

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.

SH7125 群、SH7124 群

瑞萨 32 位 RISC 单片机

SuperH™ RISC engine 族

SH7125

R5F7125

SH7124

R5F7124

Notes regarding these materials

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

注意

本文只是参考译文，前页所载英文版“Cautions”具有正式效力。

关于利用本资料时的注意事项

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

有关产品的一般注意事项

1. NC 管脚的处理

【注意】 NC 管脚什么也不要连接。

NC(Non-Connection) 管脚除了不连接内部电路以外，还有用于测试管脚和降低噪声等目的。因此，对于 NC 管脚，请什么也不要连接。

2. 未使用的输入管脚的处理

【注意】 未使用的输入管脚必须固定成高电平或者低电平。

CMOS 产品的输入管脚一般为高阻抗输入。如果将未使用的输入管脚处于开放状态，就可能由于周围噪声的感应而产生中间电平，产生内部浸透电流，引起误动作。因此，对于未使用的输入管脚，请固定成高电平或者低电平。

3. 初始化前的处理

【注意】 接通电源时，产品的状态不定。

从给所有电源管脚外加电压开始，到给复位管脚输入低电平为止，内部电路处于不定状态，并且寄存器的设定和各管脚的输出状态也不定。因此，在进行系统设计时，请采用对策，避免因此不定状态而引起的系统误动作。对于具有复位功能的产品，在接通电源后，请首先执行复位运行。

4. 禁止存取未定义的地址或者保留地址

【注意】 禁止存取未定义的地址或者保留地址。

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

本手册的构成

本手册由以下内容构成：

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

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

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

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

11. 索引

前言

SH7124 群和 SH7125 群是以瑞萨独创的 RISC 方式 CPU 为核心，集聚了系统构成所需外围功能的 RISC 单片机。

对象 本手册是以设计“采用 SH7124 群和 SH7125 群的应用系统”的用户为对象。
使用本手册的读者需要具备电路、逻辑电路以及微型计算机的基础知识。

目的 本手册是以“能让用户理解 SH7124 群和 SH7125 群的硬件功能和电特性”为目的。
关于执行指令的详细内容，已记述在《SH-1/SH-2/SH-DSP Software Manual》中，请对照阅读。

阅读方法

- 希望了解全部功能时
→ 请按照目录的顺序阅读。
本书大致按 CPU、系统控制功能、外围功能、电特性的顺序构成。
- 希望了解详细的 CPU 功能时
→ 请参照《SH-1/SH-2/SH-DSP Series Software Manual》。
- 知道寄存器名，希望了解寄存器的详细功能时
→ 在本手册的后面附有“索引”，请从索引检索页号。
有关地址、位内容及其初始化，汇总在“第 20 章 寄存器一览”中。

范例 寄存器的表示：如串行通信接口等相同或类似的功能存在于多个通道时，使用以下的表示方法：

XXX_N（XXX 为基本寄存器名称，N 为通道号）

位的表示：以左侧为高位、右侧为低位的顺序表示

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划线：xxxx

相关资料一览表 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 SH7124 群和 SH7125 群的用户手册

资料名	资料编号
SH7124 群、SH7125 群 硬件手册	本手册
SH-1/SH-2/SH-DSP Software Manual	REJ09B0171

- 有关开发工具的用户手册

资料名	资料编号
SuperHTM RISC engine C/C++ Compiler, Assembler, Optimizing Linkage Editor Compiler Package V.9.00 User's Manual	REJ10B0105
SuperHTM RISC engine High-Performance Embedded Workshop 3 User'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

目 录

第 1 章	概 要	1
1.1	SH7125 和 SH7124 的特点	1
1.2	框图	4
1.3	管脚排列图	5
1.4	管脚功能	7
第 2 章	CPU	10
2.1	特点	10
2.2	寄存器结构	11
2.2.1	通用寄存器 (Rn)	12
2.2.2	控制寄存器	12
2.2.3	系统寄存器	13
2.2.4	寄存器的初始值	13
2.3	数据格式	14
2.3.1	寄存器的数据格式	14
2.3.2	存储器的数据格式	14
2.3.3	立即数的数据格式	14
2.4	指令特点	15
2.4.1	RISC 方式	15
2.4.2	寻址方式	17
2.4.3	指令格式	20
2.5	指令系统	22
2.5.1	指令系统的分类	22
2.5.2	数据传送指令	25
2.5.3	算术运算指令	26
2.5.4	逻辑运算指令	27
2.5.5	移位指令	28
2.5.6	转移指令	29
2.5.7	系统控制指令	30
2.6	处理状态	31
第 3 章	MCU 运行模式	32
3.1	运行模式的选择	32
3.2	输入 / 输出管脚	32
3.3	各运行模式的说明	32
3.3.1	模式 3 (单芯片模式)	32
3.4	地址映像	33
3.5	本 LSI 的初始状态	35
3.6	更改运行模式时的注意事项	35
第 4 章	时钟振荡器 (CPG)	36
4.1	特点	36
4.2	输入 / 输出管脚	39
4.3	时钟运行模式	39
4.4	寄存器说明	41
4.4.1	频率控制寄存器 (FRQCR)	41
4.4.2	振荡停止检测控制寄存器 (OSCCR)	42
4.5	频率的变更方法	43
4.6	振荡器	43

4.6.1	连接晶体谐振器的方法	43
4.6.2	输入外部时钟的方法	44
4.7	振荡停止检测功能	44
4.8	使用注意事项	45
4.8.1	有关谐振器的注意事项	45
4.8.2	电路板设计的注意事项	45
第 5 章	异常处理	46
5.1	概要	46
5.1.1	异常处理的种类和优先级	46
5.1.2	异常处理的运行	47
5.1.3	异常处理向量表	48
5.2	复位	49
5.2.1	复位的种类	49
5.2.2	加电复位	49
5.2.3	手动复位	50
5.3	地址错误	51
5.3.1	地址错误的发生源	51
5.3.2	地址错误的异常处理	51
5.4	中断	52
5.4.1	中断源	52
5.4.2	中断优先级	52
5.4.3	中断异常处理	53
5.5	由指令引起的异常	53
5.5.1	由指令引起的异常种类	53
5.5.2	陷阱指令	53
5.5.3	槽非法指令	53
5.5.4	一般非法指令	54
5.6	异常处理的接受	54
5.7	异常处理后的堆栈状态	55
5.8	使用注意事项	56
5.8.1	堆栈指针 (SP) 的值	56
5.8.2	向量基址寄存器 (VBR) 的值	56
5.8.3	在地址错误异常处理的堆栈存取中发生的地址错误	56
5.8.4	有关槽非法指令异常处理的注意事项	56
第 6 章	中断控制器 (INTC)	57
6.1	特点	57
6.2	输入 / 输出管脚	58
6.3	寄存器说明	58
6.3.1	中断控制寄存器 0 (ICR0)	59
6.3.2	IRQ 控制寄存器 (IRQCR)	60
6.3.3	IRQ 状态寄存器 (IRQSR)	61
6.3.4	中断优先级寄存器 A ~ F、H ~ M (IPRA ~ IPRF、IPRH ~ IPRM)	64
6.4	中断源	65
6.4.1	外部中断源	65
6.4.2	内部外围模块中断	66
6.4.3	用户断点中断	66
6.5	中断异常处理向量表	66
6.6	运行说明	68
6.6.1	中断运行的流程	68
6.6.2	中断异常处理结束后的堆栈状态	70

6.7	中断响应时间	71
6.8	使用注意事项	71
第 7 章	用户断点控制器 (UBC)	72
7.1	特点	72
7.2	寄存器说明	74
7.2.1	断点地址寄存器 A (BARA)	74
7.2.2	断点地址屏蔽寄存器 A (BAMRA)	75
7.2.3	断点总线周期寄存器 A (BBRA)	76
7.2.4	断点数据寄存器 A (BDRA)	77
7.2.5	断点数据屏蔽寄存器 A (BDMRA)	78
7.2.6	断点地址寄存器 B (BARB)	78
7.2.7	断点地址屏蔽寄存器 B (BAMRB)	79
7.2.8	断点数据寄存器 B (BDRB)	79
7.2.9	断点数据屏蔽寄存器 B (BDMRB)	80
7.2.10	断点总线周期寄存器 B (BBRB)	81
7.2.11	中断控制寄存器 (BRCR)	82
7.2.12	执行次数断点寄存器 (BETR)	84
7.2.13	转移源寄存器 (BRSR)	84
7.2.14	转移目标寄存器 (BRDR)	85
7.3	运行说明	86
7.3.1	用户断点运行的流程	86
7.3.2	取指令周期的断点	86
7.3.3	数据存取周期的断点	87
7.3.4	顺序断点	87
7.3.5	被保存的程序计数器值	88
7.3.6	PC 跟踪	88
7.3.7	使用例	89
7.4	使用注意事项	93
第 8 章	总线状态控制器 (BSC)	94
8.1	特点	94
8.2	地址映像	94
8.3	内部 FLASH 和内部 RAM 的存取	94
8.4	内部外围 I/O 寄存器的存取	95
第 9 章	多功能定时器脉冲单元 2 (MTU2)	97
9.1	特点	97
9.2	输入 / 输出管脚	101
9.3	寄存器的说明	102
9.3.1	定时器的控制寄存器 (TCR)	105
9.3.2	定时器的模式寄存器 (TMDR)	108
9.3.3	定时器的 I/O 控制寄存器 (TIOR)	110
9.3.4	定时器的比较匹配清除寄存器 (TCNTCMPCLR)	128
9.3.5	定时器的中断允许寄存器 (TIER)	129
9.3.6	定时器的状态寄存器 (TSR)	132
9.3.7	定时器的缓冲运行传送模式寄存器 (TBTM)	136
9.3.8	定时器的输入捕捉控制寄存器 (TICCR)	137
9.3.9	定时器的 A/D 转换开始请求的控制寄存器 (TADCR)	138
9.3.10	定时器的 A/D 转换开始请求的周期设定寄存器 (TADCORA/B_4)	140
9.3.11	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)	140
9.3.12	定时器的计数器 (TCNT)	140

9.3.13	定时器的通用寄存器 (TGR)	141
9.3.14	定时器的开始寄存器 (TSTR)	141
9.3.15	定时器的同步寄存器 (TSYR)	143
9.3.16	定时器的计数器同步开始寄存器 (TCSYSTR)	144
9.3.17	定时器的读写允许寄存器 (TRWER)	145
9.3.18	定时器的输出主控允许寄存器 (TOER)	146
9.3.19	定时器的输出控制寄存器 1 (TOCR1)	147
9.3.20	定时器的输出控制寄存器 2 (TOCR2)	149
9.3.21	定时器的输出电平缓冲寄存器 (TOLBR)	152
9.3.22	定时器的门控寄存器 (TGCR)	153
9.3.23	定时器的子计数器 (TCNTS)	154
9.3.24	定时器的空载时间数据寄存器 (TDDR)	154
9.3.25	定时器的周期数据寄存器 (TCDR)	155
9.3.26	定时器的周期缓冲寄存器 (TCBR)	155
9.3.27	定时器的跳过中断设定寄存器 (TITCR)	155
9.3.28	定时器的跳过中断次数计数器 (TITCNT)	157
9.3.29	定时器的缓冲传送设定寄存器 (TBTER)	158
9.3.30	定时器的空载时间允许寄存器 (TDER)	159
9.3.31	定时器的波形控制寄存器 (TWCR)	160
9.3.32	和总线主控的接口	160
9.4	运行说明	161
9.4.1	基本运行	161
9.4.2	同步运行	167
9.4.3	缓冲运行	169
9.4.4	级联运行	173
9.4.5	PWM 模式	177
9.4.6	相位计数模式	181
9.4.7	复位同步 PWM 模式	187
9.4.8	互补 PWM 模式	190
9.4.9	A/D 转换开始请求的延迟功能	216
9.4.10	外部脉宽的测量功能	219
9.4.11	空载时间的补偿功能	220
9.4.12	互补 PWM 在“波峰 / 波谷”的 TCNT 捕捉运行	221
9.5	中断源	222
9.5.1	中断源和优先级	222
9.5.2	A/D 转换器的启动	223
9.6	运行时序	225
9.6.1	输入 / 输出时序	225
9.6.2	中断信号的时序	231
9.7	使用注意事项	235
9.7.1	模块待机模式的设定	235
9.7.2	输入时钟的限制事项	235
9.7.3	周期设定的注意事项	235
9.7.4	TCNT 的写操作和清除的竞争	236
9.7.5	TCNT 的写操作和递增计数的竞争	236
9.7.6	TGR 的写操作和比较匹配的竞争	237
9.7.7	缓冲寄存器的写操作和比较匹配的竞争	238
9.7.8	缓冲寄存器的写操作和 TCNT 清除的竞争	239
9.7.9	TGR 的读操作和输入捕捉的竞争	240
9.7.10	TGR 的写操作和输入捕捉的竞争	241
9.7.11	缓冲寄存器的写操作和输入捕捉的竞争	242
9.7.12	级联中 TCNT_2 的写操作和上溢 / 下溢的竞争	243

9.7.13	互补 PWM 模式停止时的计数器值	244
9.7.14	互补 PWM 模式中缓冲运行的设定	244
9.7.15	复位同步 PWM 模式的缓冲运行和比较匹配标志	245
9.7.16	复位同步 PWM 模式的上溢标志	246
9.7.17	上溢 / 下溢和计数器清除的竞争	246
9.7.18	TCNT 的写操作和上溢 / 下溢的竞争	247
9.7.19	从通常运行或者 PWM 模式 1 转移到复位同步 PWM 模式时的注意事项	247
9.7.20	互补 PWM 模式或者复位同步 PWM 模式的输出电平	247
9.7.21	模块待机时的中断	247
9.7.22	级联中 TCNT_1 和 TCNT_2 的同时输入捕捉	247
9.8	MTU2 输出管脚的初始化方法	248
9.8.1	运行模式	248
9.8.2	开始复位时的运行	248
9.8.3	因运行中的异常等而重新设定时的运行	248
9.8.4	因运行中的异常等而初始化管脚的步骤和模式转移的概要	249
第 10 章	端口输出的允许 (POE)	279
10.1	特点	279
10.2	输入 / 输出管脚	281
10.3	寄存器说明	282
10.3.1	输入电平控制 / 状态寄存器 1 (ICSR1)	282
10.3.2	输出电平控制 / 状态寄存器 1 (OCSR1)	284
10.3.3	输入电平控制 / 状态寄存器 3 (ICSR3)	285
10.3.4	软件端口的输出允许寄存器 (SPOER)	286
10.3.5	端口的输出允许控制寄存器 1 (POECR1)	287
10.3.6	端口的输出允许控制寄存器 2 (POECR2)	288
10.4	运行说明	289
10.4.1	输入电平的检测	290
10.4.2	输出电平的比较	291
10.4.3	高阻抗状态的解除	291
10.5	中断	291
第 11 章	看门狗定时器 (WDT)	292
11.1	特点	292
11.2	输入 / 输出管脚	293
11.3	寄存器说明	293
11.3.1	看门狗定时器的计数器 (WTCNT)	293
11.3.2	看门狗定时器的控制 / 状态寄存器 (WTCR)	294
11.3.3	寄存器存取时的注意事项	296
11.4	运行说明	297
11.4.1	软件待机模式的解除步骤	297
11.4.2	看门狗定时器模式的用法	297
11.4.3	间隔定时器模式的用法	298
11.5	使用注意事项	298
第 12 章	串行通信接口 (SCI)	299
12.1	特点	299
12.2	输入 / 输出管脚	300
12.3	寄存器说明	301
12.3.1	接收移位寄存器 (SCSR)	302
12.3.2	接收数据寄存器 (SCRDR)	302
12.3.3	发送移位寄存器 (SCTSR)	302

12.3.4	发送数据寄存器 (SCTDR)	302
12.3.5	串行模式寄存器 (SCSMR)	303
12.3.6	串行控制寄存器 (SCSCR)	305
12.3.7	串行状态寄存器 (SCSSR)	307
12.3.8	串行端口寄存器 (SCSPTR)	309
12.3.9	串行方向控制寄存器 (SCSDCR)	310
12.3.10	位速率寄存器 (SCBRR)	311
12.4	运行说明	317
12.4.1	概要	317
12.4.2	异步模式中的运行	318
12.4.3	时钟同步模式中的运行 (不能使用 SH7124 的通道 1)	326
12.4.4	多处理器通信功能	333
12.4.5	多处理器串行数据的发送	334
12.4.6	多处理器串行数据的接收	335
12.5	SCI 的中断源	338
12.6	串行端口寄存器 (SCSPTR) 和 SCI 管脚的关系	339
12.7	使用注意事项	340
12.7.1	有关 SCTDR 的写操作和 TDRE 标志的关系	340
12.7.2	有关同时发生多个接收错误时的运行	340
12.7.3	有关中止的检测和处理	340
12.7.4	中止的发送	340
12.7.5	异步模式的接收数据采样时序和接收容限	341
12.7.6	时钟同步外部时钟模式的注意事项	341
12.7.7	模块待机模式的设定	341
第 13 章	A/D 转换器 (ADC)	342
13.1	特点	342
13.2	输入 / 输出管脚	344
13.3	寄存器说明	344
13.3.1	A/D 数据寄存器 0 ~ 7 (ADDR0 ~ ADDR7)	345
13.3.2	A/D 控制 / 状态寄存器 _0、1 (ADCSR_0、1)	345
13.3.3	A/D 控制寄存器 _0、1 (ADCR_0、1)	347
13.3.4	A/D 触发选择寄存器 _0 (ADTSR_0)	349
13.4	运行说明	351
13.4.1	单通道模式	351
13.4.2	连续扫描模式	351
13.4.3	单周期扫描模式	352
13.4.4	输入采样和 A/D 转换时间	353
13.4.5	通过 MTU2 启动 A/D 转换器	354
13.4.6	外部触发的输入时序	355
13.5	中断源	355
13.6	A/D 转换精度的定义	356
13.7	使用注意事项	357
13.7.1	模块待机模式的设定	357
13.7.2	关于容许信号源阻抗	357
13.7.3	关于对绝对精度的影响	358
13.7.4	模拟电源管脚等的设定范围	358
13.7.5	电路板设计的注意事项	358
13.7.6	噪声对策的注意事项	359
第 14 章	比较匹配定时器 (CMT)	360
14.1	特点	360

14.2	寄存器说明	361
14.2.1	比较匹配定时器的开始寄存器 (CMSTR)	361
14.2.2	比较匹配定时器的控制 / 状态寄存器 (CMCSR)	362
14.2.3	比较匹配计数器 (CMCNT)	363
14.2.4	比较匹配常数寄存器 (CMCOR)	363
14.3	运行说明	364
14.3.1	周期计数运行	364
14.3.2	CMCNT 的计数时序	364
14.4	中断	365
14.4.1	CMT 的中断源	365
14.4.2	比较匹配标志的置位时序	365
14.4.3	比较匹配标志的清除时序	365
14.5	使用注意事项	366
14.5.1	模块待机模式的设定	366
14.5.2	CMCNT 的写操作和比较匹配的竞争	366
14.5.3	CMCNT 的字写和递增计数的竞争	366
14.5.4	CMCNT 的字节写和递增计数的竞争	367
14.5.5	CMCNT 和 CMCOR 的比较匹配	367
第 15 章	管脚功能控制器 (PFC)	368
15.1	寄存器说明	375
15.1.1	端口 A 的 IO 寄存器 L (PAIORL)	375
15.1.2	端口 A 的控制寄存器 L1 ~ L4 (PACRL1 ~ PACRL4)	376
15.1.3	端口 B 的 IO 寄存器 L、H (PBIORL、PBIORH)	383
15.1.4	端口 B 的控制寄存器 L1、L2、H1 (PBCRL1、PBCRL2、PBCRH1)	384
15.1.5	端口 E 的 IO 寄存器 L (PEIORL)	387
15.1.6	端口 E 的控制寄存器 L1 ~ L4 (PECRL1 ~ PECRL4)	388
15.1.7	IRQOUT 功能控制寄存器 (IFCR)	395
15.2	使用注意事项	396
第 16 章	I/O 端口	397
16.1	端口 A	397
16.1.1	寄存器说明	398
16.1.2	端口 A 的数据寄存器 L (PADRL)	398
16.1.3	端口 A 的端口寄存器 L (PAPRL)	400
16.2	端口 B	402
16.2.1	寄存器说明	402
16.2.2	端口 B 的数据寄存器 H、L (PBDRH、PBDRL)	403
16.2.3	端口 B 的端口寄存器 H、L (PBPRH、PBPRL)	405
16.3	端口 E	407
16.3.1	寄存器说明	408
16.3.2	端口 E 的数据寄存器 L (PEDRL)	408
16.3.3	端口 E 的端口寄存器 L (PEPRL)	410
16.4	端口 F	412
16.4.1	寄存器说明	412
16.4.2	端口 F 的数据寄存器 L (PFDRL)	413
第 17 章	闪存	414
17.1	特点	414
17.2	概要	415
17.2.1	框图	415
17.2.2	运行模式	415

17.2.3	模式比较	416
17.2.4	闪存结构	417
17.2.5	块分割	417
17.2.6	编程 / 擦除的接口	418
17.3	管脚结构	419
17.4	寄存器结构	420
17.4.1	寄存器一览表	420
17.4.2	编程 / 擦除的接口寄存器	421
17.4.3	编程 / 擦除的接口参数	425
17.5	板上编程模式	434
17.5.1	引导模式	434
17.5.2	用户编程模式	437
17.6	保护	444
17.6.1	硬件保护	444
17.6.2	软件保护	444
17.6.3	错误保护	445
17.7	使用注意事项	446
17.7.1	执行编程 / 擦除过程中的中断	446
17.7.2	其他注意事项	447
17.8	附录	448
17.8.1	引导模式的标准串行通信接口规格	448
17.8.2	过程程序或者编程数据的可保存区	465
17.9	编程器模式	467
第 18 章	RAM	468
18.1	使用注意事项	468
18.1.1	模块待机模式的设定	468
18.1.2	地址错误	468
第 19 章	低功耗模式	469
19.1	特点	469
19.1.1	低功耗模式的种类	469
19.2	输入 / 输出管脚	469
19.3	寄存器说明	470
19.3.1	待机控制寄存器 1 (STBCR1)	470
19.3.2	待机控制寄存器 2 (STBCR2)	471
19.3.3	待机控制寄存器 3 (STBCR3)	472
19.3.4	待机控制寄存器 4 (STBCR4)	473
19.3.5	待机控制寄存器 5 (STBCR5)	474
19.3.6	待机控制寄存器 6 (STBCR6)	475
19.3.7	RAM 控制寄存器 (RAMCR)	476
19.4	睡眠模式	477
19.4.1	向睡眠模式的转移	477
19.4.2	睡眠模式的解除	477
19.5	软件待机模式	477
19.5.1	向软件待机模式的转移	477
19.5.2	软件待机模式的解除	478
19.6	模块待机功能	478
19.6.1	向模块待机功能的转移	478
19.6.2	模块待机功能的解除	478
19.7	使用注意事项	478
19.7.1	振荡稳定待机中的功耗	478

第 20 章 寄存器一览	479
20.1 寄存器地址一览表（地址顺序）	480
20.2 寄存器位一览	486
20.3 各运行模式的寄存器状态	497
第 21 章 电特性	502
21.1 绝对最大额定值	502
21.2 DC 特性	503
21.3 AC 特性	505
21.3.1 时钟时序	505
21.3.2 控制信号的时序	507
21.3.3 多功能定时器脉冲单元 2（MTU2）时序	509
21.3.4 I/O 端口的时序	510
21.3.5 看门狗定时器（WDT）的时序	511
21.3.6 串行通信接口（SCI）的时序	512
21.3.7 端口输出允许（POE）的时序	514
21.3.8 A/D 转换器的时序	515
21.3.9 AC 特性测量条件	516
21.4 A/D 转换器特性	517
21.5 闪存特性	518
21.6 使用注意事项	519
21.6.1 VCL 电容连接方法	519
附录	520
附录 A.管脚状态	520
附录 B.型号一览	523
附录 C.外形尺寸图	524
索引	527

第 1 章 概要

1.1 SH7125 和 SH7124 的特点

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

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

作为系统构成上需要的外围功能，本 LSI 内置 ROM、RAM、定时器、串行通信接口（SCI）、A/D 转换器、中断控制器（INTC）和 I/O 端口等。

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

本 LSI 的特点如表 1.1 所示。

【注】* F-ZTAT™ 株式会社瑞萨科技的商标。

表 1.1 SH7125 和 SH7124 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 位提高代码效率 • 装入 - 存储结构 (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 • 内置乘法器: 以 2 ~ 5 周期执行 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】 请注意: 槽非法指令规格和以前的 SH-2 有所不同。详细内容请参照 “5.8.4, 有关槽非法指令异常处理的注意事项”。
运行模式	• 运行模式 单芯片模式 • 处理状态 程序执行状态 异常处理状态 • 低功耗状态 睡眠模式 软件待机模式 模块待机模式
用户断点控制器 (UBC)	• 能将地址、数据值、存取类型和数据大小全部设定为断点条件 • 支持顺序断点功能 • 2 个断点通道
内部 ROM	• 128K 字节 (SH71253、SH71243) • 64K 字节 (SH71252、SH71242)
内部 RAM	• 8K 字节
中断控制器 (INTC)	• 外部中断管脚 SH7125: 5 个 (NMI、IRQ3 ~ IRQ0) SH7124: 4 个 (NMI、IRQ3 ~ IRQ1) • 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: 对输入时钟可选择外部输入或者晶体谐振器 • 生成 4 种时钟 CPU 时钟: 50MHz (Max.) 总线时钟: 40MHz (Max.) 外围时钟: 40MHz (Max.) MTU2 专用时钟: 40MHz (Max.)
看门狗定时器 (WDT)	• 1 个通道的看门狗定时器 • 能请求中断

项目	特 点
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 (SH7125) • 以 6 个通道的 16 位定时器为基础，最多能输入 / 输出 12 种脉冲以及输入 3 个脉冲 (SH7124) • 21 个输出比较匹配寄存器兼输入捕捉寄存器 • 共有 21 个独立的比较匹配电路 • 能选择 8 种计数器输入时钟 • 输入捕捉功能 • 脉冲输出模式 单触发 / 交替 / PWM / 互补 PWM / 复位同步 PWM • 多个计数器的同步功能 • 互补 PWM 输出模式 输出用于控制 6 相反相器的非重叠波形 空载时间的自动设定 可在 0 ~ 100% 范围内任意设定 PWM 占空比 输出 OFF 功能 A/D 转换请求的延迟功能 空载时间的补偿功能 波峰 / 波谷的跳过中断功能 • 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 能处理 2 相编码器的计数
端口输出的允许 (POE)	• 对 MTU2 波形输出管脚和 MTU2 的通道 0 管脚进行高阻抗控制
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 / 异步方式 • 3 个通道
A/D 转换器 (ADC)	• 10 位 × 8 个通道 • 能通过外部触发或者 MTU2 请求转换 • 内置 2 个单元的采样 & 保持功能 (能同时进行 2 个通道的采样)
I/O 端口	• 37 个通用输入 / 输出管脚和 8 个通用输入管脚 (SH7125) • 23 个通用输入 / 输出管脚和 8 个通用输入管脚 (SH7124) • 输入 / 输出兼用端口能按位进行输入 / 输出的转换
封装	• QFP1414-64 (0.8 节距) (SH7125) • LQFP1010-64 (0.5 节距) (SH7125) • LQFP1010-48 (0.65 节距) (SH7124)
电源电压	Vcc: 4.0 ~ 5.5V, AVcc: 4.0 ~ 5.5V

1.2 框图

SH7125 和 SH7124 的框图如图 1.1 所示。

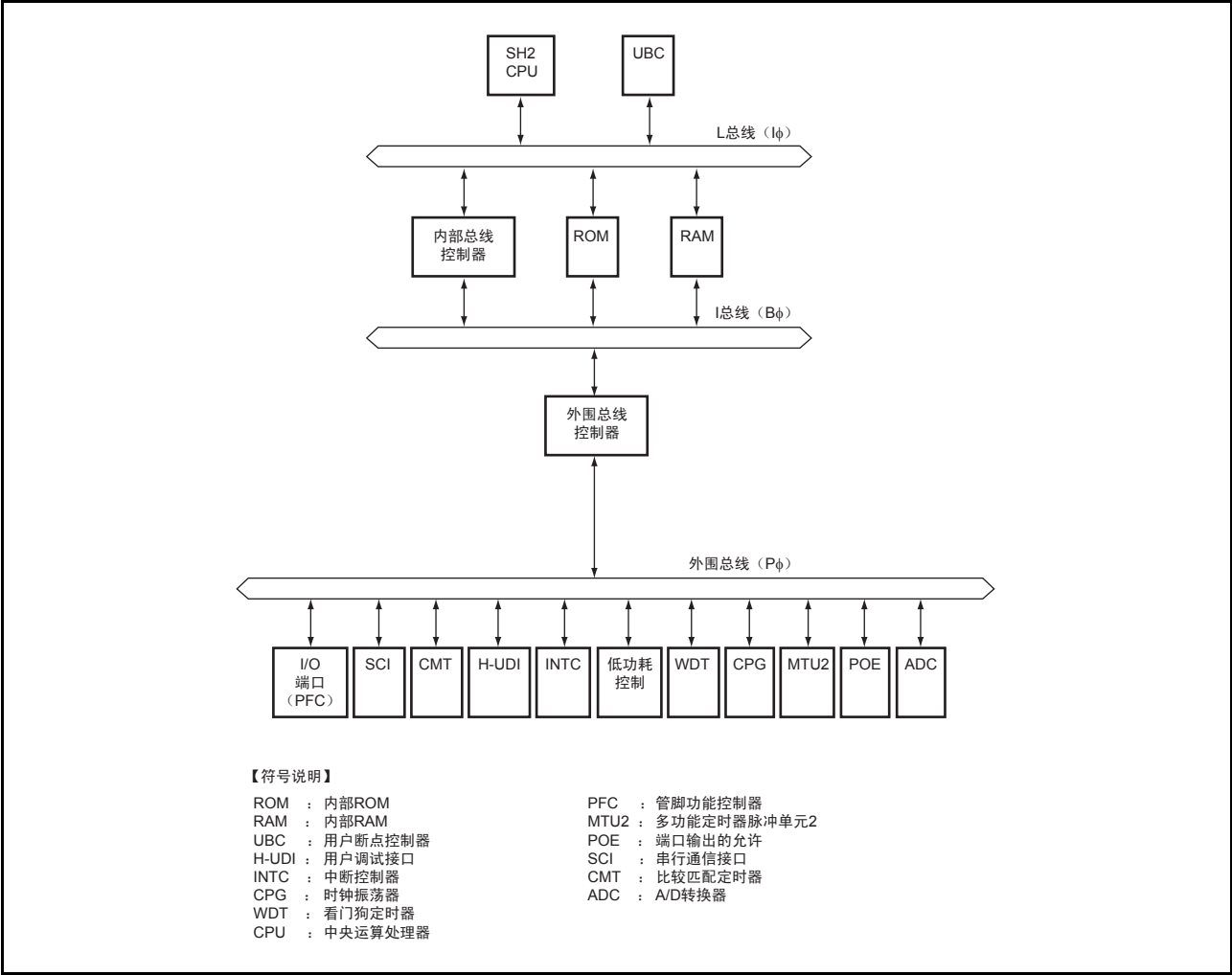


图 1.1 SH7125 和 SH7124 的框图

1.3 管脚排列图

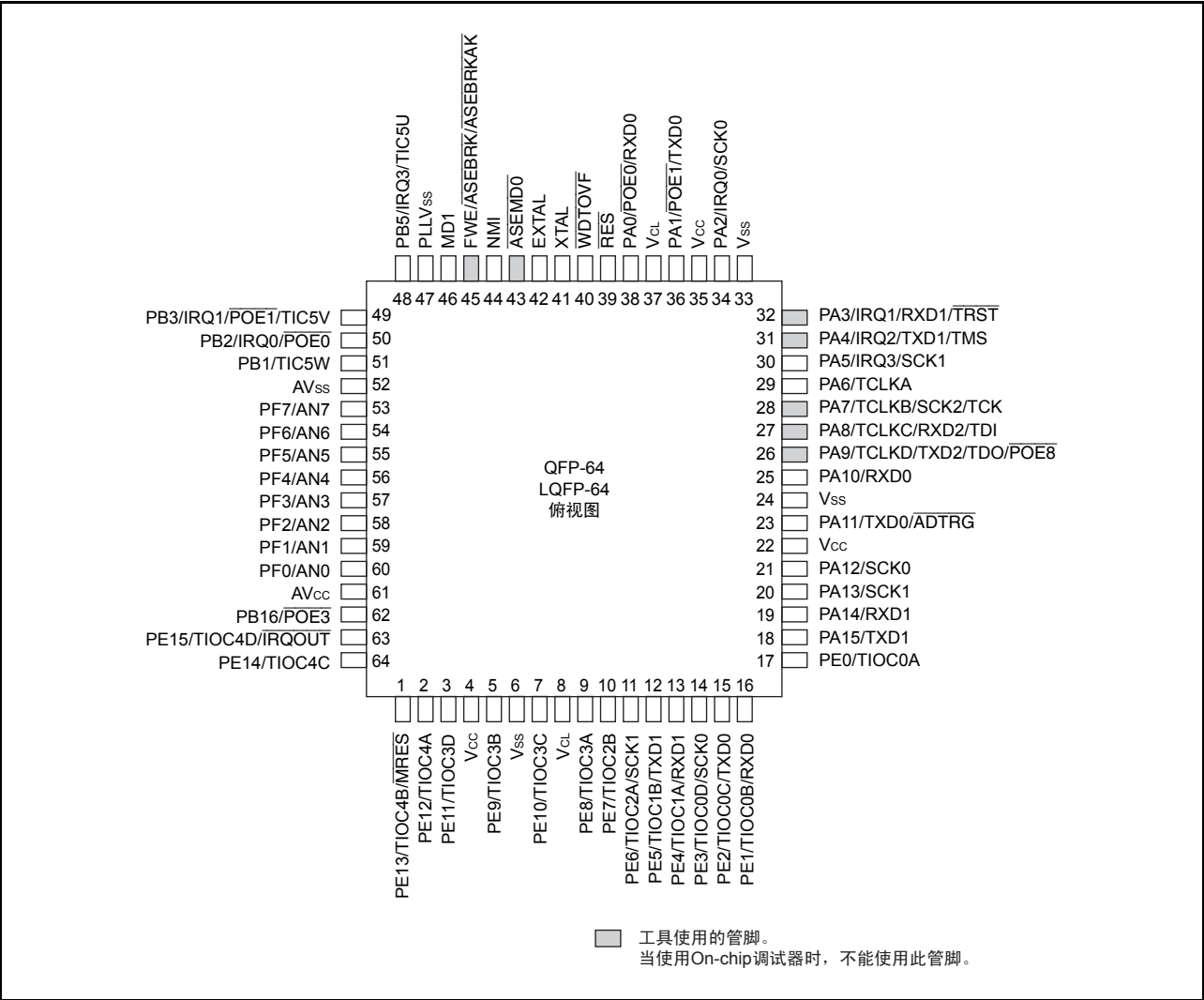


图 1.2 SH7125 的管脚排列图

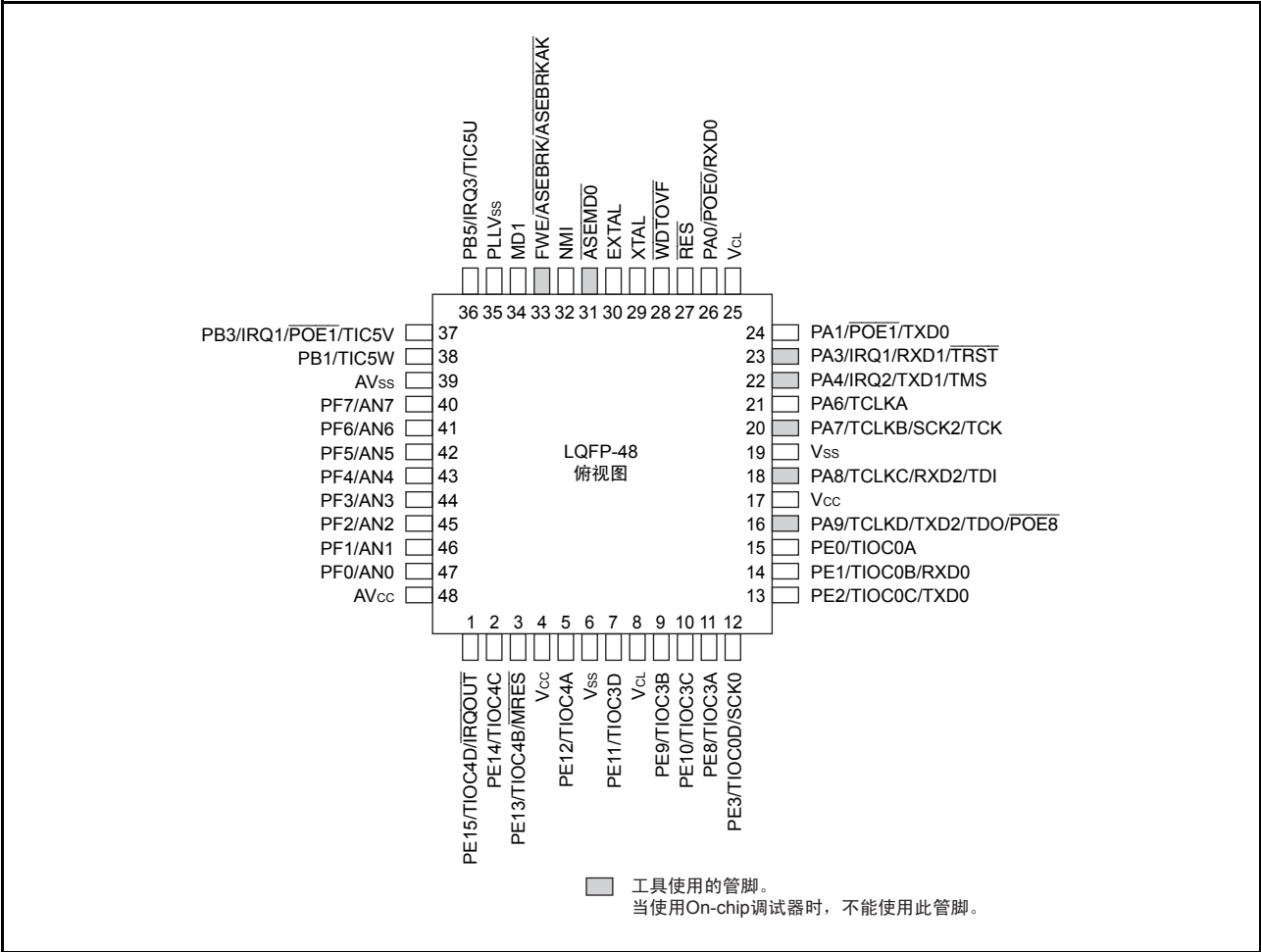


图 1.3 SH7124 的管脚排列图

1.4 管脚功能

各管脚的功能如表 1.2 所示。

表 1.2 管脚功能

分类	符号	输入 / 输出	名称	功 能
电源	VCC	输入	电源	是电源管脚。必须将全部 VCC 管脚连接到系统电源。如果有开路管脚，就不运行。
	VSS	输入	接地	是接地管脚。必须将全部 VSS 管脚连接到系统电源 (0V)。如果有开路管脚，就不运行。
	VCL	输出	内部降压电源	是用于内部降压电源的外接电容管脚。必须通过 0.1 ~ 0.47 μ F 的电容器将全部的 VCL 管脚连接到 VSS (靠近管脚)。
时钟	PLL $\overline{\text{VSS}}$	输入	用于 PLL 的接地	是用于内部 PLL 振荡器的接地管脚。
	EXTAL	输入	外部时钟	连接晶体谐振器，也能输入外部时钟。
	XTAL	输出	晶体	连接晶体谐振器。
运行模式的控制	MD1	输入	模式设定	设定运行模式。请不要在运行中改变此管脚。
	$\overline{\text{FWE}}$	输入	允许闪存的编程	是用于闪存的管脚。能防止闪存被编程 / 擦除。
系统控制	$\overline{\text{RES}}$	输入	加电复位	当此管脚为低电平时，处于加电复位状态。
	$\overline{\text{MRES}}$	输入	手动复位	当此管脚为低电平时，处于手动复位状态。
	$\overline{\text{WDTOVF}}$	输出	看门狗定时器的上溢	是来自 WDT 的上溢输出信号。 如果需要下拉，就必须使用大于等于 1M Ω 的电阻。
中断	NMI	输入	非屏蔽中断	是非屏蔽的中断请求管脚。如果不使用，就必须固定为高电平或者低电平。
	IRQ3 ~ IRQ0 (SH7125) IRQ3 ~ IRQ1 (SH7124)	输入	中断请求 3 ~ 0	是可屏蔽的中断请求管脚。 可选择电平输入或者边沿输入。在选择边沿输入时，可选择上升沿、下降沿或者双边沿。
	$\overline{\text{IRQOUT}}$	输出	中断请求的输出	表示已发生中断源。也能在总线释放时得知中断的发生。

分类	符号	输入 / 输出	名称	功 能
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管脚。
	TIOC0A、 TIOC0B、 TIOC0C、 TIOC0D	输入 / 输出	MTU2 的输入捕 捉 / 输出比较 (通道 0)	是 TGRA_0 ~ TGRD_0 的输入捕捉的输入管脚 / 输出比较的输出管脚 / PWM 的输出管脚。
	TIOC1A、 TIOC1B (仅限 SH7125)	输入 / 输出	MTU2 的输入捕 捉 / 输出比较 (通道 1)	是 TGRA_1、TGRB_1 的输入捕捉的输入管脚 / 输出比较的输出管脚 / PWM 的输出管脚。
	TIOC2A、 TIOC2B (仅限 SH7125)	输入 / 输出	MTU2 的输入捕 捉 / 输出比较 (通道 2)	是 TGRA_2、TGRB_2 的输入捕捉的输入管脚 / 输出比较的输出管脚 / PWM 的输出管脚。
	TIOC3A、 TIOC3B、 TIOC3C、 TIOC3D	输入 / 输出	MTU2 的输入捕 捉 / 输出比较 (通道 3)	是 TGRA_3 ~ TGRD_3 的输入捕捉的输入管脚 / 输出比较的输出管脚 / PWM 的输出管脚。
	TIOC4A、 TIOC4B、 TIOC4C、 TIOC4D	输入 / 输出	MTU2 的输入捕 捉 / 输出比较 (通道 4)	是 TGRA_4 ~ TGRD_4 的输入捕捉的输入管脚 / 输出比较的输出管脚 / PWM 的输出管脚。
	TIC5U、 TIC5V、 TIC5W	输入	MTU2 的输入捕 捉 (通道 5)	是 TGRU_5、TGRV_5、TGRW_5 的输入捕捉的输入管脚。
端口输出的允许 (POE)	$\overline{\text{POE8}}$ 、 $\overline{\text{POE3}}$ 、 $\overline{\text{POE1}}$ 、 $\overline{\text{POE0}}$ (SH7125) $\overline{\text{POE8}}$ 、 $\overline{\text{POE1}}$ 、 $\overline{\text{POE0}}$ (SH7124)	输入	端口输出的控制	是将 MTU2 波形输出管脚和 MTU2 的通道 0 管脚变为高阻抗状态的请求信号的输入管脚。 在 SH7125 的情况下, 当通过 PFC 选择了 $\overline{\text{POE3}}$ 功能时, 如果管脚没有任何输入, 就在内部进行上拉。
串行通信接口 (SCI)	TXD2 ~ TXD0	输出	发送数据	是用于发送数据的管脚。
	RXD2 ~ RXD0	输入	接收数据	是用于接收数据的管脚。
	SCK2 ~ SCK0 (SH7125) SCK2、SCK0 (SH7124)	输入 / 输出	串行时钟	是时钟的输入 / 输出管脚。

分类	符号	输入 / 输出	名称	功 能
A/D 转换器 (ADC)	AN7 ~ AN0	输入	模拟输入管脚	是模拟输入管脚。
	ADTRG (只限 SH7125)	输入	A/D 转换的 触发输入	是用于 A/D 转换开始的外部触发输入管脚。
	AVcc	输入	模拟电源	是 A/D 转换器的电源管脚。如果不使用 A/D 转换器，必须连接到系统电源 (Vcc)。必须将全部 AVcc 管脚连接到系统电源 (Vcc)。如果有开路管脚，就不运行。
	AVss	输入	模拟接地	是 A/D 转换器的接地管脚，必须连接到系统电源 (0V)。必须将全部 AVss 管脚连接到系统电源 (0V)。如果有开路管脚，就不运行。
I/O 端口	PA15 ~ PA0 (SH7125)	输入 / 输出	通用端口	是 16 位输入 / 输出端口的管脚。
	PA9 ~ PA6、 PA4、PA3、 PA1、PA0 (SH7124)			是 8 位输入 / 输出端口的管脚。
	PB16、PB5、 PB3 ~ PB1 (SH7125)	输入 / 输出	通用端口	是 5 位输入 / 输出端口的管脚。
	PB5、PB3、PB1 (SH7124)			是 3 位输入 / 输出端口的管脚。
	PE15 ~ PE0 (SH7125)	输入 / 输出	通用端口	是 16 位输入 / 输出端口的管脚。
	PE15 ~ PE8、 PE3 ~ PE0 (SH7124)			是 12 位输入 / 输出端口的管脚。
	PF7 ~ PF0	输入	通用端口	是 8 位输入端口的管脚。
用户调试接口 (H-UDI)	TCK	输入	测试时钟	是测试时钟的输入管脚。
	TMS	输入	测试模式的选择	是测试模式选择信号的输入管脚。
	TDI	输入	测试数据的输入	是指令和数据的串行输入管脚。
	TDO	输出	测试数据的输出	是指令和数据的串行输出管脚。
	$\overline{\text{TRST}}$	输入	测试复位	是初始化信号的输入管脚。
E10A 接口	ASEMD0	输入	ASE 模式	设定 ASE 模式。 当给本管脚输入低电平时，为 ASE 模式；当输入高电平时，为通常模式。在 ASE 模式中，能使用仿真器专用功能。如果管脚没有任何输入，就在内部进行上拉。
	ASEBRK	输入	断点请求	是 E10A 仿真器的断点输入。
	ASEBRKAK	输出	断点模式应答	表示 E10A 仿真器已进入断点模式。

【使用注意事项】

请不要下拉 $\overline{\text{WDT0VF}}$ 管脚。如果需要下拉，必须用大于等于 1M Ω 的电阻进行下拉。

第 2 章 CPU

2.1 特点

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

2.2 寄存器结构

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

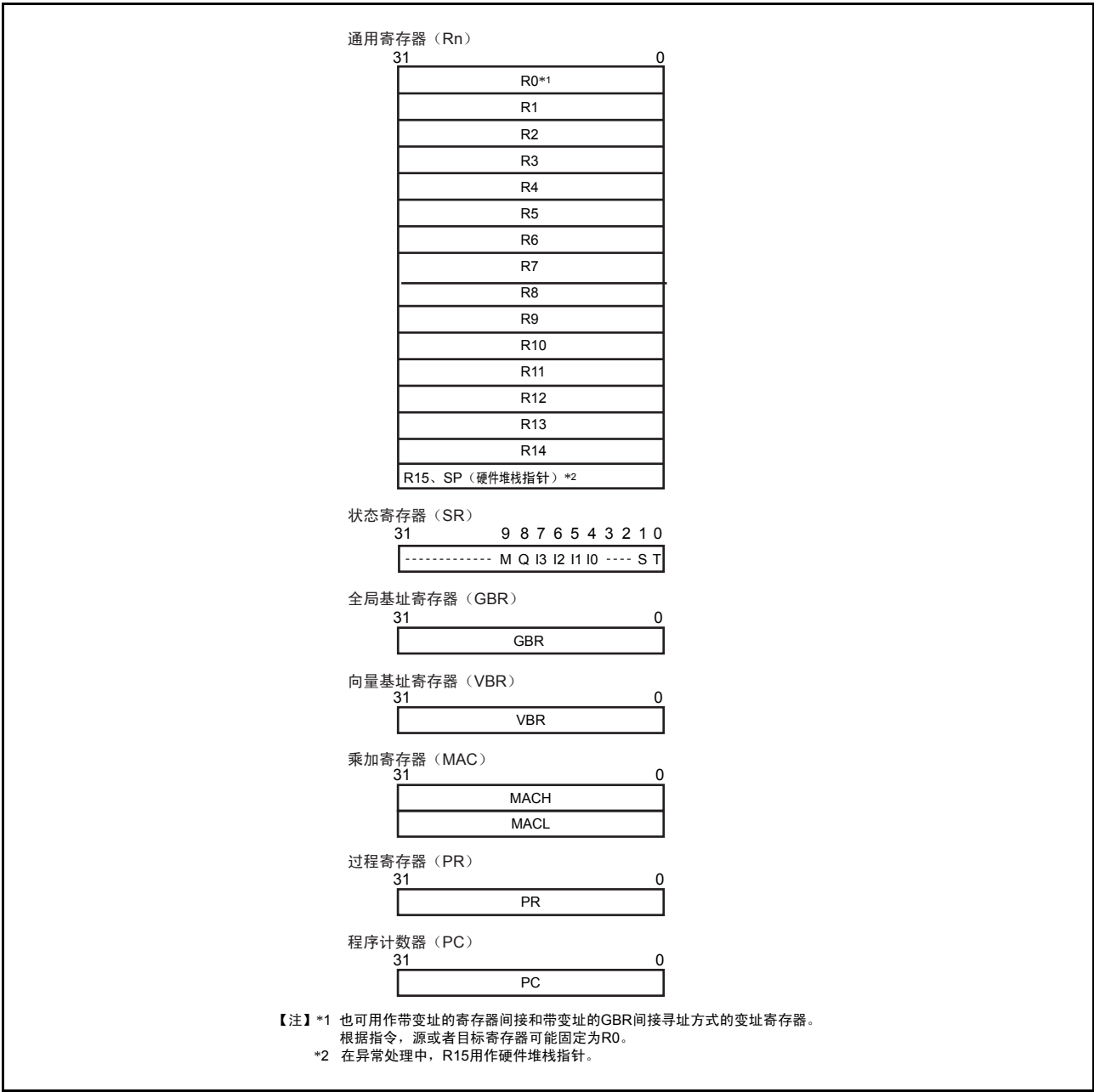


图 2.1 CPU 内部寄存器结构

2.2.1 通用寄存器 (Rn)

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

2.2.2 控制寄存器

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

(1) 状态寄存器 (SR)

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

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

(2) 全局基址寄存器 (GBR)

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

(3) 向量基址寄存器 (VBR)

表示异常处理向量区的基地址。

2.2.3 系统寄存器

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

(1) 乘加寄存器 (MACH 和 MACL)

这是保存乘法运算和乘加运算结果的寄存器。

(2) 过程寄存器 (PR)

这是保存子程序过程返回地址的寄存器。

(3) 程序计数器 (PC)

PC 表示当前执行指令的 4 字节 (2 条指令) 后的地址。

2.2.4 寄存器的初始值

复位后的寄存器值如表 2.1 所示。

表 2.1 寄存器的初始值

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

2.3 数据格式

2.3.1 寄存器的数据格式

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



图 2.2 寄存器的数据格式

2.3.2 存储器的数据格式

有字节、字和长字数据格式。

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

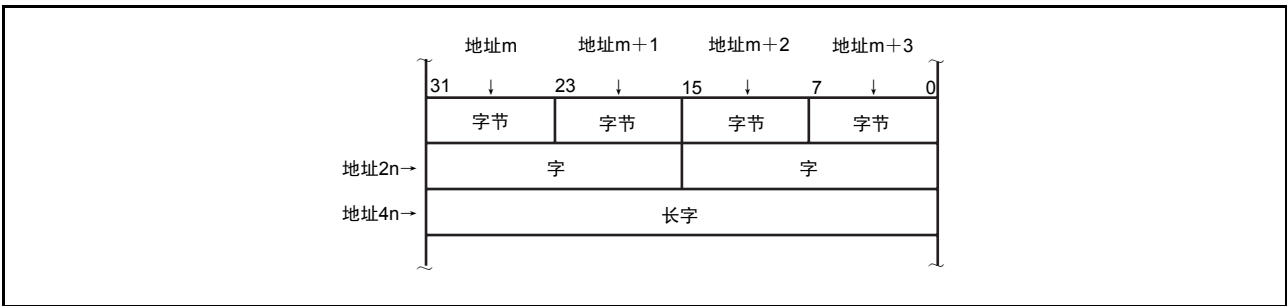


图 2.3 存储器的数据格式

2.3.3 立即数的数据格式

字节（8 位）的立即数分配在指令码中。

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

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

2.4 指令特点

2.4.1 RISC 方式

指令为 RIS 方式，特点如下：

(1) 16 位固定长指令

指令长度全部是 16 位固定长，因此能提高程序码的效率。

(2) 1 条指令 / 1 个状态

采用流水线方式，基本指令能以 1 条指令执行 1 个状态。

(3) 数据长度

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

表 2.2 字数据的符号扩展

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

【注】 用 @ (disp, PC) 参照立即数。

(4) 装入 - 存储结构

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

(5) 延迟转移

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

表 2.3 延迟转移指令

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

(6) 乘法 / 乘加运算

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

(7) T 位

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

表 2.4 T 位

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

(8) 立即数

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

表 2.5 立即数的参照

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

【注】用 @ (disp, PC) 参照立即数。

(9) 绝对地址

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

表 2.6 绝对地址的参照

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

【注】用 @ (disp, PC) 参照立即数。

(10) 16 位 / 32 位的位移量

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

表 2.7 位移量的参照

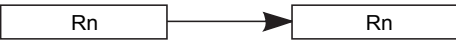
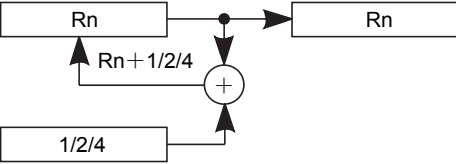
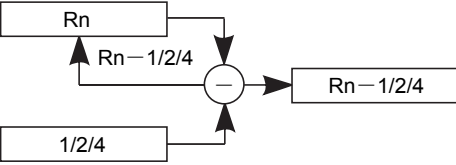
区 分	本 LSI 的 CPU	其他 CPU 的例子
16 位的位移量	MOV.W @(disp, PC), R0	MOV.W @(H'1234, R1), R2
	MOV.W @(R0, R1), R2	
		
	.DATA.W H'1234	

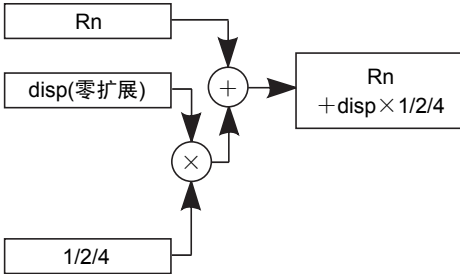
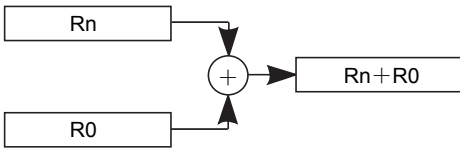
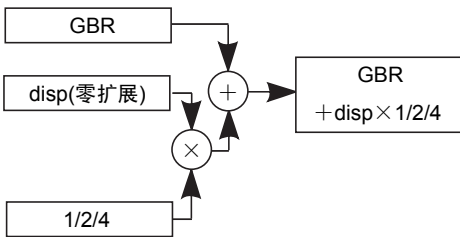
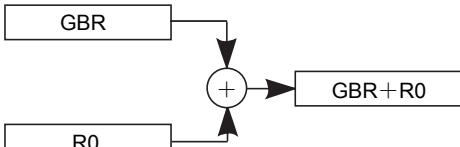
【注】 用@ (disp, PC) 参照立即数。

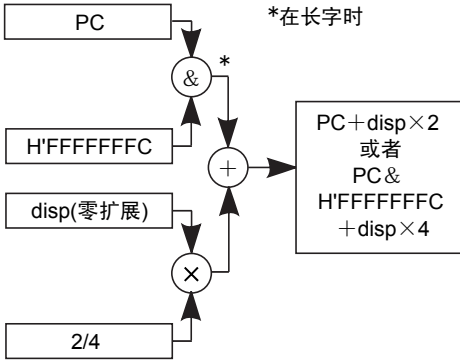
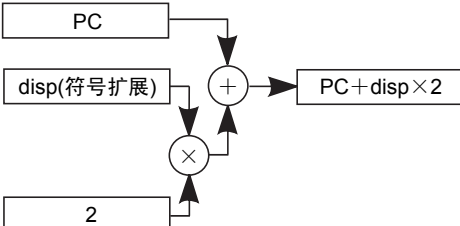
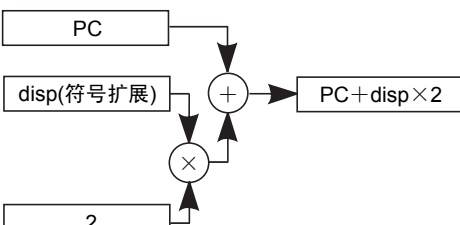
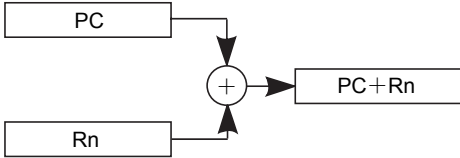
2.4.2 寻址方式

寻址方式和有效地址的计算方法如表 2.8 所示。

表 2.8 寻址方式和有效地址

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

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

寻址方式	指令格式	有效地址的计算方法	计算式
带位移量的 PC 相对	@(disp: 8, PC)	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: $PC \& H'FFFFFFFC + disp \times 4$
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。 	$PC + disp \times 2$
	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、XOR 指令的 8 位立即数 imm 进行零扩展。	—
	#imm:8	MOV、ADD、CMP/EQ 指令的 8 位立即数 imm 进行符号扩展。	—
	#imm:8	TRAPA 指令的 8 位立即数 imm 进行零扩展后乘 4。	—

2.4.3 指令格式

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

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

表 2.9 指令格式

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

指令格式	源操作数	目标操作数	指令例子
nmd 格式	15 xxxx nnnn mmmm dddd 0	mmmm: 寄存器直接	nnnndddd: 带位移量的寄存器间接 MOV.L Rm, @(disp, Rn)
		mmmmdddd: 带位移量的寄存器间接	nnnn: 寄存器直接 MOV.L @(disp, Rm), Rn
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	R0 (寄存器直接) 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	
		MOV T	传送 T 位	
		SWAP	交换高位和低位	
		XTRCT	抽出连接寄存器的中间部分	
算术运算指令	21	ADD	2 进制加法	33
		ADDC	带进位的 2 进制加法	
		ADDV	带上溢的 2 进制加法	
		CMP/cond	比较	
		DIV1	除法	
		DIV0S	带符号的除法初始化	
		DIV0U	无符号的除法初始化	
		DMULS	带符号的双精度乘法	
		DMULU	无符号的双精度乘法	
		DT	递减和测试	
		EXTS	符号扩展	
		EXTU	零扩展	
		MAC	乘加运算、双精度乘加运算	
		MUL	双精度乘法	
		MULS	带符号乘法	
		MULU	无符号乘法	
		NEG	符号取反	
		NEGC	带借位的符号取反	
		SUB	2 进制减法	
		SUBC	带借位的 2 进制减法	
		SUBV	带下溢的 2 进制减法	
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 (当 T=0 时转移)	11
		BT	条件转移、带延迟的条件转移 (当 T=1 时转移)	
		BRA	无条件转移	
		BRAF	无条件转移	
		BSR	转移到子程序过程	
		BSRF	转移到子程序过程	
		JMP	无条件转移	
		JSR	转移到子程序过程	
		RTS	从子程序过程返回	
系统控制指令	11	CLRT	清除 T 位	31
		CLRMAC	清除 MAC 寄存器	
		LDC	装入到控制寄存器	
		LDS	装入到系统寄存器	
		NOP	无操作	
		RTE	从异常处理返回	
		SETT	T 位的置位	
		SLEEP	转移到低功耗状态	
		STC	保存控制寄存器的内容	
		STS	保存系统寄存器的内容	
		TRAPA	陷阱异常处理	
	计 62			142

通过以下格式按照分类顺序说明指令的指令码、操作和执行状态：

指令	指令码	操作概略	执行状态	T 位
用助记符表示。	按照 MSB $\leftarrow$ $\rightarrow$ LSB 的顺序表示。	表示操作的概略。	是无等待时的值。*1	表示执行指令后的 T 位的值。
符号说明 OP.Sz SRC, DEST	符号说明 mmmm: 源寄存器 nnnn: 目标寄存器 0000: R0 0001: R1 1111: R15	符号说明 $\rightarrow$ 、 $\leftarrow$: 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 ~: 每位的逻辑否定 <<n: 左移 n 位 >>n: 右移 n 位		符号说明 —: 不变
Rm: 源寄存器 Rn: 目标寄存器 imm: 立即数 disp: 位移量 *2	iiii: 立即数 dddd: 位移量			

【注】 *1 有关指令的执行状态

表中所示的执行状态是最小值。实际上根据以下条件，指令的执行状态数将会增加：

(1) 当取指令和数据存取发生竞争时

(2) 当装入指令（存储器 $\rightarrow$ 寄存器）的目标寄存器和紧接着的指令所使用的寄存器相同时

*2 根据指令的操作数长度等进行倍增（X1、X2、X4）。

详细内容请参照《SH-1/SH-2/SH-DSP 软件手册》。

2.5.2 数据传送指令

表 2.11 数据传送指令

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

指令	指令码	操作	执行状态	T 位
SWAP.W Rm, Rn	0110nnnnmmmm1001	Rm→ 交换高低字 →Rn	1	—
XTRCT Rm, Rn	0010nnnnmmmm1101	Rm: Rn 中间 32 位 →Rn	1	—

2.5.3 算术运算指令

表 2.12 算术运算指令

指 令	指令码	操作	执行状态	T 位
ADD Rm, Rn	0011nnnnmmmm1100	Rn+Rm→Rn	1	—
ADD #imm, Rn	0111nnnniiiiiii	Rn+imm→Rn	1	—
ADDC Rm, Rn	0011nnnnmmmm1110	Rn+Rm+T→Rn, 进位 →T	1	进位
ADDV Rm, Rn	0011nnnnmmmm1111	Rn+Rm→Rn, 上溢 →T	1	上溢
CMP/EQ #imm, R0	10001000iiiiiii	当 R0=imm 时, 1→T	1	比较结果
CMP/EQ Rm, Rn	0011nnnnmmmm0000	当 Rn=Rm 时, 1→T	1	比较结果
CMP/HS Rm, Rn	0011nnnnmmmm0010	当无符号 Rn ≥ Rm 时, 1→T	1	比较结果
CMP/GE Rm, Rn	0011nnnnmmmm0011	当带符号 Rn ≥ Rm 时, 1→T	1	比较结果
CMP/HI Rm, Rn	0011nnnnmmmm0110	当无符号 Rn > Rm 时, 1→T	1	比较结果
CMP/GT Rm, Rn	0011nnnnmmmm0111	当带符号 Rn > Rm 时, 1→T	1	比较结果
CMP/PL Rn	0100nnnn00010101	当 Rn > 0 时, 1→T	1	比较结果
CMP/PZ Rn	0100nnnn00010001	当 Rn ≥ 0 时, 1→T	1	比较结果
CMP/STR Rm, Rn	0010nnnnmmmm1100	当任意字节相等时, 1→T	1	比较结果
DIV1 Rm, Rn	0011nnnnmmmm0100	单步除法 (Rn ÷ Rm)	1	计算结果
DIV0S Rm, Rn	0010nnnnmmmm0111	Rn 的 MSB→Q, Rm 的 MSB→M, M^Q→T	1	计算结果
DIV0U	0000000000011001	0→M/Q/T	1	0
DMULS.L Rm, Rn	0011nnnnmmmm1101	带符号 Rn×Rm→MACH 和 MACL 32×32→64 位	2 ~ 5*	—
DMULU.L Rm, Rn	0011nnnnmmmm0101	无符号 Rn×Rm→MACH 和 MACL 32×32→64 位	2 ~ 5*	—
DT Rn	0100nnnn00010000	Rn - 1→Rn 当 Rn 为 0 时, 1→T 当 Rn 不为 0 时, 0→T	1	比较结果
EXTS.B Rm, Rn	0110nnnnmmmm1110	将 Rm 的字节进行符号扩展 →Rn	1	—
EXTS.W Rm, Rn	0110nnnnmmmm1111	将 Rm 的字进行符号扩展 →Rn	1	—
EXTU.B Rm, Rn	0110nnnnmmmm1100	将 Rm 的字节进行零扩展 →Rn	1	—
EXTU.W Rm, Rn	0110nnnnmmmm1101	将 Rm 的字进行零扩展 →Rn	1	—
MAC.L @Rm+, @Rn+	0000nnnnmmmm1111	带符号 (Rn)×(Rm)+MAC→MAC 32×32 + 64→64 位	2 ~ 5*	—
MAC.W @Rm+, @Rn+	0100nnnnmmmm1111	带符号 (Rn)×(Rm)+MAC→MAC 16×16 + 64→64 位	2 ~ 4*	—
MUL.L Rm, Rn	0000nnnnmmmm0111	Rn×Rm→MACL 32×32→32 位	2 ~ 5*	—

指 令	指令码	操作	执行状态	T 位
MULS.W Rm, Rn	0010nnnnmmmm1111	带符号 $Rn \times Rm \rightarrow MACL$ 16×16→32 位	1 ~ 3*	—
MULU.W Rm, Rn	0010nnnnmmmm1110	无符号 $Rn \times Rm \rightarrow MACL$ 16×16→32 位	1 ~ 3*	—
NEG Rm, Rn	0110nnnnmmmm1011	$0 - Rm \rightarrow Rn$	1	—
NEGC Rm, Rn	0110nnnnmmmm1010	$0 - Rm - T \rightarrow Rn$, 借位 $\rightarrow T$	1	借位
SUB Rm, Rn	0011nnnnmmmm1000	$Rn - Rm \rightarrow Rn$	1	—
SUBC Rm, Rn	0011nnnnmmmm1010	$Rn - Rm - T \rightarrow Rn$, 借位 $\rightarrow T$	1	借位
SUBV Rm, Rn	0011nnnnmmmm1011	$Rn - Rm \rightarrow Rn$, 下溢 $\rightarrow T$	1	下溢

【注】 * 表示通常的执行状态。

2.5.4 逻辑运算指令

表 2.13 逻辑运算指令

指令	指令码	操作	执行状态	T 位
AND Rm, Rn	0010nnnnmmmm1001	$Rn \& Rm \rightarrow Rn$	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	$Rn Rm \rightarrow Rn$	1	—
OR #imm, R0	11001011iiiiiii	$R0 imm \rightarrow R0$	1	—
OR.B #imm, @(R0, GBR)	11001111iiiiiii	$(R0 + GBR) imm \rightarrow (R0 + GBR)$	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 时, $1 \rightarrow T$	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】 * 在不转移时, 为 1 个状态。

2.5.7 系统控制指令

表 2.16 系统控制指令

指令	指令码	操作	执行状态	T 位
CLRT	0000000000001000	0→T	1	0
CLRMAC	0000000000101000	0→MACH 和 MACL	1	—
LDC Rm, SR	0100mmmm00001110	Rm→SR	6	LSB
LDC Rm, GBR	0100mmmm00011110	Rm→GBR	4	—
LDC Rm, VBR	0100mmmm00101110	Rm→VBR	4	—
LDC.L @Rm+, SR	0100mmmm00000111	(Rm)→SR, Rm+4→Rm	8	LSB
LDC.L @Rm+, GBR	0100mmmm00010111	(Rm)→GBR, Rm+4→Rm	4	—
LDC.L @Rm+, VBR	0100mmmm00100111	(Rm)→VBR, Rm+4→Rm	4	—
LDS Rm, MACH	0100mmmm00001010	Rm→MACH	1	—
LDS Rm, MACL	0100mmmm00011010	Rm→MACL	1	—
LDS Rm, PR	0100mmmm00101010	Rm→PR	1	—
LDS.L @Rm+, MACH	0100mmmm00000110	(Rm)→MACH, Rm+4→Rm	1	—
LDS.L @Rm+, MACL	0100mmmm00010110	(Rm)→MACL, Rm+4→Rm	1	—
LDS.L @Rm+, PR	0100mmmm00100110	(Rm)→PR, Rm+4→Rm	1	—
NOP	0000000000001001	无操作	1	—
RTE	0000000000101011	延迟转移 堆栈区 →PC/SR	5	—
SETT	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】* 是转移到睡眠状态前的状态数。

有关指令的执行状态

表中所示的执行状态为最小值。实际上根据以下条件，指令的执行状态将会增加：

(1) 当发生取指令和数据存取的竞争时

(2) 当装入指令（存储器→寄存器）的目标寄存器和紧接着的指令所使用的寄存器相同时

2.6 处理状态

CPU 的处理状态有复位状态、异常处理状态、程序执行状态和低功耗状态共 4 种。状态之间的转移如图 2.4 所示。

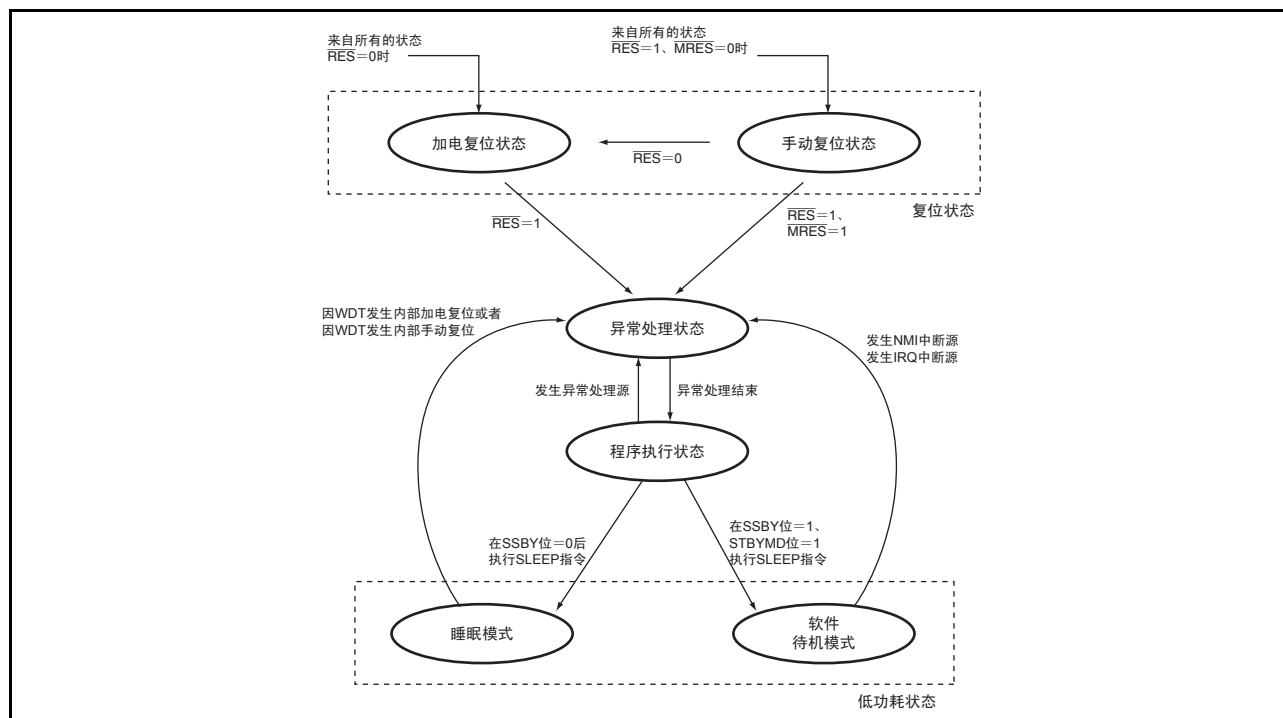


图 2.4 处理状态的转移图

(1) 复位状态

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

(2) 异常处理状态

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

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

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

以后的处理状态为程序执行状态。

(3) 程序执行状态

这是 CPU 依次执行程序的状态。

(4) 低功耗状态

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

第 3 章 MCU 运行模式

3.1 运行模式的选择

本 LSI 有 1 种 MCU 运行模式和 3 种用于内部闪存编程的模式。

通过组合 FWE 管脚和 MD1 管脚，设定运行模式。

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

另外，在接通系统电源时，必须进行加电复位处理。

MCU 运行模式有单芯片模式。

用于内部闪存编程的模式有引导模式（单板编程模式）和用户编程模式。

表 3.1 运行模式的选择

MCU 运行模式	管脚设定		模式	内部 ROM
	FWE	MD1		
模式 3	0	1	单芯片模式	有效
模式 4*	1	0	引导模式	有效
模式 6*	1	1	用户编程模式	有效

【注】 * 闪存的编程模式。

3.2 输入 / 输出管脚

运行模式关联的管脚结构如表 3.2 所示。

表 3.2 管脚结构

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

3.3 各运行模式的说明

3.3.1 模式 3（单芯片模式）

在单芯片模式中能使用全部端口。

3.4 地址映像

地址映像如图 3.1 和图 3.2 所示。

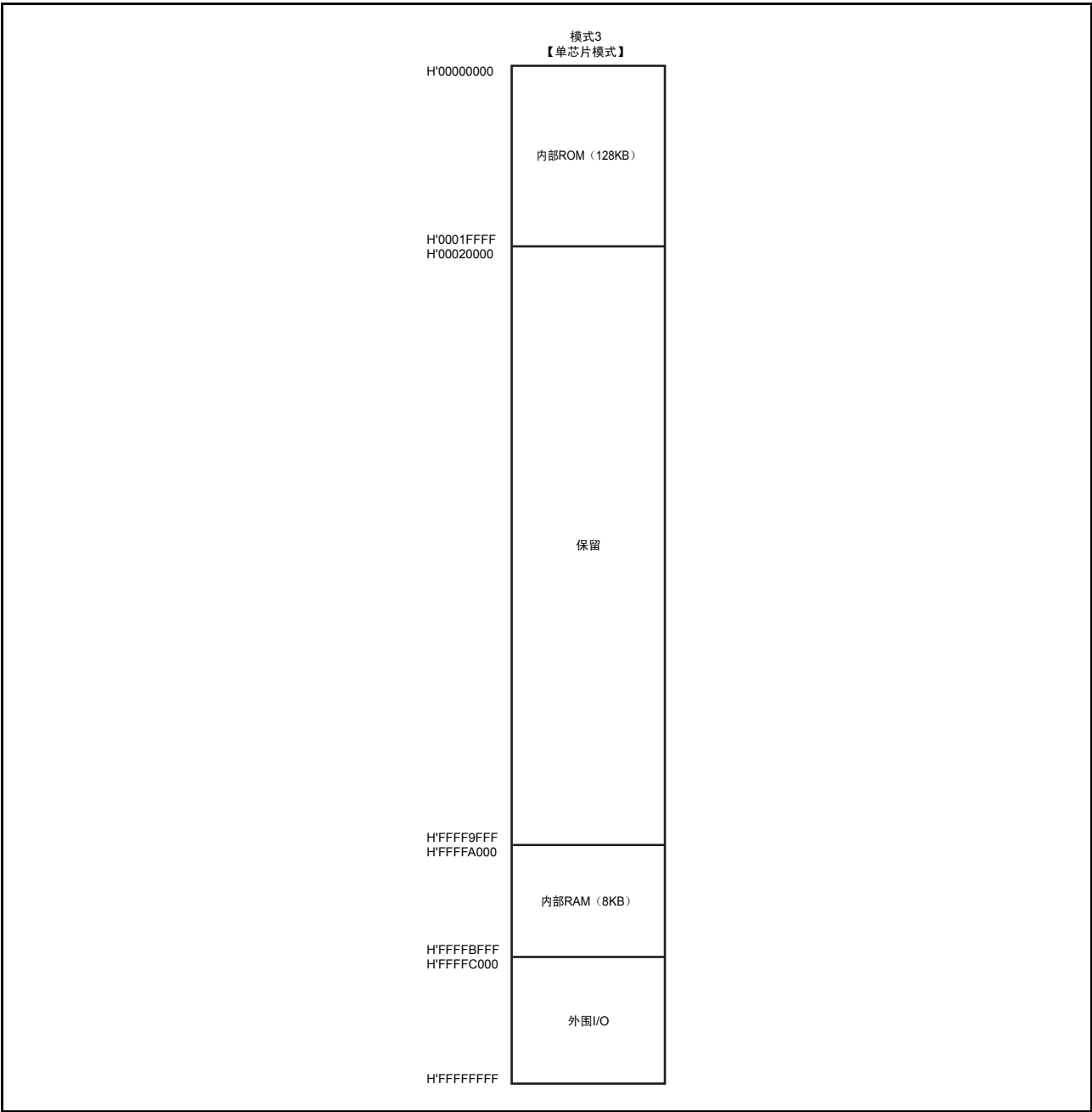


图 3.1 SH7125 和 SH7124（闪存 128KB 版）的地址映像

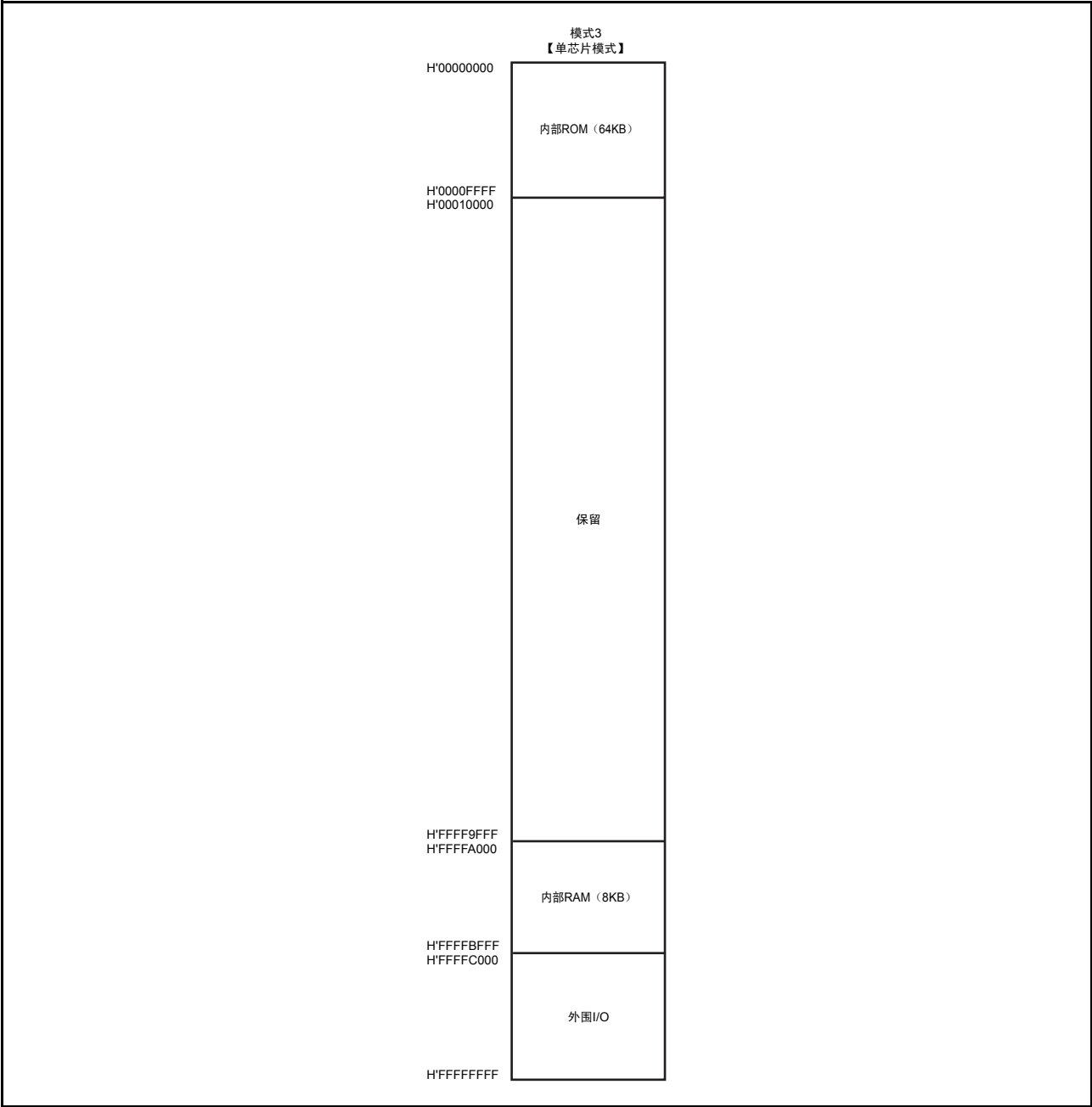


图 3.2 SH7125 和 SH7124（闪存 64KB 版）的地址映像

3.5 本 LSI 的初始状态

为了降低功耗，本 LSI 在初始状态下将一部分内部模块设定为模块待机状态。因此，要使这些模块运行时，必须解除模块待机状态。详细内容请参照“第 19 章 低功耗模式”。

3.6 更改运行模式时的注意事项

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

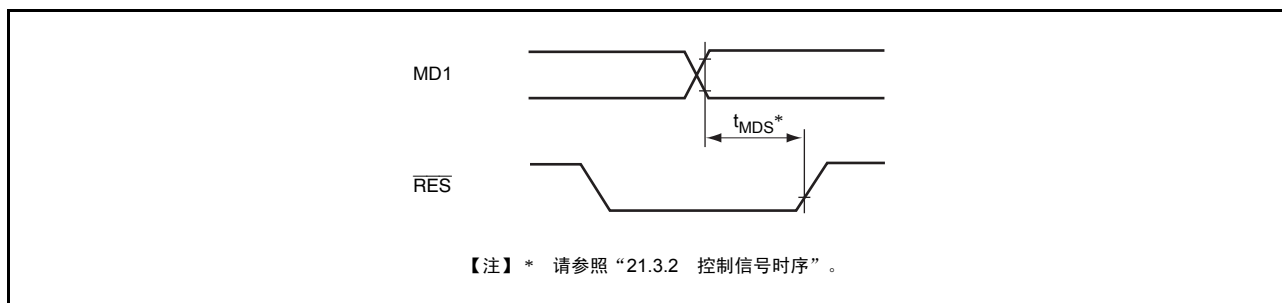


图 3.3 更改运行模式时的复位输入时序

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 ($P\phi$) 和用于 MTU2 模块的时钟 ($MP\phi$) 并进行低功耗模式的控制。

4.1 特点

- 能独立生成 4 种时钟
能独立生成用于 CPU 和高速缓存的内部时钟 ($I\phi$)、用于外围模块的外围时钟 ($P\phi$)、用于内部总线的总线时钟 ($B\phi=CK$) 以及用于内部 MTU2 模块的 MTU2 时钟 ($MP\phi$)。
- 频率的变更功能
能通过 CPG 的内部分频电路独立更改内部时钟 ($I\phi$)、总线时钟 ($B\phi$)、外围时钟 ($P\phi$) 和 MTU2 时钟 ($MP\phi$) 的频率。根据频率控制寄存器 (FRQCR) 的设定, 通过软件更改频率。
- 低功耗模式的控制
能停止睡眠模式和待机模式中的时钟以及模块待机功能中的特定模块。
- 振荡停止的检测功能
如果因某种理由而停止从时钟输入管脚提供时钟, 定时器管脚就能自动进入高阻抗状态。

CPG 的框图如图 4.1 所示。

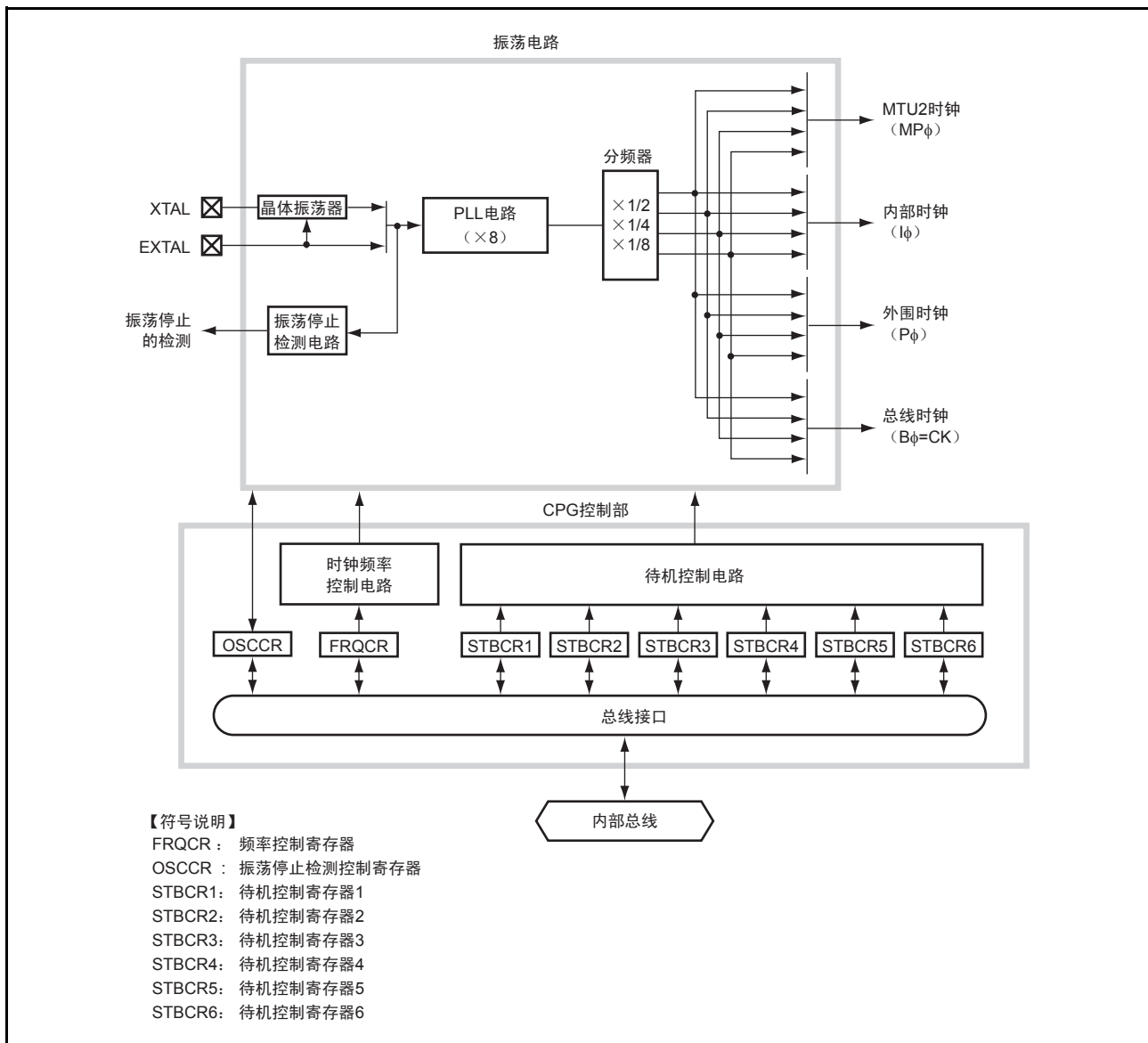


图 4.1 CPG 的框图

CPG 中的各模块的功能如下：

(1) PLL 电路

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

(2) 晶体振荡器

这是将晶体谐振器连接到 XTAL、EXTAL 管脚时的振荡电路。

(3) 分频器

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

运行频率可选择 PLL 电路的输出频率的 1/2 倍、1/4 倍 (初始值) 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

这是检测晶体振荡器异常的电路。

(5) 时钟频率的控制电路

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

(6) 待机控制电路

待机控制电路控制睡眠 / 待机模式中的内部振荡电路和其他模块的状态。

(7) 频率控制寄存器 (FRQCR)

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

(8) 振荡停止检测控制寄存器 (OSCCR)

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

(9) 待机控制寄存器 1 ~ 6 (STBCR1 ~ STBCR6)

待机控制寄存器 (STBCR) 中分配了低功耗模式的各控制位。有关待机控制寄存器请参照“第 19 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 A/D CMT WDT
总线时钟（B ϕ ）	—	MTU2 时钟（MP ϕ ）	MTU2

4.2 输入 / 输出管脚

CPG 的管脚结构和功能如表 4.2 所示。

表 4.2 时钟振荡器的管脚结构和功能

名称	略称	输入 / 输出	功 能
晶体谐振器输入 / 输出管脚 (时钟输入管脚)	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管脚。

4.3 时钟运行模式

本 LSI 的时钟运行模式如表 4.3 所示。

表 4.3 时钟运行模式

供给源	PLL 电路	分频器的输入
EXTAL 输入 晶体谐振器	ON (X8)	X8

因为从 EXTAL 管脚输入外部时钟，通过 PLL 电路将频率递增 8 倍后提供给本 LSI 内部，所以降低了外部生成的时钟频率。能使用的输入时钟频率为 10MHz 到 12.5MHz，内部时钟（I ϕ ）的频率范围为 10MHz 到 50MHz。

最大运行频率：I ϕ = 50MHz、B ϕ = 40MHz、P ϕ = 40MHz、MP ϕ = 40MHz

能通过 FRQCR 设定的分频比如表 4.4 所示。

表 4.4 由 FRQCR 设定的分频比

PLL 倍增率	FRQCR 的分频比设定				时钟比				时钟频率（MHz）*				
	I ϕ	B ϕ	P ϕ	MP ϕ	I ϕ	B ϕ	P ϕ	MP ϕ	输入时钟	I ϕ	B ϕ	P ϕ	MP ϕ
X8	1/8	1/8	1/8	1/8	1	1	1	1	10	10	10	10	10
	1/4	1/8	1/8	1/8	2	1	1	1	10	20	10	10	10
	1/4	1/4	1/4	1/4	2	2	2	2	10	20	20	20	20
	1/2	1/4	1/4	1/4	4	2	2	2	10	40	20	20	20
	1/2	1/2	1/2	1/2	4	4	4	4	10	40	40	40	40
	1/8	1/8	1/8	1/8	1	1	1	1	12.5	12.5	12.5	12.5	12.5
	1/4	1/8	1/8	1/8	2	1	1	1	12.5	25	12.5	12.5	12.5
	1/4	1/4	1/4	1/4	2	2	2	2	12.5	25	25	25	25
	1/2	1/4	1/4	1/4	4	2	2	2	12.5	50	25	25	25

【注】* 时钟频率是假定了输入时钟频率时的值。必须将 I ϕ 设定为 10 ~ 50MHz、P ϕ 设定为 10 ~ 40MHz，B ϕ 和 P ϕ 相等。

- PLL 电路的倍增率只为 X8。分频器的分频比能选择 X1/2、X1/4 或者 X1/8。
通过频率控制寄存器，给要设定的各时钟设定分频比。
- PLL 电路的输出频率为晶体谐振器或者 EXTAL 管脚输入的时钟频率乘以 PLL 电路的倍增率（8 倍）。此频率必须小于等于 50MHz。
- 分频器的输入总是 PLL 电路的输出。
- 内部时钟（I ϕ ）的频率为晶体谐振器或者 EXTAL 管脚输入的时钟频率乘以 PLL 电路的倍增率（8 倍）和分频器的分频比。
必须将内部时钟（I ϕ ）的频率设定为小于等于最大运行频率（50MHz）。
- 外围时钟（P ϕ ）的频率为晶体谐振器或者 EXTAL 管脚输入的时钟频率乘以 PLL 电路的倍增率（8 倍）和分频器的分频比。
必须将外围时钟（P ϕ ）的频率设定为小于等于 40MHz。
- 在使用 MTU2 时，MTU2 时钟（MP ϕ ）的频率必须大于等于外围时钟（P ϕ ）的频率。
- CK 的频率总是和总线时钟（B ϕ ）的频率相等。
- 总线时钟（B ϕ ）的频率必须和外围时钟（P ϕ ）的频率相等。

4.4 寄存器说明

CPG 有以下寄存器。有关寄存器的地址和各处理模式的寄存器状态，请参照“第20章 寄存器一览”。

表 4.5 寄存器结构

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

4.4.1 频率控制寄存器（FRQCR）

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

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

另外，要更改 FRQCR 时，必须将待机控制寄存器 2、3、4、5、6 的模块停止位设定为 1，然后在已停止给各模块提供时钟的状态下进行。

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]			-	-	-	MPFC[2:0]		
初始值:	0	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读写的值总是 0。
14 ~ 12	IFC[2:0]	011	R/W	内部时钟（I ϕ ）的分频比 对于 PLL 电路的输出频率，指定内部时钟（I ϕ ）的分频比。 000: 禁止设定 001: X1/2 倍 010: 禁止设定 011: X1/4 倍（初始值） 100: X1/8 倍 上述以外: 禁止设定
11 ~ 9	BFC[2:0]	011	R/W	总线时钟（B ϕ ）的分频比 对于 PLL 电路的输出频率，指定总线时钟（B ϕ ）的分频比。但是，必须和外围时钟（P ϕ ）的设定相同。 000: 禁止设定 001: X1/2 倍 010: 禁止设定 011: X1/4 倍（初始值） 100: X1/8 倍 上述以外: 禁止设定

位	位名	初始值	R/W	说明
8 ~ 6	PFC[2:0]	011	R/W	外围时钟（P ϕ ）的分频比 对于 PLL 电路的输出频率，指定外围时钟（P ϕ ）的分频比。 000: 禁止设定 001: X1/2 倍 010: 禁止设定 011: X1/4 倍（初始值） 100: X1/8 倍 上述以外: 禁止设定
5 ~ 3	—	011	R/W	保留位 读写的值总是为 B'011。
2 ~ 0	MPFC[2:0]	011	R/W	MTU2 时钟（MP ϕ ）的分频比 对于 PLL 电路的输出频率，指定 MTU2 时钟（MP ϕ ）的分频比。 000: 禁止设定 001: X1/2 倍 010: 禁止设定 011: X1/4 倍（初始值） 100: X1/8 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，选择振荡停止检测标志和外部管脚的标志输出。OSCCR 只能进行字节存取。

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

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

4.5 频率的变更方法

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

1. 在初始状态下， $IFC2 \sim IFC0 = B'011$ (X1/4 倍)、 $BFC2 \sim BFC0 = B'011$ (X1/4 倍)、 $PFC2 \sim PFC0 = B'011$ (X1/4 倍)、 $MPFC2 \sim MPFC0 = B'011$ (X1/4 倍)。
2. 停止除 CPU、内部 ROM 和内部 RAM 以外的模块。
3. 将 $IFC2 \sim IFC0$ 、 $BFC2 \sim BFC0$ 、 $PFC2 \sim PFC0$ 、 $MPFC2 \sim MPFC0$ 位设定为目标值。由于 PLL 电路的频率倍增率固定为 X8 倍，所以只能通过设定分频比决定频率。此时的频率必须设定为内部时钟 ($I\phi$) $\geq$ 总线时钟 ($B\phi$) = 外围时钟 ($P\phi$)。另外，在使用 MTU2 时钟的情况下，必须设定内部时钟 ($I\phi$) $\geq$ MTU2 时钟 ($MP\phi$) $\geq$ 外围时钟 ($P\phi$)。
4. 在设定 FRQCR 后，各时钟频率立即变为已设定的值。

4.6 振荡器

提供时钟的方法有连接晶体谐振器和输入外部时钟的方法。

4.6.1 连接晶体谐振器的方法

晶体谐振器的连接例子如图 4.2 所示，必须使用表 4.6 所示的阻尼电阻 R_d 和频率为 10 ~ 12.5MHz 的晶体谐振器。

有关晶体谐振器和 LSI 的匹配，请与晶体谐振器厂家洽谈。

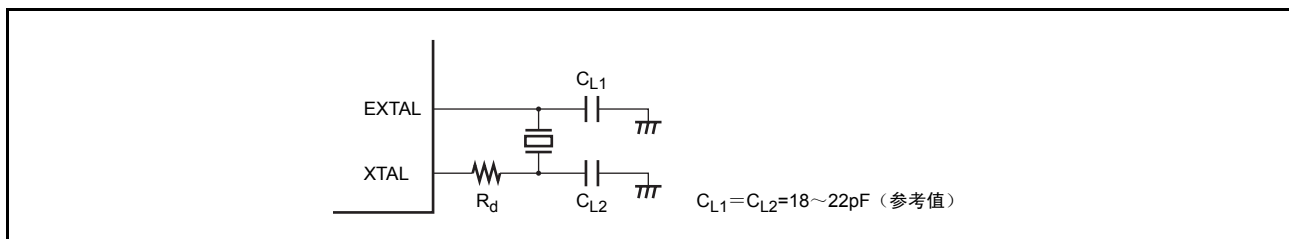


图 4.2 晶体谐振器的连接例子

表 4.6 阻尼电阻值 (参考值)

频率 (MHz)	10	12.5
R_d (Ω) (参考值)	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。

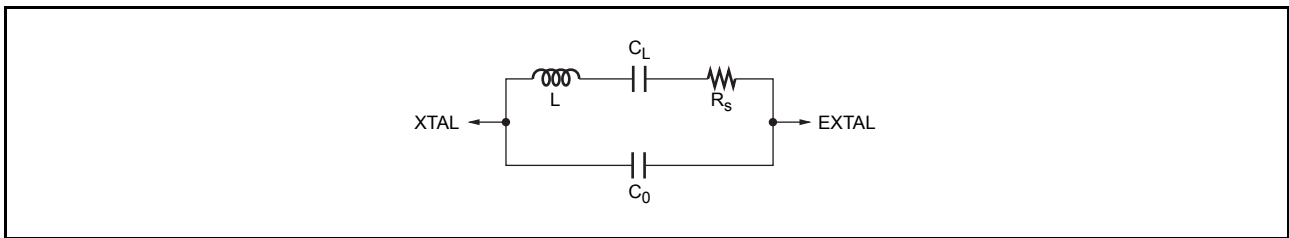


图 4.3 晶体谐振器的等效电路

表 4.7 晶体谐振器的特性

频率 (MHz)	10	12.5
R_s Max. (Ω) (参考值)	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。在运行时，外部输入时钟的频率必须在 10 ~ 12.5MHz 之间。XTAL 管脚的寄生电容不能大于 10pF。

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



图 4.4 外部时钟的连接例子

4.7 振荡停止检测功能

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

如果检测到 EXTAL 输入在一定的期间内不发生变化，就将 OSCCR 寄存器的 OSCSTOP 位设定为 1，并且将此状态保持到 RES 管脚的加电复位或者解除软件待机模式为止。此时，一旦将 OSCERS 位设定为 1，就从 WDTOVF 管脚输出振荡停止检测标志信号。另外，大电流端口 (MTU2 的 TI0C3B、TI0C3D、TI0C4A ~ TI0C4D 为多路复用管脚) 和 PFC 的设定无关，总为高阻抗。详细内容请参照“附录 A. 管脚状态”。

即使在软件待机状态下，上述管脚也总为高阻抗。详细内容请参照“附录 A. 管脚状态”。在解除软件待机状态后，变为通常运行状态。另外，在非软件待机状态下振荡异常停止时，LSI 的其他运行不稳定。此时，即使重新开始振荡，包括上述管脚的 LSI 运行也不稳定。

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

4.8 使用注意事项

4.8.1 有关谐振器的注意事项

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

4.8.2 电路板设计的注意事项

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

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

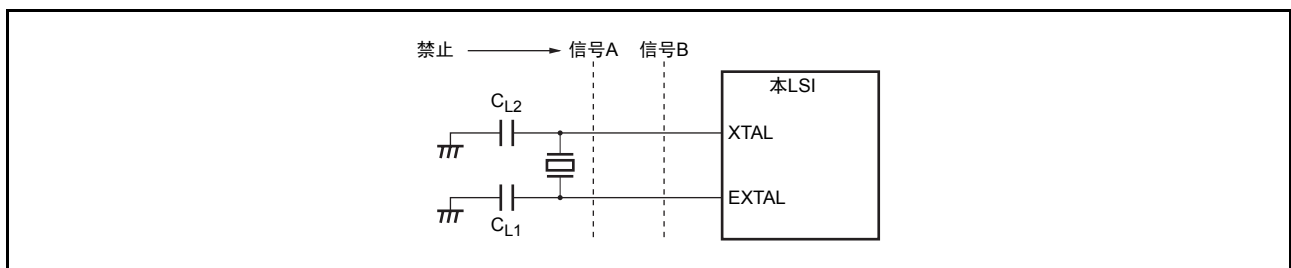


图 4.5 有关振荡电路电路板设计的注意事项

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

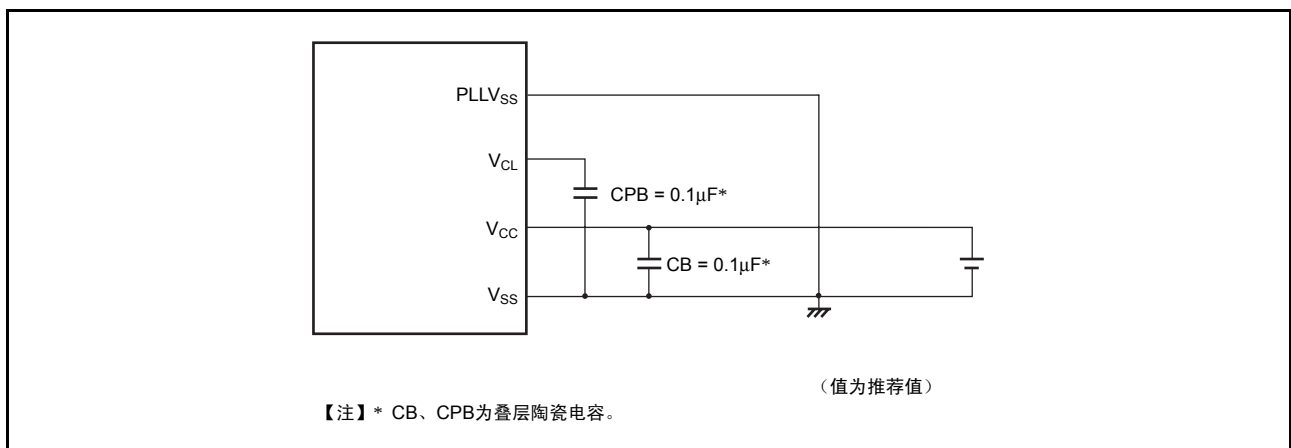


图 4.6 PLL 电路的外接推荐电路

第 5 章 异常处理

5.1 概要

5.1.1 异常处理的种类和优先级

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

表 5.1 异常源的种类和优先级

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

在如表 5.2 所示的时序中检测到各异常源后开始处理。

表 5.2 异常源检测和异常处理的开始时序

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

当启动异常处理时，CPU 的运行如下：

(1) 由复位引起的异常处理

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

(2) 由地址错误、中断和指令引起的异常处理

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

5.1.3 异常处理向量表

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

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

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

表 5.3 异常处理向量表

异常源		向量号	向量表地址偏移量
加电复位	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
系统保留		10	H'00000028 ~ H'0000002B
中断	NMI	11	H'0000002C ~ H'0000002F
	用户断点	12	H'00000030 ~ H'00000033
(系统保留)		13	H'00000034 ~ H'00000037
		⋮	⋮
		31	H'0000007C ~ H'0000007F
陷阱指令 (用户向量)		32	H'00000080 ~ H'00000083
		⋮	⋮
		63	H'000000FC ~ H'000000FF
中断	IRQ0 (SH7125)	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
		⋮	⋮
		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，有加电复位和手动复位 2 种。如表 5.5 所示，在加电复位或者手动复位时，CPU 状态都被初始化。另外，内部外围模块的寄存器在加电复位时被初始化，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
加电复位	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	不初始化
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 加电复位

(1) 由 RES 管脚引起的加电复位

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

在加电复位的状态下，如果将 RES 管脚保持一定期间的低电平后变为高电平，就开始加电复位的异常处理。此时 CPU 的运行如下：

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

另外，必须在接通系统电源时进行加电复位处理。

(2) 由 WDT 引起的加电复位

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

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

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

1. 从异常处理向量表中取出程序计数器（PC）的初始值（执行起始地址）。
2. 从异常处理向量表中取出堆栈指针（SP）的初始值。
3. 将向量基准寄存器（VBR）清除为 H'00000000，状态寄存器（SR）的中断屏蔽位（I3 ~ I0）置为 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 $\overline{\text{RES}}$ 管脚为高电平时将 $\overline{\text{MRES}}$ 管脚置为低电平，本 LSI 就进入手动复位状态。为了确保本 LSI 的复位， $\overline{\text{MRES}}$ 管脚的 low 电平在软件待机时（在时钟停止时），至少必须保持 WDT 设定的振荡稳定时间；在时钟运行时，至少保持 20tcyc。在手动复位的状态下，CPU 的内部状态被初始化，而内部外围模块的各寄存器不被初始化。如果在总线周期中进入手动复位的状态，就在等待总线周期结束后开始手动复位的异常处理。因此，总线周期不会因手动复位而中途停止。但是，如果将 $\overline{\text{MRES}}$ 置为低电平，就必须将低电平保持到总线周期结束并进入手动复位状态为止（低电平必须保持大于最长的总线周期）。有关手动复位状态下的各管脚状态，请参照“附录 A. 管脚状态”。在手动复位的状态下，如果将 $\overline{\text{MRES}}$ 管脚保持一定期间的低电平后变为高电平，就开始手动复位的异常处理。此时，CPU 按照和加电复位异常处理相同的步骤运行。

5.3 地址错误

5.3.1 地址错误的发生源

如表 5.6 所示，在取指令或者读写数据时发生地址错误。

表 5.6 总线周期和地址错误

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

5.3.2 地址错误的异常处理

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

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

5.4 中断

5.4.1 中断源

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

表 5.7 中断源

种类	请求源	源数
NMI	NMI 管脚（来自外部的输入）	1
用户断点	用户断点控制器（UBC）	1
IRQ	IRQ0 ~ IRQ3 管脚（来自外部的输入）	4（SH7125） 3（SH7124）
内部外围模块	多功能定时器脉冲单元 2（MTU2）	28
	看门狗定时器（WDT）	1
	A/D 转换器（A/D_0、A/D_1）	2
	比较匹配定时器（CMT_0、CMT_1）	2
	串行通信接口（SCI_0、SCI_1、SCI_2）	12
	端口输出的允许（POE）	2

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

5.4.2 中断优先级

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

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

表 5.8 中断优先级

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

5.4.3 中断异常处理

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

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

5.5 由指令引起的异常

5.5.1 由指令引起的异常种类

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

表 5.9 由指令引起的异常种类

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

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

5.5.2 陷阱指令

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

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

5.5.3 槽非法指令

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

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

5.5.4 一般非法指令

如果对配置在除紧接着延迟转移指令（延迟槽）以外的未定义代码进行译码，就开始一般非法指令的异常处理。此时，CPU 按照和槽非法指令的异常处理相同的步骤运行。但是，和槽非法指令的异常处理不同，保存的 PC 值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，如果在紧接着延迟槽或者中断禁止指令之后发生除复位以外的全部异常源，就有可能不立即被接受而被保留。这种情况下，在接受的指令被译码后才能接受以上的异常源。

表 5.10 发生在紧接着延迟槽和中断禁止指令之后的异常源

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

【符号说明】○：接受。

X：不接受。

—：不可能的情况。

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

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

*3 在延迟转移后（在延迟槽指令和延迟转移目标指令之间）被接受。

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

5.7 异常处理后的堆栈状态

异常处理结束后的堆栈状态如表 5.11 所示。

表 5.11 异常处理结束后的堆栈状态

种类	堆栈状态	种类	堆栈状态
地址错误 (引起异常发生的指令在延迟槽时)	SP → 延迟转移指令的地址 (32位) SR (32位)	地址错误 (引起异常发生的指令不在延迟槽时)	SP → 引起异常的指令地址 (32位) SR (32位)
中断	SP → 已执行指令的下一个指令地址 (32位) SR (32位)	陷阱指令	SP → TRAPA指令的下一个指令地址 (32位) SR (32位)
槽非法指令	SP → 延迟转移指令的地址 (32位) SR (32位)	一般非法指令	SP → 一般非法指令的地址 (32位) SR (32位)

5.8 使用注意事项

5.8.1 堆栈指针 (SP) 的值

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

5.8.2 向量基址寄存器 (VBR) 的值

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

5.8.3 在地址错误异常处理的堆栈存取中发生的地址错误

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

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

5.8.4 有关槽非法指令异常处理的注意事项

本 LSI 的槽非法指令异常处理的规格和以前的 SH-2 有以下不同点：

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

与此有关的本公司软件产品的对应情况如下：

(1) 编译程序

V.4 以后版本的编译程序不将该指令配置到延迟槽。

(2) 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。

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

第 6 章 中断控制器 (INTC)

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

6.1 特点

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

INTC 的框图如图 6.1 所示。

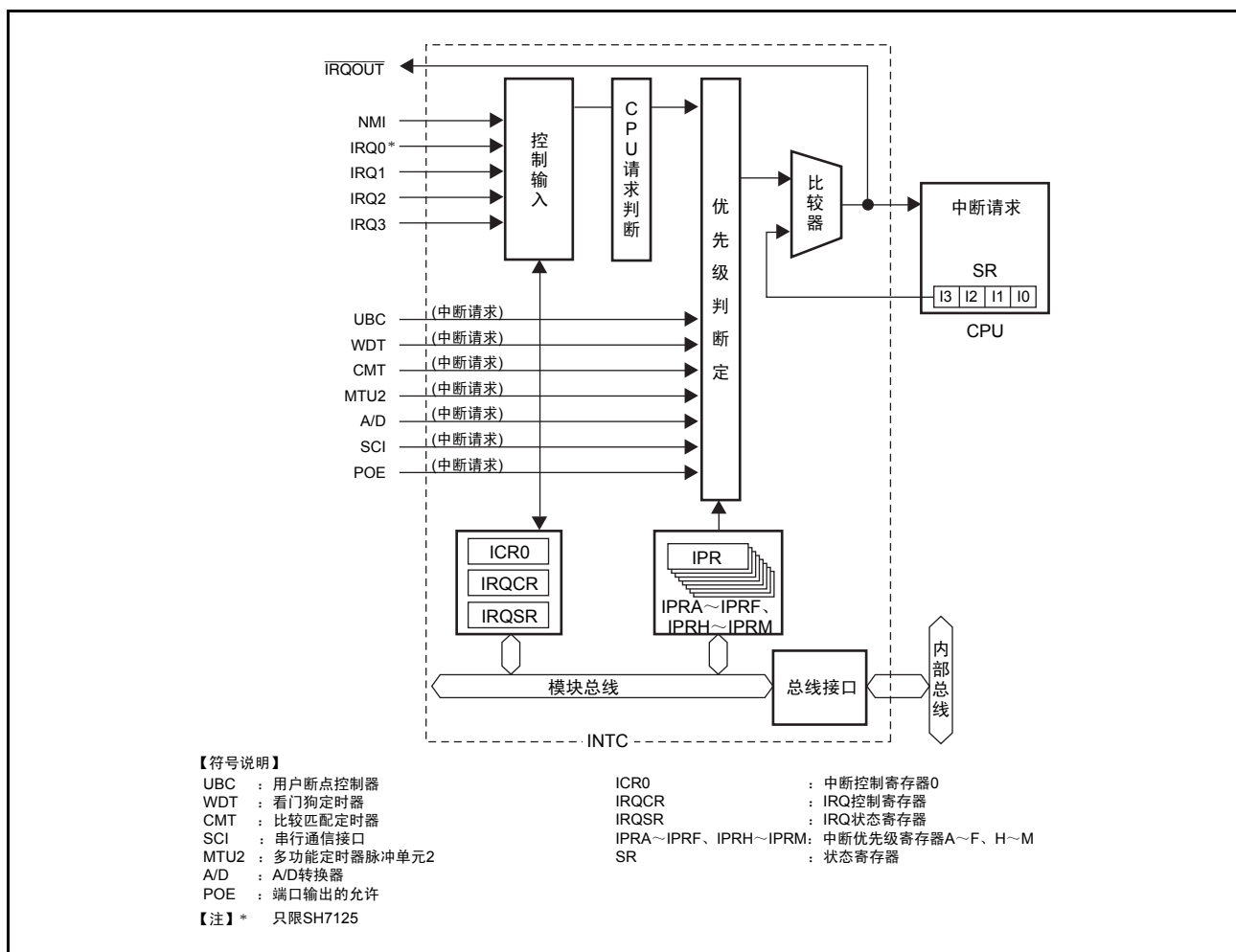


图 6.1 INTC 的框图

6.2 输入 / 输出管脚

INTC 的管脚如表 6.1 所示。

表 6.1 管脚结构

名称	管脚名	输入 / 输出	功 能
非屏蔽中断输入管脚	NMI	输入	输入不可屏蔽的中断请求信号
中断请求输入管脚	IRQ0 ~ IRQ3	输入	输入可屏蔽的中断请求信号
中断请求输出管脚	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 20 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 B	IPRB	R/W	H'0000	H'FFFFE908	8、16
中断优先级寄存器 C	IPRC	R/W	H'0000	H'FFFFE980	16
中断优先级寄存器 D	IPRD	R/W	H'0000	H'FFFFE982	16
中断优先级寄存器 E	IPRE	R/W	H'0000	H'FFFFE984	16
中断优先级寄存器 F	IPRF	R/W	H'0000	H'FFFFE986	16
中断优先级寄存器 H	IPRH	R/W	H'0000	H'FFFFE98A	16
中断优先级寄存器 I	IPRI	R/W	H'0000	H'FFFFE98C	16
中断优先级寄存器 J	IPRJ	R/W	H'0000	H'FFFFE98E	16
中断优先级寄存器 K	IPRK	R/W	H'0000	H'FFFFE990	16
中断优先级寄存器 L	IPRL	R/W	H'0000	H'FFFFE992	16
中断优先级寄存器 M	IPRM	R/W	H'0000	H'FFFFE994	16

6.3.1 中断控制寄存器 0（ICR0）

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

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

6.3.2 IRQ 控制寄存器 (IRQCR)

IRQCR 是 16 位寄存器，设定外部中断输入管脚 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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 8	—	全 0	R/W	保留位 读写的值总是 0。
7 6	IRQ31S IRQ30S	0 0	R/W R/W	IRQ3 的检测选择 选择 IRQ3 管脚的中断信号检测模式。 00: 在 IRQ3 输入管脚的低电平检测中断请求 01: 在 IRQ3 输入管脚的下降沿检测中断请求 10: 在 IRQ3 输入管脚的上升沿检测中断请求 11: 在 IRQ3 输入管脚的双边沿检测中断请求
5 4	IRQ21S IRQ20S	0 0	R/W R/W	IRQ2 的检测选择 选择 IRQ2 管脚的中断信号检测模式。 00: 在 IRQ2 输入管脚的低电平检测中断请求 01: 在 IRQ2 输入管脚的下降沿检测中断请求 10: 在 IRQ2 输入管脚的上升沿检测中断请求 11: 在 IRQ2 输入管脚的双边沿检测中断请求
3 2	IRQ11S IRQ10S	0 0	R/W R/W	IRQ1 的检测选择 选择 IRQ1 管脚的中断信号检测模式。 00: 在 IRQ1 输入管脚的低电平检测中断请求 01: 在 IRQ1 输入管脚的下降沿检测中断请求 10: 在 IRQ1 输入管脚的上升沿检测中断请求 11: 在 IRQ1 输入管脚的双边沿检测中断请求
1 0	IRQ01S IRQ00S	0 0	R/W R/W	IRQ0 的检测选择 (SH7125) 选择 IRQ0 管脚的中断信号检测模式。 00: 在 IRQ0 输入管脚的低电平检测中断请求 01: 在 IRQ0 输入管脚的下降沿检测中断请求 10: 在 IRQ0 输入管脚的上升沿检测中断请求 11: 在 IRQ0 输入管脚的双边沿检测中断请求 保留位 (SH7124) 读写的值总是 0。

6.3.3 IRQ 状态寄存器 (IRQSR)

IRQSR 是 16 位寄存器，表示外部中断输入管脚 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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

【注】* 对应的IRQ管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 总是读 1。
11	IRQ3L	*	R	表示 IRQ3 的管脚状态。 0: IRQ3 管脚为低电平 1: IRQ3 管脚为高电平
10	IRQ2L	*	R	表示 IRQ2 的管脚状态。 0: IRQ2 管脚为低电平 1: IRQ2 管脚为高电平
9	IRQ1L	*	R	表示 IRQ1 的管脚状态。 0: IRQ1 管脚为低电平 1: IRQ1 管脚为高电平
8	IRQ0L	*	R	表示 IRQ0 的管脚状态。(SH7125) 0: IRQ0 管脚为低电平 1: IRQ0 管脚为高电平 保留位 (SH7124) 总是读 1。
7 ~ 4	—	全 0	R/W	保留位 读写的值总是 0。
3	IRQ3F	0	R/W	表示 IRQ3 中断请求的状态。 在已设定电平检测时 0: 没有 IRQ3 中断请求 [清除条件] • 给 IRQ3 管脚输入高电平 1: 有 IRQ3 中断请求 [置位条件] • 给 IRQ3 管脚输入低电平 在已设定边沿检测时， 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 管脚的指定边沿

位	位名	初始值	R/W	说 明
2	IRQ2F	0	R/W	表示 IRQ2 中断请求的状态。 在已设定电平检测时 0: 没有 IRQ2 中断请求 [清除条件] • 给 IRQ2 管脚输入高电平 1: 有 IRQ2 中断请求 [置位条件] • 给 IRQ2 管脚输入低电平 在已设定边沿检测时 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 管脚的指定边沿
1	IRQ1F	0	R/W	表示 IRQ1 中断请求的状态。 在已设定电平检测时 0: 没有 IRQ1 中断请求 [清除条件] • 给 IRQ1 管脚输入高电平 1: 有 IRQ1 中断请求 [置位条件] • 给 IRQ1 管脚输入低电平 在已设定边沿检测时 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 管脚的指定边沿

位	位名	初始值	R/W	说 明
0	IRQ0F	0	R/W	表示 IRQ0 中断请求的状态。(SH7125) 在已设定电平检测时 0: 没有 IRQ0 中断请求 [清除条件] - 给 IRQ0 管脚输入高电平 1: 有 IRQ0 中断请求 [置位条件] - 给 IRQ0 管脚输入低电平 在已设定边沿检测时 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 IRQ 中断请求 [置位条件] - 检测到 IRQ0 管脚的指定边沿 保留位 (SH7124) 读写的值总是 0。

【注】 * 对应的 IRQ 管脚为高电平时此位是 1, 为低电平时是 0。

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

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

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

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

【注】 在此表中位名用一般名称表示，而在寄存器一览表中用模块名表示。

6.4 中断源

6.4.1 外部中断源

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

(1) NMI 中断

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

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

(2) IRQ3 ~ IRQ0 中断

IRQ 中断是由 IRQ3 ~ IRQ0 管脚输入引起的中断。能通过设定 IRQ 控制寄存器 (IRQCR) 的 IRQ 检测选择位 (IRQ31S、IRQ30S ~ IRQ01S、IRQ00S), 给每个管脚选择低电平检测、下降沿检测、上升沿检测或者双边沿检测。另外, 能通过中断优先级寄存器 A (IPRA), 在 0 ~ 15 范围内给每个管脚设定优先级。如果将 IRQ 中断设定为低电平检测, 就在 IRQ 管脚为低电平期间将中断请求信号送给 INTC, 当 IRQ 管脚变为高电平时, 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) 确认有无中断请求。

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

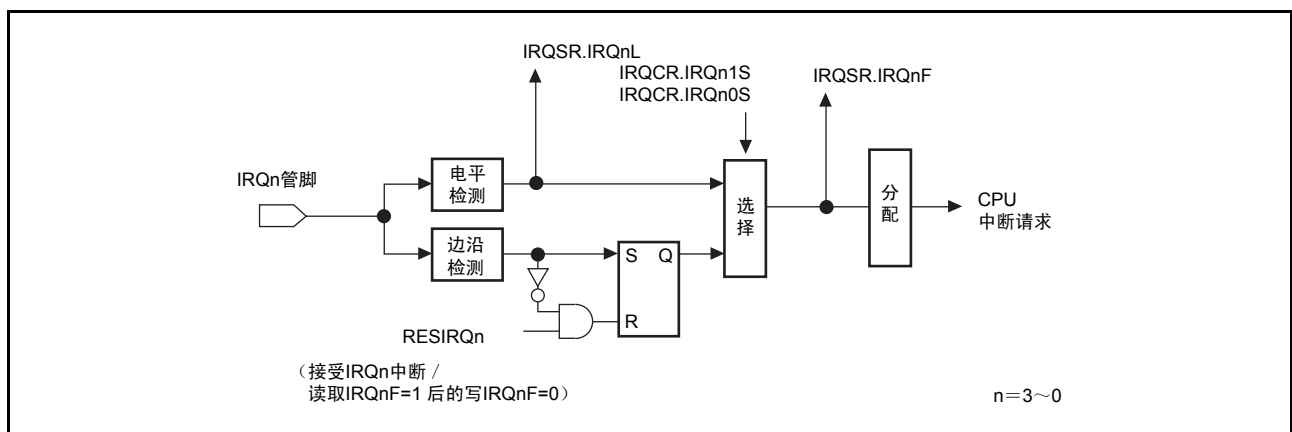


图 6.2 IRQ0 ~ IRQ3 中断控制

6.4.2 内部外围模块中断

内部外围模块中断是由以下所示的内部外围模块发生的中断。

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

6.4.3 用户断点中断

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

6.5 中断异常处理向量表

中断源、向量号、向量表地址偏移量和中断优先级如表 6.3 所示。

各中断源分别分配了不同的向量号和向量表地址偏移量，向量表地址由向量号和向量表地址偏移量算出。通过中断异常处理从向量表地址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。

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

表 6.3 中断源、向量地址和中断优先级一览表

中断发生源	名 称	向量号	向量表起始地址	IPR	默认优先级
用户断点		12	H'00000030	—	高 ↑ ↓ 低
外部管脚	NMI	11	H'0000002C	—	
	IRQ0（仅限 SH7125）	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		

中断发生源	名 称	向量号	向量表起始地址	IPR	默认优先级
MTU2_1	TGIA_1	96	H'00000180	IPRD7 ~ IPRD4	高 ↑

6.6 运行说明

6.6.1 中断运行的流程

以下说明发生中断时的运行流程，运行流程如图 6.3 所示。

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

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

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

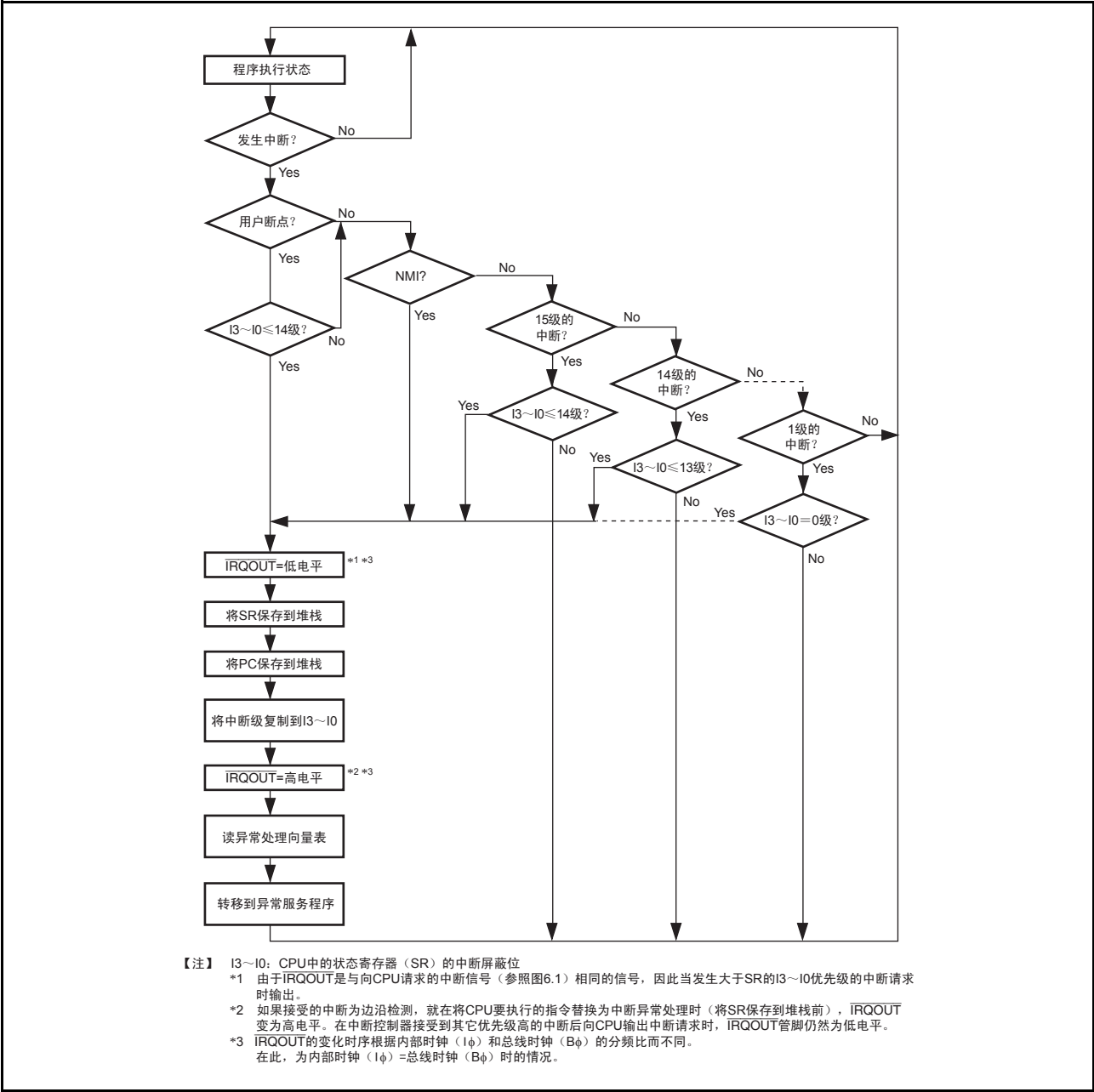


图 6.3 中断运行流程

6.6.2 中断异常处理结束后的堆栈状态

中断异常处理结束后的堆栈状态如图 6.4 所示。

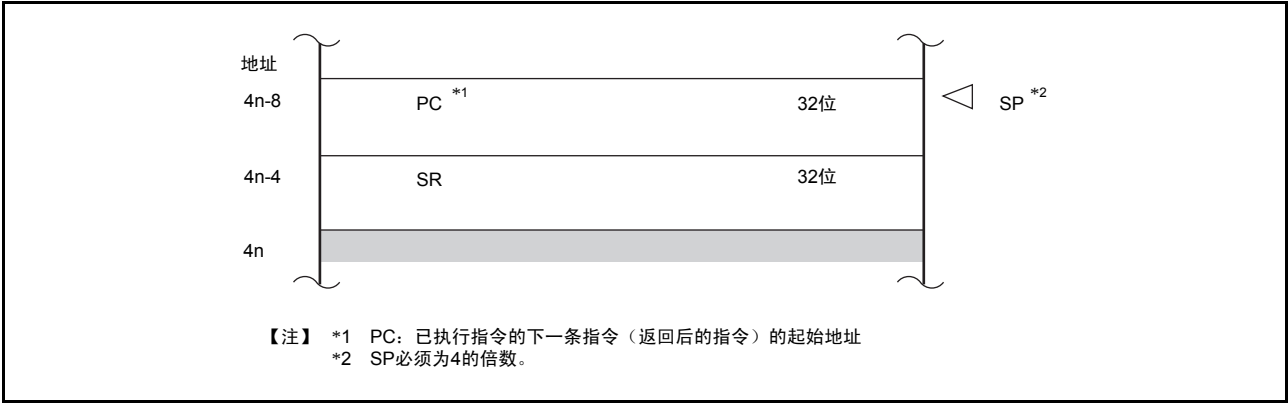


图 6.4 中断异常处理结束后的堆栈状态

6.7 中断响应时间

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

表 6.4 中断响应时间

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

【注】 m1 ~ m4 为下述存储器存取时需要的状态数：

m1: 保存 SR（写长字）

m2: 保存 PC（写长字）

m3: 读向量地址（读长字）

m4: 取中断服务程序的起始指令

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

6.8 使用注意事项

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

第 7 章 用户断点控制器 (UBC)

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

7.1 特点

1. 能设定如下的断点比较条件:

断点通道数: 2 个通道 (通道 A 和 B)

能将用户断点设定为通道 A、B 独立或者连续 (顺序) 的条件。

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

- 地址
能按位屏蔽 32 位 (地址) 的比较。能选择 L 地址总线 (LAB) 或者 I 地址总线 (IAB)。
- 数据
能屏蔽 32 位。
能选择 L 数据总线 (LDB) 或者 I 数据总线 (IDB)。
- 总线周期
取指令或者存取数据。
- 读或者写
- 操作数长度
支持字节、字和长字。

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

UBC 的框图如图 7.1 所示。

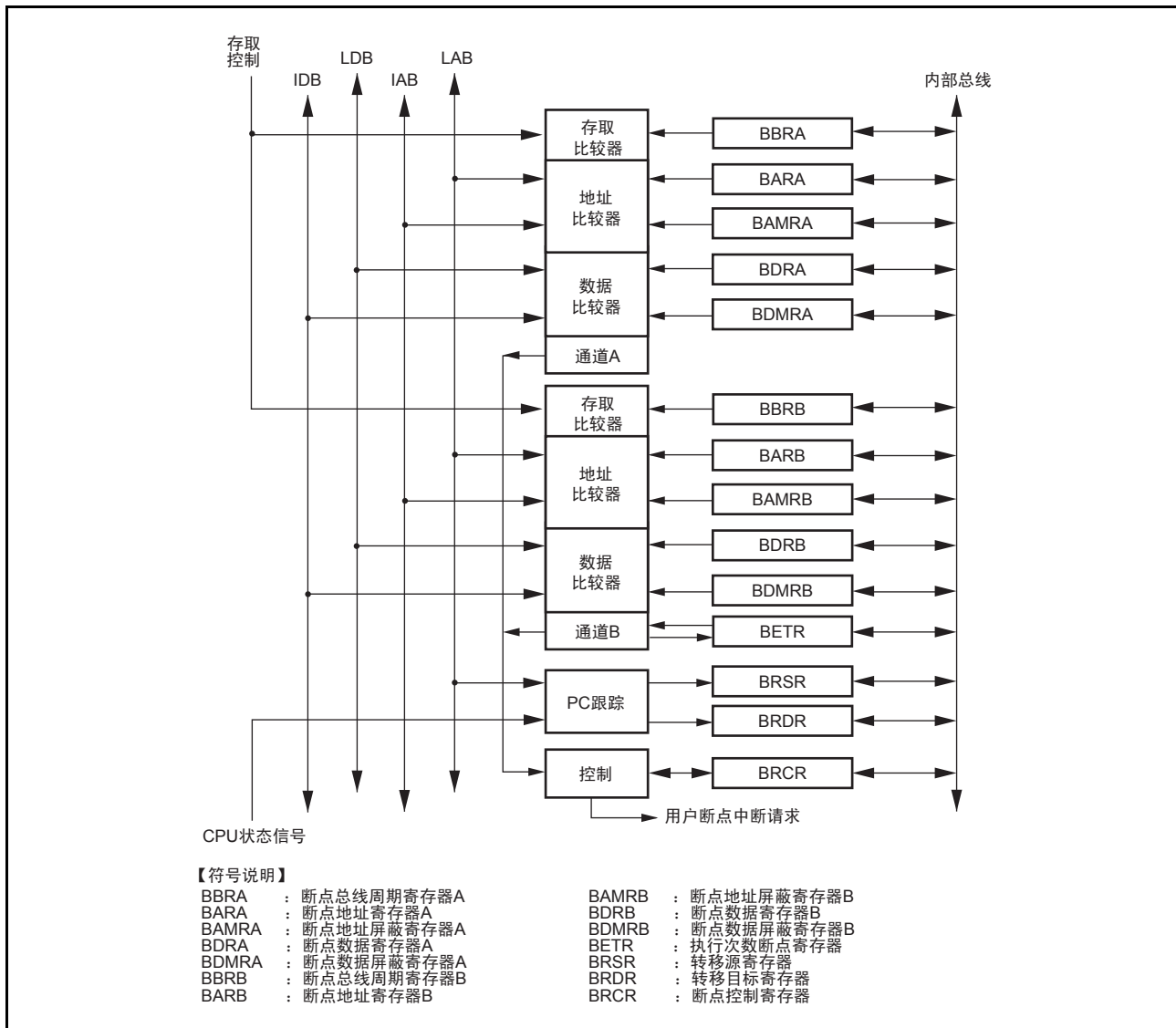


图 7.1 UBC 的框图

7.2 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 20 章 寄存器一览”。

表 7.1 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'FFFFF300	32
断点地址屏蔽寄存器 A	BAMRA	R/W	H'00000000	H'FFFFF304	32
断点总线周期寄存器 A	BBRA	R/W	H'0000	H'FFFFF308	16
断点数据寄存器 A	BDRA	R/W	H'00000000	H'FFFFF310	32
断点数据屏蔽寄存器 A	BDMRA	R/W	H'00000000	H'FFFFF314	32
断点地址寄存器 B	BARB	R/W	H'00000000	H'FFFFF320	32
断点地址屏蔽寄存器 B	BAMRB	R/W	H'00000000	H'FFFFF324	32
断点总线周期寄存器 B	BBRB	R/W	H'0000	H'FFFFF328	16
断点数据寄存器 B	BDRB	R/W	H'00000000	H'FFFFF330	32
断点数据屏蔽寄存器 B	BDMRB	R/W	H'00000000	H'FFFFF334	32
断点控制寄存器	BRCR	R/W	H'00000000	H'FFFFF3C0	32
转移源寄存器	BRSR	R	H'0xxxxxxx	H'FFFFF3D0	32
转移目标寄存器	BRDR	R	H'0xxxxxxx	H'FFFFF3D4	32
执行次数断点寄存器	BETR	R/W	H'0000	H'FFFFF3DC	16

7.2.1 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。

7.2.2 断点地址屏蔽寄存器 A（BAMRA）

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

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

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

位	位名	初始值	R/W	说 明
31 ~ 0	BAMA31 ~ BAMA0	全 0	R/W	断点地址屏蔽 A 在 BARA（BAA31 ~ BAA0）指定的通道 A 的断点地址位中，指定要屏蔽的位。 0: 断点地址位 BAA _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

7.2.3 断点总线周期寄存器 A (BBRA)

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

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	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 读写的值总是 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，本位无效。 000: 不进行条件比较 xx1: 断点条件包含 CPU 周期 x1x: 禁止设定 1xx: 禁止设定
7、6	CDA[1:0]	00	R/W	L 总线周期 / I 总线周期的选择 A 将 L 总线周期或者 I 总线周期选择为通道 A 断点条件的总线周期。 00: 不进行条件比较 01: 断点条件为 L 总线周期 10: 断点条件为 I 总线周期 11: 断点条件为 L 总线周期
5、4	IDA[1:0]	00	R/W	取指令 / 数据存取的选择 A 将取指令周期或者数据存取周期选择为通道 A 断点条件的总线周期。 00: 不进行条件比较 01: 断点条件为取指令周期 10: 断点条件为数据存取周期 11: 断点条件为取指令周期或者数据存取周期
3、2	RWA[1:0]	00	R/W	读 / 写的选择 A 将读周期或者写周期选择为通道 A 断点条件的总线周期。 00: 不进行条件比较 01: 断点条件为读周期 10: 断点条件为写周期 11: 断点条件为读或者写周期
1、0	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

7.2.4 断点数据寄存器 A（BDRA）

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

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

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

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

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

7.2.5 断点数据屏蔽寄存器 A (BDMRA)

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

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

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

位	位名	初始值	R/W	说 明
31 ~ 0	BDMA31 ~ BDMA0	全 0	R/W	断点数据屏蔽 A 在 BDRA (BDA31 ~ BDA0) 指定的通道 A 的断点数据位中，指定要屏蔽的位。 0: 断点数据位 BDA _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】 1. 如果断点条件包含数据总线的值，就必须指定操作数长度。
2. 如果将字节长度指定为断点条件，作为 BDMRA 中的断点屏蔽数据，就必须给 bit15 ~ 8 和 bit7 ~ 0 设定相同的字节数据。

7.2.6 断点地址寄存器 B (BARB)

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

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

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

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

7.2.7 断点地址屏蔽寄存器 B（BAMRB）

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

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

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

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

7.2.8 断点数据寄存器 B（BDRB）

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

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

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

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

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

7.2.9 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 中的断点屏蔽数据，就必须给 bit15 ~ 8 和 bit7 ~ 0 设定相同的字节数据。

7.2.10 断点总线周期寄存器 B（BBRB）

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

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	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 读写的值总是 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，本位无效。 000: 不进行条件比较 xx1: 断点条件中包含 CPU 周期 x1x: 禁止设定 1xx: 禁止设定
7、6	CDB[1:0]	00	R/W	L 总线周期 / I 总线周期的选择 B 将 L 总线周期或者 I 总线周期选择为通道 B 断点条件的总线周期。 00: 不进行条件比较 01: 断点条件为 L 总线周期 10: 断点条件为 I 总线周期 11: 断点条件为 L 总线周期
5、4	IDB[1:0]	00	R/W	取指令 / 数据存取的选择 B 将取指令周期或者数据存取周期选择为通道 B 断点条件的总线周期。 00: 不进行条件比较 01: 断点条件为取指令周期 10: 断点条件为数据存取周期 11: 断点条件为取指令周期或者数据存取周期
3、2	RWB[1:0]	00	R/W	读 / 写的选择 B 将读周期或者写周期选择为通道 B 断点条件的总线周期。 00: 不进行条件比较 01: 断点条件为读周期 10: 断点条件为写周期 11: 断点条件为读或者写周期
1、0	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

7.2.11 中断控制寄存器（BRCR）

BRCR 设定以下条件：

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

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

位：	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	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	R	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 ~ 20	—	全 0	R	保留位 读写的值总是 0。
19	UBIDB	0	R/W	用户断点的禁止 B 选择在满足通道 B 的断点条件时，是否禁止用户断点中断请求。 0：在满足断点条件时，允许用户断点中断请求 1：在满足断点条件时，禁止用户断点中断请求
18	—	0	R	保留位 读写的值总是 0。
17	UBIDA	0	R/W	用户断点的禁止 A 选择在满足通道 A 的断点条件时，是否禁止用户断点中断请求。 0：在满足断点条件时，允许用户断点中断请求 1：在满足断点条件时，禁止用户断点中断请求
16	—	0	R	保留位 读写的值总是 0。
15	SCMFCA	0	R/W	L 总线周期条件一致的标志 A 当满足给通道 A 设定的断点条件的 L 总线周期条件时，就将此标志置 1。 通过给此位写 0 清除此标志。 0：通道 A 的 L 总线周期条件不一致 1：通道 A 的 L 总线周期条件一致
14	SCMFCB	0	R/W	L 总线周期条件一致的标志 B 当满足给通道 B 设定的断点条件的 L 总线周期条件时，就将此标志置 1。 通过给此位写 0 清除此标志。。 0：通道 B 的 L 总线周期条件不一致 1：通道 B 的 L 总线周期条件一致

位	位名	初始值	R/W	说 明
13	SCMFDA	0	R/W	I 总线周期条件一致的标志 A 当满足给通道 A 设定的断点条件的 I 总线周期条件时, 就将此标志置 1。 通过给此位写 0 清除此标志。 0: 通道 A 的 I 总线周期条件不一致 1: 通道 A 的 I 总线周期条件一致
12	SCMFDB	0	R/W	I 总线周期条件一致的标志 B 当满足给通道 B 设定的断点条件的 I 总线周期条件时, 就将此标志置 1。 通过给此位写 0 清除此标志。 0: 通道 B 的 I 总线周期条件不一致 1: 通道 B 的 I 总线周期条件一致
11	PCTE	0	R/W	PC 跟踪的允许 0: 禁止 PC 跟踪 1: 允许 PC 跟踪
10	PCBA	0	R/W	PC 断点选择 A 选择通道 A 取指令周期的断点时序是在指令执行前还是在指令执行后。 0: 将通道 A 的 PC 断点设定在指令执行前 1: 将通道 A 的 PC 断点设定在指令执行后
9、8	—	全 0	R	保留位 读写的值总是 0。
7	DBEA	0	R/W	数据中断允许 A 选择通道 A 的断点条件是否包含数据总线条件。 0: 通道 A 的断点条件不包含数据总线条件 1: 通道 A 的断点条件包含数据总线条件
6	PCBB	0	R/W	PC 中断选择 B 选择通道 B 取指令周期的断点时序是在指令执行前还是在指令执行后。 0: 将通道 B 的 PC 断点设定在指令执行前 1: 将通道 B 的 PC 断点设定在指令执行后
5	DBEB	0	R/W	数据中断允许 B 选择通道 B 的断点条件是否包含数据总线条件。 0: 通道 B 的断点条件不包含数据总线条件 1: 通道 B 的断点条件包含数据总线条件
4	—	0	R	保留位 读写的值总是 0。
3	SEQ	0	R/W	顺序条件选择 选择通道 A 和 B 是 2 个独立条件还是连续的 1 个条件。 0: 在独立的条件下比较通道 A 和通道 B 1: 在连续的条件下比较通道 A 和通道 B (先是通道 A, 其次是通道 B)
2、1	—	全 0	R	保留位 读写的值总是 0。
0	ETBE	0	R/W	执行次数中断的允许 只对通道 B 将执行次数断点条件置为有效。如果此位是 1, 就在断点条件的发生次数和 BETR 寄存器指定的执行次数相等时, 请求用户断点中断。 0: 将通道 B 的执行次数断点条件置为无效 1: 将通道 B 的执行次数断点条件置为有效

7.2.12 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，此寄存器就指定暂停的次数。最大值为 $2^{12}-1$ 。每满足一次断点条件，BETR 就减 1。如果 BETR 为 H'0001 并且满足断点条件，就请求用户断点中断。

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

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

7.2.13 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 1 个在发生转移时被置 1 的标志位，此标志位在读 BRSR、或者将 PC 跟踪禁止状态设定为允许或者加电复位时被清 0，而其他位在复位时不被初始化。8 个 BRSR 寄存器具有队列结构，被保存的寄存器每转移一次就移位一次。

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

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

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

7.2.14 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 1 个在发生转移时被置 1 的标志位，此标志位在读 BRDR、或者将 PC 跟踪禁止状态设定为允许或者加电复位时被清 0，而其他位在复位时不被初始化。8 个 BRDR 寄存器具有队列结构，被保存的寄存器每转移一次就移位一次。

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

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

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

7.3 运行说明

7.3.1 用户断点运行的流程

从设定断点条件到用户断点异常处理的运行流程如下：

1. 将断点地址设定在断点地址寄存器 (BARA 或者 BARB)、要屏蔽的地址设定在断点地址屏蔽寄存器 (BAMRA 或者 BAMRB)、断点数据设定在断点数据寄存器 (BDRA 或者 BDRB)、要屏蔽的数据设定在断点数据屏蔽寄存器 (BDMRA 或者 BDMRB)、总线断点条件设定在断点总线周期寄存器 (BBRA 或者 BBRB)。BBRA 或者 BBRB 的 3 对控制位 (即 L 总线周期 / I 总线周期的选择位、取指令 / 数据存取的选择位和读 / 写的选择位) 中的任意 1 对是 B'00, 就不发生用户断点的暂停。将中断控制设定在 BRCCR 的位。必须在设定完其他全部的中断关联寄存器后, 设定 BBRA 或者 BBRB。
2. 当满足断点条件时, UBC 在将用户断点中断请求送给 CPU 的同时, 将各通道的 L 总线条件一致标志 (SCMFCA 或者 SCMFCA) 和 I 总线条件一致标志 (SCMFDA 或者 SCMFDB) 置位。
3. 能使用对应的条件一致标志 (SCMFCA、SCMFDA、SCMFCA、SCMFDB) 检查设定条件是否一致。根据条件一致标志置位, 而不能复位。为了能再次使用标志, 必须先写 0, 然后将标志复位。
4. 有时几乎同时发生通道 A 和通道 B 设定的断点条件一致, 此时即使送给 CPU 的用户断点中断请求只有 1 个, 也有可能设定 2 个条件一致标志。
5. 当选择 I 总线作为断点条件时, 必须注意以下事项:
 - CPU 连接 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, 并进行条件一致比较。
 - 将由 CPU 的 L 总线上的取指令引起的 I 总线周期 (包含读周期) 定义为 I 总线上的取指令周期, 其他定义为数据存取周期。
 - 当给 I 总线设定断点条件时, 即使在由 CPU 执行指令引起的 I 总线周期中条件一致, 也不能决定是在哪条指令接受用户断点。

7.3.2 取指令周期的断点

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

【注】 如果延迟转移指令不发生转移, 就不将后续的指令视为延迟槽。

3. 如果在断点条件中将断点设定在指令执行后, 就在执行和断点条件一致的指令后并在执行下一条指令前暂停。和执行前的暂停情况一样, 此功能不能用于因预取而取出的指令。如果对延迟转移指令及其延迟槽设定这种断点, 就不会在转移地址的第一条指令前暂停。
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断点条件设定断点数据。
5. 如果给取指令周期的断点条件设定 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 “7.3.1 用户断点运行的流程” 的第 5 项。

7.3.3 数据存取周期的断点

- 1. 对数据存取的断点，如果将 L 总线指定为断点条件，就对由执行指令存取的地址（和数据）进行条件比较并暂停；如果将 I 总线指定为断点条件，就对包含 I 总线上 CPU 的全部总线主控所发行的数据存取周期的地址（和数据）进行条件比较并暂停。有关发行到 I 总线上的 CPU 总线周期，请参照“7.3.1 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 7.2 所示。

表 7.2 数据存取周期地址和操作数长度的比较条件

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

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

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

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

7.3.4 顺序断点

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

7.3.5 被保存的程序计数器值

在暂停时，将需要继续执行的指令地址保存到堆栈，然后进入异常处理状态。如果将 L 总线指定为断点条件，就能决定暂停的指令（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，就不能决定暂停的指令。

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

7.3.6 PC 跟踪

1. 通过将 BRCR 的 PCTE 设定为 1，允许 PC 跟踪。如果发生转移（转移指令和中断异常），就将转移源地址和转移目标地址保存到 BRSR 和 BRDR。
2. 按照转移的种类，被保存到 BRSR 和 BRDR 的值分别如下：
 - 当因转移指令而发生转移时，转移指令的地址保存到 BRSR、转移目标指令的地址保存到 BRDR。
 - 当因中断或者一般异常而发生转移时，因发生异常而被保存的堆栈值保存到 BRSR、异常处理程序的起始地址保存到 BRDR。
3. BRSR 和 BRDR 由 4 组队列结构构成。当读被保存在 PC 跟踪寄存器的地址时，先读队列的第一项。BRSR 和 BRDR 共享读指针，必须按照 BRSR、BRDR 的顺序读取，队列只在读 BRDR 后进行移位。如果将 BRCR 的 PCTE 位从 OFF 切换到 ON，队列的值就无效。

7.3.7 使用例

(1) 指定为 L 总线取指令周期的断点条件

(例 1-1)

- 寄存器指定

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

指定条件：通道 A / 通道 B 独立模式

<通道 A>

地址：H'00000404、地址屏蔽：H'00000000

数据：H'00000000、数据屏蔽：H'00000000

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

<通道 B>

地址：H'00008010、地址屏蔽：H'00000006

数据：H'00000000、数据屏蔽：H'00000000

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

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

(例 1-2)

