 PWM 模式 1，通道 0 的计数器清除源为 TGRB_0 的比较匹配，通道 1、2 的计数器清除源为同步清除。

从 TI0C0A、TI0C1A 和 TI0C2A 引脚输出 3 相的 PWM 波形。此时，通道 0 ~ 2 的 TCNT 进行同步预置或者通过 TGRB_0 的比较匹配进行同步清除，TGRB_0 设定的数据为 PWM 周期。

有关 PWM 模式请参照“10.4.5 PWM 模式”。

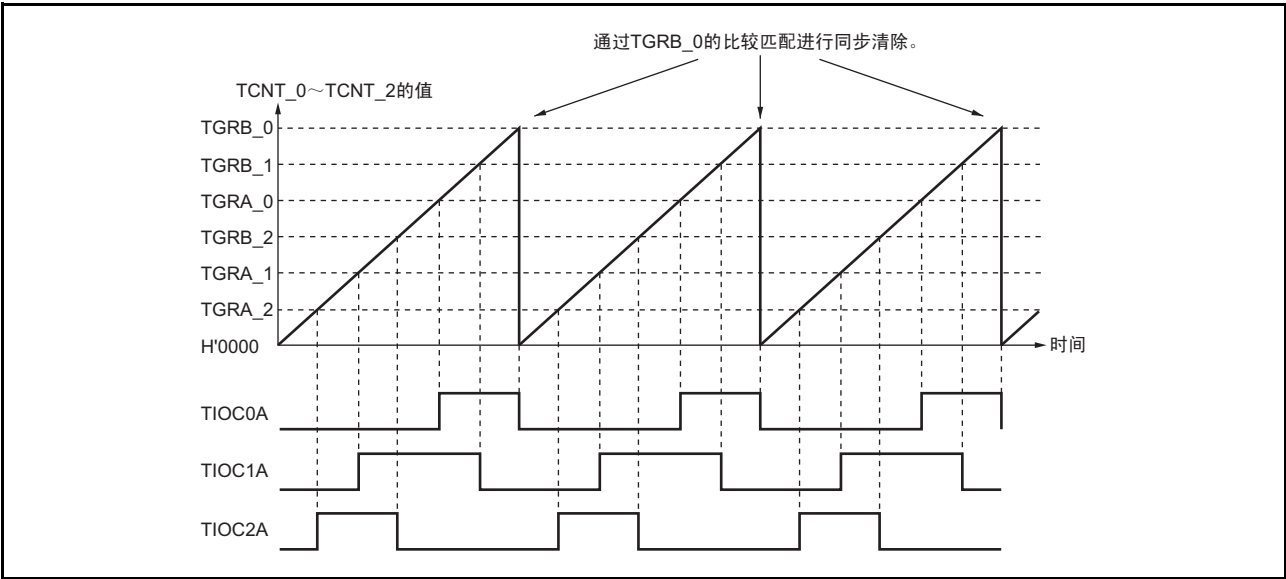


图 10.13 同步运行的例子

10.4.3 缓冲运行

缓冲运行是通道 0、3、4 具有的功能。TGRC 和 TGRD 能用作缓冲寄存器，并且通道 0 的 TGRF 也能用作缓冲寄存器。

根据 TGR 是设定为输入捕捉寄存器还是设定为比较匹配寄存器，缓冲运行的内容不同。

【注】 TGRE_0 不设定为输入捕捉寄存器，而只能作为比较匹配寄存器运行。

缓冲运行时的寄存器组合如表 10.43 所示。

表 10.43 寄存器的组合

通道	定时器的通用寄存器	缓冲寄存器
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0
	TGRE_0	TGRF_0
3	TGRA_3	TGRC_3
	TGRB_3	TGRD_3
4	TGRA_4	TGRC_4
	TGRB_4	TGRD_4

• TGR用作输出比较寄存器的情况

如果发生比较匹配，就将对应通道的缓冲寄存器的值传送到定时器的通用寄存器。
此运行如图 10.14 所示。

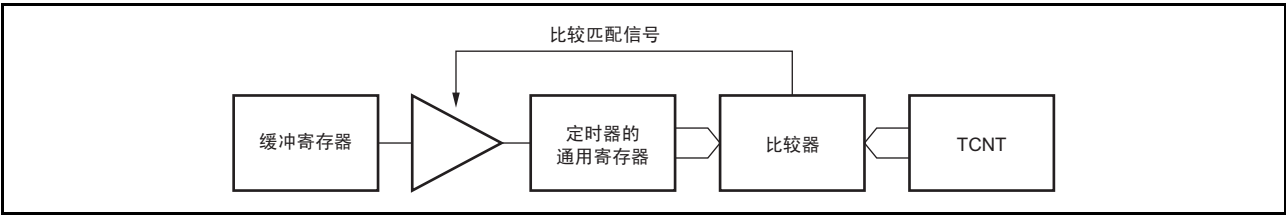


图 10.14 比较匹配的缓冲运行

- TGR用作输入捕捉寄存器的情况
如果发生输入捕捉，就在将 TCNT 的值传送到 TGR 的同时，将以前保存在 TGR 的值传送到缓冲寄存器。此运行如图 10.15 所示。

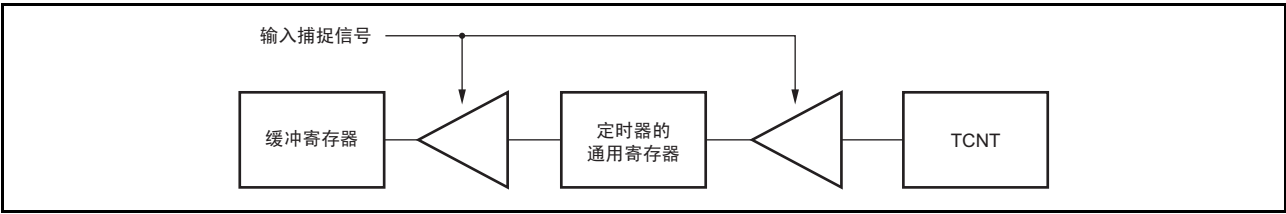


图 10.15 输入捕捉的缓冲运行

(1) 缓冲运行的设定步骤例子

缓冲运行的设定步骤例子如图 10.16 所示。

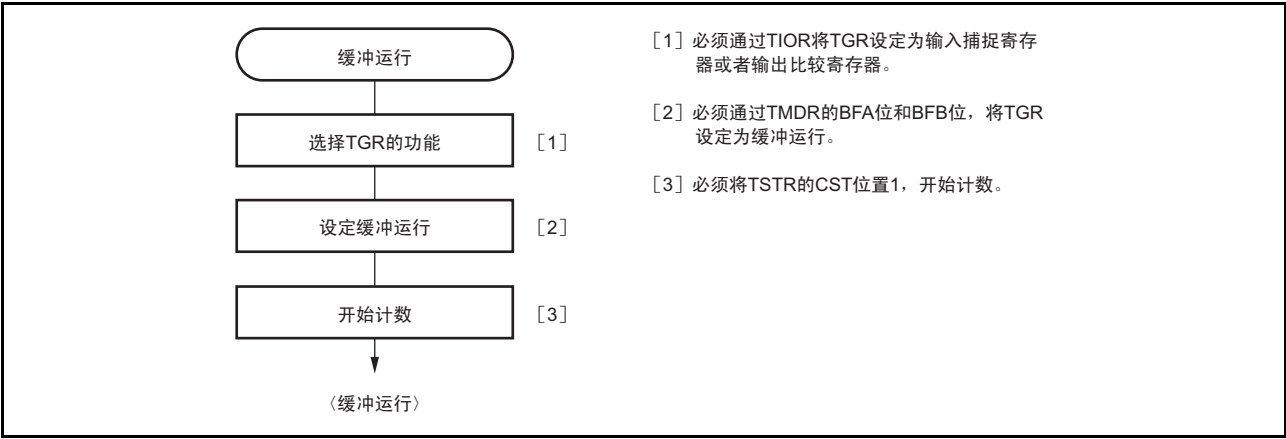


图 10.16 缓冲运行的设定步骤例子

(2) 缓冲运行的例子

(a) TGR 用作输出比较寄存器的情况

将通道 0 设定为 PWM 模式 1 以及 TGRA 和 TGRC 设定为缓冲运行时的运行例子如图 10.17 所示。在此例中，通过比较匹配 B 进行 TCNT 清除、通过比较匹配 A 进行 1 输出、通过比较匹配 B 进行 0 输出，并将 TBTM 的 TTSA 位置 0。

因为设定了缓冲运行，所以当发生比较匹配 A 时，就在输出发生变化的同时，将缓冲寄存器 TGRC 的值传送到定时器的通用寄存器 TGRA。每当发生比较匹配 A 时，重复此运行。

有关 PWM 模式请参照“10.4.5 PWM 模式”。

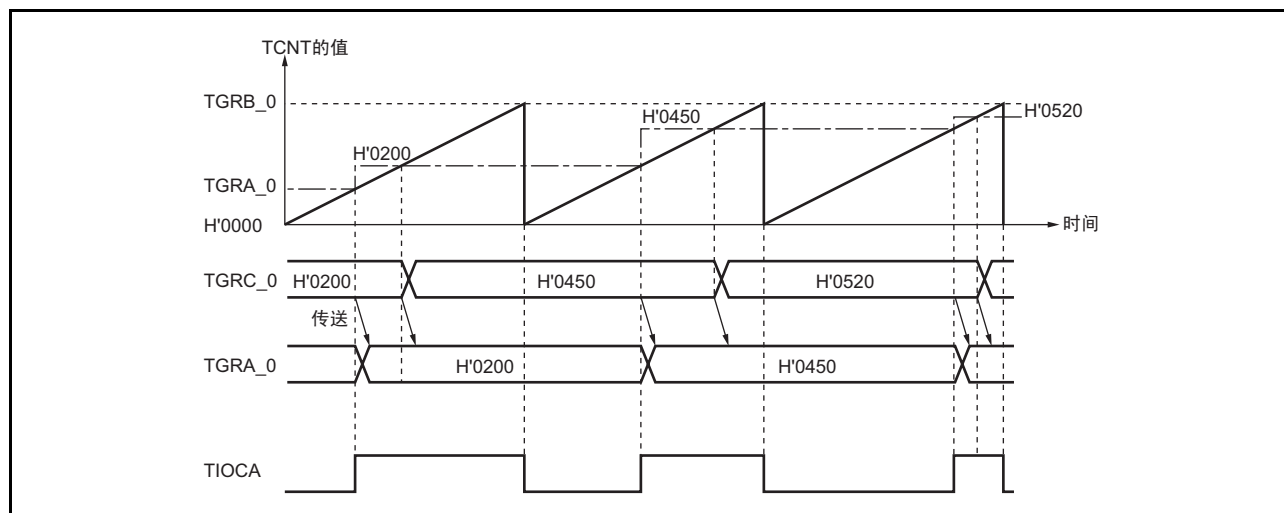


图 10.17 缓冲运行的例子 (1)

(b) TGR 用作输入捕捉寄存器的情况

将 TGRA 设定为输入捕捉寄存器以及 TGRA 和 TGRC 设定为缓冲运行时的运行例子如图 10.18 所示。

TCNT 通过 TGRA 的输入捕捉进行计数器清除，选择上升/下降的双边沿作为 TIOCA 引脚输入捕捉的输入边沿。

因为设定了缓冲运行，所以在通过输入捕捉 A 将 TCNT 的值保存到 TGRA 的同时，将以前保存在 TGRA 的值传送到 TGRC。

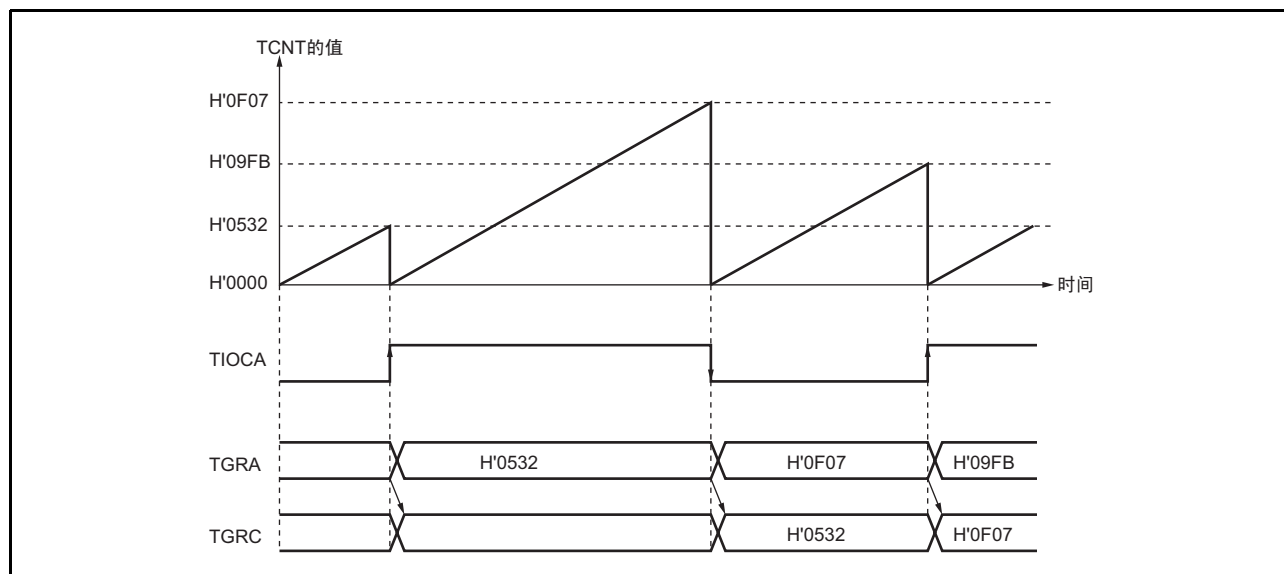


图 10.18 缓冲运行的例子 (2)

(3) 选择缓冲运行时的缓冲寄存器到定时器的通用寄存器的传送时序

能通过设定缓冲运行的传送模式寄存器 (TBTM_0、TBTM_3 和 TBTM_4)，选择缓冲寄存器 (通道 0 为 PWM 模式 1、2 的缓冲寄存器；通道 3、4 为 PWM 模式 1 的缓冲寄存器) 到定时器的通用寄存器的传送时序。能选择的缓冲传送时序是在发生比较匹配时 (初始值) 或者清除 TCNT 时的时序。在此，所谓清除 TCNT 时是指以下的任意条件成立时：

- 当 TCNT 发生上溢 (H'FFFF→H'0000) 时
- 在计数器运行中，给 TCNT 写 H'0000 时
- 当通过 TCR 的 CCLR2～CCLR0 位设定的清除源将 TCNT 清除为 H'0000 时

【注】 必须在 TCNT 停止运行的状态下进行 TBTM 的设定。

将通道 0 设定为 PWM 模式 1 以及 TGRA_0 和 TGRC_0 设定为缓冲运行时的运行例子如图 10.19 所示。在此例中，通过比较匹配 B 进行 TCNT_0 清除、通过比较匹配 A 进行 1 输出、通过比较匹配 B 进行 0 输出，并将 TBTM_0 的 TTSA 位置 1。

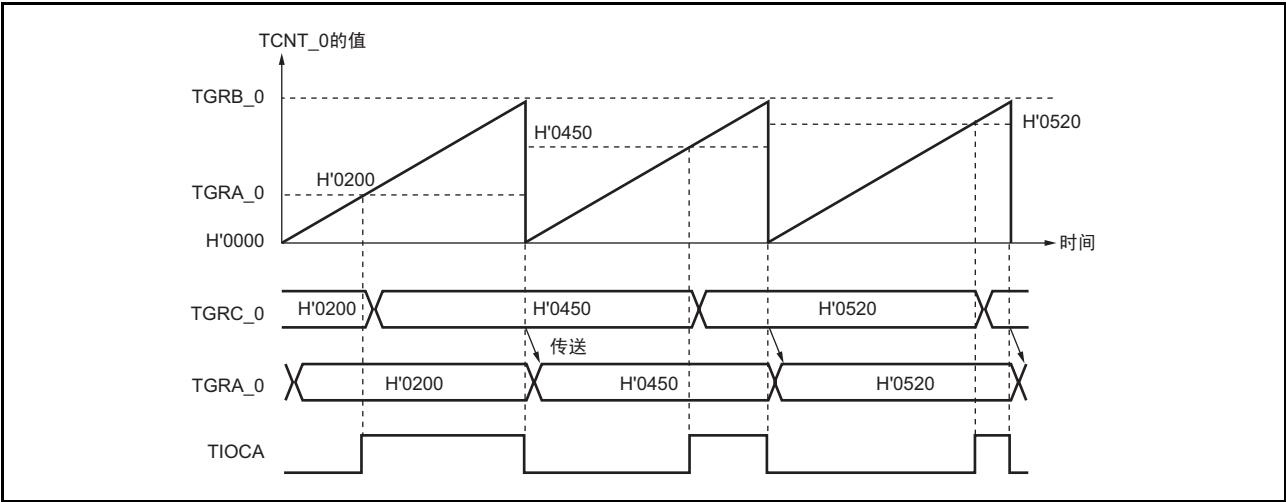


图 10.19 将清除 TCNT_0 作为 TGRC_0 到 TGRA_0 的缓冲传送时序的运行例子

10.4.4 级联运行

级联运行是将 2 个通道的 16 位计数器连接为 32 位计数器的功能。

用 TCR 的 TPSC2～TPSC0 位将通道 1 的计数器时钟设定为通过 TCNT_2 的上溢 / 下溢进行计数，运行此功能。

只在低 16 位的 TCNT 为相位计数模式时发生下溢。

级联的组合如表 10.44 所示。

【注】 如果将通道 1 设定为相位计数模式，计数器时钟的设定就无效，并在相位计数模式中独立运行。

表 10.44 级联的组合

组合	高 16 位	低 16 位
通道 1 和通道 2	TCNT_1	TCNT_2

如果在级联运行时 TCNT_1 和 TCNT_2 同时进行输入捕捉，就能通过输入捕捉控制寄存器 (TICCR) 的设置，将输入引脚追加到输入捕捉条件。有关级联时的输入捕捉，请参照“10.7.22 级联中 TCNT_1 和 TCNT_2 的同时输入捕捉”。

TICCR 的设定值和输入捕捉的输入引脚的对应如表 10.45 所示。

表 10.45 TICCR 的设定值和输入捕捉的输入引脚的对应

对象输入捕捉	TICCR 的设定值	输入捕捉的输入引脚
从 TCNT_1 到 TGRA_1 的输入捕捉	I2AE 位 =0 (初始值)	TIOC1A
	I2AE 位 =1	TIOC1A、TIOC2A
从 TCNT_1 到 TGRB_1 的输入捕捉	I2BE 位 =0 (初始值)	TIOC1B
	I2BE 位 =1	TIOC1B、TIOC2B
从 TCNT_2 到 TGRA_2 的输入捕捉	I1AE 位 =0 (初始值)	TIOC2A
	I1AE 位 =1	TIOC2A、TIOC1A
从 TCNT_2 到 TGRB_2 的输入捕捉	I1BE 位 =0 (初始值)	TIOC2B
	I1BE 位 =1	TIOC2B、TIOC1B

(1) 级联运行的设定步骤例子

级联运行的设定步骤例子如图 10.20 所示。

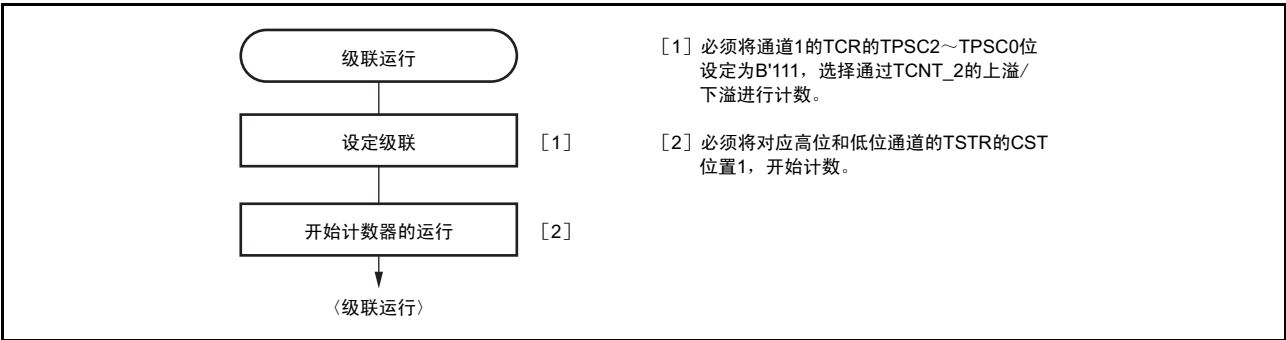


图 10.20 级联运行的设定步骤

(2) 级联运行的例子 (a)

TCNT_1 通过 TCNT_2 的上溢 / 下溢进行计数，并将通道 2 设定为相位计数模式时的运行如图 10.21 所示。TCNT_1 通过 TCNT_2 的上溢进行递增计数并通过 TCNT_2 的下溢进行递减计数。

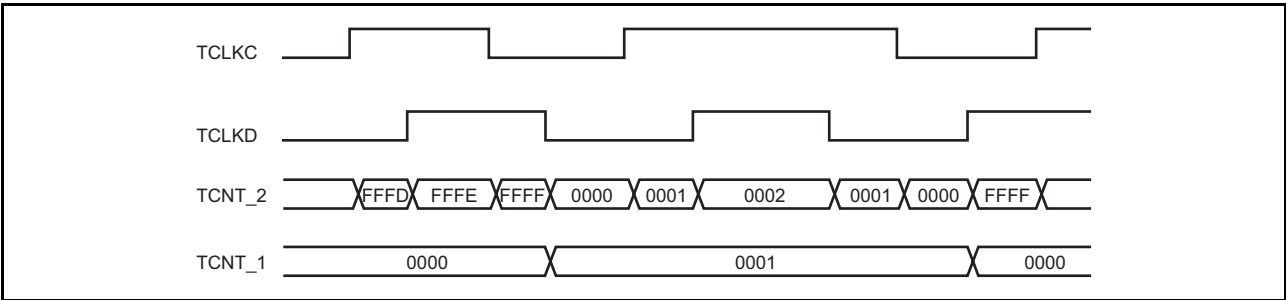


图 10.21 级联的运行例子 (a)

(3) 级联运行的例子 (b)

将 TCNT_1 和 TCNT_2 进行级联, TICCR 的 I2AE 位置 1, 并且将 TIOC2A 引脚追加到 TGRA_1 的输入捕捉条件时的运行如图 10.22 所示。在此例中, 将 TIOR_1 的 IOA0 ~ IOA3 设定为在 TIOC1A 的上升沿进行输入捕捉, TIOR_2 的 IOA0 ~ IOA3 设定为在 TIOC2A 的上升沿进行输入捕捉。

此时, 将 TIOC1A 和 TIOC2A 的上升沿设定为 TGRA_1 的输入捕捉条件, TIOC2A 的上升沿设定为 TGRA_2 的输入捕捉条件。

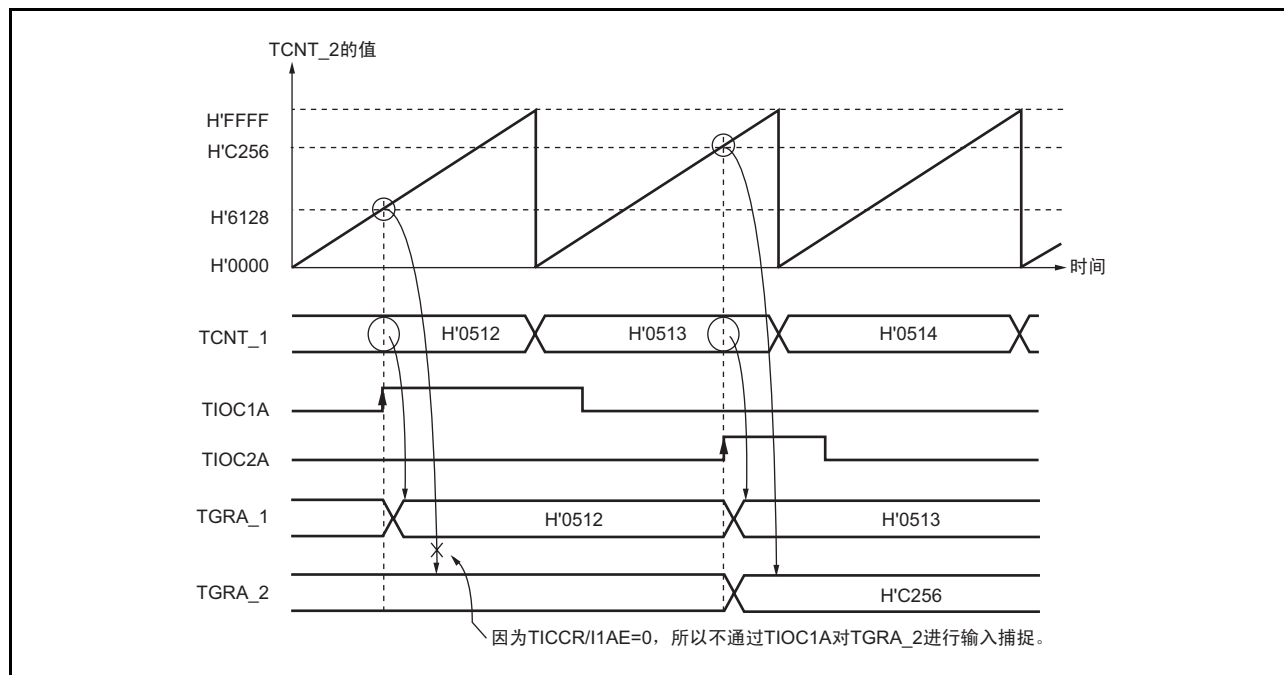


图 10.22 级联的运行例子 (b)

(4) 级联运行的例子 (c)

将 TCNT_1 和 TCNT_2 进行级联, TICCR 的 I2AE 位和 I1AE 位置 1, TIOC2A 引脚追加到 TGRA_1 的输入捕捉条件, 并且将 TIOC1A 引脚追加到 TGRA_2 的输入捕捉条件时的运行如图 10.23 所示。在此例中, 将 TIOR_1 和 TIOR_2 的 IOA0 ~ IOA3 都设定为在 TIOC1A 和 TIOC2A 的双边沿进行输入捕捉。此时, TIOC1A 和 TIOC2A 输入的 OR 为 TGRA_1 和 TGRA_2 的输入捕捉条件。

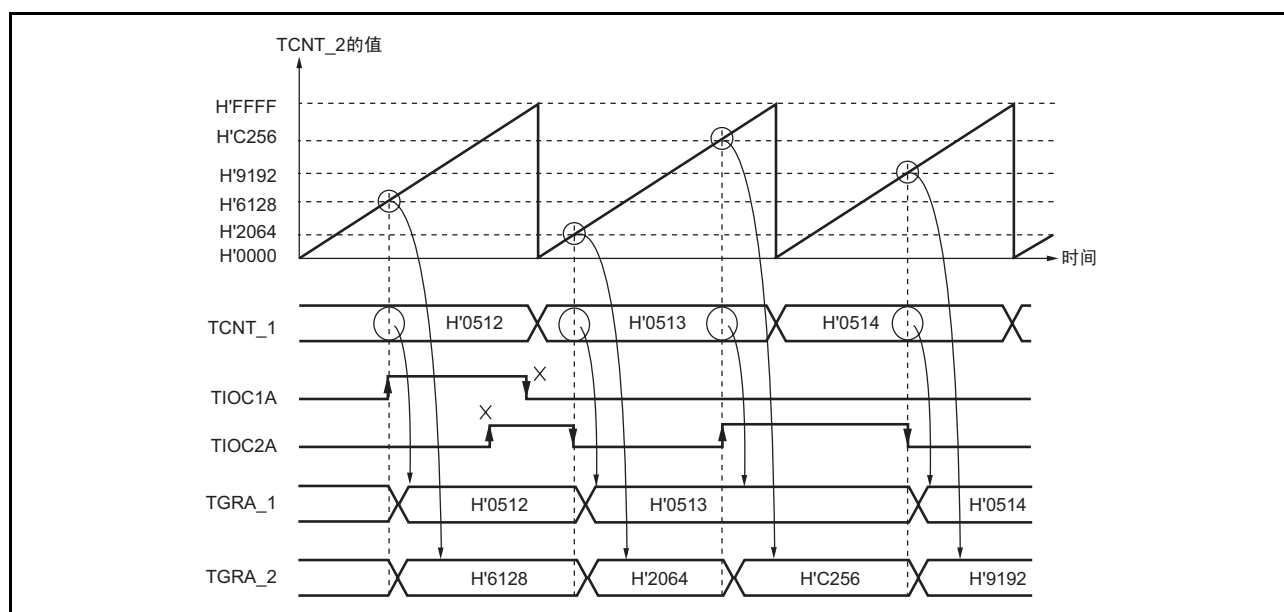


图 10.23 级联的运行例子 (c)

(5) 级联运行的例子 (d)

将 TCNT_1 和 TCNT_2 进行级联, TICCIR 的 I2AE 位置 1, 并且将 TIOC2A 引脚追加到 TGRA_1 的输入捕捉条件时的运行如图 10.24 所示。在此例中, 将 TIOR_1 的 IOA0 ~ IOA3 设定为通过 TGRA_0 的比较匹配 / 输入捕捉进行输入捕捉, 将 TIOR_2 的 IOA0 ~ IOA3 设定为在 TIOC2A 的上升沿进行输入捕捉。

此时, 因为 TIOR_1 设定为通过 TGRA_0 的比较匹配 / 输入捕捉进行输入捕捉, 所以, 即使将 TICCIR 的 I2AE 位置 1, TIOC2A 的边沿也不作为 TGRA_1 的输入捕捉条件。

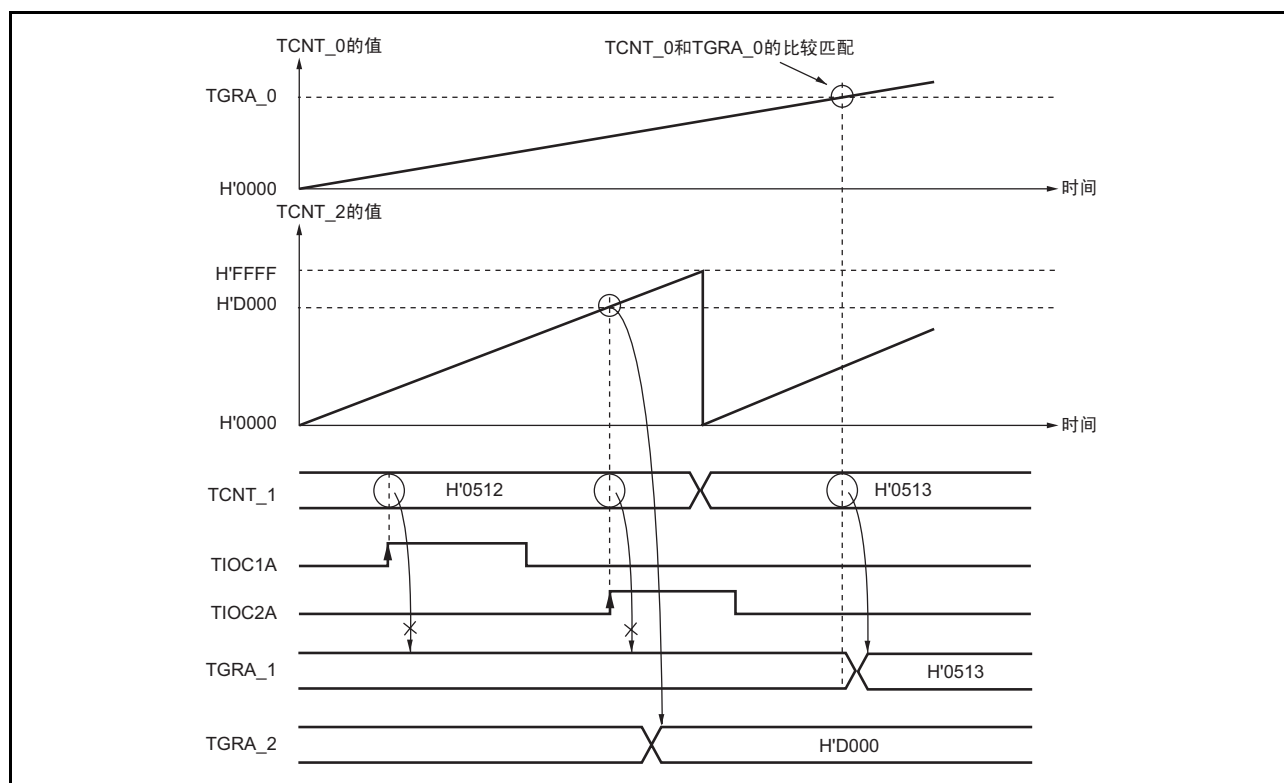


图 10.24 级联的运行例 (d)

10.4.5 PWM 模式

PWM 模式是从输出引脚输出各种 PWM 波形的模式。能从 0 输出 / 1 输出 / 交替输出中选择各 TGR 比较匹配的输电平。

能通过设定各 TGR，输出占空比为 0 ~ 100% 的 PWM 波形。

能通过将 TGR 的比较匹配设定为计数器清除源，给该寄存器设定周期。全部通道能独立设定为 PWM 模式，也能进行同步运行。

PWM 模式有以下 2 种：

(a) PWM 模式 1

配对使用 TGRA 和 TGRB、TGRC 和 TGRD，从 TIOCA 和 TIOCC 引脚生成 PWM 输出。通过比较匹配 A、C 从 TIOCA 和 TIOCC 引脚进行 TIOR 的 IOA3 ~ IOA0 和 IOC3 ~ IOC0 位指定的输出，并通过比较匹配 B、D 进行 TIOR 的 IOB3 ~ IOB0 和 IOD3 ~ IOD0 位指定的输出。设定在 TGRA 和 TGRC 的值为初始输出值。如果配对使用的 TGR 设定值相同，即使发生比较匹配，输出值也不变。

在 PWM 模式 1 中最多能进行 8 相的 PWM 输出。

(b) PWM 模式 2

将 1 个 TGR 用于周期寄存器，其他 TGR 用于占空比寄存器，生成 PWM 输出。通过比较匹配进行 TIOR 指定的输出。另外，在通过同步寄存器的比较匹配进行计数器清除后，各引脚的输出值为 TIOR 设定的初始值。如果周期寄存器和占空比寄存器的设定值相同，即使发生比较匹配，输出值也不变。

在 PWM 模式 2 中，能通过同步运行的并用进行最多 8 相的 PWM 输出。

PWM 输出引脚和寄存器的对应如表 10.46 所示。

表 10.46 各 PWM 输出的寄存器和输出引脚

通道	寄存器	输出引脚		
		PWM 模式 1	PWM 模式 2	
0	TGRA_0	TIOC0A	TIOC0A	
	TGRB_0		TIOC0B	
	TGRC_0	TIOC0C	TIOC0C	
	TGRD_0		TIOC0D	
1	TGRA_1	TIOC1A	TIOC1A	
	TGRB_1		TIOC1B	
2	TGRA_2	TIOC2A	TIOC2A	
	TGRB_2		TIOC2B	
3	TGRA_3	TIOC3A	不能设定	
	TGRB_3	TIOC3C		
	TGRC_3			
	TGRD_3			
4	TGRA_4	TIOC4A		
	TGRB_4	TIOC4C		
	TGRC_4			
	TGRD_4			

【注】 在 PWM 模式 2 中，不能对已设定周期的 TGR 进行 PWM 输出。

(1) PWM 模式的设定步骤例子

PWM 模式的设定步骤例子如图 10.25 所示。

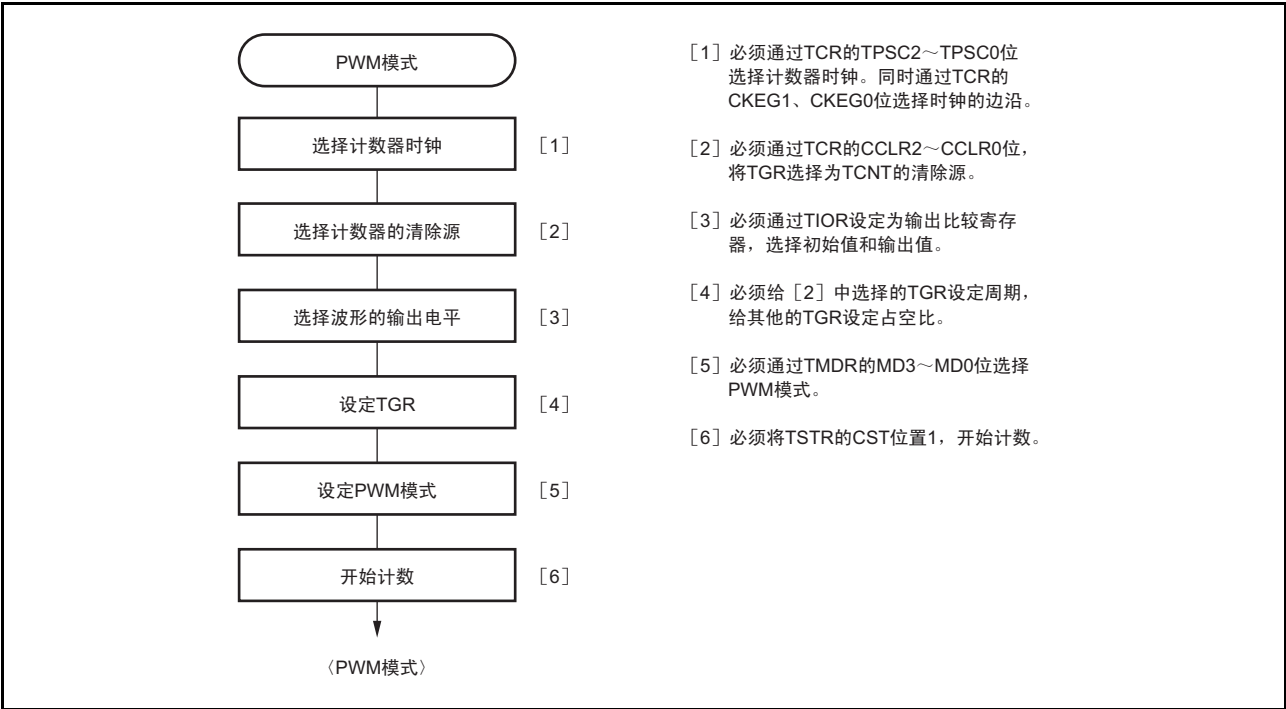


图 10.25 PWM 模式的设定步骤例子

(2) PWM 模式的运行例子

PWM 模式 1 的运行例子如图 10.26 所示。

在此例中，将 TGRA 的比较匹配作为 TCNT 的清除源，并将 TGRA 的初始输出值和输出值置 0、TGRB 的输出值置 1。

此时，设定在 TGRA 的值为周期，设定在 TGRB 的值为占空比。

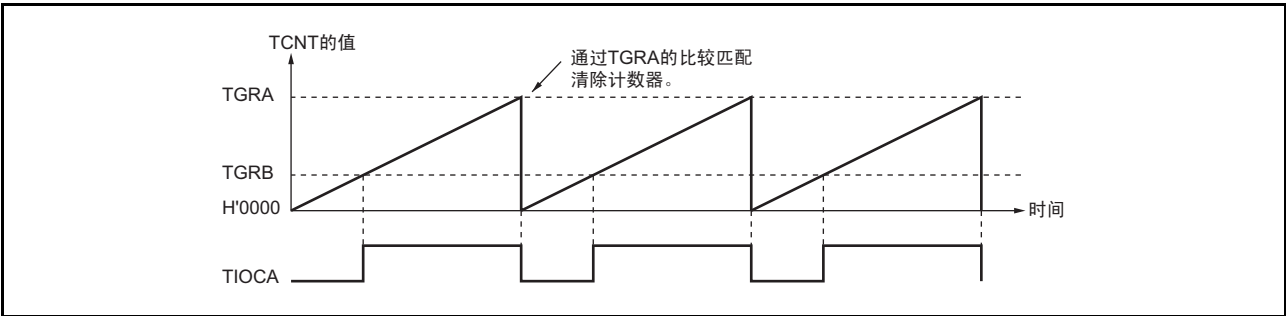


图 10.26 PWM 模式的运行例子

PWM 模式 2 的运行例子如图 10.27 所示。

在此例中，通道 0 和 1 同步运行，将 TGRB_1 的比较匹配作为 TCNT 的清除源，并将其他的 TGR (TGRA_0 ~ TGRD_0、TGRA_1) 的初始输出值置 0、输出值置 1，输出 5 相的 PWM 波形。

此时，设定在 TGR1B 的值为周期，设定在其他 TGR 的值为占空比。

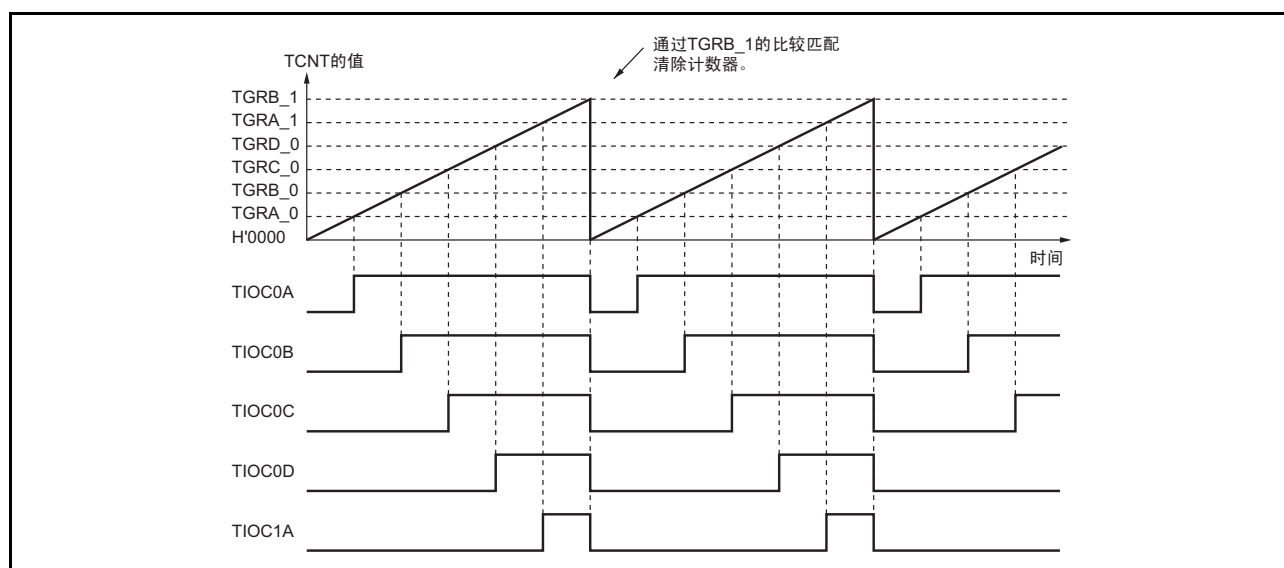


图 10.27 PWM 模式的运行例子

在 PWM 模式中，输出占空比为 0% 和 100% 的 PWM 波形的例子如图 10.28 所示。

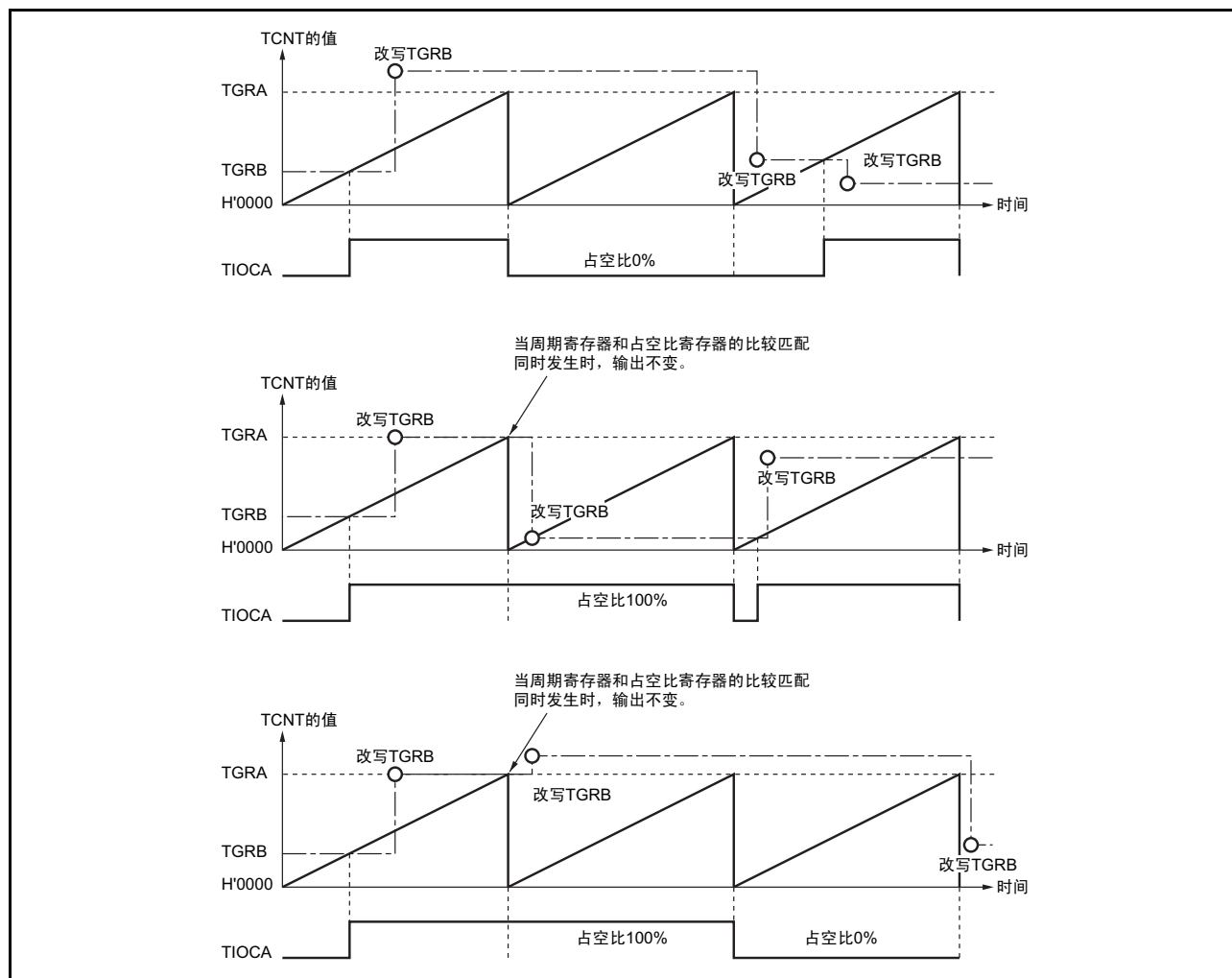


图 10.28 PWM 模式的运行例子

10.4.6 相位计数模式

相位计数模式通过设定通道 1、2 检测 2 个外部时钟输入的相位差，使 TCNT 进行递增 / 递减计数。

当设定为相位计数模式时，与 TCR 的 TPSC2 ~ TPSC0 位、CKEG1 位和 CKEG0 位的设定无关，选择外部时钟作为计数器输入时钟，TCNT 作为递增 / 递减计数器运行。TCR 的 CCLR1 位、CCLR0 位、TIOR 位、TIER 位和 TGR 位的功能有效，因此能使用输入捕捉 / 比较匹配功能和中断功能。

能用作 2 相编码器脉冲的输入。

在 TCNT 进行递增计数时，如果发生上溢，TSR 的 TCFV 标志就被置位；在 TCNT 进行递减计数时，如果发生下溢，TCFU 标志就被置位。

TSR 的 TCFD 位是计数方向标志。能通过读 TCFD 标志，确认 TCNT 是在进行递增计数还是在进行递减计数。

外部时钟引脚和通道的对应如表 10.47 所示。

表 10.47 相位计数模式时钟的输入引脚

通道	外部时钟引脚	
	A 相	B 相
将通道 1 设定为相位计数模式	TCLKA	TCLKB
将通道 2 设定为相位计数模式	TCLKC	TCLKD

(1) 相位计数模式的设定步骤例子

相位计数模式的设定步骤例子如图 10.29 所示。

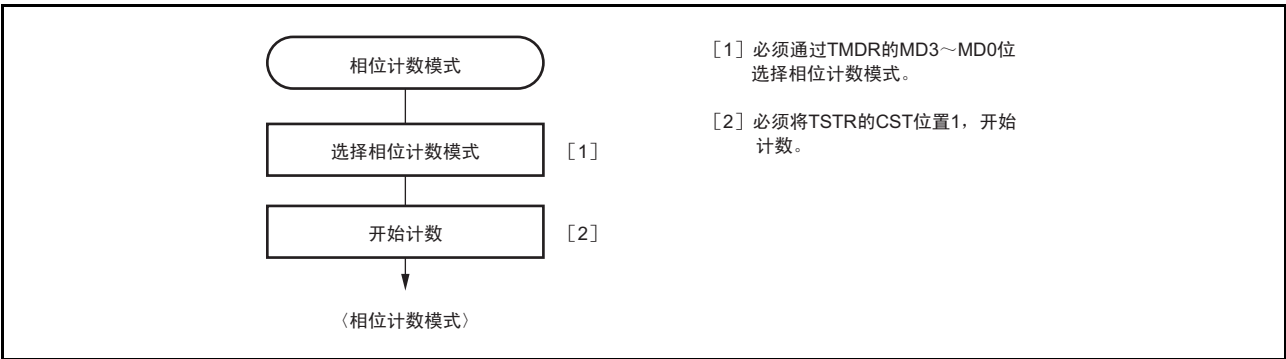


图 10.29 相位计数模式的设定步骤例子

(2) 相位计数模式的运行例子

在相位计数模式中，TCNT 通过 2 个外部时钟的相位差进行递增 / 递减计数。根据计数条件，有 4 种模式。

(a) 相位计数模式 1

相位计数模式 1 的运行例子和 TCNT 递增 / 递减计数的条件分别如图 10.30 和表 10.48 所示。

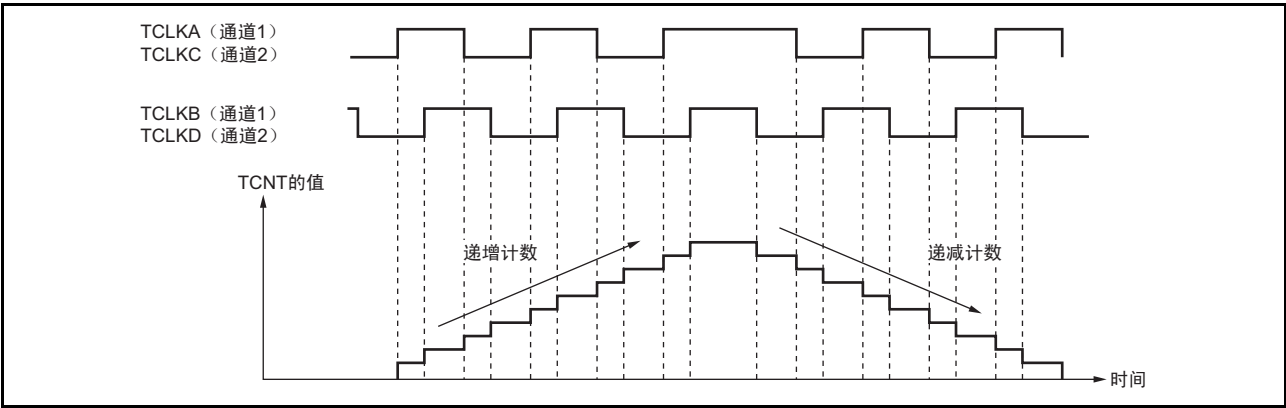


图 10.30 相位计数模式 1 的运行例子

表 10.48 相位计数模式 1 的递增 / 递减计数的条件

TCLKA (通道 1)	TCLKC (通道 2)	TCLKB (通道 1)	TCLKD (通道 2)	运行内容
高电平		上升沿		递增计数
低电平		下降沿		
上升沿		低电平		
下降沿		高电平		
高电平		下降沿		递减计数
低电平		上升沿		
上升沿		高电平		
下降沿		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 的运行例子和 TCNT 递增 / 递减计数的条件分别如图 10.31 和如表 10.49 所示。

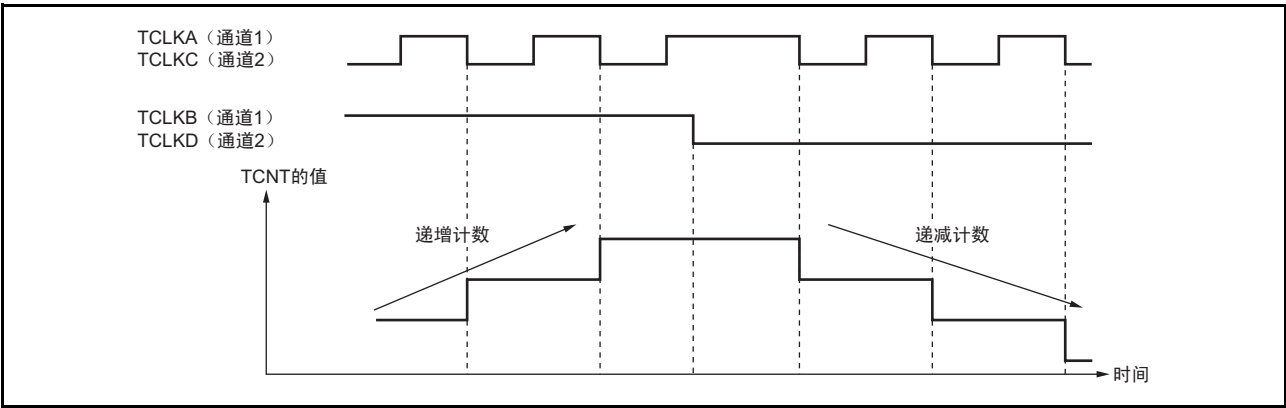


图 10.31 相位计数模式 2 的运行例

表 10.49 相位计数模式 2 的递增 / 递减计数的条件

TCLKA (通道 1) TCLKC (通道 2)	TCLKB (通道 1) TCLKD (通道 2)	运行内容
高电平	上升沿	不计数 (Don't care)
低电平	下降沿	不计数 (Don't care)
上升沿	低电平	不计数 (Don't care)
下降沿	高电平	递增计数
高电平	下降沿	不计数 (Don't care)
低电平	上升沿	不计数 (Don't care)
上升沿	高电平	不计数 (Don't care)
下降沿	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 的运行例子和 TCNT 递增 / 递减计数的条件分别如图 10.32 和如表 10.50 所示。

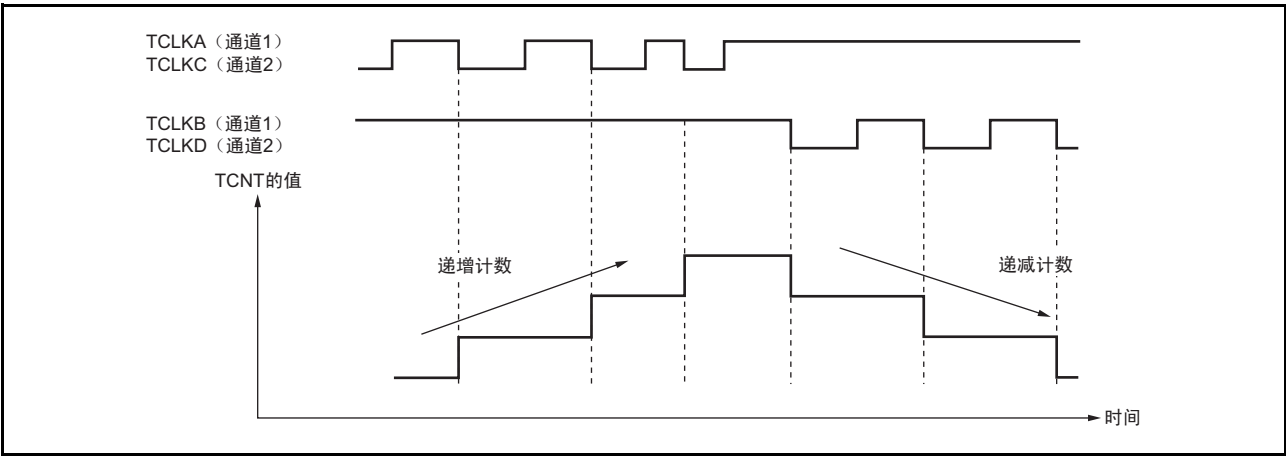


图 10.32 相位计数模式 3 的运行例子

表 10.50 相位计数模式 3 的递增 / 递减计数的条件

TCLKA (通道 1) TCLKC (通道 2)	TCLKB (通道 1) TCLKD (通道 2)	运行内容
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 的运行例子和 TCNT 递增 / 递减计数的条件分别如图 10.33 和如表 10.51 所示。

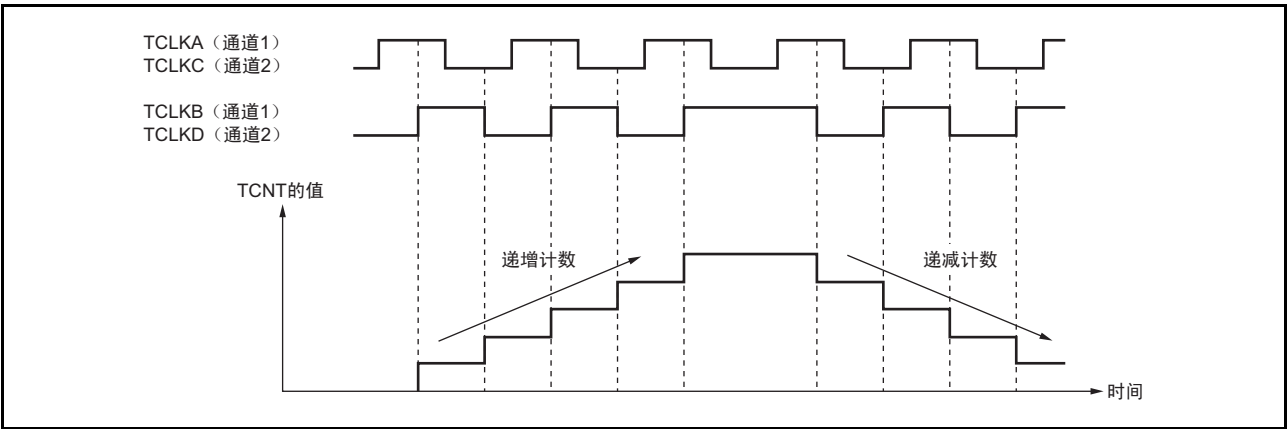


图 10.33 相位计数模式 4 的运行例子

表 10.51 相位计数模式 4 的递增 / 递减计数的条件

TCLKA（通道 1） TCLKC（通道 2）	TCLKB（通道 1） TCLKD（通道 2）	运行内容
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

将通道 1 设定为相位计数模式，并与通道 0 配合输入伺服马达的 2 相编码器脉冲后检测位置或者速度的例子如图 10.34 所示。

将通道 1 设定为相位计数模式 1，给 TCLKA 和 TCLKB 输入编码器脉冲的 A 相和 B 相。

通道 0 通过 TGRC_0 的比较匹配清除 TCNT 计数器， TGRA_0 和 TGRC_0 用于比较匹配功能，并设定速度控制周期和位置控制周期； TGRB_0 用于输入捕捉功能，并使 TGRB_0 和 TGRD_0 进行缓冲运行。将 TGRB_0 的输入捕捉源作为通道 1 的计数器输入时钟，检测 2 相编码器的 4 倍增脉冲的脉宽。

将通道 1 的 TGRA_1 和 TGRB_1 设定为输入捕捉功能，选择通道 0 的 TGRA_0 和 TGRC_0 的比较匹配作为输入捕捉源，并保存各控制周期时的递增 / 递减计数器的值。

用此方法能检测正确的位置和速度。

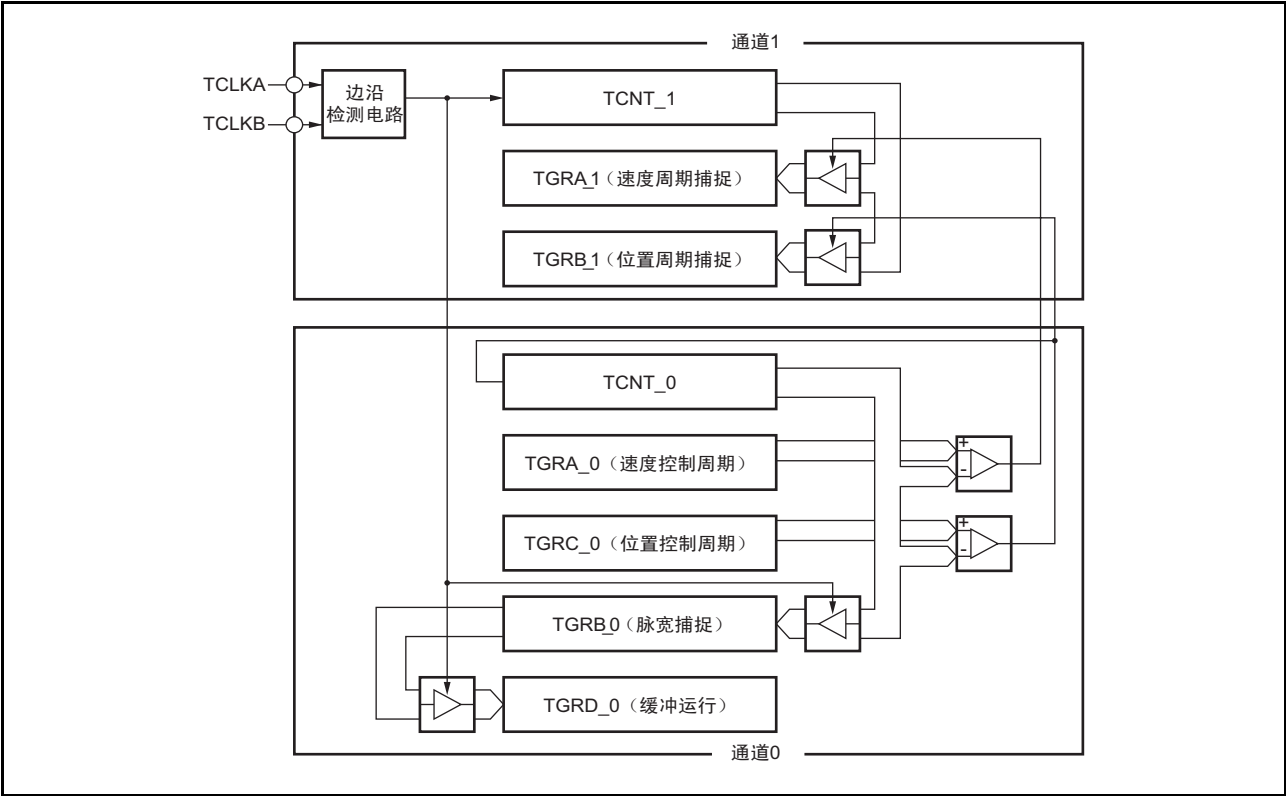


图 10.34 相位计数模式的应用例子

10.4.7 复位同步 PWM 模式

在复位同步 PWM 模式中，通过通道 3、4 的组合将一方的波形变化点为共同关系的 PWM 波形（正相和反相）进行 3 相输出。

当设定为复位同步 PWM 模式时，TIOC3B、TIOC3D、TIOC4A、TIOC4C、TIOC4B 和 TIOC4D 引脚为 PWM 输出引脚，定时器的计数器 3（TCNT_3）用作递增计数器。

使用的 PWM 输出引脚和寄存器设定分别如表 10.52 和表 10.53 所示。

表 10.52 复位同步 PWM 模式的输出引脚

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 10.53 复位同步 PWM 模式的寄存器设定

寄存器	设定内容
TCNT_3	初始设定 H'0000。
TCNT_4	初始设定 H'0000。
TGRA_3	设定 TCNT_3 的计数周期。
TGRB_3	设定从 TIOC3B、TIOC3D 引脚输出的 PWM 波形的变化点。
TGRA_4	设定从 TIOC4A、TIOC4C 引脚输出的 PWM 波形的变化点。
TGRB_4	设定从 TIOC4B、TIOC4D 引脚输出的 PWM 波形的变化点。

(1) 复位同步 PWM 模式的设定步骤例子

复位同步 PWM 模式的设定步骤例子如图 10.35 所示。

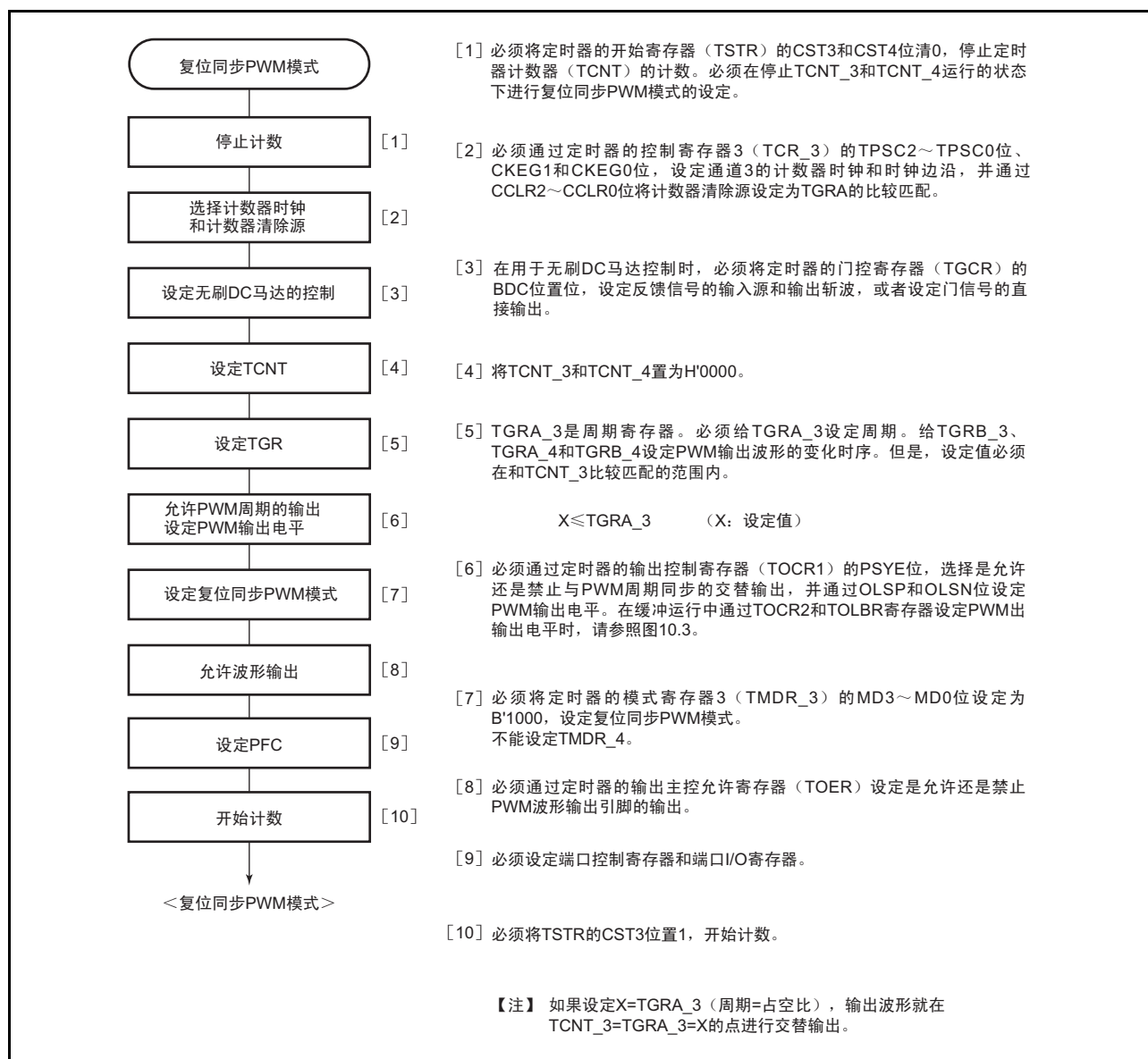


图 10.35 复位同步 PWM 模式的设定步骤例子

(2) 复位同步 PWM 模式的运行例子

复位同步 PWM 模式的运行例子如图 10.36 所示。

在复位同步 PWM 模式中，TCNT_3 和 TCNT_4 作为递增计数器运行。如果 TCNT_3 和 TGRA_3 比较匹配，就清除计数器，并从 H'0000 重新开始递增计数。每当发生各 TGRB_3、TGRA_4、TGRB_4 的比较匹配和计数器清除时，PWM 输出引脚就进行交替输出。

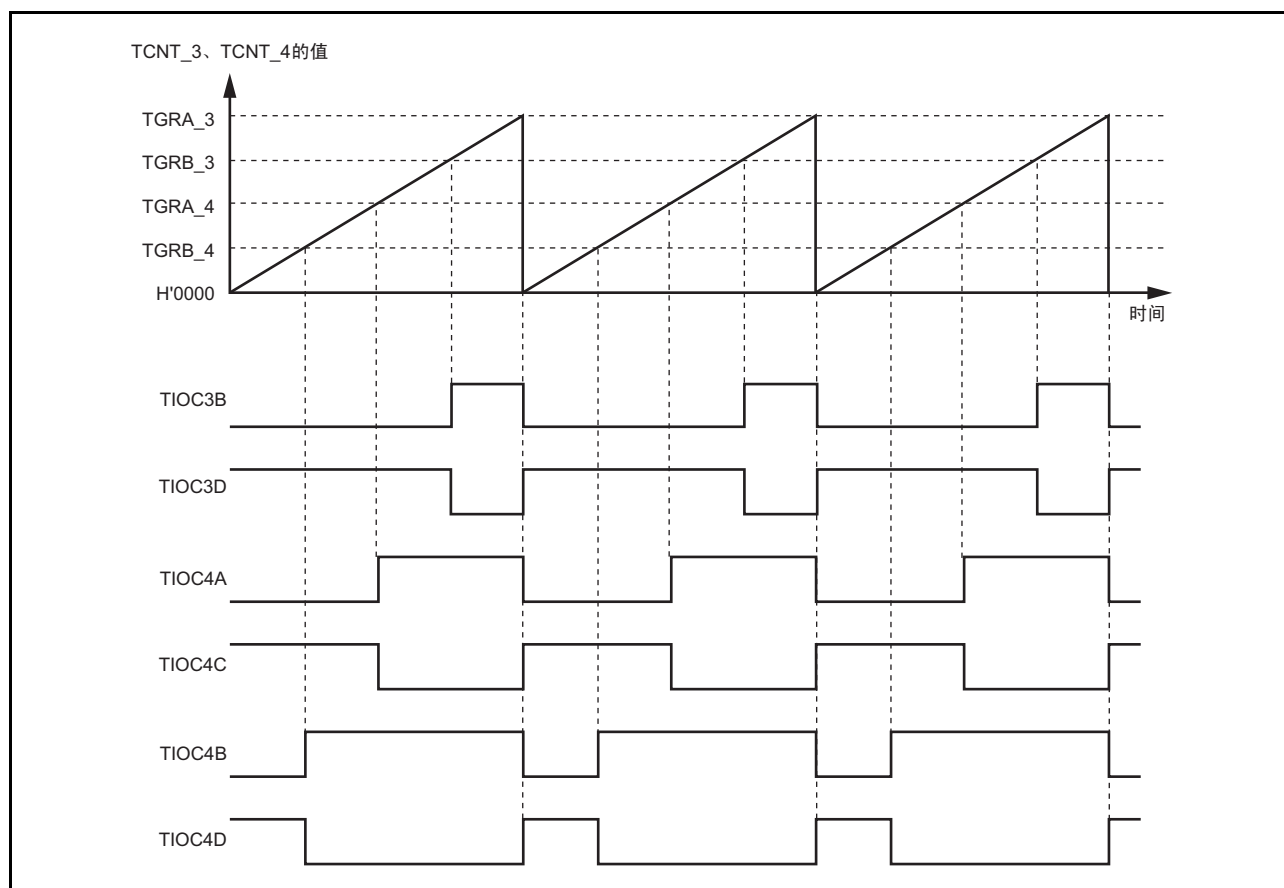


图 10.36 复位同步 PWM 模式的运行例子（当设定 TOCR 的 OLSN=1、OLSP=1 时）

10.4.8 互补 PWM 模式

在互补 PWM 模式中，通过通道 3、4 的组合将正反相为非重叠关系的 PWM 波形进行 3 相输出，也能设定无非重叠时间。

当设定为互补 PWM 模式时，TIO3B、TIO3D、TIO4A、TIO4B、TIO4C、TIO4D 引脚为 PWM 输出引脚，能将 TIO3A 引脚设定为与 PWM 周期同步的交替输出。

另外，TCNT_3 和 TCNT_4 用作递增 / 递减计数器。

使用的 PWM 输出引脚和寄存器设定分别如表 10.54 和表 10.55 所示。

作为端口功能，支持通过外部信号直接截止 PWM 输出的功能。

表 10.54 互补 PWM 模式的输出引脚

通道	输出引脚	说明
3	TIOC3A	与 PWM 周期同步的交替输出 (或者输入 / 输出端口)
	TIOC3B	PWM 输出引脚 1
	TIOC3C	输入 / 输出端口 *
	TIOC3D	PWM 输出引脚 1' (PWM 输出 1 的非重叠关系的反相波形。也能设定无非重叠时间)
4	TIOC4A	PWM 输出引脚 2
	TIOC4C	PWM 输出引脚 2' (PWM 输出 2 的非重叠关系的反相波形。也能设定无非重叠时间)
	TIOC4B	PWM 输出引脚 3
	TIOC4D	PWM 输出引脚 3' (PWM 输出 3 的非重叠关系的反相波形。也能设定无非重叠时间)

【注】 * 在互补 PWM 模式中, 不能将 TIOC3C 引脚设定为定时器的输入 / 输出引脚。

表 10.55 互补 PWM 模式的寄存器设定

通道	计数器 / 寄存器	说明	从 CPU 进行的读写
3	TCNT_3	从空载时间寄存器的设定值开始递增计数。	能通过设定 *TRWER 进行屏蔽。
	TGRA_3	设定 TCNT_3 的上限值 (1/2 的载波周期 + 空载时间)。	能通过设定 *TRWER 进行屏蔽。
	TGRB_3	PWM 输出 1 的比较寄存器	能通过设定 *TRWER 进行屏蔽。
	TGRC_3	TGRA_3 的缓冲寄存器	总是能读写。
	TGRD_3	PWM 输出 1/TGRB_3 的缓冲寄存器	总是能读写。
4	TCNT_4	初始设定 H'0000 并开始递增计数。	能通过设定 *TRWER 进行屏蔽。
	TGRA_4	PWM 输出 2 的比较寄存器	能通过设定 *TRWER 进行屏蔽。
	TGRB_4	PWM 输出 3 的比较寄存器	能通过设定 *TRWER 进行屏蔽。
	TGRC_4	PWM 输出 2/TGRA_4 的缓冲寄存器	总是能读写。
	TGRD_4	PWM 输出 3/TGRB_4 的缓冲寄存器	总是能读写。
定时器的空载时间数据寄存器 (TDDR)		设定 TCNT_4 和 TCNT_3 的偏移值 (空载时间的值)。	能通过设定 *TRWER 进行屏蔽。
定时器的周期数据寄存器 (TCDR)		设定 TCNT_4 的上限值 (1/2 的载波周期)。	能通过设定 *TRWER 进行屏蔽。
定时器周期缓冲寄存器 (TCBR)		TCDR 的缓冲寄存器	总是能读写。
子计数器 (TCNTS)		生成空载时间的子计数器	只能读。
暂存器 1 (TEMP1)		PWM 输出 1/TGRB_3 的暂存器	不能读写。
暂存器 2 (TEMP2)		PWM 输出 2/TGRA_4 的暂存器	不能读写。
暂存器 3 (TEMP3)		PWM 输出 3/TGRB_4 的暂存器	不能读写。

【注】 * 能通过设定 TRWER (定时器的读写允许寄存器), 允许或者禁止存取。

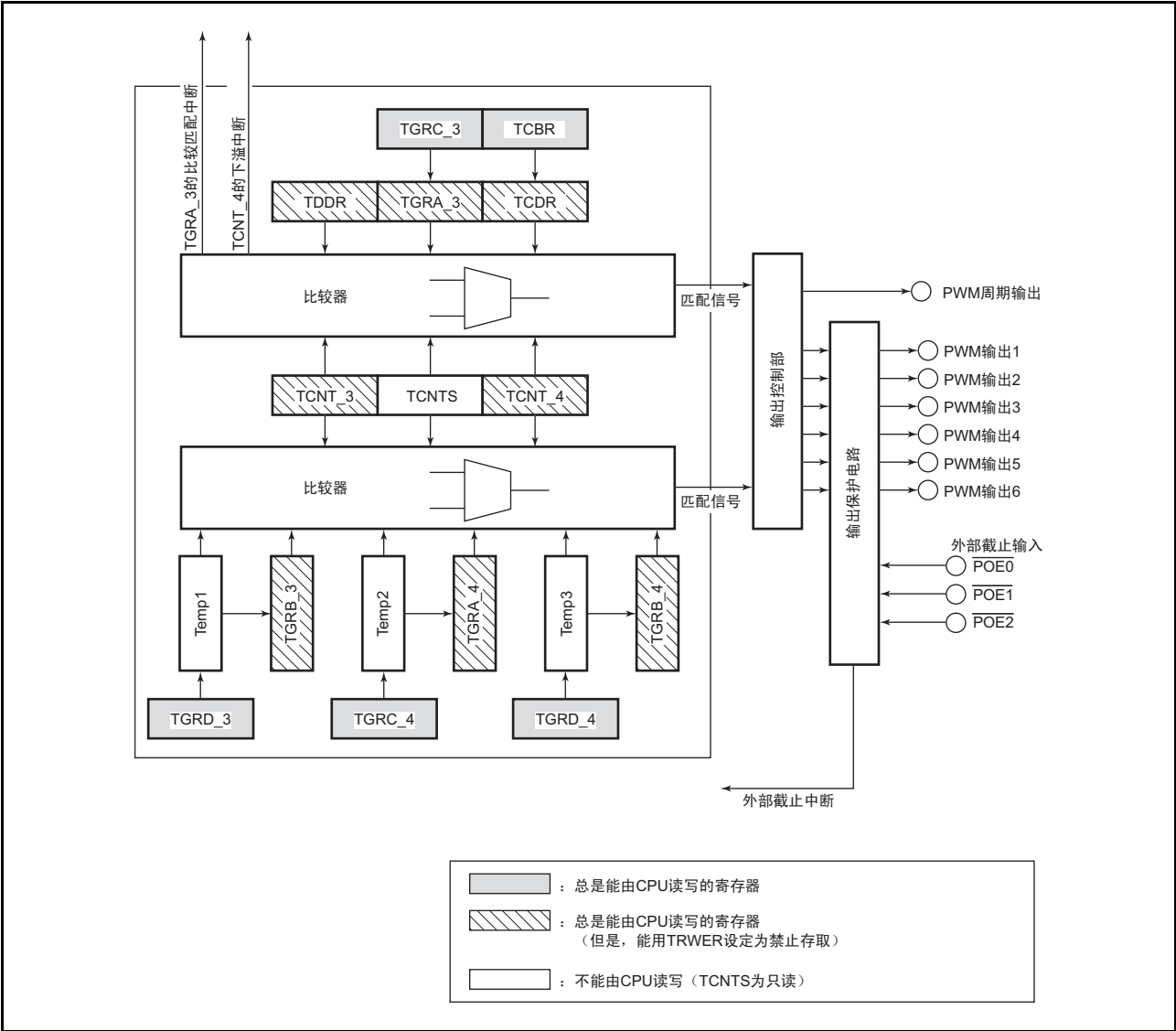


图 10.37 互补 PWM 模式的通道 3、4 框图

(1) 互补 PWM 模式的设定步骤例子

互补 PWM 模式的设定步骤例子如图 10.38 所示。



图 10.38 互补 PWM 模式的设定步骤例子

(2) 互补 PWM 模式的运行概要

在互补 PWM 模式中，能进行 6 相的 PWM 输出。互补 PWM 模式的计数器运行和运行例子分别如图 10.39 和图 10.40 所示。

(a) 计数器运行

在互补 PWM 模式中，TCNT_3、TCNT_4 和 TCNTS（3 个计数器）进行递增计数。

当设定为互补 PWM 模式并且 TSTR 的 CST 位为 0 时，TCNT_3 的初始值就自动设定为 TDDR 的设定值。

当 CST 位被置 1 时，TCNT_3 就进行递增计数（计数到 TGRA_3 的设定值为止）。当 TCNT_3 和 TGRA_3 相同时，TCNT_3 就转换为递减计数；当 TCNT_3 和 TDDR 相同时，TCNTS 就又转换为递增计数，重复此运行。

另外，将 TCNT_4 的初始值设定为 H'0000。

当 CST 位置被 1 时，TCNT_4 就与 TCNT_3 同步进行递增计数。当 TCNT_4 和 TCDR 相同时，TCNT_4 就转换为递减计数；当 TCNT_4 计数到 H'0000 时，TCNT_4 就又转换为递增计数，重复此运行。

TCNTS 是只读计数器，不需要设定初始值。

在 TCNT_3、4 进行递增计数时，如果 TCNT_3 和 TCDR 相同，TCNTS 就开始递减计数。当 TCNTS 和 TCDR 相同时，TCNTS 就转换为递增计数；当 TCNTS 和 TGRA_3 相同时，TCNTS 就被清 0。

在 TCNT_3、TCNT_4 进行递减计数时，如果 TCNT_4 和 TDDR 相同，TCNTS 就开始递增计数。当 TCNTS 和 TDDR 相同时，TCNTS 就转换为递减计数；当 TCNTS 计数到 H'0000 时，TCNTS 就被设定为 TGRA_3 的值。

TCNTS 只在计数器运行期间与设定 PWM 占空比的比较寄存器和暂寄存器比较。

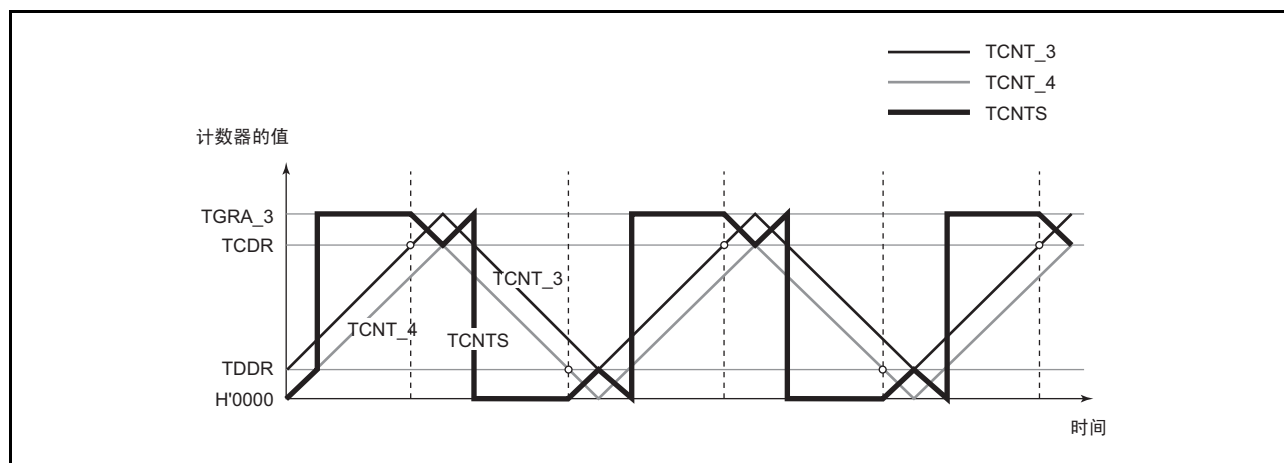


图 10.39 互补 PWM 模式的计数器运行

(b) 寄存器运行

在互补 PWM 模式中，使用比较寄存器、缓冲寄存器和暂寄存器 9 个寄存器。互补 PWM 模式的运行例子如图 10.40 所示。

为了进行 PWM 输出，寄存器 TGRB_3、TGRA_4 和 TGRB_4 总是和计数器比较。当这些寄存器和计数器相同时，定时器的输出控制寄存器（TOCR）的 OLSN 位和 OLSP 位的设定值就被输出。

这些比较寄存器的缓冲寄存器是 TGRD_3、TGRD_4 和 TGRD_4。

另外，在缓冲寄存器和比较寄存器之间有暂寄存器，但不能从 CPU 存取暂寄存器。

要更改比较寄存器的数据时，必须给对应的缓冲寄存器写要更改的数据。缓冲寄存器总是能读写。

在 Ta 区间总是将写到缓冲寄存器的数据传送到暂寄存器，而在 Tb 区间不传送到暂寄存器。在 Tb 区间结束后，将在此区间已写到缓冲寄存器的数据传送到暂寄存器。

如果 T_b 区间结束的 TCNTS 在递增计数时和 TGRA_3 相同或者在递减计数时计数到 H'0000，就将已传送到暂寄存器的值传送到比较寄存器。能通过定时器的模式寄存器 (TMDR) 的 MD3 ~ MD0 位选择此暂寄存器到比较寄存器的传送时序。图 10.40 是选择在波谷更改的模式例子。

在不向暂寄存器传送数据的 T_b (图 10.40 的 T_{b1}) 区间，暂寄存器具有和比较寄存器相同的功能，和计数器比较。在此区间，1 相的输出有 2 个比较匹配寄存器，比较寄存器存有更改前的数据，暂寄存器存有更改后的新数据。而且在此区间，TCNT_3、4 和 TCNTS (3 个计数器) 与比较寄存器和暂寄存器 (2 个寄存器) 比较，控制 PWM 输出。

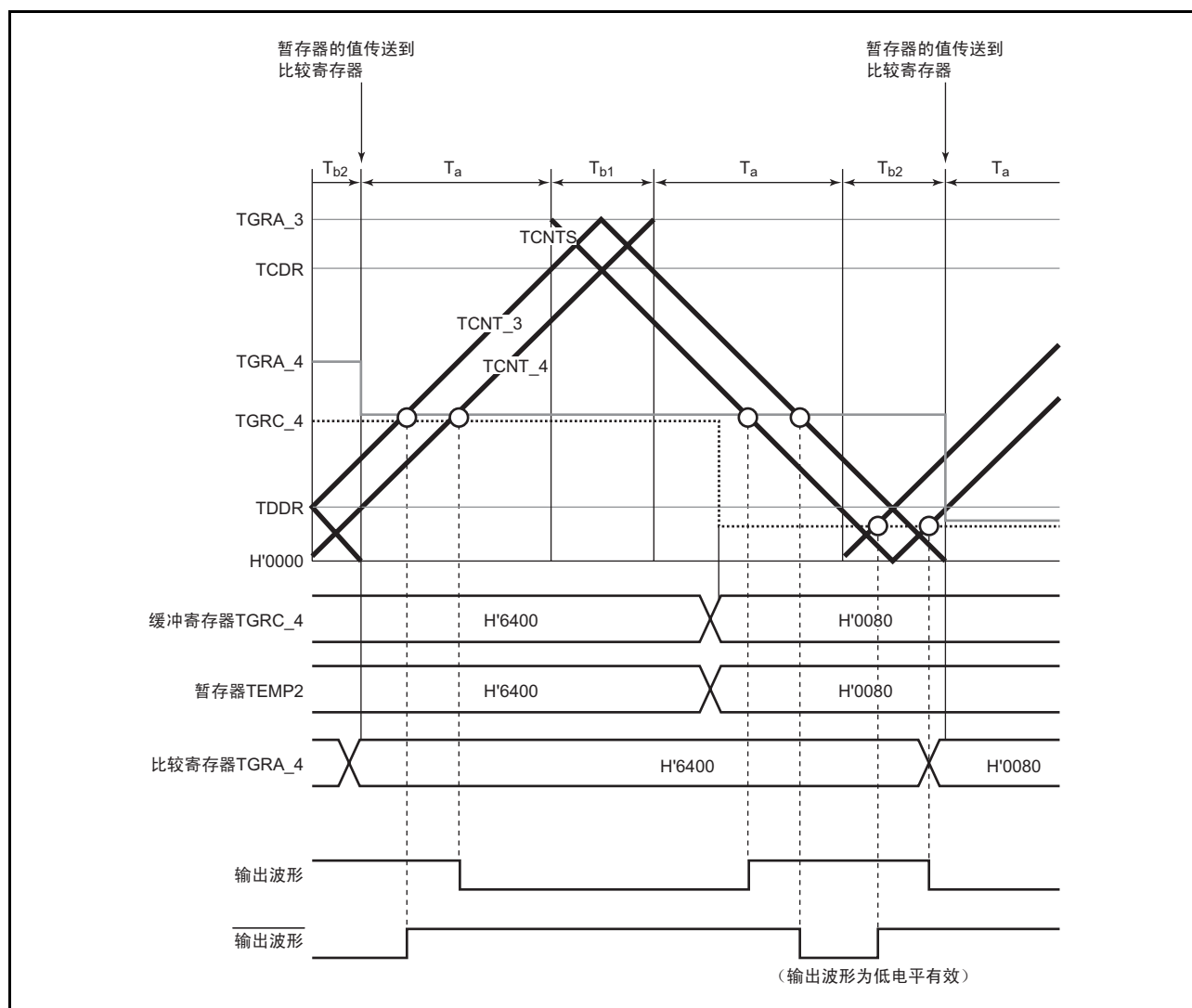


图 10.40 互补 PWM 模式的运行例子

(c) 初始设定

在互补 PWM 模式中，有 6 个需要初始设定的寄存器，还有 1 个用于设定是否生成空载时间的寄存器（只能在不生成空载时间的情况下进行设定）。

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

TGRC_3 用作 TGRA_3 的缓冲寄存器，设定 1/2 的 PWM 载波周期 + 空载时间 T_d ；定时器的周期缓冲寄存器 (TCBR) 用作定时器的周期数据寄存器 (TCDR) 的缓冲寄存器，设定 1/2 的 PWM 载波周期，并给定时器

的空载时间寄存器 (TDDR) 设定空载时间 T_d 。

如果不生成空载时间, 就将定时器的空载时间允许寄存器 (TDER) 的 TDER 位置 0, 给 TGRC_3 和 TGRA_3 设定 1/2 的 PWM 载波周期 +1, 并将 TDDR 置 1。

给缓冲寄存器 TGRD_3、TGRC_4 和 TGRD_4 (3 个寄存器) 分别设定 PWM 占空比的初始值。

在设定为互补 PWM 模式的同时, 将 5 个缓冲寄存器 (TDDR 除外) 的设定值分别传送到对应的比较寄存器。

另外, 必须在设定为互补 PWM 模式前将 TCNT_4 置 0。

表 10.56 需要初始设定的寄存器和计数器

寄存器 / 计数器	设定值
TGRC_3	1/2 的 PWM 载波周期 + 空载时间 T_d (如果通过 TDER 设定为不生成空载时间, 就为 1/2 的 PWM 载波周期 +1)
TDDR	空载时间 T_d (如果通过 TDER 设定为不生成空载时间, 就为 1)
TCBR	1/2 的 PWM 载波周期
TGRD_3、TGRC_4、TGRD_4	各相 PWM 占空比的初始值
TCNT_4	H'0000

【注】 TGRC_3 的设定值必须为 TCBR 设定的 PWM 载波周期的 1/2 值 +TDDR 设定的空载时间 T_d 的值。如果通过 TDER 设定为不生成空载时间, 就为 1/2 的 PWM 载波周期 +1。

(d) PWM 输出电平の設定

在互补 PWM 模式中, 通过定时器的输出控制寄存器 1 (TOCR1) 的 OLSN 位和 OLSP 位或者定时器的输出控制寄存器 2 (TOCR2) 的 OLS1P ~ OLS3P 位和 OLS1N ~ OLS3N 位, 设定 PWM 脉冲的输出电平。

能按 3 相正相和 3 相反相 (6 相输出) 设定输出电平。

另外, 必须在解除互补 PWM 模式的状态下设定或者更改输出电平。

(e) 空载时间的设定

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

将非重叠时间设定到定时器的空载时间数据寄存器 (TDDR)。TDDR 的设定值作为 TCNT_3 计数器的初始值, 生成 TCNT_3 和 TCNT_4 的非重叠关系。必须在解除互补 PWM 模式的状态下更改 TDDR 的内容。

(f) 不生成空载时间的设定

通过将定时器的空载时间允许寄存器 (TDER) 的 TDER 位置 0, 设定为不生成空载时间。只能在 TDER=1 的状态下读 TDER 后给 TDER 写 0 时, 将 TDER 置 0。

给 TGRA_3 和 TGRC_3 设定 1/2 的 PWM 载波周期 +1, 并将定时器的空载时间数据寄存器 (TDDR) 置 1。

如果设定为不生成空载时间, 就能输出无空载时间的 PWM 波形。不生成空载时间的运行例子如图 10.41 所示。

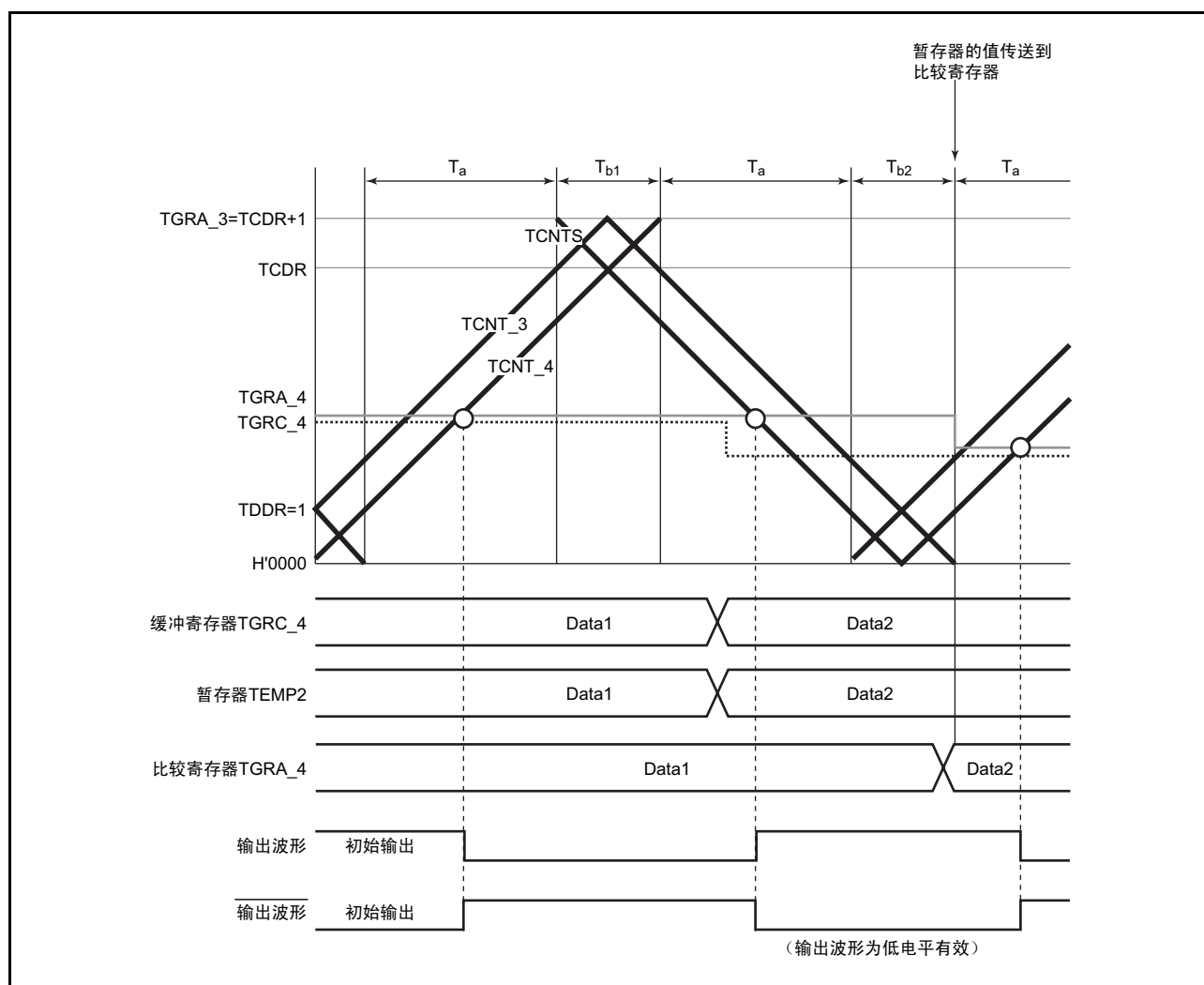


图 10.41 不生成空载时间的运行例子

(g) PWM 周期的设定

在互补 PWM 模式中，将 PWM 脉冲的周期设定到 TGRA_3（设定 TCNT3 的上限值）和 TCDR（设定 TCNT_4 的上限值）这 2 个寄存器。必须将此 2 个寄存器设定为以下的关系：

生成空载时间：TGRA_3 的设定值 = TCDR 的设定值 + TDDR 的设定值

不生成空载时间：TGRA_3 的设定值 = TCDR 的设定值 + 1

另外，必须通过给缓冲寄存器的 TGRC_3 和 TCBR 设定值进行 TGRA_3 和 TCDR 的设定。在定时器的模式寄存器（TMDR）的 MD3 ~ MD0 选择的传送时序中，将 TGRC_3 和 TCBR 的设定值同时传送到 TGRA_3 和 TCDR。

如果在波峰更新数据，就从下一个周期反映更改的 PWM 周期；如果在波谷更新数据，就从该周期反映更改的 PWM 周期。在波峰更改 PWM 周期时的运行例子如图 10.42 所示。

另外，有关各缓冲寄存器数据的更新方法，请参照以下的“(h) 寄存器数据的更新”。

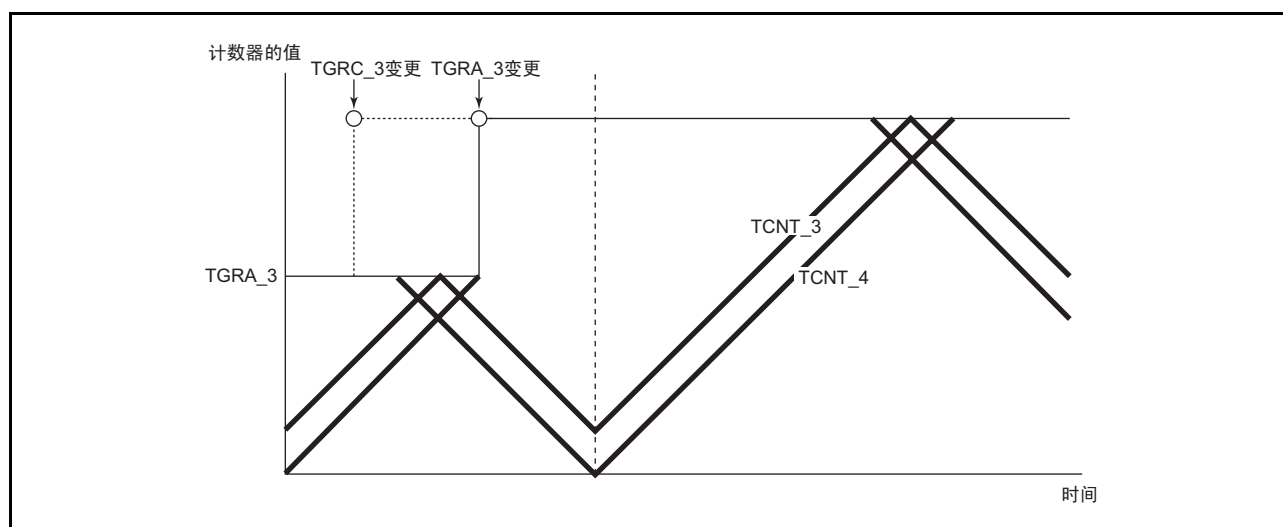


图 10.42 PWM 周期的更改例子

(h) 寄存器数据的更新

在互补 PWM 模式中更新比较寄存器的数据时，使用缓冲寄存器。更新的数据随时能写到缓冲寄存器。另外，能在缓冲寄存器运行中更改的寄存器有 5 个 PWM 占空比和载波周期的寄存器。

在这些寄存器和缓冲寄存器之间有各自的暂存器。在子计数器 TCNTS 不计数的期间，如果更新寄存器的数据，暂存器的值就会被改写。在 TCNTS 计数时，不将缓冲寄存器的值传送到暂存器，而在 TCNTS 停止计数后将写到缓冲寄存器的值传送到暂存器。

在定时器的模式寄存器 (TMDR) 的 MD3 ~ MD0 位设定的数据更新时序中，将暂存器的值传送到比较寄存器。互补 PWM 模式的数据更新例子如图 10.43 所示，此图是在计数器的波峰和波谷更新数据的模式。

在改写缓冲寄存器的数据时，最后必须写 TGRD_4。在写 TGRD_4 后，5 个寄存器同时将缓冲寄存器的数据传送到暂存器。

如果不全部更新 5 个寄存器或者不更新 TGRD_4 的数据，就必须在写要更新的寄存器数据后写 TGRD_4。此时，写到 TGRD_4 的数据必须和写之前的数据相同。

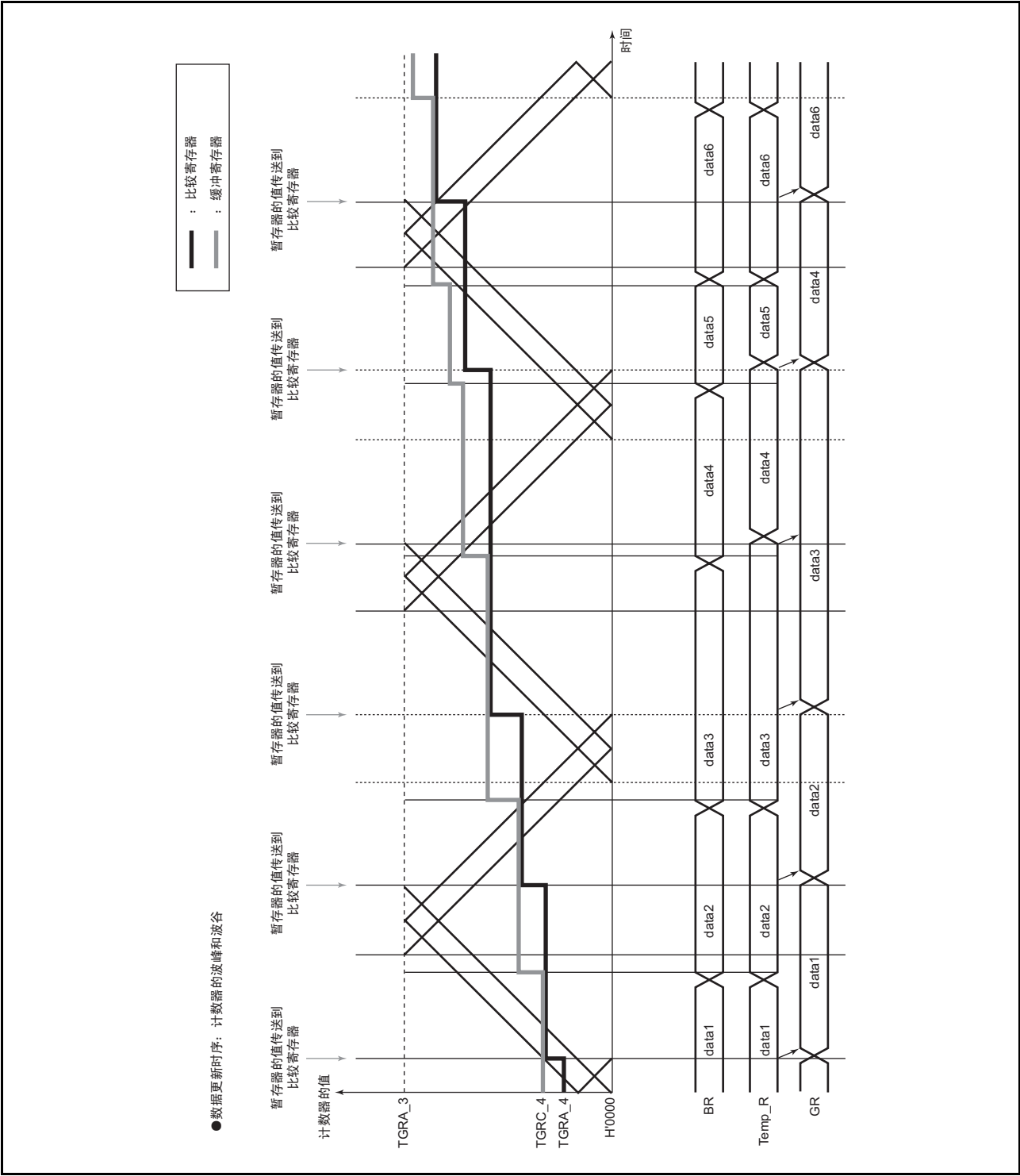


图 10.43 互补 PWM 模式的数据更新例子

(i) 互补 PWM 模式的初始输出

在互补 PWM 模式中，通过设定定时器的输出控制寄存器 1 (TOCR1) 的 OLSN 位和 OLSP 位或者定时器的输出控制寄存器 2 (TOCR2) 的 OLS1N ~ OLS3N 位和 OLS1P ~ OLS3P 位，决定初始输出。

此初始输出为 PWM 脉冲的无效电平，从通过定时器的模式寄存器 (TMDR) 设定互补 PWM 模式到 TCNT_4 大于空载时间寄存器 (TDDR) 的设定值为止，输出此初始输出。互补 PWM 模式的初始输出例子如图 10.44 所示。

另外，PWM 占空比的初始值小于 TDDR 值的波形例子如图 10.45 所示。

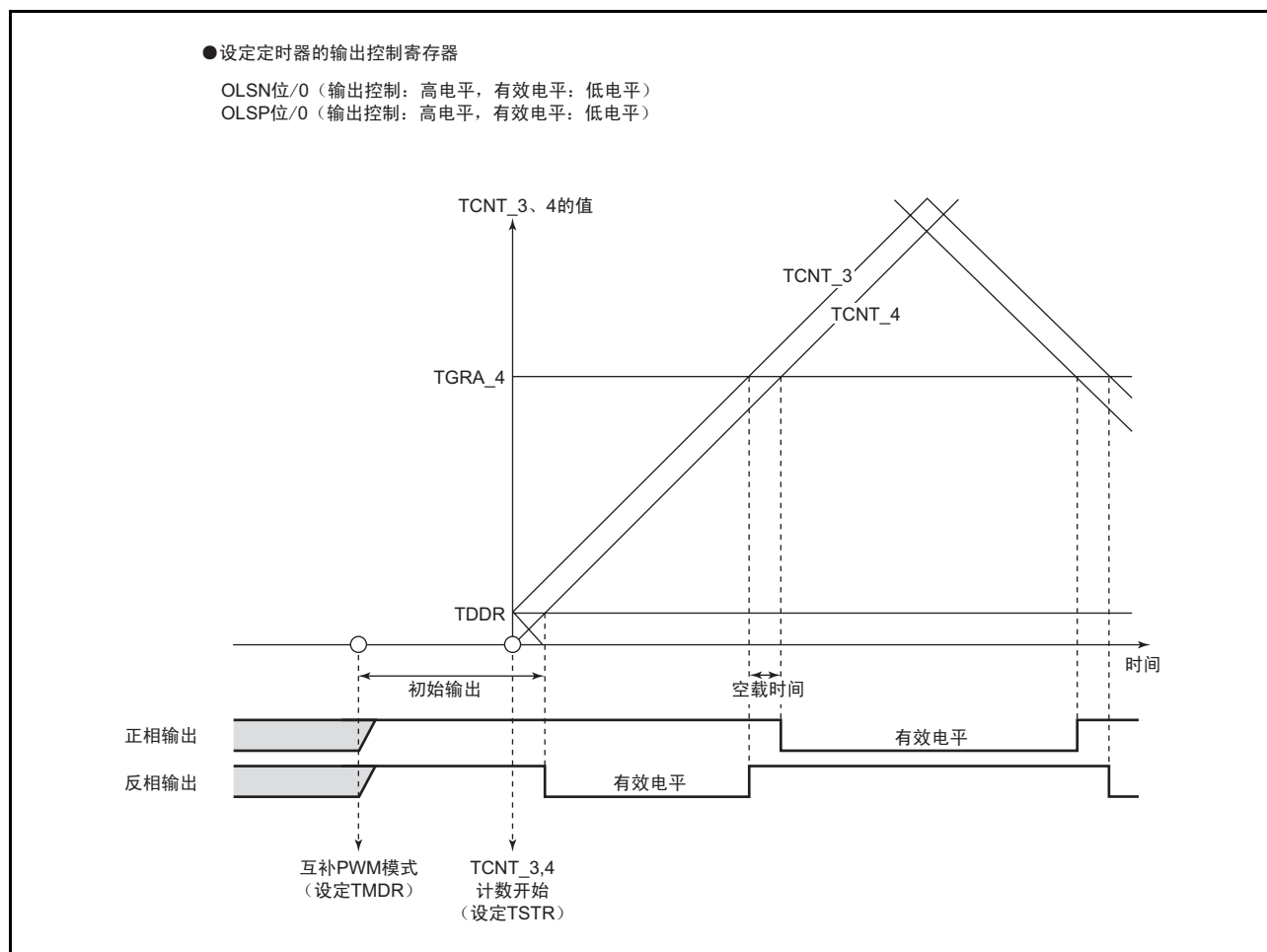


图 10.44 互补 PWM 模式的初始输出例子 (1)

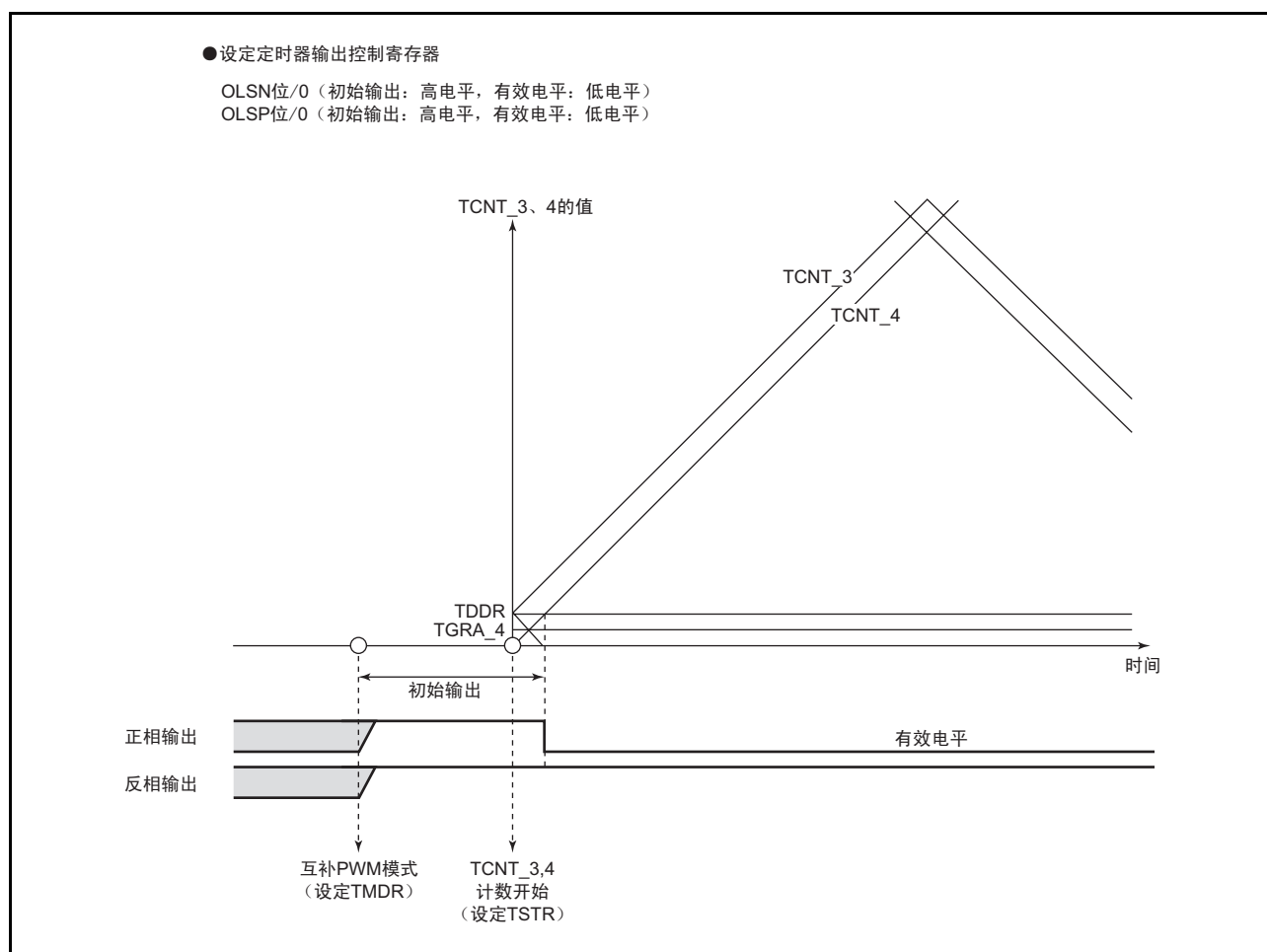


图 10.45 互补 PWM 模式的初始输出例子 (2)

(j) 互补 PWM 模式的 PWM 输出生成方法

在互补 PWM 模式中, 将正反相有非重叠时间的 PWM 波形进行 3 相输出。此非重叠时间称为空载时间。

在发生计数器和数据寄存器的比较匹配时, 通过输出定时器的输出控制寄存器选择的输出电平生成 PWM 波形。另外, 在 TCNTS 计数期间, 因为产生 0 ~ 100% 连续的 PWM 脉冲, 所以同时比较数据寄存器和暂存器的值。此时, 与 ON、OFF 比较匹配产生的时序会变, 为了确保空载时间并且不使正相 / 反相的 ON 时间重叠, 必须优先 OFF 各相的比较匹配。互补 PWM 模式的波形生成例子如图 10.46 ~ 图 10.48 所示。

通过和实线计数器的比较匹配生成正相 / 反相的 OFF 时序, 通过和虚线计数器 (比实线计数器迟空载时间后的计数器运行) 的比较匹配生成 ON 时序。在 T1 期间, 最优先使反相 OFF 的 a 的比较匹配, 忽视比 a 先产生的比较匹配。另外, 在 T2 期间, 最优先使正相 OFF 的 c 的比较匹配, 忽视比 c 先产生的比较匹配。

如图 10.46 所示, 通常按照 a→b→c→d (或者 c→d→a'→b') 的顺序产生比较匹配。

当比较匹配偏离 a→b→c→d 的顺序时, 因为使反相 OFF 的时间短于空载时间的 2 倍, 所以表示正相不为 ON; 当比较匹配偏离 c→d→a'→b' 的顺序时, 因为使正相 OFF 的时间短于空载时间的 2 倍, 所以表示反相不为 ON。

如图 10.47 所示, 如果在 a 的比较匹配之后先产生 c 的比较匹配, 就忽视 b 的比较匹配, 而通过 d 的比较匹配使反相 OFF。这是为了通过比 b 的比较匹配 (正相 ON 时序) 先产生正相 OFF 的 c 的比较匹配, 优先使正相 OFF (因为正相是从 OFF 到 OFF, 所以波形不变)。

同样地, 在图 10.48 所示的例子中, 比 c 的比较匹配先产生和暂存器的新数据比较匹配的 a', 但是在产生

使正相 OFF 的 c 前忽视其他比较匹配，因此反相不为 ON。

如此，在互补 PWM 模式中，优先 OFF 时序的比较匹配，即使 ON 时序的比较匹配比 OFF 先产生也被忽视。

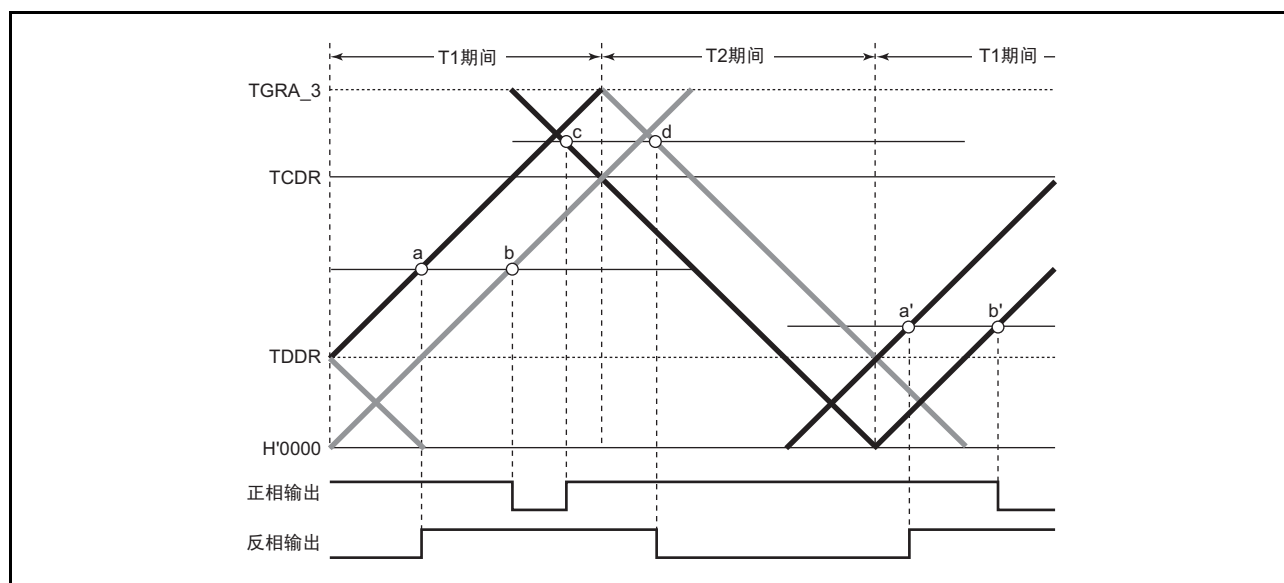


图 10.46 互补 PWM 模式的波形输出例子 (1)

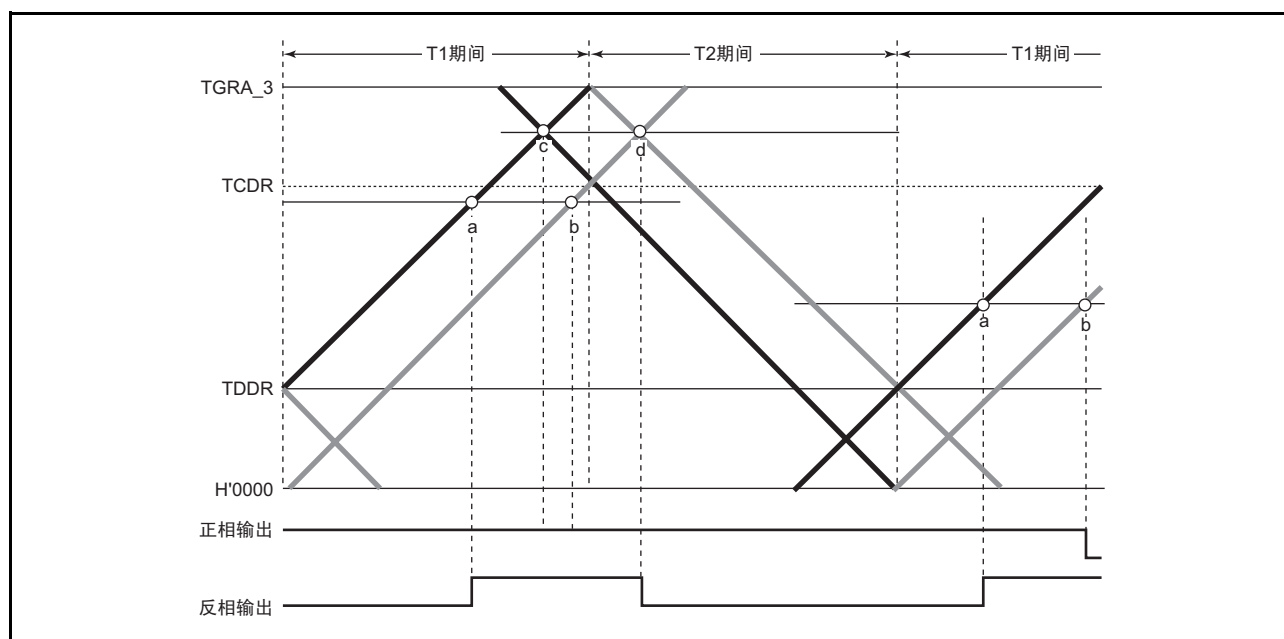


图 10.47 互补 PWM 模式的波形输出例子 (2)

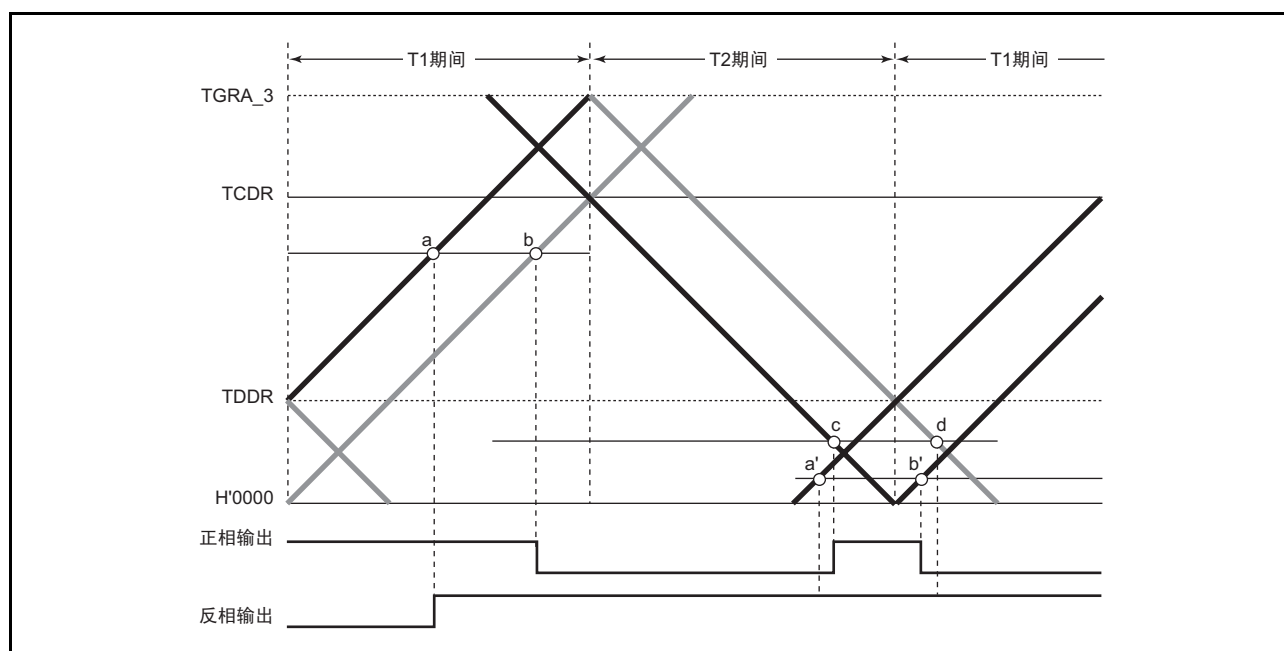


图 10.48 互补 PWM 模式的波形输出例子 (3)

(k) 互补 PWM 模式的 0% 和 100% 占空比输出

在互补 PWM 模式中, 能任意输出 0% 和 100% 的占空比, 输出例子如图 10.49 ~ 图 10.53 所示。

如果将数据寄存器的值置 0, 就输出 100% 的占空比, 此时的波形为正相 100%ON 状态的波形。另外, 如果将数据寄存器的值设定为 TGRA_3 的相同值, 就输出 0% 的占空比, 此时的波形为正相 100%OFF 状态的波形。

此时, 同时产生 ON 和 OFF 的比较匹配, 如果同相 ON 的比较匹配和 OFF 的比较匹配同时产生, 双方的比较匹配就都被忽视而波形不变。

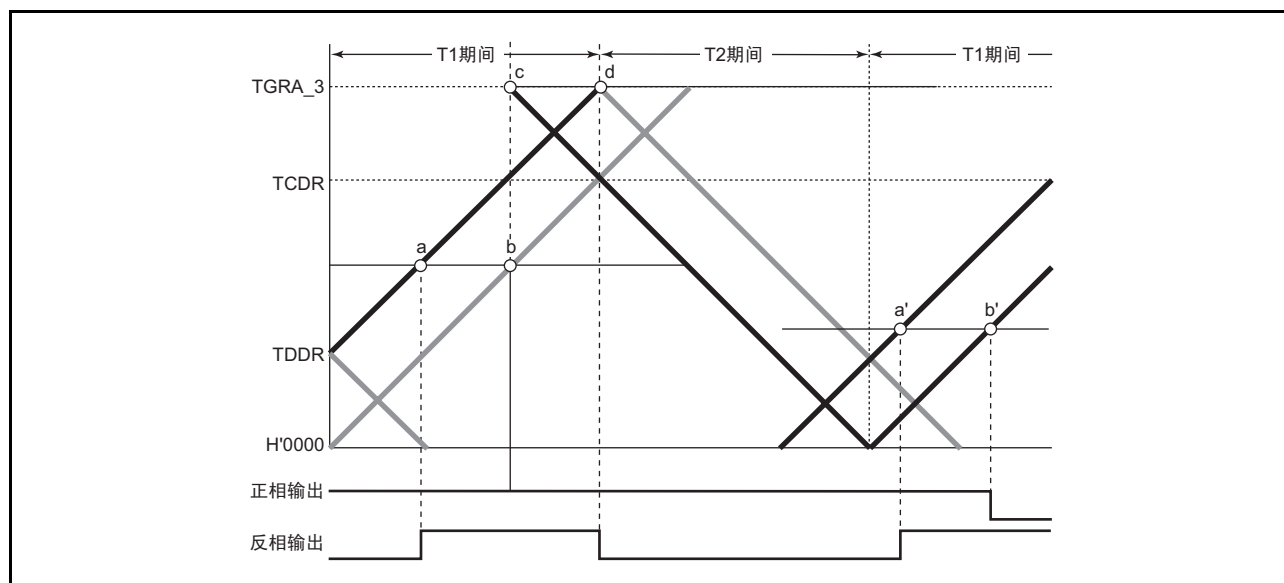


图 10.49 互补 PWM 模式的 0% 和 100% 波形输出例子 (1)

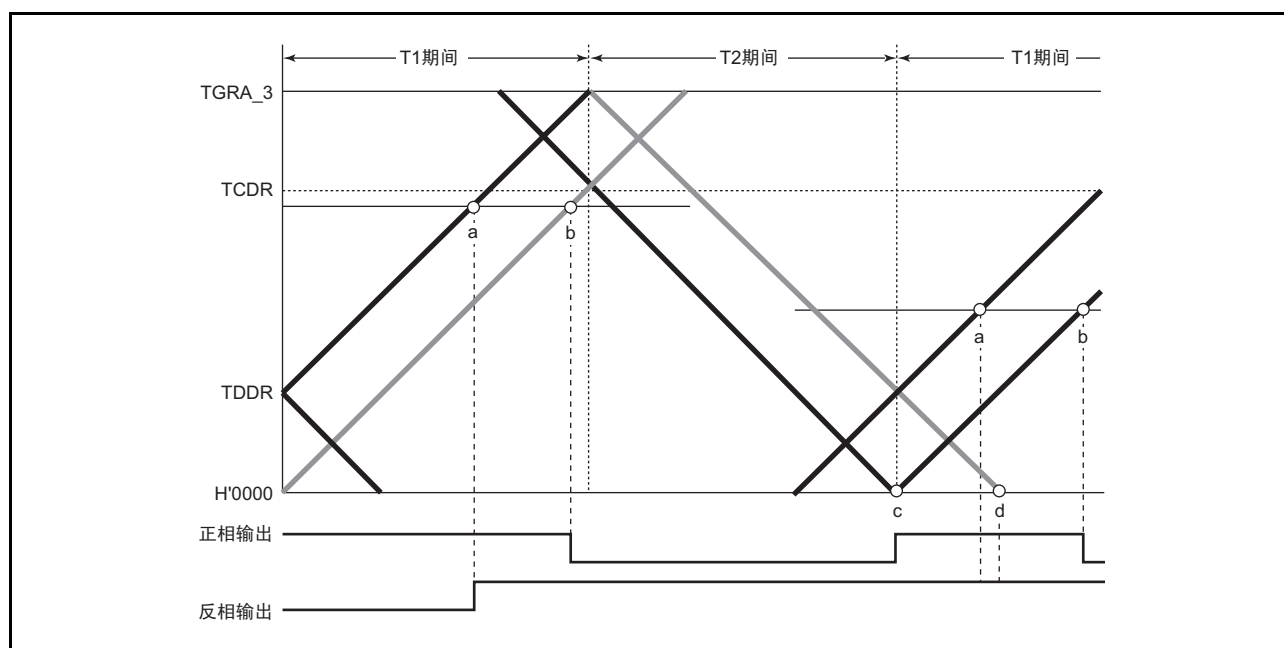


图 10.50 互补 PWM 模式的 0% 和 100% 波形输出例子 (2)

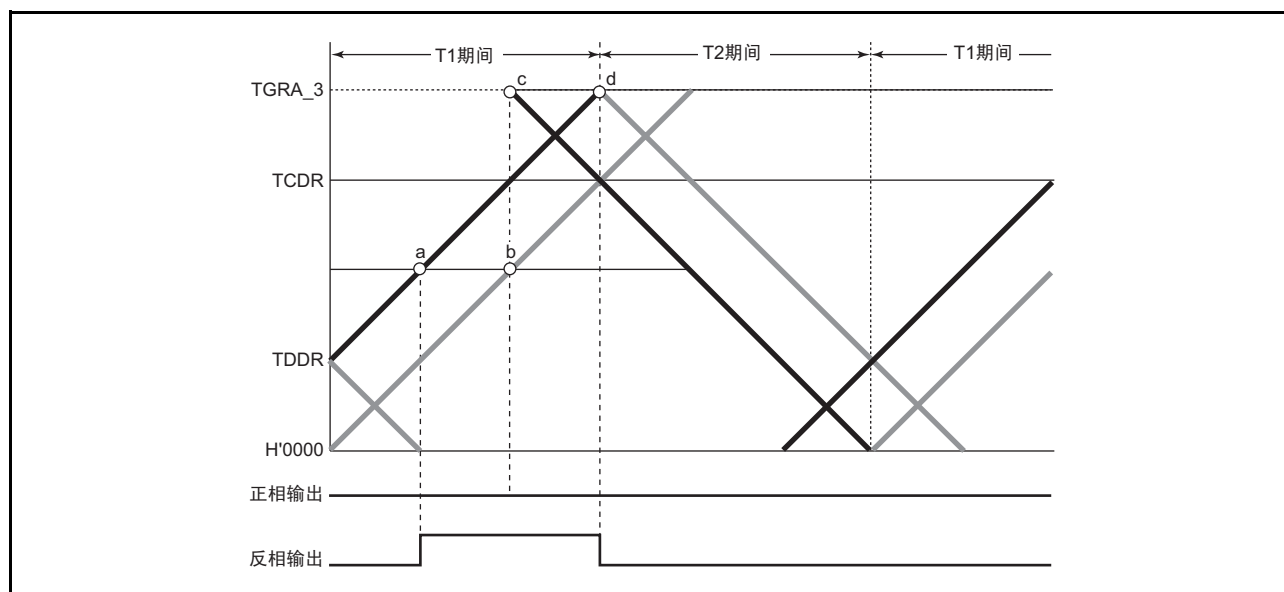


图 10.51 互补 PWM 模式的 0% 和 100% 波形输出例子 (3)

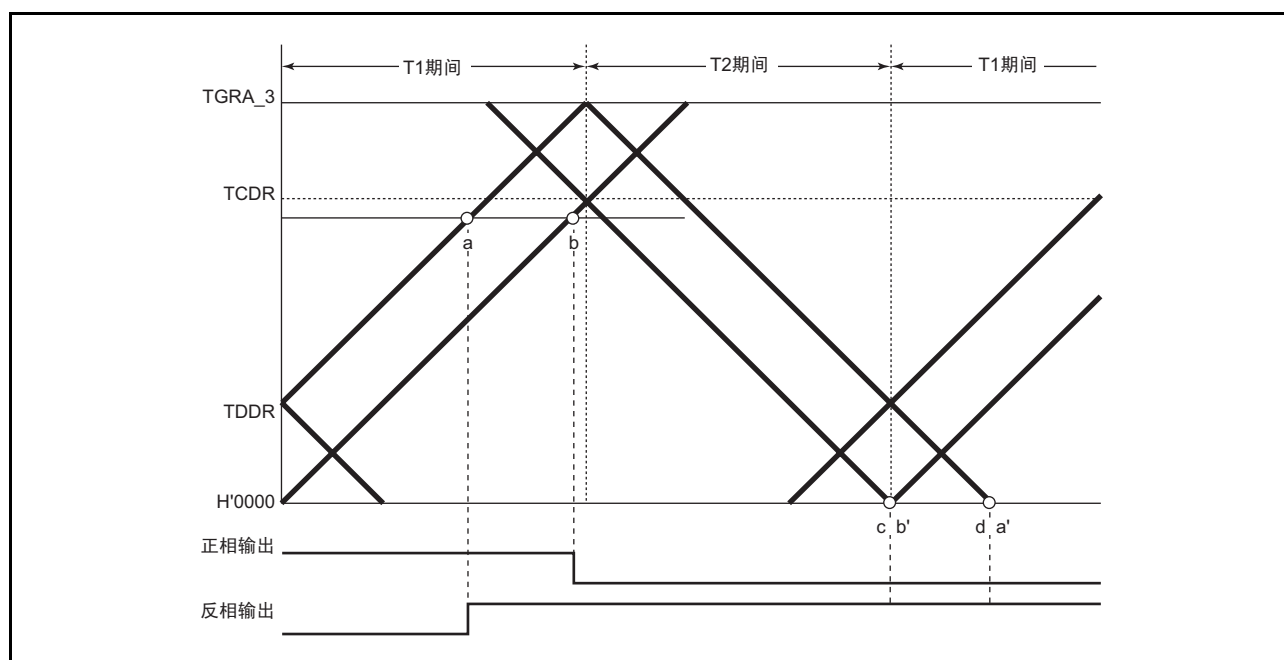


图 10.52 互补 PWM 模式的 0% 和 100% 波形输出例子 (4)

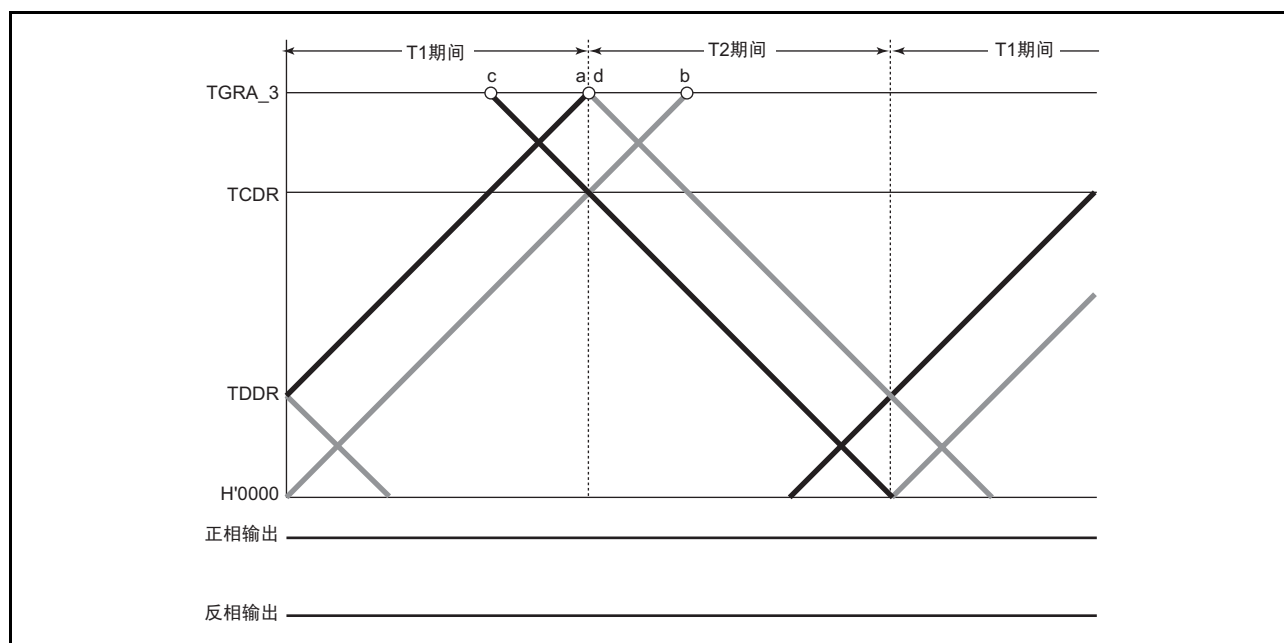


图 10.53 互补 PWM 模式的 0% 和 100% 波形输出例子 (5)

(I) 与 PWM 周期同步的交替输出

在互补 PWM 模式中, 能通过将定时器的输出控制寄存器 (TOCR) 的 PSYE 位置 1 进行与 PWM 载波周期同步的交替输出。交替输出的波形例子如图 10.54 所示。

通过 TCNT_3 和 TGRA_3 的比较匹配以及 TCNT4 和 H'0000 的比较匹配进行交替输出。

此交替输出的输出引脚为 TIOC3A 引脚, 初始输出为 1。

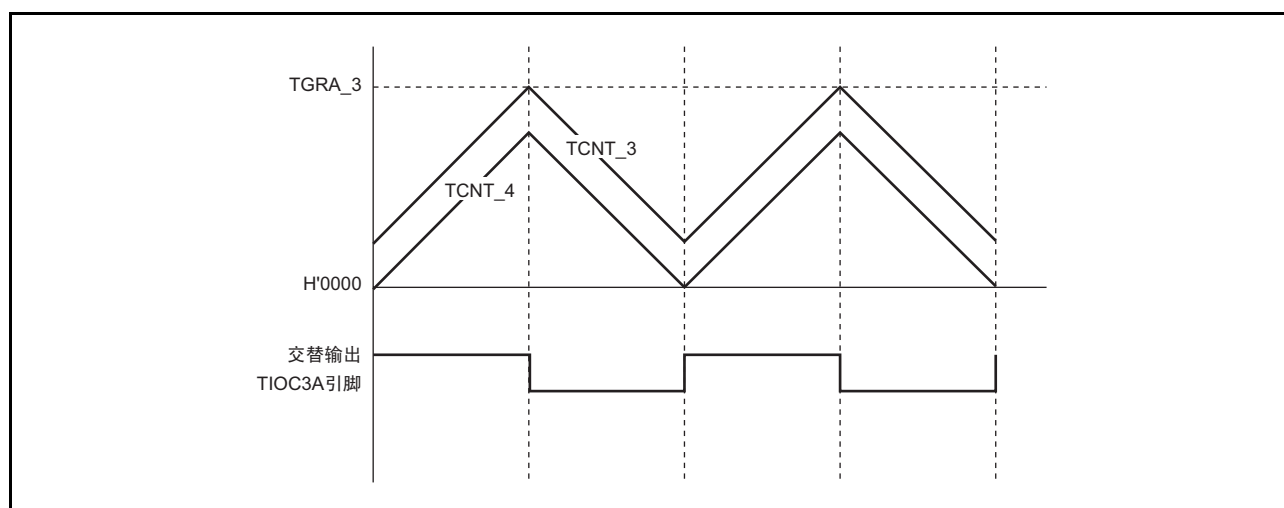


图 10.54 与 PWM 输出同步的交替输出的波形例子

(m) 其他通道的计数器清除

在互补 PWM 模式中，通过定时器的同步寄存器 (TSYR) 设定为与其他通道同步的模式，并通过定时器的控制寄存器 (TCR) 的 CCLR2 ~ CCLR0 选择同步清除，就能通过其他通道进行 TCNT_3、TCNT_4 和 TCNTS 的清除。

运行例子如图 10.55 所示。

使用此功能，能通过外部信号清除计数器和重新开始计数。

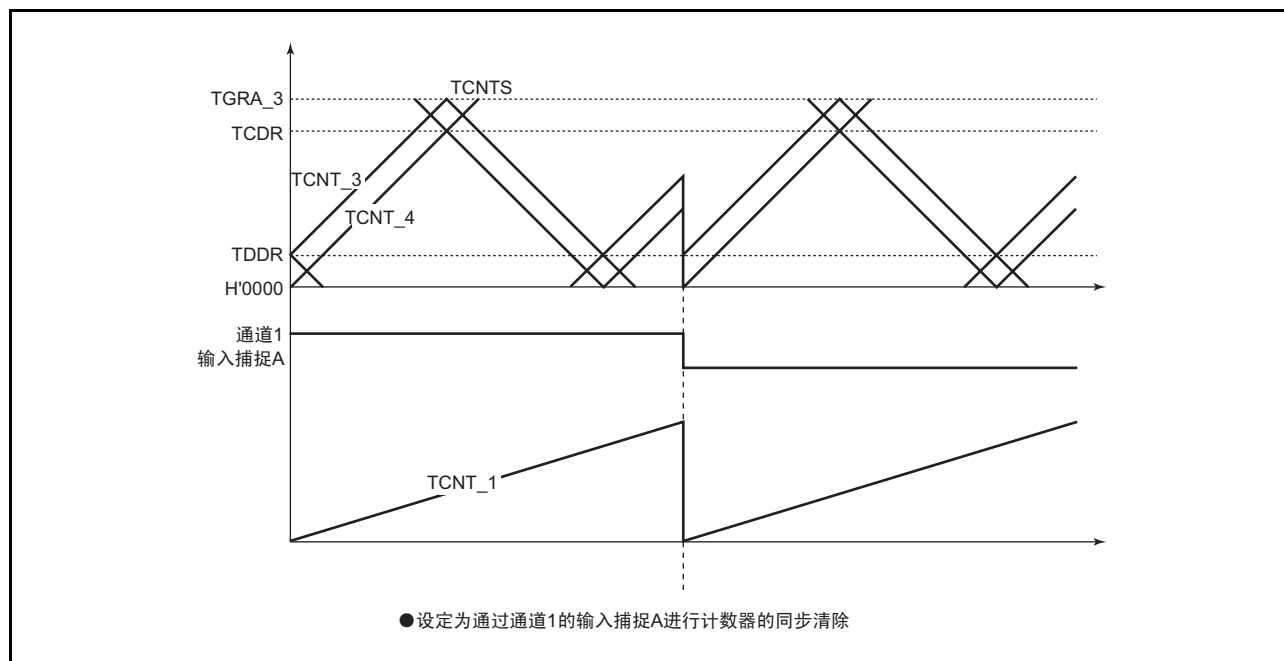


图 10.55 与其他通道同步的计数器清除

(n) 互补 PWM 模式的同步计数器清除时的输出波形控制

能通过将 TWCR 寄存器的 WRE 位置 1, 抑制在互补 PWM 模式的波谷 Tb 区间发生同步计数器清除时的初始输出。因此, 能抑制同步计数器清除时的占空比的急剧变化。

只在如图 10.56 的(10)、(11)的波谷 Tb 区间进行同步清除时, 才能将 WRE 位置 1 来抑制初始输出。如果在其他时序中发生同步清除, 就输出 TOCR 寄存器的 OLS 位设定的初始值。另外, 即使在波谷的 Tb 区间, 如果在图 10.56 的(1)所示的计数器刚开始的初始输出期间发生同步清除, 也不抑制初始输出。

此功能能用于 MTU2 和 MTU2S。MTU2 和 MTU2S 计数器的清除源分别是 MTU2 的通道 0 ~ 2 的同步清除和 MTU2 的通道 0 ~ 2 的标志置位 (比较匹配 / 输入捕捉)。

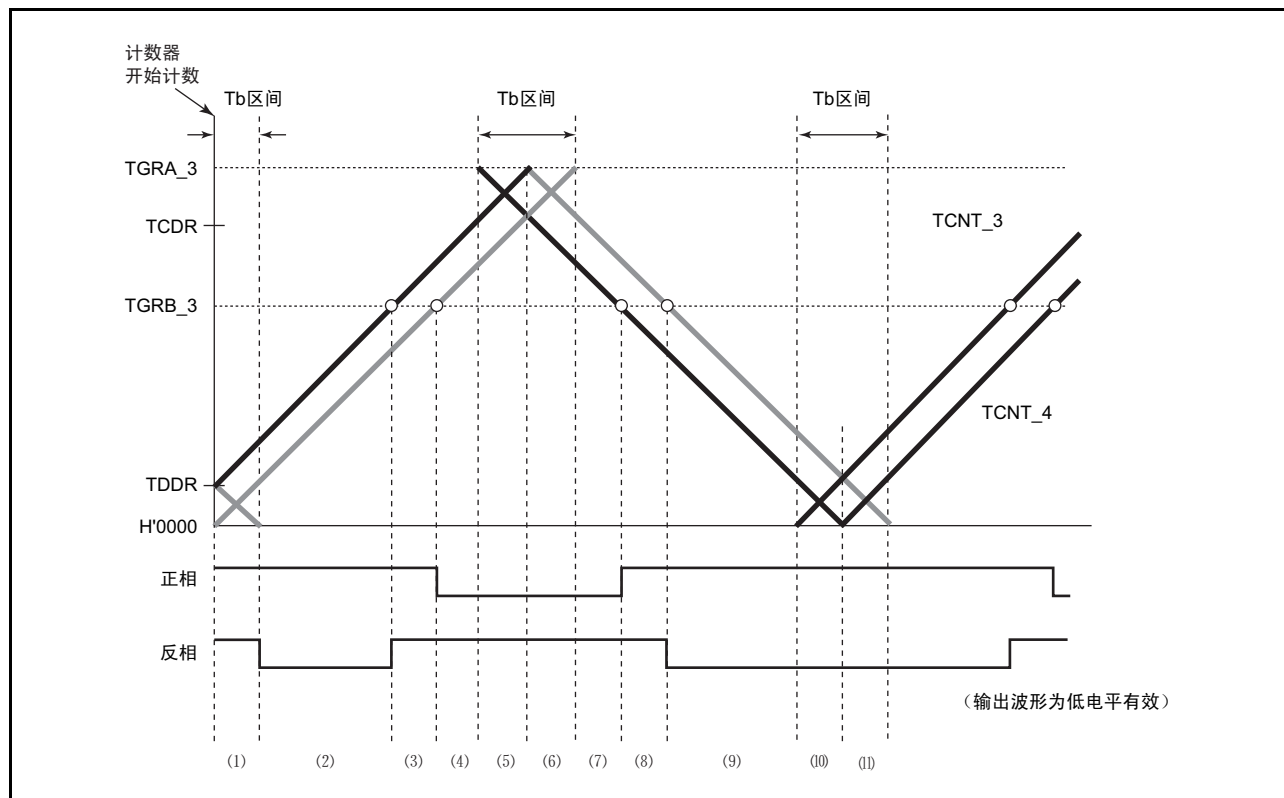


图 10.56 同步计数器清除的时序

- 互补 PWM 模式的同步计数器清除时的输出波形控制的设定步骤例子

互补 PWM 模式的同步计数器清除时的输出波形控制的设定步骤例子如图 10.57 所示。

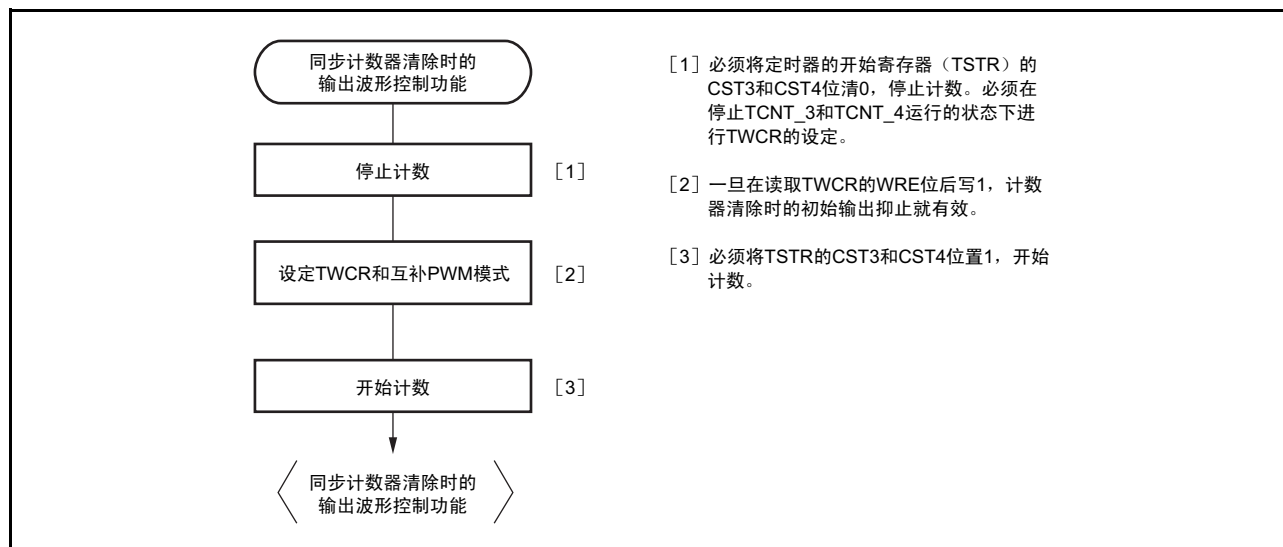


图 10.57 互补 PWM 模式的同步计数器清除时的输出波形控制的设定步骤例子

- 互补 PWM 模式的同步计数器清除时的输出波形控制的运行例子

在将 TWCR 的 WRE 位置 1 的状态下使 MTU2 进行互补 PWM 运行，并进行同步计数器清除时的运行例子如图 10.58 ~ 图 10.61 所示。在此，图 10.58 ~ 图 10.61 的同步计数器清除的时序分别为图 10.56 的 (3)、(6)、(8)、(11) 所示的时序。

在此例中，对于 MTU2S，在将 TWCR 的 SCC 位置 0 以及将 WRE 位置 1 的状态下，使 MTU2S 进行互补 PWM 运行，相当于同步清除计数器的情况。

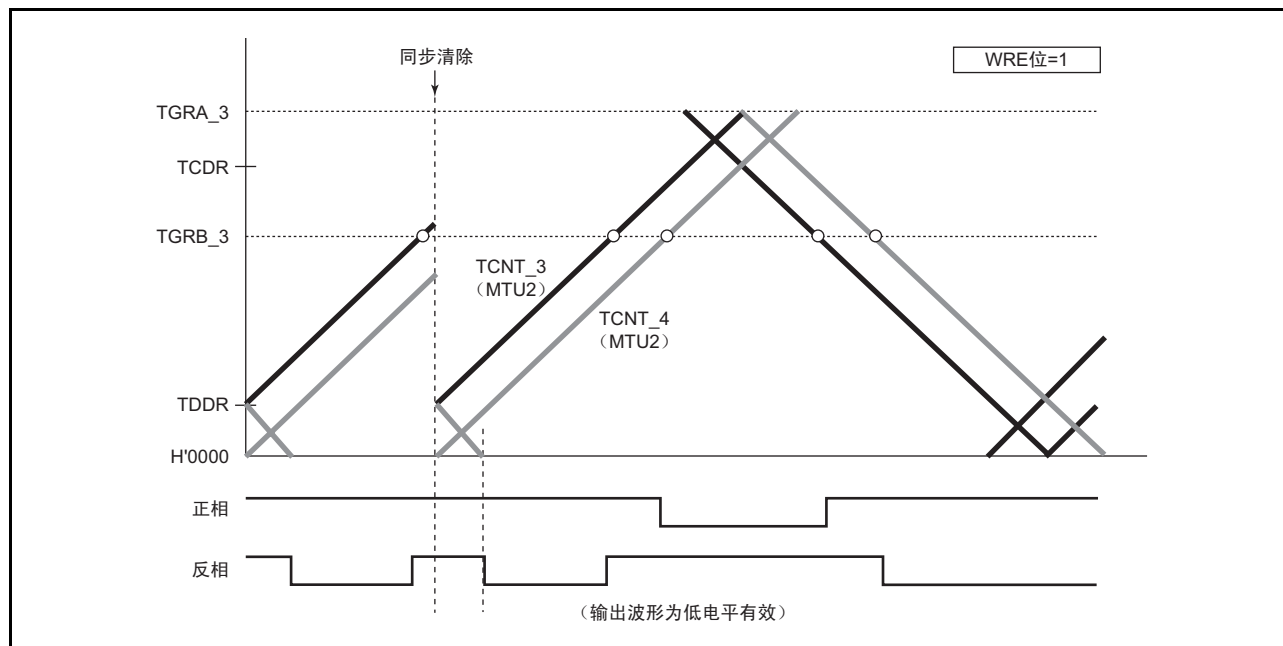


图 10.58 在递增计数中的空载时间，发生同步清除
(图 10.56 的时序(3)，MTU2 的 TWCR 寄存器的 WRE 位 = 1)

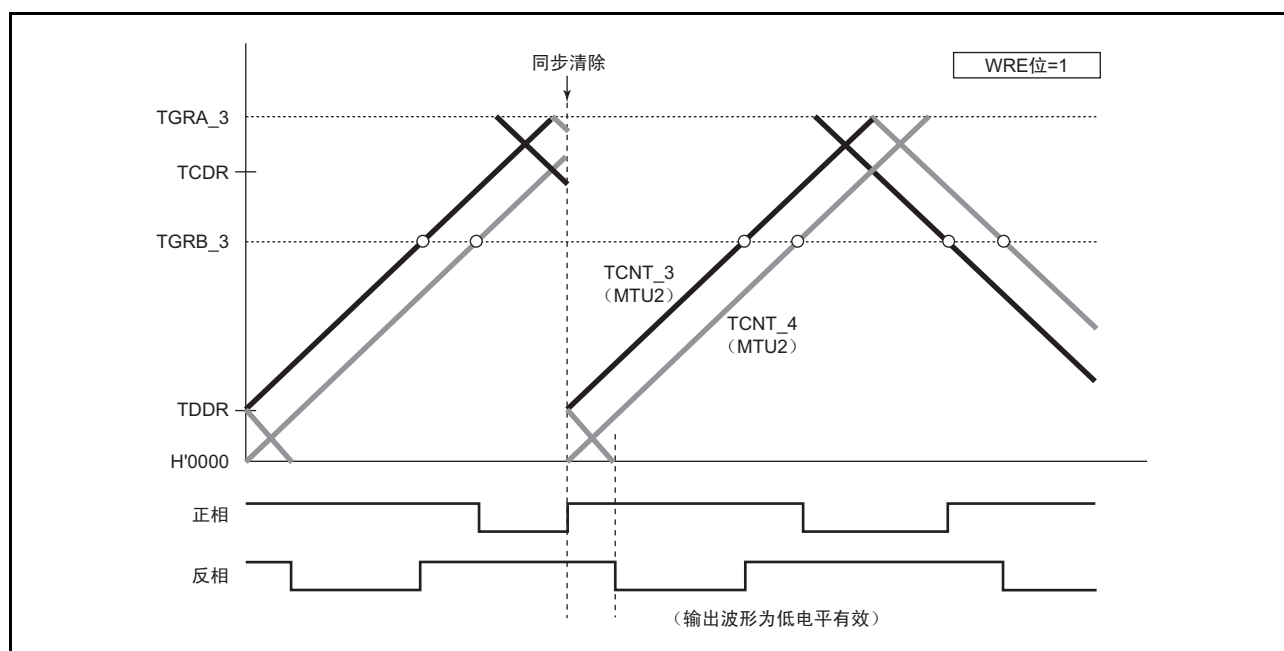


图 10.59 在波峰的 Tb 区间，发生同步清除
(图 10.56 的时序(6)，MTU2 的 TWCR 寄存器的 WRE 位 =1)

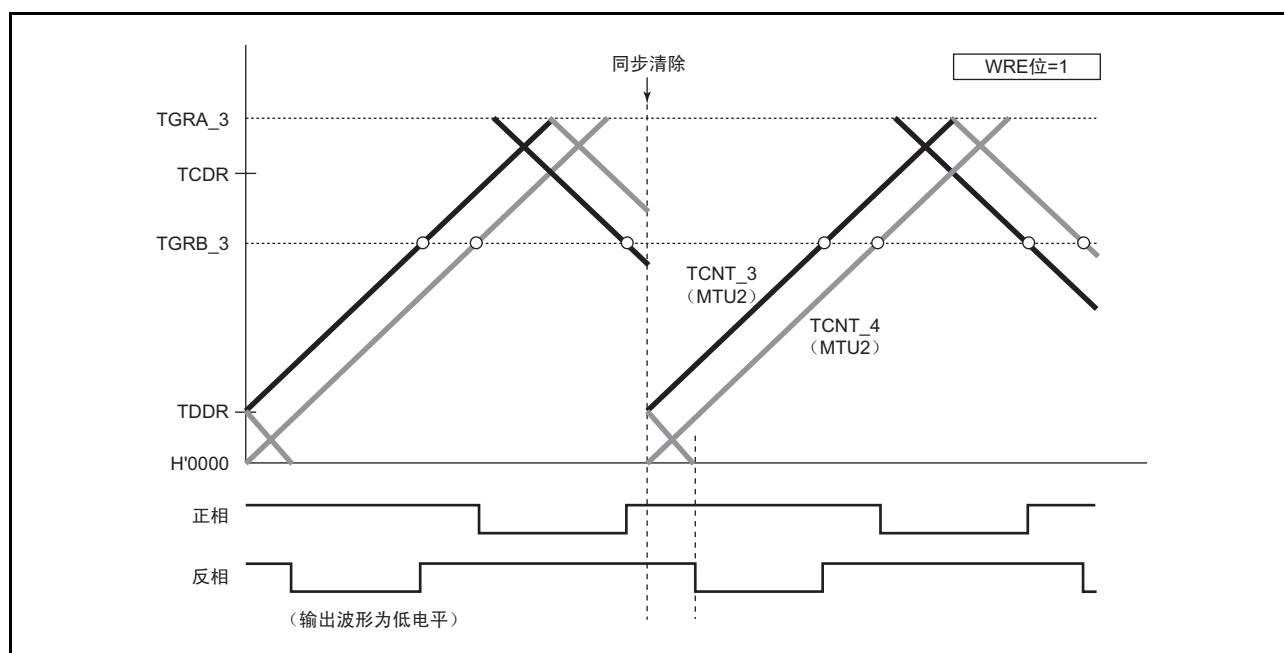


图 10.60 在递减计数中的空载时间，发生同步清除
(图 10.56 的时序(8)，TWCR 寄存器的 WRE 位 =1)

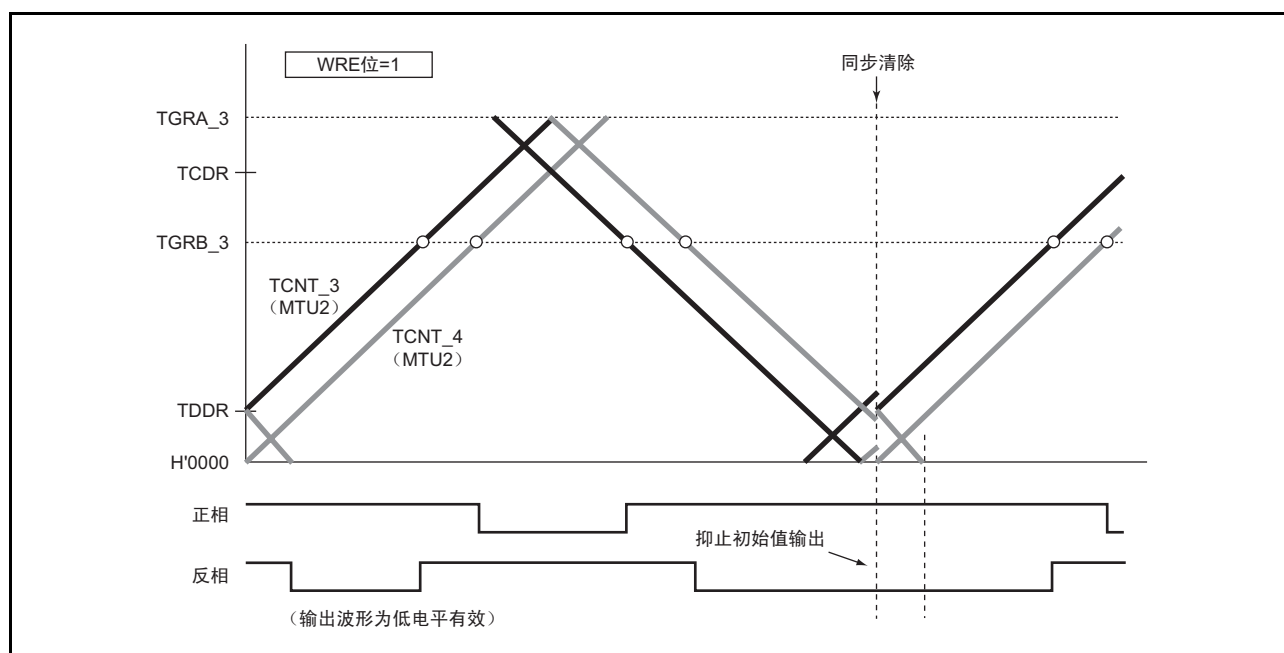


图 10.61 在波谷的 Tb 区间，发生同步清除
(图 10.56 的时序(1), TWCR 寄存器的 WRE 位 = 1)

(o) MTU2-MTU2S 计数器同步清除的抑止功能

对于 MTU2S，能通过将 TWCR 寄存器的 SCC 位置 1，抑止由 MTU2 进行同步清除。

能通过设定 SCC 位抑止由 MTU2 进行同步清除的区间如图 10.62 所示。

在使用此功能时，必须将 MTU2S 设定为互补 PWM 模式。

有关通过 MTU2 进行同步清除的详细内容，请参照“10.4.10(2) 利用 MTU2 标志的置位源进行 MTU2S 计数器清除 (MTU2-MTU2S 计数器的同步清除)”。

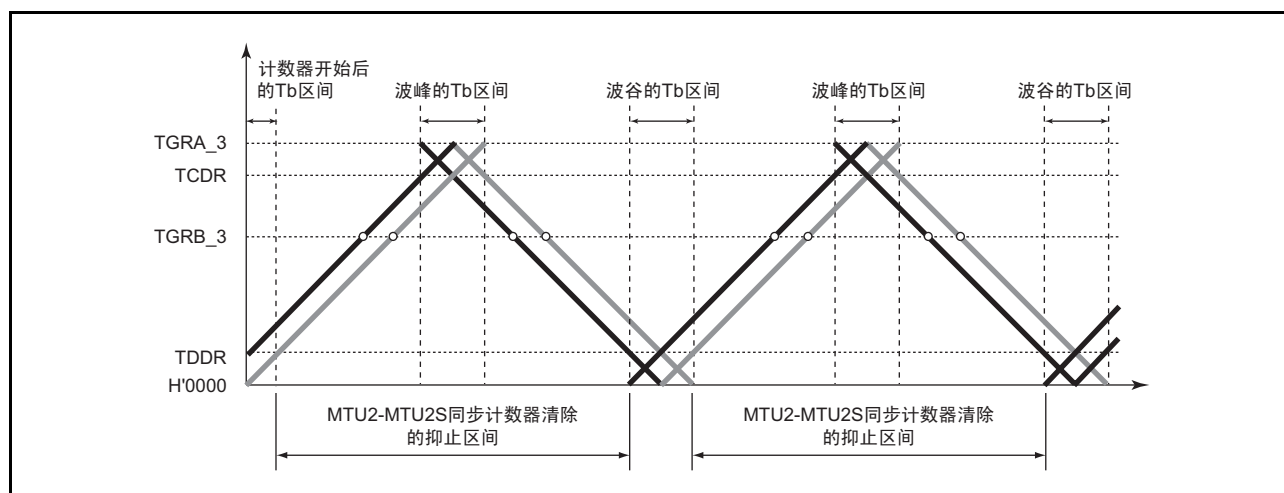


图 10.62 通过 TWCR 的 SCC 位的置位，抑止 MTU2-MTU2S 同步清除的区间

- 设定 MTU2-MTU2S 计数器同步清除抑止功能的设定步骤例子

设定 MTU2-MTU2S 计数器同步清除抑止功能的设定步骤例子如图 10.63 所示。

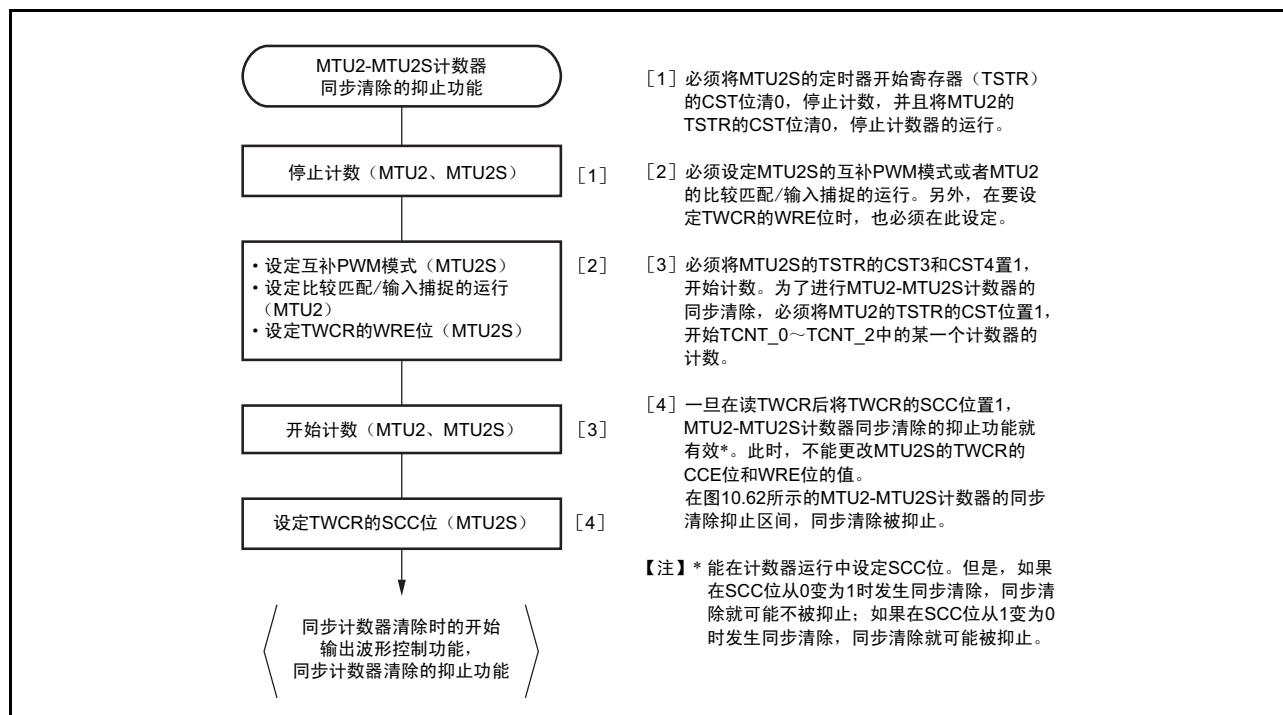


图 10.63 设定 MTU2-MTU2S 计数器同步清除抑止功能的设定步骤例子

- MTU2-MTU2S 计数器同步清除抑止功能的运行例子

如图 10.64~图 10.67 所示，将 MTU2S 的 TWCR 的 SCC 位置 1，使 MTU2S 进行互补 PWM 运行，并使 MTU2-MTU2S 计数器同步清除抑止功能有效。在此，图 10.64~图 10.67 的计数器同步清除时序分别为图 10.56 的(3)、(6)、(8)、(11)所示的时序，并将 MTU2S 的 TWCR 的 WRE 位置 1。

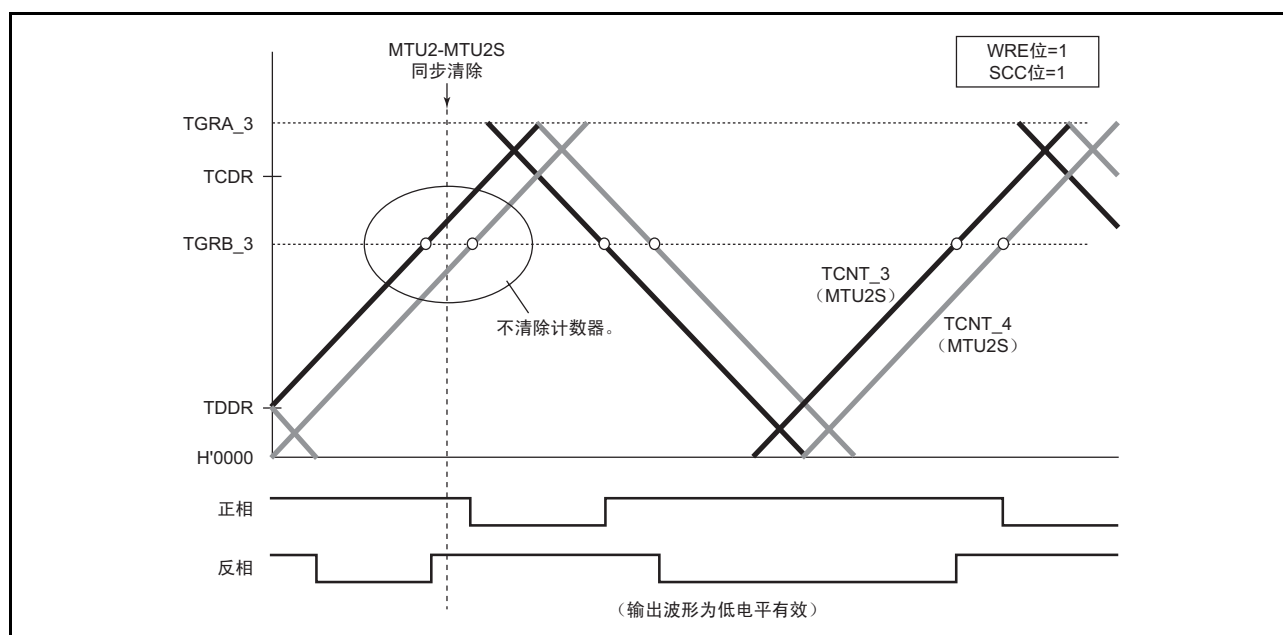


图 10.64 在递增计数中的空载时间，发生同步清除
(图 10.56 的时序(3)，MTU2S 的 TWCR 寄存器的 WRE 位 =1、SCC 位 =1)

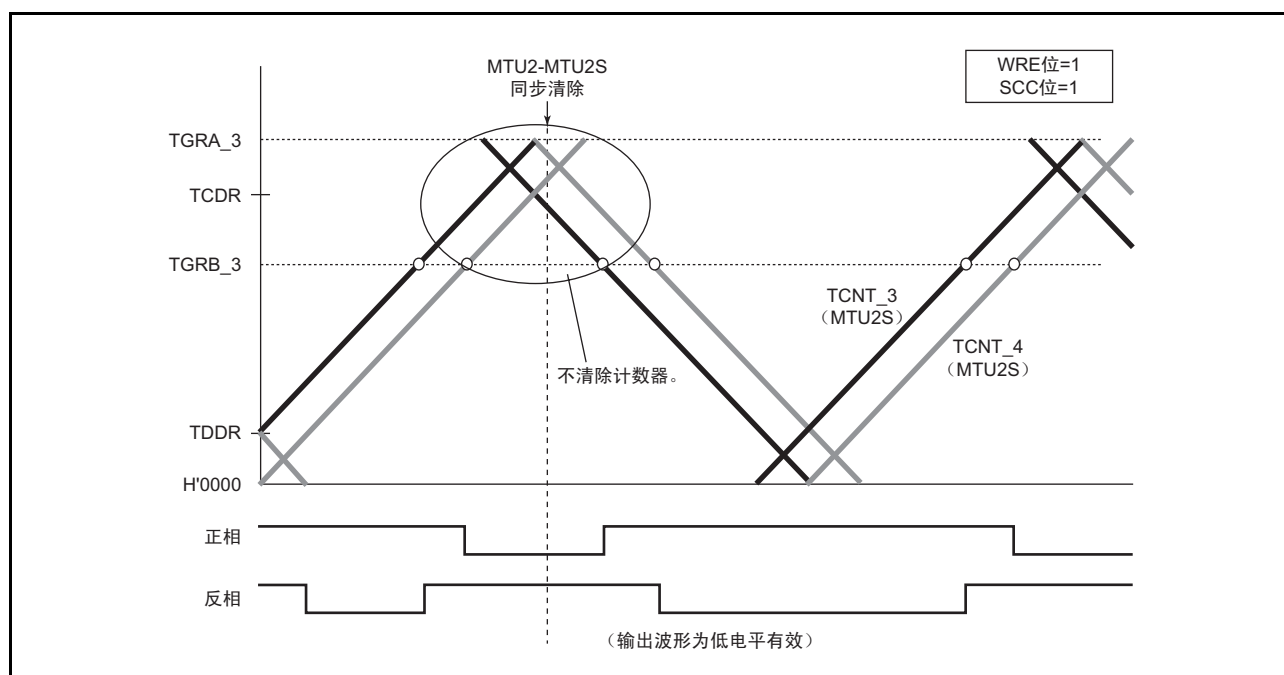


图 10.65 在波峰的 T_d 区间，发生同步清除
(图 10.56 的时序(6)，MTU2S 的 TWCR 寄存器的 WRE 位 =1、SCC 位 =1)

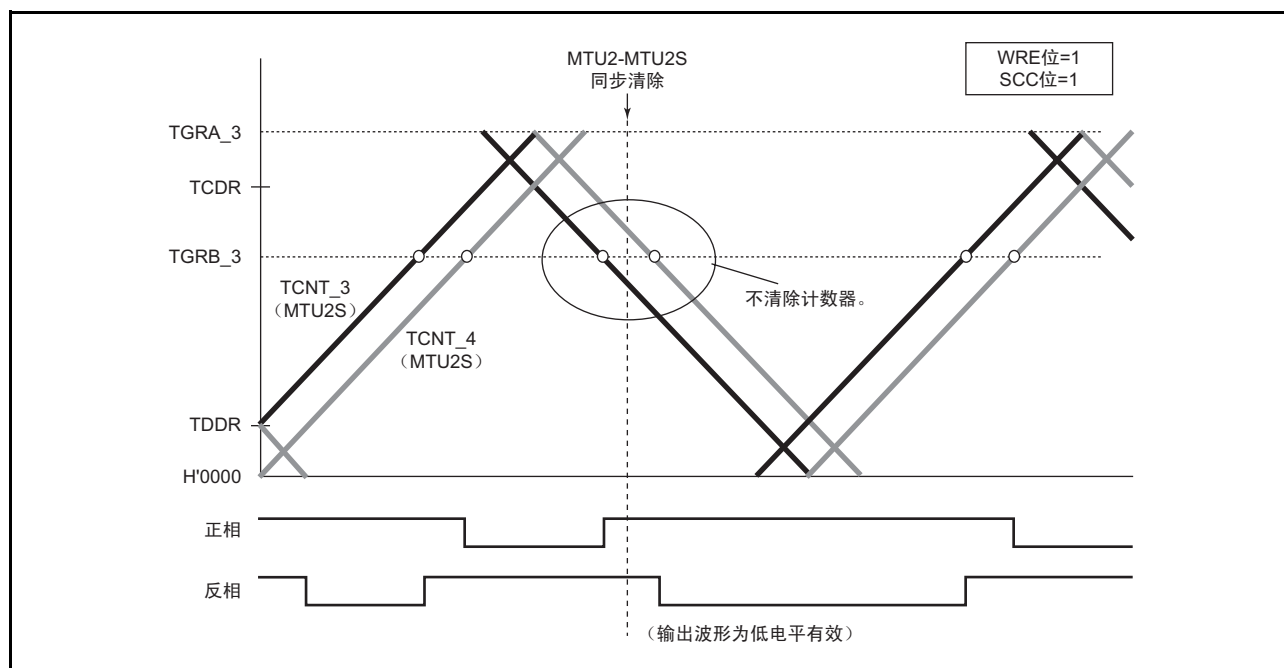


图 10.66 在递减计数中的空载时间，发生同步清除
(图 10.56 的时序(8)，MTU2S 的 TWCR 寄存器的 WRE 位 =1、SCC 位 =1)

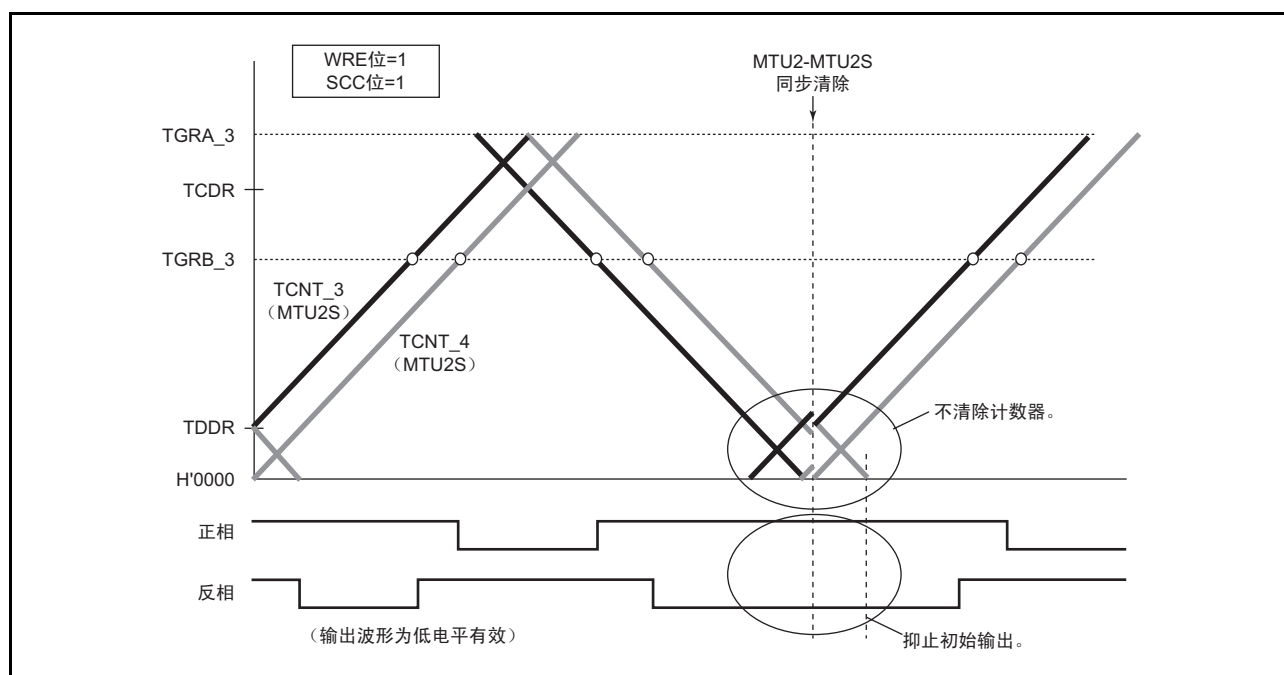


图 10.67 在波谷的 Tb 区间，发生同步清除
(图 10.56 的时序(II)，MTU2S 的 TWCR 寄存器的 WRE 位 =1、SCC 位 =1)

(p) 通过 TGRA_3 的比较匹配清除计数器

在互补 PWM 模式中，能通过设定定时器的波形控制寄存器 (TWCR) 的 CCE 位，在 TGRA_3 的比较匹配清除 TCNT_3、TCNT_4 和 TCNTS。

运行例子如图 10.68 所示。

- 【注】
1. 只能用于互补 PWM 模式 1 (在波峰传送)。
 2. 不能设定为与其他通道同步的清除功能。(不能将定时器的同步寄存器 (TSYR) 的 SYNC0 ~ SYNC4 位和定时器的同步清除寄存器 (TSYCR) 的 CE0A/B/C/D 位、CE1A/B/C/D 位置 1)
 3. 不能将 PWM 占空比置 0。
 4. 不能将定时器的输出控制寄存器 1 (TOCR1) 的 PSYE 位置 1。

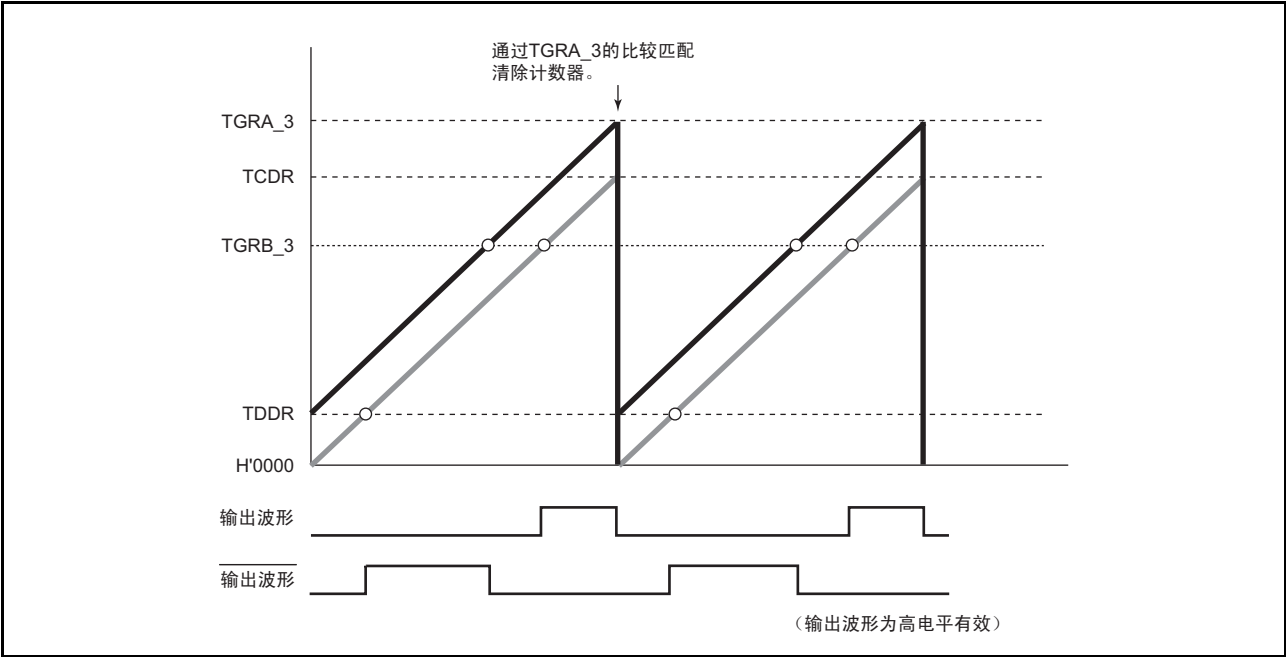


图 10.68 通过 TGRA_3 的比较匹配清除计数器的运行例子

(q) AC 同步马达（无刷 DC 马达）的驱动波形输出例子

在互补 PWM 模式中，能使用定时器的门控寄存器（TGCR），简单地控制无刷 DC 马达。使用 TGCR 的无刷 DC 马达的驱动波形例子如图 10.69 ~ 图 10.72 所示。

为了转换 3 相无刷 DC 马达的输出相，在使用霍尔元件等检测外部信号时，将 TGCR 的 FB 位置 0。此时，将表示磁极位置的外部信号输入到通道 0 的定时器的输入引脚 TI0C0A、TI0C0B 和 TI0C0C 引脚（必须通过 PFC 进行设定）。如果 TI0C0A、TI0C0B 和 TI0C0C 这 3 个引脚产生边沿，输出的 ON/OFF 就自动转换。

当 FB 位为 1 时，如果将 TGCR 的 UF 位、VF 位和 WF 位置 0 或者 1，就会转换输出的 ON/OFF。

从互补 PWM 模式的 6 相输出引脚进行驱动波形的输出。对于此 6 相输出，能通过将 N 位或者 P 位置 1，在 ON 输出时，使用互补 PWM 模式的输出进行载波输出。在 N 位或者 P 位为 0 时，为电平输出。

另外，6 相输出的有效电平（ON 输出时的电平）与 N 位和 P 位的设定无关，能通过定时器的输出控制寄存器（TOCR）的 OLSN 位和 OLSP 位进行设定。

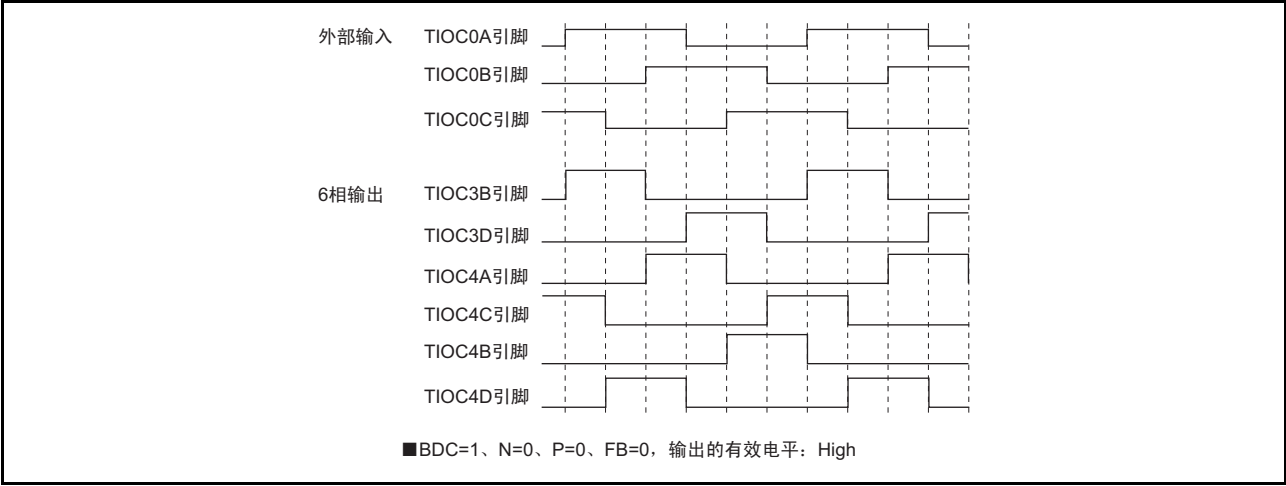


图 10.69 通过外部输入转换输出相的运行例子（1）

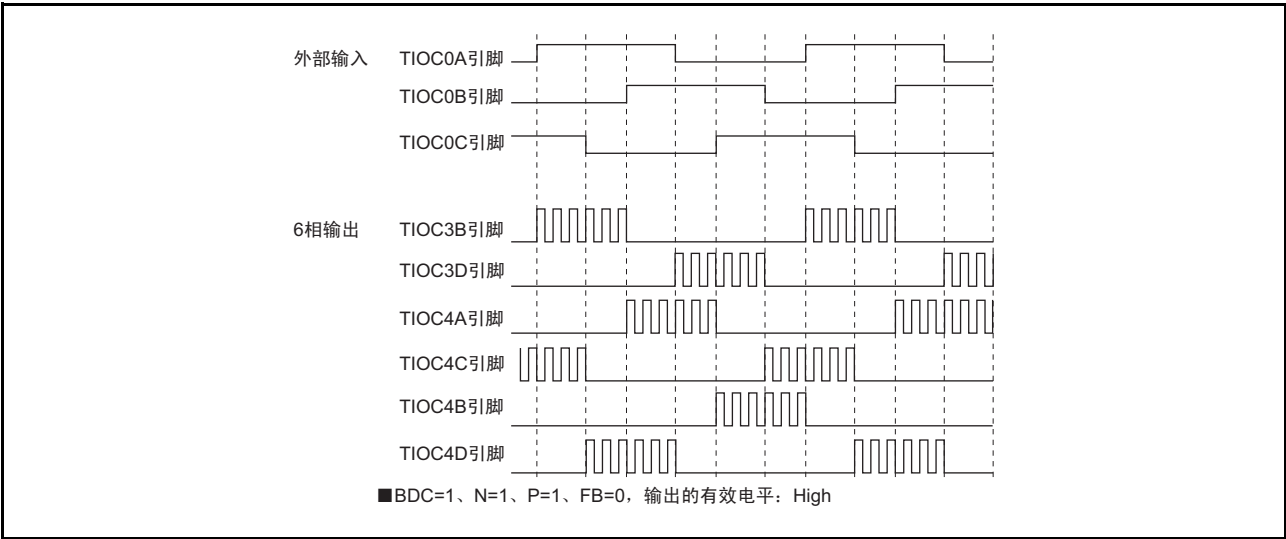


图 10.70 通过外部输入转换输出相的运行例子 (2)

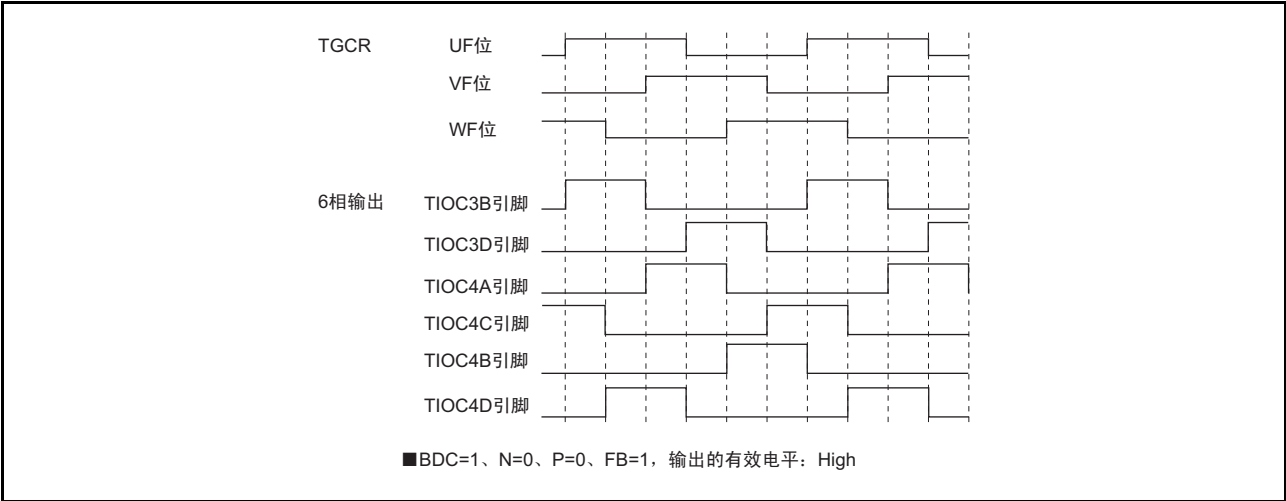


图 10.71 通过设定 UF 位、VF 位和 WF 位转换输出相的运行例子 (1)

