

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.

SH7764 群

瑞萨 32 位 RISC 单片机

SuperH™ RISC engine 族/SH-4A 系列

SH77641

R5S77641

SH77640

R5S77640

Notes regarding these materials

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

注意

本文只是参考译文，前页所载英文版“Cautions”具有正式效力。

关于利用本资料时的注意事项

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

产品使用上的注意事项

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

1. 未使用的引脚的处理

【注意】将未使用的引脚按照本文的“未使用引脚的处理”进行处理。

CMOS产品的输入引脚一般为高阻抗输入。如果将未使用的输入引脚处于开放状态运行，由于感应现象，外加LSI周围的噪声，在LSI内部产生穿透电流，被误认为是输入信号，引起误动作。未使用的引脚，按照本文的“未使用引脚的处理”中说明的指示进行处理。

2. 通电时的处理

【注意】通电时产品处于不定状态。

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

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

3. 禁止存取未定义的地址

【注意】禁止存取未定义的地址

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

4. 关于时钟

【注意】复位时，时钟稳定后请解除复位。

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

5. 关于产品间的差异

【注意】在变更不同型号的产品时，请事先确认是否没问题。

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

本手册的构成

本手册由以下内容构成：

1. 产品使用上的注意事项
2. 本书的构成
3. 前言
4. 目录
5. 概要
6. 各功能模块的说明

- CPU 和系统控制
- 内部外围模块

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

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

7. 附录
8. 索引

前言

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

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

目 的 本手册是以“能让用户理解本 LSI 的硬件功能和电特性”为目的。

阅读方法

- 希望了解全部功能时
 - 请按照目录的顺序进行阅读。
 - 本书大致按 CPU、系统控制功能、外围功能等的顺序进行分类。
- 希望了解 FPU 功能和各指令的详细内容时
 - 请参照《SH-4A 扩展功能软件手册》。

范 例 位的表示 : 以左侧为高位、右侧为低位的顺序进行表示。
数字的表示 : 2 进制数为 B'XXXX, 16 进制数为 H'XXXX, 10 进制数为 XXXX
符号的表示 : 对低电平有效的信号加上划线 ($\overline{\text{XXXX}}$)。

在本手册中使用的省略语或者略称如下所示：

- 本产品固有的略称

略称	英文名	日文名
ATAPI	ATAPI Controller	ATAPI 控制器
CPG	Clock Pulse Generator	时钟振荡器
DMAC	Direct Memory Access Controller	直接存储器存取控制器
E-DMAC	Ethernet Controller Direct Memory Access Controller (E-DMAC)	用于以太网控制器的直接存储器存取控制器
EtherC	Ethernet Controller	以太网控制器
FLCTL	NAND Flash Memory Controller	NAND 闪存控制器
G2D	2D Graphics Engine	2D 图形引擎
GPIO	General Purpose I/O	通用输入 / 输出端口
H-UDI	User Debug Interface	用户调试接口
IIC	I2C Bus Interface	I2C 总线接口
INTC	Interrupt Controller	中断控制器
MCU	Memory Controller Unit	存储器控制单元
MMU	Memory Management Unit	存储器管理单元
SCIF	Serial Communication Interface with FIFO	FIFO 内置型串行通信接口
TMU	Timer Unit	复位和定时器单元
UBC	User Break Controller	用户断点控制器
USB	USB Host/Function Interface	USB 主机 / 功能接口
VDC2	Video Display Controller2	显示控制器 2
WDT	Watchdog Timer and Reset	复位和看门狗定时器
SSI	Serial Sound Interface	串行音频接口
LCDC	LCD Controller	LCD 控制器
SRC	Sampling Rate Converter	采样速率转换器

- 其他省略语或者略称

省略语 / 略称	英文名	日文名
bps	bits per second	传送速度的单位
CRC	Cyclic Redundancy Check	循环冗余校验
DMA	Direct Memory Access	直接存储器存取
DMAC	Direct Memory Access Controller	直接存储器存取控制器
Hi-Z	High Impedance	高阻抗
I/O	Input/Output	输入 / 输出
LSB	Least Significant Bit	最低有效位
MSB	Most Significant Bit	最高有效位
NC	Non-Connection	未连接引脚
PLL	Phase Locked Loop	锁相环路

【注册商标和商标】

所有的商标及注册商标分别归属于其所有者。

目录

第 1 章	概要	1
1.1	SH7764 特点	1
1.2	框图	8
1.3	引脚排列	9
1.4	引脚功能	10
1.5	地址映像	16
第 2 章	编程模型	18
2.1	数据格式	18
2.2	寄存器结构	18
2.2.1	特权模式和存储体	18
2.2.2	通用寄存器	20
2.2.3	浮点寄存器	21
2.2.4	控制寄存器	23
2.2.5	系统寄存器	25
2.3	存储器映像寄存器	28
2.4	寄存器的数据格式	28
2.5	存储器的数据格式	28
2.6	处理状态	29
2.7	使用时的注意事项	30
2.7.1	自改写代码的注意事项	30
第 3 章	指令系统	31
3.1	执行环境	31
3.2	寻址方式	32
3.3	指令系统	35
第 4 章	流水线操作	44
4.1	流水线	44
4.2	并行执行性	54
4.3	发行速率和执行状态	56
第 5 章	异常处理	63
5.1	概要	63
5.2	寄存器说明	63
5.2.1	TRAPA 异常寄存器 (TRA)	64
5.2.2	异常事件寄存器 (EXPEVT)	64
5.2.3	中断事件寄存器 (INTEVT)	65
5.2.4	非支持检测异常寄存器 (EXPMASK)	65
5.3	异常处理功能	66
5.3.1	异常处理的流程	66
5.3.2	异常处理向量地址	67
5.4	异常的种类和优先顺序	67
5.5	异常流程	68
5.5.1	异常流程	68
5.5.2	异常源的接受	68
5.5.3	异常请求和 BL 位	69
5.5.4	异常处理的返回	69
5.6	各异常的说明	70
5.6.1	复位	70

5.6.2	一般异常	71
5.6.3	中断	80
5.6.4	发生多次异常时的优先顺序	81
5.7	注意事项	82
第 6 章	浮点单元 (FPU)	83
6.1	概要	83
6.2	数据格式	83
6.2.1	浮点数格式	83
6.2.2	非数 (NaN)	85
6.2.3	非规格化数	85
6.3	寄存器	86
6.3.1	浮点寄存器	86
6.3.2	浮点状态 / 控制寄存器 (FPSCR)	88
6.3.3	浮点通信寄存器 (FPUL)	89
6.4	舍入	90
6.5	浮点异常	90
6.6	图形支持功能	92
6.6.1	几何运算指令	92
6.6.2	单精度数据对的传送	93
第 7 章	存储器管理单元 (MMU)	94
7.1	MMU 的概要	94
7.1.1	地址空间	95
7.2	寄存器说明	100
7.2.1	页表入口高位寄存器 (PTEH)	100
7.2.2	页表入口低位寄存器 (PTEL)	101
7.2.3	转换表基址寄存器 (TTB)	102
7.2.4	TLB 异常地址寄存器 (TEA)	102
7.2.5	MMU 控制寄存器 (MMUCR)	102
7.2.6	页表入口辅助寄存器 (PTEA)	104
7.2.7	物理地址空间控制寄存器 (PASCR)	105
7.2.8	重新取指令禁止控制寄存器 (IRMCR)	106
7.3	TLB 功能 (TLB 兼容模式: MMUCR.ME=0)	107
7.3.1	共用 TLB (UTLB) 的结构	107
7.3.2	指令 TLB (ITLB) 的结构	108
7.3.3	地址转换方式	109
7.4	TLB 功能 (TLB 扩展模式: MMUCR.ME=1)	111
7.4.1	共用 TLB (UTLB) 的结构	111
7.4.2	指令 TLB (ITLB) 的结构	113
7.4.3	地址转换方式	114
7.5	MMU 的功能	116
7.5.1	MMU 的硬件管理	116
7.5.2	MMU 的软件管理	116
7.5.3	MMU 的指令 (LDTLB)	116
7.5.4	硬件 ITLB 未命中处理	118
7.5.5	同义问题的避免	118
7.6	MMU 异常	119
7.6.1	指令 TLB 多重命中异常	119
7.6.2	指令 TLB 未命中异常	119
7.6.3	指令 TLB 保护违反异常	120
7.6.4	数据 TLB 多重命中异常	120

7.6.5	数据 TLB 命中异常	121
7.6.6	数据 TLB 保护违反异常	121
7.6.7	初始页写异常	122
7.7	存储器映像 TLB 的结构	123
7.7.1	ITLB 地址阵列	123
7.7.2	ITLB 数据阵列 (TLB 兼容模式)	124
7.7.3	ITLB 数据阵列 (TLB 扩展模式)	124
7.7.4	UTLB 地址阵列	125
7.7.5	UTLB 数据阵列 (TLB 兼容模式)	126
7.7.6	UTLB 数据阵列 (TLB 扩展模式)	127
7.8	使用时的注意事项	128
7.8.1	LDTLB 指令的使用注意事项	128
第 8 章	高速缓存	129
8.1	特点	129
8.2	寄存器说明	131
8.2.1	高速缓存控制寄存器 (CCR)	132
8.2.2	队列地址控制寄存器 0 (QACR0)	133
8.2.3	队列地址控制寄存器 1 (QACR1)	133
8.2.4	内部存储器控制寄存器 (RAMCR)	134
8.3	操作数高速缓存的运行说明	135
8.3.1	读操作	135
8.3.2	预取操作	135
8.3.3	写操作	136
8.3.4	回写缓冲器	137
8.3.5	直写缓冲器	137
8.3.6	OC 2 通路模式	137
8.4	指令高速缓存的操作说明	138
8.4.1	读操作	138
8.4.2	预取操作	138
8.4.3	IC 2 通路模式	138
8.4.4	指令高速缓存的通路预测	138
8.5	高速缓存操作指令	139
8.5.1	高速缓存和外部存储器的一致性	139
8.5.2	预取操作	140
8.6	存储器映像高速缓存的结构	141
8.6.1	IC 地址阵列	141
8.6.2	IC 数据阵列	142
8.6.3	OC 地址阵列	142
8.6.4	OC 数据阵列	143
8.6.5	存储器映像的联想写操作	144
8.7	存储队列	145
8.7.1	SQ 的结构	145
8.7.2	写 SQ	145
8.7.3	传送到外部存储器	145
8.7.4	SQ 存取的异常判断	146
8.7.5	读 SQ	146
第 9 章	内部存储器	147
9.1	特点	147
9.2	寄存器说明	147
9.2.1	内部存储器的控制寄存器 (RAMCR)	148

9.3	运行说明	149
9.3.1	从 CPU 取指令的存取	149
9.3.2	从 CPU 的操作数存取和从 FPU 的存取	149
9.3.3	从 SuperHyway 总线主模块的存取	149
9.4	内部存储器的保护功能	149
9.5	使用时的注意事项	150
9.5.1	页竞争	150
9.5.2	页转换	150
9.5.3	一致性	150
9.5.4	睡眠模式	150
第 10 章	时钟振荡器 (CPG)	151
10.1	特点	151
10.2	输入 / 输出引脚	153
10.3	时钟运行模式	154
10.4	寄存器说明	154
10.4.1	频率控制寄存器 (FRQCR)	155
10.4.2	PLL 控制寄存器 (PLLCR)	156
10.4.3	VDC2 时钟控制寄存器 (VDC2CLKCR)	157
第 11 章	存储器控制单元 (MCU)	158
11.1	特点	158
11.2	输入 / 输出引脚	161
11.3	区域的概要	162
11.3.1	空间分割	162
11.3.2	存储器的总线宽度	163
11.3.3	字节排序的设定	163
11.4	寄存器说明	164
11.4.1	版本控制寄存器 (VCR)	165
11.4.2	存储器接口模式寄存器 (MIM)	166
11.4.3	SDRAM 控制寄存器 (SCR)	168
11.4.4	SDRAM 时序寄存器 (STR)	169
11.4.5	SDRAM 行属性寄存器 (SDRA)	171
11.4.6	SDRAM 模式寄存器 (SDMR)	172
11.4.7	仲裁模式寄存器 (AMR)	173
11.4.8	线性平铺转换控制寄存器 (LTCn)	174
11.4.9	线性平铺转换区起始地址寄存器 (LTADn)	175
11.4.10	线性平铺转换区起始地址屏蔽寄存器 (LTAMn)	176
11.4.11	请求屏蔽设定寄存器 (RQM)	177
11.4.12	总线控制寄存器 (BCR)	178
11.4.13	CS0 总线控制寄存器 (CS0BCR)	180
11.4.14	CSn 等待控制寄存器 (CSnWCR)	182
11.4.15	CS3 总线控制寄存器 (CS3BCR)	185
11.5	运行说明	188
11.5.1	字节排序 / 存取长度和数据调整	188
11.5.2	各模块的数据调整	195
11.6	SRAM 接口	196
11.6.1	基本时序	196
11.6.2	等待状态控制	199
11.6.3	读选通无效时序	201
11.7	SDRAM 接口	202
11.7.1	SDRAM 直接连接方式	202

11.7.2	地址多路复用	204
11.7.3	突发读	205
11.7.4	突发写	206
11.7.5	单读	207
11.7.6	单写	208
11.7.7	存储体打开模式	209
11.7.8	刷新	213
11.7.9	SDRAM 初始化顺序	214
11.8	存取周期期间的等待	214
11.8.1	区域 0、3 周期期间的等待	214
11.8.2	区域 1、2 周期期间的等待	214
11.8.3	区域 1、2- 区域 0、3 周期期间的等待	214
11.9	总线仲裁	215
11.9.1	内部模块存取的仲裁	215
11.9.2	多阶段仲裁的运行	217
11.9.3	外部设备的总线请求	218
11.9.4	总线释放和获得的顺序	219
11.9.5	主控和从属的协调	220
11.10	数据的一致性	221
11.11	线性和平铺的地址转换	223
11.12	使用时的注意事项	225
11.12.1	刷新	225
11.12.2	外部总线的仲裁	225
11.12.3	Pixel 总线的传送长度和存取地址边界	225
第 12 章	直接存储器存取控制器 (DMAC)	226
12.1	特点	226
12.2	输入 / 输出引脚	228
12.3	寄存器说明	228
12.3.1	DMA 源地址寄存器 0 ~ 5 (SAR0 ~ SAR5)	231
12.3.2	DMA 源地址寄存器 B0 ~ 3 (SARB0 ~ SARB3)	231
12.3.3	DMA 目标地址寄存器 0 ~ 5 (DAR0 ~ DAR5)	231
12.3.4	DMA 目标地址寄存器 B0 ~ 3 (DARB0 ~ DARB3)	232
12.3.5	DMA 传送计数寄存器 0 ~ 5 (TCR0 ~ TCR5)	232
12.3.6	DMA 传送计数寄存器 B0 ~ 3 (TCRB0 ~ TCRB3)	232
12.3.7	DMA 通道控制寄存器 0 ~ 5 (CHCR0 ~ CHCR5)	233
12.3.8	DMA 操作寄存器 0 (DMAOR0)	237
12.3.9	DMA 扩展资源选择器 0 ~ 2 (DMARS0 ~ DMARS2)	239
12.4	运行说明	241
12.4.1	DMA 传送请求	241
12.4.2	通道优先顺序	243
12.4.3	DMA 传送的种类	245
12.4.4	传送流程	249
12.4.5	重复模式的传送	250
12.4.6	重加载模式的传送	251
12.4.7	DREQ 引脚的采样时序	252
12.5	使用时的注意事项	255
12.5.1	模块停止	255
12.5.2	地址错误	255
12.5.3	突发模式传送时的注意事项	255
12.5.4	DACK 的分割输出	255
12.5.5	禁止对 DMAC 的 DMA 传送	255

12.5.6	NMI 中断	255
第 13 章	中断控制器 (INTC)	256
13.1	特点	256
13.1.1	中断方式	258
13.1.2	INTC 的中断	258
13.2	输入 / 输出引脚	260
13.3	寄存器说明	260
13.3.1	中断控制寄存器 0 (ICR0)	263
13.3.2	中断控制寄存器 1 (ICR1)	264
13.3.3	中断优先顺序设定寄存器 (INTPRI)	264
13.3.4	中断源寄存器 (INTREQ)	265
13.3.5	中断屏蔽寄存器 (INTMSK)	266
13.3.6	中断屏蔽清除寄存器 (INTMSKCLR)	266
13.3.7	NMI 标志控制寄存器 (NMIFCR)	267
13.3.8	用户中断屏蔽级设定寄存器 (USERIMASK)	267
13.3.9	中断优先顺序设定寄存器 (INT2PRI0 ~ INT2PRI12)	269
13.3.10	中断源寄存器 0 (不影响屏蔽状态) (INT2A0)	269
13.3.11	中断源寄存器 01 (不影响屏蔽状态) (INT2A01)	271
13.3.12	中断源寄存器 1 (影响屏蔽状态) (INT2A1)	272
13.3.13	中断源寄存器 11 (影响屏蔽状态) (INT2A11)	273
13.3.14	中断屏蔽寄存器 (INT2MSKR)	273
13.3.15	中断屏蔽寄存器 1 (INT2MSKR1)	275
13.3.16	中断屏蔽清除寄存器 (INT2MSKCR)	276
13.3.17	中断屏蔽清除寄存器 1 (INT2MSKCR1)	277
13.3.18	各内部模块中断源寄存器 (INT2B0、INT2B2 ~ INT2B7)	278
13.3.19	GPIO 中断设定寄存器 (INT2GPIC)	281
13.4	中断源	283
13.4.1	NMI 中断	283
13.4.2	IRQ 中断	283
13.4.3	内部外围模块的中断	283
13.4.4	内部外围模块的中断优先顺序	284
13.4.5	中断异常处理和优先顺序	284
13.5	运行说明	287
13.5.1	中断的运行流程	287
13.5.2	多重中断	289
13.5.3	通过 MAI 位进行中断屏蔽	289
13.6	中断响应时间	290
13.7	使用时的注意事项	291
13.7.1	中断和电平检测设定时的 IRQ 中断处理程序例子	291
13.7.2	设定 IRQ 引脚功能的注意事项	292
13.7.3	IRQ 中断请求的清除方法	292
第 14 章	定时器单元 (TMU)	293
14.1	特点	293
14.2	输入 / 输出引脚	295
14.3	寄存器说明	295
14.3.1	定时器输出控制寄存器 (TOCR)	297
14.3.2	定时器启动寄存器 (TSTRn) (n=0、1)	297
14.3.3	定时器常数寄存器 (TCORn) (n=0 ~ 5)	298
14.3.4	定时器计数器 (TCNTn) (n=0 ~ 5)	299
14.3.5	定时器控制寄存器 (TCRn) (n=0 ~ 5)	299

14.3.6	输入捕捉寄存器 2 (TCPR2)	301
14.4	运行说明	301
14.4.1	计数器运行	301
14.4.2	输入捕捉功能	303
14.5	中断	304
14.6	使用时的注意事项	304
14.6.1	寄存器的写操作	304
14.6.2	TCNT 寄存器的读操作	304
14.6.3	外部时钟频率	304
第 15 章	FIFO 内置型串行通信接口 (SCIF)	305
15.1	特点	305
15.2	输入 / 输出引脚	306
15.3	寄存器说明	307
15.3.1	接收移位寄存器 (SCRSR)	309
15.3.2	接收 FIFO 数据寄存器 (SCFRDR)	309
15.3.3	发送移位寄存器 (SCTSR)	309
15.3.4	发送 FIFO 数据寄存器 (SCFTDR)	309
15.3.5	串行模式寄存器 (SCSMR)	310
15.3.6	串行控制寄存器 (SCSCR)	311
15.3.7	串行状态寄存器 (SCFSR)	314
15.3.8	位速率寄存器 (SCBRR)	317
15.3.9	FIFO 控制寄存器 (SCFCR)	321
15.3.10	FIFO 数据计数置位寄存器 (SCFDR)	323
15.3.11	串行端口寄存器 (SCSPTR)	323
15.3.12	线路状态寄存器 (SCLSR)	325
15.3.13	串行扩展模式寄存器 (SCEMR)	326
15.4	运行说明	327
15.4.1	概要	327
15.4.2	异步模式的运行	328
15.4.3	时钟同步模式的运行	336
15.5	SCIF 的中断	343
15.6	使用时的注意事项	343
15.6.1	SCFTDR 的写操作和 TDFE 标志	343
15.6.2	SCFRDR 的读操作和 RDF 标志	343
15.6.3	中止的检测和处理	344
15.6.4	中止信号的发送	344
15.6.5	异步模式的接收数据采样时序和接收容限	344
15.6.6	异步基本时钟的选择	345
第 16 章	I²C 总线接口 (IIC)	346
16.1	特点	346
16.2	输入 / 输出引脚	346
16.3	寄存器说明	347
16.3.1	从属控制寄存器 (ICSCR)	348
16.3.2	从属状态寄存器 (ICSSR)	349
16.3.3	从属中断允许寄存器 (ICSIER)	350
16.3.4	从属地址寄存器 (ICSAR)	351
16.3.5	主控控制寄存器 (ICMCR)	351
16.3.6	主控状态寄存器 (ICMSR)	352
16.3.7	主控中断允许寄存器 (ICMIER)	354
16.3.8	主控地址寄存器 (ICMAR)	355

16.3.9	时钟控制寄存器 (ICCCR)	355
16.3.10	接收 / 发送数据寄存器 (ICRXD/ICTXD)	356
16.4	运行说明	357
16.4.1	数据和时钟滤波器	357
16.4.2	时钟发生器	357
16.4.3	主控 / 从属接口	357
16.4.4	软件状态的联锁	357
16.4.5	I ² C 总线数据格式	358
16.4.6	7 位地址格式	358
16.4.7	10 位地址格式	359
16.4.8	主控发送的运行	361
16.4.9	主控接收的运行	362
16.5	程序例子	364
16.5.1	主控发送	364
16.5.2	主控接收	365
16.5.3	主控发送 - 重新开始 - 主控接收	366
第 17 章	ATAPI 接口 (ATAPI)	367
17.1	特点	367
17.2	输入 / 输出引脚	368
17.3	寄存器说明	369
17.3.1	ATAPI 控制寄存器 (ATAPI_CONTROL)	370
17.3.2	ATAPI 状态寄存器 (ATAPI_STATUS)	372
17.3.3	中断允许 (ATAPI_INT_ENABLE)	373
17.3.4	PIO 时序寄存器 (ATAPI_PIO_TIMING)	374
17.3.5	多字 DMA 时序寄存器 (ATAPI_MULTI_TIMING)	375
17.3.6	超级 DMA 时序寄存器 (ATAPI_ULTRA_TIMING)	376
17.3.7	描述符表基址寄存器 (ATAPI_DTB_ADR)	377
17.3.8	描述符表	377
17.3.9	结束标志和描述符 DMA 起始地址	378
17.3.10	描述符 DMA 传送计数	378
17.3.11	DMA 起始地址寄存器 (ATAPI_DMA_START_ADR)	379
17.3.12	DMA 传送计数寄存器 (ATAPI_DMA_TRANS_CNT)	380
17.3.13	ATAPI 控制 2 寄存器 (ATAPI_CONTROL2)	380
17.3.14	ATAPI 信号状态寄存器 (ATAPI_SIG_ST)	381
17.3.15	字节交换寄存器 (ATAPI_BYTE_SWAP)	381
17.3.16	ATAPI 的数据总线调整	382
17.4	功能说明	383
17.4.1	数据传送模式	383
17.4.2	描述符功能	383
17.5	运行步骤	384
17.5.1	初始化	384
17.5.2	PIO 传送模式的步骤	384
17.5.3	多字 DMA 传送模式的步骤	385
17.5.4	超级 DMA 传送模式的步骤	387
17.5.5	ATAPI 设备的硬件复位步骤	388
第 18 章	串行音频接口 (SSI)	389
18.1	特点	389
18.2	输入 / 输出引脚	390
18.3	寄存器说明	391
18.3.1	DMA 模式寄存器 0 ~ 5 (SSIDMMR0 ~ 5)	397

18.3.2	RDMA 传送源地址寄存器 0 ~ 5 (SSIRDMADR0 ~ 5)	398
18.3.3	RDMA 传送字节数寄存器 0 ~ 5 (SSIRDMCNTR0 ~ 5)	398
18.3.4	WDMA 传送目的地址寄存器 0 ~ 5 (SSIWDMADR0 ~ 5)	399
18.3.5	WDMA 传送字节数寄存器 0 ~ 5 (SSIWDMCNTR0 ~ 5)	399
18.3.6	DMA 控制寄存器 0 ~ 5 (SSIDMCOR0 ~ 5)	400
18.3.7	发送暂停块计数器 0 ~ 5 (SSISTPBLCNT0 ~ 5)	405
18.3.8	发送暂停中传送数据寄存器 0 ~ 5 (SSISTPDR0 ~ 5)	406
18.3.9	块计数源寄存器 0 ~ 5 (SSIBLCNTR0 ~ 5)	406
18.3.10	块计数器 0 ~ 5 (SSIBLCNT0 ~ 5)	407
18.3.11	模块 xn 次中断计数源寄存器 0 ~ 5 (SSIBLNCNTR0 ~ 5)	407
18.3.12	块 xn 次计数器 0 ~ 5 (SSIBLNCNT0 ~ 5)	408
18.3.13	DMA 操作寄存器 0 ~ 1 (SSIDMAOR0 ~ 1)	408
18.3.14	中断状态寄存器 0 ~ 1 (SSIDMINTSR0 ~ 1)	410
18.3.15	中断屏蔽寄存器 0 ~ 1 (SSIDMINTMR0 ~ 1)	412
18.3.16	控制寄存器 0 ~ 5 (SSICR0 ~ 5)	414
18.3.17	状态寄存器 0 ~ 5 (SSISR0 ~ 5)	417
18.3.18	发送数据寄存器 0 ~ 5 (SSITDR0 ~ 5)	420
18.3.19	接收数据寄存器 0 ~ 5 (SSIRDOR0 ~ 5)	420
18.4	运行说明	421
18.4.1	SSI_CLKSEL 的运行说明	421
18.4.2	SSI_DMAC0/1 的运行说明	421
18.4.3	SSI_CH0 ~ 5 的运行说明	422
18.5	使用时的注意事项	435
18.5.1	接收 DMA 运行中发生上溢时的限制事项	435
18.5.2	从属模式中运行时的限制事项	435
18.5.3	各种寄存器设定时的注意事项	435
第 19 章	以太网控制器 (EtherC)	436
19.1	特点	436
19.2	输入 / 输出引脚	437
19.3	寄存器说明	438
19.3.1	EtherC 模式寄存器 (ECMR)	440
19.3.2	EtherC 状态寄存器 (ECSR)	442
19.3.3	EtherC 中断允许寄存器 (ECSIPR)	443
19.3.4	PHY 部接口寄存器 (PIR)	443
19.3.5	MAC 地址高位设定寄存器 (MAHR)	444
19.3.6	MAC 地址低位设定寄存器 (MALR)	444
19.3.7	接收帧长度上限寄存器 (RFLR)	445
19.3.8	PHY 部状态寄存器 (PSR)	446
19.3.9	试发送失败计数寄存器 (TROCR)	446
19.3.10	延迟冲突检测计数寄存器 (CDCR)	447
19.3.11	载波消失计数寄存器 (LCCR)	447
19.3.12	载波未检测计数寄存器 (CNDCR)	448
19.3.13	CRC 错误帧接收计数寄存器 (CEFCR)	448
19.3.14	帧接收错误计数寄存器 (FRECR)	449
19.3.15	64 字节未帧接收计数寄存器 (TSFRCR)	449
19.3.16	指定字节帧超过接收计数寄存器 (TLFRCR)	450
19.3.17	剩余位帧接收计数寄存器 (RFCR)	450
19.3.18	多播地址帧接收计数寄存器 (MAFCR)	451
19.3.19	IPG 设定寄存器 (IPGR)	451
19.3.20	自动 PAUSE 帧设定寄存器 (APR)	452
19.3.21	手动 PAUSE 帧设定寄存器 (MPR)	452

19.3.22	自动 PAUSE 帧再发送次数设定寄存器 (TPAUSER)	453
19.3.23	随机数生成计数器上限值设定 (RDMLR)	453
19.3.24	接收 PAUSE 帧计数器 (RFCF)	454
19.3.25	PAUSE 帧再发送次数计数器 (TPAUSECR)	454
19.3.26	Broadcast 帧接收次数设定寄存器 (BCFRR)	455
19.4	运行说明	456
19.4.1	发送运行	456
19.4.2	接收运行	457
19.4.3	MII 帧时序	458
19.4.4	MII 寄存器的存取方法	459
19.4.5	Magic Packet 的检测	461
19.4.6	根据 IPG 设定的运行	462
19.4.7	流控制	462
19.5	与 PHY-LSI 的连接	463
19.6	使用时的注意事项	463
第 20 章	用于以太网控制器的直接存储器存取控制器 (E-DMAC)	464
20.1	特点	464
20.2	寄存器说明	465
20.2.1	E-DMAC 模式寄存器 (EDMR)	467
20.2.2	E-DMAC 发送请求寄存器 (EDTRR)	467
20.2.3	E-DMAC 接收请求寄存器 (EDRRR)	468
20.2.4	发送描述符列表起始地址寄存器 (TDLAR)	469
20.2.5	接收描述符列表起始地址寄存器 (RDLAR)	469
20.2.6	EtherC/E-DMAC 状态寄存器 (EESR)	470
20.2.7	EtherC/E-DMAC 状态中断允许寄存器 (EESIPR)	473
20.2.8	发送 / 接收状态复制指示寄存器 (TRSCER)	475
20.2.9	丢失帧计数寄存器 (RMFCR)	476
20.2.10	发送 FIFO 阈值指定寄存器 (TFTR)	477
20.2.11	FIFO 容量指定寄存器 (FDR)	478
20.2.12	接收方式控制寄存器 (RMCR)	479
20.2.13	发送 FIFO 下溢计数 (TFUCR)	480
20.2.14	接收 FIFO 上溢计数 (RFOCR)	480
20.2.15	接收缓冲器写地址寄存器 (RBWAR)	481
20.2.16	接收描述符取地址寄存器 (RDFAR)	481
20.2.17	发送缓冲器读地址寄存器 (TBRAR)	482
20.2.18	发送描述符取地址寄存器 (TDFAR)	482
20.2.19	流控制开始 FIFO 阈值设定寄存器 (FCFTR)	483
20.2.20	接收数据填充设定寄存器 (RPADIR)	484
20.2.21	发送中断设定寄存器 (TRIMD)	484
20.2.22	各输出信号设定寄存器 (IOSR)	485
20.3	运行说明	486
20.3.1	描述符列表和数据缓冲器	486
20.3.2	发送功能	490
20.3.3	接收功能	491
20.3.4	多缓冲器帧的发送和接收处理	492
第 21 章	USB2.0 主机 / 功能模块 (USB)	494
21.1	特点	494
21.2	输入 / 输出引脚	495
21.3	寄存器说明	496
21.3.1	系统配置控制寄存器 (SYSCFG)	500

21.3.2	CPU 总线等待寄存器 (BUSWAIT)	502
21.3.3	系统配置状态寄存器 (SYSSTS)	503
21.3.4	设备状态控制寄存器 (DVSTCTR)	504
21.3.5	测试模式寄存器 (TESTMODE)	507
21.3.6	DMA-FIFO 总线配置寄存器 (D0FBCFG、D1FBCFG)	508
21.3.7	FIFO 端口寄存器 (CFIFO、D0FIFO、D1FIFO)	509
21.3.8	FIFO 端口选择寄存器 (CFIFOSEL、D0FIFOSEL、D1FIFOSEL)	511
21.3.9	FIFO 端口控制寄存器 (CFIFOCTR、D0FIFOCTR、D1FIFOCTR)	515
21.3.10	中断允许寄存器 0 (INTENB0)	516
21.3.11	中断允许寄存器 1 (INTENB1)	518
21.3.12	BRDY 中断允许寄存器 (BRDYENB)	519
21.3.13	NRDY 中断允许寄存器 (NRDYENB)	520
21.3.14	BEMP 中断允许寄存器 (BEMPENB)	521
21.3.15	SOF 控制寄存器 (SOFCFG)	522
21.3.16	中断状态寄存器 0 (INTSTS0)	523
21.3.17	中断状态寄存器 1 (INTSTS1)	525
21.3.18	BRDY 中断状态寄存器 (BRDYSTS)	528
21.3.19	NRDY 中断状态寄存器 (NRDYSTS)	529
21.3.20	BEMP 中断状态寄存器 (BEMPSTS)	530
21.3.21	帧号寄存器 (FRMNUM)	531
21.3.22	μ 帧号寄存器 (UFRMNUM)	532
21.3.23	USB 地址寄存器 (USBADDR)	532
21.3.24	USB 请求类型寄存器 (USBREQ)	533
21.3.25	USB 请求值寄存器 (USBVAL)	534
21.3.26	USB 请求索引寄存器 (USBINDX)	534
21.3.27	USB 请求长度寄存器 (USBLENG)	535
21.3.28	DCP 配置寄存器 (DCPCFG)	535
21.3.29	DCP 最大包长度寄存器 (DCPMAXP)	536
21.3.30	DCP 控制寄存器 (DCPCTR)	536
21.3.31	管道窗口选择寄存器 (PIPESEL)	541
21.3.32	管道配置寄存器 (PIPECFG)	541
21.3.33	管道缓冲器指定寄存器 (PIPEBUF)	545
21.3.34	管道最大包长度寄存器 (PIPEMAXP)	547
21.3.35	管道周期控制寄存器 (PIPEPERI)	548
21.3.36	管道 n 控制寄存器 (PIPE _n CTR) (n=1 ~ 9)	549
21.3.37	管道 n 事务计数允许寄存器 (PIPE _n TRE) (n=1 ~ 5)	558
21.3.38	管道 n 事务计数寄存器 (PIPE _n TRN) (n=1 ~ 5)	559
21.3.39	设备地址 n 配置寄存器 (DEVADD _n) (n=0 ~ A)	560
21.4	运行说明	561
21.4.1	系统控制和振荡控制	561
21.4.2	中断功能	562
21.4.3	管道控制	575
21.4.4	FIFO 缓冲存储器	580
21.4.5	控制传送 (DCP)	586
21.4.6	批量传送 (管道 1 ~ 5)	587
21.4.7	中断传送 (管道 6 ~ 9)	588
21.4.8	等时传送 (管道 1、2)	589
21.4.9	SOF 内插功能	595
21.4.10	管道的安排	596
21.5	使用时的注意事项	597
21.5.1	USB 模块的启动 / 停止步骤	597

第 22 章	LCD 控制器 (LCDC)	598
22.1	特点	598
22.2	输入 / 输出引脚	599
22.3	寄存器说明	600
22.3.1	LCDC 输入时钟寄存器 (LDICKR)	602
22.3.2	LCDC 模块类型寄存器 (LDMTR)	603
22.3.3	LCDC 数据格式寄存器 (LDDFR)	605
22.3.4	LCDC 上部显示屏的数据开始地址寄存器 (LDSARU)	606
22.3.5	LCDC 下部显示屏的数据开始地址寄存器 (LDSARL)	606
22.3.6	LCDC 显示屏的数据行地址偏移寄存器 (LDLAOR)	607
22.3.7	LCDC 配色控制寄存器 (LDPALCR)	607
22.3.8	配色数据寄存器 00 ~ FF (LDPR00 ~ LDPRFF)	608
22.3.9	LCDC 水平字符数寄存器 (LDHCNR)	608
22.3.10	LCDC 水平同步信号寄存器 (LDHSYNR)	609
22.3.11	LCDC 垂直显示行数寄存器 (LDVDLNR)	609
22.3.12	LCDC 垂直总行数寄存器 (LDVTLNR)	610
22.3.13	LCDC 垂直同步信号寄存器 (LDVSYNR)	611
22.3.14	LCDC AC 调制信号交替行数寄存器 (LDACLNR)	612
22.3.15	LCDC 中断控制寄存器 (LDINTR)	612
22.3.16	LCDC 电源管理模式寄存器 (LDPMMR)	614
22.3.17	LCDC 电源顺序周期寄存器 (LDPSPR)	615
22.3.18	LCDC 控制寄存器 (LDCNTR)	616
22.3.19	LCDC 用户指定中断控制寄存器 (LDUINTR)	617
22.3.20	LCDC 用户指定中断行数寄存器 (LDUINLNR)	618
22.3.21	LCDC 存储器存取间隔数寄存器 (LDLIRNR)	618
22.4	运行说明	619
22.4.1	LCDC 能显示的液晶模块的尺寸	619
22.4.2	彩色调色板规格	620
22.4.3	数据格式	621
22.4.4	显示分辨率的设定	624
22.4.5	电源控制顺序的处理	624
22.5	时钟与 LCD 数据信号例子	628
22.6	使用时的注意事项	636
22.6.1	显示数据保存的 VRAM (区域 1、2 的 SDRAM) 的存取停止步骤	636
22.6.2	MCU 请求接受停止时的注意事项	636
第 23 章	2D 图形引擎 (G2D)	637
23.1	基本功能	637
23.1.1	命令、绘图属性一览	637
23.1.2	基本功能	639
23.1.3	坐标系	645
23.1.4	数据格式	649
23.1.5	绘图属性	649
23.2	显示列表	657
23.2.1	4 个顶点的屏幕绘图命令	657
23.2.2	线的描绘	665
23.2.3	工作屏幕的绘图命令	683
23.2.4	工作线的绘图命令	691
23.2.5	矩形的绘图命令	694
23.2.6	控制命令	702
23.3	寄存器规格	713
23.3.1	系统控制寄存器	719

23.3.2	存储器控制寄存器	723
23.3.3	颜色控制寄存器	727
23.3.4	绘图控制寄存器	729
23.3.5	坐标变换控制寄存器	734
第 24 章	显示控制器 (VDC2)	739
24.1	概要	739
24.2	特点	739
24.3	输入 / 输出引脚	740
24.4	VDC2 结构	741
24.5	功能说明	742
24.5.1	图形 (层 1 ~ 层 4)	742
24.5.2	同步信号的生成	742
24.5.3	外部同步模式	744
24.5.4	数字视频的输出生成	745
24.5.5	RGB565→YC444 的转换	745
24.5.6	YC444→YC422 的转换	746
24.5.7	数据允许信号 (合成)	746
24.6	寄存器说明	747
24.6.1	图形部控制寄存器 (GRCMEN1、2、3、4)	750
24.6.2	总线控制寄存器 (GRCBUSCNT1、2、3、4)	751
24.6.3	图形图像基址寄存器 (GROPSADR1、2、3、4)	752
24.6.4	图形图像区寄存器 (GROPSWH1、2、3、4)	752
24.6.5	图形图像行偏移寄存器 (GROPSOFST1、2、3、4)	753
24.6.6	图形图像起始位置寄存器 (GROPDPHV1、2、3、4)	754
24.6.7	α 控制区寄存器 (GROPEWH2、3、4)	755
24.6.8	α 控制区起始位置寄存器 (GROPEDPHV2、3、4)	755
24.6.9	α 控制寄存器 (GROPEDPA2、3、4)	757
24.6.10	色键控制寄存器 (GROPCRKY0_2、3、4)	758
24.6.11	色度颜色指定寄存器 (GROPCRKY1_2、3、4)	759
24.6.12	图形图像区外的颜色设定寄存器 (GROPBASERGB1、2、3、4)	759
24.6.13	SG 模式设定寄存器 (SGMODE)	761
24.6.14	中断输出控制寄存器 (SGINTCNT)	762
24.6.15	同步信号控制寄存器 (SYNCNT)	763
24.6.16	外部输入同步信号时序控制寄存器 (EXTSYNCNT)	765
24.6.17	同步信号大小寄存器 (SYNSIZE)	766
24.6.18	垂直同步信号时序控制寄存器 (VSYNCTIM)	766
24.6.19	水平同步信号时序控制寄存器 (HSYNCTIM)	767
24.6.20	栅极时钟信号时序控制寄存器 (CLSTIM)	767
24.6.21	采样起始信号时序控制寄存器 (SPLTIM)	768
24.6.22	栅极控制信号时序控制寄存器 (COMTIM)	769
24.6.23	SGDE 区起始位置寄存器 (SGDESTART)	770
24.6.24	SGDE 区大小寄存器 (SGDESIZE)	771
24.6.25	CDE 色度颜色指定寄存器 (CDECRKY)	771
24.6.26	T1004 控制寄存器 (T1004CNT)	772
24.6.27	T1004 视频起始位置寄存器 (T1004OFFSET)	772
24.7	运行步骤	774
24.7.1	显示控制部	774
24.7.2	图形部	774
第 25 章	NAND 闪存控制器 (FLCTL)	775
25.1	特点	775

25.2	输入 / 输出引脚	778
25.3	寄存器说明	779
25.3.1	共用控制寄存器 (FLCMNCR)	780
25.3.2	命令控制寄存器 (FLCMD CR)	781
25.3.3	命令码寄存器 (FLCMCDR)	783
25.3.4	地址寄存器 (FLADR)	783
25.3.5	地址寄存器 2 (FLADR2)	784
25.3.6	数据计数寄存器 (FLDTCNTR)	785
25.3.7	数据寄存器 (FLDATAR)	786
25.3.8	中断 DMA 控制寄存器 (FLINTDMACR)	787
25.3.9	就绪 / 忙超时设定寄存器 (FLBSYTMR)	789
25.3.10	就绪 / 忙超时计数器 (FLBSYCNT)	790
25.3.11	数据 FIFO 寄存器 (FLDTFIFO)	791
25.3.12	管理码 FIFO 寄存器 (FLECFIFO)	792
25.3.13	传送控制寄存器 (FLTRCR)	793
25.4	运行说明	794
25.4.1	运行模式	794
25.4.2	寄存器的设定步骤	794
25.4.3	命令存取模式	795
25.4.4	扇区存取模式	797
25.4.5	ECC 的纠错	799
25.4.6	状态的读操作	799
25.5	中断处理	800
25.6	DMA 传送的设定	800
第 26 章	采样速率转换器 (SRC)	801
26.1	特点	801
26.2	寄存器说明	802
26.2.1	SRC 输入数据寄存器 (SRCID)	802
26.2.2	SRC 输出数据寄存器 (SRCOD)	803
26.2.3	SRC 输入数据控制寄存器 (SRCIDCTRL)	804
26.2.4	SRC 输出数据控制寄存器 (SRCODCTRL)	805
26.2.5	SRC 控制寄存器 (SRCCTRL)	806
26.2.6	SRC 状态寄存器 (SRCSTAT)	808
26.3	运行说明	810
26.3.1	初始设定	810
26.3.2	数据输入	810
26.3.3	数据输出	811
26.4	中断	812
26.5	使用时的注意事项	813
26.5.1	寄存器存取注意事项	813
26.5.2	有关清除处理的注意事项	813
第 27 章	通用输入 / 输出端口 (GPIO)	814
27.1	特点	814
27.2	寄存器说明	820
27.2.1	端口 A 控制寄存器 (PTIO_A)	822
27.2.2	端口 B 控制寄存器 (PTIO_B)	823
27.2.3	端口 C 控制寄存器 (PTIO_C)	824
27.2.4	端口 D 控制寄存器 (PTIO_D)	825
27.2.5	端口 E 控制寄存器 (PTIO_E)	826
27.2.6	端口 F 控制寄存器 (PTIO_F)	827

27.2.7	端口 G 控制寄存器 (PTIO_G)	828
27.2.8	端口 H 控制寄存器 (PTIO_H)	829
27.2.9	端口 I 控制寄存器 (PTIO_I)	830
27.2.10	端口 J 控制寄存器 (PTIO_J)	831
27.2.11	端口 A 数据寄存器 (PTDAT_A)	832
27.2.12	端口 B 数据寄存器 (PTDAT_B)	832
27.2.13	端口 C 数据寄存器 (PTDAT_C)	833
27.2.14	端口 D 数据寄存器 (PTDAT_D)	833
27.2.15	端口 E 数据寄存器 (PTDAT_E)	834
27.2.16	端口 F 数据寄存器 (PTDAT_F)	834
27.2.17	端口 G 数据寄存器 (PTDAT_G)	835
27.2.18	端口 H 数据寄存器 (PTDAT_H)	835
27.2.19	端口 I 数据寄存器 (PTDAT_I)	836
27.2.20	端口 J 数据寄存器 (PTDAT_J)	836
27.2.21	输入引脚上拉控制寄存器 (PTPUL_SPCL)	837
27.2.22	引脚选择寄存器 0 (PTSEL_A)	838
27.2.23	引脚选择寄存器 1 (PTSEL_B)	839
27.2.24	引脚选择寄存器 2 (PTSEL_C)	840
27.2.25	引脚选择寄存器 3 (PTSEL_D)	841
27.2.26	引脚选择寄存器 4 (PTSEL_E)	842
27.2.27	引脚选择寄存器 5 (PTSEL_F)	843
27.2.28	引脚选择寄存器 6 (PTSEL_G)	844
27.2.29	引脚选择寄存器 7 (PTSEL_H)	845
27.2.30	引脚选择寄存器 8 (PTSEL_I)	846
27.2.31	引脚选择寄存器 9 (PTSEL_J)	847
27.2.32	引脚选择寄存器 10 (PTSEL_K)	848
27.2.33	引脚选择寄存器 11 (PTSEL_P)	849
27.2.34	引脚选择寄存器 12 (PTSEL_R)	849
27.2.35	引脚选择寄存器 13 (PTSEL_S)	851
27.2.36	HI-Z 寄存器 A (PTHIZ_A)	853
27.2.37	HI-Z 寄存器 B (PTHIZ_B)	854
27.2.38	特殊选择寄存器 (PTSEL_SPCL)	854
27.3	使用例子	855
27.3.1	端口功能的选择	855
27.3.2	端口输出功能	855
27.3.3	端口输入功能	855
27.3.4	外围模块功能	855
第 28 章	低功耗模式	856
28.1	特点	856
28.1.1	功耗模式的种类	856
28.2	输入 / 输出引脚	857
28.3	寄存器说明	857
28.3.1	待机控制寄存器 (STBCR)	858
28.3.2	模块停止寄存器 0 (MSTPCR0)	858
28.3.3	模块停止寄存器 1 (MSTPCR1)	861
28.4	睡眠模式	862
28.4.1	睡眠模式的转移	862
28.4.2	睡眠模式的解除	862
28.5	刷新待机模式	862
28.5.1	刷新待机模式的转移	862
28.5.2	刷新待机模式的解除	862

28.6	模块待机功能	863
28.6.1	模块待机功能的转移	863
28.6.2	模块待机功能的解除	863
28.7	STATUS 引脚的变化时序	863
28.7.1	复位时	863
28.7.2	解除睡眠模式时	863
第 29 章	复位和看门狗定时器 (WDT)	864
29.1	特点	864
29.2	输入 / 输出引脚	865
29.3	寄存器说明	865
29.3.1	看门狗定时器停止时间寄存器 (WDTST)	866
29.3.2	看门狗定时器控制 / 状态寄存器 (WDTCSR)	867
29.3.3	看门狗定时器基准停止时间寄存器 (WDTBST)	868
29.3.4	看门狗定时器计数器 (WDTCNT)	868
29.3.5	看门狗定时器基准计数器 (WDTBCNT)	868
29.4	运行说明	869
29.4.1	复位请求	869
29.4.2	看门狗定时器模式的使用方法	869
29.4.3	间隔定时器模式的使用方法	870
29.4.4	WDT 发生的上溢时间	870
29.4.5	WDT 计数器的清除方法	871
29.5	复位中的引脚时序	871
29.5.1	PRESET 引脚的上电复位	871
29.5.2	通过 WDT 上溢产生的上电复位	873
第 30 章	用户断点控制器 (UBC)	874
30.1	特点	874
30.2	寄存器说明	876
30.2.1	匹配条件设定寄存器 0、1 (CBR0、CBR1)	877
30.2.2	匹配运行设定寄存器 0、1 (CRR0、CRR1)	881
30.2.3	匹配地址设定寄存器 0、1 (CAR0、CAR1)	882
30.2.4	匹配地址屏蔽设定寄存器 0、1 (CAMR0、CAMR1)	883
30.2.5	匹配数据设定寄存器 1 (CDR1)	884
30.2.6	匹配数据屏蔽设定寄存器 1 (CDMR1)	885
30.2.7	执行次数断点寄存器 1 (CETR1)	885
30.2.8	通道匹配标志寄存器 (CCMFR)	886
30.2.9	断点控制寄存器 (CBCR)	886
30.3	运行说明	887
30.3.1	有关存储的术语说明	887
30.3.2	用户断点的运行流程	887
30.3.3	取指令周期的断点	888
30.3.4	操作数存取周期的断点	888
30.3.5	连续断点	889
30.3.6	被保存的程序计数器的值	890
30.4	用户断点调试支持功能	891
30.5	用户断点的使用例子	892
30.6	使用时的注意事项	896
第 31 章	用户调试接口 (H-UDI)	897
31.1	特点	897
31.2	输入 / 输出引脚	898

31.3	边界扫描 TAP 控制器 (IDCODE、EXTEST、SAMPLE/PRELOAD、BYPASS、CLAMP、HIGHZ)	899
31.4	寄存器说明	900
31.4.1	指令寄存器 (SDIR)	901
31.4.2	中断源寄存器 (SDINT)	902
31.4.3	旁路寄存器 (SDBPR)	902
31.4.4	边界扫描寄存器 (SDBSR)	902
31.5	运行说明	918
31.5.1	TAP 控制	918
31.5.2	H-UDI 复位	919
31.5.3	H-UDI 中断	919
31.6	注意事项	919
第 32 章	寄存器一览	920
32.1	寄存器地址一览	920
32.2	各处理模式的寄存器状态	939
第 33 章	电特性	955
33.1	绝对最大额定值	955
33.2	接通和切断电源的顺序	955
33.3	DC 特性	956
33.4	AC 特性	960
33.4.1	时钟 / 控制信号时序	960
33.4.2	控制信号时序	962
33.4.3	总线时序	963
33.4.4	INTC 模块的信号时序	980
33.4.5	DMAC 模块的信号时序	981
33.4.6	TMU 模块的信号时序	981
33.4.7	IIC 模块的信号时序	982
33.4.8	SCIF 模块的信号时序	983
33.4.9	SSI 模块的信号时序	984
33.4.10	ATAPI 接口模块的信号时序	986
33.4.11	USB 模块的信号时序	1005
33.4.12	GPIO 信号时序	1006
33.4.13	H-UDI 模块的信号时序	1006
33.4.14	EtherC 模块的信号时序	1008
33.4.15	FLCTL 模块的信号时序	1011
33.4.16	LCDC 模块的信号时序	1015
33.4.17	VDC2 模块的信号时序	1016
33.5	AC 特性的测量条件	1018
附录		1019
附录 A.	CPU 运行模式寄存器 (CPUOPM)	1019
附录 B.	指令预取及其副作用	1020
附录 C.	子程序返回的预测执行	1021
附录 D.	版本寄存器 (PVR、PRR)	1022
附录 E.	引脚状态	1023
附录 F.	未使用引脚的处理	1029
附录 G.	型号一览表	1035
附录 H.	封装尺寸图	1036
索引		1037

第 1 章 概要

1.1 SH7764 特点

本 LSI 是以瑞萨科技独创的 RISC 方式 CPU 为核心，集聚了系统构成所需外围功能的系统 LSI。

本 LSI 的 CPU 有 RISC (Reduced Instruction Set Computer) 方式的指令系统，由于采用了超标量体系结构，所以飞跃地提高了指令的执行速度。采用了 SH-4A 的 CPU 内核，对于以往的单片机不可能实现的高速性实时控制等应用系统，能构成低成本、高性能和高功能的系统。

本 LSI 分别内置 32K 字节的指令高速缓存和操作数高速缓存，操作数高速缓存可选择回拷和直写模式；内置能存取 4G 字节的虚拟地址空间的存储器管理单元 (MMU)，指令有 4 个入口的全相联 TLB、指令和操作数共用的 64 个入口的全相联 TLB；内置 16K 字节的 SRAM，内部 SRAM 可被高速存取，能用作系统堆栈区和高性能功能的核心部分的常驻区。

本 LSI 内置了能进行高速显示处理的 2D 图形引擎 (G2D)，能经由 LCD 控制器 (LCDC) 等来显示用 G2D 绘图的画面。

作为系统构成所需的外围功能，本 LSI 内置以太网控制器 (EtherC)、USB 主机接口 (V2.0 高速和全速)、ATAPI 控制器 (支持 Ultra DMA)、FIFO 内置型串行通信接口 (SCIF)、I²C 总线接口 (IIC)、专用 DMAC 内置型串行音频接口 (SSI)、32 位定时器 (TMU)、看门狗定时器 (WDT)、中断控制器 (INTC)、I/O 端口等。

另外，本 LSI 能通过外部存储器的存取支持功能，直接与 SDRAM 等存储器和外围 LSI 连接，大幅度降低系统成本。

本 LSI 的特点如表 1.1 所示。

表 1.1 SH7764 特点

项目	特点
CPU	<ul style="list-style-type: none"> • 瑞萨独创的 SuperH 体系结构 (SH-4A) • 目标代码级与 SH-1、SH-2、SH-3、SH-4 兼容 • 32 位内部数据总线 • 通用寄存器： <ul style="list-style-type: none"> — 16 个 32 位通用寄存器 (和 8 个 32 位阴影寄存器) — 7 个 32 位控制寄存器 — 4 个 32 位系统寄存器 — 用于高速中断应答的寄存器组 • RISC 型指令系统 (与 SH 系列向上兼容)： <ul style="list-style-type: none"> — 指令长度：改善代码效率的 16 位固定长 — 加载/存储体系结构 — 延迟转移指令 — 带条件执行 — 基于 C 语言的指令系统 • 含 FPU 的 2 条指令同时执行超标量体系结构 • 指令执行时间：最大 2 条指令 / 周期 • 地址空间：4G 字节 • 空间标识符 ASID：8 位、256 个虚拟地址空间 • 内置乘法器 • 8 段流水线 • 哈佛体系结构
FPU	<ul style="list-style-type: none"> • 内置浮点协处理器 (FPU) • 支持单精度 (32 位) 和双精度 (64 位) • 支持符合 IEEE-754 的数据格式和异常 • 舍入模式：接近方向舍入和零方向舍入 • 非规格化数的处理：舍为 0 或者产生符合 IEEE754 的中断 • 浮点寄存器：32 位 × 16 字 × 2 组 (单精度 × 16 字或者双精度 × 8 字) × 2 组 • 32 位 CPU-FPU 浮点通信寄存器 (FPUL) • 支持 FMAC (乘法和累加：乘加) 指令 • 支持 FDIV (除法) /FSQRT (平方根) 指令 • 支持 FLDI0/FLD1 (加载常数 0/1) 指令 • 指令执行时间 <ul style="list-style-type: none"> — 等待时间 (FADD/FSUB)：3 个周期 (单精度)，5 个周期 (双精度) — 等待时间 (FMAC/FMUL)：5 个周期 (单精度)，7 个周期 (双精度) — 间距 (FADD/FSUB)：1 个周期 (单精度)，1 个周期 (双精度) — 间距 (FMAC/FMUL)：1 个周期 (单精度)，3 个周期 (双精度) • 【注】 FMAC 仅支持单精度。 • 3D 图形指令 (仅单精度) <ul style="list-style-type: none"> — 4 维向量的变换和矩阵运算 (FTRV)：4 个周期 (间距)、8 个周期 (等待时间) — 4 维向量的内积 (FIPR)：1 个周期 (间距)、5 个周期 (等待时间) • 11 段流水线

项目	特点
存储器管理单元 (MMU)	<ul style="list-style-type: none"> • 4G 字节的地址空间，256 个地址空间标识符 (ASID 8 位) • 单虚拟存储模式和多重虚拟存储模式 • 支持多种页大小:1K、4K、8K、64K、256K、1M、4M、64M 字节 • 对指令的 4 个入口的全相联 TLB • 对指令和操作数的 64 个入口的全相联 TLB • 支持软件控制的替换方法和随机计数器方式的替换算法 • 能通过地址映像直接存取 TLB 内容 • 存取权检查功能
高速缓存	<ul style="list-style-type: none"> • 指令高速缓存 (IC) <ul style="list-style-type: none"> — 32K 字节，4 路组相联 — 256 个入口/路，32 字节块长 — 低功耗功能 (通路预测功能) • 操作数高速缓存 (OC) <ul style="list-style-type: none"> — 32K 字节，4 路组相联 — 256 个入口/路，32 字节块长 • 1 段回拷缓冲器，1 段直写缓冲器 • 存储队列 (32 字节 × 2 个入口)
内部存储器 (IL 存储器)	<ul style="list-style-type: none"> • 能高速存取的 16K 字节 RAM • 1 页结构 • 能从 3 个端口存取，具有以下 3 个读写端口 <ul style="list-style-type: none"> — SuperHyway 总线 — 高速缓存/RAM 内置型总线 — 指令总线 • 能从 CPU 存取 8/16/32/64 位操作数 • 能通过外部请求存取 8/16/32/64 位和 16/32 字节
用户断点控制器 (UBC)	<ul style="list-style-type: none"> • 支持由用户断点中断控制的调试 • 2 个断点通道 • 能将地址、数值、存取类型、数据长度全部作为断点条件进行设定 • 支持连续断点功能
时钟振荡器 (CPG)	<ul style="list-style-type: none"> • 主时钟选择电路: EXTAL 输入的时钟频率的 10 ~ 12 倍 • 时钟运行模式: <ul style="list-style-type: none"> — CPU 频率: max 324MHz — 局部总线: max 108MHz — SDRAM: max 108MHz — USB: 48MHz — VDC2: 输入对应显示屏尺寸的时钟频率
看门狗定时器 (WDT)	<ul style="list-style-type: none"> • 支持由计数器上溢进行内部复位的看门狗定时器模式和产生中断的间隔定时器模式 • 在看门狗定时器模式时，将上溢信号输出到外部，并且能指定 LSI 的内部复位 (上电复位)。 • 内置 1 个通道
中断控制器 (INTC)	<ul style="list-style-type: none"> • 直接转移方式 (与 SH-4 兼容) • 外部中断: 3 个外部中断引脚 (NMI、IRQ1、IRQ0) • 内部外围中断: 能对每个模块设定优先级

项目	特点
直接存储器存取控制器 (DMAC)	<ul style="list-style-type: none"> • 通道数: 6 个通道 (其中 2 个通道允许外部请求) • 传送数据长度: 字节、字 (2 字节)、长字 (4 字节)、16 字节、32 字节 • 最大传送字节数: 16,777,216 • 地址模式: 双地址模式 • 总线模式: 能选择周期挪用模式或者突发模式 • 传送请求: 能选择外部请求 (通道 0 和通道 1)、内部外围模块请求、自动请求模式 • 优先顺序: 能选择通道优先顺序固定模式或者循环调度模式
存储器控制单元 (MCU)	<ul style="list-style-type: none"> • 支持外部存储器的存取 <ul style="list-style-type: none"> — 输出 4 个外部存储器的选择信号 — 4 个区域 (FLASH 和 SDRAM), 分别最大为 64M 字节的外部存储空间 • SRAM: 能选择 32 位 / 16 位 / 8 位的数据总线 • SDRAM: 能选择 64 位 / 32 位的数据总线 • 能设定大端法或者小端法 <p>【SRAM 接口】</p> <ul style="list-style-type: none"> • 能连接 NOR 型的闪存 • 周期等待功能: (通过硬件进行的等待控制: 信号) • 避免数据总线冲突的等待控制 (插入空闲周期): <ul style="list-style-type: none"> — 读周期之间的等待设定 — 读周期和写周期之间的等待设定 <p>【SDRAM 接口】</p> <ul style="list-style-type: none"> • 刷新功能: <ul style="list-style-type: none"> — 自动刷新 (内置可编程刷新计数器) — 自刷新 • 时序设定: <p>行列等待时间、列等待时间、行激活周期、写恢复周期、行预充电周期、自动刷新请求间隔、初始化预充电周期数、初始化自动刷新请求间隔</p> • 突发存取方式: 随机列 (SDRAM 突发长度: 32 位总线时为 8, 64 位总线时为 4) • 初始化定时器功能: 发行预充电和自动刷新命令
定时器 (TMU)	<ul style="list-style-type: none"> • 6 个通道自动重加载型 32 位定时器 • 只有通道 2 具有输入捕捉功能 • 各通道都能选择 6 种计数器的输入时钟 <ul style="list-style-type: none"> — 外部时钟、5 种外围时钟 (Pck/4、Pck/16、Pck/64、Pck/256、Pck/1024) (Pck 为外围时钟)
FIFO 内部串行通信接口 (SCIF)	<ul style="list-style-type: none"> • 内置 3 个通道 • 分别内置 16 字节的发送和接收 FIFO • 能选择异步 / 时钟同步模式 • 能进行全双工通信 • 能从波特率发生器的内部时钟或者 SCK 引脚的外部时钟, 选择发送 / 接收的时钟源 • 调制解调器控制功能 (异步模式时)
I ² C 总线接口 (IIC)	<ul style="list-style-type: none"> • 对应飞利浦公司提倡的 I²C 总线 (Inter IC Bus) 接口方式 • 主控 / 从属功能 • 多主控功能 • 支持最大 400Kbps 的传送 • 从系统时钟的时钟可编程生成

项目	特点
ATAPI 接口 (ATAPI)	<ul style="list-style-type: none"> • 支持主通道 • 主控 / 从属功能 • 传送模式： PIO 模式 0 ~ 4、多字 DMA 模式 0 ~ 2、超级 DMA 模式 0 ~ 4 (最大 66Mbps) • 支持 32 字节双缓冲，能进行高速传送 • 支持描述符模式 • 内置 1 个专用 DMAC 通道 • I/O: 对应 3.3V <p>【注】 在 ATAPI 控制器的输入 / 输出引脚中，有主输入 / 输出引脚群和副输入 / 输出引脚群 (镜像引脚)。无论哪个引脚群，一般都进行相同的输入 / 输出操作。但是，不能混合使用 2 个引脚群。</p>
串行音频接口 (SSI)	<ul style="list-style-type: none"> • 6 个通道的双向串行传送 • 支持多种实时音频格式 • 支持主控 / 从属功能 • 字时钟、位时钟的可编程生成功能 • 多通道格式功能 • 支持 8/16/18/20/22/24/32 位的数据格式 • SSI 网络功能 能连接任意音频时钟的通道。例如： <ul style="list-style-type: none"> — 例 1 连接所有音频时钟。 — 例 2 连接通道 0-2 的音频时钟。 — 例 3 分别连接通道 0-1、通道 2-3、通道 4-5 的音频时钟。 — 例 4 独立使用所有通道的音频时钟。 同样，能成组连接任意通道的 SSISCK 引脚 / SSIWS 引脚。此设定能独立指定音频时钟的连接。 • SSI-DMAC 装载了 SSI 专用 DMAC，在 6 个通道的 SSI 与外部存储器或者内部存储器之间进行数据传送。 <ul style="list-style-type: none"> — 通道数：6 个发送通道、6 个接收通道 — 传送数据长度：8 字节、16 字节、32 字节 — 最大传送字节数：4,294,967,296 — 总线模式：周期挪用模式 — 优先顺序：能选择通道优先顺序固定模式或者循环调度模式
以太网控制器 (EtherC)	<ul style="list-style-type: none"> • 以太网的 MAC (Media Access Control) 功能 数据帧的组合 / 分解 (符合 IEEE802.3 的帧格式) CSMA/CD 方式的连接管理 (避免数据冲突、检测到冲突发生时的处理) CRC 计算 内置 FIFO (发送和接收分别为 2K 字节) 全双工 / 半双工发送和接收 短信息包、长信息包的检测 • 对应 MII (Media Independent Interface) 标准规格 站管理 (STA 功能) 传送速率：10/100Mbps • 具有 Magic 包检测功能 (WOL: Wake on LAN 输出)

项目	特点
以太网控制器的 DMAC (E-DMAC)	<ul style="list-style-type: none"> 通过描述符管理方式的数据传送, 减轻 CPU 的负担 从 Ether 接收 FIFO (2K 字节) 向接收缓冲器的传送: 1 个通道 从发送缓冲器向 EtherC 发送 FIFO (2K 字节) 的传送: 1 个通道 通过 32 字节的突发传送, 提高总线的使用效率 能对应单帧、多缓冲器
USB 主机 / 功能接口 (USB)	<ul style="list-style-type: none"> 符合 USB 2.0 支持 480Mbps 和 12Mbps 的传送速度 能用作主机、功能 (可通过软件设定) 内置 PHY 能经过 1 段集线器与多个外围装置连接 作为通信缓冲器, 内置 5K 字节的 RAM
LCD 控制器 (LCDC)	<ul style="list-style-type: none"> 支持 16×1 ~ 1024×1024 点的显示尺寸 支持 4、8、15、16bpp 彩色模式 支持 1、2、4、6bpp 灰度模式 支持 TFT/DSTN/STN 显示器 能设定信号极性 24 位彩色配色存储器 (在 24 位中, R:5/G:6/B:5 的 16 位有效) 统一图形存储器结构
2D 图形引擎 (2D Engine: G2D)	<p>【图形绘图功能】</p> <ul style="list-style-type: none"> 4 顶点的描绘 多角形的描绘 线的描绘 高功能的粗线描绘 图形保真处理 光栅操作 / 带 α 混合的位 BLT <p>【坐标变换功能】</p> <ul style="list-style-type: none"> 进行 4×4 矩阵运算 + 透视法的 W 除法运算 <p>【色彩显示】</p> <ul style="list-style-type: none"> 源极: 1、8、16 位 / 象素, 绘图: 8、16 位 / 象素 工作: 2 值 <p>【屏幕坐标】</p> <ul style="list-style-type: none"> X 方向: 0 ~ 4095, Y 方向: 0 ~ 4095 <p>【寄存器设定】</p> <ul style="list-style-type: none"> 当前指针的设定 局部偏移的设定 特定地址映像寄存器的设定 <p>【顺序控制】</p> <ul style="list-style-type: none"> 等待 Vsync 转移 子程序 (嵌套层: 1)

项目	特点
显示控制器 2 (VDC2)	<p>【图形处理功能】</p> <ul style="list-style-type: none"> 面结构: 图像显示面 4 个面结构 图形的 α 混合、色键功能 (输入的数据对应 RGB16 格式) <p>【输出功能】</p> <ul style="list-style-type: none"> 数字 RGB 输出 (各 6 位) 对应 VESA 规格的显示屏输出 (RGB6:6:6、HD、VD、DE) BTA T-1004 数字 (8:4:4 并列) IF 输出 支持外部同步模式
NAND 闪存控制器 (FLCTL)	<ul style="list-style-type: none"> 能直接连接 NAND 型闪存 以扇区 (512+16 字节) 为单位进行读写, 并且进行 ECC 处理 进行字节单位的读写 内置 256 字节的 FIFO 不对应多层单元 (MLC) 闪存
采样率转换器 (SRC)	<ul style="list-style-type: none"> 数据格式: 立体声 32 位 (L/R 各 16 位)、非立体声 16 位 输入采样率: 8/11.025/12/16/22.05/24/32/44.1/48kHz 输出采样率: 44.1/48kHz
I/O 端口 (GPIO)	<ul style="list-style-type: none"> 77 个通用输入 / 输出端口 输入 / 输出端口能按位进行输入 / 输出转换 端口与中断引脚、内部功能等多路复用
低功耗模式	<ul style="list-style-type: none"> 为降低本 LSI 的功耗, 支持 3 种低功耗模式 <ul style="list-style-type: none"> 睡眠模式: 对 CPU 停止时钟的提供 刷新待机模式: CPU 和内部外围模块停止运行, CPG 继续运行, 并且能继续进行 SDRAM 的刷新。 模块待机功能: 停止对内部外围模块提供时钟
调试接口	<ul style="list-style-type: none"> H-UDI (User Debugging Interface) AUD (Advanced User Debugger)
电源电压	<ul style="list-style-type: none"> VDD, VDD-PLL: 1.2+0.1V VDDQ: 3.3+0.3V
封装	<ul style="list-style-type: none"> BGA-404 引脚 (19mm×19mm)
工艺	<ul style="list-style-type: none"> 90μm CMOS 工艺

1.2 框图

SH7764 框图如图 1.1 所示。

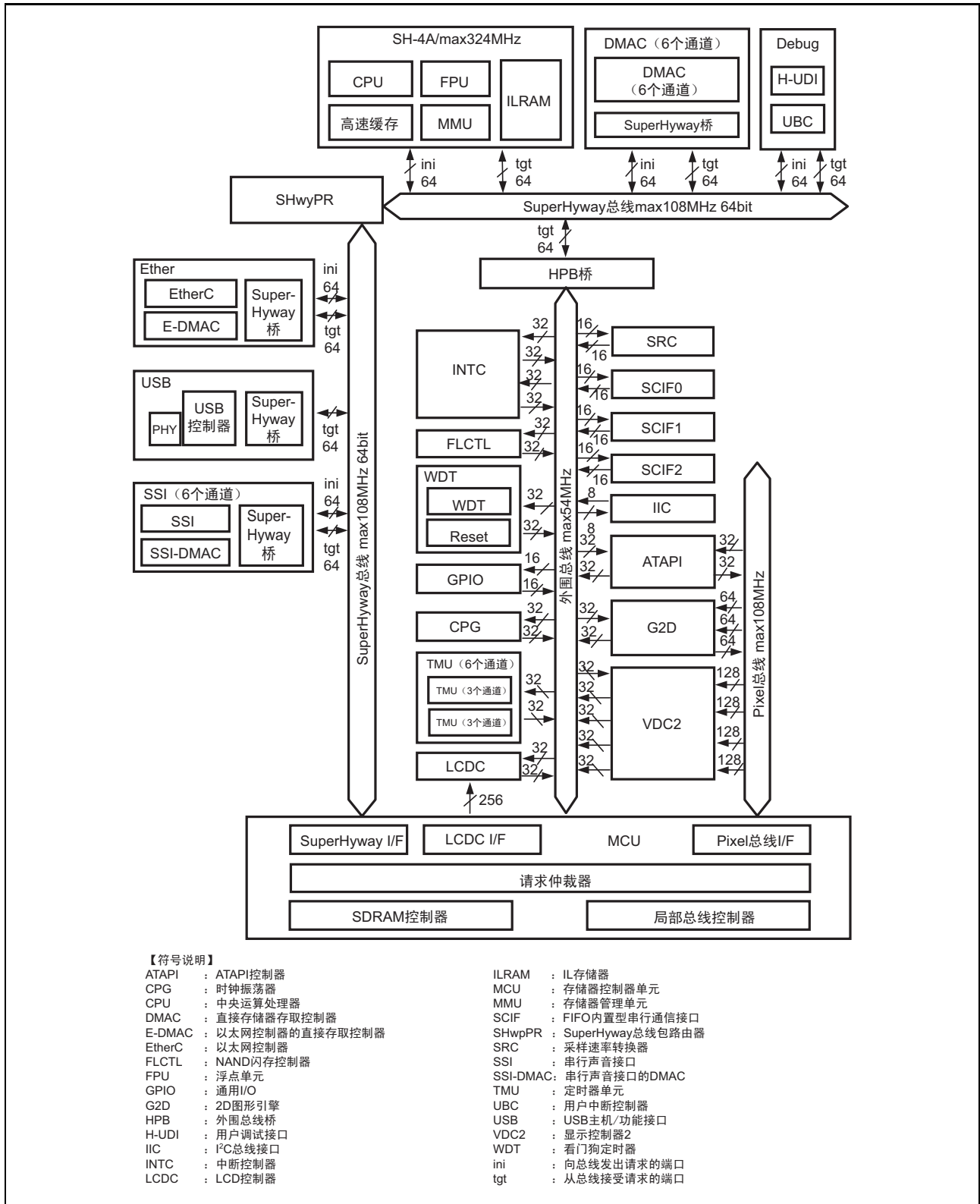


图 1.1 框图

1.4 引脚功能

各引脚功能如表 1.2 所示。

表 1.2 引脚功能

分类	引脚名	输入 / 输出	功能	
电源	VDD	输入	内部电源的电源	是内部核心电源的电源引脚。必须将所有 VDD 引脚连接到系统电源。开路时不工作。
	VSS	输入	内部电源、输入 / 输出电路的接地	是内部核心电源、输入 / 输出电路的接地引脚。必须将所有 VSS 引脚连接到系统电源 (0V)。开路时不工作。
	VDDQ	输入	输入 / 输出电路的电源	是输入 / 输出引脚 (IO) 的电源引脚。必须将所有 VDDQ 引脚连接到系统电源。开路时不工作。
	VDD_PLL1	输入	PLL1 的电源	是内部 PLL1 振荡器的电源引脚。开路时不工作。
	VSS_PLL1	输入	PLL1 的接地	是内部 PLL1 振荡器的接地引脚。开路时不工作。
	VDD_PLL2	输入	PLL2 的电源	是内部 PLL2 振荡器的电源引脚。开路时不工作。
	VSS_PLL2	输入	PLL2 的接地	是内部 PLL2 振荡器的接地引脚。开路时不工作。
时钟	EXTAL	输入	晶体振荡器 / 外部时钟	连接晶体振荡器, EXTAL 引脚也可输入外部时钟。
	XTAL	输出	晶体振荡器	连接晶体振荡器。
	CLKOUT	输出	系统时钟输出	给外部器件提供系统时钟。
运行模式控制	MODE2、 MODE1、 MODE0	输入	时钟运行模式的设定	设定时钟运行模式。不能在运行中改变这些引脚的电平。
	MODE4、 MODE3	输入	总线模式的设定	设定总线运行模式。不能在运行中改变这些引脚的电平。
	MODE5	输入	字节排序的设定	选择 CPU 的字节排序。不能在运行中改变这些引脚的电平。
	MODE7	输入	XIN/XOUT 引脚的功能设定	指定 USB 使用的外部时钟或者晶体振荡器。
	MODE8	输入	EXTAL/XTAL 引脚的功能设定	指定使用的外部时钟或者晶体振荡器。
系统控制	$\overline{\text{PRESET}}$	输入	上电复位	当此引脚为低电平时, 就进入上电复位状态。
	$\overline{\text{WDTOVF}}$	输出	看门狗定时器的上溢	是 WDT 的上溢输出信号。
	$\overline{\text{BREQ}}$	输入	总线请求	在外部器件请求释放总线权时为低电平。
	$\overline{\text{BACK}}$	输出	总线请求的应答	表示总线权释放给外部器件。输出 $\overline{\text{BREQ}}$ 信号的器件能通过接受 $\overline{\text{BACK}}$ 信号, 得知总线权的获得。
中断	NMI	输入	非屏蔽中断	是非屏蔽中断请求引脚。不使用时必须固定为高电平。
	IRQ1、IRQ0	输入	中断请求 1、0	是可屏蔽中断请求引脚。能选择电平或者边沿输入。在边沿输入时, 能选择上升沿、下降沿或者双边沿。
	$\overline{\text{PINT15}} \sim \overline{\text{PINT0}}$	输入	端口中断	是端口的中断请求引脚。使用 PA7 ~ PA0 和 PB7 ~ PB0 产生中断。为低电平输入。
	$\overline{\text{IRQOUT}}$	输出	中断检测	是表示检测、接受到中断请求的状态信号。
地址总线	A25 ~ A0	输出	地址总线	输出地址。
数据总线	D63 ~ D0	输入 / 输出	数据总线	是双向数据总线。

分类	引脚名	输入 / 输出	功能	
运行状态	STATUS1、STATUS0	输出	内部运行状态的表示	用 STATUS[1:0] 表示下列状态 00: 正常状态 01: 待机状态 10: 睡眠状态 11: 复位状态
总线控制	$\overline{CS3} \sim \overline{CS0}$	输出	片选 3 ~ 0	是用于外部存储器或者器件的片选信号。
	\overline{BS}	输出	总线周期的开始	是表示总线周期开始的信号。在多个总线周期构成的总线事务的最初总线周期时, 输出有效电平。
	\overline{RD}	输出	读	表示读外部器件的数据。
	$\overline{R/W}$	输出	读 / 写	表示读写外部器件的状态。读时输出高电平, 写时输出低电平。
	\overline{RDY}	输入	等待	是在存取外部空间时的总线周期中, 表示插入等待周期的输入信号。
	$\overline{WE0}$	输出	字节的指定	表示写外部存储器或者器件的数据的 bit7 ~ 0 (在 8/16/32 位总线存取时)。与 DQM64LL 引脚兼用。
	$\overline{WE1}$	输出	字节的指定	表示写外部存储器或者器件的数据的 bit15 ~ 8 (在 16/32 位总线存取时)。与 DQM64LU 引脚兼用。
	$\overline{WE2}$	输出	字节的指定	表示写外部存储器或者器件的数据的 bit23 ~ 16 (在 32 位总线存取时)。与 DQM64UL 引脚兼用。
总线控制 DRAM 接口	\overline{RAS}	输出	RAS	在连接 SDRAM 时, 连接 \overline{RAS} 引脚。
	\overline{CAS}	输出	CAS	在连接 SDRAM 时, 连接 \overline{CAS} 引脚。
	CKE	输出	CK 允许	在连接 SDRAM 时, 连接 CKE 引脚。
	DQMUU	输出	字节选择 0	在连接 SDRAM 时, 选择数据总线的 bit31 ~ 24。
	DQMUL	输出	字节选择 1	在连接 SDRAM 时, 选择数据总线的 bit23 ~ 16。
	DQMLU	输出	字节选择 2	在连接 SDRAM 时, 选择数据总线的 bit15 ~ 8。
	DQMLL	输出	字节选择 3	在连接 SDRAM 时, 选择数据总线的 bit7 ~ 0。
	DQM64UU	输出	64 位字节选择 0	在连接 SDRAM 时, 选择数据总线的 bit63 ~ 56。与 $\overline{WE3}$ 引脚兼用。
	DQM64UL	输出	64 位字节选择 1	在连接 SDRAM 时, 选择数据总线的 bit55 ~ 48。与 $\overline{WE2}$ 引脚兼用。
	DQM64LU	输出	64 位字节选择 2	在连接 SDRAM 时, 选择数据总线的 bit47 ~ 40。与 $\overline{WE1}$ 引脚兼用。
	DQM64LL	输出	64 位字节选择 3	在连接 SDRAM 时, 选择数据总线的 bit39 ~ 32。与 $\overline{WE0}$ 引脚兼用。
显示控制器 2 (VDC2)	DR5 ~ DR0	输出	数字红色数据的输出	是视频数据的输出引脚。
	DG5 ~ DG0	输出	数字绿色数据的输出	是视频数据的输出引脚。
	DB5 ~ DB0	输出	数字蓝色数据的输出	是视频数据的输出引脚。

分类	引脚名	输入 / 输出	功能	
显示控制器 2 (VDC2)	VSYNC/SPS	输出	垂直同步信号 / 栅极开始信号	是垂直同步信号 / 栅极开始信号。
	HSYNC/SPL	输出	水平同步信号 / 采样开始信号	是水平同步信号 / 采样开始信号。
	DE_V/CLS	输出	垂直数据允许信号 / 栅极时钟信号	是垂直数据的允许信号 / 栅极时钟信号。
	DE_H/DE_C	输出	水平数据允许信号 / 显示允许信号	是水平数据的允许信号 / 显示允许信号。
	COM/CDE	输出	栅极控制信号 / 色度数据允许信号	是栅极控制信号 / 显示允许 (当和寄存器设定的色键对象色相同时, 输出有效电平)。
	BT_DATA7 ~ BT_DATA0	输入 / 输出	BTA-T1004 显示数据	是 BTA-T1004 显示数据的输出引脚。
	BT_HSYNC	输出	BTA-T1004 水平同步	是用于 BTA-T1004 的水平同步信号。
	BT_VSYNC	输出	BTA-T1004 垂直同步	是用于 BTA-T1004 的垂直同步信号。
	BT_DE_C	输出	BTA-T1004 显示允许	是 BTA-T1004 显示允许信号。
	EX_HSYNC	输入	HSYNC 输入	在外部同步模式时, 输入 HSYNC 信号。
	EX_VSYNC	输入	VSYNC 输入	在外部同步模式时, 输入 VSYNC 信号。
	DCLKIN	输入	显示屏源极时钟输入	输入用于显示的源极时钟。输入适合显示屏尺寸的频率。
	DCLKOUT	输出	显示屏时钟输出	是显示屏时钟的输出引脚。
	直接存储器存取 (DMAC)	$\overline{\text{DREQ0}}$ 、 $\overline{\text{DREQ1}}$	输入	DMA 传送请求
$\overline{\text{DACK0}}$ 、 $\overline{\text{DACK1}}$		输出	DMA 传送请求的应答	是对外部 DMA 传送请求输出 DMA 应答的引脚。
$\overline{\text{DTEND0}}$ 、 $\overline{\text{DTEND1}}$		输出	DMA 传送结束的输出	是 DMA 传送结束的输出信号。
以太网控制器 (EtherC)	CRS	输入	载波检测	是载波检测信号的输入引脚。
	COL	输入	冲突	是信号冲突检测的信号输入引脚。
	MII_TXD3 ~ MII_TXD0	输出	发送数据	是 4 位发送数据的引脚。连接到 PHY 的数据发送引脚。
	TX_EN	输出	发送允许	表示 MII_TXD 引脚的发送数据准备完毕。
	TX_CLK	输入	发送时钟	是 TX_EN、TX_ER、MII_TXD 的时钟信号。
	TX_ER	输出	发送错误	是将发送错误通知 PHY-LSI 的信号。
	MII_RXD3 ~ MII_RXD0	输入	接收数据	是 4 位发送数据的引脚。连接到 PHY 的数据接收引脚。
	RX_DV	输入	接收数据有效	表示 MII_RXD 引脚的接收数据准备完毕。
	RX_CLK	输入	接收时钟	是 RX_DV、RX_ER、MII_RXD 的时钟信号。
	RX_ER	输入	接收错误	是通知接收发生错误的引脚。
	MDC	输出	管理用的时钟	是通过 MDIO 传送信息的时钟信号。
	MDIO	输入 / 输出	管理用的数据	是用于交换管理信息的双向数据引脚。
	WOL	输出	MAGIC 包的接收	是用于接收 Magic 包的引脚。
	LNKSTA	输入	连接状态	是从 PHY-LSI 输入连接状态的引脚。
EXOUT	输出	通用输出	是外部输出引脚。	

分类	引脚名	输入 / 输出	功能	
ATAPI 接口 (ATAPI)	IDED15 ~ IDED0、 IDED15_M ~ IDED0_M	输入 / 输出	IDE 数据总线	是双向数据总线。 IDED15_M ~ IDED0_M 为镜像引脚。
	IDEA2 ~ IDEA0、 IDEA2_M ~ IDEA0_M	输出	IDE 地址总线	是 IDE 的地址输出引脚。 IDEA2_M ~ IDEA0_M 为镜像引脚。
	$\overline{\text{IODACK}}$ 、 $\overline{\text{IODACK_M}}$	输出	IDEDMA 应答	是主通道的 DMA 应答信号 (低电平有效)。 $\overline{\text{IODACK_M}}$ 为镜像引脚。
	$\overline{\text{IODREQ}}$ 、 $\overline{\text{IODREQ_M}}$	输入	IDEDMA 请求	是主通道的 DMA 请求信号 (高电平有效)。 $\overline{\text{IODREQ_M}}$ 为镜像引脚。
	$\overline{\text{IDECS1}}$ 、 $\overline{\text{IDECS0}}$ 、 $\overline{\text{IDECS1_M}}$ 、 $\overline{\text{IDECS0_M}}$	输出	IDE 片选	是主通道片选信号 (低电平有效)。 $\overline{\text{IDECS1_M}}$ 、 $\overline{\text{IDECS0_M}}$ 为镜像引脚。
	$\overline{\text{IDEIOWR}}$ 、 $\overline{\text{IDEIOWR_M}}$	输出	IDE 写信号	是主通道的写信号 (低电平有效)。 $\overline{\text{IDEIOWR_M}}$ 为镜像引脚。
	$\overline{\text{IDEIORD}}$ 、 $\overline{\text{IDEIORD_M}}$	输出	IDE 读信号	是主通道的读信号 (低电平有效)。 $\overline{\text{IDEIORD_M}}$ 为镜像引脚。
	$\overline{\text{IDEIORDY}}$ 、 $\overline{\text{IDEIORDY_M}}$	输入	IDE 就绪信号	是主通道的就绪信号 (高电平有效)。 $\overline{\text{IDEIORDY_M}}$ 为镜像引脚。
	$\overline{\text{IDEINT}}$ 、 $\overline{\text{IDEINT_M}}$	输入	IDE 中断信号	是主通道的中断请求信号 (高电平有效)。 $\overline{\text{IDEINT_M}}$ 为镜像引脚。
	$\overline{\text{IDERST}}$ 、 $\overline{\text{IDERST_M}}$	输出	IDE 复位信号	是主通道的 ATAPI 设备复位信号 (高电平有效)。 $\overline{\text{IDERST_M}}$ 为镜像引脚。
	$\overline{\text{DIRECTION}}$ 、 $\overline{\text{DIRECTION_M}}$	输出	方向信号	是外部电平移位器的方向信号 (高电平有效)。 $\overline{\text{DIRECTION_M}}$ 为镜像引脚。
	FIFO 内部串行 通信接口 (SCIF)	SCK0、 SCK1、SCK2	输入 / 输出	串行时钟
TXD0、TXD1、 TXD2		输出	发送数据	是串行数据的输出引脚。
RXD0、 RXD1、RXD2		输入	接收数据	是串行数据的输入引脚。
$\overline{\text{CTS0}}$ 、 $\overline{\text{CTS1}}$ 、 $\overline{\text{CTS2}}$		输入 / 输出	调制解调器控制发 送可能信号	是控制调制解调器的停止 / 重新开始发送数据的信号。
$\overline{\text{RTS0}}$ 、 $\overline{\text{RTS1}}$ 、 $\overline{\text{RTS2}}$		输入 / 输出	调制解调器控制发 送请求信号	是控制调制解调器的停止 / 重新开始接收数据的信号。
I ² C 总线接口 (IIC)	SCL	输入 / 输出	串行时钟	是串行时钟的输入 / 输出引脚。
	SDA	输入 / 输出	串行数据	是串行数据的输入 / 输出引脚。
USB 主机 / 功能 控制器 (USB)	XIN	输入	USB 使用的晶体 振荡器 / 外部时钟	连接 USB 使用的晶体振荡器或者外部时钟。
	XOUT	输出	USB 使用的晶体 振荡器	连接 USB 使用的晶体振荡器。
	DP	输入 / 输出	D+ 信号	USB D+ 信号
	DM	输入 / 输出	D- 信号	USB D- 信号
	VBUS	输入	Vbus 信号	USB Vbus 信号

分类	引脚名	输入 / 输出	功能	
USB 主机 / 功能 控制器 (USB)	REFRIN	输入	基准输入	通过 5.6kΩ (±1%) 的电阻连接模拟接地
	VDD_USB	数字电源	USB PHY 数字部 分电源	是 USB PHY 数字部的电源。输入 1.2V。
	VSS_USB	数字接地	USB PHY 数字部 的接地	是 USB PHY 数字部的接地。输入 0V。
	VDDQ_USB	数字电源	USB PHY 数字部 的电源	是 USB PHY 数字部的电源。输入 3.3V。
	VSSQ_USB	数字接地	USB PHY 数字部 的接地	是 USB PHY 数字部的接地。输入 0V。
	VDDA_USB	模拟电源	USB PHY 模拟部 的电源	是 USB PHY 模拟部的电源。输入 1.2V。
	VSSA_USB	模拟接地	USB PHY 模拟部 的接地	是 USB PHY 模拟部的接地。输入 0V。
	VDDQA_USB	模拟电源	USB PHY 模拟部 的电源	是 USB PHY 模拟部的电源。输入 3.3V。
	VSSQA_USB	模拟接地	USB PHY 模拟部 的接地	是 USB PHY 模拟部的接地。输入 0V。
	UV12	模拟电源	USB480MHz 的电 源	是 USB 的 480MHz 的工作电源。输入 1.2V。
UG12	模拟接地	USB480MHz 的接 地	是 USB 的 480MHz 的工作电源。输入 0V。	
32 位定时器 (TMU)	TCLK	输入	定时器时钟	是定时器的外部时钟输入。 也能用作通道 2 的输入捕捉信号。
串行音频接口 (SSI)	SSIDATA0、 SSIDATA1、 SSIDATA2、 SSIDATA3、 SSIDATA4、 SSIDATA5	输入 / 输出	SSI 数据的输入 / 输出	是串行数据的输入 / 输出引脚。
	SSISCK0、 SSISCK1、 SSISCK2、 SSISCK3、 SSISCK4、 SSISCK5	输入 / 输出	SSI 时钟的输入 / 输出	是串行时钟的输入 / 输出引脚。
	SSIWS0、 SSIWS1、 SSIWS2、 SSIWS3、 SSIWS4、 SSIWS5	输入 / 输出	SSI 时钟 LR 的输 入 / 输出	是字选择的输入 / 输出引脚。
	AUDIO_CLK0、 AUDIO_CLK1、 AUDIO_CLK2、 AUDIO_CLK3、 AUDIO_CLK4、 AUDIO_CLK5	输入	SSI 音频的外部时 钟	输入用于音频的外部时钟。将此时钟输入到分频器后 使用。

分类	引脚名	输入 / 输出	功能	
LCD 控制器 (LCDC)	LCD_DATA15 ~ LCD_DATA0	输出	LCD 数据	是 LCD 显示屏的数据输出引脚。
	LCD_DON	输出	显示开始信号	是显示开始信号 (DON) 的引脚。
	LCD_CL1	输出	移位时钟 1	是 LCD 移位时钟 1/ 水平同步信号的引脚。
	LCD_CL2	输出	移位时钟 2	是 LCD 移位时钟 2/ 点时钟的引脚。
	LCD_CLK	输入	时钟源	是 LCD 时钟源的输入引脚。
	LCD_FLM	输出	行标记	是首行标记 / 垂直同步信号的引脚。
	LCD_VCPWC	输出	电源控制 (VCC)	是 LCD 液晶模块的电源控制 (VCC) 引脚。
	LCD_VEPWC	输出	电源控制 (VEE)	是 LCD 液晶模块的电源控制 (VEE) 引脚。
	LCD_M_DISP	输出	液晶交流化	是液晶交流化信号 / DISP 信号的引脚。
NAND 闪存 控制器 (FLCTL)	FCE	输出	芯片允许	是芯片允许引脚。
	FD7 ~ FD0	输入 / 输出	数据输入 / 输出	是命令、地址、数据的输入 / 输出引脚。
	FCLE	输出	命令锁存允许	是命令锁存的允许 (CLE) 引脚。 在命令输出时为有效电平。
	FALE	输出	地址锁存允许	是地址锁存的允许 (ALE) 引脚。 在地址输出时为有效电平。 在数据输入 / 输出时为无效电平。
	$\overline{\text{FRE}}$	输出	读允许	是读允许 (RE) 的引脚。 在 RE 的下降沿读取数据。
	$\overline{\text{FWE}}$	输出	写允许	是写允许的引脚。 在 WE 的上升沿, 闪存锁存命令、地址和数据。
	$\overline{\text{FR/B}}$	输入	就绪 / 忙	就绪 / 忙。表示高电平为就绪状态、低电平为忙状态。
I/O 端口 (GPIO)	PA7 ~ PA0	输入 / 输出	通用端口	是 8 位通用输入 / 输出端口。
	PB7 ~ PB0	输入 / 输出	通用端口	是 8 位通用输入 / 输出端口。
	PC7 ~ PC0	输入 / 输出	通用端口	是 8 位通用输入 / 输出端口。
	PD7 ~ PD0	输入 / 输出	通用端口	是 8 位通用输入 / 输出端口。
	PE7 ~ PE0	输入 / 输出	通用端口	是 8 位通用输入 / 输出端口。
	PF7 ~ PF0	输入 / 输出	通用端口	是 8 位通用输入 / 输出端口。
	PG7 ~ PG0	输入 / 输出	通用端口	是 8 位通用输入 / 输出端口。
	PH7 ~ PH0	输入 / 输出	通用端口	是 8 位通用输入 / 输出端口。
	PI4 ~ PI0	输入 / 输出	通用端口	是 5 位通用输入 / 输出端口。
PJ7 ~ PJ0	输入 / 输出	通用端口	是 8 位通用输入 / 输出端口。	
用户调试接口 (H-UDI)	TCK	输入	测试时钟	是测试时钟的输入引脚。
	TMS	输入	测试模式的选择	是测试模式选择信号的输入引脚。
	$\overline{\text{TRST}}$	输入	测试复位	是初始化信号的输入引脚。
	TDI	输入	测试数据的输入	是指令和数据的串行输入引脚。
	TDO	输出	测试数据的输出	是指令和数据的串行输出引脚。
高级用户调试 器 (AUD)	AUDATA3 ~ AUDATA0、 AUDCK、 AUDSYNC	输出	仿真器用的引脚	使仿真器的专用引脚
	$\overline{\text{ASEBRK/}}\br/>BRKACK$	输入 / 输出	仿真器用的引脚	是仿真器的专用引脚
	MPMD	输入	芯片模式的指定 引脚	指定工作在仿真支持模式 (MPMD= 低电平) 或者工 工作在芯片模式 (MPMD= 高电平)。

1.5 地址映像

本 LSI 内部的地址映像如图 1.3 所示。

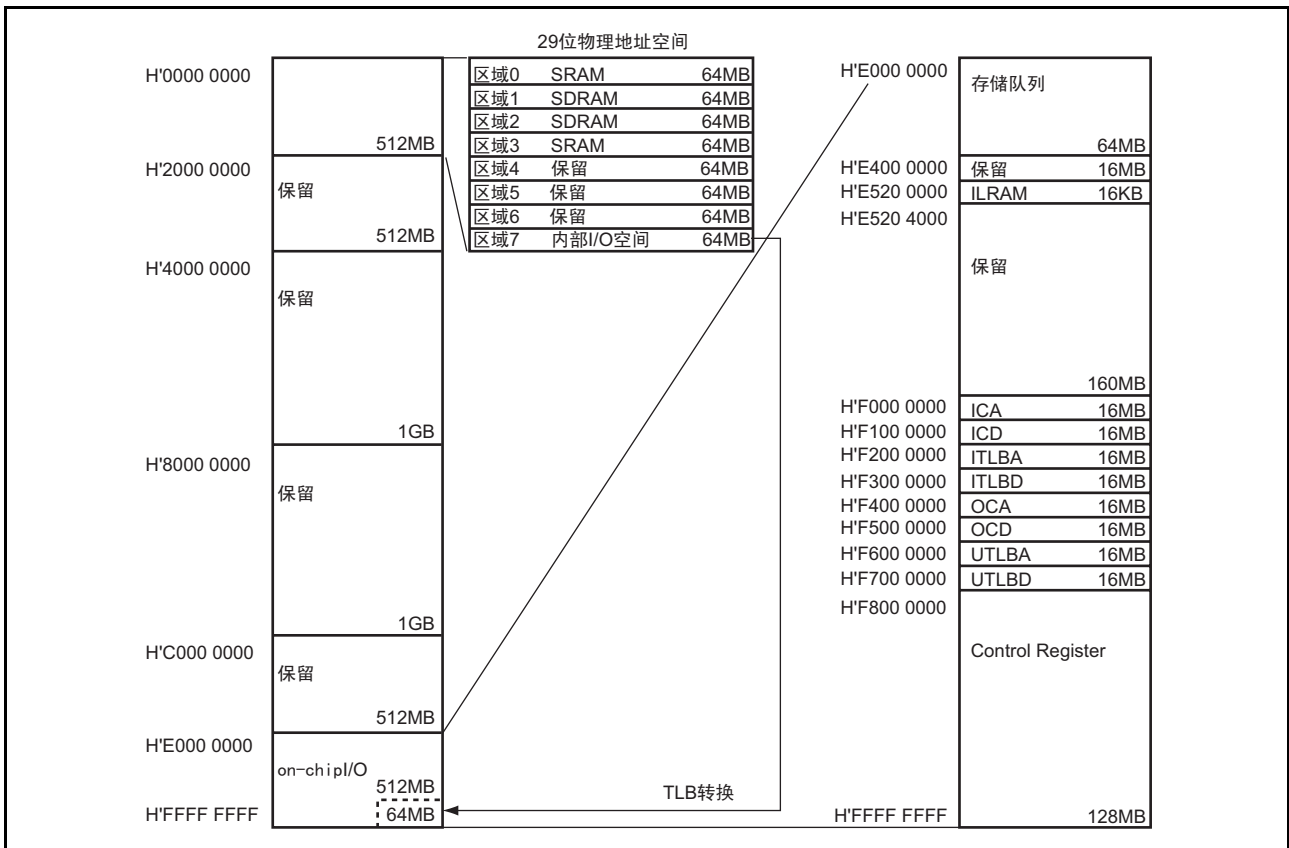


图 1.3 物理地址空间 (1) .

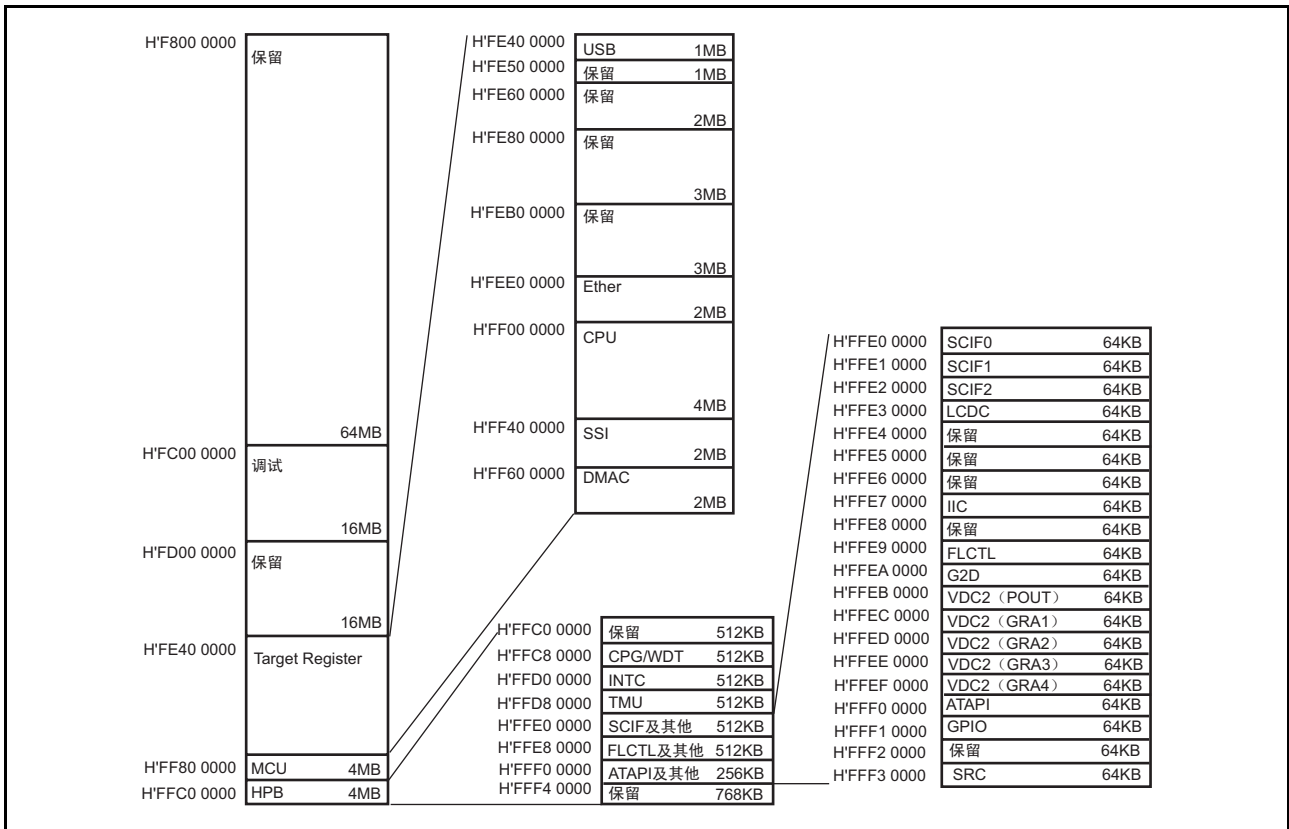


图 1.3 物理地址空间 (2)

第 2 章 编程模型

本章记述 SH-4A 的编程模型。SH-4A 有以下所示的寄存器和数据格式。

2.1 数据格式

SH-4A 支持的数据格式如图 2.1 所示。

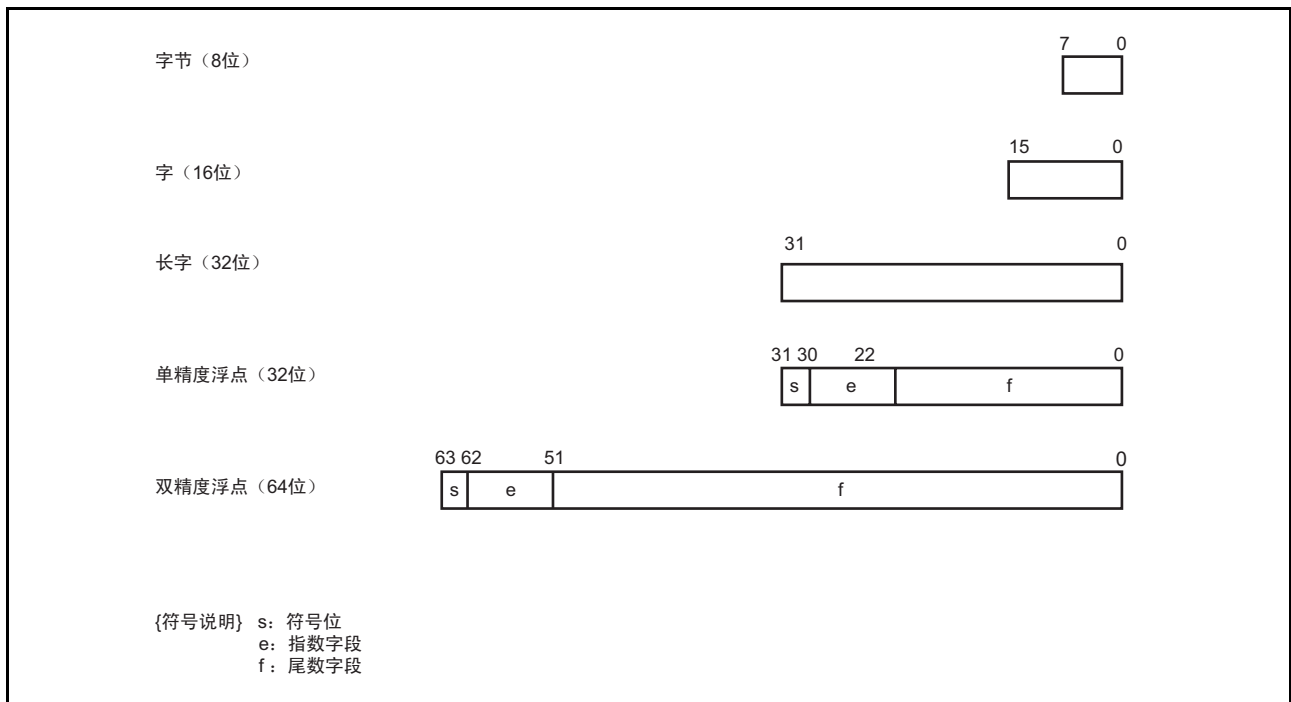


图 2.1 数据格式

2.2 寄存器结构

2.2.1 特权模式和存储体

(1) 处理模式

处理模式有用户模式和特权模式 2 种。通常，在用户模式运行，当发生异常或者接受中断时变为特权模式。寄存器有通用寄存器、系统寄存器、控制寄存器和浮点寄存器，能存取的寄存器因各自的处理模式而不同。

(2) 通用寄存器

通用寄存器有 R0 ~ R15 的 16 个寄存器。通用寄存器 R0 ~ R7 为存储体寄存器，能通过处理模式进行转换。

- 特权模式的情况

根据状态寄存器 (SR) 的寄存器存储体位 (RB)，决定是否作为通用寄存器进行存取。不能作为通用寄存器进行存取的寄存器，通过控制寄存器的加载指令 (LDC) 和存储指令 (STC) 进行存取。当 RB 位为 1 时 (选择存储体 1 时)，存储体 1 的通用寄存器 R0_BANK1 ~ R7_BANK1 以及与存储体无关的 R8 ~ R15 的 16 个寄存器能作为通用寄存器 R0 ~ R15 进行存取，存储体 0 的通用寄存器 R0_BANK0 ~ R7_BANK0 的 8 个寄存器能通过 LDC/STC 指令进行存取。

当RB位为0时（选择存储体0时），存储体0的通用寄存器R0_BANK0～R7_BANK0以及与存储体无关的R8～R15的16个寄存器能作为通用寄存器R0～R15进行存取，存储体1的通用寄存器R0_BANK1～R7_BANK1的8个寄存器能通过LDC/STC指令进行存取。

- 用户模式的情况

存储体0的通用寄存器R0_BANK0～R7_BANK0以及与存储体无关的R8～R15的16个寄存器能作为通用寄存器R0～R15进行存取，存储体1的通用寄存器R0_BANK1～R7_BANK1的8个寄存器不能存取。

(3) 控制寄存器

在处理模式中，控制寄存器有共用的全局基址寄存器（GBR）和状态寄存器（SR）以及只能在特权模式中存取的保存状态寄存器（SSR）、保存程序计数器（SPC）、向量基址寄存器（VBR）、保存通用寄存器15（SGR）、调试基址寄存器（DBR）。状态寄存器有只能在特权模式中存取的位（如RB位）。

(4) 系统寄存器

系统寄存器有乘加寄存器（MACH/MACL）、过程寄存器（PR）和程序计数器（PC），与处理模式无关。

(5) 浮点寄存器和 FPU 的相关系统寄存器

浮点寄存器有FR0～FR15和XF0～XF15共32个寄存器。能将FR0～FR15和XF0～XF15分配到各自的FPR0_BANK0～FPR15_BANK0或者FPR0_BANK1～FPR15_BANK1存储体。

FR0～FR15能用作DR0/2/4/6/8/10/12/14（双精度浮点寄存器或者寄存器对）或者FV0/4/8/12（寄存器向量），XF0～XF15能用作XD0/2/4/6/8/10/12/14（寄存器对）或者XMTRX（寄存器矩阵）。

FPU的相关系统寄存器有浮点通信寄存器（FPUL）和浮点状态/控制寄存器（FPSCR），进行FPU-CPU间的通信和异常处理的设定。

复位后的寄存器值如表 2.1 所示。

表 2.1 寄存器的初始值

分类	寄存器	初始值*
通用寄存器	R0_BANK0～R7_BANK0、 R0_BANK1～R7_BANK1、 R8～R15	不定值
控制寄存器	SR	MD位、RB位和BL位为1，FD位为0， IMASK为B'1111，保留位为0，其他为不定值。
	GBR、SSR、SPC、SGR、DBR	不定值
	VBR	H'00000000
系统寄存器	MACH、MACL、PR	不定值
	PC	H'A0000000
浮点寄存器	FR0～FR15、XF0～XF15、FPUL	不定值
	FPSCR	H'00040001

【注】* 在上电复位时被初始化。

各处理模式的CPU寄存器结构如图 2.2 所示。

通过状态寄存器的处理模式位（MD）进行用户模式和特权模式的转换。

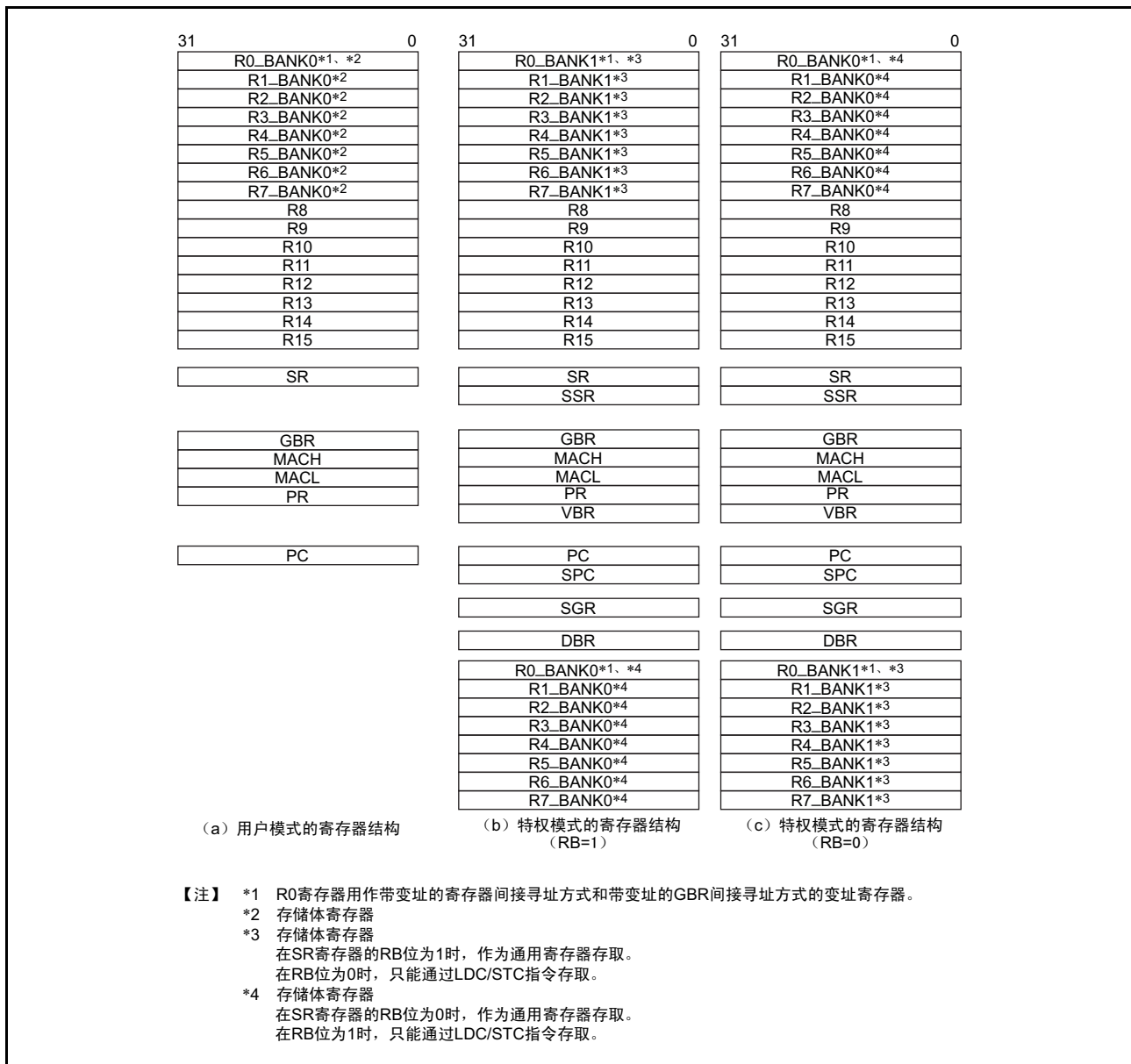


图 2.2 各处理模式的 CPU 寄存器结构

2.2.2 通用寄存器

处理模式和通用寄存器的关系如图 2.3 所示。SH-4A 有 24 个 32 位通用寄存器（R0_BANK0 ~ R7_BANK0、R0_BANK1 ~ R7_BANK1、R8 ~ R15），但是其中只有 16 个寄存器能在一个处理模式中作为通用寄存器 R0 ~ R15 进行存取。SH-4A 有特权模式和用户模式 2 个处理模式，根据这 2 个模式对 R0 ~ R7 进行以下分配：

- R0_BANK0 ~ R7_BANK0
在用户模式（SR.MD=0）中，总是分配到 R0 ~ R7。
在特权模式（SR.MD=1）中，只在（SR.RB=0）时分配到 R0 ~ R7。
- R0_BANK1 ~ R7_BANK1
在用户模式中，不能存取。
在特权模式中，只在（SR.RB=1）时分配到 R0 ~ R7。

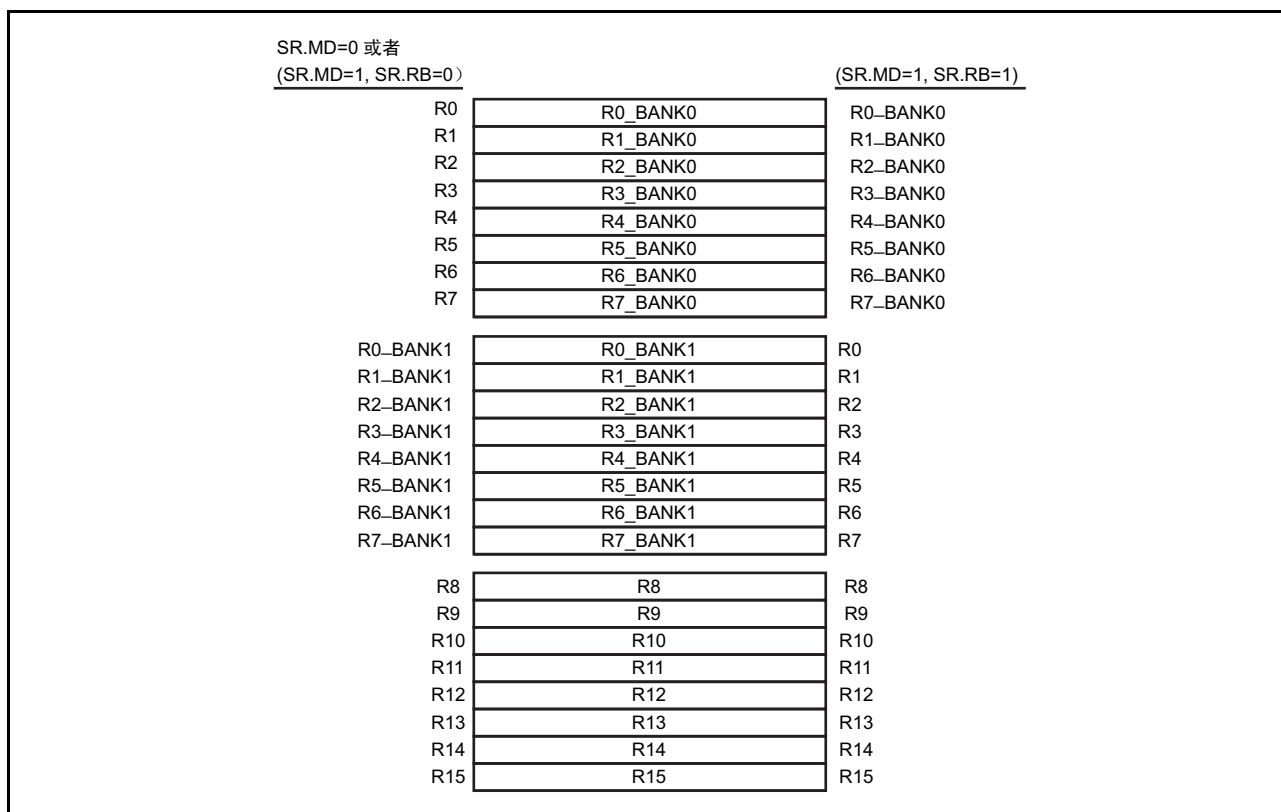


图 2.3 通用寄存器

【编程注意事项】

因为用户模式的 R0~R7 被分配到 R0_BANK0~R7_BANK0 并且异常和中断后的 R0~R7 被分配到 R0_BANK1~R7_BANK1，所以中断处理程序不需要保存或者恢复用户模式的 R0~R7（R0_BANK0~R7_BANK0）。

2.2.3 浮点寄存器

浮点寄存器如图 2.4 所示，有 32 个 32 位浮点寄存器。这些寄存器由 2 个存储体 FPR0_BANK0~FPR15_BANK0 和 FPR0_BANK1~FPR15_BANK1 构成。这 32 个寄存器作为 FR0~FR15、DR0/2/4/6/8/10/12/14、FV0/4/8/12、XF0~XF15、XD0/2/4/6/8/10/12/14、XMTRX 被参照。FPRn_BANKi 和参照名的对应由 FPSCR 的 FR 位决定，请参照图 2.4。

(1) 浮点寄存器 FPRn_BANKi（32 个寄存器）

FPR0_BANK0、FPR1_BANK0、FPR2_BANK0、FPR3_BANK0、
 FPR4_BANK0、FPR5_BANK0、FPR6_BANK0、FPR7_BANK0、
 FPR8_BANK0、FPR9_BANK0、FPR10_BANK0、FPR11_BANK0、
 FPR12_BANK0、FPR13_BANK0、FPR14_BANK0、FPR15_BANK0
 FPR0_BANK1、FPR1_BANK1、FPR2_BANK1、FPR3_BANK1、
 FPR4_BANK1、FPR5_BANK1、FPR6_BANK1、FPR7_BANK1、
 FPR8_BANK1、FPR9_BANK1、FPR10_BANK1、FPR11_BANK1、
 FPR12_BANK1、FPR13_BANK1、FPR14_BANK1、FPR15_BANK1

(2) 单精度浮点寄存器 FRi (16 个寄存器)

当 FPSCR.FR=0 时, FR0 ~ FR15 分配到 FPR0_BANK0 ~ FPR15_BANK0。

当 FPSCR.FR=1 时, FR0 ~ FR15 分配到 FPR0_BANK1 ~ FPR15_BANK1。

(3) 双精度浮点寄存器对或者单精度浮点寄存器的寄存器对 DRi (8 个寄存器)

DR 寄存器由 2 个 FR 寄存器构成。

DR0={FR0, FR1}、DR2={FR2, FR3}、

DR4={FR4, FR5}、DR6={FR6, FR7}、

DR8={FR8, FR9}、DR10={FR10, FR11}、

DR12={FR12, FR13}、DR14={FR14, FR15}

(4) 单精度浮点向量寄存器 FVi (4 个寄存器)

FV 寄存器由 4 个 FR 寄存器构成。

FV0={FR0, FR1, FR2, FR3}、

FV4={FR4, FR5, FR6, FR7}、

FV8={FR8, FR9, FR10, FR11}、

FV12={FR12, FR13, FR14, FR15}

(5) 单精度浮点扩展寄存器 XFi (16 个寄存器)

当 FPSCR.FR=0 时, XF0 ~ XF15 分配到 FPR0_BANK1 ~ FPR15_BANK1。

当 FPSCR.FR=1 时, XF0 ~ XF15 分配到 FPR0_BANK0 ~ FPR15_BANK0。

(6) 单精度浮点扩展寄存器的寄存器对 XD_i (8 个寄存器)

XD 寄存器由 2 个 XF 寄存器构成。

XD0={XF0, XF1}、XD2={XF2, XF3}、

XD4={XF4, XF5}、XD6={XF6, XF7}、

XD8={XF8, XF9}、XD10={XF10, XF11}、

XD12={XF12, XF13}、XD14={XF14, XF15}

(7) 单精度浮点扩展寄存器矩阵 XMTRX

XMTRX 由 16 个 XF 寄存器构成。

$$\text{XMTRX} = \begin{pmatrix} \text{XF0} & \text{XF4} & \text{XF8} & \text{XF12} \\ \text{XF1} & \text{XF5} & \text{XF9} & \text{XF13} \\ \text{XF2} & \text{XF6} & \text{XF10} & \text{XF14} \\ \text{XF3} & \text{XF7} & \text{XF11} & \text{XF15} \end{pmatrix}$$

FPSCR.FR=0			FPSCR.FR=1			
FV0	DR0	FR0	FPR0_BANK0	XF0	XD0	XMTRX
		FR1	FPR1_BANK0	XF1		
FV4	DR2	FR2	FPR2_BANK0	XF2	XD2	
		FR3	FPR3_BANK0	XF3		
		FR4	FPR4_BANK0	XF4		
FV8	DR6	FR5	FPR5_BANK0	XF5	XD6	
		FR6	FPR6_BANK0	XF6		
		FR7	FPR7_BANK0	XF7		
FV12	DR8	FR8	FPR8_BANK0	XF8	XD8	
		FR9	FPR9_BANK0	XF9		
		FR10	FPR10_BANK0	XF10		
FV12	DR10	FR11	FPR11_BANK0	XF11	XD12	
		FR12	FPR12_BANK0	XF12		
		FR13	FPR13_BANK0	XF13		
FV12	DR12	FR14	FPR14_BANK0	XF14	XD14	
		FR15	FPR15_BANK0	XF15		
XMTRX	XD0	XF0	FPR0_BANK1	FR0	DR0	FV0
		XF1	FPR1_BANK1	FR1		
XMTRX	XD2	XF2	FPR2_BANK1	FR2	DR2	
		XF3	FPR3_BANK1	FR3		
		XF4	FPR4_BANK1	FR4		
XMTRX	XD4	XF5	FPR5_BANK1	FR5	DR6	
		XF6	FPR6_BANK1	FR6		
		XF7	FPR7_BANK1	FR7		
XMTRX	XD6	XF8	FPR8_BANK1	FR8	DR8	FV8
		XF9	FPR9_BANK1	FR9		
		XF10	FPR10_BANK1	FR10		
XMTRX	XD8	XF11	FPR11_BANK1	FR11	DR12	FV12
		XF12	FPR12_BANK1	FR12		
		XF13	FPR13_BANK1	FR13		
XMTRX	XD10	XF14	FPR14_BANK1	FR14	DR14	
		XF15	FPR15_BANK1	FR15		

图 2.4 浮点寄存器

2.2.4 控制寄存器

(1) 状态寄存器 (SR)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	MD	RB	BL	—	—	—	—	—	—	—	—	—	—	—	—
初始值:	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FD	—	—	—	—	—	M	Q	IMASK			—	—	S	T	
初始值:	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
R/W:	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说明
31	—	0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。
30	MD	1	R/W	处理模式位 选择处理模式。 0: 用户模式 (指令中有不能执行的指令，资源中有不能存取的资源。) 1: 特权模式 在异常或者中断时，此位被置 1。
29	RB	1	R/W	特权模式的通用寄存器存储体的指定位 0: R0_BANK0 ~ R7_BANK0 能作为通用寄存器 R0 ~ R7 进行存取，R0_BANK1 ~ R7_BANK1 能通过 LDC/STC 指令进行存取。 1: R0_BANK1 ~ R7_BANK1 能作为通用寄存器 R0 ~ R7 进行存取，R0_BANK0 ~ R7_BANK0 能通过 LDC/STC 指令进行存取。 在异常或者中断时，此位被置 1。
28	BL	1	R/W	异常 / 中断块位 此位为 1 时，中断请求被屏蔽，当发生用户断点以外的一般异常时，处理器转移到复位状态。 在一般异常或者中断时，此位被置 1。
27 ~ 16	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。
15	FD	0	R/W	FPU 禁止位 此位为 1 时，FPU 指令产生一般 FPU 禁止异常，如果 FPU 指令在延迟槽，就产生槽 FPU 禁止异常（FPU 指令：H'F*** 指令、对于 FPUL/FPSCR 的 LDS(L)/STS(L) 指令）。
14 ~ 10	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。
9	M	0	R/W	M 位 DIV0S、DIV0U、DIV1 指令使用此位。
8	Q	0	R/W	Q 位 DIV0S、DIV0U、DIV1 指令使用此位。
7 ~ 4	IMASK	全 1	R/W	中断屏蔽级位 屏蔽优先级小于等于 IMASK 的中断。IMASK 在发生中断时是否变为中断接受级，能使用 CPU 运行模式寄存器（CPUOPM）进行切换。 有关 CPUOPM 的变化，请参照“附录 A. CPU 运行模式寄存器（CPUOPM）”。
3、2	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。
1	S	0	R/W	S 位 指定 MAC 指令的饱和运行。
0	T	0	R/W	T 位 表示真假条件、进位、借位、上溢或者下溢等。 详细内容请参照“第 3 章 指令系统”。

- (2) 保存状态寄存器 (SSR) (32 位, 特权保护, 初始值 = 不定值)

在发生异常或者中断时, 将 SR 的内容保存到 SSR。

- (3) 保存程序计数器 (SPC) (32 位, 特权保护, 初始值 = 不定值)

将发生异常或者中断的指令地址保存到 SPC。

- (4) 全局基址寄存器 (GBR) (32 位, 初始值 = 不定值)

GBR 作为 @(disp,GBR) 和 @(R0,GBR) 寻址方式的基址被参照。

- (5) 向量基址寄存器 (VBR) (32 位, 特权保护, 初始值 =H'0000 0000)

在发生异常或者中断时, VBR 作为转移目标的基址被参照。详细内容请参照“第 5 章 异常处理”。

- (6) 保存通用寄存器 15 (SGR) (32 位, 特权保护, 初始值 = 不定值)

在发生异常或者中断时, 将 R15 的内容保存到 SGR。

- (7) 调试基址寄存器 (DBR) (32 位, 特权保护, 初始值 = 不定值)

当用户断点调试功能有效时 (CBCR.UBDE=1), DBR 代替 VBR, 作为指向用户断点处理程序的转移目标地址被参照。

2.2.5 系统寄存器

- (1) 乘加高位寄存器 (MACH) (32 位, 初始值 = 不定值)、
乘加低位寄存器 (MACL) (32 位, 初始值 = 不定值)

MACH/MACL 用作 MAC 指令的运算值。也用于保存 MAC 指令和 MUL 指令的运算结果。

- (2) 过程寄存器 (PR) (32 位, 初始值 = 不定值)

将使用 BSR、BSRF 和 JSR 指令的子程序调用的返回地址保存到 PR。由子程序的返回指令 (RTS) 参照 PR。

- (3) 程序计数器 (PC) (32 位, 初始值 =H'A000 0000)

PC 表示执行中的指令地址。

- (4) 浮点状态 / 控制寄存器 (FPSCR)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	FR	SZ	PR	DN	Cause		
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	Cause				Enable (EN)						Flag				RM	
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 22	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。
21	FR	0	R/W	浮点寄存器存储体 0: FPR0_BANK0 ~ FPR15_BANK0 分配到 FR0 ~ FR15, FPR0_BANK1 ~ FPR15_BANK1 分配到 XF0 ~ XF15。 1: FPR0_BANK0 ~ FPR15_BANK0 分配到 XF0 ~ XF15, FPR0_BANK1 ~ FPR15 ~ BANK1 分配到 FR0 ~ FR15。
20	SZ	0	R/W	传送长度模式位 0: FMOV 指令的数据长度为 32 位。 1: FMOV 指令的数据长度为 32 位对或者 64 位。 SZ 位和 PR 位与字节排序的关系请参照图 2.5。
19	PR	0	R/W	精度模式位 0: 作为单精度运算执行浮点指令。 1: 作为双精度运算执行浮点指令（未定义图形支持指令）。 PR 位和 SZ 位与字节排序的关系请参照图 2.5。
18	ND	1	R/W	非规格化模式位 0: 将非规格化数作为非规格化数进行处理。 1: 将非规格化数作为 0 处理。
17 ~ 12	Cause	全 0	R/W	FPU 异常源字段 FPU 异常允许字段 FPU 异常标志字段 当执行 FPU 运算指令时，最初将 FPU 异常源字段置 0。然后，如果发生 FPU 异常，就将 FPU 异常源字段和 FPU 异常标志字段的相应位置 1。 最后，清除 FPU 异常标志字段。此后，FPU 异常标志字段保持发生异常的状态。 有关各字段的位分配请参照表 2.2。
11 ~ 7	Enable (EN)	全 0	R/W	
6 ~ 2	Flag	全 0	R/W	
1、0	RM	01	R/W	舍入模式位 选择舍入方法。 00: 接近方向的舍入 01: 零方向的舍入 10: 保留（禁止设定） 11: 保留（禁止设定）

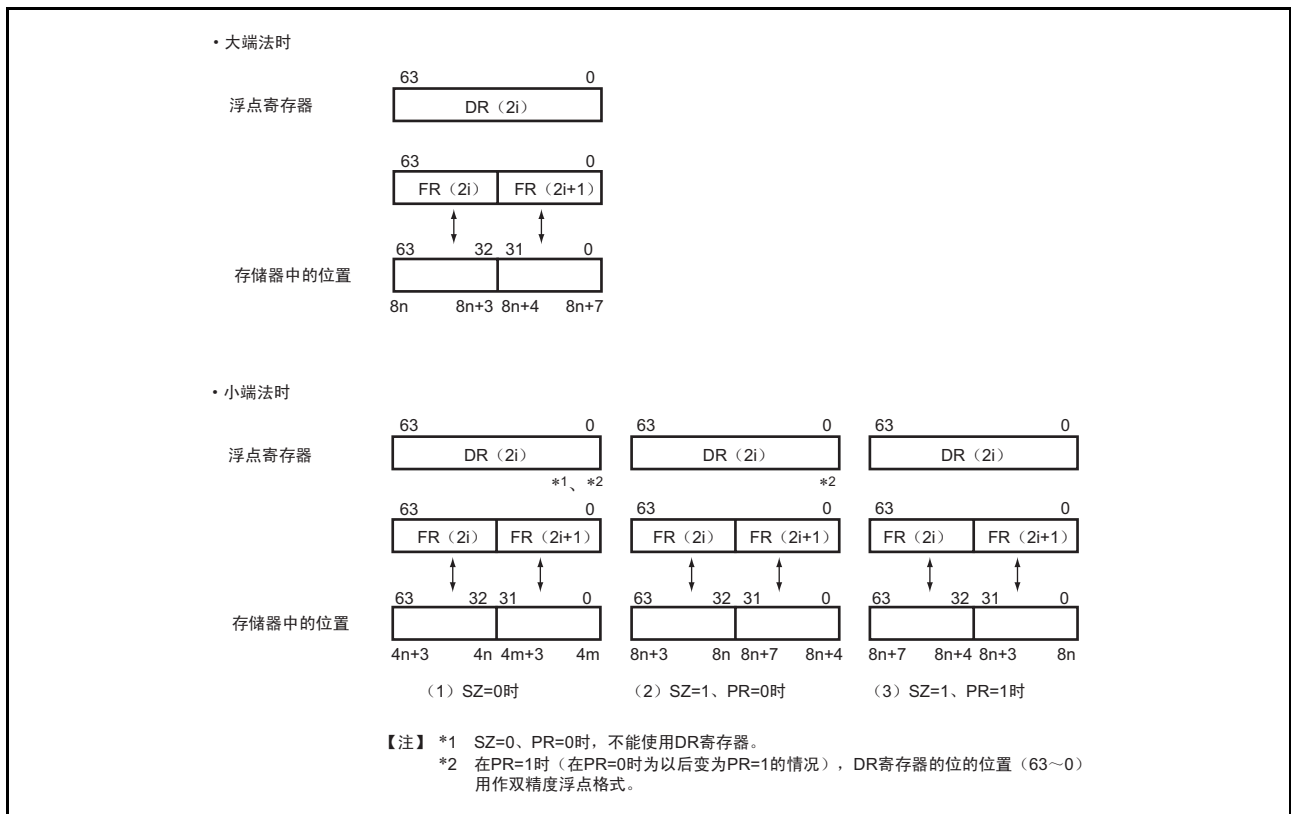


图 2.5 SZ 位和字节排序的关系

表 2.2 FPU 异常处理的位分配

	FPU 错误 (E)	无效运算 (V)	被零除 (Z)	上溢 (O)	下溢 (U)	不精确 (I)	
Cause	FPU 异常源字段	bit17	bit16	bit15	bit14	bit13	bit12
Enable	FPU 异常允许字段	无	bit11	bit10	bit9	bit8	bit7
Flag	FPU 异常标志字段	无	bit6	bit5	bit4	bit3	bit2