- 寄存器指定

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

指定条件：通道 A / 通道 B 顺序模式

<通道 A>

地址：H'00037226、地址屏蔽：H'00000000

数据：H'00000000、数据屏蔽：H'00000000

总线周期：L 总线 / 取指令（执行指令前） / 读 / 字

<通道 B>

地址：H'0003722E、地址屏蔽：H'00000000

数据：H'00000000、数据屏蔽：H'00000000

总线周期：L 总线 / 取指令（执行指令前） / 读 / 字

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

(例 1-3)

- 寄存器指定

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

指定条件：通道 A / 通道 B 独立模式

<通道 A>

地址：H'00027128、地址屏蔽：H'00000000

数据：H'00000000、数据屏蔽：H'00000000

总线周期：L 总线 / 取指令（执行指令前） / 写 / 字

<通道 B>

地址：H'00031415、地址屏蔽：H'00000000

数据：H'00000000、数据屏蔽：H'00000000

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

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

(例 1-4)

- 寄存器指定

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

指定条件：通道 A / 通道 B 顺序模式

<通道 A>

地址：H'00037226、地址屏蔽：H'00000000

数据：H'00000000、数据屏蔽：H'00000000

总线周期：L 总线 / 取指令（执行指令前） / 写 / 字

<通道 B>

地址：H'0003722E、地址屏蔽：H'00000000

数据：H'00000000、数据屏蔽：H'00000000

总线周期：L 总线 / 取指令（执行指令前） / 写 / 字

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

(例 1-5)

• 寄存器指定

BARA=H'00000500、BAMRA=H'00000000、BBRA=H'0057、BDRA=H'00000000、
BDMRA=H'00000000、

BARB=H'00001000、BAMRB=H'00000000、BBRB=H'0057、BDRB=H'00000000、
BDMRB=H'00000000、

BRCR=H'00000001、BETR=H'0005

指定条件：通道 A / 通道 B 独立模式

<通道 A>

地址：H'00000500、地址屏蔽：H'00000000

数据：H'00000000、数据屏蔽：H'00000000

总线周期：L 总线 / 取指令（执行指令前） / 读 / 长字

允许执行次数中断（5 次）

<通道 B>

地址：H'00001000、地址屏蔽：H'00000000

数据：H'00000000、数据屏蔽：H'00000000

总线周期：L 总线 / 取指令（执行指令前） / 读 / 长字

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

(例 1-6)

• 寄存器指定

BARA=H'00008404、BAMRA=H'00000FFF、BBRA=H'0054、BDRA=H'00000000、
BDMRA=H'00000000、

BARB=H'00008010、BAMRB=H'00000006、BBRB=H'0054、BDRB=H'00000000、
BDMRB=H'00000000、

BRCR=H'00000400

指定条件：通道 A / 通道 B 独立模式

<通道 A>

地址：H'00008404、地址屏蔽：H'00000FFF

数据：H'00000000、数据屏蔽：H'00000000

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

<通道 B>

地址：H'00008010、地址屏蔽：H'00000006

数据：H'00000000、数据屏蔽：H'00000000

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

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

(2) 指定为 L 总线数据存取周期的断点条件

(例 2-1)

• 寄存器指定

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

指定条件：通道 A / 通道 B 独立模式

<通道 A>

地址：H'00123456、地址屏蔽：H'00000000

数据：H'12345678 数据屏蔽：H'FFFFFFFF

总线周期：L 总线 / 数据存取 / 读 (断点条件不包含操作数长度)

<通道 B>

地址：H'000ABCDE、地址屏蔽：H'000000FF

数据：H'0000A512、数据屏蔽：H'00000000

总线周期：L 总线 / 数据存取 / 写 / 字

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

(3) 给 I 总线数据存取周期指定的断点条件

(例 3-1)

• 寄存器指定

BARA=H'00314154、BAMRA=H'00000000、BBRA=H'0194、BDRA=H'12345678、
 BDMRA=H'FFFFFFFF、
 BARB=H'00055555、BAMRB=H'00000000、BBRB=H'01A9、BDRB=H'00007878、
 BDMRB=H'00000F0F、
 BRCCR=H'00000080

指定条件：通道 A / 通道 B 独立模式

<通道 A>

地址：H'00314154、地址屏蔽：H'00000000

数据：H'12345678、数据屏蔽：H'FFFFFFFF

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

<通道 B>

地址：H'00055555、地址屏蔽：H'00000000

数据：H'00000078、数据屏蔽：H'0000000F

总线周期：I 总线 (CPU 周期) / 数据存取 / 写 / 字节

在通道 A 中，在外部存储空间的地址 H'00314154 取指令时发生用户断点的暂停；在通道 B 中，在 CPU 将字节数据 H'7x 写到外部存储空间的地址 H'00055555 时发生用户断点的暂停。

7.4 使用注意事项

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

第 8 章 总线状态控制器（BSC）

总线状态控制器（BSC）控制内部总线（L 总线、I 总线、外围总线）之间的数据发送和接收以及控制 CPU 对内部 FLASH、内部 RAM 和内部外围 I/O 的存取。

8.1 特点

- 内部 FLASH 和 RAM 接口
内部 FLASH 和 RAM 以 1 个周期（I_φ 同步）存取 32 位数据

8.2 地址映像

地址映像如表 8.1 所示。

表 8.1 地址映像

地址	存储器种类	大小		总线宽度
		128KB 版	64KB 版	
H'00000000 ~ H'0000FFFF	内部 FLASH	128KB	64KB	32
H'00010000 ~ H'0001FFFF			保留	
H'00020000 ~ H'83FFFFFF	保留	—	—	—
H'84000000 ~ H'8400FFFF	内部 FLASH 编程区	128KB	64KB	8
H'84010000 ~ H'8401FFFF			保留	
H'84020000 ~ H'FFFF9FFF	保留	—	—	—
H'FFFA000 ~ H'FFFBFFF	内部 RAM	8KB	8KB	32
H'FFFC000 ~ H'FFFFFFF	内部外围 I/O	128KB	64KB	8/16

8.3 内部 FLASH 和内部 RAM 的存取

和 I_φ 时钟同步，以 1 个时钟读内部 FLASH。有关编程和擦除请参照“第 17 章 闪存”。
和 I_φ 时钟同步，以 1 个时钟读写内部 RAM。详细内容请参照“第 18 章 RAM”。

8.4 内部外围 I/O 寄存器的存取

如表 8.2 所示，内部外围 I/O 寄存器由总线状态控制器 (BSC) 进行存取。

表 8.2 内部外围模块的连接总线宽度和存取周期数

内部外围模块	INTC	UBC	MTU2	POE2	WDT	SCI	ADC	CMT	PFC、端口
连接的总线宽度	16	16	16	16	16	8	16	16	16
存取周期数	$(1+n) \times \text{Iclk} + (1+m) \times \text{Bclk} + 2 \times \text{Pclk}$								

【注】 当 Bclk:Pclk=4:1 时， $m=0 \sim 3$

当 Iclk:Bclk=4:1 时， $n=0 \sim 3$

当 Bclk:Pclk=2:1 时， $m=0、1$

当 Iclk:Bclk=2:1 时， $n=0、1$

当 Bclk:Pclk=1:1 时， $m=0$

当 Iclk:Bclk=1:1 时， $n=0$

本产品采用了同步逻辑，各总线数据和各自的时钟上升沿同步进行数据的输入 / 输出。L 总线的存取为 1 个 Iclk 周期、I 总线的存取为 1 个 Bclk 周期、外围总线的存取为 2 个 Pclk 周期。当 Iclk:Bclk:Pclk=4:1:1 时，外围总线的写存取的时序例子如图 8.1 所示。连接 CPU 的 L 总线和 Iclk 同步输出数据。因为当 Iclk:Bclk=4:1 时，在 1 个 Bclk 周期中有 4 个 Iclk 周期，所以在将数据传送到 L 总线时，1 个 Bclk 周期有 4 个时序，对于将数据从 L 总线传送到 I 总线的时序的 Bclk 上升沿，最多需要 4 个 Iclk 周期的时间（在图 8.1 的例中为 4 个 Iclk 周期）。因此，当 Iclk:Bclk=4:1 时，在 $(\text{TBD}+n) \times \text{Iclk}$ 周期 ($n=0 \sim 3$) 期间，数据从 L 总线传送到 I 总线。将数据传送到 L 总线的时序和 Bclk 上升沿时序的关系取决于程序的执行状态。如图 8.1 所示，因为 Bclk:Pclk=1:1，所以 CPU 的存取时间为 $(\text{TBD}+n) \times \text{Iclk}$ 周期 + $1 \times \text{Bclk}$ 周期 + $2 \times \text{Pclk}$ 周期。

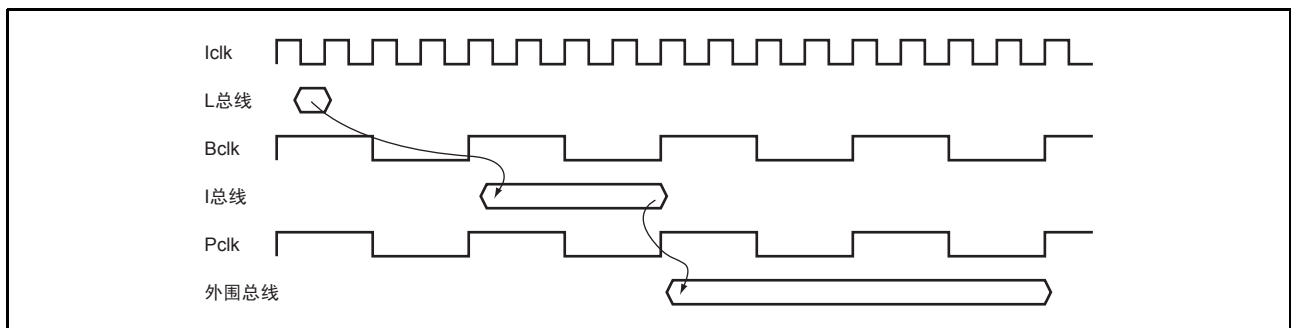


图 8.1 Iclk:Bclk:Pclk=4:1:1 时的外围总线的写存取时序

当 $Iclk:Bclk:Pclk=4:4:1$ 时，外围总线的写存取时序的例子如图 8.2 所示。连接 CPU 的 L 总线和 Iclk 同步输出数据。当 $Iclk:Bclk=1:1$ 时，数据从 L 总线传送到 I 总线需要 Iclk 周期 + Bclk 周期的时间。因为当 $Bclk:Pclk=4:1$ 时，在 1 个 Pclk 周期中有 4 个 Bclk 周期，所以在将数据从 I 总线传送到外围总线时，1 个 Pclk 周期有 4 个时序，对于将数据从 I 总线传送到外围总线的时序的 Pclk 上升沿，最多需要 4 个 Bclk 周期的时间（在图 8.2 的例中为 4 个 Bclk 周期）。因此，当 $Bclk:Pclk=P4:1$ 时，在 $(1+m) \times Iclk$ 周期 ($m=0 \sim 3$) 期间，将数据从 I 总线传送到外围总线。将数据传送到 I 总线的时序和 Pclk 的上升沿时序的关系取决于程序的执行状态。如图 8.2 所示，因为 $Iclk:Bclk=1:1$ ，所以 CPU 的存取时间为 $1 \times Iclk$ 周期 + $(1+m) \times Bclk$ 周期 + $2 \times Pclk$ 周期。

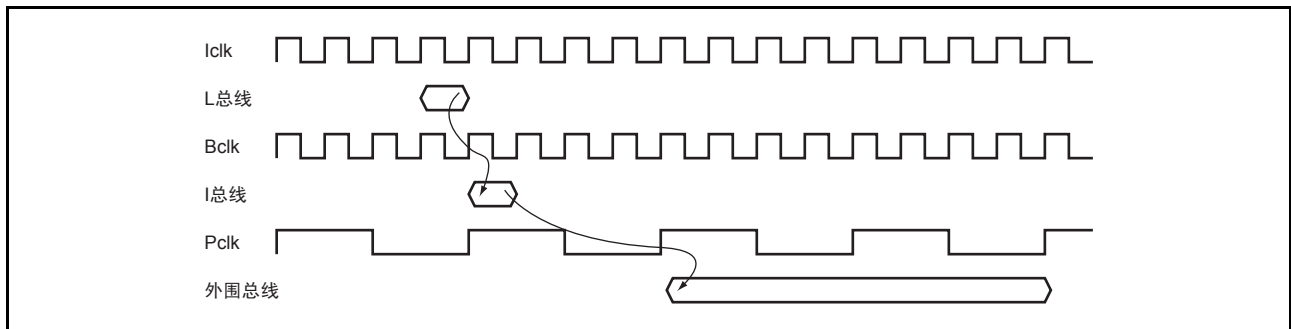


图 8.2 $Iclk:Bclk:Pclk=4:4:1$ 时的外围总线的写存取时序

第 9 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)。

9.1 特点

- 能进行最多 16 个 (SH7125) 或者 12 个 (SH7124) 的脉冲输入 / 输出和 3 个脉冲输入
- 各通道可选择 8 种计数器输入时钟 (通道 5 可选择 4 种)
- 通道 0 ~ 4 能设定以下的运行: 通过比较匹配进行的波形输出、输入捕捉功能、计数器的清除运行、多个定时器的计数器 (TCNT) 的同时写、通过比较匹配 / 输入捕捉进行的同时清除、通过计数器的同步运行进行的各寄存器的同步输入 / 输出、通过和同步运行组合进行的最多 12 相 PWM 输出
- 通道 0、3、4 能设定缓冲运行
- 通道 1、2 能各自独立设定相位计数模式
- 级联运行
- 进行内部 16 位总线的高速存取
- 28 种中断源
- 能进行寄存器数据的自动传送
- 能生成 A/D 转换器的转换开始触发
- 能设定模块待机模式
- 能通过通道 3、4 的联动运行设定互补 PWM 或者复位 PWM 的正负 3 相共 6 相的波形输出
- 能通过通道 0、3、4 联动运行设定使用互补 PWM 或者复位 PWM 的 AC 同步马达 (无刷 DC 马达) 驱动模式, 并可选择 2 种 (斩波和电平) 波形输出
- 通道 5 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 9.1 MTU2 的功能一览表

项 目	通道 0	通道 1	通道 2	通道 3	通道 4	通道 5
计数时钟	MP ϕ / 1 MP ϕ / 4 MP ϕ / 16 MP ϕ / 64 TCLKA TCLKB TCLKC TCLKD	MP ϕ / 1 MP ϕ / 4 MP ϕ / 16 MP ϕ / 64 MP ϕ / 256 TCLKA TCLKB	MP ϕ / 1 MP ϕ / 4 MP ϕ / 16 MP ϕ / 64 MP ϕ / 1024 TCLKA TCLKB TCLKC	MP ϕ / 1 MP ϕ / 4 MP ϕ / 16 MP ϕ / 64 MP ϕ / 256 MP ϕ / 1024 TCLKA TCLKB	MP ϕ / 1 MP ϕ / 4 MP ϕ / 16 MP ϕ / 64 MP ϕ / 256 MP ϕ / 1024 TCLKA TCLKB	MP ϕ / 1 MP ϕ / 4 MP ϕ / 16 MP ϕ / 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 / 输出管脚	TIOC0A TIOC0B TIOC0C TIOC0D	TIOC1A*1 TIOC1B*1	TIOC2A*1 TIOC2B*1	TIOC3A TIOC3B TIOC3C TIOC3D	TIOC4A TIOC4B TIOC4C TIOC4D	输入管脚 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	—	—	—	—	—	○
A/D 转换的开始触发	TGRA_0 的 比较匹配或者 输入捕捉 TGRE_0 的比 较匹配	TGRA_1 的 比较匹配或者 输入捕捉	TGRA_2 的 比较匹配或者 输入捕捉	TGRA_3 的 比较匹配或者 输入捕捉	TGRA_4 的 比较匹配或者 输入捕捉 在互补 PWM 模 式中 TCNT_4 的 下溢 (波谷)	—

项 目	通道 0	通道 1	通道 2	通道 3	通道 4	通道 5
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*2 • 比较匹配 / 输入捕捉 1B*2 • 上溢 • 下溢	4 个源 • 比较匹配 / 输入捕捉 2A*2 • 比较匹配 / 输入捕捉 2B*2 • 上溢 • 下溢	5 个源 • 比较匹配 / 输入捕捉 3A • 比较匹配 / 输入捕捉 3B • 比较匹配 / 输入捕捉 3C • 比较匹配 / 输入捕捉 3D • 上溢	5 个源 • 比较匹配 / 输入捕捉 4A • 比较匹配 / 输入捕捉 4B • 比较匹配 / 输入捕捉 4C • 比较匹配 / 输入捕捉 4D • 上溢 / 下溢	3 个源 • 比较匹配 / 输入捕捉 5U • 比较匹配 / 输入捕捉 5V • 比较匹配 / 输入捕捉 5W
A/D 转换的开始请求延迟功能	—	—	—	—	• TADCORA_4 和 TCNT_4 一致时, 请求 A/D 转换的开始 • TADCORB_4 和 TCNT_4 一致时, 请求 A/D 转换的开始	—
跳过中断功能	—	—	—	• 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

【注】 *1 只限 SH7125 的管脚。

*2 只限 SH7125 能使用输入捕捉

MTU2 的框图如图 9.1 所示。

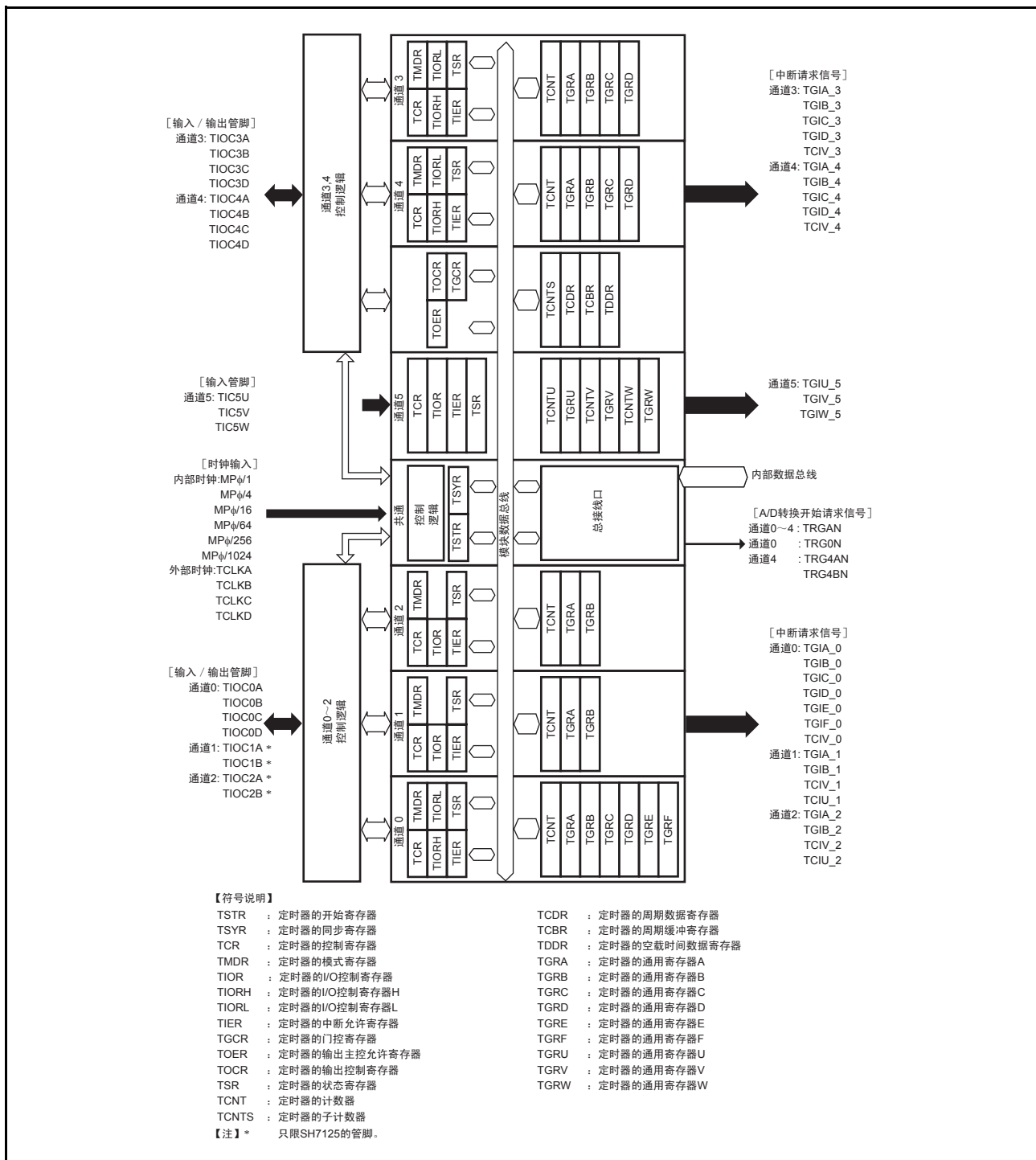


图 9.1 MTU2 的框图

9.2 输入 / 输出管脚

表 9.2 管脚结构

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

【注】 * 只限 SH7125 的管脚。

9.3 寄存器的说明

在 MTU2 中各通道有以下寄存器。有关这些寄存器的地址和各处理模式的寄存器状态，请参照“第 20 章 寄存器一览”。为了区分各通道的寄存器名，将通道 0 的 TCR 记载为 TCR_0。

表 9.3 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
定时器的控制寄存器_3	TCR_3	R/W	H'00	H'FFFC200	8、16、32
定时器的控制寄存器_4	TCR_4	R/W	H'00	H'FFFC201	8
定时器的模式寄存器_3	TMDR_3	R/W	H'00	H'FFFC202	8、16
定时器的模式寄存器_4	TMDR_4	R/W	H'00	H'FFFC203	8
定时器的 I/O 控制寄存器 H_3	TIORH_3	R/W	H'00	H'FFFC204	8、16、32
定时器的 I/O 控制寄存器 L_3	TIORL_3	R/W	H'00	H'FFFC205	8
定时器的 I/O 控制寄存器 H_4	TIORH_4	R/W	H'00	H'FFFC206	8、16
定时器的 I/O 控制寄存器 L_4	TIORL_4	R/W	H'00	H'FFFC207	8
定时器的中断允许寄存器_3	TIER_3	R/W	H'00	H'FFFC208	8、16
定时器的中断允许寄存器_4	TIER_4	R/W	H'00	H'FFFC209	8
定时器的输出主控允许寄存器	TOER	R/W	H'C0	H'FFFC20A	8
定时器的门控寄存器	TGCR	R/W	H'80	H'FFFC20D	8
定时器的输出控制寄存器 1	TOCR1	R/W	H'00	H'FFFC20E	8、16
定时器的输出控制寄存器 2	TOCR2	R/W	H'00	H'FFFC20F	8
定时器的计数器_3	TCNT_3	R/W	H'0000	H'FFFC210	16、32
定时器的计数器_4	TCNT_4	R/W	H'0000	H'FFFC212	16
定时器的周期数据寄存器	TCDR	R/W	H'FFFF	H'FFFC214	16、32
定时器的空载时间数据寄存器	TDDR	R/W	H'FFFF	H'FFFC216	16
定时器的通用寄存器 A_3	TGRA_3	R/W	H'FFFF	H'FFFC218	16、32
定时器的通用寄存器 B_3	TGRB_3	R/W	H'FFFF	H'FFFC21A	16
定时器的通用寄存器 A_4	TGRA_4	R/W	H'FFFF	H'FFFC21C	16、32
定时器的通用寄存器 B_4	TGRB_4	R/W	H'FFFF	H'FFFC21E	16
定时器的子计数器	TCNTS	R	H'0000	H'FFFC220	16、32
定时器的周期缓冲寄存器	TCBR	R/W	H'FFFF	H'FFFC222	16
定时器的通用寄存器 C_3	TGRC_3	R/W	H'FFFF	H'FFFC224	16、32
定时器的通用寄存器 D_3	TGRD_3	R/W	H'FFFF	H'FFFC226	16
定时器的通用寄存器 C_4	TGRC_4	R/W	H'FFFF	H'FFFC228	16、32
定时器的通用寄存器 D_4	TGRD_4	R/W	H'FFFF	H'FFFC22A	16
定时器的状态寄存器_3	TSR_3	R/W	H'C0	H'FFFC22C	8、16
定时器的状态寄存器_4	TSR_4	R/W	H'C0	H'FFFC22D	8
定时器的跳过中断设定寄存器	TITCR	R/W	H'00	H'FFFC230	8、16
定时器的跳过中断次数计数器	TITCNT	R	H'00	H'FFFC231	8
定时器的缓冲传送设定寄存器	TBTER	R/W	H'00	H'FFFC232	8
定时器的空载时间允许寄存器	TDER	R/W	H'01	H'FFFC234	8
定时器的输出电平缓冲寄存器	TOLBR	R/W	H'00	H'FFFC236	8
定时器的缓冲运行传送模式寄存器_3	TBTM_3	R/W	H'00	H'FFFC238	8、16
定时器的缓冲运行传送模式寄存器_4	TBTM_4	R/W	H'00	H'FFFC239	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'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	R/W	H'FFFF	H'FFFC244	16、32

寄存器名	略称	R/W	初始值	地址	存取长度
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
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

寄存器名	略称	R/W	初始值	地址	存取长度
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	TIO RV_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

9.3.1 定时器的控制寄存器 (TCR)

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

位:	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 9.4 和表 9.5。
4、3	CKEG[1:0]	00	R/W	时钟边沿 1、0 选择输入时钟的边沿。如果在双边沿对内部时钟进行计数, 输入时钟的周期就为 1 / 2 (例如: $MP\phi / 4$ 的双边沿 = $MP\phi / 2$ 的上升沿)。在通道 1、2 使用相位计数模式时, 忽视本设定, 优先设定相位计数模式。内部时钟的边沿选择在输入时钟为 $MP\phi / 4$ 或者慢于 $MP\phi / 4$ 时有效。如果选择 $MP\phi / 1$ 或者其他通道的上溢 / 下溢为输入时钟, 虽然能写值, 但是在运行时为初始值。 00: 在上升沿进行计数 01: 在下降沿进行计数 1x: 在双边沿进行计数
2 ~ 0	TPSC[2:0]	000	R/W	定时器的预定标器 2、1、0 选择 TCNT 的计数器时钟, 各通道能独立选择时钟源。详细内容请参照表 9.6 ~ 表 9.10。

【符号说明】 x: Don't care

表 9.4 CCLR2 ~ CCLR0 (通道 0、3、4)

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

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

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

表 9.5 CCLR2 ~ CCLR0 (通道 1、2)

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

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

*2 在通道 1、2 中, bit7 为保留位。读的值总是 0, 写操作无效。

表 9.6 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 管脚的输入进行计数
	1	0	1	外部时钟: 通过 TCLKB 管脚的输入进行计数
	1	1	0	外部时钟: 通过 TCLKC 管脚的输入进行计数
	1	1	1	外部时钟: 通过 TCLKD 管脚的输入进行计数

表 9.7 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 管脚的输入进行计数
	1	0	1	外部时钟: 通过 TCLKB 管脚的输入进行计数
	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 9.8 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 管脚的输入进行计数
	1	0	1	外部时钟: 通过 TCLKB 管脚的输入进行计数
	1	1	0	外部时钟: 通过 TCLKC 管脚的输入进行计数
	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 9.9 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 管脚的输入进行计数
	1	1	1	外部时钟: 通过 TCLKB 管脚的输入进行计数

表 9.10 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。读写的值总是 0。

9.3.2 定时器的模式寄存器（TMDR）

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

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

位	位名	初始值	R/W	说 明
7	—	0	—	保留位 读写的值总是 0。
6	BFE	0	R/W	缓冲运行 E 选择使 TGRE_0 和 TGRF_0 进行通常运行还是进行缓冲运行。 即使将 TGRF 用作缓冲寄存器，也会发生 TGRF 的比较匹配。 在通道 1、2、3、4 中，此位为保留位，读写的值总是 0。 0: TGRE_0 和 TGRF_0 进行通常运行 1: TGRE_0 和 TGRF_0 进行缓冲运行
5	BFB	0	R/W	缓冲运行 B 设定使 TGRB 进行通常运行还是组合 TGRB 和 TGRD 进行缓冲运行。如果将 TGRD 用作缓冲寄存器，除了互补 PWM 模式以外，不发生 TGRD 的输入捕捉 / 输出比较。但是，在互补 PWM 模式中发生 TGRD 的比较匹配。另外，如果在互补 PWM 模式的 Tb 区间发生比较匹配，TGFD 就被置位，所以必须将定时器的中断允许寄存器_3/4（TIER_3/4）的 TGIED 位设定为 0。 在没有 TGRD 的通道 1、2 中，此位为保留位，读写的值总是 0。 0: TGRB 和 TGRD 进行通常运行 1: TGRB 和 TGRD 进行缓冲运行
4	BFA	0	R/W	缓冲运行 A 设定使 TGRA 进行通常运行还是组合 TGRA 和 TGRC 进行缓冲运行。如果将 TGRC 用作缓冲寄存器，除了互补 PWM 模式以外，不发生 TGRC 的输入捕捉 / 输出比较。但是，在互补 PWM 模式中发生 TGRC 的比较匹配。另外，如果在互补 PWM 模式的 Tb 区间发生通道 4 的比较匹配，TGFC 就被置位，所以必须将定时器的中断允许寄存器_4（TIER_4）的 TGIEC 位设定为 0。 在没有 TGRC 的通道 1、2 中，此位为保留位，读写的值总是 0。 0: TGRA 和 TGRC 进行通常运行 1: TGRA 和 TGRC 进行缓冲运行
3～0	MD[3:0]	0000	R/W	模式 3～0 MD3～MD0 设定定时器的运行模式。 详细内容请参照表 9.11。

表 9.11 设定 MD3 ~ MD0 位的运行模式

bit3	bit2	bit1	bit0	说 明
MD3	MD2	MD1	MD0	
0	0	0	0	通常运行
0	0	0	1	禁止设定
0	0	1	0	PWM 模式 1
0	0	1	1	PWM 模式 2*1
0	1	0	0	相位计数模式 1*2
0	1	0	1	相位计数模式 2*2
0	1	1	0	相位计数模式 3*2
0	1	1	1	相位计数模式 4*2
1	0	0	0	复位同步 PWM 模式 *3
1	0	0	1	禁止设定
1	0	1	x	禁止设定
1	1	0	0	禁止设定
1	1	0	1	互补 PWM 模式 1 (在波峰进行传送) *3
1	1	1	0	互补 PWM 模式 2 (在波谷进行传送) *3
1	1	1	1	互补 PWM 模式 3 (在波峰和波谷进行传送) *3

【符号说明】 x: Don't care

【注】 *1 不能对通道 3、4 设定 PWM 模式 2。

*2 不能对通道 0、3、4 设定相位计数模式。

*3 只有通道 3 能设定复位同步 PWM 模式或者互补 PWM 模式。

如果对通道 3 设定复位同步 PWM 模式或者互补 PWM 模式，通道 4 的设定无效，自动服从通道 3 的设定。但是，不能对通道 4 设定复位同步 PWM 模式或者互补 PWM 模式。

不能对通道 0、1、2 设定复位同步 PWM 模式或者互补 PWM 模式。

9.3.3 定时器的 I/O 控制寄存器 (TIOR)

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

在 TMDR 的设定为通常运行、PWM 模式、相位计数模式时, 设定 TIOR。

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

如果将 TGRC 或者 TGRD 设定为缓冲运行, 本设定无效, 作为缓冲寄存器运行。

- TIORH_0、TIOR_1、TIOR_2、TIORH_3、TIORH_4

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

位	位名	初始值	R/W	说 明
7 ~ 4	IOB[3:0]	0000	R/W	I/O 控制 B3 ~ B0 IOB3 ~ IOB0 位设定 TGRB 的功能。 请参照下述表: TIORH_0: 表 9.12 TIOR_1: 表 9.14 TIOR_2: 表 9.15 TIORH_3: 表 9.16 TIORH_4: 表 9.18
3 ~ 0	IOA[3:0]	0000	R/W	I/O 控制 A3 ~ A0 IOA3 ~ IOA0 设定 TGRA 的功能。 请参照下述表: TIORH_0: 表 9.20 TIOR_1: 表 9.22 TIOR_2: 表 9.23 TIORH_3: 表 9.24 TIORH_4: 表 9.26

- 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。 请参照下述表: TIORL_0: 表 9.13 TIORL_3: 表 9.17 TIORL_4: 表 9.19
3 ~ 0	IOC[3:0]	0000	R/W	I/O 控制 C3 ~ C0 IOC3 ~ IOC0 位设定 TGRC 的功能。 请参照下述表: TIORL_0: 表 9.21 TIORL_3: 表 9.25 TIORL_4: 表 9.27

- 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 读写的值总是 0。
4 ~ 0	IOC[4:0]	00000	R/W	I/O 控制 C4 ~ C0 IOC4 ~ IOC0 位设定 TGRU/V/W_5 的功能。 详细内容请参照表 9.28。

表 9.12 TIORH_0 (通道 0)

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

【符号说明】 x: Don't care

【注】 * 从加电复位后到设定 TIOR 为止, 输出 0。

表 9.13 TIORL_0 (通道 0)

bit7	bit6	bit5	bit4	说 明	
IOD3	IOD2	IOD1	IOD0	TGRD_0 的功能	TIOC0D 管脚的功能
0	0	0	0	输出比较寄存器 *2	保持输出 *1
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配进行 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	0	0	0	输入捕捉寄存器 *2	在上升沿进行输入捕捉
1	0	0	1		在下降沿进行输入捕捉
1	0	1	x		在双边沿进行输入捕捉
1	1	x	x		捕捉输入源是通道 1 / 计数时钟, 通过 TCNT_1 的递增计数 / 递减计数进行输入捕捉

【符号说明】 x: Don't care

【注】 *1 从加电复位后到设定 TIOR 为止, 输出 0。

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

表 9.14 TIOR_1 (通道 1)

bit7	bit6	bit5	bit4	说 明	
IOB3	IOB2	IOB1	IOB0	TGRB_1 的功能	TIOC1B 管脚的功能 *2
0	0	0	0	TGRB_1 为输出比较寄存器	保持输出 *1
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	0	0	0	输入捕捉寄存器	在上升沿进行输入捕捉
1	0	0	1		在下降沿进行输入捕捉
1	0	1	x		在双边沿进行输入捕捉
1	1	x	x		通过发生 TGRC_0 的比较匹配 / 输入捕捉进行输入捕捉

【符号说明】 x: Don't care

【注】 *1 从加电复位后到设定 TIOR 为止, 输出 0。

*2 只限 SH7125 能使用 TIOC1B 管脚的输入 / 输出功能。

表 9.15 TIOR_2 (通道 2)

bit7	bit6	bit5	bit4	说 明	
IOB3	IOB2	IOB1	IOB0	TGRB_2 的功能	TIOC2B 管脚的功能 *2
0	0	0	0	TGRB_2 为输出比较寄存器	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	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 从加电复位后到设定 TIOR 为止, 输出 0。

*2 只限 SH7125 能使用 TIOC2B 管脚的输入 / 输出功能。

表 9.16 TIORH_3 (通道 3)

bit7	bit6	bit5	bit4	说 明	
IOB3	IOB2	IOB1	IOB0	TGRB_3 的功能	TIOC3B 管脚的功能
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】 * 从加电复位后到设定 TIOR 为止, 输出 0。

表 9.17 TIORL_3 (通道 3)

bit7	bit6	bit5	bit4	说 明	
IOD3	IOD2	IOD1	IOD0	TGRD_3 的功能	TIOC3D 管脚的功能
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 从加电复位后到设定 TIOR 为止, 输出 0。

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

表 9.18 TIORH_4 (通道 4)

bit7	bit6	bit5	bit4	说 明	
IOB3	IOB2	IOB1	IOB0	TGRB_4 的功能	TIOC4B 管脚的功能
0	0	0	0	TGRB_4 为输出比较寄存器	保持输出 *
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	x	0	0	输入捕捉寄存器	在上升沿进行输入捕捉
1	x	0	1		在下降沿进行输入捕捉
1	x	1	x		在双边沿进行输入捕捉

【符号说明】 x: Don't care

【注】 * 从加电复位后到设定 TIOR 为止, 输出 0。

表 9.19 TIORL_4 (通道 4)

bit7	bit6	bit5	bit4	说 明	
IOD3	IOD2	IOD1	IOD0	TGRD_4 的功能	TIOC4D 管脚的功能
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 从加电复位后到设定 TIOR 为止, 输出 0。

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

表 9.20 TIORH_0 (通道 0)

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

【符号说明】 x: Don't care

【注】 * 从加电复位后到设定 TIOR 为止, 输出 0。

表 9.21 TIORL_0 (通道 0)

bit3	bit2	bit1	bit0	说 明	
IOC3	IOC2	IOC1	IOC0	TGRC_0 的功能	TIOC0C 管脚的功能
0	0	0	0	输出比较寄存器 *2	保持输出 *1
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	0	0	0	输入捕捉寄存器 *2	在上升沿进行输入捕捉
1	0	0	1		在下降沿进行输入捕捉
1	0	1	x		在双边沿进行输入捕捉
1	1	x	x		捕捉输入源是通道 1 / 计数时钟, 通过 TCNT_1 的递增计数 / 递减计数进行输入捕捉

【符号说明】 x: Don't care

【注】 *1 从加电复位后到设定 TIOR 为止, 输出 0。

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

表 9.22 TIOR_1 (通道 1)

bit3	bit2	bit1	bit0	说 明	
IOA3	IOA2	IOA1	IOA0	TGRA_1 的功能	TIOC1A 管脚的功能 *2
0	0	0	0	输出比较寄存器	保持输出 *1
0	0	0	1		初始输出: 0 输出 比较匹配: 0 输出
0	0	1	0		初始输出: 0 输出 比较匹配: 1 输出
0	0	1	1		初始输出: 0 输出 比较匹配: 交替输出
0	1	0	0		保持输出
0	1	0	1		初始输出: 1 输出 比较匹配: 0 输出
0	1	1	0		初始输出: 1 输出 比较匹配: 1 输出
0	1	1	1		初始输出: 1 输出 比较匹配: 交替输出
1	0	0	0	输入捕捉寄存器	在上升沿进行输入捕捉
1	0	0	1		在下降沿进行输入捕捉
1	0	1	x		在双边沿进行输入捕捉
1	1	x	x		通过发生 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】 *1 从加电复位后到设定 TIOR 为止, 输出 0。

*2 只限 SH7125 能使用 TIOC1A 管脚的输入 / 输出功能。

表 9.23 TIOR_2 (通道 2)

bit3	bit2	bit1	bit0	说 明	
IOA3	IOA2	IOA1	IOA0	TGRA_2 的功能	TIOC2A 管脚的功能 *2
0	0	0	0	输出比较寄存器	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	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 从加电复位后到设定 TIOR 为止, 输出 0。

*2 只限 SH7125 能使用 TIOC2A 管脚的输入 / 输出功能。

表 9.24 TIORH_3 (通道 3)

bit3	bit2	bit1	bit0	说 明	
IOA3	IOA2	IOA1	IOA0	TGRA_3 的功能	TIOC3A 管脚的功能
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】 * 从加电复位后到设定 TIOR 为止, 输出 0。

表 9.25 TIORL_3 (通道 3)

bit3	bit2	bit1	bit0	说 明	
IOC3	IOC2	IOC1	IOC0	TGRC_3 的管脚	TIOC3C 管脚的功能
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 从加电复位后到设定 TIOR 为止, 输出 0。

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

表 9.26 TIORH_4 (通道 4)

bit3	bit2	bit1	bit0	说 明	
IOA3	IOA2	IOA1	IOA0	TGRA_4 的功能	TIOC4A 管脚的功能
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】 * 从加电复位后到设定 TIOR 为止, 输出 0。

表 9.27 TIORL_4 (通道 4)

bit3	bit2	bit1	bit0	说 明	
IOC3	IOC2	IOC1	IOC0	TGRC_4 的功能	TIOC4C 管脚的功能
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 从加电复位后到设定 TIOR 为止, 输出 0。

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

表 9.28 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 管脚的功能
0	0	0	0	0	比较匹配寄存器	比较匹配
0	0	0	0	1		禁止设定
0	0	0	1	x		禁止设定
0	0	1	x	x		禁止设定
0	1	x	x	x		禁止设定
1	0	0	0	0	输入捕捉寄存器	禁止设定
1	0	0	0	1		在上升沿进行输入捕捉
1	0	0	1	0		在下降沿进行输入捕捉
1	0	0	1	1		在双边沿进行输入捕捉
1	0	1	x	x		禁止设定
1	1	0	0	0		禁止设定
1	1	0	0	1		用于测量外部输入信号 Low 电平脉宽，在波谷进行捕捉
1	1	0	1	0		用于测量外部输入信号 Low 电平脉宽，在波峰进行捕捉
1	1	0	1	1		用于测量外部输入信号 Low 电平脉宽，在波峰和波谷进行捕捉
1	1	1	0	0		禁止设定
1	1	1	0	1		用于测量外部输入信号 High 电平脉宽，在波谷进行捕捉
1	1	1	1	0		用于测量外部输入信号 High 电平脉宽，在波峰进行捕捉
1	1	1	1	1		用于测量外部输入信号 High 电平脉宽，在波峰和波谷进行捕捉

【符号说明】 x: Don't care

9.3.4 定时器的比较匹配清除寄存器 (TCNTCMPCLR)

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

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

位	位名	初始值	R/W	说 明
7 ~ 3	—	全 0	R	保留位 读写的值总是 0。
2	CMPCLR5U	0	R/W	TCNT 比较清除 5U 允许或者禁止因 TGRU_5 的比较匹配 / 输入捕捉而产生的 TCNTU_5 的清除请求。 0: 禁止通过 TCNTU_5 和 TGRU_5 的比较匹配 / 输入捕捉将 TCNTU_5 清 H'0000 1: 允许通过 TCNTU_5 和 TGRU_5 的比较匹配 / 输入捕捉将 TCNTU_5 清 H'0000
1	CMPCLR5V	0	R/W	TCNT 比较清除 5V 允许或者禁止因 TGRV_5 的比较匹配 / 输入捕捉而产生的 TCNTV_5 的清除请求。 0: 禁止通过 TCNTV_5 和 TGRV_5 的比较匹配 / 输入捕捉将 TCNTV_5 清 H'0000 1: 允许通过 TCNTV_5 和 TGRV_5 的比较匹配 / 输入捕捉将 TCNTV_5 清 H'0000
0	CMPCLR5W	0	R/W	TCNT 比较清除 5W 允许或者禁止因 TCNTW_5 和 TGRW_5 的比较匹配 / 输入捕捉而产生的 TCNTW_5 的清除请求。 0: 禁止通过 TCNTW_5 和 TGRW_5 的比较匹配 / 输入捕捉将 TCNTW_5 清 H'0000 1: 允许通过 TCNTW_5 和 TGRW_5 的比较匹配 / 输入捕捉将 TCNTW_5 的清 H'0000

9.3.5 定时器的中断允许寄存器 (TIER)

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

- TIER_0、TIER_1、TIER_2、TIER_3、TIER_4

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

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

位	位名	初始值	R/W	说 明
1	TGIEB	0	R/W	TGR 中断允许 B 当 TSR 的 TGFB 位被置 1 时, 允许或者禁止 TGFB 位的中断请求 (TGIB)。 0: 禁止 TGFB 位的中断请求 (TGIB) 1: 允许 TGFB 位的中断请求 (TGIB)
0	TGIEA	0	R/W	TGR 中断允许 A 当 TSR 的 TGFA 位被置 1 时, 允许或者禁止 TGFA 位的中断请求 (TGIA)。 0: 禁止 TGFA 位的中断请求 (TGIA) 1: 允许 TGFA 位的中断请求 (TGIA)

• TIER2_0

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

位	位名	初始值	R/W	说 明
7	TTGE2	0	R/W	A/D 转换开始请求的允许 2 允许或者禁止因 TCNT_0 和 TGRE_0 的比较匹配而产生的 A/D 转换开始请求。 0: 禁止因 TCNT_0 和 TGRE_0 的比较匹配而产生的 A/D 转换开始请求 1: 允许因 TCNT_0 和 TGRE_0 的比较匹配而产生的 A/D 转换开始请求
6 ~ 2	—	全 0	R	保留位 读写的值总是 0。
1	TGIEF	0	R/W	TGR 中断允许 F 允许或者禁止因 TCNT_0 和 TGRF_0 的比较匹配而产生的中断请求。 0: 禁止 TGFE 位的中断请求 (TGIF) 1: 允许 TGFE 位的中断请求 (TGIF)
0	TGIEE	0	R/W	TGR 中断允许 E 允许或者禁止因 TCNT_0 和 TGRE_0 的比较匹配而产生的中断请求。 0: 禁止 TGEE 位的中断请求 (TGIE) 1: 允许 TGEE 位的中断请求 (TGIE)

• TIER_5

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

位	位名	初始值	R/W	说 明
7 ~ 3	—	全 0	R	保留位 读写的值总是 0。
2	TGIE5U	0	R/W	TGR 中断允许 5U 当 TSR_5 的 CMFU5 位被置 1 时, 允许或禁止 CMFU5 位的中断请求 (TGIU_5)。 0: 禁止 TGIU_5 的中断请求 1: 允许 TGIU_5 的中断请求
1	TGIE5V	0	R/W	TGR 中断允许 5V 当 TSR_5 的 CMFV5 位被置 1 时, 允许或者禁止 CMFV5 位的中断请求 (TGIV_5)。 0: 禁止 TGIV_5 的中断请求 1: 允许 TGIV_5 的中断请求
0	TGIE5W	0	R/W	TGR 中断允许 5W 当 TSR_5 的 CMFW5 位被置 1 时, 允许或者禁止 CMFW5 位的中断请求 (TGIW_5)。 0: 禁止 TGIW_5 的中断请求 1: 允许 TGIW_5 的中断请求

9.3.6 定时器的状态寄存器 (TSR)

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

- TSR_0、TSR_1、TSR_2、TSR_3、TSR_4

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

【注】*1 为了清除标志，只能在读1后写0。

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

位	位名	初始值	R/W	说 明
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，读写的值总是 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 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 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 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 前发生下一个标志的置位，即使写 0 也不清除标志，所以必须重新读 1 再写 0。

• TSR2_0

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	TGFF	TGFE
初始值:	1	1	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/(W)*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 读写的值总是 1。
5 ~ 2	—	全 0	R	保留位 读写的值总是 0。
1	TGFF	0	R/(W)*1	比较匹配标志 F 是表示 TCNT_0 和 TGRF_0 比较匹配的状态标志。 [置位条件] • 在将 TGRF_0 用作比较寄存器的情况下，当 TCNT_0=TGRF_0 时 [清除条件] • 在 TGFF=1 的状态下，当读 TGFF 后，给 TGFF 写 0 时 *2
0	TGFE	0	R/(W)*1	比较匹配标志 E 是表示 TCNT_0 和 TGRE_0 比较匹配的状态标志。 [置位条件] • 在将 TGRE_0 用作比较寄存器的情况下，当 TCNT_0=TGFE_0 时 [清除条件] • 在 TGFE=1 的状态下，当读 TGFE 后，给 TGFE 写 0 时 *2

【注】 *1 为了清除标志，只能在读 1 后写 0。

*2 如果在读 1 后写 0 前发生下一个比较匹配标志的置位，即使写 0 也不清除标志，所以必须重新读 1 再写 0。

• TSR_5

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

【注】*1 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说 明
7 ~ 3	—	全 0	R	保留位 读写的值总是 0。
2	CMFU5	0	R/(W)*1	比较匹配 / 输入捕捉标志 U5 是表示 TGRU_5 的输入捕捉或者比较匹配的状态标志。为了清除标志，只能写 0。 [置位条件] - 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 用作测量外部输入信号脉宽的情况下，当将 TCNTU_5 的值传送到 TGRU_5 时 *2 [清除条件] - 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 用作测量外部输入信号脉宽的情况下，当将 TCNTV_5 的值传送到 TGRV_5 *2 时 [清除条件] - 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 用作测量外部输入信号脉宽的情况下，将 TCNTW_5 的值传送到 TGRW_5 时 *2 [清除条件] - 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 (TIOUR_5/V_5/W_5) 的 IOC 位设定要传送的时序。

9.3.7 定时器的缓冲运行传送模式寄存器 (TBTM)

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

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

位	位名	初始值	R/W	说 明
7 ~ 3	—	全 0	R	保留位 读写的值总是 0。
2	TTSE	0	R/W	时序的选择 E 设定在缓冲运行时 TGRF_0 到 TGRE_0 的传送时序。 在通道 3、4 中，此位为保留位，读写的值总是 0。另外，在 PWM 以外的模式中使用通道 0 时，不能将本位设定为 1。 0: 当发生通道 0 的比较匹配 E 时 1: 当清除 TCNT_0 时
1	TTSB	0	R/W	时序的选择 B 设定在各通道的缓冲运行时 TGRD 到 TGRB 的传送时序。另外，在 PWM 以外的模式使用的通道，不能将本位设定为 1。 0: 当各通道发生比较匹配 B 时 1: 当清除各通道的 TCNT 时
0	TTSA	0	R/W	时序的选择 A 设定在各通道的缓冲运行时 TGRC 到 TGRA 的传送时序。另外，在 PWM 以外的模式使用的通道，不能将本位设定为 1。 0: 当各通道发生比较匹配 A 时 1: 当清除各通道的 TCNT 时

9.3.8 定时器的输入捕捉控制寄存器 (TICCR)

TICCR 是 8 位可读写寄存器，控制 TCNT_1 和 TCNT_2 级联时的输入捕捉条件。在 MTU2 中，通道 1 有 1 个 TICCR。

位:	7	6	5	4	3	2	1	0
	-	-	-	-	I2BE	I2AE	I1BE	I1AE
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7 ~ 4	—	全 0	R	保留位 读写的值总是 0。
3	I2BE	0	R/W	输入捕捉的允许 选择是否将 TIOC2B 管脚追加到 TGRB_1 的输入捕捉条件。 0: 不将 TIOC2B 管脚追加到 TGRB_1 的输入捕捉条件 1: 将 TIOC2B 管脚追加到 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 管脚追加到 TGRA_1 的输入捕捉条件。 0: 不将 TIOC2A 管脚追加到 TGRA_1 的输入捕捉条件 1: 将 TIOC2A 管脚追加到 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 管脚追加到 TGRB_2 的输入捕捉条件。 0: 不将 TIOC1B 管脚追加到 TGRB_2 的输入捕捉条件 1: 将 TIOC1B 管脚追加到 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 管脚追加到 TGRA_2 的输入捕捉条件。 0: 不将 TIOC1A 管脚追加到 TGRA_2 的输入捕捉条件 1: 将 TIOC1A 管脚追加到 TGRA_2 的输入捕捉条件

【注】 只限 SH7125 使用本功能。在 SH7124 中必须写 H'00。

9.3.9 定时器的 A/D 转换开始请求的控制寄存器 (TADCR)

TADCR 是 16 位可读写寄存器，允许或者禁止 A/D 转换开始请求以及设定 A/D 转换开始请求是否与跳中断功能联动。在 MTU2 中，通道 4 有 1 个 TADCR。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BF[1:0]		-	-	-	-	-	-	UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
初始值:	0	0	0	0	0	0	0	0	0	0*	0	0*	0*	0*	0*	0*
R/W:	R/W	R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

【注】* 除了互补PWM模式以外，不能置1。

位	位名	初始值	R/W	说 明
15、14	BF[1:0]	00	R/W	TADCOBRA/B_4 传送时序的选择 选择 TADCOBRA/B_4 到 TADCORA/B_4 的传送时序。 详细内容请参照表 9.29。
13 ~ 8	—	全 0	R	保留位 读写的值总是 0。
7	UT4AE	0	R/W	递增计数 TRG4AN 的允许 在进行 TCNT_4 的递增计数时，允许或者禁止 A/D 转换的开始请求 (TRG4AN)。 0: 在进行 TCNT_4 的递增计数时，禁止 A/D 转换的开始请求 (TRG4AN) 1: 在进行 TCNT_4 的递增计数时，允许 A/D 转换的开始请求 (TRG4AN)
6	DT4AE	0*	R/W	递减计数 TRG4AN 的允许 在进行 TCNT_4 的递减计数时，允许或者禁止 A/D 转换的开始请求 (TRG4AN)。 0: 在进行 TCNT_4 的递减计数时，禁止 A/D 转换的开始请求 (TRG4AN) 1: 在进行 TCNT_4 的递减计数时，允许 A/D 转换的开始请求 (TRG4AN)
5	UT4BE	0	R/W	递增计数 TRG4BN 的允许 在进行 TCNT_4 的递增计数时，允许或者禁止 A/D 转换的开始请求 (TRG4BN)。 0: 在进行 TCNT_4 的递增计数时，禁止 A/D 转换的开始请求 (TRG4BN) 1: 在进行 TCNT_4 的递增计数时，允许 A/D 转换的开始请求 (TRG4BN)
4	DT4BE	0*	R/W	递减计数 TRG4BN 的允许 在进行 TCNT_4 的递减计数时，允许或者禁止 A/D 转换的开始请求 (TRG4BN)。 0: 在进行 TCNT_4 的递减计数时，禁止 A/D 转换的开始请求 (TRG4BN) 1: 在进行 TCNT_4 的递减计数时，允许 A/D 转换的开始请求 (TRG4BN)

位	位名	初始值	R/W	说 明
3	ITA3AE	0*	R/W	允许 TGIA_3 跳过中断功能的联动 选择 A/D 转换的开始请求 (TRG4AN) 是否与 TGIA_3 的跳过中断功能联动。 0: 不与 TGIA_3 的跳过中断功能联动 1: 与 TGIA_3 的跳过中断功能联动
2	ITA4VE	0*	R/W	允许 TCIV_4 跳过中断功能的联动 选择 A/D 转换的开始请求 (TRG4AN) 是否与 TCIV_4 的跳过中断功能联动。 0: 不与 TCIV_4 的跳过中断功能联动 1: 与 TCIV_4 的跳过中断功能联动
1	ITB3AE	0*	R/W	允许 TGIA_3 跳过中断功能的联动 选择 A/D 转换的开始请求 (TRG4BN) 是否与 TGIA_3 的跳过中断功能联动。 0: 不与 TGIA_3 的跳过中断功能联动 1: 与 TGIA_3 的跳过中断功能联动
0	ITB4VE	0*	R/W	允许 TCIV_4 跳过中断功能的联动 选择 A/D 转换的开始请求 (TRG4BN) 是否与 TCIV_4 的跳过中断功能联动。 0: 不与 TCIV_4 的跳过中断功能联动 1: 与 TCIV_4 的跳过中断功能联动

- 【注】 1. 禁止以 8 位为单位存取 TADCR，必须以 16 位为单位进行存取。
2. 在禁止跳过中断功能时（将定时器的跳过中断设定寄存器（TITCR）的 T3AEN、T4VEN 位或者 TITCR 的减少次数设定位（3ACOR 和 4VCOR）设定为 0 时），必须设定为不与跳过中断功能联动（将定时器的 A/D 转换开始请求控制寄存器（TADCR）的 ITA3AE、ITA4VE、ITB3AE、ITB4VE 位设定为 0）。
3. 在禁止跳过中断功能时，如果设定为与跳过中断功能联动，就不进行 A/D 转换的开始请求。
- * 除了互补 PWM 模式以外，不能置 1。

表 9.29 通过 BF1、BF0 位设定的传送时序

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

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

*2 除了互补 PWM 模式以外，禁止设定。

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

TADCORA/B_4 是 16 位可读写寄存器。当 TADCORA/B_4 和 TCNT_4 一致时, 就发生对应的 A/D 转换开始请求。

TADCORA/B_4 的初始值为 H'FFFF。

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。

9.3.11 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。

9.3.12 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。

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。

9.3.13 定时器的通用寄存器 (TGR)

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

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

TGRE_0 和 TGRF_0 用作比较寄存器, 当 TCNT_0 和 TGRE_0 一致时, 就会发生 A/D 转换开始请求。能将 TGRF 作为缓冲寄存器。TGR 和缓冲寄存器的组合为 TGRE-TGRF。

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

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 的初始值是 H'FFFF。

9.3.14 定时器的开始寄存器 (TSTR)

TSTR 是 8 位可读写寄存器, 选择通道 0 ~ 4 的 TCNT 的运行 / 停止。

TSTR_5 是 8 位可读写寄存器, 选择通道 5 的 TCNTU/V/W_5 的运行 / 停止。

在对 TMDR 设定运行模式时或者对 TCR 设定 TCNT 的计数时钟时, 必须停止 TCNT 的计数器运行。

• TSTR

位:	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 管脚为输出的状态下, 如果在运行中给 CST 位写 0, 就停止计数器并且保持 TIOC 管脚的输出比较的输出电平。如果在 CST 位为 0 的状态下写 TIOR, 管脚的输出电平就被更新为已设定的初始输出值。 0: 停止 TCNT_4、TCNT_3 的计数 1: TCNT_4、TCNT_3 计数
6	CST3	0	R/W	
5 ~ 3	—	全 0	R	保留位 读写的值总是 0。
2	CST2	0	R/W	计数器的开始 2 ~ 0 选择 TCNT 的运行或者停止。 在 TIOC 管脚为输出的状态下, 如果在运行中给 CST 位写 0, 就停止计数器并且保持 TIOC 管脚的输出比较的输出电平。如果在 CST 位为 0 的状态下写 TIOR, 管脚的输出电平就被更新为已设定的初始输出值。 0: 停止 TCNT_2 ~ TCNT_0 的计数 1: TCNT_2 ~ TCNT_0 计数
1	CST1	0	R/W	
0	CST0	0	R/W	

• TSTR_5

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

位	位名	初始值	R/W	说 明
7 ~ 3	—	全 0	R	保留位 读写的值总是 0。
2	CSTU5	0	R/W	计数器的开始 U5 选择 TCNTU_5 的运行或者停止。 0: 停止 TCNTU_5 的计数 1: TCNTU_5 计数
1	CSTV5	0	R/W	计数器的开始 V5 选择 TCNTV_5 的运行或者停止。 0: 停止 TCNTV_5 的计数 1: TCNTV_5 计数
0	CSTW5	0	R/W	计数器的开始 W5 选择 TCNTW_5 的运行或者停止。 0: 停止 TCNTW_5 的计数 1: TCNTW_5 计数

9.3.15 定时器的同步寄存器 (TSYR)

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

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

位	位名	初始值	R/W	说 明
7	SYNC4	0	R/W	定时器的同步 4、3 选择和其他通道的独立运行或者同步运行。 当选择同步运行时，能对多个 TCNT 进行同步预置，并且能通过其他通道的计数器清除进行同步清除。 要设定同步运行时，至少需要将 2 个通道的 SYNC 位设定为 1。要设定同步清除时，除了 SYNC 位以外，还需要通过 TCR 的 CCLR2 ~ CCLR0 位设定 TCNT 的清除源。 0: TCNT_4、TCNT_3 进行独立运行 (TCNT 的预置 / 清除与其他通道无关) 1: TCNT_4、TCNT_3 进行同步运行能进行 TCNT 的同步预置 / 同步清除
6	SYNC3	0	R/W	
5 ~ 3	—	全 0	R	保留位 读写的值总是 0。
2	SYNC2	0	R/W	定时器的同步 2 ~ 0 选择和其他通道的独立运行或者同步运行。 当选择同步运行时，能对多个 TCNT 进行同步预置，并且能通过其他通道的计数器清除进行同步清除。 要设定同步运行时，至少需要将 2 个通道的 SYNC 位设定为 1。要设定同步清除时，除了 SYNC 位以外，还需要通过 TCR 的 CCLR2 ~ CCLR0 位设定 TCNT 的清除源。 0: TCNT_2 ~ TCNT_0 进行独立运行 (TCNT 的预置 / 清除与其他通道无关) 1: TCNT_2 ~ TCNT_0 进行同步运行能进行 TCNT 的同步预置 / 同步清除
1	SYNC1	0	R/W	
0	SYNC0	0	R/W	

9.3.16 定时器的计数器同步开始寄存器 (TCSYSTR)

TCSYSTR 是 8 位可读写寄存器，进行 MTU2 计数器的同步开始。

位:	7	6	5	4	3	2	1	0
	SCH0	SCH1	SCH2	SCH3	SCH4	-	-	-
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	R

【注】* 为了对寄存器置位，只能写1。

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

【注】* 为了将寄存器置位，只能写 1。

9.3.17 定时器的读写允许寄存器 (TRWER)

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

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

位	位名	初始值	R/W	说 明
7 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	RWE	1	R/W	允许读写 允许 / 禁止误写防止对象寄存器的读写。 0: 禁止寄存器的读写 1: 允许寄存器的读写 [清除条件] • RWE=1 的状态下，当读 RWE 后，给 RWE 写 0 时

- 误写防止的对象寄存器 / 计数器

TCR_3、4、TMDR_3、4、TIO RH_3、4、TIO RL_3、4、TIER_3、4、TGRA_3、4、TGRB_3、4、TOER、TOCR1、TOCR2、TGCR、TCDR、TDDR 和 TCNT_3、4，共计 22 寄存器。

9.3.18 定时器的输出主控允许寄存器（TOER）

TOER 是 8 位可读写寄存器，允许或者禁止输出管脚的 TIOC4D、TIOC4C、TIOC3D、TIOC4B、TIOC4A、TIOC3B 的输出设定。如果不设定 TOER 的各位，就不能正确地输出这些管脚。对于通道 3、4，必须在设定通道 3、4 的 TIOR 前设定 TOER 的值。

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

位	位名	初始值	R/W	说 明
7、6	—	全 1	R	保留位 读写的值总是 1。
5	OE4D	0	R/W	主控的允许 TIOC4D 允许或者禁止 TIOC4D 管脚的 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 管脚的 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 管脚的 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 管脚的 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 管脚的 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 管脚的 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“9.3.19 定时器的输出控制寄存器 1（TOCR1）”和“9.3.20 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。

9.3.19 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】* 在加电复位后，只能写1次的1。写1后不能写0。

位	位名	初始值	R/W	说 明
7	—	0	R	保留位 读写的值总是 0。
6	PSYE	0	R/W	PWM 同步输出的允许 允许或者禁止与 PWM 周期同步的交替输出。 0: 禁止交替输出 1: 允许交替输出
5、4	—	全 0	R	保留位 读写的值总是 0。
3	TOCL	0	R/(W)*	TOC 寄存器的写禁止位 *1 禁止或者允许写 TOCR1 寄存器的 TOCS 位、OLSN 位、OLSP 位。 0: 允许写 TOCS 位、OLSN 位、OLSP 位 1: 禁止写 TOCS 位、OLSN 位、OLSP 位
2	TOCS	0	R/W	TOC 的选择位 对于互补 PWM 模式 / 复位同步 PWM 模式的输出电平的设定，是选择 TOCR1 的设定有效还是选择 TOCR2 的设定有效。 0: TOCR1 的设定有效 1: TOCR2 的设定有效
1	OLSN	0	R/W	输出电平的选择 N*2 在复位同步 PWM 模式 / 互补 PWM 模式中，选择反相的输出电平。请参照表 9.30。
0	OLSP	0	R/W	输出电平的选择 P*2 在复位同步 PWM 模式 / 互补 PWM 模式中，选择正相的输出电平。请参照表 9.31。

【注】 *1 能通过将 TOCL 位设定为 1，防止 CPU 失控时的误写。

*2 通过将 TOCS 位设定为 0，本设定有效。

表 9.30 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 9.31 输出电平的选择功能

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

OLSN=1、OLSP=1 时的互补 PWM 模式的输出例（1 相）如图 9.2 所示。

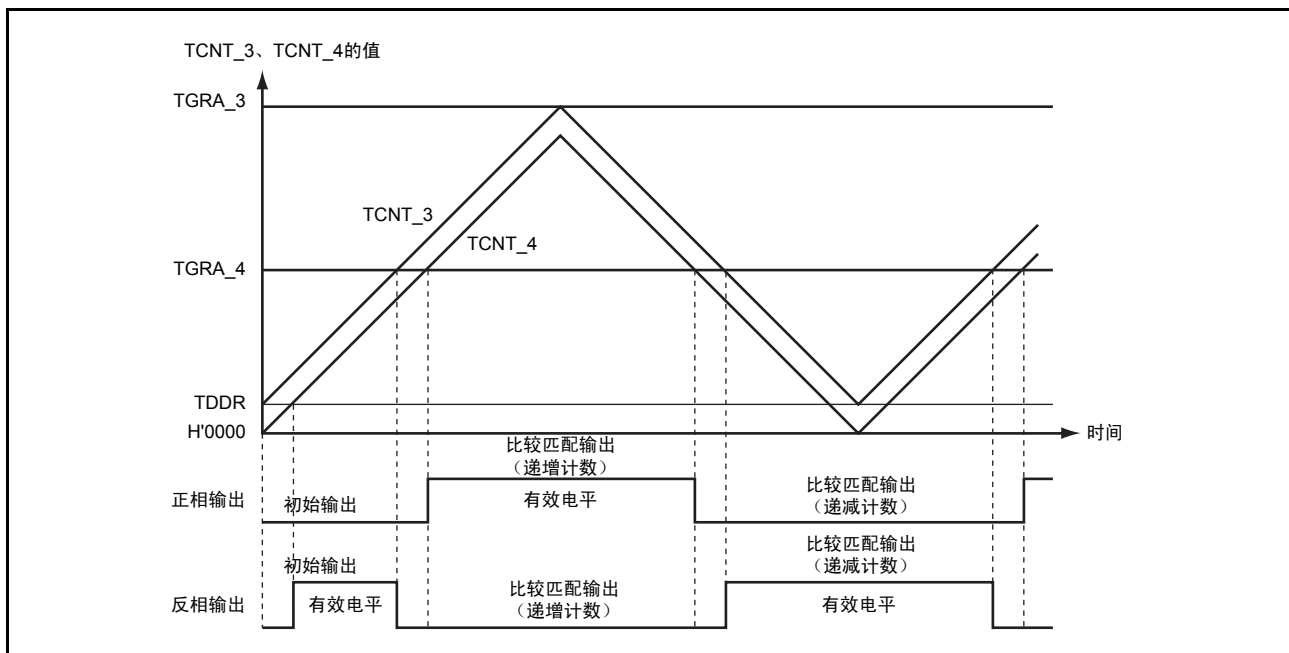


图 9.2 互补 PWM 模式的输出电平的例子

9.3.20 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 9.32。
5	OLS3N	0	R/W	输出电平的选择 3N* 在复位同步 PWM 模式 / 互补 PWM 模式中，选择 TIOC4D 的输出电平， 请参照表 9.33。
4	OLS3P	0	R/W	输出电平的选择 3P* 在复位同步 PWM 模式 / 互补 PWM 模式中，选择 TIOC4B 的输出电平。 请参照表 9.34。
3	OLS2N	0	R/W	输出电平的选择 2N* 在复位同步 PWM 模式 / 互补 PWM 模式中，选择 TIOC4C 的输出电平。 请参照表 9.35。
2	OLS2P	0	R/W	输出电平的选择 2P* 在复位同步 PWM 模式 / 互补 PWM 模式中，选择 TIOC4A 的输出电平。 请参照表 9.36。
1	OLS1N	0	R/W	输出电平的选择 1N* 在复位同步 PWM 模式 / 互补 PWM 模式中，选择 TIOC3D 的输出电平。 请参照表 9.37。
0	OLS1P	0	R/W	输出电平的选择 1P* 在复位同步 PWM 模式 / 互补 PWM 模式中，选择 TIOC3B 的输出电平。 请参照表 9.38。

【注】 * 通过将 TOCR1 的 TOCS 位设定为 1，本设定有效。

表 9.32 BF1、BF0 位的设定

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

表 9.33 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 9.34 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 9.35 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 9.36 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 9.37 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 9.38 TIOC3B 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

9.3.21 定时器的输出电平缓冲寄存器 (TOLBR)

TOLBR 是 TOCR2 的缓冲寄存器，设定互补 PWM 模式 / 复位同步 PWM 模式中的 PWM 输出电平，是 8 位可读写寄存器。

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

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

在缓冲运行中设定 PWM 输出电平时的设定步骤例子如图 9.3 所示。

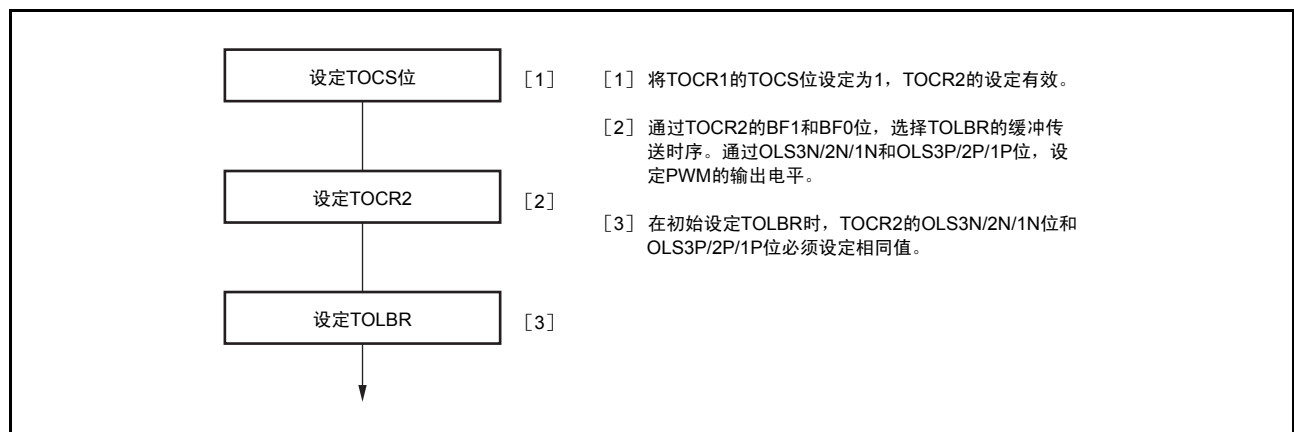


图 9.3 在缓冲运行中设定 PWM 输出电平时的设定步骤例子

9.3.22 定时器的门控寄存器 (TGCR)

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

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

位	位名	初始值	R/W	说 明
7	—	1	R	保留位 读写的值总是 1。
6	BDC	0	R/W	无刷 DC 马达 选择是将本寄存器的功能设定为有效还是无效。 0: 通常输出 1: 本寄存器的功能有效
5	N	0	R/W	反相输出 (N) 控制 当反相管脚 (TIOC3D 管脚、TIOC4C 管脚、TIOC4D 管脚) 为输出管脚时，选择电平输出或者复位同步 PWM / 互补 PWM 输出。 0: 电平输出 1: 复位同步 PWM / 互补 PWM 输出
4	P	0	R/W	正相输出 (P) 控制 当正相管脚 (TIOC3B 管脚、TIOC4A 管脚、TIOC4B 管脚) 为输出管脚时，选择电平输出或者复位同步 PWM / 互补 PWM 输出。 0: 电平输出 1: 复位同步 PWM / 互补 PWM 输出
3	FB	0	R/W	外部反馈信号的允许 选择是通过 MTU2 / 通道 0 的 TGRA、TGRB、TGRC 的输入捕捉信号还是通过给 TGCR 的 bit2 ~ 0 写 0 或者 1 自动进行正相 / 反相的输出转换。 0: 通过外部输入进行输出的转换 (输入源为通道 0 的 TGRA、TGRB、TGRC 的输入捕捉信号) 1: 通过软件进行输出的转换 (TGCR 的 UF、VF、WF 的设定值)
2	WF	0	R/W	输出相的转换 2 ~ 0 设定正相 / 反相的输出相 ON/OFF。这些位的设定只在本寄存器的 FB 位为 1 时有效。此时，bit2 ~ 0 的设定取代外部输入。请参照表 9.39。
1	VF	0	R/W	
0	UF	0	R/W	

表 9.39 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

9.3.23 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。

9.3.24 定时器的空载时间数据寄存器 (TDDR)

TDDR 是只在互补 PWM 模式中使用的 16 位寄存器，设定 TCNT_3 和 TCNT_4 计数器的偏移值。在互补 PWM 模式中清除 TCNT_3、TCNT_4 计数器后重新开始计数时，将 TDDR 寄存器的值装入到 TCNT_3 计数器并开始计数。TDDR 的初始值为 H'FFFF。

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。

9.3.25 定时器的周期数据寄存器 (TCDR)

TCDR 是只在互补 PWM 模式中使用的 16 位寄存器。TCDR 寄存器的值必须设定为 PWM 载波周期的 1/2。本寄存器在互补 PWM 模式中随时和 TCNTS 计数器比较，如果一致，TCNTS 计数器就转换计数方向（递减计数 → 递增计数）。TCDR 的初始值为 H'FFFF。

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。

9.3.26 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。

9.3.27 定时器的跳过中断设定寄存器 (TITCR)

TITCR 是 8 位可读写寄存器，禁止或者允许跳过中断功能以及设定跳过中断次数。MTU2 有 1 个 TITCR。

位:	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 的跳过中断功能。 0: 禁止 TGIA_3 的跳过中断功能 1: 允许 TGIA_3 的跳过中断功能
6 ~ 4	3ACOR[2:0]	000	R/W	以 0 ~ 7 次设定 TGIA_3 的跳过中断次数。* 详细内容请参照表 9.40。
3	T4VEN	0	R/W	T4VEN 禁止或者允许 TCIV_4 的跳过中断功能。 0: 禁止 TCIV_4 的跳过中断功能 1: 允许 TCIV_4 的跳过中断功能
2 ~ 0	4VCOR[2:0]	000	R/W	以 0 ~ 7 次设定 TCIV_4 的跳过中断次数。* 详细内容请参照表 9.41。

【注】 * 如果将跳过中断次数设定为 0，就不减少中断。

另外，在设定跳过中断次数前，必须在将 T3AEN、T4VEN 位设定为 0 后清除跳过中断次数计数器（TITCNT）。

表 9.40 通过 3ACOR2 ~ 3ACOR0 位设定跳过中断次数

bit6	bit5	bit4	说 明
3ACOR2	3ACOR1	3ACOR0	
0	0	0	不减少 TGIA_3 的中断
0	0	1	将 TGIA_3 的跳过中断次数设定为 1 次
0	1	0	将 TGIA_3 的跳过中断次数设定为 2 次
0	1	1	将 TGIA_3 的跳过中断次数设定为 3 次
1	0	0	将 TGIA_3 的跳过中断次数设定为 4 次
1	0	1	将 TGIA_3 的跳过中断次数设定为 5 次
1	1	0	将 TGIA_3 的跳过中断次数设定为 6 次
1	1	1	将 TGIA_3 的跳过中断次数设定为 7 次

表 9.41 通过 4VCOR2 ~ 4VCOR0 位设定跳过中断次数

bit2	bit1	bit0	说 明
4VCOR2	4VCOR1	4VCOR0	
0	0	0	不减少 TCIV_4 的中断
0	0	1	将 TCIV_4 的跳过中断次数设定为 1 次
0	1	0	将 TCIV_4 的跳过中断次数设定为 2 次
0	1	1	将 TCIV_4 的跳过中断次数设定为 3 次
1	0	0	将 TCIV_4 的跳过中断次数设定为 4 次
1	0	1	将 TCIV_4 的跳过中断次数设定为 5 次
1	1	0	将 TCIV_4 的跳过中断次数设定为 6 次
1	1	1	将 TCIV_4 的跳过中断次数设定为 7 次

9.3.28 定时器的跳过中断次数计数器 (TITCNT)

TITCNT 是 8 位可读计数器。MTU2 有 1 个 TITCNT。

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

位	位名	初始值	R/W	说 明
7	—	0	R	保留位 总是读 0。
6 ~ 4	3ACNT[2:0]	000	R	TGIA_3 中断计数器 如果将 TITCR 的 T3AEN 位设定为 1，就在出现 TGIA_3 中断源时进行递增计数（加 1）。 [清除条件] • TITCR 的 3ACOR2 ~ 3ACOR0 和 TITCNT 的 3ACNT2 ~ 3ACNT0 一致时 • TITCR 的 T3AEN 位为 0 时 • 当 TITCR 的 3ACOR2 ~ 3ACOR0 为 0 时
3	—	0	R	保留位 总是读 0。
2 ~ 0	4VCNT[2:0]	000	R	TCIV_4 中断计数器 如果将 TITCR 的 T4VEN 位设定为 1，就在出现 TCIV_4 中断源时进行递增计数（加 1）。 [清除条件] • 当 TITCR 的 4VCOR2 ~ 4VCOR0 和 TITCNT 的 4VCNT2 ~ 4VCNT0 一致时 • 当 TITCR 的 T4VEN 位为 0 时 • 当 TITCR 的 4VCOR2 ~ 4VCOR0 为 0 时

9.3.29 定时器的缓冲传送设定寄存器 (TBTER)

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

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

位	位名	初始值	R/W	说 明
7 ~ 2	—	全 0	R	保留位 读写的值总是 0。
1、0	BTE[1:0]	00	R/W	设定是否抑制缓冲寄存器* (互补 PWM 模式中使用的缓冲寄存器) 到暂存器的传送, 或者设定是否与跳过中断功能联动。详细内容请参照表 9.42。

【注】*对象缓冲寄存器

TGRC_3、TGRD_3、TGRC_4、TGRD_4、TCBR

表 9.42 BTE1 和 BTE0 的位设定

bit1	bit0	说 明
BTE1	BTE0	
0	0	不抑制缓冲寄存器到暂存器的传送*1 并且不与跳过中断功能联动
0	1	抑制缓冲寄存器到暂存器的传送
1	0	将缓冲寄存器到暂存器的传送*2 与跳过中断功能联动*2
1	1	禁止设定

【注】*1 按照 TMDR 的 MD3 ~ MD0 的设定进行传送。详细内容请参照“9.4.8 互补 PWM 模式”。

*2 在禁止跳过中断功能时 (在将定时器的跳过中断设定寄存器 (TITCR) 的 T3AEN、T4VEN 位或者 TITCR 的减少间隔次数设定 (3ACOR 和 4VCOR) 置 0 时), 必须设定为缓冲传送不与跳过中断功能联动 (将定时器的缓冲传送寄存器 (TBTER) 的 BTE1 置 0)。如果设定为缓冲传送与跳过中断功能联动, 就不进行缓冲传送。

9.3.30 定时器的空载时间允许寄存器 (TDER)

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

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

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

【注】 * 必须设定为 $TDDR \geq 1$ 。

9.3.31 定时器的波形控制寄存器 (TWCR)

TWCR 是 8 位可读写寄存器, 设定在互补 PWM 模式中发生 TNCT_3、TNCT_4 的同步计数器清除时的输出波形的控制以及设定是否进行由 TGRA_3 比较匹配引起的计数器清除。必须在停止 TCNT 运行的状态下进行 TWCR 的 CCE 位和 WRE 位的设定。

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

【注】* 除了互补PWM模式1以外, 不能置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 模式中进行由 TGRA_3 比较匹配引起的计数器清除。 0: 不进行由 TGRA_3 比较匹配引起的计数器清除 1: 进行由 TGRA_3 比较匹配引起的计数器清除 [置位条件] • CCE=0 的状态下, 当读 CCE 后, 给 CCE 写 1 时
6 ~ 1	—	全 0	R	保留位 读写的值总是 0。
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。本功能只在互补 PWM 模式的波谷 Tb 区间发生同步清除时保持波形。如果在其他区间发生同步清除, 就与 WRE 位的设定无关, 输出 TOCR 寄存器设定的初始值。另外, 如果在 TCNT_3、TCNT_4 刚开始的波谷 Tb 区间发生同步清除, 也输出 TOCR 寄存器设定的初始值。 有关互补 PWM 模式的波谷 Tb 区间, 请参照图 9.40。 0: 输出 TOCR 寄存器设定的初始输出值 1: 保持同步清除前的波形 [置位条件] • WRE=0 的状态下, 当读 WRE 后, 给 WRE 写 1 时

9.3.32 和总线主控的接口

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

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

9.4 运行说明

9.4.1 基本运行

各通道有 TCNT 和 TGR。TCNT 能进行递增计数运行、自由运行、周期计数器运行和外部事件计数运行。TGR 能分别用作输入捕捉寄存器和输出比较寄存器。

必须通过管脚功能控制器 (PFC) 设定 MTU2 的外部管脚功能。

(1) 计数器的运行

如果将 TSTR 的 CST0 ~ CST4 位、TSTR_5 的 CSTU5、CSTV5、CSTW5 位设定为 1，对应通道的 TCNT 就开始计数运行，能进行自由运行计数器的运行和周期计数器运行等。

(a) 计数运行的设定步骤例子

计数运行的设定步骤例子如图 9.4 所示。

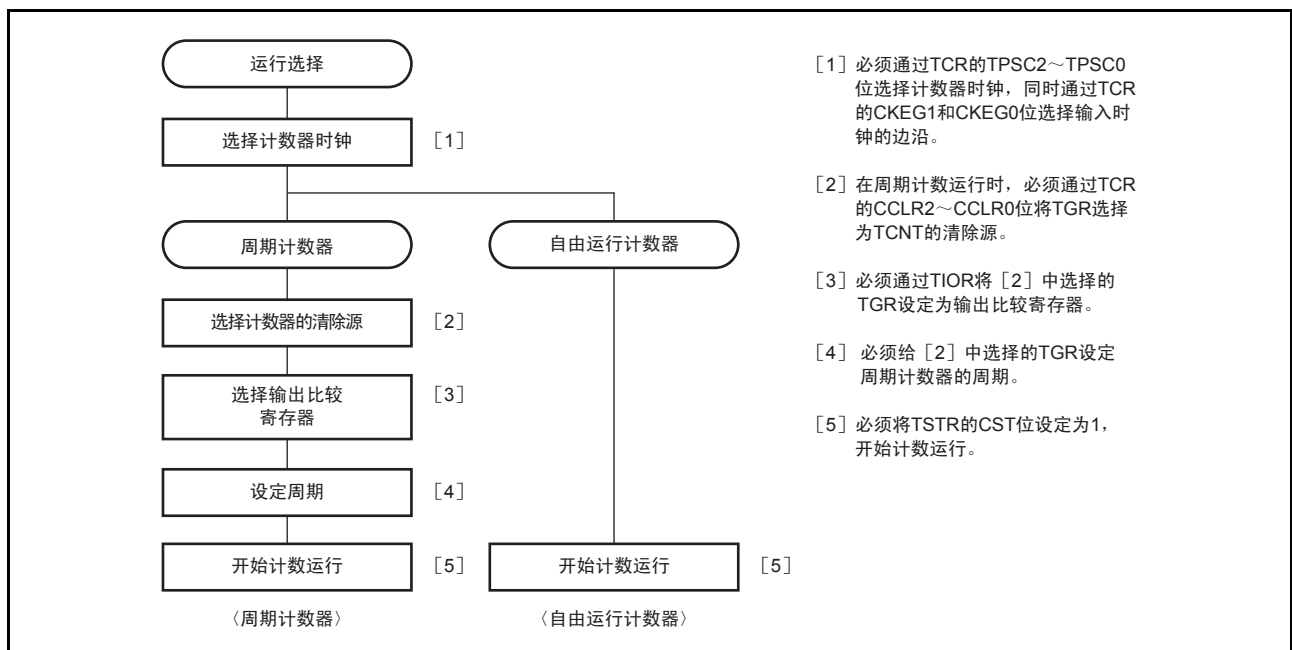


图 9.4 计数器运行的设定步骤例子

(b) 自由运行计数的运行和周期计数运行

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

自由运行计数器的运行如图 9.5 所示。

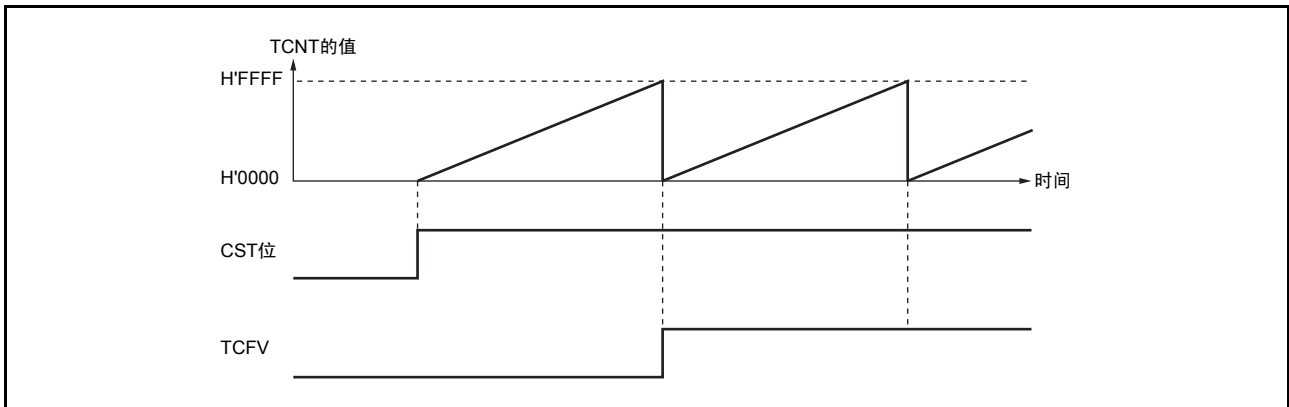


图 9.5 自由运行计数器的运行

当选择比较匹配为 TCNT 的清除源时, 对应通道的 TCNT 就进行周期计数运行。将用于周期设定的 TGR 设定为输出比较寄存器, 并通过 TCR 的 CCLR2 ~ CCLR0 位选择比较匹配的计数器清除。在设定后, 如果将 TSTR 的对应位设定为 1, 就作为周期计数器开始递增计数运行。当计数值和 TGR 的值一致时, TSR 的 TGF 位就被置 1, TCNT 被清 $H'0000$ 。

此时, 如果对应 TIER 的 TGIE 位是 1, MTU2 就请求中断。TCNT 在进行比较匹配后, 从 $H'0000$ 开始继续递增计数。

周期计数器的运行如图 9.6 所示。

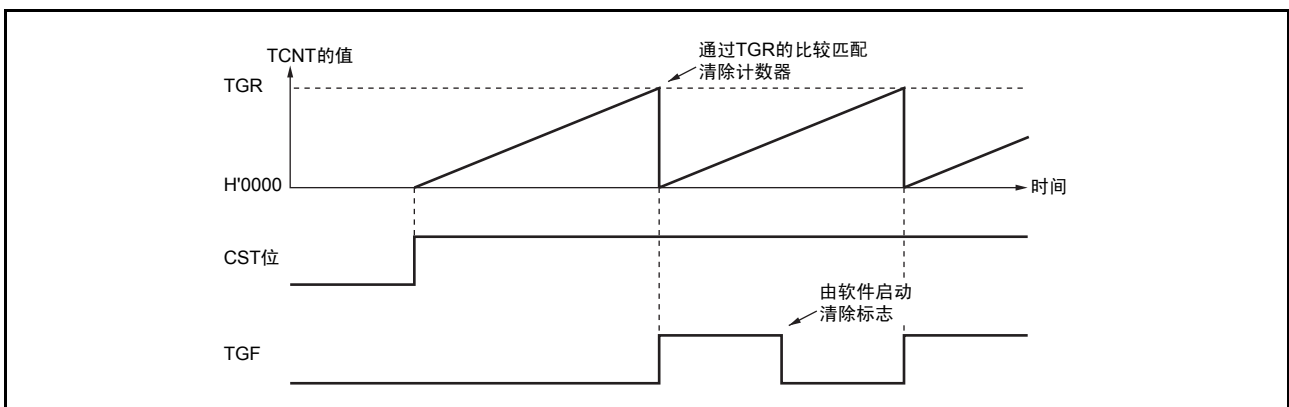


图 9.6 周期计数器的运行

(2) 比较匹配的波形输出功能

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

(a) 通过比较匹配进行波形输出运行的设定步骤例子

通过比较匹配进行波形输出运行的设定步骤例子如图 9.7 所示。

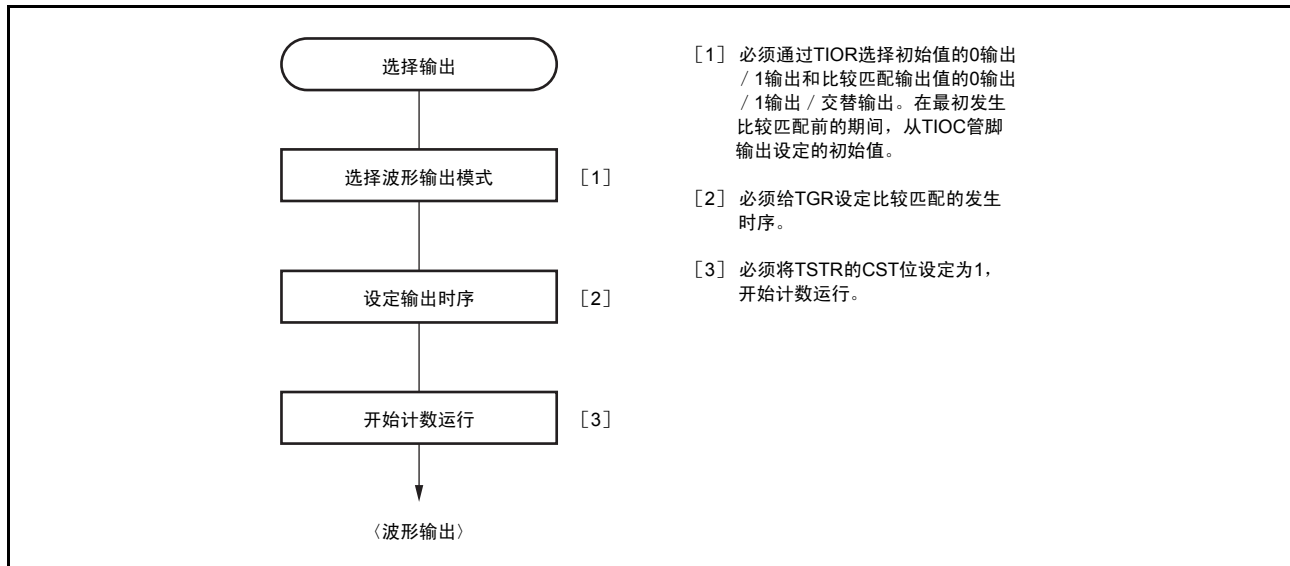


图 9.7 通过比较匹配进行波形输出的运行例子

(b) 波形输出运行的例子

0 输出 / 1 输出的例子如图 9.8 所示。

在此例中，将 TCNT 作为自由运行的计数运行，并设定为通过比较匹配 A 进行 1 输出、通过比较匹配 B 进行 0 输出。当设定的电平和管脚的电平一致时，管脚的电平不变。

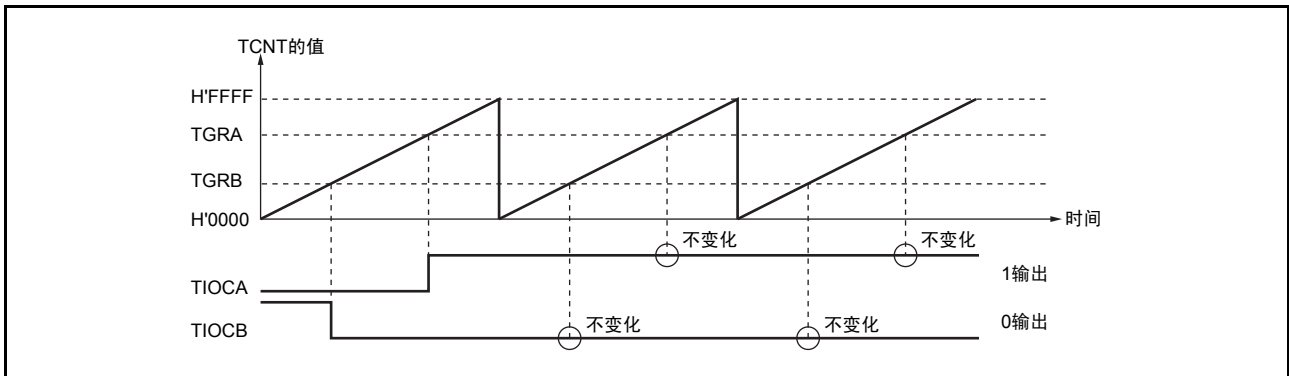


图 9.8 0 输出 / 1 输出的运行例

交替输出的例子如图 9.9 所示。

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

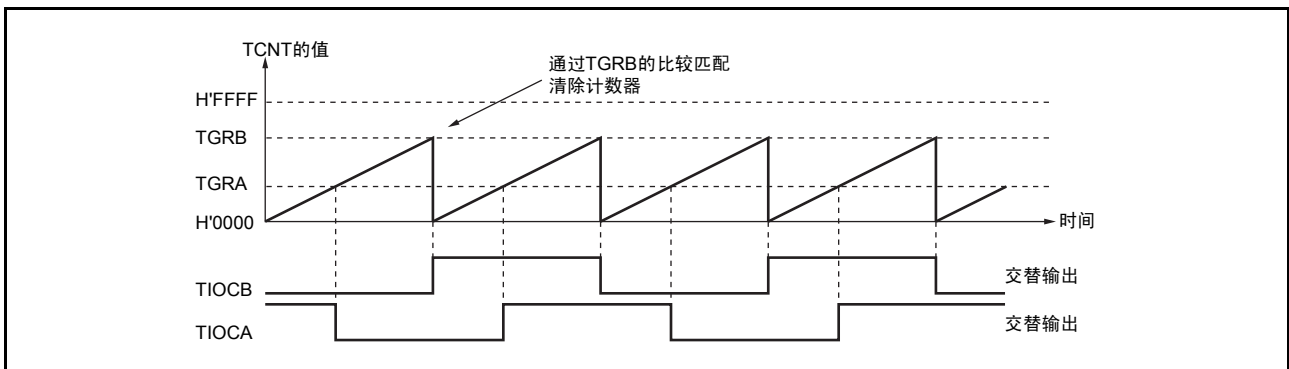


图 9.9 交替输出的运行例

(3) 输入捕捉功能

能检测 TIOC 管脚的输入边沿，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 $MP\phi / 1$ 作为输入捕捉的计数器输入时钟。如果选择 $MP\phi / 1$ ，就不发生输入捕捉。

(a) 输入捕捉运行的设定步骤例子

输入捕捉运行的设定步骤例子如图 9.10 所示。

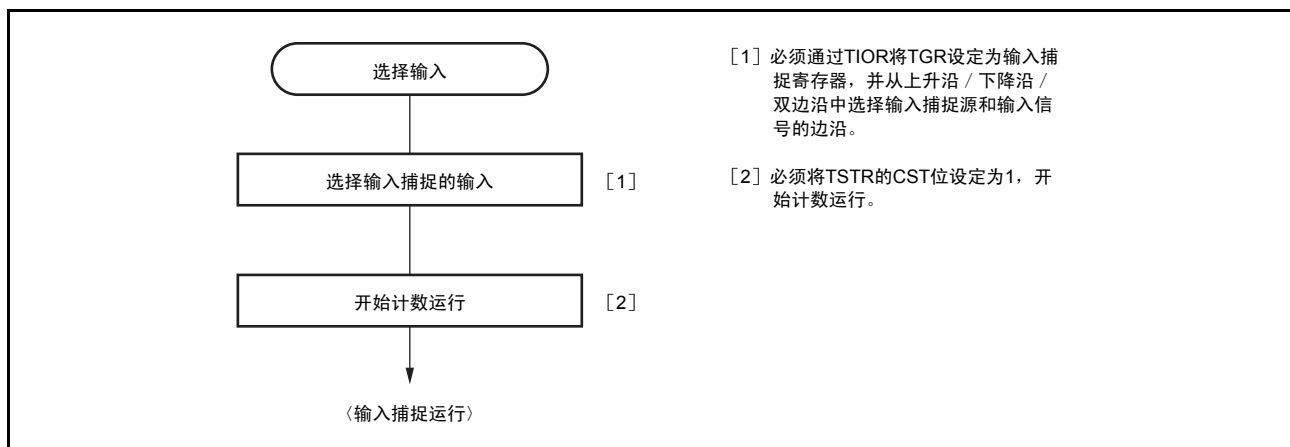


图 9.10 输入捕捉运行的设定例

(b) 输入捕捉运行的例子

输入捕捉的运行例子如图 9.11 所示。

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

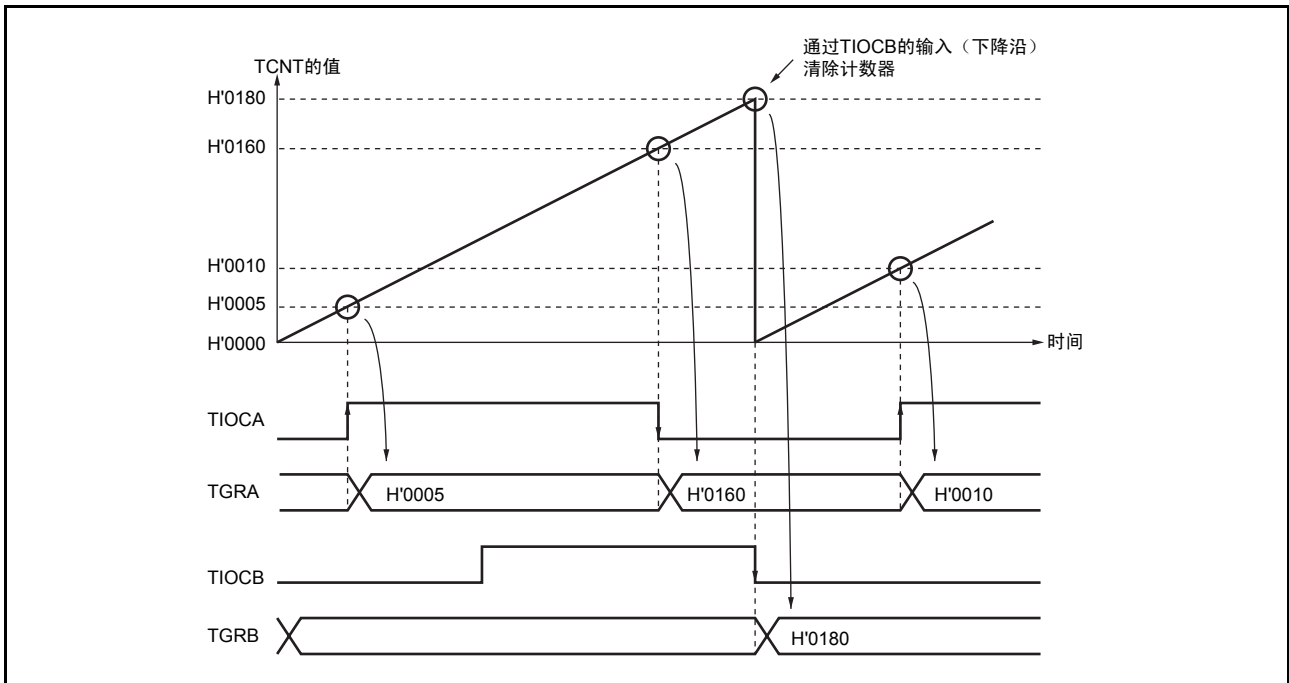


图 9.11 输入捕捉的运行例

9.4.2 同步运行

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

通过同步运行，能增加对于 1 个时基要运行的 TGR 个数。

通道 0 ~ 4 都能设定为同步运行。

通道 5 不能进行同步运行。

(1) 同步运行的设定步骤例子

同步运行的设定步骤例子如图 9.12 所示。

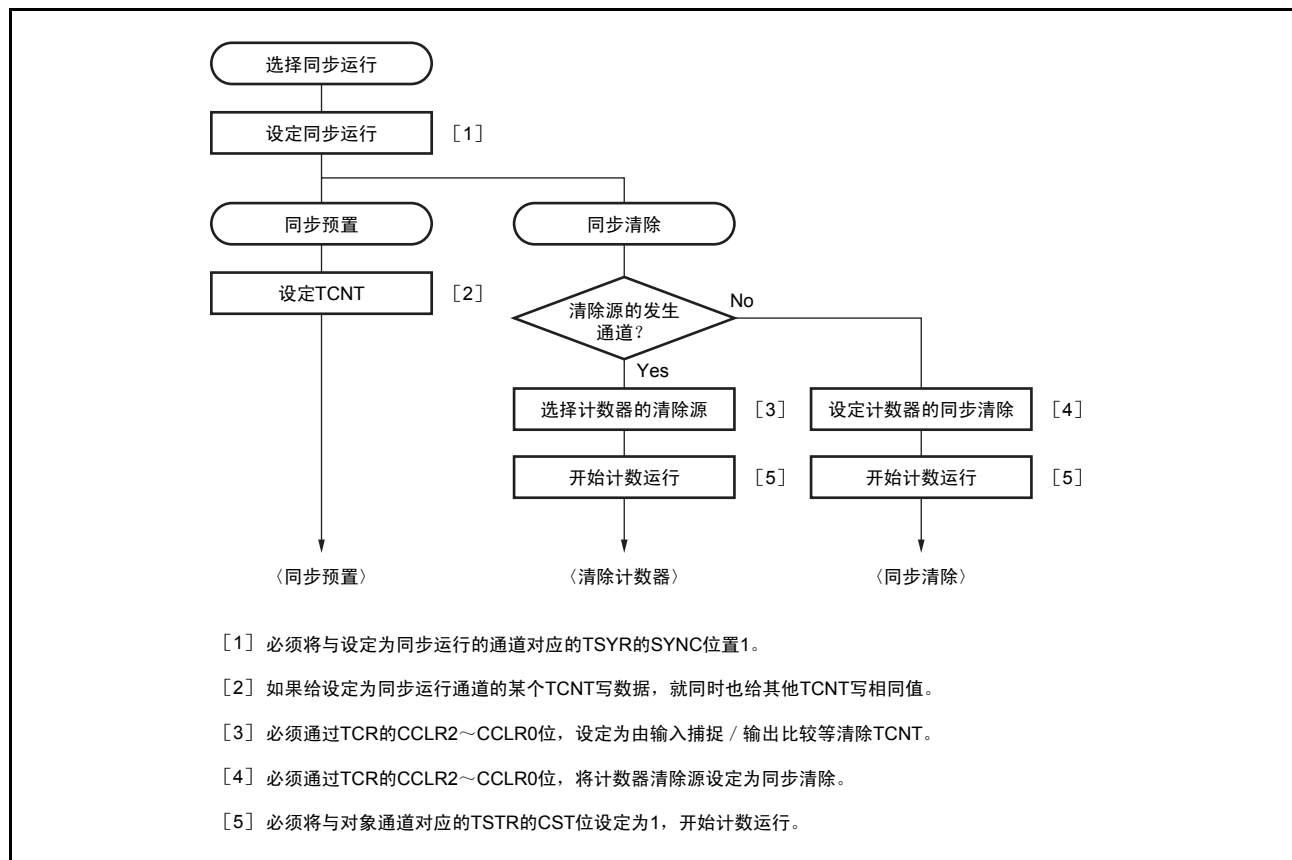


图 9.12 同步运行的设定步骤例子

(2) SH7125 的同步运行的例子

同步运行的例子如图 9.13 所示。

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

从 TIOC0A、TIOC1A、TIOC2A 管脚输出 3 相的 PWM 波形。此时，通道 0 ~ 2 的 TCNT 进行同步预置或者由 TGRB_0 比较匹配引起的同步清除，并且设定在 TGRB_0 的数据为 PWM 周期。

有关 PWM 模式请参照“9.4.5 PWM 模式”。

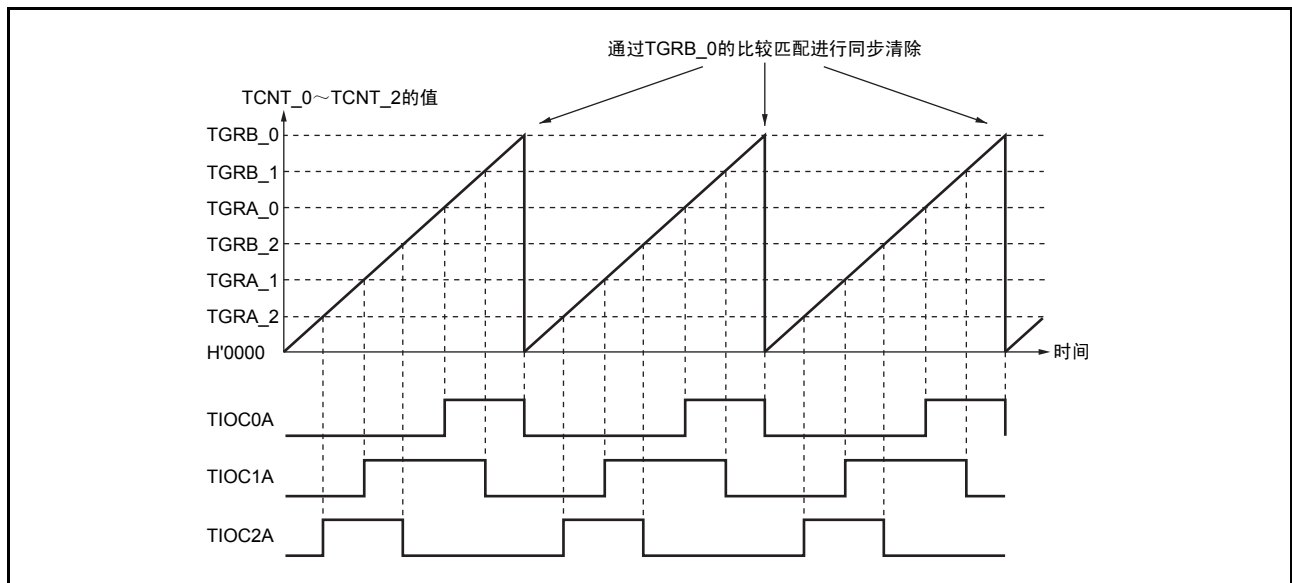


图 9.13 同步运行的例子

9.4.3 缓冲运行

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

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

【注】 TGRE_0 不能设定为输入捕捉寄存器，只作为比较匹配寄存器运行。

缓冲运行时的寄存器组合如表 9.43 所示

表 9.43 寄存器的组合

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

• TGR 为输出比较寄存器的情况

如果发生比较匹配，就将对应通道的缓冲寄存器的值传送到定时器的通用寄存器。

此运行如图 9.14 所示。

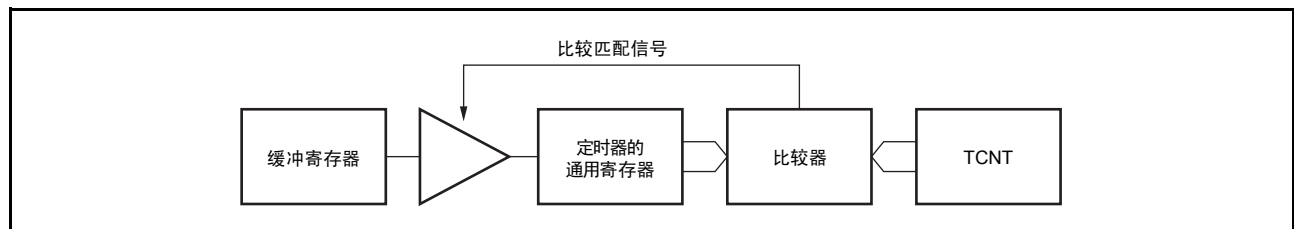


图 9.14 比较匹配的缓冲运行

• TGR 为输入捕捉寄存器的情况

如果发生输入捕捉，就在将 TCNT 的值传送到 TGR 的同时，将以前保存在 TGR 的值传送到缓冲寄存器。

此运行如图 9.15 所示。

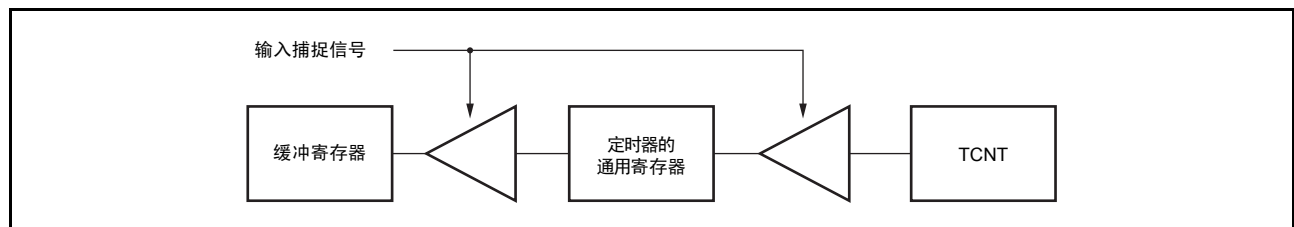


图 9.15 输入捕捉的缓冲运行

(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 9.16 所示。

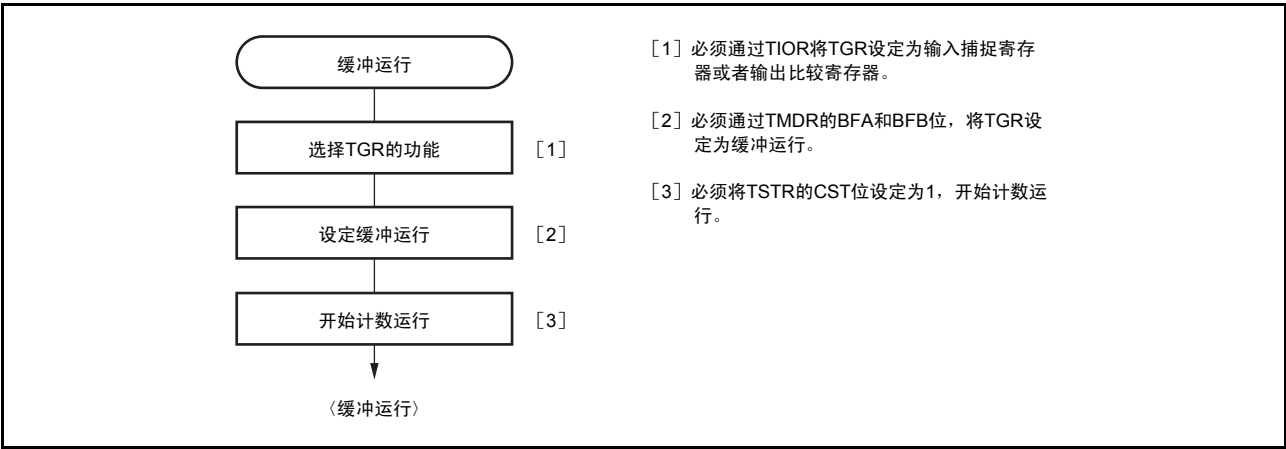


图 9.16 缓冲运行的设定步骤例子

(2) 缓冲运行的例子

(a) TGR 为输出比较寄存器的情况

设定通道 0 为 PWM 模式 1 以及 TGRA 和 TGRC 为缓冲运行时的运行例子如图 9.17 所示。在此例中，通过比较匹配 B 清除 TCNT、通过比较匹配 A 进行 1 输出以及通过比较匹配 B 进行 0 输出，并将 TBTM 的 TTSA 位设定为 0。

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

有关 PWM 模式请参照“9.4.5 PWM 模式”。

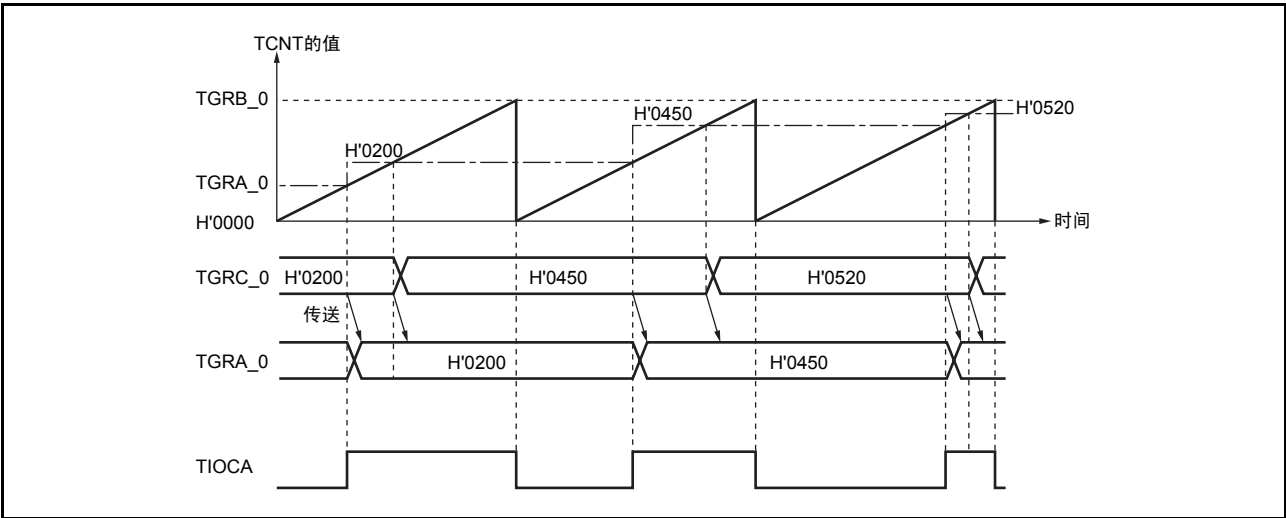


图 9.17 缓冲运行的例子 (1)

(b) TGR 为输入捕捉寄存器的情况

设定 TGRA 为输入捕捉寄存器以及 TGRA 和 TGRC 为缓冲运行时的运行例子如图 9.18 所示。

TCNT 通过 TGRA 的输入捕捉进行计数器清除，选择上升 / 下降的双边沿为 TIOCA 管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。

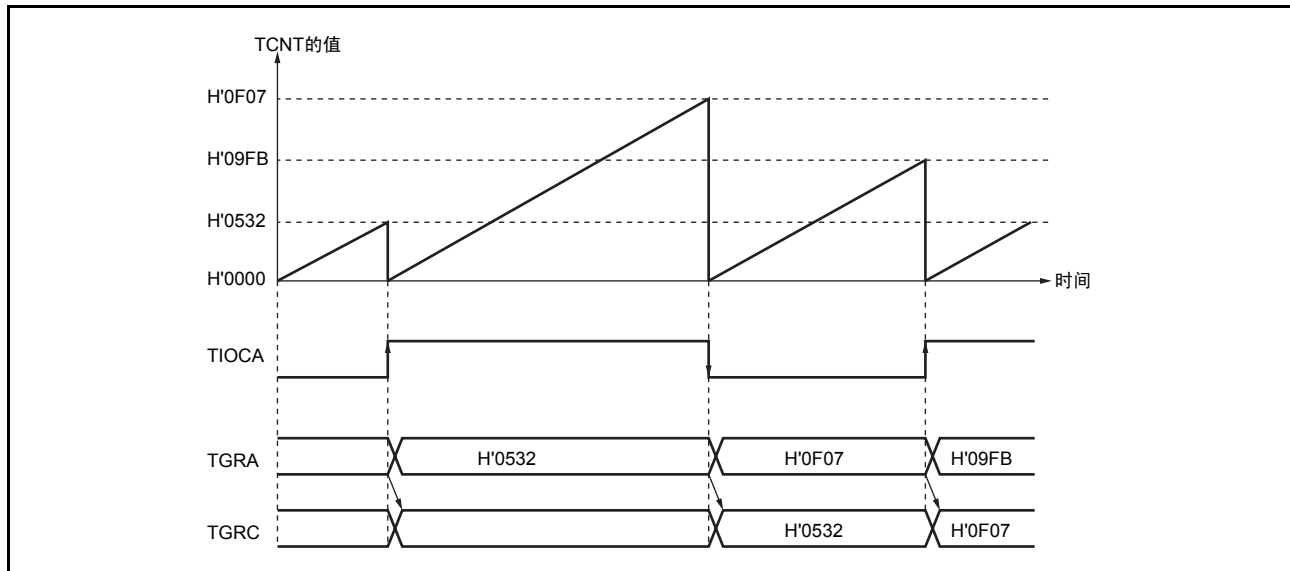


图 9.18 缓冲运行的例子 (2)

(3) 选择缓冲运行时的缓冲寄存器到定时器的通用寄存器的传送时序

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

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

【注】 必须在停止 TCNT 运行的状态下进行 TBTM 的设定。

设定通道 0 为 PWM 模式 1 以及 TGRA_0 和 TGRC_0 为缓冲运行时的运行例子如图 9.19 所示。在此例中，通过比较匹配 B 清除 TCNT_0、通过比较匹配 A 进行 1 输出以及通过比较匹配 B 进行 0 输出，并将 TBTM_0 的 TTSA 位设定为 1。

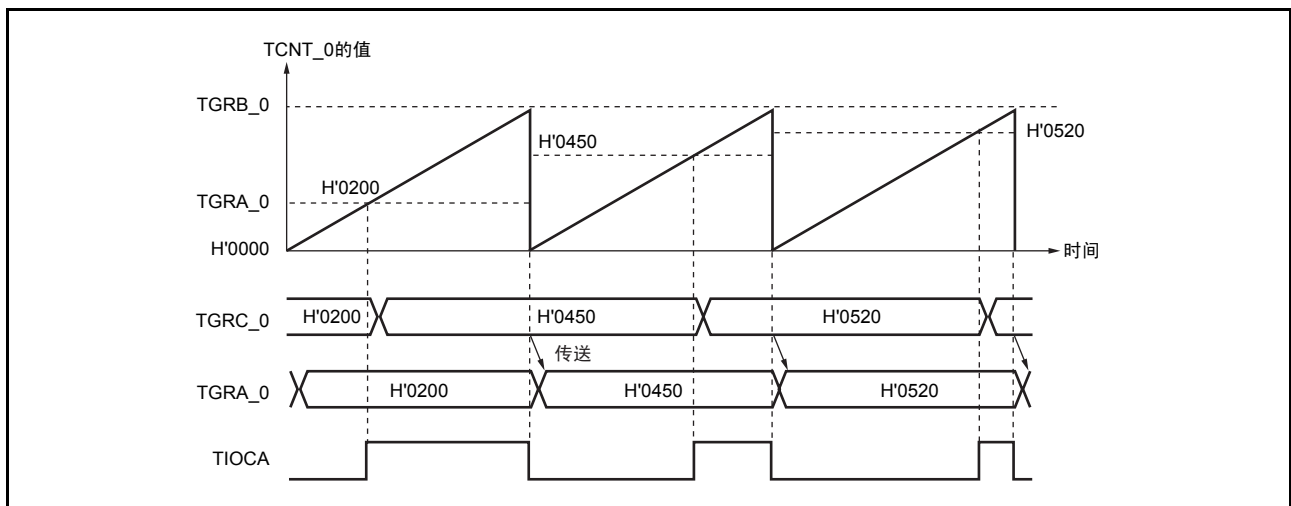


图 9.19 将 TGRC_0 到 TGRA_0 的缓冲传送时序选择为清除 TCNT_0 时的运行例

9.4.4 级联运行

级联运行是将2个通道的16位计数器连接为32位计数器的功能。

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

只在低16位的TCNT为相位计数模式时发生下溢。

级联的组合如表9.44所示。

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

表 9.44 级联的组合

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

如果在级联运行时进行TCNT_1和TCNT_2的同时输入捕捉，就能通过输入捕捉控制寄存器（TICCR）的设定，将输入管脚追加到输入捕捉条件。有关级联时的输入捕捉，请参照“9.7.22 级联中TCNT_1和TCNT_2的同时输入捕捉”。

TICCR设定值和输入捕捉的输入管脚的对应如表9.45所示。

表 9.45 TICCR 设定值和输入捕捉的输入管脚的对应

对象输入捕捉	TICCR 设定值	输入捕捉的输入管脚
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9.20所示。

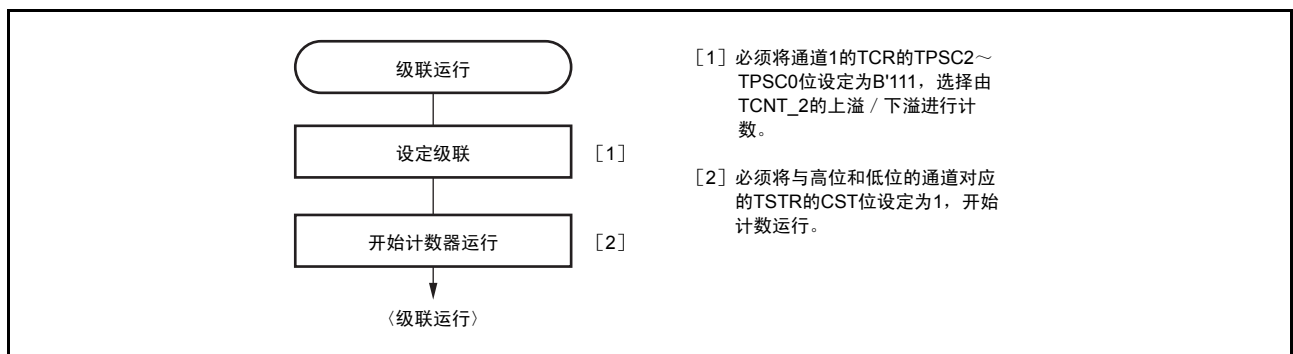


图 9.20 级联运行的设定步骤

(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 9.21 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。

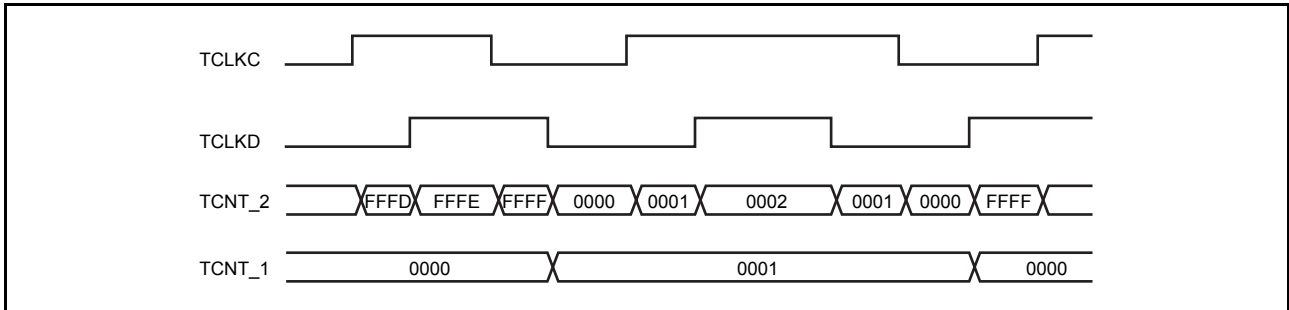


图 9.21 级联的运行例 (a)

(3) SH7125 的级联运行的例子 (b)

将 TCNT_1 和 TCNT_2 进行级联并给 TICCRR 的 I2AE 位设定 1 (将 TIOC2A 管脚追加到 TGRA_1 的输入捕捉条件) 时的运行如图 9.22 所示。在此例中, 将 TIOR_1 的 IOA0 ~ IOA3 设定为在 TIOC1A 的上升沿进行输入捕捉, 将 TIOR_2 的 IOA0 ~ IOA3 设定为在 TIOC2A 的上升沿进行输入捕捉。

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

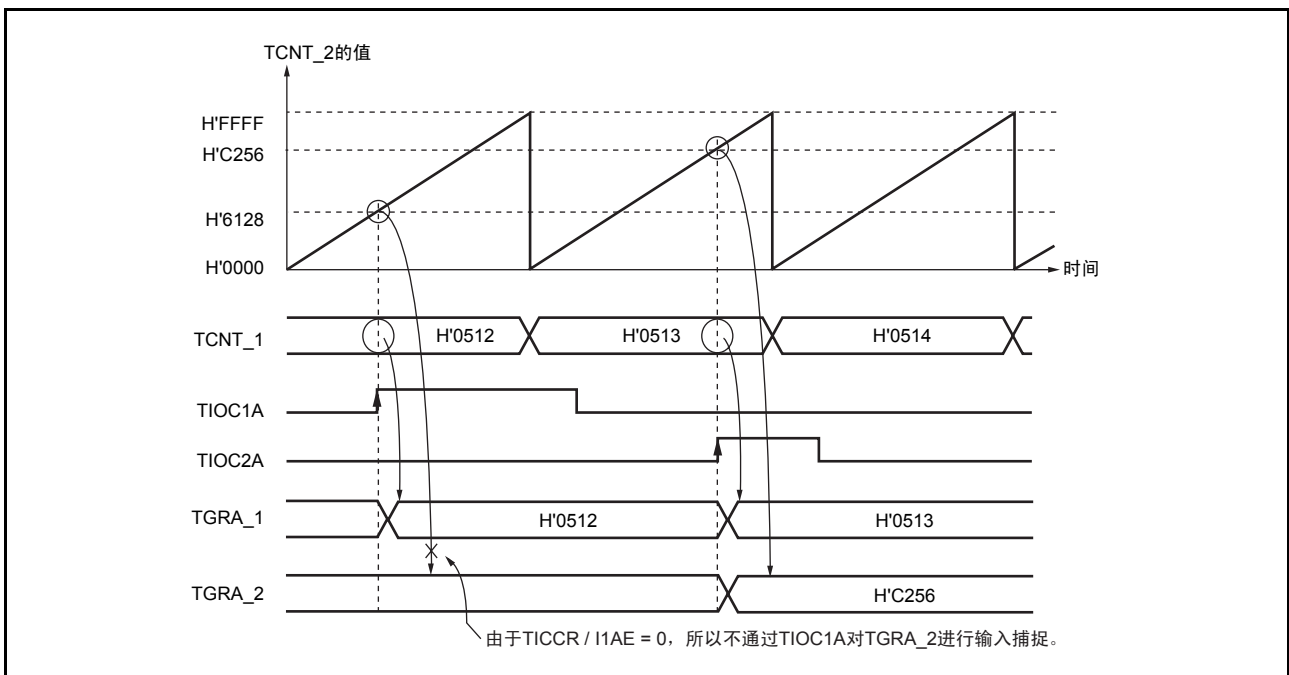


图 9.22 级联的运行例 (b)

(4) SH7125 的级联运行的例子 (c)

将 TCNT_1 和 TCNT_2 进行级联并给 TICC1 的 I2AE 位和 I1AE 位设定 1 (将 TIOC2A 管脚追加到 TGRA_1 的输入捕捉条件并将 TIOC1A 管脚追加到 TGRA_2 的输入捕捉条件) 时的运行如图 9.23 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。

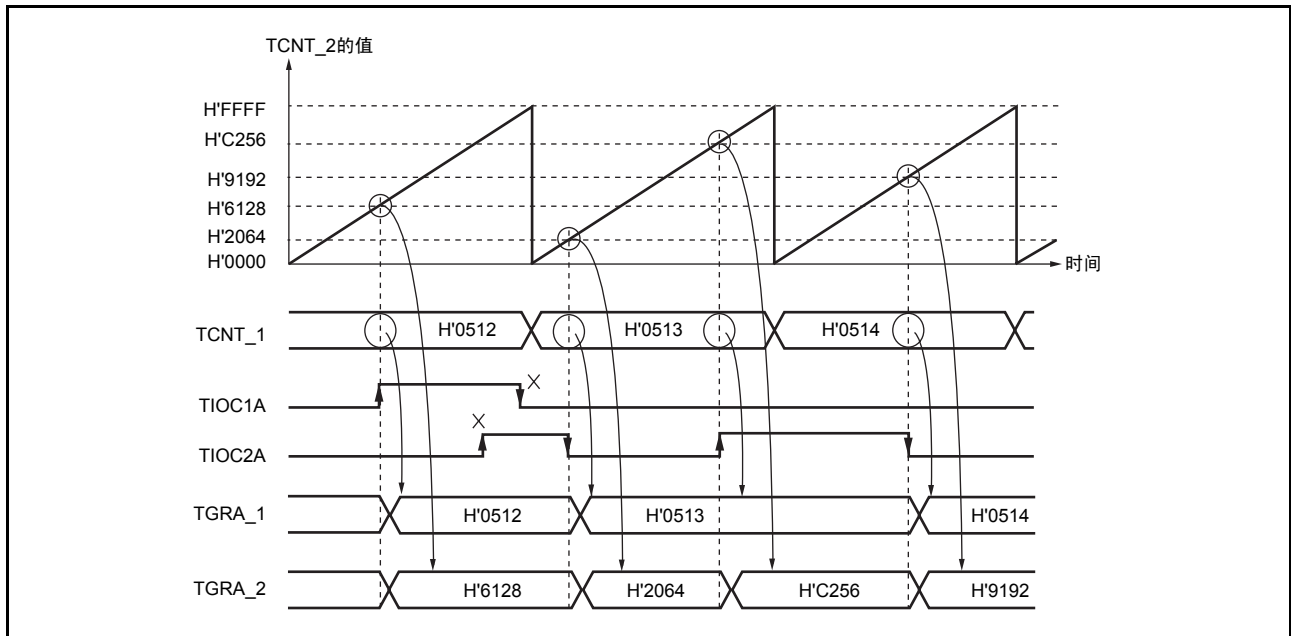


图 9.23 级联的运行例 (c)

(5) SH7125 的级联运行的例子 (d)

将 TCNT_1 和 TCNT_2 进行级联并给 TICCRR 的 I2AE 位设定 1 (将 TIOC2A 管脚追加到 TGRA_1 的输入捕捉条件) 时的运行如图 9.24 所示。在此例中, 将 TIOR_1 的 IOA0 ~ IOA3 设定为通过发生 TGRA_0 的比较匹配进行输入捕捉, 将 TIOR_2 的 IOA0 ~ IOA3 设定为在 TIOC2A 的上升沿进行输入捕捉。

此时, 因为 TIOR_1 设定为通过发生 TGRA_0 的比较匹配 / 输入捕捉进行输入捕捉, 所以即使给 TICCRR 的 I2AE 位设定 1, TIOC2A 的边沿也不为 TGRA_1 的输入捕捉条件。

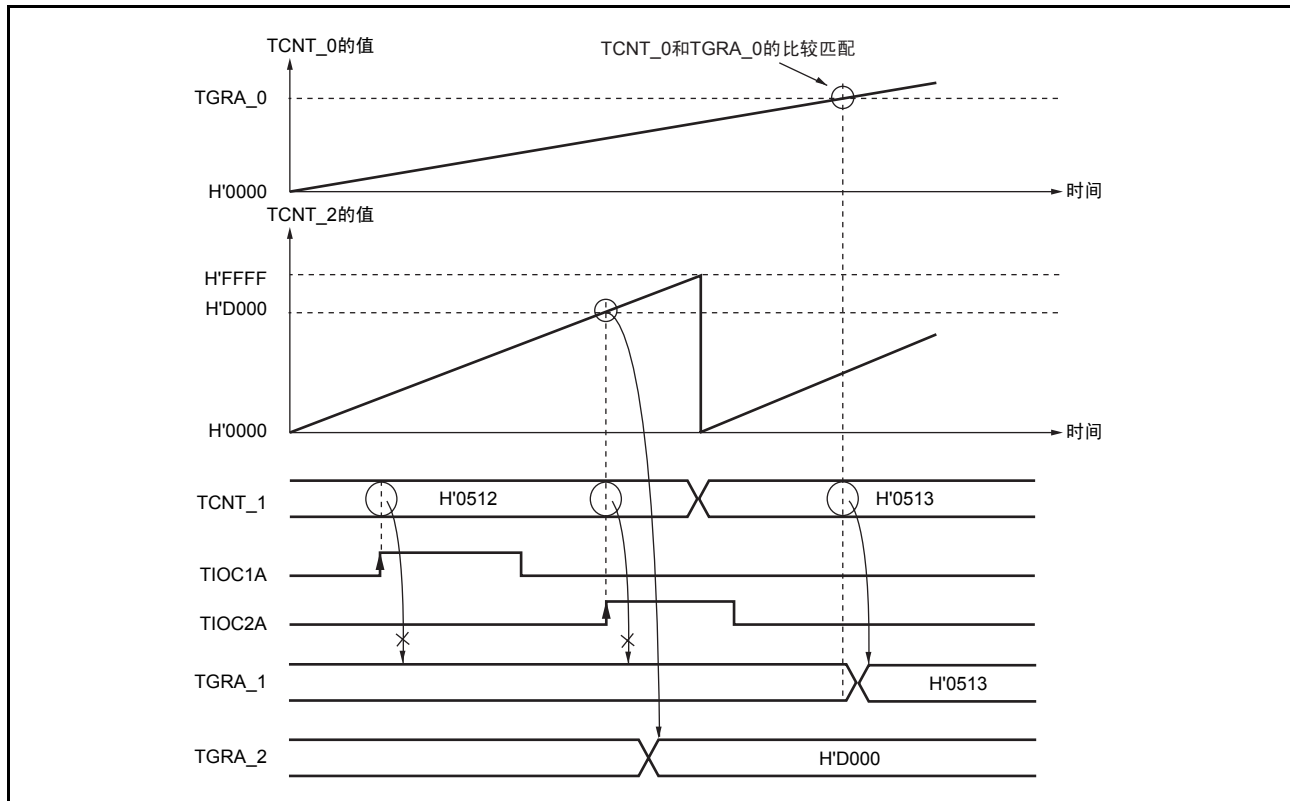


图 9.24 级联的运行例 (d)

9.4.5 PWM 模式

PWM 模式是从输出管脚输出各种 PWM 波形的模式。可从 0 输出 / 1 输出 / 交替输出中选择各 TGR 比较匹配的输出现平。

能通过设定各 TGR，输出占空比为 0 ~ 100% 的 PWM 波形。

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

PWM 模式有以下 2 种：

1. PWM 模式 1

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

在 PWM 模式 1 中能进行最多 8 相的 PWM 输出。

2. 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，各管脚的输出值为 TIOR 设定的初始值。如果周期寄存器和占空比寄存器的设定值相同，即使发生比较匹配，输出值也不变。

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

PWM 输出管脚和寄存器的对应如表 9.46 所示。

表 9.46 各 PWM 输出的寄存器和输出管脚

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

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

* 只限 SH7125 的管脚。

(1) PWM 模式的设定步骤例子

PWM 模式的设定步骤例子如图 9.25 所示。

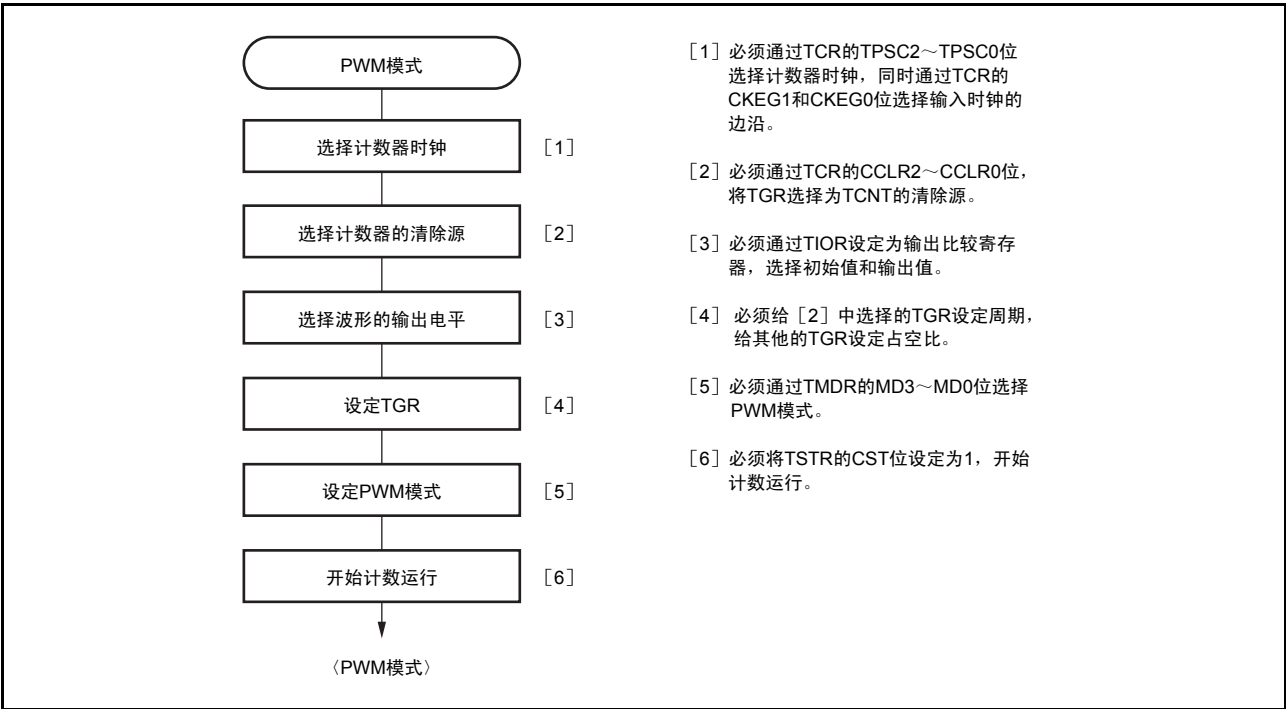


图 9.25 PWM 模式的设定步骤例子

(2) PWM 模式的运行例

PWM1 模式的运行例子如图 9.26 所示。

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

此时，设定在 TGRA 的值为周期，设定在 TGRB 的值为占空比。

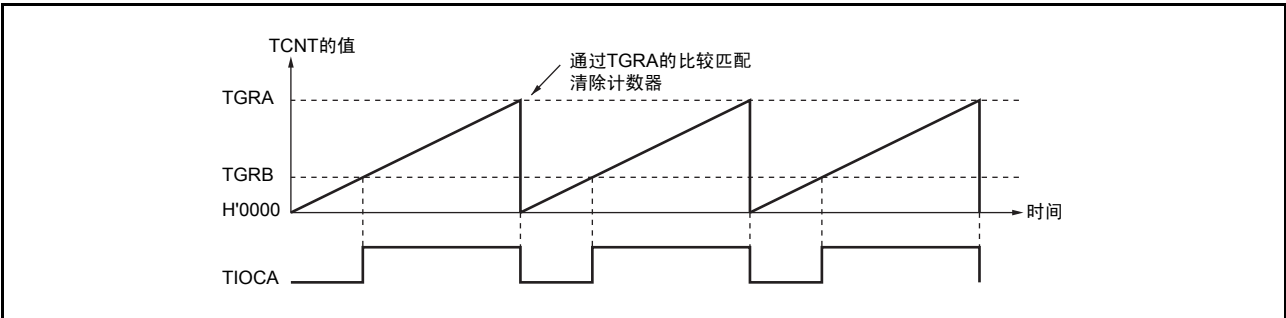


图 9.26 PWM 模式的运行例

SH7125 的 PWM 模式 2 的运行例子如图 9.27 所示。

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

此时，设定在 TGR1B 的值为周期，设定在其他 TGR 的值为占空比。

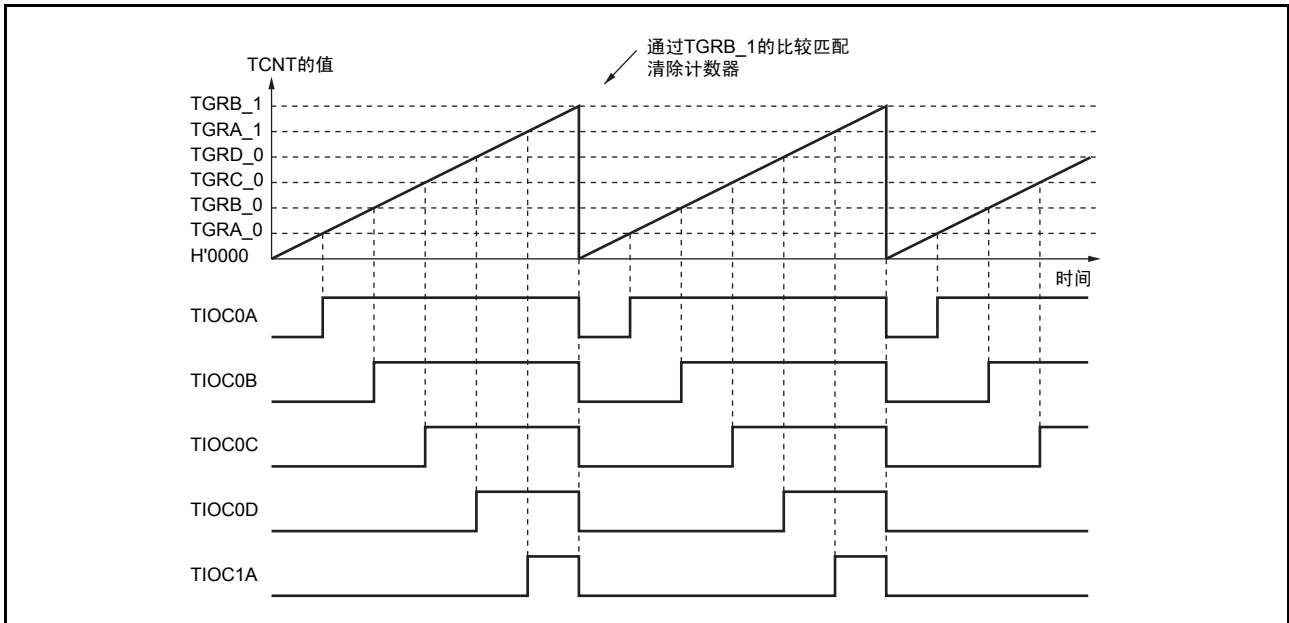


图 9.27 PWM 模式的运行例

在 PWM 模式中，输出占空比为 0% 和占空比为 100% 的 PWM 波形的例子如图 9.28 所示。

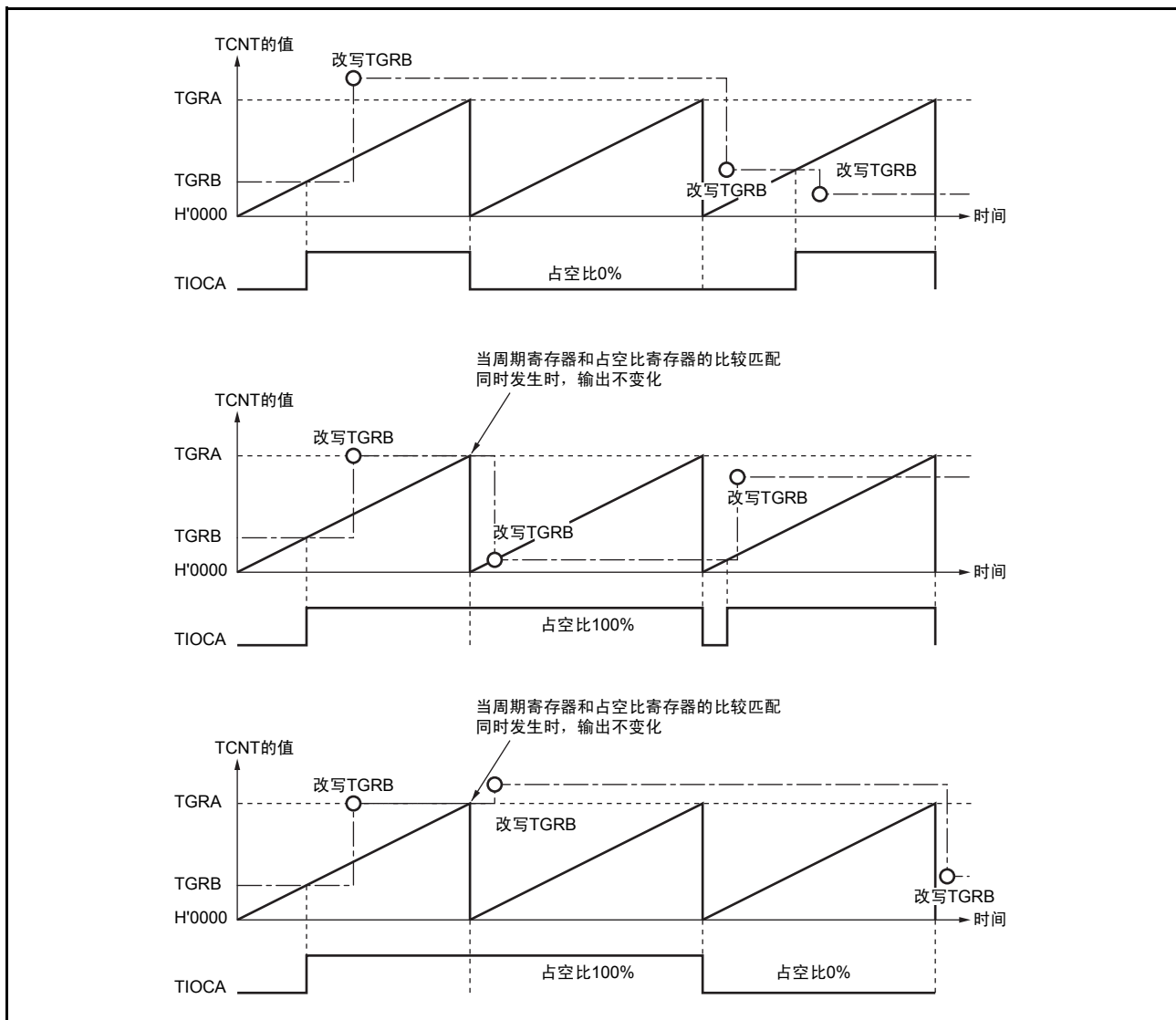


图 9.28 PWM 模式的运行例

9.4.6 相位计数模式

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

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

能用作 2 相编码器脉冲的输入。

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

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

外部时钟管脚和通道的对应如表 9.47 所示。

表 9.47 相位计数模式时钟的输入管脚

通道	外部时钟管脚	
	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 9.29 所示。

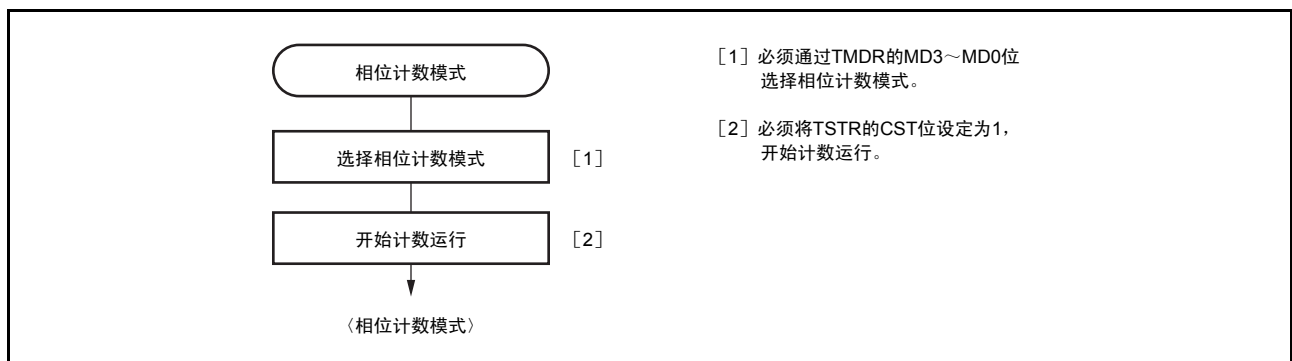


图 9.29 相位计数模式的设定步骤例子

(2) 相位计数模式的运行例

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

(a) 相位计数模式 1

相位计数模式 1 的运行例子如图 9.30 所示，TCNT 递增计数 / 递减计数的条件如表 9.48 所示。

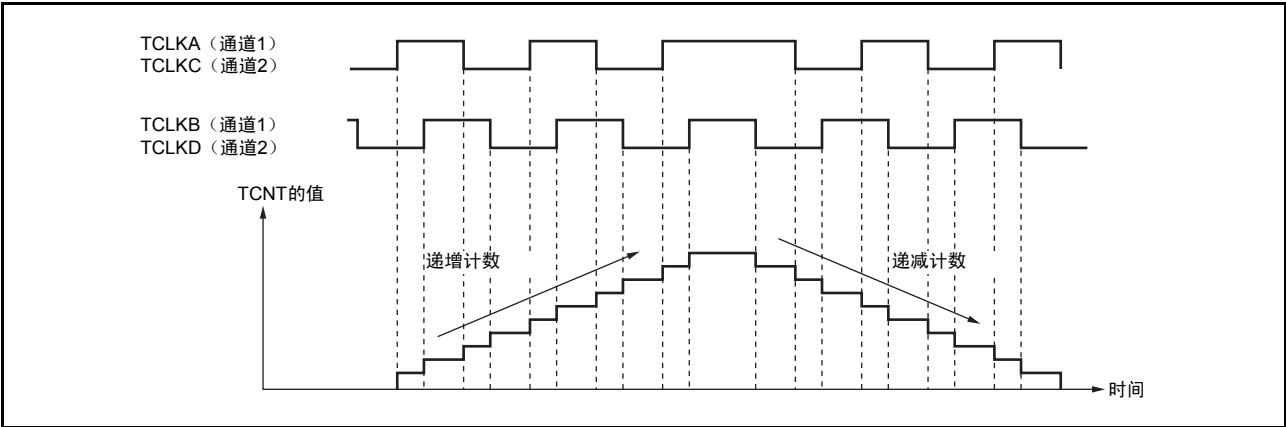


图 9.30 相位计数模式 1 的运行例

表 9.48 相位计数模式 1 的递增计数 / 递减计数的条件

TCLKA (通道 1) TCLKC (通道 2)	TCLKB (通道 1) TCLKD (通道 2)	运行内容
High 电平		递增计数
Low 电平		
	Low 电平	
	High 电平	
High 电平		递减计数
Low 电平		
	High 电平	
	Low 电平	