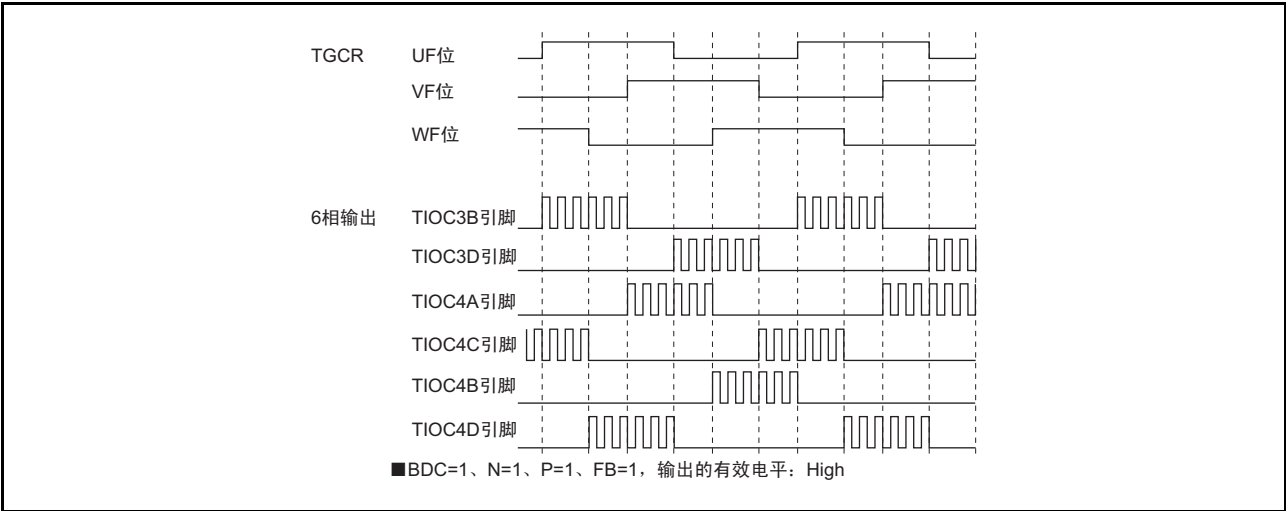


图 10.72 通过设定 UF 位、VF 位和 WF 位转换输出相的运行例子 (2)

(r) A/D 转换开始请求的设定

在互补 PWM 模式中，能使用 TGRA_3 的比较匹配、TCNT_4 的下溢（波谷）和通道 3、4 以外通道的比较匹配进行 A/D 转换的开始请求。

如果使用 TGRA_3 的比较匹配设定开始请求，就能在 TCNT_3 的波峰开始 A/D 转换。

能通过将定时器的中断允许寄存器（TIER）的 TTGE 位置 1，设定 A/D 转换的开始请求，并能通过将 TIER_4 的 TTGE2 位置 1，设定 TCNT_4 的下溢（波谷）的 A/D 转换的开始请求。

(3) 互补 PWM 模式的中断减少功能

通过设定定时器的中断减少设定寄存器（TITCR），最多能使通道 3 和通道 4 的 TGIA_3（波峰的中断）和 TCIV_4（波谷的中断）减少 7 次中断。

通过设定定时器的缓冲传送寄存器（TBTER），能联动缓冲寄存器到暂寄存器 / 比较寄存器的数据传送，减少中断。有关和缓冲寄存器的联动，请参照“(c) 和中断减少功能联动的缓冲传送控制”。

通过设定定时器的 A/D 转换请求控制寄存器（TADCR），能联动 A/D 转换开始请求延迟功能的 A/D 转换开始请求，减少中断。有关和 A/D 转换请求延迟功能的联动，请参照“10.4.9 A/D 转换开始请求的延迟功能”。

必须通过设定 TIER_3 和 TIER_4 寄存器，在禁止 TGIA_3 和 TCIV_4 的中断请求以及不产生比较匹配引起的 TGFA_3 和 TCFV_4 标志置位的状态下，设定定时器的中断减少设定寄存器（TITCR）。另外，在更改减少次数前，必须将 T3AEN 位和 T4VEN 位置 0，清除减少计数器。

(a) 中断减少功能的设定步骤例子

中断减少功能的设定步骤例子和中断减少次数的可更改期间分别如图 10.73 和图 10.74 所示。

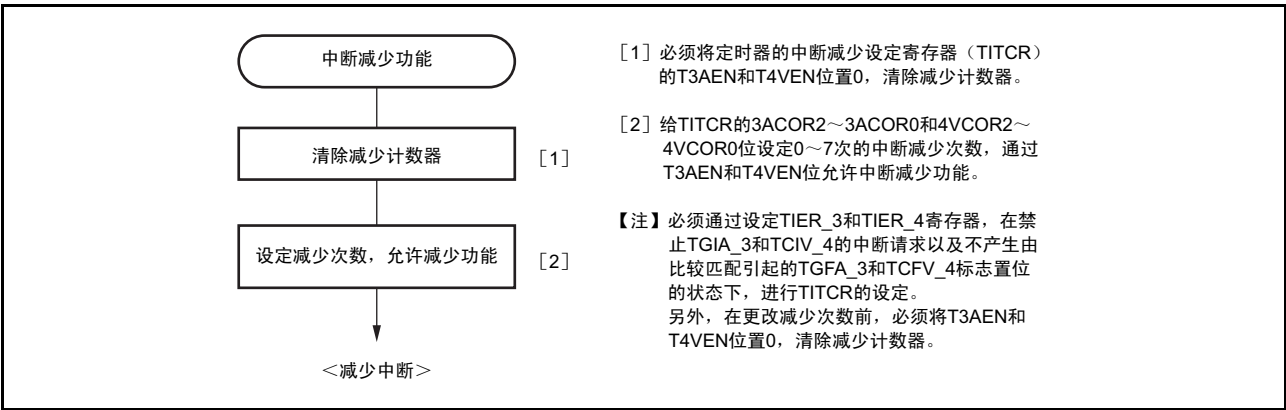


图 10.73 中断减少功能的设定步骤例子

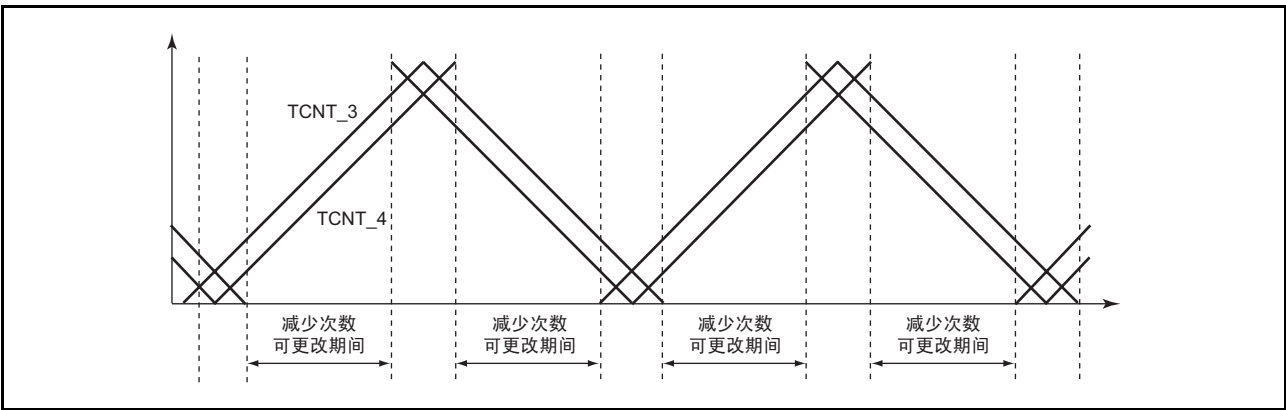


图 10.74 中断减少次数的可更改期间

(b) 中断减少功能的运行例子

在通过定时器的中断减少设定寄存器 (TITCR) 的 3ACOR 位将中断减少次数设定为 3 次并将 T3AEN 位置 1 时, TGIA_3 中断减少功能的运行例子如图 10.75 所示。

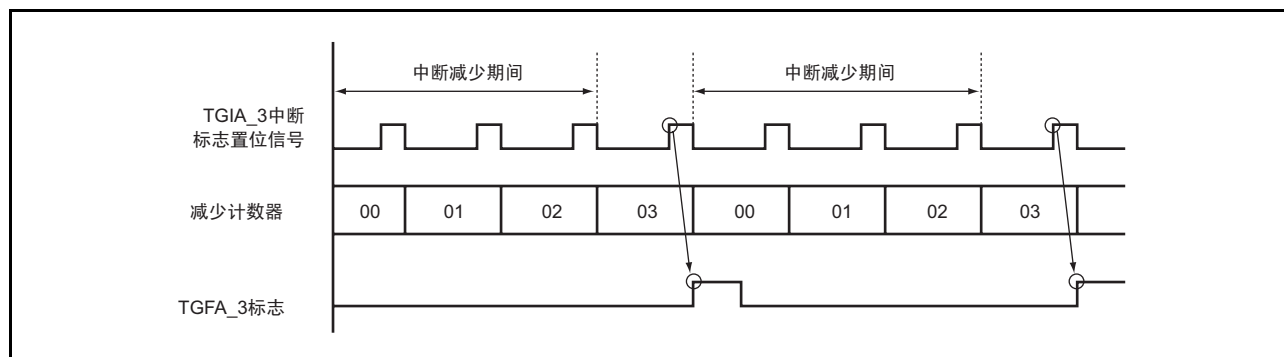


图 10.75 中断减少功能的运行例子

(c) 和中断减少功能联动的缓冲传送控制

通过设定定时器的缓冲传送设定寄存器 (TBTER) 的 BTE1 位和 BTE0 位, 能选择在互补 PWM 模式中是否进行缓冲寄存器到暂寄存器的缓冲传送, 或者设定是否联动中断减少功能。

设定为抑制缓冲传送 (BTE1=0、BTE0=1) 时的运行例子如图 10.76 所示。在此设定期间, 不将缓冲寄存器的值传送到暂寄存器。

设定为缓冲传送联动中断减少功能 (BTE1=1、BTE0=0) 时的运行例子如图 10.77 所示。在缓冲传送允许期间以外的期间不进行缓冲寄存器到暂寄存器的传送。

另外, 如果将定时器的中断减少设定寄存器 (TITCR) 的 T3AEN 位、T4VEN 位或者 T3AEN/T4VEN 位置 1 时, 各缓冲传送允许期间就不同。TITCR 的 T3AEN 位和 T4VEN 位的设定与缓冲传送允许期间的关系如图 10.78 所示。

【注】 此功能必须和中断减少功能配合使用。

在禁止中断减少功能时 (将定时器的中断减少设定寄存器 (TITCR) 的 T3AEN 位、T4VEN 位或者 TITCR 的减少次数设定位 (3ACOR 和 4VCOR) 置 0 时), 必须设定为缓冲传送不联动中断减少功能 (将定时器的缓冲传送设定寄存器 (TBTER) 的 BTE1 置 0), 否则不进行缓冲传送。

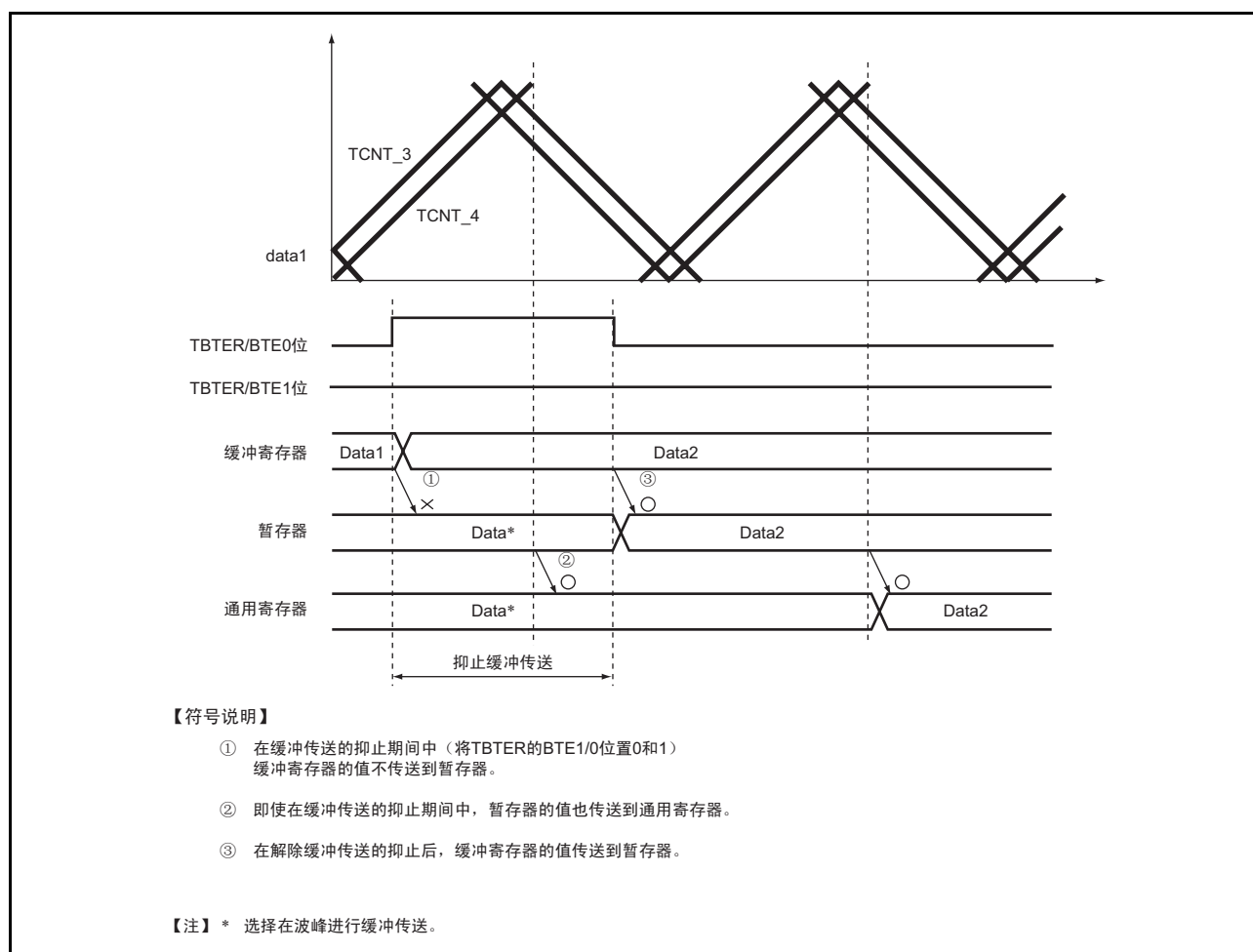


图 10.76 设定为抑制缓冲传送（BTE1=0、BTE0=1）时的运行例子

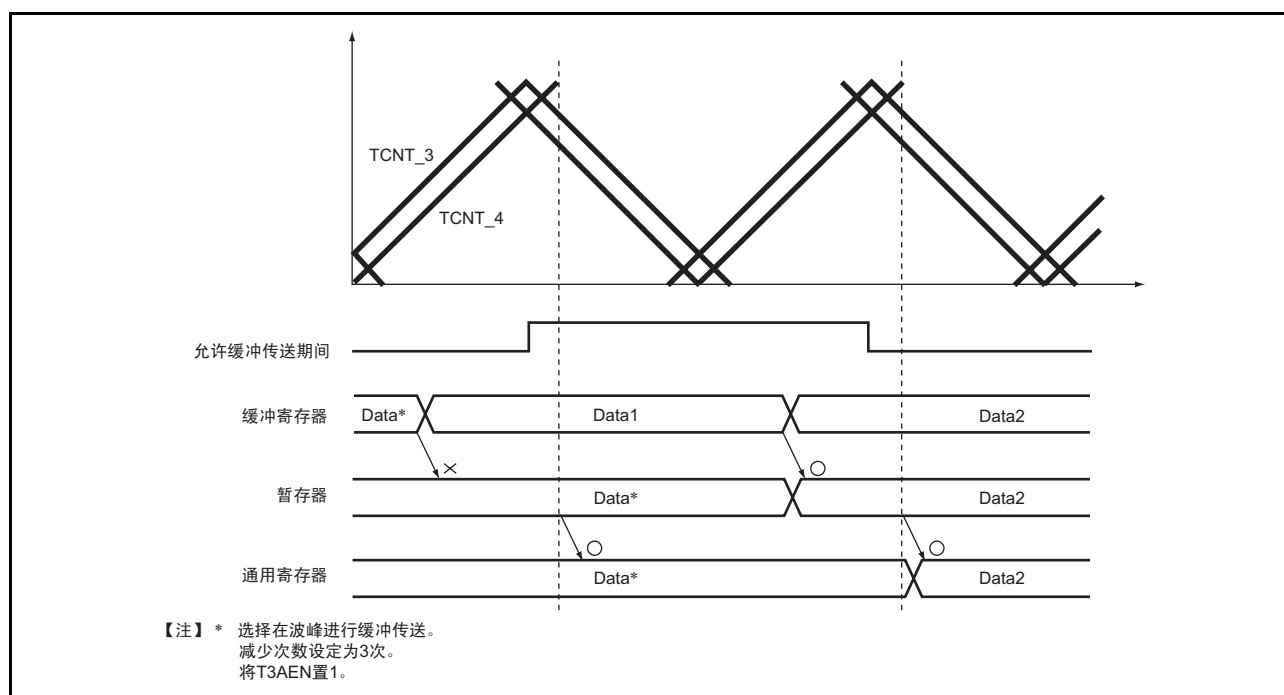


图 10.77 设定为缓冲传送联动中断减少功能（BTE1=1、BTE0=0）时的运行例子

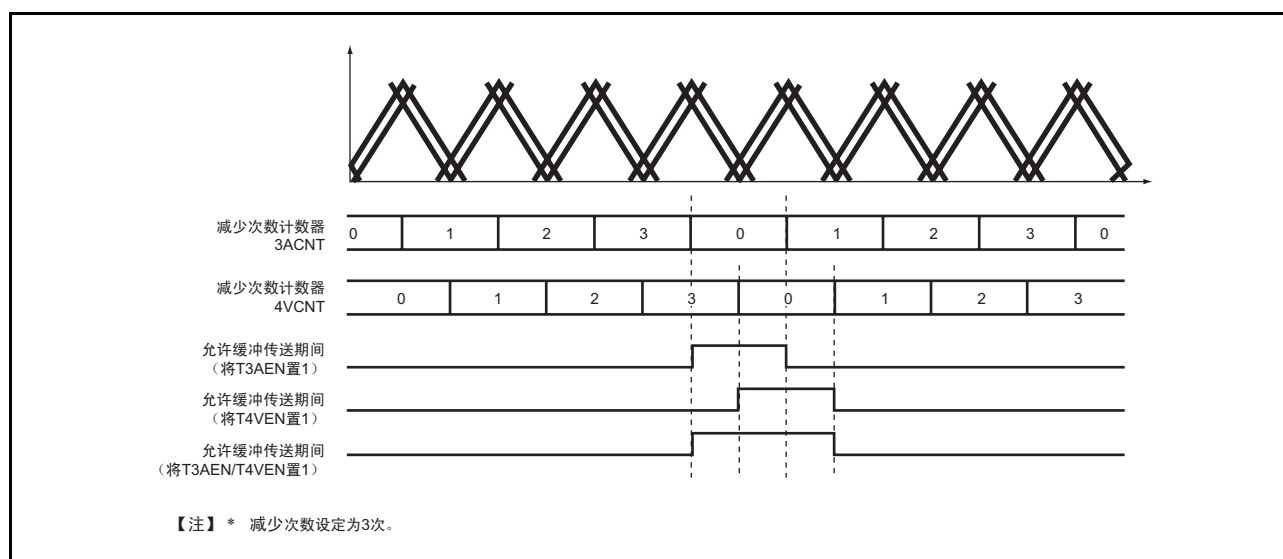


图 10.78 定时器的中断减少设定寄存器 (TITCR) 的 T3AEN 位和 T4VEN 位的设定与缓冲传送允许期间的关系

(4) 互补 PWM 模式的输出保护功能

互补 PWM 模式的输出具有以下保护功能:

(a) 寄存器和计数器的误写防止功能

在互补 PWM 模式使用的寄存器和计数器中, 除了随时能改写的缓冲寄存器以外, 能通过设定定时器的读写允许寄存器 (TRWER) 的 RWE 位, 允许或者禁止由 CPU 存取模式寄存器、控制寄存器、比较寄存器和计数器。对象寄存器为通道 3 和通道 4 的一部分寄存器, 如下所示:

TCR_3 和 TCR_4、TMDR_3 和 TMDR_4、TIOH_3 和 TIOH_4、TIOH_3 和 TIOH_4、TIOH_3 和 TIOH_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。

通过此功能, 将模式寄存器、控制寄存器和计数器等设定为禁止由 CPU 进行的存取, 能防止因 CPU 失控而产生的误写。在禁止存取的状态下读对象寄存器时, 读取值为不定值, 写操作无效。

(b) 通过外部信号停止 PWM 输出的功能

能通过输入所指定的外部信号, 将 6 相 PWM 输出引脚自动置为高阻抗状态。

详细内容请参照“第 12 章 端口输出的允许 (POE)”。

(c) 振荡停止时的 PWM 输出停止功能

当检测出输入到本 LSI 的时钟已停止时, 6 相 PWM 输出引脚就自动变为高阻抗状态。如果时钟又开始振荡, 就不保证引脚的状态。

详细内容请参照“4.7 振荡停止的检测功能”。

10.4.9 A/D 转换开始请求的延迟功能

能通过设定通道 4 的定时器 A/D 转换开始请求的控制寄存器 (TADCR)、定时器的 A/D 转换开始请求的周期寄存器 (TADCORA_4 和 TADCORB_4) 和定时器的 A/D 转换开始请求的周期缓冲寄存器 (TADCOBRA_4 和 TADCOBRB_4)，进行 A/D 转换的开始请求。

TCNT_4 和 TADCORA_4、TADCORB_4 比较，当 TCNT_4 和 TADCORA_4、TADCORB_4 相同时，A/D 转换开始请求的延迟功能就能进行各自的 A/D 转换开始请求 (TRG4AN 和 TRG4BN)。

另外，通过设定 TADCR 的 ITA3AE 位、ITA4VE 位、ITB3AE 位和 ITB4VE 位，能联动中断减少功能来减少 A/D 转换的开始请求 (TRG4AN 和 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 10.79 所示。

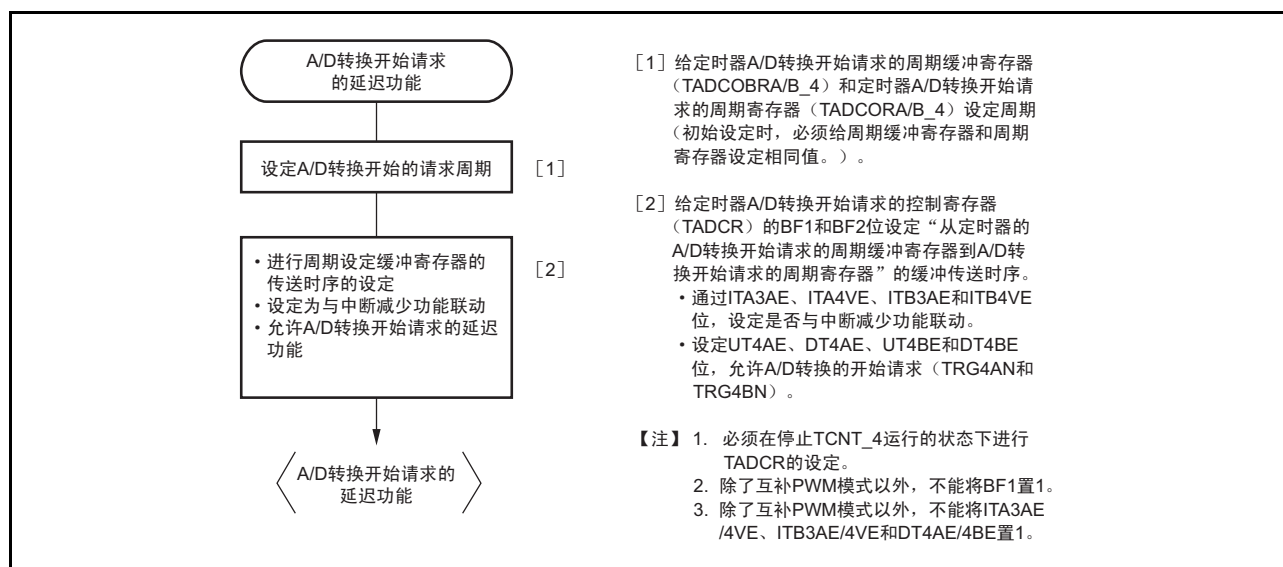


图 10.79 A/D 转换开始请求延迟功能的设定步骤例子

(b) A/D 转换开始请求延迟功能的基本运行例子

将缓冲传送时序设定为 TCNT_4 的波谷，并且在 TCNT_4 进行递减计数时输出 A/D 转换的开始请求信号 (TRG4AN) 的情况下，A/D 转换的开始请求信号 (TRG4AN) 的基本运行例子如图 10.80 所示。

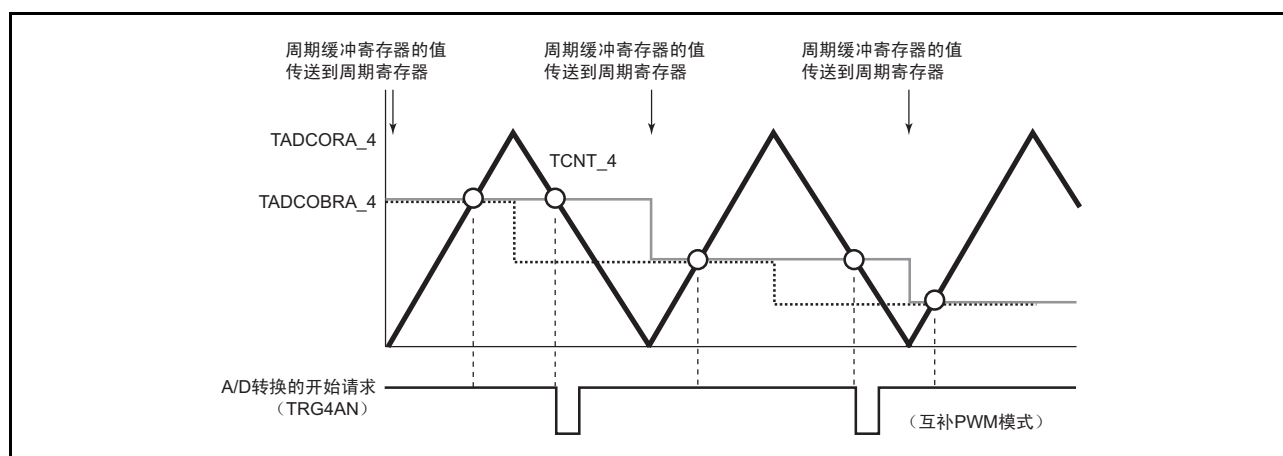


图 10.80 A/D 转换的开始请求信号 (TRG4AN) 的基本运行例子

(c) 缓冲传送

通过将数据写到定时器的 A/D 转换开始请求的周期设定缓冲寄存器 (TADCOBRA/B_4)，更新定时器的 A/D 转换开始请求的周期设定寄存器 (TADCORA/B_4) 的数据。能通过设定定时器 A/D 转换开始请求控制寄存器 (TADCR_4) 的 BF1 位和 BF0 位，选择从定时器的 A/D 转换开始请求的周期设定缓冲寄存器到定时器的 A/D 转换开始请求的周期设定寄存器的传送时序。

(d) 和中断减少功能联动的 A/D 转换开始请求的延迟功能

通过设定定时器的 A/D 转换开始请求控制寄存器 (TADCR) 的 ITA3AE 位、ITA4VE 位、ITB3AE 位和 ITB4VE 位，能联动中断减少功能进行 A/D 转换开始请求 (TRG4AN 和 TRG4BN)。在 TCNT_4 进行递增计数或者递减计数时允许 TRG4AN 输出，并且在联动中断减少功能的情况下 A/D 转换开始请求信号 (TRG4AN) 的运行例子如图 10.81 所示。

另外，在 TCNT_4 进行递增计数时允许 TRG4AN 输出，并且在联动中断减少功能的情况下 A/D 转换开始请求信号 (TRG4AN) 的运行例子如图 10.82 所示。

【注】 此功能必须和中断减少功能配合使用。

在禁止中断减少功能时 (将定时器的中断减少设定寄存器 (TITCR) 的 T3AEN 位、T4VEN 位或者 TITCR 减少次数的设定位 (3ACOR 和 4VCOR) 置 0 时)，必须设定为不联动中断减少功能 (将定时器的 A/D 转换开始请求的控制寄存器 (TADCR) 的 ITA3AE 位、ITA4VE 位、ITB3AE 位和 ITB4VE 位置 0)。

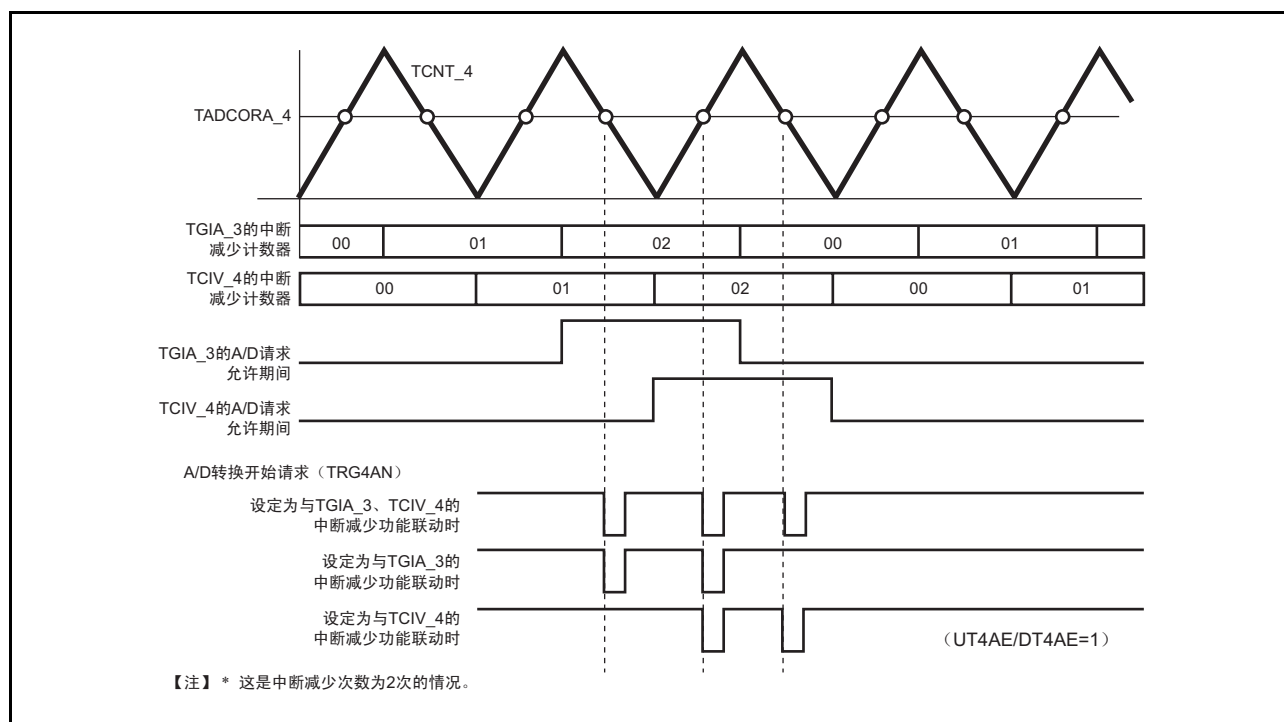


图 10.81 联动中断减少功能时的 A/D 转换开始请求信号 (TRG4AN) 的运行例子

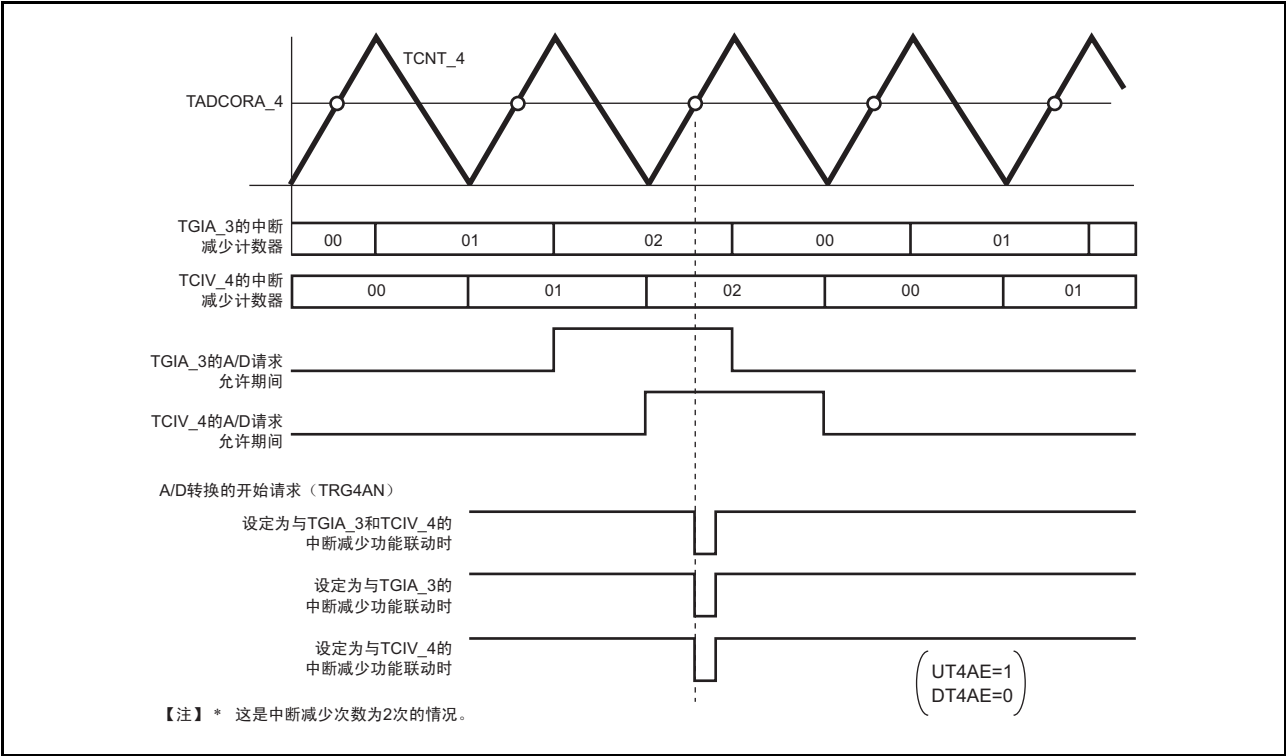


图 10.82 联动中断减少功能时的 A/D 转换开始请求信号 (TRG4AN) 的运行例子

10.4.10 MTU2-MTU2S 的同步运行

(1) MTU2-MTU2S 计数器的同步开始

通过设定 MTU2 的 TCSYSTR 寄存器，能同步开始不同时钟运行的 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 10.83 所示。

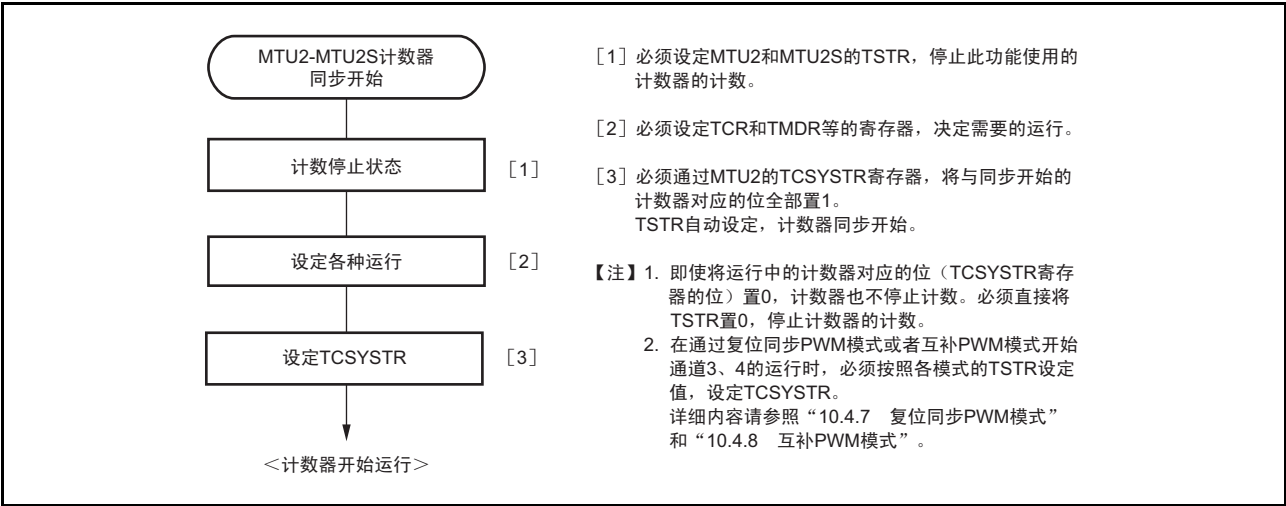


图 10.83 计数器同步开始的设定步骤例子

(b) 计数器同步开始的运行例子

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的运行例子分别如图 10.84（1）、图 10.84（2）、图 10.84（3）、图 10.84（4）所示。

在这些例中，计数时钟设定为 MPφ/1。

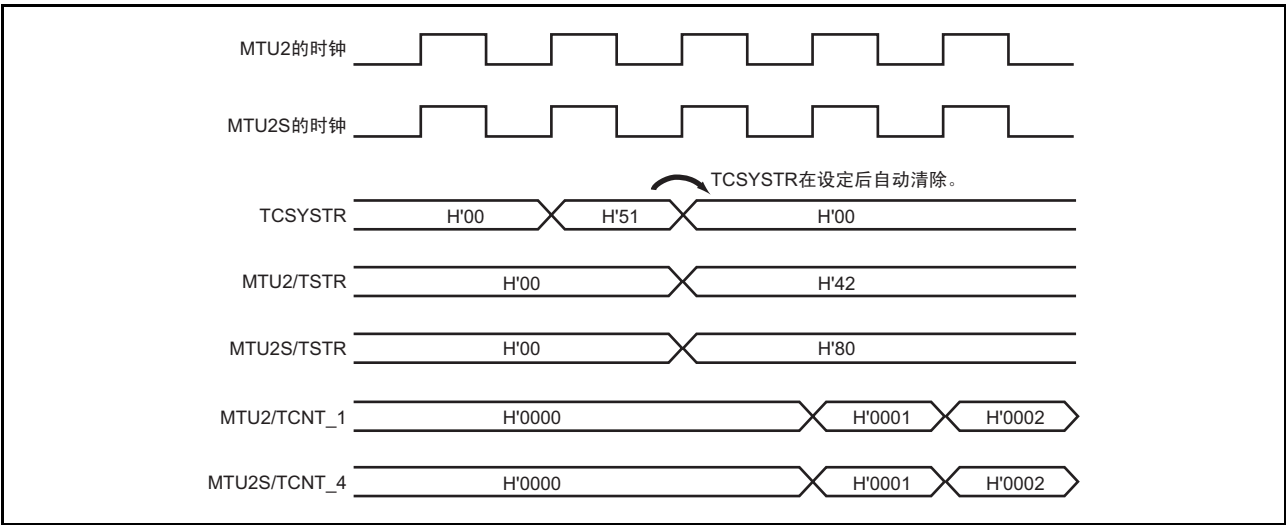


图 10.84 （1）计数器同步开始的运行例子（MTU2 和 MTU2S 的时钟频率比为 1:1）

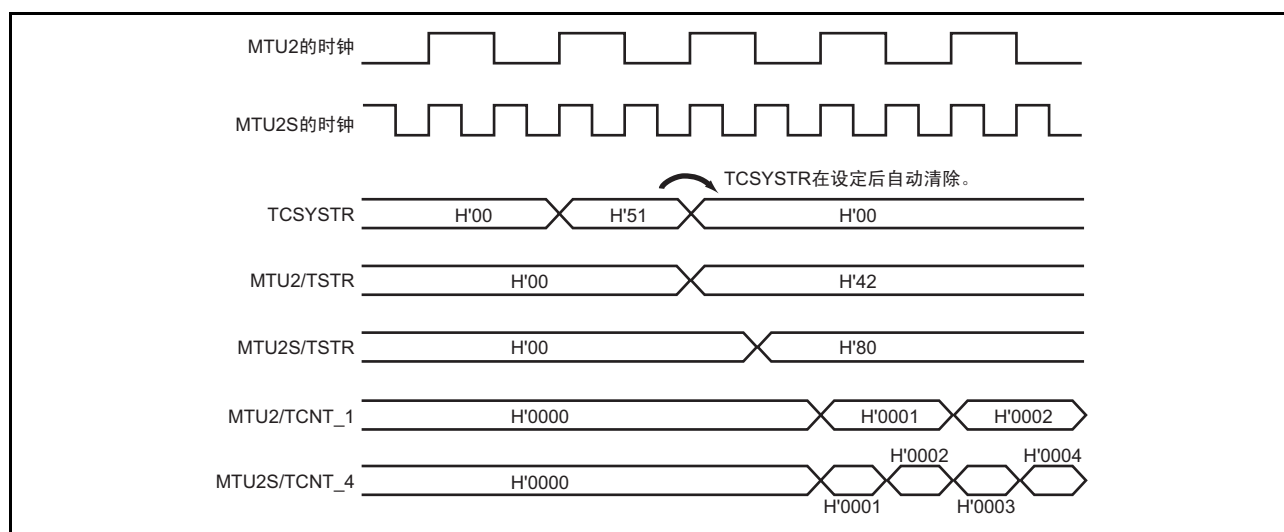


图 10.84 (2) 计数器同步开始的运行例子 (MTU2 和 MTU2S 的时钟频率比为 1:2)

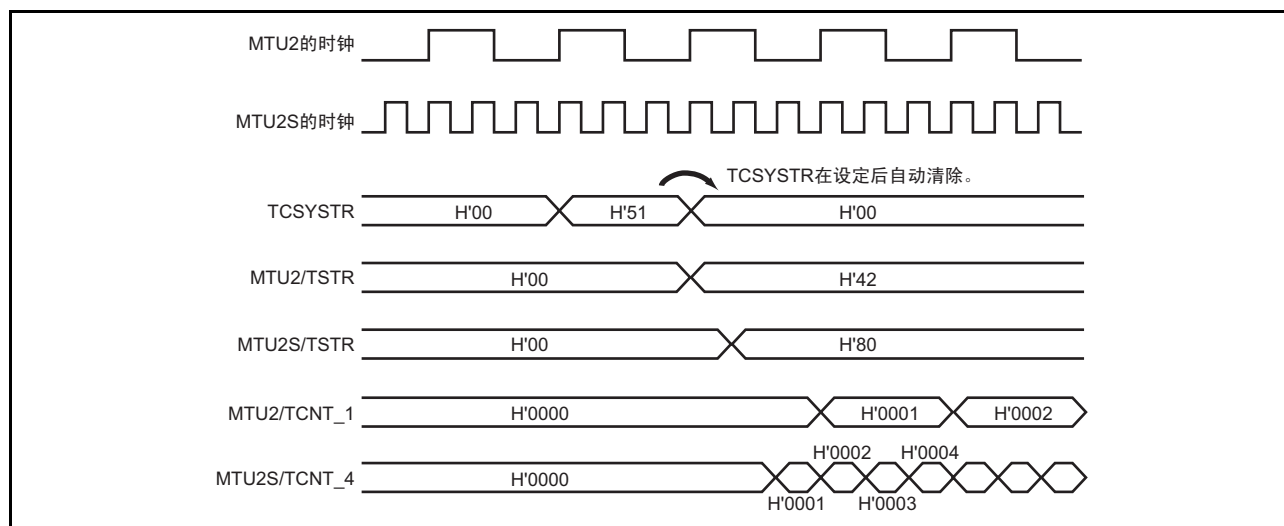


图 10.84 (3) 计数器同步开始的运行例子 (MTU2 和 MTU2S 的时钟频率比为 1:3)

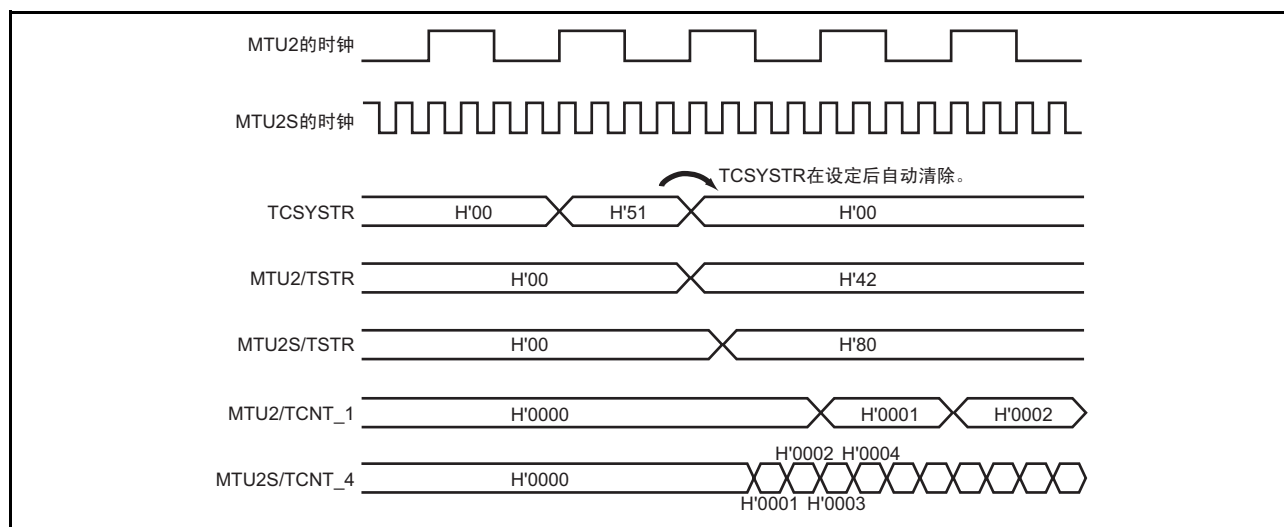


图 10.84 (4) 计数器同步开始的运行例子 (MTU2 和 MTU2S 的时钟频率比为 1:4)

(2) 利用 MTU2 标志的置位源进行 MTU2S 计数器清除 (MTU2-MTU2S 计数器的同步清除)

通过设定 TSYCR_3 寄存器, MTU2S 能利用 MTU2 的 TSR_0 ~ TSR_2 标志的置位源进行计数器的清除。

(a) 利用 MTU2 标志的置位源进行 MTU2S 计数器清除的设定步骤例子

利用 MTU2 标志的置位源进行 MTU2S 计数器清除的设定步骤例子如图 10.85 所示。

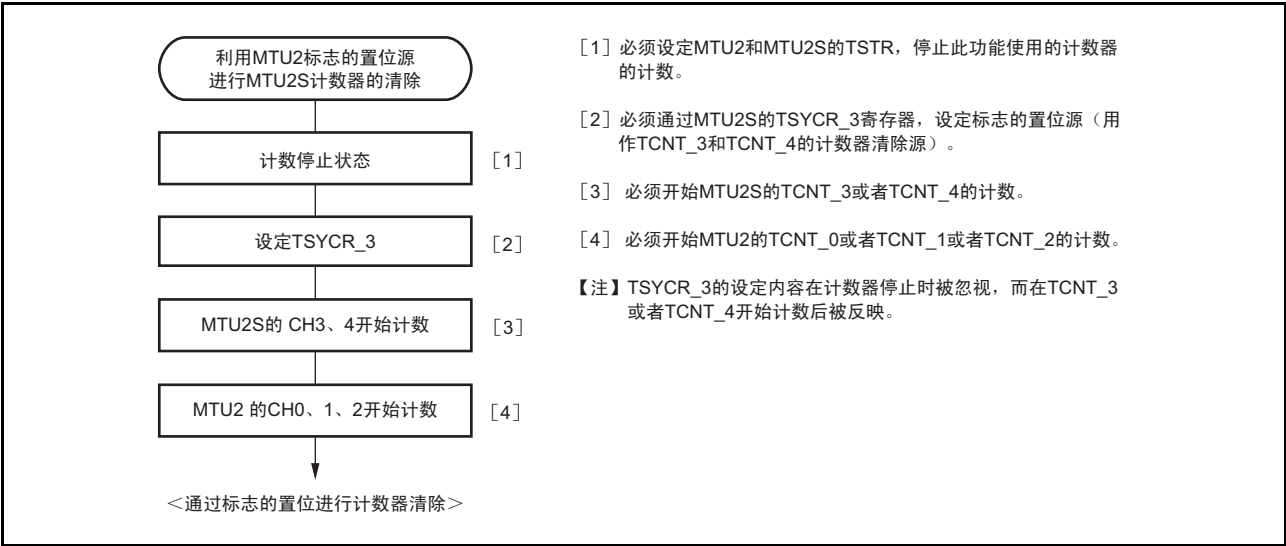


图 10.85 利用 MTU2 标志的置位源进行 MTU2S 计数器清除的设定步骤例子

(b) 利用 MTU2 标志的置位源进行 MTU2S 计数器清除的运行例子

利用 MTU2 标志的置位源进行 MTU2S 计数器清除的运行例子如图 10.86 (1) 和图 10.86 (2) 所示。

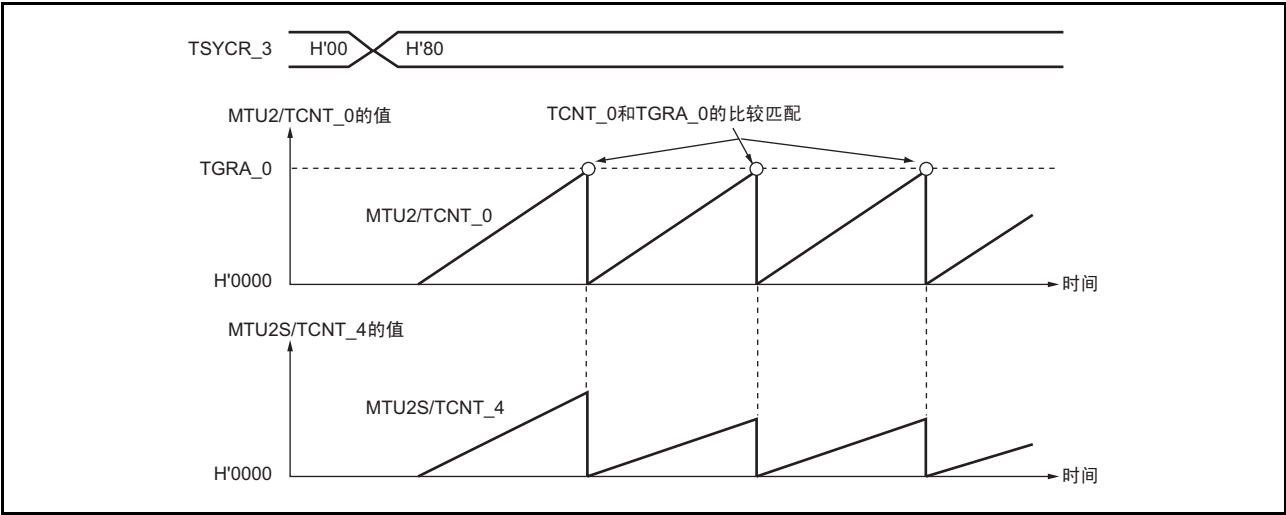


图 10.86 (1) 利用 MTU2 标志的置位源进行 MTU2S 计数器清除的运行例子 (1)

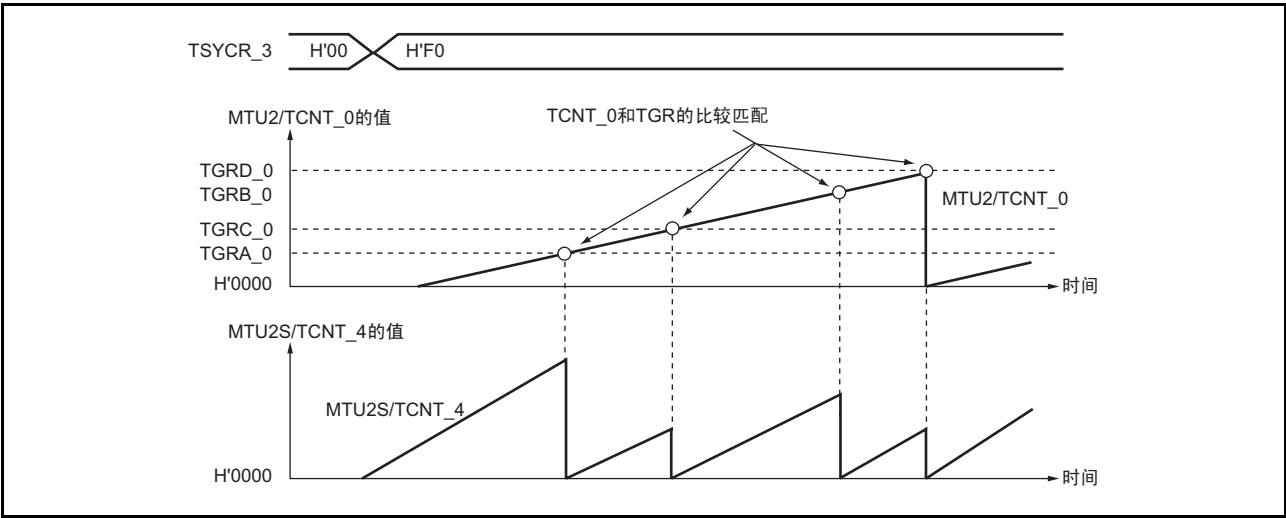


图 10.86 (2) 利用 MTU2 标志的置位源进行 MTU2S 计数器清除的运行例子 (2)

10.4.11 外部脉宽的测量功能

通道 5 最多能测量 3 个外部脉宽。

(1) 测量外部脉宽的设定步骤例子

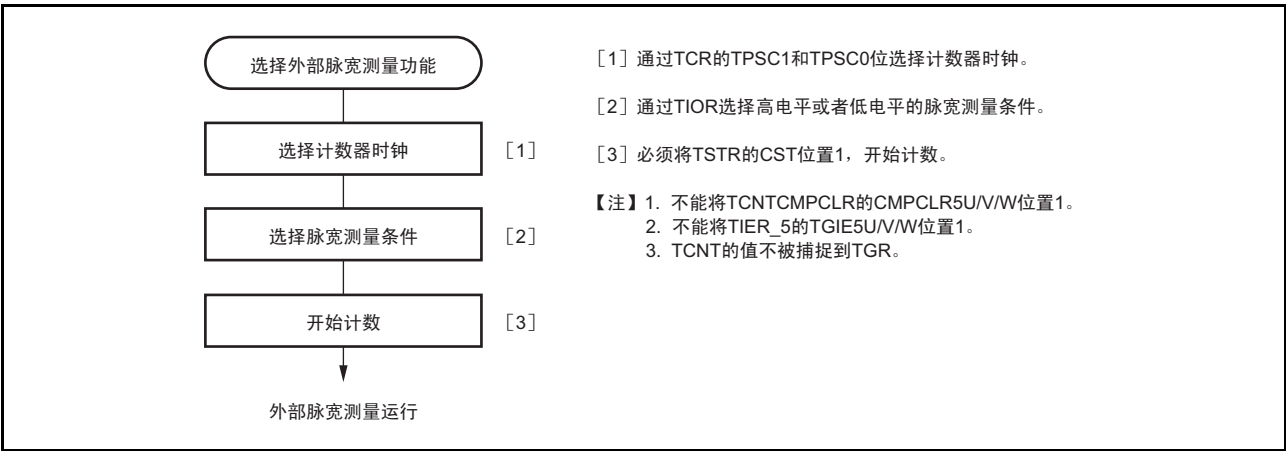


图 10.87 测量外部脉宽的设定步骤例子

(2) 测量外部脉宽的运行例子

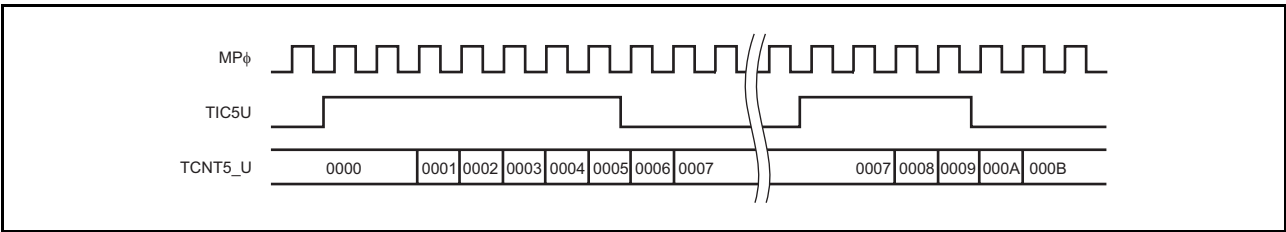


图 10.88 测量外部脉宽的运行例子 (测量高电平的脉宽)

10.4.12 空载时间的补偿功能

通过测量输出波形的延迟并将此延迟反映到占空比，能将外部脉宽测量功能用作互补 PWM 运行时 PWM 输出波形的空载时间补偿功能。

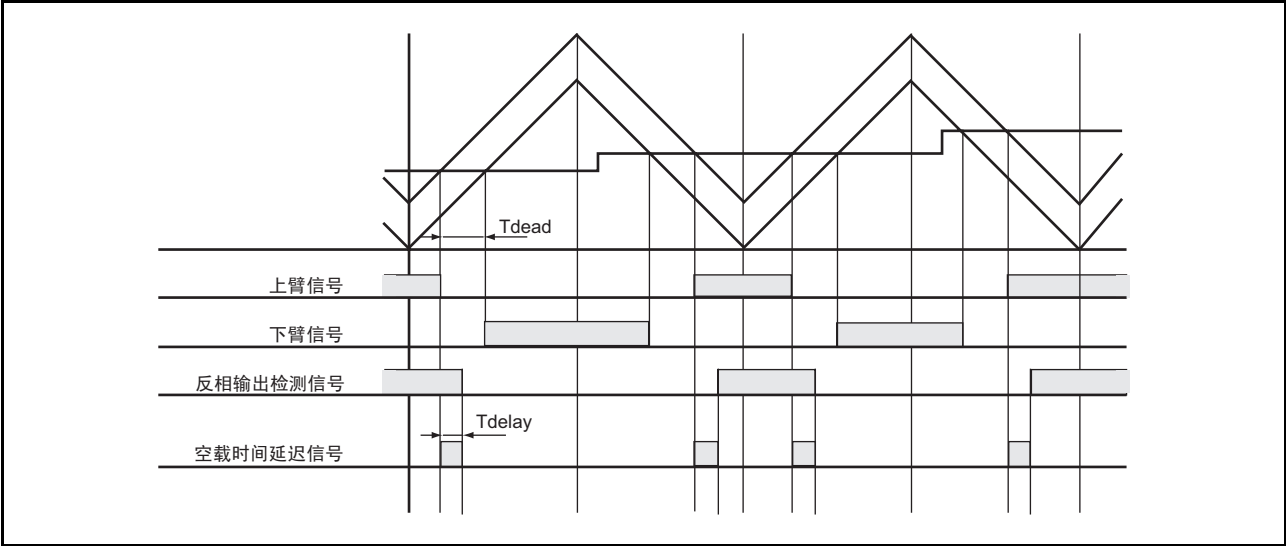


图 10.89 互补 PWM 运行时的空载时间延迟

(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 10.90 所示。

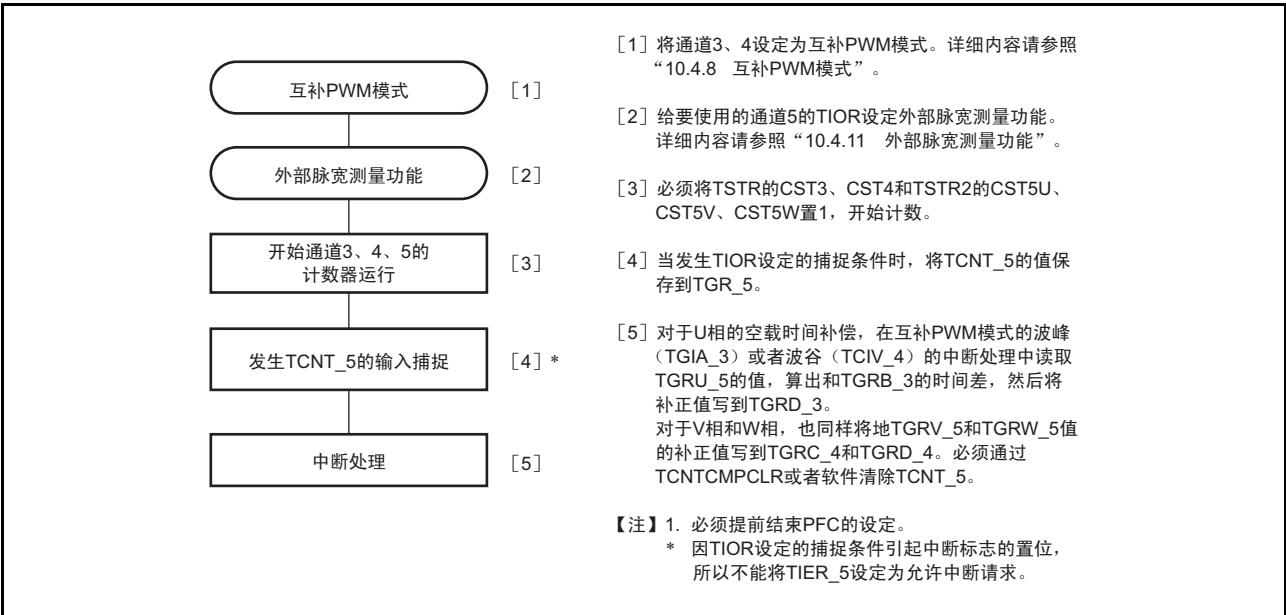


图 10.90 空载时间补偿功能的设定步骤例子

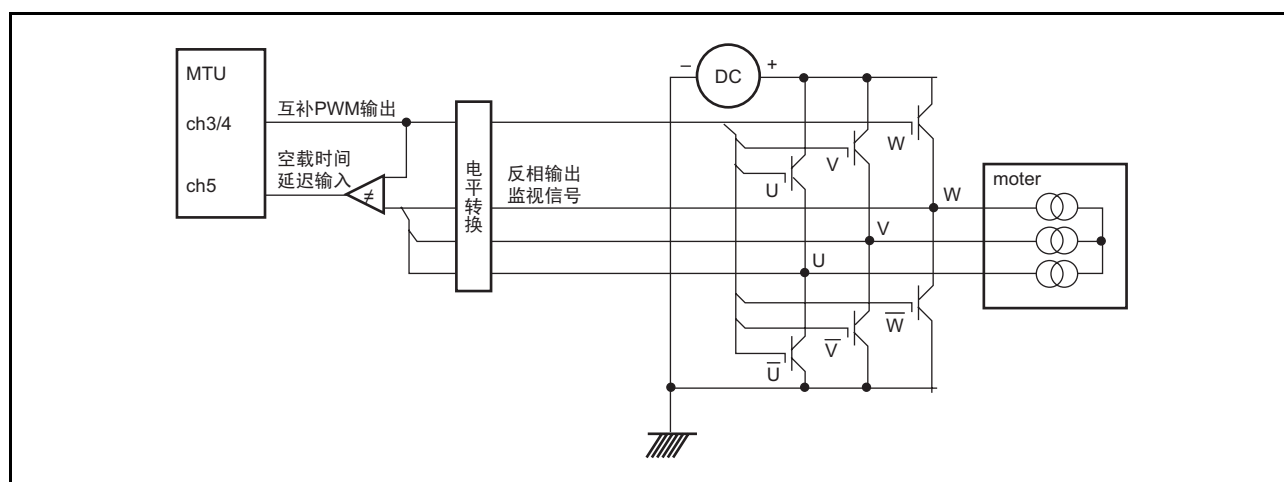


图 10.91 马达控制电路的构成例子

10.4.13 互补 PWM 在“波峰 / 波谷”的 TCNT 捕捉运行

当互补 PWM 运行时，在“波峰、波谷、波峰和波谷”将 TCNT 的值保存到 TGR。通过 TIOR 选择要保存到 TGR 的时序转换。

TCNT 用作自由运行计数器（不被清除）并在设定的“波峰 / 波谷”对 TGR 进行捕捉的运行例子如图 10.92 所示。

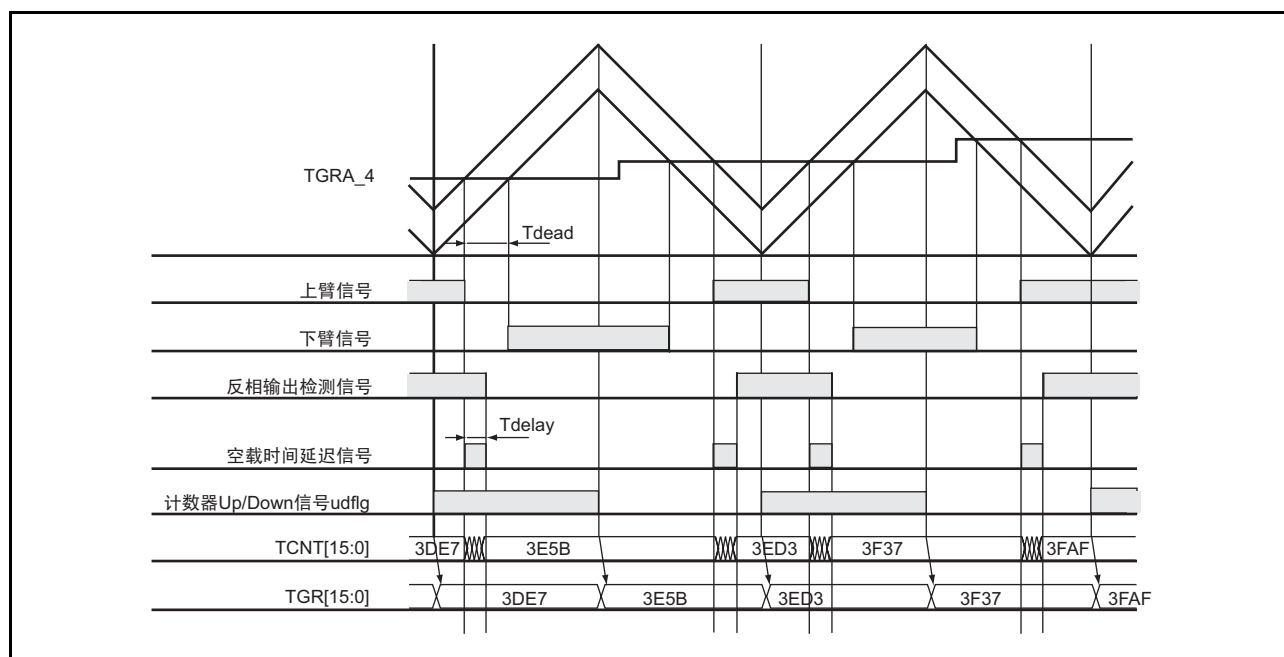


图 10.92 互补 PWM 在“波峰 / 波谷”的 TCNT 捕捉运行

10.5 中断源

10.5.1 中断源和优先级

MTU2 的中断源有 TGR 的输入捕捉 / 比较匹配、TCNT 的上溢和 TCNT 的下溢 3 种。因为各中断源具有各自专用的状态标志和允许 / 禁止位，所以能独立允许或者禁止中断请求信号的发生。

当发生中断源时，TSR 对应的状态标志就被置 1。此时，如果 TIER 对应的允许 / 禁止位已被置 1，就请求中断。通过将状态标志清 0 解除中断请求。

能通过中断控制器更改通道之间的优先级，通道内的优先级是固定的。详细内容请参照“第 6 章 中断控制器 (INTC)”。

MTU2 的中断源一览表如表 10.57 所示。

表 10.57 MTU2 中断源

通道	名称	中断源	中断标志	DTC 的启动	优先级
0	TGIA_0	TGRA_0 的输入捕捉 / 比较匹配	TGFA_0	可	高
	TGIB_0	TGRB_0 的输入捕捉 / 比较匹配	TGFB_0	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。能通过中断控制器更改通道间的优先级。

(1) 输入捕捉 / 比较匹配中断

在通过各通道的 TGR 输入捕捉 / 比较匹配将 TSR 的 TGF 标志置 1 时, 如果 TIER 的 TGIE 位已被置 1, 就请求中断。通过将 TGF 标志清 0 解除中断请求。MTU2 的通道 0 有 6 个, 通道 3、4 各有 4 个, 通道 1、2 各有 2 个, 通道 5 各有 3 个, 共 21 个输入捕捉 / 比较匹配中断。通道 0 的 TGFE_0 和 TGFF_0 标志在输入捕捉时不被置位。

(2) 上溢中断

在通过各通道的 TCNT 上溢将 TSR 的 TCFV 标志置 1 时, 如果 TIER 的 TCIEV 位已被置 1, 就请求中断。通过将 TCFV 标志清 0 解除中断请求。MTU2 的各通道有 1 个, 共 5 个上溢中断。

(3) 下溢中断

在通过各通道的 TCNT 下溢将 TSR 的 TCFU 标志置 1 时, 如果 TIER 的 TCIEU 位已被置 1, 就请求中断。通过将 TCFU 标志清 0 解除中断请求。MTU2 的通道 1、2 各有 1 个, 共 2 个下溢中断。

10.5.2 DTC 的启动

能通过各通道的 TGR 的输入捕捉 / 比较匹配中断或者通道 4 的上溢中断, 启动 DTC。详细内容请参照“第 8 章 数据传送控制器 (DTC)”。

MTU2 有 20 个输入捕捉 / 比较匹配中断和上溢中断, 能作为 DTC 的启动源。其中, 通道 0、3 各有 4 个、通道 1、2 各有 2 个、通道 4 有 5 个、通道 5 有 3 个。

10.5.3 A/D 转换器的启动

MTU2 能通过以下 3 种方法启动 A/D 转换器。

各中断源和 A/D 转换开始请求的对应如表 10.58 所示。

(1) TGRA 的输入捕捉 / 比较匹配和在互补 PWM 模式中的 TCNT_4 波谷启动 A/D 转换器

能通过各通道的 TGRA 输入捕捉 / 比较匹配启动 A/D 转换器。另外, 在将 TIER_4 的 TTGE2 位置 1 的状态下, 如果互补 PWM 运行, 即使在 TCNT_4 为波谷 (TCNT_4=H'0000) 时也能启动 A/D 转换器。

在以下所示的条件下, 对 A/D 转换器产生 A/D 转换的开始请求 TRGAN。

- 在通过各通道的 TGRA 输入捕捉 / 比较匹配将 TSR 的 TGFA 标志置 1 时, TIER 的 TTGE 位已被置 1。
- 在将 TIER_4 的 TTGE2 位置 1 的状态下, 当互补 PWM 运行并且 TCNT_4 为波谷 (TCNT_4=H'0000) 时, 如果已在 A/D 转换器侧选择 MTU2 的转换开始触发 TRGAN, 就开始 A/D 转换。

(2) 通过 TCNT_0 和 TGRE_0 的比较匹配启动 A/D 转换器

能通过通道 0 的 TCNT_0 和 TGRE_0 比较匹配, 产生 A/D 转换的开始请求 TRG0N, 启动 A/D 转换器。

在通过通道 0 的 TCNT_0 和 TGRE_0 比较匹配将 TSR2_0 的 TGFE 标志置 1 时, 如果 TIER2_0 的 TTGE2 位已被置 1, 就对 A/D 转换器产生 A/D 转换的开始请求 TRG0N。此时, 如果已在 A/D 转换器侧选择 MTU2 的转换开始触发 TRG0N, 就开始 A/D 转换。

(3) 通过 A/D 转换开始请求的延迟功能启动 A/D 转换器

如果将 A/D 转换开始请求的控制寄存器 (TADCR) 的 TAD4AE 位和 TAD4BE 位置 1, 就能在 TADCORA、TADCORB 和 TCNT_4 的相同时产生 TRG4AN 和 TRG4BN, 启动 A/D 转换器。详细内容请参照“10.4.9 A/D 转换开始请求的延迟功能”。

当发生 TRG4AN 时, 如果已在 A/D 转换器侧选择 MTU2 的转换开始触发 TRG4AN, 就开始 A/D 转换; 当发生 TRG4BN 时, 如果已在 A/D 转换器侧选择 MTU2 的转换开始触发 TRG4BN, 就开始 A/D 转换。

表 10.58 各中断源和 A/D 转换开始请求的对应

对象	中断源	A/D 转换的开始请求
TGRA_0 和 TCNT_0	输入捕捉 / 比较匹配	TRGAN
TGRA_1 和 TCNT_1		
TGRA_2 和 TCNT_2		
TGRA_3 和 TCNT_3		
TGRA_4 和 TCNT_4		
TCNT_4	互补 PWM 模式的 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 10.93 和图 10.94、外部时钟运行（正常模式和相位计数模式）时的 TCNT 计数时序如图 10.95 和图 10.96 所示。

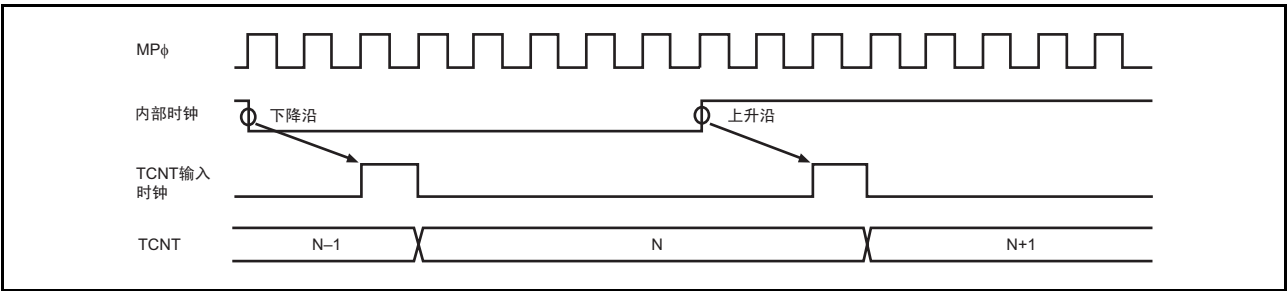


图 10.93 内部时钟运行时的计数时序（通道 0 ~ 4）

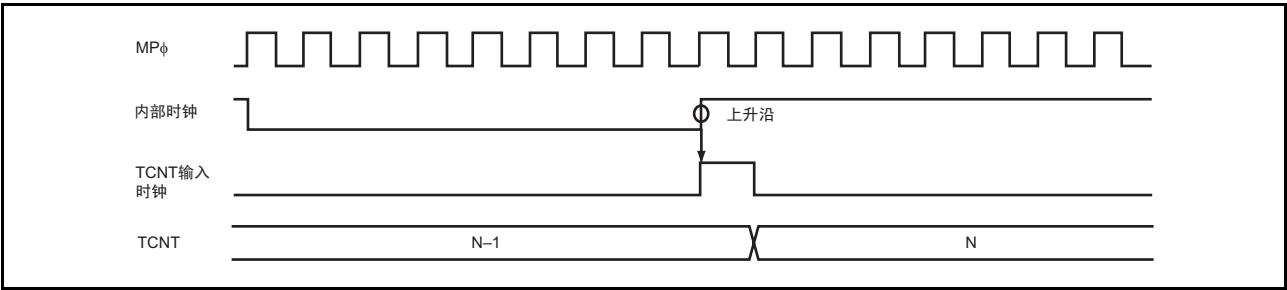


图 10.94 内部时钟运行时的计数时序（通道 5）

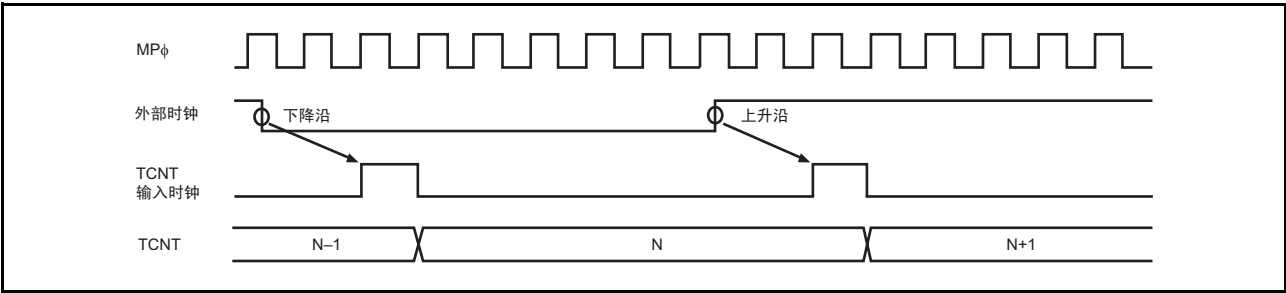


图 10.95 外部时钟运行时的计数时序 (通道 0 ~ 4)

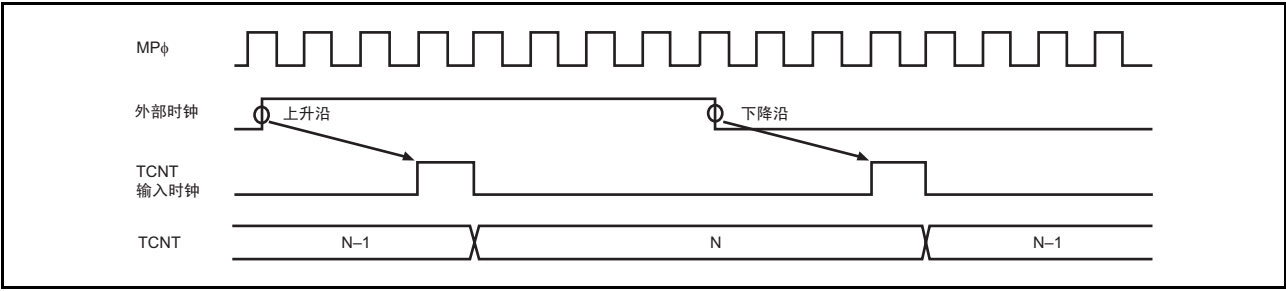


图 10.96 外部时钟运行时的计数时序 (相位计数模式)

(2) 输出比较的输出时序

在 TCNT 和 TGR 相同的最后状态 (更新 TCNT 相同后的计数值时) 产生比较匹配信号。在产生比较匹配信号时, TIOR 设定的输出值被输出到输出比较的输出引脚 (TIOC 引脚)。在 TCNT 和 TGR 相同后到产生 TCNT 输入时钟为止, 不产生比较匹配信号。

输出比较的输出时序 (正常模式和 PWM 模式) 如图 10.97、输出比较的输出时序 (互补 PWM 模式和复位同步 PWM 模式) 如图 10.98 所示。

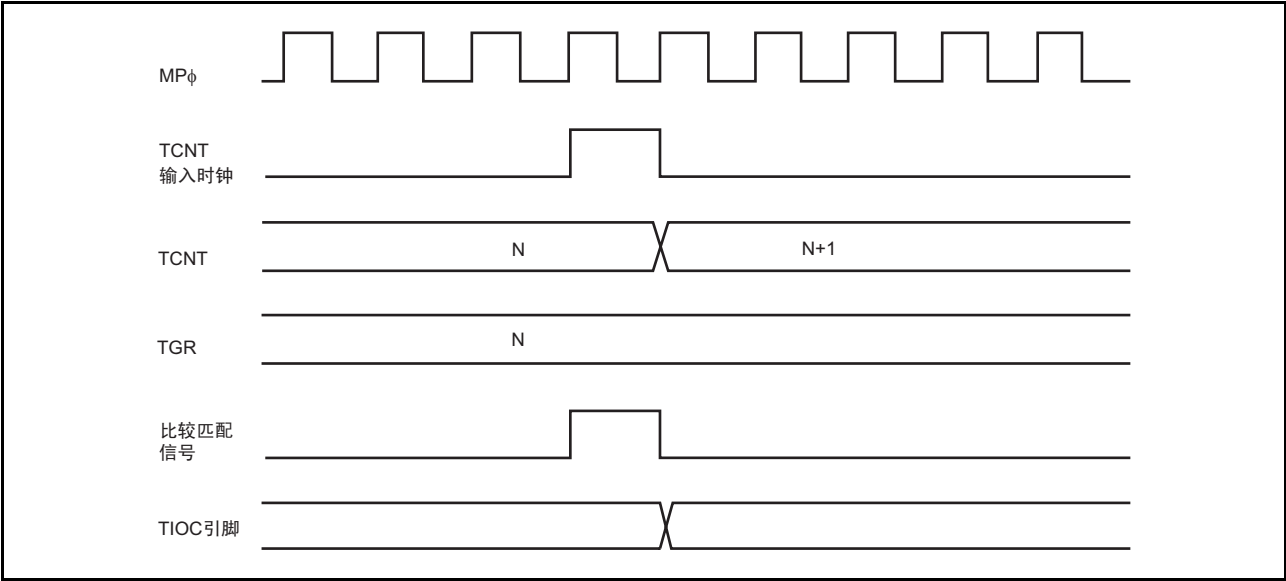


图 10.97 输出比较的输出时序 (正常模式和 PWM 模式)

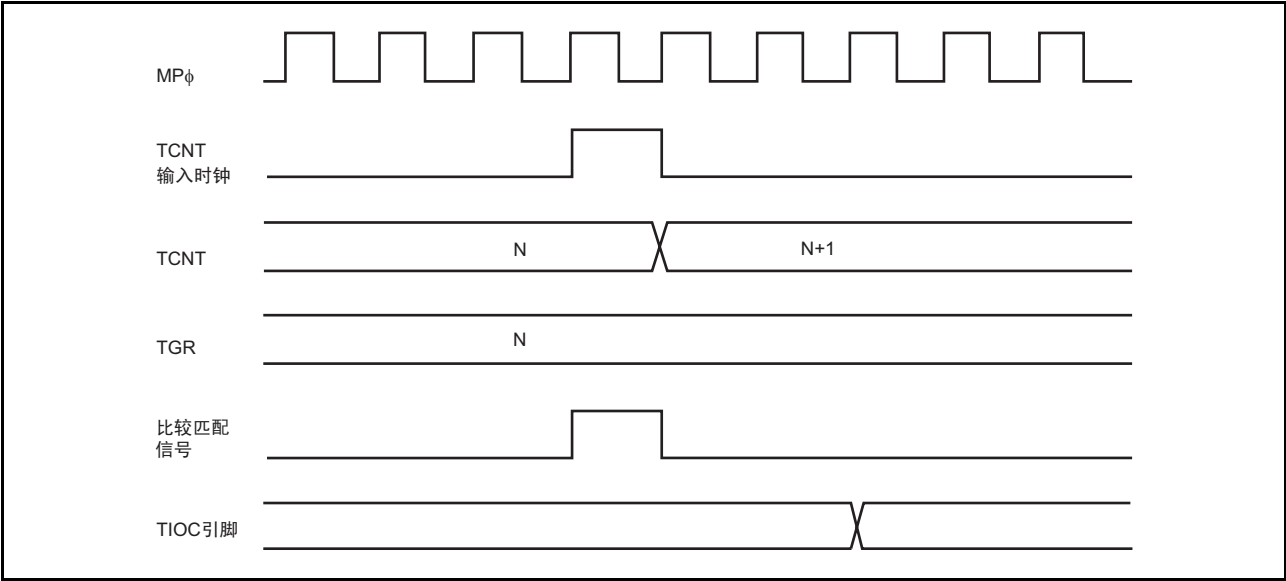


图 10.98 输出比较的输出时序（互补 PWM 模式和复位同步 PWM 模式）

(3) 输入捕捉的信号时序

输入捕捉的输入信号时序如图 10.99 所示。

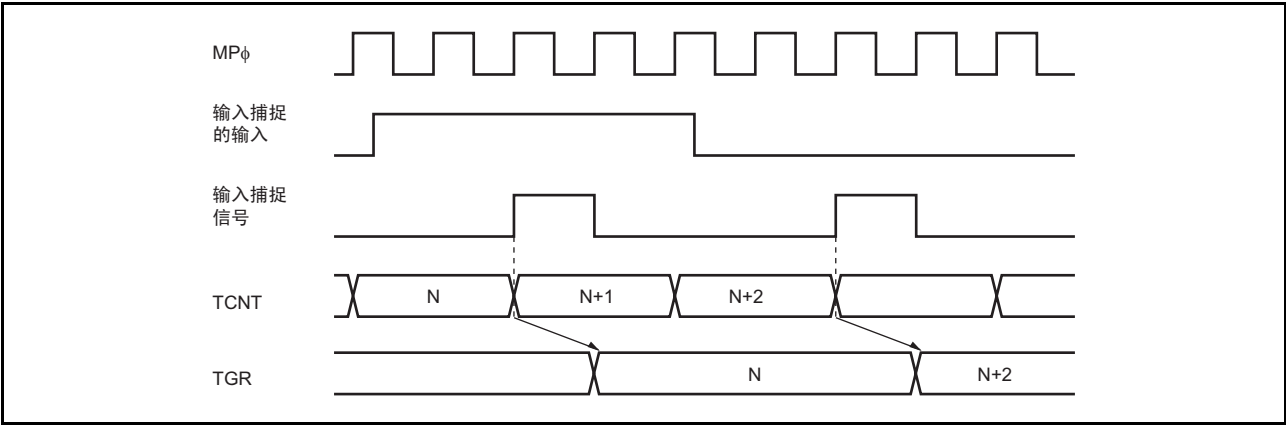


图 10.99 输入捕捉的输入信号时序

(4) 通过比较匹配 / 输入捕捉进行计数器清除的时序

指定通过比较匹配进行计数器清除时的时序如图 10.100 和图 10.101 所示。

指定通过输入捕捉进行计数器清除时的时序如图 10.102 所示。

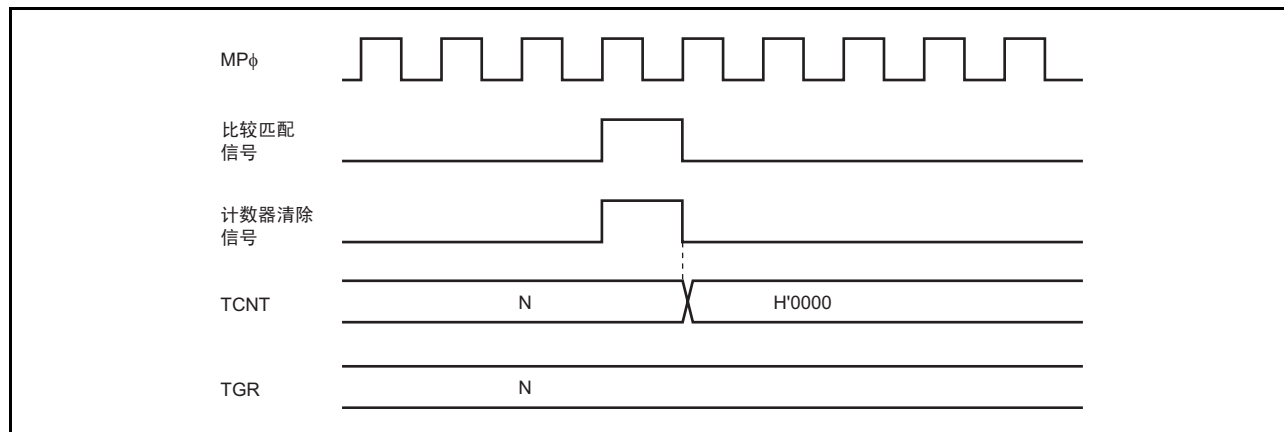


图 10.100 计数器清除的时序 (比较匹配) (通道 0 ~ 4)

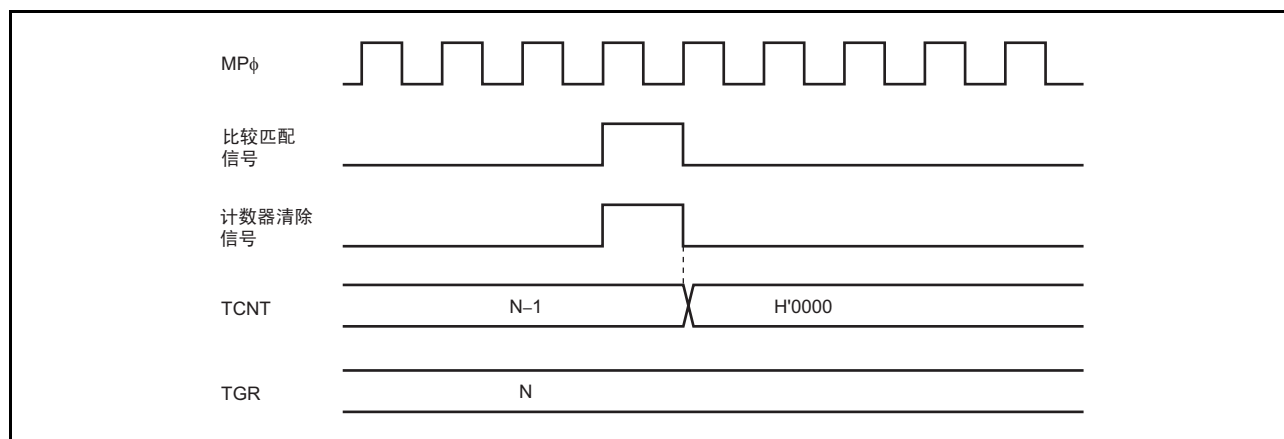


图 10.101 计数器清除的时序 (比较匹配) (通道 5)

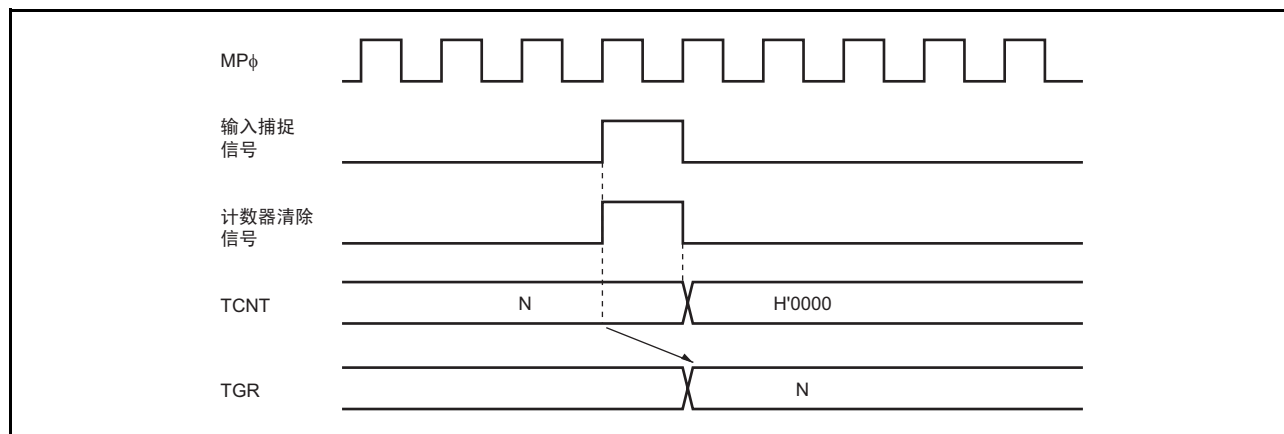


图 10.102 计数器清除的时序 (输入捕捉) (通道 0 ~ 5)

(5) 缓冲运行的时序

缓冲运行的时序如图 10.103 ~ 图 10.105 所示。

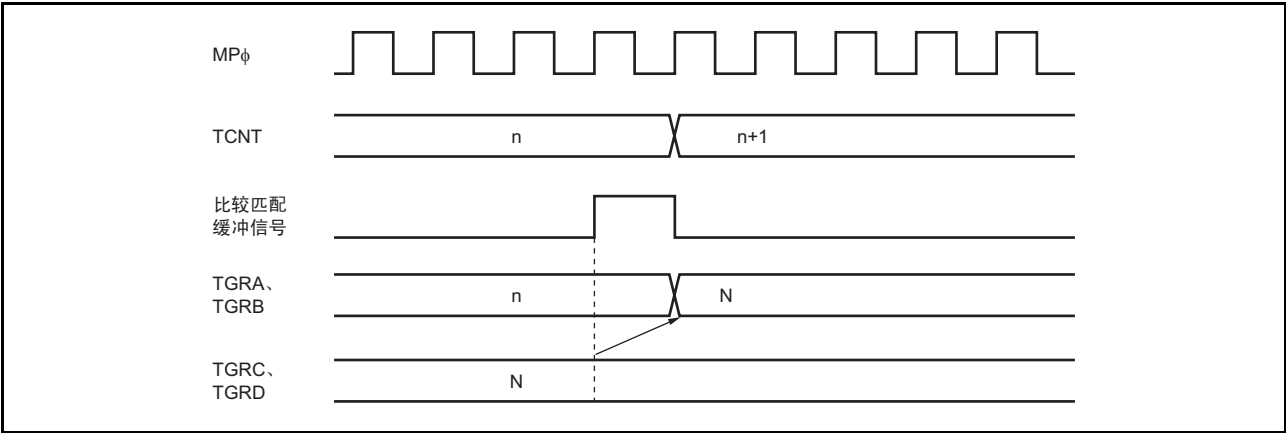


图 10.103 缓冲运行的时序 (比较匹配)

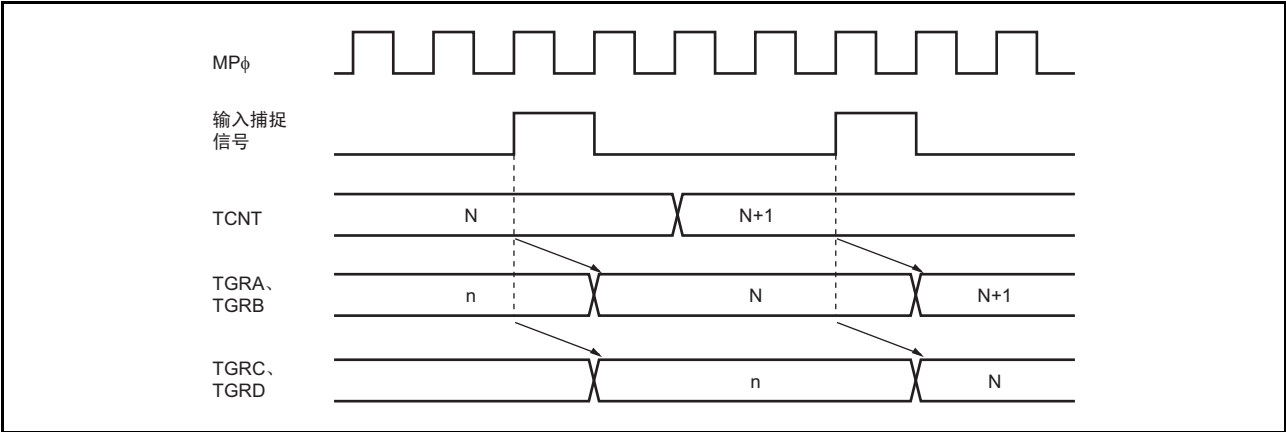


图 10.104 缓冲运行的时序 (输入捕捉)

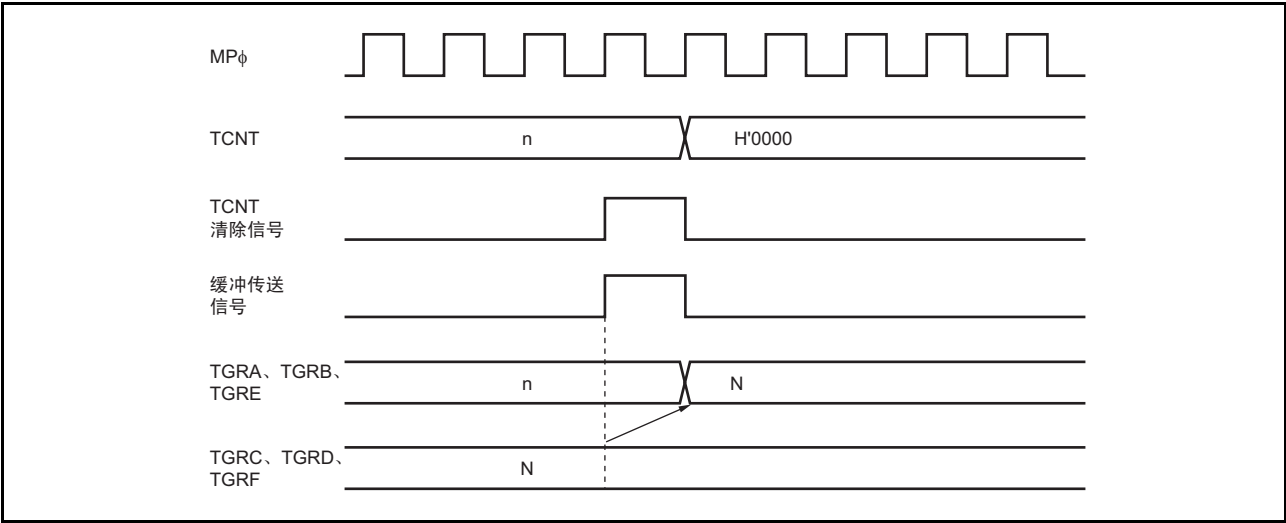


图 10.105 缓冲运行的时序 (清除 TCNT)

(6) 缓冲传送的时序 (互补 PWM 模式)

互补 PWM 模式的缓冲传送时序如图 10.106 ~ 图 10.108 所示。

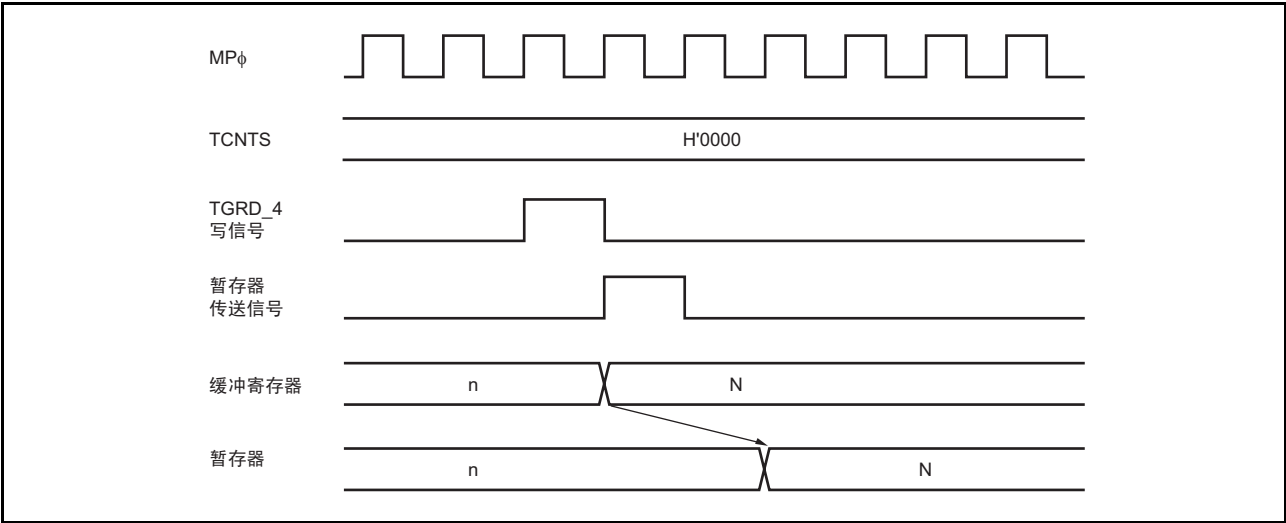


图 10.106 缓冲寄存器到暂存器的传送时序 (TCNTS 停止运行)

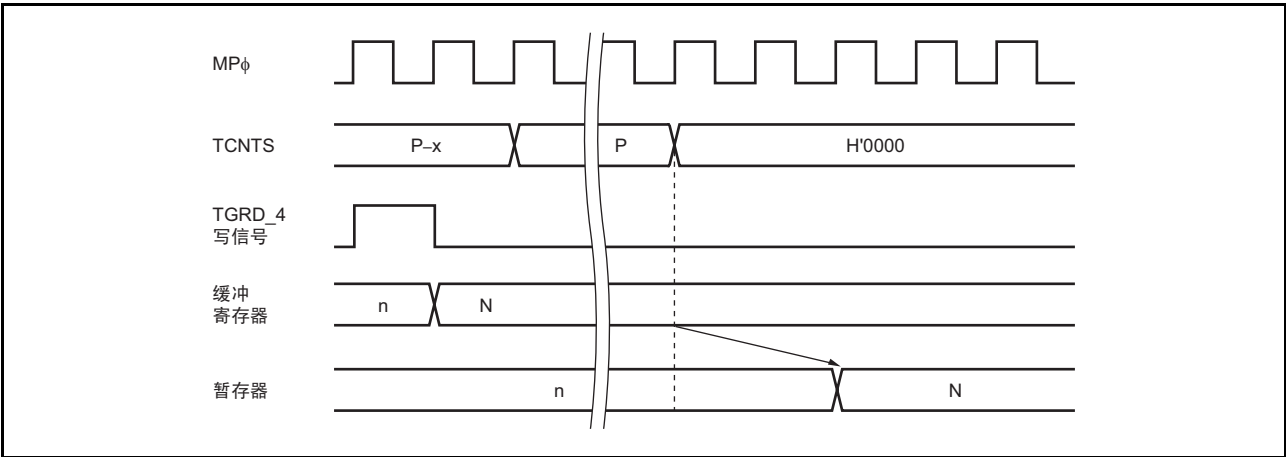


图 10.107 缓冲寄存器到暂存器的传送时序 (TCNTS 运行中)

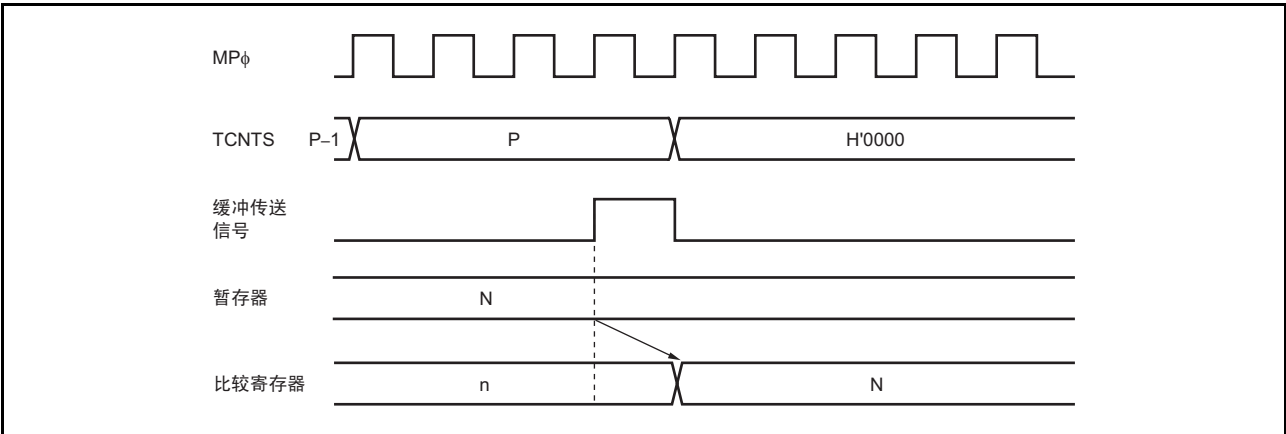


图 10.108 暂存器到比较寄存器的传送时序

10.6.2 中断信号的时序

(1) 比较匹配时的 TGF 标志的置位时序

比较匹配产生的 TSR 的 TGF 标志置位时序和 TGI 中断请求信号的时序如图 10.109 和图 10.110 所示。

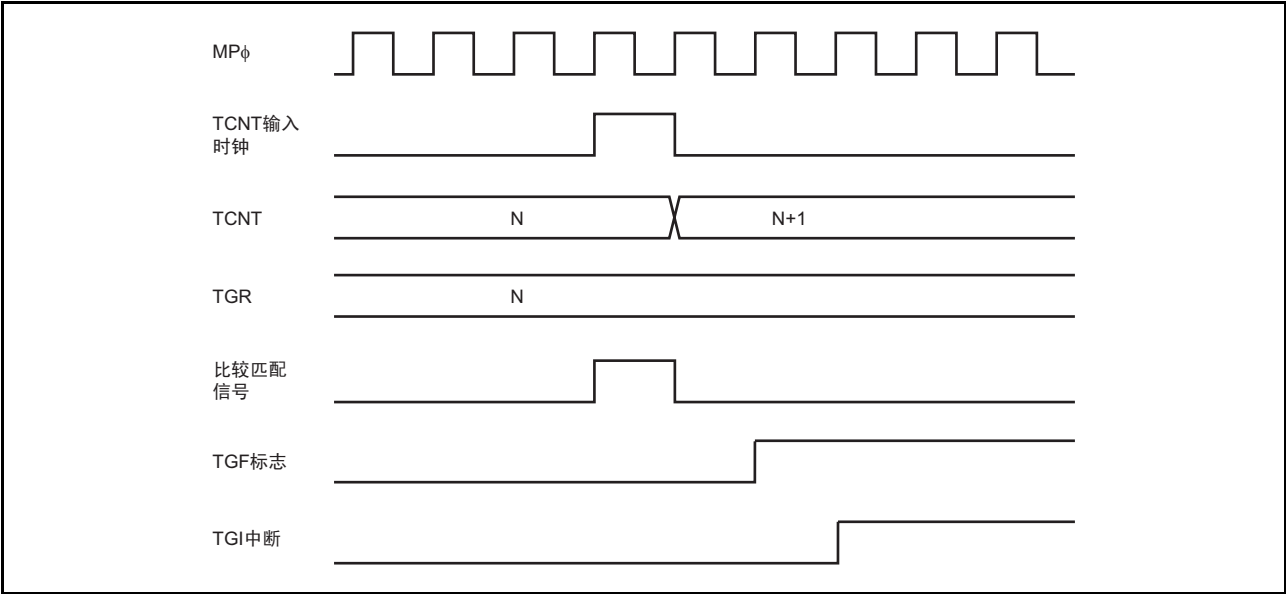


图 10.109 TGI 中断时序（比较匹配）（通道 0 ~ 4）

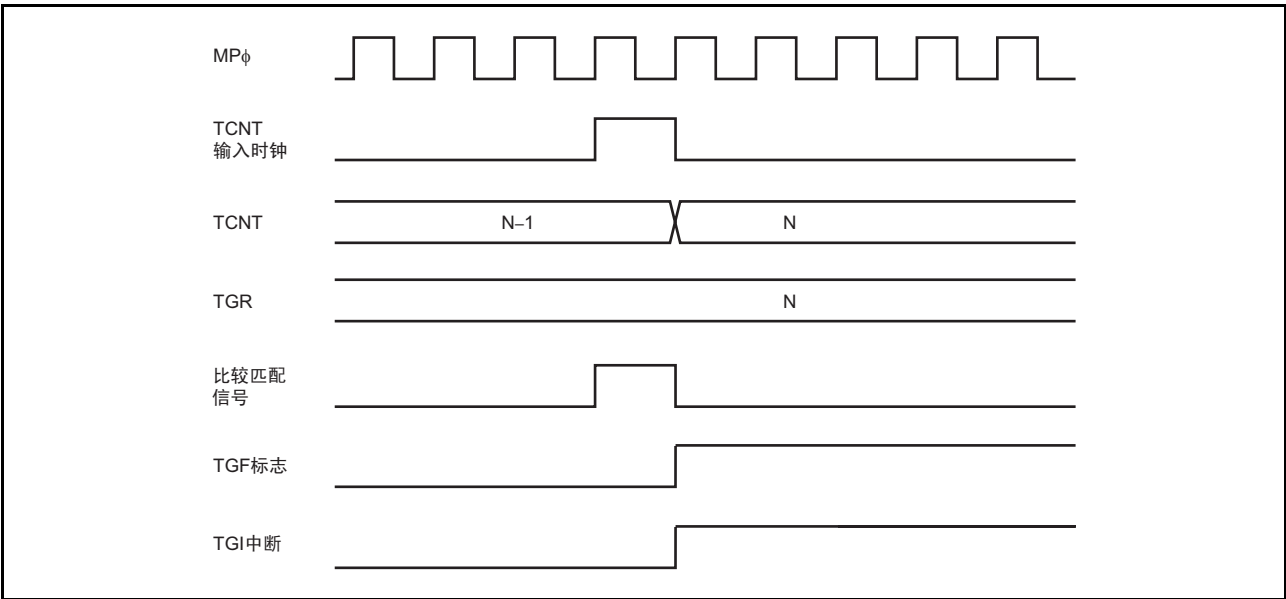


图 10.110 TGI 中断时序（比较匹配）（通道 5）

(2) 输入捕捉时的 TGF 标志的置位时序

输入捕捉产生 TSR 的 TGF 标志的置位时序和 TGI 中断请求信号的时序如图 10.111 和图 10.112 所示。

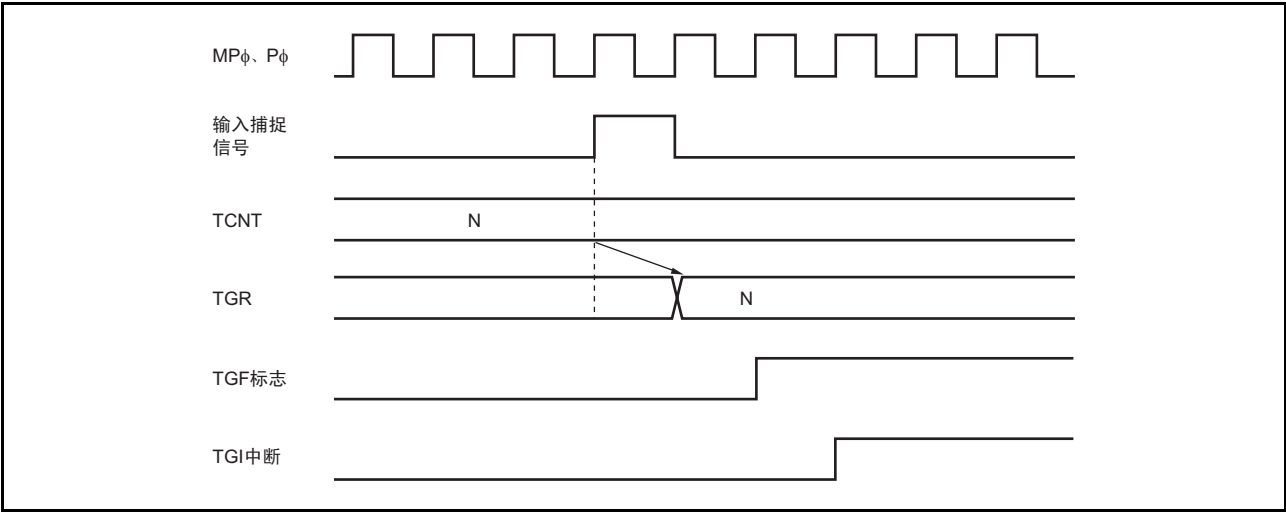


图 10.111 TGI 中断时序（输入捕捉）（通道 0 ~ 4）

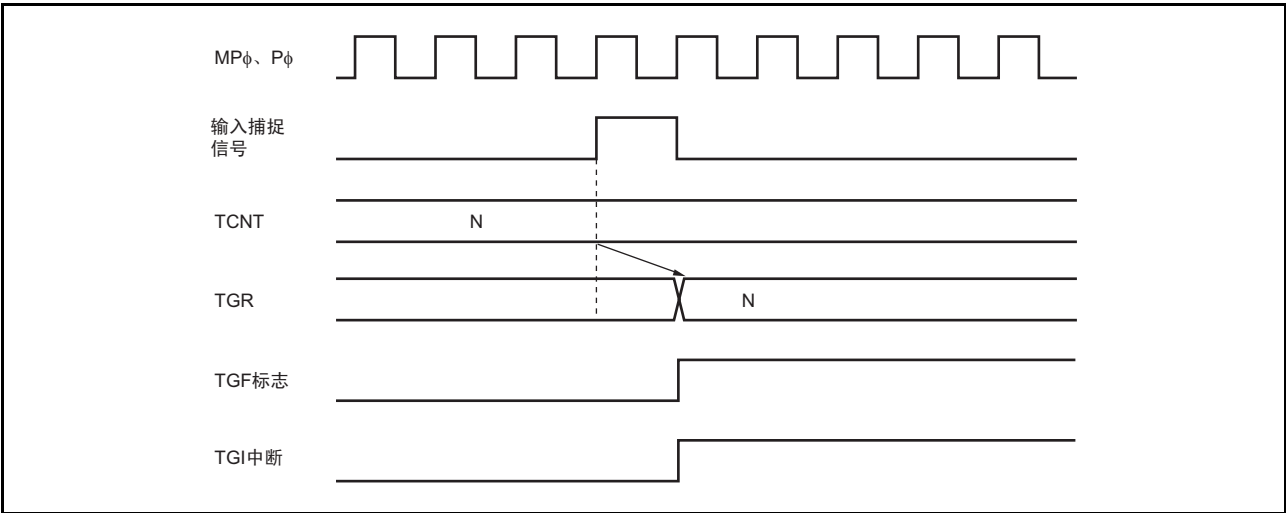


图 10.112 TGI 中断时序（输入捕捉）（通道 5）

(3) TCFV 标志 /TCFU 标志的置位时序

上溢产生的 TSR 的 TCFV 标志置位时序和 TCIV 中断请求信号的时序如图 10.113 所示。
下溢产生的 TSR 的 TCFU 标志置位时序和 TCIU 中断请求信号的时序如图 10.114 所示。

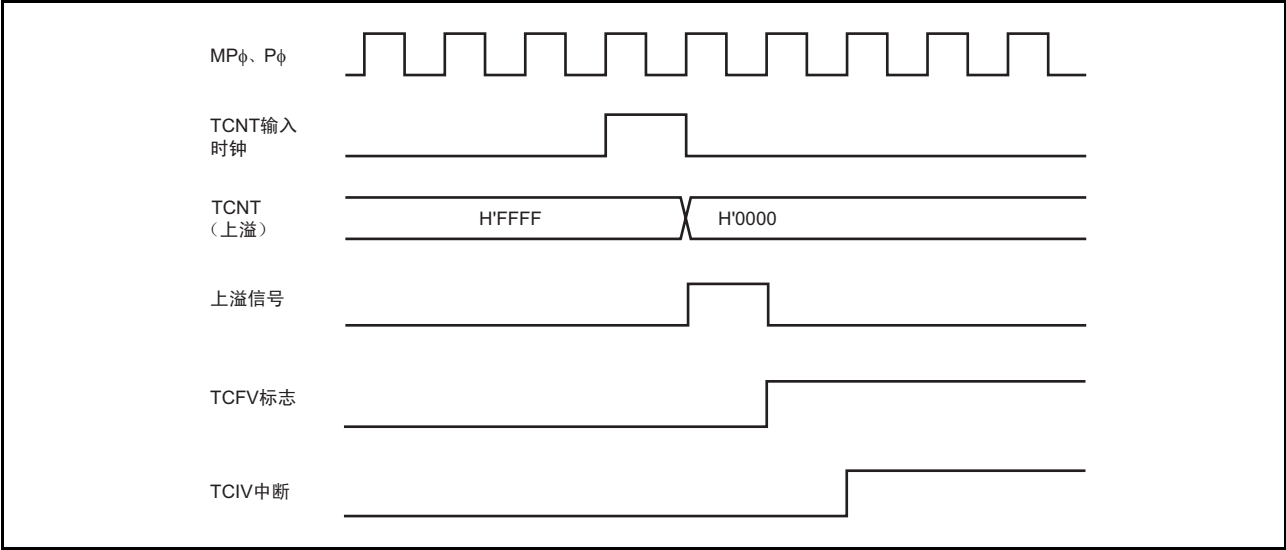


图 10.113 TCIV 中断的置位时序

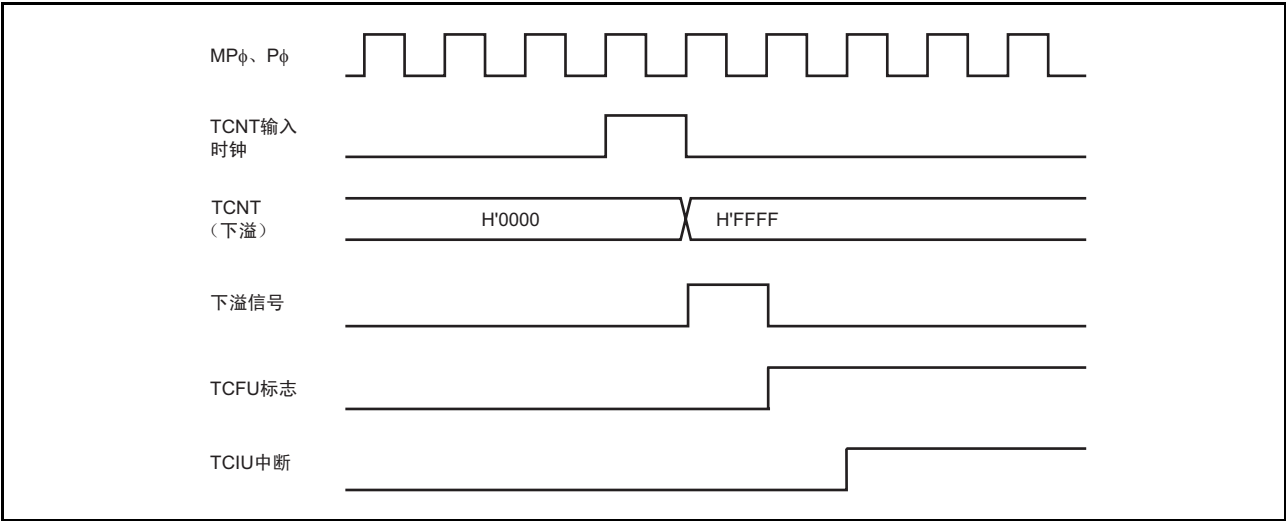


图 10.114 TCIU 中断的置位时序

(4) 状态标志的清除时序

如果 CPU 在读 1 的状态后写 0，就清除状态标志。在启动 DTC 时，能自动清除此状态标志。通过 CPU 清除状态标志的时序如图 10.115 和图 10.116、通过 DTC 清除状态标志的时序如图 10.117 ~ 图 10.118 所示。

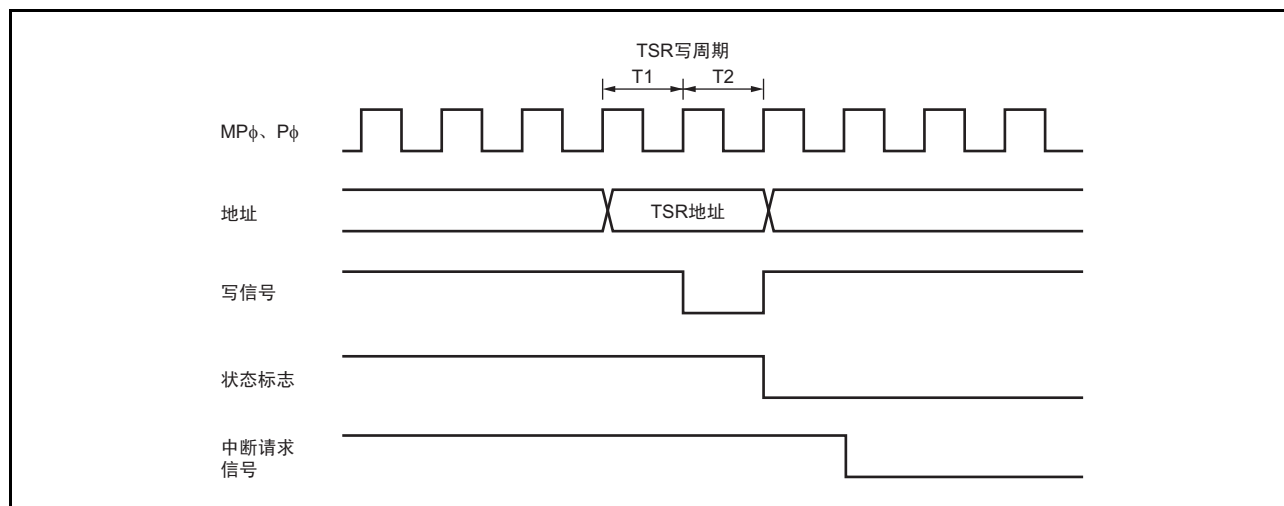


图 10.115 通过 CPU 清除状态标志的时序 (通道 0 ~ 4)

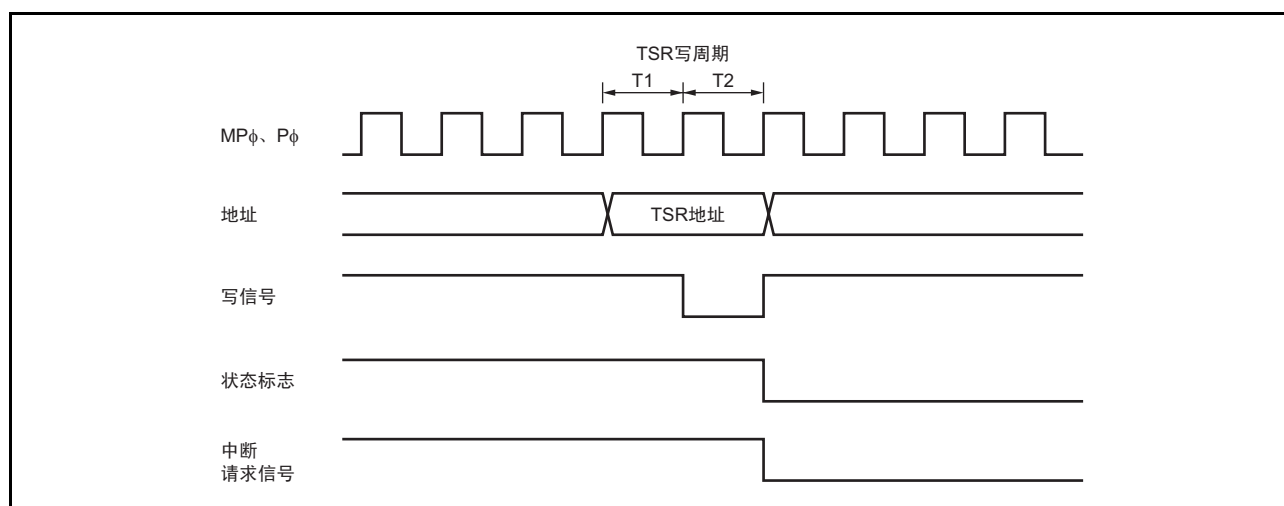


图 10.116 通过 CPU 清除状态标志的时序 (通道 5)

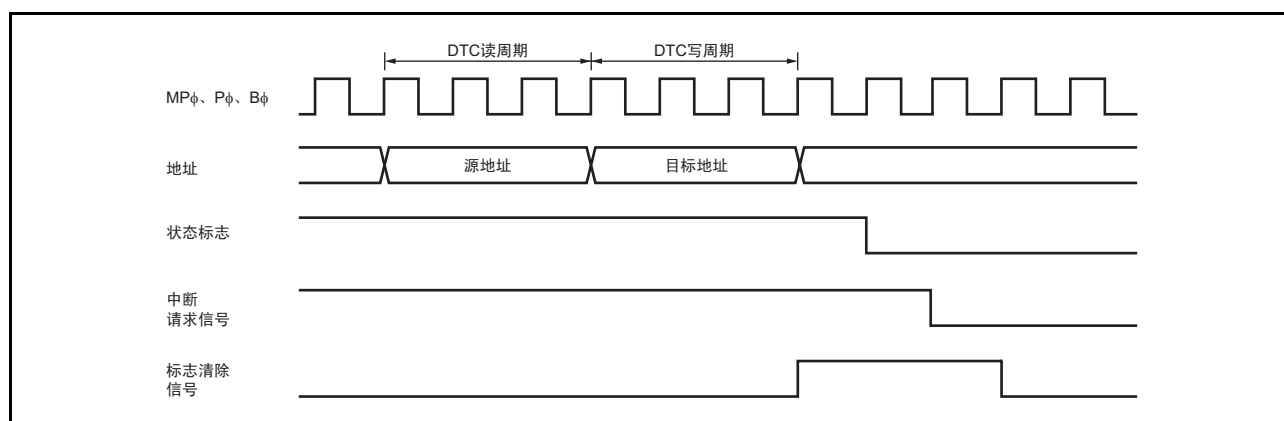


图 10.117 通过 DTC 的启动清除状态标志的时序 (通道 0 ~ 4)

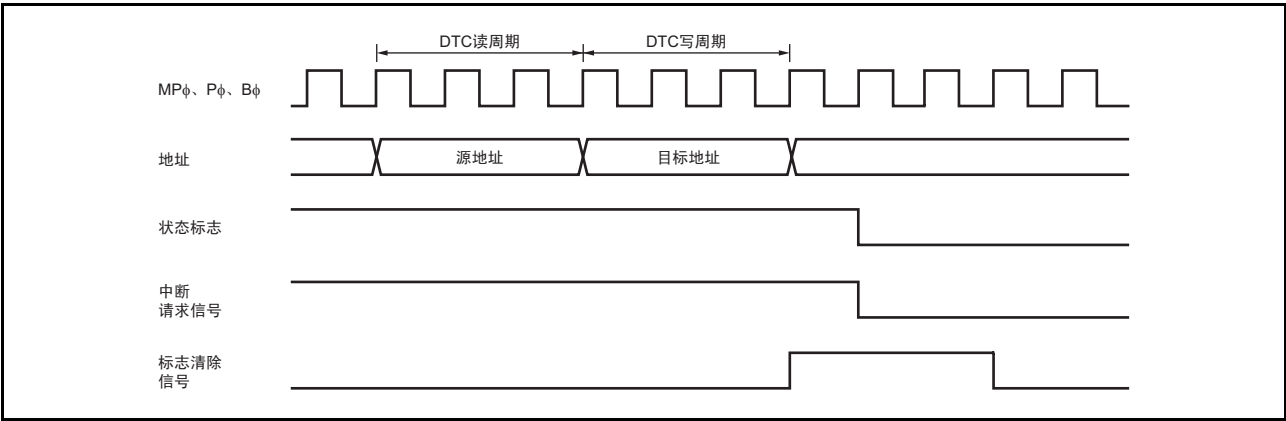


图 10.118 通过 DTC 的启动清除状态标志的时序（通道 5）

10.7 使用时的注意事项

10.7.1 模块待机模式的设定

MTU2 能通过待机控制寄存器设定允许或者禁止本模块的运行，初始值为 MTU2 停止运行。另外，能通过解除模块待机模式允许寄存器的存取。详细内容请参照“第 24 章 低功耗模式”。

10.7.2 输入时钟的限制事项

在单边沿的情况下，输入时钟的脉宽至少需要 1.5 个状态；在双边沿的情况下，至少需要 2.5 个状态。必须注意：如果小于上述脉宽就不能正常运行。

在相位计数模式的情况下，2 个输入时钟的相位差和重叠都至少需要 1.5 个状态，并且脉宽至少需要 2.5 个状态。相位计数模式的输入时钟条件如图 10.119 所示。

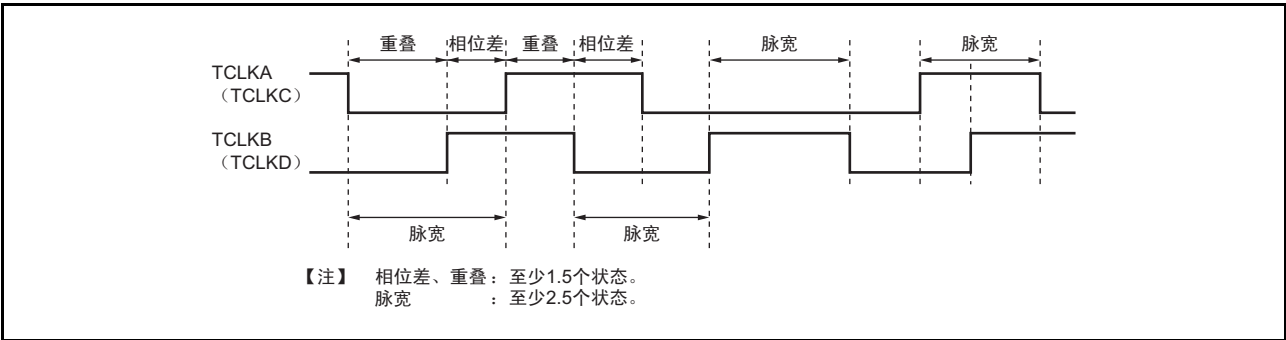


图 10.119 相位计数模式的相位差、重叠和脉宽

10.7.3 周期设定的注意事项

如果设定为通过比较匹配进行计数器清除，就在 TCNT 和 TGR 的值相同的最后状态（更新 TCNT 相同后的计数值时）清除 TCNT。因此，实际的计数器频率由以下的表达式表示：

- 通道0～4

$$f = \frac{MP\phi}{(N+1)}$$

- 通道5

$$f = \frac{MP\phi}{N}$$

f : 计数器频率

MP ϕ : MTU2 的时钟工作频率

N : TGR 的设定值

10.7.4 TCNT 的写操作和清除的竞争

如果在 TCNT 写周期中的 T2 状态产生计数器的清除信号，就不写 TCNT 而优先清除 TCNT。此时序如图 10.120 所示。

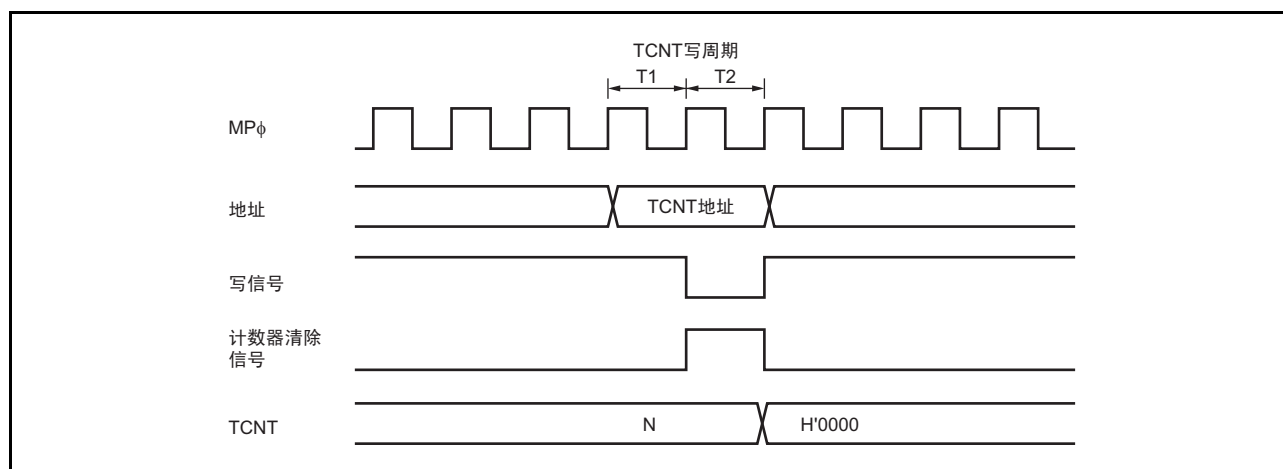


图 10.120 TCNT 的写操作和清除的竞争

10.7.5 TCNT 的写操作和递增计数的竞争

即使在 TCNT 写周期的 T2 状态产生递增计数信号，也不进行递增计数而优先写 TCNT。此时序如图 10.121 所示。

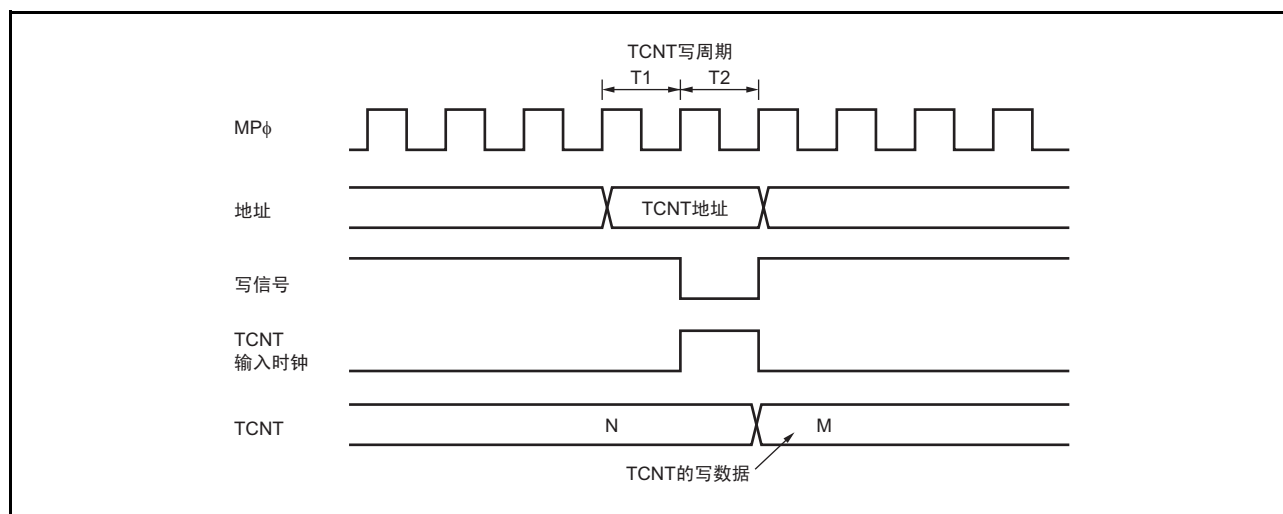


图 10.121 TCNT 的写操作和递增计数的竞争

10.7.6 TGR 的写操作和比较匹配的竞争

如果在 TGR 写周期的 T2 状态产生比较匹配信号，就写 TGR 并产生比较匹配信号。
此时序如图 10.122 所示。

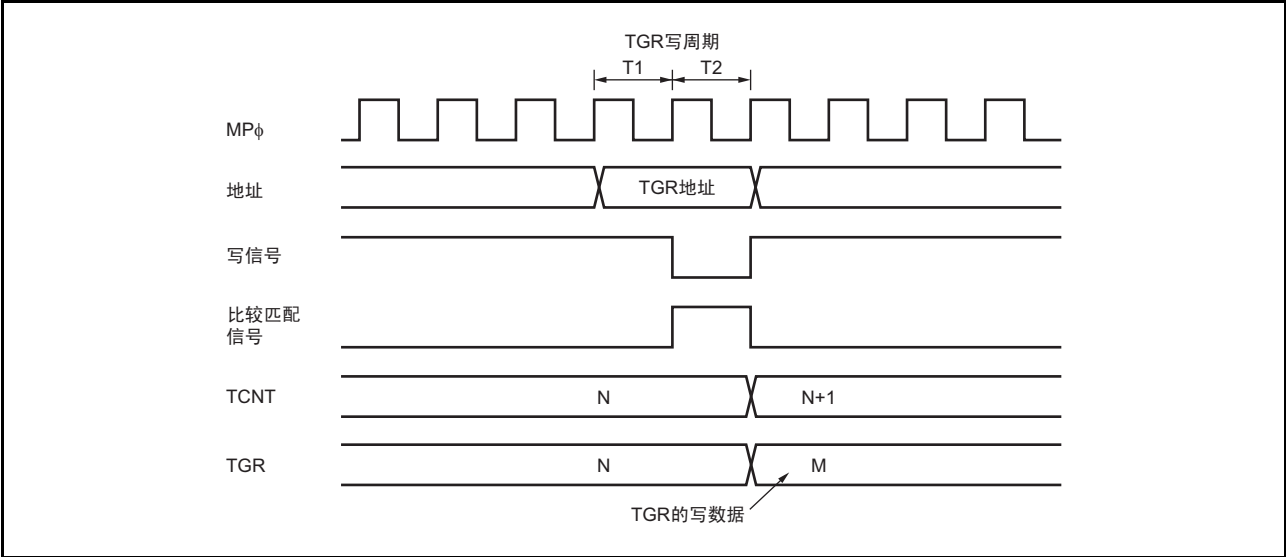


图 10.122 TGR 的写操作和比较匹配的竞争

10.7.7 缓冲寄存器的写操作和比较匹配的竞争

如果在 TGR 写入周期的 T2 状态产生比较匹配信号，通过缓冲运行传送到 TGR 的数据就为写操作之前的数据。
此时序如图 10.123 所示。

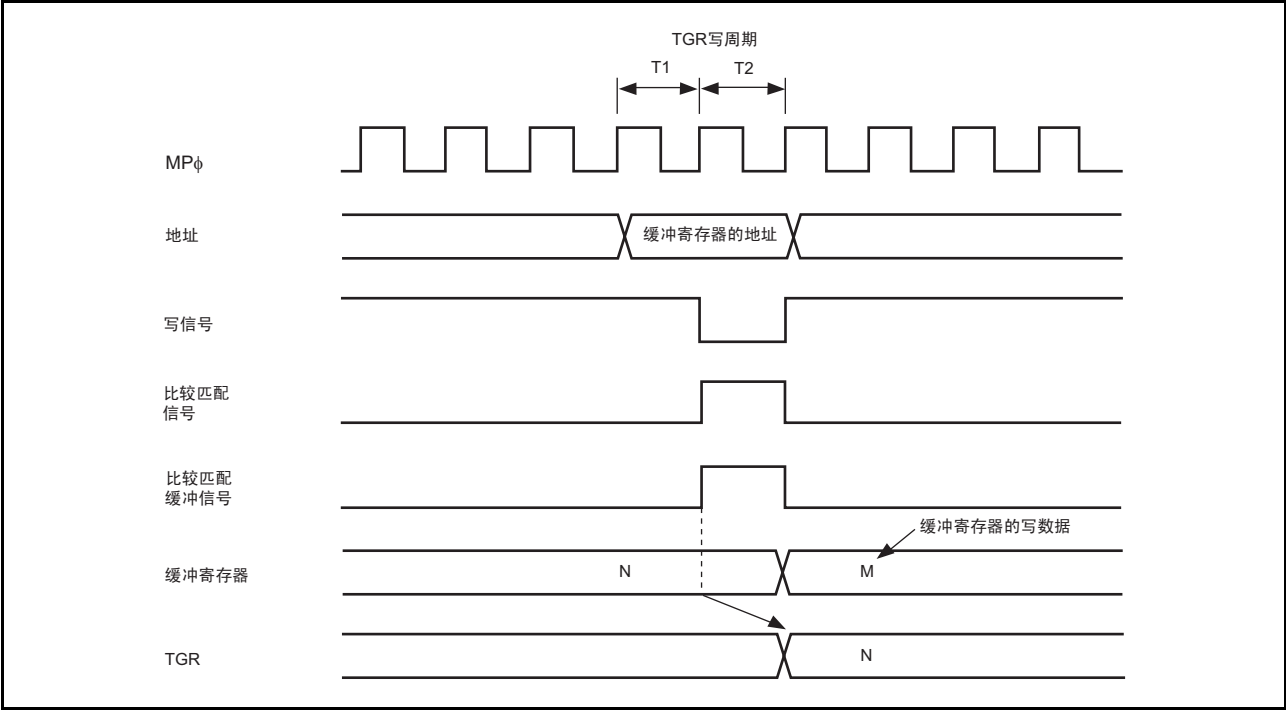


图 10.123 缓冲寄存器的写操作和比较匹配的竞争

10.7.8 缓冲寄存器的写操作和清除 TCNT 的竞争

当通过缓冲传送模式寄存器 (TBTM) 将缓冲传送时序设定为清除 TCNT 时, 如果在 TGR 写周期的 T2 状态产生 TCNT 清除信号, 通过缓冲运行传送到 TGR 的数据就为写操作之前的数据。

此时序如图 10.124 所示。

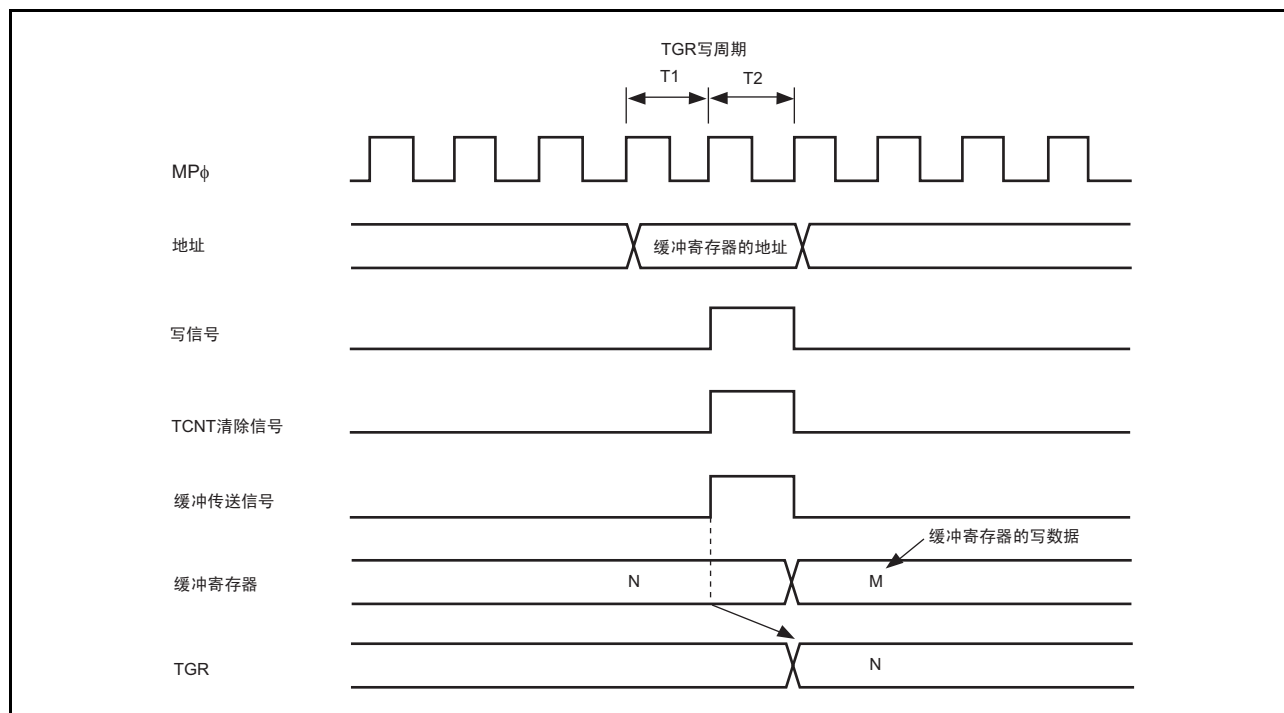


图 10.124 缓冲寄存器的写操作和清除 TCNT 的竞争

10.7.9 TGR 的读操作和输入捕捉的竞争

如果在 TGR 读周期的 T1 状态产生输入捕捉信号, 读取的数据在通道 0 ~ 4 时为输入捕捉传送前的数据, 而在通道 5 时为输入捕捉传送后的数据。

此时序如图 10.125 和图 10.126 所示。

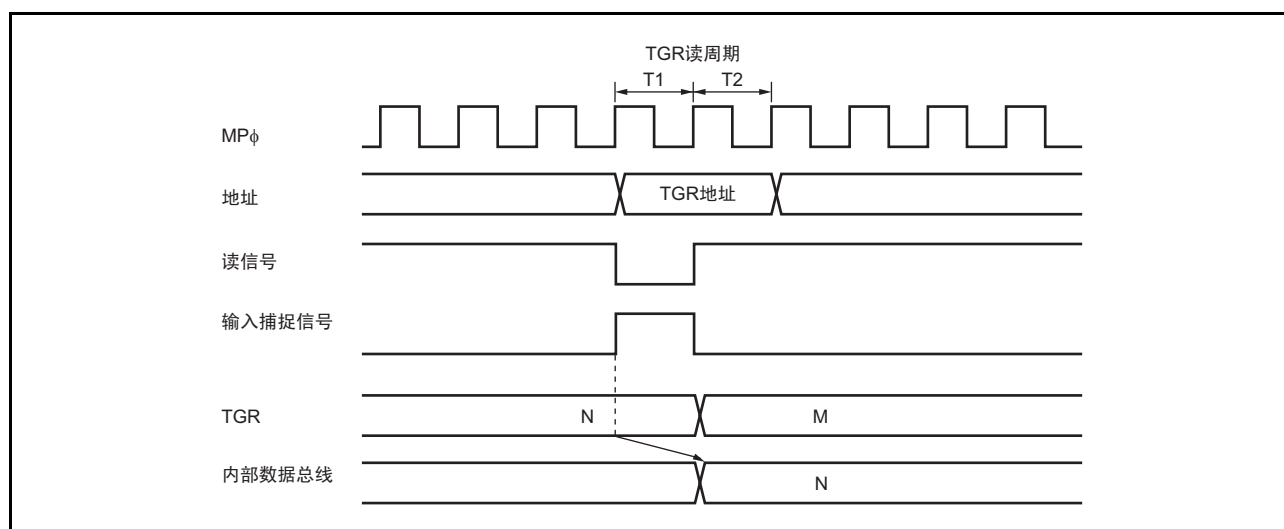


图 10.125 TGR 的读操作和输入捕捉的竞争 (通道 0 ~ 4)

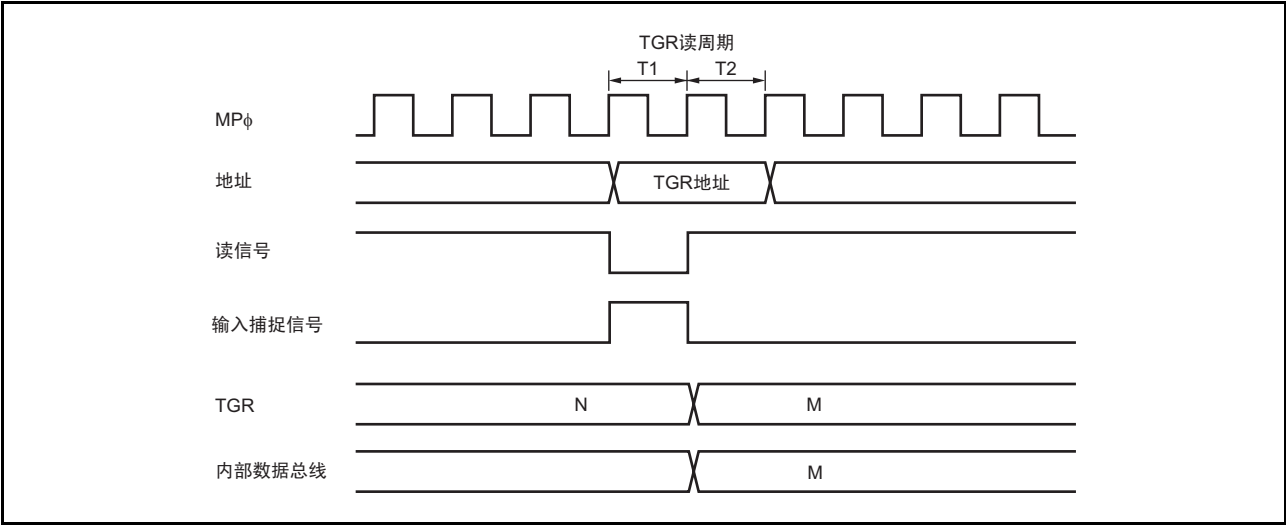


图 10.126 TGR 的读操作和输入捕捉的竞争（通道 5）

10.7.10 TGR 的写操作和输入捕捉的竞争

如果在 TGR 写周期的 T2 状态产生输入捕捉信号，在通道 0 ~ 4 时不写 TGR 而优先输入捕捉，但是在通道 5 时写 TGR 并产生输入捕捉信号。

此时序如图 10.127 和图 10.128 所示。

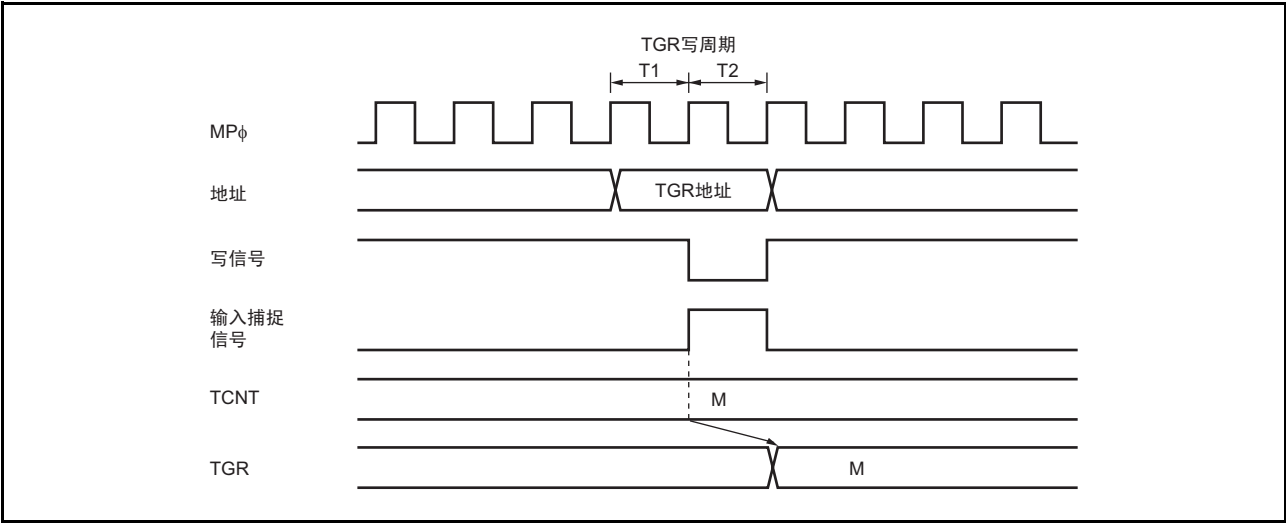


图 10.127 TGR 的写操作和输入捕捉的竞争（通道 0 ~ 4）

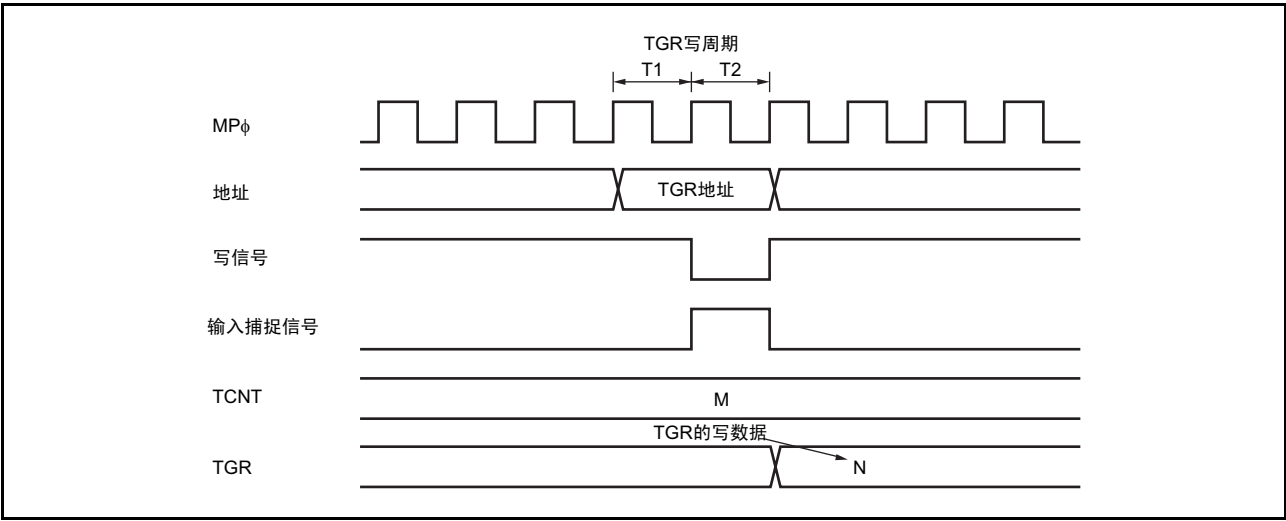


图 10.128 TGR 的写操作和输入捕捉的竞争 (通道 5)

10.7.11 缓冲寄存器的写操作和输入捕捉的竞争

如果在缓冲器写周期的 T2 状态产生输入捕捉信号，就不写缓冲寄存器而优先缓冲运行。此时序如图 10.129 所示。

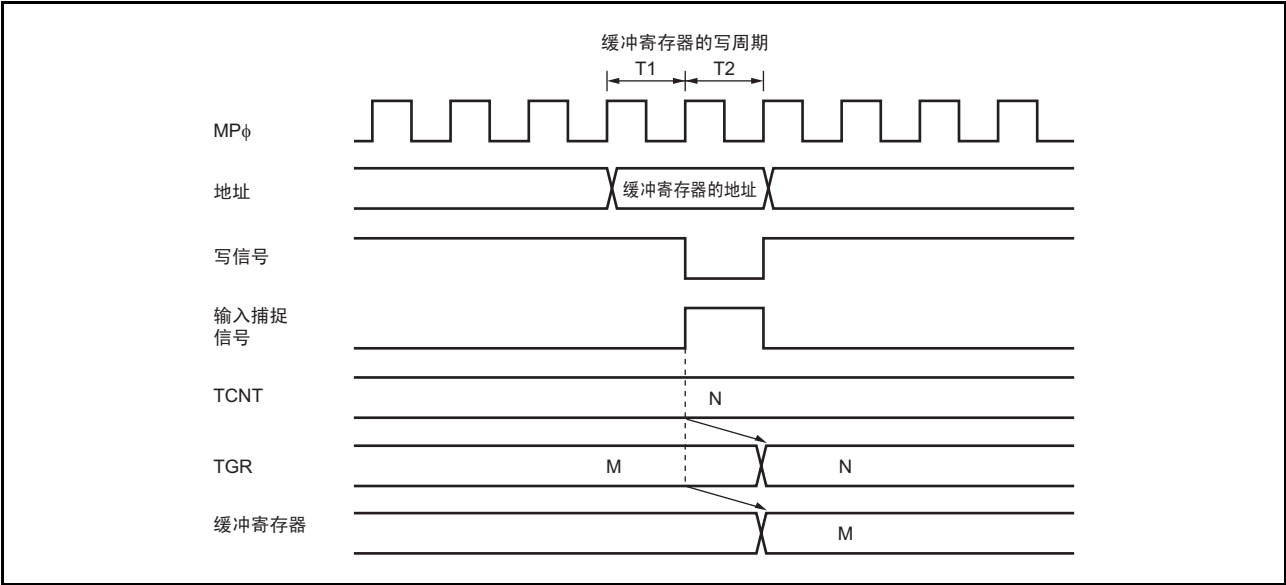


图 10.129 缓冲寄存器的写操作和输入捕捉的竞争

10.7.12 级联中 TCNT_2 的写操作和上溢 / 下溢的竞争

将定时器的计数器 (TCNT_1 和 TCNT_2) 进行级联, 如果 TCNT_1 在计数的瞬间 (TCNT_2 发生上溢 / 下溢的瞬间) 和 TCNT_2 写周期的 T2 状态发生竞争, 就写 TCNT_2 并禁止 TCNT_1 的计数信号。此时, TGRA_1 用作比较匹配寄存器, 当 TGRA_1 和 TCNT_1 的值相同时, 就产生比较匹配信号。

如果选择 TCNT_1 计数时钟作为通道 0 的输入捕捉源, TGRA_0 ~ D_0 就进行输入捕捉; 如果选择 TGRC_0 的比较匹配 / 输入捕捉作为 TGRB_1 的输入捕捉源, TGRB_1 就进行输入捕捉。

此时序如图 10.130 所示。

另外, 在级联运行中设定 TCNT 的清除时, 必须同步设定通道 1 和通道 2。

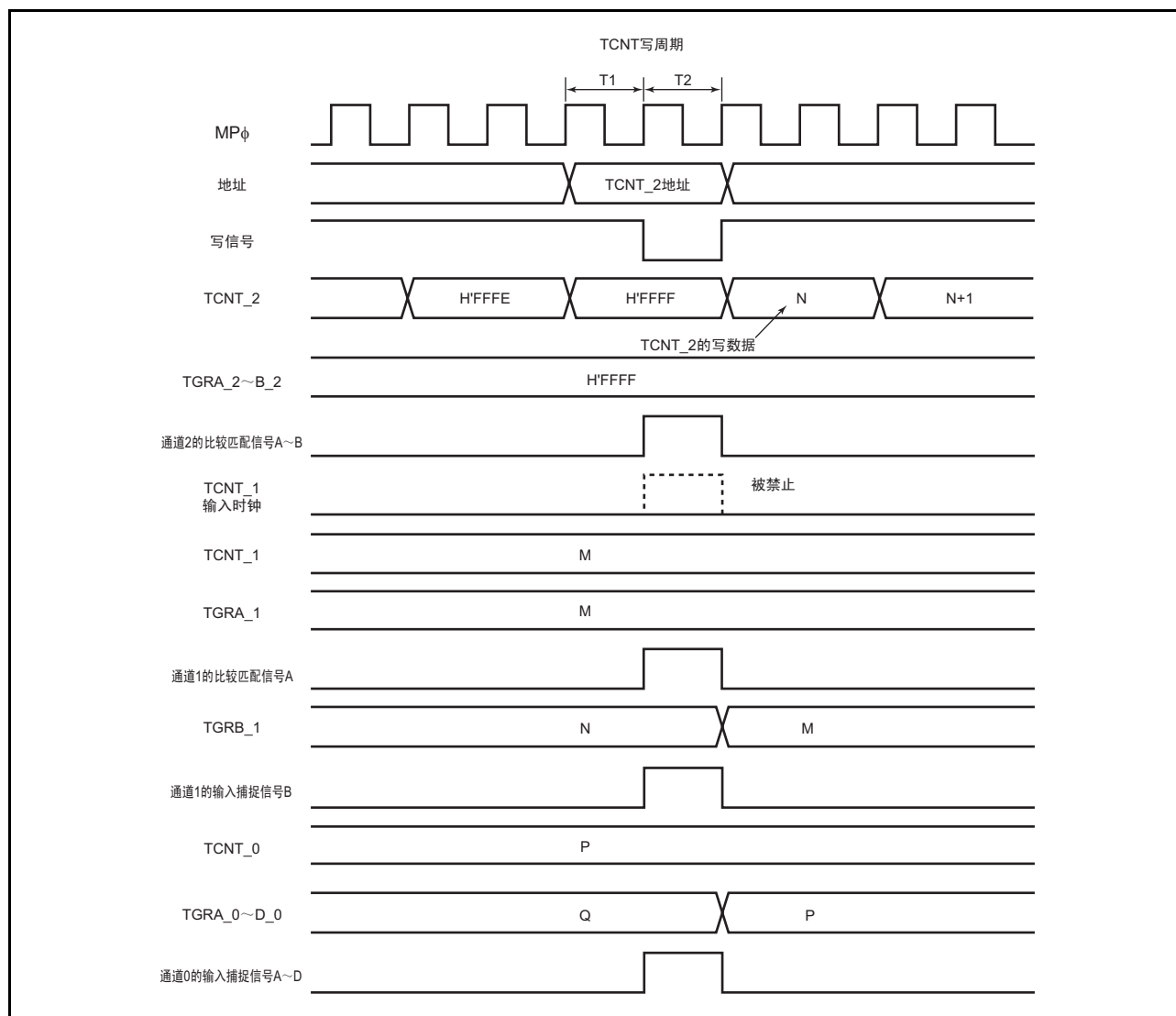


图 10.130 级联中 TCNT_2 的写操作和上溢 / 下溢的竞争

10.7.13 互补 PWM 模式停止时的计数器值

在互补 PWM 模式中, 如果 TCNT_3 和 TCNT_4 在运行中停止计数, TCNT_3 就为定时器的空载时间寄存器 (TDDR) 的值, TCNT_4 为 H'0000。

当重新开始互补 PWM 模式时, 自动从初始状态开始计数。

此说明如图 10.131 所示。

另外, 在其他运行模式中开始计数时, 必须给 TCNT_3 和 TCNT_4 设定计数的初始值。

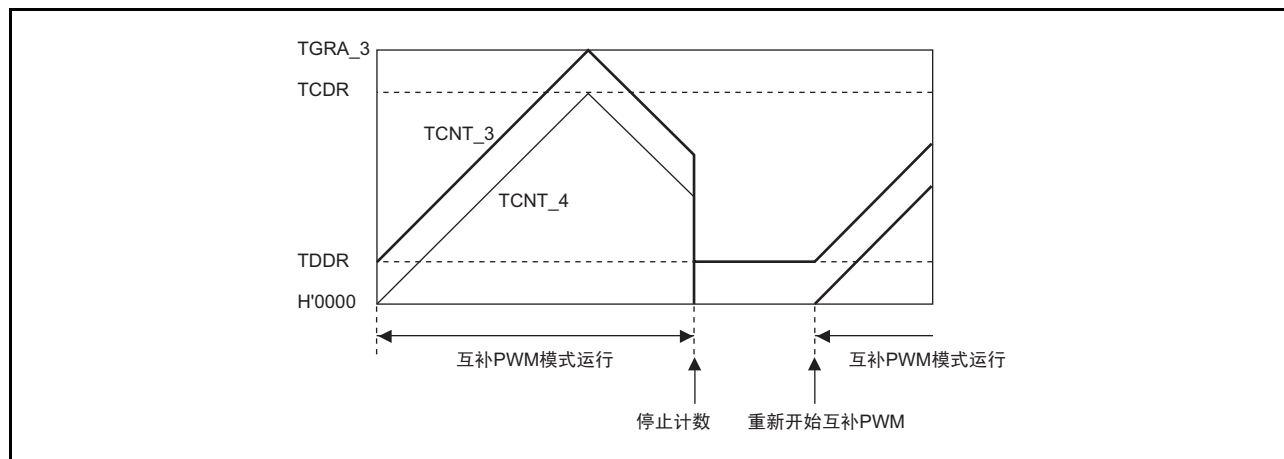


图 10.131 互补 PWM 模式停止时的计数器值

10.7.14 互补 PWM 模式的缓冲运行设定

在互补 PWM 模式中, 必须在缓冲运行中改写 PWM 周期设定寄存器 (TGRA_3)、定时器的周期数据寄存器 (TCDR) 和占空比设定寄存器 (TGRB_3、TGRA_4、TGRB_4)。

按照 TMDR_3 的 BFA 位和 BFB 位的设定, 进行互补 PWM 模式的通道 3 和通道 4 的缓冲运行。如果将 TMDR_3 的 BFA 位置 1, TGRC_3、TGRC_4 和 TCBR 就分别用作 TGRA_3、TGRA_4 和 TCDR 的缓冲寄存器。

10.7.15 复位同步 PWM 模式的缓冲运行和比较匹配标志

要在复位同步 PWM 模式中设定缓冲运行时, 必须将 TMDR_4 的 BFA 和 BFB 位置 0。如果将 TMDR_4 的 BFA 位置 1, TIOC4C 引脚就不能输出波形。

按照 TMDR_3 的 BFA 位和 BFB 位的设定, 进行复位同步 PWM 模式的通道 3 和通道 4 的缓冲运行。例如, 如果将 TMDR_3 的 BFA 位置 1, TGRC_3 和 TGRC_4 就分别用作 TGRA_3 和 TGRA_4 的缓冲寄存器。

当 TGRC_3 和 TGRD_3 用作缓冲寄存器时, TSR_3、TSR_4 的 TGFC 位和 TGFD 位不被置位。

在将 TMDR_3 的 BFA 和 BFB 位置 1、TMDR_4 的 BFA 和 BFB 位置 0 时, TGR_3、TGR_4、TIOC3 和 TIOC4 的运行例子如图 10.132 所示。

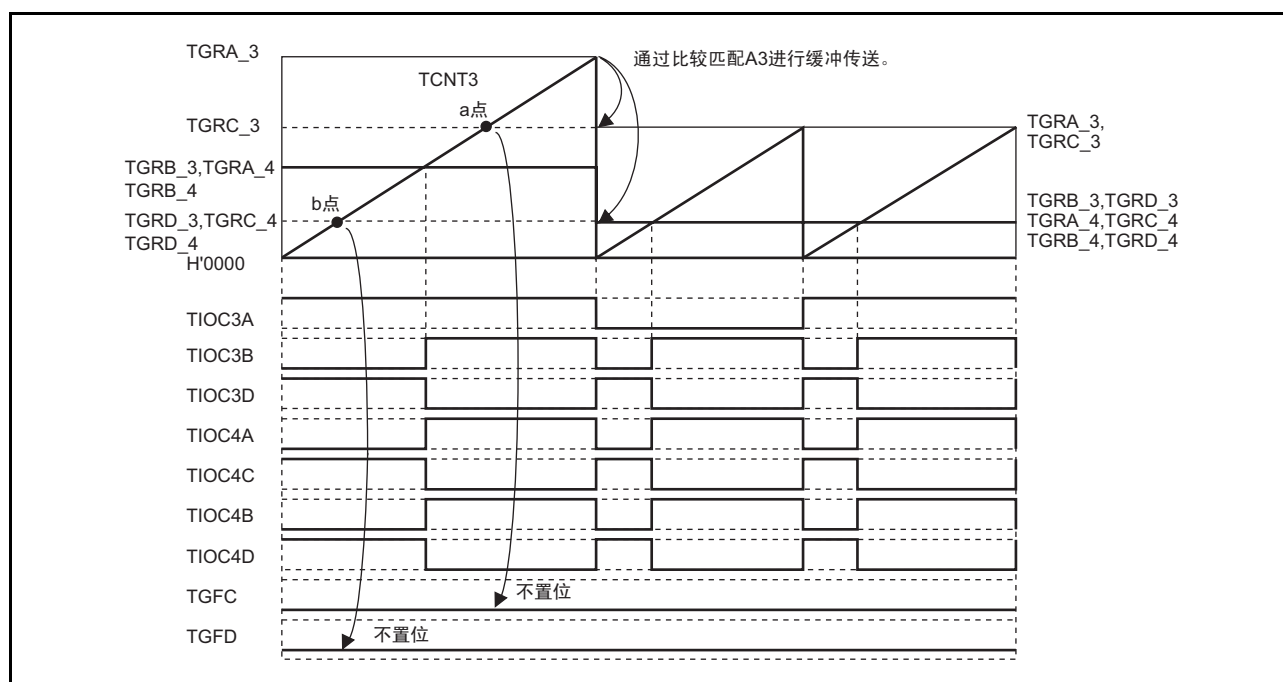


图 10.132 复位同步 PWM 模式的缓冲运行和比较匹配标志

10.7.16 复位同步 PWM 模式的上溢标志

如果设定复位同步 PWM 模式并将 TSTR 的 CST3 位置 1, TCNT_3 和 TCNT_4 就开始计数。此时, TCNT_4 的计数时钟源和计数边沿服从 TCR_3 的设定。

在复位同步 PWM 模式中, 当周期寄存器 TGRA_3 的设定值为 H'FFFF 并将 TGRA_3 的比较匹配指定为计数器清除源时, 如果 TCNT_3 和 TCNT_4 递增计数到 H'FFFF, 就产生和 TGRA_3 的比较匹配, 并同时清除 TCNT_3 和 TCNT_4。此时, TSR 的上溢标志 TCFV 位不被置位。

在复位同步 PWM 模式中, 当周期寄存器 TGRA_3 的设定值为 H'FFFF 并将 TGRA_3 的比较匹指定为计数器清除源而未进行同步设定时, TCFV 位的运行例子如图 10.133 所示。

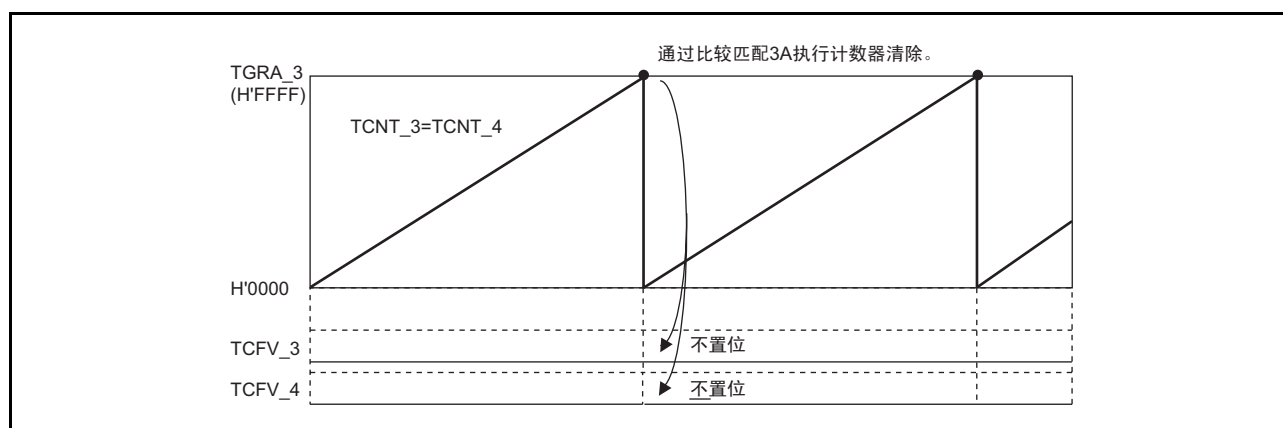


图 10.133 复位同步 PWM 模式的上溢标志

10.7.17 上溢 / 下溢和计数器清除的竞争

如果同时发生上溢 / 下溢和计数器清除，就不将 TSR 的 TCFV/TCFU 标志置位而优先清除 TCNT。将 TGR 的比较匹配作为清除源并给 TGR 设定 H'FFFF 时的运行时序如图 10.134 所示。

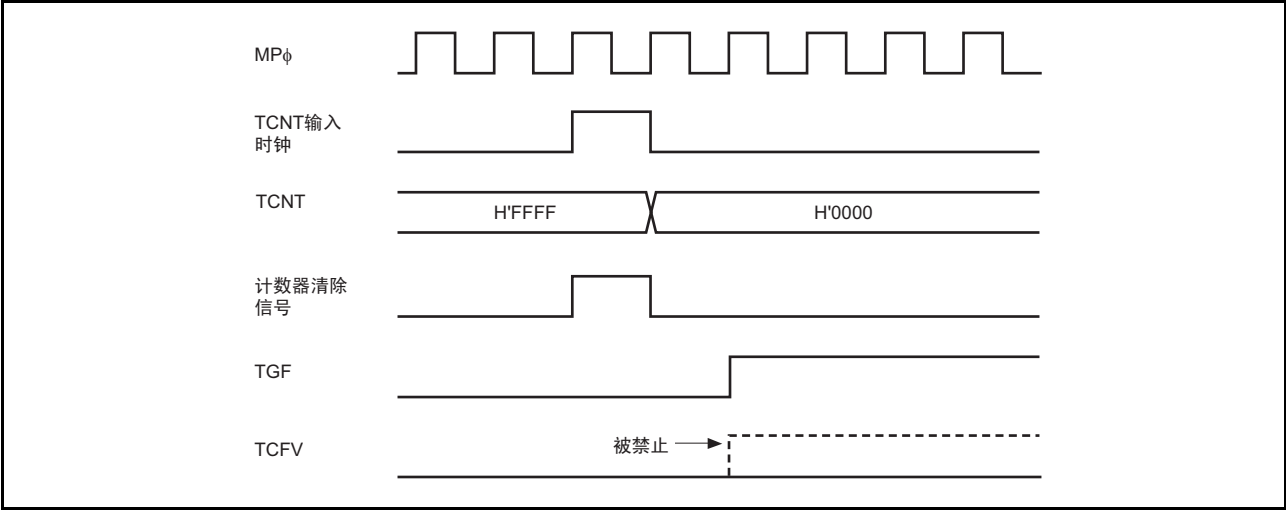


图 10.134 上溢和计数器清除的竞争

10.7.18 TCNT 的写操作和上溢 / 下溢的竞争

即使在 TCNT 写周期的 T2 状态发生递增 / 递减计数和上溢 / 下溢，也优先写 TCNT 而不将 TSR 的 TCFV/TCFU 标志置位。

TCNT 的写操作和上溢发生竞争时的运行时序如图 10.135 所示。

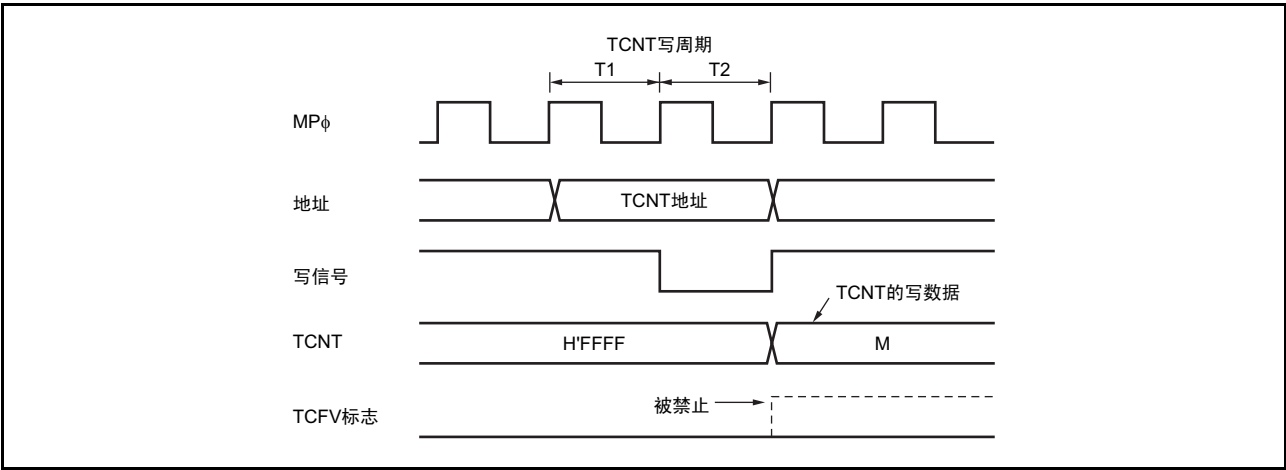


图 10.135 TCNT 的写操作和上溢的竞争

10.7.19 从正常模式或者 PWM 模式 1 转移到复位同步 PWM 模式时的注意事项

从通道 3、4 的正常模式或者 PWM 模式转移到复位同步 PWM 模式时，在将输出引脚（TIOC3B、TIOC3D、TIOC4A、TIOC4C、TIOC4B 和 TIOC4D）置为高电平的状态下停止计数器。必须注意：如果在转移到同步 PWM 模式后运行，就不能正确地进行引脚的初始输出。

要从正常模式转移到同步 PWM 模式时，必须在将 H'11 写到 TIORH_3、TIO RL_3、TIORH_4 和 TIO RL_4 寄存器并将输出引脚初始化为低电平后，先设定寄存器的初始值 H'00 再进行模式转移。

要从 PWM 模式 1 转移到复位同步 PWM 模式时，必须暂且先转移到正常模式，在将输出引脚初始化为低电平后，先设定寄存器的初始值 H'00 再转移到复位同步 PWM 模式。

10.7.20 互补 PWM 模式或者复位同步 PWM 模式的输出电平

在通道 3、4 为互补 PWM 模式或者复位同步 PWM 模式时，通过定时器的输出控制寄存器（TOCR）的 OLSP 位和 OLSN 位设定 PWM 波形的输出电平，并且 TIOR 必须为 H'00。

10.7.21 模块待机时的中断

如果在请求中断的状态下变为模块待机，就不能清除 CPU 的中断源或者 DTC 的启动源。

必须预先禁止中断再设定为模块待机模式。

10.7.22 级联中 TCNT_1 和 TCNT_2 的同时输入捕捉

在将定时器的计数器 1、2（TCNT_1 和 TCNT_2）进行级联后用作 32 位计数器时，即使 TIOC1A 和 TIOC2A 或者 TIOC1B 和 TIOC2B 同时进行输入捕捉的输入，也会与内部时钟同步将输入到 TCNT_1 和 TCNT_2 的外部输入捕捉信号输入取到内部，此时有可能因 TIOC1A 和 TIOC2A 或者 TIOC1B 和 TIOC2B 的取时序产生偏差而不能正确地捕捉级联计数器的值。

例如，TCNT_1（高 16 位的计数器）应该捕捉 TCNT_2（低 16 位的计数器）的上溢产生的递增计数值，却会捕捉递增计数前的计数值。此时，应该将 TCNT_1=H'FFF1 和 TCNT_2=H'0000 的值传送到 TGRA_1 和 TGRA_2 或者 TGRB_1 和 TGRB_2，却会误传送 TCNT_1=H'FFF0 和 TCNT_2=H'0000 的值。

在 MTU2 中，追加了通过 1 个输入捕捉的输入能同时捕捉 TCNT_1 和 TCNT_2 的功能，通过使用此功能，TCNT_1 和 TCNT_2 的捕捉时序不会产生偏差，并能读取 32 位计数器。详细内容请参照“10.3.8 定时器的输入捕捉控制寄存器（TICCR）”。

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，能在任意模式中进行波形输出。

- 正常模式（通道 0～4）
- PWM 模式 1（通道 0～4）
- PWM 模式 2（通道 0～2）
- 相位计数模式 1～4（通道 1、2）
- 互补 PWM 模式（通道 3、4）
- 复位同步 PWM 模式（通道 3、4）

在此说明各模式的 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*) 在复位或者待机模式时初始化为 L 电平。因为通过引脚功能控制器 (PFC) 选择 MTU2 的引脚功能, 所以在设定 PFC 时, 将当时的 MTU2 的引脚状态输出到端口。如果在复位后立即通过 PFC 选择 MTU2 的输出, 就将 MTU2 输出的初始状态 (L 电平) 输出到端口。在有效电平为 L 电平时, 系统能立刻运行, 所以必须在完成 MTU2 输出引脚的初始设定后, 再设定 PFC。

【注】 * 通道号 + 端口符号。

10.8.3 因运行中的异常等而重新设定时的运行

如果在 MTU2 运行中发生异常, 就必须通过系统截止 MTU2 的输出。即, 通过 PFC 将引脚的输出转换为端口输出, 并输出有效电平的反相信号。另外, 对于大电流引脚, 也能使用端口输出的允许 (POE), 通过硬件截止输出。因运行中的异常等而重新设定时的引脚的初始化步骤以及重新设定后在其他运行模式中重新开始时的步骤如下所示。

如上所述, MTU2 有 6 种运行模式, 所以有 36 种模式转移的组合, 但是有通道和模式的组合中不存在的转移, 模式转移的组合一览表如表 10.59 所示。

表中使用下述符号表示:

Normal: 正常模式

PWM1: PWM 模式 1

PWM2: PWM 模式 2

PCM: 相位计数模式 1 ~ 4

CPWM: 互补 PWM 模式

RPWM: 复位同步 PWM 模式

表 10.59 模式转移的组合

	Normal	PWM1	PWM2	PCM	CPWM	RPWM
Normal	(1)	(2)	(3)	(4)	(5)	(6)
PWM1	(7)	(8)	(9)	(10)	(11)	(12)
PWM2	(13)	(14)	(15)	(16)	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 因运行中的异常等而初始化引脚的步骤和模式转移的概要

- 在转移到定时器的 I/O 控制寄存器 (TIOCR) 选择的引脚输出电平模式 (Normal、PWM1、PWM2、PCM) 时, 必须通过设定 TIOCR 初始化引脚。
- 因为在 PWM 模式 1 中不将波形输出到 TIOC*B (TIOC*D) 引脚, 所以, 即使设定 TIOCR 也不初始化引脚。要进行初始化时, 必须在正常模式中进行初始化后转移到 PWM 模式 1。
- 因为在 PWM 模式 2 中不将波形输出到周期寄存器的引脚, 所以, 即使设定 TIOCR 也不初始化引脚。要进行初始化时, 必须在正常模式中进行初始化后转移到 PWM 模式 2。
- 在正常模式或者 PWM 模式 2 中, 如果 TGRC 和 TGRD 都用作缓冲寄存器, 即使设定 TIOCR 也不初始化缓冲寄存器的引脚。要进行初始化时, 必须在解除缓冲模式进行初始化后重新设定缓冲模式。
- 在 PWM 模式 1 中, 如果 TGRC 或者 TGRD 用作缓冲寄存器, 即使设定 TIOCR 也不初始化 TGRC 的引脚。要初始化 TGRC 的引脚时, 必须在解除缓冲模式进行初始化后重新设定缓冲模式。

- 在转移到定时器的输出控制寄存器 (TOCR) 选择的引脚输出电平模式 (CPWM、RPWM) 时, 必须转移到正常模式并通过 TIOR 进行初始化, 在将 TIOR 返回到初始值后, 通过定时器的输出主控允许寄存器 (TOER) 暂时禁止通道 3、4 的输出, 然后按照模式设定步骤 (TOCR 的设定、TMDR 的设定和 TOER 的设定) 运行。

【注】 本项记述中的 * 为通道号。

按照表 10.59 的组合 No. 初始化引脚的步骤如下所示。另外, 有效电平为 L 电平。

(1) 在正常模式的运行中发生异常并在正常模式中重新开始时的运行

在正常模式中发生异常, 重新设定后在正常模式中重新开始时的说明如图 10.136 所示。

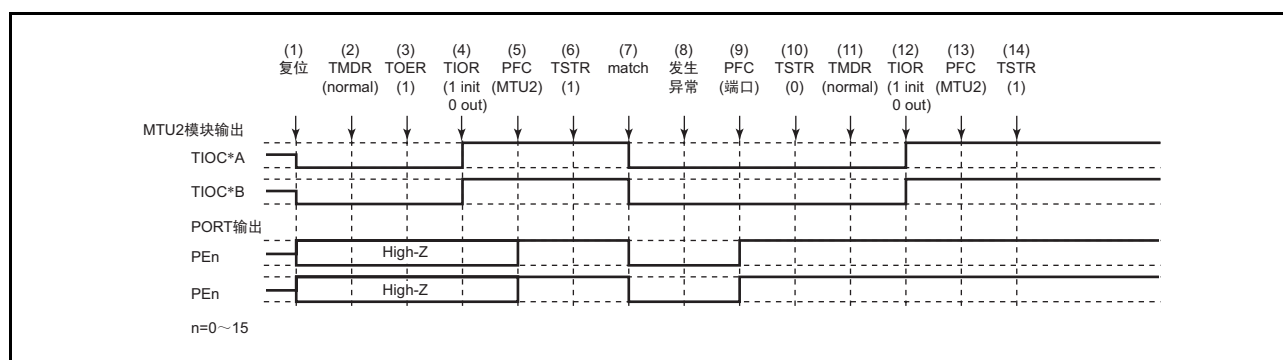


图 10.136 在正常模式中发生异常并在正常模式中恢复

- (1) 在复位后, MTU2 的输出为低电平, 端口为高阻抗状态。
- (2) 在复位后, TMDR 为正常模式。
- (3) 在通道 3、4 中通过 TIOR 初始化引脚前, 必须通过 TOER 允许输出。
- (4) 必须通过 TIOR 初始化引脚 (在例中, 初始输出为高电平, 比较匹配为低电平输出)。
- (5) 必须通过 PFC 设定为 MTU2 输出。
- (6) 通过 TSTR 开始计数。
- (7) 通过产生比较匹配, 输出低电平。
- (8) 发生了异常。
- (9) 必须通过 PFC 设定为端口输出, 输出有效电平的反相信号。
- (10) 通过 TSTR 停止计数。
- (11) 在正常模式中重新开始时不需要此步骤。
- (12) 必须通过 TIOR 初始化引脚。
- (13) 必须通过 PFC 设定为 MTU2 输出。
- (14) 通过 TSTR 重新开始。

(2) 在正常模式的运行中发生异常并在 PWM 模式 1 中重新开始时的运行

在正常模式中发生异常，重新设定后在 PWM 模式 1 中重新开始时的说明如图 10.137 所示。

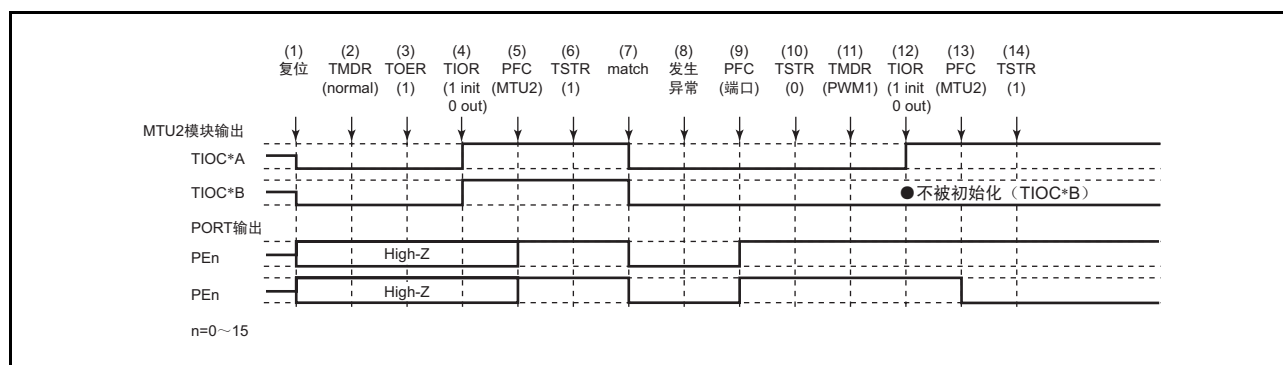


图 10.137 在正常模式中发生异常并在 PWM 模式 1 中恢复

(1) ~ (10) 与图 10.136 相同。

(11) 设定 PWM 模式 1。

(12) 必须通过 TIOR 初始化引脚 (在 PWM 模式 1 中不初始化 TIOC*B。要进行初始化时，必须在正常模式中进行初始化后转移到 PWM 模式 1)。

(13) 必须通过 PFC 设定为 MTU2 输出。

(14) 通过 TSTR 重新开始。

(3) 在正常模式的运行中发生异常并在 PWM 模式 2 中重新开始时的运行

在正常模式中发生异常，重新设定后在 PWM 模式 2 中重新开始时的说明如图 10.138 所示。

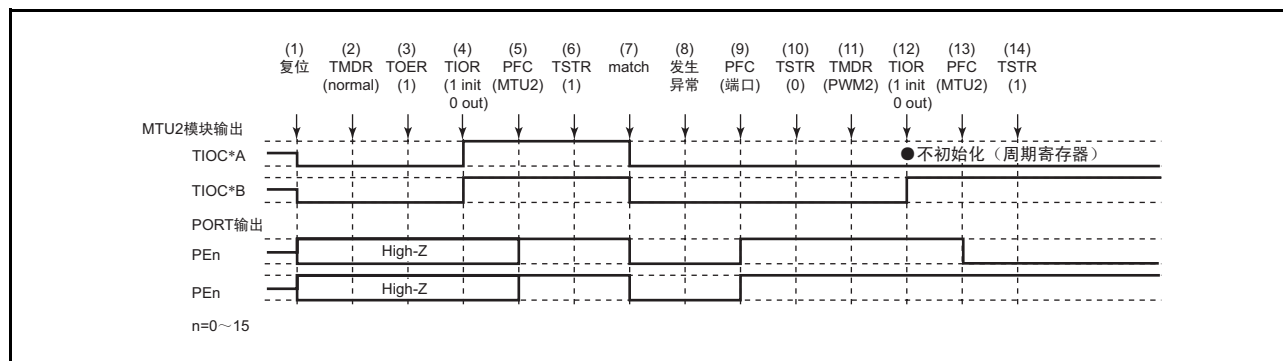


图 10.138 在正常模式中发生异常并在 PWM 模式 2 中恢复

(1) ~ (10) 与图 10.136 相同。

(11) 设定 PWM 模式 2。

(12) 必须通过 TIOR 初始化引脚 (在 PWM 模式 2 中不初始化周期寄存器的引脚。要进行初始化时，必须在正常模式中进行初始化后转移到 PWM 模式 2)。

(13) 通过 PFC 设定为 MTU2 输出。

(14) 通过 TSTR 重新开始。

【注】 只能在通道 0 ~ 2 中设定 PWM 模式 2，因此不需要设定 TOER。

(4) 在正常模式的运行中发生异常并在相位计数模式中重新开始时的运行

在正常模式中发生异常，重新设定后在相位计数模式中重新开始的说明如图 10.139 所示。

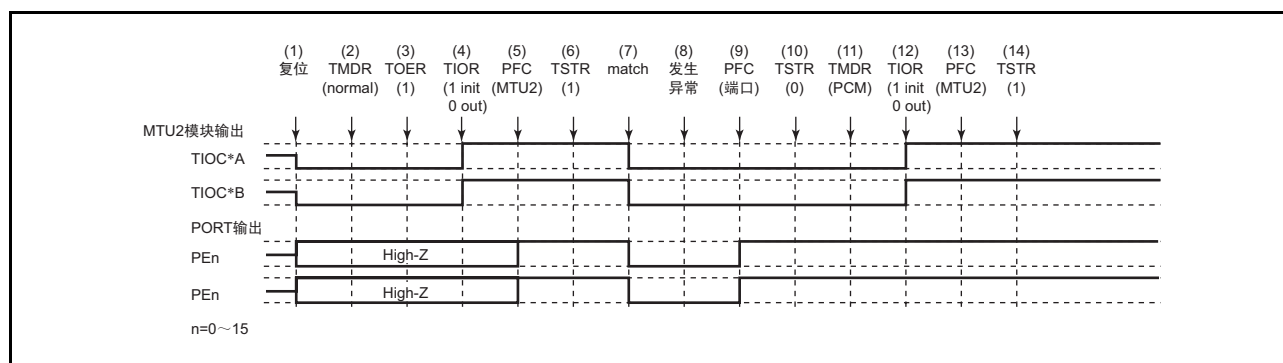


图 10.139 在正常模式中发生异常并在相位计数模式中恢复

- (1) ~ (10) 与图 10.136 相同。
- (11) 设定相位计数模式。
- (12) 必须通过 TIOR 初始化引脚。
- (13) 必须通过 PFC 设定为 MTU2 输出。
- (14) 通过 TSTR 重新开始。

【注】 只能在通道 1、2 中设定相位计数模式，因此不需要设定 TOER。

(5) 在正常模式的运行中发生异常并在互补 PWM 模式中重新开始时的运行

在正常模式中发生异常，重新设定后在互补 PWM 模式中重新开始的说明如图 10.140 所示。

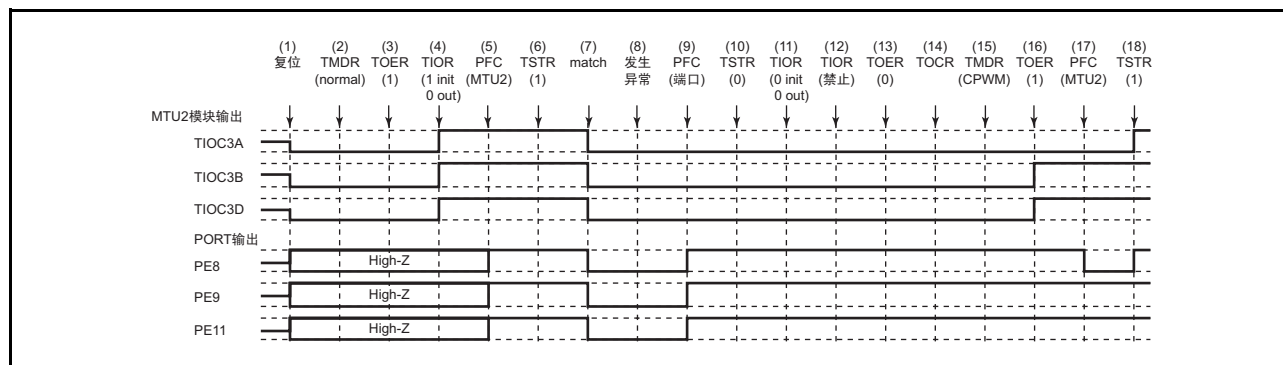


图 10.140 在正常模式中发生异常并在互补 PWM 模式中恢复

- (1) ~ (10) 与图 10.136 相同。
- (11) 必须通过 TIOR 初始化正常模式的波形生成部。
- (12) 必须通过 TIOR 禁止正常模式的波形生成部的运行。
- (13) 必须通过 TOER 禁止通道 3、4 的输出。
- (14) 必须通过 TOCR 选择互补 PWM 的输出电平并允许或者禁止周期输出。
- (15) 设定互补 PWM。
- (16) 必须通过 TOER 允许通道 3、4 的输出。
- (17) 必须通过 PFC 设定为 MTU2 输出。
- (18) 通过 TSTR 重新开始。