(5) 浮点通信寄存器 (FPUL) (32 位, 初始值 = 不定值)

通过 FPUL 进行 FPU 寄存器和 CPU 寄存器之间的数据传送。

2.3 存储器映像寄存器

有些控制寄存器被映像到以下的存储区。分配到这些存储区的寄存器有 2 个地址。

H'1C00 0000 ~ H'1FFF FFFF

H'FC00 0000 ~ H'FFFF FFFF

以上 2 个区域的使用如下：

- H'1C00 0000 ~ H'1FFF FFFF

此区域必须使用 MMU 的地址转换功能进行存取。通过将此区域的页号设定到 TLB 的相应段，能存取存储器映像寄存器。如果对此区域不使用 MMU 的地址转换功能而进行存取，就不能保证正常的运行。

- H'FC00 0000 ~ H'FFFF FFFF

在用户模式中存取区域 H'FC00 0000 ~ H'FFFF FFFF 时，会产生地址错误。在用户模式中，通过地址转换进行的存取，能参照存储器映像寄存器。

【注】 不能存取 2 个区域中未分配寄存器的地址。如果存取未分配寄存器的地址，运行就不确定。另外，存储器映像分配寄存器必须以固定的数据长度进行存取，否则运行也不确定。

2.4 寄存器的数据格式

寄存器操作数的数据长度总是为长字（32 位）。在将存储器中的数据加载到寄存器时，如果存储器操作数的数据长度为字节（8 位）或者字（16 位），就将其符号扩展为长字，然后保存到寄存器。

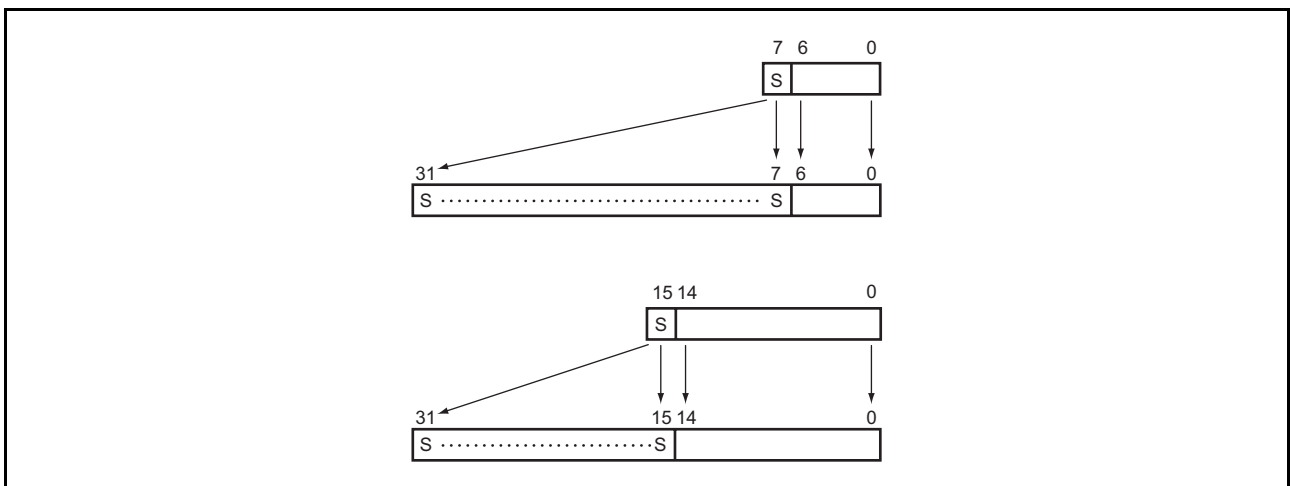


图 2.6 字节数据、字数据寄存器的数据格式

2.5 存储器的数据格式

存储器的数据格式有字节、字、长字。能以 8 位的字节、16 位的字或者 32 位的长字的格式存取存储器。将未满 32 位的存储器操作数进行符号扩展后保存到寄存器。

必须从字边界（每 2 个字节的偶数地址：地址 $2n$ ）存取字操作数，从长字边界（每 4 个字节的偶数地址：地址 $4n$ ）存取长字操作数，否则将发生地址错误。但是，能从任意地址存取字节操作数。

数据格式可选择大端法或者小端法的字节排序。必须在上电复位时通过外部引脚设定字节排序。不能动态更改字节排序。但是，位的位置总是按最高位（most-significant）到最低位（least-significant）的顺序从左到右递减编号。即对于 32 位的长字，最左边的位是 bit31，为最高位；最右边的位是 bit0，为最低位。存储器的数据格式如图 2.7 所示。

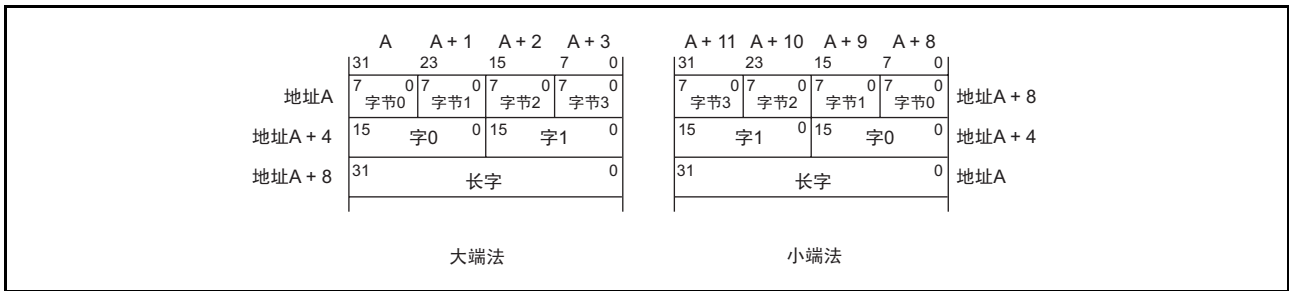


图 2.7 存储器的数据格式

有关 64 位的数据格式请参照图 2.5。

2.6 处理状态

处理状态大致分为复位状态、指令执行状态和低功耗状态 3 种。

(1) 复位状态

CPU 处于复位状态。

在上电复位状态下，对 CPU 内部状态和内部外围模块寄存器进行初始化。详细内容请参照各章的寄存器结构。

(2) 指令执行状态

这是 CPU 依次执行程序的状态。指令执行状态有一般的程序执行状态和异常处理状态。

(3) 低功耗状态

这是 CPU 停止运行、降低功耗的状态。通过睡眠指令进入低功率状态，有睡眠模式和待机模式 2 种。有关低功耗状态的详细内容，请参照“第 28 章 低功耗模式”。

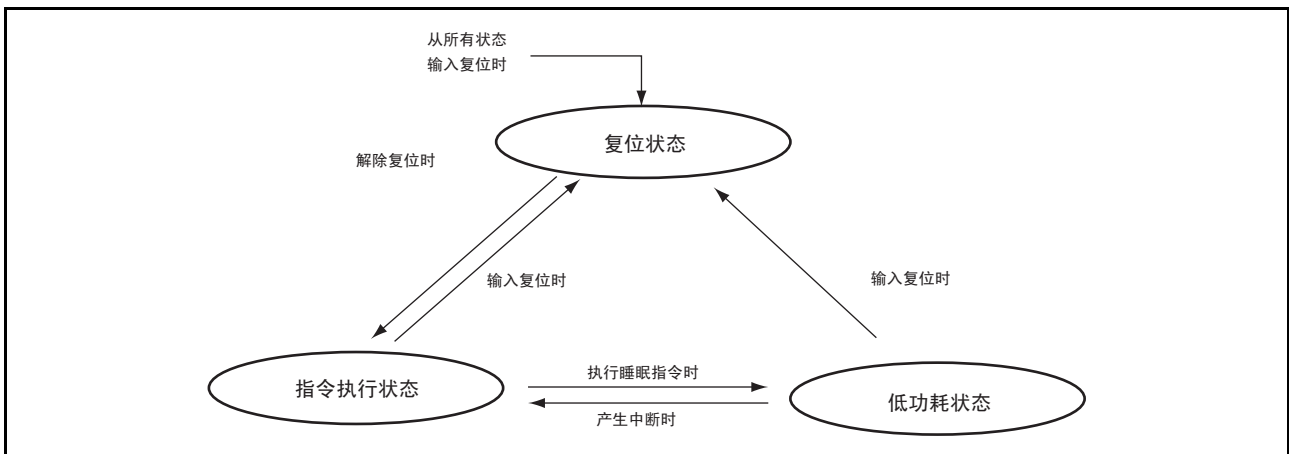


图 2.8 处理状态转移图

2.7 使用时的注意事项

2.7.1 自改写代码的注意事项

为了进行高速处理，和以前的 SH-4 相比，SH-4A 大幅度强化了预取指令的功能。为此，如果要在改写存储器中的指令后立即执行该指令，就非常有可能执行已预取的更新前的指令。为了确实地执行更新后的指令，必须在要改写的指令和已改写的指令之间执行下列指令：

(1) 改写的指令在不能缓存的区域时

```
SYNCO  
ICBI @Rn
```

通过 ICBI 指令的 Rn 指定的地址，可以是不发生地址错误的范围中的任意地址。

(2) 改写的指令在能缓存的区域时（直写模式时）

```
SYNCO  
ICBI @Rn
```

必须通过 ICBI 指令使对应改写指令的指令高速缓存的全部区域无效。ICBI 以块为单位，1 块为 32 字节。

(3) 改写指令在能缓存的区域时（回拷模式时）

```
OCBP @Rm 或者 OCBWB @Rm  
SYNCO  
ICBI @Rn
```

必须通过 OCBP 指令或者 OCBWB 指令将对应改写指令的操作数高速缓存的全部区域回写到主存储器，之后通过 ICBI 指令使对应的指令高速缓存区域无效。ICBI、OCBP 和 OCBWB 以块为单位，1 块为 32 字节。

【注】 自改写代码：在动态地改写存储器中的指令的同时执行的指令。

第 3 章 指令系统

SH-4A 的指令系统通过固定长度的 16 位指令来实现。SH-4A 以字节（8 位）、字（16 位）、长字（32 位）、4 个字（64 位）的数据长度对存储器进行存取。单精度浮点数据（32 位）能用长字或者 4 个字，而双精度浮点数据（64 位）能用 4 个字与存储器进行数据存取。SH-4A 在将字节和字的数据从存储器传送到寄存器时，将数据进行符号扩展。

3.1 执行环境

(1) PC

PC 表示此指令本身的指令地址。

(2) 加载 / 存储体系结构

SH-4A 的特点是通过寄存器执行基本运算的加载 / 存储体系结构。除了能在存储器直接执行的如逻辑 AND 的位操作运算以外，存取存储器的运算需要在将数据加载到寄存器后，通过寄存器执行。

(3) 延迟转移

除 BF 和 BT 这 2 条转移指令以外，SH-4A 的转移指令和 RTE 为延迟转移指令。在延迟转移时，在转移目标指令前执行转移指令的下一条指令。

(4) 延迟槽

延迟转移后的执行槽称为“延迟槽”。例如，BRA 的执行顺序如下：

表 3.1 延迟转移指令的执行顺序

指令队列			执行顺序
BRA	TARGET	(延迟转移指令)	BRA
ADD		(延迟槽)	↓
:			ADD
:			↓
TARGET	target-inst	(转移目标指令)	target-inst

有些指令在延迟槽执行时会产生槽非法指令异常。详细内容请参照“第 5 章 异常处理”。转移未成立的 BF/S 和 BT/S 的下一条指令也是延迟槽指令。

(5) T 位

状态寄存器（SR）的 T 位用于表示比较运算的结果等，通过条件转移指令来参照。条件转移指令的例子如下所示：

```
ADD    #1, R0    ; ADD 运算不变更 T 位。
CMP/EQ R1, R0    ; 当 R0=R1 时，T 位置 1。
BT     TARGET    ; 当 T 位 =1 (R0=R1) 时，转移到 TARGET。
```

在 RTE 的延迟槽中，进行以下的状态寄存器（SR）位的参照：

指令存取是使用变更前的 MD 位，而数据存取是存取变更后的 MD 位。变更后的 S、T、M、Q、FD、BL 和 RB 位用于延迟槽指令的执行。STC 和 STC.L SR 指令是存取变更后的全部 SR 位。

(6) 常数

8 位的常数能通过指令码和立即数指定，而 16 位和 32 位的常数能通过存储器定义，能通过 PC 相对加载指令进行参照。

MOV.W @(disp, PC), Rn


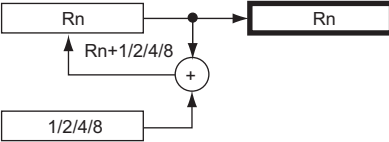
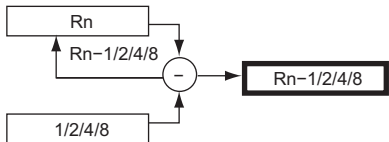
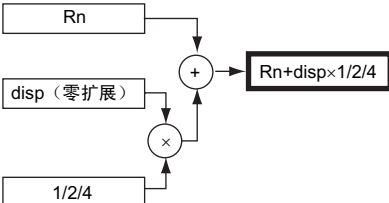
MOV.L @(disp, PC), Rn

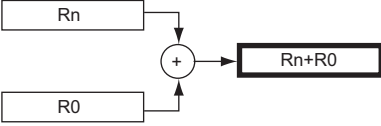
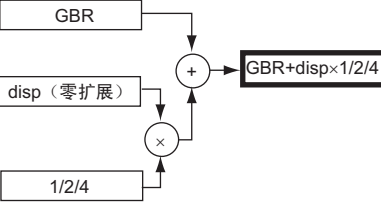
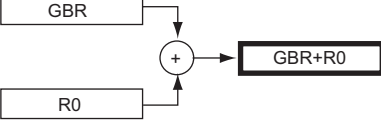
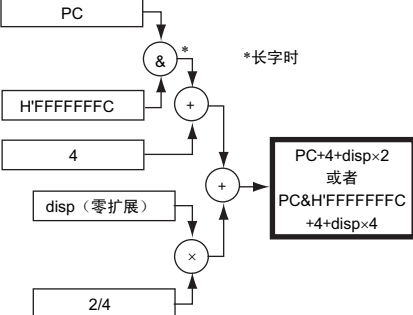
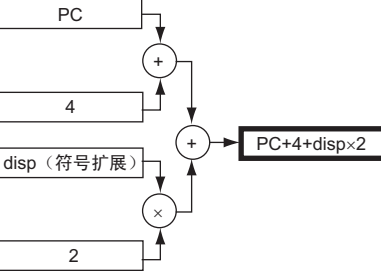
对于浮点，没有 PC 相对加载指令。但是，能通过使用 FLDI0 和 FLDI1 指令给单精度浮点寄存器设定 0.0 或者 1.0。

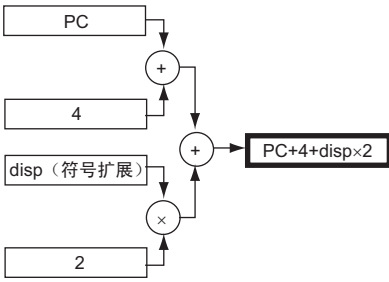
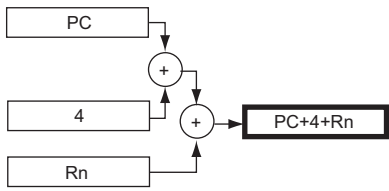
3.2 寻址方式

寻址方式和有效地址的计算如表 3.2 所示。当存取虚拟地址空间的某位置时（MMUCR.AT=1），有效地址就被转换为物理地址。如果选择多个虚拟存储空间系统（MMUCR.SV=0），PTEH 的最低位也作为存取的 ASID 被参照。请参照“第 7 章 存储器管理单元（MMU）”。

表 3.2 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。 	$Rn \rightarrow EA$ （EA: 有效地址）
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，为 4 个字时是 8。 	$Rn \rightarrow EA$ 在指令执行后 字节: $Rn+1 \rightarrow Rn$ 字: $Rn+2 \rightarrow Rn$ 长字: $Rn+4 \rightarrow Rn$ 4 个字: $Rn+8 \rightarrow Rn$
先减寄存器间接	@-Rn	有效地址为先减去常数后的寄存器 Rn 的内容。常数为字节时是 1，为字时是 2，为长字时是 4，为 4 个字时是 8。 	字节: $Rn-1 \rightarrow Rn$ 字: $Rn-2 \rightarrow Rn$ 长字: $Rn-4 \rightarrow Rn$ 4 个字: $Rn-8 \rightarrow Rn$ $Rn \rightarrow EA$ （通过计算后的 Rn 执行指令）
带位移量的寄存器间接	@(disp:4, Rn)	有效地址为给寄存器 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: $Rn+disp \rightarrow EA$ 字: $Rn+disp \times 2 \rightarrow EA$ 长字: $Rn+disp \times 4 \rightarrow EA$

寻址方式	指令格式	有效地址的计算方法	计算式
带变址的寄存器间接	@(R0, Rn)	有效地址为给寄存器 Rn 加上 R0 后的内容。 	$Rn+R0 \rightarrow EA$
带位移量的 GBR 间接	@(disp:8, GBR)	有效地址为给寄存器 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: $GBR+disp \rightarrow EA$ 字: $GBR+disp \times 2 \rightarrow EA$ 长字: $GBR+disp \times 4 \rightarrow EA$
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。 	$GBR+R0 \rightarrow EA$
带位移量的 PC 相对	@(disp:8, PC)	有效地址为 PC+4 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，为长字时屏蔽 PC 的低 2 位。 	字: $PC+4+disp \times 2 \rightarrow EA$ 长字: PC & $H'FFFFFFFC+4+disp \times 4 \rightarrow EA$
PC 相对	disp:8	有效地址为将 8 位位移量 disp 进行符号扩展后乘 2，再加上 PC+4 的内容。 	$PC+4+disp \times 2 \rightarrow Branch-Target$

寻址方式	指令格式	有效地址的计算方法	计算式
PC 相对	disp:12	有效地址为将 12 位位移量 disp 进行符号扩展后乘以 2，再加上 PC+4 的内容。 	$PC+4+disp \times 2 \rightarrow \text{Branch-Target}$
	Rn	有效地址为 PC+4 加上 Rn 后的内容。 	$PC+4+Rn \rightarrow \text{Branch-Target}$
立即	#imm:8	将 TST、AND、OR、XOR 指令的 8 位立即 imm 进行零扩展。	—
	#imm:8	将 MOV、ADD、CMP/EQ 指令的 8 位立即 imm 进行符号扩展。	—
	#imm:8	将 TRAPA 指令的 8 位立即 imm 进行零扩展后乘 4。	—

【注】 在下列带位移量（disp）的寻址方式中，本手册的汇编程序的记述内容是根据操作数长度进行倍增（×1、×2、×4）前的值。这是为了明确 LSI 的运行，实际汇编程序的记述内容，请参照各汇编程序的记载规则。

- @(disp:4, Rn) ;带位移量的寄存器间接
- @(disp:8, GBR) ;带位移量的 GBR 间接
- @(disp:8, PC) ;带位移量的 PC 相对
- disp :8, disp :12 ;PC 相对

3.3 指令系统

在表 3.4 ~ 表 3.13 所示的 SuperH 指令说明中使用的记法如表 3.3 所示。

表 3.3 指令清单的记法

项目	格式	说明
指令助记符	OP.Sz SRC,DEST	OP: 操作码 Sz: 长度 SRC: 源操作数 DEST: 源 / 目标操作数 Rm: 源寄存器 Rn: 目标寄存器 imm: 立即数 disp: 位移量
运算要点		→、←: 传送方向 (xx): 存储器操作数 M/Q/T: SR 的标志位 &: 各位的逻辑与 : 各位的逻辑或 ^: 各位的异或 ~: 各位的逻辑非 <<n,>>n: n 位移位
指令码	MSB←→LSB	m m m m: 寄存器号 (Rm, FRm) n n n n: 寄存器号 (Rn, FRn) 0000: R0, FR0 0001: R1, FR1 : 1111: R15, FR15 m m m: 寄存器号 (DRm, XDm, Rm_BANK) n n n: 寄存器号 (DRn, XDn, Rn_BANK) 000: DR0, XD0, R0_BANK 001: DR2, XD2, R1_BANK : 111: DR14, XD14, R7_BANK m m: 寄存器号 (FVm) n n: 寄存器号 (FVn)
指令码	MSB←→LSB	00: FV0 01: FV4 10: FV8 11: FV12 iiii: 立即数 dddd: 位移量
特权模式	—	当记载为“特权”时, 只能在特权模式中执行。
T 位	指令执行后的 T 位值	—: 不变
新规	—	当记载为“新规”时, 处理器版本寄存器 (PVR) 的 VER 位的值为 H'20 的 SH-4A, 是新追加的指令。

【注】 根据指令操作数的长度进行倍增 (×1、×2、×4、×8)。

表 3.4 定点传送指令

指令	操作	指令码	特权	T 位	新规
MOV #imm,Rn	imm→符号扩展→Rn	1110nmmniiiiiii	—	—	—
MOV.W @(disp*,PC),Rn	(disp×2+PC+4)→符号扩展→Rn	1001nnnnnddddddd	—	—	—
MOV.L @(disp*,PC),Rn	(disp×4+PC&H'FFFFFFFC+4)→Rn	1101nnnnnddddddd	—	—	—
MOV Rm,Rn	Rm→Rn	0110nnnnmmmm0011	—	—	—
MOV.B Rm,@Rn	Rm→(Rn)	0010nnnnmmmm0000	—	—	—
MOV.W Rm,@Rn	Rm→(Rn)	0010nnnnmmmm0001	—	—	—
MOV.L Rm,@Rn	Rm→(Rn)	0010nnnnmmmm0010	—	—	—
MOV.B @Rm,Rn	(Rm)→符号扩展→Rn	0110nnnnmmmm0000	—	—	—
MOV.W @Rm,Rn	(Rm)→符号扩展→Rn	0110nnnnmmmm0001	—	—	—
MOV.L @Rm,Rn	(Rm)→Rn	0110nnnnmmmm0010	—	—	—
MOV.B Rm,@-Rn	Rn-1→Rn, Rm→(Rn)	0010nnnnmmmm0100	—	—	—
MOV.W Rm,@-Rn	Rn-2→Rn, Rm→(Rn)	0010nnnnmmmm0101	—	—	—
MOV.L Rm,@-Rn	Rn-4→Rn, Rm→(Rn)	0010nnnnmmmm0110	—	—	—
MOV.B @Rm+,Rn	(Rm)→符号扩展→Rn, Rm+1→Rm	0110nnnnmmmm0100	—	—	—
MOV.W @Rm+,Rn	(Rm)→符号扩展→Rn, Rm+2→Rm	0110nnnnmmmm0101	—	—	—
MOV.L @Rm+,Rn	(Rm)→Rn, Rm+4→Rm	0110nnnnmmmm0110	—	—	—
MOV.B R0,@(disp*,Rn)	R0→(disp+Rn)	10000000nnnnndddd	—	—	—
MOV.W R0,@(disp*,Rn)	R0→(disp×2+Rn)	10000001nnnnndddd	—	—	—
MOV.L Rm,@(disp*,Rn)	Rm→(disp×4+Rn)	0001nnnnmmmmndddd	—	—	—
MOV.B @(disp*,Rm),R0	(disp+Rm)→符号扩展→R0	10000100mmmmndddd	—	—	—
MOV.W @(disp*,Rm),R0	(disp×2+Rm)→符号扩展→R0	10000101mmmmndddd	—	—	—
MOV.L @(disp*,Rm),Rn	(disp×4+Rm)→Rn	0101nnnnmmmmndddd	—	—	—
MOV.B Rm,@(R0,Rn)	Rm→(R0+Rn)	0000nnnnmmmm0100	—	—	—
MOV.W Rm,@(R0,Rn)	Rm→(R0+Rn)	0000nnnnmmmm0101	—	—	—
MOV.L Rm,@(R0,Rn)	Rm→(R0+Rn)	0000nnnnmmmm0110	—	—	—
MOV.B @(R0,Rm),Rn	(R0+Rm)→符号扩展→Rn	0000nnnnmmmm1100	—	—	—
MOV.W @(R0,Rm),Rn	(R0+Rm)→符号扩展→Rn	0000nnnnmmmm1101	—	—	—
MOV.L @(R0,Rm),Rn	(R0+Rm)→Rn	0000nnnnmmmm1110	—	—	—
MOV.B R0,@(disp*,GBR)	R0→(disp+GBR)	11000000ddddddd	—	—	—
MOV.W R0,@(disp*,GBR)	R0→(disp×2+GBR)	11000001ddddddd	—	—	—
MOV.L R0,@(disp*,GBR)	R0→(disp×4+GBR)	11000010ddddddd	—	—	—
MOV.B @(disp*,GBR),R0	(disp+GBR)→符号扩展→R0	11000100ddddddd	—	—	—
MOV.W @(disp*,GBR),R0	(disp×2+GBR)→符号扩展→R0	11000101ddddddd	—	—	—
MOV.L @(disp*,GBR),R0	(disp×4+GBR)→R0	11000110ddddddd	—	—	—
MOVA @(disp*,PC),R0	disp×4+PC&H'FFFFFFFC+4→R0	11000111ddddddd	—	—	—
MOVCO.L R0,@Rn	LDST→T if(T==1)R0→(Rn), 0→LDST	0000nnnn01110011	—	LDST	新规
MOVLI.L @Rm,R0	1→LDST (Rm)→R0 当发生中断/异常时, 0→LDST	0000mmmm01100011	—	—	新规
MOVUA.L @Rm,R0	(Rm)→R0 非边界调整数据的加载	0100mmmm10101001	—	—	新规
MOVUA.L @Rm+,R0	(Rm)→R0, Rm+4→Rm 非边界调整数据的加载	0100mmmm11101001	—	—	新规
MOVT Rn	T→Rn	0000nnnn00101001	—	—	—
SWAP.B Rm,Rn	Rm→低位 2 字节的高 低字节交换→Rn	0110nnnnmmmm1000	—	—	—
SWAP.W Rm,Rn	Rm→高低字交换→Rn	0110nnnnmmmm1001	—	—	—
XTRCT Rm,Rn	Rm:Rn 中间 32 位→Rn	0010nnnnmmmm1101	—	—	—

【注】* 在瑞萨的汇编程序中，给 disp 设定倍增后（×1、×2、×4）的值。

表 3.5 算术运算指令

指令	操作	指令码	特权	T 位	新规
ADD Rm,Rn	$Rn+Rm \rightarrow Rn$	0011nnnnmmmm1100	—	—	—
ADD #imm,Rn	$Rn+imm \rightarrow Rn$	0111nnnniiiiiii	—	—	—
ADDC Rm,Rn	$Rn+Rm+T \rightarrow Rn$, 进位 $\rightarrow T$	0011nnnnmmmm1110	—	进位	—
ADDV Rm,Rn	$Rn+Rm \rightarrow Rn$, 上溢 $\rightarrow T$	0011nnnnmmmm1111	—	上溢	—
CMP/EQ #imm,R0	当 $R0=imm$ 时, $1 \rightarrow T$ 否则 $0 \rightarrow T$	10001000iiiiiii	—	比较 结果	—
CMP/EQ Rm,Rn	当 $Rn=Rm$ 时, $1 \rightarrow T$ 否则 $0 \rightarrow T$	0011nnnnmmmm0000	—	比较 结果	—
CMP/HS Rm,Rn	当无符号 $Rn \geq Rm$ 时, $1 \rightarrow T$ 否则 $0 \rightarrow T$	0011nnnnmmmm0010	—	比较 结果	—
CMP/GE Rm,Rn	当带符号 $Rn \geq Rm$ 时, $1 \rightarrow T$ 否则 $0 \rightarrow T$	0011nnnnmmmm0011	—	比较 结果	—
CMP/HI Rm,Rn	当无符号 $Rn > Rm$ 时, $1 \rightarrow T$ 否则 $0 \rightarrow T$	0011nnnnmmmm0110	—	比较 结果	—
CMP/GT Rm,Rn	当带符号 $Rn > Rm$ 时, $1 \rightarrow T$ 否则 $0 \rightarrow T$	0011nnnnmmmm0111	—	比较 结果	—
CMP/PZ Rn	当 $Rn \geq 0$ 时, $1 \rightarrow T$ 否则 $0 \rightarrow T$	0100nnnn00010001	—	比较 结果	—
CMP/PL Rn	当 $Rn > 0$ 时, $1 \rightarrow T$ 否则 $0 \rightarrow T$	0100nnnn00010101	—	比较 结果	—
CMP/STR Rm,Rn	当任意 1 个字节相等时, $1 \rightarrow T$ 否则 $0 \rightarrow T$	0010nnnnmmmm1100	—	比较 结果	—
DIV1 Rm,Rn	单步除法 ($Rn \div Rm$)	0011nnnnmmmm0100	—	计算 结果	—
DIV0S Rm,Rn	Rn 的 MSB $\rightarrow Q$, Rm 的 MSB $\rightarrow M$, $M^Q \rightarrow T$	0010nnnnmmmm0111	—	计算 结果	—
DIV0U	$0 \rightarrow M/Q/T$	000000000011001	—	0	—
DMULS.L Rm,Rn	带符号 $Rn \times Rm \rightarrow MAC$ $32 \times 32 \rightarrow 64$ 位	0011nnnnmmmm1101	—	—	—
DMULU.L Rm,Rn	无符号 $Rn \times Rm \rightarrow MAC$ $32 \times 32 \rightarrow 64$ 位	0011nnnnmmmm0101	—	—	—
DT Rn	$Rn-1 \rightarrow Rn$ 当 Rn 为 0 时, $1 \rightarrow T$ 否则 $0 \rightarrow T$	0100nnnn00010000	—	比较 结果	—
EXTS.B Rm,Rn	将 Rm 进行字节的符号扩展 $\rightarrow Rn$	0110nnnnmmmm1110	—	—	—
EXTS.W Rm,Rn	将 Rm 进行字的符号扩展 $\rightarrow Rn$	0110nnnnmmmm1111	—	—	—
EXTU.B Rm,Rn	将 Rm 进行字节的零扩展 $\rightarrow Rn$	0110nnnnmmmm1100	—	—	—
EXTU.W Rm,Rn	将 Rm 进行字的零扩展 $\rightarrow Rn$	0110nnnnmmmm1101	—	—	—
MAC.L @Rm+,@Rn+	带符号 $(Rn) \times (Rm) + MAC \rightarrow MAC$ $Rn+4 \rightarrow Rn$, $Rm+4 \rightarrow Rm$ $32 \times 32 + 64 \rightarrow 64$ 位	0000nnnnmmmm1111	—	—	—
MAC.W @Rm+,@Rn+	带符号 $(Rn) \times (Rm) + MAC \rightarrow MAC$ $Rn+2 \rightarrow Rn$, $Rm+2 \rightarrow Rm$ $16 \times 16 + 64 \rightarrow 64$ 位	0100nnnnmmmm1111	—	—	—
MUL.L Rm,Rn	$Rn \times Rm \rightarrow MACL$ $32 \times 32 \rightarrow 32$ 位	0000nnnnmmmm0111	—	—	—

指令	操作	指令码	特权	T 位	新规
MULS.W Rm,Rn	带符号 $Rn \times Rm \rightarrow MACL$ 16×16→32 位	0010nnnnmmmm1111	—	—	—
MULU.W Rm,Rn	无符号 $Rn \times Rm \rightarrow MACL$ 16×16→32 位	0010nnnnmmmm1110	—	—	—
NEG Rm,Rn	$0-Rm \rightarrow Rn$	0110nnnnmmmm1011	—	—	—
NEGC Rm,Rn	$0-Rm-T \rightarrow Rn$, 借位 →T	0110nnnnmmmm1010	—	借位	—
SUB Rm,Rn	$Rn-Rm \rightarrow Rn$	0011nnnnmmmm1000	—	—	—
SUBC Rm,Rn	$Rn-Rm-T \rightarrow Rn$, 借位 →T	0011nnnnmmmm1010	—	借位	—
SUBV Rm,Rn	$Rn-Rm \rightarrow Rn$, 下溢 →T	0011nnnnmmmm1011	—	下溢	—

表 3.6 逻辑运算指令

指令	操作	指令码	特权	T 位	新规
AND Rm,Rn	$Rn \& Rm \rightarrow Rn$	0010nnnnmmmm1001	—	—	—
AND #imm,R0	$R0 \& imm \rightarrow R0$	11001001iiiiiiii	—	—	—
AND.B #imm,@(R0,GBR)	$(R0+GBR) \& imm \rightarrow (R0+GBR)$	11001101iiiiiiii	—	—	—
NOT Rm,Rn	$\sim Rm \rightarrow Rn$	0110nnnnmmmm0111	—	—	—
OR Rm,Rn	$Rn Rm \rightarrow Rn$	0010nnnnmmmm1011	—	—	—
OR #imm,R0	$R0 imm \rightarrow R0$	11001011iiiiiiii	—	—	—
OR.B #imm,@(R0,GBR)	$(R0+GBR) imm \rightarrow (R0+GBR)$	11001111iiiiiiii	—	—	—
TAS.B @Rn	当 (Rn) 为 0 时, $1 \rightarrow T$ 否则 $0 \rightarrow T1 \rightarrow (Rn)$ 的 MSB	0100nnnn00011011	—	测试结果	—
TST Rm,Rn	$Rn \& Rm$ 当结果为 0 时, $1 \rightarrow T$ 否则 $0 \rightarrow T$	0010nnnnmmmm1000	—	测试结果	—
TST #imm,R0	$R0 \& imm$ 当结果为 0 时, $1 \rightarrow T$ 否则 $0 \rightarrow T$	11001000iiiiiiii	—	测试结果	—
TST.B #imm,@(R0,GBR)	$(R0+GBR) \& imm$ 当结果为 0 时, $1 \rightarrow T$ 否则 $0 \rightarrow T$	11001100iiiiiiii	—	测试结果	—
XOR Rm,Rn	$Rn \wedge Rm \rightarrow Rn$	0010nnnnmmmm1010	—	—	—
XOR #imm,R0	$R0 \wedge imm \rightarrow R0$	11001010iiiiiiii	—	—	—
XOR.B #imm,@(R0,GBR)	$(R0+GBR) \wedge imm \rightarrow (R0+GBR)$	11001110iiiiiiii	—	—	—

表 3.7 移位指令

指令	操作	指令码	特权	T 位	新规
ROTL Rn	$T \leftarrow Rn \leftarrow \text{MSB}$	0100nnnn00000100	—	MSB	—
ROTR Rn	$\text{LSB} \rightarrow Rn \rightarrow T$	0100nnnn00000101	—	LSB	—
ROTCL Rn	$T \leftarrow Rn \leftarrow T$	0100nnnn00100100	—	MSB	—
ROTCR Rn	$T \rightarrow Rn \rightarrow T$	0100nnnn00100101	—	LSB	—
SHAD Rm, Rn	当 $Rm \geq 0$ 时, $Rn \ll Rm \rightarrow Rn$ 当 $Rm < 0$ 时, $Rn \gg Rm \rightarrow [MSB \rightarrow Rn]$	0100nnnnmmmm1100	—	—	—
SHAL Rn	$T \leftarrow Rn \leftarrow 0$	0100nnnn00100000	—	MSB	—
SHAR Rn	$\text{MSB} \rightarrow Rn \rightarrow T$	0100nnnn00100001	—	LSB	—
SHLD Rm, Rn	当 $Rm \geq 0$ 时, $Rn \ll Rm \rightarrow Rn$ 当 $Rm < 0$ 时, $Rn \gg Rm \rightarrow [0 \rightarrow Rn]$	0100nnnnmmmm1101	—	—	—
SHLL Rn	$T \leftarrow Rn \leftarrow 0$	0100nnnn00000000	—	MSB	—
SHLR Rn	$0 \rightarrow Rn \rightarrow T$	0100nnnn00000001	—	LSB	—
SHLL2 Rn	$Rn \ll 2 \rightarrow Rn$	0100nnnn00001000	—	—	—
SHLR2 Rn	$Rn \gg 2 \rightarrow Rn$	0100nnnn00001001	—	—	—
SHLL8 Rn	$Rn \ll 8 \rightarrow Rn$	0100nnnn00011000	—	—	—
SHLR8 Rn	$Rn \gg 8 \rightarrow Rn$	0100nnnn00011001	—	—	—
SHLL16 Rn	$Rn \ll 16 \rightarrow Rn$	0100nnnn00101000	—	—	—
SHLR16 Rn	$Rn \gg 16 \rightarrow Rn$	0100nnnn00101001	—	—	—

表 3.8 转移指令

指令	操作	指令码	特权	T 位	新规
BF label	当 $T=0$ 时, $\text{disp} \times 2 + PC + 4 \rightarrow PC$ 当 $T=1$ 时, nop	10001011ddddddd	—	—	—
BF/S label	延迟转移 当 $T=0$ 时, $\text{disp} \times 2 + PC + 4 \rightarrow PC$ 当 $T=1$ 时, nop	10001111ddddddd	—	—	—
BT label	当 $T=1$ 时, $\text{disp} \times 2 + PC + 4 \rightarrow PC$ 当 $T=0$ 时, nop	10001001ddddddd	—	—	—
BT/S label	延迟转移 当 $T=1$ 时, $\text{disp} \times 2 + PC + 4 \rightarrow PC$ 当 $T=0$ 时, nop	10001101ddddddd	—	—	—
BRA label	延迟转移 $\text{disp} \times 2 + PC + 4 \rightarrow PC$	1010ddddddddddd	—	—	—
BRAF Rn	延迟转移 $Rn + PC + 4 \rightarrow PC$	0000nnnn00100011	—	—	—
BSR label	延迟转移 $PC + 4 \rightarrow PR$ $\text{disp} \times 2 + PC + 4 \rightarrow PC$	1011ddddddddddd	—	—	—
BSRF Rn	延迟转移 $PC + 4 \rightarrow PR$ $Rn + PC + 4 \rightarrow PC$	0000nnnn00000011	—	—	—
JMP @Rn	延迟转移 $Rn \rightarrow PC$	0100nnnn00101011	—	—	—
JSR @Rn	延迟转移 $PC + 4 \rightarrow PR$, $Rn \rightarrow PC$	0100nnnn00001011	—	—	—
RTS	延迟转移 $PR \rightarrow PC$	0000000000001011	—	—	—

表 3.9 系统控制指令

指令	操作	指令码	特权	T 位	新规
CLRMAC	0→MACH 和 MACL	000000000101000	—	—	—
CLRS	0→S	000000001001000	—	—	—
CLRT	0→T	000000000001000	—	0	—
ICBI @Rn	使高速缓存无效	0000nnnn11100011	—	—	新规
LDC Rm,SR	Rm→SR	0100mmmm00001110	特权	LSB	—
LDC Rm,GBR	Rm→GBR	0100mmmm00011110	—	—	—
LDC Rm,VBR	Rm→VBR	0100mmmm00101110	特权	—	—
LDC Rm,SGR	Rm→SGR	0100mmmm00111010	特权	—	新规
LDC Rm,SSR	Rm→SSR	0100mmmm00111110	特权	—	—
LDC Rm,SPC	Rm→SPC	0100mmmm01001110	特权	—	—
LDC Rm,DBR	Rm→DBR	0100mmmm11111010	特权	—	—
LDC Rm,Rn_BANK	Rm→Rn_BANK (n=0~7)	0100mmmm1nnn1110	特权	—	—
LDC.L @Rm+,SR	(Rm)→SR, Rm+4→Rm	0100mmmm00000111	特权	LSB	—
LDC.L @Rm+,GBR	(Rm)→GBR, Rm+4→Rm	0100mmmm00010111	—	—	—
LDC.L @Rm+,VBR	(Rm)→VBR, Rm+4→Rm	0100mmmm00100111	特权	—	—
LDC.L @Rm+,SGR	(Rm)→SGR, Rm+4→Rm	0100mmmm00110110	特权	—	新规
LDC.L @Rm+,SSR	(Rm)→SSR, Rm+4→Rm	0100mmmm00110111	特权	—	—
LDC.L @Rm+,SPC	(Rm)→SPC, Rm+4→Rm	0100mmmm01000111	特权	—	—
LDC.L @Rm+,DBR	(Rm)→DBR, Rm+4→Rm	0100mmmm11110110	特权	—	—
LDC.L @Rm+,Rn_BANK	(Rm)→Rn_BANK, Rm+4→Rm	0100mmmm1nnn0111	特权	—	—
LDS Rm,MACH	Rm→MACH	0100mmmm00001010	—	—	—
LDS Rm,MACL	Rm→MACL	0100mmmm00011010	—	—	—
LDS Rm,PR	Rm→PR	0100mmmm00101010	—	—	—
LDS.L @Rm+,MACH	(Rm)→MACH, Rm+4→Rm	0100mmmm00000110	—	—	—
LDS.L @Rm+,MACL	(Rm)→MACL, Rm+4→Rm	0100mmmm00010110	—	—	—
LDS.L @Rm+,PR	(Rm)→PR, Rm+4→Rm	0100mmmm00100110	—	—	—
LDTLB	PTEH/PTEL(/PTEA)→TLB	000000000111000	特权	—	—
MOVCA.L R0,@Rn	(不取高速缓存块) R0→(Rn)	0000nnnn11000011	—	—	—
NOP	无操作	000000000001001	—	—	—
OCBI @Rn	操作数高速缓存块无效	0000nnnn10010011	—	—	—
OCBP @Rn	回写操作数高速缓存块, 使其无效	0000nnnn10100011	—	—	—
OCBWB @Rn	回写操作数高速缓存块	0000nnnn10110011	—	—	—
PREF @Rn	(Rn)→操作数高速缓存	0000nnnn10000011	—	—	—
PREFI @Rn	将 32 字节的指令块读到指令高速缓存	0000nnnn11010011	—	—	新规
RTE	延迟转移 SSR/SPC→SR/PC	000000000101011	特权	—	—
SETS	1→S	000000001011000	—	—	—
SETT	1→T	000000000011000	—	1	—
SLEEP	睡眠或者待机	000000000011011	特权	—	—
STC SR,Rn	SR→Rn	0000nnnn00000010	特权	—	—
STC GBR,Rn	GBR→Rn	0000nnnn00010010	—	—	—
STC VBR,Rn	VBR→Rn	0000nnnn00100010	特权	—	—
STC SSR,Rn	SSR→Rn	0000nnnn00110010	特权	—	—
STC SPC,Rn	SPC→Rn	0000nnnn01000010	特权	—	—
STC SGR,Rn	SGR→Rn	0000nnnn00111010	特权	—	—
STC DBR,Rn	DBR→Rn	0000nnnn11111010	特权	—	—
STC Rm_BANK,Rn	Rm_BANK→Rn (m=0~7)	0000nnnn1mmmm0010	特权	—	—
STC.L SR,@-Rn	Rn-4→Rn, SR→(Rn)	0100nnnn00000011	特权	—	—
STC.L GBR,@-Rn	Rn-4→Rn, GBR→(Rn)	0100nnnn00010011	—	—	—
STC.L VBR,@-Rn	Rn-4→Rn, VBR→(Rn)	0100nnnn00100011	特权	—	—

指令	操作	指令码	特权	T 位	新规
STC.L SSR,@-Rn	Rn-4→Rn, SSR→(Rn)	0100nnnn00110011	特权	—	—
STC.L SPC,@-Rn	Rn-4→Rn, SPC→(Rn)	0100nnnn01000011	特权	—	—
STC.L SGR,@-Rn	Rn-4→Rn, SGR→(Rn)	0100nnnn00110010	特权	—	—
STC.L DBR,@-Rn	Rn-4→Rn, DBR→(Rn)	0100nnnn11110010	特权	—	—
STC.L Rm_BANK,@-Rn	Rn-4→Rn, Rm_BANK→(Rn) (m=0~7)	0100nnnn1mmm0011	特权	—	—
STS MACH,Rn	MACH→Rn	0000nnnn00001010	—	—	—
STS MACL,Rn	MACL→Rn	0000nnnn00011010	—	—	—
STS PR,Rn	PR→Rn	0000nnnn00101010	—	—	—
STS.L MACH,@-Rn	Rn-4→Rn, MACH→(Rn)	0100nnnn00000010	—	—	—
STS.L MACL,@-Rn	Rn-4→Rn, MACL→(Rn)	0100nnnn00010010	—	—	—
STS.L PR,@-Rn	Rn-4→Rn, PR→(Rn)	0100nnnn00100010	—	—	—
SYNCO	在本指令前的数据存取结束之前, 本指令后的指令不开始数据存取	0000000010101011	—	—	新规
TRAPA #imm	imm<<2→TRA, PC+2→SPC SR→SSR, R15→SGR 1→SR.MD/BL/RB, H'160→EXPEVT VBR+H'0100→PC	11000011iiiiiiii	—	—	—

表 3.10 浮点单精度指令

指令	操作	指令码	特权	T 位	新规
FLDI0 FRn	H'00000000→FRn	1111nnnn10001101	—	—	—
FLDI1 FRn	H'3F800000→FRn	1111nnnn10011101	—	—	—
FMOV FRm,FRn	FRm→FRn	1111nnnnmmmm1100	—	—	—
FMOV.S @Rm,FRn	(Rm)→FRn	1111nnnnmmmm1000	—	—	—
FMOV.S @(R0,Rm),FRn	(R0+Rm)→FRn	1111nnnnmmmm0110	—	—	—
FMOV.S @Rm+,FRn	(Rm)→FRn, Rm+4→Rm	1111nnnnmmmm1001	—	—	—
FMOV.S FRm,@Rn	FRm→(Rn)	1111nnnnmmmm1010	—	—	—
FMOV.S FRm,@-Rn	Rn-4→Rn, FRm→(Rn)	1111nnnnmmmm1011	—	—	—
FMOV.S FRm,@(R0,Rn)	FRm→(R0+Rn)	1111nnnnmmmm0111	—	—	—
FMOV DRm,DRn	DRm→DRn	1111nnnn0mmmm01100	—	—	—
FMOV @Rm,DRn	(Rm)→DRn	1111nnnn0mmmm1000	—	—	—
FMOV @(R0,Rm),DRn	(R0+Rm)→DRn	1111nnnn0mmmm0110	—	—	—
FMOV @Rm+,DRn	(Rm)→DRn, Rm+8→Rm	1111nnnn0mmmm1001	—	—	—
FMOV DRm,@Rn	DRm→(Rn)	1111nnnnmmmm01010	—	—	—
FMOV DRm,@-Rn	Rn-8→Rn, DRm→(Rn)	1111nnnnmmmm01011	—	—	—
FMOV DRm,@(R0,Rn)	DRm→(R0+Rn)	1111nnnnmmmm00111	—	—	—
FLDS FRm,FPUL	FRm→FPUL	1111mmmm00011101	—	—	—
FSTS FPUL,FRn	FPUL→FRn	1111nnnn00001101	—	—	—
FABS FRn	FRn & H'7FFF FFFF→FRn	1111nnnn01011101	—	—	—
FADD FRm,FRn	FRn+FRm→FRn	1111nnnnmmmm0000	—	—	—
FCMP/EQ FRm,FRn	当 FRn=FRm 时, 1→T 否则 0→T	1111nnnnmmmm0100	—	比较结果	—
FCMP/GT FRm,FRn	当 FRn > FRm 时, 1→T 否则 0→T	1111nnnnmmmm0101	—	比较结果	—
FDIV FRm,FRn	FRn/FRm→FRn	1111nnnnmmmm0011	—	—	—
FLOAT FPUL,FRn	(float)FPUL→FRn	1111nnnn00101101	—	—	—
FMAC FR0,FRm,FRn	FR0×FRm+FRn→FRn	1111nnnnmmmm1110	—	—	—
FMUL FRm,FRn	FRn×FRm→FRn	1111nnnnmmmm0010	—	—	—
FNEG FRn	FRn ^ H'80000000→FRn	1111nnnn01001101	—	—	—

指令	操作	指令码	特权	T 位	新规
FSQRT FRn	$\text{sqrt}(\text{FRn}) \rightarrow \text{FRn}^*$	1111nnnn01101101	—	—	—
FSUB FRm,FRn	$\text{FRn} - \text{FRm} \rightarrow \text{FRn}$	1111nnnnmmmm0001	—	—	—
FTRC FRm,FPUL	$(\text{long})\text{FRm} \rightarrow \text{FPUL}$	1111mmmm00111101	—	—	—

【注】 * $\text{sqrt}(\text{FRn})$ 表示 FRn 的平方根。

表 3.11 浮点双精度指令

指令	操作	指令码	特权	T 位	新规
FABS DRn	$\text{DRn} \& \text{H}'7\text{FFF FFFF FFFF FFFF} \rightarrow \text{DRn}$	1111nnn001011101	—	—	—
FADD DRm,DRn	$\text{DRn} + \text{DRm} \rightarrow \text{DRn}$	1111nnn0mmmm00000	—	—	—
FCMP/EQ DRm,DRn	当 $\text{DRn} = \text{DRm}$ 时, $1 \rightarrow \text{T}$ 否则 $0 \rightarrow \text{T}$	1111nnn0mmmm00100	—	比较结果	—
FCMP/GT DRm,DRn	当 $\text{DRn} > \text{DRm}$ 时, $1 \rightarrow \text{T}$ 否则 $0 \rightarrow \text{T}$	1111nnn0mmmm00101	—	比较结果	—
FDIV DRm,DRn	$\text{DRn} / \text{DRm} \rightarrow \text{DRn}$	1111nnn0mmmm00011	—	—	—
FCNVDS DRm,FPUL	$\text{double_to_float}(\text{DRm}) \rightarrow \text{FPUL}$	1111mmmm010111101	—	—	—
FCNVSD FPUL,DRn	$\text{float_to_double}(\text{FPUL}) \rightarrow \text{DRn}$	1111nnn010101101	—	—	—
FLOAT FPUL,DRn	$(\text{float})\text{FPUL} \rightarrow \text{DRn}$	1111nnn000101101	—	—	—
FMUL DRm,DRn	$\text{DRn} \times \text{DRm} \rightarrow \text{DRn}$	1111nnn0mmmm00010	—	—	—
FNEG DRn	$\text{DRn} \wedge \text{H}'8000 0000 0000 0000 \rightarrow \text{DRn}$	1111nnn001001101	—	—	—
FSQRT DRn	$\text{sqrt}(\text{DRn}) \rightarrow \text{DRn}^*$	1111nnn001101101	—	—	—
FSUB DRm,DRn	$\text{DRn} - \text{DRm} \rightarrow \text{DRn}$	1111nnn0mmmm00001	—	—	—
FTRC DRm,FPUL	$(\text{long})\text{DRm} \rightarrow \text{FPUL}$	1111mmmm000111101	—	—	—

【注】 * $\text{sqrt}(\text{DRn})$ 表示 DRn 的平方根。

表 3.12 浮点控制指令

指令	操作	指令码	特权	T 位	新规
LDS Rm,FPSCR	$\text{Rm} \rightarrow \text{FPSCR}$	0100mmmm01101010	—	—	—
LDS Rm,FPUL	$\text{Rm} \rightarrow \text{FPUL}$	0100mmmm01011010	—	—	—
LDS.L @Rm+,FPSCR	$(\text{Rm}) \rightarrow \text{FPSCR}$, $\text{Rm} + 4 \rightarrow \text{Rm}$	0100mmmm01100110	—	—	—
LDS.L @Rm+,FPUL	$(\text{Rm}) \rightarrow \text{FPUL}$, $\text{Rm} + 4 \rightarrow \text{Rm}$	0100mmmm01010110	—	—	—
STS FPSCR,Rn	$\text{FPSCR} \rightarrow \text{Rn}$	0000nnnn01101010	—	—	—
STS FPUL,Rn	$\text{FPUL} \rightarrow \text{Rn}$	0000nnnn01011010	—	—	—
STS.L FPSCR,@-Rn	$\text{Rn} - 4 \rightarrow \text{Rn}$, $\text{FPSCR} \rightarrow (\text{Rn})$	0100nnnn01100010	—	—	—
STS.L FPUL,@-Rn	$\text{Rn} - 4 \rightarrow \text{Rn}$, $\text{FPUL} \rightarrow (\text{Rn})$	0100nnnn01010010	—	—	—

表 3.13 浮点图形强化指令

指令	操作	指令码	特权	T 位	新规
FMOV DRm, XDn	DRm→XDn	1111nnn1mmm01100	—	—	—
FMOV XDm, DRn	XDm→DRn	1111nnn0mmm11100	—	—	—
FMOV XDm, XDn	XDm→XDn	1111nnn1mmm11100	—	—	—
FMOV @Rm, XDn	(Rm)→XDn	1111nnn1mmmm1000	—	—	—
FMOV @Rm+, XDn	(Rm)→XDn, Rm+8→Rm	1111nnn1mmmm1001	—	—	—
FMOV @(R0, Rm), XDn	(R0+Rm)→XDn	1111nnn1mmmm0110	—	—	—
FMOV XDm, @Rn	XDm→(Rn)	1111nnnnmmmm11010	—	—	—
FMOV XDm, @-Rn	Rn-8→Rn, XDm→(Rn)	1111nnnnmmmm11011	—	—	—
FMOV XDm, @(R0, Rn)	XDm→(R0+Rn)	1111nnnnmmmm10111	—	—	—
FIPR FVm, FVn	inner_product(FVm, FVn)→FR[n+3]	1111nnmm11101101	—	—	—
FTRV XMTRX, FVn	transform_vector(XMTRX, FVn)→FVn	1111nn0111111101	—	—	—
FRCHG	~FRSCR.FR→FRSCR.FR	1111101111111101	—	—	—
FSCHG	~FPSCR.SZ→FPSCR.SZ	1111001111111101	—	—	—
FPCHG	~FPSCR.PR→FPSCR.PR	1111011111111101	—	—	新规
FSRRA FRn	1/sqrt(FRn)→FRn*	1111nnnn01111101	—	—	新规
FSCA FPUL, DRn	sin(FPUL)→FRn cos(FPUL)→FR[n+1]	1111nnn011111101	—	—	新规

【注】 * sqrt(FRn) 表示 FRn 的平方根。

第 4 章 流水线操作

SH-4A 是 2 条指令并行型（2-ILP, Instruction-Level-Parallelism）的超标量流水线处理微处理器。指令的执行被流水线化，能并行执行 2 条指令。

4.1 流水线

基本流水线如图 4.1 所示。通常，流水线由取指令（I1、I2、I3）、解码/读寄存器（ID）、执行（E1、E2、E3）、回写（WB）8 个阶段构成。通过基本流水线的组合，执行 1 条指令。

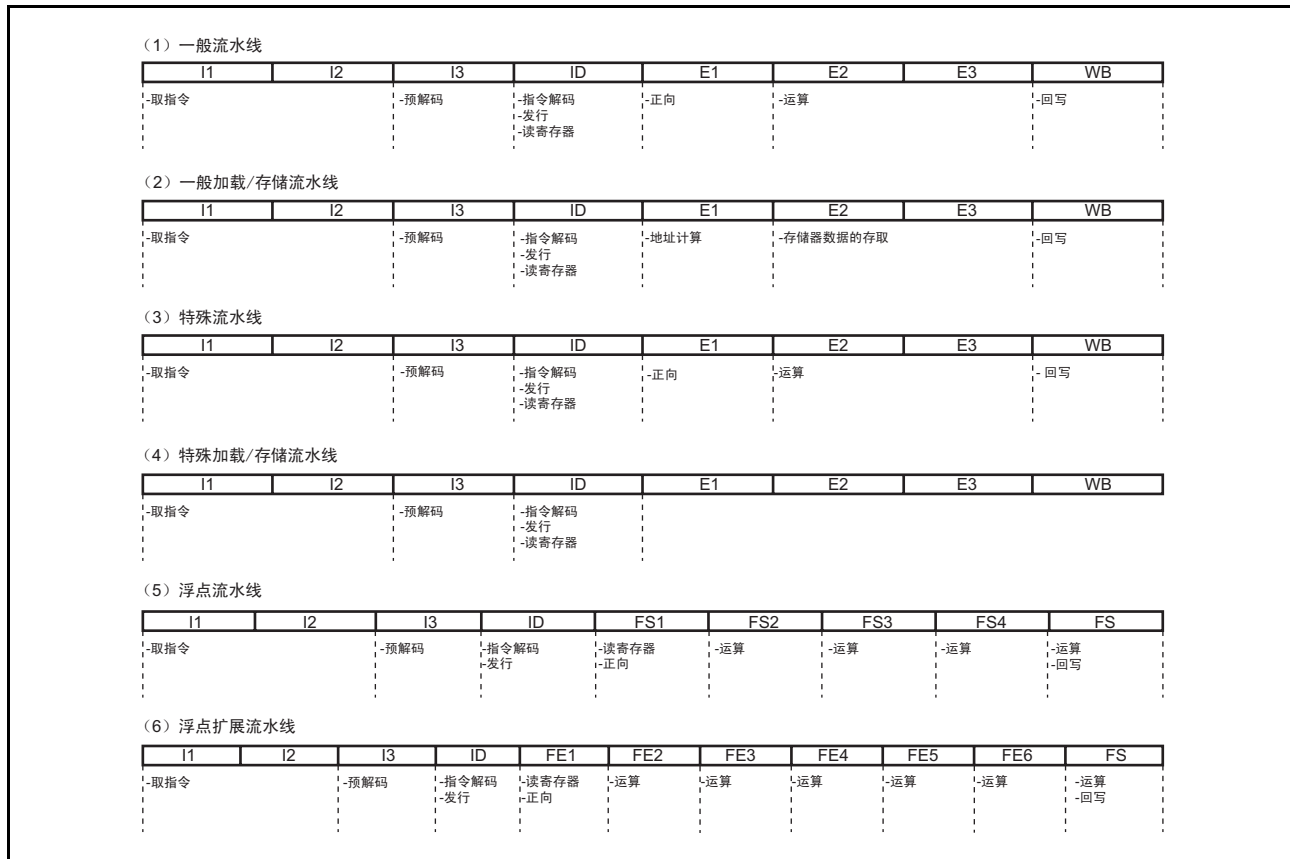
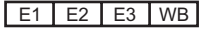
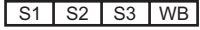
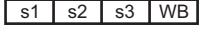


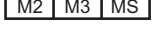

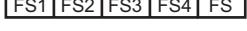




图 4.1 基本流水线

指令执行模式如图 4.2 所示。图 4.2 使用的记述及其含义如下所示：

表 4.1 指令执行模式的记述说明

记述	含义
	占有 CPU EX 流水线
	占有 CPU LS 流水线（有存储器存取时）
	占有 CPU LS 流水线（没有存储器存取时）
	占有 CPU EX 或者 LS
	占有 CPU EX 和 LS
	占有 CPU MULT 运算器
	占有 FPU-EX 流水线
	占有 FPU-LS 流水线
	锁定 ID 阶段
	占有 CPU 和 FPU 流水线

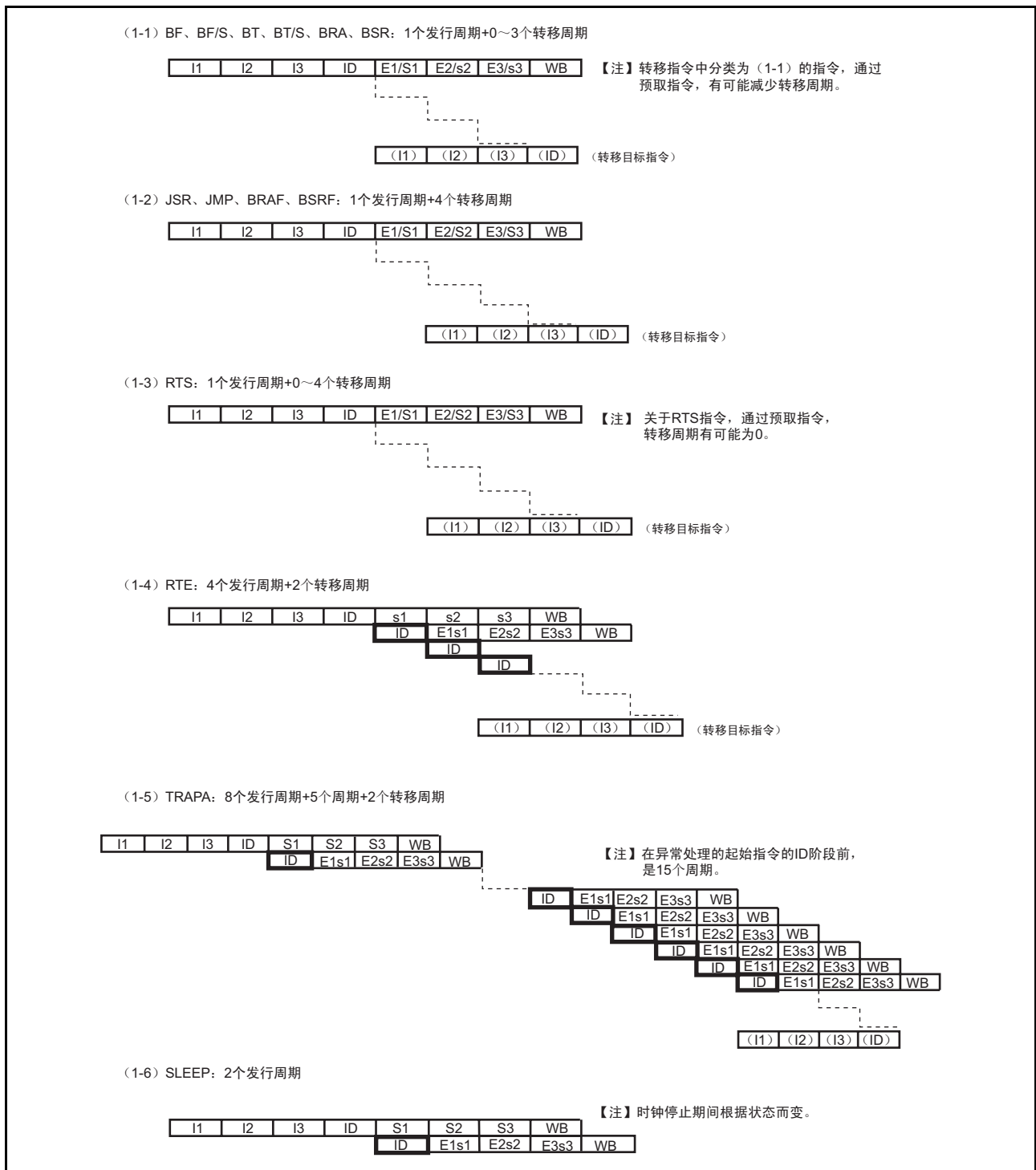


图 4.2 指令执行模式 (1)

(2-1) 单步运算 (EX类型) : 1个发行周期

EXT[SU],[BW], MOV_T, SWAP, XTRCT, ADD*, CMP*, DIV*, DT, NEG*, SUB*, AND, AND#, NOT, OR, OR#, TST, TST#, XOR, XOR#, ROT*, SHA*, SHL*, CLRS, CLRT, SETS, SETT

【注】AND#, OR#, TST#和XOR#不为GBR相对寻址方式。

I1	I2	I3	ID	E1	E2	E3	WB
----	----	----	----	----	----	----	----

(2-2) 单步运算 (LS类型) : 1个发行周期

MOVA

I1	I2	I3	ID	s1	s2	s3	WB
----	----	----	----	----	----	----	----

(2-3) 单步运算 (MT类型) : 1个发行周期

MOV#, NOP

I1	I2	I3	ID	E1/S1	E2/s2	E3/s3	WB
----	----	----	----	-------	-------	-------	----

(2-4) MOV (MT类型) : 1个发行周期

MOV

I1	I2	I3	ID	E1/s1	E2/s2	E3/S3	WB
----	----	----	----	-------	-------	-------	----

图 4.2 指令执行模式 (2)

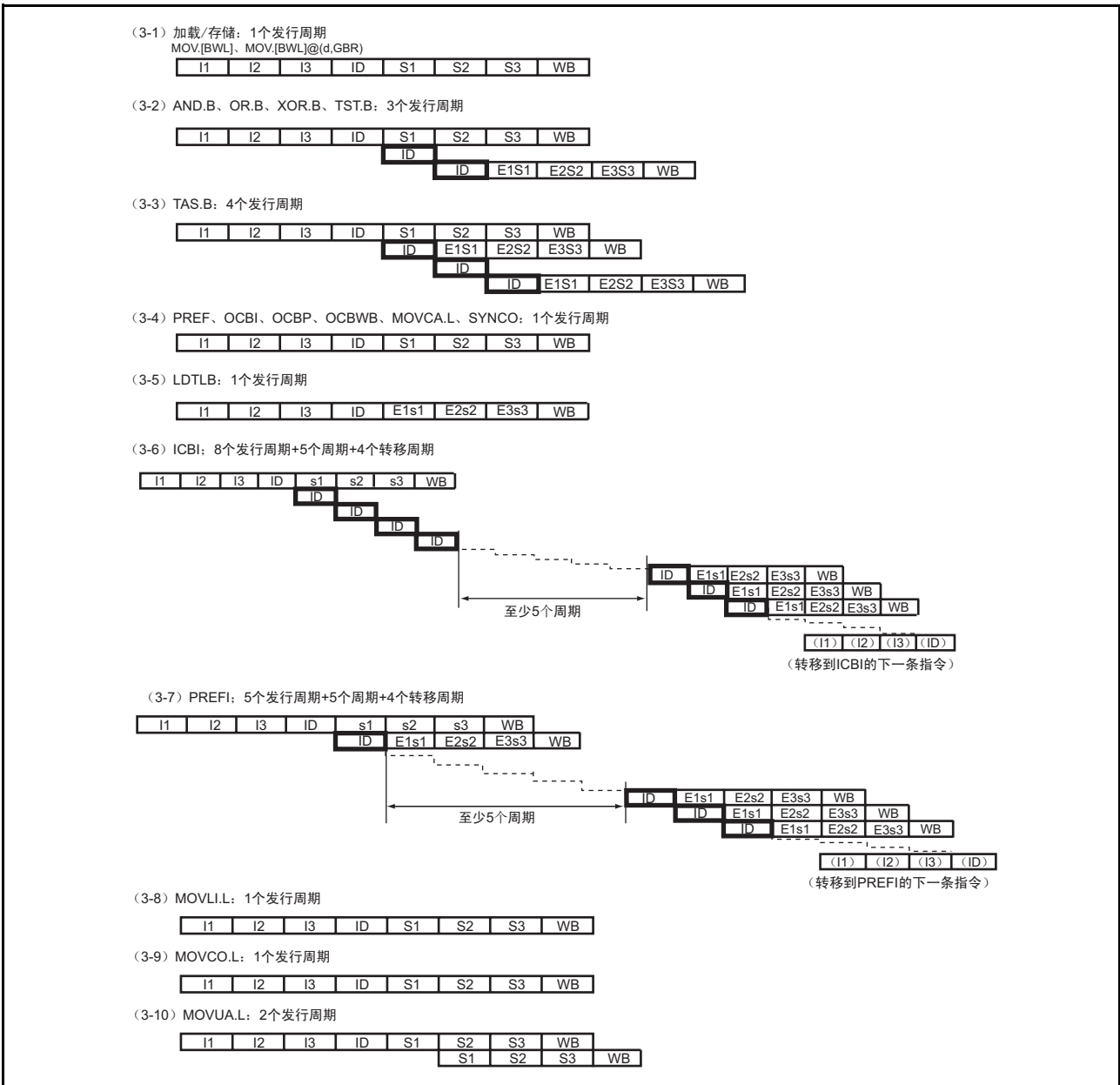


图 4.2 指令执行模式 (3)

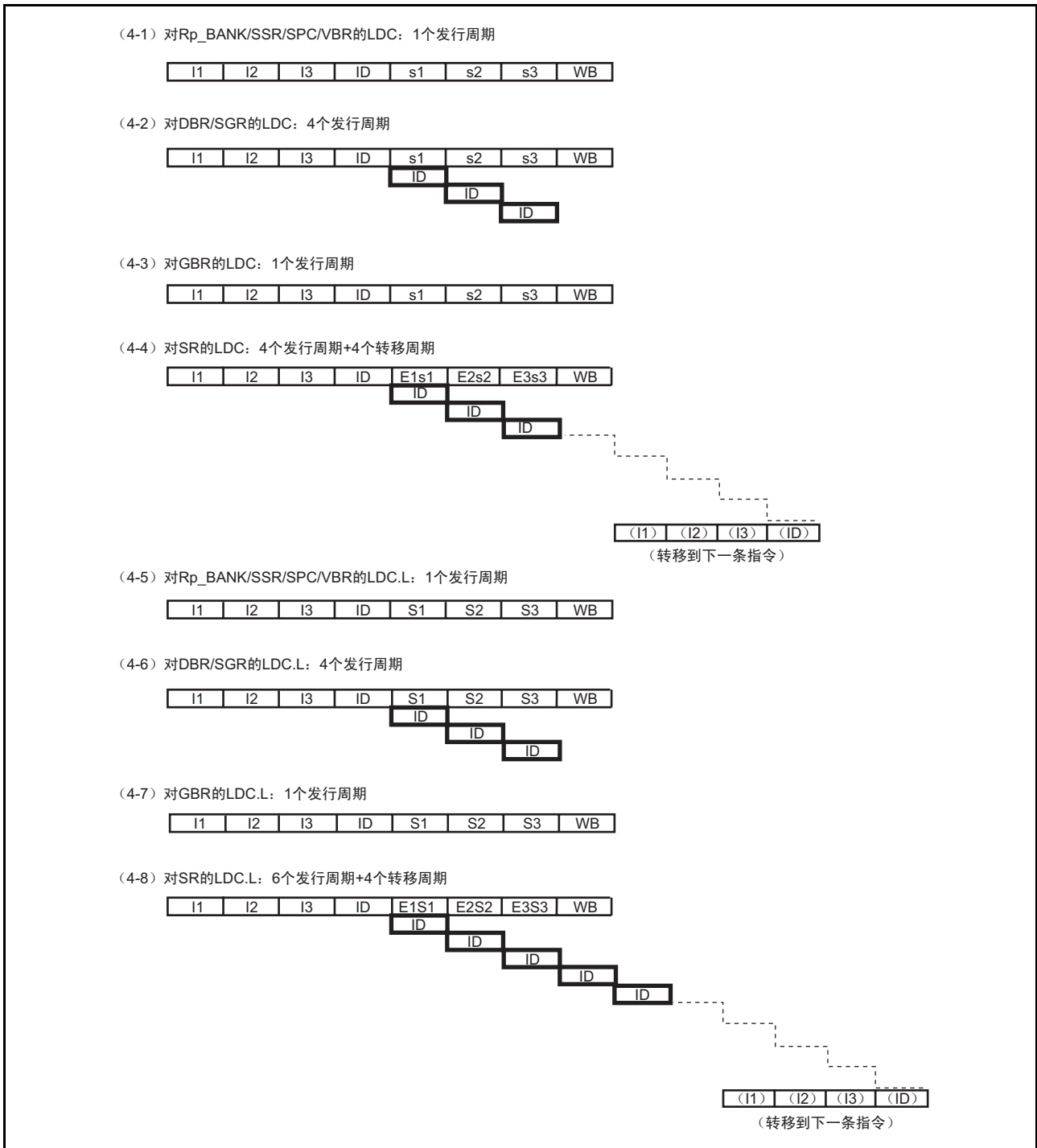


图 4.2 指令执行模式 (4)

(4-9) 来自DBR/GBR/Rp_BANK/SSR/SPC/VBR/SGR的STC: 1个发行周期

I1	I2	I3	ID	s1	s2	s3	WB
----	----	----	----	----	----	----	----

(4-10) 来自SR的STC: 1个发行周期

I1	I2	I3	ID	E1s1	E2s2	E3s3	WB
----	----	----	----	------	------	------	----

(4-11) 来自DBR/GBR/Rp_BANK/SSR/SPC/VBR/SGR的STC.L: 1个发行周期

I1	I2	I3	ID	S1	S2	S3	WB
----	----	----	----	----	----	----	----

(4-12) 来自SR的STC.L: 1个发行周期

I1	I2	I3	ID	E1S1	E2S2	E3S3	WB
----	----	----	----	------	------	------	----

(4-13) 对PR的LDS: 1个发行周期

I1	I2	I3	ID	s1	s2	s3	WB
----	----	----	----	----	----	----	----

(4-14) 对PR的LDS.L: 1个发行周期

I1	I2	I3	ID	S1	S2	S3	WB
----	----	----	----	----	----	----	----

(4-15) 来自PR的STS: 1个发行周期

I1	I2	I3	ID	s1	s2	s3	WB
----	----	----	----	----	----	----	----

(4-16) 来自PR的STS.L: 1个发行周期

I1	I2	I3	ID	S1	S2	S3	WB
----	----	----	----	----	----	----	----

(4-17) BSRF、BSR、JSR的延迟槽指令 (PR置位): 0个发行周期

(I1)	(I2)	(I3)	(ID)	(??1)	(??2)	(??3)	(WB)
------	------	------	------	-------	-------	-------	------

【注】在延迟槽指令的E3阶段PR值被更新。
在对延迟槽使用来自PR的STS、STS.L指令时, 使用更新后的PR值。

图 4.2 指令执行模式 (5)

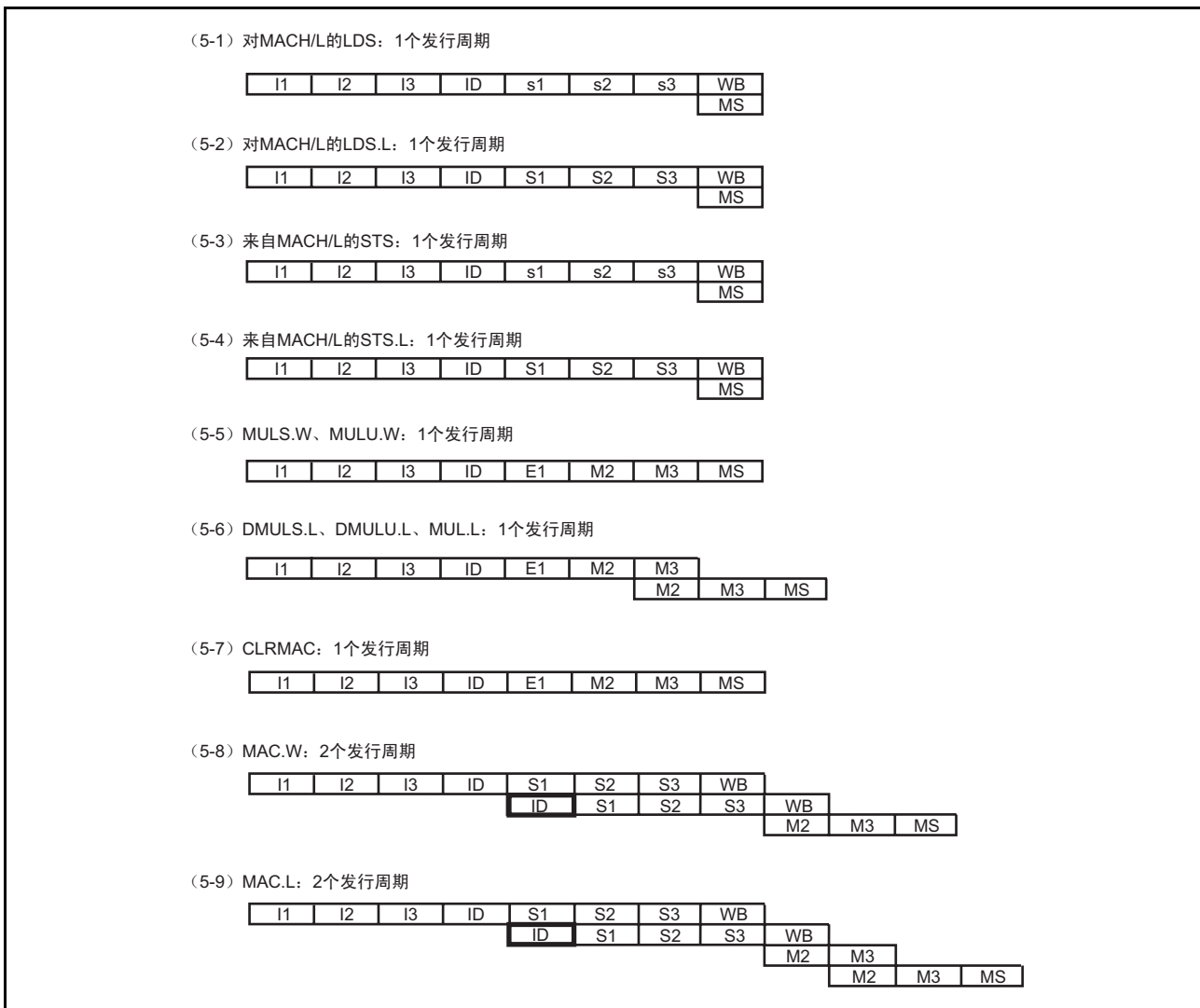


图 4.2 指令执行模式 (6)

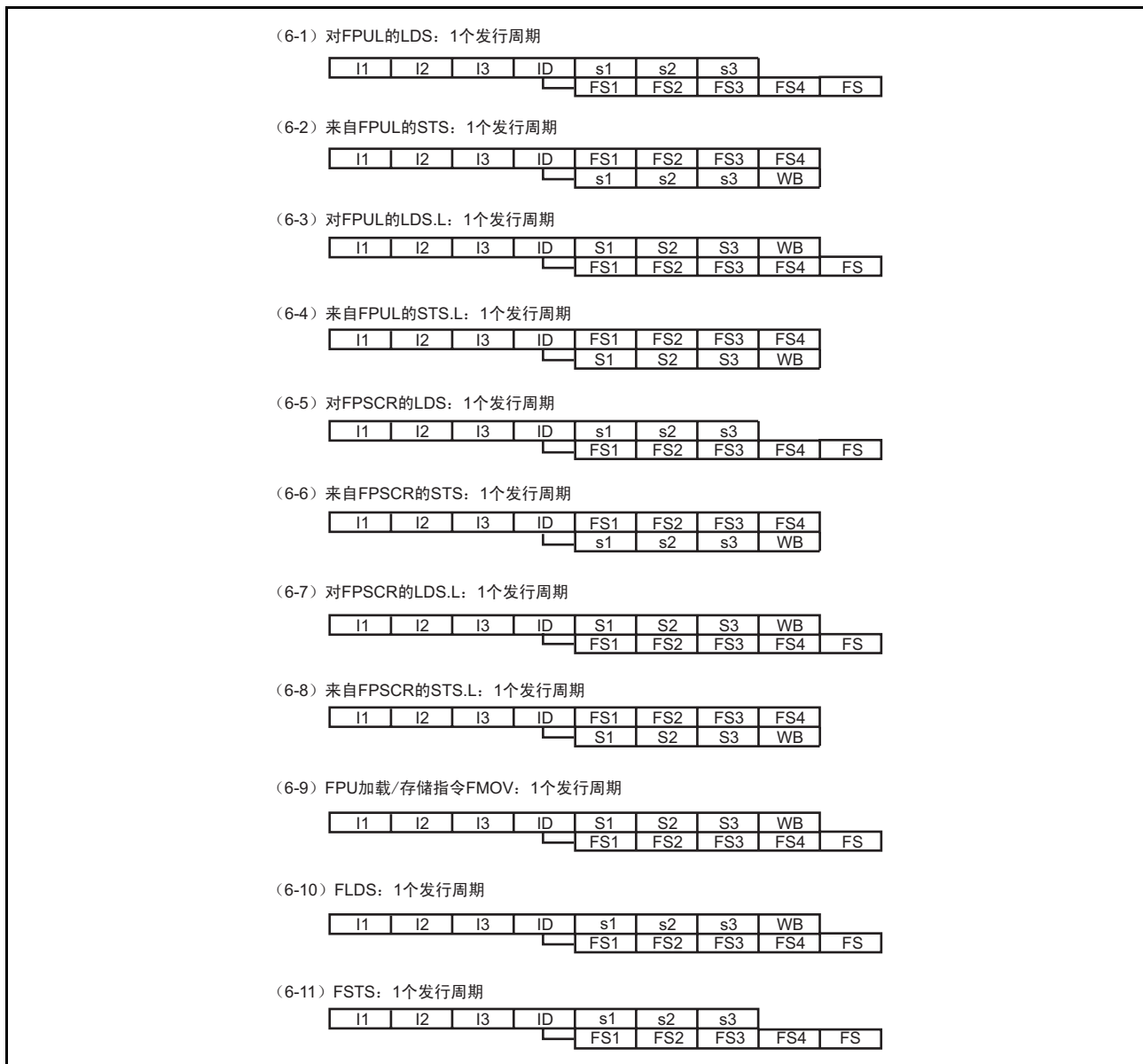


图 4.2 指令执行模式 (7)

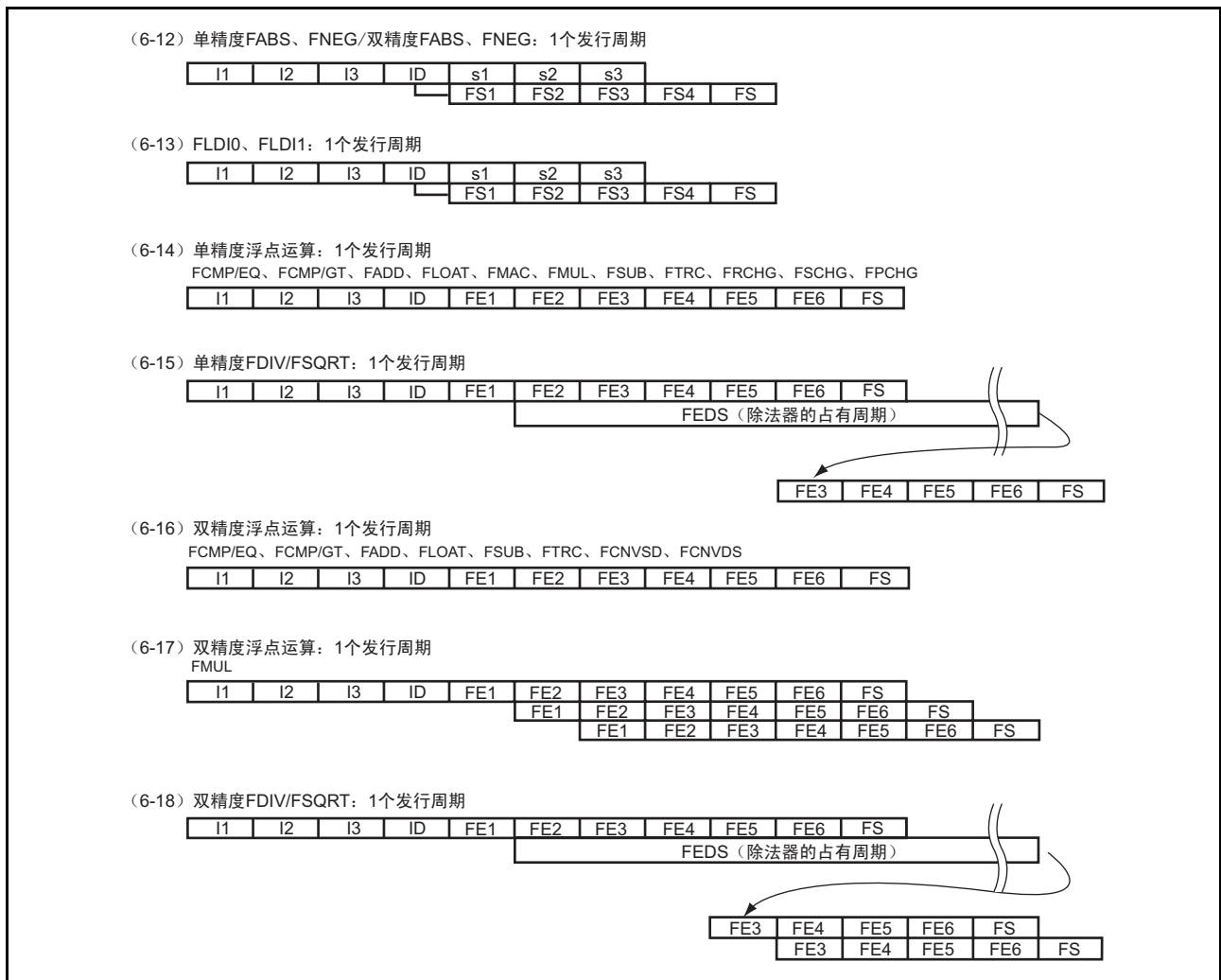


图 4.2 指令执行模式 (8)

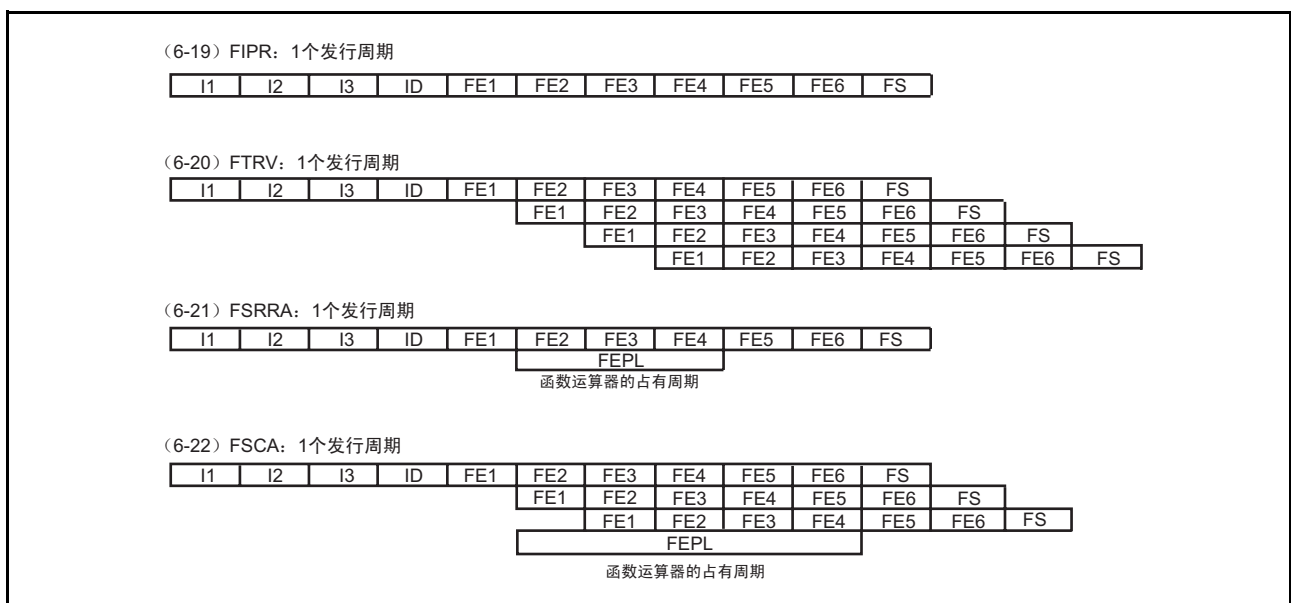


图 4.2 指令执行模式 (9)

4.2 并行执行性

按照使用的内部功能块，将指令分为表 4.2 所示的群。表 4.3 按群表示 2 条指令能并行执行的指令组合。例如，EX 群的 ADD 和 BR 群的 BRA 能并行执行。

表 4.2 指令群

指令群	指令			
EX	ADD ADDC ADDV AND #imm,R0 AND Rm,Rn CLRMAC CLRS CLRT CMP DIV0S DIV0U DIV1 DMUS.L DMULU.L	DT EXTS EXTU MOVT MUL.L MULS.W MULU.W NEG NEGC NOT OR #imm,R0 OR Rm,Rn ROTCL ROTCR	ROTL ROTR SETS SETT SHAD SHAL SHAR SHLD SHLL SHLL2 SHLL8 SHLL16 SHLR SHLR2	SHLR8 SHLR16 SUB SUBC SUBV SWAP TST #imm,R0 TST Rm,Rn XOR #imm,R0 XOR Rm,Rn XTRCT
MT	MOV #imm,Rn	MOV Rm,Rn	NOP	
BR	BF BF/S BRA	BRAF BSR BSRF	BT BT/S JMP	JSR RTS
LS	FABS FNEG FLDI0 FLDI1 FLDS FMOV @adr,FR FMOV FR,@adr FMOV FR,FR FMOV.S @adr,FR	FMOV.S FR,@adr FSTS LDC Rm,CR1 LDC.L @Rm+,CR1 LDS Rm,SR1 LDS Rm,SR2 LDS.L @adr,SR2 LDS.L @Rm+,SR1 LDS.L @Rm+,SR2	MOV.[BWL]@adr,R MOV.[BWL]R,@adr MOVA MOVCA.L MOVUA OCBI OCBP OCBWB PREF	STC CR2,Rn STC.L CR2,@-Rn STS SR2,Rn STS.L SR2,@-Rn STS SR1,Rn STS.L SR1,@-Rn
FE	FADD FSUB FCMP(S/D) FCNVDS FCNVSD	FDIV FIPR FLOAT FMAC FMUL	FRCHG FSCHG FSQRT FTRC FTRV	FSCA FSRRA FPCHG
CO	AND.B #imm,@(R0,GBR) ICBI LDC Rm,DBR LDC Rm,SGR LDC Rm,SR LDC.L @Rm+,DBR LDC.L @Rm+,SGR	LDC.L @Rm+,SR LDTLB MAC.L MAC.W MOVCO MOVLI OR.B #imm,@(R0,GBR)	PREFI RTE SLEEP STC SR,Rn STC.L SR,@-Rn SYNCO TAS.B	TRAPA TST.B #imm,@(R0,GBR) XOR.B #imm,@(R0,GBR)

【符号说明】 R : Rm/Rn
 @adr: 地址
 SR1 : MACH/MACL/PR
 SR2 : FPUL/FPSCR
 CR1 : GBR/Rp_BANK/SPC/SSR/VBR
 CR2 : CR1/DBR/SGR
 FR : FRm/FRn/DRm/DRn/XDm/XDn

2 条指令的同时执行限于以下情况：

1. $addr$ （前指令）和 $addr+2$ （后指令）的 2 条指令不超过 1K 字节（最小的页大小）。
2. 在表 4.3（前后指令的组合表）中能同时执行（为○）的指令。
3. $addr$ 的指令和此前的指令无数据冲突。
4. $addr+2$ 的指令和此前的指令无数据冲突。
5. 2 条指令都有效。

表 4.3 前后指令的组合表

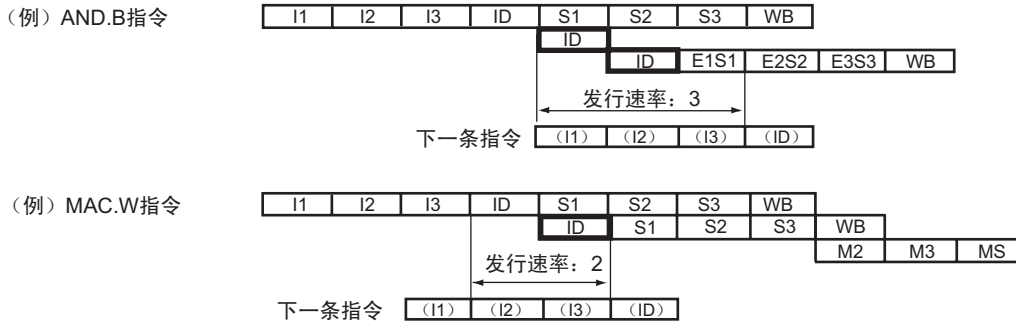
		前指令 ($addr$)					
		EX	MT	BR	LS	FE	CO
后指令 ($addr+2$)	EX	×	○	○	○	○	
	MT	○	○	○	○	○	
	BR	○	○	×	○	○	
	LS	○	○	○	×	○	
	FE	○	○	○	○	×	
	CO						

4.3 发行速率和执行状态

指令的发行速率和执行状态如表 4.4 所示。表 4.4 中的指令群对应表 4.2 的分类。另外，本节所示的发行速率和执行状态不考虑流水线延迟的补偿周期。

(1) 发行速率

发行速率表示发行一条指令和下一条指令的间隔。



(2) 执行状态

执行状态是用以下基准表示指令占有流水线的周期数。

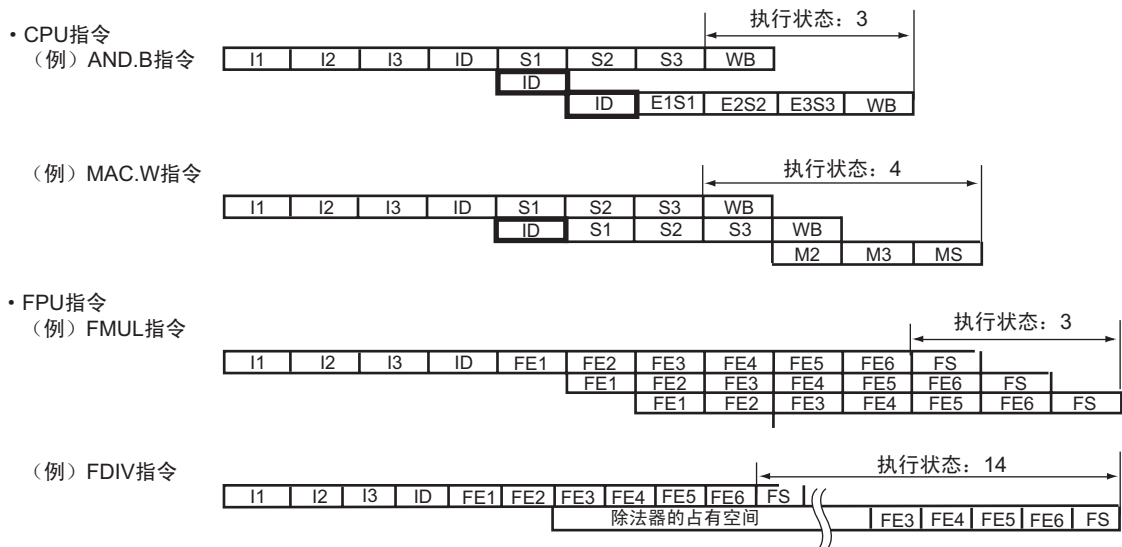


表 4.4 发行速率和执行状态

功能分类	No.	指令	指令群	发行速率	执行状态	执行模式	
数据传送指令	1	EXTS.B	Rm,Rn	EX	1	2-1	
	2	EXTS.W	Rm,Rn	EX	1	2-1	
	3	EXTU.B	Rm,Rn	EX	1	2-1	
	4	EXTU.W	Rm,Rn	EX	1	2-1	
	5	MOV	Rm,Rn	MT	1	1	2-4
	6	MOV	#imm,Rn	MT	1	1	2-3
	7	MOVA	@(disp,PC),R0	LS	1	1	2-2
	8	MOV.W	@(disp,PC),Rn	LS	1	1	3-1
	9	MOV.L	@(disp,PC),Rn	LS	1	1	3-1
	10	MOV.B	@Rm,Rn	LS	1	1	3-1
	11	MOV.W	@Rm,Rn	LS	1	1	3-1
	12	MOV.L	@Rm,Rn	LS	1	1	3-1
	13	MOV.B	@Rm+,Rn	LS	1	1	3-1
	14	MOV.W	@Rm+,Rn	LS	1	1	3-1
	15	MOV.L	@Rm+,Rn	LS	1	1	3-1
	16	MOV.B	@(disp,Rm),R0	LS	1	1	3-1
	17	MOV.W	@(disp,Rm),R0	LS	1	1	3-1
	18	MOV.L	@(disp,Rm),Rn	LS	1	1	3-1
	19	MOV.B	@(R0,Rm),Rn	LS	1	1	3-1
	20	MOV.W	@(R0,Rm),Rn	LS	1	1	3-1
	21	MOV.L	@(R0,Rm),Rn	LS	1	1	3-1
	22	MOV.B	@(disp,GBR),R0	LS	1	1	3-1
	23	MOV.W	@(disp,GBR),R0	LS	1	1	3-1
	24	MOV.L	@(disp,GBR),R0	LS	1	1	3-1
	25	MOV.B	Rm,@Rn	LS	1	1	3-1
	26	MOV.W	Rm,@Rn	LS	1	1	3-1
	27	MOV.L	Rm,@Rn	LS	1	1	3-1
	28	MOV.B	Rm,@-Rn	LS	1	1	3-1
	29	MOV.W	Rm,@-Rn	LS	1	1	3-1
	30	MOV.L	Rm,@-Rn	LS	1	1	3-1
	31	MOV.B	R0,@(disp,Rn)	LS	1	1	3-1
	32	MOV.W	R0,@(disp,Rn)	LS	1	1	3-1
	33	MOV.L	Rm,@(disp,Rn)	LS	1	1	3-1
	34	MOV.B	Rm,@(R0,Rn)	LS	1	1	3-1
	35	MOV.W	Rm,@(R0,Rn)	LS	1	1	3-1
	36	MOV.L	Rm,@(R0,Rn)	LS	1	1	3-1
	37	MOV.B	R0,@(disp,GBR)	LS	1	1	3-1
	38	MOV.W	R0,@(disp,GBR)	LS	1	1	3-1
	39	MOV.L	R0,@(disp,GBR)	LS	1	1	3-1
	40	MOVCA.L	R0,@Rn	LS	1	1	3-4
	41	MOVCO.L	R0,@Rn	CO	1	1	3-9
	42	MOVLI.L	@Rm,R0	CO	1	1	3-8
	43	MOVUA.L	@Rm,R0	LS	2	2	3-10
	44	MOVUA.L	@Rm+,R0	LS	2	2	3-10
	45	MOV.T	Rn	EX	1	1	2-1
	46	OCBI	@Rn	LS	1	1	3-4
	47	OCBP	@Rn	LS	1	1	3-4

功能分类	No.	指令	指令群	发行速率	执行状态	执行模式	
数据传送指令	48	OCBWB @Rn	LS	1	1	3-4	
	49	PREF @Rn	LS	1	1	3-4	
	50	SWAP.B Rm,Rn	EX	1	1	2-1	
	51	SWAP.W Rm,Rn	EX	1	1	2-1	
	52	XTRCT Rm,Rn	EX	1	1	2-1	
定点算术指令	53	ADD Rm,Rn	EX	1	1	2-1	
	54	ADD #imm,Rn	EX	1	1	2-1	
	55	ADDC Rm,Rn	EX	1	1	2-1	
	56	ADDV Rm,Rn	EX	1	1	2-1	
	57	CMP/EQ #imm,R0	EX	1	1	2-1	
	58	CMP/EQ Rm,Rn	EX	1	1	2-1	
	59	CMP/GE Rm,Rn	EX	1	1	2-1	
	60	CMP/GT Rm,Rn	EX	1	1	2-1	
	61	CMP/HI Rm,Rn	EX	1	1	2-1	
	62	CMP/HS Rm,Rn	EX	1	1	2-1	
	63	CMP/PL Rn	EX	1	1	2-1	
	64	CMP/PZ Rn	EX	1	1	2-1	
	65	CMP/STR Rm,Rn	EX	1	1	2-1	
	66	DIV0S Rm,Rn	EX	1	1	2-1	
	67	DIV0U	EX	1	1	2-1	
	68	DIV1 Rm,Rn	EX	1	1	2-1	
	69	DMULS.L Rm,Rn	EX	1	2	5-6	
	70	DMULU.L Rm,Rn	EX	1	2	5-6	
	71	DT Rn	EX	1	1	2-1	
	72	MAC.L @Rm+,@Rn+	CO	2	5	5-9	
	73	MAC.W @Rm+,@Rn+	CO	2	4	5-8	
	74	MUL.L Rm,Rn	EX	1	2	5-6	
	75	MULS.W Rm,Rn	EX	1	1	5-5	
	76	MULU.W Rm,Rn	EX	1	1	5-5	
	77	NEG Rm,Rn	EX	1	1	2-1	
	78	NEGC Rm,Rn	EX	1	1	2-1	
	79	SUB Rm,Rn	EX	1	1	2-1	
	80	SUBC Rm,Rn	EX	1	1	2-1	
	81	SUBV Rm,Rn	EX	1	1	2-1	
	逻辑指令	82	AND Rm,Rn	EX	1	1	2-1
		83	AND #imm,R0	EX	1	1	2-1
84		AND.B #imm,@(R0,GBR)	CO	3	3	3-2	
85		NOT Rm,Rn	EX	1	1	2-1	
86		OR Rm,Rn	EX	1	1	2-1	
87		OR #imm,R0	EX	1	1	2-1	
88		OR.B #imm,@(R0,GBR)	CO	3	3	3-2	
89		TAS.B @Rn	CO	4	4	3-3	
90		TST Rm,Rn	EX	1	1	2-1	
91		TST #imm,R0	EX	1	1	2-1	
92		TST.B #imm,@(R0,GBR)	CO	3	3	3-2	
93		XOR Rm,Rn	EX	1	1	2-1	
94		XOR #imm,R0	EX	1	1	2-1	
95		XOR.B #imm,@(R0,GBR)	CO	3	3	3-2	

功能分类	No.	指令		指令群	发行速率	执行状态	执行模式
移位指令	96	ROTL	Rn	EX	1	1	2-1
	97	ROTR	Rn	EX	1	1	2-1
	98	ROTCL	Rn	EX	1	1	2-1
	99	ROTCR	Rn	EX	1	1	2-1
	100	SHAD	Rm,Rn	EX	1	1	2-1
	101	SHAL	Rn	EX	1	1	2-1
	102	SHAR	Rn	EX	1	1	2-1
	103	SHLD	Rm,Rn	EX	1	1	2-1
	104	SHLL	Rn	EX	1	1	2-1
	105	SHLL2	Rn	EX	1	1	2-1
	106	SHLL8	Rn	EX	1	1	2-1
	107	SHLL16	Rn	EX	1	1	2-1
	108	SHLR	Rn	EX	1	1	2-1
	109	SHLR2	Rn	EX	1	1	2-1
	110	SHLR8	Rn	EX	1	1	2-1
111	SHLR16	Rn	EX	1	1	2-1	
转移指令	112	BF	disp	BR	1+0 ~ 2	1	1-1
	113	BF/S	disp	BR	1+0 ~ 2	1	1-1
	114	BT	disp	BR	1+0 ~ 2	1	1-1
	115	BT/S	disp	BR	1+0 ~ 2	1	1-1
	116	BRA	disp	BR	1+0 ~ 2	1	1-1
	117	BRAF	Rm	BR	1+3	1	1-2
	118	BSR	disp	BR	1+0 ~ 2	1	1-1
	119	BSRF	Rm	BR	1+3	1	1-2
	120	JMP	@Rn	BR	1+3	1	1-2
	121	JSR	@Rn	BR	1+3	1	1-2
	122	RTS		BR	1+0 ~ 3	1	1-3
	系统控制指令	123	NOP		MT	1	1
124		CLRMAC		EX	1	1	5-7
125		CLRS		EX	1	1	2-1
126		CLRT		EX	1	1	2-1
127		ICBI	@Rn	CO	8+5+3	13	3-6
128		SETS		EX	1	1	2-1
129		SETT		EX	1	1	2-1
130		PREFI	@Rn	CO	5+5+3	10	3-7
131		SYNCO		CO	不定值	不定值	3-4
132		TRAPA	#imm	CO	8+5+1	13	1-5
133		RTE		CO	4+1	4	1-4
134		SLEEP		CO	不定值	不定值	1-6
135		LDTLB		CO	1	1	3-5
136		LDC	Rm,DBR	CO	4	4	4-2
137		LDC	Rm,SGR	CO	4	4	4-2
138		LDC	Rm,GBR	LS	1	1	4-3
139		LDC	Rm,Rp_BANK	LS	1	1	4-1
140		LDC	Rm,SR	CO	4+3	4	4-4
141		LDC	Rm,SSR	LS	1	1	4-1
142	LDC	Rm,SPC	LS	1	1	4-1	
143	LDC	Rm,VBR	LS	1	1	4-1	

功能分类	No.	指令	指令群	发行速率	执行状态	执行模式	
系统控制指令	144	LDC.L	@Rm+,DBR	CO	4	4	4-6
	145	LDC.L	@Rm+,SGR	CO	4	4	4-6
	146	LDC.L	@Rm+,GBR	LS	1	1	4-7
	147	LDC.L	@Rm+,Rp_BANK	LS	1	1	4-5
	148	LDC.L	@Rm+,SR	CO	6+3	4	4-8
	149	LDC.L	@Rm+,SSR	LS	1	1	4-5
	150	LDC.L	@Rm+,SPC	LS	1	1	4-5
	151	LDC.L	@Rm+,VBR	LS	1	1	4-5
	152	LDS	Rm,MACH	LS	1	1	5-1
	153	LDS	Rm,MACL	LS	1	1	5-1
	154	LDS	Rm,PR	LS	1	1	4-13
	155	LDS.L	@Rm+,MACH	LS	1	1	5-2
	156	LDS.L	@Rm+,MACL	LS	1	1	5-2
	157	LDS.L	@Rm+,PR	LS	1	1	4-14
	158	STC	DBR,Rn	LS	1	1	4-9
	159	STC	SGR,Rn	LS	1	1	4-9
	160	STC	GBR,Rn	LS	1	1	4-9
	161	STC	Rp_BANK,Rn	LS	1	1	4-9
	162	STC	SR,Rn	CO	1	1	4-10
	163	STC	SSR,Rn	LS	1	1	4-9
	164	STC	SPC,Rn	LS	1	1	4-9
	165	STC	VBR,Rn	LS	1	1	4-9
	166	STC.L	DBR,@-Rn	LS	1	1	4-11
	167	STC.L	SGR,@-Rn	LS	1	1	4-11
	168	STC.L	GBR,@-Rn	LS	1	1	4-11
	169	STC.L	Rp_BANK,@-Rn	LS	1	1	4-11
	170	STC.L	SR,@-Rn	CO	1	1	4-12
	171	STC.L	SSR,@-Rn	LS	1	1	4-11
	172	STC.L	SPC,@-Rn	LS	1	1	4-11
	173	STC.L	VBR,@-Rn	LS	1	1	4-11
	174	STS	MACH,Rn	LS	1	1	5-3
	175	STS	MACL,Rn	LS	1	1	5-3
	176	STS	PR,Rn	LS	1	1	4-15
177	STS.L	MACH,@-Rn	LS	1	1	5-4	
178	STS.L	MACL,@-Rn	LS	1	1	5-4	
179	STS.L	PR,@-Rn	LS	1	1	4-16	
单精度浮点指令	180	FLDI0	FRn	LS	1	1	6-13
	181	FLDI1	FRn	LS	1	1	6-13
	182	FMOV	FRm,FRn	LS	1	1	6-9
	183	FMOV.S	@Rm,FRn	LS	1	1	6-9
	184	FMOV.S	@Rm+,FRn	LS	1	1	6-9
	185	FMOV.S	@(R0,Rm),FRn	LS	1	1	6-9
	186	FMOV.S	FRm,@Rn	LS	1	1	6-9
	187	FMOV.S	FRm,@-Rn	LS	1	1	6-9
	188	FMOV.S	FRm,@(R0,Rn)	LS	1	1	6-9
	189	FLDS	FRm,FPUL	LS	1	1	6-10
	190	FSTS	FPUL,FRn	LS	1	1	6-11
	191	FABS	FRn	LS	1	1	6-12

功能分类	No.	指令		指令群	发行速率	执行状态	执行模式
单精度浮点指令	192	FADD	FRm,FRn	FE	1	1	6-14
	193	FCMP/EQ	FRm,FRn	FE	1	1	6-14
	194	FCMP/GT	FRm,FRn	FE	1	1	6-14
	195	FDIV	FRm,FRn	FE	1	14	6-15
	196	FLOAT	FPUL,FRn	FE	1	1	6-14
	197	FMAC	FR0,FRm,FRn	FE	1	1	6-14
	198	FMUL	FRm,FRn	FE	1	1	6-14
	199	FNEG	FRn	LS	1	1	6-12
	200	FSQRT	FRn	FE	1	30	6-15
	201	FSUB	FRm,FRn	FE	1	1	6-14
	202	FTRC	FRm,FPUL	FE	1	1	6-14
	203	FMOV	DRm,DRn	LS	1	1	6-9
	204	FMOV	@Rm,DRn	LS	1	1	6-9
	205	FMOV	@Rm+,DRn	LS	1	1	6-9
	206	FMOV	@(R0,Rm),DRn	LS	1	1	6-9
	207	FMOV	DRm,@Rn	LS	1	1	6-9
	208	FMOV	DRm,@-Rn	LS	1	1	6-9
	209	FMOV	DRm,@(R0,Rn)	LS	1	1	6-9
	双精度浮点指令	210	FABS	DRn	LS	1	1
211		FADD	DRm,DRn	FE	1	1	6-16
212		FCMP/EQ	DRm,DRn	FE	1	1	6-16
213		FCMP/GT	DRm,DRn	FE	1	1	6-16
214		FCNVDS	DRm,FPUL	FE	1	1	6-16
215		FCNVSD	FPUL,DRn	FE	1	1	6-16
216		FDIV	DRm,DRn	FE	1	14	6-18
217		FLOAT	FPUL,DRn	FE	1	1	6-16
218		FMUL	DRm,DRn	FE	1	3	6-17
219		FNEG	DRn	LS	1	1	6-12
220		FSQRT	DRn	FE	1	30	6-18
221		FSUB	DRm,DRn	FE	1	1	6-16
222		FTRC	DRm,FPUL	FE	1	1	6-16
FPU 系统控制指令	223	LDS	Rm,FPUL	LS	1	1	6-1
	224	LDS	Rm,FPSCR	LS	1	1	6-5
	225	LDS.L	@Rm+,FPUL	LS	1	1	6-3
	226	LDS.L	@Rm+,FPSCR	LS	1	1	6-7
	227	STS	FPUL,Rn	LS	1	1	6-2
	228	STS	FPSCR,Rn	LS	1	1	6-6
	229	STS.L	FPUL,@-Rn	LS	1	1	6-4
	230	STS.L	FPSCR,@-Rn	LS	1	1	6-8
图形强化指令	231	FMOV	DRm,XDn	LS	1	1	6-9
	232	FMOV	XDm,DRn	LS	1	1	6-9
	233	FMOV	XDm,XDn	LS	1	1	6-9
	234	FMOV	@Rm,XDn	LS	1	1	6-9
	235	FMOV	@Rm+,XDn	LS	1	1	6-9
	236	FMOV	@(R0,Rm),XDn	LS	1	1	6-9
	237	FMOV	XDm,@Rn	LS	1	1	6-9
	238	FMOV	XDm,@-Rn	LS	1	1	6-9
	239	FMOV	XDm,@(R0,Rn)	LS	1	1	6-9

功能分类	No.	指令		指令群	发行速率	执行状态	执行模式
图形强化指令	240	FIPR	FVm,FVn	FE	1	1	6-19
	241	FRCHG		FE	1	1	6-14
	242	FSCHG		FE	1	1	6-14
	243	FPCHG		FE	1	1	6-14
	244	FSRRA	FRn	FE	1	1	6-21
	245	FSCA	FPUL,DRn	FE	1	3	6-22
	246	FTRV	XMTRX,FVn	FE	1	4	6-20

第 5 章 异常处理

5.1 概要

异常处理是指在检测到复位、一般异常和中断时，通过特定的程序进行必要的处理。例如，当执行中的指令发生异常结束时，需要通过适当的处理，控制原程序的返回、异常报告以及异常结束。为了支持此功能，对异常结束，产生异常处理请求，并且将控制流程递交给用户编制的异常处理程序等，这样的处理统称为异常处理。

SH-4A 的异常处理分为复位、一般异常和中断 3 种。

5.2 寄存器说明

有关异常处理的寄存器结构如表 5.1 所示。

表 5.1 寄存器结构

名称	略称	R/W	P4 区地址 *	区域 7 地址	存取长度
TRAPA 异常寄存器	TRA	R/W	H'FF00 0020	H'1F00 0020	32
异常事件寄存器	EXPEVT	R/W	H'FF00 0024	H'1F00 0024	32
中断事件寄存器	INTEVT	R/W	H'FF00 0028	H'1F00 0028	32
非支持检测异常寄存器	EXPMASK	R/W	H'FF2F 0004	H'1F2F 0004	32

【注】 * P4 区地址是使用虚拟地址空间 P4 区的地址，区域 7 地址是使用 TLB 存取物理地址空间区域 7 的地址。

表 5.2 各处理模式的寄存器状态

名称	略称	上电复位	睡眠	待机
TRAPA 异常寄存器	TRA	不定值	保持	保持
异常事件寄存器	EXPEVT	H'0000 0000	保持	保持
中断事件寄存器	INTEVT	不定值	保持	保持
非支持检测异常寄存器	EXPMASK	H'0000 0000	保持	保持

5.2.1 TRAPA 异常寄存器 (TRA)

TRAPA 异常寄存器 (TRA) 是设定 TRAPA 指令的 8 位立即数 (imm) 的寄存器。在执行 TRAPA 指令时, 通过硬件自动设定 TRA。也能通过软件更改 TRA。

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
	—	—	—	—	—	—	TRACODE								—	—
初始值:	0	0	0	0	0	0	—	—	—	—	—	—	—	—	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

位	位名	初始值	R/W	说明
31 ~ 10	—	全 0	R	保留位 有关此位的读写, 请参照“产品使用上的注意事项”。
9 ~ 2	TRACODE	不定值	R/W	TRAPA 码 设定 TRAPA 指令的 8 位立即数。
1、0	—	全 0	R	保留位 有关此位的读写, 请参照“产品使用上的注意事项”。

5.2.2 异常事件寄存器 (EXPEVT)

给异常事件寄存器 (EXPEVT) 设定由复位或者一般异常事件所产生的 12 位异常码。在接受异常时, 通过硬件自动设定异常码。也能通过软件更改 EXPEVT。

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	
	—	—	—	—	EXPCODE											—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0/1	0	0	0	0	0	
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

位	位名	初始值	R/W	说明
31 ~ 12	—	全 0	R	保留位 有关此位的读写, 请参照“产品使用上的注意事项”。
11 ~ 0	EXPCODE	H'000 或者 H'020	R/W	异常码 设定复位或者一般异常的异常码。详细内容请参照表 5.3。

5.2.3 中断事件寄存器 (INTEVT)

给中断事件寄存器 (INTEVT) 设定由中断请求产生的 14 位异常码。在接受异常时, 通过硬件自动设定异常码。也能通过软件更改 INTEVT。

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
	—	—	INTCODE													
初始值:	0	0	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 14	—	全 0	R	保留位 有关此位的读写, 请参照“产品使用上的注意事项”。
13 ~ 0	INTCODE	不定值	R/W	异常码 设定中断的异常码。详细内容请参照表 5.3。

5.2.4 非支持检测异常寄存器 (EXPMASK)

EXPMASK 寄存器在使用下列 1 ~ 3 的功能时能允许和禁止发生异常。今后的 SuperH 系列有可能不支持此 1 ~ 3 的功能。通过预先使用 EXPMASK 寄存器的异常发生功能, 软件能调查是否正在使用这些功能, 并且在今后的 SuperH 系列不支持此功能时能容易地进行软件移植。

1. 当 RTE 指令的延迟槽不是 NOP 指令时
2. 当转移指令的延迟槽是 SLEEP 指令时
3. 当执行了 IC/OC 存储器映像联想写操作时

根据非支持检测异常寄存器 (EXPMASK) 的值, 功能 1 ~ 2 能发生槽非法指令异常, 功能 3 能发生数据地址错误异常。

通过给 EXPMASK 寄存器的相应位写 1, 能禁止发生异常。但是为了维持今后的兼容性, 强烈建议编制不使用上述功能的程序。

必须用 CPU 的存储指令更新 EXPMASK 寄存器, 而且在更新后读取一次寄存器以后, 必须执行下列的任意一项操作。通过执行此操作, 能用更新后的寄存器值保证运行。

- 执行 RTE 指令
- 对于任意地址 (也可以是不能缓存的区域) 执行 ICBI 指令

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
	—	—	—	—	—	—	—	—	—	—	—	MM CAW	—	—	BRDS SLP	RTE DS
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 5	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。
4	MMCAW	0	R/W	存储器映像高速缓存联想写操作位 0: 禁止存储器映像联想写操作（发生数据地址错误异常）。 1: 允许存储器映像联想写操作。 详细内容请参照“8.6.5 存储器映像的联想写操作”。
3、2	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。
1	BRDSSLP	0	R/W	延迟槽 SLEEP 指令位 0: 禁止延迟槽的 SLEEP 指令（SLEEP 指令为槽非法指令）。 1: 允许延迟槽的 SLEEP 指令。
0	RTEDS	0	R/W	RTE 延迟槽位 0: 禁止 RTE 指令延迟槽的非 NOP 指令（非 NOP 指令为槽非法指令）。 1: 允许 RTE 指令延迟槽的非 NOP 指令。

5.3 异常处理功能

5.3.1 异常处理的流程

在异常处理中，将程序计数器（PC）、状态寄存器（SR）和 R15 的内容分别保存到保存程序计数器（SPC）、保存状态寄存器（SSR）和保存通用寄存器（SGR），然后根据向量地址开始执行相应的异常处理程序。异常处理程序是用户按照各种异常内容而编制的程序，为了结束异常处理程序并返回原程序，需要从异常处理执行返回指令（RTE）。通过此指令能恢复 PC 和 SR 的内容，返回到发生异常时的一般处理程序。RTE 指令不将 SGR 的内容回写到 R15。

基本的异常处理流程如下，SR 位的含义请参照“第 2 章 编程模型”。

1. 将 PC、SR 和 R15 的内容分别保存到 SPC、SSR 和 SGR。
2. 将 SR 的块位（BL）置 1。
3. 将 SR 的模式位（MD）置 1。
4. SR 的寄存器存储体位（RB）置 1。
5. 在复位时，将 SR 的 FPU 禁止位（FD）置 0。
6. 给异常源的异常事件寄存器（EXPEVT）或者中断事件寄存器（INTEVT）的 bit13~0 设定异常码。
7. 当 CPUOPM 的中断模式转换位（INTMU）为 1 时，将 SR 的中断屏蔽级的位（IMASK）更改为中断接受级。
8. 转移到规定的异常处理向量地址，开始执行异常处理程序。

5.3.2 异常处理向量地址

复位向量地址固定在 H'A000 0000。异常和中断的向量地址为向量基址加上各事件偏移量的地址，向量基址通过软件设定到向量基址寄存器（VBR）。例如，TLB 未命中异常的偏移量为 H'0000 0400，如果预先将 H'9C08 0000 设定到 VBR，异常处理向量地址就为 H'9C08 0400。如果在异常处理向量地址再次发生异常，就为双重异常，难以恢复，因此必须对向量地址指定不为地址转换对象的 P1 和 P2 区的地址。

5.4 异常的种类和优先顺序

异常的种类、优先顺序、向量地址和异常 / 中断码如表 5.3 所示。

表 5.3 异常一览表

异常分类	执行形态	异常	优先级	优先顺序	异常转移目标		异常码	
					向量基址	偏移量		
复位	中断型	上电复位	1	1	H'A000 0000	—	H'000	
		H-UDI 复位	1	1	H'A000 0000	—	H'000	
		指令 TLB 多重命中异常	1	2	H'A000 0000	—	H'140	
		数据 TLB 多重命中异常	1	3	H'A000 0000	—	H'140	
一般异常	再执行型	指令执行前的用户断点*	2	0	(VBR/DBR)	H'100/—	H'1E0	
		指令地址错误	2	1	(VBR)	H'100	H'0E0	
		指令 TLB 未命中异常	2	2	(VBR)	H'400	H'040	
		指令 TLB 保护违反异常	2	3	(VBR)	H'100	H'0A0	
		一般非法指令异常	2	4	(VBR)	H'100	H'180	
		槽非法指令异常	2	4	(VBR)	H'100	H'1A0	
		一般 FPU 禁止异常	2	4	(VBR)	H'100	H'800	
		槽 FPU 禁止异常	2	4	(VBR)	H'100	H'820	
		数据地址错误（读）	2	5	(VBR)	H'100	H'0E0	
		数据地址错误（写）	2	5	(VBR)	H'100	H'100	
		数据 TLB 未命中异常（读）	2	6	(VBR)	H'400	H'040	
		数据 TLB 未命中异常（写）	2	6	(VBR)	H'400	H'060	
		数据 TLB 保护违反异常（读）	2	7	(VBR)	H'100	H'0A0	
		数据 TLB 保护违反异常（写）	2	7	(VBR)	H'100	H'0C0	
		FPU 异常	2	8	(VBR)	H'100	H'120	
		初始页写异常	2	9	(VBR)	H'100	H'080	
		完成型	无条件陷阱（TRAPA）	2	4	(VBR)	H'100	H'160
			执行指令后的用户断点*	2	10	(VBR/DBR)	H'100/—	H'1E0
	中断	完成型	非屏蔽中断	3	—	(VBR)	H'600	H'1C0
			一般中断请求	4	—	(VBR)	H'600	—

优先级：首先按优先级进行排序，然后在同一优先级内按优先顺序排序（数值越小优先级越高）。

异常转移目标：在复位时，将控制转移到 H'A000 0000；否则转移到（VBR+ 偏移）。

异常码：在复位或者一般异常时，保存到 EXPEVT；在中断时，保存到 INTEVT。

【注】* 当 CBCR.UBDE=1 时，PC=DBR；否则 PC=VBR+H'100。

5.5 异常流程

5.5.1 异常流程

指令执行和异常处理的基本运行大致如图 5.1 所示。在此为了便于说明，原则上以逐条执行指令进行说明。在图 5.1 中，表示了异常种类（复位、一般异常、中断）间的优先顺序，异常成立时的寄存器设定限于 SSR、SPC、SGR、EXPEVT/INTEVT、SR 和 PC，但是根据异常，也有通过硬件自动设定的寄存器。详细内容请参照“5.6 各异常的说明”。另外，有关执行延迟转移指令和延迟槽指令过程中的异常处理以及产生 2 次数据存取指令，请参照“5.6.4 发生多次异常时的优先顺序”。

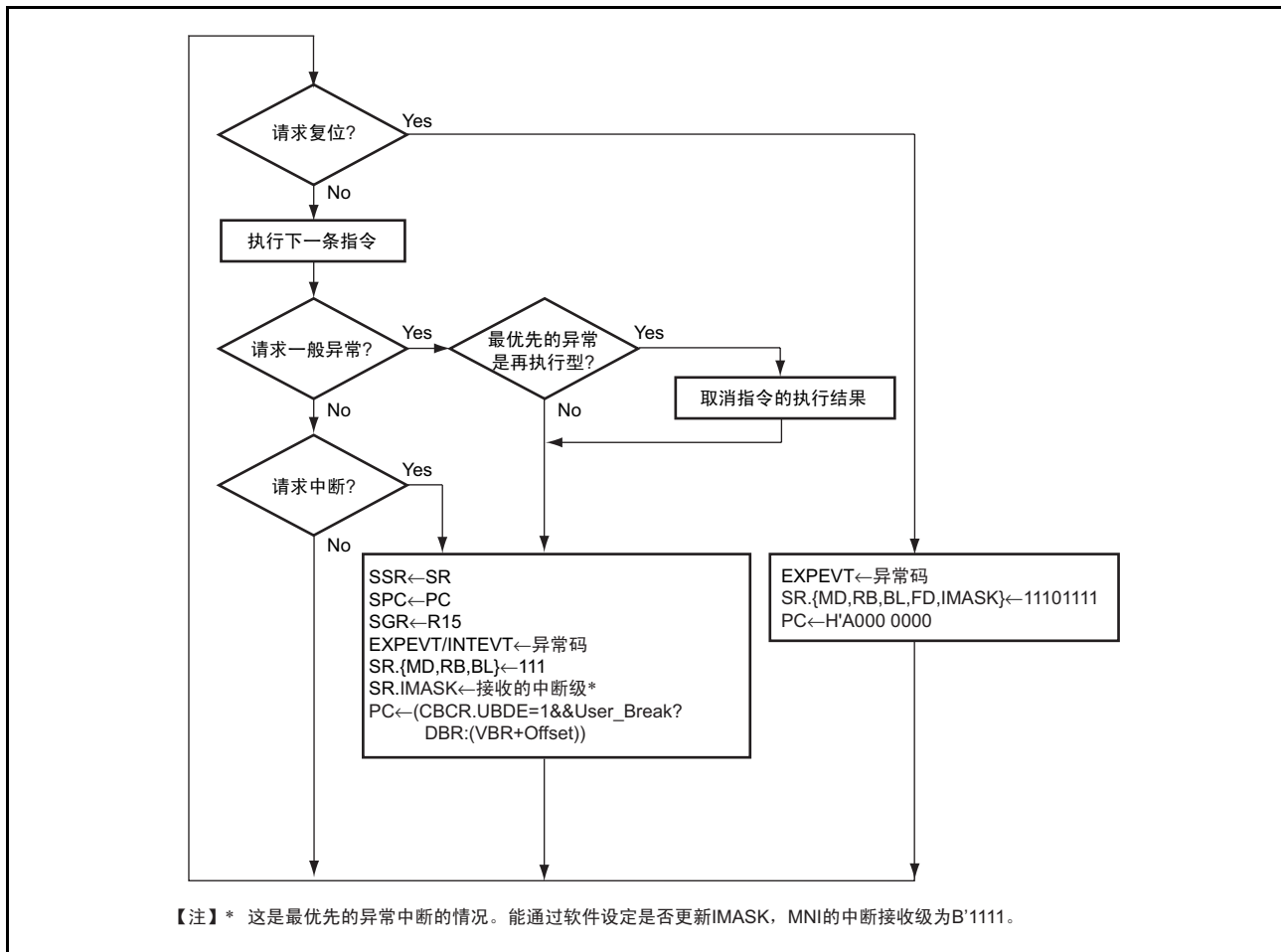


图 5.1 指令执行和异常处理

5.5.2 异常源的接受

因为在同时发生 2 个以上的异常时需决定要接受的异常，所以给全部的异常规定了优先顺序。一般异常中的 5 种异常：一般非法指令异常、槽非法指令异常、一般 FPU 禁止异常、槽 FPU 禁止异常、无条件陷阱异常是在各自的指令分析过程进行检测并且在指令流水线中不同时发生，因此优先顺序为相同的值。但是，按照指令的流程顺序（程序顺序）进行异常处理，即与后续指令的异常相比，优先接受先前指令的异常。一般异常的接受顺序例子如图 5.2 所示。

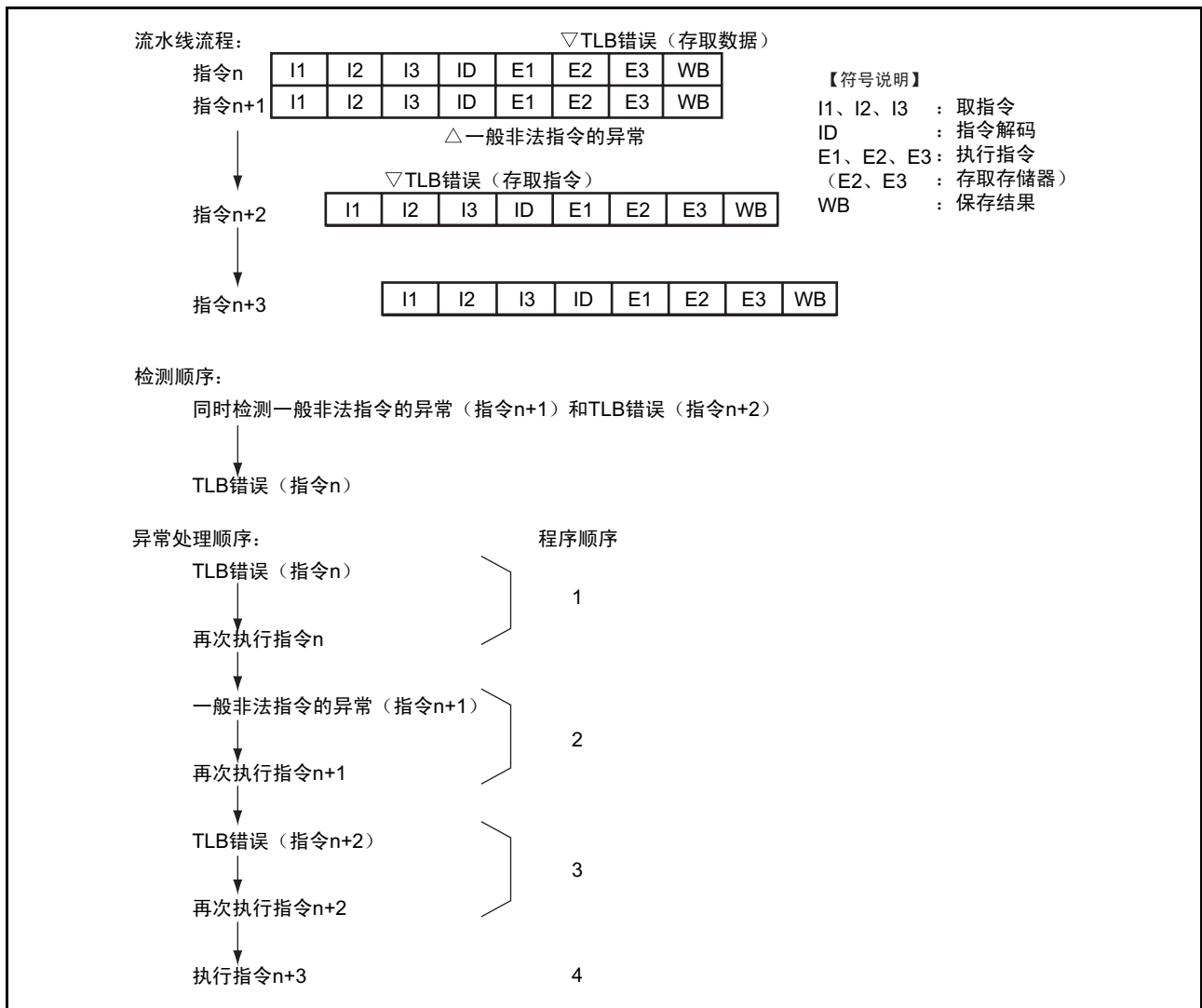


图 5.2 一般异常的接受顺序例子

5.5.3 异常请求和 BL 位

当 SR 的 BL 位为 0 时，接受一般异常和中断。

当 SR 的 BL 位为 1 时，如果发生除用户断点以外的一般异常，CPU 内部寄存器和其他模块的寄存器就变为上电复位后的状态，并转移到和复位相同的地址 (H'A000 0000)。有关发生用户断点时的运行，请参照“第 30 章 用户断点控制器 (UBC)”。另外，当发生一般中断时，就保留中断请求，在通过软件将 BL 位清 0 后接受中断。当发生非屏蔽中断 (NMI) 时，能通过软件设定是保留还是接受。

通常，为了能多重接受异常状态，在保存 SPC 和 SSR 后将 SR 的 BL 位清 0。

5.5.4 异常处理的返回

使用 RTE 指令从异常处理返回。通过 RTE 指令，将 SPC 恢复为 PC，SSR 恢复为 SR，然后转移到 SPC 的地址，从异常处理程序返回。如果将 SPC 和 SSR 保存到外部存储器，就必须先将 SR 的 BL 位置 1，然后恢复 SPC 和 SSR，发行 RTE 指令。