【符号说明】

上升沿

下降沿

(b) 相位计数模式 2

相位计数模式 2 的运行例子如图 9.31 所示，TCNT 递增计数 / 递减计数的条件如表 9.49 所示。

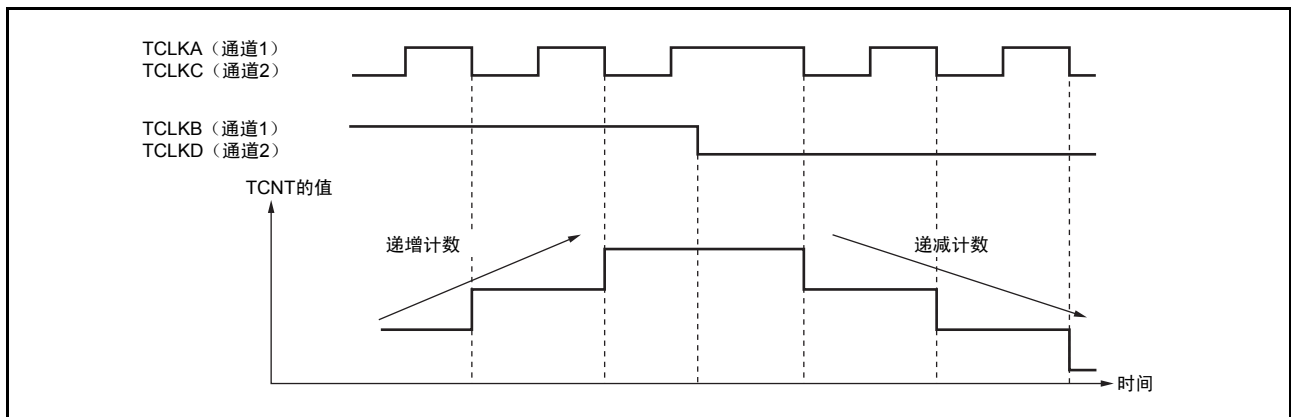


图 9.31 相位计数模式 2 的运行例

表 9.49 相位计数模式 2 的递增计数 / 递减计数的条件

TCLKA (通道 1) TCLKC (通道 2)	TCLKB (通道 1) TCLKD (通道 2)	运行内容
High 电平		不计数 (Don't care)
Low 电平		不计数 (Don't care)
	Low 电平	不计数 (Don't care)
	High 电平	递增计数
High 电平		不计数 (Don't care)
Low 电平		不计数 (Don't care)
	High 电平	不计数 (Don't care)
	Low 电平	递减计数

【符号说明】

上升沿

下降沿

(c) 相位计数模式 3

相位计数模式 3 的运行例子如图 9.32 所示，TCNT 递增计数 / 递减计数的条件如表 9.50 所示。

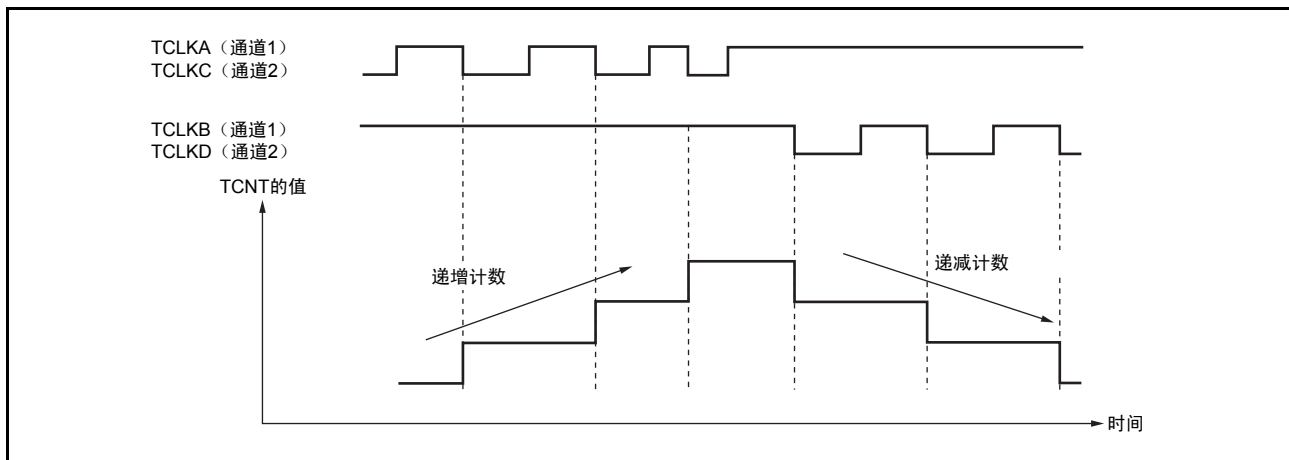


图 9.32 相位计数模式 3 的运行例

表 9.50 相位计数模式 3 的递增计数 / 递减计数的条件

TCLKA (通道 1) TCLKC (通道 2)	TCLKB (通道 1) TCLKD (通道 2)	运行内容
High 电平		不计数 (Don't care)
Low 电平		不计数 (Don't care)
	Low 电平	不计数 (Don't care)
	High 电平	递增计数
High 电平		递减计数
Low 电平		不计数 (Don't care)
	High 电平	不计数 (Don't care)
	Low 电平	不计数 (Don't care)

【符号说明】

上升沿

下降沿

(d) 相位计数模式 4

相位计数模式 4 的运行例子如图 9.33 所示，TCNT 递增计数 / 递减计数的条件如表 9.51 所示。

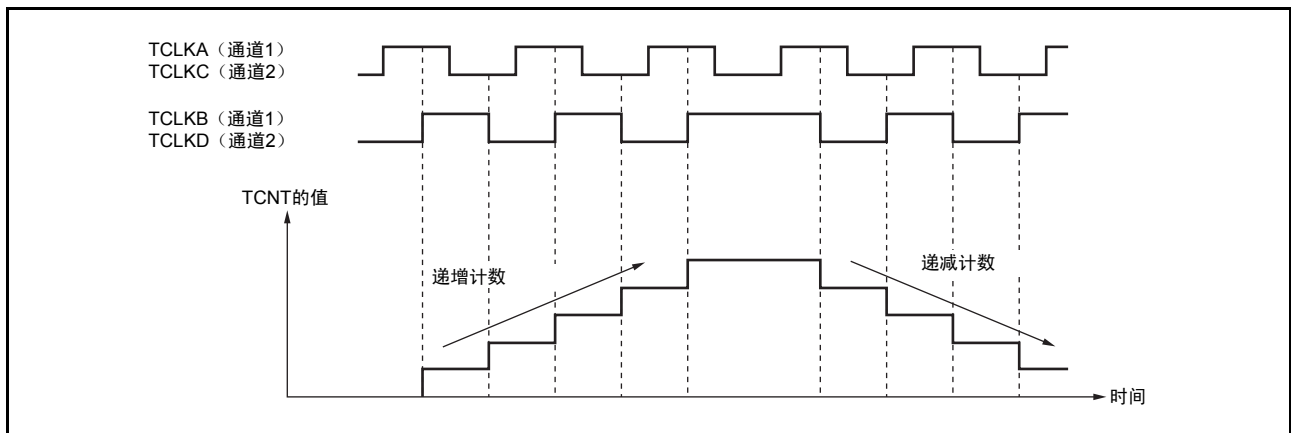


图 9.33 相位计数模式 4 的运行例

表 9.51 相位计数模式 4 的递增计数 / 递减计数的条件

TCLKA (通道 1) TCLKC (通道 2)	TCLKB (通道 1) TCLKD (通道 2)	运行内容
High 电平		递增计数
Low 电平		
	Low 电平	不计数 (Don't care)
	High 电平	
High 电平		递减计数
Low 电平		
	High 电平	不计数 (Don't care)
	Low 电平	

【符号说明】

上升沿

下降沿

(3) 相位计数模式的应用例

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 9.34 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_0 计数器，TGRA_0 和 TGRC_0 用于比较匹配功能，并设定速度控制周期和位置控制周期。TGRB_0 用于输入捕捉功能，并使 TGRB_0 和 TGRD_0 进行缓冲运行。将 TGRB_0 的输入捕捉源作为通道 1 的计数器输入时钟，检测 2 相译码器的 4 倍增脉冲的脉宽。

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

用此方法能检测正确的位置 / 速度。

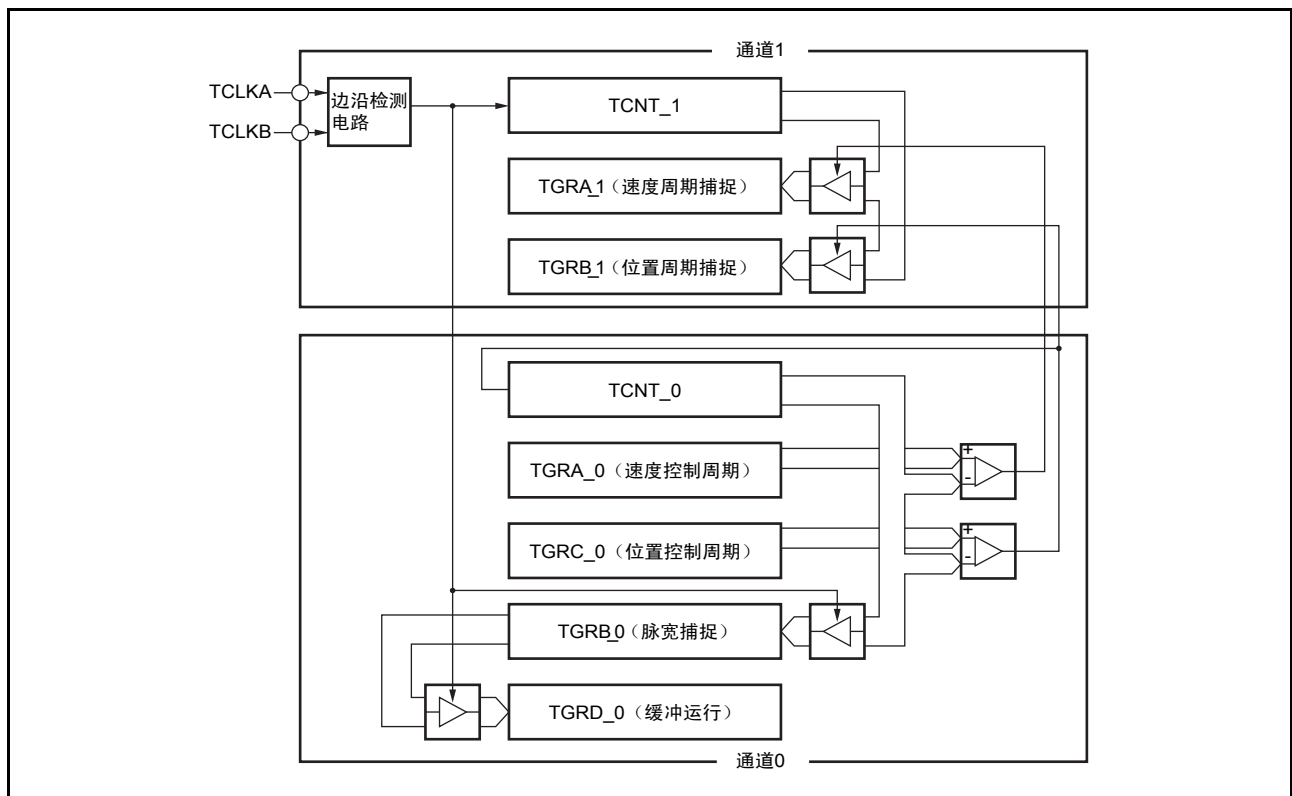


图 9.34 相位计数模式的应用例

9.4.7 复位同步 PWM 模式

在复位同步 PWM 模式中，通过组合通道 3、4，输出持有相互关系的三相 PWM 波形（正相和反相）。

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

使用的 PWM 输出管脚和寄存器设定分别如表 9.52 和表 9.53 所示。

表 9.52 复位同步 PWM 模式中的输出管脚

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

表 9.53 复位同步 PWM 模式中的寄存器设定

寄存器	设定内容
TCNT_3	初始设定 H'0000
TCNT_4	初始设定 H'0000
TGRA_3	设定 TCNT_3 的计数周期
TGRB_3	设定从 TIOC3B、TIOC3D 管脚输出的 PWM 波形的变化点
TGRA_4	设定从 TIOC4A、TIOC4C 管脚输出的 PWM 波形的变化点
TGRB_4	设定从 TIOC4B、TIOC4D 管脚输出的 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 9.35 所示。

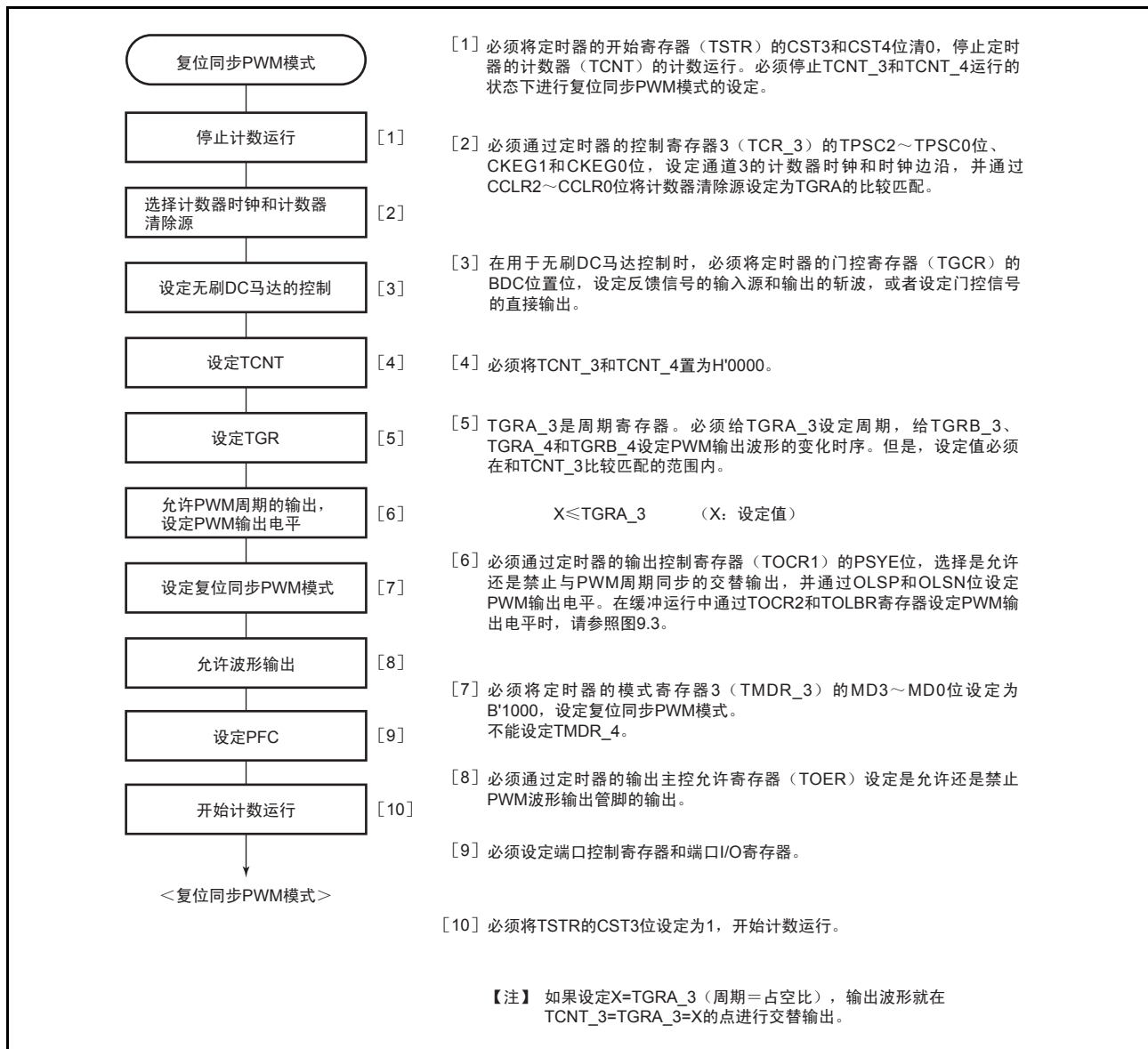


图 9.35 复位同步 PWM 模式的设定步骤例子

(2) 复位同步 PWM 模式的运行例

复位同步 PWM 模式的运行例子如图 9.36 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 输出管脚就进行交替输出。

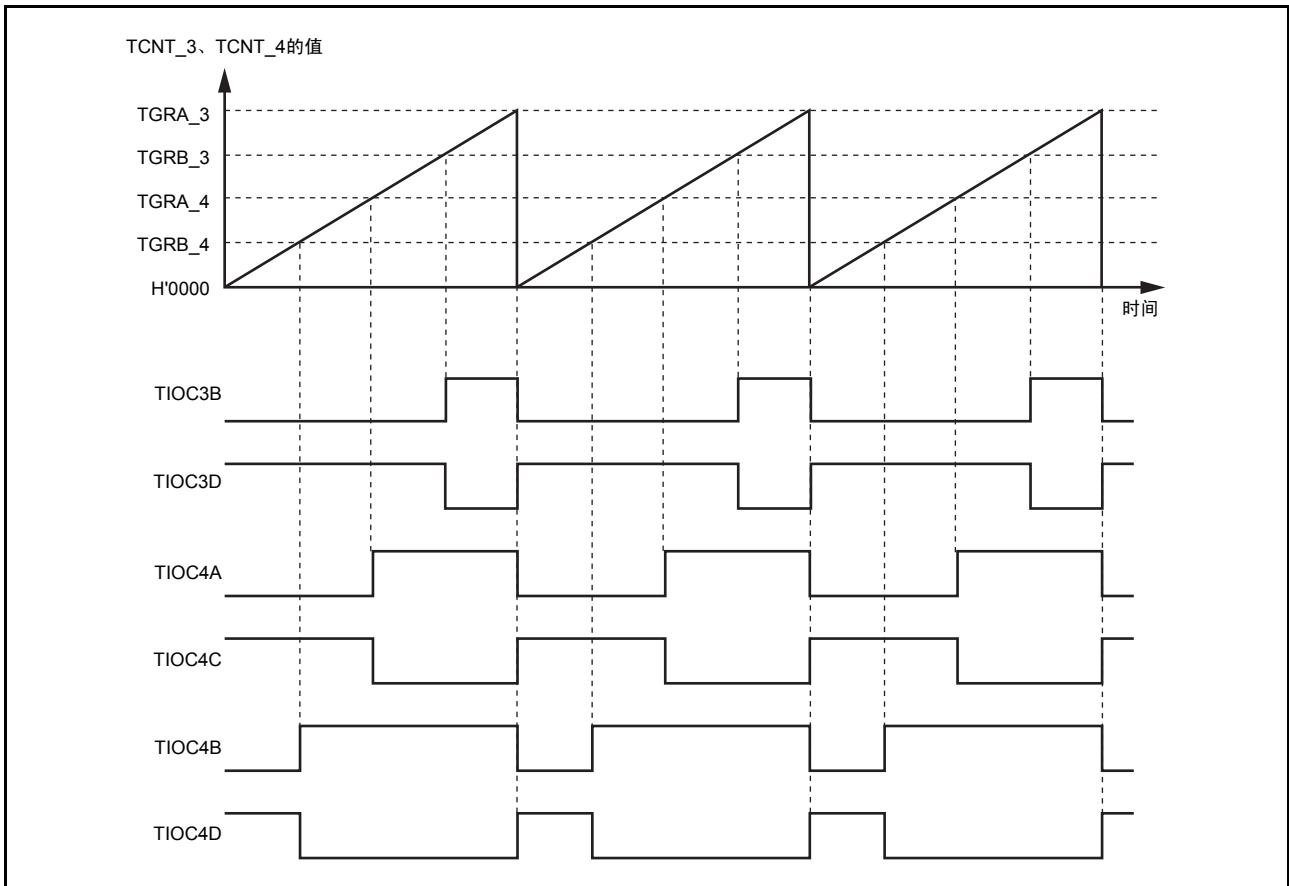


图 9.36 复位同步 PWM 模式的运行例（当设定 TOCR 的 OLSN=1、OLSP=1 时）

9.4.8 互补 PWM 模式

在互补 PWM 模式中，通过组合通道 3、4 将正相和反相为非重叠关系的 PWM 波形进行 3 相输出，也能设定没有非重叠时间。

当设定为互补 PWM 模式时，TIOC3B、TIOC3D、TIOC4A、TIOC4B、TIOC4C、TIOC4D 管脚为 PWM 输出管脚，能将 TIOC3A 管脚设定为与 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 输出管脚和寄存器设定分别如表 9.54 和表 9.55 所示。

作为端口功能，支持通过外部信号将 PWM 输出进行直接 OFF 的功能。

表 9.54 互补 PWM 模式中的输出管脚

通道	输出管脚	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 输出管脚 1
	TIOC3C	输入 / 输出端口 *
	TIOC3D	PWM 输出管脚 1'（和 PWM 输出 1 非重叠关系的反相波形。也能设定没有非重叠时间）
4	TIOC4A	PWM 输出管脚 2
	TIOC4C	PWM 输出管脚 2'（和 PWM 输出 2 非重叠关系的反相波形。也能设定没有非重叠时间）
	TIOC4B	PWM 输出管脚 3
	TIOC4D	PWM 输出管脚 3'（和 PWM 输出 3 非重叠关系的反相波形。也能设定没有非重叠时间）

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

表 9.55 互补 PWM 模式中的寄存器设定

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

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

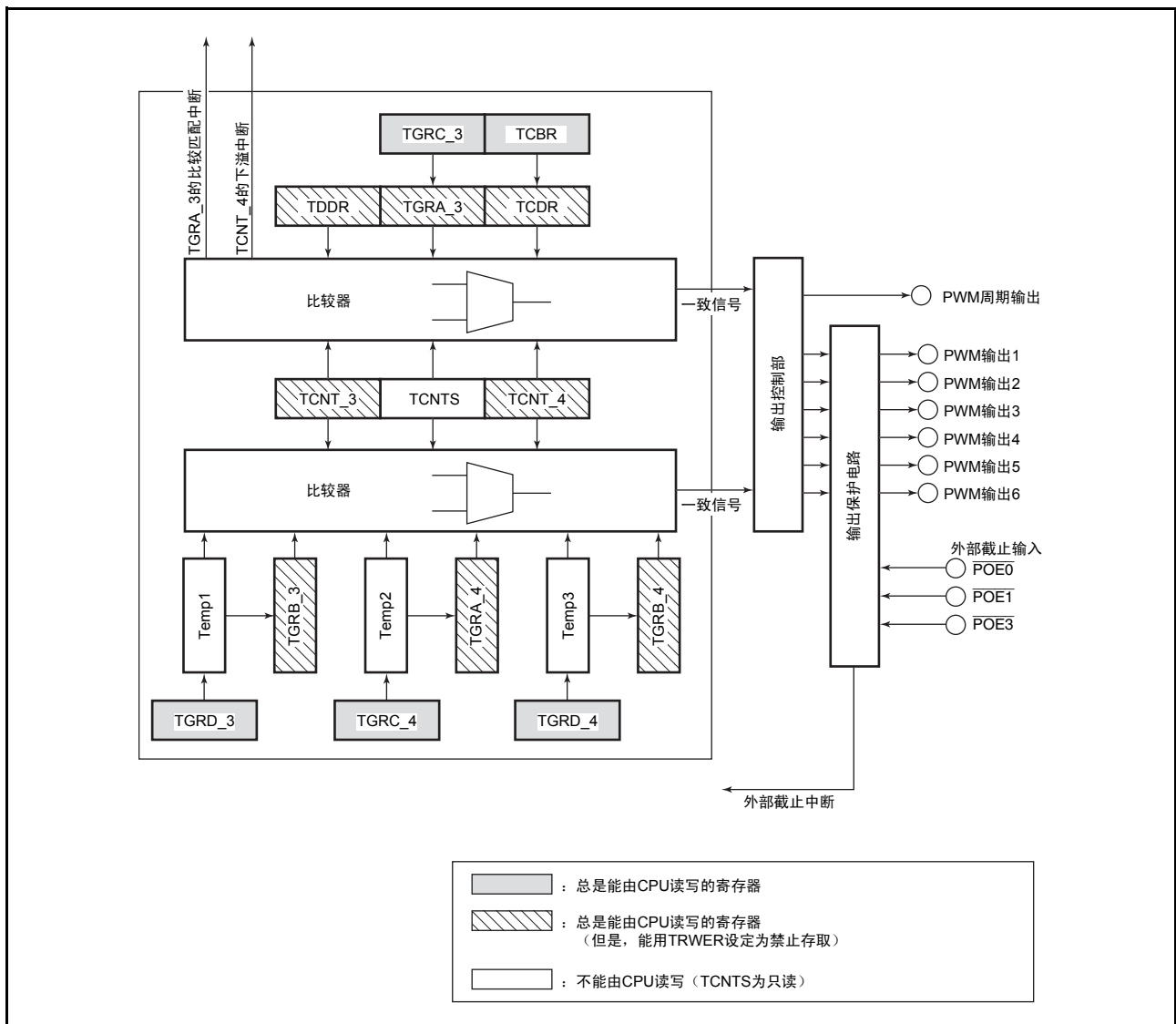


图 9.37 互补 PWM 模式中的通道 3、4 框图

(1) 互补 PWM 模式的设定步骤例子

互补 PWM 模式的设定步骤例子如图 9.38 所示。



图 9.38 互补 PWM 模式的设定步骤例子

(2) 互补 PWM 模式的运行概要

在互补 PWM 模式中能进行 6 相的 PWM 输出。互补 PWM 模式的计数器运行和运行例分别如图 9.39 和图 9.40 所示。

(a) 计数器运行

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

当 TCNT_3 被设定为互补 PWM 模式并且 TSTR 的 CST 位是 0 时，设定在 TDDR 的值就自动置为初始值。

当 CST 位被置 1 时，TCNT_3 就进行递增计数（计数到 TGRA_3 的设定值为止）。当 TCNT_3 和 TGRA_3 一致时，TCNT_3 就转换为递减计数；当 TCNT_3 和 TDDR 一致时，TCNTS 又转换为递增计数，重复此运行。

另外，将 TCNT_4 的初始值设定为 H'0000。

当 CST 位被置 1 时，TCNT_4 就与 TCNT_3 同步进行递增计数。当 TCNT_4 和 TCDR 一致时，TCNT_4 就转换为递减计数；当 TCNT_4 计数到 H'0000 时，TCNT_4 又转换为递增计数，重复此运行。

TCNTS 是只读计数器，不需要设定初始值。

在 TCNT_3、4 进行递增计数时，如果 TCNT_3 和 TCDR 一致，TCNTS 就开始递减计数。当 TCNTS 和 TCDR 一致时，TCNTS 就转换为递增计数；当 TCNTS 和 TGRA_3 一致时，TCNTS 就被清除为 H'0000。

在 TCNT_3、TCNT_4 进行递减计数时，如果 TCNT_4 和 TDDR 一致，TCNTS 就开始递增计数。当 TCNTS 和 TDDR 一致时，TCNTS 就转换为递减计数；当 TCNTS 计数到 H'0000 时，TCNTS 就被设定为 TGRA_3 的值。

TCNTS 只在计数运行的期间与设定 PWM 占空比的比较寄存器和暂存器比较。

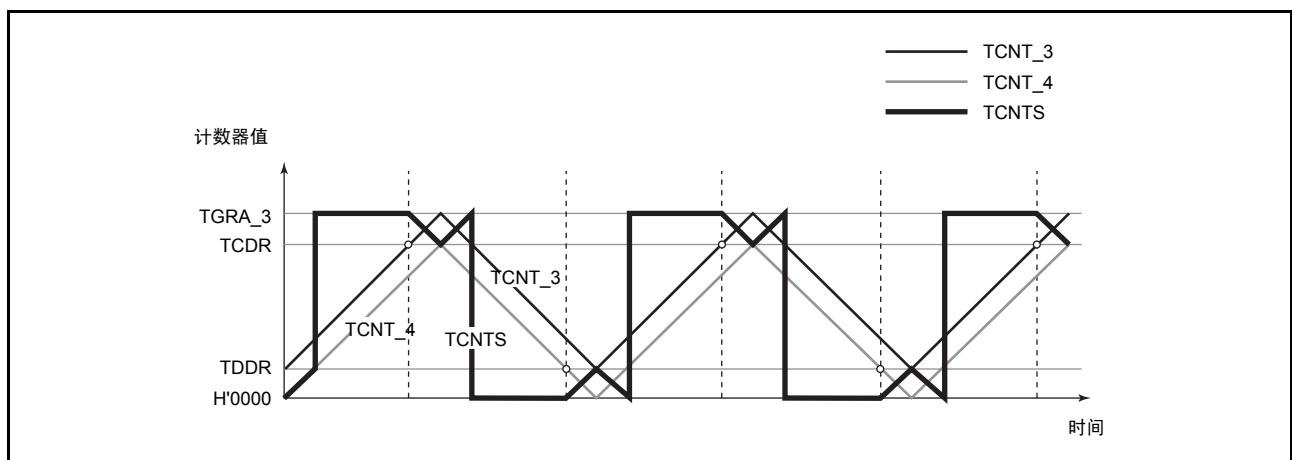


图 9.39 互补 PWM 模式的计数器运行

(b) 寄存器运行

在互补 PWM 模式中使用比较寄存器、缓冲寄存器和暂存器等 9 个寄存器。互补 PWM 模式的运行例子如图 9.40 所示。

为了进行 PWM 输出，寄存器 TGRB_3、TGRA_4、TGRB_4 总是和计数器比较。当这些寄存器和计数器一致时，定时器的输出控制寄存器 (TOCR) 的 OLSN、OLSP 位设定的值就被输出。

这些比较寄存器的缓冲寄存器为 TGRD_3、TGRC_4、TGRD_4。

另外，在缓冲寄存器和比较寄存器之间有暂寄存器，但是不能从 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 Tb 区间结束的 TCNTS 在进行递增计数时和 TGRA_3 一致或者在进行递减计数时计数到 H'0000，就将已传送到暂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 9.40 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 Tb (图 9.40 中的 Tb1) 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，对于 1 相的输出有 2 个比较匹配寄存器，比较寄存器存有更改前的数据，暂寄存器存有新更改后的数据。而且在此区间，TCNT_3、4 和 TCNTS (3 个计数器) 与比较寄存器和暂寄存器 (2 个寄存器) 比较，控制 PWM 输出。

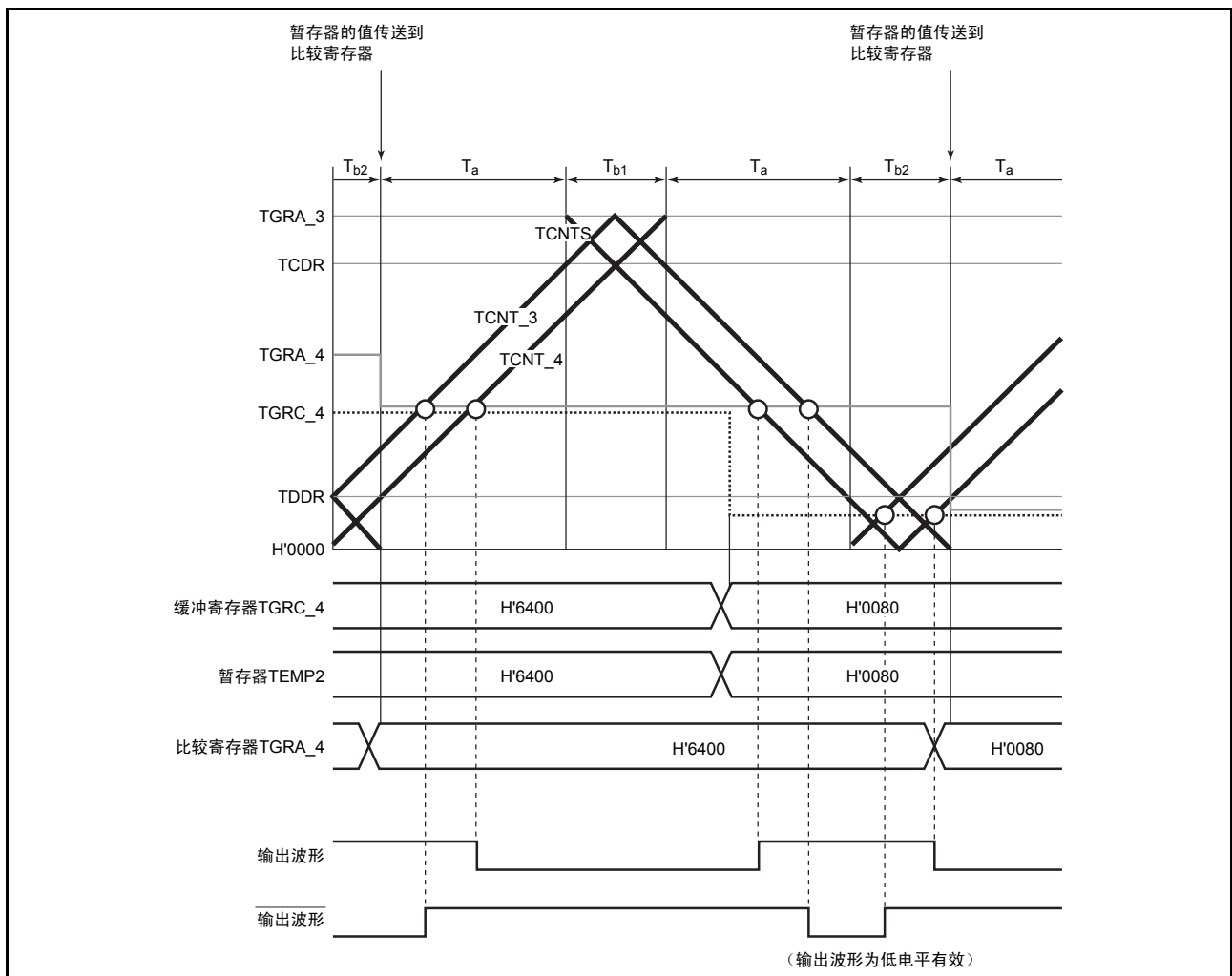


图 9.40 互补 PWM 模式的运行例

(c) 初始设定

在互补 PWM 模式中，有 6 个需要初始设定的寄存器，还有 1 个设定是否生成空载时间的寄存器（只能在不生成空载时间的情况下进行设定）。

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

TGRC_3 用作 TGRA_3 的缓冲寄存器，设定 1/2 的 PWM 载波周期 + 空载时间 Td。定时器的周期缓冲寄存器（TCBR）用作定时器的周期数据寄存器（TCDR）的缓冲寄存器，设定 1/2 的 PWM 载波周期，还给定定时器的空载时间寄存器（TDDR）设定空载时间 Td。

如果不生成空载时间，就将定时器的空载时间允许寄存器（TDER）的 TDER 位设定为 0，给 TGRC_3 和 TGRA_3 设定 1/2 的 PWM 载波周期 +1，并且给 TDDR 置 1。

给缓冲寄存器 TGRD_3、TGRC_4 和 TGRD_4（3 个寄存器）分别设定 PWM 占空比的初始值。

在设定为互补 PWM 模式的同时，将 5 个缓冲寄存器（TDDR 除外）的设定值分别传送到对应的比较寄存器。

另外，必须在设定为互补 PWM 模式前将 TCNT_4 设定为 H'0000。

表 9.56 需要初始设定的寄存器和计数器

寄存器 / 计数器	设定值
TGRC_3	1/2 的 PWM 载波周期 + 空载时间 Td (如果通过 TDER 设定为不生成空载时间，就为 1/2 的 PWM 载波周期 +1)
TDDR	空载时间 Td (如果通过 TDER 设定为不生成空载时间，就为 1)
TCBR	1/2 的 PWM 载波周期
TGRD_3、TGRC_4、TGRD_4	各相 PWM 占空比的初始值
TCNT_4	H'0000

【注】 TGRC_3 的设定值必须为设定在 TCBR 的 PWM 载波周期的 1/2 值和设定在 TDDR 的空载时间 Td 的值的和。但是，如果通过 TDER 设定为不生成空载时间，就为 1/2 的 PWM 载波周期 +1。

(d) PWM 输出电平的设定

在互补 PWM 模式中，通过定时器的输出控制寄存器 1（TOCR1）的 OLSN、OLSP 位或者定时器的输出控制寄存器 2（TOCR2）的 OLS1P ~ OLS3P、OLS1N ~ OLS3N 位，设定 PWM 脉冲的输出电平。

能按 3 相正相和 3 相反相（6 相输出）设定输出电平。

另外，必须在解除互补 PWM 模式的状态下进行输出电平的设定或者变更。

(e) 空载时间的设定

在互补 PWM 模式中，输出正相和反相为非重叠关系的 PWM 脉冲。此非重叠时间称为空载时间。

将非重叠时间设定到定时器的空载时间数据寄存器（TDDR）。设定在 TDDR 的值为 TCNT_3 计数器的初始值，生成 TCNT_3 和 TCNT_4 的非重叠关系。必须在解除互补 PWM 模式的状态下更改 TDDR 的内容。

(f) 不生成空载时间的设定

通过将定时器的空载时间允许寄存器 (TDER) 的 TDER 位设定为 0, 进行不生成空载时间的设定。只能在 TDER=1 的状态下读 TDER 后给 TDER 写 0 时, 将 TDER 置 0。

给 TGRA_3、TGRC_3 设定 1/2 的 PWM 载波周期 +1, 并且给定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 9.41 所示。

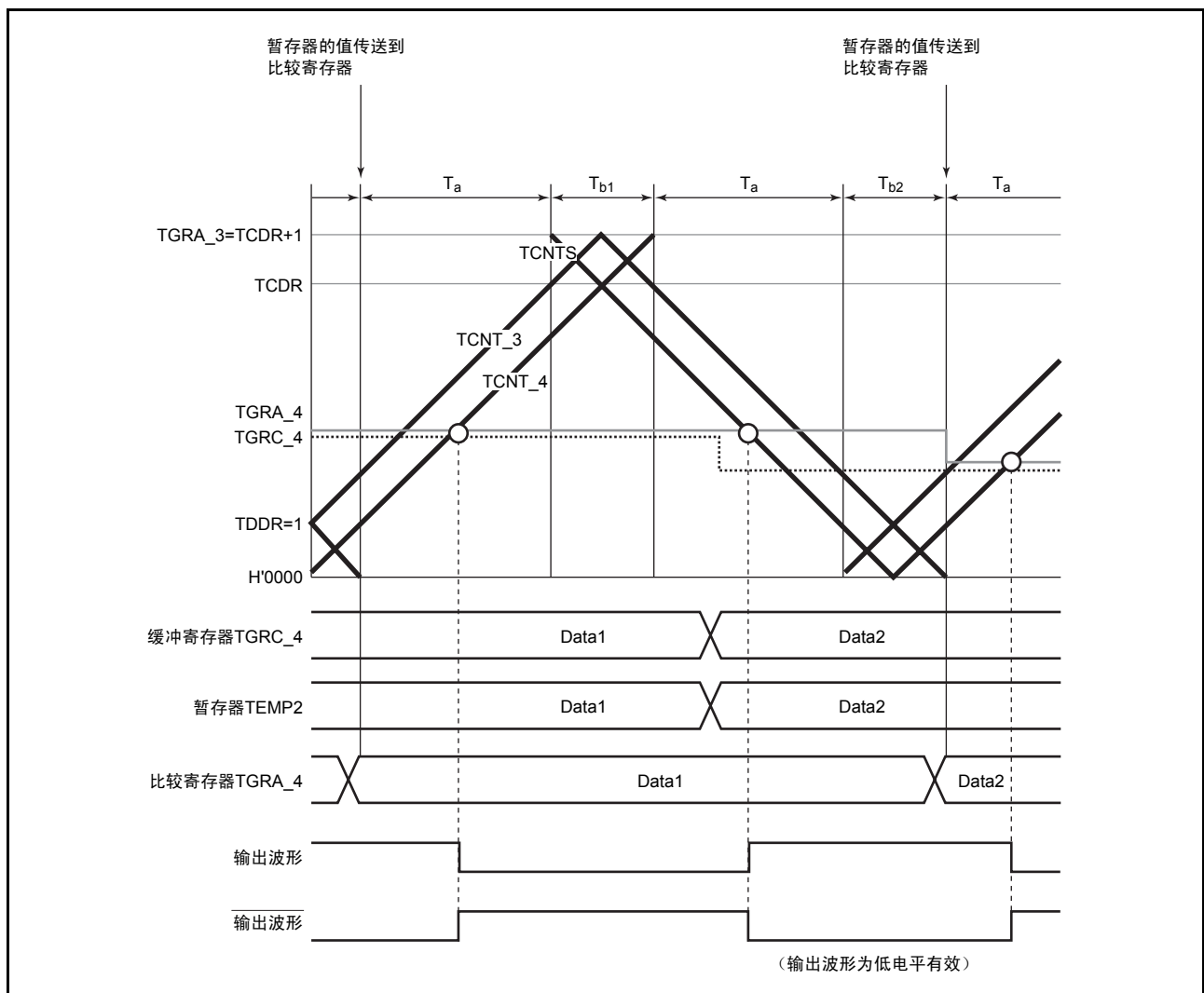


图 9.41 不生成空载时间的运行例

(g) PWM 周期的设定

在互补 PWM 模式中，将 PWM 脉冲的周期设定到 TGRA_3（设定 TCNT3 的上限值）和 TCDR（设定 TCNT_4 的上限值）这 2 个寄存器。必须将这 2 个寄存器设定为以下的关系：

生成空载时间：TGRA_3 的设定值 = TCDR 的设定值 + TDDR 的设定值

不生成空载时间：TGRA_3 的设定值 = TCDR 的设定值 + 1

另外，必须通过给缓冲寄存器的 TGRC_3、TCBR 设定值进行 TGRA_3、TCDR 的设定。在定时器的模式寄存器（TMDR）的 MD3 ~ MD0 选择的传送时序中，将设定在 TGRC_3 和 TCBR 的值同时传送到 TGRA_3 和 TCDR。

如果在波峰更新数据，就从下一个周期反映更改的 PWM 周期；如果在波谷更新数据，就从该周期反映更改的 PWM 周期。在波峰更改 PWM 周期时的运行例子如图 9.42 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”。

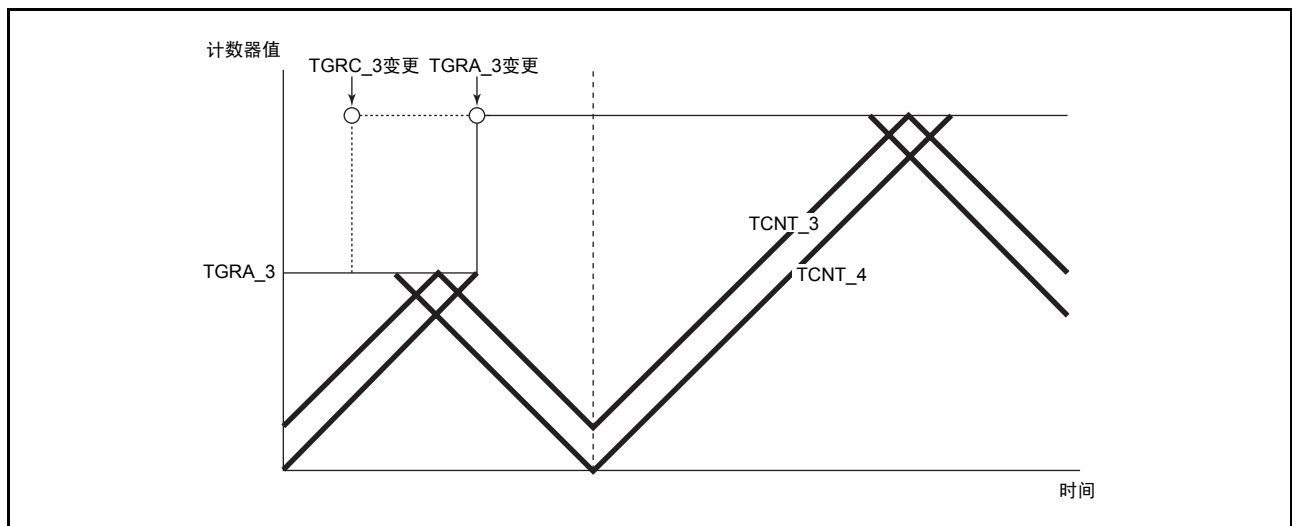


图 9.42 PWM 周期的变更例子

(h) 寄存器数据的更新

在互补 PWM 模式中更新比较寄存器的数据时，使用缓冲寄存器。更新的数据随时能写到缓冲寄存器。另外，能在有缓冲寄存器的运行中更改的寄存器为 5 个用于 PWM 占空比和载波周期的寄存器。

在这些寄存器和缓冲寄存器之间有各自的暂存器。在子计数器 TCNTS 不进行计数的期间，如果更新寄存器的数据，暂存器的值也会被改写。TCNTS 在进行计数中，不将缓冲寄存器的值传送到暂存器，而在 TCNTS 停止计数后将写在缓冲寄存器的值传送到暂存器。

在定时器的模式寄存器（TMDR）的 MD3 ~ MD0 位设定的数据更新时序中，将暂存器的值传送到比较寄存器。互补 PWM 模式中的数据更新例子如图 9.43 所示，此图是在计数器的波峰和波谷更新数据的模式。

在改写缓冲寄存器的数据时，最后必须写 TGRD_4。在写 TGRD_4 后，5 个寄存器全部同时将缓冲寄存器的数据传送到暂存器。

如果不全部更新 5 个寄存器或者不更新 TGRD_4 的数据，就必须在写要更新的寄存器数据后写 TGRD_4。此时，写到 TGRD_4 的数据必须和写之前的数据相同。

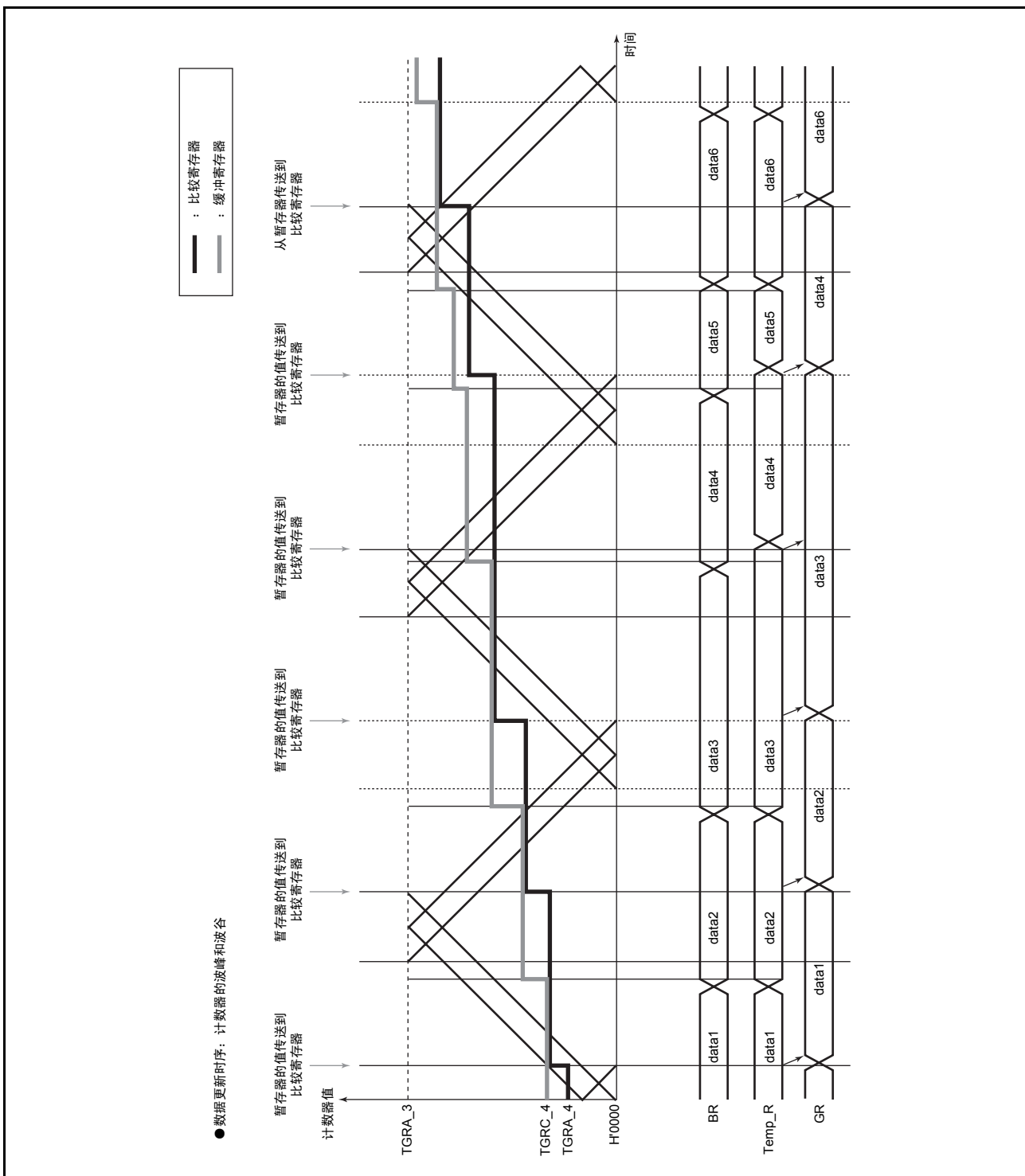


图 9.43 互补 PWM 模式的数据更新例

(i) 互补 PWM 模式的初始输出

在互补 PWM 模式中，通过设定定时器的输出控制寄存器 1 (TOCR1) 的 OLSN、OLSP 位或者定时器的输出控制寄存器 2 (TOCR2) 的 OLS1N ~ OLS3N、OLS1P ~ OLS3P 位，决定初始输出。

此初始输出为 PWM 脉冲的无效电平，从通过定时器的模式寄存器 (TMDR) 设定互补 PWM 模式到 TCNT_4 大于空载时间寄存器 (TDDR) 的设定值为止，输出此初始输出。互补 PWM 模式的初始输出例子如图 9.44 所示。

另外，PWM 占空比的初始值小于 TDDR 值的波形例子如图 9.45 所示。

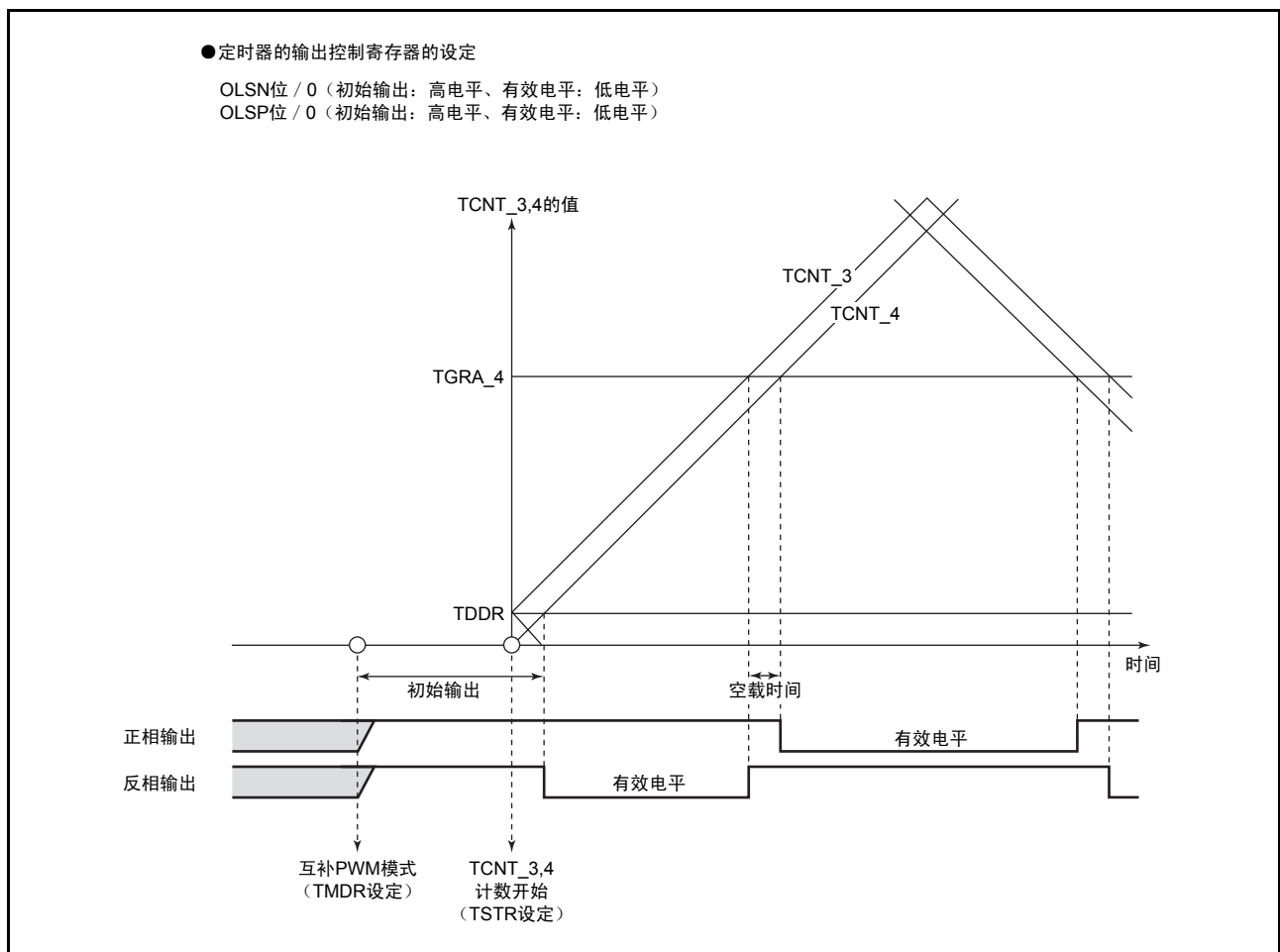


图 9.44 互补 PWM 模式的初始输出例 (1)

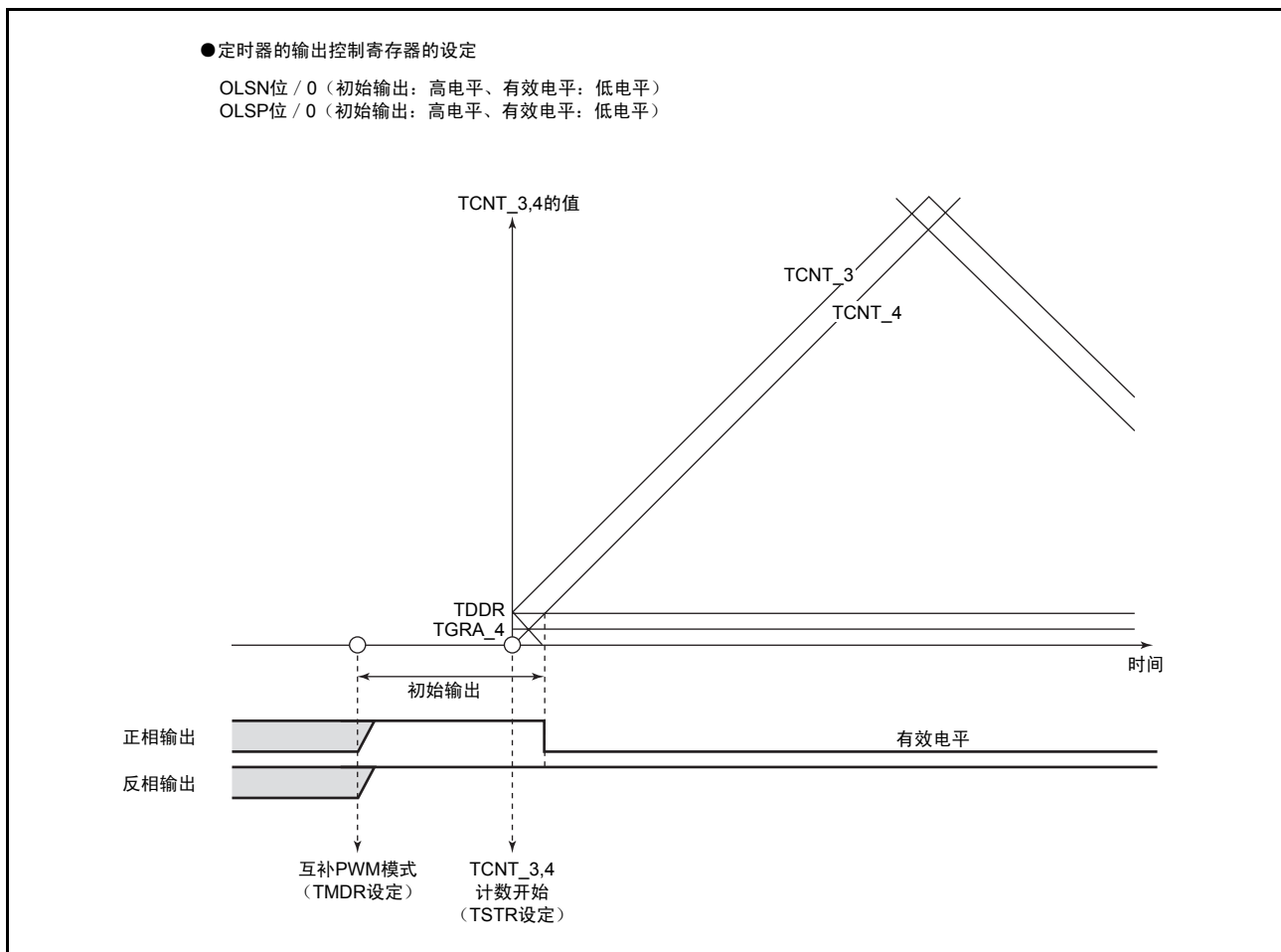


图 9.45 互补 PWM 模式的初始输出例 (2)

(j) 互补 PWM 模式的 PWM 输出生成方法

在互补 PWM 模式中, 将正相和反相有非重叠时间的 PWM 波形进行 3 相输出。此非重叠时间称为空载时间。

在发生计数器和数据寄存器的比较匹配时, 通过输出定时器的输出控制寄存器选择的输出电平生成 PWM 波形。另外, 在 TCNTS 进行计数的期间, 因为产生 0 ~ 100% 连续的 PWM 脉冲, 所以同时比较数据寄存器和暂存器的值。此时, 与 ON、OFF 比较匹配产生的时序会改变, 为了确保空载时间并且不使正相 / 反相的 ON 时间重叠, 必须优先 OFF 各相的比较匹配。互补 PWM 模式的波形生成例子如图 9.46 ~ 图 9.48 所示。

通过和实线计数器的比较匹配生成正相 / 反相的 OFF 时序, 通过和点线计数器 (比实线计数器迟空载时间后的计数器运行) 的比较匹配生成 ON 时序。在 T1 期间, 最优先使反相 OFF 的 a 的比较匹配, 忽视比 a 先产生的比较匹配。另外, 在 T2 期间, 最优先使正相 OFF 的 c 的比较匹配, 忽视比 c 先产生的比较匹配。

如图 9.46 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 9.47 所示,如果在 a 的比较匹配之后先产生 c 的比较匹配,就忽视 b 的比较匹配,而通过 d 的比较匹配使反相 OFF。这是为了通过比 b 的比较匹配 (正相 ON 时序) 先产生正相 OFF 的 c 的比较匹配,优先使正相 OFF (在 OFF 状态时,即使出现“OFF”命令信号,波形也不发生变化)。

同样地,在图 9.48 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。

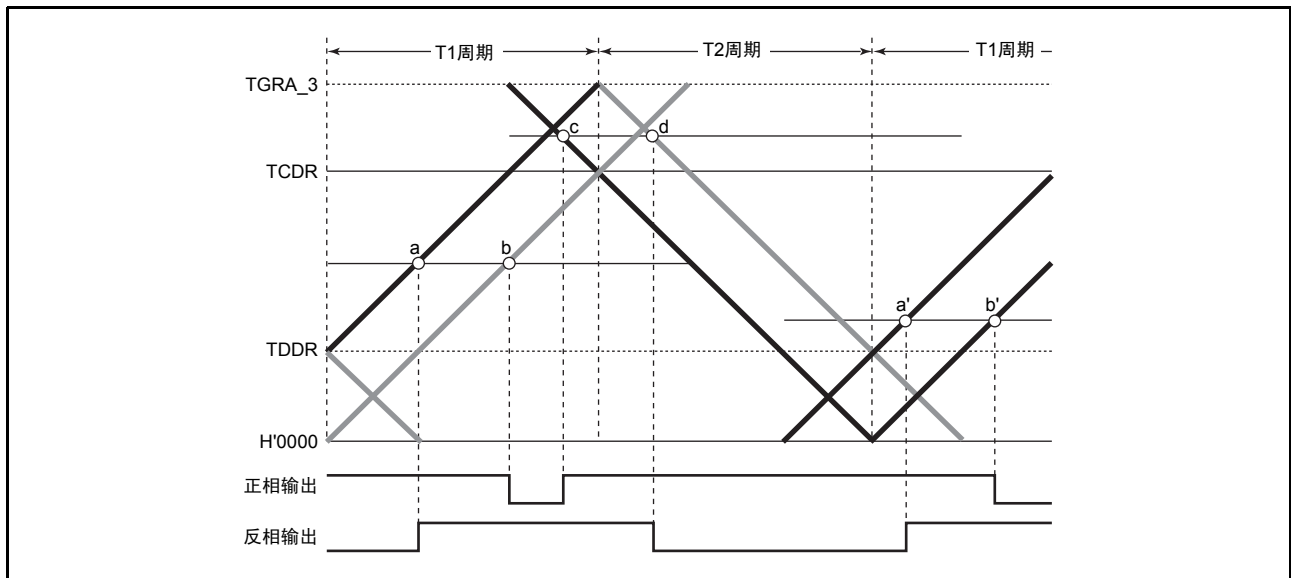


图 9.46 互补 PWM 模式的波形输出例 (1)

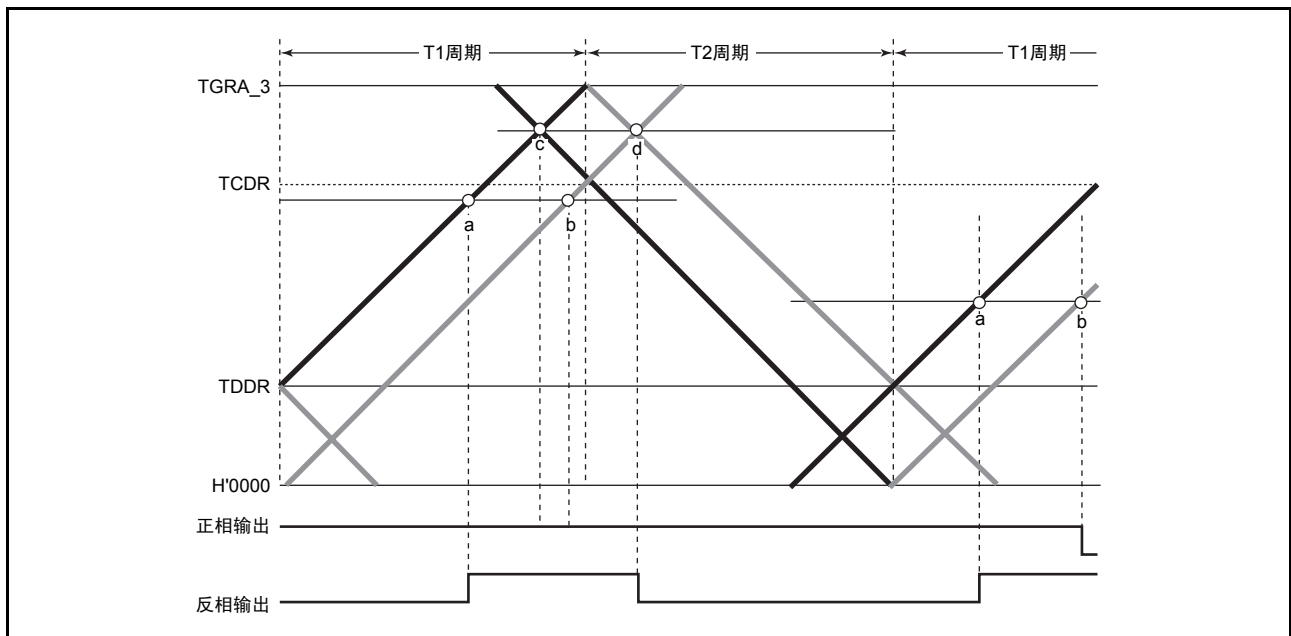


图 9.47 互补 PWM 模式的波形输出例 (2)

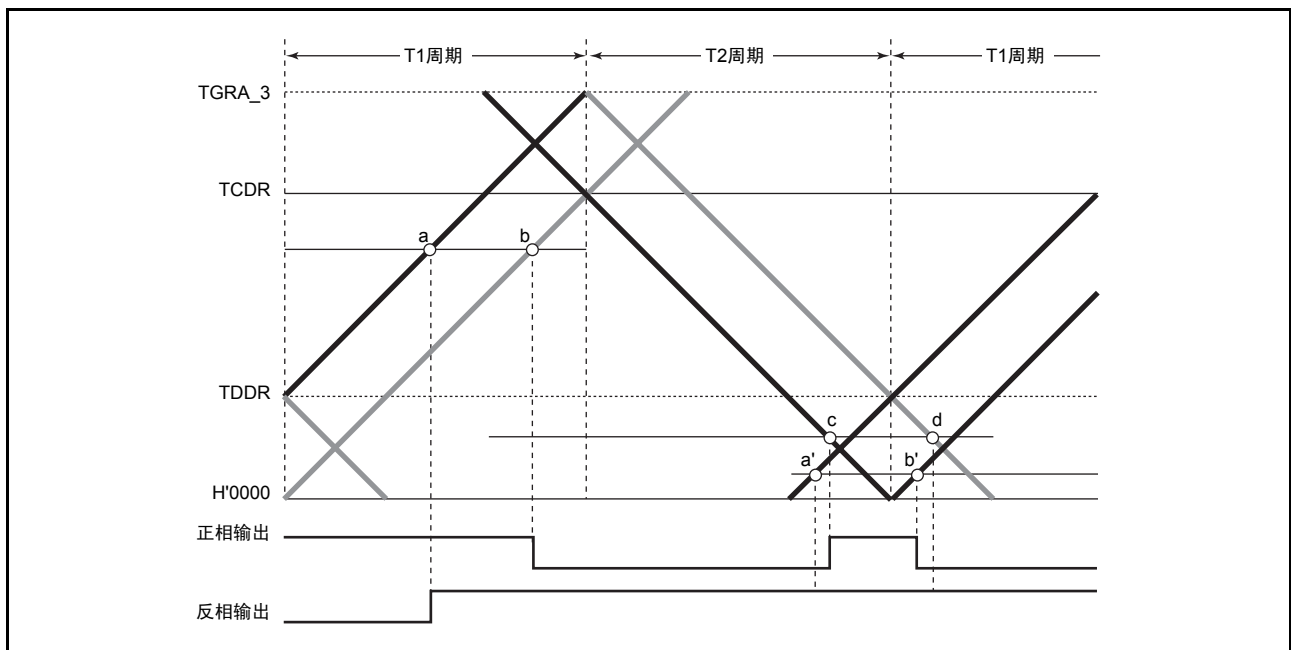


图 9.48 互补 PWM 模式的波形输出例 (3)

(k) 互补 PWM 模式的 0%、100% 占空比输出

在互补 PWM 模式中，能任意输出 0%、100% 的占空比，输出例子如图 9.49 ~ 图 9.53 所示。

如果将数据寄存器的值设定为 H'0000，就输出 100% 的占空比，此时的波形是正相为 100%ON 状态的波形。另外，如果将数据寄存器的值设定为 TGRA_3 的相同值，就输出 0% 的占空比，此时的波形是正相为 100%OFF 状态的波形。

此时，同时产生 ON 和 OFF 的比较匹配，但是，如果同相 ON 的比较匹配和 OFF 的比较匹配同时产生，双方的比较匹配都被忽视而波形不变。

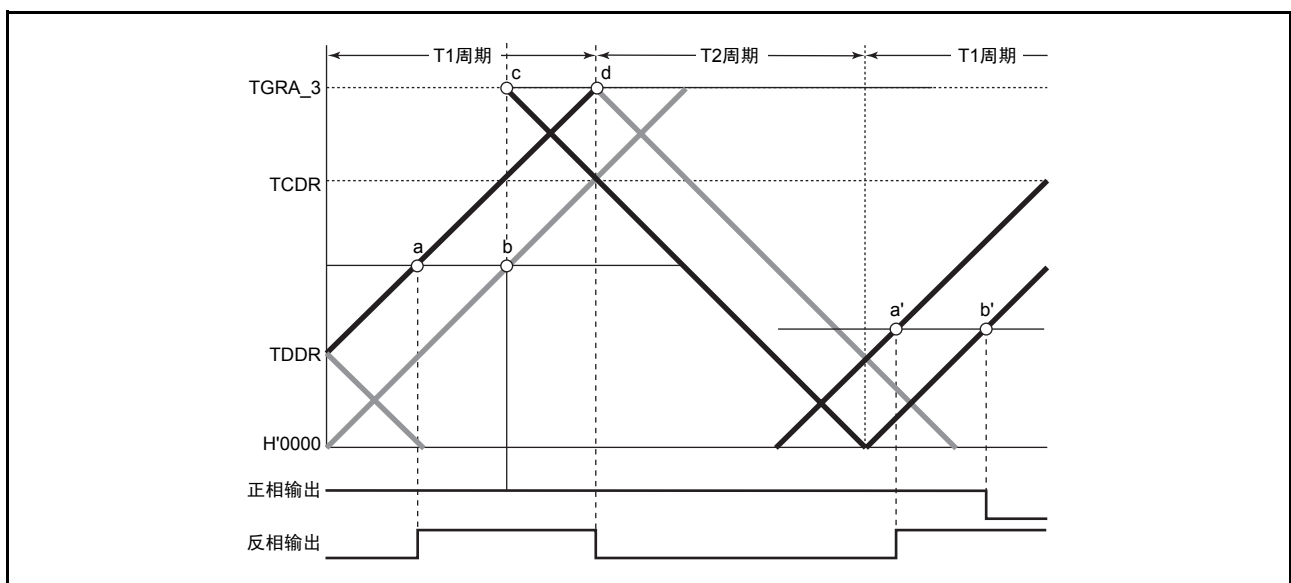


图 9.49 互补 PWM 模式的 0%、100% 波形输出例 (1)

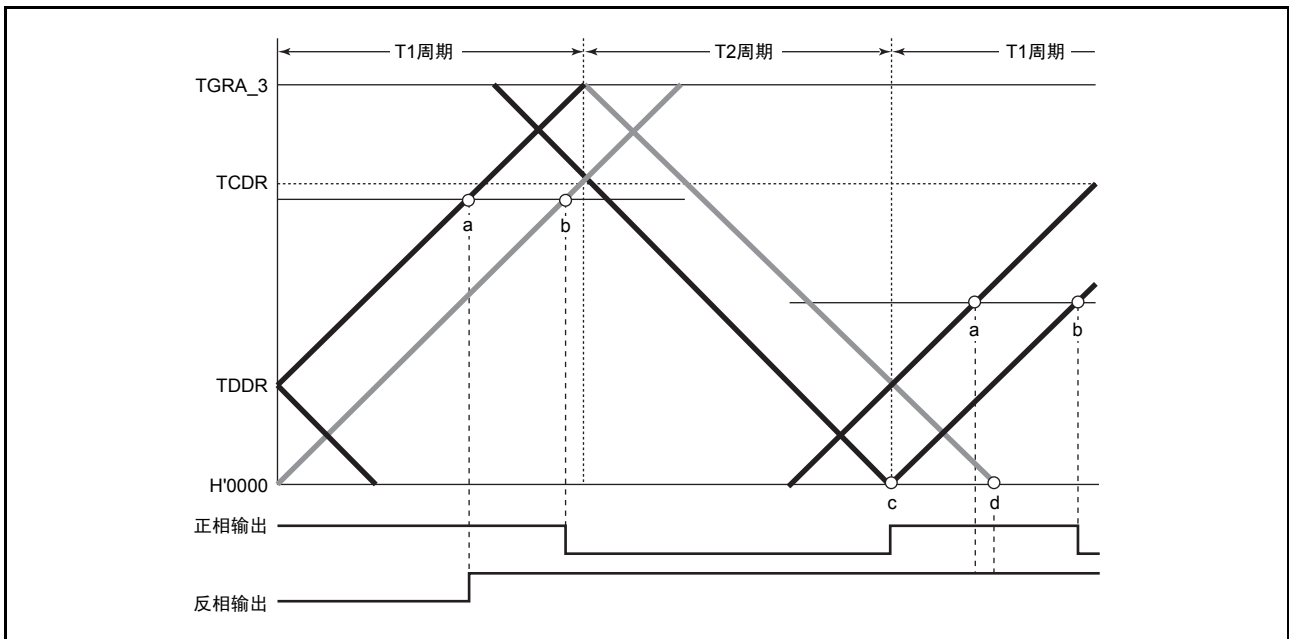


图 9.50 互补 PWM 模式的 0%、100% 波形输出例 (2)

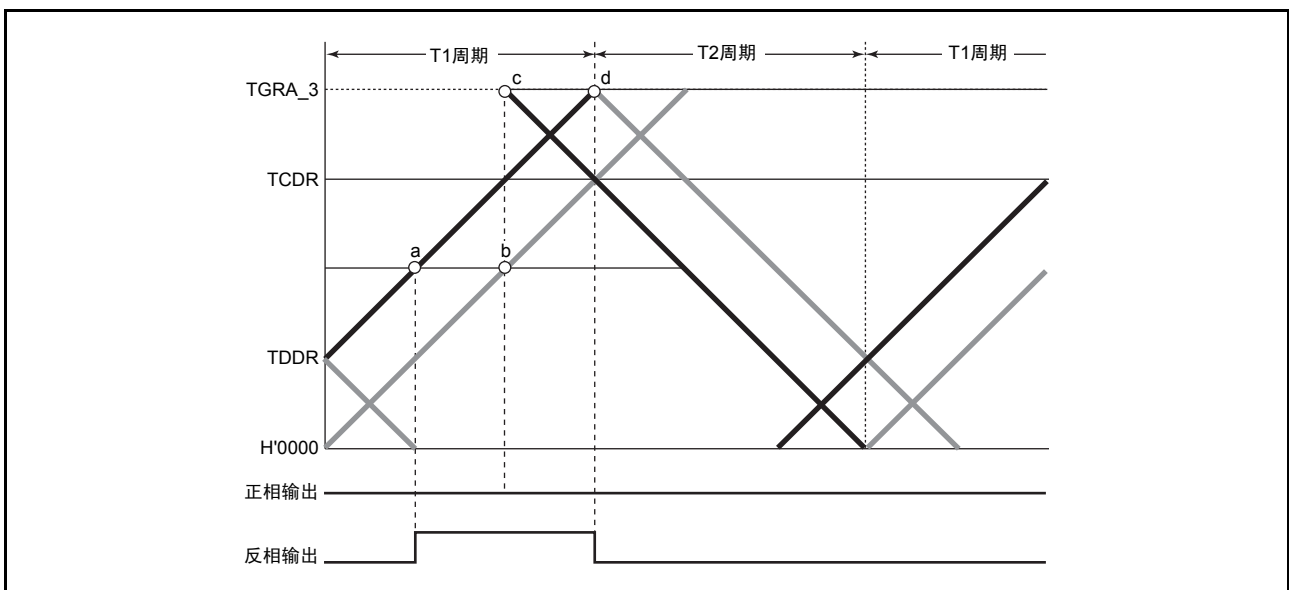


图 9.51 互补 PWM 模式的 0%、100% 波形输出例 (3)

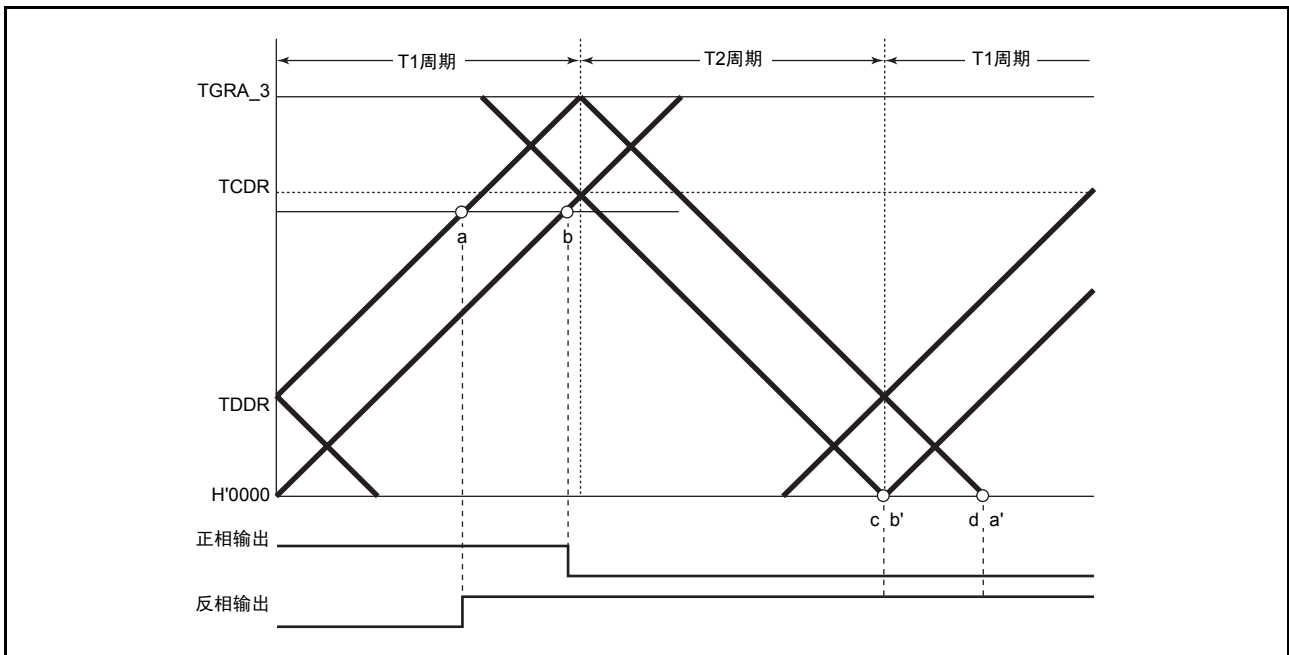


图 9.52 互补 PWM 模式的 0%、100% 波形输出例 (4)

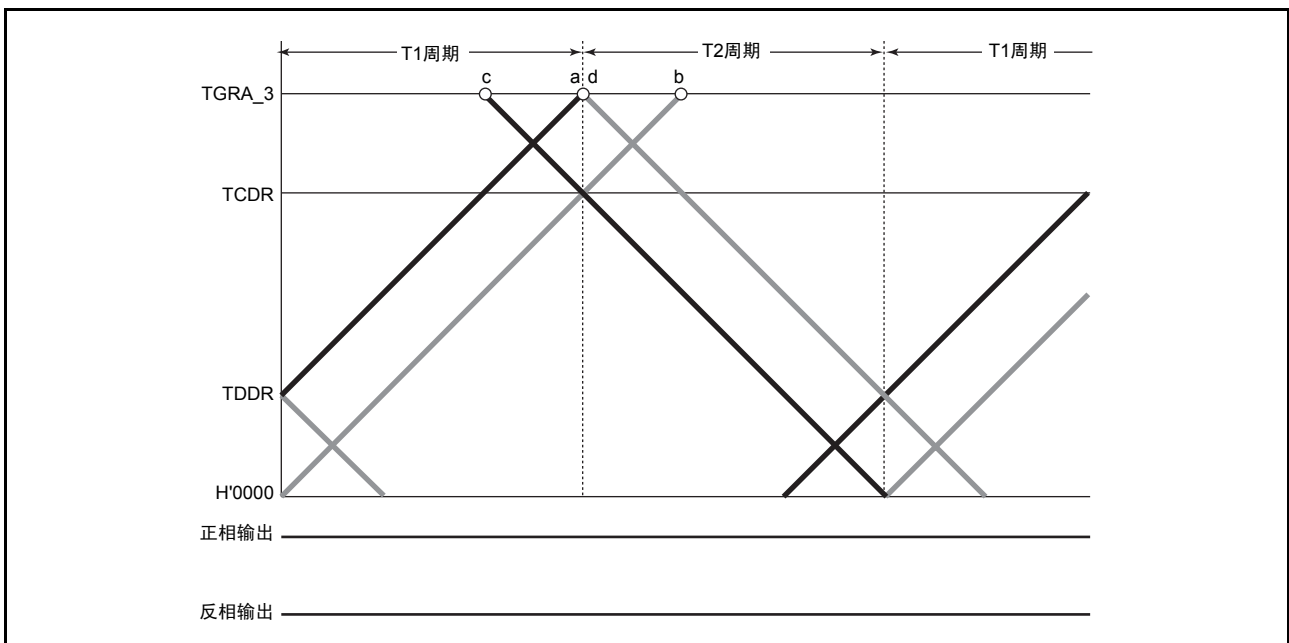


图 9.53 互补 PWM 模式的 0%、100% 波形输出例 (5)

(l) 与 PWM 周期同步的交替输出

在互补 PWM 模式中，能通过将定时器的输出控制寄存器（TOCR）的 PSYE 位设定为 1 进行与 PWM 载波周期同步的交替输出。交替输出的波形例子如图 9.54 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。

此交替输出的输出管脚是 TIOC3A 管脚，初始输出为 1。

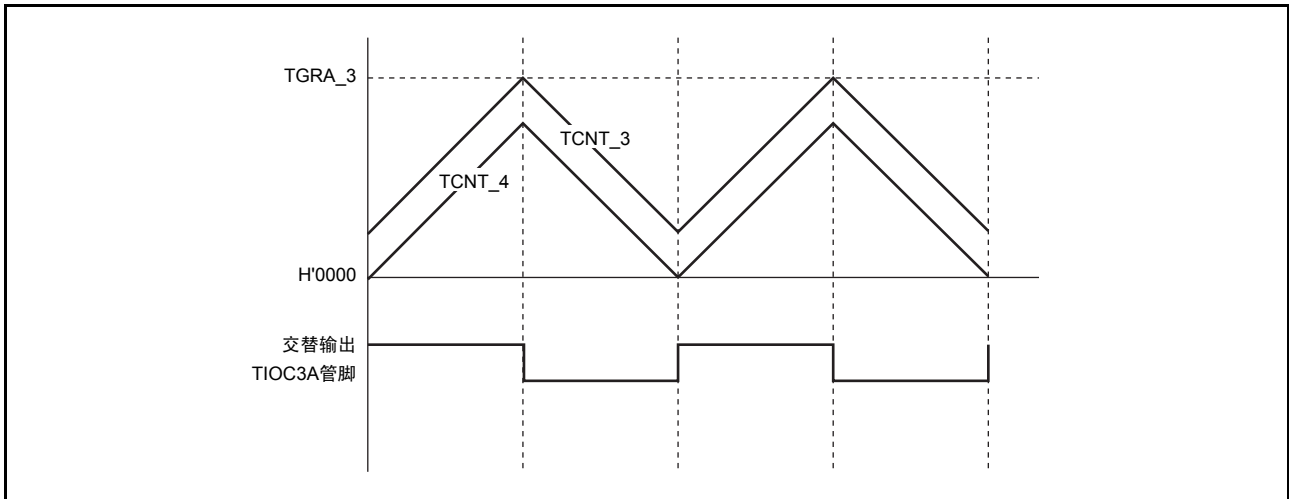


图 9.54 与 PWM 输出同步的交替输出的波形例子

(m) 其他通道的计数器清除

在互补 PWM 模式中，通过定时器的同步寄存器（TSYR）设定为与其他通道的同步模式，并且通过定时器的控制寄存器（TCR）的 CCLR2 ~ CCLR0 选择同步清除，就能由其他通道进行 TCNT_3、TCNT_4 和 TCNTS 的清除。

运行例子如图 9.55 所示。

使用此功能，能通过外部信号进行计数器的清除和重新开始。

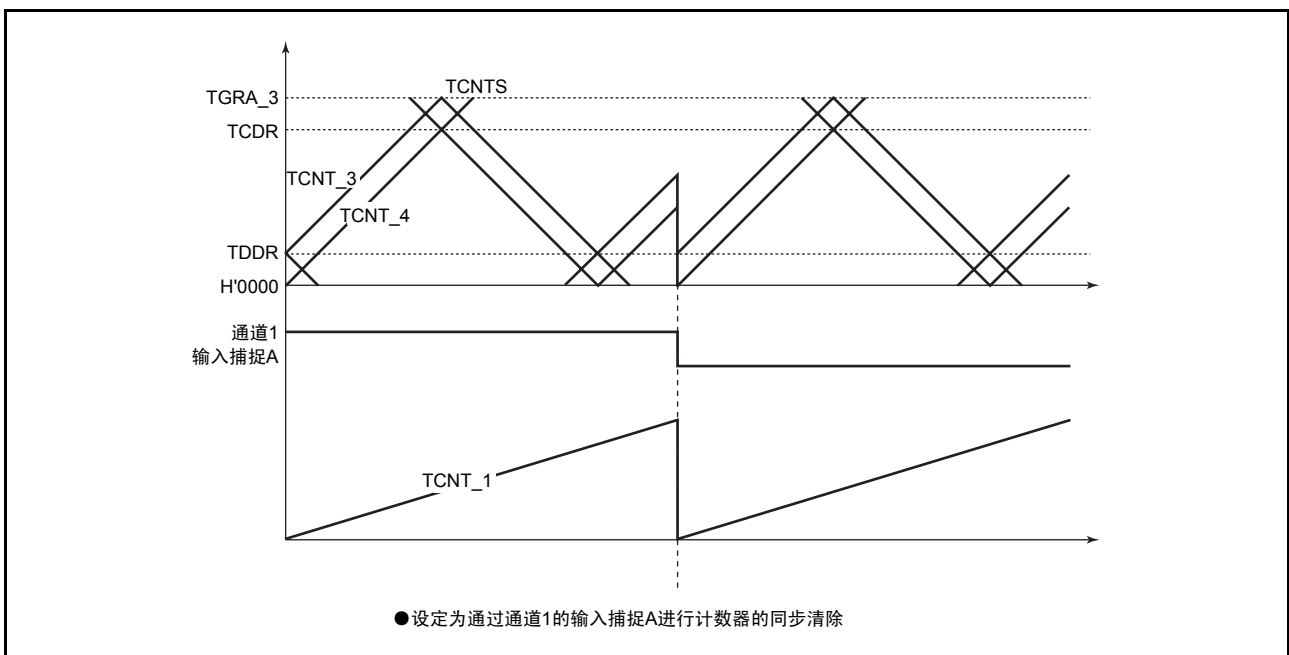


图 9.55 与其他通道同步的计数器清除

(n) 互补 PWM 模式中的同步计数器清除时的输出波形控制

能通过将 TWCR 寄存器的 WRE 位设定为 1，抑制在互补 PWM 模式的波谷 Tb 区间发生同步计数器清除时的初始输出。因此，能抑制同步计数器清除时占空比的急剧变化。

只在如图 9.56 的(10)、(11)的波谷 Tb 区间进行同步清除时，才能抑制初始输出（通过将 WRE 位设定为 1）。如果在其他时序中发生同步清除，就输出 TOCR 寄存器的 OLS 位设定的初始值。另外，即使在波谷的 Tb 区间，如果在图 9.56 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。

MTU2 计数器的清除源是来自 MTU2 的通道 0 ~ 2 的同步清除。

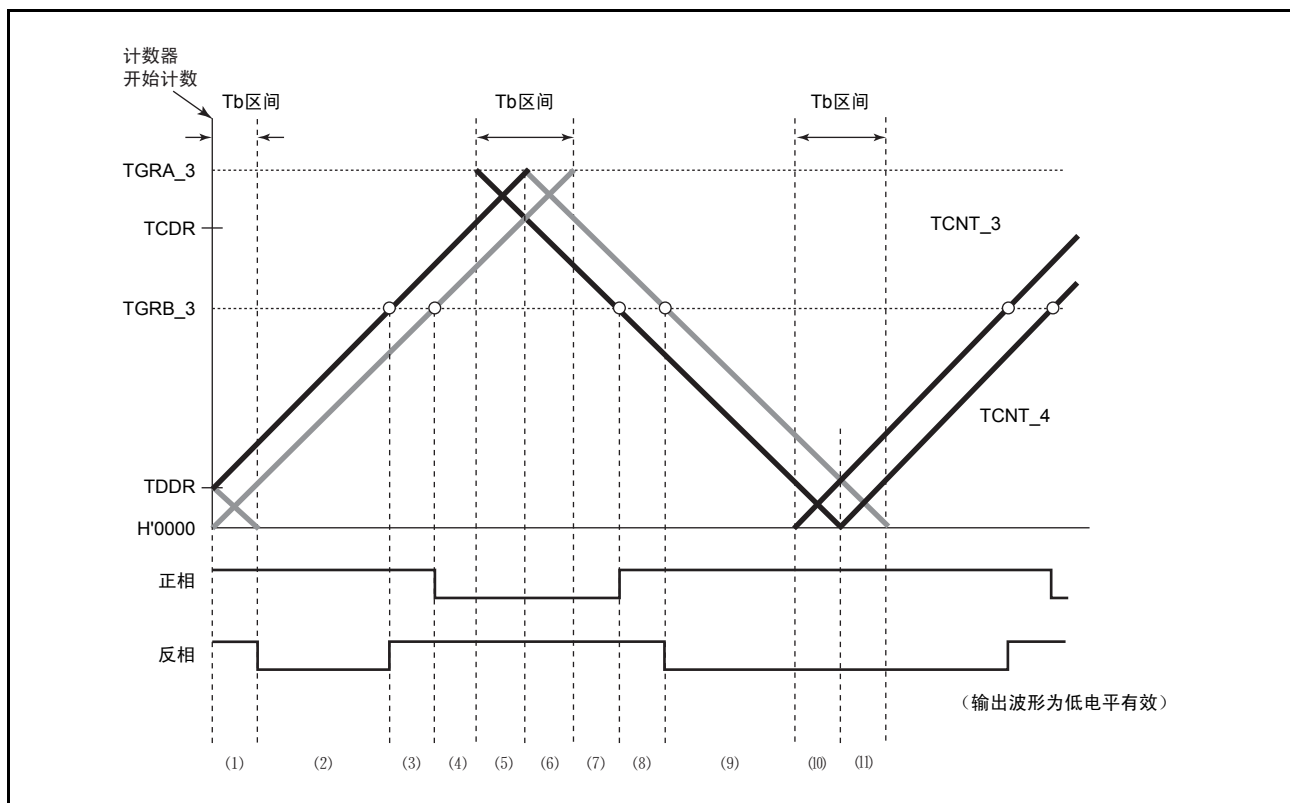


图 9.56 同步计数器清除的时序

• 互补 PWM 模式中的同步计数器清除时的输出波形控制的设定步骤例子

互补 PWM 模式中的同步计数器清除时的输出波形控制的设定步骤例子如图 9.57 所示。

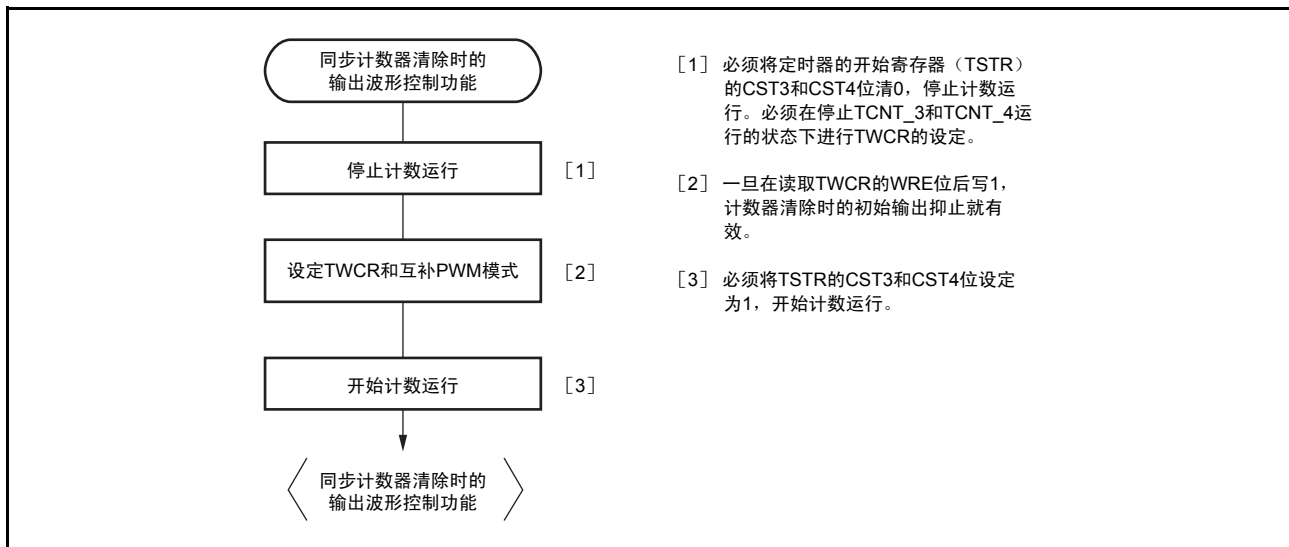


图 9.57 互补 PWM 模式中的同步计数器清除时的输出波形控制的设定步骤例子

• 互补 PWM 模式中的同步计数器清除时的输出波形控制的运行例

在将 TWCR 的 WRE 位设定为 1 的状态下使 MTU2 进行互补 PWM 运行，并进行同步计数器清除时的运行例子如图 9.58 ~ 图 9.61 所示。在此，图 9.58 ~ 图 9.61 的同步计数器清除的时序分别是图 9.56 的(3)、(6)、(8)、(11)所示的时序。

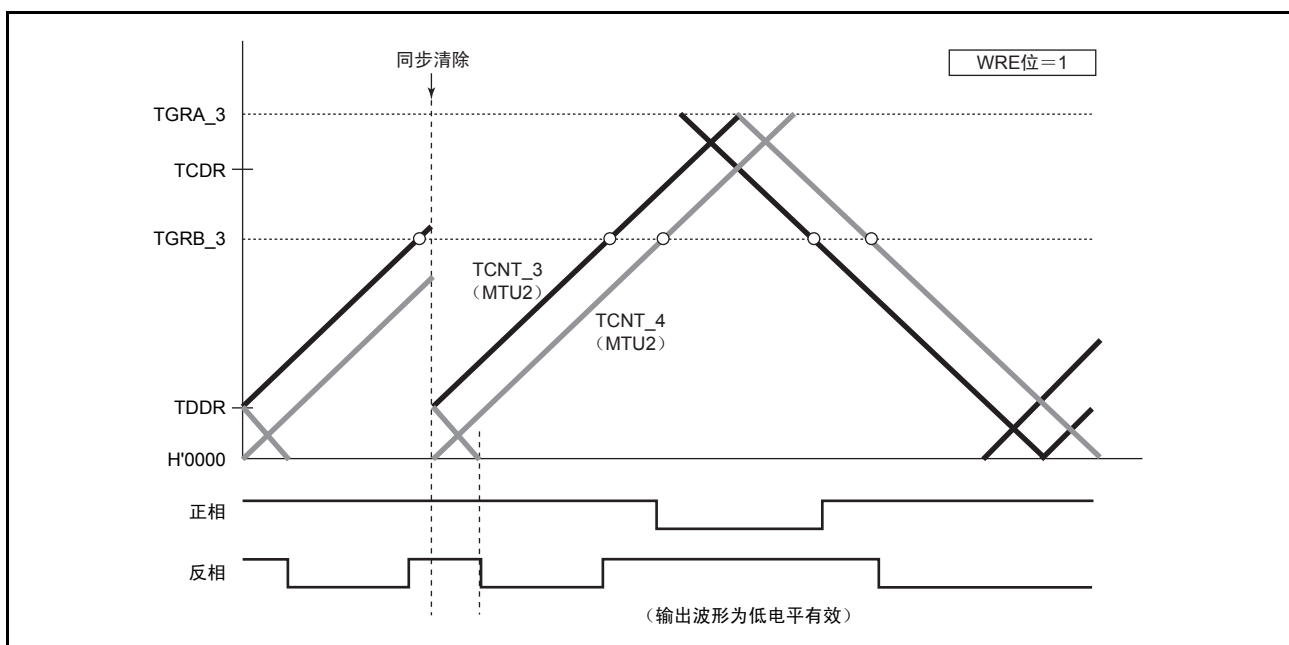


图 9.58 在递增计数中的空载时间发生同步清除
(图 9.56 的时序(3)、MTU2 的 TWCR 寄存器的 WRE 位 = 1)

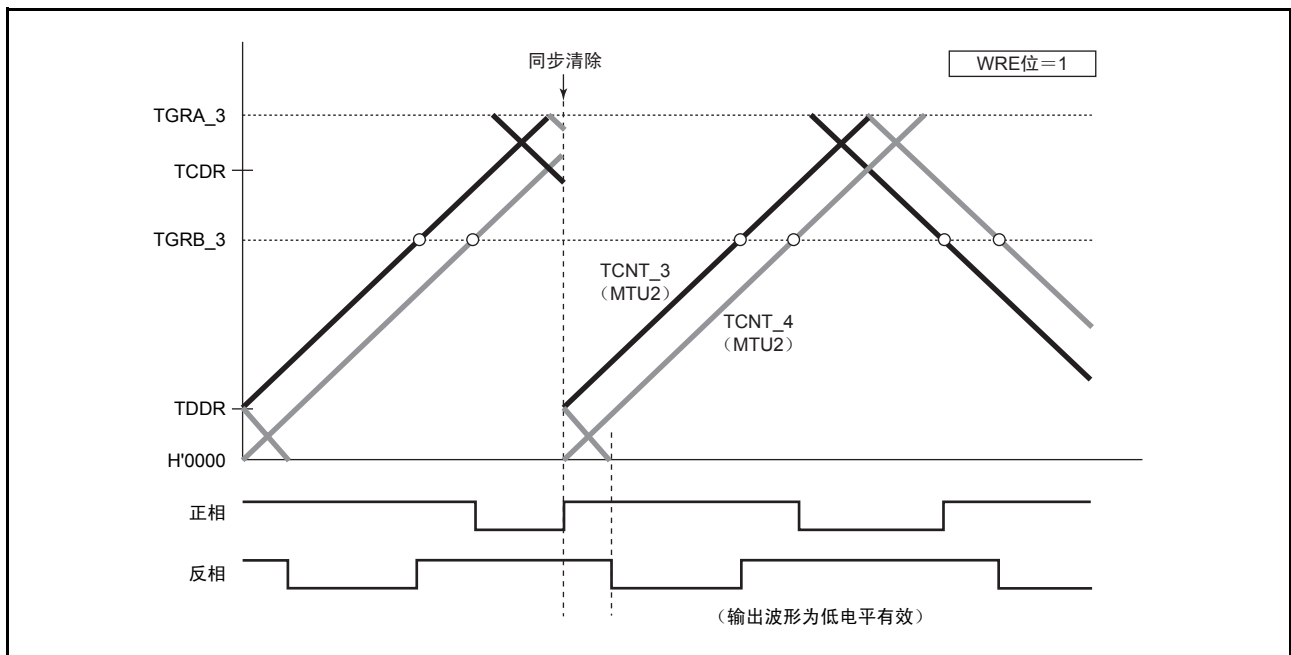


图 9.59 在波峰的 Tb 区间发生同步清除
(图 9.56 的时序(6)、MTU2 的 TWCR 寄存器的 WRE 位 =1)

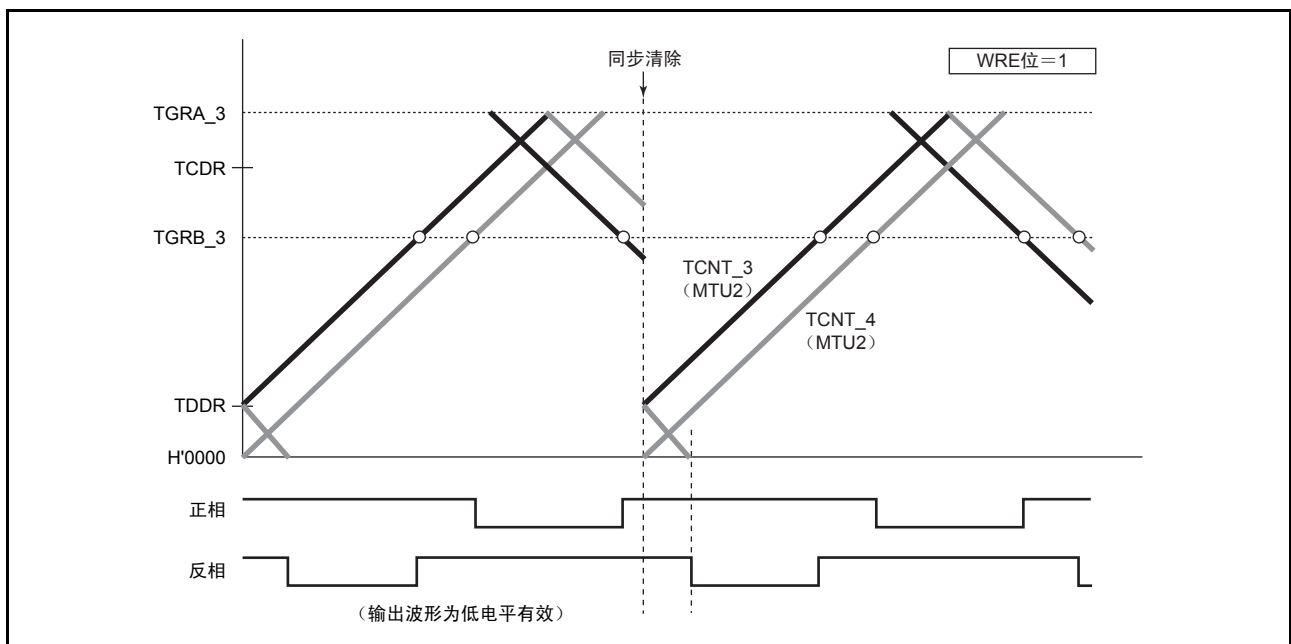


图 9.60 在递减计数中的空载时间发生同步清除
(图 9.56 的时序(8)、TWCR 寄存器的 WRE 位 =1)

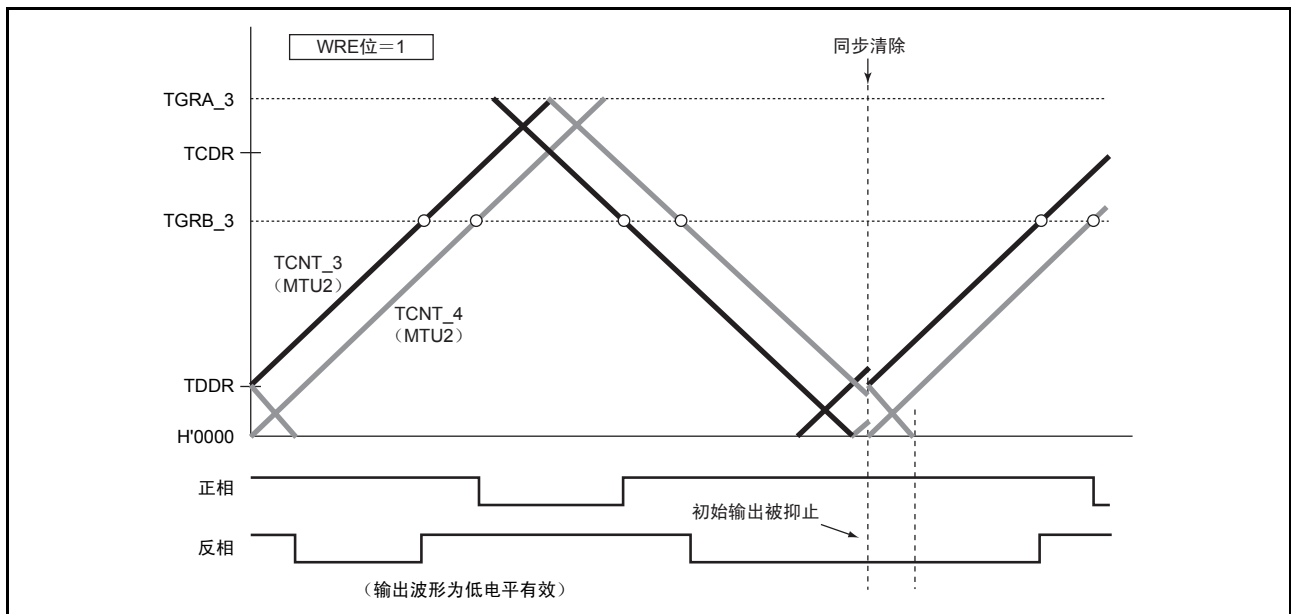


图 9.61 在波谷的 Tb 区间发生同步清除
(图 9.56 的时序(II)、TWCR 寄存器的 WRE 位 =1)

(o) 通过 TGRA_3 的比较匹配清除计数器

在互补 PWM 模式中，能通过设定定时器的波形控制寄存器 (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 9.62 所示。

- 【注】
1. 只能在互补 PWM 模式 1 (在波峰进行传送) 中使用。
 2. 不能设定为和其他通道同步的清除功能 (不能将定时器的同步寄存器 (TSYR) 的 SYNC0 ~ SYNC4 位和定时器的同步清除寄存器 (TSYCR) 的 CE0A/B/C/D、CE1A/B/C/D 位设定为 1)。
 3. PWM 占空比不能置 H'0000。
 4. 不能将定时器的输出控制寄存器 1 (TOCR1) 的 PSYE 位设定为 1。

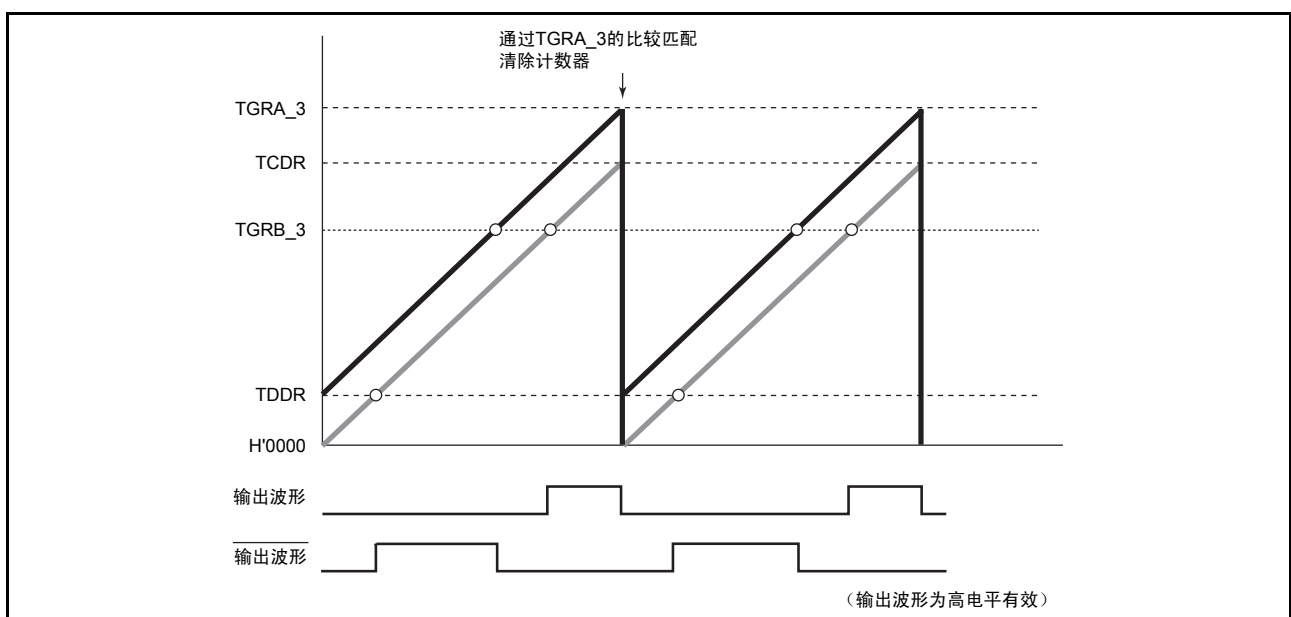


图 9.62 通过 TGRA_3 的比较匹配清除计数器的运行例

(p) AC 同步马达 (无刷 DC 马达) 的驱动波形输出例

在互补 PWM 模式中, 能使用定时器的门控寄存器 (TGCR), 简单地控制无刷 DC 马达。使用 TGCR 的无刷 DC 马达的驱动波形例子如图 9.63 ~ 图 9.66 所示。

为了转换 3 相无刷 DC 马达的输出相, 在使用霍尔元件等检测外部信号时, 将 TGCR 的 FB 位设定为 0。此时, 将表示磁极位置的外部信号输入到通道 0 的定时器的输入管脚 TIOC0A、TIOC0B 和 TIOC0C 管脚 (必须通过 PFC 进行设定)。如果 TIOC0A、TIOC0B 和 TIOC0C 这 3 个管脚检测到边沿, 输出的 ON/OFF 就自动转换。

当 FB 位是 1 时, 如果给 TGCR 的 UF、VF、WF 各位设定 0 或者 1, 就会转换输出的 ON/OFF。

从互补 PWM 模式的 6 相输出管脚进行驱动波形的输出。

对于此 6 相输出, 能通过将 N 位或者 P 位设定为 1, 在 ON 输出时, 使用互补 PWM 模式的输出, 进行载波输出。在 N 位或者 P 位是 0 时, 为电平输出。

另外, 6 相输出的有效电平 (ON 输出时的电平) 与 N 位和 P 位的设定无关, 能通过定时器的输出控制寄存器 (TOCR) 的 OLSN 位和 OLSP 位进行设定。

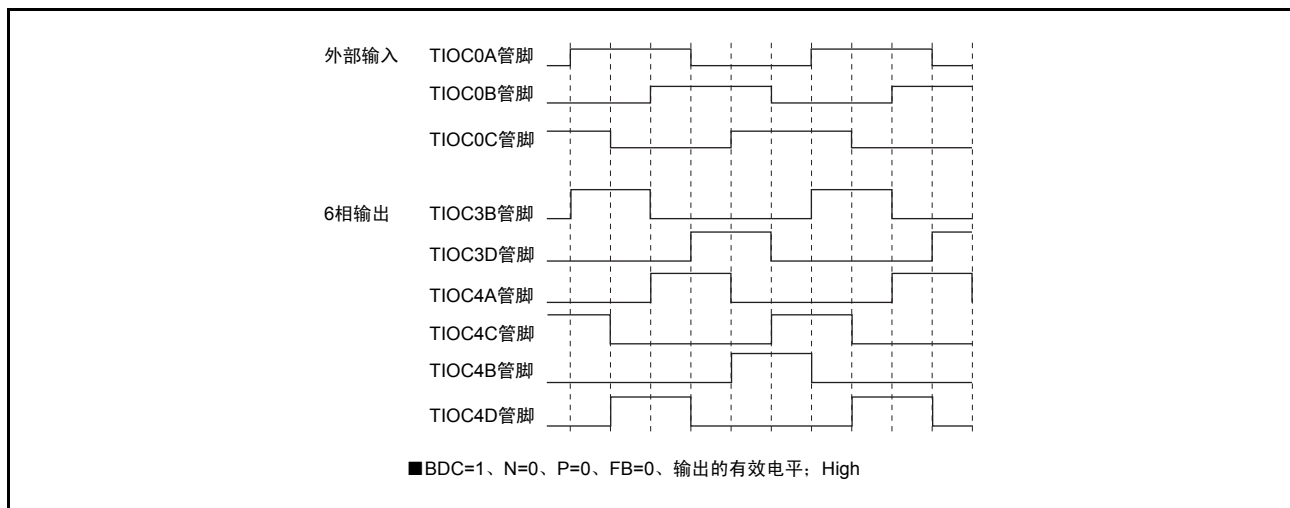


图 9.63 通过外部输入进行输出相转换的运行例 (1)

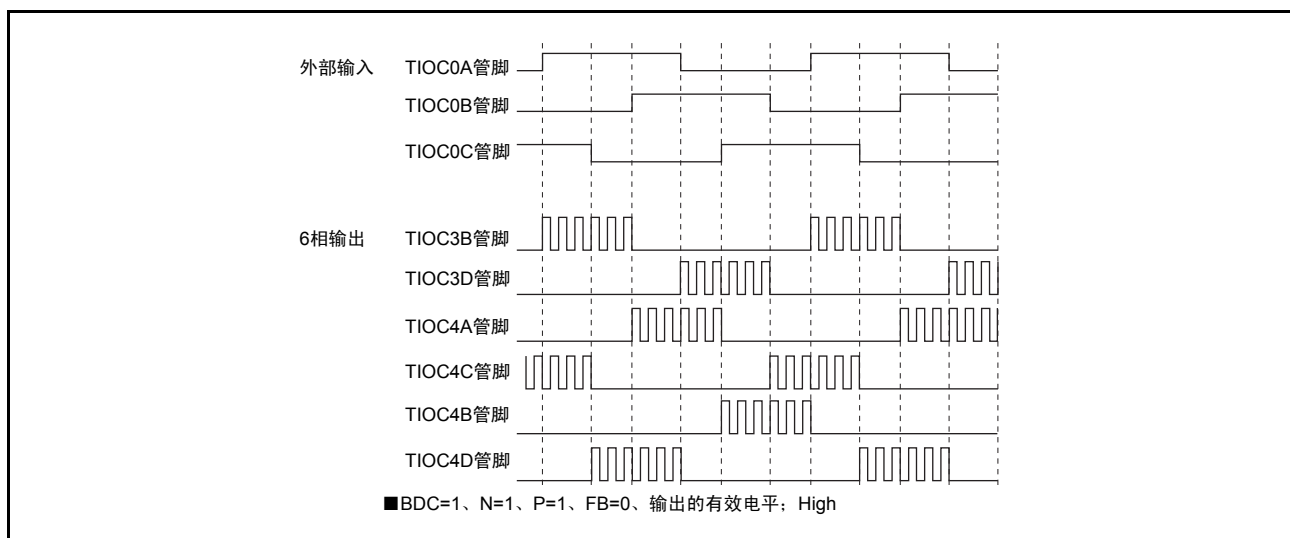


图 9.64 通过外部输入进行输出相转换的运行例 (2)

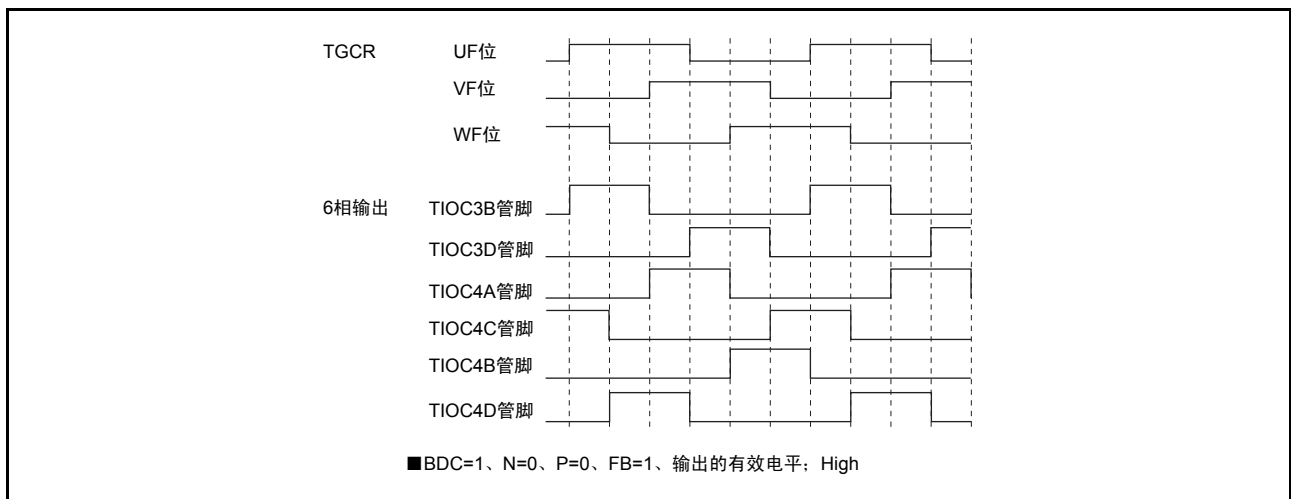


图 9.65 通过设定 UF、VF、WF 位进行输出相转换的运行例 (3)

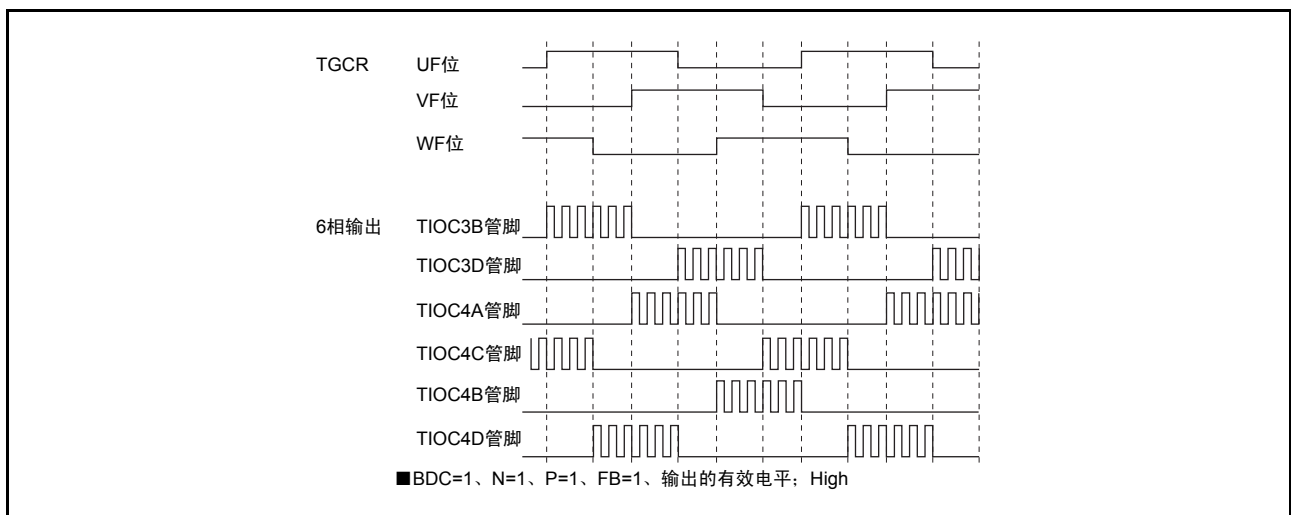


图 9.66 通过设定 UF、VF、WF 位进行输出相转换的运行例 (4)

(q) A/D 转换开始请求的设定

在互补 PWM 模式中，能使用 TGRA_3 的比较匹配、TCNT_4 的下溢（波谷）和通道 3、4 以外通道的比较匹配进行 A/D 转换的开始请求。

如果使用 TGRA_3 的比较匹配设定开始请求，就能在 TCNT_3 的波峰开始 A/D 转换。

能通过将定时器的中断允许寄存器（TIER）的 TTGE 位设定为 1，设定 A/D 转换的开始请求，并能通过将 TIER_4 的 TTGE2 位设定为 1，设定 TCNT_4 的下溢（波谷）的 A/D 转换的开始请求。

(3) 互补 PWM 模式的跳过中断功能

通过设定定时器的跳过中断设定寄存器 (TITCR)，最多能使通道 3 和通道 4 的 TGIA_3 (波峰的中断) 和 TCIV_4 (波谷的中断) 减少 7 次中断。

通过设定定时器的缓冲传送寄存器 (TBTER)，能与缓冲寄存器到暂存器 / 比较寄存器的数据传送联动，减少中断。有关和缓冲寄存器的联动，请参照“(c) 和跳过中断功能联动的缓冲传送控制”。

通过设定定时器的 A/D 转换请求控制寄存器 (TADCR)，能与 A/D 转换开始请求延迟功能的 A/D 转换开始请求联动，减少中断。有关和 A/D 转换请求延迟功能的联动，请参照“9.4.9 A/D 转换开始请求的延迟功能”。

必须通过设定 TIER_3、TIER_4 寄存器，在将 TGIA_3 和 TCIV_4 中断请求设定为禁止状态后，并且在不产生由比较匹配引起的 TGFA_3 和 TCFV_4 标志置位的状态下，设定定时器的跳过中断设定寄存器 (TITCR)。另外，在更改减少次数前，必须将 T3AEN 和 T4VEN 位设定为 0，清除减少计数器。

(a) 跳过中断功能的设定步骤例子

跳过中断功能的设定步骤例子和跳过中断次数的可更改期间分别如图 9.67 和图 9.68 所示。

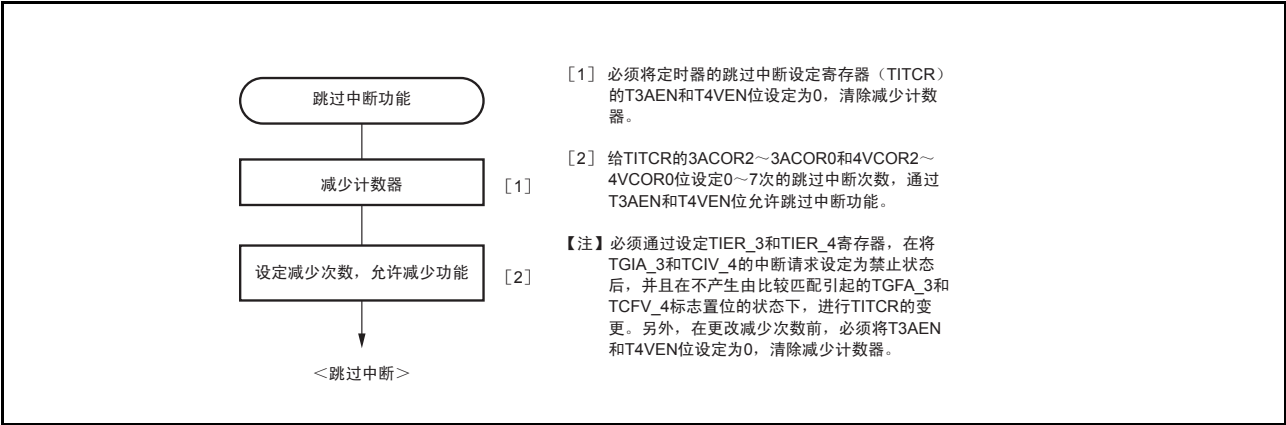


图 9.67 跳过中断功能的设定步骤例子

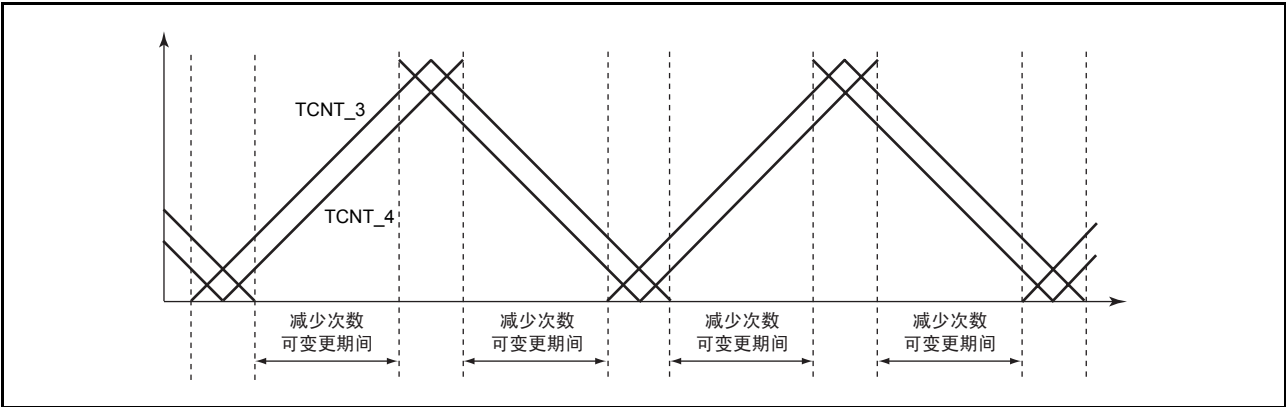


图 9.68 跳过中断次数的可更改期间

(b) 跳过中断功能的运行例

在通过定时器的跳过中断设定寄存器 (TITCR) 的 3ACOR 位将跳过中断次数设定为 3 次并将 T3AEN 位设定为 1 时, TGIA_3 跳过中断功能的运行例子如图 9.69 所示。

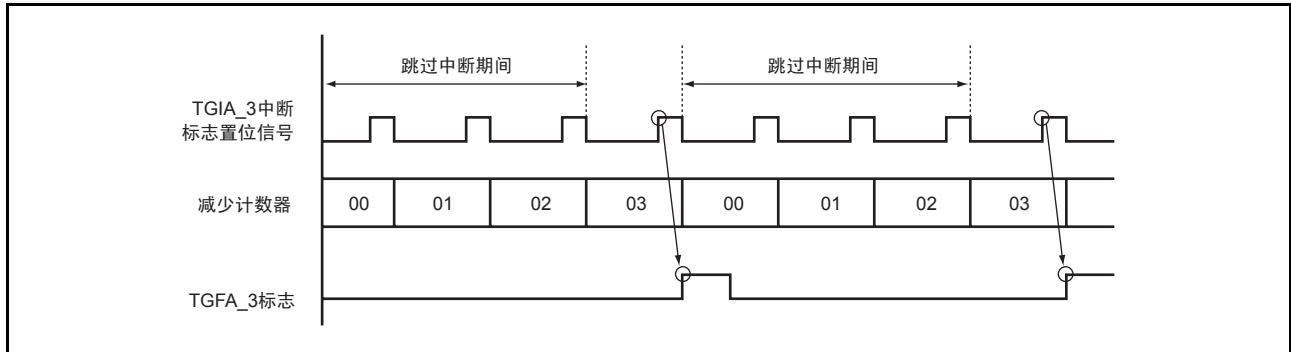


图 9.69 跳过中断功能的运行例

(c) 和跳过中断功能联动的缓冲传送控制

通过设定定时器的缓冲传送设定寄存器 (TBTER) 的 BTE1、BTE0 位, 可选择在互补 PWM 模式中是否进行缓冲寄存器到暂寄存器的缓冲传送, 或者设定是否与跳过中断功能联动。

设定为抑制缓冲传送 (BTE1=0、BTE0=1) 时的运行例子如图 9.70 所示。在此设定期间, 不将缓冲寄存器的值传送到暂寄存器。

设定为缓冲传送与跳过中断功能联动 (BTE1=1、BTE0=0) 时的运行例子如图 9.71 所示。此设定在缓冲传送允许期间以外, 不将缓冲寄存器的值传送到暂寄存器。

另外, 如果将定时器的跳过中断设定寄存器 (TITCR) 的 T3AEN 位或者 T4VEN 位或者 T3AEN/T4VEN 位设定为 1, 各缓冲传送允许期间就不同。TITCR 的 T3AEN、T4VEN 位的设定和缓冲传送允许期间的关系如图 9.72 所示。

【注】 本功能必须和跳过中断功能配合使用。

在禁止跳过中断功能时 (在将定时器的跳过中断设定寄存器 (TITCR) 的 T3AEN、T4VEN 位或者 TITCR 的减少次数设定位 (3ACOR 和 4VCOR) 置 0 时), 必须设定为缓冲传送不与跳过中断功能联动 (将定时器的缓冲传送设定寄存器 (TBTER) 的 BTE1 置 0), 否则就不进行缓冲传送。

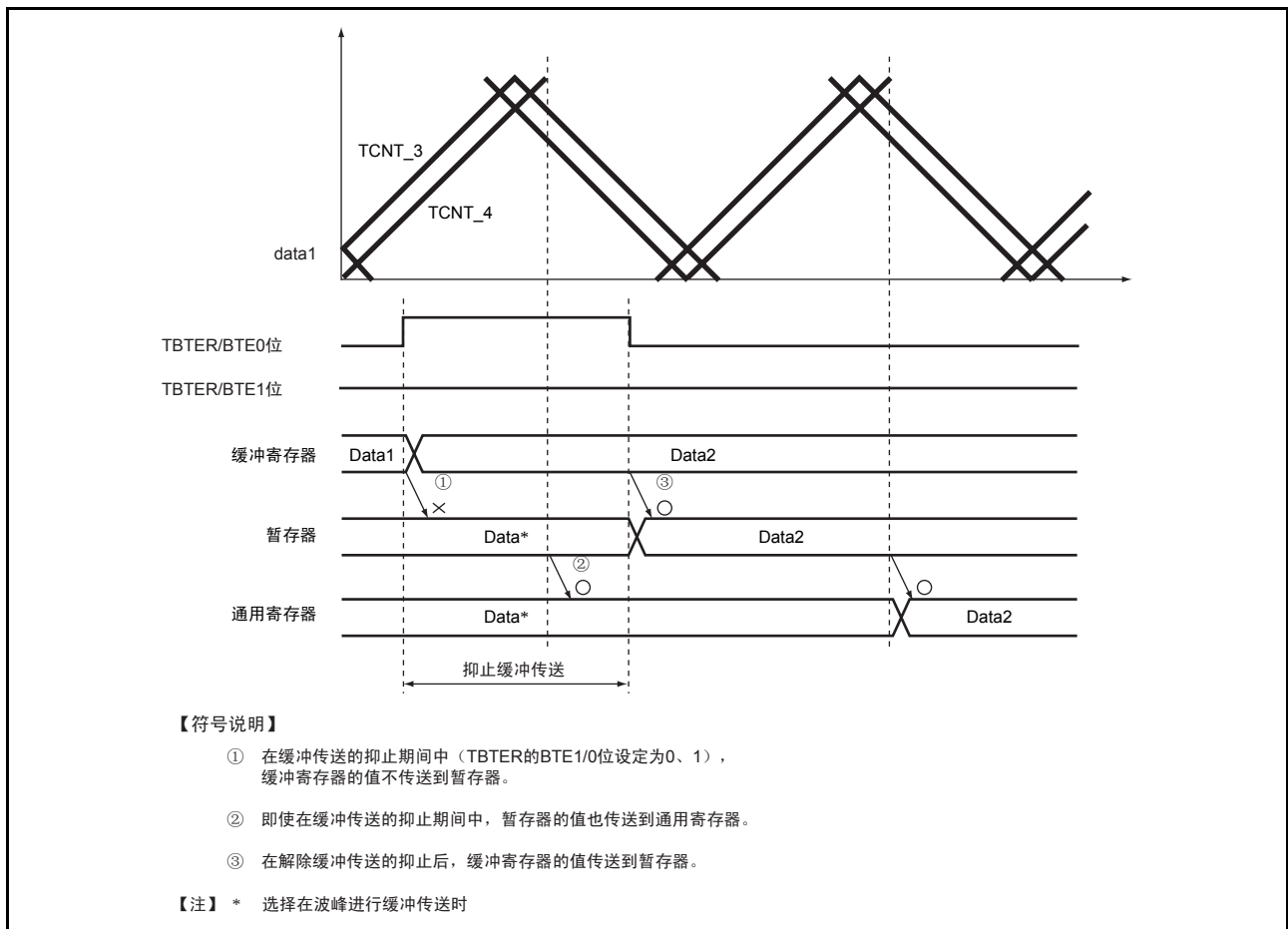


图 9.70 设定为抑制缓冲传送 (BTE1=0、BTE0=1) 时的运行例

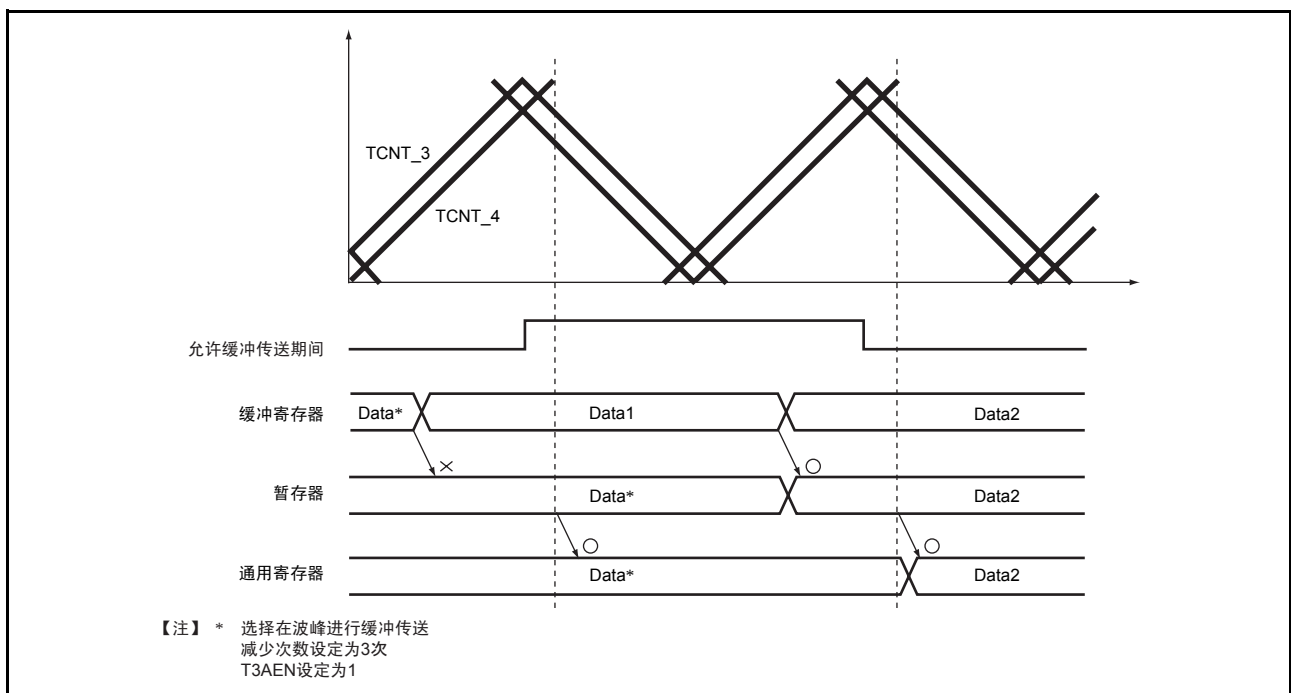


图 9.71 设定为缓冲传送与跳过中断功能联动 (BTE1=1、BTE0=0) 时的运行例

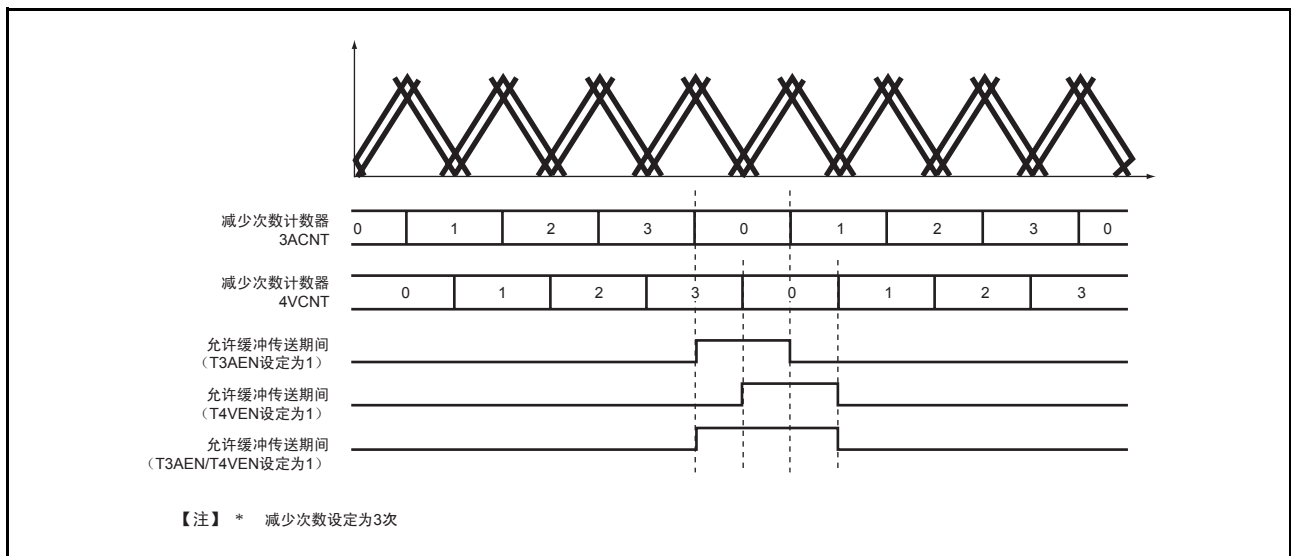


图 9.72 定时器的跳过中断设定寄存器 (TITCR) 的 T3AEN、T4VEN 位的设定和缓冲传送允许期间的关系

(4) 互补 PWM 模式的输出保护功能

互补 PWM 模式的输出具有以下保护功能:

(a) 寄存器和计数器的误写防止功能

在互补 PWM 模式使用的寄存器和计数器中, 除了随时能改写的缓冲寄存器以外, 可通过设定定时器的读写允许寄存器 (TRWER) 的 RWE 位, 允许或者禁止由 CPU 存取模式寄存器、控制寄存器、比较寄存器和计数器。对象寄存器为通道 3 和通道 4 的一部分寄存器, 如下所示:

TCR_3 和 TCR_4、TMDR_3 和 TMDR_4、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

此功能通过将模式寄存器、控制寄存器和计数器等设定为禁止由 CPU 进行存取, 能防止因 CPU 失控而产生的误写。在禁止存取的状态下读对象寄存器时, 读的值不定并且写操作无效。

(b) 由外部信号停止 PWM 输出的功能

能通过输入所指定的外部信号, 将 6 相 PWM 输出管脚自动置为高阻抗状态。

详细内容请参照“第 10 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 输出管脚就自动变为高阻抗状态。但是, 如果时钟又开始振荡, 就不保证管脚的状态。

详细内容请参照“4.7 振荡停止检测功能”。

9.4.9 A/D 转换开始请求的延迟功能

能通过设定通道 4 定时器的 A/D 转换开始请求的控制寄存器 (TADCR)、定时器的 A/D 转换开始请求的周期寄存器 (TADCORA_4 和 TADCORB_4) 和定时器的 A/D 转换开始请求的周期缓冲寄存器 (TADCOBRA_4 和 TADCOBRB_4)，进行 A/D 转换的开始请求。

TCNT_4 和 TADCORA_4、TADCORB_4 比较，当 TCNT_4 和 TADCORA_4、TADCORB_4 一致时，A/D 转换开始请求的延迟功能就进行各自的 A/D 转换开始请求 (TRG4AN 和 TRG4BN)。

另外，通过设定 TADCR 的 ITA3AE、ITA4VE、ITB3AE 和 ITB4VE 位，能与跳过中断功能联动来减少 A/D 转换的开始请求 (TRG4AN 和 TRG4BN)。

(a) A/D 转换开始请求的延迟功能的设定步骤例子

A/D 转换开始请求的延迟功能的设定步骤例子如图 9.73 所示。

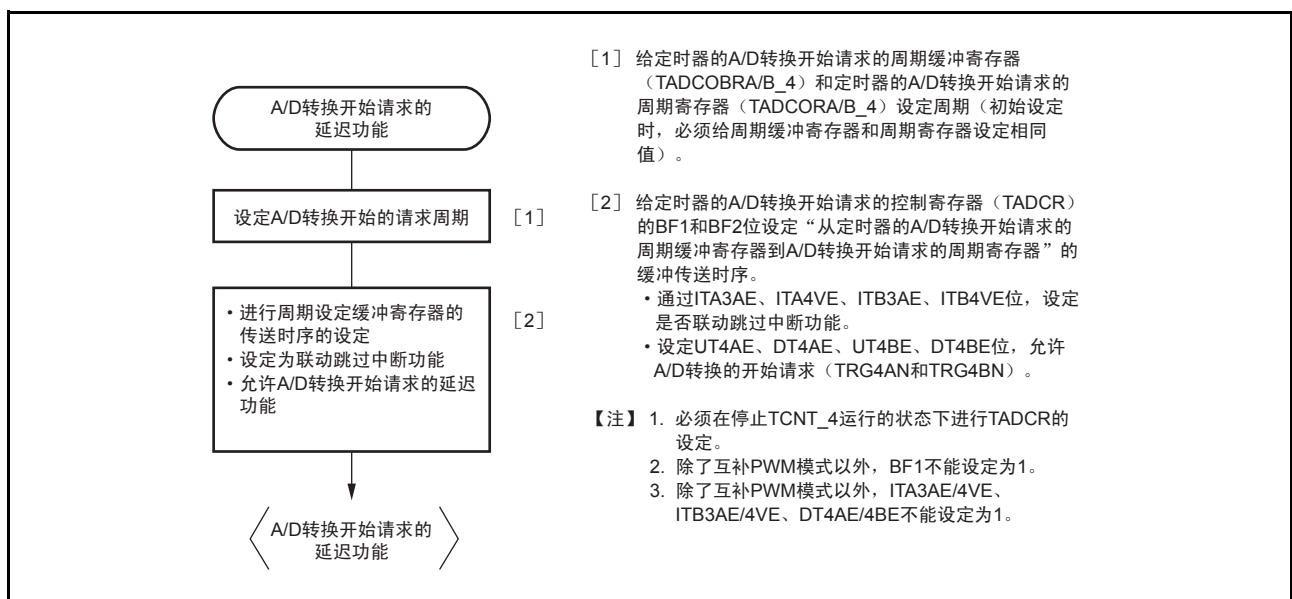


图 9.73 A/D 转换开始请求的延迟功能的设定步骤例子

(b) A/D 转换开始请求的延迟功能的基本运行例

将缓冲传送时序设定为 TCNT_4 的波谷，并且在 TCNT_4 进行递减计数时输出 A/D 转换的开始请求信号 (TRG4AN) 的情况下，A/D 转换的开始请求信号 (TRG4AN) 的基本运行例子如图 9.74 所示。

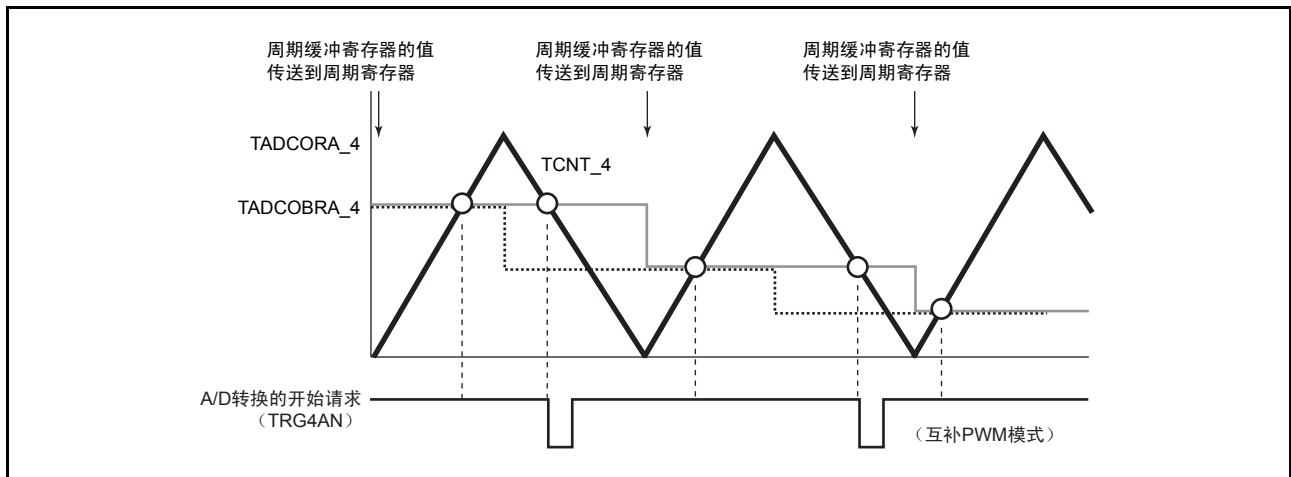


图 9.74 A/D 转换的开始请求信号 (TRG4AN) 的基本运行例

(c) 缓冲传送

通过将数据写到定时器的 A/D 转换开始请求的周期设定缓冲寄存器 (TADCOBRA/B_4)，更新定时器的 A/D 转换开始请求的周期设定寄存器 (TADCORA/B_4) 的数据。能通过设定定时器的 A/D 转换开始请求的控制寄存器 (TADCR_4) 的 BF1、BF0 位，选择从定时器的 A/D 转换开始请求的周期设定缓冲寄存器到定时器的 A/D 转换开始请求的周期设定寄存器的传送时序。

(d) 和跳过中断功能联动的 A/D 转换开始请求的延迟功能

通过设定定时器的 A/D 转换开始请求的控制寄存器 (TADCR) 的 ITA3AE、ITA4VE、ITB3AE 和 ITB4VE 位，能与跳过中断功能联动进行 A/D 转换的开始请求 (TRG4AN 和 TRG4BN)。

在 TCNT_4 进行递增计数或者递减计数时允许 TRG4AN 输出，并且在与跳过中断功能联动的情况下，A/D 转换的开始请求信号 (TRG4AN) 的运行例子如图 9.75 所示。

另外，在 TCNT_4 进行递增计数时允许 TRG4AN 输出，并且在与跳过中断功能联动的情况下，A/D 转换的开始请求信号 (TRG4AN) 的运行例子如图 9.76 所示。

【注】 本功能必须和跳过中断功能配合使用。

在禁止跳过中断功能时 (将定时器的跳过中断设定寄存器 (TITCR) 的 T3AEN、T4VEN 位或者 TITCR 减少次数的设定位 (3ACOR 和 4VCOR) 置 0 时)，必须设定为不与跳过中断功能联动 (将定时器的 A/D 转换开始请求的控制寄存器 (TADCR) 的 ITA3AE、ITA4VE、ITB3AE 和 ITB4VE 位设定为 0)。