(6) 在正常模式的运行中发生异常并在复位同步 PWM 模式中重新开始时的运行

在正常模式中发生异常，重新设定后在复位同步 PWM 模式中重新开始的说明如图 10.141 所示。

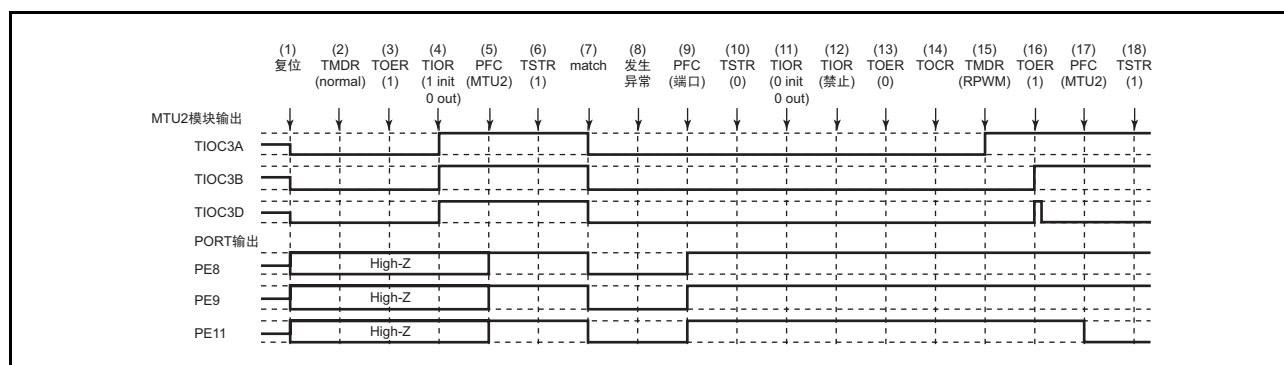


图 10.141 在正常模式中发生异常并在复位同步 PWM 模式中恢复

(1) ~ (13) 与图 10.136 相同。

(14) 必须通过 TOCR 选择复位同步 PWM 的输出电平并允许或者禁止周期输出。

(15) 设定复位同步 PWM。

(16) 必须通过 TOER 允许通道 3、4 的输出。

(17) 必须通过 PFC 设定为 MTU2 输出。

(18) 通过 TSTR 重新开始。

(7) 在 PWM 模式 1 的运行中发生异常并在正常模式中重新开始时的运行

在 PWM 模式 1 中发生异常，重新设定后在正常模式中重新开始的说明如图 10.142 所示。

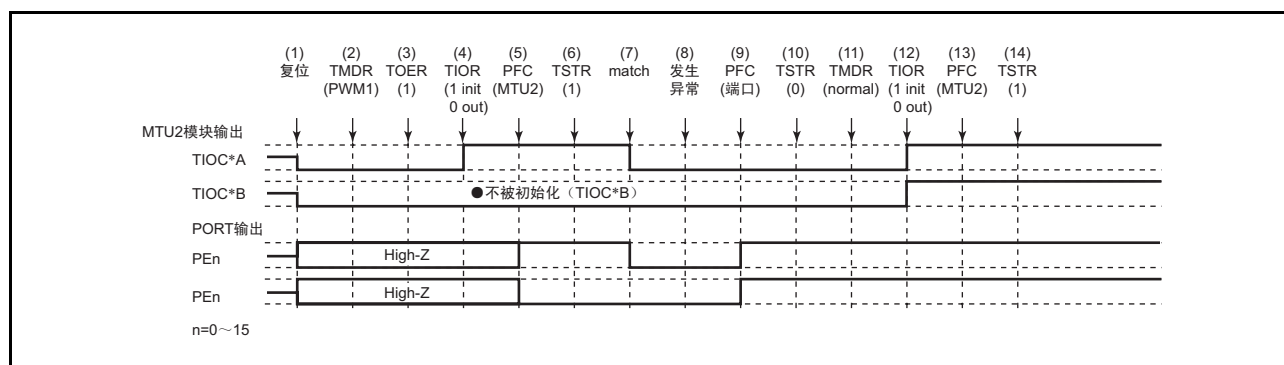


图 10.142 在 PWM 模式 1 中发生异常并在正常模式中恢复

(1) 在复位后，MTU2 输出为低电平，端口为高阻抗状态。

(2) 必须设定 PWM 模式 1。

(3) 在通道 3、4 中通过 TIOR 初始化引脚前，必须通过 TOER 允许输出。

(4) 通过 TIOR 初始化引脚（在例中，初始输出为高电平，比较匹配为低电平输出，在 PWM 模式 1 中不初始化 TIOC*B）。

(5) 通过 PFC 设定为 MTU2 输出。

(6) 通过 TSTR 开始计数。

(7) 通过产生比较匹配，输出 L 电平。

(8) 发生了异常。

- (9) 必须通过 PFC 设定为端口输出，输出有效电平的反相信号。
- (10) 通过 TSTR 停止计数。
- (11) 必须设定正常模式。
- (12) 必须通过 TIOR 初始化引脚。
- (13) 必须通过 PFC 设定为 MTU2 输出。
- (14) 通过 TSTR 重新开始。

(8) 在 PWM 模式 1 的运行中发生异常并在 PWM 模式 1 中重新开始时的运行

在 PWM 模式 1 中发生异常，重新设定后在 PWM 模式 1 中重新开始时的说明如图 10.143 所示。

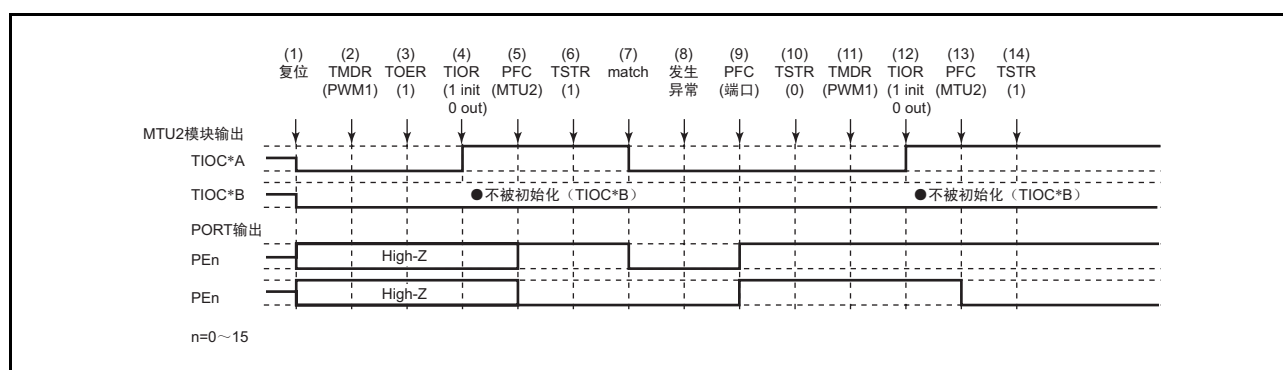


图 10.143 在 PWM 模式 1 中发生异常并在 PWM 模式 1 中恢复

- (1) ~ (10) 与图 10.142 相同。
- (11) 在 PWM 模式 1 中重新开始时不需要此步骤。
- (12) 必须通过 TIOR 初始化引脚（在 PWM 模式 1 中不初始化 TIOC*B）。
- (13) 必须通过 PFC 设定为 MTU2 输出。
- (14) 通过 TSTR 重新开始。

(9) 在 PWM 模式 1 的运行中发生异常并在 PWM 模式 2 中重新开始时的运行

在 PWM 模式 1 中发生异常，重新设定后在 PWM 模式 2 中重新开始时的说明如图 10.144 所示。

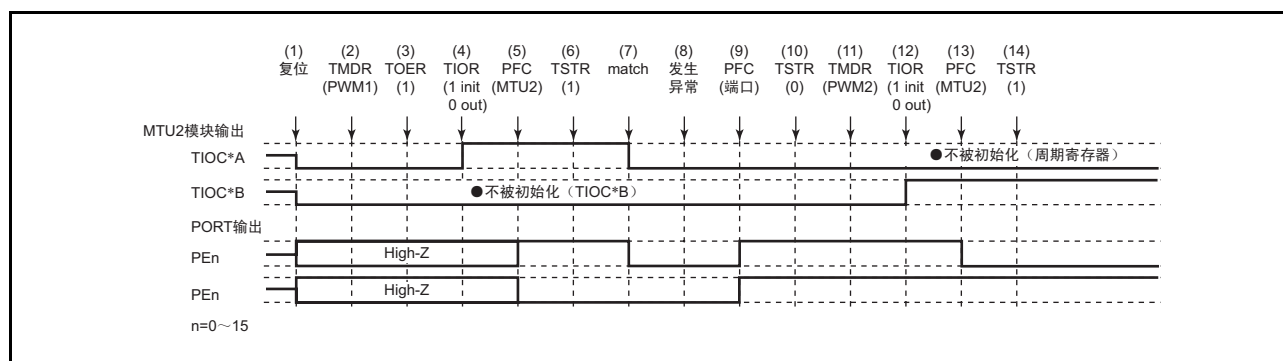


图 10.144 在 PWM 模式 1 中发生异常并在 PWM 模式 2 中恢复

- (1) ~ (10) 与图 10.142 相同。
- (11) 设定 PWM 模式 2。

- (12) 必须通过 TIOR 初始化引脚 (在 PWM 模式 2 中不初始化周期寄存器的引脚)。
- (13) 必须通过 PFC 设定为 MTU2 输出。
- (14) 通过 TSTR 重新开始。

【注】 只能在通道 0 ~ 2 中设定 PWM 模式 2，因此不需要设定 TOER。

(10) 在 PWM 模式 1 的运行中发生异常并在相位计数模式中重新开始时的运行

在 PWM 模式 1 发生异常，重新设定后在相位计数模式中重新开始时的说明如图 10.145 所示。

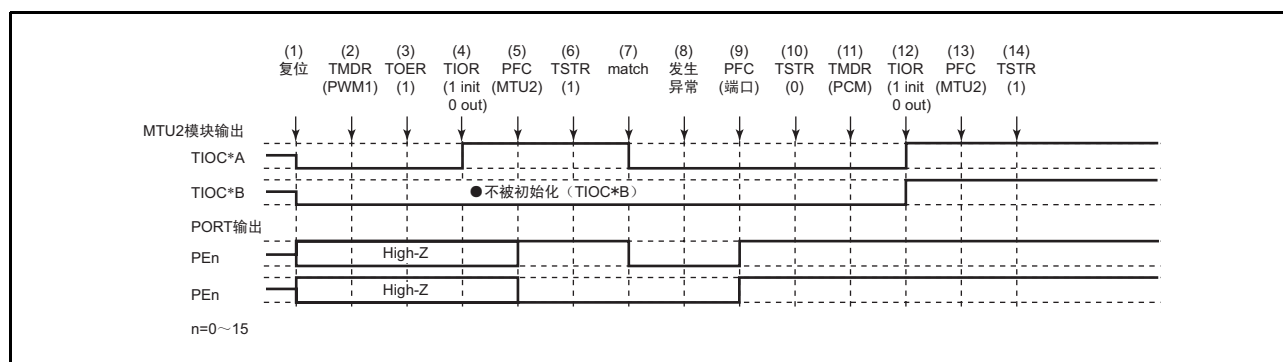


图 10.145 在 PWM 模式 1 中发生异常并在相位计数模式中恢复

- (1) ~ (10) 与图 10.142 相同。
- (11) 设定相位计数模式。
- (12) 必须通过 TIOR 初始化引脚。
- (13) 必须通过 PFC 设定为 MTU2 输出。
- (14) 通过 TSTR 重新开始。

【注】 只能在通道 1、2 中设定相位计数模式，因此不需要设定 TOER。

(11) 在 PWM 模式 1 的运行中发生异常并在互补 PWM 模式中重新开始时的运行

在 PWM 模式 1 中发生异常，重新设定后在互补 PWM 模式中重新开始时的说明如图 10.146 所示。

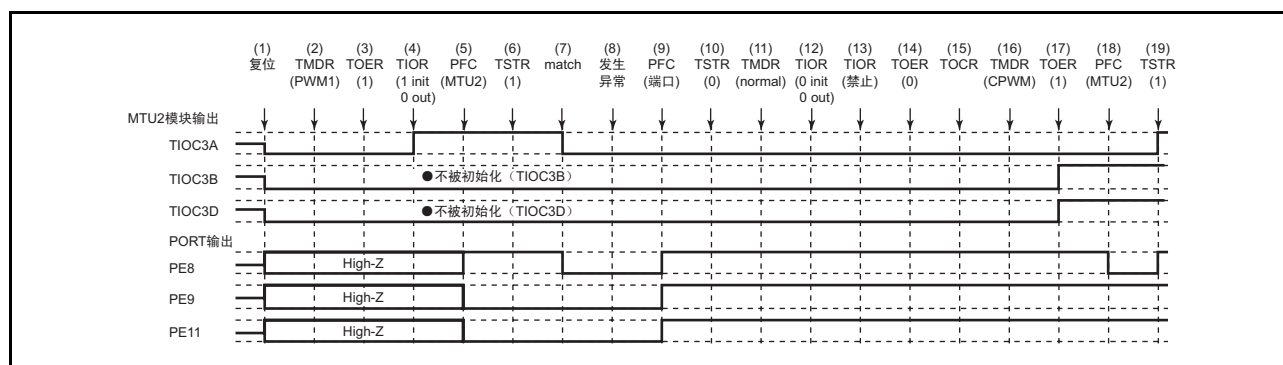


图 10.146 在 PWM 模式 1 中发生异常并在互补 PWM 模式中恢复

- (1) ~ (10) 与图 10.142 相同。
- (11) 为了初始化波形生成部，必须设定正常模式。
- (12) 必须通过 TIOR 初始化 PWM 模式 1 的波形生成部。

- (13) 必须通过 TIOR 禁止 PWM 模式 1 的波形生成部的运行。
- (14) 必须通过 TOER 禁止通道 3、4 的输出。
- (15) 必须通过 TOCR 选择互补 PWM 的输出电平并允许或者禁止周期输出。
- (16) 设定互补 PWM。
- (17) 必须通过 TOER 设定允许通道 3、4 的输出。
- (18) 必须通过 PFC 设定为 MTU2 输出。
- (19) 通过 TSTR 重新开始。

(12) 在 PWM 模式 1 中的运行发生异常并在复位同步 PWM 模式中重新开始时的运行

在 PWM 模式 1 中发生异常，重新设定后在复位同步 PWM 模式中重新开始时的说明如图 10.147 所示。

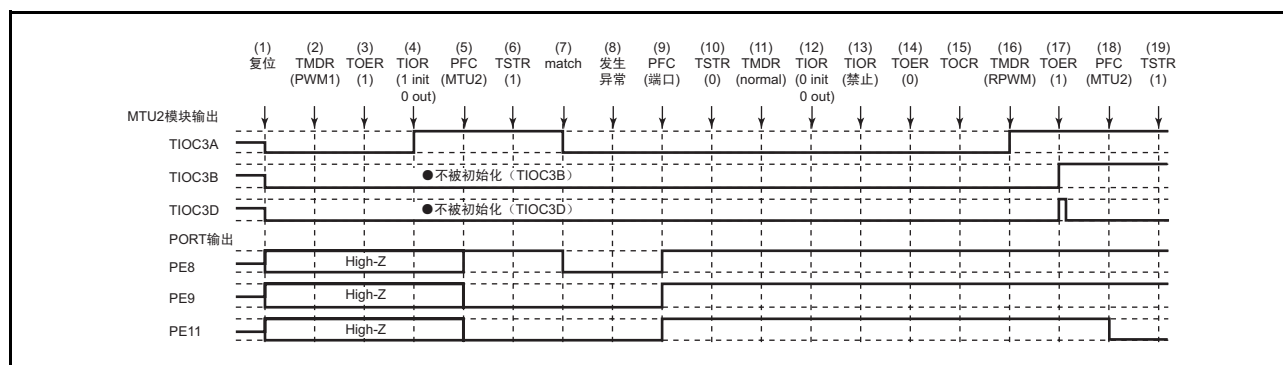


图 10.147 在 PWM 模式 1 中发生异常并在复位同步 PWM 模式中恢复

- (1) ~ (14) 与图 10.146 相同。
- (15) 必须通过 TOCR 选择复位同步 PWM 的输出电平并允许或者禁止周期输出。
- (16) 设定复位同步 PWM。
- (17) 必须通过 TOER 允许通道 3、4 的输出。
- (18) 通过 PFC 设定为 MTU2 输出。
- (19) 通过 TSTR 重新开始。

(13) 在 PWM 模式 2 的运行中发生异常并在正常模式中重新开始时的运行

在 PWM 模式 2 中发生异常，重新设定后在正常模式中重新开始时的说明如图 10.148 所示。

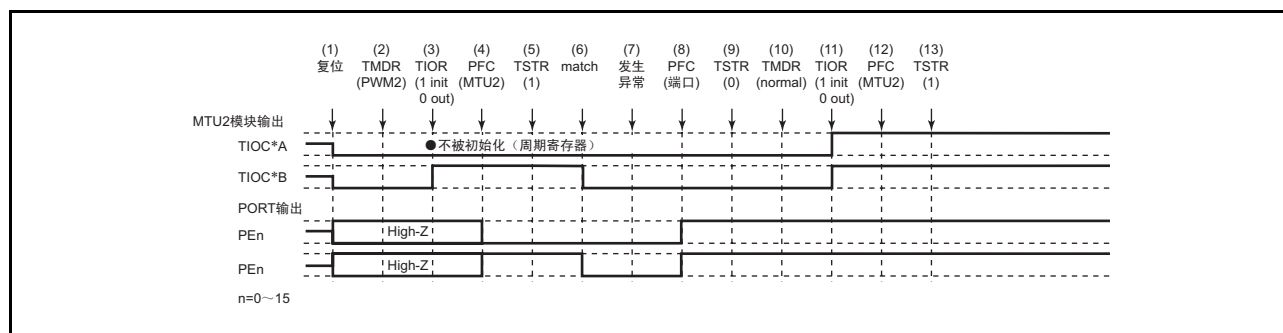


图 10.148 在 PWM 模式 2 中发生异常并在正常模式中恢复

- (1) 在复位后，MTU2 输出为低电平，端口为高阻抗状态。
- (2) 必须设定 PWM 模式 2。
- (3) 必须通过 TIOR 初始化引脚（在例中，初始输出为高电平，比较匹配为低电平输出，在 PWM 模式 2 中不初始化周期寄存器的引脚，TIOC*A 为周期寄存器）。
- (4) 必须通过 PFC 设定为 MTU2 输出。
- (5) 通过 TSTR 开始计数。
- (6) 通过产生比较匹配，输出低电平。
- (7) 发生了异常。
- (8) 必须通过 PFC 设定端口输出，输出有效电平的反相信号。
- (9) 通过 TSTR 停止计数。
- (10) 设定正常模式。
- (11) 必须通过 TIOR 初始化引脚。
- (12) 通过 PFC 设定为 MTU2 输出。
- (13) 通过 TSTR 重新开始。

(14) 在 PWM 模式 2 的运行中发生异常并在 PWM 模式 1 中重新开始时的运行

在 PWM 模式 2 中发生异常，重新设定后在 PWM 模式 1 中重新开始的说明如图 10.149 所示。

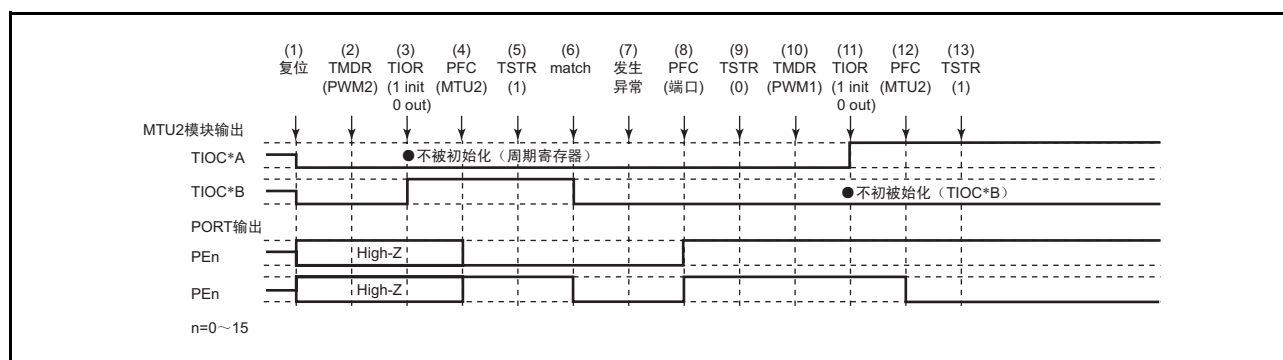


图 10.149 在 PWM 模式 2 中发生异常并在 PWM 模式 1 中恢复

- (1) ~ (9) 与图 10.148 相同。
- (10) 设定 PWM 模式 1。
- (11) 必须通过 TIOR 初始化引脚（在 PWM 模式 1 中不初始化 TIOC*B）。
- (12) 必须通过 PFC 设定为 MTU2 输出。
- (13) 通过 TSTR 重新开始。

(15) 在 PWM 模式 2 的运行中发生异常并在 PWM 模式 2 中重新开始时的运行

在 PWM 模式 2 中发生异常，重新设定后在 PWM 模式 2 中重新开始时的说明如图 10.150 所示

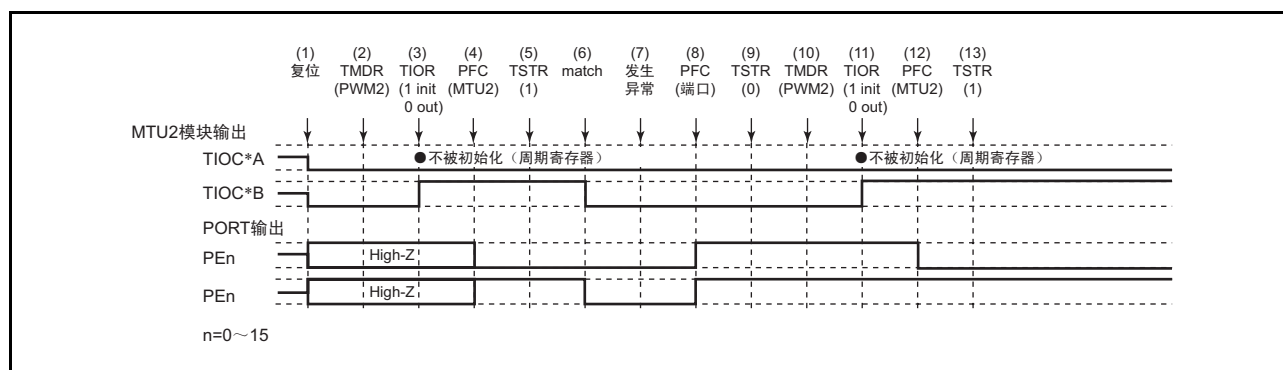


图 10.150 在 PWM 模式 2 中发生异常并在 PWM 模式 2 中恢复

(1) ~ (9) 与图 10.148 相同。

(10) 在 PWM 模式 2 中重新开始时不需要此步骤。

(11) 必须通过 TIOR 初始化引脚 (在 PWM 模式 2 中不初始化周期寄存器的引脚)。

(12) 必须通过 PFC 设定为 MTU2 输出。

(13) 通过 TSTR 重新开始。

(16) 在 PWM 模式 2 的运行中发生异常并在相位计数模式中重新开始时的运行

在 PWM 模式 2 中发生异常，重新再设定后在位相计数模式中重新开始时的说明如图 10.151 所示。

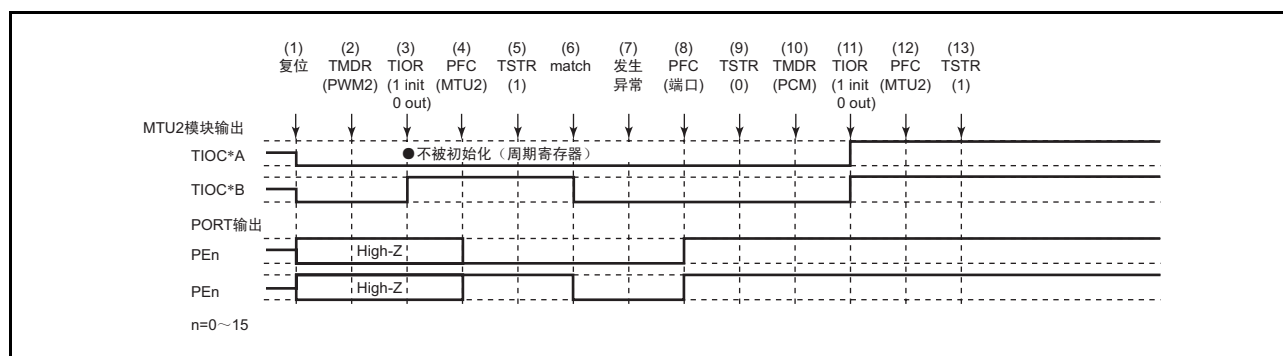


图 10.151 在 PWM 模式 2 中发生异常并在相位计数模式中恢复

(1) ~ (9) 与图 10.148 相同。

(10) 设定相位计数模式。

(11) 必须通过 TIOR 初始化引脚。

(12) 必须通过 PFC 设定为 MTU2 输出。

(13) 通过 TSTR 重新开始。

(17) 在相位计数模式的运行中发生异常并在正常模式中重新开始时的运行

在相位计数模式中发生异常，重新设定后在正常模式中重新开始时的说明如图 10.152 所示。

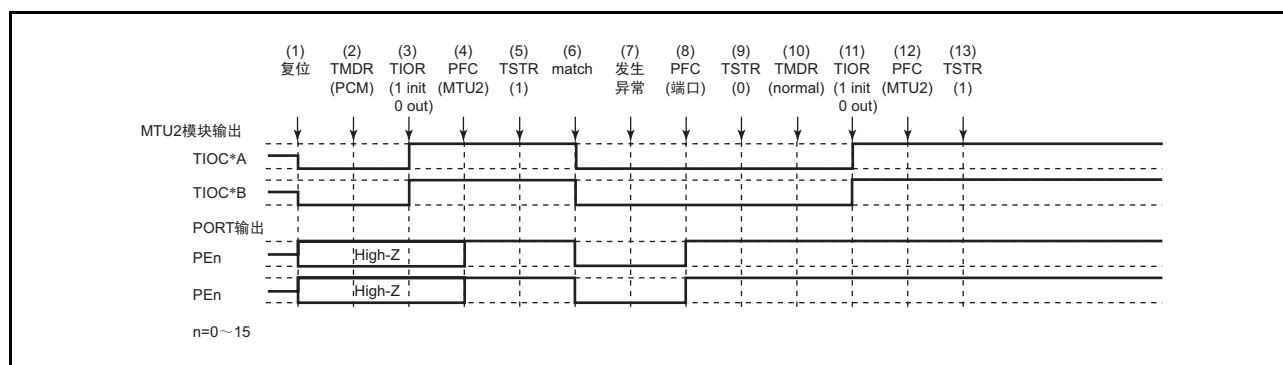


图 10.152 在相位计数模式中发生异常并在正常模式中恢复

- (1) 在复位后，MTU2 输出为低电平，端口为高阻抗状态。
- (2) 设定相位计数模式。
- (3) 必须通过 TIOR 初始化引脚（在例中，初始输出为高电平，比较输出为低电平输出）。
- (4) 必须通过 PFC 设定为 MTU2 输出。
- (5) 通过 TSTR 开始计数。
- (6) 通过产生比较匹配，输出低电平。
- (7) 发生了异常。
- (8) 必须通过 PFC 设定为端口输出，输出有效电平的反相信号。
- (9) 通过 TSTR 停止计数。
- (10) 在正常模式中进行设定。
- (11) 必须通过 TIOR 初始化引脚。
- (12) 必须通过 PFC 设定为 MTU2 输出。
- (13) 通过 TSTR 重新开始。

(18) 在相位计数模式的运行中发生异常并在 PWM 模式 1 中重新开始时的运行

在相位计数模式中发生异常，重新设定后在 PWM 模式 1 中重新开始时的说明如图 10.153 所示。

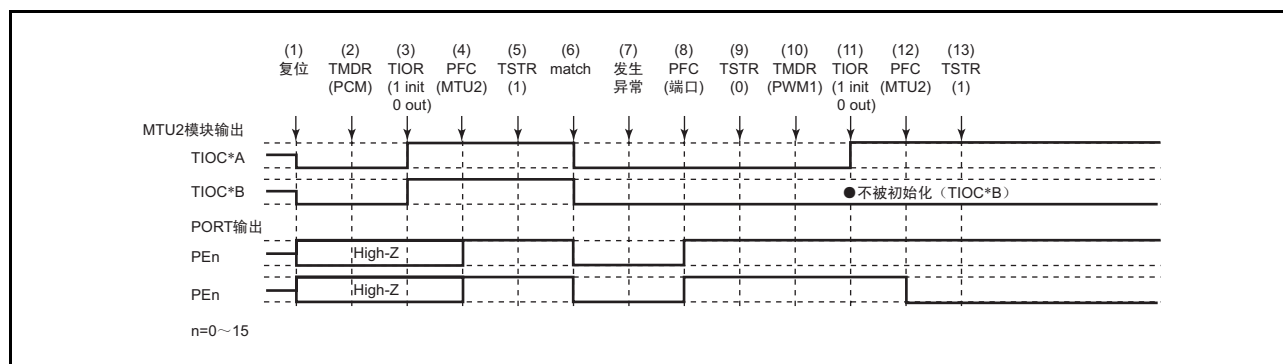


图 10.153 在相位计数模式中发生异常并在 PWM 模式 1 中恢复

- (1) ~ (9) 与图 10.152 相同。
- (10) 设定 PWM 模式 1。

- (11) 必须通过 TIOR 初始化引脚 (在 PWM 模式 1 中不初始化 TIOC*B)。
- (12) 必须通过 PFC 设定为 MTU2 输出。
- (13) 通过 TSTR 重新开始。

(19) 在相位计数模式的运行中发生异常并在 PWM 模式 2 中重新开始时的运行

在相位计数模式中发生异常, 重新设定后在 PWM 模式 2 中重新开始时的说明如图 10.154 所示。

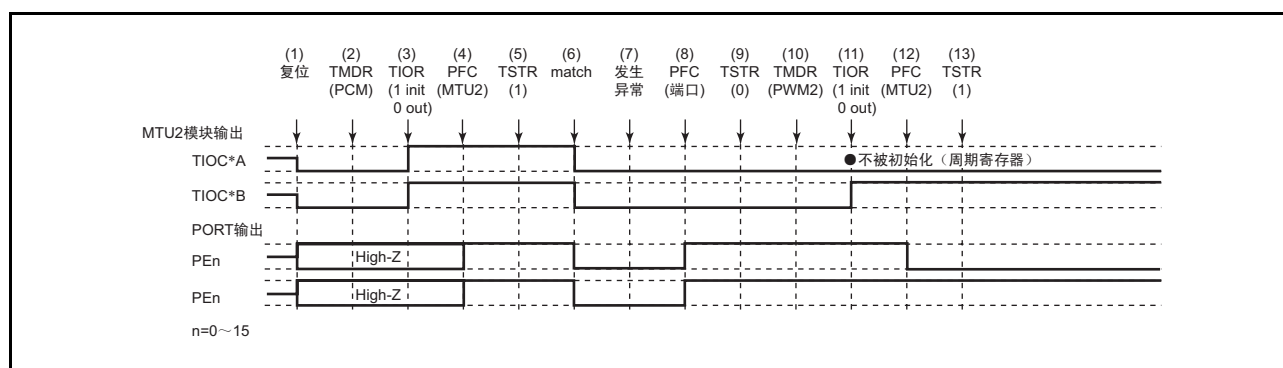


图 10.154 在相位计数模式中发生异常并在 PWM 模式 2 中恢复

- (1) ~ (9) 与图 10.152 相同。
- (10) 设定 PWM 模式 2。
- (11) 必须通过 TIOR 初始化引脚 (在 PWM 模式 2 中不初始化周期寄存器的引脚)。
- (12) 必须通过 PFC 设定为 MTU2 输出。
- (13) 通过 TSTR 重新开始。

(20) 在相位计数模式的运行中发生异常并在相位计数模式中重新开始时的运行

在相位计数模式中发生异常, 重新设定后在相位计数模式中重新开始时的说明如图 10.155 所示。

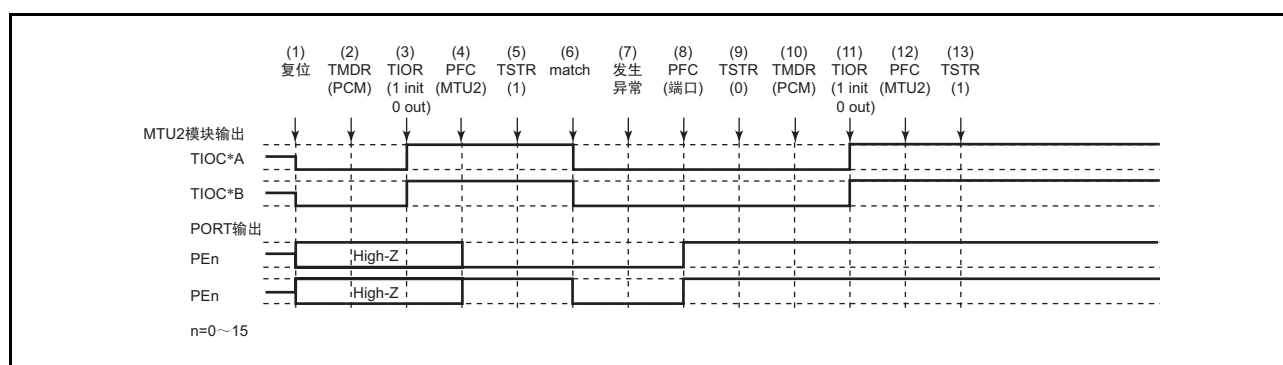


图 10.155 在相位计数模式中发生异常并在相位计数模式中恢复

- (1) ~ (9) 与图 10.152 相同。
- (10) 在相位计数模式中重新开始时不需要此步骤。
- (11) 必须通过 TIOR 初始化引脚。
- (12) 必须通过 PFC 设定为 MTU2 输出。
- (13) 通过 TSTR 重新开始。

(21) 在互补 PWM 模式的运行中发生异常并在正常模式中重新开始时的运行

在互补 PWM 模式中发生异常，重新设定后在正常模式中重新开始时的说明如图 10.156 所示。

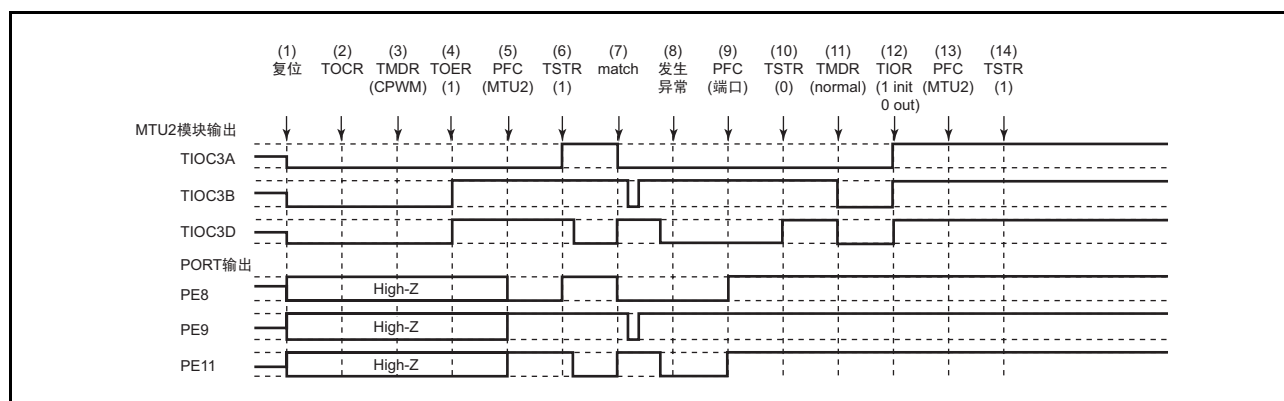


图 10.156 在互补 PWM 模式中发生异常并在正常模式中恢复

- (1) 在复位后，MTU2 输出为低电平，端口为高阻抗状态。
- (2) 必须通过 TOCR 选择互补 PWM 的输出电平并允许或者禁止周期输出。
- (3) 设定互补 PWM。
- (4) 必须通过 TOER 允许通道 3、4 的输出。
- (5) 必须通过 PFC 设定为 MTU2 输出。
- (6) 通过 TSTR 开始计数。
- (7) 通过产生比较匹配，输出互补 PWM 波形。
- (8) 发生了异常。
- (9) 必须通过 PFC 设定为端口输出，输出有效电平的反相信号。
- (10) 通过 TSTR 停止计数 (MTU2 输出为互补 PWM 输出的初始值)。
- (11) 必须设定正常模式 (MTU2 输出为低电平)。
- (12) 必须通过 TIOR 初始化引脚。
- (13) 必须通过 PFC 设定为 MTU2 输出。
- (14) 通过 TSTR 重新开始。

(22) 在互补 PWM 模式的运行中发生异常并在 PWM 模式 1 中重新开始时的运行

在互补 PWM 模式中发生异常，重新设定后在 PWM 模式 1 中重新开始时的说明如图 10.157 所示。

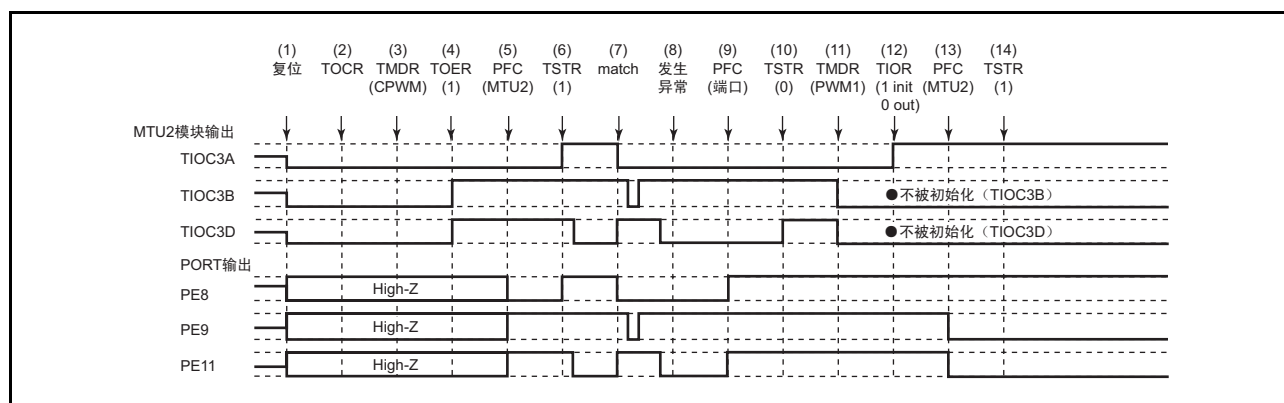


图 10.157 在互补 PWM 模式中发生异常并在 PWM 模式 1 中恢复

- (1) ~ (10) 与图 10.156 相同。
- (11) 必须设定 PWM 模式 1 (MTU2 输出为低电平)。
- (12) 必须通过 TIOR 初始化引脚 (在 PWM 模式 1 中不初始化 TIOC*B)。
- (13) 必须通过 PFC 设定为 MTU2 输出。
- (14) 通过 TSTR 重新开始。

(23) 在互补 PWM 模式的运行中发生异常并在互补 PWM 模式中重新开始时的运行

在互补 PWM 模式中发生异常, 重新设定后在互补 PWM 模式中重新开始 (从停止计数器时的周期和占空比的设定值重新开始) 时的说明如图 10.158 所示。

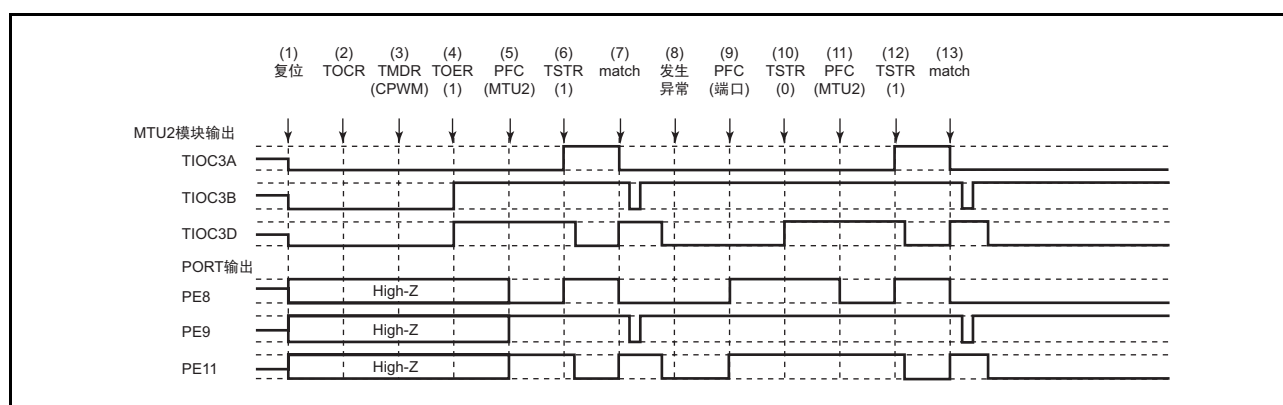


图 10.158 在互补 PWM 模式中发生异常并在互补 PWM 模式中恢复

- (1) ~ (10) 与图 10.156 相同。
- (11) 必须通过 PFC 设定为 MTU2 输出。
- (12) 通过 TSTR 重新开始。
- (13) 通过产生比较匹配, 输出互补 PWM 波形。

(24) 在互补 PWM 模式的运行中发生异常并在互补 PWM 模式中重新开始时的运行

在互补 PWM 模式中发生异常, 重新设定后在互补 PWM 模式中重新开始 (从周期和占空比的新设定值重新开始) 时的说明如图 10.159 所示。

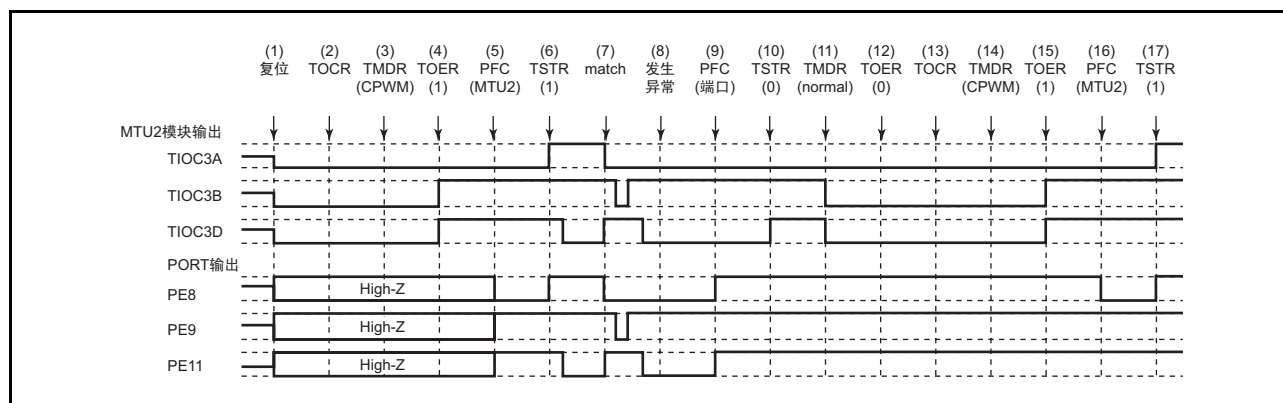


图 10.159 在互补 PWM 模式中发生异常并在互补 PWM 模式中恢复

- (1) ~ (10) 与图 10.156 相同。
- (11) 必须设定正常模式和新的设定值 (MTU2 输出为低电平)。
- (12) 必须通过 TOER 禁止通道 3、4 的输出。
- (13) 必须通过 TOCR 选择互补 PWM 模式的输出电平并允许或者禁止周期输出。
- (14) 设定互补 PWM。
- (15) 必须通过 TOER 允许通道 3、4 的输出。
- (16) 必须通过 PFC 设定为 MTU2 输出。
- (17) 通过 TSTR 重新开始。

(25) 在互补 PWM 模式的运行中发生异常并在复位同步 PWM 模式中重新开始时的运行

在互补 PWM 模式中发生异常，重新设定后在复位同步 PWM 模式中重新开始时的说明如图 10.160 所示。

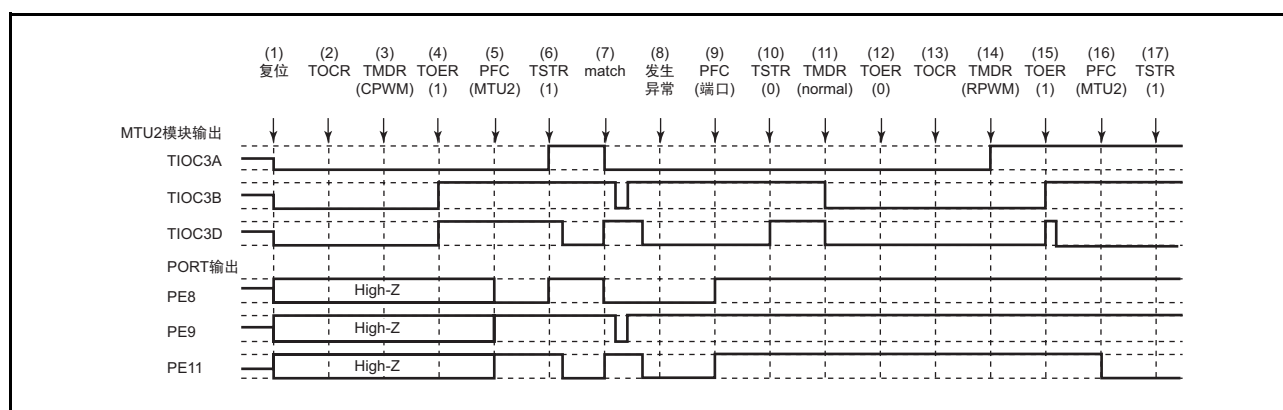


图 10.160 在互补 PWM 模式中发生异常并在复位同步 PWM 模式中恢复

- (1) ~ (10) 与图 10.156 相同。
- (11) 必须设定正常模式 (MTU2 输出为低电平)。
- (12) 必须通过 TOER 禁止通道 3、4 的输出。
- (13) 必须通过 TOCR 选择复位同步 PWM 模式的输出电平并允许或者禁止周期输出。
- (14) 设定复位同步 PWM。
- (15) 必须通过 TOER 允许通道 3、4 的输出。
- (16) 必须通过 PFC 设定为 MTU2 输出。
- (17) 通过 TSTR 重新开始。

(26) 在复位同步 PWM 模式的运行中发生异常并在正常模式中重新开始时的运行

在复位同步 PWM 模式中发生异常，重新设定后在正常模式中重新开始时的说明如图 10.161 所示。

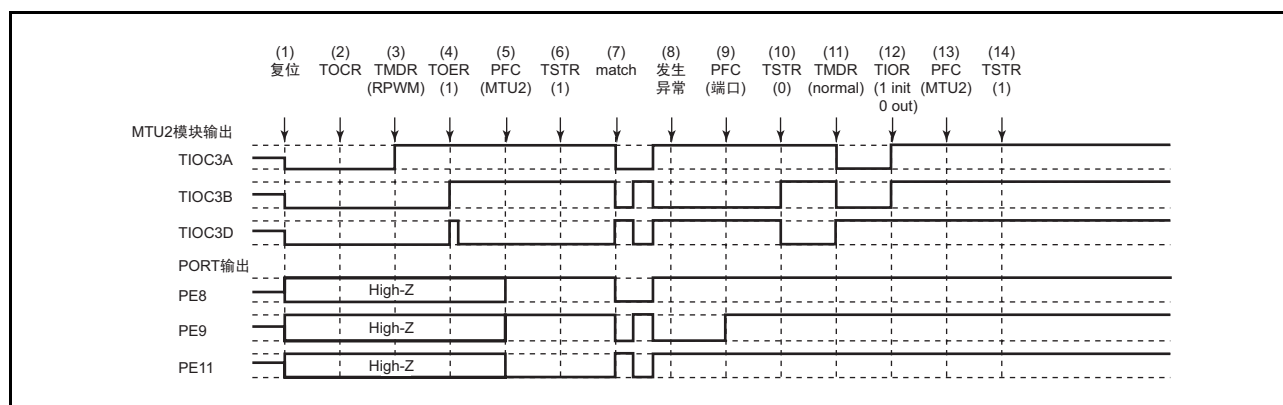


图 10.161 在复位同步 PWM 模式中发生异常并在正常模式中恢复

- (1) 在复位后，MTU2 输出为低电平，端口为高阻抗状态。
- (2) 必须通过 TOCR 选择复位同步 PWM 的输出电平并允许或者禁止周期输出。
- (3) 设定复位同步 PWM。
- (4) 必须通过 TOER 允许通道 3、4 的输出。
- (5) 必须通过 PFC 设定为 MTU2 输出。
- (6) 通过 TSTR 开始计数。
- (7) 通过产生比较匹配，输出复位同步 PWM 波形。
- (8) 发生了异常。
- (9) 必须通过 PFC 设定为端口输出，输出有效电平的反相信号。
- (10) 通过 TSTR 停止计数 (MTU2 输出为复位同步 PWM 输出的初始值)。
- (11) 必须设定正常模式 (MTU2 输出的正相为低电平，反相为高电平)。
- (12) 必须通过 TIOR 初始化引脚。
- (13) 必须通过 PFC 设定为 MTU2 输出。
- (14) 通过 TSTR 重新开始。

(27) 在复位同步 PWM 模式的运行中发生异常并在 PWM 模式 1 中重新开始时的运行

在复位同步 PWM 模式中发生异常，重新设定后在 PWM 模式 1 中重新开始时的说明如图 10.162 所示。

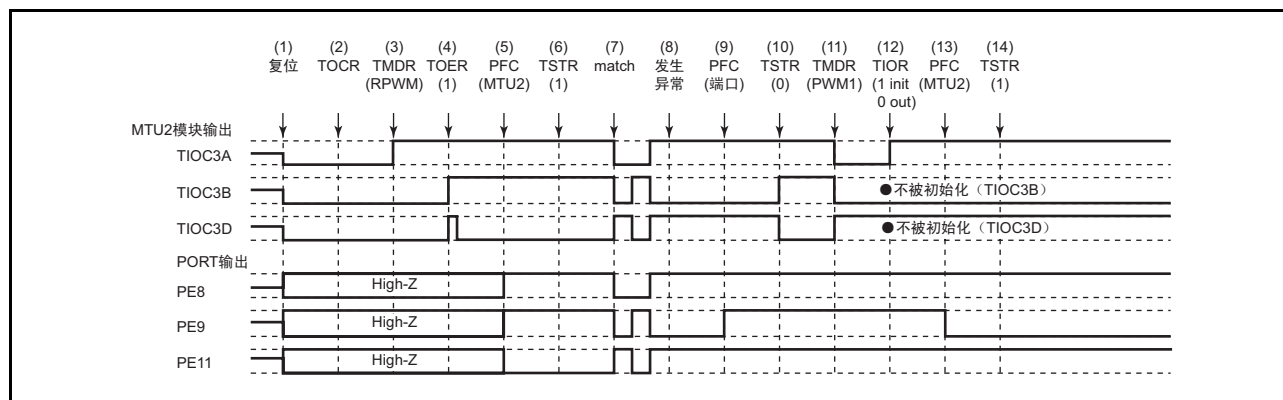


图 10.162 在复位同步 PWM 模式中发生异常并在 PWM 模式 1 中恢复

- (1) ~ (10) 与图 10.161 相同。
- (11) 必须设定 PWM 模式 1 (MTU2 输出的正相为低电平, 反相为高电平)。
- (12) 通过 TIOR 初始化引脚 (在 PWM 模式 1 中不初始化 TIOC*B)。
- (13) 必须通过 PFC 设定为 MTU2 输出。
- (14) 通过 TSTR 重新开始。

(28) 在复位同步 PWM 模式的运行中发生异常并在互补 PWM 模式中重新开始时的运行

在复位同步 PWM 模式中发生异常, 重新设定后在互补 PWM 模式中重新开始时的说明如图 10.163 所示。

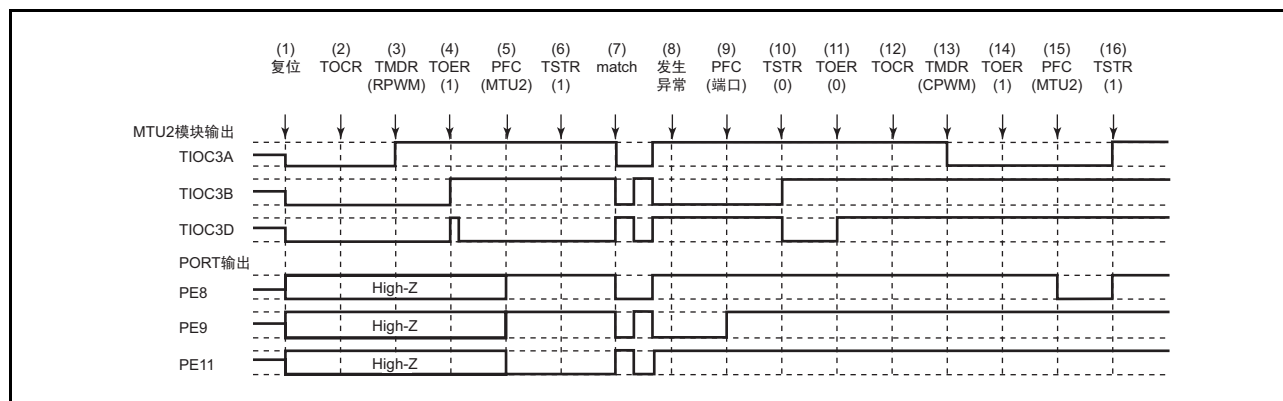


图 10.163 在复位同步 PWM 模式中发生异常并在互补 PWM 模式中恢复

- (1) ~ (10) 与图 10.161 相同。
- (11) 必须通过 TOER 禁止通道 3、4 的输出。
- (12) 必须通过 TOCR 选择互补 PWM 的输出电平并允许或者禁止周期输出。
- (13) 设定互补 PWM (MTU2 的周期输出引脚为低电平)。
- (14) 必须通过 TOER 允许通道 3、4 的输出。
- (15) 必须通过 PFC 设定为 MTU2 输出。
- (16) 通过 TSTR 重新开始。

(29) 在复位同步 PWM 模式的运行中发生异常并在复位同步 PWM 模式中重新开始时的运行

在复位同步 PWM 模式中发生异常, 重新设定后在复位同步 PWM 模式中重新开始时的说明如图 10.164 所示。

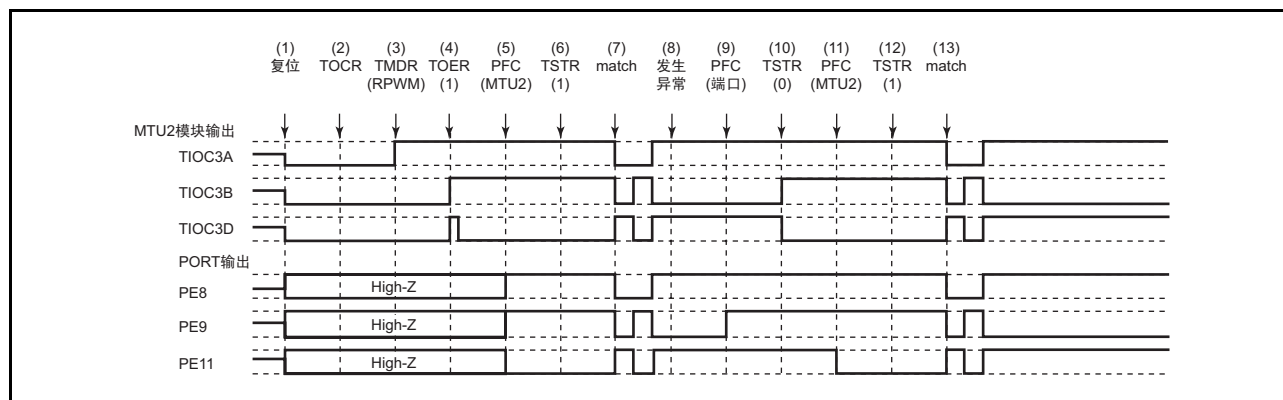


图 10.164 在复位同步 PWM 模式中发生异常并在复位同步 PWM 模式中恢复

- (1) ~ (10) 与图 10.161 相同。
- (11) 必须通过 PFC 设定为 MTU2 输出。
- (12) 通过 TSTR 重新开始。
- (13) 通过产生比较匹配，输出复位同步 PWM 波形。

第 11 章 多功能定时器脉冲单元 2S (MTU2S)

本 LSI 内置由 3 个通道 16 位定时器构成的多功能定时器脉冲单元 2S (MTU2S)，MTU2S 是内置 MTU2 的通道 3、4、5 的模块，详细内容参请照“第 10 章 多功能定时器脉冲单元 2 (MTU2)”。为了区别 MTU2，在输入 / 输出引脚名和寄存器名的末尾追加“S”。例如，TIOC3A 记为 TIOC3AS，TGRA_3 记为 TGRA_3S 等。

MTU2S 只在互补 PWM 模式的输出时才能以最大 80MHz 的工作频率运行，否则只能以最大 40MHz 的工作频率运行。

表 11.1 MTU2S 的功能一览表

项目	通道 3	通道 4	通道 5
计数时钟	MIφ/1 MIφ/4 MIφ/16 MIφ/64 MIφ/256 MIφ/1024	MIφ/1 MIφ/4 MIφ/16 MIφ/64 MIφ/256 MIφ/1024	MIφ/1 MIφ/4 MIφ/16 MIφ/64
通用寄存器 (TGR)	TGRA_3S TGRB_3S	TGRA_4S TGRB_4S	TGRU_5S TGRV_5S TGRW_5S
通用寄存器 / 缓冲寄存器	TGRC_3S TGRD_3S	TGRC_4S TGRD_4S	—
输入 / 输出引脚	TIOC3BS TIOC3DS	TIOC4AS TIOC4BS TIOC4CS TIOC4DS	输入引脚 TIC5US TIC5VS TIC5WS
计数器的清除功能	TGR 的比较匹配或者输入捕捉	TGR 的比较匹配或者输入捕捉	TGR 的比较匹配或者输入捕捉
比较匹配 的输出	0 输出	○	—
	1 输出	○	—
	交替输出	○	—
输入捕捉功能	○	○	○
同步运行	○	○	—
PWM 模式 1	○	○	—
PWM 模式 2	—	—	—
互补 PWM 模式	○	○	—
复位 PWM 模式	○	○	—
AC 同步马达驱动模式	—	—	—
相位计数模式	—	—	—
缓冲器运行	○	○	—
用于空载时间补偿 的计数器功能	—	—	○
DTC 启动	TGR 的比较匹配或者输入捕捉	TGR 的比较匹配或者输入捕捉 TCNT 的上溢 / 下溢	TGR 的比较匹配或者输入捕捉

项目	通道 3	通道 4	通道 5
A/D 转换的开始触发	TGRA_3S 的比较匹配或者输入捕捉	TGRA_4S 的比较匹配或者输入捕捉 在互补 PWM 模式中， TCNT_4S 的下溢（波谷）	—
中断源	5 个源 - 比较匹配 / 输入捕捉 3AS - 比较匹配 / 输入捕捉 3BS - 比较匹配 / 输入捕捉 3CS - 比较匹配 / 输入捕捉 3DS - 上溢	5 个源 - 比较匹配 / 输入捕捉 4AS - 比较匹配 / 输入捕捉 4BS - 比较匹配 / 输入捕捉 4CS - 比较匹配 / 输入捕捉 4DS - 上溢 / 下溢	3 个源 - 比较匹配 / 输入捕捉 5US - 比较匹配 / 输入捕捉 5VS - 比较匹配 / 输入捕捉 5WS
A/D 转换开始请求的延迟功能	—	- TADCORA_4S 和 TCNT_4S 相同时，进行 A/D 转换的开始请求 - TADCORB_4S 和 TCNT_4S 相同时，进行 A/D 转换的开始请求	—
中断减少功能	减少 TGRA_3S 的比较匹配中断	减少 TCIV_4S 中断	—

【符号说明】

○：能

—：不能

11.1 输入 / 输出引脚

表 11.2 引脚结构

通道	引脚名	输入 / 输出	功能
3	TIOC3BS	输入 / 输出	TGRB_3S 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	TIOC3DS	输入 / 输出	TGRD_3S 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
4	TIOC4AS	输入 / 输出	TGRA_4S 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	TIOC4BS	输入 / 输出	TGRB_4S 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	TIOC4CS	输入 / 输出	TGRC_4S 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
	TIOC4DS	输入 / 输出	TGRD_4S 的输入捕捉的输入引脚 / 输出比较的输出引脚 / PWM 的输出引脚
5	TIC5US	输入	TGRU_5S 的输入捕捉的输入引脚 / 外部脉冲的输入引脚
	TIC5VS	输入	TGRV_5S 的输入捕捉的输入引脚 / 外部脉冲的输入引脚
	TIC5WS	输入	TGRW_5S 的输入捕捉的输入引脚 / 外部脉冲的输入引脚

11.2 寄存器说明

MTU2S 的各通道有以下寄存器，有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 25 章 寄存器一览”。如同通道 3 的 TCR 记为 TCR_3S 一样，记载各通道的寄存器。

表 11.3 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
定时器的控制寄存器_3S	TCR_3S	R/W	H'00	H'FFFFC600	8、16、32
定时器的控制寄存器_4S	TCR_4S	R/W	H'00	H'FFFFC601	8
定时器的模式寄存器_3S	TMDR_3S	R/W	H'00	H'FFFFC602	8、16
定时器模式的寄存器_4S	TMDR_4S	R/W	H'00	H'FFFFC603	8
定时器的 I/O 控制寄存器 H_3S	TIORH_3S	R/W	H'00	H'FFFFC604	8、16、32
定时器的 I/O 控制寄存器 L_3S	TIORL_3S	R/W	H'00	H'FFFFC605	8
定时器的 I/O 控制寄存器 H_4S	TIORH_4S	R/W	H'00	H'FFFFC606	8、16
定时器的 I/O 控制寄存器 L_4S	TIORL_4S	R/W	H'00	H'FFFFC607	8
定时器的中断允许寄存器_3S	TIER_3S	R/W	H'00	H'FFFFC608	8、16
定时器的中断允许寄存器_4S	TIER_4S	R/W	H'00	H'FFFFC609	8
定时器的输出主控允许寄存器 S	TOERS	R/W	H'C0	H'FFFFC60A	8
定时器的门控制寄存器 S	TGCRS	R/W	H'80	H'FFFFC60D	8
定时器的输出控制寄存器 1S	TOCR1S	R/W	H'00	H'FFFFC60E	8、16
定时器的输出控制寄存器 2S	TOCR2S	R/W	H'00	H'FFFFC60F	8
定时器的计数器_3S	TCNT_3S	R/W	H'0000	H'FFFFC610	16、32
定时器的计数器_4S	TCNT_4S	R/W	H'0000	H'FFFFC612	16
定时器的周期数据寄存器 S	TCDRS	R/W	H'FFFF	H'FFFFC614	16、32
定时器的空载时间数据寄存器 S	TDDRS	R/W	H'FFFF	H'FFFFC616	16
定时器的通用寄存器 A_3S	TGRA_3S	R/W	H'FFFF	H'FFFFC618	16、32
定时器的通用寄存器 B_3S	TGRB_3S	R/W	H'FFFF	H'FFFFC61A	16
定时器的通用寄存器 A_4S	TGRA_4S	R/W	H'FFFF	H'FFFFC61C	16、32
定时器的通用寄存器 B_4S	TGRB_4S	R/W	H'FFFF	H'FFFFC61E	16
定时器的子计数器 S	TCNTSS	R	H'0000	H'FFFFC620	16、32
定时器的周期缓冲寄存器 S	TCBRS	R/W	H'FFFF	H'FFFFC622	16
定时器的通用寄存器 C_3S	TGRC_3S	R/W	H'FFFF	H'FFFFC624	16、32
定时器的通用寄存器 D_3S	TGRD_3S	R/W	H'FFFF	H'FFFFC626	16
定时器的通用寄存器 C_4S	TGRC_4S	R/W	H'FFFF	H'FFFFC628	16、32
定时器的通用寄存器 D_4S	TGRD_4S	R/W	H'FFFF	H'FFFFC62A	16
定时器的状态寄存器_3S	TSR_3S	R/W	H'C0	H'FFFFC62C	8、16
定时器的状态寄存器_4S	TSR_4S	R/W	H'C0	H'FFFFC62D	8
定时器的中断减少设定寄存器 S	TITCRS	R/W	H'00	H'FFFFC630	8、16
定时器的中断减少次数计数器 S	TITCNTS	R	H'00	H'FFFFC631	8
定时器的缓冲传送设定寄存器 S	TBTERS	R/W	H'00	H'FFFFC632	8
定时器的空载时间允许寄存器 S	TDERS	R/W	H'01	H'FFFFC634	8
定时器的输出电平缓冲寄存器 S	TOLBRS	R/W	H'00	H'FFFFC636	8

寄存器名	略称	R/W	初始值	地址	存取长度
定时器的缓冲器运行传送模式寄存器_3S	TBTM_3S	R/W	H'00	H'FFFFC638	8、16
定时器的缓冲器运行传送模式寄存器_4S	TBTM_4S	R/W	H'00	H'FFFFC639	8
定时器的 A/D 转换开始请求的控制寄存器 S	TADCRS	R/W	H'0000	H'FFFFC640	16
定时器的 A/D 转换开始请求的周期设定寄存器 A_4S	TADCORA_4S	R/W	H'FFFF	H'FFFFC644	16、32
定时器的 A/D 转换开始请求的周期设定寄存器 B_4S	TADCORB_4S	R/W	H'FFFF	H'FFFFC646	16
定时器的 A/D 转换开始请求的周期设定缓冲寄存器 A_4S	TADCOBRA_4S	R/W	H'FFFF	H'FFFFC648	16、32
定时器的 A/D 转换开始请求的周期设定缓冲寄存器 B_4S	TADCOBRB_4S	R/W	H'FFFF	H'FFFFC64A	16
定时器的同步清除寄存器 S	TSYCRS	R/W	H'00	H'FFFFC650	8
定时器的波形控制寄存器 S	TWCRS	R/W	H'00	H'FFFFC660	8
定时器的开始寄存器 S	TSTRS	R/W	H'00	H'FFFFC680	8、16
定时器的同步寄存器 S	TSYRS	R/W	H'00	H'FFFFC681	8
定时器的读写允许寄存器 S	TRWERS	R/W	H'01	H'FFFFC684	8
定时器的计数器 U_5S	TCNTU_5S	R/W	H'0000	H'FFFFC880	16、32
定时器的通用寄存器 U_5S	TGRU_5S	R/W	H'FFFF	H'FFFFC882	16
定时器的控制寄存器 U_5S	TCRU_5S	R/W	H'00	H'FFFFC884	8
定时器的 I/O 控制寄存器 U_5S	TIORU_5S	R/W	H'00	H'FFFFC886	8
定时器的计数器 V_5S	TCNTV_5S	R/W	H'0000	H'FFFFC890	16、32
定时器的通用寄存器 V_5S	TGRV_5S	R/W	H'FFFF	H'FFFFC892	16
定时器的控制寄存器 V_5S	TCRV_5S	R/W	H'00	H'FFFFC894	8
定时器的 I/O 控制寄存器 V_5S	TIORV_5S	R/W	H'00	H'FFFFC896	8
定时器的计数器 W_5S	TCNTW_5S	R/W	H'0000	H'FFFFC8A0	16、32
定时器的通用寄存器 W_5S	TGRW_5S	R/W	H'FFFF	H'FFFFC8A2	16
定时器的控制寄存器 W_5S	TCRW_5S	R/W	H'00	H'FFFFC8A4	8
定时器的 I/O 控制寄存器 W_5S	TIORW_5S	R/W	H'00	H'FFFFC8A6	8
定时器的状态寄存器_5S	TSR_5S	R/W	H'00	H'FFFFC8B0	8
定时器的中断允许寄存器_5S	TIER_5S	R/W	H'00	H'FFFFC8B2	8
定时器的开始寄存器_5S	TSTR_5S	R/W	H'00	H'FFFFC8B4	8
定时器的比较匹配清除寄存器 S	TCNTCMPCLRS	R/W	H'00	H'FFFFC8B6	8

第 12 章 端口输出的允许 (POE)

根据 $\overline{\text{POE0}} \sim \overline{\text{POE2}}$ 、 $\overline{\text{POE4}} \sim \overline{\text{POE6}}$ 、 $\overline{\text{POE8}}$ 引脚的输入变化、大电流引脚 (MTU2 的 TIOC3B、TIOC3D、TIOC4A、TIOC4B、TIOC4C、TIOC4D 以及 MTU2S 的 TIOC3BS、TIOC3DS、TIOC4AS、TIOC4BS、TIOC4CS、TIOC4DS 为多路复用引脚) 的输出状态或者寄存器的设定, 端口输出的允许 (POE) 能将大电流引脚和 MTU2 的 CH0 引脚 (TIOC0A、TIOC0B、TIOC0C、TIOC0D 为多路复用引脚) 置为高阻抗状态, 同时能产生传送中断请求。

12.1 特点

- 能给 $\overline{\text{POE0}} \sim \overline{\text{POE2}}$ 、 $\overline{\text{POE4}} \sim \overline{\text{POE6}}$ 和 $\overline{\text{POE8}}$ 的各输入引脚设定下降沿、 $P\phi/8 \times 16$ 次、 $P\phi/16 \times 16$ 次或者 $P\phi/128 \times 16$ 次的低电平采样。
- 能通过 $\overline{\text{POE0}} \sim \overline{\text{POE2}}$ 、 $\overline{\text{POE4}} \sim \overline{\text{POE6}}$ 和 $\overline{\text{POE8}}$ 引脚的下降沿或者低电平采样, 将大电流引脚和 MTU2 的 CH0 引脚置为高阻抗状态。
- 在和大电流引脚的输出电平进行比较的同时, 如果有效电平持续输出了 1 个周期以上, 就能将大电流引脚置为高阻抗状态。
- 能通过写 POE 的寄存器将大电流引脚和 MTU2 的 CH0 引脚置为高阻抗状态。
- 输入电平的采样或者输出电平的比较结果能分别产生中断。

如图 12.1 的框图所示, POE 由输入电平的检测电路、输出电平的比较电路和高阻抗请求 / 中断请求的生成电路构成。

不同的是: 即使在振荡器停止振荡或者软件待机状态下, 也能将大电流引脚置为高阻抗状态。详细内容请参照 “附录 A. 引脚状态”。

POE 的框图如图 12.1 所示。

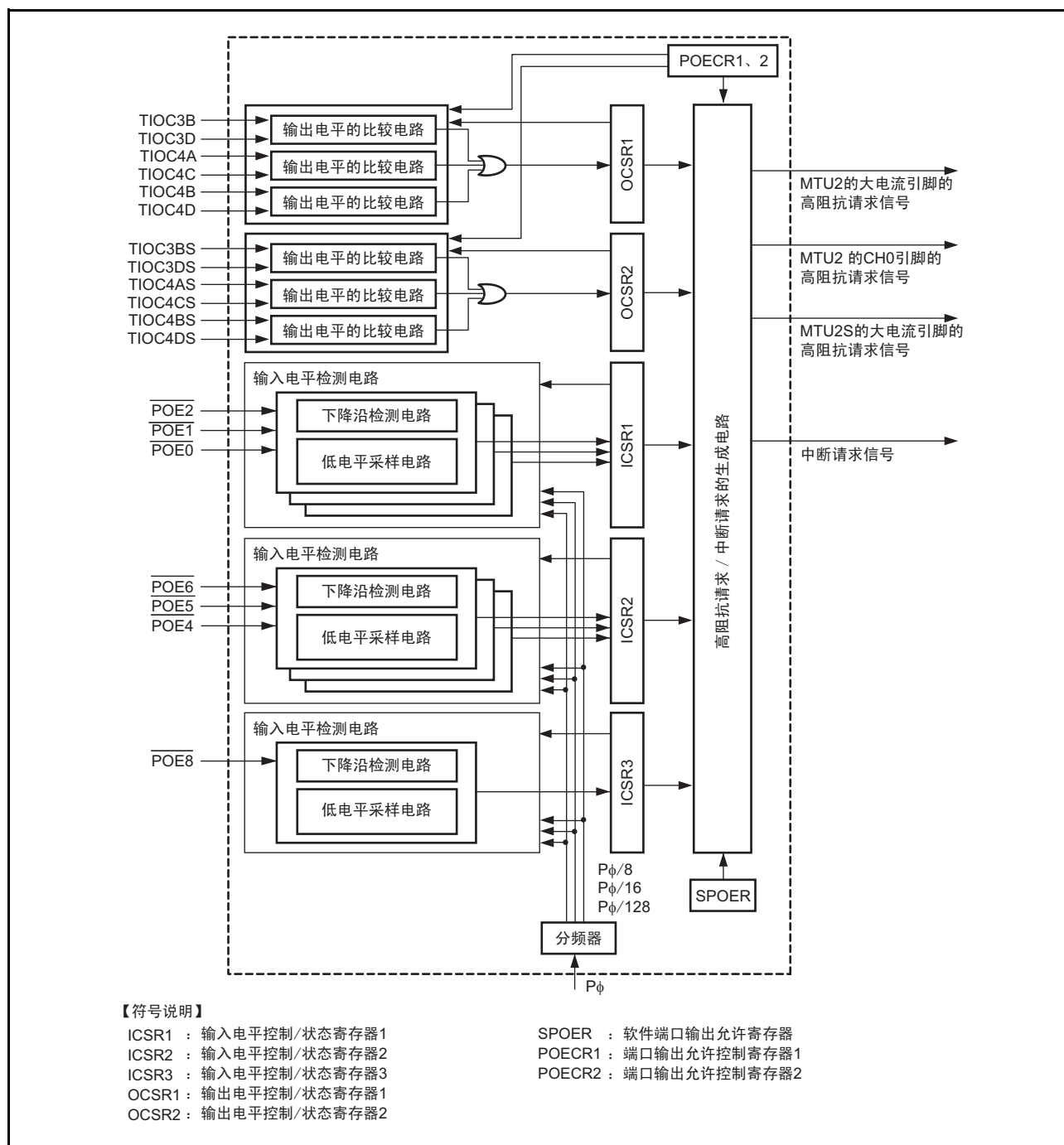


图 12.1 POE 的框图

12.2 输入 / 输出引脚

表 12.1 引脚结构

名称	引脚名	输入 / 输出	功能
端口输出允许的 输入引脚 0 ~ 2	POE0 ~ POE2	输入	输入将 MTU2 的大电流引脚置为高阻抗状态的请求信号。
端口输出允许的 输入引脚 4 ~ 6	POE4 ~ POE6	输入	输入将 MTU2S 的大电流引脚置为高阻抗状态的请求信号。
端口输出允许的 输入引脚 8	POE8	输入	输入将 MTU2 的 CH0 引脚置为高阻抗状态的请求信号。

通过表 12.2 所示的引脚组合进行输出电平的比较。

表 12.2 引脚组合

引脚组合	输入 / 输出	功能
PE9/TIOC3B 和 PE11/TIOC3D	输出	当 2 个引脚的有效电平 (MTU2 定时器的输出控制寄存器 (TOCR) 的输出电平选择 P 位 (OLSP) 是 0 时为低电平输出, 是 1 时为高电平输出) 至少持续输出了 1 个外围时钟 (P ϕ) 周期时, 将 MTU2 的大电流引脚置为高阻抗状态。当通过设定引脚功能控制器选择 MTU2 的输出功能或者通用输出功能时, 进行上述有效电平的比较, 否则不进行比较。 能通过 POE 的寄存器设定对哪个组合进行输出比较以及是否进行高阻抗控制。
PE12/TIOC4A 和 PE14/TIOC4C	输出	
PE13/TIOC4B 和 PE15/TIOC4D	输出	
PE16/TIOC3BS 和 PE17/TIOC3DS	输出	当 2 个引脚的有效电平 (MTU2S 定时器的输出控制寄存器 (TOCR) 的输出电平选择 P 位 (OLSP) 是 0 时为低电平输出, 是 1 时为高电平输出) 至少持续输出了 1 个外围时钟 (P ϕ) 周期时, 将 MTU2S 的大电流引脚置为高阻抗状态。当通过设定引脚功能控制器选择 MTU2S 的输出功能或者通用输出功能时, 进行上述有效电平的比较, 否则不进行比较。 能通过 POE 的寄存器设定对哪个组合进行输出比较以及是否进行高阻抗控制。
PE18/TIOC4AS 和 PE20/TIOC4CS	输出	
PE19/TIOC4BS 和 PE21/TIOC4DS	输出	

12.3 寄存器说明

POE 有以下寄存器。有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 25 章 寄存器一览”。

表 12.3 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
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

12.3.1 输入电平控制 / 状态寄存器 1 (ICSR1)

ICSR1 是 16 位可读写寄存器，选择 POE0 ~ POE2 引脚的输入模式、控制中断的允许 / 禁止以及表示各状态。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	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，只能在读1后写0。

*2 只能在上电复位后写1次。

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	POE2F	0	R/(W)*1	POE2 标志位 表示给 POE2 引脚输入了高阻抗请求。 [清除条件] - 当读 POE2F=1 后给 POE2F 写 0 时（当通过 ICSR1 的 bit5 和 bit4 设定为下降沿时） - 在通过 Pφ/8、16、128 时钟对 POE2 输入引脚进行高电平采样时，当读 POE2F=1 后给 POE2F 写 0 时（当通过 ICSR1 的 bit5 和 bit4 设定为低电平采样时） [置位条件] - 当 POE2 引脚有 ICSR1 的 bit5 和 bit4 设定的输入时

位	位名	初始值	R/W	说明
13	POE1F	0	R/(W)*1	POE1 标志位 表示给 $\overline{\text{POE1}}$ 引脚输入了高阻抗请求。 [清除条件] - 当读 POE1F=1 后给 POE1F 写 0 时 (当通过 ICSR1 的 bit3 和 bit2 设定为下降沿时) - 在通过 $P\phi/8$、16、128 时钟对 POE1 输入引脚进行高电平采样时, 当读 POE1F=1 后给 POE1F 写 0 时 (当通过 ICSR1 的 bit3 和 bit2 设定为低电平采样时) [置位条件] - 当 POE1 引脚有 ICSR1 的 bit3 和 bit2 设定的输入时
12	POE0F	0	R/(W)*1	POE0 标志位 表示给 $\overline{\text{POE0}}$ 引脚输入了高阻抗请求。 [清除条件] - 当读 POE0F=1 后给 POE0F 写 0 时 (当通过 ICSR1 的 bit1 和 bit0 设定为下降沿时) - 在通过 $P\phi/8$、16、128 时钟对 POE0 输入引脚进行高电平采样时, 当读 POE0F=1 后给 POE0F 写 0 时 (当通过 ICSR1 的 bit1 和 bit0 设定为低电平采样时) [置位条件] - 当 POE0 引脚有 ICSR1 的 bit1 和 bit0 设定的输入时
11 ~ 9	—	全 0	R	保留位 读写值总是 0。
8	PIE1	0	R/W	端口中断允许位 1 指定在 ICSR1 的 POE0F ~ POE2F 位中有 1 位被置 1 时是否请求中断。 0: 禁止中断请求 1: 允许中断请求
7、6	—	全 0	R	保留位 读写值总是 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 输入引脚的下降沿接受请求。 01: 通过 $P\phi/8$ 时钟对 POE2 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。 10: 通过 $P\phi/16$ 时钟对 POE2 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。 11: 通过 $P\phi/128$ 时钟对 POE2 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。

位	位名	初始值	R/W	说明
3、2	POE1M[1:0]	00	R/W*2	POE1 模式位 1、0 选择 POE1 引脚的输入模式。 00: 在 POE1 输入的下降沿接受请求。 01: 通过 P ϕ /8 时钟对 POE1 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。 10: 通过 P ϕ /16 时钟对 POE1 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。 11: 通过 P ϕ /128 时钟对 POE1 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。
1、0	POE0M[1:0]	00	R/W*2	POE0 模式位 1、0 选择 POE0 引脚的输入模式。 00: 在 POE0 输入的下降沿接受请求。 01: 通过 P ϕ /8 时钟对 POE0 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。 10: 通过 P ϕ /16 时钟对 POE0 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。 11: 通过 P ϕ /128 时钟对 POE0 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。

【注】 *1 为了清除标志, 只能在读 1 后写 0。

*2 只能在上电复位后写 1 次。

12.3.2 输出电平控制 / 状态寄存器 1 (OCSR1)

OCSR1 是 16 位可读写寄存器, 允许或者禁止输出电平的比较、控制中断的允许 / 禁止以及表示各状态。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSF1	—	—	—	—	—	OCE1	OIE1	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R	R	R	R	R	R/W*2	R/W	R	R	R	R	R	R	R	R

【注】 *1 为了清除标志, 只能在读 1 后写 0。

*2 只能在上电复位后写 1 次。

位	位名	初始值	R/W	说明
15	OSF1	0	R/(W)*1	输出短路的标志位 1 表示在 MTU2 要比较的 3 组 2 相输出中至少有 1 组同时为有效电平。 [清除条件] • 当读 OSF1=1 后给 OSF1 写 0 时 [置位条件] • 当 3 组 2 相输出中至少有 1 组同时为有效电平时
14 ~ 10	—	全 0	R	保留位 读写值总是 0。
9	OCE1	0	R/W*2	输出短路的高阻抗允许位 1 指定在 OCSR1 的 OSF1 位被置位时是否将引脚置为高阻抗状态。 0: 不将引脚置为高阻抗状态。 1: 将引脚置为高阻抗状态。

位	位名	初始值	R/W	说明
8	OIE1	0	R/W	输出短路的中断允许位 1 指定在 OCSR1 的 OSF1 位被置位时是否请求中断。 0: 禁止中断请求 1: 允许中断请求
7 ~ 0	—	全 0	R	保留位 读写值总是 0。

12.3.3 输入电平控制 / 状态寄存器 2 (ICSR2)

ICSR2 是 16 位可读写寄存器, 选择 $\overline{\text{POE4}} \sim \overline{\text{POE6}}$ 引脚的输入模式、控制中断的允许 / 禁止以及表示各状态。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	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, 只能在读1后写0。

*2 只能在上电复位后写1次。

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	POE6F	0	R/(W)*1	POE6 标志位 表示给 $\overline{\text{POE6}}$ 引脚输入了高阻抗请求 [清除条件] - 当读 POE6F=1 后给 POE6F 写 0 时 (当通过 ICSR2 的 bit5 和 bit4 设定为下降沿时) - 在通过 Pϕ/8、16、128 时钟对 POE6 输入引脚进行高电平采样时, 当读 POE6F=1 后给 POE6F 写 0 时 (当通过 ICSR2 的 bit5 和 bit4 设定为低电平采样时) [置位条件] - 当 $\overline{\text{POE6}}$ 引脚有 ICSR2 的 bit5 和 bit4 设定的输入时
13	POE5F	0	R/(W)*1	POE5 标志位 表示给 $\overline{\text{POE5}}$ 引脚输入了高阻抗请求。 [清除条件] - 当读 POE5F=1 后给 POE5F 写 0 时 (当通过 ICSR2 的 bit3 和 bit2 设定为下降沿时) - 在通过 Pϕ/8、16、128 时钟对 POE5 输入引脚进行高电平采样时, 当读 POE5F=1 后给 POE5F 写 0 时 (当通过 ICSR2 的 bit3 和 bit2 设定为低电平采样时) [置位条件] - 当 $\overline{\text{POE5}}$ 引脚有 ICSR2 的 bit3 和 bit2 设定的输入时

位	位名	初始值	R/W	说明
12	POE4F	0	R/(W)*1	POE4 标志位 表示给 POE4 引脚输入了高阻抗请求。 [清除条件] - 当读 POE4F=1 后给 POE4F 写 0 时 (当通过 ICSR2 的 bit1 和 bit0 设定为下降沿时) - 在通过 Pϕ/8、16、128 时钟对 POE4 输入引脚进行高电平采样时, 当读 POE4F=1 后给 POE4F 写 0 时 (当通过 ICSR2 的 bit1 和 bit0 设定为低电平采样时) [置位条件] - 当 POE4 引脚有 ICSR2 的 bit1 和 bit0 设定的输入时
11 ~ 9	—	全 0	R	保留位 读写值总是 0。
8	PIE2	0	R/W	端口中断允许位 2 指定在 ICSR2 的 POE4F ~ POE6F 位中有 1 位被置 1 时是否请求中断。 0: 禁止中断请求 1: 允许中断请求
7、6	—	全 0	R	保留位 读写值总是 0。
5、4	POE6M[1:0]	00	R/W*2	POE6 模式位 1、0 选择 POE6 引脚的输入模式。 00: 在 POE6 输入的下降沿接受请求 01: 通过 Pϕ/8 时钟对 POE6 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。 10: 通过 Pϕ/16 时钟对 POE6 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。 11: 通过 Pϕ/128 时钟对 POE6 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。
3、2	POE5M[1:0]	00	R/W*2	POE5 模式位 1、0 选择 POE5 引脚的输入模式。 00: 在 POE5 输入的下降沿接受请求。 01: 通过 Pϕ/8 时钟对 POE5 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。 10: 通过 Pϕ/16 时钟对 POE5 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。 11: 通过 Pϕ/128 时钟对 POE5 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。
1、0	POE4M[1:0]	00	R/W*2	POE4 模式位 1、0 选择 POE4 引脚的输入模式。 00: 在 POE4 输入的下降沿接受请求。 01: 通过 Pϕ/8 时钟对 POE4 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。 10: 通过 Pϕ/16 时钟对 POE4 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。 11: 通过 Pϕ/128 时钟对 POE4 输入引脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。

【注】 *1 为了清除标志, 只能在读 1 后写 0。

*2 只能在上电复位后写 1 次。

12.3.4 输出电平控制 / 状态寄存器 2 (OCSR2)

OCSR2 是 16 位可读写寄存器，允许或者禁止输出电平的比较、控制中断的允许 / 禁止以及表示各状态。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSF2	—	—	—	—	—	OCE2	OIE2	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R	R	R	R	R	R/W*2	R/W	R	R	R	R	R	R	R	R

【注】*1 为了清除标志，只能在读 1 后写 0。

*2 只能在上电复位后写 1 次。

位	位名	初始值	R/W	说明
15	OSF2	0	R/(W)*1	输出短路的标志位 2 表示在 MTU2S 要比较的 3 组 2 相输出中至少有 1 组同时为有效电平。 [清除条件] • 当读 OSF2=1 后给 OSF2 写 0 时 [置位条件] • 当 3 组 2 相输出中至少有 1 组同时为有效电平时
14 ~ 10	—	全 0	R	保留位 读写值总是 0。
9	OCE2	0	R/W*2	输出短路的高阻抗允许位 2 指定在 OCSR2 的 OSF2 位被置位时是否将引脚置为高阻抗状态。 0: 不将引脚置为高阻抗状态。 1: 将引脚置为高阻抗状态。
8	OIE2	0	R/W	输出短路的中断允许位 2 指定在 OCSR2 的 OSF2 位被置位时是否请求中断。 0: 禁止中断请求 1: 允许中断请求
7 ~ 0	—	全 0	R	保留位 读写值总是 0。

【注】*1 为了清除标志，只能在读 1 后写 0。

*2 只能在上电复位后写 1 次。

12.3.5 输入电平控制 / 状态寄存器 3 (ICSR3)

ICSR3 是 16 位可读写寄存器，选择 POE8 引脚的输入模式、控制中断的允许 / 禁止以及表示各状态。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	POE8F	—	—	POE8E	PIE3	—	—	—	—	—	—	POE8M[1:0]	
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/(W)*1	R	R	R/W*2	R/W	R	R	R	R	R	R	R/W*2	R/W*2

【注】*1 为了清除标志，只能在读1后写0。

*2 只能在上电复位后写1次。

位	位名	初始值	R/W	说明
15 ~ 13	—	全 0	R	保留位 读写值总是 0。
12	POE8F	0	R/(W)*1	POE8 的标志位 表示给 POE8 引脚输入了高阻抗请求。 [清除条件] - 当读 POE8F=1 后给 POE8F 写 0 时（当通过 ICSR3 的 bit1 和 bit0 设定为下降沿时） - 在通过 Pϕ/8、16、128 时钟对 POE8 输入引脚进行高电平采样时，当读 POE8F=1 后给 POE8F 写 0 时（当通过 ICSR3 的 bit1 和 bit0 设定为低电平采样时） [置位条件] - 当 POE8 引脚有 ICSR3 的 bit1 和 bit0 设定的输入时
11、10	—	全 0	R	保留位 读写值总是 0。
9	POE8E	0	R/W*2	POE8 的高阻抗允许位 指定在 ICSR3 的 POE8F 位被置位时是否将引脚置为高阻抗状态。 0: 不将引脚置为高阻抗状态。 1: 将引脚置为高阻抗状态。
8	PIE3	0	R/W	端口的中断允许位 3 指定在 ICSR3 的 POE8 位被置 1 时是否请求中断。 0: 禁止中断请求 1: 允许中断请求
7 ~ 2	—	全 0	R	保留位 读写值总是 0。
1、0	POE8M [1:0]	00	R/W*2	POE8 模式位 1、0 选择 POE8 引脚的输入模式。 00: 在 POE8 输入的下降沿接受请求。 01: 通过 P ϕ /8 时钟对 POE8 输入引脚进行 16 次低电平采样，当全部为低电平时，接受请求。 10: 通过 P ϕ /16 时钟对 POE8 输入引脚进行 16 次低电平采样，当全部为低电平时，接受请求。 11: 通过 P ϕ /128 时钟对 POE8 输入引脚进行 16 次低电平采样，当全部为低电平时，接受请求。

【注】*1 为了清除标志，只能在读 1 后写 0。

*2 只能在上电复位后写 1 次。

12.3.6 软件端口的输出允许寄存器 (SPOER)

SPOER 是 8 位可读写寄存器，对引脚进行高阻抗控制。

位:	7	6	5	4	3	2	1	0
	—	—	—	—	—	MTU2S HIZ	MTU2 CH0HIZ	MTU2 CH34HIZ
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7 ~ 3	—	全 0	R	保留位 读写值总是 0。
2	MTU2SHIZ	0	R/W	MTU2S 的输出高阻抗 对 MTU2S 的大电流引脚进行高阻抗控制。 0: 不置为高阻抗状态。 [清除条件] • 上电复位时 • 当读 MTU2SHIZ=1 后给 MTU2SHIZ 写 0 时 1: 置为高阻抗状态。 [置位条件] • 当给 MTU2SHIZ 写 1 时
1	MTU2CH0HIZ	0	R/W	MTU2 CH0 的输出高阻抗 对 MTU2 的 CH0 引脚进行高阻抗控制。 0: 不置为高阻抗状态。 [清除条件] • 上电复位时 • 当读 MTU2CH0HIZ=1 后给 MTU2CH0HIZ 写 0 时 1: 置为高阻抗状态。 [置位条件] • 当给 MTU2CH0HIZ 写 1 时
0	MTU2CH34HIZ	0	R/W	MTU2 CH3、4 的输出高阻抗 对 MTU2 的大电流引脚进行高阻抗控制。 0: 不置为高阻抗状态。 [清除条件] • 上电复位时 • 当读 MTU2CH34HIZ=1 后给 MTU2CH34HIZ 写 0 时 1: 置为高阻抗状态。 [置位条件] • 当给 MTU2CH34HIZ 写 1 时

12.3.7 端口输出的允许控制寄存器 1（POECR1）

POECR1 是 8 位可读写寄存器，对引脚进行高阻抗控制。

位:	7	6	5	4	3	2	1	0
	—	—	—	—	MTU2 PE3ZE	MTU2 PE2ZE	MTU2 PE1ZE	MTU2 PE0ZE
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W*	R/W*	R/W*	R/W*

【注】* 只能在上电复位后写1次。

位	位名	初始值	R/W	说明
7 ~ 4	—	全 0	R	保留位 读写值总是 0。
3	MTU2PE3ZE	0	R/W*	MTU2 PE3 的高阻抗允许位 设定在 POE8F 和 MTU2CH0HIZ 位中有 1 位被置位时，是否将 MTU2 的 CH0 引脚 PE3/TIOC0D 置为高阻抗状态。 0: 不置为高阻抗状态。 1: 置为高阻抗状态。
2	MTU2PE2ZE	0	R/W*	MTU2 PE2 的高阻抗允许位 设定在 POE8F 和 MTU2CH0HIZ 位中有 1 位被置位时，是否将 MTU2 的 CH0 引脚 PE2/TIOC0C 置为高阻抗状态。 0: 不置为高阻抗状态。 1: 置为高阻抗状态。
1	MTU2PE1ZE	0	R/W*	MTU2 PE1 的高阻抗允许位 设定在 POE8F 和 MTU2CH0HIZ 位中有 1 位被置位时，是否将 MTU2 的 CH0 引脚 PE1/TIOC0B 置为高阻抗状态。 0: 不置为高阻抗状态。 1: 置为高阻抗状态。
0	MTU2PE0ZE	0	R/W*	MTU2 PE0 的高阻抗允许位 设定在 POE8F 和 MTU2CH0HIZ 位中有 1 位被置位时，是否将 MTU2 的 CH0 引脚 PE0/TIOC0A 置为高阻抗状态。 0: 不置为高阻抗状态。 1: 置为高阻抗状态。

12.3.8 端口的输出的允许控制寄存器 2（POECR2）

POECR2 是 16 位可读写寄存器，对引脚进行高阻抗控制。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	MTU2 P1CZE	MTU2 P2CZE	MTU2 P3CZE	—	MTU2S P1CZE	MTU2S P2CZE	MTU2S P3CZE	—	—	—	—	—	—	—	—
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】* 只能在上电复位后写1次。

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	MTU2P1CZE	1	R/W*	MTU2 端口 1 的输出比较 / 高阻抗允许位 设定是否对 MTU2 的大电流引脚 PE9/TIOC3B 和 PE11/TIOC3D 进行输出电平的比较, 并且设定在 OCE1 位是 1 时 OSF1 位被置位的情况下或者在 POE0F 位、POE1F 位、POE2F 位和 MTU2CH34HIZ 位中有 1 位被置位的情况下, 是否将这些引脚置为高阻抗状态。 0: 不进行输出电平的比较并且不置为高阻抗状态。 1: 进行输出电平的比较并且置为高阻抗状态。
13	MTU2P2CZE	1	R/W*	MTU2 端口 2 的输出比较 / 高阻抗允许位 设定是否对 MTU2 的大电流引脚 PE12/TIOC4A 和 PE14/TIOC4C 进行输出电平的比较, 并且设定在 OCE1 位是 1 时 OSF1 位被置位的情况下或者在 POE0F 位、POE1F 位、POE2F 位和 MTU2CH34HIZ 位中有 1 位被置位的情况下, 是否将这些引脚置为高阻抗状态。 0: 不进行输出电平的比较并且不置为高阻抗状态。 1: 进行输出电平的比较并且置为高阻抗状态。
12	MTU2P3CZE	1	R/W*	MTU2 端口 3 的输出比较 / 高阻抗允许位 设定是否对 MTU2 的大电流引脚 PE13/TIOC4B 和 PE15/TIOC4D 进行输出电平的比较, 并且设定在 OCE1 位是 1 时 OSF1 位被置位的情况下或者在 POE0F 位、POE1F 位、POE2F 位和 MTU2CH34HIZ 位中有 1 位被置位的情况下, 是否将这些引脚置为高阻抗状态。 0: 不进行输出电平的比较并且不置为高阻抗状态 1: 进行输出电平的比较并且置为高阻抗状态。
11	—	0	R	保留位 读写值总是 0。
10	MTU2SP1CZE	1	R/W*	MTU2S 端口 1 的输出比较 / 高阻抗允许位 设定是否对 MTU2S 的大电流引脚 PE16/TIOC3BS 和 PE17/TIOC3DS 进行输出电平的比较, 并且设定在 OCE2 位是 1 时 OSF2 位被置位的情况下或者在 POE4F 位、POE5F 位、POE6F 位和 MTU2SHIZ 位中有 1 位被置位的情况下, 是否将这些引脚置为高阻抗状态。 0: 不进行输出电平的比较并且不置为高阻抗状态 1: 进行输出电平的比较并且置为高阻抗状态。
9	MTU2SP2CZE	1	R/W*	MTU2S 端口 2 的输出比较 / 高阻抗允许位 设定是否对 MTU2S 的大电流引脚 PE18/TIOC4AS 和 PE20/TIOC4CS 进行输出电平的比较, 并且设定在 OCE2 位是 1 时 OSF2 位被置位的情况下或者在 POE4F 位、POE5F 位、POE6F 位和 MTU2SHIZ 位中有 1 位被置位时, 是否将这些引脚置为高阻抗状态。 0: 不进行输出电平的比较并且不置为高阻抗状态 1: 进行输出电平的比较并且置为高阻抗状态。

位	位名	初始值	R/W	说明
8	MTU2SP3CZE	1	R/W*	MTU2S 端口 3 的输出比较 / 高阻抗允许位 设定是否对 MTU2S 的大电流引脚 PE19/TIOC4BS 和 PE21/TIOC4DS 进行输出电平的比较, 并且设定在 OCE2 位是 1 时 OSF2 位被置位的情况下或者在 POE4F 位、POE5F 位、POE6F 位和 MTU2SHIZ 位中有 1 位被置位的情况下, 是否将这些引脚置为高阻抗状态。 0: 不进行输出电平的比较并且不置为高阻抗状态 1: 进行输出电平的比较并且置为高阻抗状态。
7 ~ 0	—	全 0	R	保留位 读写值总是 0。

【注】 * 只能在上电复位后写 1 次。

12.4 运行说明

高阻抗控制对象的引脚和条件如表 12.4 所示。

表 12.4 高阻抗控制对象的引脚和条件

引脚	条件	详细内容
MTU2 的大电流引脚 (PE9/TIOC3B 和 PE11/TIOC3D)	进行输入电平的检测、 输出电平的比较或者 SPOER 寄存器的设定	MTU2P1CZE • ((POE2F+POE1F+POE0F)+ (OSF1 • OCE1)+(MTU2CH34HIZ))
MTU2 的大电流引脚 (PE12/TIOC4A 和 PE14/TIOC4C)	进行输入电平的检测、 输出电平的比较或者 SPOER 寄存器的设定	MTU2P2CZE • ((POE2F+POE1F+POE0F)+ (OSF1 • OCE1)+(MTU2CH34HIZ))
MTU2 的大电流引脚 (PE13/TIOC4B 和 PE15/TIOC4D)	进行输入电平的检测、 输出电平的比较或者 SPOER 寄存器的设定	MTU2P3CZE • ((POE2F+POE1F+POE0F)+(OSF1 • OCE1)+ (MTU2CH34HIZ))
MTU2S 的大电流引脚 (PE16/TIOC3BS 和 PE17/TIOC3DS)	进行输入电平的检测、 输出电平的比较或者 SPOER 寄存器的设定	MTU2SP1CZE • ((POE4F+POE5F+POE6F)+(OSF2 • OCE2)+ (MTU2SHIZ))
MTU2S 的大电流引脚 (PE18/TIOC4AS 和 PE20/TIOC4CS)	进行输入电平的检测、 输出电平的比较或者 SPOER 寄存器的设定	MTU2SP2CZE • ((POE4F+POE5F+POE6F)+(OSF2 • OCE2)+ (MTU2SHIZ))
MTU2S 的大电流引脚 (PE19/TIOC4BS 和 PE21/TIOC4DS)	进行输入电平的检测、 输出电平的比较或者 SPOER 寄存器的设定	MTU2SP3CZE • ((POE4F+POE5F+POE6F)+(OSF2 • OCE2)+ (MTU2SHIZ))
MTU2 的 CH0 引脚 (PE0/TIOC0A)	进行输入电平的检测或者 SPOER 寄存器的设定	MTU2PE0ZE • ((POE8F • POE8E)+(MTU2CH0HIZ))
MTU2 的 CH0 引脚 (PE1/TIOC0B)	进行输入电平的检测或者 SPOER 寄存器的设定	MTU2PE1ZE • ((POE8F • POE8E)+(MTU2CH0HIZ))
MTU2 的 CH0 引脚 (PE2/TIOC0C)	进行输入电平的检测或者 SPOER 寄存器的设定	MTU2PE2ZE • ((POE8F • POE8E)+(MTU2CH0HIZ))
MTU2 的 CH0 引脚 (PE3/TIOC0D)	进行输入电平的检测或者 SPOER 寄存器的设定	MTU2PE3ZE • ((POE8F • POE8E)+(MTU2CH0HIZ))

12.4.1 输入电平的检测

当 $\overline{POE0} \sim \overline{POE2}$ 、 $\overline{POE4} \sim \overline{POE6}$ 或者 $\overline{POE8}$ 引脚发生 ICSR1 ~ ICSR3 设定的输入条件时，将大电流引脚和 MTU2 的 CH0 引脚置为高阻抗状态。但是，只在选择了通用输入 / 输出功能或者 MTU2、MTU2S 功能时，才能将大电流引脚和 MTU2 的 CH0 引脚置为高阻抗状态。

(1) 下降沿检测

当 $\overline{POE0} \sim \overline{POE2}$ 、 $\overline{POE4} \sim \overline{POE6}$ 、 $\overline{POE8}$ 引脚从高电平变为低电平时，将大电流引脚和 MTU2 的 CH0 引脚置为高阻抗状态。

从 $\overline{POE0} \sim \overline{POE2}$ 、 $\overline{POE4} \sim \overline{POE6}$ 、 $\overline{POE8}$ 的引脚输入到将引脚置为高阻抗状态的时序例子如图 12.2 所示。

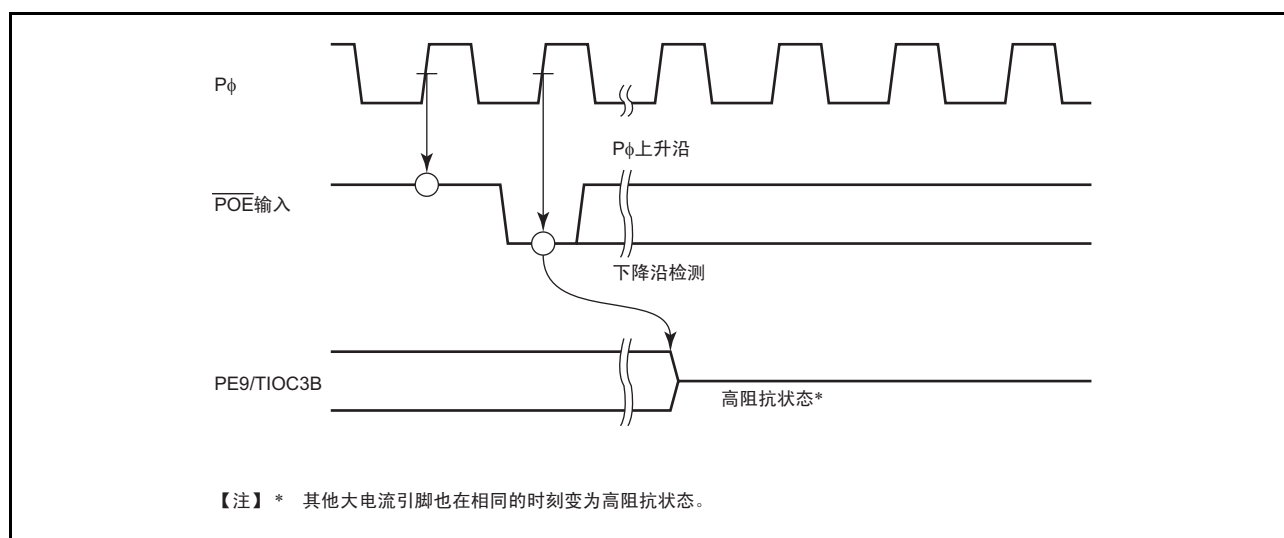


图 12.2 下降沿检测

(2) 低电平检测

低电平检测如图 12.3 所示。通过 ICSR1 ~ ICSR3 设定的采样时钟连续进行 16 次低电平采样。此时，只要有 1 次检测到高电平，就不接受请求。

另外，从采样时钟到大电流引脚变为高阻抗状态的时序与下降沿检测和低电平检测的情况相同。

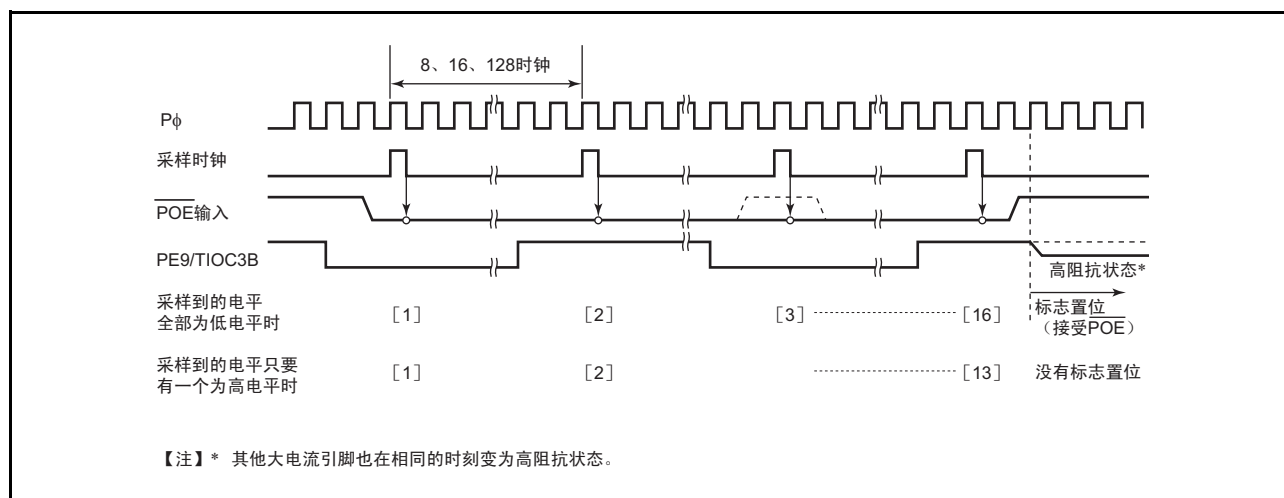


图 12.3 低电平检测

12.4.2 输出电平的比较

以 TIOC3B 和 TIOC3D 的组合为例，输出电平比较的运行如图 12.4 所示。其他引脚的组合也一样。

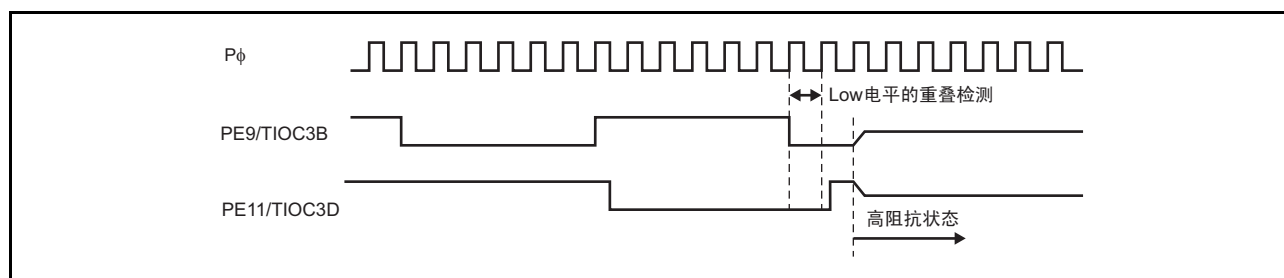


图 12.4 输出电平的比较

12.4.3 高阻抗状态的解除

因输入电平检测而变为高阻抗状态的大电流引脚通过上电复位恢复到初始状态，或者通过清除 ICSR1 ~ ICSR3 的 POE0F ~ POE2F、POE4F ~ POE6F、POE8F 标志 (bit12 ~ 15) 来解除大电流引脚的高阻抗状态。但是，当通过 ICSR1 ~ ICSR3 的 bit0 ~ 7 设定为低电平采样时，如果不是在 POE 引脚输入高电平并进行高电平采样后，即使给标志写 0 也无效，并且不能清除标志。

因输出电平检测而变为高阻抗状态的大电流引脚通过上电复位恢复到初始状态，或者通过清除 OCSR1、OCSR2 的 OCF1、2 标志 (bit15) 来解除大电流引脚的高阻抗状态。但是，如果不是在从大电流引脚输出无效电平后，即使给标志位写 0 也无效，并且不能清除标志。能通过 MTU2、MTU2S 内的寄存器设定进行无效电平的输出。

12.5 中断

如果在输入电平检测或者输出电平比较时满足条件，POE 就能在发出中断请求后产生中断。中断的种类以及产生中断请求的条件如表 12.5 所示。

表 12.5 中断请求的种类和条件

名称	中断源	中断标志	条件
OIE1	输出的允许中断 1	POE2F、POE1F、POE0F、OSF1	$PIE1 \cdot (POE2F + POE1F + POE0F) + OIE1 \cdot OSF1$
OIE2	输出的允许中断 2	POE8F	$PIE3 \cdot POE8F$
OIE3	输出的允许中断 3	POE4F、POE5F、POE6F、OSF2	$PIE2 \cdot (POE4F + POE5F + POE6F) + OIE2 \cdot OSF2$

12.6 使用时的注意事项

12.6.1 从看门狗定时器产生上电复位时的引脚状态

如果从看门狗定时器 (WDT) 产生上电复位, 引脚功能控制器 (PFC) 就被初始化, I/O 端口变为通用输入 (初始值)。但是, 在因端口输出的允许 (POE) 而进行的引脚高阻抗处理中, 如果从 WDT 产生上电复位, 就在转换为通用输入前的 1 个外围时钟 ($P\phi$) 周期的期间, 引脚为输出状态。

在因 MTU2 和 MTU2S 的短路检测而进行的高阻抗处理中, 如果从 WDT 产生上电复位, 也为同样的状态。

在因选择定时器输出时的 POE 输入而进行的高阻抗处理中, 从 WDT 产生上电复位时的引脚状态如图 12.5 所示。

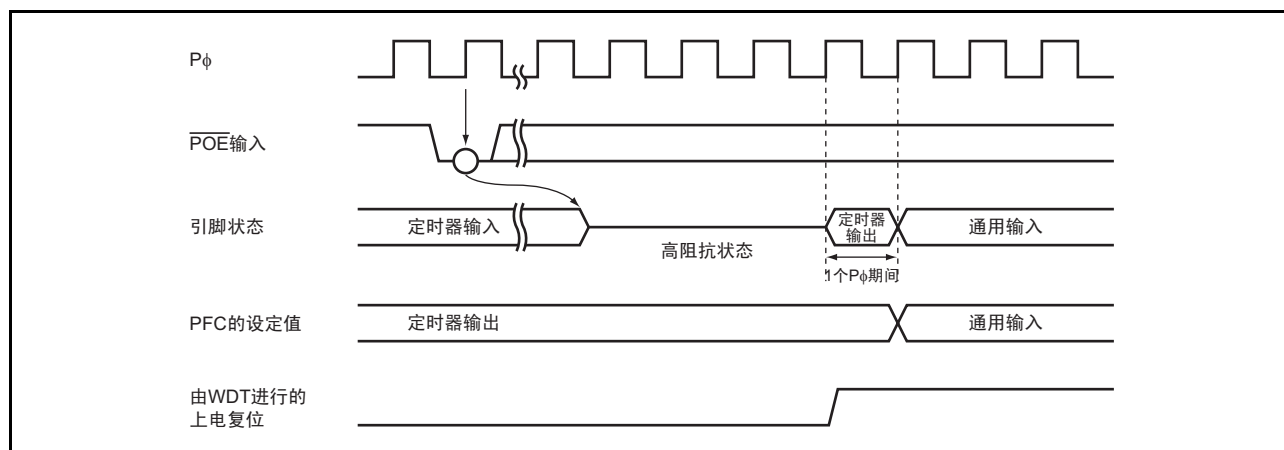


图 12.5 从看门狗定时器产生上电复位时的引脚状态

第 13 章 看门狗定时器 (WDT)

本 LSI 内置看门狗定时器 (WDT)，当因系统失控等原因引起计数器的值不能改写而上溢时，能对本 LSI 内部进行复位。

WDT 是 1 个通道的定时器，以外围时钟为输入时钟，能在解除软件待机模式时用作时钟稳定时间的计数器，并还能用作间隔定时器。

13.1 特点

- 能用于确保时钟的稳定时间
在解除软件待机模式时使用。
- 能在看门狗定时器模式和间隔定时器模式之间进行转换
- 在看门狗定时器模式中产生内部复位
因计数器上溢而产生内部复位。
- 在间隔定时器模式中产生中断
因计数器上溢而产生间隔定时器中断。
- 能选择 8 种计数器的输入时钟
能从 8 种分频时钟 ($\times 1 \sim \times 1/4096$) 中选择外围时钟。
- 能从上电复位和手动复位中选择复位的种类。

WDT 的框图如图 13.1 所示。

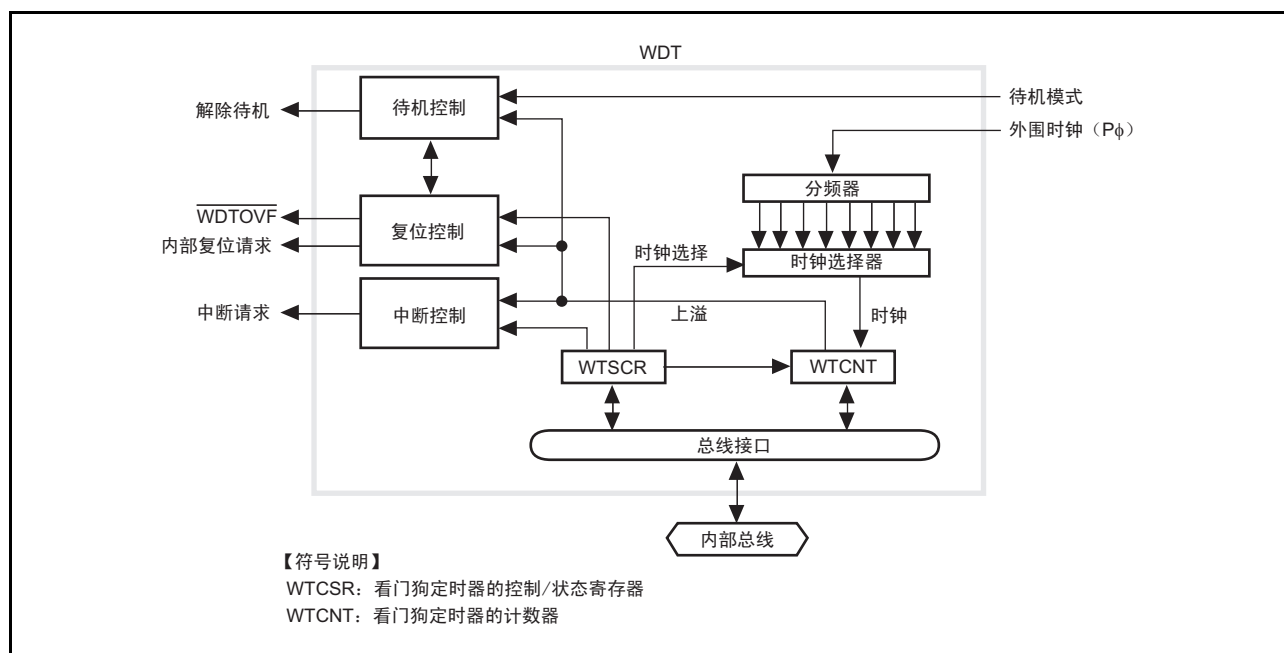


图 13.1 WDT 的框图

13.2 输入 / 输出引脚

WDT 的引脚功能如表 13.1 所示。

表 13.1 WDT 的引脚结构

名称	引脚名	输入 / 输出	功能
看门狗定时器的上溢	$\overline{\text{WDTOVF}}$	输出	因在看门狗定时器模式中发生上溢而产生内部复位，在 WTCSR 的 CKS2 ~ CKS0 位设定的 1 个时钟周期内，此引脚输出低电平。

13.3 寄存器说明

WDT 有以下寄存器。有关这些寄存器的地址以及各处理模式的寄存器状态，请参照“第 25 章 寄存器一览”。

表 13.2 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
看门狗定时器的计数器	WTCNT	R/W	H'00	H'FFFFE810	8、16
看门狗定时器的控制 / 状态寄存器	WTCSR	R/W	H'00	H'FFFFE812	8、16

13.3.1 看门狗定时器的计数器 (WTCNT)

WTCNT 是 8 位可读写寄存器，也是通过所选时钟进行累加计数的计数器。如果发生上溢，就会在看门狗定时器模式中产生复位，而在间隔定时器模式中产生中断。在因 WDT 的上溢而产生内部复位时，不对 WTCNT 计数器进行初始化，而只在通过 $\overline{\text{RES}}$ 引脚进行上电复位时，才将 WTCNT 计数器初始化为 H'00。

必须在将高位字节置为 H'5A 后，以字为单位写 WTCNT 计数器，而以字节为单位读 WTCNT。

【注】为了使此寄存器不被误改写，写数据的方法和一般寄存器不同，详细内容请参照“13.3.3 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

13.3.2 看门狗定时器的控制 / 状态寄存器 (WTCSR)

WTCSR 是 8 位可读写寄存器，由选择计数时钟的位、上溢标志和允许位构成。

在因 WDT 的上溢而产生内部复位时，保持 WTCSR 寄存器的值，而只在通过 $\overline{\text{RES}}$ 引脚进行上电复位时，才将 WTCSR 寄存器初始化为 H'00。如果将 WTCSR 寄存器用于软件待机解除时的时钟稳定时间的计数，就在计数器发生上溢后仍保持该值。

必须在将高位字节置为 H'A5 后，以字为单位写 WTCSR 寄存器，而以字节为单位读 WTCSR 寄存器。

【注】为了使此寄存器不被误改写，写数据的方法和一般寄存器不同。详细内容请参照“13.3.3 寄存器存取时的注意事项”。

位:	7	6	5	4	3	2	1	0
	TME	WT/IT	RSTS	WOVF	IOVF	CKS[2:0]		
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7	TME	0	R/W	定时器的允许 开始或者停止定时器的运行。如果在解除软件待机模式时使用 WDT，就必须将此位置 0。 0: 禁止定时器的运行。停止累加计数，保持 WTCNT 的值。 1: 允许定时器的运行
6	WT/IT	0	R/W	定时器模式的选择 指定是用作看门狗定时器还是用作间隔定时器。 0: 间隔定时器模式 1: 看门狗定时器模式 【注】 如果在 WDT 运行时改写 WT/IT，就有可能无法正确地进行累加计数。
5	RSTS	0	R/W	复位选择 指定在看门狗定时器模式中 WTCNT 发生上溢时产生的复位种类，而在间隔定时器模式中忽视此设定值。 0: 上电复位 1: 手动复位
4	WOVF	0	R/W	看门狗定时器的上溢 表示在看门狗定时器模式中 WTCNT 已发生上溢，而在间隔定时器模式中此位不被置位。 0: 没有发生上溢 1: 在看门狗定时器模式中 WTCNT 已发生上溢
3	IOVF	0	R/W	间隔定时器的上溢 表示在间隔定时器模式中 WTCNT 已发生上溢，而在看门狗定时器模式中此位不被置位。 0: 没有发生上溢 1: 在间隔定时器模式中 WTCNT 已发生上溢
2 ~ 0	CKS[2:0]	000	R/W	时钟选择 2 ~ 0 从分频外围时钟 ($P\phi$) 得到的 8 种时钟中选择用于 WTCNT 计数的时钟。括弧内为外围时钟 $P\phi=40\text{MHz}$ 时的上溢周期值。 000: $P\phi$ (6.4 μs) 001: $P\phi/4$ (25.6 μs) 010: $P\phi/16$ (102.4 μs) 011: $P\phi/32$ (204.8 μs) 100: $P\phi/64$ (409.6 μs) 101: $P\phi/256$ (1.64ms) 110: $P\phi/1024$ (6.55ms) 111: $P\phi/4096$ (26.21ms) 【注】 如果在 WDT 运行时改写 CKS2 ~ CKS0 位，就有可能无法正确地进行累加计数，所以在改写 CKS2 ~ CKS0 位时，必须使 WDT 停止运行。

13.3.3 寄存器存取时的注意事项

为了使看门狗定时器的计数器 (WTCNT) 和看门狗定时器的控制 / 状态寄存器不易被改写, 写数据的方法和一般寄存器不同。必须按照以下方法进行写操作:

- 在写 WTCNT 和 WTCSR 时, 必须使用字传送指令, 而字节传送和长字传送指令不能用于写操作。

如图 13.2 所示, 在写 WTCNT 时, 必须在将高位字节置为 H'5A、低位字节置为写数据后进行传送; 在写 WTCSR 时, 必须在将高位字节置为 H'A5、低位字节置为写数据后进行传送。如此进行传送, 就能将低位字节的数据写到 WTCNT 或者 WTCSR。

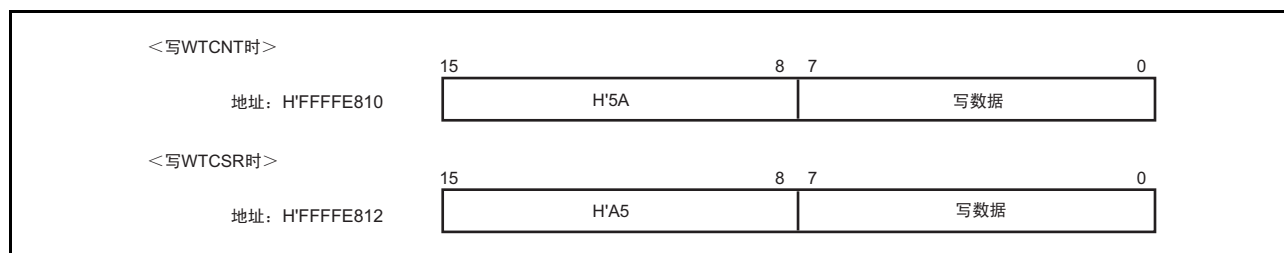


图 13.2 WTCNT 和 WTCSR 的写操作

13.4 运行说明

13.4.1 软件待机模式的解除步骤

在通过 NMI 中断或者外部中断 (IRQ) 解除软件待机模式时使用 WDT, 步骤如下所示 (因为在通过复位进行解除时 WDT 不运行, 所以在时钟稳定前, $\overline{\text{RES}}$ 引脚必须保持低电平)。

- 在进入软件待机模式前, 必须将 WTCSR 寄存器的 TME 位置 0。如果将 TME 位置 1, 就有可能在计数上溢时产生不正常的复位或者间隔定时器中断。
- 预先将要使用的计数时钟的种类和计数器的初始值分别设定到 WTCSR 寄存器 CKS2 ~ CKS0 位和 WTCNT 计数器。在设定这些值时, 计数上溢前的时间必须长于时钟振荡稳定时间。
- 通过执行 SLEEP 指令进入软件待机模式, 停止时钟的运行。
- 如果 NMI 引脚或者 IRQ 引脚的输入电平发生变化, 就通过边沿检测开始 WDT 的计数。
- 如果 WDT 发生计数上溢, CPG 就开始提供时钟, 本 LSI 重新开始运行。此时, WTCSR 寄存器的 WOVF 标志不被置位。

13.4.2 看门狗定时器模式的用法

在看门狗定时器模式的运行中, 每当计数器发生上溢时就产生由 WTCSR 寄存器的 RSTS 位选择的内部复位, 并且 WDTOVF 引脚输出有效信号。

- 将 WTCSR 寄存器的 WT/IT 位置 1, 并将复位的种类设定到 RSTS 位、计数时钟的种类设定到 CKS2 ~ CKS0 位、计数器的初始值设定到 WTCNT 计数器。
- 如果 WTCSR 寄存器的 TME 位被置 1, 就在看门狗定时器模式中开始计数。
- 在看门狗定时器模式的运行中, 如果要使计数器不发生上溢, 就必须定期改写计数器。
- 如果计数器发生上溢, WDT 就将 WTCSR 寄存器的 WOVF 标志置 1, 在 CKS2 ~ CKS0 位设定的计数时钟的 1 个周期内 WDTOVF 引脚输出有效信号, 并且产生由 RSTS 位指定类型的复位, 然后计数器继续计数。

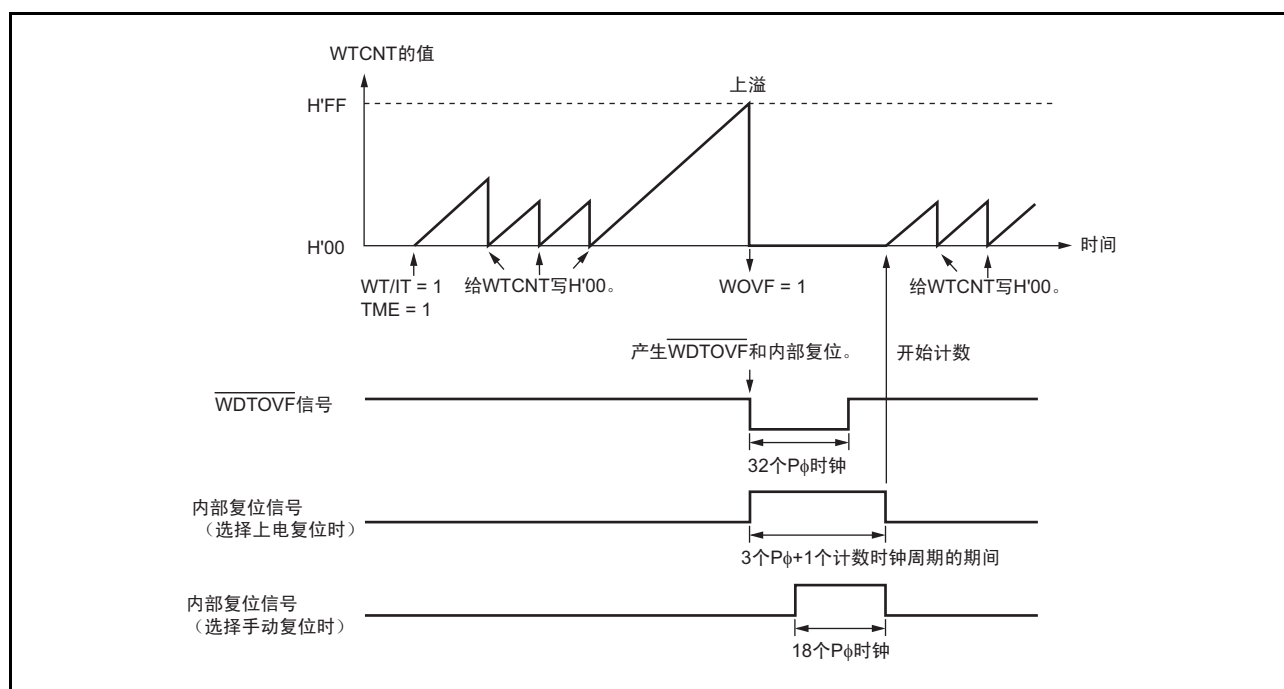


图 13.3 看门狗定时器模式中的运行
(通过 CKS0 ~ CKS2 位将 WTCNT 的计数器时钟设定为 $P\phi/32$)

13.4.3 间隔定时器模式的用法

在间隔定时器模式的运行中，每当计数器发生上溢时，就产生间隔定时器中断，因此能每隔一定的时间产生 1 次中断。

1. 将 WTCNT 寄存器的 WT/IT 位置 0，并将计数时钟的种类设定到 CKS2 ~ CKS0 位、计数器的初始值设定到 WTCNT 计数器。
2. 如果 WTCNT 寄存器的 TME 位被置 1，就在间隔定时器模式中开始计数。
3. 当计数器发生上溢时，WDT 将 WTCNT 寄存器的 IOVF 标志置 1，并将间隔定时器的中断请求传送到 INTC，然后计数器继续计数。

13.5 使用时的注意事项

13.5.1 WTCNT 的设定值

如果在间隔定时器模式中将 WTCNT 置 H'FF，就在 1 个计数时钟后的 H'FF→H'00 时不发生上溢，而在 257 个计数时钟后的 H'FF→H'00 时发生上溢。

如果在看门狗定时器模式中将 WTCNT 置 H'FF，就在 1 个计数时钟后的 H'FF→H'00 时发生上溢。

第 14 章 串行通信接口 (SCI)

本 LSI 具有 3 个独立通道的串行通信接口 (SCI: Serial Communication Interface)。SCI 能以异步通信和时钟同步通信 2 种方式进行串行通信。在异步模式中, 能与 Universal Asynchronous Receiver/Transmitter (UART) 或者 Asynchronous Communication Interface Adapter (ACIA) 等标准异步通信 LSI 进行串行数据的通信。另外, 在异步模式中具有多个处理器之间的串行通信功能 (多处理器通信功能)。

14.1 特点

- 可从异步模式或者时钟同步模式中选择串行通信模式
- 异步模式
通过以字符为单位取得同步的异步方式进行串行数据的通信, 能与 Universal Asynchronous Receiver/Transmitter (UART) 和 Asynchronous Communication Interface Adapter (ACIA) 等标准异步通信 LSI 进行串行数据的通信。
能从 12 种格式中选择串行数据的通信格式。
数据长: 7 位或者 8 位
停止位长: 1 位或者 2 位
奇偶校验: 偶校验、奇校验或者无奇偶校验
能在多个处理器之间进行通信
接收错误的检测: 检测奇偶校验错误、溢出错误和帧错误
中止的检测: 发生帧错误时, 能直接读 RXD 引脚的电平检测中止。
- 时钟同步模式
与时钟同步进行串行数据通信, 能与具有时钟同步通信功能的其他 LSI 进行串行数据的通信。
串行数据的通信格式为 1 种。
数据长: 8 位
接收错误的检测: 检测溢出错误
- 能进行全双工通信
因为具有独立的发送部和接收部, 所以能同时发送和接收。另外, 因为发送部和接收部都为双缓冲结构, 所以能连续发送和接收串行数据。
- 能通过内部波特率发生器选择任意的位速率
- 发送和接收的时钟源能选择波特率发生器的内部时钟或者 SCK 引脚的外部时钟
- 能选择 LSB 先或者 MSB 先 (异步 7 位数据除外)
- 4 种中断源
中断源有发送数据空、发送结束、接收数据满和接收错误共 4 种, 能分别独立地请求中断。另外, 能通过发送数据空请求和接收数据满请求, 启动数据传送控制器 (DTC) 进行数据传送。
- 能设定模块待机模式

SCI 的框图如图 14.1 所示。

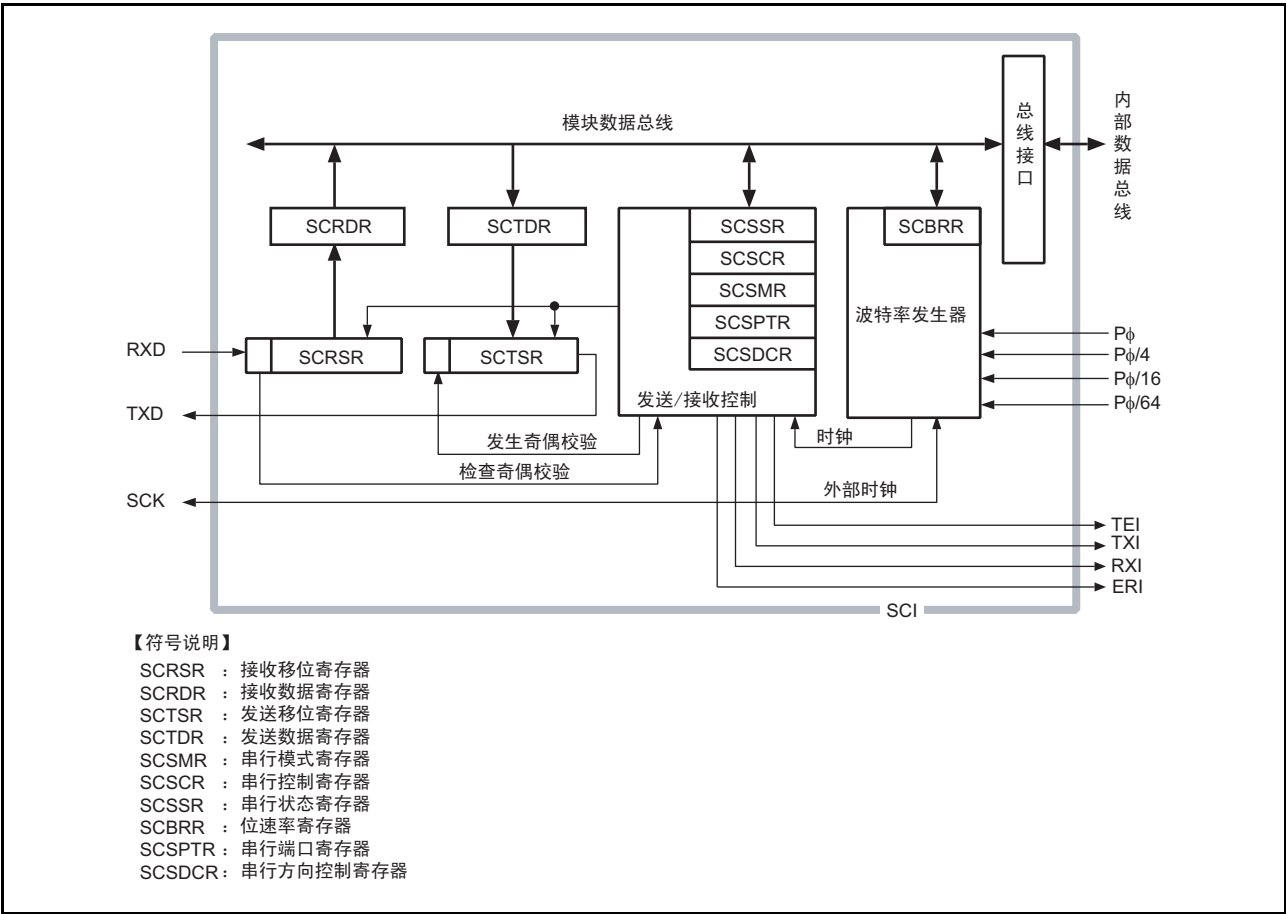


图 14.1 SCI 的框图

14.2 输入 / 输出引脚

SCI 有表 14.1 所示的输入 / 输出引脚。

表 14.1 引脚结构

通道	引脚名 *	输入 / 输出	功能
0	SCK0	输入 / 输出	通道 0 的时钟输入 / 输出引脚
	RXD0	输入	通道 0 的接收数据的输入引脚
	TXD0	输出	通道 0 的发送数据的输出引脚
1	SCK1	输入 / 输出	通道 1 的时钟输入 / 输出引脚
	RXD1	输入	通道 1 的接收数据的输入引脚
	TXD1	输出	通道 1 的发送数据的输出引脚
2	SCK2	输入 / 输出	通道 2 的时钟输入 / 输出引脚
	RXD2	输入	通道 2 的接收数据的输入引脚
	TXD2	输出	通道 2 的发送数据的输出引脚

【注】 * 在本文中省略通道，分别略称为 SCK、RXD 和 TXD。

14.3 寄存器说明

SCI 有以下各通道的寄存器。有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 25 章 寄存器一览”。

表 14.2 寄存器结构

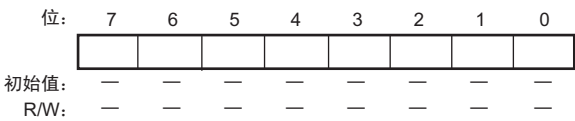
通道	寄存器名	略称	R/W	初始值	地址	存取长度
0	串行模式寄存器_0	SCSMR_0	R/W	H'00	H'FFFFC000	8
	位速率寄存器_0	SCBRR_0	R/W	H'FF	H'FFFFC002	8
	串行控制寄存器_0	SCSCR_0	R/W	H'00	H'FFFFC004	8
	发送数据寄存器_0	SCTDR_0	—	—	H'FFFFC006	8
	串行状态寄存器_0	SCSSR_0	R/W	H'84	H'FFFFC008	8
	接收数据寄存器_0	SCRDR_0	—	—	H'FFFFC00A	8
	串行方向控制寄存器_0	SCSDCR_0	R/W	H'F2	H'FFFFC00C	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

14.3.1 接收移位寄存器（SCRSR）

SCRSR 是接收串行数据的寄存器。

SCI 按照从 LSB（bit0）开始接收的顺序，将 RXD 引脚输入的串行数据保存到 SCRSR，并转换为并行数据。当 1 个字节的数据接受结束时，数据就自动传送到 SCRDR。

CPU 不能直接读写 SCRSR。



14.3.2 接收数据寄存器 (SCRDR)

SCRDR 是保存接收到的串行数据的寄存器。

当 1 个字节的串行数据接收结束时, SCI 就将接收到的串行数据从接收移位寄存器 (SCRSR) 传送并保存到 SCRDR, 然后结束接收运行, SCRSR 为可接收状态。

因为 SCRSR 和 SCRDR 为双缓冲, 所以能连续接收。

SCRDR 是只读寄存器, 所以 CPU 不能写此寄存器。

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

14.3.3 发送移位寄存器 (SCTSR)

SCTSR 是发送串行数据的寄存器。

一旦将发送数据从发送数据寄存器 (SCTDR) 传送到 SCTSR, SCI 就从 LSB (bit0) 开始按顺序将数据送到 TXD 引脚, 进行串行数据的发送。

当 1 个字节的数据发送结束时, 就自动将下一个发送数据从 SCTDR 传送到 SCTSR, 开始发送。但是, 在串行状态寄存器 (SCSSR) 的 TDRE 标志被置 1 的情况下, 不将数据从 SCTDR 传送到 SCTSR。

CPU 不能直接读写 SCTSR。

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

14.3.4 发送数据寄存器 (SCTDR)

SCTDR 是保存串行发送数据的 8 位寄存器。

如果检测到发送移位寄存器 (SCTSR) 为空, SCI 就将写到 SCTDR 的发送数据传送到 SCTSR, 开始串行发送。如果在 SCTSR 的串行数据发送中将下一个发送数据写到 SCTDR, 就能连续进行串行发送。

CPU 随时能读写 SCTDR。

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

14.3.5 串行模式寄存器 (SCSMR)

SCSMR 是设定 SCI 的串行通信格式以及选择波特率发生器时钟源的 8 位寄存器。

CPU 随时能读写 SCSMR。

位:	7	6	5	4	3	2	1	0
	C/Ā	CHR	PE	O/Ē	STOP	MP	CKS[1:0]	
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7	C/A	0	R/W	通信模式 SCI 的运行模式能选择异步模式或者时钟同步模式。 0: 异步模式 1: 时钟同步模式
6	CHR	0	R/W	字符长 异步模式的数据长能选择 7 位或者 8 位数据。在时钟同步模式中, 与 CHR 的设定无关, 数据长固定为 8 位数据。如果选择 7 位数据, 就不发送 SCTDR (发送数据寄存器) 的 MSB (bit7)。 0: 8 位数据 1: 7 位数据
5	PE	0	R/W	奇偶校验的允许 在异步模式中, 选择是否在发送时附加奇偶校验位以及在接收时检查奇偶校验位。在时钟同步模式中, 与 PE 位的设定无关, 不附加和检查奇偶校验位。 0: 禁止附加和检查奇偶校验位 1: 允许附加和检查奇偶校验位 * 【注】 * 如果将 PE 位置 1, 就在发送时将 $O/\bar{E}$ 位指定的偶校验或者奇校验附加到发送数据后进行发送; 而在接收时检查接收的奇偶校验位是否为 $O/\bar{E}$ 位指定的偶校验或者奇校验。
4	$O/\bar{E}$	0	R/W	奇偶校验模式 选择是通过偶校验还是通过奇校验进行奇偶校验的附加和检查。只在异步模式中将 PE 位置 1 (允许附加和检查奇偶校验位) 时, $O/\bar{E}$ 位的设定才有效。在时钟同步模式或者异步模式中禁止附加和检查奇偶校验时, $O/\bar{E}$ 位的指定无效。 0: 偶校验 1: 奇校验 如果设定为偶校验, 就在发送时附加奇偶校验位, 使奇偶校验位和发送字符中的 1 的个数为偶数, 然后进行发送。在接收时检查奇偶校验位和接收字符中的 1 的个数是否为偶数。 如果设定为奇校验, 就在发送时附加奇偶校验位, 使奇偶校验位和发送字符中的 1 的个数为奇数, 然后进行发送。在接收时检查奇偶校验位和接收字符中的 1 的个数是否为奇数。
3	STOP	0	R/W	停止位长 从 1 位或者 2 位中选择异步模式中的停止位长。STOP 位的设定只在异步模式中有效。在设定为时钟同步模式时, 因为不附加停止位, 所以此位的设定无效。 0: 1 个停止位 *1 1: 2 个停止位 *2 另外, 在接收时, 与 STOP 位的设定无关, 只检查接收到的第 1 个停止位。在第 2 个停止位为 1 时, 作为停止位处理, 而在第 2 个停止位为 0 时, 作为下一个发送字符的起始位处理。 【注】 *1 发送时, 在发送字符的末尾附加 1 位的 1 (停止位) 后进行发送。 *2 发送时, 在发送字符的末尾附加 2 位的 1 (停止位) 后进行发送。

位	位名	初始值	R/W	说明
2	MP	0	R/W	多处理器模式 (只在异步模式中有效) 允许或者禁止多处理器功能。在多处理器模式中, PE 位和 O $\bar{E}$ 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	时钟的选择 1、0 选择内部波特率发生器的时钟源。能通过设定 CKS1 位和 CKS0 位, 从 P ϕ 、P ϕ /4、P ϕ /16、P ϕ /64 中选择时钟源。 有关时钟源、位速率寄存器的设定值和波特率的关系, 请参照“14.3.10 位速率寄存器 (SCBRR)”。 00: P ϕ 时钟源 01: P ϕ /4 时钟源 10: P ϕ /16 时钟源 11: P ϕ /64 时钟源 【注】 P ϕ : 外围时钟源

14.3.6 串行控制寄存器 (SCSCR)

SCSCR 是进行 SCI 的发送 / 接收、输出异步模式中的串行时钟、允许或者禁止中断请求, 以及选择发送和接收时钟源的寄存器。

CPU 随时能读写 SCSCR。

位:	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7	TIE	0	R/W	发送的中断允许 在串行状态寄存器 (SCSSR) 的 TDRE 标志被置 1 时 (串行发送数据从发送数据寄存器 (SCTDR) 传送到发送移位寄存器 (SCTSR) 后), 允许或者禁止发送数据空的中断 (TXI) 请求。 能通过在读 TDRE 标志的 1 后将其清 0、或者将 TIE 位清 0, 解除 TXI 中断请求。 0: 禁止发送数据空的中断 (TXI) 请求 1: 允许发送数据空的中断 (TXI) 请求
6	RIE	0	R/W	接收的中断允许 在 SCSSR 的 RDRF 标志被置 1 时 (串行接收数据从接收移位寄存器 (SCRSR) 传送到接收数据寄存器 (SCRDR) 后), 允许或者禁止接收数据满的中断 (RXI) 请求和接收错误的中断 (ERI) 请求。 能通过读 RDRF 标志或者 FER、PER、ORER 标志的 1 后将其清 0、或者将 RIE 位清 0, 解除 RXI 和 ERI 中断请求。 0: 禁止接收数据满的中断 (RXI) 请求和接收错误的中断 (ERI) 请求 1: 允许接收数据满的中断 (RXI) 请求和接收错误的中断 (ERI) 请求

位	位名	初始值	R/W	说明
5	TE	0	R/W	发送允许 允许或者禁止 SCI 的串行发送。 0: 禁止发送 *1 1: 允许发送 *2 【注】 *1 SCSSR 的 TDRE 标志固定为 1。 *2 在此状态下, 如果在发送数据写到 SCTDR 后将 SCSSR 的 TDRE 标志清 0, 就开始串行发送。另外, 必须在将 TE 位置 1 前设定串行模式寄存器 (SCSMR), 决定发送格式。
4	RE	0	R/W	接收允许 允许或者禁止 SCI 的串行接收。 0: 禁止接收 *1 1: 允许接收 *2 【注】 *1 必须注意: 即使将 RE 位清 0, RDRF、FER、PER、ORER 的各标志也不受影响而保持状态。 *2 在此状态下, 如果在异步模式中检测到起始位或者在时钟同步模式中检测到同步时钟输入, 就开始串行接收。另外, 必须在将 RE 位置 1 前设定 SCSMR, 决定接收格式。
3	MPIE	0	R/W	多处理器的中断允许 (在异步模式中 SCSMR 的 MP=1 时有效) 如果将此位置 1, 就跳读多处理器位为 0 的数据, 禁止设定 SCSSR 的 RDRF、FER、ORER 的各状态标志。如果接收到多处理器位为 1 的数据, 就自动清除此位, 返回到通常的接收运行。详细内容请参照“14.4.4 多处理器通信功能”。
2	TEIE	0	R/W	发送结束的中断允许 在发送 MSB 数据并且 SCTDR 中无有效的发送数据时, 允许或者禁止发送结束的中断 (TEI) 请求。 能通过在读 SCSSR 的 TDRE 标志的 1 后将此标志和 TEND 标志清 0、或者将 TEIE 位清 0, 解除 TEI 中断请求。 0: 禁止发送结束的中断 (TEI) 请求 1: 允许发送结束的中断 (TEI) 请求
1、0	CKE[1:0]	00	R/W	时钟的允许 1、0 选择 SCI 的时钟源, 允许或者禁止从 SCK 引脚输出时钟。通过 CKE1 位和 CKE0 位的组合, 将 SCK 引脚设定为串行时钟的输出引脚或者串行时钟的输入引脚。 在时钟同步模式中, 如果设定为同步时钟输出, 就必须在将 SCSMR 的 C/A 位置 1 后, 设定 CKE1 位和 CKE0 位。有关 SCI 的时钟源选择, 请参照“14.4 运行说明”的表 14.14。 异步模式 00: 内部时钟 /SCK 引脚为输入引脚 (忽视输入信号) 01: 内部时钟 /SCK 引脚为时钟的输出引脚 *1 10: 外部时钟 /SCK 引脚为时钟的输入引脚 *2 11: 外部时钟 /SCK 引脚为时钟的输入引脚 *2 时钟同步模式 00: 内部时钟 /SCK 引脚为同步时钟的输出引脚 01: 内部时钟 /SCK 引脚为同步时钟的输出引脚 10: 外部时钟 /SCK 引脚为同步时钟的输入引脚 11: 外部时钟 /SCK 引脚为同步时钟的输入引脚 【注】 *1 输出时钟的频率是位速率的 16 倍 *2 输入时钟的频率是位速率的 16 倍

14.3.7 串行状态寄存器 (SCSSR)

SCSSR 是表示 SCI 运行状态标志的 8 位寄存器。

CPU 随时能读写 SCSSR。但是，不能给 TDRE、RDRF、ORER、PER、FER 的各标志写 1。另外，为了将这些标志清 0，需要预先读 1。而且，TEND 标志为只读位，不能写。

位:	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
初始值:	1	0	0	0	0	1	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

【注】* 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说明
7	TDRE	1	R/(W)*	发送数据寄存器空 表示数据已从发送数据寄存器 (SCTDR) 传送到发送移位寄存器 (SCTSR)，并且处于能将下一个串行发送数据写到 SCTDR 的状态。 0: 表示已将有效的发送数据写到 SCTDR [清除条件] - 当读到 TDRE=1 的状态后写 0 时 - 在通过 TXI 中断启动 DTC 并且在 DTC 的 MRB 的 DISEL 位为 0 的情况下给 SCTDR 传送发送数据时 1: 表示 SCTDR 中无有效的发送数据 [置位条件] - 当发生上电复位或者处于待机模式时 - 当 SCSCR 的 TE 位为 0 时 - 当数据已从 SCTDR 传送到 SCTSR 并且处于能将数据写到 SCTDR 的状态时
6	RDRF	0	R/(W)*	接收数据寄存器满 表示接收的数据被保存到接收数据寄存器 (SCRDR) 0: 表示有效的接收数据没被保存到 SCRDR [清除条件] - 当发生上电复位或者处于待机模式时 - 当读到 RDRF=1 的状态后写 0 时 - 在通过 RXI 中断启动 DTC 并且在 DTC 的 MRB 的 DISEL 位为 0 的情况下从 SCRDR 传送数据时 1: 表示有效的接收数据已被保存到 SCRDR [置位条件] - 当串行接收正常结束并将接收数据从 SCRSR 传送到 SCRDR 时 【注】 在接收时检测到错误或者将串行控制寄存器 (SCSCR) 的 RE 位清 0 时，SCRDR 和 RDRF 标志不受影响而保持以前的状态。 必须注意: 如果在 RDRF 标志被置 1 的状态下结束下一个数据的接收，就会发生溢出错误，丢失接收到的数据。

位	位名	初始值	R/W	说明
5	ORER	0	R/(W)*	溢出错误 表示接收时发生溢出错误并异常结束。 0: 表示正在接收或者接收正常结束 *1 [清除条件] - 当发生上电复位或者处于待机模式时 - 当读到 ORER=1 的状态后写 0 时 1: 表示在接收时发生溢出错误 *2 [置位条件] - 在 RDRF=1 的状态下结束下一个串行接收时 【注】 *1 在将 SCSCR 的 RE 位清 0 时, ORER 标志不受影响而保持以前的状态。 *2 SCRDR 保持溢出错误发生前的接收数据, 后接收的数据会丢失。另外, 在 ORER=1 的状态下, 不能继续以后的串行接收。
4	FER	0	R/(W)*	帧错误 在异步模式中, 表示在接收时发生帧错误并异常结束。 0: 表示正在接收或者接收正常结束 *1 [清除条件] - 当发生上电复位或者处于待机模式时 - 当读到 FER=1 的状态后写 0 时 1: 表示在接收时发生帧错误 [置位条件] - SCI 在结束接收时检查接收数据末尾的停止位是否为 1, 当停止位为 0 时 *2 【注】 *1 在将 SCSCR 的 RE 位清 0 时, FER 标志不受影响而保持以前的状态。 *2 在 2 个停止位模式中, 只判断第 1 位的停止位是否为 1, 而不检查第 2 位的停止位。另外, 将发生帧错误时的接收数据传送到 SCRDR, 但是 RDRF 标志不置位。在 FER 标志被置 1 的状态下, 不能继续以后的串行接收。
3	PER	0	R/(W)*	奇偶校验错误 在异步模式中, 表示在附加奇偶校验的接收时发生奇偶校验错误并异常结束。 0: 表示正在接收或者接收正常结束 *1 [清除条件] - 当发生上电复位或者待机模式时 - 当读到 PER=1 的状态后写 0 时 1: 表示在接收时发生奇偶校验错误 *2 [置位条件] - 当接收时的接收数据和奇偶校验位的 1 的个数与串行模式寄存器 (SCSMR) 的 O/E 位指定的偶校验或者奇校验的设定不一致时 【注】 *1 在将 SCSCR 的 RE 位清 0 时, PER 标志不受影响而保持以前的状态。 *2 将发生奇偶校验错误时的接收数据传送到 SCRDR, 但是 RDRF 标志不被置位。在 PER 标志被置 1 的状态下, 不能继续以后的串行接收。

位	位名	初始值	R/W	说明
2	TEND	1	R	发送结束 表示在发送字符的最后位时 SCTDR 中无有效数据并发送结束。 TEND 标志为只读位，不能写。 0: 表示正在发送 [清除条件] • 当读到 TDRE=1 的状态后给 TDRE 标志写 0 时 1: 表示发送已结束 [置位条件] • 当发生上电复位或者处于待机模式时 • 当 SCSCR 的 TE 位为 0 时 • 在发送 1 个字节的串行发送字符的最后位时 TDRE=1 【注】 在通过 TXI 中断启动 DTC 并且将数据写到 SCTDR 时，TEND 标志为不定值，所以 TEND 标志不能用作发送结束标志。
1	MPB	0	R	多处理器位 保存接收帧中的多处理器位的值。当 SCSCR 的 RE 为 0 时，此位不变。
0	MPBT	0	R/W	多处理器位的发送 设定给发送帧附加的多处理器位的值。

【注】 * 为了清除标志，只能在读 1 后写 0。

14.3.8 串行端口寄存器 (SCSPTR)

SCSPTR 对串行通信接口 (SCI) 引脚多路复用的端口进行输入 / 输出和数据的控制。能通过写 TXD 引脚的输出数据，控制串行发送 / 接收的中止，并能用 bit3 和 bit2 读 SCK 引脚的数据以及写 SCK 引脚的输出数据，bit7 控制 RXI 中断的允许 / 禁止。SCSPTR 是 8 位寄存器，CPU 能随时 CPU 读写此寄存器。另外，读 SCI 引脚的值时，必须使用端口寄存器。详细内容请参照“第 21 章 I/O 端口”。

位:	7	6	5	4	3	2	1	0
	EIO	—	—	—	SPB1IO	SPB1DT	SPB0IO	SPB0DT
初始值:	0	0	0	0	0	不定	0	不定
R/W:	R/W	—	—	—	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7	EIO	0	R/W	仅错误中断 当 EIO 位为 1 时，即使将 RIE 位置 1，也不向 CPU 请求 RXI 中断。 此位设定是允许还是禁止 RXI 中断。 0: RIE 位允许或者禁止 RXI 和 ERI 中断 当 RIE 位为 1 时，将 RXI 和 ERI 中断送给 INTC 1: 当 RIE 位为 1 时，只将 ERI 中断送给 INTC
6 ~ 4	—	全 0	—	保留位 读写值总是 0。

位	位名	初始值	R/W	说明																				
3	SPB1IO	0	R/W	串行端口的时钟端口输入 / 输出 指定串行端口的 SCK 引脚的输入 / 输出。实际上在 SCK 引脚作为端口输出引脚并输出 SPB1DT 位的设定值时，必须将 SCSMR 的 C/A 位和 SCSCR 的 CKE1 位、CKE0 位置 0。 0: 不将 SPB1DT 位的值输出到 SCK 引脚 1: 将 SPB1DT 位的值输出到 SCK 引脚																				
2	SPB1DT	不定	R/W	串行端口的时钟端口数据 指定串行端口的 SCK 引脚的输出数据。通过 SPB1IO 位指定输出是否有效（详细内容请参照 SPB1IO 位的说明）。输出时，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	串行端口的中止输出 通过 SPB0DT 位和 SCSCR 的 TE 位的组合控制 TXD 引脚。																				
0	SPB0DT	不定	R/W	串行端口的中止数据 通过 SPB0IO 位与 SCSCR 的 TE 位的组合控制 TXD 引脚。但是，必须预先通过引脚功能控制器（PFC）选择 TXD 引脚功能。 <table><tr><th>SCSCR 的 TE 位设定值</th><th>SPB0IO 位设定值</th><th>SPB0DT 位设定值</th><th>TXD 引脚状态</th></tr><tr><td>0</td><td>0</td><td>*</td><td>SPB0DT 输出无效状态（初始状态）</td></tr><tr><td>0</td><td>1</td><td>0</td><td>低电平输出</td></tr><tr><td>0</td><td>1</td><td>1</td><td>高电平输出</td></tr><tr><td>1</td><td>*</td><td>*</td><td>根据串行核心逻辑，输出发送数据</td></tr></table>	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
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																								

14.3.9 串行方向控制寄存器 (SCSDCR)

SCSDCR 通过 DIR 位选择 LSB 先或者 MSB 先。在 8 位长的情况下, 不论哪种串行通信模式都能选择 LSB 先或者 MSB 先。

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

位	位名	初始值	R/W	说明
7 ~ 4	—	全 1	R	保留位 读写值总是 1。
3	DIR	0	R/W	数据的发送方向 选择串行 / 并行转换格式。发送 / 接收格式在 8 位时有效。 0: 以 LSB 先发送 SCTDR 的内容 以 LSB 先将接收数据保存到 SCRDR 1: 以 MSB 先发送 SCTDR 的内容 以 MSB 先将接收数据保存到 SCRDR

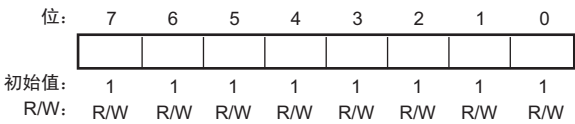
位	位名	初始值	R/W	说明
2	—	0	R	保留位 读写值总是 0。
1	—	1	R	保留位 读写值总是 1。
0	—	0	R	保留位 读写值总是 0。

14.3.10 位速率寄存器 (SCBRR)

SCBRR 是 8 位寄存器, 根据串行模式寄存器 (SCSMR) 的 CKS1 位和 CKS0 位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。

CPU 随时能读写 SCBRR。

用以下计算式求 SCBRR 的设定值:



(异步模式)

$$N = \frac{P\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

(时钟同步模式)

$$N = \frac{P\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: 位速率 (bit/s)

N: 波特率发生器的 SCBRR 设定值 ($0 \leq N \leq 255$)

(必须设定满足电特性的值)

Pφ: 外围模块的工作频率 (MHz)

n: 波特率发生器输入时钟 (n=0、1、2、3)

(n 和时钟的关系请参照表 14.3)

表 14.3 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

用以下计算式求异步模式的位速率误差:

$$\text{误差}(\%) = \left\{ \frac{P\phi \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

异步模式的 SCBRR 的设定例子如表 14.4 ~ 表 14.6 所示, 时钟同步模式的 SCBRR 的设定例子如表 14.7 ~ 表 14.9 所示。

表 14.4 位速率的 SCBRR 的设定例子 (异步模式) (1)

位速率 (bit/s)	P ϕ (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 14.5 位速率的 SCBRR 的设定例子 (异步模式) (2)

位速率 (bit/s)	P ϕ (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 14.6 位速率的 SCBRR 的设定例子 (异步模式) (3)

位速率 (bit/s)	P ϕ (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 14.7 位速率的 SCBRR 的设定例子 (时钟同步模式) (1)

位速率 (bit/s)	P ϕ (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 14.8 位速率的 SCBRR 的设定例子 (时钟同步模式) (2)

位速率 (bit/s)	P ϕ (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 14.9 位速率的 SCBRR 的设定例子 (时钟同步模式) (3)

位速率 (bit/s)	P ϕ (MHz)							
	34		36		38		40	
	n	N	n	N	n	N	n	N
250								
500								
1000	3	132	3	140	3	147	3	155
2500	2	212	2	224	2	237	2	249
5000	2	105	2	112	2	118	2	124
10000	1	212	1	224	1	237	1	249
25000	1	84	1	89	1	94	1	99
50000	0	169	0	179	0	189	0	199
100000	0	84	0	89	0	94	0	99
250000	0	33	0	35	0	37	0	39
500000	0	16	0	17	0	18	0	19
1000000	—	—	0	8	—	—	0	9
2500000	—	—	—	—	—	—	0	3
5000000	—	—	—	—	—	—	0	1

【注】 必须尽量将误差设定在 1% 以内。

【符号说明】

空白栏：不能设定。

—：能设定，但是会出现误差。

*：不能连续发送和接收。

使用波特率发生器时异步模式的各频率的最大位速率如表 14.10 所示，输入外部时钟时的最大位速率如表 14.11 和表 14.12 所示。

表 14.10 使用波特率发生器时的各频率的最大位速率（异步模式）

P ϕ (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 14.11 输入外部时钟时的最大位速率（异步模式）

P ϕ (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 14.12 输入外部时钟时的最大位速率 (时钟同步模式)

P _φ (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

14.4 运行说明

14.4.1 概要

SCI 能通过异步模式（以字符为单位边同步边通信）和时钟同步模式（通过时钟脉冲边同步边通信）2 种方式进行串行通信。

如表 14.13 所示，通过串行模式寄存器（SCSMR）选择异步模式、时钟同步模式和发送格式。另外，如表 14.14 所示，SCI 的时钟源取决于 SCSMR 的 $C/\bar{A}$ 位和串行控制寄存器（SCSCR）的 CKE1 位、CKE0 位的组合。

(1) 异步模式

- 数据长：能从 7 位/8 位中选择
- 能选择附加奇偶校验、或者附加 1 位或者 2 位的停止位（根据这些组合，决定发送/接收格式和字符长）
- 在接收时，能检测帧错误、奇偶校验错误、溢出错误和中止
- SCI 的时钟源：能从内部时钟或者外部时钟中选择
 选择内部时钟：能通过波特率发生器的时钟运行 SCI，输出频率为 16 倍位速率的时钟
 选择外部时钟：需要输入频率为 16 倍位速率的时钟（不使用内部波特率发生器）

(2) 时钟同步模式

- 发送/接收格式：固定为 8 位数据
- 在接收时，能检测溢出错误
- SCI 的时钟源：能从内部时钟或者外部时钟中选择
 选择内部时钟：通过波特率发生器的时钟运行 SCI，将同步时钟输出到外部
 选择外部时钟：不使用内部波特率发生器而通过输入的同步时钟运行

表 14.13 SCSMR 的设定值和串行发送 / 接收格式

SCSMR 的设定值				模式	SCI 的发送 / 接收格式		
bit7	bit6	bit5	bit3		数据长	奇偶校验位	停止位长
$C/\bar{A}$	CHR	PE	STOP				
0	0	0	0	异步模式	8 位数据	无	1 位
			1				2 位
		1	0			有	1 位
			1				2 位
	1	0	0		7 位数据	无	1 位
			1				2 位
		1	0			有	1 位
			1				2 位
1	x	x	x	时钟同步模式	8 位数据	无	无

【符号说明】 x: Don't care

表 14.14 SCSMR、SCSCR 的设定和 SCI 时钟源的选择

SCSMR	SCSCR 的设定值		模式	时钟源	SCK 引脚的功能
bit7	bit1	bit0			
C/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			

14.4.2 异步模式的运行

异步模式是以字符为单位边同步边进行串行通信的模式，将表示通信开始的起始位和通信结束的停止位附加到数据后进行字符的发送和接收。

因为在 SCI 内部发送部和接收部是独立的，所以能进行全双工通信。另外，发送部和接收部都为双缓冲结构，所以能在发送和接收时读写数据，连续进行发送和接收。

异步串行通信的一般格式如图 14.2 所示。

在异步串行通信时，通信线路通常保持为标记状态（高电平）。SCI 监视通信线路，将空闲状态（低电平）视为起始位，开始串行通信。

以起始位（低电平）开始，数据（LSB 先：从最低位开始）、奇偶校验位（高 / 低电平），最后为停止位（高电平）的顺序构成串行通信的 1 个字符。

在异步模式中，SCI 在接收时的起始位下降沿取得同步。另外，因为 SCI 在 1 个频率为 16 倍位速率的第 8 个时钟对数据进行采样，所以在各位的中央取得通信数据。

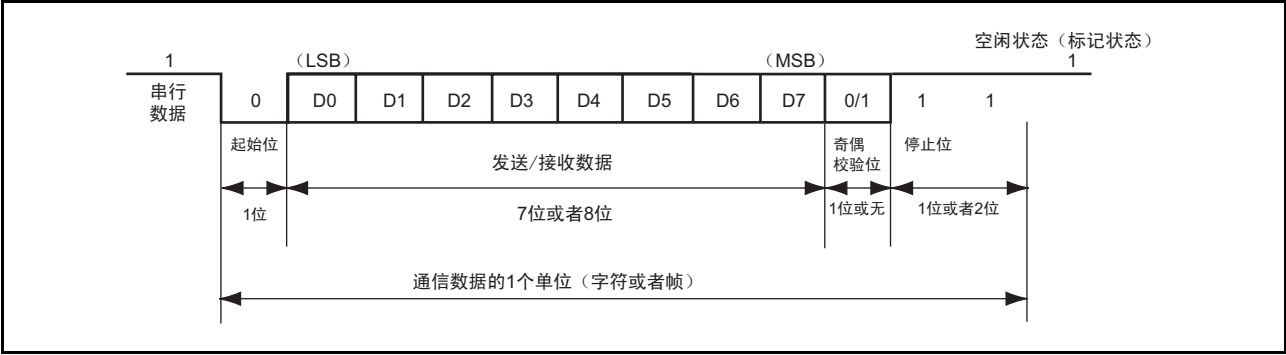


图 14.2 异步通信的数据格式
(8 位数据 / 有奇偶校验 / 2 个停止位的例子)

(1) 发送 / 接收的格式

在异步模式中，能设定的发送 / 接收格式如表 14.15 所示。
发送 / 接收格式有 12 种，能通过串行模式寄存器 (SCSMR) 的设定进行选择。

表 14.15 串行发送 / 接收格式 (异步模式)

SCSMR的设定				串行发送/接收格式和帧长											
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	S	8位数据								STOP		
0	0	0	1	S	8位数据								STOP	STOP	
0	1	0	0	S	8位数据								P	STOP	
0	1	0	1	S	8位数据								P	STOP	STOP
1	0	0	0	S	7位数据							STOP			
1	0	0	1	S	7位数据							STOP	STOP		
1	1	0	0	S	7位数据							P	STOP		
1	1	0	1	S	7位数据							P	STOP	STOP	
0	x	1	0	S	8位数据								MPB	STOP	
0	x	1	1	S	8位数据								MPB	STOP	STOP
1	x	1	0	S	7位数据							MPB	STOP		
1	x	1	1	S	7位数据							MPB	STOP	STOP	

【符号说明】

- S : 起始位
- STOP : 停止位
- P : 奇偶校验位
- MPB : 多处理器位
- x : Don't care

(2) 时钟

能通过设定 SCSMR 的 C/A 位和串行控制寄存器（SCSCR）的 CKE1 位、CKE0 位，从内部波特率发生器生成的内部时钟或者 SCK 引脚输入的外部时钟中选择 SCI 的发送和接收时钟。有关 SCI 时钟源的选择请参照表 14.14。

在将外部时钟输入到 SCK 引脚时，必须输入频率为 16 倍位速率的时钟。
在通过内部时钟运行时，能从 SCK 引脚输出时钟。此时，输出的时钟频率是位速率的 16 倍。

(3) 数据的发送 / 接收

• SCI 的初始化（异步模式）

在发送和接收数据前，必须首先将 SCSCR 的 TE 位和 RE 位清 0，然后按照以下顺序初始化 SCI。在更改运行模式和通信格式等时，也必须在将 TE 位和 RE 位清 0 后按照以下步骤进行。如果将 TE 位清 0，TDRE 标志就被置 1，发送移位寄存器（SCTSR）被初始化。必须注意：即使将 RE 位清 0，也保持 RDRF、PER、FER、ORER 的各标志和接收数据寄存器（SCRDR）的内容。
在使用外部时钟时，不能在含有初始化的运行中停止时钟，否则运行就不稳定。

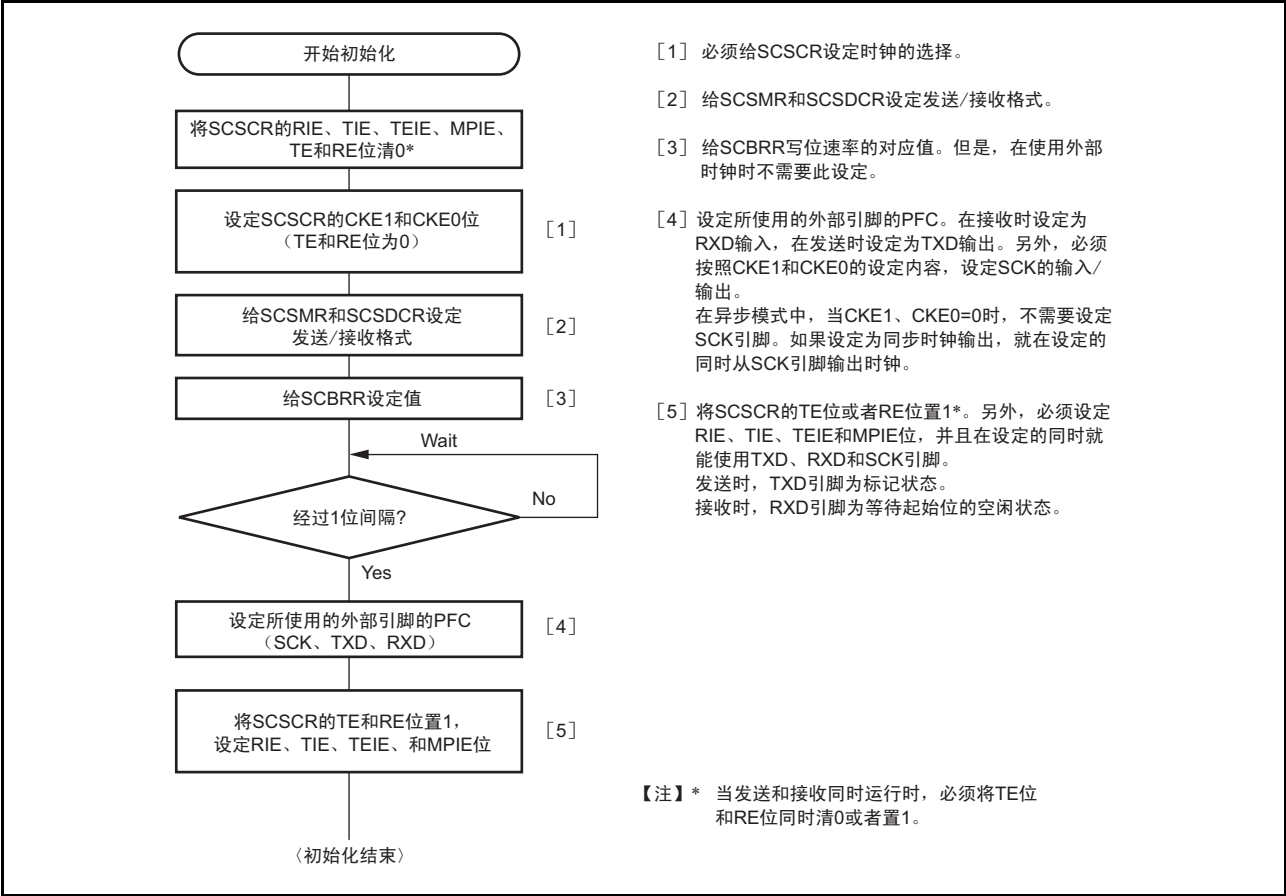


图 14.3 SCI 的初始化流程图例子

• 串行数据的发送（异步模式）

串行发送的流程图例子如图 14.4 所示。
在将 SCI 设定为可发送的运行状态后，按照以下步骤发送串行数据。

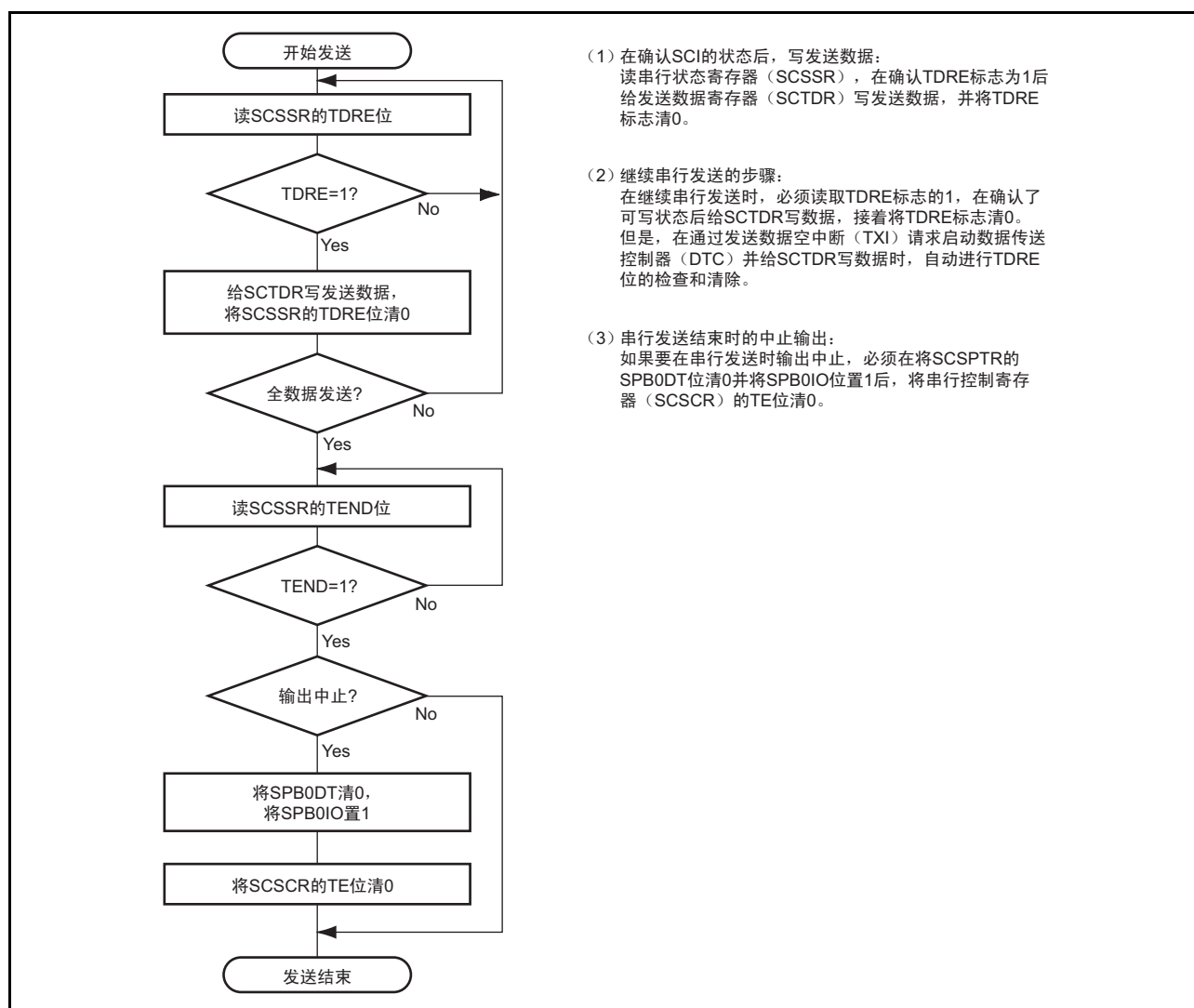


图 14.4 串行发送的流程图例子

串行发送时的 SCI 运行如下：

1. SCI 监视串行状态寄存器（SCSSR）的 TDRE 标志，如果该标志为 0，就认为已将数据写到发送数据寄存器（SCTDR），然后将数据从 SCTDR 传送到发送移位寄存器（SCTSR）。
2. 在数据从 SCTDR 传送到 SCTSR 后，将 TDRE 标志置 1，开始发送。此时，如果串行控制寄存器（SCSCR）的 TIE 位已被置 1，就产生发送数据空的中断（TXI）请求。

按照以下顺序从 TXD 引脚送出串行发送数据：

- a. 起始位：输出 1 位的 0。
 - b. 发送数据：从 LSB 按顺序输出 8 位或者 7 位数据。
 - c. 奇偶校验位或者多处理器位：输出 1 位的奇偶校验位（偶校验或者奇校验）或者 1 位的多处理器位。另外，也能选择不输出奇偶校验位或者多处理器位的格式。
 - d. 停止位：输出 1 位或者 2 位的 1（停止位）。
 - e. 标记状态：在送出开始下一次发送的起始位前连续输出 1。
3. SCI 在送出停止位时检查 TDRE 标志。

如果 TDRE 标志为 0，就将数据从 SCTDR 传送到 SCTSR，在送出停止位后，开始下一帧的串行发送。如果 TDRE 标志为 1，就将串行状态寄存器（SCSSR）的 TEND 位置 1，在送出停止位后变为输出 1 的标记状态。此时，如果 SCSCR 的 TEIE 标志已被置 1，就产生 TEI 中断请求。

异步模式发送时的运行例子如图 14.5 所示。

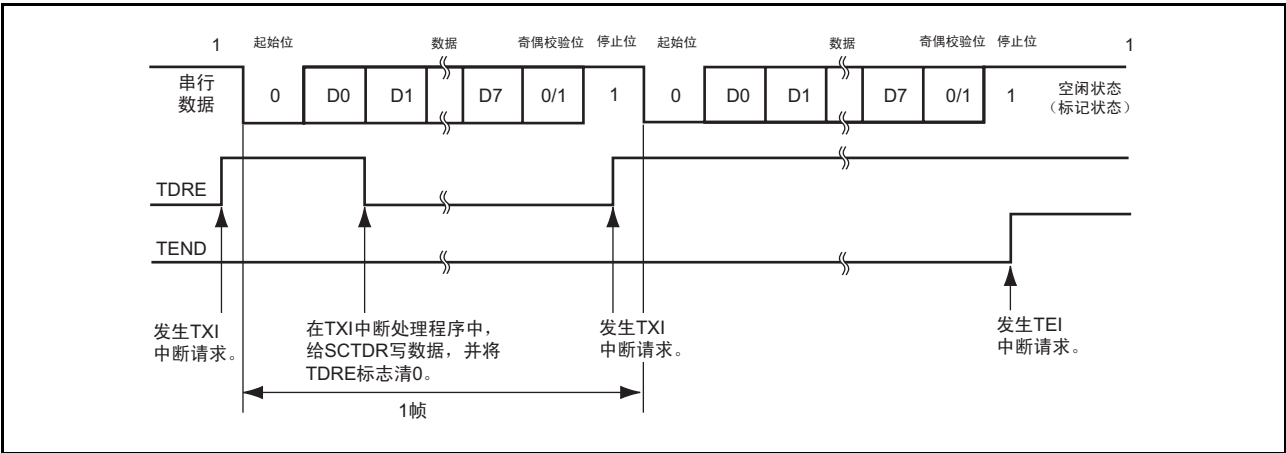


图 14.5 异步模式发送时的运行例子
(8 位数据 / 有奇偶校验 / 1 个停止位的例子)

• 串行数据的接收 (异步模式)

串行接收的流程图例子如图 14.6 所示。

在将 SCI 设定为可接受的运行状态后，按照以下步骤接收串行数据。

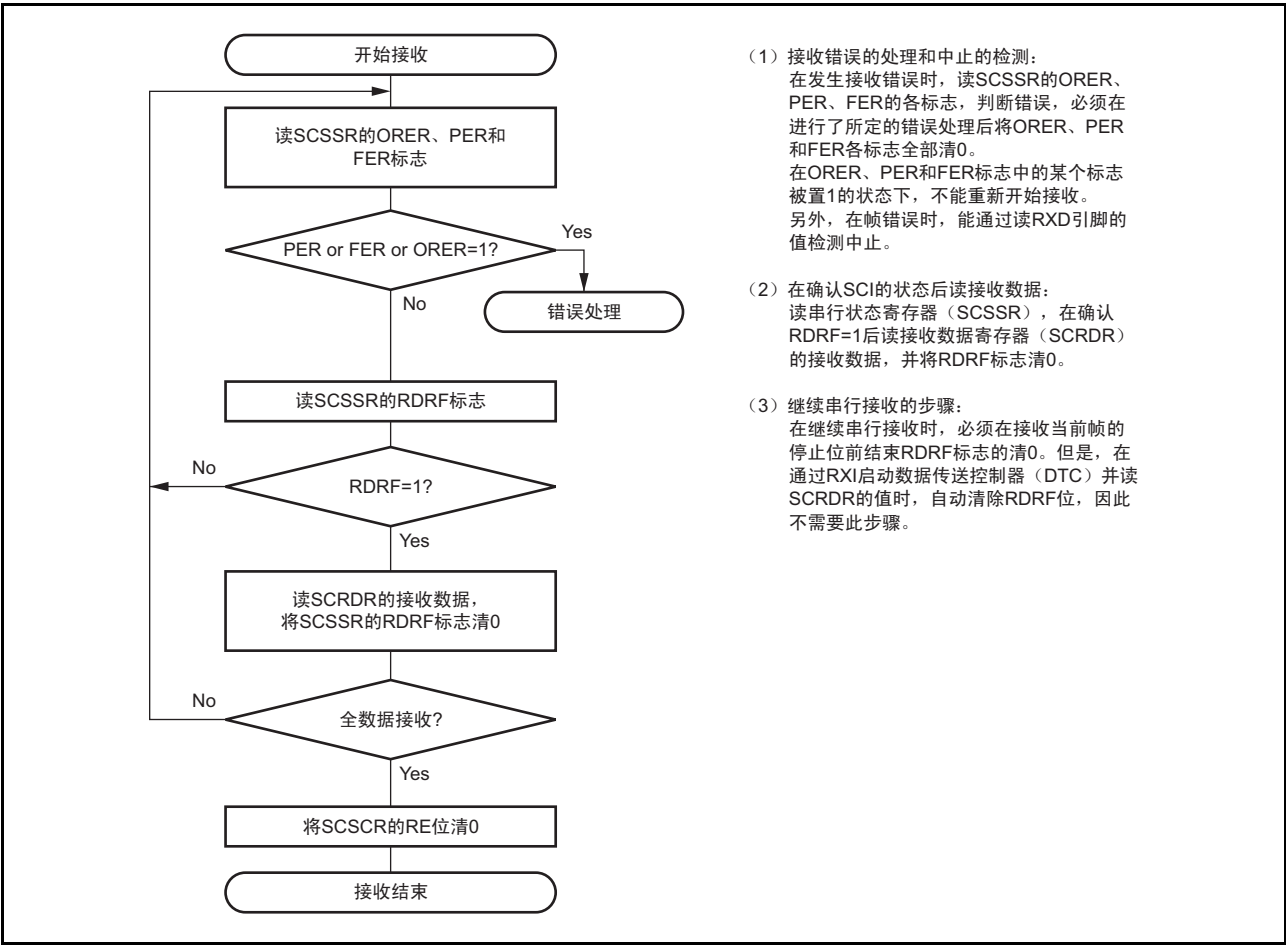


图 14.6 串行接收的流程图例子 (1)

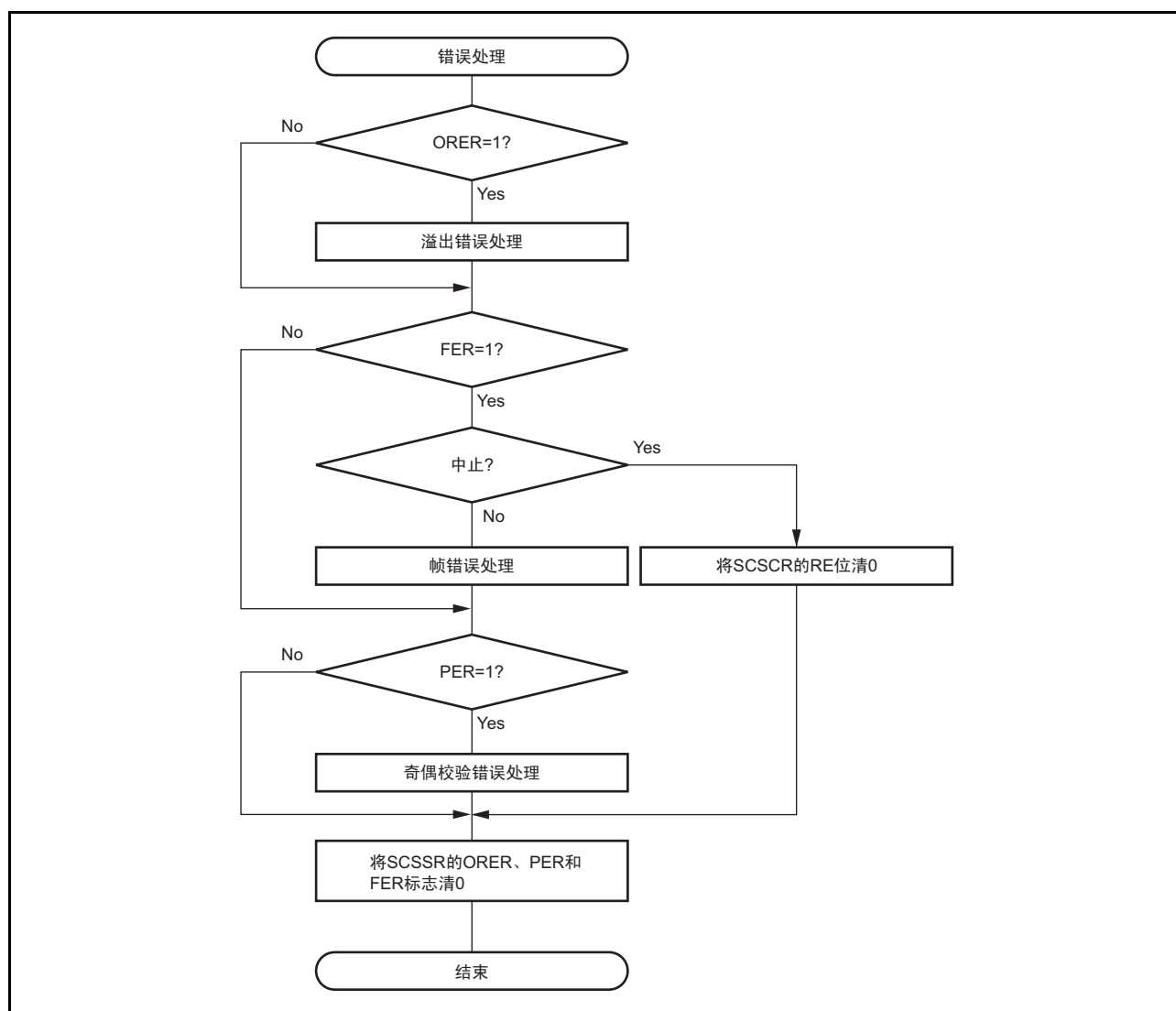


图 14.7 串行接收的流程图例子 (2)

接收时的 SCI 运行如下:

1. SCI 监视通信线路, 如果检测到起始位的 0, 就与内部同步开始接收。
2. 按照从 SCSSR 的 LSB 到 MSB 的顺序保存已接收的数据。
3. 接收奇偶校验位和停止位。

在接收后, SCI 进行以下检查:

- a. 奇偶校验的检查: 检查接收数据的 1 的个数是否为串行模式寄存器 (SCSMR) 的 O/E 位设定的偶校验或者奇校验。
- b. 停止位的检查: 检查停止位是否为 1。
但是, 在 2 个停止位的情况下, 只检查第 1 位的停止位。
- c. 状态检查: 检查 RDRF 标志是否为 0, 即是否为能将接收数据从接收移位寄存器 (SCSSR) 传送到 SCRDR 的状态。

在以上检查全部通过后, 将 RDRF 标志置 1, 并将接收数据保存到 SCRDR。

如果在错误检查时发生接收错误, 就进行如表 14.16 所示的运行。

【注】 在发生接收错误的状态下, 不能进行以后的接收。另外, 因为在接收时 RDRF 标志不被置 1, 所以必须将错误标志清 0。

4. 在RDRF标志为1时，如果SCSPTR的EIO位和SCSCR的RIE位已分别被置0和1，就产生接收数据满的中断（RXI）请求。另外，在ORER、PER和FER标志中的某个标志为1时，如果SCSCR的RIE位已被置1，就产生接收错误的中断（ERI）请求。

表 14.16 接收错误和发生条件

接收错误	略称	发生条件	数据传送
溢出错误	ORER	在 SCSSR 的 RDRF 标志被置 1 的状态下，当下一个数据接收结束时	不将接收数据从 SCRSR 传送到 SCRDR。
帧错误	FER	当停止位为 0 时	将接收数据从 SCRSR 传送到 SCRDR。
奇偶校验错误	PER	当 SCSSR 设置的偶校验 / 奇校验和接收的数据不同时	将接收数据从 SCRSR 传送到 SCRDR。

异步模式接收时的运行例子如图 14.8 所示。

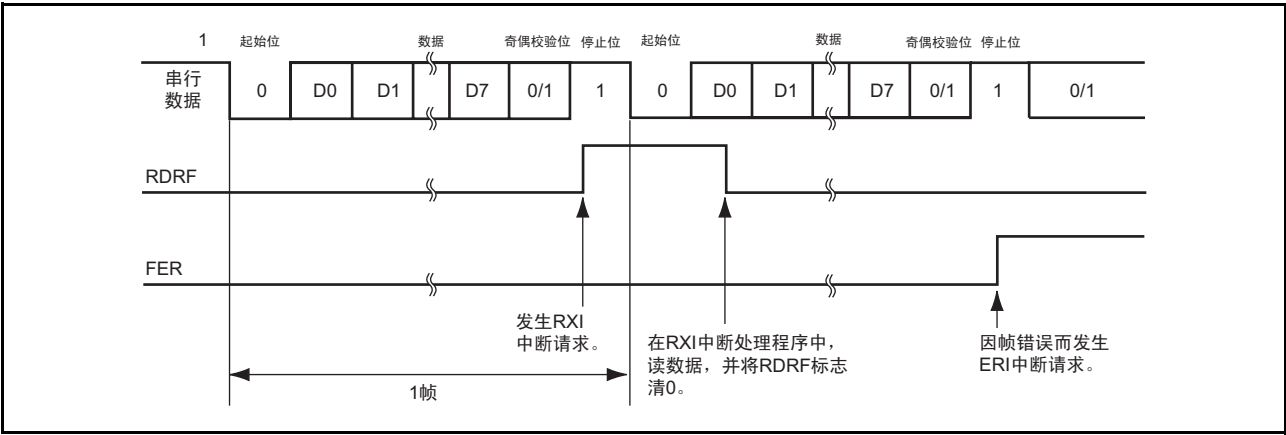


图 14.8 SCI 接收时的运行例子
(8 位数据 / 有奇偶校验 / 1 个停止位的例子)

14.4.3 时钟同步模式的运行

时钟同步模式是与时钟脉冲同步进行数据发送和接收的模式，适用于高速串行通信。因为在 SCI 内部发送部和接收部是独立的，所以能通过共享时钟进行全双工通信。另外，因为发送部和接收部都为双缓冲结构，所以能在发送和接收时读写数据，连续进行发送和接收。时钟同步串行通信的一般格式如图 14.9 所示。

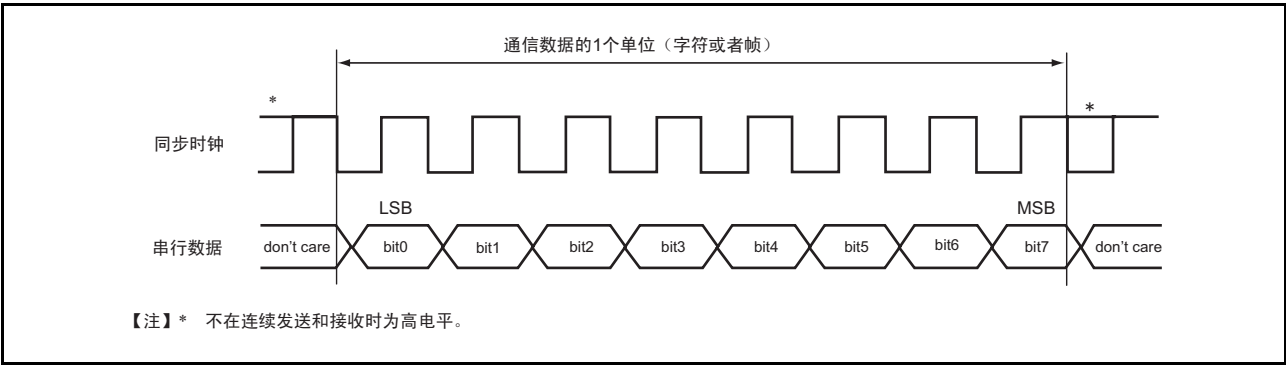


图 14.9 时钟同步通信的数据格式

在时钟同步串行通信时，在同步时钟的下降沿和下一个下降沿之间将数据输出到通信线路，并且在同步时钟的上升沿保证数据有效。

按照数据的 LSB（最初）到 MSB（最后）的顺序输出串行通信的 1 个字符，输出 MSB 后的通信线路状态保持 MSB 的状态。

在时钟同步模式中，SCI 与同步时钟的上升沿同步接收数据。

(1) 发送 / 接收格式

发送 / 接收格式固定为 8 位数据。

不能附加奇偶校验位。

(2) 时钟

能通过设定 SCSMR 的 $\overline{C/A}$ 位和 SCSCR 的 CKE1 位、CKE0 位，选择内部波特率发生器生成的内部时钟或者 SCK 引脚输入的外部同步时钟。有关 SCI 时钟源的选择请参照表 14.14。