5.6 各异常的说明

对于各异常处理的运行，说明其发生源、发生时的转移目标地址以及转移时的处理器运行。

5.6.1 复位

(1) 上电复位

- 条件：
上电复位请求
- 运行：
将H'000设定到EXPEVT，在进行CPU和内部外围模块的初始化后转移到复位向量（H'A0000000）。
详细内容请参照各章的寄存器说明。在接通电源时必须进行上电复位。

(2) H-UDI 复位

- 异常源：SDIR.TI[7:4]为B'0110（无效）或者B'0111（有效）。
- 转移目标地址：H'A000 0000
- 转移时的运行：
将异常码H'000设定到EXPEVT。进行VBR和SR的初始化，转移到PC=H'A000 0000。进行CPU和内部外围模块的初始化。详细内容请参照“第31章 用户调试接口（H-UDI）”和各章的寄存器说明。

(3) 指令 TLB 多重命中异常

- 异常源：多个ITLB地址相同。
- 转移目标地址：H'A000 0000
- 转移时的运行：
将发生此异常的虚拟地址（32位）设定到TEA，对应的虚拟页码（22位）设定到PTEH[31:10]。
PTEH的ASID表示此异常发生时的ASID。
将异常码H'140设定到EXPEVT。进行VBR和SR的初始化，转移到PC=H'A000 0000。和上电复位时一样进行CPU和内部外围模块的初始化。详细内容请参照各章的寄存器说明。

(4) 数据 TLB 多重命中异常

- 异常源：多个UTLB地址相同。
- 转移目标地址：H'A000 0000
- 转移时的运行：
将发生此异常的虚拟地址（32位）设定到TEA，对应的虚拟页码（22位）设定到PTEH[31:10]。
PTEH的ASID表示此异常发生时的ASID。
将异常码H'140设定到EXPEVT。进行VBR和SR的初始化，转移到PC=H'A000 0000。和上电复位时一样进行CPU和内部外围模块的初始化。详细内容请参照各章的寄存器说明。

5.6.2 一般异常

(1) 数据 TLB 未命中异常

- 异常源：UTLB 的地址比较结果为地址不相同。
- 转移目标地址：VBR+H'0000 0400
- 转移时的运行：

将发生此异常的虚拟地址（32位）设定到TEA，对应的虚拟页码（22位）设定到PTEH[31:10]。PTEH的ASID表示此异常发生时的ASID。

将发生此异常的指令的PC和SR分别保存到SPC和SSR，此时的R15保存到SGR。

在读操作时，将异常码H'040设定到EXPEVT；在写操作时，将异常码H'060设定到EXPEVT。将SR的BL位、MD位和RB位置1，转移到PC=VBR+H'0400。

为了提高TLB未命中处理的速度，区分了不同于其他异常的偏移量。

```
Data_TLB_miss_exception()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = read_access ? H'00000040 : H'00000060;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'00000400;
}
```

(2) 指令 TLB 未命中异常

- 异常源：ITLB 的地址比较结果为地址不相同。
- 转移目标地址：VBR+H'0000 0400
- 转移时的运行：

将发生此异常的虚拟地址（32位）设定到TEA，对应的虚拟页码（22位）设定到PTEH[31:10]。PTEH的ASID表示此异常发生时的ASID。

将发生此异常的指令的PC和SR分别保存到SPC和SSR，此时的R15保存到SGR。

将异常码H'040设定到EXPEVT。将SR的BL位、MD位和RB位置1，转移到PC=VBR+H'0400。

为了提高TLB未命中处理的速度，区分了不同于其他异常的偏移量。

```
ITLB_miss_exception()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'00000040;
    SR.MD = 1;
}
```

```

    SR.RB =1;
    SR.BL =1;
    PC = VBR + H'00000400;
}

```

(3) 初始页写异常

- 异常源：虽然在存储存取中命中 TLB，但是已写位 D=0。
- 转移目标地址：VBR+H'0000 0100
- 转移时的运行：

将发生此异常的虚拟地址（32位）设定到 TEA，对应的虚拟页码（22位）设定到 PTEH[31:10]。PTEH 的 ASID 表示此异常发生时的 ASID。

将发生此异常的指令的 PC 和 SR 分别保存到 SPC 和 SSR，此时的 R15 保存到 SGR。

将异常码 H'080 设定到 EXPEVT。将 SR 的 BL 位、MD 位和 RB 位置 1，转移到 PC=VBR+H'0100。

```

Initial_write_exception()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'00000080;
    SR.MD =1;
    SR.RB =1;
    SR.BL =1;
    PC = VBR + H'00000100;
}

```

(4) 数据 TLB 保护违反异常

- 异常源：存取操作与表 5.4 和表 5.5 所示的 UTLB 保护信息（PR 位或者 EPR 位）不一致。

表 5.4 UTLB 保护信息（TLB 兼容模式）

PR	特权模式	用户模式
00	只读	不可存取
01	可读写	不可存取
10	只读	只读
11	可读写	可读写

表 5.5 UTLB 保护信息 (TLB 扩展模式)

EPR[5]	可否在特权模式中进行读操作
1	可读
0	不可读
EPR[3]	可否在特权模式中进行写操作
1	可写
0	不可写
EPR[2]	可否在用户模式中进行读操作
1	可读
0	不可读
EPR[1]	可否在用户模式中进行写操作
1	可写
0	不可写

- 转移目标地址: VBR+H'0000 0100
- 转移时的运行:

将发生此异常的虚拟地址 (32 位) 设定到 TEA, 对应的虚拟页码 (22 位) 设定到 PTEH[31:10]。PTEH 的 ASID 表示此异常发生时的 ASID。

将发生此异常的指令的 PC 和 SR 分别保存到 SPC 和 SSR, 此时的 R15 保存到 SGR。

在读操作时, 将异常码 H'0A0 设定到 EXPEVT; 在写操作时, 将异常码 H'0C0 设定到 EXPEVT。将 SR 的 BL 位、MD 位和 RB 位置 1, 转移到 PC=VBR+H'0100。

```
Data_TLB_protection_violation_exception()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = read_access ? H'000000A0 : H'000000C0;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'00000100;
}
```

(5) 指令 TLB 保护违反异常

- 异常源: 存取操作与表 5.6 和表 5.7 所示的 ITLB 保护信息 (PR 位或 EPR) 不一致。

表 5.6 ITLB 保护信息 (TLB 兼容模式)

PR	特权模式	用户模式
0	可存取	不可存取
1	可存取	可存取

表 5.7 ITLB 保护信息 (TLB 扩展模式)

EPR[5]、EPR[3]	可否在特权模式中执行
11、01	可执行
10	只有 ICBI 不可执行，其他可执行
00	不可执行

EPR[2]、EPR[0]	可否在用户模式中执行
11、01	可执行
10	只有 ICBI 不可执行，其他可执行
00	不可执行

- 转移目标地址：VBR+H'0000 0100
- 转移时的运行：
将发生此异常的虚拟地址（32位）设定到TEA，对应的虚拟页码（22位）设定到PTEH[31:10]。PTEH的ASID表示此异常发生时的ASID。
将发生此异常的指令的PC和SR分别保存到SPC和SSR，此时的R15保存到SGR。
将异常码H'0A0设定到EXPEVT。将SR的BL位、MD位和RB位置1，转移到PC=VBR+H'0100。

```
ITLB_protection_violation_exception()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'000000A0;
    SR.MD =1;
    SR.RB =1;
    SR.BL =1;
    PC = VBR + H'00000100;
}
```

(6) 数据地址错误

- 异常源：
 - 从非字边界（2n+1）存取字数据。
 - 从非长字数据边界（4n+1，4n+2，4n+3）存取长字数据（MOVUA指令除外。）。
 - 从非4个字数据边界（8n+1，8n+2，8n+3，8n+4，8n+5，8n+6，8n+7）存取4个字数据。
 - 在用户模式中存取区域H'8000 0000～H'FFFF FFFF。
但是，在用户模式中，能存取H'E000 0000～H'E3FF FFFF和H'E500 0000～H'E5FF FFFF。详细内容请参照“第7章 存储器管理单元（MMU）”和“第9章 内部存储器”。

— EXPMASK 寄存器的 MMCAW 位为 0 并且进行 IC/OC 存储器映像联想写操作。

有关存储器映像联想写操作，请参照“8.6.5 存储器映像的联想写操作”。

- 转移目标地址：VBR+H'0000 0100

- 转移时的运行：

将发生此异常的虚拟地址（32 位）设定到 TEA，对应的虚拟页号码（22 位）设定到 PTEH[31:10]。

PTEH 的 ASID 表示此异常发生时的 ASID。

将发生此异常的指令的 PC 和 SR 分别保存到 SPC 和 SSR，此时的 R15 保存到 SGR。

在读操作时，将异常码 H'0E0 设定到 EXPEVT；在写操作时，将异常码 H'100 设定到 EXPEVT。将 SR 的 BL 位、MD 位和 RB 位置 1，转移到 PC=VBR+H'0100。详细内容请参照“第 7 章 存储器管理单元 (MMU)”。

```
Data_address_error()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = read_access? H'000000E0: H'00000100;
    SR.MD = 1;
    SR.RB = 1;
    SR.BL = 1;
    PC = VBR + H'00000100;
}
```

(7) 指令地址错误

- 异常源：

— 从非字边界（2n+1）取指令。

— 在用户模式中从区域 H'8000 0000 ~ H'FFFF FFFF 取指令。

但是，在用户模式中，能存取 H'E500 0000 ~ H'E5FF FFFF。详细内容请参照“第 9 章 内部存储器”。

- 转移目标地址：VBR+H'0000 0100

- 转移时的运行：

将发生此异常的虚拟地址（32 位）设定到 TEA，对应的虚拟页号码（22 位）设定到 PTEH[31:10]。

PTEH 的 ASID 表示此异常发生时的 ASID。

将发生此异常的指令的 PC 和 SR 分别保存到 SPC 和 SSR，此时的 R15 保存到 SGR。

将异常码 H'0E0 设定到 EXPEVT。将 SR 的 BL 位、MD 位和 RB 位置 1，转移到 PC=VBR+H'0100。详细内容请参照“第 7 章 存储器管理单元 (MMU)”。

```
Instruction_address_error()
{
    TEA = EXCEPTION_ADDRESS;
    PTEH.VPN = PAGE_NUMBER;
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'000000E0;
```

```

    SR.MD =1;
    SR.RB =1;
    SR.BL =1;
    PC = VBR + H'00000100;
}

```

(8) 无条件陷阱

- 异常源：TRAPA 指令的执行
- 转移目标地址：VBR+H'0000 0100
- 转移时的运行：

由于是处理完成型的异常，所以将 TRAPA 指令的下一条指令的 PC 保存到 SPC，执行 TRAPA 指令时的 SR 和 R15 保存到 SSR 和 SGR。将 TRAPA 指令中的 8 位立即数乘 4，设定到 TRA[9:0]。将异常码 H'160 设定到 EXPEVT。将 SR 的 BL 位、MD 位和 RB 位置 1，转移到 PC=VBR+H'0100。

```

TRAPA_exception()
{
    SPC = PC +2;
    SSR = SR;
    SGR = R15;
    TRA = imm <<2;
    EXPEVT = H'00000160;
    SR.MD =1;
    SR.RB =1;
    SR.BL =1;
    PC = VBR + H'00000100;
}

```

(9) 一般非法指令异常

- 异常源：
 - 对延迟槽以外的未定义指令进行解码。
延迟转移指令：JMP、JSR、BRA、BRAf、BSR、BSRf、RTS、RTE、BT/S、BF/S
未定义指令：H'FFFD
 - 在用户模式中对延迟槽以外的特权指令进行解码。
特权指令：LDC、STC、RTE、LDTLB、SLEEP
但是，通过 LDC 和 STC 存取 GBR 的指令除外。

- 转移目标地址：VBR+H'0000 0100

- 转移时的运行：

将发生此异常的指令的 PC 和 SR 分别保存到 SPC 和 SSR，此时的 R15 保存到 SGR。

将异常码 H'180 设定到 EXPEVT。将 SR 的 BL 位、MD 位和 RB 位置 1，转移到 PC=VBR+H'0100。如果对 H'FFFD 以外的未定义码进行解码，就不保证运行。

```

General_illegal_instruction_exception()
{
    SPC = PC;
    SSR = SR;
    SGR = R15;
}

```

```

    EXPEVT = H'00000180;
    SR.MD =1;
    SR.RB =1;
    SR.BL =1;
    PC = VBR + H'00000100;
}

```

(10) 槽非法指令异常

- 异常源:

- 对延迟槽的未定义指令进行解码。

延迟转移指令: JMP、JSR、BRA、BRAf、BSR、BSRF、RTS、RTE、BT/S、BF/S

未定义指令: H'FFFD

- 对改写延迟槽内 PC 的指令进行解码。

改写 PC 的指令: JMP、JSR、BRA、BRAf、BSR、BSRF、RTS、RTE、BT、BF、BT/S、BF/S、TRAPA、LDC Rm,SR、LDC.L @Rm+,SR、ICBI、PREFI

- 在用户模式中对延迟槽内的特权指令进行解码。

特权指令: LDC、STC、RTE、LDTLB、SLEEP

但是, 通过 LDC 和 STC 存取 GBR 的指令除外。

- 对延迟槽内的 PC 相对 MOV 指令和 MOVA 指令进行解码

- EXPMASK 寄存器的 BRDSSLP 位为 0 并且执行延迟槽的 SLEEP 指令。

- EXPMASK 寄存器的 RTEDS 位为 0 并且执行延迟槽的非 NOP 指令。

- 转移目标地址: VBR+H'0000 0100

- 转移时的运行:

将发生此异常之前的延迟转移指令的 PC 保存到 SPC, 发生此异常时的 SR 和 R15 保存到 SSR 和 SGR。

将异常码 H'1A0 设定到 EXPEVT。将 SR 的 BL 位、MD 位和 RB 位置 1, 转移到 PC=VBR+H'0100。如果对 H'FFFD 以外的未定义指令进行解码, 就不保证运行。

```

Slot_illegal_instruction_exception()
{
    SPC = PC -2;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'000001A0;
    SR.MD =1;
    SR.RB =1;
    SR.BL =1;
    PC = VBR + H'00000100;
}

```

(11) 一般 FPU 禁止异常

- 异常源：在 SR.FD=1 时对延迟槽以外的 FPU 指令*进行解码。
- 转移目标地址：VBR+H'0000 0100
- 转移时的运行：

将发生此异常的指令的 PC 和 SR 分别保存到 SPC 和 SSR，此时的 R15 保存到 SGR。

将异常码 H'800 设定到 EXPEVT。将 SR 的 BL 位、MD 位和 RB 位置 1，转移到 PC=VBR+H'0100。

```
General_fpu_disable_exception()
{
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'00000800;
    SR.MD =1;
    SR.RB =1;
    SR.BL =1;
    PC = VBR + H'00000100;
}
```

【注】 * FPU 指令是指指令码的前 4 位为 F 的指令（未定义指令 H'FFFD 除外）和对于 FPUL、FPSCR 的 LDS、STS、LDS.L、STS.L 指令。

(12) 槽 FPU 禁止异常

- 异常源：在 SR.FD=1 时对延迟槽的 FPU 指令进行解码。
- 转移目标地址：VBR+H'0000 0100
- 转移时的运行：

将发生此异常之前的延迟转移指令的 PC 保存到 SPC，发生此异常时的 SR 和 R15 保存到 SSR 和 SGR。

将异常码 H'820 设定到 EXPEVT。将 SR 的 BL 位、MD 位和 RB 位置 1，转移到 PC=VBR+H'0100。

```
Slot_fpu_disable_exception()
{
    SPC = PC -2;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'00000820;
    SR.MD =1;
    SR.RB =1;
    SR.BL =1;
    PC = VBR + H'00000100;
}
```


(13) 指令执行前的用户断点 / 指令执行后的用户断点

- 异常源：用户断点控制器设定的中断条件成立
- 转移目标地址：VBR+H'0000 0100 或者 DBR
- 转移时的运行：

在执行指令后暂停时，将设定断点指令之后的指令PC保存到SPC；在执行指令前暂停时，将设定断点的指令的PC保存到SPC。

将暂停时的SR和R15保存到SSR和SGR。将异常码H'1E0设定到EXPEVT。

将SR的BL位、MD位和RB位置1，转移到PC=VBR+H'0100，但是也能转移到PC=DBR。

有关设定数据断点时的PC等详细内容，请参照“第30章 用户断点控制器（UBC）”。

```

User_break_exception()
{
    SPC =(pre_execution break? PC : PC +2);
    SSR = SR;
    SGR = R15;
    EXPEVT = H'000001E0;
    SR.MD =1;
    SR.RB =1;
    SR.BL =1;
    PC =(CBCR.UBDE==1? DBR : VBR + H'00000100);
}

```

(14) FPU 异常

- 异常源：执行浮点运算的异常
- 转移目标地址：VBR+H'0000 0100
- 转移时的运行：

将发生此异常的指令的PC和SR分别保存到SPC和SSR，此时的R15保存到SGR。将异常码H'120设定到EXPEVT。将SR的BL位、MD位和RB位置1，转移到PC=VBR+H'0100。

```

FPU_exception()
{
    SPC = PC;
    SSR = SR;
    SGR = R15;
    EXPEVT = H'00000120;
    SR.MD =1;
    SR.RB =1;
    SR.BL =1;
    PC = VBR + H'00000100;
}

```

5.6.3 中断

(1) NMI（非屏蔽中断）

- 中断源：检测到NMI引脚的边沿。
- 转移目标地址：VBR+H'0000 0600
- 转移时的运行：

将接受此中断的指令之后的PC和SR分别保存到SPC和SSR，此时的R15保存到SGR。

将异常码H'1C0设定到INTEVT。将SR的BL位、MD位和RB位置1，转移到PC=VBR+H'0600。当SR的BL位为0时，此中断不通过SR的中断屏蔽位进行屏蔽，而被优先接受。当SR的BL位为1时，此中断能通过软件设定是被屏蔽还是被接受。当CPUOPM的INTMU位为1时，如果接受NMI中断，就将B'1111设定到SR的IMASK位。详细内容请参照“第13章 中断控制器（INTC）”。

```
NMI()
{
    SPC = PC;
    SSR = SR;
    SGR = R15;
    INTEVT = H'000001C0;
    SR.MD =1;
    SR.RB =1;
    SR.BL =1;
    If(cond)SR.IMASK = B'1111;
    PC = VBR + H'00000600;
}
```

(2) 一般中断请求

- 中断源：
SR的中断屏蔽位低于中断请求的中断级并且SR的BL为0（在指令边界处接受。）。
- 转移目标地址：VBR+H'0000 0600
- 转移时的运行：

将接受此中断的指令之后的PC设定到SPC，此时的SR和R15设定到SSR和SGR，各中断源的相应代码设定到INTEVT。将SR的BL位、MD位和RB位置1，转移到VBR+H'0600。当CPUOPM的INTMU位为1时，如果接受一般中断，就将已接受的中断级设定到SR的IMASK位。详细内容请参照“第13章 中断控制器（INTC）”。

```
Module_interruption()
{
    SPC = PC;
    SSR = SR;
    SGR = R15;
    INTEVT = H'00000400 ~ H'00003FE0;
    SR.MD =1;
    SR.RB =1;
    SR.BL =1;
    if(cond)SR.IMASK = level_of_accepted_interrupt();
    PC = VBR + H'00000600;
}
```

5.6.4 发生多次异常时的优先顺序

2 次存取存储器的指令、必须成对执行的延迟转移指令和延迟槽指令等会发生多次异常。此时，由于和通常的异常优先顺序不同，必须注意。

(1) 2 次存取存储器的指令

由于 MAC 指令、存储器 - 存储器之间的逻辑运算指令、TAS 指令和 MOVUA 指令在 1 条指令中有 2 次数据传送，所以在各自的数据传送时检测到异常的发生。此时，需要按以下顺序来判断：

1. 第 1 次数据传送的数据地址错误
2. 第 1 次数据传送的 TLB 未命中
3. 第 1 次数据传送的 TLB 保护违反
4. 第 1 次数据传送的初始页写异常
5. 第 2 次数据传送的数据地址错误
6. 第 2 次数据传送的 TLB 未命中
7. 第 2 次数据传送的 TLB 保护违反
8. 第 2 次数据传送的初始页写异常

(2) 必须成对执行的延迟转移指令和延迟槽指令

由于延迟转移指令和延迟槽指令不可分开，所以将它们作为 1 条指令处理。因此，有关各指令的异常处理的优先顺序和通常不同。以下表示延迟槽指令只进行 1 次数据传送时的优先顺序。

1. 检测延迟转移指令的优先级 1、2 的中断型和再执行型的异常。
2. 检测延迟槽指令的优先级 1、2 的中断型和再执行型的异常。
3. 检测延迟转移指令的优先级 2 的完成型异常。
4. 检测延迟槽指令的优先级 2 的完成型异常。
5. 检测延迟转移指令的优先级 3 和延迟槽指令的优先级 3（这 2 个优先级之间没有优先顺序）。
6. 检测延迟转移指令的优先级 4 和延迟槽指令的优先级 4（这 2 个优先级之间没有优先顺序）。

当延迟槽指令有第 2 次数据传送时，在 2 中，按照（1）进行 2 次检测。

当接受的异常（优先级最高的异常）为延迟槽指令的再执行型异常时，就不禁止转移指令的 PR 寄存器写操作（BSR、BSRF、JSR 的 PC→PR）。但是，不保证此时的 PR 寄存器内容。

5.7 注意事项

(1) 异常处理的返回

1. 必须通过软件检测SR的BL位。如果将SPC和SSR保存到存储器，就必须先将SR的BL位置1，然后恢复SPC和SSR。
2. 必须发行RTE指令。通过RTE指令将SPC设定到PC，SSR设定到SR，转移到SPC的地址，然后从异常处理返回。

(2) 当 SR.BL=1 时发生一般异常或者中断的情况

1. 一般异常
如果发生除用户断点以外的一般异常，就将发生异常的指令的PC设定到SPC，进行上电复位。此时，EXPEVT为H'0000 0020，SSR为不定值。
2. 中断
如果发生一般中断，就保留中断请求，在通过软件将SR的BL位清0后接受中断。如果发生非屏蔽中断(NMI)，能通过软件设定是保留还是接受。
但是，在睡眠或者待机状态下，即使SR的BL位为1也接受中断。

(3) 发生异常时的 SPC

1. 再执行型的一般异常
将发生异常的指令PC设定到SPC，从异常处理返回后重新执行。但是，如果因延迟槽指令而发生异常，与之前的延迟转移指令的条件是否成立无关，将延迟转移指令的PC设定到SPC。
2. 完成型的异常和中断
将发生异常的指令的下一条指令的PC设定到SPC。但是，如果因延迟槽转移的指令而发生异常，就将转移目标的PC设定到SPC。

(4) RTE 指令的延迟槽

1. 在将保存在SSR的值恢复到SR后，执行被配置在RTE指令的延迟槽中的指令。有关指令存取的异常接受判断取决于恢复前的SR值，其他的异常接受判断取决于恢复后的SR的处理模式或者BL位。对于完成型的异常，在执行RTE的转移目标前被接受，但是如果发生再执行型的异常，就不保证运行。
2. 对于配置在RTE指令的延迟槽中的指令，不接受用户断点。

(5) SR 寄存器值的变更和异常的接受

1. 如果通过LDC指令操作SR寄存器的MD位和BL位，就根据新的SR寄存器值从下一条指令开始重新判断异常的接受*。对于完成型的异常，在执行下一条指令后被接受，但是对于完成型异常中的中断，在执行下一条指令前被接受。

【注】 * 当执行SR的LDC指令时，再次取后续指令，根据新的SR值进行取指令异常的重新评价。

第 6 章 浮点单元 (FPU)

6.1 概要

FPU 有以下特点:

- 符合 IEEE754 规格
- 32 个单精度浮点寄存器 (也能作为 16 个双精度寄存器被参照)
- 2 种舍入模式: 接近方向舍入和 0 方向舍入?
- 2 种非规格化数处理模式: 清 0 和非规格化数的处理
- 6 个异常源:
FPU 错误、无效运算、被 0 除、上溢、下溢、不精确
- 指令:
单精度、双精度、图形支持、系统控制
- SH-4A 是在 SH-4 上追加了以下 3 条指令:
FSRRA、FSCA、FPCHG

如果将 SR 的 FD 位置 1, 就不能使用浮点单元 (FPU)。此时, 如果执行 FPU 指令, 就会发生 FPU 禁止异常 (一般 FPU 禁止异常或者槽 FPU 禁止异常)。

6.2 数据格式

6.2.1 浮点数格式

浮点数由以下 3 个字段构成:

- 符号位 (s)
- 指数字段 (e)
- 尾数字段 (f)

SH-4A 能用如图 6.1 和图 6.2 所示的格式处理单精度和双精度浮点数。

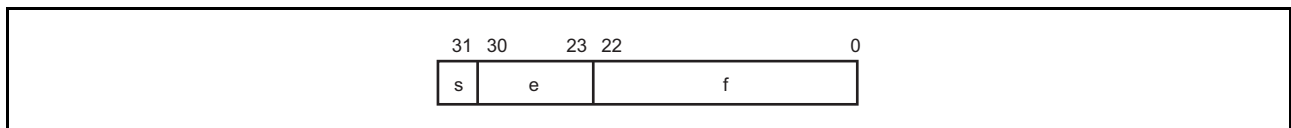


图 6.1 单精度浮点数格式

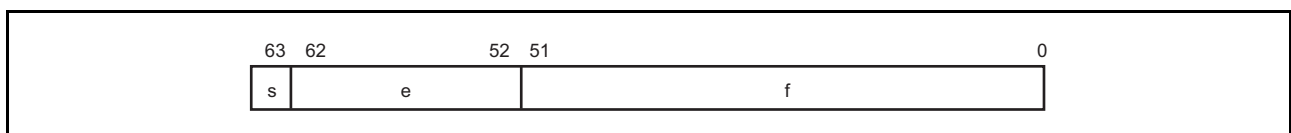


图 6.2 双精度浮点数格式

指数用带偏移量表示如下:

$$e = E + \text{偏移量}$$

无偏移量指数 E 的范围是 $E_{\min}-1 \sim E_{\max}+1$ 。 $E_{\min}-1$ 和 $E_{\max}+1$ 两个值区别如下: $E_{\min}-1$ 表示 0 (正、负符号) 和非规格化数; $E_{\max}+1$ 表示正 / 负的无限大或者非数 (NaN)。浮点数的格式和参数如表 6.1 所示。

表 6.1 浮点数的格式和参数

参数	单精度	双精度
总位数	32 位	64 位
符号位 (s)	1 位	1 位
指数字段 (e)	8 位	11 位
尾数字段 (f)	23 位	52 位
精度	24 位	53 位
偏移量	+127	+1023
E_{\max}	+127	+1023
E_{\min}	-126	-1022

如下决定浮点数值 v ：

- 当 $E=E_{\max}+1$ 并且 $f \neq 0$ 时， v 与符号 s 无关，为非数 (NaN)。
- 当 $E=E_{\max}+1$ 并且 $f=0$ 时， v 为 $(-1)^s$ (无限大) “正或者负的无限大”。
- 当 $E_{\min} \leq E \leq E_{\max}$ 时， v 为 $(-1)^s 2^E(1.f)$ “规格化数”。
- 当 $E=E_{\min}-1$ 并且 $f \neq 0$ 时， v 为 $(-1)^s 2^{E_{\min}}(0.f)$ “非规格化数”。
- 当 $E=E_{\min}-1$ 并且 $f=0$ 时， v 为 $(-1)^s 0$ “正或者负的 0”。

16 进制数各类型的范围如表 6.2 所示。信令非数和静态非数请参照“6.2.2 非数 (NaN)”，非规格化数请参照“6.2.3 非规格化数”。

表 6.2 浮点数的范围

参数	单精度	双精度
信令非数	H'7FFFFFFF ~ H'7FC00000	H'7FFFFFFF FFFFFFFF ~ H'7FF80000 00000000
静态非数	H'7FBFFFFFF ~ H'7F800001	H'7FF7FFFF FFFFFFFF ~ H'7FF00000 00000001
正的无限大	H'7F800000	H'7FF00000 00000000
正的规格化数	H'7F7FFFFFF ~ H'00800000	H'7FEFFFFFF FFFFFFFF ~ H'00100000 00000000
正的非规格化数	H'007FFFFFF ~ H'00000001	H'000FFFFFF FFFFFFFF ~ H'00000000 00000001
正的 0	H'00000000	H'00000000 00000000
负的 0	H'80000000	H'80000000 00000000
负的非规格化数	H'80000001 ~ H'807FFFFFF	H'80000000 00000001 ~ H'800FFFFFF FFFFFFFF
负的规格化数	H'80800000 ~ H'FF7FFFFFF	H'80100000 00000000 ~ H'FFEFFFFFF FFFFFFFF
负的无限大	H'FF800000	H'FFF00000 00000000
静态非数	H'FF800001 ~ H'FFBFFFFFF	H'FFF00000 00000001 ~ H'FFF7FFFF FFFFFFFF
信令非数	H'FFC00000 ~ H'FFFFFFF	H'FFF80000 00000000 ~ H'FFFFFFF FFFFFFFF

6.2.2 非数 (NaN)

非数 (NaN) 的位格式如图 6.3 所示。以下情况的值为 NaN。

- 符号位: Don't care
- 指数字段: 全部的位为 1
- 尾数字段: 至少 1 个位为 1

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

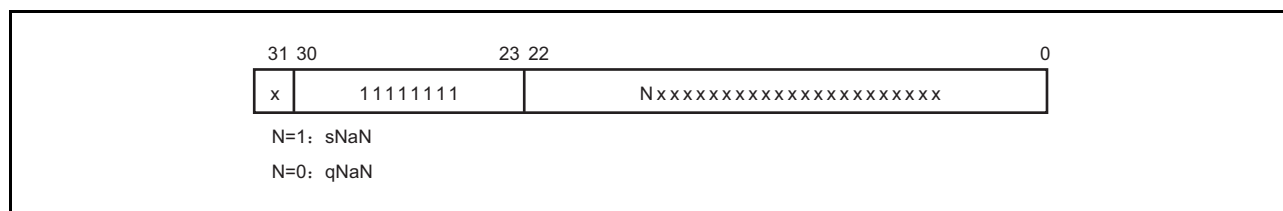


图 6.3 单精度的 NaN 位格式

假设 sNaN 为生成浮点数的运算 (寄存器和寄存器之间的传送指令、FABS 和 FNEG 除外的运算) 的输入数据,

- 当 FPSCR 的 EN.V 位为 0 时, 运算结果 (输出) 为 qNaN。
- 当 FPSCR 的 EN.V 位为 1 时, 产生无效运算异常。此时运算的目标寄存器内容不变。

寄存器和寄存器之间的传送指令有以下 3 条:

- FMOV FRm, FRn
- FLDS FRm, FPUL
- FSTS FPUL, FRn

如果在生成浮点数的运算时输入 qNaN 而不输入 sNaN, 就与 FPSCR 寄存器的 EN.V 位的设定无关, 输出总是为 qNaN。此时不产生异常。

作为运算结果, SH-4A 生成的 qNaN 值总是为以下的值:

- 单精度 qNaN: H'7FBFFFF
- 双精度 qNaN: H'7FF7FFFF FFFFFFFF

有关输入非数 (NaN) 时的浮点运算的详细内容, 请参照《SH-4A 扩展功能软件手册》的“第 11 章 各指令说明”。

6.2.3 非规格化数

非规格化数的浮点数以指数字段为 0、尾数字段不为 0 的值来表示。

当 FPU 的状态寄存器 FPSCR 的 DN 位为 1 时, 非规格化数 (源操作数或者运算结果) 通过生成 (寄存器和寄存器之间的传送指令、FNEG 和 FABS 除外的运算) 值的浮点运算, 变为正或者负的 0。

当 FPSCR 的 DN 位为 0 时, 作为非规格化数 (源操作数或者运算结果) 处理。有关输入非规格化数时的浮点运算的详细内容, 请参照《SH-4A 扩展功能软件手册》的“第 11 章 各指令说明”。

6.3 寄存器

6.3.1 浮点寄存器

浮点寄存器的结构如图 6.4 所示，有 32 个 32 位浮点寄存器。这些寄存器由 2 个存储体 FPR0_BANK0 ~ FPR15_BANK0 和 FPR0_BANK1 ~ FPR15_BANK1 构成。这 32 个寄存器作为 FR0 ~ FR15、DR0/2/4/6/8/10/12/14、FV0/4/8/12、XF0 ~ XF15、XD0/2/4/6/8/10/12/14、XMTRX 被参照。FPRn_BANKi 和参照名的对应由 FPSCR 的 FR 位决定。

(1) 浮点寄存器 FPRn_BANKi (32 个寄存器)

FPR0_BANK0 ~ FPR15_BANK0

FPR0_BANK1 ~ FPR15_BANK1

(2) 单精度浮点寄存器 FRi (16 个寄存器)

当 FPSCR.FR=0 时，FR0 ~ FR15 分配到 FPR0_BANK0 ~ FPR15_BANK0。

当 FPSCR.FR=1 时，FR0 ~ FR15 分配到 FPR0_BANK1 ~ FPR15_BANK1。

(3) 双精度浮点寄存器对或者单精度浮点寄存器对 DRi (8 个寄存器)

DR 寄存器由 2 个 FR 寄存器构成。

DR0={FR0, FR1}、DR2={FR2, FR3}、

DR4={FR4, FR5}、DR6={FR6, FR7}、

DR8={FR8, FR9}、DR10={FR10, FR11}、

DR12={FR12, FR13}、DR14={FR14, FR15}

(4) 单精度浮点向量寄存器 FVi (4 个寄存器)

FV 寄存器由 4 个 FR 寄存器构成。

FV0={FR0, FR1, FR2, FR3}、

FV4={FR4, FR5, FR6, FR7}、

FV8={FR8, FR9, FR10, FR11}、

FV12={FR12, FR13, FR14, FR15}

(5) 单精度浮点扩展寄存器 XFi (16 个寄存器)

当 FPSCR.FR=0 时，XF0 ~ XF15 分配到 FPR0_BANK1 ~ FPR15_BANK1。

当 FPSCR.FR=1 时，XF0 ~ XF15 分配到 FPR0_BANK0 ~ FPR15_BANK0。

(6) 单精度浮点扩展寄存器对 XD_i (8 个寄存器)

XD 寄存器由 2 个 XF 寄存器构成。

XD0={XF0, XF1}、XD2={XF2, XF3}、

XD4={XF4, XF5}、XD6={XF6, XF7}、

XD8={XF8, XF9}、XD10={XF10, XF11}、

XD12={XF12, XF13}、XD14={XF14, XF15}

(7) 单精度浮点扩展寄存器矩阵 XMTRX

XMTRX 由 16 个 XF 寄存器构成。

$$\text{XMTRX} = \begin{pmatrix} \text{XF0} & \text{XF4} & \text{XF8} & \text{XF12} \\ \text{XF1} & \text{XF5} & \text{XF9} & \text{XF13} \\ \text{XF2} & \text{XF6} & \text{XF10} & \text{XF14} \\ \text{XF3} & \text{XF7} & \text{XF11} & \text{XF15} \end{pmatrix}$$

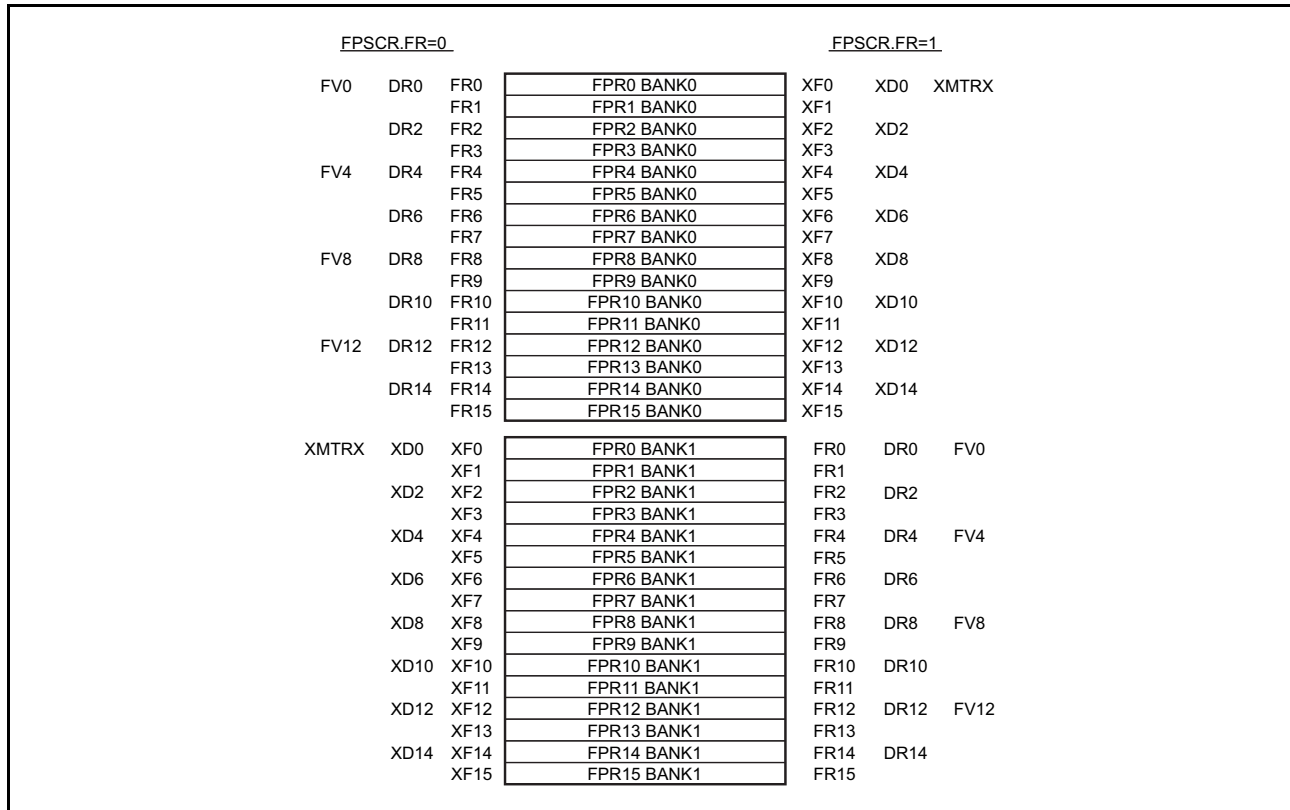


图 6.4 浮点寄存器

6.3.2 浮点状态 / 控制寄存器 (FPSCR)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	FR	SZ	PR	DN	Cause	
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	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
	Cause				Enable(EN)				Flag				RM			
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名称	初始值	R/W	说明
31 ~ 22	—	全 0	R	保留位 有关此位的读写, 请参照“产品使用上的注意事项”。
21	FR	0	R/W	浮点寄存器存储体 0: FPR0_BANK0 ~ FPR15_BANK0 分配到 FR0 ~ FR15, FPR0_BANK1 ~ FPR15_BANK1 分配到 XF0 ~ XF15。 1: FPR0_BANK0 ~ FPR15_BANK0 分配到 XF0 ~ XF15, FPR0_BANK1 ~ FPR15_BANK1 分配到 FR0 ~ FR15。
20	SZ	0	R/W	传送长度模式位 0: FMOV 指令的数据长度为 32 位。 1: FMOV 指令的数据长度为 32 位对或者 64 位。 SZ 位和 PR 位与字节排序的关系请参照图 6.5。
19	PR	0	R/W	精度模式位 0: 作为单精度运算执行浮点指令。 1: 作为双精度运算执行浮点指令 (图形支持指令未定义)。 PR 位和 SZ 位与字节排序的关系请参照图 6.5。
18	DN	1	R/W	非规格化模式位 0: 将非规格化数作为非规格化数进行处理。 1: 将非规格化数作为 0 进行处理。
17 ~ 12	Cause	000000	R/W	FPU 异常源字段
11 ~ 7	Enable(EN)	00000	R/W	FPU 异常允许字段
6 ~ 2	Flag	00000	R/W	FPU 异常标志字段 当执行 FPU 运算指令时, 最初将 FPU 异常源字段置 0。然后, 如果发生 FPU 异常时, 就将 FPU 异常源字段和 FPU 异常标志字段的相应位置 1。 最后, 清除 FPU 异常标志字段。此后, FPU 异常标志字段保持发生异常的状态。 有关各字段的位分配请参照图 6.3。
1、0	RM	01	R/W	舍入模式位 选择舍入的方法。 00: 接近方向的舍入 01: 零方向的舍入 10: 保留 (禁止设定) 11: 保留 (禁止设定)

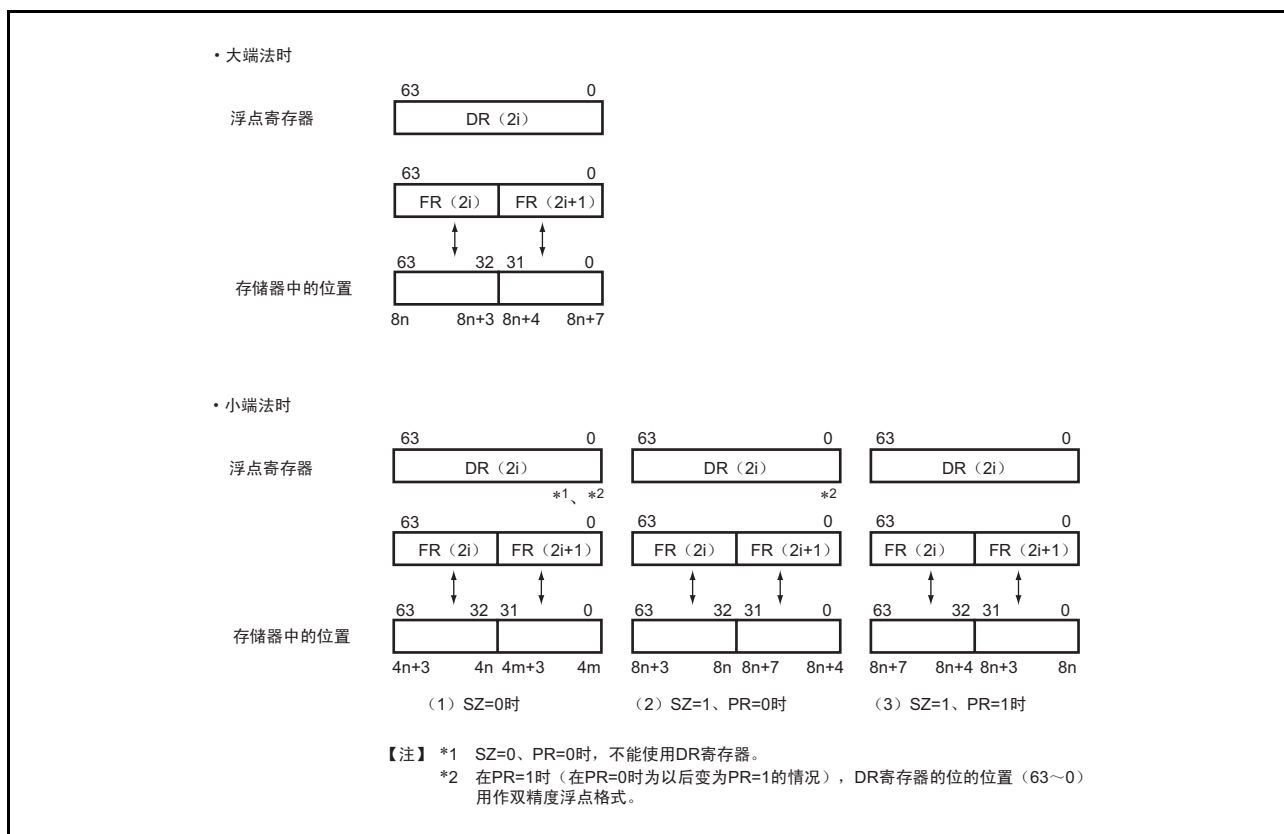


图 6.5 SZ 位和大端法的关系

表 6.3 FPU 异常处理的位分配

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

6.3.3 浮点通信寄存器 (FPUL)

通过 FPUL 寄存器进行 FPU 和 CPU 之间的信息传递。FPUL 寄存器是 32 位系统寄存器，CPU 也能通过 LDS 和 STS 指令存取。例如，将保存在通用寄存器 R1 的整数转换为单精度浮点的处理流程如下：

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

6.4 舍入

对于浮点指令，从中间结果生成最终运算结果时进行舍入。因此，FMAC、FTRV、FIPR 等组合指令的结果不同于使用 FADD、FSUB、FMUL 等基本指令的结果。这是因为 FMAC 进行 1 次舍入，而 FADD、FSUB 和 FMUL 进行 2 次舍入。

有 2 种舍入方法，由 FPSCR 的 RM 字段决定。

RM=00: 接近方向的舍入

RM=01: 零方向的舍入

(1) 接近方向的舍入

运算结果被舍入到最能接近表现的值。如果有 2 个最能接近表现的值，就选择 LSB 为 0。

如果舍入前的值大于等于 $2^{E_{\max}(2-2-p)}$ ，就为和舍入前符号相同的无穷大。在此， E_{\max} 和 p 在单精度时分别为 127 和 24；在双精度时分别为 1023 和 53。

(2) 零方向的舍入

舍去舍入位以下的位。

但是，如果舍入前的值的绝对值大于能表现的最大绝对值，就为和舍入前符号相同的能表现的最大绝对值的数。

6.5 浮点异常

有关 FPU 的异常如下所示。

(1) 一般 FPU 禁止异常 / 槽 FPU 禁止异常

当 SR.FD=1 时，如果执行 FPU 指令，就产生一般 FPU 禁止异常 / 槽 FPU 禁止异常。当 FPU 指令不在延迟槽时，产生一般 FPU 禁止异常；当 FPU 指令在延迟槽时，产生槽 FPU 禁止异常。

(2) FPU 异常

异常源如下所示：

- FPU 错误 (E):
当 FPSCR.DN=0 并且输入非规格化数时
- 无效运算 (V):
当发生如同 NaN 输入的无效运算时
- 被 0 除 (Z):
除数为 0 的除法
- 上溢 (O):
当运算结果发生上溢时
- 下溢 (U):
当运算结果发生下溢时
- 不精确异常 (I):
当发生舍入时

FPSCR 的 FPU 异常源字段中包含上述 E、V、Z、O、U、I 的所有位，FPSCR 的标志和允许字段包含 V、Z、O、U、I 的位，但是不包含 E 位。这样就不能禁止 FPU 错误。

如果发生 FPU 异常，FPU 异常源字段的相应位就置 1，FPU 异常标志字段的相应位加 1。如果不发生 FPU 异常，FPU 异常源字段的相应位就清 0，而 FPU 异常标志字段的相应位不变。

(3) FPU 异常处理

在以下情况下产生 FPU 异常：

- FPU 错误 (E)：
当 FPSCR.DN=0 并且给无法处理非规格化数的指令输入非规格化数时
- 无效运算 (V)：
当 FPSCR.EN.V=1 并且 (指令为 FTRV 或者无效运算) 时
- 被 0 除 (Z)：
当 FPSCR.EN.Z=1 并且除数为 0 的除法或者 FSRRA 的输入为 0 时
- 上溢 (O)：
当 FPSCR.EN.O=1 并且运算结果有可能发生上溢时
- 下溢 (U)：
当 FPSCR.EN.U=1 并且运算结果有可能发生下溢时
- 不精确异常 (I)：
当 FPSCR.EN.I=1 并且使用运算结果可能不精确的指令时

有关 FPU 异常产生时的详细内容，请参照《SH-4A 扩展功能软件手册》的“第 11 章 各指令说明”。

FPU 运算引起的全部异常现象作为相同的异常现象进行分配。通过软件读取系统寄存器 FPSCR 并解释所保持的信息，决定异常的含义。

另外，任何 FPU 异常处理都不改变目标寄存器的内容。

如果发生上述以外的 FPU 异常源，V、Z、O、U、I 的相应位就置 1，作为运算结果生成默认值。

- 无效运算 (V)：
作为结果，生成 qNaN。
- 被 0 除 (Z)：
生成和舍入前符号相同的无限大。
- 上溢 (O)：
当零方向舍入时，生成和舍入前符号相同的最大规格化数。
当接近方向舍入时，生成和舍入前符号相同的无穷大。
- 下溢 (U)：
当 FPSCR.DN=0 时，生成和舍入前符号相同的非规格化数或者和舍入前符号相同的 0。
当 FPSCR.DN=1 时，生成和舍入前符号相同的 0。
- 不精确异常 (I)：
生成不精确的结果。

6.6 图形支持功能

SH-4A 支持 2 种图形功能。1 种是几何运算指令，另 1 种是能进行高速数据传送的单精度数据对的传送指令。

6.6.1 几何运算指令

几何运算指令能通过最少的硬件进行高速运算，所以 SH-4A 忽视 4 个乘法的部分运算结果中的相对小的值。因此，运算结果产生以下误差：

$$\text{最大误差} = \text{MAX}(\text{各乘法结果} \times 2^{-\text{MIN}(\text{乘数的有效数字位数}-1, \text{被乘数的有效数字位数}-1)}) + \text{MAX}(\text{结果值} \times 2^{-23}, 2^{-149})$$

其中，有效数字位数的规格化数为 24，非规格化数为 23（尾数部分的前导零的位数）。

对于同一个运算，保证今后 SuperH 系列的运算误差，但是不保证不同处理器内核之间的运算误差。

(1) FIPR FVm, FVn (m, n: 0, 4, 8, 12)

此指令的用途例子如下所示：

- 内积 (m≠n)：
通常，此运算用于判断多边形表面的亮度以及判断表面和里面。
- 各要素的平方和 (m=n)：
通常，此运算用于得到向量的长度。

由于 FIPR 指令不检测不精确异常，所以当执行指令时，FPU 异常源字段和 FPU 异常标志字段的不精确异常 (I) 位总是被置 1。因此，如果 FPU 异常允许字段的 I 位被置位，就执行 FPU 异常处理。

(2) FTRV XMTRX, FVn (n: 0, 4, 8, 12)

此指令的用途例子如下所示：

- 矩阵(4×4)×向量(4)：
通常，此运算用于视点变更、角度变更或者移动向量转换（4维）。原则上，用于角度+平行移动的仿射变换处理需要4×4矩阵。因此，SH-4A支持4维运算。
- 矩阵(4×4)×矩阵(4×4)：
此运算需要执行4次FTRV指令。

由于 FIRV 指令不检测不精确异常，所以当执行指令时，FPU 异常源字段和 FPU 异常标志字段的不精确异常 (I) 位总是被置 1。因此，如果允许字段的 I 位被置位，就执行 FPU 异常处理。另外，在执行 FTRV 指令时，不能在执行前检查寄存器内的全部数据类型。如果 FPU 异常允许字段的 V 位被置位，就执行 FPU 异常处理。

(3) FRCHG

此指令更改存储体寄存器。例如，在使用 FTRV 指令时，需要在后台存储体中设定矩阵要素。但是，在建立变换矩阵的要素时，使用前台存储体寄存器的方法比较简单。当对 FPSCR 使用 LDS 指令时，为了维持 FPU 的状态，此指令需要 4～5 个周期。FRCHG 指令能用 1 个周期进行 FPSCR.FR 位的变更。

6.6.2 单精度数据对的传送

除了强力的几何运算指令以外，SH-4A 还提供高速数据传送指令。
当 FPSCR.SZ=1 时，通过单精度数据对的传送指令进行数据传送。

- FMOV DRm/XDm, DRn/XDRn (m, n: 0, 2, 4, 6, 8, 10, 12, 14)
- FMOV DRm/XDm, @Rn (m: 0, 2, 4, 6, 8, 10, 12, 14; n: 0~15)
通过这些指令，能传送 2 个单精度 (2×32 位) 数据。即，这些指令的传送性能为通常的 2 倍。
- FSCHG
此指令更改 FPSCR 的 SZ 位的值，能高速转换是否进行单精度数据对的传送。

第 7 章 存储器管理单元 (MMU)

SH-4A, 可从 8 位地址空间标识符与 32 位虚拟地址空间中进行 29 位物理地址空间的处理。使用内置于 SH-4A 的存储器管理单元 (MMU: Memory Management Unit) 进行从虚拟地址到物理地址的地址转换。MMU 通过将用户编制的地址转换表信息保存到旁路转换缓冲器 (TLB: Translation Lookaside Buffer), 快速地进行地址转换。

SH-4A 有 4 个指令 TLB (ITLB) 入口和 64 个共用 TLB (UTLB) 入口, 通过硬件将 UTLB 的副本保存到 ITLB。地址转换方式为分页方式, 在特权模式和用户模式中, 能设定虚拟地址空间的存取权, 进行存储保护。

MMU 的标志功能有 TLB 兼容模式 (4 种页大小, 4 个保护位) 和 TLB 扩展模式 (8 种页大小, 6 个保护位)。

通过控制寄存器的软件设定 (MMUCR 寄存器的 ME 位) 进行 TLB 兼容模式和 TLB 扩展模式的选择。有关 MMU 的标志功能, 分别在 TLB 兼容模式和 TLB 扩展模式中进行说明。

7.1 MMU 的概要

MMU 是为了有效利用物理存储器而设计的一种功能。如图 7.1 (0) 所示, 当进程的大小小于物理存储器时, 能将全部进程映像到物理存储器。但是, 当进程的大小增大并且物理存储器保存不下时, 需要分割进程, 将要执行的部分随时映像到物理存储器 (图 7.1 (1)), 从进程本身考虑, 这种物理存储器的映像方法会增加进程的负担。为减轻此负担, 导出了一次性地进行物理存储器映像的虚拟存储方式 (7.1 (2))。虚拟存储方式与物理存储器相比, 具有足够大的虚拟存储器, 将进程映像到此虚拟存储器, 因此只要考虑进程在虚拟存储器中的运行即可。通过 MMU 实现从虚拟存储器到物理存储器的映像。通常, OS 管理 MMU, 为了使进程所需的虚拟存储器能顺利地映像到物理存储器, 进行物理存储器的置换。通过 2 次存储等进行物理存储器的置换。

由此产生的虚拟存储方式在多个进程同时运行的分时系统 (TSS) 中发挥巨大作用 (图 7.1 (3))。由于在 TSS 中运行的多个进程需要意识各自的物理存储器的映像, 所以效率并未提高。为了提高效率, 减轻各进程的负担, 使用虚拟存储方式 (图 7.1 (4)) 给每个进程分配虚拟存储器。MMU 高效率地将多个虚拟存储器映像到物理存储器, 并且为了使某个进程不会错误地存取其他进程的物理存储器, MMU 还具有存储保护功能。

在通过 MMU 进行从虚拟存储器到物理存储器的地址转换时, 如果此转换信息未注册在 MMU, 就可能会错误地存取其他进程的虚拟存储器。此时, MMU 发生异常, 更改物理存储器的映像, 并且注册新的地址转换信息。

只通过软件也能实现 MMU 功能, 但是由于进程每次存取物理存储器都要通过软件来进行转换, 所以会降低效率。因此, 在硬件中准备了用于地址转换的缓冲器 (TLB), 频繁使用的地址转换信息预先存放在 TLB。TLB 也能称为地址转换信息的高速缓存, 但是与高速缓存不同是在地址转换失败时 (即发生异常时) 通过软件进行地址转换信息的置换, 因此通过软件能灵活地进行存储器管理。

作为进行从虚拟存储器到物理存储器的映像方式, MMU 有固定长度的地址转换方式 (分页方式) 和可变长度的地址转换方式 (分段方式)。在分页方式中, 固定大小页的地址空间为转换单位。

在以下各节中, 对于 SH-4A 将虚拟存储器中的地址空间称为虚拟地址空间, 物理存储器中的地址空间称为物理地址空间。

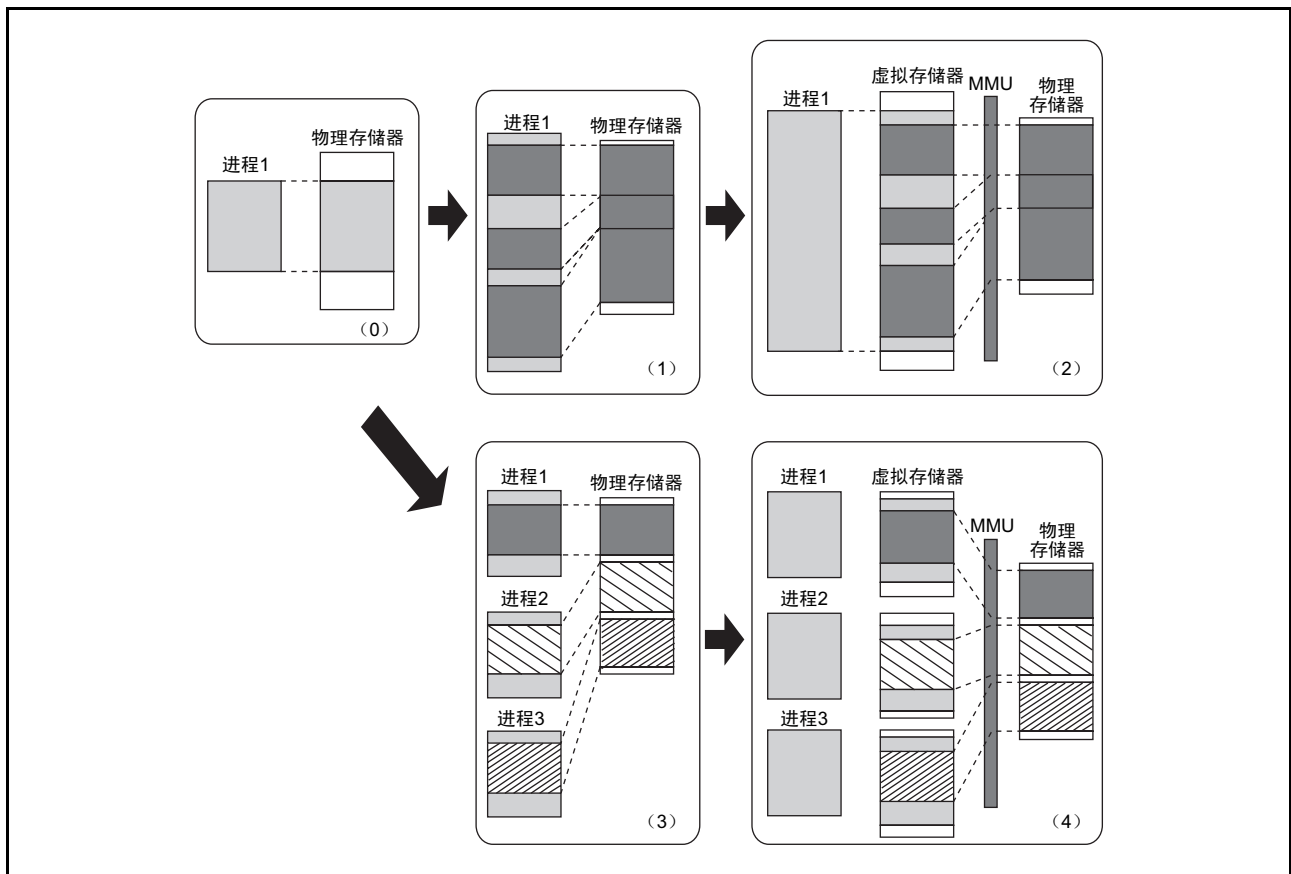


图 7.1 MMU 的作用

7.1.1 地址空间

(1) 虚拟地址空间

SH-4A 支持 32 位虚拟地址空间，能存取 4G 字节的地址空间。虚拟地址空间如图 7.2 和图 7.3 所示，分为若干区域。在特权模式中，能存取 P0 区~P4 区的 4G 字节空间；在用户模式中，能存取 U0 区的 2G 字节空间。另外，在 MMU 控制寄存器 (MMUCR) 的 SQMD 位为 0 时，能存取存储队列区的 64M 字节空间；在内部存储器控制寄存器 (RAMCR) 的 RMD 位为 1 时，能存取内部存储区的 16M 字节空间。在用户模式中，如果存取的空间不是 U0 区、存储队列区和内部存储区时，就发生地址错误。

在将 MMUCR 的 AT 位置 1 (允许 MMU) 时，这些区域中的 P0、P3 和 U0 区能通过 1K/4K/64K/1M 字节的页单位 (在 TLB 兼容模式时) 或者 1K/4K/8K/64K/256K/1M/4M/64M 字节的页单位 (在 TLB 扩展模式时) 映像到任意的物理地址空间。另外，通过使用 8 位地址空间标识符，能将 P0、P3 和 U0 区增加到 256 个。通过 TLB 实现从虚拟地址空间到 29 位物理地址空间的映像。

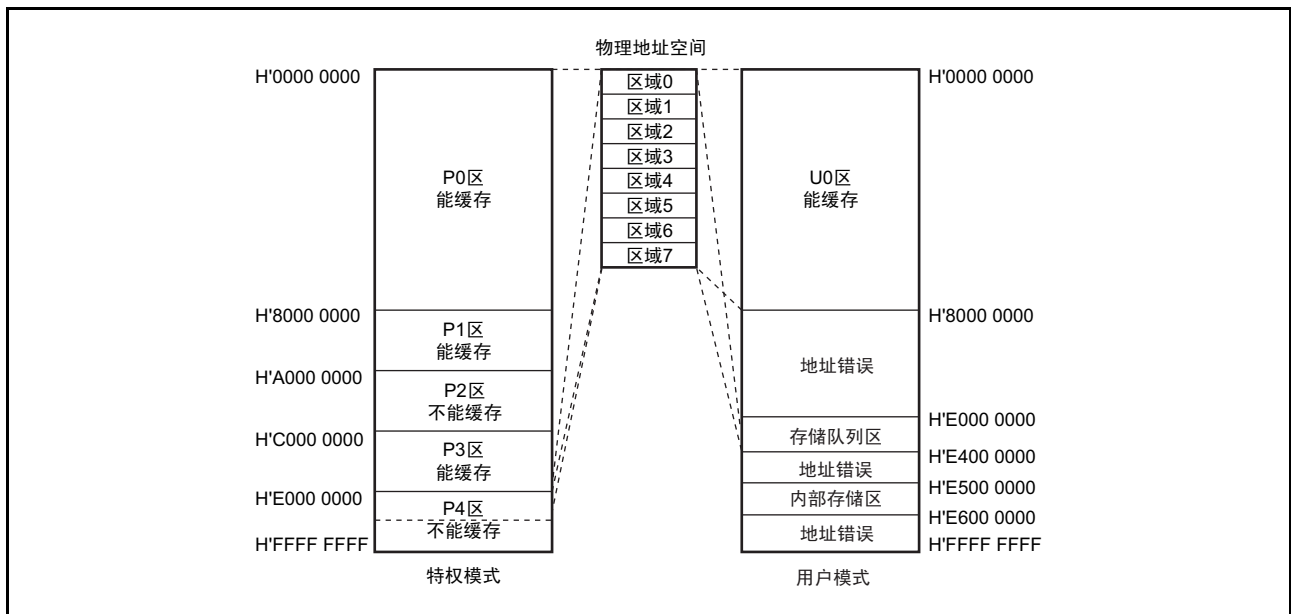


图 7.2 虚拟地址空间 (MMUCR.AT=0)

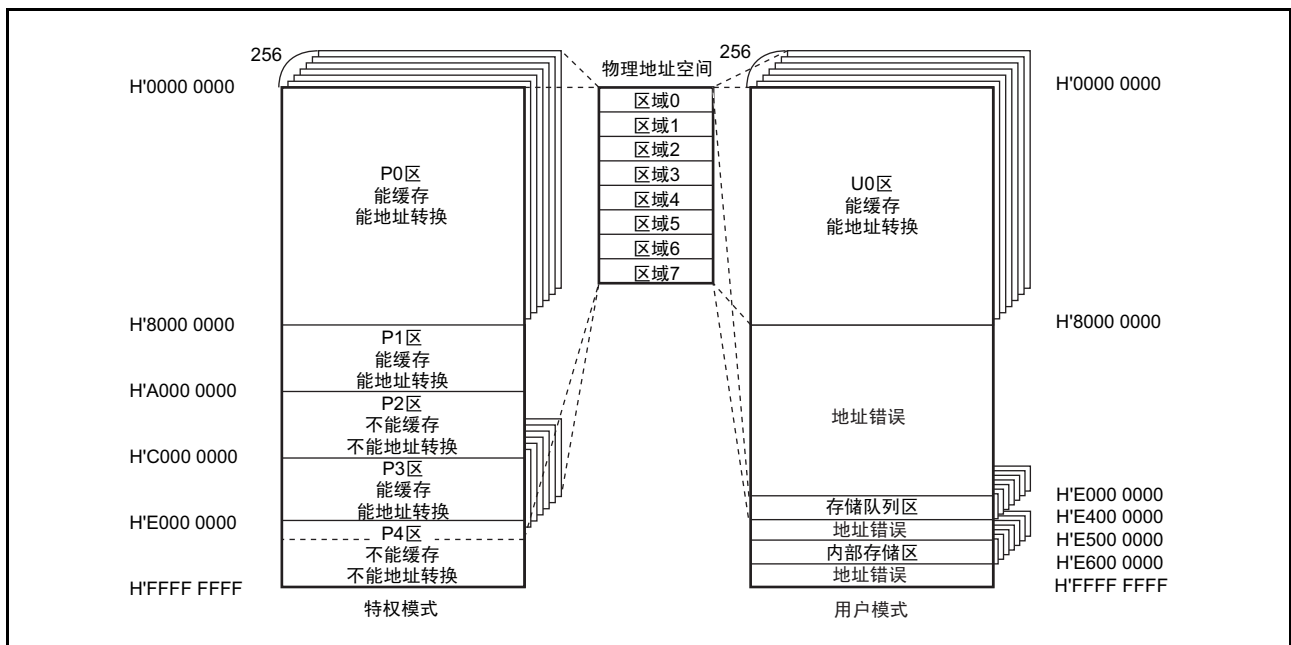


图 7.3 虚拟地址空间 (MMUCR.AT=1)

(a) P0、P3 和 U0 区

P0、P3 和 U0 区是能使用 TLB 进行地址转换以及使用高速缓存进行存取的区域。

在禁止 MMU 时，高 3 位置 0 的地址为对应的物理地址空间的地址。由高速缓存控制寄存器（CCR）选择是否使用高速缓存。在使用高速缓存时，通过 CCR 的 WT 位进行写存取的回拷模式和直写模式的转换。

在允许 MMU 时，这些区域能通过 TLB 以 1K/4K/64K/1M 字节的页单位（在 TLB 兼容模式时）或者 1K/4K/8K/64K/256K/1M/4M/64M 字节的页单位（在 TLB 扩展模式时）映像到任意的物理地址空间。在 CCR 为高速缓存允许状态并且 TLB 入口的相应页的缓存可能位（C 位）为 1 时，能使用高速缓存进行存取。在使用高速缓存时，通过 TLB 的 WT 位进行写存取的回拷模式和直写模式的转换。

在通过 TLB 将这些区域映像到物理地址空间的区域 7 中的控制寄存器区时，必须将该页的 C 位置 0。

(b) P1 区

P1 区是不能使用 TLB 进行地址转换而能使用高速缓存进行存取的区域。

无论 MMU 是否允许，高 3 位置 0 的地址为对应的物理地址空间的地址。由 CCR 选择是否使用高速缓存。在使用高速缓存时，通过 CCR 的 CB 位进行写存取的回拷模式和直写模式的转换。

(c) P2 区

P2 区是既不能使用 TLB 进行地址转换也不能使用高速缓存进行存取的区域。

无论 MMU 是否允许，高 3 位置 0 的地址为对应的物理地址空间的地址。

(d) P4 区

P4 区是映到像 SH-4A 内部资源的区域。此区域除存储队列和内部存储区外，不能使用 TLB 进行地址转换，并且也不能使用高速缓存进行存取。P4 区的详细内容如图 7.4 所示。

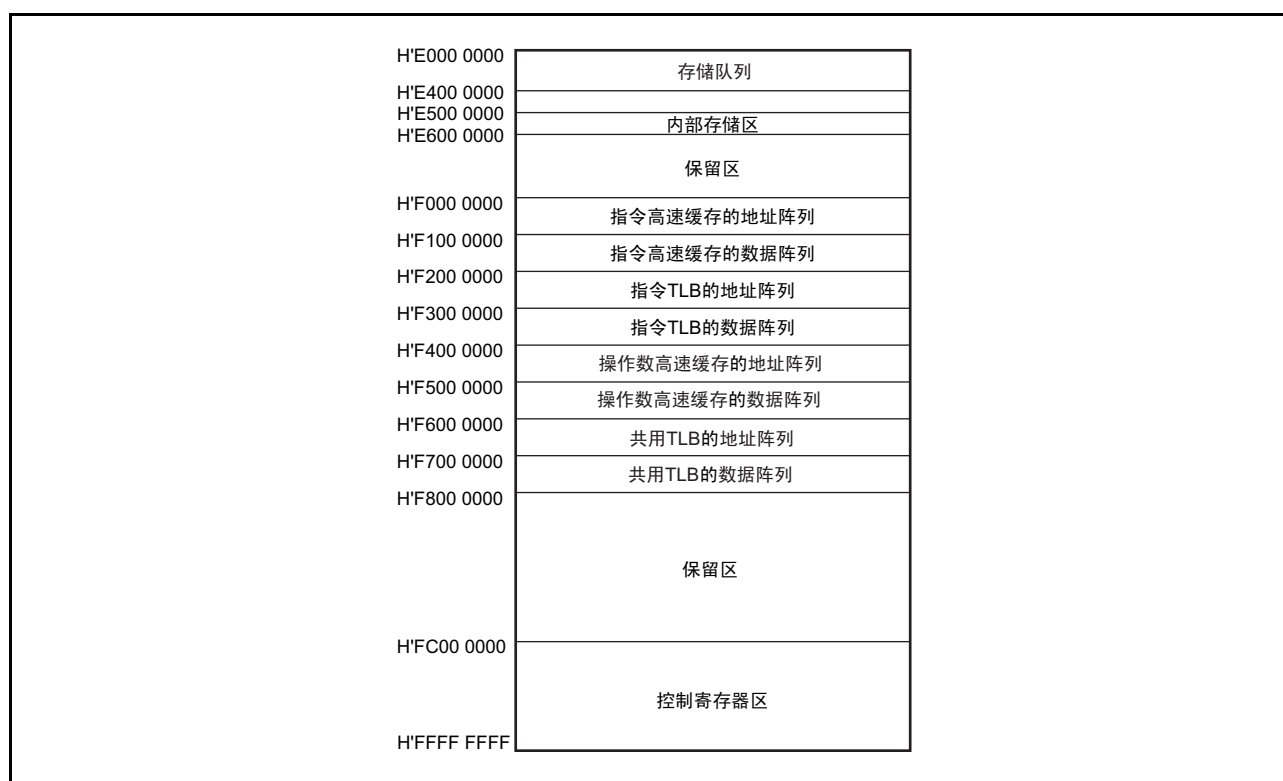


图 7.4 P4 区

H'E000 0000 ~ H'E3FF FFFF 是用于存取存储队列 (SQ) 的区域。通过 MMUCR 的 SQMD 位指定用户模式的存取权。详细内容请参照“8.7 存储队列”。

H'E500 0000 ~ H'E5FF FFFF 是用于存取内部存储器的区域。通过 RAMCR 寄存器的 RMD 位指定用户模式的存取权。详细内容请参照“第 9 章 内部存储器”。

H'F000 0000 ~ H'FOFF FFFF 是用于直接存取指令高速缓存地址阵列的区域。详细内容请参照“8.6.1 IC 地址阵列”。

H'F100 0000 ~ H'F1FF FFFF 是用于直接存取指令高速缓存数据阵列的区域。详细内容请参照“8.6.2 IC 数据阵列”。

H'F200 0000 ~ H'F2FF FFFF 是用于直接存取指令 TLB 地址阵列的区域。详细内容请参照“7.7.1 ITLB 地址阵列”。

H'F300 0000 ~ H'F37F FFFF 是用于直接存取指令 TLB 数据阵列的区域。详细内容请参照“7.7.2 ITLB 数据阵列 (TLB 兼容模式)”和“7.7.3 ITLB 数据阵列 (TLB 扩展模式)”。

H'F400 0000 ~ H'F4FF FFFF 是用于直接存取操作数高速缓存地址阵列的区域。详细内容请参照“8.6.3 OC 地址阵列”。

H'F500 0000 ~ H'F5FF FFFF 是用于直接存取操作数高速缓存数据阵列的区域。详细内容请参照“8.6.4 OC 数据阵列”。

H'F600 0000 ~ H'F60F FFFF 是用于直接存取共用 TLB 地址阵列的区域。详细内容请参照“7.7.4 UTLB 地址阵列”。

H'F700 0000 ~ H'F70F FFFF 是用于直接存取共用 TLB 数据阵列的区域。详细内容请参照“7.7.5 UTLB 数据阵列 (TLB 兼容模式)”和“7.7.6 UTLB 数据阵列 (TLB 扩展模式)”。

H'FC00 0000 ~ H'FFFF FFFF 是内部外围模块控制寄存器的区域。详细内容请参照各章的寄存器说明。

(2) 物理地址空间

SH-4A 支持 29 位物理地址空间。物理地址空间如图 7.5 所示, 分为 8 个区域, 区域 7 为保留区。详细内容请参照“第 11 章 存储器控制单元 (MCU)”。

只在使用 TLB 存取物理地址空间的区域 7 时, 区域 7 的 H'1C00 0000 ~ H'1FFF FFFF 区域不为保留区, 而与虚拟地址空间的 P4 区中的控制寄存器区等效。



图 7.5 物理地址空间

(3) 地址转换

在使用 MMU 时, 虚拟地址空间被分割为页单位, 并以页为单位转换为物理地址。外部存储器中的地址转换表保存与虚拟地址对应的物理地址和存储保护码等的附加信息, 为了高速地进行地址转换, 将外部存储器中的地址转换表的内容保存到 TLB。SH-4A 在存取指令时使用 ITLB, 在存取数据时使用 UTLB。如果存取 P4 区以外的区域, 就将被存取的虚拟地址转换为物理地址。在该虚拟地址属于 P1 或者 P2 区时, 不存取 TLB 而只决定物理地址。在该虚拟地址属于 P0、U0 或者 P3 区时, 在 TLB 中检索虚拟地址, 如果该虚拟地址注册在 TLB 中, 就为 TLB 命中, 从 TLB 读取对应的物理地址; 如果被存取的虚拟地址未注册在 TLB 中, 就产生 TLB 未命中异常, 处理转移到 TLB 未命中异常处理程序。在 TLB 未命中异常处理程序中, 检索外部存储器中的地址转换表, 并将对应的物理地址和页管理信息注册到 TLB, 在从异常处理程序返回后, 重新执行发生 TLB 未命中异常的指令。

(4) 单虚拟存储模式和多重虚拟存储模式

虚拟存储模式有单虚拟存储模式和多重虚拟存储模式，能通过 MMUCR 的 SV 位来选择。在单虚拟存储模式中，多个进程排他性地使用虚拟地址空间，并且只决定某个虚拟地址对应的物理地址。在多重虚拟存储模式中，由于多个进程共用虚拟地址空间，所以某个虚拟地址根据进程而被转换成不同的物理地址。单虚拟存储模式运行和多重虚拟存储模式运行的不同点只在 TLB 的地址比较方式（参照“7.3.3 地址转换方式”）。

(5) 地址空间标识符 (ASID)

在多重虚拟存储模式时，8 位地址空间标识符 (ASID) 用于区别共用虚拟地址空间运行的多个进程。ASID 为 8 位，能通过软件给 MMU 内的 PTEH 设定当前正在运行的进程的 ASID。另外，在通过 ASID 转换进程时不需要清除 TLB。

在单虚拟存储模式时，ASID 用于排他地使用虚拟地址空间运行的多个进程的存储保护。

【注】 在设定单虚拟存储模式时，不能将具有不同 ASID 的相同虚拟页号 (VPN) 的多个入口同时设定到 TLB。

7.2 寄存器说明

有关 MMU 处理的寄存器如下所示。

表 7.1 寄存器结构

名称	略称	R/W	P4 区地址 *	区域 7 地址 *	长度
页表入口高位寄存器	PTEH	R/W	H'FF00 0000	H'1F00 0000	32
页表入口低位寄存器	PTL	R/W	H'FF00 0004	H'1F00 0004	32
转换表基址寄存器	TTB	R/W	H'FF00 0008	H'1F00 0008	32
TLB 异常地址寄存器	TEA	R/W	H'FF00 000C	H'1F00 000C	32
MMU 控制寄存器	MMUCR	R/W	H'FF00 0010	H'1F00 0010	32
页表入口辅助寄存器	PTEA	R/W	H'FF00 0034	H'1F00 0034	32
物理地址空间控制寄存器	PASCR	R/W	H'FF00 0070	H'1F00 0070	32
重新取指令禁止控制寄存器	IRMCR	R/W	H'FF00 0078	H'1F00 0078	32

【注】 * P4 区地址是使用虚拟地址空间 P4 区的地址，区域 7 地址是使用 TLB 存取物理地址空间区域 7 的地址。

表 7.2 各处理模式的寄存器状态

名称	略称	上电复位	睡眠	待机
页表入口高位寄存器	PTEH	不定值	保持	保持
页表入口低位寄存器	PTL	不定值	保持	保持
转换表基址寄存器	TTB	不定值	保持	保持
TLB 异常地址寄存器	TEA	不定值	保持	保持
MMU 控制寄存器	MMUCR	H'0000 0000	保持	保持
页表入口辅助寄存器	PTEA	H'0000 xxx0	保持	保持
物理地址空间控制寄存器	PASCR	H'0000 0000	保持	保持
重新取指令禁止控制寄存器	IRMCR	H'0000 0000	保持	保持

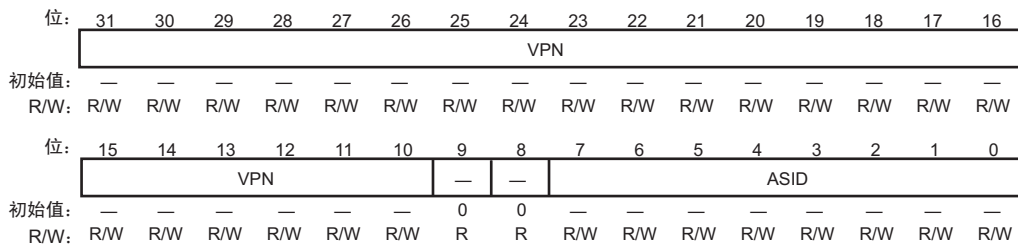
7.2.1 页表入口高位寄存器 (PTEH)

PTEH 由虚拟页号 (VPN) 和地址空间标识符 (ASID) 构成。在发生 MMU 异常或者地址错误异常时，通过硬件设定发生异常的虚拟地址的 VPN。VPN 因页大小而不同，在发生异常时由硬件设定的 VPN 为发生异常的虚拟地址的高 22 位。VPN 的设定也能通过软件进行，用软件给 ASID 设定当前正在执行的进程号。硬件不会对 ASID 进行更新。此 VPN 与 ASID 通过 LDTLB 指令注册到 UTLB。

必须在更新 PTEH 寄存器的 ASID 字段后并且在存取 (含取指令) 使用更新后的 ASID 值的 P0、P3 和 U0 区前，执行以下 1 ~ 3 的任意一项：

1. 必须通过 RTE 指令进行转移。此时，转移目标可以是 P0、P3 和 U0 区。
2. 必须对任意地址 (也可包含不能缓存的区域) 执行 ICBI 指令。
3. 如果在更新 PTEH 前预先设定 IRMCR.R2=0 (初始值)，就不需要执行特定的指令。但是，必须注意：此方法需要重新取 PTEH 更新指令的下一条指令，会降低 CPU 的处理效率。

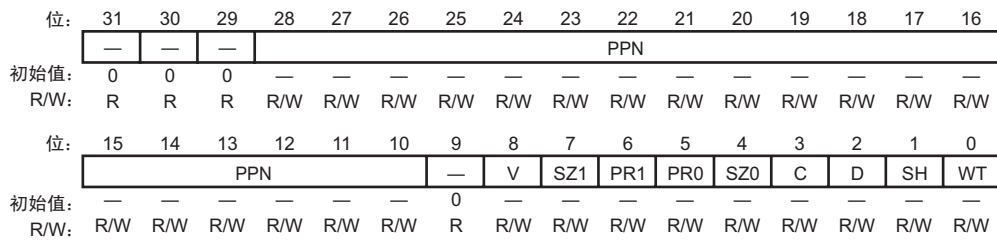
其中，方法 3 有可能不保证今后的 SuperH 系列。为了保证今后 SuperH 系列的兼容性，建议使用方法 1 或者方法 2。



位	位名	初始值	R/W	说明
31 ~ 10	VPN	不定值	R/W	虚拟页号
9、8	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。
7 ~ 0	ASID	不定值	R/W	地址空间标识符

7.2.2 页表入口低位寄存器 (PTEL)

PTEL 保存通过 LDTLB 指令注册到 UTLB 的物理页号和页管理信息。除了软件指示以外，此寄存器的内容不会变化。



位	位名	初始值	R/W	说明
31 ~ 29	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。
28 ~ 10	PPN	不定值	R/W	物理页号
9	—	0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。
8	V	不定值	R/W	页管理信息
7	SZ1	不定值	R/W	各位的含义与共用 TLB (UTLB) 对应的位相同。
6	PR1	不定值	R/W	详细内容请参照“7.3 TLB 功能 (TLB 兼容模式: MMUCR.ME=0)”
5	PR0	不定值	R/W	和“7.4 TLB 功能 (TLB 扩展模式: MMUCR.ME=1)”。
4	SZ0	不定值	R/W	【注】 SZ1、PR1、SZ0 和 PR0 只在 TLB 兼容模式时有效。
3	C	不定值	R/W	
2	D	不定值	R/W	
1	SH	不定值	R/W	
0	WT	不定值	R/W	

7.2.3 转换表基址寄存器 (TTB)

TTB 保存当前正在使用的页表基址寄存器。除了软件指示以外，TTB 的内容不会变化。能通过软件自由使用此寄存器。

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

7.2.4 TLB 异常地址寄存器 (TEA)

TEA 在发生 MMU 异常或者地址错误异常后保存发生异常的虚拟地址。能通过软件更改此寄存器。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TEA 发生MMU异常/地址错误的虚拟地址															
初始值:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEA 发生MMU异常/地址错误的虚拟地址															
初始值:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7.2.5 MMU 控制寄存器 (MMUCR)

MMUCR 的各位进行如下所示的 MMU 设定。必须通过 P1 和 P2 区的程序改写 MMUCR。

必须在更新 MMUCR 寄存器后并且在存取（含取指令）P0、P3、U0 和存储队列区前，执行以下 1～3 中的任意一项：

1. 必须通过 RTE 指令进行转移。此时，转移目标可以是 P0、P3 和 U0 区。
2. 必须对任意地址（也可包含不能缓存的区域）执行 ICBI 指令。
3. 如果在更新 MMUCR 前预先设定 IRMC.R2=0（初始值），就不需要执行特定的指令。但是，必须注意：此方法需要重新取 MMUCR 更新指令的下一条指令，会降低 CPU 的处理效率。

其中，方法 3 有可能不保证今后的 SuperH 系列。为了保证今后 SuperH 系列的兼容性，建议使用方法 1 或者方法 2。

能通过软件更改 MMUCR，能通过硬件更新 LRUI 位和 URC 位。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LRUI						—	—	URB						—	—
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/W	R/W	R/W	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	URC						SQMD	SV	ME	—	—	—	—	TI	—	AT
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R	R/W

位	位名	初始值	R/W	说明
31 ~ 26	LRUI	000000	R/W	<p>表示进行置换的 ITLB 入口的 LRU 位 为了在发生 ITLB 未命中时决定置换的 ITLB 入口, 使用 LRU 方式 (Least Recently Used)。能使用 LRUI 位确定 ITLB 的清除入口。 通过以下算法更新 LRUI。以下的“x”表示不更新。 000xxx: 在使用 ITLB 的入口 0 时 1xx00x: 在使用 ITLB 的入口 1 时 x1x1x0: 在使用 ITLB 的入口 2 时 xx1x11: 在使用 ITLB 的入口 3 时 xxxxxx: 上述以外</p> <p>另外, 在 LRUI 为以下状态时, 由 ITLB 未命中更新对应的 ITLB 入口。 不能通过软件设定以下禁止设定的值。因为在上电复位后 LRUI 被初始化为 0, 所以通过硬件不会将 LRUI 更新为禁止设定的值。以下的“x”表示 Don't care。 111xxx: 更新 ITLB 的入口 0 0xx11x: 更新 ITLB 的入口 1 x0x0x1: 更新 ITLB 的入口 2 xx0x00: 更新 ITLB 的入口 3 上述以外: 禁止设定</p>
25、24	—	全 0	R	保留位 有关此位的读写, 请参照“产品使用上的注意事项”。
23 ~ 18	URB	000000	R/W	表示进行置换的 UTLB 入口边界的位 在 URB ≠ 0 时有效。
17、16	—	全 0	R	保留位 有关此位的读写, 请参照“产品使用上的注意事项”。
15 ~ 10	URC	000000	R/W	表示通过 LDTLB 指令进行置换的 UTLB 入口的随机计数器 在每次存取 UTLB 时进行递增计数。但是, 在 URB > 0 时, 如果 URC=URB 的条件成立, URC 就被清 0。另外, 在通过软件将 URC > URB 的值写到 URC 时, 在最初的 URC=H'3F 前递增计数会超过 URB, 所以必须注意。LDTLB 指令不会使 URC 进行递增计数。
9	SQMD	0	R/W	存储队列模式位 指定存储队列的存取权。 0: 能进行用户 / 特权存取 1: 能进行特权存取 (用户存取时发生地址错误异常)
8	SV	0	R/W	单虚拟存储模式 / 多重虚拟存储模式转换位 在更改此位时, 必须将 TI 位置 1。 0: 多重虚拟存储模式 1: 单虚拟存储模式
7	ME	0	R/W	TLB 扩展模式转换位 0: TLB 兼容模式 1: TLB 扩展模式 在更改 ME 位时, 必须将 TI 位置 1, 使 ITLB 和 UTLB 的内容无效。
6 ~ 3	—	全 0	R	保留位 有关此位的读写, 请参照“产品使用上的注意事项”。
2	IT	0	R/W	TLB 无效位 如果将此位置 1, UTLB/ITLB 的有效位就全部被清 0。读取值为 0。
1	—	0	R	保留位 有关此位的读写, 请参照“产品使用上的注意事项”。

位	位名	初始值	R/W	说明
0	AT	0	R/W	地址转换有效位 指定是允许（有效）还是禁止（无效）MMU。 0：禁止 MMU 1：允许 MMU 在 AT 位为 0 的状态下不发生 MMU 异常，因此必须通过不使用 MMU 的软件将 AT 位置 0。

7.2.6 页表入口辅助寄存器 (PTEA)

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

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

位	位名	初始值	R/W	说明
31 ~ 14	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。
13 ~ 8	EPR	不定值	R/W	页管理信息 各位的含义与共用 TLB (UTLB) 对应的位相同。 详细内容请参照 “7.4 TLB 功能 (TLB 扩展模式: MMUCR.ME=1)”。
7 ~ 4	ESZ	不定值	R/W	
3 ~ 0	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。

7.2.7 物理地址空间控制寄存器 (PASCR)

PASCR 控制物理地址空间的运行。

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		
	—	—	—	—	—	—	—	—	UB								—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

位	位名	初始值	R/W	说明
31 ~ 8	—	全 0	R	保留位 有关此位的读写, 请参照“产品使用上的注意事项”。
7 ~ 0	UB	H'00	R/W	各区域 (64M 字节) 的缓冲写控制 按区域指定在不使用高速缓存的写总线存取结束前是否等待下一个 CPU 的总线存取。 0: 缓冲写 (CPU 不等待写总线存取结束而进行下一个总线存取) 1: 不进行缓冲写 (CPU 在等待写总线存取结束后进行下一个总线存取) UB[7]: 对应控制寄存器区 UB[6]: 对应区域 6 UB[5]: 对应区域 5 UB[4]: 对应区域 4 UB[3]: 对应区域 3 UB[2]: 对应区域 2 UB[1]: 对应区域 1 UB[0]: 对应区域 0

7.2.8 重新取指令禁止控制寄存器 (IRMCR)

IRMCR 控制是否在更改特定资源后重新取下一条指令。特定资源是指控制寄存器的一部分、TLB 和高速缓存。

初始状态设定为在更改资源后重新取下一条指令，但是在此状态下每更改一次资源就会重新取指令，降低 CPU 的处理效率。因此，建议将 IRMCR 的各位置 1，在一次性地进行所需资源的变更后执行特定的指令，转移到使用变更后的资源的程序。

有关特定的顺序，请参照各资源的说明。

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
	—	—	—	—	—	—	—	—	—	—	—	R2	R1	LT	MT	MC
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 5	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。
4	R2	0	R/W	更改寄存器后重新取指令的禁止 2 控制是否在更改 MMUCR、PASCR、CCR、RAMCR、PTEH 的各寄存器后重新取下一条指令。 0: 重新取指令 1: 不重新取指令
3	R1	0	R/W	更改寄存器后重新取指令的禁止 1 控制是否在更改地址 H'FF200000 ~ H'FF2FFFFFF 中的寄存器后重新取下一条指令。 0: 重新取指令 1: 不重新取指令
2	LT	0	R/W	执行 LDTLB 后重新取指令的禁止 控制是否在执行 LDTLB 指令后重新取下一条指令。 0: 重新取指令 1: 不重新取指令
1	MT	0	R/W	写存储器映像 TLB 后重新取指令的禁止 在 MMUCR.AT=1 状态下，控制是否在写存储器映像 ITLB/UTLB 后重新取下一条指令。 0: 重新取指令 1: 不重新取指令
0	MC	0	R/W	写存储器映像 IC 后重新取指令的禁止 在 CCN.ICE=1 状态下，控制是否在写存储器映像 IC 后重新取下一条指令。 0: 重新取指令 1: 不重新取指令

7.3 TLB 功能 (TLB 兼容模式: MMUCR.ME=0)

7.3.1 共用 TLB (UTLB) 的结构

UTLB 用于以下 2 个目的:

1. 在数据存取时将虚拟地址转换为物理地址。
2. 在指令 TLB 未命中时用注册到 ITLB 的地址转换信息表。

因此, 称为共用 TLB。将存放在外部存储器中的地址转换表信息保存到 UTLB。在地址转换表中保存虚拟页号和地址空间标识符以及与之对应的物理页号和页管理信息。UTLB 的结构如图 7.6 所示, 由全相联方式的 64 个入口构成。页大小与地址的关系如图 7.7 所示。

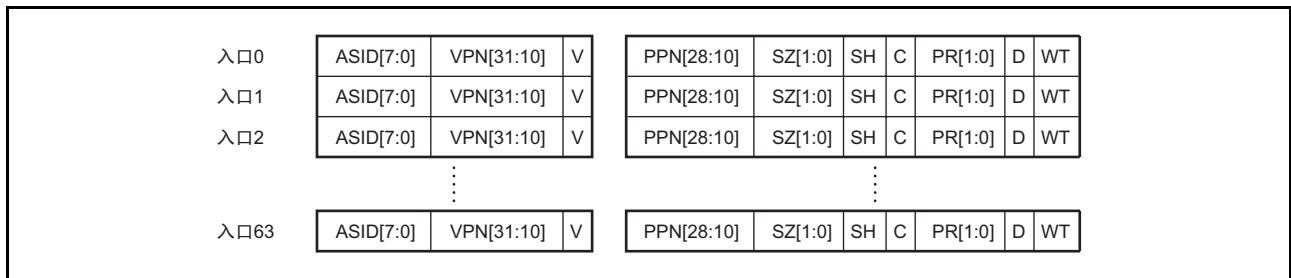


图 7.6 UTLB 的结构 (TLB 兼容模式)

【符号说明】

VPN: 虚拟页号	1K字节页时, 虚拟地址的高22位 4K字节页时, 虚拟地址的高20位 64K字节页时, 虚拟地址的高16位 1M字节页时, 虚拟地址的高12位
ASID: 地址空间标识符	表示能存取虚拟页的进程。 在单虚拟存储模式并且用户模式时, 或者在多重虚拟存储模式时, 如果SH位为0, 就在地址比较时与PTEH中的ASID进行比较。
SH: 共有状态位	0: 多个进程不共有页。 1: 多个进程共有页。
SZ[1:0]: 页大小位	指定页大小。 00: 1K字节页 01: 4K字节页 10: 64K字节页 11: 1M字节页
V: 有效位	表示入口是否有效。 0: 无效 1: 有效
PPN: 物理页号	在上电复位时被清0。 物理地址的高22位 1K字节页时, PPN[28:10]有效。 4K字节页时, PPN[28:12]有效。 64K字节页时, PPN[28:16]有效。 1M字节页时, PPN[28:20]有效。
PR[1:0]: 保护键数据	在PPN的设定中, 必须注意同义问题 (参照“7.5.5 同义问题的避免”)。 用代码表示页存取权的2位数据 00: 在特权模式中, 只能读 01: 在特权模式中, 能读写 10: 在特权/用户模式中, 只能读 11: 在特权/用户模式中, 能读写

- C: 缓存可能位 表示页是否能缓存。
0: 不能缓存。
1: 能缓存。
在进行控制寄存器空间的映像时，必须将此位置0。
- D: 已写位 表示是否已写页。
0: 未写。
1: 已写。
- WT: 直写位 指定高速缓存的写模式。
0: 回拷模式
1: 直写模式

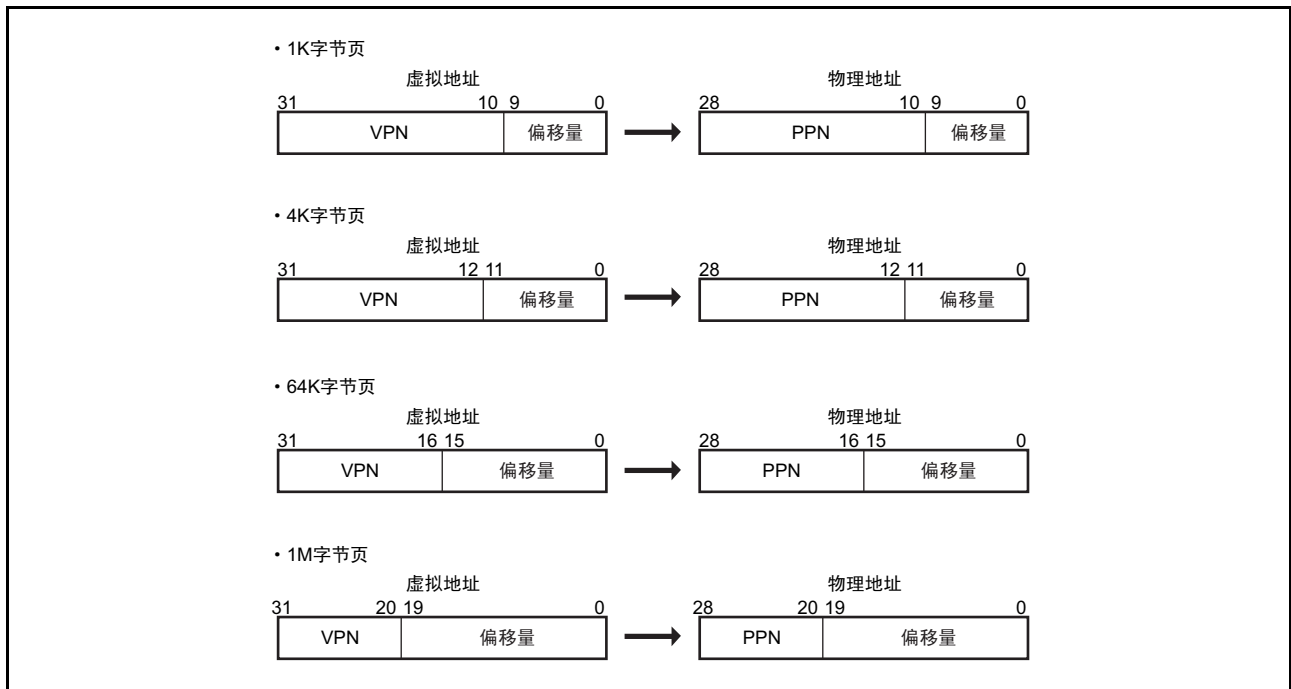


图 7.7 页大小与地址的关系 (TLB 兼容模式)

7.3.2 指令 TLB (ITLB) 的结构

ITLB 在指令存取时将虚拟地址转换为物理地址。将存放在 UTLB 中的地址转换表的信息保存到 ITLB。ITLB 的结构如图 7.8 所示，由全相联的 4 个入口构成。

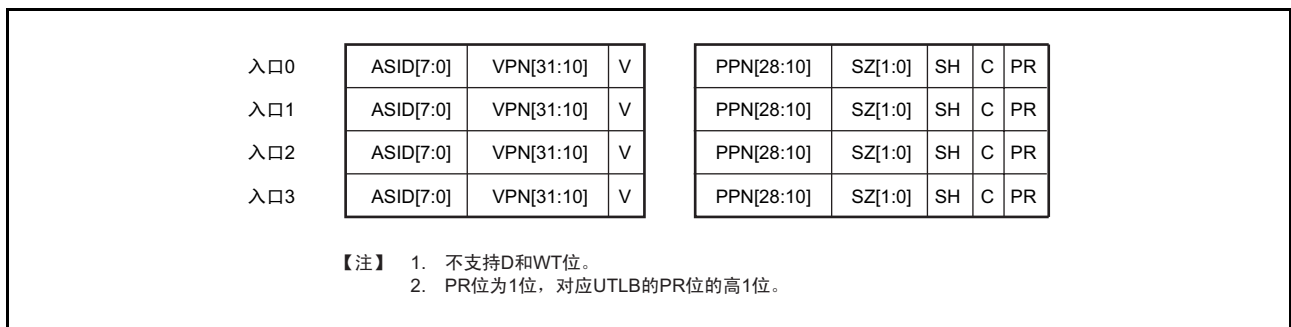


图 7.8 ITLB 的结构 (TLB 兼容模式)

7.3.3 地址转换方式

使用 UTLB 的存储器存取流程如图 7.9 所示。

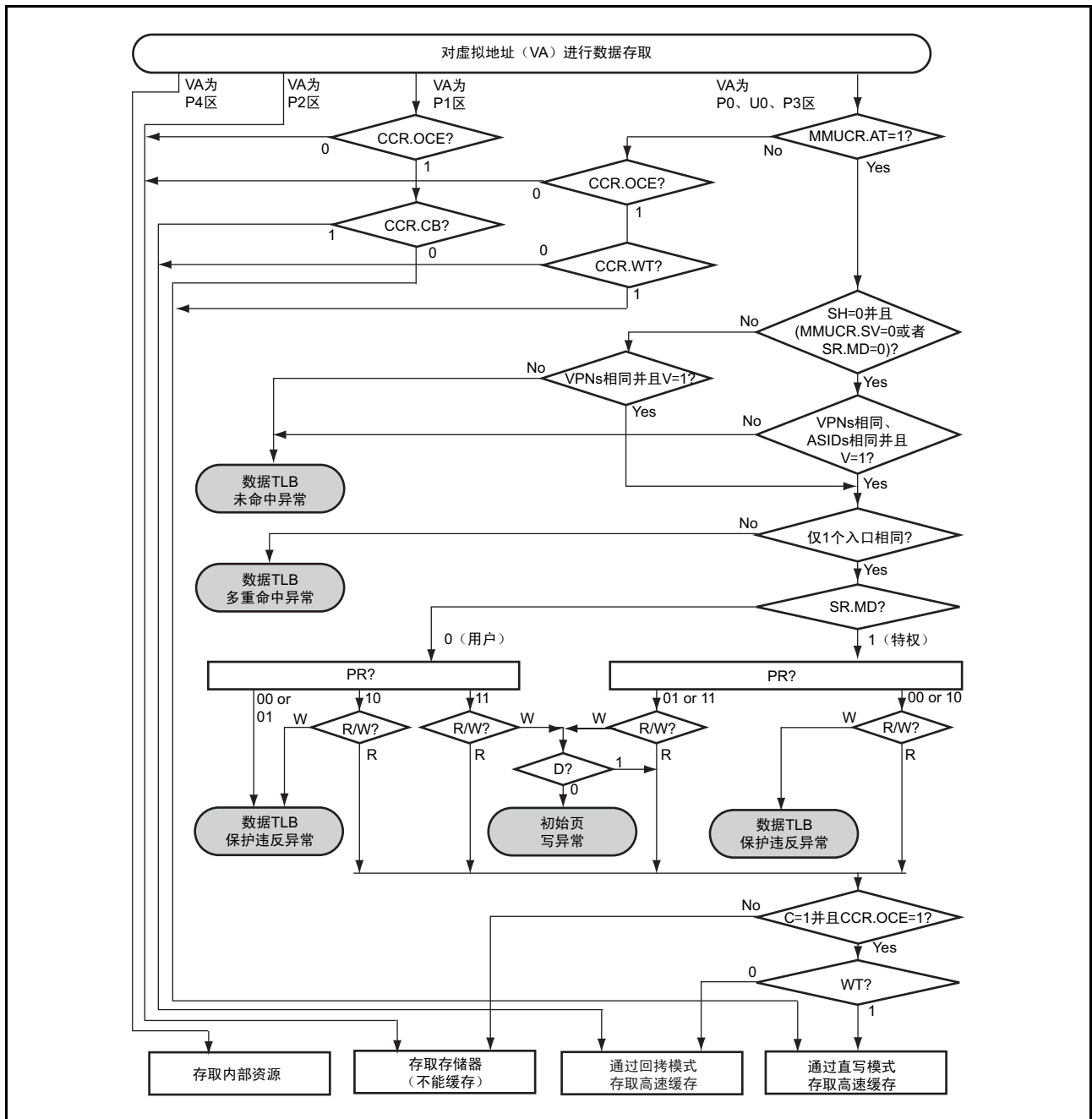


图 7.9 使用 UTLB 的存储器存取流程 (TLB 兼容模式)

使用 ITLB 的存储器存取流程如图 7.10 所示。

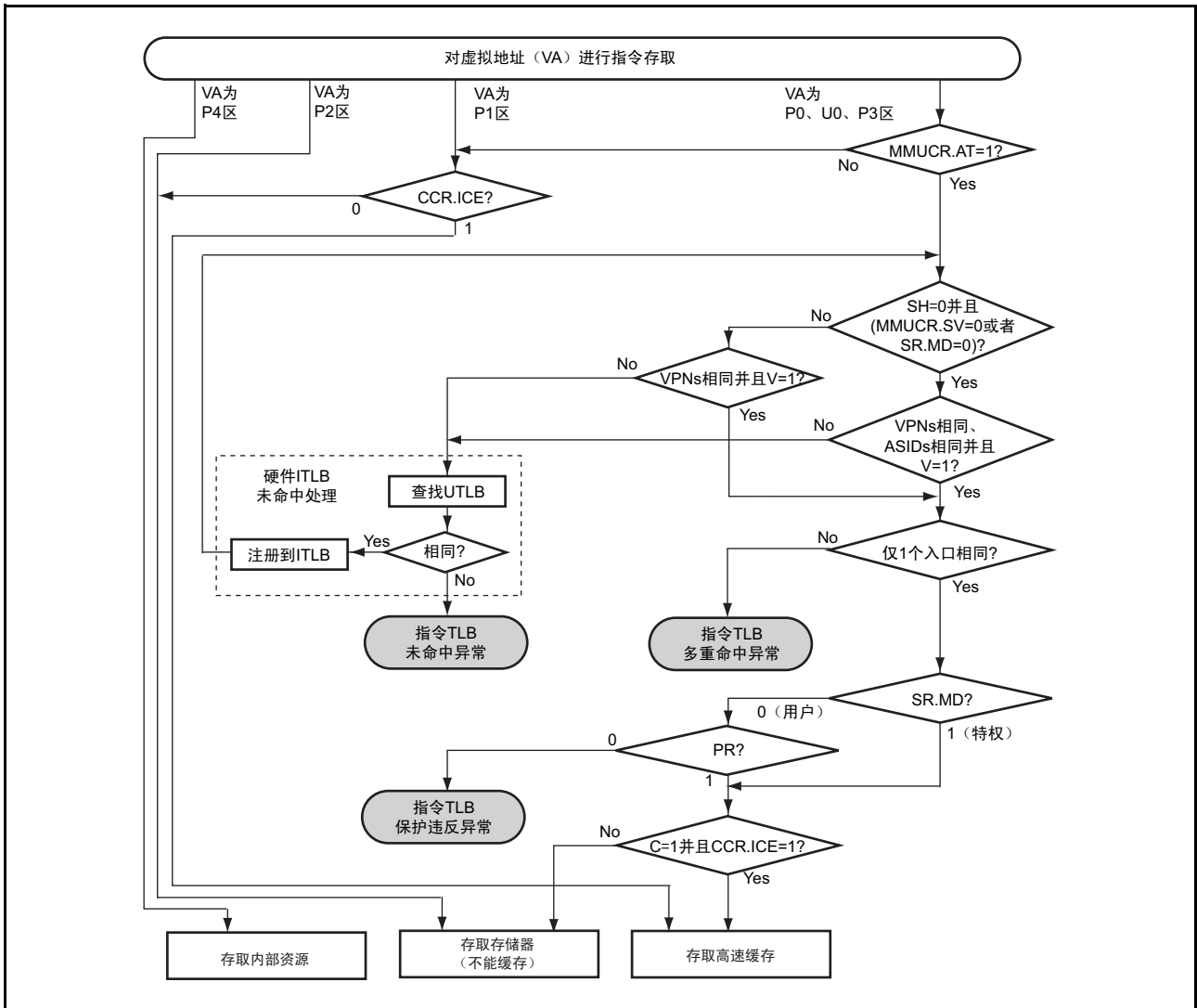


图 7.10 使用 ITLB 的存储器存取流程 (TLB 兼容模式)

7.4 TLB 功能 (TLB 扩展模式: MMUCR.ME=1)

7.4.1 共用 TLB (UTLB) 的结构

TLB 扩展模式的 UTLB 结构如图 7.11 所示, 页大小与地址的关系如图 7.12 所示。

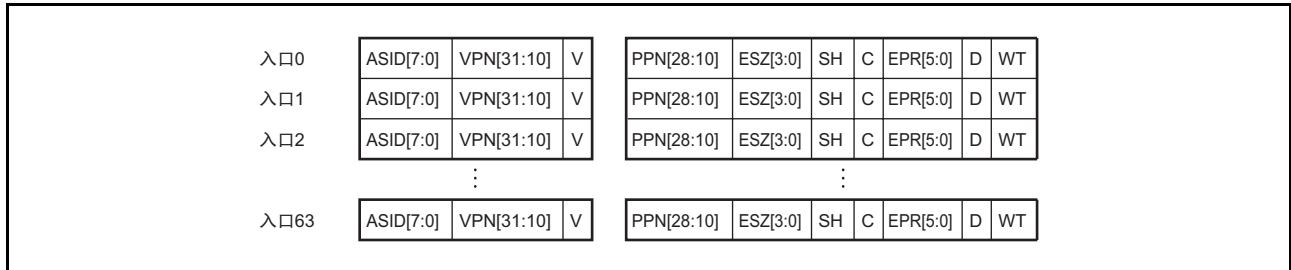


图 7.11 UTLB 的结构 (TLB 扩展模式)

【符号说明】

VPN: 虚拟页号	1K 字节页时, 虚拟地址的高 22 位 4K 字节页时, 虚拟地址的高 20 位 8K 字节页时, 虚拟地址的高 19 位 64K 字节页时, 虚拟地址的高 16 位 256K 字节页时, 虚拟地址的高 14 位 1M 字节页时, 虚拟地址的高 12 位 4M 字节页时, 虚拟地址的高 10 位 64M 字节页时, 虚拟地址的高 6 位
ASID: 地址空间标识符	表示能存取虚拟页的进程。 在单虚拟存储模式并且用户模式时, 或者在多重虚拟存储模式时, 如果 SH 位为 0, 就在地址比较时与 PTEH 中的 ASID 进行比较。
SH: 共有状态位	0: 多个进程不共有页。 1: 多个进程共有页。
ESZ: 页大小位	指定页大小。 0000: 1K 字节页 0001: 4K 字节页 0010: 8K 字节页 0100: 64K 字节页 0101: 256K 字节页 0111: 1M 字节页 1000: 4M 字节页 1100: 64M 字节页
【注】	如果注册上述以外的值, 就不保证运行。
V: 有效位	表示入口是否有效。 0: 无效 1: 有效 在上电复位时被清 0。
PPN: 物理页号	物理地址的高 19 位 1K 字节页时, PPN[28:10] 有效。 4K 字节页时, PPN[28:12] 有效。 8K 字节页时, PPN[28:13] 有效。 64K 字节页时, PPN[28:16] 有效。 256K 字节页时, PPN[28:18] 有效。 1M 字节页时, PPN[28:20] 有效。 4M 字节页时, PPN[28:22] 有效。 64M 字节页时, PPN[28:26] 有效。 在 PPN 的设定中, 必须注意同义问题 (参照 “7.5.5 同义问题的避免”)。

EPR: 保护键数据	<p>用代码表示页存取权的6位数据</p> <p>对于特权模式的读/写/执行（取指令）以及用户模式的读/写/执行（取指令），能独立设定允许。各位分别用0表示不允许，用1表示允许。</p> <p>EPR[5]: 特权模式的读</p> <p>EPR[4]: 特权模式的写</p> <p>EPR[3]: 特权模式的执行（取指令）</p> <p>EPR[2]: 用户模式的读</p> <p>EPR[1]: 用户模式的写</p> <p>EPR[0]: 用户模式的执行（取指令）</p>
C: 缓存可能位	<p>表示页是否能缓存。</p> <p>0: 不能缓存。</p> <p>1: 能缓存。</p> <p>在进行控制寄存器空间的映像时，必须将此位置0。</p>
D: 已写位	<p>表示是否写页。</p> <p>0: 未写。</p> <p>1: 已写。</p>
WT: 直写位	<p>指定高速缓存的写模式。</p> <p>0: 回拷模式</p> <p>1: 直写模式</p>

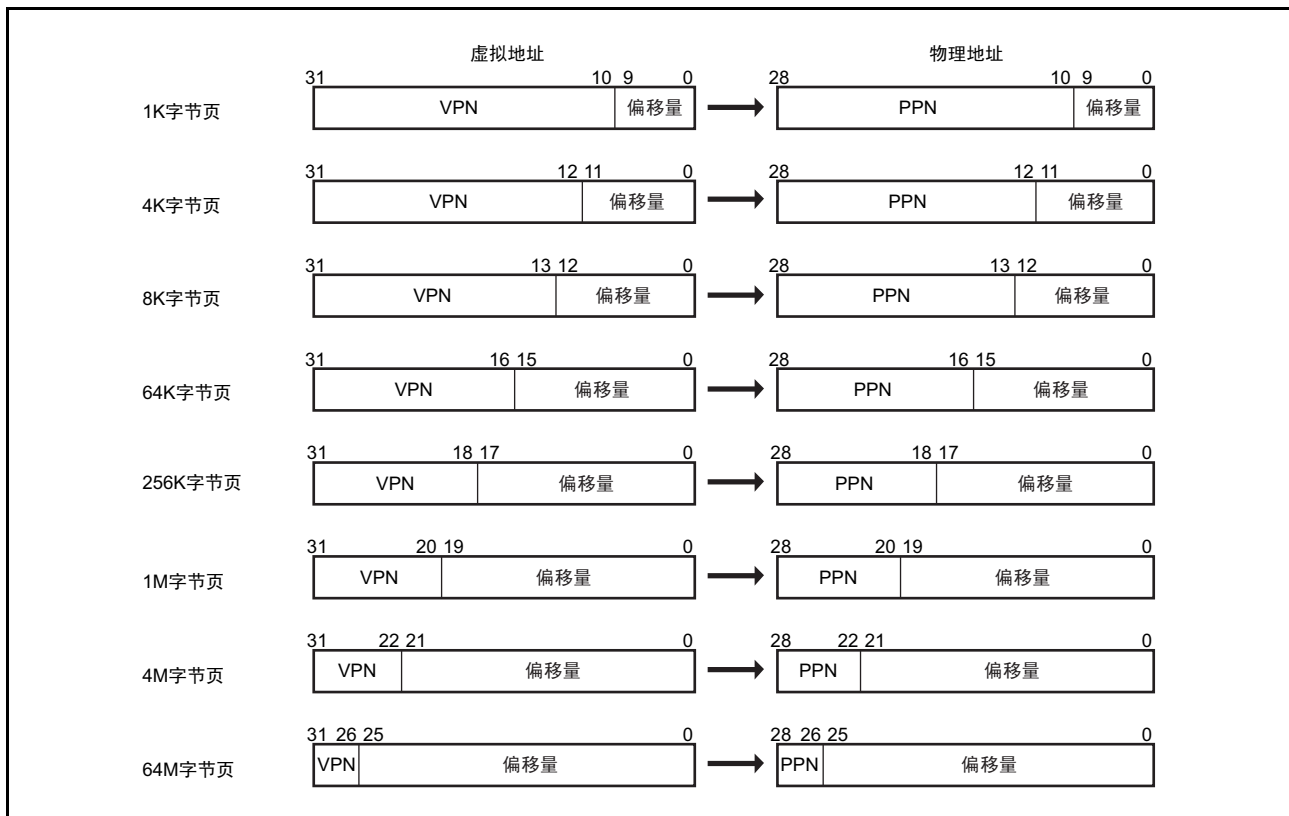


图 7.12 页大小与地址的关系 (TLB 扩展模式)

7.4.2 指令 TLB (ITLB) 的结构

TLB 扩展模式的 ITLB 结构如图 7.13 所示。

入口 0	ASID[7:0]	VPN[31:10]	V	PPN[28:10]	ESZ[3:0]	SH	C	EPR[5]	EPR[3]	EPR[2]	EPR[0]
入口 1	ASID[7:0]	VPN[31:10]	V	PPN[28:10]	ESZ[3:0]	SH	C	EPR[5]	EPR[3]	EPR[2]	EPR[0]
入口 2	ASID[7:0]	VPN[31:10]	V	PPN[28:10]	ESZ[3:0]	SH	C	EPR[5]	EPR[3]	EPR[2]	EPR[0]
入口 3	ASID[7:0]	VPN[31:10]	V	PPN[28:10]	ESZ[3:0]	SH	C	EPR[5]	EPR[3]	EPR[2]	EPR[0]

【注】 不支持EPR[4]、EPR[1]、D和WT位。

图 7.13 ITLB 的结构 (TLB 扩展模式)

7.4.3 地址转换方式

TLB 扩展模式时使用 UTLB 存储器存取流程如图 7.14 所示。

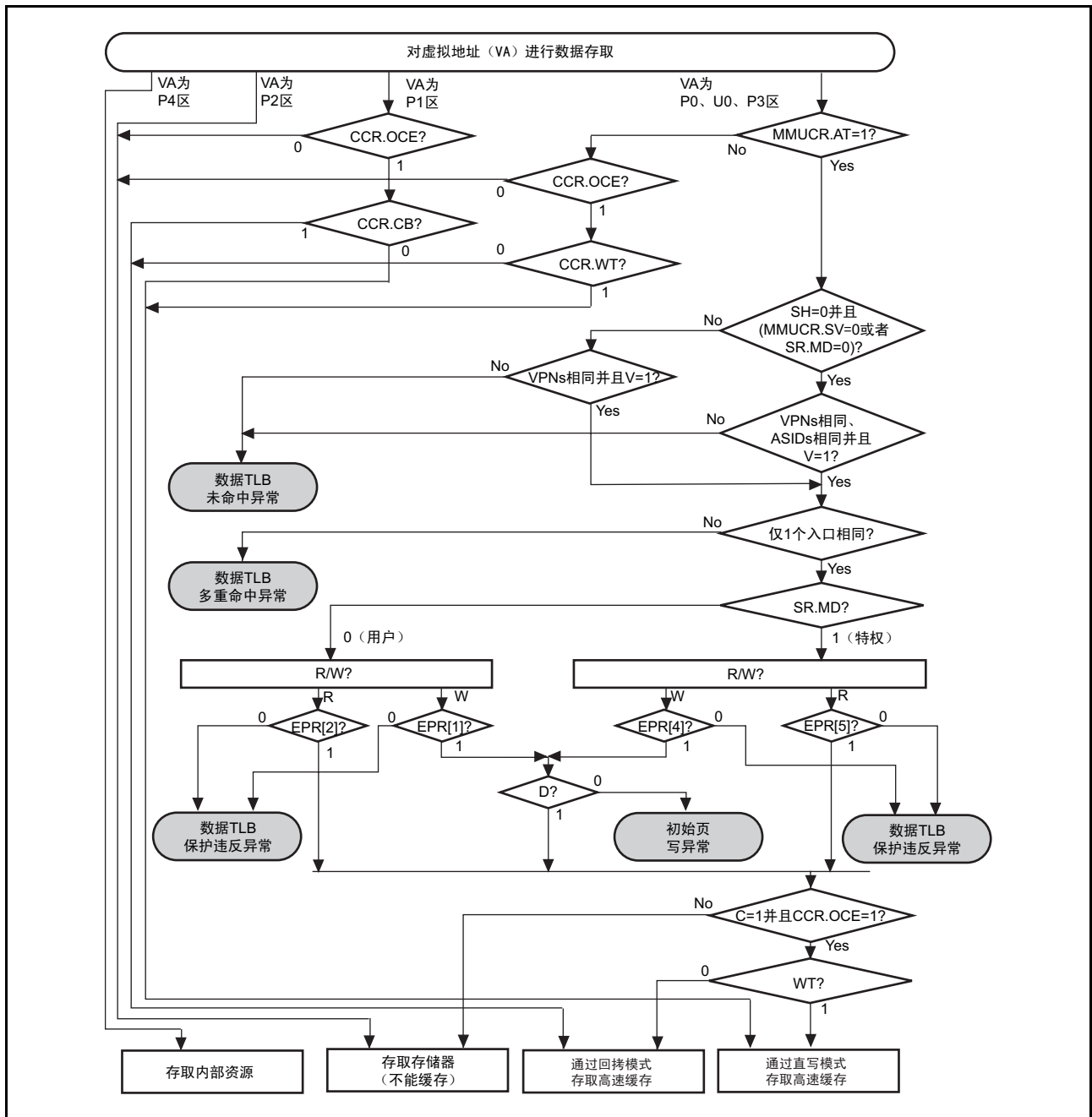


图 7.14 使用 UTLB 存储器存取的流程 (TLB 扩展模式)

TLB 扩展模式时使用 ITLB 存储器存取的流程如图 7.15 所示。

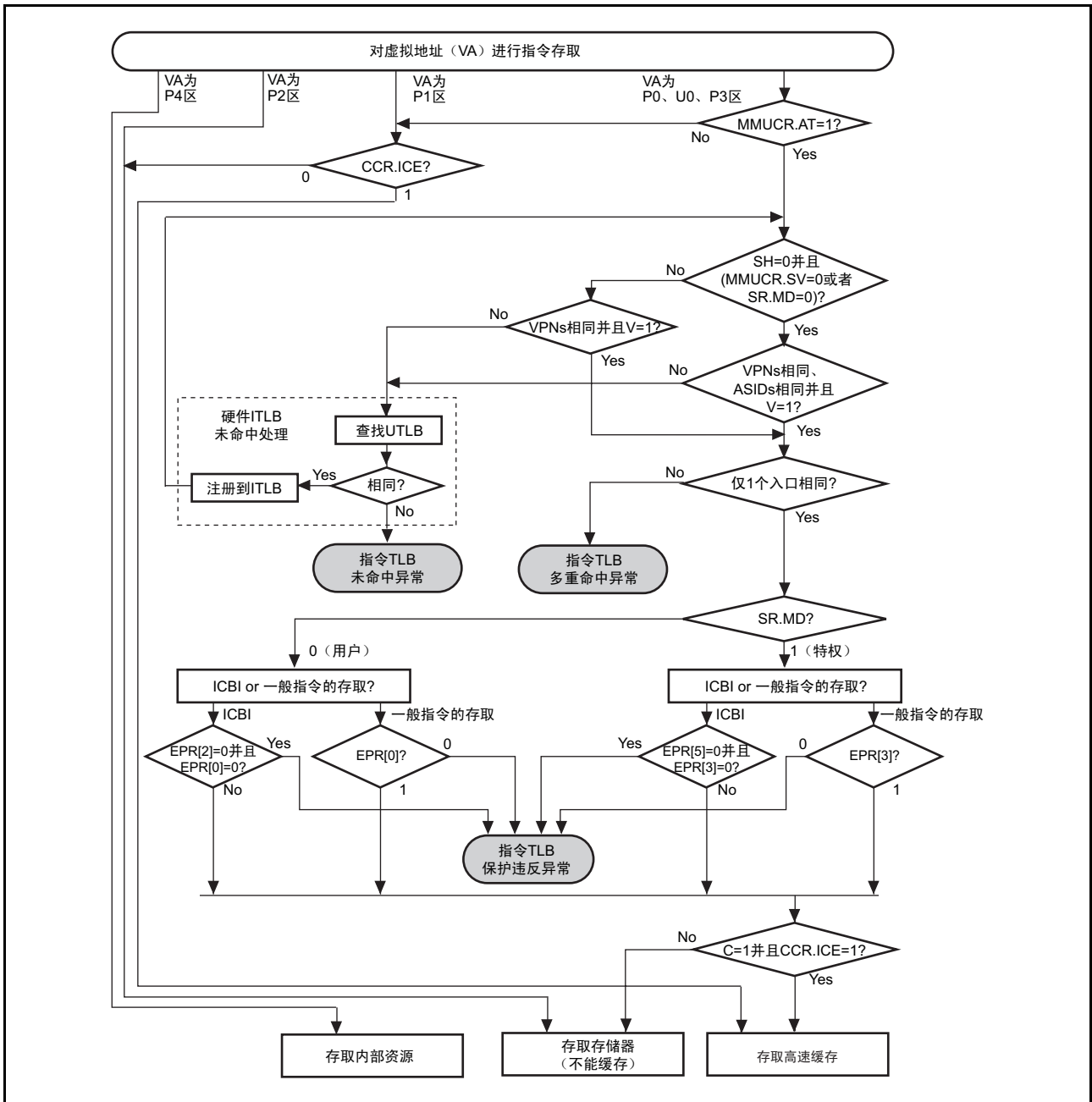


图 7.15 使用 ITLB 的存储器存取的流程 (TLB 扩展模式)

7.5 MMU 的功能

7.5.1 MMU 的硬件管理

SH-4A 支持的 MMU 功能如下：

1. 对软件存取的虚拟地址进行解码，根据 MMUCR 的设定控制 UTLB 和 ITLB，进行地址转换。
2. 以地址转换时读取的页管理信息为基础，判断高速缓存的存取状态（C 和 WT 位）。
3. 如果在数据存取或者指令存取时不能正常进行地址转换，就通过 MMU 异常的发生通知软件。
4. 如果在指令存取时地址转换信息未注册在 ITLB，就检索 UTLB。当必要的地址转换信息已注册在 UTLB 时，就根据 MMUCR 的 LRUI 位，将该地址转换信息复制到 ITLB。

7.5.2 MMU 的软件管理

对 MMU 进行的软件处理如下：

1. 进行 MMU 相关寄存器的设定。也有通过硬件自动更新的部分寄存器。
2. 进行 TLB 入口的注册、删除和读取。UTLB 入口的注册有使用 LDTLB 指令的方法和直接写存储器映像 UTLB 的方法，而 ITLB 入口的注册只有直接写存储器映像 ITLB 的方法。能通过存取存储器映像 UTLB、ITLB 删除和读取 UTLB、ITLB 入口。
3. 进行 MMU 异常处理。在发生 MMU 异常时，以硬件设定的信息为基础进行处理。

7.5.3 MMU 的指令 (LDTLB)

作为注册 UTLB 入口的指令有 TLB 加载指令 (LDTLB)。当发行 LDTLB 指令时，SH-4A 就将 PTEH 和 PTEL 的内容（在 TLB 扩展模式时还有 PTEA 的内容）复制到 URC 位所指示的 UTLB 入口。LDTLB 指令不进行 ITLB 入口的更新，所以从 UTLB 入口被清除的地址转换信息有可能留在 ITLB 入口。由于 LDTLB 指令为更改地址转换信息的指令，因此必须通过 P1 或者 P2 区的程序来发行。必须在执行 LDTLB 指令后并且在存取 TLB 的有效区域（包含取指令）前，执行以下 1～3 中的任意一项：

1. 必须通过 RTE 指令进行转移。此时，转移目标可以是 TLB 的有效区域。
2. 必须对任意地址（也包含不能缓存的区域）执行 ICBI 指令。
3. 如果在执行 LDTLB 指令前预先设定 IRMC.RLT=0（初始值），就不需要执行特定的指令。但是，必须注意：此方法需要重新取 LDTLB 指令的下一条指令，会降低 CPU 的处理效率。

其中，方法 3 有可能不保证今后的 SuperH 系列。为了保证今后 SuperH 系列的兼容性，建议使用方法 1 或者方法 2。

LDTLB 指令的运行如图 7.16 和图 7.17 所示。

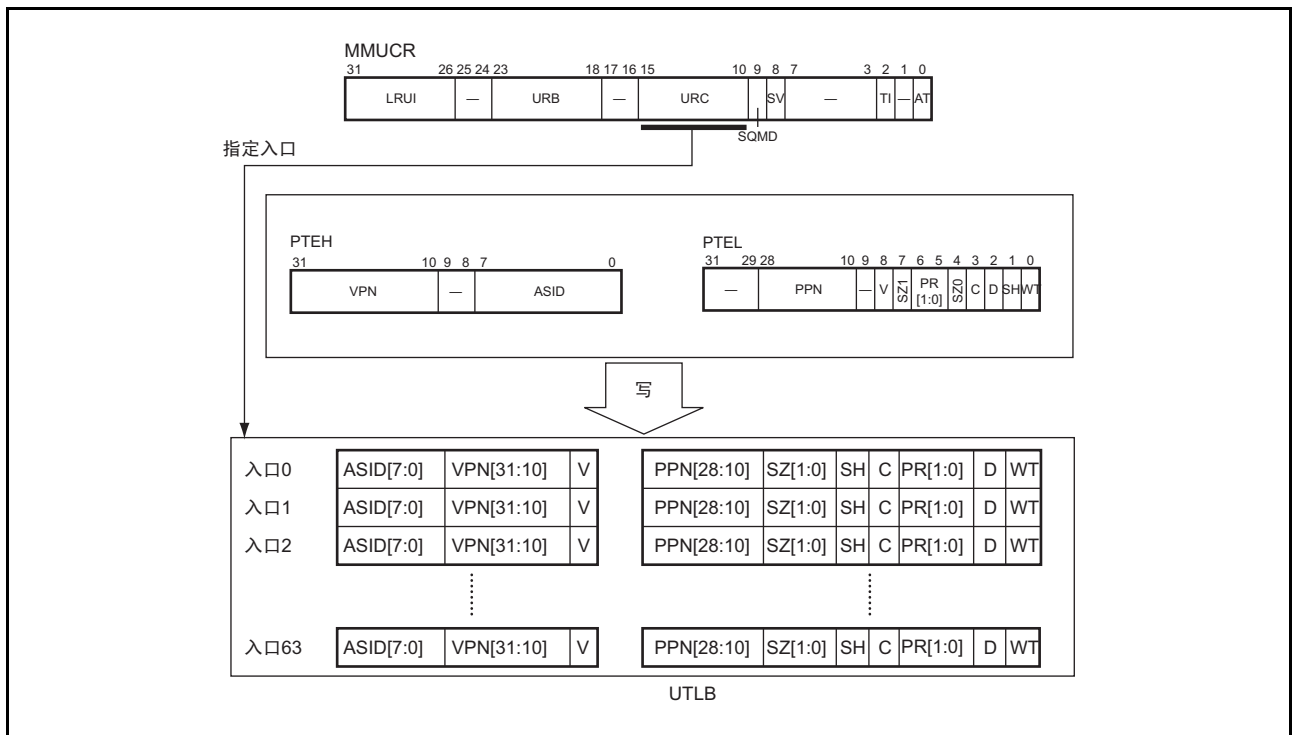


图 7.16 LDTLB 指令的运行 (TLB 兼容模式)

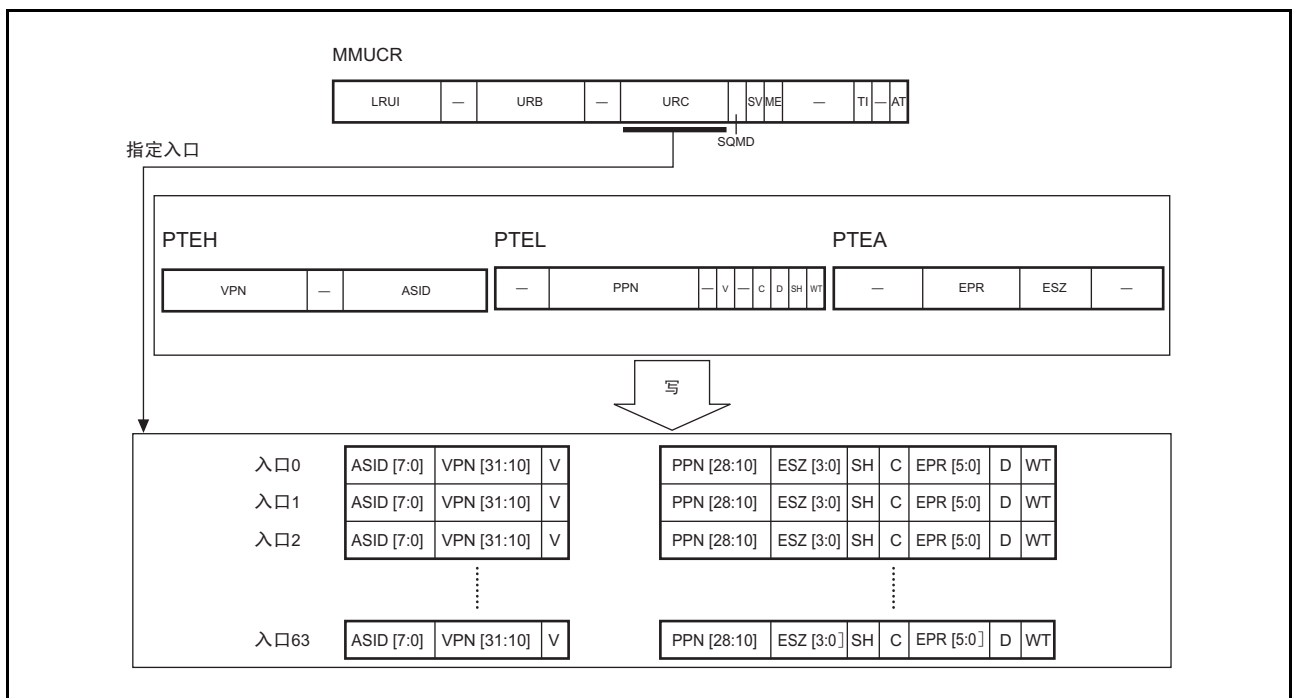


图 7.17 LDTLB 指令的运行 (TLB 扩展模式)