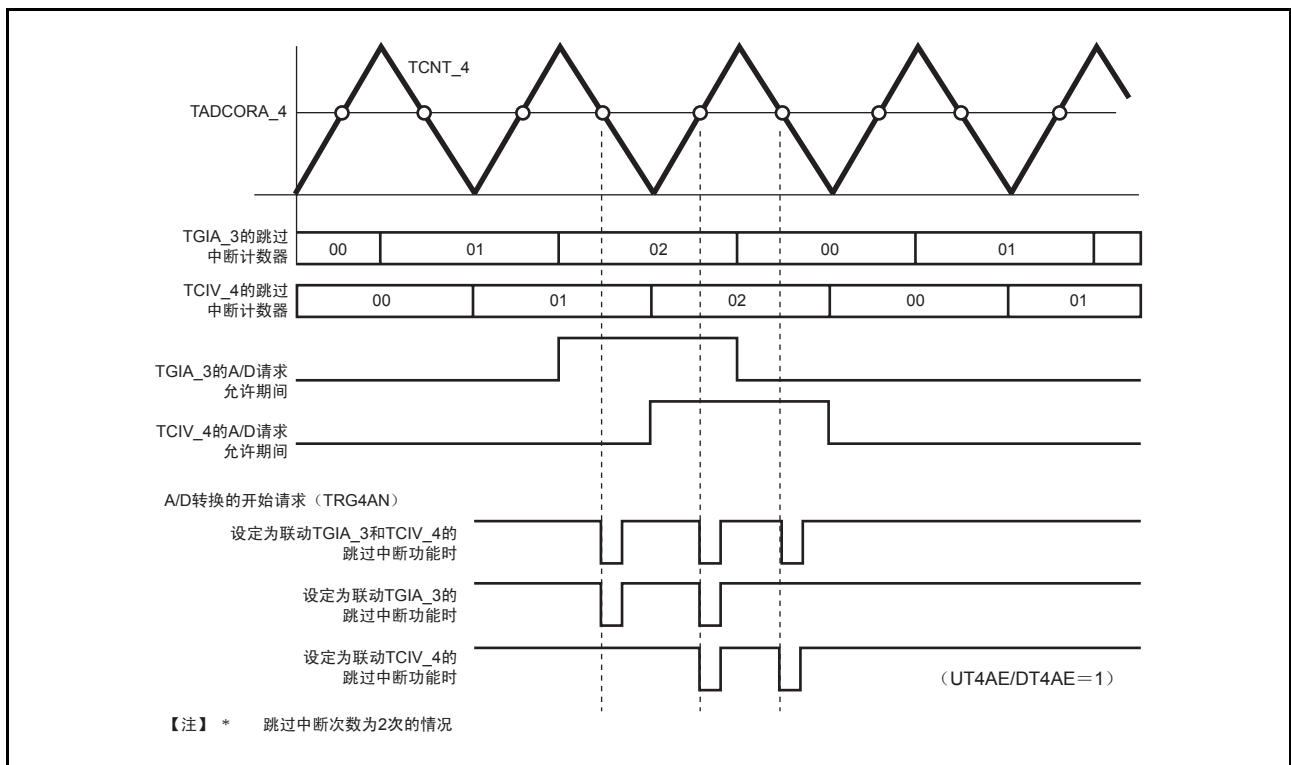


图 9.75 与跳过中断功能联动时 A/D 转换的开始请求信号 (TRG4AN) 的运行例

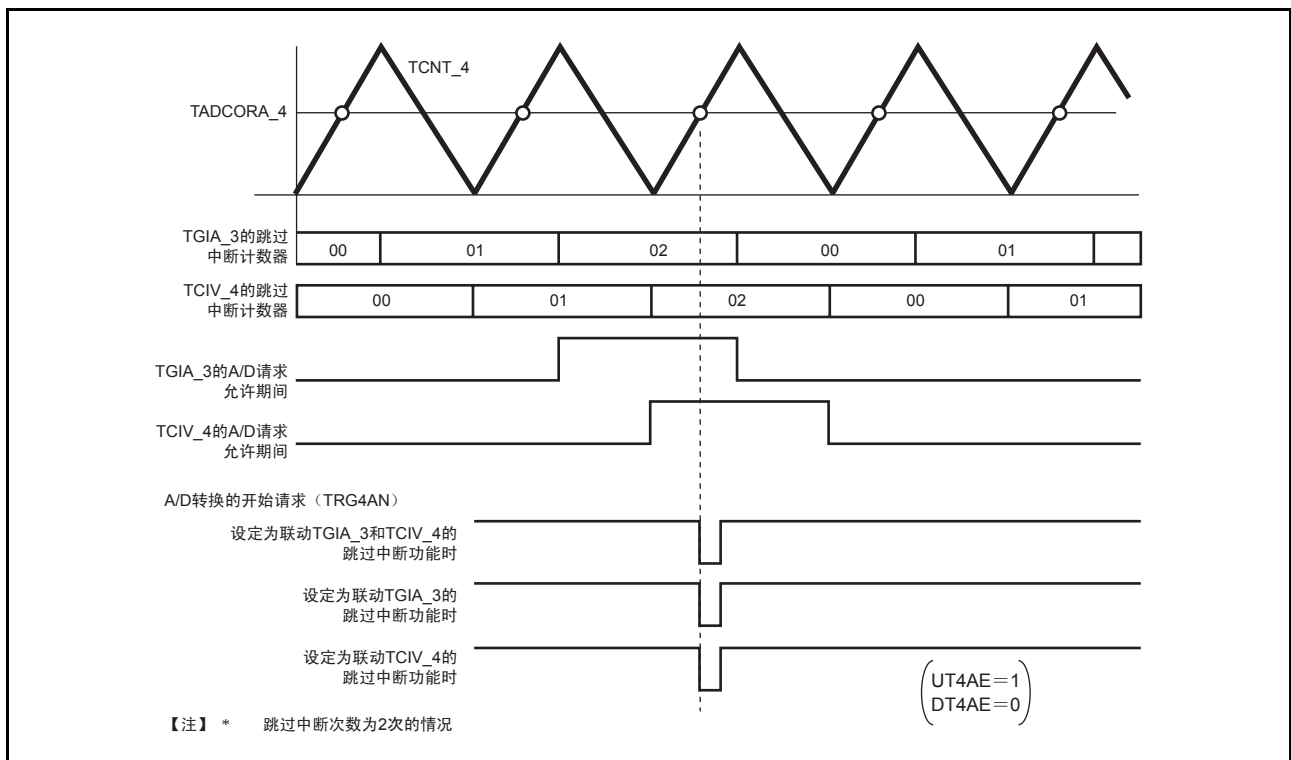


图 9.76 与跳过中断功能联动时 A/D 转换的开始请求信号 (TRG4AN) 的运行例

9.4.10 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

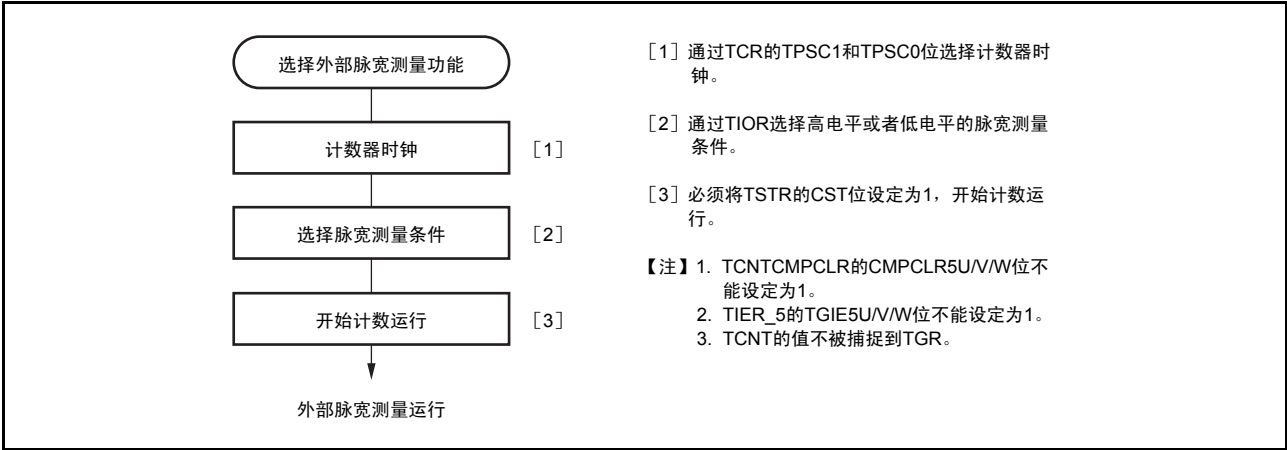


图 9.77 测量外部脉宽的设定步骤例子

(2) 测量外部脉宽的运行例

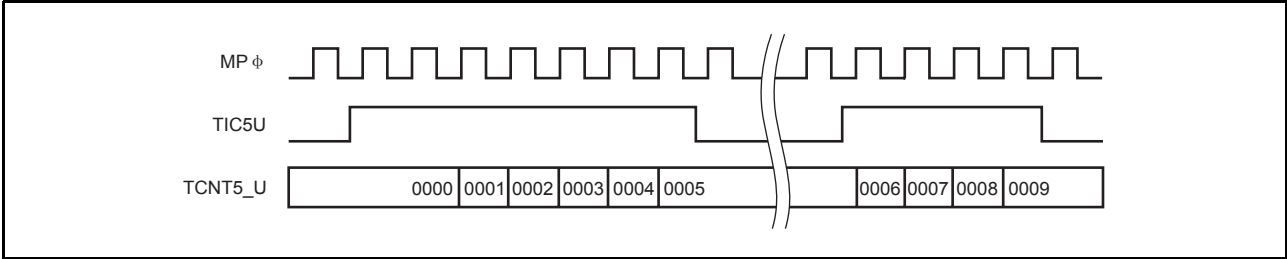


图 9.78 测量外部脉宽的运行例（测量 High 电平的脉宽）

9.4.11 空载时间的补偿功能

通过测量输出波形的延迟并将此延迟反映到占空比，能将外部脉宽的测量功能用作互补 PWM 运行时的空载时间补偿功能。

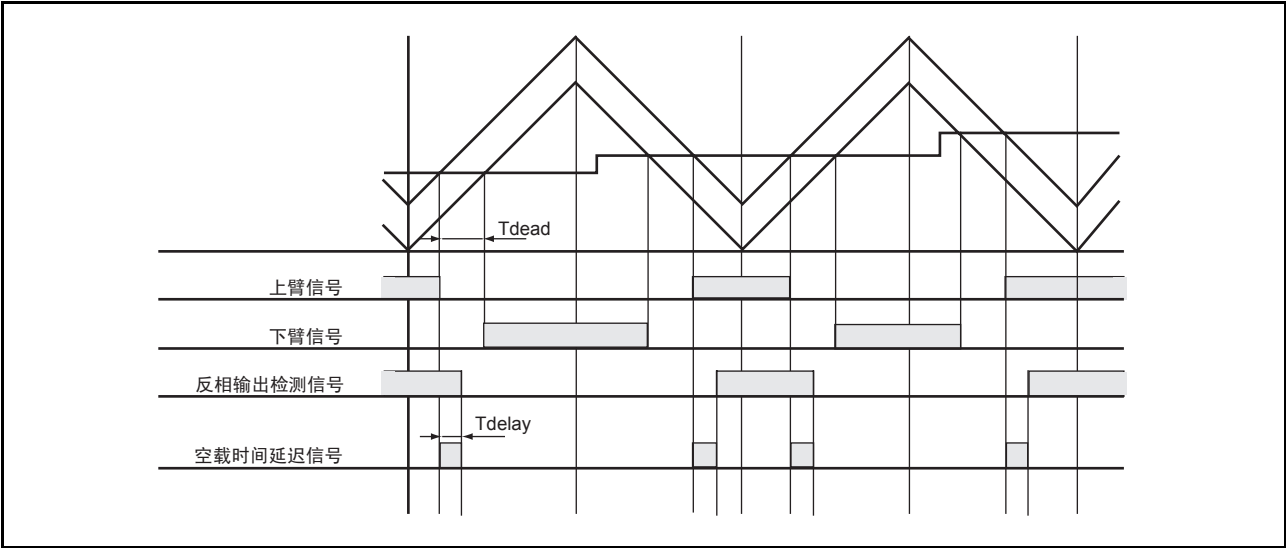


图 9.79 互补 PWM 运行时的空载时间延迟

(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 9.80 所示。

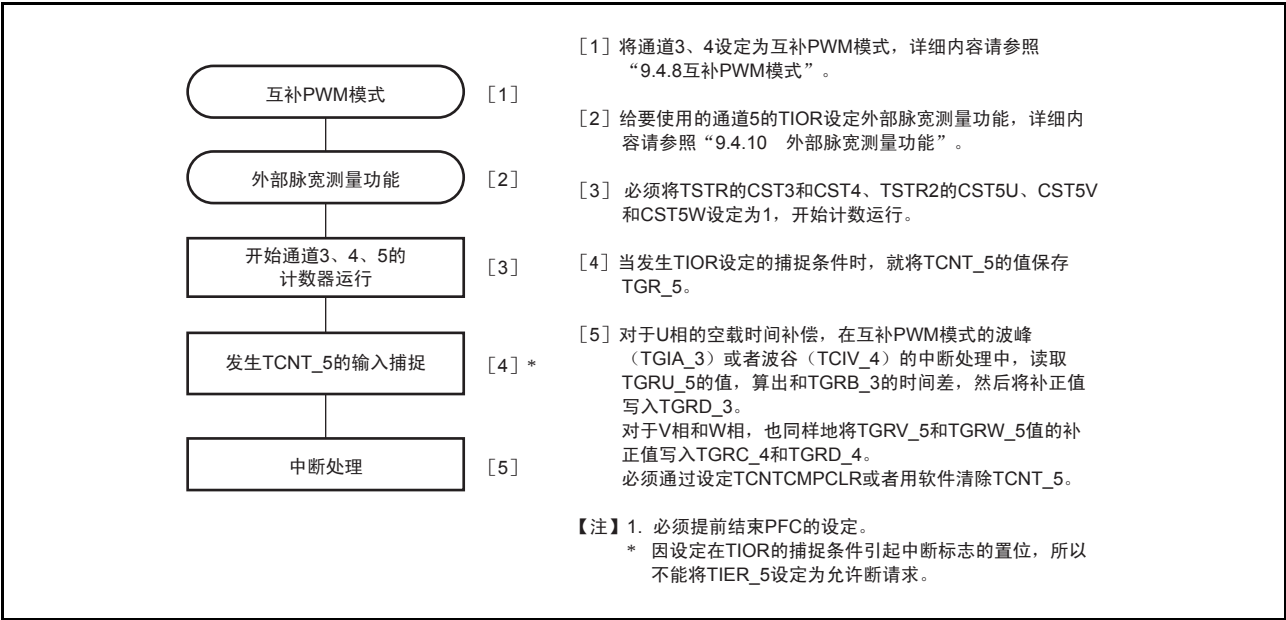


图 9.80 空载时间补偿功能的设定步骤例子

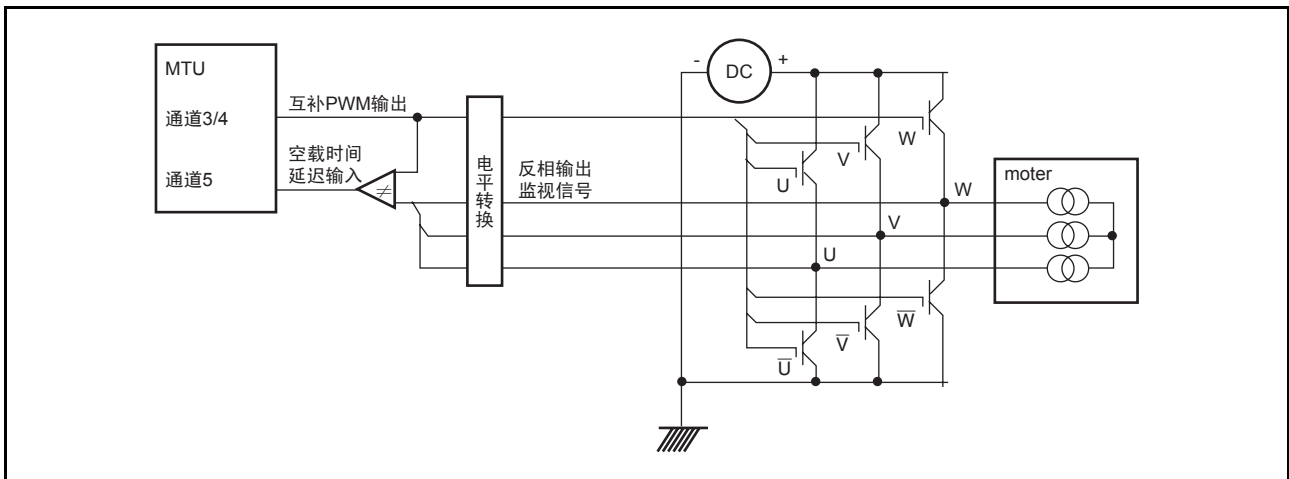


图 9.81 构成马达控制电路的例子

9.4.12 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 9.82 所示。

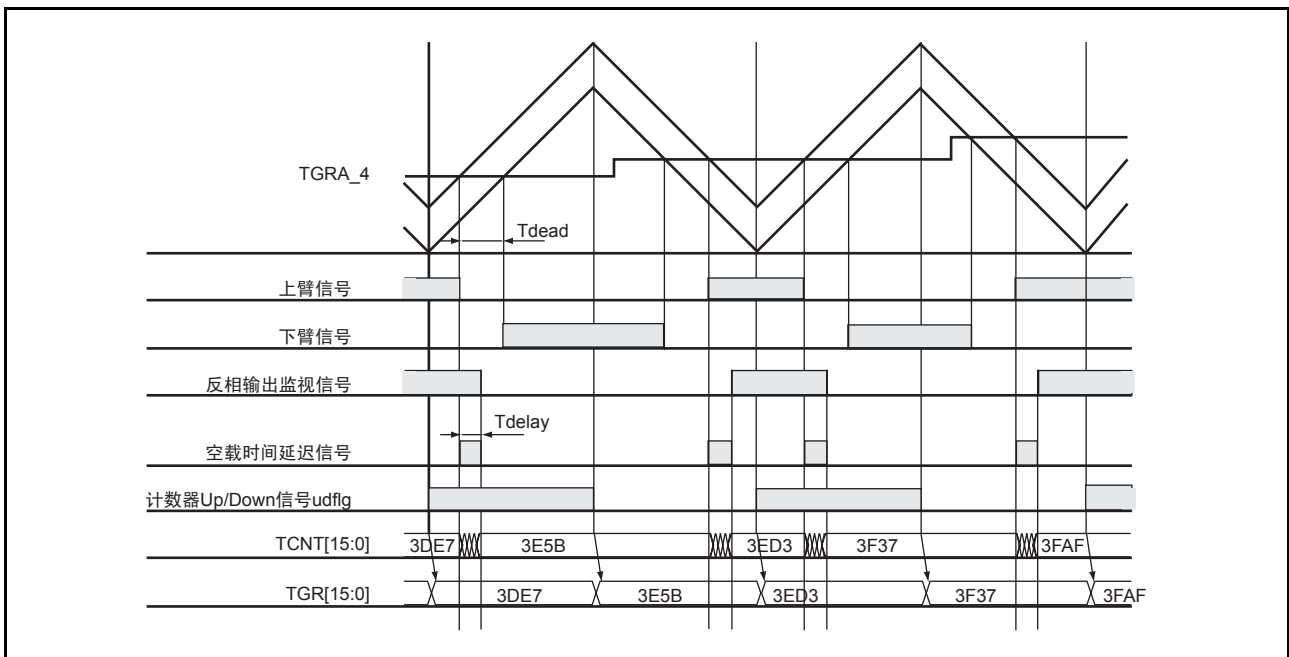


图 9.82 互补 PWM 在“波峰 / 波谷”的 TCNT 捕捉运行

(1) 输入捕捉 / 比较匹配的中断

在通过各通道的 TGR 输入捕捉 / 比较匹配将 TSR 的 TGF 标志置 1 时, 如果 TIER 的 TGIE 位已被置 1, 就请求中断。通过将 TGF 标志清 0 解除中断请求。在 MTU2 中, 通道 0 有 6 个, 通道 3、4 各有 4 个, 通道 1、2 各有 2 个, 通道 5 各有 3 个, 共 21 个输入捕捉 / 比较匹配的中断。通道 0 的 TGFE_0 和 TGFF_0 标志在输入捕捉时不被置位。

(2) 上溢中断

在通过各通道的 TCNT 上溢将 TSR 的 TCFV 标志置 1 时, 如果 TIER 的 TCIEV 位已被置 1, 就请求中断。通过将 TCFV 标志清 0 解除中断请求。在 MTU2 中, 各通道有 1 个, 共 5 个上溢中断。

(3) 下溢中断

在通过各通道的 TCNT 下溢将 TSR 的 TCFU 标志置 1 时, 如果 TIER 的 TCIEU 位已被置 1, 就请求中断。通过将 TCFU 标志清 0 解除中断请求。在 MTU2 中, 通道 1、2 各有 1 个, 共 2 个下溢中断。

9.5.2 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 9.58 所示。

(1) TGRA 的输入捕捉 / 比较匹配和在互补 PWM 模式中的 TCNT_4 波谷启动 A/D 转换器

能通过各通道的 TGRA 输入捕捉 / 比较匹配启动 A/D 转换器。另外, 在给 TIER_4 的 TTGE2 位设定 1 的状态下, 如果互补 PWM 运行, 即使在 TCNT_4 为波谷 (TCNT_4=H'0000) 时也能启动 A/D 转换器。

在以下所示的条件中, 对 A/D 转换器产生 A/D 转换的开始请求 TRGAN。

- 在通过各通道的 TGRA 输入捕捉 / 比较匹配将 TSR 的 TGFA 标志置 1 时, TIER 的 TTGE 位已被置 1
- 在给 TIER_4 的 TTGE2 位设定 1 的状态下, 当互补 PWM 运行并且 TCNT_4 为波谷 (TCNT_4=H'0000) 时

此时, 如果已在 A/D 转换器侧选择 MTU2 的转换开始触发 TRGAN, 就开始进行 A/D 转换。

(2) 通过 TCNT_0 和 TGRE_0 的比较匹配启动 A/D 转换器

通过通道 0 的 TCNT_0 和 TGRE_0 的比较匹配, 能产生 A/D 转换的开始请求 TRG0N, 启动 A/D 转换器。

在通过通道 0 的 TCNT_0 和 TGRE_0 的比较匹配将 TSR2_0 的 TGFE 标志置 1 时, 如果 TIER2_0 的 TTGE2 位已被置 1, 就对 A/D 转换器产生 A/D 转换的开始请求 TRG0N。此时, 如果已在 A/D 转换器侧选择 MTU2 的转换开始触发 TRG0N, 就开始进行 A/D 转换。

(3) 通过 A/D 转换开始请求的延迟功能启动 A/D 转换器

如果给 A/D 转换开始请求的控制寄存器 (TADCR) 的 TAD4AE、TAD4BE 位设定 1, 能通过 TADCORA、TADCORB 和 TCNT_4 的一致产生 TRG4AN 和 TRG4BN, 启动 A/D 转换器。详细内容请参照“9.4.9 A/D 转换开始请求的延迟功能”。

当发生 TRG4AN 时, 如果已在 A/D 转换器侧选择 MTU2 的转换开始触发 TRG4AN, 就开始进行 A/D 转换; 当发生 TRG4BN 时, 如果已在 A/D 转换器侧选择 MTU2 的转换开始触发 TRG4BN, 就开始进行 A/D 转换。

表 9.58 各中断源和 A/D 转换开始请求的对应

对象	中断源	A/D 转换的开始请求
TGRA_0 和 TCNT_0	输入捕捉 / 比较匹配	TRGAN
TGRA_1 和 TCNT_1		
TGRA_2 和 TCNT_2		
TGRA_3 和 TCNT_3		
TGRA_4 和 TCNT_4		
TCNT_4	互补 PWM 模式中 TCNT_4 的波谷	
TGRE_0 和 TCNT_0	比较匹配	TRG0N
TADCORA 和 TCNT_4		TRG4AN
TADCORB 和 TCNT_4		TRG4BN

9.6 运行时序

9.6.1 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 9.83 和图 9.84、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 9.85 和图 9.86 所示。

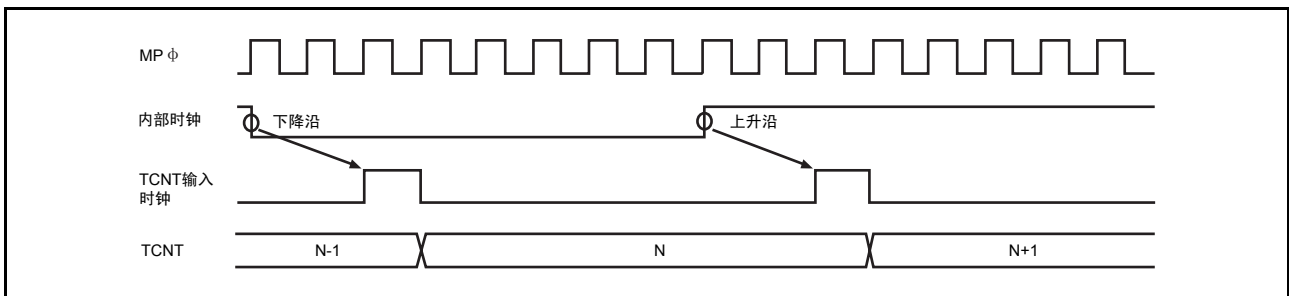


图 9.83 内部时钟运行时的计数时序（通道 0 ~ 4）

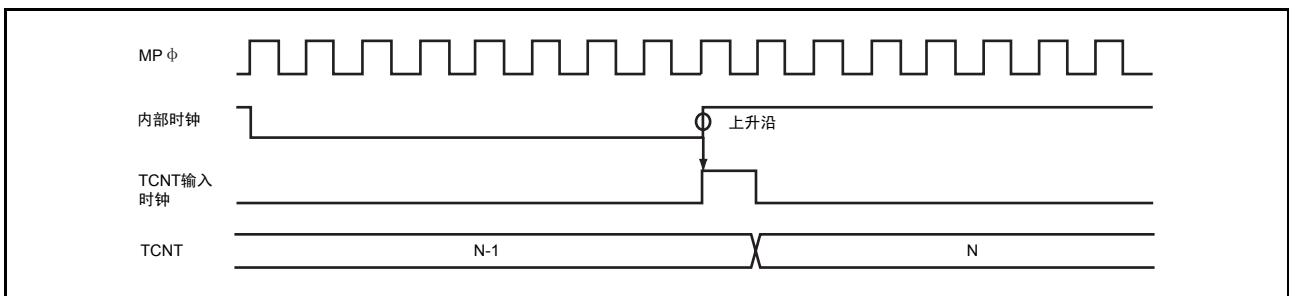


图 9.84 内部时钟运行时的计数时序（通道 5）

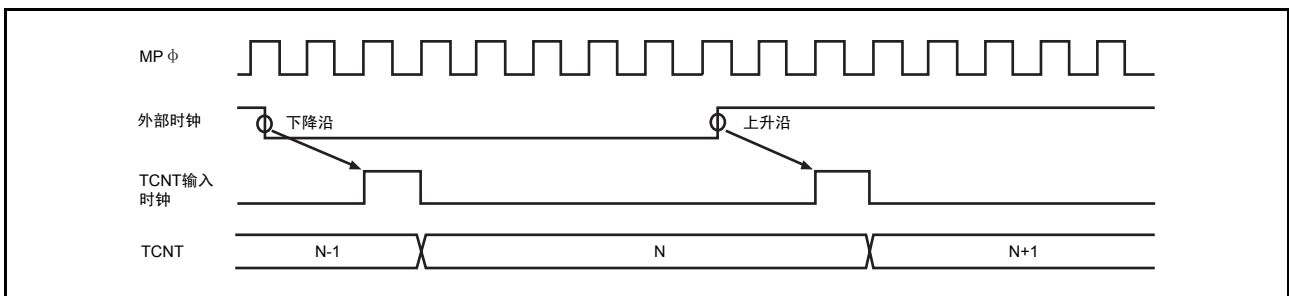


图 9.85 外部时钟运行时的计数时序（通道 0 ~ 4）

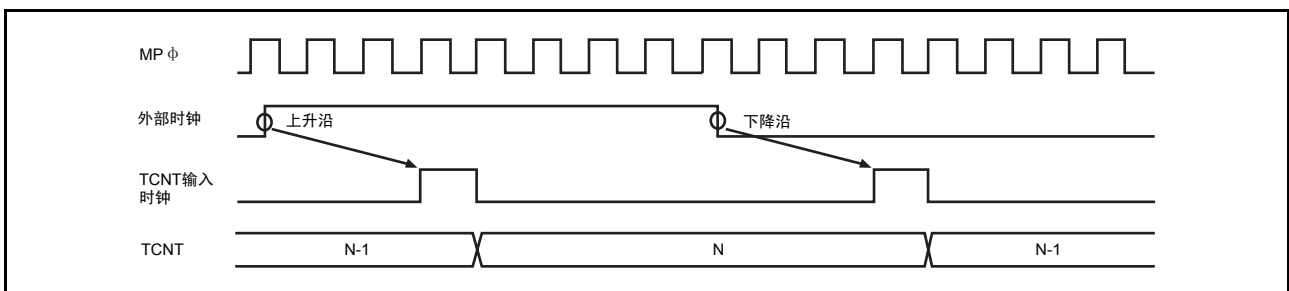


图 9.86 外部时钟运行时的计数时序（相位计数模式）

(2) 输出比较的输出时序

在 TCNT 和 TGR 一致的最后状态（更新 TCNT 一致后的计数值时）产生比较匹配信号。在产生比较匹配信号时，将 TIOR 设定的输出值输出到输出比较的输出管脚（TIOC 管脚）。在 TCNT 和 TGR 一致后到产生 TCNT 输入时钟为止，不产生比较匹配信号。

输出比较的输出时序（正常模式和 PWM 模式）如图 9.87、输出比较的输出时序（互补 PWM 模式和复位同期 PWM 模式）如图 9.88 所示。

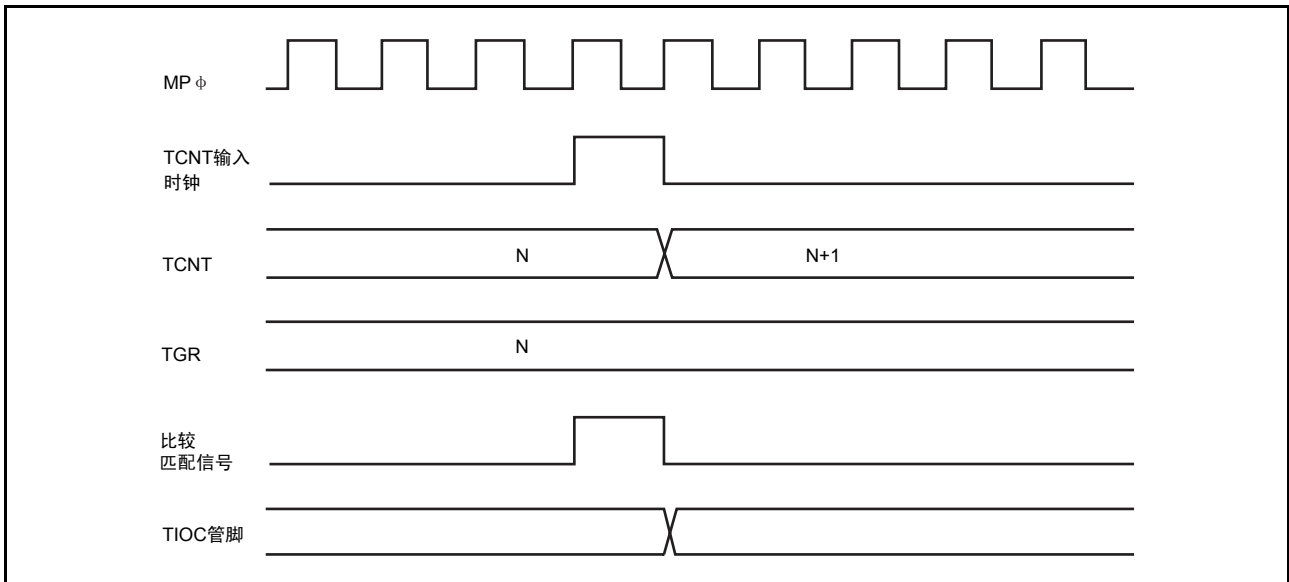


图 9.87 输出比较的输出时序（正常模式和 PWM 模式）

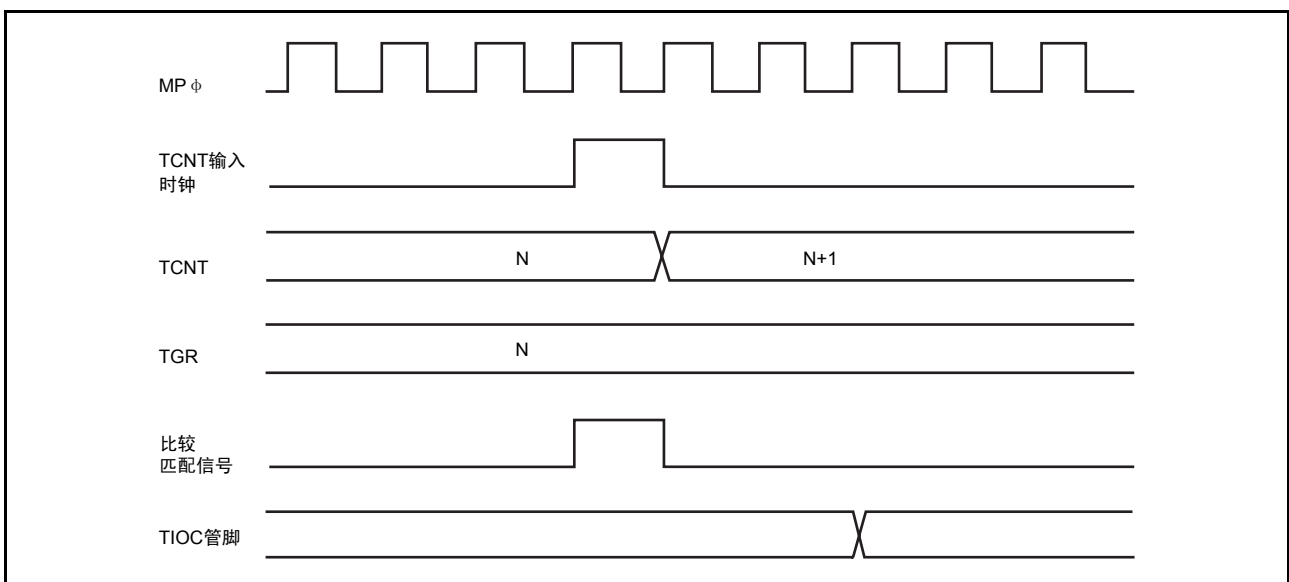


图 9.88 输出比较的输出时序（互补 PWM 模式和复位同期 PWM 模式）

(3) 输入捕捉的输入信号时序

输入捕捉的输入信号时序如图 9.89 所示。

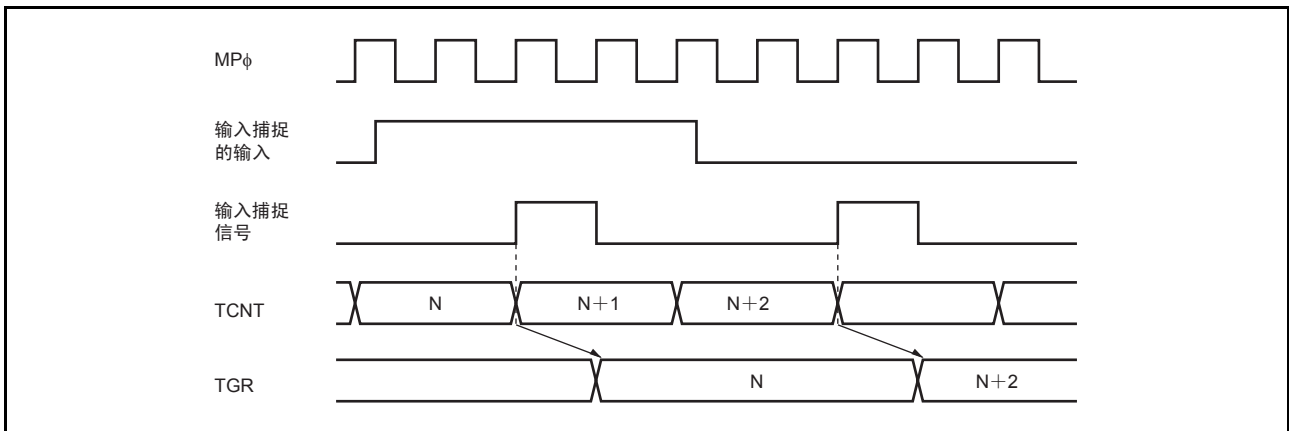


图 9.89 输入捕捉的输入信号时序

(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 9.90 和图 9.91 所示。

指定通过输入捕捉进行计数器清除时的时序如图 9.92 所示。

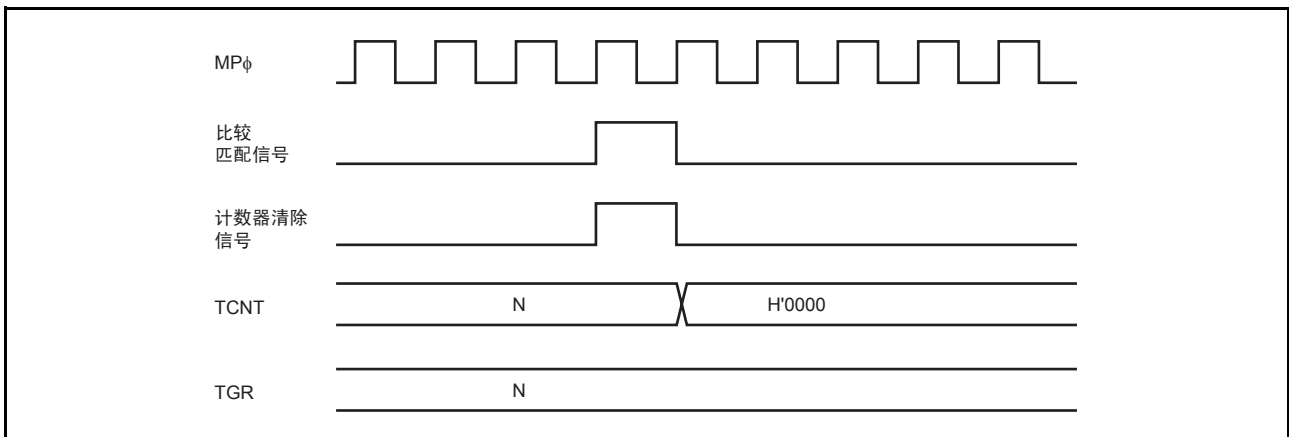


图 9.90 计数器清除的时序 (比较匹配)

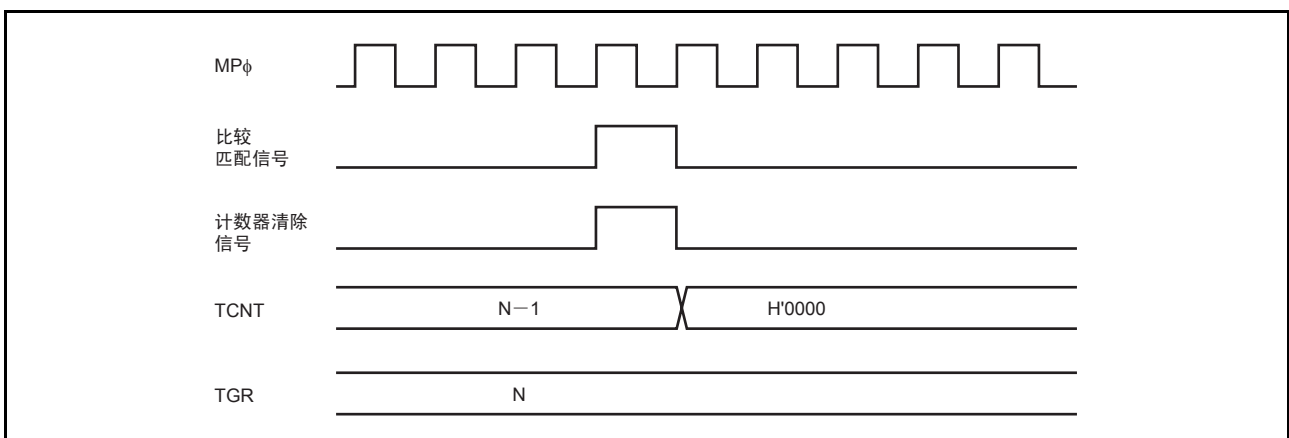


图 9.91 计数器清除的时序 (比较匹配) (通道 5)

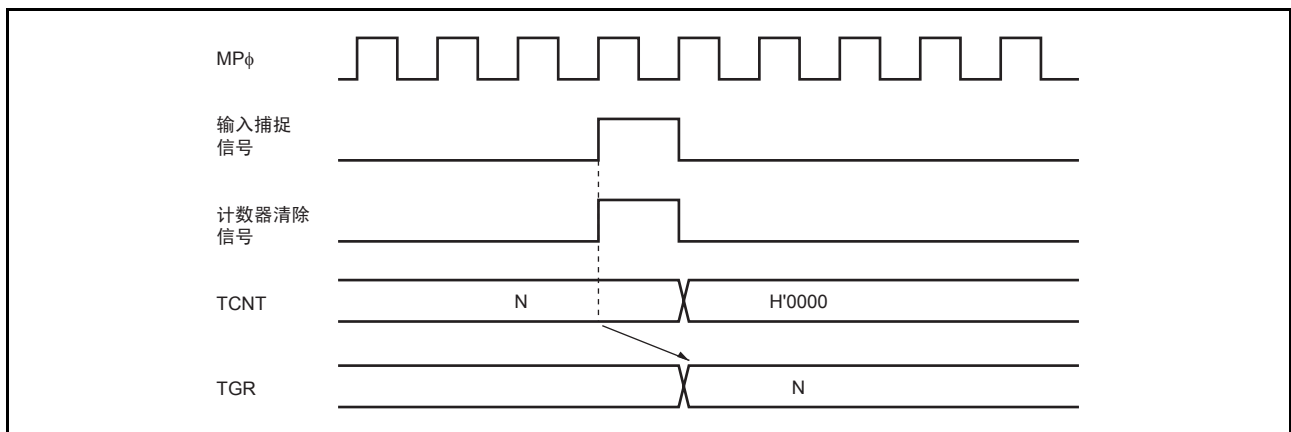


图 9.92 计数器清除的时序 (输入捕捉)

(5) 缓冲运行的时序

缓冲运行的时序如图 9.93 ~ 图 9.95 所示。

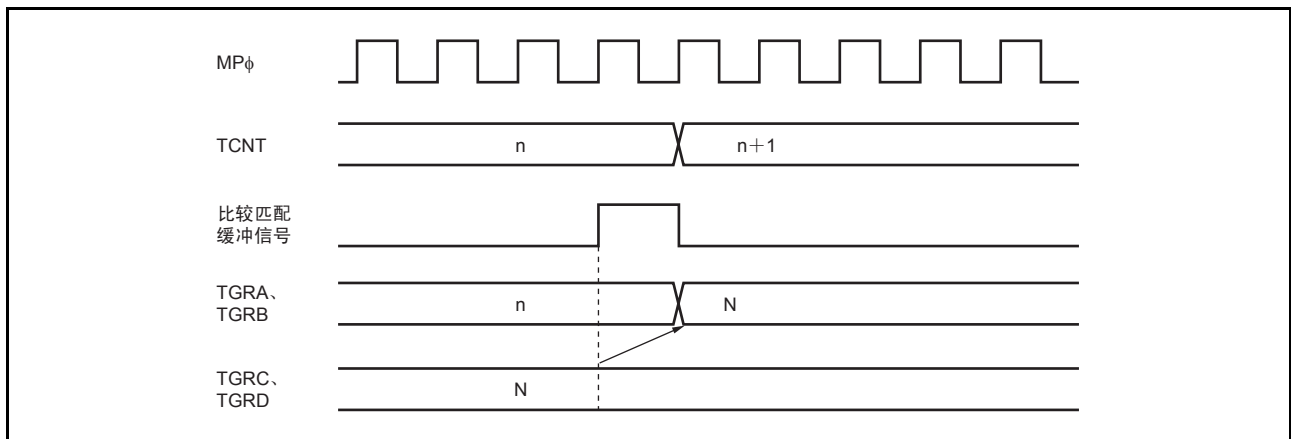


图 9.93 缓冲运行的时序 (比较匹配)

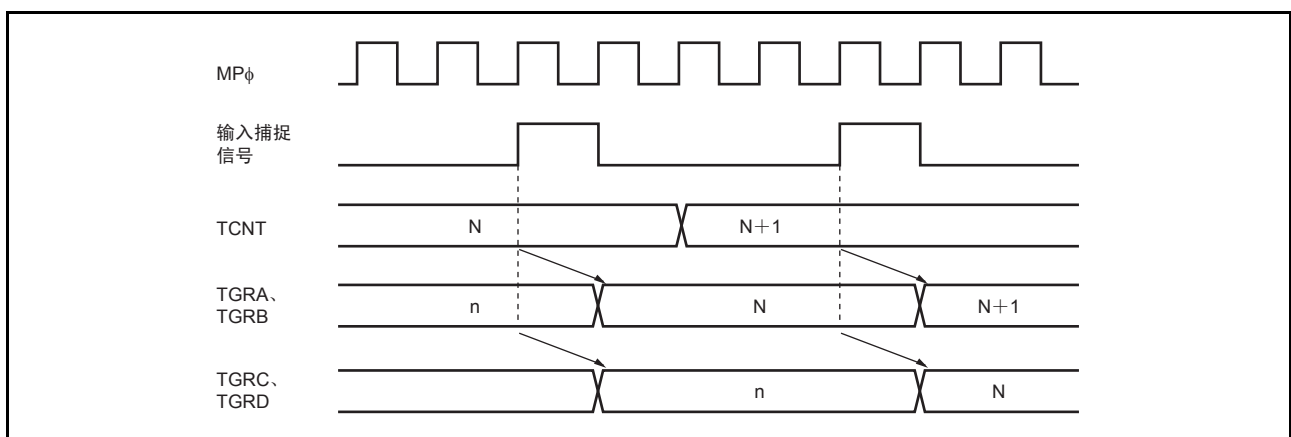


图 9.94 缓冲运行的时序 (输入捕捉)

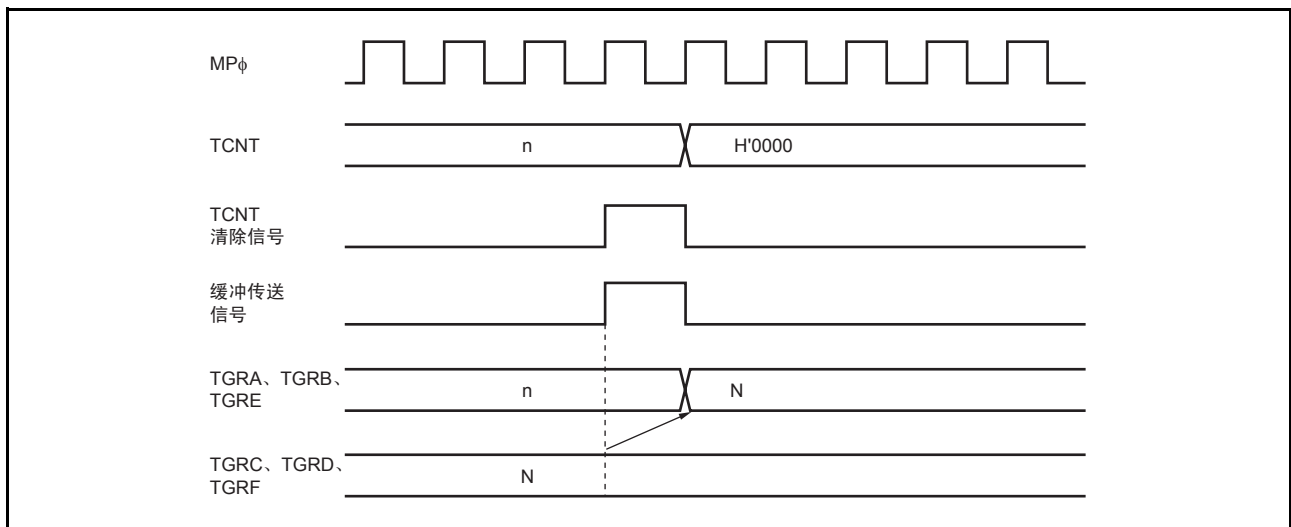


图 9.95 缓冲运行的时序 (清除 TCNT)

(6) 缓冲传送的时序 (互补 PWM 模式)

互补 PWM 模式中缓冲传送的时序如图 9.96 ~ 图 9.98 所示。

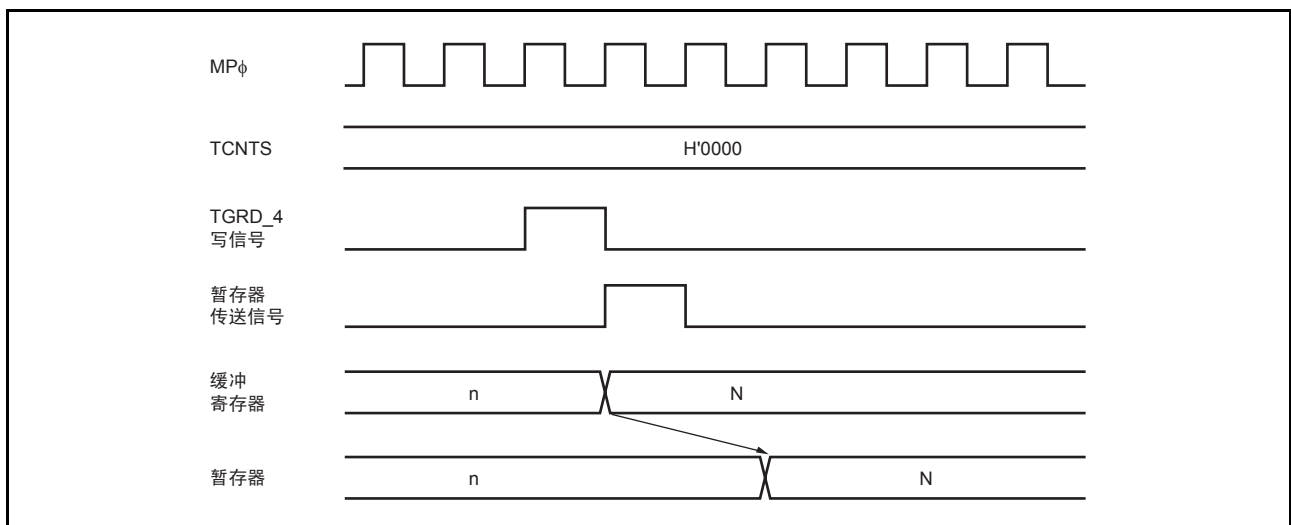


图 9.96 缓冲寄存器到暂存器的传送时序 (TCNTS 停止中)

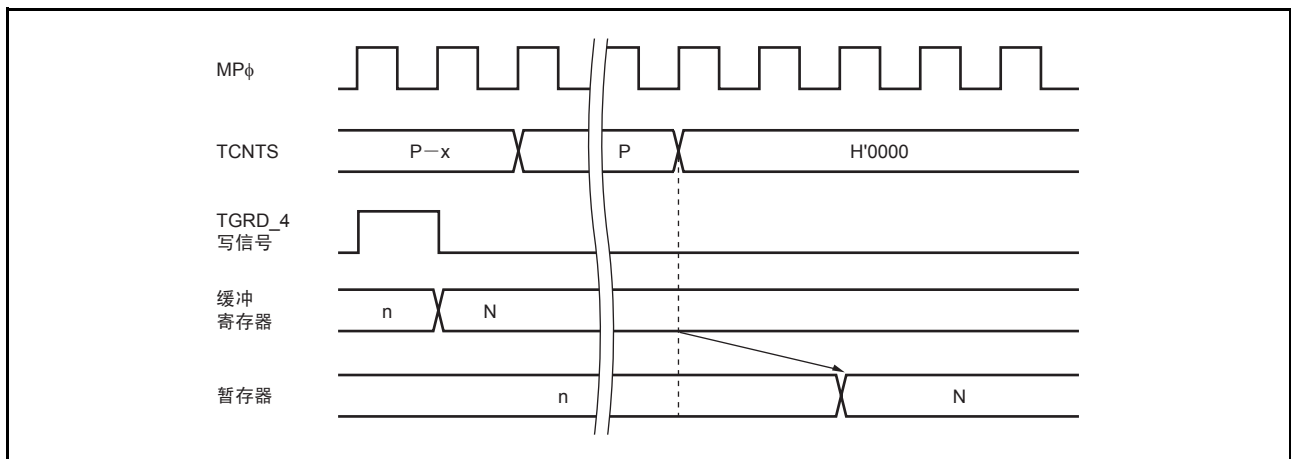


图 9.97 缓冲寄存器到暂存器的传送时序 (TCNTS 运行中)

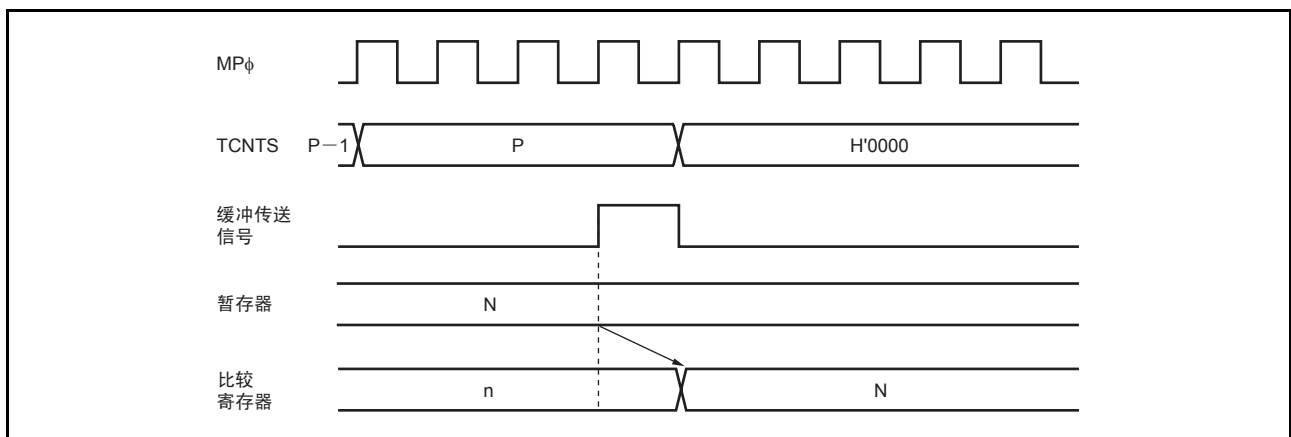


图 9.98 暂存器到比较寄存器的传送时序

9.6.2 中断信号的时序

(1) 比较匹配时的 TGF 标志的置位时序

由比较匹配产生的 TSR 的 TGF 标志置位时序和 TGI 中断请求信号的时序如图 9.99 和图 9.100 所示。

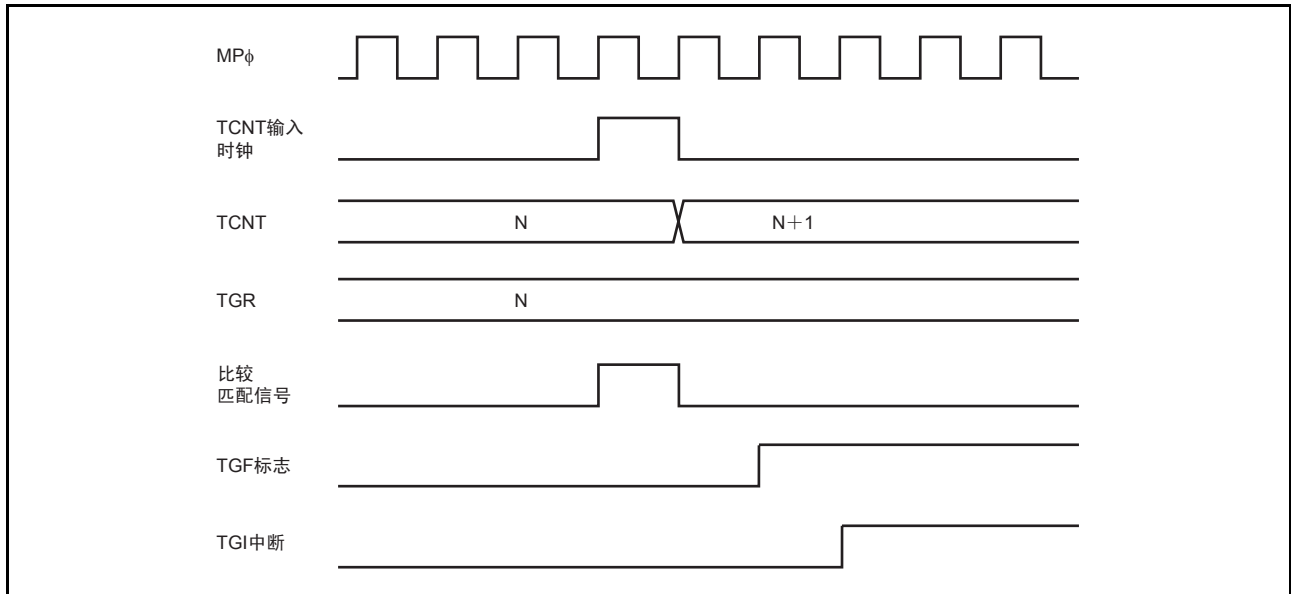


图 9.99 TGI 中断时序（比较匹配）（通道 0 ~ 4）

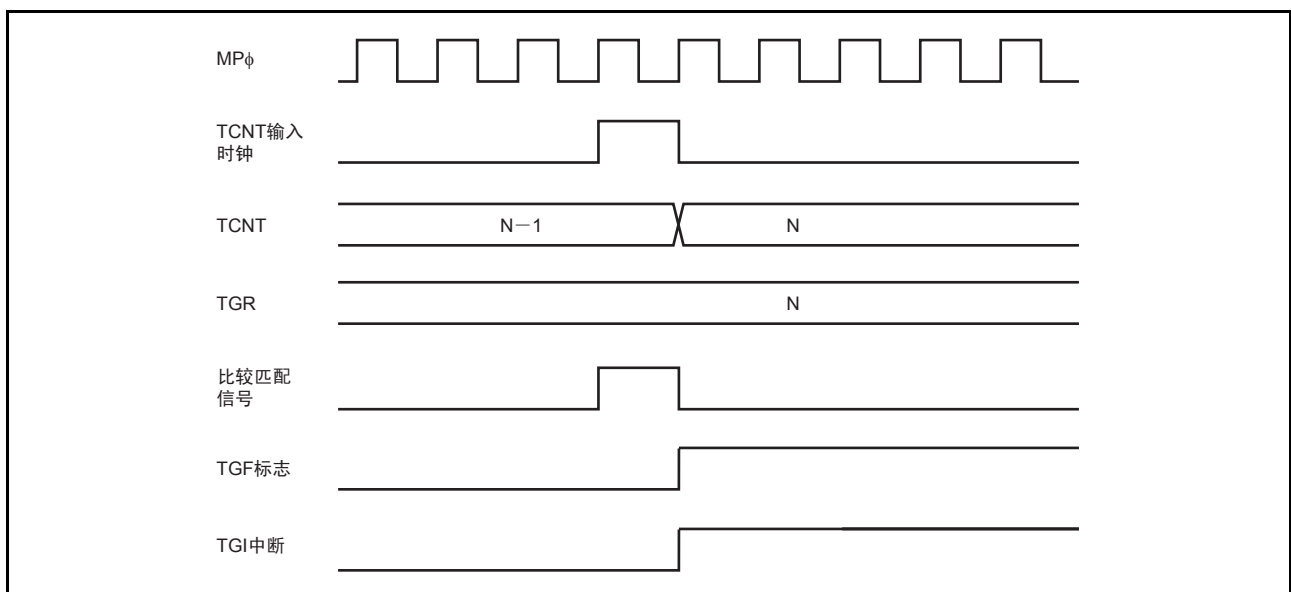


图 9.100 TGI 中断时序（比较匹配）（通道 5）

(2) 输入捕捉时的 TGF 标志的置位时序

由输入捕捉产生 TSR 的 TGF 标志的置位时序和 TGI 中断请求信号的时序如图 9.101 和图 9.102 所示。

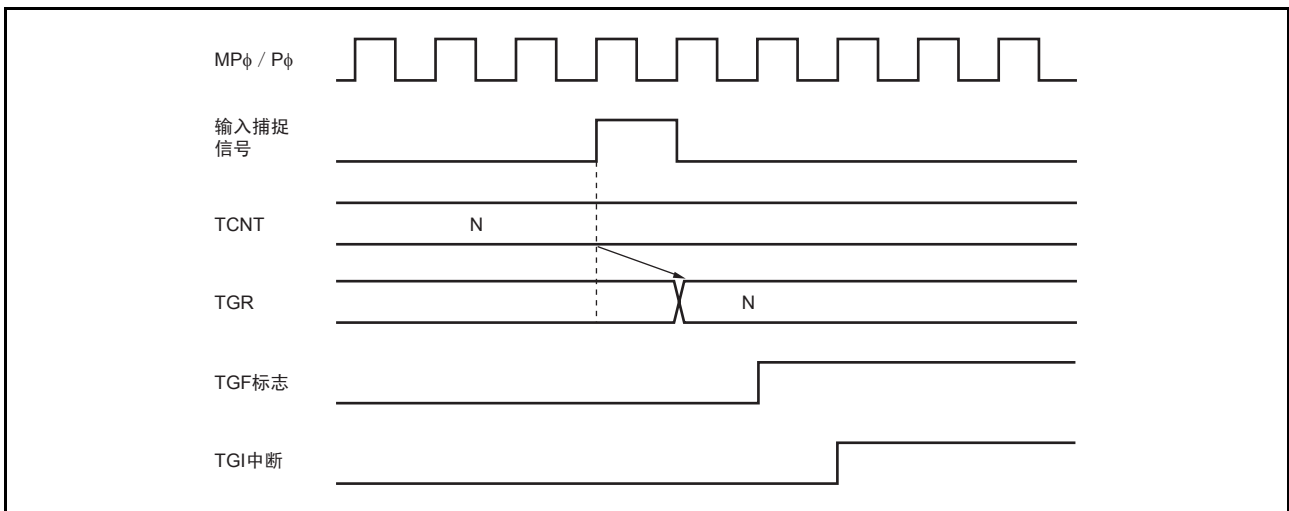


图 9.101 TGI 中断时序 (输入捕捉) (通道 0 ~ 4)

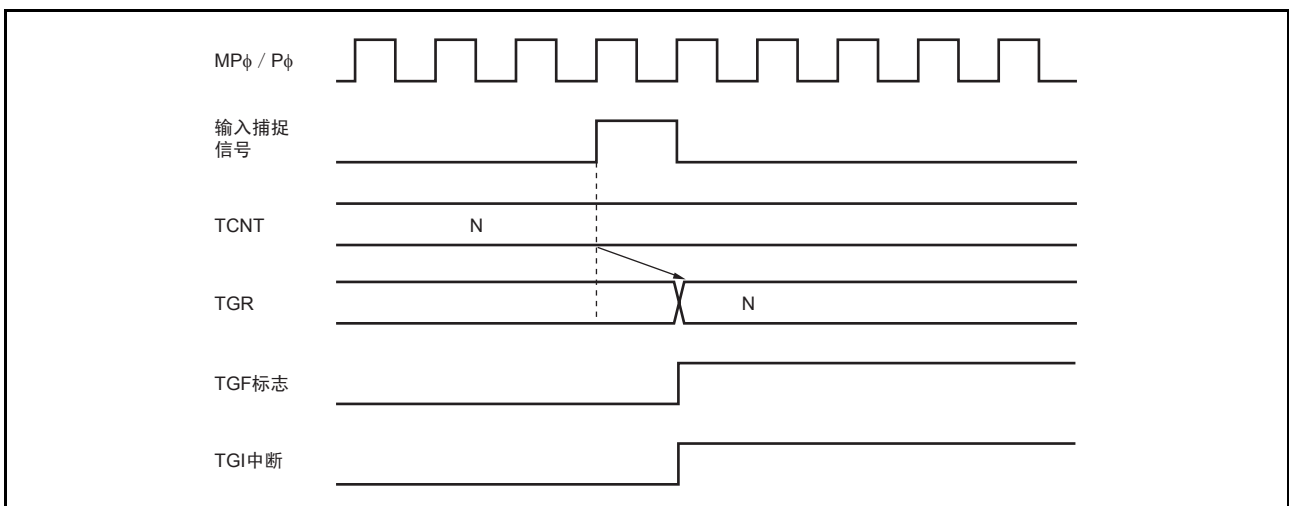


图 9.102 TGI 中断时序 (输入捕捉) (通道 5)

(3) TCFV 标志 / TCFU 标志的置位时序

由上溢产生的 TSR 的 TCFV 标志的置位时序和 TCIV 中断请求信号的时序如图 9.103 所示。

由下溢产生的 TSR 的 TCFU 标志的置位时序和 TCIU 中断请求信号的时序如图 9.104 所示。

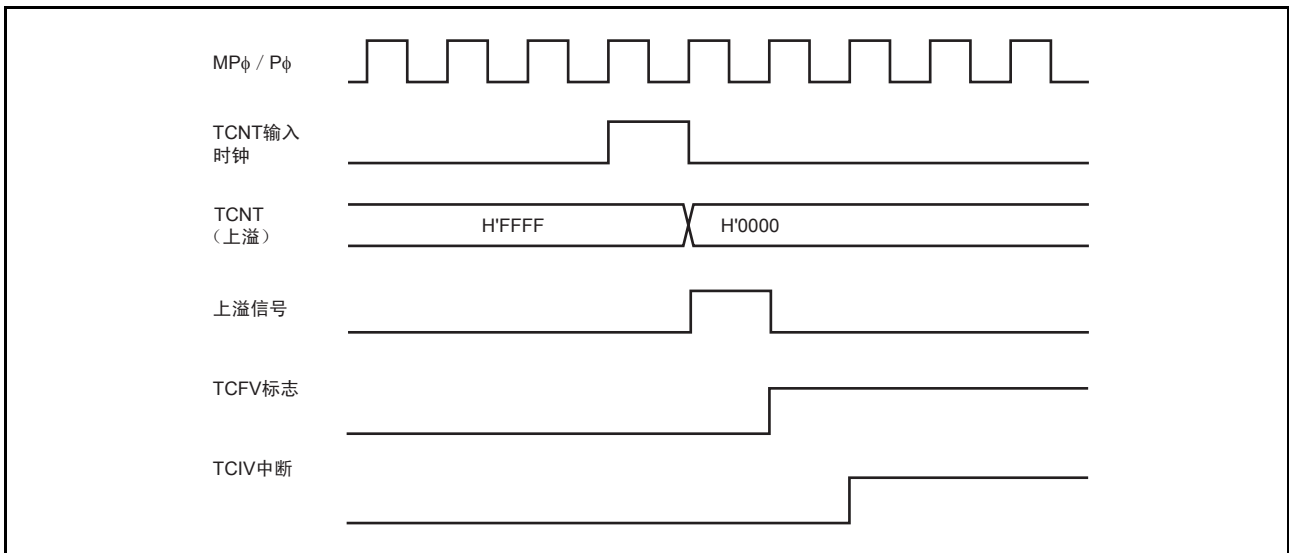


图 9.103 TCIV 中断的置位时序

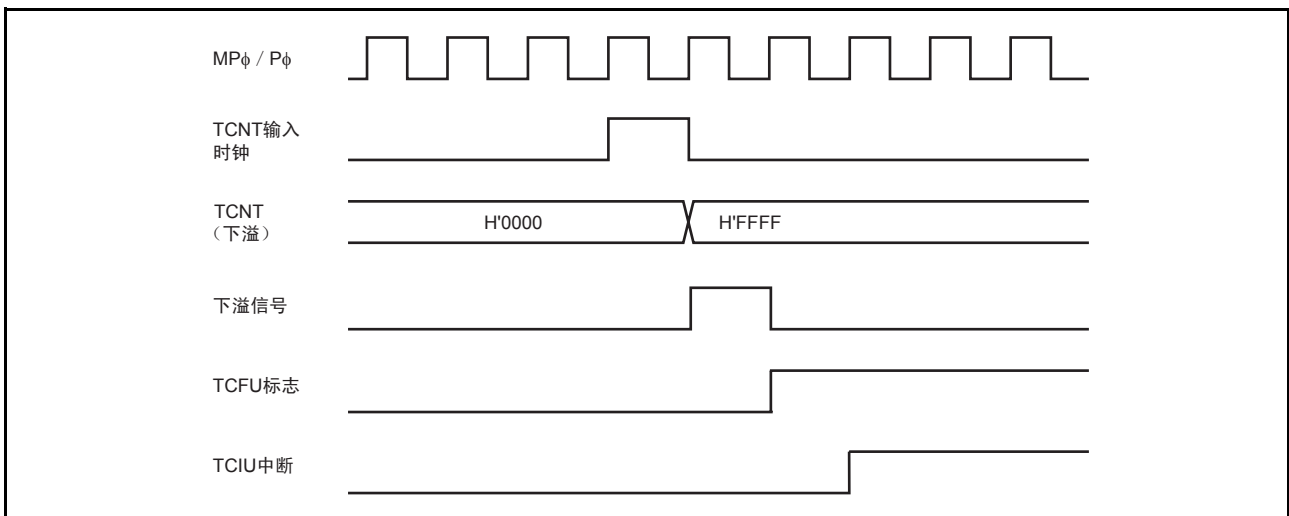


图 9.104 TCIU 中断的置位时序

(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。通过 CPU 清除状态标志的时序如图 9.105 和图 9.106 所示。

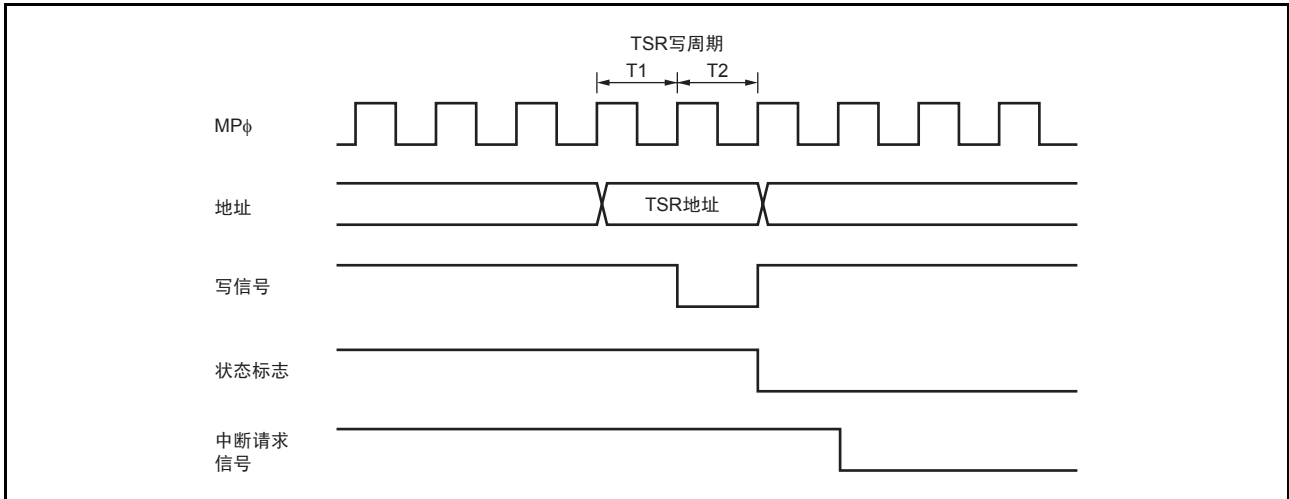


图 9.105 通过 CPU 清除状态标志的时序 (通道 0 ~ 4)

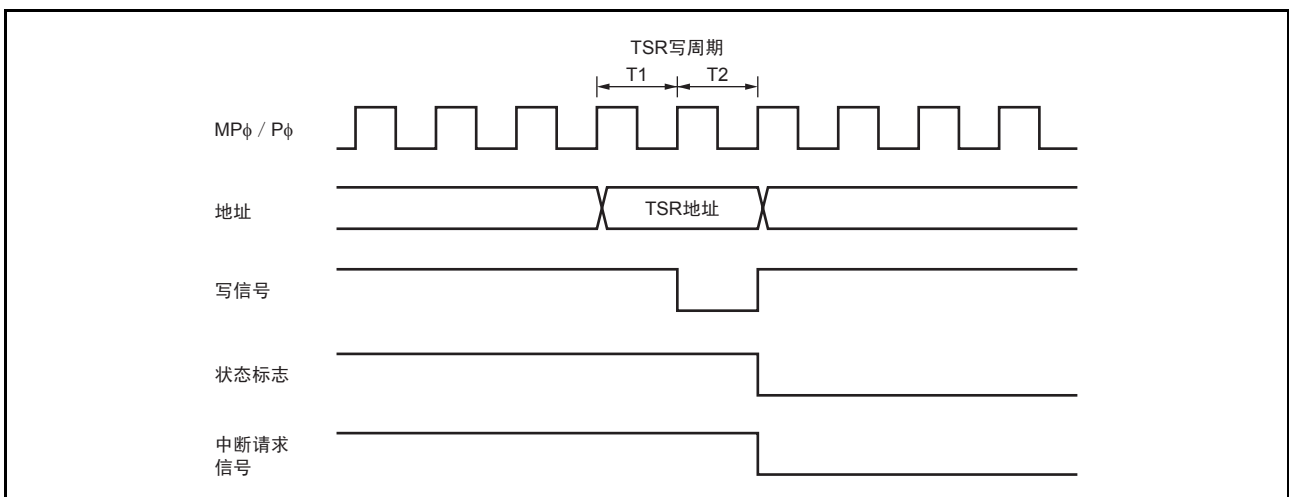


图 9.106 通过 CPU 清除状态标志的时序 (通道 5)

9.7 使用注意事项

9.7.1 模块待机模式的设定

MTU2 能通过待机控制寄存器允许或者禁止本模块的运行，初始值为停止 MTU2 的运行。另外，能通过解除模块待机模式进行寄存器的存取。详细内容请参照“第 19 章 低功耗模式”。

9.7.2 输入时钟的限制事项

在单边沿的情况下，输入时钟的脉宽需要在 1.5 个状态以上；在双边沿的情况下，输入时钟的脉宽需要在 2.5 个状态以上。必须注意：如果小于上述脉宽就不能正常运行。

在相位计数模式的情况下，2 个输入时钟的相位差和重叠都需要在 1.5 个状态以上，并且脉宽需要在 2.5 个状态以上。相位计数模式的输入时钟条件如图 9.107 所示。

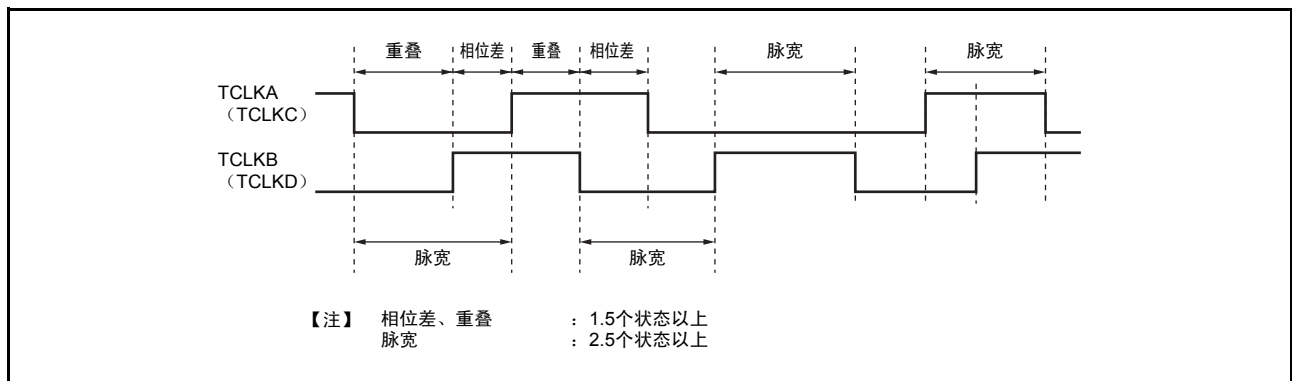


图 9.107 相位计数模式中的相位差、重叠和脉宽

9.7.3 周期设定的注意事项

如果设定为通过比较匹配清除计数器，就在 TCNT 和 TGR 的值一致的最后状态（更新 TCNT 一致后的计数值时）清除 TCNT。因此，实际的计数器频率为以下的表达式：

- 通道 0 ~ 4

$$f = \frac{MP\phi}{(N + 1)}$$

- 通道 5

$$f = \frac{MP\phi}{N}$$

f : 计数器频率
MP ϕ : MTU2 的时钟工作频率
N : TGR 的设定值

9.7.4 TCNT 的写操作和清除的竞争

如果在 TCNT 写周期中的 T2 状态产生计数器清除信号，就不写 TCNT 而优先清除 TCNT。此时序如图 9.108 所示。

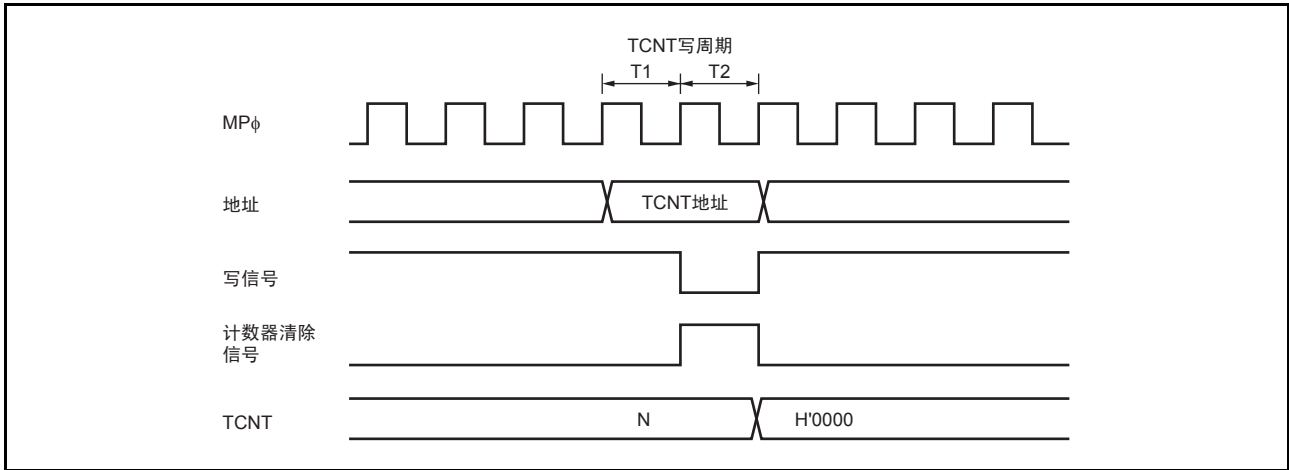


图 9.108 TCNT 的写操作和清除的竞争

9.7.5 TCNT 的写操作和递增计数的竞争

即使在 TCNT 写周期中的 T2 状态发生递增计数信号，也不进行递增计数而优先写 TCNT。此时序如图 9.109 所示。

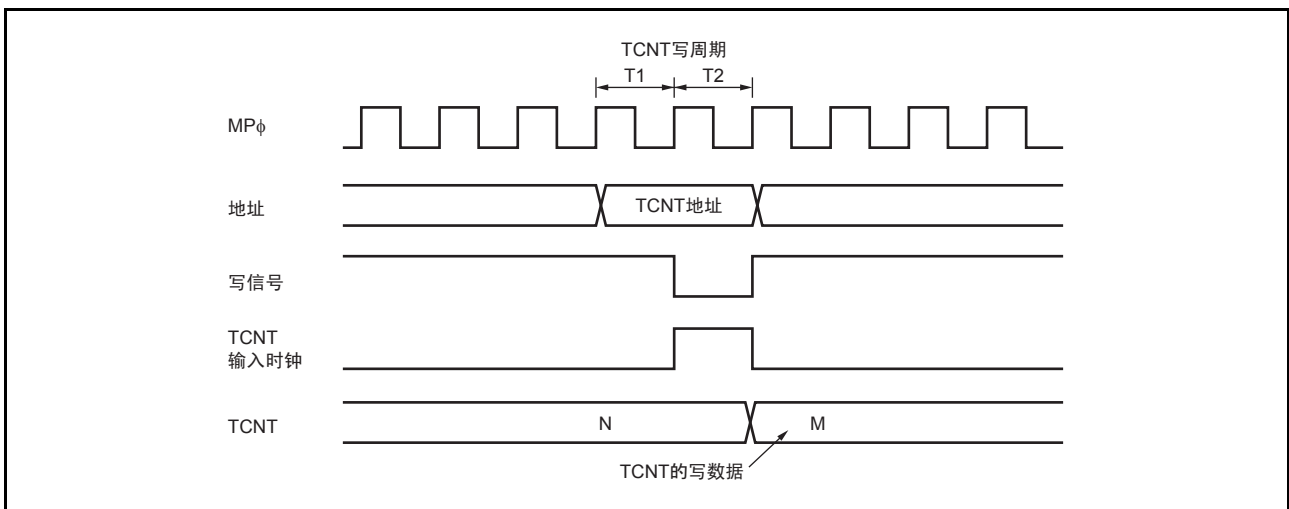


图 9.109 TCNT 的写操作和递增计数的竞争

9.7.6 TGR 的写操作和比较匹配的竞争

如果在 TGR 写周期中的 T2 状态产生比较匹配信号，就写 TGR 并且产生比较匹配信号。此时序如图 9.110 所示。

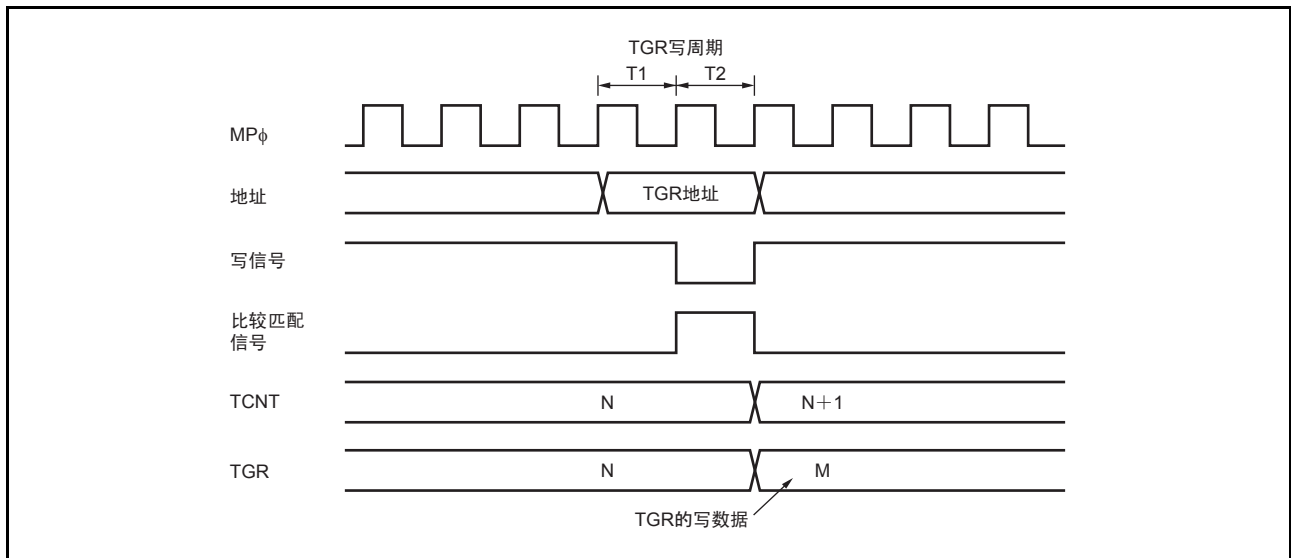


图 9.110 TGR 的写操作和比较匹配的竞争

9.7.7 缓冲寄存器的写操作和比较匹配的竞争

如果在 TGR 写周期中的 T2 状态产生比较匹配信号，由缓冲运行传送到 TGR 的数据就为写操作前的数据。此时序如图 9.111 所示。

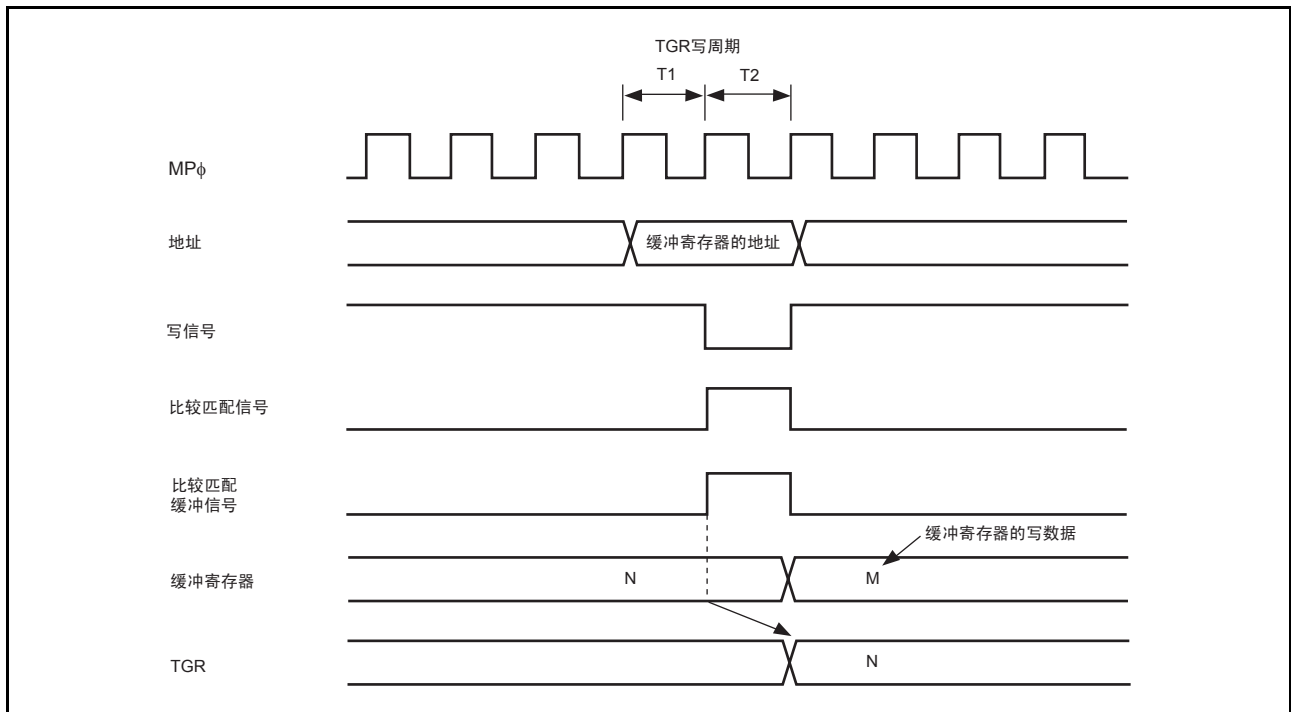


图 9.111 缓冲寄存器的写操作和比较匹配的竞争

9.7.8 缓冲寄存器的写操作和 TCNT 清除的竞争

通过缓冲传送模式寄存器 (TBTM) 将缓冲传送时序设定为清除 TCNT 时, 如果在 TGR 写周期中的 T2 状态产生 TCNT 的清除信号, 由缓冲运行传送到 TGR 的数据就为写操作前的数据。

此时序如图 9.112 所示。

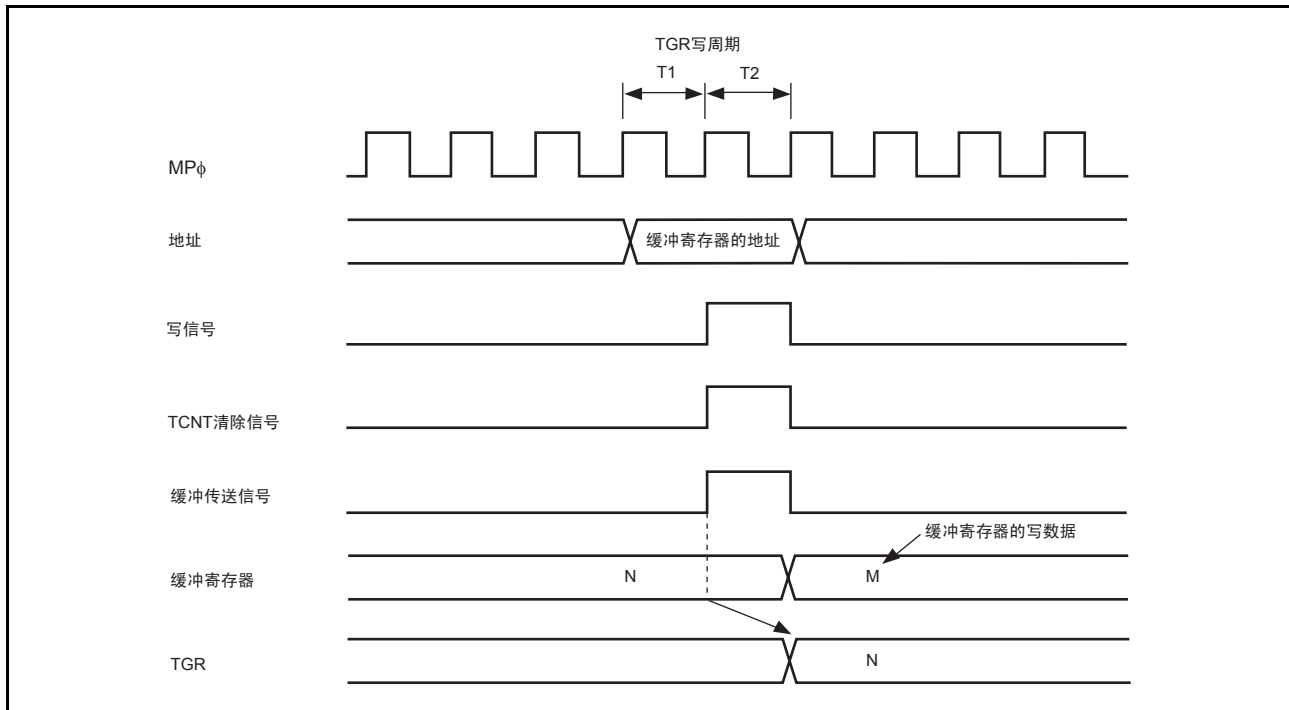


图 9.112 缓冲寄存器的写操作和 TCNT 清除的竞争

9.7.9 TGR 的读操作和输入捕捉的竞争

如果在 TGR 读周期中的 T1 状态产生输入捕捉信号，读出的数据在通道 0 ~ 4 中为输入捕捉传送前的数据，而在通道 5 中为输入捕捉传送后的数据。

此时序如图 9.113 和图 9.114 所示。

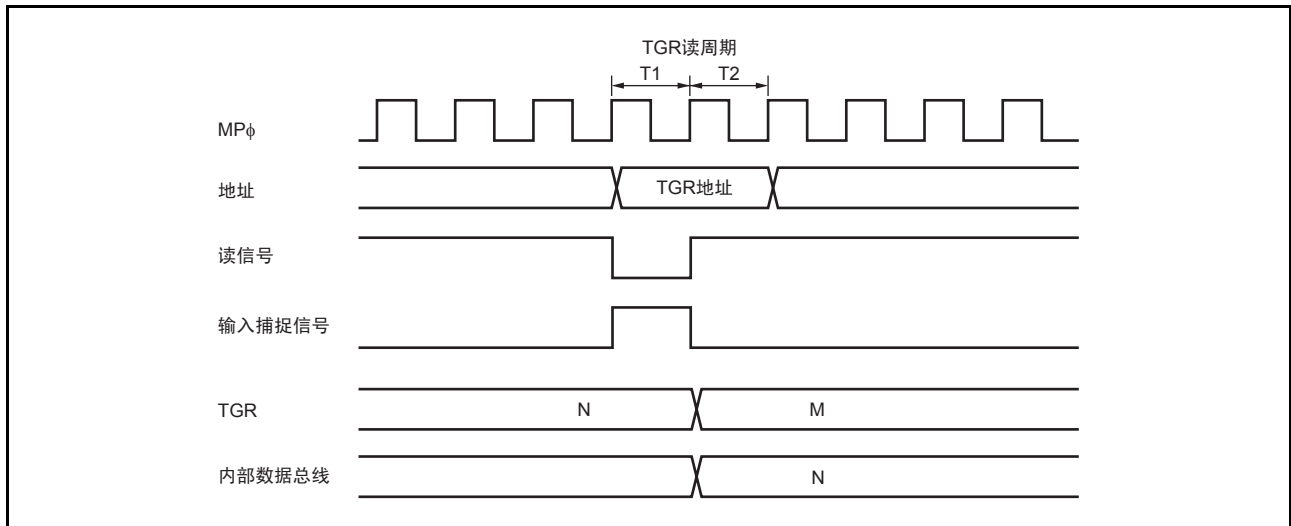


图 9.113 TGR 的读操作和输入捕捉的竞争（通道 0 ~ 4）

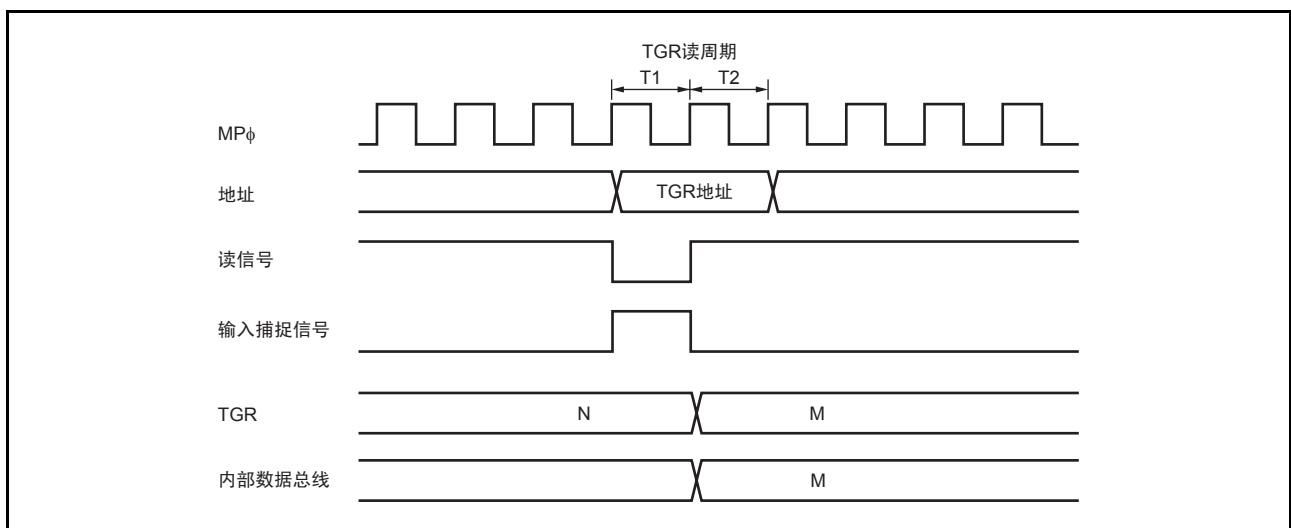


图 9.114 TGR 的读操作和输入捕捉的竞争（通道 5）

9.7.10 TGR 的写操作和输入捕捉的竞争

如果在 TGR 写周期中的 T2 状态产生输入捕捉信号，在通道 0 ~ 4 中不写 TGR 而优先输入捕捉，但是在通道 5 中写 TGR 并且产生输入捕捉信号。

此时序如图 9.115 和图 9.116 所示。

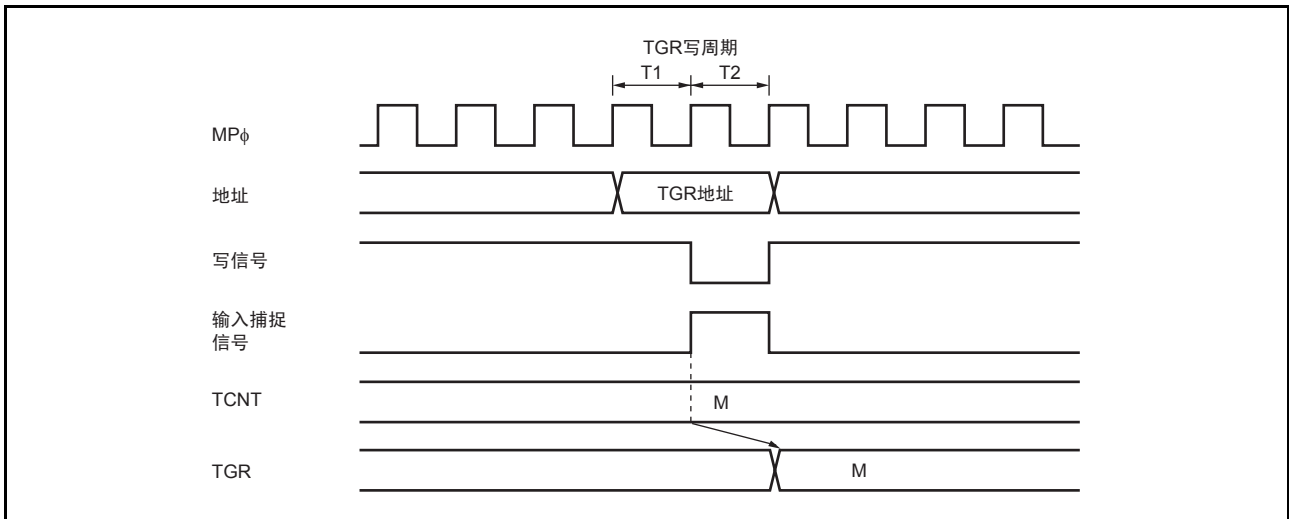


图 9.115 TGR 的写操作和输入捕捉的竞争（通道 0 ~ 4）

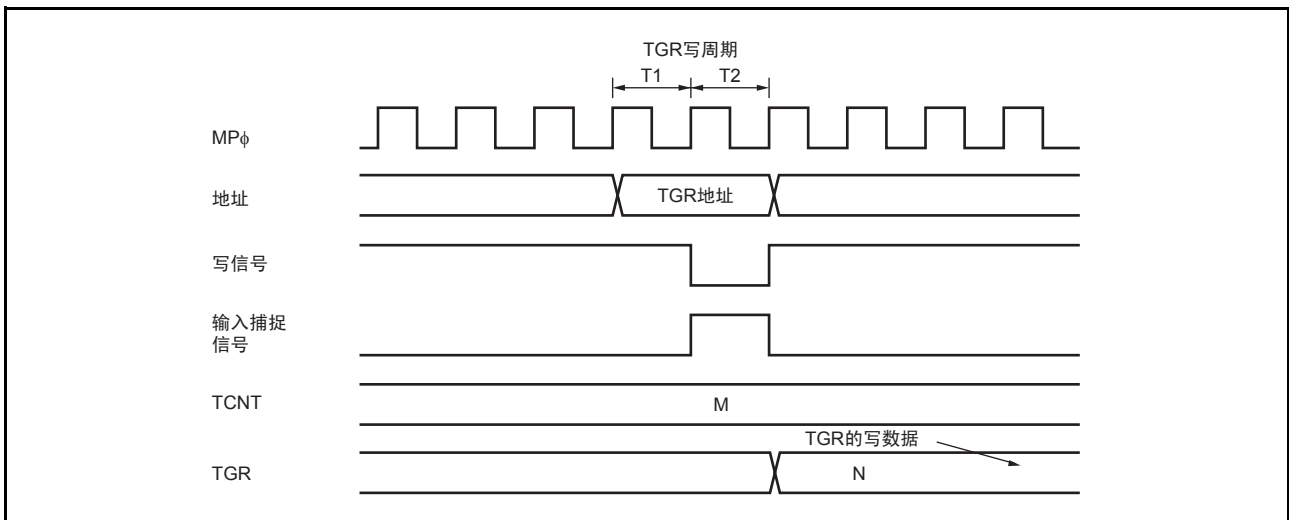


图 9.116 TGR 的写操作和输入捕捉的竞争（通道 5）

9.7.11 缓冲寄存器的写操作和输入捕捉的竞争

如果在缓冲器写周期中的 T2 状态产生输入捕捉信号，就不写缓冲寄存器而优先缓冲运行。此时序如图 9.117 所示。

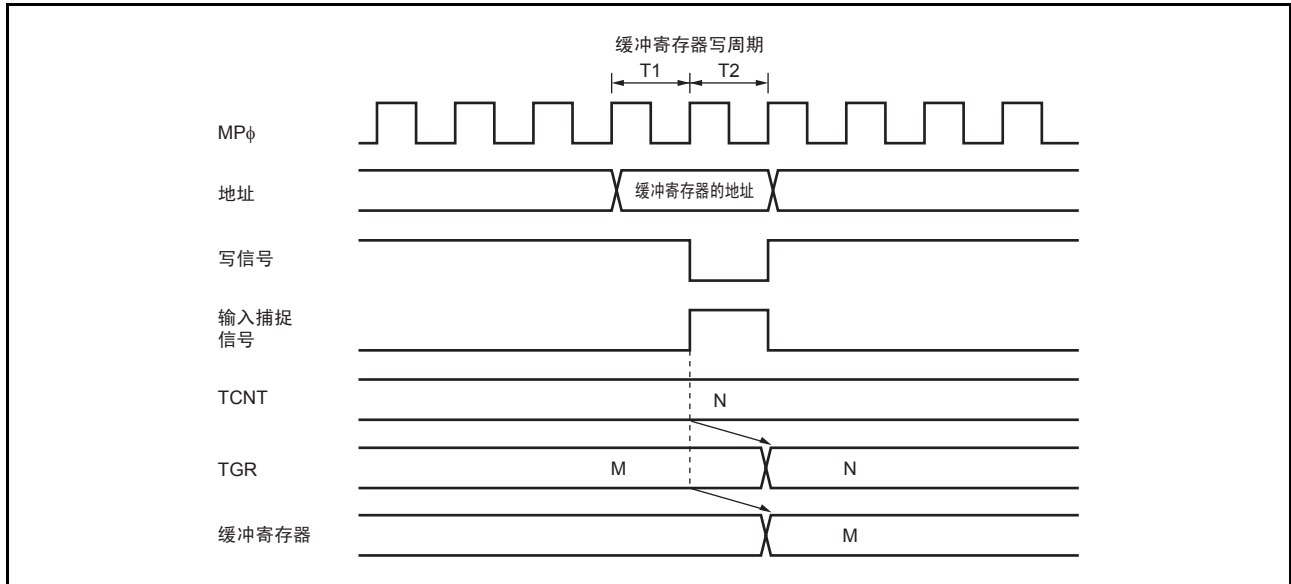


图 9.117 缓冲寄存器的写操作和输入捕捉的竞争

9.7.12 级联中 TCNT_2 的写操作和上溢 / 下溢的竞争

将定时器的计数器 (TCNT_1 和 TCNT_2) 进行级联, 如果 TCNT_1 在进行计数的瞬间 (TCNT_2 发生上溢 / 下溢的瞬间) 和 TCNT_2 写周期中的 T2 状态发生竞争, 就写 TCNT_2 来禁止 TCNT_1 的计数信号。此时, TGRA_1 用作比较匹配寄存器, 当 TGRA_1 和 TCNT_1 的值一致时, 就产生比较匹配信号。

如果选择 TCNT_1 计数时钟为通道 0 的输入捕捉源, TGRA_0 ~ D_0 就进行输入捕捉; 如果选择 TGRC_0 的比较匹配 / 输入捕捉为 TGRB_1 的输入捕捉源, TGRB_1 就进行输入捕捉。

此时序如图 9.118 所示。

另外, 在级联运行中设定 TCNT 的清除时, 必须进行通道 1 和通道 2 的同步设定。

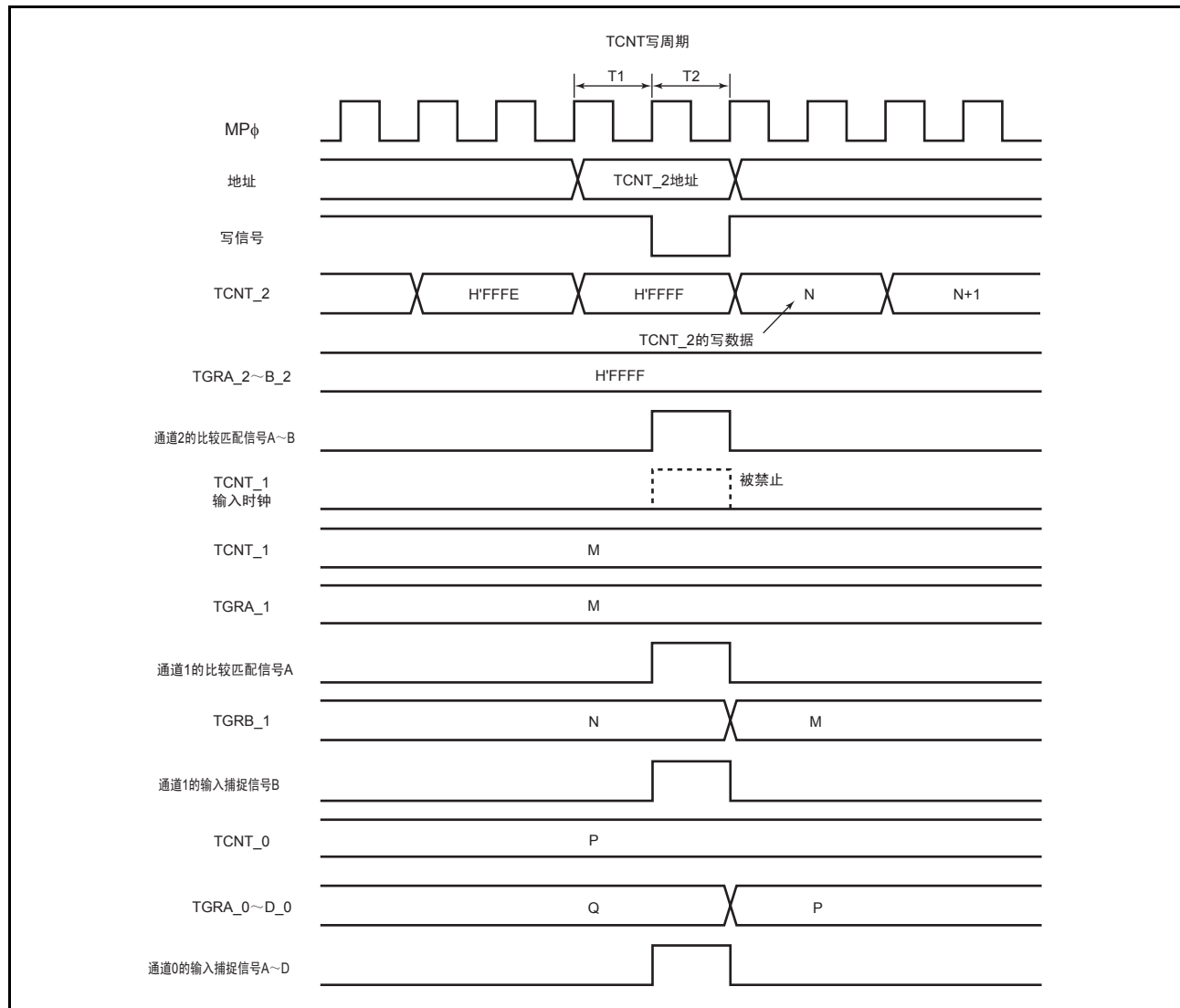


图 9.118 级联中 TCNT_2 的写操作和上溢 / 下溢的竞争

9.7.13 互补 PWM 模式停止时的计数器值

在互补 PWM 模式中，如果 TCNT_3 和 TCNT_4 在运行中停止计数，TCNT_3 就为定时器的空载时间寄存器 (TDDR) 的值，TCNT_4 为 H'0000。

当重新开始互补 PWM 模式时，就自动从初始状态开始计数。

此说明如图 9.119 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。

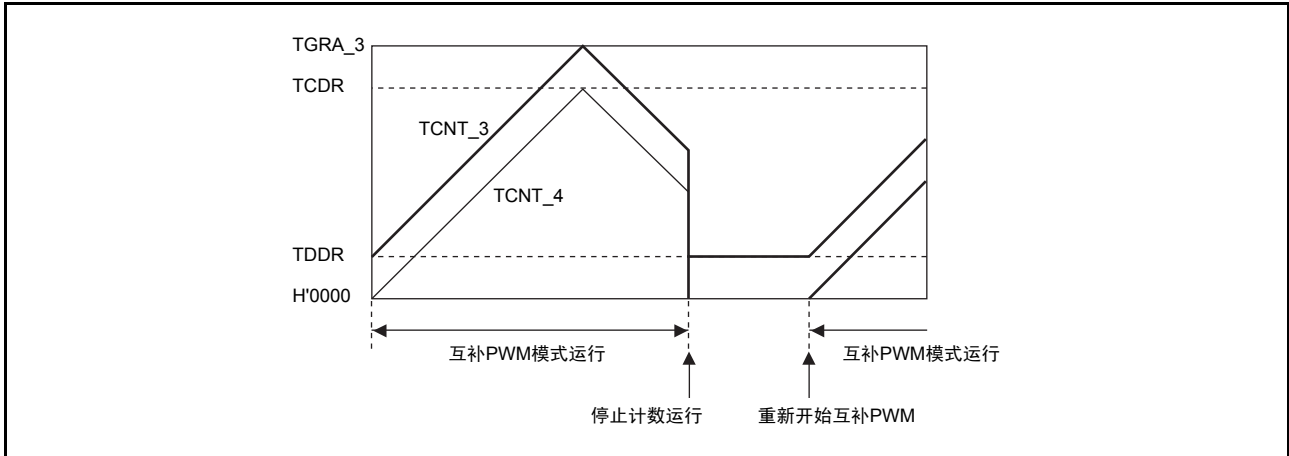


图 9.119 互补 PWM 模式停止时的计数器值

9.7.14 互补 PWM 模式中缓冲运行的设定

在互补 PWM 模式中，必须在缓冲运行中改写 PWM 周期设定寄存器 (TGRA_3)、定时器的周期数据寄存器 (TCDR) 和占空比设定寄存器 (TGRB_3、TGRA_4、TGRB_4)。

按照 TMDR_3 的 BFA 和 BFB 位的设定，进行互补 PWM 模式中的通道 3 和通道 4 的缓冲运行。如果将 TMDR_3 的 BFA 位设定为 1，TGRC_3 就用作 TGRA_3 的缓冲寄存器，TGRC_4 用作 TGRA_4 的缓冲寄存器，TCBR 用作 TCDR 的缓冲寄存器。

9.7.15 复位同步 PWM 模式的缓冲运行和比较匹配标志

要在复位同步 PWM 模式中设定缓冲运行时，必须将 TMDR_4 的 BFA 和 BFB 位设定为 0。如果将 TMDR_4 的 BFA 位设定为 1，TIOC4C 管脚就不能输出波形。

按照 TMDR_3 的 BFA 和 BFB 位的设定，进行复位同步 PWM 模式中的通道 3 和通道 4 的缓冲运行。例如，如果将 TMDR_3 的 BFA 位设定为 1，TGRC_3 就用作 TGRA_3 的缓冲寄存器，TGRC_4 用作 TGRA_4 的缓冲寄存器。

当 TGRC_3 和 TGRD_3 用作缓冲寄存器时，TSR_3、TSR_4 的 TGFC 位和 TGFD 位不被置位。

在将 TMDR_3 的 BFA 和 BFB 位设定为 1 并且 TMDR_4 的 BFA 和 BFB 位设定为 0 时，TGR_3、TGR_4、TIOC3 和 TIOC4 的运行例子如图 9.120 所示。

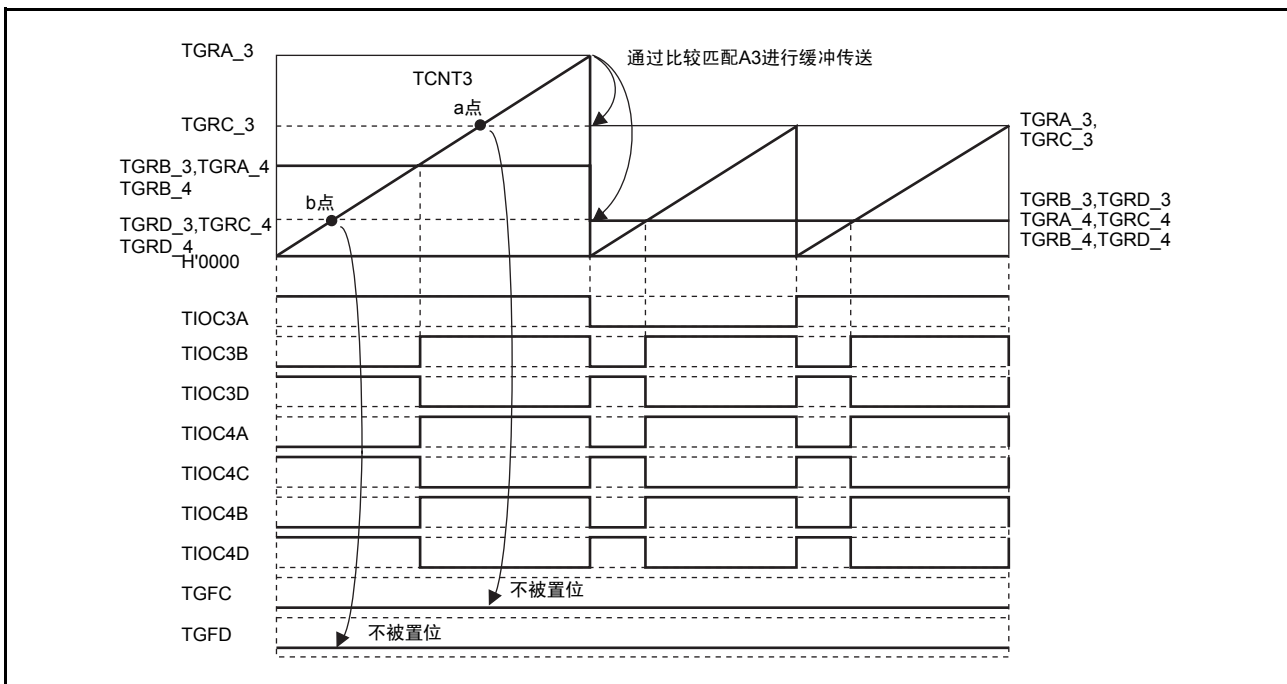


图 9.120 复位同步 PWM 模式的缓冲运行和比较匹配标志

9.7.16 复位同步 PWM 模式的上溢标志

如果设定复位同步 PWM 模式并将 TSTR 的 CST3 位设定为 1, 就开始 TCNT_3 和 TCNT_4 的计数运行。此时, TCNT_4 的计数时钟源和计数边沿服从 TCR_3 的设定。

在复位同步 PWM 模式中, 当周期寄存器 TGRA_3 的设定值为 H'FFFF 并给计数器清除源指定 TGRA_3 的比较匹配时, 如果 TCNT_3 和 TCNT_4 递增计数到 H'FFFF, 就产生和 TGRA_3 的比较匹配, 并且同时清除 TCNT_3 和 TCNT_4。此时, TSR 的上溢标志 TCFV 位不被置位。

在复位同步 PWM 模式中, 当周期寄存器 TGRA_3 的设定值为 H'FFFF 并给计数器清除源指定 TGRA_3 的比较匹配而没有进行同步设定时, TCFV 位的运行例子如图 9.121 所示。

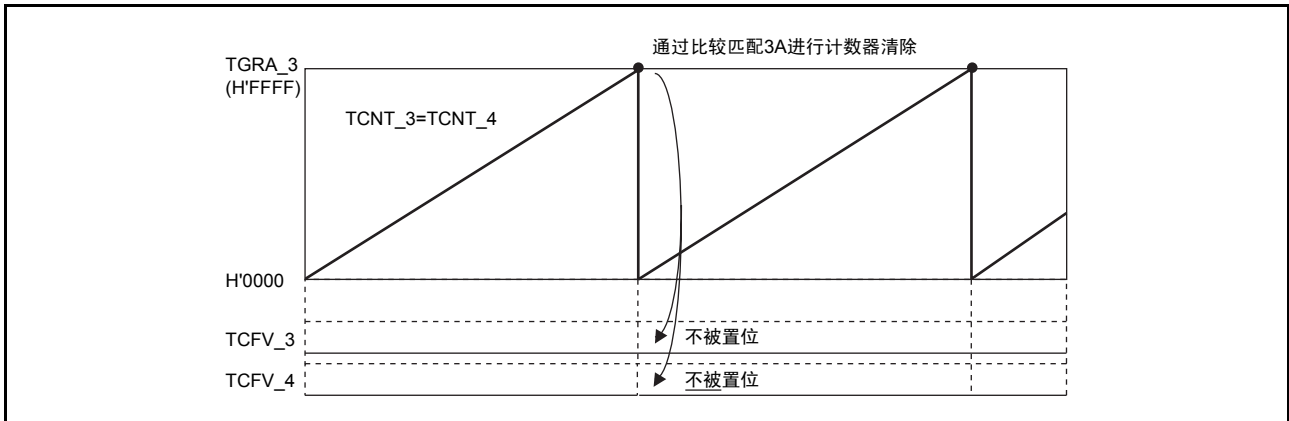


图 9.121 复位同步 PWM 模式的上溢标志

9.7.17 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 9.122 所示。

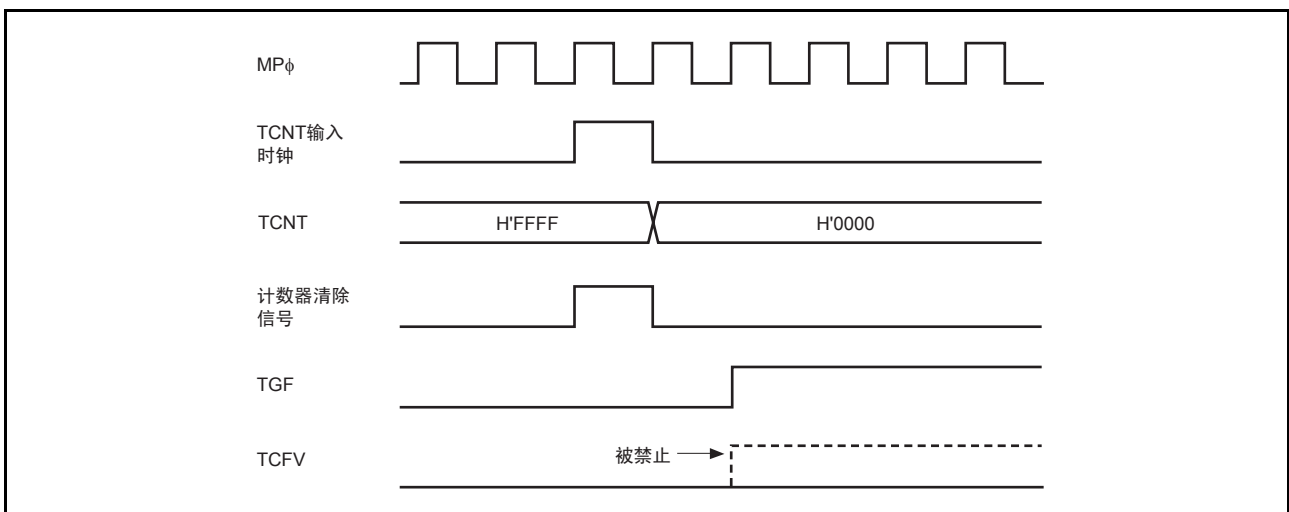


图 9.122 上溢和计数器清除的竞争

9.7.18 TCNT 的写操作和上溢 / 下溢的竞争

即使在 TCNT 写周期中的 T2 状态发生递增计数 / 递减计数和上溢 / 下溢，也优先写 TCNT 而不将 TSR 的 TCFV / TCFU 标志置位。

TCNT 的写操作和上溢发生竞争时的运行时序如图 9.123 所示。

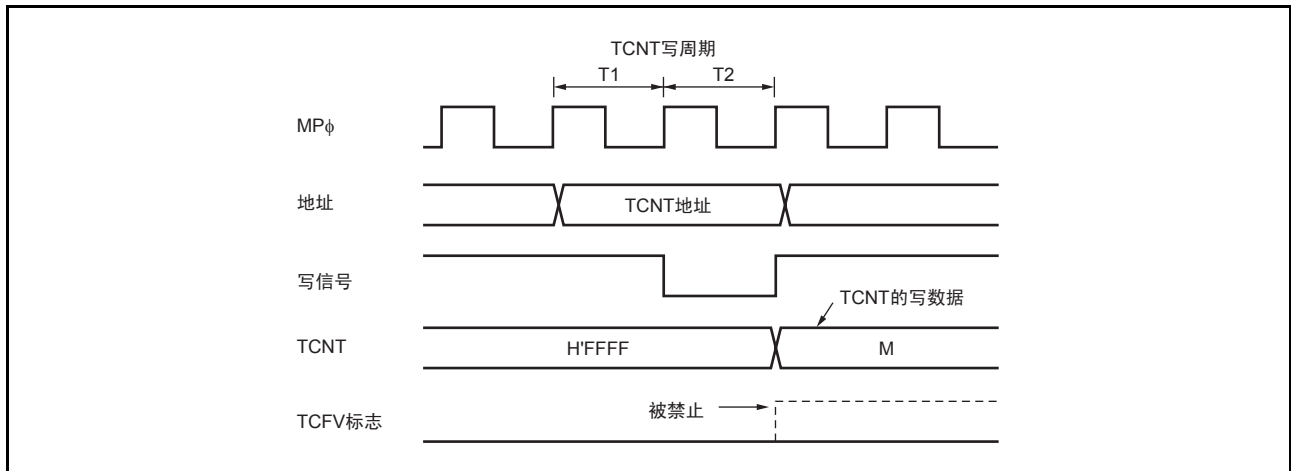


图 9.123 TCNT 的写操作和上溢的竞争

9.7.19 从通常运行或者 PWM 模式 1 转移到复位同步 PWM 模式时的注意事项

从通道 3、4 的通常运行或者 PWM 模式转移到复位同步 PWM 模式时，在将输出管脚（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，就不能正确地进行管脚的初始输出。

要从通常运行转移到同步 PWM 模式时，必须在将 H'11 写到 TIORH_3、TIO RL_3、TIORH_4 和 TIO RL_4 寄存器并将输出管脚初始化为低电平后，先设定寄存器的初始值 H'00 再进行模式转移。

要从 PWM 模式 1 转移到复位同步 PWM 模式时，必须暂且先转移到通常运行，在将输出管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。

9.7.20 互补 PWM 模式或者复位同步 PWM 模式的输出电平

在通道 3、4 为互补 PWM 模式或者复位同步 PWM 模式时，通过定时器的输出控制寄存器（TOCR）的 OLSP 和 OLSN 位设定 PWM 波形的输出电平，并且 TIOR 必须为 H'00。

9.7.21 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。

必须预先禁止中断再设定为模块待机模式。

9.7.22 级联中 TCNT_1 和 TCNT_2 的同时输入捕捉

在将定时器的计数器 1、2（TCNT_1 和 TCNT_2）进行级联后用作 32 位计数器时，即使 TIOC1A 和 TIOC2A 或者 TIOC1B 和 TIOC2B 同时进行输入捕捉的输入，也会在与内部时钟同步将输入到 TCNT_1 和 TCNT_2 的外部输入捕捉信号取到内部时，有可能因 TIOC1A、TIOC2A 或者 TIOC1B、TIOC2B 的取时序产生偏差而不能正确地捕捉级联计数器的值。

例如，TCNT_1（高 16 位的计数器）应该捕捉由 TCNT_2（低 16 位的计数器）的上溢产生的递增计数值，却会捕捉递增计数前的计数值。此时，应该将 TCNT_1=H'FFF1 和 TCNT_2=H'0000 的值传送到 TGRA_1 和 TGRA_2 或者 TGRB_1 和 TGRB_2，却会误传送 TCNT_1=H'FFF0 和 TCNT_2=H'0000 的值。

9.8 MTU2 输出管脚的初始化方法

9.8.1 运行模式

MTU2 有以下 6 种运行模式，能在任意的模式中进行波形输出。

- 正常模式（通道 0 ~ 4）
- PWM 模式 1（通道 0 ~ 4）
- PWM 模式 2（通道 0 ~ 2）
- 相位计数模式 1 ~ 4（通道 1、2）
- 互补 PWM 模式（通道 3、4）
- 复位同步 PWM 模式（通道 3、4）

在此说明各模式中的 MTU2 输出管脚的初始化方法。

9.8.2 开始复位时的运行

MTU2 的输出管脚（TIOC*）在复位或者待机模式时被初始化为低电平。因为通过管脚功能控制器（PFC）选择 MTU2 的管脚功能，所以在设定 PFC 时将当时的 MTU2 的管脚状态输出到端口。如果在复位后立即通过 PFC 选择 MTU2 的输出，就将 MTU2 输出的初始状态（低电平）输出到端口。在有效电平为低电平的情况下，因为系统马上就运行，所以必须在结束 MTU2 的输出管脚的初始设定后进行 PFC 的设定。

【注】 * 通道号 + 端口符号。

9.8.3 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 将管脚的输出转换为端口输出，并且输出有效电平的反相信号。另外，有关大电流管脚，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管脚的初始化步骤以及重新设定后在其他运行模式中重新开始时的步骤如下所示。

如上所述，因为 MTU2 有 6 种运行模式，所以有 36 种模式转移的组合，但是有通道和模式的组合中不存在的转移，模式转移的组合一览表如表 9.59 所示。

表中使用下述的符号表示：

Normal：正常模式

PWM1：PWM 模式 1

PWM2：PWM 模式 2

PCM：相位计数模式 1 ~ 4

CPWM：互补 PWM 模式

RPWM：复位同步 PWM 模式

表 9.59 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)

9.8.4 因运行中的异常等而初始化管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) 选择的管脚输出电平模式 (Normal、PWM1、PWM2、PCM) 时, 必须通过设定 TIOCR 初始化管脚。
- 因为在 PWM 模式 1 中不将波形输出到 TIOC*B (TIOC*D) 管脚, 所以即使设定 TIOCR 也不初始化管脚。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管脚, 所以即使设定 TIOCR 也不初始化管脚。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管脚。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 的管脚。要初始化 TGRC 的管脚时, 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) 选择的管脚输出电平模式 (CPWM、RPWM) 时, 必须转移到正常模式并且通过 TIOCR 进行初始化, 在将 TIOCR 返回到初始值后, 通过定时器的输出器件允许寄存器 (TOER) 暂时禁止通道 3、4 的输出, 然后按照模式的设定步骤 (TOCR 的设定、TMDR 的设定和 TOER 的设定) 运行。

【注】 本项记述中的 * 为通道号。

按照表 9.59 的组合 No. 初始化管脚的步骤如下所示。另外，有效电平为低电平。

(1) 在正常模式的运行中发生异常并在正常模式中重新开始时的运行

在正常模式中发生异常，重新设定后在正常模式中重新开始时的说明如图 9.124 所示。

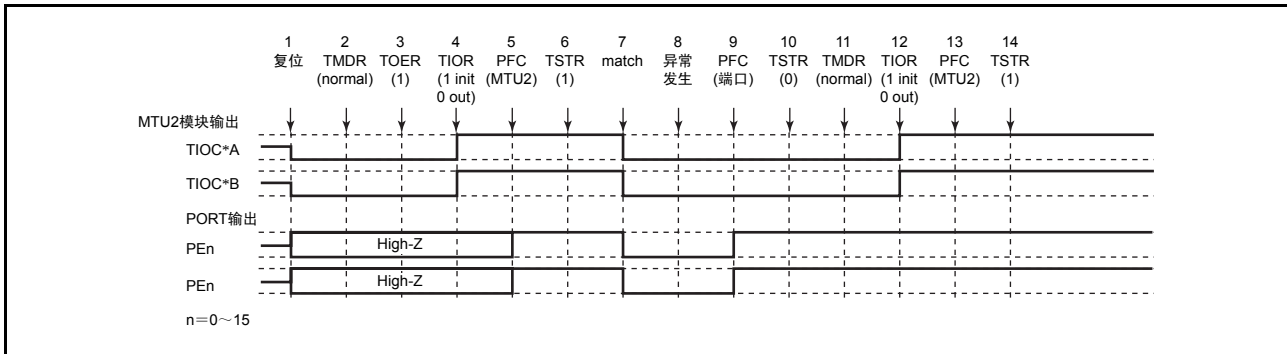


图 9.124 在正常模式中发生异常并在正常模式中恢复

1. 在复位后，MTU2 的输出为 Low 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 初始化管脚前，必须通过 TOER 允许输出。
4. 必须通过 TIOR 初始化管脚（例中，初始输出为高电平，比较匹配为低电平输出）。
5. 必须通过 PFC 设定为 MTU2 的输出。
6. 通过 TSTR 开始计数。
7. 通过产生比较匹配，输出低电平。
8. 发生了异常。
9. 必须通过 PFC 设定为端口输出，输出有效电平的反相信号。
10. 通过 TSTR 停止计数。
11. 在正常模式中重新开始时不需要此步骤。
12. 必须通过 TIOR 初始化管脚。
13. 必须通过 PFC 设定为 MTU2 的输出。
14. 通过 TSTR 重新开始。

(2) 在正常模式的运行中发生异常并在 PWM 模式 1 中重新开始时的运行

在正常模式中发生异常，重新设定后在 PWM 模式 1 中重新开始时的说明如图 9.125 所示。

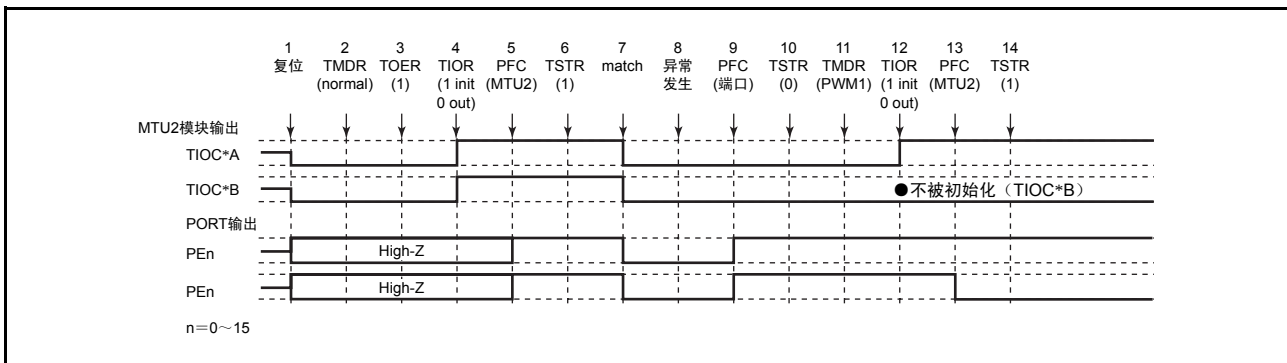


图 9.125 在正常模式中发生异常并在 PWM 模式 1 中恢复

1 ~ 10 和图 9.124 通用。

11. 设定 PWM 模式 1。

12. 必须通过 TIOR 初始化管脚（在 PWM 模式 1 中不初始化 TIOC*B。要进行初始化时，必须在正常模式中进行初始化后转移到 PWM 模式 1）。

13. 必须通过 PFC 设定为 MTU2 的输出。

14. 通过 TSTR 重新开始。

(3) 在正常模式的运行中发生异常并在 PWM 模式 2 中重新开始时的运行

在正常模式中发生异常，重新设定后在 PWM 模式 2 中重新开始时的说明如图 9.126 所示。

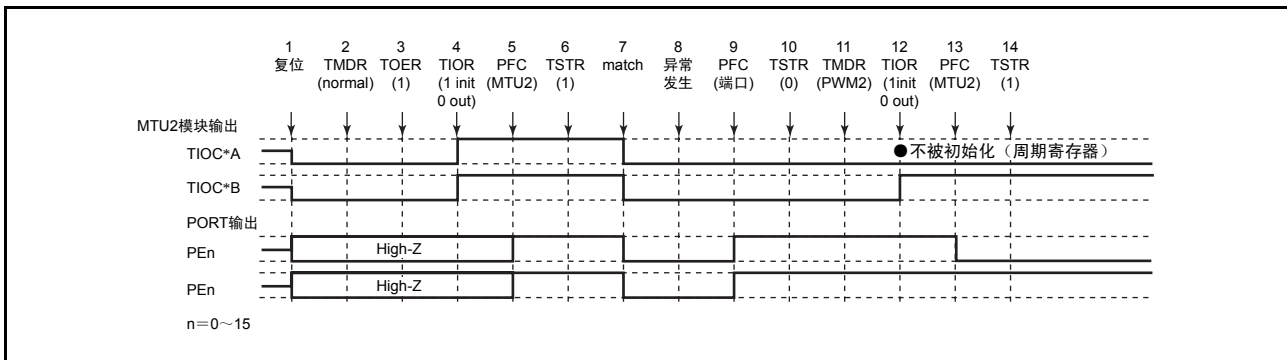


图 9.126 在正常模式中发生异常并在 PWM 模式 2 中恢复

1 ~ 10 和图 9.124 通用。

11. 设定 PWM 模式 2。

12. 必须通过 TIOR 初始化管脚（在 PWM 模式 2 中不初始化周期寄存器的管脚。要进行初始化时，必须在正常模式中进行初始化后转移到 PWM 模式 2）。

13. 通过 PFC 设定为 MTU2 的输出。

14. 通过 TSTR 重新开始。

【注】 只能在通道 0 ~ 2 中设定 PWM 模式 2，因此不需要设定 TOER。

(4) 在正常模式的运行中发生异常并在相位计数模式中重新开始时的运行

在正常模式中发生异常，重新设定后在相位计数模式中重新开始时的说明如图 9.127 所示。

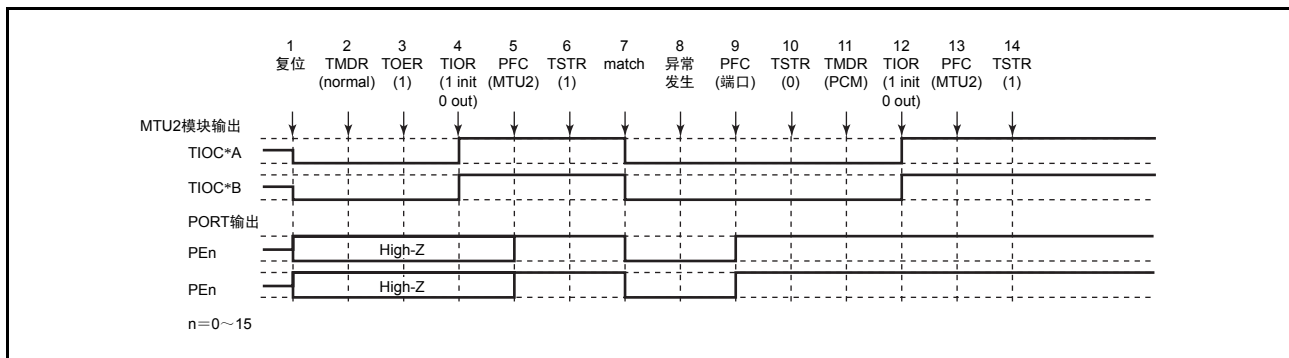


图 9.127 在正常模式中发生异常并在相位计数模式中恢复

1 ~ 10 和图 9.124 通用。

11. 设定相位计数模式。

12. 必须通过 TIOR 初始化管脚。

13. 必须通过 PFC 设定为 MTU2 的输出。

14. 通过 TSTR 重新开始。

【注】 只能在通道 1、2 中设定相位计数模式，因此不需要设定 TOER。

(5) 在正常模式的运行中发生异常并在互补 PWM 模式中重新开始时的运行

在正常模式中发生异常，重新设定后在互补 PWM 模式中重新开始的说明如图 9.128 所示。

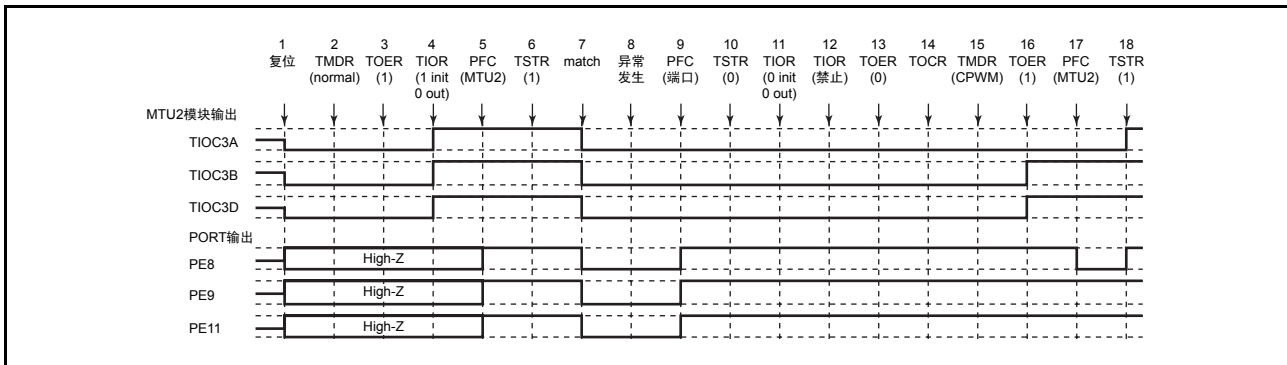


图 9.128 在正常模式中发生异常并在互补 PWM 模式中恢复

1 ~ 10 和图 9.124 通用。

11. 必须通过 TIOR 初始化正常模式的波形生成部。
12. 必须通过 TIOR 禁止正常模式的波形生成部的运行。
13. 必须通过 TOER 禁止通道 3、4 的输出。
14. 必须通过 TOCR 选择互补 PWM 的输出电平并允许或者禁止周期输出。
15. 设定互补 PWM。
16. 必须通过 TOER 允许通道 3、4 的输出。
17. 必须通过 PFC 设定为 MTU2 的输出。
18. 通过 TSTR 重新开始。

(6) 在正常模式的运行中发生异常并在复位同步 PWM 模式中重新开始时的运行

在正常模式中发生异常，重新设定后在复位同步 PWM 模式中重新开始的说明如图 9.129 所示。

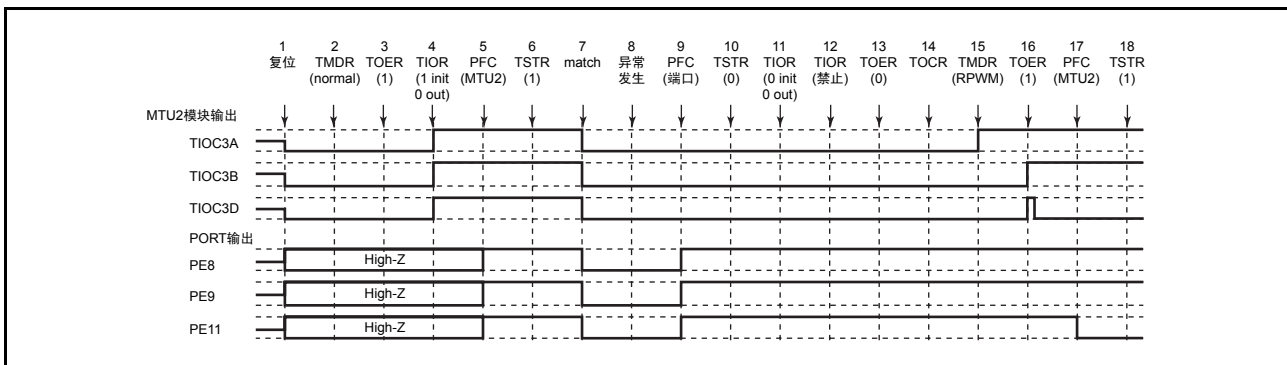


图 9.129 在正常模式中发生异常并在复位同步 PWM 模式中恢复

1 ~ 13 和图 9.124 通用。

14. 必须通过 TOCR 选择复位同步 PWM 的输出电平并允许或者禁止周期输出。

15. 设定复位同步 PWM。

16. 必须通过 TOER 允许通道 3、4 的输出。

17. 必须通过 PFC 设定为 MTU2 的输出。

18. 通过 TSTR 重新开始。

(7) 在 PWM 模式 1 的运行中发生异常并在正常模式中重新开始时的运行

在 PWM 模式 1 中发生异常，重新设定后在正常模式中重新开始时的说明如图 9.130 所示。

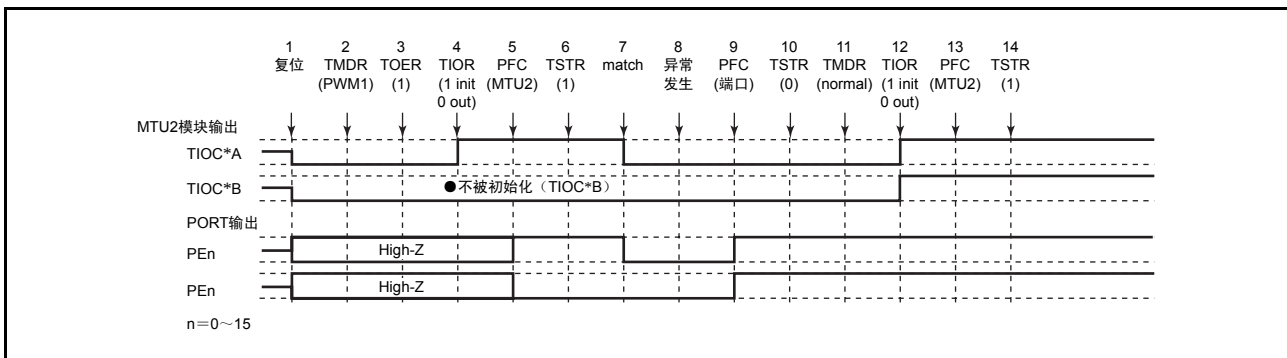


图 9.130 在 PWM 模式 1 中发生异常并在正常模式中恢复

1. 在复位后，MTU2 的输出为低电平，端口为高阻抗状态。
2. 必须设定 PWM 模式 1。
3. 在通道 3、4 中通过 TIOR 初始化管脚前，必须通过 TOER 允许输出。
4. 必须通过 TIOR 初始化管脚（例中，初始输出为高电平，比较匹配为低电平输出，在 PWM 模式 1 中不初始化 TIOC*B）。
5. 通过 PFC 设定为 MTU2 的输出。
6. 通过 TSTR 开始计数。
7. 通过产生比较匹配，输出低电平。
8. 发生了异常。
9. 必须通过 PFC 设定为端口输出，输出有效电平的反相信号。
10. 通过 TSTR 停止计数。
11. 必须设定正常模式。
12. 必须通过 TIOR 初始化管脚。
13. 必须通过 PFC 设定为 MTU2 的输出。
14. 通过 TSTR 重新开始。