在通过内部时钟运行时，从 SCK 引脚输出同步时钟。

同步时钟在发送和接收 1 个字符时输出 8 个脉冲，而在不发送和接收时固定为高电平。但是，当只进行接收时，在发生溢出错误前或者 RE 位被清 0 前输出同步时钟。要接收 n 个字符的数据时，必须将外部时钟设定为时钟源。在使用内部时钟时，必须先设定 RE=1 且 TE=1，然后在发送 n 个字符的虚拟数据的同时接收 n 个字符的数据。

(3) 数据的发送和接收

• SCI 的初始化（时钟同步模式）

在发送和接收数据前，必须在将串行控制寄存器（SCSCR）的 TE 位和 RE 位清 0 后按照以下步骤初始化 SCI。

在更改模式和通信格式等时，必须在将 TE 位和 RE 位清 0 后按照以下步骤进行。如果将 TE 位清 0，TDRE 标志就被置 1，发送移位寄存器（SCTSR）被初始化。

必须注意：即使将 RE 位清 0，也保持 RDRE、PER、FER、ORER 的各标志和接收数据寄存器（SCRDR）的内容。

SCI 的初始化流程图例子如图 14.10 所示。

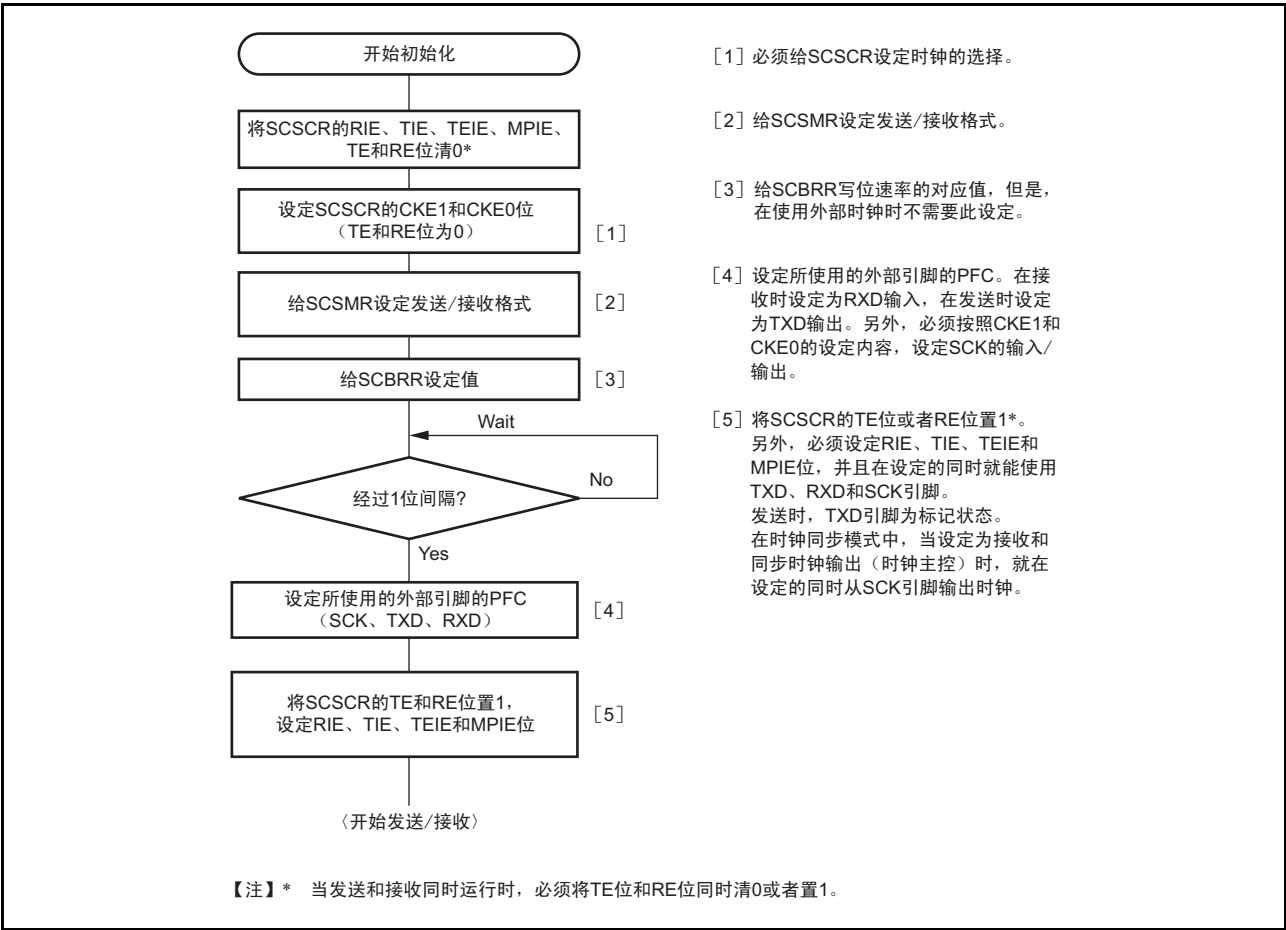


图 14.10 SCI 的初始化流程图例子

- 串行数据的发送（时钟同步模式）
串行发送的流程图例子如图 14.11 所示。
在将SCI设定为可发送的运行状态后，按照以下步骤发送串行数据。

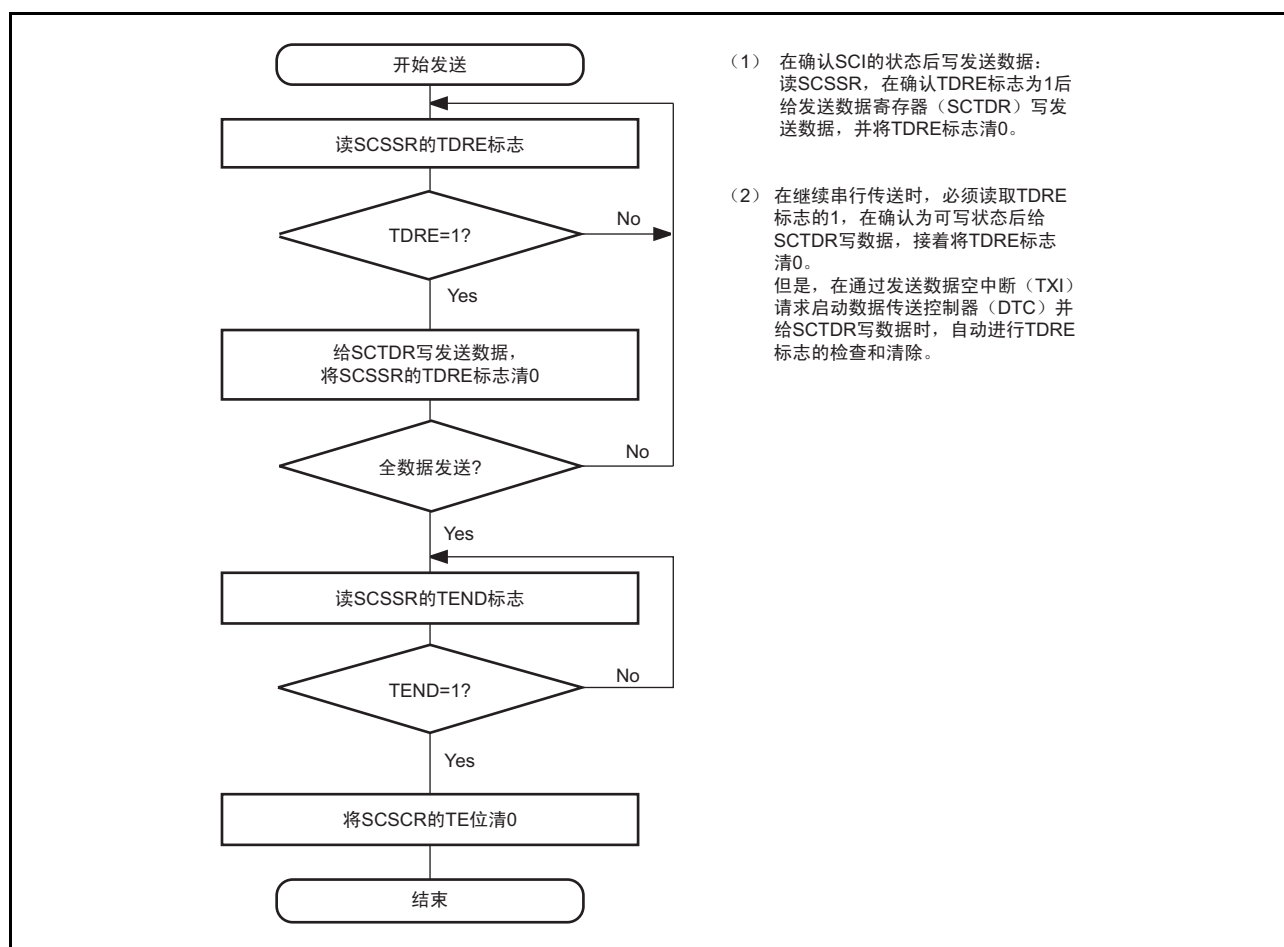


图 14.11 串行发送的流程图例子

串行发送时的 SCI 运行如下:

1. SCI 监视串行状态寄存器 (SCSSR) 的 TDRE 标志, 如果该标志为 0, 就认为已将数据写到发送数据寄存器 (SCTDR), 然后将数据从 SCTDR 传送到发送移位寄存器 (SCTSR)。
2. 在数据从 SCTDR 传送到 SCTSR 后, 将 TDRE 标志置 1, 开始发送。
此时, 如果串行控制寄存器 (SCSCR) 的发送数据空的中断允许位 (TIE) 已被置 1, 就产生发送数据空的中断 (TXI) 请求。
在设定为时钟输出模式时, SCI 输出 8 个同步时钟的脉冲。
在设定为外部时钟时, 与输入时钟同步输出数据。
按照 LSB (bit0) ~ MSB (bit7) 的顺序从 TXD 引脚送出串行发送数据。
3. SCI 在送出 MSB (bit7) 时检查 TDRE 标志。
如果 TDRE 标志为 0, 就将数据从 SCTDR 传送到 SCTSR, 开始下一帧的串行发送。
如果 TDRE 标志为 1, 就将串行状态寄存器 (SCSSR) 的 TEND 标志置 1, 在送出 MSB (bit7) 后发送数据的引脚 (TXD 引脚) 保持状态。
此时, 如果 SCSCR 的发送结束的中断允许位 (TEIE) 已被置 1, 就产生发送结束的中断请求 (TEI)。
4. 在串行发送结束后, SCK 引脚固定为高电平。
SCI 发送时的运行例子如图 14.12 所示。

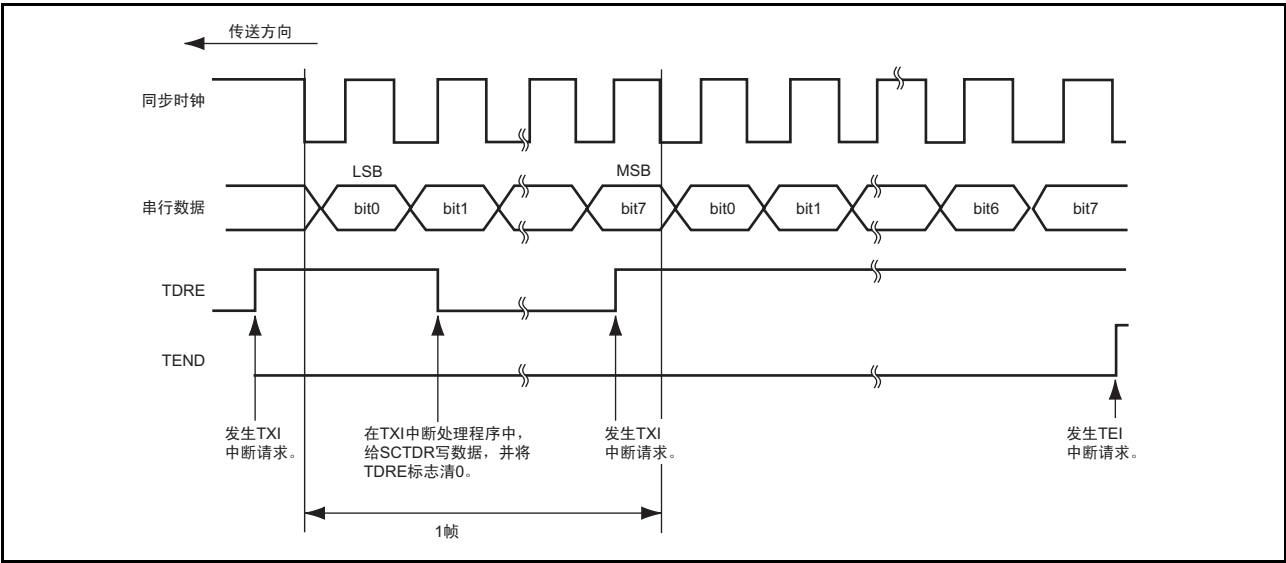


图 14.12 SCI 发送时的运行例子

- 串行数据的接收（时钟同步模式）
串行接收的流程图例子如图 14.13 和图 14.14 所示。
在将 SCI 设定为可接收的运行状态后，必须按照以下步骤接收串行数据。
在将运行模式从异步模式转换为时钟同步模式时，必须确认 ORER、PER、FER 的各标志是否已被清 0。
如果 FER、PER 标志已被置 1，RDRF 标志就不能被置位，也不能进行接收。

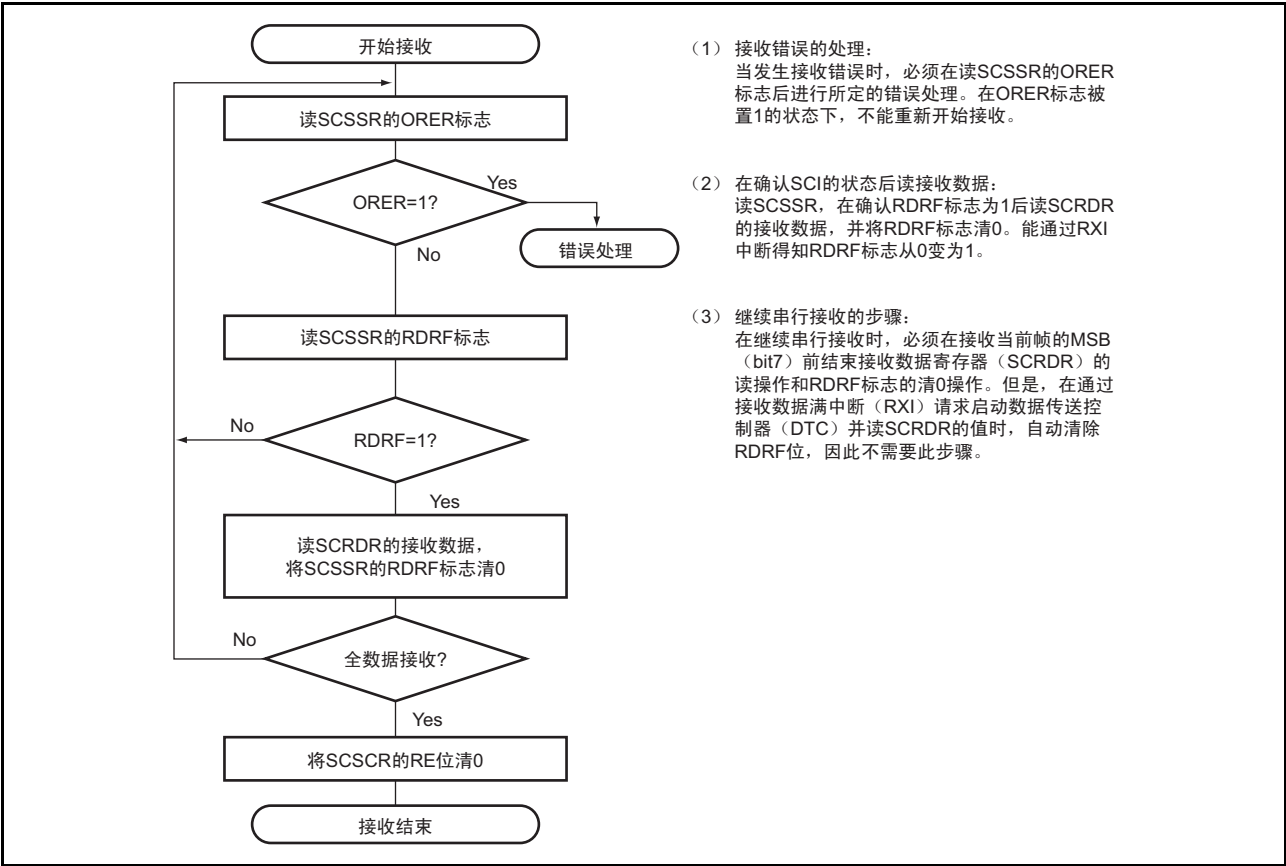


图 14.13 串行数据接收的流程图例子（1）

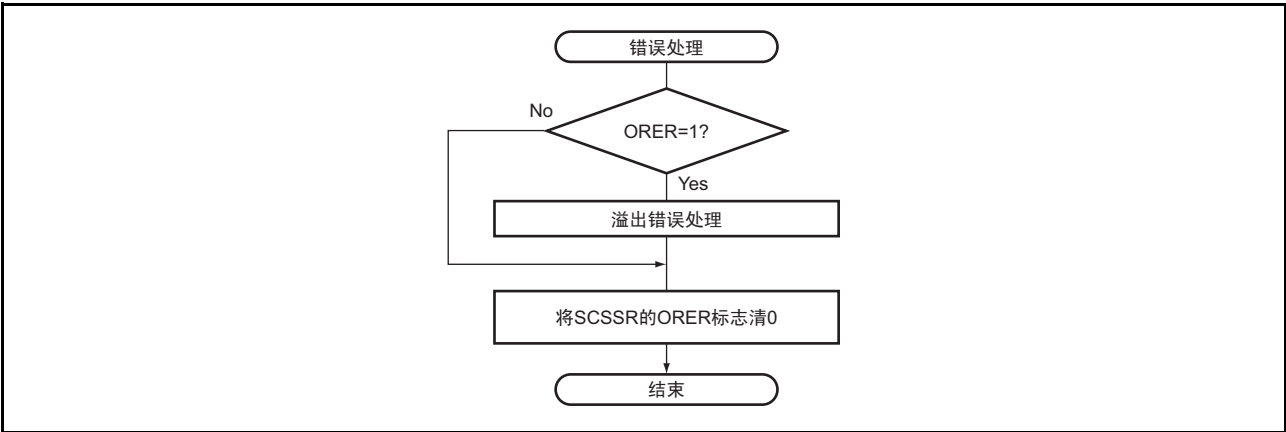


图 14.14 串行数据接收的流程图例子（2）

接收时的 SCI 运行如下：

1. SCI与同步时钟的输入或者输出同步，开始接收。
2. 按照从接收移位寄存器（SCRSR）的LSB到MSB的顺序保存接收到的数据。
在接收后，SCI检查RDRF标志是否为0，即是否为能将接收数据从SCRSR传送到接收数据寄存器（SCRDR）的状态。
在此检查通过后，将RDRF标志置1，并将接收数据保存到SCRDR。
如果在错误检查时发生接收错误，就进行如表 14.16所示的运行，在此状态下不能进行以后的发送和接收。
另外，因为在接收时RDRF标志不被置1，所以必须将此标志清0。
3. 当RDRF标志为1时，如果串行控制寄存器（SCSCR）的RIE位已被置1，就产生接收数据满的中断（RXI）请求。
在ORER标志为1时，如果SCSCR的RIE位已被置1，就产生接收错误的中断（ERI）请求。

SCI 接收时的运行例子如图 14.15 所示。

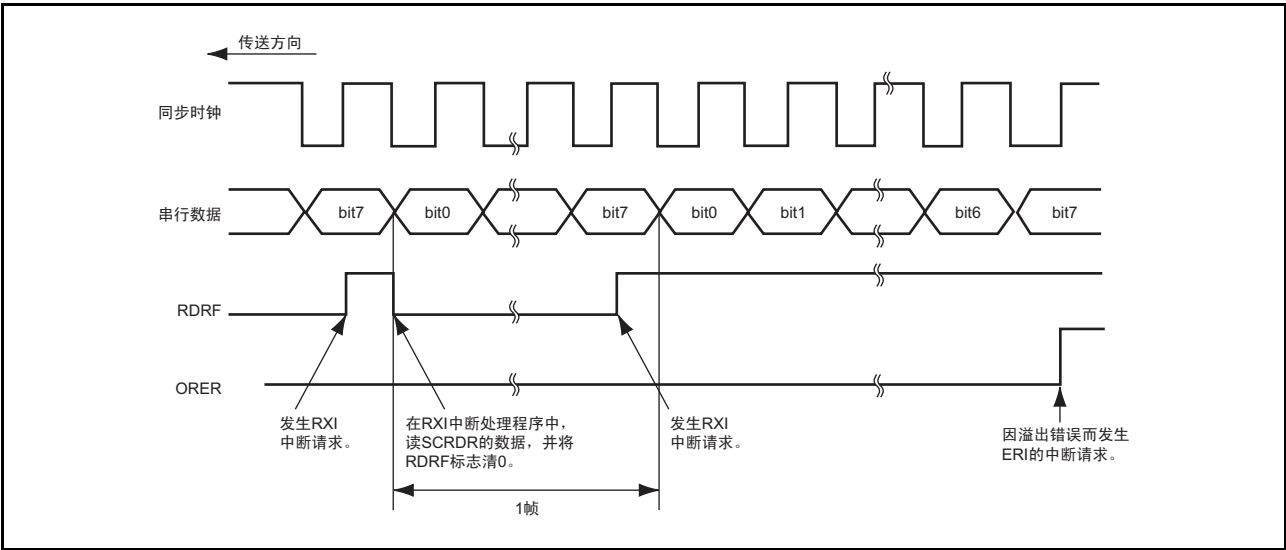


图 14.15 SCI 接收时的运行例子

- 串行数据的同时发送和接收 (时钟同步模式)

串行发送和接收同时运行的流程图例子如图 14.16 所示。

将 SCI 设定为可接收和发送的运行状态后, 必须按照以下步骤同时发送和接收串行数据。

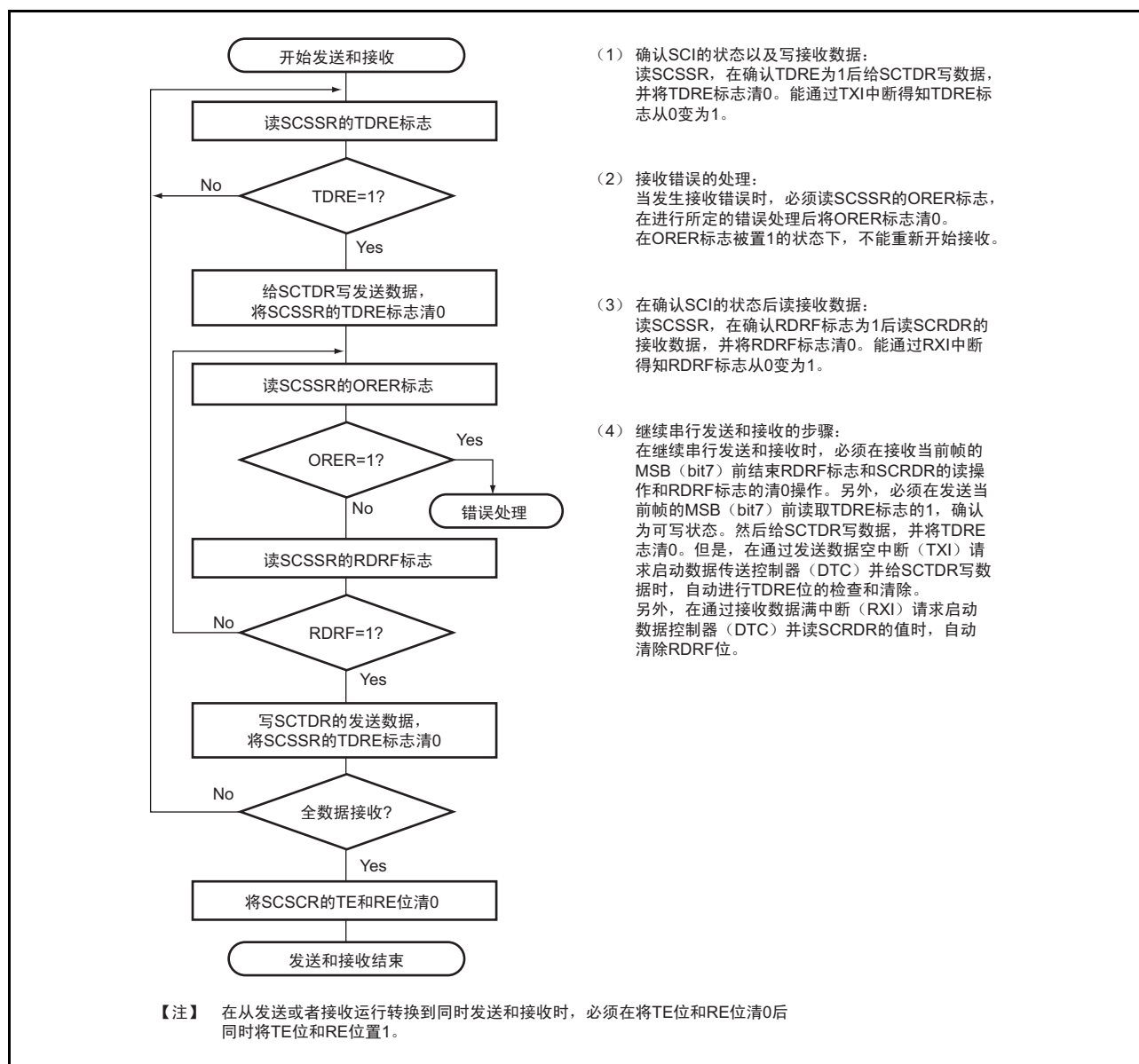


图 14.16 串行数据发送和接收的流程图例子

14.4.4 多处理器通信功能

如果使用多处理器通信功能，就能通过附加多处理器位的异步串行通信，在多个处理器之间共享通信线路进行数据的发送和接收。在多处理器通信中，给接收站分配各自特有的 ID 码。串行通信周期由指定接收站的 ID 发送周期和指定接收站的数据发送周期构成，用多处理器位区分 ID 发送周期和数据发送周期。当多处理器位是 1 时，为 ID 发送周期；当多处理器位是 0 时，为数据发送周期。使用多处理器格式的处理器之间的通信例子如图 14.17 所示。发送站首先发送多处理器位为 1 的接收站 ID 码，接着发送多处理器位为 0 的发送数据。如果接收站接收到多处理器位为 1 的通信数据，就将接收数据与本站的 ID 比较，如果一致，就继续接收被发送的通信数据。否则，就在接收到下一个多处理器位为 1 的通信数据之前，跳读通信数据。

SCI 为了支持此功能，在 SCSCR 中设有 MPIE 位。如果将 MPIE 位置 1，就在接收到多处理器位为 1 的数据之前，禁止将接收数据从 SCRSR 传送到 SCRDR，并禁止检测接收错误以及禁止将 SCSSR 的 RDRF、FER、OER 各状态标志置位。如果接收到多处理器位为 1 的字符，就在将 SCSSR 的 MPBR 位置 1 的同时自动清除 MPIE 位，然后返回到通常的接收运行状态。此时，如果 SCSCR 的 RIE 位已被置位，就产生 RXI 中断。

在指定多处理器格式时，奇偶校验位的指定无效。除此以外与通常的异步模式相同，多处理器通信时的时钟也和通常的异步模式相同。

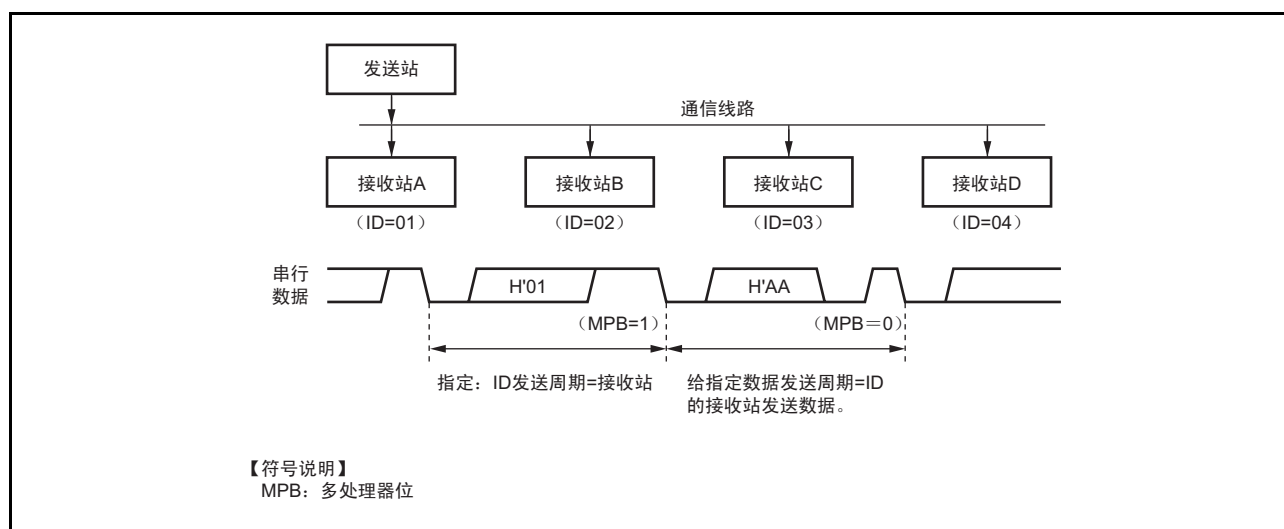


图 14.17 使用多处理器格式的通信例子
(将数据 H'AA 发送到接收站 A 的例子)

14.4.5 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 14.18 所示。必须在 ID 发送周期将 SCSSR 的 MPBT 位置 1 后发送 ID 码，在数据发送周期将 SCSSR 的 MPBT 位清 0 后发送数据。其他运行和异步模式的运行相同。

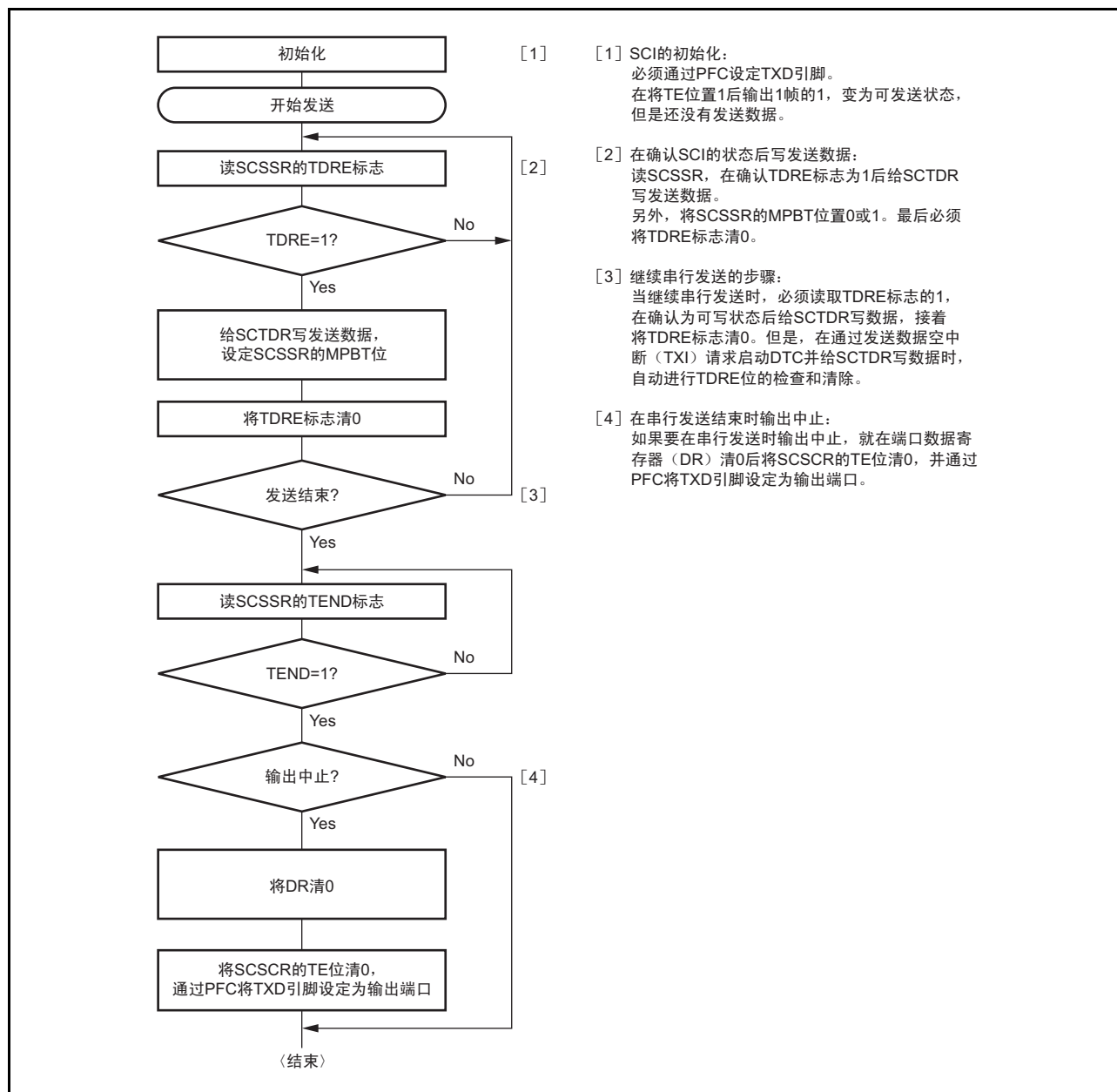


图 14.18 多处理器串行发送的流程图例子

14.4.6 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 14.20 和图 14.21 所示。如果将 SCSCR 的 MPIE 位置 1，就在接收到多处理器位为 1 的通信数据之前，跳读通信数据。如果接收到多处理器位为 1 的通信数据，就将接收数据传送到 SCRDR，此时产生 RXI 中断请求。其他运行和异步模式的运行相同。接收时的运行例子如图 14.19 所示。

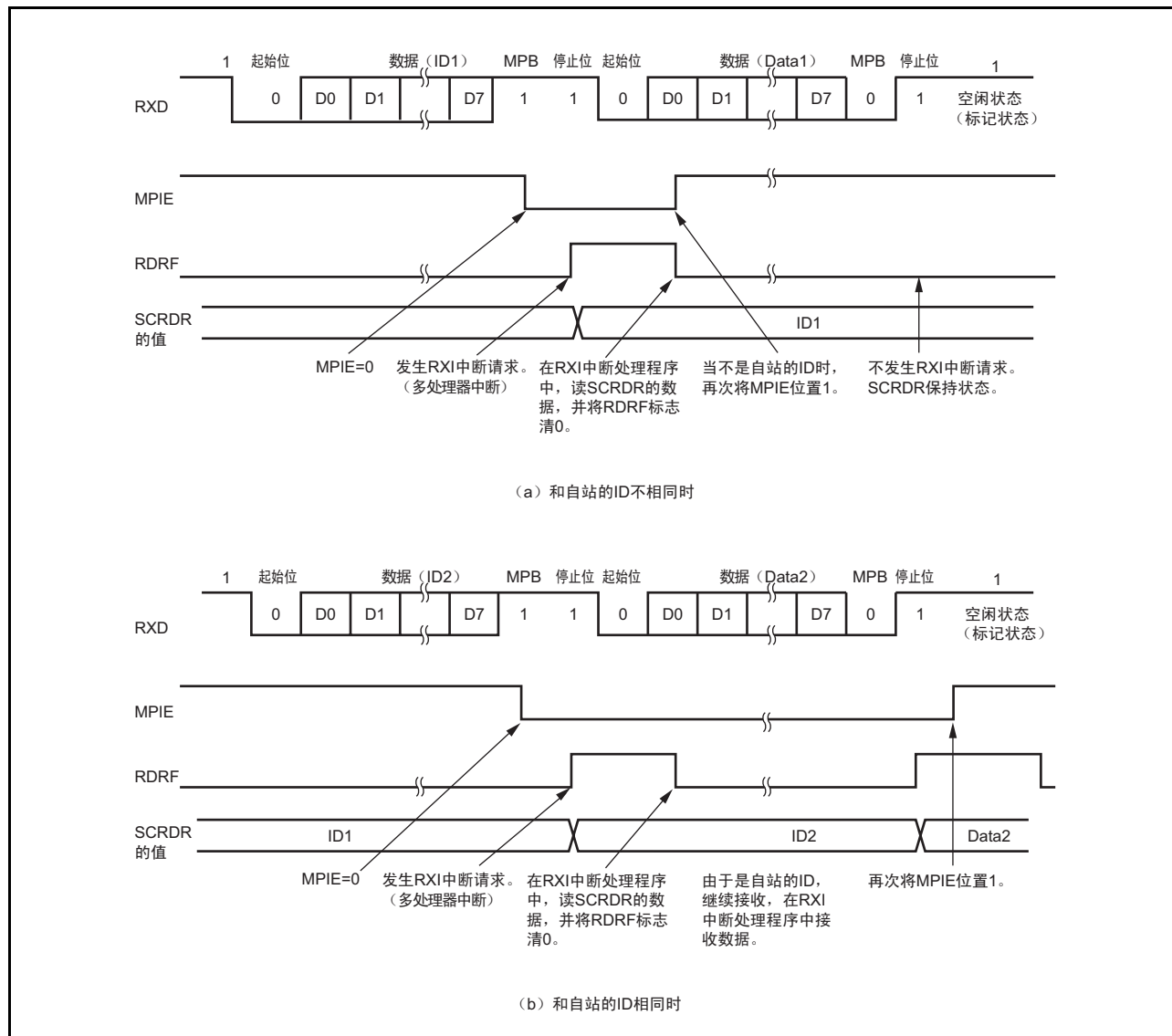


图 14.19 SCI 接收时的运行例子
(8 位数据 / 有多处理器位 / 1 个停止位的例子)

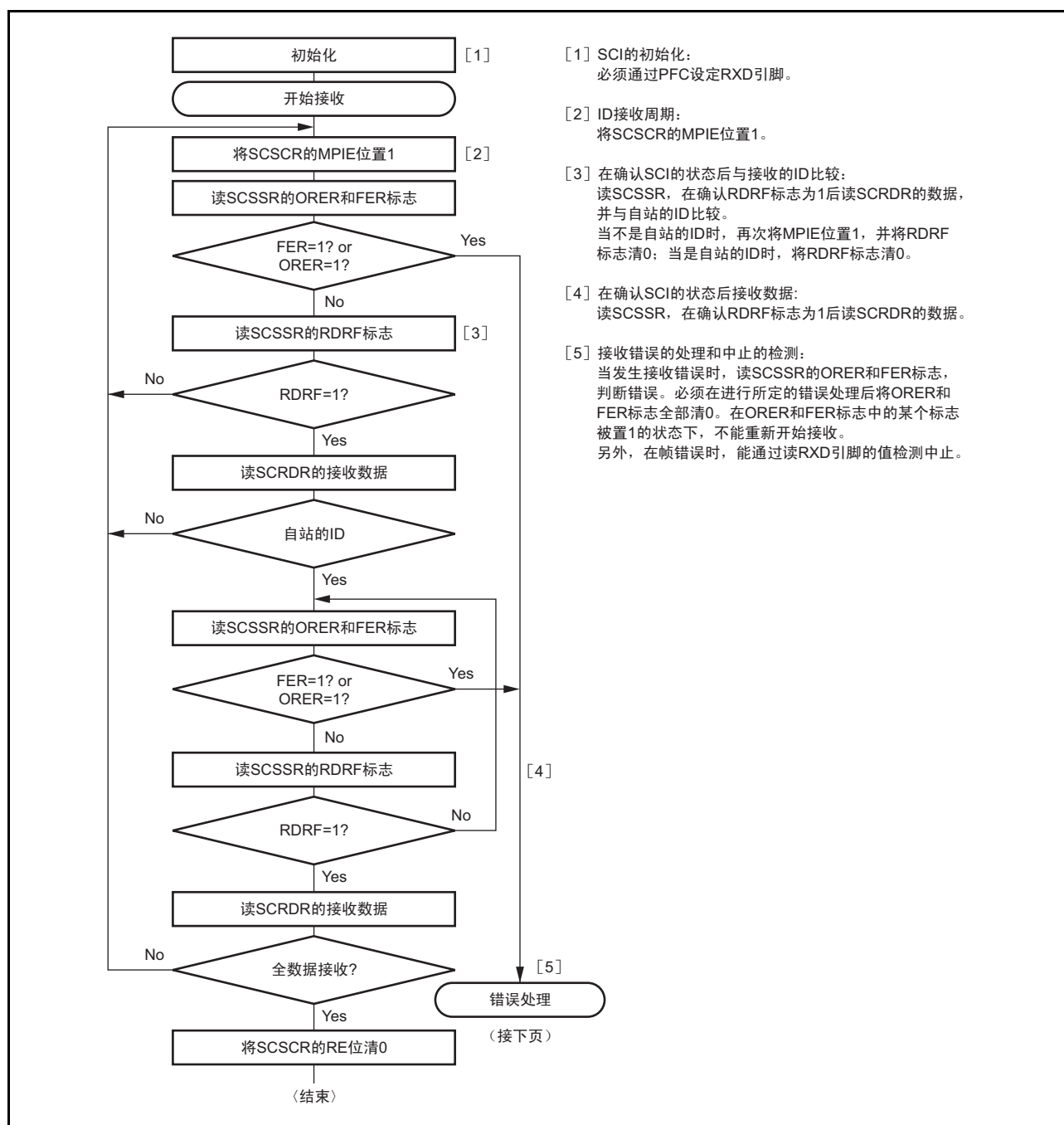


图 14.20 多处理器串行接收的流程图例子 (1)

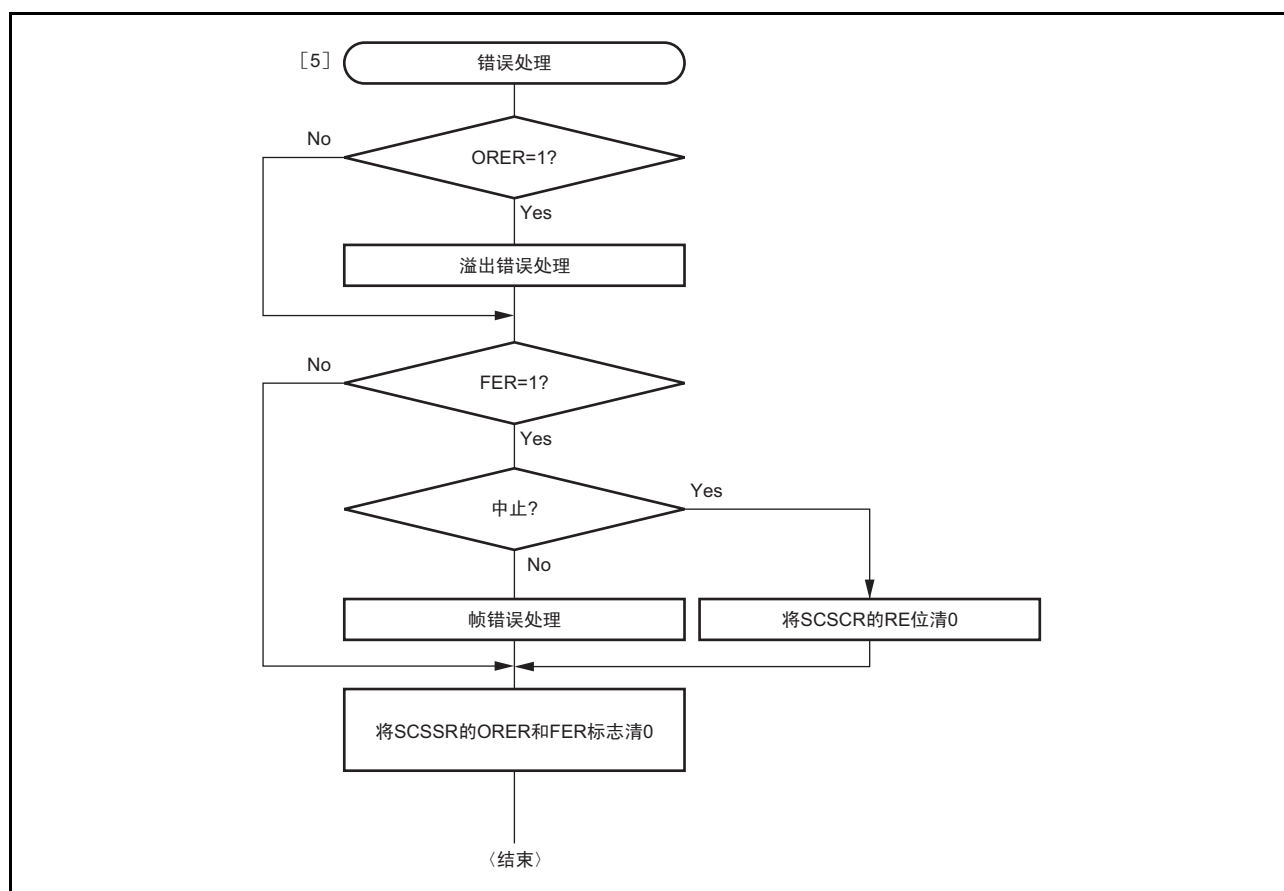


图 14.21 多处理器串行接收的流程图例子 (2)

14.5 SCI 的中断源和 DTC

SCI 有发送结束的中断 (TEI) 请求、接收错误的中断 (ERI) 请求、接收数据满的中断 (RXI) 请求和发送数据空的中断 (TXI) 请求共 4 种。

各中断源和优先级如表 14.17 所示。能通过 SCSCR 的 TIE 位、RIE 位、TEIE 位以及 SCSPTR 的 EIO 位的设定, 允许或者禁止各中断源。另外, 分别将各中断请求独立传送到中断控制器。

如果串行状态寄存器 (SCSSR) 的 TDRE 标志被置 1, 就产生 TDR 空的中断请求。能通过 TDR 空的中断请求启动数据传送控制器 (DTC), 进行数据传送。当通过 DTC 给发送数据寄存器 (SCTDR) 写数据时, 自动清除 TDRE 标志。

如果 SCSSR 的 RDRF 标志被置 1, 就产生 RDR 满的中断请求。能通过 RDR 满的中断请求启动 DTC, 进行数据传送。

通过 DTC 读接收数据寄存器 (SCRDR) 时, 自动清除 RDRF 标志。

如果 SCSSR 的 ORER、FER 标志或者 PER 位被置 1, 就产生 ERI 中断请求, 不能通过此 ERI 中断请求启动 DTC。如果通过 DTC 进行接收数据处理并通过向 CPU 请求中断进行接收错误处理, 就必须将 RIE 位和 SCSPTR 的 EIO 位同时置 1, 使中断错误只发生在接收错误时。如果将 EIO 位置 0, 即使在正常接收数据时, 也对 CPU 产生中断。

如果 SCSSR 的 TEND 标志被置 1, 就产生 TEI 中断请求, 不能通过此 TEI 中断请求启动 DTC。

另外, TXI 中断表示可写发送数据, TEI 中断表示发送结束。

表 14.17 SCI 中断源

中断源	内容	DTC 的启动
ERI	接收错误 (ORER、FER、PER) 的中断	不可
RXI	接收数据满 (RDRF) 的中断	可
TXI	发送数据空 (TDRE) 的中断	可
TEI	发送结束 (TEND) 的中断	不可

14.6 串行端口寄存器 (SCSPTR) 和 SCI 引脚的关系

SCSPTR 和 SCI 引脚的关系如图 14.22 和图 14.23 所示。

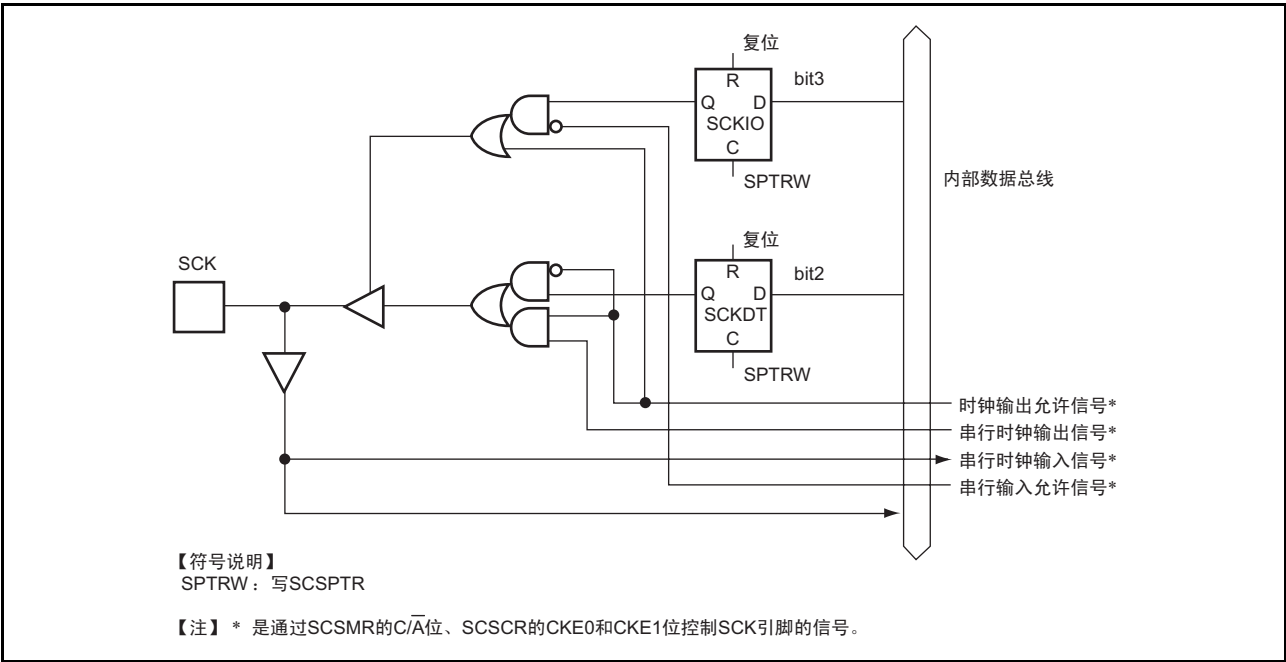


图 14.22 SCKIO 位、SCKDT 位和 SCK 引脚的关系

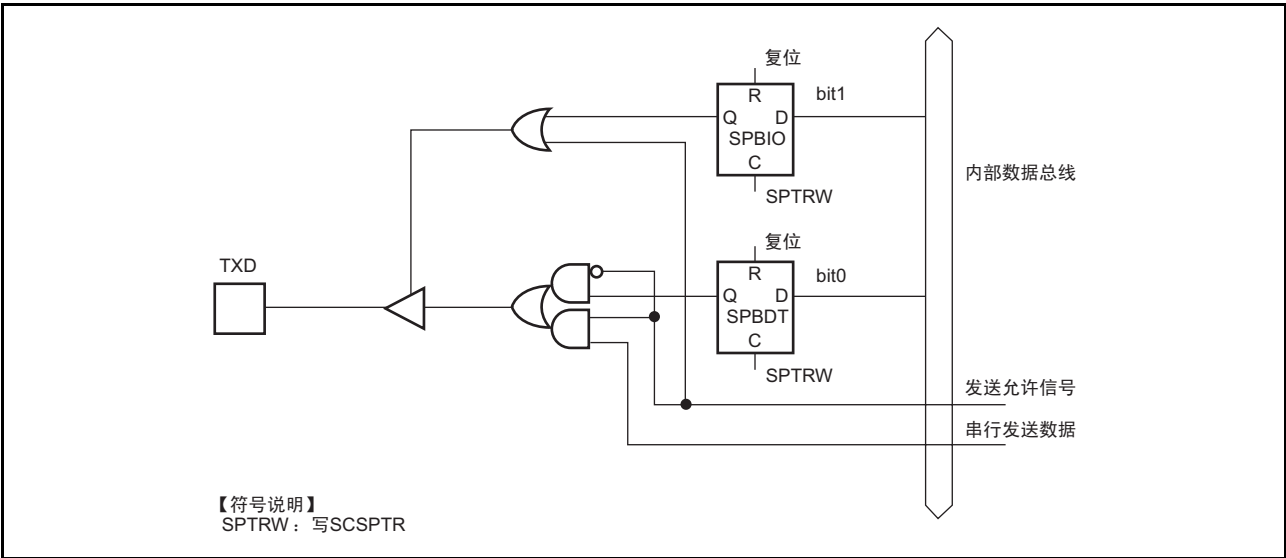


图 14.23 SPBIO 位、SPBDT 位和 TXD 引脚的关系

14.7 使用时的注意事项

14.7.1 有关 SCTDR 的写操作和 TDRE 标志的关系

串行状态寄存器 (SCSSR) 的 TDRE 标志是表示已将发送数据从发送数据寄存器 (SCTDR) 传送到发送移位寄存器 (SCTSR) 的状态标志。如果 SCI 将数据从 SCTDR 传送到 SCTSR, TDRE 位就被置 1。

与 TDRE 标志无关, 能将数据写到 SCTDR。但是, 如果在 TDRE 标志为 0 的状态下将新数据写到 SCTDR, 因为还没有传送到 SCTSR, 所以被保存到 SCTDR 的数据就会丢失。因此, 必须在确认 TDRE 标志已被置 1 后给 SCTDR 写发送数据。

14.7.2 有关同时发生多个接收错误时的运行

当同时发生多个接收错误时, SCSSR 各状态标志的状态如表 14.18 所示。另外, 在发生溢出错误时, 因为不将数据从接收移位寄存器 (SCRSR) 传送到接收数据寄存器 (SCRDR), 所以接收数据就会丢失。

表 14.18 SCSSR 状态标志的状态和接收数据的传送

接收错误的状态	SCSSR 的状态标志				接收数据的传送 SCRSR→SCRDR
	RDRF	ORER	FER	PER	
溢出错误	1	1	0	0	×
帧错误	0	0	1	0	○
奇偶校验错误	0	0	0	1	○
溢出错误 + 帧错误	1	1	1	0	×
溢出错误 + 奇偶校验错误	1	1	0	1	×
帧错误 + 奇偶校验错误	0	0	1	1	○
溢出错误 + 帧错误 + 奇偶校验错误	1	1	1	1	×

【符号说明】

○: 将接收数据从 SCRSR 传送到 SCRDR。

×: 不将接收数据从 SCRSR 传送到 SCRDR。

14.7.3 中止的检测和处理

在检测帧错误 (FER) 时, 能通过直接读 RXD 引脚的值检测中止。在中止时, RXD 引脚的输入始终为 0, 所以 FER 标志被置位, 并且奇偶校验错误 (PER) 位也可能被置位。

必须注意: SCI 在接收中止后还继续接收, 但是接收到的数据不传送到 SCRDR。

14.7.4 中止的发送

能根据串行端口寄存器 (SCSPTR) 的 SPB0IO 位和 SPB0DT 位决定 TXD 引脚的输入 / 输出条件和电平, 所以利用这一功能可以发送中止。在从串行发送的初始化到 TE 位被置 1 (能发送) 的期间, TXD 引脚不工作。在此期间, 标记状态被 SPB0DT 位的值代替。因此, 要先将 SPB0IO 位和 SPB0DT 位置 1 (输出高电平)。

如果在串行发送时要中止发送, 就必须在将 SPB0DT 位清 0 (低电平) 后将 TE 位清 0 (停止发送)。如果将 TE 位清 0, 就与当前的发送状态无关, 初始化发送部并从 TXD 引脚输出 0。

14.7.5 异步模式的接收数据采样时序和接收容限

在异步模式中，SCI 通过频率为 16 倍位速率的基本时钟运行。

在接收时，SCI 通过基本时钟对起始位的下降沿进行采样，取得内部同步，而在基本时钟的第 8 个时钟脉冲的上升沿将接收数据取到内部。

如图 14.24 所示。

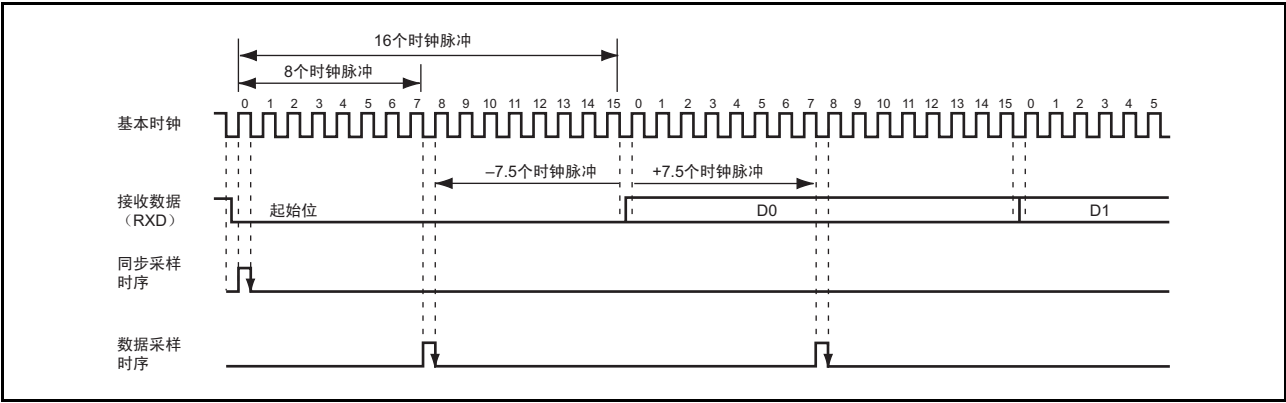


图 14.24 异步模式的接收数据采样时序

因此，能用表达式 (1) 表示异步模式的接收容限。

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5)F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100[\%] \quad \dots \text{表达式 (1)}$$

M: 接收容限 (%)

N: 对应时钟的位速率比 (N=16)

D: 时钟占空比 (D=0 ~ 1.0)

L: 帧长 (L=9 ~ 12)

F: 时钟频率的偏差绝对值

当表达式 (1) 中的 F=0、D=0.5 时，根据表达式 (2)，接收容限为 46.875%。

当 D=0.5、F=0 时

$$M = (0.5 - 1/(2 \times 16)) \times 100\% = 46.875\% \quad \dots \text{表达式 (2)}$$

此值为计算值，所以在系统设计时必须留有 20 ~ 30% 的余地。

14.7.6 DTC 使用时的注意事项

1. 当将外部时钟源用于同步时钟时，必须在通过 DTC 更新 SCTDR 后至少经过 5 个外围运行时钟周期，然后输入外部时钟。如果在更新 SCTDR 后的 4 个周期以内输入发送时钟，就可能发生误动作 (参照图 14.25)。

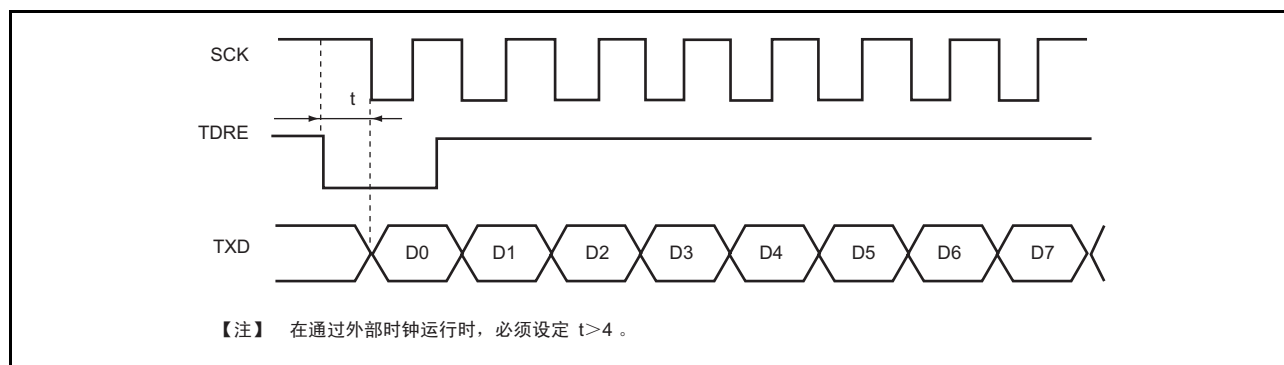


图 14.25 通过 DTC 进行同步时钟传送的例子

2. 在通过TXI中断启动DTC后给SCTDR写数据时，TEND标志为不定值，所以TEND标志不能用作发送结束标志。

14.7.7 时钟同步外部时钟模式的注意事项

必须在将外部时钟 SCK 从 0 变为 1 后至少经过 4 个外围运行时钟周期，然后设定 TE=1、RE=1。

必须在外部时钟 SCK 为 1 时设定 TE=RE=1。

14.7.8 模块待机模式的设定

SCI 能通过待机控制寄存器，禁止或者允许本模块的运行。初始值为 SCI 停止运行。能通过解除模块待机模式允许寄存器的存取。详细内容请参照“第 24 章 低功耗模式”。

第 15 章 同步串行通信单元 (SSU)

本 LSI 有 1 个通道的同步串行通信单元 (SSU: Synchronous Serial communication Unit)。SSU 有主控模式 (以本 LSI 为主控器件, 将时钟输出到外部进行同步串行通信) 和从属模式 (从外部器件输入时钟进行同步串行通信), 能与不同时钟极性和不同时钟相位的器件进行同步串行通信。

15.1 特点

- 能选择 SSU 模式和时钟同步通信模式
- 能选择主控模式和从属模式
- 能选择标准模式和双向模式
- 能与不同时钟相位和不同时钟极性的其他器件进行同步串行通信
- 发送/接收数据长度能选择 8 位、16 位或者 32 位
- 能进行全双工通信
装载了能同时发送和接收的移位寄存器
- 能进行连续的串行通信
- 能选择 LSB 先方式或者 MSB 先方式
- 能选择 7 种内部时钟 (Pφ/4、Pφ/8、Pφ/16、Pφ/32、Pφ/64、Pφ/128、Pφ/256) 和外部时钟作为时钟源
- 中断源: 5 种
发送结束、发送数据空、接收数据满、溢出错误和冲突错误中断源
能通过发送数据空请求或者接收数据满请求, 启动数据传送控制器 (DTC) 进行数据传送。
- 能设定模块待机模式

SSU 的框图如图 15.1 所示。

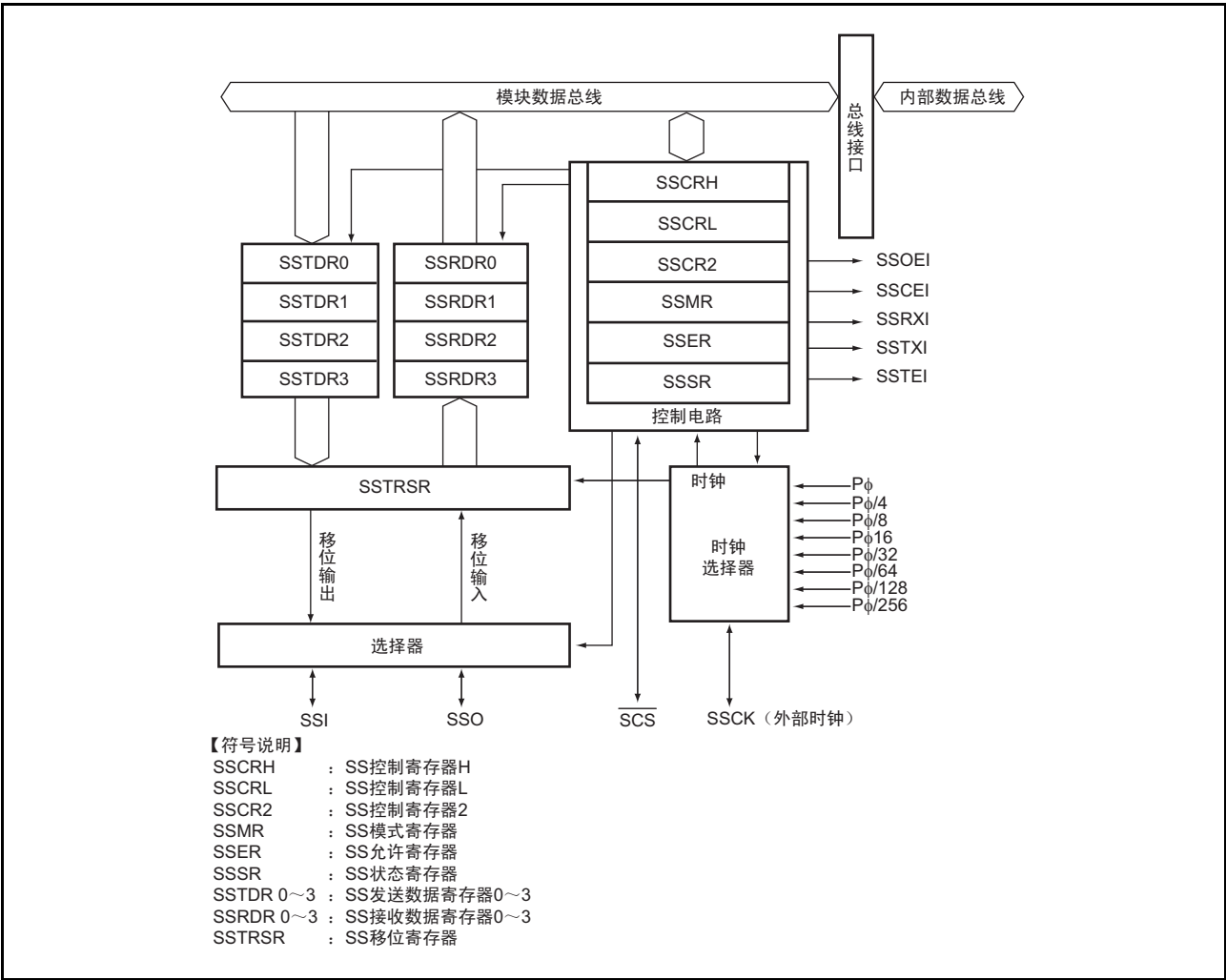


图 15.1 SSU 的框图

15.2 输入 / 输出引脚

SSU 有表 15.1 中的输入 / 输出引脚。

表 15.1 引脚结构

引脚名	输入 / 输出	功能
SSCK	输入 / 输出	SSU 的时钟输入 / 输出引脚
SSI	输入 / 输出	SSU 的数据输入 / 输出引脚
SSO	输入 / 输出	SSU 的数据输入 / 输出引脚
SCS	输入 / 输出	SSU 的片选输入 / 输出引脚

15.3 寄存器说明

SSU 有以下寄存器。有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 25 章 寄存器一览”。

表 15.2 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
SS 控制寄存器 H	SSCRH	R/W	H'0D	H'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

15.3.1 SS 控制寄存器 H (SSCRH)

SSCRH 进行主控制器件 / 从属器件的选择、双向模式的允许、SSO 引脚输出值的选择以及 SSCK 引脚和 SCS 引脚的选择。

位:	7	6	5	4	3	2	1	0
	MSS	BIDE	—	SOL	SOLP	—	CSS[1:0]	
初始值:	0	0	0	0	1	1	0	1
R/W:	R/W	R/W	R	R/W	R/W	R	R/W	R/W

位	位名	初始值	R/W	说明
7	MSS	0	R/W	主控制器件 / 从属器件的选择 选择是将 SSU 用作主控模式还是用作从属模式。当用作主控模式时，从 SSCK 引脚输出传送时钟。如果 SSSR 的 CE 位被置位，此位就自动清除。 0: 选择从属模式 1: 选择主控模式
6	BIDE	0	R/W	双向模式的允许 选择是使用串行数据输入和输出 2 个引脚还是只使用 1 个引脚。如果选择双向模式，就不能同时发送和接收。详细内容请参照“15.4.3 数据输入 / 输出引脚和移位寄存器的关系”。 0: 标准模式（使用数据输入和数据输出 2 个引脚进行通信） 1: 双向模式（只使用 1 个引脚进行数据输入和数据输出的通信）

位	位名	初始值	R/W	说明
5	—	0	R	保留位 读写值总是 0。
4	SOL	0	R/W	串行数据输出值的选择 在发送结束后, 串行数据的输出保持发送数据最后位的值, 但是能在发送前或者发送后更改串行数据的输出电平。当要更改输出电平时, 必须用 MOV 指令将 SOLP 位置 0。另外, 如果在数据传送中写此位, 就会产生误动作, 所以不能在发送中更改此位。 0: 将串行数据的输出电平更改为 Low 电平 1: 将串行数据的输出电平更改为 High 电平
3	SOLP	1	R/W	SOL 位的写保护 当要更改串行数据的输出电平时, 必须用 MOV 指令设定 SOL=1 且 SOLP=0、或者 SOL=0 且 SOLP=0。 0: 能通过 SOL 的值更改输出电平 1: 不能通过 SOL 的值更改输出电平 读取值总是 1。
2	—	1	R	保留位 读写值总是 1。
1、0	CSS[1:0]	01	R/W	$\overline{SCS}$ 引脚的选择 选择是将 $\overline{SCS}$ 引脚用作 $\overline{SCS}$ 输入还是用作 $\overline{SCS}$ 输出。 00: 禁止设定 01: 用作 $\overline{SCS}$ 输入 10: 用作 $\overline{SCS}$ 自动输入 / 输出 (在传送前后为 $\overline{SCS}$ 输入, 在传送中输出 Low 电平) 11: 用作 $\overline{SCS}$ 自动输出 (在传送前后为 High 电平输出, 在传送中输出 Low 电平)

15.3.2 SS 控制寄存器 L (SSCRL)

SSCRL 选择运行模式、软件复位以及发送 / 接收数据的长度。

位:	7	6	5	4	3	2	1	0
	FCLRM	SSUMS	SRES	—	—	—	DATS[1:0]	
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R/W	R/W

位	位名	初始值	R/W	说明
7	FCLRM	0	R/W	标志清除模式 选择是在写 SSTDR 或者读 SSRDR 时还是在 DTC 传送结束时清除 SSRXI 和 SSTXI 的中断标志。在使用 DTC 时, 必须将此位置 0。 0: 在 DTC 传送结束时 1: 在存取寄存器时
6	SSUMS	0	R/W	SSU 模式和时钟同步通信模式的选择。 0: SSU 模式 1: 时钟同步通信模式

位	位名	初始值	R/W	说明
5	SRES	0	R/W	软件复位 如果将此位置 1，就对 SSU 的内部定序器进行强制复位，然后自动清除此位，并对 SSSR 的 ORER 位、TEND 位、TDRE 位、RDRF 位、CE 位和 SSER 的 TE 位、RE 位进行初始化，保持其他 SSU 内部寄存器的值。 要在中途中断传送时，必须给此位写 1，并对内部定序器进行复位。
4 ~ 2	—	全 0	R	保留位 读写值总是 0。
1、0	DATS[1:0]	00	R/W	发送 / 接收数据长度的选择 选择串行数据的长度。 00: 8 位 01: 16 位 10: 32 位 11: 设定无效

15.3.3 SS 模式寄存器 (SSMR)

SSMR 选择 MSB 先 /LSB 先、时钟极性、时钟相位和传送时钟速率。

位:	7	6	5	4	3	2	1	0
	MLS	CPOS	CPHS	—	—	CKS[2:0]		
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7	MLS	0	R/W	MSB 先 /LSB 先的选择 选择是以 MSB 先方式还是以 LSB 先方式进行串行数据的传送。 0: LSB 先 1: MSB 先
6	CPOS	0	R/W	时钟极性的选择 选择 SSCK 时钟的极性。 0: 空闲时为 High 电平输出，激活时为 Low 电平输出 1: 空闲时为 Low 电平输出，激活时为 High 电平输出
5	CPHS	0	R/W	时钟相位的选择 (只在 SSU 模式中有效) 选择 SSCK 时钟的相位。 0: 在最初的边沿，数据发生变化 1: 在最初的边沿，数据被锁存
4、3	—	全 0	R	保留位 读写值总是 0。
2 ~ 0	CKS[2:0]	000	R/W	传送时钟速率的选择 在选择内部时钟时，选择传送时钟速率 (预分频器的分频比)。 000: 保留 001: P ϕ /4 010: P ϕ /8 011: P ϕ /16 100: P ϕ /32 101: P ϕ /64 110: P ϕ /128 111: P ϕ /256

15.3.4 SS 允许寄存器 (SSER)

SSER 设定发送和接收的允许以及中断请求的允许。

位:	7	6	5	4	3	2	1	0
	TE	RE	—	—	TEIE	TIE	RIE	CEIE
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7	TE	0	R/W	发送的允许 当此位被置 1 时, 能进行发送。
6	RE	0	R/W	接收的允许 当此位被置 1 时, 能进行接收。
5、4	—	全 0	R	保留位 读写值总是 0。
3	TEIE	0	R/W	发送结束中断的允许 当此位被置 1 时, 允许 SSTEI 中断请求。
2	TIE	0	R/W	发送中断的允许 当此位被置 1 时, 允许 SSTXI 中断请求。
1	RIE	0	R/W	接收中断的允许 当此位被置 1 时, 允许 SSRXI 中断请求和 SSOEI 中断请求。
0	CEIE	0	R/W	冲突错误中断的允许 当此位被置 1 时, 允许 SSCEI 中断请求。

15.3.5 SS 状态寄存器 (SSSR)

SSSR 是各种中断的状态标志寄存器。

位:	7	6	5	4	3	2	1	0
	—	ORER	—	—	TEND	TDRE	RDRF	CE
初始值:	0	0	0	0	0	1	0	0
R/W:	R	R/W	R	R	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7	—	0	R	保留位 读写值总是 0。
6	ORER	0	R/W	溢出错误 此位表示在 RDRF=1 的状态下接收下一个数据时, 因发生溢出错误而异常结束。SSRDR 保持溢出错误发生前的 1 帧接收数据, 后接收的数据会丢失。另外, 不能在 ORER=1 的状态下继续进行串行接收和串行发送。 [置位条件] - 在 RDRF=1 的状态下, 当下一个串行接收的 1 字节数据接收结束时 [清除条件] - 当读到 1 的状态后写 0 时

位	位名	初始值	R/W	说明
5、4	—	全 0	R	保留位 读写值总是 0。
3	TEND	0	R/W	发送结束 [置位条件] - 在 SSCR2 的 TENDSTS 为 0 并在 TDRE=1 的状态下, 发送数据的最后位时 - 在 SSCR2 的 TENDSTS 为 1 并在 TDRE=1 的状态下, 发送数据的最后位后 [清除条件] - 当读到 TEND=1 的状态后给 TEND 标志写 0 时 - 当将数据写到 SSTDR 时
2	TDRE	1	R/W	发送数据空 表示 SSTDR 内有无数据。 [置位条件] - 当 SSER 的 TE 为 0 时 - 当将数据从 SSTDR 传送到 SSTRSR 并能将数据写到 SSTDR 时 [清除条件] - 当读到 TDRE=1 的状态后给 TDRE 标志写 0 时 - 当 TE=1 并将数据写到 SSTDR 时 - 在通过 SSTXI 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位为 0 的状态下, 将传送数据写到 SSTDR 时
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] - 当串行接收正常结束并将接收数据从 SSTRSR 传送到 SSRDR 时 [清除条件] - 当读到 RDRF=1 的状态后给 RDRF 标志写 0 时 - 当从 SSRDR 读接收数据时 - 在通过 SSRXI 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位为 0 的状态下, 从 SSRDR 读接收数据时
0	CE	0	R/W	冲突错误 / 不完全错误 此位表示在 SSUMS=0 (SSU 模式) 并且 MSS=1 (主控器件) 的状态下, 因从外部给 $\overline{\text{SCS}}$ 输入 0 而发生冲突错误; 在 SSUMS=0 (SSU 模式) 并且 MSS=0 (从属器件) 的状态下, 因在 $\overline{\text{SCS}}$ 引脚变为 1 时判断为主控器件已停止传送而发生不完全错误。另外, 在 CE=1 的状态下, 不能继续进行串行接收和串行发送。必须在重新开始发送前, 将 SSCRL 的 SRES 位置 1, 对内部定时器进行复位。 [置位条件] - 当 SSCRH 的 MSS=1 (主控器件) 并给 $\overline{\text{SCS}}$ 引脚输入 Low 电平时 - 当 SSCRH 的 MSS=0 (从属器件) 并在传送中 $\overline{\text{SCS}}$ 引脚变为 1 时 [清除条件] - 当读到 1 的状态后写 0 时

15.3.6 SS 控制寄存器 2 (SSCR2)

SSCR2 设定 $\overline{\text{SCS}}$ 引脚的有效时序、SSO 引脚的数据输出时序和 TEND 位的置位时序。

位:	7	6	5	4	3	2	1	0
	—	—	—	TENDSTS	SCSATS	SSODTS	—	—
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R	R

位	位名	初始值	R/W	说明
7 ~ 5	—	全 0	R	保留位 读写值总是 0。
4	TENDSTS	0	R/W	选择 TEND 位的置位时序 (只在设定 SSU 模式和主控模式时有效) 0: 在发送最后位时, 将 TEND 位置位 1: 在发送最后位后, 将 TEND 位置位
3	SCSATS	0	R/W	选择 $\overline{\text{SCS}}$ 引脚的有效时序 (只在设定 SSU 模式和主控模式时有效) 0: t_{LEAD} 和 t_{LAG} 的输出期间 Min. 为 $1/2 \times t_{\text{SUcyc}}$ 1: t_{LEAD} 和 t_{LAG} 的输出期间 Min. 为 $3/2 \times t_{\text{SUcyc}}$
2	SSODTS	0	R/W	选择 SSO 引脚的数据输出时序 (只在设定 SSU 模式和主控模式时有效) 0: 在 BIDE=0 且 MSS=1 且 TE=1、或者 BIDE=1 且 TE=1 且 RE=0 时, SSO 引脚输出数据 1: 在 BIDE=0 且 $\overline{\text{MSS}}$ =1 且 TE=1、或者 BIDE=1 且 TE=1 且 RE=0 时, 并且在 $\overline{\text{SCS}}$ 引脚为 Low 电平期间, SSO 引脚输出数据
1、0	—	全 0	R	保留位 读写值总是 0。

15.3.7 SS 传送数据寄存器 0 ~ 3 (SSTDR0 ~ SSTDR3)

SSTDR 是保存发送数据的 8 位寄存器。当通过 SSCRL 的 DATS1 位和 DATS0 位的设定来选择 8 位数据长时, SSTDR0 有效; 当选择 16 位数据长时, SSTDR0 和 SSTDR1 有效; 当选择 32 位数据长时, SSTDR0、SSTDR1、SSTDR2 和 SSTDR3 有效。不能存取无效的 SSTDR。

如果检测到 SSTRSR 为空, SSU 就将写在 SSTDR 的发送数据传送到 SSTRSR, 开始串行发送。如果在发送 SSTRSR 的串行数据时预先将下一个数据写到 SSTDR, 就能进行连续的串行发送。

CPU 和 DTC 随时能读写 SSTDR, 但是, 为了可靠地进行串行通信, 必须在确认 SSSR 的 TDRE 已被置 1 后写 SSTDR。

位:	7	6	5	4	3	2	1	0
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7 ~ 0		全 0	R/W	串行发送数据

表 15.3 SSCRL 的 DATS 位的设定和 SSTDR 的对应表

	DATS[1:0] 的设定			
	00	01	10	11 (设定无效)
SSTDR0	有效	有效	有效	无效
SSTDR1	无效	有效	有效	无效
SSTDR2	无效	无效	有效	无效
SSTDR3	无效	无效	有效	无效

15.3.8 SS 接收数据寄存器 0 ~ 3 (SSRDR0 ~ SSRDR3)

SSRDR 是保存接收数据的 8 位寄存器。当通过 SSCRL 的 DATS1 位和 DATS0 位的设定来选择 8 位数据长时, SSRDR0 有效; 当选择 16 位数据长时, SSRDR0 和 SSRDR1 有效; 当选择 32 位数据长时, SSRDR0、SSRDR1、SSRDR2 和 SSRDR3 有效。不能存取无效的 SSRDR。

当 1 字节的数据接收结束时, SSU 将接收的串行数据从 SSTRSR 传送并保存到 SSRDR, 然后 SSTRSR 变为可接收状态。因为 SSTRSR 和 SSRDR 为双缓冲结构, 所以能进行连续的接收。

必须在确认 SSSR 寄存器的 RDRF 位已被置 1 后才能读 SSRDR。

SSRDR 为只读寄存器, 所以 CPU 不能写此寄存器。

位:	7	6	5	4	3	2	1	0
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
7 ~ 0		全 0	R	串行接收数据

表 15.4 SSCRL 的 DATS 位的设定和 SSRDR 的对应表

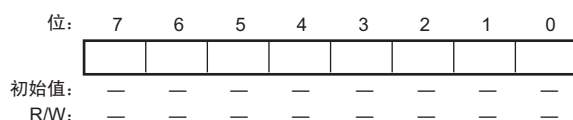
	DATS[1:0] 的设定			
	00	01	10	11 (设定无效)
SSRDR0	有效	有效	有效	无效
SSRDR1	无效	有效	有效	无效
SSRDR2	无效	无效	有效	无效
SSRDR3	无效	无效	有效	无效

15.3.9 SS 移位寄存器 (SSTRSR)

SSTRSR 是发送和接收串行数据的移位寄存器。

在将发送数据从 SSTDR 传送到 SSTRSR 时, 如果 SSMR 的 MLS=0, 就将 SSTDR 的 bit0 (LSB 先) 传送到 SSTRSR 的 bit0; 如果 MLS=1, 就将 SSTDR 的 bit7 (MSB 先) 传送到 SSTRSR 的 bit0。然后从 SSTRSR 的 LSB (bit0) 开始按顺序将数据传送到 SSO 引脚, 进行串行数据的传送。

在接收时, 按照从 LSB (bit0) 开始接收的顺序, 将 SSI 引脚输入的串行数据设定到 SSTRSR。当 1 字节的数据接收结束时, 就自动将 SSTRSR 的数据传送到 SSRDR。CPU 不能直接存取 SSTRSR。



15.4 运行说明

15.4.1 传送时钟

能从 7 种内部时钟和外部时钟中选择传送时钟。在使用本模块时, 需要通过 PFC 预先将 SSCK 引脚设定为有效。当 SSCRH 的 MSS=1 时, 选择内部时钟并且 SSCK 引脚为输出引脚, 在开始传送时, 从 SSCK 引脚输出 SSMR 的 CKS2 ~ CKS0 所设传送率的时钟; 当 MSS=0 时, 选择外部时钟并且 SSCK 引脚为输入引脚。

15.4.2 时钟的相位、极性和数据的关系

如图 15.2 所示, 当 SSCRL 的 SSUMS=0 时, 时钟的相位、极性和传送数据的关系因 SSMR 的 CPOS 位和 CPHS 位的组合而不同。当 SSUMS=1 时, CPOS 位的设定有效, 而 CPHS 位的设定无效。

能通过 SSMR 的 MLS 的设定, 选择是以 MSB 先还是以 LSB 先进行传送; 当 MLS=0 时, 按照从 LSB 到 MSB 的顺序进行传送; 当 MLS=1 时, 按照从 MSB 到 LSB 的顺序进行传送。

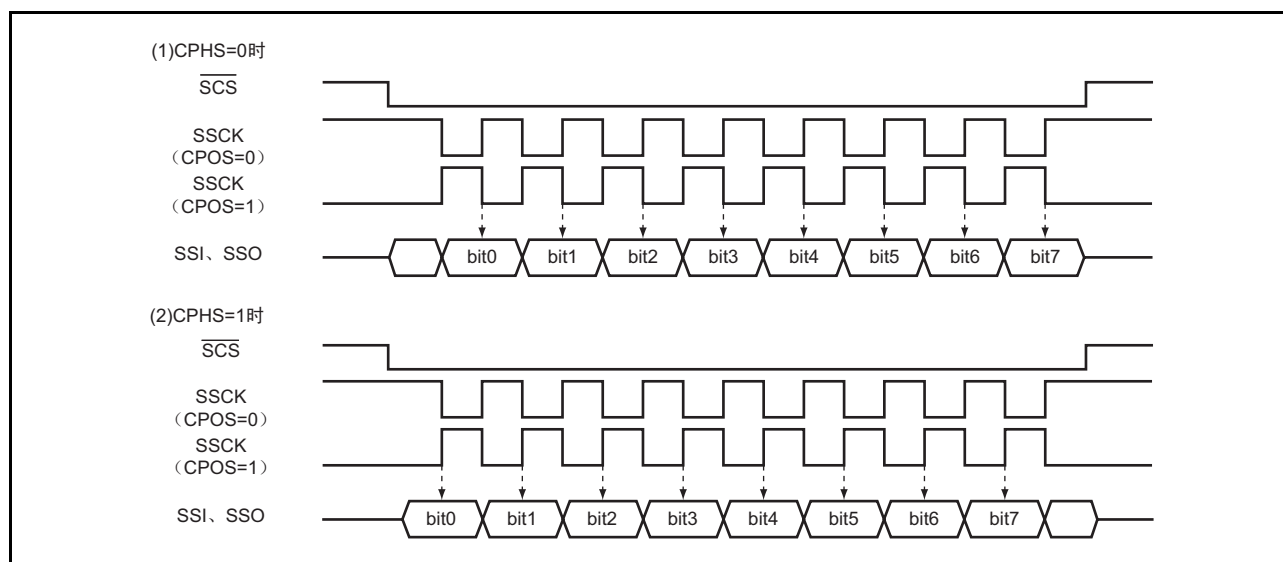


图 15.2 时钟的相位、极性和数据的关系

15.4.3 数据输入 / 输出引脚和移位寄存器的关系

如图 15.3 所示, 数据输入 / 输出引脚和 SS 移位寄存器 (SSTRSR) 的连接关系因 SSCRH 的 MSS、BIDE 以及 SSCRL 的 SSUMS 的组合而不同。

当在 BIDE=0 且 MSS=1 (标准、主控模式) 的状态下运行时, SSU 从 SSO 引脚发送串行数据, 从 SSI 引脚接收串行数据 (图 15.3(1)); 当在 BIDE=0 且 MSS=0 (标准、从属模式) 的状态下运行时, SSU 从 SSI 引脚发送串行数据, 从 SSO 引脚接收串行数据 (图 15.3(2))。

在 BIDE=1 (双向模式) 的状态下, 与主控模式和从属模式无关, SSU 从 SSO 引脚发送或者接收串行数据 (图 15.3(3) 和图 15.3(4))。但是不能通过同时将 TE 和 RE 置 1 进行同时发送和接收, 只能选择 TE 或者 RE。

当在 SSUMS=1 的状态下运行时, SSU 从 SSO 引脚发送串行数据, 从 SSI 引脚接收串行数据。当 MSS=1 时, SSU 从 SSCK 引脚输出内部时钟; 当 MSS=0 时, SSCK 引脚为输入引脚 (图 15.3(5) 和图 15.3(6))。

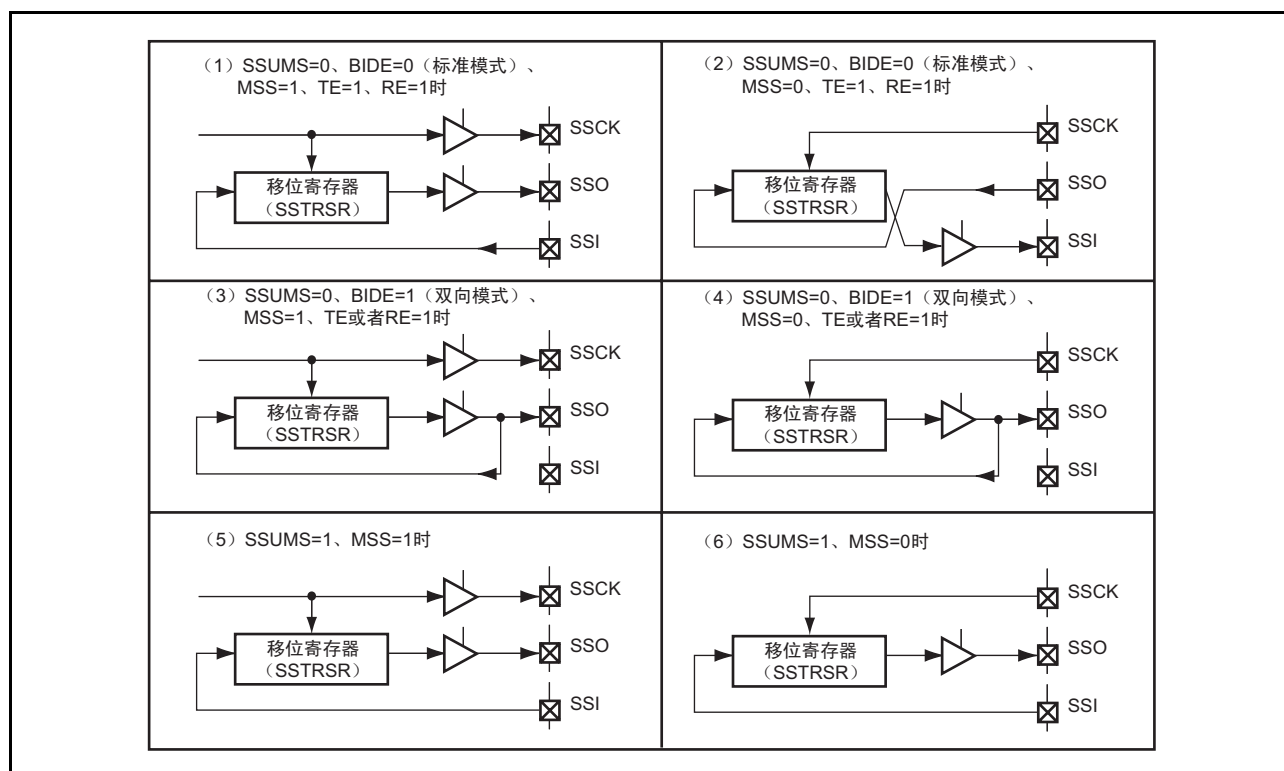


图 15.3 数据输入 / 输出引脚和移位寄存器的关系

15.4.4 各通信模式和引脚功能

SSU 通过各通信模式和寄存器的设定来转换输入 / 输出引脚 (SSI、SSO、SSCK、SCS) 的功能。必须通过端口 I/O 寄存器设定引脚的输入 / 输出转换。各通信模式和输入 / 输出引脚的关系如表 15.5 ~ 表 15.7 所示。

表 15.5 各通信模式和 SSI、SSO 引脚的状态

通信模式	寄存器状态					引脚状态	
	SSUMS	BIDE	MSS	TE	RE	SSI	SSO
SSU 通信模式	0	0	0	0	1	—	输入
				1	0	输出	—
					1	输出	输入
			1	0	1	输入	—
				1	0	—	输出
					1	输入	输出
SSU (双向) 通信模式	0	1	0	0	1	—	输入
				1	0	—	输出
			1	0	1	—	输入
				1	0	—	输出
时钟同步通信模式	1	0	0	0	1	输入	—
				1	0	—	输出
					1	输入	输出
			1	0	1	输入	—
				1	0	—	输出
					1	输入	输出

【符号说明】 —: 不用作 SSU 引脚

表 15.6 各通信模式和 SSCK 引脚的状态

通信模式	寄存器状态		引脚状态
	SSUMS	MSS	SSCK
SSU 通信模式	0	0	输入
		1	输出
时钟同步通信模式	1	0	输入
		1	输出

【符号说明】 —: 不用作 SSU 引脚

表 15.7 各通信模式 $\overline{\text{SCS}}$ 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

15.4.5 SSU 模式

在 SSU 模式中, 使用时钟线 (SSCK)、数据输入线 (SSI 或者 SSO)、数据输出线 (SSI 或者 SSO) 和片选 ($\overline{\text{SCS}}$) 4 条总线进行数据通信。

此模式也对应 1 个引脚用作数据输入线和数据输出线的双向模式。

(1) SSU 模式的初始设定

SSU 模式的初始设定例子如图 15.4 所示。在发送或者接收数据前, 必须在将 SSER 的 TE 和 RE 清 0 后进行初始设定。

【注】 必须在将 TE 和 RE 清 0 后进行运行模式和通信格式的更改。必须注意: 如果将 TE 清 0, TDRE 就被置 1。但是, 即使将 RE 清 0, 也保持 RDRF 和 ORER 的各标志以及 SSRDR 的内容。

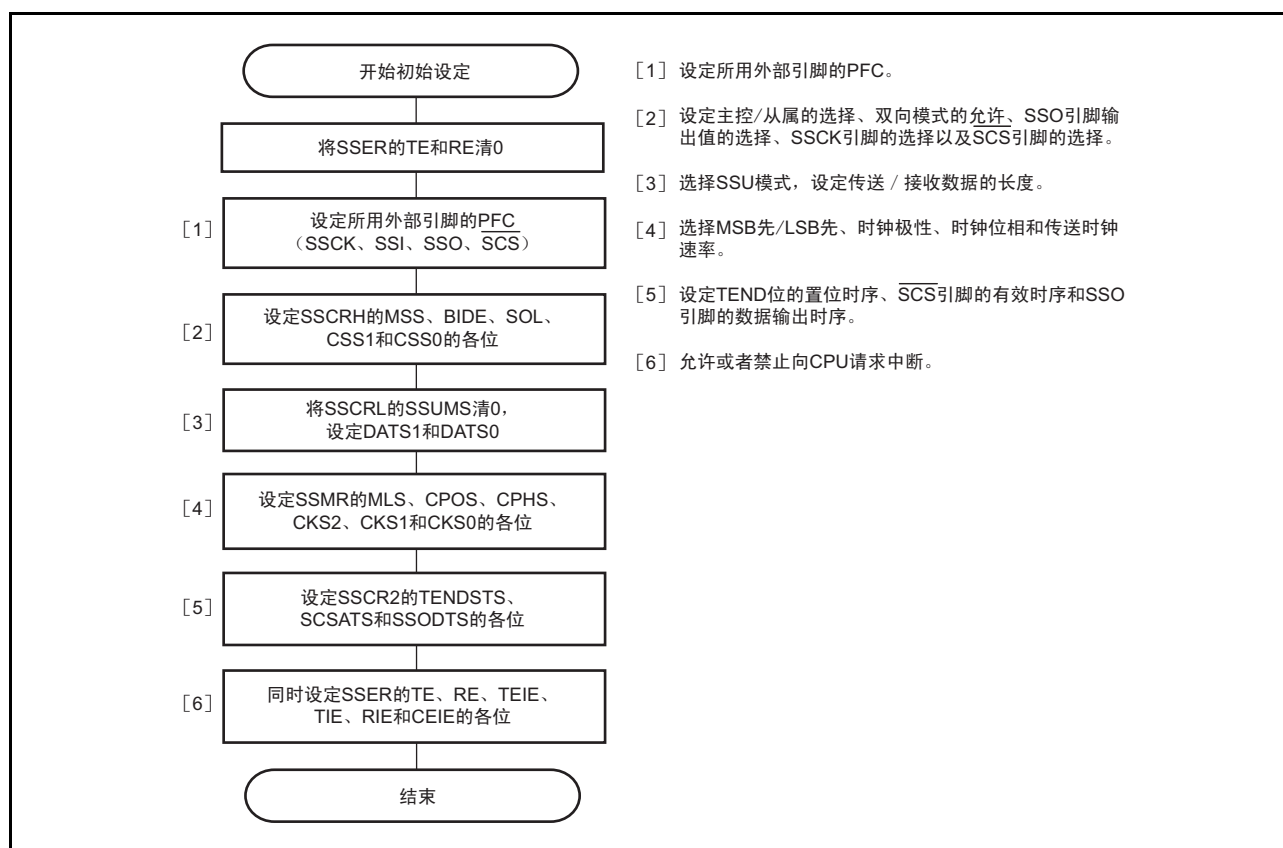


图 15.4 SSU 模式的初始设定例子

(2) 数据的发送

发送时的运行例子和数据发送的流程图例子分别如图 15.5 和图 15.6 所示。

在发送数据时, SSU 的运行如下:

当设定 SSU 为主控器件时, 输出传送时钟和数据; 当设定 SSU 为从属器件时, 如果给 $\overline{\text{SCS}}$ 引脚输入 Low 电平并且从 SSCK 引脚输入传送时钟, SSU 就和此传送时钟同步输出数据。

在 SSER 的 TE 被置 1 后, 如果将发送数据写到 SSTDR, SSU 就将 SSSR 的 TDRE 自动清 0 并将数据从 SSTDR 传送到 SSTRSR, 然后将 TDRE 置 1 开始发送。此时, 如果 SSER 的 TIE 已被置 1, 就产生 TXI 中断请求。

如果在 TDRE=0 的状态下 1 帧数据传送结束, 就将数据从 SSTDR 传送到 SSTRSR, 开始下一帧的发送。

如果在 TDRE=1 的状态下送出第 8 位，SSSR 的 TEND 就被置 1 并保持此状态。此时，如果 SSER 的 TEIE 已被置 1，就产生 TEI 中断。在发送结束后，当 SSMR 的 CPOS=0 时，SSCK 引脚固定为 High 电平；当 CPOS=1 时，SSCK 引脚固定为 Low 电平。

不能在 SSSR 的 ORER 被置 1 的状态下发送数据，必须在发送前确认 ORER 是否已被清 0。

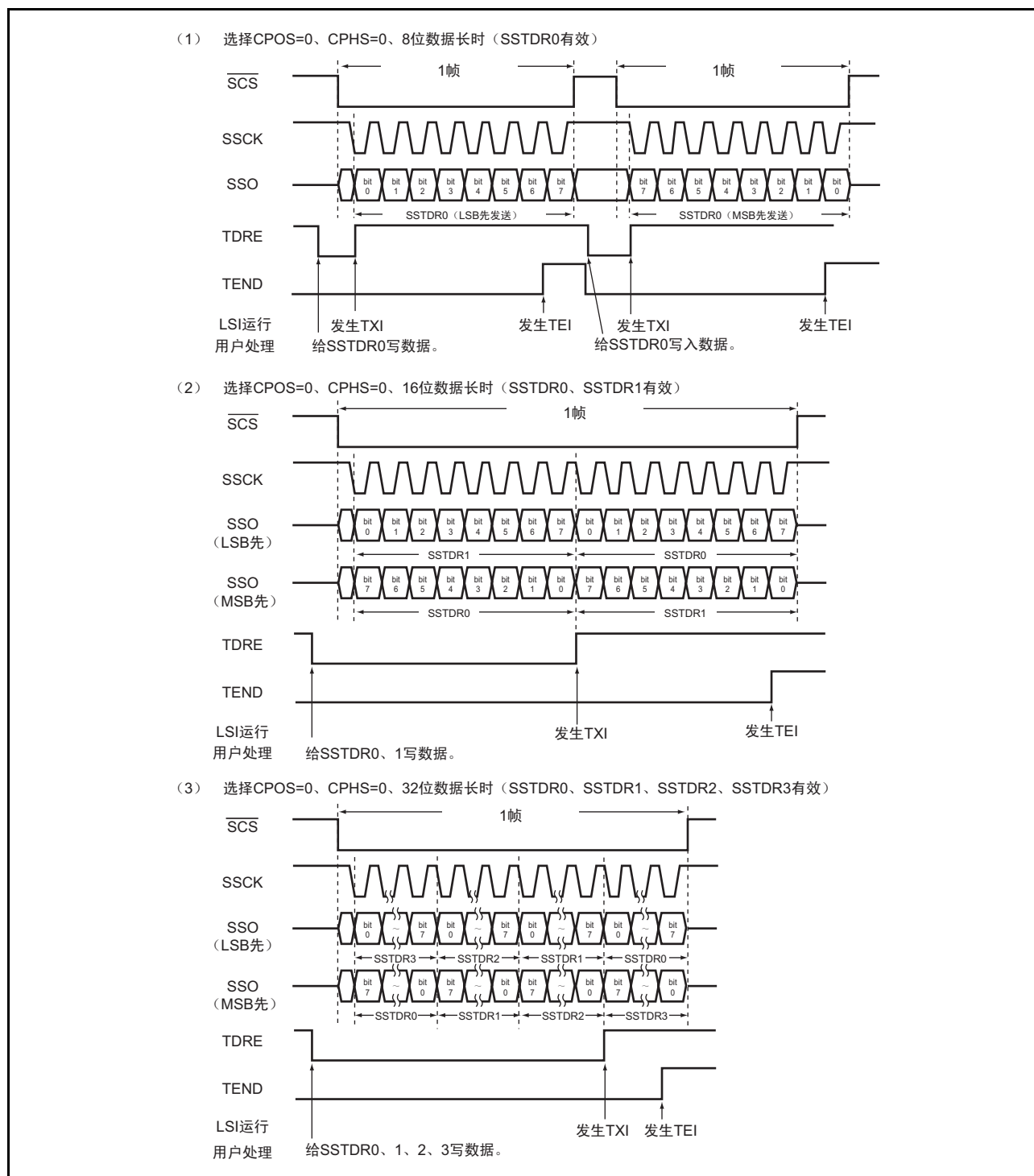


图 15.5 发送时的运行例子 (SSU 模式)

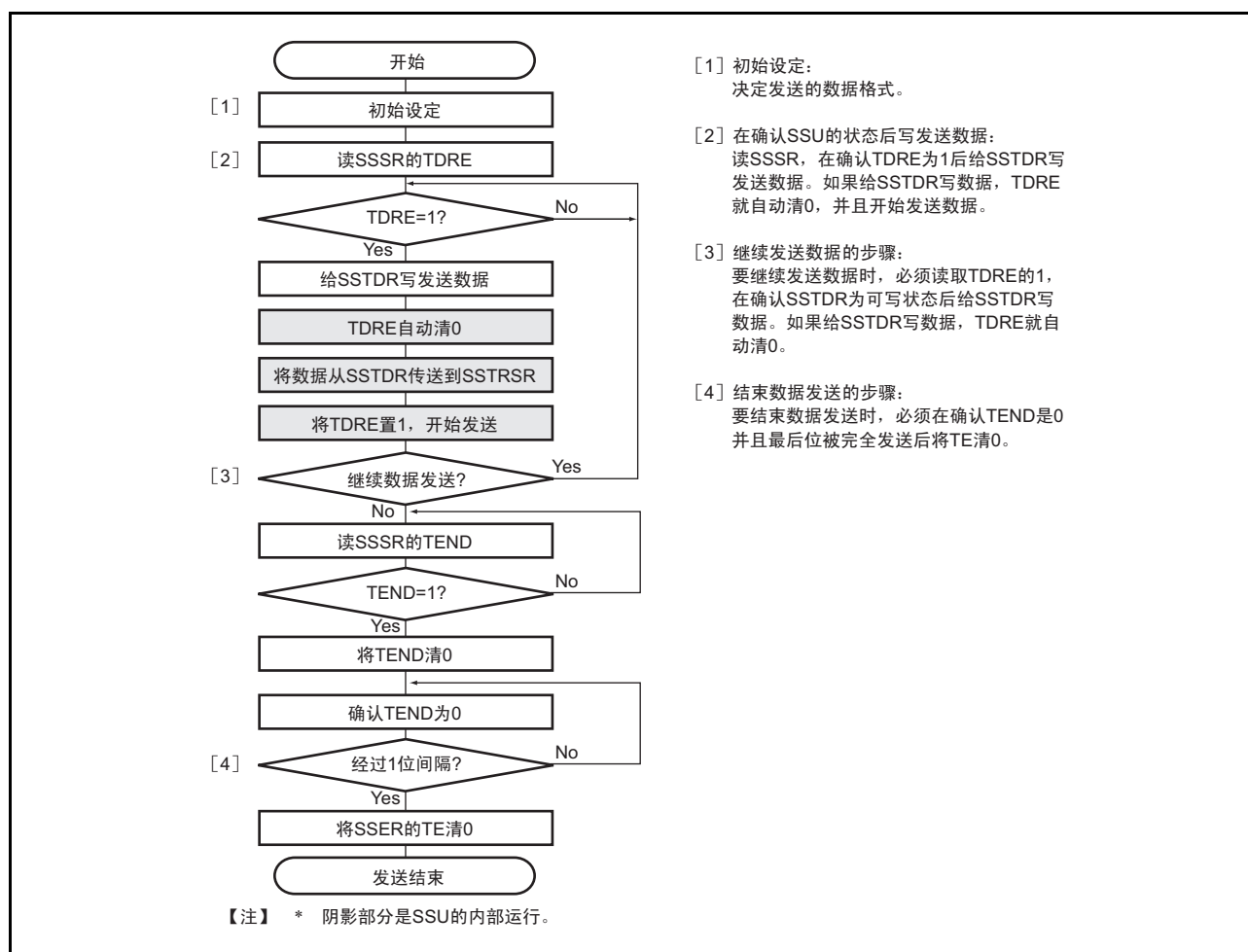


图 15.6 数据发送的流程图例子 (SSU 模式)

(3) 数据的接收

接收时的运行例子和数据接收的流程图例子分别如图 15.7 和图 15.8 所示。

在接收数据时, SSU 的运行如下:

当将 SSER 的 RE 置 1 并读 SSRDR 时, SSU 开始接收。

当设定 SSU 为主控器件时, 输出传送时钟并输入接收数据; 当设定 SSU 为从属器件时, 如果给 $\overline{\text{SCS}}$ 引脚输入 Low 电平并从 SSCK 引脚输入传送时钟, SSU 就与此传送时钟同步输入接收数据。

在接收 1 帧的数据后, 将 SSSR 的 RDRF 置 1, 并将接收数据保存到 SSRDR。此时, 如果 SSER 的 RIE 已被置 1, 就产生 RXI 中断请求。当读 SSRDR 时, RDRF 自动清 0。

如果在 RDRF=1 的状态下第 8 个时钟上升, SSSR 的 ORER 就被置 1, 并且因发生溢出错误 (OEI) 而停止接收。因为在 ORER=1 的状态下不能接收, 所以在重新开始接收时, 必须将 ORER 清 0。

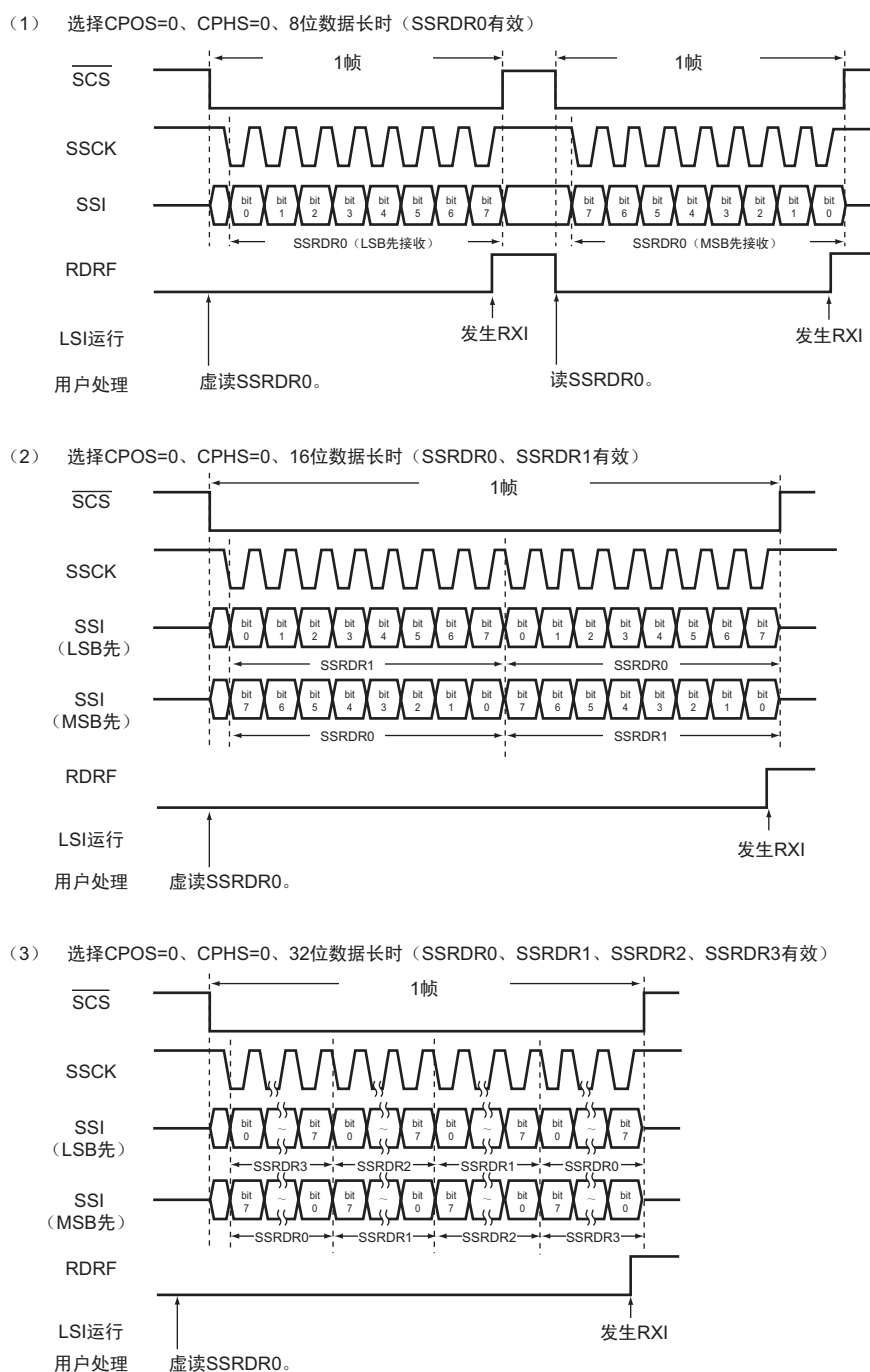


图 15.7 接收时的运行例子 (SSU 模式)

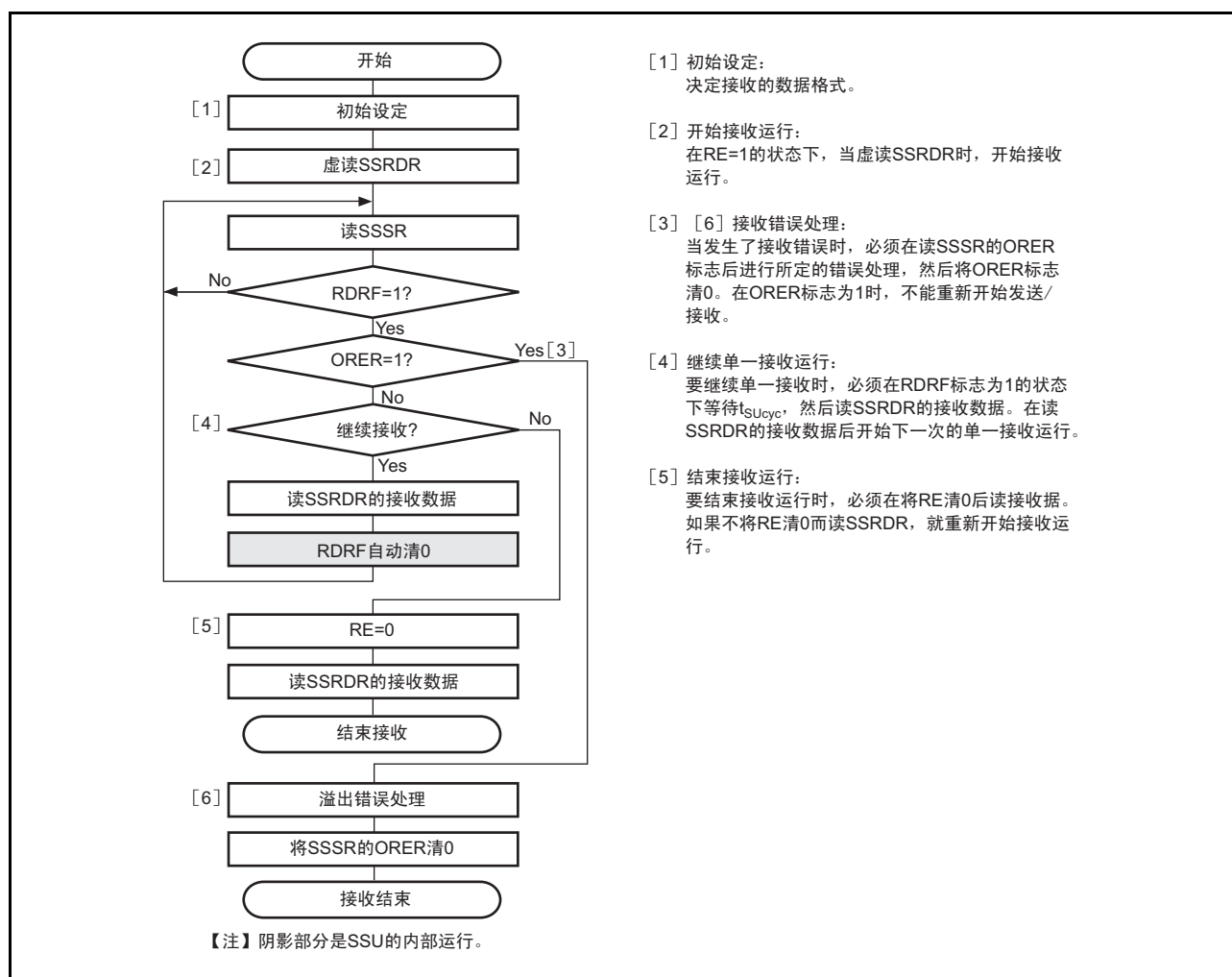


图 15.8 数据接收的流程图例子 (SSU 模式)

(4) 数据的发送和接收

同时发送和接收数据的流程图例子如图 15.9 所示, 数据的发送和接收是发送数据和接收数据的复合运行。在 $TE=RE=1$ 的状态下, 通过将发送数据写到 SSTDR, 开始发送和接收数据。

在从发送模式 ($TE=1$) 或者接收模式 ($RE=1$) 转换为发送和接收模式 ($TE=RE=1$) 时, 必须在将 TE 和 RE 清 0 后进行转换。另外, 必须在确认 $TEND$ 、 $RDRF$ 和 $ORER$ 已被清 0 后将 TE 和 RE 置 1。

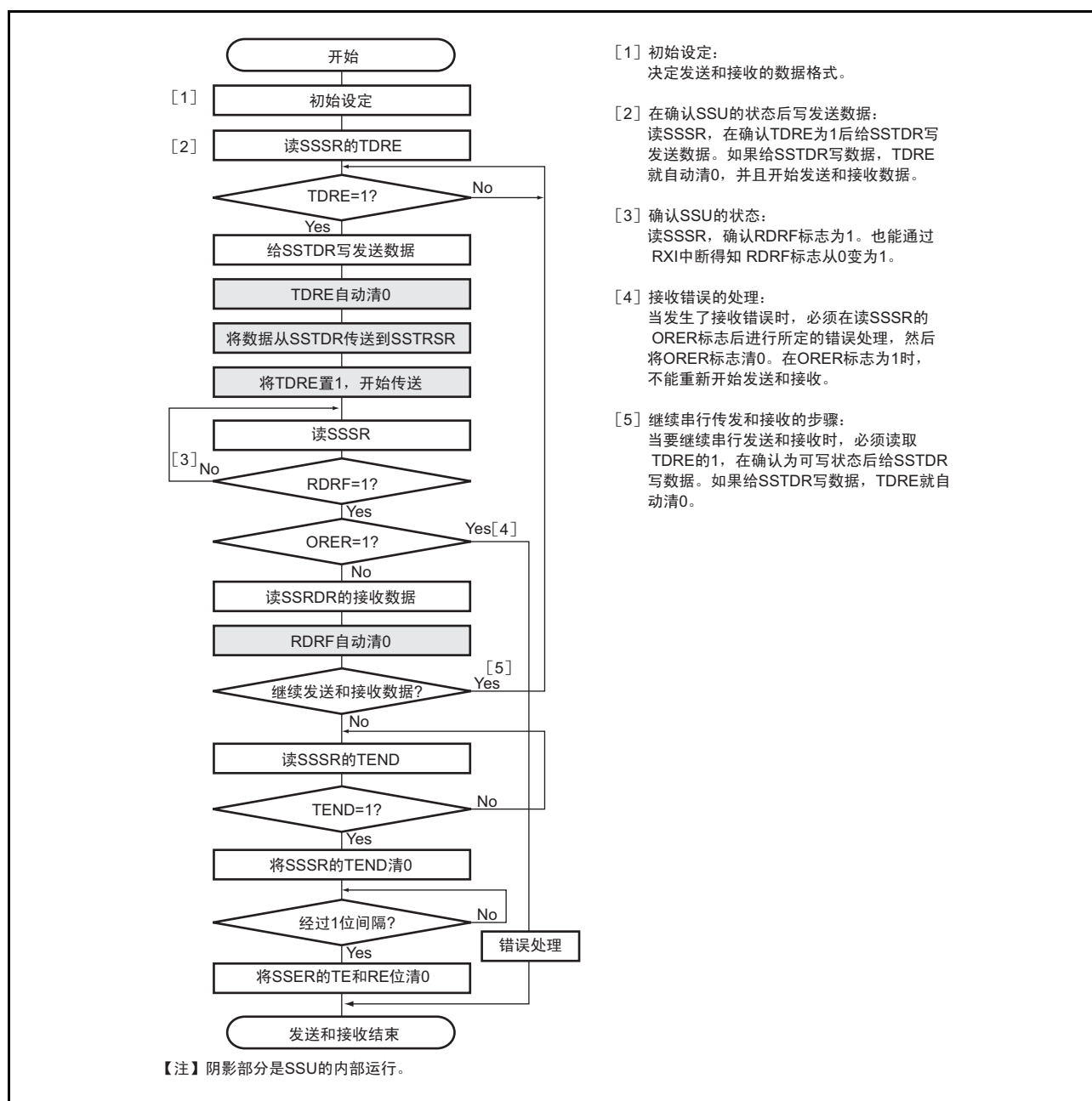


图 15.9 同时发送和接收数据的流程图例子 (SSU 模式)

15.4.6 SCS 引脚控制和冲突错误

当设定 SSCRH 的 CSS1、CSS0=10 和 SSCRL 的 SSUMS=0 并将 SSCRH 的 MSS 置 1 时, 就在串行传送开始前和传送结束后 SCS 引脚变为输入引脚 (Hi-Z), 并检测冲突错误。在此期间, 如果从 SCS 引脚输入 Low 电平, 就发生冲突错误, SSSR 的 CE 被置位, 并且 MSS 被清除。

【注】在冲突错误被置位的状态下, 不能进行以后的发送或者接收。在开始发送或者接收前, 必须将 CE 清 0。

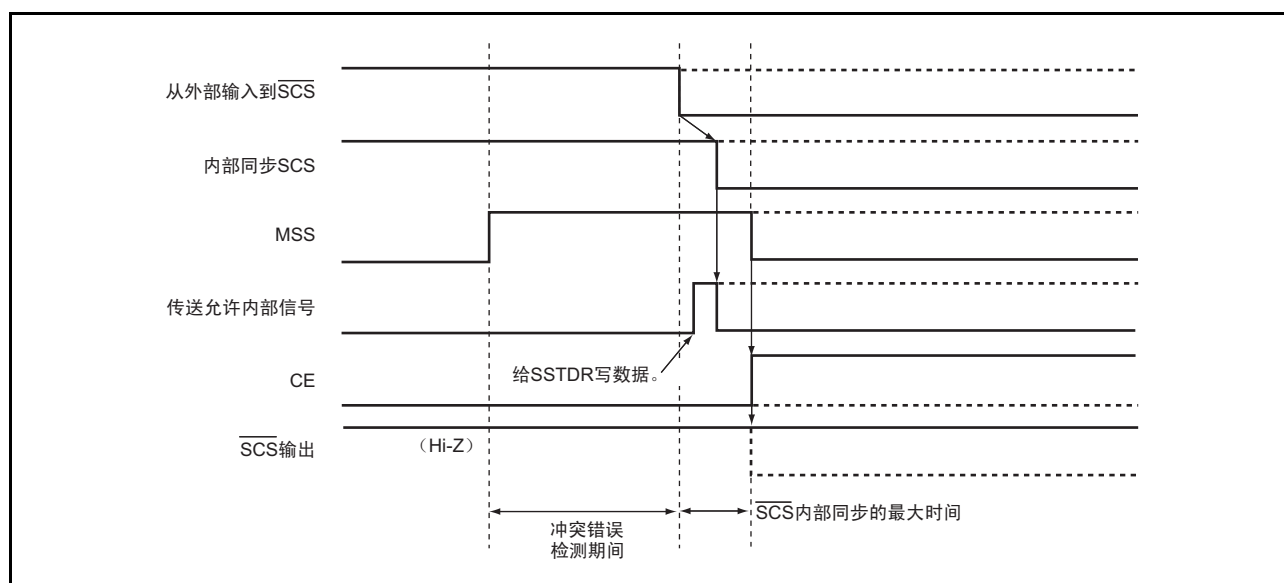


图 15.10 冲突错误的检测时序 (传送前)

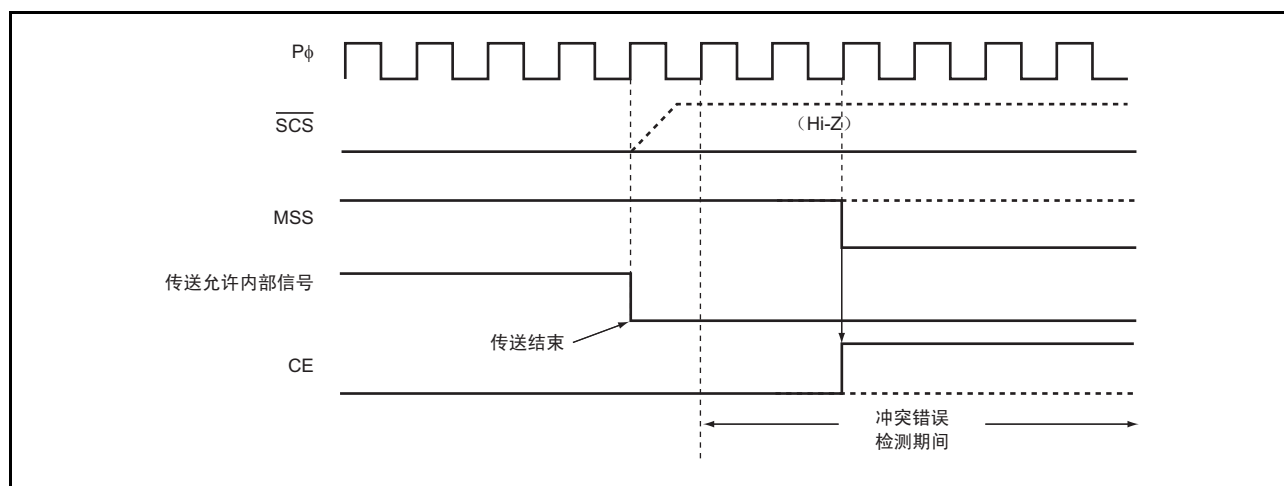


图 15.11 冲突错误的检测时序 (传送结束后)

15.4.7 时钟同步通信模式

在时钟同步通信模式中, 使用时钟线 (SSCK)、数据输入线 (SSI) 和数据输出线 (SSO) 3 条总线进行数据通信。

(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 15.12 所示, 在发送或者接收数据前, 必须将 SSER 的 TE 和 RE 清 0, 然后进行初始设定。

【注】 必须在将 TE 和 RE 清 0 后进行运行模式和通信格式的更改。必须注意: 如果将 TE 清 0, TDRE 就被置 1。但是, 即使将 RE 清 0, 也保持 RDRF 和 ORER 的各标志以及 SSRDR 的内容。

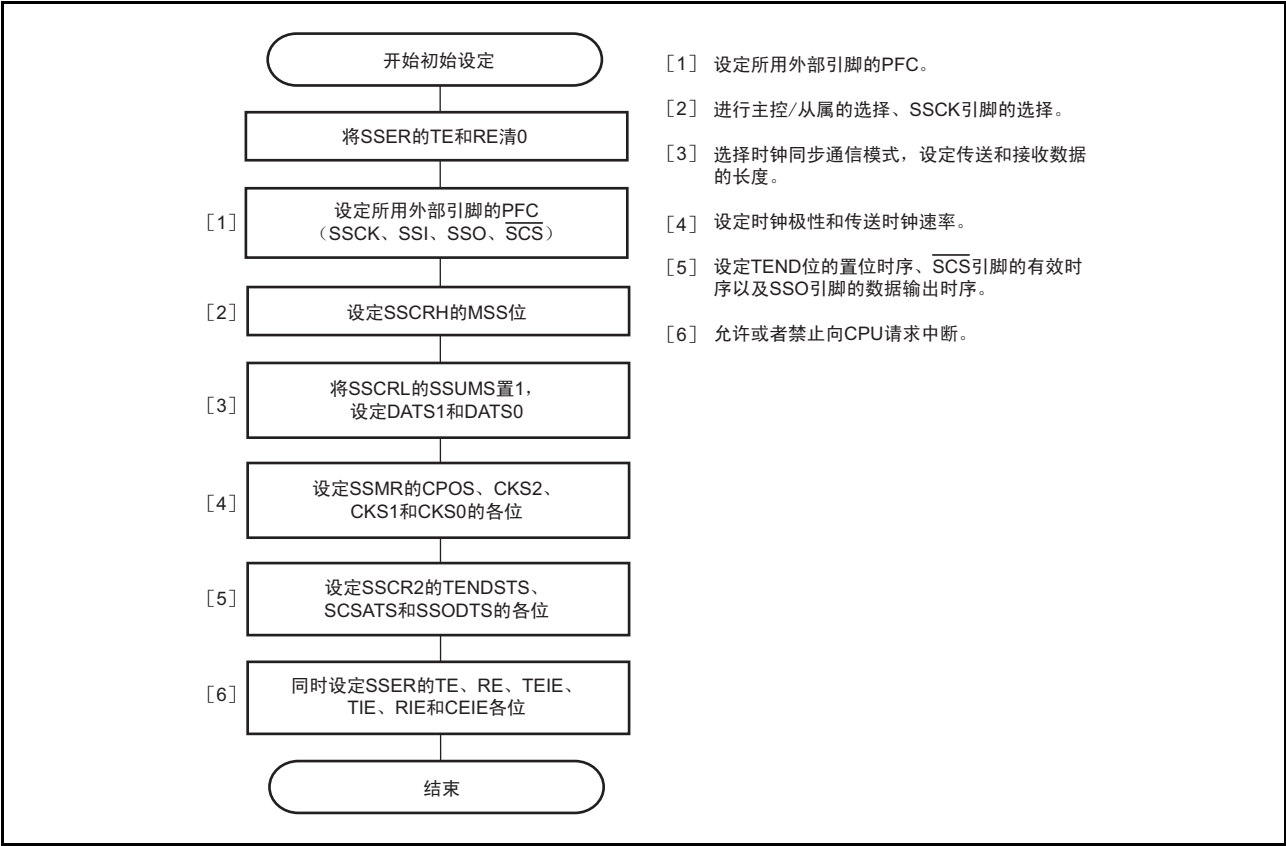


图 15.12 时钟同步通信模式的初始设定例子

(2) 数据的发送

发送时的运行例子和发送数据的流程图例子分别如图 15.13 和图 15.14 所示。

在时钟同步通信模式中发送数据时，SSU 的运行如下：

当设定 SSU 为主控器件时，输出传送时钟和数据；当设定 SSU 为从属器件时，如果从 SSCK 引脚输入传送时钟，SSU 就和此传送时钟同步输出数据。

在 SSER 的 TE 被置 1 后，如果将发送数据写到 SSTDR，SSU 就将 SSSR 的 TDRE 自动清 0 并将数据从 SSTDR 传送到 SSTRSR，然后将 TDRE 置 1 开始发送。此时，如果 SSER 的 TIE 已被置 1，就产生 TXI 中断请求。

如果在 TDRE=0 的状态下 1 帧数据传送结束，就将数据从 SSTDR 传送到 SSTRSR，开始下一帧的发送。如果在 TDRE=1 的状态下送出第 8 位，SSSR 的 TEND 就被置 1 并保持此状态。此时，如果 SSER 的 TEIE 已被置 1，就产生 TEI 中断请求。

不能在 SSSR 的 ORER 被置 1 的状态下发送数据，必须在发送前确认 ORER 是否已被清 0。

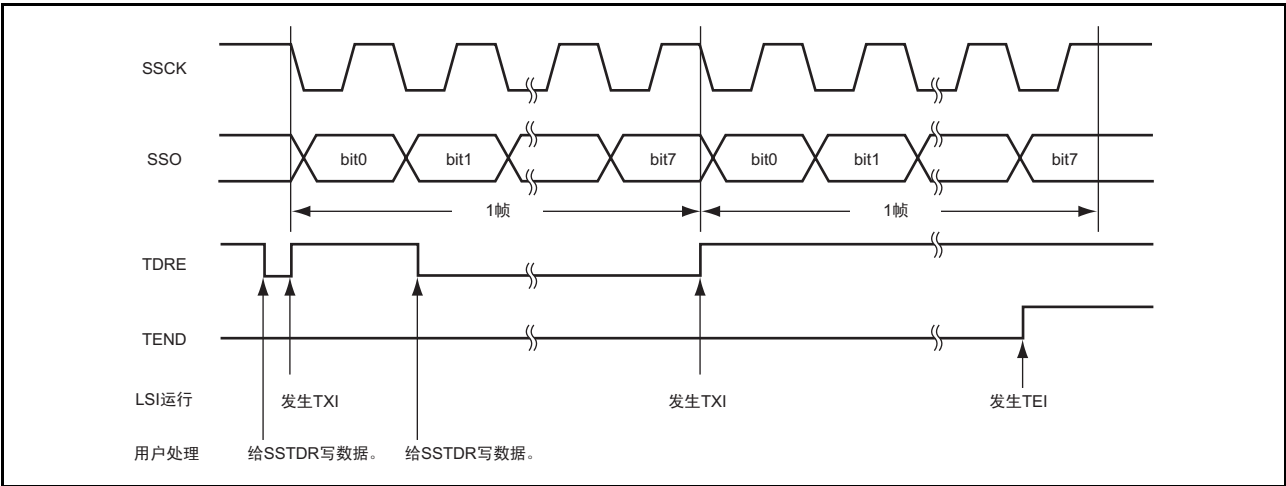


图 15.13 发送时的运行例子（时钟同步通信模式）

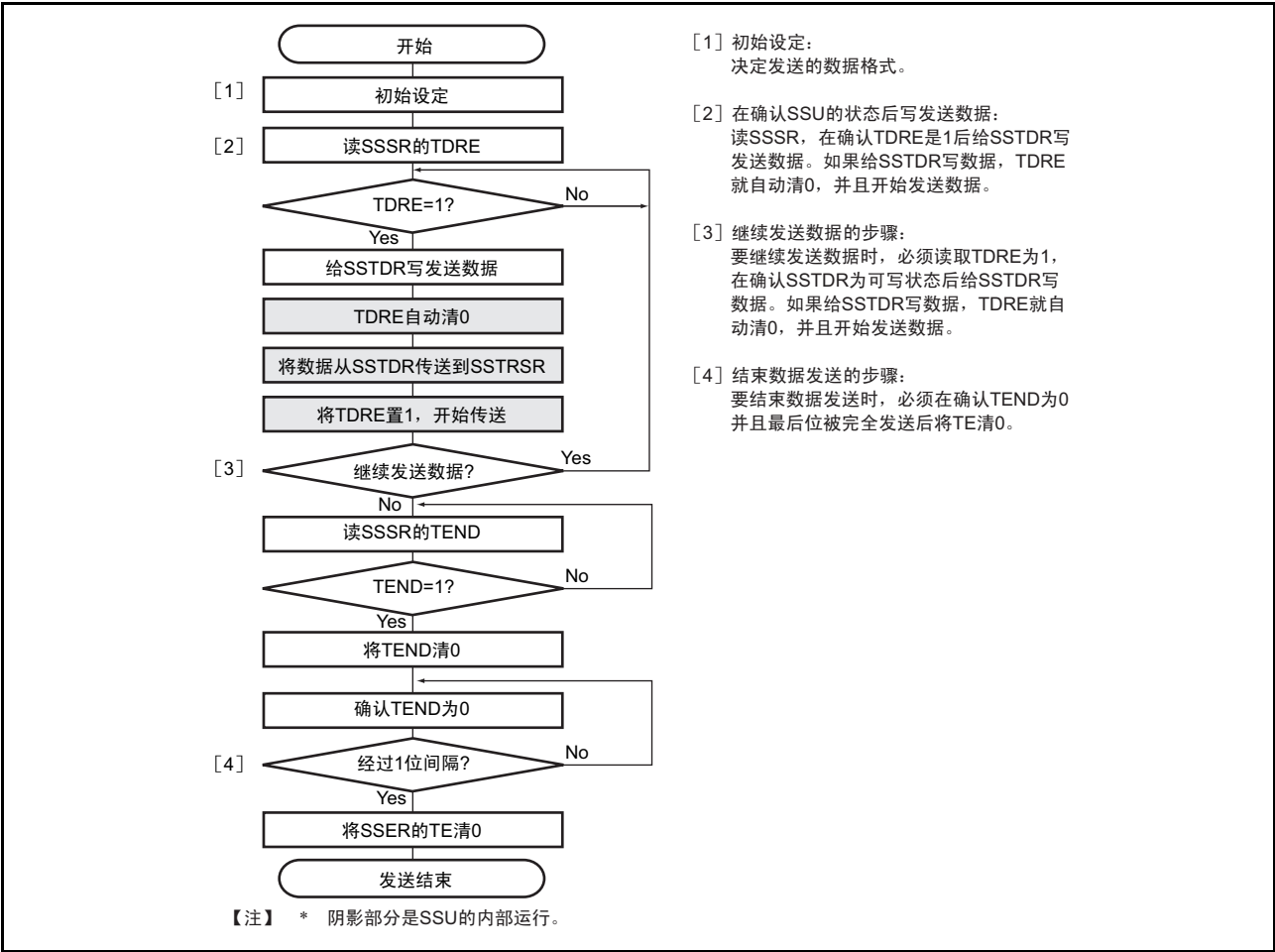


图 15.14 发送数据的流程图例子（时钟同步通信模式）

(3) 数据的接收

接收时的运行例子和接收数据的流程图例子分别如图 15.15 和图 15.16 所示。
在接收数据时, SSU 的运行如下:

当将 SSER 的 RE 置 1 时，SSU 开始接收数据。

当设定 SSU 为主控器件时，输出传送时钟并输入接收数据；当设定 SSU 为从属器件时，如果从 SSCK 引脚输入传送时钟，SSU 就和此传送时钟同步输入接收数据。

在接收 1 帧的数据后，将 SSSR 的 RDRF 置 1，并将接收数据保存到 SSRDR。此时，如果 SSER 的 RIE 已被置 1，就产生 RXI 中断请求。当读 SSRDR 时，RDRF 自动清 0。

如果在 RDRF=1 的状态下第 8 个时钟上升，SSSR 的 ORER 就被置 1，并且因发生溢出错误 (OEI) 而停止接收。因为在 ORER=1 的状态下不能接收，所以在重新开始接收时，必须将 ORER 清 0。

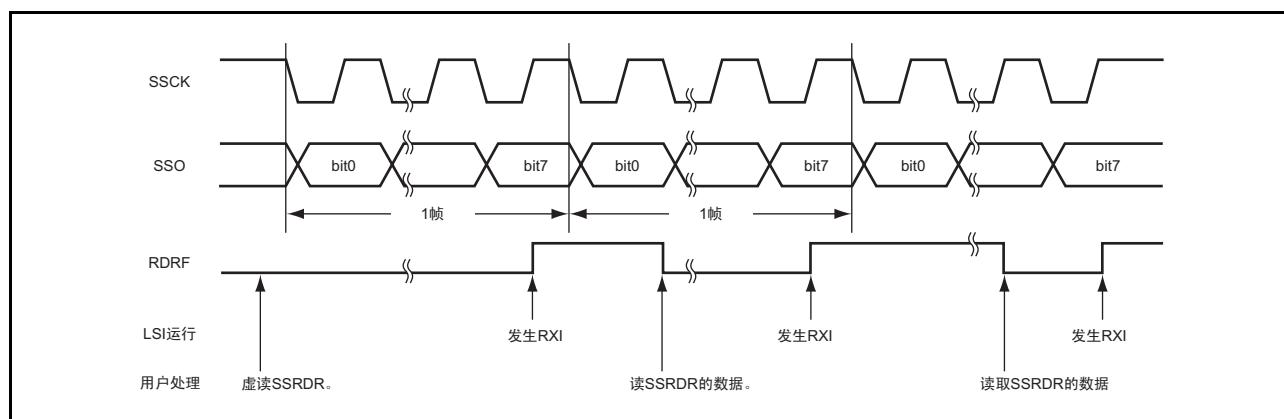


图 15.15 接收时的运行例子 (时钟同步通信模式)

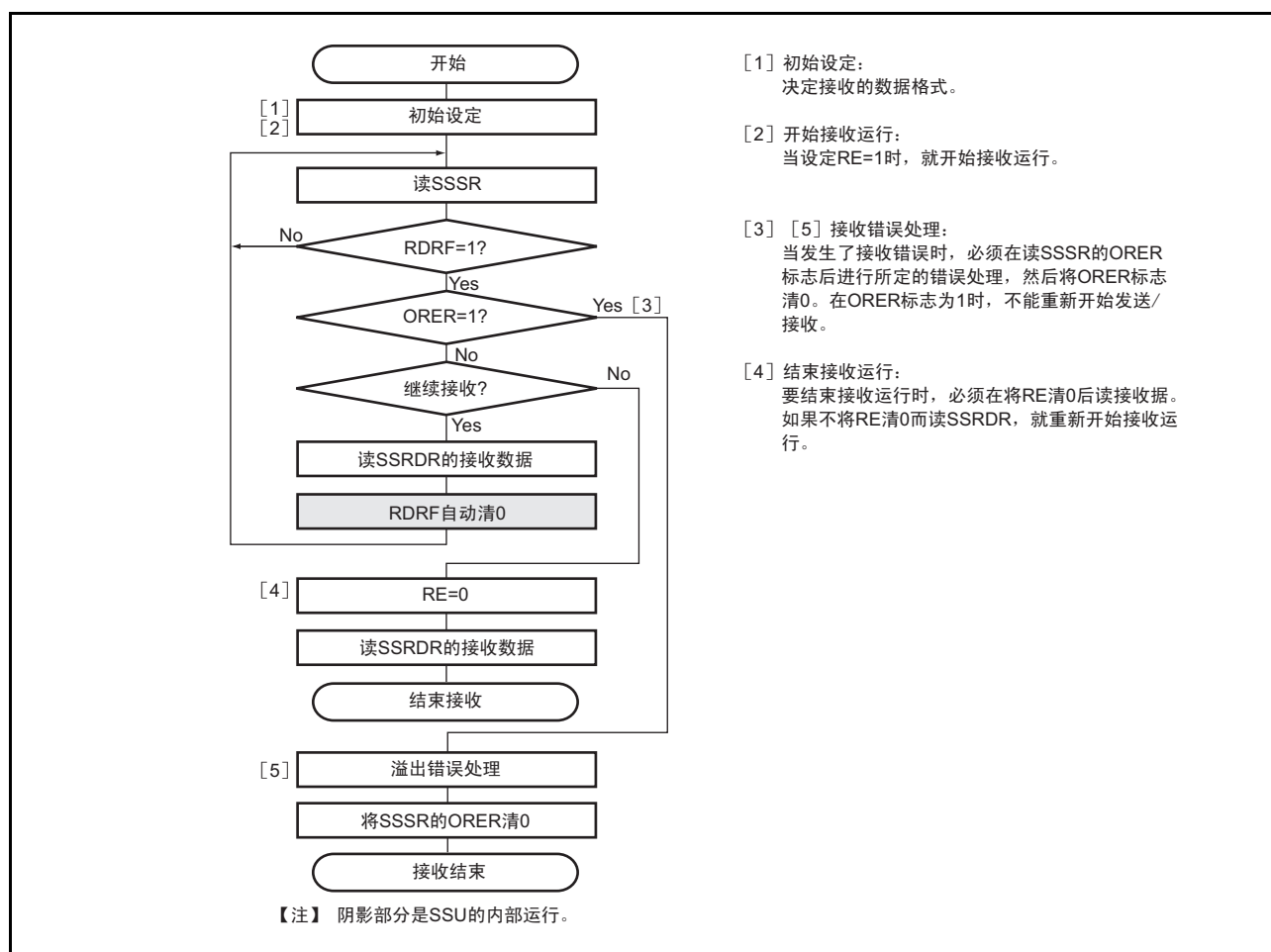


图 15.16 接收数据的流程图例子 (时钟同步通信模式)

(4) 数据的发送和接收

同时发送和接收数据的流程图例子如图 15.17 所示，数据的发送和接收是发送数据和接收数据的复合运行。在 TE=RE=1 的状态下，通过将发送数据写到 SSTDR，开始发送和接收数据。

在从发送模式 (TE=1) 或者接收模式 (RE=1) 转换为发送和接收模式 (TE=RE=1) 时，必须在将 TE 和 RE 清 0 后进行转换。另外，必须在确认 TEND、RDRF 和 ORER 已被清 0 后将 TE 和 RE 置 1。

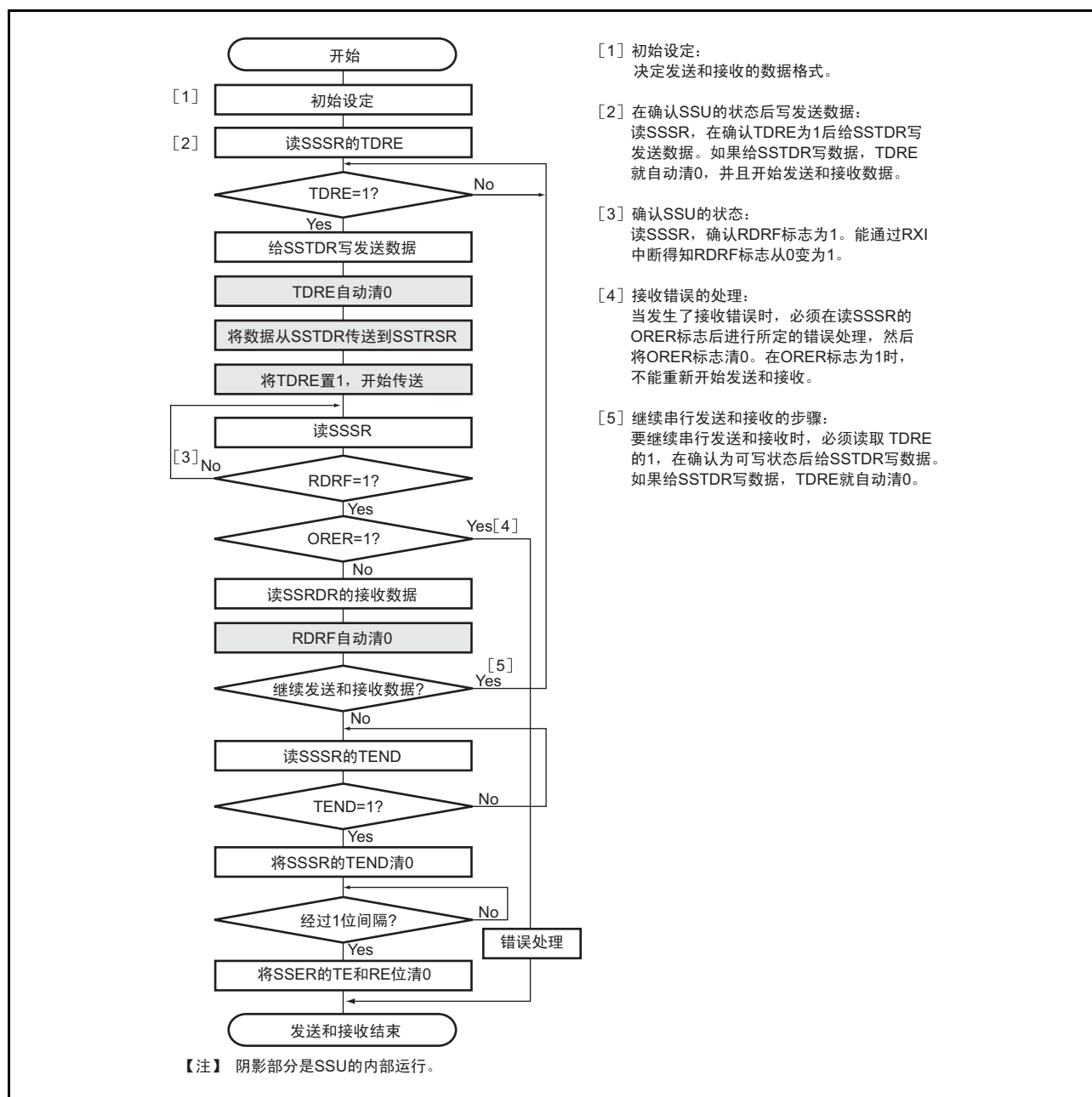


图 15.17 同时发送和接收数据的流程图例子 (时钟同步通信模式)

15.5 SSU 的中断源和 DTC

SSU 的中断请求有溢出错误、冲突错误、接收数据满、发送数据空和发送结束中断，还能通过接收数据满、发送数据空的中断请求启动 DTC 进行数据传送。

由于溢出错误、冲突错误的中断请求被分配到 SSERI 的向量地址，发送数据空、发送结束的中断请求被分配到 SSTXI 的向量地址，所以需要根据标志判断中断源，中断源如表 15.8 所示。

当表 15.8 的中断条件成立时，就产生中断请求。必须通过 CPU 或者 DTC 的数据传送来清除中断源。

表 15.8 SSU 中断源

名称	中断源	略称	中断条件	DTC 的启动
SSERI	溢出错误	SSOEI	$(RIE=1) \cdot (ORER=1)$	—
	冲突错误	SSCEI	$(CEIE=1) \cdot (CE=1)$	—
SSRXI	接收数据满	SSRXI	$(RIE=1) \cdot (RDRF=1)$	○
SSTXI	发送数据空	SSTXI	$(TIE=1) \cdot (TDRE=1)$	○
	发送结束	SSTEI	$(TEIE=1) \cdot (TEND=1)$	—

15.6 使用时的注意事项

15.6.1 模块待机模式的设定

SSU 能通过待机控制寄存器禁止或者允许本模块的运行，初始值为 SSU 停止运行。能通过解除模块待机模式允许寄存器的存取，详细内容请参照“第 24 章 低功耗模式”。

15.6.2 SSTDR 和 SSRDR 寄存器的存取

不能存取由 SSCRL 寄存器的 DATS 位设定为无效的 SSTDR 和 SSRDR 的寄存器，否则有可能无法正常进行以后的发送和接收。

15.6.3 SSU 从属模式中的连续发送和接收时的注意事项

在 SSU 从属模式中进行连续发送和接收时，必须按帧将 $\overline{SCS}$ 引脚设定为无效（高电平）。如果 $\overline{SCS}$ 引脚有效（低电平）状态超过 1 帧，就不能正常地发送和接收。

第 16 章 I²C 总线接口 2 (I²C2)

I²C 总线接口 2 基于菲利普斯公司提倡的 I²C 总线 (Inter IC Bus) 接口方式, 具有子集功能。必须注意: 控制 I²C 总线的一部分寄存器结构和菲利普斯公司不同。

I²C 总线接口 2 的框图和输入 / 输出引脚的外部电路连接例子分别如图 16.1 和图 16.2 所示。

16.1 特点

- 能选择 I²C 总线格式或者时钟同步串行格式
- 能连续发送 / 接收
因为移位寄存器、发送数据寄存器和接收数据寄存器各自独立, 所以能连续发送 / 接收。
- 能设定模块待机模式

I²C 总线格式

- 在主控模式中, 自动生成开始条件和停止条件。
- 在接收时, 能选择应答的输出电平。
- 在发送时, 自动加载应答位。
- 内置了位同步功能
在可控模式中, 按位监视 SCL 状态, 自动取得同步。在尚未做好传送准备时, 将 SCL 置为 Low 电平使之处于待机状态。
- 中断源: 6 种
发送数据空 (包括从属地址匹配时)、发送结束、接收数据满 (包括从属地址匹配时)、仲裁失败、NACK 检测、停止条件的检测。
能通过发送数据空请求或者接收数据满请求, 启动数据传送控制器 (DTC) 进行数据传送。
- 能直接驱动总线
在选择总线驱动功能时, SCL 引脚和 SDA 引脚为 NMOS 漏极开路输出。

时钟同步串行格式

- 中断源: 4 种
发送数据空、发送结束、接收数据满、溢出错误。
能通过发送数据空请求或者接收数据满请求, 启动数据传送控制器 (DTC) 进行数据传送。

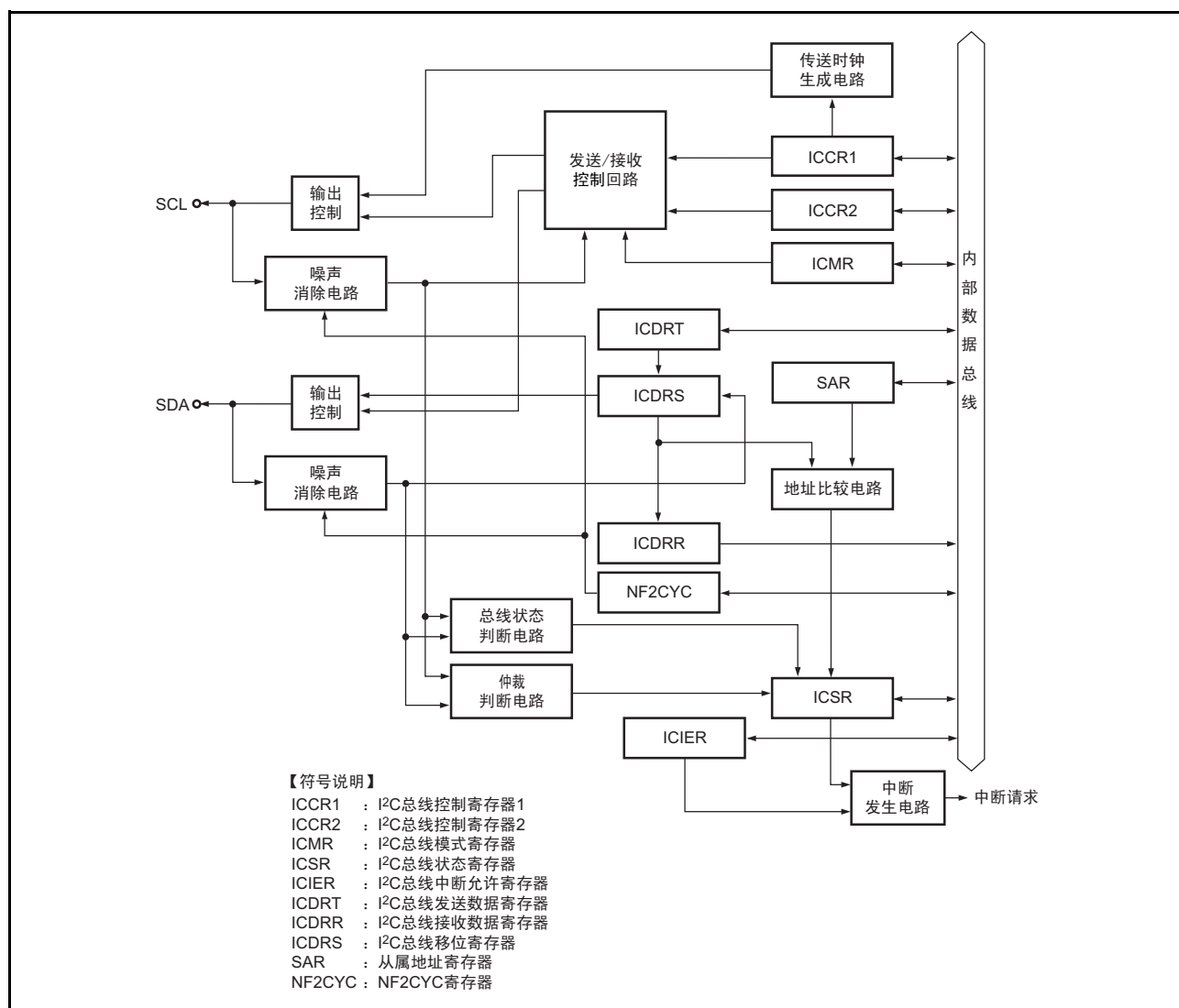
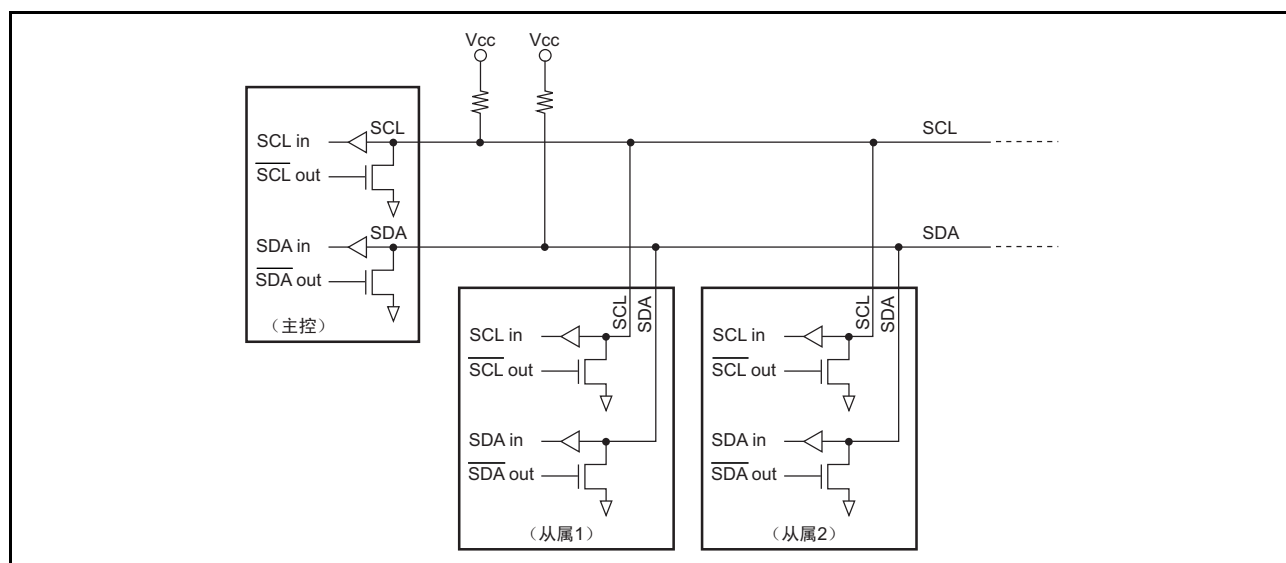
图 16.1 I²C 总线接口 2 的框图

图 16.2 输入 / 输出引脚的外部电路连接例子

16.2 输入 / 输出引脚

I²C 总线接口 2 使用的引脚结构如表 16.1 所示。

表 16.1 引脚结构

名称	引脚名	输入 / 输出	功能
串行时钟引脚	SCL	输入 / 输出	I ² C 串行时钟的输入 / 输出引脚
串行数据引脚	SDA	输入 / 输出	I ² C 串行数据的输入 / 输出引脚

16.3 寄存器说明

I²C 总线接口 2 有以下寄存器。有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 25 章 寄存器一览”。

表 16.2 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
I ² C 总线控制寄存器 1	ICCR1	R/W	H'00	H'FFFFCD80	8
I ² C 总线控制寄存器 2	ICCR2	R/W	H'7D	H'FFFFCD81	8
I ² C 总线模式寄存器	ICMR	R/W	H'38	H'FFFFCD82	8
I ² C 总线中断允许寄存器	ICIER	R/W	H'00	H'FFFFCD83	8
I ² C 总线状态寄存器	ICSR	R/W	H'00	H'FFFFCD84	8
从属地址寄存器	SAR	R/W	H'00	H'FFFFCD85	8
I ² C 总线发送数据寄存器	ICDRT	R/W	H'FF	H'FFFFCD86	8
I ² C 总线接收数据寄存器	ICDRR	R/W	H'FF	H'FFFFCD87	8
NF2CYC 寄存器	NF2CYC	R/W	H'00	H'FFFFCD88	8

16.3.1 I²C 总线控制寄存器 1 (ICCR1)

ICCR1 是 8 位可读写寄存器，进行 I²C 总线接口 2 的运行 / 停止、发送 / 接收的控制以及主控模式 / 从属模式、发送 / 接收模式、主控模式传送时钟频率的选择。

位:	7	6	5	4	3	2	1	0
	ICE	RCVD	MST	TRS	CKS[3:0]			
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7	ICE	0	R/W	I ² C 总线接口 2 的允许 0: 本模块处于功能停止状态 1: 本模块处于可传送状态 (SCL/SDA 处于总线驱动状态)
6	RCVD	0	R/W	接收禁止 在 TRS=0 的状态下，， 允许或者禁止不读 ICDRR 而进行的连续接收。 在主控接收模式中， 如果在 SCL 的第 8 个时钟上升前无法读 ICDRR， 就必须在设定 RCVD=1 后按字节进行接收。 0: 允许连续接收 1: 禁止连续接收

位	位名	初始值	R/W	说明
5 4	MST TRS	0 0	R/W R/W	主控模式和从属模式的选择 发送模式和接收模式的选择 如果在 I²C 总线格式的主控模式中总线竞争失败，就通过硬件对 MST 和 TRS 进行复位，变为从属接收模式。 另外，必须在传送帧期间更改 TRS。在从属接收模式中，如果开始条件后的 7 位与 SAR 设定的从属地址相同并且第 8 位为 1，TRS 就自动置 1。如果在时钟同步串行格式的主控接收模式中发生溢出错误，就将 MST 清 0，变为从属接收模式。 根据 MST 和 TRS 的组合，设定以下运行模式。在选择时钟同步串行格式的情况下，当 MST=1 时，为时钟输出。 00: 从属接收模式 01: 从属发送模式 10: 主控接收模式 11: 主控发送模式
3 ~ 0	CKS[3:0]	0000	R/W	传送时钟的选择 3 ~ 0 在主控模式中，必须根据需要的传送率（参照表 16.3）进行设定。在从属模式中，用于确保发送模式中的数据准备时间。当 CKS3=0 时，数据准备时间为 10 个 t_{pcyc}；当 CKS3=1 时，为 20 个 t_{pcyc}（t_{pcyc} 表示 $P\phi$ 周期）。

表 16.3 传送率

bit3	bit2	bit1	bit0	时钟	传送率					
CKS3	CKS2	CKS1	CKS0		$P\phi=10\text{MHz}$	$P\phi=16\text{MHz}$	$P\phi=20\text{MHz}$	$P\phi=25\text{MHz}$	$P\phi=33\text{MHz}$	$P\phi=40\text{MHz}$
0	0	0	0	$P\phi/28$	357kHz	571kHz	714kHz	893kHz	1.18MHz	1.43MHz
			1	$P\phi/40$	250kHz	400kHz	500kHz	625kHz	825kHz	1.00MHz
		1	0	$P\phi/48$	208kHz	333kHz	417kHz	521kHz	688kHz	833kHz
			1	$P\phi/64$	156kHz	250kHz	313kHz	391kHz	516kHz	625kHz
	1	0	0	$P\phi/80$	125kHz	200kHz	250kHz	313kHz	413kHz	500kHz
			1	$P\phi/100$	100kHz	160kHz	200kHz	250kHz	330kHz	400kHz
		1	0	$P\phi/112$	89.3kHz	143kHz	179kHz	223kHz	295kHz	357kHz
			1	$P\phi/128$	78.1kHz	125kHz	156kHz	195kHz	258kHz	313kHz
1	0	0	0	$P\phi/112$	89.3kHz	143kHz	179kHz	223kHz	295kHz	357kHz
			1	$P\phi/160$	62.5kHz	100kHz	125kHz	156kHz	206kHz	250kHz
		1	0	$P\phi/192$	52.1kHz	83.3kHz	104kHz	130kHz	172kHz	208kHz
			1	$P\phi/256$	39.1kHz	62.5kHz	78.1kHz	97.7kHz	129kHz	156kHz
	1	0	0	$P\phi/320$	31.3kHz	50.0kHz	62.5kHz	78.1kHz	103kHz	125kHz
			1	$P\phi/400$	25.0kHz	40.0kHz	50.0kHz	62.5kHz	82.5kHz	100kHz
		1	0	$P\phi/448$	22.3kHz	35.7kHz	44.6kHz	55.8kHz	73.7kHz	89.3kHz
			1	$P\phi/512$	19.5kHz	31.3kHz	39.1kHz	48.8kHz	64.5kHz	78.1kHz

16.3.2 I²C 总线控制寄存器 2 (ICCR2)

ICCR2 是 8 位可读写寄存器，控制开始 / 停止条件的发行、SDA 引脚的操作、SCL 引脚的监视以及 I²C 控制部的复位。

位:	7	6	5	4	3	2	1	0
	BBSY	SCP	SDAO	SDAOP	SCLO	—	IICRST	—
初始值:	0	1	1	1	1	1	0	1
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R

位	位名	初始值	R/W	说明
7	BBSY	0	R/W	总线忙 有表示 I ² C 总线占有 / 释放状态的标志功能和主控模式开始 / 停止条件的发行功能。在时钟同步串行格式时，此位的读取值总是 0。在 I ² C 总线格式时并且在 SCL=High 电平的状态下，如果 SDA 从 High 电平变为 Low 电平，就认为已发行开始条件，此位被置 1；如果 SDA 从 Low 电平变为 High 电平，就认为已发行停止条件，此位被清 0。在发行开始条件时，给 BBSY 和 SCP 分别写 1 和 0，在重新发行开始条件时也同样；在发行停止条件时，给 BBSY 和 SCP 写 0。
6	SCP	1	R/W	开始 / 停止条件的发行禁止位 SCP 位控制主控模式的开始 / 停止条件的发行。在发行开始条件时，给 BBSY 和 SCP 分别写 1 和 0，在重新发行开始条件时也同样；在发行停止条件时，给 BBSY 和 SCP 写 0。此位的读取值总是 1，即使写 1 也不保存数据。
5	SDAO	1	R/W	SDA 输出值的控制 要更改 SDA 的输出电平时，将 SDAO 和 SDAOP (bit4) 组合使用。 另外，不能在传送中操作此位。 0: 读时，SDA 引脚的输出为 Low 电平。 写时，将 SDA 引脚的输出更改为 Low 电平。 1: 读时，SDA 引脚的输出为 High 电平。 写时，将 SDA 引脚的输出更改为 Hi-Z。 (通过外部上拉电阻输出 High 电平)
4	SDAOP	1	R/W	SDAO 的写保护 通过改写 SDAO 位来更改 SDA 引脚的输出电平。要更改输出电平时，给 SDAO 和 SDAOP 写 0 或者给 SDAO 和 SDAOP 分别写 1 和 0。此位的读取值总是 1。
3	SCLO	1	R	SCLO 监视 SCL 的输出电平。读时，如果 SCLO 为 1，SCL 引脚的输出就为 High 电平；如果 SCLO 为 0，SCL 引脚的输出就为 Low 电平。
2	—	1	R	保留位 读写值总是 1。
1	IICRST	0	R/W	IIC 控制部的复位 IICRST 对 I ² C 寄存器以外的控制部进行复位。在 I ² C 运行中因通信不正常等引起中止时，如果将 IICRST 位置 1，就能对 I ² C 的一部分寄存器和控制部进行复位。
0	—	1	R	保留位 读写值总是 1。

16.3.3 I²C 总线模式寄存器 (ICMR)

ICMR 是 8 位可读写寄存器，选择 MSB 先 /LSB 先和传送位数。

位:	7	6	5	4	3	2	1	0
	MLS	—	—	—	BCWP	BC[2:0]		
初始值:	0	0	1	1	1	0	0	0
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明																		
7	MLS	0	R/W	MSB 先 /LSB 先的选择 0: MSB 先 1: LSB 先 在用于 I ² C 总线格式时，必须将此位置 0。																		
6	—	0	R	保留位 读写值总是 0。																		
5、4	—	全 1	R	保留位 读写值总是 1。																		
3	BCWP	1	R/W	BC 的写保护 控制 BC2 ～ BC0 的写操作。在改写 BC2 ～ BC0 时，必须将此位置 0。 另外，不能在时钟同步串行格式时改写 BC。 0: 写时，设定 BC2 ～ BC0 的值。 1: 读时，读取值总是 1。写时，BC2 ～ BC0 的设定值无效。																		
2 ～ 0	BC[2:0]	000	R/W	位计数器 2 ～ 0 指定下一次要传送的数据位数。如果读这些位，就能得知剩余的传送位数。在 I ² C 总线格式时，数据加上 1 位应答位进行传送。必须在传送帧期间设定这些位。如果要设定非 000 值，就必须在 SCL 为 Low 电平的状态下进行设定。在包含应答位的数据传送结束后，这些位自动返回 000；在检测到停止条件后，这些位自动变为 111。通过上电复位、待机模式或者将 ICCR2 的 IICRST 置 1 来清除这些位。当选择时钟同步串行格式时，不能改写这些位。 <table><tr><td>I²C 总线格式</td><td>时钟同步串行格式</td></tr><tr><td>000: 9 位</td><td>000: 8 位</td></tr><tr><td>001: 2 位</td><td>001: 1 位</td></tr><tr><td>010: 3 位</td><td>010: 2 位</td></tr><tr><td>011: 4 位</td><td>011: 3 位</td></tr><tr><td>100: 5 位</td><td>100: 4 位</td></tr><tr><td>101: 6 位</td><td>101: 5 位</td></tr><tr><td>110: 7 位</td><td>110: 6 位</td></tr><tr><td>111: 8 位</td><td>111: 7 位</td></tr></table>	I ² C 总线格式	时钟同步串行格式	000: 9 位	000: 8 位	001: 2 位	001: 1 位	010: 3 位	010: 2 位	011: 4 位	011: 3 位	100: 5 位	100: 4 位	101: 6 位	101: 5 位	110: 7 位	110: 6 位	111: 8 位	111: 7 位
I ² C 总线格式	时钟同步串行格式																					
000: 9 位	000: 8 位																					
001: 2 位	001: 1 位																					
010: 3 位	010: 2 位																					
011: 4 位	011: 3 位																					
100: 5 位	100: 4 位																					
101: 6 位	101: 5 位																					
110: 7 位	110: 6 位																					
111: 8 位	111: 7 位																					

16.3.4 I²C 总线中断允许寄存器 (ICIER)

ICIER 是 8 位可读写寄存器，允许各种中断源、选择应答的有效 / 无效，设定发送应答以及确认接收应答。

位:	7	6	5	4	3	2	1	0
	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

位	位名	初始值	R/W	说明
7	TIE	0	R/W	发送中断的允许 在 ICSR 的 TDRE 被置位时，允许或者禁止发送数据空的中断 (IITXI)。 0: 禁止发送数据空的中断请求 (IITXI) 1: 允许发送数据空的中断请求 (IITXI)
6	TEIE	0	R/W	发送结束中断的允许 在 ICSR 的 TDRE 为 1 的状态下第 9 个时钟上升时，TEIE 允许或者禁止发送结束中断 (IITEI)。能通过将 TEND 或者 TEIE 清 0 来清除 IITEI。 0: 禁止发送结束的中断请求 (IITEI) 1: 允许发送结束的中断请求 (IITEI)
5	RIE	0	R/W	接收中断的允许 在将接收数据从 ICDRS 传送到 ICDRR 并且 ICSR 的 RDRF 被置 1 时，RIE 允许或者禁止接收数据满的中断请求 (IIRXI) 以及时钟同步格式时的溢出错的中断请求 (IIERI)。能通过将 RDRF 或者 RIE 清 0 来清除 IIRXI。 0: 禁止接收数据满的中断请求 (IIRXI) 1: 允许接收数据满的中断请求 (IIRXI)
4	NAKIE	0	R/W	NACK 接收中断的允许 在 ICSR 的 NACKF 或者 AL/OVE 被置位时，NAKIE 允许或者禁止 NACK 检测的中断请求 (IINAKI) 以及时钟同步格式时的溢出错 (ICSR 的 OVE 置位) 的中断请求 (IIERI)。能通过将 NACKF、AL/OVE 或者 NAKIE 清 0 来清除 IINAKI。 0: 禁止 NACK 接收的中断请求 (IINAKI) 1: 允许 NACK 接收的中断请求 (IINAKI)
3	STIE	0	R/W	停止条件检测中断的允许 在 ICSR 的 STOP 被置位时，STIE 允许或者禁止停止条件检测的中断请求 (IISTPI)。 0: 禁止停止条件检测的中断请求 (IISTPI) 1: 允许停止条件检测的中断请求 (IISTPI)
2	ACKE	0	R/W	应答位的判断选择 0: 忽视接收应答的内容，进行连续传送。 1: 当接收应答位为 1 时，中断传送。
1	ACKBR	0	R	接收应答 在发送模式中，保存从接收器件接收到的应答位的内容，写操作无效。通过将 ICCR2 的 BBSY 置 1 来清除此位。 0: 接收应答 =0 1: 接收应答 =1

位	位名	初始值	R/W	说明
0	ACKBT	0	R/W	发送应答 在接收模式中，设定应答时要发送的位的内容。 0: 应答时发送 0 1: 应答时发送 1

16.3.5 I²C 总线状态寄存器 (ICSR)

ICSR 是 8 位可读写寄存器，确认各种中断请求标志及其状态。

位:	7	6	5	4	3	2	1	0
	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

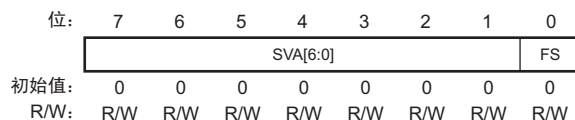
位	位名	初始值	R/W	说明
7	TDRE	0	R/W	发送数据空 [置位条件] - 当将数据从 ICDRT 传送到 ICDRS 并且 ICDRT 为空时 - 当将 TRS 置位时 - 当发行开始条件（包含重新发行）时 - 在从属模式中从接收模式变为发送模式时 [清除条件] - 当读到 1 的状态后写 0 时 - 当将数据写到 ICDRT 时 - 当通过 IITXI 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位为 0 时
6	TEND	0	R/W	发送结束 [置位条件] - I²C 总线格式: 在 TDRE 为 1 的状态下，SCL 的第 9 个时钟上升时 - 时钟同步串行格式: 当送出发送帧的最后位时 [清除条件] - 当读到 1 的状态后写 0 时 - 当将数据写到 ICDRT 时 - 当通过 IITXI 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位为 0 时
5	RDRF	0	R/W	接收数据寄存器满 [置位条件] - 当将接收数据从 ICDRS 传送到 ICDRR 时 [清除条件] - 当读到 1 的状态后写 0 时 - 当读 ICDRR 时 - 当通过 IIRXI 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位为 0 时
4	NACKF	0	R/W	无应答的检测标志 * [置位条件] - 在 ICIER 的 ACKE=1 的状态下，发送时接收器件没有应答 [清除条件] - 当读到 1 的状态后写 0 时

位	位名	初始值	R/W	说明
3	STOP	0	R/W	停止条件的检测标志 [置位条件] - 主控模式：在帧传送结束后检测到停止条件时 - 从属模式：在检测到开始条件后的第 1 字节的从属地址和 SAR 设定的地址相同的情况下，检测到停止条件时 [清除条件] - 当读到 1 的状态后写 0 时
2	AL/OVE	0	R/W	仲裁失败标志 / 溢出错误标志 在 I ² C 总线格式时，AL/OVE 表示在主导模式中总线竞争失败；在时钟同步串行格式时，表示在 RDRF=1 的状态下接收到最后的位。 在多个主控器件几乎同时要占有总线时，如果 I ² C 总线接口 2 监视到的 SDA 数据和自己输出的数据不同，就将 AL 标志置 1，表示总线被其他主控器件占有。 [置位条件] - 主控发送模式：在 SCL 的上升沿，内部 SDA 和 SDA 引脚的电平不相同 - 主控模式：在检测到开始条件的情况下，SDA 引脚为 High 电平时 - 时钟同步串行格式：在 RDRF=1 的状态下，接收到最后的位时 [清除条件] - 当读到 1 的状态后写 0 时
1	AAS	0	R/W	从属地址的识别标志 在从属接收模式中，如果紧接在开始条件后的第 1 帧和 SAR 的 SVA6 ~ SVA0 相同，此位就被置位。 [置位条件] - 当在从属接收模式中检测到从属地址时 - 当在从属接收模式中检测到一般调用地址时 [清除条件] - 当读到 1 的状态后写 0 时
0	ADZ	0	R/W	一般调用地址的识别标志 此位在 I ² C 总线格式的从属接收模式中有效 [置位条件] - 在从属接收模式中检测到一般调用地址时 [清除条件] - 当读到 1 的状态后写 0 时

【注】* 如果检测到 NACKF=1，就必须通过传送结束处理清除 NACKF，而且在清除 NACKF 前不能进行以后的发送或者接收。

16.3.6 从属地址寄存器 (SAR)

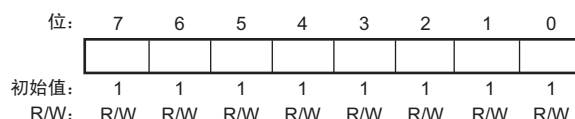
SAR 是 8 位可读写寄存器，选择格式以及设定从属地址。在 I²C 总线格式的从属模式中，如果开始条件后送来的第 1 帧的高 7 位和 SAR 的高 7 位相同，SAR 就作为从属器件运行。



位	位名	初始值	R/W	说明
7 ~ 1	SVA[6:0]	全 0	R/W	从属地址 6 ~ 0 设定与连接 I ² C 总线的其他从属器件地址不同的唯一地址。
0	FS	0	R/W	格式的选择 (位) 0: 选择 I ² C 总线格式 1: 选择时钟同步串行格式

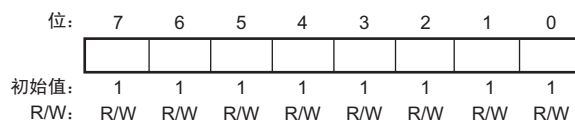
16.3.7 I²C 总线发送数据寄存器 (ICDRT)

ICDRT 是保存发送数据的 8 位可读写寄存器，当检测到移位寄存器 (ICDRS) 为空时，就将写到 ICDRT 的发送数据传送到 ICDRS，开始数据传送。在发送 ICDRS 的数据时，如果预先将下一个发送数据写到 ICDRT，就能进行连续发送。如果将 ICMR 的 MLS 位置 1 并在将发送数据写到 ICDRT 后读 ICDRT，就读到 MSB/LSB 顺序相反的数据。ICDRT 的初始值为 H'FF。



16.3.8 I²C 总线接收数据寄存器 (ICDRT)

ICDRT 是保存接收数据的 8 位寄存器。当 1 字节的数据接收结束时，就将接收数据从 ICDRS 传送到 ICDRT，并能接收下一个数据。因为 ICDRT 是接收专用寄存器，所以 CPU 不能写 ICDRT。ICDRT 的初始值为 H'FF。



16.3.9 I²C 总线移位寄存器 (ICDRS)

ICDRS 是发送或者接收数据的寄存器。在发送时，将发送数据从 ICDRT 传送到 ICDRS，然后从 SDA 引脚发送数据。在接收时，当 1 字节的数据接收结束时，就将数据从 ICDRS 传送到 ICDRR。CPU 不能直接读此寄存器。

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

16.3.10 NF2CYC 寄存器 (NF2CYC)

NF2CYC 是 8 位可读写寄存器，能选择 SCL 引脚和 SDA 引脚的噪声消除范围。有关运行的详细内容请参照“16.4.7 噪声消除电路”。

位:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	NF2CYC
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

位	位名	初始值	R/W	说明
7 ~ 1	—	全 0	R	保留位 读写值总是 0。
0	NF2CYC	0	R/W	噪声消除范围的选择 0: 能通过外围时钟消除 1 个周期以内的噪声 1: 能通过外围时钟消除 2 个周期以内的噪声

16.4 运行说明

I²C 总线接口 2 能通过 SAR 的 FS 的设定，以 I²C 总线格式或者时钟同步串行格式进行通信。

16.4.1 I²C 总线格式

I²C 总线格式和 I²C 总线时序分别如图 16.3 和图 16.4 所示。开始条件后的第 1 帧一定由 8 位构成。

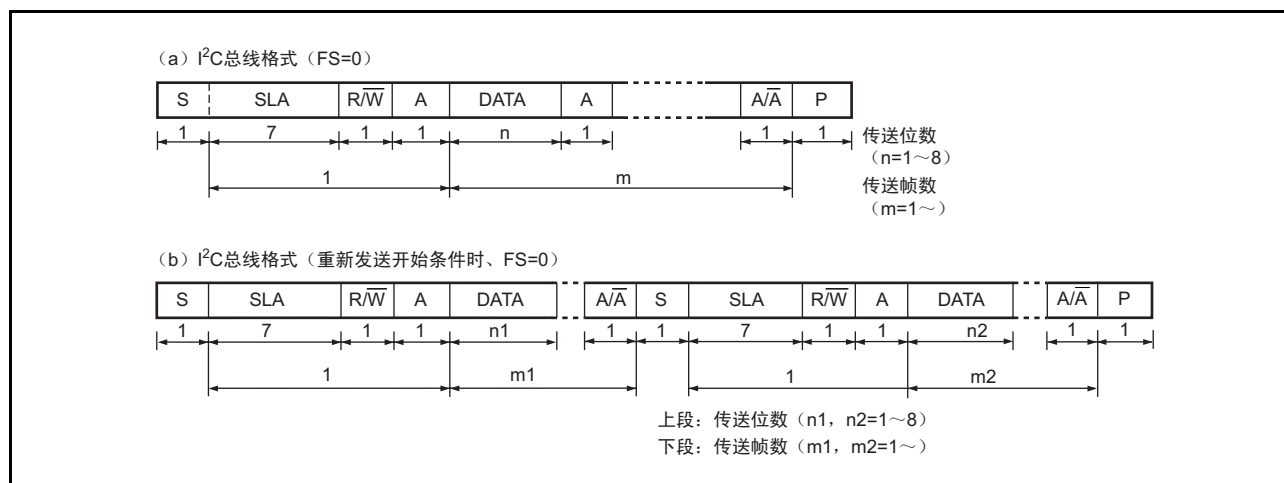


图 16.3 I²C 总线格式

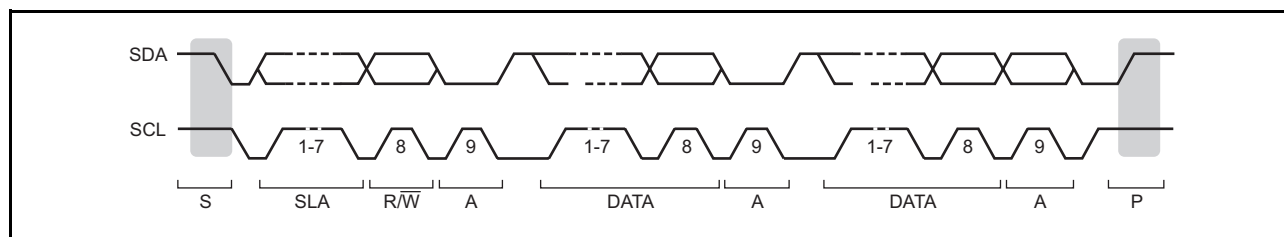


图 16.4 I²C 总线时序

【符号说明】

S : 开始条件。在 SCL=High 电平的状态下，主控制器件将 SDA 从 High 电平变为 Low 电平。

SLA : 从属地址

R/W : 表示发送和接收的方向。当 $\overline{R/W}$ 为 1 时，将数据从从属器件发送到主控制器件；当 $\overline{R/W}$ 为 0 时，将数据从主控制器件发送到从属器件。

A : 应答。接收器件将 SDA 置 Low 电平。

DATA : 发送和接收的数据

P : 停止条件。在 SCL=High 电平的状态下，主控制器件将 SDA 从 Low 电平变为 High 电平。

16.4.2 主控发送

在发送模式中，主控制器输出发送时钟和发送数据，从属器件返回应答。有关发送模式的运行顺序，请参照图 16.5 和图 16.6。

发送模式的发送步骤和运行如下所示：

1. 将ICCR1的ICE位置1，然后设定ICMR的MLS和ICCR1的CKS3～CKS0等（初始设定）。
2. 读ICCR2的BBSY标志，在确认总线处于释放状态后，将ICCR1的MST和TRS设定为主控发送模式，然后写BBSY=1和SCP=0（发行开始条件），生成开始条件。
3. 在确认ICSR的TDRE已被置位后，将发送数据写到ICDRT（第1个字节表示从属地址和R/W的数据）。此时，TDRE自动清0，在将数据从ICDRT传送到ICDRS后重新将TDRE置位。
4. 在TDRE置位的状态下结束1字节数据的发送，并在发送时钟的第9个时钟的上升沿将ICSR的TEND置位。读ICIER的ACKBR，在确认已选择从属器件后，将第2个字节的数据写到ICDRT。因为在ACKBR为1时从属器件未被认识，所以发行停止条件。通过写BBSY=0和SCP=0发行停止条件。另外，在数据准备完成前或者在发送停止条件前，将SCL固定为Low电平。
5. 每当TDRE被置位时，将第2个字节以后的发送数据写到ICDRT。
6. 如果将要发送的字节数写到ICDRT，就在TDRE被置位的状态下等待TEND的置位（最后字节发送结束），或者在ICIER的ACKE被置位的状态下等待接收器件的NACK（ICSR的NACKF=1）。然后发行停止条件，清除TEND或者NACKF。
7. 如果ICSR的STOP被置位，就返回到从属接收模式。

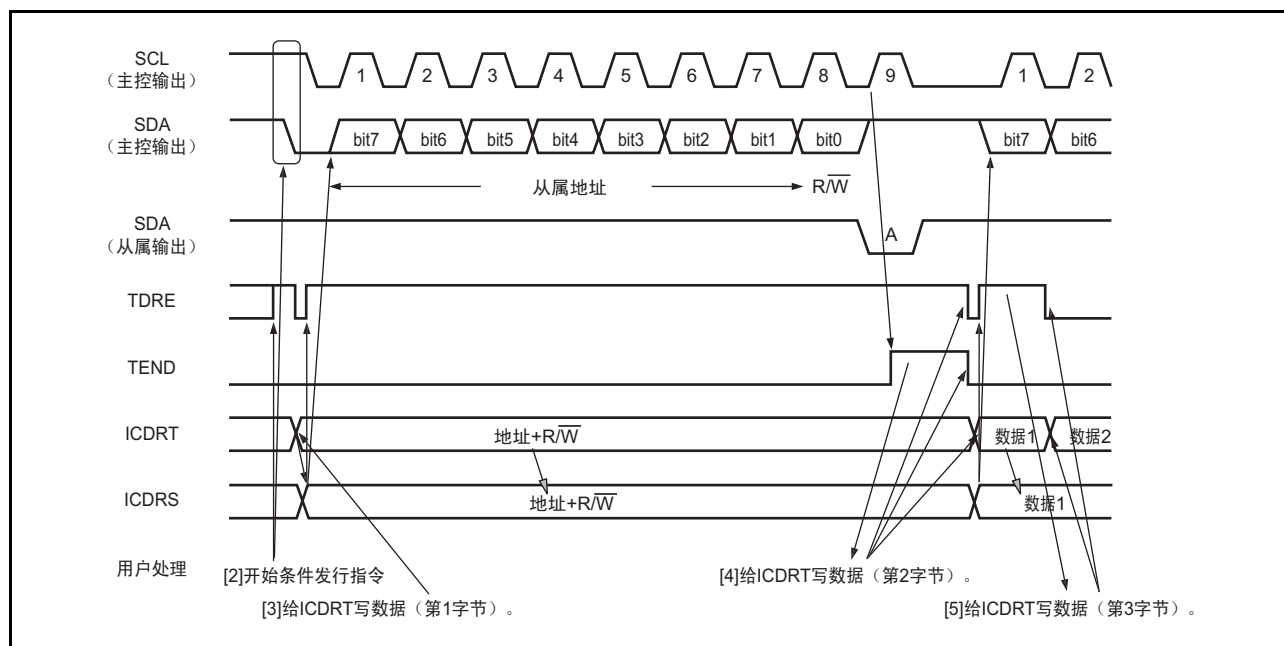


图 16.5 主控发送模式的运行时序（1）

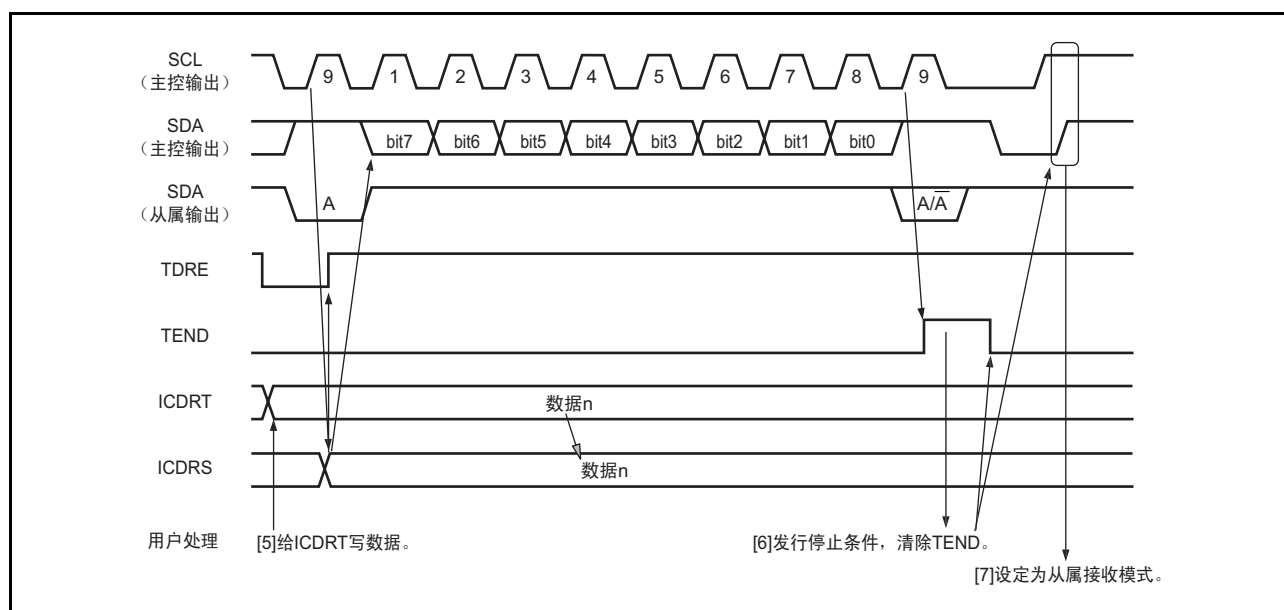


图 16.6 主控发送模式的运行时序 (2)

16.4.3 主控接收

在主控接收模式中，主控制器输出接收时钟，从从属器件接收数据并返回应答。有关主控接收模式的运行时序，请参照图 16.7 和图 16.8。

主控接收模式的接收步骤和运行如下：

1. 在清除ICSR的TEND后，清除ICCR1的TRS，从主控发送模式转换为主控接收模式，然后清除TDRE。
2. 当虚读ICDRT时，开始接收*，与内部时钟同步输出接收时钟并接收数据。在接收时钟的第9个时钟，主控制器将ICIER的ACKBT设定的电平输出到SDA。
3. 结束1帧数据的接收，并在接收时钟的第9个时钟的上升沿将ICSR的RDRF置位。此时，能通过读ICDRT来读取所接收的数据，同时清除RDRF。
4. 如果在每次RDRF被置位时读ICDRT，就能进行连续接收。如果在RDRF置位的状态下因其他处理而推迟ICDRT的读操作并且第8个时钟下降，就在读ICDRT前将SCL固定为Low电平。
5. 在下次接收为最后1帧的情况下，要在读ICDRT前将ICCR1的RCVD置位，从而能在下次接收后处于可发行停止条件的状态。
6. 如果在接收时钟的第9个时钟的上升沿将RDRF置位，就发行停止条件。
7. 如果ICSR的STOP被置位，就读ICDRT，然后清除RCVD。
8. 返回到从属接收模式。

【注】 * 如果只要接收 1 个字节，就必须在 ICCR1 的 RCVD 被置位后虚读 ICDRT。

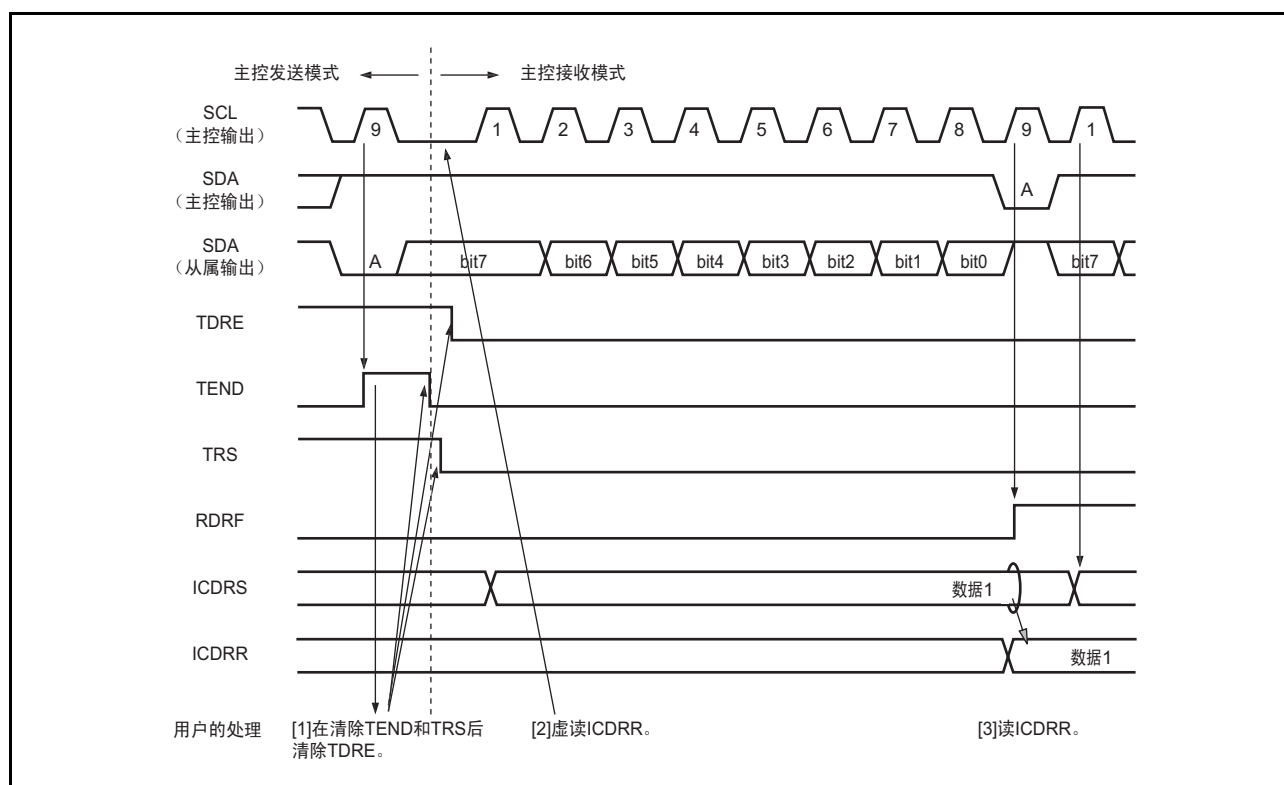


图 16.7 主控接收模式的运行时序 (1)