7.5.4 硬件 ITLB 未命中处理

在存取指令时检索 ITLB 而未找到必要的地址转换信息 (ITLB 未命中) 的情况下, SH-4A 通过硬件检索 UTLB, 如果存在必要的地址转换信息, 就注册到 ITLB。此处理称为硬件 ITLB 未命中处理。如果检索 UTLB 仍未找到必要的地址转换信息, 就产生指令 TLB 未命中异常, 转移到软件处理。

7.5.5 同义问题的避免

说明有关 32K 字节操作数高速缓存的情况。

在 TLB 入口注册 1K 或者 4K 字节页时, 有可能发生同义问题。所谓同义问题就是在多个虚拟地址映像到 1 个物理地址时, 相同物理地址的数据注册到高速缓存的多个入口, 从而不能保证数据的一致性。因为指令 TLB 和指令高速缓存只进行数据的读操作, 所以不会发生此问题。为了实现操作数高速缓存的高速运行, SH-4A 使用虚拟地址的 [12:5] 进行入口的指定。但是, 1K 字节页和 4K 字节页分别以虚拟地址的 [12:10] 和虚拟地址的 [12] 为地址转换的对象, 因此转换后的物理地址的 [12:10] 和虚拟地址的 [12:10] 可能不同。

注册到 UTLB 入口的地址转换信息有以下限制:

1. 在将多个 1K 字节页的 UTLB 入口转换为相同物理地址的地址转换信息注册到 UTLB 时, VPN[12:10] 必须相等。
2. 在将多个 4K 字节页的 UTLB 入口转换为相同的物理地址的地址转换信息注册到 UTLB 时, VPN[12] 必须相等。
3. 对于不同页大小的 UTLB 入口, 不能使用 1K 字节页的 UTLB 入口的物理地址。
4. 对于不同页大小的 UTLB 入口, 不能使用 4K 字节页的 UTLB 入口的物理地址。

上述限制适用于用高速缓存进行的存取。

如果高速缓存的容量不为 32K 字节, 就会改变以上说明的可能发生同义问题的页大小和 UTLB 入口注册时必须相等的 VPN 的位的位置。对于 8K 字节~ 64K 字节的高速缓存, 有可能发生同义问题的页大小如表 7.3 所示。

表 7.3 高速缓存容量和同义问题的避免措施

高速缓存容量	可能发生同义问题的页大小	必须相等的 VPN 的位的位置
8K 字节	1K 字节页	VPN[10]
16K 字节	1K 字节页	VPN[11:10]
32K 字节	1K 字节页	VPN[12:10]
	4K 字节页	VPN[12]
64K 字节	1K 字节页	VPN[13:10]
	4K 字节页	VPN[13:12]

【注】 为了将来 SuperH RISC engine 族的扩展, 在多个地址转换信息使用相同的物理存储器时, VPN[20:10] 必须相等。另外, 在不同页大小的地址转换信息中不能使用相同的物理地址。

7.6 MMU 异常

MMU 异常有指令 TLB 多重命中异常、指令 TLB 未命中异常、指令 TLB 保护违反异常、数据 TLB 多重命中异常、数据 TLB 未命中异常、数据 TLB 保护违反异常、初始页写异常等 7 种异常。有关各异常的发生条件请参照图 7.9、图 7.10、图 7.14、图 7.15 和“第 5 章 异常处理”。

7.6.1 指令 TLB 多重命中异常

在存在多个与存取指令的虚拟地址相同的 ITLB 入口时，发生指令 TLB 多重命中异常。通过硬件 ITLB 未命中处理，在检索 UTLB 时发生多重命中的情况下，也发生指令 TLB 多重命中异常。

在发生指令 TLB 多重命中异常时，进入复位状态，并且不保证高速缓存的一致性。

- **硬件处理**

在发生指令 TLB 多重命中异常时，硬件进行以下处理：

- a. 给 TEA 设定发生异常的虚拟地址。
- b. 给 EXPEVT 设定异常码 H'140。
- c. 转移到复位处理程序 (H'A000 0000)。

- **软件处理 (复位程序)**

通过复位处理程序确认发生多重命中的 ITLB 入口。由于此异常用于程序调试，所以在通常情况下不能发生此异常。

7.6.2 指令 TLB 未命中异常

通过硬件 ITLB 未命中处理，在 UTLB 入口未找到存取指令的虚拟地址所对应的地址转换信息时，发生指令 TLB 未命中异常。指令 TLB 未命中异常的硬件处理和软件处理如下。与发生数据 TLB 未命中异常时的处理相同。

- **硬件处理**

在发生指令 TLB 未命中异常时，硬件进行以下处理：

- a. 给 PTEH 设定发生异常的虚拟地址的 VPN。
- b. 给 TEA 设定发生异常的虚拟地址。
- c. 给 EXPEVT 设定异常码 H'040。
- d. 给 SPC 设定指向发生异常的指令地址的 PC 值。如果异常发生在延迟槽，就给 SPC 设定指向延迟转移指令地址的 PC 值。
- e. 给 SSR 设定发生异常时的 SR 内容，给 SGR 设定此时的 R15。
- f. 将 SR 的 MD 位置 1，转换为特权模式。
- g. 将 SR 的 BL 位置 1，屏蔽此后的异常请求。
- h. 将 SR 的 RB 位置 1。
- i. 转移到 VBR 的内容加上 H'0000 0400 偏移量的地址，开始执行指令 TLB 未命中异常的处理程序。

- **软件处理 (指令 TLB 未命中异常的处理程序)**

通过软件检索外部存储器的页表，并且分配必要的页表入口。为了查找并分配必要的页表入口，软件必须进行以下处理：

- a. 在 TLB 兼容模式时，将记录在外部存储器地址转换表中的页表入口的 PPN、PR、SZ、C、D、SH、V 和 WT 各位的值写到 PTEL；在 TLB 扩展模式时，将记录在外部存储器地址转换表中的页表入口的 PPN、EPR、ESZ、C、D、SH、V 和 WT 各位的值写到 PTEL 和 PTEA。
- b. 在入口置换处通过软件指定被置换的入口时，将该值写到 MMUCR 的 URC。此时，如果 URC 超过 URB，必须在发行 LDTLB 指令后更改为适当的值。

- c. 执行 LDTLB 指令。在 TLB 兼容模式时，将 PTEH 和 PTEL 的内容写到 TLB；在 TLB 扩展模式时，将 PTEH、PTEL 和 PTEA 的内容写到 UTLB。
- d. 最后，必须执行从异常处理返回的指令（RTE），结束异常处理程序，并且将控制返回到通常的流程。但是，必须在 LDTLB 指令的下一条指令之后发行 RTE 指令。
有关 LDTLB 指令的执行，请参照“7.8.1 LDTLB 指令的使用注意事项”。

7.6.3 指令 TLB 保护违反异常

尽管与存取指令的虚拟地址相同的地址转换信息存在于 ITLB 中，但是实际上用 PR 或者 EPR 位指定的存取权是不允许存取类型时，发生指令 TLB 保护违反异常。指令 TLB 保护违反异常的硬件处理和软件处理如下：

- **硬件处理**
在发生指令 TLB 保护违反异常时，硬件进行以下处理：
 1. 给 PTEH 设定发生异常的虚拟地址的 VPN。
 2. 给 TEA 设定发生异常的虚拟地址。
 3. 给 EXPEVT 设定异常码 H'0A0。
 4. 给 SPC 设定指向发生异常的指令地址的 PC 值。如果异常发生在延迟槽，就给 SPC 设定指向延迟转移指令地址的 PC 值。
 5. 给 SSR 设定发生异常时的 SR 内容，给 SGR 设定此时的 R15。
 6. 将 SR 的 MD 位置 1，转换为特权模式。
 7. 将 SR 的 BL 位置 1，屏蔽此后的异常请求。
 8. 将 SR 的 RB 位置 1。
 9. 转移到 VBR 的内容加上 H'0000 0100 偏移量的地址，开始执行指令 TLB 保护违反异常的处理程序。
- **软件处理（指令 TLB 保护违反异常的处理程序）**
必须解决指令 TLB 保护违反，执行从异常处理返回的指令（RTE），结束异常处理程序，并且将控制返回到通常的流程。但是，必须在 LDTLB 指令的下一条指令之后发行 RTE 指令。

7.6.4 数据 TLB 多重命中异常

在存在多个与存取数据的虚拟地址相同的 UTLB 入口时，发生数据 TLB 多重命中异常。

在发生数据 TLB 多重命中异常时，进入复位状态，并且不保证高速缓存的一致性。另外，还可能会损坏发生异常前的 UTLB 中的 PPN 内容。

- **硬件处理**
在发生数据 TLB 多重命中异常时，硬件进行以下处理：
 - a. 给 TEA 设定发生异常的虚拟地址。
 - b. 给 EXPEVT 设定异常码 H'140。
 - c. 转移到复位处理程序（H'A000 0000）。
- **软件处理（复位程序）**
通过复位处理程序确认发生多重命中的 UTLB 入口。由于此异常用于程序调试，所以在通常情况下不能发生此异常。

7.6.5 数据 TLB 命中异常

在 UTLB 中未找到存取数据的虚拟地址所对应的地址转换信息时，发生数据 TLB 未命中异常。数据 TLB 未命中异常的硬件处理和软件处理如下：

- **硬件处理**

在发生数据 TLB 未命中异常时，硬件进行以下处理：

- a. 给 PTEH 设定发生异常的虚拟地址的 VPN。
- b. 给 TEA 设定发生异常的虚拟地址。
- c. 读时，给 EXPEVT 设定异常码 H'040；写时，给 EXPEVT 设定异常码 H'060。
(OCBP、OCBWB：读，OCBI、MOVCA.L：写)
- d. 给 SPC 设定指向发生异常的指令地址的 PC 值。如果异常发生在延迟槽，就给 SPC 设定指向延迟转移指令地址的 PC 值。
- e. 给 SSR 设定发生异常时的 SR 的内容，给 SGR 设定此时的 R15。
- f. 将 SR 的 MD 位置 1，转换为特权模式。
- g. 将 SR 的 BL 位置 1，屏蔽此后的异常请求。
- h. 将 SR 的 RB 位置 1。
- i. 转移到 VBR 的内容加上 H'0000 0400 偏移量的地址，并且开始执行数据 TLB 未命中异常的处理程序。

- **软件处理（数据 TLB 未命中异常的处理程序）**

通过软件检索外部存储器的页表，并且分配必要的页表入口。为了查找并分配必要的页表入口，软件必须进行以下处理：

- a. 在 TLB 兼容模式时，将记录在外部存储器的地址转换表中的页表入口的 PPN、PR、SZ、C、D、SH、V 和 WT 各位的值写到 PTEL；在 TLB 扩展模式时，将记录在外部存储器的地址转换表中的页表入口的 PPN、EPR、ESZ、C、D、SH、V 和 WT 各位的值写到 PTEL 和 PTEA。
- b. 在入口置换处通过软件指定被置换的入口时，将该值写到 MMUCR 的 URC。此时，如果 URC 超过 URB，就必须在发行 LDTLB 指令后更改为适当的值。
- c. 执行 LDTLB 指令。在 TLB 兼容模式时，将 PTEH 和 PTEL 的内容写到 UTLB；在 TLB 扩展模式时，将 PTEH、PTEL 和 PTEA 的内容写到 UTLB。
- d. 最后，必须执行从异常处理返回的指令（RTE），结束异常处理程序，并且将控制返回到通常的流程。但是，必须在 LDTLB 指令的下一条指令之后发行 RTE 指令。
有关 LDTLB 指令的执行，请参照“7.8.1 LDTLB 指令的使用注意事项”。

7.6.6 数据 TLB 保护违反异常

尽管与存取数据的虚拟地址相同的地址转换信息存在于 UTLB 中，但是实际上用 PR 或者 EPR 位指定的存取权是未允许的存取类型时，发生数据 TLB 保护违反异常。数据 TLB 保护违反异常的硬件处理和软件处理如下：

- **硬件处理**

在发生数据 TLB 保护违反异常时，硬件进行以下处理：

- a. 给 PTEH 设定发生异常的虚拟地址的 VPN。
- b. 给 TEA 设定发生异常的虚拟地址。
- c. 读时，给 EXPEVT 设定异常码 H'0A0；写时，给 EXPEVT 设定异常码 H'0C0。
(OCBP、OCBWB：读，OCBI、MOVCA.L：写)
- d. 给 SPC 设定指向发生异常的指令地址的 PC 值。如果异常发生在延迟槽，就给 SPC 设定指向延迟转移指令地址的 PC 值。

- e. 给 SSR 设定发生异常时的 SR 的内容, 给 SGR 设定此时的 R15。
 - f. 将 SR 的 MD 位置 1, 转换为特权模式。
 - g. 将 SR 的 BL 位置 1, 屏蔽此后的异常请求。
 - h. 将 SR 的 RB 位置 1。
 - i. 转移到 VBR 的内容加上 H'0000 0100 偏移量的地址, 并且开始执行数据 TLB 保护违反异常的处理程序。
- **软件处理 (数据 TLB 保护违反异常的处理程序)**
必须解决数据 TLB 保护违反, 执行从异常处理返回的指令 (RTE), 结束异常处理程序, 并且将控制返回到通常的流程。但是, 必须在 LDTLB 指令的下一条指令之后发行 RTE 指令。

7.6.7 初始页写异常

尽管与存取数据 (写) 的虚拟地址相同的地址转换信息存在于 UTLB 入口中, 并且存取权也是允许的存取类型, 但是在 D 位为 0 时发生初始页写异常。初始页写异常的硬件处理和软件处理如下:

- **硬件处理**
在发生初始页写异常时, 硬件进行以下处理:
 - a. 给 PTEH 设定发生异常的虚拟地址的 VPN。
 - b. 给 TEA 设定发生异常的虚拟地址。
 - c. 给 EXPEVT 设定异常码 H'080。
 - d. 给 SPC 设定指向发生异常的指令地址的 PC 值。如果异常发生在延迟槽, 就给 SPC 设定指向延迟转移指令地址的 PC 值。
 - e. 给 SSR 设定发生异常时的 SR 的内容, 给 SGR 设定此时的 R15。
 - f. 将 SR 的 MD 位置 1, 转换为特权模式。
 - g. 将 SR 的 BL 位置 1, 屏蔽此后的异常请求。
 - h. 将 SR 的 RB 位置 1。
 - i. 转移到 VBR 的内容加上 H'0000 0100 偏移量的地址, 并且开始执行初始页写异常的处理程序。
- **软件处理 (初始页写异常的处理程序)**
通过软件进行以下处理:
 - a. 从外部存储器查找必要的页表入口。
 - b. 必须将外部存储器的页表入口的 D 位置 1。
 - c. 在 TLB 兼容模式时, 将记录在外部存储器的地址转换表中的页表入口的 PPN、PR、SZ、C、D、SH、V 和 WT 位的值写到 PTEL; 在 TLB 扩展模式时, 将记录在外部存储器的地址转换表中的页表入口中的 PPN、EPR、ESZ、C、D、SH、V 和 WT 各位的值写到 PTEL 和 PTEA。
 - d. 在入口置换处通过软件指定被置换的入口时, 将该值写到 MMUCR 的 URC。此时, 如果 URC 超过 URB, 就必须在发行 LDTLB 指令后更改为适当的值。
 - e. 执行 LDTLB 指令。在 TLB 兼容模式时, 将 PTEH 和 PTEL 的内容写到 UTLB; 在 TLB 扩展模式时, 将 PTEH、PTEL 和 PTEA 的内容写到 UTLB。
 - f. 最后, 必须执行从异常处理返回的指令 (RTE), 结束异常处理程序, 并且将控制返回到通常的流程。但是, 必须在 LDTLB 指令的下一条指令之后发行 RTE 指令。

7.7 存储器映像 TLB 的结构

为了通过软件管理 ITLB 和 UTLB，在特权模式时，能从 P1/P2 区的程序用 MOV 指令读写 ITLB 和 UTLB 的内容。从其他区域的程序存取时，不保证运行。

必须在存取存储器映像 TLB 后并且在存取（含取指令）P1/P2 区以外的区域前，执行以下 1～3 中的任意一项：

1. 必须通过 RTE 指令进行转移。此时，转移目标可以是 P1/P2 区以外的区域。
2. 必须对任意地址（也包含不能缓存的区域）执行 ICBI 指令。
3. 如果在存取存储器映像 TLB 前预先设定 IRMCR.MT=0（初始值），就不需要执行特定的指令。但是，必须注意：此方法需要重新取 MMUCR 更新指令的下一条指令，会降低 CPU 的处理效率

其中，方法 3 有可能不保证今后的 SuperH 系列。为了保证今后 SuperH 系列的兼容性，建议使用方法 1 或者方法 2。

ITLB 和 UTLB 分配到虚拟地址空间的 P4 区。

在 TLB 兼容模式时，对于 ITLB，能将 VPN、V 和 ASID 作为地址阵列、PPN、V、SZ、PR、C 和 SH 作为数据阵列进行存取；对于 UTLB，能将 VPN、D、V 和 ASID 作为地址阵列、PPN、V、SZ、PR、C、D、WT、SH 作为数据阵列进行存取。V 和 D 能从地址阵列或者数据阵列进行存取。

在 TLB 扩展模式时，对于 ITLB，能将 VPN、V 和 ASID 作为地址阵列、PPN、V、ESZ、EPR、C 和 SH 作为数据阵列进行存取。对于 UTLB，能将 VPN、D、V 和 ASID 作为地址阵列、PPN、V、ESZ、EPR、C、D、WT 和 SH 作为数据阵列进行存取。V 和 D 能从地址阵列和数据阵列进行存取。

在 TLB 兼容模式或者 TLB 扩展模式时，只能进行长字存取。不能对此区域取指令。对保留位只能写 0，不保证读取值。

7.7.1 ITLB 地址阵列

ITLB 的地址阵列分配到 P4 区的 H'F200 0000～H'F2FF FFFF。在存取地址阵列时，需要指定 32 位的地址部（读写时）和 32 位的数据部（写时）。地址部指定用于选择存取入口的信息，数据部指定写地址阵列的 VPN、V 和 ASID。

地址部中的 [31:24] 为 H'F2，表示 ITLB 地址阵列；[9:8] 选择入口。为了进行长字存取，地址部的 [1:0] 必须指定 0。

数据部中的 [31:10] 表示 VPN，[8] 表示 V，[7:0] 表示 ASID。

能对 ITLB 地址阵列进行以下 2 种操作：

1. ITLB 地址阵列 读
从地址部设定的入口所对应的 ITLB 入口，将 VPN、V 和 ASID 读到数据部。
2. ITLB 地址阵列 写
对地址部设定的入口所对应的 ITLB 入口，写数据部指定的 VPN、V 和 ASID。

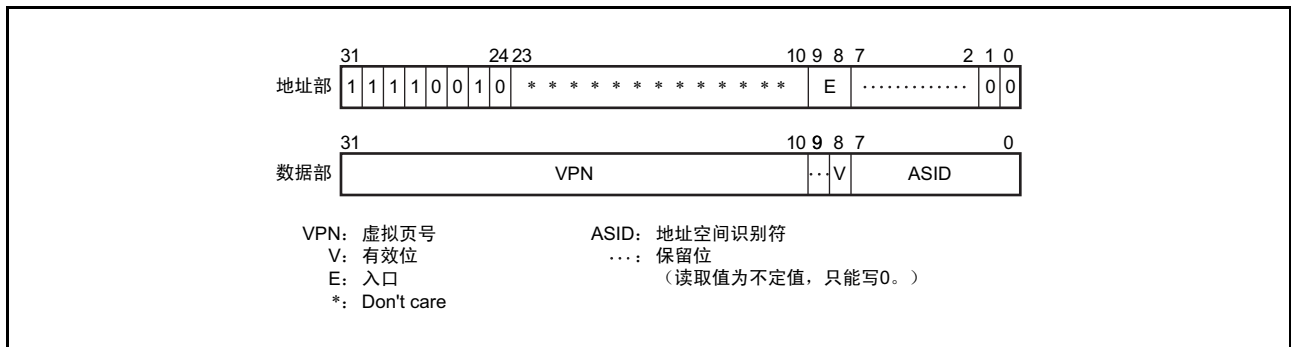


图 7.18 存储器映像 ITLB 地址阵列

7.7.2 ITLB 数据阵列 (TLB 兼容模式)

ITLB 的数据阵列分配到 P4 区的 H'F300 0000 ~ H'F37F FFFF。在存取数据阵列时，需要指定 32 位的地址部 (读写时) 和 32 位的数据部 (写时)。地址部指定用于选择存取入口的信息，数据部指定写地址阵列的 PPN、V、SZ、PR、C 和 SH。

地址部中的 [31:23] 为 H'F30，表示 ITLB 数据阵列；[9:8] 选择入口。

数据部中的 [28:10] 表示 PPN，[8] 表示 V，[7] 和 [4] 表示 SZ，[6] 表示 PR，[3] 表示 C，[1] 表示 SH。能对 ITLB 数据阵列进行以下 2 种操作：

1. ITLB 数据阵列 读
从地址部设定的入口所对应的 ITLB 入口，将 PPN、V、SZ、PR、C 和 SH 读到数据部。
2. ITLB 数据阵列 写
对地址部设定的入口对应的 ITLB 入口，写数据部指定的 PPN、V、SZ、PR、C 和 SH。

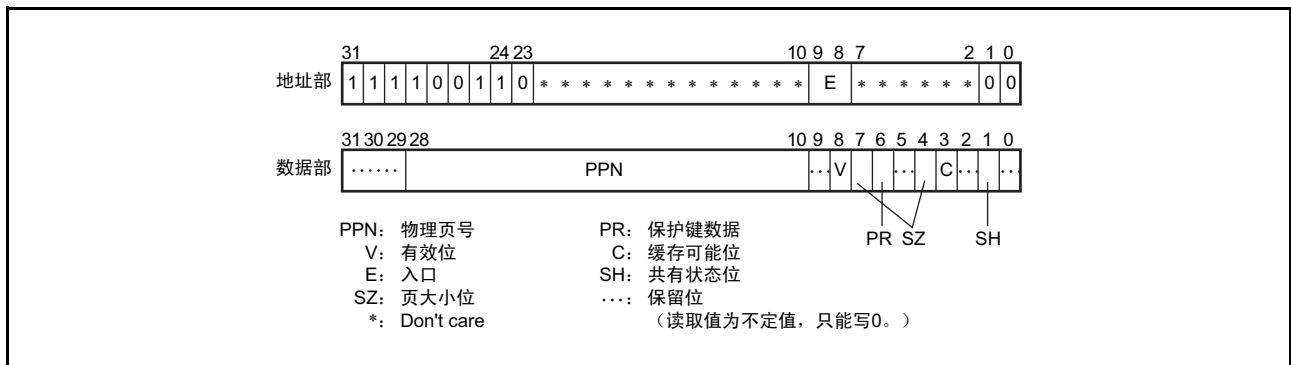


图 7.19 存储器映像 ITLB 数据阵列 (TLB 兼容模式)

7.7.3 ITLB 数据阵列 (TLB 扩展模式)

在 TLB 扩展模式时，ITLB 数据阵列的名称变为 ITLB 数据阵列 1，并且追加 ITLB 数据阵列 2，能存取 EPR 和 ESZ。在 TLB 扩展模式中，ITLB 数据阵列 1 的 PR 和 SZ 位为保留位，只能写 0。在写 ITLB 数据阵列 1 时，必须随后写相同入口的 ITLB 数据阵列 2。

在 TLB 兼容模式 (MMUCR.ME=0) 时，不能存取 ITLB 数据阵列 2。不保证存取操作。

(1) ITLB 数据阵列 1

在 TLB 扩展模式中，兼容模式的 PR 和 SZ 位所对应的数据部的 bit7、6、4 为保留位，只能写 0。

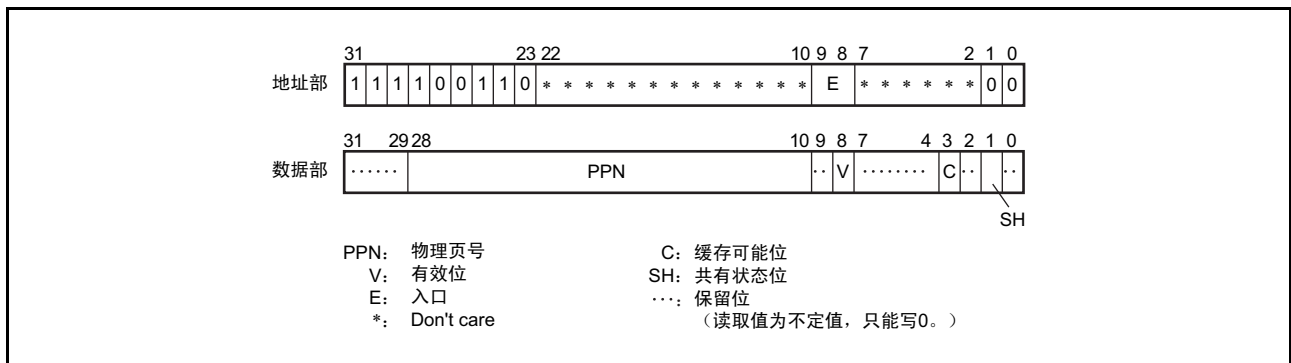


图 7.20 存储器映像 ITLB 数据阵列 1 (TLB 扩展模式)

(2) ITLB 数据阵列 2

ITLB 的数据阵列分配到 P4 区的 H'F380 0000 ~ H'F3FF FFFF。在存取数据阵列 2 时，需要指定 32 位的地址部（读写时）和 32 位的数据部（写时）。地址部指定用于选择存取入口的信息，数据部指定写数据阵列 2 的 EPR 和 ESZ。

地址部中的 [31:23] 为 H'F38，表示 ITLB 数据阵列 2；[9:8] 选择入口。

数据部中的 [13][11][10][8] 分别表示 EPR[5][3][2][0]，[7:4] 表示 ESZ。

能对 ITLB 数据阵列 2 进行以下 2 种操作：

1. TLB 数据阵列 2 读
从地址部指定的入口所对应的 ITLB 入口，将 EPR 和 ESZ 读到数据部。
2. ITLB 数据阵列 2 写
对地址部指定的入口所对应的 ITLB 入口，写数据部指定的 EPR 和 ESZ。

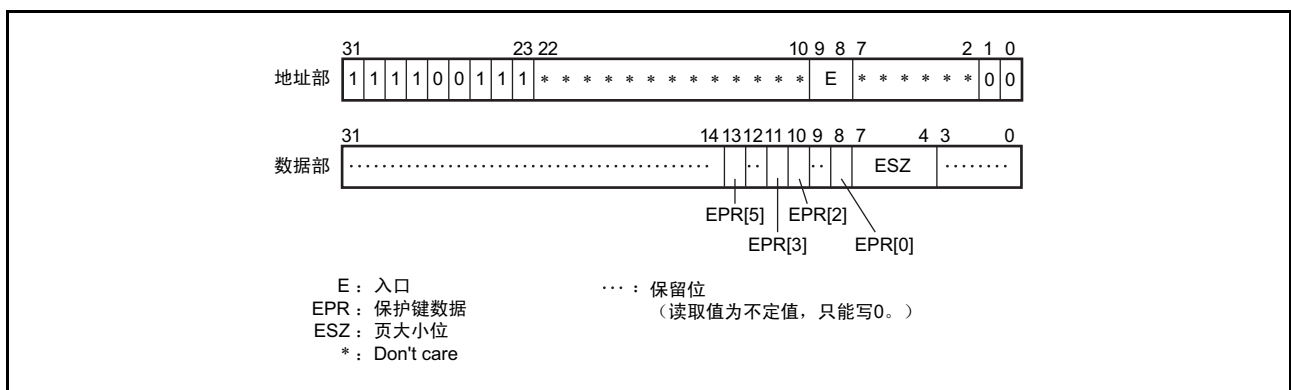


图 7.21 存储器映像 ITLB 数据阵列 2 (TLB 扩展模式)

7.7.4 UTLB 地址阵列

UTLB 的地址阵列分配到 P4 区的 H'F600 0000 ~ H'F60F FFFF。在存取地址阵列时，需要指定 32 位的地址部（读写时）和 32 位的数据部（写时）。地址部指定用于选择存取入口的信息，数据部指定写地址阵列的 VPN、D、V 和 ASID。

地址部中的 [31:20] 为 H'F60，表示 UTLB 地址阵列；[13:8] 选择入口。地址部 [7] 的联想位 (A 位) 指定在写 UTLB 地址阵列时是否进行地址比较。

数据部中的 [31:10] 表示 VPN，[9] 表示 D，[8] 表示 V，[7:0] 表示 ASID。

能对 UTLB 地址阵列进行以下 3 种操作：

1. UTLB 地址阵列 读
从地址部设定的入口所对应的 UTLB 入口，将 VPN、D、V 和 ASID 读到数据部。读操作时，无论地址部指定的联想位是 1 还是 0，都不进行联想操作。
2. UTLB 地址阵列 写 (无联想)
对地址部设定的入口所对应的 UTLB 入口，写数据部指定的 VPN、D、V 和 ASID。必须将地址部的 A 位置 0。
3. UTLB 地址阵列 写 (有联想)
在地址部的 A 位为 1 时，用数据部指定的 VPN 和 PTEH.ASID，与 UTLB 的全入口进行比较。比较规则为普通地址的比较规则，但是在发生 UTLB 未命中时不产生异常，为无操作。在比较中，如果存在数据部指定的 VPN 所对应的 UTLB 入口，就对该入口写数据部指定的 D 和 V。此联想操作也同时对 ITLB 进行，如果在 ITLB 中存在相同的入口，就对该入口写 V。在 UTLB 的比较时为无操作，但是在 ITLB 中存在相同的入口，就只在 ITLB 侧进行写操作。另外在 UTLB 和 ITLB 中都存在相同的入口，UTLB 的信息也被写到 ITLB。

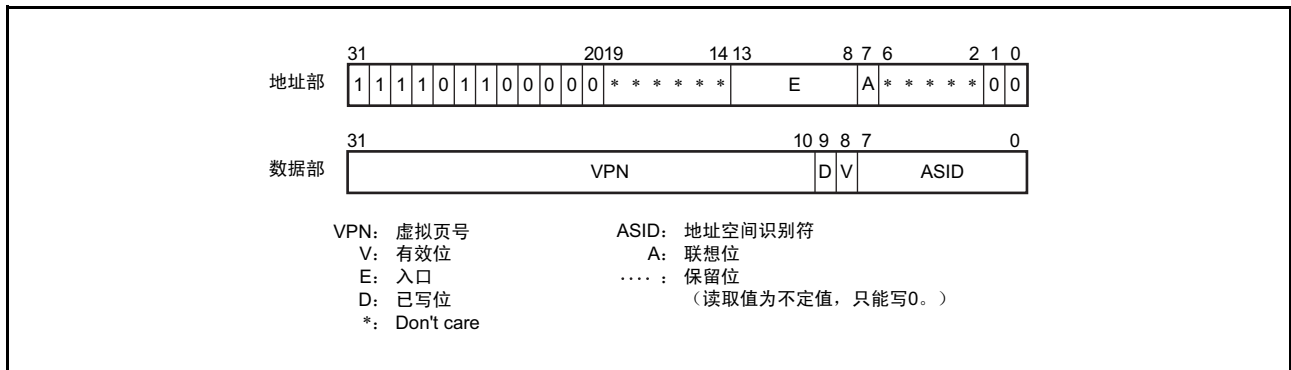


图 7.22 存储器映像 UTLB 地址阵列

7.7.5 UTLB 数据阵列 (TLB 兼容模式)

UTLB 的地址阵列分配到 P4 区的 H'F700 0000 ~ H'F70F FFFF。在存取地址阵列时, 需要指定 32 位的地址部 (读写时) 和 32 位的数据部 (写时)。地址部指定用于选择存取入口的信息, 数据部指定写地址阵列的 PPN、V、SZ、PR、C、D、SH 和 WT。

地址部中的 [31:20] 为 H'F70, 表示 UTLB 地址阵列; [13:8] 选择入口。

数据部中的 [28:10] 表示 PPN, [8] 表示 V, [7] 和 [4] 表示 SZ, [6:5] 表示 PR, [3] 表示 C, [2] 表示 D, [1] 表示 SH, [0] 表示 WT。

能对 UTLB 地址阵列进行以下 2 种操作:

1. UTLB 数据阵列 读

从地址部设定的入口所对应的 UTLB 入口, 将 PPN、V、SZ、PR、C、D、SH 和 WT 读到数据部。

2. UTLB 数据阵列 写

对地址部设定的入口所对应的 UTLB 入口, 写数据部指定的 PPN、V、SZ、PR、C、D、SH 和 WT。

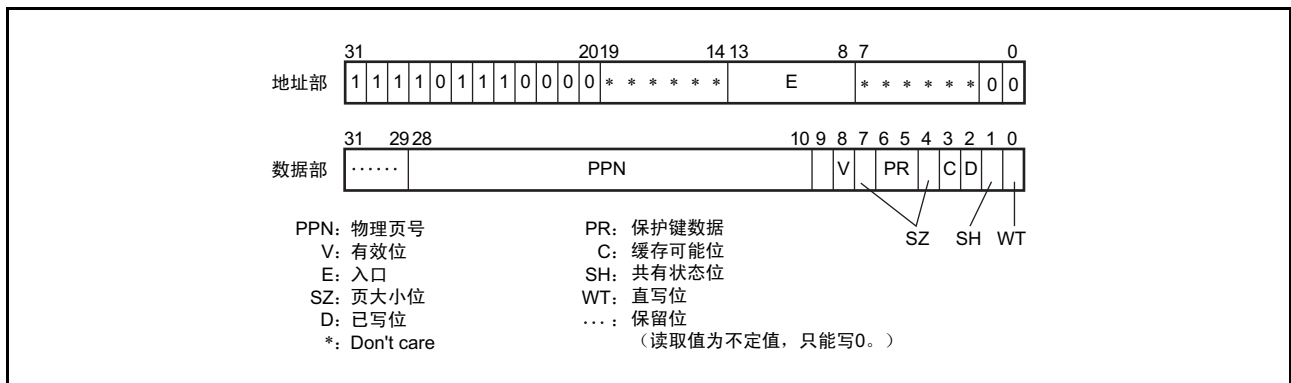


图 7.23 存储器映像 UTLB 数据阵列 (TLB 兼容模式)

7.7.6 UTLB 数据阵列 (TLB 扩展模式)

在 TLB 扩展模式时，UTLB 数据阵列的名称变为 UTLB 数据阵列 1，并且追加 UTLB 数据阵列 2，能存取 EPR 和 ESZ。在 TLB 扩展模式中，UTLB 数据阵列 1 的 PR 和 SZ 位为保留位，只能写 0。在写 UTLB 数据阵列 1 时，必须随后写相同入口的 UTLB 数据阵列 2。

在 TLB 兼容模式 (MMUCR.ME=0) 时，不能存取 UTLB 数据阵列 2。不保证存取操作。

(1) UTLB 数据阵列 1

在 TLB 扩展模式中，兼容模式的 PR 和 SZ 位所对应的数据部的 bit7 ~ 4 为保留位，只能写 0。

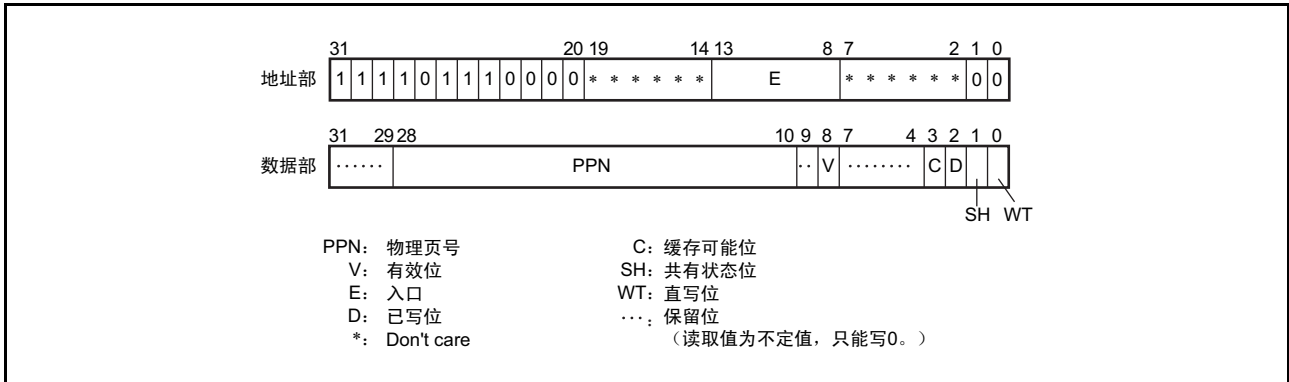


图 7.24 存储器映像 UTLB 数据阵列 1 (TLB 扩展模式)

(2) UTLB 数据阵列 2

UTLB 的数据阵列分配到 P4 区的 H'F780 0000 ~ H'F78F FFFF。在存取数据阵列 2 时，需要指定 32 位的地址部 (读写时) 和 32 位的数据部 (写时)。地址部指定用于选择存取入口的信息，数据部指定写数据阵列 2 的 EPR 和 ESZ。

地址部中的 [31:20] 为 H'F78，表示 UTLB 数据阵列 2；[13:8] 选择入口。

数据部中的 [13:8] 表示 EPR，[7:4] 表示 ESZ。

能对 UTLB 数据阵列 2 进行以下 2 种操作：

1. UTLB 数据阵列 2 读

从地址部指定的入口所对应的 UTLB 入口，将 EPR 和 ESZ 读到数据部。

2. UTLB 数据阵列 2 写

对地址部指定的入口所对应的 UTLB 入口，写数据部指定的 EPR 和 ESZ。

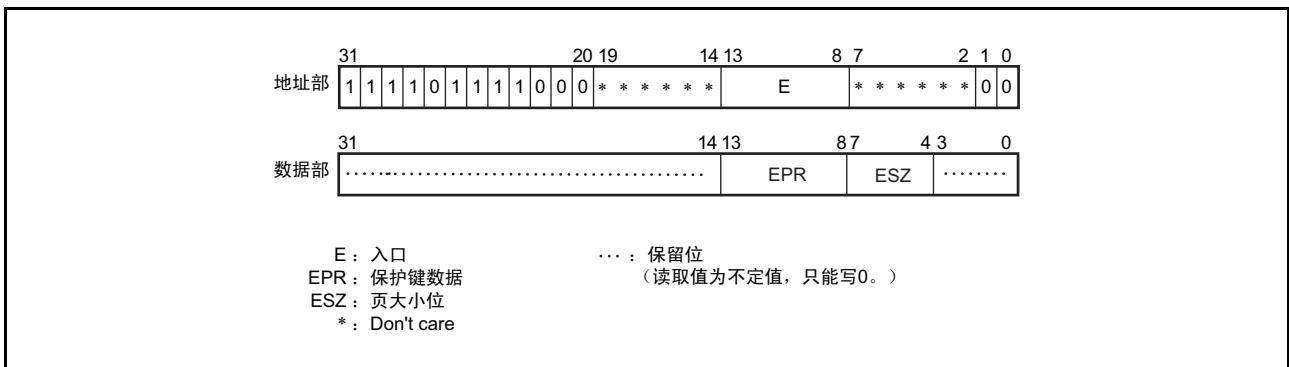


图 7.25 存储器映像 UTLB 数据阵列 2 (TLB 扩展模式)

7.8 使用时的注意事项

7.8.1 LDTLB 指令的使用注意事项

在不通过软件写 MMUCR.URC 而执行 LDTLB 指令时，必须进行以下 1 或者 2 中的任意一项：

1. TLB 未命中异常处理程序*1 必须分配在 P1 区、P2 区或者内部存储区，TLB 未命中异常处理程序中的指令存取*2 必须为 P1 区、P2 区或者内部存储区。在将 TLB 未命中异常处理程序分配到内部存储器时，必须将 RAMCR 的 PR 位清 0（初始值）。另外，不能在 TLB 未命中异常处理程序中使用 FDIV 指令或者 FSQRT 指令。
2. 在发生 TLB 未命中异常时，必须在将 MMUCR.URC 加 1 后执行 LDTLB 指令。

【注】 *1 异常处理程序：是指从发生异常时的 VBR+ 偏移的地址到返回原程序的 RTE 和 RTE 的延迟槽所执行的所有指令。

*2 指令存取：包含 PREFI 指令和 ICBI 指令的存取。

第 8 章 高速缓存

本 LSI 内置用于指令的 32K 字节指令高速缓存（IC）和用于数据的 32K 字节操作数高速缓存（OC）。

8.1 特点

高速缓存的特点如表 8.1 所示。

为了高速写外部存储器，SH-4A 支持 32 字节×2 的存储队列（SQ）。SQ 的特点如表 8.2 所示。

表 8.1 高速缓存的特点

项目	指令高速缓存	操作数高速缓存
容量	32K 字节高速缓存	32K 字节高速缓存
方式	4 路组相联，虚拟地址变址 / 物理地址特征位	4 路组相联，虚拟地址变址 / 物理地址特征位
块大小	32 字节	32 字节
入口数	256 个入口 / 路	256 个入口 / 路
写方式	—	能选择回拷 / 直写
置换方式	LRU（Least Recently Used）算法	LRU（Least Recently Used）算法

表 8.2 存储队列的特点

项目	存储队列
容量	2×32 字节
地址	H'E000 0000 ~ H'E3FF FFFF
写	存储指令（1 个周期写）
回写	预取指令（PREF 指令）
存取权	禁止 MMU 时：由 MMU 控制寄存器（MMUCR）的 SQMD 位决定 允许 MMU 时：由各页的 PR 决定

SH-4A 的操作数高速缓存为 4 路组相联方式，各路由 256 个高速缓存块构成。操作数高速缓存的结构如图 8.1 所示。

指令高速缓存为 4 路组相联方式，各路由 256 个高速缓存块构成。指令高速缓存的结构如图 8.2 所示。

为了降低功耗，SH-4A 具有 IC 的通路预测机构。另外，能通过非支持检测异常寄存器（EXPMASK）将存储器映像联想写功能作为异常进行检测。详细内容请参照“第 5 章 异常处理”。

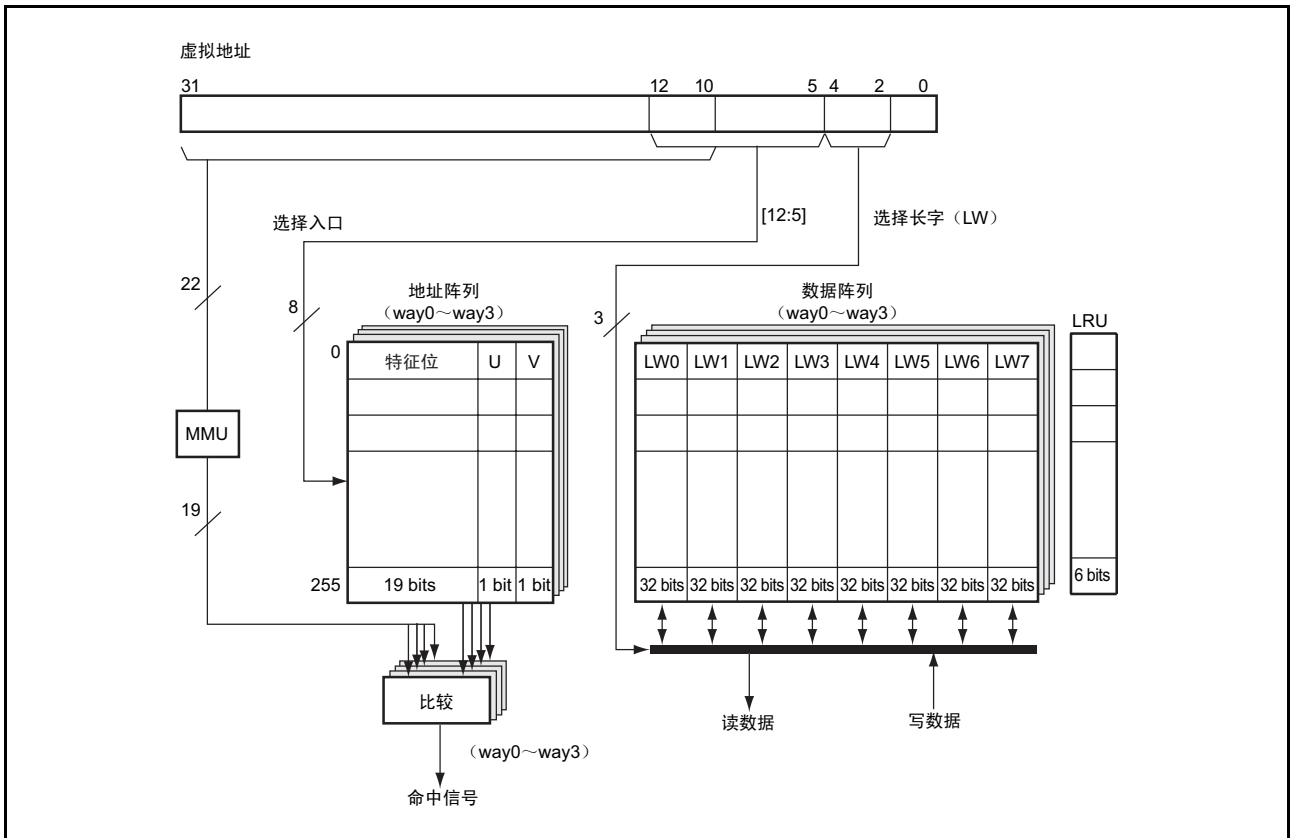


图 8.1 操作数高速缓存的结构（高速缓存容量 =32K 字节时）

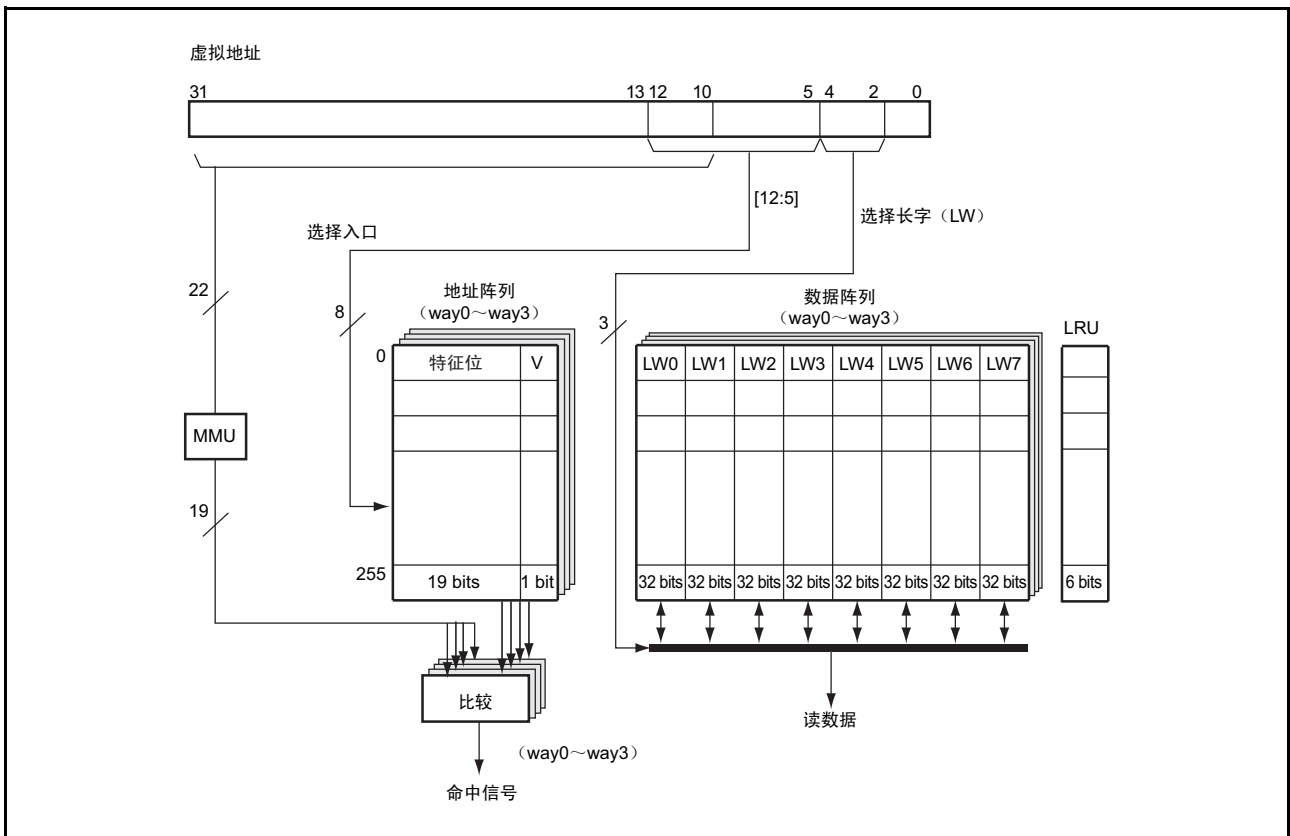


图 8.2 指令高速缓存的结构（高速缓存容量 =32K 字节时）

(1) 特征位

保存被缓存的数据线的物理地址 29 位的高 19 位。在上电复位时特征位不被初始化。

(2) V 位（有效位）

表示在高速缓存块中是否保存了有效数据。当此位为 1 时，该高速缓存块的数据有效。在上电复位时，V 位被初始化为 0。

(3) U 位（已写位）

在使用高速缓存期间，如果以回拷模式给高速缓存块写数据，U 位就变为 1，表示高速缓存块中的数据 and 外部存储器中的数据不相同。只要不通过存取存储器映像高速缓存（参照“8.6 存储器映像高速缓存的结构”）来改写 U 位，就在使用高速缓存期间，直写模式不会使 U 位变为 1。在上电复位时，U 位被初始化为 0。

(4) 数据部

数据部保存高速缓存块的 32 字节（256 位）的数据。在上电复位时，数据阵列不被初始化。

(5) LRU 部

通过 4 路组相联方式，最多能将 4 个相同入口地址的数据注册到高速缓存。在注册入口时，LRU 位表示注册到 4 路中的哪一路。LRU 位由各入口的 6 位构成，由硬件控制。作为通路的选择算法，使用选择最初存取的通路的 LRU（Least Recently Used）算法。在上电复位时，LRU 位被初始化为 0。

8.2 寄存器说明

有关高速缓存的寄存器如下所示：

表 8.3 寄存器结构

名称	略称	R/W	P4 区地址*	区域 7 地址*	长度
高速缓存控制寄存器	CCR	R/W	H'FF00 001C	H'1F00 001C	32
队列地址控制寄存器 0	QACR0	R/W	H'FF00 0038	H'1F00 0038	32
队列地址控制寄存器 1	QACR1	R/W	H'FF00 003C	H'1F00 003C	32
内部存储器控制寄存器	RAMCR	R/W	H'FF00 0074	H'1F00 0074	32

【注】 * P4 区地址是使用虚拟地址空间 P4 区的地址，区域 7 地址是使用 TLB 存取物理地址空间区域 7 的地址。

表 8.4 各处理模式的寄存器状态

名称	略称	上电复位	睡眠	待机
高速缓存控制寄存器	CCR	H'0000 0000	保持	保持
队列地址控制寄存器 0	QACR0	不定值	保持	保持
队列地址控制寄存器 1	QACR1	不定值	保持	保持
内部存储器控制寄存器	RAMCR	H'0000 0000	保持	保持

8.2.1 高速缓存控制寄存器 (CCR)

CCR 选择高速缓存的运行模式和高速缓存的写模式，并且设定高速缓存的所有入口的无效。

必须通过不能缓存的 P2 区或者 IL 存储器中的程序进行 CCR 的改写。必须在更新 CCR 后并且在存取（含取指令）能缓存的区域前，执行以下 1～3 中的任意一项：

1. 必须通过 RTE 指令进行转移。此时，转移目标可以是能缓存的区域。
2. 必须对任意地址（也包含不能缓存的区域）执行 ICBI 指令。
3. 如果在更新 CCR 前预先设定 IRMC.R2=0（初始值），就不需要执行特定的指令。但是，必须注意：此方法需要重新取 CCR 更新指令的下一条指令，所以会降低 CPU 的处理性能。

其中，方法 3 有可能不保证今后的 SuperH 系列。为了保证今后 SuperH 系列的兼容性，建议使用方法 1 或者方法 2。

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
	—	—	—	—	ICI	—	—	ICE	—	—	—	—	OCI	CB	WT	OCE
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31～12	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。
11	ICI	0	R/W	IC 无效化位 如果将此位置 1，IC 的所有入口的 V 位就被置 0。读取值为 0。
10、9	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。
8	ICE	0	R/W	IC 有效位 选择使用 IC。但是，在进行地址转换时，如果页管理信息的 C 位不为 1，就不能使用 IC。 0：不使用 IC 1：使用 IC
7～4	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。
3	OCI	0		OC 无效化位 如果将此位置 1，OC 的所有入口的 V 和 U 位就被置 0。读取值为 0。
2	CB	0	R/W	回拷位 表示 P1 区的高速缓存的写模式。 0：直写模式 1：回拷模式
1	WT	0	R/W	直写模式 表示 P0、U0 和 P3 区的高速缓存的写模式。但是，在进行地址转换时，优先页管理信息的 WT 位的值。 0：回拷模式 1：直写模式
0	OCE	0	R/W	OC 有效位 选择使用 OC。但是，在进行地址转换时，如果页管理信息的 C 位不为 1，就不能使用 OC。 0：不使用 OC 1：使用 OC

8.2.2 队列地址控制寄存器 0 (QACR0)

QACR0 设定禁止 MMU 时的存储队列 0 (SQ0) 的映像区域。

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
	—	—	—	—	—	—	—	—	—	—	AREA0			—	—	
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/W	R/W	R/W	R	R

位	位名	初始值	R/W	说明
31 ~ 5	—	全 0	R	保留位 有关此位的读写, 请参照“产品使用上的注意事项”。
4 ~ 2	AREA0	不定值	R/W	在禁止 MMU 时, 对 SQ0 生成物理地址 28 ~ 26。
1、0	—	全 0	R	保留位 有关此位的读写, 请参照“产品使用上的注意事项”。

8.2.3 队列地址控制寄存器 1 (QACR1)

QACR1 设定禁止 MMU 时的存储队列 1 (SQ1) 的映像区域。

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
	—	—	—	—	—	—	—	—	—	—	AREA1			—	—	
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/W	R/W	R/W	R	R

位	位名	初始值	R/W	说明
31 ~ 5	—	全 0	R	保留位 有关此位的读写, 请参照“产品使用上的注意事项”。
4 ~ 2	AREA1	不定值	R/W	在禁止 MMU 时, 对 SQ1 生成物理地址 28 ~ 26。
1、0	—	全 0	R	保留位 有关此位的读写, 请参照“产品使用上的注意事项”。

8.2.4 内部存储器控制寄存器 (RAMCR)

RAMCR 控制 IC 和 OC 的通路数以及 IC 通路预测。

必须通过不能缓存的 P2 区中的程序进行 RAMCR 的改写。必须在更新 RAMCR 后并且在存取（含取指令）能缓存的区域或者 IL 存储区前，执行以下 1～3 中的任意一项：

1. 必须通过 RTE 指令进行转移。此时，转移目标可以是不能缓存的区域或者 IL 存储区。
2. 必须对任意地址（也包含不能缓存的区域）执行 ICBI 指令。
3. 如果在更新 RAMCR 前预先设定 IRMCR.R2=0（初始值），就不需要执行特定的指令。但是，必须注意：此方法需要重新取 RAMCR 更新指令的下一条指令，所以会降低 CPU 的处理性能。

其中，方法 3 有可能不保证今后的 SuperH 系列。为了保证今后 SuperH 系列的兼容性，建议使用方法 1 或者方法 2。

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
	—	—	—	—	—	—	RMD	RP	IC2W	OC2W	ICWPD	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R

位	位名	初始值	R/W	说明
31～10	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。
9	RMD	0	R/W	内部存储器存取模式位 详细内容请参照“9.4 内部存储器的保护功能”。
8	RP	0	R/W	内部存储器保护模式位 详细内容请参照“9.4 内部存储器的保护功能”。
7	IC2W	0	R/W	IC 2 通路模式位 0: IC 为 4 通路工作 1: IC 为 2 通路工作 详细内容请参照“8.4.3 IC 2 通路模式”。
6	OC2W	0	R/W	OC 2 通路模式位 0: OC 为 4 通路工作 1: OC 为 2 通路工作 详细内容请参照“8.3.6 OC 2 通路模式”。
5	ICWPD	0	R/W	IC 通路预测禁止位 选择使用 IC 通路预测。 0: 指令高速缓存进行通路预测 1: 指令高速缓存不进行通路预测
4～0	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。

8.3 操作数高速缓存的运行说明

8.3.1 读操作

在操作数高速缓存（OC）有效（CCR.OCE=1）并且从能缓存的区域读数据时，OC 进行以下运行：

1. 从由虚拟地址的位[12:5]指向的各路高速缓存块，读取特征位、V位、U位和LRU位。
2. 将由MMU进行虚拟地址转换后的物理地址的位[28:10]和从各路读取的特征位进行比较，
 - 当特征位相同并且存在V位为1的通路时 →3.
 - 当特征位相同而不存在V位为1的通路并且由LRU位选择的置换对象通路的U位为0时 →4.
 - 当特征位相同而不存在V位为1的通路并且由LRU位选择的置换对象通路的U位为1时 →5.
3. 高速缓存命中
从命中通路的数据部，以存取长度为单位，读取由虚拟地址的位[4:0]指向的数据。为了保持最新的命中通路，更新LRU位。
4. 高速缓存未命中（无回写）
从对应虚拟地址的物理地址空间，将数据读到置换对象通路的高速缓存块。从包含高速缓存未命中数据的4个字（8字节），按顺序以环绕方式读数据，当该数据到达高速缓存时，将读取的数据返回给CPU。在读取剩余的1个高速缓存块的数据时，CPU能进行下一个处理。在读完1个高速缓存块的数据时，注册物理地址中的特征位，并且将V位和U位分别置1和0。为了保持最新的置换通路，更新LRU位。
5. 高速缓存未命中（有回写）
将置换对象通路的高速缓存块的特征位和数据部保存到回写缓冲器。此后，从对虚拟地址的物理地址空间，将数据读到置换对象通路的高速缓存块。从包含高速缓存未命中数据的4个字（8字节），按顺序以环绕的方式读数据，当该数据到达高速缓存时，将读取的数据返回给CPU。在读取剩余的1个高速缓存块的数据时，CPU能进行下一个处理。在读完1个高速缓存块的数据时，注册物理地址中的特征位，并且将V位和U位分别置1和0。为了保持最新的置换通路，更新LRU位。此后，将回写缓冲器的数据回写到外部存储器。

8.3.2 预取操作

在操作数高速缓存（OC）有效（CCR.OCE=1）并且从能缓存的区域将数据预取到OC时，OC进行以下运行：

1. 从由虚拟地址的位[12:5]指向的各路高速缓存块，读取特征位、V位、U位和LRU位。
2. 将由MMU进行虚拟地址转换后的物理地址的位[28:10]和从各路读取的特征位进行比较，
 - 当特征位相同并且存在V位为1的通路时 →3
 - 当特征位相同而不存在V位为1的通路并且由LRU位选择的置换对象通路的U位为0时 →4
 - 当特征位相同而不存在V位为1的通路并且由LRU位选择的置换对象通路的U位为1时 →5.
3. 高速缓存命中
为了保持最新的命中通路，更新LRU位。
4. 高速缓存未命中（无回写）
从对应虚拟地址的物理地址空间，将数据读到置换对象通路的高速缓存块。从包含高速缓存未命中数据的4个字（8字节），按顺序以环绕方式读数据。在预取操作中，CPU不等待数据的到达而在读取1个高速缓存块的数据时能进行下一个处理。在读完1个高速缓存块的数据时，注册物理地址中的特征位，并且将V位和U位分别置1和0。为了保持最新的置换通路，更新LRU位。
5. 高速缓存未命中（有回写）

将置换对象通路的高速缓存块的特征位和数据部保存到回写缓冲器。此后，从对应虚拟地址的物理地址空间，将数据读到置换对象通路的高速缓存块。从包含高速缓存未命中数据的4个字（8字节），按顺序以环绕方式读数据。在预取操作中，CPU不等待数据的到达而能在读取1个高速缓存块的数据时进行下一个处理。在读完1个高速缓存块的数据时，注册物理地址中的特征位，并且将V位和U位分别置1和0。为了保持最新的置换通路，更新LRU位。此后，将回写缓冲器的数据回写到外部存储器。

8.3.3 写操作

在操作数高速缓存（OC）有效（CCR.OCE=1）并且将数据写到能缓存的区域时，OC进行以下运行：

1. 从由虚拟地址的位[12:5]指向的各路高速缓存块，读取特征位、V位、U位和LRU位。
2. 将由MMU进行虚拟地址转换后的物理地址的位[28:10]和从各路读取的特征位进行比较，并且根据对象区域的属性，

- | | 回拷 | 直写 |
|---|-----|-----|
| • 当特征位相同并且存在V位为1的通路时 | →3. | →4. |
| • 当特征位相同而不存在V位为1的通路并且由LRU位选择的置换对象通路的U位为0时 | →5. | →7. |
| • 当特征位相同而不存在V位为1的通路并且由LRU位选择的置换对象通路的U位为1时 | →6. | →7. |

3. 高速缓存命中（回拷）

对命中通路数据部，以存取长度为单位，将数据写到由虚拟地址的位[4:0]指向的数据位置，并且将U位置1。为了保持最新的命中通路，更新LRU位。

4. 高速缓存命中（直写）

对命中通路的数据部，以存取长度为单位，将数据写到由虚拟地址的位[4:0]指向的数据位置，同时也将数据写到对应虚拟地址的外部存储器。为了保持最新的命中通路，更新LRU位，此时不更新U位。

5. 高速缓存未命中（回拷、无回写）

对置换对象通路的数据部，以存取长度为单位，将数据写到由虚拟地址的位[4:0]指向的数据位置，并且从对应虚拟地址的物理地址空间，将数据读到置换对象通路的高速缓存块（但是，已写的高速缓存未命中数据除外）。从包含高速缓存未命中数据的4个字（8字节），按顺序以环绕方式读数据。在读取1个高速缓存块的数据时，CPU能进行下一个处理。在读完1个高速缓存块的数据时，注册物理地址中的特征位，并且将V位和U位置1。为了保持最新的置换通路，更新LRU位。

6. 高速缓存未命中（回拷、有回写）

将置换对象通路的高速缓存块的特征位和数据部保存到回写缓冲器。此后，对置换对象通路的数据部，以存取长度为单位，将数据写到由虚拟地址的位[4:0]指向的数据位置，并且从对应虚拟地址的物理地址空间，将数据读到置换对象通路的高速缓存块（但是，已写的高速缓存未命中数据除外）。从含高速缓存未命中数据的4个字（8字节），按顺序以环绕方式读数据。在读取1个高速缓存块的数据时，CPU能进行下一个处理。在读完1个高速缓存块的数据时，注册物理地址中的特征位，并且将V位和U位置1。为了保持最新的置换通路，更新LRU位。此后，将回写缓冲器的数据回写到外部存储器。

7. 高速缓存未命中（直写）

按照指定的存取长度，将数据写到对应虚拟地址的外部存储器此时不写高速缓存，并且也不更新特征位、V位、U位和LRU位。

8.3.4 回写缓冲器

在因高速缓存未命中而需要将已写的高速缓存的入口清除到外部存储器时，为了优先高速缓存的读数据并且提高性能，SH-4A 内置了用于保存要清除的高速缓存块数据的回写缓冲器。回写缓冲器由 1 个高速缓存块的数据和清除目标的物理地址构成。

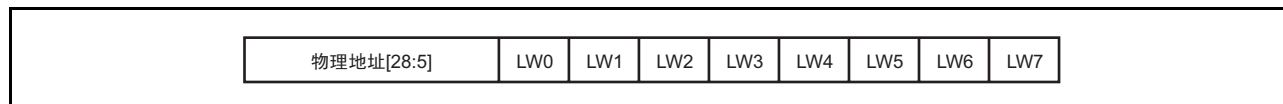


图 8.3 回写缓冲器的结构

8.3.5 直写缓冲器

对于直写方式时的写操作以及对不能缓存的区域的写操作，SH-4A 内置了用于保存写数据的 64 位缓冲器。在完成直写缓冲器的写操作时，CPU 不等待完成外部存储器的写操作而转移到下一个操作。

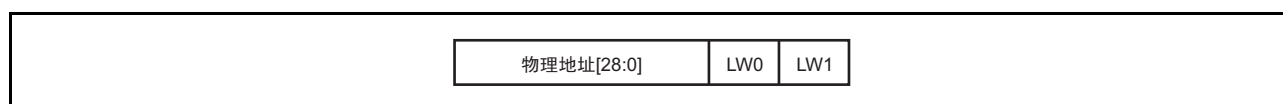


图 8.4 直写缓冲器的结构

8.3.6 OC 2 通路模式

如果将 RAMCR 寄存器的 OC2W 位置 1，就变为只使用 OC 的通路 0 和通路 1 的 OC 2 通路模式，能降低功耗。本模式也包含存储器映像 OC 的存取，只使用通路 0 和通路 1。

必须通过 P2 区的程序进行 OC2W 位的改写，如果在改写时已在 OC 中注册了有效块，就必须在改写 OC2W 位之前，根据需要通过软件进行回写，然后将 CCR 寄存器的 OCI 位置 1，使 OC 的所有入口变为无效。

8.4 指令高速缓存的操作说明

8.4.1 读操作

在指令高速缓存 (IC) 有效 (CCR.ICE=1) 并且从能缓存的区域取指令时, IC 进行以下运行:

1. 从由虚拟地址的位 [12:5] 指向的各路高速缓存块, 读取特征位、V 位和 LRU 位。
2. 将由 MMU 进行虚拟地址转换后的物理地址的位 [28:10] 和从各路读取的特征位进行比较,
 - 当特征位相同并且存在 V 位为 1 的通路时 →3
 - 当特征位相同而不存在 V 位为 1 的通路时 →4
3. 高速缓存命中
从命中通路的数据部, 将由虚拟地址的位 [4:3] 指向的数据作为指令进行读取。为了保持最新的命中通路, 更新 LRU 位。
4. 高速缓存未命中
从对应虚拟地址的物理地址空间, 将数据读到由 LRU 位选择的置换对象通路的高速缓存块。从包含高速缓存未命中数据的 4 个字 (8 字节), 按顺序以环绕方式读数据, 当该数据到达高速缓存时, 将读取的数据作为指令返回给 CPU。在读取剩余的 1 个高速缓存块的数据时, CPU 能进行下一个处理。在读完 1 个高速缓存块的数据时, 注册物理地址中的特征位, 并且将 V 位置 1。为了保持最新的置换通路, 更新 LRU 位。

8.4.2 预取操作

在指令高速缓存 (IC) 有效 (CCR.ICE=1) 并且从能缓存的区域将指令预取到 IC 时, IC 进行以下运行:

1. 从由虚拟地址的位 [12:5] 指向的各路高速缓存块, 读取特征位、V 位和 LRU 位。
2. 将由 MMU 进行虚拟地址转换后的物理地址的位 [28:10] 和从各路读取的特征位进行比较,
 - 当特征位相同并且存在 V 位为 1 的通路时 →3
 - 当特征位相同而不存在 V 位为 1 的通路时 →4
3. 高速缓存命中
为了保持最新的命中通路, 更新 LRU 位。
4. 高速缓存未命中
从对应虚拟地址的物理地址空间, 将数据读到置换对象通路的高速缓存块。从包含高速缓存未命中数据的 4 个字 (8 字节), 按顺序以环绕方式进行数据的读入。在预取操作中, CPU 不等待数据的到达而能在读取 1 个高速缓存块的数据时进行下一个处理。在读完 1 个高速缓存块的数据时, 注册物理地址中的特征位, 并且将 V 位置 1。为了保持最新的置换通路, 更新 LRU 位。

8.4.3 IC 2 通路模式

如果将 RAMCR 寄存器的 IC 2W 位置 1, 就变为只使用 IC 的通路 0 和通路 1 的 IC 2 通路模式, 能降低功耗。本模式也包含存储器映像 IC 的存取, 只使用通路 0 和通路 1。

必须通过 P2 区的程序进行 IC2W 位的改写, 如果在改写时已在 IC 中注册了有效块, 就必须在改写 IC2W 位之前, 将 CCR 寄存器的 ICI 位置 1, 使 IC 的所有入口变为无效。

8.4.4 指令高速缓存的通路预测

为了降低功耗, SH-4A 内置了指令高速缓存 (IC) 的通路预测机构, 只启动预测到的通路的数据阵列。在发生通路预测错误时, 重新读取正确的通路数据, 所以会降低取指令的性能。当将 ICWPD 位置 1 时, 就能停止 IC 通路预测机构。因为在本模式中不发生通路预测错误, 所以不会降低取指令的性能, 但是会增加 IC 的功耗。另外, 必须通过在不能缓存的 P2 区中运行的程序, 进行 ICWPD 位的转换。如果在改写时已在 IC 中注册了有效块, 就必须在改写 ICWPD 位之前, 将 CCR 寄存器的 ICI 位置 1, 使 IC 的所有入口变为无效。

8.5 高速缓存操作指令

8.5.1 高速缓存和外部存储器的一致性

(1) 高速缓存操作指令一览

必须通过软件保证高速缓存和外部存储器的一致性。SH-4A 支持操作高速缓存的以下 6 种指令，各指令的详细内容请参照《SH-4A 扩展功能软件手册》的“第 11 章 各指令的说明”。

- 操作数高速缓存的无效指令：OCBI @Rn
使操作数高速缓存无效（无回写）。
- 操作数高速缓存的清除指令：OCBP @Rn
使操作数高速缓存无效（有回写）。
- 操作数高速缓存的回写指令：OCBWB @Rn
进行操作数高速缓存的回写。
- 操作数高速缓存的分配指令：MOVCA.L R0,@Rn
确保操作数高速缓存。
- 指令高速缓存的无效指令：ICBI @Rn
使指令高速缓存无效。
- 操作数存取同步指令：SYNCO
等待数据传送的结束。

(2) 一致性控制

为了控制操作数高速缓存的一致性，能接受来自 SuperHyway 总线的 PURGE 和 FLUSH 事务。由 PURGE/FLUSH 事务提供的地址为物理地址，因此在允许 MMU 时，为了避免高速缓存的同义问题，不能使用 1K 字节的页。

- PURGE 事务
在允许操作数高速缓存时，检索操作数高速缓存，使命中的入口无效。如果无效块为已写状态，就将数据回写到外部存储器。未命中时为无操作。
- FLUSH 事务
在允许操作数高速缓存时，检索操作数高速缓存，如果存在命中的入口并且为已写状态，就将数据回写到外部存储器。不使命中的入口无效。在未命中时或者在命中的入口不为已写状态时为无操作。

(3) 有关一致性控制的指令规格的变更

在操作数高速缓存的操作指令中，有关 OCBI、OCBP 和 OCBWB 的一致性控制的规格变更是在处理器版本寄存器（PVR）的 VER 位的值为 H'20 的 SH-4A 时的规格。

1. 无效指令：OCBI@Rn 的变更点

当 Rn 指示的地址是非高速缓存区时，如果处理器版本寄存器（PVR）的 VER 位的值为 H'20 的 SH-4A，本指令就作为 NOP 指令执行。但是，对于具备扩展功能的 SH-4A，只在 Rn[31:24] 为 H'F4（OC 地址阵列区）时，使通路=Rn[14:13] 和入口=Rn[12:5] 指向的操作数高速缓存块无效。此时，即使无效块为已写状态也不进行回写处理。本操作只能在特权模式中执行，在用户模式中会发生地址错误异常。另外，不产生 TLB 的有关异常。

不能对 Rn[31:24] ≠ H'F4 的存储器映像阵列区、控制寄存器区和保留区（H'F0～H'F3、H'F5～H'FF）执行本指令。

2. 清除指令：OCBP @Rn 的变更点

当 Rn 指示的地址为非高速缓存区时，如果处理器版本寄存器（PVR）的 VER 位的值为 H'20 的 SH-4A，本指令就作为 NOP 指令执行。但是，对于具备扩展功能的 SH-4A，只在 Rn[31:24] 为 H'F4（OC 地址阵列区）时，使通路 =Rn[14:13] 和入口 =Rn[12:5] 指向的操作数高速缓存块无效。此时，如果无效块为已写状态就进行回写处理。本操作只能在特权模式中执行，在用户模式中会发生地址错误异常。另外，不产生 TLB 的有关异常。

不能对 Rn[31:24] ≠ H'F4 的存储器映像阵列区、控制寄存器区和保留区（H'F0 ~ H'F3、H'F5 ~ H'FF）执行本指令。

3. 回写指令：OCBWB @Rn 的变更点

当 Rn 指示的地址为非高速缓存区时，如果处理器版本寄存器（PVR）的 VER 位的值为 H'20 的 SH-4A，本指令就作为 NOP 指令执行。但是，对于具备扩展功能的 SH-4A，只在 Rn[31:24] 为 H'F4（OC 地址阵列区）时，如果通路 =Rn[14:13] 和入口 =Rn[12:5] 指向的操作数高速缓存块为已写状态，就进行回写处理，并且将已写位置 0 状态。本操作只能执行在特权模式，在用户模式中会发生地址错误异常。另外，不产生 TLB 的有关异常。

不能对 Rn[31:24] ≠ H'F4 的存储器映像阵列区、控制寄存器区和保留区（H'F0 ~ H'F3、H'F5 ~ H'FF）执行本指令。

8.5.2 预取操作

为了减少因高速缓存未命中而发生高速缓存填数的补偿，SH-4A 支持预取指令。如果知道因读写操作而发生高速缓存未命中时，通过预取指令预先将数据填充到高速缓存，能防止在读写操作时发生高速缓存未命中，从而提高软件的性能。对已保存在高速缓存中的数据执行预取指令时、或者准备预取的地址在 UTLB 中未命中时或者发生违反保护时，为无操作，不发生异常。有关预取指令的详细内容，请参照《SH-4A 扩展功能软件手册》的“第 11 章 各指令的说明”。

- 预取指令（OC）：PREF @Rn
- 预取指令（IC）：PREFI @Rn

8.6 存储器映像高速缓存的结构

为了通过软件管理 IC 和 OC，能在特权模式时用 MOV 指令从 P2 区的程序读写 IC 数据阵列的内容，并且还能在特权模式时用 MOV 指令从 P2 区或者 IL 存储区的程序读写 IC 地址阵列的内容。不保证从其他区域程序的存取。此时，必须通过以下 1 ~ 3 中的任意一种方法转移到 P0、U0、P1 或者 P3 区：

1. 必须通过 RTE 指令进行转移。
2. 必须在对任意地址（也包含不能缓存的区域）执行 ICBI 指令后，才能转移到 P0、U0、P1 或者 P3 区。
3. 如果在存取存储器映像 IC 前预先设定 IRMCR.MC=0（初始值）时，就不需要执行特定的指令。但是，必须注意：此方法需要重新取存储器映像 IC 存取指令的下一条指令，所以会降低 CPU 的处理性能。

其中，方法 3 有可能不保证今后的 SuperH 系列。为了保证今后 SuperH 系列的兼容性，建议使用方法 1 或者方法 2。

能在特权模式时用 MOV 指令从 P1 或者 P2 区的程序读写 OC 的内容。不保证从其他区域程序的存取。IC 和 OC 分配到虚拟地址空间的 P4 区。对 IC 的地址阵列 / 数据阵列和 OC 的地址阵列 / 数据阵列只能进行数据存取，存取长度固定为长字。不能对此区域进行取指令。必须将保留位置 0，保留位的读取值为不定值。

8.6.1 IC 地址阵列

IC 的地址阵列分配到 P4 区的 H'F000 0000 ~ H'FOFF FFFF。在存取地址阵列时，需要指定 32 位的地址部（读写时）和 32 位的数据部。地址部指定要存取的通路和入口，数据部指定要写的特征位和 V 位。

地址部中的 [31:24] 为 HF0，表示 IC 地址阵列；[14:13] 指定通路，[12:5] 指定入口。地址部 [3] 的联想位（A 位）指定在写 IC 地址阵列时是否进行联想。长度固定为长字，所以地址部的 [1:0] 必须指定 0。

数据部中的 [31:10] 表示特征位，[0] 表示 V 位。IC 地址阵列的特征位为 19 位，在不进行联想写时不使用数据部的 [31:29]，只在进行联想写时数据部的 [31:29] 用于指定虚拟地址。

能对 IC 地址阵列进行以下 3 种操作。

1. IC 地址阵列 读

从地址部设定的通路和入口所对应的 IC 入口，将特征位和 V 位读到数据部。读取时，无论地址部指定的联想位是 1 还是 0，都不进行联想操作。

2. IC 地址阵列 写（无联想）

对地址部设定的通路和入口所对应的 IC 入口，写数据部指定的特征位和 V 位。必须将地址部的 A 位置 0。

3. IC 地址阵列 写（有联想）

在地址部的 A 位为 1 时，将保存在地址部指定的入口中的各路特征位与数据部指定的特征位进行比较判断。不使用地址部 [14:13] 的通路号。此时，如果允许 MMU，就通过 ITLB 将数据部 [31:10] 指定的虚拟地址转换为物理地址，然后进行比较判断。如果地址相同并且此通路的 V 位为 1，就给 IC 的入口写数据部指定的 V 位；否则为无操作。本操作用于 IC 特定入口的无效处理。如果在地址转换时发生 ITLB 未命中或者在比较判断时出现不相同，就不产生异常而为无操作，并且不进行写操作。

【注】 今后的 SuperH 系列可能不支持 IC 地址阵列的联想写功能。建议采用 ITLB 未命中处理或者指令 TLB 未命中异常的通知，使用确实可进行 IC 操作的 ICBI 指令。

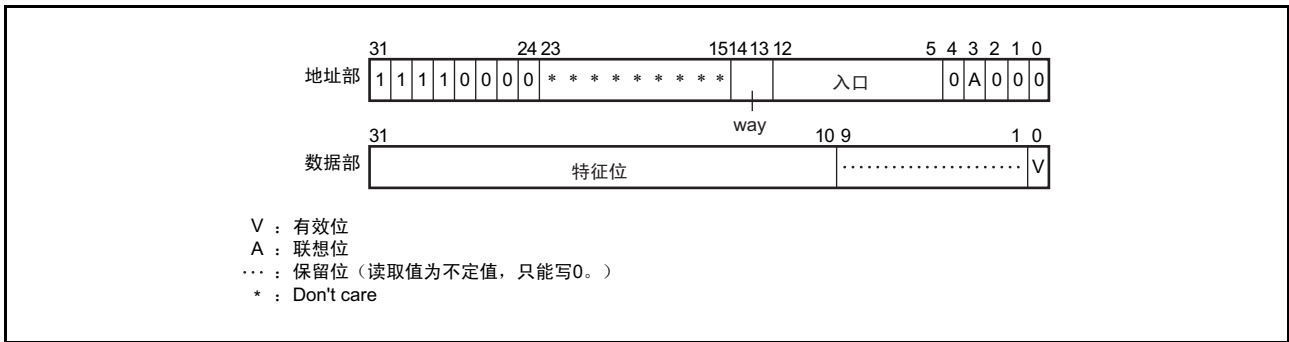


图 8.5 存储器映像 IC 地址阵列 (高速缓存容量 =32K 字节时)

8.6.2 IC 数据阵列

IC 的数据阵列分配到 P4 区的 H'F100 0000 ~ H'F1FF FFFF。在存取数据阵列时，需要指定 32 位的地址部 (读写时) 和 32 位的数据部。地址部指定要存取的通路和入口，数据部指定要写的长字数据。

地址部中的 [31:24] 为 H'F1，表示 IC 数据阵列；[14:13] 指定通路，[12:5] 指定入口。地址部的 [4:2] 用于指定入口中的长字数据。存取长度固定为长字，所以地址部的 [1:0] 必须指定 0。

数据部用于指定长字数据。

能对 IC 数据阵列进行以下 2 种操作：

1. IC 数据阵列 读

在地址部设定的通路和入口所对应的 IC 入口中，从地址部的长字指定位所指定的数据，将长字数据读到数据部。

2. IC 数据阵列 写

在地址部设定的通路和入口所对应的 IC 入口中，对地址部的长字指定位所指定的数据，写数据部指定的长字数据。

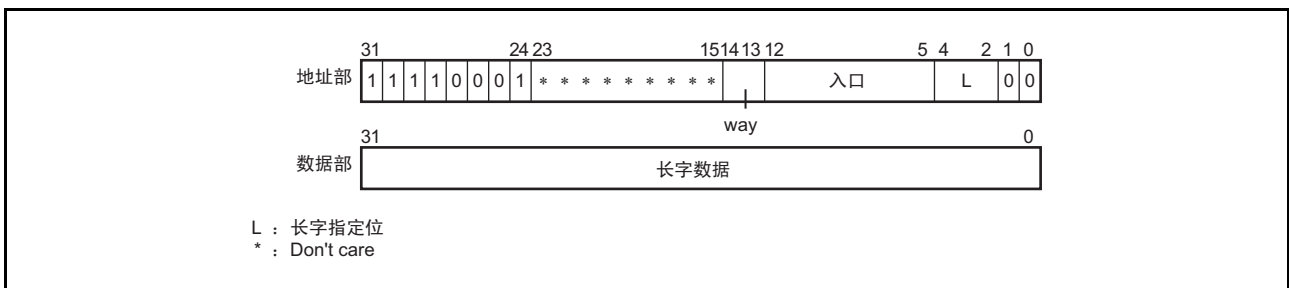


图 8.6 存储器映像 IC 数据阵列 (高速缓存容量 =32K 字节时)

8.6.3 OC 地址阵列

OC 的地址阵列分配到 P4 区的 H'F400 0000 ~ H'F4FF FFFF。在存取地址阵列时，需要指定 32 位的地址部 (读写时) 和 32 位的数据部。地址部指定要存取的通路和入口，数据部指定要写的特征位、U 位和 V 位。

地址部中的 [31:24] 为 H'F4，表示 OC 地址阵列；[14:13] 指定通路，[12:5] 指定入口。地址部 [3] 的联想位 (A 位) 指定在写 OC 地址阵列时是否进行联想。存取长度固定为长字，所以地址部的 [1:0] 必须指定 0。

数据部中的 [31:10] 表示特征位，[1] 表示 U 位，[0] 表示 V 位。OC 地址阵列的特征位为 19 位，在不进行联想写时不使用数据部的 [31:29]，只在进行联想写时数据部的 [31:29] 用于指定虚拟地址。

能对 OC 地址阵列进行以下 3 种操作：

1. OC地址阵列 读

从地址部设定的通路和入口所对应的OC入口，将特征位、U位和V位读到数据部。读取时，无论地址部指定的联想位是1还是0，都不进行联想操作。

2. OC地址阵列 写（无联想）

对地址部设定的通路和入口所对应的OC入口，写数据部指定的特征位、U位和V位。必须将地址部的A位置0。

在写U位和V位都为1的高速缓存块时，在回写此高速缓存块后，写数据部指定的特征位、U位和V位。

3. OC地址阵列 写（有联想）

在地址部的A位为1时，将保存在地址部指定的入口中的各路特征位与数据部指定的特征位进行比较判断。不使用地址部[14:13]的通路号。此时，如果允许MMU，就通过UTLB将数据部[31:10]指定的虚拟地址转换为物理地址，然后进行比较判断。如果地址相同并且此通路的V位为1，就给OC的入口写数据部指定的U位和V位；否则为无操作。本操作用于OC特定入口的无效处理。如果在此时OC入口的U位为1并且将V位或者U位置0，就进行回写。如果在地址转换时发生UTLB未命中或者在比较判断时出现不相同，就不产生异常而变为无操作，并且不进行写操作。

【注】 今后的 SuperH 系列可能不支持 OC 地址阵列的联想写功能。建议采用数据 TLB 未命中异常的通知，使用确实可进行 OC 操作的 OCB1、OCBP 和 OCBWB 指令。

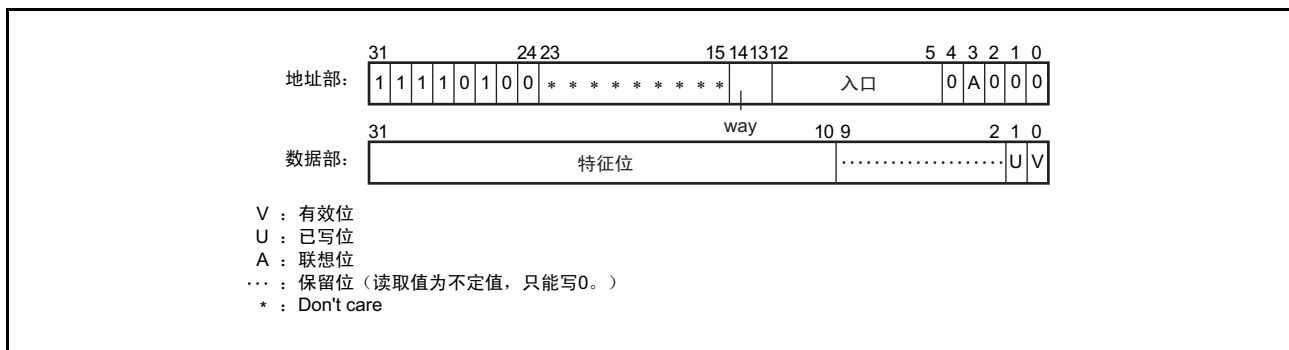


图 8.7 存储器映像 OC 地址阵列（高速缓存容量 = 32K 字节时）

8.6.4 OC 数据阵列

OC 的数据阵列分配到 P4 区的 H'F500 0000 ~ H'F5FF FFFF。在存取数据阵列时，需要指定 32 位的地址部（读写时）和 32 位的数据部。地址部指定要存取的通路和入口，数据部指定要写的长字数据。

地址部中的 [31:24] 为 H'F5，表示 OC 数据阵列；[14:13] 指定通路，[12:5] 指定入口。地址部的 [4:2] 用于指定入口中的长字数据。存取长度固定为长字，所以地址部的 [1:0] 必须指定 0。

数据部用于指定长字数据。

能对 OC 数据阵列进行以下 2 种操作：

1. OC数据阵列 读

在地址部设定的通路和入口所对应的OC入口中，从地址部的长字指定位所指定的数据，将长字数据读到数据部。

2. OC数据阵列 写

在地址部设定的通路和入口所对应的OC入口中，对地址部的长字指定位所指定的数据，写数据部指定的长字数据。此写操作不会使地址阵列侧的U位变为1。

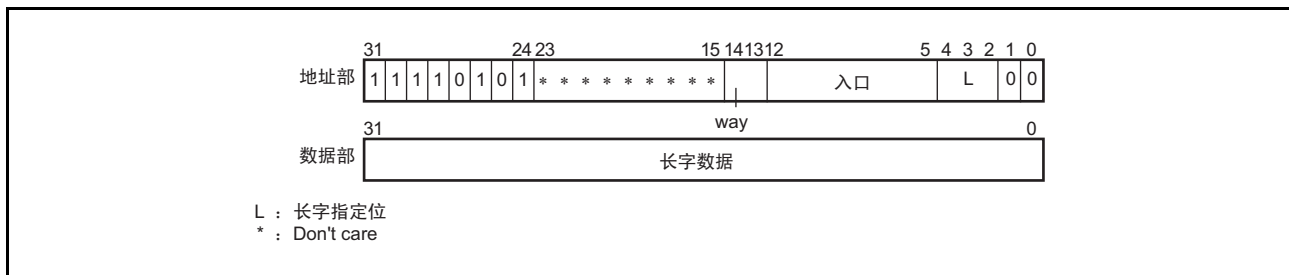


图 8.8 存储器映像 OC 数据阵列（高速缓存容量 =32K 字节时）

8.6.5 存储器映像的联想写操作

今后的 SuperH 系列可能不支持 IC 地址阵列和 OC 地址阵列的联想写。建议采用 ITLB 未命中处理、指令 TLB 未命中异常的通知和数据 TLB 未命中异常的通知，使用确实可进行 IC 和 OC 操作的 ICBI、OCBI、OCBP 和 OCBWB 指令。在作为过渡性措施而使用本功能时，SH-4A 会产生地址错误。在重视与以前产品的兼容性时，通过将 EXPMASK 寄存器（H'FF2F 0004）的 MMCAW 位置 1，能使用本功能。但是，为了保证今后 SuperH 系列的兼容性，必须使用 ICBI、OCBI、OCBP 和 OCBWB 指令。

8.7 存储队列

为了高速写外部存储器，SH-4A 支持 32 字节 ×2 的存储队列（SQ）。

8.7.1 SQ 的结构

SQ 如图 8.9 所示，由 32 字节的 SQ0 和 32 字节的 SQ1 构成，SQ0 和 SQ1 可分别独立设定。

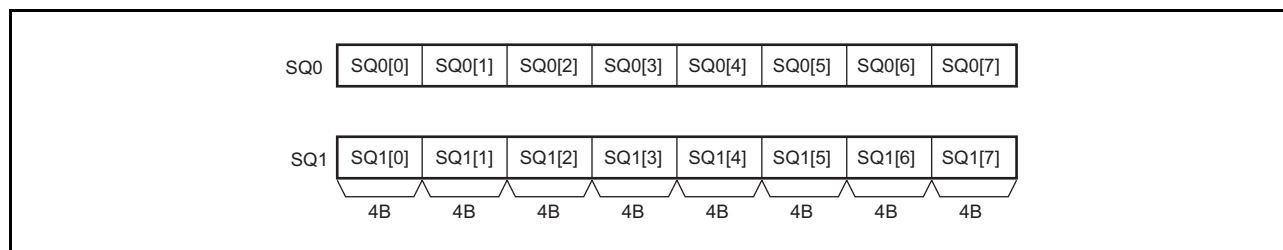


图 8.9 存储队列的结构

8.7.2 写 SQ

能通过存储指令对 P4 区的 H'E000 0000 ~ H'E3FF FFFC 进行存取，实现 SQ 的写操作。存取长度为长字或者 4 个字。此地址的含义如下：

[31:26]	:111000	指定存储队列
[25:6]	:Don't care	用于外部存储器的传送和存取权
[5]	:0/1	0: 指定 SQ0 1: 指定 SQ1
[4:2]	:LW 指定	指定 SQ0 和 SQ1 中的长字位置
[1:0]	:00	固定为 0

8.7.3 传送到外部存储器

能通过预取指令（PREF）将 SQ 传送到外部存储器。通过对 P4 区的 H'E000 0000 ~ H'E3FF FFFC 发行 PREF 指令，开始将 SQ 传送到外部存储器。传送长度固定为 32 字节，起始地址必须为 32 字节边界。在将一方的 SQ 传送到外部存储器的过程中，能在不需要补偿周期的情况下实现另一方 SQ 的写操作。但是，在传送到外部存储器的过程中，SQ 的写操作需要等到外部存储器的传送结束才能执行。

SQ 的传送目标的物理地址 [28:0] 根据允许 / 禁止 MMU 进行以下指定：

1. 允许 MMU（MMUCR.AT=1）的情况

给 UTLB 的 VPN 设定 SQ 区（H'E000 0000 ~ H'E3FF FFFF），给 PPN 设定传送目标的物理地址。ASID、V、SZ、SH、PR 和 D 位与通常地址转换的含义相同，但是，对于此页 C 和 WT 位无意义。

如果给 SQ 区发行预取指令，就进行地址转换，根据 SZ 位的指定，生成物理地址的 [28:10]。物理地址的 [9:5] 与禁止 MMU 的情况相同，从地址转换前的地址生成。物理地址的 [4:0] 固定为 0。对此地址进行从 SQ 到外部存储器的传送。

2. 禁止 MMU（MMUCR.AT=0）的情况

给发行 PREF 指令的地址指定 SQ 区（H'E000 0000 ~ H'E3FF FFFF）。此地址 [31:0] 的含义如下：

[31:26]	:111000	指定存储队列
[25:6]	:地址	传送目标的物理地址 [25:6]
[5]	:0/1	0: 指定 SQ0 1: 指定 SQ1 并且传送目标的物理地址 [5]
[4:2]	:Don't care	预取时无意义。
[1:0]	:00	固定为 0

不能从上述地址生成的物理地址 [28:26] 从 QACR0 和 QACR1 生成。

QACR0[4:2] : 对应 SQ0 的物理地址 [28:26]

QACR1[4:2] : 对应 SQ1 的物理地址 [28:26]

由于突发传送的开始为 32 字节边界，所以物理地址 [4:0] 总是固定为 0。

8.7.4 SQ 存取异常判断

写 SQ 和传送（PREF 指令）到外部存储器时的异常判断根据允许 / 禁止 MMU 进行以下操作。如果在写 SQ 时发生异常，SQ 的内容就保持原来的值；在将 SQ 传送到外部存储器时发生异常，就禁止对外部存储器的传送。

1. 允许 MMU（MMUCR.AT=1）的情况

根据注册在 UTLB 的地址转换信息和 SQMD 位进行操作。在写 SQ 时进行写类型的异常判断，在将 SQ 传送（PREF 指令）到外部存储器时进行读类型的异常判断，产生 TLB 未命中异常或者保护违反异常。但是，当只允许在特权模式中根据 SQMD 位存取 SQ 时，即使在用户模式中地址转换成功也发生地址错误。

2. 禁止 MMU（MMUCR.AT=0）的情况

根据 SQMD 位进行。

0: 能进行特权/用户存取

1: 能进行特权存取

在 SQMD 位为 1 时，如果在用户模式中存取 SQ 区，就发生地址错误。

8.7.5 读 SQ

在特权模式时，SH-4A 能通过加载指令对 P4 区的 H'FF00 1000 ~ H'FF00 103C 进行存取，实现 SQ 的读操作。存取长度固定为长字。

31:6] : H'FF00 1000 : 指定存储队列

[5] : 0/1 0: 指定 SQ0 1: 指定 SQ1

[4:2] : LW 指定 指定 SQ0 和 SQ1 中的长字位置

[1:0] : 00 固定为 0

第 9 章 内部存储器

本 LSI 内置 IL 存储器。IL 存储器适用于保存指令。

9.1 特点

(1) IL 存储器

- 容量：
IL 存储器的容量为 16K 字节。
- 页：
IL 存储器分为 4 页（页 0～页 3）。
- 存储器映像：
IL 存储器和虚拟地址空间、物理地址空间一起分配到表 9.1 所示的地址。

表 9.1 IL 存储器地址

页	存储器容量
	16K 字节
页 0	H'E520 0000 ~ H'E520 0FFF
页 1	H'E520 1000 ~ H'E520 1FFF
页 2	H'E520 2000 ~ H'E520 2FFF
页 3	H'E520 3000 ~ H'E520 3FFF

- 端口：
具有 3 个独立的读写端口，与 SuperHyway 总线、高速缓存/RAM 内部总线和指令总线连接。取指令时使用指令总线，存取操作数时使用高速缓存/RAM 内部总线，从 SuperHyway 总线主模块存取时使用 SuperHyway 总线。
- 优先顺序：
在不同的总线同时对相同的页发出存取请求时，根据优先顺序进行存取处理。优先顺序按由高到低的顺序为 SuperHyway 总线、高速缓存/RAM 内部总线、指令总线。

9.2 寄存器说明

有关内部存储器的寄存器如下：

表 9.2 寄存器结构

名称	略称	R/W	P4 区地址 *	区域 7 地址 *	长度
内部存储器的控制寄存器	RAMCR	R/W	H'FF00 0074	H'1F00 0074	32

【注】 * P4 区地址是使用虚拟地址空间 P4 区的地址，区域 7 地址是使用 TLB 存取物理地址空间区域 7 的地址。

表 9.3 各处理模式的寄存器状态

名称	略称	上电复位	睡眠	待机
内部存储器的控制寄存器	RAMCR	H'0000 0000	保持	保持

9.2.1 内部存储器的控制寄存器（RAMCR）

RAMCR 控制内部存储器的保护功能。

位名:	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
位名:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RMD	RP	IC2W	OC2W	ICWPD	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R

位	位名	初始值	R/W	说明
31 ~ 10	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。
9	RMD	0	R/W	内部存储器的存取模式位 指定从虚拟地址空间存取内部存储器的存取权。 0: 能进行特权存取（用户存取时发生地址错误异常） 1: 能进行用户 / 特权存取
8	RP	0	R/W	内部存储器的保护有效位 在从虚拟地址空间存取内部存储器时，选择使用 ITLB 和 UTLB 的保护功能。 0: 不使用保护功能 1: 使用保护功能 详细内容请参照“9.4 内部存储器的保护功能”。
7	IC2W	0	R/W	IC 2 通路模式位 详细内容请参照“8.4.3 IC 2 通路模式”。
6	OC2W	0	R/W	OC 2 通路模式位 详细内容请参照“8.3.6 OC 2 通路模式”。
5	ICWPD	0	R/W	IC 通路预测禁止位 详细内容请参照“8.4.4 指令高速缓存的通路预测”。
4 ~ 0	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。

9.3 运行说明

9.3.1 从 CPU 取指令的存取

从 CPU 进行取指令的存取时，通过虚拟地址从指令总线直接存取。在从指令总线连续存取 IL 存储器的相同页并且不发生竞争时，需要 1 个周期。

9.3.2 从 CPU 的操作数存取和从 FPU 的存取

从 CPU 的操作数存取和从 FPU 的存取通过高速缓存 /RAM 内部总线进行。从高速缓存 /RAM 内部总线的存取需要多个周期。

【注】 能通过 PC 相对 (@disp.pc) 进行操作数的存取。

9.3.3 从 SuperHyway 总线主模块的存取

从 DMAC 等 SuperHyway 总线主模块存取内部存储器时，通过物理地址总线的 SuperHyway 总线进行。必须使用与虚拟地址相同的地址。

9.4 内部存储器的保护功能

SH-4A 对于内部存储器，使用内部存储器的控制寄存器 RAMCR 的内部存储器存取模式位 (RMD) 和内部存储器保护有效位 (RP)，实现以下保护功能：

- 对于从 CPU 和 FPU 的存取的保护功能

在 RAMCR.RMD=0 时，将对用户模式中内部存储区的存取判断为地址错误异常。

在 MMUCR.AT=1 并且 RAMCR.RP=1 时，除了地址错误异常的判断以外，P4 区的部分内部存储区与 P0/P3/U0 区相同，被判断为 MMU 异常。

将上述汇总于表 9.4。

表 9.4 对内部存储器的存取，由保护功能产生的异常

MMUCR.AT	RAMCR.RP	SR.MD	RAMCR.RMD	必定产生的异常	可能产生的异常	
0	X	0	0	地址错误异常	—	
			1	—	—	
		1	X	—	—	
1	0	0	0	地址错误异常	—	
			1	—	—	
		1	X	—	—	
	1	0	0	0	地址错误异常	—
				1	—	MMU 异常
		1	X	—	MMU 异常	

【符号说明】 x:Don't care

9.5 使用时的注意事项

9.5.1 页竞争

在不同的总线同时对相同的页发出存取请求时，发生页竞争。各存取能正常结束，但是这样的竞争会导致存储器存取性能的降低。因此，建议尽可能通过软件不使竞争发生。例如，在各总线存取各不相同的页时，不会发生竞争。

9.5.2 页转换

在从指令总线连续存取相同的页时，需要 1 个周期。但是，在跨页存取或者存取从 IL 存储器以外的区域切换到 IL 存储器时，需要多个周期（最多 2 个等待周期）。因此，从性能优化的观点考虑，建议在从指令总线存取时，通过软件避免频繁的页切换。例如，每页分配独立的程序，就能提高效率。

9.5.3 一致性

在将指令配置于 IL 存储器时，必须在给 IL 存储器写指令后执行以下指令，然后转移到改写后的指令。

- SYNCO
- ICBI @Rn

此时，ICBI 指令的对象可以是在不发生地址错误异常的范围中的任意地址（也可以是 IL 存储器的地址），高速缓存命中 / 未命中均可。

9.5.4 睡眠模式

在睡眠模式中，不能从 DMAC 等 SuperHyway 总线主模块存取 IL 存储器。

第 10 章 时钟振荡器 (CPG)

时钟振荡器 (CPG) 生成提供给本 LSI 的内置外围模块和外部总线接口的时钟并且进行低功耗的控制。时钟振荡器由晶体振荡电路、PLL 电路、分频电路和控制部构成。

10.1 特点

- 生成本 LSI 的内部时钟
生成 CPU、FPU、高速缓冲器、TLB 使用的 CPU 时钟 (Ick)、SuperHyway 使用的 SHwy 时钟 (SHck) 以及提供给内部外围模块的外围时钟 (Pck)。
- 生成本 LSI 的外部时钟
生成外部总线接口使用的总线时钟 (Bck)。
- 时钟模式
作为 CPG 的时钟输入, 能选择晶体振荡器输入或者外部时钟输入。
能从 2 种时钟运行模式选择上电复位后的 CPU 时钟、SHwy 时钟、总线时钟、外围时钟的倍频率组合。
- 低功耗模式的控制
能停止睡眠模式和刷新待机模式的时钟以及模块待机功能的特定模块。

CPG 的框图如图 10.1 所示。

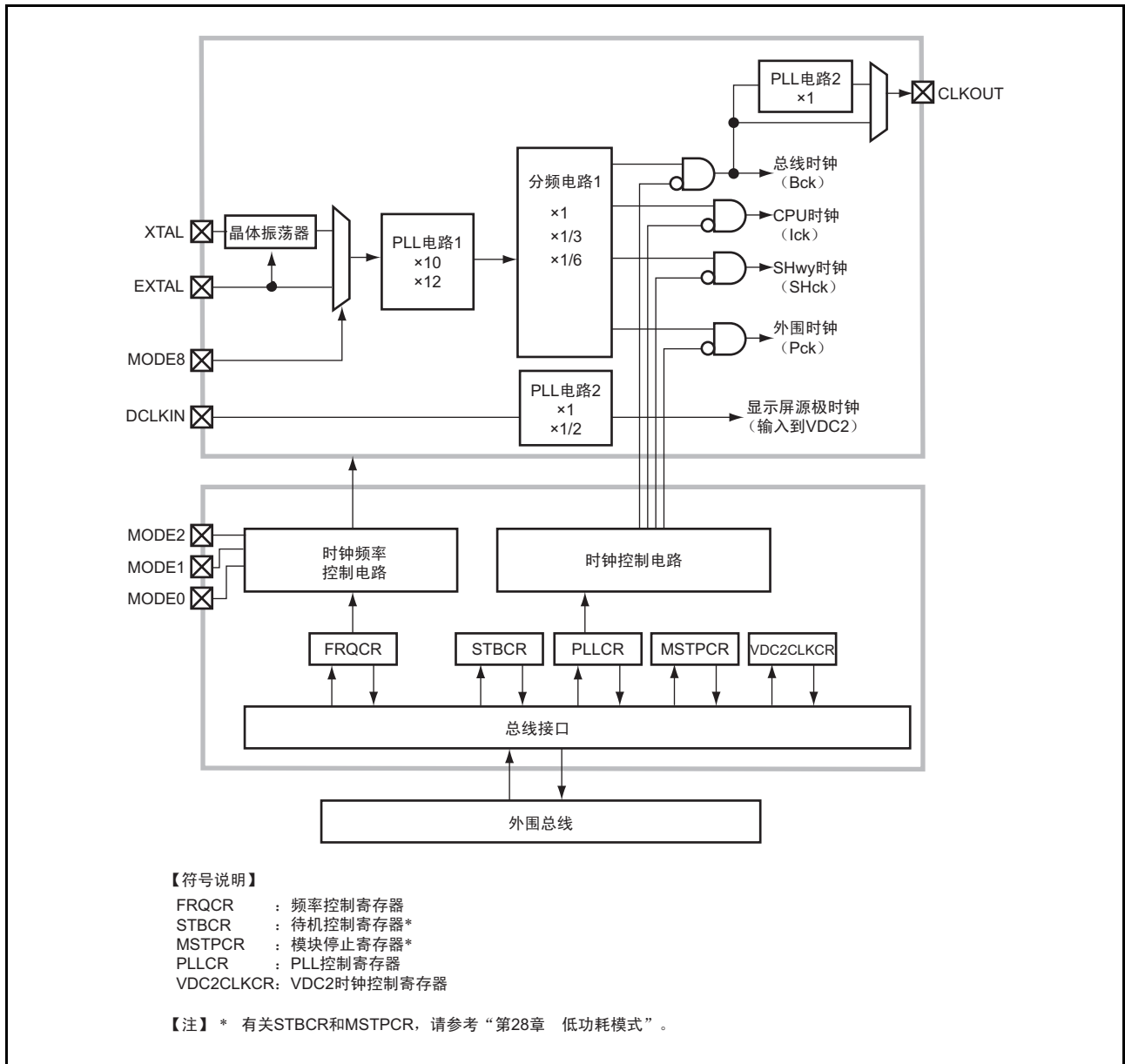


图 10.1 CPG 框图

CPG 的各模块具有以下功能：

(1) PLL 电路 1

PLL 电路 1 具有将晶体振荡电路或者 EXTAL 引脚输入的时钟频率递增 10 倍或者 12 倍的功能。时钟的倍频率取决于模式引脚 MODE0、MODE1 和 MODE2 的组合。

(2) PLL 电路 2

PLL 电路 2 具有将总线时钟相位 (Bck) 和外部外围接口使用的 CLKOUT 时钟相位进行匹配的功能。

(3) 晶体振荡电路

这是晶体谐振器连接 XTAL 和 EXTAL 引脚使用时的振荡电路。
通过设定 MODE8 引脚，能使用晶体振荡电路。

(4) 分频电路 1

分频电路 1 生成 CPU 时钟 (Ick)、SHwy 时钟 (SHck)、外围时钟 (Pck) 和总线时钟 (Bck)。

(5) 频率控制寄存器 (FRQCR)

能通过频率控制寄存器确认 Ick、SHck、Pck、Bck 的分频率，是只读寄存器。

(6) PLL 控制寄存器 (PLLCR)

给 PLL 控制寄存器分配了 CLKOUT 输出的 ON/OFF 控制位。

(7) 模块停止寄存器 (MSTPCR)

给模块停止寄存器分配了控制各外围模块的运行 / 停止位。有关此寄存器的规格请参照“第 28 章 低功耗模式”。

(8) 待机控制寄存器 (STBCR)

给待机控制寄存器分配了待机位。有关此寄存器的规格请参照“第 28 章 低功耗模式”。

10.2 输入 / 输出引脚

CPG 的引脚结构如表 10.1 所示。

表 10.1 CPG 的引脚结构和功能

引脚名	功能	输入 / 输出	说明
MODE0 MODE1 MODE2	模式控制引脚 0、1、2 时钟运行模式	输入	设定上电复位后的时钟运行模式。
MODE8	模式控制引脚 8 时钟输入模式	输入	设定是否使用晶体振荡器。 当 MODE8=L 时，从 EXTAL 输入外部时钟。 当 MODE8=H 时，晶体振荡器连接 EXTAL 和 XTAL。
XTAL	时钟引脚	输出	连接晶体振荡器。
EXTAL		输入	连接晶体振荡器或者用作外部时钟输入引脚。
CLKOUT		输出	用作外部总线时钟输出引脚。

【注】 有关 CLKOUT 的 AC 时序保证，请参照电特性章节。必须注意晶体振荡电路的输入频率与倍频率的关系。

10.3 时钟运行模式

模式控制引脚 (MODE0、MODE1 和 MODE2) 的组合与上电复位后的时钟运行模式的关系如表 10.2 所示。

表 10.2 时钟运行模式

时钟运行模式	外部引脚的组合 *1			PLL1	PLL2	EXTAL 频率 (MHz)		CPG 生成的时钟				FRQCR 初始值
	MODE2	MODE1	MODE0					lck	SHck	Bck	Pck	
2	0	1	0	ON	ON	24 ~ 32.4	频率比 *2	10	10/3	10/3	10/6	H'30320044
							最大频率	324	108	108	54	
3	0	1	1	ON	ON	24 ~ 27	频率比 *2	12	4	4	2	H'40320044
							最大频率	324	108	108	54	

【注】 *1 禁止设定上述以外的模式引脚 MODE0、MODE1 和 MODE2 的组合。

*2 这是晶体振荡电路或者 EXTAL 引脚输入的时钟频率为 1 时的各时钟频率比。

10.4 寄存器说明

CPG 的寄存器结构如表 10.3 所示, 各处理模式的寄存器状态如表 10.4 所示。

表 10.3 寄存器结构

名称	略称	R/W	P4 区地址	区域 7 地址	存取长度
频率控制寄存器	FRQCR	R	H'FFC8 0000	H'1FC8 0000	32
PLL 控制寄存器	PLLCR	R/W	H'FFC8 0024	H'1FC8 0024	32
VDC2 时钟控制寄存器	VDC2CLKCR	R/W	H'FFC8 0004	H'1FC8 0004	32

表 10.4 各处理模式的寄存器状态

名称	略称	上电复位	睡眠	待机
频率控制寄存器	FRQCR	H'x032 0044*	保持	保持
PLL 控制寄存器	PLLCR	H'0000 E001	保持	保持
VDC2 时钟控制寄存器	VDC2CLKCR	H'0000 0080	保持	保持

【注】 * 通过时钟运行模式 (由 MODE0、MODE1 和 MODE2 设定) 设定的初始值。

10.4.1 频率控制寄存器 (FRQCR)

FRQCR 是 32 位只读寄存器，能确认 CPU 时钟 (Ick)、SHwy 时钟 (SHck)、外围时钟 (Pck) 和总线时钟 (Bck) 的上电复位后的分频率。频率比请参照表 10.2 时钟运行模式表。FRQCR 只能进行长字存取，而不能保证写操作。

此寄存器只在 PRESET 引脚或者 WDT 上溢产生的上电复位时进行复位。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	CFC[2:0]		—	BFC[2:0]			
初始值:	—	—	—	—	0	0	0	0	0	0	1	1	0	0	1	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
	—	—	—	—	—	—	—	—	—	P0FC[2:0]		—	P1FC[2:0]			
初始值:	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
31 ~ 28	—	不定值	R	保留位 当时钟运行模式为模式 2 时读取值为 0011；为模式 3 时读取值为 0100。
27 ~ 23	—	00000	R	保留位 读取值为 0。
22 ~ 20	CFC[2:0]	011	R	CPU 时钟 (Ick) 和 SHwy 时钟 (SHck) 的分频率 Ick SHck 011: 1 倍 1/3 倍
19	—	0	R	保留位 读取值为 0。
18 ~ 16	BFC[2:0]	010	R	总线时钟 (Bck) 的分频率 010: ×1/3 倍
15 ~ 7	—	000000000	R	保留位 读取值为 0。
6 ~ 4	PFC[2:0]	100	R	外围时钟 (Pck) 的分频率 100: ×1/6 倍
3	—	0	R	保留位 读取值为 0。
2 ~ 0	—	100	R	保留位 读取值为 100。

10.4.2 PLL 控制寄存器 (PLLCR)

PLLCR 是 32 位可读写寄存器，指定是否从 CLKOUT 引脚输出时钟。PLLCR 只能进行长字存取。

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CKOFF	CKONE
初始值:	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 16	—	全 0	R	保留位 写的值必须和初始值相同。
15 ~ 13	—	全 1	R	保留位 写的值必须和初始值相同。
12 ~ 2	—	全 0	R	保留位 写的值必须和初始值相同。
1	CKOFF	0	R/W	CLKOUT 输出停止位 0: 从 CLKOUT 引脚输出时钟。 1: 停止 CLKOUT 引脚的时钟输出。引脚状态为 Hi-Z。
0	CKONE	1	R/W	时钟输出允许位 在软件待机时，设定 CLKOUT 引脚是输出时钟还是固定为低电平。 0: 固定为低电平 1: 输出时钟

10.4.3 VDC2 时钟控制寄存器 (VDC2CLKCR)

VDC2CLKCR 是 32 位可读写寄存器，能选择 VDC2 时钟。VDC2CLKCR 只能进行长字存取。

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
	—	—	—	—	—	—	—	—	CKSEL	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
31 ~ 8	—	全 0	R	保留位 读取值为 0，只能写 0。
7	CKSEL	1	R/W	显示屏源极时钟的分频率选择位 指定从 DCLKIN 引脚输入并提供给 VDC2 模块的显示屏源极时钟的分频率。 0: DCLKIN×1/1 1: DCLKIN×1/2
6 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

第 11 章 存储器控制单元 (MCU)

存储器控制单元 (MCU) 是仲裁 CPU 和各种模块的存取以及输出 SRAM 和 SDARM 接口控制信号的模块。通过本模块能直接连接 SRAM、ROM 和 SDRAM 等。

本模块具有 SuperHyway 总线接口 (SHIF)、Pixel 总线接口 (PXIF)、LCD 控制器直接总线接口 (LCDIF)、SRAM 控制器 (LBSC)、SDRAM 控制器 (SBSC) 以及对接口模块对各控制器的存取进行仲裁的仲裁器 (ARBT)。

11.1 特点

- 支持外部存储器的存取
 - 输出 4 个外部存储器的选择信号
 - 4 个区域 (FLASH 和 SDRAM)，分别最大为 64M 字节的外部存储空间
- SRAM: 能选择 32 位/16 位/8 位的数据总线
- SDRAM: 能选择 64 位/32 位的数据总线
- 能设定大端法或者小端法

【SRAM 接口】

- 能连接 NOR 型的闪存
- 周期等待功能: (通过硬件进行的等待控制: 信号)
- 避免数据总线冲突的等待控制 (插入空闲周期):
 - 读周期之间的等待设定
 - 读周期和写周期之间的等待设定

【SDRAM 接口】

- 刷新功能:
 - 自动刷新 (内置可编程刷新计数器)
 - 自刷新
- 时序设定:
 - 行列等待、列等待、行激活周期、
 - 写恢复周期、行预充电周期、自动刷新请求间隔、
 - 初始化预充电周期数、初始化自动刷新请求间隔
- 突发存取方式: 随机列 (SDRAM 突发长度: 32 位总线时为 8, 64 位总线时为 4)
- 初始化定序器功能: 发行预充电和自动刷新命令

本存储器控制器的框图如图 11.1 所示。

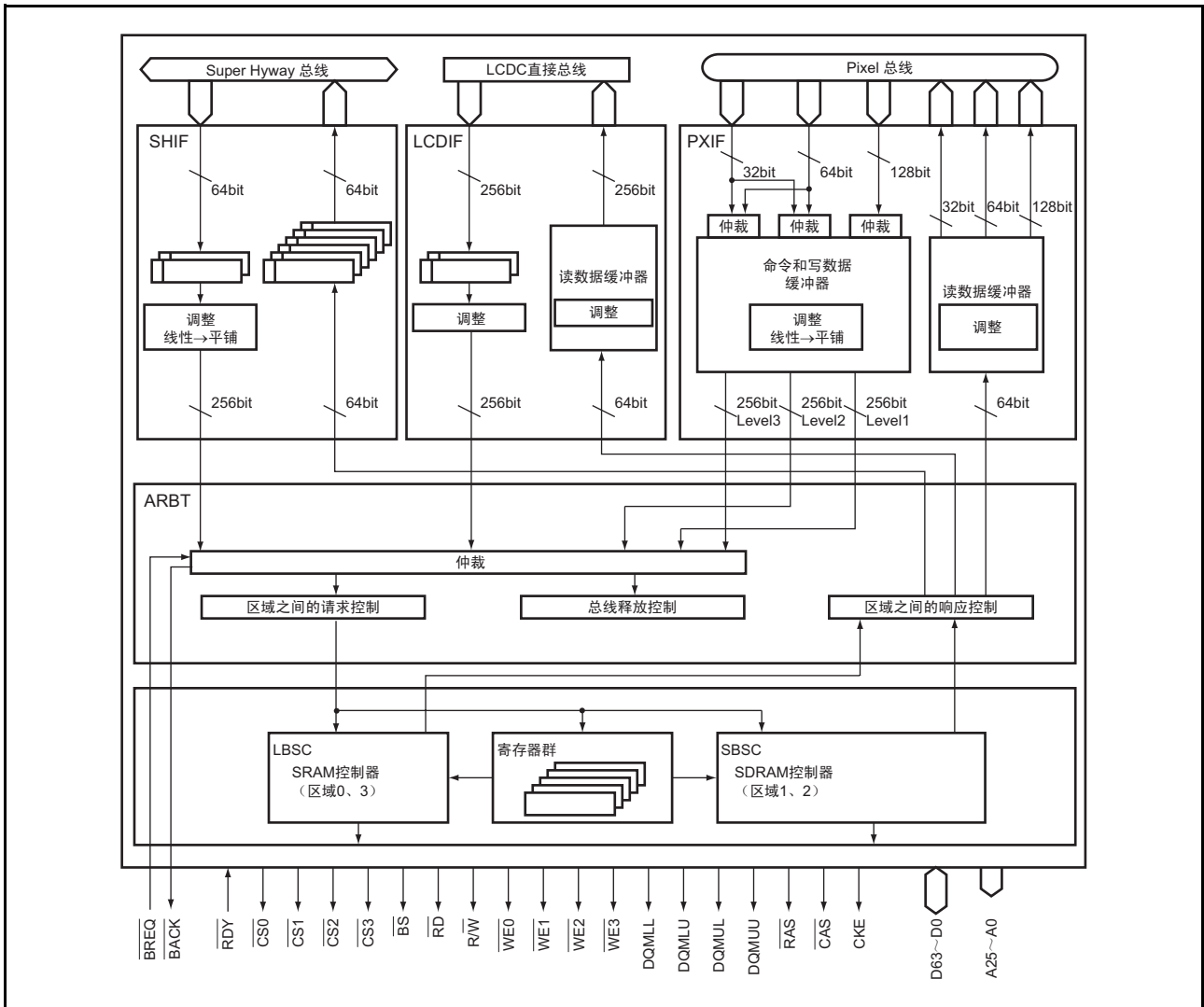


图 11.1 MCU 框图

1. **SHIF (SuperHyway Interface)**: 是CPU-SRAM和SDRAM之间的接口, 用SuperHyway协议进行接口连接, 总线宽度为64位。SuperHyway总线是分离事务的总线, 能进行有效的数据传送。所谓分离事务, 就是将通信分为请求包和响应包进行传送的方式。如果使用此方式, 在一个通信结束前不会连续占有总线。
2. **PXIF (Pixel Bus Interface)**: 是连接要存取统一存储器 (SDRAM) 的外围模块的接口块。每个模块有独立的端口, 各信号用Pixel总线协议进行一对一连接。支持32位、64位和128位的总线宽度。
3. **LCDIF (LCD controller bus interface)**: 是连接LCDC的总线。只连接LCDC, 是专用的总线协议。总线宽度为256位, 能存取的区域只为SDRAM空间。
4. **ARBT (Arbiter)**: 进行另外规定的优先顺序判断, 并且进行SHIF、PXIF、LCDIF各接口之间的仲裁。
5. **LBSC (SRAM Local Bus Controller)**: 控制对SRAM的读写存取。

LBSC有以下特点:

- 管理外部存储空间中的区域0和区域3
 - 存储容量: 各区域最大为64M字节
 - 能设定8/16/32位的总线 (区域0通过外部引脚设定, 区域3通过寄存器设定)
 - 能通过RDY引脚进行等待状态的插入
 - 能通过程序控制等待状态的插入
 - 在连续存取存储器时, 用于避免数据总线冲突的等待周期自动插入功能
 - 为了连接低速存储器, 能在写周期中插入写选通的准备时间和保持时间
6. **SBSC (SDRAM Bus Controller)**: 控制对SDRAM的读写存取。根据SDRAM的时序规定进行命令的发行和读写数据的发送和接收。对SDRAM支持32位和64位的总线, 突发长度在32位时为8, 在64位时为4, 原则上以32字节进行传送。支持自动刷新和自刷新。在设定为存储体打开模式时, 命中行地址时的数据传送为连续的突发传送。

11.2 输入 / 输出引脚

MCU 的引脚结构如表 11.1 所示。

表 11.1 引脚结构

引脚名	功能	输入 / 输出	说明
A25 ~ A0	地址总线	输出	地址输出
D63 ~ D32	数据总线	输入 / 输出	数据输入 / 输出 (和其他引脚多路复用)
D31 ~ D0	数据总线	输入 / 输出	数据输入 / 输出
\overline{BS}	总线周期开始	输出	表示总线周期开始的信号
$\overline{CS3} \sim \overline{CS0}$	片选	输出	表示存取区域的片选信号
\overline{RD}	读	输出	外部设备的读信号
$\overline{R/W}$	读 / 写	输出	数据总线的输入 / 输出方向指示信号 存取 SDRAM 时为 WE 信号。
\overline{RAS}	行地址选通	输出	SDRAM RAS 信号
\overline{CAS}	列地址选通	输出	SDRAM CAS 信号
CKE	时钟允许	输出	SDRAM 时钟允许信号
DQMLL	数据屏蔽	输出	SDRAM D7 ~ D0 数据屏蔽信号
DQMLU	数据屏蔽	输出	SDRAM D15 ~ D8 数据屏蔽信号
DQMUL	数据屏蔽	输出	SDRAM D23 ~ D16 数据屏蔽信号
DQMUU	数据屏蔽	输出	SDRAM D31 ~ D24 数据屏蔽信号
$\overline{WE0}$	数据允许 0	输出	存取 SRAM 时, 为对应 D7 ~ D0 的写选通信号; 设定 SDRAM 接口时, 为 D39 ~ D32 的数据屏蔽信号 (高电平有效)。
$\overline{WE1}$	数据允许 1	输出	存取 SRAM 时, 为对应 D15 ~ D8 的写选通信号; 设定 SDRAM 接口时, 为 D47 ~ D40 数据屏蔽信号 (高电平有效)。
$\overline{WE2}$	数据允许 2	输出	存取 SRAM 时, 为对应 D23 ~ D16 的写选通信号; 设定 SDRAM 接口时, 为 D55 ~ D48 的数据屏蔽信号 (高电平有效)。
$\overline{WE3}$	数据允许 3	输出	存取 SRAM 时, 为对应 D31 ~ D24 的写选通信号; 设定 SDRAM 接口时, 为 D63 ~ D56 的数据屏蔽信号 (高电平有效)。
\overline{RDY}	就绪	输入	等待状态请求信号
\overline{BREQ}	总线释放请求	输入	总线释放请求信号
BACK	总线允许使用 / 总线返回请求	输出	总线使用的允许信号 / 总线的返回请求信号
MODE3 MODE4	区域 0 总线宽度	输入	上电复位时, 为区域 0 的总线宽度设定信号。
MODE5	字节排序转换	输入	上电复位时, 为字节排序的设定信号。
$\overline{DACK0^*}$	DMAC0 应答信号	输出	DMAC 通道 0 的数据应答
$\overline{DACK1^*}$	DMAC1 应答信号	输出	DMAC 通道 1 的数据应答
$\overline{DTEND0^*}$	DMAC0 传送结束信号	输出	DMAC 通道 0 的传送结束
$\overline{DTEND1^*}$	DMAC1 传送结束信号	输出	DMAC 通道 1 的传送结束

【注】 * 能通过 DMAC 的 CHCR0 和 CHCR1 的 AL 位选择 $\overline{DACK0}$ 、 $\overline{DACK1}$ 、 $\overline{DTEND0}$ 、 $\overline{DTEND1}$ 引脚的极性 (初始值为低电平有效)。

11.3 区域的概要

11.3.1 空间分割

作为体系结构，本 LSI 有 32 位虚拟地址空间。通过高位地址的值将虚拟地址分为 5 个区域。另外，外部存储空间有 29 位地址空间，分为 4 个区域。

虚拟地址能通过地址转换单元 (MMU) 分配到任意的外部地址。详细内容请参照“第 7 章 存储器管理单元 (MMU)”。本章节记述了有关外部地址的区域分割。

如表 11.2 所示，本 LSI 能将 SRAM 和 SDRAM 连接到外部地址的 4 个区域。

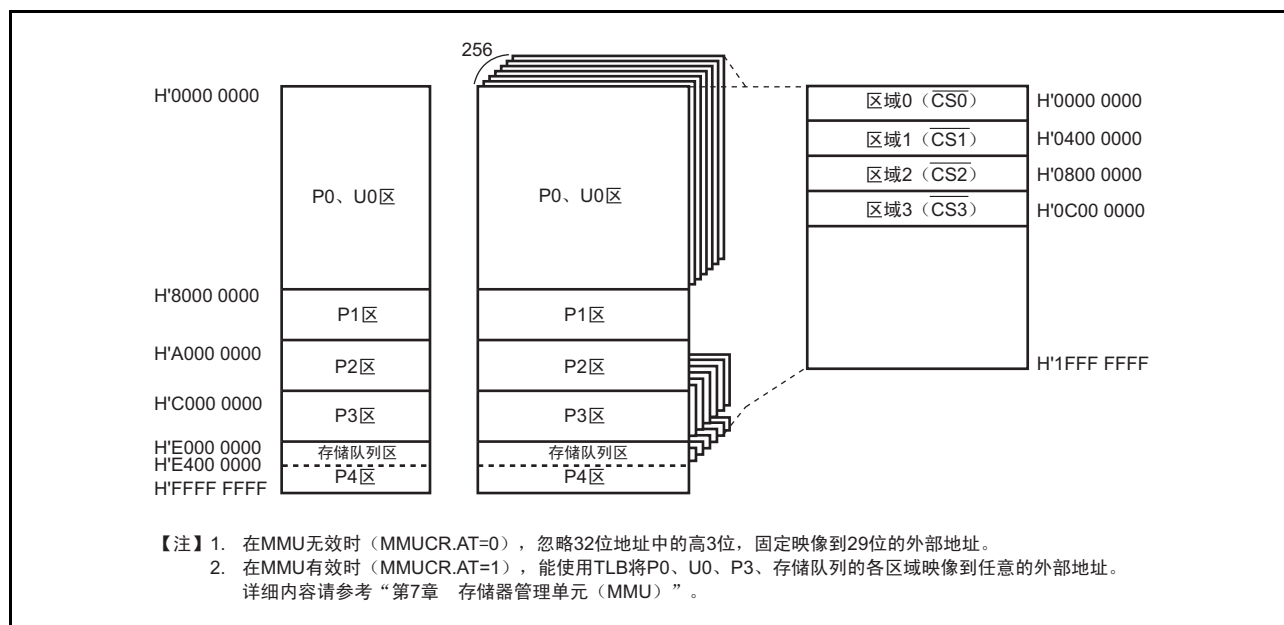


图 11.2 虚拟地址空间和外部存储空间的对应

表 11.2 外部存储空间的映像

区域	外部地址	容量	可连接的存储器	可连接的总线宽度 (位)	存取长度
0	H'0000 0000 ~ H'03FF FFFF	64M 字节	SRAM	8、16、32*1	8、16、32
1	H'0400 0000 ~ H'07FF FFFF	64M 字节	SDRAM	32、64*2	32、64
2	H'0800 0000 ~ H'0BFF FFFF	64M 字节	SDRAM	32、64*2	32、64
3	H'0C00 0000 ~ H'0FFF FFFF	64M 字节	SRAM	8、16、32*2	8、16、32

【注】 *1 通过外部引脚指定存储器的总线宽度。

*2 通过寄存器指定存储器的总线宽度。

11.3.2 存储器的总线宽度

存储器总线宽度的设定方法因空间而不同。区域 0 能在通过 **PRESET** 引脚进行上电复位时，使用外部引脚选择 8 位、16 位或者 32 位的总线。外部引脚 (MODE4 和 MODE3) 和总线宽度的关系如表 11.3 所示。

表 11.3 区域 0 存储器的总线宽度的设定

MODE4	MODE3	总线宽度
0	0	保留
0	1	8 位
1	0	16 位
1	1	32 位

区域 1、2 能通过存储器接口寄存器 (MIM) 选择 32 位或者 64 位的总线 (区域 1 和区域 2 共用总线宽度)。详细内容请参照“11.4.2 存储器接口模式寄存器 (MIM)”。

区域 3 能通过 CS3 总线控制寄存器 (CS3BCR) 选择 8 位、16 位或者 32 位的总线。详细内容请参照“11.4.15 CS3 总线控制寄存器 (CS3BCR)”。

11.3.3 字节排序的设定

字节排序的设定取决于外部引脚 (MODE5) 的状态，区域 0、3 和区域 1、2 共用此字节排序的设定。外部引脚 (MODE5) 和字节排序的关系如表 11.4 所示。

表 11.4 字节排序的设定

MODE5	字节排序
0	大端法
1	小端法

11.4 寄存器说明

寄存器结构如表 11.5 所示。这些寄存器是 64 位寄存器，但是必须以长字（32 位）进行存取。在写数据时，按长字单位反映状态；在读数据时，能参照在进行存取时的长字值。存取 bit63 ~ 32 时必须指定地址 8n+0，存取 bit31 ~ 0 时必须指定地址 8n+4。通过这些寄存器控制各种存储器的接口和等待状态等。

表 11.5 寄存器结构

地址	寄存器名	略称	初始值	存取
H'FF80 0000	版本控制寄存器	VCR	H'0B04 0000 0000 0000	32
H'FF80 0008	存储器接口模式寄存器	MIM	H'0000 0000 061A 0x40	32
H'FF80 0010	SDRAM 控制寄存器	SCR	H'0000 0000 0000 0000	32
H'FF80 0018	SDRAM 时序寄存器	STR	H'0000 0000 00FF FFE7	32
H'FF80 0030	SDRAM 行属性寄存器	SDRA	H'0000 0000 0000 0200	32
H'FFAx xxxx	SDRAM 模式寄存器	SDMR	—	32
H'FF80 0200	仲裁模式寄存器	AMR	H'0000 0000 0400 0000	32
H'FF80 0100	线性平铺转换控制寄存器	LTC0	H'0000 0000 0000 0000	32
H'FF80 0108	线性平铺转换区起始地址寄存器	LTAD0	H'0000 0000 0000 0000	32
H'FF80 0110	线性平铺转换区起始地址屏蔽寄存器	LTAM0	H'0000 0000 0000 0000	32
H'FF80 0118	线性平铺转换控制寄存器	LTC1	H'0000 0000 0000 0000	32
H'FF80 0120	线性平铺转换区起始地址寄存器	LTAD1	H'0000 0000 0000 0000	32
H'FF80 0128	线性平铺转换区起始地址屏蔽寄存器	LTAM1	H'0000 0000 0000 0000	32
H'FF80 0130	线性平铺转换控制寄存器	LTC2	H'0000 0000 0000 0000	32
H'FF80 0138	线性平铺转换区起始地址寄存器	LTAD2	H'0000 0000 0000 0000	32
H'FF80 0140	线性平铺转换区起始地址屏蔽寄存器	LTAM2	H'0000 0000 0000 0000	32
H'FF80 0148	线性平铺转换控制寄存器	LTC3	H'0000 0000 0000 0000	32
H'FF80 0150	线性平铺转换区起始地址寄存器	LTAD3	H'0000 0000 0000 0000	32
H'FF80 0158	线性平铺转换区起始地址屏蔽寄存器	LTAM3	H'0000 0000 0000 0000	32
H'FF80 0160	线性平铺转换控制寄存器	LTC4	H'0000 0000 0000 0000	32
H'FF80 0168	线性平铺转换区起始地址寄存器	LTAD4	H'0000 0000 0000 0000	32
H'FF80 0170	线性平铺转换区起始地址屏蔽寄存器	LTAM4	H'0000 0000 0000 0000	32
H'FF80 0178	线性平铺转换控制寄存器	LTC5	H'0000 0000 0000 0000	32
H'FF80 0180	线性平铺转换区起始地址寄存器	LTAD5	H'0000 0000 0000 0000	32
H'FF80 0188	线性平铺转换区起始地址屏蔽寄存器	LTAM5	H'0000 0000 0000 0000	32
H'FF80 0190	线性平铺转换控制寄存器	LTC6	H'0000 0000 0000 0000	32
H'FF80 0198	线性平铺转换区起始地址寄存器	LTAD6	H'0000 0000 0000 0000	32
H'FF80 01A0	线性平铺转换区起始地址屏蔽寄存器	LTAM6	H'0000 0000 0000 0000	32
H'FF80 01A8	线性平铺转换控制寄存器	LTC7	H'0000 0000 0000 0000	32
H'FF80 01B0	线性平铺转换区起始地址寄存器	LTAD7	H'0000 0000 0000 0000	32
H'FF80 01B8	线性平铺转换区起始地址屏蔽寄存器	LTAM7	H'0000 0000 0000 0000	32
H'FF80 0218	请求屏蔽设定寄存器	RQM	H'0000 0000 0000 0000	32
H'FF80 1000	总线控制寄存器	BCR	H'0000 0000 3800 0000	32
H'FF80 2000	CS0 总线控制寄存器	CS0BCR	H'0000 0000 7777 7x80	32
H'FF80 2008	CS0 等待控制寄存器	CS0WCR	H'0000 0000 7777 770F	32
H'FF80 2030	CS3 总线控制寄存器	CS3BCR	H'0000 0000 7777 7380	32
H'FF80 2038	CS3 等待控制寄存器	CS3WCR	H'0000 0000 7777 770F	32