(8) 在 PWM 模式 1 的运行中发生异常并在 PWM 模式 1 中重新开始时的运行

在 PWM 模式 1 中发生异常，重新设定后在 PWM 模式 1 中重新开始时的说明如图 9.131 所示。

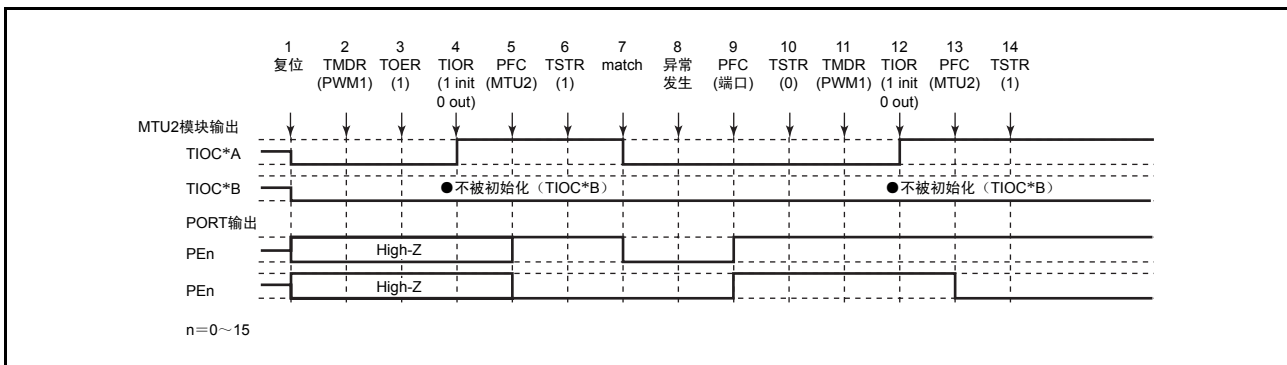


图 9.131 在 PWM 模式 1 中发生异常并在 PWM 模式 1 中恢复

1 ~ 10 和图 9.130 通用。

11. 在 PWM 模式 1 中重新开始时不需要此步骤。

12. 必须通过 TIOR 初始化管脚（在 PWM 模式 1 中不初始化 TIOC*B）。

13. 必须通过 PFC 设定为 MTU2 的输出。

14. 通过 TSTR 重新开始。

(9) 在 PWM 模式 1 的运行中发生异常并在 PWM 模式 2 中重新开始时的运行

在 PWM 模式 1 中发生异常，重新设定后在 PWM 模式 2 中重新开始时的说明如图 9.132 所示。

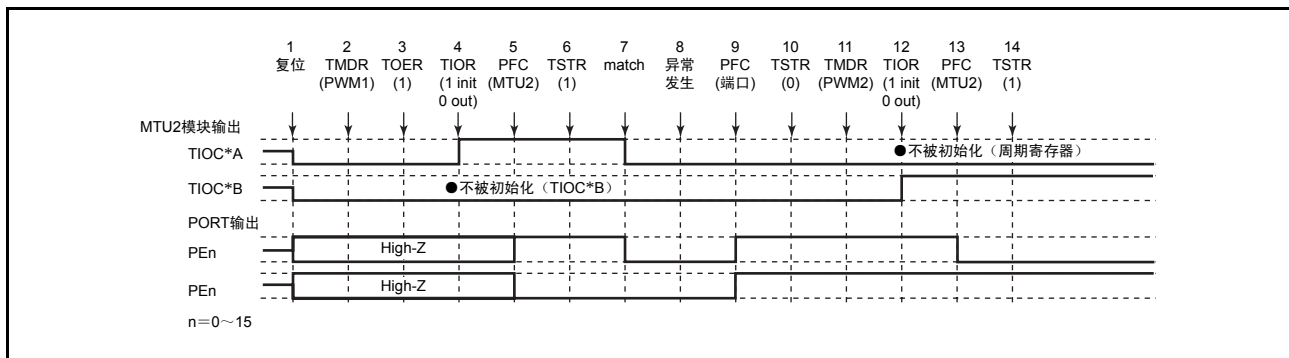


图 9.132 在 PWM 模式 1 中发生异常并在 PWM 模式 2 中恢复

1 ~ 10 和图 9.130 通用。

11. 设定 PWM 模式 2。

12. 必须通过 TIOR 初始化管脚（在 PWM 模式 2 中不初始化周期寄存器的管脚）。

13. 必须通过 PFC 设定为 MTU2 的输出。

14. 通过 TSTR 重新开始。

【注】 只能在通道 0 ~ 2 中设定 PWM 模式 2，因此不需要设定 TOER。

(10) 在 PWM 模式 1 的运行中发生异常并在相位计数模式中重新开始时的运行

在 PWM 模式 1 中发生异常，重新设定后在相位计数模式中重新开始时的说明如图 9.133 所示。

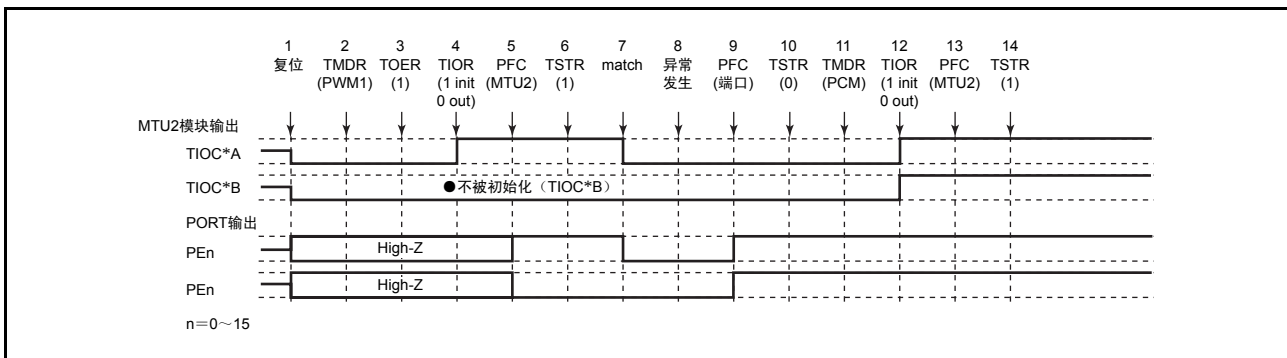


图 9.133 在 PWM 模式 1 中发生异常并在相位计数模式中恢复

1 ~ 10 和图 9.130 通用。

11. 设定相位计数模式。

12. 必须通过 TIOR 初始化管脚。

13. 必须通过 PFC 设定为 MTU2 的输出。

14. 通过 TSTR 重新开始。

【注】 只能在通道 1、2 中设定相位计数模式，因此不需要设定 TOER。

(11) 在 PWM 模式 1 的运行中发生异常并在互补 PWM 模式中重新开始时的运行

在 PWM 模式 1 中发生异常，重新设定后在互补 PWM 模式中重新开始时的说明如图 9.134 所示。

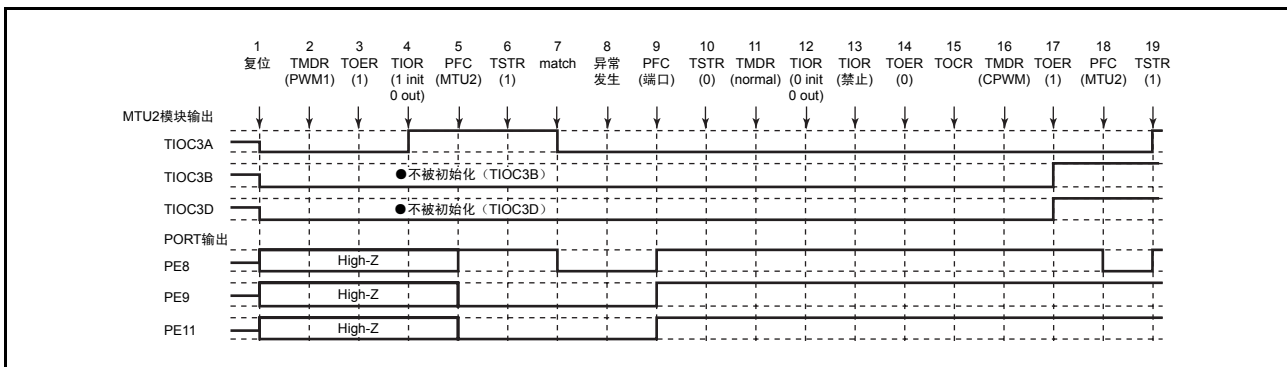


图 9.134 在 PWM 模式 1 中发生异常并在互补 PWM 模式中恢复

1 ~ 10 和图 9.130 通用。

11. 为了初始化波形生成部，必须设定正常模式。
12. 必须通过 TIOR 初始化 PWM 模式 1 的波形生成部。
13. 必须通过 TIOR 禁止 PWM 模式 1 的波形生成部的运行
14. 必须通过 TOER 禁止通道 3、4 的输出。
15. 必须通过 TOCR 选择 PWM 的输出电平并允许或者禁止周期输出。
16. 设定互补 PWM。
17. 必须通过 TOER 允许通道 3、4 的输出。
18. 必须通过 PFC 设定为 MTU2 的输出。
19. 通过 TSTR 重新开始。

(12) 在 PWM 模式 1 的运行中发生异常并在复位同步 PWM 模式中重新开始时的运行

在 PWM 模式 1 中发生异常，重新设定后在复位同步 PWM 模式中重新开始的说明如图 9.135 所示。

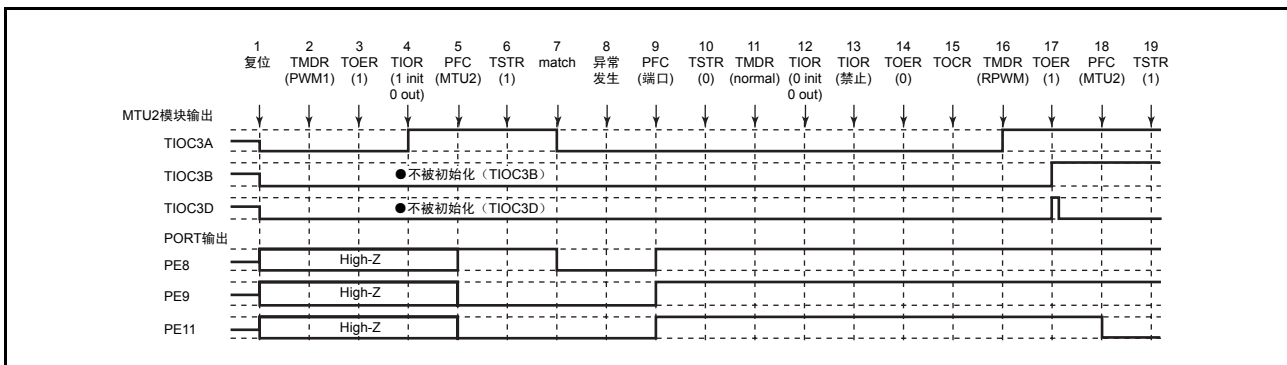


图 9.135 在 PWM 模式 1 中发生异常并在复位同步 PWM 模式中恢复

1 ~ 14 和图 9.134 通用。

15. 必须通过 TOCR 选择同步 PWM 的输出电平并允许或者禁止周期输出。

16. 设定复位同步 PWM。

17. 必须通过 TOER 允许通道 3、4 的输出。

18. 通过 PFC 设定为 MTU2 的输出。

19. 通过 TSTR 重新开始。

(13) 在 PWM 模式 2 的运行中发生异常并在正常模式中重新开始时的运行

在 PWM 模式 2 中发生异常，重新设定后在正常模式中重新开始时的说明如图 9.136 所示。

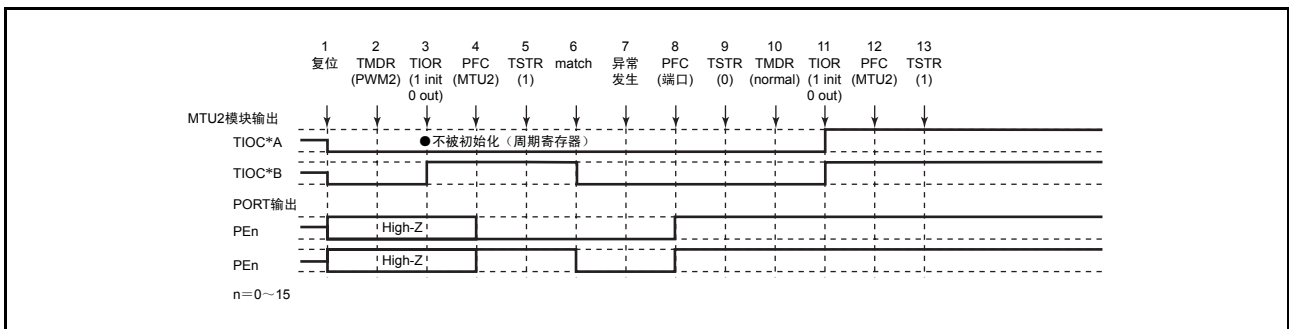


图 9.136 在 PWM 模式 2 中发生异常并在正常模式中恢复

1. 在复位后，MTU2 的输出为低电平，端口为高阻抗状态。
2. 必须设定 PWM 模式 2。
3. 必须通过 TIOR 初始化管脚（例中，初始输出为高电平，比较匹配为低电平输出，在 PWM 模式 2 中不初始化周期寄存器的管脚，TIOC*A 为周期寄存器）。
4. 必须通过 PFC 设定为 MTU2 的输出。
5. 通过 TSTR 开始计数。
6. 通过产生比较匹配，输出低电平。
7. 发生了异常。
8. 必须通过 PFC 设定端口输出，输出有效电平的反相信号。
9. 通过 TSTR 停止计数。
10. 必须设定正常模式。
11. 必须通过 TIOR 初始化管脚。
12. 必须通过 PFC 设定为 MTU2 的输出。
13. 通过 TSTR 重新开始。

(14) 在 PWM 模式 2 的运行中发生异常并在 PWM 模式 1 中重新开始时的运行

在 PWM 模式 2 中发生异常，重新设定后在 PWM 模式 1 中重新开始时的说明如图 9.137 所示。

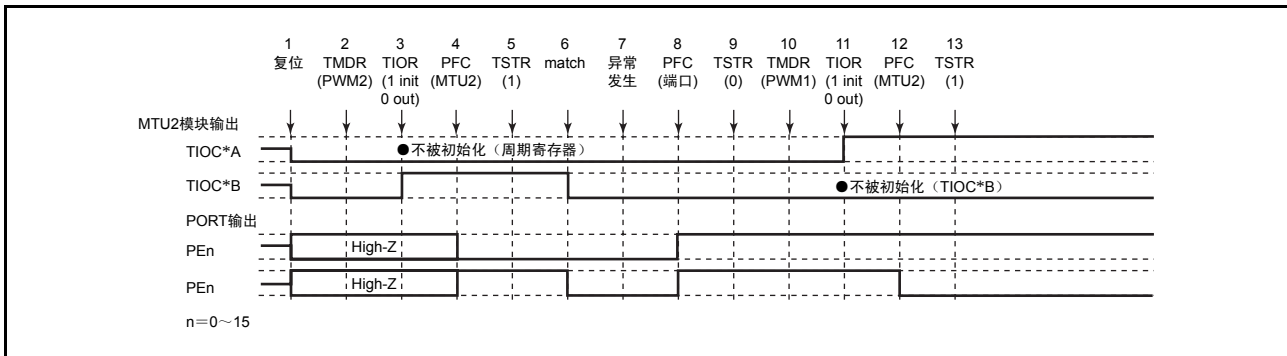


图 9.137 在 PWM 模式 2 中发生异常并在 PWM 模式 1 中恢复

1 ~ 9 和图 9.136 通用。

10. 设定 PWM 模式 1。

11. 必须通过 TIOR 初始化管脚（在 PWM 模式 1 中不初始化 TIOC*B）。

12. 必须通过 PFC 设定为 MTU2 的输出。

13. 通过 TSTR 重新开始。

(15) 在 PWM 模式 2 的运行中发生异常并在 PWM 模式 2 中重新开始时的运行

在 PWM 模式 2 中发生异常，重新设定后在 PWM 模式 2 中重新开始时的说明如图 9.138 所示。

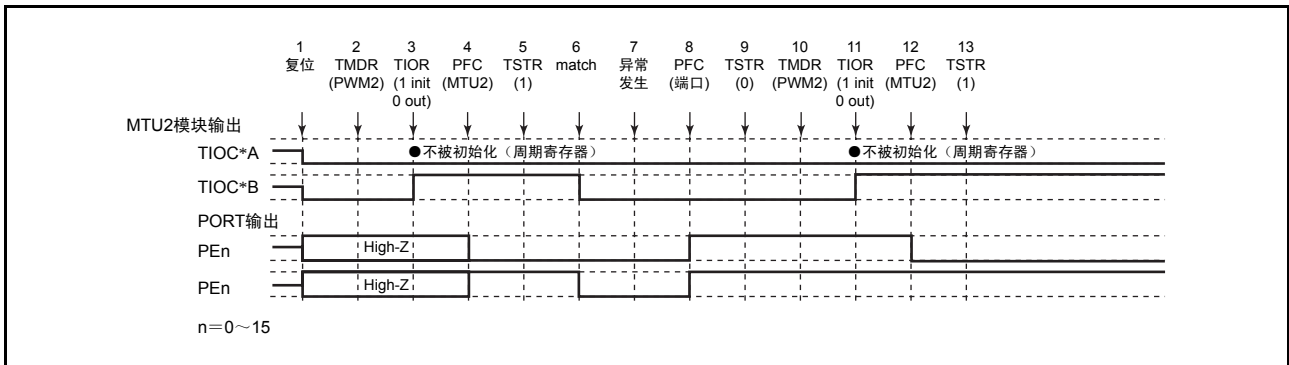


图 9.138 PWM 模式 2 中发生异常并在 PWM 模式 2 中恢复

1 ~ 9 和图 9.136 通用。

10. 在 PWM 模式 2 中重新开始时不需要此步骤。

11. 必须通过 TIOR 初始化管脚（在 PWM 模式 2 中不初始化周期寄存器的管脚）。

12. 必须通过 PFC 设定为 MTU2 的输出。

13. 通过 TSTR 重新开始。

(16) 在 PWM 模式 2 的运行中发生异常并在相位计数模式中重新开始时的运行

在 PWM 模式 2 中发生异常，重新再设定后在位相计数模式中重新开始的说明如图 9.139 所示。

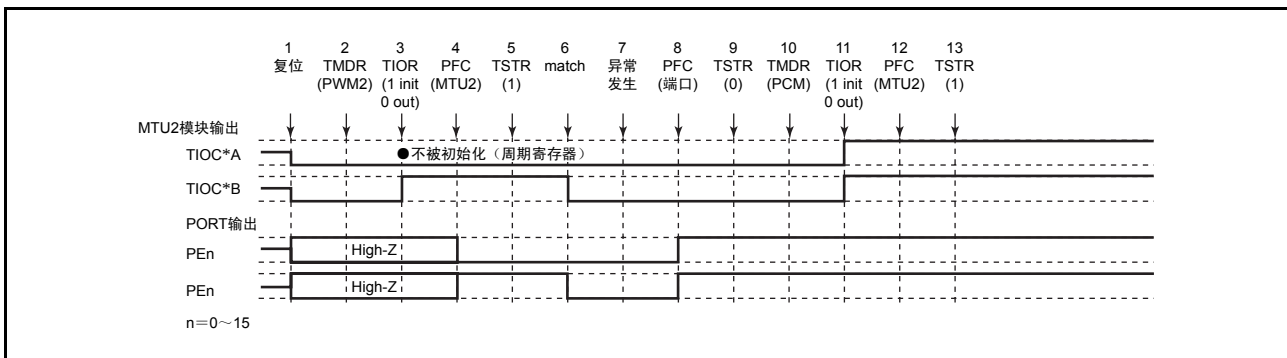


图 9.139 在 PWM 模式 2 中发生异常并在相位计数模式中恢复

1 ~ 9 和图 9.136 通用。

10. 设定相位计数模式。

11. 必须通过 TIOR 初始化管脚。

12. 必须通过 PFC 设定为 MTU2 的输出。

13. 通过 TSTR 重新开始。

(17) 在相位计数模式的运行中发生异常并在正常模式中重新开始时的运行

在相位计数模式中发生异常，重新设定后在正常模式中重新开始的说明如图 9.140 所示。

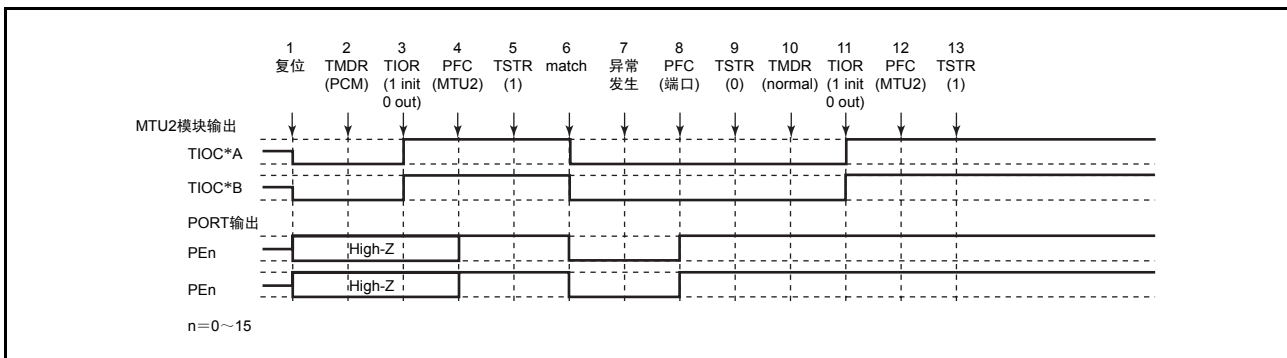


图 9.140 在相位计数模式中发生异常并在正常模式中恢复

1. 在复位后，MTU2 的输出为低电平，端口为高阻抗状态。
2. 必须设定相位计数模式。
3. 必须通过 TIOR 初始化管脚（例中，初始输出为高电平，比较输出为低电平输出）。
4. 必须通过 PFC 设定为 MTU2 的输出。
5. 通过 TSTR 开始计数。
6. 通过产生比较匹配，输出低电平。
7. 发生了异常。
8. 必须通过 PFC 设定端口输出，输出有效电平的反相信号。
9. 通过 TSTR 停止计数。
10. 必须在正常模式中进行设定。
11. 必须通过 TIOR 初始化管脚。
12. 必须通过 PFC 设定为 MTU2 的输出。
13. 通过 TSTR 重新开始。

(18) 在相位计数模式的运行中发生异常并在 PWM 模式 1 中重新开始时的运行

在相位计数模式中发生异常，重新设定后在 PWM 模式 1 中重新开始时的说明如图 9.141 所示。

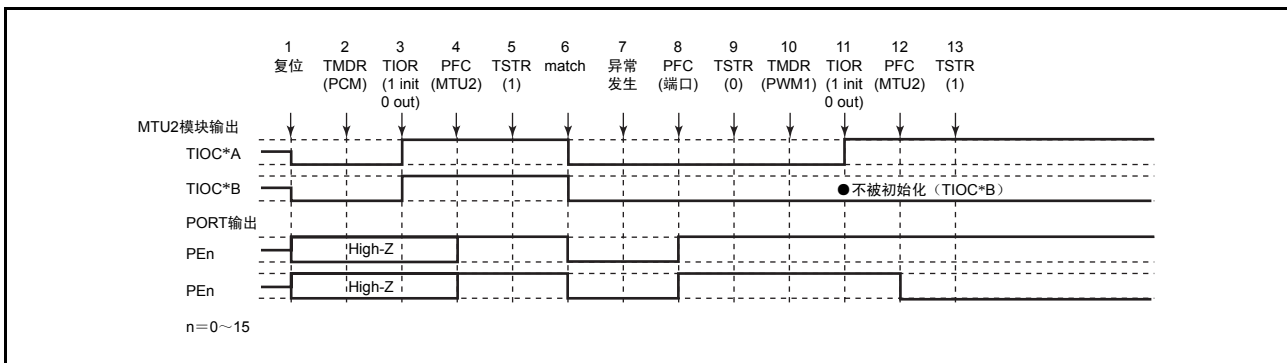


图 9.141 在相位计数模式中发生异常并在 PWM 模式 1 中恢复

1 ~ 9 和图 9.140 通用。

10. 设定 PWM 模式 1。

11. 必须通过 TIOR 初始化管脚（在 PWM 模式 1 中不初始化 TIOC*B）。

12. 必须通过 PFC 设定为 MTU2 的输出。

13. 通过 TSTR 重新开始。

(19) 在相位计数模式的运行中发生异常并在 PWM 模式 2 中重新开始时的运行

在相位计数模式中发生异常，重新设定后在 PWM 模式 2 中重新开始时的说明如图 9.142 所示。

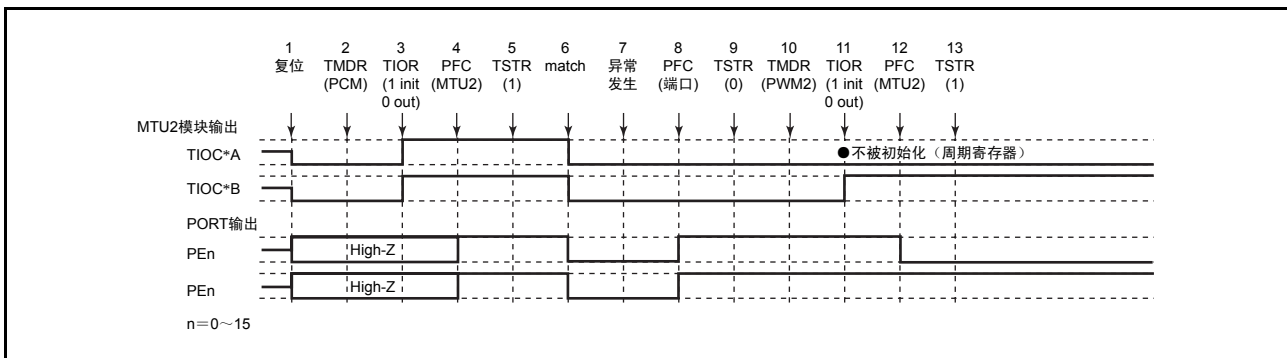


图 9.142 在相位计数模式中发生异常并在 PWM 模式 2 中恢复

1 ~ 9 和图 9.140 通用。

10. 设定 PWM 模式 2。

11. 必须通过 TIOR 初始化管脚（在 PWM 模式 2 中不初始化周期寄存器的管脚）。

12. 必须通过 PFC 设定为 MTU2 的输出。

13. 通过 TSTR 重新开始。

(20) 在相位计数模式的运行中发生异常并在相位计数模式中重新开始时的运行

在相位计数模式中发生异常，重新设定后在相位计数模式中重新开始时的说明如图 9.143 所示。

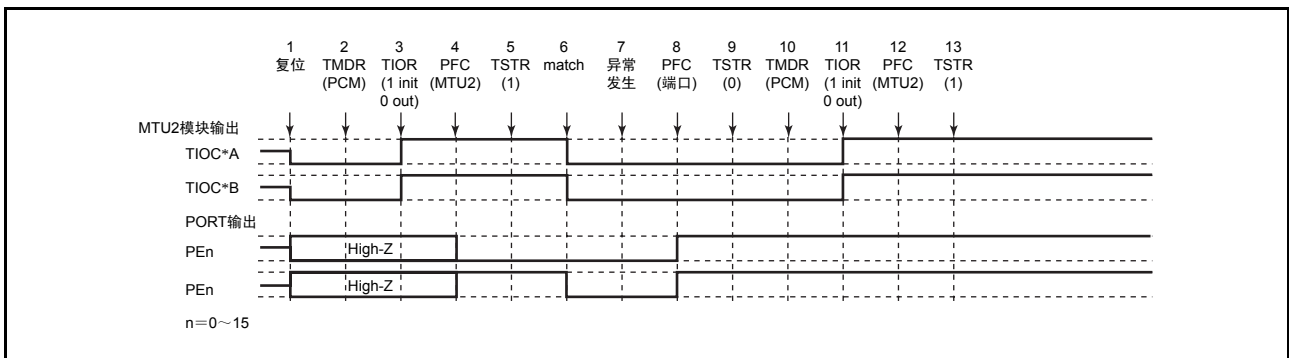


图 9.143 在相位计数模式中发生异常并在相位计数模式中恢复

1 ~ 9 和图 9.140 通用。

10. 在相位计数模式中重新开始时不需要此步骤。

11. 必须通过 TIOR 初始化管脚。

12. 必须通过 PFC 设定为 MTU2 的输出。

13. 通过 TSTR 重新开始。

(21) 在互补 PWM 模式的运行中发生异常并在正常模式中重新开始时的运行

在互补 PWM 模式中发生异常，重新设定后在正常模式中重新开始的说明如图 9.144 所示。

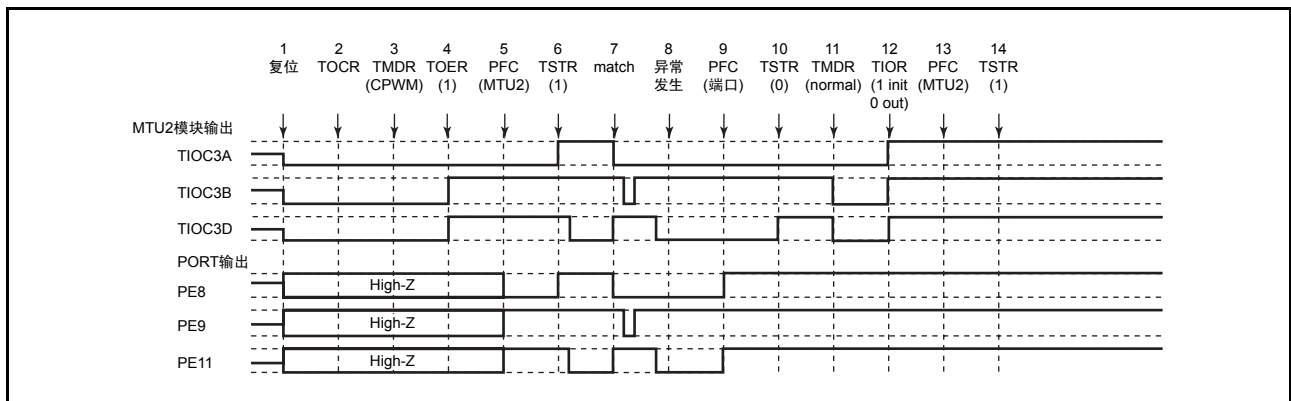


图 9.144 在互补 PWM 模式中发生异常并在正常模式中恢复

1. 在复位后，MTU2 的输出为低电平，端口为高阻抗状态。
2. 必须通过 TOCR 选择互补 PWM 的输出电平并允许或者禁止周期输出。
3. 设定互补 PWM。
4. 必须通过 TOER 允许通道 3、4 的输出。
5. 必须通过 PFC 设定为 MTU2 的输出。
6. 通过 TSTR 开始计数。
7. 通过产生比较匹配，输出互补 PWM 波形。
8. 发生了异常。
9. 必须通过 PFC 设定为端口输出，输出有效电平的反相信号。
10. 通过 TSTR 停止计数（MTU2 的输出为互补 PWM 输出的初始值）。
11. 必须设定正常模式（MTU2 的输出为低电平）。
12. 必须通过 TIOR 初始化管脚。
13. 必须通过 PFC 设定为 MTU2 的输出。
14. 通过 TSTR 重新开始。

(22) 在互补 PWM 模式的运行中发生异常并在 PWM 模式 1 中重新开始时的运行

在互补 PWM 模式中发生异常，重新设定后在 PWM 模式 1 中重新开始时的说明如图 9.145 所示。

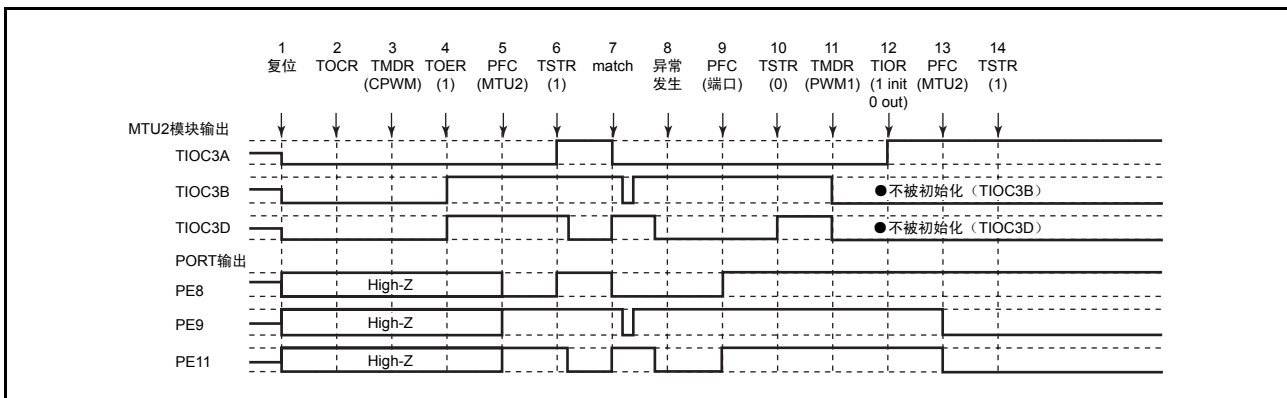


图 9.145 在互补 PWM 模式中发生异常并在 PWM 模式 1 中恢复

1 ~ 10 和图 9.144 通用。

11. 必须设定 PWM 模式 1 (MTU2 的输出为低电平)。

12. 必须通过 TIOR 初始化管脚 (在 PWM 模式 1 中不初始化 TIOC*B)。

13. 必须通过 PFC 设定为 MTU2 的输出。

14. 通过 TSTR 重新开始。

(23) 在互补 PWM 模式的运行中发生异常并在互补 PWM 模式中重新开始时的运行

在互补 PWM 模式中发生异常，重新设定后在互补 PWM 模式中重新开始（从停止计数器时的周期和占空比的设定值重新开始）时的说明如图 9.146 所示。

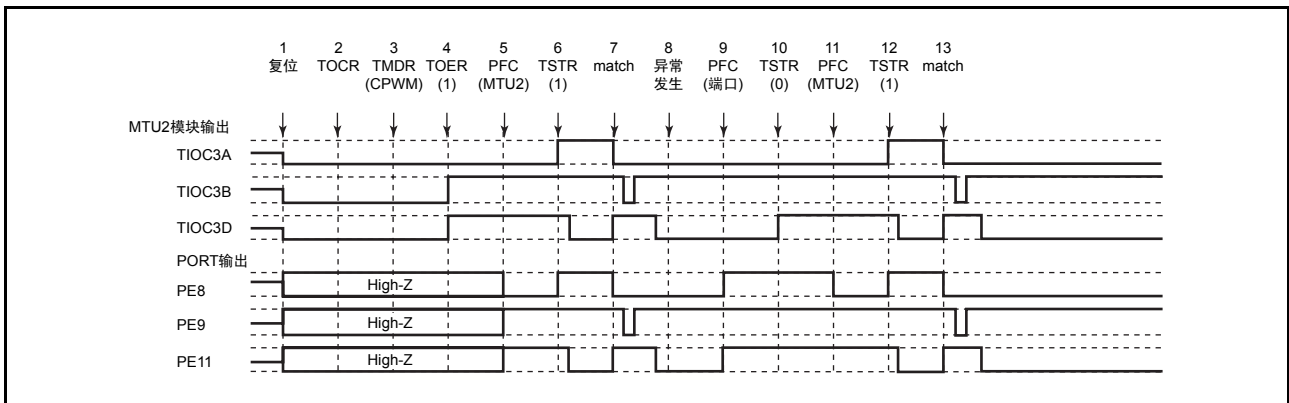


图 9.146 在互补 PWM 模式中发生异常并在互补 PWM 模式中恢复

1 ~ 10 和图 9.144 通用。

11. 必须通过 PFC 设定为 MTU2 的输出。

12. 通过 TSTR 重新开始。

13. 通过产生比较匹配，输出互补 PWM 波形。

(24) 在互补 PWM 模式的运行中发生异常并在互补 PWM 模式中重新开始时的运行

在互补 PWM 模式中发生异常，重新设定后在互补 PWM 模式中重新开始（从周期和占空比的新设定值重新开始）时的说明如图 9.147 所示。

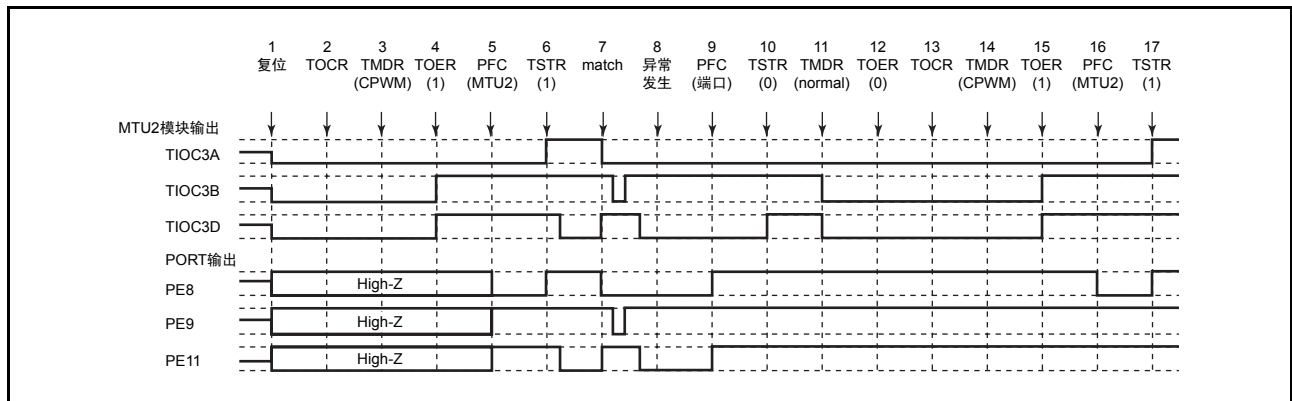


图 9.147 在互补 PWM 模式中发生异常并在互补 PWM 模式中恢复

1 ~ 10 和图 9.144 通用。

11. 必须设定正常模式和新的设定值（MTU2 的输出为低电平）。

12. 必须通过 TOER 禁止通道 3、4 的输出。

13. 必须通过 TOCR 选择互补 PWM 模式的输出电平并允许或者禁止周期输出。

14. 设定互补 PWM。

15. 必须通过 TOER 允许通道 3、4 的输出。

16. 必须通过 PFC 设定为 MTU2 的输出。

17. 通过 TSTR 重新开始。

(25) 在互补 PWM 模式的运行中发生异常并在复位同步 PWM 模式中重新开始时的运行

在互补 PWM 模式中发生异常，重新设定后在复位同步 PWM 模式中重新开始的说明如图 9.148 所示。

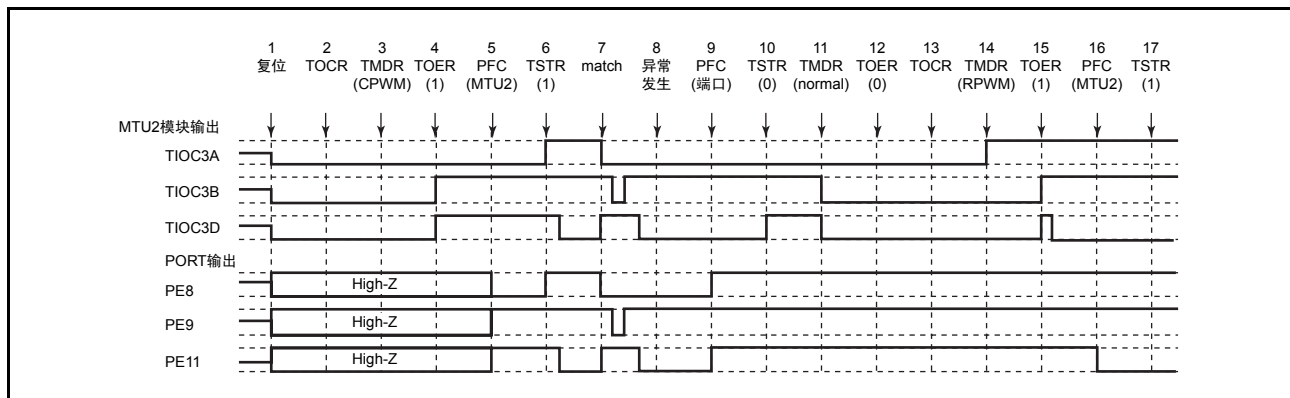


图 9.148 在互补 PWM 模式中发生异常并在复位同步 PWM 模式中恢复

1 ~ 10 和图 9.144 通用。

11. 必须设定正常模式 (MTU2 的输出为低电平)。
12. 必须通过 TOER 禁止通道 3、4 的输出。
13. 必须通过 TOCR 选择复位同步 PWM 模式的输出电平并允许或者禁止周期输出。
14. 设定复位同步 PWM。
15. 必须通过 TOER 允许通道 3、4 的输出。
16. 必须通过 PFC 设定为 MTU2 的输出。
17. 通过 TSTR 重新开始。

(26) 在复位同步 PWM 模式的运行中发生异常并在正常模式中重新开始时的运行

在复位同步 PWM 模式中发生异常，重新设定后在正常模式中重新开始的说明如图 9.149 所示。

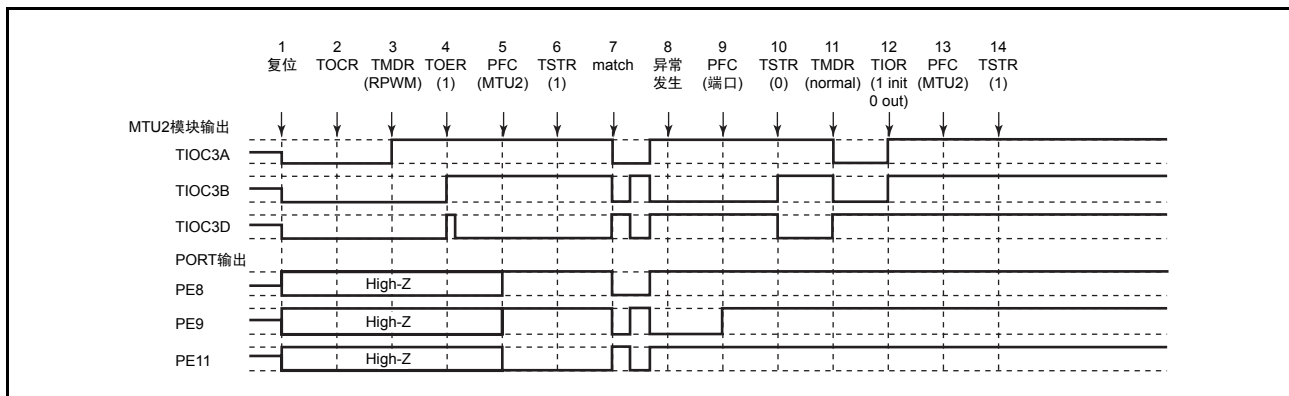


图 9.149 在复位同步 PWM 模式中发生异常并在正常模式中恢复

1. 在复位后，MTU2 的输出为低电平，端口为高阻抗状态。
2. 必须通过 TOCR 选择复位同步 PWM 的输出电平并允许或者禁止周期输出。
3. 设定复位同步 PWM。
4. 必须通过 TOER 允许通道 3、4 的输出。
5. 必须通过 PFC 设定为 MTU2 的输出。
6. 通过 TSTR 开始计数。
7. 通过产生比较匹配，输出复位同步 PWM 波形。
8. 发生了异常。
9. 必须通过 PFC 设定为端口输出，输出有效电平的反相信号。
10. 通过 TSTR 停止计数（MTU2 的输出为复位同步 PWM 输出的初始值）。
11. 必须设定正常模式（MTU2 输出的正相为低电平，反相为高电平）。
12. 必须通过 TIOR 初始化管脚。
13. 必须通过 PFC 设定为 MTU2 的输出。
14. 通过 TSTR 重新开始。

(27) 在复位同步 PWM 模式的运行中发生异常并在 PWM 模式 1 中重新开始时的运行

在复位同步 PWM 模式中发生异常，重新设定后在 PWM 模式 1 中重新开始的说明如图 9.150 所示。

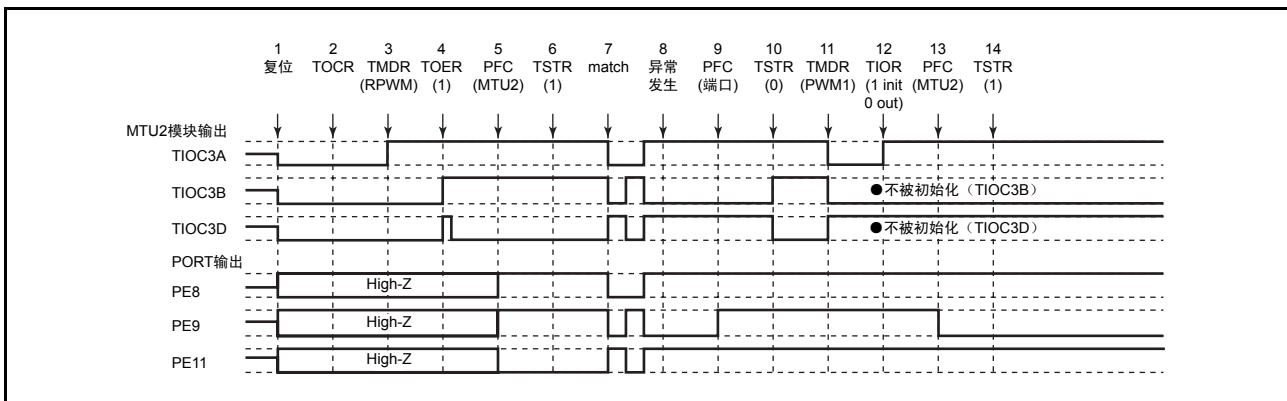


图 9.150 在复位同步 PWM 模式中发生异常并在 PWM 模式 1 中恢复

1 ~ 10 和图 9.149 通用。

11. 必须设定 PWM 模式 1 (MTU2 输出的正相为低电平，反相为高电平)。

12. 通过 TIOR 不初始化管脚 (在 PWM 模式 1 中不初始化 TIOC*B)。

13. 必须通过 PFC 设定为 MTU2 的输出。

14. 通过 TSTR 重新开始。

(28) 在复位同时 PWM 模式的运行发生异常并在互补 PWM 模式中重新开始时的运行

在复位同步 PWM 模式中发生异常，重新设定后在互补 PWM 模式中重新开始的说明如图 9.151 所示。

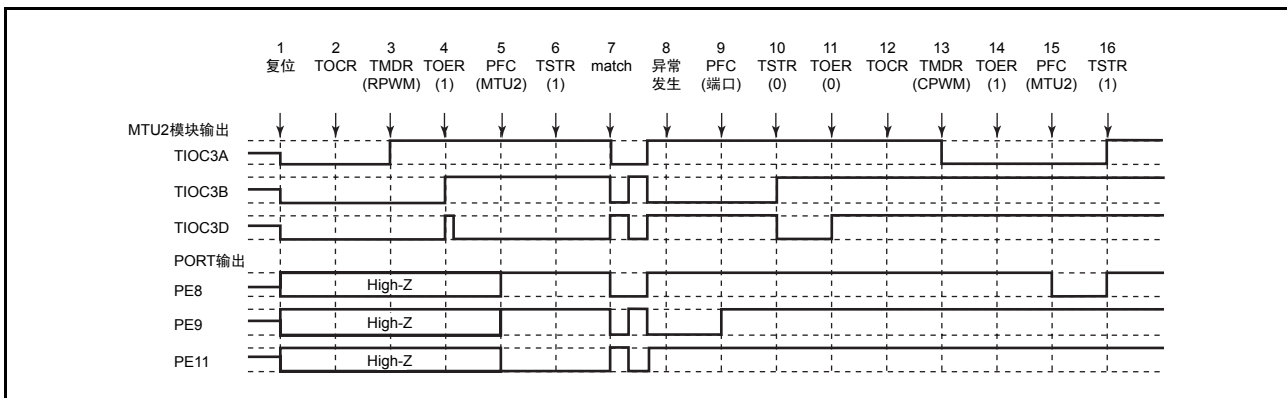


图 9.151 在复位同步 PWM 模式中发生异常并在互补 PWM 模式中恢复

1 ~ 10 和图 9.149 通用。

11. 必须通过 TOER 禁止通道 3、4 的输出。
12. 必须通过 TOCR 选择互补 PWM 的输出电平并允许或者禁周期输出。
13. 设定互补 PWM (MTU2 的周期输出管脚为低电平)。
14. 必须通过 TOER 允许通道 3、4 的输出。
15. 必须通过 PFC 设定为 MTU2 的输出。
16. 通过 TSTR 重新开始。

(29) 在复位同步 PWM 模式的运行中发生异常并在复位同步 PWM 模式中重新开始时的运行

在复位同步 PWM 模式中发生异常，重新设定后在复位同步 PWM 模式中重新开始时的说明如图 9.152 所示。

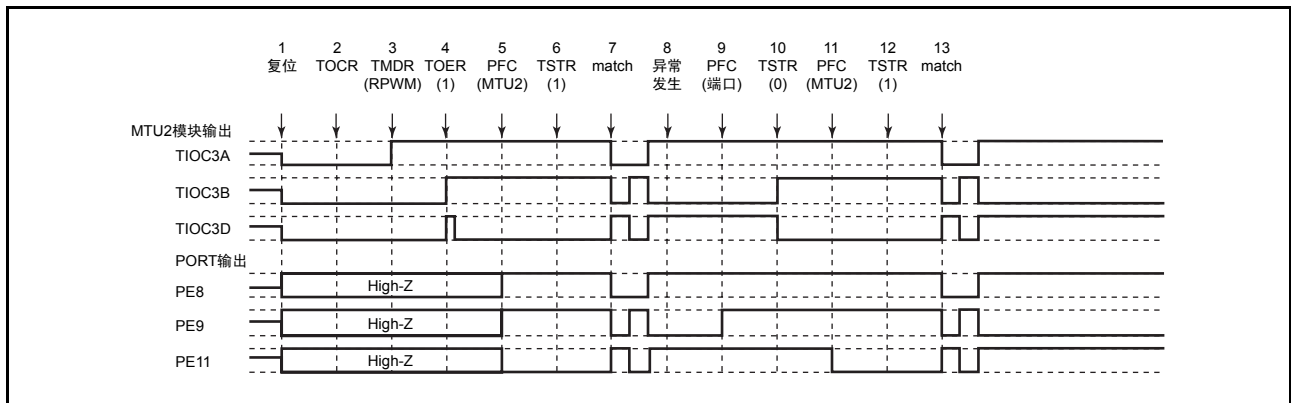


图 9.152 在复位同步 PWM 模式中发生异常并在复位同步 PWM 模式中恢复

1 ~ 10 和图 9.149 通用。

11. 必须通过 PFC 设定为 MTU2 的输出。

12. 通过 TSTR 重新开始。

13. 通过产生比较匹配，输出复位同步 PWM 波形。

第 10 章 端口输出的允许 (POE)

根据 $\overline{\text{POE0}}$ 、 $\overline{\text{POE1}}$ 、 $\overline{\text{POE3}}^*$ 、 $\overline{\text{POE8}}$ 管脚的输入变化、大电流管脚 (MTU2 的 TIOC3B、TIOC3D、TIOC4A、TIOC4B、TIOC4C、TIOC4D 为多路复用管脚) 的输出状态或者寄存器的设定, 端口输出的允许 (POE) 能将大电流管脚和 MTU2 通道 0 的管脚 (TIOC0A、TIOC0B、TIOC0C、TIOC0D 为多路复用管脚) 置为高阻抗状态, 同时能产生中断请求。

10.1 特点

- 能给 $\overline{\text{POE0}}$ 、 $\overline{\text{POE1}}$ 、 $\overline{\text{POE3}}^*$ 、 $\overline{\text{POE8}}$ 的各输入管脚设定下降沿、 $P\phi/8 \times 16$ 次、 $P\phi/16 \times 16$ 次或者 $P\phi/128 \times 16$ 次的低电平采样。
- 能通过 $\overline{\text{POE0}}$ 、 $\overline{\text{POE1}}$ 、 $\overline{\text{POE3}}^*$ 、 $\overline{\text{POE8}}$ 管脚的下降沿或者低电平采样, 将大电流管脚和 MTU2 的通道 0 管脚置为高阻抗状态。
- 如果在和大电流管脚的输出电平进行比较的同时, 有效电平持续输出了 1 个周期以上, 就能将大电流管脚置为高阻抗状态。
- 能通过写 POE 的寄存器将大电流管脚和 MTU2 的通道 0 管脚置为高阻抗状态。
- 可分别通过输入电平的采样或者输出电平的比较结果产生中断。

如图 10.1 的框图所示, POE 由输入电平的检测电路、输出电平的比较电路和高阻抗请求 / 中断请求的生成电路构成。

不同的是: 即使在振荡器停止或者软件待机状态下, 也能将大电流管脚置为高阻抗状态。详细内容请参照“附录 A. 管脚状态”。

【注】 * $\overline{\text{POE3}}$ 只是 SH7125 的管脚。

POE 的框图如图 10.1 所示。

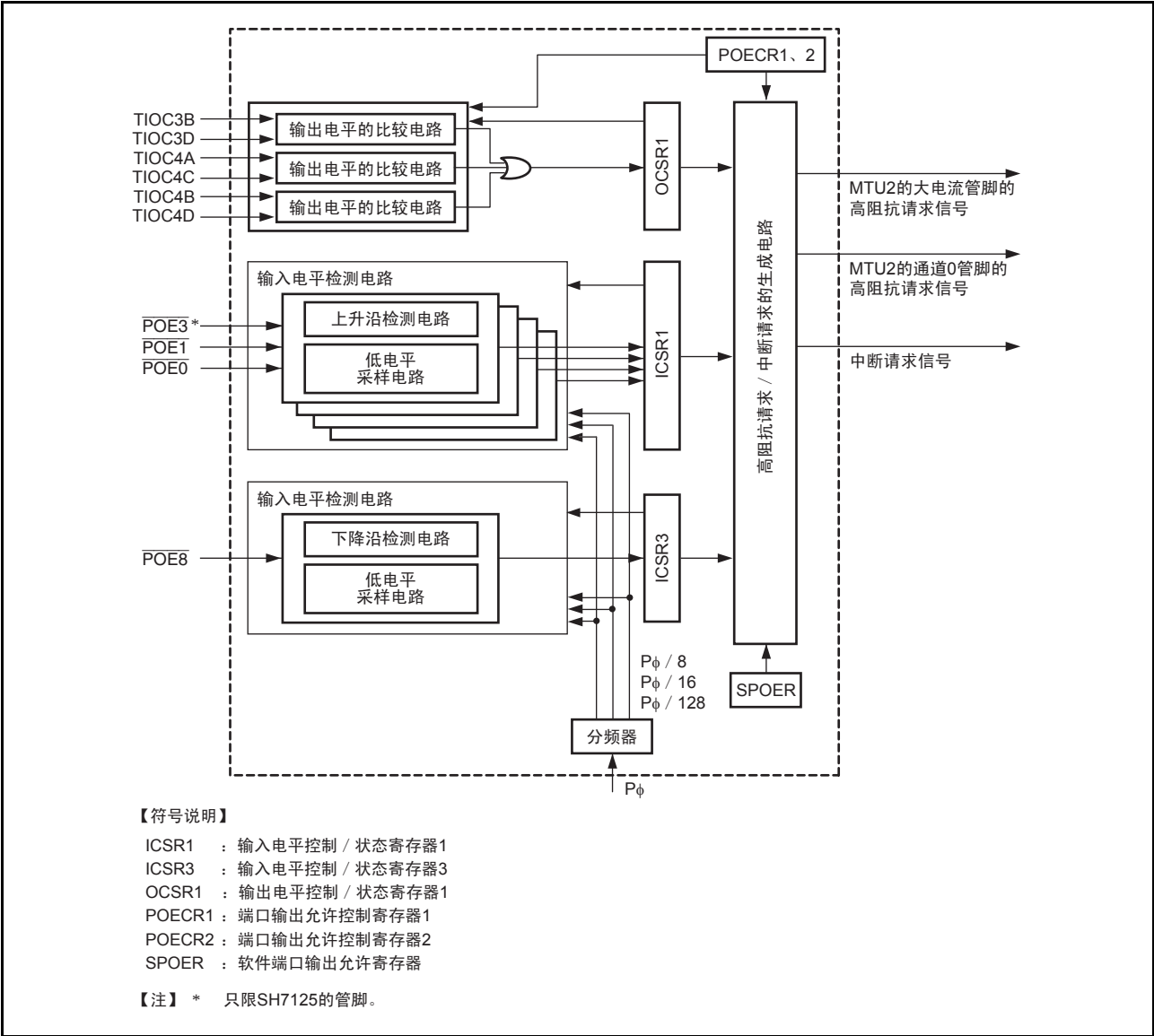


图 10.1 POE 的框图

10.2 输入 / 输出管脚

表 10.1 管脚结构

名称	管脚名	输入 / 输出	功 能
端口输出允许的 输入管脚 0、1、3	$\overline{\text{POE0}}$ 、 $\overline{\text{POE1}}$ 、 $\overline{\text{POE3}}$ *	输入	输入将 MTU2 的大电流管脚置为高阻抗状态的请求信号 *
端口输出允许的 输入管脚 8	$\overline{\text{POE8}}$	输入	输入将 MTU2 的通道 0 管脚置为高阻抗状态的请求信号 *

【注】 * 当通过 PFC 选择了 $\overline{\text{POE3}}$ 功能时，如果管脚没有任何输入，就在内部进行上拉。
 $\overline{\text{POE3}}$ 只是 SH7125 的管脚。

通过表 10.2 所示的管脚组合进行输出电平的比较。

表 10.2 管脚组合

管脚组合	输入 / 输出	功 能
PE9/TIOC3B 和 PE11/TIOC3D	输出	当 2 个管脚的有效电平 (MTU2 定时器的输出控制寄存器 (TOCR) 的输出电平选择 P 位 (OLSP) 是 0 时为低电平输出，是 1 时为高电平输出) 持续输出了 1 个外围时钟 ($P\phi$) 周期以上时，将 MTU2 的大电流管脚置为高阻抗状态。当通过设定管脚功能控制器选择 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	

10.3 寄存器说明

POE 有以下寄存器。有关这些寄存器的地址和各处理模式中的寄存器状态，请参照“第 20 章 寄存器一览”。

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
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 3	ICSR3	R/W	H'0000	H'FFFFD008	8、16、32
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

10.3.1 输入电平控制 / 状态寄存器 1（ICSR1）

ICSR1 是 16 位可读写寄存器，选择 $\overline{\text{POE0}}$ 、 $\overline{\text{POE1}}$ 、 $\overline{\text{POE3}}$ 管脚的输入模式、控制中断的允许 / 禁止以及表示各状态。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	POE3F	-	POE1F	POE0F	-	-	-	PIE1	POE3M[1:0]	-	-	POE1M[1:0]	POE0M[1:0]			
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R	R	R	R/W	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

【注】*1 为了清除标志，只能在读1后写0。

*2 只能在加电复位后写1次。

位	位名	初始值	R/W	说 明
15	POE3F	0	R/(W)*1	POE3 标志位（SH7125 的功能） 表示给 $\overline{\text{POE3}}$ 管脚输入了高阻抗请求。 [清除条件] - 当读 POE3F=1 的状态后，给 POE3F 写 0 时（当通过 ICSR1 的 bit7、6 设定为下降沿时） - 在通过 $P\phi/8$、16、128 时钟对 POE3 输入管脚进行高电平采样时，当读 POE3F=1 的状态后，给 POE3F 写 0 时（当通过 ICSR1 的 bit7、6 设定为低电平采样时） [置位条件] - 当 $\overline{\text{POE3}}$ 管脚有 ICSR1 的 bit7、6 设定的输入时
14	—	0	R/(W)*1	保留位 读写的值总是 0。

位	位名	初始值	R/W	说 明
13	POE1F	0	R/(W)*1	POE1 标志位 表示给 $\overline{\text{POE1}}$ 管脚输入了高阻抗请求。 [清除条件] - 当读 POE1F=1 的状态后, 给 POE1F 写 0 时 (当通过 ICSR1 的 bit3、2 设定为下降沿时) - 在通过 $P\phi/8$、16、128 时钟对 POE1 输入管脚进行高电平采样时, 当读 POE1F=1 的状态后, 给 POE1F 写 0 时 (当通过 ICSR1 的 bit3、2 设定为低电平采样时) [置位条件] - 当 $\overline{\text{POE1}}$ 管脚有 ICSR1 的 bit3、2 设定的输入时
12	POE0F	0	R/(W)*1	POE0 标志位 表示给 $\overline{\text{POE0}}$ 管脚输入了高阻抗请求。 [清除条件] - 当读 POE0F=1 的状态后, 给 POE0F 写 0 时 (当通过 ICSR1 的 bit1、0 设定为下降沿时) - 在通过 $P\phi/8$、16、128 时钟对 POE0 输入管脚进行高电平采样时, 当读 POE0F=1 的状态后, 给 POE0F 写 0 时 (当通过 ICSR1 的 bit1、0 设定为低电平采样时) [置位条件] - 当 $\overline{\text{POE0}}$ 管脚有 ICSR1 的 bit1、0 设定的输入时
11 ~ 9	—	全 0	R	保留位 读写的值总是 0。
8	PIE1	0	R/W	端口中断允许位 1 指定在 ICSR1 的 POE0F ~ POE3F 位中有 1 位被置 1 时是否请求中断。 0: 禁止中断请求 1: 允许中断请求
7、6	POE3M[1:0]	00	R/W*2	POE3 模式位 1、0 (这是 SH7125 的功能。在 SH7124 的情况下必须写 0) 选择 $\overline{\text{POE3}}$ 管脚的输入模式。 00: 在 POE3 输入管脚的下降沿接受请求。 01: 通过 $P\phi/8$ 时钟对 POE3 输入管脚进行 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 POE3 输入管脚进行 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 POE3 输入管脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。
5、4	—	全 0	R/W*2	保留位 读写的值总是 0。
3、2	POE1M[1:0]	00	R/W*2	POE1 模式位 1、0 选择 $\overline{\text{POE1}}$ 管脚的输入模式。 00: 在 POE1 输入管脚的下降沿接受请求。 01: 通过 $P\phi/8$ 时钟对 POE1 输入管脚进行 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 POE1 输入管脚进行 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 POE1 输入管脚进行 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
1、0	POE0M[1:0]	00	R/W*2	POE0 模式位 1、0 选择 POE0 管脚的输入模式。 00: 在 POE0 输入管脚的下降沿接受请求。 01: 通过 $P\phi/8$ 时钟对 POE0 输入管脚进行 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 POE0 输入管脚进行 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 POE0 输入管脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。

【注】 *1 为了清除标志, 只能在读 1 后写 0

*2 在加电复位后, 只能写 1 次。

10.3.2 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 只能在加电复位后写 1 次。

位	位名	初始值	R/W	说 明
15	OSF1	0	R/(W)*1	输出短路的标志位 1 表示在 MTU2 要比较的 3 组 2 相输出中至少有 1 组同时为有效电平。 [清除条件] - 当读 OSF1=1 的状态后, 给 OSF1 写 0 时 [置位条件] - 当 3 组 2 相输出中至少有 1 组同时为有效电平时
14 ~ 10	—	全 0	R	保留位 读写的值总是 0。
9	OCE1	0	R/W*2	输出短路的高阻抗允许位 1 设定在 OCSR1 的 OSF1 位被置位时是否将管脚置为高阻抗状态。 0: 不将管脚置为高阻抗状态。 1: 将管脚置为高阻抗状态。
8	OIE1	0	R/W	输出短路的中断允许位 1 指定在 OCSR1 的 OSF1 位被置位时是否请求中断。 0: 禁止中断请求 1: 允许中断请求
7 ~ 0	—	全 0	R	保留位 读写的值总是 0。

10.3.3 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 管脚的输入模式、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 只能在加电复位后写1次。

位	位名	初始值	R/W	说 明
15 ~ 13	—	全 0	R	保留位 读写的值总是 0。
12	POE8F	0	R/(W)*1	POE8 的标志位 表示给 POE8 管脚输入了高阻抗请求。 [清除条件] - 当读 POE8F=1 的状态后, 给 POE8F 写 0 时 (当通过 ICSR3 的 bit1、0 设定为下降沿时) - 在通过 Pϕ/8、16、128 时钟对 POE8 输入管脚进行高电平采样时, 当读 POE8F=1 的状态后, 给 POE8F 写 0 时 (当通过 ICSR3 的 bit1、0 设定为低电平采样时) [置位条件] - 当 POE8 管脚有 ICSR3 的 bit1、0 设定的输入时
11、10	—	全 0	R	保留位 读写的值总是为 0。
9	POE8E	0	R/W*2	POE8 的高阻抗允许位 设定在 ICSR3 的 POE8F 位被置位时是否将管脚置为高阻抗状态。 0: 不将管脚置为高阻抗状态。 1: 将管脚置为高阻抗状态。
8	PIE3	0	R/W	端口的中断允许位 3(这是 SH7125 的功能。在 SH7124 的情况下必须写 0) 指定在 ICSR3 的 POE8 位被置 1 时是否请求中断。 0: 禁止中断请求 1: 允许中断请求
7 ~ 2	—	全 0	R	保留位 读写的值总是 0。
1、0	POE8M[1:0]	00	R/W*2	POE8 模式位 1、0 选择 POE8 管脚的输入模式。 00: 在 POE8 输入管脚的下降沿接受请求。 01: 通过 P ϕ /8 时钟对 POE8 输入管脚进行 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 输入管脚进行 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 输入管脚进行 16 次低电平采样, 当全部为低电平时, 接受请求。

【注】*1 为了清除标志, 只能在读 1 后写 0。

*2 在加电复位后, 只能写 1 次。

10.3.4 软件端口的输出允许寄存器（SPOER）

SPOER 是 8 位可读写寄存器，对管脚进行高阻抗控制。

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	MTU2 CH0HIZ	MTU2 CH34HIZ
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7 ~ 3	—	全 0	R	保留位 读写的值总是 0。
2	—	0	R/W	保留位 读写的值总是 0。
1	MTU2CH0HIZ	0	R/W	MTU2 通道 0 的高阻抗输出 对 MTU2 通道 0 的管脚进行高阻抗控制。 0: 不置为高阻抗状态。 [清除条件] • 加电复位 • 当读 MTU2CH0HIZ=1 的状态后, 给 MTU2CH0HIZ 写 0 时 1: 置为高阻抗状态。 [置位条件] • 当给 MTU2CH0HIZ 写 1 时
0	MTU2CH34HIZ	0	R/W	MTU2 通道 3、4 的高阻抗输出 对 MTU2 的大电流管脚进行高阻抗控制。 0: 不置为高阻抗状态 [清除条件] • 加电复位 • 当读 MTU2CH34HIZ=1 的状态后, 给 MTU2CH34HIZ 写 0 时 1: 置为高阻抗状态。 [置位条件] • 当给 MTU2CH34HIZ 写 1 时

10.3.5 端口的输出允许控制寄存器 1（POECR1）

POECR1 是 8 位可读写寄存器，对管脚进行高阻抗控制。

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】* 只能在加电复位后写1次。

位	位名	初始值	R/W	说 明
7 ~ 4	—	全 0	R	保留位 读写的值总是 0。
3	MTU2PE3ZE	0	R/W*	MTU2 PE3 的高阻抗允许位 设定在 POE8F 和 MTU2CH0HIZ 位被置位时是否将 MTU2 通道 0 的管脚 PE3/TIOC0D 置为高阻抗状态。 0: 不置为非高阻抗状态。 1: 置为高阻抗状态。
2	MTU2PE2ZE	0	R/W*	MTU2 PE2 的高阻抗允许位 设定在 POE8F 和 MTU2CH0HIZ 位被置位时是否将 MTU2 通道 0 的管脚 PE2/TIOC0C 置为高阻抗状态。 0: 不置为非高阻抗状态。 1: 置为高阻抗状态。
1	MTU2PE1ZE	0	R/W*	MTU2 PE1 的高阻抗允许位 设定在 POE8F 和 MTU2CH0HIZ 位被置位时是否将 MTU2 通道 0 的管脚 PE1/TIOC0B 置为高阻抗状态。 0: 不置为非高阻抗状态。 1: 置为高阻抗状态。
0	MTU2PE0ZE	0	R/W*	MTU2 PE0 的高阻抗允许为 设定在 POE8F 和 MTU2CH0HIZ 位被置位时是否将 MTU2 通道 0 的管脚 PE0/TIOC0A 置为高阻抗状态。 0: 不置为非高阻抗状态。 1: 置为高阻抗状态。

【注】* 在加电复位后，只能写 1 次。

10.3.6 端口的输出允许控制寄存器 2 (POECR2)

POECR2 是 16 位可读写寄存器，对管脚进行高阻抗控制。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	MTU2 P1CZE	MTU2 P2CZE	MTU2 P3CZE	-	-	-	-	-	-	-	-	-	-	-	-
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】* 只能在加电复位后写1次。

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	MTU2P1CZE	1	R/W*	MTU2 端口 1 的输出比较 / 高阻抗允许位 设定是否对 MTU2 的大电流管脚 PE9/TIOC3B 和 PE11/TIOC3D 进行输出电平比较，并且设定在 OCE1 位是 1 时 OSF1 位被置位的情况下或者在 POE0F、POE1F、POE3F 和 TU2CH34HIZ 位中有 1 位被置位的情况下，是否将这些管脚置为高阻抗状态。 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 的大电流管脚 PE12/TIOC4A 和 PE14/TIOC4C 进行输出电平比较，并且设定在 OCE1 位是 1 时 OSF1 位被置位的情况下或者在 POE0F、POE1F、POE3F 和 MTU2CH34HIZ 位中有 1 位被置位的情况下，是否将这些管脚置为高阻抗状态。 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 的大电流管脚 PE13/TIOC4B 和 PE15/TIOC4D 进行输出电平比较，并且设定在 OCE1 位是 1 时 OSF1 位被置位的情况下或者在 POE0F、POE1F、POE3F 和 MTU2CH34HIZ 位中有 1 位被置位的情况下，是否将这些管脚置为高阻抗状态。 0: 不进行输出电平的比较并且不置为高阻抗状态。 1: 进行输出电平的比较并且置为高阻抗状态。
11	—	0	R	保留位 读写的值总是 0。
10 ~ 8	—	全 1	R/W*	保留位 读写的值总是 1。
7 ~ 0	—	全 0	R	保留位 读写的值总是 0。

【注】* 在加电复位后，只能写 1 次。

10.4 运行说明

为高阻抗控制对象的管脚及其条件如表 10.4 所示。

表 10.4 高阻抗控制对象的管脚及其条件

管脚	条件	详细内容
MTU2 的大电流管脚 (PE9/TIOC3B 和 PE11/TIOC3D)	进行输入电平的检测、 输出电平的比较或者 SPOER 寄存器的设定	MTU2P1CZE • ((POE3F+POE1F+POE0F)+ (OSF1 • OCE1)+(MTU2CH34HIZ))
MTU2 的大电流管脚 (PE12/TIOC4A 和 PE14/TIOC4C)	进行输入电平的检测、 输出电平的比较或者 SPOER 寄存器的设定	MTU2P2CZE • ((POE3F+POE1F+POE0F)+ (OSF1 • OCE1)+(MTU2CH34HIZ))
MTU2 的大电流管脚 (PE13/TIOC4B 和 PE15/TIOC4D)	进行输入电平的检测、 输出电平的比较或者 SPOER 寄存器的设定	MTU2P3CZE • ((POE3F+POE1F+POE0F)+ (OSF1 • OCE1)+(MTU2CH34HIZ))
MTU2 通道 0 的管脚 (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 通道 0 的管脚 (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 通道 0 的管脚 (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 通道 0 的管脚 (PE3/TIOC0D)	进行输入电平的检测或者 SPOER 寄存器的设定	MTU2PE3ZE ((POE8F • POE8E)+(MTU2CH0HIZ))

10.4.1 输入电平的检测

当 $\overline{\text{POE0}}$ 、 $\overline{\text{POE1}}$ 、 $\overline{\text{POE3}}^*$ 或者 $\overline{\text{POE8}}$ 管脚发生 ICSR1 设定的输入条件时，将大电流管脚和 MTU2 的 CH0 的管脚置为高阻抗状态。但是，只在选择了大电流管脚和 MTU2 的 CH0 管脚的通用输入 / 输出功能或者 MTU2 功能时才置为高阻抗状态。

(1) 下降沿检测

当 $\overline{\text{POE0}}$ 、 $\overline{\text{POE1}}$ 、 $\overline{\text{POE3}}^*$ 、 $\overline{\text{POE8}}$ 管脚从高电平变为低电平时，将大电流管脚和 MTU2 的 CH0 管脚置为高阻抗状态。

从 $\overline{\text{POE0}}$ 、 $\overline{\text{POE1}}$ 、 $\overline{\text{POE3}}^*$ 、 $\overline{\text{POE8}}$ 的管脚输入到将管脚置为高阻抗状态的时序例子如图 10.2 所示。

【注】 * 只限 SH7125 的管脚。

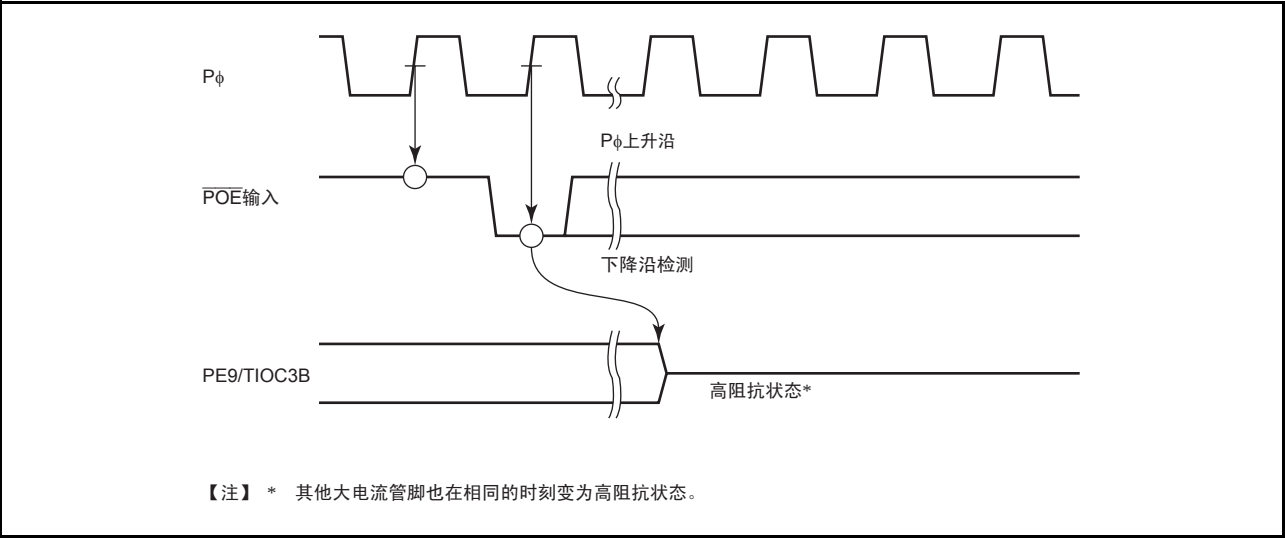


图 10.2 下降沿检测

(2) 低电平检测

低电平检测如图 10.3 所示。通过 ICSR1 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管脚变为高阻抗状态的时序与下降沿检测的情况相同。

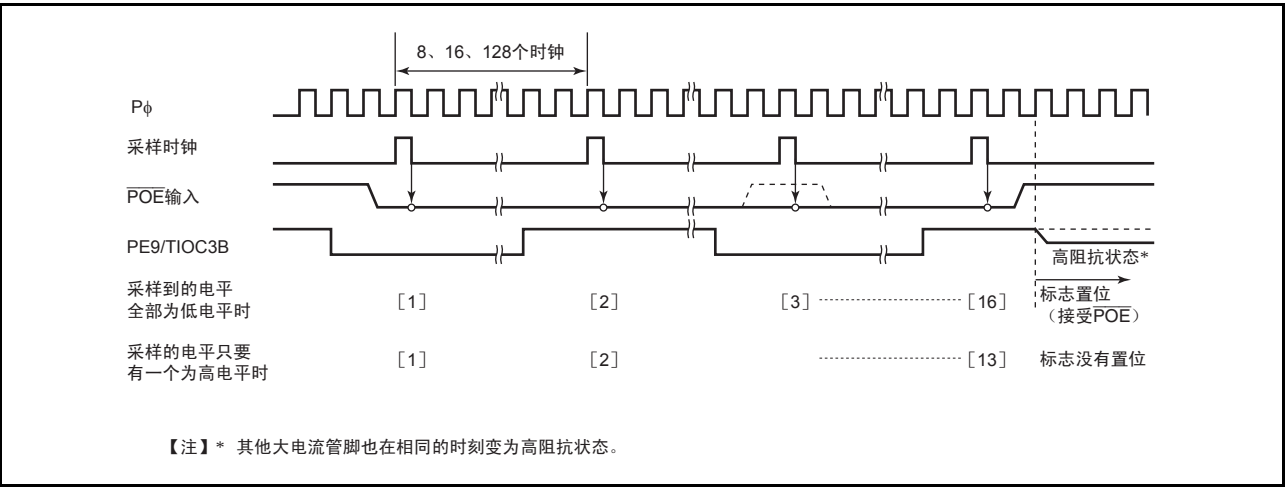


图 10.3 低电平检测

10.4.2 输出电平的比较

以 TI0C3B 和 TI0C3D 的组合为例，输出电平比较的运行如图 10.4 所示。其他管脚的组合也一样。

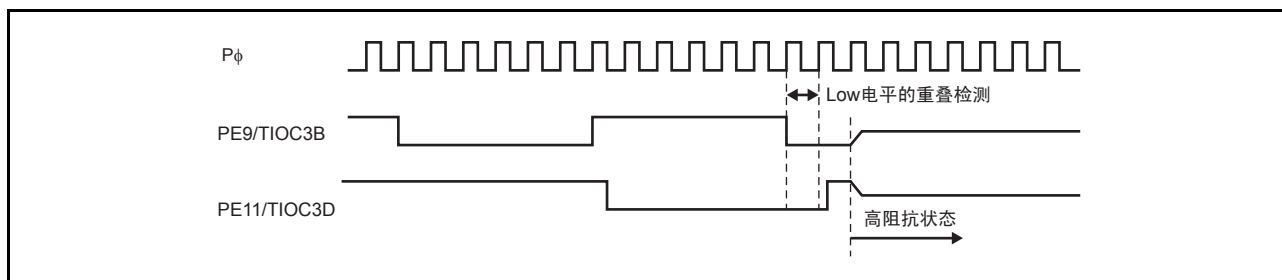


图 10.4 输出电平比较

10.4.3 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管脚通过加电复位恢复到初始状态，或者通过清除 ICSR1 的 POE0F ~ POE3F 和 POE8F 标志（bit12 ~ 15）解除大电流管脚的高阻抗状态。但是，当通过 ICSR1 的 bit0 ~ 7 设定为低电平采样时，如果不是在 POE 管脚输入高电平并进行高电平采样后，即使给标志位写 0 也无效并且不能清除标志。

解除因输出电平检测而变为高阻抗状态的大电流管脚通过加电复位恢复到初始状态，或者通过清除 OCSR1 的 OSF1 标志（bit15）解除大电流管脚的高阻抗状态。但是，如果不是在从大电流管脚输出无效电平后，即使对标志位写 0 也无效并且不能清除标志。能通过设定 MTU2 内的寄存器进行无效电平的输出。

10.5 中断

如果在输入电平检测或者输出电平比较时条件一致，POE 就能在发出中断请求后产生中断。中断的种类以及产生中断请求的条件如表 10.5 所示。

表 10.5 中断请求的种类和条件

名称	中断源	中断标志位	条 件
OEI1	输出的允许中断 1	POE3F、POE1F、POE0F、OSF1	PIE1(POE3F+POE1F+POE0F)+ OIE1 • OSF1
OEI3	输出的允许中断 2	POE8F	PIE3 • POE8F

第 11 章 看门狗定时器 (WDT)

本 LSI 内置看门狗定时器 (WDT)，当因系统失控等原因 CPU 不能改写计数器的值而发生上溢时，能对本 LSI 内部进行复位。

WDT 是 1 个通道的定时器，能以外围时钟为输入时钟，在解除软件待机模式时用作时钟稳定时间的计数，并且还能用作间隔定时器。

11.1 特点

- 能用于确保时钟的稳定时间
在解除软件待机模式时使用。
- 可在看门狗定时器模式和间隔定时器模式之间进行转换
- 在看门狗定时器模式中产生内部复位
因计数器上溢而产生内部复位。
- 在间隔定时器模式中产生中断
因计数器上溢而产生间隔定时器中断。
- 可选择 8 种计数器的输入时钟
可从 8 种分频时钟 ($X \times 1 \sim X \times 1/4096$) 中选择外围时钟。
- 可从加电复位和手动复位中选择复位的种类。

WDT 的框图如图 11.1 所示。

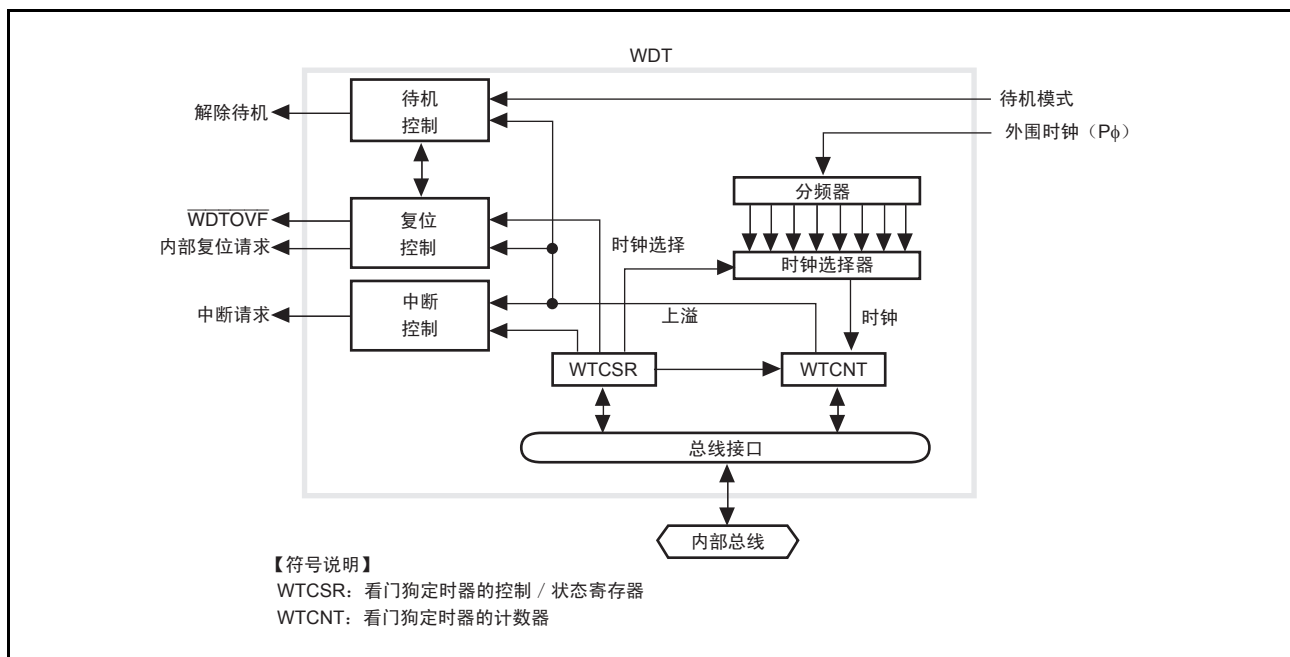


图 11.1 WDT 的框图

11.2 输入 / 输出管脚

WDT 的管脚功能如表 11.1 所示。

表 11.1 WDT 的管脚结构

名称	管脚名	输入 / 输出	功 能
看门狗定时器的上溢	WDTOVF	输出	因在看门狗定时器模式中发生上溢而产生内部复位，在 WTCNT 的 CKS2 ~ CKS0 位设定的 1 个时钟周期内，本管脚输出低电平。

11.3 寄存器说明

WDT 有以下寄存器。有关这些寄存器的地址以及各处理模式中的寄存器状态，请参照“第 20 章 寄存器一览”。

表 11.2 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

11.3.1 看门狗定时器的计数器 (WTCNT)

WTCNT 是 8 位可读写寄存器，也是通过所选时钟进行累加计数的计数器。如果发生上溢，就会在看门狗定时器模式中产生复位，而在间隔定时器模式中产生中断。在因 WDT 的上溢而产生内部复位时，对 WTCNT 计数器进行初始化。但是，只在通过 RES 管脚进行加电复位时，WTCNT 计数器才被初始化为 H'00。

必须在将高位字节置为 H'5A 后以字单位写 WTCNT 计数器，而以字节单位读 WTCNT。

【注】 为了本寄存器不被误改写，写数据的方法和一般寄存器不同，详细内容请参照“11.3.3 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

11.3.2 看门狗定时器的控制 / 状态寄存器 (WTCSR)

WTCSR 是 8 位可读写寄存器，由选择计数时钟的位、上溢标志位和允许位构成。

在因 WDT 的上溢而产生内部复位时，保持 WTCSR 寄存器的值。只在通过 **RES** 管脚进行加电复位时，WTCSR 寄存器才被初期化为 H'00。如果将 WTCSR 计数器用于软件待机解除时的时钟稳定时间的计数，就在计数器发生上溢后仍保持该值。

必须在将高位字节置为 H'A5 后以字单位写 WTCSR 寄存器，而以字节单位读 WTCSR 寄存器。

【注】 为了本寄存器不被误改写，写数据的方法和一般寄存器不同。详细内容请参照“11.3.3 寄存器存取时的注意事项”。

位:	7	6	5	4	3	2	1	0
	TME	WT/IT	RSTS	WOVF	IOVF	CKS[2:0]		
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7	TME	0	R/W	定时器的允许 开始或者停止定时器的运行。如果 WDT 用于解除软件待机模式，就必须将此位设定为 0。 0: 禁止定时器的运行。停止累加计数，保持 WTCNT 的值 1: 允许定时器的运行
6	WT/IT	0	R/W	定时器模式的选择 指定是用作看门狗定时器还是用作间隔定时器。 0: 间隔定时器模式 1: 看门狗定时器模式 【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 发生上溢后产生的复位种类，而在间隔定时器模式中忽视此设定值。 0: 加电复位 1: 手动复位
4	WOVF	0	R/W	看门狗定时器的上溢 表示在看门狗定时器模式中 WTCNT 已发生上溢，而在间隔定时器模式中此位不被置位。 0: 没有发生上溢 1: 在看门狗定时器模式中 WTCNT 已发生上溢
3	IOVF	0	R/W	间隔定时器的上溢 表示在间隔定时器模式中 WTCNT 已发生上溢，而在看门狗定时器模式中此位不被置位。 0: 没有发生上溢 1: 在间隔定时器模式中 WTCNT 已发生上溢

位	位名	初始值	R/W	说 明
2 ~ 0	CKS[2:0]	000	R/W	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: $P\phi$ (6.4μs) 001: $P\phi/4$ (25.6μs) 010: $P\phi/16$ (102.4μs) 011: $P\phi/32$ (204.8μs) 100: $P\phi/64$ (409.6μs) 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。

11.3.3 寄存器存取时的注意事项

为了看门狗定时器的计数器 (WTCNT) 和看门狗定时器的控制 / 状态寄存器不易被改写, 写数据的方法和一般寄存器不同。必须按照以下方法进行写操作:

- 在写 WTCNT 和 WTCSR 时, 必须使用字传送指令, 而字节传送和长字传送指令不能用于写操作。

如图 11.2 所示, 在写 WTCNT 时, 必须将高位字节置为 H'5A、低位字节置为写数据后进行传送; 在写 WTCSR 时, 必须将高位字节置为 H'A5、低位字节置为写数据后进行传送。如此进行传送, 就能将低位字节的数据写到 WTCNT 或者 WTCSR。

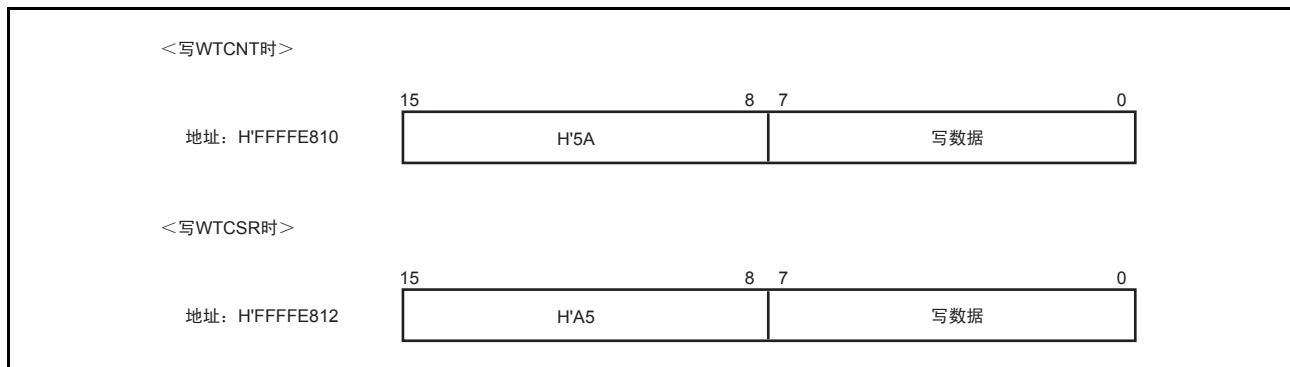


图 11.2 WTCNT 和 WTCSR 的写操作

11.4 运行说明

11.4.1 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， $\overline{\text{RES}}$ 管脚必须保持低电平）。

1. 在进入软件待机模式前，必须将 WTCSR 寄存器的 TME 位设定为 0。如果将 TME 位设定为 1，就有可能在计数上溢时产生不正常的复位或者间隔定时器中断。
2. 预先将要使用的计数时钟的种类和计数器的初始值分别设定到 WTCSR 寄存器 CKS2 ~ CKS0 位和 WTCNT 计数器。在设定这些值时，计数上溢前的时间必须长于时钟振荡稳定时间。
3. 通过执行 SLEEP 指令进入软件待机模式，停止时钟的运行。
4. 如果 NMI 管脚或者 IRQ 管脚的输入电平发生变化，就通过边沿检测开始 WDT 的计数。
5. 如果 WDT 发生计数上溢，CPG 就开始提供时钟，本 LSI 重新开始运行。此时，WTCSR 寄存器的 WOVF 标志不被置位。

11.4.2 看门狗定时器模式的用法

在看门狗定时器模式的运行中，每当计数器发生上溢，就产生由 WTCSR 寄存器的 RSTS 位选择的内部复位，并且 WDTOVF 管脚输出有效信号。

1. 将 WTCSR 寄存器的 WT/IT 位设定为 1，并将复位的种类设定到 RSTS 位、计数时钟的种类设定到 CKS2 ~ CKS0 位、计数器的初始值设定到 WTCNT 计数器。
2. 如果 WTCSR 寄存器的 TME 位被置 1，就在看门狗定时器模式中开始计数。
3. 在看门狗定时器模式的运行中，如果要使计数器不发生上溢，就必须定期改写计数器。
4. 如果计数器发生上溢，WDT 就将 WTCSR 寄存器的 WOVF 标志置 1，在 CKS2 ~ CKS0 位设定的计数器时钟的 1 个周期内 WDTOVF 管脚输出有效信号，并且产生由 RSTS 位指定种类的复位，然后计数器继续计数。

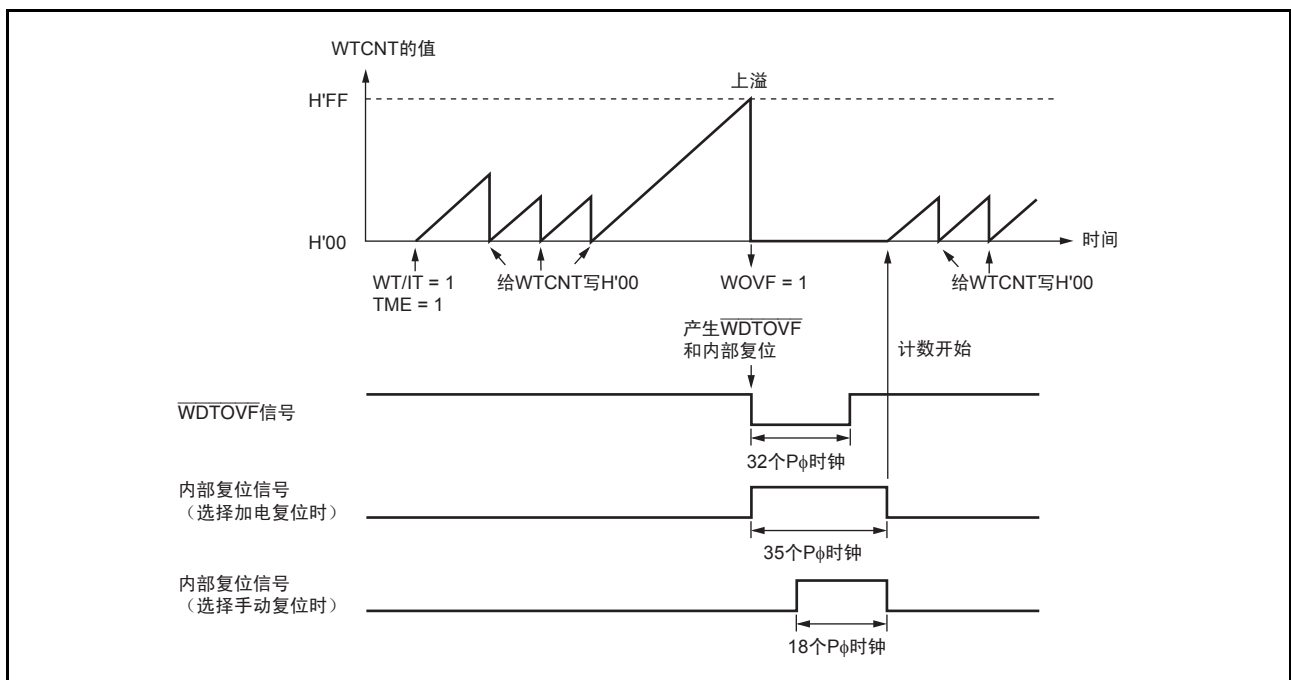


图 11.3 看门狗定时器模式中的运行
（通过 CKS0 ~ CKS2 位将 WTCNT 的计数器时钟设定为 $P\phi/32$ ）

11.4.3 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 WTCSR 寄存器的 WT/IT 位清 0，并将计数时钟的种类设定到 CKS2 ~ CKS0 位、计数器的初始值设定到 WTCNT 计数器。
2. 如果 WTCSR 寄存器的 TME 位被置 1，就在间隔定时器模式中开始计数。
3. 当计数器发生上溢时，WDT 将 WTCSR 寄存器的 IOVF 标志置 1，并将间隔定时器的中断请求传送到 INTC，然后计数器继续计数。

11.5 使用注意事项

在间隔定时器模式中，如果在设定 WTCNT 时写 H'FF，紧接其后的 H'00 就不发生上溢，而在 257 个计数时钟后的 H'FF→H'00 时发生上溢。

在看门狗定时器模式中，如果在设定 WTCNT 时写 H'FF，就在 1 个计数时钟后的 H'FF→H'00 时发生上溢。

第 12 章 串行通信接口 (SCI)

本 LSI 具有 3 个独立通道的串行通信接口 (SCI: Serial Communication Interface)。SCI 能以异步通信和时钟同步通信的 2 种方式进行串行通信, 在异步模式中, 能与 Universal Asynchronous Receiver/Transmitter (UART) 或者 Asynchronous Communication Interface Adapter (ACIA) 等标准异步通信 LSI 进行串行数据的通信。另外, 在异步模式中具有多个处理器之间的串行通信功能 (多处理器通信功能)。

12.1 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 (SH7125: 通道 0 ~ 2、SH7124: 通道 0 ~ 2)
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。
数据长度: 7 位或者 8 位
停止位长: 1 位或者 2 位
奇偶校验: 偶校验、奇校验或者无奇偶校验
可进行多处理器间的通信
接收错误的检测: 检测奇偶校验错误、溢出错误和帧错误
中止的检测: 当发生帧错误时, 能通过直接读 RXD 管脚的电平检测中止。
- 时钟同步模式 (SH7125: 通道 0 ~ 2、SH7124: 通道 0 ~ 2)
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。
数据长度: 8 位
接收错误的检测: 检测溢出错误
- 能进行全双工通信
因为具有独立的发送部和接收部, 所以能同时发送和接收。另外, 因为发送部和接收部都为双缓冲结构, 所以能进行串行数据的连续发送和连续接收。
- 能通过内部波特率发生器选择任意的位速率
- 发送和接收的时钟源可选择波特率发生器的内部时钟或者 SCK 管脚的外部时钟
- 4 种中断源
中断源有发送数据空、发送结束、接收数据满和接收错误共 4 种, 能分别独立请求中断。
- 能设定模块待机模式

SCI 的框图如图 12.1 所示。

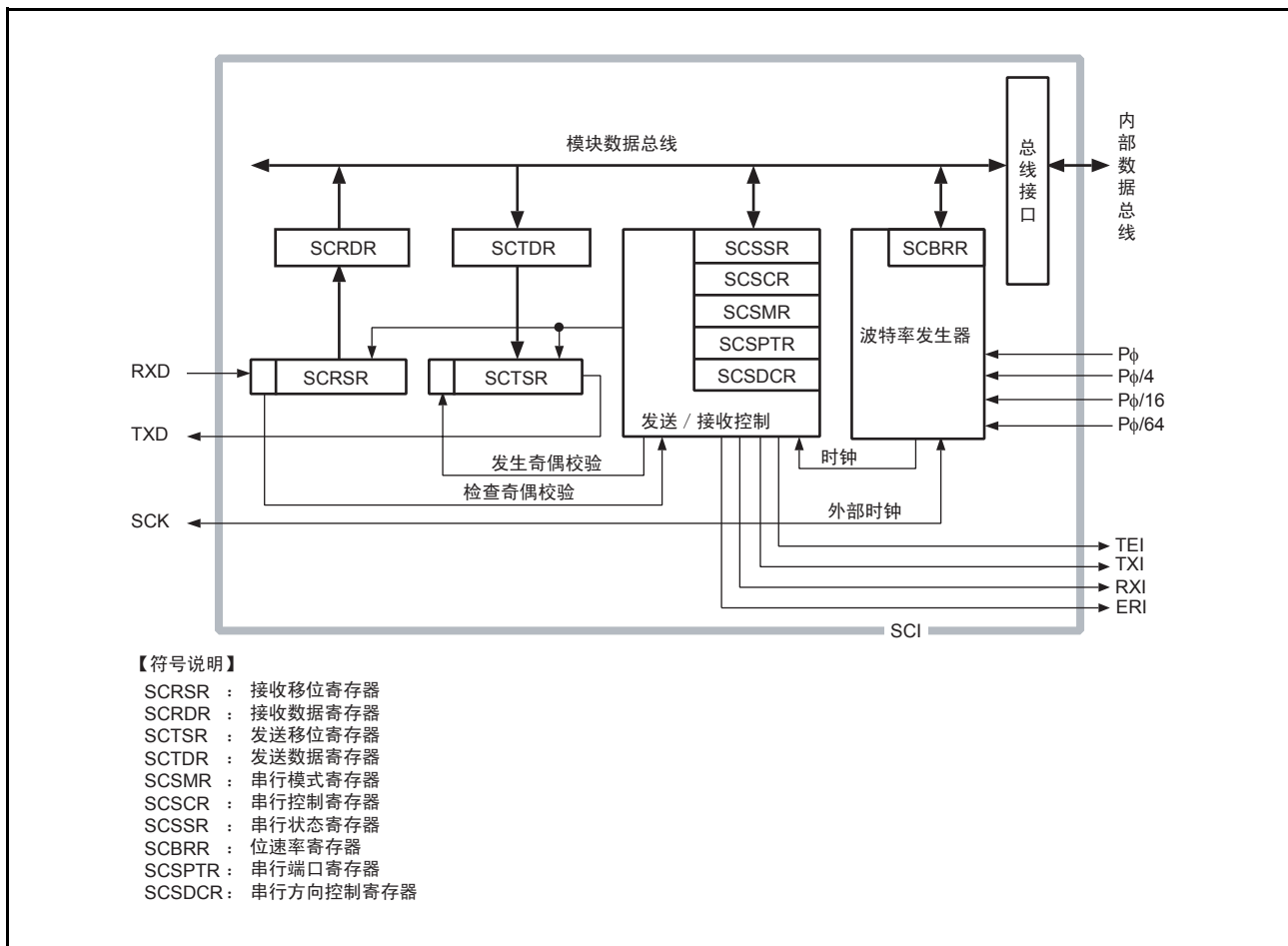


图 12.1 SCI 的框图

12.2 输入 / 输出管脚

SCI 有表 12.1 所示的输入 / 输出管脚。

表 12.1 管脚结构

通道	管脚名 *1	输入 / 输出	功 能
0	SCK0	输入 / 输出	通道 0 的时钟输入 / 输出管脚
	RXD0	输入	通道 0 的接收数据的输入管脚
	TXD0	输出	通道 0 的发送数据的输出管脚
1	SCK1*2	输入 / 输出	通道 1 的时钟输入 / 输出管脚
	RXD1	输入	通道 1 的接收数据的输入管脚
	TXD1	输出	通道 1 的发送数据的输出管脚
2	SCK2	输入 / 输出	通道 2 的时钟输入 / 输出管脚
	RXD2	输入	通道 2 的接收数据的输入管脚
	TXD2	输出	通道 2 的发送数据的输出管脚

【注】 *1 在本文中省略通道，分别略称为 SCK、RXD、TXD。

*2 只为 SH7125 管脚。SH7124 的通道 1 为异步模式专用。

12.3 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 20 章 寄存器一览”。

表 12.2 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

12.3.1 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 管脚输入的串行数据置给 SCRSR, 并转换为并行数据。当结束 1 字节数据的接收时, 数据就自动传送到 SCRDR。

CPU 不能直接读写 SCRSR。

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

12.3.2 接收数据寄存器 (SCRDR)

SCRDR 是保存已接收串行数据的寄存器。

当结束 1 字节串行数据的接收时, SCI 就将接收的串行数据从接收移位寄存器 (SCRSR) 传送到 SCRDR 后保存, 结束接收运行。然后, 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:	-	-	-	-	-	-	-	-

12.3.3 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 就按照从 LSB (bit0) 的顺序将数据送到 TXD 管脚, 进行串行数据的发送。

当结束 1 字节数据的发送时, 就自动将下一个发送数据从 SCTDR 传送到 SCTSR, 开始发送。但是, 在串行状态寄存器 (SCSSR) 的 TDRE 标志被置 1 时, 不将数据从 SCTDR 传送到 SCTSR。

CPU 不能直接读写 SCTSR。

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

12.3.4 发送数据寄存器 (SCTDR)

SCTDR 是保存串行发送数据的 8 位寄存器。

如果检测到发送移位寄存器 (SCTSR) 为空, SCI 就将写到 SCTDR 的发送数据传送到 SCTSR, 开始串行发送。如果在 SCTSR 的串行数据发送中将下一个发送数据写到 SCTDR, 就能连续串行发送。

CPU 随时能读写 SCTDR。

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

12.3.5 串行模式寄存器（SCSMR）

SCSMR 是设定 SCI 的串行通信格式和选择波特率发生器时钟源的 8 位寄存器。

CPU 随时能读 SCSMR。

位:	7	6	5	4	3	2	1	0
	C/A	CHR	PE	O/E	STOP	MP	CKS[1:0]	
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7	C/A	0	R/W	通信模式 从异步模式或者时钟同步模式中选择 SCI 的运行模式。 0: 以异步模式运行 1: 以时钟同步模式运行（SH7124 的通道 1 不能使用）
6	CHR	0	R/W	字符长度 从 7 位或者 8 位数据中选择异步模式的数据长度。在时钟同步模式中 与 CHR 的设定无关，数据长度固定为 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，选择在发送时是否附加奇偶校验位、在接收时是否检查奇 偶校验位。在时钟同步模式中，与 PE 位的设定无关，不附加和检查奇偶 校验位。 0: 禁止奇偶校验位的附加和检查 1: 允许奇偶校验位的附加和检查 * 【注】* 如果将 PE 位设定为 1，就在发送时将 O/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/E 位指定的偶校验或者奇校验。
4	O/E	0	R/W	奇偶校验模式 通过偶校验或者奇校验选择奇偶校验的附加和检查。O/E 位的设定只在异 步模式中将 PE 位设定为 1（允许奇偶校验位的附加和检查）时有效。在 时钟同步模式或者异步模式中，如果禁止奇偶校验的附加和检查，O/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。

位	位名	初始值	R/W	说 明
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，只检查接收的停止位的第 1 位。在停止位的第 2 位为 1 时，作为停止位处理，但是在为 0 时，作为下一个发送字符的起始位处理。 【注】 *1 在发送时，给发送字符的末尾附加 1 位的 1（停止位）后进行发送。 *2 在发送时，给发送字符的末尾附加 2 位的 1（停止位）后进行发送。
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、O$\overline{E}$ 位的设定无效。 0: 禁止多处理器 1: 允许多处理器
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、CKS0 位，从 Pϕ、Pϕ/4、Pϕ/16、Pϕ/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“12.3.10 位速率寄存器（SCBRR）”。 00: Pϕ 钟源 01: Pϕ/4 钟源 10: Pϕ/16 钟源 11: Pϕ/64 钟源 【注】 Pϕ: 外围钟源

12.3.6 串行控制寄存器 (SCSCR)

SCSCR 是进行 SCI 的发送和接收、输出异步模式中的串行时钟、允许或者禁止中断请求，以及选择发送和接收时钟源的寄存器。

CPU 随时能读写 SCSCR。

位:	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7	TIE	0	R/W	发送的中断允许 在将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，并在串行状态寄存器 (SCSSR) 的 TDRE 标志被置 1 时，允许或者禁止发生发送数据空的中断 (TXI) 请求。 能通过在读取 TDRE 标志的 1 后将该位清 0 或者通过将 TIE 位清 0，解除 TXI 中断请求。 0: 禁止发送数据空的中断 (TXI) 请求 1: 允许发送数据空的中断 (TXI) 请求
6	RIE	0	R/W	接收的中断允许 在将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，并在 SCSSR 的 RDRF 标志被置 1 时，允许或者禁止发生接收数据满的中断 (RXI) 请求和接收错误的中断 (ERI) 请求。 能通过在读取 RDRF 标志或者 FER、PER、ORER 标志的 1 后将这些位清 0、或者通过将 RIE 位清 0，解除 RXI 和 ERI 中断请求。 0: 禁止接收数据满的中断 (RXI) 请求和接收错误的中断 (ERI) 请求 1: 允许接收数据满的中断 (RXI) 请求和接收错误的中断 (ERI) 请求
5	TE	0	R/W	发送允许 允许或者禁止开始 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，在将 TE 位设定为 1 前，必须设定串行模式寄存器 (SCSMR)，决定发送格式。
4	RE	0	R/W	接收的允许 允许或者禁止开始 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，在将 RE 位设定为 1 前，必须设定 SCSMR，决定接收格式。

位	位名	初始值	R/W	说 明
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） 如果将此位设定为 1，就跳读多处理器位是 0 的数据，禁止设定 SCSSR 的 RDRF、FER、ORER 的各状态标志。如果接收到多处理器位是 1 的数据，就自动清除此位，返回到通常的接收运行。详细内容请参照“12.4.4 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。 能通过在读取 SCSSR 的 TDRE 标志的 1 后将此位和 TEND 标志清 0、或者通过将 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 SCK 管脚的时钟输出。通过 CKE1 位和 CKE0 位的组合，决定将 SCK 管脚设定为串行时钟输出管脚还是设定为串行时钟输入管脚。 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 位设定为 1 后，设定 CKE1、CKE0 位。有关 SCI 的时钟源选择，请参照“12.4 运行说明”的表 12.14。 异步模式 00：内部时钟 / SCK 管脚为输入管脚（忽视输入信号） 01：内部时钟 / SCK 管脚为时钟输出 *1 10：外部时钟 / SCK 管脚为时钟输入 *2 11：外部时钟 / SCK 管脚为时钟输入 *2 时钟同步模式 00：内部时钟 / SCK 管脚为同步时钟输出 01：内部时钟 / SCK 管脚为同步时钟输出 10：外部时钟 / SCK 管脚为同步时钟输入 11：外部时钟 / SCK 管脚为同步时钟输入 【注】 *1 输出位速率的 16 倍的频率时钟 *2 输入位速率的 16 倍的频率时钟

12.3.7 串行状态寄存器 (SCSSR)

SCSSR 是表示 SCI 运行状态的状态标志的 8 位寄存器。

CPU 随时能读写 SSR。但是，不能给 TDRE、RDRF、ORER、PER、FER 的各标志写 1。另外，为了将这些标志清 0，需要预先读取 1。而且，TEND 标志为只读，不能写。

位:	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
初始值:	1	0	0	0	0	1	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

【注】* 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说 明
7	TDRE	1	R/(W)*	发送数据寄存器空 表示数据已从发送数据寄存器 (SCTDR) 传送到发送移位寄存器 (SCTSR)，并且为能将下一个串行发送数据写到 SCTDR 的状态。 0: 表示已将有效的发送数据写到 SCTDR [清除条件] - 当读 TDRE=1 的状态后写 0 时 1: 表示 SCTDR 中无有效的发送数据 [置位条件] - 当加电复位或者硬件待机模式时 - 当 SCSCR 的 TE 位为 0 时 - 当数据已从 SCTDR 传送到 SCTSR 并且为能将数据写到 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] - 当加电复位或者硬件待机模式时 - 当读 RDRF=1 后写 0 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且将接收数据从 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，就发生溢出错误，丢失接收的数据。

位	位名	初始值	R/W	说 明
5	ORER	0	R/(W)*	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] - 当加电复位或者硬件待机模式时 - 当读 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 的状态下结束到下一个串行接收时 【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] - 在加电复位或者硬件待机模式时 - 当读 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] - 当 SCI 在结束接收时检查接收数据末尾的停止位是否为 1、停止位是否为 0 时 *2 【注】*1 在将 SCSCR 的 RE 位清 0 时, FER 标志不受影响而保持以前的状态。 *2 在 2 个停止位模式中, 只判断第 1 位的停止位是否为 1, 不检查第 2 位的停止位。另外, 将发生帧错误时的接收数据传送到 SCRDR, 但是, RDRF 标志不被置位。另外, 在 FER 标志被置 1 的状态下, 不能继续以后的串行接收。
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] - 在加电复位或者硬件待机模式时 - 当读 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。另外, 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		发送的结束 表示在进行发送字符的最后位时 SCTDR 中无有效数据并结束发送。 TEND 标志为只读，不能写。 0: 表示正在发送 [清除条件] • 当读 TDRE=1 的状态后给 TDRE 标志写 0 时 1: 表示发送已结束 [置位条件] • 在加电复位或者硬件待机模式时 • 当 SCSCR 的 TE 位为 0 时 • 在发送 1 字节的串行发送字符的最后位时 TDRE=1
1	MPB	0		多处理器位 保存接收帧中的多处理器位的值。当 SCSCR 的 RE 为 0 时，此位不变化。
0	MPBT	0		多处理器位的传送 设定给发送帧附加的多处理器位的值。

【注】 * 为了清除标志，只能在读 1 后写 0。

12.3.8 串行端口寄存器 (SCSPTR)

SCSPTR 对串行通信接口 (SCI) 管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 管脚的输出数据，控制串行发送和接收的中止，并且能用 bit3 和 bit2 读取 SCK 管脚的数据以及写 SCK 管脚的输出数据，bit7 控制 RXI 中断的允许和禁止。SCSPTR 为 8 位寄存器，能随时由 CPU 读写。另外，读 SCI 管脚的位时，必须使用端口寄存器。详细内容请参照“第 16 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 此位设定是允许还是禁止 RXI 中断。当 EIO 位是 1 时，即使将 RIE 位置 1，也不向 CPU 请求 RXI 中断。 0: RIE 位允许或者禁止 RXI 和 ERI 中断 当 RIE 位是 1 时，将 RXI 和 ERI 中断送给 INTC 1: 当 RIE 位是 1 时，只将 ERI 中断送给 INTC
6 ~ 4	—	全 0	—	保留位 读写的值总是 0。
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 管脚的输入 / 输出。实际上在 SCK 管脚作为端口输出管脚并输出由 SPB1DT 位设定的值时，必须将 SCSMR 的 C/A 位和 SCSCR 的 CKE1、CKE0 位设定为 0。 0: 不将 SPB1DT 位的值输出到 SCK 管脚 1: 将 SPB1DT 位的值输出到 SCK 管脚

位	位名	初始值	R/W	说 明																				
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 管脚的输出数据。通过 SPB1IO 位指定输出是否有效（详细内容请参照 SPB1IO 位的说明）。输出时，SPB1DT 位的值被输出到 SCK 管脚。 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 与 SPB0DT 位、SCSCR 的 TE 位一起控制 TXD 管脚。																				
0	SPB0DT	不定	R/W	串行端口的中止数据 与 SPB0IO 位、SCSCR 的 TE 位一起控制 TXD 管脚。但是，需要预先通过 PFC（管脚功能控制器）选择 TXD 管脚功能。 <table border="1"><thead><tr><th>SCSCR 的 TE 位的设定值</th><th>SPB0IO 位的设定值</th><th>SPB0DT 位的设定值</th><th>TXD 管脚状态</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>*</td><td>禁止设定（初始状态）</td></tr><tr><td>0</td><td>1</td><td>0</td><td>输出低电平</td></tr><tr><td>0</td><td>1</td><td>1</td><td>输出高电平</td></tr><tr><td>1</td><td>*</td><td>*</td><td>按照串行核心逻辑输出发送数据</td></tr></tbody></table> 【注】* Don't care	SCSCR 的 TE 位的设定值	SPB0IO 位的设定值	SPB0DT 位的设定值	TXD 管脚状态	0	0	*	禁止设定（初始状态）	0	1	0	输出低电平	0	1	1	输出高电平	1	*	*	按照串行核心逻辑输出发送数据
SCSCR 的 TE 位的设定值	SPB0IO 位的设定值	SPB0DT 位的设定值	TXD 管脚状态																					
0	0	*	禁止设定（初始状态）																					
0	1	0	输出低电平																					
0	1	1	输出高电平																					
1	*	*	按照串行核心逻辑输出发送数据																					

12.3.9 串行方向控制寄存器（SCSDCR）

SCSDCR 通过 DIR 位选择 LSB 先或者 MSB 先。在 8 位长的情况下，不论那种串行通信模式都能选择 LSB 先或者 MSB 先。

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

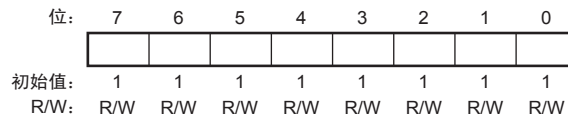
位	位名	初始值	R/W	说 明
7 ~ 4	—	全 1	R	保留位 读写的值总是 1。
3	DIR	0	R/W	数据的传送方向 选择串行或者并行转换格式。发送 / 接收格式在 8 位时有效。 0：以 LSB 先发送 SCTDR 的内容 以 LSB 先将接收数据保存到 SCRDR 1：以 MSB 先发送 SCTDR 的内容 以 MSB 先将接收数据保存到 SCRDR
2	—	0	R	保留位 读写的值总是 0。
1	—	1	R	保留位 读写的值总是 1。
0	—	0	R	保留位 读写的值总是 0。

12.3.10 位速率寄存器 (SCBRR)

SCBRR 是 8 位寄存器，根据串行模式寄存器 (SCSMR) 的 CKS1、CKS0 位选择的波特率发生器的运行时钟，设定串行发送和接收的位速率。

CPU 随时能读 SCBRR。

通过以下计算式求 SCBRR 的设定值：



(异步模式)

$$N = \frac{P_{\phi}}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

(时钟同步模式)

$$N = \frac{P_{\phi}}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: 位速率 (bit/s)

N: 波特率发生器的 SCBRR 设定值 ($0 \leq N \leq 255$)

(必须设定满足电特性的值)

P_{ϕ} : 外围模块的运行频率 (MHz)

n: 波特率发生器输入时钟 (n=0、1、2、3)

(n 和时钟的关系请参照表 12.3)。

表 12.3 SCSMR 的设定值

n	时钟	SCSMR 的设定值	
		CKS1	CKS0
0	P_{ϕ}	0	0
1	$P_{\phi}/4$	0	1
2	$P_{\phi}/16$	1	0
3	$P_{\phi}/64$	1	1

通过以下计算式求异步模式的位速率误差：

$$\text{误差 (\%)} = \left\{ \frac{P_{\phi} \times 10^6}{(N+1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

异步模式的 SCBRR 的设定例子如表 12.4 ~ 表 12.6 所示，时钟同步模式的 SCBRR 的设定例子如表 12.7 ~ 表 12.9 所示。

表 12.4 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 12.5 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 12.6 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 12.7 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 12.8 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 12.9 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 12.10 所示。另外，输入外部时钟时的最大位速率如表 12.11 和表 12.12 所示。

表 12.10 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 12.11 输入外部时钟时的最大位速率（异步模式）

P ϕ (MHz)	输入外部时钟 (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 12.12 输入外部时钟时的最大位速率 (时钟同步模式)

P _φ (MHz)	输入外部时钟 (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

12.4 运行说明

12.4.1 概要

SCI 能通过异步模式（以字符为单位边同步边通信）和时钟同步模式（通过时钟脉冲边同步边通信）的 2 种方式进行串行通信。

如表 12.13 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 12.14 所示，SCI 的时钟源由 SCSMR 的 $\overline{C/A}$ 位和串行控制寄存器（SCSCR）的 CKE1、CKE0 位的组合决定。

(1) 异步模式（SH7125：通道 0 ~ 2、SH7124：通道 0 ~ 2）

- 数据长度：可从 7 位或者 8 位中选择
- 可选择奇偶校验的附加、1 位或 2 位的停止位的附加（根据这些组合，决定发送 / 接收格式和字符长）
- 在接收时，能检测帧错误、奇偶校验错误、溢出错误和中止
- SCI 的时钟源：可从内部时钟或者外部时钟中选择
 选择内部时钟：能通过波特率发生器的时钟运行 SCI，输出位速率的 16 倍频率时钟
 选择外部时钟：需要输入位速率的 16 倍频率时钟（不使用内部波特率发生器）

(2) 时钟同步模式（SH7125：通道 0 ~ 2、SH7124：通道 0 ~ 2）

- 发送 / 接收格式：固定为 8 位数据
- 在接收时，能检测溢出错误
- SCI 的时钟源：可从内部时钟或者外部时钟中选择
 选择内部时钟：通过波特率发生器的时钟运行 SCI，将同步时钟输出到外部
 选择外部时钟：不使用内部波特率发生器而通过输入的同步时钟运行

表 12.13 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		数据长度	奇偶校验位	停止位长
$\overline{C/A}$	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 12.14 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 管脚的功能
	bit7	bit1			
	C/ $\overline{A}$	CKE1			
0	0	0	异步模式	内部	SCI 不使用 SCK 管脚
		1			输出位速率的 16 倍频率时钟
	1	0		外部	输入位速率的 16 倍频率时钟
		1			
1	0	0	时钟同步模式	内部	输出同步时钟
		1			
	1	0		外部	输入同步时钟
		1			

12.4.2 异步模式中的运行

异步模式以字符为单位边同步边进行串行通信的模式，将表示开始通信的起始位和表示结束通信的停止位附加到数据后进行字符的发送和接收。

因为在 SCI 内部发送部和接收部独立，所以能进行全双工通信。另外，因为发送部和接收部都为双缓冲结构，能在发送和接收中读写数据，所以能连续发送和接收。

异步串行通信的一般格式如图 12.2 所示。

在异步串行通信中，通信线路通常保持为标记状态（高电平）。SCI 监视通信线路，将空闲状态（低电平）视为起始位，开始串行通信。

以起始位（低电平）开始，数据（LSB 先：从最低位开始）、奇偶校验位（高 / 低电平），最后为停止位（高电平）的顺序构成串行通信的 1 个字符。

在异步模式中，SCI 在接收时通过起始位的下降沿取得同步。另外，因为 SCI 在 1 个为 16 倍位速率频率的第 8 个时钟采样数据，所以在各位的中央取得通信数据。

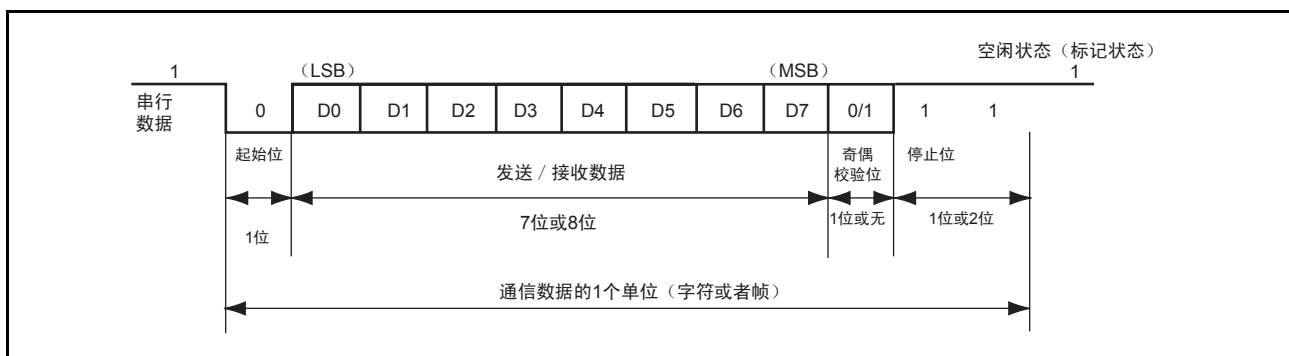


图 12.2 异步通信的数据格式
(8 位数据 / 有奇偶校验 / 2 个停止位的例子)

(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 12.15 所示。

发送 / 接收格式有 12 种，能通过设定串行模式寄存器 (SCSMR) 进行选择。

表 12.15 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、CKE0 位, 从内部波特率发生器生成的内部时钟或者 SCK 管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 12.14。

在将外部时钟输入到 SCK 管脚时, 必须输入使用的位速率的 16 倍频率时钟。

在通过内部时钟运行时, 能从 SCK 管脚输出时钟。此时, 输出的时钟频率为位速率的 16 倍。

(3) 数据的发送和接收运行

• SCI 的初始化 (异步模式)

在发送和接收数据前, 必须首先将 SCSCR 的 TE 位和 RE 位清 0, 然后按照以下顺序初始化 SCI。

在更改运行模式和通信格式等时, 也必须在将 TE 位和 RE 位清 0 后按照以下步骤进行。如果将 TE 位清 0, TDRE 标志就被置 1, 发送移位寄存器 (SCTSR) 被初始化。必须注意: 即使将 RE 位清 0, 也保持 RDRF、PER、FER、ORER 的各标志和接收数据寄存器 (SCRDR) 的内容。在使用外部时钟时, 不能在含有初始化的运行中停止时钟, 否则运行不稳定。

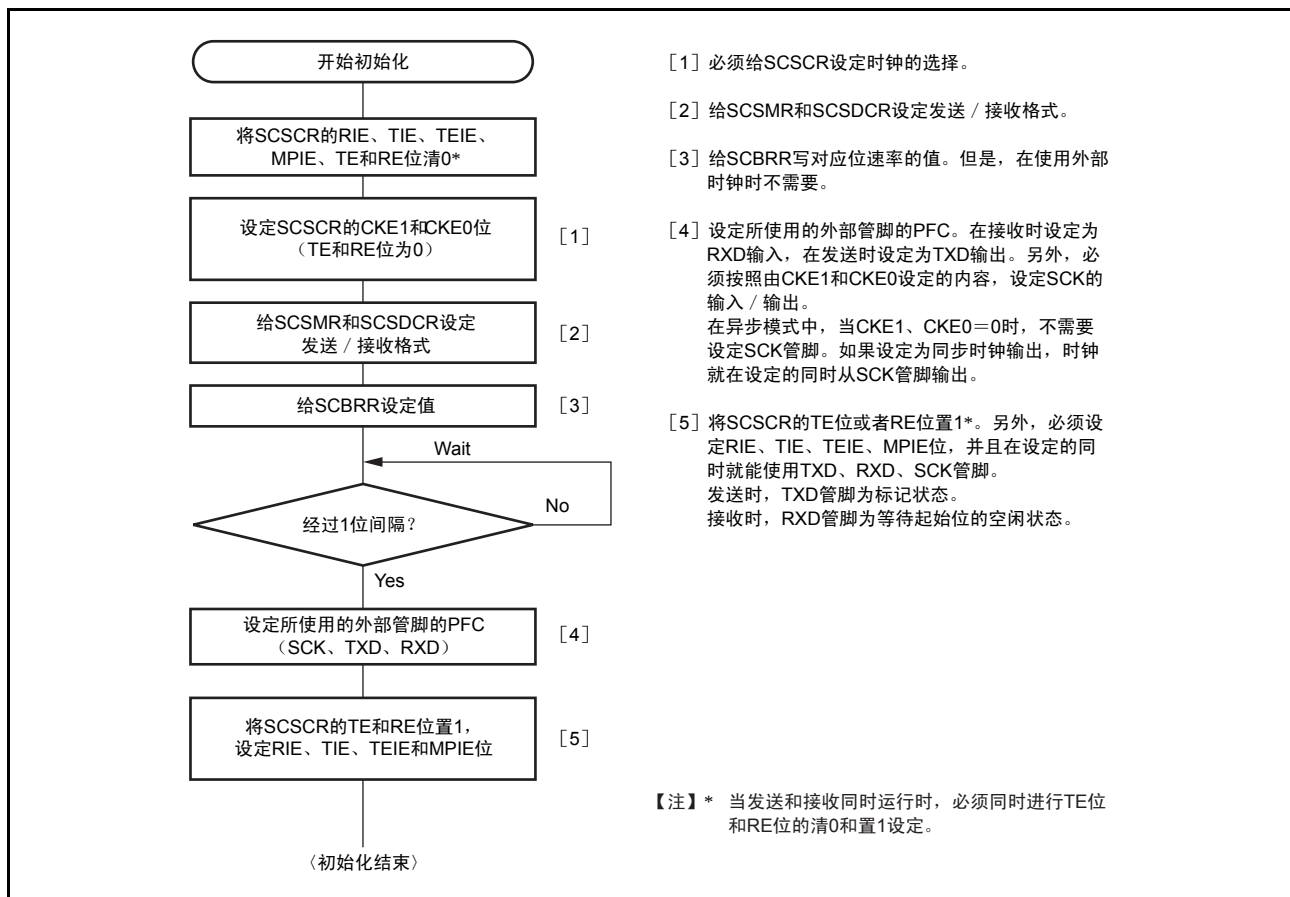


图 12.3 SCI 的初始化流程图例子

- 串行数据的发送（异步模式）

串行发送的流程图例子如图 12.4 所示。

必须按照以下步骤，在将 SCI 设定为可发送的运行状态后发送串行数据。

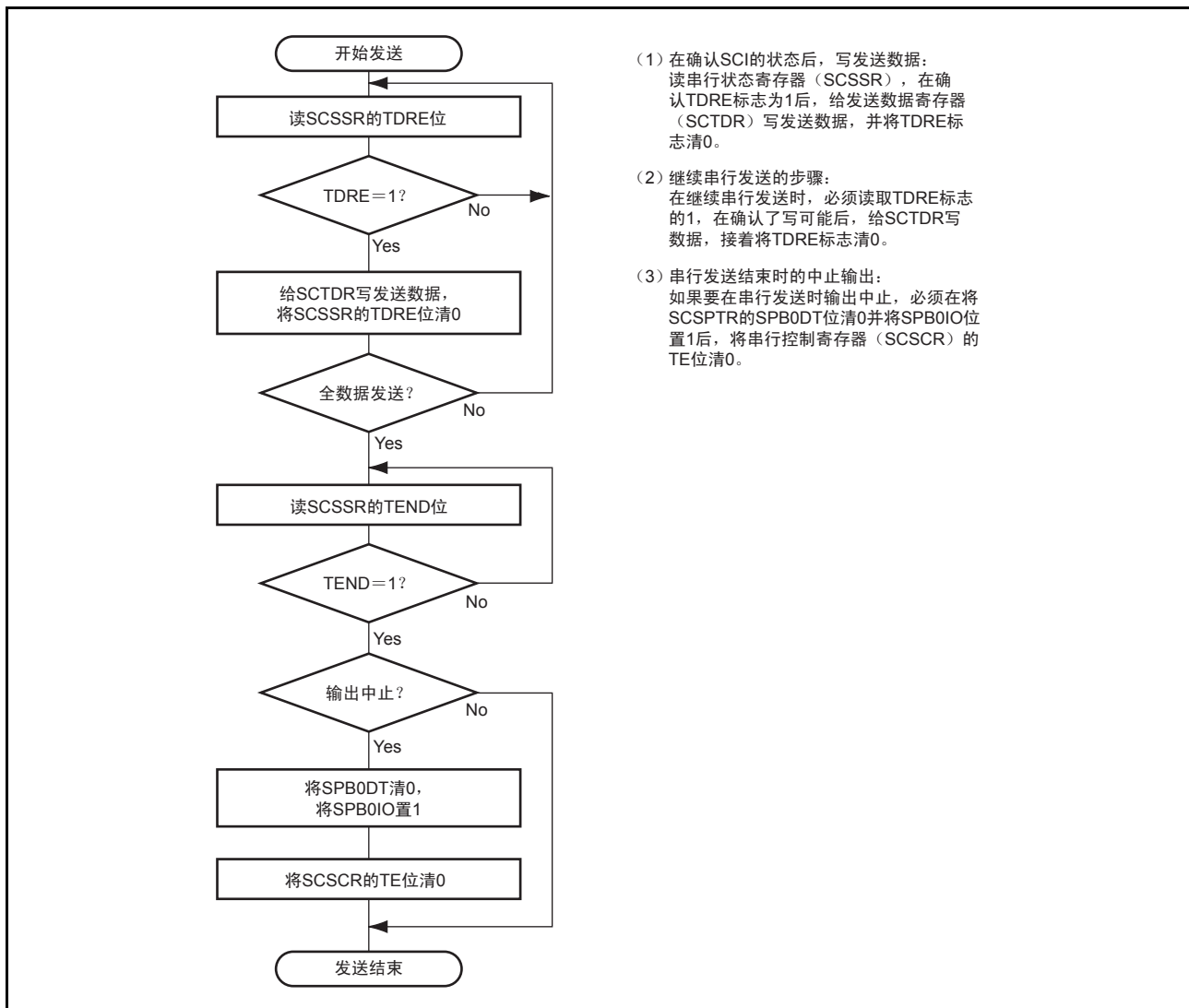


图 12.4 串行发送的流程图例子

串行发送时的 SCI 运行如下：

- 1.SCI 监视串行状态寄存器 (SCSSR) 的 TDRE 标志，如果该位为 0，就认为已将数据写到发送数据寄存器 (SCTDR)，将数据从 SCTDR 传送到发送移位寄存器 (SCTSR)。
2. 在数据从 SCTDR 传送到 SCTSR 后，将 TDRE 标志设定为 1，开始发送。
此时，如果串行控制寄存器 (SCSCR) 的 TIE 位已被置 1，就产生发送数据空的中断 (TXI) 请求。

按照以下顺序从 TXD 管脚送出串行发送数据：

- (a) 起始位：输出 1 位的 0。
- (b) 发送数据：从 LSB 按顺序输出 8 位或者 7 位的数据。
- (c) 奇偶校验位或者多处理器位：输出 1 位的奇偶校验位（偶校验或者奇校验）或者 1 位的多处理器位。
另外，也能选择不输出奇偶校验位或者多处理器位的格式。
- (d) 停止位：输出 1 位或者 2 位的 1（停止位）。
- (e) 标记状态：连续输出 1 直到送出开始下一个发送的起始位为止。

- 3.SCI 在送出停止位时检查 TDRE 标志。

如果 TDRE 标志为 0，就将数据从 SCTDR 传送到 SCTSR，在送出停止位后，开始下一帧的串行发送。

如果 TDRE 标志为 1，就将串行状态寄存器 (SCSSR) 的 TEND 位设定为 1，在送出停止位后变为输出 1 的标记状态。此时，如果 SCSCR 的 TEIE 标志已被置 1，就产生 TEI 中断请求。

异步模式发送时的运行例子如图 12.5 所示。

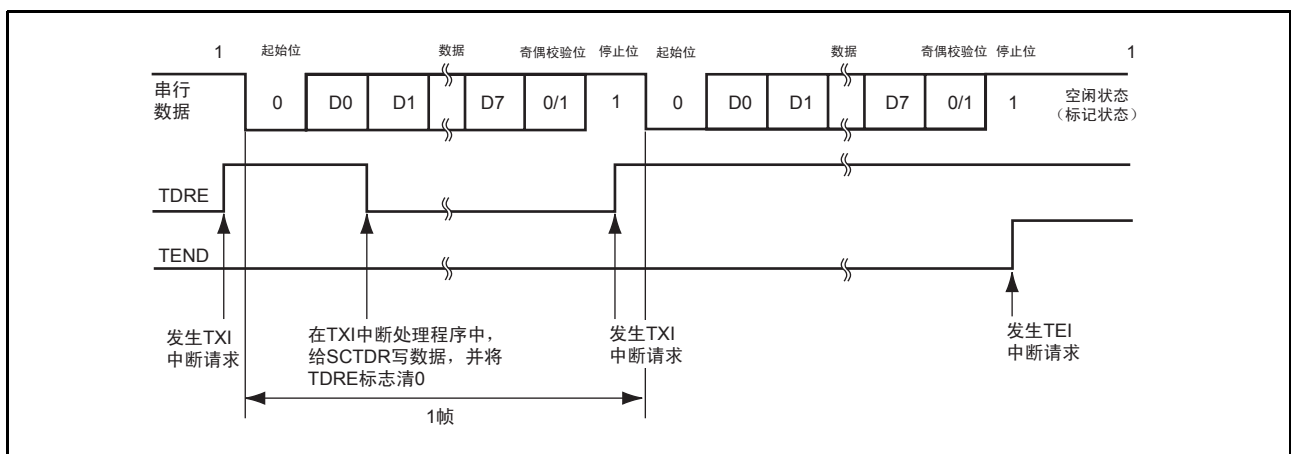


图 12.5 异步模式发送时的运行例子
(8 位数据 / 有奇偶校验 / 1 个停止位的例子)

- 串行数据的接收（异步模式）

串行接收的流程图例子如图 12.6 和图 12.7 所示。

必须按照以下步骤，在将 SCI 设定为可接受的运行状态后接收串行数据。

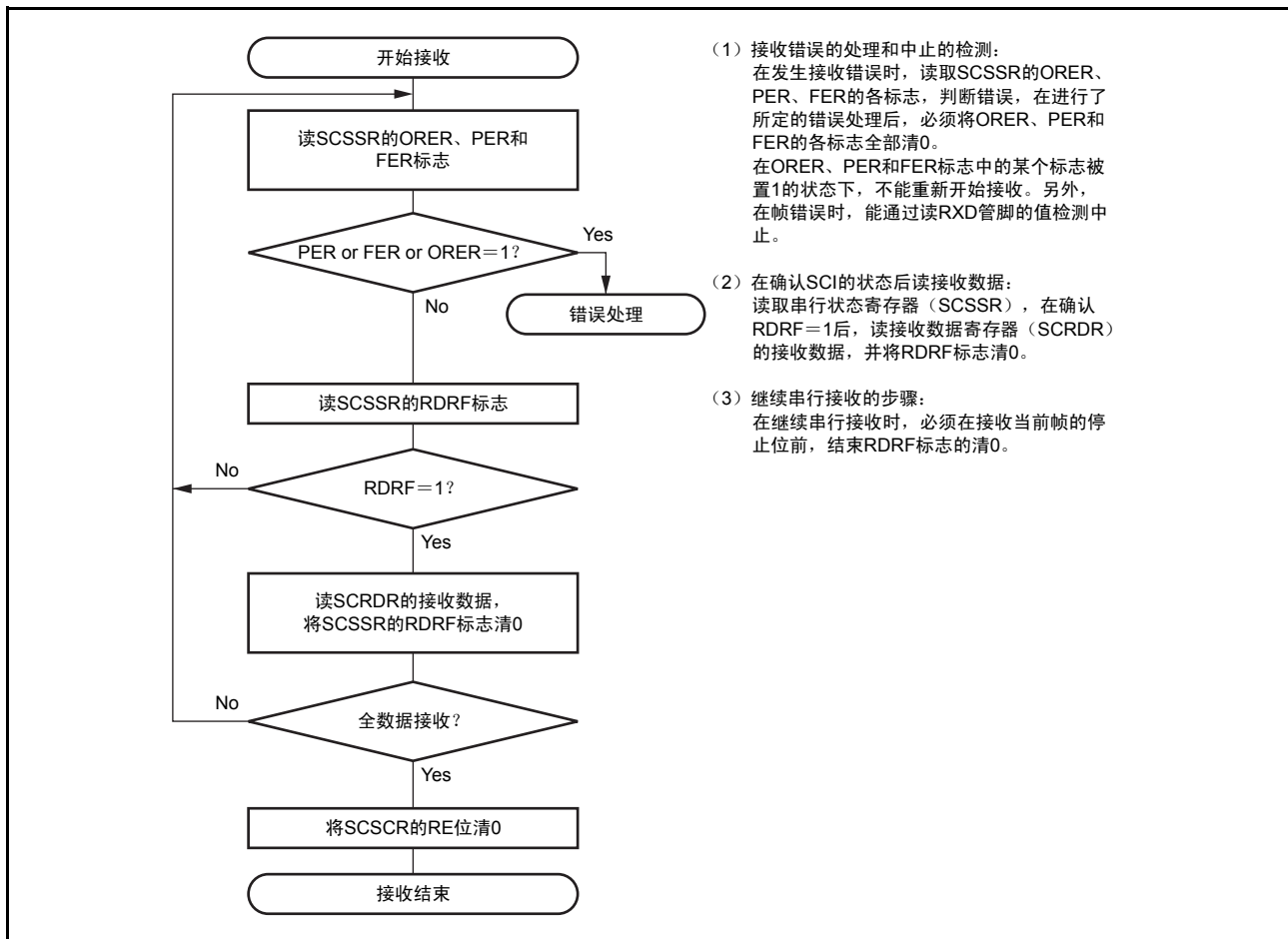


图 12.6 串行接收的流程图例子（1）

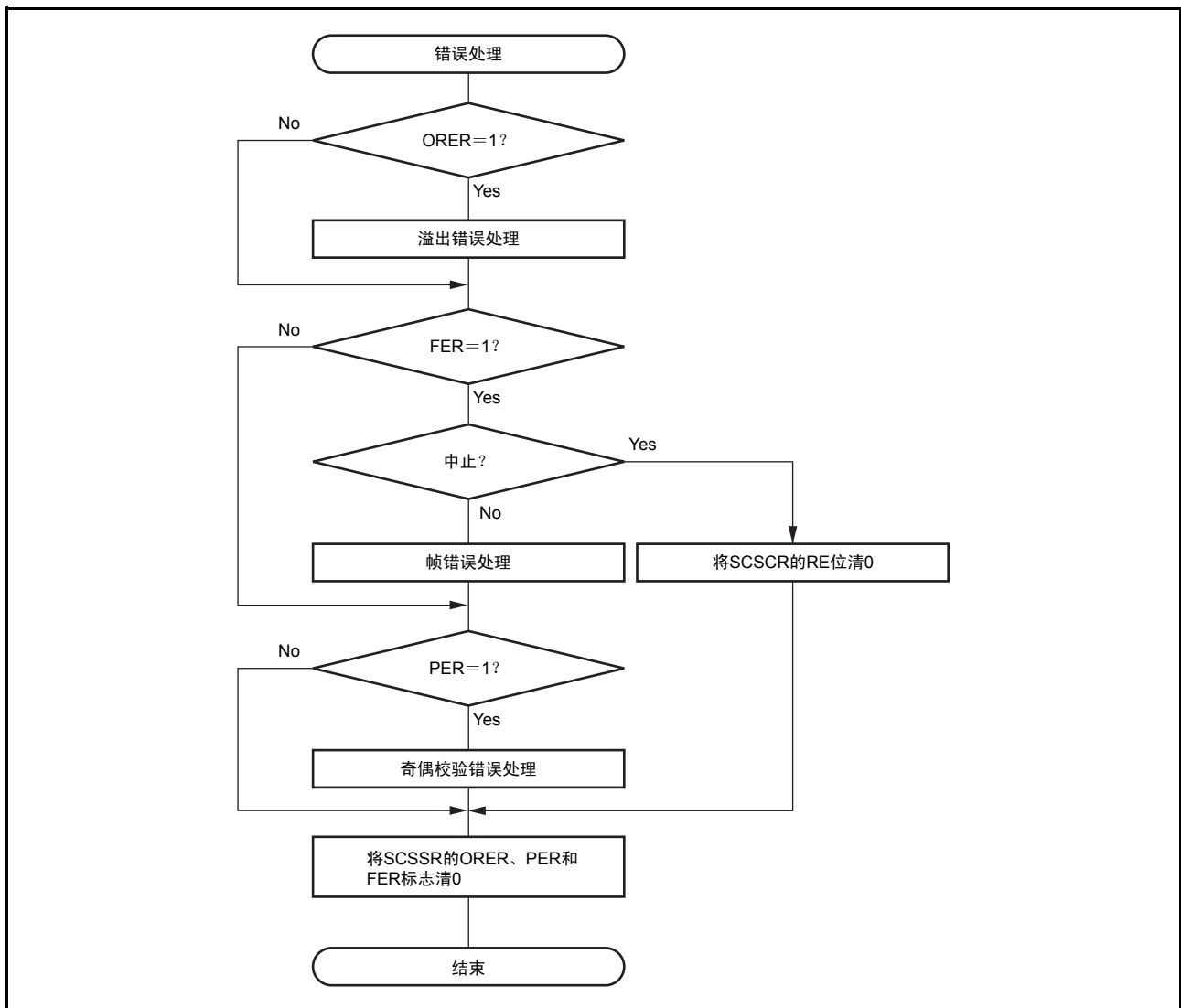


图 12.7 串行接收的流程图例子 (2)

接收时的 SCI 运行如下：

1. SCI 监视通信线路，如果检测到起始位的 0，就与内部同步开始接收。
2. 按照从 SCRSR 的 LSB 到 MSB 的顺序保存接收的数据。
3. 接收奇偶校验位和停止位。

在接收后，SCI 进行以下检查：

- (a) 奇偶校验的检查：检查接收数据的 1 的个数是否为串行模式寄存器 (SCSMR) 的 $\overline{\text{O/E}}$ 位设定的偶校验或奇校验。
- (b) 停止位的检查：检查停止位是否为 1。
但是，在 2 个停止位的情况下，只检查第 1 位的停止位。
- (c) 状态检查：检查 RDRF 标志是否为 0，即是否为能将接收数据从接收移位寄存器 (SCRSR) 传送到 SCRDR 的状态。

在以上检查全部通过后，将 RDRF 标志设定为 1，并且将接收数据保存到 SCRDR。

如果在错误检查时发生接收错误，就进行如表 12.16 所示的运行。

【注】在发生接收错误的状态下，不能进行以后的接收。另外，因为在接收时 RDRF 标志不被置 1，所以必将错误标志清 0。

4. 在 RDRF 标志为 1 时，如果 SCSCR 的 RIE 位已被置 1，就产生接收数据满的中断 (RXI) 请求。
另外，在 ORER、PER 和 FER 标志中的某个标志为 1 时，如果 SCSCR 的 RIE 位已被置 1，就产生接收错误的中断 (ERI) 请求。

表 12.16 接收错误和发生条件

接收错误	略称	发生条件	数据传送
溢出错误	ORER	在 SCSSR 的 RDRF 标志被置 1 的状态下结束下一个数据接收时	不将接收数据从 SCRSR 传送到 SCRDR。
帧错误	FER	当停止位为 0 时	将接收数据从 SCRSR 传送到 SCRDR。
奇偶校验错误	PER	当 SCSMR 设定的偶校验 / 奇校验和接收的数据不同时	将接收数据从 SCRSR 传送到 SCRDR。

异步模式接收时的运行例子如图 12.8 所示。

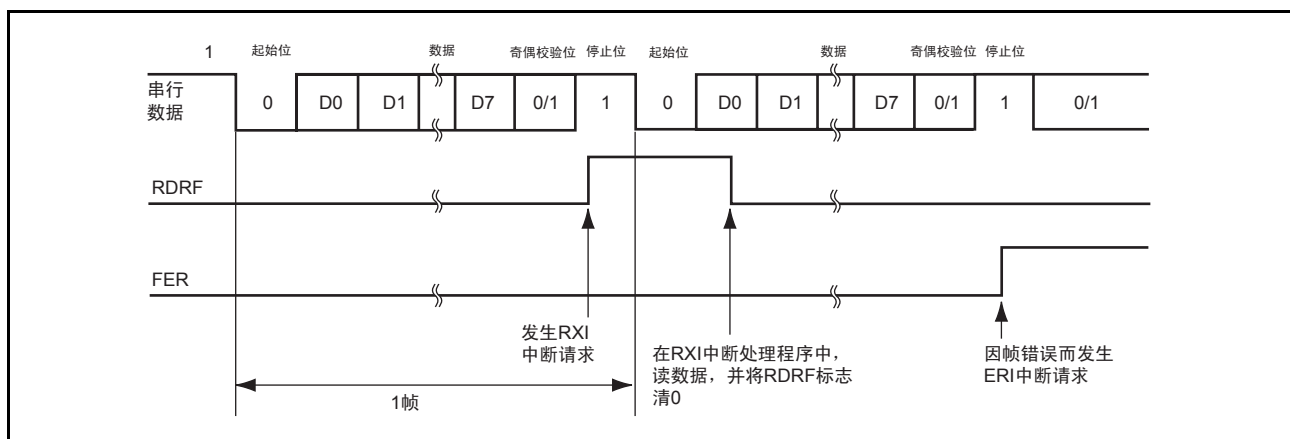


图 12.8 SCI 接收时的运行例子（8 位数据 / 有奇偶校验 / 1 个停止位的例子）

12.4.3 时钟同步模式中的运行（不能使用 SH7124 的通道 1）

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独立，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中读写数据，进行连续的发送和接收。

时钟同步串行通信的一般格式如图 12.9 所示。

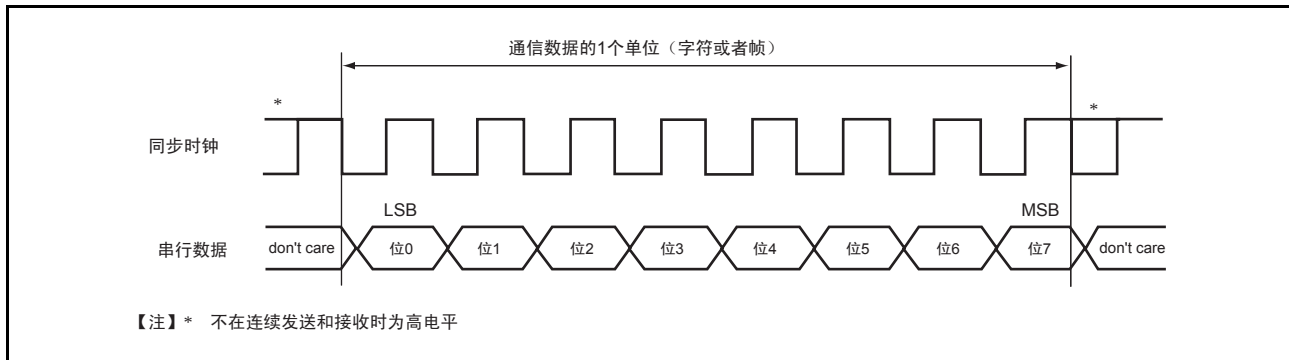


图 12.9 时钟同步通信的数据格式

在时钟同步串行通信中，在同步时钟的下降沿和下一个下降沿之间将数据输出到通信线路，在同步时钟的上升沿保证数据的有效。

以数据的 LSB（最初）到 MSB（最后）的顺序输出串行通信的 1 个字符，输出 MSB 后的通信线路状态保持 MSB 的状态。

在时钟同步模式中，SCI 与同步时钟的上升沿同步接收数据。

(1) 发送 / 接收格式

发送 / 接收格式固定为 8 位数据。

不能附加奇偶校验位。

(2) 时钟

时钟能通过设定 SCSMR 的 C/A 位和 SCSCR 的 CKE1、CKE0 位，从内部波特率发生器生成的内部时钟或者 SCK 管脚输入的外部同步时钟中选择。有关 SCI 时钟源的选择请参照表 12.14。

在通过内部时钟运行时，从 SCK 管脚输出同步时钟。

同步时钟在发送和接收 1 个字符时输出 8 个脉冲，而在不进行发送和接收时固定为高电平。

(3) 数据的发送和接收运行

• SCI 的初始化（时钟同步）

在发送和接收数据前，必须在将串行控制寄存器（SCSCR）的 TE 和 RE 位清 0 后，按照以下步骤初始化 SCI。

在更改模式和通信格式等时，必须在将 TE 和 RE 位清 0 后按照以下步骤进行。如果将 TE 位清 0，TDRE 标志就被置 1，发送移位寄存器（SCTSR）被初始化。