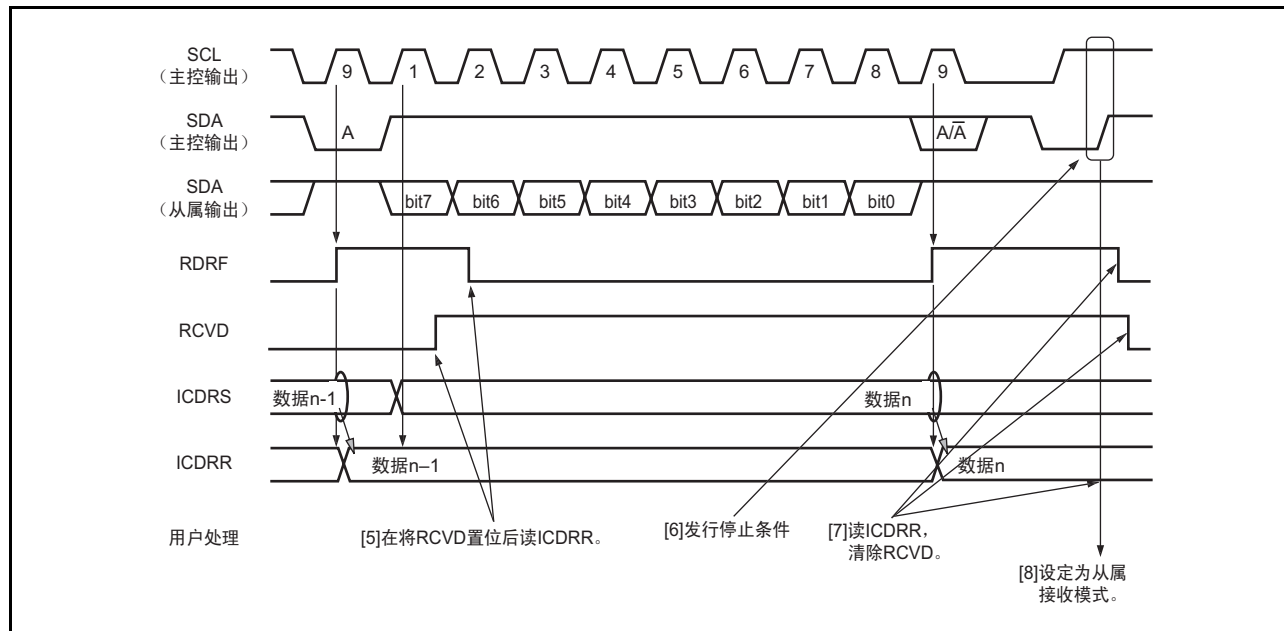


图 16.8 主控接收模式的运行时序 (2)

16.4.4 从属发送

在从属发送模式中，从属器件输出发送数据，主控器件输出接收时钟并返回应答。有关从属发送模式的运行时序，请参照图 16.9 和图 16.10。

从属发送模式的发送步骤和运行如下:

1. 将ICCR1的ICE位置1，然后设定ICMR的MLS和ICCR1的CKS3~CKS0等（初始设定）。将ICCR1的MST和TRS设定为从属接收模式，等待相同的从属地址。
2. 如果检测到开始条件后的第1帧从属地址相同，从属器件就在第9个时钟的上升沿将ICIER的ACKBT设定的电平输出到SDA。此时，如果第8位数据（R/W）为1，ICCR1的TRS和ICSR的TDRE就被置位，自动转换为从属发送模式。如果在每次TDRE被置位时将发送数据写到ICDRT，就能进行连续发送。
3. 如果在最后的发送数据写到ICDRT后将TDRE置位，就在TDRE=1的状态下等待ICSR的TEND的置位。如果TEND被置位，就清除TEND。
4. 为了结束处理，清除TRS并虚读ICDRR，从而能释放SCL。
5. 清除TDRE。

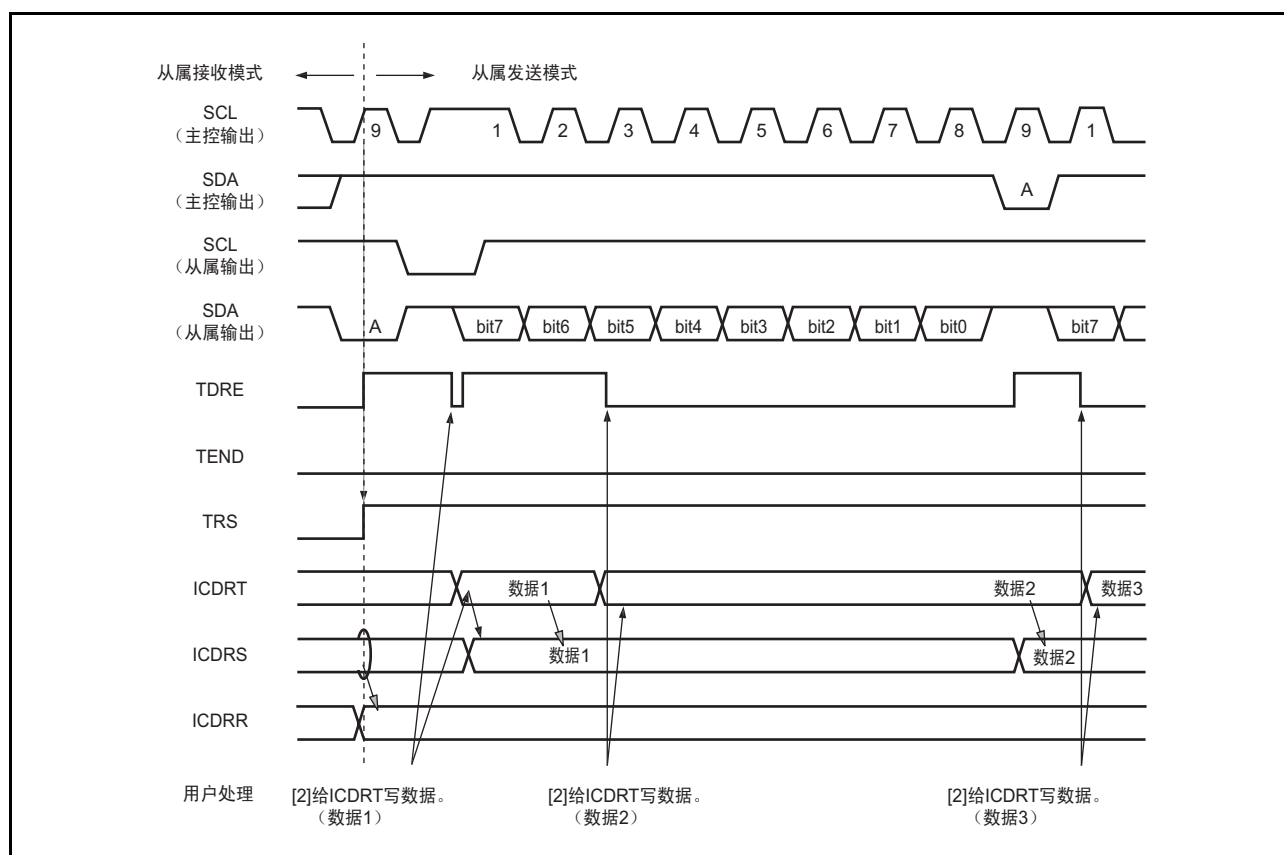


图 16.9 从属发送模式的运行时序 (1)

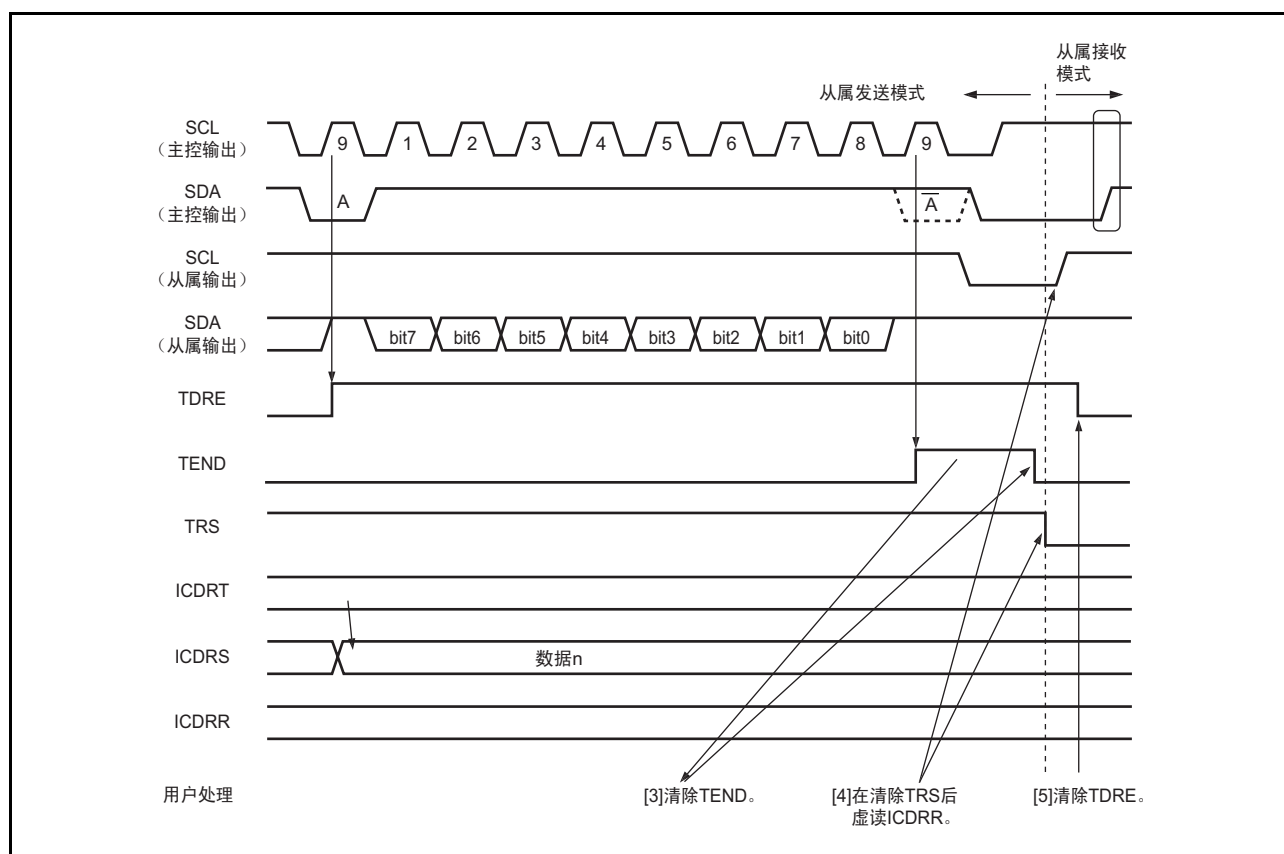


图 16.10 从属发送模式的运行时序 (2)

16.4.5 从属接收

在从属接收模式中，主控器件输出发送时钟和发送数据，从属器件返回回答。有关从属接收模式的运行时序，请参照图 16.11 和图 16.12。

从属接收模式的接收步骤和运行如下所示:

1. 将ICCR1的ICE位置1，然后设定ICMR的MLS和ICCR1的CKS3~CKS0等（初始设定）。将ICCR1的MST和TRS设定为从属接收模式，等待相同的从属地址。
2. 如果检测到开始条件后的第1帧从属地址相同，从属器件就在第9个时钟的上升沿将ICIER的ACKBT设定的电平输出到SDA。因为ICSR的RDRF同时被置位，所以虚读ICDRR（读取的数据表示从属地址+R/W，所以不要）。
3. 每当RDRF被置位时，虚读ICDRR。如果在RDRF置位的状态下第8个时钟下降，就在读ICDRR前将SCL固定为Low电平。读ICDRR前所更改的应答设定（返回给主控器件）反映在下一个传送帧。
4. 同样，通过读ICDRR来读取最后的字节。

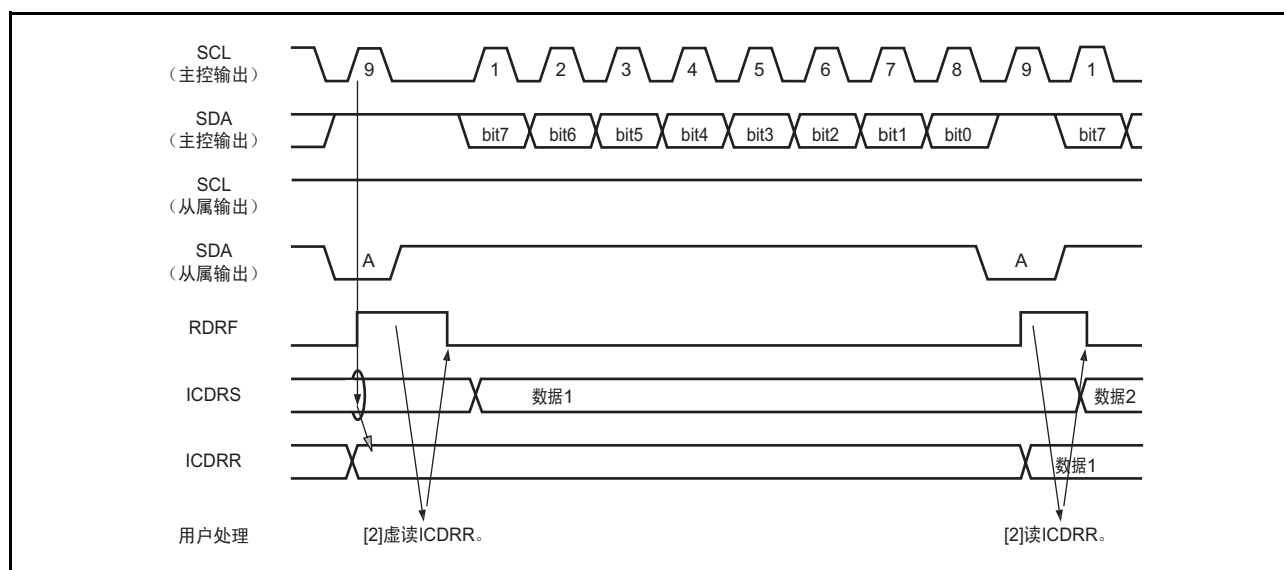


图 16.11 从属接收模式的运行时序 (1)

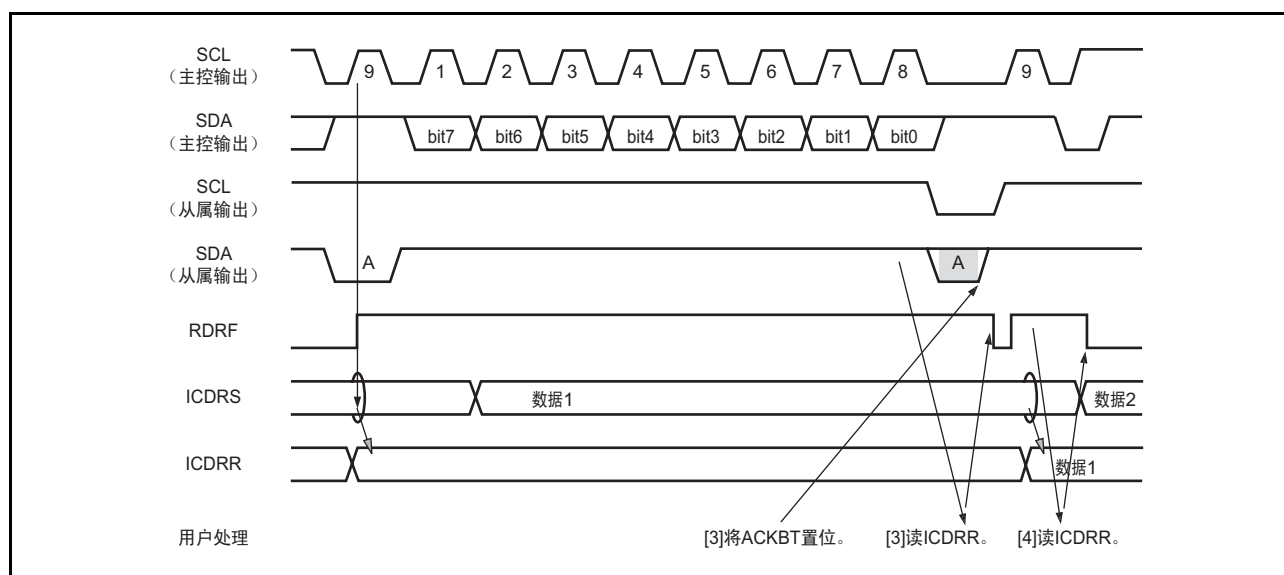


图 16.12 从属接收模式的运行时序 (2)

16.4.6 时钟同步串行格式

此模块能通过将 SAR 的 FS 置 1，以时钟同步串行格式进行通信。当 ICCR1 的 MST=1 时，从 SCL 输出传送时钟；当 MST=0 时，输入外部时钟。

(1) 数据的传送格式

时钟同步串行格式的传送格式如图 16.13 所示。

在 SCL 时钟的下降沿和下一个下降沿之间输出传送数据，并确保 SCL 时钟上升沿的数据。能通过 ICMR 的 MLS 选择 MSB 先或者 LSB 先的数据传送顺序，还能通过 ICCR2 的 SDAO，在传送待机时更改 SDA 的输出电平。

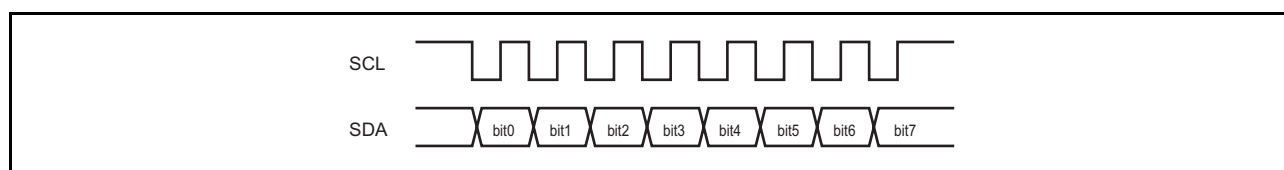


图 16.13 时钟同步串行格式的传送格式

(2) 发送

在发送模式中，与传送时钟的下降沿同步从 SDA 输出发送数据。当 ICCR1 的 MST=1 时，输出传送时钟；当 MST=0 时，输入传送时钟。有关发送模式的运行时序请参照图 16.14。

发送模式的步骤和运行如下所示：

1. 将 ICCR1 的 ICE 位置 1，然后设定 ICCR1 的 MST 和 CKS3 ~ CKS0 等（初始设定）。
2. 设定 ICCR1 的 TRS，转换为发送模式，ICSR 的 TDRE 被置位。
3. 在确认 TDRE 已被置位时，将发送数据写到 ICDRT，数据从 ICDRT 传送到 ICDRS 并且 TDRE 自动置位。如果在每次 TDRE 被置位时将数据写到 ICDRT，就能进行连续发送。要从发送模式转换为接收模式时，必须在 TDRE 置位的状态下清除 TRS。

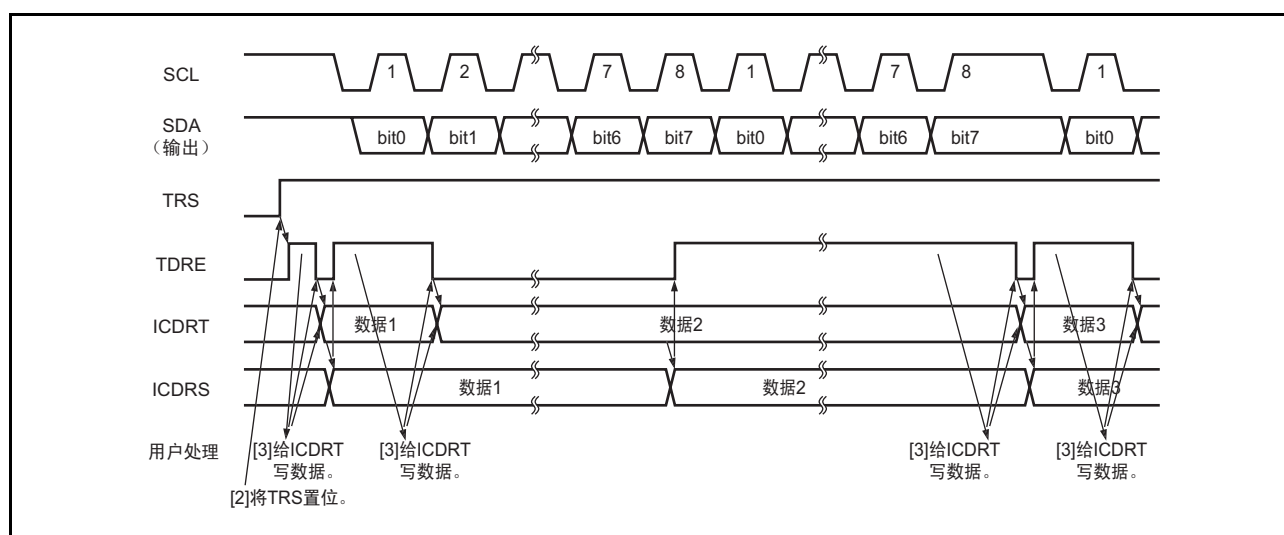


图 16.14 发送模式的运行时序

(3) 接收

在接收模式中，在传送时钟的上升沿锁存数据。当 ICCR1 的 MST=1 时，输出传送时钟；当 MST=0 时，输入传送时钟。有关接收模式的运行时序请参照图 16.15。

接收模式的步骤和运行如下：

1. 将 ICCR1 的 ICE 位置 1，然后设定 ICCR1 的 MST 和 CKS3 ~ CKS0 等（初始设定）。
2. 在输出传送时钟时，设定 MST=1，开始输出接收时钟。
3. 当接收结束时，数据从 ICDRS 传送到 ICDRR，并且 ICSR 的 RDRF 被置位。当 MST=1 时，由于为可接收下一个字节的状态，所以能连续输出时钟。如果在每次 RDRF 被置位时读 ICDRR，就能进行连续接收。如果在 RDRF 置位的状态下第 8 个时钟上升，就检测到溢出，将 ICSR 的 AL/OVE 置位。此时 ICDRR 保存前一个接收数据。

4. 当MST=1时，为了停止接收，先将ICCR1的RCVD置位，然后读ICDRR。于是在下一个字节接收结束后，SCL固定为High电平。

【注】 当MST=1并且只要接收1个字节时，必须按照以下步骤进行，运行时序请参照图16.16。

1. 将ICCR1的ICE位置1，然后设定ICCCR1的CKS3～CKS0等（初始设定）。
2. 在ICCR1的RCVD位为0的状态下，设定MST=1，开始输出接收时钟。
3. 必须在确认ICMR的BC2位已被置1后设定ICCR1的RCVD=1。于是在输出1个字节的接收时钟后SCL固定为High电平。

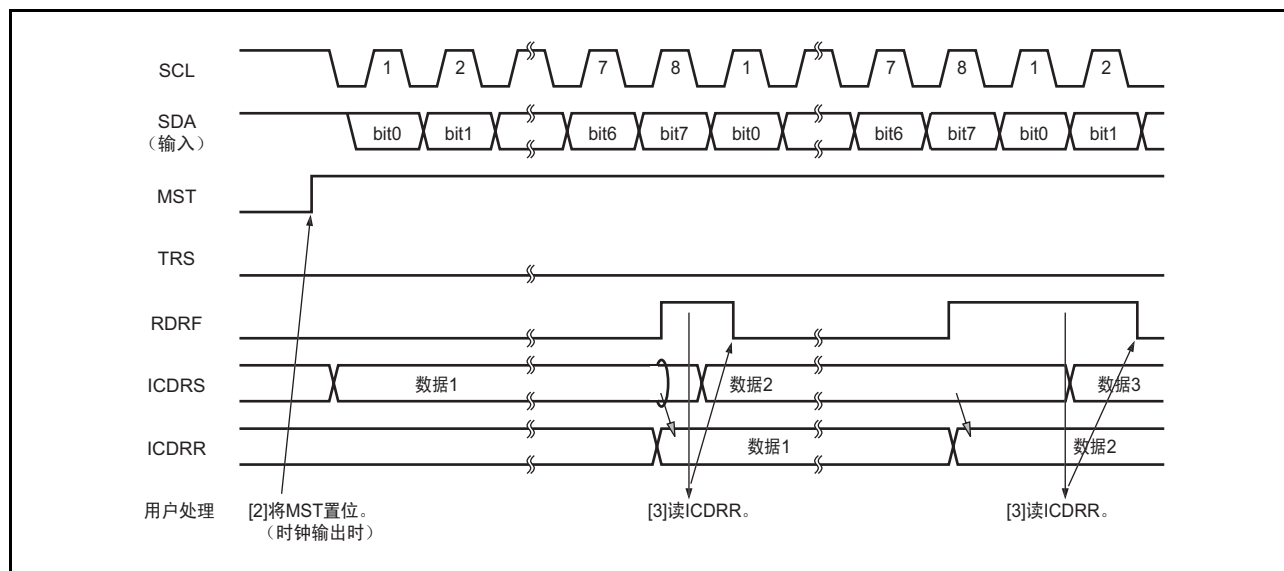


图 16.15 接收模式的运行时序

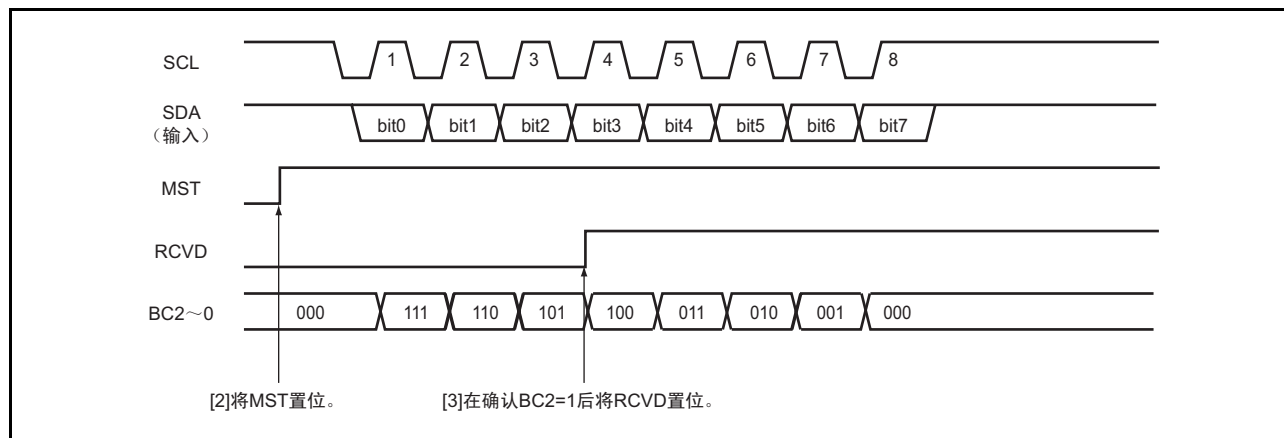


图 16.16 接收1字节数据的运行时序

16.4.7 噪声消除电路

SCL 引脚状态和 SDA 引脚状态经过噪声消除电路被取到内部，噪声消除电路的框图如图 16.17 所示。

噪声消除电路由 3 段串联的锁存电路和一致检测电路构成。通过外围时钟对 SCL 引脚输入信号（或者 SDA 引脚输入信号）进行采样，当 NF2CYC 寄存器的值为 0 并且 2 个锁存器的输出信号相同时或者 NF2CYC 寄存器的值为 1 并且 3 个锁存器的输出信号相同时，将电平传送到后段。如果锁存器的输出信号不相同，就保持以前的值。

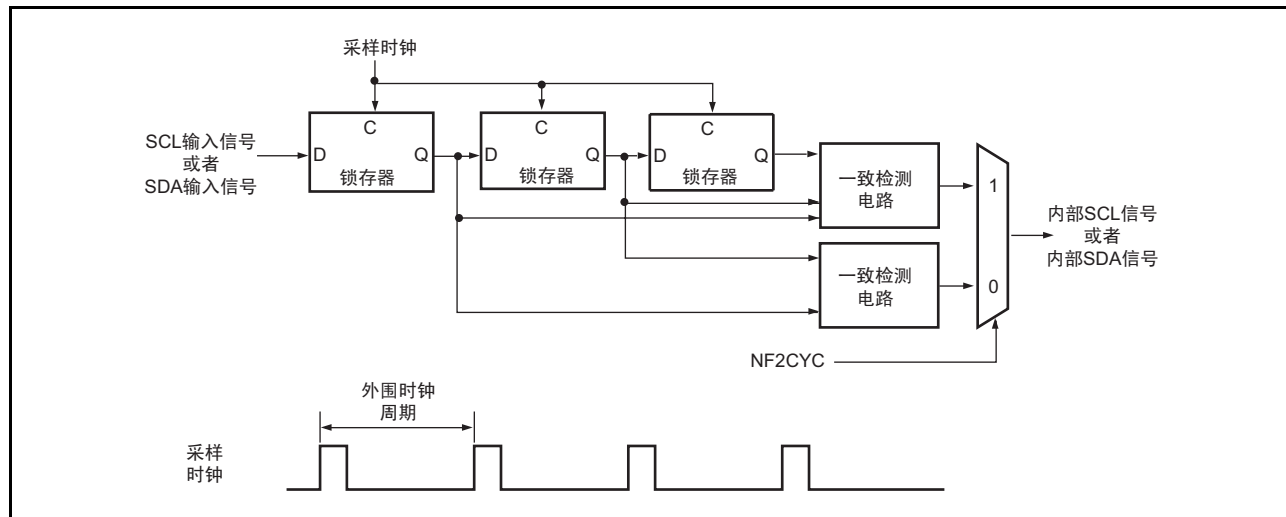


图 16.17 噪声消除电路的框图

16.4.8 使用例

在使用 I²C 总线接口 2 时，各模式的流程图例子如图 16.18 ~ 图 16.21 所示。

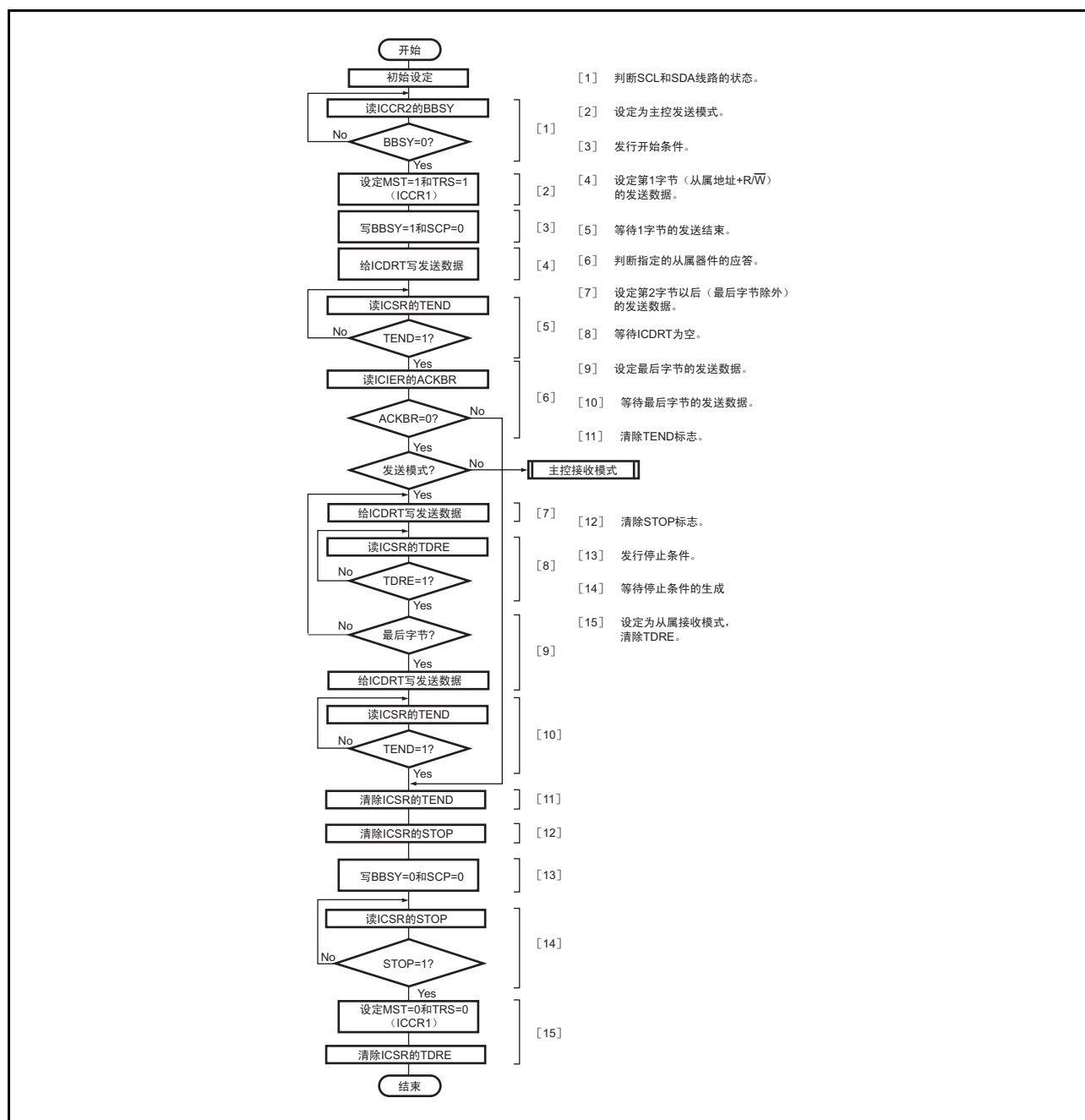


图 16.18 主控发送模式的流程图例子

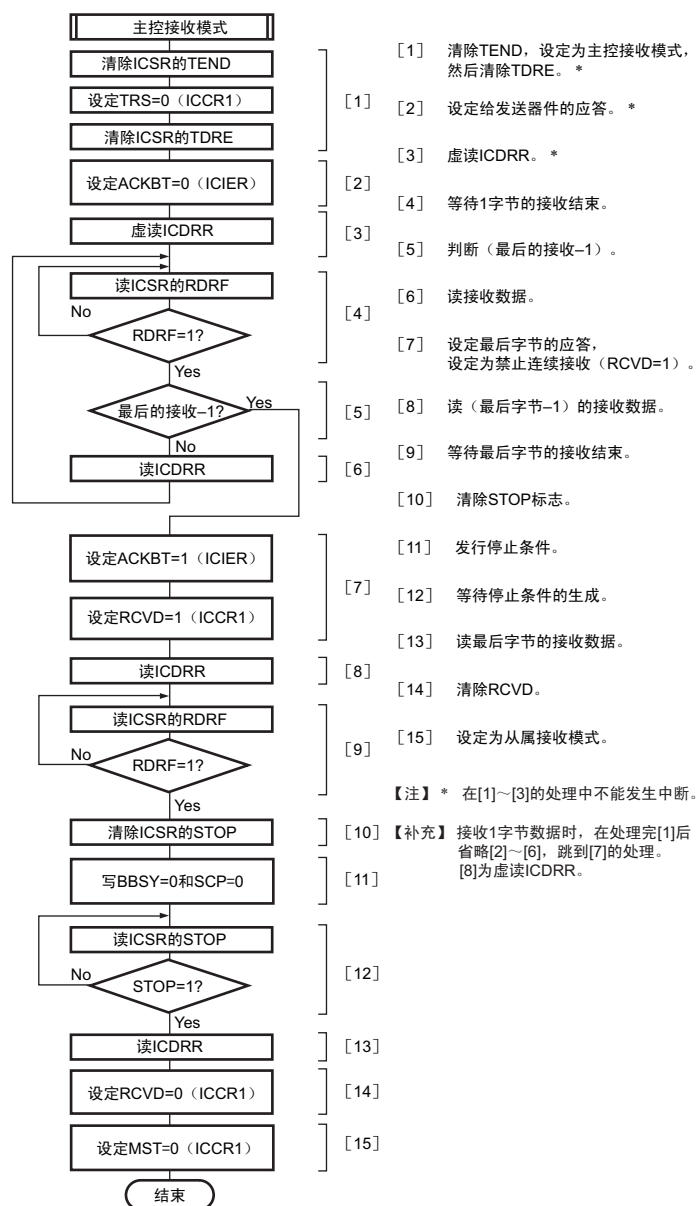


图 16.19 主控接收模式的流程图例子

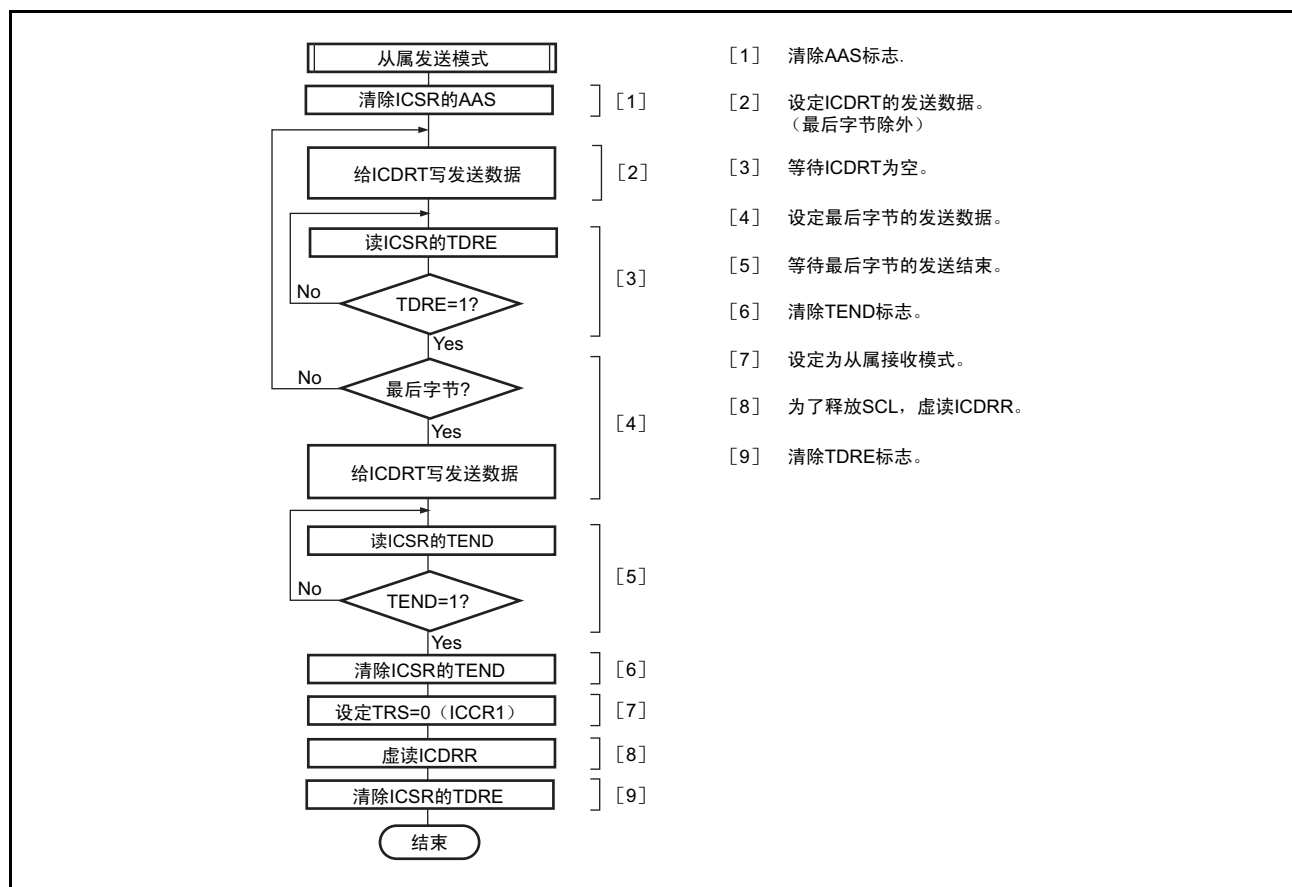


图 16.20 从属发送模式的流程图例子

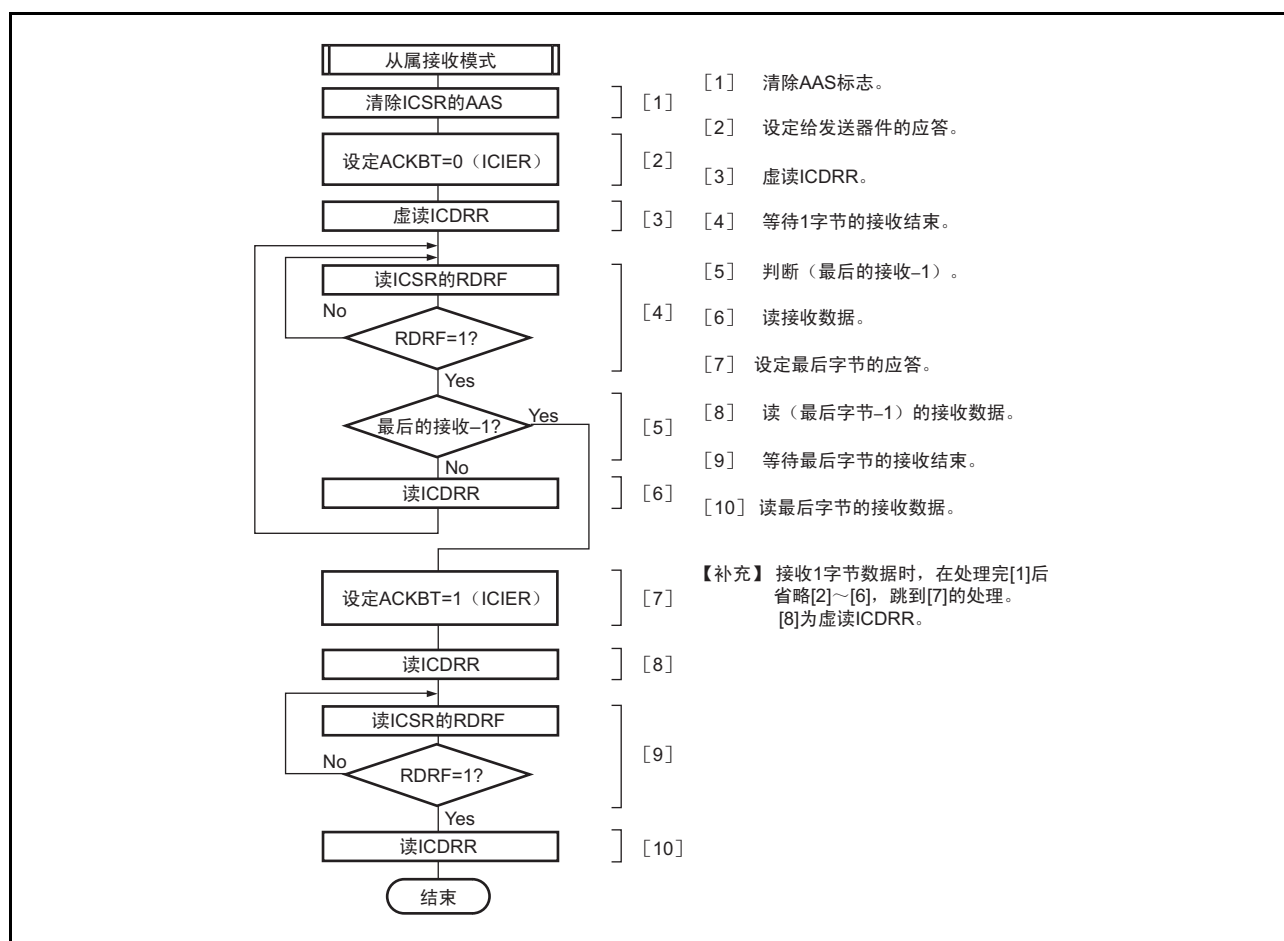


图 16.21 从属接收模式的流程图例子

16.5 I²C2 的中断源

此模块的中断请求有发送数据空、发送结束、接收数据满、NACK 的检测、停止条件的检测和仲裁失败 / 溢出错误中断，各中断请求的内容如表 16.4 所示。

表 16.4 中断请求一览表

中断请求	略称	中断条件	I ² C 总线格式	时钟同步串行格式	DTC 的启动
发送数据空	IITXI	(TDRE=1) · (TIE=1)	○	○	○
发送结束	IITEI	(TEND=1) · (TEIE=1)	○	○	×
接收数据满	IIRXI	(RDRF=1) · (RIE=1)	○	○	○
停止条件的检测	IISTPI	(STOP=1) · (STIE=1)	○	×	×
NACK 的检测	IINAKI	{(NACKF=1)+(AL=1)} · (NAKIE=1)	○	×	×
仲裁失败 / 溢出错误			○	○	×

当表 16.4 的中断条件为 1 时，CPU 就进行中断异常处理。在异常处理中，必须清除各自的中断源。但是必须注意：通过将发送数据写到 ICDRT，自动清除 TDRE 和 TEND；通过读 ICDRR，自动清除 RDRF。尤其在将发送数据写到 ICDRT 的同时 TDRE 再次被置位，此时，如果清除 TDRE，就有可能多发送 1 个字节。另外，在 DTC 进行指定次数的传送时，自动清除 TDRE、TEND 和 RDRF，但是在传送结束时，不自动清除 TDRE、TEND 和 RDRF。

16.6 DTC 的运行

在 I²C 总线格式时，通过从属地址和 $\overline{R/W}$ 位选择从属器件和传送方向，通过应答位进行接收确认和最后帧的表示，所以使用 DTC 进行的数据连续传送需要通过中断结合 CPU 处理来进行。

使用 DTC 的处理例子如表 16.5 所示，假设在从属模式中知道传送数据的个数。

表 16.5 使用 DTC 的运行例子

项目	主控发送模式	主控接收模式	从属发送模式	从属接收模式
从属地址 + $\overline{R/W}$ 位的发送 / 接收	通过 DTC 发送 (写 ICDR)	通过 CPU 发送 (写 ICDR)	通过 CPU 接收 (读 ICDR)	通过 CPU 接收 (读 ICDR)
虚拟数据的读操作	—	通过 CPU 处理 (读 ICDR)	—	—
数据的发送 / 接收	通过 DTC 发送 (写 ICDR)	通过 DTC 接收 (读取 ICDR)	通过 DTC 发送 (写 ICDR)	通过 DTC 接收 (读 ICDR)
最后帧的处理	不要	通过 CPU 接收 (读 ICDR)	不要	通过 CPU 接收 (读 ICDR)
DTC 发送数据的帧数的 设定	发送：实际的数据个数 +1 (+1 是指从属地址 + $\overline{R/W}$ 位数)	接收：实际的数据个数	发送：实际的数据个数	接收：实际的数据个数

16.7 位同步电路

在设定主控模式时，此模块在以下 2 种状态下有可能缩短 High 电平期间，所以需要监视 SCL，按位边取得同步边进行通信。

- 当 SCL 被从属器件拉到 Low 电平时
- 因 SCL 线的负载（负载电容或者上拉电阻）而使 SCL 的上升变得缓慢时

位同步电路的时序如图 16.22 所示，SCL 从 Low 电平输出变为 Hi-Z 到监视 SCL 的时间如表 16.6 所示。

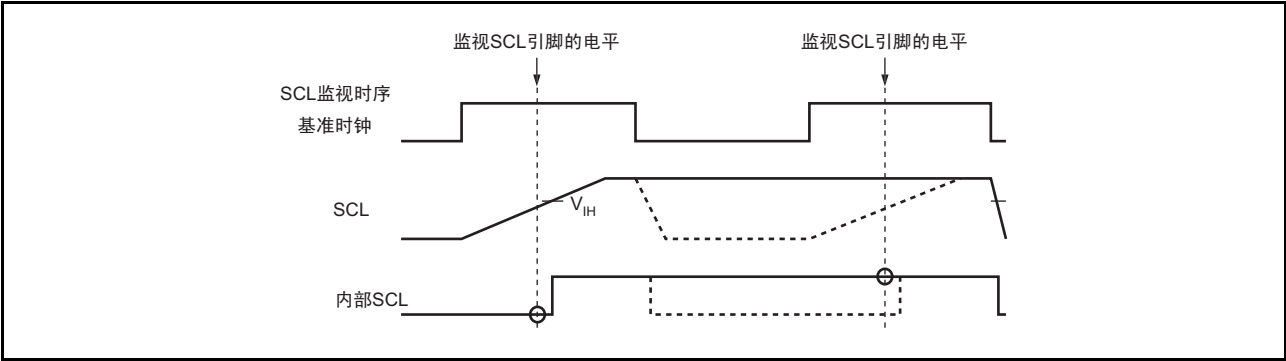


图 16.22 位同步电路的时序

表 16.6 SCL 的监视时间

CKS3	CKS2	NF2CYC	SCL 的监视时间 *1
0	0	0	6.5 t _{pcyc} *2
		1	5.5 t _{pcyc} *2
	1	0	18.5 t _{pcyc} *2
		1	17.5 t _{pcyc} *2
1	0	0	16.5 t _{pcyc} *2
		1	15.5 t _{pcyc} *2
	1	0	40.5 t _{pcyc} *2
		1	39.5 t _{pcyc} *2

【注】 *1 监视从“SCL 监视时序的基准时钟”的上升沿到“监视 SCL 的时间”后的 SCL 引脚电平。

*2 t_{pcyc} 表示外围时钟的周期。

16.8 使用时的注意事项

16.8.1 模块待机模式的设定

能通过待机控制寄存器的设定，禁止或者允许 I²C2 模块的运行，初始值为 I²C2 停止运行。能通过解除模块待机模式允许寄存器的存取，详细内容请参照“第 24 章 低功耗模式”。

16.8.2 停止条件和开始条件（重新传送）的发行

必须在认识到第 9 个时钟下降沿后发行停止条件和开始条件（重新发送）。能通过检查 I²C 总线控制寄存器 2 (ICCR2) 的 SCLO 位认识第 9 个时钟下降沿。如果在下述 1. 或 2. 的条件下并且在特定的时序发行停止条件和开始条件（重新发送），就有可能无法正常地输出停止条件和开始条件（重新发送），否则就没有问题。

1. 因 SCL 的总线负载（负载电容或者上拉电阻）而使 SCL 的上升时间长于“16.7 位同步电路”中规定的时间
2. 因从属器件拉长了第 8 个时钟和第 9 个时钟的 Low 电平时间而使位同步电路起作用时

16.8.3 开始条件与停止条件的连续发行

不能连续发行开始条件和停止条件。如果要连续发行，就必须在发送从属地址后发行停止条件。

16.8.4 使用多主控时的设定

1. 传送率的设定值
在使用多主控时，传送率的设定值必须至少为其他主控中最大传送率的 1/1.8。例如，其他主控中最大传送率为 400kbps 时，本 LSI 的 IIC 的传送率的设定值必须至少为 223kbps ($=400/1.8$)。
2. ICCR1 的 MST 位和 TRS 位
在使用多主控时，必须用 MOV 指令设定 ICCR1 的 MST 位和 TRS 位。
3. 发生仲裁失败时
如果发生仲裁失败，就必须确认 ICCR1 的 MST 位和 TRS 位是否为 0。如果 ICCR1 的 MST 位和 TRS 位不是 0 时，就必须清 0。

16.8.5 主控接收模式中的 ICDRR 的读操作

在主控接收模式中，必须在 SCL 的第 8 个时钟的上升前读 ICDRR。否则，就在 ICSR 的 RDRF 位为 1 的状态下接收下一个数据时，第 8 个时钟固定为低电平，并输出第 9 个时钟。

如果在 SCL 的第 8 个时钟的上升前来不及读 ICDRR，就必须在将 ICRR1 的 RCVD 位置 1 后，按字节进行通信。

16.8.6 仿真器的支持

E200F 不支持 I²C2 运行。在调试 I²C2 时，必须使用 E10A。

第 17 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。

17.1 特点

- 分辨率: 12 位
- 输入通道数:
SH7136 有 12 个通道 (内置 2 个独立的 A/D 转换模块)
SH7137 有 16 个通道 (内置 2 个独立的 A/D 转换模块)
- 高速转换
P ϕ =40MHz 运行时: 平均 1 个通道的转换时间=1.25 μ s、AD 时钟=40MHz、转换状态=50 个状态
- 运行模式: 2 种
单周期扫描模式: 1~8 通道的连续 A/D 转换
连续扫描模式: 1~8 通道的重复 A/D 转换
- 12 位 A/D 数据寄存器
SH7136 有 12 个 16 位 A/D 数据寄存器 (ADDR)。其中, A/D_0 有 4 个, A/D_1 有 8 个。
SH7137 有 16 个 16 位 A/D 数据寄存器 (ADDR)。其中, A/D_0 和 A/D_1 各有 8 个。将 A/D 转换结果保存到对应各输入通道的 A/D 数据寄存器 (ADDR)。
- 采样和保持功能
因为本 LSI 的 A/D 转换器内置了采样和保持电路, 所以能简单地构成外部模拟输入电路。而且, 通道 0~2 和通道 8~10 拥有各通道专用的采样和保持电路, 所以能实现多个通道的同时采样。能同时采样的组合如下:
A 组 (GrA): 同时对通道 0、1、2 中选择的模拟输入引脚进行采样
B 组 (GrB): 同时对通道 8、9、10 中选择的模拟输入引脚进行采样
- A/D 转换开始请求: 3 种
软件: ADCR 的 ADST 位的设定
定时器: MTU2 的 TRGAN、TRG0N、TRG4AN、TRG4BN, MTU2S 的 TRGAN、TRG4AN、TRG4BN
外部触发: ADTRG (LSI 引脚)
- 模拟输入通道的选择功能
能通过 A/D 模拟输入通道选择寄存器 (ADANSR) 的设定, 选择任意的通道进行 A/D 变换。
- 支持 A/D 转换结束中断和 DTC 传送功能
在 A/D 转换结束时, 能对 CPU 产生 A/D 转换结束的中断请求 (ADI_3 和 ADI_4), 还能通过 ADI_3 和 ADI_4 启动 DTC。

A/D 转换器的框图如图 17.1 所示。

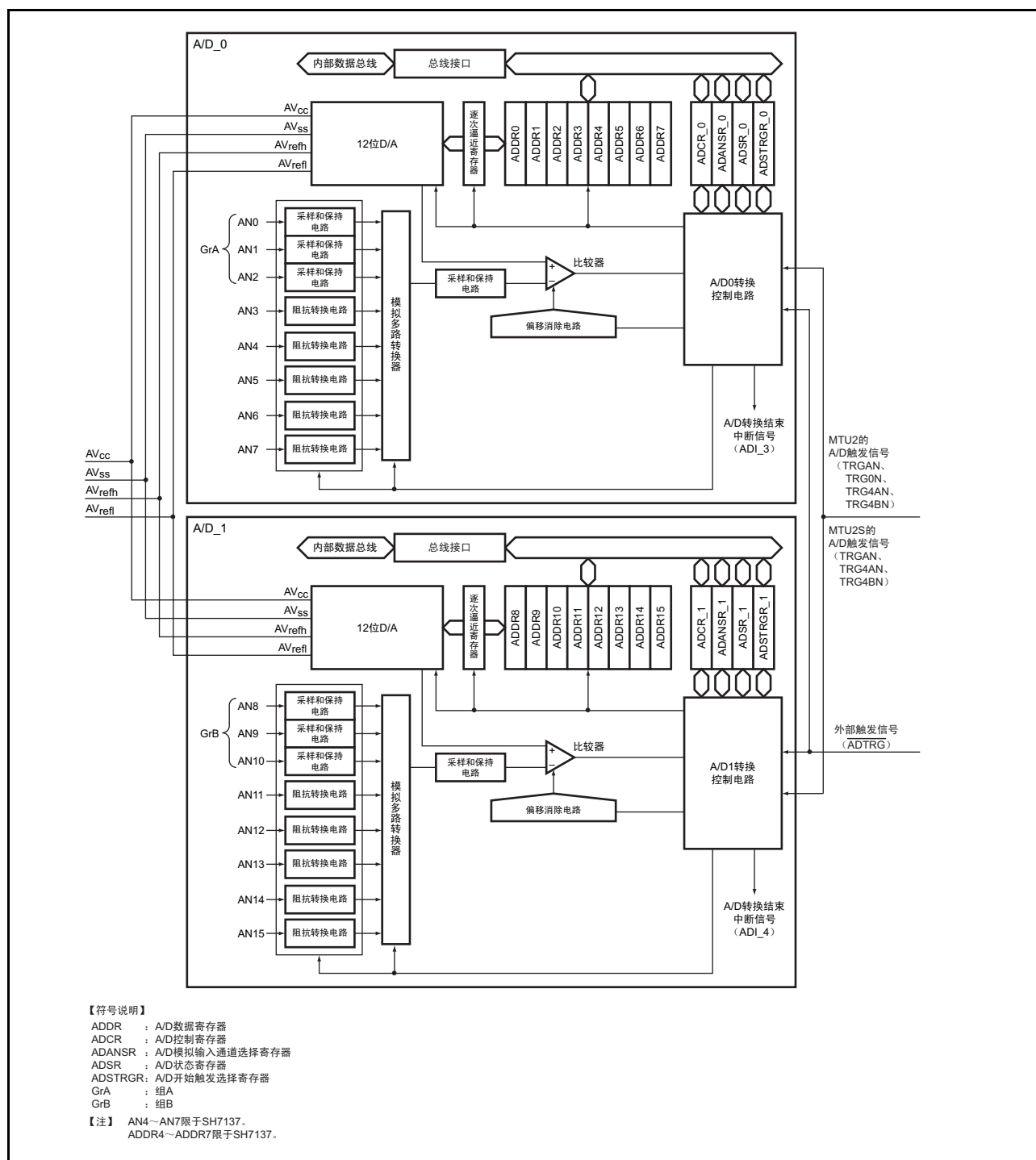


图 17.1 A/D 转换器的框图

17.2 输入 / 输出引脚

A/D 转换器使用的引脚如表 17.1 所示。有关引脚使用时的注意事项，请参照“17.7 使用时的注意事项”。

表 17.1 引脚结构

模块区分	引脚名	输入 / 输出	功能	产品区分	
				SH7137	SH7136
共通	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})	○	○
	ADTRG	输入	A/D 外部触发的输入引脚	○	○
A/D 模块 0 (A/D_0)	AN0	输入	模拟输入引脚 0（A 组）	○	○
	AN1	输入	模拟输入引脚 1（A 组）	○	○
	AN2	输入	模拟输入引脚 2（A 组）	○	○
	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（B 组）	○	○
	AN9	输入	模拟输入引脚 9（B 组）	○	○
	AN10	输入	模拟输入引脚 10（B 组）	○	○
	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	○	○

17.3 寄存器说明

A/D 转换器有以下的寄存器。

表 17.2 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
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

17.3.1 A/D 控制寄存器 _0、_1 (ADCR_0、ADCR_1)

ADCR 是 8 位可读写寄存器，选择 A/D_0 和 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	初始值	R/W	说明
7	ADST	0	R/W	A/D 开始 如果将此位清 0，就中止 A/D 转换，进入待机状态；如果将此位置 1，就开始 A/D 转换。在单周期扫描模式中，当所选通道的 A/D 转换结束时，此位就自动清除；在连续扫描模式中，在通过软件、复位、软件待机模式或者模块待机模式清除此位前，依次对所选的通道进行连续转换。
6	ADCS	0	R/W	A/D 连续扫描 选择扫描模式中的单周期扫描模式或者连续扫描模式。此位只在扫描模式时有效。 0: 单周期扫描 1: 连续扫描 必须在 ADST 为 0 的状态下转换运行模式。
5	ACE	0	R/W	自动清除的允许 允许或者禁止在 CPU 和 DTC 读 ADDR 寄存器后自动清除 ADDR 寄存器。如果将此位置 1，就在 CPU 和 DTC 读 ADDR 寄存器后自动将 ADDR 寄存器清 0。通过此功能，能检测到 ADDR 寄存器未更新的故障。 0: 禁止在读 ADDR 后自动清除 ADDR 1: 允许在读 ADDR 后自动清除 ADDR
4	ADIE	0	R/W	A/D 中断的允许 允许或者禁止对 CPU 产生 A/D 转换结束中断 (ADI_3 和 ADI_4)。为了防止误动作，必须在 ADST 为 0 的状态下切换运行模式。 当 A/D 转换结束并且 ADSR 寄存器的 ADF 位被置 1 时，如果此位已被置 1，就对 CPU 产生 A/D 转换结束中断 (ADI_3 和 ADI_4)。能通过将 ADF 或者 ADIE 清 0 来清除 ADI_3 和 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	—	全 0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
1	TRGE	0	R/W	触发的允许 允许或者禁止由外部触发输入 ($\overline{\text{ADTRG}}$)、或者 MTU2 和 MTU2S 的 A/D 转换开始触发 (MTU2 有 TRGAN、TRG0N、TRG4AN、TRG4BN, MTU2S 有 TRGAN、TRG4AN、TRG4BN) 控制的 A/D 转换开始。有关外部触发和 MTU2、MTU2S 的 A/D 转换开始触发的选择, 请参照 EXTRG 位的说明。 0: 禁止由外部触发、或者 MTU2 和 MTU2S 的 A/D 转换开始触发控制的 A/D 转换开始 1: 允许由外部触发、或者 MTU2 和 MTU2S 的 A/D 转换开始触发控制的 A/D 转换开始
0	EXTRG	0	R/W	触发的选择 是选择外部触发 ($\overline{\text{ADTRG}}$) 还是选择 MTU2 和 MTU2S 的 A/D 转换开始触发作为 A/D 转换开始触发。 在设定外部触发 (EXTRG=1) 的情况下, 如果在将 TRGE 置 1 后给 $\overline{\text{ADTRG}}$ 引脚输入低电平脉冲, A/D 转换就能检测到脉冲的下降沿并将 ADCR 寄存器的 ADST 位置 1。此后的运行与通过软件给 ADST 位写 1 时的运行相同。但是, 由外部触发输入控制的 A/D 转换开始功能只在 ADST 位被清 0 时有效。 另外, 在使用外部触发开始功能时, $\overline{\text{ADTRG}}$ 引脚的输入低电平脉宽必须至少为 1.5 个 P_{ϕ} 时钟。 0: 通过 MTU2/MTU2S 的 A/D 转换开始触发启动 A/D 转换器 1: 通过外部引脚 ($\overline{\text{ADTRG}}$) 启动 A/D 转换器

17.3.2 A/D 状态寄存器 _0、_1 (ADSR_0、ADSR_1)

ADSR 是 8 位可读写寄存器, 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, 只能在读 1 后写 0。当标志为 0 时, 不能重写 0。

位	位名	初始值	R/W	说明
7 ~ 1	—	全 0	R	保留位 读写值总是 0。
0	ADF	0	R/(W)*	A/D 结束标志 是表示 A/D 转换结束的状态标志。 [置位条件] - 当在扫描模式中所选通道的 A/D 转换全部结束时 [清除条件] - 当读到 1 的状态后写 0 时 - 当通过 ADI 中断启动 DTC 并读 ADDR 时

17.3.3 A/D 开始触发选择寄存器 _0、_1 (ADSTRGR_0、ADSTRGR_1)

ADSTRGR 是在 ADCR 寄存器的 TRGE 位置 1、ADCR 寄存器的 EXTRG 位置 0 时，选择 MTU2 和 MTU2S 的 A/D 转换开始触发作为 A/D 转换开始源的寄存器。

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	初始值	R/W	说明
7	—	0	R	保留位 读写值总是 0。
6	STR6	0	R/W	开始触发 6 允许或者禁止从 MTU2S 输入的 A/D 转换开始请求。 0: 禁止由 TRGAN 触发 (MTU2S) 控制的 A/D 转换开始 1: 允许由 TRGAN 触发 (MTU2S) 控制的 A/D 转换开始
5	STR5	0	R/W	开始触发 5 允许或者禁止从 MTU2S 输入的 A/D 转换开始请求。 0: 禁止由 TRG4AN 触发 (MTU2S) 控制的 A/D 转换开始 1: 允许由 TRG4AN 触发 (MTU2S) 控制的 A/D 转换开始
4	STR4	0	R/W	开始触发 4 允许或者禁止从 MTU2S 输入的 A/D 转换开始请求。 0: 禁止由 TRG4BN 触发 (MTU2S) 控制的 A/D 转换开始 1: 允许由 TRG4BN 触发 (MTU2S) 控制的 A/D 转换开始
3	STR3	0	R/W	开始触发 3 允许或者禁止从 MTU2 输入的 A/D 转换开始请求。 0: 禁止由 TRG0N 触发 (MTU2) 控制的 A/D 转换开始 1: 允许由 TRG0N 触发 (MTU2) 控制的 A/D 转换开始
2	STR2	0	R/W	开始触发 2 允许或者禁止从 MTU2 输入的 A/D 转换开始请求 0: 禁止由 TRGAN 触发 (MTU2) 控制的 A/D 转换开始 1: 允许由 TRGAN 触发 (MTU2) 控制的 A/D 转换开始
1	STR1	0	R/W	开始触发 1 允许或者禁止从 MTU2 输入的 A/D 转换开始请求。 0: 禁止由 TRG4AN 触发 (MTU2) 控制的 A/D 转换开始 1: 允许由 TRG4AN 触发 (MTU2) 控制的 A/D 转换开始
0	STR0	0	R/W	开始触发 0 允许或者禁止 MTU2 输入的 A/D 转换开始请求 0: 禁止由 TRG4BN 触发 (MTU2) 控制的 A/D 转换开始 1: 允许由 TRG4BN 触发 (MTU2) 控制的 A/D 转换开始

17.3.4 A/D 模拟输入通道选择寄存器 _0、_1（ADANSR_0、ADANSR_1）

ADANSR 是 8 位可读写寄存器，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	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置 1，选择与设定位对应的通道。模拟输入引脚与位的对应请参照表 17.3。 为了防止误动作，必须在 ADCR 寄存器的 ADST 位为 0 的状态下切换模拟输入通道。
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 17.3 通道选择一览表

位名	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

17.3.5 A/D 数据寄存器 0 ~ 15（ADDR0 ~ ADDR15）

ADDR 是保存 A/D 转换结果的 16 位只读寄存器，各模拟输入通道的转换结果保存到对应序号的 ADDR（参照表 17.4）。

12 位转换数据保存在 ADDR 的 bit11 ~ bit0。

ADDR 的初始值为 H'0000。

另外，通过将 ADCR 寄存器的 ACE 位置 1，能在读 ADDR 寄存器后自动将 ADDR 寄存器清 0。

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	初始值	R/W	说明
15 ~ 12	—	全 0	R	保留位
11 ~ 0	ADD[11:0]	全 0	R	位数据（12 位）

表 17.4 模拟输入通道与 ADDR0 ~ ADDR15 寄存器的对应

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

17.3.6 和 CPU 的接口

因为连接 CPU 的芯片内部总线是 16 位宽，所以不需要分别读取高位字节 / 低位字节的数据。

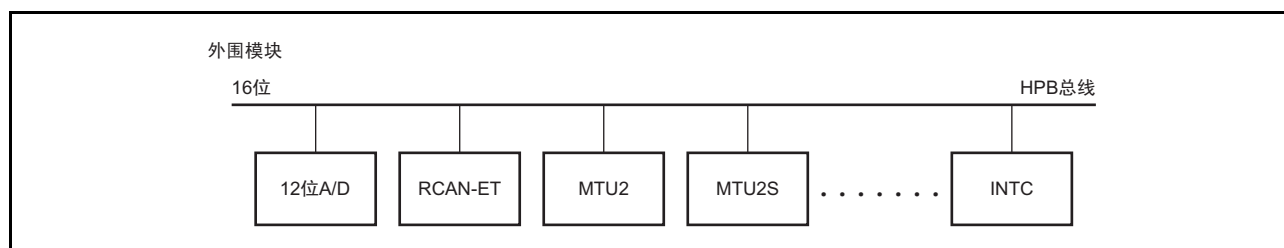


图 17.2 CPU 和 12 位 A/D 的接口

17.4 运行说明

A/D 转换的运行模式有单周期扫描模式和连续扫描模式 2 种。单周期扫描模式是在进行 1 次至少 1 个通道的 A/D 转换就结束的模式。连续扫描模式是在 ADST 位被清 0 前重复进行指定的至少 1 个通道的 A/D 转换的模式。

使用 A/D 控制寄存器 (ADCR) 的 ADCS 位选择运行模式。要设定为单周期扫描模式时，将 ADCS 位置 0；要设定为连续扫描模式时，将 ADCS 位置 1。如果开始转换，就根据 A/D 模拟输入通道选择寄存器 (ADANSR) 所选的模拟输入通道号从小到大 (A/D_0 为 AN0→AN7、A/D_1 为 AN8→AN15) 进行 A/D 转换。

在单周期扫描模式中，如果设定的全部通道 1 次转换结束，就将 ADSR 的 ADF 位置 1，而 ADST 位自动清 0。在连续扫描模式中，如果设定的全部通道转换结束，就将 ADSR 的 ADF 位置 1。在停止 A/D 转换时，必须将 ADST 位置 0。如果在将 ADF 位置 1 时 ADCR 的 ADIE 位已被置 1，就产生 A/D 转换结束中断 (ADI)。在将 ADF 位清 0 时，必须在读到 ADF 位的 1 后写 0。但是，在通过 ADI 中断启动 DTC 的情况下，ADF 位自动清 0。

17.4.1 单周期扫描模式

选择模拟输入通道 0 ~ 3 (AN0 ~ AN3)，在 4 个通道中对 A/D_0 转换进行单周期扫描的运行例子如下所示。此例也适用于 A/D_1 的 A/D 转换。

1. 将 A/D 控制寄存器_0 (ADCR_0) 的 ADCS 位置 0。
2. 将 A/D 模拟输入通道选择寄存器_0 (ADANSR_0) 的 ANS0 ~ ANS3 位全部置 1。

3. 将A/D控制寄存器_0 (ADCR_0)的ADST位置1, 开始A/D转换。
4. 在同时对通道0~通道2 (GrA)进行采样后, 进行偏移取消处理 (OFC)。然后, 开始通道0的A/D转换, 在A/D转换结束时将A/D转换结果传送到ADDR0寄存器。接着开始通道1的转换, 在A/D转换结束时将A/D转换结果传送到ADDR1寄存器。然后, 同样地进行通道2和通道3的转换, 将A/D转换结果分别传送到ADDR2寄存器和ADDR3寄存器。
5. 当设定的全部通道 (AN0~AN3)转换结束时, 就将ADF位置1, 而ADST位自动清0, 停止A/D转换。此时, 如果ADIE位已被置1, 就在A/D转换结束后产生ADI_3中断。

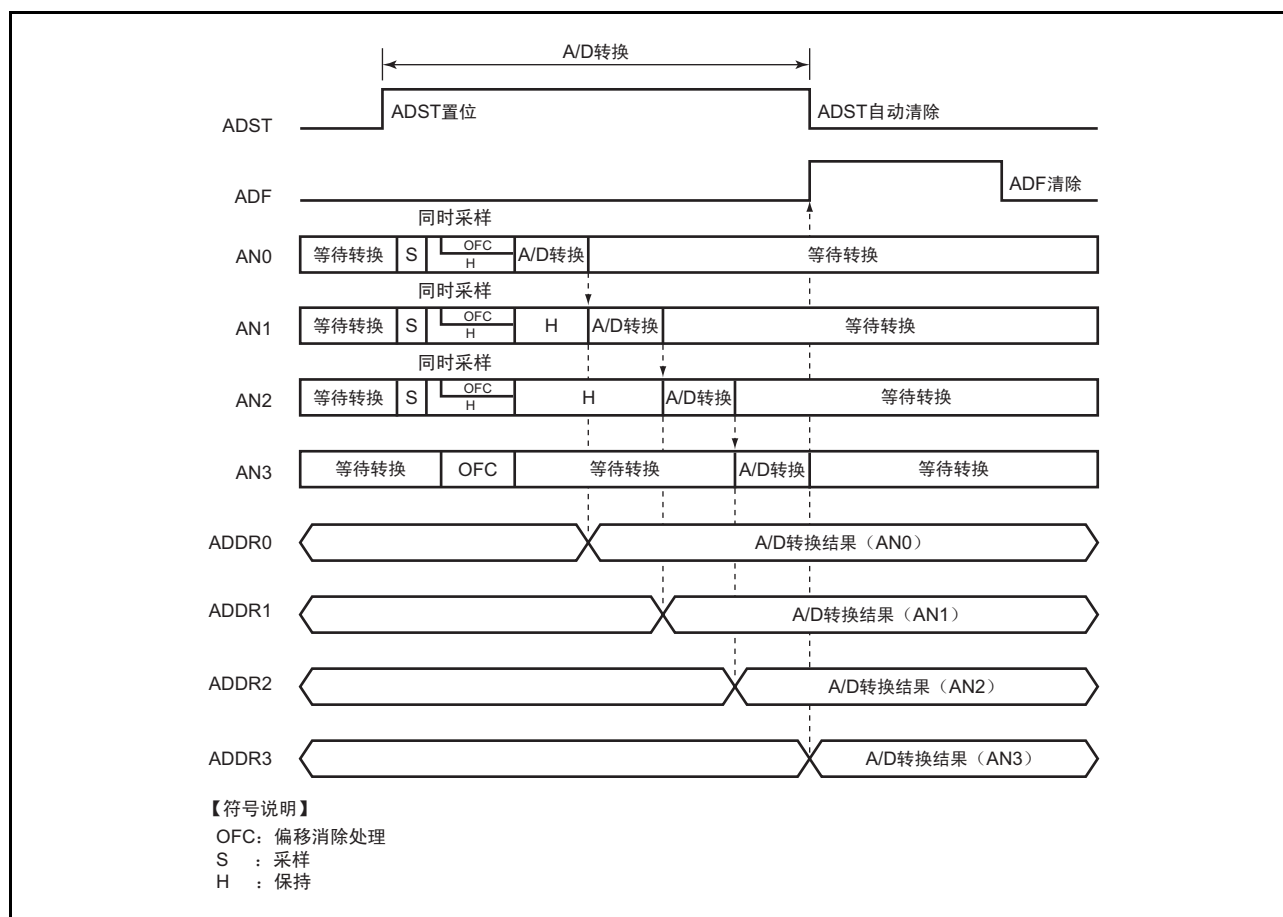


图 17.3 A/D_0 转换器的运行例 (单周期扫描模式)

17.4.2 连续扫描模式

选择模拟输入通道 0、2、3 (AN0、AN2 和 AN3), 在 3 个通道中对 A/D_0 转换进行连续扫描的运行例子如下所示。此例也适用于 A/D_1 的 A/D 转换。

1. 将A/D控制寄存器_0 (ADCR_0)的ADCS位置1。
2. 将A/D模拟输入通道选择寄存器_0 (ADANSR_0)的ANS0、ANS2和ANS3位全部置1。
3. 将A/D控制寄存器_0 (ADCR_0)的ADST位置1, 开始A/D转换。
4. 同时对通道0、2 (GrA)进行采样。此时, 因为ADANSR_0寄存器的ANS1位已被置0, 所以不对通道1进行采样。在采样后进行偏移取消处理 (OFC)。然后, 开始通道0的A/D转换, 在A/D转换结束时将A/D转换结果传送到ADDR0寄存器。接着开始通道2的转换, 在A/D转换结束时将A/D转换结果传送到ADDR2寄存器。不对通道1进行A/D转换。
5. 然后, 开始通道3的A/D转换, 在A/D转换结束时将A/D转换结果传送到ADDR3寄存器。

- 当设定的全部通道 (AN0、AN2、AN3) 转换结束时, 就将 ADF 位置 1。此时, 如果 ADIE 位已被置 1, 就在 A/D 转换结束后产生 ADI_3 中断。
- 在 ADST 位被置 1 期间, 重复 4.~6.。如果将 ADST 位清 0, 就停止 A/D 转换。之后, 如果将 ADST 位置 1, 就重新开始 A/D 转换, 重复 4.~6.。

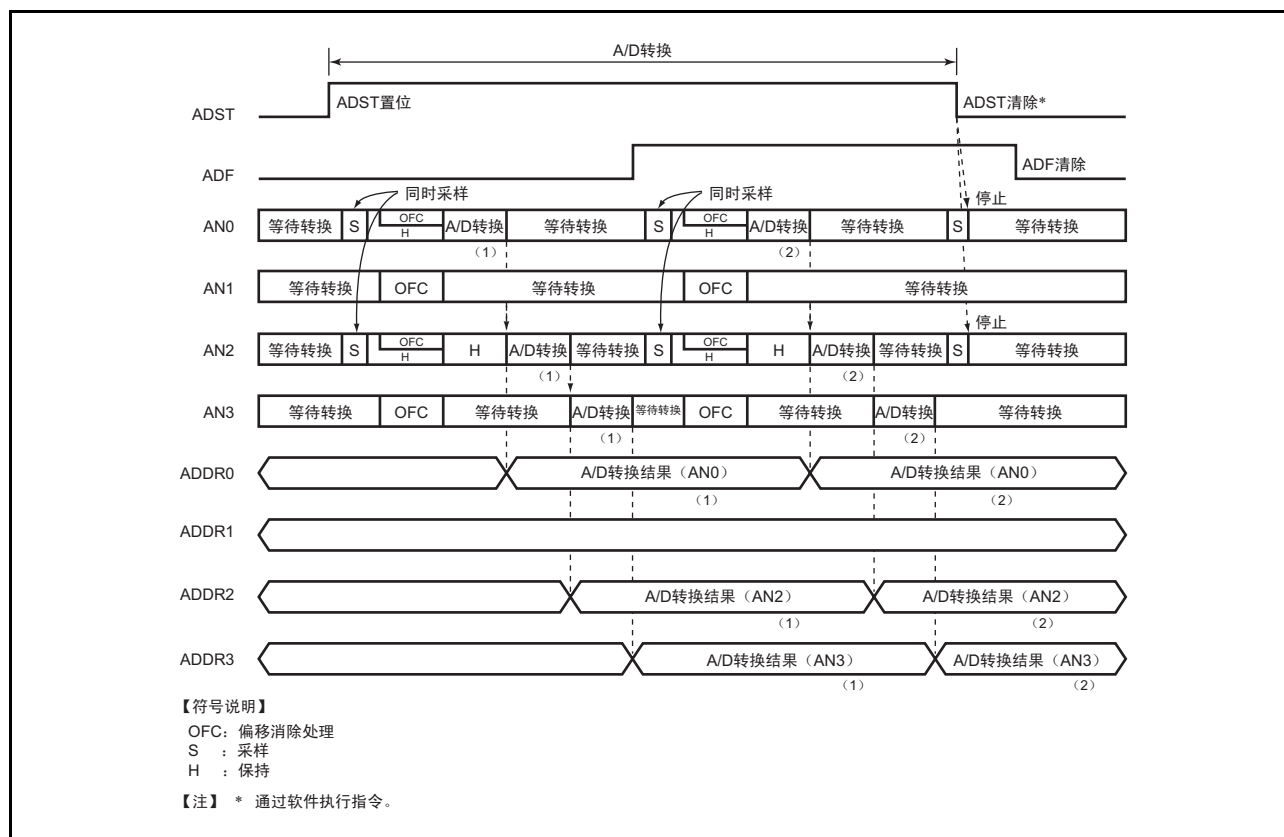


图 17.4 A/D_0 转换器的运行例 (连续扫描模式)

17.4.3 输入采样和 A/D 转换时间

A/D_0 内置了各通道共通的采样和保持电路, A/D_0 的通道 0 ~ 2 内置了各通道专用的采样和保持电路。A/D_0 的通道 0 ~ 2 能作为一组同时进行采样。此组称为 A 组 (GrA) (表 17.5)。即使通过 ADANSR 寄存器只选择组内的 1 个通道, 也能使用专用的采样和保持电路进行采样和保持运行。而且, 即使通过 ADANSR 寄存器只设定没有专用采样和保持电路的通道, 也要经过与使用专用采样和保持电路时相同的时间。

上述的内容也适用于 A/D_1。

如果发生 ADST 置位源 (由 CPU 产生的写操作、由 MTU2 和 MTU2S 产生的 A/D 启动请求和外部触发等), 就在经过 A/D 转换开始延迟时间 (t_D) 后, 通过各通道专用的采样和保持电路进行模拟输入的采样, 然后进行偏移取消处理 (OFC)。接着在通过各通道通用的采样和保持电路进行模拟输入的采样后, 开始 A/D 转换。此时的 A/D 转换时序如图 17.5 所示, 此时的 A/D 转换时间 (t_{CONV}) 包括 t_D 、偏移取消处理时间 (t_{OFC})、各通道专用的采样和保持电路进行的模拟输入采样时间 (t_{SPLSH})、各通道共通的采样和保持电路进行的模拟输入采样时间 (t_{SPL})。但是, t_{SPLSH} 与同时采样的通道数无关。

在连续扫描模式中, 表 17.6 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 个周期以后的全部通道转换结束前的转换时间表示为 $t_{CONV} - t_D + 6$ 。

表 17.5 模拟输入通道与能同时采样的组的对应

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 17.6 A/D 转换时间

项目	符号	所需状态数		
		Min.	Typ.	Max.
A/D 转换开始的延迟时间	t_D	11*1	—	15*2
GrA/GrB 专用的采样和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	—
各通道共通的采样和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 是通过 MTU2 和 MTU2S 的触发信号启动 A/D 的情况。

*2 是通过外部触发信号启动 A/D 的情况。

*3 n 为转换通道数 (n=1 ~ 8)。

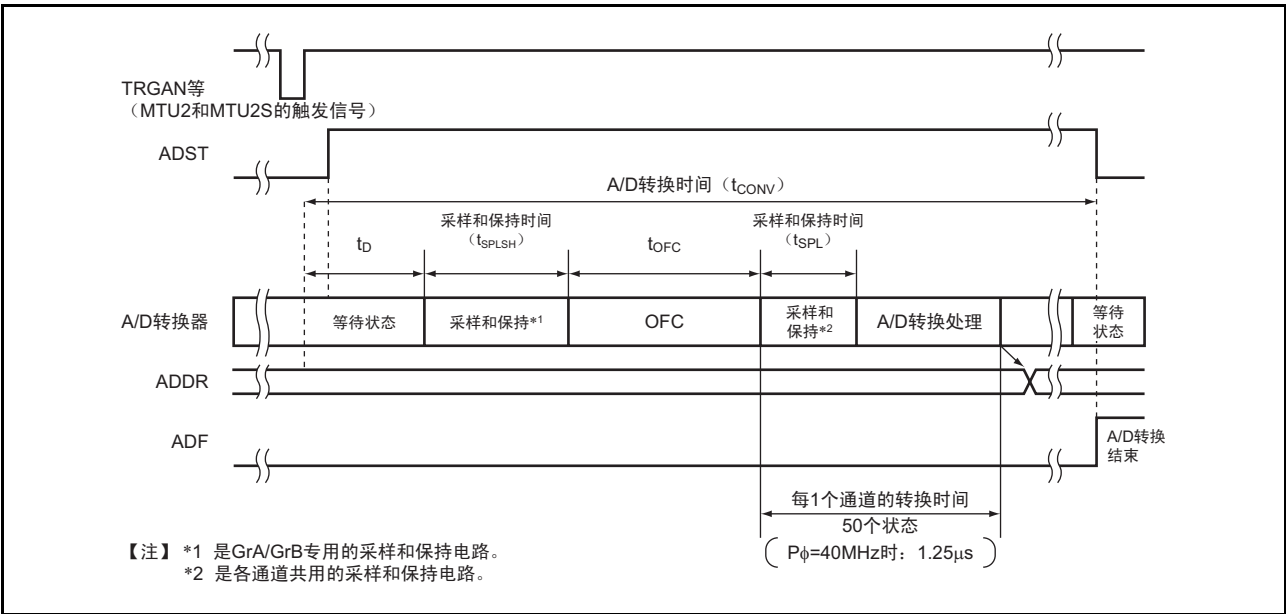


图 17.5 A/D 转换时序 (单周期扫描模式)

17.4.4 通过 MTU2 和 MTU2S 启动 A/D 转换器

能通过 MTU2 的 A/D 转换开始触发 (TRGAN、TRG0N、TRG4AN、TRG4BN) 和 MTU2S 的 A/D 转换开始触发 (TRGAN、TRG4AN、TRG4BN) 启动 A/D 转换。为了要使此功能有效, 首先将 ADCR 寄存器的 TRGE 位置 1, 并将 EXTRG 位清 0。如果在此状态下发生 ADSTRGR 寄存器设置的 MTU2 和 MTU2S 的 A/D 转换开始触发, ADST 位就被置 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。

必须在设定 ADCR 寄存器、ADSTRGR 寄存器和 ADANSR 寄存器后输入 A/D 转换开始触发信号。

17.4.5 外部触发输入时序

也通过外部触发输入开始 A/D 转换。为了进行外部触发输入, 必须通过引脚功能控制器 (PFC) 将引脚功能设定为 $\overline{\text{ADTRG}}$, 给 $\overline{\text{ADTRG}}$ 引脚输入高电平, 并在将 ADCR 寄存器的 TRGE 位和 EXTRG 位置 1 的状态下, 从 $\overline{\text{ADTRG}}$ 引脚输入低电平。在 $\overline{\text{ADTRG}}$ 的下降沿, 将 ADCR 的 ADST 位置 1, 开始 A/D 转换。其他运行与 A/D 转换启动源无关。此时序如图 17.6 所示。

ADST 位置 1 的时序是 A/D 转换器检测到 $\overline{\text{ADTRG}}$ 引脚下降沿后的 5 个状态。

必须在设定 ADCR 寄存器、ADSTRGR 寄存器和 ADANSR 寄存器后给 $\overline{\text{ADTRG}}$ 引脚输入低电平。

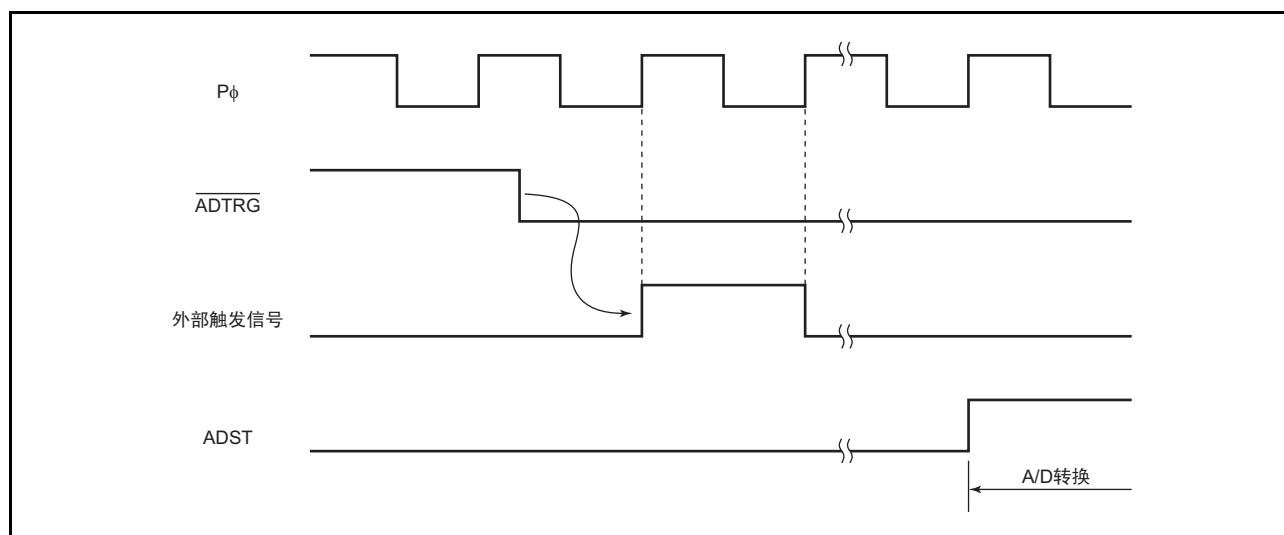


图 17.6 外部触发输入时序

17.4.6 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, 能在 CPU 和 DTC 读 A/D 数据寄存器 (ADDR) 后自动将 ADDR 寄存器清 0。通过使用此功能, 能检测到 ADDR 寄存器未更新的故障。

ADDR 寄存器自动清除功能无效 (通常状态) / 有效时的例子如图 17.7 所示。

在 ACE 位为 0 (初始值) 时, 如果因某种原因而没有将 A/D 转换结果 (H'0222) 写到 ADDR 寄存器, ADDR 寄存器的值就为旧数据 (H'0111)。此时, 如果利用 A/D 转换结束中断将 ADDR 寄存器的值读到通用寄存器, 旧数据 (H'0111) 就被保存到通用寄存器。在进行未更新检测时, 需要边将旧数据保存到 RAM 或者通用寄存器等边进行检测。

在 ACE 位为 1 时, 如果通过 CPU/DTC 读 ADDR=H'0111, ADDR 寄存器就自动清 0。此后, 如果因某种原因而没有将 A/D 转换结果 (H'0222) 写到 ADDR 寄存器, 被清除的数据 (H'0000) 就保留在 ADDR 寄存器。此时, 如果利用 A/D 转换结束中断将 ADDR 寄存器的值读到通用寄存器, H'0000 就被保存到通用寄存器。所以只要检查读取的数据值为 H'0000, 就能判断 ADDR 寄存器未更新的故障。

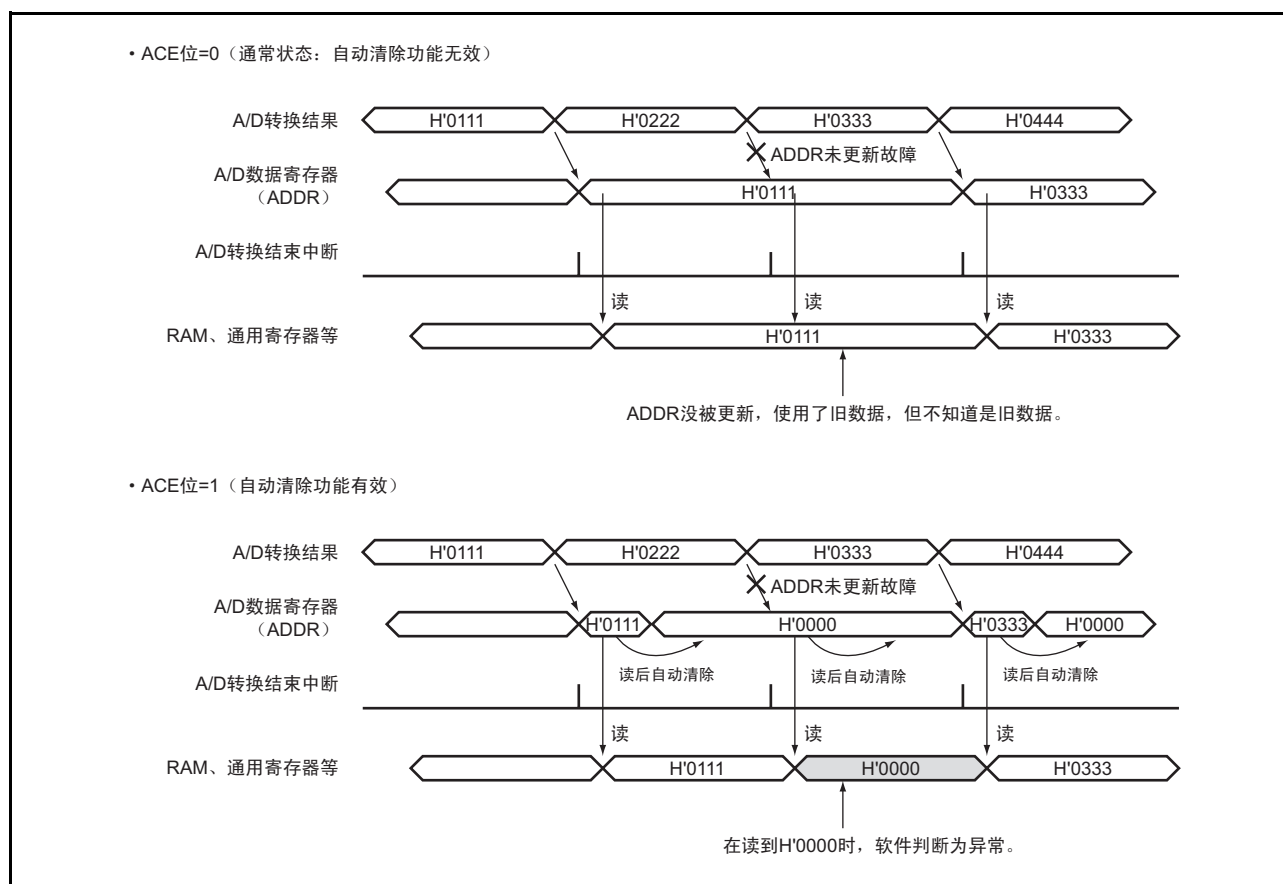


图 17.7 ADDR 寄存器的自动清除功能无效 (通常状态) / 有效时的例子

17.5 中断源和 DTC 传送请求

A/D 转换器能产生 A/D 转换结束中断 (ADI_3 和 ADI_4)。如果将 ADCR_0 寄存器的 ADIE 位置 1, 就允许 ADI_3 中断的产生; 如果将 ADCR_1 寄存器的 ADIE 位置 1, 就允许 ADI_4 中断的产生。反之, 如果将 ADCR_0 寄存器的 ADIE 位清 0, 就禁止 ADI_3 中断的产生; 如果将 ADCR_1 寄存器的 ADIE 位清 0, 就禁止 ADI_4 中断的产生。另外, 通过 DTC 的设定, 能在发生 ADI_3 中断或者 ADI_4 中断时启动 DTC。如果通过 ADI_3 中断或者 ADI_4 中断启动 DTC, 就在通过 DTC 进行数据传送时自动清除 ADSR_0 和 ADSR_1 的 ADF 位。

17.6 A/D 转换精度的定义

本 LSI 的 A/D 转换精度定义如下:

- 分辨率
A/D 转换器的数字转换输出码的位数。
- 偏移误差
是在数字输出值从最小值 (零电压) B'000000000000 变为 B'000000000001 时, 实际 A/D 转换特性和理想 A/D 转换特性的偏差。但是, 不包含量化误差 (图 17.8)。
- 满刻度误差
是在数字输出值从 B'111111111110 变为最大值 (满刻度电压) B'111111111111 时, 实际 A/D 转换特性和理想 A/D 转换特性的偏差。但是, 不包含量化误差 (图 17.8)。

- 量化误差
是A/D转换器固有的偏差，为1/2LSB（图17.8）。
- 非线性误差
是在零电压和满刻度电压之间，实际A/D转换特性和理想A/D转换特性的偏差，但是，不包含偏移误差、满刻度误差和量化误差（图17.8）。
- 绝对精度
是数字值和模拟输入值的偏差，包含偏移误差、满刻度误差、量化误差和非线性误差。

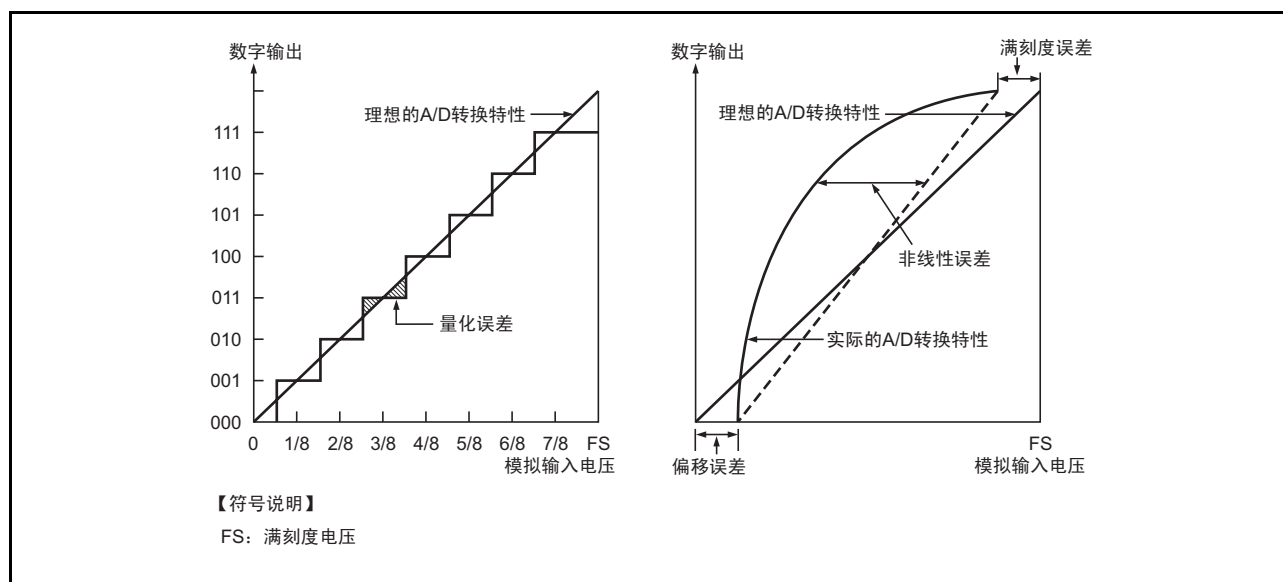


图 17.8 A/D 转换精度的定义

17.7 使用时的注意事项

17.7.1 模拟输入电压的设定范围

在 A/D 转换时，模拟输入引脚（ANn）的外加电压必须在 $AV_{refl} \leq ANn \ (n=0 \sim 15) \leq AV_{refh}$ 的范围内。

17.7.2 AV_{CC} 、 AV_{SS} 与 V_{CC} 、 V_{SS} 的关系

在使用 A/D 转换器时，必须 $AV_{CC}=5.0V \pm 0.5V$ 且 $AV_{SS}=V_{SS}$ ；在不使用 A/D 转换器时，必须 $AV_{SS}=V_{SS}$ ，并且不能将 AV_{CC} 引脚置为开路。

17.7.3 AV_{refh} 和 AV_{refl} 引脚的设定范围

在使用 A/D 转换器时，必须 $AV_{refh}=4.5 \sim AV_{CC}$ ；在不使用 A/D 转换器时，必须 $AV_{refh} \leq AV_{CC}$ 。否则会给 LSI 的可靠性带来不良影响。必须 $AV_{refl}=AV_{SS}=V_{SS}$ 。

17.7.4 电路板设计的注意事项

在设定电路板时，必须尽量将数字电路和模拟电路分开，不能使数字电路的信号线和模拟电路的信号线交叉或者靠近。否则，会因电感等引起模拟电路的误动作并给 A/D 转换值带来不良影响。

模拟输入信号（AN0～AN15）、模拟基准电压（ AV_{refh} 、 AV_{refl} ）、模拟电源（ AV_{CC} ）和模拟接地（ AV_{SS} ）必须与数字电路分开，并且 AV_{SS} 必须单点连接到电路板上稳定的数字接地（ V_{SS} ）。

17.7.5 噪声对策的注意事项

如图 17.9 所示, 为了防止因过大电涌等异常电压引起对模拟输入引脚 (AN0 ~ AN15) 和模拟基准电压 (AV_{refh} 、 AV_{refl}) 的破坏, 必须在 $AV_{CC}-AV_{SS}$ 之间连接保护电路。将连接 AV_{refh} 和 AV_{refl} 的旁路电容及连接 ANn 的过滤器电容连接到 AV_{SS} 。另外, 如图 17.9 所示, 如果连接过滤器电容, 模拟输入引脚 (ANn) 的输入电流就被平均, 因此有可能产生误差。因此, 在决定电路常数时, 必须进行充分的探讨。

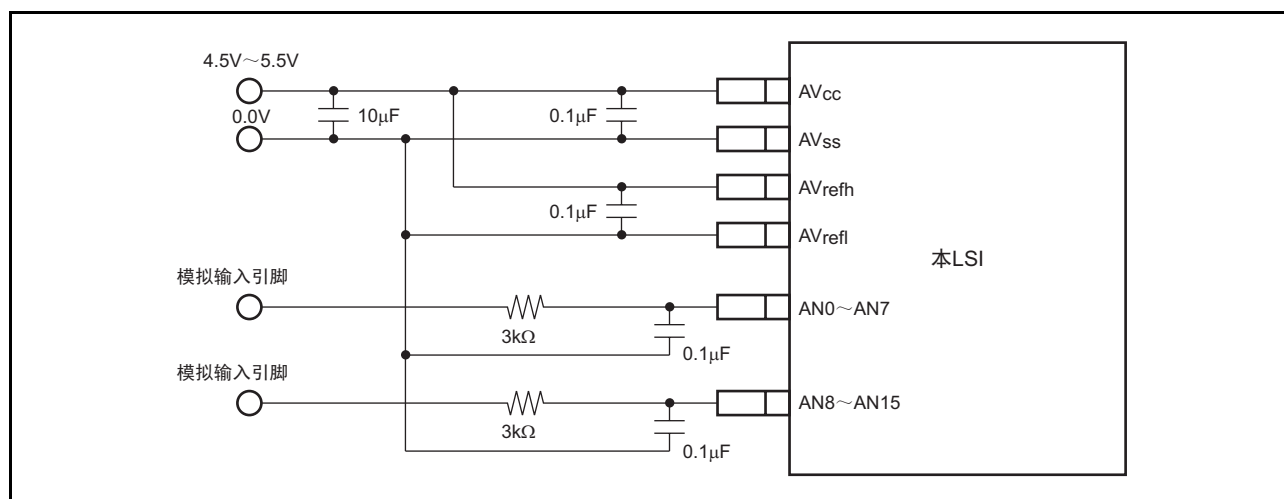


图 17.9 模拟输入引脚的保护电路例子

17.7.6 寄存器设定时的注意事项

- 必须在 A/D 开始触发选择寄存器 (ADSTRGR) 和 A/D 模拟输入通道选择寄存器 (ADANSR) 的设定结束后, 将 A/D 控制寄存器 (ADCR) 的 ADST 位置位。而且, 不能在 ADCR 的 ADST 位为 1 的状态下更改 ADCR 的 ADCS 位、ACE 位、ADIE 位、TRGE 位和 EXTRG 位的设定。
- 不能在 A/D 模拟输入通道选择寄存器 (ADANSR) 的 ANS7 ~ ANS0 位全部为 0 时开始 A/D 转换。

第 18 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，能按各设定的周期产生中断。

18.1 特点

- 2 个通道能独立选择 4 种计数器输入时钟
能选择 4 种内部时钟 ($P\phi/8$ 、 $P\phi/32$ 、 $P\phi/128$ 、 $P\phi/512$)
- 能在比较匹配时请求中断
- 能设定模块待机模式

CMT 的框图如图 18.1 所示。

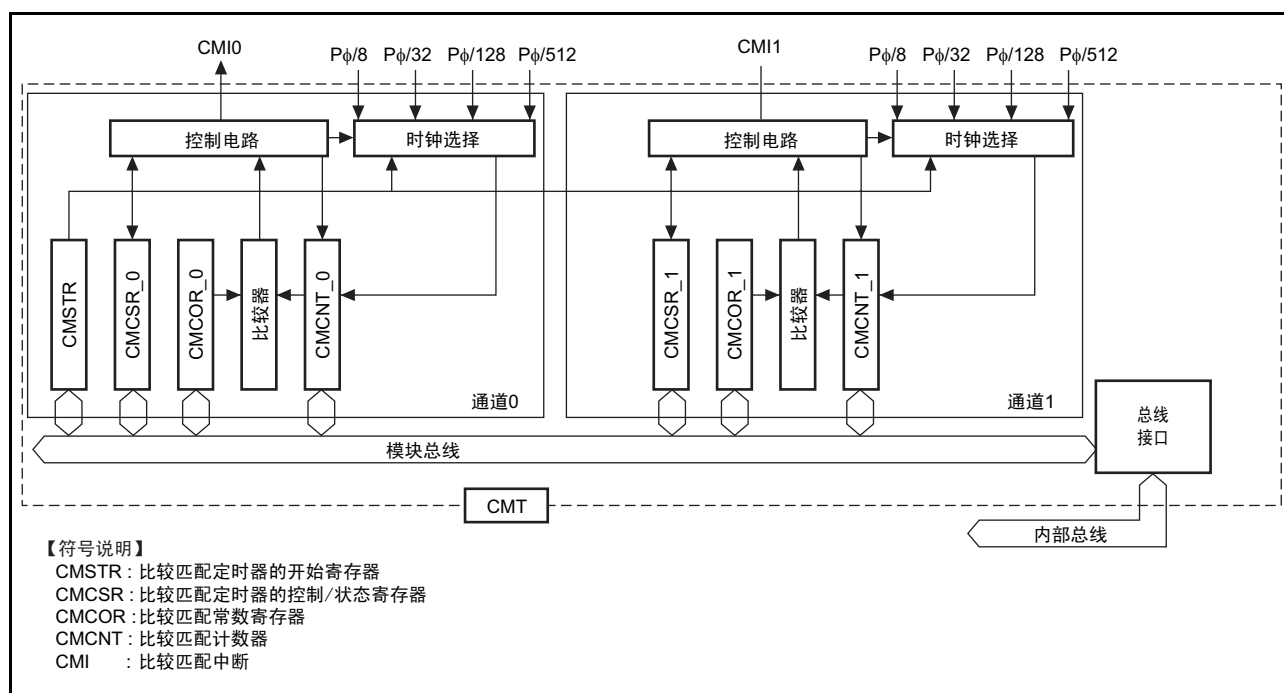


图 18.1 CMT 的框图

18.2 寄存器说明

CMT 有以下寄存器。有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 25 章 寄存器一览”。另外，本节中省略了通道号。

表 18.1 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
比较匹配定时器的开始寄存器	CMSTR	R/W	H'0000	H'FFFCE00	8、16、32
比较匹配定时器的控制 / 状态寄存器_0	CMCSR_0	R/W	H'0000	H'FFFCE02	8、16
比较匹配计数器_0	CMCNT_0	R/W	H'0000	H'FFFCE04	8、16、32
比较匹配常数寄存器_0	CMCOR_0	R/W	H'FFFF	H'FFFCE06	8、16
比较匹配定时器的控制 / 状态寄存器_1	CMCSR_1	R/W	H'0000	H'FFFCE08	8、16、32
比较匹配计数器_1	CMCNT_1	R/W	H'0000	H'FFFCE0A	8、16
比较匹配常数寄存器_1	CMCOR_1	R/W	H'FFFF	H'FFFCE0C	8、16、32

18.2.1 比较匹配定时器的开始寄存器（CMSTR）

CMSTR 是 16 位寄存器，选择比较匹配计数器（CMCNT）的运行 / 停止。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STR1	STR0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 2	—	全 0	R	保留位 读写值总是 0。
1	STR1	0	R/W	计数开始 1 选择比较匹配计数器_1 的运行 / 停止。 0: CMCNT_1 停止计数 1: CMCNT_1 开始计数
0	STR0	0	R/W	计数开始 0 选择比较匹配计数器_0 的运行 / 停止。 0: CMCNT_0 停止计数 1: CMCNT_0 开始计数

18.2.2 比较匹配定时器的控制 / 状态寄存器 (CMCSR)

CMCSR 是 16 位寄存器，表示比较匹配的产生，设定中断和计数器的输入时钟。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CMF	CMIE	—	—	—	—	CKS[1:0]	
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	(R/W)*1	R/W	R	R	R	R	R/W	R/W

【注】*1 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说明
15 ~ 8	—	全 0	R	保留位 读入值总是 0。
7	CMF	0	(R/W)*1	比较匹配标志 表示 CMCNT 和 CMCOR 的值是否相同。 0: CMCNT 和 CMCOR 的值不相同 [清除条件] • 当读到 CMF=1 后写 0 时 *2 • 在通过 CMI 中断启动 DTC 并且 DTC 的 MRB 的 DISEL 位为 0 时， 存取 CMT 寄存器 [置位条件] 1: CMCNT 和 CMCOR 的值相同
6	CMIE	0	R/W	比较匹配的中断允许 选择在 CMCNT 和 CMCOR 值相同时 (CMF=1) 是允许还是禁止比较匹配中断的产生 (CMI)。 0: 禁止比较匹配中断 (CMI) 1: 允许比较匹配中断 (CMI)
5 ~ 2	—	全 0	R	保留位 读写值总是 0。
1、0	CKS[1:0]	00	R/W	时钟选择 1、0 从分频外围运行时钟 (Pφ) 后的 4 种内部时钟中，选择 CMCNT 的输入时钟。当 CMSTR 的 STR 位被置 1 时，就通过 CKS1 位和 CKS0 位选择时钟，CMCNT 开始计数。 00: Pφ/8 01: Pφ/32 10: Pφ/128 11: Pφ/512

【注】*1 为了清除标志，只能在读 1 后写 0。

*2 如果在读 1 后写 0 前发生由下一个比较匹配引起的标志置位，即使写 0 也不能清除标志，所以必须重新读 1 后再写 0。

18.2.3 比较匹配计数器 (CMCNT)

CMCNT 是 16 位寄存器，用作递增计数器。如果通过 CMCSR 的 CKS1 位和 CKS0 位选择计数器的输入时钟，并将 CMSTR 的 STR 位置 1，CMCNT 就通过所选时钟开始计数。当 CMCNT 的值和比较匹配常数寄存器 (CMCOR) 的值相同时，CMCNT 就被清 0，并且 CMCSR 的 CMF 标志被置 1。

CMCNT 的初始值为 H'0000。

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

18.2.4 比较匹配常数寄存器 (CMCOR)

CMCOR 是 16 位寄存器，设定和 CMCNT 比较匹配前的时间，初始值为 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

18.3 运行说明

18.3.1 周期计数运行

如果通过 CMCSR 的 CKS1 位和 CKS0 位选择内部时钟，并将 CMSTR 的 STR 位置 1，CMCNT 就通过所选时钟开始递增计数。当 CMCNT 的值和 CMCOR 的值相同时，CMCNT 就被清 0，并且 CMCSR 的 CMF 标志被置 1。此时，如果 CMCSR 寄存器的 CMIE 位已被置 1，就请求比较匹配中断 (CMI)，CMCNT 从 H'0000 重新开始递增计数。

比较匹配计数器的运行如图 18.2 所示。

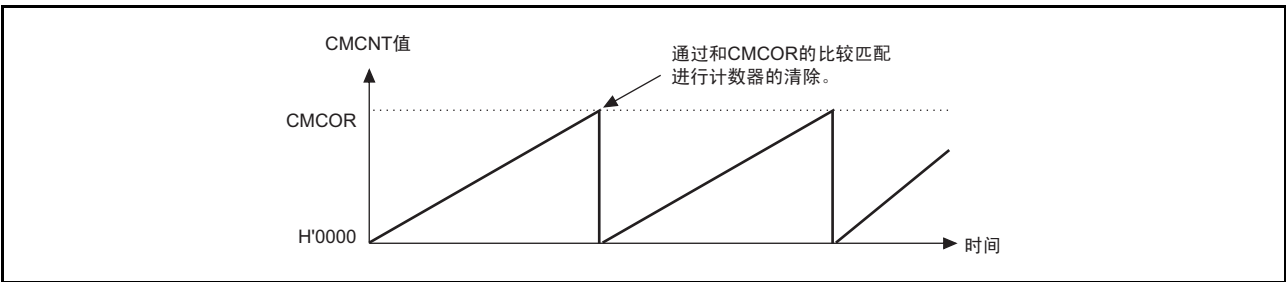


图 18.2 计数器的运行

18.3.2 CMCNT 的计数时序

能通过 CMCSR 的 CKS1 位和 CKS0 位，从分频时钟 (Pφ) 后得到的 4 种时钟 (Pφ/8、Pφ/32、Pφ/128、Pφ/512) 中选择一个时钟。此时序如图 18.3 所示。

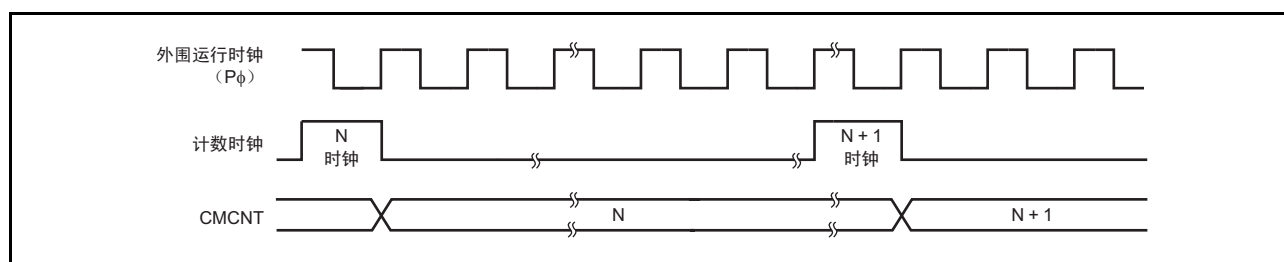


图 18.3 计数时序

18.4 中断

18.4.1 CMT 的中断源和 DTC

CMT 有各通道的比较匹配中断，分别分配了独立的向量地址。当中断请求标志 (CMF) 和中断允许位 (CMIE) 都被置 1 时，就输出对应的中断请求。在通过中断请求启动 CPU 中断的情况下，能通过设定中断控制器更改通道之间的优先级。详细内容请参照“第 6 章 中断控制器 (INTC)”。

另外，还能将中断请求作为数据传送控制器 (DTC) 的启动源。此时，通道间的优先级是固定的。详细内容请参照“第 8 章 数据传送控制器 (DTC)”。

18.4.2 比较匹配标志的置位时序

如果 CMCOR 和 CMCNT 相同，就产生比较匹配信号，并且 CMCSR 的 CMF 位被置 1。在相同的最后状态（将 CMCNT 的值更新为 H'0000 时）下产生比较匹配信号。即，如果在 CMCOR 和 CMCNT 相同后不输入用于 CMCNT 计数器的时钟，就不产生比较匹配信号。CMF 位的置位时序如图 18.4 所示。

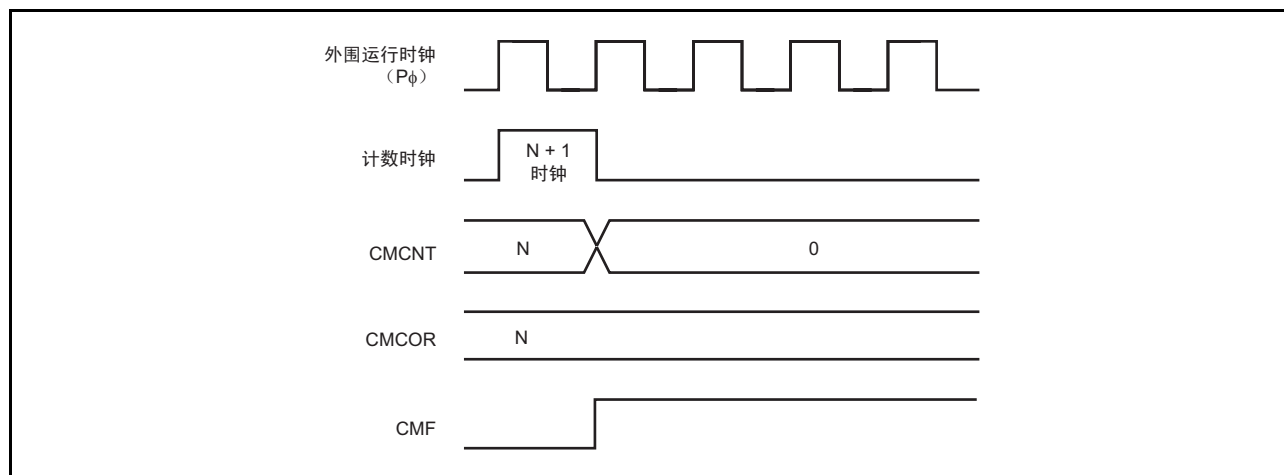


图 18.4 CMF 位的置位时序

18.4.3 比较匹配标志的清除时序

通过在读 CMF=1 后写 0，清除 CMCSR 的 CMF 位。

18.5 使用时的注意事项

18.5.1 模块待机模式的设定

CMT 能通过待机控制寄存器的设定允许或者禁止本模块的运行，初始值为 CMT 停止运行。能通过解除模块待机模式允许寄存器的存取。详细内容请参照“第 24 章 低功耗模式”。

18.5.2 CMCNT 的写操作和比较匹配的竞争

如果在 CMCNT 计数器写周期中的 T2 状态产生比较匹配信号，就不写 CMCNT 计数器而优先清除 CMCNT 计数器。此时序如图 18.5 所示。

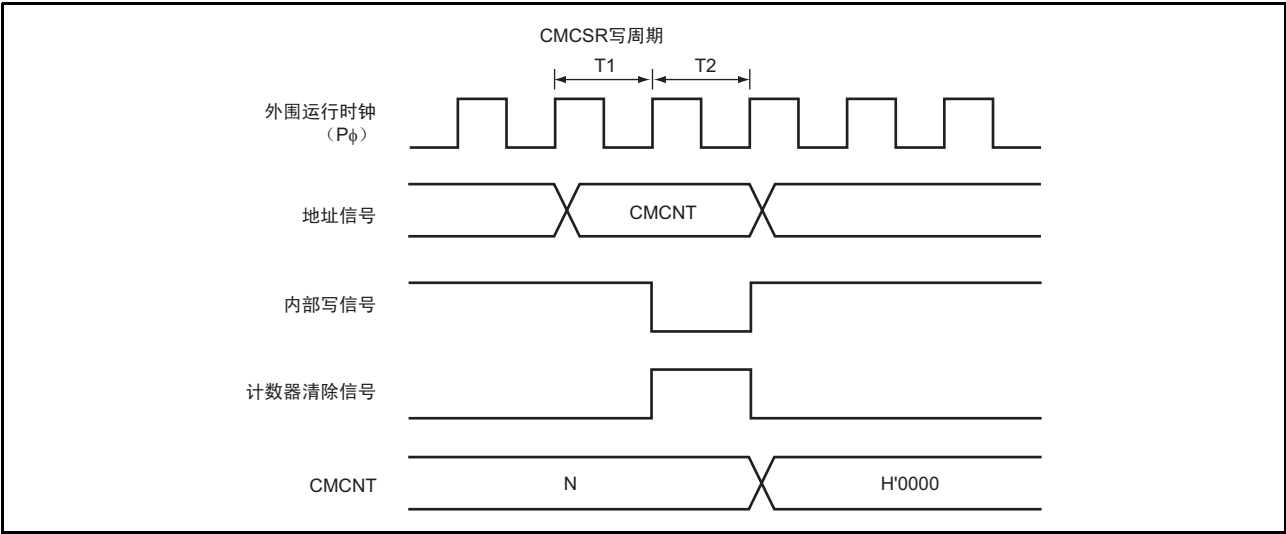


图 18.5 CMCNT 的写操作和比较匹配的竞争

18.5.3 CMCNT 的字写和递增计数的竞争

如果在 CMCNT 计数器字写周期中的 T2 状态产生递增计数信号，就不进行递增计数而优先写计数器。此时序如图 18.6 所示。

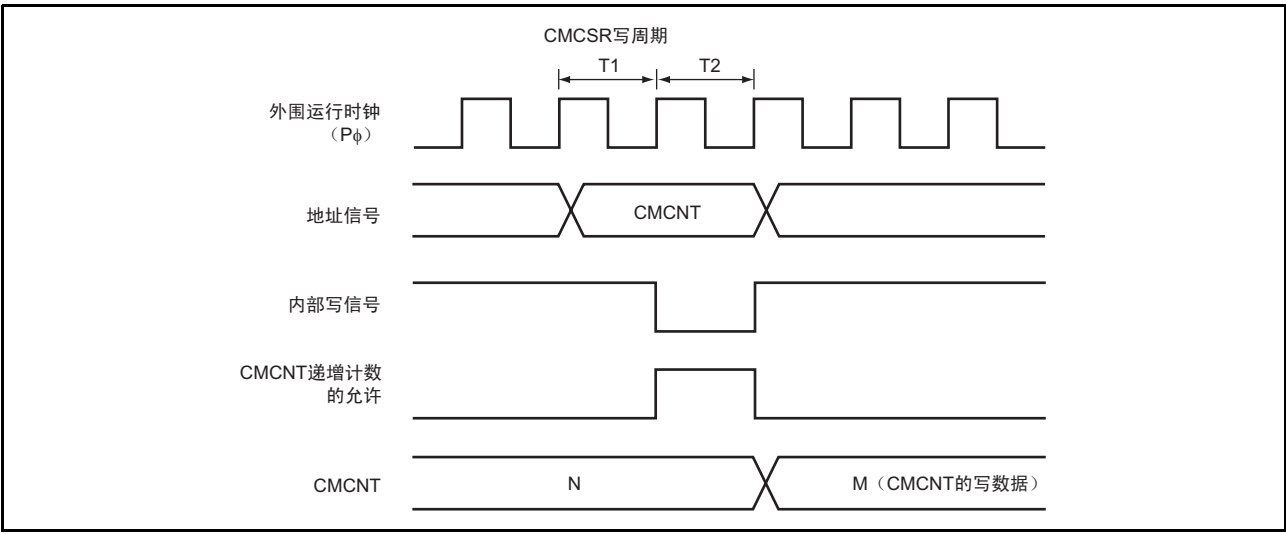


图 18.6 CMCNT 的字写和递增计数的竞争

18.5.4 CMCNT 的字节写和递增计数的竞争

如果在 CMCNT 字节写周期中的 T2 状态产生递增计数信号，写操作侧的数据就不进行递增计数而优先计数器的写操作。而且，不是写操作侧的字节数据也不进行递增计数而保持写操作之前的内容。

在 CMCNTH 写周期中的 T2 状态产生递增计数信号时的时序如图 18.7 所示。

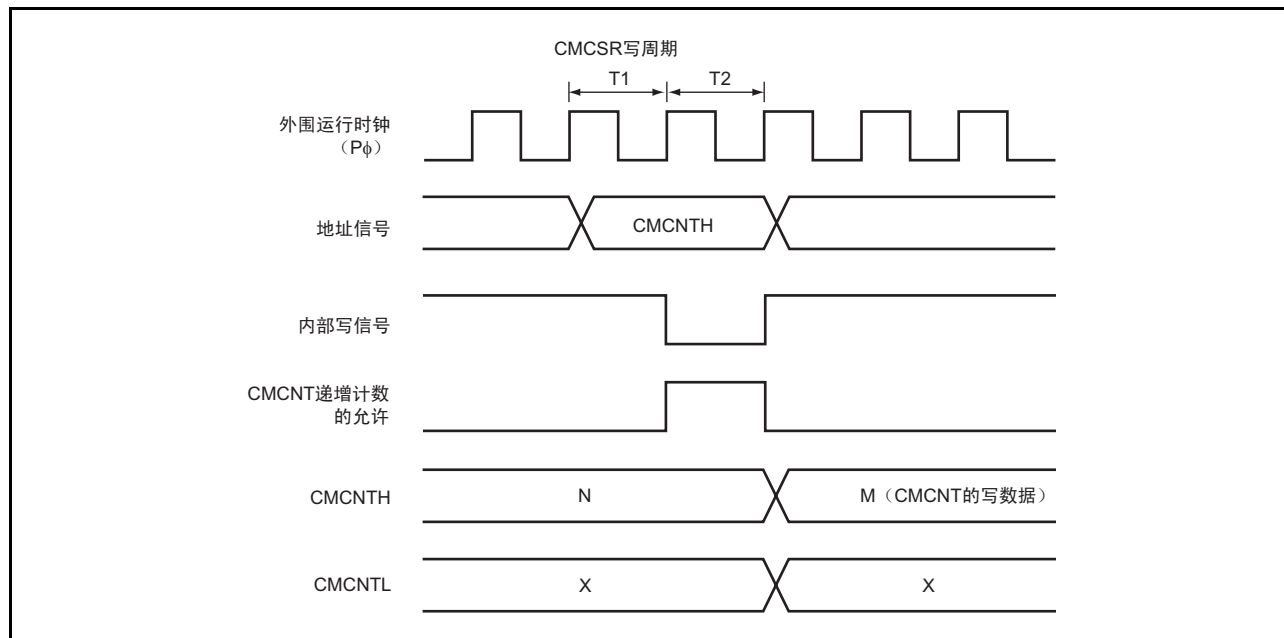


图 18.7 CMCNT 的字节写和递增计数的竞争

18.5.5 CMCNT 和 CMCOR 的比较匹配

不能在 CMCNT 停止计数的状态下给 CMCNT 和 CMCOR 设定相同的值。否则，CMCSR 的 CMF 位就被置 1，并且 CMCNT 被清 0。

第 19 章 控制器局域网 (RCAN-ET)

控制器局域网 (RCAN-ET) 模块用于控制以汽车和工业设备等系统的实时通信为目标的 CAN (Controller Area Network)。

本章说明 RCAN-ET 的程序接口。

有关 CAN 的数据链控制器的功能, 请参照下述参考文献。

[参考文献]

1. CAN Specification Version 2.0 part A, Robert Bosch GmbH, 1991
2. CAN Specification Version 2.0 part B, Robert Bosch GmbH, 1991
3. Implementation Guide for the CAN Protocol, CAN Specification 2.0 Addendum, CAN In Automation, Erlangen, Germany, 1997
4. Road vehicles-Controller area network(CAN): Part 1: Data link layer and physical signaling(ISO-CD-11898-1, 2002)

19.1 特点

- 对应CAN规格2.0B
- 位时序符合ISO-11898规格
- 16个邮箱
- 时钟频率: 16~40MHz
- 15个可编程的发送和接收邮箱和1个接收邮箱
- 低功耗的CAN睡眠模式, 在检测到CAN总线激活后自动解除CAN的睡眠模式
- 对应所有邮箱的可编程接收过滤屏蔽 (标准ID和扩展ID)
- 最大1Mbps的可编程CAN数据速率
- 为了防止实时应用引起的优先级颠倒, 准备了具有优先级内部区分功能的传送信息队列
- 丰富的中断源
- 内置测试功能 (只监听模式和错误被动模式)

19.2 结构

19.2.1 框图

RCAN-ET 提供灵活和简洁的方法进行 CAN 帧（支持 CAN2.0B Active 和 ISO-11898 的 CAN 帧）的构成和控制。RCAN-ET 由微处理器接口（MPI）、邮箱、邮箱控制和 CAN 接口 4 种功能块构成。

RCAN-ET 的框图如图 19.1 所示。

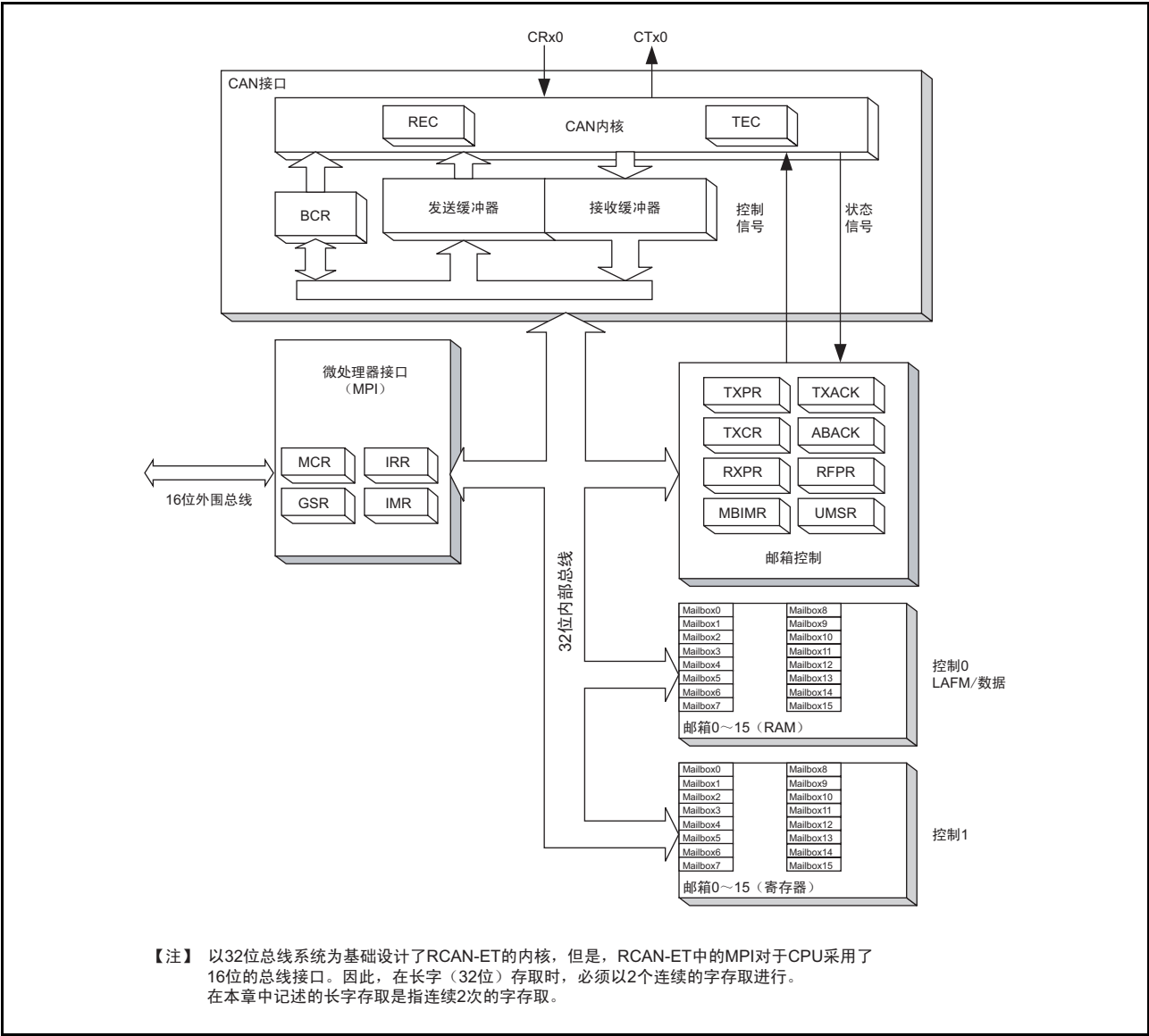


图 19.1 RCAN-ET 的框图

19.2.2 各块的功能

(1) 微处理器接口 (MPI)

MPI 能在 CPU 和 RCAN-ET 的寄存器、邮箱之间进行通信，并控制存储器接口。因为 MPI 具有唤醒控制逻辑，检测 CAN 总线激活，并将 CAN 总线的激活情况通知给 MPI 自身或者 RCAN-ET 其他模块，所以 RCAN-ET 能自动解除 CAN 睡眠模式。MPI 的寄存器有 MCR、IRR、GSR 和 IMR。

(2) 邮箱

将邮箱作为信息缓冲器配置到 RAM 和寄存器中。

RAM 和寄存器各有 16 个邮箱，保存以下信息：

[RAM]

- CAN 信息控制 (ID、RTR 和 IDE 等)
- CAN 信息数据 (用于 CAN 数据帧)
- 用于接收的局部接收过滤屏蔽 (LAFM)

[寄存器]

- CAN 信息控制 (DLC)
- 3 位邮箱结构、自动再发送无效位、用于远程请求的自动发送位和新信息控制位

(3) 邮箱控制

邮箱控制具有以下功能：

- 在接收信息时，比较 ID，生成 RAM 地址和数据 (用于将 CAN 接口的信息保存到邮箱)，并将对应的寄存器置位或者清除。
- 在发送信息时，RCAN-ET 启动内部仲裁选择优先级正确的信息，并从邮箱将信息加载到 CAN 接口的发送缓冲器。然后，将对应的寄存器置位或者清除。
- 在 CPU 和邮箱控制之间的邮箱存取进行仲裁。
- 寄存器有 TXPR、TXCR、TXACK、ABACK、RXPR、RFPR、MBIMR 和 UMSR。

(4) CAN 接口

本功能块支持参考文献 [2] 和 [4] 的 CAN 总线数据链控制器规格，具有 OSI 模型所规定的数据链控制器的全部功能。而且，也提供 CAN 总线专用的寄存器和逻辑。具体地说，有接收错误计数器、发送错误计数器、位时序结构寄存器和各种测试模式等，还具有保存 CAN 数据链控制器发送和接收数据的功能。

19.2.3 引脚结构

RCAN-ET 的引脚结构如表 19.1 所示。

表 19.1 引脚结构

名称	引脚名	输入 / 输出	功能
发送数据引脚	CTx0	输出	是 CAN 总线的发送引脚。
接收数据引脚	CRx0	输入	是 CAN 总线的接收引脚。

19.2.4 存储器映像

RCAN-ET 的存储器映像如图 19.2 所示。

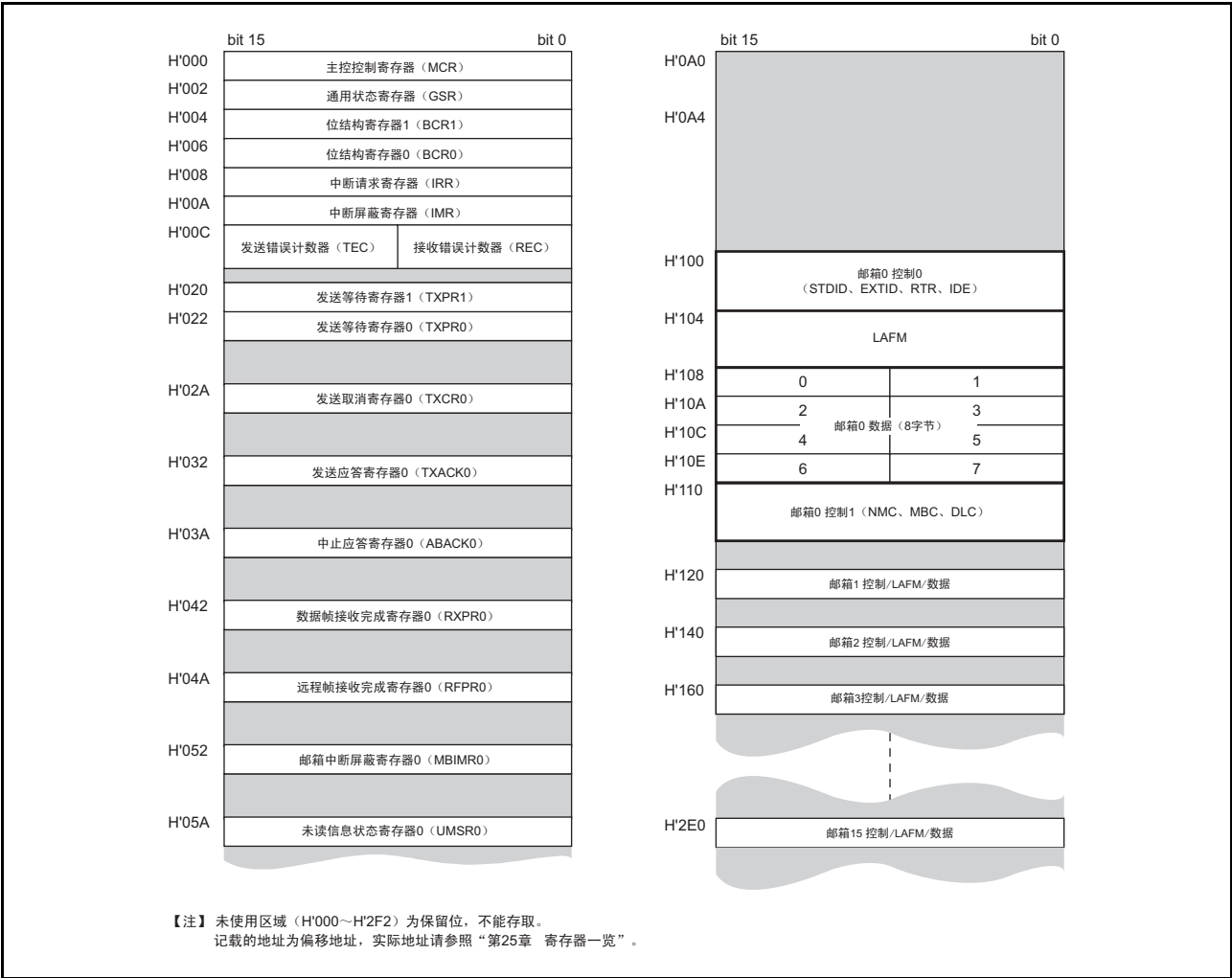


图 19.2 RCAN-ET 的存储器映像

19.3 邮箱

19.3.1 邮箱的结构

邮箱用作发送和接收 CAN 帧的信息缓冲器。各邮箱由信息控制、局部接收过滤屏蔽 (LAFM)、信息数据 3 个存储字段构成。

各邮箱的信息控制、LAFM、信息数据的地址映像如表 19.2 所示。

表 19.2 各邮箱的地址映像

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 (只限接收)	100 ~ 103	104 ~ 107	108 ~ 10F	110 ~ 111
1	120 ~ 123	124 ~ 127	128 ~ 12F	130 ~ 131
2	140 ~ 143	144 ~ 147	148 ~ 14F	150 ~ 151
3	160 ~ 163	164 ~ 167	168 ~ 16F	170 ~ 171
4	180 ~ 183	184 ~ 187	188 ~ 18F	190 ~ 191
5	1A0 ~ 1A3	1A4 ~ 1A7	1A8 ~ 1AF	1B0 ~ 1B1
6	1C0 ~ 1C3	1C4 ~ 1C7	1C8 ~ 1CF	1D0 ~ 1D1
7	1E0 ~ 1E3	1E4 ~ 1E7	1E8 ~ 1EF	1F0 ~ 1F1
8	200 ~ 203	204 ~ 207	208 ~ 20F	210 ~ 211
9	220 ~ 223	224 ~ 227	228 ~ 22F	230 ~ 231
10	240 ~ 243	244 ~ 247	248 ~ 24F	250 ~ 251
11	260 ~ 263	264 ~ 267	268 ~ 26F	270 ~ 271
12	280 ~ 283	284 ~ 287	288 ~ 28F	290 ~ 291
13	2A0 ~ 2A3	2A4 ~ 2A7	2A8 ~ 2AF	2B0 ~ 2B1
14	2C0 ~ 2C3	2C4 ~ 2C7	2C8 ~ 2CF	2D0 ~ 2D1
15	2E0 ~ 2E3	2E4 ~ 2E7	2E8 ~ 2EF	2F0 ~ 2F1

邮箱 0 是接收专用邮箱。邮箱 1 ~ 15 能通过信息控制的 MBC (邮箱结构) 位的设定, 进行发送和接收。邮箱结构的详细内容如图 19.3 所示。

• 邮箱0（接收邮箱）

寄存器名	地址	数据总线																存取长度	字段名
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
MB[0].CONTROL0H	H'100+N*32	IDE	RTR	0	STDID[10:0]										EXTID[17:16]			16/32位	控制0
MB[0].CONTROL0L	H'102+N*32	EXTID[15:0]																16位	
MB[0].LAFMH	H'104+N*32	IDE LAFM	0	0	STDID_LAFM[10:0]										EXTID_LAFM[17:16]			16/32位	LAFM
MB[0].LAFML	H'106+N*32	EXTID_LAFM[15:0]																16位	
MB[0].MSG_DATA[0][1]	H'108+N*32	MSG_DATA_0（最初的Rx/Tx字节）							MSG_DATA_1							8/16/32位	数据		
MB[0].MSG_DATA[2][3]	H'10A+N*32	MSG_DATA_2							MSG_DATA_3							8/16位			
MB[0].MSG_DATA[4][5]	H'10C+N*32	MSG_DATA_4							MSG_DATA_5							8/16/32位			
MB[0].MSG_DATA[6][7]	H'10E+N*32	MSG_DATA_6							MSG_DATA_7							8/16位			
MB[0].CONTROL1H、L	H'110+N*32	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（接收邮箱）

寄存器名	地址	数据总线																存取长度	字段名
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
MB[x].CONTROL0H	H'100+N*32	IDE	RTR	0	STDID[10:0]										EXTID[17:16]			16/32位	控制0
MB[x].CONTROL0L	H'102+N*32	EXTID[15:0]																16位	
MB[x].LAFMH	H'104+N*32	IDE LAFM	0	0	STDID_LAFM[10:0]										EXTID_LAFM[17:16]			16/32位	LAFM
MB[x].LAFML	H'106+N*32	EXTID_LAFM[15:0]																16位	
MB[x].MSG_DATA[0][1]	H'108+N*32	MSG_DATA_0（最初的Rx/Tx字节）							MSG_DATA_1							8/16/32位	数据		
MB[x].MSG_DATA[2][3]	H'10A+N*32	MSG_DATA_2							MSG_DATA_3							8/16位			
MB[x].MSG_DATA[4][5]	H'10C+N*32	MSG_DATA_4							MSG_DATA_5							8/16/32位			
MB[x].MSG_DATA[6][7]	H'10E+N*32	MSG_DATA_6							MSG_DATA_7							8/16位			
MB[x].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. 灰色表示的位为保留位，只能写0，读取值不一定为0。

2. 邮箱0的MCR1位的值总是1。

3. 邮箱0不支持ATX和DART，并且邮箱0的MBC的设定值有限制。

4. 在MCR15位为1时，信息控制以及LAFM的STDID、RTR、IDE、EXTID的顺序与HCAN2不同。

图 19.3 邮箱 N 的结构

19.3.2 信息控制字段

寄存器名	地址	位	位名	说明
MB[x]. CONTROL0H	H'100+N*32	15	IDE	ID 扩展 区分 CAN 数据帧和远程帧是标准格式还是扩展格式。 0: 标准格式 1: 扩展格式
		14	RTR	远程发送请求 区分数据帧和远程帧。根据数据帧或者远程帧, 接收 CAN 帧改写此位。 【重点】 如果在 MBC=B'001 时将数据帧的自动发送 (ATX) 位置位, 就无法将 RTR 置位。如果接收远程帧, 就通过对应的 RFPR 位或者 IRR2 (远程帧请求中断) 位通知 CPU。但是, 因为 RCAN-ET 需要将当前的信息作为数据帧发送, 所以 RTR 位不变。 【重点】 为了在 MBC=B'001 且 ATX=1 时自动对远程帧进行应答, 必须通过程序将 RTR 标志置 0, 以便允许发送数据帧。 【注意】 在为了发送远程帧而设定邮箱时, 用于传送的 DLC 为保存在邮箱中的内容。 0: 数据帧 1: 远程帧
		13	—	保留位 只能写 0, 不保证读取值。
		12 ~ 2	STDID [10:0]	标准 ID 设定数据帧和远程帧的 ID (标准 ID)。
		1、0	EXTID [17:16]	扩展 ID 设定数据帧和远程帧的 ID (扩展 ID)。
MB[x]. CONTROL0L	H'102+N*32	15 ~ 0	EXTID [15:0]	

• 邮箱 0

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

【注】 MBC1 的值总是 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	地址	位	位名	说明
MB[x]. CONTROL1H	H'110+N*32	15、14	—	保留位 只能写 0，不保证读取值。
		13	NMC	新信息控制 当此位被置 0 时，RXPR 或者 RFPR 已被置位的邮箱就不保存新的信息而保持以前的信息，并将 USMR 的对应位置位；当此位被置 1 时，RXPR 或者 RFPR 已被置位的邮箱就重写新的信息，并将 USMR 的对应位置位。 【重点】 如果用数据帧改写远程帧或者用远程帧改写数据帧，就有可能将 RXPR 和 RFPR 标志（和 USMR 一起）设定在同一邮箱。此时，也写邮箱控制字段内的 RTR 位。 0: 溢出模式 1: 重写模式
MB[x]. CONTROL1H	H'110+N*32	12	ATX	数据帧的自动发送 如果此位被置 1 并且用邮箱接收远程帧，就保存 DLC。而且，通过将 TXPR 置位，使用当前信息数据和被改写的 DLC，从相同邮箱自动传送数据帧。如由信息发送优先级位（MCR2）构成的那样，按照 ID 优先级或者邮箱优先级控制发送时序。要使用此功能时，必须将 MBC[2:0] 置 B'001。在使用此功能进行发送时，所使用的长度码（DLC）为接收到的 DLC，所以远程帧的 DLC 必须是对应所要求的数据帧的 DLC。 【重点】 因为在使用 ATX 并且 MBC 为 B'001 时，和应答信息时的情况相同，远程帧的 ID 必须与数据帧的 ID 完全相同，所以不能使用 IDE 位的过滤器。 【重点】 在使用此功能时，与接收远程帧无关，不将 RTR 位置位。如果接收远程帧，虽然通过已置位的 RFPR 通知 CPU。但是，RCAN-ET 需要将当前信息作为数据帧进行发送，所以 RTR 位不变。 【重点】 在溢出状态（设定为 NMC=0 的邮箱的 USMR 被置位）下，放弃接收信息。当已设定为 ATX=1 的邮箱因远程帧而进入溢出状态时，就有可能接受旧信息的自动发送请求。 0: 数据帧的自动发送无效 1: 数据帧的自动发送有效 【注】 此位只存在于邮箱 1～15。在邮箱 0 中，此位为保留位。
		11	DART	自动再发送无效 当此位被置 1 时，就在发生 CAN 总线错误事件或者 CAN 总线仲裁失败时信息的自动再发送无效。实际上，在使用此功能时，对应的 TXCR 位在发送最初就自动被置位。当此位被置 0 时，RCAN-ET 在发送正常结束之前或者通过 TXCR 取消之前只继续发送所请求的次数。 0: 再发送有效 1: 再发送无效 【注】 此位只存在于邮箱 1～15。在邮箱 0 中，此位为保留位。

寄存器名	地址	位	位名	说明
MB[x]. CONTROL1H	H'110+N*32	10 ~ 8	MBC[2:0]	邮箱结构 这些位设定如表 19.3 所示的各邮箱功能。当 MBC 为 B'111 时，邮箱无效。即：与 TXPR 或其他设定无关，不发送和接收信息。 禁止将 MBC 置 B'100、B'101 或者 B'110。如果将 MBC 设定为其他值，就能使用 LAFM 字段。 在设定 MBC 为接收时，不能设定 TXPR。无硬件保护功能，TXPR 保持置位的状态。通过硬件，将邮箱 0 的 MBC1 固定为 1。
MB[x]. CONTROL1L	H'111+N*32	7 ~ 4	—	保留位 只能写 0，不保证读取值。
		3 ~ 0	DLC[3:0]	数据长度码 对用数据帧发送的数据字节数（0 ~ 8）进行编码。在发送远程帧请求时，所使用的 DLC 值必须和所请求的数据帧的 DLC 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

表 19.3 邮箱功能的设定

MBC2	MBC1	MBC0	数据帧 的发送	远程帧 的发送	数据帧 的接收	远程帧 的接收	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	• 能通过 ATX 使用 * • 禁止使用邮箱 0 • 能使用 LAFM
0	1	0	不可	不可	可	可	• 能使用邮箱 0 • 能使用 LAFM
0	1	1	不可	不可	可	不可	• 能使用邮箱 0 • 能使用 LAFM
1	0	0	禁止设定				
1	0	1	禁止设定				
1	1	0	禁止设定				
1	1	1	邮箱无效（初始值）				

【注】 * 为了支持自动再发送功能，在 MBC 为 B'001 且 ATX 为 1 时，必须将 RTR 置 0。在 ATX 为 1 时，不能使用 IDE 过滤器。

19.3.3 局部接收过滤屏蔽（LAFM）

当 MBC 为 B'001、B'010 或者 B'011 时，此字段用于接收 LAFM。LAFM 允许邮箱接受多个接收 ID。如图 19.4 所示，LAFM 有 2 个 16 位可读写区域。

寄存器名	地址	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	存取长度	字段名	
MB[0].LAFMH	H'104+N*32	IDE_LAFM	0	0	STDID_LAFM[10:0]												EXTID_LAFM[17:16]		16/32位	LAFM
MB[0].LAFML	H'106+N*32	EXTID_LAFM[15:0]																16位		

图 19.4 局部接收过滤屏蔽（LAFM）

如果将 1 个位设定到 LAFM，RCAN-ET 就在检索相同 CAN-ID 的邮箱时忽视已接收的 CAN-ID 所对应的位。如果此位已被清除，已接收的 CAN-ID 所对应的位就必须和保存的邮箱设定的 STDID/IDE/EXTID 相同。LAFM 的结构和邮箱的信息控制相同。在不使用此功能时，必须填写 0。

- 【注】
1. RCAN-ET 开始从邮箱 15 到邮箱 0 检索相同的 ID。RCAN-ET 一旦检索到相同的 ID，就立即停止检索，与 NMC 或者 RXPR/RFPR 标志无关，保存此信息。这表示即使使用 LAFM，接收信息也只保存在 1 个邮箱。
 2. 当接收到一条信息并检索到相同的邮箱时，就将全部信息保存到邮箱。因为 STDID、RTR、IDE 和 EXTID 在使用 LAFM 时被更新为所接收信息的 STDID、RTR、IDE 和 EXTID，所以和接收前设定的内容不同。

寄存器名	地址	位	位名	说明
MB[x]. LAFMH	H'104+N*32	15	IDE_LAFM	IDE 位的过滤屏蔽位 0: 对应的 IDE 位有效 1: 对应的 IDE 位无效
		14、13	—	保留位 只能写 0，不保证读取值。
		12 ~ 2	STDID_LAFM [10:0]	STDID[10:0] 的过滤屏蔽位 0: 对应的 STDID 位有效 1: 对应的 STDID 位无效
		1、0	EXTID_LAFM [17:16]	EXTID[17:0] 的过滤屏蔽位 0: 对应的 EXTID 位有效 1: 对应的 EXTID 位无效
MB[x]. LAFML	H'106+N*32	15 ~ 0	EXTID_LAFM [15:0]	

19.3.4 信息数据字段

保存发送和接收的 CAN 信息。MSG_DATA_0 对应最初发送和接收的数据字节。CAN 总线上的位的排列顺序为从 bit7 到 bit0。

19.4 RCAN-ET 的控制寄存器

以下说明 RCAN-ET 的控制寄存器。RCAN-ET 的控制寄存器只能以字为单位（16 位）进行存取。
RCAN-ET 的控制寄存器如表 19.4 所示。

表 19.4 RCAN-ET 的控制寄存器结构

寄存器名	略称	存取长度（位）
主控控制寄存器	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

19.4.1 主控控制寄存器（MCR）

MCR 是 16 位可读写寄存器，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	初始值	R/W	说明
15	MCR15	1	R/W	ID 排列的变更 此位能设定和 HCAN2 兼容的信息控制以及 LAFM 的 STDID、RTR、IDE 和 EXTID 的顺序。此位只能在复位模式中进行更改。有关 ID 排列的变更顺序请参照图 19.5。 0: RCAN-ET 和 HCAN2 的顺序相同 1: RCAN-ET 和 HCAN2 的顺序不同
14	MCR14	0	R/W	自动总线断开时的停机 在此位和 MCR6 同时被置位时，如果 RCAN-ET 进入总线断开状态，MCR1 就立即被自动置位。此位只能在复位模式中进行更改。 0: 在通常的返回顺序（128×11 隐性位）中保持 RCAN-ET 总线断开状态。 1: 如果 MCR6 被置位，RCAN-ET 就在进行总线断开状态后立即转移到停机模式。
13 ~ 11	—	全 0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
10 ~ 8	TST[2:0]	000	R/W	测试模式 此位设定测试模式的有效 / 无效。必须注意：在启动测试模式之前，需要将 RCAN-ET 设定为停机模式或者复位模式。这是为了避免测试模式的转移影响正在进行的发送和接收。详细内容请参照 “19.6.2 测试模式的设定”。 因为测试模式只用于诊断和测试，所以在 RCAN-ET 正常运行时不能使用。 000：正常模式 001：只监听模式（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 此位设定 CAN 睡眠模式的自动唤醒模式的有效 / 无效。当此位被置位时，RCAN-ET 就在检测到 CAN 总线激活（显性位）后自动解除 CAN 睡眠模式（MCR5）；当清除此位时，RCAN-ET 就不自动解除 CAN 睡眠模式。 RCAN-ET 不能保存唤醒 RCAN-ET 的信息。 0：由 CAN 总线激活产生的自动唤醒模式无效 1：由 CAN 总线激活产生的自动唤醒模式有效 【注】 在 CAN 睡眠模式中，不能更改 MCR7 位。
6	MCR6	0	R/W	总线断开时的停机 在总线断开时 MCR1 被置位的情况下，此位设定是否允许立即进入停机模式。此位只能在复位模式或者停机模式中进行更改。必须注意：如果在总线断开时进入如停机模式，CAN 控制器就立即返回到错误主动模式。 0：即使 MCR1 被置位，也不在总线断开时进入停机模式，而等待返回顺序结束。 1：如果在总线断开时 MCR1 或者 MCR14 有效，就立即进入停机模式

位	位名	初始值	R/W	说明
5	MCR5	0	R/W	CAN 睡眠模式 此位设定是否允许 CAN 睡眠模式的转移。如果在 RCAN-ET 处于停机模式时将此位置位，就允许转移到 CAN 睡眠模式。在进入停机模式后允许设定此位。2 个错误计数器 (REC 和 TEC) 在 CAN 睡眠模式期间不变。解除 CAN 睡眠模式的方法有以下 2 种： • 给 MCR5 位写 0 • 在 MCR7 有效时，检测 CAN 总线的显性位。 如果自动唤醒模式无效，RCAN-ET 就在 CAN 睡眠模式结束之前，忽视全部 CAN 总线的运行。在解除 CAN 睡眠模式时，RCAN-ET 在进入 CAN 总线运行之前检查 11 个隐性位，取得与 CAN 总线的同步。这意味着在使用第 2 种方法时，RCAN-ET 无法接收最初的信息。同样，CAN 收发器在结束待机模式时也无法接收最初的信息，因此需要采用此方法设计软件。 在 CAN 睡眠模式中，只能存取 MCR、GSR、IRR 和 IMR 寄存器。 0：已解除 CAN 睡眠模式 1：允许转移到 CAN 睡眠模式 【注】 建议将 RCAN-ET 最初设定为停机模式后转移到 CAN 睡眠模式，以便 CPU 能在转移到 CAN 睡眠模式之前清除等待状态的全部中断。在清除全部中断的同时，必须将 RCAN-ET 从停机模式转移到 CAN 睡眠模式（同时给 MCR5 和 MCR1 分别写 1 和 0。）。
4、3	—	全 0	R	保留位 读写值总是 0。
2	MCR2	0	R/W	信息发送优先级 此位选择等待中的发送数据的发送顺序。当此位被置 1 时，按发送等待寄存器 (TXPR) 的位顺序发送数据。最优先开始邮箱 15 的发送，然后连续发送到邮箱 1 为止（将邮箱设定为发送邮箱时）。当此位被清除时，全部发送信息（通过运行内部仲裁）就按照优先级进入队列。最优先进入队列的信息有最小数字的仲裁字段 (STDID+IDE 位 +EXTID (IDE=1 时) +RTR 位)，被最先发送。内部仲裁包含 RTR 位和 IDE 位（使用与 2 个 CAN 节点之间 CAN 总线上的仲裁同时开始转移的相同方法，运行内部仲裁）。此位只能在复位模式或者停机模式中进行更改。 0：按信息 ID 优先级进行发送 1：按邮箱号顺序（邮箱 15→邮箱 1）进行发送

位	位名	初始值	R/W	说明
1	MCR1	0	R/W	停机请求 当此位被置位时，CAN 控制器在结束当前运行后立即进入停机模式（在此，从 CAN 总线分离）。在此位被清除前，RCAN-ET 一直保持停机模式。在停机模式中，CAN 接口与 CAN 总线运行无关，不进行信息的保存或者发送。除了用于向自身通知停机状态的 IRR0 和 GSR4 以外，保持用户寄存器（包含邮箱内容和 TEC/REC）的内容。在 CAN 总线处于空闲或者间歇状态时，与 MCR6 无关，RCAN-ET 在 1 位时间内为停机模式。当 MCR6 被置位时，总线断开中的停机请求也在 1 位时间内运行。除此以外，事前结束总线断开返回顺序。如果进入停机模式，就通过 IRR0 和 GSR4 通知进入停机模式。 当 MCR14 和 MCR6 被置位时，如果 RCAN-ET 为总线断开状态，此位就立即被自动置位。 在停机模式中，RCAN-ET 与总线运行无关，因此除了位时序的设定以外，能更改 RCAN-ET 的设定。为了再次加入 CAN 总线运行，需要将此位清 0。在清除后，RCAN-ET 在等到检测出 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. 在发行停机请求后并且在转移到停机模式之前，CPU 不能存取（通过 IRR0 和 GSR4 进行通知）TXPR 和 TXCR，也不能清除此位。在 MCR1 被置位后，只能在进入停机模式后或者通过软件、硬件进行复位解除此状态。 2. 只有在给 BCR1 和 BCR0 寄存器设定适当的波特率时，才能转移到停机模式或者从停机模式转移。
0	MCR0	1	R/W	复位请求 此位控制 RCAN-ET 模块的复位。当此位从 0 变为 1 时，RCAN-ET 控制器进入复位程序，为了在重新初始化内部逻辑后通知复位模式，将 GSR3 和 IRR0 置位。初始化全部用户寄存器。 在此位被置位期间，能重新设定 RCAN-ET（结构模式）。为了加入 CAN 总线，需要给此位写 0 进行清除。在清除后，RCAN-ET 在等到检测出 11 个隐性位后加入 CAN 总线。 为了对 CAN 总线的值进行采样，需要设定适当的波特率。在上电复位后，总是将此位和 GSR3 置位。这表示请求复位并需要设定 RCAN-ET。复位请求相当于上电复位，通过软件控制。 0: CAN 接口正常模式（MCR0=0 且 GSR3=0） [清除条件] 当在 RCAN-ET 复位后写 0 时 1: CAN 接口的复位模式转移请求

• MCR15（ID排列の変更）=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
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排列の変更）=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
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

图 19.5 ID 排列の変更

19.4.2 通用状态寄存器（GSR）

GSR 是 16 位只读寄存器，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	初始值	R/W	说明
15 ~ 6	—	全 0	R	保留位 读写值总是 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，就立即将此位置 1，如果再次变为错误主动状态，就清除此位。这意味着在错误被动中和总线断开状态下，GSR5 保持为 1。因此，要想得知正确的状态时，必须检查 GSR5 和 GSR0。 0: RCAN-ET 既不处于错误被动状态也不处于总线断开状态 [清除条件] 在 RCAN-ET 处于错误主动状态时 1: RCAN-ET 处于错误被动状态（GSR0=0 时）或者总线断开状态（GSR0=1 时） [置位条件] 当 $TEC \geq 128$ 或者 $REC \geq 128$、或者选择错误被动模式时

位	位名	初始值	R/W	说明
4	GSR4	0	R	停机 / 睡眠状态 表示 CAN 控制器是否处于停机 / 睡眠状态。必须注意：此标志的清除时间和 IRR12 的置位时间不同。因为此标志反映 CAN 控制器状态，所以不充分反映 RCAN-ET 的状态。当结束 CAN 睡眠模式并清除 MCR5 时，RCAN-ET 就变为可存取状态。在经过传送时钟 2 个位后，结束 CAN 控制器的 CAN 睡眠模式。 0: RCAN-ET 既不处于停机模式也不处于 CAN 睡眠模式 1: 处于停机模式 (MCR1=1 时) 或者 CAN 睡眠模式 (MCR5=1 时) [置位条件] 当 MCR1 被置位并且 CAN 总线处于间歇或者空闲状态时，或者当 MCR5 被置位并且 RCAN-ET 处于停机模式时，或者当 MCR14 和 MCR6 都被置位并且 RCAN-ET 转移到总线断开模式时。
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] 在 RCAN-ET 的软件或者硬件复位后
2	GSR2	1	R	信息发送进行标志 此标志向 CPU 表示 RCAN-ET 是处于总线断开状态、还是正在发送信息或者正在发送因在发送中检测到错误所产生的错误 / 超载标志。TXACK 的置位时序和 GSR2 的清除时序不同。在帧结束的第 7 位将 TXACK 置位；在没有等待发送的信息时，在间歇的第 3 位将 GSR2 置位。另外，也能在仲裁失败、总线空闲、接收、复位、停机转移进行置位。 0: RCAN-ET 处于总线断开状态或者正在发送信息 1: [置位条件] 既不处于总线断开状态，也不在发送信息
1	GSR1	0	R	发送 / 接收警告标志 此标志表示错误警告。 0: [清除条件] 当 $(TEC < 96 \text{ 且 } REC < 96)$ 或者总线断开时 1: [置位条件] 当 $96 \leq TEC < 256$ 或者 $96 \leq REC < 256$ 时 【注】 为了对总线断开返回顺序所需 11 个隐性位的反复次数进行计数，REC 在总线断开时进行递增计数。但是，在总线断开时，不将此位置位。
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者在硬件 / 软件复位后 1: [置位条件] 当 $TEC \geq 256$ (总线断开状态) 时

19.4.3 位结构寄存器 0、1 (BCR0、BCR1)

BCR0 和 BCR1 分别为 16 位可读写寄存器，设定 CAN 位时序参数和 CAN 接口的波特率预分频器。
时间量子的定义如下：

$$\text{时间量子} = 2 \times \text{BRP} / f_{\text{clk}}$$

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 和 TSEG2 的设定，请参照表 19.5。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG1[3:0]				—	TSG2[2:0]				—	—	SJW[1:0]		—	—	BSP
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R	R/W

位	位名	初始值	R/W	说明
15 ~ 12	TSG1[3:0]	0000	R/W	时间量子 1 因为使用正相位误差补偿 CAN 总线的边沿，所以这些位用于设定段 TSEG1 (=PRSEG+PHSEG1)。能设定 4 ~ 16 的时间量子。 0000: 禁止设定 0001: 禁止设定 0010: 禁止设定 0011: PRSEG+PHSEG1=4 个时间量子 0100: PRSEG+PHSEG1=5 个时间量子 : : 1111: PRSEG+PHSEG1=16 个时间量子
11	—	0	R	保留位 读写值总是 0。
10 ~ 8	TSG2[2:0]	000	R/W	时间量子 2 因为使用负相位误差补偿 CAN 总线的边沿，所以这些位用于设定段 TSEG2 (=PRSEG2)。能设定 2 ~ 8 的时间量子 000: 禁止设定 001: PHSEG2=2 时间量子 (根据条件，禁止设定。请参照表 19.5) 010: PHSEG2=3 个时间量子 011: PHSEG2=4 个时间量子 100: PHSEG2=5 个时间量子 101: PHSEG2=6 个时间量子 110: PHSEG2=7 个时间量子 111: PHSEG2=8 个时间量子
7、6	—	全 0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
5、4	SJW[1:0]	00	R/W	再同步跳转宽度 设定同步跳转宽度。 00: 同步跳转宽度 =1 个时间量子 01: 同步跳转宽度 =2 个时间量子 10: 同步跳转宽度 =3 个时间量子 11: 同步跳转宽度 =4 个时间量子
3 ~ 1	—	全 0	R	保留位 读写值总是 0。
0	BSP	0	R/W	位采样点 设定数据的采样点。 0: 在一处进行位采样 (时间段 1 的最后) 1: 在三处进行位采样 (PHSEG1 的最后 3 个时钟周期的上升沿)

(2) BCR0

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

位	位名	初始值	R/W	说明
15 ~ 8	—	全 0	R	保留位 读写值总是 0。
7 ~ 0	BRP[7:0]	0000000	R/W	波特率预分频器 这些位设定对应 1 个时间量子的外围总线时钟数。 0000000: 2× 外围总线时钟 0000001: 4× 外围总线时钟 0000010: 6× 外围总线时钟 : : 2×(寄存器的值 +1)× 外围总线时钟 1111111 : 512× 外围总线时钟

• 有关位结构寄存器



- SYNC_SEG : 在 CAN 总线上建立节点同步的段 (通常在此段中产生位边沿转移)
- PRSEG : 调整网络间物理性延迟的段
- PHSEG1 : 相位漂移 (正向) 的缓冲段 (在同步或者再同步时进行扩大)
- PHSEG2 : 相位漂移 (负向) 的缓冲段 (在同步或者再同步时进行缩小)
- TSEG1 : TSG1+1
- TSEG2 : TSG2+1

RCAN-ET 位速率根据以下表达式进行计算：

$$\text{位速率} = f_{\text{clk}} / (2 \times (\text{BRP} + 1) \times (\text{TSEG1} + \text{TSEG2} + 1))$$

此时，BRP 是寄存器的值赋予的值，TSEG1 和 TSEG2 是从后述表中算出的值。上述位速率计算式的时间量子“+1”是根据 SYNC_SEG 的 1 个时间量子得出的。

f_{clk} = 外围总线时钟

BCR 设定时的限制事项：

$$\text{TSEG1}(\text{Min.}) > \text{TSEG2} \geq \text{SJW}(\text{Max.}) \quad (\text{SJW} = 1 \sim 4)$$

$$8 \leq \text{TSEG1} + \text{TSEG2} + 1 \leq 25 \text{ 个时间量子} \quad (\text{TSEG1} + \text{TSEG2} + 1 \neq 7)$$

$$\text{TSEG2} \geq 2$$

如果位结构寄存器 TSEG1 和 TSEG2 的值在表 19.5 所示设定值的范围内，就满足上述限制事项。表 19.5 的值为 SJW 的可设定值。以“不可”表示无法取得 TSEG1 和 TSEG2 的组合。

表 19.5 TSG 和 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：假设 f_{clk} 为 40MHz，位速率为 500kbps，BRP=3、TSEG1=6、TSEG2=3 满足条件。此时，给 BCR1 和 BCR0 分别写 H'5200 和 H'0003。

例 2：假设 f_{clk} 为 35MHz，位速率为 250kbps，BRP=4、TSEG1=8、TSEG2=5 满足条件。此时，给 BCR1 和 BCR0 分别写 H'7400 和 H'0004。

例 3：假设 f_{clk} 为 32MHz，位速率为 500kbps，BRP=1、TSEG1=11、TSEG2=4 满足条件。此时，给 BCR1 和 BCR0 分别写 H'A300 和 H'0001。

例 4：假设 f_{clk} 为 20MHz，位速率为 500kbps，BRP=1、TSEG1=6、TSEG2=3 满足条件。此时，给 BCR1 和 BCR0 分别写 H'5200 和 H'0001。

19.4.4 中断请求寄存器 (IRR)

IRR 是 16 位可读写寄存器，由各中断源的状态标志构成。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	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	初始值	R/W	说明
15、14	—	全 0	R	保留位
13	IRR13	0	R/W	信息错误中断 此中断位表示在测试模式中发生了信息错误。但是，即使在测试模式中发生信息超载，也不将此位置位。另外，此位只在测试模式中有效。 0: 在测试模式中没有发生信息错误 [清除条件] 写 1 1: [置位条件] 在测试模式中发生信息错误
12	IRR12	0	R/W	CAN 睡眠模式时的总线激活 此位表示 CAN 总线激活的存在。在 RCAN-ET 处于 CAN 睡眠模式时，如果在 CAN 总线上检测到显性位，就将此位置位。如果给此位写 1，就清除中断。忽视写 0。在不使用自动唤醒功能并且不请求此中断时，必须通过对应的屏蔽位将此中断设定为无效。如果不使用自动唤醒功能而请求此中断，就在从 CAN 睡眠模式返回后清除此位。这是为了防止因接收线的新的下降沿再次将中断置位。 必须注意：此中断的置位时间和 GSR4 的清除时间不同。 0: 总线空闲状态 [清除条件] 写 1 1: 在 CAN 睡眠模式中检测到 CAN 总线激活 [置位条件] 当在 CAN 睡眠模式中从 CRx0 检测到显性位的状态变化时
11、10	—	全 0	R	保留位
9	IRR9	0	R	信息溢出 / 重写中断标志 此位表示虽然接收了信息，但是由于对应的 RXPR 或者 RFPR 已被置 1，所以不读邮箱内的信息。因为 CPU 没有清除此位，所以通过 NMC 位的设定，放弃（溢出）或者重写新接收的信息。如果给 UMSR 寄存器的对应位写 1，就清除此位。另外，如果给 MBIMR 的所有对应位写 1，也能清除此位。此位的写操作无效。 0: 没有信息溢出 / 重写的通知 [清除条件] 当清除 UMSR 的全部位、或者将与已置位的 UMSR 对应的全部 MBIMR 置位时 1: 因发生溢出而放弃接收信息，或者重写信息 [置位条件] 当在对应的 PXPR 或者 RFPR=1，且 MBIMR=0 的情况下接收信息时

位	位名	初始值	R/W	说明
8	IRR8	0	R	邮箱空的中断标志 当 1 条被设定用于发送的信息被正常发送（对应的 TXACK 标志被置位）或者发送中止（对应发送取消信息的 ABACK 标志被置位）时，就将此位置位。 此时，清除对应的 TXPR，邮箱能接收下一个发送信息数据。 实际上，根据未被 MBIMR 标志屏蔽的 TXACK 和 ABACK 位的 OR 信号，将此位置位。因此，如果清除全部 TXACK 位和 ABACK 位，就自动清除此位。另外，通过给 MBIMR 对应的全部位写 1，也能清此位除。此位的写操作无效。 0：发送或取消传送的信息未在处理中 [清除条件] 当清除全部 TXACK 位和 ABACK 的位、或者将与已置位的 TXACK 位和 ABACK 位对应的全部 MBIMR 置位时 1：信息正在被发送或者发送中止（取消发送）并能保存下一条信息 [置位条件] 当 TXACK 位或者 ABACK 位被置位时（MBIMR=0 时）
7	IRR7	0	R/W	超载帧 此标志表示 RCAN-ET 检测到超载帧的发送。如果给 IRR7 写 1，就清除此位。忽视写 0。 0：[清除条件] 写 1 1：[置位条件] 检测到超载帧条件
6	IRR6	0	R/W	总线断开的中断标志 当 RCAN-ET 处于线断开状态后或者从总线断开状态返回到错误主动状态时，将此位置位。因此，中断源为节点的 $TEC \geq 256$、总线断开返回顺序结束（128 次连续接收到 11 个隐性位）或者从总线断开状态转移到停止状态（自动或者手动）。 因为在 RCAN-ET 解除总线断开后，此位仍处于置位状态，所以需要软件清除此位。必须通过软件读 GSR0，判断 RCAN-ET 是处于总线断开状态还是错误主动状态。即使节点仍处于总线断开状态，也能通过写 1 清除此位。忽视写 0。 0：[清除条件] 写 1 1：因发送错误而转移到总线断开状态，或者从总线断开状态返回到错误主动状态 [置位条件] 当 $TEC \geq 256$ 或者在 128 次连续接收到 11 个隐性位后总线断开结束或者从总线断开状态转移到停止状态时
5	IRR5	0	R/W	错误被动的中断标志 表示因接收或者发送错误计数器而进入错误被动状态，或者因测试模式而强制进入错误被动状态。如果给此位写 1，就清除此位。忽视写 0。有时即使清除此位，节点仍保持错误被动状态。需要通过软件检查 GSR0 和 GSR5，判断 RCAN-ET 是处于错误被动状态还是总线断开状态。 0：[清除条件] 写 1 1：因发送 / 接收错误而进入错误被动状态 [置位条件] 当 $TEC \geq 128$ 或者 $REC \geq 128$ 或者使用错误被动测试模式时

位	位名	初始值	R/W	说明
4	IRR4	0	R/W	接收错误计数器的警告中断标志 在 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，就将此位置位。如果给此位写 1，就清除中断。忽视写 0。 0: [清除条件] 写 1 1: 因接收错误而产生的错误警告状态 [置位条件] 当 RCAN-ET 不处于总线断开状态且 $REC \geq 96$ 时
3	IRR3	0	R/W	发送错误计数器的警告中断标志 如果发送错误计数器 (TEC) 超过 95，就将此位置位。如果给此位写 1，就清除中断。忽视写 0。 0: [清除条件] 写 1 (在使用中断并通过 CPU 清除中断时，必须在写 1 后读标志) 1: 因发送错误而产生的错误警告状态 [置位条件] 当 $TEC \geq 96$ 时
2	IRR2	0	R	远程帧请求的中断标志 表示邮箱接收到远程帧。如果至少有 1 个邮箱 (对应的 MBIMR 未被置位) 保存了远程帧的发送请求，就将此位置位。如果清除远程帧接收完成寄存器 (RFPR) 的全部位，就清除此位。另外，通过给 MBIMR 对应的全部位写 1，也能清除此位。此位的写操作无效。 0: [清除条件] 清除 RFPR 的全部位 1: 至少有 1 个远程帧的请求处于处理等待状态 [置位条件] 当在对应的 MBIMR=0 的状态下接收远程帧时
1	IRR1	0	R	数据帧接收的中断标志 表示有处于处理等待状态的接收数据帧。当此位为 1 时，至少有 1 个邮箱保存了未处理的信息。如果清除数据帧接收完成寄存器 (RXPR) 的全部位 (任何接收邮箱都没有未处理的信息)，就清除此位。设定 MBIMR 不为 0 的各接收邮箱的 RXPR 标志的逻辑与。 另外，通过给 MBIMR 对应的全部位写 1，也能清除此位。此位的写操作无效。 0: [清除条件] 清除 RXPR 的全部位 1: 接收数据帧并保存到邮箱 [置位条件] 当在对应的 MBIMR=0 的状态下接收数据时

位	位名	初始值	R/W	说明
0	IRR0	1	R/W	复位 / 停机 / 睡眠的中断标志 此位因下述 3 个中断源而被置位。 - 在软件复位 (MCR0) 或者硬件复位后, 转移到复位模式 - 在发行停机请求 (MCR1) 后, 转移到停机模式 - 在停机模式中发行睡眠请求 (MCR5) 后, 转移到 CAN 睡眠模式 在将此位置位后读 GSR, 能得知 RCAN-ET 的状态。 【注】 在需要发行睡眠模式请求时, 必须先使用停机模式。详细内容请参照 MCR5 的说明和图 19.8。 在 GSR3 或者 GSR4 从 0 变为 1 时, 或者从停机模式转移到 CAN 睡眠模式时, 将 IRR0 置位。因此, 在 RCAN-ET 在解除停机模式之后立即重新转移到停机模式时, 不清除 GSR4, 因而也不将 IRR0 置位。同样, 在从 CAN 睡眠模式直接转移到停机模式时, 也不将 IRR0 置位。在从停机模式/CAN 睡眠模式转移到发送/接收运行后到清除 GSR4 为止, 需要[1 位时间-TSEG2]~[1 位时间×2-TSEG2]的时间。 在复位模式中, 虽然 IRR0 被置位, 但是通过初始化 IMR0 被自动置位, 因此向 CPU 请求的中断无效。 0: [清除条件] 写 1 1: 转移到软件复位模式、停机模式或者 CAN 睡眠模式 [置位条件] 当在请求复位 (MCR0 或者硬件)、停机模式 (MCR1) 或者 CAN 睡眠模式 (MCR5) 后转移到复位 / 停机 / CAN 睡眠模式时

19.4.5 中断屏蔽寄存器 (IMR)

IMR 是 16 位可读写寄存器, 屏蔽对应中断请求寄存器 (IRR) 各位的中断 IRQ 输出信号的生成。如果将某位置 1, 就屏蔽对应该位的中断请求。IMR 直接控制 IRQ 的生成, 但是不禁止 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	初始值	R/W	说明
15 ~ 0	IMR[15:0]	全 1	R/W	对应 IRR[15:0] 各位的中断源屏蔽 当此位被置位时, 即使将 IRR 对应的位置位, 也不生成对应的中断信号。 0: 不屏蔽对应的 IRR (如果产生中断源, 就生成 IRQ) 1: 屏蔽 IRR 对应的中断

19.4.6 发送错误计数器（TEC） / 接收错误计数器（REC）

TEC 和 REC 是可读和可带条件写的 16 位寄存器，是表示 CAN 接口中的发送 / 接收信息错误数的计数器。计数值在上述参考文献 [1]、[2]、[3]、[4] 中有所规定。除了写错误计数器测试模式以外，此寄存器为只读寄存器，并只能通过 CAN 接口进行改写。通过转移到复位请求（MCR0）或者总线断开状态清除此寄存器。

在写错误计数器测试模式（TST[2:0]=B'100）中，能写此寄存器。而且只能给 TEC 和 REC 写相同的值，并且写到 TEC 的值被设定到 TEC 和 REC。在写此寄存器时，需要将 RCAN-ET 设定为停机模式。

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】* 只能在MBC的TST[2:0]=B'100的测试模式时写此位。在总线断开状态下，REC对总线断开恢复顺序所需的11个隐性位的接收次数进行计数。

位	位名	初始值	R/W	说明
15 ~ 8	TEC[7:0]	全 0	R/W*	发送错误计数器 如果在发送中检测到 CAN 规格的错误，此寄存器的值就递增。
7 ~ 0	REC[7:0]	全 0	R/W*	接收错误计数器 如果在接收中检测到 CAN 规格的错误，此寄存器的值就递增。

19.5 RCAN-ET 的邮箱寄存器

以下说明 RCAN-ET 的邮箱寄存器。RCAN-ET 的邮箱寄存器控制各邮箱并显示邮箱状态。

RCAN-ET 的邮箱寄存器如表 19.6 所示。

【重要】 以 2 个连续的字存取进行长字存取。

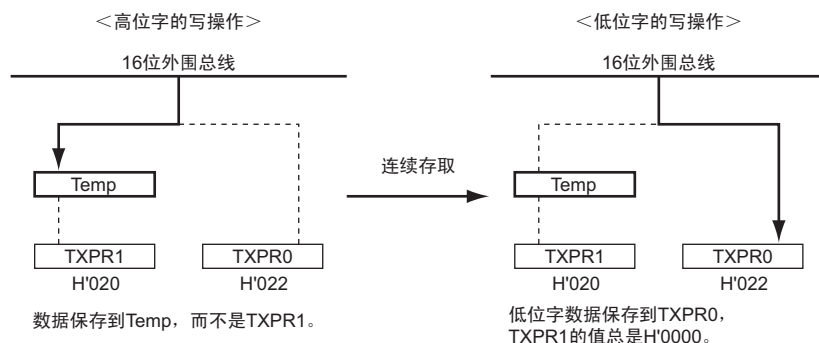
表 19.6 RCAN-ET 的邮箱寄存器

寄存器名	略称	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	

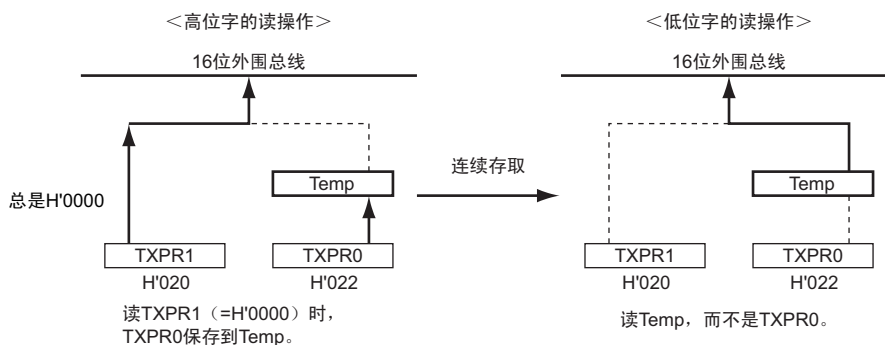
19.5.1 发送等待寄存器 1、0 (TXPR1、TXPR0)

TXPR1 和 TXPR0 相连，构成保存 CAN 模块发送等待标志的 32 位寄存器。当使用 16 位总线接口时，以 2 个连续的字存取进行长字存取。

<长字的写操作>



<长字的读操作>



TXPR1 寄存器的值总是 0，TXPR1 寄存器的写操作无效。TXPR0 寄存器控制邮箱 15 ~ 1。CPU 能通过给某位写 1 操作其对应的发送信息。写 0 无效，要清除 TXPR 时，必须将 TXCR 的对应位置位。CPU 能通过读 TXPR 得知发送等待的邮箱和正在发送中的邮箱。实际上，邮箱 0 除外的其他所有邮箱都有发送等待位。另外，禁止给未设定为发送邮箱的位写 1。

在正常发送信息后或者从 TXCR 中止发送后，RCAN-ET 清除对应的发送等待标志。因为 CAN 节点仲裁失败或者 CAN 总线发生错误，所以在还没有发送信息时不清除 TXPR 的标志。此时，如果没有将对应的邮箱信息控制字段的 DART（自动再发送无效）位置位，RCAN-ET 就自动尝试再发送；如果 DART 已被置位，就清除发送，通过邮箱空中断标志（IRR8）和中止应答寄存器（ABACK）的对应位通知 CPU。

在 TXPR 的状态发生变化时，即使总线仲裁失败或者 CAN 总线中发生错误，也总是发送 ID 所示优先级（MCR2=0）最高的信息。详细内容请参照“19.6 运行说明”。

在 RCAN-ET 将 TXPR 位的状态更改为 0 时，有可能会生成邮箱空中断标志（IRR8）。这表示邮箱发送正常结束或者发送中止。信息发送正常结束反映在 TXACK，信息的发送中止反映在 ABACK，能通过检查这些寄存器将对应的邮箱信息数据字段的内容改写到下一个发送邮箱。

(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】* 读取值总是H'0000。必须以长字为单位读写TXPR1/TXPR0，TXPR1的写操作无效。

(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】* 只能给设定为发送邮箱的对应位写1。
读取值总是H'0000。必须以长字为单位读写TXPR1/TXPR0，TXPR0的bit0的写操作无效。

位	位名	初始值	R/W	说明
15 ~ 1	TXPR0[15:1]	H'0000	R/W	表示对应的邮箱请求 CAN 帧发送。bit15 ~ 1 对应邮箱 15 ~ 1。在多个位被置位时，通过 MCR2 位的设定，按照信息 ID 优先级或者邮箱号的顺序进行发送。 0: 对应的邮箱处于发送信息空闲状态 [清除条件] 当信息发送结束或者信息发送中止（自动清除）时 1: 对应的邮箱请求发送
0	—	0	R	保留位 因为是接收专用邮箱，所以此位总是 0。忽视写 1，读取值为 0。

19.5.2 发送取消寄存器 0 (TXCR0)

TXCR0 是 16 位可读和可带条件写的寄存器，控制邮箱 15 ~ 1。CPU 使用 TXCR0 取消 TXPR 的发送请求。要清除 TXPR 的位时，必须通过 CPU 给对应的 TXCR 的位写 1。忽视写 0。

如果正常中止，CAN 控制器就清除对应的 TXPR 和 TXCR 的位，并将 ABACK 的位置位。但是，在邮箱开始发送后，不能通过此寄存器的位取消发送。此时，如果发送正常结束，CAN 控制器就清除 TXPR 和 TXCR 的位，并将 TXACK 的位置位。但是，在因仲裁失败或者总线发生错误而无法进行发送时，CAN 控制器清除对应的 TXPR 和 TXCR 的位，并将对应的 ABACK 的位置位。忽视 CPU 对不处于发送等待状态的邮箱发送的清除。此时，CPU 不能将 TXCR 的标志置位。

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】* 只能给接受到发送请求的发送邮箱的对应位写1。

位	位名	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发送取消。bit15 ~ 1 对应邮箱 15 ~ 1 (和 TXPR0[15:1])。 0: 对应的邮箱处于发送信息取消空闲状态 [清除条件] 当发送信息已被取消 (自动清除) 时 1: 对所对应的邮箱请求发送取消
0	—	0	R	保留位 因为是接收专用邮箱, 所以此位总是 0。忽视写 1, 读取值总是 0。

19.5.3 发送应答寄存器 0 (TXACK0)

TXACK0 是 16 位可读和可带条件写的寄存器, 用于将邮箱的正常发送通知 CPU。如果正常发送, RCAN-ET 就将 TXACK0 寄存器的对应位置位。CPU 能通过写 1 清除 TXACK0 的位。忽视写 0。

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】* 为了清除此位, 只能写 1。

位	位名	初始值	R/W	说明
15 ~ 1	TXACK0[15:1]	H'0000	R/W	通知对应邮箱所请求的发送已正常结束。bit15 ~ 1 对应邮箱 15 ~ 1。 0: [清除条件] 写 1 1: 正常发送了对应的邮箱信息 (数据帧或者远程帧) [置位条件] 当对应的邮箱信息发送结束时
0	—	0	R	保留位 因为是接收专用邮箱, 所以此位总是 0。忽视写 1, 读取值总是 0。

19.5.4 中止应答寄存器 0 (ABACK0)

ABACK0 是 16 位可读和可带条件写的寄存器, 用于将已按照请求中止邮箱发送通知 CPU。

在中止时, RCAN-ET 将 ABACK0 寄存器的对应位置位。CPU 能通过写 1 清除 ABACK0 的位。忽视写 0。RCAN-ET 通过将 ABACK0 的位置位, 确认已由对应的 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】* 为了清除此位, 只能写 1。

位	位名	初始值	R/W	说明
15 ~ 1	ABACK0[15:1]	H'0000	R/W	通知对应邮箱所请求的发送已被正常取消。bit15 ~ 1 对应邮箱 15 ~ 1。 0: [清除条件] 写 1 1: 已取消对应邮箱的信息 (数据帧或者远程帧) 发送 [置位条件] 当对应邮箱的信息发送已被取消时

位	位名	初始值	R/W	说明
0	—	0	R	保留位 因为是接收专用邮箱，所以此位总是 0。忽视写 1，读取值总是 0。

19.5.5 数据帧接收完成寄存器 0 (RXPR0)

RXPR0 是 16 位可读和可带条件写的寄存器，由数据帧的接收标志构成，表示设定为接收的邮箱已接收到数据帧。如果 CAN 数据帧被正常地保存到接收邮箱，就将 RXPR 的对应位置位。如果写 1 就清除对应的位。写 0 无效。但是，只有在设定为邮箱通过 MBC（邮箱结构）接收数据帧时才能将此位置位。在 RXPR 的位被置位时，如果对应的 MBIMR 没有被置位，就将 IRR1（数据帧接收中断标志）置位。此时，如果 IMR1 也没有被置位，就生成中断信号。只能通过接收数据帧将此寄存器的位置位，在接收远程帧时不将此寄存器的位置位。

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】* 为了清除此位，只能写1。

位	位名	初始值	R/W	说明
15 ~ 0	RXPR0[15:0]	H'0000	R/W	bit15 ~ 0 对应接收邮箱 15 ~ 0。 0: [清除条件] 写 1 1: 对应的邮箱接收到 CAN 数据帧 [置位条件] 当对应邮箱的数据帧接收结束时

19.5.6 远程帧接收完成寄存器 0 (RFPR0)

RFPR0 是 16 位可读和可带条件写的寄存器，由远程帧的接收标志构成，表示设定为接收的邮箱已接收到远程帧。如果 CAN 远程帧被正常地保存到接收邮箱，就将 RFPR 的对应位置位。如果写 1 就清除对应位。写 0 无效。全部邮箱都有此位，只有在设定为邮箱通过 MBC（邮箱结构）接收远程帧时才能将此位置位。在 RFPR 的位被置位时，如果对应的 MBIMR 没有被置位，就将 IRR2（远程帧请求中断标志）置位。此时，如果 IMR2 也没有被置位，就生成中断信号。只能通过接收数据帧将此寄存器的位置位，在接收数据帧时不将此寄存器的位置位。

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*

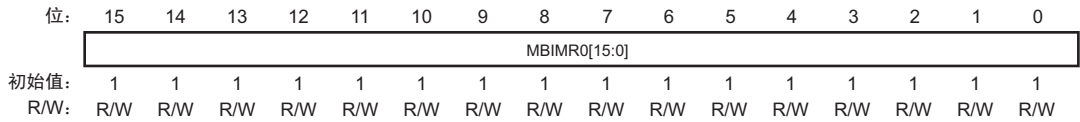
【注】* 为了清除此位，只能写1。

位	位名	初始值	R/W	说明
15 ~ 0	RFPR0[15:0]	H'0000	R/W	是邮箱 15 ~ 0 的远程请求接收标志 0: [清除条件] 写 1 1: 对应的邮箱接收到 CAN 远程帧 [置位条件] 当对应邮箱的远程帧接收结束时

19.5.7 邮箱中断屏蔽寄存器 0 (MBIMR0)

MBIMR0 是 16 位可读写寄存器。MBIMR0 是屏蔽与邮箱运行有关的 IRR (IRR1: 数据帧接收中断、IRR2: 远程帧接收中断、IRR8: 邮箱空中断、IRR9: 信息溢出中断) 的寄存器。在设定为接收邮箱时, 屏蔽由接收中断标志 (IRR1、IRR2 和 IRR9) 产生的中断, 不影响 RXPR、RFPR 和 UMSR 位的置位。在设定为发送邮箱时, 屏蔽由发送或者发送中止 (IRR8) 产生的中断或者邮箱空中断, 不影响发送产生的 TXPR/TXCR 位的清除和 TXACK 位的置位, 也不影响由发送中止产生的 TXPR/TXCR 位的清除和 ABACK 位的置位。

通过给对应要屏蔽的邮箱的位写 1 来设定屏蔽。在复位时, 屏蔽全部邮箱的中断。



位	位名	初始值	R/W	说明
15 ~ 0	MBIMR0[15:0]	H'FFFF	R/W	允许或者禁止邮箱 15 ~ 0 的中断请求。 0: 允许 IRR1、IRR2、IRR8 和 IRR9 的中断请求 1: 禁止 IRR1、IRR2、IRR8 和 IRR9 的中断请求

19.5.8 未读信息状态寄存器 0 (UMSR0)

UMSR0 是 16 位可读和可带条件写的寄存器。在接收新的信息时, 通过 CPU 记录没有存取内容的接收邮箱。在接收邮箱的新信息时, 如果 RXPR0 或者 RFPR0 的对应位没有被 CPU 清除, 就将 UMSR0 的位置位。如果写 1 就清除此位。写 0 无效。

在设定为发送邮箱时, 不对应的 UMSR0 的位置位。



【注】* 为了清除此位, 只能写1。

位	位名	初始值	R/W	说明
15 ~ 0	UMSR0[15:0]	H'0000	R/W	表示邮箱 15 ~ 0 的未读接收信息发生重写或者溢出。 0: [清除条件] 写 1 1: 未读接收信息新信息发生了重写或者溢出 [置位条件] 当在清除 RXPR0 或者 RFPR0 前接收到新的信息时

19.6 运行说明

19.6.1 RCAN-ET 的设定

以下说明硬件复位（上电复位）或者软件复位（MCR0）后的结构模式以及停机模式的 RCAN-ET 的设定。无论在何种情况下，RCAN-ET 都不能加入 CAN 总线激活，并且 RCAN-ET 设定的变更不影响 CAN 总线的通信。

(1) 复位顺序

软件复位或者硬件复位后的 RCAN-ET 的设定步骤如图 19.6 所示。在复位后，初始化全部寄存器。因此，必须在加入 CAN 总线激活之前设定 RCAN-ET。详细内容请参照图中的注释。