【注】 必须以指定的存取长度进行存取。

11.4.1 版本控制寄存器 (VCR)

位:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	1	0	1	1	0	0	0	0	0	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
位:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	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
	—	—	—	—	—	—	DRAM_SELFREF	DRAM_INACTIVE	—	—	BAD_OPC	—	—	—	ERR_SNT	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R	R	R	R/W	R

位	位名	初始值	R/W	说明
63 ~ 10	—	0 or 1	R	保留位 读取值为上述位图中的值，只能写 0。
9	DRAM_SELFREF	0	R/W	在通过 MIM 寄存器的 RMODE、DRE 和 DCE 位设定为自刷新模式时，如果存取数据块的区域，此位就被置 1。通过写 0 此位被清 0。
8	DRAM_INACTIVE	0	R/W	在通过 MIM 寄存器的 DCE 位使 SDRAM 控制器无效时，如果存取数据块的区域，此位就被置 1。通过写 0 此位被清 0。
7、6	—	全 0	R	保留位 读值取为 0，只能写 0。
5	BAD_OPC	0	R/W	在从 SuperHyway 接受到本存储器控制器不支持的请求时，此位被置 1。通过写 0 此位被清 0。
4 ~ 2	—	全 0	R	保留位 读取值为 0，只能写 0。
1	ERR_SNT	0	R/W	在从存储器控制器将错误响应返回给 SuperHyway 时，此位被置 1。通过写 0 此位被清 0。
0	—	0	R	保留位 读取值为 0，只能写 0。

11.4.2 存储器接口模式寄存器 (MIM)

位:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	BOMODE[1:0]	—	PCKE	—	—	—	—	—	—	—	—	—	—	SELFS	RMODE	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R	R	R	R	R	R	R	R	R	R	R/W	R
位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	DR[11:0]											
初始值:	0	0	0	0	0	1	1	0	0	0	0	1	1	0	1	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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DRE	ENDIAN	BW[1:0]	—	—	—	—	—	—	DCE
初始值:	0	0	0	0	0	0	0	*	0	1	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R	R/W	R/W	R	R	R	R	R	R/W

位	位名	初始值	R/W	说明
63 ~ 48	—	全 0	R	保留位 读取值为 0，只能写 0。
47、46	BOMODE [1:0]	00	R/W	此位是可读写位，进行 SDRAM 存取模式的转换。本存储器控制器支持 2 种 SDRAM 存取模式。有关各模式的运行说明，请参照“11.7.7 存储体打开模式”。 00: BANK 打开模式 01: BANK 关闭模式 10、11: 禁止设定
45	—	0	R	保留位 读取值为 0，只能写 0。
44	PCKE	0	R/W	不在存取 SDRAM 时（空闲和存储体有效状态时），通过此位将 CKE 置为低电平，设定为掉电模式。在将此位置 1 时，本功能有效。能通过本功能抑制 SDRAM 的功耗。
43 ~ 35	—	全 0	R	保留位 读取值为 0，只能写 0。
34	SELFS	0	R	能通过此位判断 SDRAM 是否处于自刷新状态。 0: 非自刷新状态 1: 自刷新中
33	RMODE	0	R/W	此位是可读写位，指定是进行自动刷新还是进行自刷新。只在 DRE 位为 1 时此位的指定才有效。 0: 自动刷新 1: 自刷新
32 ~ 28	—	全 0	R	保留位 读取值为 0，只能写 0。

位	位名	初始值	R/W	说明
27 ~ 16	DRI[11:0]	H'61A	R/W	<p>DRI (DRAM Refresh Interval) 位</p> <p>在刷新有效 (DRE 位为 1) 时, 通过此位能指定最大刷新闻隔 (自动刷新)。1 个计数与 1 个存储器时钟周期相同。</p> <p>100MHz 相当于 10ns。通过此寄存器能设定的最小值为 H'020, 如果设定值小于 H'020, 就给计数值加上 H'020。</p> <p>存储器控制器模块的内部有 12 位计数器, 在 DCE 位为 0 或者 DRE 位为 0 或者 RMODE 位为 1 时, 此计数器被清 0; 否则, 通过外部存储器时钟进行累加计数。此计数器与 DRI 位比较, 如果相同, 就在控制器内部产生自动刷新请求, 进行自动刷新。同时, 计数器被清 0, 并再次进行累加计数。在内部产生的自动刷新请求最多被记录 1 次, 如果 DCE 位为 1、DRE 位为 1 并且 RMODE 位为 0, 就在进行自动刷新前不会清除自动刷新请求。在设定此位时, 必须在将 DRE 位置 0 后进行此位的写操作, 然后将 DRE 位置 1。此时, 必须给 DRI 位置前一次的设定值。</p> <p>* 在总线释放时, 如果计数到设定值的二分之一, 就产生刷新请求。</p>
15 ~ 10	—	全 0	R	<p>保留位</p> <p>读取值为 0, 只能写 0。</p>
9	DRE	0	R/W	<p>DRE (DRAM Refresh Enable) 位</p> <p>通过此位控制刷新功能有效或者无效。</p> <p>0: 无效</p> <p>1: 有效</p>
8	ENDIAN	*	R	<p>此位表示外部数据总线是大端法还是小端法。</p> <p>0: 小端法</p> <p>1: 大端法</p> <p>此位的写无效。</p>
7、6	BW[1:0]	01	R/W	<p>BW (Bus Width) 位</p> <p>通过此位指定 SDRAM 总线的宽度。</p> <p>01: 32 位宽度</p> <p>10: 64 位宽度</p> <p>00、11: 禁止设定</p>
5 ~ 1	—	全 0	R	<p>保留位</p> <p>读取值为 0, 只能写 0。</p>
0	DCE	0	R/W	<p>DCE (DRAM 控制器允许) 位</p> <p>允许通过存储器控制器进行 SDRAM 控制。此位为 1 时, 允许 SDRAM 控制; 次位为 0 时, 存储器控制器对 SuperHyway 的请求返回错误响应。因此, 在 SDRAM 运行过程中, 必须总是将 DCE 位置 1。</p>

【注】 SDRAM 控制寄存器的设定同时适用于区域 1 和区域 2, 不能分别设定区域 1 和区域 2。

11.4.3 SDRAM 控制寄存器 (SCR)

位:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	SMS[2:0]		
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
63 ~ 3	—	全 0	R	保留位 读取值为 0，只能写 0。
2 ~ 0	SMS[2:0]	000	R/W	SMS (SDRAM Mode Select) 位 在上电时和复位解除后，通过此位进行 SDRAM 初始化。 在通过软件设定此位时，发行以下命令。 初始化步骤请参照“11.7.9 SDRAM 初始化顺序”。 写一次寄存器就发行一条命令。 000: 进行正常运行。 001: 发行 NOP 命令。(只在 MIM.DCE=1 时有效) 010: 发行 PALL 命令。(只在 MIM.DCE=1 时有效) 011: 允许 CKE。此时，发行 DESELECT 命令。 (只在 MIM.DCE=1 时有效) 100: 发行 CBR (自动) 刷新命令。 (只在 MIM.DCE=1 时有效) 禁止上述以外的设定，否则不保证运行。

11.4.4 SDRAM 时序寄存器 (STR)

位:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	WR[2:0]			RW[2:0]			SWR[1:0]	
初始值:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SRFC[2:0]			SRAS[2:0]			SRP[1:0]		SRC[2:0]			SCL[2:0]			SRCD	—
初始值:	1	1	1	1	1	1	1	1	1	1	1	0	0	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

位	位名	初始值	R/W	说明
63 ~ 24	—	全 0	R	保留位 读取值为 0，只能写 0。
23 ~ 21	WR[2:0]	111	R/W	指定给 SDRAM 发行 WRITE 命令和发行 READ 命令之间的最少周期数。 000: 4 个周期 001: 5 个周期 010: 6 个周期 011: 7 个周期 100: 8 个周期 101: 9 个周期 110: 10 个周期 111: 11 个周期
20 ~ 18	RW[2:0]	111	R/W	指定给 SDRAM 发行 READ 命令和发行 WRITE 命令之间的最少周期数。 000: 6 个周期 001: 7 个周期 010: 8 个周期 011: 9 个周期 100: 10 个周期 101: 11 个周期 110: 12 个周期 111: 13 个周期
17、16	SWR[1:0]	11	R/W	指定写数据时的最终的后同步码和发行 PRE/PALL 命令之间的周期数 (Twr)。 00: 2 个周期 01: 3 个周期 10: 4 个周期 11: 5 个周期

位	位名	初始值	R/W	说明
15 ~ 13	SRFC[2:0]	111	R/W	对相同存储体之间的下述存取时间，指定规定的周期数 (Trfc)。 (1) 从自动刷新到发行 ACT 命令 (2) 从自动刷新到自动刷新 000: 8 个周期 001: 9 个周期 010: 10 个周期 011: 11 个周期 100: 12 个周期 101: 13 个周期 110: 14 个周期 111: 15 个周期
12 ~ 10	SRAS[2:0]	111	R/W	在相同存储体之间，指定发行 ACT 命令和发行 PRE 命令之间的最少周期数 (Tras)。 000: 7 个周期 001: 8 个周期 010: 9 个周期 011: 10 个周期 100: 11 个周期 101: 12 个周期 110: 13 个周期 111: 14 个周期
9、8	SRP[1:0]	11	R/W	指定发行 PRE 命令和发行 ACT 命令之间的周期数 (Trp)。 00: 2 个周期 01: 3 个周期 10: 4 个周期 11: 5 个周期
7 ~ 5	SRC[2:0]	111	R/W	对相同存储体之间的下述存取时间，指定规定的周期数 (Trc)。 (1) 从发行 ACT 命令到进行自动刷新 (2) 从发行 ACT 命令到发行 ACT 命令 000: 8 个周期 001: 9 个周期 010: 10 个周期 011: 11 个周期 100: 12 个周期 101: 13 个周期 110: 14 个周期 111: 15 个周期
4 ~ 2	SCL[2:0]	001	R/W	指定读数据时的 CAS 等待时间 (CL)。 000: 2 个周期 001: 3 个周期 上述以外: 保留
1	SRCD	1	R/W	指定发行 RAS (ACT 命令) 和发行 CAS (READ/READA 和 WRITE/WRITEA 命令) 之间的周期数 (Trcd)。 0: 2 个周期 1: 3 个周期
0	—	1	R	保留位 读取值为 1，只能写 1。

11.4.5 SDRAM 行属性寄存器 (SDRA)

位:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	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
	—	—	—	—	SPLIT[3:0]				—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
63 ~ 12	—	全 0	R	保留位 读取值为 0, 只能写 0。
11 ~ 8	SPLIT[3:0]	0010	R/W	通过此位指定连接的 SDRAM 的 row/column 结构。 0010: 12×9 (= 8M×16bit 产品 or 8M×32bit 产品) 0100: 13×9 (= 16M×16bit 产品) 禁止上述以外的设定。 SPLIT 位和 row/column 的关系如表 11.6 所示。
7 ~ 0	—	全 0	R	保留位 读取值为 0, 只能写 0。

表 11.6 地址多路复用

- 外部总线宽度为 32 位的情况

外部总线 32bit	SDRAM 地址	BA1	BA0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
	SH7764 地址	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
128Mbit (16MB) 8M×16	Bank (2)	12	13													
	Row (12)				11	24	23	22	21	20	19	18	17	16	15	14
	Col (9)								10	9	8	7	6	5	4	3
256Mbit (32MB) 8M×32	Bank (2)	12	13													
	Row (12)				11	24	23	22	21	20	19	18	17	16	15	14
	Col (9)								10	9	8	7	6	5	4	3
256Mbit (32MB) 16M×16	Bank (2)	12	13													
	Row (13)			11	25	24	23	22	21	20	19	18	17	16	15	14
	Col (9)								10	9	8	7	6	5	4	3

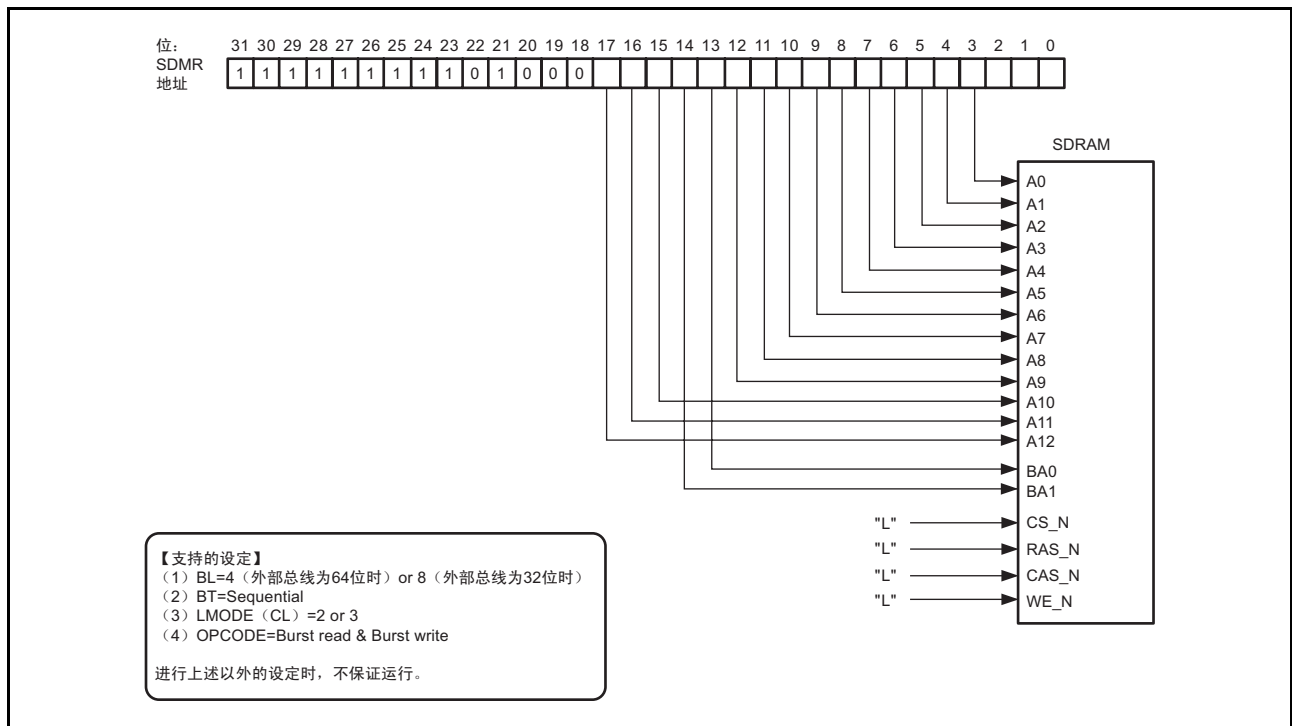
- 外部总线宽度为 64 位的情况

外部总线 64bit	SDRAM 地址	BA1	BA0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
	SH7764 地址	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
128Mbit (16MB) 8M×16	Bank (2)	14	13													
	Row (12)				12	25	24	23	22	21	20	19	18	17	16	15
	Col (9)							11	10	9	8	7	6	5	4	3
256Mbit (32MB) 8M×32	Bank (2)	14	13													
	Row (12)				12	25	24	23	22	21	20	19	18	17	16	15
	Col (9)							11	10	9	8	7	6	5	4	3

11.4.6 SDRAM 模式寄存器 (SDMR)

此寄存器用于设定 SDRAM 的模式寄存器。

因为本模块无实体，所以读无效。对于 SDRAM，只有写地址有意义，忽略写数据。



11.4.7 仲裁模式寄存器 (AMR)

位:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48		
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—		
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:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32		
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—		
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:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
	—	—	—	—	—	LAM[2:0]			—	—	—	—	—	—	—	SWAM		
初始值:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0		
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W		
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	—	—	—	—	—	—	—	—	PAM[7:0]								—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R	R		

位	位名	初始值	R/W	说明
63 ~ 27	—	全 0	R	保留位 读取值为 0，只能写 0。
26 ~ 24	LAM[2:0]	100	R/W	LCDC 仲裁选择位 设定 LCDC 的仲裁优先顺序。 100: LCDC 的优先顺序为 Level1 (默认值) 010: LCDC 的优先顺序为 Level2 001: LCDC 的优先顺序为 Level3
23 ~ 17	—	全 0	R	保留位 读取值为 0，只能写 0。
16	SWAM	0	R/W	Super Hyway 模块 Level2 仲裁允许位 将 Super Hyway 模块的优先顺序设定为 Level2。
15 ~ 8	—	全 0	R	保留位 读取值为 0，只能写 0。
7 ~ 0	PAM[7:0]	H'00	R/W	Pixel 总线模块 Level2 仲裁允许位 将 Pixel 总线模块的优先顺序设定为 Level2。各位和模块的对应如下： PAM[7]: G2D (命令) PAM[6]: G2D (数据) PAM[5]: (保留) PAM[4]: (保留) PAM[3]: (保留) PAM[2]: ATAPI PAM[1]: (保留) PAM[0]: (保留)

11.4.8 线性平铺转换控制寄存器 (LTCn)

位:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LTE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	LTMWX[3:0]			—	—	—	—	—	—	—	—	—	LTGBM
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R/W

位	位名	初始值	R/W	说明
63 ~ 32	—	全 0	R	保留位 读取值为 0，只能写 0。
31	LTE	0	R/W	线性平铺转换允许位 通过 LTAD 和 LTAM，在指定的空间中进行线性平铺转换。
30 ~ 13	—	全 0	R	保留位 读取值为 0，只能写 0。
12 ~ 9	LTMWX [3:0]	H'0	R/W	存储器宽度设定位 设定图像区的宽度。 0001: 512 0010: 1024 0100: 2048 1000: 4096 禁止上述以外的设定。
8 ~ 1	—	全 0	R	保留位 读取值为 0，只能写 0。
0	LTGBM	0	R/W	16bpp 图形模式允许位 指定图形模式。 0: 8bpp 1: 16bpp

【注】 在上电后的初始设定时，必须在任何模块都不存取 SDRAM 的状态下进行设定（自动刷新除外）。

11.4.9 线性平铺转换区起始地址寄存器 (LTADn)

位:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	LTAD[8:0]								—	—	—	—	
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
63 ~ 29	—	全 0	R	保留位 读取值为 0, 只能写 0。
28 ~ 20	LTAD[8:0]	H'000	R/W	线性平铺转换区起始地址位 指定线性平铺转换区的起始地址。
19 ~ 0	—	全 0	R	保留位 读取值为 0, 只能写 0。

【注】 在上电后的初始设定时, 必须在任何模块都不存取 SDRAM 的状态下进行设定 (自动刷新除外)。

11.4.10 线性平铺转换区起始地址屏蔽寄存器 (LTAMn)

位:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	LTAM[8:0]								—	—	—	—	
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
63 ~ 29	—	全 0	R	保留位 读取值为 0, 只能写 0。
28 ~ 20	LTAM[8:0]	H'000	R/W	线性平铺转换区起始地址屏蔽位 指定寄存器 LTAD 和实际地址的比较范围。
19 ~ 0	—	全 0	R	保留位 读取值为 0, 只能写 0。

必须给 LTAM[8:0] 指定统一存储器的物理地址。

此数据格式只为将 1 向左排列的格式。即指定 "(H'000,) (H'100,) H'180, H'1C0, H'1E0, H'1F0, H'1F8, H'1FC, H'1FE, H'1FF" 中的一个。

【注】 因为本 LSI 的统一存储空间为区域 1 和区域 2 的 64MB 空间, 所以不指定 (H'000,) (H'100,)。

例:

- LTAD[8:0] == B'01000000
LTAM[8:0] == B'11111000
此时, 进行平铺转换的区域为地址的 bit28 ~ 23 的 B'010000 区域 (8MB)。
- LTAD[8:0] == B'00101010
LTAM[8:0] == B'11111100
此时, 进行平铺转换的区域为地址的 bit28 ~ 22 的 B'0010101 区域 (4MB)。
- LTAM[8:0] == B'11111111
此时, 1MB 的空间为进行线性平铺转换的空间。
- LTAM[8:0] == B'11000000
此时, 128MB 的空间为进行线性平铺转换的空间。

在上电后的初始设定时, 必须在任何模块都不存取 SDRAM 的状态下进行设定 (自动刷新除外)。

11.4.11 请求屏蔽设定寄存器 (RQM)

位:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	NMIME	—	—	—	—	—	—	—	LCDM
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	VDCM	—	—	—	2DDM	2DCM	—	—	ATAM	—	—
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	R	R	R/W	R/W	R	R	R/W	R	R

位	位名	初始值	R/W	说明
63 ~ 25	—	全 0	R	保留位 读取值为 0, 只能写 0。
24	NMIME	0	R/W	NMI 时请求屏蔽允许位 0: 在 NMI 时不屏蔽主存储器的存取请求。 1: 在 NMI 时屏蔽主存储器的存取请求。
23 ~ 17	—	全 0	R	保留位 读取值为 0, 只能写 0。
16	LCDM	0	R/W	LCDC 请求屏蔽允许位 0: 不屏蔽 LCDC 请求。 1: 屏蔽 LCDC 请求。
15 ~ 11	—	全 0	R	保留位 读取值为 0, 只能写 0。
10	VDCM	0	R/W	VDC2 请求屏蔽允许位 0: 不屏蔽 VDC2 请求。 1: 屏蔽 VDC2 请求。
9 ~ 7	—	全 0	R	保留位 读取值为 0, 只能写 0。
6	2DDM	0	R/W	2DD (G2D 的数据) 请求屏蔽允许位 0: 不屏蔽 2DDM 请求。 1: 屏蔽 2DDM 请求。
5	2DCM	0	R/W	2DC (G2D 的命令) 请求屏蔽允许位 0: 不屏蔽 2DCM 请求。 1: 屏蔽 2DCM 请求。
4、3	—	全 0	R	保留位 读取值为 0, 只能写 0。
2	ATAM	0	R/W	ATAPI 请求屏蔽允许位 0: 不屏蔽 ATAPI 请求。 1: 屏蔽 ATAPI 请求。
1、0	—	全 0	R	保留位 读取值为 0, 只能写 0。

在 NMI 时，能按模块屏蔽存储器存取的仲裁。因为本设定反映于仲裁运行，所以不适用于存取中的存储器。

11.4.12 总线控制寄存器 (BCR)

位:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	IRSD[2:0]			DPUP	—	OPUP	—	—	—	—	BREQEN	—	—	—
初始值:	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R	R/W	R	R	R	R	R/W	R	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	IPUP	—	—	—	—	—	—	ASYNC1	ASYNC0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W	R/W

位	位名	初始值	R/W	说明
63 ~ 30	—	全 0	R	保留位 读取值为 0，只能写 0。
29 ~ 27	IRSD[2:0]	111	R/W	RAM-SDRAM 周期之间的空闲周期位 指定存取 SRAM 区（区域 0、3）和存取 SDRAM 区（区域 1、2）之间的要插入的空闲周期数。 000：插入 4 个空闲周期 001：插入 5 个空闲周期 010：插入 6 个空闲周期 011：插入 7 个空闲周期 100：插入 8 个空闲周期 101：插入 9 个空闲周期 110：插入 10 个空闲周期 111：插入 11 个空闲周期
26	DPUP	0	R/W	数据引脚的上拉电阻控制位 指定数据引脚（D63 ~ D0）的上拉电阻的状态。此位在上电复位时被初始化。 0：数据引脚（D63 ~ D0）的上拉电阻在存储器存取前后为 ON 状态。 1：数据引脚（D63 ~ D0）的上拉电阻为 OFF 状态。 【注】 需要数据引脚上拉时，建议外接上拉电阻。
25	—	0	R	保留位 读取值为 0，只能写 0。

位	位名	初始值	R/W	说明
24	OPUP	0	R/W	控制输出引脚的上拉电阻控制位 指定在 A25 ~ A0、 \overline{BS} 、 \overline{CSn} 、 \overline{RD} 、 $\overline{WEn/DQMn}$ 、 $\overline{RD/WR}$ 、 \overline{RAS} 、 \overline{CAS} 为 Hi-Z 时的上拉电阻的状态。此位在上电复位时被初始化。 0: 控制输出的上拉电阻为 ON 状态。 1: 控制输出的上拉电阻为 OFF 状态。
23 ~ 20	—	全 0	R	保留位 读取值为 0, 只能写 0。
19	BREQEN	0	R/W	\overline{BREQ} 允许位 设定是否能接受外部请求。此位在上电复位时被初始化。 0: 不接受外部请求 1: 接受外部请求
18 ~ 9	—	全 0	R	保留位 读取值为 0, 只能写 0。
8	IPUP	0	R/W	输入引脚的上拉电阻控制位 指定输入引脚 (\overline{RDY} 和 \overline{BREQ}) 的上拉电阻的状态。 此位在上电复位时被初始化。 0: 输入引脚 (\overline{RDY} 和 \overline{BREQ}) 的上拉电阻为 ON 状态。 1: 输入引脚 (\overline{RDY} 和 \overline{BREQ}) 的上拉电阻为 OFF 状态。
7 ~ 2	—	全 0	R	保留位 读取值为 0, 只能写 0。
1	ASYNC1	0	R/W	异步输入 1 允许 \overline{BREQ} 引脚的异步输入。 0: \overline{BREQ} 引脚为对 CLKOUT 同步输入。 1: \overline{BREQ} 引脚为对 CLKOUT 异步输入。
0	ASYNC0	0	R/W	异步输入 0 允许 \overline{RDY} 引脚的异步输入。 0: \overline{RDY} 引脚为对 CLKOUT 同步输入。 1: \overline{RDY} 引脚为对 CLKOUT 异步输入。

11.4.13 CS0 总线控制寄存器 (CS0BCR)

位:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	IWW[2:0]			—	IWRWD[2:0]			—	IWRWS[2:0]			—	IWRRD[2:0]		
初始值:	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	IWRRS[2:0]			—	—	SZ[1:0]		RDSPL	—	—	—	—	—	—	—
初始值:	0	1	1	1	0	0	—	—	1	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R	R	R/W	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
63 ~ 31	—	全 0	R	保留位 读取值为 0，只能写 0。
30 ~ 28	IWW[2:0]	111	R/W	写 - 读 / 写 - 写周期之间的空闲周期位 指定在对连接 SRAM 区 (区域 0、3) 的存储器进行写存取后要插入的空闲周期数。 对象周期是写 - 读周期之间和写 - 写周期之间，对象区域是存取区域 0- 区域 0 之间和存取区域 0- 区域 3 之间。 000: 不插入空闲周期 001: 插入 1 个空闲周期 010: 插入 2 个空闲周期 011: 插入 3 个空闲周期 100: 插入 4 个空闲周期 101: 插入 5 个空闲周期 110: 插入 6 个空闲周期 111: 插入 7 个空闲周期
27	—	0	R	保留位 读取值为 0，只能写 0。
26 ~ 24	IWRWD [2:0]	111	R/W	不同空间读 - 写周期之间的空闲周期位 指定在对连接区域 0 空间的存储器进行读存取后要插入的空闲周期数。 对象周期是读 - 写周期之间，对象区域为存取区域 0- 区域 3 之间。 000: 不插入空闲周期 001: 插入 1 个空闲周期 010: 插入 2 个空闲周期 011: 插入 3 个空闲周期 100: 插入 4 个空闲周期 101: 插入 5 个空闲周期 110: 插入 6 个空闲周期 111: 插入 7 个空闲周期

位	位名	初始值	R/W	说明
23	—	0	R	保留位 读取值为 0，只能写 0。
22 ~ 20	IWRWS [2:0]	111	R/W	区域 0 相同空间读 - 写周期之间的空闲周期位 指定在对连接区域 0 空间的存储器进行读存取后要插入的空闲周期数。 对象周期是连续存取的不同空间并且是读 - 写周期的情况。 000: 不插入空闲周期 001: 插入 1 个空闲周期 010: 插入 2 个空闲周期 011: 插入 3 个空闲周期 100: 插入 4 个空闲周期 101: 插入 5 个空闲周期 110: 插入 6 个空闲周期 111: 插入 7 个空闲周期
19	—	0	R	保留位 读取值为 0，只能写 0。
18 ~ 16	IWRRD [2:0]	111	R/W	不同空间读 - 读周期之间的空闲周期位 指定在对连接区域 0 空间的存储器进行读存取后要插入的空闲周期数。 对象周期是读 - 读周期之间，对象区域是存取区域 0- 区域 3 之间。 000: 不插入空闲周期 001: 插入 1 个空闲周期 010: 插入 2 个空闲周期 011: 插入 3 个空闲周期 100: 插入 4 个空闲周期 101: 插入 5 个空闲周期 110: 插入 6 个空闲周期 111: 插入 7 个空闲周期
15	—	0	R	保留位 读取值为 0，只能写 0。
14 ~ 12	IWRRS [2:0]	111	R/W	区域 0 相同空间读 - 读周期之间的空闲周期位 指定在对连接区域 0 空间的存储器进行读存取后要插入的空闲周期数。 对象周期是连续存取的不同空间并且是读 - 写周期的情况。 000: 不插入空闲周期 001: 插入 1 个空闲周期 010: 插入 2 个空闲周期 011: 插入 3 个空闲周期 100: 插入 4 个空闲周期 101: 插入 5 个空闲周期 110: 插入 6 个空闲周期 111: 插入 7 个空闲周期
11、10	—	全 0	R	保留位 读取值为 0，只能写 0。

位	位名	初始值	R/W	说明
9、8	SZ[1:0]	不定值	R	总线宽度位 在上电复位时，对指定总线宽度的外部引脚 (MODE4 和 MODE3) 进行采样。 00: 保留 01: 8 位 10: 16 位 11: 32 位
7	RDSPL	1	R/W	\overline{RD} 保持周期位 对 \overline{RD} 读数据采样时序的保持时间，指定要插入的周期数。在将此位置 1 时，必须通过 CS0WCR 的 RDH 位，至少将 \overline{RD} 无效 $\overline{CS0}$ 无效的延迟周期设定为 1。 通过将此位置 1， \overline{RD} 无效 $\overline{CS0}$ 无效的延迟周期减少 1 个周期。 0: 无 1: 插入 1 个周期
6 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

11.4.14 \overline{CSn} 等待控制寄存器 (CSnWCR)

位:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	ADS[2:0]			—	ADH[2:0]			—	RDS[2:0]			—	RDH[2:0]		
初始值:	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	WTS[2:0]			—	WTH[2:0]			—	BSH[2:0]			IW[3:0]			
初始值:	0	1	1	1	0	1	1	1	0	0	0	0	1	1	1	1
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
63 ~ 31	—	全 0	R	保留位 读取值为 0，只能写 0。

位	位名	初始值	R/W	说明
30 ~ 28	ADS[2:0]	111	R/W	地址准备周期位 对地址的 \overline{CSn} 有效的准备时间, 指定要插入的周期数。 000: 不插入周期 001: 插入 1 个周期 010: 插入 2 个周期 011: 插入 3 个周期 100: 插入 4 个周期 101: 插入 5 个周期 110: 插入 6 个周期 111: 插入 7 个周期
27	—	0	R	保留位 读取值为 0, 只能写 0。
26 ~ 24	ADH[2:0]	111	R/W	地址保持周期位 对地址的 \overline{CSn} 无效的保持时间, 指定要插入的周期数。 000: 不插入周期 001: 插入 1 个周期 010: 插入 2 个周期 011: 插入 3 个周期 100: 插入 4 个周期 101: 插入 5 个周期 110: 插入 6 个周期 111: 插入 7 个周期
23	—	0	R	保留位 读取值为 0, 只能写 0。
22 ~ 20	RDS[2:0]	111	R/W	\overline{CSn} 有效 \overline{RD} 有效的延迟周期位 在从 \overline{CSn} 有效到 \overline{RD} 有效之间, 指定要插入的周期数。 000: 不插入周期 (1 个周期延迟) 001: 插入 1 个周期 (2 个周期延迟) 010: 插入 2 个周期 (3 个周期延迟) 011: 插入 3 个周期 (4 个周期延迟) 100: 插入 4 个周期 (5 个周期延迟) 101: 插入 5 个周期 (6 个周期延迟) 110: 插入 6 个周期 (7 个周期延迟) 111: 插入 7 个周期 (8 个周期延迟)
19	—	0	R	保留位 读取值为 0, 只能写 0。
18 ~ 16	RDH[2:0]	111	R/W	\overline{RD} 无效 \overline{CSn} 无效的延迟周期位 在从 \overline{RD} 无效到 \overline{CSn} 无效之间, 指定要插入的周期数。 000: 不插入周期 (0 个周期延迟) 001: 插入 1 个周期 (1 个周期延迟) 010: 插入 2 个周期 (2 个周期延迟) 011: 插入 3 个周期 (3 个周期延迟) 100: 插入 4 个周期 (4 个周期延迟) 101: 插入 5 个周期 (5 个周期延迟) 110: 插入 6 个周期 (6 个周期延迟) 111: 插入 7 个周期 (7 个周期延迟)

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读取值为 0，只能写 0。
14 ~ 12	WTS[2:0]	111	R/W	$\overline{\text{CSn}}$ 有效 $\overline{\text{WEn}}$ 有效的延迟周期位 在从 $\overline{\text{CSn}}$ 有效到 $\overline{\text{WEn}}$ 有效之间，指定要插入的周期数。 000: 不插入周期 (0.5 个周期延迟) 001: 插入 1 个周期 (1.5 个周期延迟) 010: 插入 2 个周期 (2.5 个周期延迟) 011: 插入 3 个周期 (3.5 个周期延迟) 100: 插入 4 个周期 (4.5 个周期延迟) 101: 插入 5 个周期 (5.5 个周期延迟) 110: 插入 6 个周期 (6.5 个周期延迟) 111: 插入 7 个周期 (7.5 个周期延迟)
11	—	0	R	保留位 读取值为 0，只能写 0。
10 ~ 8	WTH[2:0]	111	R/W	$\overline{\text{WEn}}$ 无效 $\overline{\text{CSn}}$ 无效的延迟周期位 在从 $\overline{\text{WEn}}$ 无效到 $\overline{\text{CSn}}$ 无效，指定要插入的周期数。 000: 不插入周期 (0.5 个周期延迟) 001: 插入 1 个周期 (1.5 个周期延迟) 010: 插入 2 个周期 (2.5 个周期延迟) 011: 插入 3 个周期 (3.5 个周期延迟) 100: 插入 4 个周期 (4.5 个周期延迟) 101: 插入 5 个周期 (5.5 个周期延迟) 110: 插入 6 个周期 (6.5 个周期延迟) 111: 插入 7 个周期 (7.5 个周期延迟)
7	—	0	R	保留位 读取值为 0，只能写 0。
6 ~ 4	BSH[2:0]	000	R/W	为了延长 $\overline{\text{CSn}}$ 空间存取时的 $\overline{\text{BS}}$ 的有效时间，指定要插入的周期数。 只有在读时将 $\overline{\text{CSnWCR}}$ 的 RDS 设定为 000 以外的值、或者在写时将 $\overline{\text{CSnWCR}}$ 的 WTS 设定为 000 以外的值的情况下，延长才有效。另外，存取的总周期数不会因此位的设定而变。 000: 给 $\overline{\text{BS}}$ 有效插入 1 个周期 001: 给 $\overline{\text{BS}}$ 有效插入 2 个周期 010: 禁止设定 011: 禁止设定 100: 禁止设定 101: 禁止设定 110: 禁止设定 111: 禁止设定

位	位名	初始值	R/W	说明
3 ~ 0	IW[3:0]	1111	R/W	<p>在 \overline{CSn} 空间存取时, 指定要插入的等待周期数。</p> <p>在设定为不插入周期时, 不能通过 \overline{RDY} 引脚插入外部等待。</p> <p>0000: 不插入周期 0001: 插入 1 个周期 0010: 插入 2 个周期 0011: 插入 3 个周期 0100: 插入 4 个周期 0101: 插入 5 个周期 0110: 插入 6 个周期 0111: 插入 7 个周期 1000: 插入 8 个周期 1001: 插入 9 个周期 1010: 插入 11 个周期 1011: 插入 13 个周期 1100: 插入 15 个周期 1101: 插入 17 个周期 1110: 插入 21 个周期 1111: 插入 25 个周期</p>

11.4.15 $\overline{CS3}$ 总线控制寄存器 (CS3BCR)

位:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	IWW[2:0]			—	IWRWD[2:0]			—	IWRWS[2:0]			—	IWRWD[2:0]		
初始值:	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	IWRRS[2:0]			—	—	SZ[1:0]	RDSPL	—	—	—	—	—	—	—	—
初始值:	0	1	1	1	0	0	1	1	1	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
63 ~ 31	—	全 0	R	保留位 读取值为 0, 只能写 0。

位	位名	初始值	R/W	说明
30 ~ 28	IWW[2:0]	111	R/W	<p>写 - 读 / 写 - 写周期之间的空闲周期位</p> <p>指定在对连接 SRAM 区 (区域 0、3) 的存储器进行写存取后要插入的空闲周期数。对象周期是写 - 读周期之间和写 - 写周期之间, 对象区域是存取区域 3- 区域 0 之间和存取区域 3- 区域 3 之间。</p> <p>000: 不插入空闲周期 001: 插入 1 个空闲周期 010: 插入 2 个空闲周期 011: 插入 3 个空闲周期 100: 插入 4 个空闲周期 101: 插入 5 个空闲周期 110: 插入 6 个空闲周期 111: 插入 7 个空闲周期</p>
27	—	0	R	<p>保留位</p> <p>读取值为 0, 只能写 0。</p>
26 ~ 24	IWRWD [2:0]	111	R/W	<p>不同空间读 - 写周期之间的空闲周期位</p> <p>指定在对连接区域 3 空间的存储器进行读存取后要插入的空闲周期数。对象周期是读 - 写周期之间, 对象区域是存取区域 3- 区域 0 之间。</p> <p>000: 不插入空闲周期 001: 插入 1 个空闲周期 010: 插入 2 个空闲周期 011: 插入 3 个空闲周期 100: 插入 4 个空闲周期 101: 插入 5 个空闲周期 110: 插入 6 个空闲周期 111: 插入 7 个空闲周期</p>
23	—	0	R	<p>保留位</p> <p>读取值为 0, 只能写 0。</p>
22 ~ 20	IWRWS [2:0]	111	R/W	<p>区域 3 相同空间读 - 写周期之间的空闲周期位</p> <p>指定在对连接区域 3 空间的存储器进行读存取后要插入的空闲周期数。对象周期是连续存取的不同空间并且是读 - 写周期的情况。</p> <p>000: 不插入空闲周期 001: 插入 1 个空闲周期 010: 插入 2 个空闲周期 011: 插入 3 个空闲周期 100: 插入 4 个空闲周期 101: 插入 5 个空闲周期 110: 插入 6 个空闲周期 111: 插入 7 个空闲周期</p>
19	—	0	R	<p>保留位</p> <p>读取值为 0, 只能写 0。</p>

位	位名	初始值	R/W	说明
18 ~ 16	IWRRD [2:0]	111	R/W	不同空间读 - 读周期之间的空闲周期位 指定在对连接区域 3 空间的存储器进行读存取后要插入的空闲周期数。 对象周期是读 - 读周期之间, 对象区域是存取区域 3- 区域 0 之间。 000: 不插入空闲周期 001: 插入 1 个空闲周期 010: 插入 2 个空闲周期 011: 插入 3 个空闲周期 100: 插入 4 个空闲周期 101: 插入 5 个空闲周期 110: 插入 6 个空闲周期 111: 插入 7 个空闲周期
15	—	0	R	保留位 读取值为 0, 只能写 0。
14 ~ 12	IWRRS [2:0]	111	R/W	区域 3 相同空间读 - 读周期之间的空闲周期位 指定在对连接区域 3 空间的存储器进行读存取后要插入的空闲周期数。 对象周期是连续存取的相同空间并且是读 - 写周期的情况。 000: 不插入空闲周期 001: 插入 1 个空闲周期 010: 插入 2 个空闲周期 011: 插入 3 个空闲周期 100: 插入 4 个空闲周期 101: 插入 5 个空闲周期 110: 插入 6 个空闲周期 111: 插入 7 个空闲周期
11、10	—	全 0	R	保留位 读取值为 0, 只能写 0。
9、8	SZ	11	R/W	总线宽度位 指定区域 3 的总线宽度。 00: 保留 01: 8 位 10: 16 位 11: 32 位
7	RDSPL	1	R/W	\overline{RD} 保持周期位 对 \overline{RD} 读数据采样时序的保持时间, 指定要插入的周期数。在将此位置 1 时, 必须通过 CS3WCR 的 RDH 位, 至少将 \overline{RD} 无效 - $\overline{CS3}$ 无效的延迟周期设定为 1。 通过将此位置 1, \overline{RD} 无效 - $\overline{CS3}$ 无效的延迟周期减少 1 个周期。 0: 无 1: 插入 1 个周期
6 ~ 0	—	全 0	R	保留位 读取值为 0, 只能写 0。

11.5 运行说明

11.5.1 字节排序 / 存取长度和数据调整

本 LSI 支持高位字节 (MSByte) 为地址 0 的大端法和低位字节 (LSByte) 为地址 0 的小端法的字节数据的排列方法。在通过 PRESET 引脚进行上电复位时, 由外部引脚 (MODE5 引脚) 设定字节排序, 当 MODE5 引脚为低电平时为大端法, 为高电平时为小端法。

对通常的存储器 (区域 0 和区域 3) 能选择 8 位、16 位或者 32 位的数据总线; 对 SDRAM (区域 1 和区域 2) 能选择 32 位或者 64 位的数据总线。根据各设备的数据总线宽度和字节排序进行数据调整。因此, 当数据总线宽度小于存取长度时, 在到达存取长度前自动产生多个总线周期。此时, 将对应总线宽度的地址自动加 1 进行存取。例如, 通过 SRAM 接口给 8 位总线的区域进行长字存取时, 将地址逐个自动加 1 进行 4 次存取。另外, 在进行 32 字节传送时, 根据设定的总线宽度, 连续传送 32 字节的数据。最初的存取是对发生存取请求的数据进行, 剩余的存取是对 32 字节边界的数据以环绕方式进行。本 LSI 在各种接口之间自动进行数据的调整和数据长度的转换 (对于 Pixel 总线和 LCDC 的存取不进行环绕方式的存取。)

字节排序、设备的数据宽度和存取单位之间的关系如表 11.7 ~ 表 11.16 所示。

表 11.7 32 位外部设备 / 大端法的存取和数据调整 (区域 0、3)

运行		NO.	数据总线				选通信号			
存取长度	地址		D31 ~ D24	D23 ~ D16	D15 ~ D8	D7 ~ D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
字节	4n	1	数据 7 ~ 0	—	—	—	有效			
	4n+1	1	—	数据 7 ~ 0	—	—		有效		
	4n+2	1	—	—	数据 7 ~ 0	—			有效	
	4n+3	1	—	—	—	数据 7 ~ 0				有效
字	4n	1	数据 15 ~ 8	数据 7 ~ 0	—	—	有效	有效		
	4n+2	1	—	—	数据 15 ~ 8	数据 7 ~ 0			有效	有效
长字	4n	1	数据 31 ~ 24	数据 23 ~ 16	数据 15 ~ 8	数据 7 ~ 0	有效	有效	有效	有效

表 11.8 16 位外部设备 / 大端法的存取和数据调整 (区域 0、3)

运行		NO.	数据总线				选通信号			
存取长度	地址		D31 ~ D24	D23 ~ D16	D15 ~ D8	D7 ~ D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
字节	2n	1	—	—	数据 7 ~ 0	—			有效	
	2n+1	1	—	—	—	数据 7 ~ 0				有效
字	2n	1	—	—	数据 15 ~ 8	数据 7 ~ 0			有效	有效
长字	4n	1	—	—	数据 31 ~ 24	数据 23 ~ 16			有效	有效
	4n+2	2	—	—	数据 15 ~ 8	数据 7 ~ 0			有效	有效

表 11.9 8 位外部设备 / 大端法的存取和数据调整 (区域 0、3)

运行		NO.	数据总线				选通信号			
存取长度	地址		D31 ~ D24	D23 ~ D16	D15 ~ D8	D7 ~ D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
字节	n	1	—	—	—	数据 7 ~ 0				有效
字	2n	1	—	—	—	数据 15 ~ 8				有效
	2n+1	2	—	—	—	数据 7 ~ 0				有效
长字	4n	1	—	—	—	数据 31 ~ 24				有效
	4n+1	2	—	—	—	数据 23 ~ 16				有效
	4n+2	3	—	—	—	数据 15 ~ 8				有效
	4n+3	4	—	—	—	数据 7 ~ 0				有效

表 11.10 32 位外部设备 / 小端法的存取和数据调整 (区域 0、3)

运行		NO.	数据总线				选通信号			
存取长度	地址		D31 ~ D24	D23 ~ D16	D15 ~ D8	D7 ~ D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
字节	4n	1	—	—	—	数据 7 ~ 0				有效
	4n+1	1	—	—	数据 7 ~ 0	—			有效	
	4n+2	1	—	数据 7 ~ 0	—	—		有效		
	4n+3	1	数据 7 ~ 0	—	—	—	有效			
字	4n	1	—	—	数据 15 ~ 8	数据 7 ~ 0			有效	有效
	4n+2	1	数据 15 ~ 8	数据 7 ~ 0	—	—	有效	有效		
长字	4n	1	数据 31 ~ 24	数据 23 ~ 16	数据 15 ~ 8	数据 7 ~ 0	有效	有效	有效	有效

表 11.11 16 位外部设备 / 小端法的存取和数据调整 (区域 0、3)

运行		NO.	数据总线				选通信号			
存取长度	地址		D31 ~ D24	D23 ~ D16	D15 ~ D8	D7 ~ D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
字节	2n	1	—	—	—	数据 7 ~ 0				有效
	2n+1	1	—	—	数据 7 ~ 0	—			有效	
字	2n	1	—	—	数据 15 ~ 8	数据 7 ~ 0			有效	有效
长字	4n	1	—	—	数据 15 ~ 8	数据 7 ~ 0			有效	有效
	4n+2	2	—	—	数据 31 ~ 24	数据 23 ~ 16			有效	有效

表 11.12 8 位外部设备 / 小端法的存取和数据调整 (区域 0、3)

运行		NO.	数据总线				选通信号			
存取长度	地址		D31 ~ D24	D23 ~ D16	D15 ~ D8	D7 ~ D0	$\overline{WE3}$	$\overline{WE2}$	$\overline{WE1}$	$\overline{WE0}$
字节	n	1	—	—	—	数据 7 ~ 0				有效
字	2n	1	—	—	—	数据 7 ~ 0				有效
	2n+1	2	—	—	—	数据 15 ~ 8				有效
长字	4n	1	—	—	—	数据 7 ~ 0				有效
	4n+1	2	—	—	—	数据 15 ~ 8				有效
	4n+2	3	—	—	—	数据 23 ~ 16				有效
	4n+3	4	—	—	—	数据 31 ~ 24				有效

表 11.13 32 位外部总线 / 大端法的存取和数据调整 (区域 1、2)

	D63 ~ 56	D55 ~ 48	D47 ~ 40	D39 ~ 32	D31 ~ 24	D23 ~ 16	D15 ~ 8	D7 ~ 0
地址 0 的字节存取					数据 7 ~ 0			
地址 1 的字节存取						数据 7 ~ 0		
地址 2 的字节存取							数据 7 ~ 0	
地址 3 的字节存取								数据 7 ~ 0
地址 4 的字节存取					数据 7 ~ 0			
地址 5 的字节存取						数据 7 ~ 0		
地址 6 的字节存取							数据 7 ~ 0	
地址 7 的字节存取								数据 7 ~ 0
地址 0 的字存取					数据 15 ~ 8	数据 7 ~ 0		
地址 2 的字存取							数据 15 ~ 8	数据 7 ~ 0
地址 4 的字存取					数据 15 ~ 8	数据 7 ~ 0		
地址 6 的字存取							数据 15 ~ 8	数据 7 ~ 0
地址 0 的长字存取					数据 31 ~ 24	数据 23 ~ 16	数据 15 ~ 8	数据 7 ~ 0
地址 4 的长字存取					数据 31 ~ 24	数据 23 ~ 16	数据 15 ~ 8	数据 7 ~ 0
地址 0 的 4 个字 存取 (第 1 次: 地址 0)					数据 63 ~ 56	数据 55 ~ 48	数据 47 ~ 40	数据 39 ~ 32
地址 0 的 4 个字 存取 (第 2 次: 地址 4)					数据 31 ~ 24	数据 23 ~ 16	数据 15 ~ 8	数据 7 ~ 0

表 11.14 32 位外部总线 / 小端法的存取和数据调整 (区域 1、2)

	D63 ~ 56	D55 ~ 48	D47 ~ 40	D39 ~ 32	D31 ~ 24	D23 ~ 16	D15 ~ 8	D7 ~ 0
地址 0 的字节存取								数据 7 ~ 0
地址 1 的字节存取							数据 7 ~ 0	
地址 2 的字节存取						数据 7 ~ 0		
地址 3 的字节存取					数据 7 ~ 0			
地址 4 的字节存取								数据 7 ~ 0
地址 5 的字节存取							数据 7 ~ 0	
地址 6 的字节存取						数据 7 ~ 0		
地址 7 的字节存取					数据 7 ~ 0			
地址 0 的字存取							数据 15 ~ 8	数据 7 ~ 0
地址 2 的字存取					数据 15 ~ 8	数据 7 ~ 0		
地址 4 的字存取							数据 15 ~ 8	数据 7 ~ 0
地址 6 的字存取					数据 15 ~ 8	数据 7 ~ 0		
地址 0 的长字存取					数据 31 ~ 24	数据 23 ~ 16	数据 15 ~ 8	数据 7 ~ 0
地址 4 的长字存取					数据 31 ~ 24	数据 23 ~ 16	数据 15 ~ 8	数据 7 ~ 0
地址 0 的 4 个字 存取 (第 1 次: 地址 0)					数据 31 ~ 24	数据 23 ~ 16	数据 15 ~ 8	数据 7 ~ 0
地址 0 的 4 个字 存取 (第 2 次: 地址 4)					数据 63 ~ 56	数据 55 ~ 48	数据 47 ~ 40	数据 39 ~ 32

表 11.15 64 位外部总线 / 大端法的存取和数据调整 (区域 1、2)

	D63 ~ 56	D55 ~ 48	D47 ~ 40	D39 ~ 32	D31 ~ 24	D23 ~ 16	D15 ~ 8	D7 ~ 0
地址 0 的字节存取	数据 7 ~ 0							
地址 1 的字节存取		数据 7 ~ 0						
地址 2 的字节存取			数据 7 ~ 0					
地址 3 的字节存取				数据 7 ~ 0				
地址 4 的字节存取					数据 7 ~ 0			
地址 5 的字节存取						数据 7 ~ 0		
地址 6 的字节存取							数据 7 ~ 0	
地址 7 的字节存取								数据 7 ~ 0
地址 0 的字存取	数据 15 ~ 8	数据 7 ~ 0						
地址 2 的字存取			数据 15 ~ 8	数据 7 ~ 0				
地址 4 的字存取					数据 15 ~ 8	数据 7 ~ 0		
地址 6 的字存取							数据 15 ~ 8	数据 7 ~ 0
地址 0 的长字存取	数据 31 ~ 24	数据 23 ~ 16	数据 15 ~ 8	数据 7 ~ 0				
地址 4 的长字存取					数据 31 ~ 24	数据 23 ~ 16	数据 15 ~ 8	数据 7 ~ 0
地址 0 的 4 个字	数据 63 ~ 56	数据 55 ~ 48	数据 47 ~ 40	数据 39 ~ 32	数据 31 ~ 24	数据 23 ~ 16	数据 15 ~ 8	数据 7 ~ 0

表 11.16 64 位外部总线 / 小端法的存取和数据调整 (区域 1、2)

	D63 ~ 56	D55 ~ 48	D47 ~ 40	D39 ~ 32	D31 ~ 24	D23 ~ 16	D15 ~ 8	D7 ~ 0
地址的字节存取								数据 7 ~ 0
地址的字节存取							数据 7 ~ 0	
地址的字节存取						数据 7 ~ 0		
地址的字节存取					数据 7 ~ 0			
地址的字节存取				数据 7 ~ 0				
地址的字节存取			数据 7 ~ 0					
地址 6 的字节存取		数据 7 ~ 0						
地址 7 的字节存取	数据 7 ~ 0							
地址 0 的字存取							数据 15 ~ 8	数据 7 ~ 0
地址 2 的字存取					数据 15 ~ 8	数据 7 ~ 0		
地址 4 的字存取			数据 15 ~ 8	数据 7 ~ 0				
地址 6 的字存取	数据 15 ~ 8	数据 7 ~ 0						
地址 0 的长字存取					数据 31 ~ 24	数据 23 ~ 16	数据 15 ~ 8	数据 7 ~ 0
地址 4 的长字存取	数据 31 ~ 24	数据 23 ~ 16	数据 15 ~ 8	数据 7 ~ 0				
地址 0 的 4 个字	数据 63 ~ 56	数据 55 ~ 48	数据 47 ~ 40	数据 39 ~ 32	数据 31 ~ 24	数据 23 ~ 16	数据 15 ~ 8	数据 7 ~ 0

11.5.2 各模块的数据调整

存储器控制器内部的字节排序和 CPU 的字节排序相同，大端法和小端法哪一个都能使用。连接 SuperHyway 总线的模块、连接 Pixel 总线的模块和连接 LCD 直接总线的 LCDC 分别在各模块内部进行总线宽度的数据调整。

11.6 SRAM 接口

11.6.1 基本时序

SRAM 接口主要针对 SRAM 的连接，输出选通信号。无等待的通常存取需要 2 个周期结束。为了表示总线周期的开始， \overline{BS} 有效信号为 1 个周期或者 2 个周期。 $\overline{CS0}$ 和 $\overline{CS3}$ 信号在 T1 的上升沿有效，在 T2 的时钟的下一个上升沿无效。因此，在以最小间隔存取时，不产生无效期间。

在读数据时，没有存取长度的指定。地址引脚 (A[25:0]) 输出正确的存取地址，但是，因为没有存取长度的指定，所以当使用 32 位设备时总是读取 32 位；当使用 16 位设备时总是读取 16 位。在写数据时，只有写字节的 \overline{WE} 信号有效。在进行 32 字节传送时，根据设定的总线宽度，连续传送 32 字节的数据。

最初的存取是对发生存取请求的数据进行，剩余的存取是对 32 字节边界的数据以环绕方式进行。

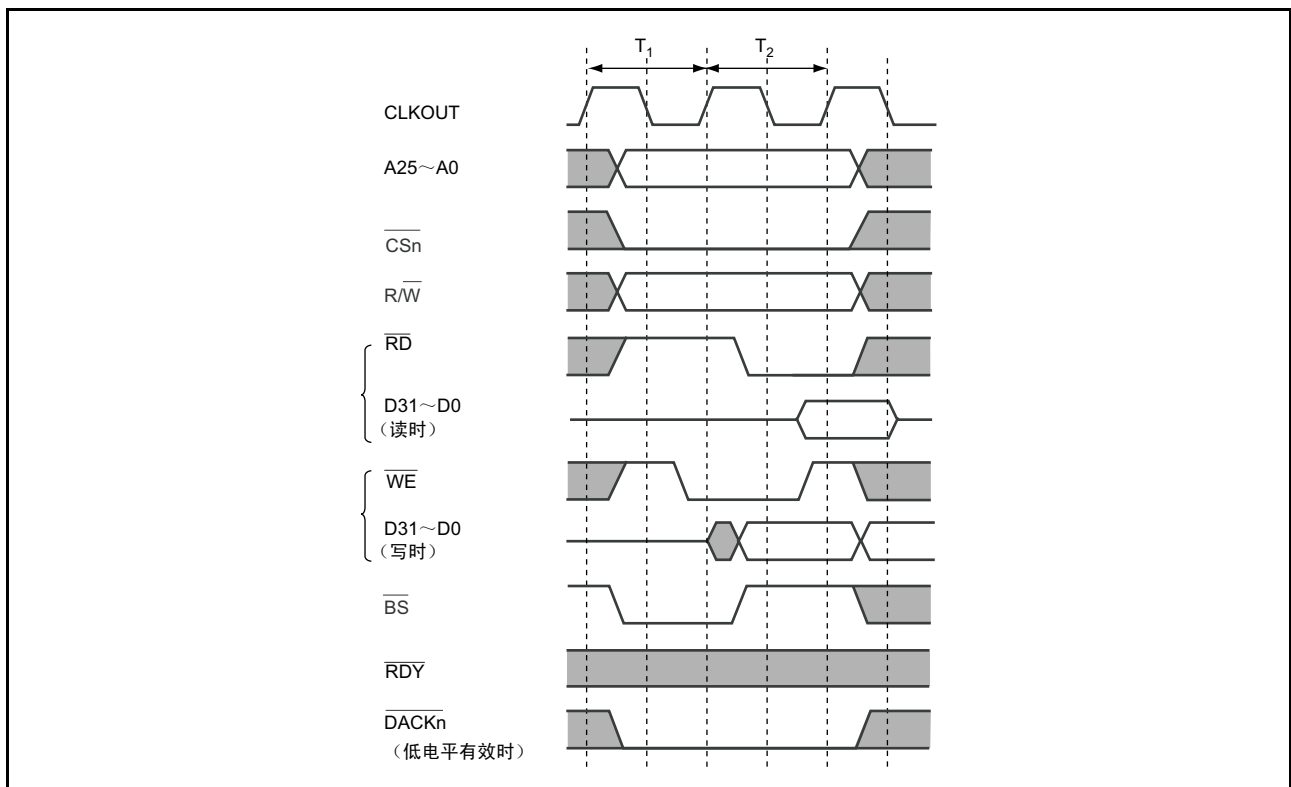


图 11.3 SRAM 接口的基本时序

与 32 位、16 位和 8 位数据宽度的 SRAM 连接的例子分别如图 11.4、图 11.5 和图 11.6 所示。

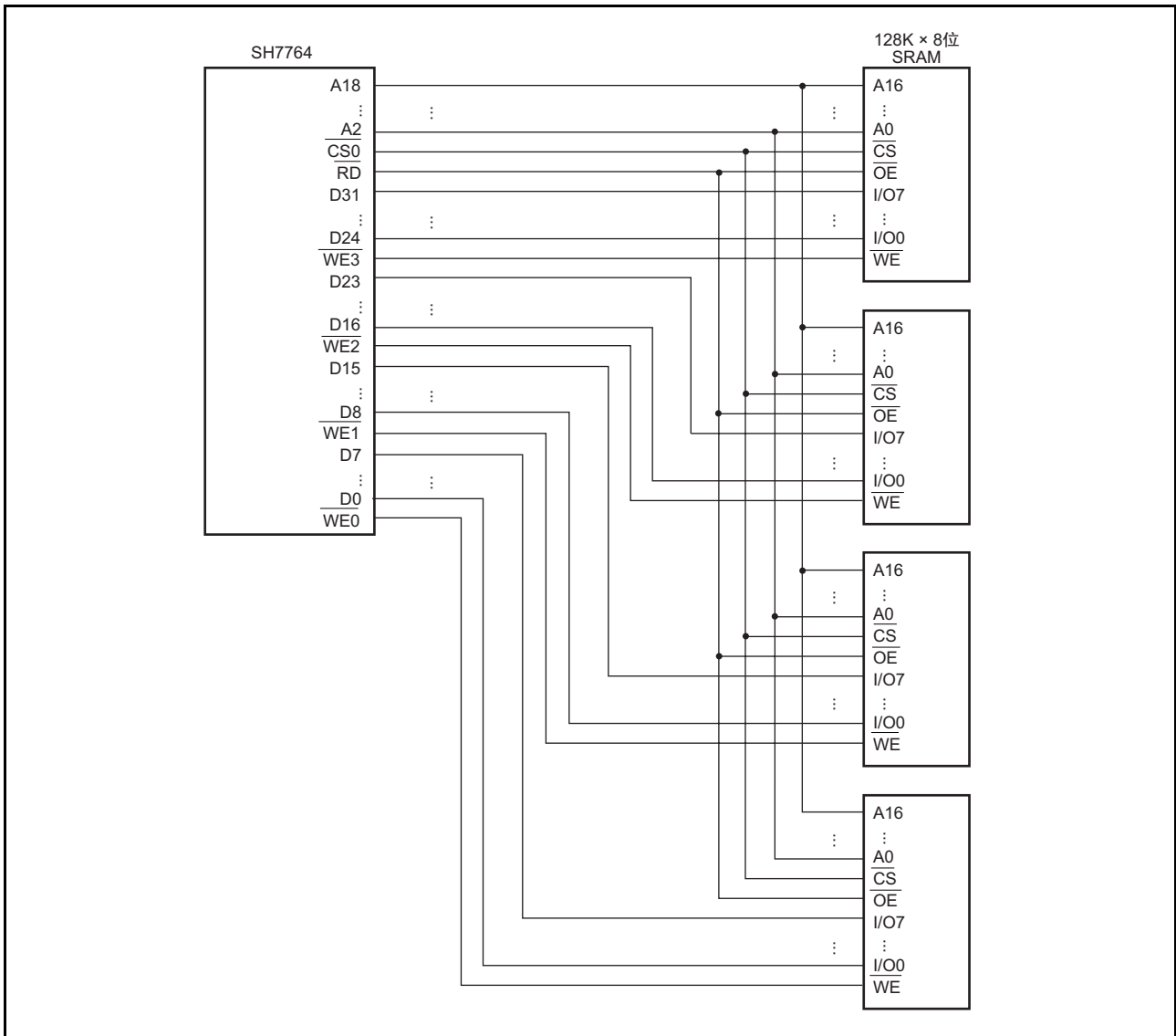


图 11.4 与 32 位数据宽度的 SRAM 连接的例子

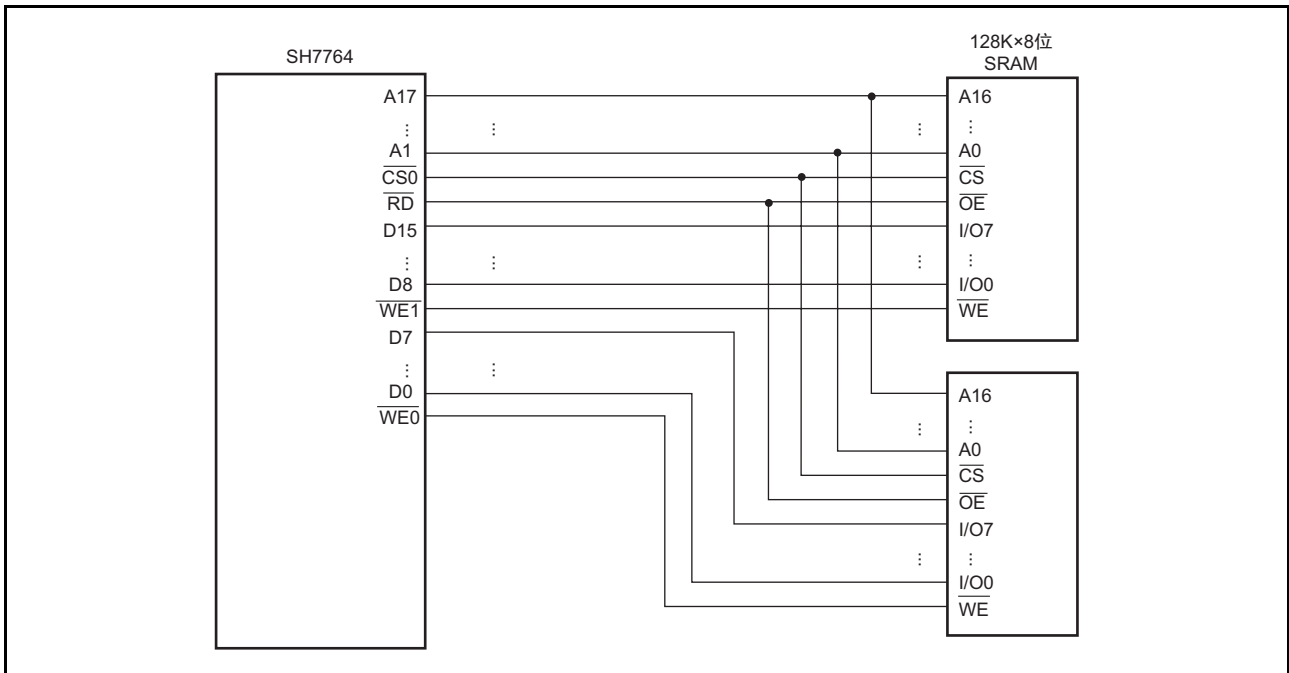


图 11.5 与 16 位数据宽度的 SRAM 连接的例子

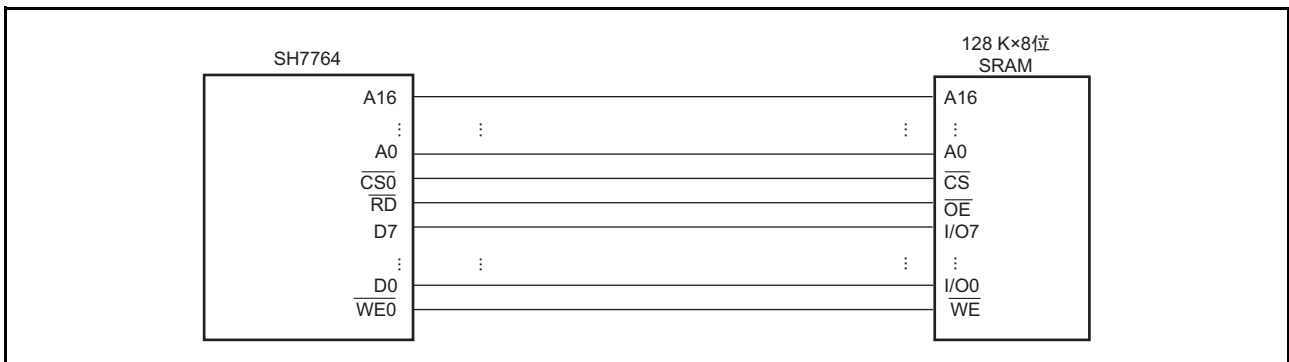


图 11.6 与 8 位数据宽度的 SRAM 连接的例子

11.6.2 等待状态控制

通过 CSnWCR 的设定，能控制 SRAM 接口的等待状态的插入。当 CSnWCR 的 IW 位不为 0 时，根据此等待的指定插入软件等待。通过 CSnWCR，在如图 11.7 所示的 SRAM 接口的等待时序中，作为等待周期，插入指定周期数的 T_w 周期。

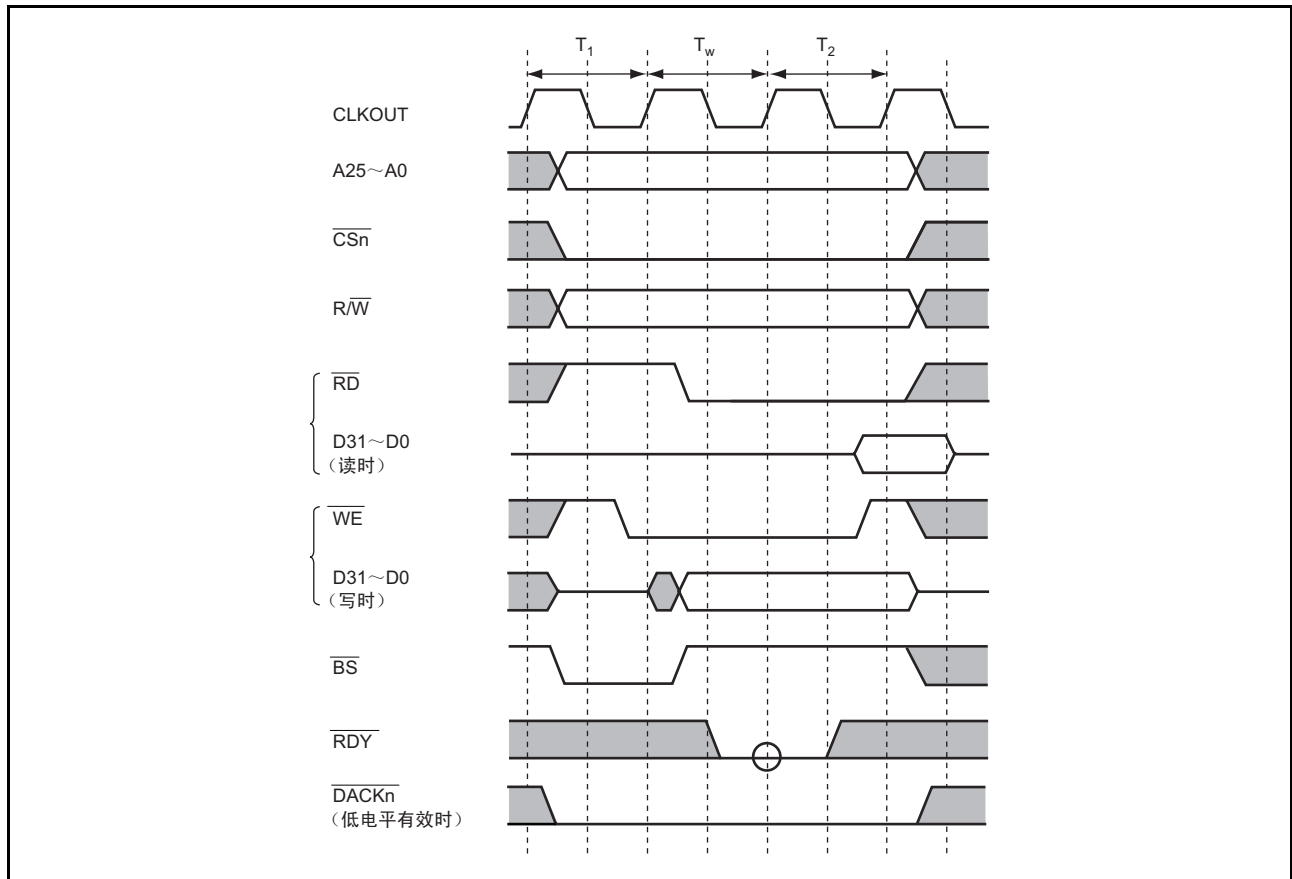


图 11.7 SRAM 接口的等待时序 (只有软件等待)

在通过 CSnWCR 指定软件等待时，也对外部的等待输入 $\overline{\text{RDY}}$ 信号进行采样。 $\overline{\text{RDY}}$ 信号的采样如图 11.8 所示。作为软件等待，指定了 1 个周期的等待，因为在从 T_w 状态转移到 T_2 状态时进行采样，所以即使在 T_1 周期和第 1 个 T_w 周期 $\overline{\text{RDY}}$ 信号有效，也不会带来任何影响。在时钟的上升沿对 $\overline{\text{RDY}}$ 信号进行采样。

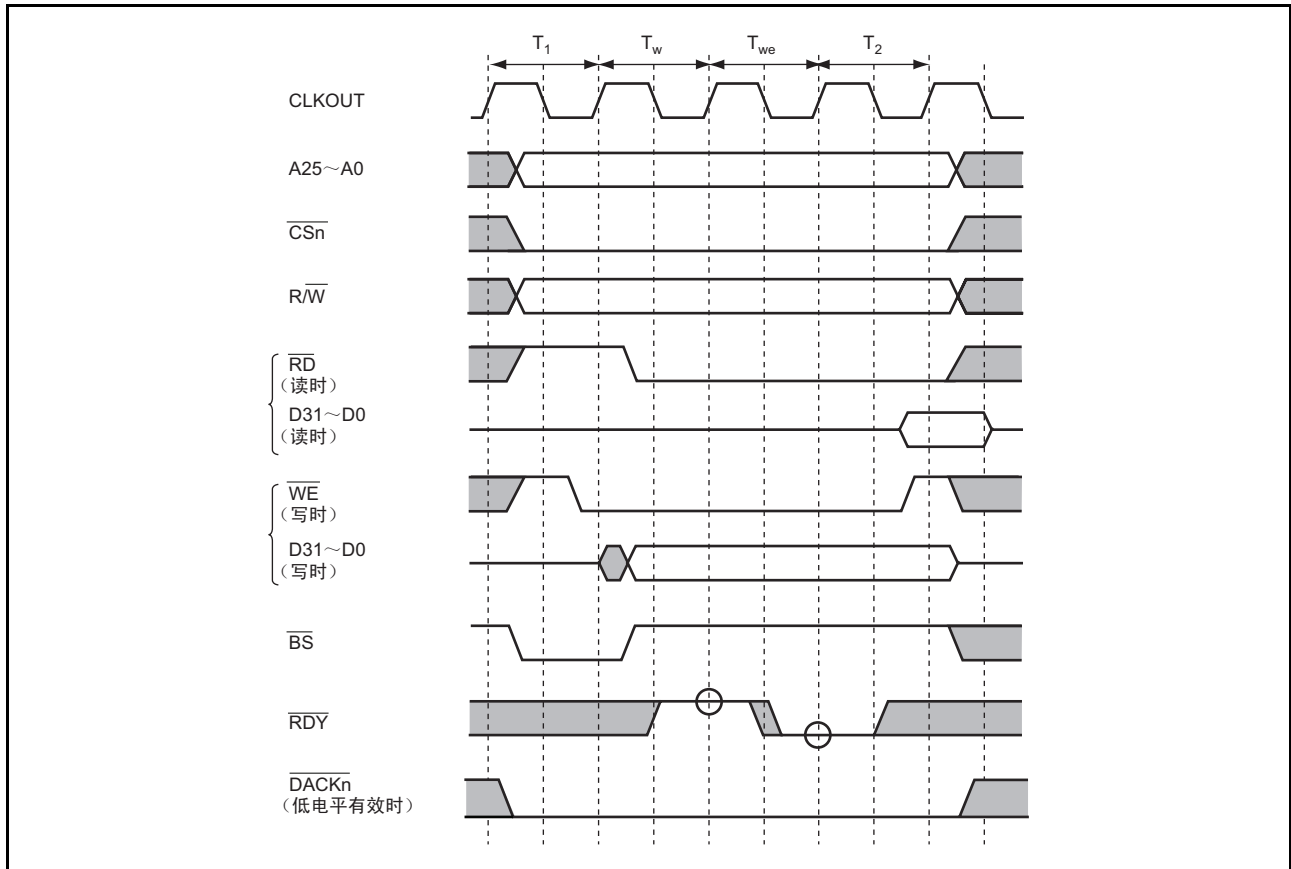


图 11.8 SRAM 接口的等待状态时序
(通过 $\overline{\text{RDY}}$ 信号插入等待状态， $\overline{\text{RDY}}$ 信号为同步输入)

11.6.3 读选通无效时序

对 SRAM 接口，通过 CSnBCR 的 RDSPL (bit7) 的设定，能设定读数据时的选通无效时序。

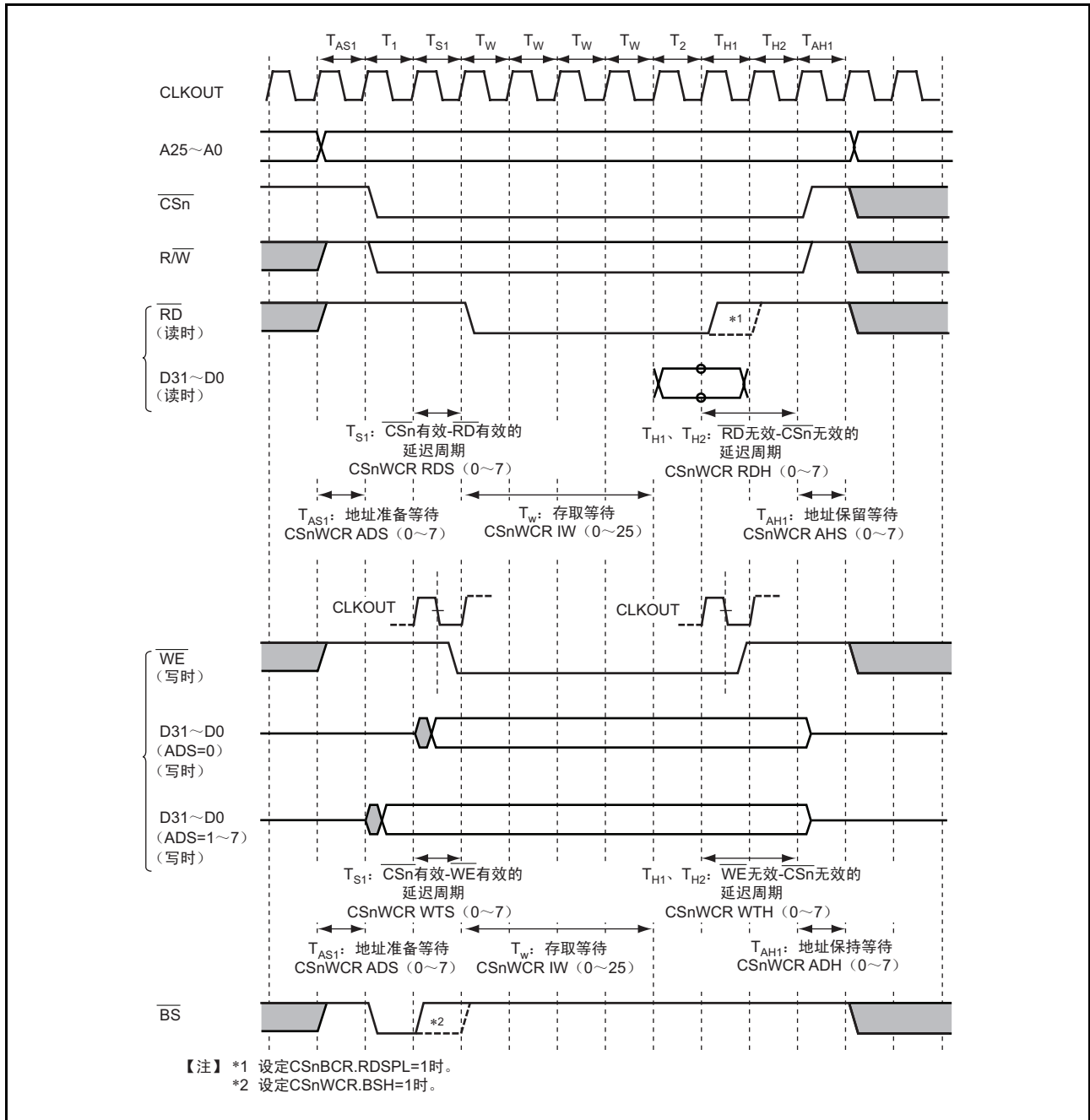


图 11.9 SRAM 接口的等待状态时序 (读选通无效时序的设定)

11.7 SDRAM 接口

11.7.1 SDRAM 直接连接方式

为了能通过 CS 信号选择 SDRAM，通过共用 $\overline{\text{RAS}}$ 等的控制信号，能连接到物理空间的区域 1 和区域 2。

用于直接连接 SDRAM 的控制信号是 $\overline{\text{RAS}}$ 、 $\overline{\text{CAS}}$ 、 $\overline{\text{R/W}}$ 、 $\overline{\text{CS1}}$ 或者 $\overline{\text{CS2}}$ 、 $\overline{\text{DQMLL}}$ 、 $\overline{\text{DQMLU}}$ 、 $\overline{\text{DQMUL}}$ 、 $\overline{\text{DQMUU}}$ 、 $\overline{\text{WE0}} \sim \overline{\text{WE3}}$ (64bit 时) 以及 CKE 信号。除了 $\overline{\text{CS1}}$ 和 $\overline{\text{CS2}}$ 以外，所有信号为各区域共用，除了 CKE 以外，所有信号只在 $\overline{\text{CS1}}$ 或者 $\overline{\text{CS2}}$ 信号有效时才有效。

本模块支持突发读 / 写模式的 SDRAM 的运行模式。数据总线的宽度为 32 位或者 64 位，通过 MIM 的 BW 位设定。

因为通过突发读 / 写模式存取 SDRAM，所以即使在单读时也读取 32 字节的数据，并且即使在单写时也进行 32 字节的数据传送，但是在不必要的传送时， $\overline{\text{DQMLL}}$ 、 $\overline{\text{DQMLU}}$ 、 $\overline{\text{DQMUL}}$ 、 $\overline{\text{DQMUU}}$ 和 $\overline{\text{WE0}} \sim \overline{\text{WE3}}$ 无效。

支持 SDRAM 的命令一览表如表 11.17 所示。

表 11.17 支持 SDRAM 的命令

功能	符号	引脚									
		CKE n-1	CKE n	$\overline{\text{CSn}}$	$\overline{\text{RAS}}$	$\overline{\text{CAS}}$	$\overline{\text{R/W}}$	A [14:13]	A [12:11]	A [10]	A [9:0]
Device deselect	DESL	H	X	H	X	X	X	X	X	X	X
No Operation	NOP	H	X	L	H	H	H	X	X	X	X
Read	READ	H	X	L	H	L	H	V	V	L	V
Read with auto precharge	READA	H	X	L	H	L	H	V	V	H	V
Write	WRITE	H	X	L	H	L	L	V	V	L	V
Write with auto precharge	WRITEA	H	X	L	H	L	L	V	V	H	V
Bank Activate	ACT	H	X	L	L	H	H	V	V	V	V
Precharge select bank	PRE	H	X	L	L	H	L	X	X	L	X
Precharge select all bank	PALL	H	X	L	L	H	L	X	X	H	X
Auto refresh	CBR	H	H	L	L	L	H	X	X	X	X
Self refresh entry from IDLE	SLFRSH	H	L	L	L	L	H	X	X	X	X
Self refresh entry exit	SLFRSHX	L	H	H	X	X	X	X	X	X	X
Power Down entry	PWRDN	H	L	H	X	X	X	X	X	X	X
Power Down exit	PWRDNX	L	H	H	X	X	X	X	X	X	X
Mode register set	MRS	H	X	L	L	L	L	V	V	V	V

64 位和 32 位外部总线时的 8M×16 位的 SDRAM 连接例子分别如图 11.10 和图 11.11 所示。

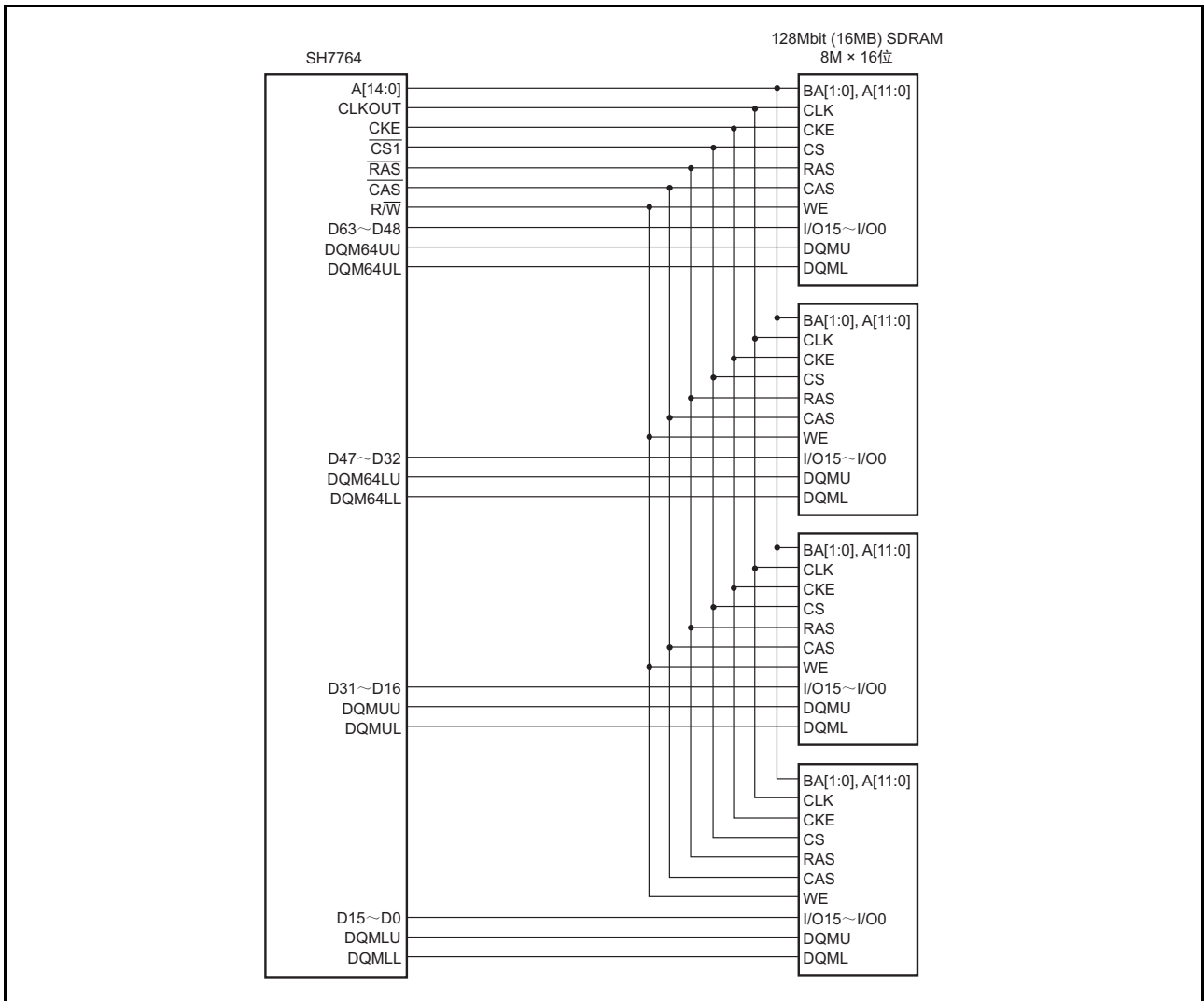


图 11.10 64 位数据总线同步 DRAM 的连接例子 (区域 1)

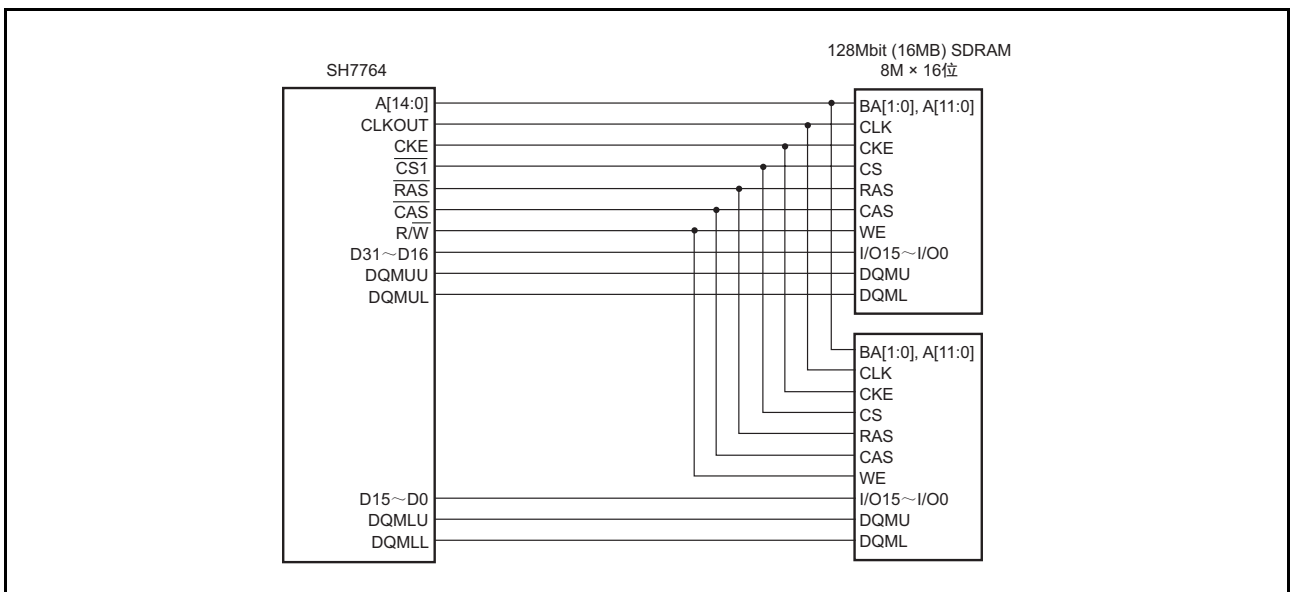


图 11.11 32 位数据总线同步 DRAM 的连接例子 (区域 1)

11.7.2 地址多路复用

根据 MIM 寄存器的 BW[1:0] 位和 SDRAM 寄存器的 SPLIT[3:0] 位的指定，没有外接多路复用电路也能连接 SDRAM，进行地址的多路复用。

多路复用的指定位和输出到地址引脚的位的关系如表 11.8 所示。

不保证输出到 A25 ~ A15 的地址。

表 11.18 SDRAM 总线宽度和地址多路复用的关系

- 外部总线为 32 位的情况

外部总线	SDRAM 地址	BA1	BA0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
32bit	SH7764 地址	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
128Mbit (16MB) 8M×16	Bank (2)	12	13													
	Row (12)				11	24	23	22	21	20	19	18	17	16	15	14
	Col (9)							10	9	8	7	6	5	4	3	2
256Mbit (32MB) 8M×32	Bank (2)	12	13													
	Row (12)				11	24	23	22	21	20	19	18	17	16	15	14
	Col (9)							10	9	8	7	6	5	4	3	2
256Mbit (32MB) 16M×16	Bank (2)	12	13													
	Row (13)			11	25	24	23	22	21	20	19	18	17	16	15	14
	Col (9)							10	9	8	7	6	5	4	3	2

- 外部总线为 64 位的情况

外部总线	SDRAM 地址	BA1	BA0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
64bit	SH7764 地址	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
128Mbit (16MB) 8M×16	Bank (2)	14	13													
	Row (12)				12	25	24	23	22	21	20	19	18	17	16	15
	Col (9)							11	10	9	8	7	6	5	4	3
256Mbit (32MB) 8M×32	Bank (2)	14	13													
	Row (12)				12	25	24	23	22	21	20	19	18	17	16	15
	Col (9)							11	10	9	8	7	6	5	4	3

11.7.3 突发读

突发读的时序图如图 11.12 所示。假设使用在数据宽度为 64 位并且存储体关闭模式的情况，突发长度为 4。

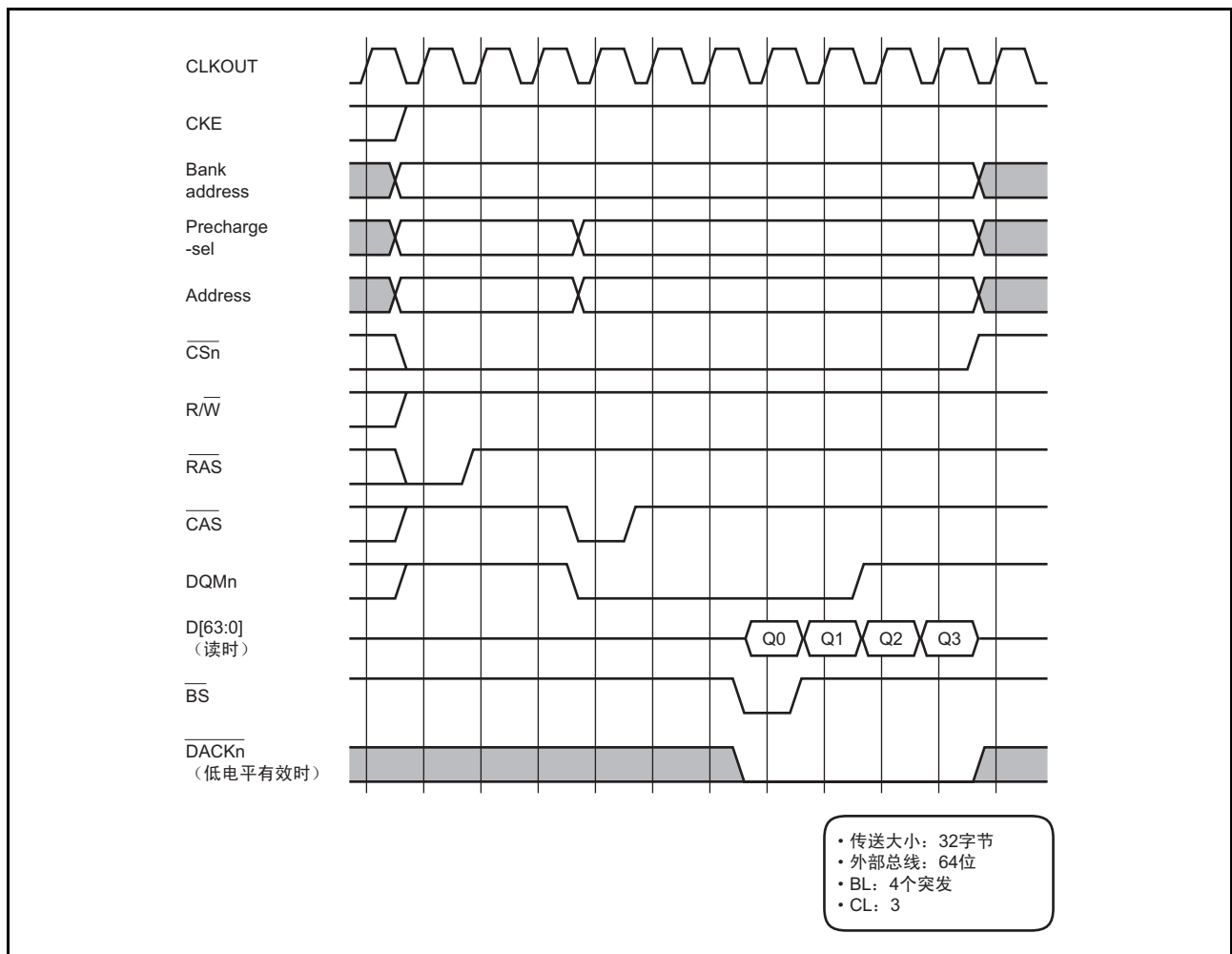


图 11.12 SDRAM 接口的基本时序 (1) 突发读

11.7.4 突发写

突发写的时序图如图 11.13 所示。假设使用在数据宽度为 64 位并且存储体关闭模式的情况，突发长度为 4。

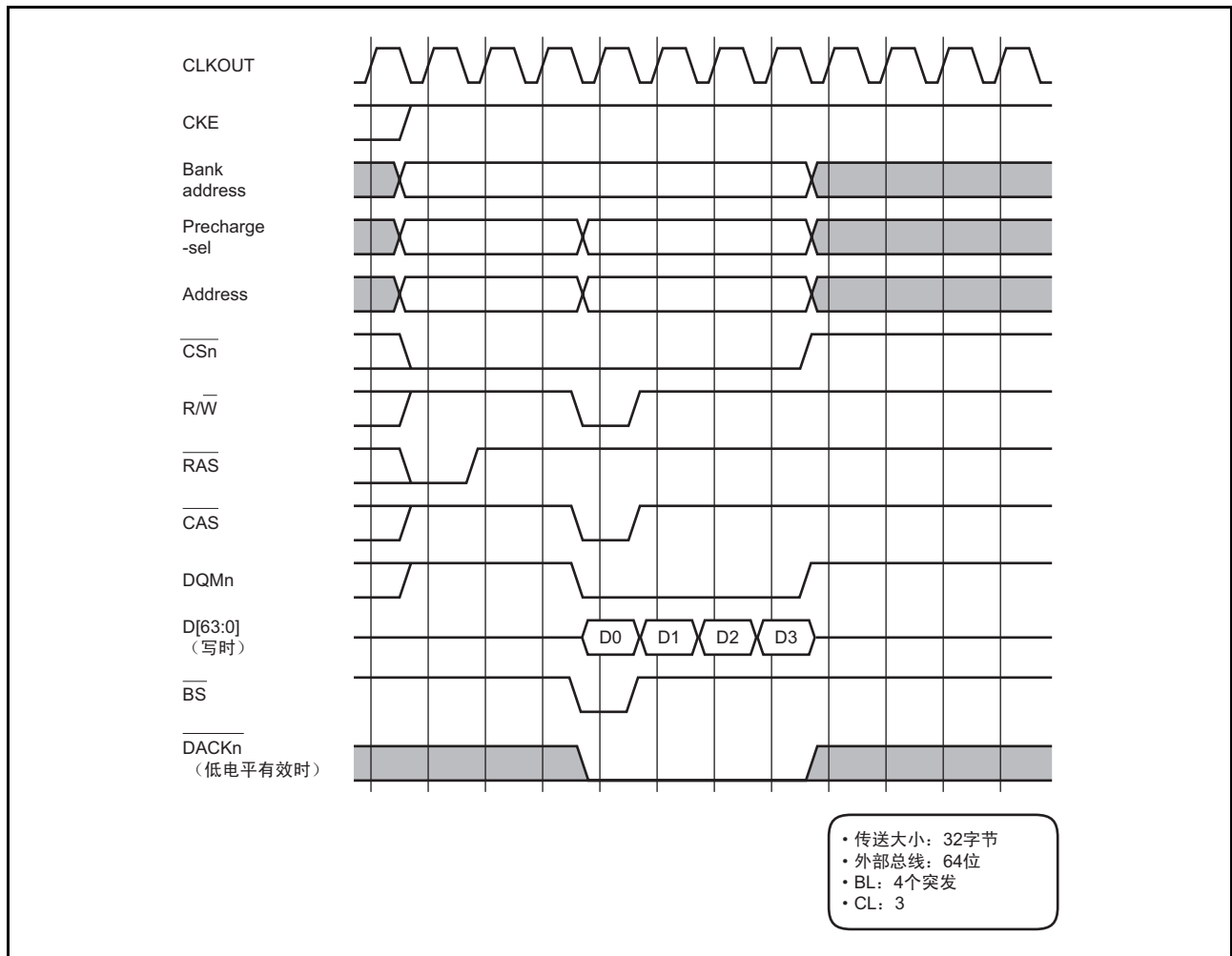


图 11.13 SDRAM 接口的基本时序（2）突发写

11.7.5 单读

单读的时序图表如图 11.14 所示。假设使用在数据宽度为 64 位并且存储体关闭模式的情况，突发长度为 4。单读和突发读相同，读数据的突发长度都为 4，在 MCU 中将数据长度调整为读请求的长度。

在有空周期时，因为会增加存储器存取时间，降低程序执行速度和 DMA 传送速度，所以将数据分配到 32 字节边界，采用能以 32 字节为单位进行传送的数据结构尤为重要。

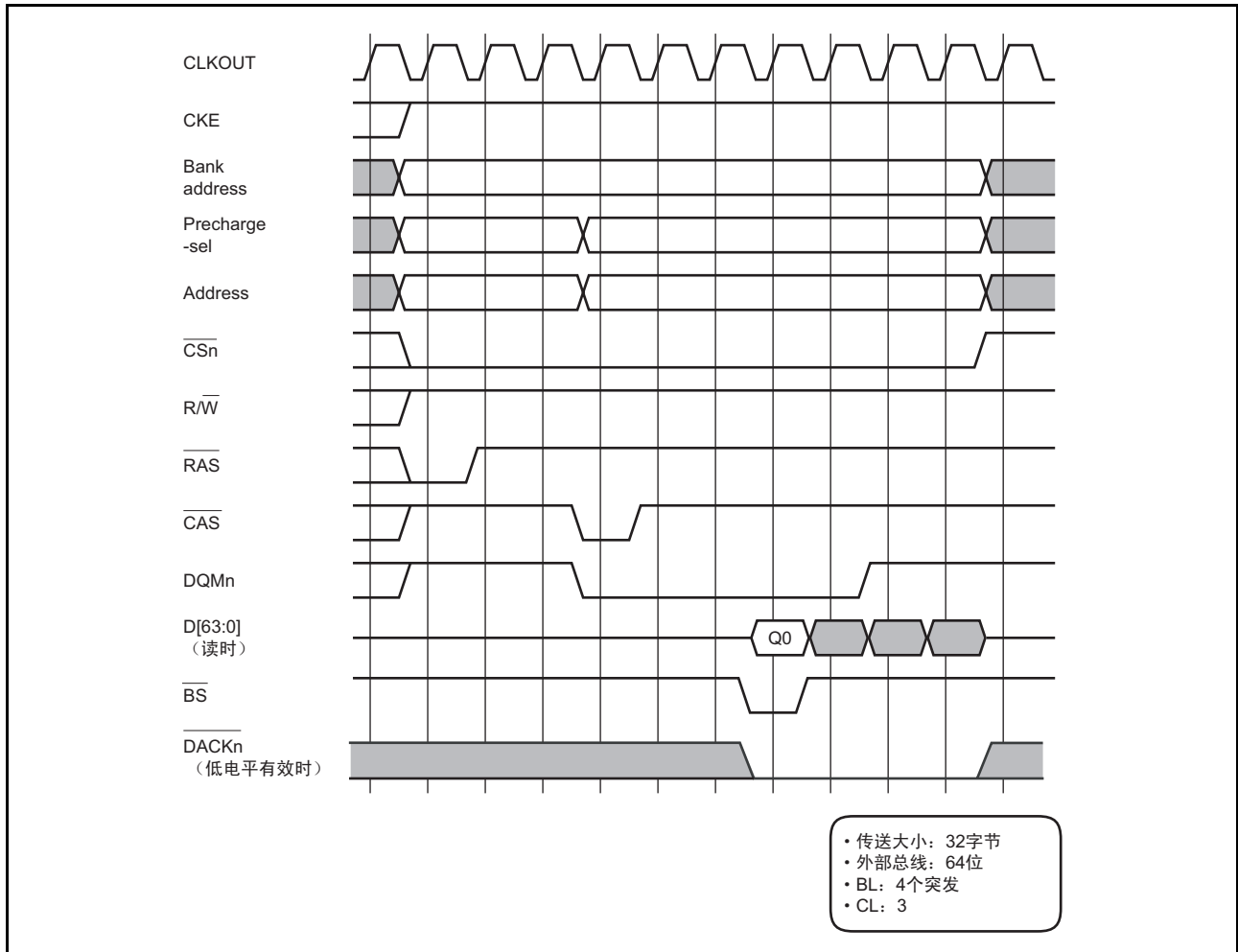


图 11.14 SDRAM 接口的基本时序 (3) 单读

11.7.6 单写

单写的时序图如图 11.15 所示。假设使用在数据宽度为 64 位并且存储体关闭模式的情况，突发长度为 4。单写和突发写相同，写数据的突发长度都为 4，但是在不需要的数据周期中，通过 DQMn 的有效屏蔽写数据。在有空周期时，因为会增加存储器存取时间，降低程序执行速度和 DMA 传送速度，所以将数据分配到 32 字节边界，采用能以 32 字节为单位进行传送的数据结构尤为重要。

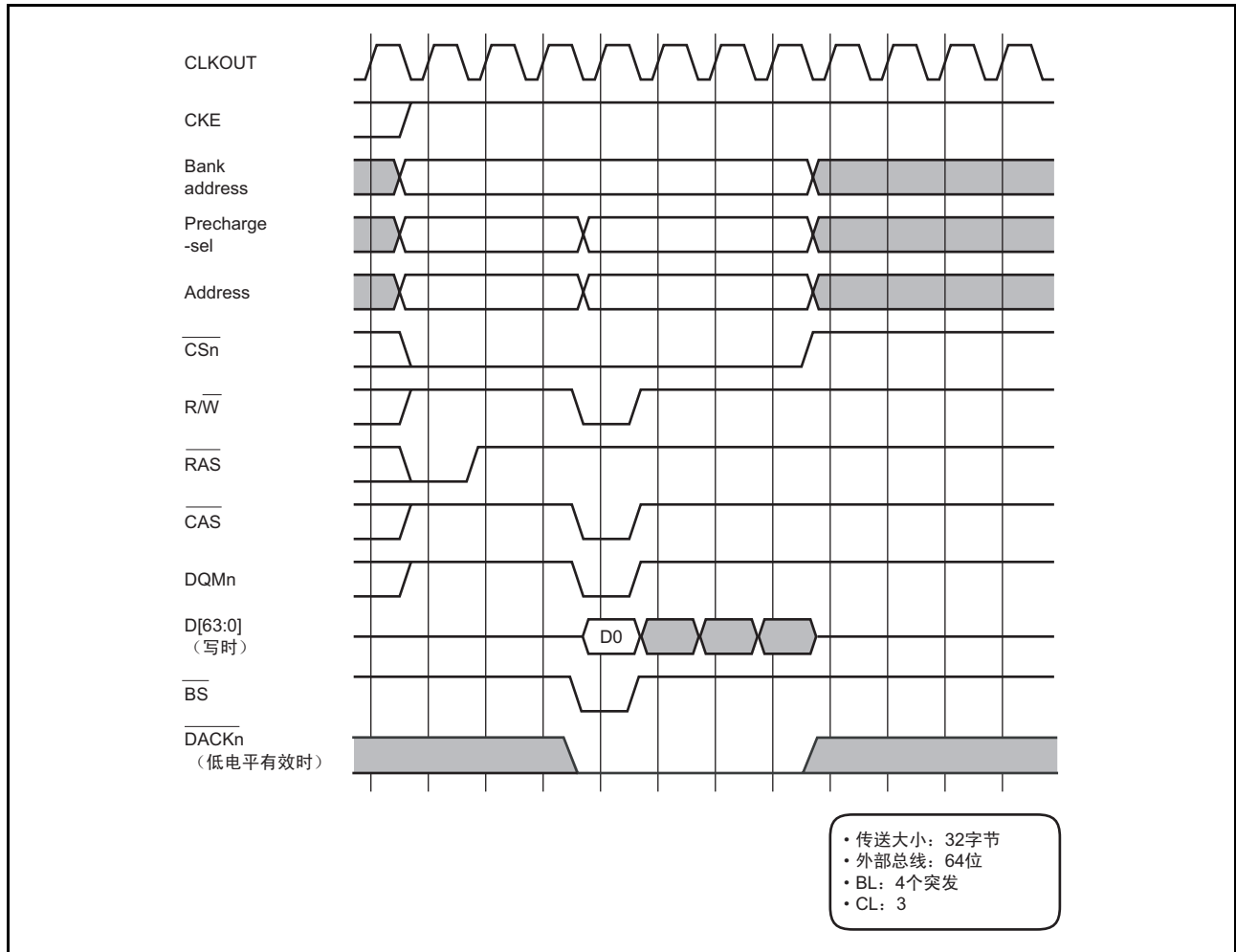


图 11.15 SDRAM 接口的基本时序 (4) 单写

11.7.7 存储体打开模式

为了支持高速存取相同的行地址，具有 SDRAM 的存储体功能。

在 MIM 寄存器的 BOMODE[1:0] 位为 00 时，读 / 写命令使用无自动预充电的命令 (READ 和 WRIT) 进行存取。此时，即使存取结束也不进行预充电。在存取同一个存储体的相同行地址时，不发行 ACTV 命令而能直接发行 READ 或者 WRIT 命令。由于 SDRAM 的内部分为 4 个存储体，所以能预先在各自的存储体中将 1 个行地址设定为有效状态，如果下一次存取不同的行地址，就在最初发行 PRE 命令，进行相应存储体的预充电，在预充电结束后按照 ACTV 命令、READ 或者 WRIT 命令的顺序进行存取。在继续存取不同的行地址时，因为在发生存取请求后进行预充电，所以延长了存取时间。

对各存储体有效状态的时间 t_{RAS} 有限制。在通过执行程序保持该值的周期中，在无法保证不命中高速缓冲而存取其他行地址的情况下，需要设定自动刷新，并且将刷新周期设定为不超过 t_{RAS} 的最大值。从而能保证各存储体的最大有效状态时间的限制。在不使用自动刷新时，需要通过程序使各存储体停留在激活状态的时间部超过所定时间。

对相同行地址和对不同行地址的突发读取周期分别如图 11.16 和图 11.17 所示，对相同行地址和对不同行地址的写周期分别如图 11.18 和图 11.19 所示。

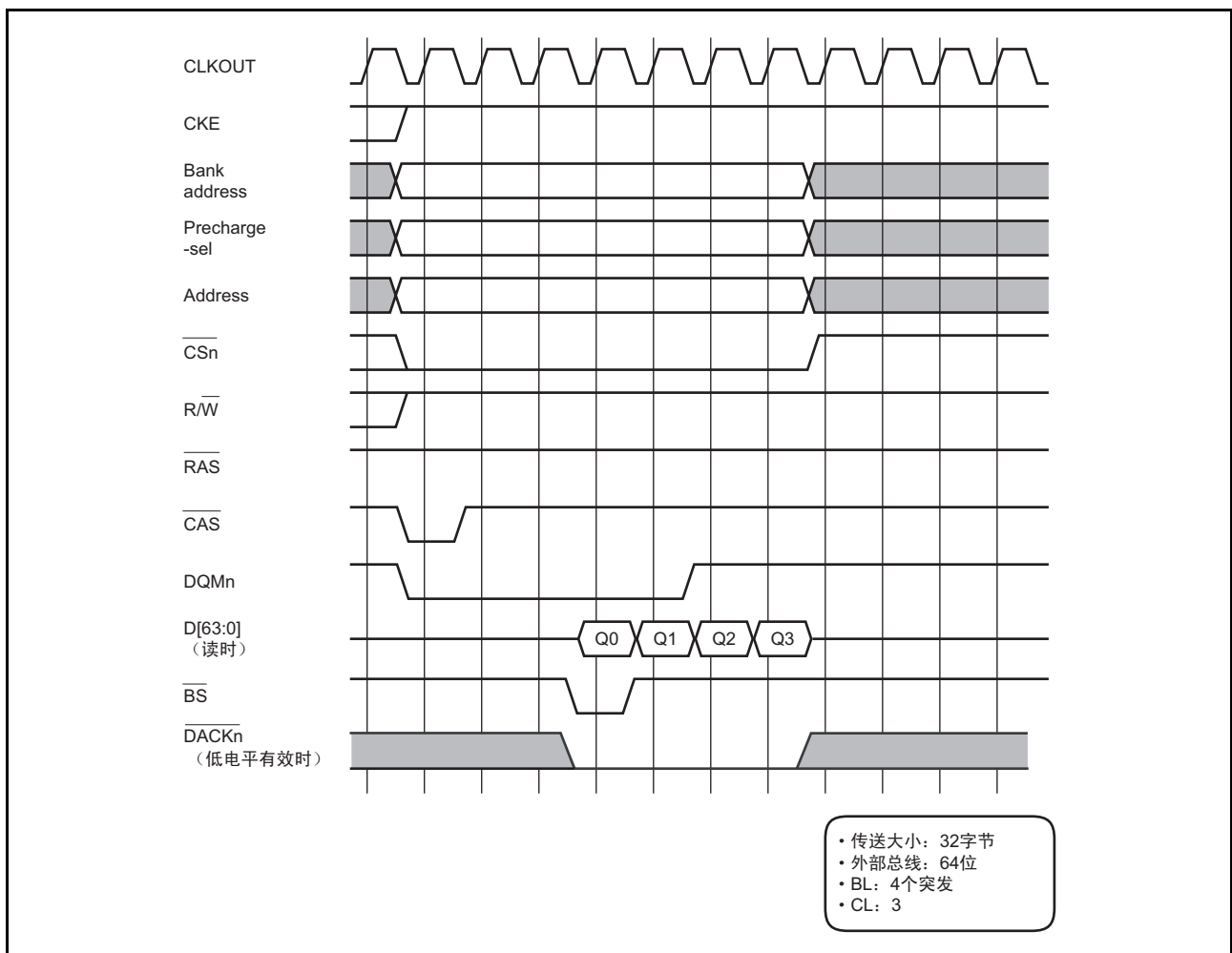


图 11.16 SDRAM 接口的基本时序 (5) 突发读
(存储体打开模式 相同行地址)

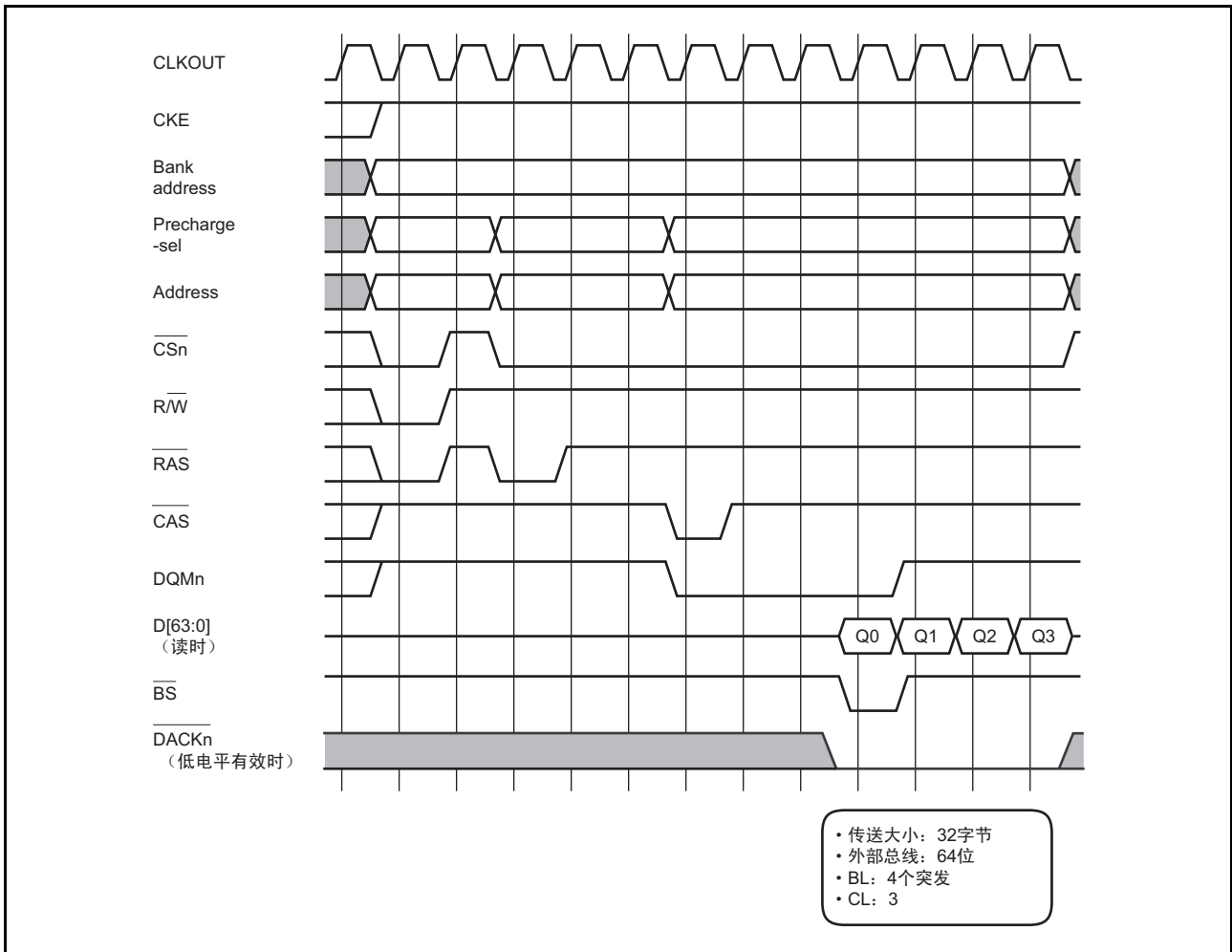


图 11.17 SDRAM 接口的基本时序 (6) 突发读
 (存储体打开模式 不同行地址)

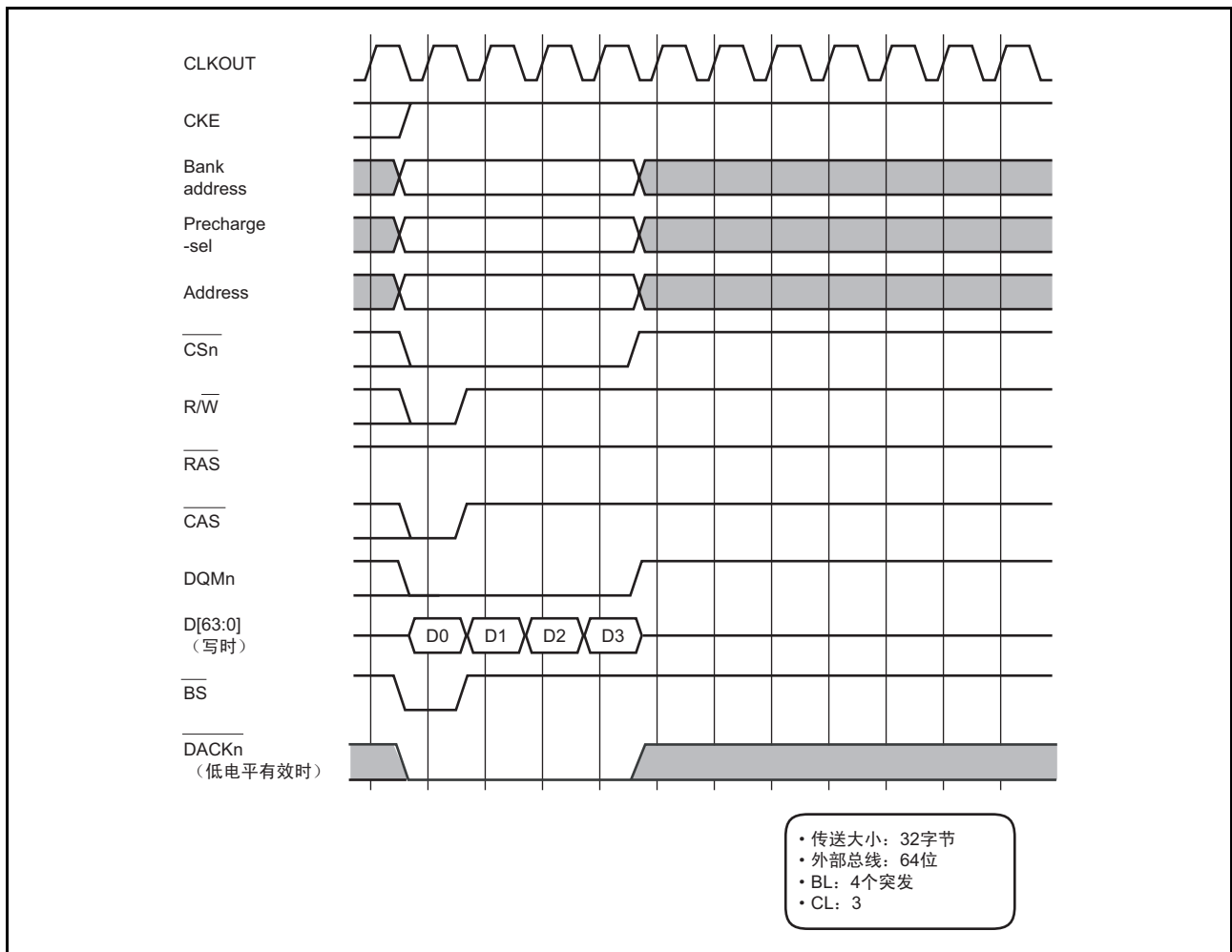


图 11.18 SDRAM 接口的基本时序 (7) 突发写
(存储体打开模式 相同行地址)

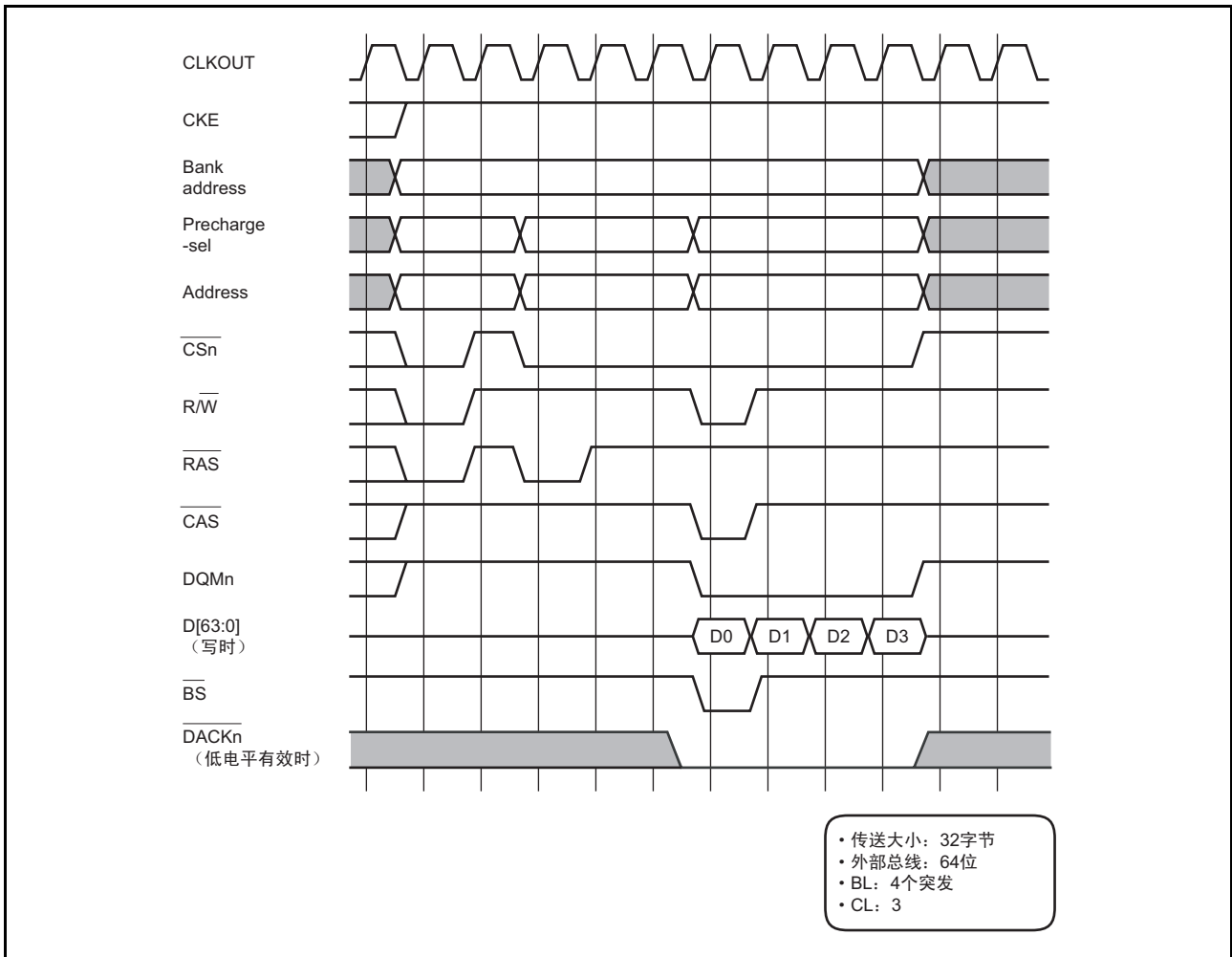


图 11.19 SDRAM 接口的基本时序 (8) 突发写
(存储体打开模式 不同行地址)

11.7.8 刷新

(1) 自动刷新

通过 MIM 寄存器的 DRI[11:0] 位的设定, 决定自动刷新闻隔, 但是如果在设定 DRI[11:0] 位的同时将 DRE 位置 1, 初次自动刷新的间隔就为设定 DRI[11:0] 前的寄存器的值, 第 2 次以后的间隔为 DRI[11:0] 的设定值。为了避免这种情况, 在设定 DRI[11:0] 位时, 必须将 DRE 位置 0, 然后, 将 DRE 位置 1, 这样就能从初次开始通过指定的间隔进行自动刷新。在将 DRE 位置 1 时, 必须给 DRI[11:0] 设定和上一次设定的周期数相同的值。

如果使用 MIM 寄存器的计数器的自动刷新机构不运行, 存储器控制器的延迟调整机构就不正常运行, 所以不能正常进行 SDRAM 的读存取。因此, 必须按照本规格书中记载的 SDRAM 初始化顺序或者从自刷新状态的返回顺序, 在设定 MIM 寄存器使自动刷新机构运行后, 进行 SDRAM 的读存取。

在通过 LBSC 进行 SRAM 存取过程中发生 SDRAM 的自动刷新时, 即使在下次接受了 SRAM 存取请求, ARBT 模块也优先自动刷新而屏蔽 SRAM 的请求。在自动刷新结束后, 经过规定周期的空闲期间, 将下一个请求传送到 LBSC。这也适用于下一个传送请求是 SDRAM 存取的情况。

(2) 自刷新

[转移到自刷新状态]

1. 必须确认对存储器控制器的事务是否结束。
2. 必须通过软件使用 SCR 寄存器的 SMS[2:0] 位, 发行 PALL (全部存储体预充电) 命令。必须在关闭打开的 SDRAM 存储体后, 使用 SCR 寄存器的 SMS[2:0] 位, 发行 CBR (自动刷新) 命令, 对存储器的全部行进行集中刷新 (CBR)。
3. 为了将 SDRAM 转移到自刷新状态, 必须将 SBSC 的 MIM 寄存器的 DRE 位和 RMODE 位置 1。此时 BW[1:0] 位保持 10, DCE 位保持 1。
4. 存储器控制器自动发行自刷新命令, 并且将外部引脚 CKE 设定为低电平。然后, SDRAM 自动变为掉电模式。
5. 能通过读 MIM 寄存器的状态位 SELFS, 判断 SDRAM 是否转移到自刷新模式。

[从自刷新状态的返回]

1. 通过将 MIM 寄存器的 DRE 位和 RMODE 位置 0, 解除自刷新状态。此时, 必须重新设定 BW[1:0] 位。
2. 能通过读 MIM 寄存器的状态位 SELFS, 判断 SDRAM 是否已解除自刷新模式。
3. 解除后, 在 SDRAM 发生请求的时间前, 等待存取。
4. 在变为可存取状态时, 使用 SCR 寄存器的 SMS[2:0] 位, 发行 CBR (自动刷新) 命令, 对存储器的全部行进行集中刷新 (CBR)。
5. 使用 SCR 寄存器的 SMS[2:0] 位, 发行 PALL (全部存储体预充电) 命令。
6. 使用 SCR 寄存器的 SMS[2:0] 位, 发行 CBR 命令。
7. 通过 MIM 寄存器的设定, 让使用计数器的自动刷新机构运行, 周期性地自动刷新。此后, 变为能正常存取的状态。

11.7.9 SDRAM 初始化顺序

初始化顺序记载如下。记载的周期数和时间是例子，所以详细内容请参照各存储器厂商的数据表。

1. 将提供给SDRAM的电源按VDD、VDDQ的顺序接通。
2. 在电源和时钟稳定后，至少将现状维持200 μ s。
3. 必须设定MIM寄存器的DCE位、BW[1:0]位和BOMODE[1:0]位，并且进行SDRAM控制器的允许设定、总线宽度和SDRAM存取模式的设定。此时，为了使DRE位保持0，不能让使用自动计数器的自动刷新机构运行。
4. 设定SDRA寄存器的SPLIT[3:0]位，选择连接存储器的地址多路复用，然后按照连接的存储器时序进行STR寄存器的设定。
5. 使用SCR寄存器的SMS[2:0]位，允许CKE。通过本设定，引脚CKE变为高电平。
6. 使用SCR寄存器的SMS[2:0]位，发行PALL（全部存储体预充电）命令。
7. 使用SCR寄存器的SMS[2:0]位，发行8次CBR（自动刷新）命令。
8. 使用SDMR寄存器，发行MRS命令，设定SDRAM的运行模式。

11.8 存取周期期间的等待

11.8.1 区域 0、3 周期期间的等待

在区域 0、3 空间的连续存取和区域 0、3- 区域 1、2 之间的连续存取时，在 $\overline{\text{CSn}}$ 总线控制寄存器 (CSnBCR) 和总线控制寄存器 (BCR) 指示的存取周期期间之间插入空闲周期。

通过 CSnBCR 的存取周期期间的等待设定位 IWW、IWRWD、IWRRD、IWRWS、IWRRS 和 BCR 的 IRS D，至少插入指定周期的空闲周期。

在区域 0、3- 区域 1、2 之间的连续存取时，至少插入 4 个周期的空闲时间。另外，在存取长度为 8 字节、16 字节或者 32 字节时，在每 4 字节存取的存取周期之间插入等待。

11.8.2 区域 1、2 周期期间的等待

在区域 1、2 空间的连续存取时，相同区域内的读 - 读和写 - 写能以最短 1 个周期进行连续存取。写 - 读和读 - 写能通过 SDRAM 时序寄存器 (STR) 的 WR 和 RW，设定写 - 读和读 - 写的命令间隔。

在区域 1- 区域 2 之间的连续存取时，至少插入 3 个周期的空闲时间。

11.8.3 区域 1、2- 区域 0、3 周期期间的等待

在区域 1、2- 区域 0、3 之间的连续存取时，至少插入 2 个周期的空闲时间。

11.9 总线仲裁

本模块具有 LSI 内部其他模块的存取仲裁功能和对 LSI 外部设备的总线请求的总线仲裁功能。

11.9.1 内部模块存取的仲裁

本模块具有 CPU、各 Pixel 总线模块、LCDC 之间的 SDRAM 和 SRAM 的存取仲裁功能（从 Pixel 总线和 LCDC 不能进行 SRAM 的存取）。SDRAM 的存取要求实时性地读取显示图像数据，为了满足各模块要求的响应时间和带宽，需要决定各自的优先级。

1. 以最优先级（Level0）进行 SDRAM 的刷新和页管理等的 SDRAM 控制。
存储器的刷新根据另外设定的存储器刷新间隔进行。
2. 显示控制器（VDC2）和 LCD 控制器（LCDC）需要高优先级（Level1），对应实时性显示输出的数据传送。
3. 其他的存取为 Level2 和 Level3。
能按各模块选择优先级。