必须注意：即使将 RE 位清 0，也保持 RDRF、PER、FER、ORER 的各标志和接收数据寄存器（SCRDR）的内容。

SCI 的初始化流程图例子如图 12.10 所示。

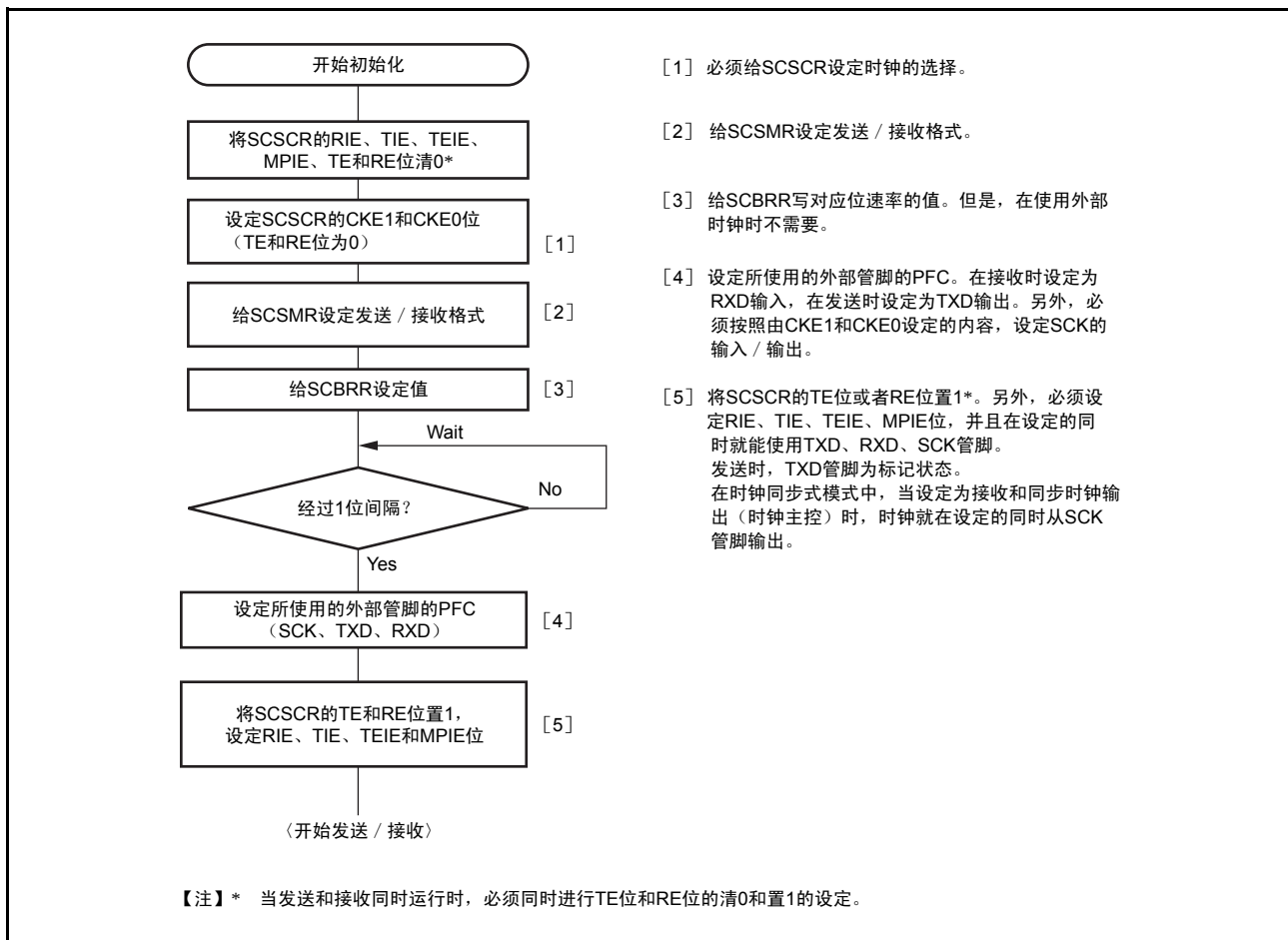


图 12.10 SCI 的初始化流程图例子

- 串行数据的发送（时钟同步）

串行发送的流程图例子如图 12.11 所示。

必须按照以下步骤，在将 SCI 设定为可发送的运行状态后发送串行数据。

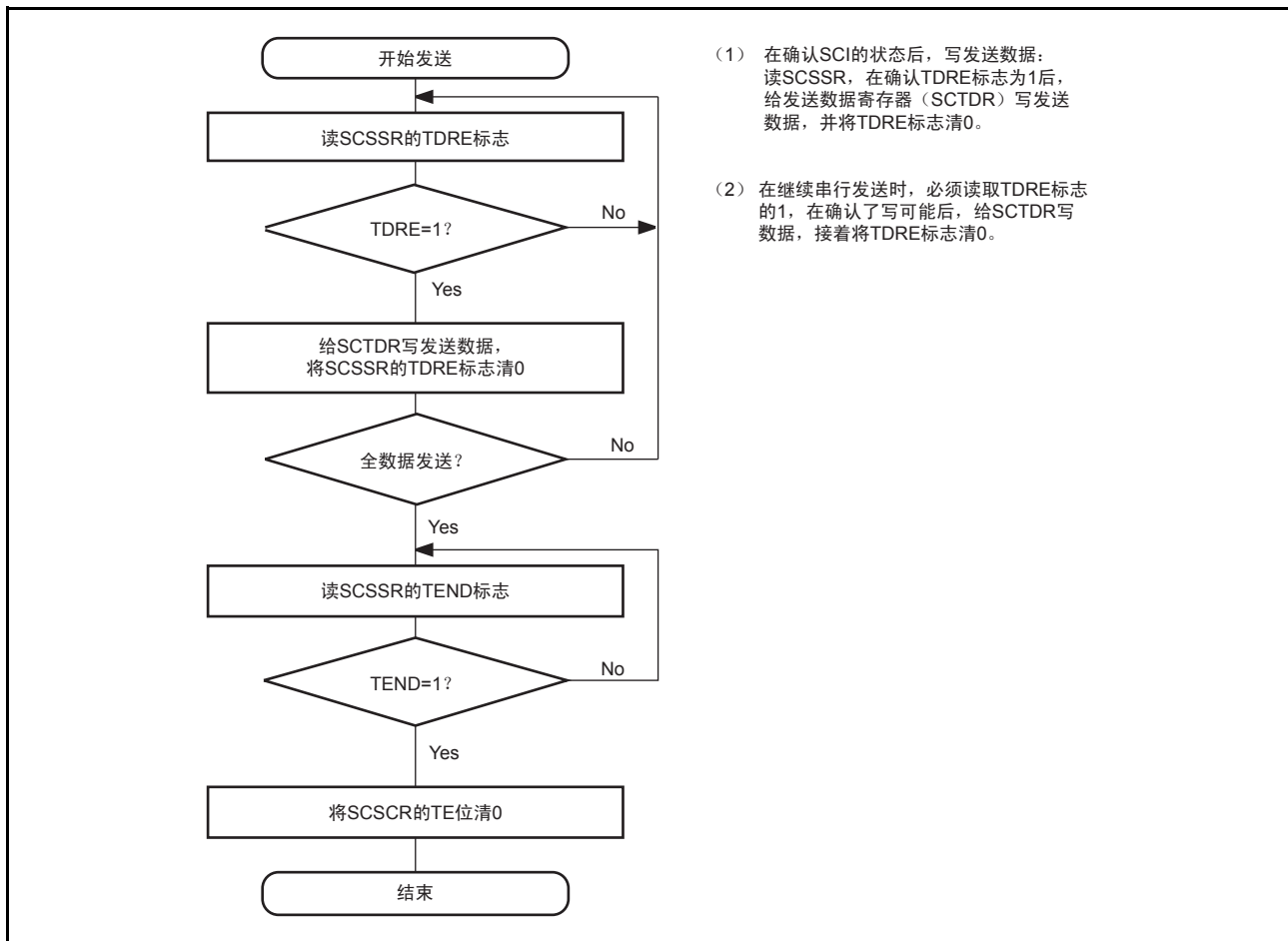


图 12.11 串行发送的流程图例子

串行发送时的 SCI 运行如下：

1. SCI 监视串行状态寄存器 (SCSSR) 的 TDRE 标志，如果为 0，就认为已将数据写到发送数据寄存器 (SCTDR)，将数据从 SCTDR 传送到发送移位寄存器 (SCTSR)。
2. 在数据从 SCTDR 传送到 SCTSR 后，将 TDRE 标志设定为 1，开始发送。
此时，如果串行控制寄存器 (SCSCR) 的发送数据空的中断允许位 (TIE) 已被置 1，就产生发送数据空的中断 (TXI) 请求。
在设定为时钟输出模式时，SCI 输出同步时钟的 8 个脉冲。
在设定为外部时钟时，与输入时钟同步输出数据。
按照 LSB (bit0) ~ MSB (bit7) 的顺序从 TXD 管脚送出串行发送数据。
3. SCI 在送出 MSB (bit7) 时检查 TDRE 标志。
如果 TDRE 标志为 0，就将数据从 SCTDR 传送到 SCTSR，开始下一帧的串行发送。
如果 TDRE 标志为 1，就将串行状态寄存器 (SCSSR) 的 TEND 标志设定为 1，在送出 MSB (bit7) 后发送数据的管脚 (TXD 管脚) 保持状态。
此时，如果 SCSCR 的发送结束的中断允许位 (TEIE) 已被置 1，就产生发送结束的中断请求 (TEI)。
4. 结束串行发送后，SCK 管脚固定为高电平。

SCI 发送时的运行例子如图 12.12 所示。

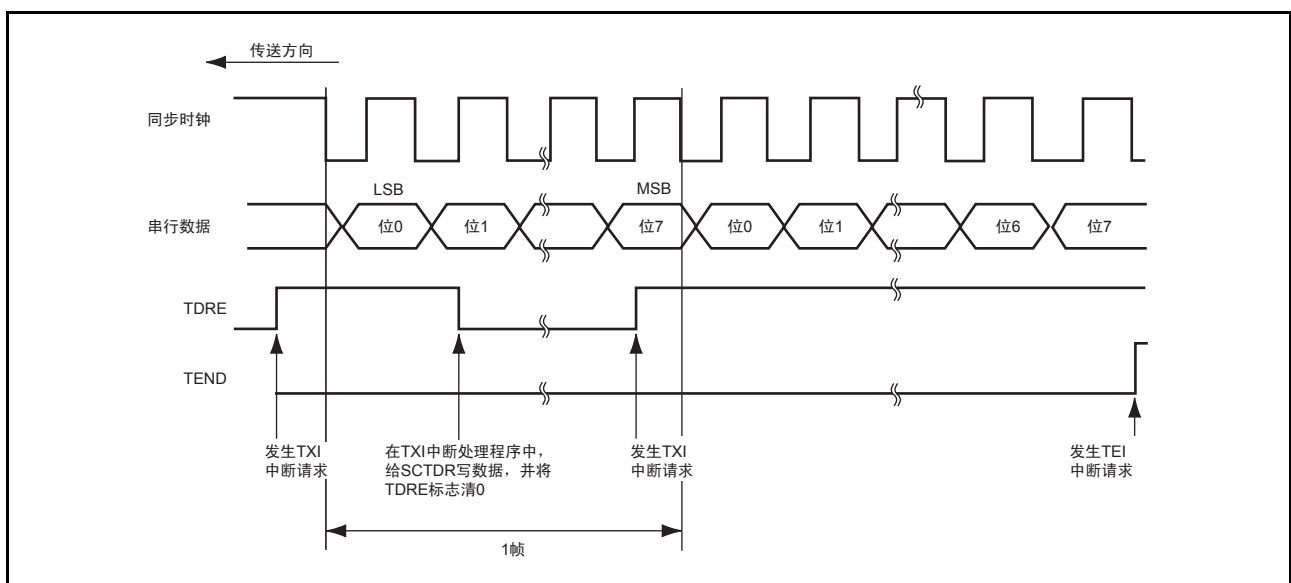


图 12.12 SCI 发送时的运行例子

- 串行数据的接收 (时钟同步)

串行接收的流程图例子如图 12.13 和图 12.14 所示。

必须按照以下步骤，在将 SCI 设定为可接收的运行状态后接收串行数据。

在将运行模式从异步模式转换为时钟同步模式时，必须确认 ORER、PER、FER 的各标志是否已被清 0。

如果 FER、PER 标志已被置 1，RDRF 标志就不能置位，并且不能发送和接收。

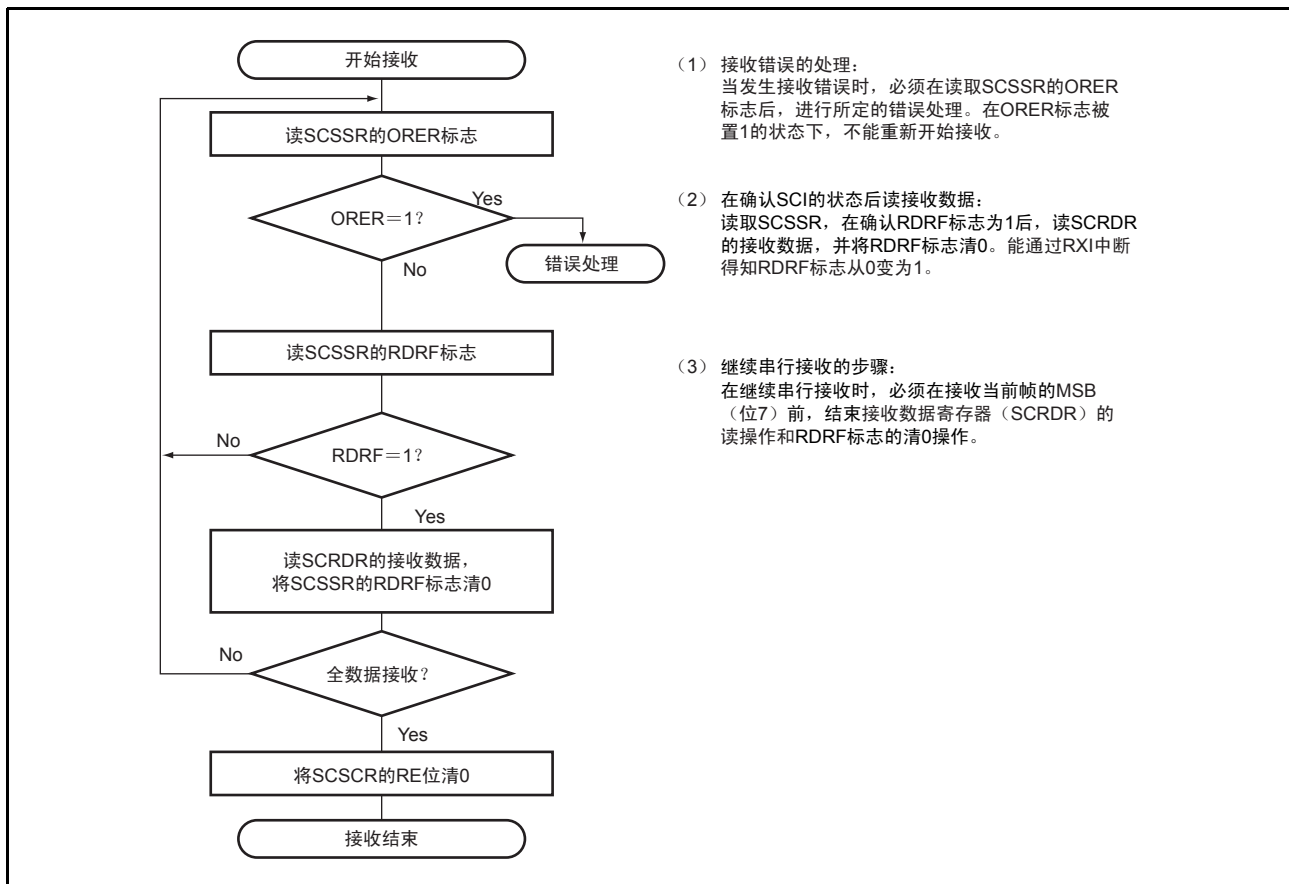


图 12.13 串行数据的接收流程图例子 (1)

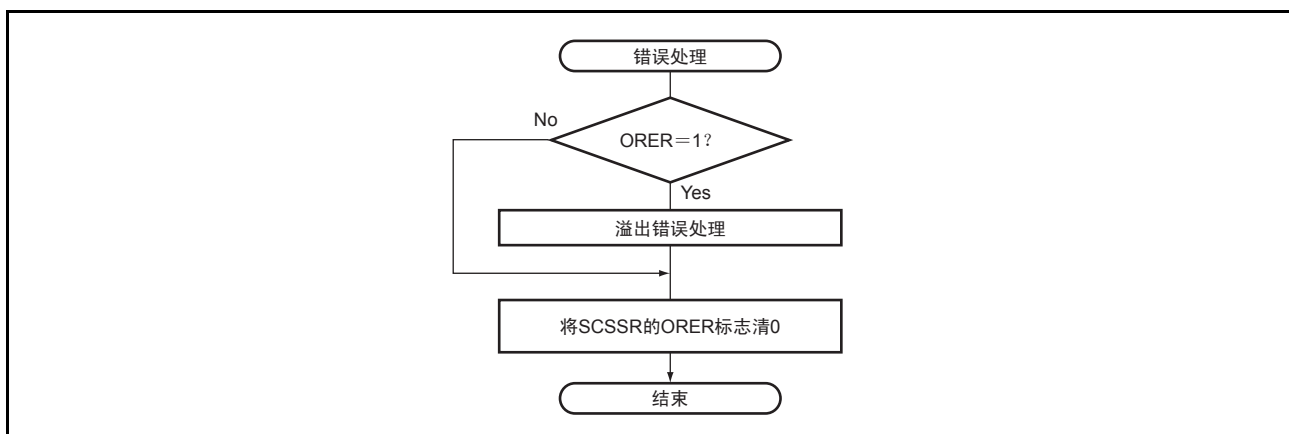


图 12.14 串行数据接收的流程图例子 (2)

接收时的 SCI 运行如下：

1. SCI 与同步时钟的输入或输出同步，开始接收。

2. 按照接收移位寄存器 (SCRSR) 的 LSB 到 MSB 的顺序保存接收的数据。

在接收后，SCI 检查 RDRF 标志是否为 0，即是否为能将接收数据从 SCRSR 传送到接收数据寄存器 (SCRDR) 的状态。

在此检查通过后，将 RDRF 标志设定为 1，并且将接收数据保存到 SCRDR。

如果在错误检查时发生接收错误，就进行如表 12.16 所示的运行，在此状态下不能进行以后的发送和接收。

另外，因为在接收时 RDRF 标志不被置 1，所以必须将此标志清 0。

3. 在 RDRF 标志为 1 时，如果串行控制寄存器 (SCSCR) 的 RIE 位已被置 1，就产生接收数据满的中断 (RXI) 请求。

另外，在 ORER 标志为 1 时，如果 SCSCR 的 RIE 位已被置 1，就产生接收错误的中断 (ERI) 请求。

SCI 接收时的运行例子如图 12.15 所示。

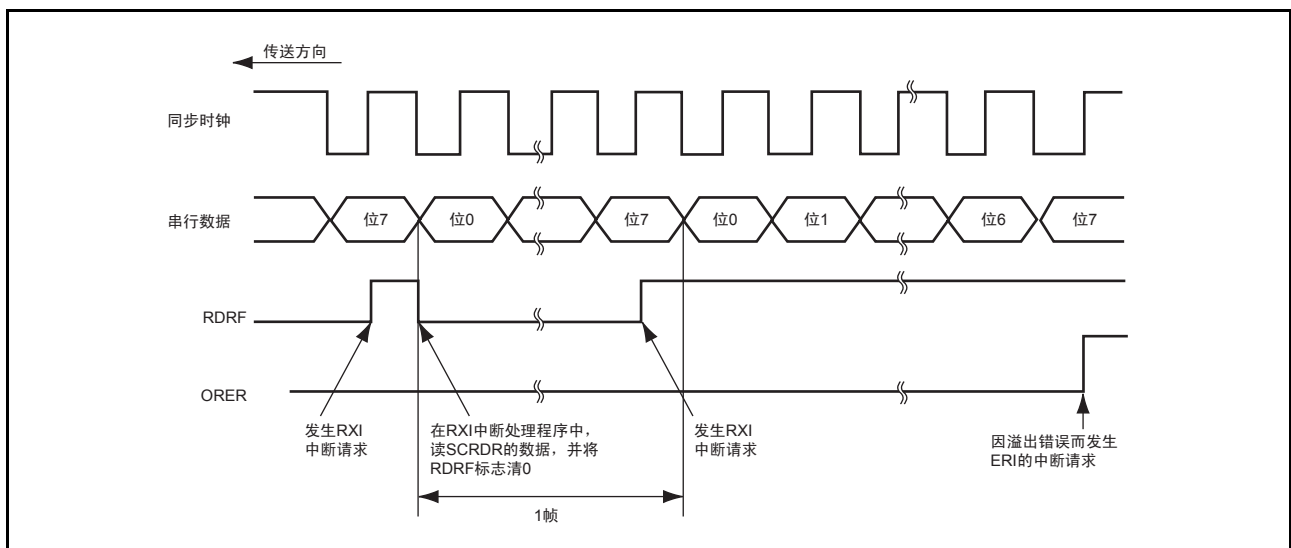


图 12.15 SCI 接收时的运行例子

- 串行数据发送和接收的同时运行（时钟同步）

串行发送和接收的同时运行的流程图例子如图 12.16 所示。

必须按照以下步骤在将 SCI 设定为可接收和发送的运行状态后进行串行数据发送和接收的同时运行。

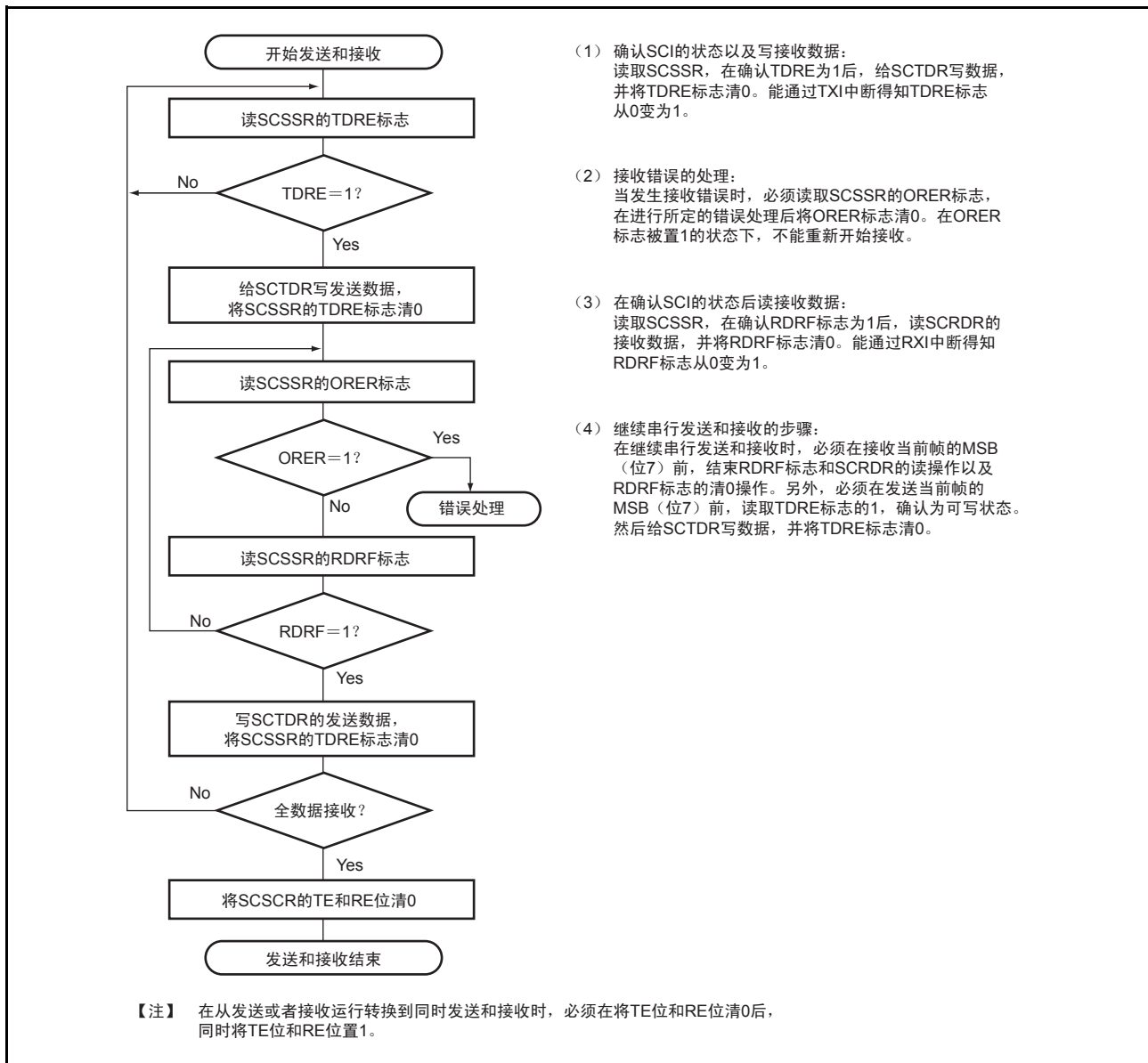


图 12.16 串行数据发送和接收的流程图例子

12.4.4 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 12.17 所示。发送站首先发送多处理器位为 1 的接收站 ID 码，接着发送多处理器位为 0 的发送数据。如果接收站接收到多处理器位为 1 的通信数据，就将接收数据与本站的 ID 比较，如果一致，就继续接收被发送的通信数据；如果不一致，就在接收到下一个多处理器位为 1 的通信数据之前，跳读通信数据。

SCI 为了支持此功能，在 SCSCR 中设有 MPIE 位。如果将 MPIE 置 1，就在接收到多处理器位为 1 的数据之前，禁止将接收数据从 SCRSR 传送到 SCRDR，并禁止接收错误的检测以及禁止 SCSSR 的 RDRF、FER、OER 各状态标志的置位。如果接收到多处理器位为 1 的字符，就在将 SCSSR 的 MPBR 置 1 的同时自动清除 MPIE，然后返回到通常的接收运行状态。此时，如果 SCSCR 的 RIE 已被置位，就产生 RXI 中断。

在指定多处理器格式时，奇偶校验位的指定无效。除此以外与通常的异步模式相同，多处理器通信时的时钟也和通常的异步模式相同。

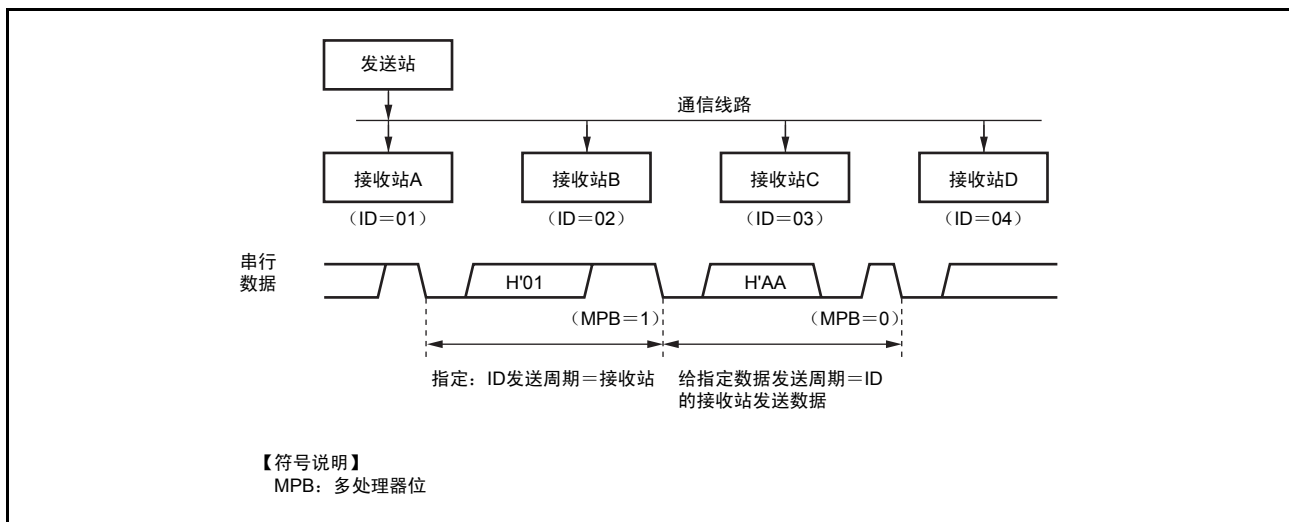


图 12.17 使用多处理器格式的通信的例子（将数据 H'AA 发送到接收站 A 的例子）

12.4.5 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 12.18 所示。必须在 ID 发送周期将 SCSSR 的 MPBT 置 1 后发送 ID 码，在数据发送周期将 SCSSR 的 MPBT 清 0 后发送数据。其它运行和异步模式的运行相同。

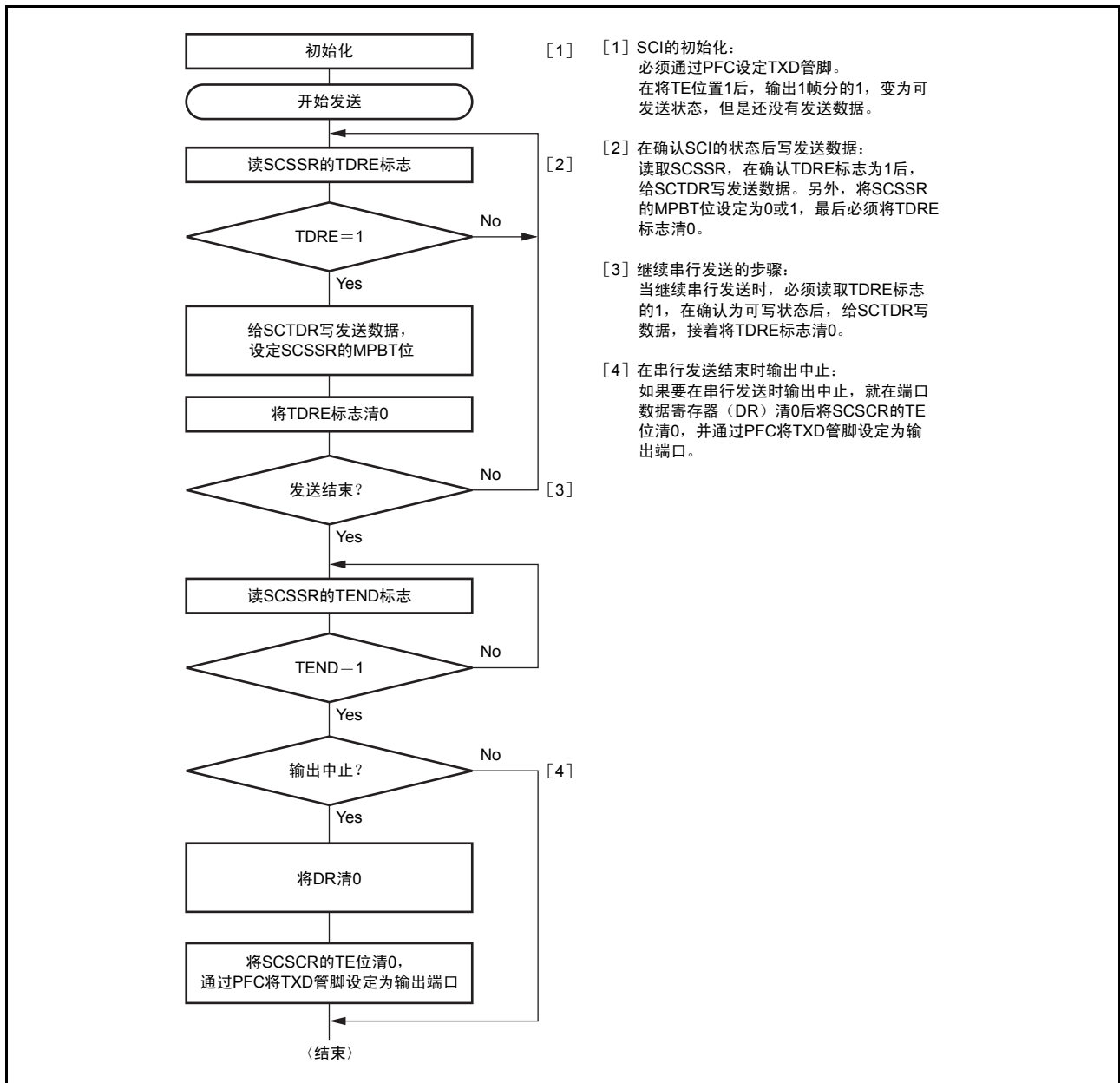


图 12.18 多处理器串行发送的流程图例子

12.4.6 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 12.20 和图 12.21 所示。如果将 SCSCR 的 MPIE 置 1，就在接收到多处理器位为 1 的通信数据之前，跳读通信数据。如果接收到多处理器位为 1 的通信数据，就将接收数据传送到 SCRDR，此时产生 RXI 中断请求。其它运行和异步模式的运行相同。接收时的运行例子如图 12.19 所示。

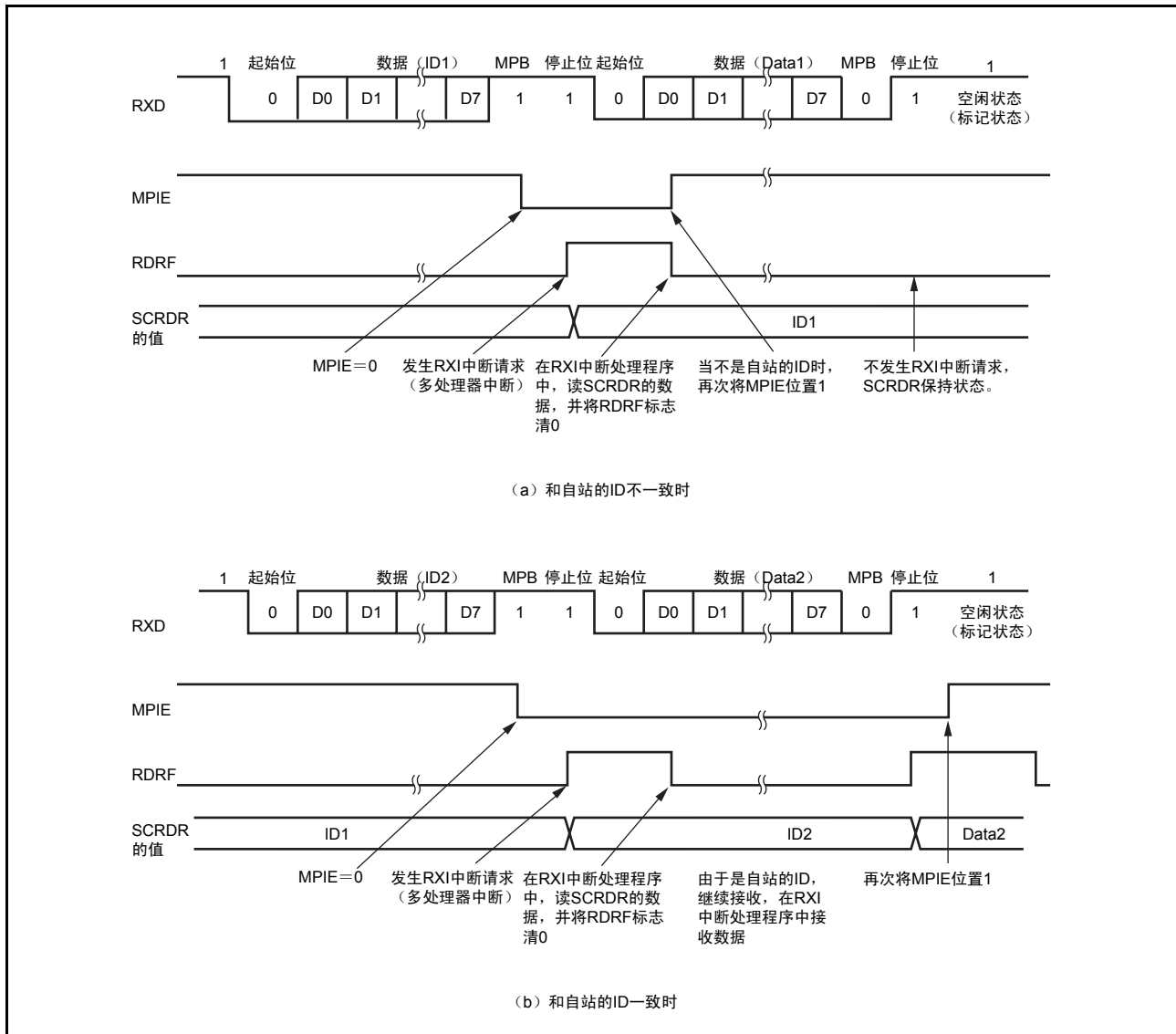


图 12.19 SCI 接收时的运行例子 (8 位数据 / 有多处理器位 / 1 个停止位的例子)

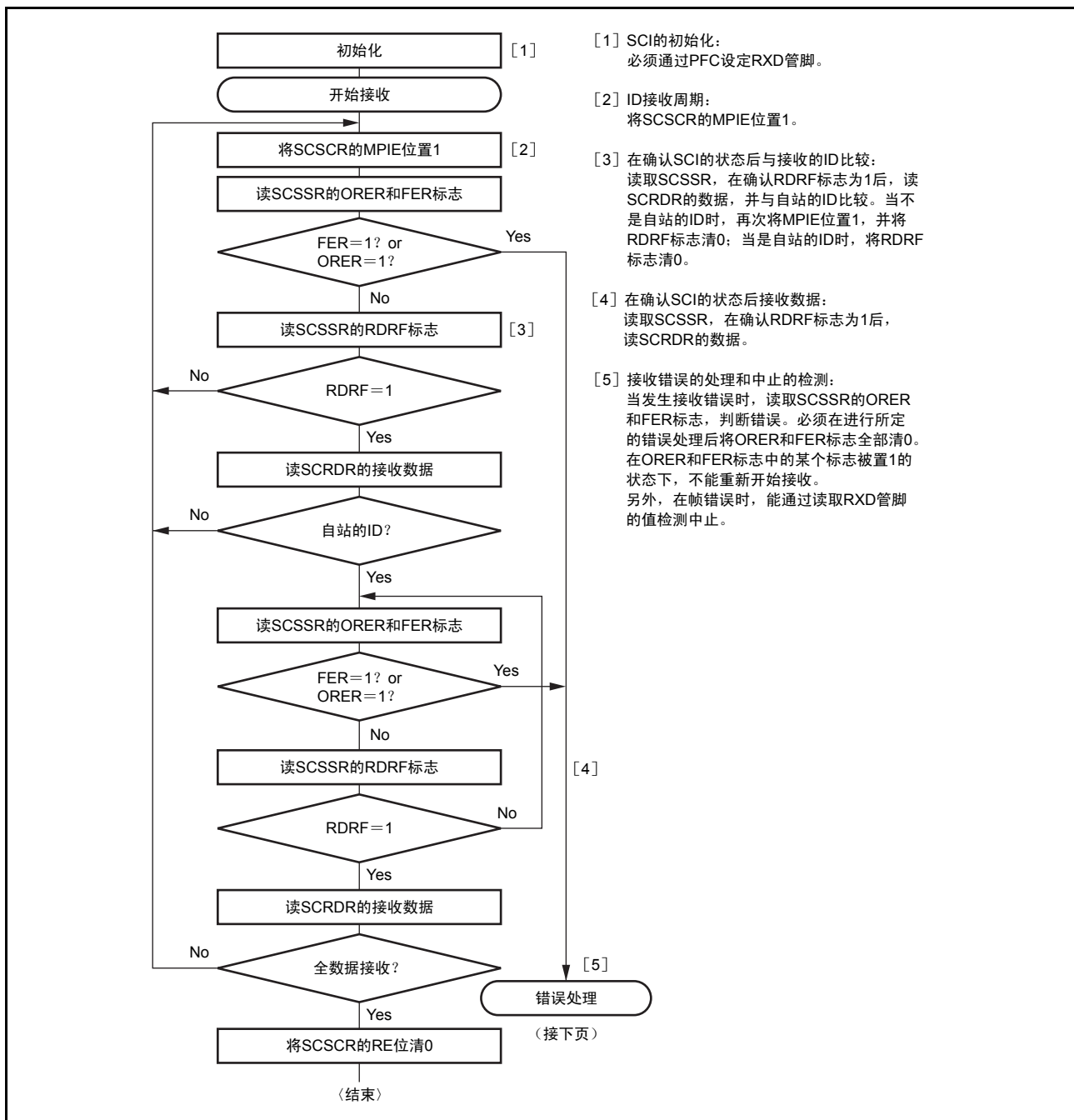


图 12.20 多处理器串行接收的流程图例子 (1)

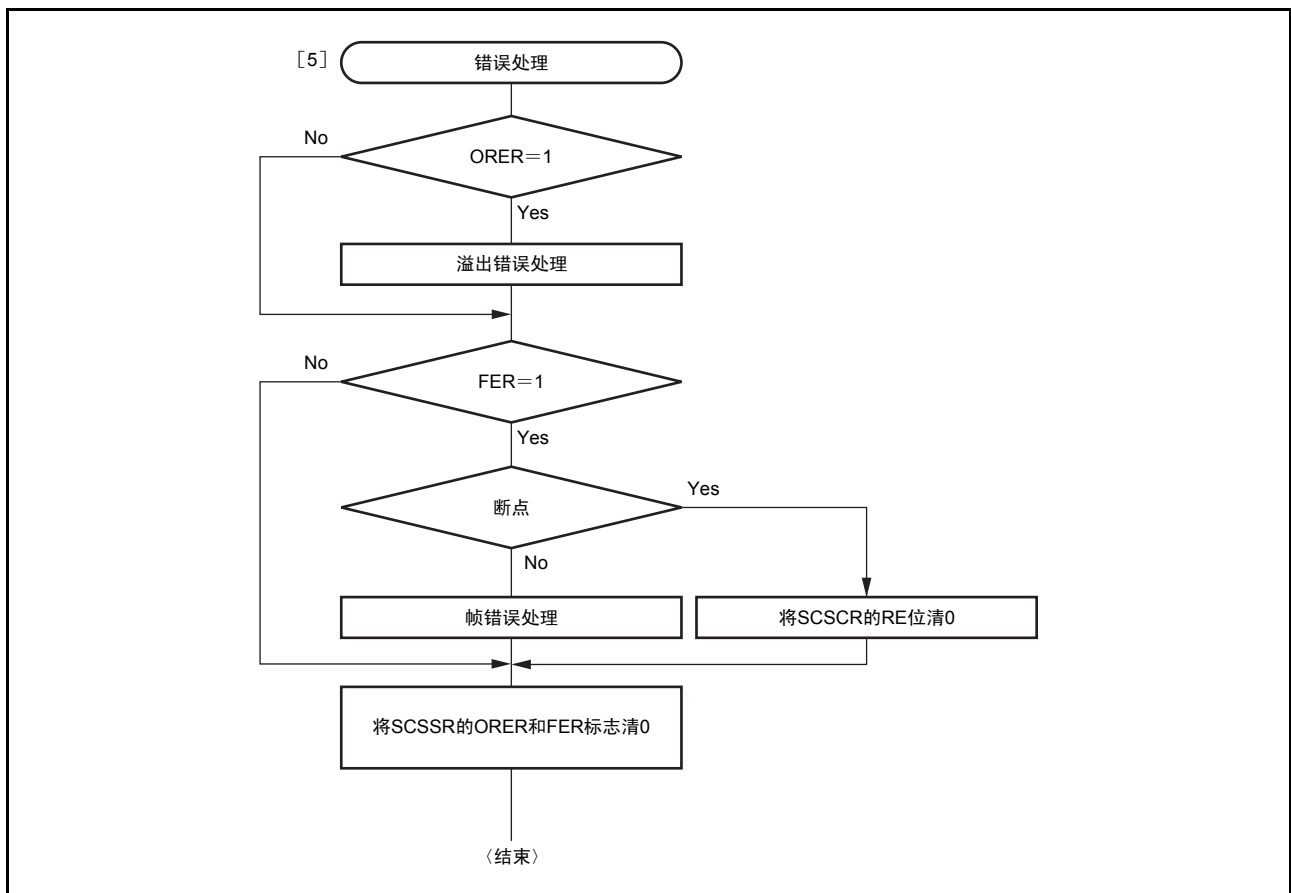


图 12.21 多处理器串行接收的流程图例子 (2)

12.5 SCI 的中断源

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 12.17 所示。能通过 SCSCR 的 TIE、RIE、TEIE 位以及 SCSPTR 的 EIO 位，允许或者禁止各中断源。另外，分别将各中断请求独立传送到中断控制器。

如果串行状态寄存器（SCSSR）的 TDRE 标志被置 1，就产生 TDR 空中断请求；如果 SCSSR 的 RDRF 标志被置 1，就产生 RDR 满中断请求；如果 SCSSR 的 ORER、FER 标志或者 PER 被置 1，就产生 ERI 中断请求；如果 SCSSR 的 TEND 标志被置 1，就产生 TEI 中断请求。

另外，TXI 中断表示可写发送数据，TEI 中断表示发送结束。

表 12.17 SCI 中断源

中断源	内 容
ERI	接收错误（ORER、FER、PER）的中断
RXI	接收数据满（RDRF）的中断
TXI	发送数据空（TDRE）的中断
TEI	发送结束（TEND）的中断

12.6 串行端口寄存器 (SCSPTR) 和 SCI 管脚的关系

SCSPTR 和 SCI 管脚的关系如图 12.22 和图 12.23 所示。

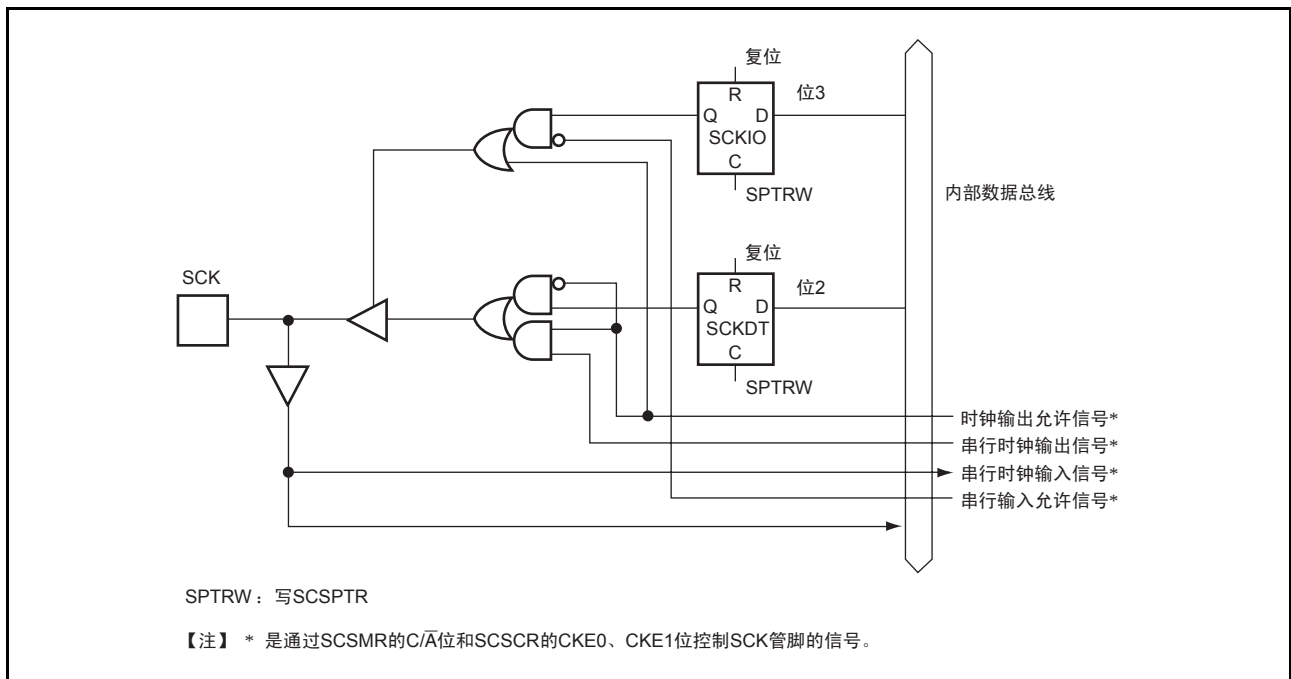


图 12.22 SPB1IO 位、SPB1DT 位和 SCK 管脚的关系

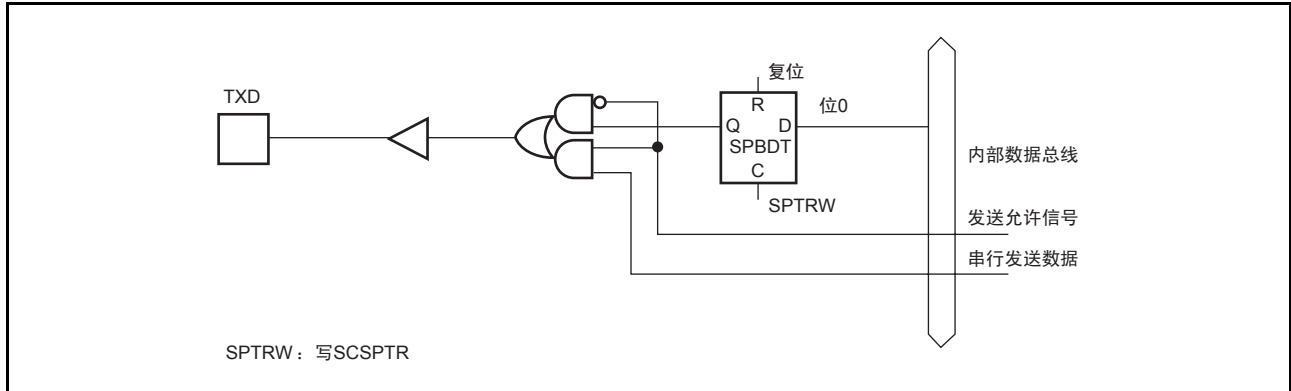


图 12.23 SPB0DT 位和 TXD 管脚的关系

12.7 使用注意事项

12.7.1 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。

12.7.2 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 12.18 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 12.18 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	X
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	X
溢出错误 + 奇偶校验错误	1	1	0	1	X
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	X

【符号说明】

○: 将接收数据从 SCRSR 传送到 SCRDR。

X: 不将接收数据从 SCRSR 传送到 SCRDR。

12.7.3 有关中止的检测和处理

在检测帧错误 (FER) 时, 能通过直接读 RXD 管脚的值检测中止。因为在中止时 RXD 管脚的输入始终为 0, 所以 FER 标志被置位, 并且奇偶校验错误 (PER) 也可能被置位。

必须注意: SCI 在接收中止后还继续接收运行, 但是接收的数据不传送到 SCRDR。

12.7.4 中止的发送

能通过串行端口寄存器 (SCSPTR) 的 SPB0IO 和 SPB0DT 位决定 TXD 管脚的输入 / 输出的条件和电平, 所以利用这一功能可以发送中止。

在串行发送的初始化后并且在将 TE 位置 1 前, TXD 管脚不工作, 此期间的标记状态由 SPB0DT 位值代替。因此, 最初需要事先将 SPB0IO 和 SPB0DT 位设定为 1 (输出高电平)。

当要在串行发送期间时发送中止时, 就要在将 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, 发送部被初始化, 从 TXD 管脚输出 0。

12.7.5 异步模式的接收数据采样时序和接收容限

在异步模式中，SCI 通过传送速率的 16 倍频率的基本时钟运行。

在接收时，SCI 通过基本时钟采样起始位的下降沿，取得内部同步。另外，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 12.24 所示。

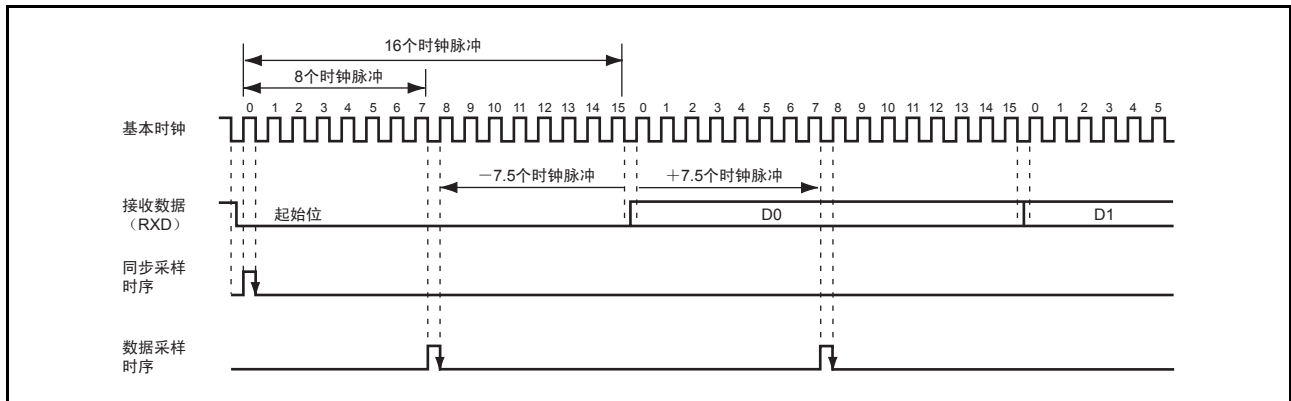


图 12.24 异步模式的接收数据采样时序

因此，能用表达式 (1) 表示异步模式的接收容限。

$$M = \left| \left(0.5 - \frac{1}{2N} \right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100[\%] \quad \cdots \text{表达式 (1)}$$

M: 接收容限 (%)

N: 对应时钟的位速率比 (N=16)

D: 时钟占空比 (D=0 ~ 1.0)

L: 帧长 (L=9 ~ 12)

F: 时钟频率的偏差绝对值

当表达式 (1) 的中 F=0、D=0.5 时，根据表达式 (2) 接收容限就为 46.875%。

当 D=0.5、F=0 时

$$M = (0.5 - 1/(2 \times 16)) \times 100\% = 46.875\% \quad \cdots \text{表达式 (2)}$$

但是，因为此值为计算值，所以在系统设计时必须留有 20 ~ 30% 的余地。

12.7.6 时钟同步外部时钟模式的注意事项

必须在将外部时钟 SCK 从 0 变为 1 后经过 4 个以上的外围运行时钟周期后，然后设定为 TE=1、RE=1。

必须在外部时钟 SCK 为 1 时设定为 TE=RE=1。

12.7.7 模块待机模式的设定

SCI 能通过待机控制寄存器，禁止或者允许本模块的运行。初始值为停止 SCI 的运行。通过解除模块待机模式，允许寄存器的存取。详细内容请参照“第 19 章 低功耗模式”。

第 13 章 A/D 转换器 (ADC)

本 LSI 内置逐次逼近方式的 10 位 A/D 转换器。

13.1 特点

- 分辨率：10 位
- 输入通道：8 个通道（内置 2 个独立的 A/D 转换模块）
- 转换时间：平均 1 个通道 2.0 μ s（暂定值，在 P ϕ =25MHz 运行时）
- 运行模式：3 种
 - 单通道模式：1 个通道的 A/D 转换
 - 连续扫描模式：最多 4 个通道的重复 A/D 转换
 - 单周期扫描模式：最多 4 个通道的连续 A/D 转换
- 数据寄存器：将 A/D 转换结果保存到对应各输入通道的 16 位数据寄存器
- 采样和保持功能
- A/D 转换的开始方法：3 种
 - 软件
 - 可选择多功能定时器脉冲单元 2（MTU2）的转换开始触发
 - 外部触发信号
- 中断源：产生 A/D 转换结束的中断请求（ADI）
- 能设定模块待机模式

A/D 转换器的框图如图 13.1 所示。

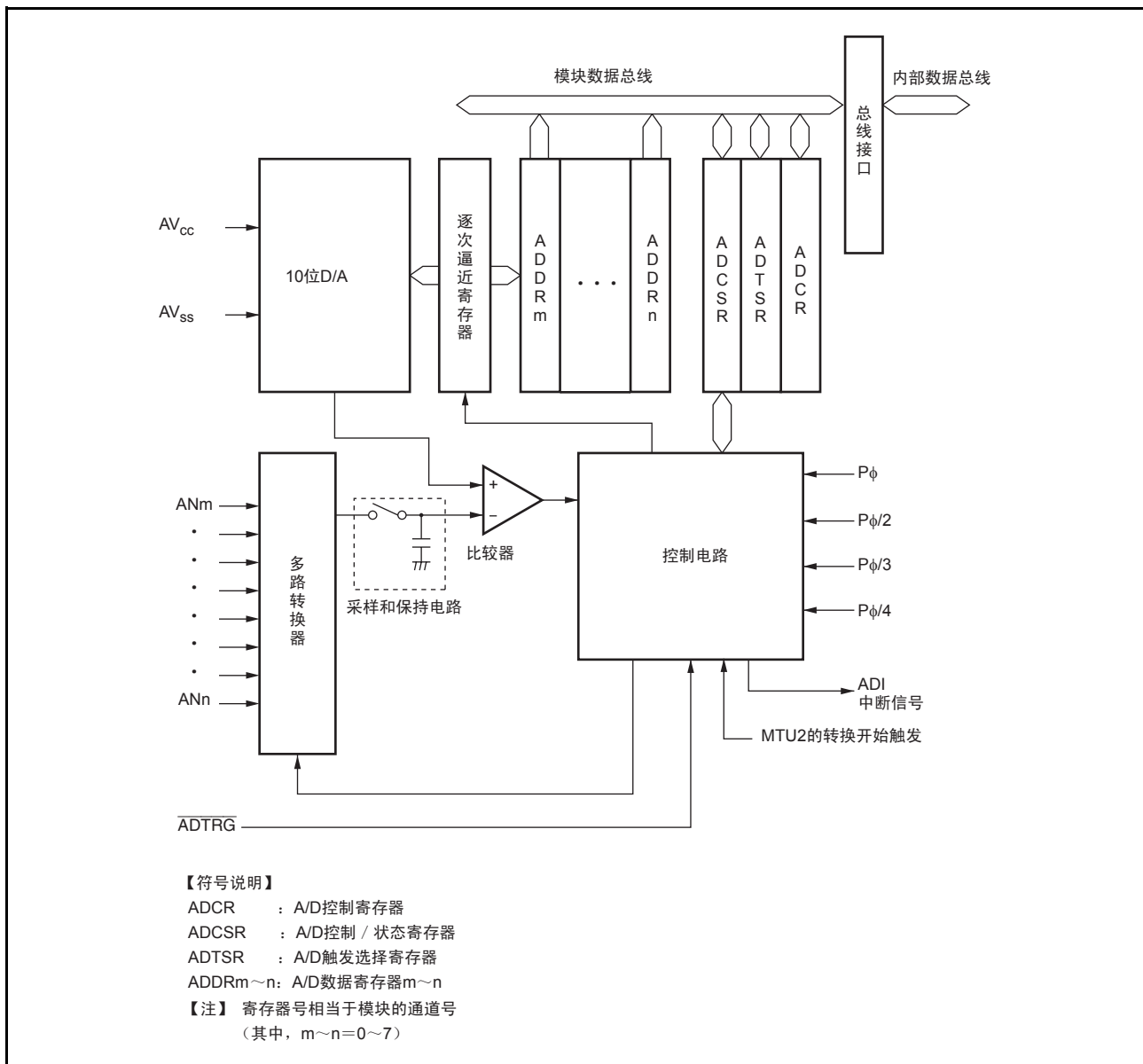


图 13.1 A/D 转换器的框图 (1 个模块)

13.2 输入 / 输出管脚

A/D 转换器使用的管脚如表 13.1 所示。A/D 转换器由 2 个能各自独立运行的 A/D 转换模块构成，A/D 模块 0、1 的输入通道分为 2 个通道组。

表 13.1 管脚结构

模块区分	管脚名	输入 / 输出	功 能	
通用	AVcc	输入	模拟部的电源管脚和基准电压	
	AVss	输入	模拟部的接地和基准电压	
	ADTRG	输入	A/D 外部触发的输入管脚 *	
A/D 模块 0 (A/D_0)	AN0	输入	模拟输入管脚 0	组 0
	AN1	输入	模拟输入管脚 1	
	AN2	输入	模拟输入管脚 2	组 1
	AN3	输入	模拟输入管脚 3	
A/D 模块 1 (A/D_1)	AN4	输入	模拟输入管脚 4	组 0
	AN5	输入	模拟输入管脚 5	
	AN6	输入	模拟输入管脚 6	组 1
	AN7	输入	模拟输入管脚 7	

【注】 连接各管脚的 A/D 模块不同。因为各模块都有控制寄存器，所以必须分别进行设定。

* 只限 SH7125 的管脚。

13.3 寄存器说明

A/D 转换器有以下寄存器。有关这些寄存器的地址和各处理模式中的寄存器状态，请参照“第 20 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
A/D 数据寄存器 0	ADDR0	R	H'0000	H'FFFC900	16
A/D 数据寄存器 1	ADDR1	R	H'0000	H'FFFC902	16
A/D 数据寄存器 2	ADDR2	R	H'0000	H'FFFC904	16
A/D 数据寄存器 3	ADDR3	R	H'0000	H'FFFC906	16
A/D 控制 / 状态寄存器 _0	ADCSR_0	R/W	H'0000	H'FFFC910	16
A/D 控制寄存器 _0	ADCR_0	R/W	H'0000	H'FFFC912	16
A/D 数据寄存器 4	ADDR4	R	H'0000	H'FFFC980	16
A/D 数据寄存器 5	ADDR5	R	H'0000	H'FFFC982	16
A/D 数据寄存器 6	ADDR6	R	H'0000	H'FFFC984	16
A/D 数据寄存器 7	ADDR7	R	H'0000	H'FFFC986	16
A/D 控制 / 状态寄存器 _1	ADCSR_1	R/W	H'0000	H'FFFC990	16
A/D 控制寄存器 _1	ADCR_1	R/W	H'0000	H'FFFC992	16
A/D 触发选择寄存器 _0	ADTSR_0	R/W	H'0000	H'FFFE890	8、16

13.3.1 A/D 数据寄存器 0 ~ 7（ADDR0 ~ ADDR7）

ADDR 是用于保存 A/D 转换结果的 16 位只读寄存器，各模拟输入通道的转换结果保存到对应序号的 ADDR（例如，AN4 的转换结果保存到 A/D 数据寄存器 ADDR4）。

10 位转换数据保存在 ADDR 的 bit15 ~ bit6，低 6 位的读出值总是 0。

与 CPU 之间的数据总线为 16 位宽度。在读 ADDR 时，必须进行字存取。ADDR 的初始值为 H'0000。

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

13.3.2 A/D 控制 / 状态寄存器 _0、1（ADCSR_0、1）

ADCSR 控制各模块的 A/D 转换运行。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADF	ADIE	-	-	TRGE	-	CONADF	STC	CKSL[1:0]	ADM[1:0]	ADCS	CH[2:0]				
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/W	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

【注】* 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说 明
15	ADF	0	R/(W)*	A/D 结束标志 表示 A/D 转换结束的状态。 [置位条件] • 在单通道模式中，当 A/D 转换结束时 • 在扫描模式中，当所选通道的 A/D 转换全部结束时 [清除条件] • 当读 1 的状态后写 0 时
14	ADIE	0	R/W	A/D 中断（ADI）的允许 如果置 1，就允许由 ADF 产生的 ADI 中断。 必须在 ADST 为 0 的状态下转换运行模式。
13、12	—	全 0	R	保留位 读写的值总是 0。
11	TRGE	0	R/W	触发的允许 设定由 <u>ADTRG</u> 或者 MTU2 触发产生的 A/D 转换开始。 0: 由触发产生的 A/D 转换开始无效 1: 由触发产生的 A/D 转换开始有效 必须在 ADST 为 0 的状态下转换运行模式。
10	—	0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说 明
9	CONADF	0	R/W	ADF 控制 控制 2 通道扫描模式中的 ADF 运行。此位只在 2 通道扫描模式并且在设定为由触发产生的 A/D 转换开始 (TRGE=1) 时有效, 而在单通道模式或者 4 通道扫描模式时被忽视。 0: 在组 0 触发或者组 1 触发的转换结束时, ADF 被置位。 1: 在组 0 触发和组 1 触发的转换都结束时, ADF 被置位, 并且不影响触发的顺序。 必须在 ADST 为 0 的状态下转换运行模式。
8	STC	0	R/W	状态控制 此位和 CKSL1、CKSL0 组合, 进行 A/D 转换时间的设定。 0: 50 个状态 1: 64 个状态 必须在 ADST 为 0 的状态下进行 A/D 转换时间的转换。
7、6	CKSL[1:0]	00	R/W	时钟的选择 1、0 设定 A/D 转换时间。 00: $P\phi/4$ 01: $P\phi/3$ 10: $P\phi/2$ 11: $P\phi$ 必须在 ADST 为 0 的状态下进行 A/D 转换时间的转换。 在 $P\phi \leq 25\text{M}[\text{Hz}]$ 时, 可设定 CKSL[1:0]=B'11。
5、4	ADM[1:0]	00	R/W	A/D 模式 1、0 选择 A/D 转换的运行模式。 00: 单通道模式 01: 4 通道扫描模式 10: 禁止设定 11: 2 通道扫描模式 必须在 ADST 为 0 的状态下转换运行模式。
3	ADCS	0	R/W	A/D 连续扫描 选择扫描模式中的单周期扫描模式或者连续扫描模式。只在扫描模式时有效。 0: 单周期扫描 1: 连续扫描 必须在 ADST 为 0 的状态下转换运行模式。
2 ~ 0	CH[2:0]	000	R/W	通道的选择 2 ~ 0 选择 A/D 转换的模拟输入通道 (参照表 13.3)。 必须在 ADST 为 0 的状态下转换运行模式。

【注】 * 为了清除标志, 只能在读 1 后写 0。

13.3.3 A/D 控制寄存器 _0、1 (ADCR_0、1)

ADCR 控制各模块的 A/D 转换运行。

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

位	位名	初始值	R/W	说 明
15、14	—	全 0	R	保留位 读写的值总是 0。
13	ADST	0	R/W	A/D 的开始 如果清 0，就中止 A/D 转换，进入待机状态；如果置 1，就开始 A/D 转换。在单通道模式和单周期扫描模式中，当所选通道的 A/D 转换结束时，此位就自动清除。在连续扫描模式中，在通过软件、复位、软件待机模式或者模块待机模式清除此位前，依次连续转换所选的通道。
12 ~ 0	—	全 0	R	保留位 读写的值总是 0。

表 13.3 通道选择一览表

bit2	bit1	bit0	模拟输入通道			
CH2	CH1	CH0	单通道模式		4 通道扫描模式 *	
			A/D_0	A/D_1	A/D_0	A/D_1
0	0	0	AN0	AN4	AN0	AN4
		1	AN1	AN5	AN0、AN1	AN4、AN5
	1	0	AN2	AN6	AN0 ~ AN2	AN4 ~ AN6
		1	AN3	AN7	AN0 ~ AN3	AN4 ~ AN7
1	0	0	禁止设定	禁止设定	禁止设定	禁止设定
		1				
	1	0				
		1				

bit2	bit1	bit0	模拟输入通道	
CH2	CH1	CH0	2 通道扫描模式 *	
			A/D_0	A/D_1
0	0	0	AN0	AN4
		1	AN0、AN1	AN4、AN5
	1	0	AN2	AN6
		1	AN2、AN3	AN6、AN7
1	0	0	禁止设定	禁止设定
		1		
	1	0		
		1		

【注】 * 能通过 ADCS 位设定连续扫描模式或者单周期扫描模式。

13.3.4 A/D 触发选择寄存器_0（ADTSR_0）

ADTSR_0 允许由外部触发产生的 A/D 转换开始。

尤其是在 2 通道扫描模式中，A/D 模块 0 内的 4 个通道分为组 0 和组 1 共 2 组，并能指定各自独立的 A/D 触发。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRG11S[3:0]				TRG01S[3:0]				TRG1S[3:0]				TRG0S[3:0]			
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 12	TRG11S[3:0]	0000	R/W	A/D 触发 1 的组 1 选择 3、2、1、0 对 A/D 模块 1 的 2 通道扫描模式中的组 1，选择外部触发或者 MTU2 的 A/D 转换开始触发。 0000: 外部触发管脚 ($\overline{\text{ADTRG}}$) 的输入 0001: MTU2 各通道的 TRGA 输入捕捉 / 比较匹配，或者在互补 PWM 模式中 TCNT_4 发生下溢（波谷）（TRGAN） 0010: MTU2 通道 0 的比较匹配（TRG0N） 0011: MTU2 的 A/D 转换开始请求延迟（TRG4AN） 0100: MTU2 的 A/D 转换开始请求延迟（TRG4BN） 0101: 禁止设定 0110: 禁止设定 0111: 禁止设定 1xxx: 禁止设定 必须在 A/D 控制寄存器（ADCR）的 ADST 为 0 的状态下进行选择器的转换。 在 2 通道扫描模式中，为了不同时产生组 0 和组 1 的转换请求，必须给组 0 和组 1 的转换请求分别指定不同的源。
11 ~ 8	TRG01S[3:0]	0000	R/W	A/D 触发 0 的组 1 选择 3、2、1、0 对 A/D 模块 0 的 2 通道扫描模式中的组 1，选择外部触发或者 MTU2 的 A/D 转换开始触发。 0000: 外部触发管脚 ($\overline{\text{ADTRG}}$) 的输入 0001: 在 MTU2 各通道的 TRGA 输入捕捉 / 比较匹配，或者在互补 PWM 模式中 TCNT_4 发生下溢（波谷）（TRGAN） 0010: MTU2 通道 0 的比较匹配（TRG0N） 0011: MTU2 的 A/D 转换开始请求延迟（TRG4AN） 0100: MTU2 的 A/D 转换开始请求延迟（TRG4BN） 0101: 禁止设定 0110: 禁止设定 0111: 禁止设定 1xxx: 禁止设定 必须在 A/D 控制寄存器（ADCR）的 ADST 为 0 的状态下进行选择器的转换。 在 2 通道扫描模式中，为了不同时产生组 0 和组 1 的转换请求，必须给组 0 和组 1 的转换请求分别指定不同的源。

位	位名	初始值	R/W	说 明
7 ~ 4	TRG1S[3:0]	0000	R/W	A/D 触发 1 的选择 3、2、1、0 对 A/D 模块 1 的单通道模式、4 通道扫描模式或者 2 通道扫描模式中的组 0，选择外部触发或者 MTU2 的 A/D 转换开始触发。 0000: 外部触发管脚 ($\overline{\text{ADTRG}}$) 的输入 0001: 在 MTU2 各通道的 TRGA 输入捕捉 / 比较匹配，或者在互补 PWM 模式中 TCNT_4 发生下溢 (波谷) (TRGAN) 0010: MTU2 通道 0 的比较匹配 (TRG0N) 0011: MTU2 的 A/D 转换开始请求延迟 (TRG4AN) 0100: MTU2 的 A/D 转换开始请求延迟 (TRG4BN) 0101: 禁止设定 0110: 禁止设定 0111: 禁止设定 1xxx: 禁止设定 必须在 A/D 控制寄存器 (ADCR) 的 ADST 为 0 的状态下进行选择器的转换。 在 2 通道扫描模式中，为了不同时产生组 0 和组 1 的转换请求，必须给组 0 和组 1 的转换请求分别指定不同的源。
3 ~ 0	TRG0S[3:0]	0000	R/W	A/D 触发 0 的选择 3、2、1、0 对 A/D 模块 0 的单通道模式、4 通道扫描模式或者 2 通道扫描模式中的组 0，选择外部触发或者 MTU2 的 A/D 转换开始触发。 0000: 外部触发管脚 ($\overline{\text{ADTRG}}$) 的输入 0001: 在 MTU2 各通道的 TRGA 输入捕捉 / 比较匹配，或者在互补 PWM 模式中 TCNT_4 发生下溢 (波谷) (TRGAN) 0010: MTU2 通道 0 的比较匹配 (TRG0N) 0011: MTU2 的 A/D 转换开始请求延迟 (TRG4AN) 0100: MTU2 的 A/D 转换开始请求延迟 (TRG4BN) 0101: 禁止设定 0110: 禁止设定 0111: 禁止设定 1xxx: 禁止设定 必须在 A/D 控制寄存器 (ADCR) 的 ADST 为 0 的状态下进行选择器的转换。 在 2 通道扫描模式中，为了不同时产生组 0 和组 1 的转换请求，必须给组 0 和组 1 的转换请求分别指定不同的源。

【符号说明】 x: Don't care

13.4 运行说明

A/D 转换器采用逐次逼近方式，分辨率为 10 位。运行模式有单通道模式和扫描模式。扫描模式有连续扫描模式和单周期扫描模式。为了避免误动作，必须在 ADCR 的 ADST 位为 0 的状态下转换运行模式或者模拟输入通道。

13.4.1 单通道模式

单通道模式将指定的 1 个通道的模拟输入进行如下的 1 次 A/D 转换：

1. 如果通过软件、MTU2 或者外部触发输入将 ADCR 的 ADST 位设定为 1，就开始对所选通道进行 A/D 转换。
2. 如果 A/D 转换结束，就将 A/D 转换结果传送到对应该通道的 A/D 数据寄存器。
3. 在 A/D 转换结束后，ADCSR 的 ADF 位被置 1。此时，如果 ADIE 位已被置 1，就产生 ADI 中断请求。
4. ADST 位在 A/D 转换中保持 1，当转换结束时，就自动清除并且 A/D 转换器变为待机状态。如果在 A/D 转换中 ADST 位被清 0，就中止转换并且 A/D 转换器变为待机状态。

13.4.2 连续扫描模式

连续扫描模式将指定通道的模拟输入依次进行如下的 A/D 转换：

1. 如果通过软件、MTU2 或者外部触发输入将 ADCR 的 ADST 位设定为 1，就按照模拟输入通道号从小到大的顺序（例如，AN0、AN1…AN3）进行 A/D 转换。
2. 如果各通道的 A/D 转换结束，就将 A/D 转换结果依次传送到对应该通道的 A/D 数据寄存器。
3. 如果所选通道的 A/D 转换全部结束，ADCSR 的 ADF 位就被置 1。此时，如果 ADIE 位已被置 1，就产生 ADI 中断请求。A/D 转换器再次从组中的第 1 个通道开始 A/D 转换。
4. ADST 位不自动清除，在置为 1 的期间重复执行 2. ~ 3.。如果 ADST 位被清 0，就中止 A/D 转换并且 A/D 转换器变为待机状态。

因为在 2 通道扫描模式中，通道被分为组 0 和组 1，所以即使在连续扫描模式中组 0 运行，也能保持组 1 的 A/D 数据寄存器；即使在连续扫描模式中组 1 运行，也能保持组 0 的 A/D 数据寄存器。另外，如果在组 0 进行转换时产生组 1 的转换请求，就忽视组 1 的转换请求。为了不同时产生组 0 和组 1 的转换请求，必须给组 0 和组 1 的转换请求分别指定不同的源。

另外，在 2 通道扫描模式中由软件开始 A/D 转换时，通过设定 ADCSR_0、1 的 CH2、1、0 选择组 0 和组 1；在由触发开始 A/D 转换时，与 ADCSR_0、1 的 CH2、1、0 的设定无关，通过 ADTSR 的 A/D 触发 0 的选择 3、2、1、0 以及 A/D 触发 1 的选择 3、2、1、0 设定的触发源，开始组 0 的 A/D 转换，并且通过 ADTSR 的 A/D 触发 0 的组 1 选择 3、2、1、0 以及 A/D 触发 1 的组 1 选择 3、2、1、0 设定的触发源，开始组 1 的 A/D 转换。

13.4.3 单周期扫描模式

单周期扫描模式将指定的通道（最多 4 个通道）的模拟输入进行如下的 1 次 A/D 转换。

1. 如果通过软件、MTU2 或者外部触发输入将 ADCR 的 ADST 位设定为 1，就按照模拟输入通道号从小到大的顺序（例如，AN0、AN1…AN3）进行 A/D 转换。
2. 如果各通道的 A/D 转换结束，就将 A/D 转换结果依次传送到对应该通道的 A/D 数据寄存器。
3. 如果所选通道的 A/D 转换全部结束，ADCSR 的 ADF 位就被置 1。此时，如果 ADIE 位已被置 1，就产生 ADI 中断请求。
4. 如果转换结束，ADST 位就自动清除并且 A/D 转换器变为待机状态。如果在 A/D 转换中 ADST 位被清 0，就中止转换并且 A/D 转换器变为待机状态。

因为在 2 通道扫描模式中，通道被分为组 0 和组 1，所以即使在单周期扫描模式中组 0 运行，也能保持组 1 的 A/D 数据寄存器；即使在连续扫描模式中组 1 运行，也能保持组 0 的 A/D 数据寄存器。另外，如果在组 0 进行转换时产生组 1 的转换请求，就忽视组 1 的转换请求。为了不同时产生组 0 和组 1 的转换请求，必须给组 0 和组 1 的转换请求分别指定不同的源。

另外，在 2 通道扫描模式中由软件开始 A/D 转换时，通过设定 ADCSR_0、1 的 CH2、1、0 选择组 0 和组 1。在由触发开始 A/D 转换时，与 ADCSR_0、1 的 CH2、1、0 的设定无关，通过 ADTSR 的 A/D 触发 0 的选择 3、2、1、0 以及 A/D 触发 1 的选择 3、2、1、0 设定的触发源，开始组 0 的 A/D 转换，并且通过 ADTSR 的 A/D 触发 0 的组 1 的选择 3、2、1、0 以及 A/D 触发 1 的组 1 的选择 3、2、1、0 设定的触发源，开始组 1 的 A/D 转换。

13.4.4 输入采样和 A/D 转换时间

A/D 转换器内置各模块的采样和保持电路。如果在将 ADCR 的 ADST 位置 1 后经过 A/D 转换开始的延迟时间 (t_D)，A/D 转换器就对输入进行采样，然后开始转换。A/D 转换的时序如图 13.2、A/D 转换时间如表 13.4 所示。

如图 13.2 所示，A/D 转换时间 (t_{CONV}) 包含 t_D 和输入采样时间 (t_{SPL})。在此， t_D 由 ADCR 的写时序决定，而不是定值。因此，转换时间在表 13.4 所示的范围内发生变化。

关于扫描模式的转换时间，表 13.4 所示的值为第 1 次转换时间，表 13.5 所示的值为第 2 次以后的转换时间。

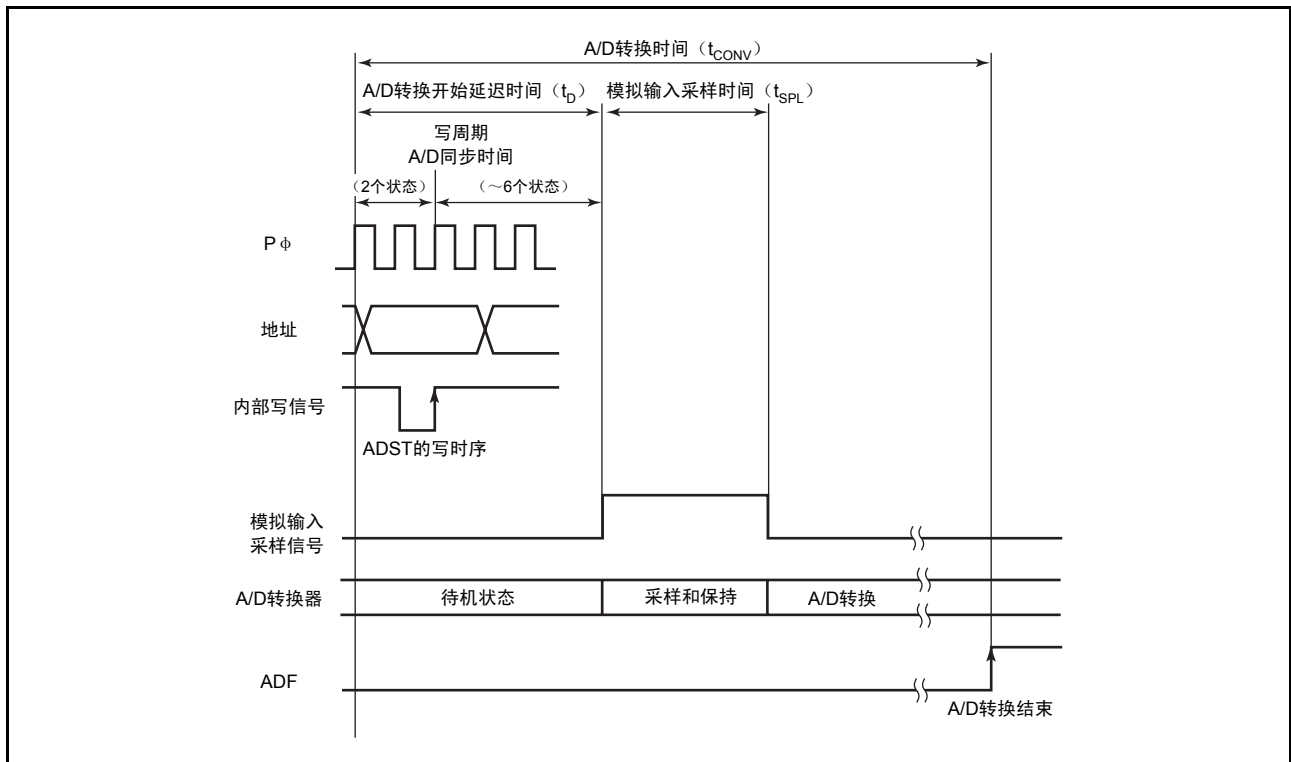


图 13.2 A/D 转换时序

表 13.4 A/D 转换时间（单通道模式）

项 目	符号	STC=0											
		CKSL1=0						CKSL1=1					
		CKSL0=0			CKSL0=1			CKSL0=0			CKSL0=1		
		Min.	Typ	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ	Max.
A/D 转换开始的延迟时间	t_D	2	—	6	2	—	5	2	—	4	2	—	3
输入采样时间	t_{SPL}	—	24	—	—	18	—	—	12	—	—	6	—
A/D 转换时间	t_{CONV}	202	—	206	152	—	155	102	—	104	52	—	53

项 目	符号	STC=1											
		CKSL1=0						CKSL1=1					
		CKSL0=0			CKSL0=1			CKSL0=0			CKSL0=1		
		Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ	Max.	Min.	Typ	Max.
A/D 转换开始的延迟时间	t_D	2	—	6	2	—	5	2	—	4	2	—	3
输入采样时间	t_{SPL}	—	36	—	—	27	—	—	18	—	—	9	—
A/D 转换时间	t_{CONV}	258	—	262	194	—	197	130	—	132	66	—	67

【注】 表中的数值单位是相对于 $P\phi$ 的状态。

表 13.5 A/D 转换时间（扫描模式）

STC	CKSL1	CKSL0	转换时间（状态）	转换时间的计算例	
				在 $P\phi=25\text{MHz}$ 时	在 $P\phi=40\text{MHz}$ 时
0	0	0	200（固定）	$8\mu\text{s}$	$5\mu\text{s}$
		1	150（固定）	$6\mu\text{s}$	$3.8\mu\text{s}$
	1	0	100（固定）	$4\mu\text{s}$	$2.5\mu\text{s}$
		1	50（固定）	$2\mu\text{s}$	禁止设定
1	0	0	256（固定）	$10.2\mu\text{s}$	$6.4\mu\text{s}$
		1	192（固定）	$7.7\mu\text{s}$	$4.8\mu\text{s}$
	1	0	128（固定）	$5.1\mu\text{s}$	$3.2\mu\text{s}$
		1	64（固定）	$2.6\mu\text{s}$	禁止设定

13.4.5 通过 MTU2 启动 A/D 转换器

能通过 MTU2 间隔定时器的 A/D 转换请求，独立启动 A/D 转换器。

要通过 MTU2 启动 A/D 转换器时，将 A/D 控制 / 状态寄存器（ADCSR）的 TRGE 位设定为 1，然后设定 A/D 触发选择寄存器（ADTSR）。在此状态下，如果发生 MTU2 间隔定时器的 A/D 转换请求，就将 ADCSR 的 ADST 位设定为 1。从 ADST 位被置 1 后到 A/D 转换开始前的时序与通过软件给 ADST 位写 1 时的时序相同。

13.4.6 外部触发的输入时序

也能通过外部触发输入来开始 A/D 转换。在 A/D 控制 / 状态寄存器（ADCSR）的 TRGE 位设定为 1 后，将 A/D 触发选择寄存器 _0（ADTSR_0）设定为外部触发管脚输入时，从 ADTRG 管脚输入外部触发。在 ADTRG 的下降沿，ADCR 的 ADST 位被置 1，开始 A/D 转换。与单通道模式和扫描模式无关，其他运行都和通过软件将 ADCSR 的 ADST 位设定为 1 的情况相同。此时序如图 13.3 所示。

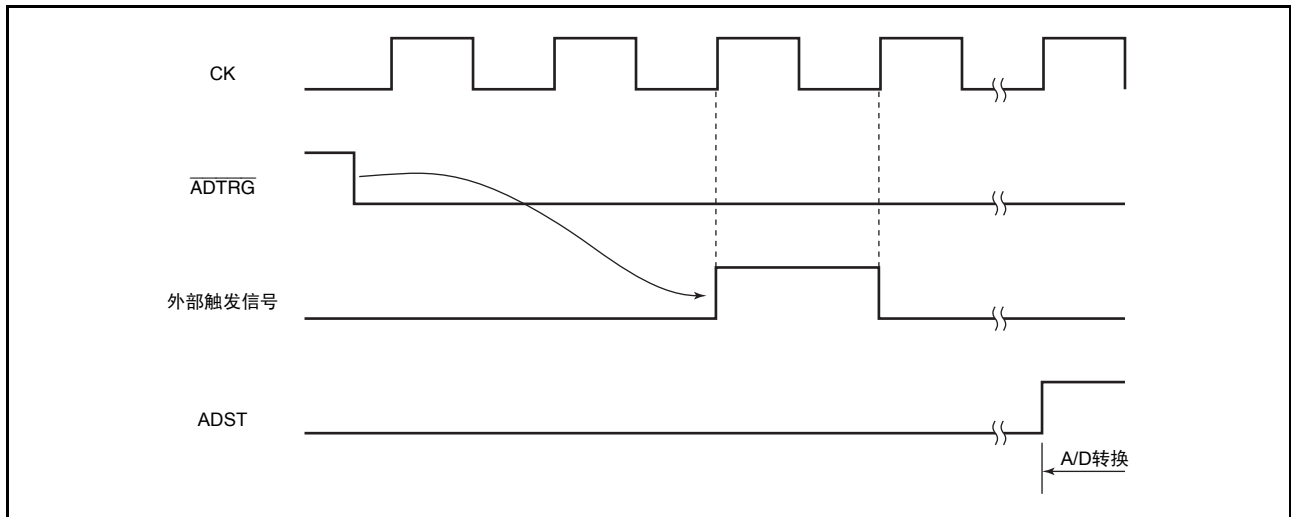


图 13.3 外部触发输入的时序

13.5 中断源

A/D 转换器能产生 A/D 转换结束的中断请求。如果将 A/D 控制 / 状态寄存器（ADCSR）的 ADIE 位设定为 1 就允许 ADI；如果清 0 就禁止 ADI。

表 13.6 A/D 转换器的中断源

名 称	中断源	中断标志
ADI0	A/D_0 转换结束	ADCSR_0 的 ADF
ADI1	A/D_1 转换结束	ADCSR_1 的 ADF

13.6 A/D 转换精度的定义

本 LSI 的 A/D 转换精度定义如下：

- 分辨率
A/D 转换器的数字输出码的位数
- 量化误差
A/D 转换固有的偏差，为 $1/2\text{LSB}$ （图 13.4）。
- 偏移误差
在数字输出从最小电压值 B'0000000000（H'00）变为 B'0000000001（H'01）时，与模拟输入电压值的理想 A/D 转换特性的偏差（图 13.5）
- 满刻度误差
在数字输出从 B'1111111110（H'3FE）变为 B'1111111111（H'3FF）时，与模拟输入电压值的理想 A/D 转换特性引起的偏差（图 13.5）
- 非线性误差
在零电压和满刻度电压之间，与理想 A/D 转换特性的误差。但是，不包括偏移误差、满刻度误差和量化误差（图 13.5）。
- 绝对精度
数字值与模拟输入值的偏差，包括偏移误差、满刻度误差、量化误差和非线性误差。

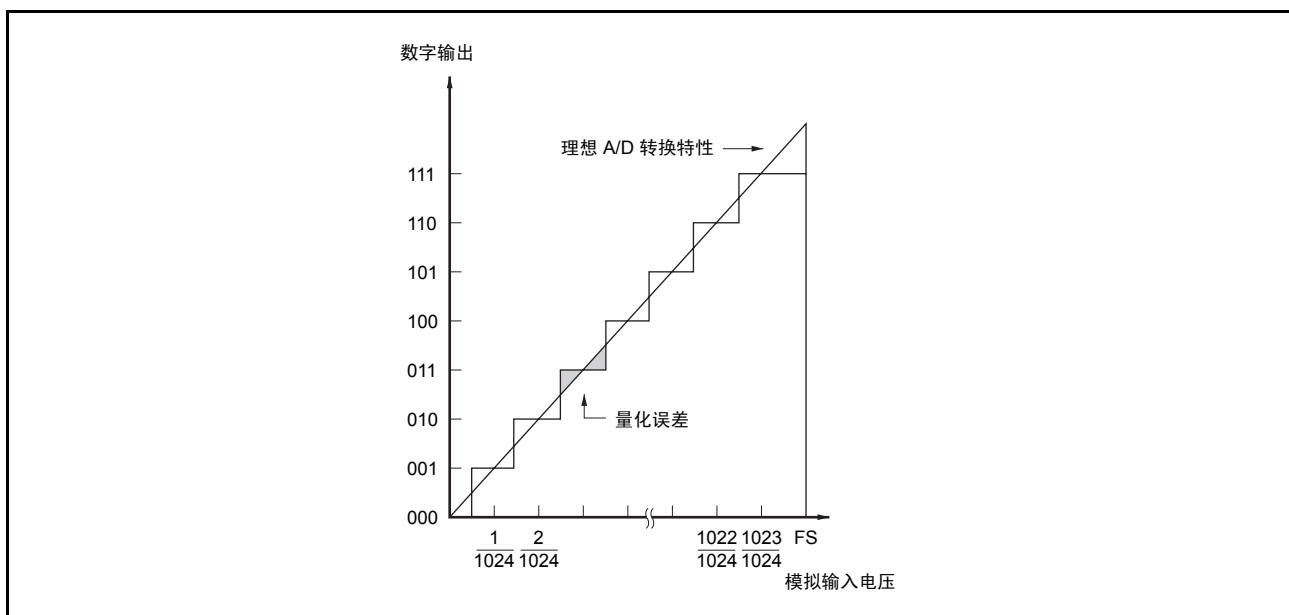


图 13.4 A/D 转换精度的定义

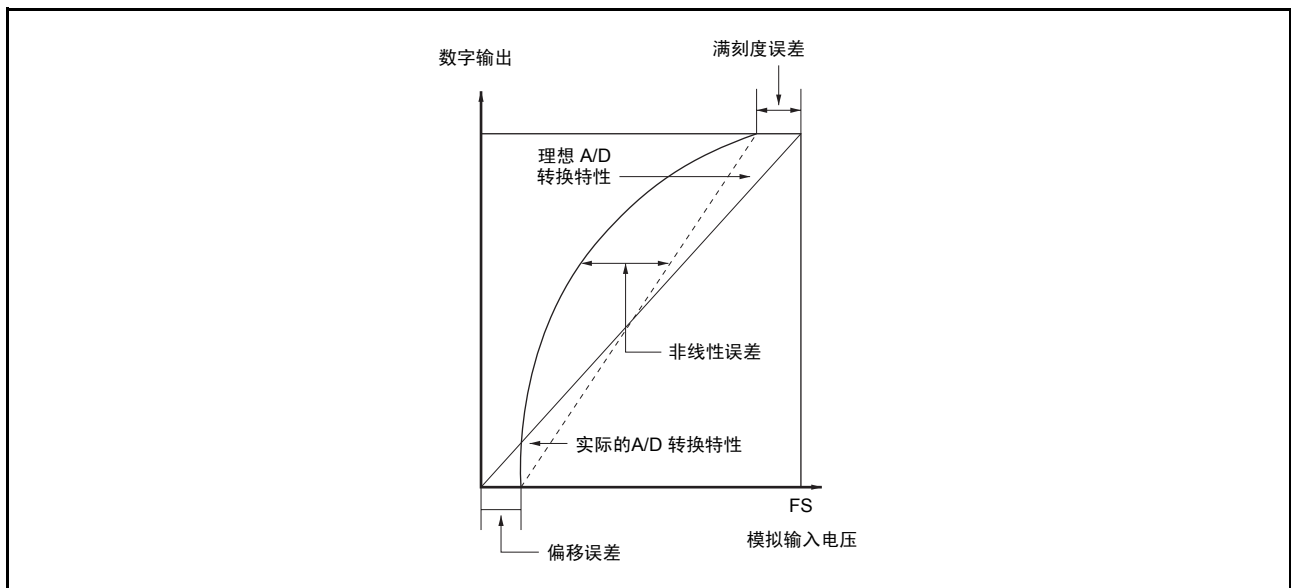


图 13.5 A/D 转换精度的定义

13.7 使用注意事项

13.7.1 模块待机模式的设定

A/D 转换器能通过待机控制寄存器允许或者禁止本模块的运行，初始值为停止 A/D 转换器的运行。另外，能通过解除模块待机模式进行寄存器的存取。详细内容请参照“第 19 章 低功耗模式”。

13.7.2 关于容许信号源阻抗

对于信号源阻抗不超过 $1\text{k}\Omega$ 的输入信号，本 LSI 的模拟输入能保证转换精度，这是为了在采样时间内，对 A/D 转换器的采样和保持电路的输入电容进行充电而制定的规格。在传感器的输出阻抗超过 $1\text{k}\Omega$ 时，有可能发生充电不足并且不能保证 A/D 转换精度的情况。在单通道模式中进行转换并且在外接大电容的情况下，因为输入负载实际上只有 $10\text{k}\Omega$ 的内部输入电阻，所以信号源阻抗可忽略不计。但是，由于形成低通滤波器，所以有可能无法跟踪大微分系数的模拟信号（例如，电压的变动率在 $5\text{mV}/\mu\text{s}$ 以上）（图 13.6）。在转换高速模拟信号或者在扫描模式中进行转换时，必须插入一个低阻抗的缓冲器。

13.7.3 关于对绝对精度的影响

由于附加电容会导致与 GND 的耦合, 因此, 如果在 GND 中有噪声, 就有可能降低绝对精度, 所以必须与 AV_{SS} 等电稳定的 GND 连接。

另外, 必须注意: 在安装电路板上滤波器电路不要干涉数字信号也不要充当天线。

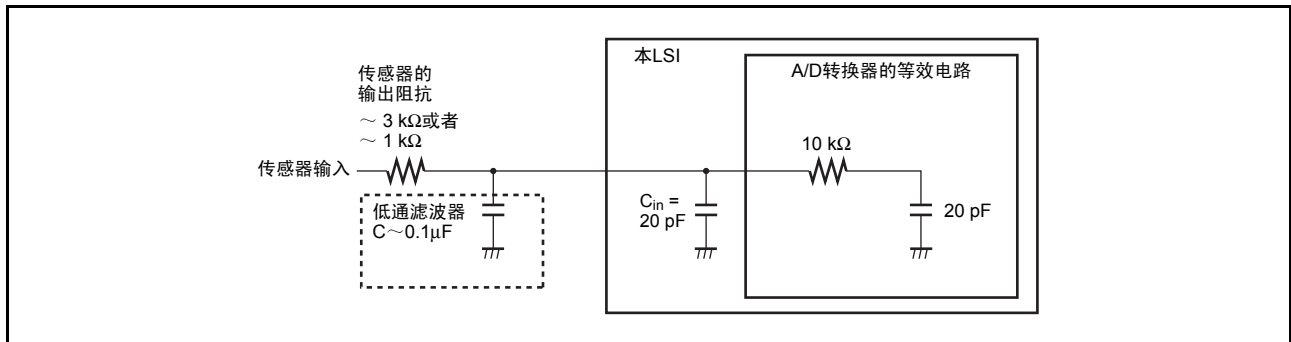


图 13.6 模拟输入电路的例子

13.7.4 模拟电源管脚等的设定范围

如果超出以下电压的设定范围使用 LSI, 就会给 LSI 的可靠性带来不良影响。

- 模拟输入电压的设定范围

在 A/D 转换中, 必须将模拟输入管脚 AN_n 的外加电压设定在 $AV_{SS} \leq V_{AN} \leq AV_{CC}$ 的范围内。

- AV_{CC}、AV_{SS} 和 V_{CC}、V_{SS} 的关系

AV_{CC}、AV_{SS} 和 V_{CC}、V_{SS} 的关系为 AV_{SS}=V_{SS}, 并且在不使用 A/D 转换器时, AV_{CC} 和 AV_{SS} 管脚不能开路。

13.7.5 电路板设计的注意事项

在设定电路板时, 必须尽量将数字电路和模拟电路分开, 不能使数字电路的信号线和模拟电路的信号布线交叉或者靠近。因电感等引起模拟电路的误动作并给 A/D 转换值带来不良影响。模拟输入管脚 (AN₀ ~ AN₇) 和模拟电源电压 (AV_{CC}) 必须通过模拟接地 (AV_{SS}) 和数字电路分开, 并且模拟接地 (AV_{SS}) 必须单点连接到电路板上稳定的接地 (V_{SS})。

13.7.6 噪声对策的注意事项

为了防止因过大电涌等异常电压引起对模拟输入管脚（AN0 ~ AN7）的破坏，如图 13.7 所示，必须将保护电路连接在 AVcc–AVss 之间，并将连接 AVcc 的旁路电容以及连接 AN0 ~ AN7 的滤波器电容连接到 AVss。

另外，如果连接用于滤波器的电容，AN0 ~ AN7 的输入电流就被平均，所以有可能产生误差。在扫描模式等模式中频繁地进行 A/D 转换时，如果对 A/D 转换器内部的采样和保持电路的电容进行充放电的电流大于从输入阻抗（R_{in}）输入的电流，模拟输入管脚的电压就会产生误差。因此，在决定电路常数时，请进行充分的探讨。

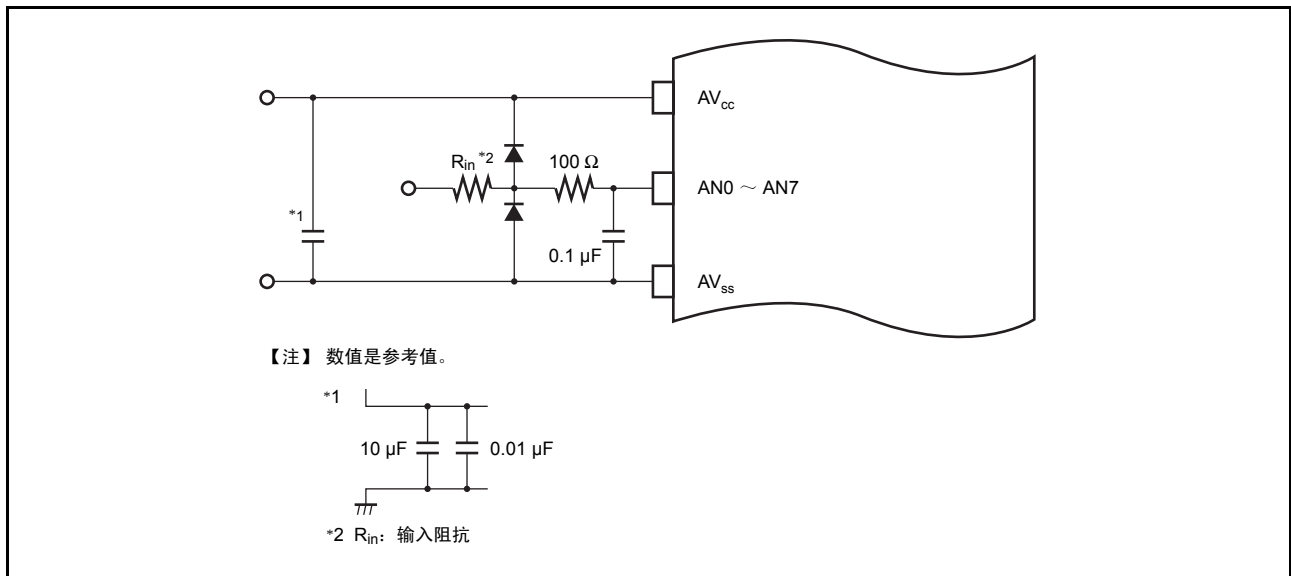


图 13.7 模拟输入保护电路的例子

表 13.7 模拟管脚的规格

项 目	Min.	Max.	单位	条件
模拟输入电容	—	20	pF	—
容许信号源阻抗	—	3	kΩ	转换时间 ≥ 4.0 μs
		1		转换时间 < 4.0 μs

第 14 章 比较匹配定时器 (CMT)

本 LSI 内置由 2 个通道的 16 位定时器构成的比较匹配定时器 (CMT)。CMT 有 16 位的计数器，能按各设定的周期产生中断。

14.1 特点

- 2 个通道可独立选择 4 种计数器输入时钟
可选择 4 种内部时钟 ($P\phi/8$ 、 $P\phi/32$ 、 $P\phi/128$ 、 $P\phi/512$)
- 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 14.1 所示。

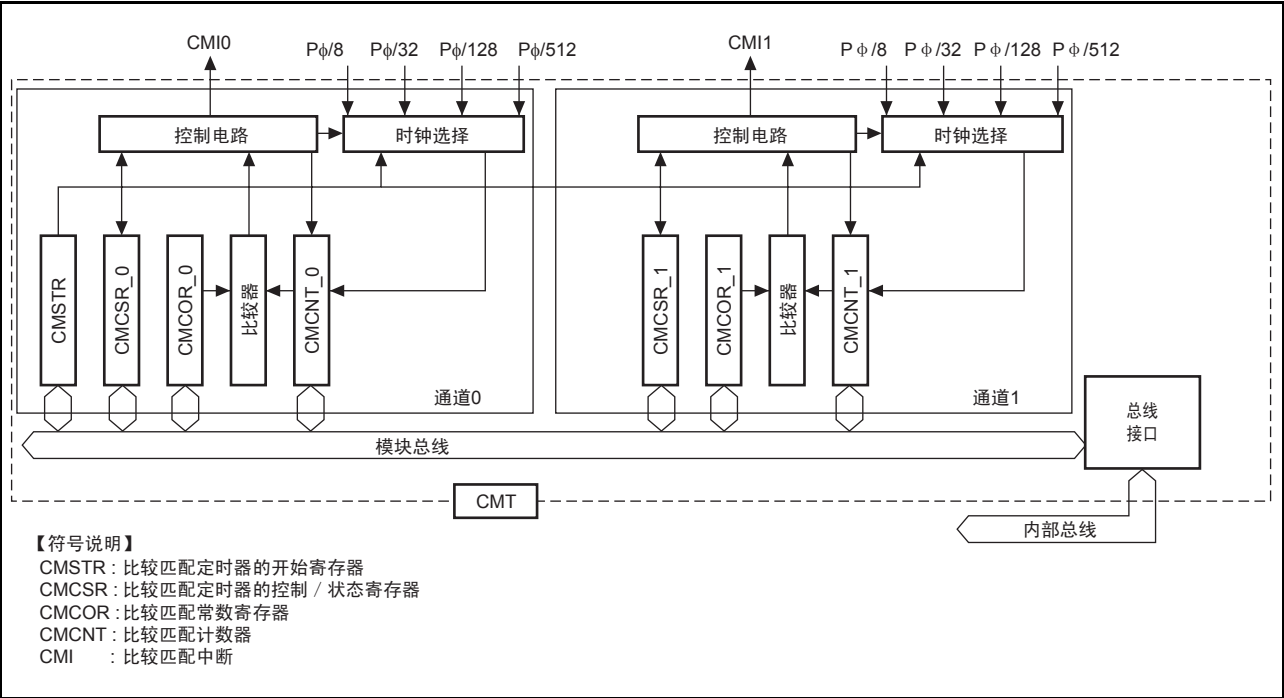


图 14.1 CMT 的框图

14.2 寄存器说明

CMT 有以下寄存器。有关这些寄存器的地址和各处理模式中的寄存器状态，请参照“第 20 章 寄存器一览”。另外，本章中省略了通道数。

表 14.1 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

14.2.1 比较匹配定时器的开始寄存器（CMSTR）

CMSTR 是 16 位寄存器，选择比较匹配计数器（CMCNT）的运行 / 停止。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	STR1	STR0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 2	—	全 0	R	保留位 读写的值总是 0。
1	STR1	0	R/W	计数开始 1 选择比较匹配计数器_1 的运行 / 停止。 0: CMCNT_1 停止计数 1: CMCNT_1 开始计数
0	STR0	0	R/W	计数开始 0 选择比较匹配计数器_0 的运行 / 停止。 0: CMCNT_0 停止计数 1: CMCNT_0 开始计数

14.2.2 比较匹配定时器的控制 / 状态寄存器（CMCSR）

CMCSR 是 16 位寄存器，表示比较匹配的产生以及设定中断和计数器输入时钟。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	CMF	CMIE	-	-	-	-	CKS[1:0]	
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	(R/W)*1	R/W	R	R	R	R	R/W	R/W

【注】*1 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说 明
15 ~ 8	—	全 0	R	保留位 读写的值总是 0。
7	CMF	0	(R/W)*1	比较匹配标志 表示 CMCNT 和 CMCOR 的值是否一致。 0: CMCNT 和 CMCOR 的值不一致 [清除条件] • 当读 CMF=1 后，写 0 时*2 [置位条件] 1: CMCNT 和 CMCOR 的值一致
6	CMIE	0	R/W	比较匹配的中断允许 选择在 CMCNT 和 CMCOR 值一致时（CMF=1）是允许还是禁止比较匹配中断（CMI）的产生。 0: 禁止比较匹配中断（CMI） 1: 允许比较匹配中断（CMI）
5 ~ 2	—	全 0	R	保留位 读写的值总是 0。
1、0	CKS[1:0]	00	R/W	时钟选择 1、0 从分频外围运行时钟（P ϕ ）后的 4 种内部时钟中，选择 CMCNT 的输入时钟。当 CMSTR 的 STR 位被置 1 时，通过 CKS1 和 CKS0 位选择的时钟，CMCNT 开始计数。 00: P ϕ /8 01: P ϕ /32 10: P ϕ /128 11: P ϕ /512

【注】*1 为了清除标志，只能在读 1 后写 0。

*2 如果在读 1 后写 0 前发生由下一个比较匹配引起的标志置位，即使写 0 也不清除标志，所以必须重新读 1 后再写 0。

14.2.3 比较匹配计数器 (CMCNT)

CMCNT 是 16 位寄存器，用作递增计数器。如果通过 CMCSR 的 CKS1、CKS0 位选择计数器输入时钟，并且将 CMST 的 STR 位设定为 1，CMCN 就通过所选的时钟开始计数。当 CMCNT 的值和比较匹配常数寄存器 (CMCOR) 的值一致时，CMCNT 被清除为 H'0000，并且 CMCSR 的 CMF 标志被置 1。

CMCNT 的初始值是 H'0000。

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

14.2.4 比较匹配常数寄存器 (CMCOR)

CMCOR 是 16 位寄存器，设定和 CMCNT 比较匹配前的时间，初始值是 H'FFFF。

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

14.3 运行说明

14.3.1 周期计数运行

如果通过 CMCSR 的 CKS1、CKS0 位选择内部时钟，并且将 CMSTR 的 STR 位设定为 1，CMCNT 就通过所选的时钟开始递增计数。当 CMCNT 的值和 CMCOR 的值一致时，CMCNT 被清除为 H'0000，并且 CMCSR 的 CMF 标志被置 1。此时，如果 CMCSR 寄存器的 CMIE 位已被置 1，就请求比较匹配中断 (CMI)，CMCNT 从 H'0000 重新开始递增计数。

比较匹配计数器的运行如图 14.2 所示。

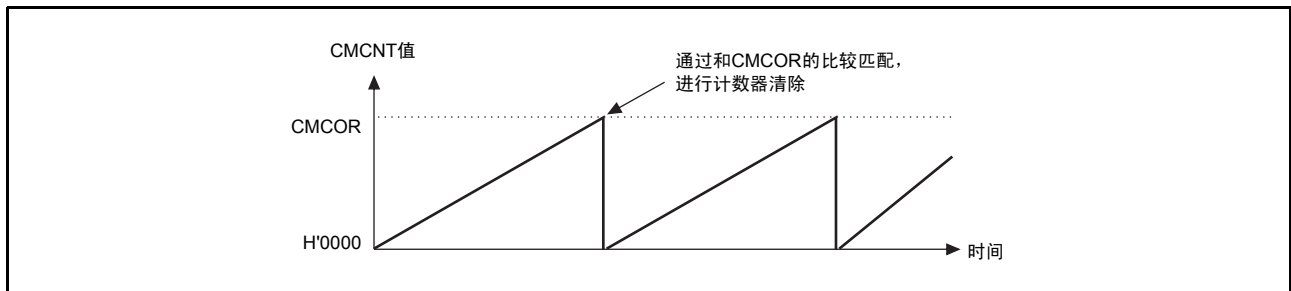


图 14.2 计数器的运行

14.3.2 CMCNT 的计数时序

能通过 CMCSR 的 CKS1、CKS0 位，从分频时钟 ($P\phi$) 后的 4 种时钟 ($P\phi/8$ 、 $P\phi/32$ 、 $P\phi/128$ 、 $P\phi/512$) 中选择一中时钟个。该时序如图 14.3 所示。

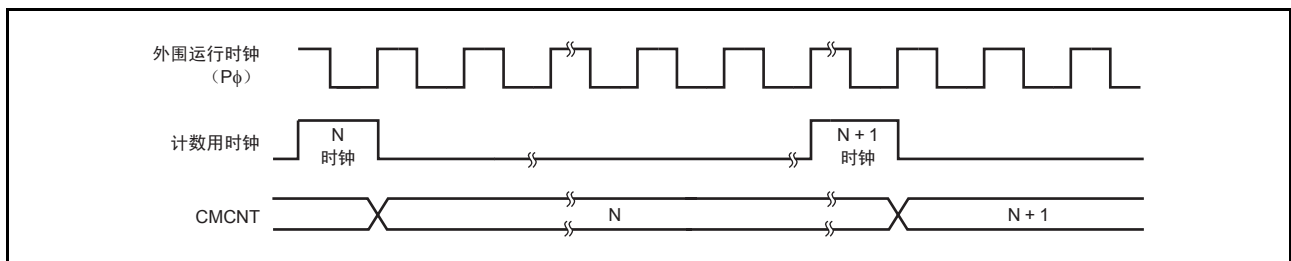


图 14.3 计数时序

14.4 中断

14.4.1 CMT 的中断源

CMT 有各通道的比较匹配中断，分别分配了独立的向量地址。当中断请求标志 (CMF) 和中断允许位 (CMIE) 都被置 1 时，就输出该中断请求。在通过中断请求启动 CPU 中断的情况下，能通过设定中断控制器更改通道之间的优先级。详细内容请参照“第 6 章 中断控制器 (INTC)”。

14.4.2 比较匹配标志的置位时序

如果 CMCOR 和 CMCNT 一致，就产生比较匹配信号，并且 CMCSR 的 CMF 位被置 1。在一致的最后状态（将 CMCNT 的值更新为 H'0000 时）产生比较匹配信号。即，如果在 CMCOR 和 CMCNT 一致后不输入用于 CMCNT 计数器的时钟，就不产生比较匹配信号。CMF 位的置位时序如图 14.4 所示。

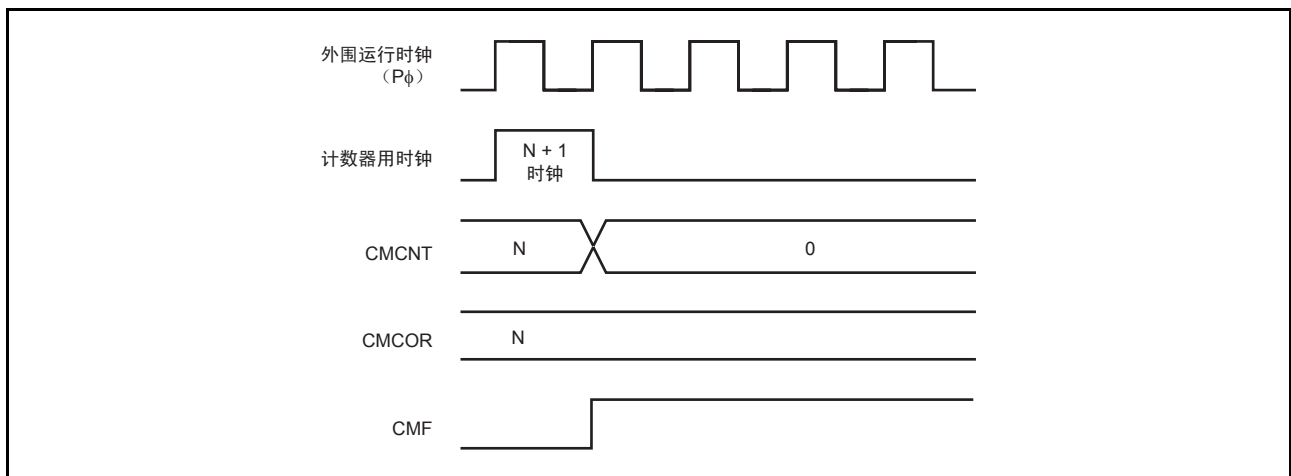


图 14.4 CMF 位的置位时序

14.4.3 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。

14.5 使用注意事项

14.5.1 模块待机模式的设定

CMT 能通过待机控制寄存器允许或者禁止本模块的运行，初始值为停止 CMT 的运行。另外，能通过解除模块待机模式进行寄存器的存取。详细内容请参照“第 19 章 低功耗模式”。

14.5.2 CMCNT 的写操作和比较匹配的竞争

如果在 CMCNT 计数器的写周期中的 T2 状态产生比较匹配信号，就不写 CMCNT 计数器而优先清除 CMCNT 计数器。此时序如图 14.5 所示。

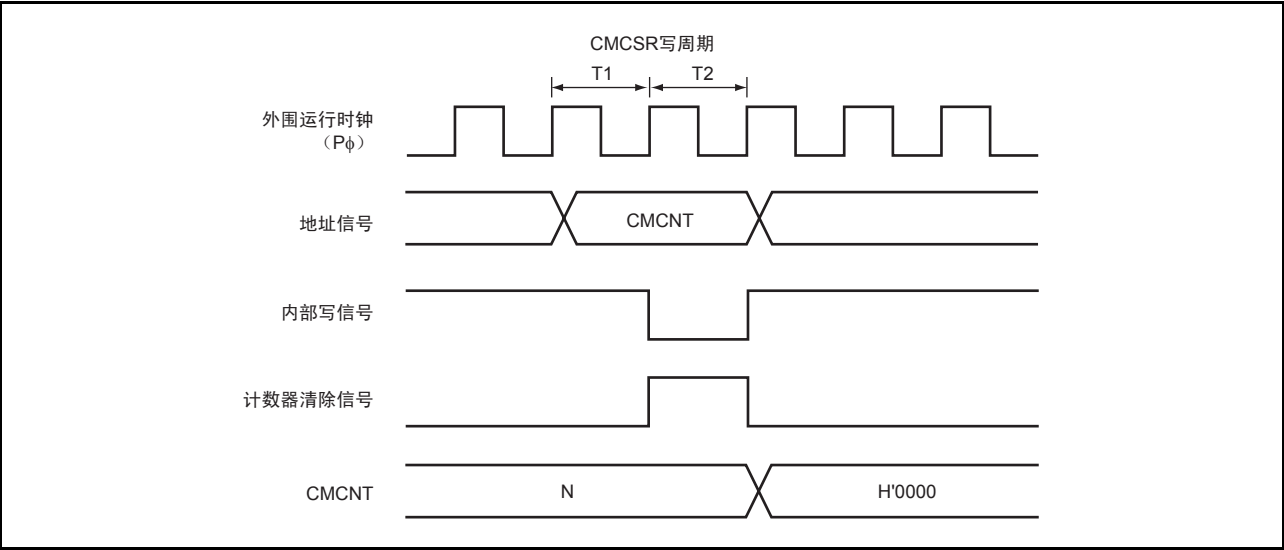


图 14.5 CMCNT 的写操作和比较匹配的竞争

14.5.3 CMCNT 的字写和递增计数的竞争

即使在 CMCNT 计数器的字写周期中的 T2 状态发生递增计数，也不进行递增计数而优先写计数器。此时序如图 14.6 所示。

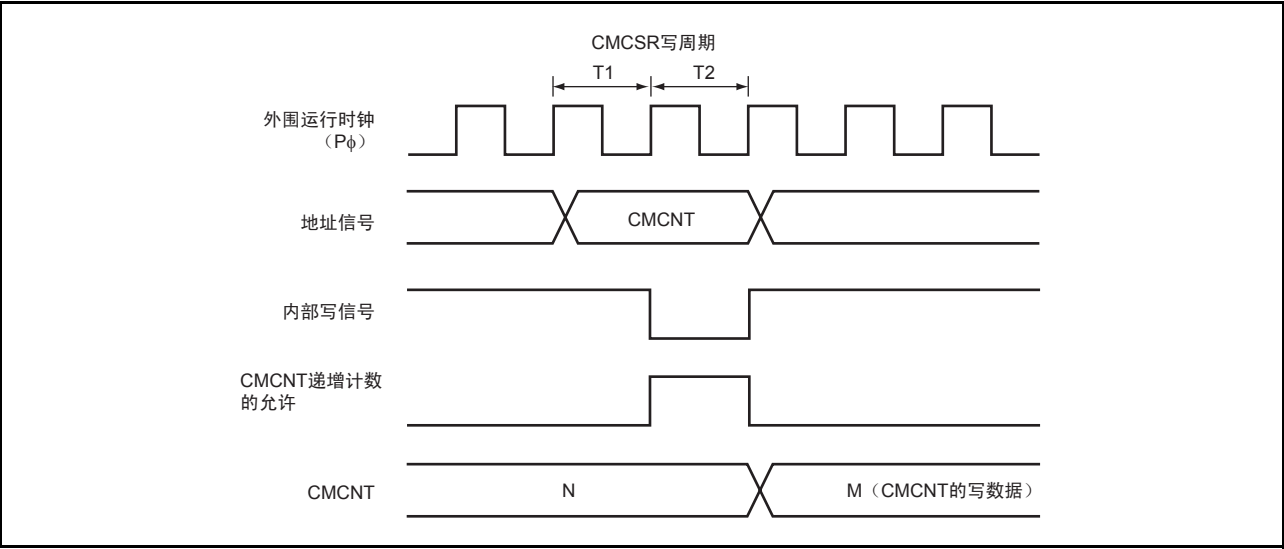


图 14.6 CMCNT 的字写和递增计数的竞争

14.5.4 CMCNT 的字节写和递增计数的竞争

即使在 CMCNT 的字节写周期中的 T2 状态发生递增计数，为字节写对象的计数器也不进行递增计数而优先写计数器。并且，不为字节写对象的计数器也不进行递增计数而为写操作之前的内容。

在 CMCNTH 写周期中的 T2 状态发生递增计数时的时序如图 14.7 所示。

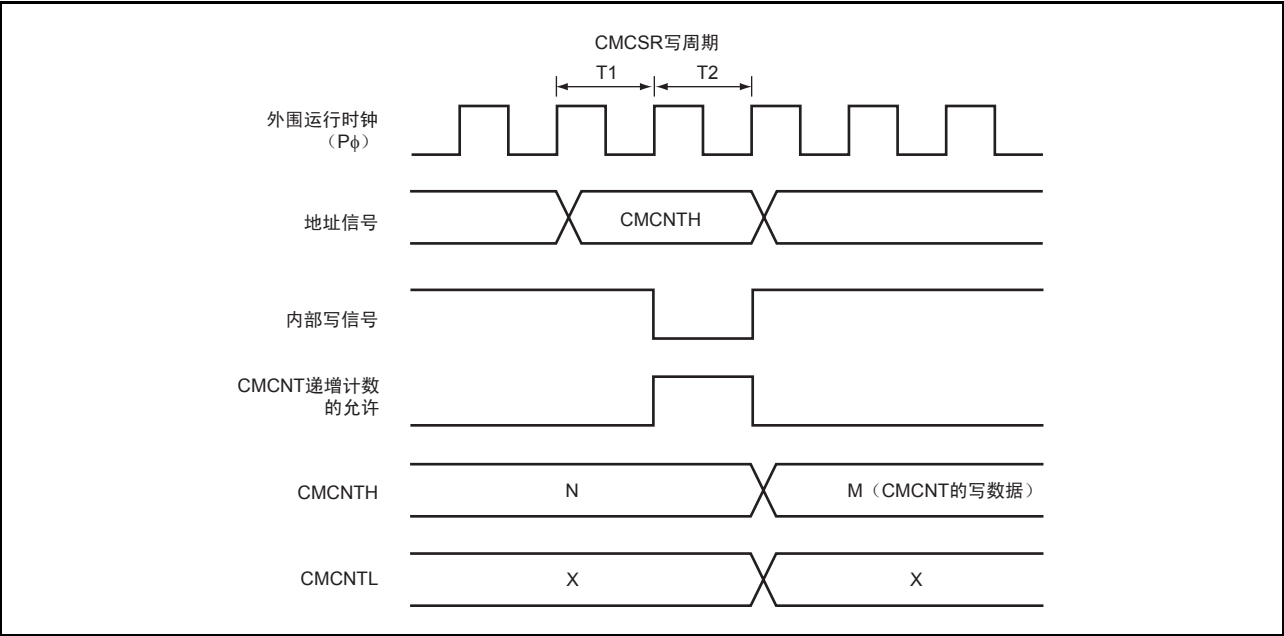


图 14.7 CMCNT 的字节写和递增计数的竞争

14.5.5 CMCNT 和 CMCOR 的比较匹配

在停止 CMCNT 计数运行的状态下，不能给 CMCNT 和 CMCOR 设定相同的值。否则，CMCSR 的 CMF 位就被置 1 并且 CMCNT 被清除为 H'0000。

第 15 章 管脚功能控制器 (PFC)

管脚功能控制器 (PFC) 由选择多路复用管脚功能及其输入 / 输出方向的寄存器构成。本 LSI 的多路复用管脚如表 15.1 和表 15.2 所示。

按运行模式分类的管脚功能一览表如表 15.3 和表 15.4 所示。

表 15.1 多路复用一览表 (SH7125)

端口	功能 1 (关联模块)	功能 2 (关联模块)	功能 3 (关联模块)	功能 4 (关联模块)	功能 5 (关联模块)
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)	SCK0 输入 / 输出 (SCI)	—	—
	PA3 输入 / 输出 (端口)	IRQ1 输入 (INTC)	RXD1 输入 (SCI)	$\overline{\text{TRST}}$ 输入 (H-UDI)	—
	PA4 输入 / 输出 (端口)	IRQ2 输入 (INTC)	TXD1 输出 (SCI)	TMS 输入 (H-UDI)	—
	PA5 输入 / 输出 (端口)	IRQ3 输入 (INTC)	SCK1 输入 / 输出 (SCI)	—	—
	PA6 输入 / 输出 (端口)	TCLKA 输入 (MTU2)	—	—	—
	PA7 输入 / 输出 (端口)	TCLKB 输入 (MTU2)	SCK2 输入 / 输出 (SCI)	TCK 输入 (H-UDI)	—
	PA8 输入 / 输出 (端口)	TCLKC 输入 (MTU2)	RXD2 输入 (SCI)	TDI 输入 (H-UDI)	—
	PA9 输入 / 输出 (端口)	TCLKD 输入 (MTU2)	TXD2 输出 (SCI)	$\overline{\text{POE8}}$ 输入 (POE)	TDO 输出 (H-UDI)
	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)	—	—	—
	PA13 输入 / 输出 (端口)	SCK1 输入 / 输出 (SCI)	—	—	—
	PA14 输入 / 输出 (端口)	RXD1 输入 (SCI)	—	—	—
	PA15 输入 / 输出 (端口)	TXD1 输出 (SCI)	—	—	—
B	PB1 输入 / 输出 (端口)	TIC5W 输入 (MTU2)	—	—	—
	PB2 输入 / 输出 (端口)	IRQ0 输入 (INTC)	$\overline{\text{POE0}}$ 输入 (POE)	—	—
	PB3 输入 / 输出 (端口)	IRQ1 输入 (INTC)	$\overline{\text{POE1}}$ 输入 (POE)	TIC5V 输入 (MTU2)	—
	PB5 输入 / 输出 (端口)	IRQ3 输入 (INTC)	TIC5U 输入 (MTU2)	—	—
	PB16 输入 / 输出 (端口)	$\overline{\text{POE3}}$ 输入 (POE)	—	—	—
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)	—	—	—

端口	功能 1 (关联模块)	功能 2 (关联模块)	功能 3 (关联模块)	功能 4 (关联模块)	功能 5 (关联模块)
E	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)	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)	IRQOUT 输出 (INTC)	—	—
F	PF0 输入 (端口)	AN0 输入 (A/D)	—	—	—
	PF1 输入 (端口)	AN1 输入 (A/D)	—	—	—
	PF2 输入 (端口)	AN2 输入 (A/D)	—	—	—
	PF3 输入 (端口)	AN3 输入 (A/D)	—	—	—
	PF4 输入 (端口)	AN4 输入 (A/D)	—	—	—
	PF5 输入 (端口)	AN5 输入 (A/D)	—	—	—
	PF6 输入 (端口)	AN6 输入 (A/D)	—	—	—
	PF7 输入 (端口)	AN7 输入 (A/D)	—	—	—

【注】 在 A/D 转换中 AN 输入功能有效。

表 15.2 多路复用一览表 (SH7124)

端口	功能 1 (关联模块)	功能 2 (关联模块)	功能 3 (关联模块)	功能 4 (关联模块)	功能 5 (关联模块)
A	PA0 输入 / 输出 (端口)	POE0 输入 (POE)	RXD0 输入 (SCI)	—	—
	PA1 输入 / 输出 (端口)	POE1 输入 (POE)	TXD0 输出 (SCI)	—	—
	PA3 输入 / 输出 (端口)	IRQ1 输入 (INTC)	RXD1 输入 (SCI)	TRST 输入 (H-UDI)	—
	PA4 输入 / 输出 (端口)	IRQ2 输入 (INTC)	TXD1 输出 (SCI)	TMS 输入 (H-UDI)	—
	PA6 输入 / 输出 (端口)	TCLKA 输入 (MTU2)	—	—	—
	PA7 输入 / 输出 (端口)	TCLKB 输入 (MTU2)	SCK2 输入 / 输出 (SCI)	TCK 输入 (H-UDI)	—
	PA8 输入 / 输出 (端口)	TCLKC 输入 (MTU2)	RXD2 输入 (SCI)	TDI 输入 (H-UDI)	—
	PA9 输入 / 输出 (端口)	TCLKD 输入 (MTU2)	TXD2 输出 (SCI)	POE8 输入 (POE)	TDO 输出 (H-UDI)
B	PB1 输入 / 输出 (端口)	TIC5W 输入 (MTU2)	—	—	—
	PB3 输入 / 输出 (端口)	IRQ1 输入 (INTC)	POE1 输入 (POE)	TIC5V 输入 (MTU2)	—
	PB5 输入 / 输出 (端口)	IRQ3 输入 (INTC)	TIC5U 输入 (MTU2)	—	—
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)	—	—
	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)	MRES 输入 (INTC)	—	—

端口	功能 1 (关联模块)	功能 2 (关联模块)	功能 3 (关联模块)	功能 4 (关联模块)	功能 5 (关联模块)
E	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)	—	—
F	PF0 输入 (端口)	AN0 输入 (A/D)	—	—	—
	PF1 输入 (端口)	AN1 输入 (A/D)	—	—	—
	PF2 输入 (端口)	AN2 输入 (A/D)	—	—	—
	PF3 输入 (端口)	AN3 输入 (A/D)	—	—	—
	PF4 输入 (端口)	AN4 输入 (A/D)	—	—	—
	PF5 输入 (端口)	AN5 输入 (A/D)	—	—	—
	PF6 输入 (端口)	AN6 输入 (A/D)	—	—	—
	PF7 输入 (端口)	AN7 输入 (A/D)	—	—	—

【注】 在 A/D 转换中 AN 输入功能有效。

表 15.3 按运行模式分类的管脚功能一览表 (SH7125)

管脚号	管脚名	
	单芯片模式 (MCU 模式 3)	
	初始功能	PFC 可设定的功能
4,22,35	Vcc	Vcc
6,24,33	Vss	Vss
8,37	VCL	VCL
61	AVcc	AVcc
52	AVss	AVss
47	PLLVss	PLLVss
42	EXTAL	EXTAL
41	XTAL	XTAL
46	MD1	MD1
45	FWE/(ASEBRKAK/ASEBRK*)	FWE
39	$\overline{\text{RES}}$	$\overline{\text{RES}}$
40	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$
44	NMI	NMI
43	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$
38	PA0	PA0/ $\overline{\text{POE0}}$ /RXD0
36	PA1	PA1/ $\overline{\text{POE1}}$ /TXD0
34	PA2	PA2/IRQ0/SCK0
32	PA3/(TRST*)	PA3/IRQ1/RXD1
31	PA4/(TMS*)	PA4/IRQ2/TXD1
30	PA5	PA5/IRQ3/SCK1
29	PA6	PA6/TCLKA
28	PA7/(TCK*)	PA7/TCLKB/SCK2
27	PA8/(TDI*)	PA8/TCLKC/RXD2
26	PA9/(TDO*)	PA9/TCLKD/TXD2/ $\overline{\text{POE8}}$
25	PA10	PA10/RXD0
23	PA11	PA11/TXD0/ $\overline{\text{ADTRG}}$
21	PA12	PA12/SCK0
20	PA13	PA13/SCK1
19	PA14	PA14/RXD1
18	PA15	PA15/TXD1
51	PB1	PB1/TIC5W
50	PB2	PB2/IRQ0/ $\overline{\text{POE0}}$
49	PB3	PB3/IRQ1/ $\overline{\text{POE1}}$ /TIC5V
48	PB5	PB5/IRQ3/TIC5U
62	$\overline{\text{POE3}}$	PB16/ $\overline{\text{POE3}}$
17	PE0	PE0/TIOC0A
16	PE1	PE1/TIOC0B/RXD0

管脚号	管脚名	
	单芯片模式 (MCU 模式 3)	
	初始功能	PFC 可设定的功能
15	PE2	PE2/TIOC0C/TXD0
14	PE3	PE3/TIOC0D/SCK0
13	PE4	PE4/TIOC1A/RXD1
12	PE5	PE5/TIOC1B/TXD1
11	PE6	PE6/TIOC2A/SCK1
10	PE7	PE7/TIOC2B
9	PE8	PE8/TIOC3A
5	PE9	PE9/TIOC3B
7	PE10	PE10/TIOC3C
3	PE11	PE11/TIOC3D
2	PE12	PE12/TIOC4A
1	PE13	PE13/TIOC4B/MRES
64	PE14	PE14/TIOC4C
63	PE15	PE15/TIOC4D/IRQOUT
60	PF0/AN0	PF0/AN0
59	PF1/AN1	PF1/AN1
58	PF2/AN2	PF2/AN2
57	PF3/AN3	PF3/AN3
56	PF4/AN4	PF4/AN4
55	PF5/AN5	PF5/AN5
54	PF6/AN6	PF6/AN6
53	PF7/AN7	PF7/AN7

【注】 * 当使用 E10A 时 (当 $\overline{\text{ASEMD0}}$ =Low 电平时), 这些管脚被固定为 TMS、 $\overline{\text{TRST}}$ 、TDI、TDO、TCK、 $\overline{\text{ASEBRKAK/ASEBRK}}$ 。

表 15.4 按运行模式分类的管脚功能一览表 (SH7124)

管脚号	管脚名	
	单芯片模式 (MCU 模式 3)	
	初始功能	PFC 可设定的功能
4,17	Vcc	Vcc
6,19	Vss	Vss
8,25	VCL	VCL
48	AVcc	AVcc
39	AVss	AVss
35	PLLvss	PLLvss
30	EXTAL	EXTAL
29	XTAL	XTAL
34	MD1	MD1
33	FWE/(ASEBRKAK/ASEBRK*)	FWE
27	$\overline{\text{RES}}$	$\overline{\text{RES}}$
28	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$
32	NMI	NMI
31	$\overline{\text{ASEMD0}}$	$\overline{\text{ASEMD0}}$
26	PA0	PA0/POE0/RXD0
24	PA1	PA1/POE1/TXD0
23	PA3/(TRST*)	PA3/IRQ1/RXD1
22	PA4/(TMS*)	PA4/IRQ2/TXD1
21	PA6	PA6/TCLKA
20	PA7/(TCK*)	PA7/TCLKB/SCK2
18	PA8/(TDI*)	PA8/TCLKC/RXD2
16	PA9/(TDO*)	PA9/TCLKD/TXD2/ $\overline{\text{POE8}}$
38	PB1	PB1/TIC5W
37	PB3	PB3/IRQ1/POE1/TIC5V
36	PB5	PB5/IRQ3/TIC5U
15	PE0	PE0/TIOC0A
14	PE1	PE1/TIOC0B/RXD0
13	PE2	PE2/TIOC0C/TXD0
12	PE3	PE3/TIOC0D/SCK0
11	PE8	PE8/TIOC3A
9	PE9	PE9/TIOC3B
10	PE10	PE10/TIOC3C
7	PE11	PE11/TIOC3D
5	PE12	PE12/TIOC4A
3	PE13	PE13/TIOC4B/ $\overline{\text{MRES}}$
2	PE14	PE14/TIOC4C
1	PE15	PE15/TIOC4D/IRQOUT

管脚号	管脚名	
	单芯片模式（MCU 模式 3）	
	初始功能	PFC 可设定的功能
47	PF0/AN0	PF0/AN0
46	PF1/AN1	PF1/AN1
45	PF2/AN2	PF2/AN2
44	PF3/AN3	PF3/AN3
43	PF4/AN4	PF4/AN4
42	PF5/AN5	PF5/AN5
41	PF6/AN6	PF6/AN6
40	PF7/AN7	PF7/AN7

【注】 * 当使用 E10A 使用时（当 ASEMD0=Low 电平时），这些管脚被固定为 TMS、TRST、TDI、TDO、TCK、ASEBRKAK/ASEBRK。

15.1 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中的寄存器状态，请参照“第 20 章 寄存器一览”。

表 15.5 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
端口 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 H	PBIORH	R/W	H'0000	H'FFFFD184	8、16、32
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 H1	PBCRH1	R/W	H'0000	H'FFFFD18E	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 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 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 功能控制寄存器	IFCR	R/W	H'0000	H'FFFFD322	8、16

15.1.1 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 的管脚输入 / 输出方向。PA15IOR ~ PA0IOR 位分别对应 PA15 ~ PA0 管脚（省略管脚名中端口以外的多路复用管脚名）。PAIOR 在端口 A 的管脚功能为通用输入 / 输出（PA15 ~ PA0）时有效，否则无效。

如果将 PAIORL 的位设定为 1，对应的管脚就为输出管脚；如果置 0 就为输入管脚。但是，在 SH7124 中 PAIORL 的 bit15 ~ 10、5、2 无效。

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

15.1.2 端口 A 的控制寄存器 L1 ~ L4 (PACRL1 ~ PACRL4)

PACRL1 ~ PACRL4 分别是 16 位可读写寄存器，选择端口 A 的多路复用管脚功能。

(1) SH7125

- 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 读写的值总是 0。
14	PA15MD2	0	R/W	PA15 模式位 选择 PA15/TXD1 管脚的功能。 000: PA15 输入 / 输出 (端口) 110: TXD1 输出 (SCI) 其他: 禁止设定
13	PA15MD1	0	R/W	
12	PA15MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PA14MD2	0	R/W	PA14 模式位 选择 PA14/RXD1 管脚的功能。 000: PA14 输入 / 输出 (端口) 110: RXD1 输入 (SCI) 其他: 禁止设定
9	PA14MD1	0	R/W	
8	PA14MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PA13MD2	0	R/W	PA13 模式位 选择 PA13/SCK1 管脚的功能。 000: PA13 输入 / 输出 (端口) 110: SCK1 输入 / 输出 (SCI) 其他: 禁止设定
5	PA13MD1	0	R/W	
4	PA13MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PA12MD2	0	R/W	PA12 模式位 选择 PA12/SCK0 管脚的功能。 000: PA12 输入 / 输出 (端口) 110: SCK0 输入 / 输出 (SCI) 其他: 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 读写的值总是 0。
14	PA11MD2	0	R/W	PA11 模式位 选择 11/TXD0/ADTRG 管脚的功能。 000: PA11 输入 / 输出 (端口) 010: ADTRG 输入 (A/D) 110: TXD0 输出 (SCI) 其他: 禁止设定
13	PA11MD1	0	R/W	
12	PA11MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PA10MD2	0	R/W	PA10 模式位 选择 PA10/RXD0 管脚的功能。 000: PA10 输入 / 输出 (端口) 110: RXD0 输入 (SCI) 其他: 禁止设定
9	PA10MD1	0	R/W	
8	PA10MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PA9MD2	0	R/W	PA9 模式位 选择 PA9/TCLKD/TXD2/TDO/POE8 管脚的功能。当使用 E10A 时 (当 ASEMD0=Low 电平时), 此管脚被固定为 TDO 输出。 000: PA9 输入 / 输出 (端口) 001: TCLKD 输入 (MTU2) 110: TXD2 输出 (SCI) 111: POE8 输入 (POE) 其他: 禁止设定
5	PA9MD1	0	R/W	
4	PA9MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PA8MD2	0	R/W	PA8 模式位 选择 PA8/TCLKC/RXD2/TDI 管脚的功能。当使用 E10A 时 (当 ASEMD0=Low 电平时), 此管脚被固定为 TDI 输入。 000: PA8 输入 / 输出 (端口) 001: TCLKC 输入 (MTU2) 110: RXD2 输入 (SCI) 其他: 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 读写的值总是 0。
14	PA7MD2	0	R/W	PA7 模式位 选择 PA7/TCLKB/SCK2/TCK 管脚的功能。当使用 E10A 时 (当 ASEMD0=Low 电平时), 此管脚被固定为 TCK 输入。 000: PA7 输入 / 输出 (端口) 001: TCLKB 输入 (MTU2) 110: SCK2 输入 / 输出 (SCI) 其他: 禁止设定
13	PA7MD1	0	R/W	
12	PA7MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PA6MD2	0	R/W	PA6 模式位 选择 PA6/TCLKA 管脚的功能。 000: PA6 输入 / 输出 (端口) 001: TCLKA 输入 (MTU2) 其他: 禁止设定
9	PA6MD1	0	R/W	
8	PA6MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PA5MD2	0	R/W	PA5 模式位 选择 PA5/IRQ3/SCK1 管脚的功能。 000: PA5 输入 / 输出 (端口) 001: SCK1 输入 / 输出 (SCI) 111: IRQ3 输入 (INTC) 其他: 禁止设定
5	PA5MD1	0	R/W	
4	PA5MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PA4MD2	0	R/W	PA4 模式位 选择 PA4/IRQ2/TXD1/TMS 管脚的功能。当使用 E10A 时 (当 ASEMD0=Low 电平时), 此管脚被固定为 TMS 输入。 000: PA4 输入 / 输出 (端口) 001: TXD1 输出 (SCI) 111: IRQ2 输入 (INTC) 其他: 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 读写的值总是 0。
14	PA3MD2	0	R/W	PA3 模式位 选择 PA3/ $\overline{\text{IRQ1}}$ / RXD1 / $\overline{\text{TRST}}$ 管脚的功能。当使用 E10A 时 (当 ASEMD0=Low 电平时), 此管脚被固定为 $\overline{\text{TRST}}$ 输入。 000: PA3 输入 / 输出 (端口) 001: RXD1 输入 (SCI) 111: IRQ1 输入 (INTC) 其他: 禁止设定
13	PA3MD1	0	R/W	
12	PA3MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PA2MD2	0	R/W	PA2 模式位 选择 PA2/ IRQ0 / SCK0 管脚的功能。 000: PA2 输入 / 输出 (端口) 001: SCK0 输入 / 输出 (SCI) 011: IRQ0 输入 (INTC) 其他: 禁止设定
9	PA2MD1	0	R/W	
8	PA2MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PA1MD2	0	R/W	PA1 模式位 选择 PA1/ $\overline{\text{POE1}}$ / TXD0 管脚的功能。 000: PA1 输入 / 输出 (端口) 001: TXD0 输出 (SCI) 111: $\overline{\text{POE1}}$ 输入 (POE) 其他: 禁止设定
5	PA1MD1	0	R/W	
4	PA1MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PA0MD2	0	R/W	PA0 模式位 选择 PA0/ $\overline{\text{POE0}}$ / RXD0 管脚的功能。 000: PA0 输入 / 输出 (端口) 001: RXD0 输入 (SCI) 111: $\overline{\text{POE0}}$ 输入 (POE) 其他: 禁止设定
1	PA0MD1	0	R/W	
0	PA0MD0	0	R/W	

(2) SH7124

• 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
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 0	—	全 0	R	保留位 读写的值总是 0。

• 端口 A 的控制寄存器 L3 (PACRL3)

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

位	位名	初始值	R/W	说 明
15 ~ 7	—	全 0	R	保留位 读写的值总是 0。
6 5 4	PA9MD2 PA9MD1 PA9MD0	0 0 0	R/W R/W R/W	PA9 模式位 选择 PA9/TCLKD/TXD2/TDO/ $\overline{\text{POE8}}$ 管脚的功能。当使用 E10A 时 (当 $\overline{\text{ASEMD0}}$ =Low 电平时), 此管脚被固定为 TDO 输出。 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) 其他: 禁止设定
3	—	0	R	保留位 读写的值总是 0。
2 1 0	PA8MD2 PA8MD1 PA8MD0	0 0 0	R/W R/W R/W	PA8 模式位 选择 PA8/TCLKC/RXD2/TDI 管脚的功能。当使用 E10A 时 (当 $\overline{\text{ASEMD0}}$ =Low 电平时), 此管脚被固定为 TDI 输入。 000: PA8 输入 / 输出 (端口) 001: TCLKC 输入 (MTU2) 110: RXD2 输入 (SCI) 其他: 禁止设定

• 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	-	-	-	-	-	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	R	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PA7MD2	0	R/W	PA7 模式位 选择 PA7/TCLKB/SCK2/TCK 管脚的功能。当使用 E10A 时 (当 ASEMD0=Low 电平时), 此管脚被固定为 TCK 输入。 000: PA7 输入 / 输出 (端口) 001: TCLKB 输入 (MTU2) 110: SCK2 输入 / 输出 (SCI) 其他: 禁止设定
13	PA7MD1	0	R/W	
12	PA7MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PA6MD2	0	R/W	PA6 模式位 选择 PA6/TCLKA 管脚的功能。 000: PA6 输入 / 输出 (端口) 001: TCLKA 输入 (MTU2) 其他: 禁止设定
9	PA6MD1	0	R/W	
8	PA6MD0	0	R/W	
7 ~ 3	—	全 0	R	保留位 读写的值总是 0。
2	PA4MD2	0	R/W	PA4 模式位 选择 PA4/IRQ2/TXD1/TMS 管脚的功能。当使用 E10A 时 (当 ASEMD0=Low 电平时), 此管脚被固定为 TMS 输入。 000: PA4 输入 / 输出 (端口) 001: TXD1 输出 (SCI) 111: IRQ2 输入 (INTC) 其他: 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	-	-	-	-	-	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	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PA3MD2	0	R/W	PA3 模式位 选择 PA3/ $\overline{\text{IRQ1}}$ /RXD1/ $\overline{\text{TRST}}$ 管脚的功能。当使用 E10A 时 (当 ASEMD0=Low 电平时), 此管脚被固定为 $\overline{\text{TRST}}$ 输入。 000: PA3 输入 / 输出 (端口) 001: RXD1 输入 (SCI) 111: IRQ1 输入 (INTC) 其他: 禁止设定
13	PA3MD1	0	R/W	
12	PA3MD0	0	R/W	
11 ~ 7	—	全 0	R	保留位 读写的值总是 0。
6	PA1MD2	0	R/W	PA1 模式位 选择 PA1/ $\overline{\text{POE1}}$ /TXD0 管脚的功能。 000: PA1 输入 / 输出 (端口) 001: TXD0 输出 (SCI) 111: $\overline{\text{POE1}}$ 输入 (POE) 其他: 禁止设定
5	PA1MD1	0	R/W	
4	PA1MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PA0MD2	0	R/W	PA0 模式位 选择 PA0/ $\overline{\text{POE0}}$ /RXD0 管脚的功能。 000: PA0 输入 / 输出 (端口) 001: RXD0 输入 (SCI) 111: $\overline{\text{POE0}}$ 输入 (POE) 其他: 禁止设定
1	PA0MD1	0	R/W	
0	PA0MD0	0	R/W	

15.1.3 端口 B 的 IO 寄存器 L、H (PBIORL、PBIORH)

PBIORL 和 PBIORH 分别是 16 位可读写寄存器，选择端口 B 的管脚输入 / 输出方向。PB16IOR、PB5IOR、PB3IOR ~ PB1IOR 位分别对应 PB16、PB5、PB3 ~ PB1 管脚（省略管脚名中端口以外的多路复用管脚名）。PBIORL 在端口 B 的管脚功能为通用输入 / 输出（PB5、PB3 ~ PB1）和 SCI 的 SCK 输入 / 输出时有效，否则无效；PBIORH 在端口 B 的管脚功能为通用输入 / 输出（PB16）时有效，否则无效。

如果将 PBIORL 和 PBIORH 位设定为 1，对应的管脚就为输出管脚；如果置 0 就为输入管脚。

但是，在 SH7124 中 PBIORL 的 bit2 和 PBIORH 的 bit0 无效。

PBIORL 的 bit15 ~ 6、4、0 和 PBIORH 的 bit15 ~ 1 为保留位，读写的值总是 0。

PBIORL 和 PBIORH 的初始值都为 H'0000。

(1) 端口 B 的 IO 寄存器 H (PBIORH)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PB16 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	R	R	R	R	R/W

(2) 端口 B 的 IO 寄存器 L (PBIORL)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PB5 IOR	-	PB3 IOR	PB2 IOR	PB1 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	R/W	R/W	R/W	R

15.1.4 端口 B 的控制寄存器 L1、L2、H1 (PBCRL1、PBCRL2、PBCRH1)

PBCRL1、PBCRL2、PBCRH1 分别是 16 位可读写寄存器，选择端口 B 的多路复用管脚功能。

(1) SH7125

- 端口 B 的控制寄存器 H1 (PBCRH1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PB16 MD
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W*

【注】* 只能在加电复位后写1次。

位	位名	初始值	R/W	说 明
15 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	PB16MD	1	R/W*	PB16 模式位 选择 PB16/POE3 管脚的功能。 0: PB16 输入 / 输出 (端口) 1: POE3 输入 (POE)

- 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
	-	-	-	-	-	-	-	-	-	PB5 MD2	PB5 MD1	PB5 MD0	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 7	—	全 0	R	保留位 读写的值总是 0。
6	PB5MD2	0	R/W	PB5 模式位 选择 PB5/IRQ3/TIC5U 管脚的功能。 000: PB5 输入 / 输出 (端口) 001: IRQ3 输入 (INTC) 011: TIC5U 输入 (MTU2) 其他: 禁止设定
5	PB5MD1	0	R/W	
4	PB5MD0	0	R/W	
3 ~ 0	—	全 0	R	保留位 读写的值总是 0。

• 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	-	-	-	-
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	R

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PB3MD2	0	R/W	PB3 模式位 选择 PB3/IRQ1/ $\overline{\text{POE1}}$ /TIC5V 管脚的功能。 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) 其他: 禁止设定
13	PB3MD1	0	R/W	
12	PB3MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PB2MD2	0	R/W	PB2 模式位 选择 PB2/IRQ0/ $\overline{\text{POE0}}$ 管脚的功能。 000: PB2 输入 / 输出 (端口) 001: IRQ0 输入 (INTC) 010: $\overline{\text{POE0}}$ 输入 (POE) 其他: 禁止设定
9	PB2MD1	0	R/W	
8	PB2MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PB1MD2	0	R/W	PB1 模式位 选择 PB1/TIC5W 管脚的功能。 000: PB1 输入 / 输出 (端口) 011: TIC5W 输入 (MTU2) 其他: 禁止设定
5	PB1MD1	0	R/W	
4	PB1MD0	0	R/W	
3 ~ 0	—	全 0	R	保留位 读写的值总是 0。

(2) SH7124

• 端口 B 的控制寄存器 H1 (PBCRH1)

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

位	位名	初始值	R/W	说 明
15 ~ 0	—	全 0	R	保留位 读写的值总是 0。

• 端口 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
	-	-	-	-	-	-	-	-	-	PB5 MD2	PB5 MD1	PB5 MD0	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 7	—	全 0	R	保留位 读写的值总是 0。
6	PB5MD2	0	R/W	PB5 模式位
5	PB5MD1	0	R/W	选择 PB5/IRQ3/TIC5U 管脚的功能。
4	PB5MD0	0	R/W	000: PB5 输入 / 输出 (端口) 001: IRQ3 输入 (INTC) 011: TIC5U 输入 (MTU2) 其他: 禁止设定
3 ~ 0	—	全 0	R	保留位 读写的值总是 0。