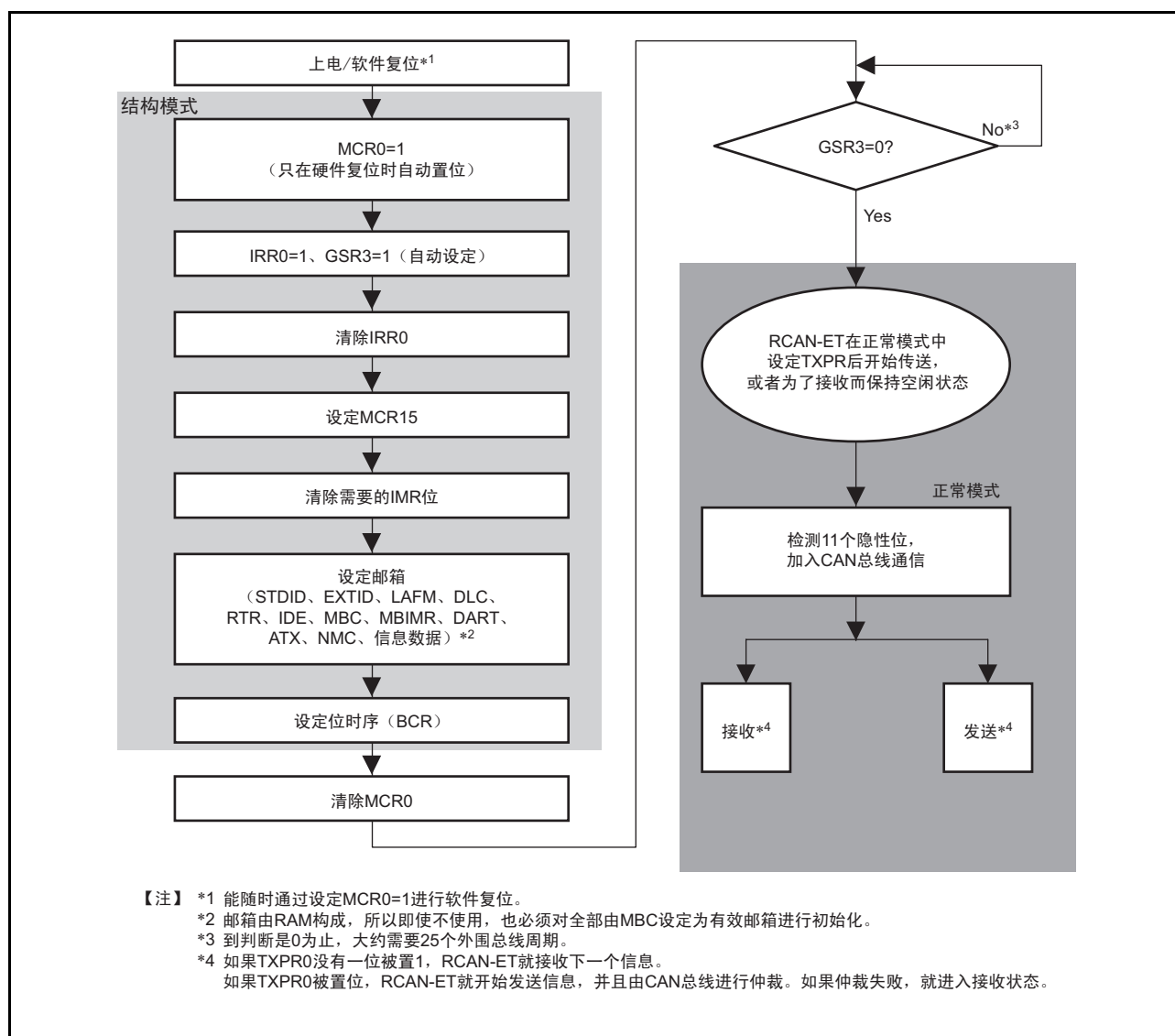


图 19.6 复位顺序

(2) 停机模式

在停机模式时，RCAN-ET 不能加入 CAN 总线激活。因此，用户能在不影响 CAN 总线通信的情况下更改所需的寄存器设定。此时必须注意：必须在更改寄存器之前等待 RCAN-ET 进入停机模式，这是因为不一定立刻就转移到停机模式（在 CAN 总线处于空闲状态或者睡眠状态时进行转移）。一旦 RCAN-ET 转移到停机模式，就将 GSR4 位置位。

必须在设定结束后解除停机请求。RCAN-ET 在从 CAN 总线检测到 11 个隐性位后加入 CAN 总线激活。

(3) CAN 睡眠模式

在 CAN 睡眠模式中，为了降低消耗电流而停止 RCAN-ET 主要模块的时钟。但是，只能存取 MCR、GSR、IRR 和 IMR 寄存器。因为在 CAN 睡眠模式中不能清除与发送 / 接收有关的中断，所以必须预先清除这些中断。RCAN-ET 的 CAN 睡眠模式的流程图如图 19.7 所示。

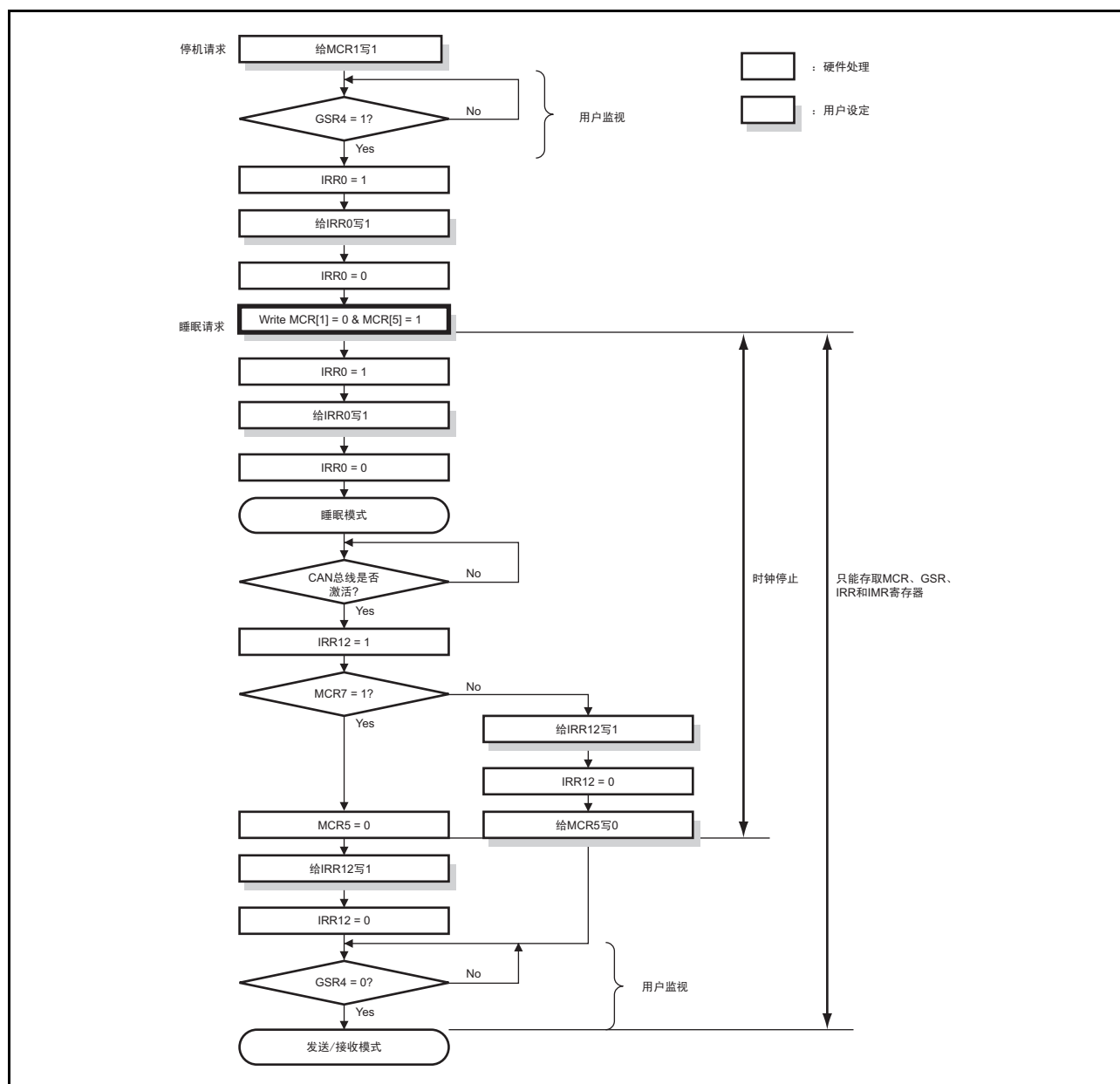


图 19.7 CAN 睡眠模式的流程图

可能的状态转移如图 19.8 所示。
除停机模式以外，不能将 MCR5（CAN 睡眠模式）位置位。
另外，在将 MCR1 位置位后，必须在清除 MCR1 前，将 GSR4 置位，并将 RCAN-ET 设定为停机模式。

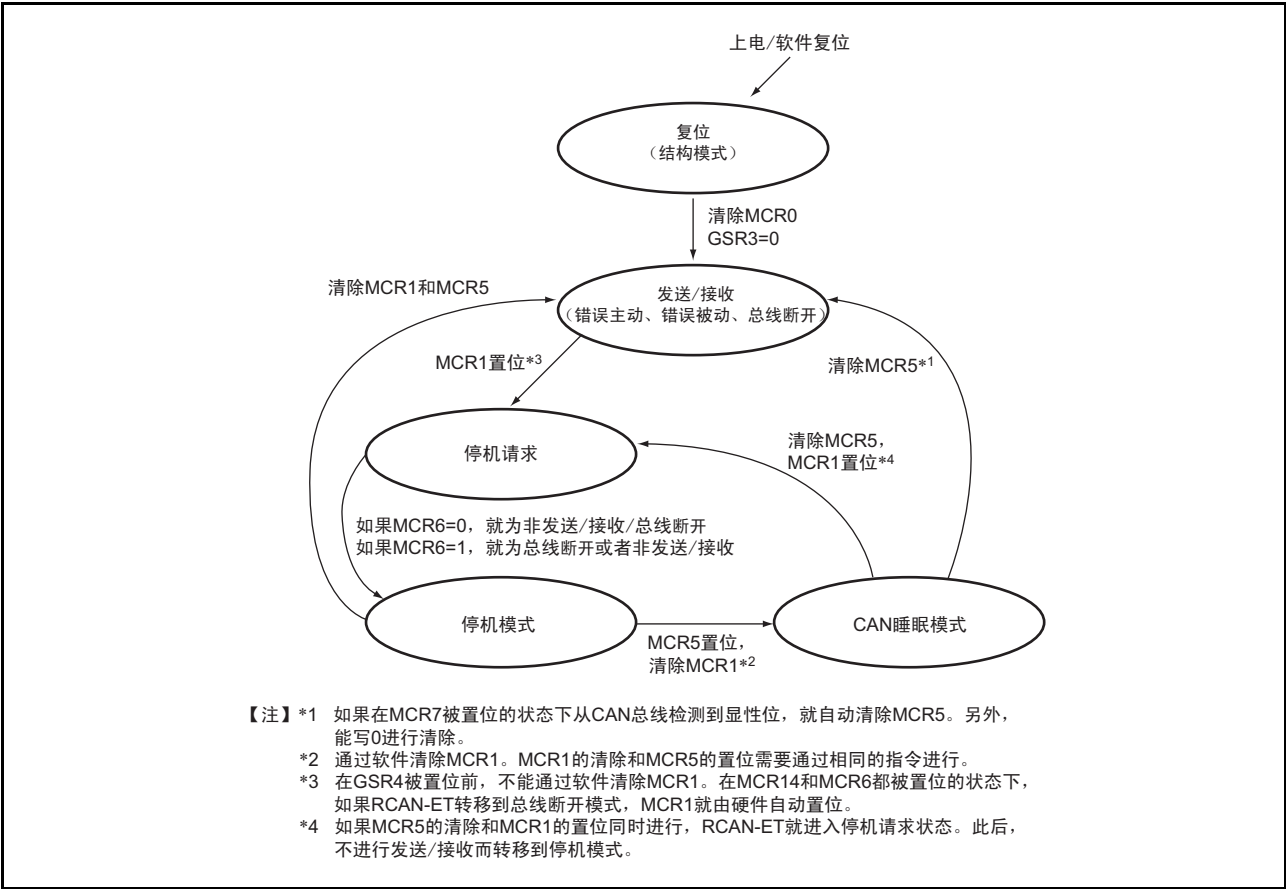


图 19.8 状态转移图

允许在各模式中进行存取的条件如表 19.7 所示。

表 19.7 能存取的寄存器

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】* 是 TXPR0 没有被置位的情况。

19.6.2 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 在正常模式中，以默认（初始值）方式运行。

测试模式的设定如表 19.8 所示。

只能在结构模式中选择测试模式。在执行所选测试模式时，必须在选择测试模式后解除结构模式（确认已设定 BCR0/BCR1）。

表 19.8 测试模式的设定

TST2	TST1	TST0	模式
0	0	0	正常模式
0	0	1	只监听模式（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

- 正常模式
RCAN-ET 正常运行。
- 只监听模式
是 ISO-11898 要求用于波特率检测的模式。在清除错误计数器后禁止运行，TEC/REC 的值不递增。另外，禁止 CTx0 输出，抑制由 RCAN-ET 生成的错误帧或者应答位。如果发生信息错误，就将 IRR13 置位。
- 自测试模式 1
RCAN-ET 自身生成应答位，如果需要就将信息保存到 RCAN-ET 的接收邮箱。必须将 CRx0/CTx0 引脚连接到 CAN 总线。
- 自测试模式 2
RCAN-ET 自身生成应答位，如果需要就将信息保存到 RCAN-ET 的接收邮箱。因为内部 CTx0 被反馈给内部 CRx0，所以不需要将 CRx0/CTx0 引脚连接到 CAN 总线和其他外部器件。CTx0 引脚只输出隐性位，CRx0 引脚无效。
- 写错误计数器
能在此模式中写 TEC/REC。能通过将大于 127 的值写到错误计数器，将 RCAN-ET 强制设定为错误被动模式。因为写到 TEC 的值被写到 REC，所以 TEC 和 REC 的值总是相同。同样，能通过写大于 95 的值，将 RCAN-ET 强制设定为错误警告模式。在写 TEC/REC 时，RCAN-ET 必须为停机模式（在写错误计数器时，MCR1=1）。按照 CAN 规格，必须在解除停机模式前解除此测试模式，然后使 TEC/REC 重新运行。
- 错误被动模式
能将 RCAN-ET 强制设定为错误被动模式。
虽然在执行错误被动模式时 REC 的值不变。但是，一旦在错误被动模式运行，如果接收到错误，REC 的值就正常递增。在此模式中，当 TEC 的值达到 256 时，RCAN-ET 变为总线断开状态，如果使用此模式，RCAN-ET 就不能变为错误主动状态。因此，RCAN-ET 在总线断开返回顺序的最后转移到错误被动状态，而不是错误主动状态。

当发生信息错误时，就在所有的测试模式中将 IRR13 位置位。

19.6.3 信息的发送顺序

(1) 信息发送请求

总线上 CAN 帧的发送顺序例子如图 19.9 所示。

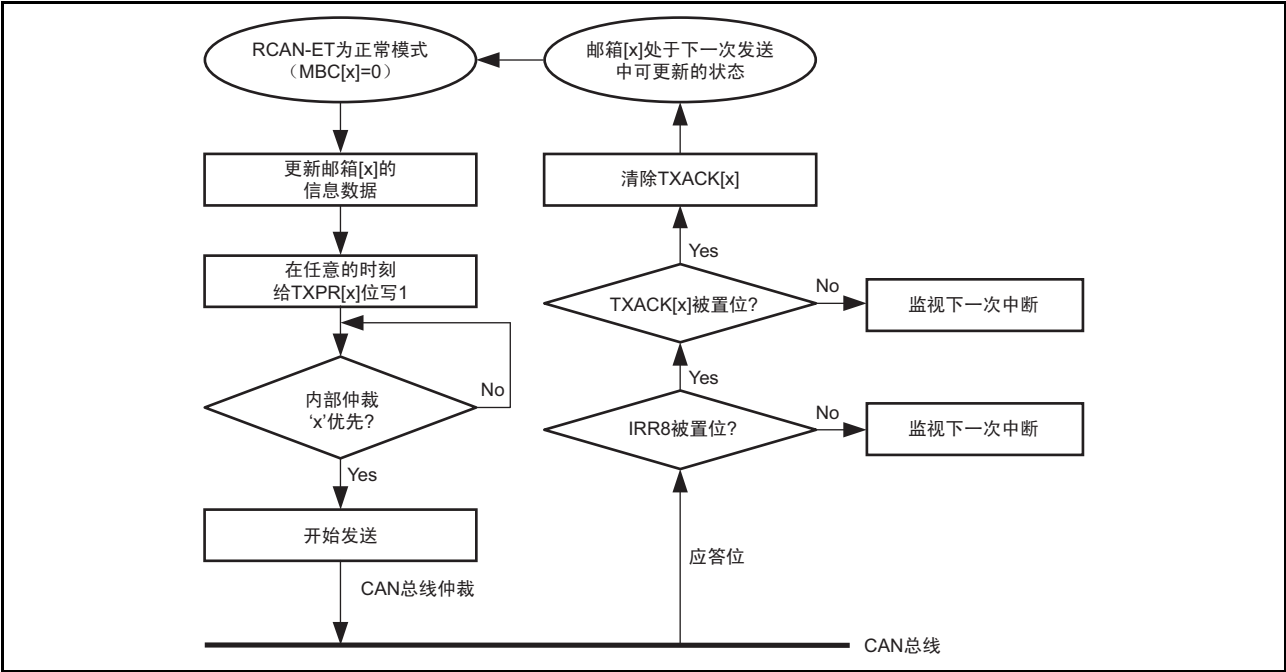


图 19.9 信息的发送请求

必须注意：根据寄存器章节的说明，当 TXACK 或者 ABACK 位中的任意一个位被置位时，就将 IRR8 置位。这就意味着任意邮箱发送结束或者发送被中止，并且对于下一次发送变为可更新状态。另一方面，GSR2 表示没有发生当前的发送请求（全部 TXPR 标志没有被置位）。

(2) 用于发送的内部仲裁

图 19.10 说明 RCAN-ET 如何按照信息 ID 的顺序调度发传请求的信息。在进行内部仲裁时，取出发送请求信息中的优先级最高的信息。

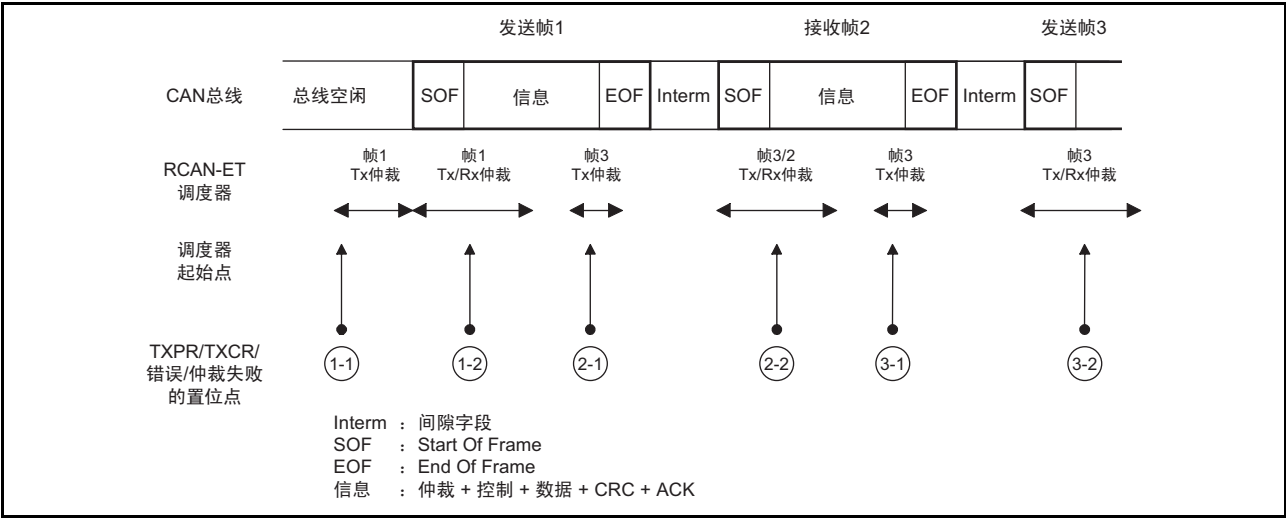


图 19.10 用于发送的内部仲裁

- 1-1 : 如果在 CAN 总线处于空闲状态时将 TXPR 位置位, 就立即进行内部仲裁, 开始发送。
- 1-2 : 在 SOF 开始发送和接收。因为不是接收帧, 所以 RCAN-ET 进行发送。
- 2-1 : 在 CRC 的定界符开始内部仲裁, 检索下一个要发送的信息。
- 2-2 : 在 SOF 开始发送和接收。因为是高优先级的接收帧, 所以 RCAN-ET 进行接收。因此, 不发送帧 3 而进行接收。
- 3-1 : 在 CRC 的定界符开始内部仲裁, 检索下一个要发送的信息。
- 3-2 : 在 SOF 开始发送和接收。因为发送帧的优先级比接收帧高, 所以 RCAN-ET 进行发送。

如果从 CAN 总线检测到错误, 就能在各错误定界符的起始位置进行下一个传送的内部仲裁, 也能在超载帧后的错误定界符的起始位置进行内部仲裁。

因为在 CRC 的定界符进行发送仲裁, 所以在 ATX=1 的邮箱接收到远程帧请求时, 根据此请求, 要发送的信息加入发送仲裁的时序为此后的总线空闲、CRC 定界符或者错误定界符。

根据 CAN 总线的状态, 在将 TXCR 置位后经过最大 1 个 CAN 帧的延迟, 进行对应信息的中止处理。

19.6.4 信息的接收顺序

信息的接收顺序如图 19.11 所示。

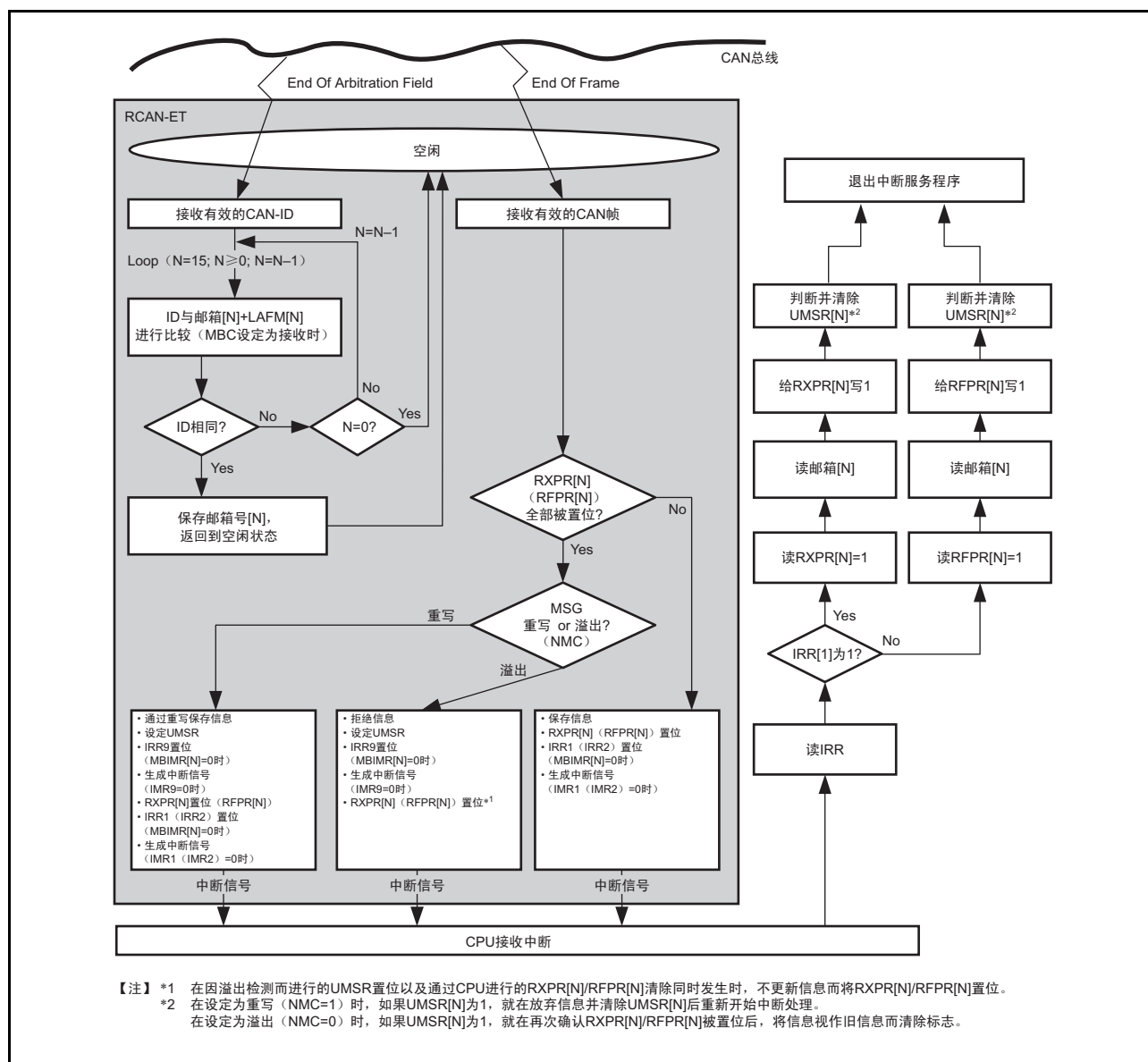


图 19.11 信息的接收顺序

如果 RCAN-ET 在接收信息中识别出仲裁字段的末尾时，就开始比较已接收的 ID 和邮箱所设定的 ID。比较顺序为邮箱 15 到邮箱 0。首先，检查 MBC，并检查是否将邮箱设定为接收。然后读 LAFM，接着读邮箱 15（设定为接收时）的 ID，与已接收的 ID 进行比较。如果不相同，就对邮箱 14（设定为接收时）进行同样的检查。如果发现相同的 ID，RCAN-ET 就将此邮箱号（N）保存到内部缓冲器，在停止检索后返回到空闲状态，等待 EOF（End Of Frame）。如果通知 EOF 的第 6 位，就通过 NMC 位的设定放弃或者写接收信息。

不能在通信中更改 RCAN-ET 的信息 ID 和 LAFM 的设定。更改设定的办法有停机模式和结构模式。如果将信息写到对应的邮箱，因为包含信息 ID，所以有可能因使用的 LAFM 而使 CAN-ID 被不同接收信息的 CAN-ID 重写。这意味着：如果接收的信息 ID 和多个邮箱的 ID+LAFM 相同，就总是将接收信息保存在邮箱号最大的邮箱，而号码小的邮箱不保存信息。因此，必须注意选择 ID 和 LAFM 的设定值。

为了在执行中断服务程序中检测信息被相同邮箱里的新信息重写的情况，在图 19.11 所示的接收数据和远

程帧时，需要在读 IRR 后清除 UMSR 标志。如果在 UMSR 的最后的检查中检测到重写时，就需要放弃信息，重新读取。

另外，在将 UMSR 置位的邮箱设定为溢出模式 (NMC=0) 时，不重写而保留信息。但是，此信息不是在 CAN 总线上监视到的最新信息。

必须在清除对应邮箱的 RXPR/RFPR 标志前读此邮箱的全部信息 (邮箱 [N] 的区域)。

必须注意：如果接收到的远程帧被数据帧重写，就将远程帧请求中断 (IRR2) 和数据帧接收中断 (IRR1) 置位，并且还将接收标志 (RXPR 和 RFPR) 置位。同样，在数据帧被远程帧重写时，将 IRR2 和 IRR1 置位。

在溢出模式 (NMC=0) 中，只有最初的邮箱才将这些标志置位。因此，在最初接收数据帧时，RXPR 和 IRR1 同时有效。此后，如果在读数据帧前接收远程帧，就不将 RFPR 和 IRR2 置位。此时，将对应邮箱的 UMSR 置位。

19.6.5 邮箱的重新设定

在需要重新设定邮箱时，必须按下述步骤进行。

(1) 变更发件箱的设定

有以下 2 种情况：

- ID、RTR、IDE、LAFM、数据、DLC、NMC、ATX、DART、MBC 的变更
只能在 MBC=B'000 时进行更改。必须确认对应的 TXPR 位没有被置位。能随时更改设定 (除 MBC 外)。
- 从发送信箱到接收信箱的设定变更
必须确认对应的 TXPR 位没有被置位。只能在停机模式或者复位状态下进行更改。当 RCAN-ET 正在接收或者发送信息时，转移到停机状态需要一定的时间 (因为在等待接收/发送结束后转移到停机状态)。另外，必须注意：在停机状态下不能接收和发送信息。
在 RCAN-ET 处于总线断开状态时，通过 MCR 寄存器的 bit6 和 bit14 的设定转移到停机状态。

(2) 接收信箱的 ID、RTR、IDE、LAFM、数据、DLC、NMC、ATX、DART、MBC 的设定变更以及从接收信箱到发送信箱的变更

只能在停机模式中更改设定。当 CAN 总线上有信息并且 RCAN-ET 为接收模式时，不会丢失此信息。RCAN-ET 在当前进行的接收结束后转移到停机模式。当 RCAN-ET 正在接收或者发送信息时，转移到停机状态需要一定时间 (因为在等待接收/发送结束后转移到停机状态)。另外，必须注意：在停机状态下不能接收和发送信息。

在 RCAN-ET 处于总线断开状态时，通过 MCR 寄存器的 bit6 和 bit14 的设定转移到停机状态。

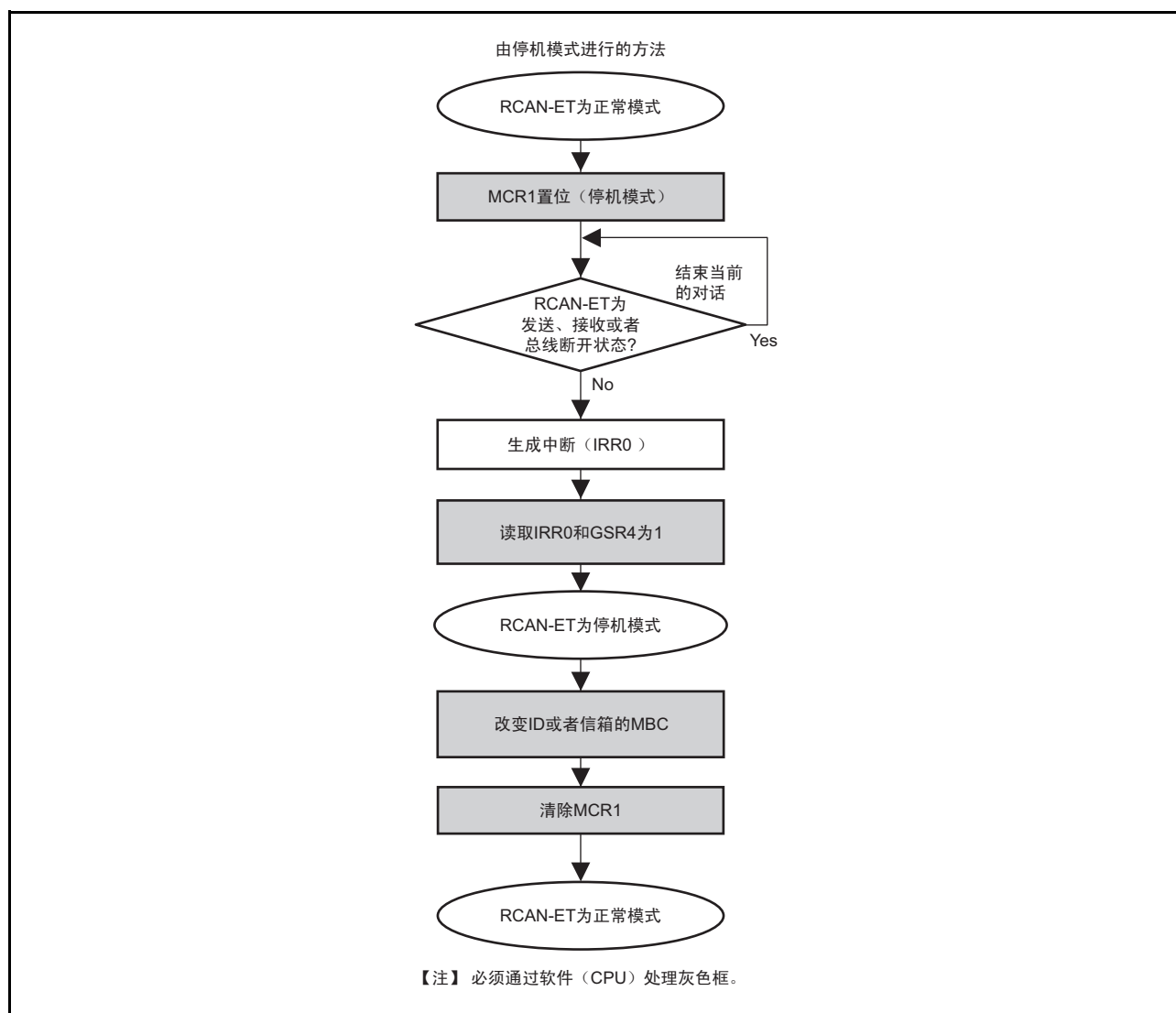


图 19.12 接收信箱的 ID 变更 / 从接收信箱更改为发送信箱

19.7 中断源

RCAN-ET 有表 19.9 所示的中断源。除了上电复位引起的复位处理中断（IRR0）以外，能屏蔽全部中断源。在屏蔽时，使用邮箱中断屏蔽寄存器 0（MBIMR0）和中断屏蔽寄存器（IMR）。有关各中断请求的中断向量，请参照“第 6 章 中断控制器（INTC）”。

表 19.9 RCAN-ET 的中断源

模块名	名称	中断源	中断标志	DTC 的启动
RCAN-ET_0	ERS_0	错误被动 (TEC $\geq$ 128 或者 REC $\geq$ 128)	IRR5	不可
		总线断开 (TEC $\geq$ 256) / 从总线断开的返回	IRR6	
		错误警告 (TEC $\geq$ 96)	IRR3	
		错误警告 (REC $\geq$ 96)	IRR4	
	OVR_0	信息错误的检测	IRR13*1	
		复位 / 停机 / CAN 睡眠转移	IRR0	
		超载帧的发送	IRR7	
		未读信息的重写 (溢出)	IRR9	
		CAN 睡眠中的 CAN 总线运行的检测	IRR12	
	RM0_0*2	数据帧的接收	IRR1*3	可 *4
	RM1_0*2	远程帧的接收	IRR2*3	
	SLE_0	信息的发送 / 取消发送 (槽空)	IRR8	不可

【注】 *1 只在测试模式中有效。

*2 RM0_0 是由邮箱 0 的远程帧接收标志 (RFPR0[0]) 或者数据帧接收标志 (RXPR0[0]) 产生的中断; RM1_0 是由邮箱 n (n=1 ~ 15) 的远程帧接收标志 (RFPR0[n]) 或者数据帧接收标志 (RXPR0[n]) 产生的中断。

*3 IRR1 是邮箱 0 ~ 15 的数据帧接收标志; IRR2 是邮箱 0 ~ 15 的远程帧接收标志。

*4 只能通过 RM0_0 中断启动 DTC。

19.8 DTC 接口

如果 RCAN-ET 的邮箱 0 接收信息, 就能启动 DTC。如果设定 DTC 启动并且 DTC 传送结束, 就自动清除 RXPR0 和 RFPR0 寄存器的标志。此时, 不会因 RCAN-ET 的接收中断而向 CPU 请求中断。DTC 的传送流程图如图 19.13 所示。

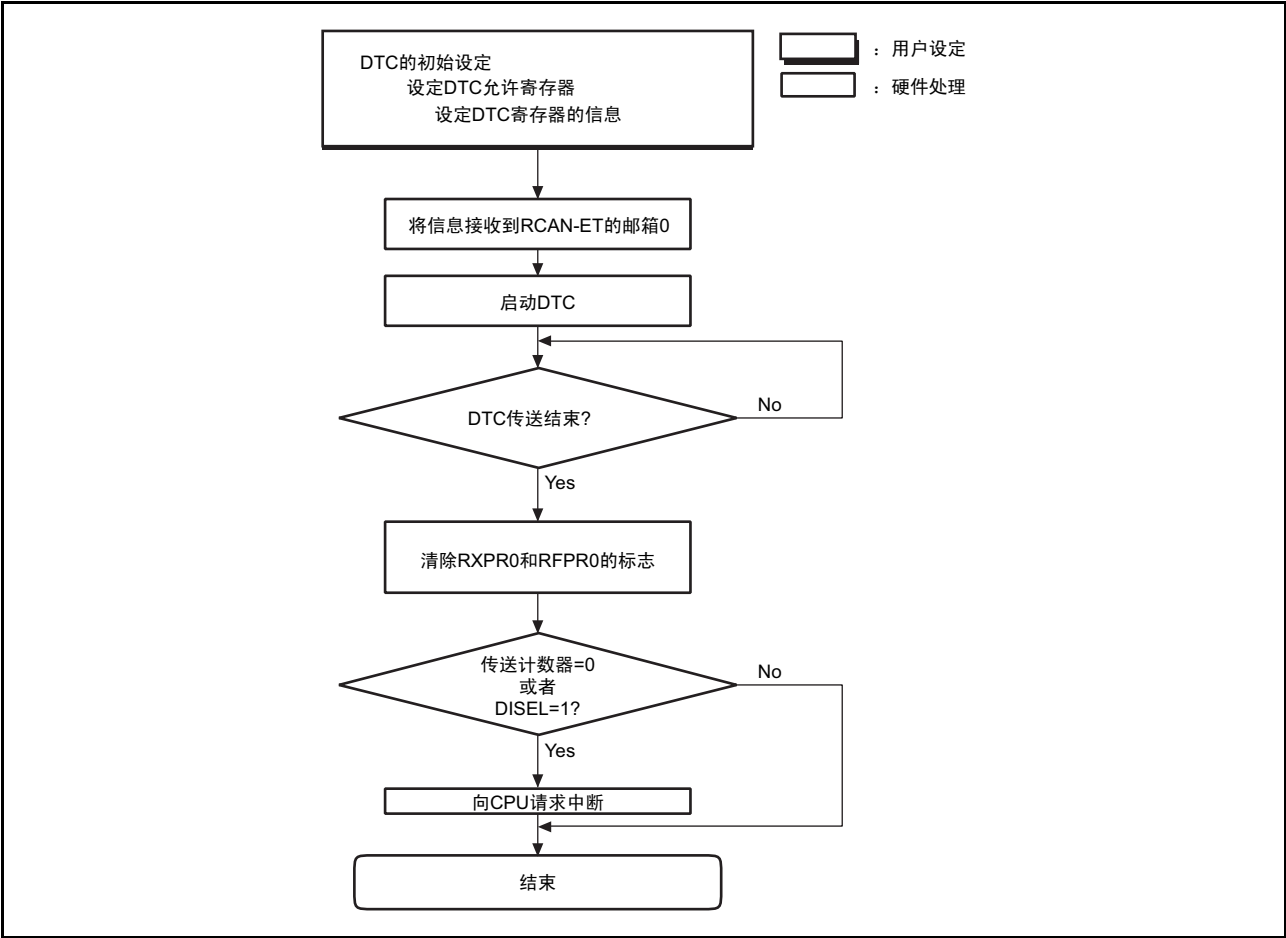


图 19.13 DTC 的传送流程图

19.9 CAN 总线接口

为了连接本 LSI 和 CAN 总线，需要总线收发器 IC。建议使用瑞萨 HA13721 收发器 IC。如果使用其他产品，就必须使用与 HA13721 兼容的产品。连接例子如图 19.14 所示。

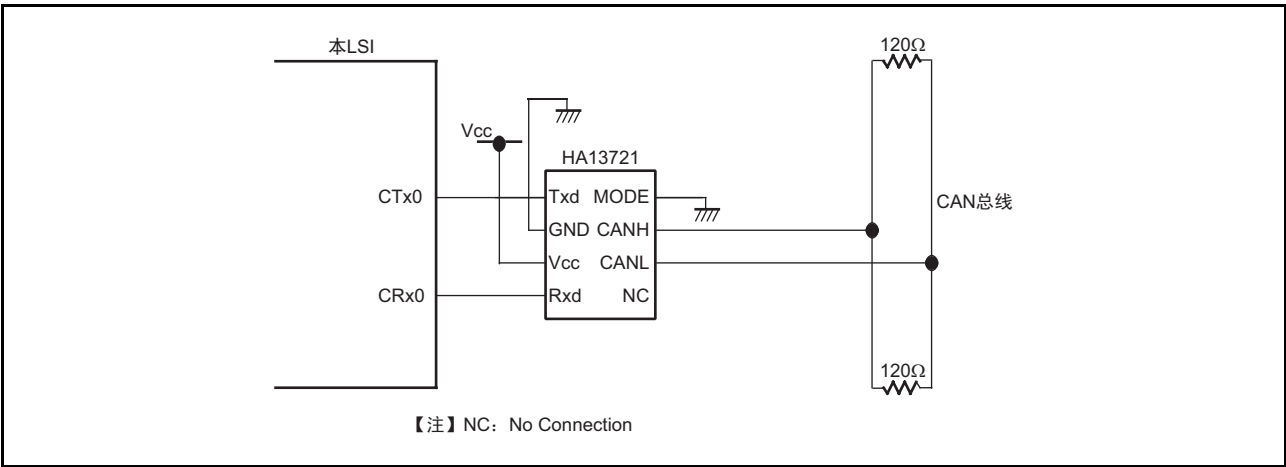


图 19.14 使用 HA13721 的高速 CAN 接口

19.10 使用时的注意事项

19.10.1 模块停止模式

通过待机控制寄存器 3 (STBCR3)，能设定 RCAN-ET 时钟的运行 / 停止。初始值为 RCAN-ET 的时钟停止运行。必须在解除模块停止模式后进行设定。

19.10.2 复位

RCAN-ET 的复位方法有硬件复位和软件复位。

- 硬件复位
通过上电复位、模块停止或者软件待机，初始化 RCAN-ET。
- 软件复位
通过主控控制寄存器 (MCR) 的 MCR0 位，初始化除 MCR0 位以外的寄存器和 CAN 通信功能。

在复位时，中断请求寄存器 (IRR) 的 IRR0 位因初始化而被置位，所以必须在复位顺序所示的结构模式中清除此位。

除了邮箱的信息控制字段 1 (CONTROL1) 以外，全部区域都由 RAM 构成，因此不能通过复位进行初始化。在上电复位后，必须在复位顺序所示的结构模式中，初始设定全部邮箱。

19.10.3 CAN 睡眠模式

在 CAN 睡眠模式中，停止模块内部的主要部分的时钟。因此，不能在 CAN 睡眠模式中存取 MCR、GSR、IRR、IMR 以外的寄存器。

19.10.4 寄存器的存取

在通过 RCAN-ET 的内部 CAN 通信功能将 CAN 总线接收帧保存到邮箱期间，如果存取邮箱区域，就要等待 0 ~ 5 个外围总线周期。

19.10.5 中断

如表 19.9 所示，能通过邮箱 0 的接收中断启动 DTC。在将邮箱 0 的接收中断设定为启动源后，如果要选择清除 DTC 传送时的中断源，就必须利用块传送模式等，读到邮箱 0 的信息控制字段 1 (CONTROL1) 为止。

第 20 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 20.1 ~ 表 20.9 所示。

按运行模式分类的引脚功能一览表如表 20.10 ~ 表 20.12 所示。

表 20.1 多路复用引脚一览表 (SH7136、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)
A	PA0 输入 / 输出 (端口)	$\overline{\text{POE0}}$ 输入 (POE)	RXD0 输入 (SCI)	—
	PA1 输入 / 输出 (端口)	$\overline{\text{POE1}}$ 输入 (POE)	TXD0 输出 (SCI)	—
	PA2 输入 / 输出 (端口)	IRQ0 输入 (INTC)	$\overline{\text{POE2}}$ 输入 (POE)	SCK0 输入 / 输出 (SCI)
	PA3 输入 / 输出 (端口)	IRQ1 输入 (INTC)	RXD1 输入 (SCI)	—
	PA4 输入 / 输出 (端口)	IRQ2 输入 (INTC)	TXD1 输出 (SCI)	—
	PA5 输入 / 输出 (端口)	IRQ3 输入 (INTC)	SCK1 输入 / 输出 (SCI)	—
	PA6 输入 / 输出 (端口)	$\overline{\text{UBCTRG}}$ 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)	TCLKC 输入 (MTU2)	$\overline{\text{POE6}}$ 输入 (POE)	RXD2 输入 (SCI)
	PA9 输入 / 输出 (端口)	TCLKD 输入 (MTU2)	$\overline{\text{POE8}}$ 输入 (POE)	TXD2 输出 (SCI)
	PA10 输入 / 输出 (端口)	RXD0 输入 (SCI)	—	—
	PA11 输入 / 输出 (端口)	TXD0 输出 (SCI)	$\overline{\text{ADTRG}}$ 输入 (A/D)	—
	PA12 输入 / 输出 (端口)	SCK0 输入 / 输出 (SCI)	$\overline{\text{SCS}}$ 输入 / 输出 (SSU)	—
	PA13 输入 / 输出 (端口)	SCK1 输入 / 输出 (SCI)	SSCK 输入 / 输出 (SSU)	—
	PA14 输入 / 输出 (端口)	RXD1 输入 (SCI)	SSI 输入 / 输出 (SSU)	—
	PA15 输入 / 输出 (端口)	TXD1 输出 (SCI)	SSO 输入 / 输出 (SSU)	—

表 20.2 多路复用引脚一览表 (SH7137、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)	$\overline{\text{IRQ0}}$ 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)	$\overline{\text{IRQ1}}$ 输入 (INTC)	RXD1 输入 (SCI)	—
	PA4 输入 / 输出 (端口)	A4 输出 (BSC)	$\overline{\text{IRQ2}}$ 输入 (INTC)	TXD1 输出 (SCI)	—
	PA5 输入 / 输出 (端口)	A5 输出 (BSC)	$\overline{\text{IRQ3}}$ 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 20.3 多路复用引脚一览表 (SH7136、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	PB2 输入 / 输出 (端口)	IRQ0 输入 (INTC)	$\overline{\text{POE0}}$ 输入 (POE)	TIC5VS 输入 (MTU2S)	SCL 输入 / 输出 (I ² C2)
	PB3 输入 / 输出 (端口)	IRQ1 输入 (INTC)	$\overline{\text{POE1}}$ 输入 (POE)	TIC5V 输入 (MTU2)	SDA 输入 / 输出 (I ² C2)
	PB4 输入 / 输出 (端口)	IRQ2 输入 (INTC)	$\overline{\text{POE4}}$ 输入 (POE)	TIC5US 输入 (MTU2S)	—
	PB5 输入 / 输出 (端口)	IRQ3 输入 (INTC)	$\overline{\text{POE5}}$ 输入 (POE)	TIC5U 输入 (MTU2)	—
	PB6 输入 / 输出 (端口)	CTx0 输出 (RCAN-ET)	—	—	—
	PB7 输入 / 输出 (端口)	CRx0 输入 (RCAN-ET)	—	—	—

表 20.4 多路复用引脚一览表 (SH7137、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)	功能 6 (相关模块)
B	PB0 输入 / 输出 (端口)	$\overline{\text{BACK}}$ 输出 (BSC)	TIC5WS 输入 (MTU2S)	—	—	—
	PB1 输入 / 输出 (端口)	$\overline{\text{BREQ}}$ 输入 (BSC)	TIC5W 输入 (MTU2)	—	—	—
	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)	SCL 输入 / 输出 (I ² C2)
	PB3 输入 / 输出 (端口)	A17 输出 (BSC)	IRQ1 输入 (INTC)	$\overline{\text{POE1}}$ 输入 (POE)	TIC5V 输入 (MTU2)	SDA 输入 / 输出 (I ² C2)
	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)	TIC5U 输入 (MTU2)	—
	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)	—	—	—

表 20.5 多路复用引脚一览表 (SH7137、端口 D)

端口	功能 1 (相关模块)	功能 2 (相关模块)	功能 3 (相关模块)	功能 4 (相关模块)
D	PD0 输入 / 输出 (端口)	D0 输入 / 输出 (BSC)	RXD0 输入 (SCI)	—
	PD1 输入 / 输出 (端口)	D1 输入 / 输出 (BSC)	TXD0 输出 (SCI)	—
	PD2 输入 / 输出 (端口)	D2 输入 / 输出 (BSC)	SCK0 输入 / 输出 (SCI)	—
	PD3 输入 / 输出 (端口)	D3 输入 / 输出 (BSC)	RXD1 输入 (SCI)	—
	PD4 输入 / 输出 (端口)	D4 输入 / 输出 (BSC)	TXD1 输出 (SCI)	—
	PD5 输入 / 输出 (端口)	D5 输入 / 输出 (BSC)	SCK1 输入 / 输出 (SCI)	—
	PD6 输入 / 输出 (端口)	D6 输入 / 输出 (BSC)	RXD2 输入 (SCI)	—
	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)
	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 20.6 多路复用引脚一览表 (SH7136、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)
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)	TIOC1A 输入 / 输出 (MTU2)	RXD1 输入 (SCI)	—
	PE5 输入 / 输出 (端口)	TIOC1B 输入 / 输出 (MTU2)	TXD1 输出 (SCI)	—
	PE6 输入 / 输出 (端口)	TIOC2A 输入 / 输出 (MTU2)	SCK1 输入 / 输出 (SCI)	—
	PE7 输入 / 输出 (端口)	TIOC2B 输入 / 输出 (MTU2)	—	—
	PE8 输入 / 输出 (端口)	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)	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)	TIOC3BS 输入 / 输出 (MTU2S)	$\overline{\text{ASEBRKAK}}$ 输出 (E10A)	$\overline{\text{ASEBRK}}$ 输入 (E10A)
	PE17 输入 / 输出 (端口)	TIOC3DS 输入 / 输出 (MTU2S)	TCK 输入 (H-UDI)	—
	PE18 输入 / 输出 (端口)	TIOC4AS 输入 / 输出 (MTU2S)	TDI 输入 (H-UDI)	—
	PE19 输入 / 输出 (端口)	TIOC4BS 输入 / 输出 (MTU2S)	TDO 输出 (H-UDI)	—
	PE20 输入 / 输出 (端口)	TIOC4CS 输入 / 输出 (MTU2S)	TMS 输入 (H-UDI)	—
	PE21 输入 / 输出 (端口)	TIOC4DS 输入 / 输出 (MTU2S)	$\overline{\text{TRST}}$ 输入 (H-UDI)	—

表 20.7 多路复用引脚一览表 (SH7137、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)	$\overline{\text{ASEBRKAK}}$ 输出 (E10A)	$\overline{\text{ASEBRK}}$ 输入 (E10A)
	PE17 输入 / 输出 (端口)	$\overline{\text{CS0}}$ 输出 (BSC)	TIOC3DS 输入 / 输出 (MTU2S)	TCK 输入 (H-UDI)	—
	PE18 输入 / 输出 (端口)	$\overline{\text{CS1}}$ 输出 (BSC)	TIOC4AS 输入 / 输出 (MTU2S)	TDI 输入 (H-UDI)	—
	PE19 输入 / 输出 (端口)	$\overline{\text{RD}}$ 输出 (BSC)	TIOC4BS 输入 / 输出 (MTU2S)	TDO 输出 (H-UDI)	—
	PE20 输入 / 输出 (端口)	TIOC4CS 输入 / 输出 (MTU2S)	TMS 输入 (H-UDI)	—	—
	PE21 输入 / 输出 (端口)	$\overline{\text{WRL}}$ 输出 (BSC)	TIOC4DS 输入 / 输出 (MTU2S)	$\overline{\text{TRST}}$ 输入 (H-UDI)	—

表 20.8 多路复用引脚一览表 (SH7136、端口 F)

端口	功能 1 (相关模块)	功能 2 (相关模块)
F	PF0 输入 (端口)	AN0 输入 (A/D)
	PF1 输入 (端口)	AN1 输入 (A/D)
	PF2 输入 (端口)	AN2 输入 (A/D)
	PF3 输入 (端口)	AN3 输入 (A/D)
	PF8 输入 (端口)	AN8 输入 (A/D)
	PF9 输入 (端口)	AN9 输入 (A/D)
	PF10 输入 (端口)	AN10 输入 (A/D)
	PF11 输入 (端口)	AN11 输入 (A/D)
	PF12 输入 (端口)	AN12 输入 (A/D)
	PF13 输入 (端口)	AN13 输入 (A/D)
	PF14 输入 (端口)	AN14 输入 (A/D)
	PF15 输入 (端口)	AN15 输入 (A/D)

【注】 在 A/D 转换中 AN 输入功能有效。

表 20.9 多路复用引脚一览表 (SH7137、端口 F)

端口	功能 1 (相关模块)	功能 2 (相关模块)
F	PF0 输入 (端口)	AN0 输入 (A/D)
	PF1 输入 (端口)	AN1 输入 (A/D)
	PF2 输入 (端口)	AN2 输入 (A/D)
	PF3 输入 (端口)	AN3 输入 (A/D)
	PF4 输入 (端口)	AN4 输入 (A/D)
	PF5 输入 (端口)	AN5 输入 (A/D)
	PF6 输入 (端口)	AN6 输入 (A/D)
	PF7 输入 (端口)	AN7 输入 (A/D)
	PF8 输入 (端口)	AN8 输入 (A/D)
	PF9 输入 (端口)	AN9 输入 (A/D)
	PF10 输入 (端口)	AN10 输入 (A/D)
	PF11 输入 (端口)	AN11 输入 (A/D)
	PF12 输入 (端口)	AN12 输入 (A/D)
	PF13 输入 (端口)	AN13 输入 (A/D)
	PF14 输入 (端口)	AN14 输入 (A/D)
	PF15 输入 (端口)	AN15 输入 (A/D)

【注】 在 A/D 转换中 AN 输入功能有效。

表 20.10 按运行模式分类的引脚功能一览表 (SH7136)

引脚号	引脚名	
	单芯片模式 (MCU 模式 3)	
	初始功能	PFC 能设定的功能
10,27,44, 57,79	V _{CC}	V _{CC}
1,13,29,49	V _{SS}	V _{SS}
15,46	V _{CL}	V _{CL}
78	AV _{CC}	AV _{CC}
63	AV _{SS}	AV _{SS}
60	PLL _{VSS}	PLL _{VSS}
68	AV _{refh}	AV _{refh}
73	AV _{refl}	AV _{refl}
56	EXTAL	EXTAL
55	XTAL	XTAL
62	MD1	MD1
59	FWE*	FWE*
54	$\overline{\text{RES}}$	$\overline{\text{RES}}$
80	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$
58	NMI	NMI
61	$\overline{\text{ASEMD0}}$ *	$\overline{\text{ASEMD0}}$ *
53	PA0	PA0/ $\overline{\text{POE0}}$ /RXD0
52	PA1	PA1/ $\overline{\text{POE1}}$ /TXD0
51	PA2	PA2/ $\overline{\text{IRQ0}}$ / $\overline{\text{POE2}}$ /SCK0
50	PA3	PA3/ $\overline{\text{IRQ1}}$ /RXD1
48	PA4	PA4/ $\overline{\text{IRQ2}}$ /TXD1
47	PA5	PA5/ $\overline{\text{IRQ3}}$ /SCK1
45	PA6	PA6/ $\overline{\text{UBCTRG}}$ / $\overline{\text{TCLKA}}$ / $\overline{\text{POE4}}$
43	PA7	PA7/ $\overline{\text{TCLKB}}$ / $\overline{\text{POE5}}$ /SCK2
42	PA8	PA8/ $\overline{\text{TCLKC}}$ / $\overline{\text{POE6}}$ /RXD2
41	PA9	PA9/ $\overline{\text{TCLKD}}$ / $\overline{\text{POE8}}$ /TXD2
40	PA10	PA10/RXD0
39	PA11	PA11/TXD0/ $\overline{\text{ADTRG}}$
38	PA12	PA12/SCK0/ $\overline{\text{SCS}}$
37	PA13	PA13/SCK1/SSCK
36	PA14	PA14/RXD1/SSI
35	PA15	PA15/TXD1/SSO
34	PB2	PB2/ $\overline{\text{IRQ0}}$ / $\overline{\text{POE0}}$ /TIC5VS/SCL
33	PB3	PB3/ $\overline{\text{IRQ1}}$ / $\overline{\text{POE1}}$ /TIC5V/SDA
32	PB4	PB4/ $\overline{\text{IRQ2}}$ / $\overline{\text{POE4}}$ /TIC5US
31	PB5	PB5/ $\overline{\text{IRQ3}}$ / $\overline{\text{POE5}}$ /TIC5U
30	PB6	PB6/CTx0
28	PB7	PB7/CRx0

引脚号	引脚名	
	单芯片模式 (MCU 模式 3)	
	初始功能	PFC 能设定的功能
26	PE0	PE0/TIOC0A
25	PE1	PE1/TIOC0B/RXD0
24	PE2	PE2/TIOC0C/TXD0
23	PE3	PE3/TIOC0D/SCK0
22	PE4	PE4/TIOC1A/RXD1
21	PE5	PE5/TIOC1B/TXD1
20	PE6	PE6/TIOC2A/SCK1
19	PE7	PE7/TIOC2B
18	PE8	PE8/TIOC3A
16	PE9	PE9/TIOC3B
17	PE10	PE10/TIOC3C
14	PE11	PE11/TIOC3D
12	PE12	PE12/TIOC4A
11	PE13	PE13/TIOC4B/MRES
9	PE14	PE14/TIOC4C
8	PE15	PE15/TIOC4D/IRQOUT
7	PE16/ (ASEBRKAK/ASEBRK*)	PE16/TIOC3BS
6	PE17/ (TCK*)	PE17/TIOC3DS
5	PE18/ (TDI*)	PE18/TIOC4AS
4	PE19/ (TDO*)	PE19/TIOC4BS
3	PE20/ (TMS*)	PE20/TIOC4CS
2	PE21/ (TRST*)	PE21/TIOC4DS
77	PF0/AN0	PF0/AN0
76	PF1/AN1	PF1/AN1
75	PF2/AN2	PF2/AN2
74	PF3/AN3	PF3/AN3
72	PF8/AN8	PF8/AN8
71	PF9/AN9	PF9/AN9
70	PF10/AN10	PF10/AN10
69	PF11/AN11	PF11/AN11
67	PF12/AN12	PF12/AN12
66	PF13/AN13	PF13/AN13
65	PF14/AN14	PF14/AN14
64	PF15/AN15	PF15/AN15

【注】 * 当使用 E10A 时 (ASEMD0=L 时)，这些引脚固定为 TMS、TRST、TDI、TDO、TCK、ASEBRKAK/ASEBRK。

表 20.11 按运行模式分类的引脚功能一览表 (SH7137 (1))

引脚号	引脚名	
	内部 ROM 无效 (MCU 模式 0)	
	初始功能	PFC 能设定的功能
3,11,36, 48,57,99	V _{CC}	V _{CC}
1,14,39, 50,64	V _{SS}	V _{SS}
16,59	V _{CL}	V _{CL}
98	AV _{CC}	AV _{CC}
79	AV _{SS}	AV _{SS}
88	AV _{refh}	AV _{refh}
93	AV _{refl}	AV _{refl}
75	PLL _{VSS}	PLL _{VSS}
72	EXTAL	EXTAL
71	XTAL	XTAL
78	MD0	MD0
77	MD1	MD1
74	FWE*	FWE*
70	$\overline{\text{RES}}$	$\overline{\text{RES}}$
100	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$
73	NMI	NMI
76	$\overline{\text{ASEMD0}}$ *	$\overline{\text{ASEMD0}}$ *
69	A0	PA0/A0/ $\overline{\text{POE0}}$ /RXD0
68	A1	PA1/A1/ $\overline{\text{POE1}}$ /TXD0
67	A2	PA2/A2/IRQ0/ $\overline{\text{POE2}}$ /SCK0
66	A3	PA3/A3/IRQ1/RXD1
65	A4	PA4/A4/IRQ2/TXD1
63	A5	PA5/A5/IRQ3/SCK1
62	$\overline{\text{RD}}$	PA6/ $\overline{\text{RD}}$ / $\overline{\text{UBCTRG}}$ / $\overline{\text{TCLKA}}$ / $\overline{\text{POE4}}$
61	PA7	PA7/ $\overline{\text{TCLKB}}$ / $\overline{\text{POE5}}$ /SCK2
60	$\overline{\text{WRL}}$	PA8/ $\overline{\text{WRL}}$ / $\overline{\text{TCLKC}}$ / $\overline{\text{POE6}}$ /RXD2
58	PA9	PA9/ $\overline{\text{WAIT}}$ / $\overline{\text{TCLKD}}$ / $\overline{\text{POE8}}$ /TXD2
56	A6	PA10/A6/RXD0
55	A7	PA11/A7/TXD0/ $\overline{\text{ADTRG}}$
54	A8	PA12/A8/SCK0/ $\overline{\text{SCS}}$
53	A9	PA13/A9/SCK1/SSCK
52	A10	PA14/A10/RXD1/SSI
51	CK	PA15/CK/TXD1/SSO
49	PB0	PB0/ $\overline{\text{BACK}}$ /TIC5WS
47	PB1	PB1/ $\overline{\text{BREQ}}$ /TIC5W
46	A16	PB2/A16/IRQ0/ $\overline{\text{POE0}}$ /TIC5VS/SCL
45	A17	PB3/A17/IRQ1/ $\overline{\text{POE1}}$ /TIC5V/SDA

引脚号	引脚名	
	内部 ROM 无效 (MCU 模式 0)	
	初始功能	PFC 能设定的功能
44	PB4	PB4/A18/IRQ2/ $\overline{\text{POE4}}$ /TIC5US
43	PB5	PB5/A19/IRQ3/ $\overline{\text{POE5}}$ /TIC5U
42	PB6	PB6/ $\overline{\text{WAIT}}$ /CTx0
41	PB7	PB7/ $\overline{\text{CS1}}$ /CRx0
40	D0	PD0/D0/RXD0
38	D1	PD1/D1/TXD0
37	D2	PD2/D2/SCK0
35	D3	PD3/D3/RXD1
34	D4	PD4/D4/TXD1
33	D5	PD5/D5/SCK1
32	D6	PD6/D6/RXD2
31	D7	PD7/D7/TXD2/ $\overline{\text{SCS}}$
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/ $\overline{\text{IRQOUT}}$
8	PE16/ ($\overline{\text{ASEBRKAK}}/\overline{\text{ASEBRK}}^*$)	PE16/ $\overline{\text{WAIT}}$ /TIOC3BS
7	PE17/ (TCK^*)	PE17/ $\overline{\text{CS0}}$ /TIOC3DS
6	PE18/ (TDI^*)	PE18/ $\overline{\text{CS1}}$ /TIOC4AS
5	PE19/ (TDO^*)	PE19/ $\overline{\text{RD}}$ /TIOC4BS
4	PE20/ (TMS^*)	PE20/TIOC4CS
2	PE21/ ($\overline{\text{TRST}}^*$)	PE21/ $\overline{\text{WRL}}$ /TIOC4DS
97	PF0/AN0	PF0/AN0
96	PF1/AN1	PF1/AN1

引脚号	引脚名	
	内部 ROM 无效 (MCU 模式 0)	
	初始功能	PFC 能设定的功能
95	PF2/AN2	PF2/AN2
94	PF3/AN3	PF3/AN3
92	PF4/AN4	PF4/AN4
91	PF5/AN5	PF5/AN5
90	PF6/AN6	PF6/AN6
89	PF7/AN7	PF7/AN7
87	PF8/AN8	PF8/AN8
86	PF9/AN9	PF9/AN9
85	PF10/AN10	PF10/AN10
84	PF11/AN11	PF11/AN11
83	PF12/AN12	PF12/AN12
82	PF13/AN13	PF13/AN13
81	PF14/AN14	PF14/AN14
80	PF15/AN15	PF15/AN15

【注】 * 当使用 E10A 时 ($\overline{\text{ASEMD0}}=\text{L}$ 时), 这些引脚固定为 TMS、 $\overline{\text{TRST}}$ 、TDI、TDO、TCK、 $\overline{\text{ASEBRKAK}}/\overline{\text{ASEBRK}}$ 。

表 20.12 按运行模式分类的引脚功能一览表 (SH7137 (2))

引脚号	引脚名			
	内部 ROM 有效 (MCU 模式 2)		单芯片模式 (MCU 模式 3)	
	初始功能	PFC 能设定的功能	初始功能	PFC 能设定的功能
3,11,36,48,57,99	V_{CC}	V_{CC}	V_{CC}	V_{CC}
1,14,39,50,64	V_{SS}	V_{SS}	V_{SS}	V_{SS}
16,59	V_{CL}	V_{CL}	V_{CL}	V_{CL}
98	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}
79	AV_{SS}	AV_{SS}	AV_{SS}	AV_{SS}
88	AV_{refh}	AV_{refh}	AV_{refh}	AV_{refh}
93	AV_{refl}	AV_{refl}	AV_{refl}	AV_{refl}
75	PLL V_{SS}	PLL V_{SS}	PLL V_{SS}	PLL V_{SS}
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	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$
100	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$
73	NMI	NMI	NMI	NMI
76	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$

引脚号	引脚名			
	内部 ROM 有效 (MCU 模式 2)		单芯片模式 (MCU 模式 3)	
	初始功能	PFC 能设定的功能	初始功能	PFC 能设定的功能
69	PA0	PA0/A0/ $\overline{\text{POE0}}$ /RXD0	PA0	PA0/ $\overline{\text{POE0}}$ /RXD0
68	PA1	PA1/A1/ $\overline{\text{POE1}}$ /TXD0	PA1	PA1/ $\overline{\text{POE1}}$ /TXD0
67	PA2	PA2/A2/ $\overline{\text{IRQ0}}$ / $\overline{\text{POE2}}$ /SCK0	PA2	PA2/ $\overline{\text{IRQ0}}$ / $\overline{\text{POE2}}$ /SCK0
66	PA3	PA3/A3/ $\overline{\text{IRQ1}}$ /RXD1	PA3	PA3/ $\overline{\text{IRQ1}}$ /RXD1
65	PA4	PA4/A4/ $\overline{\text{IRQ2}}$ /TXD1	PA4	PA4/ $\overline{\text{IRQ2}}$ /TXD1
63	PA5	PA5/A5/ $\overline{\text{IRQ3}}$ /SCK1	PA5	PA5/ $\overline{\text{IRQ3}}$ /SCK1
62	PA6	PA6/ $\overline{\text{RD}}$ / $\overline{\text{UBCTRG}}$ / $\overline{\text{TCLKA}}$ / $\overline{\text{POE4}}$	PA6	PA6/ $\overline{\text{UBCTRG}}$ / $\overline{\text{TCLKA}}$ / $\overline{\text{POE4}}$
61	PA7	PA7/ $\overline{\text{TCLKB}}$ / $\overline{\text{POE5}}$ /SCK2	PA7	PA7/ $\overline{\text{TCLKB}}$ / $\overline{\text{POE5}}$ /SCK2
60	PA8	PA8/ $\overline{\text{WRL}}$ / $\overline{\text{TCLKC}}$ / $\overline{\text{POE6}}$ /RXD2	PA8	PA8/ $\overline{\text{TCLKC}}$ / $\overline{\text{POE6}}$ /RXD2
58	PA9	PA9/ $\overline{\text{WAIT}}$ / $\overline{\text{TCLKD}}$ / $\overline{\text{POE8}}$ /TXD2	PA9	PA9/ $\overline{\text{TCLKD}}$ / $\overline{\text{POE8}}$ /TXD2
56	PA10	PA10/A6/RXD0	PA10	PA10/RXD0
55	PA11	PA11/A7/TXD0/ $\overline{\text{ADTRG}}$	PA11	PA11/TXD0/ $\overline{\text{ADTRG}}$
54	PA12	PA12/A8/SCK0/ $\overline{\text{SCS}}$	PA12	PA12/SCK0/ $\overline{\text{SCS}}$
53	PA13	PA13/A9/SCK1/SSCK	PA13	PA13/SCK1/SSCK
52	PA14	PA14/A10/RXD1/SSI	PA14	PA14/RXD1/SSI
51	CK	PA15/CK/TXD1/SSO	PA15	PA15/TXD1/SSO
49	PB0	PB0/ $\overline{\text{BACK}}$ /TIC5WS	PB0	PB0/TIC5WS
47	PB1	PB1/ $\overline{\text{BREQ}}$ /TIC5W	PB1	PB1/TIC5W
46	PB2	PB2/A16/ $\overline{\text{IRQ0}}$ / $\overline{\text{POE0}}$ /TIC5VS/SCL	PB2	PB2/ $\overline{\text{IRQ0}}$ / $\overline{\text{POE0}}$ /TIC5VS/SCL
45	PB3	PB3/A17/ $\overline{\text{IRQ1}}$ / $\overline{\text{POE1}}$ /TIC5V/SDA	PB3	PB3/ $\overline{\text{IRQ1}}$ / $\overline{\text{POE1}}$ /TIC5V/SDA
44	PB4	PB4/A18/ $\overline{\text{IRQ2}}$ / $\overline{\text{POE4}}$ /TIC5US	PB4	PB4/ $\overline{\text{IRQ2}}$ / $\overline{\text{POE4}}$ /TIC5US
43	PB5	PB5/A19/ $\overline{\text{IRQ3}}$ / $\overline{\text{POE5}}$ /TIC5U	PB5	PB5/ $\overline{\text{IRQ3}}$ / $\overline{\text{POE5}}$ /TIC5U
42	PB6	PB6/ $\overline{\text{WAIT}}$ /CTx0	PB6	PB6/CTx0
41	PB7	PB7/ $\overline{\text{CS1}}$ /CRx0	PB7	PB7/CRx0
40	PD0	PD0/D0/RXD0	PD0	PD0/RXD0
38	PD1	PD1/D1/TXD0	PD1	PD1/TXD0
37	PD2	PD2/D2/SCK0	PD2	PD2/SCK0
35	PD3	PD3/D3/RXD1	PD3	PD3/RXD1
34	PD4	PD4/D4/TXD1	PD4	PD4/TXD1
33	PD5	PD5/D5/SCK1	PD5	PD5/SCK1
32	PD6	PD6/D6/RXD2	PD6	PD6/RXD2
31	PD7	PD7/D7/TXD2/ $\overline{\text{SCS}}$	PD7	PD7/TXD2/ $\overline{\text{SCS}}$
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

引脚号	引脚名			
	内部 ROM 有效 (MCU 模式 2)		单芯片模式 (MCU 模式 3)	
	初始功能	PFC 能设定的功能	初始功能	PFC 能设定的功能
23	PE4	PE4/A11/TIOC1A/RXD1	PE4	PE4/TIOC1A/RXD1
22	PE5	PE5/A12/TIOC1B/TXD1	PE5	PE5/TIOC1B/TXD1
21	PE6	PE6/A13/TIOC2A/SCK1	PE6	PE6/TIOC2A/SCK1
20	PE7	PE7/A14/TIOC2B	PE7	PE7/TIOC2B
19	PE8	PE8/A15/TIOC3A	PE8	PE8/TIOC3A
17	PE9	PE9/TIOC3B	PE9	PE9/TIOC3B
18	PE10	PE10/ $\overline{\text{CS0}}$ /TIOC3C	PE10	PE10/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/ $\overline{\text{MRES}}$
10	PE14	PE14/TIOC4C	PE14	PE14/TIOC4C
9	PE15	PE15/TIOC4D/ $\overline{\text{IRQOUT}}$	PE15	PE15/TIOC4D/ $\overline{\text{IRQOUT}}$
8	PE16/ ($\overline{\text{ASEBRKAK}}$ / $\overline{\text{ASEBRK*}}$)	PE16/ $\overline{\text{WAIT}}$ /TIOC3BS	PE16/ ($\overline{\text{ASEBRKAK}}$ / $\overline{\text{ASEBRK*}}$)	PE16/TIOC3BS
7	PE17/ ($\overline{\text{TCK*}}$)	PE17/ $\overline{\text{CS0}}$ /TIOC3DS	PE17/ ($\overline{\text{TCK*}}$)	PE17/TIOC3DS
6	PE18/ ($\overline{\text{TDI*}}$)	PE18/ $\overline{\text{CS1}}$ /TIOC4AS	PE18/ ($\overline{\text{TDI*}}$)	PE18/TIOC4AS
5	PE19/ ($\overline{\text{TDO*}}$)	PE19/ $\overline{\text{RD}}$ /TIOC4BS	PE19/ ($\overline{\text{TDO*}}$)	PE19/TIOC4BS
4	PE20/ ($\overline{\text{TMS*}}$)	PE20/TIOC4CS	PE20/ ($\overline{\text{TMS*}}$)	PE20/TIOC4CS
2	PE21/ ($\overline{\text{TRST*}}$)	PE21/ $\overline{\text{WRL}}$ /TIOC4DS	PE21/ ($\overline{\text{TRST*}}$)	PE21/TIOC4DS
97	PF0/AN0	PF0/AN0	PF0/AN0	PF0/AN0
96	PF1/AN1	PF1/AN1	PF1/AN1	PF1/AN1
95	PF2/AN2	PF2/AN2	PF2/AN2	PF2/AN2
94	PF3/AN3	PF3/AN3	PF3/AN3	PF3/AN3
92	PF4/AN4	PF4/AN4	PF4/AN4	PF4/AN4
91	PF5/AN5	PF5/AN5	PF5/AN5	PF5/AN5
90	PF6/AN6	PF6/AN6	PF6/AN6	PF6/AN6
89	PF7/AN7	PF7/AN7	PF7/AN7	PF7/AN7
87	PF8/AN8	PF8/AN8	PF8/AN8	PF8/AN8
86	PF9/AN9	PF9/AN9	PF9/AN9	PF9/AN9
85	PF10/AN10	PF10/AN10	PF10/AN10	PF10/AN10
84	PF11/AN11	PF11/AN11	PF11/AN11	PF11/AN11
83	PF12/AN12	PF12/AN12	PF12/AN12	PF12/AN12
82	PF13/AN13	PF13/AN13	PF13/AN13	PF13/AN13
81	PF14/AN14	PF14/AN14	PF14/AN14	PF14/AN14
80	PF15/AN15	PF15/AN15	PF15/AN15	PF15/AN15

【注】 * 当使用 E10A 时 (ASEMD0=L 时), 这些引脚固定为 TMS、TRST、TDI、TDO、TCK、 $\overline{\text{ASEBRKAK}}$ 、 $\overline{\text{ASEBRK}}$ 。

20.1 寄存器说明

PFC 有以下寄存器，有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 25 章 寄存器一览”。

表 20.13 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
端口 A 的 IO 寄存器 L	PAIORL	R/W	H'0000	H'FFFFD106	8、16
端口 A 的控制寄存器 L4	PACRL4	R/W	H'0000*	H'FFFFD110	8、16、32
端口 A 的控制寄存器 L3	PACRL3	R/W	H'0000*	H'FFFFD112	8、16
端口 A 的控制寄存器 L2	PACRL2	R/W	H'0000*	H'FFFFD114	8、16、32
端口 A 的控制寄存器 L1	PACRL1	R/W	H'0000*	H'FFFFD116	8、16
端口 B 的 IO 寄存器 L	PBIORL	R/W	H'0000	H'FFFFD186	8、16
端口 B 的控制寄存器 L2	PBCRL2	R/W	H'0000	H'FFFFD194	8、16、32
端口 B 的控制寄存器 L1	PBCRL1	R/W	H'0000*	H'FFFFD196	8、16
端口 D 的 IO 寄存器 L	PDIORL	R/W	H'0000	H'FFFFD286	8、16
端口 D 的控制寄存器 L3	PDCRL3	R/W	H'0000	H'FFFFD292	8、16
端口 D 的控制寄存器 L2	PDCRL2	R/W	H'0000*	H'FFFFD294	8、16、32
端口 D 的控制寄存器 L1	PDCRL1	R/W	H'0000*	H'FFFFD296	8、16
端口 E 的 IO 寄存器 H	PEIORH	R/W	H'0000	H'FFFFD304	8、16、32
端口 E 的 IO 寄存器 L	PEIORL	R/W	H'0000	H'FFFFD306	8、16
端口 E 的控制寄存器 H2	PECRH2	R/W	H'0000	H'FFFFD30C	8、16、32
端口 E 的控制寄存器 H1	PECRH1	R/W	H'0000	H'FFFFD30E	8、16
端口 E 的控制寄存器 L4	PECRL4	R/W	H'0000	H'FFFFD310	8、16、32
端口 E 的控制寄存器 L3	PECRL3	R/W	H'0000*	H'FFFFD312	8、16
端口 E 的控制寄存器 L2	PECRL2	R/W	H'0000*	H'FFFFD314	8、16、32
端口 E 的控制寄存器 L1	PECRL1	R/W	H'0000	H'FFFFD316	8、16
IRQOUT 功能的控制寄存器	IFCR	R/W	H'0000	H'FFFFD322	8、16

【注】 * 初始值在 SH7137 的内部 ROM 有效 / 无效的外部扩展模式时不同，有关详细内容请参照本章的各寄存器说明。

20.1.1 端口 A 的 IO 寄存器 L (PAIORL)

PAIORL 是 16 位可读写寄存器，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（省略引脚名中端口以外的多路复用引脚名）。PAIORL 在端口 A 的引脚功能为通用输入 / 输出（PA15 ~ PA0）时有效，否则无效。

如果将 PAIORL 的位置 1，对应的引脚就为输出引脚：如果置 0 就为输入引脚。

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

20.1.2 端口 A 的控制寄存器 L1 ~ L4 (PACRL1 ~ PACRL4)

PACRL1 ~ PACRL4 是 16 位可读写寄存器, 选择端口 A 的多路复用引脚功能。

(1) SH7136

- 端口 A 的控制寄存器 L4 (PACRL4)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PA15 MD2	PA15 MD1	PA15 MD0	—	PA14 MD2	PA14 MD1	PA14 MD0	—	PA13 MD2	PA13 MD1	PA13 MD0	—	PA12 MD2	PA12 MD1	PA12 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	PA15MD2	0	R/W	PA15 模式位 选择 PA15/TXD1/SSO 引脚的功能。 000: PA15 输入 / 输出 (端口) 101: SSO 输入 / 输出 (SSU) 110: TXD1 输出 (SCI) 上述以外: 禁止设定
13	PA15MD1	0	R/W	
12	PA15MD0	0	R/W	
11	—	0	R	保留位 读写值总是 0。
10	PA14MD2	0	R/W	PA14 模式位 选择 PA14/RXD1/SSI 引脚的功能。 000: PA14 输入 / 输出 (端口) 101: SSI 输入 / 输出 (SSU) 110: RXD1 输入 (SCI) 上述以外: 禁止设定
9	PA14MD1	0	R/W	
8	PA14MD0	0	R/W	
7	—	0	R	保留位 读写值总是 0。
6	PA13MD2	0	R/W	PA13 模式位 选择 PA13/SCK1/SSCK 引脚的功能。 000: PA13 输入 / 输出 (端口) 101: SSCK 输入 / 输出 (SSU) 110: SCK1 输入 / 输出 (SCI) 上述以外: 禁止设定
5	PA13MD1	0	R/W	
4	PA13MD0	0	R/W	
3	—	0	R	保留位 读写值总是 0。
2	PA12MD2	0	R/W	PA12 模式位 选择 PA12/SCK0/SCS 引脚的功能。 000: PA12 输入 / 输出 (端口) 101: SCS 输入 / 输出 (SSU) 110: SCK0 输入 / 输出 (SCI) 上述以外: 禁止设定
1	PA12MD1	0	R/W	
0	PA12MD0	0	R/W	

• 端口A的控制寄存器L3 (PACRL3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PA11 MD2	PA11 MD1	PA11 MD0	—	PA10 MD2	PA10 MD1	PA10 MD0	—	PA9 MD2	PA9 MD1	PA9 MD0	—	PA8 MD2	PA8 MD1	PA8 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	PA11MD2	0	R/W	PA11 模式位 选择 PA11/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) 110: TXD0 输出 (SCI) 上述以外: 禁止设定
13	PA11MD1	0	R/W	
12	PA11MD0	0	R/W	
11	—	0	R	保留位 读写值总是 0。
10	PA10MD2	0	R/W	PA10 模式位 选择 PA10/RXD0 引脚的功能。 000: PA10 输入 / 输出 (端口) 110: RXD0 输入 (SCI) 上述以外: 禁止设定
9	PA10MD1	0	R/W	
8	PA10MD0	0	R/W	
7	—	0	R	保留位 读写值总是 0。
6	PA9MD2	0	R/W	PA9 模式位 选择 PA9/TCLKD/TXD2 引脚的功能。 000: PA9 输入 / 输出 (端口) 001: TCLKD 输入 (MTU2) 110: TXD2 输出 (SCI) 111: $\overline{\text{POE8}}$ 输入 (POE) 上述以外: 禁止设定
5	PA9MD1	0	R/W	
4	PA9MD0	0	R/W	
3	—	0	R	保留位 读写值总是 0。
2	PA8MD2	0	R/W	PA8 模式位 选择 PA8/TCLKC/ $\overline{\text{POE6}}$ /RXD2 引脚的功能。 000: PA8 输入 / 输出 (端口) 001: TCLKC 输入 (MTU2) 110: RXD2 输入 (SCI) 111: $\overline{\text{POE6}}$ 输入 (POE) 上述以外: 禁止设定
1	PA8MD1	0	R/W	
0	PA8MD0	0	R/W	

• 端口A的控制寄存器L2 (PACRL2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PA7 MD2	PA7 MD1	PA7 MD0	—	PA6 MD2	PA6 MD1	PA6 MD0	—	PA5 MD2	PA5 MD1	PA5 MD0	—	PA4 MD2	PA4 MD1	PA4 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	PA7MD2	0	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
13	PA7MD1	0	R/W	
12	PA7MD0	0	R/W	
11	—	0	R	保留位 读写值总是 0。
10	PA6MD2	0	R/W	PA6 模式位 选择 PA6/ $\overline{\text{UBCTRG}}$ /TCLKA/ $\overline{\text{POE4}}$ 引脚的功能。 000: PA6 输入 / 输出 (端口) 001: TCLKA 输入 (MTU2) 101: $\overline{\text{UBCTRG}}$ 输出 (UBC) 111: $\overline{\text{POE4}}$ 输入 (POE) 上述以外: 禁止设定
9	PA6MD1	0	R/W	
8	PA6MD0	0	R/W	
7	—	0	R	保留位 读写值总是 0。
6	PA5MD2	0	R/W	PA5 模式位 选择 PA5/IRQ3/SCK1 引脚的功能。 000: PA5 输入 / 输出 (端口) 001: SCK1 输入 / 输出 (SCI) 111: IRQ3 输入 (INTC) 上述以外: 禁止设定
5	PA5MD1	0	R/W	
4	PA5MD0	0	R/W	
3	—	0	R	保留位 读写值总是 0。
2	PA4MD2	0	R/W	PA4 模式位 选择 PA4/IRQ2/TXD1 引脚的功能。 000: PA4 输入 / 输出 (端口) 001: TXD1 输出 (SCI) 111: IRQ2 输入 (INTC) 上述以外: 禁止设定
1	PA4MD1	0	R/W	
0	PA4MD0	0	R/W	

• 端口A的控制寄存器L1 (PACRL1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PA3 MD2	PA3 MD1	PA3 MD0	—	PA2 MD2	PA2 MD1	PA2 MD0	—	PA1 MD2	PA1 MD1	PA1 MD0	—	PA0 MD2	PA0 MD1	PA0 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	PA3MD2	0	R/W	PA3 模式位 选择 PA3/IRQ1/RXD1 引脚的功能。 000: PA3 输入 / 输出 (端口) 001: RXD1 输入 (SCI) 111: IRQ1 输入 (INTC) 上述以外: 禁止设定
13	PA3MD1	0	R/W	
12	PA3MD0	0	R/W	
11	—	0	R	保留位 读写值总是 0。
10	PA2MD2	0	R/W	PA2 模式位 选择 PA2/IRQ0/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) 111: POE2 输入 (POE) 上述以外: 禁止设定
9	PA2MD1	0	R/W	
8	PA2MD0	0	R/W	
7	—	0	R	保留位 读写值总是 0。
6	PA1MD2	0	R/W	PA1 模式位 选择 PA1/POE1/TXD0 引脚的功能。 000: PA1 输入 / 输出 (端口) 001: TXD0 输出 (SCI) 111: POE1 输入 (POE) 上述以外: 禁止设定
5	PA1MD1	0	R/W	
4	PA1MD0	0	R/W	
3	—	0	R	保留位 读写值总是 0。
2	PA0MD2	0	R/W	PA0 模式位 选择 PA0/POE0/RXD0 引脚的功能。 000: PA0 输入 / 输出 (端口) 001: RXD0 输入 (SCI) 111: POE0 输入 (POE) 上述以外: 禁止设定
1	PA0MD1	0	R/W	
0	PA0MD0	0	R/W	

(2) SH7137

• 端口A的控制寄存器L4 (PACRL4)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PA15 MD2	PA15 MD1	PA15 MD0	—	PA14 MD2	PA14 MD1	PA14 MD0	—	PA13 MD2	PA13 MD1	PA13 MD0	—	PA12 MD2	PA12 MD1	PA12 MD0
初始值:	0	0	0	0*1	0	0*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/无效的外部扩展模式中，初始值为1。

*2 在内部ROM无效的外部扩展模式中，初始值为1。

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	PA15MD2	0	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
13	PA15MD1	0	R/W	
12	PA15MD0	0*1	R/W	
11	—	0	R	保留位 读写值总是 0。
10	PA14MD2	0*2	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
9	PA14MD1	0	R/W	
8	PA14MD0	0	R/W	
7	—	0	R	保留位 读写值总是 0。
6	PA13MD2	0*2	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
5	PA13MD1	0	R/W	
4	PA13MD0	0	R/W	
3	—	0	R	保留位 读写值总是 0。
2	PA12MD2	0*2	R/W	PA12 模式位 选择 PA12/A8/SCK0/SCS 引脚的功能。 000: PA12 输入 / 输出 (端口) 100: A8 输出 (BSC) *3 101: SCS 输入 / 输出 (SSU) 110: SCK0 输入 / 输出 (SCI) 上述以外: 禁止设定
1	PA12MD1	0	R/W	
0	PA12MD0	0	R/W	

【注】 *1 在内部 ROM 有效 / 无效的外部扩展模式时，初始值为 1。

*2 在内部 ROM 无效的外部扩展模式时，初始值为 1。

*3 只在内部 ROM 有效 / 无效的外部扩展模式中有效。在单芯片模式时不能设定。

• 端口 A 的控制寄存器 L3 (PACRL3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PA11 MD2	PA11 MD1	PA11 MD0	—	PA10 MD2	PA10 MD1	PA10 MD0	—	PA9 MD2	PA9 MD1	PA9 MD0	—	PA8 MD2	PA8 MD1	PA8 MD0
初始值:	0	0*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 无效的外部扩展模式中，初始值为 1。

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	PA11MD2	0*1	R/W	PA11 模式位 选择 PA11/A7/TXD0/ADTRG 引脚的功能。 000: PA11 输入 / 输出 (端口) 010: ADTRG 输入 (A/D) 100: A7 输出 (BSC) *2 110: TXD0 输出 (SCI) 上述以外: 禁止设定
13	PA11MD1	0	R/W	
12	PA11MD0	0	R/W	
11	—	0	R	保留位 读写值总是 0。
10	PA10MD2	0*1	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
9	PA10MD1	0	R/W	
8	PA10MD0	0	R/W	
7	—	0	R	保留位 读写值总是 0。
6	PA9MD2	0	R/W	PA9 模式位 选择 PA9/WAIT/TCLKD/TXD2 引脚的功能。 000: PA9 输入 / 输出 (端口) 001: TCLKD 输入 (MTU2) 100: WAIT 输入 (BSC) *2 110: TXD2 输出 (SCI) 111: POE8 输入 (POE) 上述以外: 禁止设定
5	PA9MD1	0	R/W	
4	PA9MD0	0	R/W	
3	—	0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
2	PA8MD2	0*1	R/W	PA8 模式位 选择 PA8/WRL/TCLKC/POE6/RXD2 引脚的功能。 000: PA8 输入 / 输出 (端口) 001: TCLKC 输入 (MTU2) 100: WRL 输出 (BSC) *2 110: RXD2 输入 (SCI) 111: POE6 输入 (POE) 上述以外: 禁止设定
1	PA8MD1	0	R/W	
0	PA8MD0	0	R/W	

【注】 *1 在内部 ROM 无效的外部扩展模式时, 初始值为 1。

*2 只在内部 ROM 有效 / 无效的外部扩展模式中有效。在单芯片模式时不能设定。

• 端口 A 的控制寄存器 L2 (PACRL2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PA7 MD2	PA7 MD1	PA7 MD0	—	PA6 MD2	PA6 MD1	PA6 MD0	—	PA5 MD2	PA5 MD1	PA5 MD0	—	PA4 MD2	PA4 MD1	PA4 MD0
初始值:	0	0	0	0	0	0	0*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 无效的外部扩展模式中, 初始值为 1。

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	PA7MD2	0	R/W	PA7 模式位 选择 PA7/TCLKB/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: POE5 输入 (POE) 上述以外: 禁止设定
13	PA7MD1	0	R/W	
12	PA7MD0	0	R/W	
11	—	0	R	保留位 读写值总是 0。
10	PA6MD2	0	R/W	PA6 模式位 选择 PA6/RD/UBCTRG/TCLKA/POE4 引脚的功能。 000: PA6 输入 / 输出 (端口) 001: TCLKA 输入 (MTU2) 011: RD 输出 (BSC) *2 101: UBCTRG 输出 (UBC) 111: POE4 输入 (POE) 上述以外: 禁止设定
9	PA6MD1	0*1	R/W	
8	PA6MD0	0*1	R/W	
7	—	0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
6	PA5MD2	0*1	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
5	PA5MD1	0	R/W	
4	PA5MD0	0	R/W	
3	—	0	R	
2	PA4MD2	0*1	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
1	PA4MD1	0	R/W	
0	PA4MD0	0	R/W	

【注】 *1 在内部 ROM 无效的外部扩展模式时, 初始值为 1。

*2 只在内部 ROM 有效 / 无效的外部扩展模式中有效。在单芯片模式时不能设定。

• 端口 A 的控制寄存器 L1 (PACRL1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PA3 MD2	PA3 MD1	PA3 MD0	—	PA2 MD2	PA2 MD1	PA2 MD0	—	PA1 MD2	PA1 MD1	PA1 MD0	—	PA0 MD2	PA0 MD1	PA0 MD0
初始值:	0	0*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 无效的外部扩展模式中, 初始值为 1。

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	PA3MD2	0*1	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
13	PA3MD1	0	R/W	
12	PA3MD0	0	R/W	
11	—	0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
10	PA2MD2	0*1	R/W	PA2 模式位 选择 PA2/A2/IRQ0/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: POE2 输入 (POE) 上述以外: 禁止设定
9	PA2MD1	0	R/W	
8	PA2MD0	0	R/W	
7	—	0	R	
6	PA1MD2	0*1	R/W	PA1 模式位 选择 PA1/A1/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: POE1 输入 (POE) 上述以外: 禁止设定
5	PA1MD1	0	R/W	
4	PA1MD0	0	R/W	
3	—	0	R	
2	PA0MD2	0*1	R/W	PA0 模式位 选择 PA0/A0/POE0/RXD0 引脚的功能。 000: PA0 输入 / 输出 (端口) 001: RXD0 输入 (SCI) 100: A0 输出 (BSC) *2 111: POE0 输入 (POE) 上述以外: 禁止设定
1	PA0MD1	0	R/W	
0	PA0MD0	0	R/W	

【注】 *1 在内部 ROM 无效的外部扩展模式时, 初始值为 1。

*2 只在内部 ROM 有效 / 无效的外部扩展模式中有效。在单芯片模式时不能设定。

20.1.3 端口 B 的 IO 寄存器 L (PBIORL)

PBIORL 是 16 位可读写寄存器, 选择端口 B 的引脚输入 / 输出方向。PB7IOR ~ PB0IOR 位分别对应 PB7 引脚 ~ PB0 引脚 (省略引脚名中端口以外的多路复用引脚名)。PBIORL 在端口 B 的引脚功能为通用输入 / 输出 (PB7 ~ PB0) 时有效, 否则无效。

如果将 PBIORL 的位置 1, 对应的引脚就为输出引脚; 如果置 0 就为输入引脚。

但是, 在 SH7136 中, PBIORL 的 bit1 和 bit0 无效。

PBIORL 的 bit15 ~ 8 为保留位, 读写值总是 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

20.1.4 端口 B 的控制寄存器 L1、L2 (PBCRL1、PBCRL2)

PBCRL1 和 PBCRL2 是 16 位可读写寄存器，选择端口 B 的多路复用引脚功能。

(1) SH7136

• 端口 B 的控制寄存器 L2 (PBCRL2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PB7 MD2	PB7 MD1	PB7 MD0	—	PB6 MD2	PB6 MD1	PB6 MD0	—	PB5 MD2	PB5 MD1	PB5 MD0	—	PB4 MD2	PB4 MD1	PB4 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	PB7MD2	0	R/W	PB7 模式位 选择 PB7/CRx0 引脚的功能。 000: PB7 输入 / 输出 (端口) 110: CRx0 输入 (RCAN-ET) 上述以外: 禁止设定
13	PB7MD1	0	R/W	
12	PB7MD0	0	R/W	
11	—	0	R	保留位 读写值总是 0。
10	PB6MD2	0	R/W	PB6 模式位 选择 PB6/CTx0 引脚的功能。 000: PB6 输入 / 输出 (端口) 110: CTx0 输出 (RCAN-ET) 上述以外: 禁止设定
9	PB6MD1	0	R/W	
8	PB6MD0	0	R/W	
7	—	0	R	保留位 读写值总是 0。
6	PB5MD2	0	R/W	PB5 模式位 选择 PB5/IRQ3/POE5/TIC5U 引脚的功能。 000: PB5 输入 / 输出 (端口) 001: IRQ3 输入 (INTC) 011: TIC5U 输入 (MTU2) 111: POE5 输入 (POE) 上述以外: 禁止设定
5	PB5MD1	0	R/W	
4	PB5MD0	0	R/W	
3	—	0	R	保留位 读写值总是 0。
2	PB4MD2	0	R/W	PB4 模式位 选择 PB4/IRQ2/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) 111: POE4 输入 (POE) 上述以外: 禁止设定
1	PB4MD1	0	R/W	
0	PB4MD0	0	R/W	

- 端口 B 的控制寄存器 L1 (PBCRL1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PB3 MD2	PB3 MD1	PB3 MD0	—	PB2 MD2	PB2 MD1	PB2 MD0	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	PB3MD2	0	R/W	PB3 模式位 选择 PB3/IRQ1/ $\overline{\text{POE1}}$ /TIC5V/SDA 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) 011: TIC5V 输入 (MTU2) 100: SDA 输入 / 输出 (IIC2) 上述以外: 禁止设定
13	PB3MD1	0	R/W	
12	PB3MD0	0	R/W	
11	—	0	R	保留位 读写值总是 0。
10	PB2MD2	0	R/W	PB2 模式位 选择 PB2/IRQ0/ $\overline{\text{POE0}}$ /TIC5VS/SCL 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) 100: SCL 输入 / 输出 (IIC2) 上述以外: 禁止设定
9	PB2MD1	0	R/W	
8	PB2MD0	0	R/W	
7 ~ 0	—	全 0	R	保留位 读写值总是 0。

(2) SH7137

• 端口B的控制寄存器L2 (PBCRL2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PB7 MD2	PB7 MD1	PB7 MD0	—	PB6 MD2	PB6 MD1	PB6 MD0	—	PB5 MD2	PB5 MD1	PB5 MD0	—	PB4 MD2	PB4 MD1	PB4 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	PB7MD2	0	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
13	PB7MD1	0	R/W	
12	PB7MD0	0	R/W	
11	—	0	R	保留位 读写值总是 0。
10	PB6MD2	0	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
9	PB6MD1	0	R/W	
8	PB6MD0	0	R/W	
7	—	0	R	保留位 读写值总是 0。
6	PB5MD2	0	R/W	PB5 模式位 选择 PB5/A19/ $\overline{\text{IRQ3}}$ / $\overline{\text{POE5}}$ /TIC5U 引脚的功能。 000: PB5 输入 / 输出 (端口) 001: IRQ3 输入 (INTC) 011: TIC5U 输入 (MTU2) 101: A19 输出 (BSC) * 111: $\overline{\text{POE5}}$ 输入 (POE) 上述以外: 禁止设定
5	PB5MD1	0	R/W	
4	PB5MD0	0	R/W	
3	—	0	R	保留位 读写值总是 0。
2	PB4MD2	0	R/W	PB4 模式位 选择 PB4/A18/ $\overline{\text{IRQ2}}$ / $\overline{\text{POE4}}$ /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
1	PB4MD1	0	R/W	
0	PB4MD0	0	R/W	

【注】 * 只在内部 ROM 有效 / 无效的外部扩展模式中有效。在单芯片模式时不能设定。

• 端口 B 的控制寄存器 L1 (PBCRL1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PB3 MD2	PB3 MD1	PB3 MD0	—	PB2 MD2	PB2 MD1	PB2 MD0	—	PB1 MD2	PB1 MD1	PB1 MD0	—	PB0 MD2	PB0 MD1	PB0 MD0
初始值:	0	0*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 无效的外部扩展模式中，初始值为 1。

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	PB3MD2	0*1	R/W	PB3 模式位 选择 PB3/A17/IRQ1/ $\overline{\text{POE1}}$ /TIC5V/SDA 引脚的功能。 000: PB3 输入 / 输出 (端口) 001: $\overline{\text{IRQ1}}$ 输入 (INTC) 010: $\overline{\text{POE1}}$ 输入 (POE) 011: TIC5V 输入 (MTU2) 100: SDA 输入 / 输出 (IIC2) 101: A17 输出 (BSC) *2 上述以外: 禁止设定
13	PB3MD1	0	R/W	
12	PB3MD0	0*1	R/W	
11	—	0	R	保留位 读写值总是 0。
10	PB2MD2	0*1	R/W	PB2 模式位 选择 PB2/A16/IRQ0/ $\overline{\text{POE0}}$ /TIC5VS/SCL 引脚的功能。 000: PB2 输入 / 输出 (端口) 001: $\overline{\text{IRQ0}}$ 输入 (INTC) 010: $\overline{\text{POE0}}$ 输入 (POE) 011: TIC5VS 输入 (MTU2S) 100: SCL 输入 / 输出 (IIC2) 101: A16 输出 (BSC) *2 上述以外: 禁止设定
9	PB2MD1	0	R/W	
8	PB2MD0	0*1	R/W	
7	—	0	R	保留位 读写值总是 0。
6	PB1MD2	0	R/W	PB1 模式位 选择 PB1/ $\overline{\text{BREQ}}$ /TIC5W 引脚的功能。 000: PB1 输入 / 输出 (端口) 011: TIC5W 输入 (MTU2) 101: $\overline{\text{BREQ}}$ 输入 (BSC) *2 上述以外: 禁止设定
5	PB1MD1	0	R/W	
4	PB1MD0	0	R/W	
3	—	0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
2	PB0MD2	0	R/W	PB0 模式位 选择 PB0/BACK/TIC5WS 引脚的功能。 000: PB0 输入 / 输出 (端口) 011: TIC5WS 输入 (MTU2S) 101: BACK 输出 (BSC) *2 上述以外: 禁止设定
1	PB0MD1	0	R/W	
0	PB0MD0	0	R/W	

【注】 *1 在内部 ROM 无效的外部扩展模式时, 初始值为 1。

*2 只在内部 ROM 有效 / 无效的外部扩展模式中有效。在单芯片模式时不能设定。

20.1.5 端口 D 的 IO 寄存器 L (PDIORL) (只限 SH7137)

PDIORL 是 16 位可读写寄存器, 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 (省略引脚名中端口以外的多路复用引脚名)。PDIORL 在端口 D 的引脚功能为通用输入 / 输出 (PD10 ~ PD0) 时有效, 否则无效。

如果将 PDIORL 的位置 1, 对应的引脚就为输出引脚; 如果置 0 就为输入引脚。

但是, 在 SH7136 中, PDIORL 无效。

PDIORL 的 bit15 ~ 11 为保留位, 读写值总是 0。

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

20.1.6 端口 D 的控制寄存器 L1 ~ L3 (PDCRL1 ~ PDCRL3) (只限 SH7137)

PDCRL1 ~ PDCRL3 是 16 位可读写寄存器, 选择端口 D 的多路复用引脚功能。但是, 在 SH7136 中, PDCRL1 ~ PDCRL3 无效。

• 端口 D 的控制寄存器 L3 (PDCRL3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	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	初始值	R/W	说明
15 ~ 11	—	全 0	R	保留位 读写值总是 0。
10	PD10MD2	0	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
9	PD10MD1	0	R/W	
8	PD10MD0	0	R/W	
7	—	0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
6	PD9MD2	0	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
5	PD9MD1	0	R/W	
4	PD9MD0	0	R/W	
3	—	0	R	保留位 读写值总是 0。
2	PD8MD2	0	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
1	PD8MD1	0	R/W	
0	PD8MD0	0	R/W	

• 端口 D 的控制寄存器 L2 (PDCRL2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PD7 MD2	PD7 MD1	PD7 MD0	—	PD6 MD2	PD6 MD1	PD6 MD0	—	PD5 MD2	PD5 MD1	PD5 MD0	—	PD4 MD2	PD4 MD1	PD4 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

【注】*1 在内部ROM无效的外部扩展模式中, 初始值为1。

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	PD7MD2	0	R/W	PD7 模式位 选择 PD7/D7/SCS/TXD2 引脚的功能。 000: PD7 输入 / 输出 (端口) 001: D7 输入 / 输出 (BSC) *2 101: SCS 输入 / 输出 (SSU) 110: TXD2 输出 (SCI) 上述以外: 禁止设定
13	PD7MD1	0	R/W	
12	PD7MD0	0*1	R/W	
11	—	0	R	保留位 读写值总是 0。
10	PD6MD2	0	R/W	PD6 模式位 选择 PD6/D6/RXD2 引脚的功能。 000: PD6 输入 / 输出 (端口) 001: D6 输入 / 输出 (BSC) *2 110: RXD2 输入 (SCI) 上述以外: 禁止设定
9	PD6MD1	0	R/W	
8	PD6MD0	0*1	R/W	
7	—	0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
6	PD5MD2	0	R/W	PD5 模式位 选择 PD5/D5/SCK1 引脚的功能。 000: PD5 输入 / 输出 (端口) 001: D5 输入 / 输出 (BSC) *2 110: SCK1 输入 / 输出 (SCI) 上述以外: 禁止设定
5	PD5MD1	0	R/W	
4	PD5MD0	0*1	R/W	
3	—	0	R	保留位 读写值总是 0。
2	PD4MD2	0	R/W	PD4 模式位 选择 PD4/D4/TXD1 引脚的功能。 000: PD4 输入 / 输出 (端口) 001: D4 输入 / 输出 (BSC) *2 110: TXD1 输出 (SCI) 上述以外: 禁止设定
1	PD4MD1	0	R/W	
0	PD4MD0	0*1	R/W	

【注】 *1 在内部 ROM 无效的外部扩展模式时, 初始值为 1。

*2 只在内部 ROM 有效 / 无效的外部扩展模式中有效。在单芯片模式时不能设定。

• 端口 D 的控制寄存器 L1 (PDCRL1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PD3 MD2	PD3 MD1	PD3 MD0	—	PD2 MD2	PD2 MD1	PD2 MD0	—	PD1 MD2	PD1 MD1	PD1 MD0	—	PD0 MD2	PD0 MD1	PD0 MD0
初始值:	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1	0	0	0	0*1
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

【注】 *1 在内部 ROM 无效的外部扩展模式中, 初始值为 1。

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	PD3MD2	0	R/W	PD3 模式位 选择 PD3/D3/RXD1 引脚的功能。 000: PD3 输入 / 输出 (端口) 001: D3 输入 / 输出 (BSC) *2 110: RXD1 输入 (SCI) 上述以外: 禁止设定
13	PD3MD1	0	R/W	
12	PD3MD0	0*1	R/W	
11	—	0	R	保留位 读写值总是 0。
10	PD2MD2	0	R/W	PD2 模式位 选择 PD2/D2/SCK0 引脚的功能。 000: PD2 输入 / 输出 (端口) 001: D2 输入 / 输出 (BSC) *2 110: SCK0 输入 / 输出 (SCI) 上述以外: 禁止设定
9	PD2MD1	0	R/W	
8	PD2MD0	0*1	R/W	
7	—	0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
6	PD1MD2	0	R/W	PD1 模式位 选择 PD1/D1/TXD0 引脚的功能。 000: PD1 输入 / 输出 (端口) 001: D1 输入 / 输出 (BSC) *2 110: TXD0 输出 (SCI) 上述以外: 禁止设定
5	PD1MD1	0	R/W	
4	PD1MD0	0*1	R/W	
3	—	0	R	保留位 读写值总是 0。
2	PD0MD2	0	R/W	PD0 模式位 选择 PD0/D0/RXD0 引脚的功能。 000: PD0 输入 / 输出 (端口) 001: D0 输入 / 输出 (BSC) *2 110: RXD0 输入 (SCI) 上述以外: 禁止设定
1	PD0MD1	0	R/W	
0	PD0MD0	0*1	R/W	

【注】 *1 在内部 ROM 无效的外部扩展模式时, 初始值为 1。

*2 只在内部 ROM 有效 / 无效的外部扩展模式中有效。在单芯片模式时不能设定。

20.1.7 端口 E 的 IO 寄存器 L、H (PEIORL 和 PEIORH)

PEIORL 和 PEIORH 是 16 位可读写寄存器, 选择端口 E 的引脚输入 / 输出方向。PE21IOR ~ PE0IOR 位分别对应 PE21 引脚 ~ PE0 引脚 (省略引脚名的端口以外的多路复用引脚名)。PEIORL 在端口 E 的引脚功能为通用输入 / 输出 (PE15 ~ PE0) 以及 MTU2 的 TIOC 输入 / 输出时有效, 否则无效; PEIORH 在端口 E 的引脚功能为通用输入 / 输出 (PE21 ~ PE16) 以及 MTU2S 的 TIOC 输入 / 输出时有效, 否则无效。

如果将 PEIORL 和 PEIORH 的位置 1, 对应的引脚就为输出引脚; 如果置 0 就为输入引脚。

PEIORH 的 bit15 ~ 6 为保留位, 读写值总是 0。

PEIORL 和 PEIORH 的初始值都为 H'0000。

(1) 端口 E 的 IO 寄存器 H (PEIORH)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	PE21 IOR	PE20 IOR	PE19 IOR	PE18 IOR	PE17 IOR	PE16 IOR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

(2) 端口 E 的 IO 寄存器 L (PEIORL)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15 IOR	PE14 IOR	PE13 IOR	PE12 IOR	PE11 IOR	PE10 IOR	PE9 IOR	PE8 IOR	PE7 IOR	PE6 IOR	PE5 IOR	PE4 IOR	PE3 IOR	PE2 IOR	PE1 IOR	PE0 IOR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

20.1.8 端口 E 的控制寄存器 L1 ~ L4、H1、H2 (PECRL1 ~ PECRL4、PECRH1 和 PECRH2)

PECRL1 ~ PECRL4、PECRH1 和 PECRH2 是 16 位可读写寄存器，选择端口 E 的多路复用引脚功能。

(1) SH7136

- 端口 E 的控制寄存器 H2 (PECRH2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	PE21 MD1	PE21 MD0	—	—	PE20 MD1	PE20 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 6	—	全 0	R	保留位 读写值总是 0。
5 4	PE21MD1 PE21MD0	0 0	R/W R/W	PE21 模式位 选择 PE21/TIOC4DS/TRST 引脚的功能。当使用 E10A ($\overline{\text{ASEMD0}}=\text{L}$ 电平) 时，此引脚固定为 $\overline{\text{TRST}}$ 输入。 00: PE21 输入 / 输出 (端口) 01: TIOC4DS 输入 / 输出 (MTU2S) 上述以外: 禁止设定
3、2	—	全 0	R	保留位 读写值总是 0。
1 0	PE20MD1 PE20MD0	0 0	R/W R/W	PE20 模式位 选择 PE20/TIOC4CS/TMS 引脚的功能。当使用 E10A ($\overline{\text{ASEMD0}}=\text{L}$ 电平) 时，此引脚固定为 TMS 输入。 00: PE20 输入 / 输出 (端口) 01: TIOC4CS 输入 / 输出 (MTU2S) 上述以外: 禁止设定

- 端口 E 的控制寄存器 H1 (PECRH1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PE19 MD1	PE19 MD0	—	—	PE18 MD1	PE18 MD0	—	—	PE17 MD1	PE17 MD0	—	PE16 MD2	PE16 MD1	PE16 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15、14	—	全 0	R	保留位 读写值总是 0。
13 12	PE19MD1 PE19MD0	0 0	R/W R/W	PE19 模式位 选择 PE19/TIOC4BS/TDO 引脚的功能。当使用 E10A ($\overline{\text{ASEMD0}}=\text{L}$ 电平) 时，此引脚固定为 TDO 输出。 00: PE19 输入 / 输出 (端口) 01: TIOC4BS 输入 / 输出 (MTU2S) 上述以外: 禁止设定

位	位名	初始值	R/W	说明
11、10	—	全 0	R	保留位 读写值总是 0。
9 8	PE18MD1 PE18MD0	0 0	R/W R/W	PE18 模式位 选择 PE18/TIOC4AS/TDI 引脚的功能。当使用 E10A ($\overline{\text{ASEMD0}}=\text{L}$ 电平) 时, 此引脚固定为 TDI 输入。 00: PE18 输入 / 输出 (端口) 01: TIOC4AS 输入 / 输出 (MTU2S) 上述以外: 禁止设定
7、6	—	全 0	R	保留位 读写值总是 0。
5 4	PE17MD1 PE17MD0	0 0	R/W R/W	PE17 模式位 选择 PE17/TIOC3DS/TCK 引脚的功能。当使用 E10A ($\overline{\text{ASEMD0}}=\text{L}$ 电平) 时, 此引脚固定为 TCK 输入。 00: PE17 输入 / 输出 (端口) 01: TIOC3DS 输入 / 输出 (MTU2S) 上述以外: 禁止设定
3	—	0	R	保留位 读写值总是 0。
2 1 0	PE16MD2 PE16MD1 PE16MD0	0 0 0	R/W R/W R/W	PE16 模式位 选择 PE16/TIOC3BS/ $\overline{\text{ASEBRKAK}}/\overline{\text{ASEBRK}}$ 引脚的功能。当使用 E10A ($\overline{\text{ASEMD0}}=\text{L}$ 电平) 时, 此引脚固定为 $\overline{\text{ASEBRKAK}}$ 输出 / $\overline{\text{ASEBRK}}$ 输入。 000: PE16 输入 / 输出 (端口) 001: TIOC3BS 输入 / 输出 (MTU2S) 上述以外: 禁止设定

• 端口 E 的控制寄存器 L4 (PECRL4)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PE15MD2	PE15MD1	PE15MD0	—	PE14MD2	PE14MD1	PE14MD0	—	—	PE13MD1	PE13MD0	—	PE12MD2	PE12MD1	PE12MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 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 读写值总是 0。

位	位名	初始值	R/W	说明
10	PE14MD2	0	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
9	PE14MD1	0	R/W	
8	PE14MD0	0	R/W	
7、6	—	全 0	R	保留位 读写值总是 0。
5	PE13MD1	0	R/W	PE13 模式位 选择 PE13/TIOC4B/MRES 引脚的功能。 00: PE13 输入 / 输出 (端口) 01: TIOC4B 输入 / 输出 (MTU2) 10: MRES 输入 (INTC) 上述以外: 禁止设定
4	PE13MD0	0	R/W	
3	—	0	R	保留位 读写值总是 0。
2	PE12MD2	0	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
1	PE12MD1	0	R/W	
0	PE12MD0	0	R/W	

• 端口 E 的控制寄存器 L3 (PECRL3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PE11 MD2	PE11 MD1	PE11 MD0	—	PE10 MD2	PE10 MD1	PE10 MD0	—	PE9 MD2	PE9 MD1	PE9 MD0	—	PE8 MD2	PE8 MD1	PE8 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	PE11MD2	0	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
13	PE11MD1	0	R/W	
12	PE11MD0	0	R/W	
11	—	0	R	保留位 读写值总是 0。
10	PE10MD2	0	R/W	PE10 模式位 选择 PE10/TIOC3C 引脚的功能。 000: PE10 输入 / 输出 (端口) 001: TIOC3C 输入 / 输出 (MTU2) 上述以外: 禁止设定
9	PE10MD1	0	R/W	
8	PE10MD0	0	R/W	
7	—	0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
6	PE9MD2	0	R/W	PE9 模式位 选择 PE9/TIOC3B 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
5	PE9MD1	0	R/W	
4	PE9MD0	0	R/W	
3	—	0	R	保留位 读写值总是 0。
2	PE8MD2	0	R/W	PE8 模式位 选择 PE8/TIOC3A 引脚的功能。 000: PE8 输入 / 输出 (端口) 001: TIOC3A 输入 / 输出 (MTU2) 上述以外: 禁止设定
1	PE8MD1	0	R/W	
0	PE8MD0	0	R/W	

• 端口 E 的控制寄存器 L2 (PECRL2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PE7 MD2	PE7 MD1	PE7 MD0	—	PE6 MD2	PE6 MD1	PE6 MD0	—	PE5 MD2	PE5 MD1	PE5 MD0	—	PE4 MD2	PE4 MD1	PE4 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	PE7MD2	0	R/W	PE7 模式位 选择 PE7/TIOC2B 引脚的功能。 000: PE7 输入 / 输出 (端口) 001: TIOC2B 输入 / 输出 (MTU2) 上述以外: 禁止设定
13	PE7MD1	0	R/W	
12	PE7MD0	0	R/W	
11	—	0	R	保留位 读写值总是 0。
10	PE6MD2	0	R/W	PE6 模式位 选择 PE6/TIOC2A/SCK1 引脚的功能。 000: PE6 输入 / 输出 (端口) 001: TIOC2A 输入 / 输出 (MTU2) 110: SCK1 输入 / 输出 (SCI) 上述以外: 禁止设定
9	PE6MD1	0	R/W	
8	PE6MD0	0	R/W	
7	—	0	R	保留位 读写值总是 0。
6	PE5MD2	0	R/W	PE5 模式位 选择 PE5/TIOC1B/TXD1 引脚的功能。 000: PE5 输入 / 输出 (端口) 001: TIOC1B 输入 / 输出 (MTU2) 110: TXD1 输出 (SCI) 上述以外: 禁止设定
5	PE5MD1	0	R/W	
4	PE5MD0	0	R/W	
3	—	0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
2	PE4MD2	0	R/W	PE4 模式位 选择 PE4/TIOC1A/RXD1 引脚的功能。 000: PE4 输入 / 输出 (端口) 001: TIOC1A 输入 / 输出 (MTU2) 110: RXD1 输入 (SCI) 上述以外: 禁止设定
1	PE4MD1	0	R/W	
0	PE4MD0	0	R/W	

• 端口 E 的控制寄存器 L1 (PECRL1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PE3 MD2	PE3 MD1	PE3 MD0	—	PE2 MD2	PE2 MD1	PE2 MD0	—	PE1 MD2	PE1 MD1	PE1 MD0	—	—	PE0 MD1	PE0 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	PE3MD2	0	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
13	PE3MD1	0	R/W	
12	PE3MD0	0	R/W	
11	—	0	R	保留位 读写值总是 0。
10	PE2MD2	0	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
9	PE2MD1	0	R/W	
8	PE2MD0	0	R/W	
7	—	0	R	保留位 读写值总是 0。
6	PE1MD2	0	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
5	PE1MD1	0	R/W	
4	PE1MD0	0	R/W	
3、2	—	全 0	R	保留位 读写值总是 0。
1	PE0MD1	0	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
0	PE0MD0	0	R/W	

(2) SH7137

• 端口E的控制寄存器H2 (PECRH2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	PE21 MD1	PE21 MD0	—	—	PE20 MD1	PE20 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 6	—	全 0	R	保留位 读写值总是 0。
5 4	PE21MD1 PE21MD0	0 0	R/W R/W	PE21 模式位 选择 PE21/ $\overline{\text{WRL}}$ /TIOC4DS/ $\overline{\text{TRST}}$ 引脚的功能。当使用 E10A ($\overline{\text{ASEMD0}}=\text{L}$ 电平) 时, 此引脚固定为 $\overline{\text{TRST}}$ 输入。 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	—	全 0	R	保留位 读写值总是 0。
1 0	PE20MD1 PE20MD0	0 0	R/W R/W	PE20 模式位 选择 PE20/TIOC4CS/TMS 引脚的功能。当使用 E10A ($\overline{\text{ASEMD0}}=\text{L}$ 电平) 时, 此引脚固定为 TMS 输入。 00: PE20 输入 / 输出 (端口) 01: TIOC4CS 输入 / 输出 (MTU2S) 上述以外: 禁止设定

【注】 * 只在内部 ROM 有效 / 无效的外部扩展模式中有效。在单芯片模式时不能设定。

• 端口E的控制寄存器H1 (PECRH1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PE19 MD1	PE19 MD0	—	—	PE18 MD1	PE18 MD0	—	—	PE17 MD1	PE17 MD0	—	PE16 MD2	PE16 MD1	PE16 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15、14	—	全 0	R	保留位 读写值总是 0。
13 12	PE19MD1 PE19MD0	0 0	R/W R/W	PE19 模式位 选择 PE19/ $\overline{\text{RD}}$ /TIOC4BS/TDO 引脚的功能。当使用 E10A ($\overline{\text{ASEMD0}}=\text{L}$ 电平) 时, 此引脚固定为 TDO 输出。 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	—	全 0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
9	PE18MD1	0	R/W	PE18 模式位 选择 PE18/ $\overline{\text{CS1}}$ /TIOC4AS/TDI 引脚的功能。当使用 E10A ($\overline{\text{ASEMD0}}=\text{L}$ 电平) 时, 此引脚固定为 TDI 输入。 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
8	PE18MD0	0	R/W	
7、6	—	全 0	R	保留位 读写值总是 0。
5	PE17MD1	0	R/W	PE17 模式位 选择 PE17/ $\overline{\text{CS0}}$ /TIOC3DS/TCK 引脚的功能。当使用 E10A ($\overline{\text{ASEMD0}}=\text{L}$ 电平) 时, 此引脚固定为 TCK 输入。 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
4	PE17MD0	0	R/W	
3	—	0	R	保留位 读写值总是 0。
2	PE16MD2	0	R/W	PE16 模式位 选择 PE16/ $\overline{\text{WAIT}}$ /TIOC3BS/ $\overline{\text{ASEBRKAK}}$ / $\overline{\text{ASEBRK}}$ 引脚的功能。当使用 E10A ($\overline{\text{ASEMD0}}=\text{L}$ 电平) 时, 此引脚固定为 $\overline{\text{ASEBRKAK}}$ 输出 / $\overline{\text{ASEBRK}}$ 输入。 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
1	PE16MD1	0	R/W	
0	PE16MD0	0	R/W	

【注】 * 只在内部 ROM 有效 / 无效的外部扩展模式有效。在单芯片模式时不能设定。

• 端口 E 的控制寄存器 L4 (PECRL4)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PE15 MD2	PE15 MD1	PE15 MD0	—	PE14 MD2	PE14 MD1	PE14 MD0	—	—	PE13 MD1	PE13 MD0	—	PE12 MD2	PE12 MD1	PE12 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	PE15MD2	0	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
13	PE15MD1	0	R/W	
12	PE15MD0	0	R/W	
11	—	0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
10	PE14MD2	0	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
9	PE14MD1	0	R/W	
8	PE14MD0	0	R/W	
7、6	—	全 0	R	保留位 读写值总是 0。
5	PE13MD1	0	R/W	PE13 模式位 选择 PE13/TIOC4B/MRES 引脚的功能。 00: PE13 输入 / 输出 (端口) 01: TIOC4B 输入 / 输出 (MTU2) 10: MRES 输入 (INTC) 上述以外: 禁止设定
4	PE13MD0	0	R/W	
3	—	0	R	保留位 读写值总是 0。
2	PE12MD2	0	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
1	PE12MD1	0	R/W	
0	PE12MD0	0	R/W	

• 端口 E 的控制寄存器 L3 (PECRL3)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PE11 MD2	PE11 MD1	PE11 MD0	—	PE10 MD2	PE10 MD1	PE10 MD0	—	PE9 MD2	PE9 MD1	PE9 MD0	—	PE8 MD2	PE8 MD1	PE8 MD0
初始值:	0	0	0	0	0	0*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

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	PE11MD2	0	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
13	PE11MD1	0	R/W	
12	PE11MD0	0	R/W	
11	—	0	R	保留位 读写值总是 0。
10	PE10MD2	0*1	R/W	PE10 模式位 选择 PE10/CS0/TIOC3C 引脚的功能。 000: PE10 输入 / 输出 (端口) 001: TIOC3C 输入 / 输出 (MTU2) 100: CS0 输出 (BSC) *2 上述以外: 禁止设定
9	PE10MD1	0	R/W	
8	PE10MD0	0	R/W	
7	—	0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
6	PE9MD2	0	R/W	PE9 模式位 选择 PE9/TIOC3B 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
5	PE9MD1	0	R/W	
4	PE9MD0	0	R/W	
3	—	0	R	保留位 读写值总是 0。
2	PE8MD2	0*1	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
1	PE8MD1	0	R/W	
0	PE8MD0	0	R/W	

【注】 *1 在内部 ROM 无效的外部扩展模式时, 初始值为 1。

*2 只在内部 ROM 有效 / 无效的外部扩展模式中有效。在单芯片模式时不能设定。

• 端口 E 的控制寄存器 L2 (PECRL2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PE7 MD2	PE7 MD1	PE7 MD0	—	PE6 MD2	PE6 MD1	PE6 MD0	—	PE5 MD2	PE5 MD1	PE5 MD0	—	PE4 MD2	PE4 MD1	PE4 MD0
初始值:	0	0*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 无效的外部扩展模式中, 初始值为 1。

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	PE7MD2	0*1	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
13	PE7MD1	0	R/W	
12	PE7MD0	0	R/W	
11	—	0	R	保留位 读写值总是 0。
10	PE6MD2	0*1	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
9	PE6MD1	0	R/W	
8	PE6MD0	0	R/W	
7	—	0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
6	PE5MD2	0*1	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
5	PE5MD1	0	R/W	
4	PE5MD0	0	R/W	
3	—	0	R	保留位 读写值总是 0。
2	PE4MD2	0*1	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
1	PE4MD1	0	R/W	
0	PE4MD0	0	R/W	

【注】 *1 在内部 ROM 无效的外部扩展模式，初始值为 1。

*2 只在内部 ROM 有效 / 无效的外部扩展模式中有效。在单芯片模式时不能设定。

• 端口 E 的控制寄存器 L1 (PECRL1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PE3 MD2	PE3 MD1	PE3 MD0	—	PE2 MD2	PE2 MD1	PE2 MD0	—	PE1 MD2	PE1 MD1	PE1 MD0	—	—	PE0 MD1	PE0 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写值总是 0。
14	PE3MD2	0	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
13	PE3MD1	0	R/W	
12	PE3MD0	0	R/W	
11	—	0	R	保留位 读写值总是 0。
10	PE2MD2	0	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
9	PE2MD1	0	R/W	
8	PE2MD0	0	R/W	
7	—	0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
6	PE1MD2	0	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
5	PE1MD1	0	R/W	
4	PE1MD0	0	R/W	
3、2	—	全 0	R	保留位 读写值总是 0。
1	PE0MD1	0	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
0	PE0MD0	0	R/W	

20.1.9 IRQOUT 功能的控制寄存器 (IFCR)

IFCR 是 16 位可读写寄存器, 在通过端口 E 的控制寄存器 L4 (PECRL4) 将多路复用功能设定为 IRQOUT 输出时, 用于控制 IRQOUT 引脚的输出。当 PECRL4 的设定为其他功能时, IFCR 的设定不影响引脚功能。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IRQ MD1	IRQ MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 2	—	全 0	R	保留位 读写值总是 0。
1	IRQMD1	0	R/W	端口 E 的 IRQOUT 引脚功能选择 当 PECRL4 的 bit14、bit13 和 bit12 (PE15MD2、PE15MD1、PE15MD0) 被设定为 (0、1、1) 时, 选择 IRQOUT 引脚的功能。 00: 输出中断请求的接受信号 01: 禁止设定 10: 输出中断请求的接受信号 11: 总是输出高电平
0	IRQMD0	0	R/W	

20.2 使用时的注意事项

1. 本 LSI 将同一功能作为多路复用功能分配到多个引脚。这是为了在提高引脚功能选择自由度的同时方便电路板的设计，但是在 2 个以上的引脚使用 1 个功能时，必须注意以下几点：
 - 引脚功能为输入功能的情况
通过 OR 或 AND 逻辑将多个引脚的输入信号组合为 1 个信号，传送到 LSI 内部。因此，根据其他同一功能的引脚输入状态，有可能将和输入信号不同的信号传送到 LSI 内部。分配到多个引脚的输入功能的传送格式如表 20.14 所示。在 2 个以上的引脚使用以下任意功能时，必须考虑传送格式并注意信号的极性。

表 20.14 分配到多个引脚的输入功能的传送格式

OR 型	AND 型
SCK0 ~ SCK2、RXD0 ~ RXD2	IRQ0 ~ IRQ3、 $\overline{\text{WAIT}}$ 、 $\overline{\text{POE0}}$ 、 $\overline{\text{POE1}}$ 、 $\overline{\text{POE4}}$ ~ $\overline{\text{POE5}}$

OR 型：通过 OR 逻辑将多个引脚的输入信号组合为 1 个信号，传送到 LSI 内部。

AND 型：通过 AND 逻辑将多个引脚的输入信号组合为 1 个信号，传送到 LSI 内部。

- 引脚功能为输出功能的情况
能从所选的全部引脚输出同一功能。
2. 输入/输出端口和 IRQ 是多路复用引脚，如果端口的输入从低电平状态转换为 IRQ 边沿检测，就检测该边沿。
 3. 只能设定表 20.10 ~ 表 20.12 中 PFC 能设定的功能，否则不保证运行。
 4. 关于单芯片模式（MCU 运行模式 3）中的 PFC 设定
在单芯片模式中，不能通过 PFC 选择地址总线、数据总线、总线控制信号、 $\overline{\text{BREQ}}$ 、 $\overline{\text{BACK}}$ 和 CK。否则，地址总线为高电平或者低电平输出，数据总线为高阻抗输出，其他输出信号为高电平输出。因为 $\overline{\text{BREQ}}$ 和 $\overline{\text{WAIT}}$ 为输入状态，所以不能将其置为开路。总线权请求输入和外部等待无效。

第 21 章 I/O 端口

SH7136 的端口由 A、B、E、F 共 4 个端口构成。端口 A、B、E 分别是 16 位、6 位、22 位的输入 / 输出端口，端口 F 是 12 位输入专用端口。

SH7137 的端口由 A、B、D、E、F 共 5 个端口构成。端口 A、B、D、E 分别是 16 位、8 位、11 位、22 位的输入 / 输出端口，端口 F 是 16 位输入专用端口。

各端口的引脚都是和其他功能兼用的多路复用引脚，通过引脚功能控制器（PFC）选择多路复用引脚的功能。

各端口有保存引脚数据的数据寄存器。

21.1 端口 A

如图 21.1 所示，SH7136 的端口 A 是有 16 个引脚的输入 / 输出端口。

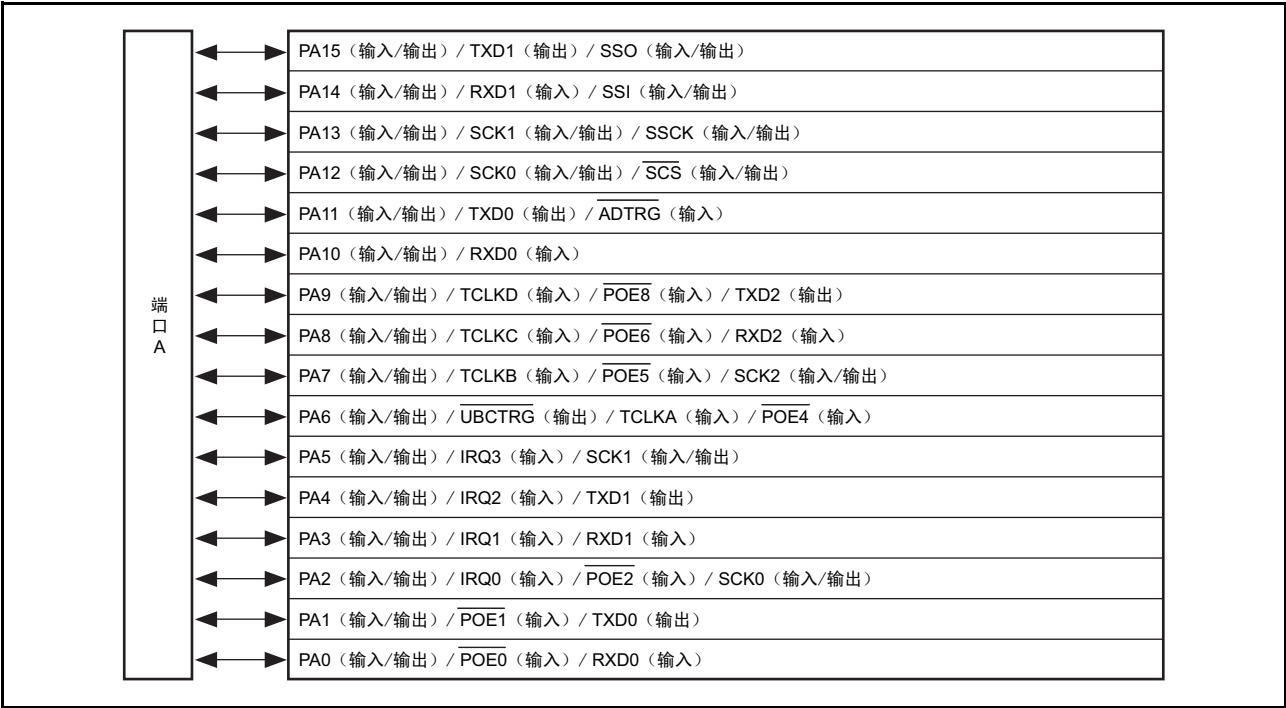


图 21.1 端口 A（SH7136）

如图 21.2 所示，SH7137 的端口 A 是有 16 个引脚的输入 / 输出端口。

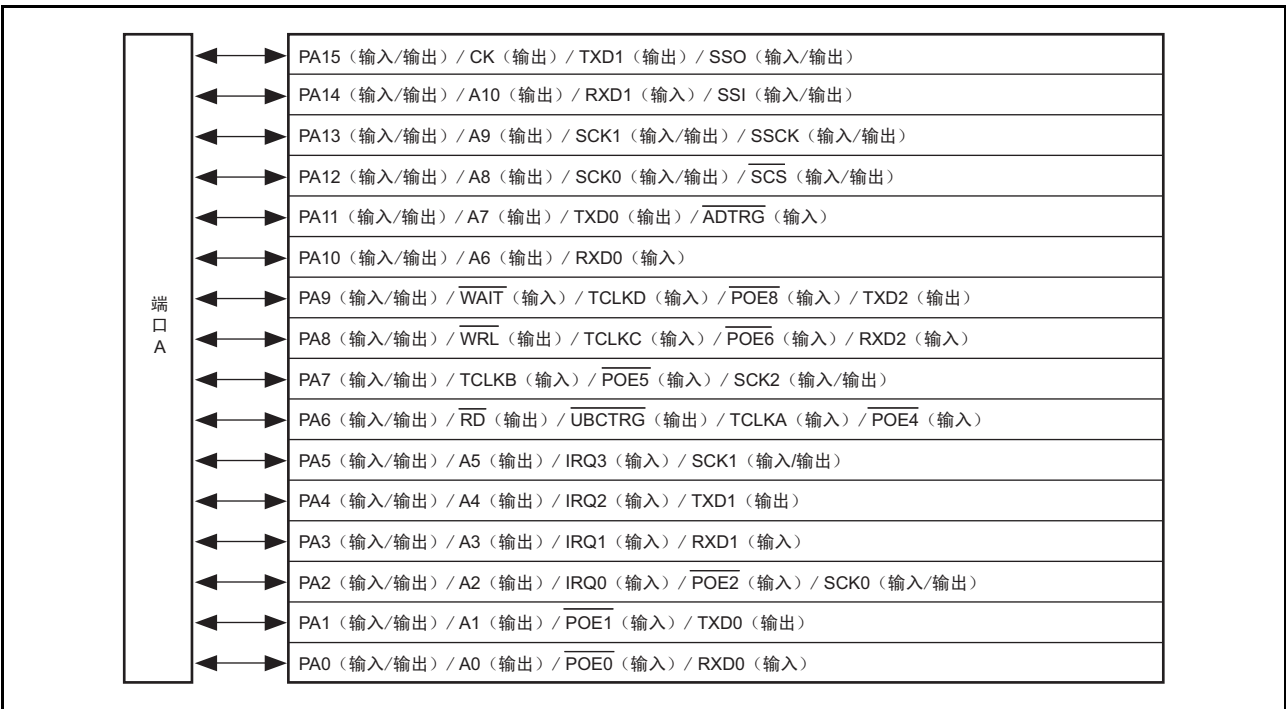


图 21.2 端口 A (SH7137)

21.1.1 寄存器说明

端口 A 是 16 位输入 / 输出端口，有以下寄存器。有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 25 章 寄存器一览”。

表 21.1 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
端口 A 的数据寄存器 L	PADRL	R/W	H'0000	H'FFFFD102	8、16
端口 A 的端口寄存器 L	PAPRL	R	H'xxxx	H'FFFFD11E	8、16

21.1.2 端口 A 的数据寄存器 L (PADRL)

PADRL 是 16 位可读写寄存器，保存端口 A 的数据。PA15DR ~ PA0DR 位分别对应 PA15 ~ PA0 引脚（省略有关兼用功能的记述）。

在引脚功能为通用输出时，如果给 PADRL 写值，就从引脚输出该值；如果读 PADRL，就与引脚的状态无关，直接读寄存器的值。

在引脚功能为通用输入时，如果读 PADRL，就直接读引脚的状态而非寄存器的值；如果给 PADRL 写值，就能将值写到 PADRL，但是不影响引脚的状态。端口 A 的数据寄存器的读写操作如表 21.2 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	初始值	R/W	说明
15	PA15DR	0	R/W	参照表 21.2。
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 21.2 端口 A 的数据寄存器 L（PADRL）的读写操作

- PADRL 的 bit15 ~ 0

PAIOR	引脚功能	读	写
0	通用输入	引脚的状态	能写 PADRL，但是不影响引脚的状态。
	非通用输入	引脚的状态	能写 PADRL，但是不影响引脚的状态。
1	通用输出	PADRL 的值	从引脚输出所写的值。
	非通用输出	PADRL 的值	能写 PADRL，但是不影响引脚的状态。

21.1.3 端口 A 的端口寄存器 L（PAPRL）

PAPRL 是 16 位只读寄存器，与 PFC 的设定无关，能随时读引脚的状态。PA15PR ~ PA0PR 位分别对应 PA15 ~ PA0 引脚（省略有关兼用功能的记述）。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PA15 PR	PA14 PR	PA13 PR	PA12 PR	PA11 PR	PA10 PR	PA9 PR	PA8 PR	PA7 PR	PA6 PR	PA5 PR	PA4 PR	PA3 PR	PA2 PR	PA1 PR	PA0 PR
初始值:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15	PA15PR	引脚的状态	R	与 PFC 的设置无关，读引脚的状态。写操作无效。
14	PA14PR	引脚的状态	R	
13	PA13PR	引脚的状态	R	
12	PA12PR	引脚的状态	R	
11	PA11PR	引脚的状态	R	
10	PA10PR	引脚的状态	R	
9	PA9PR	引脚的状态	R	
8	PA8PR	引脚的状态	R	
7	PA7PR	引脚的状态	R	
6	PA6PR	引脚的状态	R	
5	PA5PR	引脚的状态	R	
4	PA4PR	引脚的状态	R	
3	PA3PR	引脚的状态	R	
2	PA2PR	引脚的状态	R	
1	PA1PR	引脚的状态	R	
0	PA0PR	引脚的状态	R	

21.2 端口 B

如图 21.3 所示，SH7136 的端口 B 是有 6 个引脚的输入 / 输出端口。

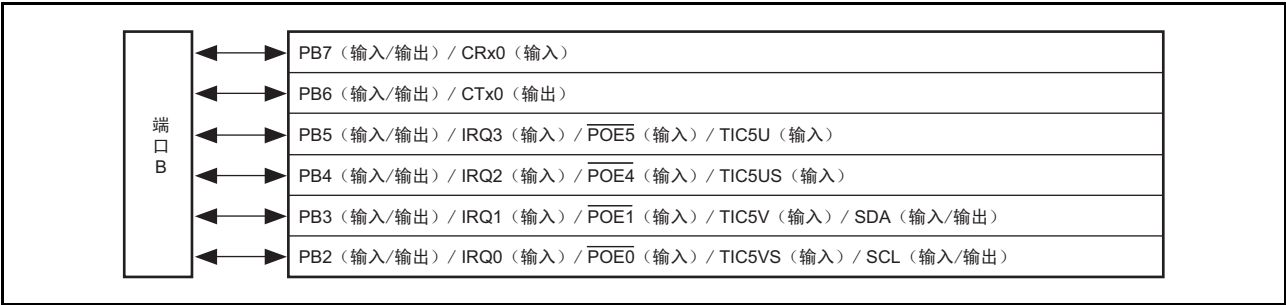


图 21.3 端口 B（SH7136）

如图 21.4 所示，SH7137 的端口 B 是有 8 个引脚的输入 / 输出端口。

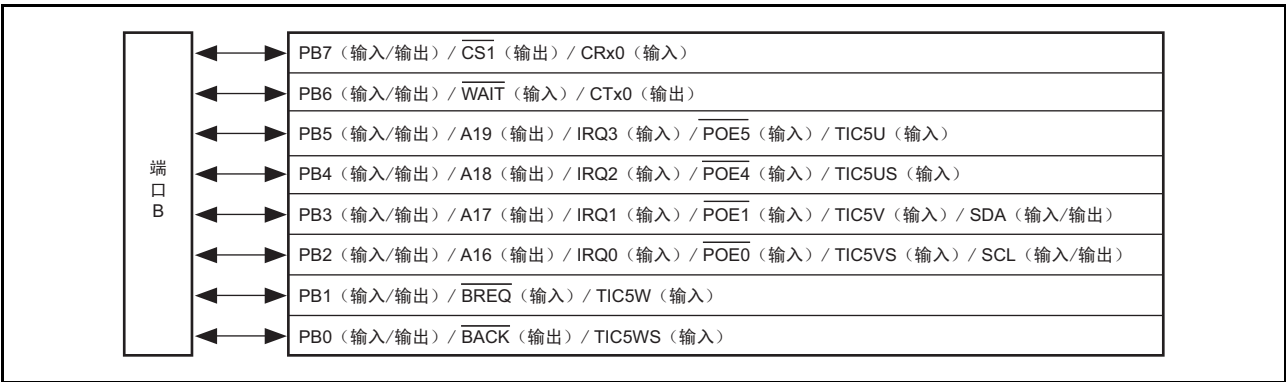


图 21.4 端口 B（SH7137）

21.2.1 寄存器说明

SH7136 的端口 B 是 6 位输入 / 输出端口，SH7137 的端口 B 是 8 位输入 / 输出端口，有以下寄存器。有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 25 章 寄存器一览”。

表 21.3 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
端口 B 的数据寄存器 L	PBDRL	R/W	H'0000	H'FFFFD182	8、16
端口 B 的端口寄存器 L	PBPRL	R	H'00xx	H'FFFFD19E	8、16

21.2.2 端口 B 的数据寄存器 L（PBDRL）

PBDRL 是 16 位可读写寄存器，保存端口 B 的数据。SH7136 的 PB7DR ~ PB2DR 位分别对应 PB7 ~ PB2 引脚（省略有关兼用功能的记述）；SH7137 的 PB7DR ~ PB0DR 位分别对应 PB7 ~ PB0 引脚（省略有关兼用功能的记述）。

在引脚功能为通用输出时，如果给 PBDRL 写值，就从引脚输出该值；如果读 PBDRL，就与引脚的状态无关，直接读寄存器的值。

在引脚功能为通用输入时，如果读 PBDRL，就直接读引脚的状态而非寄存器的值；如果给 PBDRL 写值，就能将值写到 PBDRL，但是不影响引脚的状态。端口 B 的数据寄存器的读写操作如表 21.4 所示。

- PBDRL (SH7136)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PB7 DR	PB6 DR	PB5 DR	PB4 DR	PB3 DR	PB2 DR	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R

位	位名	初始值	R/W	说明
15 ~ 8	—	全 0	R	保留位 读写值总是 0。
7	PB7DR	0	R/W	参照表 21.4。
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、0	—	全 0	R	保留位 读写值总是 0。

- PBDRL (SH7137)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PB7 DR	PB6 DR	PB5 DR	PB4 DR	PB3 DR	PB2 DR	PB1 DR	PB0 DR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 8	—	全 0	R	保留位 读写值总是 0。
7	PB7DR	0	R/W	参照表 21.4。
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 21.4 端口 B 的数据寄存器（PBDR）的读写操作

- PBDRL 的 bit7 ~ 0

PBIOR	引脚功能	读	写
0	通用输入	引脚的状态	能写 PBDRL，但是不影响引脚的状态。
	非通用输入	引脚的状态	能写 PBDRL，但是不影响引脚的状态。
1	通用输出	PBDRL 的值	从引脚输出所写的值。
	非通用输出	PBDRL 的值	能写 PBDRL，但是不影响引脚的状态。

21.2.3 端口 B 的端口寄存器 L（PBPRL）

PBPRL 是 16 位只读寄存器，与 PFC 的设定无关，能随时读引脚的状态。SH7136 的 PB7PR ~ PB2PR 位分别对应 PB7 ~ PB2 引脚（省略有关兼用功能的记述）；SH7137 的 PB7PR ~ PB0PR 位分别对应 PB7 ~ PB0 引脚（省略有关兼用功能的记述）。

- PBPRL（SH7136）

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PB7 PR	PB6 PR	PB5 PR	PB4 PR	PB3 PR	PB2 PR	—	—
初始值:	0	0	0	0	0	0	0	0	*	*	*	*	*	*	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15 ~ 8	—	全 0	R	保留位 读写值总是 0。
7	PB7PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
6	PB6PR	引脚的状态	R	
5	PB5PR	引脚的状态	R	
4	PB4PR	引脚的状态	R	
3	PB3PR	引脚的状态	R	
2	PB2PR	引脚的状态	R	
1、0	—	全 0	R	保留位 读写值总是 0。

- PBPRL（SH7137）

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PB7 PR	PB6 PR	PB5 PR	PB4 PR	PB3 PR	PB2 PR	PB1 PR	PB0 PR
初始值:	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15 ~ 8	—	全 0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
7	PB7PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
6	PB6PR	引脚的状态	R	
5	PB5PR	引脚的状态	R	
4	PB4PR	引脚的状态	R	
3	PB3PR	引脚的状态	R	
2	PB2PR	引脚的状态	R	
1	PB1PR	引脚的状态	R	
0	PB0PR	引脚的状态	R	

21.3 端口 D（只限 SH7137）

如图 21.5 所示，SH7137 的端口 D 是有 11 个引脚的输入 / 输出端口。



图 21.5 端口 D

21.3.1 寄存器说明

端口 D 是 11 位输入 / 输出端口，但是 SH7136 没有端口 D，有以下寄存器，有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 25 章 寄存器一览”。

表 21.5 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
端口 D 的数据寄存器 L	PDDRL	R/W	H'0000	H'FFFFD282	8、16
端口 D 的端口寄存器 L	PDPRL	R	H'xxxx	H'FFFFD29E	8、16

21.3.2 端口 D 的数据寄存器 L（PDDRL）

PDDRL 是 16 位可读写寄存器，保存端口 D 的数据。PD10DR ~ PD0DR 位分别对应 PD10 ~ PD0 引脚（省略有关兼用功能的记述）。

在引脚功能为通用输出时，如果给 PDDRL 写值，就从引脚输出该值；如果读 PDDRL，就与引脚的状态无关，直接读寄存器的值。

在引脚功能为通用输入时，如果读 PDDRL，就直接读引脚的状态而非寄存器的值；如果给 PDDRL 写值，就能将值写到 PDDRL，但是不影响引脚的状态。端口 D 的数据寄存器 L 的读写操作如表 21.6 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	初始值	R/W	说明
15 ~ 11	—	全 0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
10	PD10DR	0	R/W	参照表 21.6。
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 21.6 端口 D 的数据寄存器 L (PDDRL) 的读写操作

- PDDRL 的 bit10 ~ 0

PDIOR	引脚功能	读	写
0	通用输入	引脚的状态	能写 PDDRL，但是不影响引脚的状态。
	非通用输入	引脚的状态	能写 PDDRL，但是不影响引脚的状态。
1	通用输出	PDDRL 的值	从引脚输出所写的值。
	非通用输出	PDDRL 的值	能写 PDDRL，但是不影响引脚的状态。

21.3.3 端口 D 的端口寄存器 L (PDPRL)

PDPRL 是 16 位只读寄存器，与 PFC 的设定无关，能随时读引脚的状态。PD10PR ~ PD0PR 位分别对应 PD10 ~ PD0 引脚（省略有关兼用功能的记述）。

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	初始值	R/W	说明
15 ~ 11	—	全 0	R	保留位 读写值总是 0。
10	PD10PR	引脚的状态	R	与 PFC 的设定无关，读引脚的状态。写操作无效。
9	PD9PR	引脚的状态	R	
8	PD8PR	引脚的状态	R	
7	PD7PR	引脚的状态	R	
6	PD6PR	引脚的状态	R	
5	PD5PR	引脚的状态	R	
4	PD4PR	引脚的状态	R	
3	PD3PR	引脚的状态	R	
2	PD2PR	引脚的状态	R	
1	PD1PR	引脚的状态	R	
0	PD0PR	引脚的状态	R	

21.4 端口 E

如图 21.6 所示，SH7136 的端口 E 是有 22 个引脚的输入 / 输出端口。

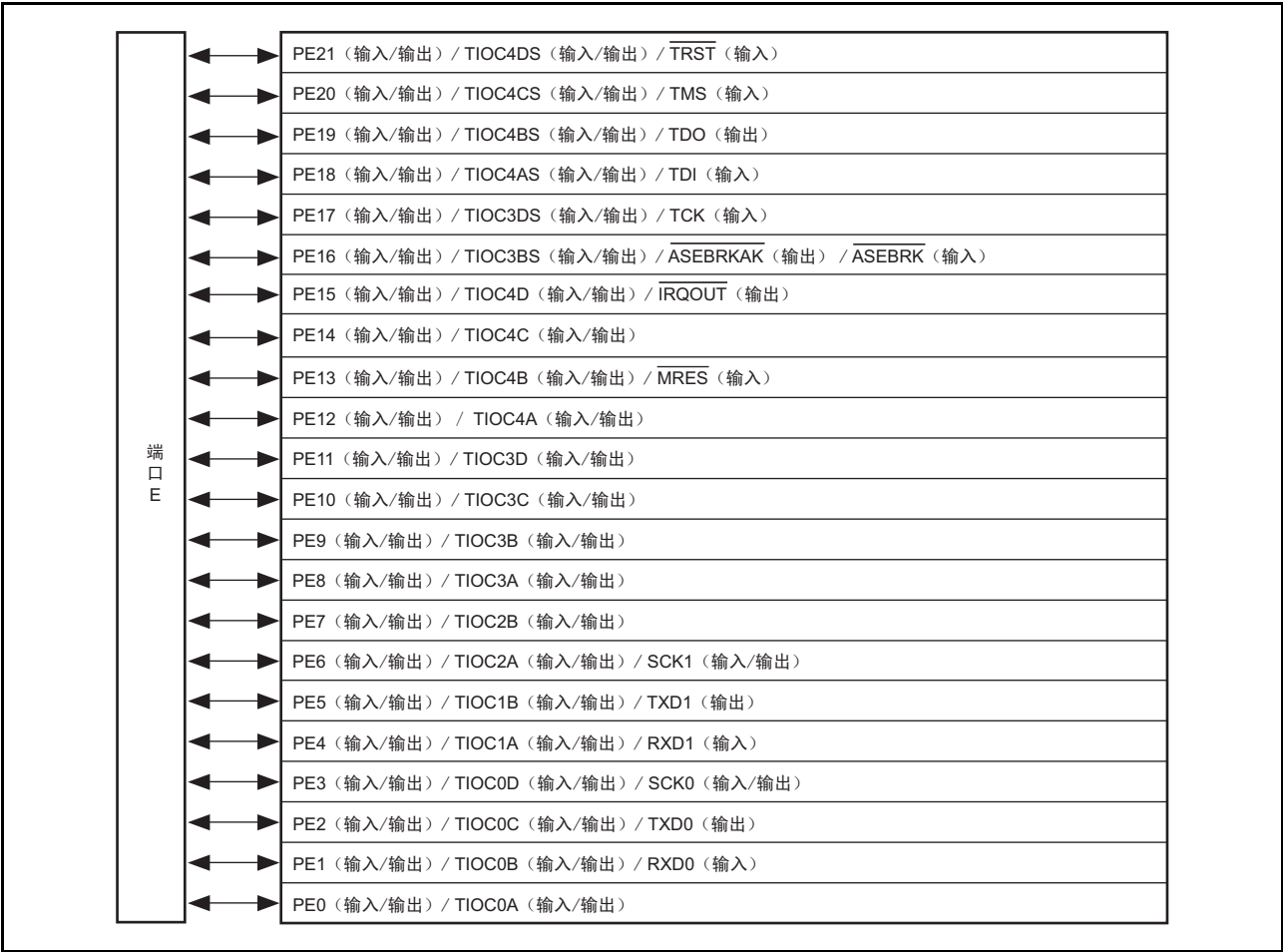


图 21.6 端口 E (SH7136)

如图 21.7 所示，SH7137 的端口 E 是有 22 个引脚的输入 / 输出端口。



图 21.7 端口 E (SH7137)

21.4.1 寄存器说明

端口 E 是 22 位输入 / 输出端口，有以下寄存器，有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 25 章 寄存器一览”。

表 21.7 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
端口 E 的数据寄存器 H	PEDRH	R/W	H'0000	H'FFFFD300	8、16、32
端口 E 的数据寄存器 L	PEDRL	R/W	H'0000	H'FFFFD302	8、16
端口 E 的端口寄存器 H	PEPRH	R	H'00xx	H'FFFFD31C	8、16、32
端口 E 的端口寄存器 L	PEPRL	R	H'xxxx	H'FFFFD31E	8、16

21.4.2 端口 E 的数据寄存器 H、L（PEDRH、PEDRL）

PEDRH 和 PEDRL 是 16 位可读写寄存器，保存端口 E 的数据。PE21DR ~ PE0DR 位分别对应 PE21 ~ PE0 引脚（省略有关兼用功能的记述）。

在引脚功能为通用输出时，如果给 PEDRH 或者 PEDRL 写值，就从引脚输出该值；如果读 PEDRH 或者 PEDRL，就与引脚的状态无关，直接读寄存器的值。

在引脚功能为通用输入时，如果读 PEDRH 或者 PEDRL，就直接读引脚的状态而非寄存器的值；如果给 PEDRH 或者 PEDRL 写值，就能将值写到 PEDRH 或者 PEDRL，但是不影响引脚的状态。端口 E 的数据寄存器的读写操作如表 21.8 所示。

• PEDRH

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	PE21 DR	PE20 DR	PE19 DR	PE18 DR	PE17 DR	PE16 DR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 6	—	全 0	R	保留位 读写值总是 0。
5	PE21DR	0	R/W	参照表 21.8。
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	

• 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	初始值	R/W	说明
15	PE15DR	0	R/W	参照表 21.8。
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 21.8 端口 E 的数据寄存器（PEDR）的读写操作

- PEDRH 的 bit5～0 和 PEDRL 的 bit15～0

PEIOR	引脚功能	读	写
0	通用输入	引脚的状态	能写 PEDRH、L，但是不影响引脚的状态。
	非通用输入	引脚的状态	能写 PEDRH、L，但是不影响引脚的状态。
1	通用输出	PEDRH、L 的值	从引脚输出所写的值。
	非通用输出	PEDRH、L 的值	能写 PEDRH、L，但是不影响引脚的状态。

21.4.3 端口 E 的端口寄存器 H、L（PEPRH、PEPRL）

PEPRH 和 PEPRL 是 16 位只读寄存器，与 PFC 的设定无关，能随时读引脚的状态。PE21PR～PE0PR 位分别对应 PE21～PE0 引脚（省略有关兼用功能的记述）。

- PEPRH

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	PE21 PR	PE20 PR	PE19 PR	PE18 PR	PE17 PR	PE16 PR
初始值:	0	0	0	0	0	0	0	0	0	0	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15～6	—	全 0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
5	PE21PR	引脚的状态	R	与 PFC 的设置无关，读引脚的状态。写操作无效。
4	PE20PR	引脚的状态	R	
3	PE19PR	引脚的状态	R	
2	PE18PR	引脚的状态	R	
1	PE17PR	引脚的状态	R	
0	PE16PR	引脚的状态	R	

- PEPR1

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PE15 PR	PE14 PR	PE13 PR	PE12 PR	PE11 PR	PE10 PR	PE9 PR	PE8 PR	PE7 PR	PE6 PR	PE5 PR	PE4 PR	PE3 PR	PE2 PR	PE1 PR	PE0 PR
初始值:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15	PE15PR	引脚的状态	R	与 PFC 的设置无关，读引脚的状态。写操作无效。
14	PE14PR	引脚的状态	R	
13	PE13PR	引脚的状态	R	
12	PE12PR	引脚的状态	R	
11	PE11PR	引脚的状态	R	
10	PE10PR	引脚的状态	R	
9	PE9PR	引脚的状态	R	
8	PE8PR	引脚的状态	R	
7	PE7PR	引脚的状态	R	
6	PE6PR	引脚的状态	R	
5	PE5PR	引脚的状态	R	
4	PE4PR	引脚的状态	R	
3	PE3PR	引脚的状态	R	
2	PE2PR	引脚的状态	R	
1	PE1PR	引脚的状态	R	
0	PE0PR	引脚的状态	R	

21.5 端口 F

如图 21.8 所示，SH7136 的端口 F 是有 12 个引脚的输入专用端口。

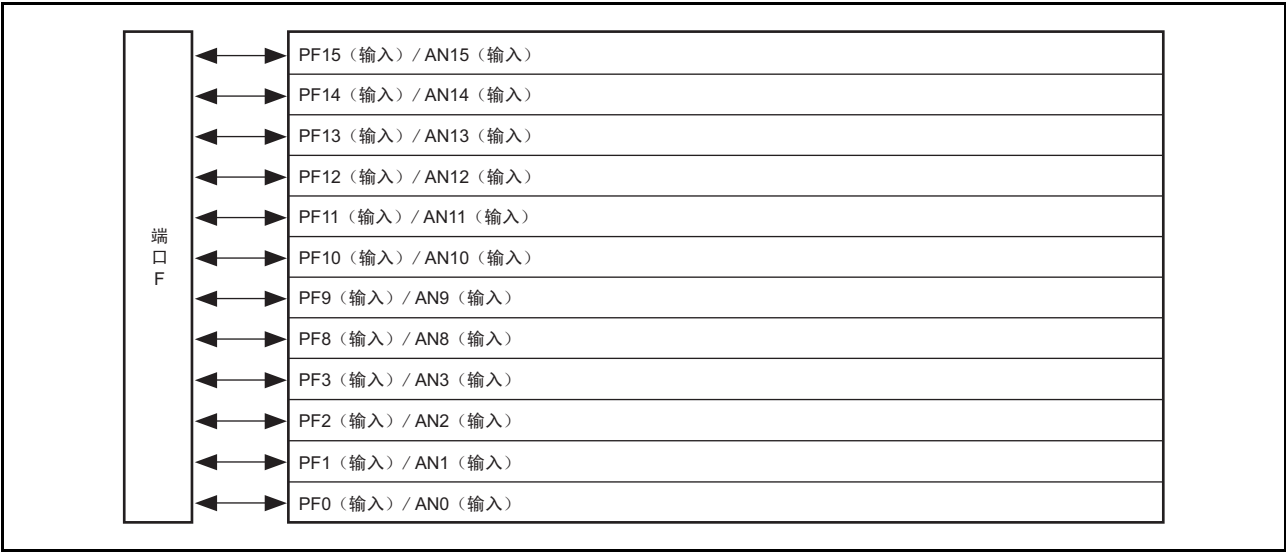


图 21.8 端口 F (SH7136)

如图 21.9 所示，SH7137 的端口 F 是有 16 个引脚的输入专用端口。

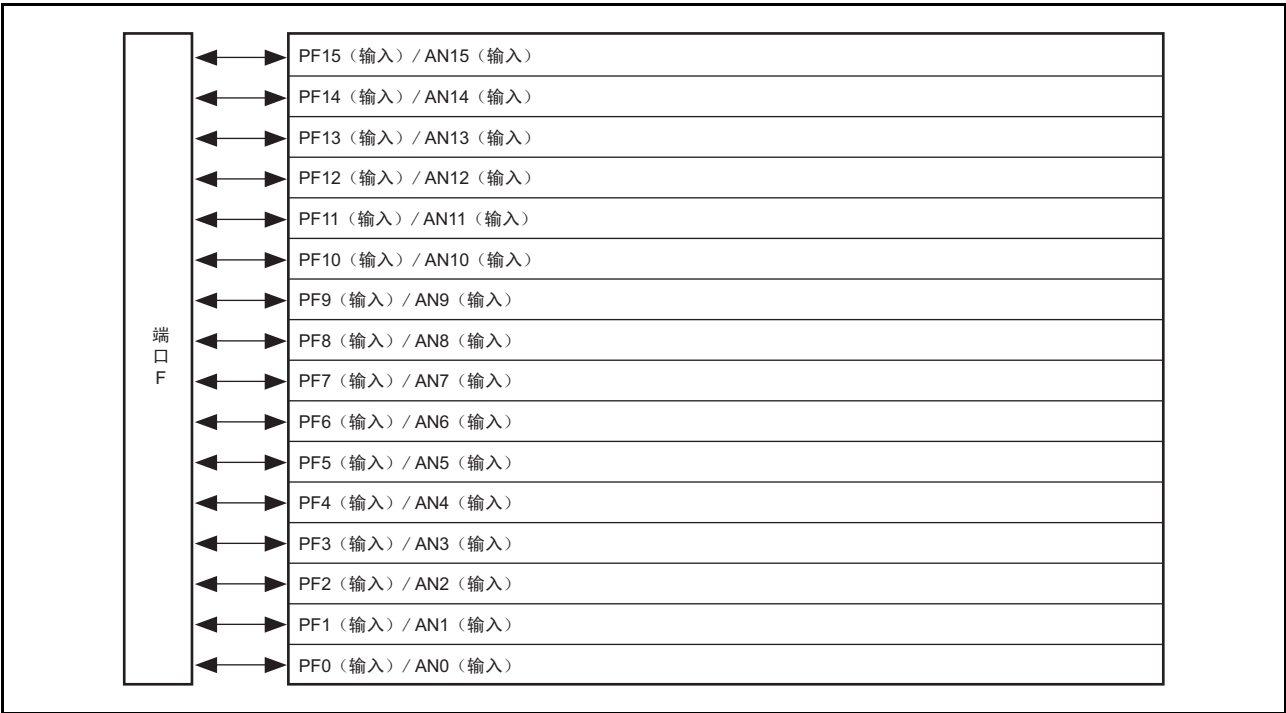


图 21.9 端口 F (SH7137)

21.5.1 寄存器说明

SH7136 的端口 F 是 12 位输入端口，SH7137 的端口 F 是 16 位输入端口，有以下寄存器。有关此寄存器的地址和各处理模式的寄存器状态，请参照“第 25 章 寄存器一览”。

表 21.9 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
端口 F 的数据寄存器 L	PFDRL	R	H'xxxx	H'FFFFD382	8、16

21.5.2 端口 F 的数据寄存器 L（PFDRL）

PFDRL 是 16 位只读寄存器，保存端口 F 的数据。SH7136 的 PF15DR ~ PF8DR 位和 PF3DR ~ PF0DR 位分别对应 PF15 ~ PF8 引脚和 PF3 ~ PF0 引脚（省略有关兼用功能的记述）；SH7137 的 PF15DR ~ PF0DR 位分别对应 PF15 ~ PF0 引脚（省略有关兼用功能的记述）。

即使给这些位写值也被忽视，不影响引脚的状态。如果读这些位，就直接读引脚的状态而非这些位的值。但是在对 A/D 转换器的模拟输入进行采样的期间，读取值是 1。端口 F 的数据寄存器的读写操作如表 21.10 所示。

• PFDRL（SH7136）

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF15 DR	PF14 DR	PF13 DR	PF12 DR	PF11 DR	PF10 DR	PF9 DR	PF8 DR	—	—	—	—	PF3 DR	PF2 DR	PF1 DR	PF0 DR
初始值:	*	*	*	*	*	*	*	*	0	0	0	0	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15	PF15DR	引脚的状态	R	参照表 21.10。
14	PF14DR	引脚的状态	R	
13	PF13DR	引脚的状态	R	
12	PF12DR	引脚的状态	R	
11	PF11DR	引脚的状态	R	
10	PF10DR	引脚的状态	R	
9	PF9DR	引脚的状态	R	
8	PF8DR	引脚的状态	R	
7 ~ 4	—	全 0	R	保留位 读写值总是 0。
3	PF3DR	引脚的状态	R	参照表 21.10。
2	PF2DR	引脚的状态	R	
1	PF1DR	引脚的状态	R	
0	PF0DR	引脚的状态	R	

• PFDRL (SH7137)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PF15 DR	PF14 DR	PF13 DR	PF12 DR	PF11 DR	PF10 DR	PF9 DR	PF8 DR	PF7 DR	PF6 DR	PF5 DR	PF4 DR	PF3 DR	PF2 DR	PF1 DR	PF0 DR
初始值:	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15	PF15DR	引脚的状态	R	参照表 21.10。
14	PF14DR	引脚的状态	R	
13	PF13DR	引脚的状态	R	
12	PF12DR	引脚的状态	R	
11	PF11DR	引脚的状态	R	
10	PF10DR	引脚的状态	R	
9	PF9DR	引脚的状态	R	
8	PF8DR	引脚的状态	R	
7	PF7DR	引脚的状态	R	
6	PF6DR	引脚的状态	R	
5	PF5DR	引脚的状态	R	
4	PF4DR	引脚的状态	R	
3	PF3DR	引脚的状态	R	
2	PF2DR	引脚的状态	R	
1	PF1DR	引脚的状态	R	
0	PF0DR	引脚的状态	R	

表 21.10 端口 F 的数据寄存器 L (PFDRL) 的读写操作

• PFDRL 的 bit15~0

引脚功能	读	写
通用输入	读引脚的状态	被忽视 (不影响引脚的状态)。
ANn 输入	读取值是 1	被忽视 (不影响引脚的状态)。

第 22 章 闪存

本 LSI 内置 256KB 的闪存，其特点如下所示。

22.1 特点

- 根据 LSI 启动模式设定的 2 种闪存 MAT
内部闪存分配在同一个地址空间的 2 种存储空间（以下称为存储器 MAT），能根据启动时的模式设定，选择从哪一个存储器 MAT 启动。在启动后，也能通过存储体转换方式转换 MAT。
在用户模式中，上电复位时启动的用户 MAT：256KB
在用户引导模式中，上电复位时启动的用户引导 MAT：12KB
- 3 种板上编程模式和 1 种板外编程模式
板上编程模式
引导模式：能通过使用内部 SCI 接口的编程模式，改写用户 MAT 和用户引导 MAT。在此模式中，能自动调整主机和本 LSI 之间的位速率。
用户编程模式：能通过任意的接口改写用户 MAT。
用户引导模式：能建立任意接口的用户引导程序，也能改写用户 MAT。

板外编程模式
编程器模式：能通过使用 PROM 编程器的编程器模式，改写用户 MAT 和用户引导 MAT。
- 通过下载内部程序，建立编程/擦除接口
本 LSI 内置专用的编程/擦除程序。在将此程序下载到内部 RAM 后，只要设定自变量参数就能进行编程和擦除，并且支持用户分支。
用户分支：以 128 字节为单位进行编程处理，编程处理由外加写脉冲和读验证等几个步骤构成；以 1 个分割块为单位进行擦除处理，擦除处理由几个处理步骤构成。能在各步骤之间设定用户处理程序的执行，此设定称为用户分支。
- 通过内部 RAM 对闪存进行的仿真功能
能通过将闪存和一部分内部 RAM 重叠，对闪存的改写进行实时仿真。
- 保护模式
保护模式有通过寄存器设定进行的软件保护和通过 FWE 引脚进行的硬件保护 2 种模式，能设定闪存的编程/擦除的保护状态。
另外，如果检测到编程/擦除中的失控等异常，就能转移到错误保护状态，并中断编程/擦除处理。
- 编程/擦除时间
在 128 字节同时编程时，闪存的编程时间为 t_{pms} (Typ.)，平均每 1 字节 $t_p/128ms$ ，擦除时间平均每块为 t_{Es} (Typ.)。
- 改写次数
最多能改写 N_{WEC} 次闪存。
- 编程/擦除时的工作频率
编程/擦除时的最大工作频率为 40MHz (Pφ)。

22.2 概要

22.2.1 框图

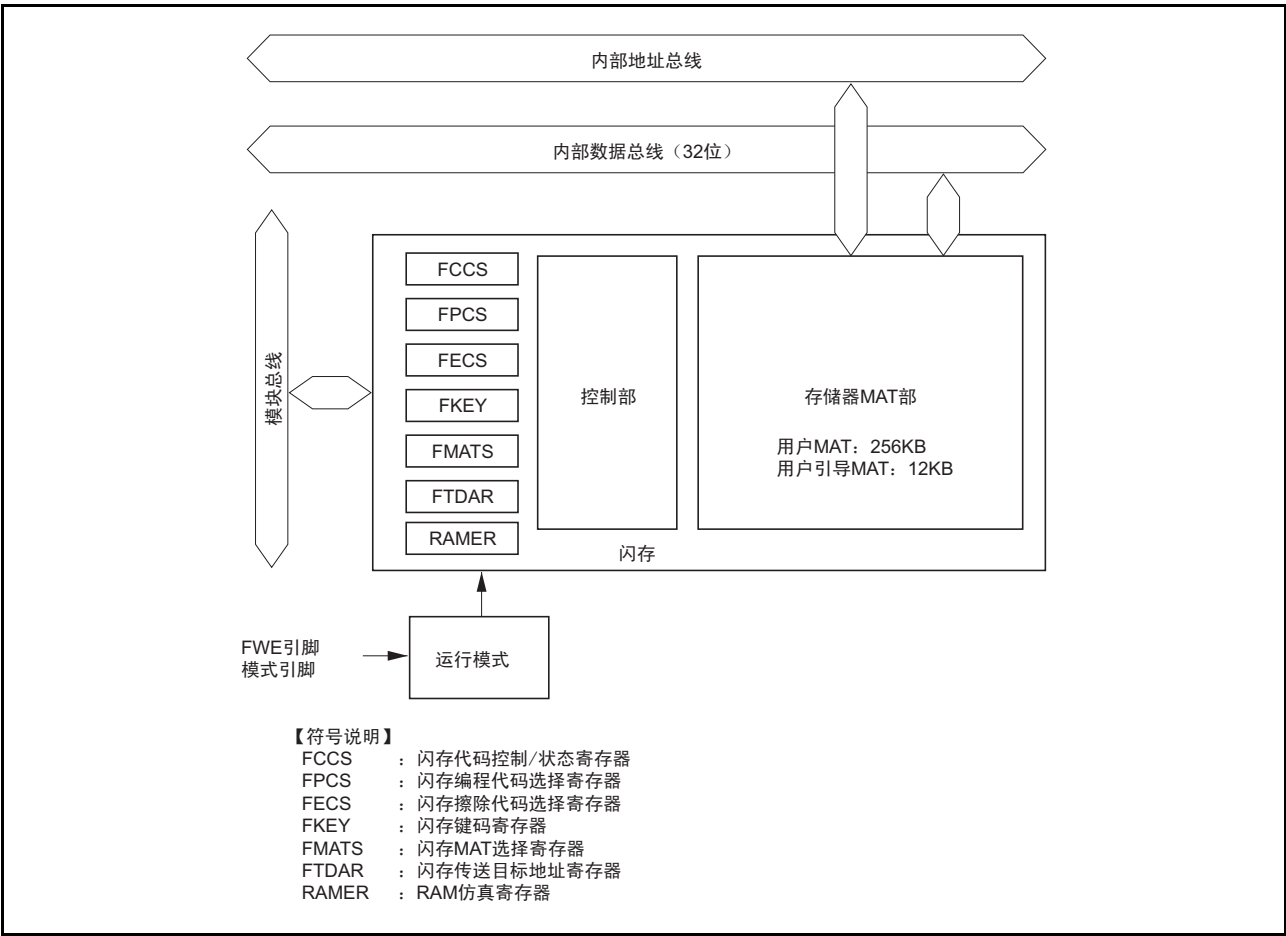


图 22.1 闪存的框图

22.2.2 运行模式

如果在复位状态下设定各模式引脚和 FWE 引脚进行复位解除，单片机就转移到图 22.2 所示的各运行模式。各模式引脚和 FWE 引脚的设定请参照表 22.1。

1. 在ROM无效模式中，不能读、编程和擦除闪存，也不能写编程/擦除接口寄存器，读取值总是H'00。
2. 在用户模式中能读闪存，但是不能编程和擦除。
3. 在用户编程模式、用户引导模式和引导模式中，能对闪存进行板上读、编程和擦除。
4. 在编程器模式中，使用**PROM**编程器对闪存进行读、编程和擦除。

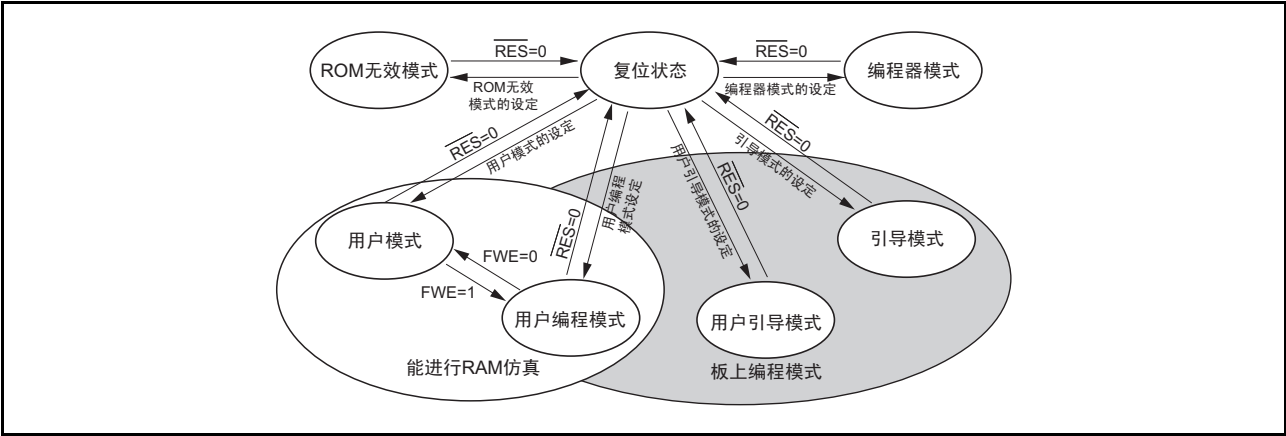


图 22.2 与闪存有关的模式转移图

表 22.1(1) FWE 引脚、MD 引脚的设定和运行模式（SH7136）

引脚	复位状态	用户模式	用户编程模式	引导模式	编程器模式
$\overline{\text{RES}}$	0	1	1	1	取决于专用的 PROM 编程器的条件。
FWE	0/1	0	1	1	
MD1	0/1	1	1	0	

【注】 * SH7136 没有外部总线扩展模式和用户引导模式。

表 22.1(2) FWE 引脚、MD 引脚的设定和运行模式（SH7137）

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	取决于专用的 PROM 编程器的条件。
FWE	0/1	0	0	1	1	1	
MD0	0/1	0*1	0/1*2	0/1*2	1	0	
MD1	0/1	0	1	1	0	0	

【注】 *1 MD0=0: 8 位外部总线。

*2 MD0=0: 能使用外部总线， MD0=1: 单芯片模式（不能使用外部总线）。

22.2.3 模式比较

有关引导模式、用户编程模式、用户引导模式和编程器模式的编程 / 擦除关联项目的比较如表 22.2 所示。

表 22.2 编程模式的比较

	引导模式	用户编程模式	用户引导模式	编程器模式
编程 / 擦除环境	板上编程			板外编程
可编程 / 擦除的 MAT	用户 MAT 用户引导 MAT	用户 MAT	用户 MAT	用户 MAT 用户引导 MAT
编程 / 擦除控制	命令方式	编程 / 擦除接口	编程 / 擦除接口	—
全面擦除	○ (自动)	○	○	○ (自动)
块分割的擦除	○ *1	○	○	×
编程数据的传送	从主机经由 SCI	从任意器件经由 RAM	从任意器件经由 RAM	经由编程器
用户分支功能	×	○	○	×
RAM 仿真	×	○	×	×
复位开始时的启动 MAT	嵌入式程序保存 MAT	用户 MAT	用户引导 MAT*2	嵌入式程序保存 MAT
向用户模式的转移	更改模式设定以及复位	更改 FWE 设定	更改模式设定以及复位	—

【注】 *1 先进行全面擦除，然后才能擦除特定块。

*2 先从嵌入式程序保存 MAT 启动，在检查了闪存关联寄存器后，从用户引导 MAT 的复位向量启动。

- 只能在引导模式和编程器模式中进行用户引导 MAT 的编程/擦除。
- 在引导模式中，先全面擦除用户 MAT 和用户引导 MAT，然后才能通过命令方式进行用户 MAT 或者用户引导 MAT 的编程，但是在进入此状态之前不能读 MAT 的内容。
- 只进行用户引导 MAT 的编程，并且在用户引导模式中改写用户 MAT 或者因不使用用户引导模式而只改写用户 MAT。
- 在用户引导模式中，能通过不同于用户编程模式的模式引脚设定，实现任意接口的引导操作。

22.2.4 闪存结构

本 LSI 的闪存由 256KB 的用户 MAT 和 12KB 的用户引导 MAT 构成。

因为用户 MAT 和用户引导 MAT 的起始地址分配在相同的地址，所以在程序执行或者数据存取跨越 2 个 MAT 时，需要通过 FMATS 寄存器进行 MAT 转换。

只要是 ROM 有效模式，无论在哪个模式中都能读用户 MAT 和用户引导 MAT。但是只能在引导模式和编程器模式中改写引导 MAT。

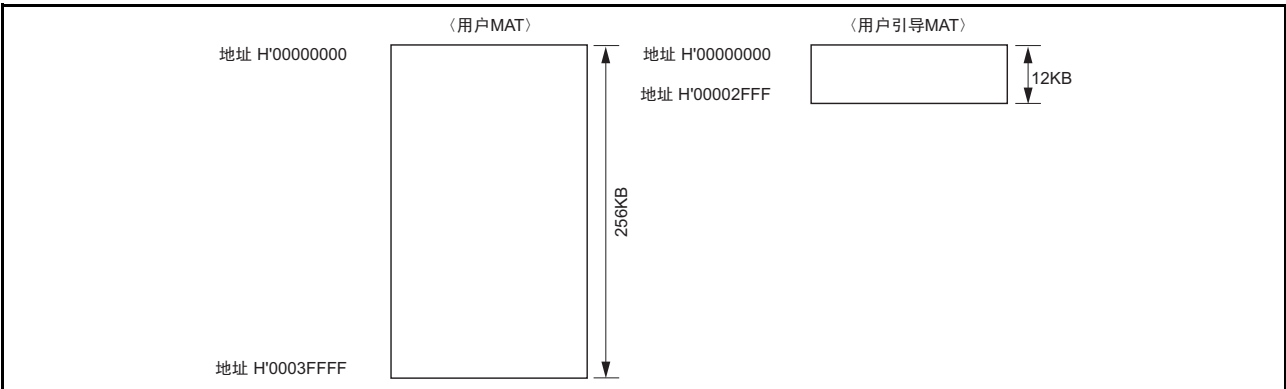


图 22.3 闪存结构图

用户 MAT 和用户引导 MAT 的存储容量不同，所以不能存取超过 12KB 空间的用户引导 MAT，如果读超过 12KB 的用户引导 MAT，读取值就为不定值。

22.2.5 块分割

如图 22.4 所示，用户 MAT 分割为 64KB（3 块）、32KB（1 块）和 4KB（8 块）。能以此分割块为单位进行擦除，在擦除时用 EB0 ~ EB11 指定擦除的块号。

4KB 分割的 8 块是可进行 RAM 仿真的区域。

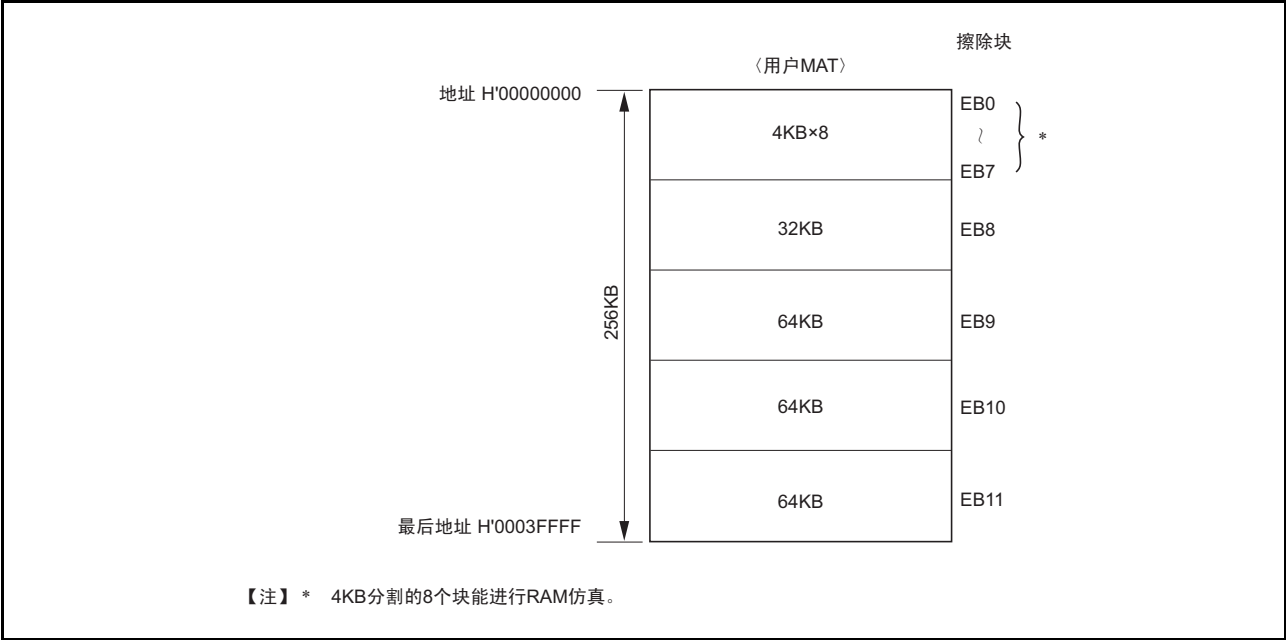


图 22.4 用户 MAT 的块分割

22.2.6 编程 / 擦除的接口

先将内部程序下载到内部 RAM，然后通过接口寄存器和参数指定编程地址、数据和擦除块等进行编程和擦除。

在用户编程模式或者用户引导模式中，这一连串的过程程序由用户建立，步骤的概要如下所示，详细内容请参照“22.5.2 用户编程模式”。

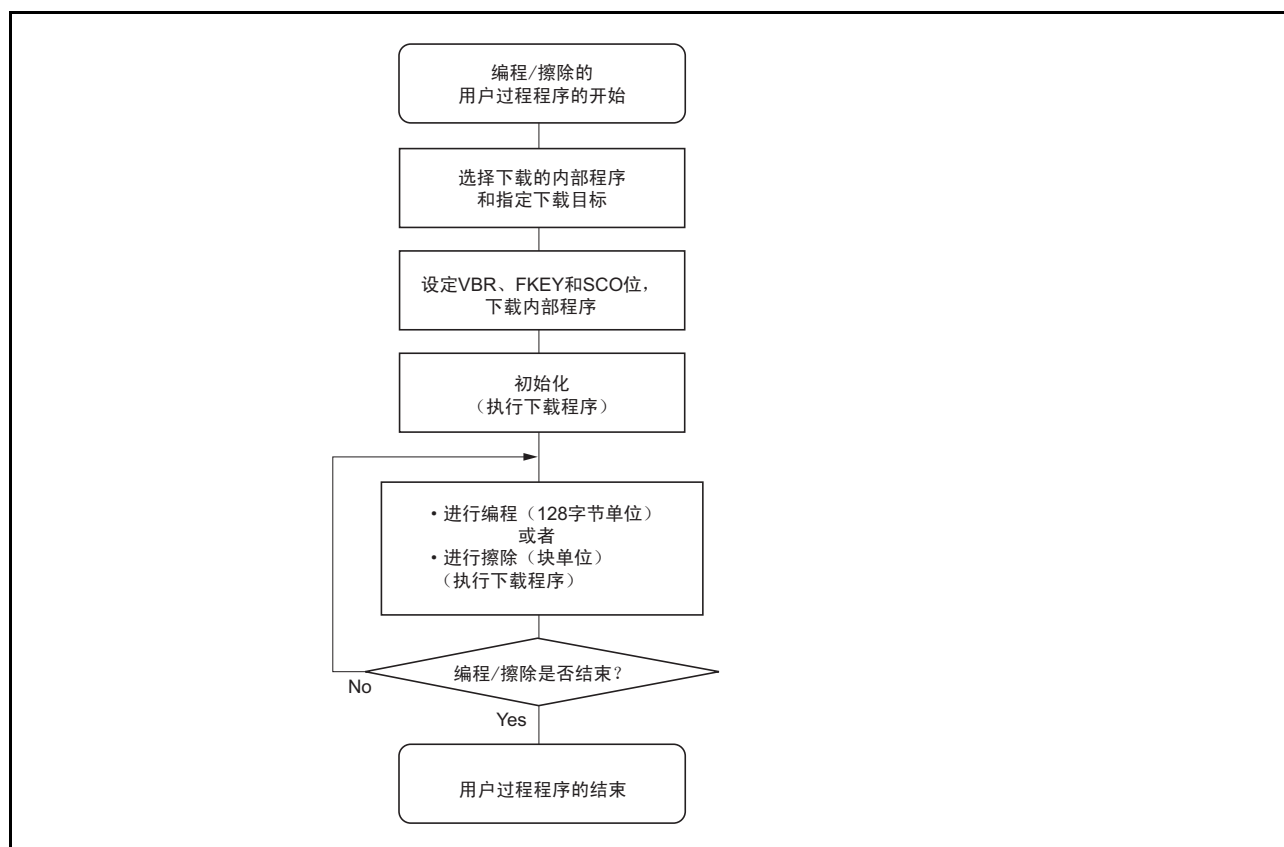


图 22.5 用户过程程序的概要

(1) 下载的内部程序的选择和下载目标的指定

本 LSI 内置编程 / 擦除的相关程序，能将这些程序下载到内部 RAM。通过将编程 / 擦除接口寄存器的对应位置位，选择要下载的内部程序，还能通过 FTDAR 寄存器指定下载目标的地址。

(2) 下载内部程序

在将 CPU 的 VBR 寄存器置 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。

在下载过程中，闪存 MAT 和嵌入式程序保存区调换。因为不能在编程 / 擦除时读闪存，所以必须在闪存以外的空间（内部 RAM 等）执行被下载的的一连串（编程 / 擦除结束为止）过程程序。

因为下载的结果能返回给编程 / 擦除接口参数，所以能确认是否正常下载。

另外，在下载结束后能更改 VBR。

(3) 编程 / 擦除的初始化

在进行编程 / 擦除前，设定工作频率和用户分支。不能将用户分支目标区设定在内部闪存区和下载内部程序的区域。通过编程 / 擦除接口参数进行设定。

(4) 编程 / 擦除

为了进行编程 / 擦除，必须将 FWE 引脚设定为高电平，将模式设定为用户编程模式。

在编程时，以 128 字节为单位指定编程数据和编程目标地址。

在擦除时，以 1 个擦除块为单位指定擦除块。

通过编程 / 擦除接口参数，设定这些指定并启动内部程序。通过 JSR 指令或者 BSR 指令调用（子程序调用）内部 RAM 中的特定地址，执行内部程序。执行结果返回给编程 / 擦除接口参数。

在编程闪存时，需要预先擦除对象区。

编程 / 擦除处理中的中断处理有一些限制和注意事项，详细内容请参照“22.8.2 编程 / 擦除过程中的中断”。

(5) 连续编程 / 擦除的情况

如果 128 字节的编程或者 1 块的擦除处理未结束，就需要更新编程地址 / 数据或者擦除块号，连续进行编程 / 擦除。

因为在处理结束后下载的内部程序仍留在内部 RAM 中，所以在连续进行相同的处理时不需要下载和初始化。

22.3 输入 / 输出引脚

闪存的控制引脚如表 22.3 所示。

表 22.3 引脚结构

名称	引脚名	输入 / 输出	功能
上电复位	RES	输入	复位
闪存编程的允许	FWE	输入	闪存改写的硬件保护
模式 1	MD1	输入	设定本 LSI 的运行模式
模式 0*	MD0	输入	设定本 LSI 的运行模式
发送数据	TXD1 (PA4)	输出	输出串行发送数据（在引导模式中使用）
接收数据	RXD1 (PA3)	输入	输入串行接收数据（在引导模式中使用）

【注】 * SH7136 没有模式 0。

22.4 寄存器说明

22.4.1 寄存器一览表

内部闪存有效时的闪存控制寄存器 / 参数如表 22.4 所示。

在存取闪存时，有读模式和写模式等几种运行模式，存储器 MAT 也有用户 MAT 和用户引导 MAT，根据各种运行模式和 MAT 选择，分配了专用的寄存器 / 参数。运行模式和使用的寄存器 / 参数的对应表如表 22.5 所示。

表 22.4(1) 寄存器结构

寄存器名	略称 *4	R/W	初始值	地址	存取长度
闪存代码控制 / 状态寄存器	FCCS	R、W*1	H'00*2 H'80*2	H'FFFFCC00	8
闪存编程代码选择寄存器	FPCS	R/W	H'00	H'FFFFCC01	8
闪存擦除代码选择寄存器	FECS	R/W	H'00	H'FFFFCC02	8
闪存键代码寄存器	FKEY	R/W	H'00	H'FFFFCC04	8
闪存 MAT 选择寄存器	FMATS	R/W	H'00*3 H'AA*3	H'FFFFCC05	8
闪存传送目标地址寄存器	FTDAR	R/W	H'00	H'FFFFCC06	8
RAM 仿真寄存器	RAMER	R/W	H'0000	H'FFFFFF108	16

【注】 *1 SCO 位为只写位（读取值总是 0），其他位为只读位。

*2 在给 FWE 引脚输入低电平时，FWE 位的初始值为 0。

在给 FWE 引脚输入高电平时，FWE 位的初始值为 1。

*3 在用户模式或者用户编程模式中启动时，初始值为 H'00。

在用户引导模式中启动时，初始值为 H'AA。

*4 RAMER 寄存器除外的各寄存器只在字节存取时有效。

RAMER 寄存器能进行字节存取和字存取。

表 22.4(2) 参数结构

参数名	略称	R/W	初始值	分配	存取长度
下载的成功 / 失败结果	DPFR	R/W	不定值	内部 RAM*	8、16、32
闪存的成功 / 失败结果	FPFR	R/W	不定值	CPU 的 R0	8、16、32
闪存多用途地址区	FMPAR	R/W	不定值	CPU 的 R5	8、16、32
闪存多用途数据目标区	FMPDR	R/W	不定值	CPU 的 R4	8、16、32
闪存擦除块的选择	FEBS	R/W	不定值	CPU 的 R4	8、16、32
闪存编程 / 擦除的频率控制	FPEFEQ	R/W	不定值	CPU 的 R4	8、16、32
闪存用户分支地址的设定	FUBRA	R/W	不定值	CPU 的 R5	8、16、32

【注】 * FTDAR 寄存器指定的内部 RAM 区起始地址的 1 个字节有效。

表 22.5 寄存器 / 参数和对象模式

		下载	初始化	编程	擦除	读	RAM 仿真
编程 / 擦除接口寄存器	FCCS	○	—	—	—	—	—
	FPCS	○	—	—	—	—	—
	PECS	○	—	—	—	—	—
	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 在用户引导模式中，对用户 MAT 进行编程 / 擦除时需要设定。

*2 根据启动模式和读对象 MAT 的组合，有可能需要设定。

22.4.2 编程 / 擦除接口寄存器

编程 / 擦除接口寄存器都是 8 位寄存器，并且只能进行字节存取。

(1) 闪存代码控制 / 状态寄存器 (FCCS)

FCCS 由 FWE 引脚状态的监视位、闪存编程 / 擦除中的错误监视位以及下载内部程序的请求位构成。

位: 7	6	5	4	3	2	1	0
FWE	MAT	—	FLER	—	—	—	SCO
初始值: 1/0	1/0	0	0	0	0	0	0
R/W: R	R	R	R	R	R	R	(R)/W

位	位名	初始值	R/W	说明
7	FWE	1/0	R	闪存编程的允许位 FWE 位监视 FWE 引脚（对闪存编程 / 擦除进行硬件保护）的输入电平。根据 FWE 引脚状态，初始值为 0 或者 1。 0: 给 FWE 引脚输入低电平时（硬件保护状态） 1: 给 FWE 引脚输入高电平时
6	MAT	1/0	R	MAT 位 表示选择的是用户 MAT 还是用户引导 MAT。 0: 选择用户 MAT 时 1: 选择用户引导 MAT 时
5	—	0	R	保留位 读写值总是 0。

位	位名	初始值	R/W	说明
4	FLER	0	R	闪存错误 此位表示在对闪存进行编程 / 擦除时发生了错误。如果 FLER=1，闪存就转移到错误保护状态。 在 FLER=1 时，由于闪存内部加有高电压，为了减少对闪存的损伤，必须在长于通常的 100μs 复位输入期间后解除复位。 0：闪存正常运行。闪存的编程 / 擦除保护（错误保护）无效 [清除条件] 当进行上电复位时 1：表示在对闪存进行编程 / 擦除时发生了错误。闪存的编程 / 擦除保护（错误保护）有效 [置位条件] 详细内容请参照“22.6.3 错误保护”。
3 ~ 1	—	全 0	R	保留位 读写值总是 0。
0	SCO	0	(R)/W	源程序的复制操作 此位请求将内部编程 / 擦除程序下载到内部 RAM。如果给此位写 1，FPCS/FECS 寄存器选择的内部程序就自动被下载到 FTDAR 寄存器指定的内部 RAM 区。为了给此位写 1，需要解除 RAM 仿真状态，给 FKEY 寄存器写 H'A5，并且在内部 RAM 中执行。在给此位写 1 后，必须立刻执行 4 条 NOP 指令。 有关下载中的中断和下载时间，请分别参照“22.8.2 编程 / 擦除过程中的中断”和“22.8.3 其他注意事项”。另外，因为在下载结束时此位被清 0，所以读不到此位为 1 的状态。 因为在通过 SCO 位进行下载时，伴有内部程序保存区的存储体转换的特殊中断处理，所以必须在请求下载（SCO=1）前将 VBR 的值置 H'84000000，否则 VBR 就会失控。在确认下载已结束，能更改 VBR。在使用 SCO 功能时，必须使用 FWE 引脚为高电平的模式。 0：内部编程 / 擦除程序没有被下载到内部 RAM。 [清除条件] 下载结束时清除此位。 1：产生将内部编程 / 擦除程序下载到内部 RAM 的请求。 [置位条件] 在满足以下全部条件的状态下写 1 时 • 给 FKEY 寄存器写 H'A5 • 在内部 RAM 中执行 • 非 RAM 仿真模式（RAMER 的 RAMS=0）