各请求的仲裁方法如图 11.20 所示。假设对 VDC2 和 LCDC 以优先级 Level1 进行循环调度，对 SuperHyway 模块（CPU、DMAC、EtherC、其他）、ATAPI、G2D 命令 / 数据以优先级 Level3 进行循环调度。在仲裁时，通过指定的请求信号进行仲裁。将 SuperHyway 模块和 G2D 模块上升到 Level2 时的例子如图 11.21 所示。

只对实线模块进行仲裁。

具有存取请求屏蔽功能，能在 NMI 的中断处理过程中限制存储器的存取，相对地提高 NMI 中断发生时的 CPU 中断处理的存储器占有率。为了根据 NMI 的用途进行最佳设定，能通过请求屏蔽设定寄存器（RQM）按模块进行设定。

另外，通过寄存器的设定，LCDC 能选择 Level1、Level2 或者 Level3 的仲裁级。

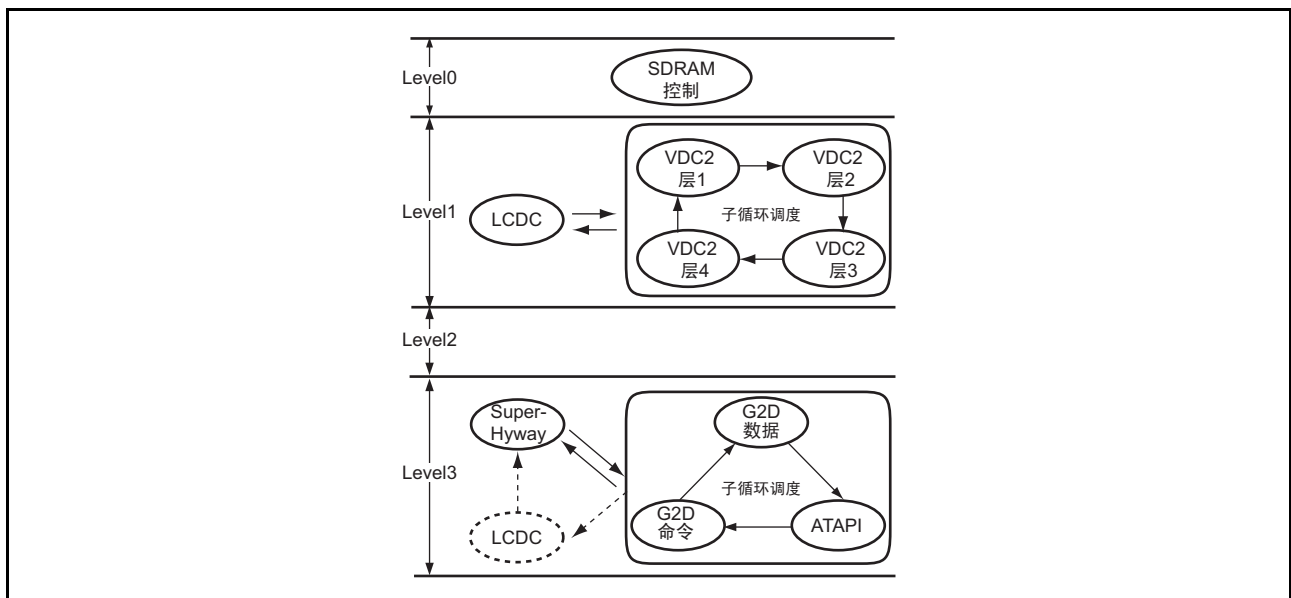


图 11.20 请求的仲裁方法 (1)

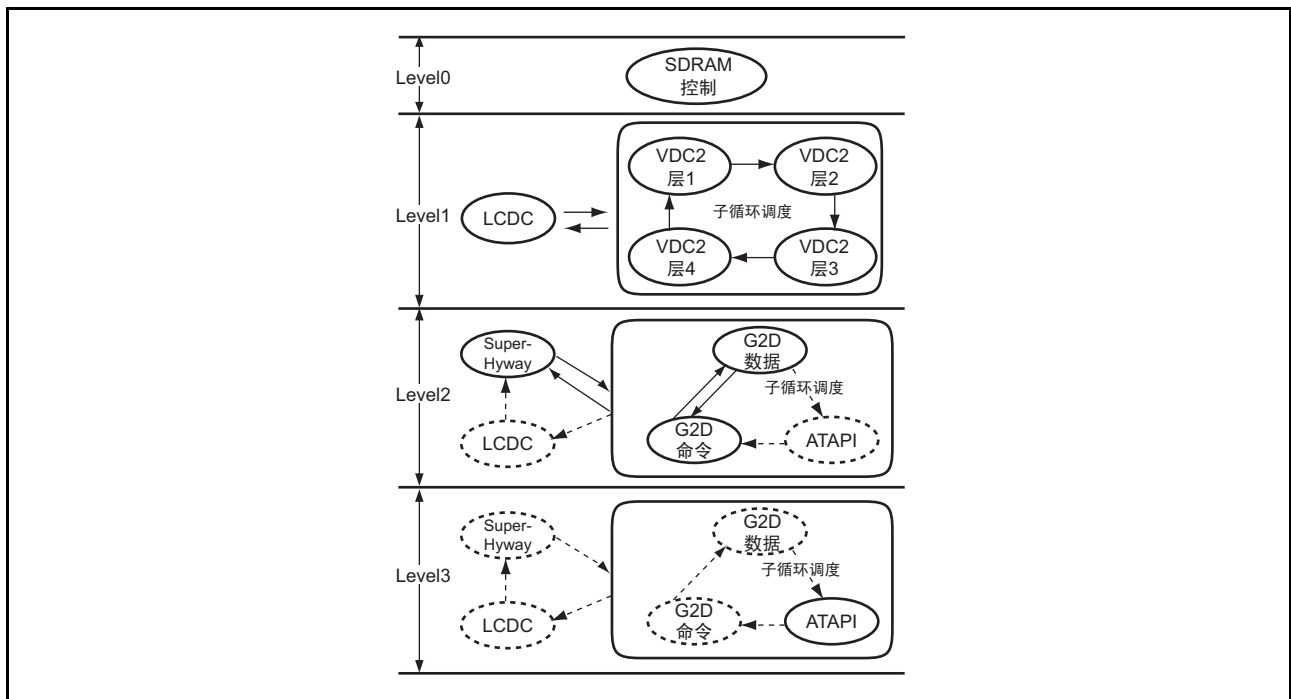


图 11.21 请求的仲裁方法 (2)

优先顺序按照优先级、优先级中的循环调度和子循环调度的顺序逐层进行判断。各判断操作说明如下：

1. 通过优先级的判断

通过优先级的判断是固定的，为 Level0 > Level1 > level2 > Level3。

2. 通过 Level1 循环调度的判断

通过循环调度判断 Level1 中的优先级。

复位后：LCDC > Pixel 总线

选择 LCDC 后：Pixel 总线 > LCDC

选择 Pixel 总线后：LCDC > Pixel 总线

其中，Pixel 总线表示由 Level1 Pixel 总线子循环调度选择的设备。

3. Level1 Pixel 总线子循环调度

当选择 Level1 的 Pixel 总线子循环调度时，仍通过循环调度判断其优先级。

复位后：VDC2（层1）> VDC2（层2）> VDC2（层3）> VDC2（层4）

选择 VDC2（层1）后：VDC2（层2）> VDC2（层3）> VDC2（层4）> VDC2（层1）

选择 VDC2（层2）后：VDC2（层3）> VDC2（层4）> VDC2（层1）> VDC2（层2）

选择 VDC2（层3）后：VDC2（层4）> VDC2（层1）> VDC2（层2）> VDC2（层3）

选择 VDC2（层4）后：VDC2（层1）> VDC2（层2）> VDC2（层3）> VDC2（层4）

4. 通过 Level2 或者 Level3 循环调度的判断

通过循环调度判断 Level2 或者 Level3 中的优先级。能通过仲裁控制寄存器对属于 Level2 或者 Level3 的各设备分别设定是哪个优先级。

当各优先级中有多个设备时，在各优先级中进行循环调度仲裁。

属于 1 个优先级的设备之间的优先级如下：

复位后：SuperHyway > Pixel 总线 > LCDC。其中，Pixel 总线表示由 Level2 或者 Level3 Pixel 总线子循环调度选择的设备。

选择 SuperHyway 后：Pixel 总线 > LCDC > SuperHyway

选择 Pixel 总线后：LCDC > SuperHyway > Pixel 总线

选择 LCDC 后: SuperHyway > Pixel 总线 > LCDC

5. 通过 Level2 或者 Level3 Pixel 总线子循环调度的判断

当选择 Level2 或者 Level3 中的 Pixel 总线子循环调度时, 仍通过循环调度判断其优先级。

复位后: ATAPI > G2D 命令 > G2D 数据

选择 ATAPI 后: G2D 命令 > G2D 数据 > ATAPI

选择 G2D 命令后: G2D 数据 > ATAPI > G2D 命令

选择 G2D 数据后: ATAPI > G2D 命令 > G2D 数据

11.9.2 多阶段仲裁的运行

(1) 3 个阶段的仲裁

经过以下 3 个阶段的仲裁, 进行 1 个模块的存储器存取。

(a) 第 1 阶段的仲裁

根据模块的物理连接关系进行仲裁, 5 种存取请求归纳如下:

A1: Level1 的 Pixel 总线请求。选择 VDC2 (层 1)、VDC2 (层 2)、VDC2 (层 3) 或者 VDC2 (层 4)。

A2: Level2 的 Pixel 总线请求。选择 ATAPI、G2D 命令或者 G2D 数据。

A3: Level3 的 Pixel 总线请求。选择 ATAPI、G2D 命令或者 G2D 数据。

A4: SuperHyWay 请求。因是单一类别, 不进行仲裁, 但是通过仲裁控制寄存器设定为 Level2 或者 Level3。

A5: LCDC 请求。因是单一类别, 不进行仲裁, 但是通过仲裁控制寄存器设定为 Level1、Level2 或者 Level3。

(b) 第 2 阶段的仲裁

B: 通过优先级中的循环调度, 判断并选择一个 A1 ~ A5 中的优先级。

(c) 第 3 阶段的仲裁

C: 进行刷新等 SDRAM 控制和 B 的仲裁。

总是优先 SDRAM 控制。

(2) 仲裁后的存取顺序

在 3 个阶段的仲裁中, 由于在第 1 阶段和第 2 阶段的仲裁中从多个请求选择一个请求, 所以需注意执行顺序。这是因为为了独立进行这些仲裁, 需要具有对应 A1 ~ A5 的存取请求的队列。运行概要说明如下:

1. 进行存储器的控制处理 (C)。

2. 执行所有被优先判断和排列的存取 (B)。

3. 从 A1 ~ A5 中选择一个 (假设为 Ax), 并排列到 B。

4. 对应所选 Ax 的各模块, 进行下一次仲裁, 并进行排列 (假设为 Ay)。

但是, 在下一轮仲裁中, Ay 不一定排列到 B。此判断由第 2 阶段优先判断的结果而定 (假设为 Az)。

按照以上内容, 进行存取的顺序变为 C→B→Ax→Az→⋯→Az→Ay 的顺序。

当不将 A5 设定为 Level1 时, 所有的 Level1 存取请求汇集到 A1, 并且其他优先级的请求无法进入 A1。

因此, Ay 为 Level1 (即 A1), 当 Az 为其他优先级时, 优先选择 Ay (优先于 Az), 变为 C→B→Ax→Ay (A1) 的顺序。另外, 如果 Level1 的存取请求在第 1 阶段的优先仲裁中没有留下, 就不作为 A1 选择, 变为 C→B→Ax→A1→⋯→A1→Ay (A1) 的顺序。

先进行其他的 Level1 存取请求, 但是并不改变在第 1 和第 2 阶段被优先判断的 B 和 Ax 之后的 Level1 请求的接受。

上述多阶段仲裁电路的符号 (A1 ~ 5、B、C) 记载于图 11.22。

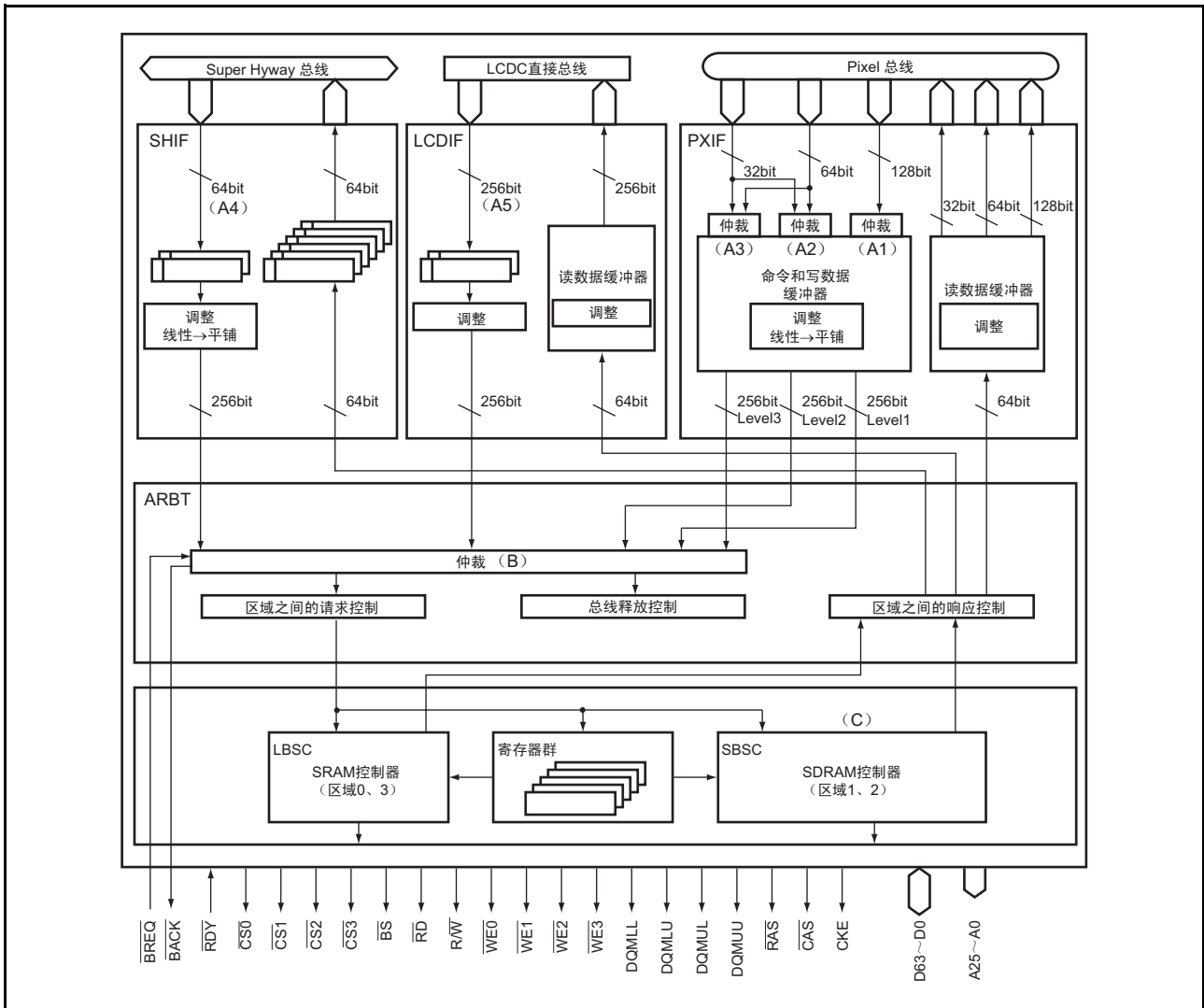


图 11.22 MCU 框图

11.9.3 外部设备的总线请求

本模块具有对外部设备的总线请求提供总线权的总线仲裁功能。

正常状态下，本 LSI 占有总线权，在接受其他外部设备的总线使用请求后释放总线，并允许其他外部设备使用总线。在释放总线时，除了部份信号以外，连接 SRAM/SDRAM 的信号为高阻抗状态。在以下的说明中，将请求总线权的外部设备称为从属设备。

本 LSI 内部存在 CPU 和 DMAC 等的 SuperHyway 总线模块、Pixel 总线连接模块、LCDIC 的总线主控，在连接 SDRAM 并进行刷新控制时，还将刷新请求作为总线主控。在从内部总线主控同时发生请求时，进行“11.9.1 内部模块存取的仲裁”和“11.9.2 多阶段仲裁的运行”所示的仲裁处理。如果从内部总线主控和从属设备同时发生请求时，优先级从高到低顺序为刷新请求、从属设备的请求、内部总线主控的请求。

在主控和从属设备之间进行总线权交换时，为了防止连接设备的误动作，必须在总线释放前将所有的总线控制信号设定为无效状态。在获得总线权时，也要在总线控制信号为无效状态下开始驱动总线。交换总线权的主控和从属设备驱动相同的值，所以能避免输出缓冲器的冲突。

总线权的转让在总线周期的边沿进行。

如果总线释放请求 (\overline{BREQ}) 有效，本模块就在全部处理完已接受的请求后，输出总线使用允许信号 (BACK)，

并释放总线。如果 $\overline{\text{BREQ}}$ 无效，就将 $\overline{\text{BACK}}$ 设定为无效，重新开始总线的使用。

在本 LSI 占有总线权时，如果发生刷新请求，本 LSI 就在结束执行中的总线周期时进行刷新运行。但是，在因数据总线宽度小于存取长度而产生多个总线周期时（例如，在长字存取 8 位总线宽度的存储器时）或者在高速缓存填数和回写等 32 字节传送过程中，需要让刷新运行等待。

由于在总线释放状态下不能进行刷新运行，所以同样也要让刷新请求等待。当发生刷新请求时，本 LSI 就将 $\overline{\text{BACK}}$ 设定为无效，并且向占有当前总线权的外部设备请求总线释放。如果 $\overline{\text{BACK}}$ 无效，外部设备就必须将 $\overline{\text{BREQ}}$ 设定为无效，然后重新取回总线权的本 LSI 开始处理。

在将 $\overline{\text{BACK}}$ 设定为无效后并不能立即取回总线权，所以从拥有和占有总线权时相比，刷新运行需要等待，有可能满足不了刷新间隔。因此，总线释放时，在刷新间隔位（MIM 的 DRI[11:0] 位）设定值的二分之一间隔发生刷新请求。

11.9.4 总线释放和获得的顺序

本 LSI 只要不接受总线请求，就总是占有总线权。

如果本 LSI 接受外部总线请求（ $\overline{\text{BREQ}}$ ）的有效信号（低电平），就在处理完已接受的内部总线主控的请求后将总线权使用允许（ $\overline{\text{BACK}}$ ）信号设定为有效（低电平），进行总线的释放。当没有出现因刷新请求而产生的总线请求时，就在接受到表示从属设备释放总线的 $\overline{\text{BREQ}}$ 的无效信号（高电平）后将 $\overline{\text{BACK}}$ 设定为无效（高电平），重新开始总线的使用。

如果在总线释放状态下出现因存储器刷新请求而产生的总线请求，就将总线使用允许（ $\overline{\text{BACK}}$ ）信号设定为无效，在接受到表示从属设备释放总线的 $\overline{\text{BREQ}}$ 的无效信号后重新开始总线的使用。

在释放总线时，除了 SDRAM 接口的 CKE、总线仲裁的 $\overline{\text{BACK}}$ 以及控制 DMA 传送的 $\overline{\text{DACK0}}$ 、 $\overline{\text{DACK1}}$ 、 $\overline{\text{DTEND0}}$ 、 $\overline{\text{DTEND1}}$ 以外，总线接口相关的全部总线控制输出信号都为高阻抗状态。

另外，给 SDRAM 的有效存储体发行预充电命令，在发行命令结束后进行总线的释放。

具体的总线释放顺序如下：

首先，与时钟的上升沿同步，将总线使用允许信号设定为有效。与此 $\overline{\text{BACK}}$ 有效的下一个时钟的上升沿同步，将地址总线、数据总线和总线控制信号（ $\overline{\text{BS}}$ 、 $\overline{\text{CSn}}$ 、 $\overline{\text{RAS}}$ 、 $\overline{\text{CAS}}$ 、 $\overline{\text{WE}}_n$ 、 $\overline{\text{RD}}$ 、 $\overline{\text{R/W}}$ 、 $\overline{\text{DQM}}_n$ ）设定为高阻抗。这些总线控制信号最迟在设定为高阻抗的 1 个周期前被置为无效。在时钟的上升沿进行总线请求信号的采样。

从属设备重新获得总线权的顺序如下：

如果在时钟的上升沿检测到 $\overline{\text{BREQ}}$ 的无效信号，就立即将 $\overline{\text{BACK}}$ 设定为无效，同时开始驱动总线控制信号，并且在同相的时钟上升沿也开始驱动地址总线。至少在开始驱动总线控制信号的 1 个周期后的时钟上升沿，开始总线的存取。

因为在重新获得总线权后开始刷新运行和总线存取，所以至少需要 2 个周期的 $\overline{\text{BREQ}}$ 无效。

在将 $\overline{\text{BACK}}$ 设定为有效并且释放总线的状态下发生刷新请求时，为了向从属设备请求释放总线权，即使在 $\overline{\text{BREQ}}$ 为有效的状态下也将 $\overline{\text{BACK}}$ 设定为无效。对于用户设计的从属设备，因仲裁可能会提高系统开销，连续产生多次总线存取，对于连续多次存取的合计时间超过刷新周期的从属设备，在检测到 $\overline{\text{BACK}}$ 无效时，必须尽快释放总线权。

另外，即使在总线释放状态下发生内部总线主控的总线存取请求，在取得总线权之前也不接受请求。但是，如果在此种情况下不将 $\overline{\text{BACK}}$ 设定为无效，就不产生总线权释放请求。在发生存储器刷新请求并重新取得总线权时，如果内部总线主控的请求处于接受等待状态，就在刷新运行之后继续进行总线的存取。因此，即使从属设备立即再次发出总线请求，在接受的总线存取结束前可能不会释放总线。

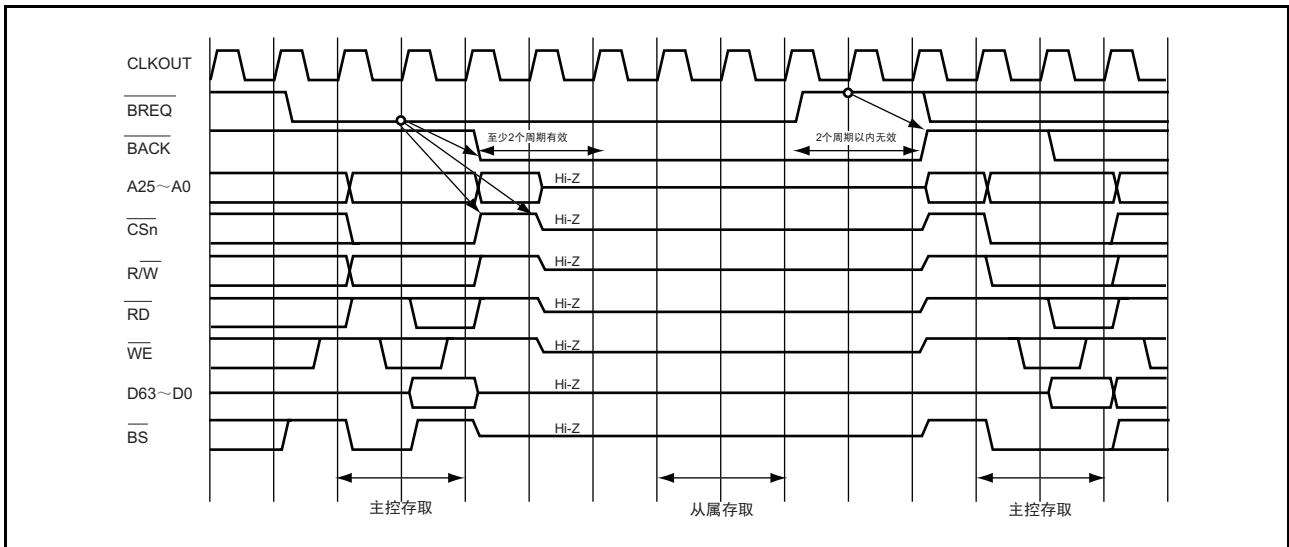


图 11.23 仲裁顺序

11.9.5 主控和从属的协调

为了使主控和从属设备不发生矛盾进行系统资源的控制，需要明确分工，并且在进行低功耗运行时也必须分工。

在设计采用本 LSI 的应用系统时，设想本 LSI 进行初始化、低功耗等所有控制。

上电复位后，本 LSI 在将 BREQ 允许位 (BCR.BREQEN) 置 1 前不接受从属设备的总线请求。

在初始化结束前，为了防止从属设备先存取需要初始化的存储器，必须在初始化结束后将 BREQ 允许位置 1。

11.10 数据的一致性

(1) 存储器存取的一般考察

存储器存取有读和写，从前后关系来看有以下 4 种组合：

(a) WAR (Write after Read) 读后写

顺序是先读后写，但是读到的是后写的数据。也就是说：读请求在总线上等待期间，其他模块写数据，被反映的写数据作为读的数据被返回。

对于上述情况，需要在写操作前先读取数据。通过软件或者系统在读操作结束后再启动写操作，以防出现这种现象。硬件不支持这种功能。

(b) RAR (Read after Read) 读后取

由于读相同数据，所以不产生一致性的问题。

(c) WAW (Write after Write) 写后写

先写的数据覆盖了后写的数据。也就是说：先前的写请求在总线上等待期间，其他模块写数据，然后反映被等待的写请求的写数据。这种情况是写相同的地址，能通过软件或者系统进行管理，但是必须保证在后续的写操作前要将先写的数据反映到存储器。

(d) RAW (Read after Write) 写后读

顺序是先写后读，但是读不到先写的数据。也就是说：先前的写请求在总线上等待期间，其他模块读数据，然后反映被等待的写请求的写数据。这种情况是读写相同的地址，能通过软件或者系统进行管理，但是必须保证在后续的操作前要将先写的数据反映到存储器。

根据上述情况，需要通过某种手段确认写数据的反映情况。

(2) 写存取的反映保证手段

(a) SuperHyway 设备的写操作

为了保证写数据反映到存储器，必须通过 CPU 执行 SYNCO 命令。

此方法能保证将存储在存储器控制器的 SuperHyway 接口部 (SHIF) 的写数据反映到存储器。举例说明“通过 CPU 将显示列表写到主存储器，给 2D 图形引擎发出开始绘图的指示”的情况 (图 11.24)。只在 CPU 连续存取时不产生一致性的问题。

1. CPU 执行“写显示列表 (最后写)”，然后执行“SYNCO 命令”。在 ack 从 SuperHyway 总线返回前 CPU 停止运行。
2. 在存储器控制器的 SuperHyway 接口部 (SHIF) 接收 SuperHyway 总线的写数据。此时，不将 ack 信号返回给 SuperHyway 设备。
3. 从 SHIF 将写数据输出到仲裁器部 (ARBT)。
4. 在与 3 相同的周期，ARBT 进行仲裁运行，如果 SHIF 取得总线权，就将接受的信号 (fin 信号) 返回给 SHIF。
5. ARBT 在下一个周期将写数据输出到 SBSC。
6. 如果 ARBT 接收数据，SHIF 就将 Response 信号返回给 SuperHyway 设备。
7. 如果从 ARBT 接收到数据，SBSC 返回接受的信号 (fin 信号)。

ARBT 在从 SBSC 接受到 fin 信号后停止将写数据输出到 SBSC。在这里，5、6、7 并列运行。

如果 ack 信号返回给 CPU，CPU 就在 SYNCO 命令后继续执行程序 (开始绘图的指示)。

通过以上运行，就能保证“ack 从存储器控制器返回到 SuperHyway 总线时，通过 SBSC 能接收 SuperHyway 的写数据”。按接受顺序处理由 SBSC 接受的写请求和读请求，所以不会出现后来到达 SBSC 的读请求比先前的写数据处理先执行的情况。根据此硬件结构，就能在通过 CPU 执行 SYNCO 命令时保证“SBSC 能接收到写数据”。

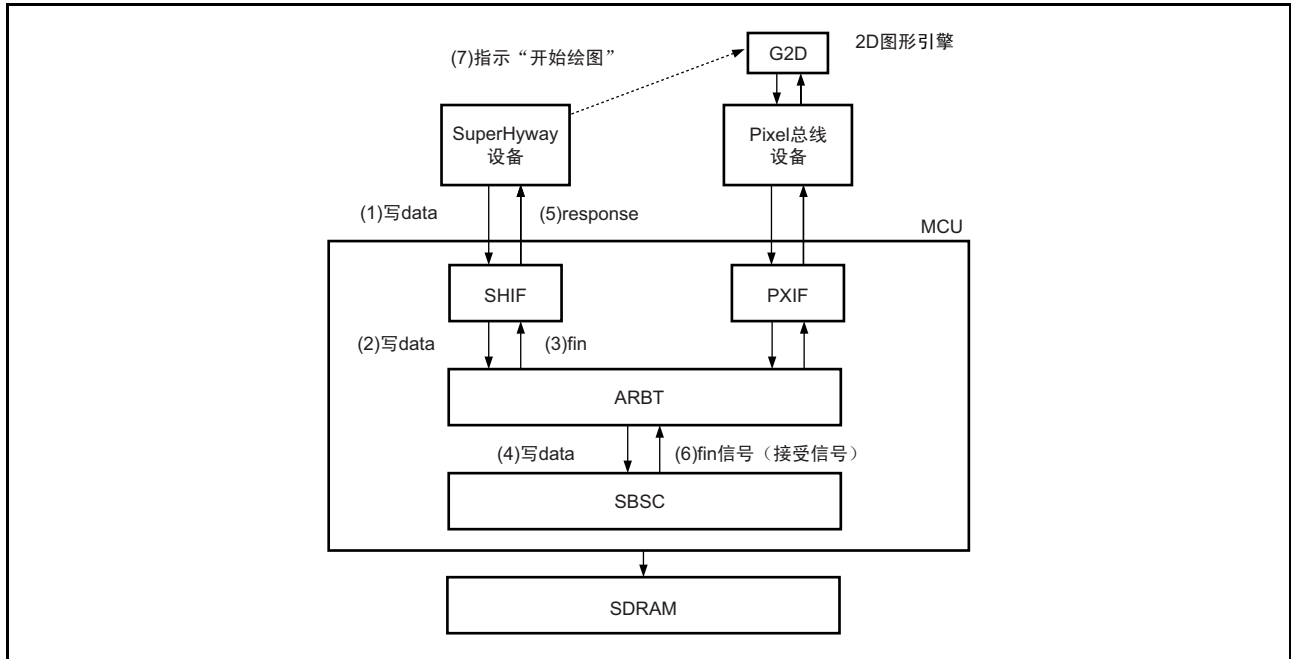


图 11.24 SuperHyway 设备的写数据的反映

(b) Pixel 总线设备的写操作

在由 BSC 接收到写数据后，Pixel 总线设备接受写数据的 send 信号（相当于最后写数据的 ack 信号）。各 Pixel 总线设备通过确认写数据的 send 信号，得知写数据已送到 SBSC。如果写数据送到 SBSC，该写数据就不会被其他写数据超过。

(3) 软件复位的确认手段

有关软件复位，请参照各模块的章节。

为了在 SH7764 中正确反映各模块的软件复位，需要和存储器存取相同的写存取反映确认手段。因此，在转移到软件复位后并且在解除复位前，必须进行以下处理：

1. 当进行软件复位的模块和 CPU 的优先顺序相同时，必须 3 次虚读 SDRAM 的任意区域。
2. 当进行软件复位的模块为 Level2 并且 CPU 为 Level3 时，必须 1 次虚读 SDRAM 的任意区域。
3. 当进行软件复位的模块为 Level3 并且 CPU 为 Level2 时，必须使除了该模块以外的 Level2 和 Level3 模块的 SDRAM 存取全部结束。

11.11 线性和平铺的地址转换

(1) 平铺寻址区

在被连接的 SDRAM 所指定的范围中，本 LSI 能更改寻址（来自 LCDC 的存取不进行线性和平铺的地址转换）。

更改此寻址的区域称为平铺寻址区。平铺寻址区适用于 2 维方向的存取多的图形区。将此区域中的 $32\text{B}\times 16$ 行的图像区称为平铺，1 个平铺分配到连续 512 字节的存储器。因此，即使在存取的位置在行方向频繁变化时，命中 SDRAM 的 1 页的概率也很高。

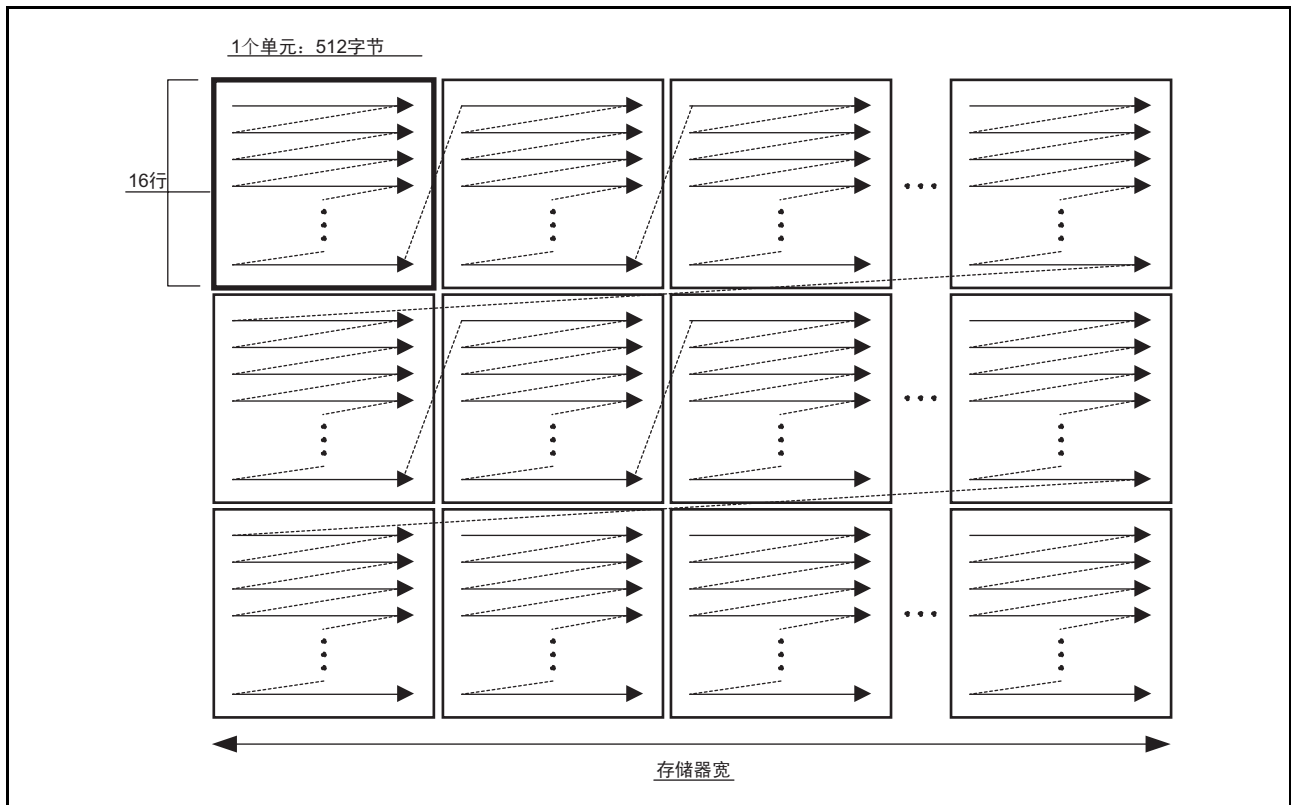


图 11.25 平铺寻址区的数据排列

(2) 线性地址和平铺地址的转换

通过寄存器 LTC[0:7] 和 LTAD[0:7] 指定平铺地址的转换，并且在存储器控制器内部进行转换。能在 SDRAM 中最多定义 8 个平铺寻址区，区域的大小最小为 1M 字节，通过起始地址低位屏蔽的指定，能扩展到 2^mM 字节，并且以此大小的边界地址来定义区域。另外，多个线性平铺转换区域不能相互重叠，如果发生重叠，就不保证地址的转换结果。禁止跨线性区和平铺区的数据传送，否则不保证存储器的内容。

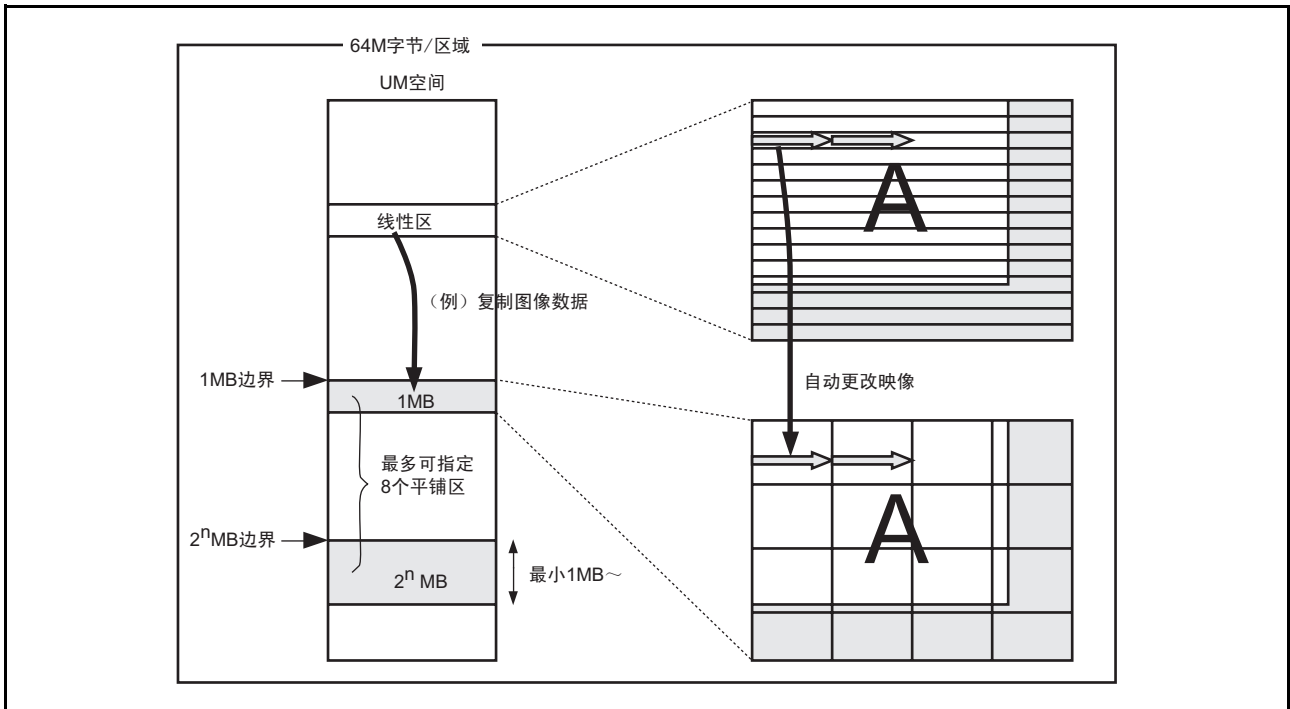


图 11.26 线性平铺转换的概念

(3) 线性和平铺的地址转换方法

从线性地址到平铺地址的规格如表 11.19 所示。

表 11.19 线性地址和平铺地址的转换对应

LT-GBM	MWX	线性地址																	
	像素数 (平铺数)	27-21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4-0
0	512(16)	↑	↑	↑	↑	↑	↑	↑	↑	8	7	6	5	12	11	10	9	↑	
	1024(32)	↑	↑	↑	↑	↑	↑	↑	9	8	7	6	5	13	12	11	10	↑	
	2048(64)	↑	↑	↑	↑	↑	↑	10	9	8	7	6	5	14	13	12	11	↑	
	4096(128)	↑	↑	↑	↑	↑	↑	11	10	9	8	7	6	5	15	14	13	12	↑
1	512(16)	↑	↑	↑	↑	↑	↑	15	14	9	8	7	6	5	13	12	11	10	↑
	1024(32)	↑	↑	↑	↑	↑	↑	10	9	8	7	6	5	14	13	12	11	↑	
	2048(64)	↑	↑	↑	↑	↑	↑	11	10	9	8	7	6	5	15	14	13	12	↑
	4096(128)	↑	↑	↑	↑	↑	12	11	10	9	8	7	6	5	16	15	14	13	↑

LTCn 寄存器的 LTGBM 位表示图形位模式中的图像数据的像素格式。当 LTGBM=0 时表示 8 位 / 像素；当 LTGBM=1 时表示 16 位 / 像素。MWX 表示存储器宽度，通过像素数指定。

存储器宽度为 512、1024、2048 或者 4096。线性平铺转换只支持 8 位 / 像素和 16 位 / 像素。

线性平铺转换的运行如图 11.27 所示，LTAM 为线性平铺转换区起始地址屏蔽寄存器，LTAD 为线性平铺转换区起始地址寄存器。

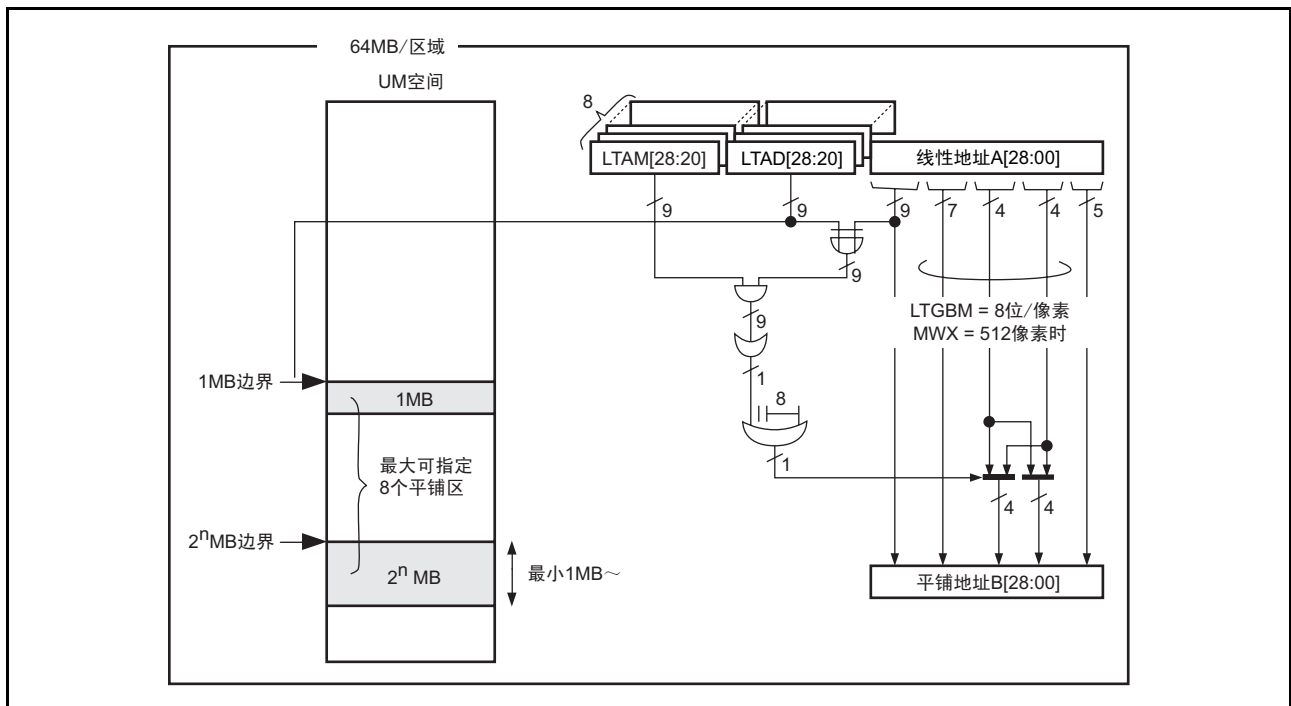


图 11.27 线性平铺转换的运行说明

11.12 使用时的注意事项

11.12.1 刷新

如果转移到刷新待机模式或者硬件待机模式，就不能进行自动刷新。在需要刷新的存储器系统的情况下，必须将存储器设定为自刷新状态后转移到刷新待机模式。

在硬件待机时，由于所有引脚为高阻抗状态，所以不进行存储器的刷新。

11.12.2 外部总线的仲裁

如果转移到刷新待机模式，就不能释放总线权。在要进行外部总线仲裁的系统中，必须在将 $\overline{\text{BREQ}}$ 允许位 (BCR.BREQEN) 置 0 后，转移到刷新待机模式。在将 $\overline{\text{BREQ}}$ 允许位置 1 的状态下转移到刷新待机模式时，不保证运行。

11.12.3 Pixel 总线的传送长度和存取地址边界

如果在 32 字节传送时存取地址不为 32 字节的边界，本模块就对发生存取请求的数据进行最初的存取，以环绕方式对 32 字节边界的数据进行剩余的存取。

必须注意，来自 ATAPI、G2D、LCDC 的存取不进行环绕方式的存取。具体地说，就是需要注意这些模块所持有的寄存器的设定值。详细内容请参照各模块的章节。

第 12 章 直接存储器存取控制器 (DMAC)

本 LSI 内置直接存储器存取控制器 (DMAC)。DMAC 能代替 CPU 在带 DACK (DMA 传送结束通知) 的外部设备、外部存储器、内部存储器、存储器映像外部设备和内部外围模块之间进行高速数据传送。

12.1 特点

- 通道数: 6 个通道 (其中, 通道 0 和通道 1 能接受外部请求)
- 地址空间: 在体系结构上为 4G 字节
- 传送数据长度: 字节、字 (2 字节)、长字 (4 字节)、16 字节、32 字节
- 最大传送次数: 16,777,216 次
- 地址模式: 双地址模式
- 传送请求:
能选择外部请求 (通道 0 和通道 1)、内部外围模块请求 (通道 0~5) 或者自动请求。
能发送内部外围模块请求的模块如下:
SCIF0、SCIF1、SCIF2、USB、FLCTL、SRC
- 总线模式:
能选择周期挪用模式 (正常模式和间歇模式) 或者突发模式
- 优先顺序: 能选择通道优先顺序固定模式或者循环模式
- 中断请求: 能在数据传送半结束、数据传送结束、或者发生地址错误时向 CPU 中断请求
- 外部请求检测: 能选择 DREQ 输入的低/高电平检测或者上升沿/下降沿检测
- 能设定 DMA 传送请求接受信号 (DACK) 和 DMA 传送结束信号 (DTEND) 的有效电平

DMAC 的框图如图 12.1 所示。

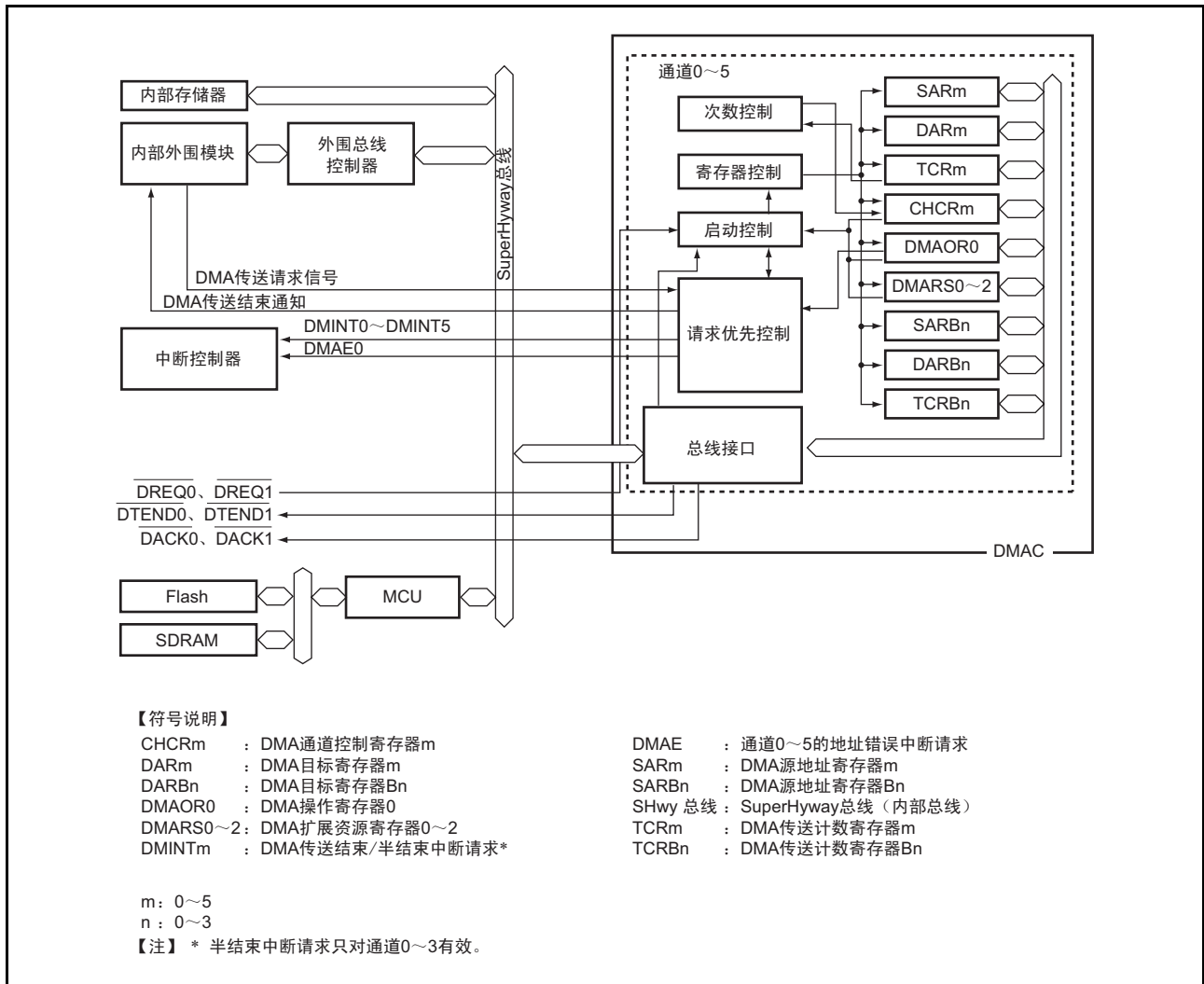


图 12.1 DMAC 框图

12.2 输入 / 输出引脚

有关 DMAC 的外部引脚如下所示。

连接外部总线的引脚的结构如表 12.1 所示。DMAC 有 2 个通道的引脚（通道 0 和通道 1）用于外部总线。

表 12.1 外部总线的引脚结构

通道	名称	略称	输入 / 输出	功能
0	DMA 传送请求	$\overline{\text{DREQ0}}^{*1}$	输入	将 DMA 传送请求从外部设备输入到通道 0。
	DREQ 接受确认	$\overline{\text{DACK0}}^{*2}$	输出	对于 DMA 传送请求，将选通信号从通道 0 输出到外部设备。
	DMA 传送结束通知	$\overline{\text{DTEND0}}^{*2}$	输出	将 DMA 传送结束信号从通道 0 输入到外部设备。
1	DMA 传送请求	$\overline{\text{DREQ1}}^{*1}$	输入	将 DMA 传送请求从外部设备输入到通道 1。
	DREQ 接受确认	$\overline{\text{DACK1}}^{*2}$	输出	对于 DMA 传送请求，将选通信号从通道 1 输入到外部设备。
	DMA 传送结束通知	$\overline{\text{DTEND1}}^{*2}$	输出	将 DMA 传送结束信号从通道 1 输入到外部设备。

【注】 *1 初始值为低电平检测。

*2 初始值为低电平有效。

12.3 寄存器说明

DMAC 的寄存器结构如表 12.2 所示，各处理模式的寄存器状态如表 12.3 所示。

表 12.2 寄存器结构

通道	名称	略称	R/W	P4 区指令	区域 7 地址	存取长度 *3
0	DMA 源地址寄存器 0	SAR0	R/W	H'FF60 8020	H'1F60 8020	32
	DMA 目标地址寄存器 0	DAR0	R/W	H'FF60 8024	H'1F60 8024	32
	DMA 传送计数寄存器 0	TCR0	R/W	H'FF60 8028	H'1F60 8028	32
	DMA 通道控制寄存器 0	CHCR0	R/W*1	H'FF60 802C	H'1F60 802C	32
1	DMA 源地址寄存器 1	SAR1	R/W	H'FF60 8030	H'1F60 8030	32
	DMA 目标地址寄存器 1	DAR1	R/W	H'FF60 8034	H'1F60 8034	32
	DMA 传送计数寄存器 1	TCR1	R/W	H'FF60 8038	H'1F60 8038	32
	DMA 通道控制寄存器 1	CHCR1	R/W*1	H'FF60 803C	H'1F60 803C	32
2	DMA 源地址寄存器 2	SAR2	R/W	H'FF60 8040	H'1F60 8040	32
	DMA 目标地址寄存器 2	DAR2	R/W	H'FF60 8044	H'1F60 8044	32
	DMA 传送计数寄存器 2	TCR2	R/W	H'FF60 8048	H'1F60 8048	32
	DMA 通道控制寄存器 2	CHCR2	R/W*1	H'FF60 804C	H'1F60 804C	32
3	DMA 源地址寄存器 3	SAR3	R/W	H'FF60 8050	H'1F60 8050	32
	DMA 目标地址寄存器 3	DAR3	R/W	H'FF60 8054	H'1F60 8054	32
	DMA 传送计数寄存器 3	TCR3	R/W	H'FF60 8058	H'1F60 8058	32
	DMA 通道控制寄存器 3	CHCR3	R/W*1	H'FF60 805C	H'1F60 805C	32
0 ~ 5 共用	DMA 操作寄存器 0	DMAOR0	R/W*2	H'FF60 8060	H'1F60 8060	16

通道	名称	略称	R/W	P4 区指令	区域 7 地址	存取长度 *3
4	DMA 源地址寄存器 4	SAR4	R/W	H'FF60 8070	H'1F60 8070	32
	DMA 目标地址寄存器 4	DAR4	R/W	H'FF60 8074	H'1F60 8074	32
	DMA 传送计数寄存器 4	TCR4	R/W	H'FF60 8078	H'1F60 8078	32
	DMA 通道控制寄存器 4	CHCR4	R/W*1	H'FF60 807C	H'1F60 807C	32
5	DMA 源地址寄存器 5	SAR5	R/W	H'FF60 8080	H'1F60 8080	32
	DMA 目标地址寄存器 5	DAR5	R/W	H'FF60 8084	H'1F60 8084	32
	DMA 传送计数寄存器 5	TCR5	R/W	H'FF60 8088	H'1F60 8088	32
	DMA 通道控制寄存器 5	CHCR5	R/W*1	H'FF60 808C	H'1F60 808C	32
0	DMA 源地址寄存器 B0	SARB0	R/W	H'FF60 8120	H'1F60 8120	32
	DMA 目标地址寄存器 B0	DARB0	R/W	H'FF60 8124	H'1F60 8124	32
	DMA 传送计数寄存器 B0	TCRB0	R/W	H'FF60 8128	H'1F60 8128	32
1	DMA 源地址寄存器 B1	SARB1	R/W	H'FF60 8130	H'1F60 8130	32
	DMA 目标地址寄存器 B1	DARB1	R/W	H'FF60 8134	H'1F60 8134	32
	DMA 传送计数寄存器 B1	TCRB1	R/W	H'FF60 8138	H'1F60 8138	32
2	DMA 源地址寄存器 B2	SARB2	R/W	H'FF60 8140	H'1F60 8140	32
	DMA 目标地址寄存器 B2	DARB2	R/W	H'FF60 8144	H'1F60 8144	32
	DMA 传送计数寄存器 B2	TCRB2	R/W	H'FF60 8148	H'1F60 8148	32
3	DMA 源地址寄存器 B3	SARB3	R/W	H'FF60 8150	H'1F60 8150	32
	DMA 目标地址寄存器 B3	DARB3	R/W	H'FF60 8154	H'1F60 8154	32
	DMA 传送计数寄存器 B3	TCRB3	R/W	H'FF60 8158	H'1F60 8158	32
0/1	DMA 扩展资源选择器 0	DMARS0	R/W	H'FF60 9000	H'1F60 9000	16
2/3	DMA 扩展资源选择器 1	DMARS1	R/W	H'FF60 9004	H'1F60 9004	16
4/5	DMA 扩展资源选择器 2	DMARS2	R/W	H'FF60 9008	H'1F60 9008	16

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

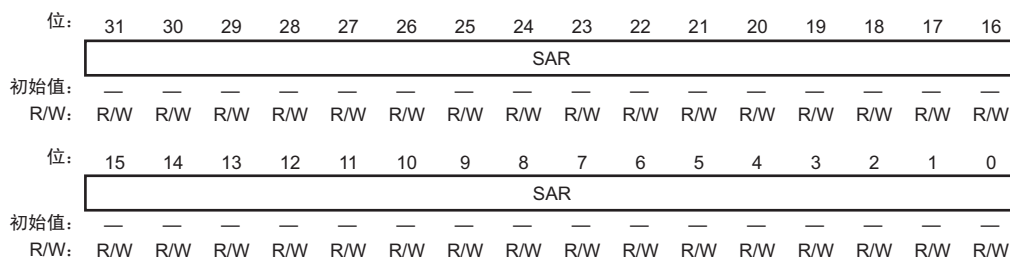
*2 为了清除标志，DMAOR 的 AE 和 NMIF 位只能在读 1 后写 0。

*3 必须以指定的存取长度进行存取。

表 12.3 各处理模式的寄存器状态

通道	名称	略称	上电复位	睡眠	模块待机
0	DMA 源地址寄存器 0	SAR0	不定值	保持	保持
	DMA 目标地址寄存器 0	DAR0	不定值	保持	保持
	DMA 传送计数寄存器 0	TCR0	不定值	保持	保持
	DMA 通道控制寄存器 0	CHCR0	H'4000 0000	保持	保持
1	DMA 源地址寄存器 1	SAR1	不定值	保持	保持
	DMA 目标地址寄存器 1	DAR1	不定值	保持	保持
	DMA 传送计数寄存器 1	TCR1	不定值	保持	保持
	DMA 通道控制寄存器 1	CHCR1	H'4000 0000	保持	保持
2	DMA 源地址寄存器 2	SAR2	不定值	保持	保持
	DMA 目标地址寄存器 2	DAR2	不定值	保持	保持
	DMA 传送计数寄存器 2	TCR2	不定值	保持	保持
	DMA 通道控制寄存器 2	CHCR2	H'4000 0000	保持	保持
3	DMA 源地址寄存器 3	SAR3	不定值	保持	保持
	DMA 目标地址寄存器 3	DAR3	不定值	保持	保持
	DMA 传送计数寄存器 3	TCR3	不定值	保持	保持
	DMA 通道控制寄存器 3	CHCR3	H'4000 0000	保持	保持
0 ~ 5 共用	DMA 操作寄存器 0	DMAOR0	H'0000	保持	保持
4	DMA 源地址寄存器 4	SAR4	不定值	保持	保持
	DMA 目标地址寄存器 4	DAR4	不定值	保持	保持
	DMA 传送计数寄存器 4	TCR4	不定值	保持	保持
	DMA 通道控制寄存器 4	CHCR4	H'4000 0000	保持	保持
5	DMA 源地址寄存器 5	SAR5	不定值	保持	保持
	DMA 目标地址寄存器 5	DAR5	不定值	保持	保持
	DMA 传送计数寄存器 5	TCR5	不定值	保持	保持
	DMA 通道控制寄存器 5	CHCR5	H'4000 0000	保持	保持
0	DMA 源地址寄存器 B0	SARB0	不定值	保持	保持
	DMA 目标地址寄存器 B0	DARB0	不定值	保持	保持
	DMA 传送计数寄存器 B0	TCRB0	不定值	保持	保持
1	DMA 源地址寄存器 B1	SARB1	不定值	保持	保持
	DMA 目标地址寄存器 B1	DARB1	不定值	保持	保持
	DMA 传送计数寄存器 B1	TCRB1	不定值	保持	保持
2	DMA 源地址寄存器 B2	SARB2	不定值	保持	保持
	DMA 目标地址寄存器 B2	DARB2	不定值	保持	保持
	DMA 传送计数寄存器 B2	TCRB2	不定值	保持	保持
3	DMA 源地址寄存器 B3	SARB3	不定值	保持	保持
	DMA 目标地址寄存器 B3	DARB3	不定值	保持	保持
	DMA 传送计数寄存器 B3	TCRB3	不定值	保持	保持
0/1	DMA 扩展资源选择器 0	DMARS0	H'0000	保持	保持
2/3	DMA 扩展资源选择器 1	DMARS1	H'0000	保持	保持
4/5	DMA 扩展资源选择器 2	DMARS2	H'0000	保持	保持

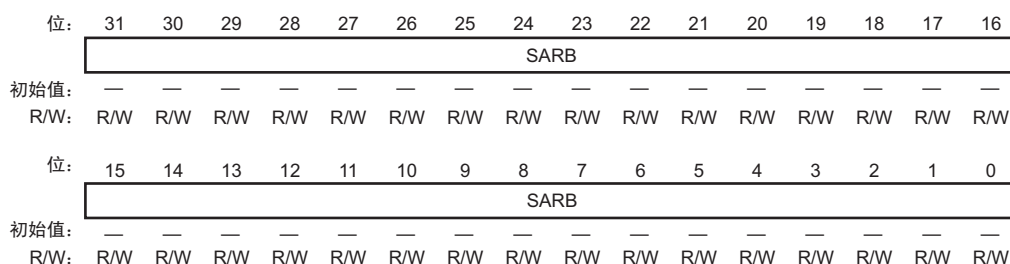
12.3.1 DMA 源地址寄存器 0 ~ 5 (SAR0 ~ SAR5)



SAR 是 32 位可读写寄存器，指定 DMA 传送源的地址，在 DMA 传送中表示下一个传送源地址。在进行字或者长字的数据传送时，必须分别指定字或者长字边界的地址。在以 16 字节或者 32 字节为单位进行传送时，必须给 16 字节或者 32 字节边界设定值。

SAR 的初始值为不定值。

12.3.2 DMA 源地址寄存器 B0 ~ 3 (SARB0 ~ SARB3)

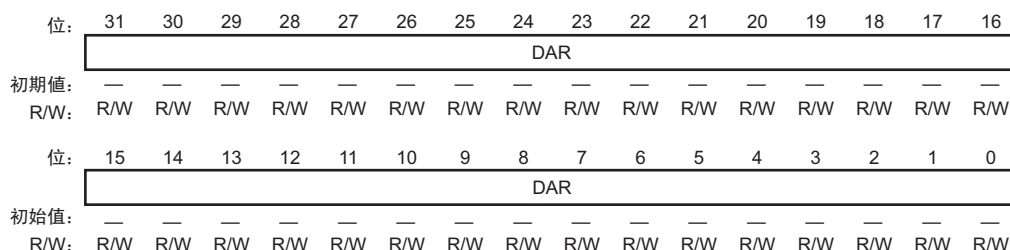


SARB 是 32 位可读写寄存器，在重复 / 重加载模式中给 SAR 指定要重新设定的 DMA 传送源地址。CPU 给 SAR 写的的数据也被写到 SARB。在要设定和 SAR 不同的地址时，必须在写 SAR 后写 SARB。

在进行字或者长字的数据传送时，必须分别指定字或者长字边界的地址。在以 16 字节或者 32 字节为单位进行传送时，必须给 16 字节或者 32 字节边界设定值。

SARB 的初始值为不定值。

12.3.3 DMA 目标地址寄存器 0 ~ 5 (DAR0 ~ DAR5)



DAR 是 32 位可读写寄存器，指定 DMA 传送目标的地址。在 DMA 传送中，表示下一个传送目标地址。

在进行字或者长字的数据传送时，必须分别指定字或者长字边界的地址。在以 16 字节或者 32 字节为单位进行传送时，必须给 16 字节或者 32 字节边界设定值。

DAR 的初始值为不定值。

12.3.4 DMA 目标地址寄存器 B0 ~ 3 (DARB0 ~ DARB3)

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

DARB 是 32 位可读写寄存器，在重复 / 重加载模式中给 DAR 指定要重新设定的 DMA 传送目标地址。CPU 给 DAR 写的的数据也被写到 DARB。在要设定和 DAR 不同的地址时，必须在写 DAR 后写 DARB。

在进行字或者长字的数据传送时，必须分别指定字或者长字边界的地址。在以 16 字节或者 32 字节为单位进行传送时，必须给 16 字节或者 32 字节边界设定值。

DARB 的初始值为不定值。

12.3.5 DMA 传送计数寄存器 0 ~ 5 (TCR0 ~ TCR5)

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

TCR 是 32 位可读写寄存器，指定 DMA 的传送次数。当设定值是 H'0000 0001 时传送次数为 1；当设定值是 H'00FF FFFF 时传送次数为 16,777,215；当设定值是 H'0000 0000 时传送次数为 16,777,216（最大传送次数）。在 DMA 传送中，表示剩余的传送次数。

TCR 的 bit31 ~ 24 的读取值为 0，只能写 0。

TCR 的初始值为不定值。

12.3.6 DMA 传送计数寄存器 B0 ~ 3 (TCRB0 ~ TCRB3)

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

TCRB 是 32 位可读写寄存器，CPU 给 TCR 写的的数据也被写到 TCRB。在使用半结束功能时，用作半结束检测的初始值保持寄存器；在重复模式中，此寄存器给 TCR 指定要重新设定的 DMA 传送次数；在重加载模式

中，用作 DMA 传送次数的设定和传送次数计数器。

在重加载模式中，bit7 ~ 0 用作传送次数计数器，如果值为 0 就更新 SAR/DAR，并且将 TCRB 的 bit23 ~ 16 加载到 bit7 ~ 0。必须给 bit23 ~ 16 设定重新加载前的传送次数。在使用重加载模式时，必须给 bit23 ~ 16 设定 bit7 ~ 0 的相同值，给 bit15 ~ 8 设定 0，并且必须将 CHCR 的 HIE 位置 0，不能使用半结束功能。

TCRB 的 bit31 ~ 24 的读取值为 0，只能写 0。

TCRB 的初始值为不定值。

12.3.7 DMA 通道控制寄存器 0 ~ 5 (CHCR0 ~ CHCR5)

CHCR 是 32 位可读写寄存器，控制 DMA 传送模式。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	LCKN	—	—	RPT[2:0]			—	DO	—	DVMD	TS[2]	HE	HIE	AM	AL
初期值:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R/W	R/W	R/W	R	R/W	R	R/W	R/W	R/(W)*	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DM[1:0]		SM[1:0]		RS[3:0]			DL	DS	TB	TS[1:0]	IE	TE	DE		
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/(W)*	R/W

【注】* 为了清除标志，只能写“0”。

位	位名	初始值	R/W	说明
31	—	0	R	保留位 读取值为 0，只能写 0。
30	LCKN	1	R/W	总线锁定信号禁止位 在 SHwy 总线读指令时，设定是输出还是抑止 SHwy 总线锁定信号。 此位只在周期挪用模式时有效。 通过抑止 SHwy 总线锁定信号，能接受 DMAC 以外的总线主控的总线请求，提高整个系统的总线使用效率。在突发模式时，此位必须置 0； 在内部模块请求模式中选择 USB 时，此位必须置 1。 0：允许输出总线锁定信号 1：禁止输出总线锁定信号
29 ~ 28	—	全 0	R	保留位 读取值为 0，只能写 0。
27 ~ 25	RPT[2:0]	全 0	R/W	DMA 设定更新指定位 此位只在 CHCR0 ~ CHCR3 有效。 000：正常模式 001：重复模式：重复 SAR/DAR/TCR 010：重复模式：重复 DAR/TCR 011：重复模式：重复 SAR/TCR 100：保留（禁止设定） 101：重加载模式：重加载 SAR/DAR/TCR 110：重加载模式：重加载 DAR/TCR 111：重加载模式：重加载 SAR/TCR
24	—	0	R	保留位 读取值为 0，只能写 0。

位	位名	初始值	R/W	说明
23	DO	0	R/W	<p>DMA 溢出位</p> <p>选择是通过溢出 0 还是通过溢出 1 检测 DREQ。此位只有 CHCR0 和 CHCR1 有效。</p> <p>0: 通过溢出 0 检测 DREQ 1: 通过溢出 1 检测 DREQ</p>
22	—	0	R	<p>保留位</p> <p>读取值为 0，只能写 0。</p>
21	DVMD	0	R/W	<p>分割传送模式指定位</p> <p>在 FLCTL 和存储器之间，指定以 16byte 为单位进行 DMA 传送。在不使用 FACTL 模块时，此位必须置为 0。</p>
20	TS[2]	0	R/W	<p>DMA 传送长度指定位</p> <p>和 TS[1:0] 一起设定 DMA 传送长度。当传送源或者传送目标是指定传送长度的内部外围模块的寄存器时，此传送长度必须和存取长度相同。对于由 SAR 或者 DAR 设定的传送源或者传送目标，传送长度必须和地址边界相同。</p> <p>TS[2:0]</p> <p>000: 字节单位 001: 字 (2 字节) 单位 010: 长字 (4 字节) 单位 011: 16 字节单位 100: 32 字节单位 上述以外: 禁止设定</p> <p>【注】 进行在将连接外围总线的外围模块 (FLCTL 和 USB 除外) 作为传送源或者传送目标进行传送时，必须将 TS[2:0] 的传送长度设定为字节或者长字。FLCTL 能设定为 16 字节，USB 能设定为 32 字节。在进行 FLCTL 的 16 字节的传送时，必须将 DVMD 位置 1。</p>
19	HE	0	R/(W)*	<p>半结束标志位</p> <p>如果将 HIE (bit18) 置 1 并且传送次数为传送开始前设定的 TCR 值的 1/2 (右移 1 位的值)，HE 就变为 1。</p> <ul style="list-style-type: none"> 偶数次时，如果传送次数为传送前设定的 $TCR \div 2$，HE 就被置 1。 奇数次时，如果传送次数为 (传送前设定的 $TCR - 1) \div 2$，HE 就被置 1。 最大传送次数时，如果传送次数为 8,388,608 次 (H'0080 0000)，HE 就被置 1。 <p>在传送次数变为传送开始前设定的值的 1/2 之前，当因 NMI 中断或者地址错误而传送结束时以及因清除 DE 位 (bit0) 或者 DMAOR 的 DME 位 (bit0) 而传送结束时，HE 位就不被置位。在 HE 位被置 1 后，当因 NMI 中断或者地址错误而传送结束时以及因清除 DE 位 (bit0) 或者 DMAOR 的 DME 位 (bit0) 而传送结束时，HE 位就保持不变。在清除 HE 位时，必须在读 HE 位的 1 后写 0。此位只有 CHCR0 ~ CHCR3 有效。</p> <p>0: 在 DMA 传送中或者发生 DMA 传送中断时并且 $TCR >$ 传送前设定的 $TCR \div 2$</p> <p>[清除条件] 在读 HE 位的 1 后写 0。</p> <p>1: $TCR =$ 传送前已设定的 $TCR \div 2$</p>

位	位名	初始值	R/W	说明
18	HIE	0	R/W	<p>半结束允许位</p> <p>指定在传送次数为传送开始前设定的 TCR 值的 1/2 时是否向 CPU 请求中断。在 HIE 位为 1 时, 如果将 HE 位置位, 就向 CPU 请求中断。在设定为重加载模式时, 必须将此位置 0。此位只有 CHCR0 ~ CHCR3 有效。</p> <p>0: 禁止半结束中断 1: 允许半结束中断</p>
17	AM	0	R/W	<p>应答模式位</p> <p>在双地址模式中, 选择是在数据读周期还是在数据写周期输出 DACK。此位只有 CHCR0 和 CHCR1 有效。</p> <p>0: 在读周期输出 DACK 1: 在写周期输出 DACK</p>
16	AL	0	R/W	<p>应答电平位</p> <p>指定 DACK 和 DTEND 信号是高电平有效还是低电平有效。此位只有 CHCR0 和 CHCR1 有效。</p> <p>0: DACK 和 DTEND 输出有效电平为低电平 1: DACK 和 DTEND 输出有效电平为高电平</p>
15、14	DM[1:0]	00	R/W	<p>目标地址模式位</p> <p>指定 DMA 传送目标地址的增减。</p> <p>00: 固定目标地址 01: 目标地址递增 在字节单位传送时 +1 在字单位传送时 +2 在长字单位传送时 +4 在 16 字节单位传送时 +16 在 32 字节单位传送时 +32 10: 目标地址递减 在字节单位传送时 -1 在字单位传送时 -2 在长字单位传送时 -4 在 16/32 字节单位传送时禁止设定 11: 禁止设定</p>
13、12	SM[1:0]	00	R/W	<p>源地址模式位</p> <p>指定 DMA 传送源地址的增减。</p> <p>00: 固定源地址 01: 源地址递增 在字节单位传送时 +1 在字单位传送时 +2 在长字单位传送时 +4 在 16 字节单位传送时 +16 在 32 字节单位传送时 +32 10: 源地址递减 在字节单位传送时 -1 在字单位传送时 -2 在长字单位传送时 -4 在 16/32 字节单位传送时禁止设定 11: 禁止设定</p>

位	位名	初始值	R/W	说明
11 ~ 8	RS[3:0]	全 0	R/W	<p>资源选择位</p> <p>指定传送请求源。必须在 DMA 允许位 (DE) 为 0 的状态下改变传送请求源。</p> <p>0000: 外部请求、双地址模式</p> <p>0100: 自动请求</p> <p>1000: 内部外围模块请求</p> <p>通过 DMA 扩展源选择器 (DMARS0 ~ 2) 选择</p> <p>上述以外: 禁止设定</p> <p>【注】 外部请求的指定只有 CHCR0 和 CHCR1 有效。 CHCR2 ~ CHCR5 禁止设定外部请求。 内部外围模块请求的指定只有 CHCR0 ~ CHCR5 有效。</p>
7 6	DL DS	0 0	R/W R/W	<p>DREQ 电平和 DREQ 边沿的选择位</p> <p>选择 DREQ 输入的检测方法和检测电平。</p> <p>此位只有 CHCR0 和 CHCR1 有效。另外, 对于通道 0 和通道 1, 在将传送请求源指定为内部外围模块或者自动请求时, 此位无效。</p> <p>00: 低电平检测</p> <p>01: 下降沿检测</p> <p>10: 高电平检测</p> <p>11: 上升沿检测</p>
5	TB	0	R/W	<p>传送总线模式位</p> <p>选择 DMA 传送的总线模式。</p> <p>0: 周期挪用模式</p> <p>1: 突发模式</p> <p>在设定内部外围模块请求时, 必须设定为周期挪用模式。</p>
4、3	TS[1:0]	00	R/W	<p>DMA 传送长度指定位</p> <p>请参照 TS[2] (bit20) 的说明。</p>
2	IE	0	R/W	<p>中断允许位</p> <p>指定在 DMA 最后传送时是否向 CPU 请求中断。如果 IE 位为 1, 就在 TE 位被置位并且在 DMA 最终传送的读周期结束时向 CPU 请求中断 (DMINT)。在要确认最终传送结束时, 必须在发生中断后虚读传送目标空间, 然后发行 SYNC0 指令。</p> <p>0: 禁止中断请求</p> <p>1: 允许中断请求</p>
1	TE	0	R(W)*	<p>传送结束标志位</p> <p>在 DMA 传送计数寄存器 (TCR) 的值为 0 (开始进行 DMA 最终传送时) 时, TE 位被置 1。在 TCR 不为 0 的情况下, 当因 NMI 中断或者 DMA 地址错误而传送结束时以及因清除 DE 位或者 DMA 操作寄存器 (DMAOR) 的 DME 位而传送结束时, TE 位不被置位。在清除 TE 位时, 必须在读 TE 位的 1 后写 0。</p> <p>在 TE 位被置位的状态下, 即使将 DE 位置 1 也不允许传送。</p> <p>0: 在 DMA 传送中或者发生 DMA 传送中断</p> <p>[清除条件] 在读 TE 位的 1 后写 0。</p> <p>1: TCR=0 (在 DMA 最终传送中或者 DMA 传送结束)</p>

位	位名	初始值	R/W	说明
0	DE	0	R/W	<p>DMA 允许位</p> <p>允许或者禁止 DMA 传送。在自动请求模式中，如果将 DE 位和 DMAOR 的 DME 位置 1，就开始传送。但是，TE 位、DMAOR 的 NMIF 位和 AE 位必须全部为 0。</p> <p>对于外部请求或者外围模块请求，在将 DE 位和 DME 位置 1 后从相应设备或者相应外围模块发生 DMA 传送请求时，开始传送。此时，也与自动请求模式相同，TE 位、NMIF 位和 AE 位必须全部为 0。如果将 DE 位清 0，就能中断传送。</p> <p>在设定内部外围模块请求时，如果要将 DE 位清 0（中断传送），就必须在清除该外围模块的 DMA 传送请求后进行。</p> <p>0：禁止 DMA 传送 1：允许 DMA 传送</p>

【注】 * 为了清除标志，只能写 0。

12.3.8 DMA 操作寄存器 0 (DMAOR0)

DMAOR 是 16 位可读写寄存器，指定 DMA 传送时的通道优先顺序，并且表示 DMA 的传送状态（状态）。

DMAOR0 是通道 0 ~ 5 的共用寄存器。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	CMS[1:0]	—	—	—	PR[1:0]	—	—	—	—	—	—	AE	NMIF	DME
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R	R	R	R/(W)*R/(W)*	R/(W)*R/(W)*	R/W

【注】 * 为了清除标志，只能写“0”。

位	位名	初始值	R/W	说明
15、14	—	全 0	R	<p>保留位</p> <p>读取值为 0，只能写 0。</p>
13、12	CMS[1:0]	00	R/W	<p>周期挪用模式选择位</p> <p>在周期挪用模式时，选择正常模式或者间歇模式。为了使间歇模式有效，DMAOR0 对应的全部通道（通道 0 ~ 5）的总线模式必须为周期挪用模式。</p> <p>00：正常模式 01：禁止设定 10：间歇模式 16 每 16 个外部总线时钟进行 1 次 DMA 传送 11：间歇模式 64 每 64 个外部总线时钟进行 1 次 DMA 传送</p>
11、10	—	全 0	R	<p>保留位</p> <p>读取值为 0，只能写 0。</p>

位	位名	初始值	R/W	说明
9、8	PR[1:0]	00	R/W	<p>优先模式位</p> <p>决定在多个通道同时发生传送请求时的执行通道的优先顺序。</p> <p>00: CH0 > CH1 > CH2 > CH3 > CH4 > CH5 (DMAOR0)</p> <p>01: CH0 > CH2 > CH3 > CH1 > CH4 > CH5 (DMAOR0)</p> <p>10: 禁止设定</p> <p>11: CH0 ~ CH5 循环调度模式 (DMAOR0)</p> <p>在指定循环调度模式时, 在 DMAOR0 对应的全部通道 (通道 0 ~ 5) 中不能同时存在周期挪用模式和突发模式。</p>
7 ~ 3	—	全 0	R	<p>保留位</p> <p>读取值为 0, 只能写 0。</p>
2	AE	00	R/(W)*	<p>地址错误标志位</p> <p>表示在 DMA 传送中发生地址错误。</p> <p>此位的置位条件如下:</p> <ul style="list-style-type: none"> 在 SAR 或者 DAR 的设定值和传送长度边界不相同 在传送源或者传送目标为无效空间 在传送源或者传送目标处于模块停止 <p>如果 DMAOR0 的 AE 位被置位, 即使将 DMAOR0 对应通道 (通道 0 ~ 5) 的 CHCR0 ~ 5 的 DE 位和 DMAOR0 的 DME 位置 1, 也不允许通道 0 ~ 5 的 DMA 传送。</p> <p>0: 无 DMAC 地址错误</p> <p>[清除条件] 在读 AE 位的 1 后写 0。</p> <p>1: 在 DMA 传送中发生地址错误</p>
1	NMIF	0	R/(W)*	<p>NMI 标志位</p> <p>表示发生 NMI 中断。如果 NMIF 位被置位, 即使将 CHCR 的 DE 位和 DMAOR 的 DME 位置 1, 也不允许 DMA 传送。</p> <p>在输入 NMI 中断后, 运行中的 DMA 传送至少进行 1 个传送单位的传送。即使在 DMAC 未运行时输入 NMI 中断, NMIF 位也被置 1。</p> <p>在输入 NMI 中断后, 停止 DMA 传送。必须在从 NMI 中断返回后重新设定全部通道, 然后开始传送。</p> <p>0: 无 NMI 中断</p> <p>[清除条件] 在读 NMIF 位的 1 后写 0。</p> <p>1: 发生 NMI 中断</p>
0	DME	0	R/W	<p>DMA 主控允许位</p> <p>允许或者禁止 DRMOR0 对应的全部通道 (通道 0 ~ 5) 的 DMA 传送。如果将 DME 位和 CHCR 的 DE 位置 1, 就允许 DMA 传送。但是, 传送通道的 CHCR 中的 TE 位和通道对应的 DMAOR 的 NMIF 位和 AE 位必须全部为 0。如果将 DME 位清 0, 就中断 DRMOR0 对应的全部通道 (通道 0 ~ 5) 的 DMA 传送。</p> <p>对于 DMAOR 对应的通道中的任意一个通道, 如果在设定为内部模块请求的情况下将 DME 位清 0 (中断 DMA 传送), 就必须在清除该外围模块的 DMA 传送请求后进行。</p> <p>0: 禁止通道 0 ~ 5 的 DMA 传送 (DRMOR0)</p> <p>1: 允许通道 0 ~ 5 的 DMA 传送 (DRMOR0)</p>

【注】 * 为了清除标志, 只能写 0。

12.3.9 DMA 扩展资源选择器 0 ~ 2 (DMARS0 ~ DMARS2)

• DMARS0

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C1MID[5:0]						C1RID[1:0]		C0MID[5:0]						C0RID[1:0]	
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

• DMARS1

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C3MID[5:0]						C3RID[1:0]		C2MID[5:0]						C2RID[1:0]	
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

• DMARS2

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	C5MID[5:0]						C5RID[1:0]		C4MID[5:0]						C4RID[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

DMARS 是 16 位可读写寄存器，DMARS0、DMARS1 和 DMARS2 分别设定通道 0 和通道 1、通道 2 和通道 3、通道 4 和通道 5 的外围模块的 DMA 传送请求源。通过此寄存器能设定 SCIF0、SCIF1、SCIF2、USB、FLCTL、SRC 的传送请求。

不能保证表 12.4 以外的 MID/RID 设定时的运行。DMARS 寄存器设定的传送请求只在将 CHCR0 ~ CHCR5 寄存器的资源选择位 (RS[3:0]) 设定为 B'1000 时才有效，在不为 B'1000 时，即使设定 DMARS 也不接受传送请求。另外，不能将一个外围模块的传送请求分配到 DMAC 的多个通道资源，否则不能保证运行。

• DMARS0 的设定

位	位名	初始值	R/W	说明
15 ~ 10	C1MID[5:0]	全 0	R/W	DMA 通道 1 传送请求源模块 ID[5:0] (MID) 参照表 12.4。
9、8	C1RID[1:0]	00	R/W	DMA 通道 1 传送请求源寄存器 ID[1:0] (RID) 参照表 12.4。
7 ~ 2	C0MID[5:0]	全 0	R/W	DMA 通道 0 传送请求源模块 ID[5:0] (MID) 参照表 12.4。
1、0	C0RID[1:0]	00	R/W	DMA 通道 0 传送请求源寄存器 ID[1:0] (RID) 参照表 12.4。

- DMARS1 的设定

位	位名	初始值	R/W	说明
15 ~ 10	C3MID[5:0]	全 0	R/W	DMA 通道 3 传送请求源模块 ID[5:0] (MID) 参照表 12.4
9、8	C3RID[1:0]	00	R/W	DMA 通道 3 传送请求源寄存器 ID[1:0] (RID) 参照表 12.4
7 ~ 2	C2MID[5:0]	全 0	R/W	DMA 通道 2 传送请求源模块 ID[5:0] (MID) 参照表 12.4
1、0	C2RID[1:0]	00	R/W	DMA 通道 2 传送请求源寄存器 ID[1:0] (RID) 参照表 12.4

- DMARS2 设定

位	位名	初始值	R/W	说明
15 ~ 10	C5MID[5:0]	全 0	R/W	DMA 通道 5 传送请求源模块 ID[5:0] (MID) 参照表 12.4。
9、8	C5RID[1:0]	00	R/W	DMA 通道 5 传送请求源寄存器 ID[1:0] (RID) 参照表 12.4。
7 ~ 2	C4MID[5:0]	全 0	R/W	DMA 通道 4 传送请求源模块 ID[5:0] (MID) 参照表 12.4。
1、0	C4RID[1:0]	00	R/W	DMA 通道 4 传送请求源寄存器 ID[1:0] (RID) 参照表 12.4。

表 12.4 传送请求源一览表

外围模块	1 个通道的设定值 (MID+RID)	MID	RID	功能
SCIF0	H'21	B'00 1000	B'01	发送
	H'22		B'10	接收
SCIF1	H'29	B'00 1010	B'01	发送
	H'2A		B'10	接收
SCIF2	H'41	B'01 0000	B'01	发送
	H'42		B'10	接收
USB	H'45	B'01 0001	B'01	发送
	H'46		B'10	接收
FLCTL	H'83	B'10 0000	B'11	数据的发送和接收
	H'87	B'10 0001	B'11	管理代码的发送和接收
SRC	H'C1	B'11 0000	B'01	从 SRCOD 传送数据
	H'C2		B'10	将数据传送到 SRCID

12.4 运行说明

DMAC 在发生 DMA 传送请求时按照决定的通道优先顺序开始传送，在满足传送结束条件时结束传送。传送请求有自动请求、外部请求和内部外围模块请求。总线模式能选择突发模式或者周期挪用模式。

12.4.1 DMA 传送请求

DMA 传送请求主要由数据的传送源或者传送目标产生，但是也能由既不是传送源也不是传送目标的外部设备和内部外围模块产生。

传送请求有自动请求、外部请求和内部外围模块请求。通过各 DMA 通道的 CHCR 的 RS[3:0] 位和 DMARS0 ~ DMARS2 寄存器进行传送请求的选择。

(1) 自动请求模式

在自动请求模式中，如同存储器之间的传送、不能产生传送请求的内部外围模块和存储器之间的传送，当没有外部传送请求信号时，DMAC 内部就自动产生传送请求信号。如果将各 DMA 通道的 CHCR 的 DE 位和通道 0 ~ 5 共用的 DMAOR0 的 DME 位置 1，就开始传送。但是，对于通道 0 ~ 5，DMAOR0 的 AE 位和 NMIF 位必须全部为 0。

(2) 外部请求模式

外部请求模式是通过 LSI 外部设备的传送请求信号 ($\overline{\text{DREQ0}}$ 和 $\overline{\text{DREQ1}}$) 开始传送的模式，只限于 DMA 通道 0 和通道 1 有效。外部请求模式的设定如表 12.5 所示。如果在允许 DMA 传送时 (DE=1、DME=1、TE=0、AE=0、NMIF=0) 输入 DREQ，就开始 DMA 传送。

表 12.5 通过 RS 位进行外部请求模式的设定

CHCR				地址模式	传送源	传送目标
RS3	RS2	RS1	RS0			
0	0	0	0	双地址模式	任意	任意

通过表 12.6 所示的 CHCR0、CHCR1 的 DREQ 电平 (DL) 位和 DREQ 边沿选择 (DS) 位，选择是以边沿检测 DREQ 还是以电平检测 DREQ。传送请求源不必为数据的传送源或者传送目标。

表 12.6 通过 DL、DS 位进行外部请求检测的选择

CHCR		外部请求检测方法	
DL	DS		
0	0	低电平检测	(初始值 DREQ)
	1	下降沿检测	
1	0	高电平检测	
	1	上升沿检测	

如果接受到 DREQ，DREQ 引脚就变为不能接受请求的状态。在对于接受的 DREQ 输出应答 DACK 后，DREQ 引脚再次变为能接受请求的状态。

在通过电平检测 DREQ 时，在输出 DACK 后，根据下一个 DREQ 的检测时序，可能在执行与请求相同次数的传送后 (溢出 0) 或者执行比请求多 1 次的传送后 (溢出 1) 出现中断的情况。通过 CHCR 的 DO 位选择溢出 0 或者溢出 1。

表 12.7 通过 DO 位进行外部请求检测的选择

CHCR	外部请求	
DO		
0	溢出 0	(初始值)
1	溢出 1	

(3) 内部外围模块请求模式

在内部外围模块请求模式中，根据内部外围模块的 DMA 传送请求信号进行传送。DMA 传送请求信号有 DMARS0 ~ DMARS2 设定的 SCIF0、SCIF1、SCIF2、USB、FLCTL、SRC 的发送数据空传送请求和接收数据满传送请求。在选择内部外围模块请求模式时，如果处于 DMA 传送允许状态 (DE=1、DME=1、TE=0、AE=0、NMIF=0)，就根据传送请求信号进行传送。

在将传送请求设定为 SCIF0 的发送数据空传送请求时，需要将传送目标设定为该 SCIF0 的传送数据寄存器。同样地，在将传送请求设定为 SCIF0 的接收数据满传送请求时，需要将传送源设定为该 SCIF0 的接收数据寄存器。对于 SCIF1、SCIF2、USB、FLCTL、SRC 也一样。

表 12.8 通过 RS[3:0] 位进行内部外围模块请求模式的选择

CHCR	DMARS		DMA 传送请求源	DMA 传送请求信号	传送源	传送目标	总线模式	
	RS[3:0]	MID						RID
1000	001000	01	SCIF0 发送部	TXI (发送 FIFO 数据空)	任意	SCFTDR0	周期挪用	
		10	SCIF0 接收部	RXI (接收 FIFO 数据满)	SCFRDR0	任意	周期挪用	
	001010	01	SCIF1 发送部	TXI (发送 FIFO 数据空)	任意	SCFTDR1	周期挪用	
		10	SCIF1 接收部	RXI (接收 FIFO 数据满)	SCFRDR1	任意	周期挪用	
	010000	01	SCIF2 发送部	TXI (发送 FIFO 数据空)	任意	SCFTDR2	周期挪用	
		10	SCIF2 接收部	RXI (接收 FIFO 数据满)	SCFTDR2	任意	周期挪用	
	010001	01	01	USB 发送部 *	发送数据空请求	任意	USB D1FIFO	周期挪用
			10	USB 接收部 *	接收数据满请求	USB D1FIFO	任意	周期挪用
		10	01	USB 发送部 *	发送数据空请求	任意	USB D0FIFO	周期挪用
			10	USB 接收部 *	接收数据满请求	USB D0FIFO	任意	周期挪用
	100000	11	01	FLCTL 数据的发送	发送 FIFO 数据空请求	任意	FLDTFIFO	周期挪用
			10	FLCTL 数据的接收	接收 FIFO 数据满请求	FLDTFIFO	任意	周期挪用
	100001	11	01	FLCTL 管理码的发送	发送 FIFO 数据空请求	任意	FLECFIF	周期挪用
			10	FLCTL 管理码的接收	接收 FIFO 数据满请求	FLECFIF	任意	周期挪用
	110000	01	01	SRC SRCOD	SRCOD FIFO 数据满请求	SRC	任意	周期挪用
			10	SRC SRCID	SRCID FIFO 数据空请求	任意	SRC	周期挪用

【注】 * USB 的发送部 / 接收部的选择取决于 USB 的设定。

12.4.2 通道优先顺序

在多个通道同时发生传送请求时，DMAC 按照决定的优先顺序进行传送。通道的优先顺序能选择固定或者循环调度模式，通过 DMAOR0（通道 0 ~ 5）的 PR[1:0] 位进行模式的选择。

(1) 固定模式

固定模式不按照通道的优先顺序而变。

固定模式有以下 2 种：

- 通道 0 ~ 5

CH0 > CH1 > CH2 > CH3 > CH4 > CH5

CH0 > CH2 > CH3 > CH1 > CH4 > CH5

通过 DMAOR0PR[1:0] 位选择。

(2) 循环调度模式

在循环调度模式中，每当 1 个通道结束 1 个传送单位（字节、字、长字、16 字节或者 32 字节单位）的传送时，改变优先顺序，使该通道的优先顺序变为最低，如图 12.2 所示。在复位后，循环调度模式的优先顺序为 CH0 > CH1 > CH2 > CH3 > CH4 > CH5。

在指定循环调度模式后，在 DMAOR0 对应的全部通道（通道 0 ~ 5）中不能同时存在周期挪用模式和突发模式。

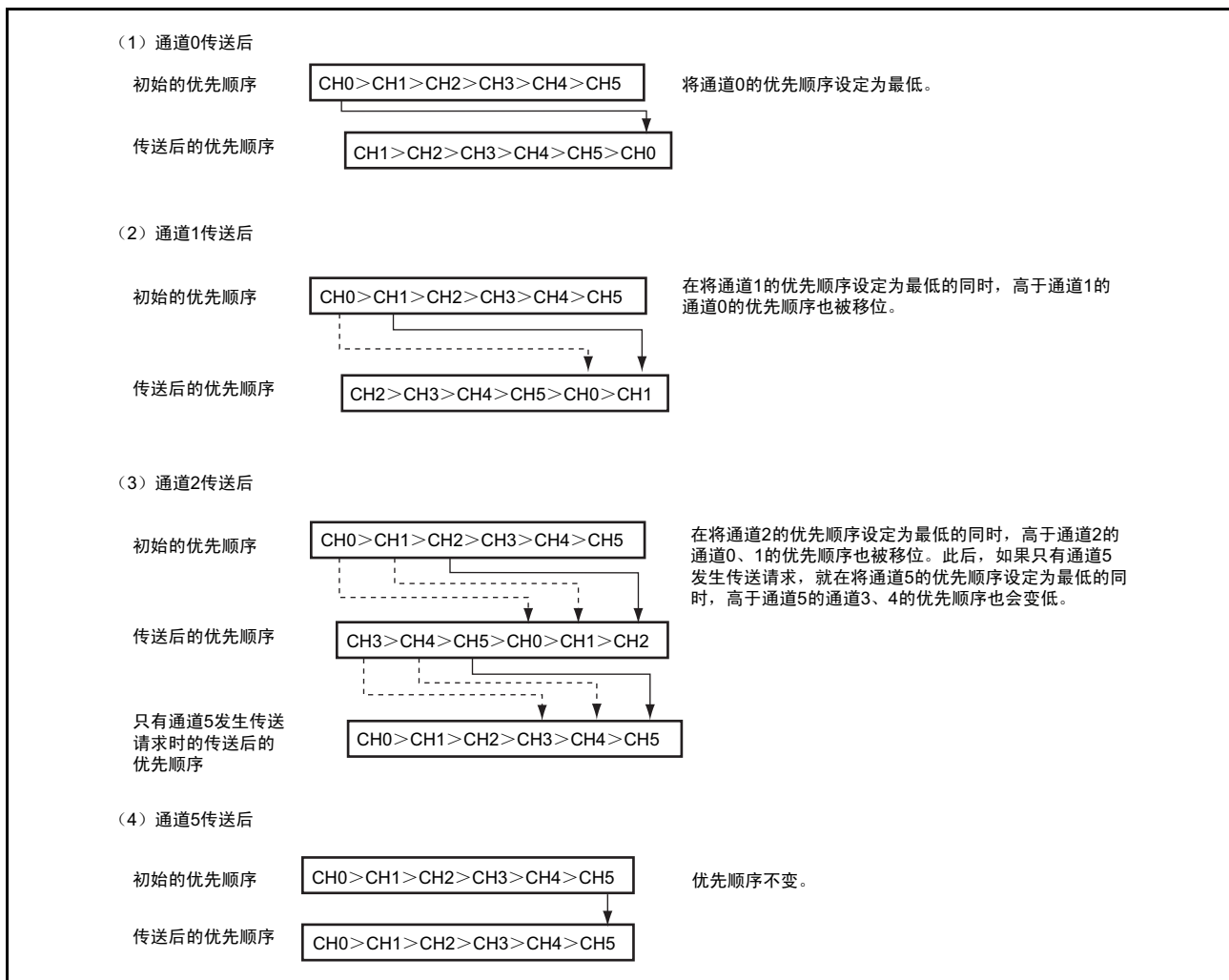


图 12.2 循环调度模式（例：通道 0 ~ 5）

在通道 0 和通道 3 同时发生传送请求并且在通道 0 的传送中通道 1 发生传送请求时，通道优先顺序的变化如图 12.3 所示。此时的 DMAC 运行如下：

1. 通道 0 和通道 3 同时发生传送请求。
2. 因为通道 0 的优先顺序高于通道 3，所以开始通道 0 的传送（通道 3 等待传送）。
3. 在通道 0 的传送中通道 1 发生传送请求（通道 1 和通道 3 等待传送）。
4. 在通道 0 传送结束时，将通道 0 的优先顺序变为最低。
5. 因为此时通道 1 的优先顺序高于通道 3，所以开始通道 1 的传送（通道 3 等待传送）。
6. 在通道 1 传送结束时，将通道 1 的优先顺序变为最低。
7. 开始通道 3 的传送。
8. 在通道 3 传送结束时，将通道 3 的优先顺序变为最低，同时将通道 2 的优先顺序变低。

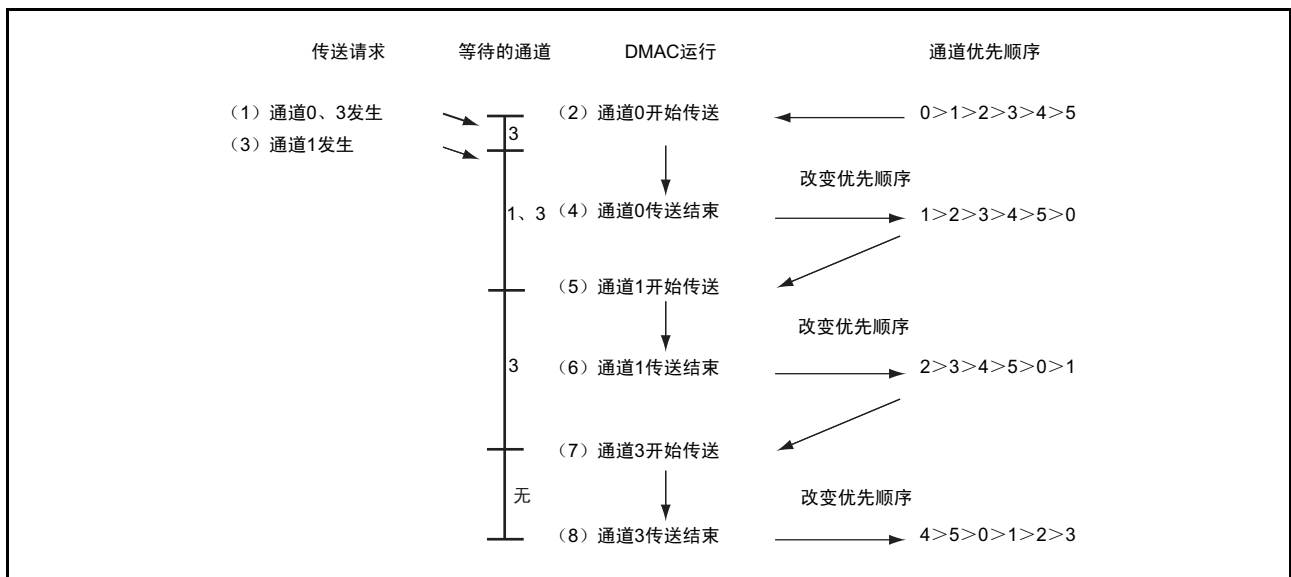


图 12.3 循环调度模式的通道优先级（例：通道 0 ~ 5）

12.4.3 DMA 传送的种类

DMAC 能支持的传送如表 12.8 所示。具体的传送运行时序因总线模式而不同，总线模式有周期挪用模式和突发模式。

(1) 双地址模式

双地址模式是用于通过地址存取传送源和传送目标的模式。传送源和传送目标可以是外部或者内部。在此模式中，DMAC 在读周期存取传送源，在写周期存取传送目标，在 2 个总线周期进行传送，此时传送数据被暂时保存到 DMAC。如图 12.4 所示的外部存储器之间的传送，在读周期将数据从一个外部存储器读到 DMAC，在连续的写周期将此数据写到另一个外部存储器。

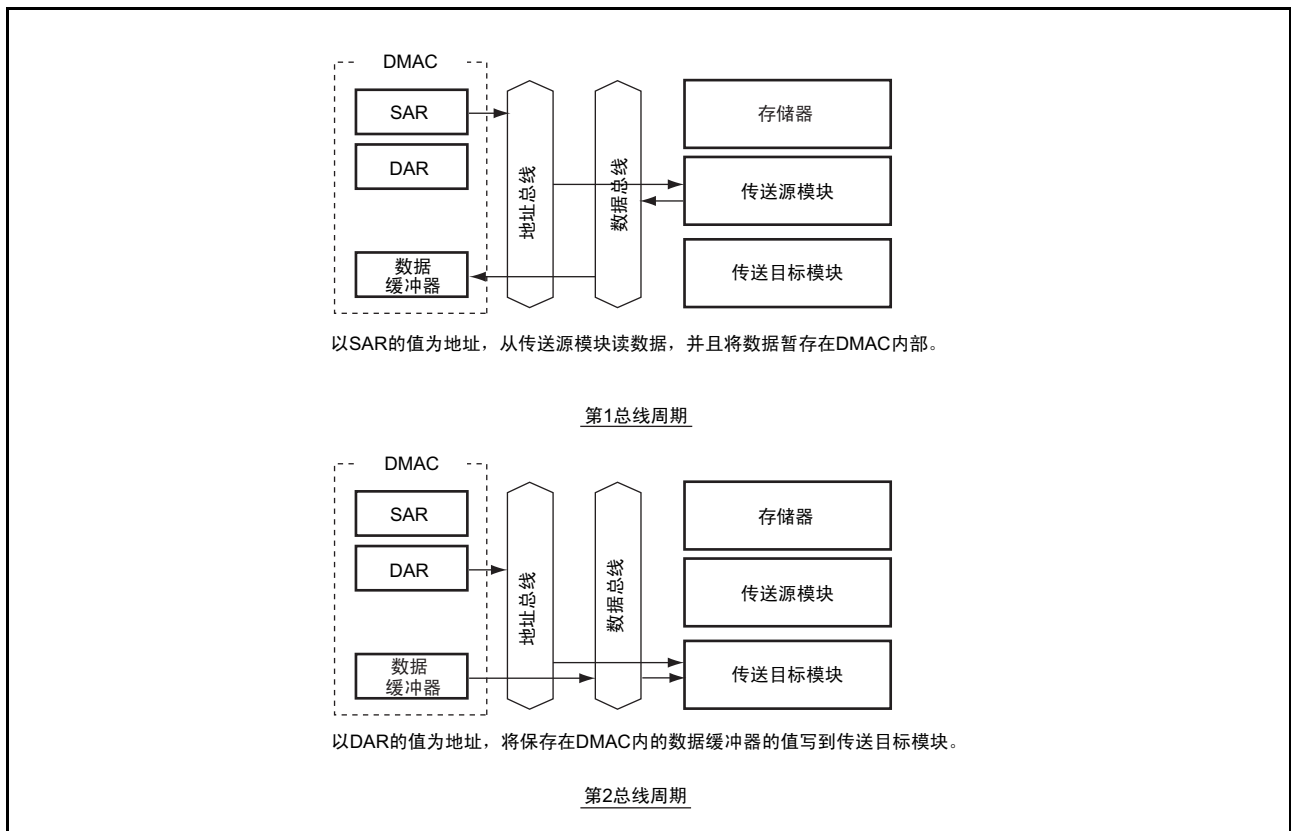


图 12.4 双地址模式的数据流程

传送请求可以是自动请求、外部请求或者内部外围模块请求。在双地址模式中，能通过通道控制寄存器 (CHCR) 设定是在读周期还是在写周期输出 DACK。

双地址模式的 DMA 传送时序例子如图 12.5 所示。

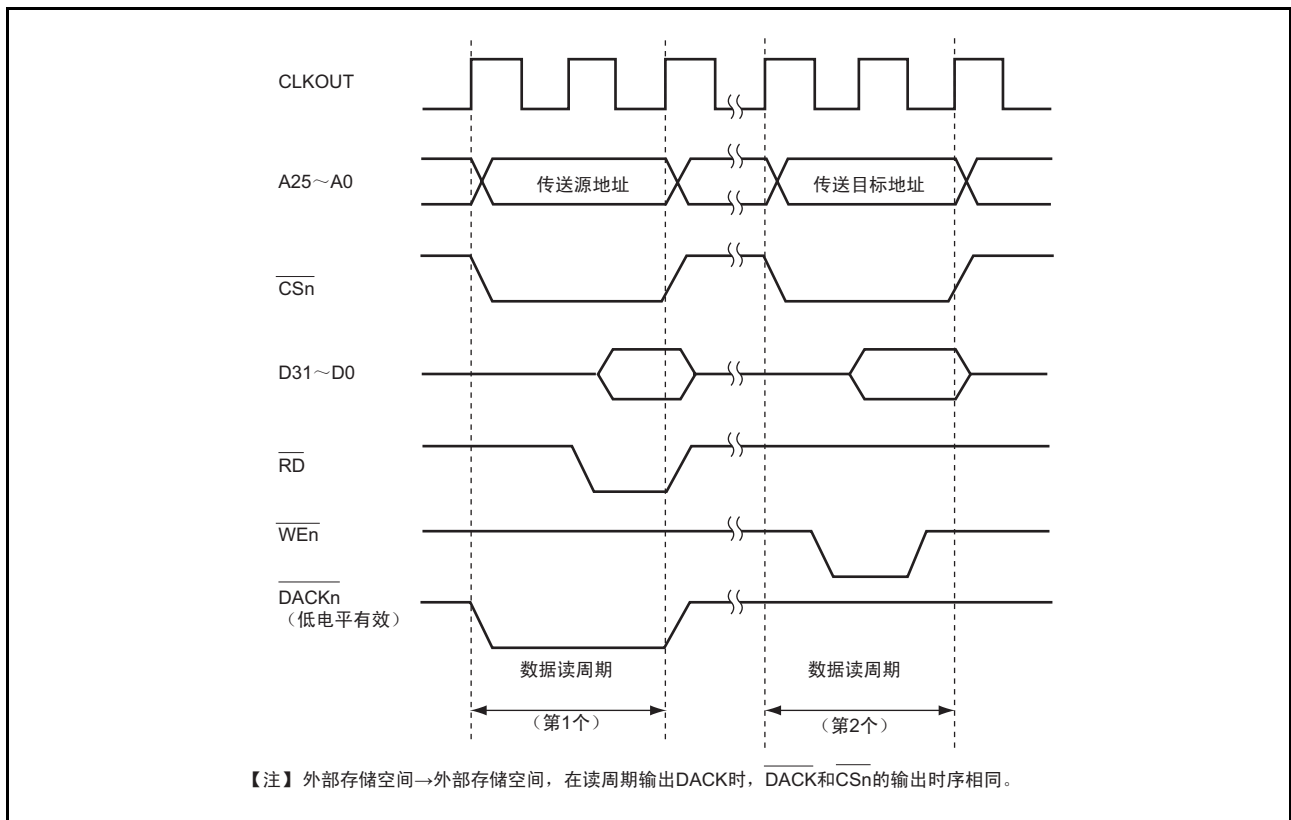


图 12.5 双地址模式的 DMA 传送时序例子 (传送源: 一般存储器, 传送目标: 一般存储器)

(2) 总线模式

总线模式有周期挪用模式和突发模式，通过通道控制寄存器 (CHCR) 的 TB 位进行模式的选择。

(a) 周期挪用模式

- 正常模式1 (LCKN=0、TB=0)

在周期挪用的正常模式1中，每当结束一个传送单位 (字节、字、长字、16字节或者32字节单位) 的传送时，DMAC就将SHwy总线的总线权释放给其他总线主控。此后，如果发生传送请求，就发行下一个传送请求，再次进行1个传送单位的传送，并且在结束该传送时再将SHwy总线的总线权释放给其他总线主控。在满足传送结束条件前重复此操作。

周期挪用正常模式1的DMA传送时序例子如图12.6所示。

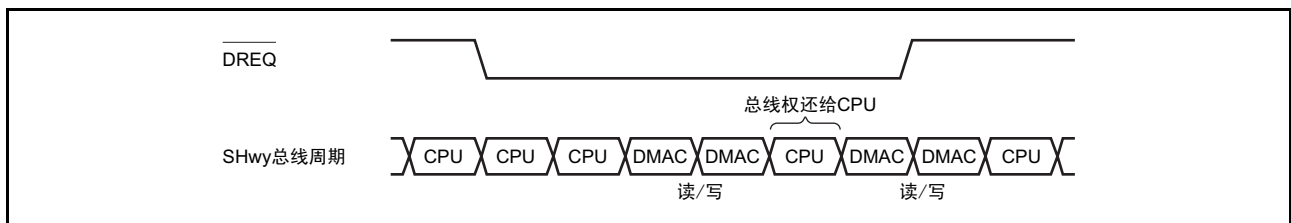


图 12.6 周期挪用正常模式1的 DMA 传送例子 (DREQ 低电平检测)

- 正常模式 2 (LCKN=1、TB=0)

在周期挪用的正常模式 2 中, DMAC 不确保 SHwy 总线的总线权, 而在每 1 个传送单位的读周期或者写周期获取 SHwy 总线的总线权进行传送。周期挪用正常模式 2 的 DMA 传送时序例子如图 12.7 所示。

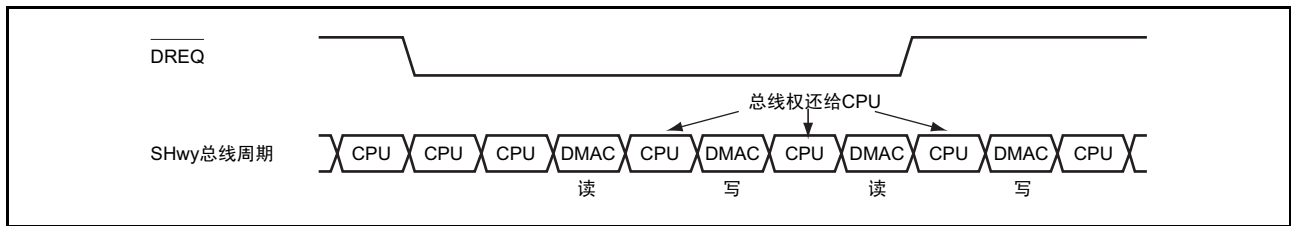


图 12.7 周期挪用正常模式 2 的 DMA 传送例子 (DREQ 低电平检测)

- 间歇模式 16 和间歇模式 64 (LCKN=0 或者 1、TB=0)

在周期挪用的间歇模式中, 每当结束一次传送单位 (字节、字、长字、16 字节或者 32 字节单位) 的传送时, DMAC 就将 SHwy 总线的总线权释放给其他的总线主控。此后, 如果发生传送请求, 就在通过 Bck 等待 16 个或者 64 个时钟后发行下一个传送请求, 再次进行 1 个传送单位的传送, 并且在结束此传送时再将 SHwy 总线的总线权释放给其他总线主控。在满足传送结束条件前重复此操作。因此与周期挪用正常模式 1 和正常模式 2 相比, 能降低 DMA 传送的总线占有率。

当 DMAC 再次发行下一个传送请求时, 因高速缓存未命中而进行入口更新等, 可能会导致 DMA 传送的等待。

在间歇模式中, DMAOR0 对应的全部通道 (通道 0~5) 的总线模式必须为周期挪用模式。周期挪用间歇模式的 DMA 传送时序例子如图 12.8 所示。

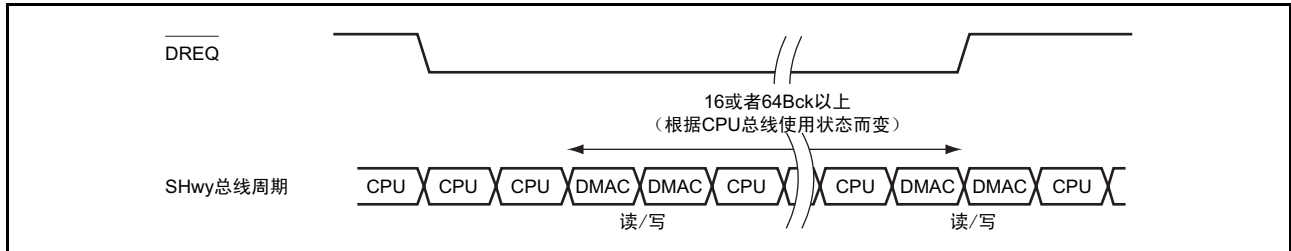


图 12.8 周期挪用间歇模式的 DMA 传送例子 (DREQ 低电平检测)

(b) 突发模式 (LCKN=0、TB=1)

在突发模式中, DMAC 一旦取得 SHwy 总线的总线权, 就在满足传送结束条件前不释放 SHwy 总线的总线权而继续进行传送。但是, 在外部请求模式中通过电平检测 DREQ 时, 如果 DREQ 不为有效电平, 即使未满足传送结束条件, 也在接受请求的 DMA 传送请求结束后将 SHwy 总线的总线权释放给其他总线主控。

在内部外围模块为传送请求源时, 不能使用突发模式。

突发模式的 DMA 传送时序如图 12.9 所示。

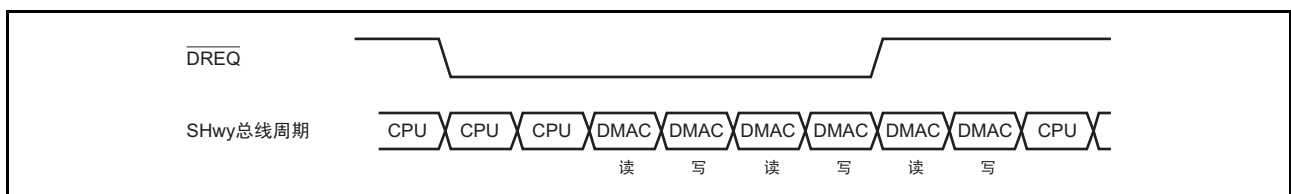


图 12.9 突发模式的 DMA 传送例子 (DREQ 低电平检测)

(3) DMA 传送区间

自动请求、外部请求和外围请求的 DMA 传送区间分别如表 12.9、表 12.10 和表 12.11 所示

表 12.9 自动请求的 DMA 传送区间

传送源	传送目标		
	MCU 空间	内部外围模块 *	IL 存储器
MCU 空间	○	○	○
内部外围模块 *	○	○	○
IL 存储器	○	○	○

【符号说明】○：能传送

【注】* 在传送源或者传送目标为外围模块时，是寄存器允许的存取长度。

表 12.10 外部请求的 DMA 传送区间 *2

传送源	传送目标		
	MCU 空间	内部外围模块 *1	IL 存储器
MCU 空间	○	○	○
内部外围模块 *1	○	○	○
IL 存储器	○	○	○

【符号说明】○：能传送，×：不能传送

【注】*1 在传送源或者传送目标为外围模块时，是寄存器允许的存取长度。

*2 外部请求只限于通道 0 和通道 1。

表 12.11 外围请求的 DMA 传送区间 *2、*3

传送源	传送目标		
	MCU 空间	内部外围模块 *1	IL 存储器
MCU 空间	×	○	×
内部外围模块 *1	○	○	○
IL 存储器	×	○	×

【符号说明】○：能传送，×：不能传送

【注】*1 在传送源或者传送目标为外围模块时，是寄存器允许的存取长度。

*2 传送源或者传送目标必须为外围请求的请求寄存器。

*3 只在周期挪用模式时才能设定。

(4) 总线模式和通道的优先顺序

在优先顺序固定模式 (CH0 > CH1) 中，即使通道 1 在突发模式传送中，如果优先顺序更高的通道 0 发生传送请求，也立刻开始通道 0 的传送。

此时，如果通道 0 也为突发模式，就在优先顺序高的通道 0 的传送全部结束后继续通道 1 的传送。

如果通道 0 为周期挪用模式，就在优先顺序高的通道 0 进行 1 个传送单位的传送后不释放总线权而连续进行通道 1 的传送。然后，按照通道 0→通道 1→通道 0→通道 1 的顺序进行交替传送。也就是说，对于总线状态，周期挪用模式传送结束后的 CPU 周期被突发模式传送替换（以下称为突发模式的优先执行）。

此例子如图 12.10 所示。在发生竞争的突发模式有多个通道时，优先执行其优先顺序最高的通道。

当多个通道进行 DMA 传送时，在发生竞争的全部突发传送结束前，不将总线权释放给总线主控。

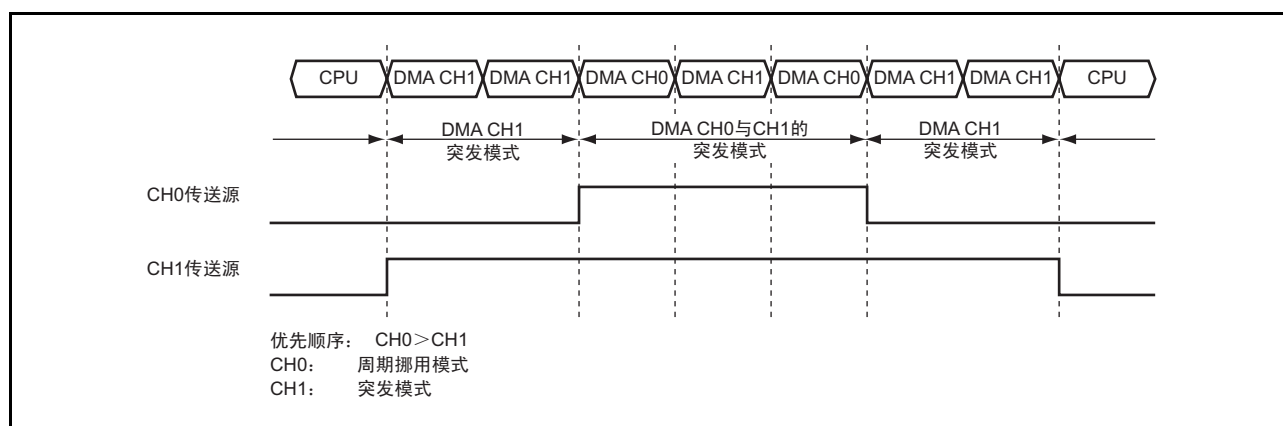


图 12.10 多个通道运行时的总线状态

在循环调度模式中，优先顺序按照如图 12.3 所示的规格而变。但是，总线模式不能同时存在周期挪用模式的通道和突发模式的通道。

12.4.4 传送流程

在给 DMA 源地址寄存器 (SAR)、DMA 目标地址寄存器 (DAR)、DMA 传送计数寄存器 (TCR)、DMA 通道控制寄存器 (CHCR)、DMA 操作寄存器 (DMAOR) 和 DMA 扩展源选择器 (DMARS) 设定目标传送条件后，DMAC 按照以下顺序传送数据：

1. 检查是否为传送允许状态 (DE=1、DME=1、TE=0、AE=0、NMIF=0)。
2. 如果在传送允许状态下发生传送请求，就传送 1 个传送单位的数据 (取决于 TS[2:0] 的设定)。如果在自动请求模式时将 DE 位和 DME 位置 1，就自动开始传送。
3. 每进行 1 次传送就将 TCR 的值减 1。具体的传送流程因地址模式和总线模式而不同。
4. 当指定次数的传送结束 (TCR 的值变为 0) 时，就正常结束传送。此时，如果 CHCR 的 IE 位已被置 1，就向 CPU 产生 DEI 中断。
5. 如果发生 DMAC 地址错误或者 NMI 中断，就中断传送。即使 CHCR 的 DE 位或者 DMAOR 的 DME 位被置 0 也中断传送。

DMA 传送的流程图如图 12.11 所示。

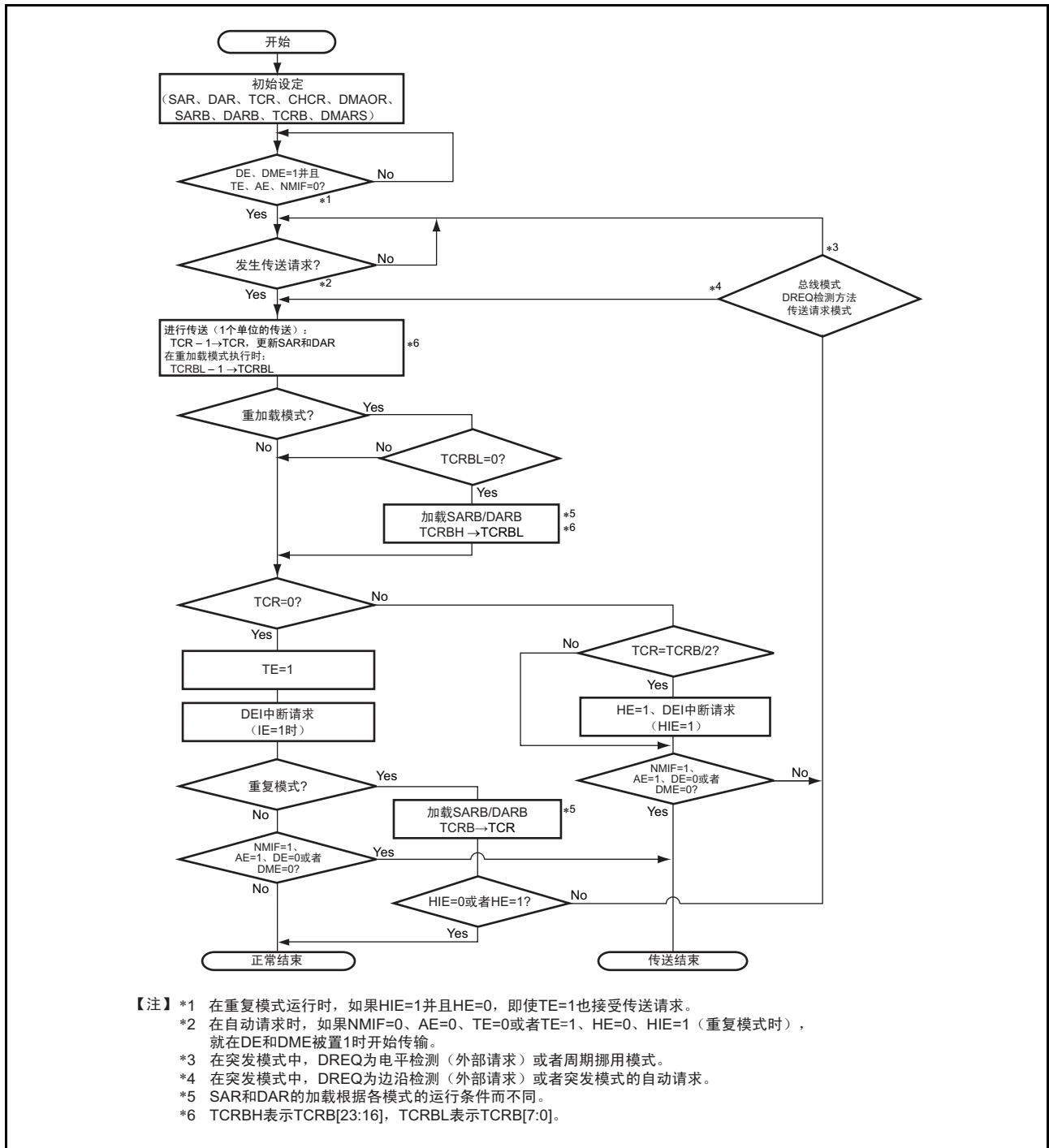


图 12.11 DMA 传送的流程图

12.4.5 重复模式的传送

如果使用 DMAC 的重复模式进行传送，就不重新设定 DMA 传送而能进行重复执行。通过将本功能和半结束功能一起使用，能虚拟地进行双重缓冲传送。

通过本功能能有效地进行以下处理。举例说明从外部存储器取数据并进行处理的情况。在此假定每取 40 个字的数据按顺序进行处理。

1. DMAC 的设定

给 SAR 设定外部存储器的地址。

给 DAR 设定内部存储器的数据保存区的地址。

给 TCR 设定 80 (H'50)。

给 CHCR 设定以下内容：

RPT (bit27、26、25) =B'010: 重复模式 (将目标侧用作重复区域)

HIE (bit18) =B'1: 产生 TCR/2 的中断

DM (bit15、14) =B'01: DAR 递增

SM (bit13、12) =B'00: SAR 固定

IE (bit2) =B'1: 允许中断

DE (bit0) =B'1: 允许 DMA 传送

此外, 根据使用条件设定 TB 和 TS 等。

根据使用条件设定 DMAOR 的 CMS 和 PR, 并且将 DME 置 1。

2. 通过 1 的设定开始 DMA 传送

3. 在 TCR 变为初始设定值的 1/2 时产生中断。

在中断处理中, 读 CHCR, 确认 HE (bit19) 是否被置 1, 并且从 DAR 设定的地址进行 40 个字的数据处理。

4. 在 TCR 变为 0 时产生中断。

在中断处理中, 读 CHCR, 确认 TE (bit1) 是否被置 1, 并且从 DAR 设定的地址加上 40 的地址进行 40 个字的数据处理。

此时, DMAC 将 DARB 的值复制到 DAR 并进行初始化, 同时 TCRB 的值也被复制到 TCR 并恢复为初始值 80。

5. 此后, 在 DME 或者 DE 被清 0 或者发生 NMI 中断前, 重复执行 2~3。也就是说, 通过使用此功能, 能按顺序将接受数据的保存缓冲器和信号处理的数据缓冲器进行相互交换的数据处理。

12.4.6 重加载模式的传送

如果使用 DMAC 的重加载模式进行传送, 就能通过 CHCR 的 RPT[2:0] 位的设定, 在每进行 TCRB[7:0] 所设次数的传送时, 分别给 SAR/DAR 和 TCRB[7:0] 重新设定 SARB/DARB 的值和 TCRB[23:16] 的值, 在 TCR 变为 0 前不需重新设定传送而重复执行。本功能在重复进行使用特定区域的数据传送时有效。此运行如图 12.12 所示。

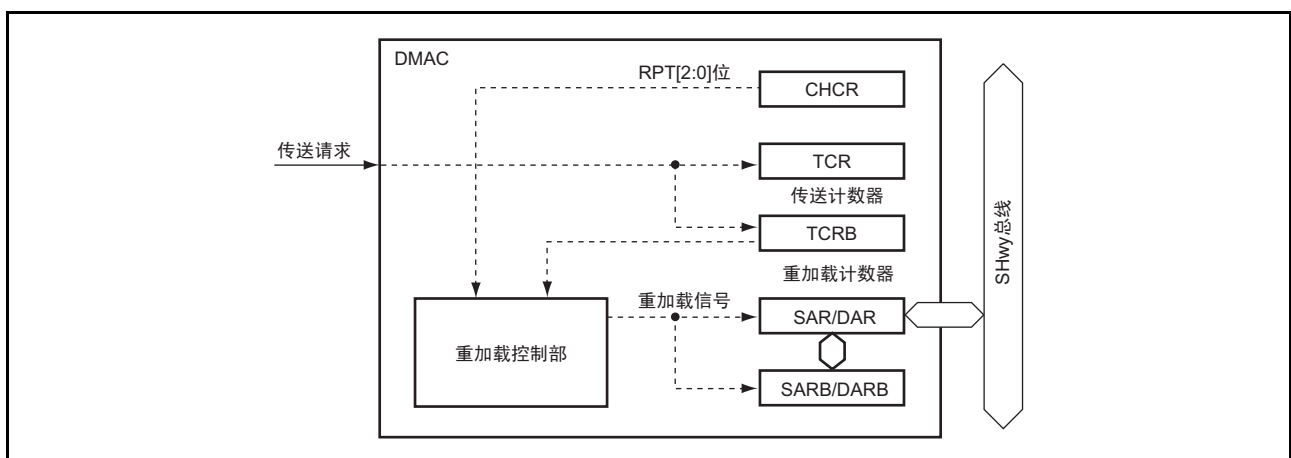


图 12.12 重加载功能图

在执行重加载模式时，TCRB 用作重加载计数器。必须参照“12.3.6 DMA 传送计数寄存器 B0 ~ 3 (TCRB0 ~ TCRB3)”进行 TCRB 的设置。