• 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	-	-	-	-	-	PB1 MD2	PB1 MD1	PB1 MD0	-	-	-	-
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PB3MD2	0	R/W	PB3 模式位 选择 PB3/IRQ1/POE1/TIC5V 管脚的功能。 000: PB3 输入 / 输出 (端口) 001: IRQ1 输入 (INTC) 010: POE1 输入 (POE) 011: TIC5V 输入 (MTU2) 其他: 禁止设定
13	PB3MD1	0	R/W	
12	PB3MD0	0	R/W	
11 ~ 7	—	全 0	R	保留位 读写的值总是 0。
6	PB1MD2	0	R/W	PB1 模式位 选择 PB1/TIC5W 管脚的功能。 000: PB1 输入 / 输出 (端口) 011: TIC5W 输入 (MTU2) 其他: 禁止设定
5	PB1MD1	0	R/W	
4	PB1MD0	0	R/W	
3 ~ 0	—	全 0	R	保留位 读写的值总是 0。

15.1.5 端口 E 的 IO 寄存器 L (PEIORL)

PEIORL 是 16 位可读写寄存器, 选择端口 E 的管脚输入 / 输出方向。PE15IOR ~ PE0IOR 位分别对应 PE15 管脚 ~ PE0 管脚 (省略管脚名中端口以外的多路复用管脚名)。PEIORL 在端口 E 的管脚功能为通用输入 / 输出 (PE15 ~ PE0) 和 MTU2 的 TIOC 输入 / 输出时有效, 否则无效。

如果将 PEIORL 的位设定为 1, 对应的管脚就为输出管脚; 如果置 0 就为输入管脚。

但是, 在 SH7124 中 PEIORL 的 bit7 ~ 4 无效。

PEIORL 的初始值为 H'0000。

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

15.1.6 端口 E 的控制寄存器 L1 ~ L4 (PECRL1 ~ PECRL4)

PECRL1 ~ PECRL4 分别是 16 位可读写寄存器，选择端口 E 的多路复用管脚功能。

(1) SH7125

- 端口 E 的控制寄存器 L4 (PECRL4)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE15 MD2	PE15 MD1	PE15 MD0	-	PE14 MD2	PE14 MD1	PE14 MD0	-	-	PE13 MD1	PE13 MD0	-	PE12 MD2	PE12 MD1	PE12 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PE15MD2	0	R/W	PE15 模式位 选择 PE15/TIOC4D/ $\overline{\text{IRQOUT}}$ 管脚的功能。 000: PE15 输入 / 输出 (端口) 001: TIOC4D 输入 / 输出 (MTU2) 011: $\overline{\text{IRQOUT}}$ 输出 (INTC) 其他: 禁止设定
13	PE15MD1	0	R/W	
12	PE15MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PE14MD2	0	R/W	PE14 模式位 选择 PE14/TIOC4C 管脚的功能。 000: PE14 输入 / 输出 (端口) 001: TIOC4C 输入 / 输出 (MTU2) 其他: 禁止设定
9	PE14MD1	0	R/W	
8	PE14MD0	0	R/W	
7、6	—	全 0	R	保留位 读写的值总是 0。
5	PE13MD1	0	R/W	PE13 模式位 选择 PE13/TIOC4B/ $\overline{\text{MRES}}$ 管脚的功能。 00: PE13 输入 / 输出 (端口) 01: TIOC4B 输入 / 输出 (MTU2) 10: $\overline{\text{MRES}}$ 输入 (INTC) 其他: 禁止设定
4	PE13MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PE12MD2	0	R/W	PE12 模式位 选择 PE12/TIOC4A 管脚的功能。 000: PE12 输入 / 输出 (端口) 001: TIOC4A 输入 / 输出 (MTU2) 其他: 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 读写的值总是 0。
14	PE11MD2	0	R/W	PE11 模式位 选择 PE11/TIOC3D 管脚的功能。 000: PE11 输入 / 输出 (端口) 001: TIOC3D 输入 / 输出 (MTU2) 其他: 禁止设定
13	PE11MD1	0	R/W	
12	PE11MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PE10MD2	0	R/W	PE10 模式位 选择 PE10/TIOC3C 管脚的功能。 000: PE10 输入 / 输出 (端口) 001: TIOC3C 输入 / 输出 (MTU2) 其他: 禁止设定
9	PE10MD1	0	R/W	
8	PE10MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PE9MD2	0	R/W	PE9 模式位 选择 PE9/TIOC3B 管脚的功能。 000: PE9 输入 / 输出 (端口) 001: TIOC3B 输入 / 输出 (MTU2) 其他: 禁止设定
5	PE9MD1	0	R/W	
4	PE9MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PE8MD2	0	R/W	PE8 模式位 选择 PE8/TIOC3A 管脚的功能。 000: PE8 输入 / 输出 (端口) 001: TIOC3A 输入 / 输出 (MTU2) 其他: 禁止设定
1	PE8MD1	0	R/W	
0	PE8MD0	0	R/W	

• 端口 E 的控制寄存器 L2 (PECRL2)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE7 MD2	PE7 MD1	PE7 MD0	-	PE6 MD2	PE6 MD1	PE6 MD0	-	PE5 MD2	PE5 MD1	PE5 MD0	-	PE4 MD2	PE4 MD1	PE4 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PE7MD2	0	R/W	PE7 模式位 选择 PE7/TIOC2B 管脚的功能。 000: PE7 输入 / 输出 (端口) 001: TIOC2B 输入 / 输出 (MTU2) 其他: 禁止设定
13	PE7MD1	0	R/W	
12	PE7MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PE6MD2	0	R/W	PE6 模式位 选择 PE6/TIOC2A/SCK1 管脚的功能。 000: PE6 输入 / 输出 (端口) 001: TIOC2A 输入 / 输出 (MTU2) 110: SCK1 输入 / 输出 (SCI) 其他: 禁止设定
9	PE6MD1	0	R/W	
8	PE6MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PE5MD2	0	R/W	PE5 模式位 选择 PE5/TIOC1B/TXD1 管脚的功能。 000: PE5 输入 / 输出 (端口) 001: TIOC1B 输入 / 输出 (MTU2) 110: TXD1 输出 (SCI) 其他: 禁止设定
5	PE5MD1	0	R/W	
4	PE5MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PE4MD2	0	R/W	PE4 模式位 选择 PE4/TIOC1A/RXD1 管脚的功能。 000: PE4 输入 / 输出 (端口) 001: TIOC1A 输入 / 输出 (MTU2) 110: RXD1 输入 (SCI) 其他: 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 读写的值总是 0。
14	PE3MD2	0	R/W	PE3 模式位 选择 PE3/TIOC0D/SCK0 管脚的功能。 000: PE3 输入 / 输出 (端口) 001: TIOC0D 输入 / 输出 (MTU2) 110: SCK0 输入 / 输出 (SCI) 其他: 禁止设定
13	PE3MD1	0	R/W	
12	PE3MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PE2MD2	0	R/W	PE2 模式位 选择 PE2/TIOC0C/TXD0 管脚的功能 000: PE2 输入 / 输出 (端口) 001: TIOC0C 输入 / 输出 (MTU2) 110: TXD0 输出 (SCI) 其他: 禁止设定
9	PE2MD1	0	R/W	
8	PE2MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PE1MD2	0	R/W	PE1 模式位 选择 PE1/TIOC0B/RXD0 管脚的功能。 000: PE1 输入 / 输出 (端口) 001: TIOC0B 输入 / 输出 (MTU2) 110: RXD0 输入 (SCI) 其他: 禁止设定
5	PE1MD1	0	R/W	
4	PE1MD0	0	R/W	
3、2	—	全 0	R	保留位 读写的值总是 0。
1	PE0MD1	0	R/W	PE0 模式位 选择 PE0/TIOC0A 管脚的功能。 00: PE0 输入 / 输出 (端口) 01: TIOC0A 输入 / 输出 (MTU2) 其他: 禁止设定
0	PE0MD0	0	R/W	

(2) SH7124

• 端口 E 的控制寄存器 L4 (PECRL4)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE15 MD2	PE15 MD1	PE15 MD0	-	PE14 MD2	PE14 MD1	PE14 MD0	-	-	PE13 MD1	PE13 MD0	-	PE12 MD2	PE12 MD1	PE12 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PE15MD2	0	R/W	PE15 模式位 选择 PE15/TIOC4D/ $\overline{\text{IRQOUT}}$ 管脚的功能。 000: PE15 输入 / 输出 (端口) 001: TIOC4D 输入 / 输出 (MTU2) 011: $\overline{\text{IRQOUT}}$ 输出 (INTC) 其他: 禁止设定
13	PE15MD1	0	R/W	
12	PE15MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PE14MD2	0	R/W	PE14 模式位 选择 PE14/TIOC4C 管脚的功能。 000: PE14 输入 / 输出 (端口) 001: TIOC4C 输入 / 输出 (MTU2) 其他: 禁止设定
9	PE14MD1	0	R/W	
8	PE14MD0	0	R/W	
7、6	—	全 0	R	保留位 读写的值总是 0。
5	PE13MD1	0	R/W	PE13 模式位 选择 PE13/TIOC4B/ $\overline{\text{MRES}}$ 管脚的功能。 00: PE13 输入 / 输出 (端口) 01: TIOC4B 输入 / 输出 (MTU2) 10: $\overline{\text{MRES}}$ 输入 (INTC) 其他: 禁止设定
4	PE13MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PE12MD2	0	R/W	PE12 模式位 选择 PE12/TIOC4A 管脚的功能。 000: PE12 输入 / 输出 (端口) 001: TIOC4A 输入 / 输出 (MTU2) 其他: 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 读写的值总是 0。
14	PE11MD2	0	R/W	PE11 模式位 选择 PE11/TIOC3D 管脚的功能。 000: PE11 输入 / 输出 (端口) 001: TIOC3D 输入 / 输出 (MTU2) 其他: 禁止设定
13	PE11MD1	0	R/W	
12	PE11MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PE10MD2	0	R/W	PE10 模式位 选择 PE10/TIOC3C 管脚的功能。 000: PE10 输入 / 输出 (端口) 001: TIOC3C 输入 / 输出 (MTU2) 其他: 禁止设定
9	PE10MD1	0	R/W	
8	PE10MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PE9MD2	0	R/W	PE9 模式位 选择 PE9/TIOC3B 管脚的功能。 000: PE9 输入 / 输出 (端口) 001: TIOC3B 输入 / 输出 (MTU2) 其他: 禁止设定
5	PE9MD1	0	R/W	
4	PE9MD0	0	R/W	
3	—	0	R	保留位 读写的值总是 0。
2	PE8MD2	0	R/W	PE8 模式位 选择 PE8/TIOC3A 管脚的功能。 000: PE8 输入 / 输出 (端口) 001: TIOC3A 输入 / 输出 (MTU2) 其他: 禁止设定
1	PE8MD1	0	R/W	
0	PE8MD0	0	R/W	

• 端口 E 的控制寄存器 L2 (PECRL2)

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

位	位名	初始值	R/W	说 明
15 ~ 0	—	全 0	R	保留位 读写的值总是 0。

• 端口 E 的控制寄存器 L1 (PECRL1)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	PE3 MD2	PE3 MD1	PE3 MD0	-	PE2 MD2	PE2 MD1	PE2 MD0	-	PE1 MD2	PE1 MD1	PE1 MD0	-	-	PE0 MD1	PE0 MD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
15	—	0	R	保留位 读写的值总是 0。
14	PE3MD2	0	R/W	PE3 模式位 选择 PE3/TIOC0D/SCK0 管脚的功能。 000: PE3 输入 / 输出 (端口) 001: TIOC0D 输入 / 输出 (MTU2) 110: SCK0 输入 / 输出 (SCI) 其他: 禁止设定
13	PE3MD1	0	R/W	
12	PE3MD0	0	R/W	
11	—	0	R	保留位 读写的值总是 0。
10	PE2MD2	0	R/W	PE2 模式位 选择 PE2/TIOC0C/TXD0 管脚的功能。 000: PE2 输入 / 输出 (端口) 001: TIOC0C 输入 / 输出 (MTU2) 110: TXD0 输出 (SCI) 其他: 禁止设定
9	PE2MD1	0	R/W	
8	PE2MD0	0	R/W	
7	—	0	R	保留位 读写的值总是 0。
6	PE1MD2	0	R/W	PE1 模式位 选择 PE1/TIOC0B/RXD0 管脚的功能。 000: PE1 输入 / 输出 (端口) 001: TIOC0B 输入 / 输出 (MTU2) 110: RXD0 输入 (SCI) 其他: 禁止设定
5	PE1MD1	0	R/W	
4	PE1MD0	0	R/W	
3、2	—	全 0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说 明
1	PE0MD1	0	R/W	PE0 模式位 选择 PE0/TIOC0A 管脚的功能。 00: PE0 输入 / 输出 (端口) 01: TIOC0A 输入 / 输出 (MTU2) 其他: 禁止设定
0	PE0MD0	0	R/W	

15.1.7 IRQOUT 功能控制寄存器 (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 管脚的输出。在 PECRL4 设定为其他功能时, 不影响管脚功能。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	IRQMD1	IRQMD0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
15 ~ 2	—	全 0	R	保留位 读写的值总是 0。
1	IRQMD1	0	R/W	端口 E 的 IRQOUT 管脚功能的选择 在 PECRL4 的 bit14、13、12 (PE15MD2、PE15MD1、PE15MD0) 被置为 (0、1、1) 时, 选择 IRQOUT 管脚功能。 00: 输出中断请求的接受信号 其他: 总是输出高电平
0	IRQMD0	0	R/W	

15.2 使用注意事项

- 1. 在本 LSI 中，作为多路复用功能，将同一功能分配到多个管脚。这是为了在提高管脚功能的选择自由度的同时方便电路板的设计，但是在 2 个以上的管脚使用 1 个功能时，必须注意以下几点：
 - 管脚功能为输入功能的情况
通过 OR 或者 AND 逻辑将多个管脚的输入信号组合为 1 个信号，传送到 LSI 内部。因此，根据其他同一功能管脚的输入状态，有可能和输入信号不同的信号传送到 LSI 内部。分配到多个管脚的输入功能传送格式如表 15.6 所示。在 2 个以上的管脚使用以下任意功能时，必须考虑传送格式和注意信号的极性。

表 15.6 分配到多个管脚的输入功能传送格式

OR 型	AND 型
SCK0 ~ SCK2、RXD0 ~ RXD2、POE0、POE1、POE3*、POE8	IRQ0* ~ IRQ3

【注】* 只限 SH7125 的管脚。

OR 型：通过 OR 逻辑将多个管脚的输入信号组合为 1 个信号，传送到 LSI 内部。
AND 型：通过 AND 逻辑将多个管脚的输入信号组合为 1 个信号，传送到 LSI 内部。

- 管脚功能为输出功能的情况
能从所选的全部管脚输出相同信号。
- 2. 输入 / 输出端口和 IRQ 是多路复用管脚，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 15.3 和表 15.4 中 PFC 可设定的功能，否则就无法正常运行。

第 16 章 I/O 端口

SH7125 的端口由 A、B、E 和 F 共 4 个端口构成。端口 A 为 16 位、端口 B 为 5 位、端口 E 为 16 位输入 / 输出端口以及端口 F 为 8 位输入专用端口。

SH7124 的端口由 A、B、E 和 F 共 4 个端口构成。端口 A 为 8 位、端口 B 为 3 位、端口 E 为 12 位输入 / 输出端口以及端口 F 为 8 位输入专用端口。

各端口的管脚都是和其他功能兼用的多路复用管脚，通过管脚功能控制器（PFC）选择多路复用管脚的功能。

各端口有保存管脚数据的数据寄存器。

16.1 端口 A

如图 16.1 所示，SH7125 的端口 A 是有 16 个管脚的输入 / 输出端口。



图 16.1 端口 A (SH7125)

如图 16.2 所示，SH7124 的端口 A 是有 8 个管脚的输入 / 输出端口。

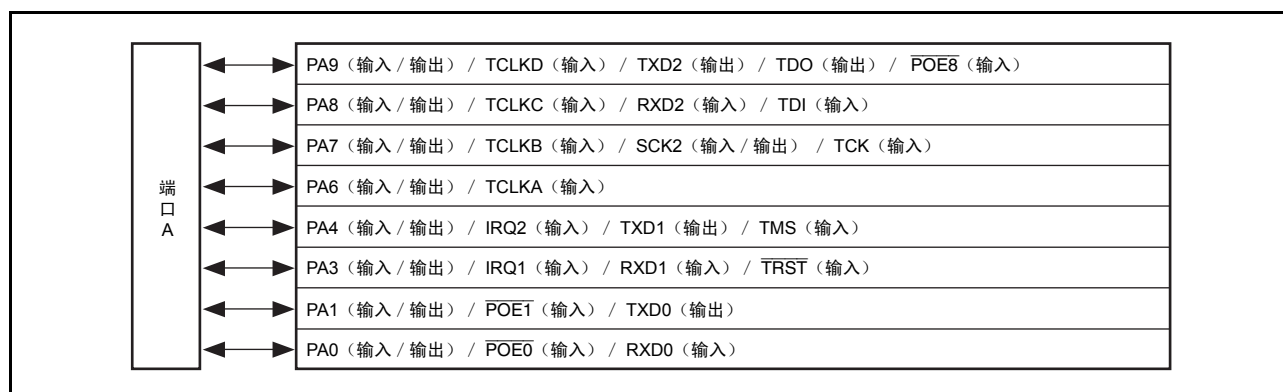


图 16.2 端口 A (SH7124)

16.1.1 寄存器说明

端口 A 是 16 位（SH7125）或者 8 位（SH7124）输入 / 输出端口，有以下寄存器。有关此寄存器的地址和各处理模式中的寄存器状态，请参照“第 20 章 寄存器一览”。

表 16.1 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'FFFFD11E	8、16

16.1.2 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。SH7125 的 PA15DR ~ PA0DR 位分别对应 PA15 ~ PA0 管脚（省略有关兼用功能的记述）。SH7124 的 PA9DR ~ PA6DR、PA4DR、PA3DR、PA1DR、PA0DR 位分别对应 PA9 ~ PA6、PA4、PA3、PA1、PA0 管脚（省略有关兼用功能的记述）。

在管脚功能为通用输出功能时，如果给 PADRL 写值，就从管脚输出该值；如果读 PADRL，就直接读寄存器的值，与管脚的状态无关。

在管脚功能为通用输入功能时，如果读 PADRL，就直接读管脚的状态而不读寄存器的值；如果给 PADRL 写值，就能将该值写到 PADRL，但是不影响管脚的状态。端口 A 的数据寄存器的读写操作如表 16.2 所示。

• PADRL（SH7125）

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 16.2
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	

- PADRL (SH7124)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PA9 DR	PA8 DR	PA7 DR	PA6 DR	-	PA4 DR	PA3 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	R	R	R	R	R	R/W	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
15 ~ 10	—	全 0	R	保留位 读写的值总是 0。
9	PA9DR	0	R/W	参照表 16.2
8	PA8DR	0	R/W	
7	PA7DR	0	R/W	
6	PA6DR	0	R/W	
5	—	0	R	保留位 读写的值总是 0。
4	PA4DR	0	R/W	参照表 16.2
3	PA3DR	0	R/W	
2	—	0	R	保留位 读写的值总是 0。
1	PA1DR	0	R/W	参照表 16.2
0	PA0DR	0	R/W	

表 16.2 端口 A 的数据寄存器 (PADRL) 的读写操作

- PADRL 的 bit15 ~ 0

PAIORH、L	管脚功能	读	写
0	通用输入	管脚的状态	能写 PADRL，但是不影响管脚的状态
	非通用输入	管脚的状态	能写 PADRL，但是不影响管脚的状态
1	通用输出	PADRL 的值	从管脚输出所写的值
	非通用输出	PADRL 的值	能写 PADRL，但是不影响管脚的状态

16.1.3 端口 A 的端口寄存器 L (PAPRL)

PAPRL 是 16 位可读写寄存器，与 PFC 的设定无关，总是能读管脚的状态。SH7125 的 PA15PR ~ PA0PR 位分别对应 PA15 ~ PA0 管脚（省略有关兼用功能的记述）。SH7124 的 PA9PR ~ PA6PR、PA4PR、PA3PR、PA1PR、PA0PR 位分别对应 PA9 ~ PA6、PA4、PA3、PA1、PA0 管脚（省略有关兼用功能的记述）。

- PAPRL (SH7125)

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	管脚的状态	R	与 PFC 的设定无关，读管脚的状态。写操作无效。
14	PA14PR	管脚的状态	R	
13	PA13PR	管脚的状态	R	
12	PA12PR	管脚的状态	R	
11	PA11PR	管脚的状态	R	
10	PA10PR	管脚的状态	R	
9	PA9PR	管脚的状态	R	
8	PA8PR	管脚的状态	R	
7	PA7PR	管脚的状态	R	
6	PA6PR	管脚的状态	R	
5	PA5PR	管脚的状态	R	
4	PA4PR	管脚的状态	R	
3	PA3PR	管脚的状态	R	
2	PA2PR	管脚的状态	R	
1	PA1PR	管脚的状态	R	
0	PA0PR	管脚的状态	R	

- PAPRL (SH7124)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	PA9 PR	PA8 PR	PA7 PR	PA6 PR	-	PA4 PR	PA3 PR	-	PA1 PR	PA0 PR
初始值:	0	0	0	0	0	0	*	*	*	*	0	*	*	0	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 10	—	全 0	R	保留位 读写的值总是 0。
9	PA9PR	管脚的状态	R	与 PFC 的设定无关，读管脚的状态。写操作无效。
8	PA8PR	管脚的状态	R	
7	PA7PR	管脚的状态	R	
6	PA6PR	管脚的状态	R	
5	—	0	R	保留位 读写的值总是 0。
4	PA4PR	管脚的状态	R	与 PFC 的设定无关，读管脚的状态。写操作无效。
3	PA3PR	管脚的状态	R	
2	—	0	R	保留位 读写的值总是 0。
1	PA1PR	管脚的状态	R	与 PFC 的设定无关，读管脚的状态。写操作无效。
0	PA0PR	管脚的状态	R	

16.2 端口 B

如图 16.3 所示，SH7125 的端口 B 是有 5 个管脚的输入 / 输出端口。

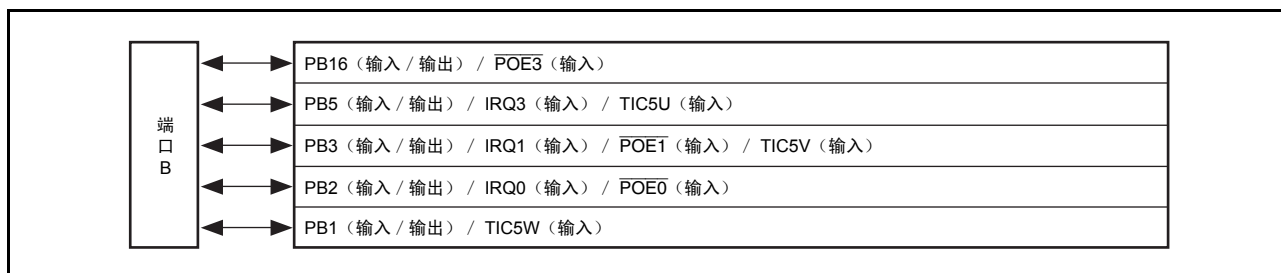


图 16.3 端口 B (SH7125)

如图 16.4 所示，SH7124 的端口 B 是有 3 个管脚的输入 / 输出端口。



图 16.4 端口 B (SH7124)

16.2.1 寄存器说明

端口 B 是 5 位 (SH7125) 或者 3 位 (SH7124) 输入 / 输出端口，有以下寄存器。有关此寄存器的地址和各处理模式中的寄存器状态，请参照“第 20 章 寄存器一览”。

表 16.3 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 H	PBDRH	R/W	H'0000	H'FFFFD180	8、16、32
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 H	PBPRH	R	—	H'FFFFD19C	8、16、32
端口 B 的端口寄存器 L	PBPRL	R	—	H'FFFFD19E	8、16

16.2.2 端口 B 的数据寄存器 H、L (PBDRH、PBDRL)

PBDRH 和 PBDRL 分别是 16 位可读写寄存器，保存端口 B 的数据。SH7125 的 PB16DR、PB5DR、PB3DR ~ PB1DR 位分别对应 PB16、PB5、PB3 ~ PB1 管脚（省略有关兼用功能的记述）。SH7124 的 PB5DR、PB3DR、PB1DR 位分别对应 PB5、PB3、PB1 管脚（省略有关兼用功能的记述）。

在管脚功能为通用输出功能时，如果给 PBDRH 或者 PBDRL 写值，就从管脚输出该值；如果读 PBDRH 或者 PBDRL，就直接读寄存器的值，与管脚的状态无关。

在管脚功能为通用输入功能时，如果读 PBDRH 或者 PBDRL，就直接读管脚的状态而不读寄存器的值；如果给 PBDRH 或者 PBDRL 写值，就能将该值写到 PBDRH 或者 PBDRL，但是不影响管脚的状态。端口 B 的数据寄存器的读写操作如表 16.4 所示。

• PBDRH (SH7125)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PB16DR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

位	位名	初始值	R/W	说 明
15 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	PB16DR	0	R/W	参照表 16.4

• PBDRH (SH7124)

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

位	位名	初始值	R/W	说 明
15 ~ 0	—	全 0	R	保留位 读写的值总是 0。

• PBDRL (SH7125)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PB5 DR	-	PB3 DR	PB2 DR	PB1 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	R/W	R/W	R/W	R

位	位名	初始值	R/W	说 明
15 ~ 6	—	全 0	R	保留位 读写的值总是 0。
5	PB5DR	0	R/W	参照表 16.4
4	—	0	R	保留位 读写的值总是 0。
3	PB3DR	0	R/W	参照表 16.4
2	PB2DR	0	R/W	
1	PB1DR	0	R/W	
0	—	0	R	保留位 读写的值总是 0。

• PBDRL (SH7124)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PB5 DR	-	PB3 DR	-	PB1 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	R/W	R	R/W	R

位	位名	初始值	R/W	说 明
15 ~ 6	—	全 0	R	保留位 读写的值总是 0。
5	PB5DR	0	R/W	参照表 16.4
4	—	0	R	保留位 读写的值总是 0。
3	PB3DR	0	R/W	参照表 16.4
2	—	0	R	保留位 读写的值总是 0。
1	PB1DR	0	R/W	参照表 16.4
0	—	0	R	保留位 读写的值总是 0。

表 16.4 端口 B 的数据寄存器 (PBDR) 的读写操作

- PBDRH 的 bit0 和 PBDRL 的 bit5、3 ~ 1

PBIOR	管脚功能	读	写
0	通用输入	管脚的状态	能写 PBDRH、L，但是不影响管脚的状态
	非通用输入	管脚的状态	能写 PBDRH、L，但是不影响管脚的状态
1	通用输出	PBDRH、L 的值	从管脚输出所写的值
	非通用输出	PBDRH、L 的值	能写 PBDRH、L，但是不影响管脚的状态

16.2.3 端口 B 的端口寄存器 H、L (PBPRH、PBPRL)

PBPRH 和 PBPRL 分别是 16 位可读写寄存器，与 PFC 的设定无关，总是能读管脚的状态。SH7125 的 PB16PR、PB5PR、PB3PR ~ PB1PR 位分别对应 PB16、PB5、PB3 ~ PB1 管脚（省略有关兼用功能的记述）。SH7124 的 PB5PR、PB3PR、PB1PR 位分别对应 PB5、PB3、PB1 管脚（省略有关兼用功能的记述）。

- PBPRH (SH7125)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PB16 PR
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	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	PB16PR	管脚的状态	R	与 PFC 的设定无关，读管脚的状态。写操作无效。

- PBPRH (SH7124)

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

位	位名	初始值	R/W	说 明
15 ~ 0	—	全 0	R	保留位 读写的值总是 0。

• PBPRL (SH7125)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PB5 PR	-	PB3 PR	PB2 PR	PB1 PR	-
初始值:	0	0	0	0	0	0	0	0	0	0	*	0	*	*	*	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 6	—	全 0	R	保留位 读写的值总是 0。
5	PB5PR	管脚的状态	R	与 PFC 的设定无关，读管脚的状态。写操作无效。
4	—	0	R	保留位 读写的值总是 0。
3	PB3PR	管脚的状态	R	与 PFC 的设定无关，读管脚的状态。写操作无效。
2	PB2PR	管脚的状态	R	
1	PB1PR	管脚的状态	R	
0	—	0	R	保留位 读写的值总是 0。

• PBPRL (SH7124)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	PB5 PR	-	PB3 PR	-	PB1 PR	-
初始值:	0	0	0	0	0	0	0	0	0	0	*	0	*	0	*	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 6	—	全 0	R	保留位 读写的值总是 0。
5	PB5PR	管脚的状态	R	与 PFC 的设定无关，读管脚的状态。写操作无效。
4	—	0	R	保留位 读写的值总是 0。
3	PB3PR	管脚的状态	R	与 PFC 的设定无关，读管脚的状态。写操作无效。
2	—	0	R	保留位 读写的值总是 0。
1	PB1PR	管脚的状态	R	与 PFC 的设定无关，读管脚的状态。写操作无效。
0	—	0	R	保留位 读写的值总是 0。

16.3 端口 E

如图 16.5 所示，SH7125 的端口 E 是有 16 个管脚的输入 / 输出端口。

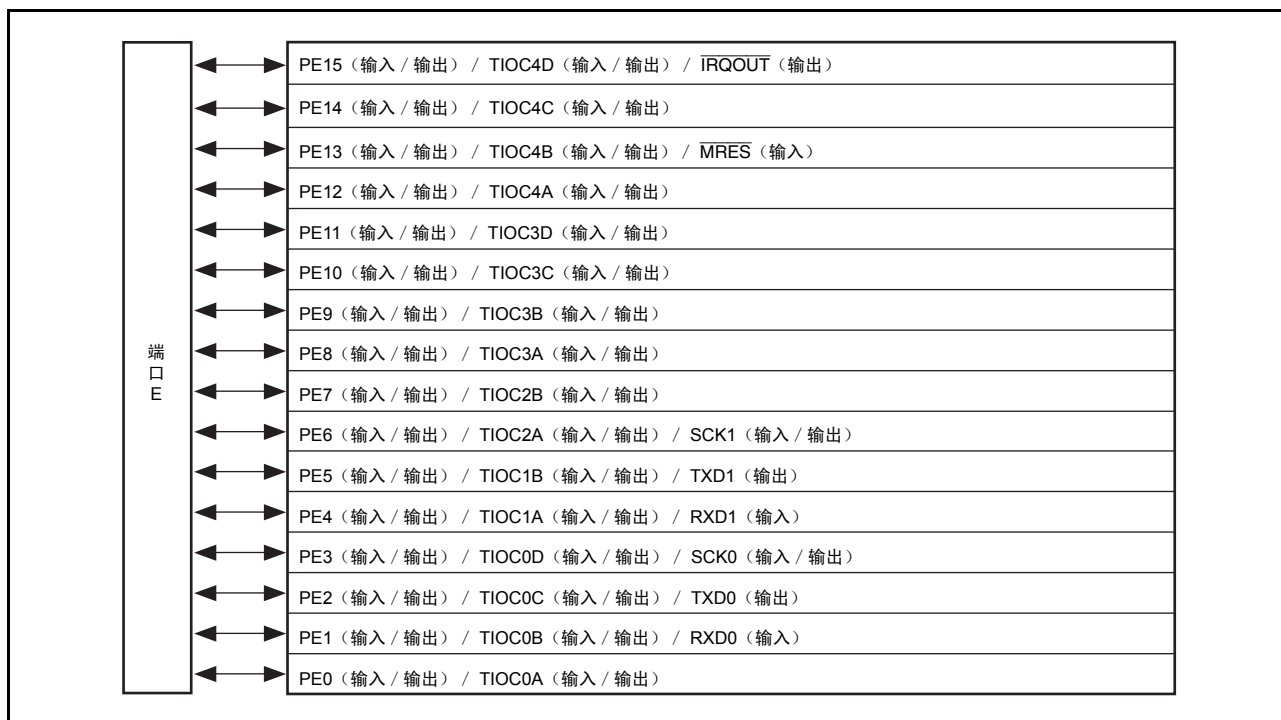


图 16.5 端口 E (SH7125)

如图 16.6 所示，SH7124 的端口 E 是有 12 个管脚的输入 / 输出端口。

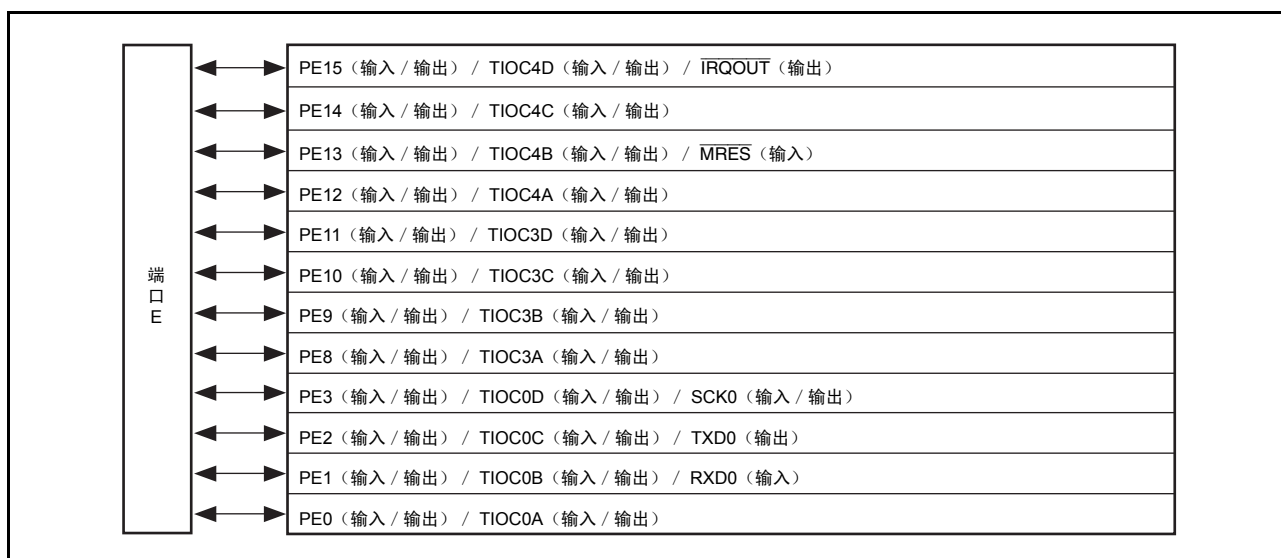


图 16.6 端口 E (SH7124)

16.3.1 寄存器说明

端口 E 是 16 位（SH7125）或者 12 位（SH7124）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中的寄存器状态请参照“第 20 章 寄存器一览”。

表 16.5 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
端口 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 L	PEPRL	R	—	H'FFFFD31E	8、16

16.3.2 端口 E 的数据寄存器 L（PEDRL）

PEDRL 是 16 位可读写寄存器，保存端口 E 的数据。SH7125 的 PE15DR ~ PE0DR 位分别对应 PE15 ~ PE0 管脚（省略有关兼用功能的记述）。SH7124 的 PE15DR ~ PE8DR、PE3DR ~ PE0DR 位分别对应 PE15 ~ PE8、PE3 ~ PE0 管脚（省略有关兼用功能的记述）。

在管脚功能为通用输出功能时，如果给 PEDRL 写值，就从管脚输出该值；如果读 PEDRL，就直接读寄存器的值，与管脚的状态无关。

在管脚功能为通用输入功能时，如果读 PEDRL，就直接读管脚的状态而不读寄存器的值；如果给 PEDRL 写值，就能将该值写到 PEDRL，但是不影响管脚的状态。端口 E 的数据寄存器的读写操作如表 16.6 所示。

- PEDRL（SH7125）

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 16.6
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	

- PEDRL (SH7124)

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	-	-	-	-	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	R	R	R	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 16.6
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 ~ 4	—	全 0	R	保留位 读写的值总是 0。
3	PE3DR	0	R/W	参照表 16.6
2	PE2DR	0	R/W	
1	PE1DR	0	R/W	
0	PE0DR	0	R/W	

表 16.6 端口 E 的数据寄存器 (PEDRL) 的读写操作

- PEDRL 的 bit15 ~ 0

PEIOR	管脚功能	读	写
0	通用输入	管脚的状态	能写 PEDRL, 但是不影响管脚的状态
	非通用输入	管脚的状态	能写 PEDRL, 但是不影响管脚的状态
1	通用输出	PEDRL 的值	从管脚输出所写的值
	非通用输出	PEDRL 的值	能写 PEDRL, 但是不影响管脚的状态

16.3.3 端口 E 的端口寄存器 L (PEPRL)

PEPRL 是 16 位可读写寄存器，与 PFC 的设定无关，总是能读管脚的状态。SH7125 的 PE15PR ~ PE0P 位分别对应 PE15 ~ PE0 管脚（省略兼用功能的记述）。SH7124 的 PE15PR ~ PE8PR、PE3PR ~ PE0PR 位分别对应 PE15 ~ PE8、PE3 ~ PE0 管脚（省略有关兼用功能的记述）。

- PEPRL (SH7125)

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	管脚的状态	R	与 PFC 的设定无关，读管脚的状态。写操作无效。
14	PE14PR	管脚的状态	R	
13	PE13PR	管脚的状态	R	
12	PE12PR	管脚的状态	R	
11	PE11PR	管脚的状态	R	
10	PE10PR	管脚的状态	R	
9	PE9PR	管脚的状态	R	
8	PE8PR	管脚的状态	R	
7	PE7PR	管脚的状态	R	
6	PE6PR	管脚的状态	R	
5	PE5PR	管脚的状态	R	
4	PE4PR	管脚的状态	R	
3	PE3PR	管脚的状态	R	
2	PE2PR	管脚的状态	R	
1	PE1PR	管脚的状态	R	
0	PE0PR	管脚的状态	R	

• PEPRL (SH7124)

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	-	-	-	-	PE3 PR	PE2 PR	PE1 PR	PE0 PR
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	PE15PR	管脚的状态	R	与 PFC 的设定无关，读管脚的状态。写操作无效。
14	PE14PR	管脚的状态	R	
13	PE13PR	管脚的状态	R	
12	PE12PR	管脚的状态	R	
11	PE11PR	管脚的状态	R	
10	PE10PR	管脚的状态	R	
9	PE9PR	管脚的状态	R	
8	PE8PR	管脚的状态	R	
7 ~ 4	—	全 0	R	保留位 读写的值总是 0。
3	PE3PR	管脚的状态	R	与 PFC 的设定无关，读管脚的状态。写操作无效。
2	PE2PR	管脚的状态	R	
1	PE1PR	管脚的状态	R	
0	PE0PR	管脚的状态	R	

16.4 端口 F

SH7125/24 的端口 F 是有 8 个管脚的输入专用端口。

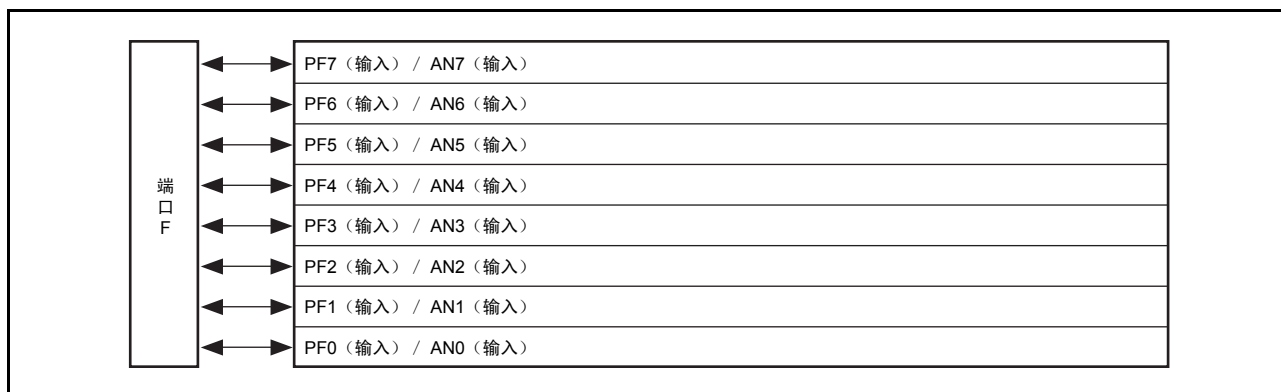


图 16.7 端口 F (SH7125/24)

16.4.1 寄存器说明

SH7125/24 的端口 F 是 8 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中的寄存器状态请参照“第 20 章 寄存器一览”。

表 16.7 寄存器结构

寄存器名	略称	R/W	初始值	地址	存取长度
端口 F 的数据寄存器 L	PFDRL	R	—	H'FFFFD382	8、16

16.4.2 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。SH7125/24 的 PF7DR ~ PF0DR 位分别对应 PF7 ~ PF0 管脚（省略有关兼用功能的记述）。

即使给这些位写值也被忽视，不影响管脚的状态。另外，如果读这些位，就直接读管脚的状态而不读位的值。但是，在对 A/D 转换器的模拟输入进行采样的期间能读到 1 的值。端口 F 的数据寄存器 L 的读写操作如表 16.8 所示。

• PFDRL (SH7125/24)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	PF7 DR	PF6 DR	PF5 DR	PF4 DR	PF3 DR	PF2 DR	PF1 DR	PF0 DR
初始值:	0	0	0	0	0	0	0	0	*	*	*	*	*	*	*	*
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说 明
15 ~ 8	—	全 0	R	保留位 读写的值总是 0。
7	PF7DR	管脚的状态	R	参照表 16.8
6	PF6DR	管脚的状态	R	
5	PF5DR	管脚的状态	R	
4	PF4DR	管脚的状态	R	
3	PF3DR	管脚的状态	R	
2	PF2DR	管脚的状态	R	
1	PF1DR	管脚的状态	R	
0	PF0DR	管脚的状态	R	

表 16.8 端口 F 的数据寄存器 L (PFDRL) 的读写操作

• PFDRL 的 bit7 ~ 0

管脚功能	读	写
通用输入	读管脚的状态	被忽视（不影响管脚的状态）
ANn 输入	读 1	被忽视（不影响管脚的状态）

第 17 章 闪存

本 LSI 内置 128KB 或者 64KB 的闪存，闪存的特点如下所示：

17.1 特点

- 容量

SH71253、SH71243: 128KB

SH71252、SH71242: 64KB

- 2 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。本模式能自动调整主机和本 LSI 之间的位速率。

用户编程模式：能通过任意的接口改写用户 MAT。

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。

- 通过下载内部程序，建立编程 / 擦除的接口

本 LSI 内置专用的编程 / 擦除的程序。在将此程序下载到内部 RAM 后，只要设定自变量参数就能进行编程 / 擦除，并且支持用户分支。

用户分支：以 128 字节为单位进行编程处理，编程处理由外加写脉冲和读验证等几个步骤构成；以 1 个分割块为单位进行擦除处理，擦除处理由几个处理步骤构成。能在各步骤之间设定用户处理程序的执行，此设定称为用户分支。

- 保护模式

保护模式有通过设定寄存器进行的软件保护和通过 FWE 管脚进行的硬件保护共 2 种，能设定闪存的编程 / 擦除的保护状态。

另外，如果在编程 / 擦除中检测到失控等异常情况，就转移到错误保护状态，并中断编程 / 擦除的处理。

- 编程 / 擦除的时间

闪存的编程时间在 128 字节同时编程时为 t_{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

闪存能改写到 N_{WEC} 次。

- 编程 / 擦除时的工作频率

编程 / 擦除时的工作频率最大为 40MHz。

17.2 概要

17.2.1 框图

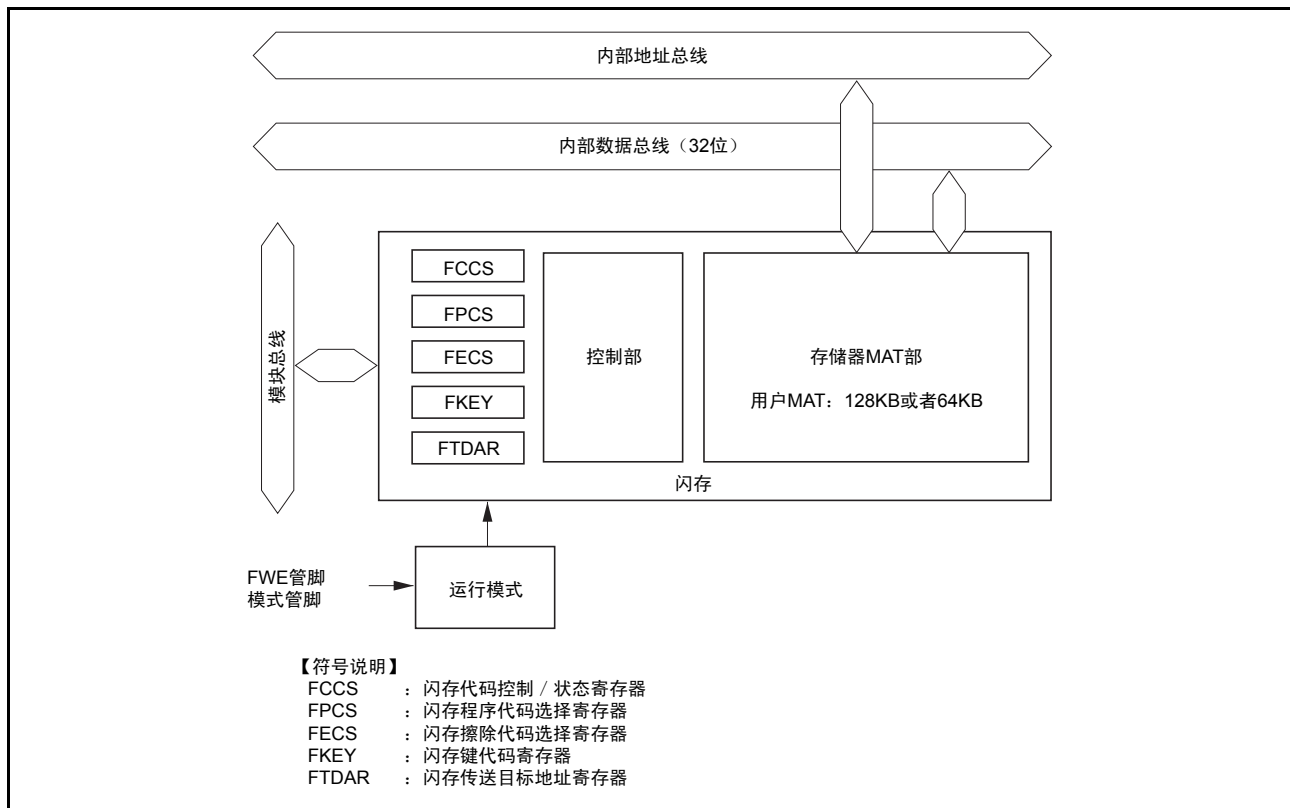


图 17.1 闪存的框图

17.2.2 运行模式

如果在复位状态下设定各模式管脚和 FWE 管脚，并进行复位解除，单片机就转移到如图 17.2 所示的各运行模式。各模式管脚和 FWE 管脚的设定请参照表 17.1。

1. 在用户模式中能读闪存，但是不能编程 / 擦除。
2. 在用户编程模式和引导模式中，能对闪存进行板上读 / 编程 / 擦除。
3. 在编程器模式中，使用 PROM 编程器对闪存进行读 / 编程 / 擦除操作。

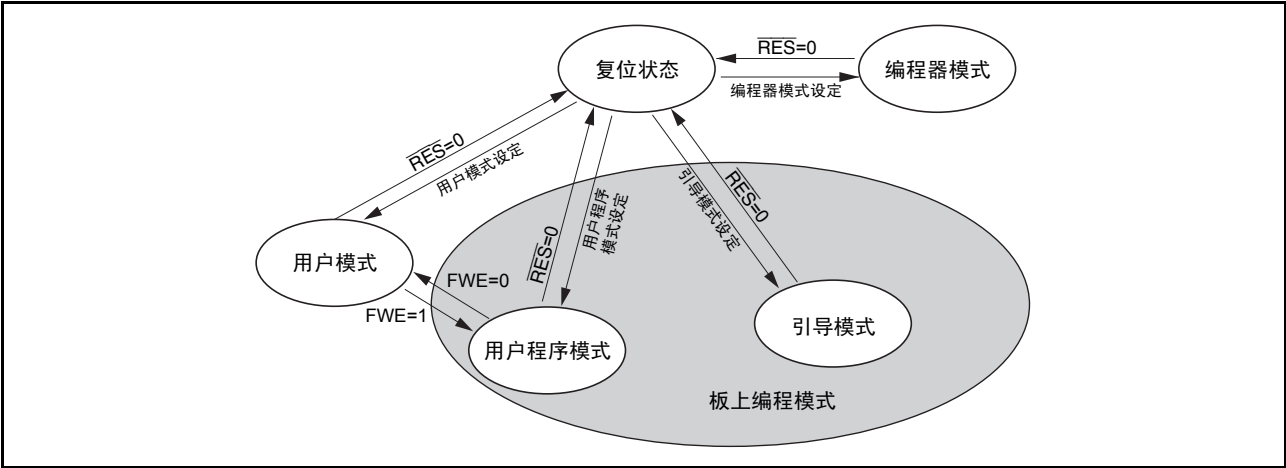


图 17.2 与闪存有关的模式转移图

表 17.1 FWE 管脚、MD 管脚的设定和运行模式

管脚	复位状态	用户模式	用户编程模式	引导模式	编程器模式
RES	0	1	1	1	与专用的 PROM 编程器的条件有关
FWE	0/1	0	1	1	
MD1	0/1	1	1	0	

17.2.3 模式比较

有关引导模式、用户编程模式和编程器模式的编程 / 擦除的比较如表 17.2 所示。

表 17.2 编程模式的比较

	引导模式	用户编程模式	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 MAT	用户 MAT
编程 / 擦除控制	命令方式	编程 / 擦除接口	—
全面擦除	有（自动）	有	有（自动）
块分割擦除	有 *	有	无
编程数据的传送	从主机经由 SCI	从任意器件经由 RAM	经由编程器
用户分支功能	无	有	无
复位开始时的启动 MAT	嵌入式程序保存 MAT	用户 MAT	嵌入式程序保存 MAT
向用户模式的转移	更改模式设定和复位	更改 FWE 设定	—

【注】 * 首先进行全面擦除，之后能进行特定块的擦除。

- 在引导模式中，首先全面擦除用户 MAT，然后通过命令方式对用户 MAT 进行编程，但是在进入此状态之前不能读取 MAT 的内容。

17.2.4 闪存结构

本 LSI 的闪存由 128KB 或者 64KB 的用户 MAT 构成。

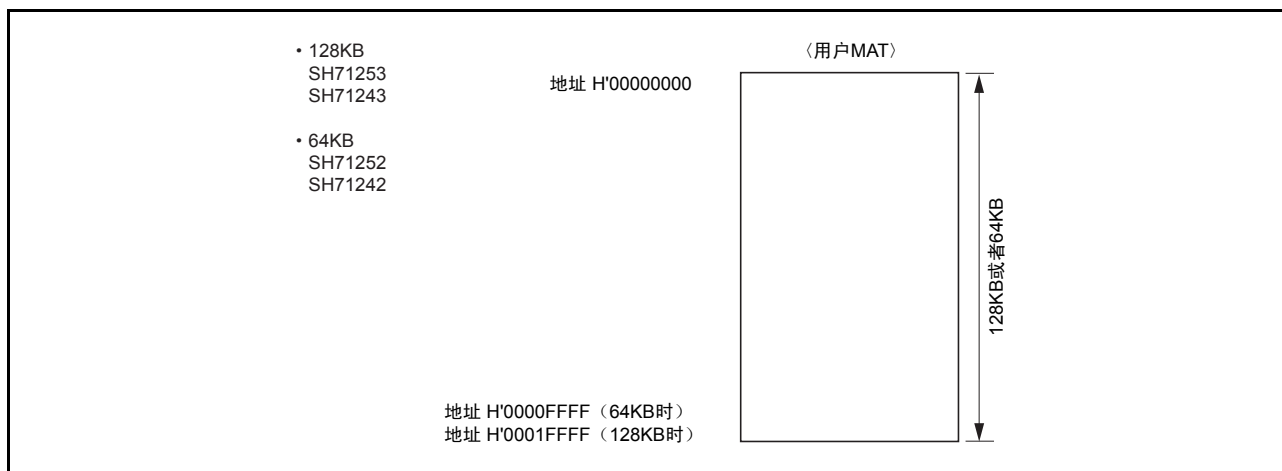


图 17.3 闪存结构图

17.2.5 块分割

如图 17.4 所示，用户 MAT 被分割为 64KB（128KB 的产品为 1 块）、32KB（1 块）和 4KB（8 块）。能以这种块分割为单位进行擦除，在擦除时用 EB0 ~ EB9 指定擦除块的号。

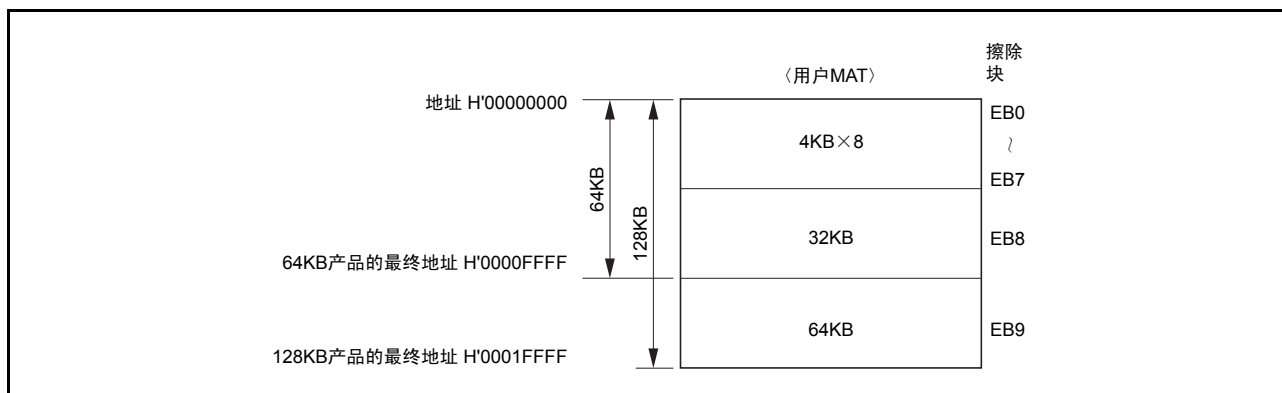


图 17.4 用户 MAT 的块分割

17.2.6 编程 / 擦除的接口

先将内部程序下载到内部 RAM，再通过接口寄存器 / 参数，指定编程地址 / 数据以及擦除块等进行编程 / 擦除。

在用户编程模式 / 用户引导模式中，这一连串的过程程序由用户建立，步骤的概要如下所示。详细内容在“17.5.2 用户编程模式”中说明。

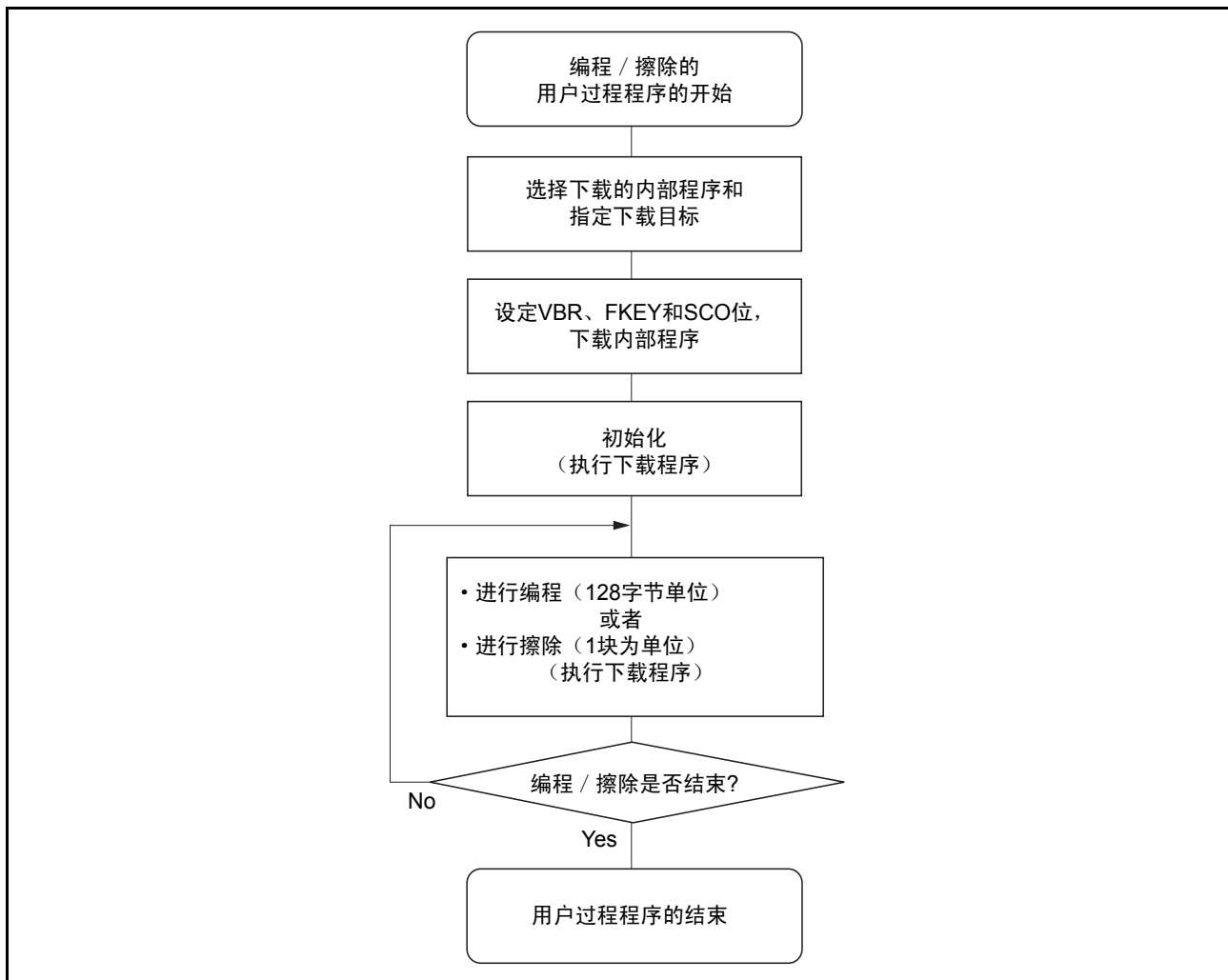


图 17.5 用户过程程序的概要

(1) 下载的内部程序的选择和下载目标的指定

本 LSI 内置编程 / 擦除的相关程序，能将此程序下载到内部 RAM。通过设定编程 / 擦除的接口寄存器的对应位，选择要下载的内部程序，并能通过 FTDAR 寄存器，指定下载目标的地址。

(2) 内部程序的下载

在将 CPU 的 VBR 寄存器设定为 H'84000000 后，通过设定编程 / 擦除的接口寄存器的闪存键代码寄存器 FKEY 和闪存代码控制 / 状态寄存器 FCCS 的 SCO 位，自动下载内部程序。

在下载中，闪存 MAT 和嵌入式程序保存区调换。另外，因为不能在编程 / 擦除时读闪存，所以从编程 / 擦除的下载到完成的一连串过程程序必须在闪存以外的存储器（如内部 RAM 等）执行。

因为下载的结果能返回给编程 / 擦除的接口参数，所以能确认是否正常下载。

另外，在下载结束后能更改 VBR。

(3) 编程 / 擦除的初始化

在进行编程 / 擦除前设定工作频率和用户分支，必须将用户分支目标设定在非内部闪存区和非下载内部程序的区域。通过编程 / 擦除的接口参数进行这些设定。

(4) 编程 / 擦除

为了编程 / 擦除，需要将 FWE 管脚设定为高电平，并且将模式设定为用户编程模式。

在编程时，以 128 字节为单位指定编程数据 / 编程目标地址。

在擦除时，以 1 个擦除块为单位指定擦除块。

通过编程 / 擦除的接口参数，设定这些指定并启动内部程序。通过 JSR 指令或者 BSR 指令调用（子程序调用）内部 RAM 中的特定地址，执行内部程序，执行结果返回给编程 / 擦除的接口参数。

对于闪存的编程，需要预先擦除对象区。

编程 / 擦除处理中的中断处理有限制和注意事项，详细内容请参照“17.7.1 执行编程 / 擦除过程中的中断”。

(5) 连续进行编程 / 擦除的情况

如果 128 字节的编程或者 1 块的擦除不能结束处理，就需要更新编程地址 / 数据或者擦除块号，连续进行编程 / 擦除。

因为在处理结束后被下载的内部程序仍留在内部 RAM 中，所以在连续进行相同的处理时不需要下载和初始化。

17.3 管脚结构

闪存由表 17.3 所示的管脚控制。

表 17.3 管脚结构

管脚名	略称	输入 / 输出	功能
加电复位	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
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）

17.4 寄存器结构

17.4.1 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 17.4 所示。

在存取闪存时，有读模式和写模式等几种运行模式。运行模式和使用的寄存器 / 参数的对应如表 17.5 所示。

表 17.4 (1) 寄存器结构

寄存器名称	略称 *3	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
闪存传送目标地址寄存器	FTDAR	R/W	H'00	H'FFFFCC06	8

【注】 *1 SCO 位为只写（读取的值总是 0），而其他位为只读，。

*2 给 FWE 管脚输入低电平时的 FEW 位的初始值为 0。

给 FWE 管脚输入高电平时的 FEW 位的初始值为 1。

*3 各寄存器只在字节存取时有效。

表 17.4 (2) 参数结构

参数名称	略称	R/W	初始值	分配	存取长度
下载的正常 / 失败结果	DPFR	R/W	不定	内部 RAM*	8、16、32
闪存的正常 / 失败结果	FPFR	R/W	不定	CPU 的 R0	8、16、32
闪存多用途地址区	FMPAR	R/W	不定	CPU 的 R5	8、16、32
闪存多用途数据目标区	FMPDR	R/W	不定	CPU 的 R4	8、16、32
闪存的擦除块选择	FEBS	R/W	不定	CPU 的 R4	8、16、32
闪存编程 / 擦除的频率控制	FPEFEQ	R/W	不定	CPU 的 R4	8、16、32
闪存用户分支地址的设定	FUBRA	R/W	不定	CPU 的 R5	8、16、32

【注】 * FTDAR 寄存器指定的内部 RAM 区起始地址的 1 字节有效。

表 17.5 使用的寄存器 / 参数和对象模式

		下载	初始化	编程	擦除	读
编程 / 擦除的接口寄存器	FCCS	○	—	—	—	—
	FPCS	○	—	—	—	—
	PECS	○	—	—	—	—
	FKEY	○	—	○	○	—
	FTDAR	○	—	—	—	—
编程 / 擦除的接口参数	DPFR	○	—	—	—	—
	FPFR	—	○	○	○	—
	FPEFEQ	—	○	—	—	—
	FUBRA	—	○	—	—	—
	FMPAR	—	—	○	—	—
	FMPDR	—	—	○	—	—
	FEBS	—	—	—	○	—

17.4.2 编程 / 擦除的接口寄存器

编程 / 擦除的接口寄存器都是 8 位寄存器，并只能以字节存取。

(1) 闪存代码控制 / 状态寄存器 (FCCS)

FCCS 由 FWE 管脚状态的监视位、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	-	-	FLER	-	-	-	SCO
初始值:	1/0	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	FWE	1/0	R	闪存编程的允许位 FWE 位监视 FWE 管脚（对闪存编程 / 擦除进行硬件保护）的输入电平。根据 FWE 管脚状态，初始值为 0 或者 1。 0: 给 FWE 管脚输入低电平时（硬件保护状态） 1: 给 FWE 管脚输入高电平时
6、5	—	全 0	R	保留位 读写的值总是 0。
4	FLER	0	R	闪存错误 此位表示在闪存的编程 / 擦除中发生错误。如果 FLER=1，闪存就转移到错误保护状态。 另外，在 FLER=1 时，由于闪存内部加有高电压，为了减少闪存的损伤，必须在长于通常的 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] 当加电复位时 1: 表示在闪存的编程 / 擦除中发生错误。闪存的编程 / 擦除保护（错误保护）有效 [置位条件] 请参照“17.6.3 错误保护”。
3 ~ 1	—	全 0	R	保留位 读写的值总是 0。

位	位名	初始值	R/W	说 明
0	SCO	0	(R)/W	源程序的复制操作 此位请求将内部的编程 / 擦除程序下载到内部 RAM。如果给本位写 1，FPCS/FECS 寄存器选择的内部程序就自动下载到 FTDAR 寄存器指定的内部 RAM 区。为了给本位写 1，需要给 FKEY 寄存器写 H'A5，并且在内部 RAM 中执行。在给本位写 1 后，必须立刻执行 4 条 NOP 指令。 有关下载中的中断和下载时间请分别参照“17.7.1 执行编程 / 擦除过程中的中断”和“17.7.2 其他注意事项”。另外，因为在下载结束时此位被清 0，所以读不到此位为 1 的状态。 因为在通过 SCO 位进行下载时，伴有内部程序保存区的存储体转换的特殊中断处理，所以必须在下载请求（SCO=1）前将 VBR 的值设定为 H'84000000，如果不设定 VBR 就会失控。当确认下载已结束，能更改 VBR。另外，在使用 SCO 功能时，必须使用 FWE 管脚为高电平的模式。 0：不将内置的编程 / 擦除程序下载到内部 RAM。 [清除条件] 当下载结束时 1：发生将内置的编程 / 擦除程序下载到内部 RAM 的请求。 [置位条件] 在满足以下全部条件的状态下写 1 - 给 FKEY 寄存器写 H'A5 - 在内部 RAM 中执行

(2) 闪存编程代码选择寄存器（FPCS）

FPCS 选择与编程有关的内部程序的下载。

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

位	位名	初始值	R/W	说 明
7 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	PPVS	0	R/W	编程脉冲 Single 选择编程的程序。 0：不选择内部编程程序。 [清除条件] 当传送结束时被清除。 1：选择内部编程程序。

(3) 闪存擦除代码选择寄存器 (FECS)

FECS 选择与擦除有关的内部程序的下载。

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

位	位名	初始值	R/W	说 明
7 ~ 1	—	全 0	R	保留位 读写的值总是 0。
0	EPVB	0	R/W	擦除脉冲验证块 选择内部擦除程序。 0: 不选择内部擦除程序。 [清除条件] 当传送结束时 1: 选择内部擦除程序。

(4) 闪存键代码寄存器 (FKEY)

FKEY 允许内部程序的下载和闪存编程 / 擦除的软件保护。在给用于下载内部程序的 SCO 位写 1 前或者在执行已下载的编程 / 擦除程序前, 如果未写键代码, 就不能执行各自的处理。

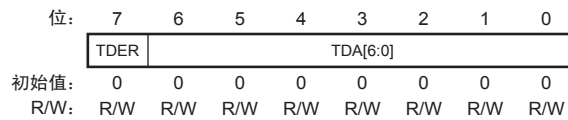
位:	7	6	5	4	3	2	1	0
	K[7:0]							
初始值:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7 ~ 0	K[7:0]	全 0	R/W	键代码 只在给 FKEY 寄存器写 H'A5 后, SCO 位的写操作才有效。如果给 FKEY 寄存器写 H'A5 以外的值, 就无法给 SCO 位写 1, 导致内部程序无法下载到内部 RAM。 只在给 FKEY 寄存器写 H'5A 后, 才能进行闪存的编程 / 擦除。如果给 FKEY 寄存器写 H'5A 以外的值, 即使执行内部编程 / 擦除程序也不能进行闪存的编程 / 擦除。 H'A5: 允许写 SCO 位 (如果值不为 H'A5, 就无法设定 SCO 位)。 H'5A: 允许编程 / 擦除 (如果值不为 H'5A, 就为软件保护状态)。 H'00: 初始值

(5) 闪存传送目标地址寄存器 (FTDAR)

FTDAR 指定内部程序下载目标的内部 RAM 地址。

必须在给 FCCS 寄存器的 SCO 位写 1 前设定此寄存器。初始值 H'00 表示内部 RAM 的起始地址 (H'FFFA000)。



位	位名	初始值	R/W	说 明
7	TDER	0	R/W	传送目标地址的设定错误 如果 bit6 ~ 0 (TDA6 ~ TDA0) 指定的下载起始地址有错误, 就被置 1。判断地址指定错误的方法: 将 FCCS 寄存器的 SCO 位设定为 1, 在下载处理执行结束时, 判断 TDA6 ~ TDA0 的值是否在 H'02 ~ H'04 的范围内。必须在将 SCO 位设定为 1 前, 将 FTDAR 寄存器设定在 H'02 ~ H'04 的范围内, 并且将此位的值置 0。 0: TDA6 ~ TDA0 的设定为正常值 1: 表示 TDER、TDA6 ~ TDA0 的设定值为 H'00 ~ H'01 或 H'05 ~ H'FF, 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'02 ~ 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'00、H'01、H'05 ~ H'7F 的值, 否则就在下载处理中将此寄存器的 bit7 (TDER) 置 1 并且不进行内部程序的下载。 H'02: 将下载的起始地址设定为 H'FFFA000 H'03: 将下载的起始地址设定为 H'FFFA800 H'04: 将下载的起始地址设定为 H'FFFB000 H'00、H'01、H'05 ~ H'7F: 不能设定, 否则就在下载中 bit7 (TDER) 变为 1 并且中断下载处理。

17.4.3 编程 / 擦除的接口参数

编程 / 擦除的接口参数对已下载的内部程序指定工作频率、用户分支目标地址、编程数据的保存位置、编程的目标地址、擦除块等以及进行处理结果的交换，此参数使用 CPU 的通用寄存器（R4、R5 和 R0）或者内部 RAM 区。初始值不定。

在下载时，保存 CPU 的全部寄存器；在初始化和执行内部程序时，保存除 R0 以外的 CPU 寄存器。R0 保存处理结果的返回值。因为堆栈区用作寄存器的保存和工作区，所以在开始处理时必须确保堆栈区（使用的堆栈区的容量最大为 128 字节）。

编程 / 擦除的接口参数被用于以下 4 个项目：

1. 下载控制
2. 编程 / 擦除前的初始化
3. 进行编程
4. 进行擦除

各项使用的参数都不相同，其对应表如表 17.6 所示。

在此，FPFR 参数返回初始化、编程和擦除的处理结果，但是根据处理内容，位的含意不同。请参照各项处理的 FPFR 说明部分。

表 17.6 使用的参数和对象模式

参数名	略称	下载	初始化	编程	擦除	R/W	初始值	分配
下载的正常 / 失败结果	DPFR	○	—	—	—	R/W	不定	内部 RAM*
闪存的正常 / 失败结果	FPFR	—	○	○	○	R/W	不定	CPU 的 R0
闪存编程 / 擦除的频率控制	FPEFEQ	—	○	—	—	R/W	不定	CPU 的 R4
闪存用户分支地址的设定	FUBRA	—	○	—	—	R/W	不定	CPU 的 R5
闪存多用途地址区	FMPAR	—	—	○	—	R/W	不定	CPU 的 R5
闪存多用途数据目标区	FMPDR	—	—	○	—	R/W	不定	CPU 的 R4
闪存擦除块的选择	FEBS	—	—	—	○	R/W	不定	CPU 的 R4

【注】* FTDAR 寄存器指定的下载目标起始地址的 1 字节

(1) 下载控制

通过将 SCO 位设定为 1，自动下载内部程序。被下载的内部 RAM 区是从 FTDAR 寄存器指定的起始地址开始的 3KB 容量。有关内部 RAM 的地址映像请参照图 17.10。

下载控制由前面所述的编程 / 擦除的接口寄存器设定，返回值由 DPFR 参数传递。

- 下载的正常 / 失败结果参数（DPFR：FTDAR 寄存器指定的内部 RAM 起始地址的 1 字节）

这是下载结果的返回值，必须通过此参数的值判断下载是否已执行完毕。由于不能确认是否已将 SCO 位设定为 1，所以必须在开始下载前（将 SCO 位设定为 1 之前），将 FTDAR 寄存器指定的内部 RAM 起始地址的 1 字节设定为下载返回值以外的值（H'FF 等），然后进行准确的判断。有关下载结果的检查方法请参照“17.5.2（2.5）”项。

位:	7	6	5	4	3	2	1	0
	-	-	-	-	-	SS	FK	SF
初始值:	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7 ~ 3	—	不定	R/W	未使用位 返回 0。
2	SS	不定	R/W	源选择的错误检测位 1 次操作只能指定 1 种可下载的内部程序。如果选择 2 种以上或者不选择或者选择未被映像，就会发生错误。 0：正常选择了下载程序 1：发生下载错误（多重选择或者选择了未被映像）
1	FK	不定	R/W	闪存键寄存器的错误检测位 此位返回检查 FKEY 寄存器的值是否为 H'A5 的结果。 0：FKEY 寄存器的设定值正常（FKEY=H'A5） 1：FKEY 寄存器的设定值错误（FKEY 不为 H'A5）
0	SF	不定	R/W	成功 / 失败位 此位返回下载是否已正常结束。 0：下载正常结束（无错误） 1：下载异常结束（发生错误）

(2) 编程 / 擦除的初始化

在被下载的编程 / 擦除内部程序中包含初始化程序。

在编程 / 擦除中，由 CPU 指令构成了所定时间宽的等待循环，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。

将这些设定作为下载的编程 / 擦除程序参数，在初始化程序中进行设定。

- 闪存编程 / 擦除的频率控制参数（FPEFEQ: CPU 的通用寄存器 R4）

这是设定 CPU 工作频率的参数。

本 LSI 的工作频率范围请参照“21.3.1 时钟时序”。

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

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	F15	F14	F13	F12	F11	F10	F9	F8	F7	F6	F5	F4	F3	F2	F1	F0
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
31 ~ 16	—	不定	R/W	未使用位 必须置 0。
15 ~ 0	F15 ~ F0	不定	R/W	频率的设定位 设定 CPU 的工作频率，必须按以下方法算出设定值： 1. 将以 MHz 单位表示的工作频率取到小数点后 2 位，小数点后的第 3 位四舍五入。 2. 将扩大 100 倍的值转换为 2 进制数，并写到 FPEFEQ 参数（通用寄存器 R4）。作为具体例子，在 CPU 的工作频率为 28.882MHz 时，设定值如下： 1. 将 28.882 的小数点后的第 3 位四舍五入，得到 28.88。 2. 将 28.88×100=2888 转换为 2 进制数，得到的 b'0000,1011,0100,1000（H'0B48）设定到 R4。

- 闪存用户分支地址的设定参数（FUBRA: CPU 的通用寄存器 R5）

这是设定用户分支目标地址的参数。能按编程 / 擦除时决定的处理单位执行已设定的用户程序。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	UA31	UA30	UA29	UA28	UA27	UA26	UA25	UA24	UA23	UA22	UA21	UA20	UA19	UA18	UA17	UA16
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UA15	UA14	UA13	UA12	UA11	UA10	UA9	UA8	UA7	UA6	UA5	UA4	UA3	UA2	UA1	UA0
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
31 ~ 0	UA31 ~ UA0	不定	R/W	用户分支目标地址 当不需要用户分支时，必须设定 H'84000000。必须将用户分支目标地址设定在非内部闪存区，或者设定在非传送内部程序的 RAM，或者设定在外部总线空间。必须注意：不能转移到没有执行码的区域，以免失控，并且不能破坏内部程序的下载区和堆栈区，否则就不能保证闪存的值。在用户分支目标的处理中，不能启动内部程序的下载程序、初始化程序和编程 / 擦除程序。不能保证从用户分支目标返回时的编程 / 擦除。另外，不能改写已准备好的编程数据。 必须保存通用寄存器 R8 ~ R15。通用寄存器 R0 ~ R7 不需保存而能被直接使用。 另外，不能在用户分支目标的处理中改写编程 / 擦除的接口寄存器。 在用户分支处理结束后，必须使用 RTS 指令返回到编程 / 擦除程序。 有关用户分支处理的执行间隔，请参照 “17.7.2 (2) 用户分支处理的间隔”。

- 闪存的正常 / 失败结果参数（FPFR：CPU 的通用寄存器 R0）

在此说明作为初始化处理结果返回值的 FPFR。

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

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

位	位名	初始值	R/W	说 明
31 ~ 3	—	不定	R/W	未使用位 返回 0。
2	BR	不定	R/W	用户分支的错误检测位 检查被指定的用户分支目标地址是否在非下载编程 / 擦除相关程序的保存区，并返回其检查结果。 0: 用户分支地址的设定值正常 1: 用户分支地址的设定值异常
1	FQ	不定	R/W	频率的错误检测位 检查被指定的 CPU 工作频率是否在支持工作频率范围内，并返回其检查结果。 0: 工作频率的设定值正常 1: 工作频率的设定值异常
0	SF	不定	R/W	成功 / 失败位 返回初始化是否正常结束。 0: 初始化正常结束（无错误） 1: 初始化异常结束（发生错误）

(3) 编程的执行

在进行闪存的编程中，需要将用户 MAT 中的编程目标地址和编程数据传递给已下载的编程程序。

1. 必须将用户 MAT 中的编程目标起始地址设定到通用寄存器 R5。此参数称为 FMPAR（闪存多用途地址区参数）。

因为编程数据总是以 128 字节为单位，所以必须将用户 MAT 中的编程起始地址的边界地址的低 8 位（A7～A0）置 H'00 或者 H'80。

2. 必须将用户 MAT 的编程数据准备到连续的区域。编程数据必须在能由 CPU 的 MOV.B 指令存取的连续空间并且在内部闪存空间以外。

如果要写的的数据未满 128 字节，必须填写虚码（H'FF），准备 128 字节的编程数据。

必须将保存准备好的编程数据的区域起始地址设定到通用寄存器 R4。此参数称为 FMPDR（闪存多用途数据目标区参数）。

有关编程处理过程的详细内容在“17.5.2 用户编程模式”中说明。

- 闪存多用途地址区参数（FMPAR：CPU 的通用寄存器 R5）

设定用户 MAT 中的编程目标起始地址。

如果设定为闪存空间以外的地址就发生错误。

另外，编程目标起始地址必须以 128 字节为边界，否则会发生错误。这些错误反映到 FPFR 参数的 bit1（WA 位）。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MOA31	MOA30	MOA29	MOA28	MOA27	MOA26	MOA25	MOA24	MOA23	MOA22	MOA21	MOA20	MOA19	MOA18	MOA17	MOA16
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MOA15	MOA14	MOA13	MOA12	MOA11	MOA10	MOA9	MOA8	MOA7	MOA6	MOA5	MOA4	MOA3	MOA2	MOA1	MOA0
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
31～0	MOA31 ～ MOA0	不定	R/W	MOA31～MOA0 保存用户 MAT 中的编程目标起始地址，从在此指定的用户 MAT 起始地址开始连续写 128 字节，所以指定的编程目标起始地址以 128 字节为边界，并且 MOA6～MOA0 总是 0。

- 闪存多用途数据目标区参数（FMPDR：CPU 的通用寄存器 R4）

设定用户 MAT 的编程数据保存区的起始地址。如果编程数据的保存目标在闪存内就发生错误，此错误反映到 FPFR 参数的 bit2（WD 位）。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MOD31	MOD30	MOD29	MOD28	MOD27	MOD26	MOD25	MOD24	MOD23	MOD22	MOD21	MOD20	MOD19	MOD18	MOD17	MOD16
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MOD15	MOD14	MOD13	MOD12	MOD11	MOD10	MOD9	MOD8	MOD7	MOD6	MOD5	MOD4	MOD3	MOD2	MOD1	MOD0
初始值:	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定	不定
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
31 ~ 0	MOD31 ~ MOD0	不定	R/W	MOD31 ~ MOD0 保存用户 MAT 的编程数据保存区的起始地址，从在此指定的起始地址开始将连续 128 字节的数据写到用户 MAT。

- 闪存的正常 / 失败参数（FPFR：CPU 的通用寄存器 R0）

在此说明作为编程处理结果返回值的 FPFR。

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

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

位	位名	初始值	R/W	说 明
31 ~ 7	—	不定	R/W	未使用位 返回 0。
6	MD	不定	R/W	编程模式相关设定的错误检测位 返回 FWE 管脚的输入值是高电平并且不是错误保护状态的检查结果。 在 FWE 管脚为低电平时或者在错误保护状态时写 1，并能通过 FCCS 寄存器的 bit7（FWE）和 bit4（FLER）确认这些状态。有关错误保护状态的转移条件，请参照“17.6.3 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	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、或者从用户分支处理返回时一部分闪存相关寄存器被改写，此位就返回 1。如果因这些原因而此位为 1，用户 MAT 就很有可能在半途被改写，所以必须在解除错误原因后从擦除开始重新执行。 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）

位	位名	初始值	R/W	说 明
4	FK	不定	R/W	闪存键寄存器的错误检测位 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=H'5A) 1: FKEY 寄存器的设定值错误 (FKEY 不为 H'5A)
3	—	不定	R/W	未使用位 返回 0。
2	WD	不定	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	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 • 指定的地址不是以 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	R/W	成功 / 失败位 此位返回编程处理是否已正常结束。 0: 正常结束 (无错误) 1: 异常结束 (发生错误)

(4) 擦除的执行

在进行闪存的擦除中，需要将用户 MAT 中的擦除块号传递给已下载的擦除程序，并将它设定到 FEBS 参数（通用寄存器 R4）。

从 0 ~ 15 的块号中指定 1 块。

有关擦除处理过程的详细内容在“17.5.2 用户编程模式”中说明。

- 闪存擦除块的选择参数（FEBS：CPU 的通用寄存器 R4）

指定擦除块号，不能指定多个块号。

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

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

位	位名	初始值	R/W	说 明
31 ~ 8	—	不定	R/W	未使用位 必须置 0。
7 ~ 0	EBS[7:0]	不定	R/W	- 为 128KB 闪存时 在 0 ~ 9 的范围内设定擦除块号。0 和 9 分别对应块 EB0 和块 EB9，如果设定 0 ~ 9（H'00 ~ H'09）以外的数就出错。 - 为 64KB 闪存时 在 0 ~ 8 的范围内设定擦除块号。0 和 8 分别对应块 EB0 和块 EB8，如果设定 0 ~ 8（H'00 ~ H'08）以外的数就出错。

- 闪存的正常 / 失败结果参数（FPFR：CPU 的通用寄存器 R0）

在此说明作为擦除处理结果返回值的 FPFR。

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

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

位	位名	初始值	R/W	说 明
31 ~ 7	—	不定	R/W	未使用位 返回 0。
6	MD	不定	R/W	擦除模式相关设定的错误检测位 返回 FWE 管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 管脚为低电平时或者在错误保护状态时写 1，并能通过 FCCS 寄存器的 bit7（FWE）和 bit4（FLER）确认这些状态。另外，有关错误保护状态的转移条件请参照“17.6.3 错误保护”。 0: FWE、FLER 状态正常（FWE=1、FLER=0） 1: FWE=0 或者 FLER=1，处于不能擦除的状态
5	EE	不定	R/W	擦除时的错误检测位 如果不能擦除用户 MAT 或者从用户分支处理返回时一部分闪存相关寄存器被改写，此位就返回 1。如果因这些原因而此位为 1，用户 MAT 就很有可能在途中被擦除，所以必须在解除错误原因后重新进行擦除。 0: 擦除处理正常结束 1: 擦除处理异常结束（不能保证擦除结果）
4	FK	不定	R/W	闪存键寄存器的错误检测位 返回开始擦除处理前的 FKEY 寄存器值的检查结果。 0: FKEY 寄存器的设定值正常（FKEY=H'5A） 1: FKEY 寄存器的设定值错误（FKEY 不为 H'5A）
3	EB	不定	R/W	擦除块的选择错误检测位 返回被指定的擦除块号是否在用户 MAT 的块范围内的检查结果。 0: 擦除块号的设定值正常 1: 擦除块号的设定值异常
2、1	—	不定	R/W	未使用位 返回 0。
0	SF	不定	R/W	成功 / 失败位 返回擦除处理是否已正常结束。 0: 正常结束（无错误） 1: 异常结束（发生错误）

17.5 板上编程模式

如果在板上编程模式中设定管脚并启动复位，就转移到能编程 / 擦除内部闪存的板上编程状态。板上编程模式有用户编程模式、用户引导模式和引导模式 3 种运行模式。

各模式转移的管脚设定方法请参照表 17.1，闪存的各模式的状态转移图请参照图 17.2。

17.5.1 引导模式

引导模式是使用内部 SCI 从主机发送控制命令和编程数据的方式，进行用户 MAT 的编程 / 擦除。需要预先给主机准备发送控制命令的工具和编程数据，并将使用的 SCI 通信模式设定为异步模式。如果在将本 LSI 的管脚设定为引导模式后启动复位，就启动预先嵌入在单片机内部的引导程序，并在自动调整 SCI 位速率后通过控制命令方式和主机进行通信。

引导模式时的系统构成如图 17.6 所示，引导模式的管脚设定请参照表 17.1。虽然引导模式的 NMI 和其他中断被忽视，但是也必须使中断不发生。

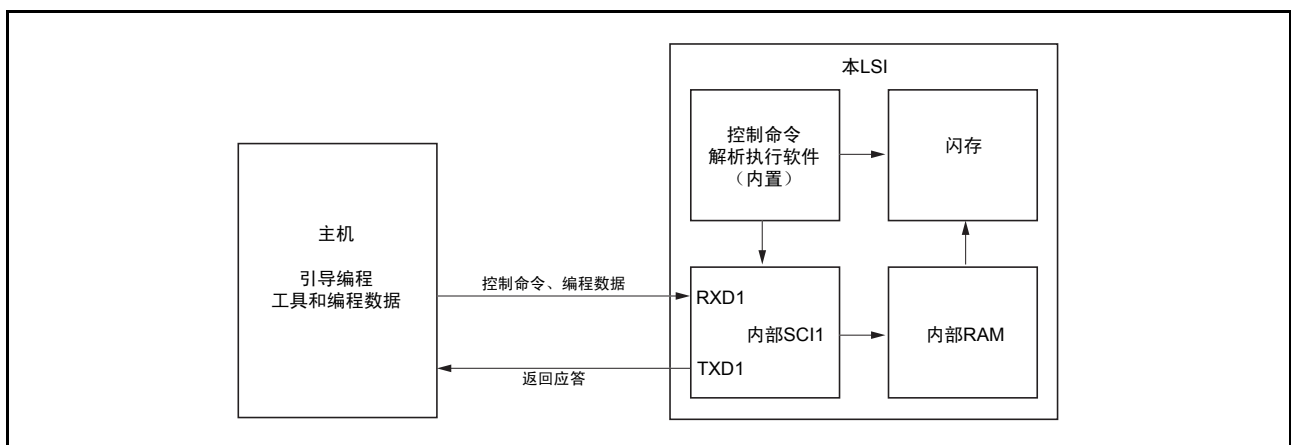


图 17.6 引导模式时的系统构成图

(1) 主机的 SCI 接口设定

如果启动引导模式，本 LSI 就测量主机连续发送来的异步 SCI 通信数据（H'00）的 Low 期间。必须将此时的 SCI 发送 / 接收格式设定为“8 位数据、1 个停止位和无奇偶校验”。本 LSI 根据测量的 Low 期间计算主机的发送位速率，并将位调整结束信号（1 字节的 H'00）发送给主机，主机必须在确认已正常接收到此调整结束信号（H'00）后，向本 LSI 发送 1 字节的 H'55。如果不能正常接收，就必须重新启动（复位）引导模式，进行上述操作。根据主机发送的位速率和本 LSI 的系统时钟的频率，主机和本 LSI 的位速率会产生误差，所以为了使 SCI 正常运行，必须将主机的传送位速率设定为 9,600bps 或者 19,200bps。

主机的传送位速率和本 LSI 的位速率能自动匹配的系统时钟的频率如表 17.7 所示，必须在此系统时钟的范围内启动引导模式。另外，在引导模式中，不支持各内部时钟的分频比为 X1/3 倍的设定。

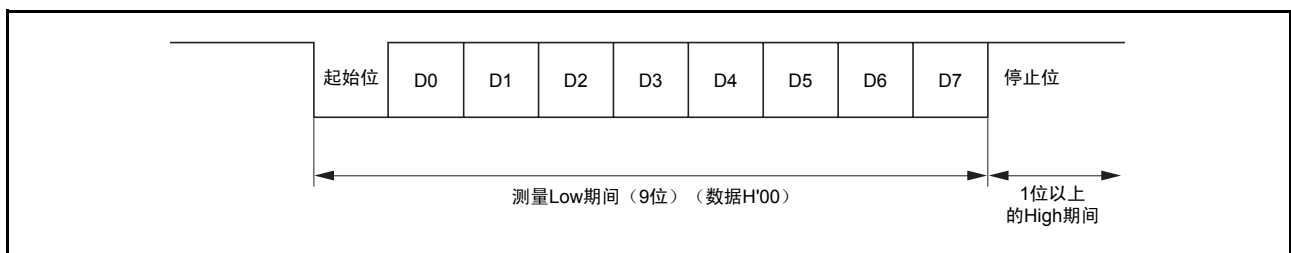


图 17.7 位速率的自动匹配

表 17.7 本 LSI 能自动匹配的系统时钟的频率

主机的位速率	本 LSI 的位速率能自动匹配的外围时钟 (P ϕ) 频率
9,600bps	20 ~ 25MHz
19,200bps	20 ~ 25MHz

【注】 在引导模式中，不支持各内部时钟的分频比为 X1/3 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 17.8 所示，有关引导模式的详细内容请参照“17.8.1 引导模式的标准串行通信接口规格”。

1. 位速率的匹配

在启动引导模式后进行主机和 SCI 接口的位速率匹配。

2. 等待查询选择命令

将查询用户 MAT 容量、用户 MAT 结构、MAT 起始地址和支持情况等时所需要的信息发送到主机。

3. 自动擦除全部的用户 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。

4. 等待编程 / 擦除命令

- 如果接收到“编程选择命令”，就转移到编程数据的等待状态。必须在编程命令后面连续发送编程的起始地址和编程数据，当编程结束时，必须将编程的起始地址设定为 H'FFFFFFF 后发送。据此从编程数据的等待状态返回到编程 / 擦除命令的等待状态。
- 如果接收到“擦除选择命令”，就转移到擦除块数据的等待状态。必须在擦除命令后面连续发送擦除块号，当擦除结束时，必须将擦除块号设定为 H'FF 后发送。据此从擦除块数据的等待状态返回到编程 / 擦除命令的等待状态。另外，在引导模式中，当编程后不需启动复位而只要改写特定块时，必须进行擦除。当通过 1 次操作就能完成编程时，因为在转移到编程 / 擦除 / 其他命令的等待状态之前已将全部块擦除，所以不需要此擦除操作。
- 除编程 / 擦除以外，还有用户 MAT 的校验和命令、用户 MAT 的空白检查（擦除检查）命令、用户 MAT 的存储器读取命令以及当前状态的信息取得命令。

必须注意：用户 MAT 的存储器读取只能在自动擦除全部的用户 MAT 后，读已被编程的数据。

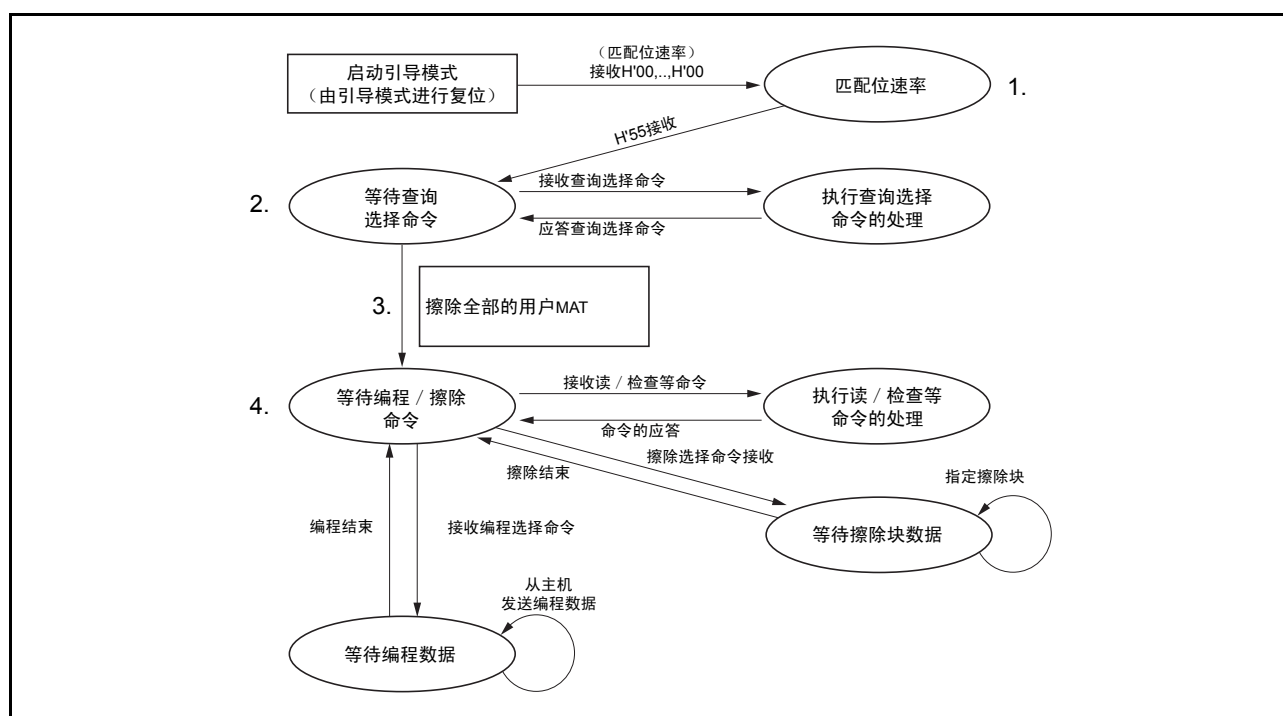


图 17.8 引导模式状态转移的概略图

17.5.2 用户编程模式

能在用户编程模式中编程 / 擦除用户 MAT。

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 17.9 所示。

另外，因为在编程 / 擦除处理中闪存内部加有高电压，所以不能在编程 / 擦除处理中进行复位，否则有可能损伤甚至破坏闪存。如果错误地进行了复位，必须在长于通常的 100 μ s 复位输入期间后进行复位解除。

有关编程步骤和擦除步骤，请分别参照后述的“(2) 用户编程模式的编程步骤”和“(3) 用户编程模式的擦除步骤”。

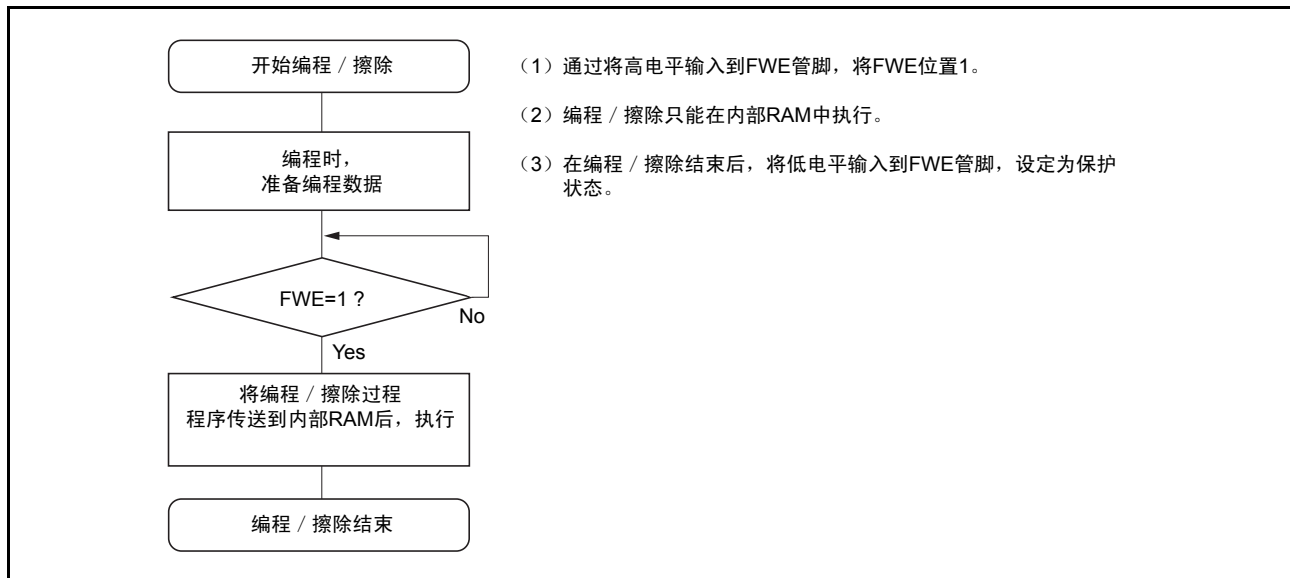


图 17.9 编程 / 擦除的概略流程

(1) 进行编程 / 擦除时的内部 RAM 的地址映像

必须在内部 RAM 中执行用户建立的下载请求、编程 / 擦除的步骤和结果判断等一部分过程程序。另外，由于被下载的内部程序全部存放在内部 RAM 中，所以为了使这些程序不重叠，必须注意内部 RAM 的区域管理。

被下载的程序区如图 17.10 所示。

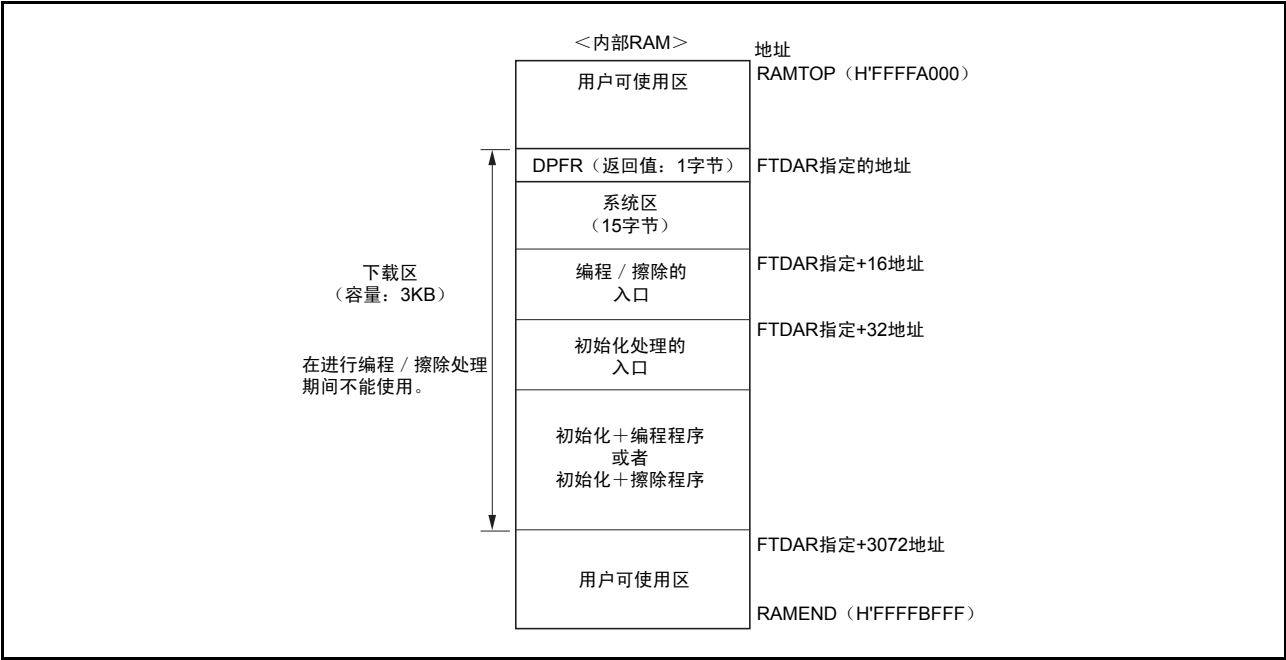


图 17.10 下载后的内部 RAM 映像

(2) 用户编程模式的编程步骤

下载、初始化和编程的步骤如图 17.11 所示。

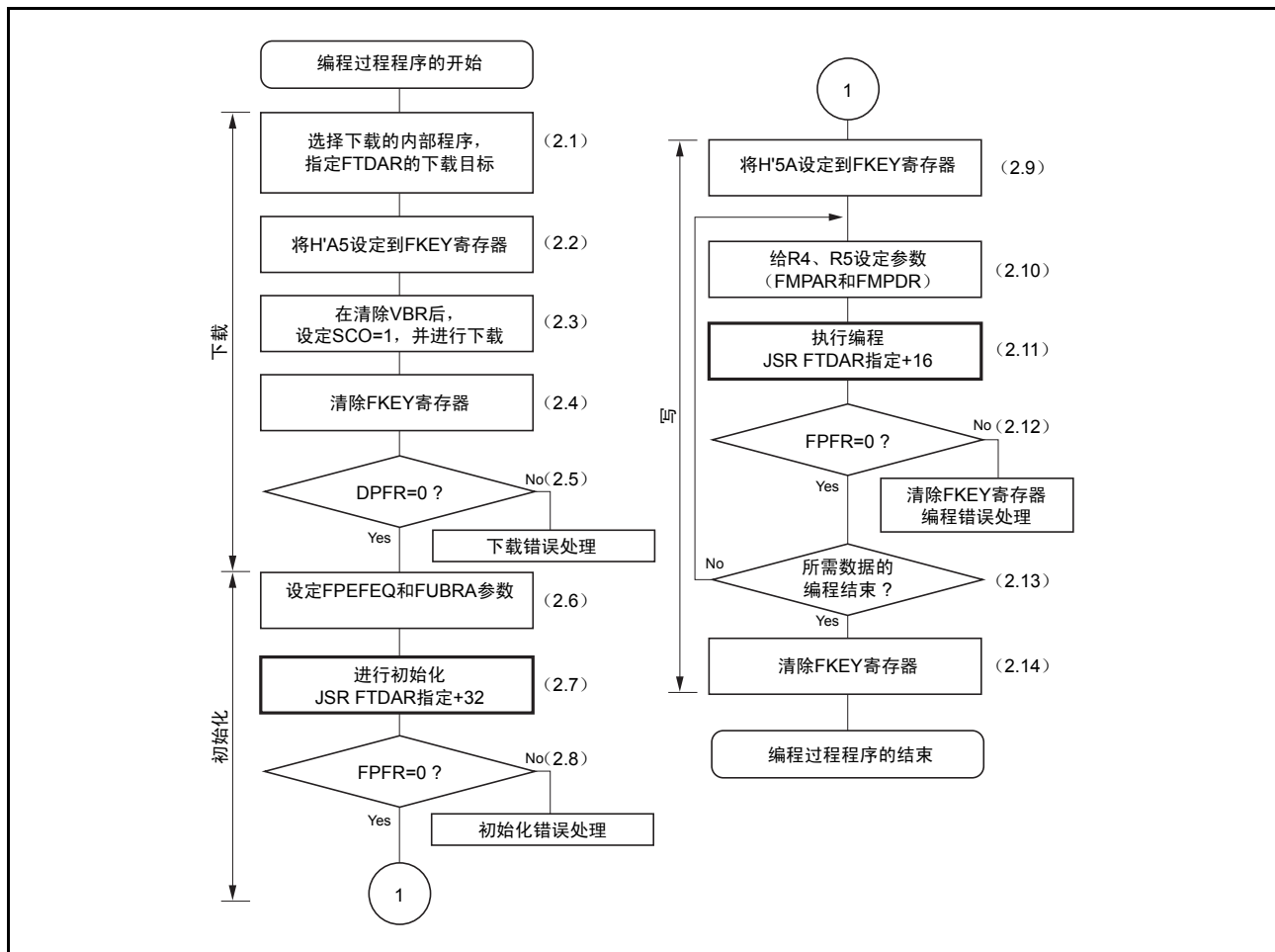


图 17.11 编程步骤

在此说明编程步骤的详细内容。必须在编程对象的闪存以外的区域执行过程程序，尤其是为了下载而将 FCCS 寄存器的 SCO 位设定为 1 的部分，必须在内部 RAM 中执行。另外，通过频率控制寄存器（FRQCR）设定的内部时钟（I ϕ ）、总线时钟（B ϕ ）、外围时钟（P ϕ ）的分频比都为 1/4（初始值）。

以下说明的前提是：已擦除用户 MAT 中的编程对象区，并且编程数据也已准备在连续区。如果还没有擦除，必须在编程前擦除。

在 1 次编程处理中进行 128 字节的编程。当超过 128 字节时，以 128 字节为单位更新编程目标地址 / 编程数据的参数，反复进行编程。

当未满 128 字节时，需要填写无效数据补齐 128 字节。如果填写的无效数据为 H'FF，就能缩短编程的处理时间。

(2.1) 选择下载的内部程序以及指定下载的目标地址。

如果将 FPCS 寄存器的 PPVS 位设定为 1，就选择编程程序。

不能选择多个编程 / 擦除程序，否则就不下载而由 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) 将 H'A5 写到 FKEY 寄存器。

如果为了保护而未将 H'A5 写到 FKEY 寄存器，就不能给下载请求的 SCO 位写 1。

(2.3) 设定 VBR 寄存器并给 FCCS 寄存器的 SCO 位写 1，进行下载。

在设定 SCO 位之前，必须将 VBR 寄存器清除为 H'84000000。

为了给 SCO 位写 1，需要满足以下的全部条件：

1. 已将 H'A5 写到 FKEY 寄存器。
2. 已在内部 RAM 中进行了 SCO 位的写操作。

因为当 SCO 位为 1 时就自动开始下载，当返回到用户的过程程序时 SCO 位就被清 0 (SCO=0)，所以在用户过程程序中不能确认 SCO=1。

因为只能通过 DPFR 参数的返回值确认下载结果，所以在 SCO=1 之前，必须将 FTDAR（作为 DPFR 参数的）所指定的内部 RAM 的起始 1 字节设定为返回值以外的值（H'FF 等），以免发生误判断。

在进行下载时，因为单片机的内部处理进行如下所示的伴有存储体转换的特殊中断处理，所以需要将 VBR 设定为 H'84000000。另外，必须在设定 SCO=1 的指令后面立刻执行 4 条 NOP 指令。

1. 将用户 MAT 空间转换到内部程序保存区。
2. 在检查下载程序的选择条件和 FTDAR 的指定地址等后，对 FTDAR 指定的内部 RAM 进行传送处理。
3. 将 FPCS 寄存器、FECS 寄存器和 FCCS 寄存器的 SCO 位清 0。
4. 给 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。

有关下载中的注意事项如下所述：

在下载处理中 CPU 的通用寄存器的值被保存。

不能在下载处理中发生中断。有关下载和中断请参照“17.7.1 执行编程 / 擦除过程中的中断”。

因为最多使用 128 字节的堆栈区，所以在使 SCO=1 之前，至少必须确保 128 字节的堆栈区。

(2.4) 将 FKEY 寄存器清除为 H'00，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 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 参数的值和下载前的设定值（H'FF 等）相同，FTDAR 的下载目标地址的设定可能异常，所以必须确认 FTDAR 的 bit7（TDER 位）。
3. 如果 DPFR 参数的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分支目标进行初始化。

1. 给 FPEFEQ 参数（通用寄存器：R4）设定当前的 CPU 时钟频率。FPEFEQ 参数的可设定范围请参照“21.3.1 时钟时序”。

如果将频率设定在此范围以外，就由初始化程序的 FPFR 参数报告错误并且不进行初始化。频率的设定方法请参照“17.4.3 编程 / 擦除的接口参数”的“闪存编程 / 擦除的频率控制参数（FPEFEQ：CPU 的通用寄存器 R4）”的说明。

2. 给 FUBRA 参数（通用寄存器：R5）设定用户分支目标的起始地址。

当不需要用户分支处理时，必须给 FUBRA 置 0。

当进行用户分支时，必须在编程对象的闪存以外的区域执行转移目标，并且不能设定到下载内部程序的区域。

必须使用 RTS 指令从用户分支处理返回到编程处理。

请参照“17.4.3 编程 / 擦除的接口参数”的“闪存用户分支地址的设定参数（FUBRA：CPU 的通用寄存器 R5）”的说明。

（2.7）执行初始化

在下载编程程序时，初始化程序一起被下载到内部 RAM。因为在从 FTDAR 设定的下载起始地址 +32 字节开始的区域中有初始化程序的入口点，所以必须通过以下方法调用子程序进行初始化。

MOV.L	#DLTOP+32, R1	;	将入口地址设定到 R1
JSR	@R1	;	调用初始化程序
NOP			

1. 在初始化程序中保存 R0 以外的通用寄存器。
2. R0 是 FPFR 参数的返回值。
3. 因为在初始化程序中使用堆栈区，所以必须预先在 RAM 中至少确保 128 字节的堆栈区。
4. 能在执行初始化程序中接受中断，但是不能破坏内部 RAM 中的程序保存区、堆栈区和寄存器的值。

（2.8）判断初始化程序的返回值 FPFR（通用寄存器 R0）。

（2.9）必须给 FKEY 寄存器设定 H'5A 以便能对用户 MAT 进行编程。

（2.10）设定编程所需要的参数。

将用户 MAT 的编程目标起始地址（FMPAR）和编程数据保存区的起始地址（FMPDR）分别设定给通用寄存器 R5 和 R4。

1. FMPAR 的设定

因为 FMPAR 指定编程起始地址，所以当指定用户 MAT 区以外的地址时，即使执行编程程序也不进行编程，而由返回值参数 FPFR 报告错误。另外，因为是以 128 字节为单位，所以低 8 位（MOA7 ~ MOA0）需要为 H'00 或者 H'80 的 128 字节的边界。

2. FMPDR 的设定

当编程数据的保存目标在闪存中时，即使执行编程执行程序也不进行编程，而由 FPFR 参数报告错误。此时必须暂时先将编程数据传送到内部 RAM，然后进行编程。

（2.11）执行编程处理

因为在从 FTDAR 指定的下载目标起始地址 +16 字节开始的区域中有编程程序的入口点，所以必须通过以下方法调用子程序进行编程处理。

MOV.L	#DLTOP+16, R1	;	将入口地址设定到 R1
JSR	@R1	;	调用编程程序
NOP			

1. 在编程程序中保存 R0 以外的通用寄存器。
2. R0 是 FPFR 参数的返回值。
3. 因为在编程程序中使用堆栈区，所以必须预先在 RAM 中至少确保 128 字节的堆栈区。

(2.12) 判断编程程序的返回值 FPFR（通用寄存器 R0）。

(2.13) 判断所需数据的编程是否已结束。

当编程超过 128 字节的数据时，以 128 字节为单位更新 FMPAR、FMPDR 的设定，然后重复上述 (2.10) ~ (2.13) 的处理。必须正确进行编程目标地址的 128 字节的递增和编程数据指针的更新。如果对已被编程的地址进行重复编程，不仅会引起编程错误，还会对闪存造成损伤。

(2.14) 必须在编程结束后清除 FKEY 寄存器并加软件保护。

如果在结束对用户 MAT 的编程后立刻通过加电复位重新启动，必须设定长于通常的 100 μ s 复位期间 (RES=0 的期间)。

(3) 用户编程模式的擦除步骤

下载、初始化和擦除的步骤如图 17.12 所示。

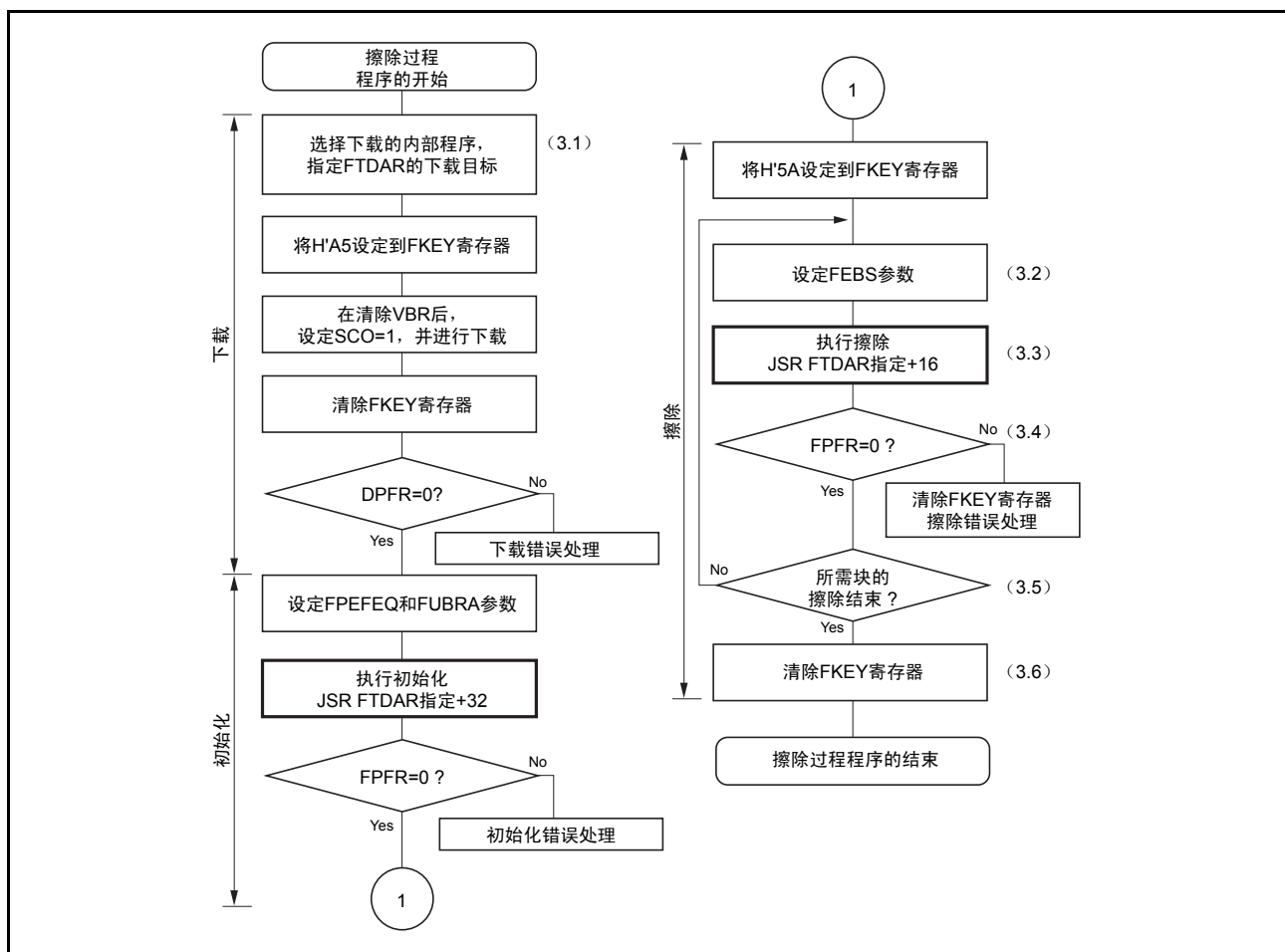


图 17.12 擦除步骤

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 位设定为 1 的部分，必须在内部 RAM 中执行。另外，通过频率控制寄存器（FRQCR）设定的内部时钟（I ϕ ）、总线时钟（B ϕ ）、外围时钟（P ϕ ）的分频比都为 1/4（初始值）。

有关被下载的内部程序的区域，请参照图 17.10 的下载后的内部 RAM 映像。

在 1 次擦除处理中进行 1 个分割块的擦除，有关块分割请参照图 17.4。当进行 2 块以上的擦除时，先更新擦除块号再重复擦除。

（3.1）选择下载的内部程序以及指定下载的目标地址。

将 FECS 寄存器的 EPVB 位设定为 1。

不能选择多个编程 / 擦除程序，否则就不下载而由 DPFR 参数的源选择错误检测位（SS）报告下载错误。

通过 FTDAR 寄存器指定下载目标的起始地址。

设定 FKEY 寄存器以后的下载和初始化等过程和编程步骤相同，请参照“17.5.2 (2) 用户编程模式的编程步骤”。

设定擦除程序参数以后的步骤如下所示：

（3.2）设定擦除时所需要的 FEBS 参数。

将用户 MAT 的擦除块号设定到闪存擦除块的选择参数 FEBS（通用寄存器 R4）。如果设定用户 MAT 分割块号以外的值，即使执行擦除处理程序也不擦除，而由返回值参数 FPFR 报告错误。

（3.3）执行擦除处理

和编程相同，因为在从 FTDAR 指定的下载目标起始地址 +16 字节开始的区有擦除程序的入口点，所以必须通过以下方法调用子程序进行擦除处理。

MOV.L	#DLTOP+16, R1	;	将入口地址设定到 R1
JSR	@R1	;	调用擦除程序
NOP			

1. 在擦除程序中保存 R0 以外的通用寄存器。

2. R0 是 FPFR 参数的返回值。

3. 因为在擦除程序中使用堆栈区，所以必须预先在 RAM 中至少确保 128 字节的堆栈区。

（3.4）判断擦除程序的返回值 FPFR（通用寄存器 R0）。

（3.5）判断所需块的擦除是否已结束。

当擦除多个块时，更新 FEBS 参数的设定，然后重复上述（3.2）～（3.5）的处理。能对已擦除的块进行擦除。

（3.6）必须在结束擦除后清除 FKEY 寄存器并加软件保护。

如果在结束用户 MAT 的擦除后立刻通过加电复位重新启动，必须设定长于通常的 100 μ s 通常复位期间（RES=0 的期间）。

17.6 保护

闪存的编程 / 擦除保护有硬件保护、软件保护和错误保护共 3 种。

17.6.1 硬件保护

硬件保护是指强制禁止或者强制中断闪存编程 / 擦除的状态。能下载内部程序并进行初始化，但是即使启动编程 / 擦除程序也不能进行用户 MAT 的编程 / 擦除，而由 FPFR 参数报告编程 / 擦除的错误。

表 17.8 硬件保护

项目	说 明	受保护的功能	
		下载	编程和擦除
FWE 管脚的保护	当给 FWE 管脚输入 Low 电平时，就清除 FCCS 寄存器的 FWE 位，变为编程 / 擦除的保护状态。	—	○
复位和待机的保护	- 在加电复位（包括由 WDT 产生的加电复位）和待机时，对编程 / 擦除的接口寄存器进行初始化，变为编程 / 擦除的保护状态。 - 通过 $\overline{\text{RES}}$ 管脚进行复位时，如果不将 $\overline{\text{RES}}$ 管脚的 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 $\overline{\text{RES}}$ 管脚保持为 Low 电平。不能保证编程 / 擦除运行中的闪存值。此时，必须在擦除后重新进行编程。	○	○

17.6.2 软件保护

软件保护有内部编程 / 擦除程序下载后的保护和通过键代码的保护。

表 17.9 软件保护

项目	说 明	受保护的功能	
		下载	编程和擦除
SCO 位的保护	因将 FCCS 寄存器的 SCO 位清 0 而不能下载编程 / 擦除的程序，所以变为编程 / 擦除的保护状态。	○	○
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。对于下载和编程 / 擦除，需要设定不同的键代码。	○	○

17.6.3 错误保护

错误保护是在闪存的编程 / 擦除中因检测到单片机失控或者没有按规定的编程 / 擦除步骤运行而强制中断编程 / 擦除运行的保护。通过中断编程 / 擦除运行能防止因重复编程或者重复擦除而造成闪存的损伤。

如果在闪存的编程 / 擦除中单片机运行异常，FCCS 寄存器的 FLER 位就被置 1，并转移到错误保护状态，中断编程 / 擦除。

FLER 位的置位条件如下所示：

1. 在编程 / 擦除中已读取闪存的该存储体区时（包括读向量和取指令）
2. 在编程 / 擦除中已执行 SLEEP 指令时（包括软件待机）

只在加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 通常复位输入期间后解除复位。因为在编程 / 擦除中闪存加有高电压，有可能在向错误保护状态转移时不能完全去除外加电压，所以需要通过延长复位期间去除外加电压，减少对闪存的损伤。

错误保护状态的状态转移如图 17.13 所示。

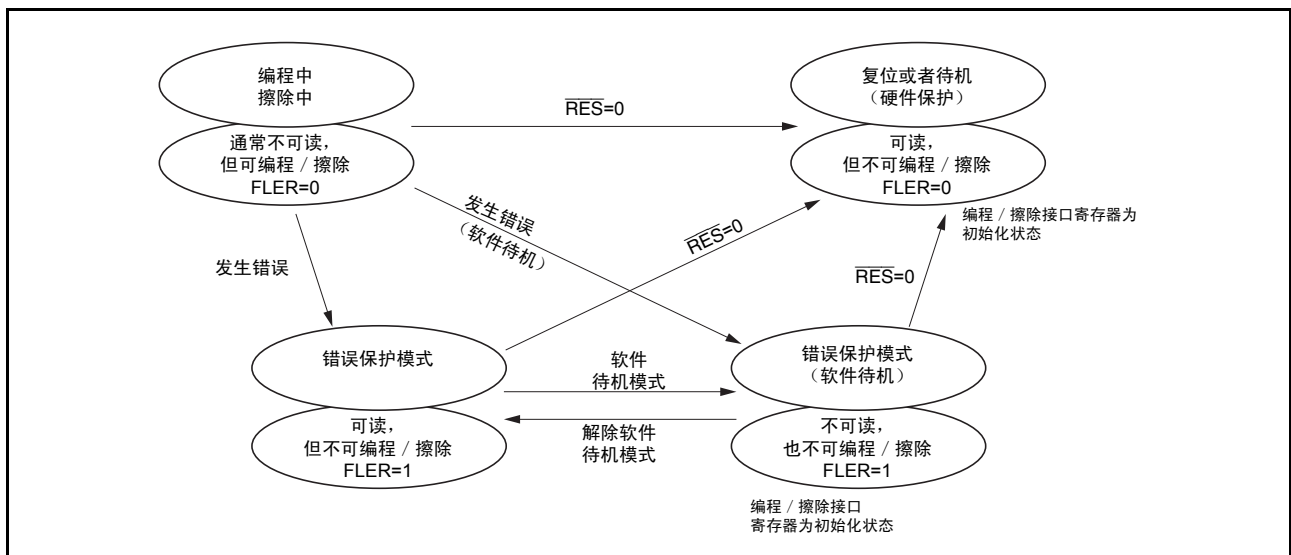


图 17.13 错误保护状态的状态转移图

17.7 使用注意事项

17.7.1 执行编程 / 擦除过程中的中断

(1) 执行内部程序的下载

(1.1) VBR 的变更

在下载内部程序前，需要将 VBR 寄存器置为 H'84000000。在设定 H'84000000 以外的值而使用 VBR 的情况下，如果将 VBR 设定为 H'84000000，中断向量表就变为用户 MAT。

另外，如果变更 VBR 和发生中断出现竞争，就有可能因参照 VBR 变更前或变更后的向量表而发生问题。

因此，在有可能发生和中断竞争的情况下，还必须在用户 MAT 的起始部分准备 VBR=H'84000000（初始值）时的参照向量表。

(1.2) SCO 下载请求和中断请求

在将 FCCS 寄存器的 SCO 位设定为 1 后下载内部的编程 / 擦除程序时，会产生伴有 MAT 转换的特殊中断。以下说明 SCO 下载请求和中断请求出现竞争时的运行：

1. SCO 下载请求和中断请求的竞争

将 FCCS 寄存器的 SCO 位设定为 1 的指令执行和接受中断的竞争时序如图 17.14 所示。

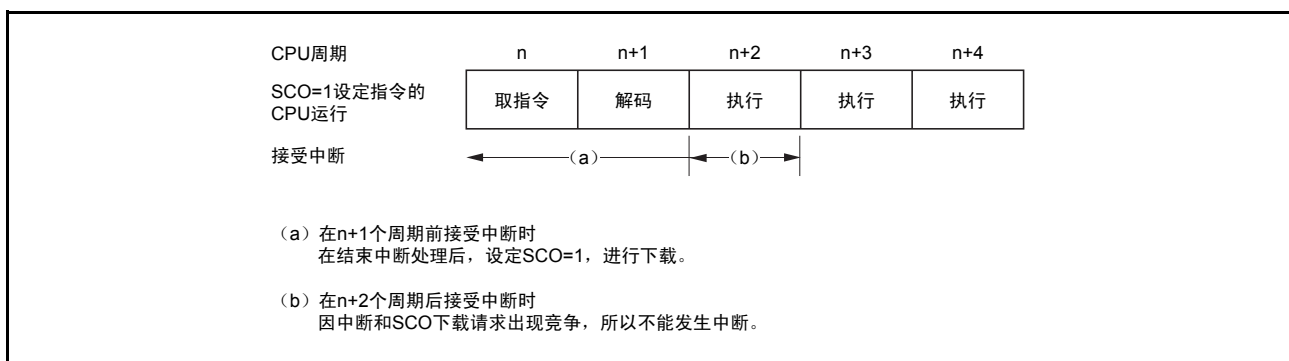


图 17.14 SCO 下载请求和中断请求的竞争时序

2. 在下载中发生的中断请求

不能在进行 SCO 下载中发生中断。

(2) 编程 / 擦除处理中的中断

能实时处理在下载内部程序的编程 / 擦除中发生的中断，但是有以下的限制和注意事项：

- 对于正在编程 / 擦除中的闪存，不能存取用户 MAT。必须将中断向量表和中断处理程序准备在内部 RAM 中。即使在中断处理程序中也不能存取正在编程 / 擦除中的闪存。如果读闪存，就不能保证读取的值。另外，如果在编程 / 擦除中存取闪存的该存储体，就转移到错误保护状态，中断编程 / 擦除处理。当存取该存储器以外的区域时，不转移到错误保护，但是不能保证读取的值。
- 不能更改 FMPDR 参数指定的编程数据。在通过中断处理准备编程数据时，必须先将数据准备到其他区域，然后在确认编程已结束后转移到 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相关寄存器和已下载的内部程序区。另外，在中断处理中，不能多重进行 SCO 请求的内部程序下载或者编程 / 擦除。
- 必须在中断处理程序的起始位置保存 CPU 的寄存器，并且在返回前恢复这些寄存器。
- 如果在中断处理程序中转移到睡眠状态或者软件待机状态，就进入错误保护状态，中断编程 / 擦除。另外，当转移到复位状态时，为了减少对闪存的损伤，必须在 100μs 以上的复位状态后解除复位。

17.7.2 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 8ms。

(2) 用户分支处理的间隔

执行用户分支处理的间隔因编程 / 擦除和处理阶段而不同。CPU 的时钟频率为 50MHz 时的最大启动间隔如表 17.10 所示。

表 17.10 用户分支处理的启动间隔

	最大间隔
编程处理	约 4ms
擦除处理	约 25ms

但是，在 CPU 时钟以 50MHz 运行时的最初用户分支处理为止的时间最大值如表 17.11 所示。

表 17.11 用户分支处理时间

	最大
编程处理	约 4ms
擦除处理	约 25ms

(3) 中断的忽视状态

在以下的模式或者期间中，即使发生中断也被忽视，不执行也不保存中断源。

- 引导模式运行中
- 编程器模式运行中

(4) 和以前的 F-ZTAT SH 单片机的编程 / 擦除程序的兼容性

不支持由 SCO 传送请求进行的内部程序的下载方式，本 LSI 不能运行以前的 F-ZTAT SH 单片机使用的闪存编程 / 擦除程序。

必须在下载内部程序后进行本 LSI 的闪存编程 / 擦除。

(5) 使用 WDT 对失控等的监视

和以前的 F-ZTAT SH 单片机不同，在通过下载的内部程序进行编程 / 擦除中，不实施由 WDT 对失控等的对策。

根据需要，必须使用考虑了编程 / 擦除执行时间的 WDT 使用对策（使用用户分支程序或者定期的定时器中断等）。

17.8 附录

17.8.1 引导模式的标准串行通信接口规格

在引导模式中启动的引导程序使用主机和 LSI 内部 SCI 进行接收和发送。主机和引导程序的串行通信接口规格如下所示：

●状态

引导程序有 3 个状态：

1. 位速率匹配状态

这是使主机和接收 / 发送的位速率进行匹配的状态。当启动引导模式时，引导程序就被启动，进入位速率匹配状态，接收主机的命令，并进行位速率的匹配。当匹配结束时，就转移到查询选择状态。

2. 查询选择状态

这是应答主机查询命令的状态。在此状态中选择器件、时钟模式和位速率。当选择结束时，通过编程 / 擦除状态转移命令转移到编程 / 擦除状态。在转移到编程 / 擦除状态前，引导程序将擦除程序传送到 RAM，并擦除用户 MAT。

3. 编程 / 擦除状态

这是进行编程 / 擦除的状态。按照主机的命令，将编程 / 擦除程序传送到 RAM，并进行编程 / 擦除。通过命令进行校验和及空白检查。

引导程序的处理流程如图 17.15 所示。

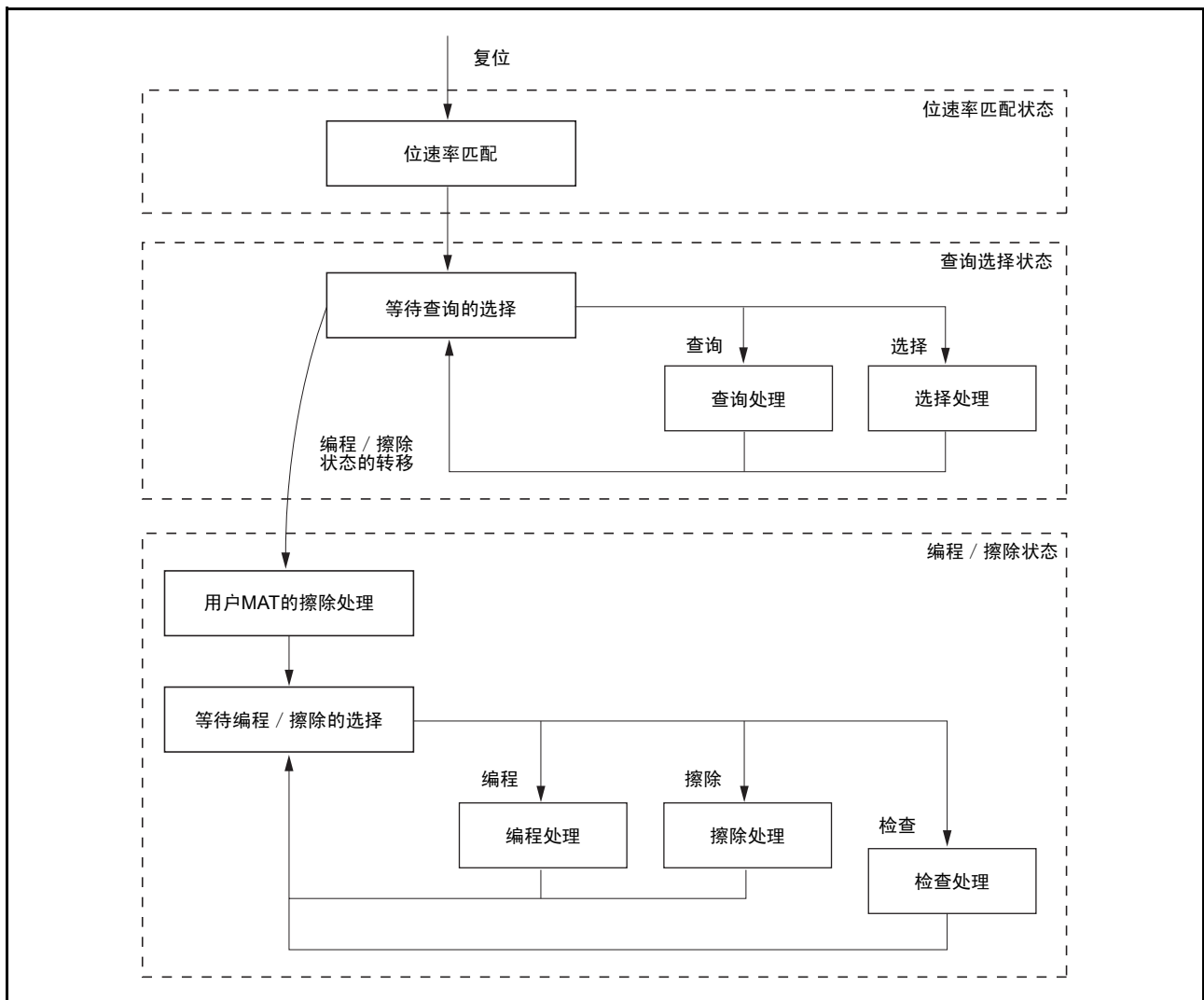


图 17.15 引导程序的处理流程

●位速率匹配状态

位速率匹配测量主机发送的 H'00 的低电平区间，计算位速率。能通过新的位速率选择命令更改位速率。当位速率匹配结束时，引导程序转移到查询选择状态。位速率匹配的顺序如图 17.16 所示。



图 17.16 位速率匹配的顺序

●通信协议

位速率匹配结束后的主机和引导程序的串行通信协议如下：

1. 个字符命令或者 1 个字符应答

因为命令或者应答只有 1 个字符，所以有查询和正常结束的 ACK。

2. n 个字符命令或者 n 个字符应答

因为命令和应答需要 n 字节的数据，所以有选择的命令和对应查询的应答。

因为对编程数据另外规定数据长，所以省略数据长。

3. 错误应答

这是对命令的错误应答，错误应答和错误码为 2 字节。

4. 128 字节的编程

这是没有长度的命令，能通过编程长度查询的应答，得知数据的长度。

5. 存储器读取的应答

这是包含 4 字节长度信息的应答。

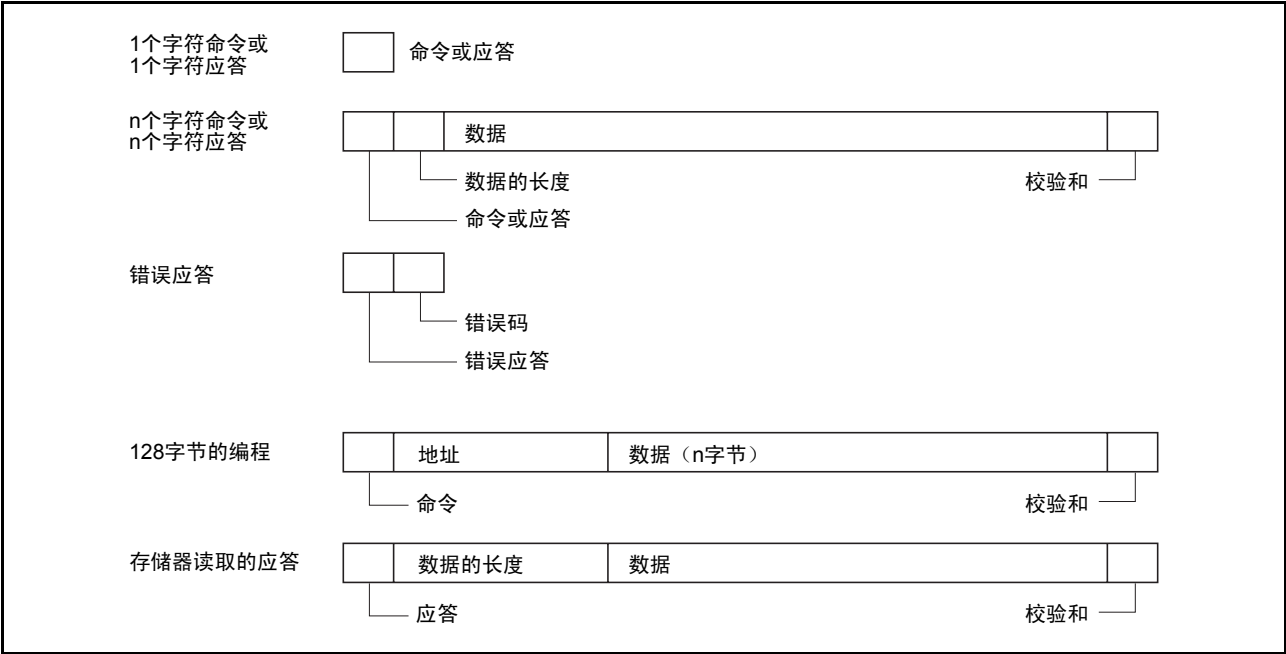


图 17.17 通信协议格式

- 命令（1 字节）：查询、选择、编程、擦除、检查等命令
- 应答（1 字节）：对查询的应答
- 长度（1 字节或者 2 字节）：除命令 / 查询、长度、校验和以外的接收和发送数据的长度
- 数据（n 字节）：命令和应答的详细数据
- 校验和（1 字节）：命令到 SUM 的总和，并且使其结果为 H'00
- 错误应答（1 字节）：对指令的错误应答
- 错误码（1 字节）：发生的错误种类
- 地址（4 字节）：编程地址
- 数据（n 字节）：编程数据。n 能从编程长度查询命令的应答得知
- 数据的长度（4 字节）：对存储器读取的 4 字节应答

● 查询选择状态

在查询选择状态中，引导程序对主机的查询命令进行闪存 ROM 信息的应答，对选择的命令选择器件、时钟模式和位速率。

查询选择命令一览表如表 17.12 所示。

表 17.12 查询选择命令一览表

命令	命令名	功能
H'20	支持器件的查询	查询器件的代码和产品名
H'10	器件选择	选择器件代码
H'21	时钟模式查询	查询时钟模式数和各自的值
H'11	时钟模式选择	通知已选择的时钟模式
H'22	倍增比查询	查询倍增比或者分频比的种类数和各自的个数及其值
H'23	工作频率查询	查询主时钟和外围时钟的最小值和最大值
H'25	用户 MAT 信息查询	查询用户 MAT 的个数以及各自的起始地址和最终地址
H'26	擦除块信息查询	查询块数以及各自的起始地址和最终地址
H'27	编程长度查询	查询编程时的数据长
H'3F	新位速率选择	选择新位速率
H'40	编程 / 擦除状态转移	擦除用户 MAT，转移到编程 / 擦除状态
H'4F	引导程序状态查询	查询引导的处理状态

必须按器件的选择（H'10）、时钟模式的选择（H'11）、新位速率的选择（H'3F）的顺序从主机发送选择命令。当同一选择命令发送 2 次以上时，后发送的选择命令有效。

除引导程序状态查询（H'4F）以外，这些命令在接受编程 / 擦除状态转移（H'40）之前都有效，主机能从上述的命令中选择需要的命令进行查询。引导程序状态查询（H'4F）在接受编程 / 擦除状态转移（H'40）后也有效。

（1）支持器件的查询

对于支持器件的查询，引导程序应答可支持器件的器件代码和产品名。

命令

H'20

- 命令 [H'20]（1 字节）：支持器件的查询

应答	H'30	长度	器件数	
	字符数	器件代码		产品名
	...			
	SUM			

- 应答 [H'30]（1 字节）：对支持器件的查询的应答
- 长度（1 字节）：命令、长度、校验和除外的接收 / 发送数据的长度。在此为器件数、字符数、器件代码和产品名的合计长度
- 器件数（1 字节）：单片机内的引导程序所支持的品种数
- 字符数（1 字节）：器件代码和引导程序产品名的字符数
- 器件代码（4 字节）：支持的产品名的 ASCII 代码
- 产品名（n 字节）：引导程序产品型号（ASCII 代码）
- SUM（1 字节）：校验和
命令到 SUM 的总和，并使其结果为 H'00。

(2) 器件选择

对于器件选择，引导程序将支持器件设定到指定的支持器件，对此后的查询，应答已选择的器件信息。

命令	H'10	长度	器件代码	SUM

- 命令 [H'10] (1 字节)：器件选择
- 长度 (1 字节)：器件代码的字符数 (是固定值，为 2)
- 器件代码 (4 字节)：支持器件的查询时应答的器件代码 (ASCII 代码)
- SUM (1 字节)：校验和

应答	H'06

- 应答 [H'06] (1 字节)：对器件选择的应答
当指定的器件代码和支持的器件代码一致时返回 ACK

错误

应答	H'90	ERROR

- 错误应答 [H'90] (1 字节)：对器件选择的错误应答
- ERROR: (1 字节)：错误码
H'11: 校验和错误
H'21: 器件代码不一致错误

(3) 时钟模式查询

对于时钟模式查询，引导程序应答可选择的时钟模式。

指令	H'21

- 命令 [H'21] (1 字节)：时钟模式查询

应答	H'31	长度	模式	...	SUM

- 应答 [H'31] (1 字节)：对时钟模式查询的应答
- 长度 (1 字节)：模式数和模式的合计长度
- 模式 (1 字节)：可选择的时钟模式 (例：H'01 时钟模式 1)
- SUM (1 字节)：校验和

(4) 时钟模式选择

对于时钟模式选择，引导程序将时钟模式设定到指定的时钟模式，对此后的查询，应答已选择的时钟模式的信息。

必须在发送器件选择命令后，发送时钟模式选择命令。

命令	H'11	长度	模式	SUM

- 命令 [H'11] (1 字节)：时钟模式选择
- 长度 (1 字节)：模式的字符数 (是固定值，为 1)
- 模式 (1 字节)：时钟模式查询时应答的时钟模式
- SUM (1 字节)：校验和

应答	H'06

- 应答 [H'06] (1 字节)：对时钟模式选择的应答
当指定的时钟模式和可选择的时钟模式一致时返回 ACK

错误

应答	H'91	ERROR

- 错误应答 [H'91]（1 字节）：对时钟模式选择的错误应答
- ERROR:（1 字节）：错误码
 - H'11: 校验和错误
 - H'22: 时钟模式不一致错误

（5）倍增比查询

对于倍增比查询，引导程序应答可选择的倍增比或者分频比。

命令

H'22

- 命令 [H'22]（1 字节）：倍增比查询

应答	H'32	长度	频率种类数					
	倍增比数	倍增比	...					
	...							
	SUM							

- 应答 [H'32]（1 字节）：对倍增比查询的应答
- 长度（1 字节）：频率种类数、倍增比数和倍增比的合计长度
- 频率种类数（1 字节）：器件可选择的倍增比的种类数
（如果是主工作频率和外围模块工作频率 2 种，为 H'02）
- 倍增比数（1 字节）：各工作频率可选择的倍增比数
主模块和外围模块可选择的倍增比数
- 倍增比（1 字节）
倍增比：倍增的数值（例 4 倍增：H'04）
分频比：分频的数值，为负数（例 2 分频：H'FE[-2]）
倍增比的重复次数为倍增比数，倍增比数和倍增比的组合重复次数为频率种类数。
- SUM（1 字节）：校验和

（6）工作频率查询

对于工作频率查询，引导程序应答工作频率的种类数及其最小值和最大值。

命令

H'23

- 命令 [H'23]（1 字节）：工作频率查询

应答	H'33	长度	频率种类数
	工作频率最小值		工作频率最大值
	...		
	SUM		

- 应答 [H'33]（1 字节）：对工作频率查询的应答
- 长度（1 字节）：频率种类数、工作频率最小值和工作频率最大值的合计长度
- 频率种类数（1 字节）：器件所需要的工作频率的种类数
例如，在主工作频率和外围模块工作频率的情况下为 2
- 工作频率最小值（2 字节）：倍增或者分频时钟的最小值
工作频率的最小值和最大值是将频率（MHz）小数点 2 位为止的值放大 100 倍的值（例如，当频率为 20.00MHz 时，放大 100 倍后为 2000，设定为 H'07D0）。
- 工作频率最大值（2 字节）：倍增或者分频时钟的最大值
工作频率的最小值和最大值的数据重复次数为频率种类数

- SUM（1 字节）：校验和

（7）用户 MAT 信息查询

对于用户 MAT 信息查询，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 字节）：用户 MAT 信息查询

应答	H'35	长度	区域数	
	区域起始地址			区域最终地址
	...			
	SUM			

- 应答 [H'35]（1 字节）：对用户 MAT 信息查询的应答
- 长度（1 字节）：区域数、区域起始地址和区域最终地址的合计长度
- 区域数（1 字节）：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 字节）：区域的起始地址
- 区域最终地址（4 字节）：区域的最终地址
区域起始地址和区域最终地址的数据重复次数为区域数
- SUM（1 字节）：校验和

（8）擦除块信息查询

对于擦除块信息查询，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 字节）：擦除块信息查询

应答	H'36	长度	块数	
	块起始地址			块最终地址
	...			
	SUM			

- 应答 [H'36]（1 字节）：对擦除块信息查询的应答
- 长度（2 字节）：块数、块起始地址和块最终地址的合计长度
- 块数（1 字节）：闪存的擦除块数
- 块起始地址（4 字节）：块的起始地址
- 块最终地址（4 字节）：块的最终地址
块起始地址和块最终地址的数据重复次数为块数
- SUM（1 字节）：校验和

（9）编程长度查询

对于编程长度查询，引导程序应答编程数据的编程单位。

命令

H'27

- 命令 [H'27]（1 字节）：编程长度查询

应答	H'37	长度	编程长度	SUM

- 应答 [H'37]（1 字节）：对编程长度查询的应答

- 长度（1 字节）：编程单位长度的字符数（是固定值，为 2）
- 编程长度（2 字节）：编程单位的长度
以此长度接受编程数据
- SUM（1 字节）：校验和

（10）新位速率选择

对于新位速率选择，引导程序将选择更改被指定的位速率，对于确认应答新位速率。
必须在发送时钟模式选择命令后，发送新位速率选择命令。

命令	H'3F	长度	位速率	输入频率
	倍增比数	倍增比 1	倍增比 2	
	SUM			

- 命令 [H'3F]（1 字节）：新位速率选择
- 长度（1 字节）：位速率、输入频率、倍增比数和倍增比的合计长度
- 位速率（2 字节）：新位速率
为 1/100 的值（例如，当位速率为 19200bps 时为 192，设定为 H'00C0）
- 输入频率（2 字节）：输入到引导程序的时钟频率
为频率（MHz）小数点 2 位为止的值（例如，当时钟频率为 28.882MHz 时，将小数点 2 位为止的值放大 100 倍后为 2888，设定为 H'0B48）
- 倍增比数（1 字节）：器件可选择的倍增比数
通常是主工作频率和外围模块工作频率，为 2
- 倍增比 1（1 字节）：主工作频率的倍增比或者分频比
倍增比：倍增的数值（例 4 倍增：H'04）
分频比：分频的数值，为负数（例 2 分频：H'FE[-2]）
- 倍增比 2（1 字节）：外围工作频率的倍增比或者分频比
倍增比：倍增的数值（例 4 倍增：H'04）
分频比：分频的数值，为负数（例 2 分频：H'FE[-2]）
- SUM（1 字节）：校验和

应答

H'06

- 应答 [H'06]（1 字节）：对新位速率选择的应答
可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 字节）：对新位速率选择的错误应答
- ERROR：（1 字节）：错误码
 - H'11：校验和错误
 - H'24：不能选择位速率的错误
不能选择被指定的位速率
 - H'25：输入频率的错误
输入频率不在最小值和最大值的范围内
 - H'26：倍增比错误
倍增比不一致
 - H'27：工作频率错误
工作频率不在最小值和最大值的范围内