(2) 闪存编程代码选择寄存器（FPCS）

FPCS 选择下载与编程有关的内部程序。

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

位	位名	初始值	R/W	说明
7 ~ 1	—	全 0	R	保留位 读写值总是 0。
0	PPVS	0	R/W	编程脉冲的选择 选择编程的程序。 0: 不选择内部编程程序。 [清除条件] 在传送结束时清除此位。 1: 选择内部编程程序。

(3) 闪存擦除代码选择寄存器 (FECS)

FECS 选择下载与擦除有关的内部程序。

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

位	位名	初始值	R/W	说明
7 ~ 1	—	全 0	R	保留位 读写值总是 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] 在传送结束时清除此位。 1: 选择内部擦除程序。

(4) 闪存键代码寄存器 (FKEY)

FKEY 允许内部程序的下载和闪存编程 / 擦除的软件保护。为了下载内部程序而给 SCO 位写 1 前以及在执行已下载的编程 / 擦除程序前，如果没有写键代码，就不能进行各自的处理。

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	初始值	R/W	说明
7 ~ 0	K[7:0]	全 0	R/W	键代码 SCO 位的写操作只在给 FKEY 寄存器写 H'A5 后有效。如果给 FKEY 寄存器写 H'A5 以外的值，就无法给 SCO 位写 1，导致无法将内部程序下载到内部 RAM。 只能在 FKEY 寄存器写 H'5A 后进行闪存的编程 / 擦除。如果给 FKEY 寄存器写 H'5A 以外的值，即使执行内部编程 / 擦除程序也无法进行闪存的编程 / 擦除。 H'A5: 允许写 SCO 位（如果不是 H'A5，就无法将 SCO 置位） H'5A: 允许编程 / 擦除（如果不是 H'5A，就进入软件保护状态） H'00: 初始值

(5) 闪存 MAT 选择寄存器（FMATS）

FMATS 指定是选择用户 MAT 还是选择用户引导 MAT。

位:	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	初始值	R/W	说明
7	MS7	0/1	R/W	MAT 选择 当这些位的值不是 H'AA 时，处于用户 MAT 选择状态；当这些位的值是 H'AA 时，处于用户引导 MAT 选择状态。通过内部 RAM 中的指令给 FMATS 写值，进行 MAT 转换。 必须按照“22.8.1 用户 MAT 和用户引导 MAT 的转换”进行 MAT 转换（即使通过 FMATS 选择用户引导 MAT，也不能在用户编程模式中改写用户引导 MAT。必须在引导模式或者编程器模式中改写用户引导 MAT）。 H'AA: 选择用户引导 MAT（当不是 H'AA 时，处于用户 MAT 选择状态），此值是在用户引导模式中启动时的初始值。 H'00: 是在用户引导模式以外的模式中启动时的初始值（用户 MAT 选择状态）。 【注】 [可编程的条件] 是内部 RAM 中的执行状态。
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 指定内部程序下载目标的内部 RAM 地址。

必须在给 FCCS 寄存器的 SCO 位写 1 前进行此寄存器的设定。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	初始值	R/W	说明
7	TDER	0	R/W	传送目标地址的设定错误 如果 bit6 ~ 0 (TDA6 ~ TDA0) 指定的下载起始地址有错误, 就将此位置 1。判断地址指定错误的方法是: 将 FCCS 寄存器的 SCO 位置 1, 在下载处理结束时, 判断 TDA6 ~ TDA0 的值是否在 H'00 ~ H'04 的范围内。必须在将 SCO 位置 1 前, 将 FTDAR 寄存器的值设定为 H'00 ~ H'04 范围内的值, 并将此位的值置 0。 0: TDA6 ~ TDA0 的设定为正常值 1: TDER、TDA6 ~ TDA0 的设定值为 H'05 ~ H'FF, 表示下载中断。
6 ~ 0	TDA[6:0]	全 0	R/W	传送目标地址 指定下载的起始地址。可设定的值为 H'00 ~ H'04, 并能以 2KB 单位指定内部 RAM 的下载起始地址。 不能设定 H'05 ~ H'7F 的值, 否则在下载处理中此寄存器的 bit7 (TDER) 就被置 1, 并且不进行内部程序的下载。 H'00: 将下载的起始地址置 H'FFFF9000 H'01: 将下载的起始地址置 H'FFFF9800 H'02: 将下载的起始地址置 H'FFFFA000 H'03: 将下载的起始地址置 H'FFFFA800 H'04: 将下载的起始地址置 H'FFFFB000 H'05 ~ H'7F: 不能设定这些值, 否则在下载中 bit7 (TDER) 就会变为 1 并且中断下载处理。

22.4.3 编程 / 擦除接口参数

通过编程 / 擦除接口参数, 给已下载的内部程序指定工作频率、用户分支目标地址、编程数据的保存位置、编程目标地址、擦除块等以及交换处理结果, 此参数使用 CPU 的通用寄存器 (R4、R5 和 R0) 或者内部 RAM 区。初始值为不定值。

在下载时, 保存 CPU 的全部寄存器; 在初始化和执行内部程序时, 保存除 R0 以外的 CPU 寄存器。R0 保存处理结果的返回值。因为堆栈区用作寄存器的保存区和工作区, 所以必须在开始处理时确保堆栈区 (使用的堆栈区容量最大为 128 字节)

编程 / 擦除接口参数用于以下 4 个项目:

1. 下载控制
2. 编程/擦除前的初始化
3. 编程
4. 擦除

各项使用的参数都不相同, 其对应表如表 22.6 所示。

此时, FPFR 参数返回初始化、编程和擦除的处理结果, 但是处理内容不同, 位的含义也不同。请参照各项处理的 FPFR 说明部分。

表 22.6 使用的参数和对象模式

参数名	略称	下载	初始化	编程	擦除	R/W	初始值	分配
下载的成功 / 失败结果	DPFR	○	—	—	—	R/W	不定值	内部 RAM*
闪存的成功 / 失败结果	FPFR	—	○	○	○	R/W	不定值	CPU 的 R0
闪存编程 / 擦除的频率控制	FPEFEQ	—	○	—	—	R/W	不定值	CPU 的 R4
闪存用户分支地址的设定	FUBRA	—	○	—	—	R/W	不定值	CPU 的 R5
闪存多用途地址区	FMPAR	—	—	○	—	R/W	不定值	CPU 的 R5
闪存多用途数据目标区	FMPDR	—	—	○	—	R/W	不定值	CPU 的 R4
闪存擦除块的选择	FEBS	—	—	—	○	R/W	不定值	CPU 的 R4

【注】 * FTDAR 寄存器指定的下载目标起始地址的 1 个字节

(1) 下载控制

通过将 SCO 位置 1，自动下载内部程序。下载的内部 RAM 区是从 FTDAR 寄存器指定的起始地址开始的 3KB 区域。有关内部 RAM 的地址映像请参照图 22.10。

下载控制通过前面所述的编程 / 擦除的接口寄存器进行设定，返回值通过 DPFR 参数传递。

- 下载的成功/失败结果参数（DPFR：FTDAR 寄存器指定的内部 RAM 起始地址的 1 个字节）
这是下载结果的返回值，必须通过此参数的值判断下载是否已执行完毕。由于不能确认是否已将 SCO 位置 1，所以必须在开始下载前（将 SCO 位置 1 前），先将 FTDAR 寄存器指定的内部 RAM 起始地址的 1 字节设定为下载返回值以外的值（H'FF 等），然后才能进行准确的判断。有关下载结果的检查方法，请参照“22.5.2(2.5)”项。

位:	7	6	5	4	3	2	1	0
	—	—	—	—	—	SS	FK	SF
初始值:	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7 ~ 3	—	不定值	R/W	未使用位 返回值为 0。
2	SS	不定值	R/W	源选择的错误检测位 1 次操作只能指定 1 种可下载的内部程序。如果选择 2 种以上（含 2 种）或者不选择、或者选择了未被映像的程序，就会发生错误。 0: 正常选择了下载程序 1: 发生了下载错误（多重选择或者选择了未被映像的程序）
1	FK	不定值	R/W	闪存键寄存器的错误检测位 此位返回 FKEY 寄存器的值是否为 H'A5 的检查结果。 0: FKEY 寄存器的设定值正常（FKEY 为 H'A5） 1: FKEY 寄存器的设定值错误（FKEY 不为 H'A5）
0	SF	不定值	R/W	成功 / 失败位 此位返回下载是否正常结束的检查结果。 0: 下载正常结束（无错误） 1: 下载异常结束（发生错误）

(2) 编程 / 擦除的初始化

下载的编程 / 擦除内部程序包含初始化程序。

在编程 / 擦除中，通过 CPU 指令构成了所定时宽的等待循环，因此需要设定 CPU 的工作频率。另外，因为支持用户分支功能，所以还需要设定用户分支目标地址。

通过初始化程序，将这些设定作为下载的编程 / 擦除程序的参数。

- 闪存编程/擦除的频率控制参数（FPEFEQ：CPU的通用寄存器R4）

这是设定CPU工作频率的参数。

有关本LSI的工作频率范围，请参照“26.3.1 时钟时序”。

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

位: 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F15	F14	F13	F12	F11	F10	F9	F8	F7	F6	F5	F4	F3	F2	F1	F0
初始值: 不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W: R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 16	—	不定值	R/W	未使用位 必须置 0。
15 ~ 0	F15 ~ F0	不定值	R/W	频率设定位 设定 CPU 的工作频率。必须按以下方法计算设定值： 1. 将以 MHz 为单位表示的工作频率取到小数点后 2 位，小数点后的第 3 位四舍五入。 2. 将扩大 100 倍的值转换为 2 进制数，并写到 FPEFEQ 参数（通用寄存器 R4）。例如，当 CPU 的工作频率为 28.882MHz 时，设定值如下： a. 28.882 的小数点后的第 3 位四舍五入，得到 28.88。 b. 28.88×100=2888 转换为 2 进制数，将得到的 B'0000、B'1011、B'0100、B'1000（H'0B48）设定到 R4。

- 闪存用户分支地址的设定参数（FUBRA：CPU的通用寄存器R5）

这是设定用户分支目标地址的参数。能按编程/擦除时的处理单位执行已设定的用户程序。

位: 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UA31	UA30	UA29	UA28	UA27	UA26	UA25	UA24	UA23	UA22	UA21	UA20	UA19	UA18	UA17	UA16
初始值: 不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W: R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位: 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UA15	UA14	UA13	UA12	UA11	UA10	UA9	UA8	UA7	UA6	UA5	UA4	UA3	UA2	UA1	UA0
初始值: 不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W: R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 0	UA31 ~ UA0	不定值	R/W	用户分支目标地址 在不需要用户分支时，必须将这些位置 H'00000000。不能将用户分支目标地址设定在内部闪存区或者传送内部程序的 RAM 区，或者设定在外部总线空间。必须注意：不能分支到没有执行码的区域，以免失控，也不能破坏内部程序的下载区和堆栈区，否则就不能保证闪存的值。 在用户分支目标的处理中，不能启动内部程序的下载程序、初始化程序和编程 / 擦除程序。否则，就不能保证从用户分支目标返回时的编程 / 擦除。不能改写已准备好的编程数据。 必须保存通用寄存器 R8 ~ R15。通用寄存器 R0 ~ R7 不需保存而被直接使用。 另外，不能在用户分支目标的处理中改写编程 / 擦除接口寄存器或者转移到 RAM 仿真模式。 在用户分支处理结束后，必须使用 RTS 指令返回编程 / 擦除程序。有关用户分支处理的执行间隔，请参照“22.8.3(2) 用户分支处理的间隔”。

- 闪存的 success/failure result parameter (FPFR: CPU 的通用寄存器 R0)

在此说明作为初始化处理结果返回值的 FPFR。

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

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

位	位名	初始值	R/W	说明
31 ~ 3	—	不定值	R/W	未使用位 返回值为 0。
2	BR	不定值	R/W	用户分支的错误检测位 此位返回指定的用户分支目标地址是否在下载编程 / 擦除相关程序的保存区外的检查结果。 0: 用户分支地址的设定值正常 1: 用户分支地址的设定值异常
1	FQ	不定值	R/W	频率的错误检测位 此位返回指定的 CPU 工作频率是否在支持工作频率范围内的检查结果。 0: 工作频率的设定值正常 1: 工作频率的设定值异常
0	SF	不定值	R/W	成功 / 失败位 返回初始化是否正常结束的检查结果。 0: 初始化正常结束 (无错误) 1: 初始化异常结束 (发生错误)

(3) 编程

在对闪存进行编程时，需要将用户 MAT 中的编程目标地址和编程数据传递给已下载的编程程序。

1. 必须将用户 MAT 中的编程目标起始地址设定到通用寄存器 R5。此参数称为 FMPAR（闪存多用途地址区参数）。

因为编程数据总是以 128 字节为单位，所以必须将用户 MAT 中的编程起始地址的边界地址低 8 位（A7～A0）置 H'00 或者 H'80。

2. 必须将用户 MAT 的编程数据传送到连续的区域。编程数据必须在能用 CPU 的 MOV.B 指令存取의连续空间内而不能在内部闪存空间内。

如果要写的数数据不满 128 个字节，就必须填充虚码（H'FF），使其成为 128 字节的编程数据。

必须将准备好的编程数据的区域起始地址设定到通用寄存器 R4。此参数称为 FMPDR（闪存多用途数据目标区参数）。

有关编程处理步骤的详细内容，请参照“22.5.2 用户编程模式”。

- 闪存多用途地址区参数（FMPAR：CPU 的通用寄存器 R5）

设定用户 MAT 中的编程目标起始地址。

如果设定闪存空间以外的地址，就发生错误。

另外，编程目标起始地址必须以 128 字节为边界，否则就会发生错误。这些错误反映在 FPFR 参数的 bit1（WA 位）。

位: 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MOA31	MOA30	MOA29	MOA28	MOA27	MOA26	MOA25	MOA24	MOA23	MOA22	MOA21	MOA20	MOA19	MOA18	MOA17	MOA16
初始值: 不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W: R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位: 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOA15	MOA14	MOA13	MOA12	MOA11	MOA10	MOA9	MOA8	MOA7	MOA6	MOA5	MOA4	MOA3	MOA2	MOA1	MOA0
初始值: 不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W: R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 0	MOA31 ~ MOA0	不定值	R/W	MOA31 ~ MOA0 保存用户 MAT 中的编程目标起始地址，从此处指定的用户 MAT 起始地址开始连续写 128 字节，因此指定的编程目标起始地址以 128 字节为边界，并且 MOA6 ~ MOA0 总是 0。

- 闪存多用途数据目标区参数（FMPDR：CPU 的通用寄存器 R4）

设定用户 MAT 中的编程数据保存区的起始地址。如果编程数据的保存目标在闪存内，就发生错误，此错误反映在 FPFR 参数的 bit2（WD 位）。

位: 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MOD31	MOD30	MOD29	MOD28	MOD27	MOD26	MOD25	MOD24	MOD23	MOD22	MOD21	MOD20	MOD19	MOD18	MOD17	MOD16
初始值: 不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W: R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位: 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOD15	MOD14	MOD13	MOD12	MOD11	MOD10	MOD9	MOD8	MOD7	MOD6	MOD5	MOD4	MOD3	MOD2	MOD1	MOD0
初始值: 不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W: R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 0	MOD31 ~ MOD0	不定值	R/W	MOD31 ~ MOD0 保存用户 MAT 中的编程数据保存区的起始地址，从此处指定的起始地址开始将连续 128 字节的数据写到用户 MAT。

- 闪存的失败/成功结果参数（FPFR：CPU 的通用寄存器 R0）
在此说明作为编程处理结果返回值的 FPFR。

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

位: 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—	—	MD	EE	FK	—	WD	WA	SF
初始值: 不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W: R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 7	—	不定值	R/W	未使用位 返回值为 0。
6	MD	不定值	R/W	编程模式相关设定的错误检测位 此位返回 FWE 引脚的输入值为高电平并且不是错误保护状态的检查结果。 在 FWE 引脚为低电平时或者在处于错误保护状态时，此位被置 1。能通过 FCCS 寄存器的 bit7（FWE）和 bit4（FLER）确认这些状态。 有关错误保护状态的转移条件，请参照“22.6.3 错误保护”。 0: FWE、FLER 状态正常（FWE=1、FLER=0） 1: FWE=0 或者 FLER=1，处于不能编程的状态
5	EE	不定值	R/W	编程时的错误检测位 如果因用户 MAT 未被擦除而引起不能对指定的数据进行编程、或者在从用户分支处理返回时一部分闪存相关寄存器被改写，此位就返回 1。在因这些原因而使此位置 1 的情况下，因为用户 MAT 很有可能在中途被改写，所以必须在解除错误原因后从擦除开始重新编程。即使在 FMATS 寄存器的值为 H'AA（用户引导 MAT 选择状态）时进行编程，也会在编程时出现错误。此时，用户 MAT 和用户引导 MAT 都没有被改写。 必须在引导模式或者编程器模式中进行用户引导 MAT 的编程。 0: 编程处理正常结束 1: 编程处理异常结束（不能保证编程结果）
4	FK	不定值	R/W	闪存键寄存器的错误检测位 此位返回开始编程处理前的 FKEY 寄存器值的检查结果。 0: FKEY 寄存器的设定值正常（FKEY 为 H'5A） 1: FKEY 寄存器的设定值错误（FKEY 不为 H'5A）
3	—	不定值	R/W	未使用位 返回值为 0。

位	位名	初始值	R/W	说明
2	WD	不定值	R/W	编程数据地址的错误检测位 如果将闪存区的地址指定为编程数据保存目标的起始地址，就会发生错误。 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	不定值	R/W	编程地址的错误检测位 如果将以下地址指定为编程目标的起始地址，就会发生错误： - 将非闪存区指定为编程目标地址 - 指定的地址不以 128 字节为边界（A6 ~ A0 不为 0） 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	不定值	R/W	成功 / 失败位 此位返回编程处理是否正常结束的检查结果。 0: 正常结束（无错误） 1: 异常结束（发生错误）

(4) 擦除

在进行闪存的擦除时，需要将用户 MAT 中的擦除块号传递给已下载的擦除程序，并设定到 FEBS 参数（通用寄存器 R4）。

从 0 ~ 15 的块号中指定 1 块。

有关擦除处理步骤的详细内容，请参照“22.5.2 用户编程模式”。

- 闪存擦除块的选择参数（FEBS: CPU 的通用寄存器 R4）

指定擦除块号，不能指定多个块号。

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

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

位	位名	初始值	R/W	说明
31 ~ 8	—	不定值	R/W	未使用位 必须置 0。
7 ~ 0	EBS[7:0]	不定值	R/W	设定 0 ~ 11 的范围内的擦除块号。0 和 11 分别对应块 EB0 和块 EB11，如果设定 0 ~ 11（H'00 ~ H'0B）以外的值，就发生错误。

- 闪存的成功 / 失败结果参数（FPFR: CPU 的通用寄存器 R0）

在此说明作为擦除处理结果返回值的 FPFR。

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

位: 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	—	—	MD	EE	FK	EB	—	—	SF
初始值: 不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W: R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 7	—	不定值	R/W	未使用位 返回值为 0。
6	MD	不定值	R/W	擦除模式相关设定的错误检测位 此位返回 FWE 引脚的输入值为高电平并且不处于错误保护状态的检查结果。在 FWE 引脚为低电平时或者处于错误保护状态时，此位被置 1。 能通过 FCCS 寄存器的 bit7 (FWE) 和 bit4 (FLER) 确认这些状态。 另外，有关错误保护状态的转移条件，请参照“22.6.3 错误保护”。 0: FWE、FLER 状态正常 (FWE=1、FLER=0) 1: FWE=0 或者 FLER=1，处于不可擦除的状态
5	EE	不定值	R/W	擦除时的错误检测位 如果不能擦除用户 MAT、或者在从用户分支处理返回时一部分闪存相关寄存器被改写，此位就返回 1。如果因这些原因而使此位置 1，用户 MAT 就很有可能在途中被擦除，因此必须在解除错误原因后重新擦除。 另外，即使在 FMATS 寄存器的值为 H'AA (用户引导 MAT 选择状态) 时进行擦除，也会在擦除时出现错误。此时，用户 MAT 和用户引导 MAT 都没有被擦除，必须在引导模式中或者编程器模式中进行用户引导 MAT 的擦除。 0: 擦除处理正常结束 1: 擦除处理异常结束 (不能保证擦除结果)
4	FK	不定值	R/W	闪存键寄存器的错误检测位 此位返回开始擦除处理前的 FKEY 寄存器值的检查结果。 0: FKEY 寄存器的设定值正常 (FKEY 为 H'5A) 1: FKEY 寄存器的设定值错误 (FKEY 不为 H'5A)
3	EB	不定值	R/W	擦除块的选择错误检测位 此位返回指定擦除块号是否在用户 MAT 的块范围内的检查结果。 0: 擦除块号的设定值正常 1: 擦除块号的设定值异常
2、1	—	不定值	R/W	未使用位 返回值为 0。
0	SF	不定值	R/W	成功 / 失败位 此位返回擦除处理是否正常结束的检查结果。 0: 正常结束 (无错误) 1: 异常结束 (发生错误)

22.4.4 RAM 仿真寄存器（RAMER）

在对用户 MAT 的实时改写进行仿真时，RAMER 设定和一部分内部 RAM 重叠的用户 MAT 区。必须在用户模式或者用户编程模式中进行 RAM 仿真。

有关用户 MAT 区的分割方法，请参照表 22.7。另外，为了准确地执行仿真功能，不能在改写此寄存器后立即存取 RAM 仿真的对象 MAT。否则，就不能保证正常的存取。

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

位	位名	初始值	R/W	说明
15 ~ 4	—	全 0	R	保留位 读写值总是 0。
3	RAMS	0	R/W	RAM 选择 设定是否选择通过 RAM 进行用户 MAT 仿真。当 RAMS=1 时，用户 MAT 的全部块为编程 / 擦除保护状态。 0: 不选择仿真 用户 MAT 全部块的编程 / 擦除保护无效 1: 选择仿真 用户 MAT 全部块的编程 / 擦除保护有效
2 ~ 0	RAM[2:0]	000	R/W	用户 MAT 区的选择 和 bit3 一起使用，选择和内部 RAM 重叠的用户 MAT 区（参照表 22.7）。

表 22.7 RAM 区和用户 MAT 区的重叠

RAM 区	块名	RAMS	RAM2	RAM1	RAM0
H'FFFA000 ~ H'FFFAFFF	RAM 区（4KB）	0	x	x	x
H'00000000 ~ H'00000FFF	EB0（4KB）	1	0	0	0
H'00001000 ~ H'00001FFF	EB1（4KB）	1	0	0	1
H'00002000 ~ H'00002FFF	EB2（4KB）	1	0	1	0
H'00003000 ~ H'00003FFF	EB3（4KB）	1	0	1	1
H'00004000 ~ H'00004FFF	EB4（4KB）	1	1	0	0
H'00005000 ~ H'00005FFF	EB5（4KB）	1	1	0	1
H'00006000 ~ H'00006FFF	EB6（4KB）	1	1	1	0
H'00007000 ~ H'00007FFF	EB7（4KB）	1	1	1	1

【注】 x: Don't care

22.5 板上编程模式

如果将引脚设定为板上编程模式后开始复位，就转移到可编程 / 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、用户引导模式和引导模式 3 种运行模式。

转移到各模式的引脚设定方法请参照表 22.1，闪存的各模式的状态转移图请参照图 22.2。

22.5.1 引导模式

引导模式以使用内部 SCI 从主机发送控制命令和编程数据的方式，对用户 MAT 或者用户引导 MAT 进行编程 / 擦除。需要预先给主机准备发送控制命令的工具和编程数据，并将使用的 SCI 通信模式设定为异步模式。如果在将本 LSI 的引脚设定为引导模式后开始复位，就启动预先嵌入在单片机内部的引导程序，并在自动调整 SCI 位速率后以控制命令方式和主机进行通信。

引导模式时的系统结构如图 22.6 所示，引导模式的引脚设定请参照表 22.1。虽然引导模式的 NMI 和其他中断被忽视，但是也要尽量不让中断发生。

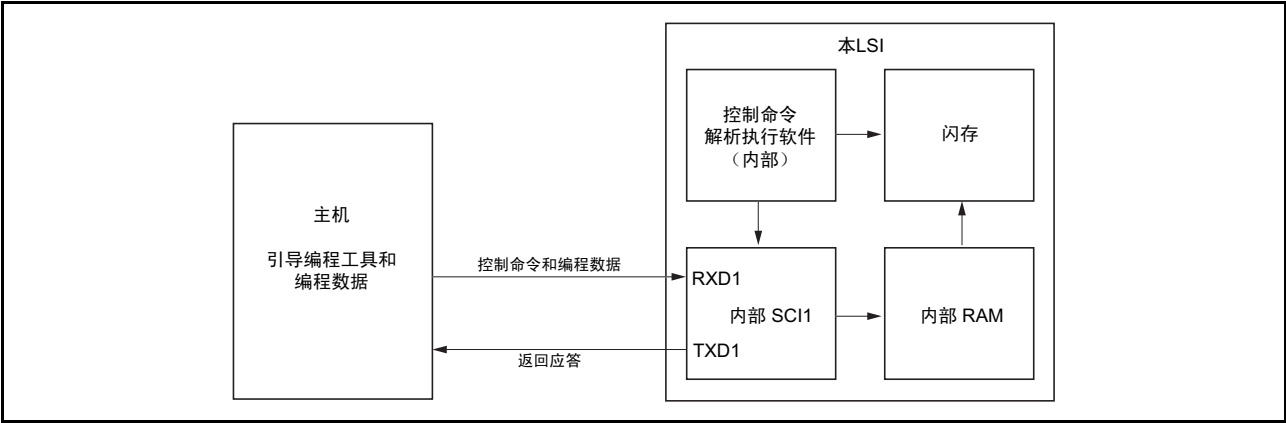


图 22.6 引导模式时的系统结构图

(1) 主机的 SCI 接口设定

一旦启动引导模式，本 LSI 就开始测量主机连续发送来的异步 SCI 通信数据（H'00）的 Low 期间。必须将此时的 SCI 发送 / 接收格式设定为“8 位数据、1 个停止位和无奇偶校验”。本 LSI 根据测量的 Low 期间计算主机发送的位速率，并将位调整结束信号（1 字节的 H'00）发送给主机，主机必须在确认已正常接收到此调整结束信号（H'00）后，向本 LSI 发送 1 字节的 H'55。如果不能正常接收，就必须重新启动（复位）引导模式，进行上述操作。根据主机发送的位速率和本 LSI 的系统时钟频率，主机和本 LSI 的位速率会产生误差，所以，为了使 SCI 正常运行，必须将主机的位速率设定为 9,600bps 或者 19,200bps。

主机的位速率和本 LSI 的位速率能自动匹配的系统时钟的频率如表 22.8 所示，必须在此系统时钟的范围内启动引导模式。在引导模式中，不支持各内部时钟分频比为 $\times 1/3$ 倍的设定。

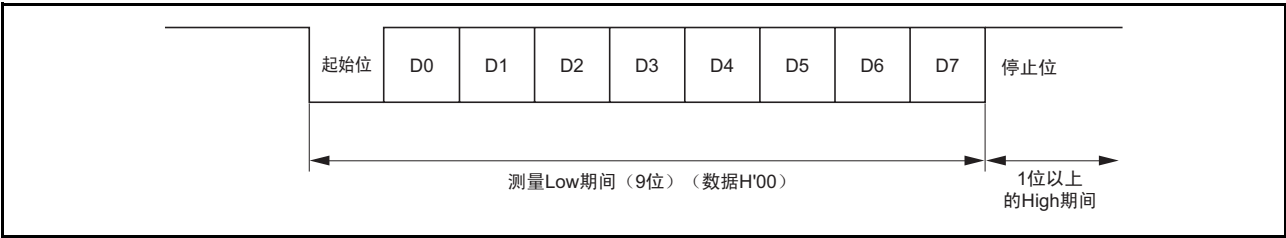


图 22.7 SCI 位速率的自动匹配

表 22.8 本 LSI 能自动匹配的系统时钟频率

主机的位速率	本 LSI 位速率能自动匹配的外围时钟（P ϕ ）频率
9,600bps	10 ~ 40MHz
19,200bps	10 ~ 40MHz

【注】 在引导模式中，不支持各内部时钟分频比为 $\times 1/3$ 倍的设定。

(2) 状态转移图

启动引导模式后的状态转移概要如图 22.8 所示，有关引导模式详细内容，请参照“22.9.1 引导模式的标准串行通信接口规格”。

1. 位速率的匹配

在启动引导模式后，进行主机和 SCI 接口的位速率匹配。

2. 等待查询选择命令

将用户 MAT 容量、用户 MAT 结构、MAT 起始地址和支持信息等所需要的查询信息发送到主机。

3. 自动擦除全部的用户 MAT 和用户引导 MAT

如果在查询选择结束后发送编程/擦除的状态转移命令，就自动擦除全部的用户 MAT 和用户引导 MAT。

4. 等待编程/擦除命令

- 如果接收到“编程选择命令”，就转移到编程数据的等待状态。必须在编程命令之后连续发送编程起始地址和编程数据。在编程结束时，必须在将编程起始地址置 H'FFFFFF 后再发送。据此，从编程数据的等待状态返回到编程/擦除命令的等待状态。
- 如果接收到“擦除选择命令”，就转移到擦除块数据的等待状态。必须在擦除命令之后连续发送擦除块号。在擦除结束时，必须在将擦除块号置 H'FF 后再发送。据此，从擦除块数据的等待状态返回到编程/擦除命令的等待状态。另外，如果在引导模式中进行编程后不开始复位而只改写特定块，就必须进行擦除。在通过 1 次操作就能完成编程的情况下，因为在转移到编程、擦除或者其他命令的等待状态前已擦除全部块，所以不需要此擦除操作。
- 除编程/擦除以外，还有用户 MAT/用户引导 MAT 的校验和命令、用户 MAT/用户引导 MAT 的空白检查（擦除检查）命令、用户 MAT/用户引导 MAT 的存储器读命令以及当前状态的信息取得命令。

必须注意：对于用户 MAT/用户引导 MAT 的存储器读操作，只能在自动擦除全部用户 MAT/用户引导 MAT 后读取已编程的数据。

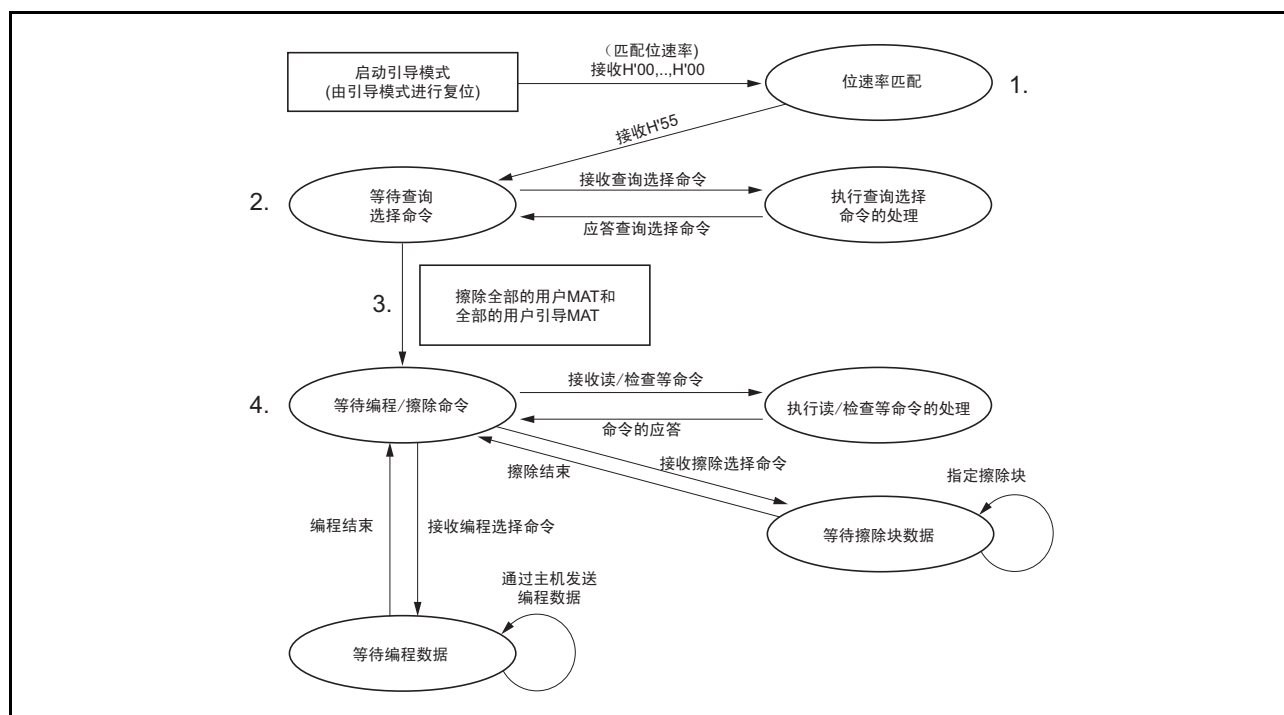


图 22.8 引导模式状态转移的概略图

22.5.2 用户编程模式

能在用户编程模式中对用户 MAT 进行编程 / 擦除（不能对用户引导 MAT 进行编程 / 擦除）。

预先将内部程序下载到单片机内，然后进行编程 / 擦除。

概略流程如图 22.9 所示。

另外，因为在编程 / 擦除处理中闪存内部加有高电压，所以不能在编程 / 擦除处理中进行复位，否则有可能损伤甚至破坏闪存。如果错误地进行了复位，就必须在长于通常的 100 μ s 复位输入期间后解除复位。

有关编程步骤和擦除步骤，请分别参照后述的“(2) 用户程序模式的编程步骤”和“(3) 用户编程模式的擦除步骤”。

有关使用 FTDAR 寄存器，将编程 / 擦除程序分别下载到内部 RAM 区，进行擦除和编程重复处理的概略，请参照后述的“(4) 用户编程模式的擦除 / 编程步骤”。

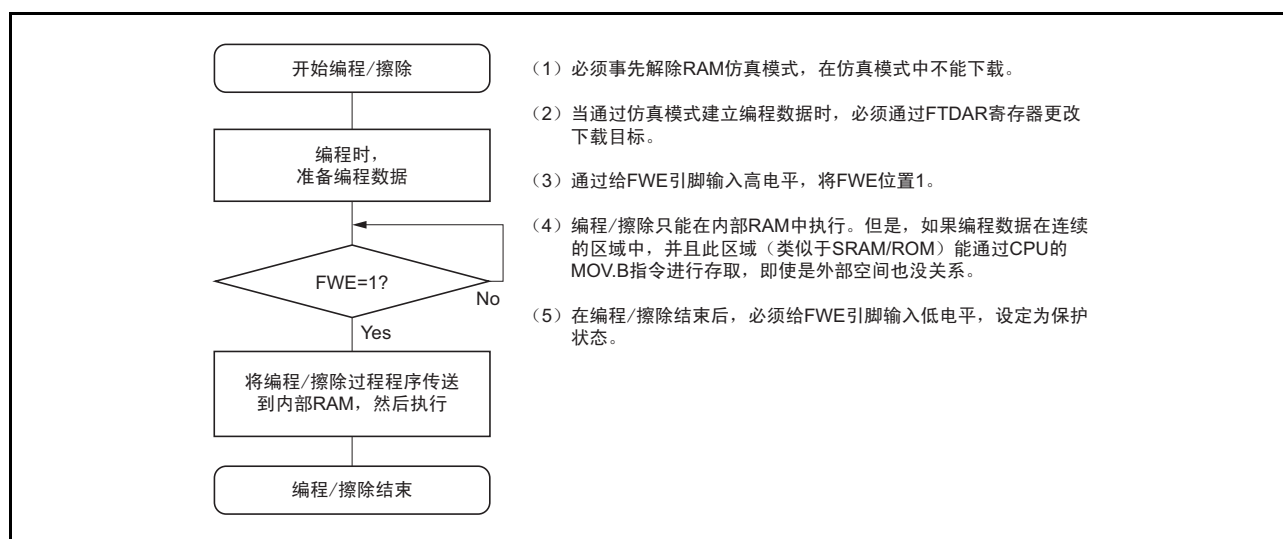


图 22.9 编程 / 擦除的概略流程

(1) 编程 / 擦除时的内部 RAM 的地址映像

必须在内部 RAM 中执行用户建立的下载请求、编程 / 擦除的步骤和结果判断等一部分过程程序。下载的内部程序全部存放在内部 RAM 中，所以，为了使这些程序不重叠，必须注意内部 RAM 的区域管理。

下载的程序区如图 22.10 所示。

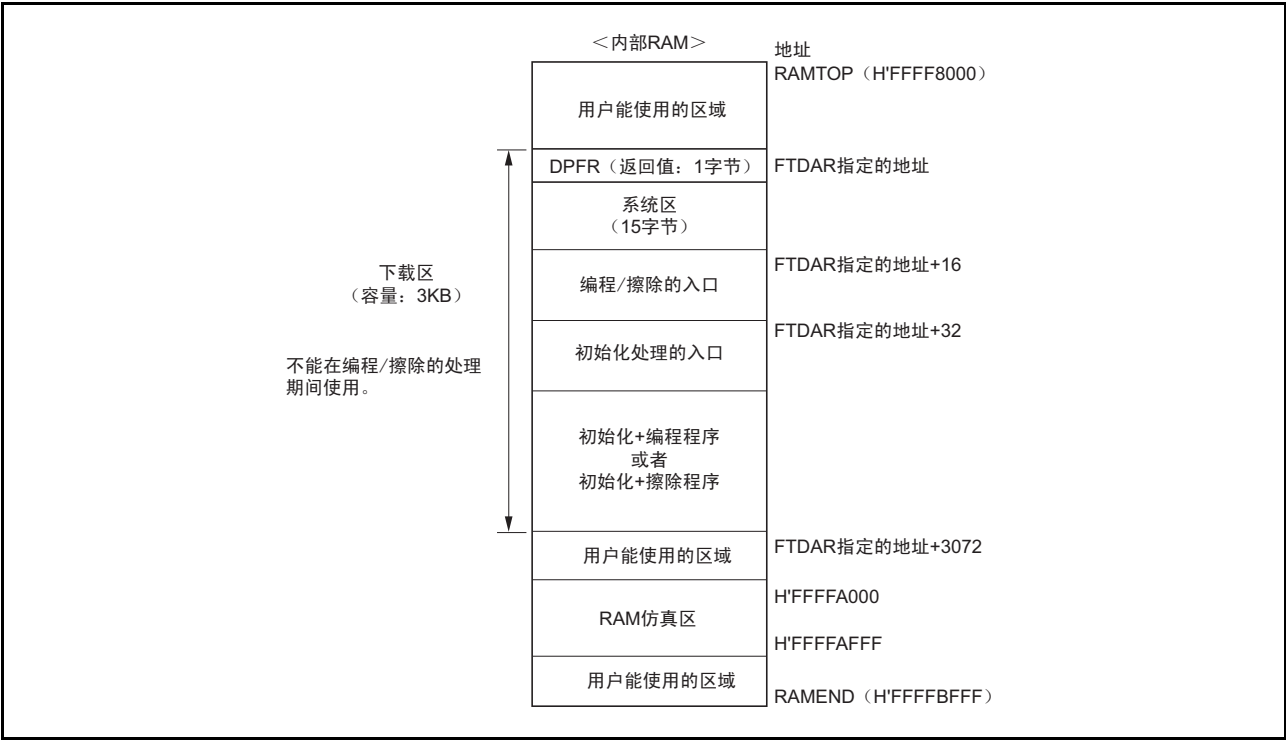


图 22.10 下载后的内部 RAM 映像

(2) 用户程序模式的编程步骤

下载、初始化和编程步骤如图 22.11 所示。

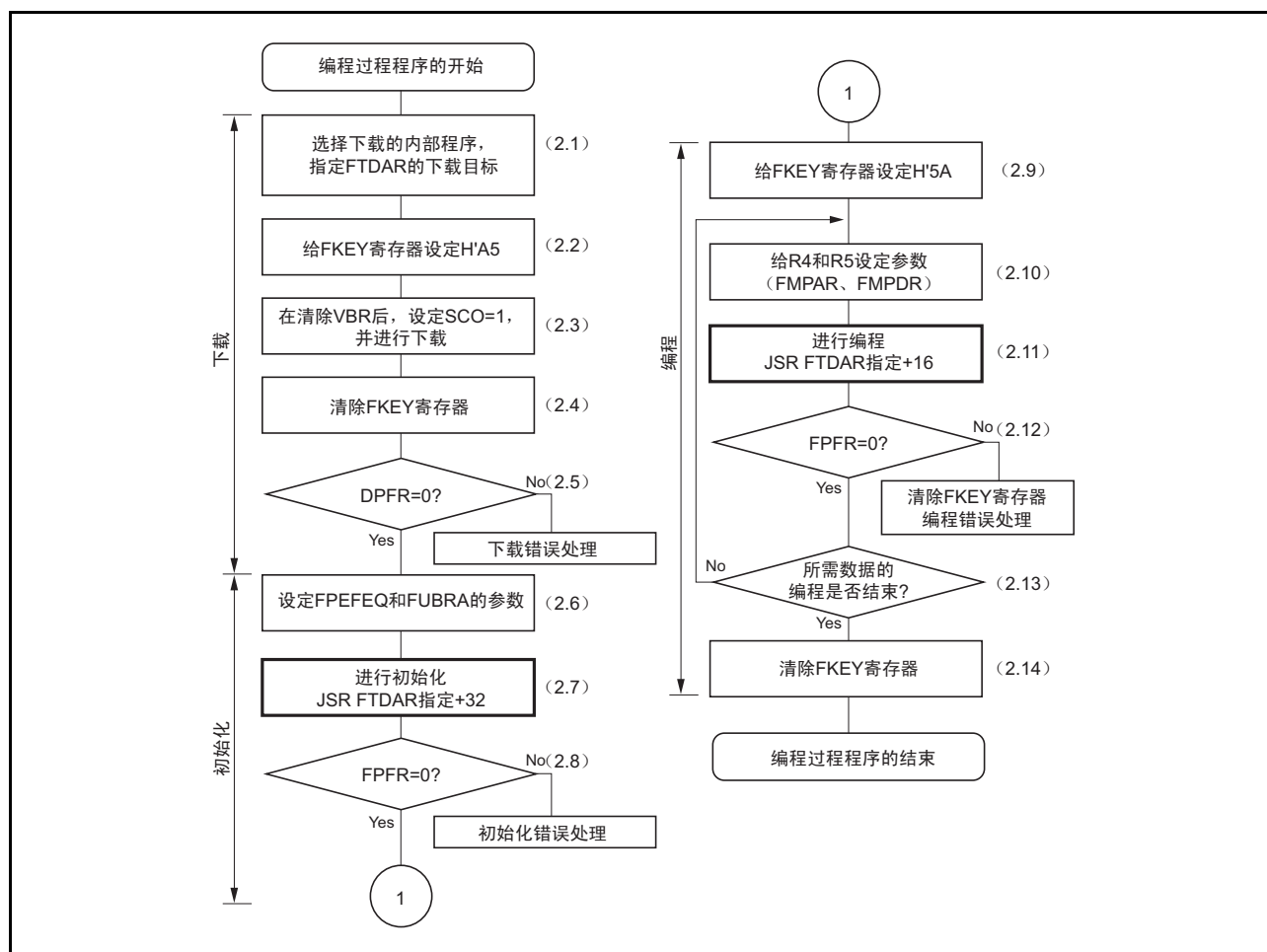


图 22.11 编程步骤

在此说明编程步骤的详细内容。必须在编程对象的闪存以外的区域执行程序，尤其是为了下载而将 FCCS 寄存器的 SCO 位置 1 的部分，必须在内部 RAM 中执行。另外，通过频率控制寄存器（FRQCR）设定的内部时钟（I ϕ ）、总线时钟（B ϕ ）、外围时钟（P ϕ ）的分频比都为 1/4（初始值）。

在编程 / 擦除程序下载结束并且 SCO 位被清 0 后，能将频率控制寄存器（FRQCR）的设定值更改为任意值。

有关能单步执行用户过程程序的区域（内部 RAM、用户 MAT、外部空间等），请参照“22.9.2 过程程序和编程数据的可保存区域”。

以下说明的前提是：已擦除用户 MAT 中的编程对象区，并且编程数据也已存放在连续区。如果还没有擦除，就必须在编程前进行擦除。

1 次编程处理进行 128 字节的编程。如果超过 128 字节，就以 128 字节为单位更新编程目标地址 / 编程数据的参数后，反复进行编程。

如果不满 128 字节，就需要填写无效数据补齐 128 字节。如果填写的无效数据为 H'FF，就能缩短编程处理时间。

(2.1) 选择下载的内部程序以及指定下载的目标地址。

如果将 FPCS 寄存器的 PPVS 位置 1，就选择编程序。

不能选择多个编程 / 擦除程序，否则就不下载而通过 DPFR 参数的源选择检测位（SS）报告下载错误。通过 FTDAR 寄存器指定下载目标的起始地址。

（2.2）给 FKEY 寄存器写 H'A5。

如果为了保护而没有给 FKEY 寄存器写 H'A5，就不能给下载请求的 SCO 位写 1。

（2.3）通过设定 VBR 寄存器并给 FCCS 寄存器的 SCO 位写 1 进行下载。

在将 SCO 位置位前，必须将 VBR 寄存器置 H'84000000。

为了给 SCO 位写 1，需要满足以下的全部条件：

1. 已解除 RAM 仿真模式。
2. 已给 FKEY 寄存器写 H'A5。
3. 已在内部 RAM 中进行 SCO 位的写操作。

因为在 SCO 位为 1 时自动开始下载，在返回到用户的过程程序时 SCO 位被清 0（SCO=0），所以用户过程程序无法确认 SCO=1。

因为只能通过 DPFR 参数的返回值确认下载结果，所以在 SCO=1 前，必须将作为 DPFR 参数的 FTDAR 所指定的内部 RAM 起始 1 个字节设定为返回值以外的值（H'FF 等），以免发生误判断。

在下载时，因为单片机的内部处理是伴有存储体转换的特殊中断处理（如下所示），所以需要将 VBR 置 H'84000000。另外，必须在设定 SCO=1 的指令后立刻执行 4 条 NOP 指令。

1. 将用户 MAT 空间转换到内部程序保存区。
2. 在检查下载程序的选择条件和 FTDAR 的指定地址等后，对 FTDAR 指定的内部 RAM 进行传送处理。
3. 将 FPCS 寄存器、FECS 寄存器和 FCCS 寄存器的 SCO 位清 0。
4. 给 DPFR 参数设定返回值。
5. 在将内部程序保存区返回到用户 MAT 空间后，返回到用户过程程序。

在下载结束并返回到用户过程程序后，能重新设定 VBR。

有关下载的注意事项如下所述：

在下载处理中保存 CPU 通用寄存器的值。

不能在下载处理中发生中断。有关下载和中断请参照“22.8.2 编程 / 擦除过程中的中断”。

因为最多使用 128 字节的堆栈区，所以必须在设定 SCO=1 前至少确保 128 字节的堆栈区。

如果在下载中通过 DTC 存取闪存，就不能保证运行，所以必须注意：不能通过 DTC 进行存取。

（2.4）将 FKEY 寄存器清 0，进行保护。

（2.5）检查 DPFR 参数的值，确认下载结果。

建议使用以下方法确认下载结果：

1. 检查 DPFR 参数（FTDAR 指定的下载目标起始地址的 1 个字节）的值。如果值为 H'00，就表示下载正常进行，否则就按以下步骤检查不能下载的原因。
2. 如果 DPFR 参数值和下载前的设定值（H'FF 等）相同，就可认为 FTDAR 的下载目标地址的设定可能出现异常，因此必须确认 FTDAR 的 bit7（TDER 位）。
3. 如果 DPFR 参数值和下载前的设定值不同，就必须通过 DPFR 参数的 bit2（SS 位）和 bit1（FK 位）确认下载程序的选择或者 FKEY 寄存器的设定是否正常。

（2.6）给 FPEFEQ 和 FUBRA 参数设定工作频率和用户转移目标。

1. 给 FPEFEQ 参数（通用寄存器：R4）设定当前的 CPU 时钟频率。有关 FPEFEQ 参数的可设定范围，请参照“26.3.1 时钟时序”。

如果设定此范围以外的频率，就通过初始化程序的FPFR参数报告错误并且不进行初始化。有关频率的设定方法，请参照“22.4.3 编程/擦除接口参数”的“闪存编程/擦除的频率控制参数（FPEFEQ：CPU的通用寄存器R4）”的说明。

2. 给FUBRA参数（通用寄存器：R5）设定用户转移目标的起始地址。

在不需要用户分支处理时，必须将FUBRA置0。

在进行用户分支时，必须在编程对象的闪存以外的区域执行转移目标，并且不能设定到下载内部程序的区域。

必须使用RTS指令从用户分支处理返回到编程处理。

请参照“22.4.3 编程/擦除接口参数”的“闪存用户分支地址的设定参数（FUBRA：CPU的通用寄存器R5）”的说明。

（2.7）初始化

在下载编程程序时，初始化程序一起被下载到内部RAM。因为在从FTDAR设定的下载起始地址+32字节开始的区域中有初始化程序的入口点，所以必须通过以下方法调用子程序进行初始化：

MOV.L	#DLTOP+32, R1	;	将入口地址设定到 R1
JSR	@R1	;	调用初始化程序
NOP			

1. 在初始化程序中保存R0以外的通用寄存器。
2. R0是FPFR参数的返回值。
3. 因为在初始化程序中使用堆栈区，所以必须预先在RAM中至少确保128字节的堆栈区。
4. 能在执行初始化程序中接受中断，但是不能破坏内部RAM中的程序保存区、堆栈区和寄存器的值。

（2.8）判断初始化程序的返回值 FPFR（通用寄存器 R0）。

（2.9）必须给 FKEY 寄存器设定 H'5A，以便对用户 MAT 进行编程。

（2.10）设定编程所需要的参数。

将用户MAT的编程目标起始地址（FMPAR）和编程数据保存区的起始地址（FMPDR）分别设定到通用寄存器R5和R4。

1. FMPAR的设定

因为FMPAR能指定编程起始地址，所以在指定用户MAT区以外的地址时，即使执行编程程序也不进行编程，而通过返回值参数FPFR报告错误。因为以128字节为单位，所以低8位（MOA7～MOA0）需要为H'00或者H'80的128字节的边界。

2. FMPDR的设定

当编程数据的保存目标在闪存中时，即使执行编程程序也不进行编程，而通过FPFR参数报告错误。此时必须暂时将编程数据传送到内部RAM，然后进行编程。

（2.11）编程处理

因为在从FTDAR指定的下载目标起始地址+16字节开始的区域中有编程程序的入口点，所以必须通过以下方法调用子程序进行编程处理：

MOV.L	#DLTOP+16, R1	;	将入口地址设定到 R1
JSR	@R1	;	调用编程程序
NOP			

1. 在编程序序中保存 R0 以外的通用寄存器。
2. R0 是 FPFR 参数的返回值。
3. 因为在编程序序中使用堆栈区，所以必须预先在 RAM 中至少确保 128 字节的堆栈区。

(2.12) 判断编程序序的返回值 FPFR (通用寄存器 R0)。

(2.13) 判断所需数据的编程是否已结束。

在编程超过 128 字节的数据时，以 128 字节为单位更新 FMPAR 和 FMPDR 的设定，然后重复上述 (2.10) ~ (2.13) 的处理。必须正确递增编程目标地址的 128 字节以及更新编程数据的指针。如果对已编程的地址重复编程，不仅会产生编程错误，还会损伤闪存。

(2.14) 必须在编程结束后清除 FKEY 寄存器并加软件保护。

如果在用户 MAT 的编程结束后立刻通过上电复位重新启动，就必须设定长于通常 100 μ s 的复位期间 ($\overline{\text{RES}}=0$ 的期间)。

(3) 用户编程模式的擦除步骤

下载、初始化和擦除步骤如图 22.12 所示。

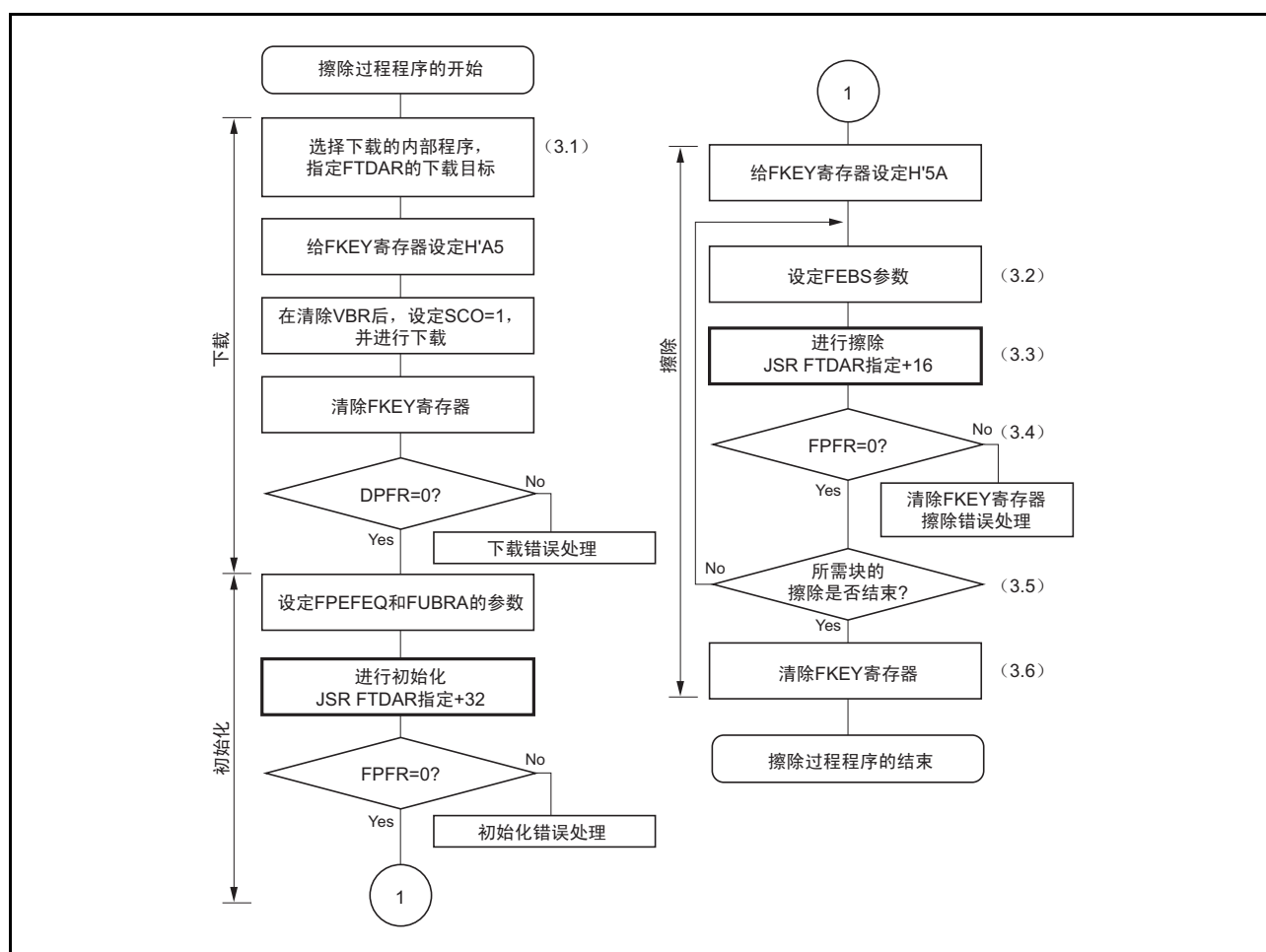


图 22.12 擦除步骤

在此说明擦除步骤的详细内容。必须在擦除对象的闪存以外的区域执行程序。尤其是为了下载而将 FCCS 寄存器的 SCO 位置 1 的部分，必须在内部 RAM 中执行。另外，通过频率控制寄存器（FRQCR）设定内部时钟（I ϕ ）、总线时钟（B ϕ ）、外围时钟（P ϕ ）的分频比都为 1/4（初始值）。

在编程 / 擦除程序下载结束并且 SCO 位被清 0 后，能将频率控制寄存器（FRQCR）的设定值更改为任意值。

有关能单步执行用户过程程序的区域（内部 RAM、用户 MAT、外部空间等），请参照“22.9.2 过程程序和编程数据的可保存区域”。

有关下载的内部程序的区域，请参照图 22.10 的下载后的内部 RAM 映像。

1 次擦除处理进行 1 个分割块的擦除，有关块分割的详细内容请参照图 22.4。在擦除 2 块以上（包含 2 块）的分割块时，要先更新擦除块号，然后重复擦除。

（3.1）选择下载的内部程序以及指定下载的目标地址。

将 FECS 寄存器的 EPVB 位置 1。

不能选择多个编程 / 擦除程序，否则就不下载而通过 DPFR 参数的源选择错误检测位（SS）报告下载错误。通过 FTDAR 寄存器指定下载目标的起始地址。

FKEY 寄存器设定以后的下载、初始化等过程和编程步骤相同，请参照“22.5.2(2) 用户程序模式的编程步骤”。

设定擦除程序参数以后的步骤如下所示：

（3.2）设定擦除时所需要的 FEBS 参数。

给闪存擦除块的选择参数 FEBS（通用寄存器 R4）设定用户 MAT 的擦除块号。如果设定用户 MAT 分割块号以外的值，即使执行擦除处理程序也不擦除，而通过返回值参数 FPFR 报告错误。

（3.3）擦除处理

和编程相同，因为在从 FTDAR 指定的下载目标起始地址 +16 字节开始的区域中有擦除程序的入口点，所以必须通过以下方法调用子程序进行擦除处理：

MOV.L	#DLTOP+16, R1	;	将入口地址设定到 R1
JSR	@R1	;	调用擦除程序
NOP			

1. 在擦除程序中保存 R0 以外的通用寄存器。
2. R0 是 FPFR 参数的返回值。
3. 因为在擦除程序中使用堆栈区，所以必须预先在 RAM 中至少确保 128 字节的堆栈区。

（3.4）判断擦除程序的返回值 FPFR（通用寄存器 R0）。

（3.5）判断所需块的擦除是否已结束。

在擦除多个块时，更新 FEBS 参数的设定，然后重复上述（3.2）～（3.5）的处理。能对已擦除的块进行擦除。

（3.6）必须在结束擦除后清除 FKEY 寄存器并加软件保护。

如果在用户 MAT 的擦除结束后立刻通过上电复位重新启动，就必须设定长于通常 100 μ s 的复位期间（RES=0 的期间）。

(4) 用户编程模式的擦除 / 编程步骤

能通过 FTDAR 寄存器更改下载目标的内部 RAM 地址，将擦除程序和编程程序预先下载到各自的内部 RAM 区。

重复进行 RAM 仿真、擦除 / 编程时的使用例子如图 22.13 所示。

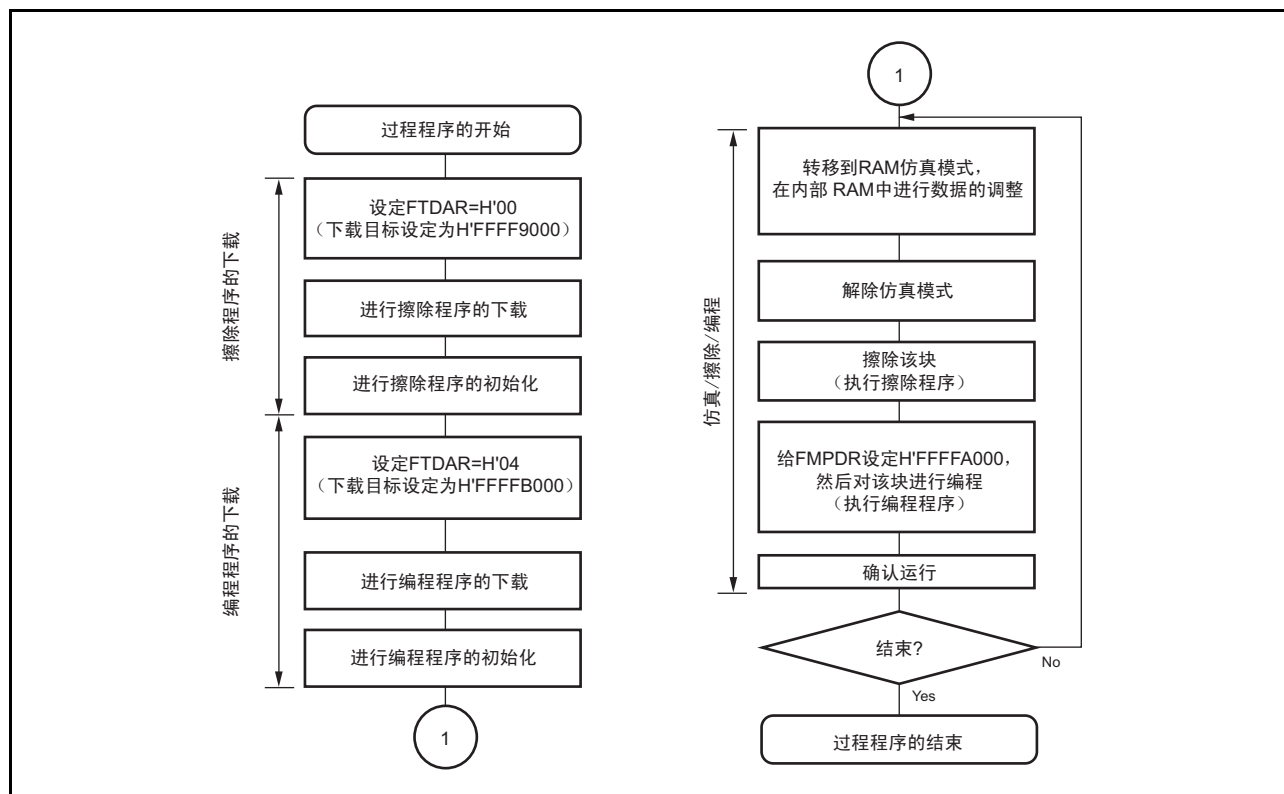


图 22.13 RAM 仿真、擦除 / 编程的重复例子 (概要)

在此例中，为了进行 RAM 仿真而将擦除 / 编程程序下载到 H'FFFA000 ~ H'FFFAFFF 以外的区域。另外，只在最初进行 1 次下载和初始化。

在执行本例的步骤时，必须注意以下事项：

1. 必须注意：不要重复破坏内部 RAM 区。

在内部 RAM 中除了 RAM 仿真区、擦除程序区和编程程序区以外，还有用户建立的过程程序区、作业区、堆栈区等，因此不能破坏这些区域。

2. 必须进行擦除程序和编程程序的初始化。

必须在擦除程序和编程程序中进行 FPEFEQ 参数和 FUBRA 参数设定的初始化。初始化的入口地址必须为擦除程序的下载起始地址+32（此例中为 H'FFFF9020）、编程程序的下载起始地址+32（此例中为 H'FFFFB820）。

22.5.3 用户引导模式

本 LSI 有通过不同于用户编程模式和引导模式的引脚设定进行用户引导模式的启动，能实现与使用内部 SCI 的引导模式不同的用户任意引导模式。

在用户引导模式中，只能对用户 MAT 进行编程 / 擦除。必须在引导模式或者编程器模式中对用户引导 MAT 进行编程 / 擦除。

(1) 用户引导模式的启动

用于启动用户引导模式的引脚设定如表 22.1 所示。

如果在用户引导模式中开始复位，就执行闪存关联寄存器的检查程序。此程序使用的 RAM 容量约为 1.2KB（从地址 H'FFFF9800 开始），堆栈使用 4 个字节（从地址 H'FFFFAFFC 开始）。在此程序执行期间，不接受 NMI 和其他中断。在以 40MHz 的内部频率运行时，此程序的执行时间约为 100μs。

然后，从用户引导 MAT 中的复位向量的执行起始地址开始处理。因为此时执行的 MAT 为用户引导 MAT，所以给闪存 MAT 选择寄存器 FMATS 设定 H'AA。

(2) 用户引导模式中的用户 MAT 编程

在用户引导模式中对用户 MAT 进行编程时，需要追加以下步骤：

- 通过 FMATS 寄存器从用户引导 MAT 选择状态转换到用户 MAT 选择状态
- 在编程结束后，再次从用户 MAT 选择状态返回到用户引导 MAT 选择状态

在用户引导模式中对用户 MAT 进行编程的步骤如图 22.14 所示。

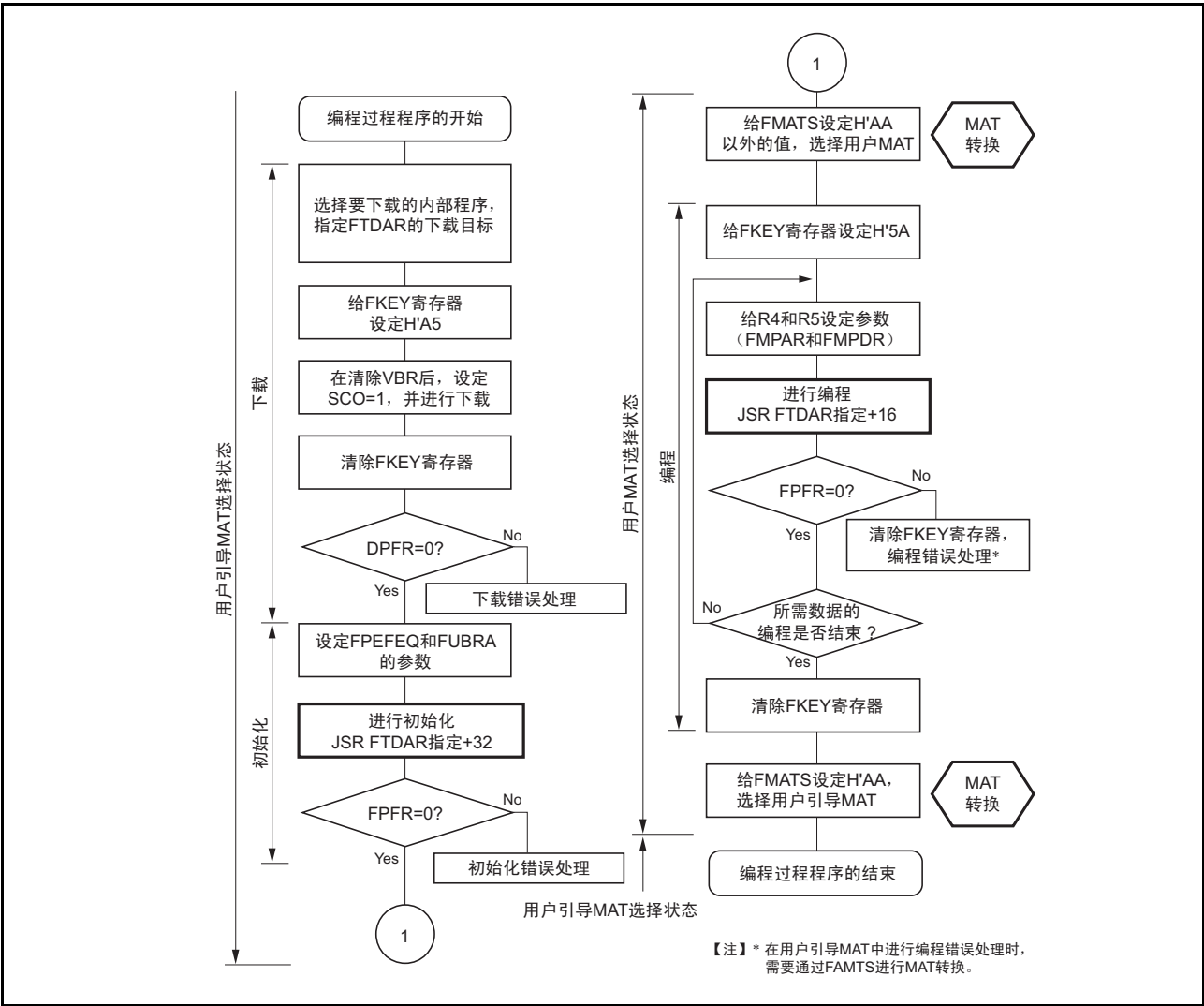


图 22.14 在用户引导模式中对用户 MAT 进行编程的步骤

如图 22.14 所示，用户编程模式和用户引导模式的编程步骤的不同在于是否进行 MAT 转换。

在用户引导模式中，能看到闪存空间的用户引导 MAT，而用户 MAT 隐藏在背后。只在用户 MAT 的编程处理期间才进行用户 MAT 和用户引导 MAT 的转换。在编程处理中，用户引导 MAT 处于隐藏状态，而用户 MAT 处于编程状态，因此需要在闪存以外的区域执行程序。当编程处理结束时，为了返回最初状态而再次进行 MAT 转换。

能通过给 FMATS 寄存器写规定的值，实现 MAT 的转换，但是在 MAT 转换完全结束前不能存取 MAT，如果在 MAT 转换中发生中断，就会出现无法确定从哪个 MAT 读中断向量的不稳定状态。有关 MAT 转换，请按照“22.8.1 用户 MAT 和用户引导 MAT 的转换”的说明进行。

除 MAT 转换以外，编程步骤与用户编程模式的步骤相同。

有关能单步执行用户过程程序的区域（内部 RAM、用户 MAT、外部空间等），请参照“22.9.2 过程程序和编程数据的可保存区域”。

(3) 用户引导模式中的用户 MAT 擦除

在用户引导模式中擦除用户 MAT 时，需要追加以下步骤：

- 通过 FMATS 寄存器从用户引导 MAT 选择状态转换到用户 MAT 选择状态
- 在擦除结束后，再次从用户 MAT 选择状态返回到用户引导 MAT 选择状态

在用户引导模式中对用户 MAT 进行擦除的步骤如图 22.15 所示。

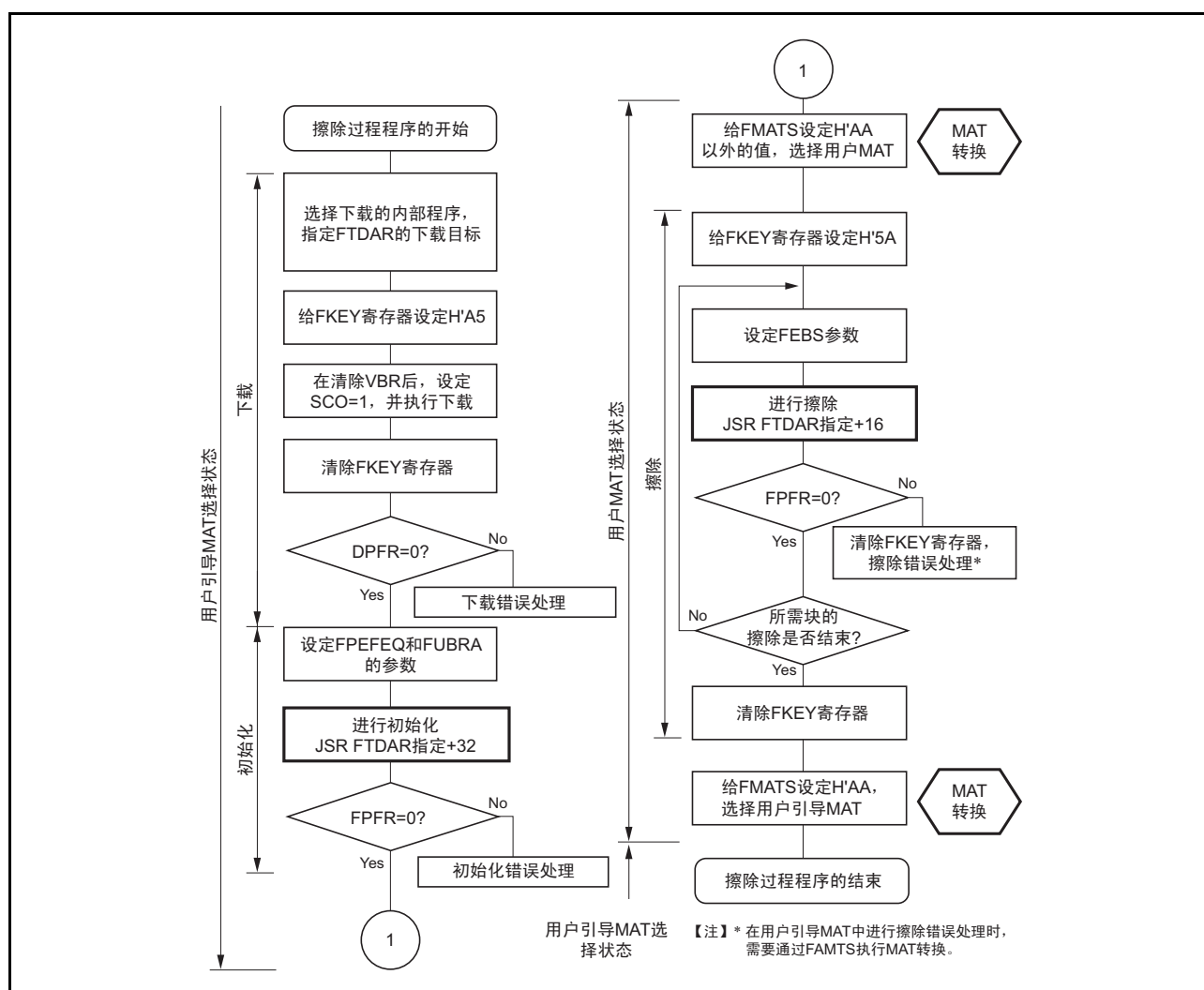


图 22.15 在用户引导模式中对用户 MAT 进行擦除的步骤

如图 22.15 所示，用户编程模式和用户引导模式的擦除步骤的不同在于是否进行 MAT 转换。

能通过给 FMATS 寄存器写规定的值，实现 MAT 的转换，但是在 MAT 转换完全结束前不能存取 MAT，如果在 MAT 转换中发生中断，就会出现无法确定从哪个 MAT 读中断向量的不稳定状态。有关 MAT 转换，请按照“22.8.1 用户 MAT 和用户引导 MAT 的转换”的说明进行。

除 MAT 转换已外，擦除步骤与用户编程模式的步骤相同。

有关能单步执行用户过程程序的区域（内部 RAM、用户 MAT、外部空间等），请参照“22.9.2 过程程序和编程数据的可保存区域”。

22.6 保护

闪存的编程 / 擦除保护有硬件保护、软件保护和错误保护 3 种。

22.6.1 硬件保护

硬件保护是指强制禁止或者强制中断闪存编程 / 擦除的状态。能下载内部程序并进行初始化，但是即使启动编程 / 擦除程序也不能对用户 MAT 进行编程 / 擦除，而通过 FPFRR 参数报告编程 / 擦除错误。

表 22.9 硬件保护

项目	说明	有效的保护功能	
		下载	编程和擦除
FWE 引脚的保护	当给 FWE 引脚输入 Low 电平时，就清除 FCCS 寄存器的 FWE 位，进入编程 / 擦除的保护状态。	—	○
复位 / 待机的保护	- 在上电复位（包括由 WDT 产生的上电复位）和待机时，初始化编程 / 擦除接口寄存器，进入编程 / 擦除保护状态。 - 在通过 RES 引脚进行复位时，如果不将 RES 引脚的 Low 电平保持到接通电源后的振荡稳定为止，就不进入复位状态。另外，运行中的复位必须在 AC 特性规定的 RES 脉宽之间将 $\overline{\text{RES}}$ 引脚保持为 Low 电平。不保证编程 / 擦除运行中的闪存值。此时，必须在擦除后重新进行编程。	○	○

22.6.2 软件保护

软件保护有内部编程 / 擦除程序下载保护、键代码保护和 RAM 仿真保护。

表 22.10 软件保护

项目	说明	有效的保护功能	
		下载	编程和擦除
SCO 位的保护	因将 FCCS 寄存器的 SCO 位清 0 而不能下载编程 / 擦除程序，所以进入编程 / 擦除保护状态。	○	○
FKEY 寄存器的保护	如果不给 FKEY 寄存器写键代码，就不能下载和编程 / 擦除。必须给下载和编程 / 擦除设定不同的键代码。	○	○
仿真保护	通过将 RAM 仿真寄存器（RAMER）的 RAMS 位置 1，进入编程 / 擦除保护状态。	○	○

22.6.3 错误保护

错误保护是在闪存的编程 / 擦除中因检测到单片机失控或者没有按规定的编程 / 擦除步骤运行而强制中断编程 / 擦除运行的保护。通过中断编程 / 擦除运行，能防止因重复编程或者重复擦除而损伤闪存。

如果在闪存的编程 / 擦除中单片机运行异常，就将 FCCS 寄存器的 FLER 位置 1，并转移到错误保护状态，中断编程 / 擦除。

FLER 位的置位条件如下所示：

1. 在编程/擦除中已读闪存的该存储体区时（包括读向量和取指令）
2. 在编程/擦除中已执行 SLEEP 指令时（包括软件待机）

只能在上电复位时解除错误保护（清除 FLER 位）。

另外，此时必须在长于通常的 100 μ s 的复位输入期间后解除复位。因为在编程 / 擦除过程中闪存加有高电压，在向错误保护状态转移时有可能无法完全去除外加电压。所以需要通过延长复位期间去除外加电压，减少对闪存的损伤。

错误保护状态的状态转移如图 22.16 所示。

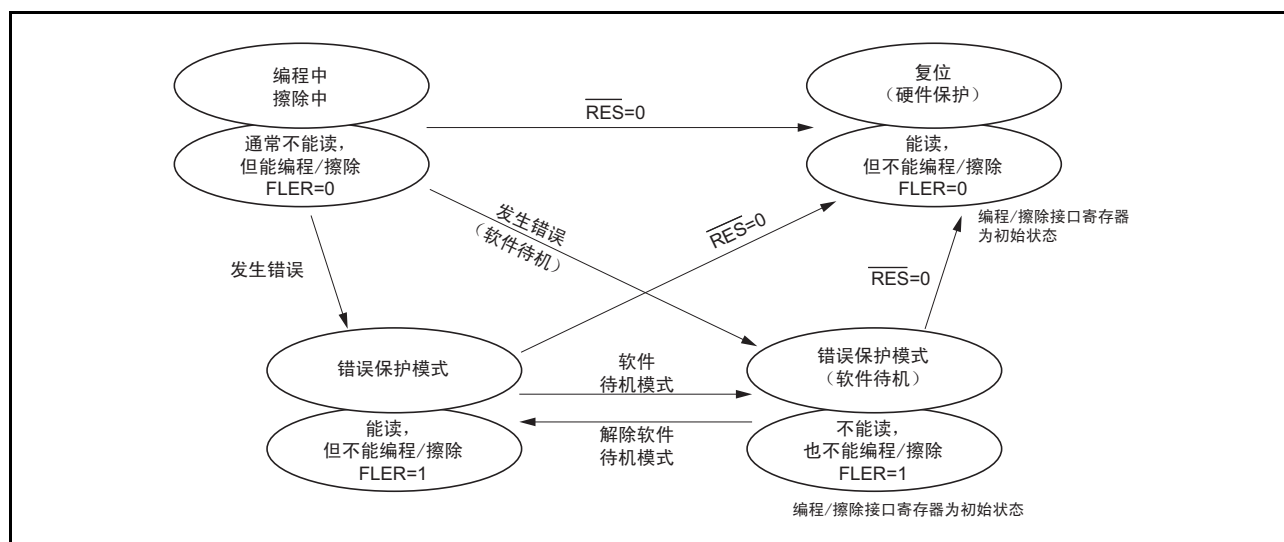


图 22.16 错误保护状态的状态转移图

22.7 通过 RAM 对闪存进行的仿真

为了通过 RAM 对闪存数据的改写进行实时仿真，能将一部分 RAM 和 RAM 仿真寄存器（RAMER）设定的闪存区（用户 MAT）重叠使用。在设定 RAMER 后，能从用户 MAT 区和与其重叠的 RAM 区进行存取。可仿真的模式为用户模式和用户编程模式。

对用户 MAT 的实时改写进行仿真的例子如图 22.17 所示。

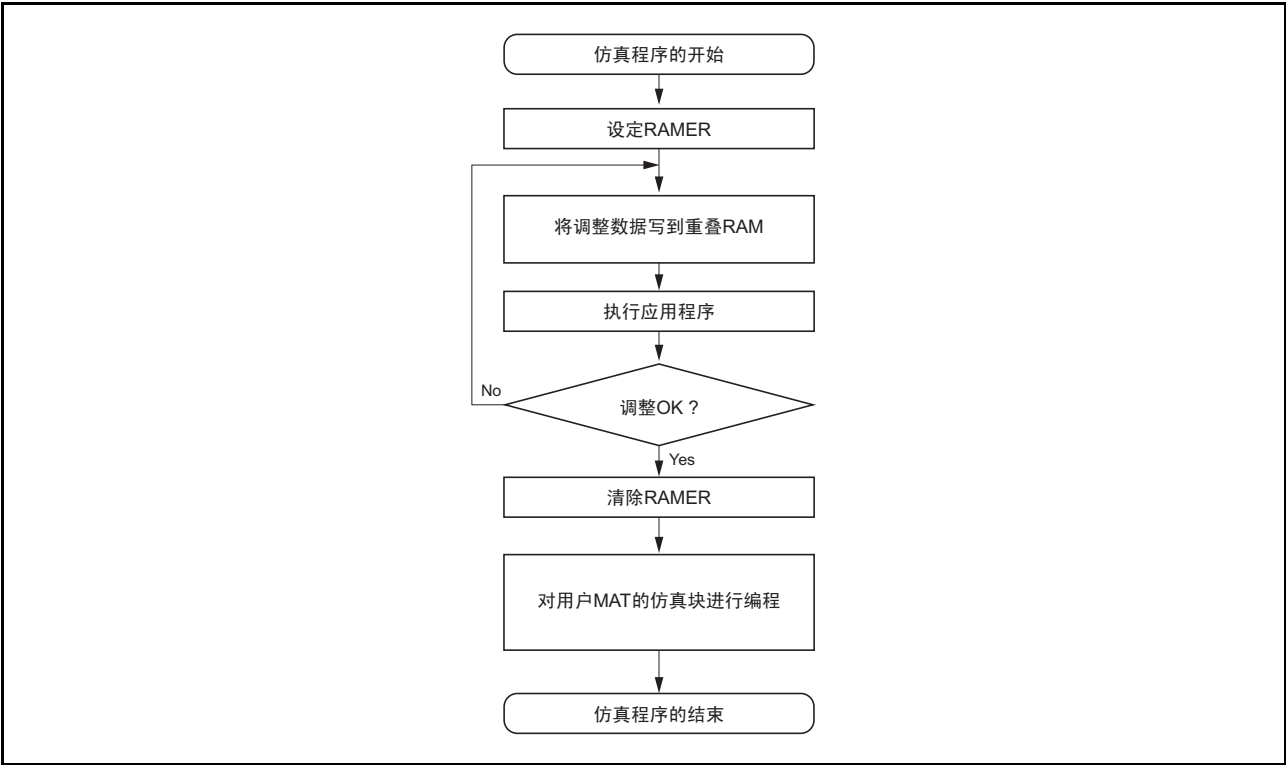


图 22.17 通过 RAM 进行的仿真

使闪存的块区 EB0 重叠的例子如图 22.18 所示。

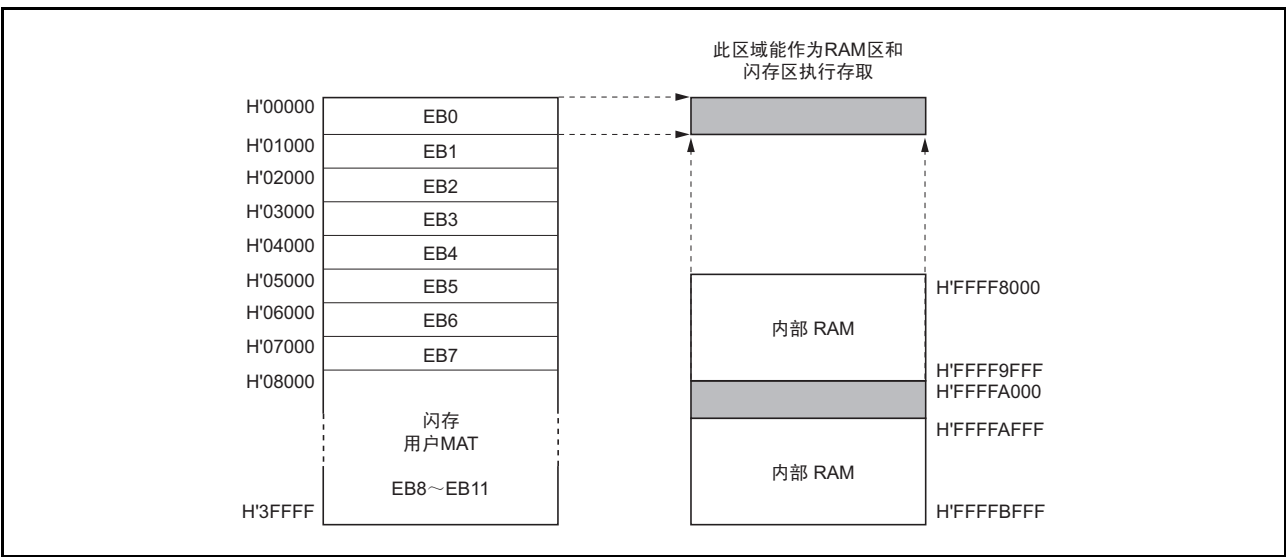


图 22.18 RAM 的重叠运行例子

可仿真的闪存区为用户 MAT 的 EB0 ~ EB7 的 8 个区域中的 1 个区域（通过 RAMER 寄存器的 RAM2 ~ RAM0 位选择）。

1. 为了使一部分 RAM 与实时改写所需的区域 EB0 重叠，必须将 RAMER 的 RAMS 位置 1，将 RAM2 ~ RAM0 位置 0。
2. 使用重叠的 RAM 进行实时改写。

在对用户 MAT 进行编程 / 擦除时，需要进行包含内部程序下载的一系列过程程序。此时，为了使 RAM 的重叠区域和下载的内部程序区不重叠，必须使用 FTDAR 寄存器设定下载区。

仿真结束后对用户 MAT 的 EB0 区进行数据编程的例子如图 22.19 所示。

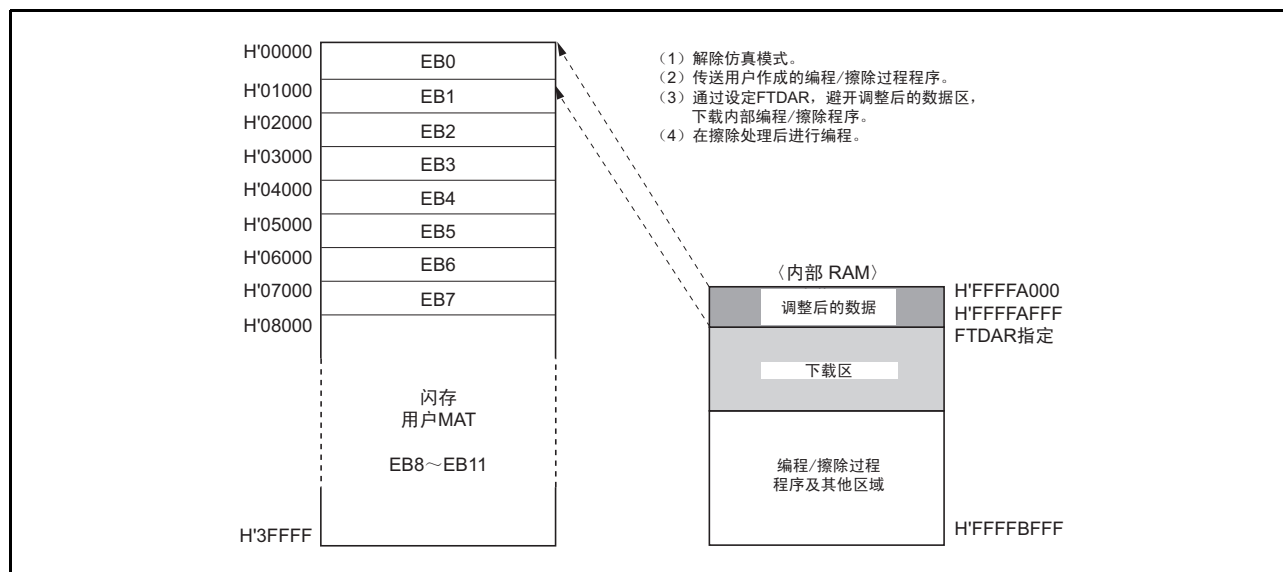


图 22.19 调整结束后的数据编程例子

1. 在确定要改写的的数据后，清除 RAMS 位，解除 RAM 的重叠。如果解除仿真模式，仿真保护也被解除。
2. 将用户建立的编程 / 擦除过程程序传送到 RAM。
3. 启动 RAM 的编程 / 擦除过程程序，将单片机的内部编程 / 擦除程序下载到 RAM。
此时，为了不使调整结束后的数据区和下载区重叠，必须通过设定 FTDAR 寄存器，指定下载起始地址。
4. 在尚未擦除用户 MAT 的 EB0 区时，必须在擦除后进行编程。必须在给编程处理参数 FMPAR 和 FMPDR 指定调整后的数据，然后进行编程处理。

【注】 如果将 RAMS 位置 1，就与 RAM2 ~ RAM0 的值无关，闪存的全部块处于编程 / 擦除保护状态（仿真保护）。在实际编程和擦除时，必须清除 RAMS 位。

即使选择了用户引导 MAT，也能进行 RAM 仿真。但是，只能在引导模式或者编程器模式中对用户引导 MAT 进行擦除 / 编程。

22.8 使用时的注意事项

22.8.1 用户 MAT 和用户引导 MAT 的转换

用户 MAT 和用户引导 MAT 能相互转换。但是，因为它们被分配到相同的地址 0，所以需要以下的步骤：
(在转换到用户引导 MAT 的状态下，不能进行编程 / 擦除。必须在引导模式或者编程器模式中改写用户引导 MAT。)

1. 必须在内部 RAM 中通过 FMATS 寄存器进行 MAT 转换。

SH 单片机预取要执行的指令，所以如果在用户 MAT 中的程序执行期间进行 MAT 转换，就会出现是预取用户 MAT 中的指令码还是预取转换后的用户引导 MAT 中的指令的不稳定运行。

2. 为了保证在转换后存取 MAT，必须在改写内部 RAM 的 FMATS 寄存器后立即执行同一内部 RAM 中的 4 条 NOP 指令（这是为了不在转换中存取闪存）。
3. 如果在转换中发生中断，就不能保证存取哪个存储器 MAT。

在 MAT 转换前，必须屏蔽可屏蔽的中断，并且在 MAT 转换中，不能让系统发生 NMI 中断。

4. 必须注意：在 MAT 转换结束后，各种中断的向量表区也被转换。

在 MAT 转换前后进行同样的中断处理时，或者在无法禁止中断发生时，必须将中断处理程序传送到内部 RAM，并通过设定 VBR 寄存器，将中断向量表设定到内部 RAM。此时，必须注意 VBR 寄存器的更改和中断发生的竞争。

5. 用户 MAT 和用户引导 MAT 的存储容量不同。不能存取超过 12KB 空间的用户引导 MAT，否则会读取不定值。

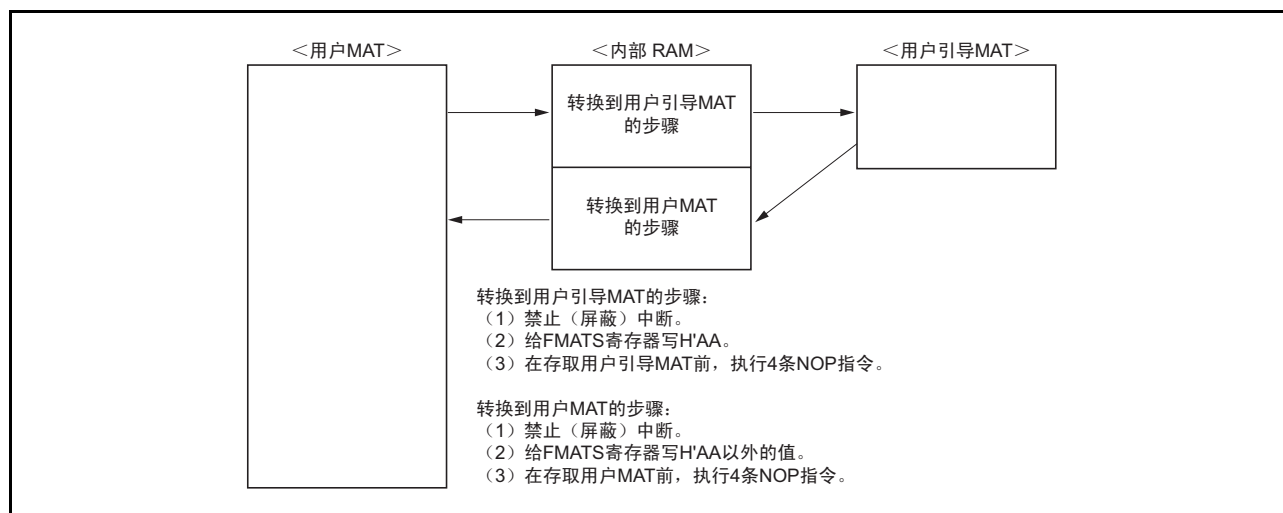


图 22.20 用户 MAT 和用户引导 MAT 的转换

22.8.2 编程 / 擦除过程中的中断

(1) 内部程序的下载

(1.1) VBR 的变更

在下载内部程序前，需要将 VBR 寄存器置 H'84000000。在设定 H'84000000 以外的值而使用 VBR 的情况下，如果将 VBR 置 H'84000000，中断向量表就变为用户 MAT（FMATS≠H'AA 时）或者用户引导 MAT（FMATS=H'AA 时）。

另外，如果 VBR 的变更和中断的发生出现竞争，就有可能因是参照 VBR 变更前还是参照变更后的向量表而发生问题。

因此，在有可能发生和中断竞争的情况下，还必须在用户 MAT 或者用户引导 MAT 的起始部分准备 VBR=H'00000000（初始值）时的参照向量表。

(1.2) SCO 的下载请求和中断请求

在将 FCCS 寄存器的 SCO 位置 1 后下载内部的编程 / 擦除程序时，会产生伴有 MAT 转换的特殊中断。以下说明 SCO 的下载请求和中断请求出现竞争时的运行：

1. SCO 下载请求和中断请求的竞争

执行将 FCCS 寄存器的 SCO 位置 1 的指令和接受中断的竞争时序如图 22.21 所示。

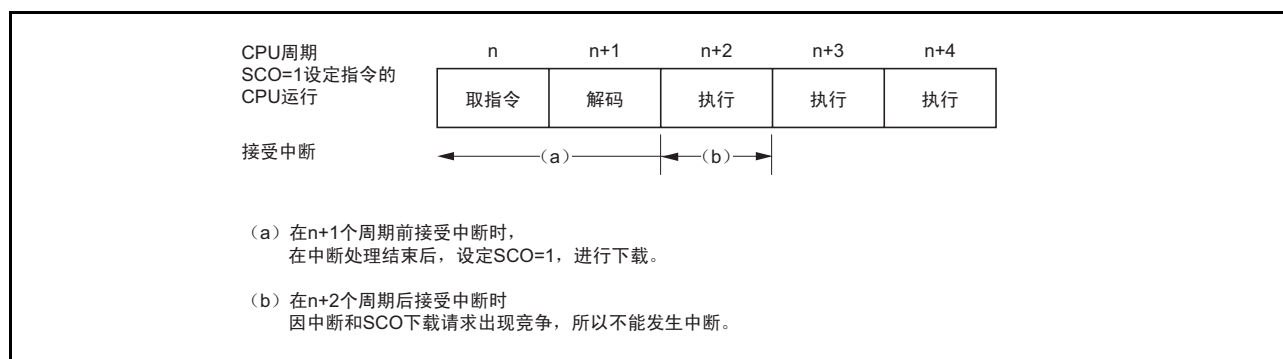


图 22.21 SCO 下载请求和中断请求的竞争时序

2. 在下载时发生中断请求

不能在 SCO 下载时发生中断。

(2) 编程 / 擦除处理中的中断

能实时处理在下载内部程序的编程 / 擦除中发生的中断，但是有以下的限制和注意事项：

- 对于正在编程 / 擦除的闪存，不能存取用户 MAT / 用户引导 MAT。必须将中断向量表和中断处理程序存放在内部 RAM 或者外部存储器中。即使在中断处理程序中也不能存取正在编程 / 擦除的闪存。如果读闪存，就不能保证读取值。如果在编程 / 擦除中存取闪存的该存储体，就转移到错误保护状态，中断编程 / 擦除处理。当存取该存储体以外的区域时，不转移到错误保护状态，但是不能保证读取的值。
- 不能更改 FMPDR 参数指定的编程数据。在通过中断处理准备编程数据时，必须先将数据存放到其他区域，然后在确认编程已结束后转移到 FMPDR 区或者将 FMPDR 更改为已准备好的其他区域。
- 不能在中断处理程序中破坏本章的闪存关联寄存器和已下载的内部程序区。另外，在中断处理中，不能进行 RAM 仿真，也不能多重进行 SCO 请求的内部程序下载或者编程 / 擦除。
- 必须在中断处理程序的起始位置保存 CPU 的寄存器，并在返回前恢复这些寄存器。
- 如果在中断处理程序中转移到睡眠状态或者软件待机状态，就进入错误保护状态，中断编程 / 擦除。另外，当转移到复位状态时，为了减少对闪存的损伤，必须在 100μs 以上的复位状态后解除复位。

22.8.3 其他注意事项

(1) 内部程序的下载执行时间

包含初始化程序的编程程序或者包含初始化程序的擦除程序的代码长度都在 3KB 以内。因此，当 CPU 的时钟频率为 20MHz 时，各程序的下载时间最大约为 10ms。

(2) 用户分支处理的间隔

进行用户分支处理的间隔因编程 / 擦除和处理阶段而不同。在 CPU 的时钟频率为 80MHz 时，最大启动间隔如表 22.11 所示。

表 22.11 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

在 CPU 时钟以 80MHz 运行时，到最初用户分支处理为止的时间最大值如表 22.12 所示。

表 22.12 用户分支处理时间

	最大
编程处理	约 2ms
擦除处理	约 15ms

(3) 通过 DTC 对闪存关联寄存器进行的写操作

只要在内部 RAM 中的执行指令期间，就也能通过 DTC 对下载所需的 FCCS 寄存器的 SCO 位或者 MAT 转换的 FMATS 寄存器进行写操作。如果没注意对这些寄存器进行写操作，就可能会进行下载，并破坏 RAM 或者发生 MAT 转换，导致 CPU 失控等。

(4) 中断的忽视状态

在以下模式或者期间中，即使发生中断也被忽视，不执行也不保存中断源。

- 引导模式运行中
- 编程器模式运行中

(5) 和以前的 F-ZTAT SH 单片机的编程 / 擦除程序的兼容性

不支持由 SCO 传送请求执行的内部程序的下载方式，本 LSI 不能运行以前的 F-ZTAT SH 单片机使用的闪存编程 / 擦除程序。

必须在下载内部程序后进行本 LSI 的闪存编程 / 擦除。

(6) 使用 WDT 对失控的监视

和以前的 F-ZTAT SH 单片机不同，在通过下载的内部程序进行编程 / 擦除时，无法通过 WDT 对应失控。

根据需要，必须使用考虑到编程 / 擦除执行时间的 WDT 对策（使用用户分支程序或者定期的定时器中断等）。

22.9 附录

22.9.1 引导模式的标准串行通信接口规格

在引导模式中启动的引导程序使用主机和 LSI 的内部 SCI 进行接收和发送。主机和引导程序的串行通信接口规格如下所示：

- 状态

引导程序有 3 个状态

1. 位速率匹配状态

这是使主机和接收/发送的位速率匹配的状态。当启动引导模式时，引导程序随即被启动，进入位速率匹配状态，接收主机的命令，并进行位速率的匹配。在匹配结束时，转移到查询选择状态。

2. 查询选择状态

这是应答主机查询命令的状态。在此状态中选择器件、时钟模式和位速率。在选择结束时，通过编程/擦除状态转移命令转移到编程/擦除状态。在转移到编程/擦除状态前，引导程序将擦除程序传送到 RAM，并擦除用户 MAT 和用户引导 MAT。

3. 编程/擦除状态

这是进行编程/擦除的状态。根据主机的命令，将编程/擦除程序传送到 RAM，并进行编程/擦除。通过命令进行校验和以及空白检查。

引导程序的处理流程如图 22.22 所示。

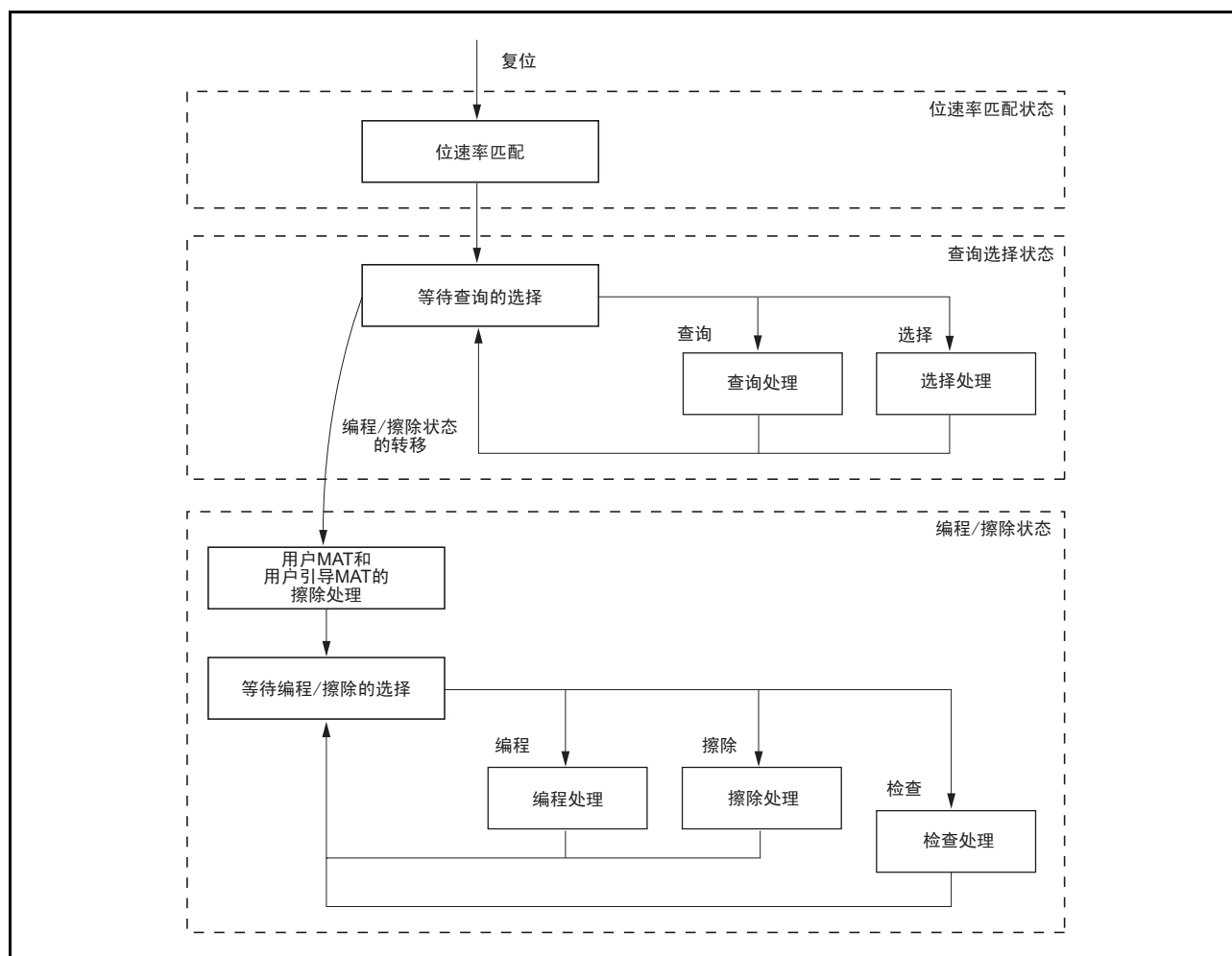


图 22.22 引导程序的处理流程

- 位速率匹配状态

在位速率匹配时，通过测量主机发送的 H'00 的低电平区间计算位速率。能通过新的位速率选择命令更改此位速率。当位速率匹配结束时，引导程序转移到查询选择状态。位速率匹配的顺序如图 22.23 所示。



图 22.23 位速率匹配的顺序

- 通信协议

在位速率匹配结束后，主机和引导程序的串行通信协议如下：

1. 1个字符命令或者1个字符应答
因为命令或者应答只有1个字符，所以有查询和正常结束的ACK。
2. n个字符命令或者n个字符应答
因为指令和应答需要n个字节的数据，所以有选择的命令和对应查询的应答。
因为对编程数据另外规定了数据长，所以省略数据长。
3. 错误应答
这是对命令的错误应答，错误应答和错误码为2个字节。
4. 128字节的编程
这是没有长度的命令，能通过编程长度查询的应答，得知数据的长度。
5. 存储器读操作的应答
这是长度为4字节的应答。

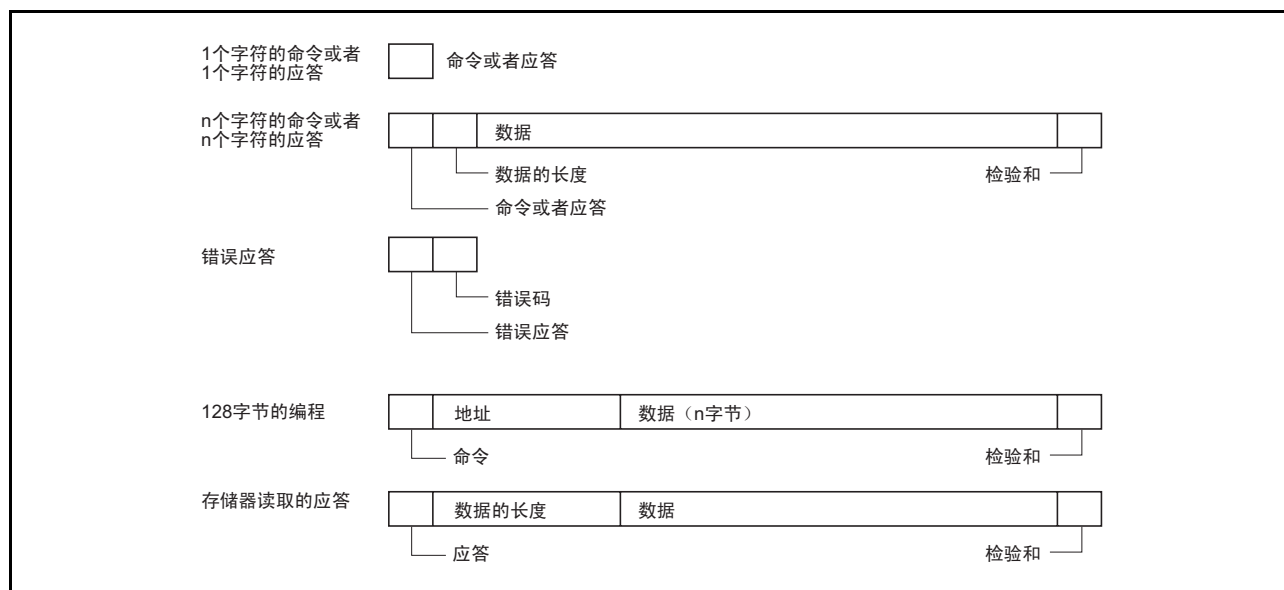


图 22.24 通信协议的格式

- 命令（1个字节）：查询、选择、编程、擦除、检查等命令
- 应答（1个字节）：对查询的应答
- 长度（1个字节或者2个字节）：命令/应答、长度、校验和除外的接收/发送数据的长度
- 数据（n个字节）：命令和应答的详细数据
- 校验和（1个字节）：命令到校验和（SUM）的总和，低位1个字节为H'00。
- 错误应答（1个字节）：对指令的错误应答
- 错误码（1个字节）：发生的错误种类
- 地址（4个字节）：编程地址
- 数据（n个字节）：编程数据，能从编程长度查询命令的应答得知n。
- 数据的长度（4个字节）：存储器读操作的4字节应答

查询选择状态

在查询选择状态下，引导程序对主机的查询命令进行闪存ROM信息的应答，对选择命令进行器件、时钟模式和位速率的选择。

查询选择命令一览表如表22.13所示。

表 22.13 查询选择命令一览表

命令	命令名	功能
H'20	支持器件的查询	查询器件的代码和产品名。
H'10	器件的选择	选择器件代码。
H'21	时钟模式的查询	查询时钟模式数和各自的值。
H'11	时钟模式的选择	通知已选择的时钟模式。
H'22	倍频比的查询	查询倍频比或者分频比的种类数和各自的个数及其值。
H'23	工作频率的查询	查询主时钟和外围时钟的最小值和最大值。
H'24	用户引导 MAT 信息的查询	查询用户引导 MAT 的个数以及各自的起始地址和最终地址。
H'25	用户 MAT 信息的查询	查询用户 MAT 的个数以及各自的起始地址和最终地址。
H'26	擦除块信息的查询	查询块数以及各自的起始地址和最终地址。
H'27	编程长度的查询	查询编程时的数据长。
H'3F	新位速率的选择	选择新的位速率。
H'40	编程 / 擦除状态的转移	擦除用户 MAT 和用户引导 MAT，转移到编程 / 擦除状态。
H'4F	引导程序状态的查询	查询引导的处理状态。

必须按器件的选择（H'10）、时钟模式的选择（H'11）和新位速率的选择（H'3F）的顺序，从主机发送选择命令。这些命令是必需的。当同一选择指令发送 2 次（含 2 次）以上时，后发送的选择命令有效。

除引导程序状态查询（H'4F）以外，这些命令在接受编程 / 擦除状态转移（H'40）之前都有效，主机能从上述的命令中选择需要的命令进行查询。引导程序状态查询（H'4F）在接受编程 / 擦除状态转移（H'40）后仍有效。

(1) 支持器件的查询

对于支持器件的查询，引导程序应答能支持的器件代码和产品名。

命令	H'20		
• 命令[H'20]（1个字节）：	查询支持器件		
应答	H'30	长度	器件数
	字符数	器件代码	产品名
	...		
	SUM		

- 应答[H'30]（1个字节）：对查询支持器件的应答
- 长度（1个字节）：命令、长度、校验和除外的接收/发送数据的长度。在此，为器件数、字符数、器件代码和产品名的合计长度。
- 器件数（1个字节）：单片机内的引导程序所支持的品种数
- 字符数（1个字节）：器件代码和引导程序产品名的字符数
- 器件代码（4个字节）：所支持产品名的 ASCII 代码
- 产品名（n个字节）：引导程序产品型号（ASCII 代码）
- SUM（1个字节）：校验和
命令到SUM的总和，使其结果为H'00。

(2) 器件的选择

对于器件的选择，引导程序将支持器件设定为指定的支持器件，对此后的查询，应答已选择的器件信息。

命令	H'10	长度	器件代码	SUM
----	------	----	------	-----

- 命令[H'10]（1个字节）：选择器件
- 长度（1个字节）：器件代码的字符数（是固定值，为2）
- 器件代码（4个字节）：查询支持器件时应答的器件代码（ASCII代码）
- SUM（1个字节）：校验和

应答	H'06
----	------

- 应答[H'06]（1个字节）：对选择器件的应答
当指定器件代码和支持器件代码相同时，返回ACK。

错误

应答	H'90	ERROR
----	------	-------

- 错误应答[H'90]（1个字节）：对选择器件的错误应答
- ERROR（1个字节）：错误码
H'11：校验和错误
H'21：器件代码不相同错误

(3) 时钟模式的查询

对于时钟模式的查询，引导程序应答能选择的时钟模式。

命令	H'21
----	------

- 命令[H'21]（1个字节）：查询时钟模式

应答	H'31	长度	模式	...	SUM
----	------	----	----	-----	-----

- 应答[H'31]（1个字节）：对查询时钟模式的应答
- 长度（1个字节）：模式数和模式的合计长度
- 模式（1个字节）：能选择的时钟模式（例：H'01时钟模式1）
- SUM（1个字节）：校验和

(4) 时钟模式的选择

对于时钟模式的选择，引导程序将时钟模式设定为指定的时钟模式，对此后的查询，应答已选择的时钟模式的信息。

必须在发送器件的选择命令后，发送时钟模式的选择命令。

命令	H'11	长度	模式	SUM
----	------	----	----	-----

- 命令[H'11]（1个字节）：选择时钟模式
- 长度（1个字节）：模式的字符数（是固定值，为1）
- 模式（1个字节）：查询时钟模式时应答的时钟模式
- SUM（1个字节）：校验和

应答	H'06
----	------

- 应答[H'06]（1个字节）：对选择时钟模式的应答
当指定的时钟模式和能选择的时钟模式相同时，返回ACK。

错误

应答	H'91	ERROR
----	------	-------

- 错误应答[H'91]（1个字节）：对选择时钟模式的错误应答

- ERROR（1个字节）：错误码
 - H'11：校验和错误
 - H'22：时钟模式不相同错误

(5) 倍频比的查询

对于倍频比的查询，引导程序应答能选择的倍频比或者分频比。

命令

H'22

- 命令[H'22]（1个字节）：查询倍频比

应答	H'32	长度	频率的种类数					
	倍频比数	倍频比	...					
	...							
	SUM							

- 应答[H'32]（1个字节）：对查询倍频比的应答
- 长度（1个字节）：频率的种类数、倍频比数和倍频比的合计长度
- 频率的种类数（1个字节）：器件能选择的倍频比的种类数（如果是主工作频率和外围模块工作频率2种，为H'02）
- 倍频比数（1个字节）：各工作频率能选择的倍频比数
主模块和外围模块能选择的倍频比数。
- 倍频比（1个字节）
倍频比：倍频的数值（例4倍频：H'04）
分频比：分频的数值，为负数（例2分频：H'FE[-2]）
倍频比的重复次数为倍频比数，倍频比数和倍频比的组合重复次数为频率的种类数。
- SUM（1个字节）：校验和

(6) 工作频率的查询

对于工作频率的查询，引导程序应答工作频率的种类数及其最小值和最大值。

命令

H'23

- 命令[H'23]（1个字节）：查询工作频率

应答	H'33	长度	频率种类数
	工作频率最小值		工作频率最大值
	...		
	SUM		

- 应答[H'33]（1个字节）：对查询工作频率的应答
- 长度（1个字节）：工作频率的种类数、工作频率最小值和工作频率最大值的合计长度
- 频率种类数（1个字节）：器件所需要的工作频率的种类数
例如，在主工作频率和外围模块工作频率时为2。
- 工作频率最小值（2个字节）：倍频或者分频时钟的最小值
工作频率的最小值和最大值是将频率（MHz）小数点后2位的值放大100倍的值（例如，当频率为20.00MHz时，放大100倍后为2000，设定为H'07D0）。
- 工作频率最大值（2个字节）：倍频或者分频时钟的最大值
工作频率的最小值和最大值的数据重复次数为频率种类数。
- SUM（1个字节）：校验和

(7) 用户引导 MAT 信息的查询

对于用户引导 MAT 信息的查询，引导程序应答用户引导 MAT 的区域数和地址。

命令

H'24

- 命令[H'24]（1个字节）：查询用户引导 MAT 信息

应答	H'34	长度	区域数	
	区域起始地址			区域最终地址
	...			
	SUM			

- 应答[H'34]（1个字节）：对查询用户引导 MAT 信息的应答
- 长度（1个字节）：区域数、区域起始地址和区域最终地址的合计长度
- 区域数（1个字节）：用户引导 MAT 的区域数
当用户引导 MAT 的区域连续时为 H'01。
- 区域起始地址（4个字节）：区域的起始地址
- 区域最终地址（4个字节）：区域的最终地址
区域起始地址和区域最终地址的数据重复次数为区域数。
- SUM（1个字节）：校验和

(8) 用户 MAT 信息的查询

对于用户 MAT 信息的查询，引导程序应答用户 MAT 的区域数和地址。

命令

H'25

- 命令[H'25]（1个字节）：查询用户 MAT 信息

应答	H'35	长度	区域数	
	区域起始地址			区域最终地址
	...			
	SUM			

- 应答[H'35]（1个字节）：对查询用户 MAT 信息的应答
- 长度（1个字节）：区域数、区域起始地址和区域最终地址的合计长度
- 区域数（1个字节）：连续的用户 MAT 的区域数
当用户 MAT 的区域连续时为 H'01。
- 区域起始地址（4个字节）：区域的起始地址
- 区域最终地址（4个字节）：区域的最终地址
区域起始地址和区域最终地址的数据重复次数为区域数。
- SUM（1个字节）：校验和

(9) 擦除块信息的查询

对于擦除块信息的查询，引导程序应答用户 MAT 的擦除块的块数和地址。

命令

H'26

- 命令[H'26]（1个字节）：查询擦除块信息

应答	H'36	长度	块数	
	块起始地址			块最终地址
	...			
	SUM			

- 应答[H'36]（1个字节）：对查询擦除块信息的应答
- 长度（2个字节）：块数、块起始地址和块最终地址的合计长度
- 块数（1个字节）：闪存的擦除块数
- 块起始地址（4个字节）：块的起始地址
- 块最终地址（4个字节）：块的最终地址
块起始地址和块最终地址的数据重复次数为块数。
- SUM（1个字节）：校验和

(10) 编程长度的查询

对于编程长度的查询，引导程序应答编程数据的编程单位。

命令

H'27

- 命令[H'27]（1个字节）：查询编程长度

应答

H'37	长度	编程长度	SUM
------	----	------	-----

- 应答[H'37]（1个字节）：对查询编程长度的应答
- 长度（1个字节）：编程单位长度的字符数（是固定值，为2）
- 编程长度（2个字节）：编程单位的长度
以此长度接受编程数据。
- SUM（1个字节）：校验和

(11) 新位速率的选择

对于新位速率的选择，引导程序将位速率更改为指定的位速率，并以新位速率应答主机的确认。
必须在发送时钟模式的选择命令后，发送新位速率的选择命令。

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个字节）：选择新的位速率
- 长度（1个字节）：位速率、输入频率、倍频比数和倍频比的合计长度
- 位速率（2个字节）：新的位速率
为1/100的值（例如，当位速率是19200bps时为192，设定为H'00C0）。
- 输入频率（2个字节）：输入到引导程序的时钟频率
为频率（MHz）小数点后2位的值（例如，当时钟频率为28.882MHz时，将小数点后2位的值放大100倍后为2888，设定为H'0B48）。
- 倍频比数（1个字节）：器件能选择的倍频比数
通常是主工作频率和外围模块工作频率，为2。
- 倍频比1（1个字节）：主工作频率的倍频比或者分频比
倍频比：倍频的数值（例 4倍频：H'04）
分频比：分频的数值，为负数（例 2分频：H'FE[-2]）
- 倍频比2（1个字节）：外围工作频率的倍频比或者分频比
倍频比：倍频的数值（例 4倍频：H'04）
分频比：分频的数值，为负数（例 2分频：H'FE[-2]）
- SUM（1个字节）：校验和

应答

H'06

- 应答[H'06]（1个字节）：对选择新位速率的应答

在选择指定的位速率时，返回ACK。

错误

应答

H'BF	ERROR
------	-------

- 错误应答[H'BF]（1个字节）：对选择新位速率的错误应答
- ERROR（1个字节）：错误码
 - H'11：校验和错误
 - H'24：不能选择位速率错误
 - 不能选择指定的位速率
 - H'25：输入频率错误
 - 输入频率不在最小值和最大值的范围内
 - H'26：倍频比错误
 - 倍频比不一致
 - H'27：工作频率错误
 - 工作频率不在最小值和最大值的范围内

接收数据的检查方法如下所示：

1. 输入频率

检查接收的输入频率值是否在输入频率的最小值和最大值的范围内（对于已选择器件的时钟模式）。如果不在范围内，就为输入频率错误。

2. 倍频比

检查已接收的倍频比或者分频比的值是否和倍频比或者分频比（对于已选择器件的时钟模式）相同。如果不相同，就为倍频比错误。

3. 工作频率

通过接收的输入频率以及倍频比或者分频比计算工作频率。输入频率是提供给LSI的频率，工作频率是LSI的实际工作频率，计算式如下所示：

工作频率=输入频率×倍频比

或者工作频率=输入频率÷分频比

检查此计算出的工作频率是否在工作频率的最小值和最大值的范围内（对于已选择器件的时钟模式）。如果不在范围内，就为工作频率错误。

4. 位速率

通过外围工作频率（ $P\phi$ ）和位速率（B）计算串行模式寄存器（SCSMR）的时钟选择（CKS）的值（n）、位速率寄存器（SCBRR）的值（N）和误差，检查误差是否小于4%。如果误差不小于4%，就为位速率选择错误。误差的计算如下：

$$\text{误差}(\%) = \left\{ \frac{P\phi \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

如果能选择新的位速率，就在应答ACK后给寄存器选择新位速率的值。主机以新的位速率发送ACK，引导程序以新的位速率应答。

确认

H'06

- 确认[H'06]（1个字节）：确认新的位速率

应答

H'06

- 应答[H'06]（1个字节）：对确认新位速率的应答
新位速率的选择顺序如图22.25所示。



图 22.25 新位速率的选择顺序

(12) 编程 / 擦除状态转移

对于编程 / 擦除状态转移，引导程序传送擦除程序，并按用户 MAT、用户引导 MAT 的顺序擦除数据。当擦除结束时，应答 ACK，进入编程 / 擦除状态。

在发送编程选择命令和编程数据前，主机必须通过器件的选择命令、时钟模式的选择命令和新位速率的选择命令，选择 LSI 的器件、时钟模式和新位速率，并将编程 / 擦除状态转移命令发送到引导程序。

命令 H'40

- 命令[H'40] (1个字节)：编程/擦除状态转移

应答 H'06

- 应答[H'06] (1个字节)：对编程/擦除状态转移的应答。在传送擦除程序后，当用户引导 MAT 和用户 MAT 被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个字节)：对编程/擦除状态转移的错误应答
- 错误码[H'51] (1个字节)：擦除错误
因发生错误而不能擦除。

命令错误

如果未定义命令、命令顺序不正确或者不能接受命令，就为命令错误。例如，器件选择前的时钟模式选择命令或者编程/擦除状态转移命令后的查询命令为命令错误。

错误

应答 H'80 H'xx

- 错误应答[H'80] (1个字节)：命令错误
- 命令[H'xx] (1个字节)：已接收的命令

命令顺序

在查询选择状态下，命令顺序的例子如下：

1. 必须通过支持器件的查询 (H'20)，查询支持的器件。

2. 必须从被应答的器件信息中选择器件，进行器件的选择（H'10）。
3. 必须通过时钟模式的查询（H'21），进行时钟模式的查询。
4. 必须从被应答的时钟模式中选择时钟模式，进行时钟模式的选择（H'11）。
5. 当器件选择和时钟模式选择结束时，必须通过倍频比的查询（H'22）和工作频率的查询（H'23），查询新位速率选择所需要的信息。
6. 必须按照倍频比和工作频率的信息，选择新位速率（H'3F）。
7. 当器件选择和时钟模式选择结束时，必须通过用户引导MAT信息的查询（H'24）、用户MAT信息的查询（H'25）、擦除块信息的查询（H'26）和编程长度的查询（H'27），查询用户引导MAT和用户MAT的编程/擦除信息。
8. 当查询和新位速率选择结束时，必须进行编程/擦除状态转移（H'40），转移到编程/擦除状态。

• 编程/擦除状态

在编程/擦除状态下，引导程序通过编程选择命令选择编程方法，通过128字节编程命令进行数据的编程，并通过擦除选择命令和块擦除命令进行块的擦除。编程/擦除命令一览表如表22.14所示。

表 22.14 编程 / 擦除命令一览表

命令	命令名	功能
H'42	用户引导 MAT 编程的选择	选择用户引导 MAT 的编程程序。
H'43	用户 MAT 编程的选择	选择用户 MAT 的编程程序。
H'50	128 字节编程	128 字节编程
H'48	擦除的选择	选择擦除程序。
H'58	块擦除	擦除块数据。
H'52	存储器的读操作	读存储器。
H'4A	用户引导 MAT 的校验和	用户引导 MAT 的校验和
H'4B	用户 MAT 的校验和	用户 MAT 的校验和
H'4C	用户引导 MAT 的空白检查	用户引导 MAT 的空白检查
H'4D	用户 MAT 的空白检查	用户 MAT 的空白检查
H'4F	引导程序状态的查询	查询引导的处理状态。

• 编程

使用编程选择命令和128字节编程命令进行编程。

首先，主机发送编程选择命令，选择编程方式和编程MAT。根据编程区和编程方式有以下2个编程选择命令：

- a. 用户引导MAT编程的选择
- b. 用户MAT编程的选择

其次，发送128字节编程命令。接在选择命令后面的128字节编程命令分别被解释为选择命令所指定的编程方式的编程数据。当编程超过128字节的数据时，必须重复128字节的编程命令。如果要结束编程，必须由主机发送地址为H'FFFFFFF的128字节编程指令。当编程结束时，进入编程/擦除选择命令的等待状态。

当继续进行其他方式和其他MAT的编程时，从编程选择命令开始。

编程选择命令和128字节编程命令的顺序如图22.26所示。

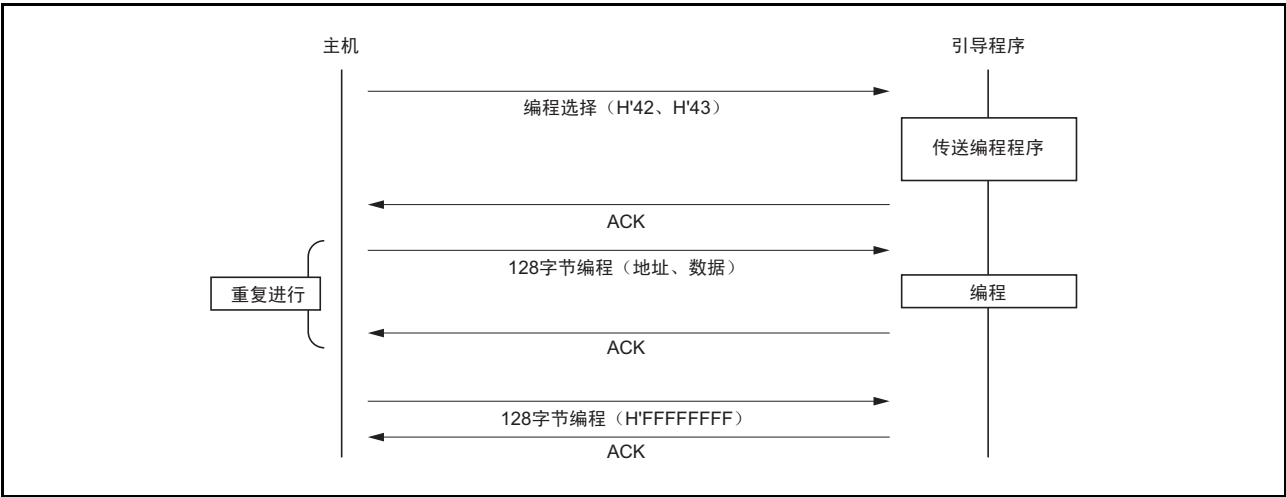


图 22.26 编程顺序

(1) 用户引导 MAT 编程的选择

对于用户引导 MAT 编程的选择，引导程序传送编程程序，并通过传送的编程程序对用户引导 MAT 进行编程。

命令

H'42

- 命令[H'42]（1个字节）：选择用户引导 MAT 的编程

应答

H'06

- 应答[H'06]（1个字节）：对选择用户引导 MAT 编程的应答。当编程程序传送结束时，返回 ACK。

错误

应答

H'C2	ERROR
------	-------

- 错误应答[H'C2]（1个字节）：对选择用户引导 MAT 编程的错误应答
- ERROR（1个字节）：错误码

H'54：选择处理错误（发生传送错误，处理未结束）

(2) 用户 MAT 编程的选择

对于用户 MAT 编程的选择，引导程序传送编程程序，并通过传送的编程程序对用户 MAT 进行编程。

命令

H'43

- 命令[H'43]（1个字节）：选择用户 MAT 的编程

应答

H'06

- 应答[H'06]（1个字节）：对选择用户 MAT 编程的应答。当编程程序传送结束时，返回 ACK。

错误

应答

H'C3	ERROR
------	-------

- 错误应答[H'C3]（1个字节）：对选择用户 MAT 编程的错误应答
- ERROR（1个字节）：错误码

H'54：选择处理错误（发生传送错误，处理未结束）

(3) 128 字节编程

对于 128 字节编程，引导程序通过编程选择传送的编程程序，对用户引导 MAT 或者用户 MAT 进行编程。

命令	H'50	编程地址						
	数据	...						
	...							
	SUM							

- 命令[H'50]（1个字节）：128 字节编程
- 编程地址（4个字节）：编程的起始地址
必须指定 128 字节边界的地址。
例）H'00、H'01、H'00、H'00：H'00010000。
- 编程数据（n个字节）：编程数据
编程数据的长度是“编程长度的查询”应答的长度。
- SUM（1个字节）：校验和

应答	H'06
----	------

- 应答[H'06]（1个字节）：对 128 字节编程的应答
当编程结束时，返回 ACK。

错误

应答	H'D0	ERROR
----	------	-------

- 错误应答[H'D0]（1个字节）：对 128 字节编程的错误应答
- ERROR（1个字节）：错误码
H'11：校验和错误
H'2A：地址错误（地址不在指定 MAT 的范围内）
H'53：编程错误（发生编程错误，不能编程）

必须按照数据编程长度指定边界地址。例如，当数据编程长度为 128 字节时，必须将地址的低位字节设定为 H'00 或者 H'80。

主机必须将 128 字节中没有编程数据的部分填写 H'FF，然后发送。

当编程处理结束时，必须发送地址 H'FFFFFFFF 的 128 字节编程指令。对地址 H'FFFFFFFF 的 128 字节编程指令，引导程序判断数据已结束，进入编程 / 擦除选择指令的等待状态。

命令	H'50	编程地址	SUM
----	------	------	-----

- 命令[H'50]（1个字节）：128 字节编程
- 编程地址（4个字节）：结束码（H'FF、H'FF、H'FF、H'FF）
- SUM（1个字节）：校验和

应答	H'06
----	------

- 应答[H'06]（1个字节）：对 128 字节编程的应答
当编程处理结束时，返回 ACK。

错误

应答	H'D0	ERROR
----	------	-------

- 错误应答[H'D0]（1个字节）：对 128 字节编程的错误应答
ERROR（1个字节）：错误码
H'11：校验和错误
H'53：编程错误

- **擦除**
使用擦除选择命令和块擦除命令，进行擦除。
首先通过擦除选择命令，选择擦除，然后对块擦除命令所指定的块进行擦除。当有多块擦除块时，重复块擦除命令。当擦除处理结束时，必须从主机发送块号为H'FF的块擦除命令。当擦除结束时，进入编程/擦除选择的等待状态。
擦除选择命令和块擦除命令的顺序如图22.27所示。

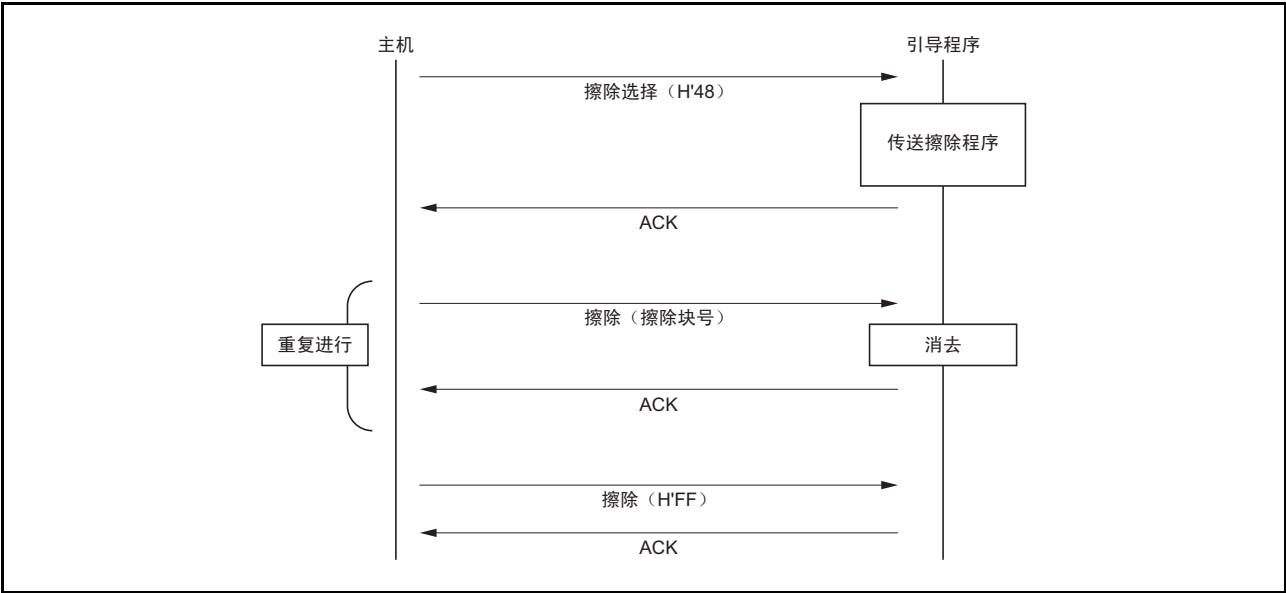


图 22.27 擦除顺序

(1) 擦除的选择

对于擦除的选择，引导程序传送擦除程序，并通过传送的擦除程序擦除用户 MAT 的数据。

命令

H'48

- 命令[H'48] (1个字节)：选择擦除

应答

H'06

- 应答[H'06] (1个字节)：对选择擦除的应答
当擦除程序传送结束时，返回ACK。

错误

应答

H'C8	ERROR
------	-------

- 错误应答[H'C8] (1个字节)：对选择擦除的错误应答
- ERROR (1字节)：错误码
H'54：选择处理错误 (发生传送错误，处理未结束)

(2) 块擦除

对于擦除，引导程序擦除所指定的用户 MAT 块。

命令

H'58	长度	块号	SUM
------	----	----	-----

- 指令[H'58] (1个字节)：擦除
- 长度 (1个字节)：擦除块号的字符数 (是固定值，为1)
- 块号 (1个字节)：擦除数据的擦除块号

- SUM（1个字节）：校验和

应答

H'06

- 应答[H'06]（1个字节）：对擦除的应答
当擦除结束时，返回ACK。

错误

应答

H'D8	ERROR
------	-------

- 错误应答[H'D8]（1个字节）：对擦除的错误应答
- ERROR（1个字节）：错误码
 - H'11：校验和错误
 - H'29：块号错误
块号不正确
 - H'51：擦除错误
在擦除时发生错误

对于块号 H'FF，引导程序结束擦除处理，进入选择命令的等待状态。

命令

H'58	长度	块号	SUM
------	----	----	-----

- 命令[H'58]（1个字节）：擦除
- 长度（1个字节）：擦除块号的字符数（是固定值，为1）
- 块号（1个字节）：H'FF（擦除处理的结束码）
- SUM（1个字节）：校验和

应答

H'06

- 应答[H'06]（1个字节）：对擦除结束的应答ACK
在将块号指定为 H'FF 后重新擦除时，从擦除选择开始进行。

存储器的读操作

对存储器的读操作，引导程序读取所指定的地址数据。

命令

H'52	长度	区域	读的起始地址
读的长度	SUM		

- 命令[H'52]（1个字节）：读存储器
- 长度（1个字节）：区域、读的起始地址和读的长度的合计长度（是固定值，为9）
- 区域（1个字节）：
 - H'00：用户引导MAT
 - H'01：用户MAT
 当指定的区域不正确时，为地址错误。
- 读的起始地址（4个字节）：读的起始地址
- 读的长度（4个字节）：读数据的长度
- SUM（1个字节）：校验和

应答

H'52	读取长度							
数据	...							
SUM								

- 应答[H'52]（1个字节）：对读存储器的应答
- 读取的长度（4个字节）：读数据的长度
- 数据（n个字节）：从读的起始地址开始，个数为读取长度的数据

- SUM（1个字节）：校验和

错误

应答	H'D2	ERROR
----	------	-------

- 错误应答[H'D2]（1个字节）：对读存储器的错误应答
- ERROR（1个字节）：错误码
 - H'11：校验和错误
 - H'2A：地址错误
 - 读的起始地址不在MAT的范围内。
 - H'2B：长度错误
 - 读的长度超过MAT的范围，或者从读的起始地址和读的长度计算的读的最终地址不在MAT的范围内，或者读的长度为0。

- 用户引导MAT的校验和

对于用户引导MAT的校验和，引导程序将用户引导MAT的数据相加，并应答其结果。

命令	H'4A
----	------

- 命令[H'4A]（1个字节）：用户引导MAT的校验和

应答	H'5A	长度	MAT的校验和	SUM
----	------	----	---------	-----

- 应答[H'5A]（1个字节）：对用户引导MAT的校验和的应答
- 长度（1个字节）：MAT的校验和的字符数（是固定值，为4）
- MAT的校验和（4个字节）：用户引导MAT的校验和值，以字节为单位相加
- SUM（1个字节）：校验和（发送数据）

- 用户MAT的校验和

对于用户MAT的校验和，引导程序将用户MAT的数据相加，并应答其结果。

命令	H'4B
----	------

- 命令[H'4B]（1个字节）：用户MAT的校验和

应答	H'5B	长度	MAT的校验和	SUM
----	------	----	---------	-----

- 应答[H'5B]（1个字节）：对用户MAT的校验和的应答
- 长度（1个字节）：MAT的校验和的字符数（是固定值，为4）
- MAT的校验和（4个字节）：用户MAT的校验和值，以字节为单位相加
- SUM（1个字节）：校验和（发送数据）

- 用户引导MAT的空白检查

对于用户引导MAT的空白检查，引导程序检查用户引导MAT是否全部为空白，并应答其结果。

命令	H'4C
----	------

- 命令[H'4C]（1个字节）：用户引导MAT的空白检查

应答	H'06
----	------

- 应答[H'06]（1个字节）：对用户引导MAT的空白检查的应答
 - 当区域全部为空白（H'FF）时，返回ACK。

错误

应答	H'CC	H'52
----	------	------

- 错误应答[H'CC]（1个字节）：对用户引导MAT的空白检查的错误应答
- 错误码[H'52]（1个字节）：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 是否全部为空白，并应答其结果。

命令

H'4D

- 命令[H'4D]（1个字节）：用户 MAT 的空白检查

应答

H'06

- 应答[H'06]（1个字节）：对用户 MAT 的空白检查的应答
当区域全部为空白（H'FF）时，返回ACK。

错误

应答

H'CD	H'52
------	------

- 错误应答[H'CD]（1个字节）：对用户 MAT 的空白检查的错误应答
- 错误码[H'52]（1个字节）：未擦除错误

- 引导程序状态的查询

对于引导程序状态的查询，引导程序应答当前的状态和错误状态。此查询在查询选择状态和编程/擦除状态时都有效。

命令

H'4F

- 命令[H'4F]（1个字节）：查询引导程序状态

应答

H'5F	长度	STATUS	ERROR	SUM
------	----	--------	-------	-----

- 应答[H'5F]（1个字节）：对查询引导程序状态的应答
- 长度（1个字节）：数据的字符数（是固定值，为2）
- STATUS（1个字节）：标准引导程序的状态
 请参照表 22.15。
- ERROR（1个字节）：错误状态
 当ERROR=0时，正常
 当ERROR≠0时，异常
 请参照表 22.16。
- SUM（1个字节）：校验和

表 22.15 状态码

代码	内容
H'11	等待器件的选择
H'12	等待时钟模式的选择
H'13	等待位速率的选择
H'1F	等待编程 / 擦除状态转移（位速率选择结束）
H'31	正在擦除用户 MAT 和用户引导 MAT
H'3F	等待编程 / 擦除的选择（擦除结束）
H'4F	等待编程数据的接收（编程结束）
H'5F	等待擦除块的指定（擦除结束）

表 22.16 错误码

代码	内容
H'00	无错误
H'11	校验和错误
H'21	器件代码不相同错误
H'22	时钟模式不相同错误
H'24	位速率的选择错误
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	块号错误
H'2A	地址错误
H'2B	数据长错误（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

22.9.2 过程程序和编程数据的可保存区域

本书说明的编程 / 擦除过程程序和编程数据的可保存区在内部 RAM 中。但是，如果满足以下条件，这些程序也能在外部空间区等其他区域执行。

1. 因为从 FTDAR 寄存器指定的内部 RAM 地址下载内部编程 / 擦除程序，所以此区域不能使用。
2. 因为内部编程 / 擦除程序至少使用 128 字节的堆栈区，所以必须确保该区域。
3. 因为在将 SCO 位置 1 进行下载请求的处理时发生 MAT 转换，所以必须在内部 RAM 中执行。
4. 在开始编程 / 擦除前（在判断下载结果前）能存取闪存。对于不能存取外部空间的模式（如单芯片模式），必须在编程 / 擦除前，将需要的过程程序、中断向量、中断处理程序 and 用户分支处理程序等传送到内部 RAM。
5. 因为在编程 / 擦除处理中不能存取闪存，所以通过下载到内部 RAM 中的程序来执行。启动此程序的过程程序、编程 / 擦除中的用户分支目标的用户程序、中断的向量表和中断处理程序的执行区也必须在内部 RAM（除闪存外）和外部总线空间中。
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。
如果在编程 / 擦除结束后立刻更改 LSI 模式进行复位时，就必须设定至少 100μs 的复位期间（ $\overline{\text{RES}}=0$ 的期间）。另外，因为在编程 / 擦除处理中禁止转移到复位状态，如果错误地进行复位，就必须在长于通常的 100μs 复位期间后解除复位。
7. 在用户引导模式中对用户 MAT 进行编程 / 擦除时，必须通过 FMATS 在内部 RAM 中进行 MAT 转换（请参照“22.8.1 用户 MAT 和用户引导 MAT 的转换”）。必须注意：现在选择了哪个 MAT。
8. 如果编程处理的参数 FMPDR 所指示的编程数据保存区域在闪存中，就判断为错误，必须暂时将编程数据传送到内部 RAM，并将 FMPDR 指示的地址设定在闪存空间以外。

基于这些条件，根据各运行模式 / 处理内容的组合，编程数据保存区域和可执行区如下表所示：

表 22.17 可执行的 MAT

处理	启动模式	
	用户编程模式	用户引导模式 *
编程	表 22.18(1)	表 22.18(3)
擦除	表 22.18(2)	表 22.18(4)

【注】 * 能对用户 MAT 进行编程 / 擦除。

表 22.18(1) 用户编程模式中编程处理可使用的区域

项目	可保存 / 执行的区域			选择的 MAT	
	内部 RAM	用户 MAT	外部空间	用户 MAT	嵌入式程序保存 MAT
编程数据的保存区	○	× *	○	—	—
下载内部程序的选择	○	○	○	○	
给键寄存器写 H'A5 的处理	○	○	○	○	
FCCS 的 SCO=1 的设定 (下载)	○	×	×		○
键寄存器的清除	○	○	○	○	
下载结果的判断	○	○	○	○	
下载的错误处理	○	○	○	○	
初始化参数的设定	○	○	○	○	
初始化	○	×	×	○	
初始化结果的判断	○	○	○	○	
初始化的错误处理	○	○	○	○	
中断处理程序	○	×	○	○	
给键寄存器写 H'5A 的处理	○	○	○	○	
编程参数的设定	○	×	○	○	
编程	○	×	×	○	
编程结果的判断	○	×	○	○	
编程的错误处理	○	×	○	○	
键寄存器的清除	○	×	○	○	

【注】 * 只要预先传送到内部 RAM，就能使用。

表 22.18(2) 用户编程模式中擦除处理可使用的区域

项目	可保存 / 执行的区域			选择的 MAT	
	内部 RAM	用户 MAT	外部空间	用户 MAT	嵌入式程序保存 MAT
下载内部程序的选择	○	○	○	○	
给键寄存器写 H'A5 的处理	○	○	○	○	
FCCS 的 SCO=1 的设定 (下载)	○	×	×		○
键寄存器的清除	○	○	○	○	
下载结果的判断	○	○	○	○	
下载的错误处理	○	○	○	○	
初始化参数的设定	○	○	○	○	
初始化	○	×	×	○	
初始化结果的判断	○	○	○	○	
初始化的错误处理	○	○	○	○	
中断处理程序	○	×	○	○	
给键寄存器写 H'5A 的处理	○	○	○	○	
擦除参数的设定	○	×	○	○	
擦除	○	×	×	○	
擦除结果的判断	○	×	○	○	
擦除的错误处理	○	×	○	○	
键寄存器的清除	○	×	○	○	

擦
除
步
骤

表 22.18(3) 用户引导模式中编程处理可使用的区域

	项目	可保存 / 执行的区域				选择的 MAT	
		内部 RAM	用户引导 MAT	外部空间	用户 MAT	用户引导 MAT	嵌入式程序保存 MAT
编 程 步 骤 ↓	编程数据的保存区	○	× *1	○	—	—	—
	下载内部程序的选择	○	○	○		○	
	给键寄存器写 H'A5 的处理	○	○	○		○	
	FCCS 的 SCO=1 的设定 (下载)	○	×	×			○
	键寄存器的清除	○	○	○		○	
	下载结果的判断	○	○	○		○	
	下载的错误处理	○	○	○		○	
	初始化参数的设定	○	○	○		○	
	初始化	○	×	×		○	
	初始化结果的判断	○	○	○		○	
	初始化的错误处理	○	○	○		○	
	中断处理程序	○	×	○		○	
	通过 FMATS 进行的 MAT 转换	○	×	×	○		
	给键寄存器写 H'5A 的处理	○	×	○	○		
	编程参数的设定	○	×	○	○		
	编程	○	×	×	○		
	编程结果的判断	○	×	○	○		
	编程的错误处理	○	× *2	○	○		
	键寄存器的清除	○	×	○	○		
	通过 FMATS 进行的 MAT 转换	○	×	×		○	

【注】 *1 只要预先传送到内部 RAM，就能使用。

*2 只要在内部 RAM 中转换 FMATS 后，就能使用。

表 22.18(4) 用户引导模式中擦除处理可使用的区域

项目	可保存 / 执行的区域				选择的 MAT	
	内部 RAM	用户引导 MAT	外部空间	用户 MAT	用户引导 MAT	嵌入式程序保存 MAT
下载内部程序的选择	○	○	○		○	
给键寄存器写 H'A5 的处理	○	○	○		○	
FCCS 的 SCO=1 的设定 (下载)	○	×	×			○
键寄存器的清除	○	○	○		○	
下载结果的判断	○	○	○		○	
下载的错误处理	○	○	○		○	
初始化参数的设定	○	○	○		○	
初始化	○	×	×		○	
初始化结果的判断	○	○	○		○	
初始化的错误处理	○	○	○		○	
中断处理程序	○	×	○		○	
通过 FMATS 进行的 MAT 转换	○	×	×		○	
给键寄存器写 H'5A 的处理	○	×	○	○		
擦除参数的设定	○	×	○	○		
擦除	○	×	×	○		
擦除结果的判断	○	×	○	○		
擦除的错误处理	○	×	○	○		
键寄存器的清除	○	×	○	○		
通过 FMATS 进行的 MAT 转换	○	×	×		○	

【注】 * 只要在内部 RAM 中转换 FMATS 后，就能使用。

22.10 编程器模式

在编程器模式中，和单一闪存相同，能通过插座适配器使用 PROM 编程器进行编程和擦除。请使用支持瑞萨 256K 字节闪存内置型单片机器件类型（F-ZTATxxxx）的 PROM 编程器。

第 23 章 RAM

本 LSI 内置高速静态 RAM。内部 RAM 通过 32 位数据总线（L 总线）连接 CPU，通过 32 位数据总线（I 总线）连接数据传送控制器（DTC）。能以 8 位、16 位或者 32 位存取内部 RAM。根据各产品，将内部 RAM 分配到图 23.1 所示的地址，按地址分为页 0 和页 1。CPU（经过 L 总线）和 DTC（经过 I 总线）能存取 RAM。当不同总线同时对同一页请求存取时，优先级为 I 总线（DTC）> L 总线（CPU）。因为这样的竞争会降低 RAM 的存取性能，所以为了尽量不发生竞争，建议使用软件对策。例如，按总线存取不同的页，这样就不会发生竞争。只要不发生页竞争，L 总线（CPU）的存取就为 1 个周期。I 总线（DTC）的存取因内部时钟（I ϕ ）和总线时钟（B ϕ ）的比或者 DTC 的运行状态等不同。在睡眠模式、软件待机模式、上电复位和手动复位时保持内部 RAM 的内容，但是在深度软件待机模式中不保持内部 RAM 的内容。

通过 RAM 控制寄存器（RAMCR）的 RAME 位，能控制 RAM 的有效或者无效。有关 RAMCR 请参照“24.3.7 RAM 控制寄存器（RAMCR）”。

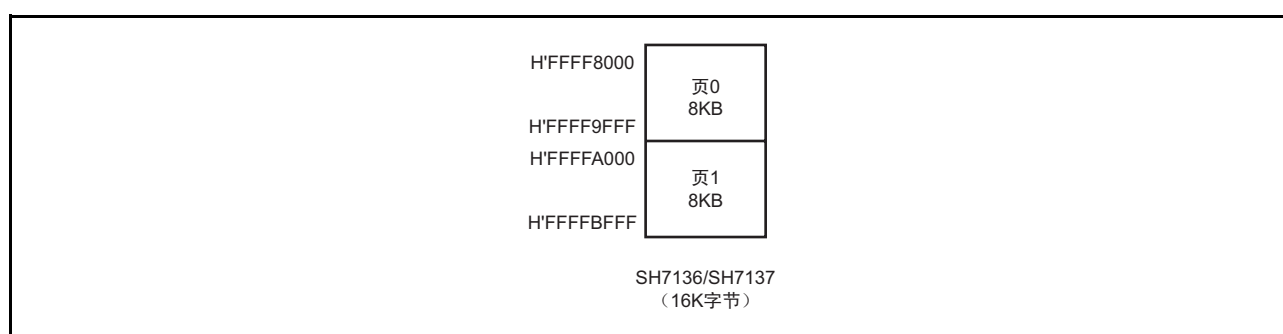


图 23.1 内部 RAM 地址

23.1 使用时的注意事项

23.1.1 模块待机模式的设定

能通过待机控制寄存器设定允许或者禁止 RAM 的存取。初始值为允许 RAM 的存取。能通过设定模块待机模式禁止 RAM 的存取。详细内容请参照“第 24 章 低功耗模式”。

23.1.2 地址错误

如果在写 RAM 时发生地址错误，就有可能破坏 RAM 的内容。

23.1.3 RAM 的初始值

在接通电源后写 RAM 前，RAM 的初始值为不定值。

第 24 章 低功耗模式

本 LSI 支持的低功耗模式有：睡眠模式、软件待机模式、深度软件待机模式和模块待机功能。

24.1 特点

- 支持睡眠模式、软件待机模式、模块待机功能和深度软件待机模式。

24.1.1 低功耗模式的种类

低功耗模式有如下的模式和功能：

- 睡眠模式
- 软件待机模式
- 深度软件待机模式
- 模块待机功能

从程序的执行状态转移到各模式的方法、各模式的 CPU 和外围模块等的状态以及各模式的解除方法如表 24.1 所示。

表 24.1 低功耗模式的状态

低功耗模式	转移方法	状态					解除方法
		CPG	CPU	CPU 寄存器	内部 存储器	内部 外围模块	
睡眠模式	在 STBCR1 的 STBY 位为 0 的状态下执行 SLEEP 指令	运行	停止	保持	运行	运行	复位
软件待机模式	在 STBCR1 的 STBY 位和 STBCR6 的 STBYMD 位都为 1 的状态下执行 SLEEP 指令	停止	停止	保持	停止 (保持内容)	停止	1. 由 NMI 或者 IRQ 引起的中断 2. 由 $\overline{\text{RES}}$ 引脚引起的上电复位
深度软件待机模式	在 STBCR1 的 STBY 位和 STBCR6 的 STBYMD 位分别为 1 和 0 的状态下执行 SLEEP 指令	停止	停止	不定	停止 (内容不定)	停止	由 $\overline{\text{RES}}$ 引脚引起的上电复位
模块待机功能	将 STBCR2 ~ 5 的 MSTP 位置 1	运行	运行	保持	停止指定的 模块 (保持内容)	停止指定的 模块	1. 将 MSTP 位清 0 2. 上电复位 (MSTP 位的初始值为 0 的模块)

【注】 有关各模式的内部外围模块的寄存器状态和引脚状态，请分别参照“25.3 各运行模式的寄存器状态”和“附录 A. 引脚状态”。

24.2 输入 / 输出引脚

低功耗模式关联的引脚结构如表 24.2 所示。

表 24.2 引脚结构

名称	引脚名	输入 / 输出	功能
上电复位	RES	输入	是上电复位的输入信号，通过低电平进行上电复位。
手动复位	MRES	输入	是手动复位的输入信号，通过低电平进行手动复位。

24.3 寄存器说明

低功耗模式关联的寄存器如下所示。有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 25 章 寄存器一览”。

表 24.3 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
待机控制寄存器 1	STBCR1	R/W	H'00	H'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

24.3.1 待机控制寄存器 1（STBCR1）

STBCR1 是 8 位可读写寄存器，指定低功耗模式的状态。

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

位	位名	初始值	R/W	说明
7	STBY	0	R/W	待机 指定向软件待机模式的转移。 0: 通过执行 SLEEP 指令，转移到睡眠模式。 1: 通过执行 SLEEP 指令，转移到软件待机模式 / 深度软件待机模式。
6 ~ 0	—	全 0	R	保留位 读写值总是 0。

24.3.2 待机控制寄存器 2 (STBCR2)

STBCR2 是 8 位可读写寄存器，控制低功耗模式时的各模块运行。

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	初始值	R/W	说明
7	MSTP7	0	R/W	模块停止位 7 如果将此位置 1，就停止向 RAM 提供时钟。 0: RAM 运行 1: 停止向 RAM 提供时钟
6	MSTP6	0	R/W	模块停止位 6 如果将此位置 1，就停止向 ROM 提供时钟。 0: ROM 运行 1: 停止向 ROM 提供时钟
5	—	1	R	保留位 读写值总是 1。
4	MSTP4	1	R/W	模块停止位 4 如果将此位置 1，就停止向 DTC 提供时钟。 0: DTC 运行 1: 停止向 DTC 提供时钟
3	—	1	R	保留位 读写值总是 1。
2 ~ 0	—	全 0	R	保留位 读写值总是 0。

24.3.3 待机控制寄存器 3 (STBCR3)

STBCR3 是 8 位可读写寄存器，控制低功耗模式时的各模块运行。

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

位	位名	初始值	R/W	说明
7	MSTP15	1	R/W	模块停止位 15 如果将此位置 1，就停止向 I ² C2 提供时钟。 0: I ² C2 运行 1: 停止向 I ² C2 提供时钟
6	—	1	R	保留位 读写值总是 1。

位	位名	初始值	R/W	说明
5	MSTP13	1	R/W	模块停止位 13 如果将此位置 1，就停止向 SCI_2 提供时钟。 0: SCI_2 运行 1: 停止向 SCI_2 提供时钟
4	MSTP12	1	R/W	模块停止位 12 如果将此位置 1，就停止向 SCI_1 提供时钟。 0: SCI_1 运行 1: 停止向 SCI_1 提供时钟
3	MSTP11	1	R/W	模块停止位 11 如果将此位置 1，就停止向 SCI_0 提供时钟。 0: SCI_0 运行 1: 停止向 SCI_0 提供时钟
2	MSTP10	1	R/W	模块停止位 10 如果将此位置 1，就停止向 SSU 提供时钟。 0: SSU 运行 1: 停止向 SSU 提供时钟
1	—	1	R	保留位 读写值总是 1。
0	MSTP8	1	R/W	模块停止位 8 如果将此位置 1，就停止向 RCAN-ET_0 提供时钟。 0: RCAN-ET_0 运行 1: 停止向 RCAN-ET_0 提供时钟

24.3.4 待机控制寄存器 4（STBCR4）

STBCR4 是 8 位可读写寄存器，控制低功耗模式时的各模块运行。

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	初始值	R/W	说明
7	MSTP23	1	R/W	模块停止位 23 如果将此位置 1，就停止向 MTU2S 提供时钟。 0: MTU2S 运行 1: 停止向 MTU2S 提供时钟
6	MSTP22	1	R/W	模块停止位 22 如果将此位置 1，就停止向 MTU2 提供时钟。 0: MTU2 运行 1: 停止向 MTU2 提供时钟
5	MSTP21	1	R/W	模块停止位 21 如果将此位置 1，就停止向 CMT 提供时钟。 0: CMT 运行 1: 停止向 CMT 提供时钟

位	位名	初始值	R/W	说明
4	MSTP20	1	R/W	模块停止位 20 如果将此位置 1，就停止向 A/D_1 提供时钟。 0: A/D_1 运行 1: 停止向 A/D_1 提供时钟
3	MSTP19	1	R/W	模块停止位 19 如果将此位置 1，就停止向 A/D_0 提供时钟。 0: A/D_0 运行 1: 停止向 A/D_0 提供时钟
2 ~ 0	—	全 1	R	保留位 读写值总是 1。

24.3.5 待机控制寄存器 5（STBCR5）

STBCR5 是 8 位可读写寄存器，指定低功耗模式的状态。

位:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	MSTP 25	MSTP 24
初始值:	0	0	0	0	0	0	1	1
R/W:	R	R	R	R	R	R	R/W	R/W

位	位名	初始值	R/W	说明
7 ~ 2	—	全 0	R	保留位 读写值总是 0。
1	MSTP25	1	R/W	模块停止位 25 如果将此位置 1，就停止向 AUD 提供时钟。 0: AUD 运行 1: 停止向 AUD 提供时钟
0	MSTP24	1	R/W	模块停止位 24 如果将此位置 1，就停止向 UBC 提供时钟。 0: UBC 运行 1: 停止向 UBC 提供时钟

24.3.6 待机控制寄存器 6（STBCR6）

STBCR6 是 8 位可读写寄存器，指定低功耗模式的状态。

位:	7	6	5	4	3	2	1	0
	AUD SRST	HIZ	—	—	—	—	STBY MD	—
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R/W	R

位	位名	初始值	R/W	说明
7	AUDSRST	0	R/W	AUD 软件复位 通过软件控制 AUD 的复位。 如果将此位置 0，AUD 模块就进入上电复位状态。 0: 设定为 AUD 复位状态 1: 解除 AUD 的复位 必须在 STBCR5 的 MSTP25 位为 0 的状态下将此位置 1。
6	HIZ	0	R/W	端口的高阻抗状态 选择在软件待机模式时是保持引脚状态还是将引脚状态设定为高阻抗状态。 0: 在软件待机模式时，保持引脚状态。 1: 在软件待机模式时，将引脚状态设定为高阻抗状态。
5 ~ 2	—	全 0	R	保留位 读写值总是 0。
1	STBYMD	0	R/W	软件待机模式的选择 在 STBCR1 的 STBY 位为 1 的状态下执行 SLEEP 指令时，选择是转移到软件待机模式还是转移到深度软件待机模式。 0: 转移到深度软件待机模式 1: 转移到软件待机模式
0	—	0	R	保留位 读写值总是 0。

24.3.7 RAM 控制寄存器 (RAMCR)

RAMCR 是 8 位可读写寄存器，允许或者禁止内部 RAM 的存取。

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

位	位名	初始值	R/W	说明
7 ~ 5	—	全 0	R	保留位 读写值总是 0。
4	RAME	1	R/W	RAM 允许 选择内部 RAM 有效或者无效。 0: 内部 RAM 无效 1: 内部 RAM 有效 如果将此位清 0，就不能存取内部 RAM。此时，读内部 RAM 和取指令的值为不定值，忽视内部 RAM 的写操作。 在将此位清 0（将内部 RAM 设定为无效）时，内部 RAM 的存取指令不能紧接在 RAMCR 的写指令之后。否则，就不能保证正常的存取。 在将此位置 1（将内部 RAM 设定为有效）时，RAMCR 的读指令必须紧接在 RAMCR 的写指令之后。如果内部 RAM 的存取指令紧接在 RAMCR 的写指令之后，就不能保证正常的存取。
3 ~ 0	—	全 0	R	保留位 读写值总是 0。

24.4 睡眠模式

24.4.1 向睡眠模式的转移

如果在 STBCR1 的 STBY 位为 0 的状态下执行 SLEEP 指令，就从程序的执行状态转移到睡眠模式，但是在释放总线的期间（给 BREQ 引脚输入低电平）不能转移到睡眠模式。CPU 在执行 SLEEP 指令后停止运行，但是保持 CPU 的寄存器内容，并且内部外围模块继续运行。

24.4.2 睡眠模式的解除

通过复位解除睡眠模式。

不能通过中断解除睡眠模式。

(1) 通过复位进行的解除

通过 RES 引脚的上电复位、MRES 引脚的手动复位或者 WDT 的内部上电复位 / 内部手动复位，解除睡眠模式。

24.5 软件待机模式

24.5.1 向软件待机模式的转移

如果在 STBCR1 的 STBY 位和 STBCR6 的 STBYMD 位都为 1 的状态下执行 SLEEP 指令，就从程序的执行状态转移到软件待机模式，但是在释放总线的期间（给 BREQ 引脚输入低电平）不能转移到软件待机模式。另外，必须在 DTC 停止运行后执行 SLEEP 指令。在软件待机模式中，CPU、时钟和内部外围模块都停止运行。

保持 CPU 的寄存器内容和内部 RAM 的数据，内部外围模块的寄存器有可能被初始化。有关软件待机模式的外围模块的寄存器状态和引脚状态，请分别参照“25.3 各运行模式的寄存器状态”和“附录 A. 引脚状态”。

转移到软件待机模式的步骤如下所示：

1. 将 WDT 的定时器控制寄存器（WTCSR）的 TME 位置 0，停止 WDT 的运行。
2. 将 WDT 的定时器计数器（WTCNT）置 0，并给 WTCSR 寄存器的 CKS2～CKS0 位设定与指定振荡稳定时间对应的值。
3. 在 DTC 运行时，停止 DTC 的运行。
4. 在总线为释放状态（给 BREQ 引脚输入低电平）时，获取总线（给 BREQ 引脚输入高电平）。
5. 在将 STBCR1 的 STBY 位和 STBCR6 的 STBYMD 位置 1 后，执行 SLEEP 指令。
6. 进入软件待机模式，停止 LSI 内部的时钟。

24.5.2 软件待机模式的解除

通过中断（NMI 和 IRQ）或者复位，解除软件待机模式。

(1) 通过中断进行的解除

能通过内部 WDT 进行热启动。如果检测到 NMI 或者 IRQ 中断（边沿检测），就在经过 WDT 的定时器的控制 / 状态寄存器设定的时间后，给整个 LSI 提供时钟，并解除软件待机模式。然后，执行中断异常处理。

但是，如果 IRQ 中断优先级低于 CPU 的状态寄存器（SR）设定的中断屏蔽级，就既不接受中断请求也不解除软件待机模式。

另外，如果将 NMI 引脚设定为下降沿检测，就必须在转移到软件待机模式前将 NMI 引脚置为高电平；如果将 NMI 引脚设定为上升沿检测，就必须在转移到软件待机模式前将 NMI 引脚置为低电平。

同样地，如果将 IRQ 引脚设定为下降沿检测，就必须在转移到软件待机模式前将 IRQ 引脚置为高电平；如果将 IRQ 引脚设定为上升沿检测，就必须在转移到软件待机模式前将 IRQ 引脚置为低电平。

(2) 通过上电复位进行的解除

通过 $\overline{\text{RES}}$ 引脚的上电复位，解除软件待机模式。 $\overline{\text{RES}}$ 引脚必须在时钟振荡稳定前保持低电平。

24.6 深度软件待机模式

24.6.1 向深度软件待机模式的转移

如果在 STBCR1 的 STBY 位和 STBCR6 的 STBYMD 位分别为 1 和 0 的状态下执行 SLEEP 指令，就从程序执行状态转移到深度软件待机模式，但是在释放总线的期间（给 $\overline{\text{BREQ}}$ 引脚输入低电平）不能转移到深度软件待机模式。另外，必须在 DTC 停止运行后执行 SLEEP 指令。在深度软件待机模式中，CPU、时钟和内部外围模块都停止运行，并且还切断本 LSI 的内部电源。

CPU 的寄存器内容和内部 RAM 的数据为不定值，内部外围模块的寄存器被初始化。有关深度软件待机模式的引脚状态，请参照“附录 A. 引脚状态”。

转移到深度软件待机模式的步骤如下所示：

1. 将 WDT 的定时器控制寄存器（WTCSR）的 TME 位置 0，停止 WDT 的运行。
2. 在 DTC 运行时，停止 DTC 的运行。
3. 在总线为释放状态（给 $\overline{\text{BREQ}}$ 引脚输入低电平）时，获取总线（给 $\overline{\text{BREQ}}$ 引脚输入高电平）。
4. 将 STBCR1 的 STBY 位和 STBCR6 的 STBYMD 位分别置 1 和 0 后，执行 SLEEP 指令。
5. 进入深度软件待机模式，停止 LSI 内部的时钟，并切断本 LSI 的内部电源。

24.6.2 深度软件待机模式的解除

通过 $\overline{\text{RES}}$ 引脚的上电复位，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。

24.7 模块待机功能

24.7.1 向模块待机功能的转移

能通过将待机控制寄存器 2～5（STBCR2～5）的各 MSTP 位置 1，停止向各自对应的内部外围模块提供时钟。通过此功能，能降低正常模式中的功耗。

不能存取被设定为模块待机模式的外围模块寄存器。有关模块待机模式的外围模块的寄存器状态，请参照“25.3 各运行模式的寄存器状态”。

24.7.2 模块待机功能的解除

通过将 STBCR2～5 的各 MSTP 位清 0，解除模块待机功能。对于 MSTP 位的初始值为 0 的模块，也能通过上电复位进行解除。

24.8 使用时的注意事项

24.8.1 振荡稳定待机中的消耗电流

振荡稳定待机中的消耗电流会增加。

24.8.2 深度软件待机模式

不能使用深度软件待机模式。

24.8.3 执行 SLEEP 指令时

在通过执行 SLEEP 指令转移到睡眠模式或者软件待机模式时，必须实施以下任意一个对策。

对策 A. 在执行 SLEEP 指令前，必须停止 DTC 运行并且不能发生内部外围模块的中断、IRQ 中断和 NMI 中断。

对策 B. 在执行 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 25 章 寄存器一览

寄存器一览汇总了有关内部寄存器的地址、位结构以及各运行模式的状态信息。记载方法如下：

1. 寄存器地址一览（地址顺序）

- 按分配地址从小到大的顺序记载。
- 在寄存器名栏用“—”表示保留地址。
不能存取保留地址。
- 在地址为 16 位或者 32 位时，记载 MSB 侧的地址。
- 根据模块名分类。
- 表示存取长度。

2. 位结构一览

- 按照“寄存器地址一览（地址顺序）”的顺序记载位结构。
- 在位名栏中用“—”表示保留位。
- 位名为空白的栏，表示该寄存器全部分配给计数器或者数据。
- 在为 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（地址顺序）”的顺序记载寄存器状态。
- 表示基本运行模式的寄存器状态。在有内部模块固有的复位等时，请参照内部模块的章节。

25.1 寄存器地址一览（地址顺序）

存取长度表示位数。

存取状态数表示指定的基准时钟的状态数。

这些值是 B:8 位存取、W:16 位存取和 L:32 位存取时的值。

【注】 1. 禁止存取未定义或者保留的地址。由于不能保证存取这些寄存器时的运行及其后续运行，因此禁止存取。

2. 在存取 RCAN-ET 的邮箱区时，可能会发生 0 ~ 5 个 P ϕ 的等待。

寄存器名	略称	位数	地址	模块	存取长度	存取状态数	连接的 总线宽度
串行模式寄存器_0	SCSMR_0	8	H'FFFC000	SCI (通道 0)	8	P ϕ 基准 B:2	16 位
位速率寄存器_0	SCBRR_0	8	H'FFFC002		8		
串行控制寄存器_0	SCSCR_0	8	H'FFFC004		8		
发送数据寄存器_0	SCTDR_0	8	H'FFFC006		8		
串行状态寄存器_0	SCSSR_0	8	H'FFFC008		8		
接收数据寄存器_0	SCRDR_0	8	H'FFFC00A		8		
串行方向控制寄存器_0	SCSDCR_0	8	H'FFFC00C		8		
串行端口寄存器_0	SCSPTR_0	8	H'FFFC00E		8		
串行模式寄存器_1	SCSMR_1	8	H'FFFC080	SCI (通道 1)	8	P ϕ 基准 B:2	16 位
位速率寄存器_1	SCBRR_1	8	H'FFFC082		8		
串行控制寄存器_1	SCSCR_1	8	H'FFFC084		8		
发送数据寄存器_1	SCTDR_1	8	H'FFFC086		8		
串行状态寄存器_1	SCSSR_1	8	H'FFFC088		8		
接收数据寄存器_1	SCRDR_1	8	H'FFFC08A		8		
串行方向控制寄存器_1	SCSDCR_1	8	H'FFFC08C		8		
串行端口寄存器_1	SCSPTR_1	8	H'FFFC08E		8		
串行模式寄存器_2	SCSMR_2	8	H'FFFC100	SCI (通道 2)	8	P ϕ 基准 B:2	16 位
位速率寄存器_2	SCBRR_2	8	H'FFFC102		8		
串行控制寄存器_2	SCSCR_2	8	H'FFFC104		8		
发送数据寄存器_2	SCTDR_2	8	H'FFFC106		8		
串行状态寄存器_2	SCSSR_2	8	H'FFFC108		8		
接收数据寄存器_2	SCRDR_2	8	H'FFFC10A		8		
串行方向控制寄存器_2	SCSDCR_2	8	H'FFFC10C		8		
串行端口寄存器_2	SCSPTR_2	8	H'FFFC10E		8		
定时器的控制寄存器_3	TCR_3	8	H'FFFC200	MTU2	8、16、32	MP ϕ 基准 B:2、W:2、L:4	16 位
定时器的控制寄存器_4	TCR_4	8	H'FFFC201		8		
定时器的模式寄存器_3	TMDR_3	8	H'FFFC202		8、16		
定时器的模式寄存器_4	TMDR_4	8	H'FFFC203		8		
定时器的 I/O 控制寄存器 H_3	TIORH_3	8	H'FFFC204		8、16、32		
定时器的 I/O 控制寄存器 L_3	TIORL_3	8	H'FFFC205		8		
定时器的 I/O 控制寄存器 H_4	TIORH_4	8	H'FFFC206		8、16		
定时器的 I/O 控制寄存器 L_4	TIORL_4	8	H'FFFC207		8		
定时器的中断允许寄存器_3	TIER_3	8	H'FFFC208		8、16		
定时器的中断允许寄存器_4	TIER_4	8	H'FFFC209		8		

寄存器名	略称	位数	地址	模块	存取长度	存取状态数	连接的 总线宽度
定时器的输出主控允许寄存器	TOER	8	H'FFFC20A	MTU2	8	MP ϕ 基准 B:2、W:2、L:4	16 位
定时器的门控寄存器	TGCR	8	H'FFFC20D		8		
定时器的输出控制寄存器 1	TOCR1	8	H'FFFC20E		8、16		
定时器的输出控制寄存器 2	TOCR2	8	H'FFFC20F		8		
定时器的计数器_3	TCNT_3	16	H'FFFC210		16、32		
定时器的计数器_4	TCNT_4	16	H'FFFC212		16		
定时器的周期数据寄存器	TCDR	16	H'FFFC214		16、32		
定时器的空载时间数据寄存器	TDDR	16	H'FFFC216		16		
定时器的通用寄存器 A_3	TGRA_3	16	H'FFFC218		16、32		
定时器的通用寄存器 B_3	TGRB_3	16	H'FFFC21A		16		
定时器的通用寄存器 A_4	TGRA_4	16	H'FFFC21C		16、32		
定时器的通用寄存器 B_4	TGRB_4	16	H'FFFC21E		16		
定时器的副计数器	TCNTS	16	H'FFFC220		16、32		
定时器的周期缓冲寄存器	TCBR	16	H'FFFC222		16		
定时器的通用寄存器 C_3	TGRC_3	16	H'FFFC224		16、32		
定时器的通用寄存器 D_3	TGRD_3	16	H'FFFC226		16		
定时器的通用寄存器 C_4	TGRC_4	16	H'FFFC228		16、32		
定时器的通用寄存器 D_4	TGRD_4	16	H'FFFC22A		16		
定时器的状态寄存器_3	TSR_3	8	H'FFFC22C		8、16		
定时器的状态寄存器_4	TSR_4	8	H'FFFC22D		8		
定时器的中断减少设定寄存器	TITCR	8	H'FFFC230		8、16		
定时器的中断减少次数计数器	TITCNT	8	H'FFFC231		8		
定时器的缓冲传送设定寄存器	TBTER	8	H'FFFC232		8		
定时器的空载时间允许寄存器	TDER	8	H'FFFC234		8		
定时器的输出电平缓冲寄存器	TOLBR	8	H'FFFC236		8		
定时器的缓冲运行传送模式寄存器_3	TBTM_3	8	H'FFFC238		8、16		
定时器的缓冲运行传送模式寄存器_4	TBTM_4	8	H'FFFC239		8		
定时器的 A/D 转换开始请求的控制寄存器	TADCR	16	H'FFFC240		16		
定时器的 A/D 转换开始请求的周期设定寄存器 A_4	TADCORA_4	16	H'FFFC244		16、32		
定时器的 A/D 转换开始请求的周期设定寄存器 B_4	TADCORB_4	16	H'FFFC246		16		
定时器的 A/D 转换开始请求的周期设定缓冲寄存器 A_4	TADCOBRA_4	16	H'FFFC248		16、32		
定时器的 A/D 转换开始请求的周期设定缓冲寄存器 B_4	TADCOBRB_4	16	H'FFFC24A		16		
定时器的波形控制寄存器	TWCR	8	H'FFFC260		8		
定时器的开始寄存器	TSTR	8	H'FFFC280		8、16		

寄存器名	略称	位数	地址	模块	存取长度	存取状态数	连接的 总线宽度
定时器的同步寄存器	TSYR	8	H'FFFFC281	MTU2	8	MP ϕ 基准 B:2、W:2、L:4	16 位
定时器的计数器同步开始寄存器	TCSYSTR	8	H'FFFFC282		8		
定时器的读写允许寄存器	TRWER	8	H'FFFFC284		8		
定时器的控制寄存器_0	TCR_0	8	H'FFFFC300		8、16、32		
定时器的模式寄存器_0	TMDR_0	8	H'FFFFC301		8		
定时器的 I/O 控制寄存器 H_0	TIORH_0	8	H'FFFFC302		8、16		
定时器的 I/O 控制寄存器 L_0	TIORL_0	8	H'FFFFC303		8		
定时器的中断允许寄存器_0	TIER_0	8	H'FFFFC304		8、16、32		
定时器的状态寄存器_0	TSR_0	8	H'FFFFC305		8		
定时器的计数器_0	TCNT_0	16	H'FFFFC306		16		
定时器的通用寄存器 A_0	TGRA_0	16	H'FFFFC308		16、32		
定时器的通用寄存器 B_0	TGRB_0	16	H'FFFFC30A		16		
定时器的通用寄存器 C_0	TGRC_0	16	H'FFFFC30C		16、32		
定时器的通用寄存器 D_0	TGRD_0	16	H'FFFFC30E		16		
定时器的通用寄存器 E_0	TGRE_0	16	H'FFFFC320		16、32		
定时器的通用寄存器 F_0	TGRF_0	16	H'FFFFC322		16		
定时器的中断允许寄存器 2_0	TIER2_0	8	H'FFFFC324		8、16		
定时器的状态寄存器 2_0	TSR2_0	8	H'FFFFC325		8		
定时器的缓冲运行传送模式寄存器_0	TBTM_0	8	H'FFFFC326		8		
定时器的控制寄存器_1	TCR_1	8	H'FFFFC380		8、16		
定时器的模式寄存器_1	TMDR_1	8	H'FFFFC381		8		
定时器的 I/O 控制寄存器_1	TIOR_1	8	H'FFFFC382		8		
定时器的中断允许寄存器_1	TIER_1	8	H'FFFFC384		8、16、32		
定时器的状态寄存器_1	TSR_1	8	H'FFFFC385		8		
定时器的计数器_1	TCNT_1	16	H'FFFFC386		16		
定时器的通用寄存器 A_1	TGRA_1	16	H'FFFFC388		16、32		
定时器的通用寄存器 B_1	TGRB_1	16	H'FFFFC38A		16		
定时器的输入捕捉控制寄存器	TICCR	8	H'FFFFC390		8		
定时器的控制寄存器_2	TCR_2	8	H'FFFFC400		8、16		
定时器的模式寄存器_2	TMDR_2	8	H'FFFFC401		8		
定时器的 I/O 控制寄存器_2	TIOR_2	8	H'FFFFC402		8		
定时器的中断允许寄存器_2	TIER_2	8	H'FFFFC404		8、16、32		
定时器的状态寄存器_2	TSR_2	8	H'FFFFC405		8		
定时器的计数器_2	TCNT_2	16	H'FFFFC406		16		
定时器的通用寄存器 A_2	TGRA_2	16	H'FFFFC408		16、32		
定时器的通用寄存器 B_2	TGRB_2	16	H'FFFFC40A		16		
定时器的计数器 U_5	TCNTU_5	16	H'FFFFC480		16、32		
定时器的通用寄存器 U_5	TGRU_5	16	H'FFFFC482		16		
定时器的控制寄存器 U_5	TCRU_5	8	H'FFFFC484		8		
定时器的 I/O 控制寄存器 U_5	TIORU_5	8	H'FFFFC486		8		

寄存器名	略称	位数	地址	模块	存取长度	存取状态数	连接的 总线宽度
定时器的计数器 V_5	TCNTV_5	16	H'FFFFC490	MTU2	16、32	MP ϕ 基准 B:2、W:2、L:4	16 位
定时器的通用寄存器 V_5	TGRV_5	16	H'FFFFC492		16		
定时器的控制寄存器 V_5	TCRV_5	8	H'FFFFC494		8		
定时器的 I/O 控制寄存器 V_5	TIORV_5	8	H'FFFFC496		8		
定时器的计数器 W_5	TCNTW_5	16	H'FFFFC4A0		16、32		
定时器的通用寄存器 W_5	TGRW_5	16	H'FFFFC4A2		16		
定时器的控制寄存器 W_5	TCRW_5	8	H'FFFFC4A4		8		
定时器的 I/O 控制寄存器 W_5	TIORW_5	8	H'FFFFC4A6		8		
定时器的状态寄存器 _5	TSR_5	8	H'FFFFC4B0		8		
定时器的中断允许寄存器 _5	TIER_5	8	H'FFFFC4B2		8		
定时器的开始寄存器 _5	TSTR_5	8	H'FFFFC4B4		8		
定时器的比较匹配清除寄存器	TCNTCMPCLR	8	H'FFFFC4B6		8		
定时器的控制寄存器 _3S	TCR_3S	8	H'FFFFC600	MTU2S	8、16、32	MI ϕ 基准 B:2、W:2、L:4	16 位
定时器的控制寄存器 _4S	TCR_4S	8	H'FFFFC601		8		
定时器的模式寄存器 _3S	TMDR_3S	8	H'FFFFC602		8、16		
定时器的模式寄存器 _4S	TMDR_4S	8	H'FFFFC603		8		
定时器的 I/O 控制寄存器 H_3S	TIORH_3S	8	H'FFFFC604		8、16、32		
定时器的 I/O 控制寄存器 L_3S	TIORL_3S	8	H'FFFFC605		8		
定时器的 I/O 控制寄存器 H_4S	TIORH_4S	8	H'FFFFC606		8、16		
定时器的 I/O 控制寄存器 L_4S	TIORL_4S	8	H'FFFFC607		8		
定时器的中断允许寄存器 _3S	TIER_3S	8	H'FFFFC608		8、16		
定时器的中断允许寄存器 _4S	TIER_4S	8	H'FFFFC609		8		
定时器的输出主控允许寄存器 S	TOERS	8	H'FFFFC60A		8		
定时器的门控寄存器 S	TGCRS	8	H'FFFFC60D		8		
定时器的输出控制寄存器 1S	TOCR1S	8	H'FFFFC60E		8、16		
定时器的输出控制寄存器 2S	TOCR2S	8	H'FFFFC60F		8		
定时器的计数器 _3S	TCNT_3S	16	H'FFFFC610		16、32		
定时器的计数器 _4S	TCNT_4S	16	H'FFFFC612		16		
定时器的周期数据寄存器 S	TCDRS	16	H'FFFFC614		16、32		
定时器的空载时间数据寄存器 S	TDDRS	16	H'FFFFC616		16		
定时器的通用寄存器 A_3S	TGRA_3S	16	H'FFFFC618		16、32		
定时器的通用寄存器 B_3S	TGRB_3S	16	H'FFFFC61A		16		
定时器的通用寄存器 A_4S	TGRA_4S	16	H'FFFFC61C		16、32		
定时器的通用寄存器 B_4S	TGRB_4S	16	H'FFFFC61E		16		
定时器的副计数器 S	TCNTSS	16	H'FFFFC620		16、32		
定时器的周期缓冲寄存器 S	TCBRS	16	H'FFFFC622		16		
定时器的通用寄存器 C_3S	TGRC_3S	16	H'FFFFC624		16、32		
定时器的通用寄存器 D_3S	TGRD_3S	16	H'FFFFC626		16		
定时器的通用寄存器 C_4S	TGRC_4S	16	H'FFFFC628		16、32		
定时器的通用寄存器 D_4S	TGRD_4S	16	H'FFFFC62A		16		

寄存器名	略称	位数	地址	模块	存取长度	存取状态数	连接的 总线宽度
定时器的状态寄存器_3S	TSR_3S	8	H'FFFC62C	MTU2S	8、16	M _l φ 基准 B:2、W:2、L:4	16 位
定时器的状态寄存器_4S	TSR_4S	8	H'FFFC62D		8		
定时器的中断减少设定寄存器 S	TITCRS	8	H'FFFC630		8、16		
定时器的中断减少次数计数器 S	TITCNTS	8	H'FFFC631		8		
定时器的缓冲传送设定寄存器 S	TBTERS	8	H'FFFC632		8		
定时器的空载时间允许寄存器 S	TDERS	8	H'FFFC634		8		
定时器的输出电平缓冲寄存器 S	TOLBRS	8	H'FFFC636		8		
定时器的缓冲运行传送模式寄存器_3S	TBTM_3S	8	H'FFFC638		8、16		
定时器的缓冲运行传送模式寄存器_4S	TBTM_4S	8	H'FFFC639		8		
定时器的 A/D 转换开始请求的控制寄存器 S	TADCRS	16	H'FFFC640		16		
定时器的 A/D 转换开始请求的周期设定寄存器 A_4S	TADCORA_4S	16	H'FFFC644		16、32		
定时器的 A/D 转换开始请求的周期设定寄存器 B_4S	TADCORB_4S	16	H'FFFC646		16		
定时器的 A/D 转换开始请求的周期设定缓冲寄存器 A_4S	TADCOBRA_4S	16	H'FFFC648		16、32		
定时器的 A/D 转换开始请求的周期设定缓冲寄存器 B_4S	TADCOBRB_4S	16	H'FFFC64A		16		
定时器的同步清除寄存器 S	TSYCRS	8	H'FFFC650		8		
定时器的波形控制寄存器 S	TWCRS	8	H'FFFC660		8		
定时器的开始寄存器 S	TSTRS	8	H'FFFC680		8、16		
定时器的同步寄存器 S	TSYRS	8	H'FFFC681		8		
定时器的读写允许寄存器 S	TRWERS	8	H'FFFC684		8		
定时器的计数器 U_5S	TCNTU_5S	16	H'FFFC880		16、32		
定时器的通用寄存器 U_5S	TGRU_5S	16	H'FFFC882		16		
定时器的控制寄存器 U_5S	TCRU_5S	8	H'FFFC884		8		
定时器的 I/O 控制寄存器 U_5S	TIORU_5S	8	H'FFFC886		8		
定时器的计数器 V_5S	TCNTV_5S	16	H'FFFC890		16、32		
定时器的通用寄存器 V_5S	TGRV_5S	16	H'FFFC892		16		
定时器的控制寄存器 V_5S	TCRV_5S	8	H'FFFC894		8		
定时器的 I/O 控制寄存器 V_5S	TIORV_5S	8	H'FFFC896		8		
定时器的计数器 W_5S	TCNTW_5S	16	H'FFFC8A0		16、32		
定时器的通用寄存器 W_5S	TGRW_5S	16	H'FFFC8A2		16		
定时器的控制寄存器 W_5S	TCRW_5S	8	H'FFFC8A4		8		
定时器的 I/O 控制寄存器 W_5S	TIORW_5S	8	H'FFFC8A6		8		
定时器的状态寄存器_5S	TSR_5S	8	H'FFFC8B0		8		
定时器的中断允许寄存器_5S	TIER_5S	8	H'FFFC8B2		8		
定时器的开始寄存器_5S	TSTR_5S	8	H'FFFC8B4		8		

寄存器名	略称	位数	地址	模块	存取长度	存取状态数	连接的 总线宽度
定时器的比较匹配清除寄存器 S	TCNTCMP CLRS	8	H'FFFFC8B6	MTU2S	8	M ϕ 基准 B:2、W:2、L:4	16 位
闪存代码控制 / 状态寄存器	FCCS	8	H'FFFFCC00	FLASH	8	P ϕ 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	P ϕ 基准 B:2、W:2、L:4	16 位
DTC 允许寄存器 B	DTCERB	16	H'FFFFCC82		8、16		
DTC 允许寄存器 C	DTCERC	16	H'FFFFCC84		8、16		
DTC 允许寄存器 D	DTCERD	16	H'FFFFCC86		8、16		
DTC 允许寄存器 E	DTCERE	16	H'FFFFCC88		8、16		
DTC 控制寄存器	DTCCR	8	H'FFFFCC90		8		
DTC 向量基址寄存器	DTCVBR	32	H'FFFFCC94		8、16、32		
I ² C 总线控制寄存器 1	ICCR1	8	H'FFFFCD80	I ² C2	8	P ϕ 基准 B:2	8 位
I ² C 总线控制寄存器 2	ICCR2	8	H'FFFFCD81		8		
I ² C 总线模式寄存器	ICMR	8	H'FFFFCD82		8		
I ² C 总线中断允许寄存器	ICIER	8	H'FFFFCD83		8		
I ² C 总线状态寄存器	ICSR	8	H'FFFFCD84		8		
从属地址寄存器	SAR	8	H'FFFFCD85		8		
I ² C 总线发送数据寄存器	ICDRT	8	H'FFFFCD86		8		
I ² C 总线接收数据寄存器	ICDRR	8	H'FFFFCD87		8		
NF2CYC 寄存器	NF2CYC	8	H'FFFFCD88		8		
SS 控制寄存器 H	SSCRH	8	H'FFFFCD00	SSU	8、16	P ϕ 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		
SS 发送数据寄存器 1	SSTDR1	8	H'FFFFCD07		8		
SS 发送数据寄存器 2	SSTDR2	8	H'FFFFCD08		8、16		
SS 发送数据寄存器 3	SSTDR3	8	H'FFFFCD09		8		
SS 接收数据寄存器 0	SSRDR0	8	H'FFFFCD0A		8、16		
SS 接收数据寄存器 1	SSRDR1	8	H'FFFFCD0B		8		
SS 接收数据寄存器 2	SSRDR2	8	H'FFFFCD0C		8、16		
SS 接收数据寄存器 3	SSRDR3	8	H'FFFFCD0D		8		
比较匹配定时器的开始寄存器	CMSTR	16	H'FFFFCE00	CMT	8、16、32	P ϕ 基准 B:2、W:2、L:4	16 位
比较匹配定时器的控制 / 状态寄存器 _0	CMCSR_0	16	H'FFFFCE02		8、16		

寄存器名	略称	位数	地址	模块	存取长度	存取状态数	连接的 总线宽度
比较匹配计数器_0	CMCNT_0	16	H'FFFCE04	CMT	8、16、32	Pφ 基准 B:2、W:2、L:4	16 位
比较匹配常数寄存器_0	CMCOR_0	16	H'FFFCE06		8、16		
比较匹配定时器的控制 / 状态寄存器_1	CMCSR_1	16	H'FFFCE08		8、16、32		
比较匹配计数器_1	CMCNT_1	16	H'FFFCE0A		8、16		
比较匹配常数寄存器_1	CMCOR_1	16	H'FFFCE0C		8、16、32		
输入电平控制 / 状态寄存器 1	ICSR1	16	H'FFFDD000	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'FFFDD002		8、16		
输入电平控制 / 状态寄存器 2	ICSR2	16	H'FFFDD004		8、16、32		
输出电平控制 / 状态寄存器 2	OCSR2	16	H'FFFDD006		8、16		
输入电平控制 / 状态寄存器 3	ICSR3	16	H'FFFDD008		8、16		
软件端口输出允许寄存器	SPOER	8	H'FFFDD00A		8		
端口输出允许控制寄存器 1	POECR1	8	H'FFFDD00B		8		
端口输出允许控制寄存器 2	POECR2	16	H'FFFDD00C		8、16		
端口 A 的数据寄存器 L	PADRL	16	H'FFFDD102	I/O	8、16	Pφ 基准 B:2、W:2、L:4	16 位
端口 A 的 IO 寄存器 L	PAIORL	16	H'FFFDD106	PFC	8、16		
端口 A 的控制寄存器 L4	PACRL4	16	H'FFFDD110		8、16、32		
端口 A 的控制寄存器 L3	PACRL3	16	H'FFFDD112		8、16		
端口 A 的控制寄存器 L2	PACRL2	16	H'FFFDD114		8、16、32		
端口 A 的控制寄存器 L1	PACRL1	16	H'FFFDD116		8、16		
端口 A 的端口寄存器 L	PAPRL	16	H'FFFDD11E	I/O	8、16		
端口 B 的数据寄存器 L	PBDRL	16	H'FFFDD182		8、16		
端口 B 的 IO 寄存器 L	PBIORL	16	H'FFFDD186	PFC	8、16		
端口 B 的控制寄存器 L2	PBCRL2	16	H'FFFDD194		8、16、32		
端口 B 的控制寄存器 L1	PBCRL1	16	H'FFFDD196		8、16		
端口 B 的端口寄存器 L	PBPRL	16	H'FFFDD19E	I/O	8、16		
端口 D 的数据寄存器 L	PDDRL	16	H'FFFDD282		8、16		
端口 D 的 IO 寄存器 L	PDIORL	16	H'FFFDD286	PFC	8、16		
端口 D 的控制寄存器 L3	PDCRL3	16	H'FFFDD292		8、16		
端口 D 的控制寄存器 L2	PDCRL2	16	H'FFFDD294		8、16、32		
端口 D 的控制寄存器 L1	PDCRL1	16	H'FFFDD296		8、16		
端口 D 的端口寄存器 L	PDPRL	16	H'FFFDD29E	I/O	8、16		
端口 E 的数据寄存器 H	PEDRH	16	H'FFFDD300		8、16、32		
端口 E 的数据寄存器 L	PEDRL	16	H'FFFDD302		8、16		
端口 E 的 IO 寄存器 H	PEIORH	16	H'FFFDD304	PFC	8、16、32		
端口 E 的 IO 寄存器 L	PEIORL	16	H'FFFDD306		8、16		
端口 E 的控制寄存器 H2	PECRH2	16	H'FFFDD30C		8、16、32		
端口 E 的控制寄存器 H1	PECRH1	16	H'FFFDD30E		8、16		
端口 E 的控制寄存器 L4	PECRL4	16	H'FFFDD310		8、16、32		
端口 E 的控制寄存器 L3	PECRL3	16	H'FFFDD312		8、16		
端口 E 的控制寄存器 L2	PECRL2	16	H'FFFDD314		8、16、32		