12.4.7 DREQ 引脚的采样时序

各总线模式的 DREQ 输入的采样时序如图 12.13、图 12.14、图 12.15 和图 12.16 所示。

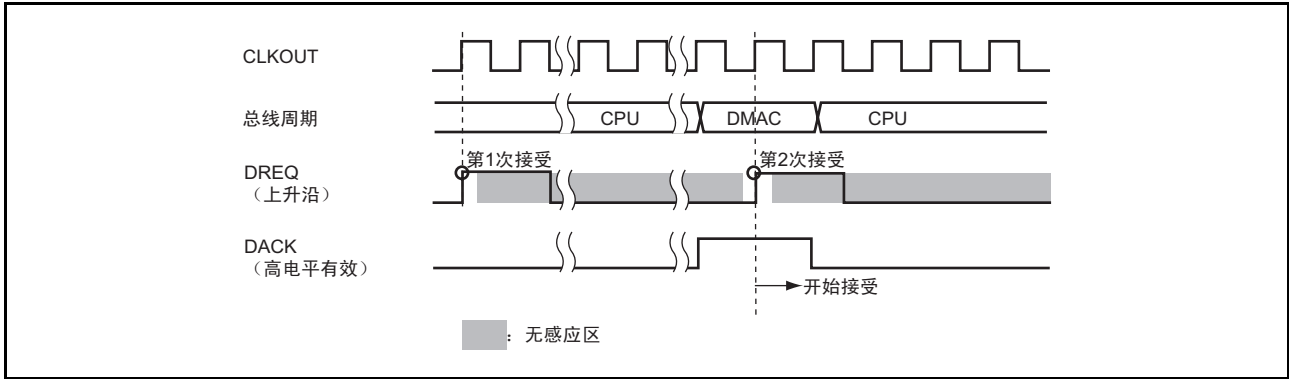


图 12.13 周期挪用模式、边沿检测时的 DREQ 输入检测时序例子

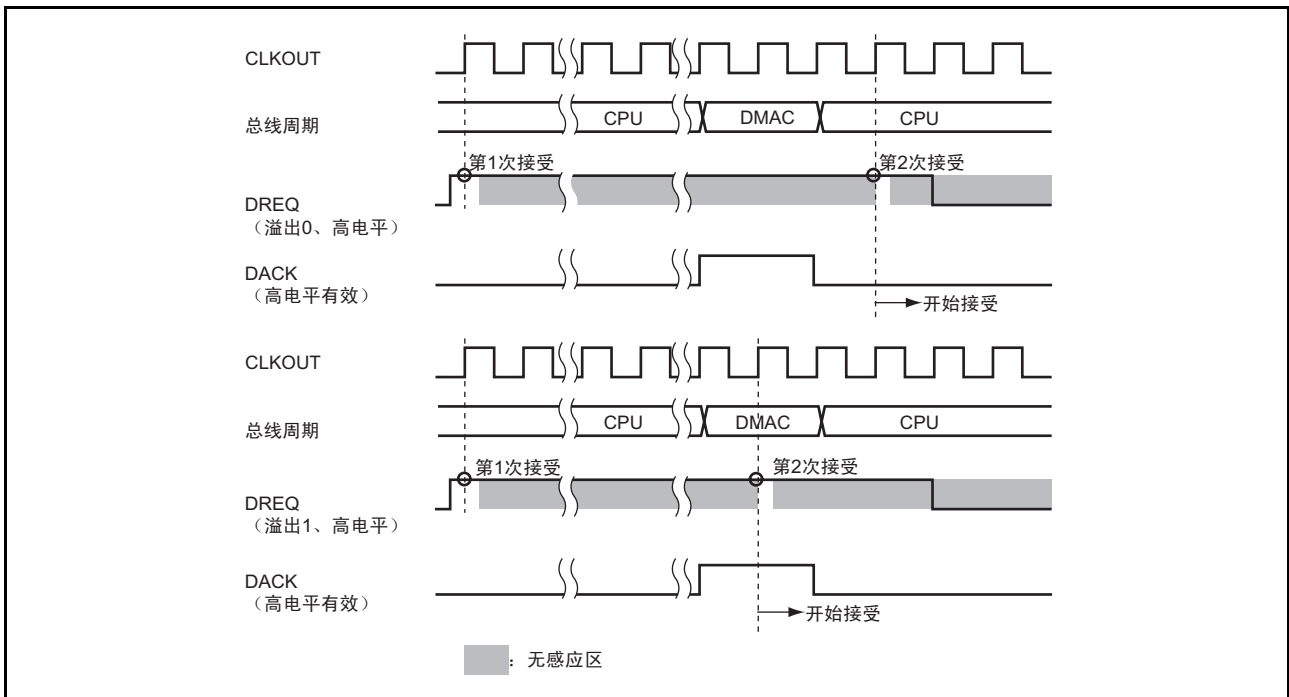


图 12.14 周期挪用模式、电平检测时的 DREQ 输入检测时序例子

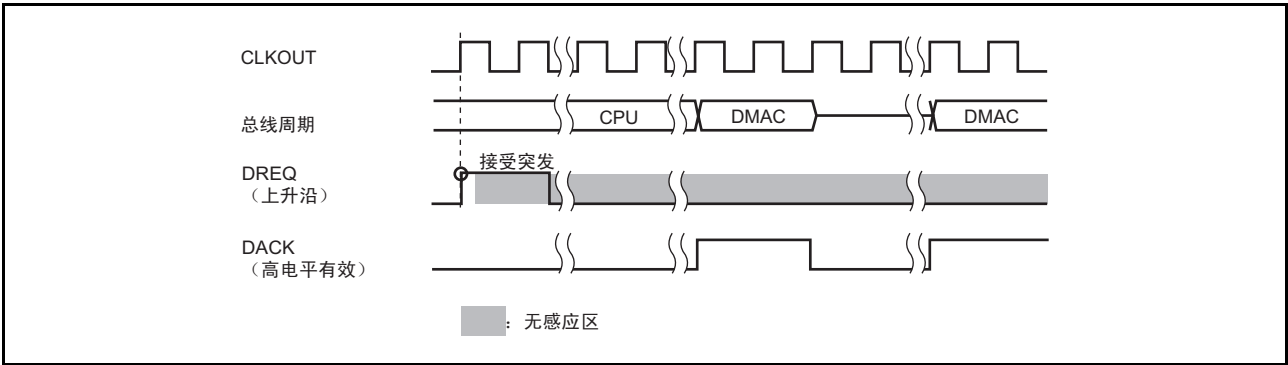


图 12.15 突发模式、边沿检测时的 DREQ 输入检测时序例子

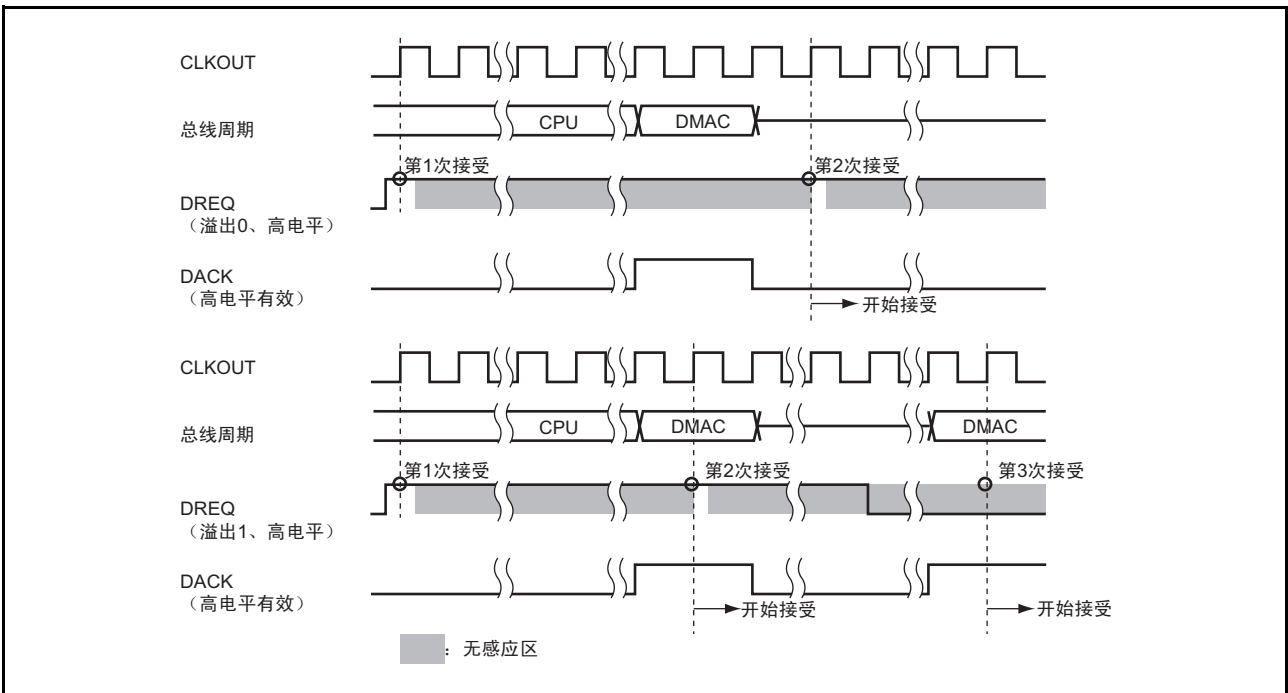


图 12.16 突发模式、电平检测时的 DREQ 输入检测时序例子

DTEND 的输出时序如图 12.17 所示。

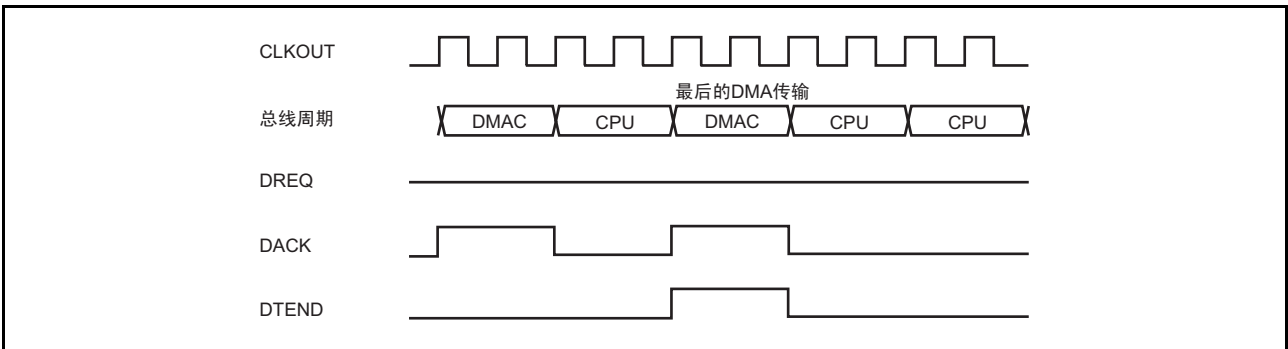


图 12.17 DMA 传送结束的信号时序 (周期挪用、电平检测)

必须注意：在以长字为单位存取 8 位或者 16 位的外部设备时，或者在以字为单位存取 8 位的外部设备时，为了调整数据， $\overline{\text{DACK}}$ 输出和 $\overline{\text{DTEND}}$ 输出被分割。此例子如图 12.18 所示。

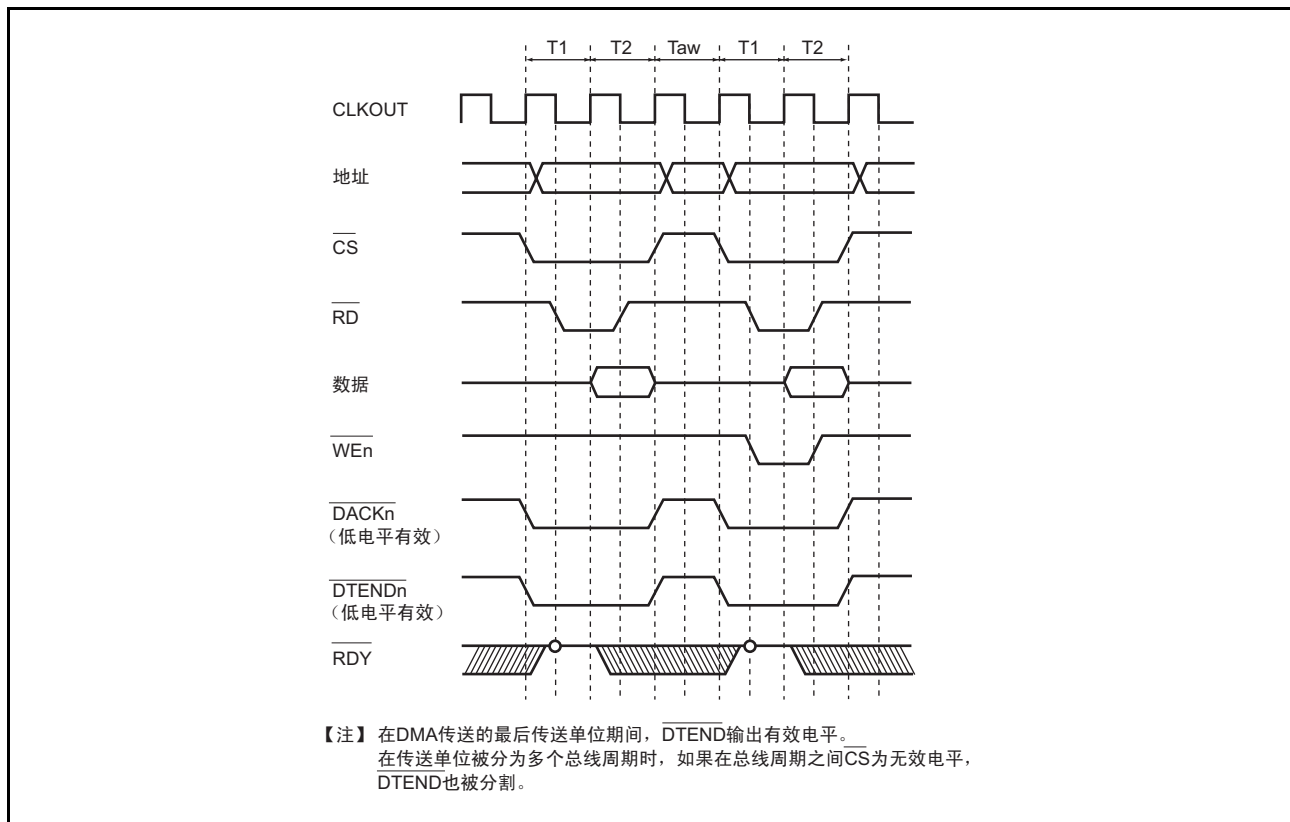


图 12.18 一般存储器的存取例子
(无等待、1 个空闲周期、以长字存取 16 位的设备)

12.5 使用时的注意事项

在使用本 DMAC 时，必须注意以下事项。

12.5.1 模块停止

在本 DMAC 运行中，不能通过 CPG 寄存器的设定使模块停止，否则不能保证运行中的传送内容。

12.5.2 地址错误

在发生 DMA 地址错误时，必须在进行下述处理后重新设定 DMAOR 对应的全部通道，然后开始传送。

1. 虚读下述模块。
 - MCU: BCR (总线状态控制寄存器)
 - INTC: INTC2B3 (各个模块的中断源寄存器3)
 - IL存储器: IL存储器
 - EtherC: EESR (EtherC/E-DMAC状态寄存器)
 - USB: USBINTSTS0 (USB中断状态0寄存器)
2. 发行 SYNC0 指令。
3. 重新设定发生 DMA 地址错误的 DMAOR 所对应的全部通道。
 - 在 DMAOR0 的 AE 位已被置 1 时，必须重新设定通道 0~5。

12.5.3 突发模式传送时的注意事项

在突发模式传送中，不能在该通道的传送结束前进行以下操作：

- 不能转移到睡眠模式。
- 不能通过 CPG 寄存器的设定使模块停止。

12.5.4 DACK 的分割输出

如以长字为单位存取 8 位或者 16 位的外部设备，或者以字为单位存取 8 位的外部设备，当 DMA 传送单位被分割为多个总线周期时，并且在总线周期之间 CS 无效时，为了调整数据，和 CS 相同地 DACK 输出也被分割。

12.5.5 禁止对 DMAC 的 DMA 传送

不能给传送源或者传送目标设定 DMAC 寄存器进行 DMA 传送。

12.5.6 NMI 中断

当发生 NMI 中断时，就停止 DMA 传送。必须在从 NMI 中断返回后重新设定全部通道，然后开始传送。

第 13 章 中断控制器 (INTC)

中断控制器 (INTC) 判断中断源的优先顺序, 控制向 CPU (SH-4A) 的中断请求。INTC 有设定各中断优先顺序的寄存器, 用户根据此寄存器设定的优先顺序进行中断请求的处理。

13.1 特点

(1) SH-4 兼容规格

INTC 有以下特点:

- 能将外部中断的中断优先顺序设定为 15 级。
通过中断优先顺序设定寄存器, 能按引脚将外部中断的优先顺序最多设定为 15 级。
- NMI 噪声消除
有表示 NMI 引脚状态的 NMI 输入电平位。在中断异常处理程序中, 能通过读此位确认引脚状态, 并能用作噪声消除器。
- SR.BL 位置 1 时的 NMI 请求的屏蔽
当 SR.BL 位置 1 时, 能选择是否屏蔽 NMI 请求。

(2) SH-4A 的扩展功能

- 能将 SR.IMASK 位自动更新为接受的中断级。
- 能将内部模块的中断优先顺序设定为 30 级。
通过 13 个中断优先顺序设定寄存器, 能按请求将内部模块的中断优先顺序最多设定为 30 级。
- 用户模式的中断禁止功能
在用户模式的运行过程中, 通过给用户中断屏蔽级设定寄存器 (USERIMASK) 设定中断屏蔽级, 能禁止设定值以下的中断级的中断。

INTC 的框图如图 13.1 所示。

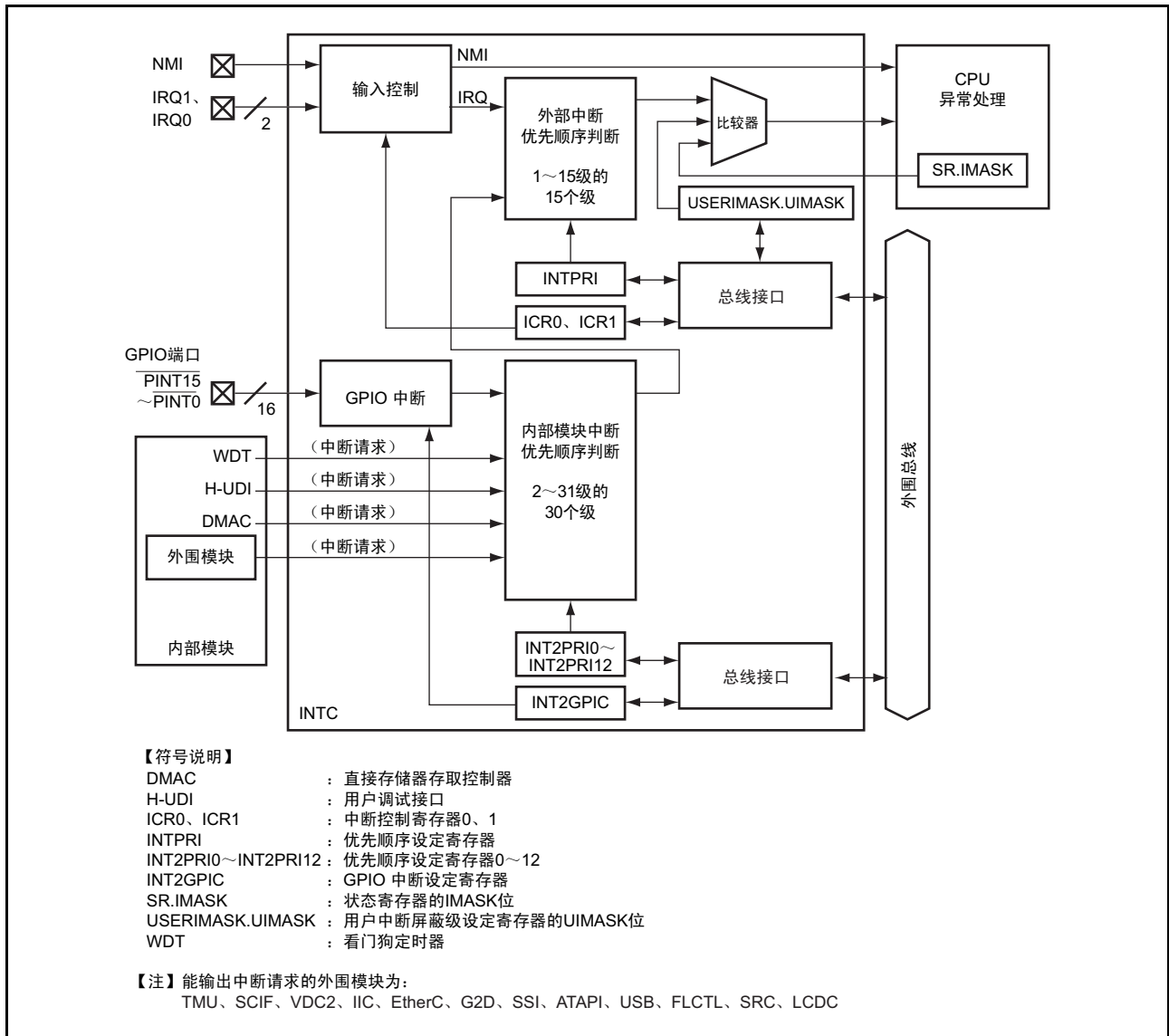


图 13.1 INTC 的框图

13.1.1 中断方式

在发生中断时，基本的异常处理流程如下所示：

在处理异常时，将程序计数器 (PC)、状态寄存器 (SR) 和通用寄存器 15 (R15) 的内容分别保存到保存程序计数器 (SPC)、保存状态寄存器 (SSR) 和保存通用寄存器 15 (SGR)，根据向量地址开始执行相应的异常处理程序。异常处理程序是指用户根据各异常内容编制的程序。为了结束异常处理程序并且返回到原来的程序，执行返回指令 (RTE)。通过此指令恢复 PC 和 SR 的内容，并且能返回到发生异常等时的正常处理程序。另外，无法通过 RTE 指令将 SGR 的内容回写到 R15。

1. 将 PC、SR 和 R15 的内容分别保存到 SPC、SSR 和 SGR。
2. 将 SR 的块位 (BL) 置 1。
3. 将 SR 的模式位 (MD) 置 1。
4. 将 SR 寄存器组位 (RB) 置 1。
5. 复位时，SR 的 FPU 禁止位 (FD) 置 0。
6. 将异常码写到中断事件寄存器 (INTEVT) 的 bit13~0。
转移到中断异常处理程序的起始地址向量基址寄存器 (VBR) +H'600。
7. 转移到规定的异常处理向量地址，开始执行异常处理程序。

13.1.2 INTC 的中断

中断的种类例子如表 13.1 所示，INTC 支持外部中断源和内部模块中断源。

外部中断源是指由外部引脚的输入而产生 NMI 和 IRQ 的中断。

IRQ 中断的检测方法能选择电平、上升沿和下降沿。

表 13.1 中断种类

源	源数 (最大)	优先顺序	INTEVT (异常码)	备注	
外部中断	NMI	1	—	H'1C0	
	IRQ	2	INTPRI 寄存器的 设定值	H'240	IRQ0
H'280				IRQ1	
内部模块中断	WDT	1	INT2PRI0 ~	H'560	ITI
	TMU0	1	INT2PRI12 寄存 器的设定值	H'580	TUNI0
	TMU1	1		H'5A0	TUNI1
	TMU2	2		H'5C0	TUNI2
				H'5E0	TICPI2
	H-UDI	1		H'600	H-UDI
	LCDC	1		H'620	LCDCI
	DMAC	7 (5/7)		H'640	DMINT0
				H'660	DMINT1
				H'680	DMINT2
				H'6A0	DMINT3
				H'6C0	DMAE (ch0 ~ ch5 共用)
	SCIF0	4		H'700	ERI0
				H'720	RXI0
				H'740	BRI0
				H'760	TXI0
DMAC	7 (2/7)	H'780		DMINT4	
		H'7A0	DMINT5		
VDC2	1	H'860	VDCI		
IIC	1	H'8A0	IICI		

源	源数 (最大)	优先顺序	INTEVT (异常码)	备注	
内部模块中断	EtherC	1	INT2PRI0 ~ INT2PRI12 寄存 器的设定值	H'920	EINT
	G2D	1		H'980	G2DI
	SSI_A	4		H'A00	SSIDMA0
				H'A20	SSICH0
				H'A40	SSICH1
				H'A60	SSICH2
	SSI_B	4		H'AA0	SSIDMA1
				H'AC0	SSICH3
				H'AE0	SSICH4
				H'B00	SSICH5
	SCIF1	4		H'B80	ERI1
				H'BA0	RXI1
				H'BC0	BRI1
				H'BE0	TXI1
	ATAPI	1		H'C00	ATAI
	USB	1		H'C60	USBI
	FLCTL	4		H'D00	FLSTE
				H'D20	FLTEND
				H'D40	FLTRQ0
				H'D60	FLTRQ1
	TMU3	1		H'E00	TUNI3
	TMU4	1		H'E20	TUNI4
	TMU5	1		H'E40	TUNI5
	SRC	3		H'E80	SRC OVF
				H'EA0	SRC IDEI
				H'EC0	SRC ODFI
	SCIF2	4		H'F00	ERI2
				H'F20	RXI2
				H'F40	BRI2
				H'F60	TXI2
	GPIO	4		H'F80	CH0
				H'FA0	CH1
H'FC0			CH2		
H'FE0			CH3		

【注】 *1 内部模块中断源使用的略称:

- ITI : WDT 的间隔定时器中断
- TUNI0 ~ TUNI5 : TMU 通道 0 ~ 5 的下溢中断
- TICPI2 : TMU 通道 2 的输入捕捉中断
- DMINT0 ~ DMINT5 : DMAC 通道 0 ~ 5 的传送结束中断
- DMAE : DMAC 的地址错误中断 (通道 0 ~ 5 共用)
- ERI0、ERI1、ERI2 : SCIF 通道 0、1、2 的接收错误中断
- RXI0、RXI1、RXI2 : SCIF 通道 0、1、2 的接收数据满中断
- BRI0、BRI1、BRI2 : SCIF 通道 0、1、2 的中止中断请求
- TXI0、TXI1、TXI2 : SCIF 通道 0、1、2 的发送数据空中断
- CH0 : PINT0 ~ PINT3 引脚的 GPIO 通道 0 中断
- CH1 : PINT4 ~ PINT7 引脚的 GPIO 通道 1 中断
- CH2 : PINT8 ~ PINT11 引脚的 GPIO 通道 2 中断
- CH3 : PINT12 ~ PINT15 引脚的 GPIO 通道 3 中断

13.2 输入 / 输出引脚

引脚结构如表 13.2 所示。

表 13.2 INTC 的引脚结构

引脚名	功能	输入 / 输出	说明
NMI	非屏蔽中断输入引脚	输入	输入不能屏蔽的中断请求信号。
IRQ1、IRQ0	外部中断输入引脚	输入	输入 IRQ1 和 IRQ0 中断请求信号。
IRQOUT	中断请求输出引脚	输出	是将发生中断请求通知外部设备的信号输出。
PINT15 ~ PINT0	端口中断输入引脚	输入	输入端口中断请求信号。

13.3 寄存器说明

INTC 的寄存器结构如表 13.3 所示，各处理模式的寄存器状态如表 13.4 所示。

表 13.3 INTC 的寄存器结构

名称	略称	R/W	P4 区域地址	区域 7 地址	存取长度
中断控制寄存器 0	ICR0	R/W	H'FFD0 0000	H'1FD0 0000	32
中断控制寄存器 1	ICR1	R/W	H'FFD0 001C	H'1FD0 001C	32
中断优先顺序设定寄存器	INTPRI	R/W	H'FFD0 0010	H'1FD0 0010	32
中断源寄存器	INTREQ	R/W	H'FFD0 0024	H'1FD0 0024	32
中断屏蔽寄存器	INTMSK	R/W	H'FFD0 0044	H'1FD0 0044	32
中断屏蔽清除寄存器	INTMSKCLR	R/W	H'FFD0 0064	H'1FD0 0064	32
NMI 标志控制寄存器	NMIFCR	R/W	H'FFD0 00C0	H'1FD0 00C0	32
用户中断屏蔽级设定寄存器	USERIMASK	R/W	H'FFD3 0000	H'1FD3 0000	32
中断优先顺序设定寄存器 0	INT2PRI0	R/W	H'FFD4 0000	H'1FD4 0000	32
中断优先顺序设定寄存器 1	INT2PRI1	R/W	H'FFD4 0004	H'1FD4 0004	32
中断优先顺序设定寄存器 2	INT2PRI2	R/W	H'FFD4 0008	H'1FD4 0008	32
中断优先顺序设定寄存器 3	INT2PRI3	R/W	H'FFD4 000C	H'1FD4 000C	32
中断优先顺序设定寄存器 4	INT2PRI4	R/W	H'FFD4 0010	H'1FD4 0010	32
中断优先顺序设定寄存器 5	INT2PRI5	R/W	H'FFD4 0014	H'1FD4 0014	32
中断优先顺序设定寄存器 6	INT2PRI6	R/W	H'FFD4 0018	H'1FD4 0018	32
中断优先顺序设定寄存器 7	INT2PRI7	R/W	H'FFD4 001C	H'1FD4 001C	32
中断优先顺序设定寄存器 8	INT2PRI8	R/W	H'FFD4 00A0	H'1FD4 00A0	32
中断优先顺序设定寄存器 9	INT2PRI9	R/W	H'FFD4 00A4	H'1FD4 00A4	32
中断优先顺序设定寄存器 10	INT2PRI10	R/W	H'FFD4 00A8	H'1FD4 00A8	32
中断优先顺序设定寄存器 11	INT2PRI11	R/W	H'FFD4 00AC	H'1FD4 00AC	32
中断优先顺序设定寄存器 12	INT2PRI12	R/W	H'FFD4 00B0	H'1FD4 00B0	32
中断源寄存器 0 (不影响屏蔽状态)	INT2A0	R	H'FFD4 0030	H'1FD4 0030	32
中断源寄存器 01 (不影响屏蔽状态)	INT2A01	R	H'FFD4 00C0	H'1FD4 00C0	32
中断源寄存器 1 (影响屏蔽状态)	INT2A1	R	H'FFD4 0034	H'1FD4 0034	32
中断源寄存器 11 (影响屏蔽状态)	INT2A11	R	H'FFD4 00C4	H'1FD4 00C4	32

名称	略称	R/W	P4 区域地址	区域 7 地址	存取长度
中断屏蔽寄存器	INT2MSKR	R/W	H'FFD4 0038	H'1FD4 0038	32
中断屏蔽寄存器 1	INT2MSKR1	R/W	H'FFD4 00D0	H'1FD4 00D0	32
中断屏蔽清除寄存器	INT2MSKCR	W	H'FFD4 003C	H'1FD4 003C	32
中断屏蔽清除寄存器 1	INT2MSKCR1	W	H'FFD4 00D4	H'1FD4 00D4	32
各模块中断源寄存器 0	INT2B0	R	H'FFD40 0040	H'1FD4 0040	32
各模块中断源寄存器 2	INT2B2	R	H'FFD4 0048	H'1FD4 0048	32
各模块中断源寄存器 3	INT2B3	R	H'FFD4 004C	H'1FD4 004C	32
各模块中断源寄存器 4	INT2B4	R	H'FFD4 0050	H'1FD4 0050	32
各模块中断源寄存器 5	INT2B5	R	H'FFD4 0054	H'1FD4 0054	32
各模块中断源寄存器 6	INT2B6	R	H'FFD4 0058	H'1FD4 0058	32
各模块中断源寄存器 7	INT2B7	R	H'FFD4 005C	H'1FD4 005C	32
GPIO 中断设定寄存器	INT2GPIC	R/W	H'FFD4 0090	H'1FD4 0090	32

表 13.4 各处理模式的寄存器状态

名称	略称	上电复位 (通过 PRESET 引脚 / WDT/H-UDI)	睡眠 (通过 SLEEP 指令)
中断控制寄存器 0	ICR0	H'x000 0000	保持
中断控制寄存器 1	ICR1	H'0000 0000	保持
中断优先顺序设定寄存器	INTPRI	H'0000 0000	保持
中断源寄存器	INTREQ	H'0000 0000	保持
中断屏蔽寄存器	INTMSK	H'FF00 0000	保持
中断屏蔽清除寄存器	INTMSKCLR	H'0000 0000	保持
NMI 标志控制寄存器	NMIFCR	H'x000 0000	保持
用户中断屏蔽级设定寄存器	USERIMASK	H'0000 0000	保持
中断优先顺序设定寄存器 0	INT2PRI0	H'0000 0000	保持
中断优先顺序设定寄存器 1	INT2PRI1	H'0000 0000	保持
中断优先顺序设定寄存器 2	INT2PRI2	H'0000 0000	保持
中断优先顺序设定寄存器 3	INT2PRI3	H'0000 0000	保持
中断优先顺序设定寄存器 4	INT2PRI4	H'0000 0000	保持
中断优先顺序设定寄存器 5	INT2PRI5	H'0000 0000	保持
中断优先顺序设定寄存器 6	INT2PRI6	H'0000 0000	保持
中断优先顺序设定寄存器 7	INT2PRI7	H'0000 0000	保持
中断优先顺序设定寄存器 8	INT2PRI8	H'0000 0000	保持
中断优先顺序设定寄存器 9	INT2PRI9	H'0000 0000	保持
中断优先顺序设定寄存器 10	INT2PRI10	H'0000 0000	保持
中断优先顺序设定寄存器 11	INT2PRI11	H'0000 0000	保持
中断优先顺序设定寄存器 12	INT2PRI12	H'0000 0000	保持
中断源寄存器 0 (不影响屏蔽状态)	INT2A0	H'xxxx xxxx	保持
中断源寄存器 01 (不影响屏蔽状态)	INT2A01	H'xxxx xxxx	保持
中断源寄存器 1 (影响屏蔽状态)	INT2A1	H'0000 0000	保持
中断源寄存器 11 (影响屏蔽状态)	INT2A11	H'0000 0000	保持

名称	略称	上电复位 (通过 PRESET 引脚 / WDT/H-UDI)	睡眠 (通过 SLEEP 指令)
中断屏蔽寄存器	INT2MSKR	H'FFFF FFFF	保持
中断屏蔽寄存器 1	INT2MSKR1	H'FFFF FFFF	保持
中断屏蔽清除寄存器	INT2MSKCR	H'0000 0000	保持
中断屏蔽清除寄存器 1	INT2MSKCR1	H'0000 0000	保持
各模块中断源寄存器 0	INT2B0	H'xxxx xxxx	保持
各模块中断源寄存器 2	INT2B2	H'xxxx xxxx	保持
各模块中断源寄存器 3	INT2B3	H'xxxx xxxx	保持
各模块中断源寄存器 4	INT2B4	H'xxxx xxxx	保持
各模块中断源寄存器 5	INT2B5	H'xxxx xxxx	保持
各模块中断源寄存器 6	INT2B6	H'xxxx xxxx	保持
各模块中断源寄存器 7	INT2B7	H'xxxx xxxx	保持
GPIO 中断设定寄存器	INT2GPIC	H'0000 0000	保持

13.3.1 中断控制寄存器 0 (ICR0)

ICR0 设定 NMI 引脚的输入信号检测模式，表示 NMI 引脚的输入电平。

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

位	位名	初始值	R/W	说明
31	NMIL	不定值	R	NMI 输入电平位 表示 NMI 引脚的输入信号电平。 通过读此位，能知道 NMI 引脚的电平。 0: NMI 引脚输入了低电平 1: NMI 引脚输入了高电平 此位的写操作无效。
30	MAI	0	R/W	MAI 中断屏蔽位 与 CPU 的 SR.BL 位无关，指定在 NMI 引脚的输入电平为低电平期间是否屏蔽全部中断。 0: 即使 NMI 为低电平也允许中断。 1: 在 NMI 为高电平期间，禁止中断。
29 ~ 26	—	全 0	R	保留位 读取值为 0，只能写 0。
25	NMIB	0	R/W	NMI 块模式位 当 CPU 的 SR.BL 位为 1 时，选择在 SR.BL 位变为 0 前是保留还是立即检测 NMI 中断。 0: 当 SR.BL=1 时，保留 NMI 中断请求（初始值）。 1: 当 SR.BL=1 时，不保留 NMI 中断。 【注】如果在 SR.BL=1 的状态下接受中断，以前的异常信息（SSR、SPC、SGR、INTEVT）就会丢失。
24	NMIE	0	R/W	NMI 边沿选择位 选择是通过下降沿还是通过上升沿检测 NMI 引脚的中断请求信号。 0: 通过 NMI 输入的下降沿检测中断请求（初始值）。 1: 通过 NMI 输入的上升沿检测中断请求。
23	—	1	R	保留位 读取值为 1，只能写 1。
22 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

13.3.2 中断控制寄存器 1 (ICR1)

ICR1 是 32 位可读写寄存器，对于外部中断输入引脚 IRQ1 和 IRQ0，给各引脚指定下降沿、上升沿、低电平或者高电平的检测模式。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IRQ0S		IRQ1S		—	—	—	—	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
31、30	IRQ0S	00	R/W	IRQn 检测选择位 选择通过下降沿、上升沿、低电平或者高电平检测 IRQ1 和 IRQ0 引脚的中断信号。 IRQnS IRQn 中断请求的检测方法 00 通过 IRQn 输入的下降沿检测中断请求 01 通过 IRQn 输入的上升沿检测中断请求 10 通过 RQn 输入的低电平检测中断请求 11 通过 IRQn 输入的高电平检测中断请求 【注】 n=0, 1
29、28	IRQ1S	00	R/W	
27 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

【注】 当 IRQ 为设定的电平输入 (IRQnS[1]) 时，就将中断源保持到 CPU 接受某个中断为止 (不限于 IRQ)。因此，从 SLEEP 返回时，即使在返回前中断源被取消，也转移到中断处理程序。能通过将对对应中断的屏蔽位 (中断屏蔽寄存器的 IM 位) 置 1 来清除所保持的中断。

13.3.3 中断优先顺序设定寄存器 (INTPRI)

INTPRI 是 32 位可读写寄存器，设定 IRQ1 和 IRQ0 中断的优先顺序 (中断级 15 ~ 0)。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IP0				IP1				—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
31 ~ 28	IP0	H'0	R/W	是 IRQ0 独立的中断请求的优先顺序。
27 ~ 24	IP1	H'0	R/W	是 IRQ1 独立的中断请求的优先顺序。
23 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

必须给各 4 位字段设定 H'F ~ H'1 的值决定中断优先顺序，值越大优先级越高。
如果设定 H'0，中断就被屏蔽（初始值）。

13.3.4 中断源寄存器 (INTREQ)

INTREQ 是 32 位可读写（带条件）寄存器，表示 INTC 发生哪种 IRQ_n（n=0, 1）中断请求。
即使通过 INTPRI 和 INTMSK 屏蔽中断，也不影响此寄存器的位。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IR0	IR1	—	—	—	—	—	—	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R
位:	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	
				边沿检测时 (ICR1.IRQnS=00 或者 01)	电平检测时 (ICR1.IRQnS=10 或者 11)
31	IR0	0	R/W	读时	读时
30	IR1	0	R/W	0: 未检测到对应的中断请求。 1: 检测到对应的中断请求。 写时 * 0: 只将读到 1 的位清 0。 1: 保持检测到的中断请求。	0: 对应的中断引脚为无效电平。 1: 对应的中断引脚为有效电平， 但是 CPU 尚未接受中断。 写无效。
29 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。	

【注】 n=0, 1

* 必须给读到 0 的位写 1。

13.3.5 中断屏蔽寄存器 (INTMSK)

INTMSK 是 32 位可读写 (带条件) 寄存器, 设定是否屏蔽各 IRQn (n=0, 1) 中断请求。在要解除中断屏蔽时, 给 INTMSKCLR 寄存器的对应位写 1。即使给此寄存器的各位写 0, 值也不变。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IM00	IM01	—	—	—	—	—	—	—	—	—	—	—	—	—	—
初始值:	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明	
31	IM00	1	R/W	屏蔽 IRQ0 独立的中断源。	读时 0: 接受中断 1: 屏蔽中断 写时 0: 无效 1: 屏蔽中断
30	IM01	1	R/W	屏蔽 IRQ1 独立的中断源。	
29 ~ 0	—	全 0	R	保留位 读取值为 0, 只能写 0。	

13.3.6 中断屏蔽清除寄存器 (INTMSKCLR)

INTMSKCLR 是 32 位只写寄存器, 清除各 IRQn (n=0, 1) 中断请求的屏蔽。不保证读取值。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IC00	IC01	—	—	—	—	—	—	—	—	—	—	—	—	—	—
初始值:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
初始值:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

位	位名	初始值	R/W	说明	
31	IC00	不定值	W	清除 IRQ0 独立的中断源的屏蔽。	读时 返回不定值。 写时 0: 无效 1: 清除中断屏蔽 (允许中断)
30	IC01	不定值	W	清除 IRQ1 独立的中断源的屏蔽。	
29 ~ 0	—	全为不定值	W	保留位 只能写 0。	

13.3.7 NMI 标志控制寄存器 (NMIFCR)

NMIFCR 是 32 位可读写 (部分带条件) 寄存器, 有 NMI 标志 (NMIFL 位)。当 INTC 检测到 NMI 时, NMIFL 位就自动置 1。通过写 0 来清除 NMIFL 位。

NMIFL 位的值不影响 CPU 接受 NMI。即, INTC 检测到的 NMI 请求是通过被 CPU 接受而清除的, 但是 NMIFL 位不会自动清除。在 CPU 接受 NMI 请求前, 即使给 NMIFL 位写 0, 也不会取消 NMI 请求。

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

位	位名	初始值	R/W	说明
31	NMIL	不定值	R	NMI 输入电平位 0: NMI 引脚输入了低电平 1: NMI 引脚输入了高电平 此位的写操作无效。
30 ~ 17	—	全 0	R	保留位 读取值为 0, 只能写 0。
16	NMIFL	0	R/W	NMI 标志位 (检测到 NMI 中断请求信号) 读时 1: 检测到 NMI 0: 未检测到 NMI 写时 0: 清除 NMI 标志 1: 无效 (忽视写 1)
15 ~ 0	—	全 0	R	保留位 读取值为 0, 只能写 0。

13.3.8 用户中断屏蔽级设定寄存器 (USERIMASK)

USERIMASK 是 32 位可读写 (部分带条件) 寄存器, 用于设定能接受的中断级。在使用 MMU 的地址转换对区域 7 地址进行存取时, 能在用户模式中存取此寄存器。不同于 INTC 的其他寄存器, USERIMASK 被分配在 64K 字节页面, 所以只有此寄存器能在用户模式中进行存取。

屏蔽 UIMASK 设定值以下的中断级的中断。当设定 H'F 时, 就屏蔽 NMI 以外的全部中断。

虽然接受高于 UIMASK 设定值的中断级的中断, 但是只在中断屏蔽寄存器的对应的中断屏蔽位为 0 (允许中断) 并且 SR.IMASK 位低于此中断级的情况下才能接受中断。

即使接受中断, UIMASK 的值也不变。

在上电复位时, 被初始化为 H'0000 0000 (允许全部中断)。

为了防止误写, 只在 bit31 ~ 24 为 H'A5 时, 此寄存器的写操作才有效。

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

位	位名	初始值	R/W	说明
31 ~ 24	—	H'00	R/W	保留位 读取值为 0。 在要给 UIMASK 位写值时，必须将此位设定为 H'A5（必须写 H'A5）。
23 ~ 8	—	全 0	R	保留位 读取值为 0，只能写 0。
7 ~ 4	UIMASK	H'0	R/W	用户中断屏蔽级位 屏蔽 UIMASK 设定值以下的中断级的中断。
3 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

(1) 用户中断屏蔽级设定寄存器的使用步骤

通过给 USERIMASK 设定中断屏蔽级，能禁止设定值以下的中断级的中断。在通过此功能处理紧急度高的任务（运行在用户模式的设备驱动程序等）时，能通过禁止紧急度低的中断来缩短时间。

不同于 INTC 的其他寄存器，USERIMASK 被分配在 64KB 空间。在用户模式中存取此寄存器时，通过 MMU 的地址转换进行存取。在多任务 OS 的情况下，必须通过 MMU 的存储保护来管理能存取 USERIMASK 的程序。另外，当结束该任务或者切换到其他任务时，必须将 UIMASK 位清 0。如果误将 UIMASK 位设定为 0 以外的值而结束任务，就有可能一直禁止该中断级以下的中断，引起不进行 OS 任务的切换等问题。

使用步骤例子如下所示：

- 先将中断分为 (A) 和 (B) 2 类，设定 (A) 的中断级比 (B) 高。
 - 设备驱动程序中应接受的中断
(OS 使用的中断，定时器中断等)
 - 设备驱动程序中应禁止的中断
- 设定 MMU，只对要禁止中断的设备驱动程序，允许存取 USERIMASK 所在的地址空间。
- 转移到设备驱动程序。
- 在用户模式运行的设备驱动程序中设定 UIMASK 位，屏蔽 (B) 的中断。
- 在设备驱动程序中进行高紧急度的处理。
- 将 UIMASK 位清 0，从设备驱动程序返回。

13.3.9 中断优先顺序设定寄存器 (INT2PRI0 ~ INT2PRI12)

中断优先顺序设定寄存器 (INT2PRI0 ~ INT2PRI12) 设定内部模块中断的优先顺序 (别 31 ~ 0)。

INT2PRI0 ~ INT2PRI12 是 32 位可读写寄存器。在复位时, 被初始化为 0。

此寄存器能用 5 位 32 种组合将各种中断源设定为 30 个优先级 (H'00 和 H'01 为屏蔽中断请求)。

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

中断请求源和 INT2PRI0 ~ INT2PRI12 寄存器各位的对应如表 13.5 所示。

表 13.5 中断请求源和 INT2PRI0 ~ INT2PRI12 寄存器

寄存器	位			
	28 ~ 24	20 ~ 16	12 ~ 8	4 ~ 0
INT2PRI0	TMU0 (TUNI0)	TMU0 (TUNI1)	TMU0 (TUNI2)	TMU0 (TICPI2)
INT2PRI1	TMU1 (TUNI3)	TMU1 (TUNI4)	TMU1 (TUNI5)	保留
INT2PRI2	SCIF0	SCIF1	WDT	保留
INT2PRI3	H-UDI	DMAC	保留	保留
INT2PRI4	保留	G2D	SSI_A (SSIDMA0)	SSI_A (SSICH0)
INT2PRI5	SSI_A (SSICH1)	SSI_A (SSICH2)	保留	SSI_B
INT2PRI6	ATAPI	保留	FLCTL	SRC (OVF)
INT2PRI7	SCIF2	GPIO	保留	保留
INT2PRI8	保留	SRC (ODFI)	SRC (IDEI)	保留
INT2PRI9	LCDC	保留	保留	IIC
INT2PRI10	保留	保留	保留	保留
INT2PRI11	保留	保留	保留	保留
INT2PRI12	VDC2	保留	USB	EtherC

【注】 值越大优先级越高。设定值 H'00 和 H'01 为屏蔽中断请求的状态。

13.3.10 中断源寄存器 0 (不影响屏蔽状态) (INT2A0)

INT2A0 是 32 位只读寄存器, 表示内部模块的中断源。即使给中断屏蔽寄存器设定了中断屏蔽, 此寄存器也表示相应位的中断源 (不产生相应位的中断)。如果不需要根据中断屏蔽寄存器的状态表示中断源, 就必须使用 INT2A1 寄存器。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GPIO	—	SRC	FLCTL	—	ATAPI	SSI_B	—	SSI_A CH2	SSI_A CH1
初始值:	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
	SSI_A CH0	SSI_A DMA0	G2D	—	—	—	—	DMAC	H-UDI	—	WDT	SCIF1	SCIF0	—	TMU1	TMU0
初始值:	—	—	—	0	0	0	0	—	—	0	—	—	—	0	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明	
31 ~ 26	—	全 0	R	保留位 读取值为 0。	按内部模块的种类表示中断源 (是不受中断屏蔽寄存器状态影响的寄存器)。 0: 无中断 1: 产生中断 【注】 通过直接读取通知 CPU 的 INTEVT (异常码), 能确定中断源。此时不需要读此寄存器。 写无效。 各内部模块保留中断源。
25	GPIO	不定值	R	表示 GPIO 中断源。	
24	—	0	R	保留位 读取值为 0。	
23	SRC	不定值	R	表示 SRC OVF 中断源。	
22	FLCTL	不定值	R	表示 FLCTL 中断源。	
21	—	0	R	保留位 读取值为 0。	
20	ATAPI	不定值	R	表示 ATAPI 中断源。	
19	SSI_B	不定值	R	表示 SSI_B 中断源。	
18	—	0	R	保留位 读取值为 0。	
17	SSI_ACH2	不定值	R	表示 SSI_A (SSICH2) 中断源。	
16	SSI_ACH1	不定值	R	表示 SSI_A (SSICH1) 中断源。	
15	SSI_ACH0	不定值	R	表示 SSI_A (SSICH0) 中断源。	
14	SSI_ADMA0	不定值	R	表示 SSI_A (SSIDMA0) 中断源。	
13	G2D	不定值	R	表示 G2D 中断源。	
12 ~ 9	—	全 0	R	保留位 读取值为 0。	
8	DMAC	不定值	R	表示 DMAC 通道 0 ~ 5 的中断源和地址错误中断。	
7	H-UDI	不定值	R	表示 H-UDI 中断源。	
6	—	0	R	保留位 读取值为 0。	
5	WDT	不定值	R	表示 WDT 中断源。	
4	SCIF1	不定值	R	表示 SCIF1 中断源。	
3	SCIF0	不定值	R	表示 SCIF0 中断源。	
2	—	0	R	保留位 读取值为 0。	
1	TMU1	不定值	R	表示 TMU1 中断源。	
0	TMU10	不定值	R	表示 TMU0 中断源。	

13.3.11 中断源寄存器 01 (不影响屏蔽状态) (INT2A01)

INT2A01 是 32 位只读寄存器，表示内部模块的中断源。即使给中断屏蔽寄存器设定了中断屏蔽，此寄存器也表示相应位的中断源（不产生相应位的中断）。如果不需要根据中断屏蔽寄存器的状态表示中断源，就必须使用 INT2A11 寄存器。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SCIF2	—	—	—	—	—	VDC2	—	USB	EtherC
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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	LCDC	—	—	IIC	—	SRC ODFI	SRC IDEI	—
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	
31 ~ 26	—	全 0	R	保留位 读取值为 0。	按内部模块的种类表示中断源（是不受中断屏蔽寄存器状态影响的寄存器）。 0: 无中断 1: 产生中断 【注】 通过直接读取通知 CPU 的 INTEVT（异常码），能确定中断源。此时不需要读此寄存器。 写无效。 各内部模块保留中断源。但是，当解除 SRC 的 ODFI 和 IDEI 中断源时，此寄存器的值也为 0。
25	SCIF2	不定值	R	表示 SCIF2 中断源。	
24 ~ 20	—	全 0	R	保留位 读取值为 0。	
19	VDC2	不定值	R	表示 VDC2 中断源。	
18	—	0	R	保留位 读取值为 0。	
17	USB	不定值	R	表示 USB 中断源。	
16	EtherC	不定值	R	表示 EtherC 中断源。	
15 ~ 8	—	全 0	R	保留位 读取值为 0。	
7	LCDC	不定值	R	表示 LCDC 中断源。	
6、5	—	全 0	R	保留位 读取值为 0。	
4	IIC	不定值	R	表示 IIC 中断源。	
3	—	0	R	保留位 读取值为 0。	
2	SRCODFI	不定值	R	表示 SRC ODFI 中断源。	
1	SRCIDEI	不定值	R	表示 SRC IDEI 中断源。	
0	—	0	R	保留位 读取值为 0。	

13.3.12 中断源寄存器 1 (影响屏蔽状态) (INT2A1)

INT2A1 是 32 位只读寄存器，表示内部模块的中断源。但是，如果给中断屏蔽寄存器设定了中断屏蔽，就不表示相应位（不置 1）。与中断屏蔽寄存器的状态无关，在确认是否发生中断时，必须使用 INT2A0 寄存器

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GPIO	—	SRC	FLCTL	—	ATAPI	SSI_B	—	SSI_A CH2	SSI_A CH1
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
	SSI_A CH0	SSI_A DMA0	G2D	—	—	—	—	DMAC	H-UDI	—	WDT	SCIF1	SCIF0	—	TMU1	TMU0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
31 ~ 26	—	全 0	R	保留位 读取值为 0。
25	GPIO	0	R	表示 GPIO 中断源。
24	—	0	R	保留位 读取值为 0。
23	SRC	0	R	表示 SRC OVF 中断源。
22	FLCTL	0	R	表示 FLCTL 中断。
21	—	0	R	保留位 读取值为 0。
20	ATAPI	0	R	表示 ATAPI 中断源。
19	SSI_B	0	R	表示 SSI_B 中断源。
18	—	0	R	保留位 读取值为 0。
17	SSI_ACH2	0	R	表示 SSI_A (SSICH2) 中断源。
16	SSI_ACH1	0	R	表示 SSI_A (SSICH1) 中断源。
15	SSI_ACH0	0	R	表示 SSI_A (SSICH0) 中断源。
14	SSI_ADMA0	0	R	表示 SSI_A (SSIDMA0) 中断源。
13	G2D	0	R	表示 G2D 中断源。
12 ~ 9	—	全 0	R	保留位 读取值为 0。
8	DMAC	0	R	表示 DMAC 通道 0 ~ 5 的中断源。
7	H-UDI	0	R	表示 H-UDI 中断源。
6	—	0	R	保留位 读取值为 0。
5	WDT	0	R	表示 WDT 中断源。
4	SCIF1	0	R	表示 SCIF1 中断源。
3	SCIF0	0	R	表示 SCIF0 中断源。
2	—	0	R	保留位 读取值为 0。
1	TMU1	0	R	表示 TMU1 中断源。
0	TMU0	0	R	表示 TMU0 中断源。

按内部模块的种类表示中断源
(是受中断屏蔽寄存器状态影响的寄存器)
0: 无中断
1: 产生中断
【注】通过直接读取通知 CPU 的 INTEVT (异常码), 能确定中断源。此时不需要读此寄存器。
写无效。
各内部模块保留中断源。

13.3.13 中断源寄存器 11 (影响屏蔽状态) (INT2A11)

INT2A11 为 32 位只读寄存器，表示内部模块的中断源。但是，如果给中断屏蔽寄存器设定了中断屏蔽，就不表示相应位（不置 1）。与中断屏蔽寄存器的状态无关，在确认是否发生中断时，必须使用 INT2A01 寄存器。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SCIF2	—	—	—	—	—	VDC2	—	USB	EtherC
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
	—	—	—	—	—	—	—	—	LCDC	—	—	IIC	—	SRC ODFI	SRC IDEI	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明	
31 ~ 26	—	全 0	R	保留位 读取值为 0。	按内部模块的种类表示中断源（是受中断屏蔽寄存器影响的寄存器）。 0: 无中断 1: 产生中断 【注】 通过直接读取通知 CPU 的 INTEVT（异常码），能确定中断源。此时不需要读此寄存器。 写无效。 各内部模块保留中断源。 但是，当解除 SRC 的 ODFI 和 IDEI 中断源时，此寄存器的值也为 0。
25	SCIF2	0	R	表示 SCIF2 中断源。	
24 ~ 20	—	全 0	R	保留位 读取值为 0。	
19	VDC2	0	R	表示 VDC2 中断源。	
18	—	0	R	保留位 读取值为 0。	
17	USB	0	R	表示 USB 中断源。	
16	EtherC	0	R	表示 EtherC 中断源。	
15 ~ 8	—	全 0	R	保留位 读取值为 0。	
7	LCDC	0	R	表示 LCDC 中断源。	
6、5	—	全 0	R	保留位 读取值为 0。	
4	IIC	0	R	表示 IIC 中断源。	
3	—	0	R	保留位 读取值为 0。	
2	SRCODFI	0	R	表示 SRC ODFI 中断源。	
1	SRCIDEI	0	R	表示 SRC IDEI 中断源。	
0	—	0	R	保留位 读取值为 0。	

13.3.14 中断屏蔽寄存器 (INT2MSKR)

INT2MSKR 是 32 位可读写寄存器，能对中断源寄存器所表示的中断源分别设定中断屏蔽。此寄存器设定为 1 的对应中断源不通知中断。在复位时，被初始化为 H'FFFF FFFF（全部屏蔽）。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GPIO	—	SRC	FLCTL	—	ATAPI	SSI_B	—	SSI_A CH2	SSI_A CH1
初始值:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R/W	R	R/W	R/W	R	R/W	R/W	R	R/W	R/W

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SSI_A CH0	SSI_A DMA0	G2D	—	—	—	—	DMAC	H-UDI	—	WDT	SCIF1	SCIF0	—	TMU1	TMU0
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	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 26	—	全 1	R	保留位 读取值为 1, 只能写 1。
25	GPIO	1	R/W	设定 GPIO 的中断屏蔽。
24	—	1	R	保留位 读取值为 1, 只能写 1。
23	SRC	1	R/W	设定 SRC OVF 的中断屏蔽。
22	FLCTL	1	R/W	设定 FLCTL 的中断屏蔽。
21	—	1	R	保留位 读取值为 1, 只能写 1。
20	ATAPI	1	R/W	设定 ATAPI 的中断屏蔽。
19	SSI_B	1	R/W	设定 SSI_B 的中断屏蔽。
18	—	1	R	保留位 读取值为 1, 只能写 1。
17	SSI_ACH2	1	R/W	设定 SSI_A (SSICH2) 的中断屏蔽。
16	SSI_ACH1	1	R/W	设定 SSI_A (SSICH1) 的中断屏蔽。
15	SSI_ACH0	1	R/W	设定 SSI_A (SSICH0) 的中断屏蔽。
14	SSI_ADMA0	1	R/W	设定 SSI_A (SSIDMA0) 的中断屏蔽。
13	G2D	1	R/W	设定 G2D 的中断屏蔽。
12 ~ 9	—	全 1	R	保留位 读取值为 1, 只能写 1。
8	DMAC	1	R/W	设定 DMAC 通道 0 ~ 5 的中断屏蔽。
7	H-UDI	1	R/W	设定 H-UDI 的中断屏蔽。
6	—	1	R	保留位 读取值为 1, 只能写 1。
5	WDT	1	R/W	设定 WDT 的中断屏蔽。
4	SCIF1	1	R/W	设定 SCIF1 的中断屏蔽。
3	SCIF0	1	R/W	设定 SCIF0 的中断屏蔽。
2	—	1	R	保留位 读取值为 1, 只能写 1。
1	TMU1	1	R/W	设定 TMU1 的中断屏蔽。
0	TMU0	1	R/W	设定 TMU0 的中断屏蔽。

按内部模块设定中断屏蔽。
写时
0: 无效
1: 设定中断屏蔽
读时
0: 无屏蔽设定
1: 有屏蔽设定

13.3.15 中断屏蔽寄存器 1 (INT2MSKR1)

INT2MSKR1 是 32 位可读写寄存器，能对中断源寄存器所表示的中断源分别设定中断屏蔽。此寄存器设定为 1 的对应中断源不通知中断。在复位时，被初始化为 H'FFFF FFFF（全部屏蔽）。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SCIF2	—	—	—	—	—	VDC2	—	USB	EtherC
初始值:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	LCDC	—	—	IIC	—	—	SRC ODFI	SRC IDEI
初始值:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R	R/W	R/W	R

位	位名	初始值	R/W	说明	
31 ~ 26	—	全 1	R	保留位 读取值为 1，只能写 1。	按内部模块设定中断屏蔽。 写时 0: 无效 1: 设定中断屏蔽 读时 0: 无屏蔽设定 1: 有屏蔽设定
25	SCIF2	1	R/W	设定 SCIF2 的中断屏蔽。	
24 ~ 20	—	全 1	R	保留位 读取值为 1，只能写 1。	
19	VDC2	1	R/W	设定 VDC2 的中断屏蔽。	
18	—	1	R	保留位 读取值为 1，只能写 1。	
17	USB	1	R/W	设定 USB 的中断屏蔽。	
16	EtherC	1	R/W	设定 EtherC 的中断屏蔽。	
15 ~ 8	—	全 1	R	保留位 读取值为 1，只能写 1。	
7	LCDC	1	R/W	设定 LCDC 的中断屏蔽。	
6、5	—	全 1	R	保留位 读取值为 1，只能写 1。	
4	IIC	1	R/W	设定 IIC 的中断屏蔽。	
3	—	1	R	保留位 读取值为 1，只能写 1。	
2	SRCODFI	1	R/W	设定 SRC ODFI 的中断屏蔽。	
1	SRCIDEI	1	R/W	设定 SRC IDEI 的中断屏蔽。	
0	—	1	R	保留位 读取值为 1，只能写 1。	

13.3.16 中断屏蔽清除寄存器 (INT2MSKCR)

INT2MSKCR 是 32 位只写寄存器，用于清除中断屏蔽寄存器设定的屏蔽。当将此寄存器的相应位置 1 时，就清除该中断源的屏蔽。读取值为 0。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GPIO	—	SRC	FLCTL	—	ATAPI	SSI_B	—	SSI_A CH2	SSI_A CH1
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SSI_A CH0	SSI_A DMA0	G2D	—	—	—	—	DMAC	H-UDI	—	WDT	SCIF1	SCIF0	—	TMU1	TMU0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

位	位名	初始值	R/W	说明	
31 ~ 26	—	全 0	W	保留位 只能写 0。	清除各内部模块的中断屏蔽。 写时 0: 无效 1: 清除中断屏蔽 (允许中断) 读取值为 0。
25	GPIO	0	W	清除 GPIO 的中断屏蔽。	
24	—	0	W	保留位 只能写 0。	
23	SRC	0	W	清除 SRC OVF 的中断屏蔽。	
22	FLCTL	0	W	清除 FLCTL 的中断屏蔽。	
21	—	0	W	保留位 只能写 0。	
20	ATAPI	0	W	清除 ATAPI 的中断屏蔽。	
19	SSI_B	0	W	清除 SSI_B 的中断屏蔽。	
18	—	0	W	保留位 只能写 0。	
17	SSI_ACH2	0	W	清除 SSI_A (SSICH2) 的中断屏蔽。	
16	SSI_ACH1	0	W	清除 SSI_A (SSICH1) 的中断屏蔽。	
15	SSI_ACH0	0	W	清除 SSI_A (SSICH0) 的中断屏蔽。	
14	SSI_ADMA0	0	W	清除 SSI_A (SSIDMA0) 的中断屏蔽。	
13	G2D	0	W	清除 G2D 的中断屏蔽。	
12 ~ 9	—	全 0	W	保留位 只能写 0。	
8	DMAC	0	W	清除 DMAC 的中断屏蔽。	
7	H-UDI	0	W	清除 H-UDI 的中断屏蔽。	
6	—	0	W	保留位 只能写 0。	
5	WDT	0	W	清除 WDT 的中断屏蔽。	
4	SCIF1	0	W	清除 SCIF1 的中断屏蔽。	
3	SCIF0	0	W	清除 SCIF0 的中断屏蔽。	
2	—	0	W	保留位 只能写 0。	
1	TMU1	0	W	清除 TMU1 的中断屏蔽。	
0	TMU0	0	W	清除 TMU0 的中断屏蔽。	

13.3.17 中断屏蔽清除寄存器 1 (INT2MSKCR1)

INT2MSKCR1 是 32 位只写寄存器，用于清除中断屏蔽寄存器设定的屏蔽。当此寄存器的相应位置 1 时，就清除该中断源的屏蔽。读取值为 0。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SCIF2	—	—	—	—	—	VDC2	—	USB	EtherC
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	LCDC	—	—	IIC	—	SRC ODFI	SRC IDEI	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

位	位名	初始值	R/W	说明	
31 ~ 26	—	全 0	W	保留位 只能写 0。	清除各内部模块的中断屏蔽。 写时 0: 无效 1: 清除中断屏蔽 (允许中断) 读取值为 0
25	SCIF2	0	W	清除 SCIF2 的中断屏蔽。	
24 ~ 20	—	全 0	W	保留位 只能写 0	
19	VDC2	0	W	清除 VDC2 的中断屏蔽。	
18	—	0	W	保留位 只能写 0。	
17	USB	0	W	清除 USB 的中断屏蔽。	
16	EtherC	0	W	清除 EtherC 的中断屏蔽。	
15 ~ 8	—	全 0	W	保留位 只能写 0。	
7	LCDC	0	W	清除 LCDC 的中断屏蔽。	
6、5	—	全 0	W	保留位 只能写 0。	
4	IIC	0	W	清除 IIC 的中断屏蔽。	
3	—	0	W	保留位 只能写 0。	
2	SRCODFI	0	W	清除 SRC ODFI 的中断屏蔽。	
1	SRCIDEI	0	W	清除 SRC IDEI 的中断屏蔽。	
0	—	0	W	保留位 只能写 0。	

13.3.18 各内部模块中断源寄存器 (INT2B0、INT2B2 ~ INT2B7)

INT2B0、INT2B2 ~ INT2B7 是 32 位只读寄存器，对中断源寄存器所表示的各模块中断源，更加详细地表示各个中断源。这些寄存器不受中断屏蔽设定寄存器的屏蔽状态的影响。在对这些详细的中断源分别设定中断屏蔽时，需要设定相应模块的中断屏蔽寄存器或者中断允许寄存器。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
初始值:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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
	—	—	—	—	—	—	—	—	—	TUNI5	TUNI4	TUNI3	TICPI2	TUNI2	TUNI1	TUNI0
初始值:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

(1) INT2B0 寄存器: TMU 模块的详细中断源

模块	位	源	说明
TMU 模块	31 ~ 7	—	保留位 读取值为 0，写无效。
	6	TUNI5	TMU 通道 5 的下溢中断
	5	TUNI4	TMU 通道 4 的下溢中断
	4	TUNI3	TMU 通道 3 的下溢中断
	3	TICPI2	TMU 通道 2 的输入捕捉中断
	2	TUNI2	TMU 通道 2 的下溢中断
	1	TUNI1	TMU 通道 1 的下溢中断
	0	TUNI0	TMU 通道 0 的下溢中断

(2) INT2B2 寄存器: SCIF 模块的详细中断源

模块	位	源	说明
SCIF1 模块	31 ~ 8	—	保留位 读取值为 0，写无效。
	7	TXI1	因 SCIF 通道 1 的发送 FIFO 数据空而产生的中断
	6	BRI1	因 SCIF 通道 1 的中止或者溢出错误而产生的中断
	5	RXI1	因 SCIF 通道 1 的接收 FIFO 数据满或者接收数据就绪而产生的中断
	4	ERI1	SCIF 通道 1 的接收错误中断
SCIF0 模块	3	TXI0	因 SCIF 通道 0 的发送 FIFO 数据空而产生的中断
	2	BRI0	因 SCIF 通道 0 的中止或者溢出错误而产生的中断
	1	RXI0	因 SCIF 通道 0 的接收 FIFO 数据满或者接收数据就绪而产生的中断
	0	ERI0	SCIF 通道 0 的接收错误中断

(3) INT2B3 寄存器: DMAC 模块的详细中断源

模块	位	源		说明
DMAC 模块	31 ~ 13	—	保留位 读取值为 0, 写无效。	表示 DMAC 的中断源。 即使中断屏蔽寄存器设定了 DMAC 屏蔽, 此寄存器也表示中 断源。
	12	DMAE	通道 0 ~ 5 的 DMA 地址错误中断	
	11 ~ 6	—	保留位 读取值为 0, 写无效。	
	5	DMINT5	通道 5 的 DMA 传送结束中断	
	4	DMINT4	通道 4 的 DMA 传送结束中断	
	3	DMINT3	通道 3 的 DMA 传送结束 / 半结束 中断	
	2	DMINT2	通道 2 的 DMA 传送结束 / 半结束 中断	
	1	DMINT1	通道 1 的 DMA 传送结束 / 半结束 中断	
	0	DMINT0	通道 0 的 DMA 传送结束 / 半结束 中断	

(4) INT2B4 寄存器: SSI 模块的详细中断源

模块	位	源		说明
SSI 模块	31 ~ 9	—	保留位 读取值为 0, 写无效。	表示 SSI 的中断源。 即使中断屏蔽寄存器设定了 SSI 屏蔽, 此寄存器也表示中断源。
	8	SSICH5	SSI ch5 中断	
	7	SSICH4	SSI ch4 中断	
	6	SSICH3	SSI ch3 中断	
	5	SSIDMA1	SSI DMA1 中断	
	4	—	保留位 读取值为 0, 写无效	
	3	SSICH2	SSI ch2 中断	
	2	SSICH1	SSI ch1 中断	
	1	SSICH0	SSI ch0 中断	
	0	SSIDMA0	SSI DMA0 中断	

(5) INT2B5 寄存器: FLCTL 模块的详细中断源

模块	位	源		说明
FLCTL 模块	31 ~ 4	—	保留位 读取值为 0, 写无效。	表示 FLCTL 的中断源。 即使中断屏蔽寄存器设定了 FLCTL 屏蔽, 此寄存器也表示中 断源。
	3	FLTRQ1	FLCTL FLECFIFO 传送请求中断	
	2	FLTRQ0	FLCTL TLDFIFO 传送请求中断	
	1	FLTEND	FLCTL 传送结束中断	
	0	FLSTE	FLCTL 状态错误或者就绪 / 忙超时 错误中断	

(6) INT2B6 寄存器: SCIF2 模块的详细中断源

模块	位	源		说明
SCIF2 模块	31 ~ 4	—	保留位 读取值为 0, 写无效。	表示 SCIF2 的中断源。 即使中断屏蔽寄存器设定了 SCIF2 屏蔽, 此寄存器也表示中断源。
	3	TXI2	通道 2 的发送 FIFO 数据空中断	
	2	BRI2	通道 2 的中止或者溢出错误中断	
	1	RXI2	通道 2 的接收 FIFO 数据满或者接收数据就绪中断	
	0	ERI2	通道 2 的接收错误中断	

(7) INT2B7 寄存器: GPIO 模块的详细中断源

模块	位	源		说明
GPIO 模块	31 ~ 28	—	保留位 读取值为 0, 写无效。	表示 GPIO 的中断源。 即使中断屏蔽寄存器设定了 GPIO 屏蔽, 此寄存器也表示中断源。
	27	PINT15I	PINT15 引脚的 GPIO 通道 3 中断	
	26	PINT14I	PINT14 引脚的 GPIO 通道 3 中断	
	25	PINT13I	PINT13 引脚的 GPIO 通道 3 中断	
	24	PINT12I	PINT12 引脚的 GPIO 通道 3 中断	
	23 ~ 20	—	保留位 读取值为 0, 写无效	
	19	PINT11I	PINT11 引脚的 GPIO 通道 2 中断	
	18	PINT10I	PINT10 引脚的 GPIO 通道 2 中断	
	17	PINT9I	PINT9 引脚的 GPIO 通道 2 中断	
	16	PINT8I	PINT8 引脚的 GPIO 通道 2 中断	
	15 ~ 12	—	保留位 读取值为 0, 写无效。	
	11	PINT7I	PINT7 引脚的 GPIO 通道 1 中断	
	10	PINT6I	PINT6 引脚的 GPIO 通道 1 中断	
	9	PINT5I	PINT5 引脚的 GPIO 通道 1 中断	
	8	PINT4I	PINT4 引脚的 GPIO 通道 1 中断	
	7 ~ 4	—	保留位 读取值为 0, 写无效。	
	3	PINT3I	PINT3 引脚的 GPIO 通道 0 中断	
	2	PINT2I	PINT2 引脚的 GPIO 通道 0 中断	
	1	PINT1I	PINT1 引脚的 GPIO 通道 0 中断	
	0	PINT0I	PINT0 引脚的 GPIO 通道 0 中断	

13.3.19 GPIO 中断设定寄存器 (INT2GPIC)

INT2GPIC 允许中断端口 A0 ~ 7 和 B0 ~ 7 的 GPIO 中断请求输入。

GPIO 中断是低电平有效的中断。在通过用作 GPIO 中断的各端口控制寄存器 (A、B) 将相应引脚设定为端口输入后, 必须允许中断请求。有关端口控制寄存器, 请参照“第 27 章 通用输入/输出端口 (GPIO)”。

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

位	位名	初始值	R/W	说明	
31 ~ 28	—	全 0	R	保留位 读取值为 0, 只能写 0。	允许各 GPIO 中断输入引脚的中断请求。 0: 禁止中断请求 1: 允许中断请求
27	PINT15E	0	R/W	允许 PINT15 引脚的 GPIO 通道 3 的中断请求。	
26	PINT14E	0	R/W	允许 PINT14 引脚的 GPIO 通道 3 的中断请求。	
25	PINT13E	0	R/W	允许 PINT13 引脚的 GPIO 通道 3 的中断请求。	
24	PINT12E	0	R/W	允许 PINT12 引脚的 GPIO 通道 3 的中断请求。	
23 ~ 20	—	全 0	R	保留位 读取值为 0, 只能写 0。	
19	PINT11E	0	R/W	允许 PINT11 引脚的 GPIO 通道 2 的中断请求。	
18	PINT10E	0	R/W	允许 PINT10 引脚的 GPIO 通道 2 的中断请求。	
17	PINT9E	0	R/W	允许 PINT9 引脚的 GPIO 通道 2 的中断请求。	
16	PINT8E	0	R/W	允许 PINT8 引脚的 GPIO 通道 2 的中断请求。	
15 ~ 12	—	全 0	R	保留位 读取值为 0, 只能写 0。	
11	PINT7E	0	R/W	允许 PINT7 引脚的 GPIO 通道 1 的中断请求。	
10	PINT6E	0	R/W	允许 PINT6 引脚的 GPIO 通道 1 的中断请求。	
9	PINT5E	0	R/W	允许 PINT5 引脚的 GPIO 通道 1 的中断请求。	
8	PINT4E	0	R/W	允许 PINT4 引脚的 GPIO 通道 1 的中断请求。	
7 ~ 4	—	全 0	R	保留位 读取值为 0, 只能写 0。	

位	位名	初始值	R/W	说明
3	PINT3E	0	R/W	允许 PINT3 引脚的 GPIO 通道 0 的中断请求。
2	PINT2E	0	R/W	允许 PINT2 引脚的 GPIO 通道 0 的中断请求。
1	PINT1E	0	R/W	允许 PINT1 引脚的 GPIO 通道 0 的中断请求。
0	PINT0E	0	R/W	允许 PINT0 引脚的 GPIO 通道 0 的中断请求。

在将 GPIO 端口用作中断输入引脚时，如果检测到中断，GPIO 就通知 INTC。但是，INTC 的中断源寄存器 INT2A0 或者 INT2A1 中的 1 位表示中断源。此时，能通过参照各模块的中断源寄存器 INT2B7，确定是哪个通道的哪个引脚产生的中断。通道的确定也能参照 CPU 的 INTEVT（异常码）。

13.4 中断源

中断源分为 NMI、IRQ 和内部外围模块 3 类。用中断优先级的值 (16 ~ 0) 表示各中断的优先顺序, 级 16 为最高, 级 1 为最低。如果设定为级 0, 就屏蔽该中断, 忽视中断请求。

13.4.1 NMI 中断

NMI 中断的级 16 为最优先中断。只要 CPU 内 SR 的 BL 位不被置 1 就能随时接受。在睡眠模式中, 即使 BL 位为 1 也能接受。

另外, 通过设定, 即使 BL 位为 1 也能接受 NMI。

能通过边沿检测到 NMI 引脚的输入。通过设定 ICR0 的 NMI 边沿选择位 (NMIE), 能选择上升沿或者下降沿作为检测边沿。如果改写 ICR0 的 NMIE 位, 就在改写后的最大 6 个总线时钟周期, 不检测 NMI 中断。

SR 的中断屏蔽级 (IMASK) 不受 NMI 中断异常处理的影响。

13.4.2 IRQ 中断

通过设定 ICR1 的 IRQnS[1:0] (n=1, 0) 位, 能检测到 IRQ 中断的下降沿、上升沿、低电平或者高电平。另外, 能通过中断优先顺序设定寄存器 (INTPRI) 设定中断优先级。

在通过低电平或者高电平检测 IRQ 中断请求时, 从接受中断到开始中断处理为止, 必须保持 IRQ 中断的引脚状态。

但是在检测到 IRQ 中断请求后, 即使在 CPU 接受之前更改 IRQ 中断的引脚状态来取消请求, INTREQ 也保持中断源。在 CPU 接受某个中断 (不限于 IRQ 中断) 或者将相应中断屏蔽位置 1 前保持中断源。在通过中断处理程序清除 IRQ 中断源后, 必须将 INTREQ 保持的中断源清 0。有关清除方法的详细内容, 请参照“13.7.3 IRQ 中断请求的清除方法”。

如果 CPUOPM.INTMU 位已被置 1, SR 的中断屏蔽级 (SR.IMASK) 就被自动设定为接受的中断级; 如果 CPUOPM.INTMU 位已被置 0, SR.IMASK 就不受中断接受的影响。

13.4.3 内部外围模块的中断

内部模块中断是由内部外围模块产生的中断。

虽然各中断源未分配不同的中断向量, 但是因为中断源反映在中断事件寄存器中, 所以在异常处理程序中将 INTEVT 的值作为偏移量进行转移, 能简单地判断中断源。

通过 INT2PRI0 ~ INT2PRI12, 能在优先级 31 ~ 0 范围内设定各模块的优先级。但是, 在通知 CPU 时, 转换为 4 位 (舍去最低位)。详细内容请参照“13.4.4 内部外围模块的中断优先顺序”。

SR 的中断屏蔽位 (IMASK) 不受内部外围模块中断处理的影响。

必须在 SR 的 BL 位为 1 时更改内部外围模块的中断源标志和中断允许标志。为了避免误接受应该被更改的中断源, 必须在读取含该标志的内部外围模块的寄存器后将 BL 位清 0。从而能确保内部所需的时序。在更新多个标志时, 能在更新最后的标志后读取含该标志的寄存器。

如果在 BL 位为 0 时更改标志, 就可能在 INTEVT 的值为 0 时转移到中断处理程序。这是由于标志的更新时序和本 LSI 内部的中断请求识别时序的关系而启动了中断处理。此时, 通过执行 RTE 指令, 能继续进行处理。

13.4.4 内部外围模块的中断优先顺序

在生成内部外围模块中断时，将作为中断源固有识别码的中断异常码输出到 CPU。当 CPU 接受中断时，INTEVT 寄存器就表示该异常码。只要读 CPU 的 INTEVT 寄存器，就不读 INTC 的源表示寄存器，中断处理程序也能知道中断源。有关内部外围模块的中断源和异常码的对应，请参照表 13.1。

如图 13.2 所示，内部外围模块中断能用 5 位将各中断源设定为 30 级（值越大优先顺序越高，H'00 和 H'01 为屏蔽中断请求）。CPU 的中断级接收接口为 4 位 15 级（H'0 为屏蔽中断请求），但是内部外围模块中断的优先顺序扩展了 1 位，用 5 位优先选择各中断源，在选择后转换为 4 位（舍去最低位）并通知 CPU。例如，设定为优先级 H'1A 和优先级 H'1B 的 2 个中断源输出的 4 位优先级都为 H'D，虽然在两个中断发生竞争时，如果通知的异常码为 5 位，优先顺序高的 H'1B 中断所对应的异常码就被优先。如果两个优先级设定为同一值的中断发生竞争，就将表 13.1 所示的优先顺序的异常码通知 CPU。

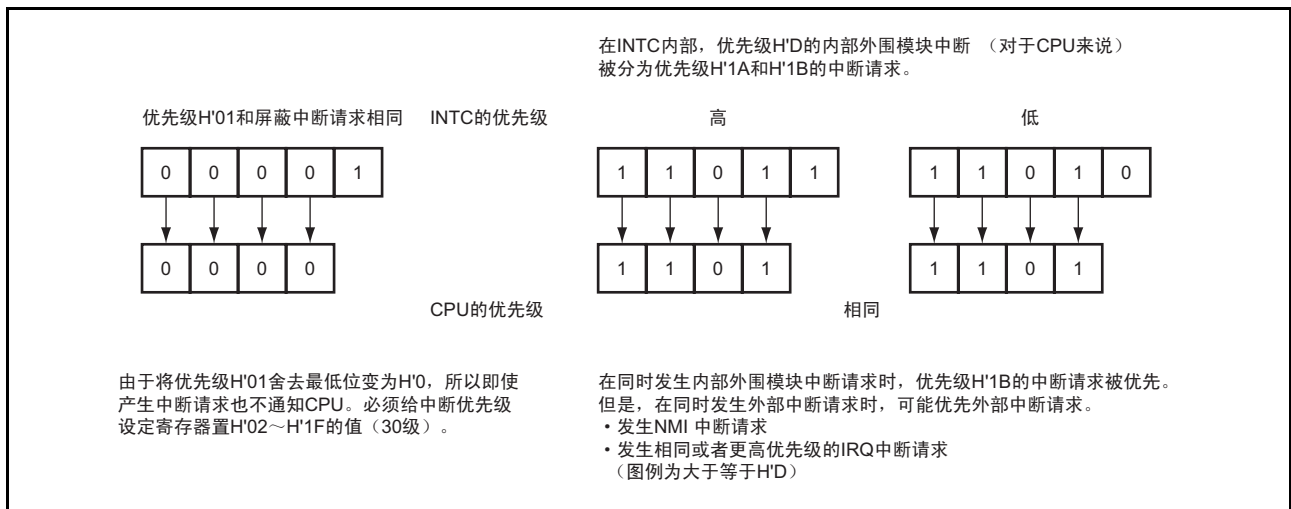


图 13.2 内部外围模块的中断优先级

13.4.5 中断异常处理和优先顺序

中断源、中断事件寄存器（INTEVT）的异常码和中断优先顺序如表 13.6 所示。

给各中断源分别分配了不同的 INTEVT 异常码，各中断源共用异常处理程序的起始地址。为了识别中断源，在异常处理程序的起始处使用 INTEVT 的值进行转移。例如，将 INTEVT 的值加上偏移量进行转移。

通过 INT2PRI0 ~ INT2PRI12，能在优先级 31 ~ 0 的范围内任意设定内部模块的优先顺序。通过复位将内部模块的优先顺序设定为优先级 0。

在将多个中断源的优先顺序设定为相同级时，如果这些中断同时发生，就根据表 13.6 所示的默认优先顺序进行处理。

必须在 SR 的 BL 位为 1 时更改 INTPRI、INT2PRI0 ~ INT2PRI12。为了避免误接受中断，必须在读取任意一个中断优先级设定寄存器后将 BL 位清 0。从而能确保内部所需的时序。

表 13.6 中断异常处理源和优先顺序

中断源		INTEVT (异常码)	中断优先顺序	INTC 的屏蔽 / 清除寄存器	中断源寄存器	各模块中断源寄存器	中断源内优先顺序	默认优先顺序
NMI	—	H'1C0	16	—	—	—	—	高
IRQ	IRQ0	H'240	INTPRI[31:28]	INTMSK[31] INTMSKCLR[31]	INTREQ[31]	—	高	↑ ↓
	IRQ1	H'280	INTPRI[27:24]	INTMSK[30] INTMSKCLR[30]	INTREQ[30]	—	低	
WDT	ITI*	H'560	INT2PRI2[12:8]	INT2MSKR[5] INT2MSKCR[5]	INT2A0[5] INT2A1[5]	—		
TMU0	TUNIO*	H'580	INT2PRI0[28:24]	INT2MSKR[0] INT2MSKCR[0]	INT2A0[0]	INT2B0[0]		
TMU1	TUNI1*	H'5A0	INT2PRI0[20:16]		INT2A1[0]	INT2B0[1]		
TMU2	TUNI2*	H'5C0	INT2PRI0[12:8]		INT2B0[2]			
	TICPI2*	H'5E0	INT2PRI0[4:0]		INT2B0[3]			
H-UDI	H-UDI	H'600	INT2PRI3[28:24]	INT2MSKR[7] INT2MSKCR[7]	INT2A0[7] INT2A1[7]	—		
LCDC	LCDC1	H'620	INT2PRI9[28:24]	INT2MSKR1[7] INT2MSKCR1[7]	INT2A01[7] INT2A11[7]	—		
DMAC	DMINT0*	H'640	INT2PRI3[20:16]	INT2MSKR[8] INT2MSKCR[8]	INT2A0[8] INT2A1[8]	INT2B3[0]	高	↑ ↓
	DMINT1*	H'660				INT2B3[1]		
	DMINT2*	H'680				INT2B3[2]		
	DMINT3*	H'6A0				INT2B3[3]	低	
	DMAE*	H'6C0				INT2B3[12]		
SCIF0	ERIO*	H'700	INT2PRI2[28:24]	INT2MSKR[3] INT2MSKCR[3]	INT2A0[3] INT2A1[3]	INT2B2[0]	高	↑ ↓
	RXIO*	H'720				INT2B2[1]		
	BRI0*	H'740				INT2B2[2]		
	TXIO*	H'760				INT2B2[3]	低	
DMAC	DMINT4*	H'780	INT2PRI3[20:16]	INT2MSKR[8] INT2MSKCR[8]	INT2A0[8] INT2A1[8]	INT2B3[4]	高	
	DMINT5*	H'7A0				INT2B3[5]	低	
VDC2	VDCI	H'860	INT2PRI12[28:24]	INT2MSKR1[19] INT2MSKCR1[19]	INT2A01[19] INT2A11[19]	—		
IIC	IICI	H'8A0	INT2PRI9[4:0]	INT2MSKR1[4] INT2MSKCR1[4]	INT2A01[4] INT2A11[4]	—		
EtherC	EINT	H'920	INT2PRI12[4:0]	INT2MSKR1[16] INT2MSKCR1[16]	INT2A01[16] INT2A11[16]	—		
G2D	G2DI	H'980	INT2PRI4[20:16]	INT2MSKR[13] INT2MSKCR[13]	INT2A0[13] INT2A1[13]	—		
SSI_A	SSIDMA0	H'A00	INT2PRI4[12:8]	INT2MSKR[14] INT2MSKCR[14]	INT2A0[14] INT2A1[14]	INT2B4[0]		↓
	SSICH0	H'A20	INT2PRI4[4:0]	INT2MSKR[15] INT2MSKCR[15]	INT2A0[15] INT2A1[15]	INT2B4[1]		
	SSICH1	H'A40	INT2PRI5[28:24]	INT2MSKR[16] INT2MSKCR[16]	INT2A0[16] INT2A1[16]	INT2B4[2]		
	SSICH2	H'A60	INT2PRI5[20:16]	INT2MSKR[17] INT2MSKCR[17]	INT2A0[17] INT2A1[17]	INT2B4[3]		

中断源		INTEVT (异常码)	中断优先顺序	INTC 的屏蔽 / 清除寄存器	中断源寄存器	各模块中断源寄存器	中断源内优先顺序	默认优先顺序
SSI_B	SSIDMA1	H'AA0	INT2PRI5[4:0]	INT2MSKR[19] INT2MSKCR[19]	INT2A0[19] INT2A1[19]	INT2B4[5]		高 ↑ ↓ 低
	SSICH3	H'AC0				INT2B4[6]		
	SSICH4	H'AE0				INT2B4[7]		
	SSICH5	H'B00				INT2B4[8]		
SCIF1	ERI1*	H'B80	INT2PRI2[20:16]	INT2MSKR[4] INT2MSKCR[4]	INT2A0[4] INT2A1[4]	INT2B2[4]	高 ↑ ↓ 低	
	RXI1*	H'BA0				INT2B2[5]		
	BRI1*	H'BC0				INT2B2[6]		
	TXI1*	H'BE0				INT2B2[7]		
ATAPI	ATAI	H'C00	INT2PRI6[28:24]	INT2MSKR[14] INT2MSKCR[14]	INT2A0[14] INT2A1[14]	—		
USB	USBI	H'C60	INT2PRI12[12:8]	INT2MSKR1[17] INT2MSKCR1[17]	INT2A01[17] INT2A11[17]	—		
FLCTL	FLSTE	H'D00	INT2PRI6[12:8]	INT2MSKR[22] INT2MSKCR[22]	INT2A0[22] INT2A1[22]	INT2B5[0]	高 ↑ ↓ 低	
	FLTEND	H'D20				INT2B5[1]		
	FLTRQ0	H'D40				INT2B5[2]		
	FLTRQ1	H'D60				INT2B5[3]		
TMU3	TUNI3*	H'E00	INT2PRI1[28:24]	INT2MSKR[1] INT2MSKCR[1]	INT2A0[1] INT2A1[1]	INT2B0[4]		
TMU4	TUNI4*	H'E20	INT2PRI1[20:16]	INT2MSKR[1] INT2MSKCR[1]	INT2A0[1] INT2A1[1]	INT2B0[5]		
TMU5	TUNI5*	H'E40	INT2PRI1[12:8]			INT2B0[6]		
SRC	OVF	H'E80	INT2PRI6[4:0]	INT2MSKR[23] INT2MSKCR[23]	INT2A0[23] INT2A1[23]	—	高 ↑ ↓ 低	
	IDEI	H'EA0	INT2PRI8[12:8]	INT2MSKR1[1] INT2MSKCR1[1]	INT2A01[1] INT2A11[1]	—		
	ODFI	H'EC0	INT2PRI8[20:16]	INT2MSKR1[2] INT2MSKCR1[2]	INT2A01[2] INT2A11[2]	—		
SCIF2	ERI2	H'F00	INT2PRI7[28:24]	INT2MSKR1[25] INT2MSKCR1[25]	INT2A01[25] INT2A11[25]	INT2B6[0]	高 ↑ ↓ 低	
	RXI2	H'F20				INT2B6[1]		
	BRI2	H'F40				INT2B6[2]		
	TXI2	H'F60				INT2B6[3]		
GPIO	CH0	H'F80	INT2PRI7[20:16]	INT2MSKR[25] INT2MSKCR[25]	INT2A0[25] INT2A1[25]	INT2B7[3:0]	高 ↑ ↓ 低	
	CH1	H'FA0				INT2B7[11:8]		
	CH2	H'FC0				INT2B7[19:16]		
	CH3	H'FE0				INT2B7[27:24]		

【符号说明】

- ITI : WDT 的间隔定时器中断
 TUNI0 ~ TUNI5 : TMU 通道 0 ~ 5 的下溢中断
 TICPI2 : TMU 通道 2 的输入捕捉中断
 DMINT0 ~ DMINT5 : DMAC 通道 0 ~ 5 的传送结束中断
 DMAE : DMAC 的地址错误中断 (通道 0 ~ 5 共用)
 ERI0、ERI1、ERI2 : SCIF 通道 0、1、2 的接收错误中断
 RXI0、RXI1、RXI2 : SCIF 通道 0、1、2 的接收数据满中断
 BRI0、BRI1、BRI2 : SCIF 通道 0、1、2 的中止中断请求
 TXI0、TXI1、TXI2 : SCIF 通道 0、1、2 的发送数据空中断

13.5 运行说明

13.5.1 中断的运行流程

以下说明发生中断时的运行流程，中断运行流程如图 13.3 所示。

1. 将各中断请求源的中断请求信号发送到中断控制器。
2. 中断控制器根据INTPRI、INT2PRI0~INT2PRI12，从送来的中断请求中选择优先顺序最高的中断，并且保留优先顺序低的中断。此时，如果有被设定为同一优先顺序的中断或者在同一模块中发生多个中断，就根据表 13.6 选择优先顺序最高的中断。
3. 将中断控制器选择的中断优先级和CPU的SR中断屏蔽级 (IMASK) 进行比较。接受高于IMASK位的中断，并且将中断请求信号发送到CPU。
4. CPU在指令结束时接受中断。
5. 将异常码 (中断源代码) 设定到中断事件寄存器 (INTEVT)。
6. 将状态寄存器 (SR) 和程序计数器 (PC) 分别保存到SSR和SPC，此时的R15保存到SGR。
7. 将SR的块位 (BL)、模式位 (MD)、寄存器存储体位 (RB) 置1。
8. 转移到中断相关的异常处理程序的起始地址 (向量基址寄存器 (VBR) 设定的值加上H'0000 0600)。

在异常处理程序中，为了识别中断源，将INTEVT寄存器的值作为偏移量进行转移。从而能简单地转移到各中断源的处理程序。

- 【注】**
1. 如果CPU运行模式寄存器的INTMU位 (CPUOPM.INTMU) 已被置1，SR的中断屏蔽级 (SR.IMASK) 就被自动设定为接受的中断级；如果CPUOPM.INTMU位已被置0，SR.IMASK就不受中断接受的影响。
 2. 必须在中断处理程序中清除中断源标志。为了避免误接受应该被清除的中断源，必须清除后读取源标志，然后在等待如表 13.7 所示的时间后清除BL位或者执行RTE指令。
 3. 在上电复位时，IRQ中断和内部模块中断被初始化为中断屏蔽状态。需要使用INTMSKCLR、INT2MSKCR和INT2MSKCR1，清除各源的中断屏蔽 (INTMSK、INT2MSKR、INT2MSKR1)。

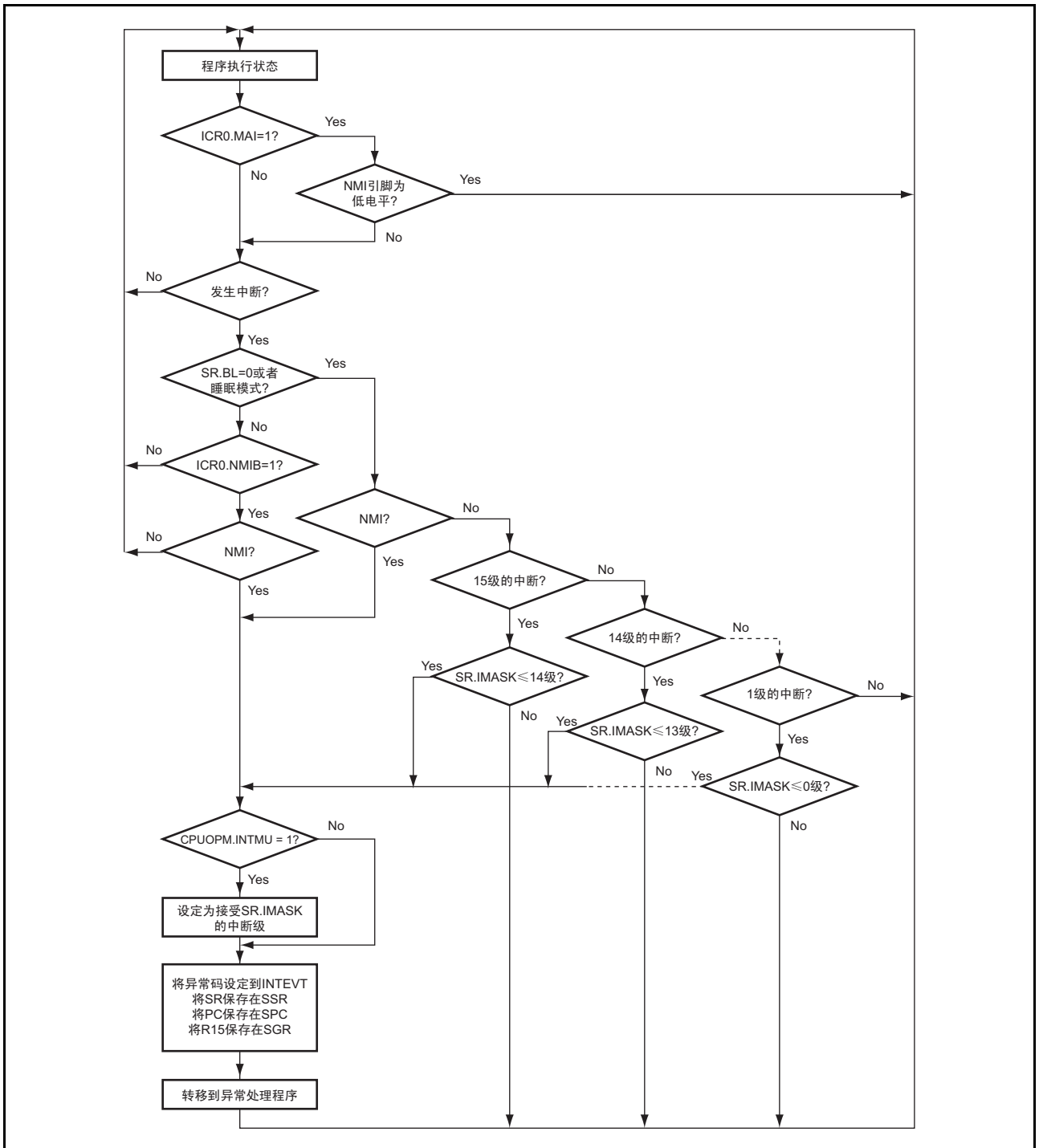


图 13.3 中断运行流程

13.5.2 多重中断

在处理多重中断时，中断处理程序的结构如下所示：

1. 为了判断中断源，将INTEVT的值作为偏移量，转移到各中断源的中断处理程序。
2. 在各中断处理程序中清除相应的中断源。
3. 将SPC和SSR压栈。
4. 清除SR的BL位。当CPUOPM.INTMU=1时，SR寄存器的中断屏蔽级（IMASK）就被自动设定为接受的中断级。当CPUOPM.INTMU=0时，能通过软件将SR的中断屏蔽级（IMASK）设定为接受的中断级。
5. 实际是要进行的处理。
6. 将SR的BL位置1。
7. 从存储器恢复SSR和SPC。
8. 执行RTE指令。

在使用上述中断处理程序结构的情况下，如果在 4. 之后发生多重中断，就能接受优先级更高的中断。因此对于紧急度高的处理，能缩短中断响应时间。

13.5.3 通过 MAI 位进行中断屏蔽

在 NMI 引脚为低电平期间，与 SR 寄存器的 BL 和 IMASK 位无关，能通过将 ICRO 的 MAI 位置 1 进行中断屏蔽。

- 在正常运行和睡眠模式时在 NMI 引脚为低电平期间，屏蔽全部中断。但是，只发生因 NMI 引脚的变化引起的 NMI 中断。

13.6 中断响应时间

从发生中断请求到进行中断异常处理以及开始取异常处理程序起始指令为止的时间（中断响应时间）如表 13.7 所示。

表 13.7 中断响应时间

项目	状态数				备注	
	NMI	IRQ	外围模块			
			非 GPIO	GPIO		
优先顺序判断时间	5Bcyc+2Pcyc	4Bcyc+2Pcyc	5Pcyc	7Pcyc		
CPU 结束执行中的顺序前的等待时间	S-1 (≥ 0) \times lcyc					
从中断异常处理（保存 SR 和 PC）到发行取异常处理程序起始指令的 SHwy 总线请求为止的时间	11lcyc+1Scyc					
响应时间	合计	(S+10) lcyc +1Scyc +5Bcyc +2Pcyc	(S+10) lcyc +1Scyc +4Bcyc +2Pcyc	(S+10) lcyc +1Scyc +5Pcyc	(S+10) lcyc +1Scyc +7Pcyc	
	最短时间	29lcyc +S \times lcyc	35lcyc +S \times lcyc	31lcyc +S \times lcyc	39lcyc +S \times lcyc	当 lcyc:Scyc: Bcyc:Pcyc =4:4:2:1 时

【符号说明】

- lcyc : 1个CPU时钟 (lck) 周期
- Scyc: 1个SHwy时钟 (SHck) 周期
- Bcyc: 1个总线时钟 (Bck) 周期
- Pcyc: 1个外围时钟 (Pck) 周期
- S : 指令执行状态数

13.7 使用时的注意事项

13.7.1 中断和电平检测设定时的 IRQ 中断处理程序例子

由于 CPU 在接受中断后还检测 IRQ 中断请求（设定为电平检测时），所以需要通过中断处理程序清除 INTC 保持的中断请求。

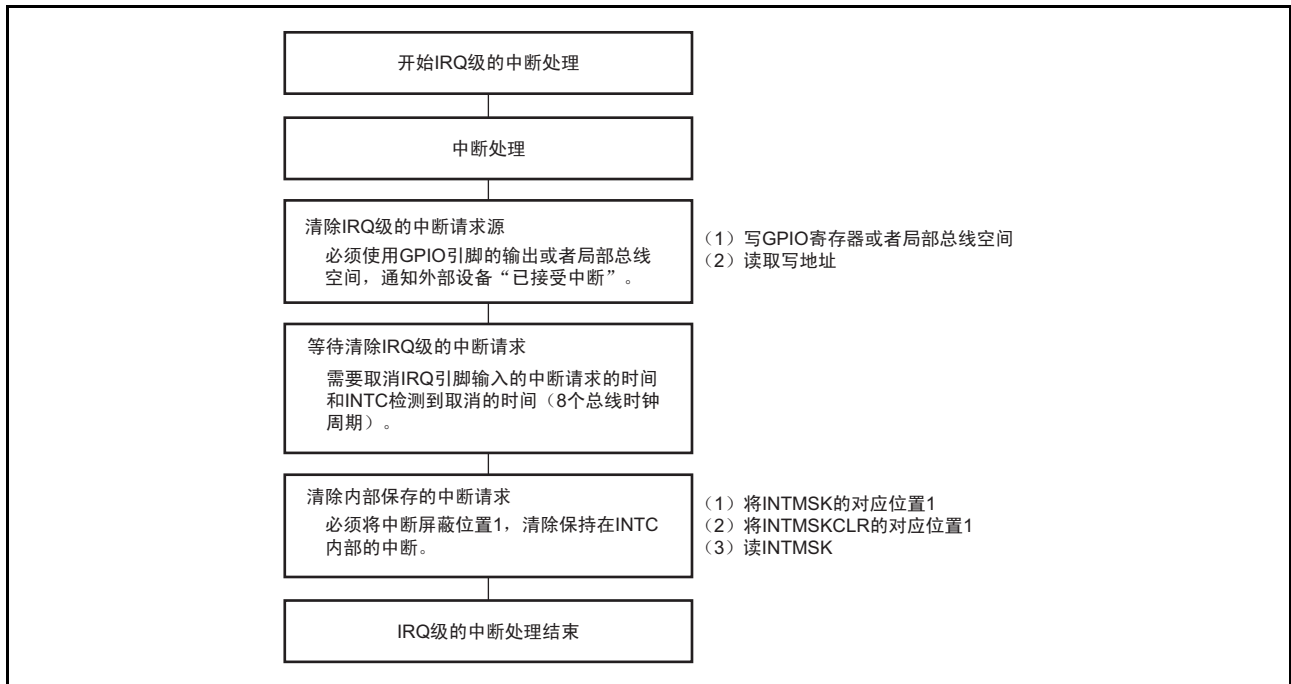


图 13.4 中断处理程序例子

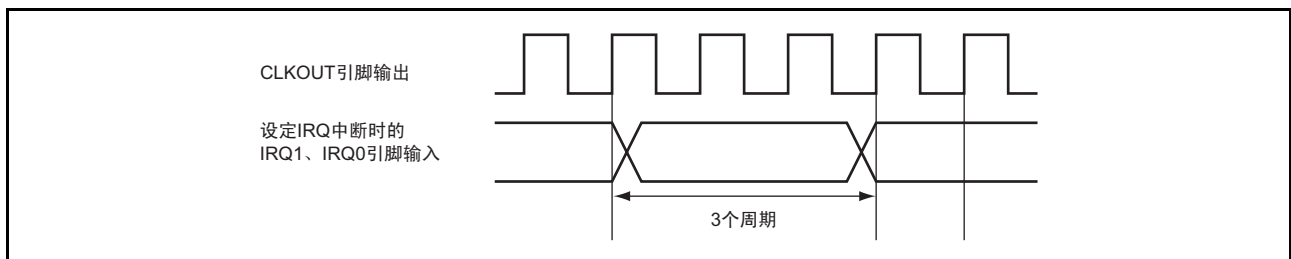


图 13.5 检测 IRQ 引脚输入的中断请求所需的时间

13.7.2 设定 IRQ 引脚功能的注意事项

在切换 IRQ1 和 IRQ0 引脚功能时，有可能会误检测到中断请求，并且将其保持在 INTC 内部。因此，必须在将 IRQ 中断请求设定为屏蔽状态后切换 IRQ1 和 IRQ0 引脚功能。

表 13.8 IRQ 引脚功能的切换步骤

步骤	项目	内容
1	屏蔽 IRQ 中断请求。	给 INTMSK 的所有位写 1。
2	将 $\overline{\text{IRQ0/DTEND1}}$ 、 $\overline{\text{WDT0VF/IRQ1/AUDUCK/DACK1}}$ 引脚设定为 IRQ0 和 IRQ1 功能。	IRQ0: 给 GPIO 的 PTSEL_S 寄存器的 PTSEL_S15 位写 0 IRQ1: 给 GPIO 的 PTSEL_K 寄存器的 PTSEL_K7[1:0] 位写 01
3	开始检测 IRQ 中断请求。	给 INTMSKCLR 的相应位写 1

13.7.3 IRQ 中断请求的清除方法

必须按以下步骤清除 INTC 保持的中断请求：

(1) 设定为电平检测时的 IRQ 中断请求的清除方法

必须给 INTMSK 的对应位 IM01 ~ IM00 写 1，清除被设定为电平检测的 IRQ1 和 IRQ0 的中断请求。即使给 INTPRI 的对应位写 0，也不能清除检测到的 IRQ 中断请求。能通过读 INTREQ 确认检测到的 IRQ 中断请求（CPU 一旦接受中断就清除）。

(2) 设定为边沿检测时的 IRQ 中断请求的清除方法

必须在从 INTREQ 的对应位 IR1 和 IR0 读到 1 后给其写 0，清除被设定为边沿检测的 IRQ1 和 IRQ0 的中断请求。即使给 INTMSK 的对应位写 1，也不能清除检测到的 IRQ 中断请求。

第 14 章 定时器单元 (TMU)

本 LSI 内置由 6 个通道 (通道 0 ~ 5) 32 位定时器构成的 32 位定时器单元 (TMU: Timer Unit)。

14.1 特点

- TMU 有以下特点: 各通道都装载了自动重加载方式的 32 位递减计数器。
- 通道 2 装载了输入捕捉功能。
- 在选择外部时钟或者使用通道 2 输入捕捉功能时, 通道 0 ~ 2 能选择上升沿或者下降沿作为外部时钟的输入边沿。
- 各通道都装载了 32 位定时器常数寄存器和 32 位递减计数器 (用于在任意时间都可读写的自动重加载)。
- 通道 0 ~ 2 能选择 6 种计数器的输入时钟。
外部时钟 (TCLK)、5 种外围时钟 (Pck/4、Pck/16、Pck/64、Pck/256、Pck/1024) (Pck 为外围时钟)
- 通道 3 ~ 5 能选择 5 种计数器的输入时钟。
5 种外围时钟 (Pck/4、Pck/16、Pck/64、Pck/256、Pck/1024)
- 2 种中断源
下溢 ×1 个源 (各通道) 和输入捕捉 ×1 个源 (通道 2)。

TMU 的框图如图 14.1 所示。

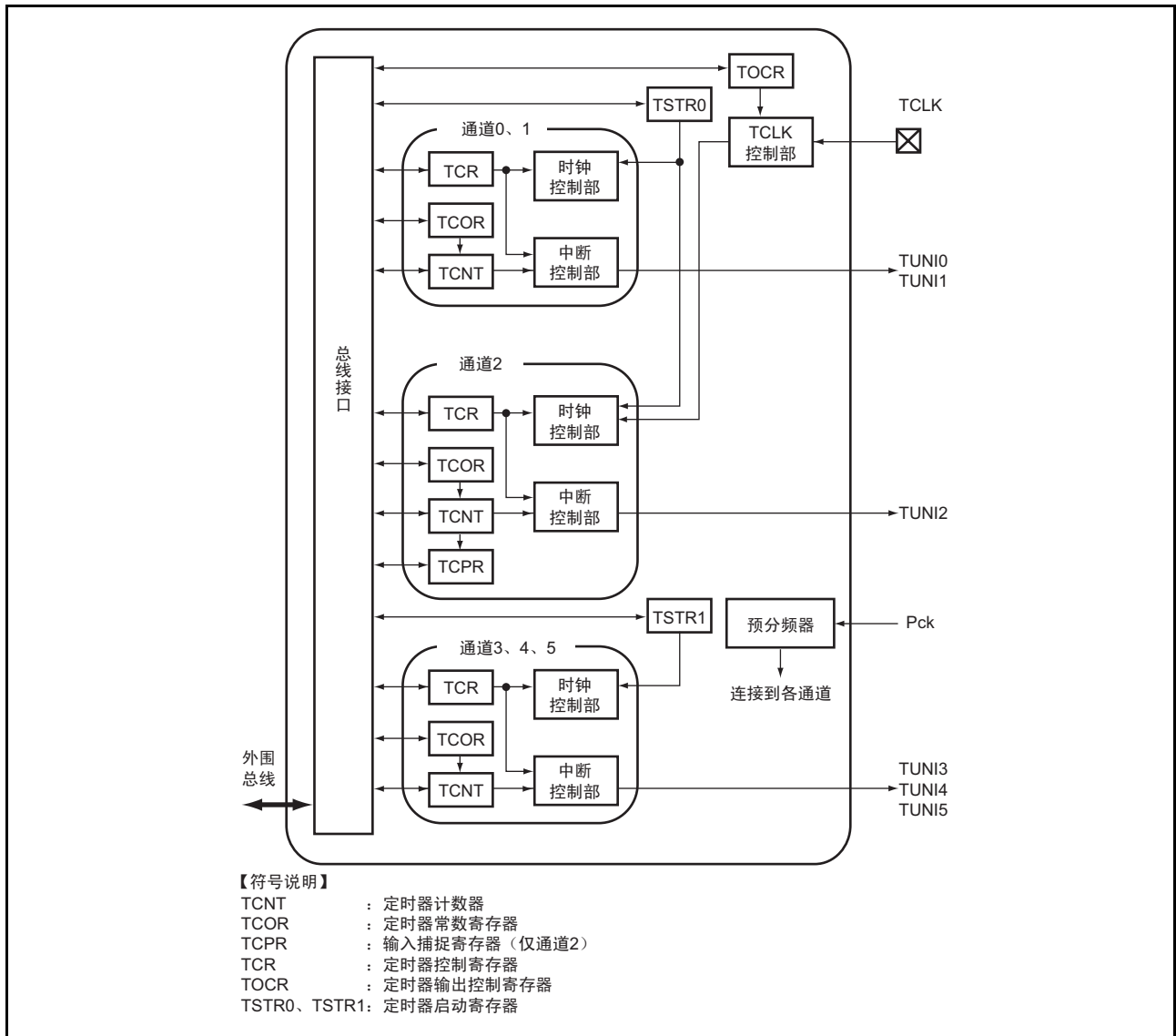


图 14.1 TMU 的框图

14.2 输入 / 输出引脚

TMU 的引脚结构如表 14.1 所示。

表 14.1 引脚结构

引脚名	功能	输入 / 输出	说明
TCLK	时钟输入	输入	是通道 0、1、2 外部时钟的输入引脚 / 通道 2 输入捕捉控制的输入引脚

14.3 寄存器说明

TMU 的寄存器结构如表 14.2 所示，各处理模式的寄存器状态如表 14.3 所示。

表 14.2 寄存器结构

通道	名称	略称	R/W	P4 区地址	区域 7 地址	存取长度
012 共用	定时器输出控制寄存器	TOCR	R/W	H'FFD8 0000	H'1FD8 0000	8
012 共用	定时器启动寄存器 0	TSTR0	R/W	H'FFD8 0004	H'1FD8 0004	8
0	定时器常数寄存器 0	TCOR0	R/W	H'FFD8 0008	H'1FD8 0008	32
	定时器计数器 0	TCNT0	R/W	H'FFD8 000C	H'1FD8 000C	32
	定时器控制寄存器 0	TCR0	R/W	H'FFD8 0010	H'1FD8 0010	16
1	定时器常数寄存器 1	TCOR1	R/W	H'FFD8 0014	H'1FD8 0014	32
	定时器计数器 1	TCNT1	R/W	H'FFD8 0018	H'1FD8 0018	32
	定时器控制寄存器 1	TCR1	R/W	H'FFD8 001C	H'1FD8 001C	16
2	定时器常数寄存器 2	TCOR2	R/W	H'FFD8 0020	H'1FD8 0020	32
	定时器计数器 2	TCNT2	R/W	H'FFD8 0024	H'1FD8 0024	32
	定时器控制寄存器 2	TCR2	R/W	H'FFD8 0028	H'1FD8 0028	16
	输入捕捉寄存器 2	TCPR2	R	H'FFD8 002C	H'1FD8 002C	32
345 共用	定时器启动寄存器 1	TSTR1	R/W	H'FFDC 0004	H'1FDC 0004	8
3	定时器常数寄存器 3	TCOR3	R/W	H'FFDC 0008	H'1FDC 0008	32
	定时器计数器 3	TCNT3	R/W	H'FFDC 000C	H'1FDC 000C	32
	定时器控制寄存器 3	TCR3	R/W	H'FFDC 0010	H'1FDC 0010	16
4	定时器常数寄存器 4	TCOR4	R/W	H'FFDC 0014	H'1FDC 0014	32
	定时器计数器 4	TCNT4	R/W	H'FFDC 0018	H'1FDC 0018	32
	定时器控制寄存器 4	TCR4	R/W	H'FFDC 001C	H'1FDC 001C	16
5	定时器常数寄存器 5	TCOR5	R/W	H'FFDC 0020	H'1FDC 0020	32
	定时器计数器 5	TCNT5	R/W	H'FFDC 0024	H'1FDC 0024	32
	定时器控制寄存器 5	TCR5	R/W	H'FFDC 0028	H'1FDC 0028	16

表 14.3 各处理模式的寄存器状态

通道	名称	略称	上电复位	睡眠	待机	模块待机
012 共用	定时器输出控制寄存器	TOCR	H'00	保持	保持	保持
012 共用	定时器启动寄存器 0	TSTR0	H'00	保持	保持	保持
0	定时器常数寄存器 0	TCOR0	H'FFFF FFFF	保持	保持	保持
	定时器计数器 0	TCNT0	H'FFFF FFFF	保持	保持	保持
	定时器控制寄存器 0	TCR0	H'0000	保持	保持	保持
1	定时器常数寄存器 1	TCOR1	H'FFFF FFFF	保持	保持	保持
	定时器计数器 1	TCNT1	H'FFFF FFFF	保持	保持	保持
	定时器控制寄存器 1	TCR1	H'0000	保持	保持	保持
2	定时器常数寄存器 2	TCOR2	H'FFFF FFFF	保持	保持	保持
	定时器计数器 2	TCNT2	H'FFFF FFFF	保持	保持	保持
	定时器控制寄存器 2	TCR2	H'0000	保持	保持	保持
	输入捕捉寄存器 2	TCPR2	H'xxxx xxxx	保持	保持	保持
345 共用	定时器启动寄存器 1	TSTR1	H'00	保持	保持	保持
3	定时器常数寄存器 3	TCOR3	H'FFFF FFFF	保持	保持	保持
	定时器计数器 3	TCNT3	H'FFFF FFFF	保持	保持	保持
	定时器控制寄存器 3	TCR3	H'0000	保持	保持	保持
4	定时器常数寄存器 4	TCOR4	H'FFFF FFFF	保持	保持	保持
	定时器计数器 4	TCNT4	H'FFFF FFFF	保持	保持	保持
	定时器控制寄存器 4	TCR4	H'0000	保持	保持	保持
5	定时器常数寄存器 5	TCOR5	H'FFFF FFFF	保持	保持	保持
	定时器计数器 5	TCNT5	H'FFFF FFFF	保持	保持	保持
	定时器控制寄存器 5	TCR5	H'0000	保持	保持	保持

14.3.1 定时器输出控制寄存器 (TOCR)

定时器输出控制寄存器 (TOCR) 是 8 位只读寄存器。

表示外部引脚 TCLK 是用于外部时钟或者输入捕捉控制的输入引脚。

位:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TCOE
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 ~ 1	—	全 0	R	保留位 读取值为 0, 只能写 0。
0	TCOE	0	R	定时器时钟引脚控制位 (TCOE) 表示定时器时钟引脚 (TCLK) 用于外部时钟或者输入捕捉控制的输入引脚。 0: 用于外部时钟或者输入捕捉控制的输入引脚 1: 无效

14.3.2 定时器启动寄存器 (TSTRn) (n=0、1)

TSTR 是 8 位可读写寄存器, 选择运行或者停止 TCNT。

- TSTR0

位:	7	6	5	4	3	2	1	0
	—	—	—	—	—	STR2	STR1	STR0
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 读取值为 0, 只能写 0。
2	STR2	0	R/W	计数器开始 2 选择运行或者停止 TCNT2。 0: 停止 TCNT2 计数 1: 进行 TCNT2 计数
1	STR1	0	R/W	计数器开始 1 选择运行或者停止 TCNT1。 0: 停止 TCNT1 计数 1: 进行 TCNT1 计数
0	STR0	0	R/W	计数器开始 0 选择运行或者停止 TCNT0。 0: 停止 TCNT0 计数 1: 进行 TCNT0 计数

• TSTR1

位:	7	6	5	4	3	2	1	0
	—	—	—	—	—	STR5	STR4	STR3
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 读取值为 0，只能写 0。
2	STR5	0	R/W	计数器开始 5 选择运行或者停止 TCNT5。 0: 停止 TCNT5 计数 1: 进行 TCNT5 计数
1	STR4	0	R/W	计数器开始 4 选择运行或者停止 TCNT4。 0: 停止 TCNT4 计数 1: 进行 TCNT4 计数
0	STR3	0	R/W	计数器开始 3 选择运行或者停止 TCNT3。 0: 停止 TCNT3 计数 1: 进行 TCNT3 计数

14.3.3 定时器常数寄存器 (TCORn) (n=0 ~ 5)

TCOR 是 32 位可读写寄存器。如果 TCNT 递减计数的结果发生下溢，就将此 TCOR 的值设定到 TCNT，TCNT 从设定的值继续进行递减计数。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
初始值:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	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

14.3.4 定时器计数器 (TCNTn) (n=0 ~ 5)

TCNT 是 32 位可读写寄存器，通过 TCR 的 TPSC[2:0] 位选择的输入时钟进行递减计数。

如果 TCNT 递减计数的结果发生下溢，就将对应通道的 TCR 的 UNF 置 1，同时给 TCNT 设定 TCOR 的值，TCNT 从设定的值继续进行递减计数。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
初始值:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	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

14.3.5 定时器控制寄存器 (TCRn) (n=0 ~ 5)

TCR 是 16 位可读写寄存器，进行计数时钟的选择、外部时钟的边沿选择以及控制 TCNT 下溢标志为 1 时的中断发生。通道 2 的 TCR 控制输入捕捉功能和输入捕捉时的中断发生。

- TCR0、TCR1、TCR3、TCR4、TCR5

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	UNF	—	—	UNIE	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

- TCR2

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ICPF	UNF	ICPE[1:0]	UNIE	CKEG[1:0]	TPSC[2:0]				
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 10	—	全 0	R	保留位 读取值为 0，只能写 0。
9	ICPF*1	0	R/W	输入捕捉中断标志位 只是通道 2 有此功能，是表示发生输入捕捉的状态标志。 0：表示未发生输入捕捉 [清除条件] 当给 ICPF 写 0 时 1：表示已发生输入捕捉 [置位条件] 当发生输入捕捉时 *2
8	UNF	0	R/W	下溢标志位 是表示 TCNT 发生下溢的状态标志。 0：表示 TCNT 未发生下溢 [清除条件] 当给 UNF 写 0 时 1：表示 TCNT 已发生下溢 [置位条件] 当 TCNT 发生下溢时 *2

位	位名	初始值	R/W	说明
7、6	ICPE[1:0]*1	00	R/W	<p>输入捕捉控制位</p> <p>只是通道 2 有此功能，控制是否使用输入捕捉功能以及在使用时是否允许发生中断。</p> <p>在 CKEG 位指定的 TCLK 引脚的上升沿或者下降沿，将 TCNT2 的值设定到 TCPR2。</p> <p>只在 TCR2 的 ICPF 位为 0 时才给 TCPR2 设定 TCNT2 的值。当 ICPF 位为 1 时，即使发生输入捕捉也不设定 TCPR2。</p> <p>00：表示不使用输入捕捉功能</p> <p>01：保留（禁止设定）</p> <p>10：表示使用输入捕捉功能，但是不允许通过输入捕捉产生中断（TICPI2）。</p> <p>11：表示使用输入捕捉功能，也允许通过输入捕捉产生中断（TICPI2）。</p>
5	UNIE	0	R/W	<p>下溢中断控制位</p> <p>在 TCNT 的下溢标志 UNF 为 1 时，控制是否允许发生中断。</p> <p>0：不允许通过下溢发生中断（TUNI）</p> <p>1：允许通过下溢发生中断（TUNI）</p>
4、3	CKEG[1:0]	00	R/W	<p>时钟边沿位</p> <p>在选择外部时钟或者使用输入捕捉功能时，选择外部时钟的输入边沿。</p> <p>00：在上升沿设定计数 / 输入捕捉寄存器</p> <p>01：在下降沿设定计数 / 输入捕捉寄存器</p> <p>1X：在上升 / 下降的双边沿设定计数 / 输入捕捉寄存器</p>
2 ~ 0	TPSC[2:0]	000	R/W	<p>定时器预分频器</p> <p>选择 TCNT 的计数时钟。</p> <p>000：通过 Pck/4 进行计数</p> <p>001：通过 Pck/16 进行计数</p> <p>010：通过 Pck/64 进行计数</p> <p>011：通过 Pck/256 进行计数</p> <p>100：通过 Pck/1024 进行计数</p> <p>101：禁止设定</p> <p>110：禁止设定</p> <p>111：用外部时钟（TCLK）计数 *3</p>

【注】 X: Don't care

*1 在通道 0、1、3、4、5 的情况下，为保留位（初始值为 0，只能读）。

*2 当写 1 时，保持原来的值。

*3 在通道 3、4、5 的情况下，禁止设定。

14.3.6 输入捕捉寄存器 2 (TCPR2)

TCPR2 是 32 位只读寄存器 (只用于通道 2 内置的输入捕捉功能)。通过 TCR2 的 ICPE 位和 CKEG 位进行输入捕捉功能的控制。如果发生输入捕捉, 就将 TCNT2 的值复制到 TCPR2。只在 TCR2 的 ICPF 位为 0 时才给 TCPR2 设定 TCR2 的值。

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

14.4 运行说明

各通道都有 32 位定时器计数器 (TCNT) 和 32 位定时器常数寄存器 (TCOR)。TCNT 进行递减计数, 能通过自动重加载功能进行周期计数或者外部事件计数。通道 2 有输入捕捉功能。

14.4.1 计数器运行

当将 TSTR1、TSTR0 的 STR5 ~ STR0 位置 1 时, 对应通道的 TCNT 就开始计数。当 TCNT 发生下溢时, 就将对应 TCR 的 UNF 标志置 1。此时, 如果 TCR 的 UNIE 位为 1, 就向 CPU 请求中断, 并且将 TCOR 的值复制到 TCNT, 继续进行递减计数 (自动重加载功能)。

(1) 计数的设定步骤例子

计数的设定步骤例子如图 14.2 所示。

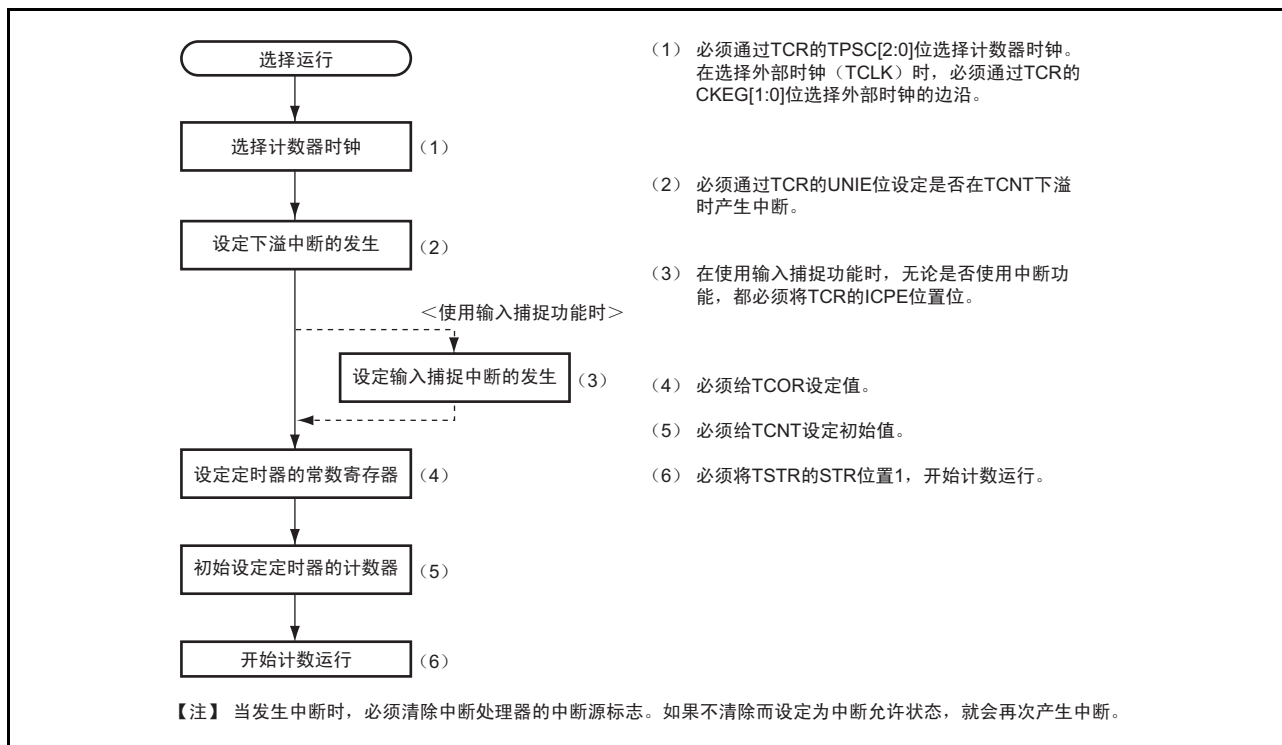


图 14.2 计数的设定步骤例子

(2) 自动重加载计数

TCNT 的自动重加载如图 14.3 所示。

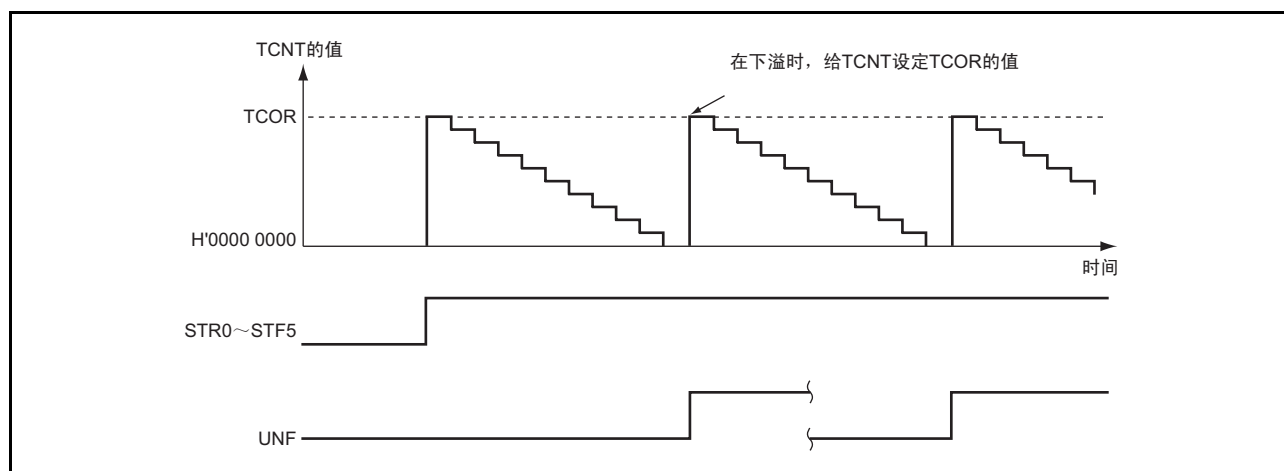


图 14.3 TCNT 的自动重加载

(3) TCNT 的计数时序

- 在内部时钟运行时
能通过TCR的TPSC[2:0]位选择外围时钟的5种分频时钟 (Pck/4、Pck/16、Pck/64、Pck/256、Pck/1024) 作为计数时钟。
此时的时序如图 14.4 所示。

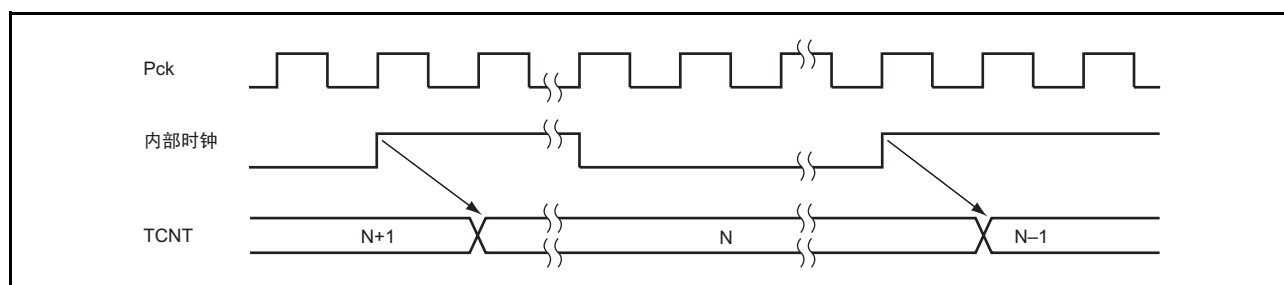


图 14.4 内部时钟运行时的计数时序

- 在外部时钟运行时
在通道0、1、2的情况下, 能通过TCR的TPSC[2:0]位选择外部时钟引脚 (TCLK) 的输入作为定时器时钟。另外, 能通过TCR的CKEG[1:0]位选择检测边沿, 外部时钟的检测能选择上升/下降/双边沿。
双边沿检测时的时序如图 14.5 所示。

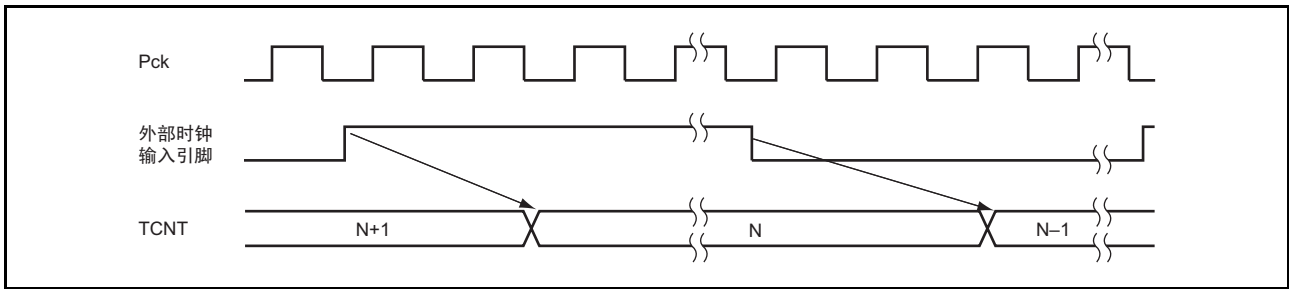


图 14.5 外部时钟运行时的计数时序

14.4.2 输入捕捉功能

通道 2 有输入捕捉功能。

在使用输入捕捉功能时：

1. 通过 TCR 的 TPSC[2:0] 位，将定时器的工作时钟设定为内部时钟。
2. 通过 TCR 的 ICPE[1:0] 位，指定使用输入捕捉功能以及在使用时是否产生中断。
3. 通过 TCR 的 CKEG[1:0] 位，指定在 TCLK 引脚的上升沿或者下降沿将 TCNT 的值设定到 TCPR2。

在发生输入捕捉的情况下，只在 TCR2 的 ICPF 位为 0 时才将 TCNT2 的值设定到 TCPR2。

使用输入捕捉功能时的运行时序如图 14.6 所示（使用 TCLK 的上升沿）。

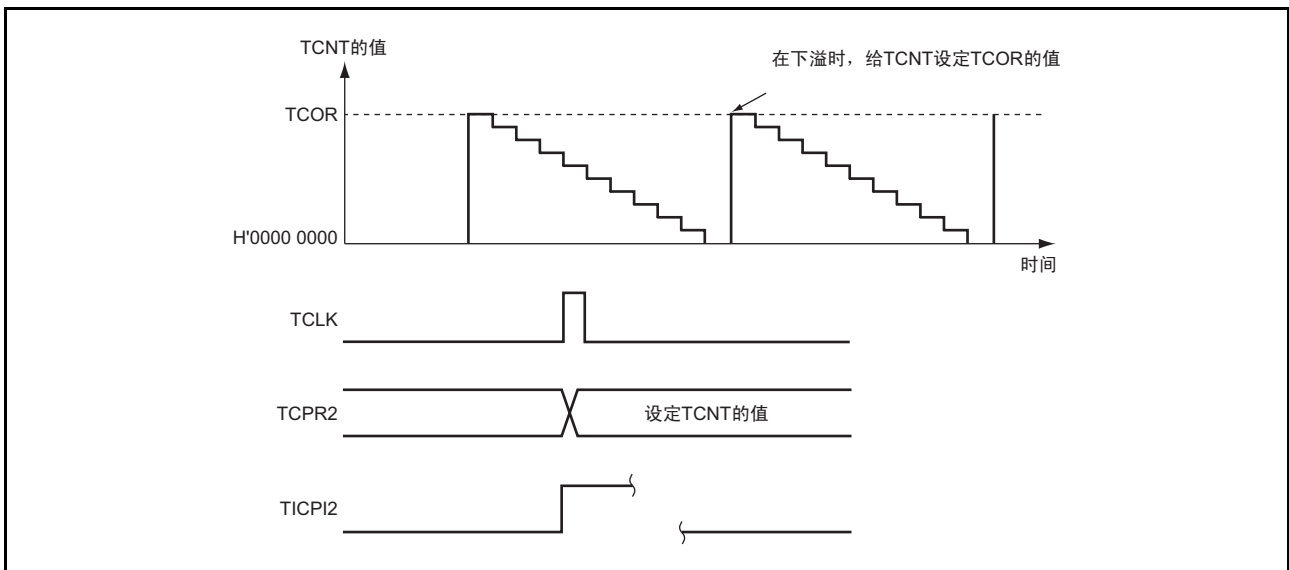


图 14.6 使用输入捕捉功能时的运行时序

14.5 中断

TMU 的中断源为下溢中断以及使用输入捕捉功能时的输入捕捉中断。在各通道产生下溢中断，而只在通道 2 产生输入捕捉中断。

在 UNF 位为 1 并且该通道的中断允许位为 1 时，产生下溢中断请求（每个通道）。

在使用输入捕捉功能时，如果发生输入捕捉请求并且 TCR2 的 ICPF 位为 1 以及 TCR2 的输入捕捉控制位 (ICPE[1:0]) 为 11，就产生中断请求。