接收数据的检查方法如下所示：

1. 输入频率

检查接收的输入频率的值是否在输入频率的最小值和最大值的范围内（对于已选择器件的时钟模式），如果不在范围内，就为输入频率错误。

2. 倍增比

检查已接收的倍增比或者分频比的值是否和倍增比或者分频比（对于已选择器件的时钟模式）一致。如果不一致，就为倍增比错误。

3. 工作频率

由接收的输入频率以及倍增比或者分频比计算工作频率。输入频率是提供给 LSI 的频率，工作频率是 LSI 实际的工作频率，计算式如下所示：

工作频率 = 输入频率 × 倍增比

或者 工作频率 = 输入频率 ÷ 分频比

检查此计算的工作频率是否在工作频率的最小值和最大值的范围内（对于已选择器件的时钟模式），如果不在范围内，就为工作频率错误。

4. 位速率

由外围工作频率（ $P\phi$ ）和位速率（B）计算串行模式寄存器（SCSMR）的时钟选择（CKS）的值（n）、位速率寄存器（SCBRR）的值（N）以及误差。检查误差是否小于 4%，如果误差大于等于 4%，就为位速率选择错误。误差的计算如下：

$$\text{误差 (\%)} = \left\{ \left[\frac{P\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} \right] - 1 \right\} \times 100$$

如果能选择新位速率，就在应答 ACK 后将新位速率的值设定到寄存器。主机以新位速率发送 ACK，引导程序以新位速率应答。

确认 H'06

- 确认 [H'06]（1 字节）：新位速率的确认

应答 H'06

- 应答 [H'06]（1 字节）：对确认新位速率的应答
选择新位速率的顺序如图 17.18 所示。



图 17.18 选择新位速率的顺序

(11) 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，并擦除用户 MAT 的数据。当擦除结束时，应答 ACK，变为编程 / 擦除状态。

在发送编程选择命令的编程数据前，主机必须通过器件选择命令、时钟模式选择命令和新位速率选择命令，选择 LSI 的器件、时钟模式和新位速率，并将编程 / 擦除状态转移命令发送到引导程序。

命令

H'40

- 命令 [H'40] (1 字节)：编程 / 擦除状态转移

应答

H'06

- 应答 [H'06] (1 字节)：对编程 / 擦除状态转移的应答。在传送擦除程序后，当用户 MAT 正常擦除数据后返回 ACK

错误

应答

H'C0	H'51

- 错误应答 [H'C0] (1 字节)：对编程 / 擦除状态转移的错误应答
- 错误码 [H'51] (1 字节)：擦除错误
因发生错误而不能擦除

●命令错误

如果未定义命令、命令顺序不正确或者不能接受命令，就为命令错误。例如，器件选择前的时钟模式选择命令或者编程 / 擦除状态转移命令后的查询命令为命令错误。

错误

应答

H'80	H'xx

- 错误应答 [H'80] (1 字节)：命令错误
- 命令 [H'xx] (1 字节)：已接收的命令

●命令顺序

在查询选择状态下的命令顺序的例子如下：

1. 必须通过支持器件的查询 (H'20)，查询支持的器件。
2. 必须从被应答的器件信息中，进行器件选择 (H'10)。
3. 必须通过时钟模式查询 (H'21)，查询时钟模式。
4. 必须从被应答的时钟模式中，进行时钟模式选择 (H'11)。
5. 当器件选择和时钟模式选择结束时，必须通过倍增比查询 (H'22) 和工作频率查询 (H'23)，查询新位速率选择所需要的信息。
6. 必须按照倍增比和工作频率的信息，选择新位速率 (H'3F)。
7. 当器件选择和时钟模式选择结束时，必须通过用户 MAT 信息查询 (H'25)、擦除块信息查询 (H'26) 和编程长度查询 (H'27)，查询用户 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 / 擦除状态中, 引导程序通过编程选择命令进行编程方法的选择, 通过 128 字节编程命令进行数据的编程, 并且通过擦除选择和块擦除命令进行块的擦除。编程 / 擦除命令一览表如表 17.13 所示。

表 17.13 编程 / 擦除命令一览表

命令	命令名	功能
H'43	用户 MAT 编程选择	选择用户 MAT 的编程程序
H'50	128 字节编程	128 字节编程
H'48	擦除选择	选择擦除程序
H'58	块擦除	块数据的擦除
H'52	存储器读取	读存储器
H'4B	用户 MAT 的校验和	用户 MAT 的校验和
H'4D	用户 MAT 的空白检查	用户 MAT 的空白检查
H'4F	引导程序状态查询	查询引导的处理状态

●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。

其次, 发送 128 字节编程命令, 接在选择命令后面的 128 字节编程命令分别被解释为选择命令所指定的编程方式的编程数据。当编程超过 128 字节的数据时, 必须重复 128 字节的编程命令。如果要结束编程, 必须由主机发送地址为 H'FFFFFFF 的 128 字节编程命令。当编程结束时, 变为编程 / 擦除选择命令的等待状态。

当继续进行其他方式和其他 MAT 的编程时, 从编程选择命令开始。

编程选择命令和 128 字节编程命令的顺序如图 17.19 所示。



图 17.19 编程顺序

(1) 用户 MAT 编程选择

对于用户 MAT 编程选择，引导程序传送编程程序，并通过传送的编程程序对用户 MAT 进行编程。

命令

H'43

- 命令 [H'43] (1 字节)：用户 MAT 编程选择

应答

H'06

- 应答 [H'06] (1 字节)：对用户 MAT 编程选择的应答。当编程程序传送结束时返回 ACK

错误

应答

H'C3	ERROR

- 错误应答 [H'C3] (1 字节)：对用户程序编程选择的错误应答
- ERROR: (1 字节)：错误码
H'54: 选择处理错误 (发生传送错误, 处理未结束)

(2) 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 MAT 进行编程。

命令	H'50	编程地址						
	数据	...						
	...							
	SUM							

- 命令 [H'50] (1 字节)：128 字节编程
- 编程地址 (4 字节)：编程的起始地址
必须指定以 128 字节为边界的地址。
例) H'00、H'01、H'00、H'00: H'00010000
- 编程数据 (n 字节)：编程数据
编程数据的长度是“编程长度查询”应答的长度
- SUM (1 字节)：校验和

应答

H'06

- 应答 [H'06] (1 字节)：对 128 字节编程的应答
当编程结束时返回 ACK

错误

应答

H'D0	ERROR

- 错误应答 [H'D0] (1 字节)：对 128 字节编程的错误应答
- ERROR: (1 字节)：错误码
H'11 : 校验和错误
H'2A : 地址错误 (地址不在指定 MAT 的范围内)
H'53 : 编程错误 (发生编程错误, 不能编程)

必须按照数据编程长度指定边界地址。例如，当数据编程长度为 128 字节时，必须将地址的低位字节设定为 H'00 或者 H'80。

主机必须将 128 字节中没有编程数据的部分填写 H'FF，然后发送。

当结束编程处理时，必须发送地址 H'FFFFFFFF 的 128 字节编程命令。对地址 H'FFFFFFFF 的 128 字节编程命令，引导程序判断数据已结束，变为编程 / 擦除选择命令的等待状态。

命令

H'50	编程地址	SUM

- 命令 [H'50] (1 字节)：128 字节编程

- 编程地址（4 字节）：结束码（H'FF、H'FF、H'FF、H'FF）
- SUM（1 字节）：校验和

应答 H'06

- 应答 [H'06]（1 字节）：对 128 字节编程的应答
当编程处理结束时返回 ACK

错误

应答 H'D0 ERROR

- 错误应答 [H'D0]（1 字节）：对 128 字节编程的错误应答
- ERROR:（1 字节）：错误码
H'11: 校验和错误
H'53: 编程错误

●擦除

使用擦除选择命令和块擦除命令，进行擦除。

首先通过擦除选择命令，选择擦除，其次对块擦除命令所指定的块进行擦除。当有多块擦除块时，重复块擦除命令。当结束擦除处理时，必须从主机发送块号为 H'FF 的块擦除命令。当擦除结束时，变为编程 / 擦除选择命令的等待状态。

擦除选择命令和擦除数据的顺序如图 17.20 所示。

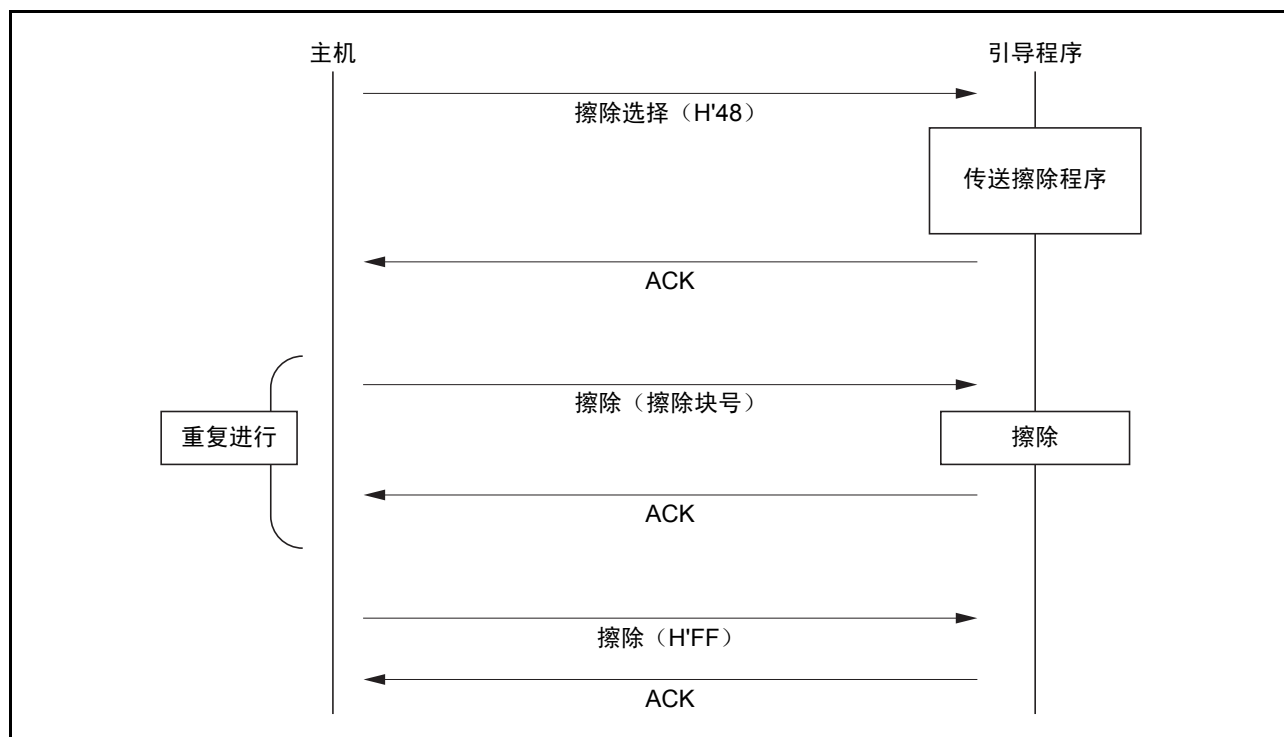


图 17.20 擦除顺序

(1) 擦除选择

对于擦除选择，引导程序传送擦除程序，并通过传送的擦除程序对用户 MAT 的数据进行擦除。

命令

H'48

- 命令 [H'48] (1 字节)：擦除选择

应答

H'06

- 应答 [H'06] (1 字节)：对擦除选择的应答
当擦除程序传送结束时返回 ACK

错误

应答

H'C8	ERROR

- 错误应答 [H'C8] (1 字节)：对擦除选择的错误应答
- ERROR: (1 字节)：错误码
H'54: 选择处理错误 (发生传送错误, 处理未结束)

(2) 块擦除

对于擦除，引导程序擦除被指定的用户 MAT 块。

命令

H'58	长度	块号码	SUM

- 命令 [H'58] (1 字节)：擦除
- 长度 (1 字节)：擦除块号的字符数 (是固定值, 为 1)
- 块号码 (1 字节)：擦除数据的擦除块号
- SUM (1 字节)：校验和

应答

H'06

- 应答 [H'06] (1 字节)：对擦除的应答
当擦除结束时返回 ACK

错误

应答

H'D8	ERROR

- 错误应答 [H'D8] (1 字节)：对擦除的错误应答
- ERROR: (1 字节)：错误码
H'11: 校验和错误
H'29: 块号错误
块号不正确
H'51: 擦除错误
在擦除中发生错误

对于块号 H'FF，引导程序结束擦除处理，变为选择命令的等待状态。

命令

H'58	长度	块号码	SUM

- 命令 [H'58] (1 字节)：擦除
- 长度 (1 字节)：擦除块号码的字符数 (是固定值, 为 1)
- 块号码 (1 字节)：H'FF (擦除处理的结束码)
- SUM (1 字节)：校验和

应答

H'06

- 应答 [H'06] (1 字节)：对擦除结束的应答 ACK

在由 H'FF 指定块号后重新进行擦除时，从擦除选择开始执行。

●存储器读取

对存储器读取，引导程序读取被指定的地址数据。

命令	H'52	长度	区域	读的起始地址
	读的起始地址	SUM		

- 命令 [H'52] (1 字节)：存储器读取
- 长度 (1 字节)：区域、读的起始地址和读的长度的合计长度 (是固定值，为 9)
- 区域 (1 字节)

H'01: 用户 MAT
当指定的区域不正确时为地址错误

- 读的起始地址 (4 字节)：读的起始地址
- 读的起始地址 (4 字节)：读数据的长度
- SUM (1 字节)：校验和

应答	H'52	读的起始地址						
	数据	...						
	SUM							

- 应答 [H'52] (1 字节)：对存储器读取的应答
- 读的起始地址 (4 字节)：读数据的长度
- 数据 (n 字节)：从读的起始地址开始，个数为读的长度的数据
- SUM (1 字节)：校验和

错误

应答	H'D2	ERROR

- 错误应答 [H'D2] (1 字节)：对存储器读的应答
- ERROR: (1 字节)：错误码
 - H'11: 校验和错误
 - H'2A: 地址错误
 - 读的起始地址不在 MAT 的范围内
 - H'2B: 长度错误
 - 读的起始地址超过 MAT 的范围，或者由读的起始地址和读的长度的最终地址不在 MAT 的范围内，或者读的起始地址为 0

●用户 MAT 的校验和

对于用户 MAT 的校验和，引导程序将用户 MAT 的数据相加，并应答其结果。

命令	H'4B

- 命令 [H'4B] (1 字节)：用户 MAT 的校验和

应答	H'5B	长度	MAT 的校验和	SUM

- 应答 [H'5B] (1 字节)：对用户 MAT 的校验和的应答
- 长度 (1 字节)：校验和数据的字符数 (是固定值，为 4)
- MAT 的校验和 (4 字节)：用户 MAT 的校验和值，以字节为单位相加
- SUM (1 字节)：校验和 (发送数据)

●用户 MAT 的空白检查

对于用户 MAT 的空白检查，引导程序检查用户 MAT 是否全部为空白，并应答其结果。

命令

H'4D

- 命令 [H'4D]（1 字节）：用户 MAT 的空白检查

应答

H'06

- 应答 [H'06]（1 字节）：对用户 MAT 的空白检查的应答。当区域全部为空白（H'FF）时返回 ACK

错误

应答

H'CD	H'52

- 错误应答 [H'CD]（1 字节）：对用户 MAT 的空白检查的错误应答
- 错误码 [H'52]（1 字节）：未擦除错误

●引导程序状态查询

对于引导程序状态查询，引导程序应答当前的状态和错误状态。此查询在查询选择状态和编程 / 擦除状态中都有效。

命令

H'4F

- 命令 [H'4F]（1 字节）：引导程序状态查询

应答

H'5F	长度	STATUS	ERROR	SUM

- 应答 [H'5F]（1 字节）：对引导程序状态查询的应答
- 长度（1 字节）：数据的字符数（是固定值，为 2）
- STATUS（1 字节）：标准引导程序的状态
请参照表 17.14。
- ERROR（1 字节）：错误状态
当 ERROR=0 时，正常
当 ERROR 不为 0 时，异常
请参照表 17.15。
- SUM（1 字节）：校验和

表 17.14 状态码

代码	内容
H'11	等待器件的选择
H'12	等待时钟模式的选择
H'13	等待位速率的选择
H'1F	等待编程 / 擦除的状态转移（位速率选择结束）
H'31	正在擦除用户 MAT
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 17.15 错误码

代码	内容
H'00	无错误
H'11	校验和错误
H'21	器件代码不一致错误
H'22	时钟模式不一致错误
H'24	位速率的选择错误
H'25	输入频率错误
H'26	倍增比错误
H'27	工作频率错误
H'29	块号错误
H'2A	地址错误
H'2B	数据长错误
H'51	擦除错误
H'52	未擦除错误
H'53	编程错误
H'54	选择处理错误
H'80	命令错误
H'FF	位速率匹配确认错误

17.8.2 过程程序或者编程数据的可保存区

编程 / 擦除过程程序和编程数据如果满足以下条件，也可在其它区域（如外部空间区等）实行。

1. 因为从 FTDAR 寄存器指定的内部 RAM 地址下载并执行内部编程 / 擦除程序，所以此区域不能使用。
2. 因为内部编程 / 擦除程序使用 128 字节以上的堆栈区，所以必须确保该区域。
3. 因为在将 SCO 位设定为 1 进行下载请求的处理中产生 MAT 转换，所以必须在内部 RAM 中进行。
4. 在开始编程 / 擦除前（在判断下载结果前）能存取闪存。必须在此之前将需要的过程程序、中断向量、中断处理程序 and 用户分支处理程序等传送到内部 RAM。
5. 因为在编程 / 擦除处理中不能存取闪存，所以通过已被下载到内部 RAM 中的程序来执行。启动此程序的过程程序、编程 / 擦除中的用户分支目标的用户程序、中断的向量表和中断处理程序的执行区也需要为闪存以外的内部 RAM。
6. 从编程 / 擦除结束后到清除 FKEY 寄存器的期间，禁止存取闪存。
在编程 / 擦除结束后立刻更改 LSI 模式进行复位时，必须设有 100μs 以上的复位期间（ $\overline{\text{RES}}=0$ 的期间）。
另外，在编程 / 擦除处理中禁止转移到复位状态，如果错误地进行了复位，就必须在长于通常的 100μs 复位期间后解除复位。
7. 如果编程处理的参数 FMPDR 所指示的编程数据保存区在闪存中，就判断为错误，所以必须暂时将编程数据传送到内部 RAM，并将 FMPDR 指示的地址设定在闪存空间以外。

按照各运行模式 / 处理内容的组合，编程数据保存区和可执行区如下表所示：

表 17.16 可执行的 MAT

处理	启动模式
	用户编程模式
编程	表 17.17 (1)
擦除	表 17.17 (2)

表 17.17 (1) 在用户编程模式中编程处理可使用的区域

	项目	可保存 / 执行的区域		选择的 MAT	
		内部 RAM	用户 MAT	用户 MAT	嵌入式程序 保存 MAT
编程 步骤 ↓	编程数据的保存区	○	X*	—	—
	下载内部程序的选择处理	○	○	○	
	对键寄存器写 H'A5 的处理	○	○	○	
	设定 FCCS 的 SCO=1 (下载)	○	X		○
	键寄存器的清除处理	○	○	○	
	下载结果的判断	○	○	○	
	下载的错误处理	○	○	○	
	初始化参数的设定处理	○	○	○	
	初始化的执行	○	X	○	
	初始化结果的判断	○	○	○	
	初始化的错误处理	○	○	○	
	中断处理程序	○	X	○	
	对键寄存器写 H'5A 的处理	○	○	○	
	编程参数的设定处理	○	X	○	
	编程的执行	○	X	○	
	编程结果的判断	○	X	○	
	编程的错误处理	○	X	○	
	键寄存器的清除处理	○	X	○	

【注】* 只要预先传送到内部 RAM，就可使用。

表 17.17 (2) 在用户编程模式中擦除处理可使用的区域

	项目	可保存 / 执行的区域		选择的 MAT	
		内部 RAM	用户 MAT	用户 MAT	嵌入式程序 保存 MAT
擦除步骤 ↓	下载内部程序的选择处理	○	○	○	
	对键寄存器写 H'5A 的处理	○	○	○	
	设定 FCCS 的 SCO=1 (下载)	○	X		○
	键寄存器的清除处理	○	○	○	
	下载结果的判断	○	○	○	
	下载的错误处理	○	○	○	
	初始化参数的设定处理	○	○	○	
	初始化的执行	○	X	○	
	初始化结果的判断	○	○	○	
	初始化的错误处理	○	○	○	
	中断处理程序	○	X	○	
	对键寄存器写 H'5A 的处理	○	○	○	
	擦除参数的设定处理	○	X	○	
	擦除的执行	○	X	○	
	擦除结果的判断	○	X	○	
	擦除的错误处理	○	X	○	
	键寄存器的清除处理	○	X	○	

17.9 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 128K 字节闪存内置型单片机芯片（F-ZTATxxxx）的 PROM 编程器。

第 18 章 RAM

本 LSI 内置高速静态 RAM。内部 RAM 通过 32 位数据总线（L 总线）连接到 CPU，能以 8、16 或者 32 位宽存取内部 RAM。根据各产品将内部 RAM 分配到图 18.1 所示的地址，能从 CPU（经过 L 总线）存取 RAM，从 L 总线（CPU）的存取为 1 个周期的存取。另外，在睡眠模式、软件待机模式、加电复位或者手动复位中保持内部 RAM 的内容。

RAM 能由 RAM 控制寄存器（RAMCR）的 RAME 位进行有效或者无效的控制。有关 RAMCR 请参照“19.3.7 RAM 控制寄存器（RAMCR）”。

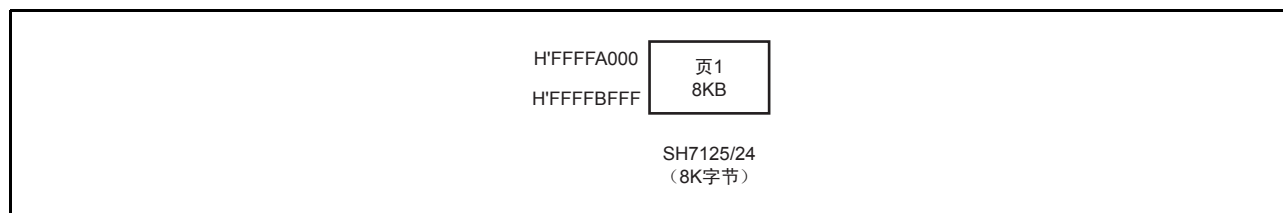


图 18.1 内部 RAM 地址

18.1 使用注意事项

18.1.1 模块待机模式的设定

能通过待机控制寄存器允许或者禁止 RAM 的存取，初始值为允许 RAM 的存取。通过设定模块待机模式，禁止 RAM 的存取。详细内容请参照“第 19 章 低功耗模式”。

18.1.2 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。

第 19 章 低功耗模式

本 LSI 支持低功耗模式的睡眠模式、软件待机模式和模块待机功能。

19.1 特点

- 支持睡眠模式、软件待机模式和模块待机功能。

19.1.1 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
- 模块待机功能

从程序的执行状态转移到各模式的方法、各模式中的 CPU 和外围模式等的状态以及各模式的解除方法如表 19.1 所示。

表 19.1 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 $\overline{\text{RES}}$ 管脚引起的加电复位
模块待机功能	将 STBCR2 ~ 5 的 MSTP 位设定为 1	运行	运行	保持	停止指定的模块 (保持内容)	停止指定的模块	(1) MSTP 位清 0 (2) 加电复位 (MSTP 位的初始值为 0 的模块)

【注】 有关各模式中的内部外围模块的寄存器状态和管脚状态请分别参照“20.3 各运行模式的寄存器状态”和“附录 A. 管脚状态”。

19.2 输入 / 输出管脚

低功耗模式关联的管脚结构如表 19.2 所示。

表 19.2 管脚结构

名称	管脚名	输入 / 输出	功 能
加电复位	$\overline{\text{RES}}$	输入	是加电复位的输入信号。通过低电平进行加电复位。
手动复位	$\overline{\text{MRES}}$	输入	是手动复位的输入信号。通过低电平进行手动复位。

19.3 寄存器说明

低功耗模式关联的寄存器有以下寄存器。有关这些寄存器的地址和各处理模式中的寄存器状态请参照“第 20 章 寄存器一览”。

表 19.3 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

19.3.1 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
6 ~ 0	—	全 0	R	保留位 读写的值总是 0。

19.3.2 待机控制寄存器 2（STBCR2）

STBCR2 是 8 位可读写寄存器，控制低功耗时的 RAM 运行。

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

位	位名	初始值	R/W	说 明
7	MSTP7	0	R/W	模块停止位 7 如果将本位设定为 1，就停止向 RAM 提供时钟。 0: RAM 运行 1: 停止向 RAM 提供时钟
6	—	0	R/W	保留位 读写的值总是 0。
5 ~ 3	—	全 1	R/W	保留位 读写的值总是 1。
2 ~ 0	—	全 0	R	保留位 读写的值总是 0。

19.3.3 待机控制寄存器 3（STBCR3）

STBCR3 是 8 位可读写寄存器，控制低功耗时各模块的运行。

位:	7	6	5	4	3	2	1	0
	-	-	MSTP 13	MSTP 12	MSTP 11	-	-	-
初始值:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7、6	—	全 1	R/W	保留位 读写的值总是 1。
5	MSTP13	1	R/W	模块停止位 13 如果将本位设定为 1，就停止向 SCI_2 提供时钟。 0: SCI_2 运行 1: 停止向 SCI_2 提供时钟
4	MSTP12	1	R/W	模块停止位 12 如果将本位设定为 1，就停止向 SCI_1 提供时钟。 0: SCI_1 运行 1: 停止向 SCI_1 提供时钟
3	MSTP11	1	R/W	模块停止位 11 如果将本位设定为 1，就停止向 SCI_0 提供时钟。 0: SCI_0 运行 1: 停止向 SCI_0 提供时钟
2 ~ 0	—	全 1	R/W	保留位 读写的值总是 1。

19.3.4 待机控制寄存器 4（STBCR4）

STBCR4 是 8 位可读写寄存器，控制低功耗时各模块的运行。

位:	7	6	5	4	3	2	1	0
	-	MSTP 22	MSTP 21	-	-	-	MSTP 17	MSTP 16
初始值:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说 明
7	—	1	R/W	保留位 读写的值总是 1。
6	MSTP22	1	R/W	模块停止位 22 如果将本位设定为 1，就停止向 MTU2 提供时钟。 0: MTU2 运行 1: 停止向 MTU2 提供时钟
5	MSTP21	1	R/W	模块停止位 21 如果将本位设定为 1，就停止向 CMT 提供时钟。 0: CMT 运行 1: 停止 CMT 提供时钟
4、3	—	全 1	R	保留位 读的值总是 1。
2	—	1	R/W	保留位 读写的值总是 1。
1	MSTP17	1	R/W	模块停止位 17 如果将本位设定为 1，就停止向 A/D_1 提供时钟。 0: A/D_1 运行 1: 停止向 A/D_1 提供时钟
0	MSTP16	1	R/W	模块停止位 16 如果将本位设定为 1，就停止向 A/D_0 提供时钟。 0: A/D_0 运行 1: 停止向 A/D_0 提供时钟

19.3.5 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[25:24]	
初始值:	0	0	0	0	0	0	1	1
R/W:	R	R	R	R	R	R	R/W	R/W

位	位名	初始值	R/W	说 明
7 ~ 2	—	全 0	R	保留位 读写的值总是 0。
1、0	MSTP[25:24]	11	R/W	模块停止位 25、24 如果将 2 位中的某一位设定为 1，就停止向 UBC 提供时钟。 00: UBC 运行 01: 停止向 UBC 提供时钟 10: 停止向 UBC 提供时钟 11: 停止向 UBC 提供时钟

19.3.6 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
	UBC RST	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	UBCRST	0	R/W	UBC 软件复位 通过软件控制 UBC 的复位。 如果给本位写 0，UBC 模块就进入加电复位状态。 0: 设定为 UBC 复位状态 1: 解除 UBC 的复位
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管脚状态还是设定为高阻抗状态。 0: 在软件待机模式时保持管脚状态 1: 在软件待机模式时将管脚状态设定为高阻抗
5 ~ 2	—	全 0	R	保留位 读写的值总是 0。
1	STBYMD	0	R/W	软件待机模式的选择 如果将本位设定为 1，就在 STBCR1 的 STBY 位为 1 的状态下执行 SLEEP 指令时转移到软件待机模式。 0: 禁止设定 1: 转移到软件待机模式
0	—	0	R	保留位 读写的值总是 0。

19.3.7 RAM 控制寄存器（RAMCR）

RAMCR 是 8 位可读写寄存器，允许或者禁止对内部 RAM 的存取。

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

位	位名	初始值	R/W	说 明
7 ~ 5	—	全 0	R	保留位 读写的值总是 0。
4	RAME	1	R/W	RAM 允许 选择内部 RAM 有效或者无效。 0: 内部 RAM 无效 1: 内部 RAM 有效 如果将本位清 0，就不能对内部 RAM 进行存取。此时，读内部 RAM 和取指令的值就不定，忽视内部 RAM 的写操作。 在将本位清 0（将内部 RAM 设定为无效）时，在 RAMCR 的写指令之后不能紧接内部 RAM 的存取指令。否则，就不能保证正常的存取。 在将本位设定为 1（将内部 RAM 置为有效）时，在 RAMCR 的写指令之后必须紧接 RAMCR 的读指令。如果在 RAMCR 的写指令之后紧接内部 RAM 的存取指令，就不能保证正常的存取。
3 ~ 0	—	全 0	R	保留位 读写的值总是 0。

19.4 睡眠模式

19.4.1 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。CPU 在执行 SLEEP 指令后停止，但是保持 CPU 的寄存器内容，并且内部外围模块继续运行。

19.4.2 睡眠模式的解除

通过复位解除睡眠模式。

(1) 通过复位的解除

通过 $\overline{\text{RES}}$ 管脚的加电复位、 $\overline{\text{MRES}}$ 管脚的手动复位或者 WDT 的内部加电复位 / 内部手动复位，解除睡眠模式。

19.5 软件待机模式

19.5.1 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。在软件待机模式中，不仅 CPU 而且时钟和内部外围模块都停止。

保持 CPU 的寄存器内容和内部 RAM 的数据，内部外围模块的寄存器可能被初始化。有关软件待机模式中的外围模块的寄存器状态和管脚状态请分别参照“20.3 各运行模式的寄存器状态”和“附录 A. 管脚状态”。

转移到软件待机模式的步骤如下所示：

1. 将 WDT 定时器的控制寄存器（WTCSR）的 TME 位设定为 0，停止 WDT。
2. 将 WDT 定时器的计数器（WTCNT）设定为 0，并给 WTCSR 寄存器的 CKS2 ~ CKS0 位设定与指定的振荡稳定时间对应的值。
3. 在将 STBCR1 的 STBY 位和 STBCR6 的 STBYMD 位设定为 1 后，执行 SLEEP 指令。
4. 进入软件待机模式，停止 LSI 内部的时钟。

19.5.2 软件待机模式的解除

通过中断（NMI、IRQ）或者复位，解除软件待机模式。

(1) 通过中断的解除

能通过内部 WDT 进行热启动。如果检测到 NMI 或者 IRQ 中断（边沿检测），就在经过 WDT 定时器的控制 / 状态寄存器设定的时间后给整个 LSI 提供时钟，并且解除软件待机模式。然后，执行中断异常处理。

但是，如果 IRQ 中断优先级低于 CPU 的状态寄存器（SR）设定的中断屏蔽级，就不接受中断也不解除软件待机模式。

另外，如果将 NMI 管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 管脚置为高电平；如果将 NMI 管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 管脚置为低电平。

同样地，如果将 IRQ 管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 管脚置为高电平；如果将 IRQ 管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 管脚置为低电平。

(2) 加电复位的解除

通过 $\overline{\text{RES}}$ 管脚的加电复位，解除软件待机模式。 $\overline{\text{RES}}$ 管脚必须在时钟振荡稳定前保持低电平。

(3) 手动复位的解除

本 LSI 不能通过手动复位解除软件待机模式。

19.6 模块待机功能

19.6.1 向模块待机功能的转移

能通过将待机控制寄存器 2 ~ 5（STBCR2 ~ 5）的各 MSTP 位设定为 1，停止向各自对应的内部外围模块提供时钟。通过使用此功能，能降低正常模式中的功耗。

不能存取被设定为模块待机模式的外围模块寄存器。有关模块待机模式中外围模块的寄存器状态，请参照“20.3 各运行模式的寄存器状态”。

19.6.2 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，也能通过加电复位进行解除。

19.7 使用注意事项

19.7.1 振荡稳定待机中的功耗

在振荡稳定待机中，功耗会增加。

第 20 章 寄存器一览

寄存器地址一览表汇总了有关内部寄存器地址、位结构以及各运行模式状态的信息。表的记述方法如下：

1. 寄存器地址一览表（地址顺序）

- 从分配地址小的寄存器顺序记载。
- 在寄存器名称栏用“—”表示保留的地址。
不能存取保留的地址。
- 在地址为 16 位或者 32 位时，记载 MSB 侧的地址。
- 根据模块名分类。
- 表示存取长度。

2. 位结构一览表

- 按照“寄存器地址一览（地址顺序）”的顺序记载位结构。
- 在位名栏中用“—”表示保留位
- 表中位名为空白的栏，表示该寄存器全部分配给计数器或者数据。
- 在 16 位或者 32 位寄存器时，从 MSB 侧的位开始记载。

3. 各运行模式的寄存器状态

- 按照“寄存器地址一览（地址顺序）”的顺序记载位结构。
- 表示基本运行模式的寄存器状态。在有内部模块固有的复位等时，请参照内部模块的章节。

20.1 寄存器地址一览表（地址顺序）

存取长度表示位数。

存取状态数表示指定的基准时钟的状态数。

但是，这些值是 B:8 位存取时、W:16 位存取时和 L:32 位存取时的值。

【注】 禁止存取未定义或者保留的地址。由于不保证存取这些寄存器时的运行及其后续运行，因此禁止存取。

寄存器名	略称	位数	地址	模块	存取长度	存取状态数
串行模式寄存器_0	SCSMR_0	8	H'FFFC000	SCI (通道 0)	8	P ϕ 基准 B:2
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
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
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
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	
定时器的输出主控允许寄存器	TOER	8	H'FFFC20A		8	
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	

寄存器名	略称	位数	地址	模块	存取长度	存取状态数
定时器的输出控制寄存器 2	TOCR2	8	H'FFFFC20F	MTU2	8	MP ϕ 基准 B:2、W:2、L:4
定时器的计数器 _3	TCNT_3	16	H'FFFFC210		16、32	
定时器的计数器 _4	TCNT_4	16	H'FFFFC212		16	
定时器的周期数据寄存器	TCDR	16	H'FFFFC214		16、32	
定时器的空载时间数据寄存器	TDDR	16	H'FFFFC216		16	
定时器的通用寄存器 A_3	TGRA_3	16	H'FFFFC218		16、32	
定时器的通用寄存器 B_3	TGRB_3	16	H'FFFFC21A		16	
定时器的通用寄存器 A_4	TGRA_4	16	H'FFFFC21C		16、32	
定时器的通用寄存器 B_4	TGRB_4	16	H'FFFFC21E		16	
定时器的子计数器	TCNTS	16	H'FFFFC220		16、32	
定时器的周期缓冲寄存器	TCBR	16	H'FFFFC222		16	
定时器的通用寄存器 C_3	TGRC_3	16	H'FFFFC224		16、32	
定时器的通用寄存器 D_3	TGRD_3	16	H'FFFFC226		16	
定时器的通用寄存器 C_4	TGRC_4	16	H'FFFFC228		16、32	
定时器的通用寄存器 D_4	TGRD_4	16	H'FFFFC22A		16	
定时器的状态寄存器 _3	TSR_3	8	H'FFFFC22C		8、16	
定时器的状态寄存器 _4	TSR_4	8	H'FFFFC22D		8	
定时器的跳过中断设定寄存器	TITCR	8	H'FFFFC230		8、16	
定时器的跳过中断次数计数器	TITCNT	8	H'FFFFC231		8	
定时器的缓冲传送设定寄存器	TBTER	8	H'FFFFC232		8	
定时器的空载时间允许寄存器	TDER	8	H'FFFFC234		8	
定时器的输出电平缓冲寄存器	TOLBR	8	H'FFFFC236		8	
定时器的缓冲运行传送模式寄存器 _3	TBTM_3	8	H'FFFFC238		8、16	
定时器的缓冲运行传送模式寄存器 _4	TBTM_4	8	H'FFFFC239		8	
定时器的 A/D 转换开始请求控制寄存器	TADCR	16	H'FFFFC240		16	
定时器的 A/D 转换开始请求周期设定寄存器 A_4	TADCORA_4	16	H'FFFFC244		16、32	
定时器的 A/D 转换开始请求周期设定寄存器 B_4	TADCORB_4	16	H'FFFFC246		16	
定时器的 A/D 转换开始请求周期设定缓冲寄存器 A_4	TADCOBRA_4	16	H'FFFFC248		16、32	
定时器的 A/D 转换开始请求周期设定缓冲寄存器 B_4	TADCOBRB_4	16	H'FFFFC24A		16	
定时器的波形控制寄存器	TWCR	8	H'FFFFC260		8	
定时器的开始寄存器	TSTR	8	H'FFFFC280		8、16	
定时器的同步寄存器	TSYR	8	H'FFFFC281		8	
定时器的计数器同步开始寄存器	TCSYSTR	8	H'FFFFC282		8	
定时器的读写允许寄存器	TRWER	8	H'FFFFC284		8	
定时器的控制寄存器 _0	TCR_0	8	H'FFFFC300		8、16、32	
定时器的模式寄存器 _0	TMDR_0	8	H'FFFFC301		8	
定时器的 I/O 控制寄存器 H_0	TIORH_0	8	H'FFFFC302		8、16	

寄存器名	略称	位数	地址	模块	存取长度	存取状态数
定时器的 I/O 控制寄存器 L_0	TIORL_0	8	H'FFFFC303	MTU2	8	MP ϕ 基准 B:2、W:2、L:4
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	
定时器的计数器 V_5	TCNTV_5	16	H'FFFFC490		16、32	
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	

寄存器名	略称	位数	地址	模块	存取长度	存取状态数
定时器的中断允许寄存器_5	TIER_5	8	H'FFFFC4B2	MTU2	8	MP ϕ 基准 B:2、W:2、L:4
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	
A/D 数据寄存器 0	ADDR0	16	H'FFFFC900	A/D (通道 0)	16	P ϕ 基准 B:2、W:2
A/D 数据寄存器 1	ADDR1	16	H'FFFFC902		16	
A/D 数据寄存器 2	ADDR2	16	H'FFFFC904		16	
A/D 数据寄存器 3	ADDR3	16	H'FFFFC906		16	
A/D 控制 / 状态寄存器_0	ADCSR_0	16	H'FFFFC910		16	
A/D 控制寄存器_0	ADCR_0	16	H'FFFFC912		16	
A/D 数据寄存器 4	ADDR4	16	H'FFFFC980	A/D (通道 1)	16	P ϕ 基准 B:2、W:2
A/D 数据寄存器 5	ADDR5	16	H'FFFFC982		16	
A/D 数据寄存器 6	ADDR6	16	H'FFFFC984		16	
A/D 数据寄存器 7	ADDR7	16	H'FFFFC986		16	
A/D 控制 / 状态寄存器_1	ADCSR_1	16	H'FFFFC990		16	
A/D 控制寄存器_1	ADCR_1	16	H'FFFFC992		16	
闪存代码控制 / 状态寄存器	FCCS	8	H'FFFFCC00	FLASH	8	P ϕ 基准 B:5
闪存编程代码选择寄存器	FPCS	8	H'FFFFCC01		8	
闪存擦除代码选择寄存器	FECS	8	H'FFFFCC02		8	
闪存键代码寄存器	FKEY	8	H'FFFFCC04		8	
闪存传送目标地址寄存器	FTDAR	8	H'FFFFCC06		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
比较匹配定时器的控制 / 状态寄存器_0	CMCSR_0	16	H'FFFFCE02		8、16	
比较匹配计数器_0	CMCNT_0	16	H'FFFFCE04		8、16、32	
比较匹配常数寄存器_0	CMCOR_0	16	H'FFFFCE06		8、16	
比较匹配定时器的控制 / 状态寄存器_1	CMCSR_1	16	H'FFFFCE08		8、16、32	
比较匹配计数器_1	CMCNT_1	16	H'FFFFCE0A		8、16	
比较匹配常数寄存器_1	CMCOR_1	16	H'FFFFCE0C		8、16、32	
输入电平控制 / 状态寄存器 1	ICSR1	16	H'FFFFD000	POE	8、16、32	P ϕ 基准 B:2、W:2、L:4
输出电平控制 / 状态寄存器 1	OCSR1	16	H'FFFFD002		8、16	
输入电平控制 / 状态寄存器 3	ICSR3	16	H'FFFFD008		8、16	
软件端口输出允许寄存器	SPOER	8	H'FFFFD00A		8	
端口输出允许控制寄存器 1	POECR1	8	H'FFFFD00B		8	
端口输出允许控制寄存器 2	POECR2	16	H'FFFFD00C		8、16	
端口 A 的数据寄存器 L	PADRL	16	H'FFFFD102	I/O	8、16	P ϕ 基准 B:2、W:2、L:4
端口 A 的 IO 寄存器 L	PAIORL	16	H'FFFFD106	PFC	8、16	
端口 A 的控制寄存器 L4	PACRL4	16	H'FFFFD110		8、16、32	
端口 A 的控制寄存器 L3	PACRL3	16	H'FFFFD112		8、16	
端口 A 的控制寄存器 L2	PACRL2	16	H'FFFFD114		8、16、32	
端口 A 的控制寄存器 L1	PACRL1	16	H'FFFFD116		8、16	
端口 A 的端口寄存器 L	PAPRL	16	H'FFFFD11E	I/O	8、16	
端口 B 的数据寄存器 H	PBDRH	16	H'FFFFD180		8、16、32	

寄存器名	略称	位数	地址	模块	存取长度	存取状态数
端口 B 的数据寄存器 L	PBDRL	16	H'FFFFD182	I/O	8、16	P ϕ 基准 B:2、W:2、L:4
端口 B 的 IO 寄存器 H	PBIORH	16	H'FFFFD184	PFC	8、16、32	
端口 B 的 IO 寄存器 L	PBIORL	16	H'FFFFD186		8、16	
端口 B 的控制寄存器 H1	PBCRH1	16	H'FFFFD18E		8、16	
端口 B 的控制寄存器 L2	PBCRL2	16	H'FFFFD194		8、16、32	
端口 B 的控制寄存器 L1	PBCRL1	16	H'FFFFD196		8、16	
端口 B 的端口寄存器 H	PBPRH	16	H'FFFFD19C	I/O	8、16、32	
端口 B 的端口寄存器 L	PBPRL	16	H'FFFFD19E		8、16	
端口 E 的数据寄存器 L	PEDRL	16	H'FFFFD302		8、16	
端口 E 的 IO 寄存器 L	PEIORL	16	H'FFFFD306	PFC	8、16	
端口 E 的控制寄存器 L4	PECRL4	16	H'FFFFD310		8、16、32	
端口 E 的控制寄存器 L3	PECRL3	16	H'FFFFD312		8、16	
端口 E 的控制寄存器 L2	PECRL2	16	H'FFFFD314		8、16、32	
端口 E 的控制寄存器 L1	PECRL1	16	H'FFFFD316		8、16	
端口 E 的端口寄存器 L	PEPRL	16	H'FFFFD31E	I/O	8、16	
IRQOUT 功能控制寄存器	IFCR	16	H'FFFFD322	PFC	8、16	
端口 F 的数据寄存器 L	PFDR L	16	H'FFFFD382	I/O	8、16	
频率控制寄存器	FRQCR	16	H'FFFFE800	CPG	16	P ϕ 基准 W:2
待机控制寄存器 1	STBCR1	8	H'FFFFE802	低功耗	8	P ϕ 基准 B:2
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	8*1、16*2	P ϕ 基准 B:2*1、W:2*2
看门狗定时器的控制 / 状态寄存器	WTCSR	8	H'FFFFE812		*1: 读时 *2: 写时 8*1、16*2	
振荡停止检测控制寄存器	OSCCR	8	H'FFFFE814	CPG	8	P ϕ 基准 B:2
RAM 控制寄存器	RAMCR	8	H'FFFFE880	低功耗	8	P ϕ 基准 B:2
A/D 触发选择寄存器 _0	ADTSR_0	16	H'FFFFE890	A/D	8、16	P ϕ 基准 B:2、W:2
中断控制寄存器 0	ICR0	16	H'FFFFE900	INTC	8、16	P ϕ 基准 B:2、W:2
IRQ 控制寄存器	IRQCR	16	H'FFFFE902		8、16	
IRQ 状态寄存器	IRQSR	16	H'FFFFE904		8、16	
中断优先级寄存器 A	IPRA	16	H'FFFFE906		8、16	
中断优先级寄存器 B	IPRB	16	H'FFFFE908		8、16	
中断优先级寄存器 C	IPRC	16	H'FFFFE980		16	
中断优先级寄存器 D	IPRD	16	H'FFFFE982		16	
中断优先级寄存器 E	IPRE	16	H'FFFFE984		16	
中断优先级寄存器 F	IPRF	16	H'FFFFE986		16	
中断优先级寄存器 H	IPRH	16	H'FFFFE98A		16	
中断优先级寄存器 I	IPRI	16	H'FFFFE98C		16	
中断优先级寄存器 J	IPRJ	16	H'FFFFE98E		16	
中断优先级寄存器 K	IPRK	16	H'FFFFE990		16	

寄存器名	略称	位数	地址	模块	存取长度	存取状态数
中断优先级寄存器 L	IPRL	16	H'FFFFE992	INTC	16	P ϕ 基准 B:2、W:2
中断优先级寄存器 M	IPRM	16	H'FFFFE994		16	
断点地址寄存器 A	BARA	32	H'FFFFFF300	UBC	32	B ϕ 基准 B:2、W:2、L:2
断点地址屏蔽寄存器 A	BAMRA	32	H'FFFFFF304		32	
断点总线周期寄存器 A	BBRA	16	H'FFFFFF308		16	
断点数据寄存器 A	BDRA	32	H'FFFFFF310		32	
断点数据屏蔽寄存器 A	BDMRA	32	H'FFFFFF314		32	
断点地址寄存器 B	BARB	32	H'FFFFFF320		32	
断点地址屏蔽寄存器 B	BAMRB	32	H'FFFFFF324		32	
断点总线周期寄存器 B	BBRB	16	H'FFFFFF328		16	
断点数据寄存器 B	BDRB	32	H'FFFFFF330		32	
断点数据屏蔽寄存器 B	BDMRB	32	H'FFFFFF334		32	
断点控制寄存器	BRCR	32	H'FFFFFF3C0		32	
转移源寄存器	BRSR	32	H'FFFFFF3D0		32	
转移目标寄存器	BRDR	32	H'FFFFFF3D4		32	
执行次数断点寄存器	BETR	16	H'FFFFFF3DC		16	

20.2 寄存器位一览

内部外围模块寄存器的地址和位名如下所示：

对于 16 位、32 位寄存器，分别以 8 位分 2 行或者 4 行表示。

寄存器略称	位 31/23/15/7	位 30/22/14/6	位 29/21/13/5	位 28/20/12/4	位 27/19/11/3	位 26/18/10/2	位 25/17/9/1	位 24/16/8/0	模块	
SCSMR_0	C/ $\overline{A}$	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		

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	
TOER	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B	MTU2	
TGCR	—	BDC	N	P	FB	WF	VF	UF		
TOCR1	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP		
TOCR2	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P		
TCNT_3										
TCNT_4										
TCDR										
TDDR										
TGRA_3										
TGRB_3										
TGRA_4										
TGRB_4										
TCNTS										
TCBR										
TGRC_3										
TGRD_3										
TGRC_4										
TGRD_4										
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		

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	
TOLBR	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P	MTU2	
TBTM_3	—	—	—	—	—	—	TTSB	TTSA		
TBTM_4	—	—	—	—	—	—	TTSB	TTSA		
TADCR	BF[1:0]		—	—	—	—	—	—		
	UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE		
TADCORA_4										
TADCORB_4										
TADCOBRA_4										
TADCOBRB_4										
TWCR	CCE	—	—	—	—	—	—	WRE		
TSTR	CST4	CST3	—	—	—	CST2	CST1	CST0		
TSYR	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0		
TCSYSTR	SCH0	SCH1	SCH2	SCH3	SCH4	—	—	—		
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										

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
TIER2_0	TTGE2	—	—	—	—	—	TGIEF	TGIEE	MTU2
TSR2_0	—	—	—	—	—	—	TGFF	TGFE	
TBTM_0	—	—	—	—	—	TTSE	TTSB	TTSA	
TCR_1	—	CCLR[1:0]		CKEG[1:0]		TPSC[2:0]			
TMDR_1	—	—	—	—	MD[3:0]				
TIOR_1	IOB[3:0]				IOA[3:0]				
TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_1									
TGRA_1									
TGRB_1									
TICCR	—	—	—	—	I2BE	I2AE	I1BE	I1AE	
TCR_2	—	CCLR[1:0]		CKEG[1:0]		TPSC[2:0]			
TMDR_2	—	—	—	—	MD[3:0]				
TIOR_2	IOB[3:0]				IOA[3:0]				
TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_2									
TGRA_2									
TGRB_2									
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]					

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
TCNTW_5									MTU2
TGRW_5									
TCRW_5	—	—	—	—	—	—	TPSC[1:0]		
TIORW_5	—	—	—	IOC[4:0]					
TSR_5	—	—	—	—	—	CMFU5	CMFV5	CMFW5	
TIER_5	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W	
TSTR_5	—	—	—	—	—	CSTU5	CSTV5	CSTW5	
TCNTCMPCLR	—	—	—	—	—	CMPCLR5U	CMPCLR5V	CMPCLR5W	
ADDR0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D (通道 0)
	AD1	AD0	—	—	—	—	—	—	
ADDR1	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR2	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR3	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADCSR_0	ADF	ADIE	—	—	TRGE	—	CONADF	STC	A/D (通道 0)
	CKSL[1:0]		ADM[1:0]		ADCS	CH[2:0]			
ADCR_0	—	—	ADST	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
ADDR4	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D (通道 1)
	AD1	AD0	—	—	—	—	—	—	
ADDR5	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR6	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDR7	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADCSR_1	ADF	ADIE	—	—	TRGE	—	CONADF	STC	
	CKSL[1:0]		ADM[1:0]		ADCS	CH[2:0]			
ADCR_1	—	—	ADST	—	—	—	—	—	
	—	—	—	—	—	—	—	—	

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
FCCS	FWE	—	—	FLER	—	—	—	SCO	FLASH
FPCS	—	—	—	—	—	—	—	PPVS	
FECS	—	—	—	—	—	—	—	EPVB	
FKEY	K[7:0]								
FTDAR	TDER	TDA[6:0]							
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	POE3F	—	POE1F	POE0F	—	—	—	PIE1	POE
	POE3M[1:0]		—	—	POE1M[1:0]		POE0M[1:0]		
OCSR1	OSF1	—	—	—	—	—	OCE1	OIE1	
	—	—	—	—	—	—	—	—	
ICSR3	—	—	—	POE8F	—	—	POE8E	PIE3	
	—	—	—	—	—	—	POE8M[1:0]		
SPOER	—	—	—	—	—	—	MTU2CH0HIZ	MTU2CH34HIZ	
POECR1	—	—	—	—	MTU2PE3ZE	MTU2PE2ZE	MTU2PE1ZE	MTU2PE0ZE	POE
POECR2	—	MTU2P1CZE	MTU2P2CZE	MTU2P3CZE	—	—	—	—	
	—	—	—	—	—	—	—	—	
PADRL (SH7125)	PA15DR	PA14DR	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR	PA8DR	I/O
	PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR	
PADRL (SH7124)	—	—	—	—	—	—	PA9DR	PA8DR	
	PA7DR	PA6DR	—	PA4DR	PA3DR	—	PA1DR	PA0DR	

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
PAIORL (SH7125)	PA15IOR	PA14IOR	PA13IOR	PA12IOR	PA11IOR	PA10IOR	PA9IOR	PA8IOR	PFC
	PA7IOR	PA6IOR	PA5IOR	PA4IOR	PA3IOR	PA2IOR	PA1IOR	PA0IOR	
PAIORL (SH7124)	—	—	—	—	—	—	PA9IOR	PA8IOR	
	PA7IOR	PA6IOR	—	PA4IOR	PA3IOR	—	PA1IOR	PA0IOR	
PACRL4 (SH7125)	—	PA15MD2	PA15MD1	PA15MD0	—	PA14MD2	PA14MD1	PA14MD0	
	—	PA13MD2	PA13MD1	PA13MD0	—	PA12MD2	PA12MD1	PA12MD0	
PACRL4 (SH7124)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
PACRL3 (SH7125)	—	PA11MD2	PA11MD1	PA11MD0	—	PA10MD2	PA10MD1	PA10MD0	
	—	PA9MD2	PA9MD1	PA9MD0	—	PA8MD2	PA8MD1	PA8MD0	
PACRL3 (SH7124)	—	—	—	—	—	—	—	—	
	—	PA9MD2	PA9MD1	PA9MD0	—	PA8MD2	PA8MD1	PA8MD0	
PACRL2 (SH7125)	—	PA7MD2	PA7MD1	PA7MD0	—	PA6MD2	PA6MD1	PA6MD0	
	—	PA5MD2	PA5MD1	PA5MD0	—	PA4MD2	PA4MD1	PA4MD0	
PACRL2 (SH7124)	—	PA7MD2	PA7MD1	PA7MD0	—	PA6MD2	PA6MD1	PA6MD0	
	—	—	—	—	—	PA4MD2	PA4MD1	PA4MD0	
PACRL1 (SH7125)	—	PA3MD2	PA3MD1	PA3MD0	—	PA2MD2	PA2MD1	PA2MD0	I/O
	—	PA1MD2	PA1MD1	PA1MD0	—	PA0MD2	PA0MD1	PA0MD0	
PACRL1 (SH7124)	—	PA3MD2	PA3MD1	PA3MD0	—	—	—	—	
	—	PA1MD2	PA1MD1	PA1MD0	—	PA0MD2	PA0MD1	PA0MD0	
PAPRL (SH7125)	PA15PR	PA14PR	PA13PR	PA12PR	PA11PR	PA10PR	PA9PR	PA8PR	
	PA7PR	PA6PR	PA5PR	PA4PR	PA3PR	PA2PR	PA1PR	PA0PR	
PAPRL (SH7124)	—	—	—	—	—	—	PA9PR	PA8PR	
	PA7PR	PA6PR	—	PA4PR	PA3PR	—	PA1PR	PA0PR	
PBDRH (SH7125)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	PB16DR	
PBDRH (SH7124)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
PBDRL (SH7125)	—	—	—	—	—	—	—	—	
	—	—	PB5DR	—	PB3DR	PB2DR	PB1DR	—	
PBDRL (SH7124)	—	—	—	—	—	—	—	—	
	—	—	PB5DR	—	PB3DR	—	PB1DR	—	
PBIORH (SH7125)	—	—	—	—	—	—	—	—	PFC
	—	—	—	—	—	—	—	PB16IOR	
PBIORH (SH7124)	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	

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
PBIORL (SH7125)	—	—	—	—	—	—	—	—	PFC
PBIORL (SH7124)	—	—	PB5IOR	—	PB3IOR	PB2IOR	PB1IOR	—	
PBCRH1 (SH7125)	—	—	—	—	—	—	—	—	
PBCRH1 (SH7124)	—	—	—	—	—	—	—	—	
PBCRL2 (SH7125)	—	—	—	—	—	—	—	—	
PBCRL2 (SH7124)	—	—	—	—	—	—	—	—	
PBCRL1 (SH7125)	—	—	—	—	—	—	—	—	
PBCRL1 (SH7124)	—	—	—	—	—	—	—	—	
PBCRL1 (SH7125)	—	—	—	—	—	—	—	—	
PBCRL1 (SH7124)	—	—	—	—	—	—	—	—	
PBPRH (SH7125)	—	—	—	—	—	—	—	—	I/O
PBPRH (SH7124)	—	—	—	—	—	—	—	—	
PBPRL (SH7125)	—	—	—	—	—	—	—	—	
PBPRL (SH7124)	—	—	—	—	—	—	—	—	
PEDRL (SH7125)	PE15DR	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR	
PEDRL (SH7124)	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR	
PEDRL (SH7125)	PE15DR	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR	
PEDRL (SH7124)	—	—	—	—	PE3DR	PE2DR	PE1DR	PE0DR	
PEIORL (SH7125)	PE15IOR	PE14IOR	PE13IOR	PE12IOR	PE11IOR	PE10IOR	PE9IOR	PE8IOR	PFC
PEIORL (SH7124)	PE7IOR	PE6IOR	PE5IOR	PE4IOR	PE3IOR	PE2IOR	PE1IOR	PE0IOR	
PEIORL (SH7124)	—	—	—	—	PE3IOR	PE2IOR	PE1IOR	PE0IOR	
PECRL4	—	—	—	—	—	—	—	—	
PECRL4	—	—	—	—	—	—	—	—	
PECRL3	—	—	—	—	—	—	—	—	
PECRL3	—	—	—	—	—	—	—	—	
PECRL2 (SH7125)	—	—	—	—	—	—	—	—	
PECRL2 (SH7125)	—	—	—	—	—	—	—	—	
PECRL2 (SH7125)	—	—	—	—	—	—	—	—	

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	
PECRL2 (SH7124)	—	—	—	—	—	—	—	—	PFC	
	—	—	—	—	—	—	—	—		
PECRL1	—	PE3MD2	PE3MD1	PE3MD0	—	PE2MD2	PE2MD1	PE2MD0		
	—	PE1MD2	PE1MD1	PE1MD0	—	—	PE0MD1	PE0MD0		
PEPRL (SH7125)	PE15PR	PE14PR	PE13PR	PE12PR	PE11PR	PE10PR	PE9PR	PE8PR	I/O	
	PE7PR	PE6PR	PE5PR	PE4PR	PE3PR	PE2PR	PE1PR	PE0PR		
PEPRL (SH7124)	PE15PR	PE14PR	PE13PR	PE12PR	PE11PR	PE10PR	PE9PR	PE8PR		
	—	—	—	—	PE3PR	PE2PR	PE1PR	PE0PR		
IFCR	—	—	—	—	—	—	—	—	PFC	
	—	—	—	—	—	—	IRQMD1	IRQMD0		
PFDRL	—	—	—	—	—	—	—	—	I/O	
	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR		
FRQCR	—	IFC[2:0]			BFC[2:0]			PFC[2]	CPG	
	PFC[1:0]		—	—	—	MPFC[2:0]				
STBCR1	STBY	—	—	—	—	—	—	—	低功耗	
STBCR2	MSTP7	—	—	—	—	—	—	—		
STBCR3	—	—	MSTP13	MSTP12	MSTP11	—	—	—		
STBCR4	—	MSTP22	MSTP21	—	—	—	MSTP17	MSTP16		
STBCR5	—	—	—	—	—	—	MSTP[25:24]			
STBCR6	UBCRST	HIZ	—	—	—	—	STBYMD	—		
WTCNT									WDT	
WTCSR	TME	WT/IT	RSTS	WOVF	IOVF	CKS[2:0]				
OSCCR	—	—	—	—	—	OSCSTOP	—	OSCERS	CPG	
RAMCR	—	—	—	RAME	—	—	—	—	低功耗	
ADTSR_0	TRG11S[3:0]				TRG01S[3:0]				A/D	
	TRG1S[3:0]				TRG0S[3:0]					
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		
IPRB	—	—	—	—	—	—	—	—		
	—	—	—	—	—	—	—	—		
IPRC	—	—	—	—	—	—	—	—		
	—	—	—	—	—	—	—	—		

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	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
IPRI	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
IPRJ	CMT_0	CMT_0	CMT_0	CMT_0	CMT_1	CMT_1	CMT_1	CMT_1	
	—	—	—	—	WDT	WDT	WDT	WDT	
IPRK	A/D_0,1	A/D_0,1	A/D_0,1	A/D_0,1	—	—	—	—	
	—	—	—	—	—	—	—	—	
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	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
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	

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	
BARB	BAB31	BAB30	BAB29	BAB28	BAB27	BAB26	BAB25	BAB24	UBC	
	BAB23	BAB22	BAB21	BAB20	BAB19	BAB18	BAB17	BAB16		
	BAB15	BAB14	BAB13	BAB12	BAB11	BAB10	BAB9	BAB8		
	BAB7	BAB6	BAB5	BAB4	BAB3	BAB2	BAB1	BAB0		
BAMRB	BAMB31	BAMB30	BAMB29	BAMB28	BAMB27	BAMB26	BAMB25	BAMB24		
	BAMB23	BAMB22	BAMB21	BAMB20	BAMB19	BAMB18	BAMB17	BAMB16		
	BAMB15	BAMB14	BAMB13	BAMB12	BAMB11	BAMB10	BAMB9	BAMB8		
	BAMB7	BAMB6	BAMB5	BAMB4	BAMB3	BAMB2	BAMB1	BAMB0		
BBRB	—	—	—	—	—	CPB[2:0]				
	CDB[1:0]		IDB[1:0]		RWB[1:0]		SZB[1:0]			
BDRB	BDB31	BDB30	BDB29	BDB28	BDB27	BDB26	BDB25	BDB24		
	BDB23	BDB22	BDB21	BDB20	BDB19	BDB18	BDB17	BDB16		
	BDB15	BDB14	BDB13	BDB12	BDB11	BDB10	BDB9	BDB8		
	BDB7	BDB6	BDB5	BDB4	BDB3	BDB2	BDB1	BDB0		
BDMRB	BDMB31	BDMB30	BDMB29	BDMB28	BDMB27	BDMB26	BDMB25	BDMB24		
	BDMB23	BDMB22	BDMB21	BDMB20	BDMB19	BDMB18	BDMB17	BDMB16		
	BDMB15	BDMB14	BDMB13	BDMB12	BDMB11	BDMB10	BDMB9	BDMB8		
	BDMB7	BDMB6	BDMB5	BDMB4	BDMB3	BDMB2	BDMB1	BDMB0		
BRCR	—	—	—	—	—	—	—	—		
	—	—	—	—	UBIDB	—	UBIDA	—		
	SCMFCA	SCMFCA	SCMFDA	SCMFDB	PCTE	PCBA	—	—		
	DBEA	PCBB	DBEB	—	SEQ	—	—	ETBE		
BRSR	SVF	—	—	—	BSA27	BSA26	BSA25	BSA24		
	BSA23	BSA22	BSA21	BSA20	BSA19	BSA18	BSA17	BSA16		
	BSA15	BSA14	BSA13	BSA12	BSA11	BSA10	BSA9	BSA8		
	BSA7	BSA6	BSA5	BSA4	BSA3	BSA2	BSA1	BSA0		
BRDR	DVF	—	—	—	BDA27	BDA26	BDA25	BDA24		
	BDA23	BDA22	BDA21	BDA20	BDA19	BDA18	BDA17	BDA16		
	BDA15	BDA14	BDA13	BDA12	BDA11	BDA10	BDA9	BDA8		
	BDA7	BDA6	BDA5	BDA4	BDA3	BDA2	BDA1	BDA0		
BETR	—	—	—	—	BET[11:8]					
	BET[7:0]									

20.3 各运行模式的寄存器状态

寄存器略称	加电复位	手动复位	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	保持	SCI (通道 0)
SCBRR_0	初始化	保持	初始化	初始化	保持	
SCSCR_0	初始化	保持	初始化	初始化	保持	
SCTDR_0	初始化	保持	初始化	初始化	保持	
SCSSR_0	初始化	保持	初始化	初始化	保持	
SCRDR_0	初始化	保持	初始化	初始化	保持	
SCSDCR_0	初始化	保持	初始化	初始化	保持	
SCSPTR_0	初始化	保持	初始化	初始化	保持	SCI (通道 1)
SCSMR_1	初始化	保持	初始化	初始化	保持	
SCBRR_1	初始化	保持	初始化	初始化	保持	
SCSCR_1	初始化	保持	初始化	初始化	保持	
SCTDR_1	初始化	保持	初始化	初始化	保持	
SCSSR_1	初始化	保持	初始化	初始化	保持	
SCRDR_1	初始化	保持	初始化	初始化	保持	
SCSDCR_1	初始化	保持	初始化	初始化	保持	SCI (通道 2)
SCSPTR_1	初始化	保持	初始化	初始化	保持	
SCSMR_2	初始化	保持	初始化	初始化	保持	
SCBRR_2	初始化	保持	初始化	初始化	保持	
SCSCR_2	初始化	保持	初始化	初始化	保持	
SCTDR_2	初始化	保持	初始化	初始化	保持	
SCSSR_2	初始化	保持	初始化	初始化	保持	
SCRDR_2	初始化	保持	初始化	初始化	保持	MTU2
SCSDCR_2	初始化	保持	初始化	初始化	保持	
SCSPTR_2	初始化	保持	初始化	初始化	保持	
TCR_3	初始化	保持	初始化	初始化	保持	
TCR_4	初始化	保持	初始化	初始化	保持	
TMDR_3	初始化	保持	初始化	初始化	保持	
TMDR_4	初始化	保持	初始化	初始化	保持	
TIORH_3	初始化	保持	初始化	初始化	保持	
TIORL_3	初始化	保持	初始化	初始化	保持	
TIORH_4	初始化	保持	初始化	初始化	保持	
TIORL_4	初始化	保持	初始化	初始化	保持	
TIER_3	初始化	保持	初始化	初始化	保持	
TIER_4	初始化	保持	初始化	初始化	保持	
TOER	初始化	保持	初始化	初始化	保持	
TGCR	初始化	保持	初始化	初始化	保持	
TOCR1	初始化	保持	初始化	初始化	保持	
TOCR2	初始化	保持	初始化	初始化	保持	
TCNT_3	初始化	保持	初始化	初始化	保持	
TCNT_4	初始化	保持	初始化	初始化	保持	
TCDR	初始化	保持	初始化	初始化	保持	
TDDR	初始化	保持	初始化	初始化	保持	

寄存器略称	加电复位	手动复位	软件待机	模块待机	睡眠	模块
TGRA_3	初始化	保持	初始化	初始化	保持	MTU2
TGRB_3	初始化	保持	初始化	初始化	保持	
TGRA_4	初始化	保持	初始化	初始化	保持	
TGRB_4	初始化	保持	初始化	初始化	保持	
TCNTS	初始化	保持	初始化	初始化	保持	
TCBR	初始化	保持	初始化	初始化	保持	
TGRC_3	初始化	保持	初始化	初始化	保持	
TGRD_3	初始化	保持	初始化	初始化	保持	
TGRC_4	初始化	保持	初始化	初始化	保持	
TGRD_4	初始化	保持	初始化	初始化	保持	
TSR_3	初始化	保持	初始化	初始化	保持	
TSR_4	初始化	保持	初始化	初始化	保持	
TITCR	初始化	保持	初始化	初始化	保持	
TITCNT	初始化	保持	初始化	初始化	保持	
TBTER	初始化	保持	初始化	初始化	保持	
TDER	初始化	保持	初始化	初始化	保持	
TOLBR	初始化	保持	初始化	初始化	保持	
TBTM_3	初始化	保持	初始化	初始化	保持	
TBTM_4	初始化	保持	初始化	初始化	保持	
TADCR	初始化	保持	初始化	初始化	保持	
TADCORA_4	初始化	保持	初始化	初始化	保持	
TADCORB_4	初始化	保持	初始化	初始化	保持	
TADCOBRA_4	初始化	保持	初始化	初始化	保持	
TADCOBRB_4	初始化	保持	初始化	初始化	保持	
TWCR	初始化	保持	初始化	初始化	保持	
TSTR	初始化	保持	初始化	初始化	保持	
TSYR	初始化	保持	初始化	初始化	保持	
TCSYSTR	初始化	保持	初始化	初始化	保持	
TRWER	初始化	保持	初始化	初始化	保持	
TCR_0	初始化	保持	初始化	初始化	保持	
TMDR_0	初始化	保持	初始化	初始化	保持	
TIORH_0	初始化	保持	初始化	初始化	保持	
TIORL_0	初始化	保持	初始化	初始化	保持	
TIER_0	初始化	保持	初始化	初始化	保持	
TSR_0	初始化	保持	初始化	初始化	保持	
TCNT_0	初始化	保持	初始化	初始化	保持	
TGRA_0	初始化	保持	初始化	初始化	保持	
TGRB_0	初始化	保持	初始化	初始化	保持	
TGRC_0	初始化	保持	初始化	初始化	保持	
TGRD_0	初始化	保持	初始化	初始化	保持	
TGRE_0	初始化	保持	初始化	初始化	保持	
TGRF_0	初始化	保持	初始化	初始化	保持	
TIER2_0	初始化	保持	初始化	初始化	保持	
TSR2_0	初始化	保持	初始化	初始化	保持	

寄存器略称	加电复位	手动复位	软件待机	模块待机	睡眠	模块
TBTM_0	初始化	保持	初始化	初始化	保持	MTU2
TCR_1	初始化	保持	初始化	初始化	保持	
TMDR_1	初始化	保持	初始化	初始化	保持	
TIOR_1	初始化	保持	初始化	初始化	保持	
TIER_1	初始化	保持	初始化	初始化	保持	
TSR_1	初始化	保持	初始化	初始化	保持	
TCNT_1	初始化	保持	初始化	初始化	保持	
TGRA_1	初始化	保持	初始化	初始化	保持	
TGRB_1	初始化	保持	初始化	初始化	保持	
TICCR	初始化	保持	初始化	初始化	保持	
TCR_2	初始化	保持	初始化	初始化	保持	
TMDR_2	初始化	保持	初始化	初始化	保持	
TIOR_2	初始化	保持	初始化	初始化	保持	
TIER_2	初始化	保持	初始化	初始化	保持	
TSR_2	初始化	保持	初始化	初始化	保持	
TCNT_2	初始化	保持	初始化	初始化	保持	
TGRA_2	初始化	保持	初始化	初始化	保持	
TGRB_2	初始化	保持	初始化	初始化	保持	
TCNTU_5	初始化	保持	初始化	初始化	保持	
TGRU_5	初始化	保持	初始化	初始化	保持	
TCRU_5	初始化	保持	初始化	初始化	保持	
TIORU_5	初始化	保持	初始化	初始化	保持	
TCNTV_5	初始化	保持	初始化	初始化	保持	
TGRV_5	初始化	保持	初始化	初始化	保持	
TCRV_5	初始化	保持	初始化	初始化	保持	
TIORV_5	初始化	保持	初始化	初始化	保持	
TCNTW_5	初始化	保持	初始化	初始化	保持	
TGRW_5	初始化	保持	初始化	初始化	保持	
TCRW_5	初始化	保持	初始化	初始化	保持	
TIORW_5	初始化	保持	初始化	初始化	保持	
TSR_5	初始化	保持	初始化	初始化	保持	
TIER_5	初始化	保持	初始化	初始化	保持	
TSTR5	初始化	保持	初始化	初始化	保持	
TCNTCMPCLR	初始化	保持	初始化	初始化	保持	
ADDR0	初始化	保持	初始化	初始化	保持	A/D (通道 0)
ADDR1	初始化	保持	初始化	初始化	保持	
ADDR2	初始化	保持	初始化	初始化	保持	
ADDR3	初始化	保持	初始化	初始化	保持	
ADCSR_0	初始化	保持	初始化	初始化	保持	
ADCR_0	初始化	保持	初始化	初始化	保持	
ADDR4	初始化	保持	初始化	初始化	保持	A/D (通道 1)
ADDR5	初始化	保持	初始化	初始化	保持	
ADDR6	初始化	保持	初始化	初始化	保持	
ADDR7	初始化	保持	初始化	初始化	保持	

寄存器略称	加电复位	手动复位	软件待机	模块待机	睡眠	模块
ADCSR_1	初始化	保持	初始化	初始化	保持	A/D (通道 1)
ADCR_1	初始化	保持	初始化	初始化	保持	
FCCS	初始化	保持	初始化	初始化	保持	FLASH
FPCS	初始化	保持	初始化	初始化	保持	
FECS	初始化	保持	初始化	初始化	保持	
FKEY	初始化	保持	初始化	初始化	保持	
FTDAR	初始化	保持	初始化	初始化	保持	
CMSTR	初始化	保持	初始化	初始化	保持	CMT
CMCSR_0	初始化	保持	初始化	初始化	保持	
CMCNT_0	初始化	保持	初始化	初始化	保持	
CMCOR_0	初始化	保持	初始化	初始化	保持	
CMCSR_1	初始化	保持	初始化	初始化	保持	
CMCNT_1	初始化	保持	初始化	初始化	保持	
CMCOR_1	初始化	保持	初始化	初始化	保持	POE
ICSR1	初始化	保持	保持	—	保持	
OCSR1	初始化	保持	保持	—	保持	
ICSR3	初始化	保持	保持	—	保持	
SPOER	初始化	保持	保持	—	保持	
POECR1	初始化	保持	保持	—	保持	
POECR2	初始化	保持	保持	—	保持	I/O
PADRL	初始化	保持	保持	—	保持	
PAIORL	初始化	保持	保持	—	保持	PFC
PACRL4	初始化	保持	保持	—	保持	
PACRL3	初始化	保持	保持	—	保持	
PACRL2	初始化	保持	保持	—	保持	
PACRL1	初始化	保持	保持	—	保持	
PAPRL	初始化	保持	保持	—	保持	I/O
PBDRH	初始化	保持	保持	—	保持	
PBDRL	初始化	保持	保持	—	保持	
PBIORH	初始化	保持	保持	—	保持	
PBIORL	初始化	保持	保持	—	保持	PFC
PBCRH1	初始化	保持	保持	—	保持	
PBCRL2	初始化	保持	保持	—	保持	
PBCRL1	初始化	保持	保持	—	保持	
PBPRH	初始化	保持	保持	—	保持	
PBPRL	初始化	保持	保持	—	保持	I/O
PEDRL	初始化	保持	保持	—	保持	
PEIORL	初始化	保持	保持	—	保持	
PECRL4	初始化	保持	保持	—	保持	
PECRL3	初始化	保持	保持	—	保持	PFC
PECRL2	初始化	保持	保持	—	保持	
PECRL1	初始化	保持	保持	—	保持	
PEPRL	初始化	保持	保持	—	保持	
IFCR	初始化	保持	保持	—	保持	I/O
						PFC

寄存器略称	加电复位	手动复位	软件待机	模块待机	睡眠	模块
PFDRL	初始化	保持	保持	—	保持	I/O
FRQCR	初始化 *1	保持	保持	—	保持	CPG
STBCR1	初始化	保持	保持	—	保持	低功耗
STBCR2	初始化	保持	保持	—	保持	
STBCR3	初始化	保持	保持	—	保持	
STBCR4	初始化	保持	保持	—	保持	
STBCR5	初始化	保持	保持	—	保持	
STBCR6	初始化	保持	保持	—	保持	
WTCNT	初始化 *1	保持	保持	—	保持	WDT
WTCSR	初始化 *1	保持	保持	—	保持	
OSCCR	初始化 *2	保持	保持 *3	—	保持	CPG
RAMCR	初始化	保持	保持	—	保持	低功耗
ADTSR_0	初始化	保持	保持	保持	保持	A/D
ICR0	初始化	初始化	保持	—	保持	INTC
IRQCR	初始化	初始化	保持	—	保持	
IRQSR	初始化	初始化	保持	—	保持	
IPRA	初始化	初始化	保持	—	保持	
IPRB	初始化	初始化	保持	—	保持	
IPRC	初始化	初始化	保持	—	保持	
IPRD	初始化	初始化	保持	—	保持	
IPRE	初始化	初始化	保持	—	保持	
IPRF	初始化	初始化	保持	—	保持	
IPRH	初始化	初始化	保持	—	保持	
IPRI	初始化	初始化	保持	—	保持	
IPRJ	初始化	初始化	保持	—	保持	
IPRK	初始化	初始化	保持	—	保持	
IPRL	初始化	初始化	保持	—	保持	
IPRM	初始化	初始化	保持	—	保持	
BARA	初始化	保持	保持	初始化	保持	UBC
BAMRA	初始化	保持	保持	初始化	保持	
BBRA	初始化	保持	保持	初始化	保持	
BDRA	初始化	保持	保持	初始化	保持	
BDMRA	初始化	保持	保持	初始化	保持	
BARB	初始化	保持	保持	初始化	保持	
BAMRB	初始化	保持	保持	初始化	保持	
BBRB	初始化	保持	保持	初始化	保持	
BDRB	初始化	保持	保持	初始化	保持	
BDMRB	初始化	保持	保持	初始化	保持	
BRCR	初始化	保持	保持	初始化	保持	
BRSR	初始化	保持	保持	初始化	保持	
BRDR	初始化	保持	保持	初始化	保持	
BETR	初始化	保持	保持	初始化	保持	

【注】 *1 在通过 WDT 进行加电复位时，不进行初始化。

*2 在通过 WDT 进行加电复位时，OSCSTOP 位不被初始化。

*3 OSCSTOP 位被初始化。

第 21 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的情况。

21.1 绝对最大额定值

绝对最大额定值如表 21.1 所示。

表 21.1 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（模拟输入管脚除外）		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_{ref}	$-0.3 \sim AV_{CC}+0.3$	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
	工业用产品		$-40 \sim +85$	°C
保存温度		T_{stg}	$-55 \sim +125$	°C

【使用注意事项】

如果使用 LSI 时超过绝对最大额定值，就会产生 LSI 的永久性破坏。

21.2 DC 特性

DC 特性如表 21.2 和表 21.3 所示。

表 21.2 DC 特性

条件: $V_{CC}=AV_{CC}=4.0V \sim 5.5V$ 、 $V_{SS}=PLL_{V_{SS}}=AV_{SS}=0V$ 、
 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业用产品)

项 目	符号	Min.	Typ.	Max.	单位	测量条件
输入高电平的电压 (施密特触发的输入管脚除外)	$\overline{RES}$ 、 $\overline{MRES}$ 、 $\overline{NMI}$ 、 $\overline{FWE}$ 、 $\overline{MD1}$ 、 $\overline{ASEMD0}$ 、 $\overline{EXTAL}$	$V_{CC}-0.7$	—	$V_{CC}+0.3$	V	
	模拟兼用端口	2.2	—	$AV_{CC}+0.3$	V	
	其它的输入管脚	2.2	—	$V_{CC}+0.3$	V	
输入低电平的电压 (施密特触发的输入管脚除外)	$\overline{RES}$ 、 $\overline{MRES}$ 、 $\overline{NMI}$ 、 $\overline{FWE}$ 、 $\overline{MD1}$ 、 $\overline{ASEMD0}$ 、 $\overline{EXTAL}$	-0.3	—	0.5	V	
	其它的输入管脚	-0.3	—	0.8	V	
施密特触发的输入电压	$IRQ3 \sim IRQ0$ 、 $\overline{POE8}$ 、 $\overline{POE3}$ 、 $\overline{POE1}$ 、 $\overline{POE0}$ 、 $\overline{TCLKA} \sim \overline{TCLKD}$ 、 $\overline{TIOC0A} \sim \overline{TIOC0D}$ 、 $\overline{TIOC1A}$ 、 $\overline{TIOC1B}$ 、 $\overline{TIOC2A}$ 、 $\overline{TIOC2B}$ 、 $\overline{TIOC3A} \sim \overline{TIOC3D}$ 、 $\overline{TIOC4A} \sim \overline{TIOC4D}$ 、 $\overline{TIC5U}$ 、 $\overline{TIC5V}$ 、 $\overline{TIC5W}$ 、 $\overline{TIC5US}$ 、 $\overline{TIC5VS}$ 、 $\overline{TIC5WS}$ 、 $\overline{SCK0} \sim \overline{SCK3}$ 、 $\overline{RXD0} \sim \overline{RXD3}$	V_{T+}	$V_{CC}-0.5$	—	V	
		V_{T-}	—	1.0	V	
		$V_{T+}-V_{T-}$	0.4	—	V	$V_{CC}=4.5V \sim 5.5V$
输入漏泄电流	全部的输入管脚 ($\overline{ASEMD0}$ 除外)	I_{in}	—	1.0	μA	
输入上拉 MOS 电流	$\overline{ASEMD0}$ 、 $\overline{POE3}$	$-I_{pu}$	—	800	μA	$V_{in}=0V$
三态漏泄电流 (off 状态)	端口 A、B、E	I_{tsi}	—	1.0	μA	
输出高电平的电压	PE9、PE11 ~ PE15	V_{OH}	$V_{CC}-0.8$	—	V	$I_{OH}=-5mA$
	$\overline{WDTOVF}$		$V_{CC}-0.5$	—	V	$I_{OH}=-100\mu A$
	其它的全部输出管脚		$V_{CC}-0.5$	—	V	$I_{OH}=-200\mu A$
			$V_{CC}-1.0$	—	V	$I_{OH}=-1mA$
			$V_{CC}-1.5$	—	V	$I_{OH}=-2mA$ (参考值)
输出低电平的电压	PE9、PE11 ~ PE15	V_{OL}	—	1.4	V	$I_{OL}=15mA$
	其它的全部输出管脚		—	0.4	V	$I_{OL}=1.6mA$

项 目		符号	Min.	Typ.	Max.	单位	测量条件
输入电容	全部输入管脚	C _{in}	—	—	20	pF	V _{in} =0V、f=1MHz、T _a =25°C
消费电流	通常运行时	I _{CC}	—	52	70	mA	I _φ =50MHz
	睡眠时		—	33	50	mA	I _φ =50MHz
	软件待机时		—	—	5	mA	T _a ≤ 50°C
			—	—	15	mA	50°C < T _a
模拟电源电流	A/D 转换中	AI _{CC}	—	3	5	mA	每个模块
	A/D 转换待机时		—	—	2	mA	每个模块
	待机时		—	—	15	μA	

【使用注意事项】

1. 在不使用 A/D 转换器时，不能将 AV_{CC} 和 AV_{SS} 管脚置为开路。
2. 消费电流是在 V_{IH} (Min.) = $V_{CC}-0.5V$ 、 V_{IL} (Max.) = $0.5V$ 的条件下全部输出管脚为无负载状态时的值。

表 21.3 输出容许电流值

条件: $V_{CC}=AV_{CC}=4.0V \sim 5.5V$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=0V$ 、
 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业用产品)

项 目	符号	Min.	Typ.	Max.	单位
输出低电平的容许电流 (每个管脚)	I_{OL}	—	—	2.0*	mA
输出低电平的容许电流 (总和)	ΣI_{OL}	—	—	80	mA
输出高电平的容许电流 (每个管脚)	$-I_{OH}$	—	—	2.0*	mA
输出高电平的容许电流 (总和)	$\Sigma -I_{OH}$	—	—	25	mA

【使用注意事项】为了确保 LSI 的可靠性，输出的电流值不能超过表 21.3 中的值。

【注】 * PE9、PE11 ~ PE15 为 $I_{OL}=15mA$ (Max.) / $-I_{OH}=5mA$ (Max.)。但是，这些管脚中同时超过 2.0mA 流过 I_{OL} / $-I_{OH}$ 的管脚只能在 3 个以内。

21.3 AC 特性

原则上输入到本 LSI 的信号是时钟同步输入信号。只要没有特殊理由，必须保持各输入信号的准备和保持时间。

表 21.4 最大工作频率

条件: $V_{CC}=AV_{CC}=4.0V \sim 5.5V$ 、 $V_{SS}=PLL V_{SS}=AV_{SS}=0V$ 、
 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业用产品)

项 目		符号	Min.	Typ.	Max.	单位	备注
工作频率	CPU ($I\phi$)	f	10	—	50	MHz	
	外围模块 ($P\phi$)		10	—	40		

21.3.1 时钟时序

表 21.5 时钟时序

条件: $V_{CC}=AV_{CC}=4.0V \sim 5.5V$ 、 $V_{SS}=PLL V_{SS}=AV_{SS}=0V$ 、
 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业用产品)

项 目	符号	Min.	Max.	单位	参照图
EXTAL 时钟的输入频率	f_{EX}	10	12.5	MHz	图 21.1
EXTAL 时钟的输入周期时间	t_{EXcyc}	80	100	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 ($B\phi$) 时钟频率 (参考值)	f_{OP}	10	40	MHz	*
CK ($B\phi$) 时钟周期时间 (参考值)	t_{cyc}	25	100	ns	
加电振荡稳定时间	t_{OSC1}	10	—	ms	图 21.2
待机返回的振荡稳定时间 1	t_{OSC2}	10	—	ms	图 21.3
待机返回的振荡稳定时间 2	t_{OSC3}	10	—	ms	图 21.4

【注】 * 取决于频率控制寄存器 (FRQCR) 的设定。

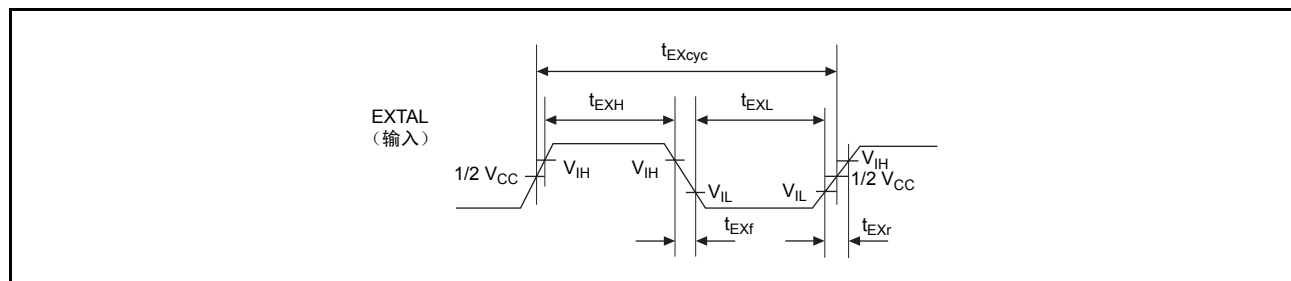


图 21.1 EXTAL 时钟的输入时序

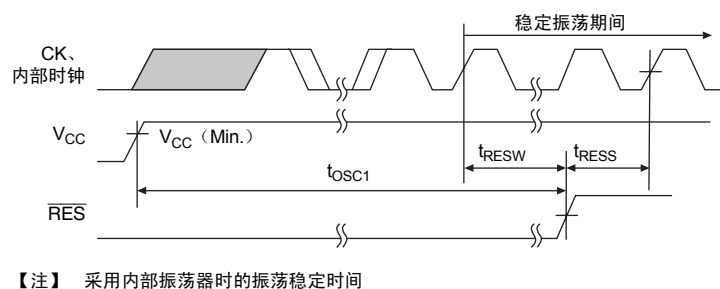


图 21.2 加电振荡稳定时间

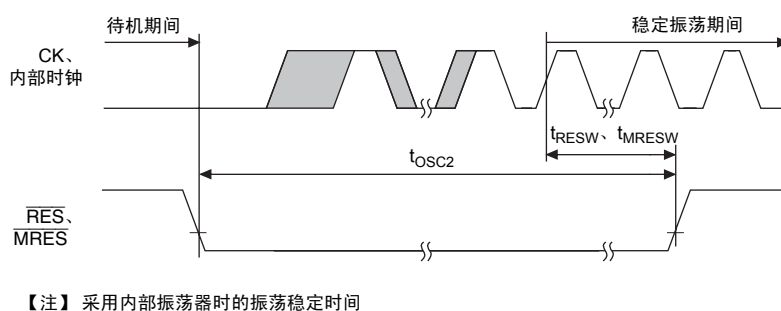


图 21.3 待机返回时的振荡稳定时间（通过复位的返回）

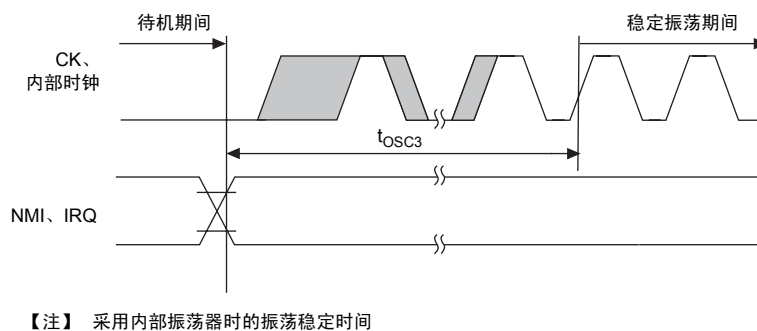


图 21.4 待机返回时的振荡稳定时间（通过 NMI、IRQ 的返回）

21.3.2 控制信号的时序

表 21.6 控制信号的时序

条件: $V_{CC}=AV_{CC}=4.0V \sim 5.5V$ 、 $V_{SS}=PLL V_{SS}=AV_{SS}=0V$ 、
 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业用产品)

项 目	符号	Min.	Max.	单位	参照图
$\overline{RES}$ 脉宽	t_{RESW}	20^{*2}	—	t_{Bcyc}^{*4}	图 21.2
$\overline{RES}$ 准备时间 *1 (参考值)	t_{RESS}	65	—	ns	图 21.3
$\overline{RES}$ 保持时间 (参考值)	t_{RESH}	15	—	ns	图 21.5
$\overline{MRES}$ 脉宽	t_{MRESW}	20^{*3}	—	t_{Bcyc}^{*4}	图 21.6
$\overline{MRES}$ 准备时间 *1 (参考值)	t_{MRESS}	25	—	ns	
$\overline{MRES}$ 保持时间 (参考值)	t_{MRESH}	15	—	ns	
MD1 准备时间	t_{MDS}	20	—	t_{Bcyc}^{*4}	图 21.5
NMI 准备时间 *1 (参考值)	t_{NMIS}	60	—	ns	图 21.6
NMI 保持时间 (参考值)	t_{NMIH}	10	—	ns	
IRQ3 ~ IRQ0 准备时间 *1 (参考值)	t_{IRQS}	35	—	ns	
IRQ3 ~ IRQ0 保持时间 (参考值)	t_{IRQH}	35	—	ns	
IRQOUT 输出延迟时间 (参考值)	t_{IRQOD}	—	100	ns	图 21.7

【注】 *1 $\overline{RES}$ 、 $\overline{MRES}$ 、NMI 和 IRQ3 ~ IRQ0 信号为异步输入。如果满足在此所示的准备时间，就在时钟上升沿检测到信号的变化；否则，信号变化的检测就会推迟到下一个时钟上升沿为止。

*2 在待机模式时，为 $t_{RESW}=t_{OSC2}$ (10ms)。

*3 在待机模式时，为 $t_{MRESW}=t_{OSC2}$ (10ms)。

*4 t_{Bcyc} 表示总线时钟 ($B\phi=CK$) 的周期。

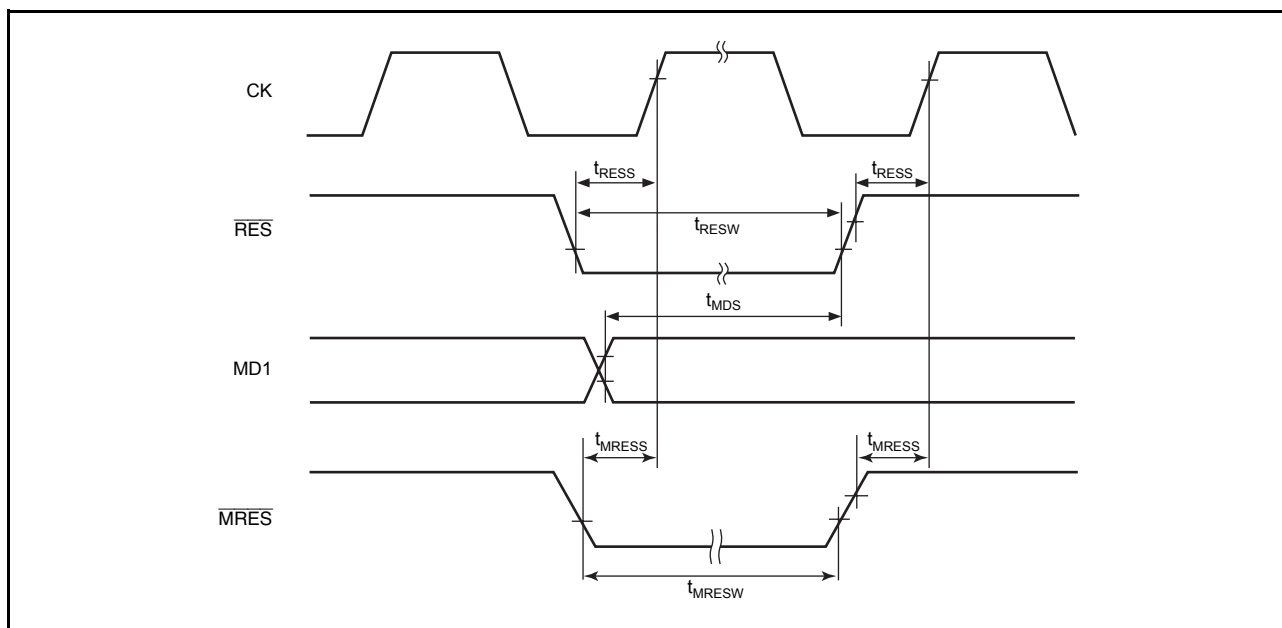


图 21.5 复位的输入时序

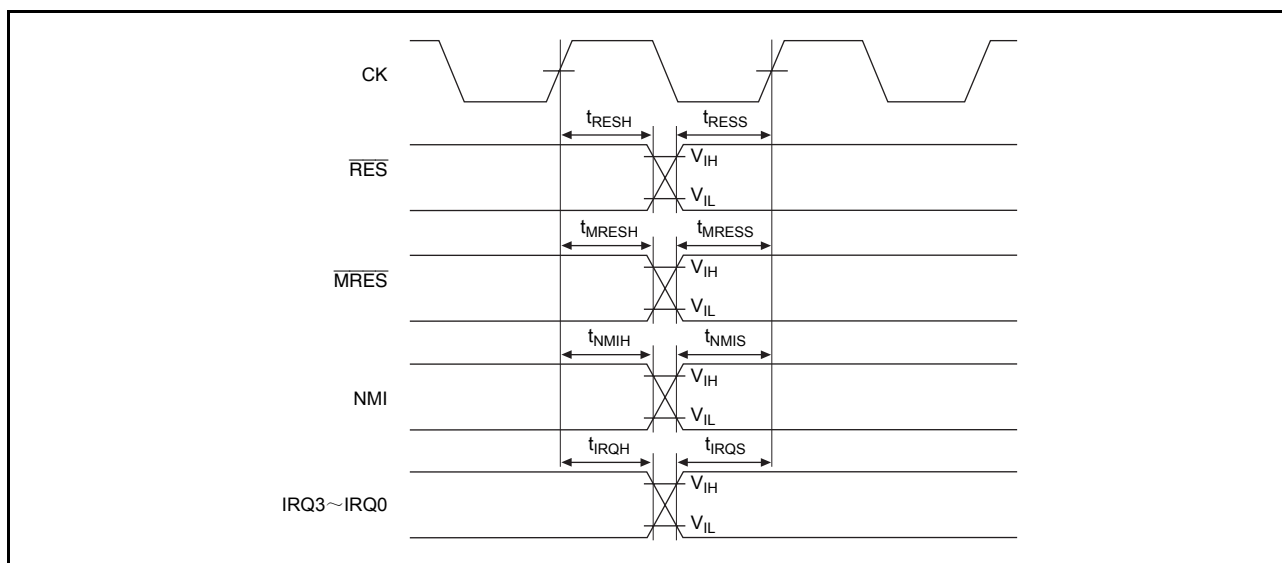


图 21.6 中断信号的输入时序

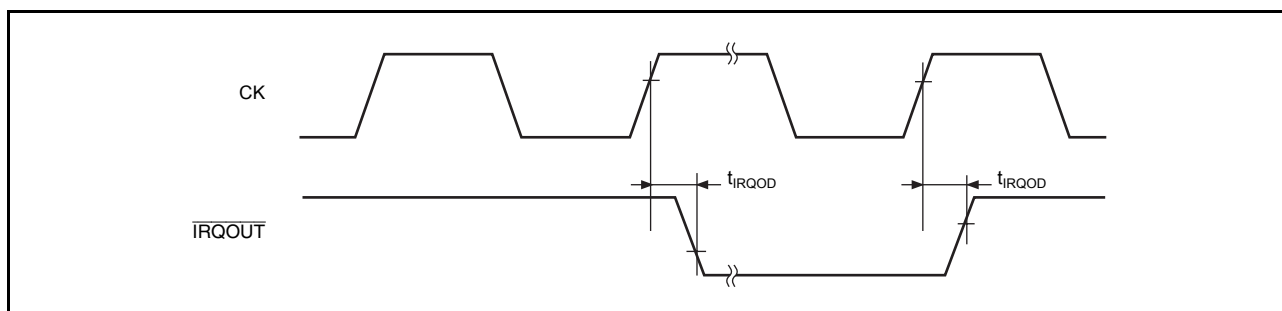


图 21.7 中断信号的输出时序

21.3.3 多功能定时器脉冲单元 2 (MTU2) 时序

表 21.7 多功能定时器脉冲单元 2 (MTU2) 的时序

条件: $V_{CC}=AV_{CC}=4.0V \sim 5.5V$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=0V$ 、
 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业用产品)

项 目	符号	Min.	Max.	单位	参照图
输出比较的输出延迟时间 (参考值)	t_{TOCD}	—	50	ns	图 21.8
输入捕捉的输入准备时间 (参考值)	t_{TICS}	20	—	ns	
输入捕捉的输入脉宽 (指定单边沿时)	$t_{TICWH/L}$	1.5	—	t_{MPcyc}	
输入捕捉的输入脉宽 (指定双边沿时)	$t_{TICWH/L}$	2.5	—	t_{MPcyc}	
定时器的输入准备时间 (参考值)	t_{TCKS}	20	—	ns	图 21.9
定时器的时钟脉宽 (指定单边沿时)	$t_{TCKWH/L}$	1.5	—	t_{MPcyc}	
定时器的时钟脉宽 (指定双边沿时)	$t_{TCKWH/L}$	2.5	—	t_{MPcyc}	
定时器的时钟脉宽 (位相计数模式)	$t_{TCKWH/L}$	2.5	—	t_{MPcyc}	

【注】 t_{MPcyc} 表示 MTU2 时钟 (MP ϕ) 的周期。

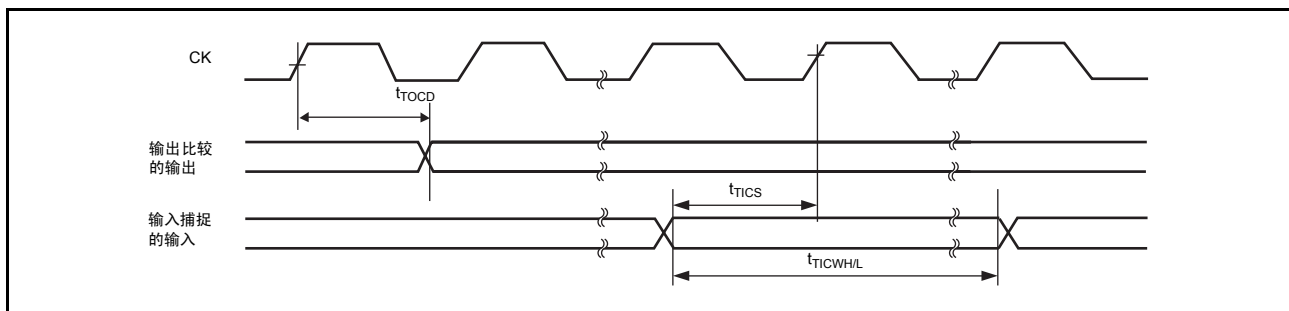


图 21.8 MTU2 的输入 / 输出时序

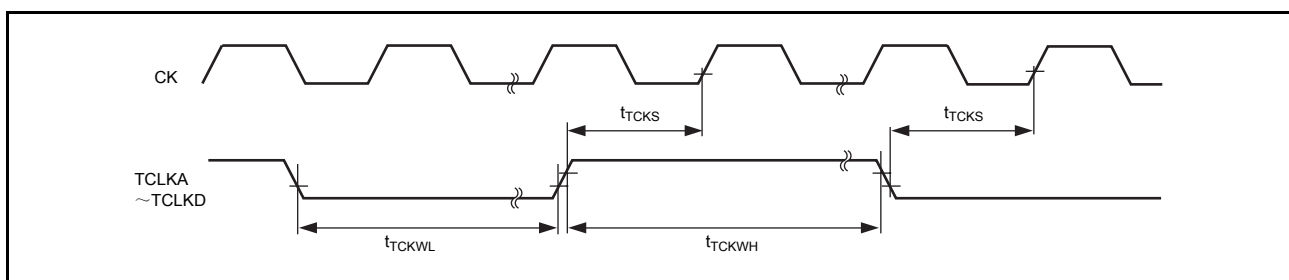


图 21.9 MTU2 时钟的输入时序

21.3.4 I/O 端口的时序

表 21.8 I/O 端口的时序

条件: $V_{CC}=AV_{CC}=4.0V \sim 5.5V$ 、 $V_{SS}=PLL V_{SS}=AV_{SS}=0V$ 、
 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业用产品)

项 目	符号	Min.	Max.	单位	参照图
端口输出数据的延迟时间 (参考值)	t_{PWD}	—	50	ns	图 21.10
端口输入的低电平脉宽	t_{PRL}	2	—	t_{Pcyc}	
端口输入的高电平脉宽	t_{PRH}	2	—	t_{Pcyc}	

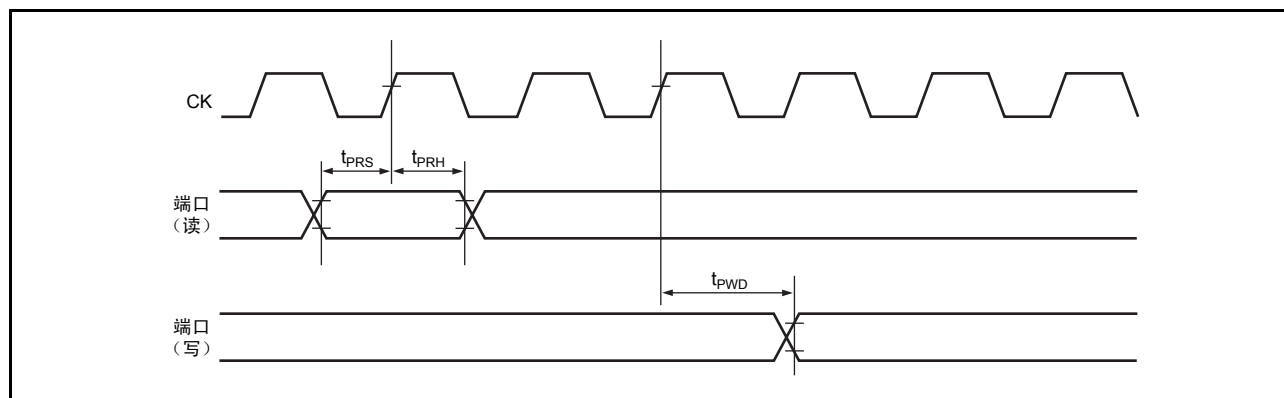


图 21.10 I/O 端口的输入 / 输出时序

21.3.5 看门狗定时器 (WDT) 的时序

表 21.9 看门狗定时器 (WDT) 的时序

条件: $V_{CC}=AV_{CC}=4.0V \sim 5.5V$ 、 $V_{SS}=PLL V_{SS}=AV_{SS}=0V$ 、
 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业用产品)

项 目	符号	Min.	Max.	单位	参照图
WDTOVF 延迟时间 (参考值)	t_{WOVD}	—	50	ns	图 21.11

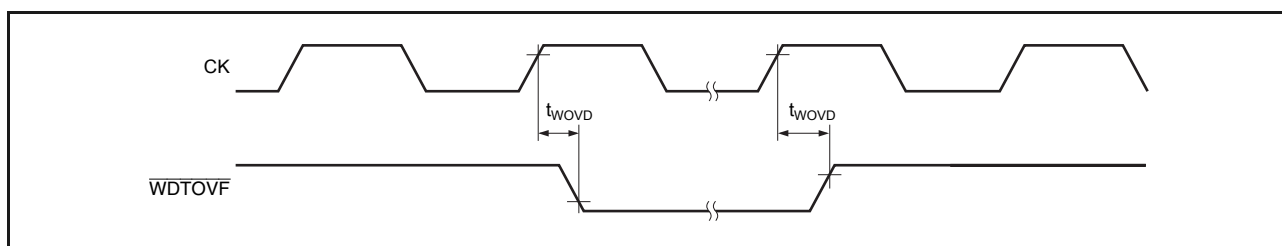


图 21.11 看门狗定时器的时序

21.3.6 串行通信接口（SCI）的时序

表 21.10 串行通信接口（SCI）的时序

条件: $V_{CC}=AV_{CC}=4.0V \sim 5.5V$ 、 $V_{SS}=PLL V_{SS}=AV_{SS}=0V$ 、
 $T_a=-20 \sim +85^{\circ}C$ （民用产品）、 $T_a=-40 \sim +85^{\circ}C$ （工业用产品）

项 目	符号	Min.	Max.	单位	参照图
输入时钟周期（异步）	t_{SCYC}	4	—	t_{PCYC}	图 21.12
输入时钟周期（时钟同步）	t_{SCYC}	6	—	t_{PCYC}	
输入时钟脉宽	t_{SCKW}	0.4	0.6	t_{SCYC}	
输入时钟的上升时间	t_{SCKR}	—	1.5	t_{PCYC}	
输入时钟的下降时间	t_{SCKF}	—	1.5	t_{PCYC}	
发送数据的延迟时间	异步	t_{TXD}	—	$4t_{PCYC}+10$	图 21.13
接收数据的准备时间		t_{RXS}	$4t_{PCYC}$	—	
接收数据的保持时间		t_{RXH}	$4t_{PCYC}$	—	
发送数据的延迟时间	时钟同步	t_{TXD}	—	$3t_{PCYC}+10$	
接收数据的准备时间		t_{RXS}	$3t_{PCYC}+50$	—	
接收数据的保持时间		t_{RXH}	$3t_{PCYC}+50$	—	

【注】 t_{PCYC} 表示外围时钟（Pφ）的周期。

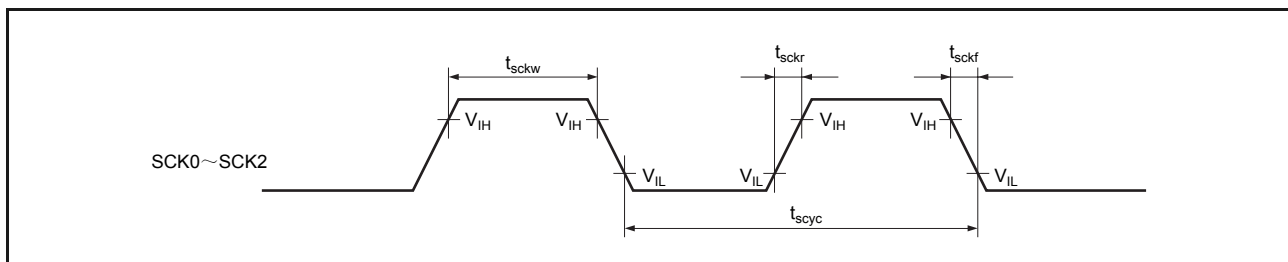


图 21.12 输入时钟的时序

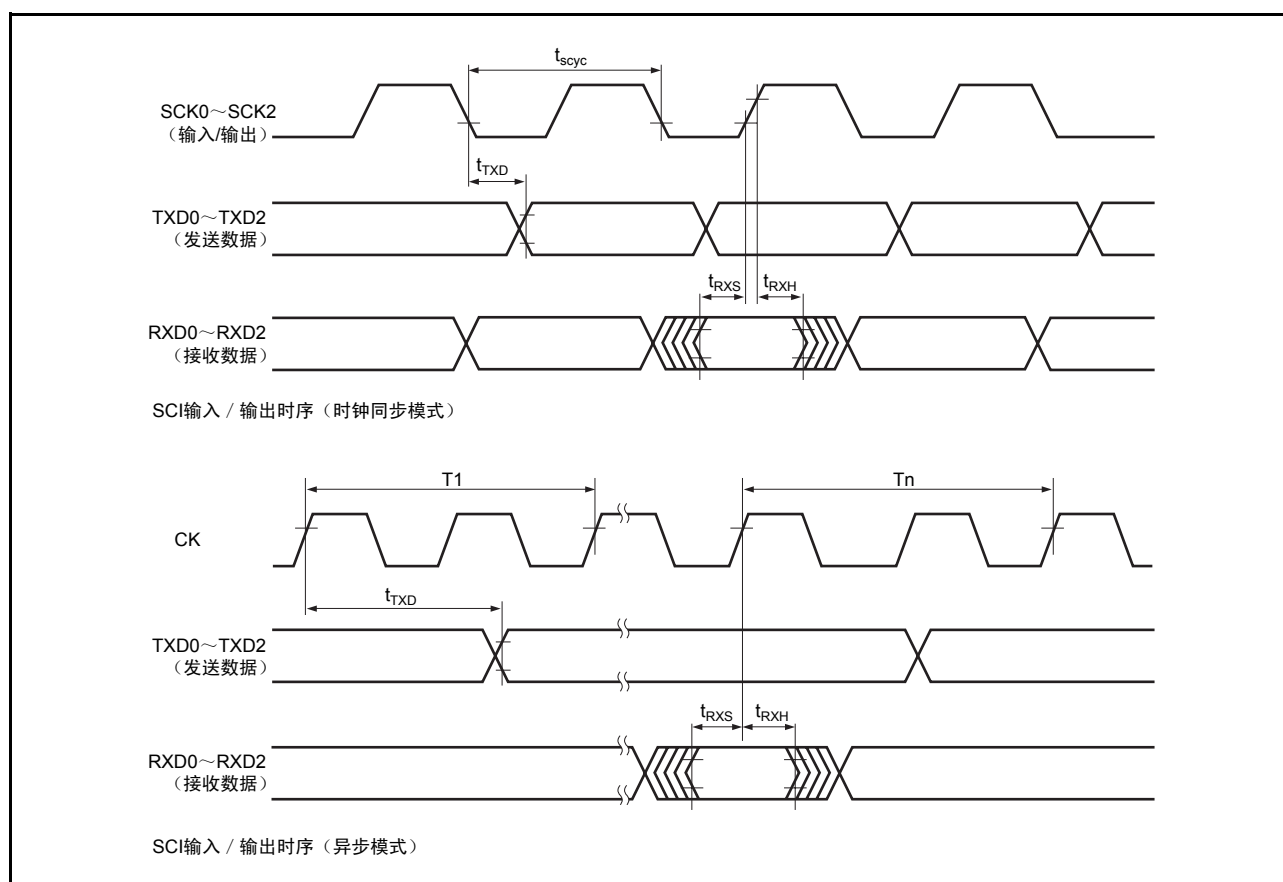


图 21.13 SCI 的输入 / 输出时序

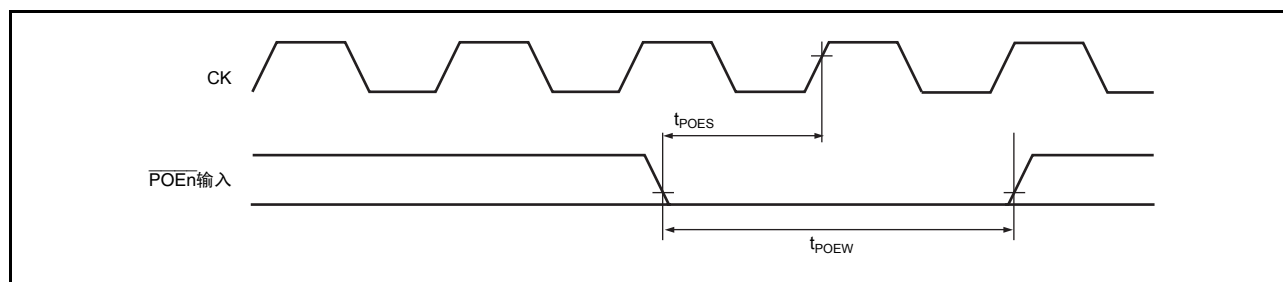
21.3.7 端口输出允许 (POE) 的时序

表 21.11 端口输出允许 (POE) 的时序

条件: $V_{CC}=AV_{CC}=4.0V \sim 5.5V$ 、 $V_{SS}=PLL_{VSS}=AV_{SS}=0V$ 、
 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业用产品)

项 目	符号	Min.	Max.	单位	参照图
$\overline{POE}$ 输入的准备时间 (参考值)	t_{POES}	50	—	ns	图 21.14
$\overline{POE}$ 输入脉宽	t_{POEW}	1.5	—	t_{pcyc}	

【注】 t_{pcyc} 表示外围时钟 ($P\phi$) 的周期。

图 21.14 $\overline{POE}$ 的输入时序

21.3.8 A/D 转换器的时序

表 21.12 A/D 转换器的时序

条件: $V_{CC}=AV_{CC}=4.0V \sim 5.5V$ 、 $V_{SS}=PLL V_{SS}=AV_{SS}=0V$ 、
 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业用产品)

项 目	符号	Min.	Typ.	Max.	单位	参照图
外部触发输入的开始延迟时间 (参考值)	t_{TRGS}	25	—	—	ns	图 21.15

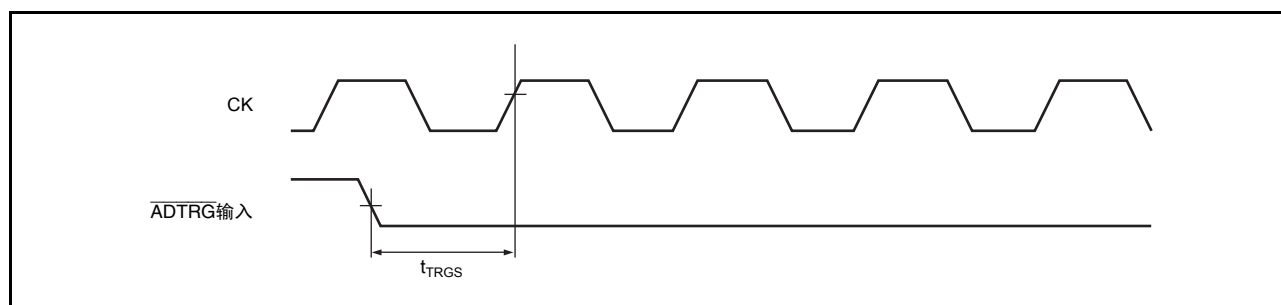


图 21.15 外部触发的输入时序

21.3.9 AC 特性测量条件

- 输入信号电平: V_{IL} (Max.) / V_{IH} (Min.)
- 输出信号参照电平: 高电平: 2.0V、低电平: 0.8V

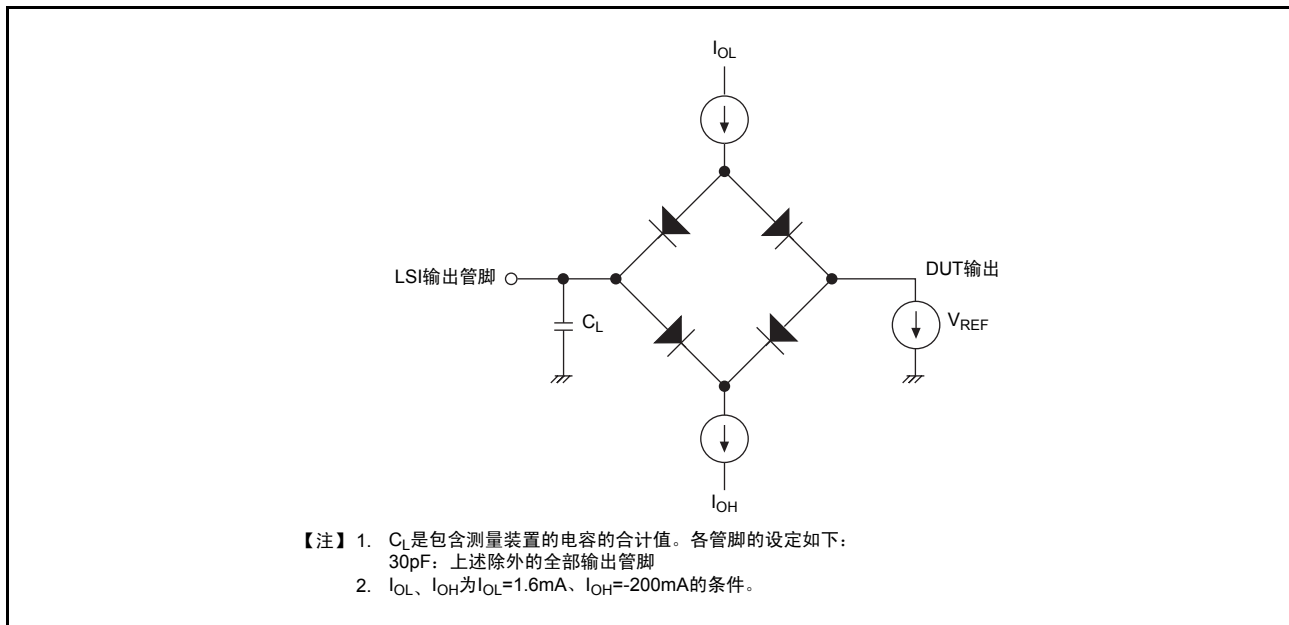


图 21.16 输出负载电路

21.4 A/D 转换器特性

表 21.13 A/D 转换器特性

条件: $V_{CC}=AV_{CC}=4.0V \sim 5.5V$ 、 $V_{SS}=PLL V_{SS}=AV_{SS}=0V$ 、
 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业用产品)

项 目	Min.	Typ.	Max.	单位
分辨率	10	10	10	位
转换时间	—	—	2.0	μs
模拟输入电容	—	—	20	pF
允许信号源阻抗	—	—	$1^{*2}/3^{*1}$	$k\Omega$
非线性误差	—	—	$\pm 3.0^{*1}/\pm 5.0^{*2}$	LSB
偏移误差	—	—	$\pm 3.0^{*1}/\pm 5.0^{*2}$	LSB
满刻度误差	—	—	$\pm 3.0^{*1}/\pm 5.0^{*2}$	LSB
量化误差	—	—	± 0.5	LSB
绝对精度	—	—	$\pm 4.0^{*1}/\pm 6.0^{*2}$	LSB

【注】 *1 转换时间 $\geq 4.0\mu s$ 时

*2 转换时间 $< 4.0\mu s$ 时

21.5 闪存特性

表 21.14 闪存特性

条件: $V_{CC}=AV_{CC}=4.0V \sim 5.5V$ 、 $V_{SS}=PLL V_{SS}=AV_{SS}=0V$ 、
 $T_a=-20 \sim +85^{\circ}C$ (民用产品)、 $T_a=-40 \sim +85^{\circ}C$ (工业用产品)

项 目	符号	Min.	Typ.	Max.	单位
编程时间 *1*2*4	t_P	—	1	10	ms/128 字节
擦除时间 *1*2*4	t_E	—	40	130	ms/4K 字节块
		—	300	800	ms/32K 字节块
		—	600	1500	ms/64K 字节块
编程时间 (总和) *1*2*4	Σt_P	—	1.2	3	s/128K 字节
		—	0.6	1.5	s/64K 字节
擦除时间 (总和) *1*2*4	Σt_E	—	1.3	3.5	s/128K 字节
		—	0.7	2	s/64K 字节
编程、擦除时间 (总和) *1*2*4	Σt_{PE}	—	2.5	6.5	s/128K 字节
		—	1.3	3.5	s/64K 字节
改写次数	N_{WEC}	100*3	—	—	次

【注】 *1 编程和擦除时间取决于数据。

*2 编程和擦除时间不包含数据的传送时间。

*3 是保证改写后所有特性的 Min. 次数 (保证范围: 1 ~ Min. 值)。

*4 是在包含 Min. 值的使用范围内进行改写时的特性。

21.6 使用注意事项

21.6.1 V_{CL} 电容连接方法

本 LSI 内置内部降压电路，将单片机内部的电源电压自动下降到适合的电平。需要在此内部降压电源（ V_{CL} 管脚）和 V_{SS} 管脚之间连接用于稳定内部电压的电容（ $0.1 \sim 0.47\mu\text{F}$ ）。外接电容的连接方法如图 21.17 所示。外接电容必须配置在管脚的附近，不能给 V_{CL} 管脚外加电源电压。

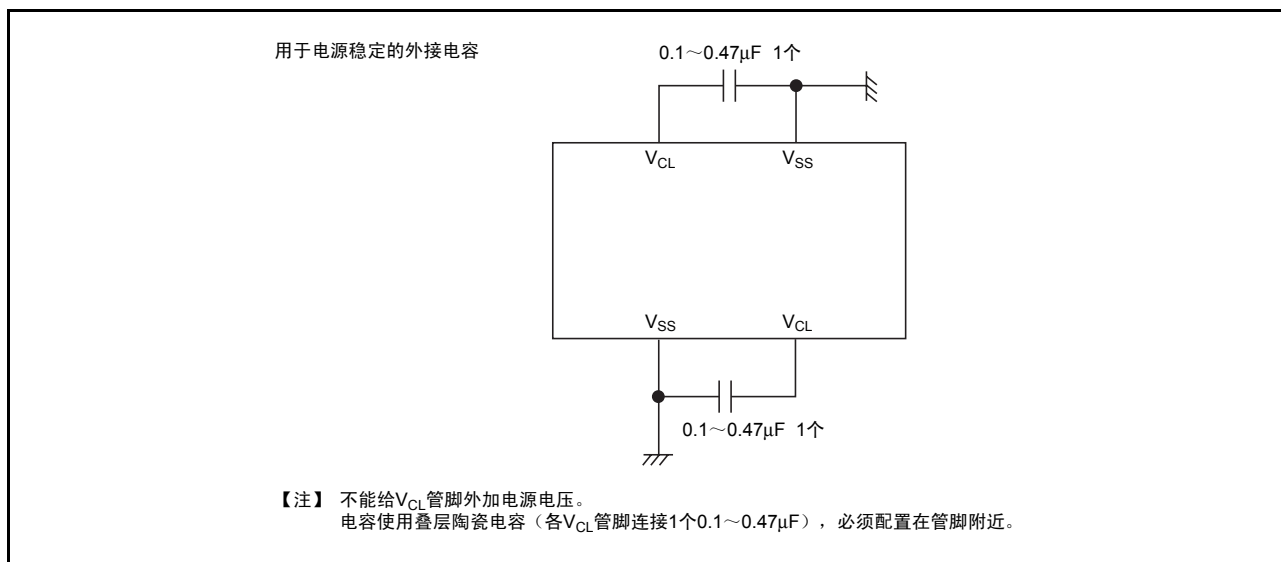


图 21.17 V_{CL} 电容的连接方法

附录

附录 A. 管脚状态

管脚的初始值因 MCU 运行模式而不同，详细内容请参照“第 15 章 管脚功能控制器（PFC）”。

表 A.1 管脚状态（SH7125）

管脚功能		管脚状态					
分类	管脚名	复位状态		低功耗状态		检测到振荡停止时	使用 POE 功能时
		加电	手动	软件待机	睡眠		
时钟	XTAL	O	O	L	O	O	O
	EXTAL	I	I	I	I	I	I
系统控制	RES	I	I	I	I	I	I
	MRES	Z	I	Z	I	Z	I
	WDTOVF	O*2	O	O	O	O	O
运行模式控制	MD1	I	I	I	I	I	I
	ASEMD0	I*3	I*3	I*3	I*3	I*3	I*3
	FWE	I	I	I	I	I	I
中断	NMI	I	I	I	I	I	I
	IRQ0 ~ IRQ3	Z	I	I	I	I	I
	IRQOUT	Z	O	Z	O	Z	O
MTU2	TCLKA ~ TCLKD	Z	I	Z	I	I	I
	TIOC0A ~ TIOC0D	Z	I/O	K*1	I/O	I/O	Z
	TIOC1A、TIOC1B	Z	I/O	K*1	I/O	I/O	I/O
	TIOC2A、TIOC2B	Z	I/O	K*1	I/O	I/O	I/O
	TIOC3A、TIOC3C	Z	I/O	K*1	I/O	I/O	I/O
	TIOC3B、TIOC3D	Z	I/O	Z	I/O	Z	Z
	TIOC4A ~ TIOC4D	Z	I/O	Z	I/O	Z	Z
	TIC5U、TIC5V、TIC5W	Z	I	Z	I	I	I
POE	POE0、POE1、POE8	Z	I	Z	I	I	I
	POE3	I*3	I*3	Z	I*3	I*3	I*3
SCI	SCK0 ~ SCK2	Z	I/O	Z	I/O	I/O	I/O
	RXD0 ~ RXD2	Z	I	Z	I	I	I
	TXD0 ~ TXD2	Z	O	O*1	O	O	O
A/D 转换器	AN0 ~ AN7	Z	I	Z	I	I	I
	ADTRG	Z	I	Z	I	I	I
I/O 端口	PA0 ~ PA15	Z	I/O	K*1	I/O	I/O	I/O
	PB1 ~ PB3、PB5、PB16	Z	I/O	K*1	I/O	I/O	I/O
	PE0 ~ PE3	Z	I/O	K*1	I/O	I/O	Z
	PE4 ~ PE8、PE10	Z	I/O	K*1	I/O	I/O	I/O
	PE9、PE11 ~ PE15	Z	I/O	Z	I/O	Z	Z
	PF0 ~ PF7	Z	I	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 : 输入管脚为高阻抗, 输出管脚为状态保持

【注】 *1 如果将待机控制寄存器 6 (STBCR6) 的 HIZ 位置为 1, 输出管脚就为高阻抗状态。

*2 在加电复位中为输入状态。为了防止误动作, 必须进行上拉。需要进行下拉时, 必须用大于等于 1MΩ 的电阻进行下拉。

*3 什么也不输入时, 在内部进行上拉。

表 A.2 管脚状态 (SH7124)

管脚功能		管脚状态					
分类	管脚名	复位状态		低功耗状态		检测到振荡停止时	使用 POE 功能时
		加电	手动	软件待机	睡眠		
时钟	XTAL	O	O	L	O	O	O
	EXTAL	I	I	I	I	I	I
系统控制	RES	I	I	I	I	I	I
	MRES	Z	I	Z	I	Z	I
	WDTOVF	O*2	O	O	O	O	O
运行模式控制	MD1	I	I	I	I	I	I
	ASEMD0	I*3	I*3	I*3	I*3	I*3	I*3
	FWE	I	I	I	I	I	I
中断	NMI	I	I	I	I	I	I
	IRQ1 ~ IRQ3	Z	I	I	I	I	I
	IRQOUT	Z	O	Z	O	Z	O
MTU2	TCLKA ~ TCLKD	Z	I	Z	I	I	I
	TIOC0A ~ TIOC0D	Z	I/O	K*1	I/O	I/O	Z
	TIOC3A、TIOC3C	Z	I/O	K*1	I/O	I/O	I/O
	TIOC3B、TIOC3D	Z	I/O	Z	I/O	Z	Z
	TIOC4A ~ TIOC4D	Z	I/O	Z	I/O	Z	Z
	TIC5U、TIC5V、TIC5W	Z	I	Z	I	I	I
POE	POE0、POE1、POE8	Z	I	Z	I	I	I
SCI	SCK0、SCK2	Z	I/O	Z	I/O	I/O	I/O
	RXD0 ~ RXD2	Z	I	Z	I	I	I
	TXD0 ~ TXD2	Z	O	O*1	O	O	O
A/D 转换器	AN0 ~ AN7	Z	I	Z	I	I	I

管脚功能		管脚状态					
分类	管脚名	复位状态		低功耗状态		检测到振荡停止时	使用 POE 功能时
		加电	手动	软件待机	睡眠		
I/O 端口	PA0、PA1、PA3、PA4、PA6 ~ PA9	Z	I/O	K*1	I/O	I/O	I/O
	PB1、PB3、PB5	Z	I/O	K*1	I/O	I/O	I/O
	PE0 ~ PE3	Z	I/O	K*1	I/O	I/O	Z
	PE8、PE10	Z	I/O	K*1	I/O	I/O	I/O
	PE9、PE11 ~ PE15	Z	I/O	Z	I/O	Z	Z
	PF0 ~ PF7	Z	I	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 : 输入管脚为高阻抗, 输出管脚为状态保持

【注】 *1 如果将待机控制寄存器 6 (STBCR6) 的 HIZ 位置为 1, 输出管脚就为高阻抗状态。

*2 在加电复位中为输入状态。为了防止误动作, 必须进行上拉。需要进行下拉时, 必须用大于等于 1MΩ 的电阻进行下拉。

*3 什么也不输入时, 在内部进行上拉。

附录 B. 型号一览

表 B.1 型号一览

产品分类			产品型号	封装（封装代码）
SH7125	闪存版 （内置 128KB 的产品）	民用	R5F71253N64FP	LQFP-64（FP-64K）
		工业用	R5F71253D64FP	
		民用	R5F71253N64FA	QFP-64（FP-64A）
		工业用	R5F71253D64FA	
	闪存版 （内置 64KB 的产品）	民用	R5F71252N64FP	LQFP-64（FP-64K）
		工业用	R5F71252D64FP	
		民用	R5F71252N64FA	QFP-64（FP-64A）
		工业用	R5F71252D64FA	
SH7124	闪存版 （内置 128KB 的产品）	民用	R5F71243N48FP	LQFP-48（FP-48F）
		工业用	R5F71243D48FP	
	闪存版 （内置 64KB 的产品）	民用	R5F71242N48FP	
		工业用	R5F71242D48FP	

附录 C. 外形尺寸图

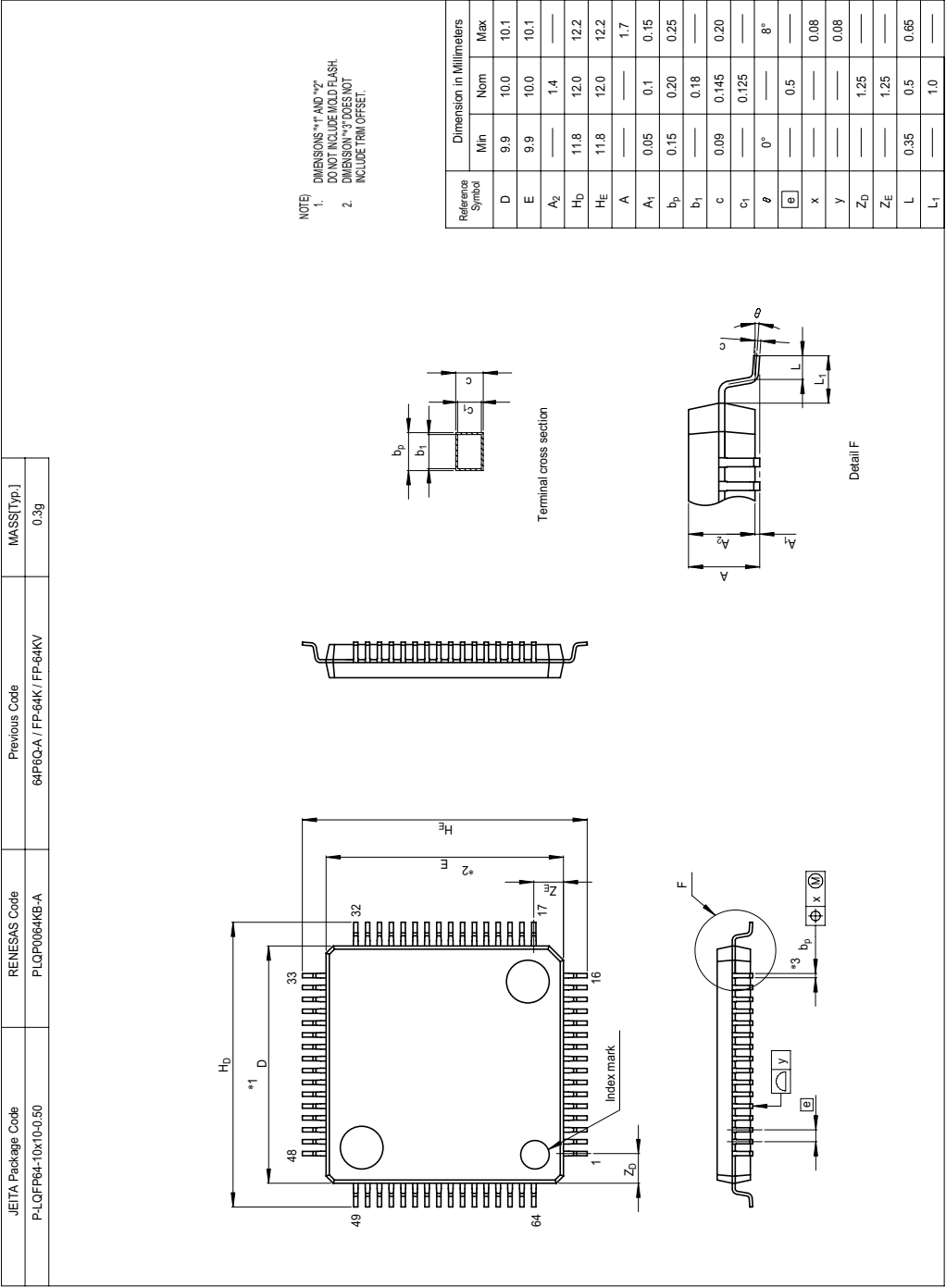


图 C.1 LQFP-64

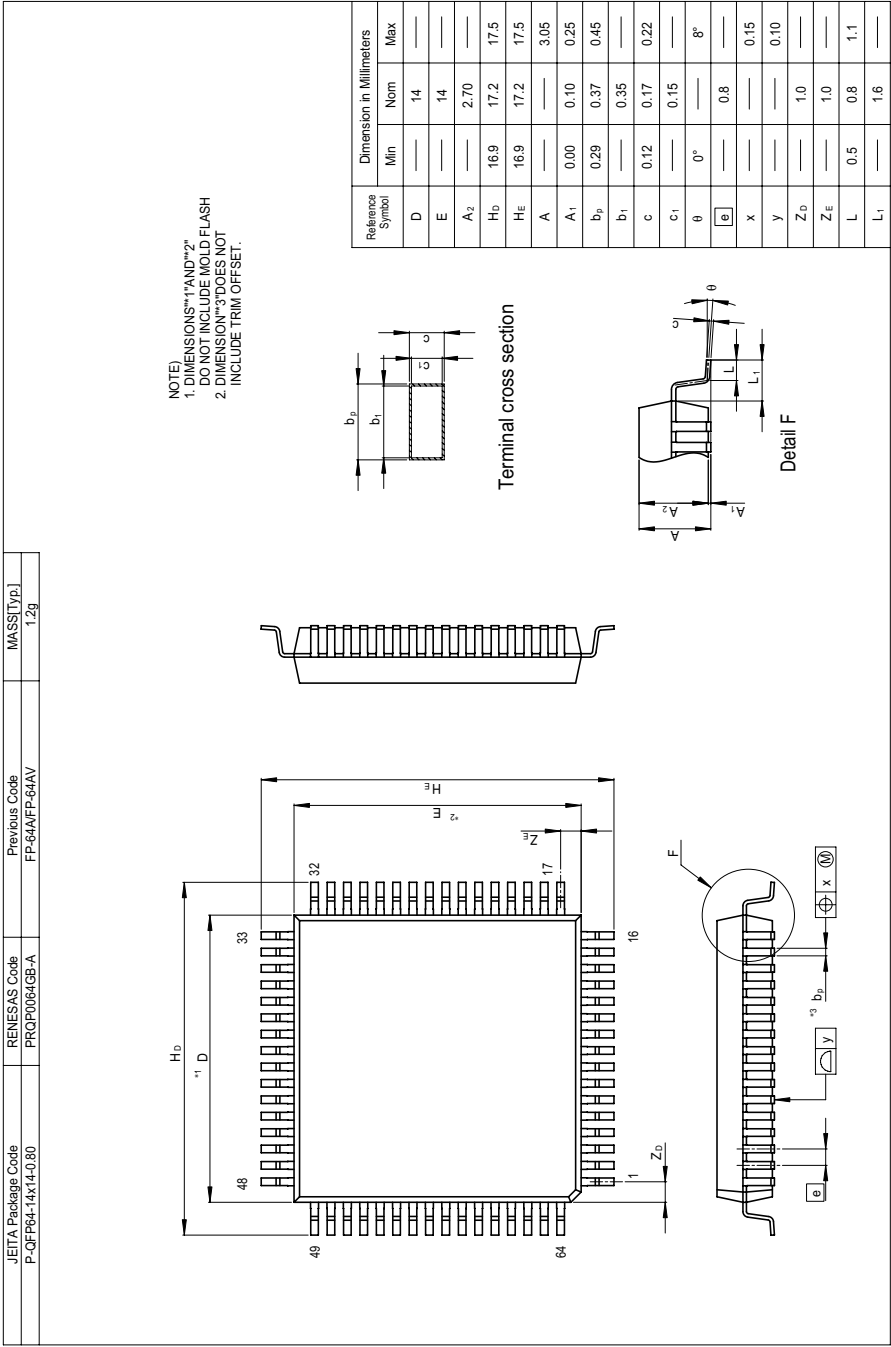


图 C.2 QFP-64

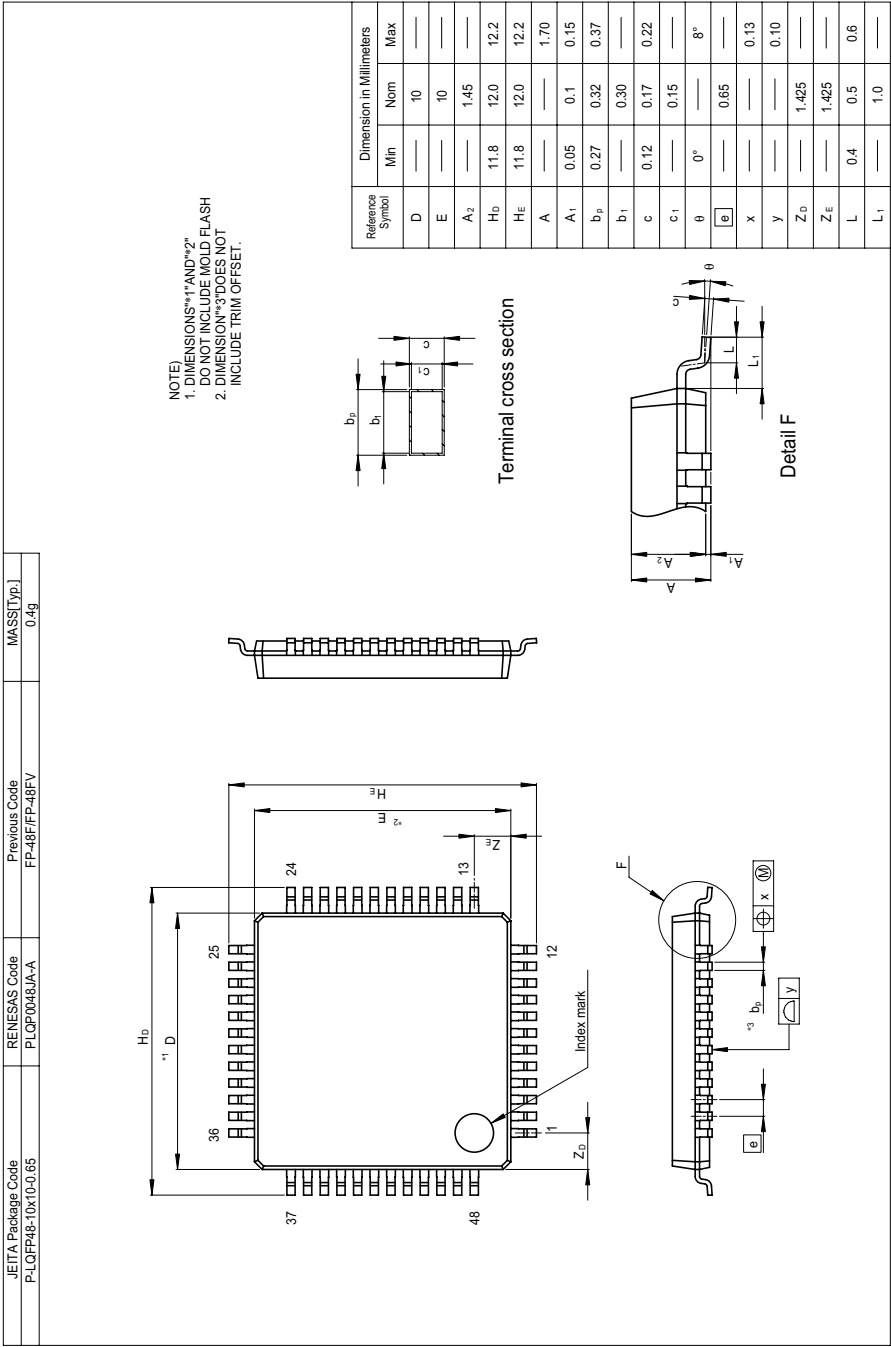


图 C.3 LQFP-48

索引

A

A/D 转换开始请求的延迟功能	216
A/D 转换器的启动	223
A/D 转换器的中断源	355
A/D 转换器特性	517
A/D 转换器 (ADC)	342
A/D 转换时间	354
AC 特性	505
AC 特性测量条件	516

B

板上编程模式	434
比较匹配定时器 (CMT)	360
编程器模式	467

C

CMT 的中断源	365
CPU	10
槽非法指令	53
槽非法指令异常处理的注意事项	56
乘加寄存器 (MACH 和 MACL)	13
程序计数器 (PC)	13
程序执行状态	31
串行通信接口 (SCI)	299
错误保护	445

D

DC 特性	503
单通道模式	351
单芯片模式	32
单周期扫描模式	352
低功耗模式	469
低功耗状态	31
地址错误	51, 468
地址映像	33
电路板设计的注意事项	45
断点比较条件	72
端口输出的允许 (POE)	279
对绝对精度的影响	358
多处理器通信功能	333
多功能定时器脉冲单元 2 (MTU2)	97

F

非线性误差	356
分频器	38
复位同步 PWM 模式	187
复位状态	31

G

高阻抗控制对象的管脚及其条件	289
各模块的运行时钟	39
各运行模式的寄存器状态	497
更改运行模式时的注意事项	35
管脚功能控制器 (PFC)	368
管脚状态	520
过程寄存器 (PR)	13

H

互补 PWM 模式	190

J

I/O 端口	397
寄存器	
ADCR	347
ADCSR	345
ADDR0 ~ ADDR7	345
ADTSR	349
BAMRA	75
BAMRB	79
BARA	74
BARB	78
BBRA	76
BBRB	81
BDMRA	78
BDMRB	80
BDRA	77
BDRB	79
BETR	84
BRDR	85
BRDR	85
BRSR	84
CMCNT	363
CMCOR	363
CMCSR	362
CMSTR	361
DPFR	426
FCCS	421
FEBS	432
FECS	423
FKEY	423
FMPAR	429
FMPDR	430
FPCS	422
FPEFEQ	427
FPFR	428, 430, 433
FRQCR	41
FTDAR	424
FUBRA	427
ICR0	59
IFCR	395
IPRA ~ IPRF、IPRH ~ IPRM	64
IRQCR	60

IRQSR	61
OSCCR	42
PACRL1 ~ PACRL4	376
PADRL	398
PAIORL	375
PAPRL	400
PBCRH1	384
PBCRL1	384
PBCRL2	384
PBDRH	403
PBDRL	403
PBIORH	383
PBIORL	383
PBPRH	405
PBPRL	405
PECRL1 ~ PECRL4	388
PEDRL	408
PEIORL	387
PEPRL	410
PFDRL	413
RAMCR	476
SCBRR	311
SCRDR	302
SCRSR	302
SCSCR	305
SCSDCR	310
SCSMR	303
SCSPTR	309
SCSSR	307
SCTDR	302
SCTSR	302
STBCR1	470
STBCR2	471
STBCR3	472
STBCR4	473
STBCR5	474
STBCR6	475
TADCOBRA/B_4	140
TADCORA/B_4	140
TADCR	138
TBTER	158
TBTM	136
TCBR	155
TCDR	155
TCNT	140
TCNTCMPCLR	128
TCNTS	154
TCR	105
TCSYSTR	144
TDDR	154
TDER	159
TGCR	153
TGR	141
TICCR	137
TIER	129
TIOR	110
TITCNT	157

TITCR	155
TMDR	108
TOCR1	147
TOCR2	149
TOER	146
TOLBR	152
TRWER	145
TSR	132
TSTR	141
TSYR	143
TWCR	160
WTCNT	293
WTCSR	294
寄存器存取时的注意事项	296
寄存器的初始值	13
寄存器的数据格式	14
寄存器地址一览表（地址顺序）	480
寄存器位一览	486
寄存器一览	479
IRQ 中断	65
加电复位	49
间隔定时器模式的用法	298
晶体振荡器	38
绝对精度	356
绝对最大额定值	502

K

看门狗定时器（WDT）	292
空载时间的补偿功能	220
控制信号的时序	507

L

立即数的数据格式	14
连接晶体谐振器的方法	43
量化误差	356
逻辑运算指令	27

M

MCU 运行模式	32
MTU2 的功能一览表	98
MTU2 的中断源	222
MTU2 模块的时钟（MP ϕ ）	36
MTU2 输出管脚的初始化方法	248
满刻度误差	356
模块待机功能	478
模块待机模式的设定	341, 357, 366, 468

N

NMI 中断	65
内部外围模块中断	66

P

PC 跟踪	88
偏移误差	356
频率的变更方法	43

Q

取指令周期的断点	86
全局基址寄存器 (GBR)	13

R

RAM	468
RISC 方式	15
容许信号源阻抗	357
软件保护	444
软件待机模式	477

S

SCI 的中断源	338
SCSPTR 和 SCI 管脚的关系	339
闪存	414
闪存结构	417
闪存特性	518
设计电路板的注意事项	358
时钟频率的控制电路	38
时钟时序	505
时钟同步模式	299, 326
时钟运行模式	39
时钟振荡器 (CPG)	36
手动复位	50
数据存取周期的断点	87
数据传送指令	25
输入外部时钟的方法	44
睡眠模式	477
顺序断点	87
算术运算指令	26

T

通用寄存器	12

W

VCL 电容连接方法	519
外部触发的输入时序	355
外部脉宽的测量功能	219
外围时钟 (P ϕ)	36
外形尺寸图	524

X

系统控制指令	30
陷阱指令	53

向量基址寄存器 (VBR)	13
型号一览	523
寻址方式	17

Y

一般非法指令	54
异步模式	299, 318
异常处理	46
异常处理后的堆栈状态	55
异常处理向量表	48
异常处理向量表地址的计算方法	49
异常处理向量表	66
异常处理状态	31
移位指令	28
引导模式	434
硬件保护	444
用户编程模式	437
用户断点运行的流程	86
用户断点中断	66
用户分支处理的启动间隔	447
用户分支处理时间	447
用户 MAT	417

Z

噪声对策的注意事项	359
振荡停止检测功能	44
指令格式	20
指令特点	15
指令系统	22
中断	52
中断控制器 (INTC)	57
中断响应时间	71
中断异常处理结束后的堆栈状态	70
中断优先级	52
中断运行的流程	68
中止的发送	340
中止的检测和处理	340
转移指令	29
状态寄存器 (SR)	12
总线时钟 (B ϕ)	36
总线状态控制器 (BSC)	94
看门狗定时器模式的用法	297
连续扫描模式	351
模拟电源管脚等的设定范围	358
用户断点控制器 (UBC)	72

瑞萨 32 位 RISC 单片机
硬件手册
SH7125 群、SH7124 群

Publication Date: 1st Edition, Mar. 13, 2007

Published by: Sales Strategic Planning Div.
Renesas Technology Corp.

Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan



RENESAS SALES OFFICES

<http://www.renesas.com>

Refer to "<http://www.renesas.com/en/network>" for the latest and detailed information.

Renesas Technology America, Inc.

450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.

Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

Renesas Technology Hong Kong Ltd.

7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.

10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology Singapore Pte. Ltd.

1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.

Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd

Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

SH7125群、SH7124群



瑞萨电子株式会社

RCJ09B0027-0100