寄存器名	略称	位数	地址	模块	存取长度	存取状态数	连接的 总线宽度
端口 E 的控制寄存器 L1	PECRL1	16	H'FFFFD316	PFC	8、16	Pφ 基准 B:2、W:2、L:4	16 位
端口 E 的端口寄存器 H	PEPRH	16	H'FFFFD31C	I/O	8、16、32		
端口 E 的端口寄存器 L	PEPRL	16	H'FFFFD31E		8、16		
IRQOUT 功能控制寄存器	IFCR	16	H'FFFFD322	PFC	8、16		
端口 E 的数据寄存器	PFDR1	16	H'FFFFD382	I/O	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	A/D (通道 0)	16	Pφ 基准 B:2、W:2	16 位
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		
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		
主控控制寄存器	MCR	16	H'FFFFD800	RCAN-ET	16	Pφ 基准 B:2、W:2、L:4	16 位
通用状态寄存器	GSR	16	H'FFFFD802		16		
位结构寄存器 1	BCR1	16	H'FFFFD804		16		
位结构寄存器 0	BCR0	16	H'FFFFD806		16		
中断请求寄存器	IRR	16	H'FFFFD808		16		
中断屏蔽寄存器	IMR	16	H'FFFFD80A		16		
发送错误计数器 / 接收错误计数器	TEC/REC	16	H'FFFFD80C		16		
发送等待寄存器 1、0	TXPR1、0	32	H'FFFFD820		32		
发送取消寄存器 0	TXCR0	16	H'FFFFD82A		16		
发送应答寄存器 0	TXACK0	16	H'FFFFD832		16		
中止应答寄存器 0	ABACK0	16	H'FFFFD83A		16		

寄存器名		略称	位数	地址	模块	存取长度	存取状态数	连接的 总线宽度
数据帧接收完成寄存器 0		RXPR0	16	H'FFFFD842	RCAN-ET	16	Pφ 基准 B:2、W:2、L:4	16 位
远程帧接收完成寄存器 0		RFPR0	16	H'FFFFD84A		16		
邮箱中断屏蔽寄存器 0		MBIMR0	16	H'FFFFD852		16		
未读信息状态寄存器 0		UMSR0	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		
	MSG_DATA[3]	—	8	H'FFFFD90B		8		
	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		

寄存器名		略称	位数	地址	模块	存取长度	存取状态数	连接的 总线宽度
MB[2].	MSG_DATA[4]	—	8	H'FFFFD94C	RCAN-ET	8、16、32	Pφ 基准 B:2、W:2、L:4	16 位
	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		
MB[3].	CONTROL0H	—	16	H'FFFFD960		16、32		
	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		

寄存器名		略称	位数	地址	模块	存取长度	存取状态数	连接的 总线宽度
MB[5].	MSG_DATA[2]	—	8	H'FFFD9AA	RCAN-ET	8、16	Pφ 基准 B:2、W:2、L:4	16 位
	MSG_DATA[3]	—	8	H'FFFD9AB		8		
	MSG_DATA[4]	—	8	H'FFFD9AC		8、16、32		
	MSG_DATA[5]	—	8	H'FFFD9AD		8		
	MSG_DATA[6]	—	8	H'FFFD9AE		8、16		
	MSG_DATA[7]	—	8	H'FFFD9AF		8		
	CONTROL1H	—	8	H'FFFD9B0		8、16		
	CONTROL1L	—	8	H'FFFD9B1		8		
MB[6].	CONTROL0H	—	16	H'FFFD9C0		16、32		
	CONTROL0L	—	16	H'FFFD9C2		16		
	LAFMH	—	16	H'FFFD9C4		16、32		
	LAFML	—	16	H'FFFD9C6		16		
	MSG_DATA[0]	—	8	H'FFFD9C8		8、16、32		
	MSG_DATA[1]	—	8	H'FFFD9C9		8		
	MSG_DATA[2]	—	8	H'FFFD9CA		8、16		
	MSG_DATA[3]	—	8	H'FFFD9CB		8		
	MSG_DATA[4]	—	8	H'FFFD9CC		8、16、32		
	MSG_DATA[5]	—	8	H'FFFD9CD		8		
	MSG_DATA[6]	—	8	H'FFFD9CE		8、16		
	MSG_DATA[7]	—	8	H'FFFD9CF		8		
	CONTROL1H	—	8	H'FFFD9D0		8、16		
	CONTROL1L	—	8	H'FFFD9D1		8		
MB[7].	CONTROL0H	—	16	H'FFFD9E0		16、32		
	CONTROL0L	—	16	H'FFFD9E2		16		
	LAFMH	—	16	H'FFFD9E4		16、32		
	LAFML	—	16	H'FFFD9E6		16		
	MSG_DATA[0]	—	8	H'FFFD9E8		8、16、32		
	MSG_DATA[1]	—	8	H'FFFD9E9		8		
	MSG_DATA[2]	—	8	H'FFFD9EA		8、16		
	MSG_DATA[3]	—	8	H'FFFD9EB		8		
	MSG_DATA[4]	—	8	H'FFFD9EC		8、16、32		
	MSG_DATA[5]	—	8	H'FFFD9ED		8		
	MSG_DATA[6]	—	8	H'FFFD9EE		8、16		
	MSG_DATA[7]	—	8	H'FFFD9EF		8		
	CONTROL1H	—	8	H'FFFD9F0		8、16		
	CONTROL1L	—	8	H'FFFD9F1		8		
MB[8].	CONTROL0H	—	16	H'FFFDA00		16、32		
	CONTROL0L	—	16	H'FFFDA02		16		
	LAFMH	—	16	H'FFFDA04		16、32		
	LAFML	—	16	H'FFFDA06		16		

寄存器名		略称	位数	地址	模块	存取长度	存取状态数	连接的 总线宽度
MB[8].	MSG_DATA[0]	—	8	H'FFFFDA08	RCAN-ET	8、16、32	Pφ 基准 B:2、W:2、L:4	16 位
	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		
	MSG_DATA[3]	—	8	H'FFFFDA4B		8		
	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		

寄存器名		略称	位数	地址	模块	存取长度	存取状态数	连接的 总线宽度
MB[11].	LAFMH	—	16	H'FFFDA64	RCAN-ET	16、32	Pφ 基准 B:2、W:2、L:4	16 位
	LAFML	—	16	H'FFFDA66		16		
	MSG_DATA[0]	—	8	H'FFFDA68		8、16、32		
	MSG_DATA[1]	—	8	H'FFFDA69		8		
	MSG_DATA[2]	—	8	H'FFFDA6A		8、16		
	MSG_DATA[3]	—	8	H'FFFDA6B		8		
	MSG_DATA[4]	—	8	H'FFFDA6C		8、16、32		
	MSG_DATA[5]	—	8	H'FFFDA6D		8		
	MSG_DATA[6]	—	8	H'FFFDA6E		8、16		
	MSG_DATA[7]	—	8	H'FFFDA6F		8		
	CONTROL1H	—	8	H'FFFDA70		8、16		
	CONTROL1L	—	8	H'FFFDA71		8		
MB[12].	CONTROL0H	—	16	H'FFFDA80	RCAN-ET	16、32	Pφ 基准 B:2、W:2、L:4	16 位
	CONTROL0L	—	16	H'FFFDA82		16		
	LAFMH	—	16	H'FFFDA84		16、32		
	LAFML	—	16	H'FFFDA86		16		
	MSG_DATA[0]	—	8	H'FFFDA88		8、16、32		
	MSG_DATA[1]	—	8	H'FFFDA89		8		
	MSG_DATA[2]	—	8	H'FFFDA8A		8、16		
	MSG_DATA[3]	—	8	H'FFFDA8B		8		
	MSG_DATA[4]	—	8	H'FFFDA8C		8、16、32		
	MSG_DATA[5]	—	8	H'FFFDA8D		8		
	MSG_DATA[6]	—	8	H'FFFDA8E		8、16		
	MSG_DATA[7]	—	8	H'FFFDA8F		8		
	CONTROL1H	—	8	H'FFFDA90		8、16		
	CONTROL1L	—	8	H'FFFDA91		8		
MB[13].	CONTROL0H	—	16	H'FFFDA A0	RCAN-ET	16、32	Pφ 基准 B:2、W:2、L:4	16 位
	CONTROL0L	—	16	H'FFFDA A2		16		
	LAFMH	—	16	H'FFFDA A4		16、32		
	LAFML	—	16	H'FFFDA A6		16		
	MSG_DATA[0]	—	8	H'FFFDA A8		8、16、32		
	MSG_DATA[1]	—	8	H'FFFDA A9		8		
	MSG_DATA[2]	—	8	H'FFFDA AA		8、16		
	MSG_DATA[3]	—	8	H'FFFDA AB		8		
	MSG_DATA[4]	—	8	H'FFFDA AC		8、16、32		
	MSG_DATA[5]	—	8	H'FFFDA AD		8		
	MSG_DATA[6]	—	8	H'FFFDA AE		8、16		
	MSG_DATA[7]	—	8	H'FFFDA AF		8		
	CONTROL1H	—	8	H'FFFDA B0		8、16		
	CONTROL1L	—	8	H'FFFDA B1		8		

寄存器名		略称	位数	地址	模块	存取长度	存取状态数	连接的 总线宽度
MB[14].	CONTROL0H	—	16	H'FFFFDAC0	RCAN-ET	16、32	Pφ 基准 B:2、W:2、L:4	16 位
	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		
	MSG_DATA[3]	—	8	H'FFFFDAEB		8		
	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		
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		
看门狗定时器的计数器		WTCNT	8	H'FFFFE810	WDT *1: 读时 *2: 写时	8*1、16*2	Pφ 基准 B:2*1、W:2*2	16 位
看门狗定时器的控制 / 状态寄存器		WTCSR	8	H'FFFFE812		8*1、16*2		
振荡停止检测控制寄存器		OSCCR	8	H'FFFFE814	CPG	8	Pφ 基准 B:2	16 位

寄存器名	略称	位数	地址	模块	存取长度	存取状态数	连接的 总线宽度
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
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		

25.2 寄存器位一览

内部外围模块寄存器的地址和位名如下所示。

16 位和 32 位寄存器分别以 8 位分 2 行和 4 行表示。

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块	
SCSMR_0	C/ $\overline{A}$	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]					
TIORL_4	IOD[3:0]				IOC[3:0]					
TIER_3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA		
TIER_4	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA		
TOER	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B		
TGCR	—	BDC	N	P	FB	WF	VF	UF		

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
TOCR1	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP	MTU2
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									
TSR_3	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TSR_4	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TITCR	T3AEN	3ACOR[2:0]			T4VEN	4VCOR[2:0]			
TITCNT	—	3ACNT[2:0]			—	4VCNT[2:0]			
TBTER	—	—	—	—	—	—	BTE[1:0]		
TDER	—	—	—	—	—	—	—	TDER	
TOLBR	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P	
TBTM_3	—	—	—	—	—	—	TTSB	TTSA	
TBTM_4	—	—	—	—	—	—	TTSB	TTSA	

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
TADCR	BF[1:0]		—	—	—	—	—	—	MTU2
	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									
TGRB_0									
TGRC_0									
TGRD_0									
TGRE_0									
TGRF_0									
TIER2_0	TTGE2	—	—	—	—	—	TGIEF	TGIEE	
TSR2_0	—	—	—	—	—	—	TGFF	TGFE	
TBTM_0	—	—	—	—	—	TTSE	TTSB	TTSA	
TCR_1	—	CCLR[1:0]		CKEG[1:0]		TPSC[2:0]			
TMDR_1	—	—	—	—	MD[3:0]				

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
TIOR_1	IOB[3:0]				IOA[3:0]				MTU2
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									
TCNTU_5									
TGRU_5									
TCRU_5	—	—	—	—	—	—	TPSC[1:0]		
TIORU_5	—	—	—	IOC[4:0]					
TCNTV_5									
TGRV_5									
TCRV_5	—	—	—	—	—	—	TPSC[1:0]		
TIORV_5	—	—	—	IOC[4:0]					
TCNTW_5									
TGRW_5									
TCRW_5	—	—	—	—	—	—	TPSC[1:0]		
TIORW_5	—	—	—	IOC[4:0]					
TSR_5	—	—	—	—	—	CMFU5	CMFV5	CMFW5	

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
TIER_5	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W	MTU2
TSTR_5	—	—	—	—	—	CSTU5	CSTV5	CSTW5	
TCNTCMPCLR	—	—	—	—	—	CMPCLR5U	CMPCLR5V	CMPCLR5W	
TCR_3S	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]			MTU2S
TCR_4S	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]			
TMDR_3S	—	—	BFB	BFA	MD[3:0]				
TMDR_4S	—	—	BFB	BFA	MD[3:0]				
TIORH_3S	IOB[3:0]				IOA[3:0]				
TIORL_3S	IOD[3:0]				IOC[3:0]				
TIORH_4S	IOB[3:0]				IOA[3:0]				
TIORL_4S	IOD[3:0]				IOC[3:0]				
TIER_3S	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TIER_4S	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TOERS	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B	
TGCRS	—	BDC	N	P	FB	WF	VF	UF	
TOCR1S	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP	
TOCR2S	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P	
TCNT_3S									
TCNT_4S									
TCDRS									
TDDRS									
TGRA_3S									
TGRB_3S									
TGRA_4S									
TGRB_4S									
TCNTSS									
TCBRS									
TGRC_3S									

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
TGRD_3S									MTU2S
TGRC_4S									
TGRD_4S									
TSR_3S	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TSR_4S	TCFD	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TITCRS	T3AEN	3ACOR[2:0]			T4VEN	4VCOR[2:0]			
TITCNTS	—	3ACNT[2:0]			—	4VCNT[2:0]			
TBTERS	—	—	—	—	—	—	BTE[1:0]		
TDERS	—	—	—	—	—	—	—	TDER	
TOLBRS	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P	
TBTM_3S	—	—	—	—	—	—	TTSB	TTSA	
TBTM_4S	—	—	—	—	—	—	TTSB	TTSA	
TADCRS	BF[1:0]		—	—	—	—	—	—	
	UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE	
TADCORA_4S									
TADCORB_4S									
TADCOBRA_4S									
TADCOBRB_4S									
TSYCRS	CE0A	CE0B	CE0C	CE0D	CE1A	CE1B	CE2A	CE2B	
TWCRS	CCE	—	—	—	—	—	SCC	WRE	
TSTRS	CST4	CST3	—	—	—	CST2	CST1	CST0	
TSYRS	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0	
TRWERS	—	—	—	—	—	—	—	RWE	
TCNTU_5S									
TGRU_5S									
TCRU_5S	—	—	—	—	—	—	TPSC[1:0]		
TIORU_5S	—	—	—	IOC[4:0]					
TCNTV_5S									
TGRV_5S									

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块	
TCRV_5S	—	—	—	—	—	—	TPSC[1:0]		MTU2S	
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]							DTC	
DTCERA	DTCERA15	DTCERA14	DTCERA13	DTCERA12	—	—	—	—		
	—	—	—	—	—	—	—	—		
DTCERB	DTCERB15	DTCERB14	DTCERB13	DTCERB12	DTCERB11	DTCERB10	DTCERB9	DTCERB8		
	DTCERB7	DTCERB6	DTCERB5	DTCERB4	DTCERB3	DTCERB2	DTCERB1	DTCERB0		
DTCERC	DTCERC15	DTCERC14	DTCERC13	DTCERC12	—	—	—	—		
	—	—	—	—	DTCERC3	DTCERC2	DTCERC1	DTCERC0		
DTCERD	DTCERD15	DTCERD14	DTCERD13	DTCERD12	DTCERD11	DTCERD10	DTCERD9	DTCERD8		
	DTCERD7	DTCERD6	—	—	—	DTCERD2	DTCERD1	—		
DTCERE	DTCERE15	DTCERE14	DTCERE13	DTCERE12	DTCERE11	DTCERE10	—	—		
	DTCERE7	DTCERE6	DTCERE5	DTCERE4	DTCERE3	—	—	—		
DTCCR	—	—	—	RRS	RCHNE	—	—	ERR		
DTCVBR										
					—	—	—	—		
	—	—	—	—	—	—	—	—		
ICCR1	ICE	RCVD	MST	TRS	CKS[3:0]				I ² C2	
ICCR2	BBSY	SCP	SDAO	SDAOP	SCLO	—	IICRST	—		
ICMR	MLS	WAIT	—	—	BCWP	BC[2:0]				
ICIER	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT		
ICSR	TDRE	TEND	RDRF	NACKF	STOP	AL/OVE	AAS	ADZ		
SAR	SVA[6:0]							FS		
ICDRT										

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
ICDRR									I ² C2
NF2CYC	—	—	—	—	—	—	—	NF2CYC	
SSCRH	MSS	BIDE	—	SOL	SOLP	—	CSS[1:0]		SSU
SSCRL	FCLRM	SSUMS	SRES	—	—	—	DATS[1:0]		
SSMR	MLS	CPOS	CPHS	—	—	CKS[2:0]			
SSER	TE	RE	—	—	TEIE	TIE	RIE	CEIE	
SSSR	—	ORER	—	—	TEND	TDRE	RDRF	CE	
SSCR2	—	—	—	TENDSTS	SCSATS	SSODTS	—	—	
SSTDR0									
SSTDR1									
SSTDR2									
SSTDR3									
SSRDR0									
SSRDR1									
SSRDR2									
SSRDR3									
CMSTR	—	—	—	—	—	—	—	—	CMT
	—	—	—	—	—	—	STR1	STR0	
CMCSR_0	—	—	—	—	—	—	—	—	
	CMF	CMIE	—	—	—	—	CKS[1:0]		
CMCNT_0									
CMCOR_0									
CMCSR_1	—	—	—	—	—	—	—	—	
	CMF	CMIE	—	—	—	—	CKS[1:0]		
CMCNT_1									
CMCOR_1									
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]		

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
SPOER	—	—	—	—	—	MTU2SHIZ	MTU2CH0HIZ	MTU2CH34HIZ	POE
POECR1	—	—	—	—	MTU2PE3ZE	MTU2PE2ZE	MTU2PE1ZE	MTU2PE0ZE	
POECR2	— —	MTU2P1CZE —	MTU2P2CZE —	MTU2P3CZE —	— —	MTU2SP1CZE —	MTU2SP2CZE —	MTU2SP3CZE —	
PADRL	PA15DR PA7DR	PA14DR PA6DR	PA13DR PA5DR	PA12DR PA4DR	PA11DR PA3DR	PA10DR PA2DR	PA9DR PA1DR	PA8DR PA0DR	I/O
PAIORL	PA15IOR PA7IOR	PA14IOR PA6IOR	PA13IOR PA5IOR	PA12IOR PA4IOR	PA11IOR PA3IOR	PA10IOR PA2IOR	PA9IOR PA1IOR	PA8IOR PA0IOR	
PACRL4	— —	PA15MD2 PA13MD2	PA15MD1 PA13MD1	PA15MD0 PA13MD0	— —	PA14MD2 PA12MD2	PA14MD1 PA12MD1	PA14MD0 PA12MD0	PFC
PACRL3	— —	PA11MD 2	PA11MD 1	PA11MD 0	— —	PA10MD2 PA8MD2	PA10MD1 PA8MD1	PA10MD0 PA8MD0	
PACRL2	— —	PA7MD2 PA5MD2	PA7MD1 PA5MD1	PA7MD0 PA5MD0	— —	PA6MD2 PA4MD2	PA6MD1 PA4MD1	PA6MD0 PA4MD0	
PACRL1	— —	PA3MD2 PA1MD2	PA3MD1 PA1MD1	PA3MD0 PA1MD0	— —	PA2MD2 PA0MD2	PA2MD1 PA0MD1	PA2MD0 PA0MD0	
PAPRL	PA15PR PA7PR	PA14PR PA6PR	PA13PR PA5PR	PA12PR PA4PR	PA11PR PA3PR	PA10PR PA2PR	PA9PR PA1PR	PA8PR PA0PR	I/O
PBDRL	— PB7DR	— PB6DR	— PB5DR	— PB4DR	— PB3DR	— PB2DR	— PB1DR	— PB0DR	
PBIORL	— PB7IOR	— PB6IOR	— PB5IOR	— PB4IOR	— PB3IOR	— PB2IOR	— PB1IOR	— PB0IOR	PFC
PBCRL2	— —	PB7MD2 PB5MD2	PB7MD1 PB5MD1	PB7MD0 PB5MD0	— —	PB6MD2 PB4MD2	PB6MD1 PB4MD1	PB6MD0 PB4MD0	
PBCRL1	— —	PB3MD2 PB1MD2	PB3MD1 PB1MD1	PB3MD0 PB1MD0	— —	PB2MD2 PB0MD2	PB2MD1 PB0MD1	PB2MD0 PB0MD0	
PBPRL	— PB7PR	— PB6PR	— PB5PR	— PB4PR	— PB3PR	— PB2PR	— PB1PR	— PB0PR	
PDDRRL	— PD7DR	— PD6DR	— PD5DR	— PD4DR	— PD3DR	PD10DR PD2DR	PD9DR PD1DR	PD8DR PD0DR	I/O
PDIORL	— PD7IOR	— PD6IOR	— PD5IOR	— PD4IOR	— PD3IOR	PD10IOR PD2IOR	PD9IOR PD1IOR	PD8IOR PD0IOR	
PDCRL3	— —	— PD9MD2	— PD9MD1	— PD9MD0	— —	PD10MD2 PD8MD2	PD10MD1 PD8MD1	PD10MD0 PD8MD0	PFC
PDCRL2	— —	PD7MD2 PD5MD2	PD7MD1 PD5MD1	PD7MD0 PD5MD0	— —	PD6MD2 PD4MD2	PD6MD1 PD4MD1	PD6MD0 PD4MD0	
PDCRL1	— —	PD3MD2 PD1MD2	PD3MD1 PD1MD1	PD3MD0 PD1MD0	— —	PD2MD2 PD0MD2	PD2MD1 PD0MD1	PD2MD0 PD0MD0	
PDPRL	— PD7PR	— PD6PR	— PD5PR	— PD4PR	— PD3PR	PD10PR PD2PR	PD9PR PD1PR	PD8PR PD0PR	I/O

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
PEDRH	—	—	—	—	—	—	—	—	I/O
	—	—	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	
PECRL2	—	PE7MD2	PE7MD1	PE7MD0	—	PE6MD2	PE6MD1	PE6MD0	
	—	PE5MD2	PE5MD1	PE5MD0	—	PE4MD2	PE4MD1	PE4MD0	
PECRL1	—	PE3MD2	PE3MD1	PE3MD0	—	PE2MD2	PE2MD1	PE2MD0	
	—	PE1MD2	PE1MD1	PE1MD0	—	—	PE0MD1	PE0MD0	
PEPRH	—	—	—	—	—	—	—	—	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	
IFCR	—	—	—	—	—	—	—	—	PFC
	—	—	—	—	—	—	IRQMD1	IRQMD0	
PFDRL	PE15DR	PE14DR	PE13R	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR	I/O
	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR	
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]								

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
ADDR4	—	—	—	—	ADD[11:8]				A/D (通道 0)
	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]								
ADDR9	—	—	—	—	ADD[11:8]				
	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	MCR15	MCR14	—	—	—	TST[2:0]			RCAN-ET
	MCR7	MCR6	MCR5	—	—	MCR2	MCR1	MCR0	
GSR	—	—	—	—	—	—	—	—	
	—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0	
BCR1	TSG1[3:0]				—	TSG2[2:0]			
	—	—	SJW[1:0]		—	—	—	BSP	
BCR0	—	—	—	—	—	—	—	—	
	BRP[7:0]								
IRR	—	—	IRR13	IRR12	—	—	IRR9	IRR8	
	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0	
IMR	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8	
	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0	

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
TEC/REC	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	RCAN-ET
	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0	
TXPR1、0	TXPR1[15:8]								
	TXPR1[7:0]								
	TXPR0[15:8]								
	TXPR0[7:1]							—	
TXCR0	TXCR0[15:8]								
	TXCR0[7:1]							—	
TXACK0	TXACK0[15:8]								
	TXACK0[7:1]							—	
ABACK0	ABACK0[15:8]								
	ABACK0[7:1]							—	
RXPR0	RXPR0[15:8]								
	RXPR0[7:0]								
RFPR0	RFPR0[15:8]								
	RFPR0[7:0]								
MBIMR0	MBIMR0[15:8]								
	MBIMR0[7:0]								
UMSR0	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
	EXTID[7:0]								
MB[0]. LAFMH	IDE_LAFM	—	—	STDID_LAFM[10:6]					RCAN-ET (MCR15=1)
	STDID_LAFM[5:0]						EXTID_LAFM[17:16]		
MB[0]. LAFMH	—	STDID_LAFM[10:4]							RCAN-ET (MCR15=0)
	STDID_LAFM[3:0]				—	IDE_LAFM	EXTID_LAFM[17:16]		
MB[0]. LAFML	EXTID_LAFM[15:8]								RCAN-ET
	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								

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
MB[0]. MSG_DATA[4]	MSG_DATA_4								RCAN-ET
MB[0]. MSG_DATA[5]	MSG_DATA_5								
MB[0]. MSG_DATA[6]	MSG_DATA_6								
MB[0]. MSG_DATA[7]	MSG_DATA_7								
MB[0]. CONTROL1H	—	—	NMC	—	—	MBC[2:0]			
MB[0]. CONTROL1L	—	—	—	—	DLC[3:0]				
MB[1]	MB[0] 相同的位结构								
MB[2]	MB[0] 相同的位结构								
MB[3]	MB[0] 相同的位结构								
↓	(重复)								
MB[13]	MB[0] 相同的位结构								
MB[14]	MB[0] 相同的位结构								
MB[15]	MB[0] 相同的位结构								
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	MSTP15	—	MSTP13	MSTP12	MSTP11	MSTP10	—	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	CSSTP1	—	CSSTP2	DTBST	DTSA	CSSTP3	DTPR	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	

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
IPRD	MTU2_0	MTU2_0	MTU2_0	MTU2_0	MTU2_0	MTU2_0	MTU2_0	MTU2_0	INTC
	MTU2_1	MTU2_1	MTU2_1	MTU2_1	MTU2_1	MTU2_1	MTU2_1	MTU2_1	
IPRE	MTU2_2	MTU2_2	MTU2_2	MTU2_2	MTU2_2	MTU2_2	MTU2_2	MTU2_2	
	MTU2_3	MTU2_3	MTU2_3	MTU2_3	MTU2_3	MTU2_3	MTU2_3	MTU2_3	
IPRF	MTU2_4	MTU2_4	MTU2_4	MTU2_4	MTU2_4	MTU2_4	MTU2_4	MTU2_4	
	MTU2_5	MTU2_5	MTU2_5	MTU2_5	POE(MTU2)	POE(MTU2)	POE(MTU2)	POE(MTU2)	
IPRH	—	—	—	—	I ² C2	I ² C2	I ² C2	I ² C2	
	MTU2S_3	MTU2S_3	MTU2S_3	MTU2S_3	MTU2S_3	MTU2S_3	MTU2S_3	MTU2S_3	
IPRI	MTU2S_4	MTU2S_4	MTU2S_4	MTU2S_4	MTU2S_4	MTU2S_4	MTU2S_4	MTU2S_4	
	MTU2S_5	MTU2S_5	MTU2S_5	MTU2S_5	POE(MTU2S)	POE(MTU2S)	POE(MTU2S)	POE(MTU2S)	
IPRJ	CMT_0	CMT_0	CMT_0	CMT_0	CMT_1	CMT_1	CMT_1	CMT_1	
	—	—	—	—	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	I ² C2	I ² C2	I ² C2	I ² C2	
	RCAN-ET_0	RCAN-ET_0	RCAN-ET_0	RCAN-ET_0	—	—	—	—	
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]		
CS1WCR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	WW[2:0]			
	—	—	SW[1:0]			WR[3:1]			
	WR[0]	WM	—	—	—	—	HW[1:0]		
RAMER	—	—	—	—	—	—	—	—	FLASH
	—	—	—	—	RAMS	RAM[2:0]			

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
BARA	BAA31	BAA30	BAA29	BAA28	BAA27	BAA26	BAA25	BAA24	UBC
	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16	
	BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8	
	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0	
BAMRA	BAMA31	BAMA30	BAMA29	BAMA28	BAMA27	BAMA26	BAMA25	BAMA24	
	BAMA23	BAMA22	BAMA21	BAMA20	BAMA19	BAMA18	BAMA17	BAMA16	
	BAMA15	BAMA14	BAMA13	BAMA12	BAMA11	BAMA10	BAMA9	BAMA8	
	BAMA7	BAMA6	BAMA5	BAMA4	BAMA3	BAMA2	BAMA1	BAMA0	
BBRA	—	—	—	—	—	CPA[2:0]			
	CDA[1:0]		IDA[1:0]		RWA[1:0]		SZA[1:0]		
BDRA	BDA31	BDA30	BDA29	BDA28	BDA27	BDA26	BDA25	BDA24	
	BDA23	BDA22	BDA21	BDA20	BDA19	BDA18	BDA17	BDA16	
	BDA15	BDA14	BDA13	BDA12	BDA11	BDA10	BDA9	BDA8	
	BDA7	BDA6	BDA5	BDA4	BDA3	BDA2	BDA1	BDA0	
BDMRA	BDMA31	BDMA30	BDMA29	BDMA28	BDMA27	BDMA26	BDMA25	BDMA24	
	BDMA23	BDMA22	BDMA21	BDMA20	BDMA19	BDMA18	BDMA17	BDMA16	
	BDMA15	BDMA14	BDMA13	BDMA12	BDMA11	BDMA10	BDMA9	BDMA8	
	BDMA7	BDMA6	BDMA5	BDMA4	BDMA3	BDMA2	BDMA1	BDMA0	
BARB	BAB31	BAB30	BAB29	BAB28	BAB27	BAB26	BAB25	BAB24	
	BAB23	BAB22	BAB21	BAB20	BAB19	BAB18	BAB17	BAB16	
	BAB15	BAB14	BAB13	BAB12	BAB11	BAB10	BAB9	BAB8	
	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1	BAB0	
BAMRB	BAMB31	BAMB30	BAMB29	BAMB28	BAMB27	BAMB26	BAMB25	BAMB24	
	BAMB23	BAMB22	BAMB21	BAMB20	BAMB19	BAMB18	BAMB17	BAMB16	
	BAMB15	BAMB14	BAMB13	BAMB12	BAMB11	BAMB10	BAMB9	BAMB8	
	BAMB7	BAMB6	BAMB5	BAMB4	BAMB3	BAMB2	BAMB1	BAMB0	
BBRB	—	—	—	—	—	CPB[2:0]			
BBRB	CDB[1:0]		IDB[1:0]		RWB[1:0]		SZB[1:0]		
BDRB	BDB31	BDB30	BDB29	BDB28	BDB27	BDB26	BDB25	BDB24	
	BDB23	BDB22	BDB21	BDB20	BDB19	BDB18	BDB17	BDB16	
	BDB15	BDB14	BDB13	BDB12	BDB11	BDB10	BDB9	BDB8	
	BDB7	BDB6	BDB5	BDB4	BDB3	BDB2	BDB1	BDB0	
BDMRB	BDMB31	BDMB30	BDMB29	BDMB28	BDMB27	BDMB26	BDMB25	BDMB24	
	BDMB23	BDMB22	BDMB21	BDMB20	BDMB19	BDMB18	BDMB17	BDMB16	
	BDMB15	BDMB14	BDMB13	BDMB12	BDMB11	BDMB10	BDMB9	BDMB8	
	BDMB7	BDMB6	BDMB5	BDMB4	BDMB3	BDMB2	BDMB1	BDMB0	
BRCCR	—	—	—	—	—	—	—	—	
	—	—	UTRGW[1:0]		UBIDB	—	UBIDA	—	
	SCMFCFA	SCMFCB	SCMFDA	SCMFDB	PCTE	PCBA	—	—	
	DBEA	PCBB	DBEB	—	SEQ	—	—	ETBE	

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块
BRSR	SVF	—	—	—	BSA27	BSA26	BSA25	BSA24	UBC
	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]								

25.3 各运行模式的寄存器状态

寄存器略称	上电复位	手动复位	软件待机	深度软件待机	模块待机	睡眠	模块
SCSMR_0	初始化	保持	初始化	初始化	初始化	保持	SCI (通道 0)
SCBRR_0	初始化	保持	初始化	初始化	初始化	保持	
SCSCR_0	初始化	保持	初始化	初始化	初始化	保持	
SCTDR_0	初始化	保持	初始化	初始化	初始化	保持	
SCSSR_0	初始化	保持	初始化	初始化	初始化	保持	
SCRDR_0	初始化	保持	初始化	初始化	初始化	保持	
SCSDCR_0	初始化	保持	初始化	初始化	初始化	保持	
SCSPTR_0	初始化	保持	初始化	初始化	初始化	保持	
SCSMR_1	初始化	保持	初始化	初始化	初始化	保持	SCI (通道 1)
SCBRR_1	初始化	保持	初始化	初始化	初始化	保持	
SCSCR_1	初始化	保持	初始化	初始化	初始化	保持	
SCTDR_1	初始化	保持	初始化	初始化	初始化	保持	
SCSSR_1	初始化	保持	初始化	初始化	初始化	保持	
SCRDR_1	初始化	保持	初始化	初始化	初始化	保持	
SCSDCR_1	初始化	保持	初始化	初始化	初始化	保持	
SCSPTR_1	初始化	保持	初始化	初始化	初始化	保持	
SCSMR_2	初始化	保持	初始化	初始化	初始化	保持	SCI (通道 2)
SCBRR_2	初始化	保持	初始化	初始化	初始化	保持	
SCSCR_2	初始化	保持	初始化	初始化	初始化	保持	
SCTDR_2	初始化	保持	初始化	初始化	初始化	保持	
SCSSR_2	初始化	保持	初始化	初始化	初始化	保持	
SCRDR_2	初始化	保持	初始化	初始化	初始化	保持	
SCSDCR_2	初始化	保持	初始化	初始化	初始化	保持	
SCSPTR_2	初始化	保持	初始化	初始化	初始化	保持	
TCR_3	初始化	保持	初始化	初始化	初始化	保持	MTU2
TCR_4	初始化	保持	初始化	初始化	初始化	保持	
TMDR_3	初始化	保持	初始化	初始化	初始化	保持	
TMDR_4	初始化	保持	初始化	初始化	初始化	保持	
TIORH_3	初始化	保持	初始化	初始化	初始化	保持	
TIORL_3	初始化	保持	初始化	初始化	初始化	保持	
TIORH_4	初始化	保持	初始化	初始化	初始化	保持	
TIORL_4	初始化	保持	初始化	初始化	初始化	保持	
TIER_3	初始化	保持	初始化	初始化	初始化	保持	
TIER_4	初始化	保持	初始化	初始化	初始化	保持	
TOER	初始化	保持	初始化	初始化	初始化	保持	
TGCR	初始化	保持	初始化	初始化	初始化	保持	
TOCR1	初始化	保持	初始化	初始化	初始化	保持	
TOCR2	初始化	保持	初始化	初始化	初始化	保持	
TCNT_3	初始化	保持	初始化	初始化	初始化	保持	

寄存器略称	上电复位	手动复位	软件待机	深度软件待机	模块待机	睡眠	模块
TCNT_4	初始化	保持	初始化	初始化	初始化	保持	MTU2
TCDR	初始化	保持	初始化	初始化	初始化	保持	
TDDR	初始化	保持	初始化	初始化	初始化	保持	
TGRA_3	初始化	保持	初始化	初始化	初始化	保持	
TGRB_3	初始化	保持	初始化	初始化	初始化	保持	
TGRA_4	初始化	保持	初始化	初始化	初始化	保持	
TGRB_4	初始化	保持	初始化	初始化	初始化	保持	
TCNTS	初始化	保持	初始化	初始化	初始化	保持	
TCBR	初始化	保持	初始化	初始化	初始化	保持	
TGRC_3	初始化	保持	初始化	初始化	初始化	保持	
TGRD_3	初始化	保持	初始化	初始化	初始化	保持	
TGRC_4	初始化	保持	初始化	初始化	初始化	保持	
TGRD_4	初始化	保持	初始化	初始化	初始化	保持	
TSR_3	初始化	保持	初始化	初始化	初始化	保持	
TSR_4	初始化	保持	初始化	初始化	初始化	保持	
TITCR	初始化	保持	初始化	初始化	初始化	保持	
TITCNT	初始化	保持	初始化	初始化	初始化	保持	
TBTER	初始化	保持	初始化	初始化	初始化	保持	
TDER	初始化	保持	初始化	初始化	初始化	保持	
TOLBR	初始化	保持	初始化	初始化	初始化	保持	
TBTM_3	初始化	保持	初始化	初始化	初始化	保持	
TBTM_4	初始化	保持	初始化	初始化	初始化	保持	
TADCR	初始化	保持	初始化	初始化	初始化	保持	
TADCORA_4	初始化	保持	初始化	初始化	初始化	保持	
TADCORB_4	初始化	保持	初始化	初始化	初始化	保持	
TADCOBRA_4	初始化	保持	初始化	初始化	初始化	保持	
TADCOBRB_4	初始化	保持	初始化	初始化	初始化	保持	
TWCR	初始化	保持	初始化	初始化	初始化	保持	
TSTR	初始化	保持	初始化	初始化	初始化	保持	
TSYR	初始化	保持	初始化	初始化	初始化	保持	
TCSYSTR	初始化	保持	初始化	初始化	初始化	保持	
TRWER	初始化	保持	初始化	初始化	初始化	保持	
TCR_0	初始化	保持	初始化	初始化	初始化	保持	
TMDR_0	初始化	保持	初始化	初始化	初始化	保持	
TIORH_0	初始化	保持	初始化	初始化	初始化	保持	
TIORL_0	初始化	保持	初始化	初始化	初始化	保持	
TIER_0	初始化	保持	初始化	初始化	初始化	保持	
TSR_0	初始化	保持	初始化	初始化	初始化	保持	
TCNT_0	初始化	保持	初始化	初始化	初始化	保持	
TGRA_0	初始化	保持	初始化	初始化	初始化	保持	
TGRB_0	初始化	保持	初始化	初始化	初始化	保持	

寄存器略称	上电复位	手动复位	软件待机	深度软件待机	模块待机	睡眠	模块
TGRC_0	初始化	保持	初始化	初始化	初始化	保持	MTU2
TGRD_0	初始化	保持	初始化	初始化	初始化	保持	
TGRE_0	初始化	保持	初始化	初始化	初始化	保持	
TGRF_0	初始化	保持	初始化	初始化	初始化	保持	
TIER2_0	初始化	保持	初始化	初始化	初始化	保持	
TSR2_0	初始化	保持	初始化	初始化	初始化	保持	
TBTM_0	初始化	保持	初始化	初始化	初始化	保持	
TCR_1	初始化	保持	初始化	初始化	初始化	保持	
TMDR_1	初始化	保持	初始化	初始化	初始化	保持	
TIOR_1	初始化	保持	初始化	初始化	初始化	保持	
TIER_1	初始化	保持	初始化	初始化	初始化	保持	
TSR_1	初始化	保持	初始化	初始化	初始化	保持	
TCNT_1	初始化	保持	初始化	初始化	初始化	保持	
TGRA_1	初始化	保持	初始化	初始化	初始化	保持	
TGRB_1	初始化	保持	初始化	初始化	初始化	保持	
TICCR	初始化	保持	初始化	初始化	初始化	保持	
TCR_2	初始化	保持	初始化	初始化	初始化	保持	
TMDR_2	初始化	保持	初始化	初始化	初始化	保持	
TIOR_2	初始化	保持	初始化	初始化	初始化	保持	
TIER_2	初始化	保持	初始化	初始化	初始化	保持	
TSR_2	初始化	保持	初始化	初始化	初始化	保持	
TCNT_2	初始化	保持	初始化	初始化	初始化	保持	
TGRA_2	初始化	保持	初始化	初始化	初始化	保持	
TGRB_2	初始化	保持	初始化	初始化	初始化	保持	
TCNTU_5	初始化	保持	初始化	初始化	初始化	保持	
TGRU_5	初始化	保持	初始化	初始化	初始化	保持	
TCRU_5	初始化	保持	初始化	初始化	初始化	保持	
TIORU_5	初始化	保持	初始化	初始化	初始化	保持	
TCNTV_5	初始化	保持	初始化	初始化	初始化	保持	
TGRV_5	初始化	保持	初始化	初始化	初始化	保持	
TCRV_5	初始化	保持	初始化	初始化	初始化	保持	
TIORV_5	初始化	保持	初始化	初始化	初始化	保持	
TCNTW_5	初始化	保持	初始化	初始化	初始化	保持	
TGRW_5	初始化	保持	初始化	初始化	初始化	保持	
TCRW_5	初始化	保持	初始化	初始化	初始化	保持	
TIORW_5	初始化	保持	初始化	初始化	初始化	保持	
TSR_5	初始化	保持	初始化	初始化	初始化	保持	
TIER_5	初始化	保持	初始化	初始化	初始化	保持	
TSTR5	初始化	保持	初始化	初始化	初始化	保持	
TCNTCMPCLR	初始化	保持	初始化	初始化	初始化	保持	
TCR_3S	初始化	保持	初始化	初始化	初始化	保持	

寄存器略称	上电复位	手动复位	软件待机	深度软件待机	模块待机	睡眠	模块
TCR_4S	初始化	保持	初始化	初始化	初始化	保持	MTU2S
TMDR_3S	初始化	保持	初始化	初始化	初始化	保持	
TMDR_4S	初始化	保持	初始化	初始化	初始化	保持	
TIORH_3S	初始化	保持	初始化	初始化	初始化	保持	
TIORL_3S	初始化	保持	初始化	初始化	初始化	保持	
TIORH_4S	初始化	保持	初始化	初始化	初始化	保持	
TIORL_4S	初始化	保持	初始化	初始化	初始化	保持	
TIER_3S	初始化	保持	初始化	初始化	初始化	保持	
TIER_4S	初始化	保持	初始化	初始化	初始化	保持	
TOERS	初始化	保持	初始化	初始化	初始化	保持	
TGCRS	初始化	保持	初始化	初始化	初始化	保持	
TOCR1S	初始化	保持	初始化	初始化	初始化	保持	
TOCR2S	初始化	保持	初始化	初始化	初始化	保持	
TCNT_3S	初始化	保持	初始化	初始化	初始化	保持	
TCNT_4S	初始化	保持	初始化	初始化	初始化	保持	
TCDRS	初始化	保持	初始化	初始化	初始化	保持	
TDDRS	初始化	保持	初始化	初始化	初始化	保持	
TGRA_3S	初始化	保持	初始化	初始化	初始化	保持	
TGRB_3S	初始化	保持	初始化	初始化	初始化	保持	
TGRA_4S	初始化	保持	初始化	初始化	初始化	保持	
TGRB_4S	初始化	保持	初始化	初始化	初始化	保持	
TCNTSS	初始化	保持	初始化	初始化	初始化	保持	
TCBRS	初始化	保持	初始化	初始化	初始化	保持	
TGRC_3S	初始化	保持	初始化	初始化	初始化	保持	
TGRD_3S	初始化	保持	初始化	初始化	初始化	保持	
TGRC_4S	初始化	保持	初始化	初始化	初始化	保持	
TGRD_4S	初始化	保持	初始化	初始化	初始化	保持	
TSR_3S	初始化	保持	初始化	初始化	初始化	保持	
TSR_4S	初始化	保持	初始化	初始化	初始化	保持	
TITCRS	初始化	保持	初始化	初始化	初始化	保持	
TITCNTS	初始化	保持	初始化	初始化	初始化	保持	
TBTERS	初始化	保持	初始化	初始化	初始化	保持	
TDERS	初始化	保持	初始化	初始化	初始化	保持	
TOLBRS	初始化	保持	初始化	初始化	初始化	保持	
TBTM_3S	初始化	保持	初始化	初始化	初始化	保持	
TBTM_4S	初始化	保持	初始化	初始化	初始化	保持	
TADCRS	初始化	保持	初始化	初始化	初始化	保持	
TADCORA_4S	初始化	保持	初始化	初始化	初始化	保持	
TADCORB_4S	初始化	保持	初始化	初始化	初始化	保持	
TADCOBRA_4S	初始化	保持	初始化	初始化	初始化	保持	
TADCOBRB_4S	初始化	保持	初始化	初始化	初始化	保持	

寄存器略称	上电复位	手动复位	软件待机	深度软件待机	模块待机	睡眠	模块
TSYCRS	初始化	保持	初始化	初始化	初始化	保持	MTU2S
TWCRS	初始化	保持	初始化	初始化	初始化	保持	
TSTRS	初始化	保持	初始化	初始化	初始化	保持	
TSYRS	初始化	保持	初始化	初始化	初始化	保持	
TRWERS	初始化	保持	初始化	初始化	初始化	保持	
TCNTU_5S	初始化	保持	初始化	初始化	初始化	保持	
TGRU_5S	初始化	保持	初始化	初始化	初始化	保持	
TCRU_5S	初始化	保持	初始化	初始化	初始化	保持	
TIORU_5S	初始化	保持	初始化	初始化	初始化	保持	
TCNTV_5S	初始化	保持	初始化	初始化	初始化	保持	
TGRV_5S	初始化	保持	初始化	初始化	初始化	保持	
TCRV_5S	初始化	保持	初始化	初始化	初始化	保持	
TIORV_5S	初始化	保持	初始化	初始化	初始化	保持	
TCNTW_5S	初始化	保持	初始化	初始化	初始化	保持	
TGRW_5S	初始化	保持	初始化	初始化	初始化	保持	
TCRW_5S	初始化	保持	初始化	初始化	初始化	保持	
TIORW_5S	初始化	保持	初始化	初始化	初始化	保持	
TSR_5S	初始化	保持	初始化	初始化	初始化	保持	
TIER_5S	初始化	保持	初始化	初始化	初始化	保持	
TSTR_5S	初始化	保持	初始化	初始化	初始化	保持	
TCNTCMPCLRS	初始化	保持	初始化	初始化	初始化	保持	FLASH
FCCS	初始化	保持	初始化	初始化	初始化	保持	
FPCS	初始化	保持	初始化	初始化	初始化	保持	
FECS	初始化	保持	初始化	初始化	初始化	保持	
FKEY	初始化	保持	初始化	初始化	初始化	保持	
FMATS	初始化	保持	初始化	初始化	初始化	保持	
FTDAR	初始化	保持	初始化	初始化	初始化	保持	
DTCERA	初始化	保持	保持	初始化	保持	保持	DTC
DTCERB	初始化	保持	保持	初始化	保持	保持	
DTCERC	初始化	保持	保持	初始化	保持	保持	
DTCERD	初始化	保持	保持	初始化	保持	保持	
DTCERE	初始化	保持	保持	初始化	保持	保持	
DTCCR	初始化	保持	保持	初始化	保持	保持	
DTCVBR	初始化	保持	保持	初始化	保持	保持	
ICCR1	初始化	保持	保持	初始化	保持	保持	I ² C2
ICCR2	初始化	保持	保持	初始化	保持	保持	
ICMR	初始化	保持	保持	初始化	保持	保持	
ICIER	初始化	保持	保持	初始化	保持	保持	
ICSR	初始化	保持	保持	初始化	保持	保持	
SAR	初始化	保持	保持	初始化	保持	保持	
ICDRT	初始化	保持	保持	初始化	保持	保持	

寄存器略称	上电复位	手动复位	软件待机	深度软件待机	模块待机	睡眠	模块
ICDRR	初始化	保持	保持	初始化	保持	保持	I ² C2
NF2CYC	初始化	保持	保持	初始化	保持	保持	
SSCRH	初始化	保持	初始化	初始化	初始化	保持	SSU
SSCRL	初始化	保持	初始化	初始化	初始化	保持	
SSMR	初始化	保持	初始化	初始化	初始化	保持	
SSER	初始化	保持	初始化	初始化	初始化	保持	
SSSR	初始化	保持	初始化	初始化	初始化	保持	
SSCR2	初始化	保持	初始化	初始化	初始化	保持	
SSTDR0	初始化	保持	初始化	初始化	初始化	保持	
SSTDR1	初始化	保持	初始化	初始化	初始化	保持	
SSTDR2	初始化	保持	初始化	初始化	初始化	保持	
SSTDR3	初始化	保持	初始化	初始化	初始化	保持	
SSRDR0	初始化	保持	初始化	初始化	初始化	保持	
SSRDR1	初始化	保持	初始化	初始化	初始化	保持	
SSRDR2	初始化	保持	初始化	初始化	初始化	保持	
SSRDR3	初始化	保持	初始化	初始化	初始化	保持	
CMSTR	初始化	保持	初始化	初始化	初始化	保持	CMT
CMCSR_0	初始化	保持	初始化	初始化	初始化	保持	
CMCNT_0	初始化	保持	初始化	初始化	初始化	保持	
CMCOR_0	初始化	保持	初始化	初始化	初始化	保持	
CMCSR_1	初始化	保持	初始化	初始化	初始化	保持	
CMCNT_1	初始化	保持	初始化	初始化	初始化	保持	
CMCOR_1	初始化	保持	初始化	初始化	初始化	保持	
ICSR1	初始化	保持	保持	初始化	—	保持	POE
OCSR1	初始化	保持	保持	初始化	—	保持	
ICSR2	初始化	保持	保持	初始化	—	保持	
OCSR2	初始化	保持	保持	初始化	—	保持	
ICSR3	初始化	保持	保持	初始化	—	保持	
SPOER	初始化	保持	保持	初始化	—	保持	
POECR1	初始化	保持	保持	初始化	—	保持	
POECR2	初始化	保持	保持	初始化	—	保持	
PADRL	初始化	保持	保持	初始化	—	保持	I/O
PAIORL	初始化	保持	保持	初始化	—	保持	PFC
PACRL4	初始化	保持	保持	初始化	—	保持	
PACRL3	初始化	保持	保持	初始化	—	保持	
PACRL2	初始化	保持	保持	初始化	—	保持	
PACRL1	初始化	保持	保持	初始化	—	保持	
PAPRL	初始化	保持	保持	初始化	—	保持	I/O
PBDRL	初始化	保持	保持	初始化	—	保持	
PBIORL	初始化	保持	保持	初始化	—	保持	PFC
PBCRL2	初始化	保持	保持	初始化	—	保持	

寄存器略称	上电复位	手动复位	软件待机	深度软件待机	模块待机	睡眠	模块
PBCRL1	初始化	保持	保持	初始化	—	保持	PFC
PBPRL	初始化	保持	保持	初始化	—	保持	I/O
PDDR1	初始化	保持	保持	初始化	—	保持	
PDIORL	初始化	保持	保持	初始化	—	保持	
PDCRL3	初始化	保持	保持	初始化	—	保持	PFC
PDCRL2	初始化	保持	保持	初始化	—	保持	
PDCRL1	初始化	保持	保持	初始化	—	保持	
PDPRL	初始化	保持	保持	初始化	—	保持	
PEDRH	初始化	保持	保持	初始化	—	保持	I/O
PEDRL	初始化	保持	保持	初始化	—	保持	
PEIORH	初始化	保持	保持	初始化	—	保持	
PEIORL	初始化	保持	保持	初始化	—	保持	PFC
PECRH2	初始化	保持	保持	初始化	—	保持	
PECRH1	初始化	保持	保持	初始化	—	保持	
PECRL4	初始化	保持	保持	初始化	—	保持	
PECRL3	初始化	保持	保持	初始化	—	保持	
PECRL2	初始化	保持	保持	初始化	—	保持	
PECRL1	初始化	保持	保持	初始化	—	保持	
PEPRH	初始化	保持	保持	初始化	—	保持	I/O
PEPRL	初始化	保持	保持	初始化	—	保持	
IFCR	初始化	保持	保持	初始化	—	保持	PFC
PFDR1	初始化	保持	保持	初始化	—	保持	I/O
ADCR_0	初始化	保持	初始化	初始化	初始化	保持	A/D (通道 0)
ADSR_0	初始化	保持	初始化	初始化	初始化	保持	
ADSTRGR_0	初始化	保持	初始化	初始化	初始化	保持	
ADANSR_0	初始化	保持	初始化	初始化	初始化	保持	
ADDR0	初始化	保持	初始化	初始化	初始化	保持	
ADDR1	初始化	保持	初始化	初始化	初始化	保持	
ADDR2	初始化	保持	初始化	初始化	初始化	保持	
ADDR3	初始化	保持	初始化	初始化	初始化	保持	
ADDR4	初始化	保持	初始化	初始化	初始化	保持	
ADDR5	初始化	保持	初始化	初始化	初始化	保持	
ADDR6	初始化	保持	初始化	初始化	初始化	保持	
ADDR7	初始化	保持	初始化	初始化	初始化	保持	
ADCR_1	初始化	保持	初始化	初始化	初始化	保持	A/D (通道 1)
ADSR_1	初始化	保持	初始化	初始化	初始化	保持	
ADSTRGR_1	初始化	保持	初始化	初始化	初始化	保持	
ADANSR_1	初始化	保持	初始化	初始化	初始化	保持	
ADDR8	初始化	保持	初始化	初始化	初始化	保持	
ADDR9	初始化	保持	初始化	初始化	初始化	保持	
ADDR10	初始化	保持	初始化	初始化	初始化	保持	

寄存器略称	上电复位	手动复位	软件待机	深度软件待机	模块待机	睡眠	模块
ADDR11	初始化	保持	初始化	初始化	初始化	保持	A/D (通道 1)
ADDR12	初始化	保持	初始化	初始化	初始化	保持	
ADDR13	初始化	保持	初始化	初始化	初始化	保持	
ADDR14	初始化	保持	初始化	初始化	初始化	保持	
ADDR15	初始化	保持	初始化	初始化	初始化	保持	
MCR	初始化	保持	初始化	初始化	初始化	保持	RCAN-ET
GSR	初始化	保持	初始化	初始化	初始化	保持	
BCR1	初始化	保持	初始化	初始化	初始化	保持	
BCR0	初始化	保持	初始化	初始化	初始化	保持	
IRR	初始化	保持	初始化	初始化	初始化	保持	
IMR	初始化	保持	初始化	初始化	初始化	保持	
TEC/REC	初始化	保持	初始化	初始化	初始化	保持	
TXPR1、0	初始化	保持	初始化	初始化	初始化	保持	
TXCR0	初始化	保持	初始化	初始化	初始化	保持	
TXACK0	初始化	保持	初始化	初始化	初始化	保持	
ABACK0	初始化	保持	初始化	初始化	初始化	保持	
RXPR0	初始化	保持	初始化	初始化	初始化	保持	
RFPR0	初始化	保持	初始化	初始化	初始化	保持	
MBIMR0	初始化	保持	初始化	初始化	初始化	保持	
UMSR0	初始化	保持	初始化	初始化	初始化	保持	
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	

寄存器略称	上电复位	手动复位	软件待机	深度软件待机	模块待机	睡眠	模块
MB[0]. MSG_DATA[7]	—	保持	—	—	—	保持	RCAN-ET
MB[0]. CONTROL1H	初始化	保持	初始化	初始化	初始化	保持	
MB[0]. CONTROL1L	初始化	保持	初始化	初始化	初始化	保持	
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						
FRQCR	初始化 *1	保持	保持	初始化	—	保持	CPG
STBCR1	初始化	保持	保持	初始化	—	保持	低功耗
STBCR2	初始化	保持	保持	初始化	—	保持	
STBCR3	初始化	保持	保持	初始化	—	保持	
STBCR4	初始化	保持	保持	初始化	—	保持	
STBCR5	初始化	保持	保持	初始化	—	保持	低功耗
STBCR6	初始化	保持	保持	初始化	—	保持	
WTCNT	初始化 *1	保持	保持	初始化	—	保持	WDT
WTCSR	初始化 *1	保持	保持	初始化	—	保持	
OSCCR	初始化 *2	保持	保持 *3	初始化	—	保持	CPG
RAMCR	初始化	保持	保持	初始化	—	保持	低功耗
BSCEHR	初始化	保持	保持	初始化	—	保持	BSC
ICR0	初始化	初始化	保持	初始化	—	保持	INTC
IRQCR	初始化	初始化	保持	初始化	—	保持	
IRQSR	初始化	初始化	保持	初始化	—	保持	
IPRA	初始化	初始化	保持	初始化	—	保持	
IPRD	初始化	初始化	保持	初始化	—	保持	
IPRE	初始化	初始化	保持	初始化	—	保持	
IPRF	初始化	初始化	保持	初始化	—	保持	
IPRH	初始化	初始化	保持	初始化	—	保持	
IPRI	初始化	初始化	保持	初始化	—	保持	
IPRJ	初始化	初始化	保持	初始化	—	保持	
IPRK	初始化	初始化	保持	初始化	—	保持	
IPRL	初始化	初始化	保持	初始化	—	保持	
IPRM	初始化	初始化	保持	初始化	—	保持	
CMNCR	初始化	保持	保持	初始化	—	保持	BSC
CS0BCR	初始化	保持	保持	初始化	—	保持	
CS1BCR	初始化	保持	保持	初始化	—	保持	
CS0WCR	初始化	保持	保持	初始化	—	保持	

寄存器略称	上电复位	手动复位	软件待机	深度软件待机	模块待机	睡眠	模块
CS1WCR	初始化	保持	保持	初始化	—	保持	BSC
RAMER	初始化	初始化	保持	初始化	保持	保持	FLASH
BARA	初始化	保持	保持	初始化	初始化	保持	UBC
BAMRA	初始化	保持	保持	初始化	初始化	保持	
BBRA	初始化	保持	保持	初始化	初始化	保持	
BDRA	初始化	保持	保持	初始化	初始化	保持	
BDMRA	初始化	保持	保持	初始化	初始化	保持	
BARB	初始化	保持	保持	初始化	初始化	保持	
BAMRB	初始化	保持	保持	初始化	初始化	保持	
BBRB	初始化	保持	保持	初始化	初始化	保持	
BDRB	初始化	保持	保持	初始化	初始化	保持	
BDMRB	初始化	保持	保持	初始化	初始化	保持	
BRCR	初始化	保持	保持	初始化	初始化	保持	
BRSR	初始化	保持	保持	初始化	初始化	保持	UBC
BRDR	初始化	保持	保持	初始化	初始化	保持	
BETR	初始化	保持	保持	初始化	初始化	保持	

【注】 *1 在通过 WDT 进行上电复位时，此寄存器不被初始化。

*2 在通过 WDT 进行上电复位时，OSCSTOP 位不被初始化。

*3 OSCSTOP 位被初始化。

第 26 章 电特性

【注】 请注意：此电特性的规格为暂定值，有可能会不预先通知而改变规格值。

26.1 绝对最大额定值

绝对最大额定值如表 26.1 所示。

表 26.1 绝对最大额定值

项 目		符号	额定值	单位
电源电压		V_{CC}	$-0.3 \sim +7.0$	V
输入电压（模拟输入引脚除外）		V_{in}	$-0.3 \sim V_{CC}+0.3$	V
模拟电源电压		AV_{CC}	$-0.3 \sim +7.0$	V
模拟基准电压		AV_{refh}	$-0.3 \sim AV_{CC}+0.3$	V
模拟输入电压		V_{an}	$-0.3 \sim AV_{CC}+0.3$	V
工作温度	民用产品	T_{opr}	$-20 \sim +85$	°C
	工业产品		$-40 \sim +85$	°C
保存温度		T_{stg}	$-55 \sim +125$	°C

【使用时的注意事项】

如果在超过绝对最大额定值的情况下使用 LSI，就会产生 LSI 的永久性破坏。

26.2 DC 特性

DC 特性如表 26.2 和表 26.3 所示。

表 26.2 DC 特性

条件: $V_{CC}=3.0V \sim 3.6V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{refh}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL V_{SS}=AV_{SS}=AV_{refl}=0V$ 、 $Ta=-20 \sim +85^{\circ}C$ (民用产品)、 $Ta=-40 \sim +85^{\circ}C$ (工业产品)

项 目	符号	Min.	Typ.	Max.	单位	测量条件
输入高电平的电压 (施密特触发输入 引脚除外)	$\overline{RES}$ 、 $\overline{MRES}$ 、 NMI 、 $\overline{FWE}$ 、 $MD1$ 、 $MD0$ 、 $\overline{ASEMD0}$ 、 $EXTAL$	$V_{CC}-0.5$	—	$V_{CC}+0.3$	V	
	模拟兼用端口	2.2	—	$AV_{CC}+0.3$	V	
	其他的输入引脚	2.2	—	$V_{CC}+0.3$	V	
输入低电平的电压 (施密特触发输入 引脚除外)	$\overline{RES}$ 、 $\overline{MRES}$ 、 NMI 、 $\overline{FWE}$ 、 $MD1$ 、 $MD0$ 、 $\overline{ASEMD0}$ 、 $EXTAL$	-0.3	—	0.5	V	
	其他的输入引脚	-0.3	—	0.8	V	
施密特触发的输入 电压	$IRQ3 \sim IRQ0$ 、 $\overline{POE8}$ 、 $\overline{POE6} \sim \overline{POE4}$ 、 $\overline{POE2} \sim \overline{POE0}$ 、 $TCLKA \sim TCLKD$ 、 $TIOC0A \sim TIOC0D$ 、 $TIOC1A$ 、 $TIOC1B$ 、 $TIOC2A$ 、 $TIOC2B$ 、 $TIOC3A \sim TIOC3D$ 、 $TIOC4A \sim TIOC4D$ 、 $TIC5U$ 、 $TIC5V$ 、 $TIC5W$ 、 $TIOC3BS$ 、 $TIOC3DS$ 、 $TIOC4AS \sim TIOC4DS$ 、 $TIC5US$ 、 $TIC5VS$ 、 $TIC5WS$ 、 $SCK0 \sim SCK2$ 、 $RXD0 \sim RXD2$ 、 $\overline{SSCK}$ 、 $\overline{SCS}$ 、 SSI 、 SSO 、 SCL 、 SDA	V_{T+}	$V_{CC}-0.5$	—	V	
		V_{T-}	—	0.5	V	
		$V_{T+}-V_{T-}$	0.2	—	V	
输入漏泄电流	所有输入引脚 ($\overline{ASEMD0}$ 除外)	$ I_{in} $	—	1.0	μA	
输入上拉 MOS 电流	$\overline{ASEMD0}$	$-I_{pu}$	—	350	μA	$V_{in}=0V$
三态漏泄电流 (OFF 状态)	端口 A、B、D、E	$ I_{tsil} $	—	1.0	μA	
输出高电平的电压	所有输出引脚	V_{OH}	$V_{CC}-0.5$	—	V	$I_{OH}=-200\mu A$
			$V_{CC}-1.0$	—	V	$I_{OH}=-1mA$
	PE9、PE11 $\sim$ PE21		$V_{CC}-1.0$	—	V	$I_{OH}=-5mA$

项 目		符号	Min.	Typ.	Max.	单位	测量条件
输出低电平电压	所有输出引脚	V _{OL}	—	—	0.4	V	I _{OL} =1.6mA
	SCL、SDA		—	—	0.4	V	I _{OL} =3mA
			—	—	0.5	V	I _{OL} =8mA
	PE9、PE11 ~ PE21		—	—	0.9	V	I _{OL} =15mA
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	正常运行	I _{CC}	—	80	105	mA	I _φ =80MHz B _φ =40MHz P _φ =40MHz MP _φ =40MHz MI _φ =80MHz
	睡眠模式		—	55	85	mA	B _φ =40MHz P _φ =40MHz MP _φ =40MHz MI _φ =80MHz
	软件待机模式		—	8	20	mA	T _a ≤ 50°C
			—	—	30	mA	50°C < T _a
	深度软件待机模式		—	2	10	μA	T _a ≤ 50°C
			—	—	40	μA	50°C < T _a
模拟电源电流	A/D 转换中	AI _{CC}	—	3	5	mA	每个 A/D 转换模 块的值
	A/D 转换待机		—	—	0.1	mA	
	待机		—	—	15	μA	
基准电源电流	A/D 转换中	AI _{refh}	—	—	2	mA	每个 A/D 转换模 块的值
	A/D 转换待机		—	—	2	mA	
	待机		—	—	2.5	μA	
RAM 待机电压		VRAM	2	—	—	V	V _{CC}

【使用时的注意事项】

- 在不使用 A/D 转换器时，不能将 AV_{CC} 、 AV_{SS} 、 AV_{refn} 和 AV_{refl} 引脚置为开路。
- 消耗电流是在 $V_{IH}(\text{Min.})=V_{CC}-0.5\text{V}$ 、 $V_{IL}(\text{Max.})=0.5\text{V}$ 的条件下所有输出引脚为无负载状态时的值。

表 26.3 DC 特性

条件: $V_{CC}=4.0V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{refH}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL V_{SS}=AV_{SS}=AV_{refL}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业产品)

项 目	符号	Min.	Typ.	Max.	单位	测量条件	
输入高电平的电压 (施密特触发输入引 脚除外)	$\overline{RES}$ 、 $\overline{MRES}$ 、 NMI 、 $\overline{FWE}$ 、 $MD1$ 、 $MD0$ 、 $\overline{ASEMD0}$ 、 $EXTAL$	$V_{CC}-0.6$	—	$V_{CC}+0.3$	V		
	模拟兼用端口	2.2	—	$AV_{CC}+0.3$	V		
	其他输入引脚	2.2	—	$V_{CC}+0.3$	V		
输入低电平的电压 (施密特触发输入引 脚除外)	$\overline{RES}$ 、 $\overline{MRES}$ 、 NMI 、 $\overline{FWE}$ 、 $MD1$ 、 $MD0$ 、 $\overline{ASEMD0}$ 、 $EXTAL$	-0.3	—	0.4	V		
	其他输入引脚	-0.3	—	0.8	V		
施密特触发的输入 电压	$IRQ3 \sim IRQ0$ 、 $\overline{POE8}$ 、 $\overline{POE6} \sim \overline{POE4}$ 、 $\overline{POE2} \sim \overline{POE0}$ 、 $TCLKA \sim TCLKD$ 、 $TIOC0A \sim TIOC0D$ 、 $TIOC1A$ 、 $TIOC1B$ 、 $TIOC2A$ 、 $TIOC2B$ 、 $TIOC3A \sim TIOC3D$ 、 $TIOC4A \sim TIOC4D$ 、 $TIC5U$ 、 $TIC5V$ 、 $TIC5W$ 、 $TIOC3BS$ 、 $TIOC3DS$ 、 $TIOC4AS \sim TIOC4DS$ 、 $TIC5US$ 、 $TIC5VS$ 、 $TIC5WS$ 、 $SCK0 \sim SCK2$ 、 $RXD0 \sim RXD2$ 、 $SSCK$ 、 $\overline{SCS}$ 、 SSI 、 SSO 、 SCL 、 SDA	$V_{T^{+}}$	$V_{CC}-0.5$	—	V		
		$V_{T^{-}}$	—	—	1.0	V	
		$V_{T^{+}}-V_{T^{-}}$	0.4	—	—	V	
输入漏泄电流	所有输入引脚 ($\overline{ASEMD0}$ 除外)	I_{inl}	—	—	1.0	μA	
输入上拉 MOS 电流	$\overline{ASEMD0}$	$-I_{pu}$	—	—	800	μA	$V_{in}=0V$
三态漏泄电流 (OFF 状态)	端口 A、B、D、E	I_{tsil}	—	—	1.0	μA	
输出高电平的电压	所有输出引脚	V_{OH}	$V_{CC}-0.5$	—	—	V	$I_{OH}=-200\mu A$
			$V_{CC}-1.0$	—	—	V	$I_{OH}=-1mA$
	PE9、PE11 $\sim$ PE21		$V_{CC}-1.0$	—	—	V	$I_{OH}=-5mA$
输出低电平的电压	所有输出引脚	V_{OL}	—	—	0.4	V	$I_{OL}=1.6mA$
	SCL、SDA		—	—	0.4	V	$I_{OL}=3mA$
			—	—	0.5	V	$I_{OL}=8mA$
	PE9、PE11 $\sim$ PE21		—	—	1.4	V	$I_{OL}=15mA$
输入电容	所有输入引脚	C_{in}	—	—	20	pF	$V_{in}=0V$ 、 $f=1MHz$ 、 $T_a=25^{\circ}C$

项 目		符号	Min.	Typ.	Max.	单位	测量条件
消耗电流	正常运行	I _{CC}	—	80	105	mA	I _φ =80MHz B _φ =40MHz P _φ =40MHz MP _φ =40MHz MI _φ =80MHz
	睡眠模式		—	55	85	mA	B _φ =40MHz P _φ =40MHz MP _φ =40MHz MI _φ =80MHz
	软件待机模式		—	8	20	mA	T _a ≤ 50°C
			—	—	30	mA	50°C < T _a
	深度软件待机模式		—	2	10	μA	T _a ≤ 50°C
			—	—	40	μA	50°C < T _a
模拟电源电流	A/D 转换中	AI _{CC}	—	3	5	mA	每个 A/D 转换模 块的值
	A/D 转换待机		—	—	0.1	mA	
	待机		—	—	15	μA	
基准电源电流	A/D 转换中	AI _{refh}	—	—	2	mA	每个 A/D 转换模 块的值
	A/D 转换待机		—	—	2	mA	
	待机		—	—	2.5	μA	
RAM 待机电压		VRAM	2	—	—	V	V _{CC}

【使用时的注意事项】

- 在不使用 A/D 转换器时，不能将 AV_{CC} 、 AV_{SS} 、 AV_{refh} 和 AV_{refl} 引脚置为开路。
- 消耗电流是在 $V_{IH}(\text{Min.})=V_{CC}-0.5\text{V}$ 、 $V_{IL}(\text{Max.})=0.5\text{V}$ 的条件下所有输出引脚为无负载状态时的值。

表 26.4 输出容许电流值

条件: $V_{CC}=3.0\text{V} \sim 3.6\text{V}$ 或者 $4.0\text{V} \sim 5.5\text{V}$ 、 $AV_{CC}=4.5\text{V} \sim 5.5\text{V}$ 、 $AV_{refh}=4.5\text{V} \sim AV_{CC}$ 、 $V_{SS}=PLL V_{SS}=AV_{SS}=AV_{refl}=0\text{V}$ 、 $T_a=-20 \sim +85^{\circ}\text{C}$ (民用产品)、 $T_a=-40 \sim +85^{\circ}\text{C}$ (工业产品)

项 目	符号	Min.	Typ.	Max.	单位
输出低电平的容许电流 (每个引脚)	I_{OL}	—	—	2.0*	mA
输出低电平的容许电流 (总和)	ΣI_{OL}	—	—	110	mA
输出高电平的容许电流 (每个引脚)	$-I_{OH}$	—	—	2.0*	mA
输出高电平的容许电流 (总和)	$\Sigma -I_{OH}$	—	—	35	mA

【使用时的注意事项】为确保 LSI 的可靠性，输出的电流值不得超过表 26.4 的值。

【注】 * PE9、PE11 ~ PE21 为 $I_{OL}=15\text{mA}(\text{Max.})$ 、 $-I_{OH}=5\text{mA}(\text{Max.})$ 。SCL、SDA 为 $I_{OL}=8\text{mA}(\text{Max.})$ 。但是，这些引脚中同时超过 2.0mA 流过 $I_{OL}/-I_{OH}$ 的引脚不能超过 6 个。

26.3 AC 特性

原则上输入到本 LSI 的信号是时钟同步输入信号。只要没有特殊理由，必须确保各输入信号的准备和保持时间。

表 26.5 最大工作频率

条件: $V_{CC}=3.0V \sim 3.6V$ 或者 $4.0V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{refh}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL V_{SS}=AV_{SS}=AV_{refl}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业产品)

项 目	符号	Min.	Typ.	Max.	单位	备注
工作频率	CPU ($I\phi$)	f	10	—	80	MHz
	外部总线 ($B\phi$)		10	—	40	
	外围模块 ($P\phi$)		10	—	40	
	MTU2 ($MP\phi$)		10	—	40	
	MTU2S ($MI\phi$)		10	—	80	

26.3.1 时钟时序

表 26.6 时钟时序

条件: $V_{CC}=3.0V \sim 3.6V$ 或者 $4.0V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{refh}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL V_{SS}=AV_{SS}=AV_{refl}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业产品)

项 目	符号	Min.	Max.	单位	参照图
EXTAL 时钟的输入频率	f_{EX}	5	12.5	MHz	图 26.1
EXTAL 时钟的输入周期时间	t_{EXcyc}	80	200	ns	
EXTAL 时钟的输入低电平脉宽	t_{EXL}	20	—	ns	
EXTAL 时钟的输入高电平脉宽	t_{EXH}	20	—	ns	
EXTAL 时钟的输入上升时间	t_{EXr}	—	5	ns	
EXTAL 时钟的输入下降时间	t_{EXf}	—	5	ns	
CK 时钟的输出频率	f_{OP}	10	40	MHz	图 26.2
CK 时钟的输出周期时间	t_{cyc}	25	100	ns	
CK 时钟的输出低电平脉宽	t_{CKL}	$1/2 t_{cyc}-7.5$	—	ns	
CK 时钟的输出高电平脉宽	t_{CKH}	$1/2 t_{cyc}-7.5$	—	ns	
CK 时钟的输出上升时间	t_{CKr}	—	5	ns	
CK 时钟的输出下降时间	t_{CKf}	—	5	ns	
上电振荡稳定时间	t_{OSC1}	10	—	ms	图 26.3
待机返回的振荡稳定时间 1	t_{OSC2}	10	—	ms	图 26.4
待机返回的振荡稳定时间 2	t_{OSC3}	10	—	ms	图 26.5

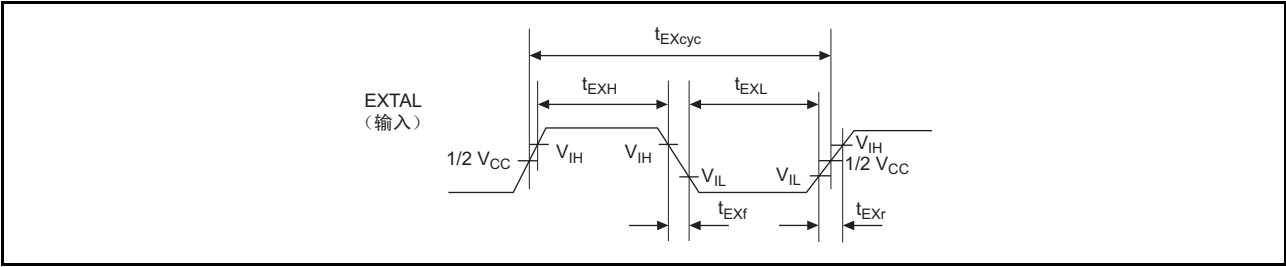


图 26.1 EXTAL 时钟的输入时序

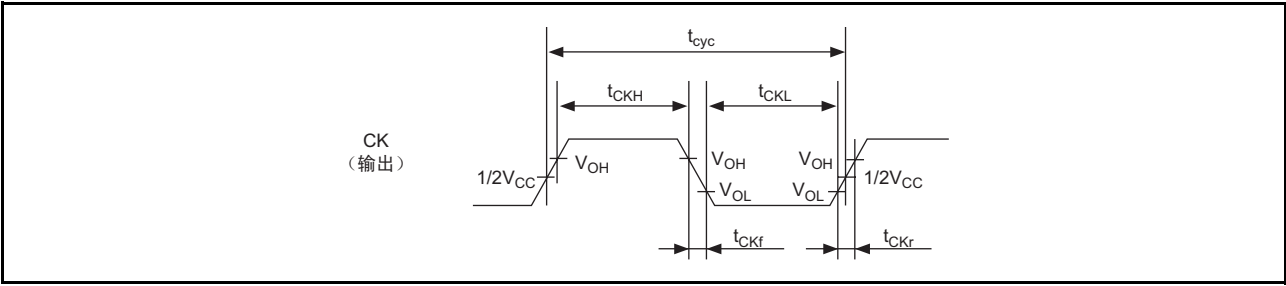


图 26.2 CK 时钟的输出时序

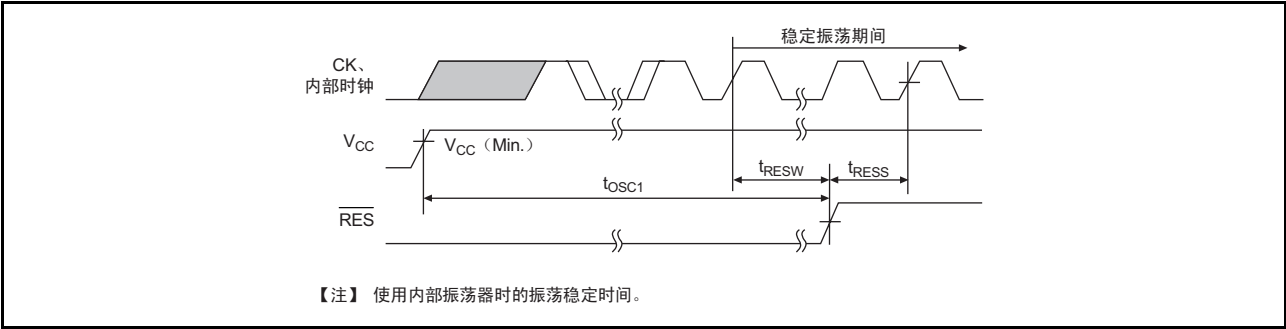


图 26.3 上电振荡稳定时间

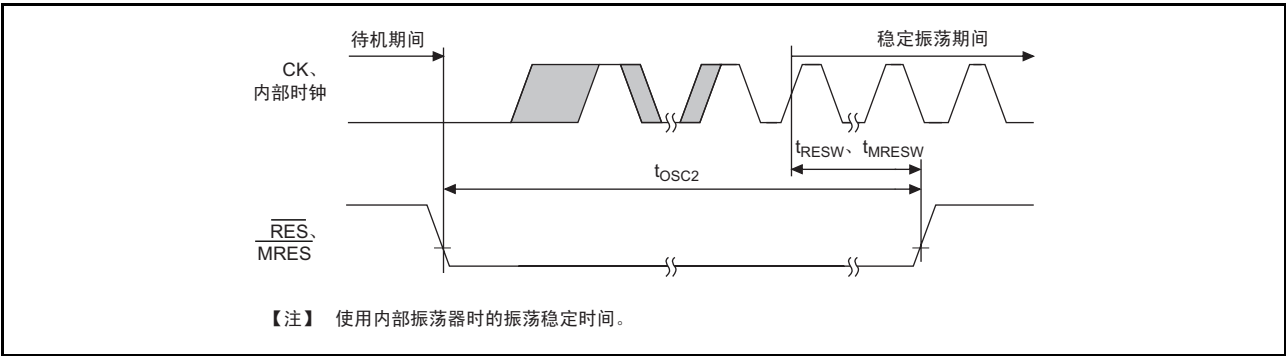


图 26.4 待机返回时的振荡稳定时间（通过复位的返回）

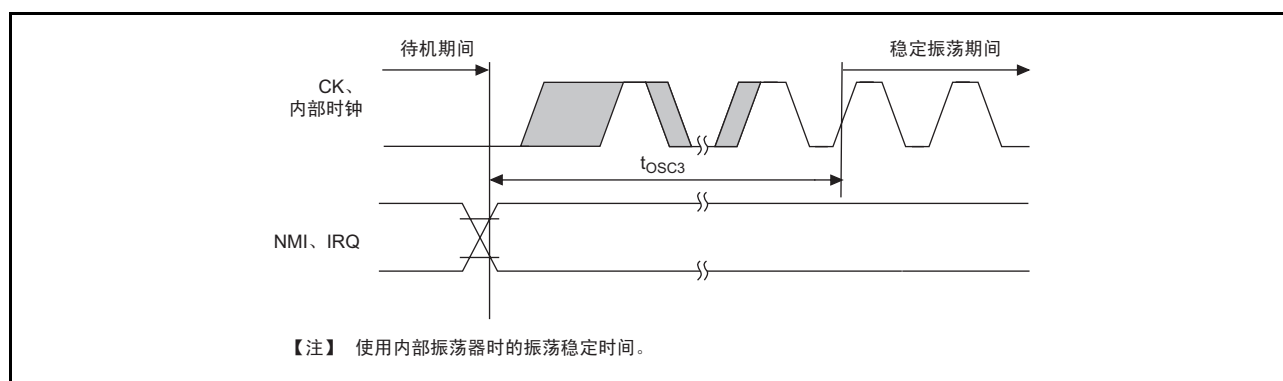


图 26.5 待机返回时的振荡稳定时间（通过 NMI 或者 IRQ 的返回）

26.3.2 控制信号的时序

表 26.7 控制信号的时序

条件: $V_{CC}=3.0V \sim 3.6V$ 或者 $4.0V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{refh}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=AV_{refl}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$ （民用产品）、 $T_a=-40 \sim +85^{\circ}C$ （工业产品）

项 目	符号	Min.	Max.	单位	参照图
RES 脉宽	t_{RESW}	20*2	—	t_{Bcyc}^{*4}	图 26.3
RES 准备时间 *1	t_{RESS}	65	—	ns	图 26.4
RES 保持时间	t_{RESH}	15	—	ns	图 26.6
MRES 脉宽	t_{MRESW}	20*3	—	t_{Bcyc}^{*4}	图 26.7
MRES 准备时间 *1	t_{MRESS}	25	—	ns	
MRES 保持时间	t_{MRESH}	15	—	ns	
MD1、MD0、FEW 准备时间	t_{MDS}	20	—	t_{Bcyc}^{*4}	图 26.6
BREQ 准备时间	t_{BREQS}	$1/2t_{Bcyc}+15$	—	ns	图 26.9
BREQ 保持时间	t_{BREQH}	$1/2t_{Bcyc}+10$	—	ns	
NMI 准备时间 *1	t_{NMIS}	60	—	ns	图 26.7
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	
IRQOUT 输出延迟时间	t_{IRQOD}	—	100	ns	图 26.8
BACK 延迟时间	t_{BACKD}	—	$1/2t_{Bcyc}+20$	ns	图 26.9
总线三态延迟时间	t_{BOFF}	0	100	ns	图 26.10
总线缓冲器 ON 时间	t_{BON}	0	100	ns	

【注】 *1 $\overline{RES}$ 、 $\overline{MRES}$ 、NMI、 $\overline{BREQ}$ 和 IRQ3 ~ IRQ0 是异步信号。如果满足在此所示的准备时间，就在时钟上升沿检测信号的变化。否则，信号变化的检测就会推迟到下一个时钟的上升沿。

*2 在待机模式时， $t_{RESW}=t_{OSC2}$ （10ms）。

*3 在待机模式时， $t_{MRESW}=t_{OSC2}$ （10ms）。

*4 t_{Bcyc} 表示外部总线时钟（ $B\phi=CK$ ）的周期。

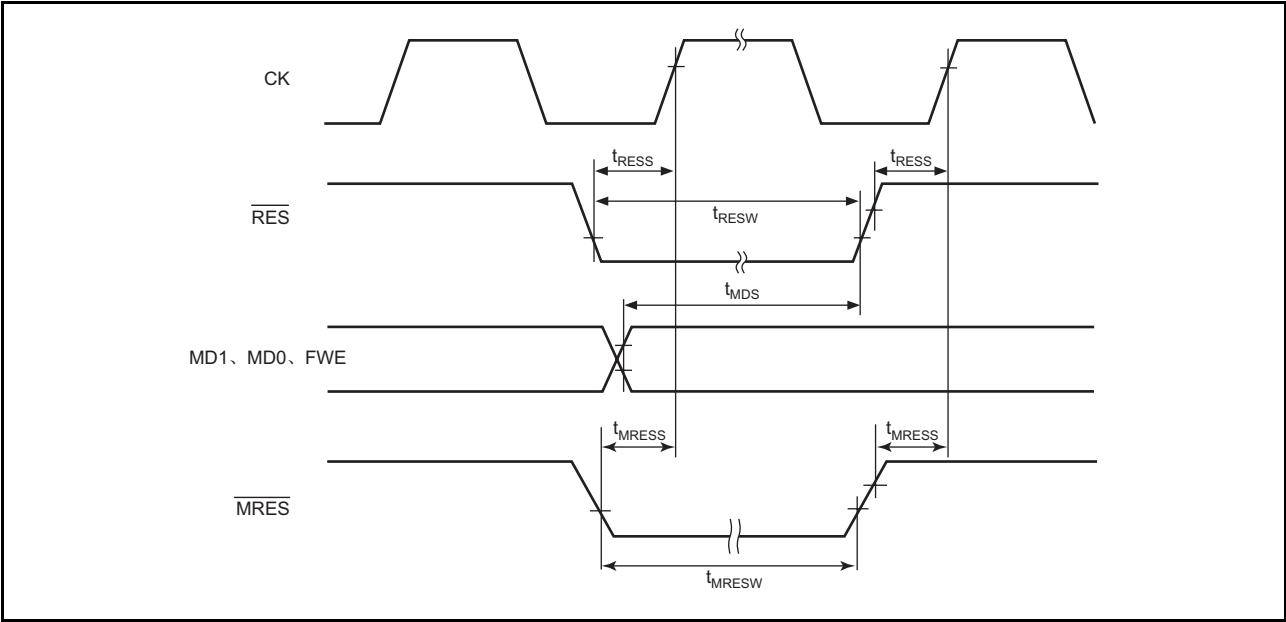


图 26.6 复位的输入时序

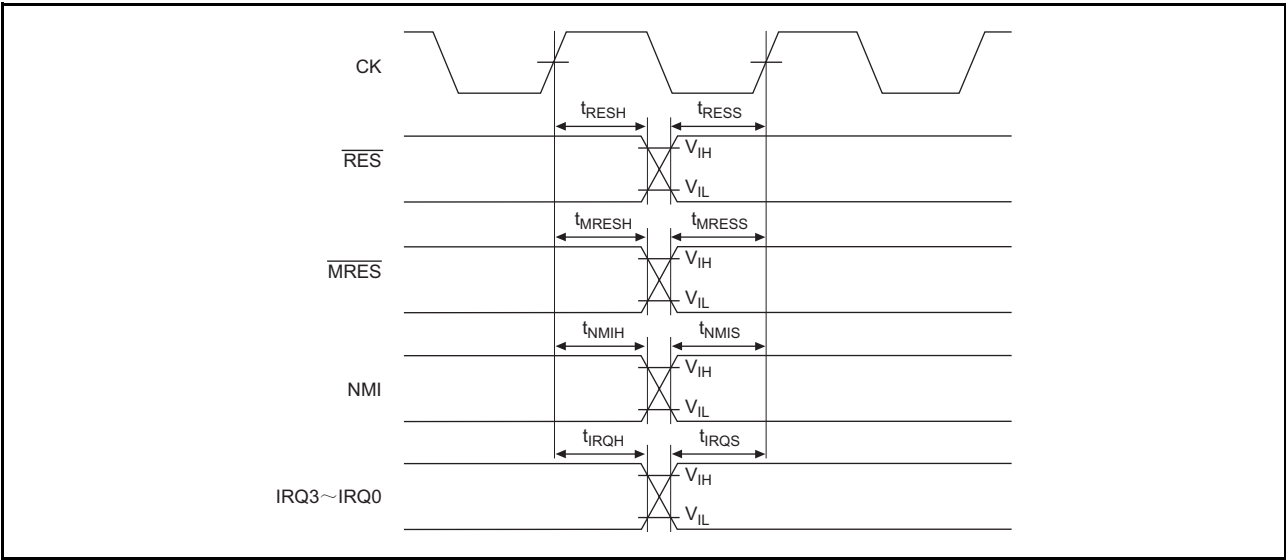


图 26.7 中断信号的输入时序

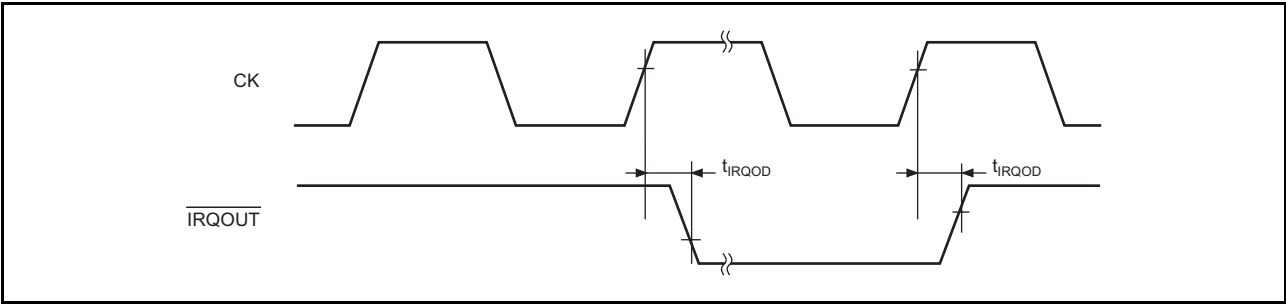


图 26.8 中断信号的输出时序

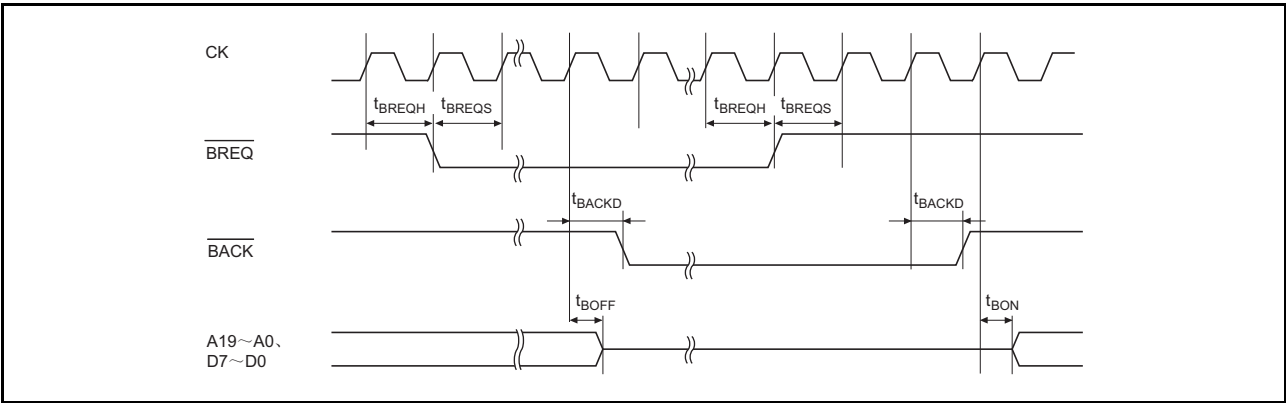


图 26.9 总线权的释放时序

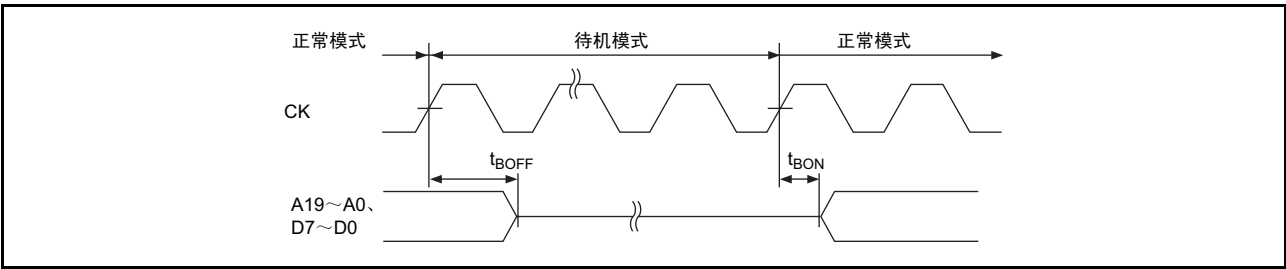


图 26.10 待机时的引脚驱动时序

26.3.3 AC 总线的时序规格

表 26.8 总线时序

条件: $V_{CC}=3.0V \sim 3.6V$ 或者 $4.0V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{refh}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{SS}=AV_{SS}=AV_{refl}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业产品)

项 目	符号	Min.	Max.	单位	参照图
地址延迟时间 1	t_{AD1}	1	20	ns	图 26.11 ~ 图 26.15
地址准备时间	t_{AS}	0	—	ns	图 26.11 ~ 图 26.14
地址保持时间	t_{AH}	0	—	ns	图 26.11 ~ 图 26.14
$\overline{CS}$ 延迟时间	t_{CSD}	1	18	ns	图 26.11 ~ 图 26.15
$\overline{CS}$ 准备时间	t_{CSS}	0	—	ns	图 26.11 ~ 图 26.14
$\overline{CS}$ 保持时间	t_{CSH}	0	—	ns	图 26.11 ~ 图 26.14
读选通的延迟时间	t_{RSD}	$1/2t_{Bcyc}+1$	$1/2t_{Bcyc}+18$	ns	图 26.11 ~ 图 26.15
读数据的准备时间 1	t_{RDS1}	$1/2t_{Bcyc}+18$	—	ns	图 26.11 ~ 图 26.15
读数据的保持时间 1	t_{RDH1}	0	—	ns	图 26.11 ~ 图 26.15
读数据的存取时间	t_{ACC}^{*2}	$t_{Bcyc} \times (n+1.5) - 33^{*1}$	—	ns	图 26.11 ~ 图 26.15
读选通开始的存取时间	t_{OE}^{*2}	$t_{Bcyc} \times (n+1) - 31^{*1}$	—	ns	图 26.11 ~ 图 26.15
写选通的延迟时间 1	t_{WSD1}	$1/2t_{Bcyc}+1$	$1/2t_{Bcyc}+18$	ns	图 26.11 ~ 图 26.15
写数据的延迟时间 1	t_{WDD1}	—	18	ns	图 26.11 ~ 图 26.15
写数据的保持时间 1	t_{WDH1}	1	11	ns	图 26.11 ~ 图 26.15
写数据的保持时间	t_{WRH}	0	—	ns	图 26.11 ~ 图 26.14
WAIT 准备时间	t_{WTS}	$1/2t_{Bcyc}+17$	—	ns	图 26.12 ~ 图 26.15
WAIT 保持时间	t_{WTH}	$1/2t_{Bcyc}+7$	—	ns	图 26.12 ~ 图 26.15

【注】 t_{Bcyc} 表示外部总线时钟 ($B\phi=CK$) 的周期。

*1 n 为等待数

*2 如果满足存取时间, 就不需要满足 t_{RDS1} 。

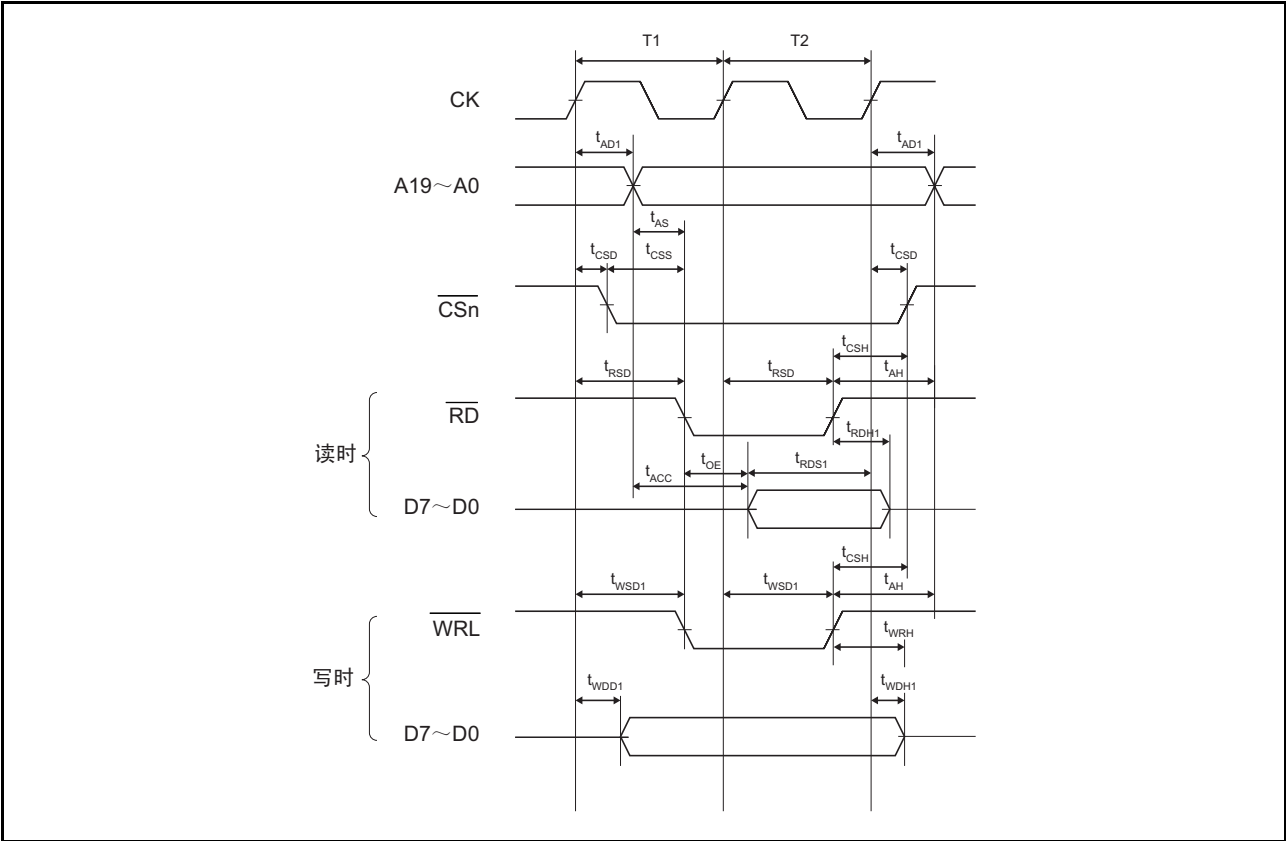


图 26.11 通常空间的基本总线周期（无等待）

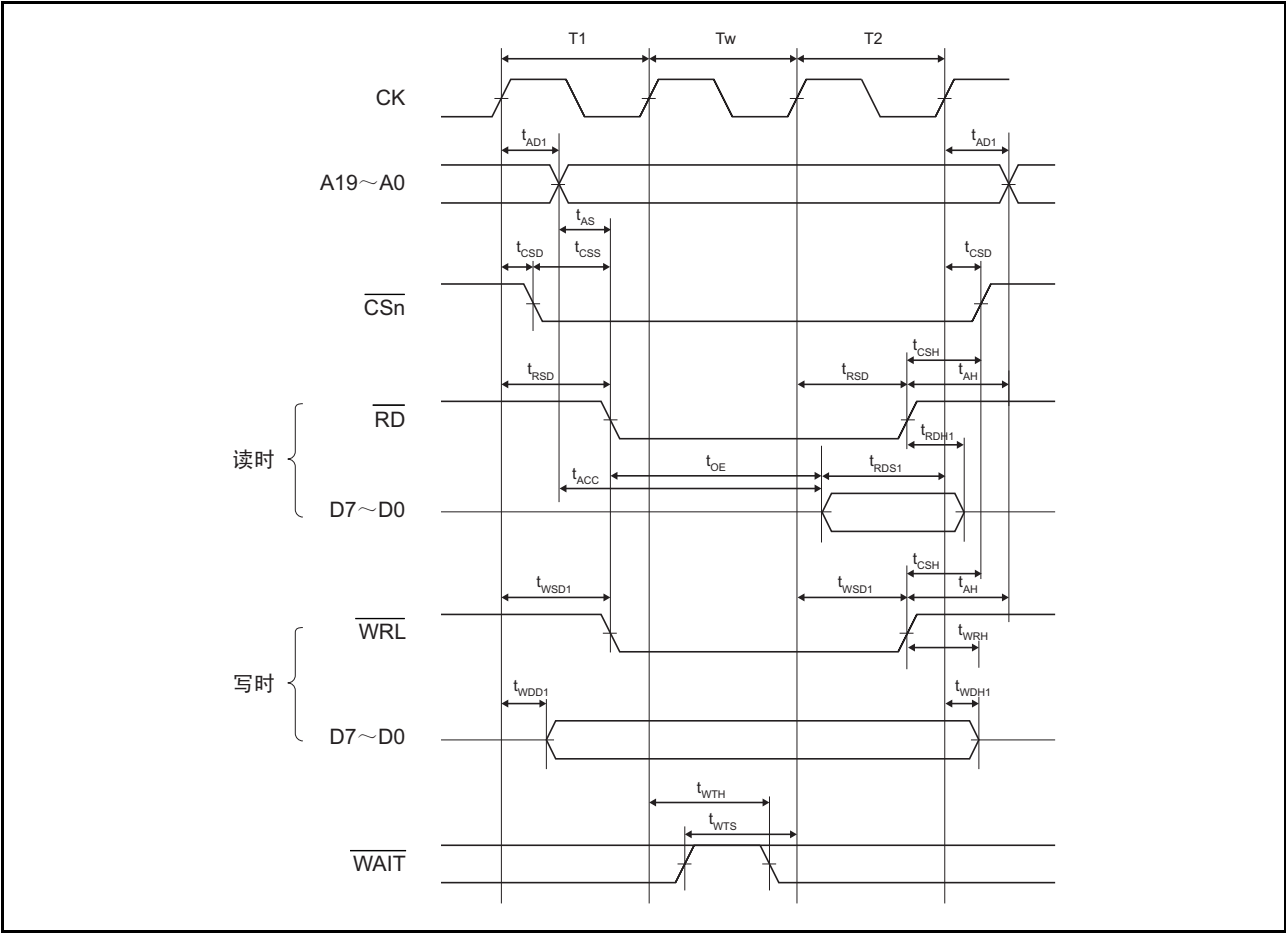


图 26.12 通常空间的基本总线周期（1 个软件等待）

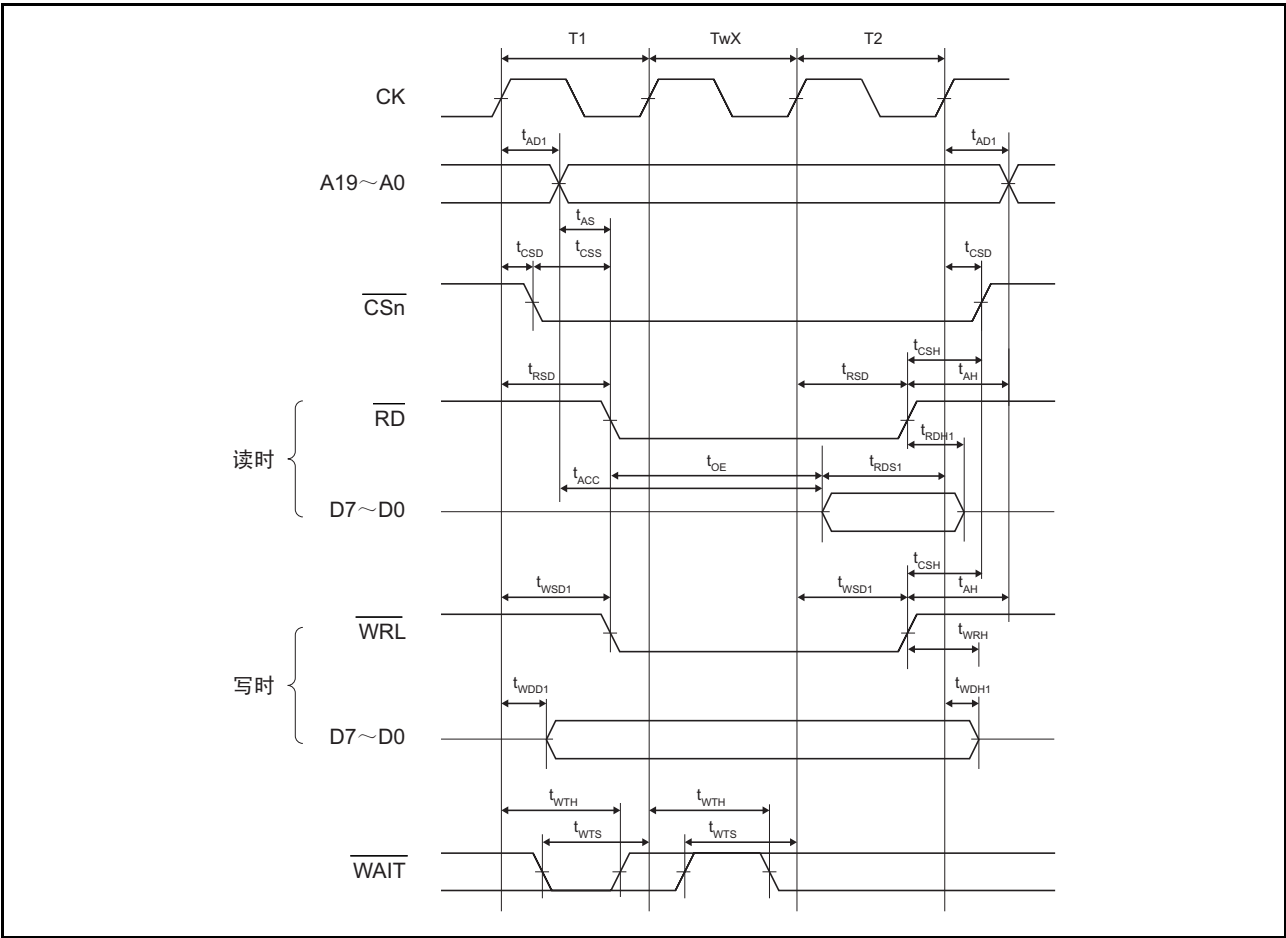


图 26.13 通常空间的基本总线周期（插入 1 个外部等待）

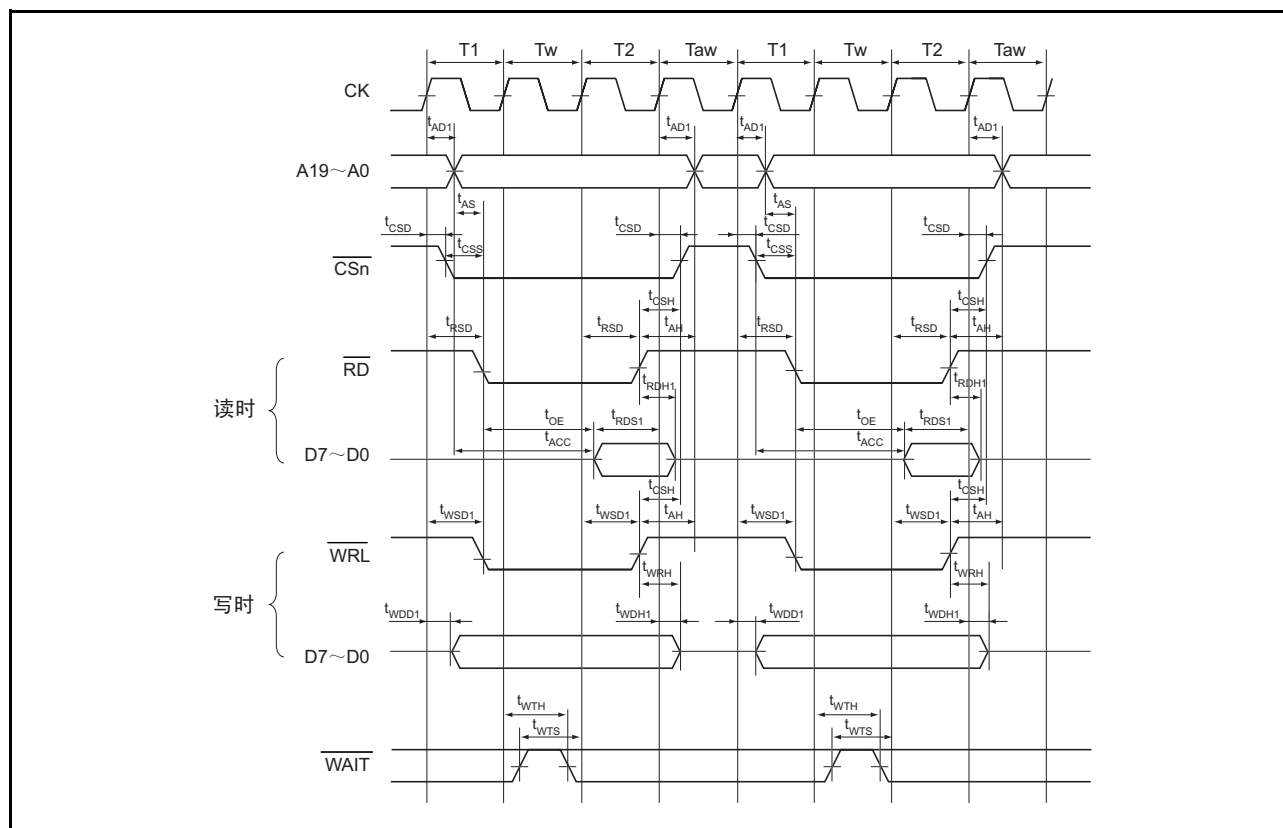


图 26.14 通常空间的基本总线周期
(1 个软件等待、外部等待有效 (WM 位 = 0)、无空闲周期)

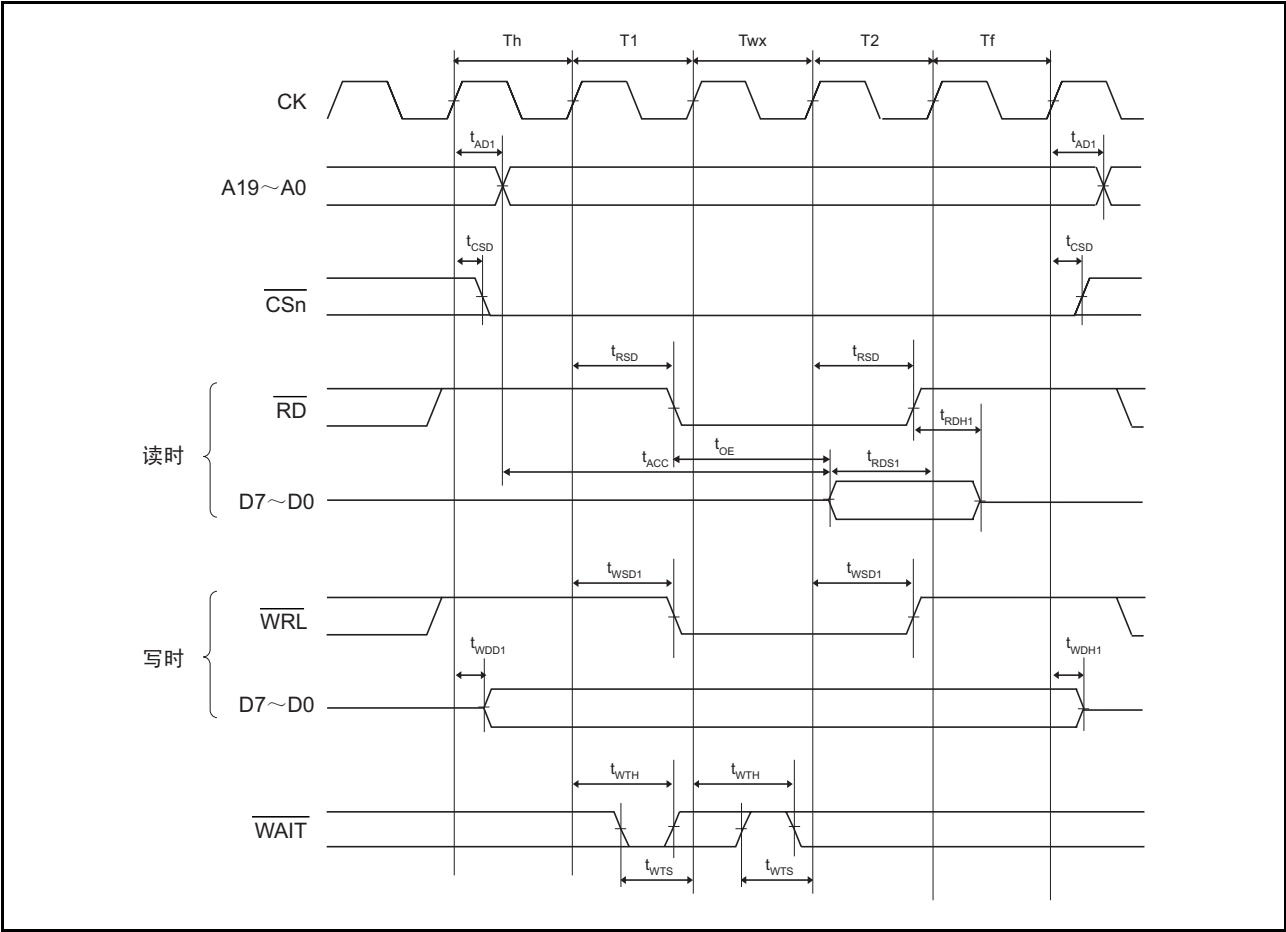


图 26.15 通常空间的 CS 扩展总线周期
(SW=1 个周期、HW=1 个周期、插入 1 个外部等待)

26.3.4 多功能定时器脉冲单元 2 (MTU2) 的时序

表 26.9 多功能定时器脉冲单元 2 (MTU2) 的时序

条件: $V_{CC}=3.0V \sim 3.6V$ 或者 $4.0V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{refh}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=AV_{refl}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业产品)

项 目	符号	Min.	Max.	单位	参照图
输出比较的输出延迟时间	t_{TOCD}	—	50	ns	图 26.16
输入捕捉的输入准备时间	t_{TICS}	20	—	ns	
输入捕捉的输入脉宽 (指定单边沿时)	t_{TICW}	1.5	—	t_{MPcyc}	
输入捕捉的输入脉宽 (指定双边沿时)	t_{TICW}	2.5	—	t_{MPcyc}	
定时器的输入准备时间	t_{TCKS}	20	—	ns	图 26.17
定时器的时钟脉宽 (指定单边沿时)	$t_{TCKWH/L}$	1.5	—	t_{MPcyc}	
定时器的时钟脉宽 (指定双边沿时)	$t_{TCKWH/L}$	2.5	—	t_{MPcyc}	
定时器的时钟脉宽 (相位计数模式)	$t_{TCKWH/L}$	2.5	—	t_{MPcyc}	

【注】 t_{MPcyc} 表示 MTU2 时钟 ($MP\phi$) 的周期。

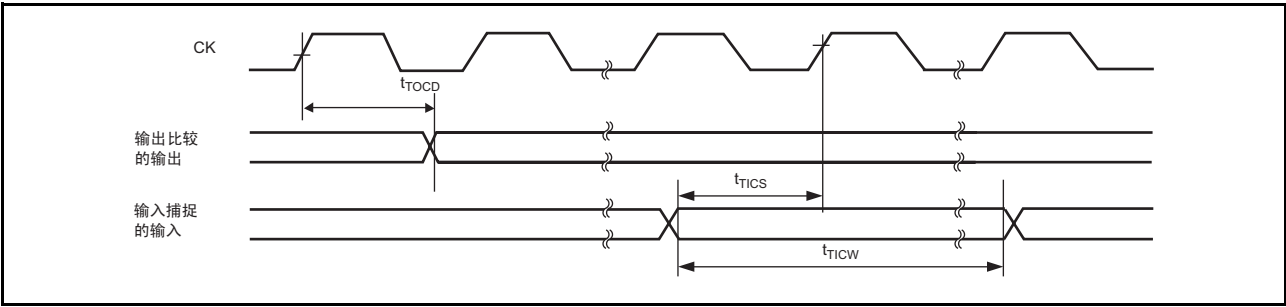


图 26.16 MTU2 的输入 / 输出时序

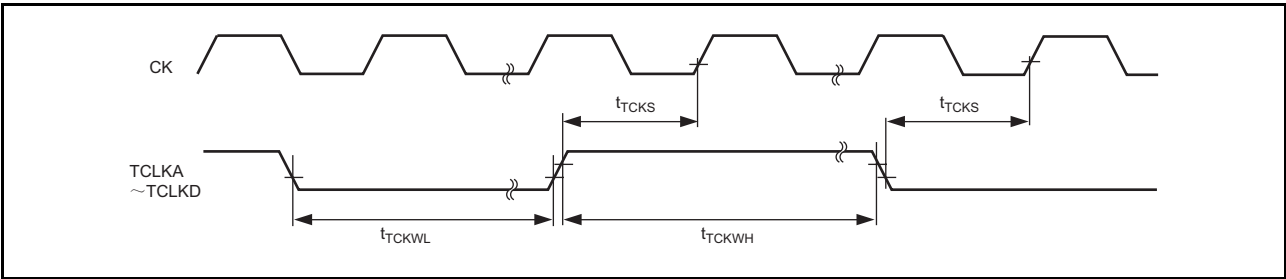


图 26.17 MTU2 的时钟输入时序

26.3.5 多功能定时器脉冲单元 2S (MTU2S) 的时序

表 26.10 多功能定时器脉冲单元 2S (MTU2S) 的时序

条件: $V_{CC}=3.0V \sim 3.6V$ 或者 $4.0V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{refh}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=AV_{refl}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业产品)

项 目	符号	Min.	Max.	单位	参照图
输出比较的输出延迟时间	t_{TOCD}	—	50	ns	图 26.18
输入捕捉的输入准备时间	t_{TICS}	20	—	ns	
输入捕捉的输入脉宽 (指定单边沿时)	t_{TICW}	1.5	—	t_{Mlcyc}	
输入捕捉的输入脉宽 (指定双边沿时)	t_{TICW}	2.5	—	t_{Mlcyc}	

【注】 t_{Mlcyc} 表示 MTU2S 时钟 (MI ϕ) 的周期。

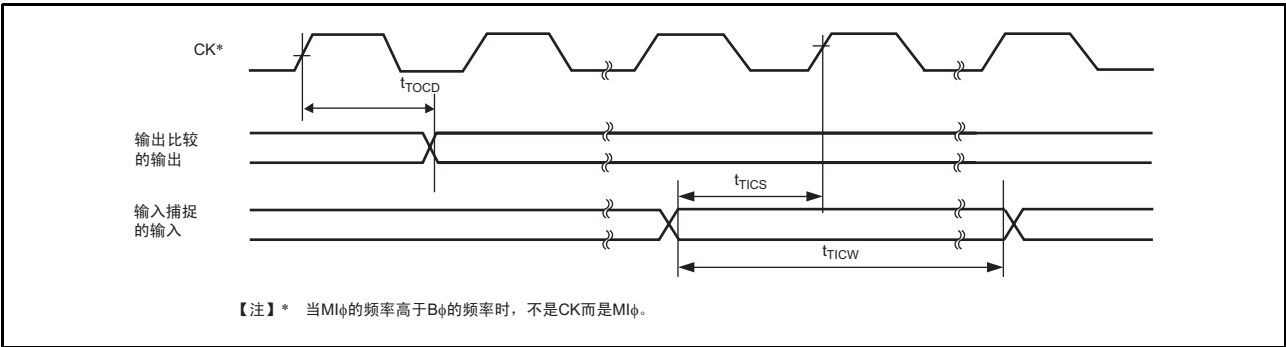


图 26.18 MTU2S 输入 / 输出时序

26.3.6 I/O 端口的时序

表 26.11 I/O 端口的时序

条件: $V_{CC}=3.0V \sim 3.6V$ 或者 $4.0V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{refh}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{SS}=AV_{SS}=AV_{refl}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业产品)

项 目	符号	Min.	Max.	单位	参照图
端口输出数据的延迟时间	t_{PWD}	—	50	ns	图 26.19
端口输入的保持时间	t_{PRH}	20	—	ns	
端口输入的准备时间	t_{PRS}	20	—	ns	

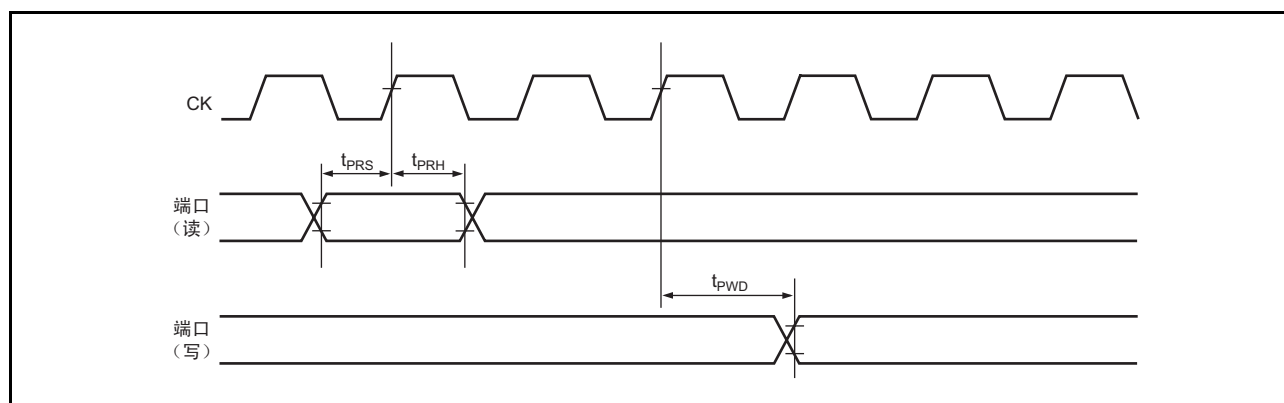


图 26.19 I/O 端口的输入 / 输出时序

26.3.7 看门狗定时器 (WDT) 的时序

表 26.12 看门狗定时器的 (WDT) 时序

条件: $V_{CC}=3.0V \sim 3.6V$ 或者 $4.0V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{refh}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{SS}=AV_{SS}=AV_{refl}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业产品)

项 目	符号	Min.	Max.	单位	参照图
WDTOVF 延迟时间	t_{WOVD}	—	50	ns	图 26.20

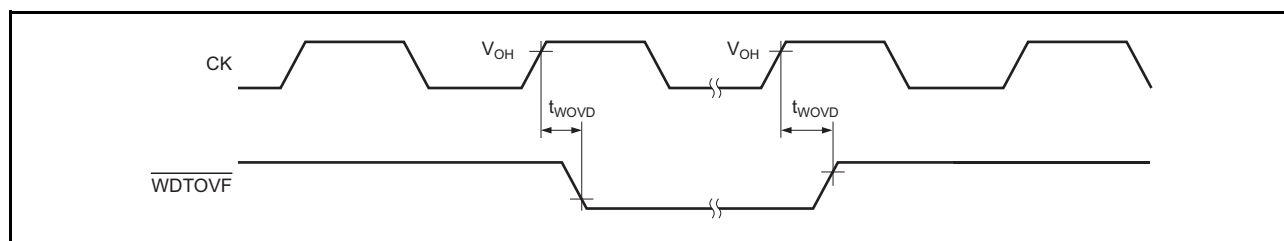


图 26.20 看门狗定时器的时序

26.3.8 串行通信接口（SCI）的时序

表 26.13 串行通信接口（SCI）的时序

条件: $V_{CC}=3.0V \sim 3.6V$ 或者 $4.0V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{refh}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=AV_{refl}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$ （民用产品）、 $T_a=-40 \sim +85^{\circ}C$ （工业产品）

项 目		符号	Min.	Max.	单位	参照图
输入时钟周期（异步）		t _{scyc}	4	—	tpcyc	图 26.21
输入时钟周期（时钟同步）		t _{scyc}	6	—	tpcyc	
输入时钟脉宽		t _{sckw}	0.4	0.6	tscyc	
输入时钟的上升时间		t _{sckr}	—	1.5	tpcyc	
输入时钟的下降时间		t _{sckf}	—	1.5	tpcyc	
发送数据的延迟时间	异步	t _{TXD}	—	4t _{pcyc} +10	ns	图 26.22
接收数据的准备时间		t _{RXS}	4t _{pcyc}	—	ns	
接收数据的保持时间		t _{RXH}	4t _{pcyc}	—	ns	
发送数据的延迟时间	时钟同步	t _{TXD}	—	3t _{pcyc} +10	ns	
接收数据的准备时间		t _{RXS}	2t _{pcyc} +50	—	ns	
接收数据的保持时间		t _{RXH}	2t _{pcyc}	—	ns	

【注】 t_{pcyc} 表示外围时钟（ $P\phi$ ）的周期。

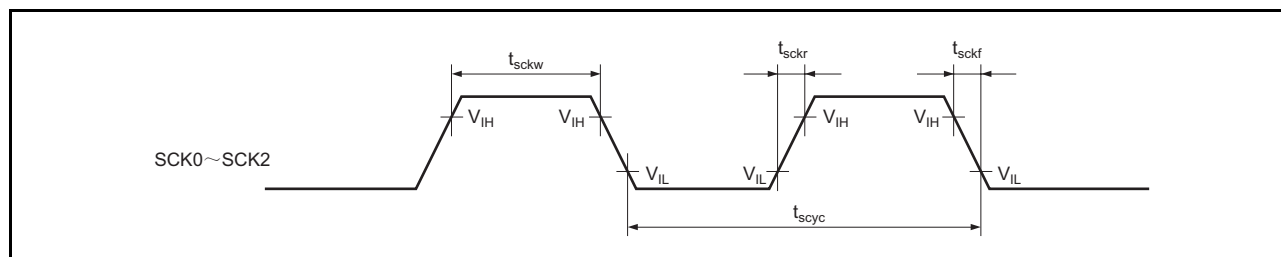


图 26.21 输入时钟的时序

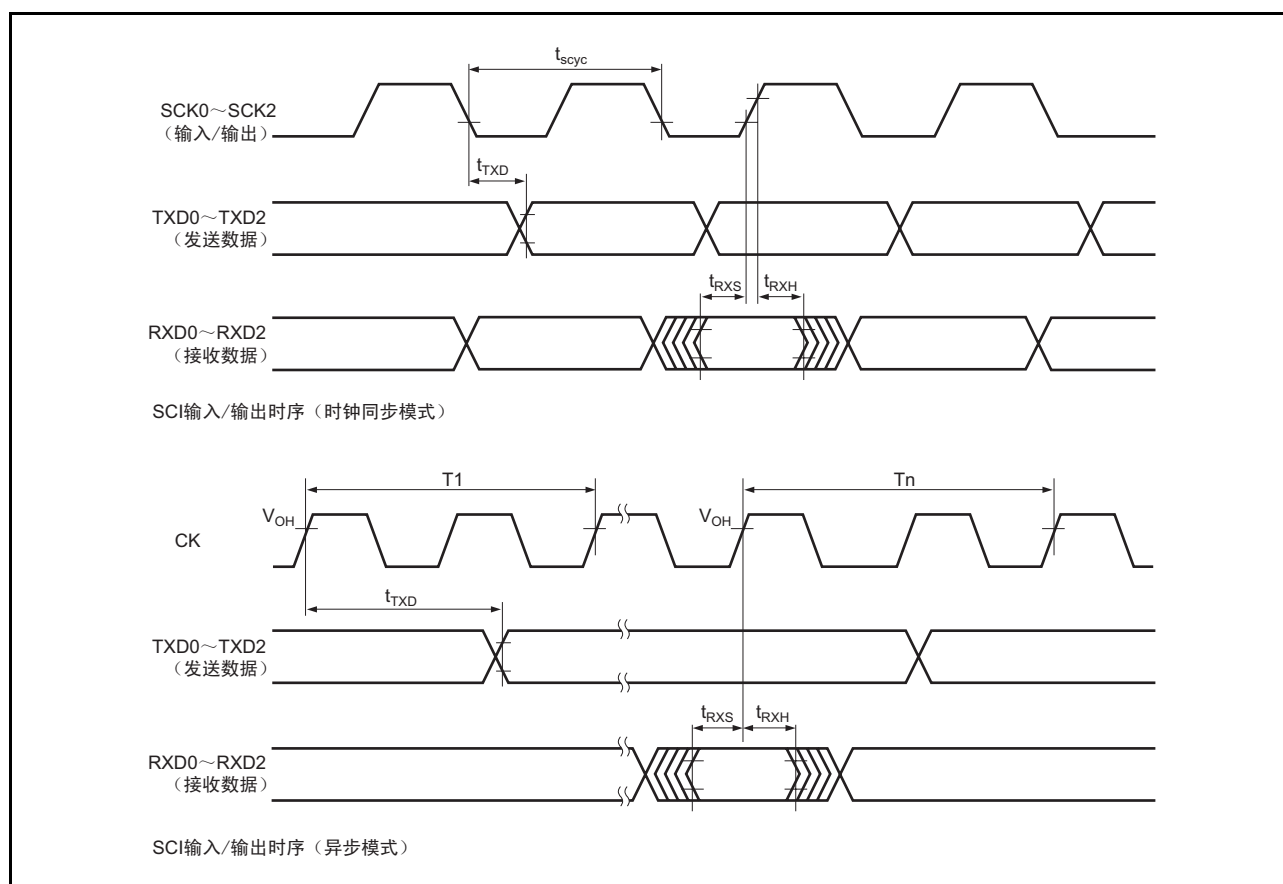


图 26.22 SCI 的输入 / 输出时序

26.3.9 串行通信单元（SSU）的时序

表 26.14 串行通信单元（SSU）的时序

条件: $V_{CC}=3.0V \sim 3.6V$ 或者 $4.0V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{refh}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{V_{SS}}=AV_{SS}=AV_{refl}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$ （民用产品）、 $T_a=-40 \sim +85^{\circ}C$ （工业产品）

项 目		符号	Min.	Max.	单位	参照图
时钟周期	主控	t_{SUcyc}	4	256	t_{pcyc}	图 26.23 图 26.24 图 26.25 图 26.26
	从属		4	256		
时钟的高电平脉宽	主控	t_{HI}	60	—	ns	
	从属		60	—		
时钟的低电平脉宽	主控	t_{LO}	60	—	ns	
	从属		60	—		
时钟的上升时间		t_{RISE}	—	20	ns	
时钟的下降时间		t_{FALL}	—	20	ns	
数据输入的准备时间	主控	t_{SU}	30	—	ns	
	从属		30	—		
数据输入的保持时间	主控	t_{H}	10	—	ns	
	从属		10	—		
SCS 准备时间	主控	t_{LEAD}	1.5	—	t_{pcyc}	
	从属		1.5	—		
SCS 保持时间	主控	t_{LAG}	1.5	—	t_{pcyc}	
	从属		1.5	—		
数据输出的延迟时间	主控	t_{OD}	—	40	ns	
	从属		—	40		
数据输出的保持时间	主控	t_{OH}	30	—	ns	
	从属		30	—		
连续发送的延迟时间	主控	t_{TD}	1.5	—	t_{pcyc}	
	从属		1.5	—		
从属器件的存取时间		t_{SA}	—	1	t_{pcyc}	图 26.25
从属器件的输出释放时间		t_{REL}	—	1	t_{pcyc}	图 26.26

【注】 t_{pcyc} 表示外围时钟（Pφ）的周期。

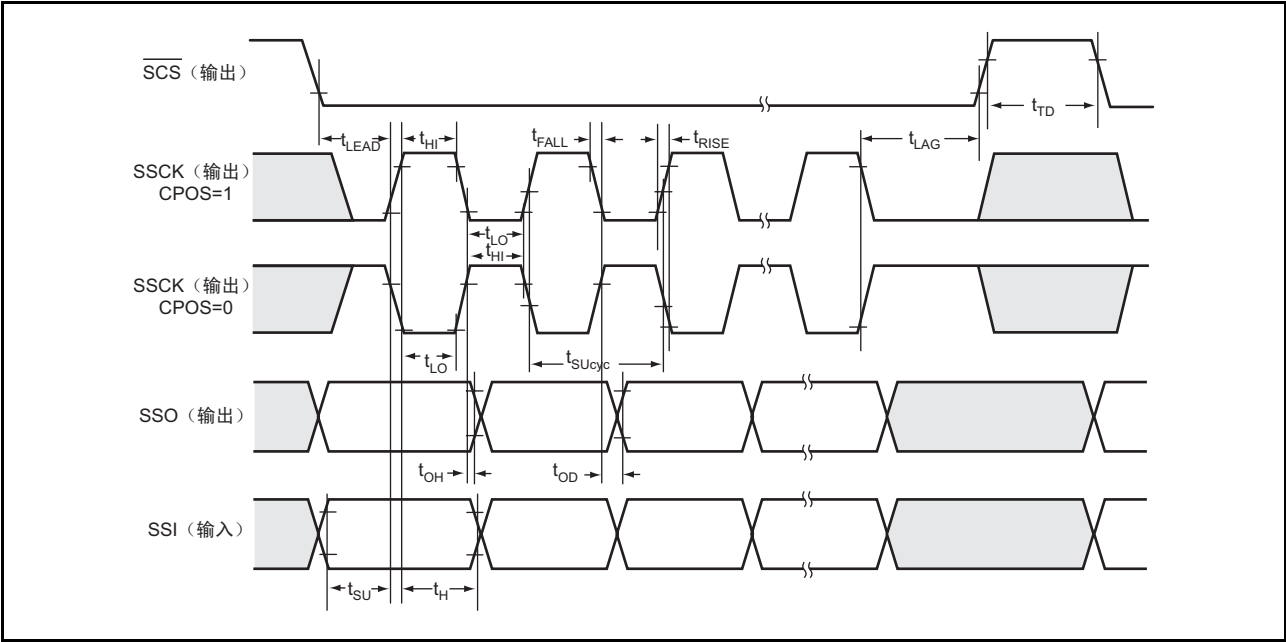


图 26.23 SSU 的时序（主控、CPHS=1）

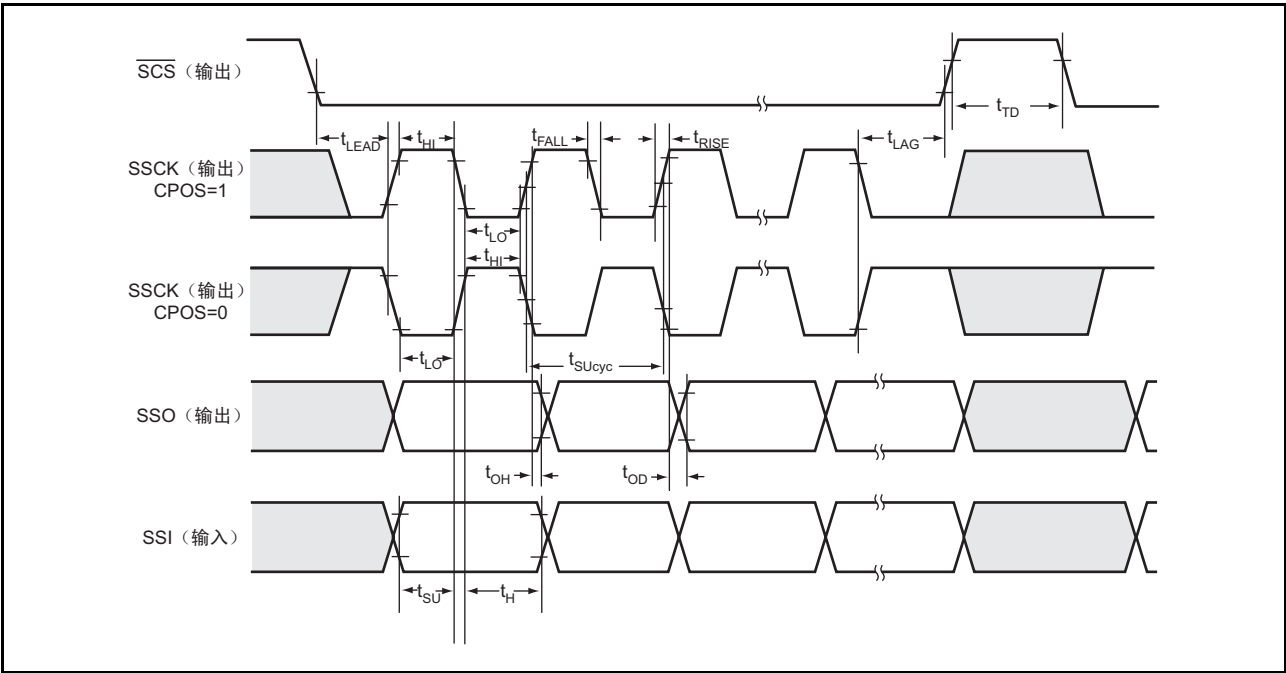


图 26.24 SSU 的时序（主控、CPHS=0）

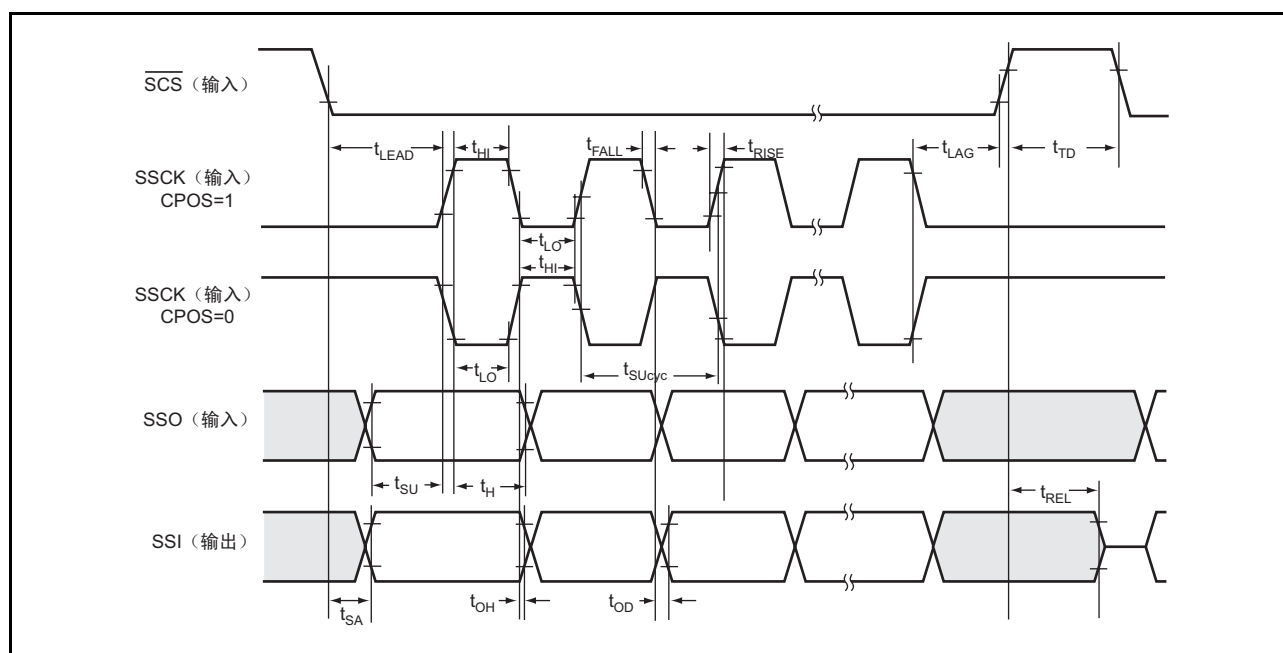


图 26.25 SSU 的时序 (从属、CPHS=1)

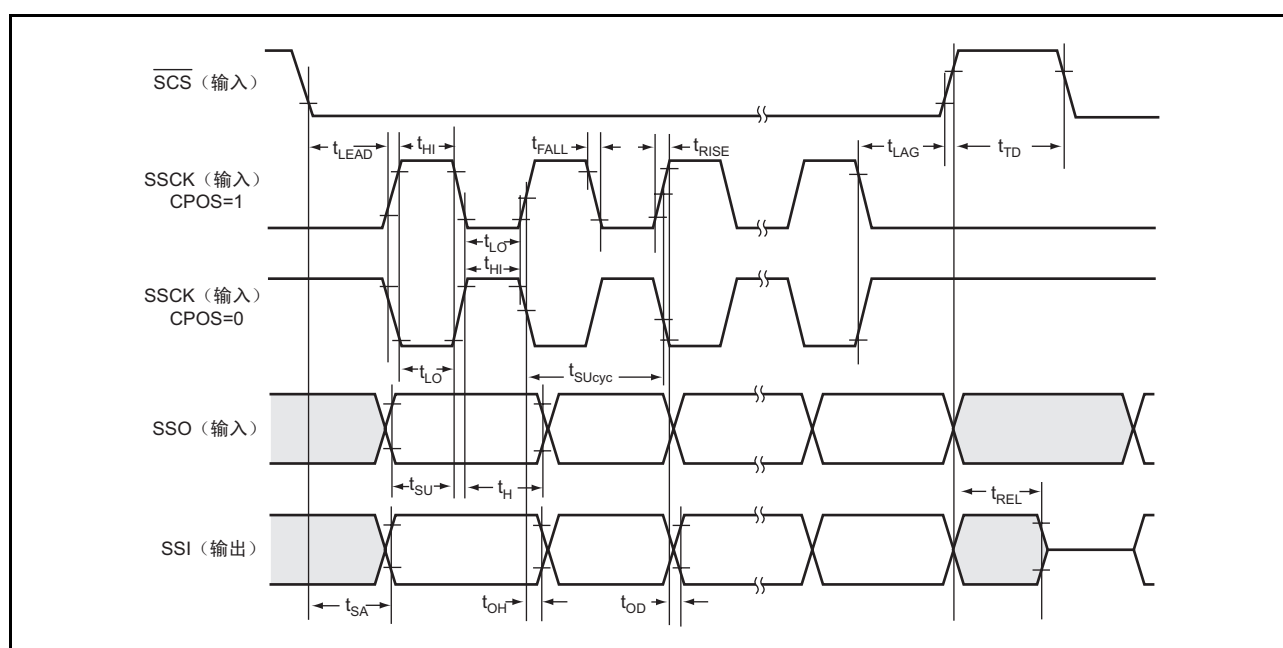


图 26.26 SSU 的时序 (从属、CPHS=0)

26.3.10 控制器局域网（RCAN-ET）的时序

RCAN-ET 的时序如表 26.15 所示。

表 26.15 控制器局域网（RCAN-ET）的时序

条件: $V_{CC}=3.0V \sim 3.6V$ 或者 $4.0V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{refh}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=AV_{refl}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$ （民用产品）、 $T_a=-40 \sim +85^{\circ}C$ （工业产品）

项 目	符号	Min.	Max.	单位	参照图
发送数据的延迟时间	t_{CTxD}	—	100	ns	图 26.27
接收数据的准备时间	t_{CRxS}	100	—	ns	
接收数据的保持时间	t_{CRxH}	100	—	ns	

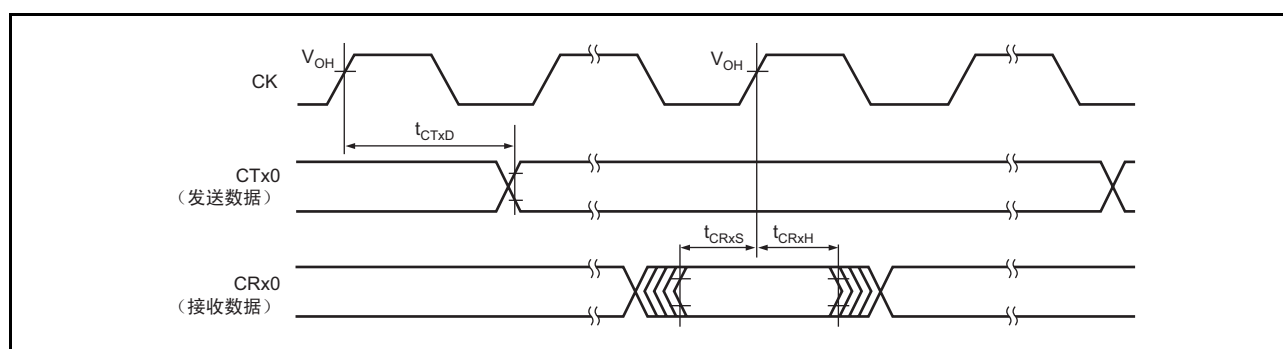


图 26.27 RCAN-ET 的输入 / 输出时序

26.3.11 端口输出允许（POE）的时序

表 26.16 端口输出允许（POE）的时序

条件: $V_{CC}=3.0V \sim 3.6V$ 或者 $4.0V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{refh}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=AV_{refl}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$ （民用产品）、 $T_a=-40 \sim +85^{\circ}C$ （工业产品）

项 目	符号	Min.	Max.	单位	参照图
POE 输入的准备时间	t_{POES}	50	—	ns	图 26.28
POE 输入脉宽	t_{POEW}	1.5	—	t_{pcyc}	

【注】 t_{pcyc} 表示外围时钟（Pφ）的周期。

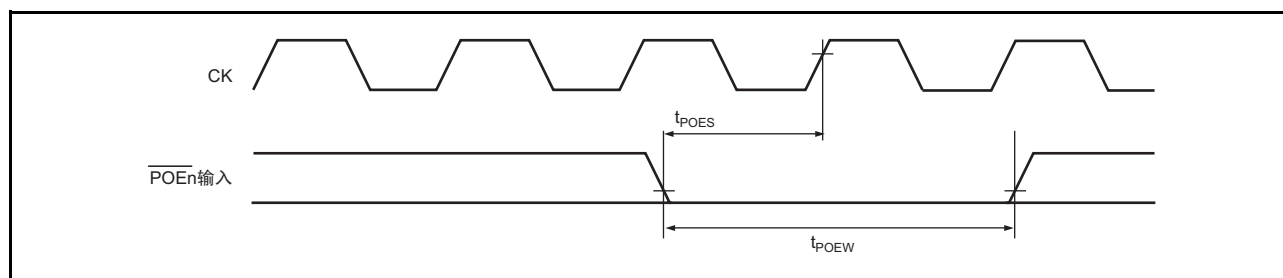


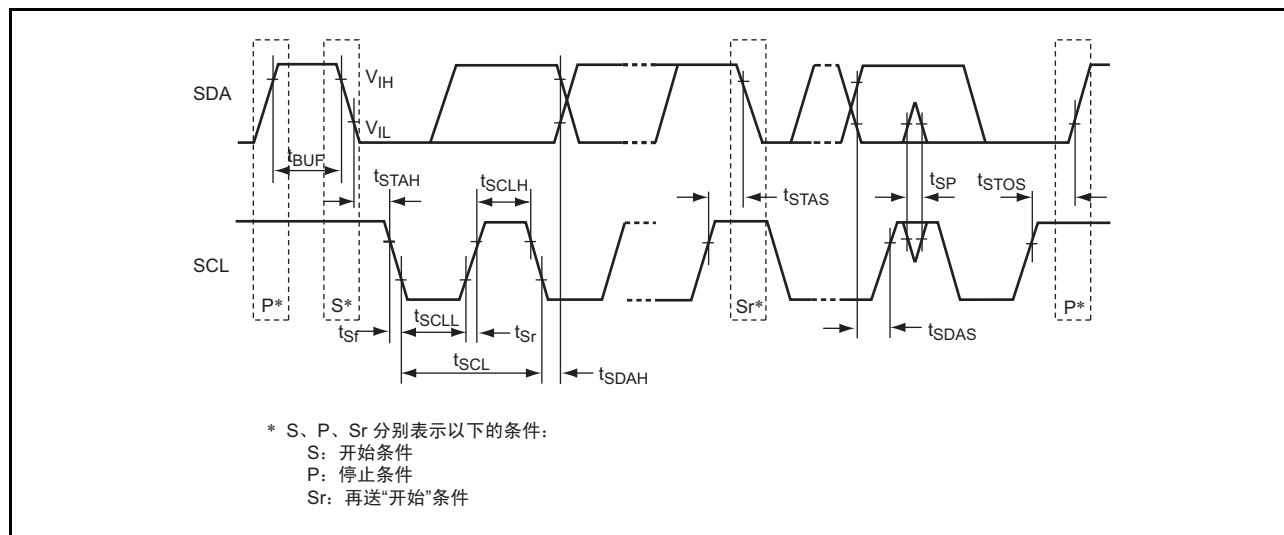
图 26.28 POE 输入的时序

26.3.12 I²C 总线接口 2 (I²C2) 的时序表 26.17 I²C 总线接口 2 (I²C2) 的时序

条件: $V_{CC}=3.0V \sim 3.6V$ 或者 $4.0V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{refh}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=AV_{refl}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业产品)

项 目	符号	Min.	Typ.	Max.	单位	参照图
SCL 输入周期时间	t_{SCL}	$12t_{pcyc}+60$ 0	—	—	ns	图 26.29
SCL 输入高电平脉宽	t_{SCLH}	$3t_{pcyc}+300$	—	—	ns	
SCL 输入低电平脉宽	t_{SCLL}	$5t_{pcyc}+300$	—	—	ns	
SCL、SDA 输入的下陷时间	t_{Sf}	—	—	300	ns	
SCL、SDA 输入的尖峰脉冲消除时间	t_{SP}	—	—	$1t_{pcyc}$	ns	
SDA 输入的总线自由时间	t_{BUF}	5	—	—	t_{pcyc}	
开始条件输入的保持时间	t_{STAH}	3	—	—	t_{pcyc}	
再送开始条件输入的准备时间	t_{STAS}	3	—	—	t_{pcyc}	
停止条件输入的准备时间	t_{STOS}	3	—	—	t_{pcyc}	
数据输入的准备时间	t_{SDAS}	$1t_{pcyc}+20$	—	—	ns	
数据输入的保持时间	t_{SDAH}	0	—	—	ns	
SCL、SDA 的电容负载	Cb	0	—	400	pF	
SCL、SDA 输出的下陷时间	t_{Sf}	—	—	250	ns	

【注】 t_{pcyc} 表示外围时钟 ($P\phi$) 的周期。

图 26.29 I²C 总线接口 2 的输入 / 输出时序

26.3.13 UBC 的触发时序

表 26.18 UBC 的触发时序

条件: $V_{CC}=3.0V \sim 3.6V$ 或者 $4.0V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{refh}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{SS}=AV_{SS}=AV_{refl}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业产品)

项 目	符号	Min.	Max.	单位	参照图
UBCTR _G 延迟时间	t_{UBCTGD}	—	150	ns	图 26.30

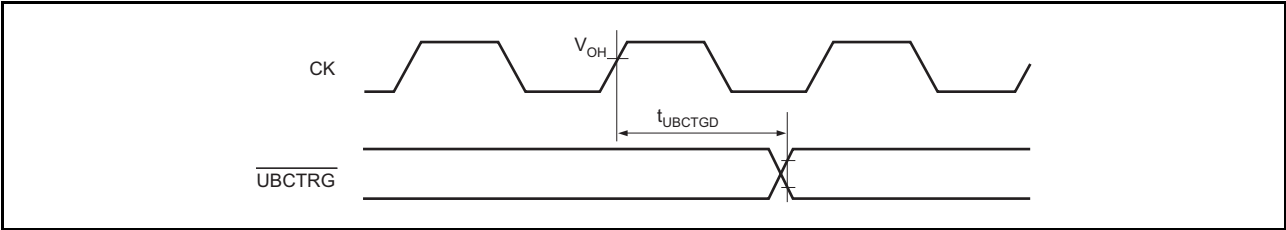


图 26.30 UBC 的触发时序

26.3.14 A/D 转换器的时序

表 26.19 A/D 转换器的时序

条件: $V_{CC}=3.0V \sim 3.6V$ 或者 $4.0V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{refh}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{SS}=AV_{SS}=AV_{refl}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业产品)

项 目	符号	Min.	Typ.	Max.	单位	参照图
外部触发输入的开始延迟时间	t_{TRGS}	25	—	—	ns	图 26.31

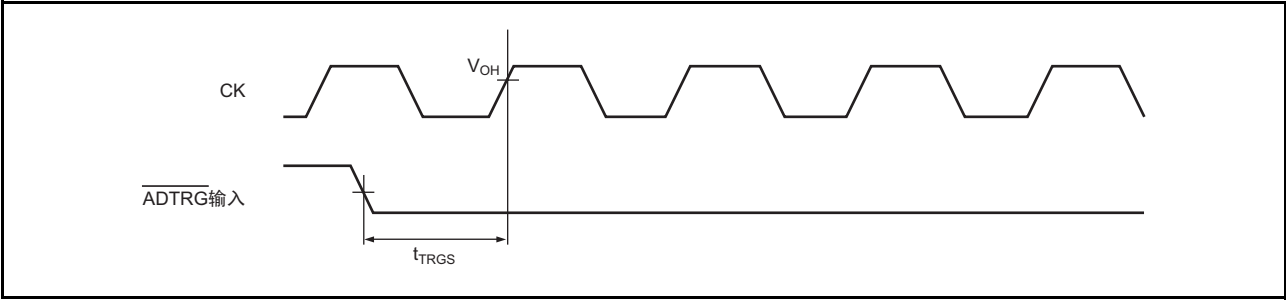


图 26.31 外部触发的输入时序

26.3.15 AC 特性的测量条件

- 输入信号电平: $V_{IL}(\text{Max.})/V_{IH}(\text{Min.})$
- 输出信号参照电平:
高电平: 2.0V, 低电平: 0.8V

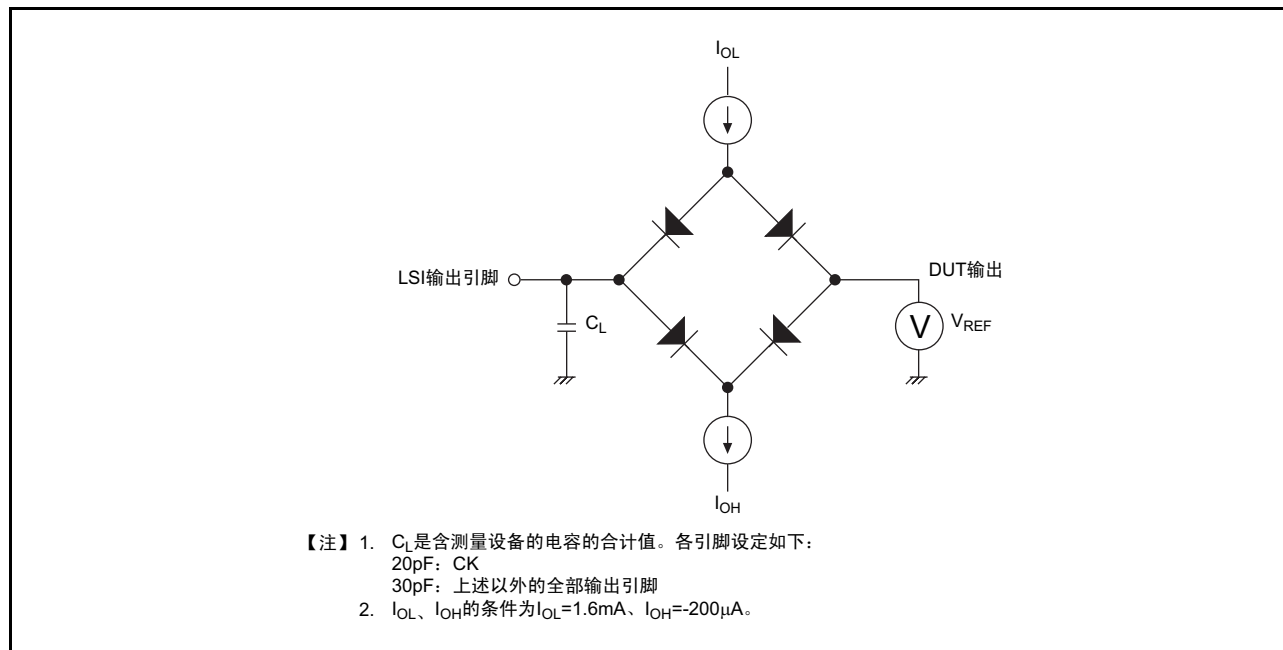


图 26.32 输出负载电路

26.4 A/D 转换器特性

表 26.20 A/D 转换器特性

条件: $V_{CC}=3.0\text{V} \sim 3.6\text{V}$ 或者 $4.0\text{V} \sim 5.5\text{V}$ 、 $AV_{CC}=4.5\text{V} \sim 5.5\text{V}$ 、 $AV_{refh}=4.5\text{V} \sim AV_{CC}$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=AV_{refl}=0\text{V}$ 、 $T_a=-20 \sim +85^\circ\text{C}$ (民用产品)、 $T_a=-40 \sim +85^\circ\text{C}$ (工业产品)

项 目	Min.	Typ.	Max.	单位
分辨率	12	12	12	位
转换时间	1.25*1	—	—	μs
模拟输入电容	—	—	5	pF
容许信号源阻抗	—	—	3	$\text{k}\Omega$
非线性误差	—	—	$\pm 4^*2$	LSB
偏移误差	—	—	$\pm 7.5^*2$	LSB
满刻度误差	—	—	$\pm 7.5^*2$	LSB
量化误差	—	—	$\pm 0.5^*2$	LSB
绝对精度 *3	—	—	± 8	LSB

【注】*1 是在未使用采样和保持电路并且 A/D 时钟 =40MHz 运行时的每个通道的转换时间。

*2 是参考值。

*3 绝对精度的保证范围为 $AV_{refl}+0.25\text{V} \sim AV_{refh}-0.25\text{V}$ 。

26.5 闪存特性

表 26.21 闪存特性

条件: $V_{CC}=3.0V \sim 3.6V$ 或者 $4.0V \sim 5.5V$ 、 $AV_{CC}=4.5V \sim 5.5V$ 、 $AV_{refh}=4.5V \sim AV_{CC}$ 、 $V_{SS}=PLL_{SS}=AV_{SS}=AV_{refl}=0V$ 、 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业产品)

项 目	符号	Min.	Typ.	Max.	单位
编程时间 *1*2*4	t_P	—	1	10	ms/128 字节
擦除时间 *1*2*4	t_E	—	30	100	ms/4K 字节块
		—	250	800	ms/32K 字节块
		—	500	1600	ms/64K 字节块
编程时间 (总和) *1*2*4	Σt_P	—	2.5	7	s/256K 字节
擦除时间 (总和) *1*2*4	Σt_E	—	2.5	7	s/256K 字节
编程、擦除时间 (总和) *1*2*4	Σt_{PE}	—	5	14	s/256K 字节
改写次数	N_{WEC}	500*3	—	—	次

【注】 *1 编程和擦除时间取决于数据。

*2 编程和擦除时间不包含数据的传送时间。

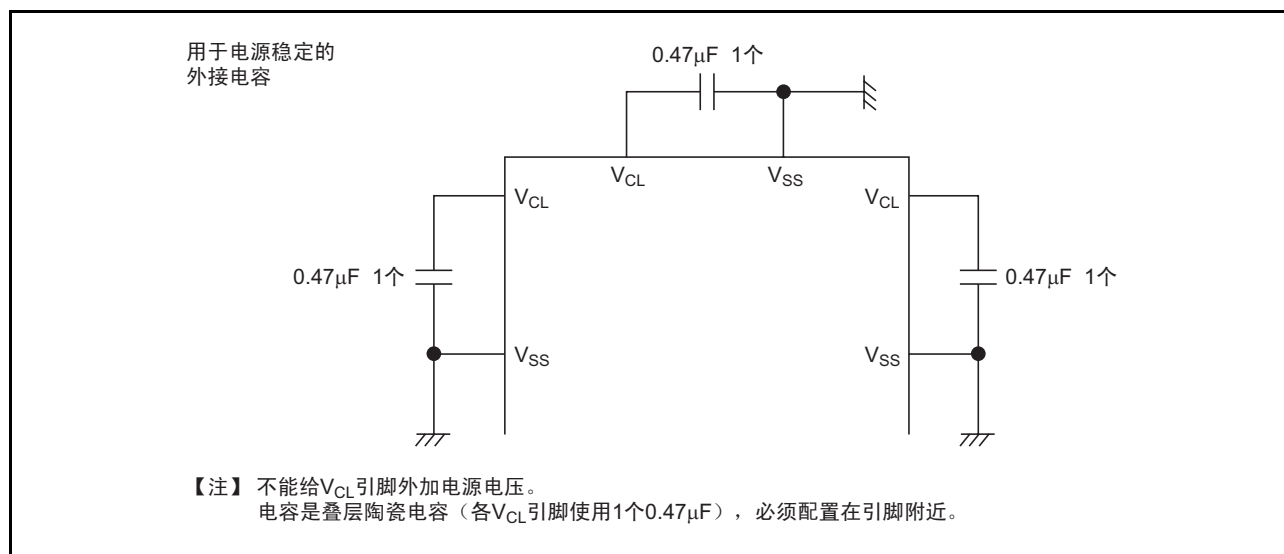
*3 是保证改写后所有特性的 Min. 次数 (保证范围: 1 ~ Min. 值)。

*4 是在包含 Min. 值的使用范围内进行改写时的特性。

26.6 使用时的注意事项

26.6.1 V_{CL} 电容的连接方法

本 LSI 内置内部降压电路, 将单片机内部的电源电压自动降到适当的电平。需要在此内部降压电源 (V_{CL} 引脚) 和 V_{SS} 引脚之间连接用于稳定内部电压的电容 ($0.47\mu F$)。外接电容的连接方法如图 26.33 所示。外接电容必须配置在引脚的附近, 不能给 V_{CL} 引脚外加电源电压。

图 26.33 V_{CL} 电容连接方法

附录

附录 A. 引脚状态

引脚的初始值因 MCU 运行模式而不同，详细内容请参照“第 20 章 引脚功能控制器（PFC）”。

表 A.1 引脚状态（SH7136）

引脚功能		引脚状态						
分类	引脚名	复位状态		低功耗状态			检测到振荡停止时	使用 POE 功能时
		上电	手动	深度软件待机	软件待机	睡眠		
时钟	XTAL	O	O	L	L	O	O	O
	EXTAL	I	I	Z	I	I	I	I
系统控制	$\overline{\text{RES}}$	I	I	I	I	I	I	I
	$\overline{\text{MRES}}$	Z	I	Z	Z	I	Z	I
	$\overline{\text{WDTOVF}}$	O*2	O	O	O	O	O	O
运行模式控制	MD1	I	I	I	I	I	I	I
	$\overline{\text{ASEMD0}}$	I*3	I*3	I*3	I*3	I*3	I*3	I*3
	FWE	I	I	I	I	I	I	I
中断	NMI	I	I	I	I	I	I	I
	IRQ0 ~ IRQ3	Z	I	Z	I	I	I	I
	$\overline{\text{IRQOUT}}$	Z	O	Z	Z	O	Z	O
MTU2	TCLKA ~ TCLKD	Z	I	Z	Z	I	I	I
	TIOC0A ~ TIOC0D	Z	I/O	Z	K*1	I/O	I/O	Z
	TIOC1A、TIOC1B	Z	I/O	Z	K*1	I/O	I/O	I/O
	TIOC2A、TIOC2B	Z	I/O	Z	K*1	I/O	I/O	I/O
	TIOC3A、TIOC3C	Z	I/O	Z	K*1	I/O	I/O	I/O
	TIOC3B、TIOC3D	Z	I/O	Z	Z	I/O	Z	Z
	TIOC4A ~ TIOC4D	Z	I/O	Z	Z	I/O	Z	Z
	TIC5U、TIC5V	Z	I	Z	Z	I	I	I
MTU2S	TIOC3BS、TIOC3DS	Z	I/O	Z	Z	I/O	Z	Z
	TIOC4AS ~ TIOC4DS	Z	I/O	Z	Z	I/O	Z	Z
	TIC5US、TIC5VS	Z	I	Z	Z	I	I	I
POE	$\overline{\text{POE0}} \sim \overline{\text{POE2}}$ 、 $\overline{\text{POE4}} \sim \overline{\text{POE6}}$ 、 $\overline{\text{POE8}}$	Z	I	Z	Z	I	I	I
SCI	SCK0 ~ SCK2	Z	I/O	Z	Z	I/O	I/O	I/O
	RXD0 ~ RXD2	Z	I	Z	Z	I	I	I
	TXD0 ~ TXD2	Z	O	Z	O*1	O	O	O
SSU	SSCK	Z	I/O	Z	Z	I/O	I/O	I/O
	$\overline{\text{SCS}}$	Z	I/O	Z	Z	I/O	I/O	I/O
	SSI	Z	I/O	Z	Z	I/O	I/O	I/O
	SSO	Z	I/O	Z	Z	I/O	I/O	I/O
I ² C2	SCL	Z	I/O	Z	Z	I/O	I/O	I/O
	SDA	Z	I/O	Z	Z	I/O	I/O	I/O

引脚功能		引脚状态						
分类	引脚名	复位状态		低功耗状态			检测到振荡停止时	使用 POE 功能时
		上电	手动	深度软件待机	软件待机	睡眠		
UBC	$\overline{\text{UBCTRG}}$	Z	O	Z	O^{*1}	O	O	O
RCAN-ET	CTx0	Z	O	Z	O^{*1}	O	O	O
	CRx0	Z	I	Z	Z	I	I	I
A/D 转换器	AN0 ~ AN3、 AN8 ~ AN15	Z	I	Z	Z	I	I	I
	$\overline{\text{ADTRG}}$	Z	I	Z	Z	I	I	I
I/O 端口	PA0 ~ PA15	Z	I/O	Z	K^{*1}	I/O	I/O	I/O
	PB2 ~ PB7	Z	I/O	Z	K^{*1}	I/O	I/O	I/O
	PE0 ~ PE3	Z	I/O	Z	K^{*1}	I/O	I/O	Z
	PE4 ~ PE8、PE10	Z	I/O	Z	K^{*1}	I/O	I/O	I/O
	PE9、PE11 ~ PE15	Z	I/O	Z	Z	I/O	Z	Z
	PE16 ~ PE21	Z	I/O	Z	Z	I/O	Z	Z
	PF0 ~ PF3、 PF8 ~ PF15	Z	I	Z	Z	I	I	I

【符号说明】

I : 输入

O : 输出

H : 高电平输出

L : 低电平输出

Z : 高阻抗

K : 输入引脚为高阻抗, 输出引脚保持状态。

【注】 *1 如果将待机控制寄存器 6 (STBCR6) 的 HIZ 位置 1, 输出引脚就为高阻抗。

*2 在上电复位中为输入状态。为了防止误动作, 必须进行上拉。当需要下拉时, 必须至少用 1MΩ 的电阻值进行下拉。

*3 当无任何输入时, 在内部进行上拉。

表 A.2 引脚状态 (SH7137)

引脚功能		引脚状态									
分类	引脚名	复位状态				低功耗状态			总线 权释 放状 态	检测 到振 荡停 止	使用 POE 功能
		上电			手动	深度 软件 待机	软件 待机	睡眠			
		无 ROM 扩展	有 ROM	单芯							
		8bit	扩展	片							
时钟	CK	O		Z	O	Z	H* ¹	O	O	O	O
	XTAL	O			O	L	L	O	O	O	O
	EXTAL	I			I	Z	I	I	I	I	I
系统控制	RES	I			I	I	I	I	I	I	I
	MRES	Z			I	Z	Z	I	I	Z	I
	WDTOVF	O* ³			O	O	O	O	O	O	O
	BREQ	Z			I	Z	Z	I	I	I	I
	BACK	Z			O	Z	Z	O	L	O	O

引脚功能		引脚状态									
分类	引脚名	复位状态				低功耗状态			总线 权释 放状 态	检测 到振 荡停 止	使用 POE 功能
		上电			手动	深度 软件 待机	软件 待机	睡眠			
		无 ROM 扩展	有 ROM	单芯							
		8bit	扩展	片							
运行模式 控制	MD0、MD1	I			I	I	I	I	I	I	I
	ASEMD0	I*4			I*4	I*4	I*4	I*4	I*4	I*4	
	FWE	I			I	I	I	I	I	I	
中断	NMI	I			I	I	I	I	I	I	
	IRQ0 ~ IRQ3	Z			I	Z	I	I	I	I	
	IRQOUT	Z			O	Z	Z	O	O	Z	O
地址总线	A0 ~ A17	O	Z		O	Z	Z*2	O	Z	O	O
	A18、A19	Z			O	Z	Z*2	O	Z	O	O
数据总线	D0 ~ D7	Z			I/O	Z	Z	I/O	Z	I/O	I/O
总线控制	WAIT	Z			I	Z	Z	I	Z	I	I
	CS0 (PE10)	H	Z		O	Z	Z*2	O	Z	O	O
	CS0 (PE17)、 CS1 (PE18)	Z			O	Z	Z*2	O	Z	O	O
	RD (PA6)	H	Z		O	Z	Z*2	O	Z	O	O
	RD (PE19)	Z			O	Z	Z*2	O	Z	O	O
	WRL (PA8)	H	Z		O	Z	Z*2	O	Z	O	O
	WRL (PE21)	Z			O	Z	Z*2	O	Z	O	O
MTU2	TCLKA ~ TCLKD	Z			I	Z	Z	I	I	I	I
	TIOC0A ~ TIOC0D	Z			I/O	Z	K*1	I/O	I/O	I/O	Z
	TIOC1A、TIOC1B	Z			I/O	Z	K*1	I/O	I/O	I/O	I/O
	TIOC2A、TIOC2B	Z			I/O	Z	K*1	I/O	I/O	I/O	I/O
	TIOC3A、TIOC3C	Z			I/O	Z	K*1	I/O	I/O	I/O	I/O
	TIOC3B、TIOC3D	Z			I/O	Z	Z	I/O	I/O	Z	Z
	TIOC4A ~ TIOC4D	Z			I/O	Z	Z	I/O	I/O	Z	Z
	TIC5U、TIC5V、 TIC5W	Z			I	Z	Z	I	I	I	I
MTU2S	TIOC3BS、 TIOC3DS	Z			I/O	Z	Z	I/O	I/O	Z	Z
	TIOC4AS ~ TIOC4DS	Z			I/O	Z	Z	I/O	I/O	Z	Z
	TIC5US、 TIC5VS、TIC5WS	Z			I	Z	Z	I	I	I	I
POE	POE0 ~ POE2、 POE4 ~ POE6、 POE8	Z			I	Z	Z	I	I	I	I
SCI	SCK0 ~ SCK2	Z			I/O	Z	Z	I/O	I/O	I/O	I/O
	RXD0 ~ RXD2	Z			I	Z	Z	I	I	I	I
	TXD0 ~ TXD2	Z			O	Z	O*1	O	O	O	O

引脚功能		引脚状态									
分类	引脚名	复位状态				低功耗状态			总线 权释 放状 态	检测 到振 荡停 止	使用 POE 功能
		上电			手动	深度 软件 待机	软件 待机	睡眠			
		无 ROM 扩展	有 ROM	单芯							
		8bit	扩展	片							
SSU	SSCK	Z			I/O	Z	Z	I/O	I/O	I/O	I/O
	$\overline{\text{SCS}}$	Z			I/O	Z	Z	I/O	I/O	I/O	I/O
	SSI	Z			I/O	Z	Z	I/O	I/O	I/O	I/O
	SSO	Z			I/O	Z	Z	I/O	I/O	I/O	I/O
I ² C2	SCL	Z			I/O	Z	Z	I/O	I/O	I/O	I/O
	SDA	Z			I/O	Z	Z	I/O	I/O	I/O	I/O
UBC	$\overline{\text{UBCTRG}}$	Z			O	Z	O* ¹	O	O	O	O
RCAN-ET	CTx0	Z			O	Z	O* ¹	O	O	O	O
	CRx0	Z			I	Z	Z	I	I	I	I
A/D 转换器	AN0 ~ AN15	Z			I	Z	Z	I	I	I	I
	$\overline{\text{ADTRG}}$	Z			I	Z	Z	I	I	I	I
I/O 端口	PA0 ~ PA15	Z			I/O	Z	K* ¹	I/O	I/O	I/O	I/O
	PB0 ~ PB7	Z			I/O	Z	K* ¹	I/O	I/O	I/O	I/O
	PD0 ~ PD10	Z			I/O	Z	K* ¹	I/O	I/O	I/O	I/O
	PE0 ~ PE3	Z			I/O	Z	K* ¹	I/O	I/O	I/O	Z
	PE4 ~ PE8、 PE10	Z			I/O	Z	K* ¹	I/O	I/O	I/O	I/O
	PE9、 PE11 ~ PE15	Z			I/O	Z	Z	I/O	I/O	Z	Z
	PE16 ~ PE21	Z			I/O	Z	Z	I/O	I/O	Z	Z
	PF0 ~ PF15	Z			I	Z	Z	I	I	I	I

【符号说明】

- I : 输入
 O : 输出
 H : 高电平输出
 L : 低电平输出
 Z : 高阻抗
 K : 输入引脚为高阻抗, 输出引脚保持状态。

【注】 *1 如果将待机控制寄存器 6 (STBCR6) 的 HIZ 位置 1, 输出引脚就为高阻抗。

*2 如果将共用控制寄存器 (CMNCR) 的 HIZMEM 位置 1, 此引脚就为输出状态。

*3 在上电复位中为输入状态。为了防止误动作, 必须进行上拉。当需要下拉时, 必须至少用 1MΩ 的电阻值进行下拉。

*4 当无任何输入时, 在内部进行上拉。

附录 B. 有关总线信号的引脚状态

表 B.1 有关总线信号的引脚状态 (1)

引脚名		内部 ROM 空间	内部 RAM 空间	内部外围模块空间
$\overline{\text{CS0}}$ 、 $\overline{\text{CS1}}$		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		High-Z	High-Z	High-Z

【符号说明】

R: 读

W: 写

【注】 * 以前存取的外部空间的地址值。

表 B.2 有关总线信号的引脚状态 (2)

引脚名		外部空间 (通常空间)
		8 位空间
$\overline{\text{CS0}}$ 、 $\overline{\text{CS1}}$		有效
$\overline{\text{RD}}$	R	L
	W	H
$\overline{\text{WRL}}$	R	H
	W	L
A19 ~ A0		地址
D7 ~ D0		数据

【符号说明】

R: 读

W: 写

有效: 对应存取区域的片选信号 =L, 其他的片选信号 =H。

附录 C. 型号一览

表 C.1 型号一览

产品分类						产品型号	封装（封装代码）
产品名	分类	ROM 容量	RAM 容量	用途	工作温度		
SH7136	F-ZTAT 版	256KB	16KB	民用	−20 ~ +85℃	R5F71364AN80FPV	LQFP1414-80 (FP-80WV)
				工业用	−40 ~ +85℃	R5F71364AD80FPV	
SH7137	F-ZTAT 版	256KB	16KB	民用	−20 ~ +85℃	R5F71374AN80FPV	LQFP1414-100 (FP-100UV)
				工业用	−40 ~ +85℃	R5F71374AD80FPV	

附录 D. 封装尺寸图

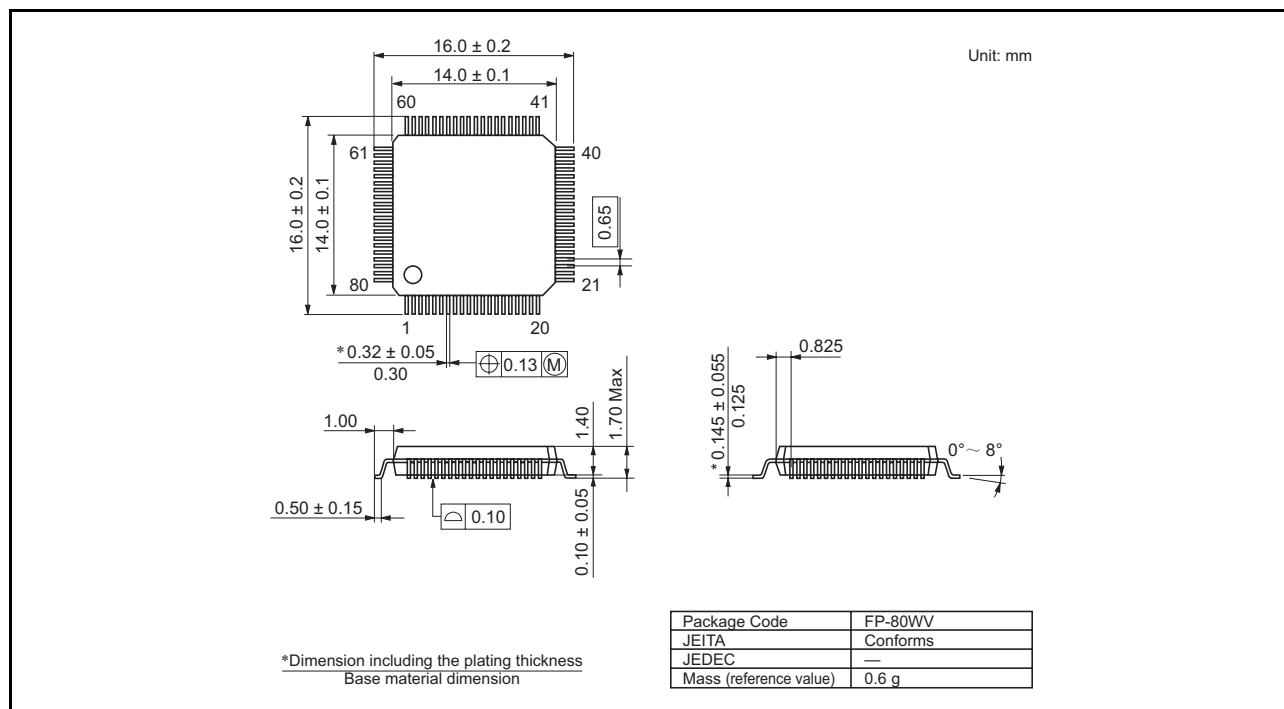


图 D.1 FP-80WV

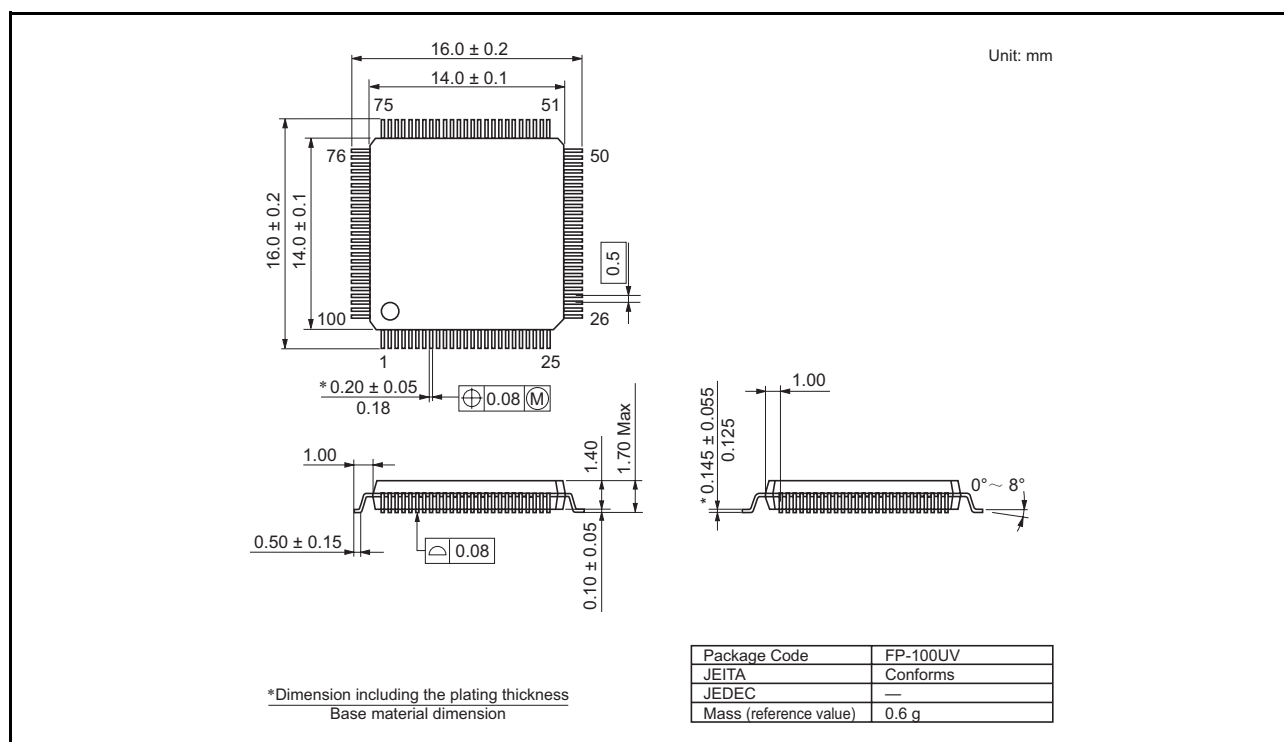


图 D.2 FP-100UV

索引

A

A/D 转换精度的定义	431
A/D 转换开始请求的延迟功能	250
A/D 转换器的启动	260
A/D 转换器特性	684
A/D 转换器 (ADC)	418
A/D 转换时间	429
AC 特性	663
AC 特性的测量条件	684
AC 总线的时序规格	668

B

板上编程模式	563
比较匹配定时器 (CMT)	434
编程器模式	604

C

CAN 接口	443
CAN 睡眠模式	471
CMT 的中断源	438
CPU	12
CSn 有效期间的扩展	134
槽非法指令	55
测试模式的设定	473
乘加寄存器 (MACH 和 MACL)	15
程序计数器 (PC)	15
程序执行状态	32
重复传送模式	109
传送时钟	375
传送信息的分配和 DTC 向量表	102
传送信息的跳读功能	108
传送信息的写回省略功能	108
串行通信接口 (SCI)	326
存取长度和数据调整	130
存取等待的控制	133
存取周期之间的等待	134
错误保护	577

D

DC 特性	659
DTC 的启动	260
DTC 的启动源	102
DTC 的执行状态	113
DTC 的总线权释放时序	114
DTC 接口	479
DTC 向量地址	103
单芯片模式	34
单周期扫描模式	426
低功耗模式	606
低功耗状态	32
地址错误	53, 57, 605

地址映像	122
电路板设计的注意事项	47, 432
断点比较条件	74
端口输出的允许 (POE)	304
多处理器通信功能	357
多功能定时器脉冲单元 2S (MTU2S)	300
多功能定时器脉冲单元 2 (MTU2)	142

F

非线性误差	432
分频器	39
封装尺寸图	692
复位同步 PWM 模式	218
复位状态	32

G

高阻抗控制对象的引脚和条件	317
各处理模式的本 LSI 引脚状态	686
各模块的运行时钟	40
各运行模式的地址映像	34
各运行模式的寄存器状态	648
更改频率的方法	45
更改运行模式时的注意事项	36
过程寄存器 (PR)	15

H

互补 PWM 模式	220
-----------	-----

I

I/O 端口	525
I ² C 总线格式	401
I ² C 总线接口 2 (I ² C2)	390
ID 排列的变更	454
IRQ 中断	66

J

寄存器	
ABACK0	467
ADANSR_0、ADANSR_1	425
ADCR_0、ADCR_1	422
ADDR0 ~ ADDR15	425
ADSR_0、ADSR_1	423
ADSTRGR_0、ADSTRGR_1	424
BAMRA	77
BAMRB	80
BARA	76
BARB	79
BBRA	77

BBRB.....	81	MBIMR0.....	469
BCR0、BCR1.....	456	MCR.....	450
BDMRA.....	79	MRA.....	96
BDMRB.....	81	MRB.....	97
BDRA.....	78	NF2CYC.....	400
BDRB.....	80	OCSR1.....	309
BETR.....	85	OCSR2.....	312
BRCR.....	83	OSCCR.....	45
BRDR.....	86	PACRL1.....	496
BRSR.....	85	PACRL2.....	496
BSCEHR.....	101, 127	PACRL3.....	496
CMCNT.....	437	PACRL4.....	496
CMCOR.....	437	PADRL.....	526
CMCSR.....	436	PAIORL.....	495
CMNCR.....	124	PAPRL.....	527
CMSTR.....	435	PBCRL1.....	505
CRA.....	99	PBCRL2.....	505
CRB.....	99	PBDRL.....	529
CS0BCR、CS1BCR.....	124	PBIORL.....	504
CS0WCR、CS1WCR.....	126	PBPRL.....	531
DAR (DTC).....	98	PDCRL1.....	509
DPFR.....	556	PDCRL2.....	509
DTCCR.....	100	PDCRL3.....	509
DTCERA ~ DTCERE.....	99	PDDRL.....	533
DTCVBR.....	101	PDIORL.....	509
FCCS.....	551	PDPRL.....	534
FEBS.....	561	PECRH1.....	513
FECS.....	553	PECRH2.....	513
FKEY.....	553	PECRL1.....	513
FMATS.....	554	PECRL2.....	513
FMPAR.....	559	PECRL3.....	513
FMPDR.....	559	PECRL4.....	513
FPCS.....	552	PEDRH.....	537
FPEFEQ.....	557	PEDRL.....	537
FPFR.....	558, 560, 561	PEIORH.....	512
FRQCR.....	43	PEIORL.....	512
FTDAR.....	554	PEPRH.....	538
FUBRA.....	557	PEPRL.....	539
GSR.....	454	PFDRL.....	541
ICCR1.....	392	POECR1.....	315
ICCR2.....	394	POECR2.....	315
ICDRR.....	399	RAMCR.....	612
ICDRS.....	400	RAMER.....	563
ICDRT.....	399	REC.....	463
ICIER.....	396	RFPR0.....	468
ICMR.....	395	RXPR0.....	468
ICR0.....	61	SAR (DTC).....	98
ICSR.....	397	SAR (I ² C2).....	399
ICSR1.....	307	SCBRR (SCI).....	337
ICSR2.....	310	SCRDR.....	329
ICSR3.....	313	SCRSR (SCI).....	328
IFCR.....	523	SCSCSR (SCI).....	331
IMR.....	462	SCSDCR.....	336
IPRA、IPRD ~ IPRF、IPRH ~ IPRM.....	64	SCSMR (SCI).....	329
IRQCR.....	61	SCSPTR.....	335
IRQSR.....	62	SCSSR.....	333
IRR.....	459	SCTDR.....	329

SCTSR (SCI)	329
SPOER	314
SSCR2	373
SSCRH	368
SSCRL	369
SSER	371
SSMR	370
SSRDR0 ~ SSRDR3	374
SSSR	371
SSTDRO ~ SSTDR3	373
SSTRSR	375
STBCR1	607
STBCR2	608
STBCR3	608
STBCR4	609
STBCR5	610
STBCR6	611
TADCOBRA_4	179
TADCOBRB_4	179
TADCORA_4	179
TADCORB_4	179
TADCR	177
TBTER	194
TBTM	174
TCBR	191
TCDR	191
TCNT	179
TCNTCMPCLR	167
TCNTS	191
TCR	150
TCSYSTR	182
TDDR	191
TDER	195
TEC	463
TGCR	189
TGR	179
TICCR	175
TIER	167
TIOR	154
TITCNT	193
TITCR	192
TMDR	153
TOCR1	185
TOCR2	186
TOER	184
TOLBR	189
TRWER	183
TSR	170
TSTR	180
TSYCR	175
TSYR	181
TWCR	195
TXACK0	467
TXCR0	466
TXPR1、TXPR0	465
UMSR0	469
WTCNT	322

WTCSR	322
寄存器存取时的注意事项 (WDT)	324
寄存器的数据格式	15
寄存器地址一览 (地址顺序)	617
寄存器位一览	632
寄存器一览	616
间隔定时器模式的用法	325
晶体振荡器	39
局部接收过滤屏蔽 (LAFM)	449
绝对精度	432
绝对最大额定值	658

K

看门狗定时器模式的用法	324
看门狗定时器 (WDT)	321
空载时间的补偿功能	257
控制寄存器的初始值	15
控制器局域网 (RCAN-ET)	441
控制信号的时序	665
块传送模式	110

L

LSI 内部总线主控的存取	138
立即数的数据格式	16
连接晶体谐振器的方法	46
连续扫描模式	427
链传送	111
量化误差	432
逻辑运算指令	29

M

MCU 扩展模式	33
MCU 运行模式	33
MTU2 的功能一览表	143
MTU2 模块的 MTU2 时钟 (MP ϕ)	37
MTU2S 的功能一览表	300
MTU2S 模块的 MTU2S 时钟 (MI ϕ)	37
MTU2 输出引脚的初始化方法	281
MTU2 中断源	259
MTU2 - MTU2S 的同步运行	253
满刻度误差	431
模块待机功能	614
模块待机模式的设定	119, 365, 389, 417, 439, 605

N

NMI 中断	66
NMI 中断和 DTC 启动的竞争	120
内部外围模块中断	66

P

PC 跟踪	89
-------------	----

偏移误差	431
------------	-----

Q

取指令周期的用户断点	87
全局基址寄存器 (GBR)	14

R

RAM	605
RCAN-ET 的存储器映像	444
RCAN-ET 的复位顺序	470
RCAN-ET 的中断源	479
RCAN-ET 位速率	458
RISC 方式	16
软件保护	576
软件待机模式	613

S

SCI 中断源	362
SCSPTR 和 SCI 引脚的关系	362
SSU 模式	378
SSU 中断源	389
闪存	543
闪存结构	546
闪存特性	685
上电复位	52
深度软件待机模式	614
时间量子的定义	456
时钟频率的控制电路	39
时钟时序	663
时钟同步串行格式 (I ² C2)	407
时钟同步模式	326, 350
时钟运行模式	40
时钟振荡器 (CPG)	37
手动复位	52
数据存取周期的用户断点	88
数据传送控制器 (DTC)	94
数据传送指令	26
输入采样和 A/D 转换时间	428
输入外部时钟的方法	46
睡眠模式	613
顺序断点	88
算术运算指令	27

T

停机模式	471
同步串行通信单元 (SSU)	366
通常空间接口	130
通过 MTU2 和 MTU2S 启动 A/D 转换器	430
通过 RAM 对闪存进行的仿真	578
通过中断启动 DTC	117
通用寄存器	14
通用寄存器的初始值	15

V

V _{CL} 电容的连接方法	685
-------------------------------	-----

W

外部触发输入时序	430
外部脉宽的测量功能	256
外围时钟 (Pφ)	37
微处理器接口 (MPI)	443
位同步电路	416

X

系统寄存器的初始值	15
系统控制指令	31
陷阱指令	55
向量基址寄存器 (VBR)	15
信息的发送顺序	474
信息的接收顺序	476
信息控制字段	446
信息数据字段	449
型号一览	691
寻址方式	19

Y

一般非法指令	56
异步模式	326, 344
异常处理	49
异常处理后的堆栈状态	57
异常处理向量表	50, 67
异常处理向量表地址的计算方法	51
异常处理状态	32
移位指令	29
引导模式	564
引脚功能控制器 (PFC)	482
硬件保护	576
用户编程模式	566
用户断点控制器 (UBC)	74
用户断点运行的流程	87
用户断点中断	67
用户分支处理的启动间隔	582
用户分支处理时间	582
用户 MAT	547
用户引导模式	573
有关槽非法指令异常处理的注意事项	58
有关谐振器的注意事项	47
有关总线信号的引脚状态	690
邮箱	443
邮箱的地址映像	445
邮箱的结构	445
邮箱的重新设定	477
邮箱控制	443

Z

噪声对策的注意事项	433
噪声消除电路	410
振荡停止的检测功能	47
正常传送模式	108
指令格式	22
指令特点	16
指令系统	24
中断	54
中断控制器（INTC）	59
中断响应时间	71
中断异常处理结束后的堆栈状态	71
中断优先级	54
中断运行的流程	69
中止的发送	363
中止的检测和处理	363
转移指令	30
状态寄存器（SR）	14
总线权释放状态	32
总线时钟（Bφ）	37
总线仲裁	136
总线状态控制器（BSC）	121

**瑞萨 32 位 RISC 单片机
硬件手册
SH7137 群**

Publication Date: Rev1.00, Jan. 23, 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, AZIA Center, 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



SH7137 群



瑞萨电子株式会社

RCJ09B0068-0100