TMU 的中断源如表 14.4 所示。

表 14.4 TMU 的中断源

通道	中断源	内容
0	TUNI0	下溢中断 0
1	TUNI1	下溢中断 1
2	TUNI2	下溢中断 2
	TICPI2	输入捕捉中断 2
3	TUNI3	下溢中断 3
4	TUNI4	下溢中断 4
5	TUNI5	下溢中断 5

14.6 使用时的注意事项

14.6.1 寄存器的写操作

在写 TMU 寄存器时，必须清除 TSTR 相应通道的起始位 (STR5 ~ STR0)，停止定时器的计数。

能在计数时写 TSTR 以及清除 TCR 的 UNF 和 ICPF 位。如果要在计数时清除标志 (UNF 和 ICPF)，就不能更改清除位以外的位。

14.6.2 TCNT 寄存器的读操作

TCNT 寄存器的读操作和定时器的计数进行同步处理。当同时进行定时器的计数和寄存器的读操作时，根据同步处理，读到的是 TCNT 计数器递减计数前的值。

14.6.3 外部时钟频率

输入到通道 0、1、2 的外部时钟 (TCLK) 频率不能超过 Pck/4。

第 15 章 FIFO 内置型串行通信接口 (SCIF)

本 LSI 具备支持异步通信和时钟同步通信 2 种方式的 3 个通道的 FIFO 内置型串行通信接口 (SCIF)。各通道都内置了分别用于发送和接收的 16 段 FIFO 寄存器, 能进行高效率和高速度的连续通信。

15.1 特点

- 异步串行通信
通过以字符为单位取得同步的异步方式进行串行数据通信, 能与 Universal Asynchronous Receiver/Transmitter (UART) 或者 Asynchronous Communication Interface Adapter (ACIA) 等标准异步通信的 LSI 进行串行数据通信, 能从 8 种格式中选择串行数据的通信格式。
数据长度 : 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
接收错误的检测: 检测奇偶校验错误、帧错误和溢出错误
中止的检测 : 当发生帧错误后并且至少有连续 1 帧的 0 (低电平) 时, 就检测到中止。另外, 当发生帧错误时, 也能通过从串行端口寄存器直接读取 RxD 引脚的电平来检测到中止。
- 时钟同步串行通信
与时钟同步进行串行数据通信, 能与具有时钟同步通信功能的其他 LSI 进行串行数据通信。串行数据通信格式只有 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, 所以能同时进行发送和接收。发送部和接收部都为 16 段 FIFO 缓冲器结构, 能进行串行数据的高速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。
- 内部或者外部的发送和接收时钟源
能选择波特率发生器 (内部时钟) 或者 SCK 引脚 (外部时钟)。
- 4 种中断源
有发送 FIFO 数据空中断、中止中断、接收 FIFO 数据满中断和接收错误中断 4 种中断源, 能分别独立请求中断。
- 在不使用 SCIF 时, 为了降低功耗, 能停止向 SCIF 提供时钟, 使 SCIF 停止运行。
- 在异步模式中, 内置调制解调器控制功能 (RTS 和 CTS)
- 能检测到发送和接收 FIFO 数据寄存器的数据个数以及接收 FIFO 数据寄存器的数据接收错误的个数。
- 在异步模式的接收时, 能检测到超时错误 (DR)。
- 在异步模式中, 能选择 16 或者 8 倍位速率的基本时钟频率。
- 在异步模式并且将内部时钟作为时钟源、SCK 引脚为输入引脚时, 波特率发生器能选择正常模式或者倍速模式
- 在异步模式中, 也能支持 3Mbps 以上的高速通信。

SCIF 的框图如图 15.1 所示。

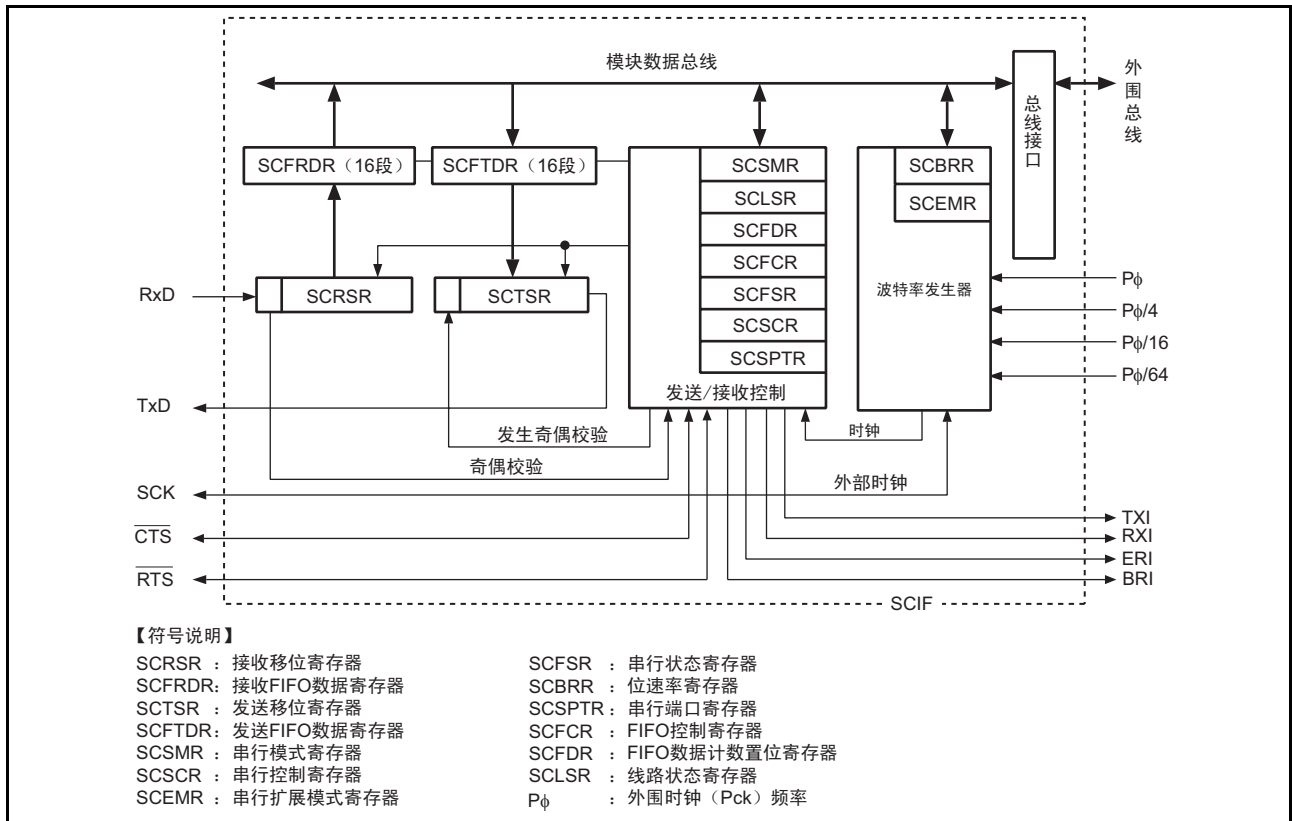


图 15.1 SCIF 的框图

15.2 输入 / 输出引脚

SCIF 的引脚结构如表 15.1 所示。

表 15.1 引脚结构

通道	名称	引脚名	输入 / 输出	功能
0 ~ 2	串行时钟引脚	SCK0 ~ SCK2	输入 / 输出	时钟的输入 / 输出
	接收数据引脚	RxD0 ~ RxD2	输入	接收数据的输入
	发送数据引脚	TxD0 ~ TxD2	输出	发送数据的输出
	请求发送引脚	$\overline{\text{RTS0}} \sim \overline{\text{RTS2}}$	输入 / 输出	请求发送
	清除发送引脚	$\overline{\text{CTS0}} \sim \overline{\text{CTS2}}$	输入 / 输出	清除发送

15.3 寄存器说明

SCIF 的寄存器结构如表 15.2 所示, 各处理模式的寄存器状态如表 15.3 所示。因为各通道的寄存器结构都相同, 所以在本文中省略了通道的说明。

表 15.2 寄存器结构

通道	寄存器名	略称	R/W	P4 区地址	地址	存取长度
0	串行模式寄存器_0	SCSMR_0	R/W	H'FFE00000	H'1FE00000	16
	位速率寄存器_0	SCBRR_0	R/W	H'FFE00004	H'1FE00004	8
	串行控制寄存器_0	SCSCR_0	R/W	H'FFE00008	H'1FE00008	16
	发送 FIFO 数据寄存器_0	SCFTDR_0	W	H'FFE0000C	H'1FE0000C	8
	串行状态寄存器_0	SCFSR_0	R/(W)*1	H'FFE00010	H'1FE00010	16
	接收 FIFO 数据寄存器_0	SCFRDR_0	R	H'FFE00014	H'1FE00014	8
	FIFO 控制寄存器_0	SCFCR_0	R/W	H'FFE00018	H'1FE00018	16
	FIFO 数据计数置位寄存器_0	SCFDR_0	R	H'FFE0001C	H'1FE0001C	16
	串行端口寄存器_0	SCSPTR_0	R/W	H'FFE00020	H'1FE00020	16
	线路状态寄存器_0	SCLSR_0	R/(W)*2	H'FFE00024	H'1FE00024	16
	串行扩展模式寄存器_0	SCEMR_0	R/W	H'FFE00028	H'1FE00028	16
1	串行模式寄存器_1	SCSMR_1	R/W	H'FFE10000	H'1FE10000	16
	位速率寄存器_1	SCBRR_1	R/W	H'FFE10004	H'1FE10004	8
	串行控制寄存器_1	SCSCR_1	R/W	H'FFE10008	H'1FE10008	16
	发送 FIFO 数据寄存器_1	SCFTDR_1	W	H'FFE1000C	H'1FE1000C	8
	串行状态寄存器_1	SCFSR_1	R/(W)*1	H'FFE10010	H'1FE10010	16
	接收 FIFO 数据寄存器_1	SCFRDR_1	R	H'FFE10014	H'1FE10014	8
	FIFO 控制寄存器_1	SCFCR_1	R/W	H'FFE10018	H'1FE10018	16
	FIFO 数据计数置位寄存器_1	SCFDR_1	R	H'FFE1001C	H'1FE1001C	16
	串行端口寄存器_1	SCSPTR_1	R/W	H'FFE10020	H'1FE10020	16
	线路状态寄存器_1	SCLSR_1	R/(W)*2	H'FFE10024	H'1FE10024	16
	串行扩展模式寄存器_1	SCEMR_1	R/W	H'FFE10028	H'1FE10028	16
2	串行模式寄存器_2	SCSMR_2	R/W	H'FFE20000	H'1FE20000	16
	位速率寄存器_2	SCBRR_2	R/W	H'FFE20004	H'1FE20004	8
	串行控制寄存器_2	SCSCR_2	R/W	H'FFE20008	H'1FE20008	16
	发送 FIFO 数据寄存器_2	SCFTDR_2	W	H'FFE2000C	H'1FE2000C	8
	串行状态寄存器_2	SCFSR_2	R/(W)*1	H'FFE20010	H'1FE20010	16
	接收 FIFO 数据寄存器_2	SCFRDR_2	R	H'FFE20014	H'1FE20014	8
	FIFO 控制寄存器_2	SCFCR_2	R/W	H'FFE20018	H'1FE20018	16
	FIFO 数据计数置位寄存器_2	SCFDR_2	R	H'FFE2001C	H'1FE2001C	16
	串行端口寄存器_2	SCSPTR_2	R/W	H'FFE20020	H'1FE20020	16
	线路状态寄存器_2	SCLSR_2	R/(W)*2	H'FFE20024	H'1FE20024	16
	串行扩展模式寄存器_2	SCEMR_2	R/W	H'FFE20028	H'1FE20028	16

【注】 *1 为了清除标志, 只能写 0。bit15 ~ 8、3、2 是只读位, 不能写。

*2 为了清除标志, 只能写 0。bit15 ~ 1 是只读位, 不能写。

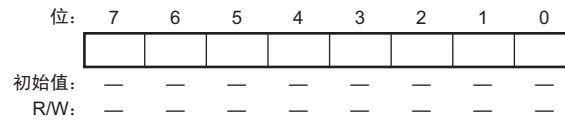
表 15.3 各处理模式的寄存器状态

通道	寄存器名	略称	初始值	待机	睡眠	模块待机
0	串行模式寄存器_0	SCSMR_0	H'0000	保持	保持	保持
	位速率寄存器_0	SCBRR_0	H'FF	保持	保持	保持
	串行控制寄存器_0	SCSCR_0	H'0000	保持	保持	保持
	发送 FIFO 数据寄存器_0	SCFTDR_0	不定值	保持	保持	保持
	串行状态寄存器_0	SCFSR_0	H'0060	保持	保持	保持
	接收 FIFO 数据寄存器_0	SCFRDR_0	不定值	保持	保持	保持
	FIFO 控制寄存器_0	SCFCR_0	H'0000	保持	保持	保持
	FIFO 数据计数置位寄存器_0	SCFDR_0	H'0000	保持	保持	保持
	串行端口寄存器_0	SCSPTR_0	H'0050	保持	保持	保持
	线路状态寄存器_0	SCLSR_0	H'0000	保持	保持	保持
	串行扩展模式寄存器_0	SCEMR_0	H'0000	保持	保持	保持
1	串行模式寄存器_1	SCSMR_1	H'0000	保持	保持	保持
	位速率寄存器_1	SCBRR_1	H'FF	保持	保持	保持
	串行控制寄存器_1	SCSCR_1	H'0000	保持	保持	保持
	发送 FIFO 数据寄存器_1	SCFTDR_1	不定值	保持	保持	保持
	串行状态寄存器_1	SCFSR_1	H'0060	保持	保持	保持
	接收 FIFO 数据寄存器_1	SCFRDR_1	不定值	保持	保持	保持
	FIFO 控制寄存器_1	SCFCR_1	H'0000	保持	保持	保持
	FIFO 数据计数置位寄存器_1	SCFDR_1	H'0000	保持	保持	保持
	串行端口寄存器_1	SCSPTR_1	H'0050	保持	保持	保持
	线路状态寄存器_1	SCLSR_1	H'0000	保持	保持	保持
	串行扩展模式寄存器_1	SCEMR_1	H'0000	保持	保持	保持
2	串行模式寄存器_2	SCSMR_2	H'0000	保持	保持	保持
	位速率寄存器_2	SCBRR_2	H'FF	保持	保持	保持
	串行控制寄存器_2	SCSCR_2	H'0000	保持	保持	保持
	发送 FIFO 数据寄存器_2	SCFTDR_2	不定值	保持	保持	保持
	串行状态寄存器_2	SCFSR_2	H'0060	保持	保持	保持
	接收 FIFO 数据寄存器_2	SCFRDR_2	不定值	保持	保持	保持
	FIFO 控制寄存器_2	SCFCR_2	H'0000	保持	保持	保持
	FIFO 数据计数置位寄存器_2	SCFDR_2	H'0000	保持	保持	保持
	串行端口寄存器_2	SCSPTR_2	H'0050	保持	保持	保持
	线路状态寄存器_2	SCLSR_2	H'0028	保持	保持	保持
	串行扩展模式寄存器_2	SCEMR_2	H'0000	保持	保持	保持

15.3.1 接收移位寄存器 (SCRSR)

SCRSR 是用于接收串行数据的寄存器。SCIF 从 LSB (bit0) 开始按接收的顺序, 将 RxD 引脚输入的串行数据设定到 SCRSR, 并且转换为并行数据。当接收完 1 字节的数据时, 数据就自动传送到接收 FIFO 数据寄存器 (SCFRDR)。

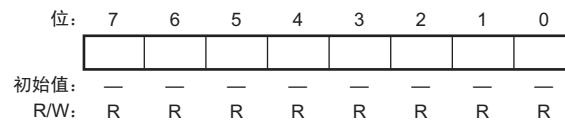
CPU 不能直接读写 SCRSR。



15.3.2 接收 FIFO 数据寄存器 (SCFRDR)

SCFRDR 是保存已接收的串行数据的 16 段 FIFO 寄存器。当接收完 1 字节的串行数据时, SCIF 就将接收的串行数据从接收移位寄存器 (SCRSR) 传送到 SCFRDR, 结束数据的接收。能连续接收数据, 直到存满 16 字节为止。CPU 能读 SCFRDR, 但是不能写。在接收 FIFO 数据寄存器没有接收数据的状态下读数据时, 读取值为不定值。

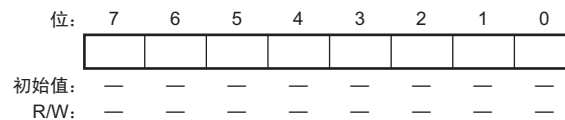
如果 SCFRDR 存满接收数据, 此后接收的串行数据就会丢失。



15.3.3 发送移位寄存器 (SCTSR)

SCTSR 是用于发送串行数据的寄存器。SCIF 一旦将发送数据从发送 FIFO 数据寄存器 (SCFTDR) 传送到 SCTSR, 就将数据从 LSB (bit0) 开始按顺序发送到 TxD 引脚, 进行串行数据的发送。当发送完 1 字节的数据时, 就自动将下一个发送数据从 SCFTDR 传送到 SCTSR, 开始发送。

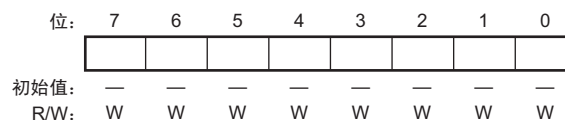
CPU 不能直接读写 SCTSR。



15.3.4 发送 FIFO 数据寄存器 (SCFTDR)

SCFTDR 是保存串行发送数据的 16 段 FIFO 寄存器。如果检测到发送移位寄存器 (SCTSR) 为空, SCIF 就将已写在 SCFTDR 的发送数据传送到 SCTSR, 开始进行串行发送。能连续发送串行数据, 直到 SCFTDR 没有发送数据为止。CPU 能随时写 SCFTDR。

如果 SCFTDR 存满发送数据 (16 字节), 就不能写下一个数据。即使想写, 数据也会被忽视。



15.3.5 串行模式寄存器 (SCSMR)

SCSMR 是用于设定 SCIF 的串行通信格式以及选择波特率发生器时钟源的寄存器。

CPU 能随时读写 SCSMR。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	C/A	CHR	PE	O/E	STOP	—	—	CKS[1:0]
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 8	—	全 0	R	保留位 读取值为 0，只能写 0。
7	C/A	0	R/W	通信模式位 SCIF 的运行模式能选择异步模式或者时钟同步模式。 0: 异步模式 1: 时钟同步模式
6	CHR	0	R/W	字符长度位 异步模式的数据长度能选择 7 位或者 8 位。与 CHR 的设定无关，时钟同步模式的数据长度固定为 8 位。 0: 8 位数据 1: 7 位数据 * 【注】 * 当选择 7 位数据时，发送 FIFO 数据寄存器的 MSB (bit7) 不被发送。
5	PE	0	R/W	奇偶校验允许位 在异步模式中，选择在发送时是否附加奇偶校验位并且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，不附加也不检查奇偶校验位。 0: 禁止附加和检查奇偶校验位 1: 允许附加和检查奇偶校验位 * 【注】 * 如果将 PE 位置 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再发送；在接收时检查已接收的奇偶校验位是否为 O/E 位指定的偶校验或者奇校验。
4	O/E	0	R/W	奇偶校验模式位 选择是通过偶校验还是通过奇校验进行奇偶校验的附加和检查。O/E 位的设定只在异步模式中将 PE 位置 1 (允许附加和检查奇偶校验位) 时才有效。在时钟同步模式或者异步模式中禁止附加和检查奇偶性时，O/E 位的指定无效。 0: 偶校验 *1 1: 奇校验 *2 【注】 *1 如果设定为偶校验，就在发送时附加奇偶校验位，使奇偶校验位和发送字符中 1 的个数为偶数，然后发送；在接收时检查奇偶校验位和接收字符中 1 的个数是否为偶数。 *2 如果设定为奇校验，就在发送时附加奇偶校验位，使奇偶校验位和发送字符中 1 的个数为奇数，然后发送；在接收时检查奇偶校验位和接收字符中 1 的个数是否为奇数。

位	位名	初始值	R/W	说明
3	STOP	0	R/W	<p>停止位长位</p> <p>选择异步模式中停止位的长度为 1 位或者 2 位。STOP 位的设定只在异步模式中有效。如果设定为时钟同步模式，因为不附加停止位，所以此位的设定无效。在接收时与 STOP 位的设定无关，只检查已接收的停止位的第 1 位。当停止位的第 2 位是 1 时，作为停止位处理；是 0 时，作为下一个发送字符的起始位处理。</p> <p>0: 1 个停止位 在发送时，给发送字符的末尾附加 1 位的 1 (停止位) 后发送。</p> <p>1: 2 个停止位 在发送时，给发送字符的末尾附加 2 位的 1 (停止位) 后发送。</p>
2	—	0	R	<p>保留位</p> <p>读取值为 0，只能写 0。</p>
1、0	CKS[1:0]	00	R/W	<p>时钟选择位</p> <p>选择内部波特率发生器的内部时钟源。</p> <p>有关时钟源、位速率寄存器的设定值和波特率的关系，请参照“15.3.8 位速率寄存器 (SCBRR)”。</p> <p>00: Pϕ 时钟 01: Pϕ/4 时钟 10: Pϕ/16 时钟 11: Pϕ/64 时钟</p> <p>【注】 Pϕ: 外围时钟</p>

15.3.6 串行控制寄存器 (SCSCR)

SCSCR 是进行 SCIF 的发送和接收、允许 / 禁止中断请求以及选择发送和接收时钟源的寄存器。CPU 能随时读写 SCSCR。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TIE	RIE	TE	RE	REIE	—	—	CKE[1:0]
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 8	—	全 0	R	<p>保留位</p> <p>读取值为 0，只能写 0。</p>
7	TIE	0	R/W	<p>发送中断允许位</p> <p>将串行发送数据从发送 FIFO 数据寄存器 (SCFTDR) 传送到发送移位寄存器 (SCTSR)，在发送 FIFO 数据寄存器的数据个数少于指定的发送触发个数并且串行状态寄存器 (SCFSR) 的 TDFE 标志为 1 时，允许或者禁止发送 FIFO 数据空中断 (TXI) 请求。</p> <p>0: 禁止发送 FIFO 数据空中断 (TXI) 请求 1: 允许发送 FIFO 数据空中断 (TXI) 请求 *</p> <p>【注】 * 能通过以下方法解除 TXI:</p> <ul style="list-style-type: none"> • 给 SCFTDR 写发送数据 (多于指定的发送触发个数)。 • 在读到 TDFE 标志的 1 后清 0。 • 将 TIE 清 0。

位	位名	初始值	R/W	说明
6	RIE	0	R/W	<p>接收中断允许位</p> <p>允许或者禁止以下的中断请求：</p> <ul style="list-style-type: none"> • 串行状态寄存器 (SCFSR) 的 RDF 标志或者 DR 标志被置 1 时的接收 FIFO 数据满中断 (RXI) 请求 • 在 SCFSR 的 ER 标志被置 1 时的接收错误中断 (ERI) 请求 • 在 SCFSR 的 BRK 标志或者线路状态寄存器 (SCLSR) 的 ORER 标志被置 1 时的中止中断 (BRI) 请求 <p>0: 禁止接收 FIFO 数据满中断 (RXI) 请求、接收错误中断 (ERI) 请求和中止中断 (BRI) 请求</p> <p>1: 允许接收 FIFO 数据满中断 (RXI) 请求、接收错误中断 (ERI) 请求和中止中断 (BRI) 请求 *</p> <p>【注】 * 能通过以下方法解除 RXI 中断请求：</p> <ul style="list-style-type: none"> • 读 DR 或者 RDF 标志的 1 后清 0。 • 将 RIE 位清 0。 <p>* 能通过以下方法解除 ERI 和 BRI 中断请求：</p> <ul style="list-style-type: none"> • 读 ER、BRK 或者 ORER 标志的 1 后清 0。 • 将 RIE 和 REIE 位清 0。
5	TE	0	R/W	<p>发送允许位</p> <p>允许或者禁止开始串行发送。</p> <p>0: 禁止发送</p> <p>1: 允许发送 *</p> <p>【注】 * 如果在此状态下给 SCFTDR 写发送数据，就开始串行发送。必须在将 TE 位置 1 前设定 SCSMR 和 SCFCR，决定发送格式，并且清除发送 FIFO。</p>
4	RE	0	R/W	<p>接收允许位</p> <p>允许或者禁止开始串行接收。</p> <p>0: 禁止接收 *1</p> <p>1: 允许接收 *2</p> <p>【注】 *1 必须注意：即使将 RE 位清 0 也不影响 DR、ER、BRK、RDF、FER、PER、ORER 各位，各位保持状态。</p> <p>*2 在此状态下，如果在异步模式中检测到起始位或者在同步模式中检测到同步时钟，就开始串行接收。必须在将 RE 位置 1 前设定串行模式寄存器 (SCSMR) 和 FIFO 控制寄存器 (SCFCR)，决定接收格式，并且清除接收 FIFO。</p>

位	位名	初始值	R/W	说明
3	REIE	0	R/W	<p>接收错误的中断允许</p> <p>允许 / 禁止接收错误中断 (ERI) 请求和中止中断 (BRI) 请求的发生。但是, REIE 位的设定只在 RIE 位为 0 时有效。</p> <p>0: 禁止接收错误中断 (ERI) 请求和中止中断 (BRI) 请求 1: 允许接收错误中断 (ERI) 请求和中止中断 (BRI) 请求 *</p> <p>【注】 * 能通过以下方法解除 ERI 和 BRI 中断请求:</p> <ul style="list-style-type: none"> • 读 ER、BRK 或者 ORER 标志的 1 后清 0。 • 将 RIE 和 REIE 位清 0。 <p>即使将 RIE 清 0, 只要将 REIE 置 1, 也产生 ERI、BRI 中断请求。在 DMAC 传送时, 要将 ERI、BRI 中断请求通知中断控制器时设定 REIE。</p>
2	—	0	R	<p>保留位</p> <p>读取值为 0, 只能写 0。</p>
1、0	CKE[1:0]	00	R/W	<p>时钟允许位</p> <p>选择 SCIF 的时钟源以及允许或者禁止 SCK 引脚的时钟输出。由 CKE[1:0] 决定是将 SCK 引脚设定为串行时钟输出引脚还是串行时钟输入引脚。在时钟同步模式中设定为同步时钟输出时, 必须在将 SCSMR 的 C/A 位置 1 后设定 CKE[1:0]。</p> <ul style="list-style-type: none"> • 异步模式 <ul style="list-style-type: none"> 00: 内部时钟 /SCK 引脚为输入引脚 (忽视输入信号) 01: 内部时钟 /SCK 引脚为时钟输出 (输出时钟的频率是位速率的 16 或者 8 倍) 10: 外部时钟 /SCK 引脚为时钟输入 (输入时钟的频率是位速率的 16 或者 8 倍) 11: 禁止设定 • 时钟同步模式 <ul style="list-style-type: none"> 00: 内部时钟 /SCK 引脚为同步时钟输出 01: 内部时钟 /SCK 引脚为同步时钟输出 10: 外部时钟 /SCK 引脚为同步时钟输入 11: 禁止设定

15.3.7 串行状态寄存器 (SCFSR)

SCFSR 是 16 位寄存器。高 8 位表示接收 FIFO 数据寄存器的数据接收错误个数，低 8 位表示 SCIF 的运行状态标志。

CPU 能随时读写 SCFSR。但是，不能给 ER、TEND、TDFE、BRK、RDF、DR 的各状态标志写 1，需要预先读到 1 后才能清 0。PER 标志 (bit15 ~ 12、2) 和 FER 标志 (bit11 ~ 8、3) 只能读不能写。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PER[3:0]				FER[3:0]				ER	TEND	TDFE	BRK	FER	PER	RDF	DR
初始值:	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/(W)*	R/(W)*

【注】* 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说明
15 ~ 12	PER[3:0]	0000	R	奇偶校验错误个数位 表示接收 FIFO 数据寄存器 (SCFRDR) 保存的接收数据中发生奇偶校验错误的个数。 在 SCFSR 的 ER 位被置位后，bit15 ~ 12 的值表示发生奇偶校验错误的个数。当 SCFRDR 的 16 字节接收数据都发生奇偶校验错误时，PER[3:0] 为 0000。
11 ~ 8	FER[3:0]	0000	R	帧错误个数位 表示接收 FIFO 数据寄存器 (SCFRDR) 保存的接收数据中发生帧错误的个数。 在 SCFSR 的 ER 位被置位后，bit11 ~ 8 的值表示发生帧错误的个数。当 SCFRDR 的 16 字节接收数据都发生帧错误时，FER[3:0] 为 0000。
7	ER	0	R/(W)*	接收错误位 表示发生了帧错误或者接收有奇偶校验的数据时发生了奇偶校验错误。 *1 0: 表示正在接收或者接收正常结束 [清除条件] • 上电复位 • 在读 ER=1 的状态后写 0 时 1: 表示在接收时发生了帧错误或者奇偶校验错误 [置位条件] • 在 1 次数据接收结束时，检查到接收数据的最后停止位是 0 时 *2 • 在接收时的接收数据和奇偶校验位的 1 的个数与串行模式寄存器 (SCSMR) 的 O/E 位指定的偶校验 / 奇校验的设定不一致时 【注】 *1 在将 SCSCR 的 RE 位清 0 时，ER 位不受影响而保持以前的状态。即使发生接收错误也将接收数据传送到 SCFRDR，并继续接收运行。能通过 SCFSR 的 FER 位和 PER 位判断从 SCFRDR 读取的数据是否发生接收错误。 *2 在 2 个停止位模式中，只检查第 1 个停止位而不检查第 2 个停止位。

位	位名	初始值	R/W	说明
6	TEND	1	R/(W)*	<p>发送结束位</p> <p>表示在发送字符的最后位时，SCFTDR 无有效数据而结束发送。</p> <p>0: 表示正在发送</p> <p>[清除条件]</p> <ul style="list-style-type: none"> 在将发送数据写到 SCFTDR 并且读 TEND=1 的状态后给 TEND 标志写 0 时 <p>1: 表示发送结束</p> <p>[置位条件]</p> <ul style="list-style-type: none"> 上电复位 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，SCFTDR 没有发送数据
5	TDFE	1	R/(W)*	<p>发送 FIFO 数据空位</p> <p>表示数据从发送 FIFO 数据寄存器 (SCFTDR) 传送到发送移位寄存器 (SCTSR) 以及 SCFTDR 的数据个数少于 FIFO 控制寄存器 (SCFCR) 的 TTRG[1:0] 指定的发送触发数据个数，允许给 SCFTDR 写发送数据。</p> <p>0: 表示写到 SCFTDR 的发送数据个数多于指定的发送触发个数</p> <p>[清除条件]</p> <ul style="list-style-type: none"> 在读 TDFE=1 的状态后将多于指定发送触发个数的数据写到 SCFTDR 并且给 TDFE 写 0 时 在通过发送 FIFO 数据空中断 (TXI) 启动 DMAC 并且将多于指定发送触发个数的数据写到 SCFTDR 时 <p>1: 表示写到 SCFTDR 的发送数据个数小于等于指定的发送触发个数 *1</p> <p>[置位条件]</p> <ul style="list-style-type: none"> 上电复位 在写到 SCFTDR 的发送数据个数小于等于指定的发送触发个数时 <p>【注】 *1 由于 SCFTDR 是 16 字节的 FIFO 寄存器，所以在 TDFE=1 的状态下能写的最大数据个数为 16 减去指定的发送触发个数。即使写更多的数据，此数据也被忽视。SCFDR 的高 8 位表示 SCFTDR 的数据个数。</p>
4	BRK	0	R/(W)*	<p>中止检测位</p> <p>表示在接收数据时检测到中止信号。</p> <p>0: 无中止信号</p> <p>[清除条件]</p> <ul style="list-style-type: none"> 上电复位 在读 BRK=1 的状态后给 BRK 标志写 0 时 <p>1: 接收到中止信号 *1</p> <p>[置位条件]</p> <ul style="list-style-type: none"> 在接收含有帧错误的帧数据后至少继续接收到 1 帧的 0 (低电平) 时 <p>【注】 *1 如果检测到中止，就在检测后停止给 SCFRDR 传送接收数据 (H'00)。当中止结束并且接收信号变为标记的 1 时，就重新开始传送接收数据。</p>

位	位名	初始值	R/W	说明
3	FER	0	R	<p>帧错误位</p> <p>表示在异步模式中从接收 FIFO 数据寄存器 (SCFRDR) 读取的数据是否发生帧错误。</p> <p>0: 表示从 SCFRDR 读到的下一个接收数据未发生帧错误</p> <p>[清除条件]</p> <ul style="list-style-type: none"> • 上电复位 • 下一个 SCFRDR 的读数据未发生帧错误 <p>1: 表示从 SCFRDR 读到的下一个接收数据发生了帧错误</p> <p>[置位条件]</p> <ul style="list-style-type: none"> • 下一个 SCFRDR 的读数据发生了帧错误
2	PER	0	R	<p>奇偶校验错误位</p> <p>表示在异步模式中从接收 FIFO 数据寄存器 (SCFRDR) 读到的数据是否发生奇偶校验错误。</p> <p>0: 表示从 SCFRDR 读到的下一个接收数据未发生奇偶校验错误</p> <p>[清除条件]</p> <ul style="list-style-type: none"> • 上电复位 • 下一个 SCFRDR 的读数据未发生奇偶校验错误 <p>1: 表示从 SCFRDR 读到的下一个接收数据发生了奇偶校验错误</p> <p>[置位条件]</p> <ul style="list-style-type: none"> • 下一个 SCFRDR 的读数据发生了奇偶校验错误
1	RDF	0	R/(W)*	<p>接收 FIFO 数据满位</p> <p>表示将接收数据传送到接收 FIFO 数据寄存器 (SCFRDR)，并且 SCFRDR 的数据个数多于 FIFO 控制寄存器 (SCFCR) 的 RTRG[1:0] 指定的接收触发个数。</p> <p>0: 表示写到 SCFRDR 的接收数据个数少于指定的接收触发个数</p> <p>[清除条件]</p> <ul style="list-style-type: none"> • 上电复位 • 在读 RDF=1 后，从 SCFRDR 读取少于指定接收触发个数的接收数据并且给 RDF 写 0 时 • 在通过接收 FIFO 数据满中断 (RXI) 启动 DMAC 并且从 SCFRDR 读到少于指定接收触发个数的接收数据时 <p>1: 表示 SCFRDR 的接收数据个数大于等于指定的接收触发个数</p> <p>[置位条件]</p> <ul style="list-style-type: none"> • 在将大于等于指定接收触发个数的接收数据个数保存到 SCFRDR 时 *1 <p>【注】 *1 由于 SCFRDR 是 16 字节的 FIFO 寄存器，所以在 RDF 为 1 时能读到的数据最大个数为指定的接收触发个数。在读取 SCFRDR 的全部数据后，如果还继续读数据，读取的数据就为不定值。SCFDR 的低 8 位表示 SCFRDR 的接收数据个数。</p>

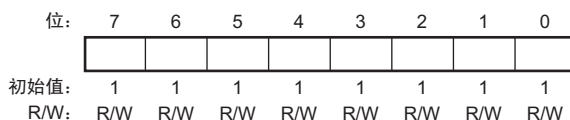
位	位名	初始值	R/W	说明
0	DR	0	R/(W)*	<p>接收数据就绪位</p> <p>表示在异步模式中将少于指定接收触发个数的数据保存到接收 FIFO 数据寄存器 (SCFRDR) 并且从最后的停止位经过 15ETU 时间后没有接收到下一个数据。在设定为时钟同步模式时, 此位不被置位。</p> <p>0: 表示正在接收或者在接收正常结束后 SCFRDR 没有接收数据</p> <p>[清除条件]</p> <ul style="list-style-type: none"> • 上电复位 • 在读 DR=1 状态后, 读到 SCFRDR 中的全部接收数据并且给 SCFRDR 写 0 时 • 在通过接收 FIFO 数据满中断 (RXI) 启动 DMAC 并且读到 SCFRDR 中的全部接收数据时 <p>1: 表示没有接收到下一个接收数据</p> <p>[置位条件]</p> <ul style="list-style-type: none"> • 在 SCFRDR 保存了少于指定接收触发个数的数据并且从最后的停止位经过 15ETU 时间 *1 后没有接收到下一个数据时 <p>【注】 *1 相当于 8 位、1 个停止位格式的 1.5 帧。(ETU: Element Time Unit: 要素时间单位)</p>

【注】 * 为了清除标志, 只能在读取 1 后写 0。

15.3.8 位速率寄存器 (SCBRR)

SCBRR 是 8 位寄存器, 根据串行模式寄存器 (SCSMR) 的 CKS[1:0] 和串行扩展模式寄存器 (SCEMR) 的 BGDM 位、ABCS 位, 设定串行发送 / 接收的位速率。

CPU 能随时读写 SCBRR。在上电复位时, SCBRR 被初始化为 H'FF。由于各通道的波特率发生器是独立控制的, 所以能分别给 3 个通道设定不同的值。



用以下计算式求 SCBRR 的设定值。

【异步模式】

- 波特率发生器为正常模式 (SCEMR 的 BGDM=0)

$$N = (P\phi / (64 \times 2^{2n-1} \times B)) \times 10^6 - 1$$
 (以 16 倍位速率的基本时钟运行时)

$$N = (P\phi / (32 \times 2^{2n-1} \times B)) \times 10^6 - 1$$
 (以 8 倍位速率的基本时钟运行时)
- 波特率发生器为倍速模式 (SCEMR 的 BGDM=1)

$$N = (P\phi / (32 \times 2^{2n-1} \times B)) \times 10^6 - 1$$
 (以 16 倍位速率的基本时钟运行时)

$$N = (P\phi / (16 \times 2^{2n-1} \times B)) \times 10^6 - 1$$
 (以 8 倍位速率的基本时钟运行时)

【时钟同步模式】

$$N = \frac{P\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: 位速率 (bit/s)

N: 波特率发生器的 SCBRR 的设定值 ($0 \leq N \leq 255$)

(设定值必须满足电特性)

Pφ: 外围模块 (Pck) 频率 (MHz)

n: 波特率发生器的输入时钟 (n=0 ~ 3)

(n 与时钟的关系请参照表 15.4)

表 15.4 SCSMR 的设定

n	时钟	SCSMR 的设定值	
		CKS[1]	CKS[0]
0	Pφ	0	0
1	Pφ/4	0	1
2	Pφ/16	1	0
3	Pφ/64	1	1

用以下计算式求异步模式的位速率误差。

- 波特率发生器为正常模式 (SCEMR 的 BGDM=0)
 - 误差 (%) = $\{((P\phi \times 10^6) / ((N+1) \times B \times 64 \times 2^{2n-1})) - 1\} \times 100$
(以 16 倍位速率的基本时钟运行时)
 - 误差 (%) = $\{((P\phi \times 10^6) / ((N+1) \times B \times 32 \times 2^{2n-1})) - 1\} \times 100$
(以 8 倍位速率的基本时钟运行时)
- 波特率发生器为倍速模式 (SCEMR 的 BGDM=1)
 - 误差 (%) = $\{((P\phi \times 10^6) / ((N+1) \times B \times 32 \times 2^{2n-1})) - 1\} \times 100$
(以 16 倍位速率的基本时钟运行时)
 - 误差 (%) = $\{((P\phi \times 10^6) / ((N+1) \times B \times 16 \times 2^{2n-1})) - 1\} \times 100$
(以 8 倍位速率的基本时钟运行时)

在波特率发生器为正常模式 (SCEMR 的 BGDM=0) 并且以 16 倍位速率的基本时钟运行 (SCEMR 的 ABCS=0) 时, 异步模式和时钟同步模式的 SCBRR 设定例子分别如表 15.5 和表 15.6 所示。

表 15.5 位速率的 SCBRR 的设定例子 (异步模式、BGDM=0、ABCS=0)

位速率 (bit/s)	P ϕ (MHz)								
	45			50			54		
	n	N	误差 (%)	n	N	误差 (%)	n	N	误差 (%)
110	3	199	-0.12	3	221	-0.02	3	239	-0.12
150	3	145	0.33	3	162	-0.15	3	175	-0.12
300	3	72	0.33	3	80	0.47	3	87	-0.12
600	2	145	0.33	2	162	-0.15	2	175	-0.12
1200	2	72	0.33	2	80	0.47	2	87	-0.12
2400	1	145	0.33	1	162	-0.15	1	175	-0.12
4800	1	72	0.33	1	80	0.47	1	87	-0.12
9600	0	145	0.33	0	162	-0.15	0	175	-0.12
19200	0	72	0.33	0	80	0.47	0	87	-0.12
31250	0	44	0.00	0	49	0.00	0	53	0.00
38400	0	36	-1.02	0	40	-0.76	0	43	-0.12

表 15.6 位速率的 SCBRR 的设定例子 (时钟同步模式)

位速率 (bit/s)	P ϕ (MHz)					
	45		50		54	
	n	N	n	N	n	N
110	—	—	—	—	—	—
250	—	—	—	—	—	—
500	—	—	—	—	—	—
1k	3	175	3	194	3	210
2.5k	3	69	3	77	3	83
5k	2	140	2	155	2	168
10k	2	69	2	77	2	83
25k	1	112	1	124	1	134
50k	0	224	0	249	1	67
100k	0	112	0	124	0	134
250k	0	44	0	49	0	53
500k	0	22	0	24	0	26
1M	0	10	0	12	0	13
2M	0	5	0	5	0	6

【符号说明】

—: 虽然能设定, 但是有误差。

使用波特率发生器时的异步模式的各频率的最大位速率如表 15.7 所示, 外部时钟输入时的异步模式的最大位速率如表 15.8 所示, 外部时钟输入 ($t_{\text{Scyc}}=12t_{\text{pCyc}}$ 时*) 时的时钟同步模式的最大位速率如表 15.9 所示。

【注】* 必须确认是否满足本 LSI 和通信对象的电特性。

表 15.7 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)	设定值				最大位速率 (bit/s)
	BGDM	ABCS	n	N	
45	0	0	0	0	1406250
		1	0	0	2812500
	1	0	0	0	2812500
		1	0	0	5625000
50	0	0	0	0	1562500
		1	0	0	3125000
	1	0	0	0	3125000
		1	0	0	6250000
54	0	0	0	0	1687500
		1	0	0	3375000
	1	0	0	0	3375000
		1	0	0	6750000

表 15.8 外部时钟输入时的最大位速率 (异步模式)

P ϕ (MHz)	外部输入时钟 (MHz)	设定值	最大位速率 (bit/s)
		ABCS	
45	11.2500	0	703125
		1	1406250
50	12.5000	0	781250
		1	1562500
54	13.5000	0	843750
		1	1687500

表 15.9 外部时钟输入时的最大位速率 (时钟同步模式、 $t_{Scyc}=12t_{pcyc}$ 时)

P ϕ (MHz)	外部输入时钟 (MHz)	最大位速率 (bit/s)
45	3.7500	3750000.0
50	4.1666	4166666.6
54	4.5000	4500000.0

15.3.9 FIFO 控制寄存器 (SCFCR)

SCFCR 是清除发送 FIFO 数据寄存器和接收 FIFO 数据寄存器的数据个数以及设定触发数据个数的寄存器，包括循环反馈测试的允许位。

CPU 能随时读写 SCFCR。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	RSTRG[2:0]			RTRG[1:0]		TTRG[1:0]		MCE	TFRST	RFRST	LOOP
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 11	—	全 0	R	保留位 读取值为 0，只能写 0。
10 ~ 8	RSTRG[2:0]	000	R/W	RTS 输出有效触发位 当接收 FIFO 数据寄存器 (SCFRDR) 保存的接收数据个数大于等于如下所示的触发设定个数时，RTS 信号为高电平。 000: 15 001: 1 010: 4 011: 6 100: 8 101: 10 110: 12 111: 14
7、6	RTRG[1:0]	00	R/W	接收 FIFO 数据个数触发位 作为将串行状态寄存器 (SCFSR) 的 RDF 标志置位的基准，设定接收数据个数 (指定接收触发个数)。当接收 FIFO 数据寄存器 (SCFRDR) 保存的接收数据个数大于等于如下所示的设定触发个数时，将 RDF 标志置 1。 • 异步模式 00: 1 01: 4 10: 8 11: 14 • 时钟同步模式 00: 1 01: 2 10: 8 11: 14 【注】 在时钟同步模式中通过 DMAC 传送接收数据时，必须将接收触发个数设定为 1，否则需要通过 CPU 读取 SCFRDR 中留下的接收数据。

位	位名	初始值	R/W	说明
5、4	TTRG[1:0]	00	R/W	<p>发送 FIFO 数据个数触发位</p> <p>作为将串行状态寄存器 (SCFSR) 的 TDFE 标志置位的基准, 设定发送数据个数 (指定发送触发个数)。当发送 FIFO 数据寄存器 (SCFTDR) 保存的发送数据个数小于等于如下所示的设定触发个数时, 将 TDFE 标志置 1。</p> <p>00: 8 (8) *</p> <p>01: 4 (12) *</p> <p>10: 2 (14) *</p> <p>11: 0 (16) *</p> <p>【注】 * () 内的数值表示 TDFE 标志被置 1 时的 SCFTDR 寄存器的空字节数。</p>
3	MCE	0	R/W	<p>调制解调器控制允许位</p> <p>允许或者禁止调制解调器控制信号 $\overline{\text{CTS}}$ 和 $\overline{\text{RTS}}$。</p> <p>在通道 0 ~ 2 和时钟同步模式时, 必须将 MCE 置 0。</p> <p>0: 禁止调制解调器信号 *</p> <p>1: 允许调制解调器信号</p> <p>【注】 * 与输入值无关, 将 $\overline{\text{CTS}}$ 和 $\overline{\text{RTS}}$ 都固定为 0。</p>
2	TFRST	0	R/W	<p>发送 FIFO 数据寄存器复位</p> <p>将发送 FIFO 数据寄存器清空, 使发送数据无效。</p> <p>0: 禁止复位 *</p> <p>1: 允许复位</p> <p>【注】 * 在上电复位时进行复位。</p>
1	RFRST	0	R/W	<p>接收 FIFO 数据寄存器复位</p> <p>将接收 FIFO 数据寄存器清空, 使接收数据无效。</p> <p>0: 禁止复位 *</p> <p>1: 允许复位</p> <p>【注】 * 在上电复位时进行复位。</p>
0	LOOP	0	R/W	<p>循环反馈测试位</p> <p>在内部将发送输出引脚 (TxD) 和接收输入引脚 (RxD)、$\overline{\text{RTS}}$ 引脚和 $\overline{\text{CTS}}$ 引脚连接, 允许循环反馈测试。</p> <p>0: 禁止循环反馈测试</p> <p>1: 允许循环反馈测试</p>

15.3.10 FIFO 数据计数置位寄存器 (SCFDR)

SCFDR 表示发送 FIFO 数据寄存器 (SCFTDR) 和接收 FIFO 数据寄存器 (SCFRDR) 保存的数据个数。高 8 位表示 SCFTDR 的发送数据个数, 低 8 位表示 SCFRDR 的接收数据个数。CPU 能随时读 SCFDR。

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

位	位名	初始值	R/W	说明
15 ~ 13	—	全 0	R	保留位 读取值为 0, 只能写 0。
12 ~ 8	T[4:0]	0000	R	表示 SCFTDR 保存的未发送数据个数。 H'00 表示没有发送数据, H'10 表示发送数据全部保存在 SCFTDR。
7 ~ 5	—	全 0	R	保留位 读取值为 0, 只能写 0。
4 ~ 0	R[4:0]	0000	R	表示 SCFRDR 保存的接收数据个数。 H'00 表示没有接收数据, H'10 表示接收数据全部保存在 SCFRDR。

15.3.11 串行端口寄存器 (SCSPTR)

SCSPTR 控制多路复用引脚 SCIF 的端口输入 / 输出和数据。能通过 bit7 和 bit6、bit5 和 bit4、bit3 和 bit2 分别对 $\overline{\text{RTS}}$ 引脚、 $\overline{\text{CTS}}$ 引脚和 SCK 引脚进行数据的读操作和输出数据的写操作。能通过 bit1 和 bit0, 读取 RxD 引脚的输入数据并且将输出数据写到 TxD 引脚。控制串行发送 / 接收的中止信号。

CPU 能随时读写 SCSPTR。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RTSIO	RTSDT	CTSIO	CTSDT	SCKIO	SCKDT	SPB2IO	SPB2DT
初始值:	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 8	—	全 0	R	保留位 读取值为 0, 只能写 0。
7	RTSIO	0	R/W	$\overline{\text{RTS}}$ 端口输入 / 输出位 指定串行端口 $\overline{\text{RTS}}$ 引脚的输入 / 输出。实际上, 在将 $\overline{\text{RTS}}$ 引脚作为端口输出引脚并且输出 RTSDT 位的设定值时, 必须将 SCFCR 的 MCE 位置 0。 0: 不将 RTSDT 位的值输出到 $\overline{\text{RTS}}$ 引脚 1: 将 RTSDT 位的值输出到 $\overline{\text{RTS}}$ 引脚
6	RTSDT	1	R/W	$\overline{\text{RTS}}$ 端口数据位 指定串行端口 $\overline{\text{RTS}}$ 引脚的输入 / 输出数据。通过 RTSIO 位指定输入或者输出。输出时, 将 RTSDT 位的值输出到 $\overline{\text{RTS}}$ 引脚。与 RTSIO 位的值无关, 能从 RTSDT 位读到 $\overline{\text{RTS}}$ 引脚的值。但是, 需要通过 PFC 预先设定为 $\overline{\text{RTS}}$ 输入 / 输出。 0: 输入 / 输出数据为低电平 1: 输入 / 输出数据为高电平

位	位名	初始值	R/W	说明
5	CTSIO	0	R/W	<p>CTS 端口输入 / 输出位</p> <p>指定串行端口 $\overline{\text{CTS}}$ 引脚的输入 / 输出。实际上，在将 $\overline{\text{CTS}}$ 引脚作为端口输出引脚并且输出 CTSDT 位的设定值时，必须将 SCFCR 的 MCE 位置 0。</p> <p>0: 不将 CTSDT 位的值输出到 $\overline{\text{CTS}}$ 引脚 1: 将 CTSDT 位的值输出到 $\overline{\text{CTS}}$ 引脚</p>
4	CTSDT	1	R/W	<p>CTS 端口数据位</p> <p>指定串行端口 $\overline{\text{CTS}}$ 引脚的输入 / 输出数据。通过 CTSIO 位指定输入或者输出。输出时，将 CTSDT 位的值输出到 $\overline{\text{CTS}}$ 引脚。与 CTSIO 位的值无关，能从 CTSDT 位读到 $\overline{\text{CTS}}$ 引脚的值。但是，需要通过 PFC 预先设定为 $\overline{\text{CTS}}$ 输入 / 输出。</p> <p>0: 输入 / 输出数据为低电平 1: 输入 / 输出数据为高电平</p>
3	SCKIO	0	R/W	<p>SCK 端口输入 / 输出位</p> <p>指定串行端口 SCK 引脚的输入 / 输出。实际上，在将 SCK 引脚作为端口输出引脚并且输出 SCKDT 位的设定值时，必须将 SCSCR 的 CKE1 和 CKE0 位置 0。</p> <p>0: 不将 SCKDT 位的值输出到 SCK 引脚 1: 将 SCKDT 位的值输出到 SCK 引脚</p>
2	SCKDT	0	R/W	<p>SCK 端口数据位</p> <p>指定串行端口 SCK 引脚的输入 / 输出数据。通过 SCKIO 位指定输入或者输出。输出时，将 SCKDT 位的值输出到 SCK 引脚。与 SCKIO 位的值无关，能从 SCKDT 位读到 SCK 引脚的值。但是，需要通过 PFC 预先设定为 SCK 输入 / 输出。</p> <p>0: 输入 / 输出数据为低电平 1: 输入 / 输出数据为高电平</p>
1	SPB2IO	0	R/W	<p>串行端口中止输入 / 输出位</p> <p>指定串行端口 TxD 引脚的输出条件。实际上，在将 TxD 引脚作为端口输出引脚并且输出 SPB2DT 位的设定值时，必须将 SCSCR 的 TE 位置 0。</p> <p>0: 不将 SPB2DT 位的值输出到 TxD 引脚 1: 将 SPB2DT 位的值输出到 TxD 引脚</p>
0	SPB2DT	0	R/W	<p>串行端口中止数据位</p> <p>指定串行端口 RxD 引脚的输入数据和 TxD 引脚的输出数据。通过 SPB2IO 位指定输入或者输出。如果将 TxD 引脚设定为输出，SPB2DT 位的值就输出到 TxD 引脚。与 SPB2IO 位的值无关，能从 SPB2DT 位读到 RxD 引脚的值。但是，需要通过 PFC 预先设定为 RxD 输入和 TxD 输出。</p> <p>0: 输入 / 输出数据为低电平 1: 输入 / 输出数据为高电平</p>

15.3.12 线路状态寄存器 (SCLSR)

CPU 能随时读写 SCLSR，但是不能给 ORER 的状态标志写 1。需要预先读 1 才能清 0。

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

【注】* 为了清除标志，只能在读1后写0。

位	位名	初始值	R/W	说明
15 ~ 1	—	全 0	R	保留位 读取值为 0，只能写 0。
0	ORER	0	R/(W)*	<p>溢出错误 表示在接收时发生溢出错误，异常结束。</p> <p>0: 正在接收或者接收正常结束 *1</p> <p>[清除条件]</p> <ul style="list-style-type: none"> 上电复位 在读 ORER=1 的状态后写 0 时 <p>1: 表示在接收时发生溢出错误 *2</p> <p>[置位条件]</p> <ul style="list-style-type: none"> 在接收到 16 字节数据 (接收 FIFO 满) 的状态下结束下一个串行接收时 <p>【注】 *1 在将串行控制寄存器 (SCSCR) 的 RE 位清 0 时，ORER 标志不受影响而保持以前的状态。</p> <p>*2 接收 FIFO 数据寄存器 (SCFRDR) 保持发生溢出错误前的接收数据，以后接收的数据将会丢失。而且，在 ORER=1 的状态下，不能继续进行以后的串行接收。</p>

15.3.13 串行扩展模式寄存器 (SCEMR)

CPU 能随时读写 SCEMR。在异步模式 (SCSMR 的 $\overline{C/A}=0$) 并且将内部时钟作为时钟源、SCK 引脚为输入引脚 (SCSCR 的 $CKE[1:0]=00$) 时, 如果将 BGDM 位置 1, SCIF 内部的波特率发生器就在倍速模式中运行。

通过更改 ABCS 位的设定, 能选择异步模式的位速率的基本时钟。

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

位	位名	初始值	R/W	说明
15 ~ 8	—	全 0	R	保留位 读取值为 0, 只能写 0。
7	BGDM	0	R/W	波特率发生器倍速模式位 如果将 BGDM 位置 1, SCIF 内部的波特率发生器就在倍速模式中运行。在异步模式 (SCSMR 的 $\overline{C/A}=0$) 并且将内部时钟作为时钟源、SCK 引脚为输入引脚 (SCSCR 的 $CKE[1:0]=00$) 时, BGDM 的设定有效。在上述设定以外的情况下必须使用正常模式。 0: 正常模式 1: 倍速模式
6 ~ 1	—	全 0	R	保留位 读取值为 0, 只能写 0。
0	ABCS	0	R/W	异步基本时钟选择位 选择异步模式的位速率的基本时钟。在异步模式 (SCSMR 的 $\overline{C/A}=0$) 中, ABCS 位的设定有效。 0: 以 16 倍位速率的基本时钟频率运行 1: 以 8 倍位速率的基本时钟频率运行

15.4 运行说明

15.4.1 概要

SCIF 能通过异步模式（以字符为单位取得同步进行通信）和时钟同步模式（通过时钟脉冲取得同步进行通信）2 种方式进行串行通信。

SCIF 内置分别用于发送和接收的 16 段 FIFO 缓冲器，能减少 CPU 的系统开销，进行高速连续通信。另外，RTS 信号和 CTS 信号能用作调制解调器的控制信号。通过串行模式寄存器（SCSMR）选择发送 / 接收格式，如表 15.5 所示。另外，SCIF 的时钟源取决于串行控制寄存器（SCSCR）CKE[1:0] 的组合，如表 15.11 所示。

(1) 异步模式

- 数据长度：能选择 7 位或者 8 位
- 能选择附加奇偶校验和附加 1 位 / 2 位的停止位。
（通过这些组合决定发送 / 接收格式和字符长度）
- 能检测到接收时的帧错误、奇偶校验错误、接收 FIFO 数据满、溢出错误、接收数据就绪和中止。
- 表示发送和接收 FIFO 寄存器各自的保存数据个数。
- SCIF 的时钟源：能选择内部时钟或者外部时钟。
当选择内部时钟时：SCIF 通过波特率发生器的时钟运行。
当选择外部时钟时：需要输入 16 倍或者 8 倍位速率的频率时钟（不使用内部波特率发生器）。

(2) 时钟同步模式

- 发送 / 接收格式：固定为 8 位数据
- 能检测到接收时的溢出错误。
- SCIF 的时钟源：能选择内部时钟或者外部时钟。
当选择内部时钟时：SCIF 通过波特率发生器的时钟运行，将该时钟作为同步时钟输出到外部
当选择外部时钟时：不使用内部波特率发生器，通过输入的外部同步时钟运行

表 15.10 SCSMR 的设定值和 SCIF 发送 / 接收格式

SCSMR				模式	SCIF 发送 / 接收格式		
bit7	bit6	bit5	bit3		数据长度	奇偶校验位	停止位长度
C/A	CHR	PE	STOP				
0	0	0	0	异步模式	8 位	无	1 位
			1				2 位
		1	0			1 位	
			1			2 位	
	1	0	0		7 位	无	1 位
			1				2 位
		1	0			1 位	
			1			2 位	
1	x	x	x	时钟同步模式	8 位	无	无

【符号说明】x: Don't care

表 15.11 SCSMR、SCSCR 的设定值和 SCIF 时钟源的选择

SCSMR	SCSCR	模式	时钟源	SCK 引脚的功能
bit7	bit1、0			
C/A	CKE[1:0]			
0	00	异步模式	内部	SCIF 不使用 SCK 引脚。
	01			输出 16 或者 8 倍位速率的频率时钟。
	10		外部	输入 16 或者 8 倍位速率的频率时钟。
	11		禁止设定	
1	0x	时钟同步模式	内部	输出同步时钟。
	10		外部	输入同步时钟。
	11		禁止设定	
			禁止设定	

【符号说明】x: Don't care

【注】在使用波特率发生器的倍速模式 (BGDM=1) 时, 必须设定为异步模式 (C/A=0), 并且将内部时钟设定为时钟源以及不使用 SCK 引脚 (CKE[1:0]=00)。

15.4.2 异步模式的运行

异步模式是以字符为单位取得同步进行串行通信的模式, 发送或者接收给数据附加了起始位 (表示通信开始) 和停止位 (表示通信结束) 的字符。

在 SCIF 内部, 发送部和接收部相互独立, 所以能进行全双工通信。发送部和接收部都是 16 段的 FIFO 缓冲器结构, 在发送和接收时能读写数据, 因此能连续进行发送和接收。

异步串行通信的一般格式如图 15.2 所示。

在异步串行通信的情况下, 通信线路通常保持在标记状态 (高电平)。SCIF 监视通信线路, 将 0 状态 (低电平) 的信号作为起始位, 开始串行通信。

串行通信的 1 个字符由起始位 (低电平), 数据 (LSB 先: 从最低位开始)、奇偶校验位 (高电平/低电平)、停止位 (高电平) 的顺序构成。

在异步模式中接收数据时, SCIF 通过起始位的下降沿取得同步。SCIF 通过 16 倍或者 8 倍位速率频率的第 8 个或者第 4 个时钟对数据进行采样, 因此能在各位的中央取到通信数据。

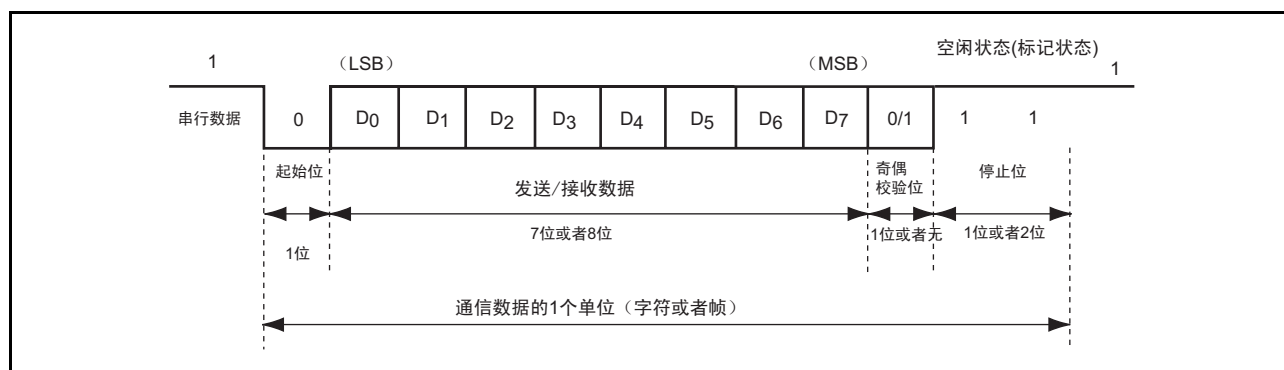


图 15.2 异步通信的数据格式
(8 位数据 / 有奇偶校验 / 2 个停止位的例子)

(1) 发送 / 接收格式

能设定的发送 / 接收格式如表 15.12 所示。

发送 / 接收格式有 8 种，能通过设定串行模式寄存器 (SCSMR) 进行选择。

表 15.12 串行发送 / 接收格式 (异步模式)

SCSMR 的设定			串行发送 / 接收格式和帧长												
CHR	PE	STOP	1	2	3	4	5	6	7	8	9	10	11	12	
0	0	0	START	8位数据							STOP				
		1	START	8位数据							STOP	STOP			
	1	0	START	8位数据							P	STOP			
		1	START	8位数据							P	STOP	STOP		
1	0	0	START	7位数据						STOP					
		1	START	7位数据						STOP	STOP				
	1	0	START	7位数据						P	STOP				
		1	START	7位数据						P	STOP	STOP			

【符号说明】

START: 起始位

STOP: 停止位

P: 奇偶校验位

(2) 时钟

通过设定 SCSMR 的 C/A 位和串行控制寄存器 (SCSCR) 的 CKE[1:0]，能从内部波特率发生器生成的内部时钟和 SCK 引脚输入的外部时钟中选择 SCIF 的发送 / 接收时钟。有关 SCIF 的时钟源选择，请参照表 15.11。

在将外部时钟输入到 SCK 引脚时，必须输入 16 倍或者 8 倍位速率的频率时钟。

在通过内部时钟运行时，能从 SCK 引脚输出时钟，此时输出的时钟频率为位速率的 16 倍或者 8 倍。

(3) 数据的发送和接收

• SCIF的初始化（异步模式）

在发送和接收数据前，必须先将串行控制寄存器（SCSCR）的TE位和RE位清0，然后按以下顺序对SCIF进行初始化。

在要更改运行模式和通信格式等的情况下，必须在将TE位和RE位清0后按以下顺序进行。当TE位清0时，发送移位寄存器（SCTSR）就被初始化。必须注意：即使将TE位和RE位清0，串行状态寄存器（SCFSR）、发送FIFO数据寄存器（SCFTDR）和接收FIFO数据寄存器（SCFRDR）也不被初始化，这些寄存器的内容不变。必须在发送数据全部被发送并且SCFSR的TEND标志被置位后将TE位清0。TE位在发送过程中也能清0，但是在发送数据被清0后就变为标记状态。另外，在将TE位再次置1后并且在开始发送前，必须将SCFCR的TFRST位置1，将SCFTDR清空。

在使用外部时钟时，由于运行不确定，所以在有初始化的运行过程中不能停止时钟。SCIF初始化的流程图例子如图 15.3 所示。

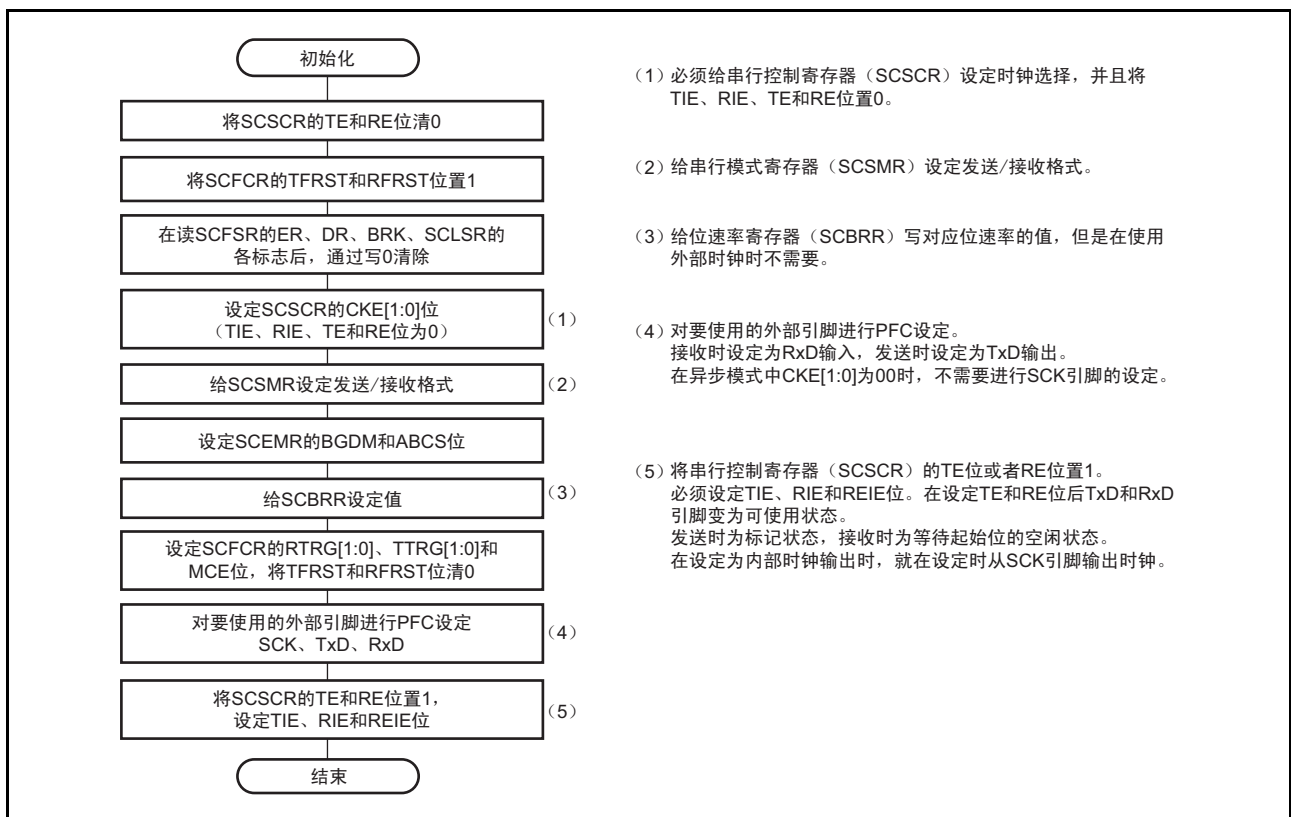


图 15.3 SCIF 初始化的流程图例子

- 串行数据发送 (异步模式)

串行发送的流程图例子如图 15.4 所示。

在允许 SCIF 的发送后, 必须按以下步骤发送串行数据。

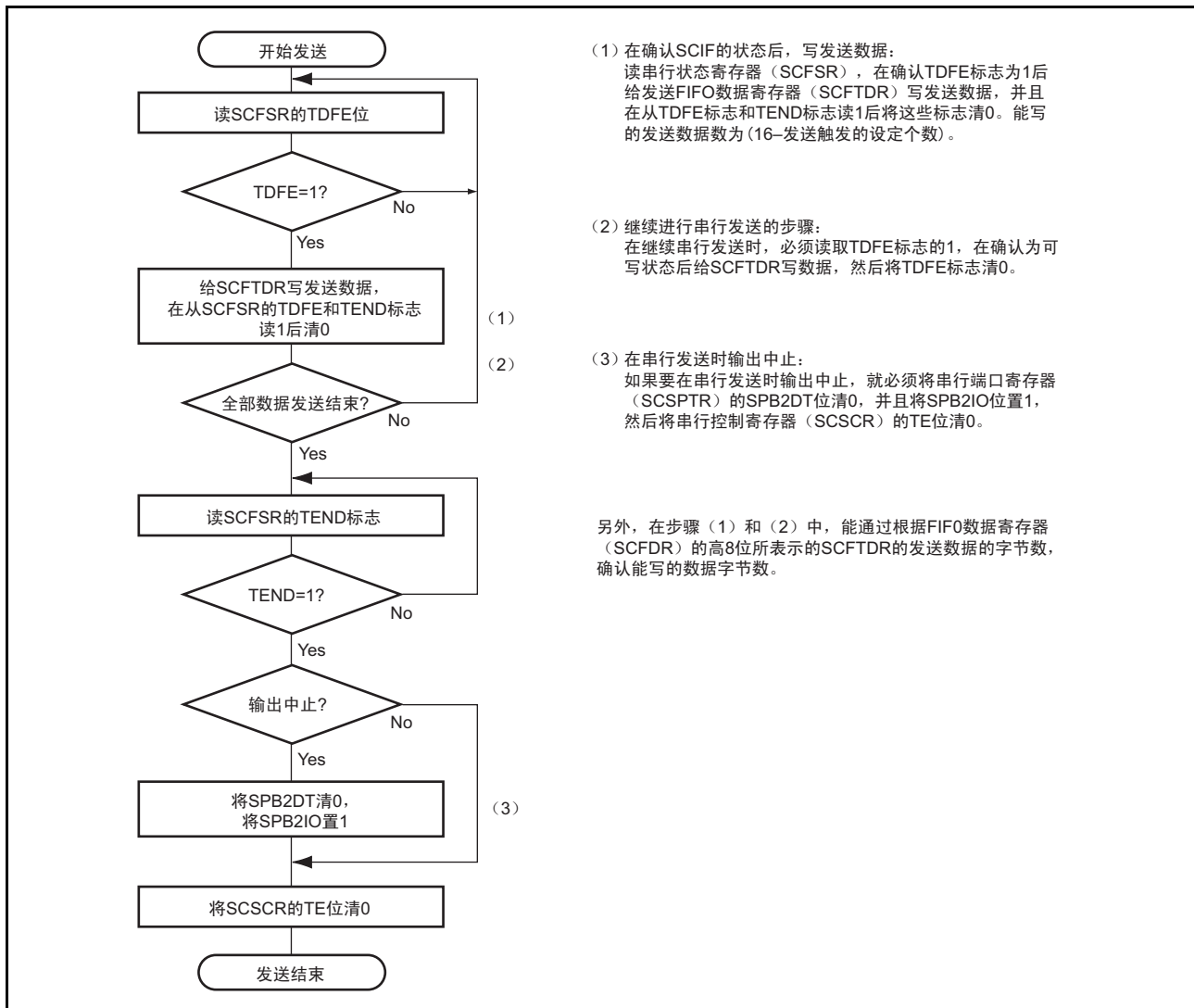


图 15.4 串行发送的流程图例子

在串行发送时, SCIF 的运行如下:

- 当数据被写到发送 FIFO 数据寄存器 (SCFTDR) 时, SCIF 就将数据从 SCFTDR 传送到发送移位寄存器 (SCTSR)。必须在发送数据被写到 SCFTDR 前确认串行状态寄存器 (SCFSR) 的 TDFE 标志是否已被置 1。能写的发送数据字节数为 (16-发送触发设定个数)。
- 当将数据从 SCFTDR 传送到 SCTSR 后开始发送时, 就连续进行发送, 直到 SCFTDR 中没有发送数据为止。如果 SCFTDR 的发送数据字节数小于等于 FIFO 控制寄存器 (SCFCR) 设定的发送触发个数, TDFE 标志就被置位。此时, 如果串行控制寄存器 (SCSCR) 的 TIE 位为 1, 就产生发送 FIFO 数据空中断 (TXI) 请求。

按以下顺序从 TxD 引脚发送串行发送数据:

- 起始位 : 输出 1 位的 0。
- 发送数据 : 从 LSB 开始按顺序输出 8 位或者 7 位数据。

- c. 奇偶校验位：输出 1 位奇偶校验位（偶校验或奇校验）
（也能选择不输出奇偶校验位的格式）。
 - d. 停止位：输出 1 位或者 2 位的 1（停止位）。
 - e. 标记状态：继续输出 1，直到下一个起始位被发送为止。
3. SCIF 通过发送停止位的时序来检查 SCFTDR 的发送数据。当有数据时，就将数据从 SCFTDR 传送到 SCTSR，并在发送停止位后开始下一个帧的串行发送。
在异步模式中，发送的运行例子如图 15.5 所示。

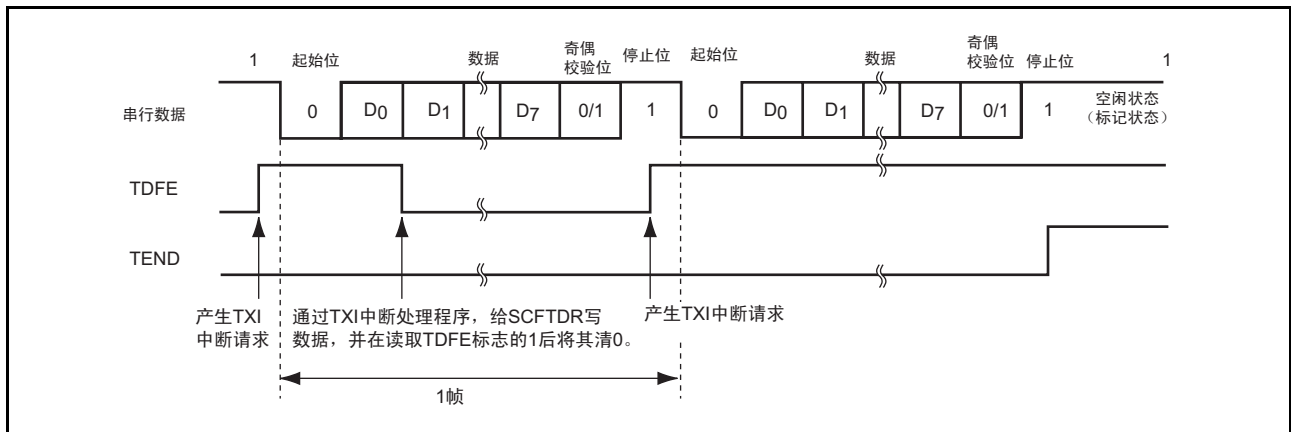


图 15.5 发送的运行例子（8 位数据 / 有奇偶性 / 1 个停止位的例子）

4. 如果允许调制解调器控制，就能通过 $\overline{\text{CTS}}$ 的输入值，停止或者重新开始发送。在发送过程中，如果 $\overline{\text{CTS}}$ 被置 1，就在 1 帧数据发送结束后变为标记状态；如果 $\overline{\text{CTS}}$ 被置 0，就从起始位开始输出下一个发送数据。
使用调制解调器控制的运行例子如图 15.6 所示。

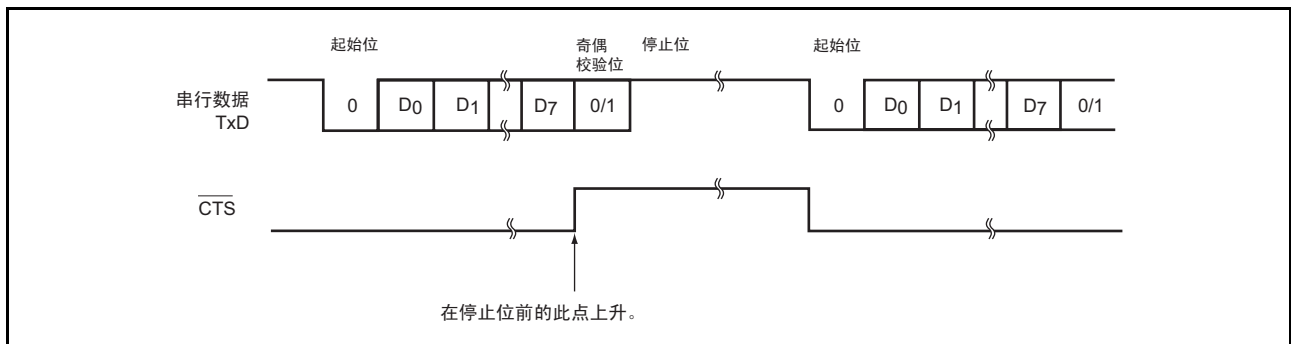


图 15.6 使用调制解调器控制的运行例子（ $\overline{\text{CTS}}$ ）

- 串行数据接收 (异步模式)

串行接收的流程图例子如图 15.7 和图 15.8 所示。

在允许 SCIF 的接收后, 必须按以下步骤接收串行数据。

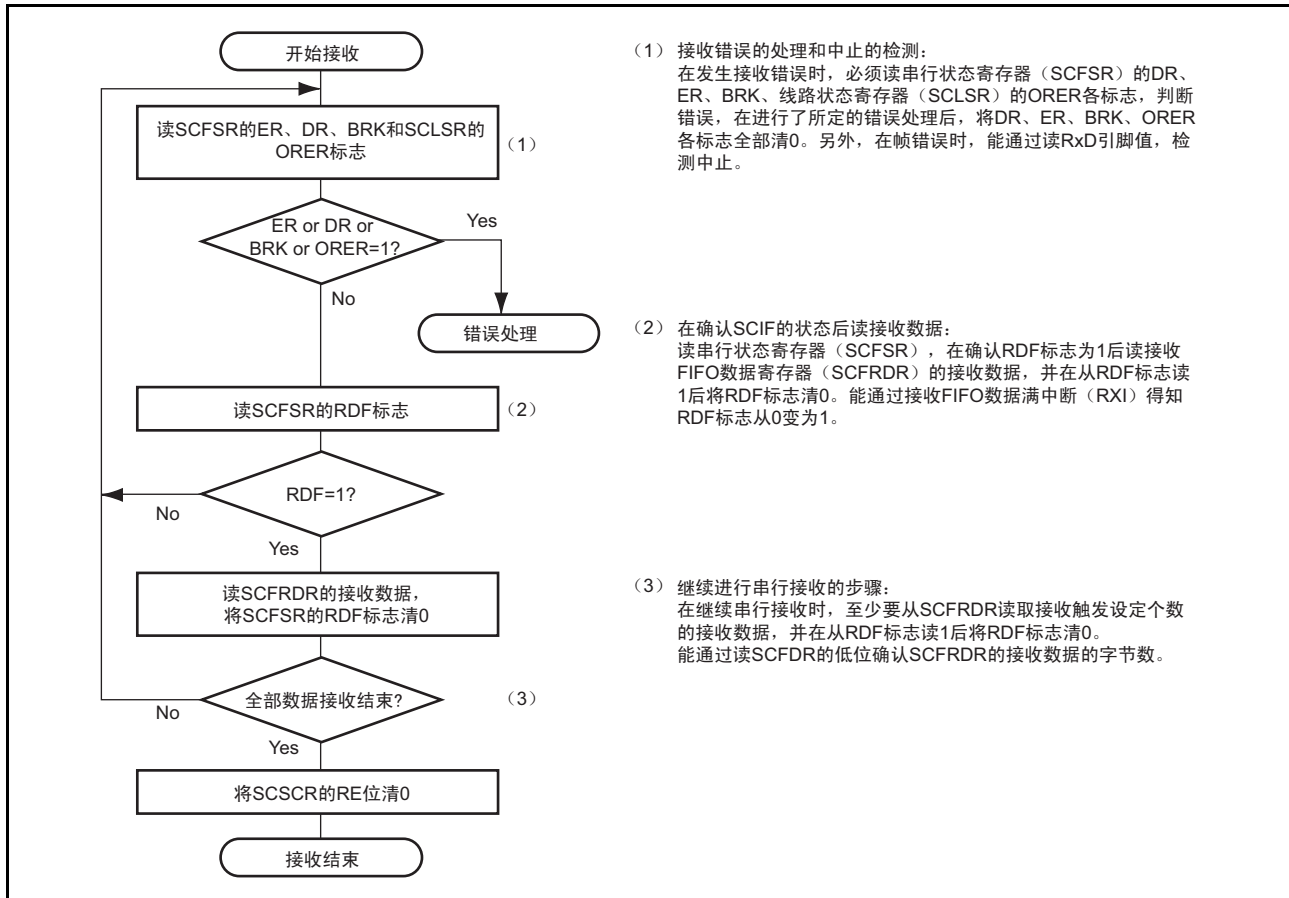


图 15.7 串行接收的流程图例子 (1)

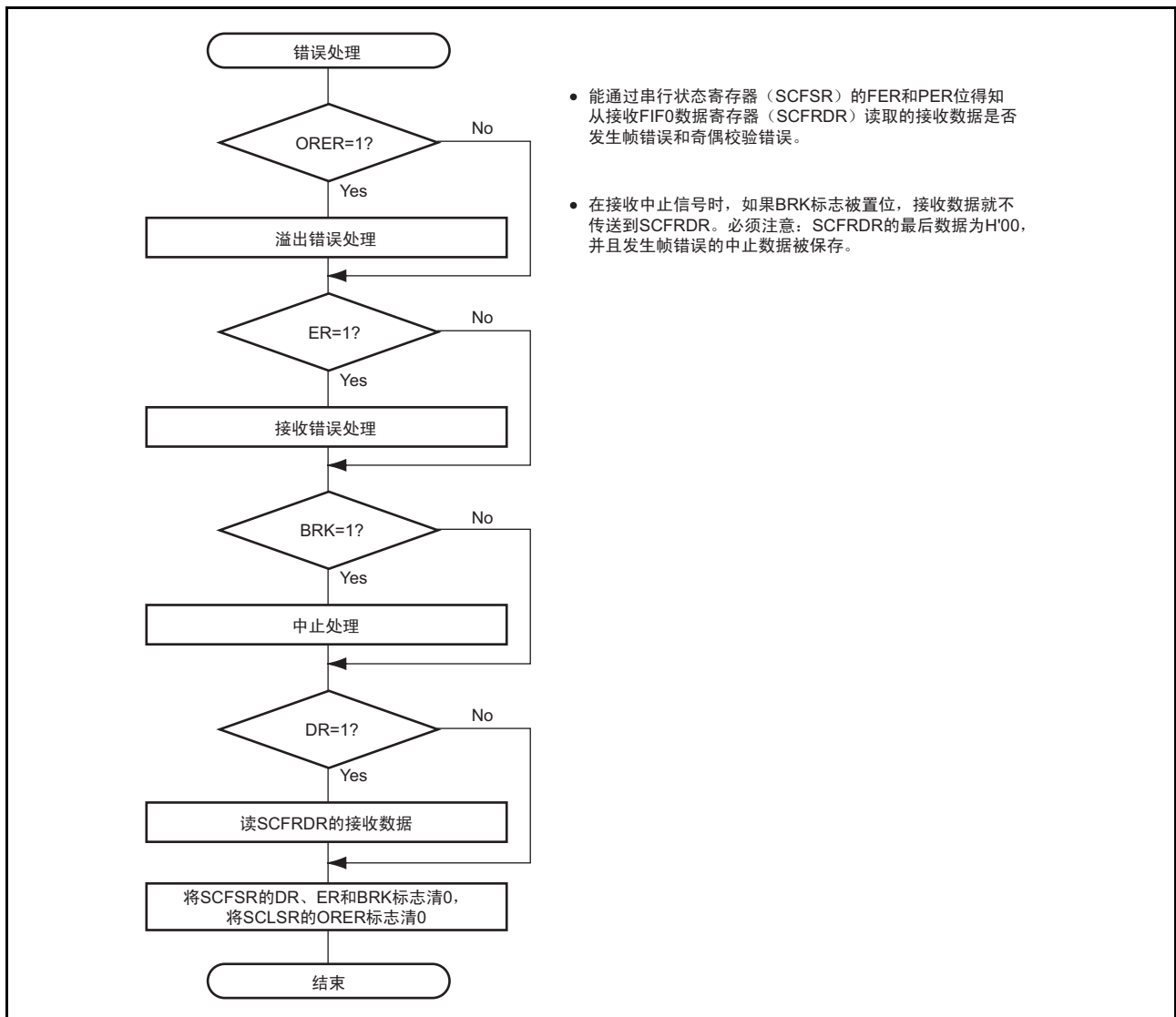


图 15.8 串行接收的流程图例子 (2)

在接收时，SCIF 的运行如下：

1. SCIF 监视通信线路，如果检测到起始位的 0，就在内部取得同步，开始接收。
2. 按从 SCRSR 的 LSB 到 MSB 的顺序保存已接收的数据。
3. 接收奇偶校验位和停止位。

在接收后，SCIF 进行以下检查：

- a. 检查停止位：检查停止位是否为 1。但是在 2 个停止位的情况下，只检查第 1 个停止位。
- b. 检查是否处于能将接收数据从接收移位寄存器 (SCRSR) 传送到 SCFRDR 的状态。
- c. 检查溢出错误：检查 ORER 标志是否为 0（表示未发生溢出错误）。
- d. 中止检查：检查 BRK 标志是否为 0（表示未设定中止状态）。

在以上检查全部通过时，将接收数据保存到 SCFRDR。

【注】 即使发生奇偶校验错误或者帧错误，也继续接收数据。

4. 当RDF或者DR标志变为1时，如果SCSCR的RIE位已被置1，就产生接收FIFO数据满中断（RXI）请求；当ER标志变为1时，如果SCSCR的RIE位或者REIE位已被置1，就产生接收错误中断（ERI）请求；当BRK标志或者ORER标志变为1时，如果SCSCR的RIE位或者REIE位已被置1，就产生中止接收中断（BRI）请求。

在异步模式中，接收的运行例子如图 15.9 所示。

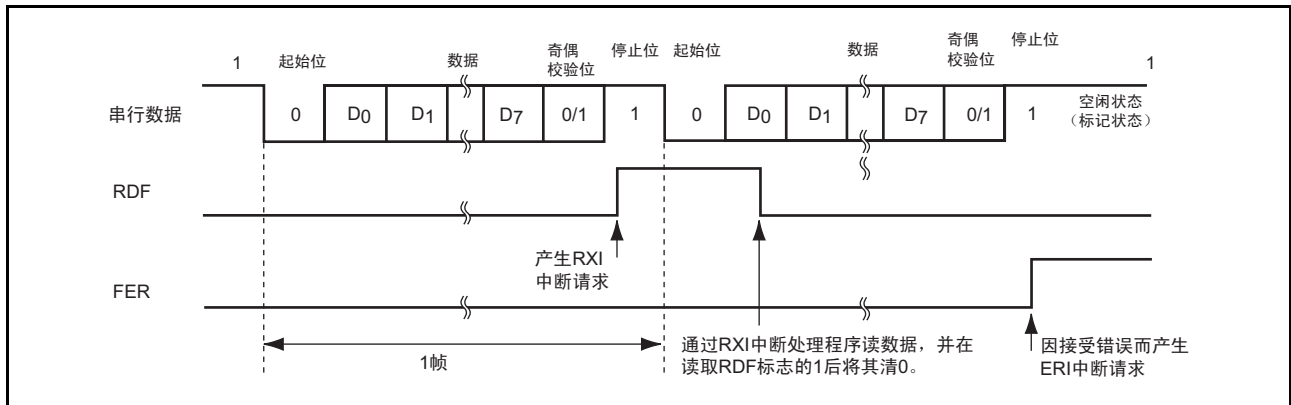


图 15.9 SCIF 接收的运行例子（8 位数据 / 有奇偶性 / 1 个停止位的例子）

5. 当调制解调器控制有效并且SCFRDR为空时，输出 $\overline{\text{RTS}}$ 信号。当 $\overline{\text{RTS}}$ 为0时能进行接收；当 $\overline{\text{RTS}}$ 为1时，表示SCFRDR的数据个数大于等于RTS输出的有效触发设定个数。使用调制解调器控制的运行例子如图 15.10 所示。

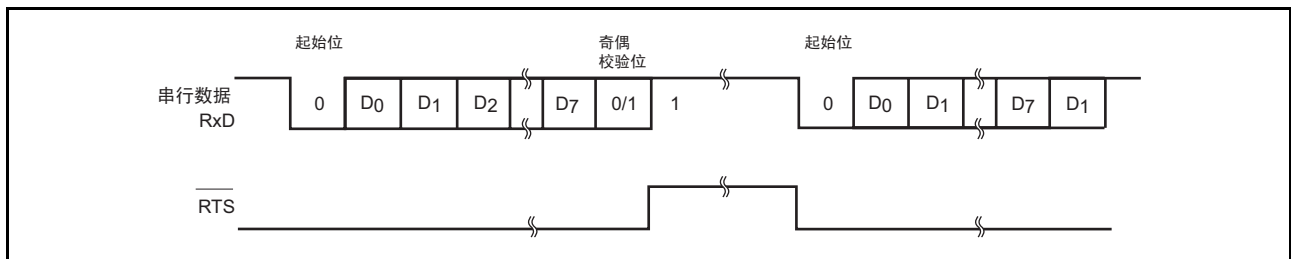


图 15.10 使用调制解调器控制的运行例子（ $\overline{\text{RTS}}$ ）

15.4.3 时钟同步模式的运行

时钟同步模式是与时钟脉冲同步进行数据发送和接收的模式，适用于高速串行通信。

在 SCIF 内部，发送部和接收部相互独立，所以能通过共享时钟进行全双工通信。发送部和接收部都是 16 段 FIFO 缓冲器结构，在发送和接收时能读写数据，所以能连续进行发送和接收。

时钟同步串行通信的一般格式如图 15.11 所示。

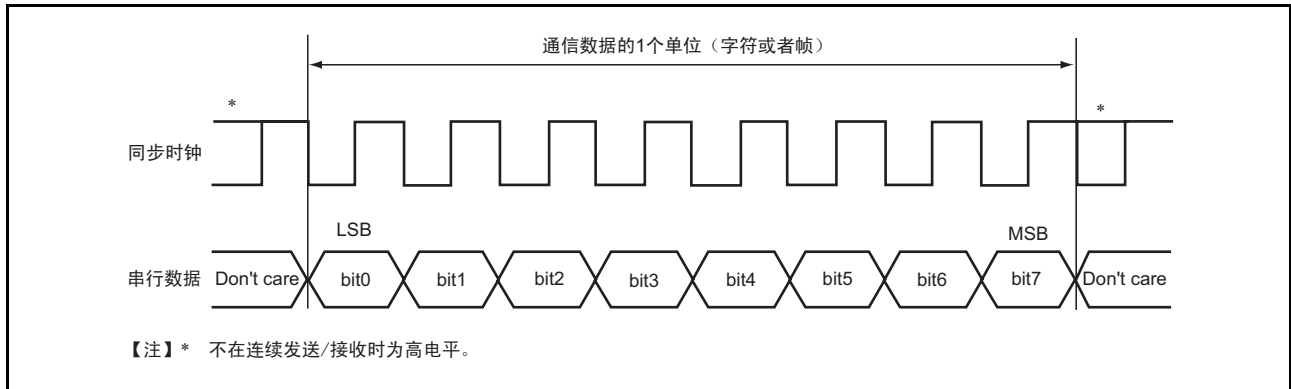


图 15.11 时钟同步通信的数据格式

在时钟同步串行通信的情况下，从同步时钟的下降沿到下一个下降沿之前将数据输出到通信线路，在同步时钟的上升沿保证数据有效。

按从数据的 LSB 开始到最后 MSB 的顺序输出串行通信的 1 个字符。MSB 输出后的通信线路状态保持 MSB 状态。

在时钟同步模式中，SCIF 与同步时钟的上升沿同步进行数据接收。

(1) 发送 / 接收格式

固定为 8 位数据。

不能附加奇偶校验位。

(2) 时钟

通过设定 SCSMR 的 $\overline{C/A}$ 位和 SCSCR 的 CKE[1:0]，能选择内部波特率发生器生成的内部时钟或者 SCK 引脚输入的外部同步时钟。

在通过内部时钟运行时，从 SCK 引脚输出同步时钟。同步时钟在 1 个字符的发送和接收时输出 8 个脉冲，而在不进行发送和接收时固定为高电平。在只进行接收时，如果选择内部时钟，就在 SCSCR 的 RE 位为 1 的期间，输出时钟脉冲，直到接收 FIFO 内的数据个数达到接收触发设定个数为止。

(3) 数据的发送和接收

• SCIF 的初始化（时钟同步模式）

在发送和接收数据前，必须先将串行控制寄存器（SCSCR）的 TE 和 RE 位清 0，然后按以下顺序对 SCIF 进行初始化。

在更改模式和通信格式等的情况下，必须在将 TE 和 RE 位清 0 后按以下顺序进行。当 TE 位清 0 时，发送移位寄存器（SCTSR）就被初始化。必须注意：即使将 RE 位清 0，也保持 RDF、PER、FER、ORER 的各标志和接收数据寄存器（SCRDR）的内容。

SCIF 初始化的流程图例子如图 15.12 所示。

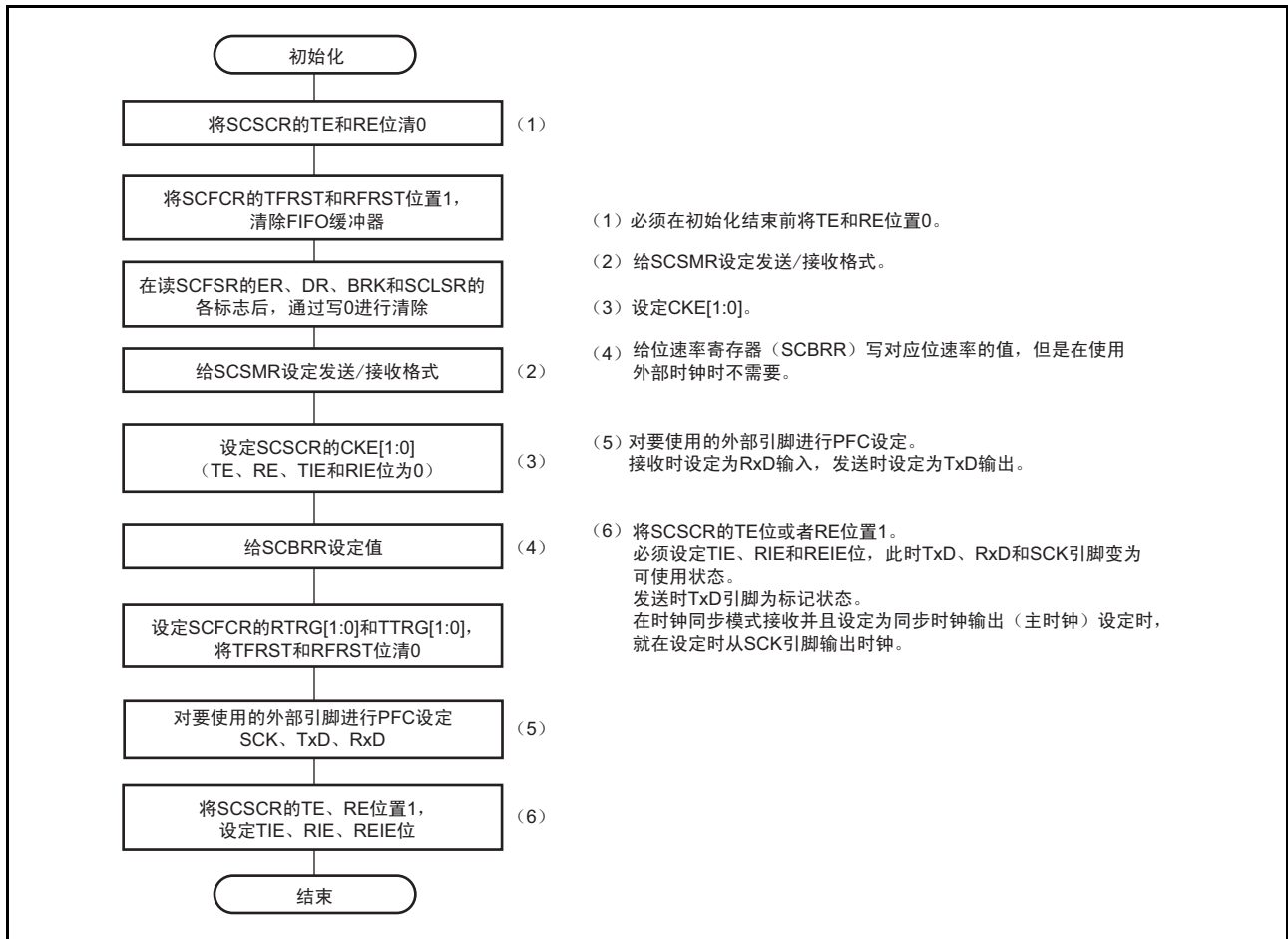


图 15.12 SCIF 初始化的流程图例子

- 串行数据发送 (时钟同步模式)

串行发送的流程图例子如图 15.13 所示。

在将 SCIF 设定为能进行发送的状态后, 必须按以下步骤发送串行数据。

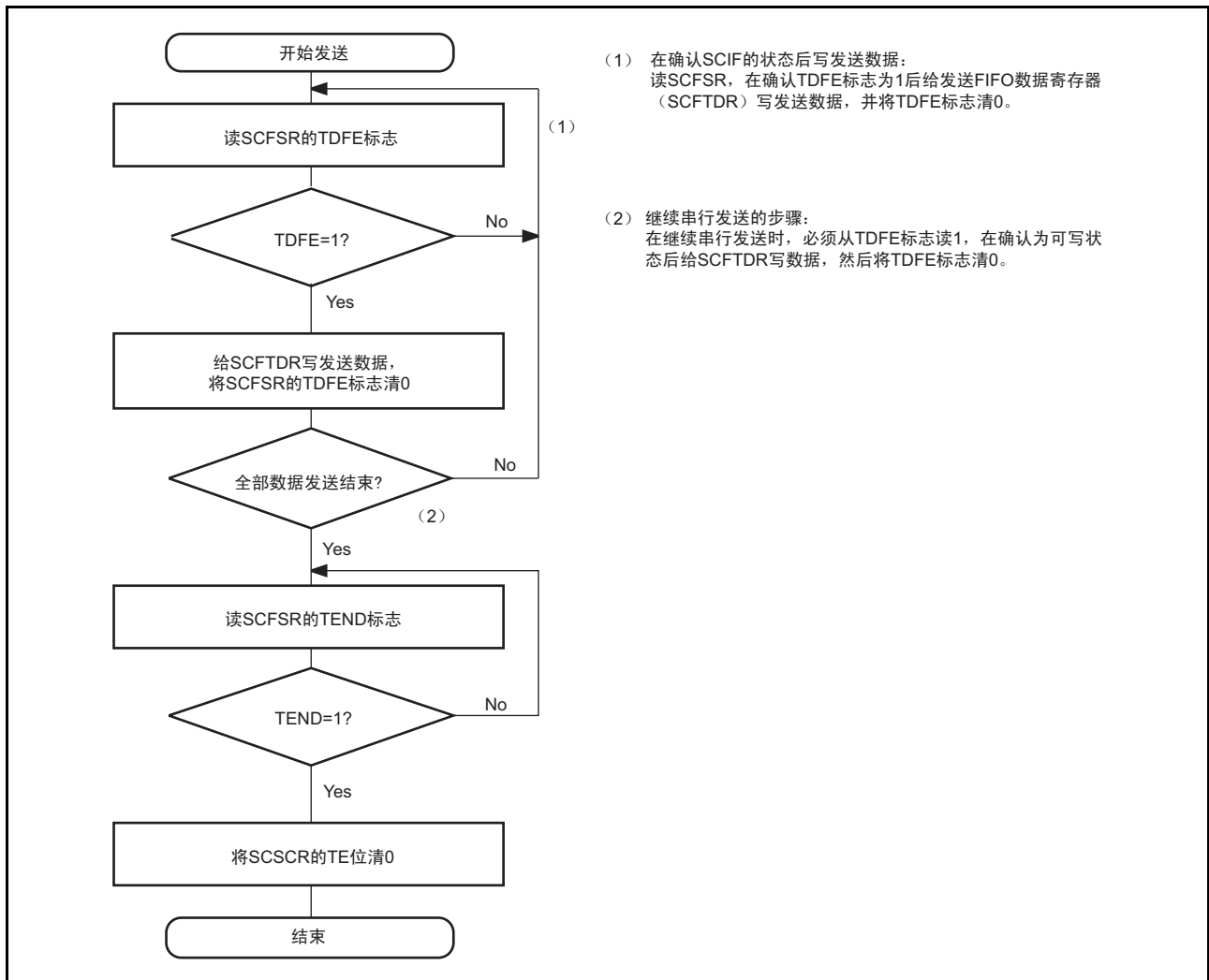


图 15.13 串行发送的流程图例子

在串行发送时, SCIF 的运行如下:

- 当数据被写到发送 FIFO 数据寄存器 (SCFTDR) 时, SCIF 就将数据从 SCFTDR 传送到发送移位寄存器 (SCTSR)。必须在将发送数据写到 SCFTDR 前确认串行状态寄存器 (SCFSR) 的 TDFE 标志是否已被置 1。能写的发送数据字节数为 (16 - 发送触发设定个数)。
- 当将数据从 SCFTDR 传送到 SCTSR 后开始发送时, 就连续进行发送, 直到 SCFTDR 中没有发送数据为止。如果 SCFTDR 的发送数据字节数小于等于 FIFO 控制寄存器 (SCFCR) 设定的发送触发个数, TDFE 标志就被置位。此时, 如果串行控制寄存器 (SCSCR) 的 TIE 位为 1, 就产生发送 FIFO 数据空中断 (TXI) 请求。

当设定为时钟输出模式时, SCIF 输出 8 个同步时钟脉冲。当设定为外部时钟时, 与输入时钟同步输出数据。按 LSB (bit0) ~ MSB (bit7) 的顺序从 TxD 引脚发送串行发送数据。

- SCIF 通过发送最后位的时序来检查 SCFTDR 的发送数据。当有发送数据时, 就将数据从 SCFTDR 传送到 SCTSR, 并开始下一个帧的串行发送; 当没有数据时, 就将 SCFSR 的 TEND 标志置 1, 在发送最后位后, TxD 引脚保持状态。

4. 在串行发送结束后, SCK 引脚固定为高电平。
SCIF 发送的运行例子如图 15.14 所示。

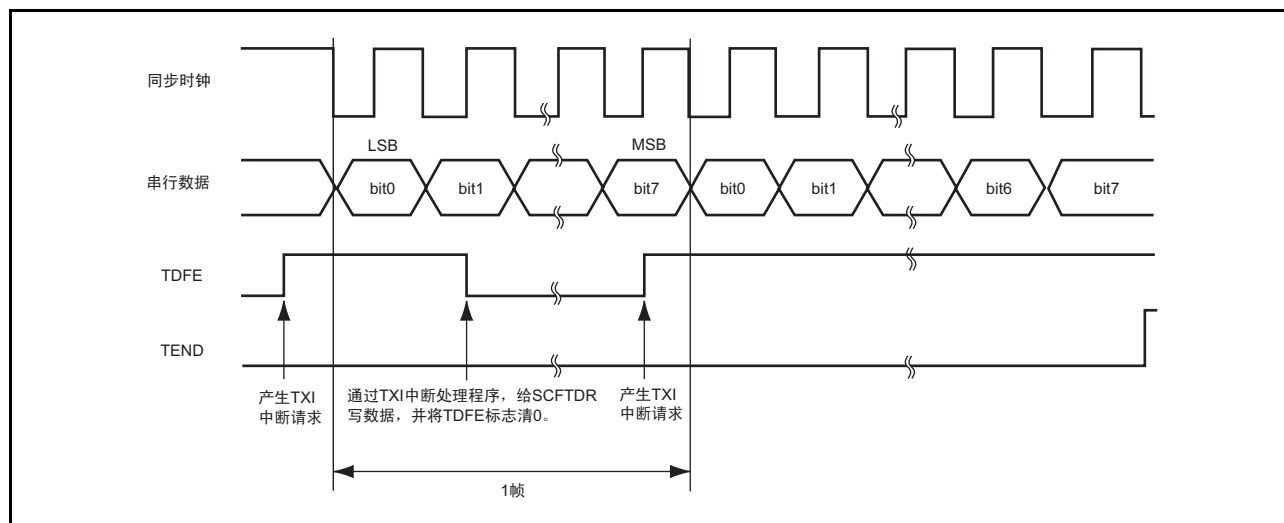


图 15.14 SCIF 发送的运行例子

- 串行数据接收 (时钟同步模式)

串行接收的流程图例子如图 15.15 和图 15.16 所示。

在将 SCIF 设定为能接收的状态后, 必须按以下步骤接收串行数据。

在不对 SCIF 进行初始化而将运行模式从异步模式切换到时钟同步模式时, 必须确认 ORER、PER、FER 的各标志是否已被清 0。

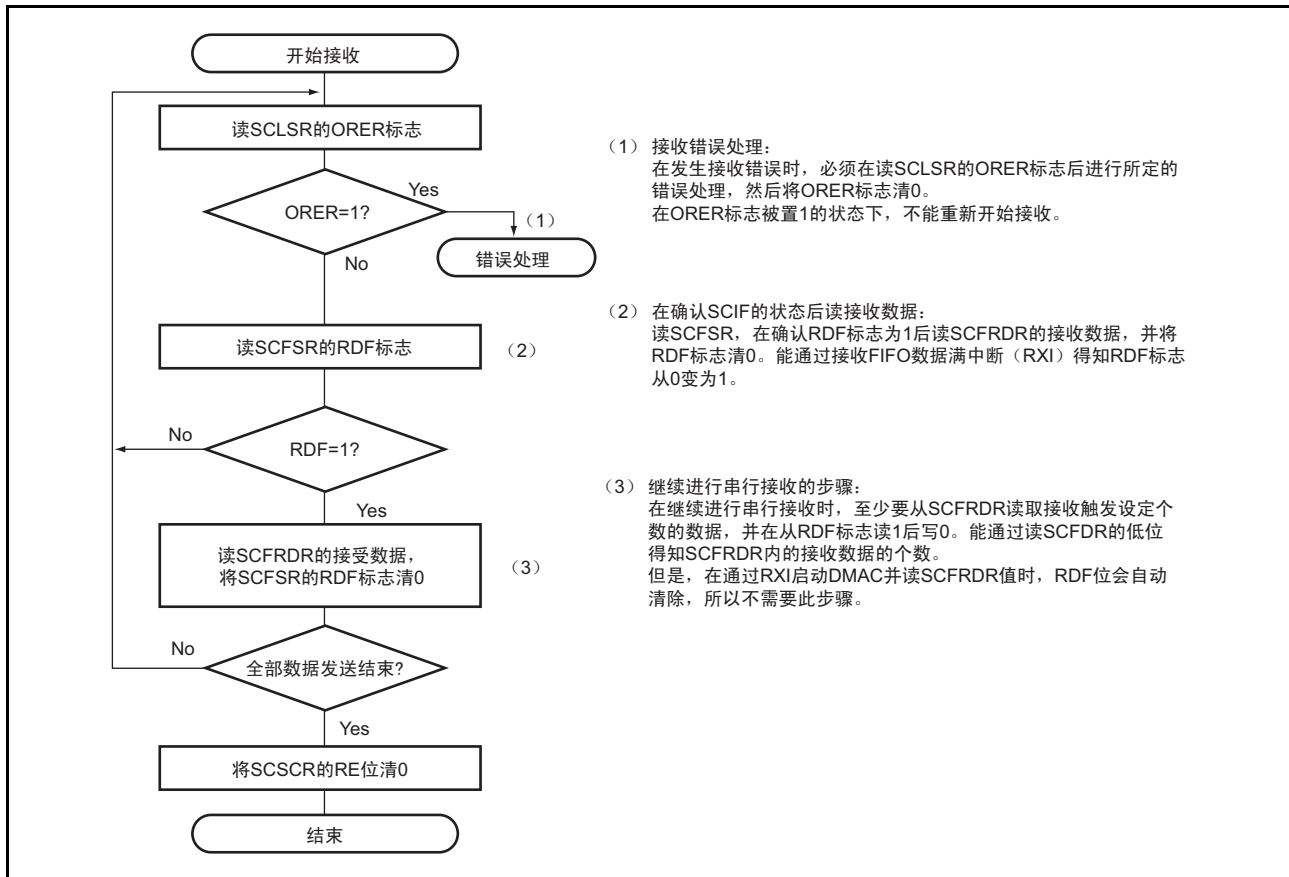


图 15.15 串行接收的流程图例子 (1)

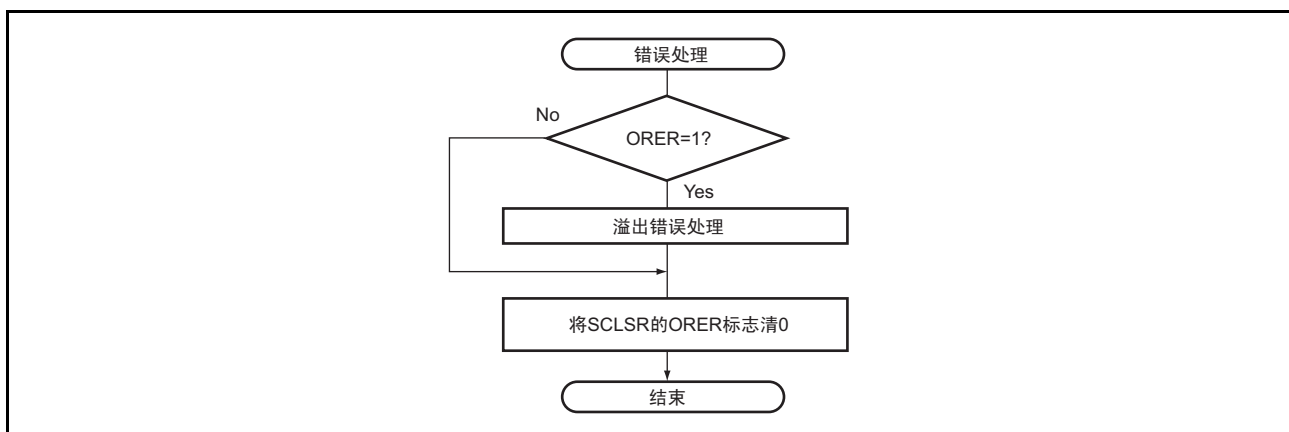


图 15.16 串行接收的流程图例子 (2)

在串行接收时，SCIF 的运行如下：

1. SCIF与同步时钟的输入或者输出同步，开始接收。
2. 按从接收移位寄存器（SCRSR）的LSB到MSB的顺序保存已接收的数据。在接收后，SCIF检查是否处于能将接收数据从SCRSR传送到SCFRDR的状态。如果检查通过，就将RDF标志置1，并将接收数据保存到SCFRDR。如果在错误检查时检测到溢出错误，就不能进行以后的接收。
3. 当RDF标志变为1时，如果串行控制寄存器（SCSCR）的RIE位已被置1，就产生接收FIFO数据满中断（RXI）请求；当ORER标志变为1时，如果SCSCR的RIE位或者REIE位已被置1，就产生中止中断（BRI）请求。

SCIF 接收时的运行例子如图 15.17 所示。

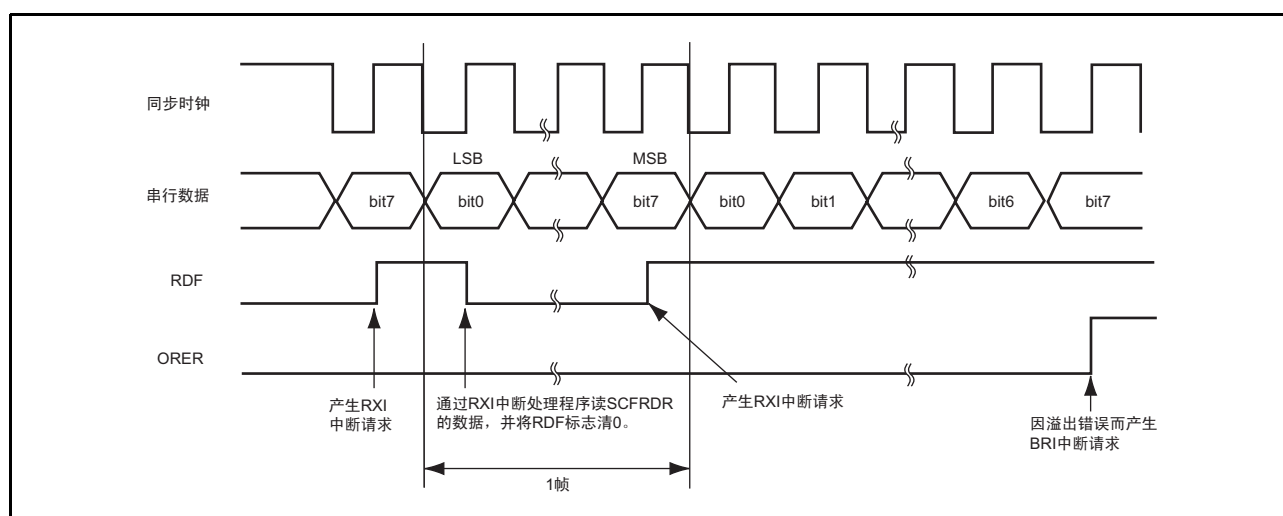


图 15.17 SCIF 接收的运行例子

- 串行数据的同时发送和接收 (时钟同步模式)

同时进行串行发送和接收的流程图例子如图 15.18 所示。

在将 SCIF 设定为能发送和接收的状态后, 必须按以下步骤同时进行串行数据的发送和接收。

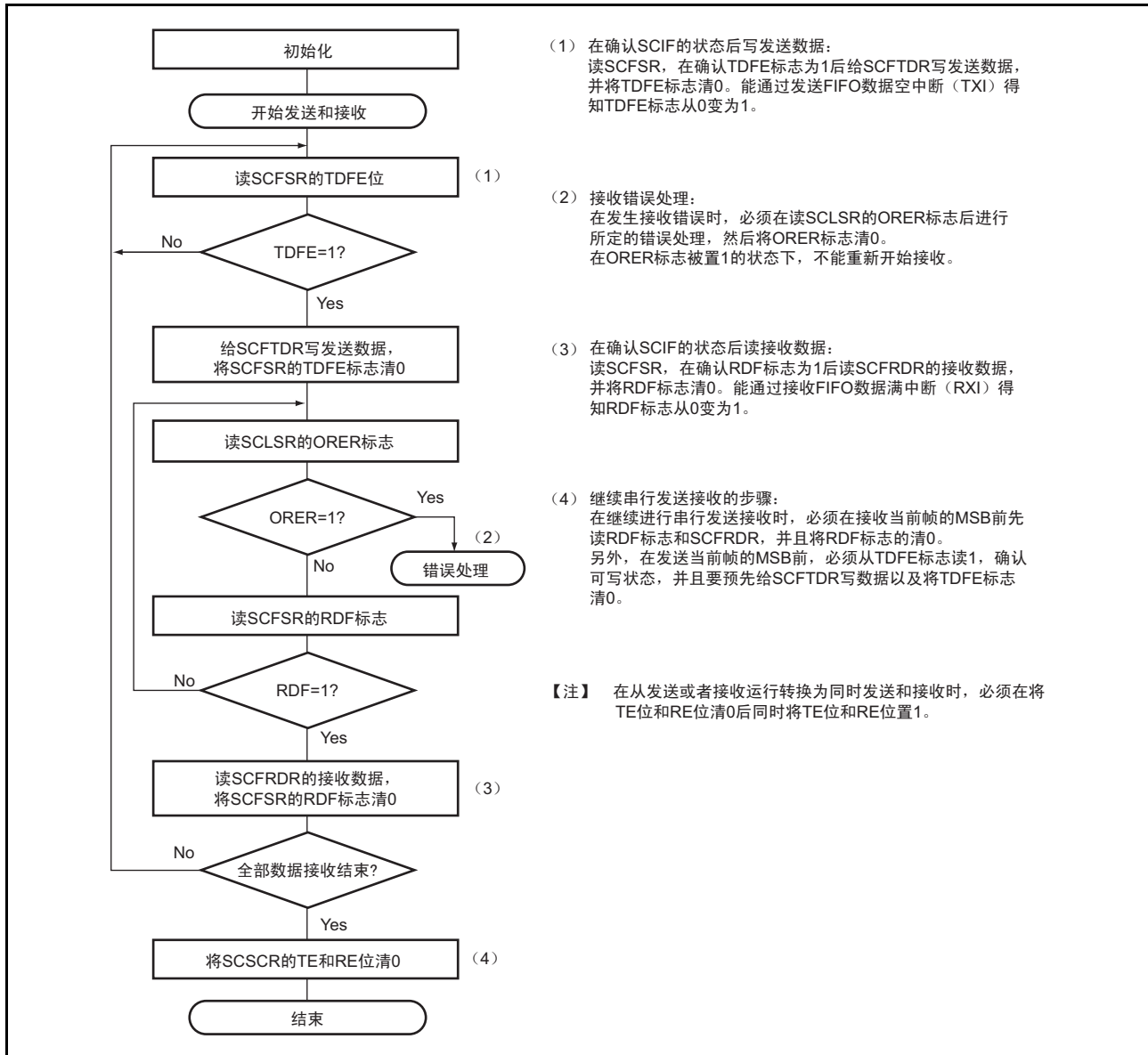


图 15.18 串行数据发送和接收的流程图例子

15.5 SCIF 的中断

SCIF 有发送 FIFO 数据空中断 (TXI) 请求、接收错误中断 (ERI) 请求、接收 FIFO 数据满中断 (RXI) 请求和中止中断 (BRI) 请求 4 种中断源。

各中断源和优先顺序如表 15.13 所示。能通过 SCSCR 的 TIE、RIE 和 REIE 位允许或者禁止各中断源，还能分别独立地将各中断请求发送到中断控制器。

当通过 TIE 位允许 TXI 时，如果串行状态寄存器 (SCFSR) 的 TDFE 标志已被置 1，就产生 TXI 中断请求。能通过此 TXI 中断请求启动 DMAC 进行数据传送。此时不向 CPU 请求中断。

当通过 RIE 位允许 RXI 时，如果 SCFSR 的 RDF 标志或者 DR 标志已被置 1，就产生 RXI 中断请求。能通过此 RXI 中断请求启动 DMAC 进行数据传送。此时不向 CPU 请求中断。但是，通过将 DR 标志置 1 而产生的 RXI 中断请求只在异步模式中有效。

如果将 SCSCR 的 RIE 位置 0 并且 REIE 位置 1，就不产生 RXI 中断请求而只产生 ERI 中断请求。

TXI 表示能写发送数据，RXI 表示 SCFRDR 有接收数据。

表 15.13 SCIF 中断源

中断源	内容	启动 DMAC	复位解除时 优先顺序
BRI	中止 (BRK) 或者溢出 (ORER) 产生的中断	不可	高 ↑ ↓ 低
ERI	接收错误 (ER) 产生的中断	不可	
RXI	接收 FIFO 数据满 (RDF) 或者数据就绪 (DR) 产生的中断	可	
TXI	发送 FIFO 数据空 (TDFE) 产生的中断	可	

15.6 使用时的注意事项

使用 SCIF 时，必须注意以下事项。

15.6.1 SCFTDR 的写操作和 TDFE 标志

当写到发送 FIFO 数据寄存器 (SCFTDR) 的发送数据字节数少于 FIFO 控制寄存器 (SCFCR) 的 TTRG[1:0] 设定的发送触发个数时，串行状态寄存器 (SCFSR) 的 TDFE 标志就被置位。在 TDFE 标志被置位后，能写发送数据，直到 SCFTDR 满为止，进行高效率的连续发送。

当写到 SCFTDR 的数据字节数小于等于发送触发个数时，即使在从 TDFE 标志读 1 后清 0，也会再次被置 1。因此，当写到 SCFTDR 的数据字节数超过发送触发个数时，必须在读 1 后清除 TDFE 标志。

能从 FIFO 数据计数寄存器 (SCFDR) 的高 8 位得知 SCFTDR 的发送数据字节数。

15.6.2 SCFRDR 的读操作和 RDF 标志

当接收 FIFO 数据寄存器 (SCFRDR) 的接收数据字节数大于等于 FIFO 控制寄存器 (SCFCR) 的 RTRG[1:0] 设定的接收触发个数时，串行状态寄存器 (SCFSR) 的 RDF 标志就被置位。在 RDF 标志被置位后，能从 SCFRDR 读到触发个数的接收数据，进行高效率的连续接收。

当 SCFRDR 的数据字节数超过接收触发个数时，即使在从 RDF 标志读 1 后清 0，也会再次被置 1。因此，必须在读到接收数据后，读 RDF 标志的 1 并将 RDF 标志清 0，使接收 FIFO 数据寄存器 (SCFRDR) 内的数据个数少于触发个数。

能从 FIFO 数据计数寄存器 (SCFDR) 的低 8 位得知 SCFRDR 的接收数据字节数。

15.6.3 中止的检测和处理

在检测帧错误 (FER) 时, 能通过直接读取 RxD 引脚的值进行中止的检测。因为在中止时 RxD 引脚的输入都为 0, 所以 FER 标志和奇偶校验错误标志 (PER) 都可能被置位。

当检测到中止时, SCIF 停止给 SCFRDR 传送接收数据, 但是继续接收数据。

15.6.4 中止信号的发送

TxD 引脚的输入 / 输出条件和电平取决于串行端口寄存器 (SCSPTR) 的 SPB2IO 位和 SPB2DR 位, 能使用这些位发送中止信号。

从串行发送的初始化到 TE 位被置 1 (能发送) 的期间, 不能用作 TxD 引脚。在此期间, 标记状态被 SPB2DT 位的值代替, 因此要先将 SPB2IO 和 SPB2DT 位置 1 (输出高电平)。

为了在串行发送时发送中止信号, 必须在将 SPB2DT 位清 0 (指定低电平) 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, 并且 TxD 引脚输出 0。

15.6.5 异步模式的接收数据采样时序和接收容限

SCIF 以 16 倍或者 8 倍位速率的基本时钟频率运行。在接收时, SCIF 通过基本时钟对起始位的下降沿进行采样, 在内部取得同步, 并且在基本时钟的第 8 个或者第 4 个时钟的上升沿将接收数据取到内部。以 16 倍位速率的基本时钟频率运行的时序如图 15.19 所示。

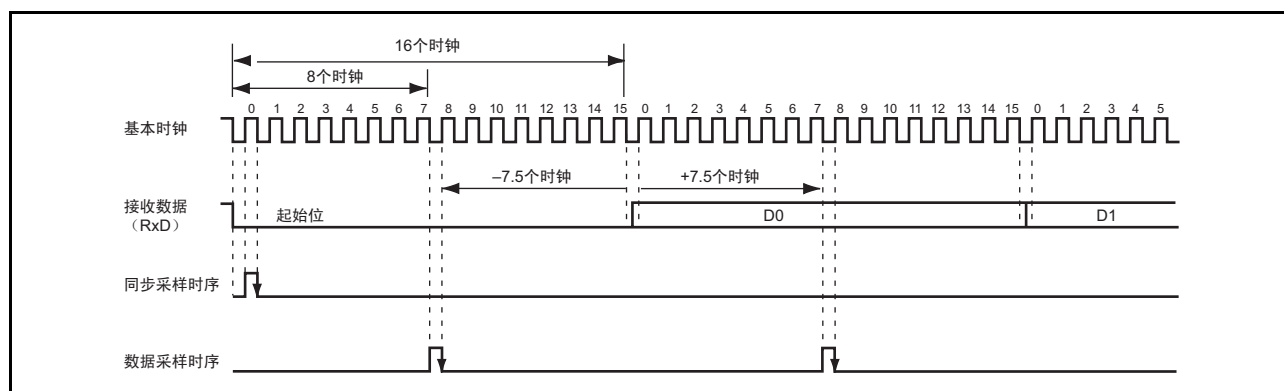


图 15.19 异步模式的接收数据采样时序
(以 16 倍位速率的基本时钟频率运行)

因此, 异步模式的接收容限如算式 (1) 所示:

$$M = \left\{ \left(0.5 - \frac{1}{2N} \right) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1 + F) \right\} \times 100[\%] \quad \dots \text{公式 (1)}$$

M: 接收容限 (%)

N: 时钟频率和位速率的比 (N=16 或者 8)

D: 时钟占空比 (D:0 ~ 1.0)

L: 帧长 (L=9 ~ 12)

F: 时钟频率的偏差绝对值

在算式 (1) 中, 假设 F=0、D=0.5、N=16, 接收容限根据算式 (2) 得出 46.875%。

D=0.5、F=0 时

$$M = (0.5 - 1/(2 \times 16)) \times 100\% = 46.875\% \quad \dots \text{公式 (2)}$$

但是此值毕竟是计算值, 所以系统设计时必须留有 20 ~ 30% 的余地。

15.6.6 异步基本时钟的选择

本 LSI 能通过更改串行扩展模式寄存器 (SCEMR) 的 ABCS 位, 将异步模式中的基本时钟设定为 16 倍或者 8 倍位速率的频率。

必须注意: 从“15.6.5 异步模式的接收数据采样时序和接收容限”的算式 (1) 可知, 如果将基本时钟的频率降低到位速率的 8 倍, 接收容限就会减小。

如果能只通过串行模式寄存器 (SCSMR) 的 CKS[1:0] 和位速率寄存器 (SCBRR) 设定所要的位速率, 就建议将基本时钟设定为 16 倍位速率的频率 (SCEMR 的 ABCS=0)。另外, 如果将内部时钟作为时钟源并且不使用 SCK 引脚, 就能通过将波特率发生器设定为倍速模式 (SCEMR 的 BGDM=1) 使接收容限不受影响而提高位速率。

第 16 章 I²C 总线接口 (IIC)

16.1 特点

I²C 总线接口有以下特点:

- 对应 I²C 总线接口方式
- 对应多主控
- 7 位/10 位地址兼容主控
- 7 位从属地址
- 对应快速模式
- 可变的 SCL 时钟频率

I²C 总线接口的框图如图 16.1 所示。

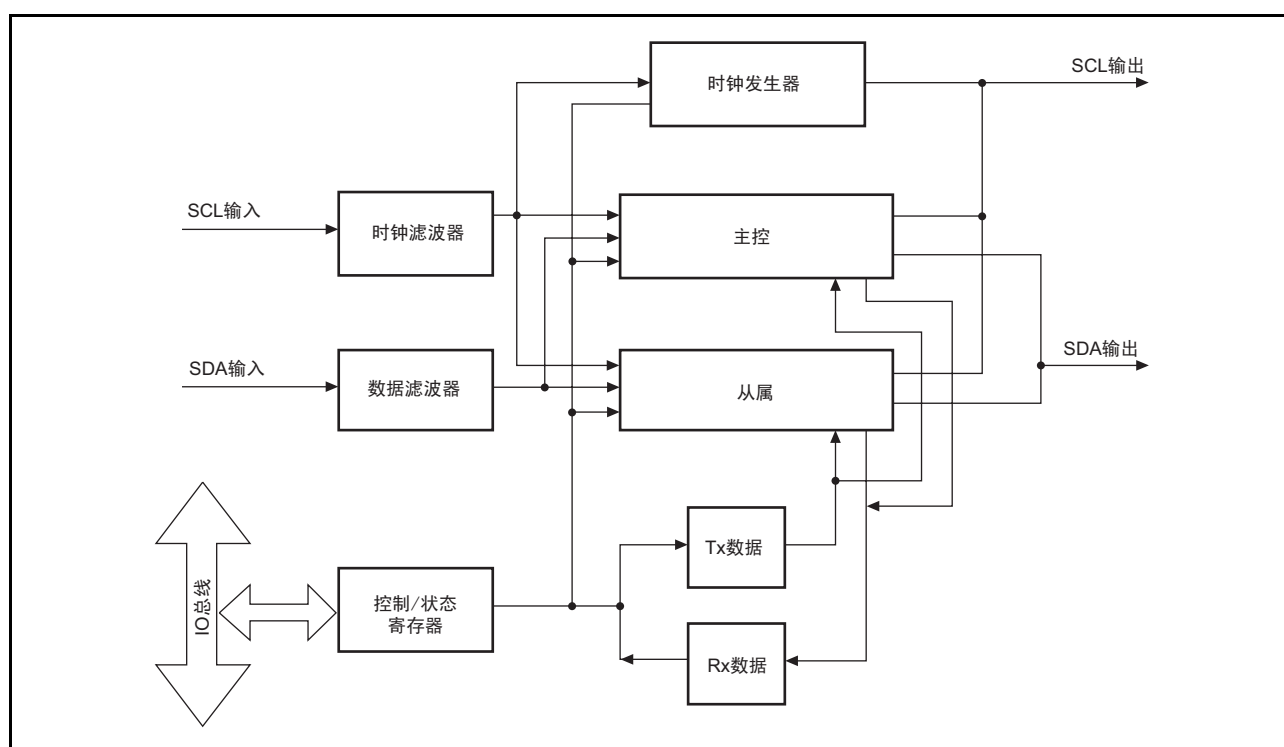


图 16.1 I²C 总线接口的框图

16.2 输入 / 输出引脚

I²C 总线接口使用的引脚如表 16.1 所示。

表 16.1 I²C 总线接口的引脚结构

名称	输入 / 输出	功能
SCL	输入 / 输出	是 I ² C 串行时钟的输入 / 输出引脚 *
SDA	输入 / 输出	是 I ² C 串行数据的输入 / 输出引脚 *

【注】 * I²C 总线的 SCL/SDA 输入 / 输出引脚为漏极开路，是 3.3V I/O。

16.3 寄存器说明

I²C 总线接口的寄存器结构如表 16.2 所示，各处理模式的寄存器状态如表 16.3 所示。

表 16.2 寄存器结构

名称	略称	R/W	P4 区地址	区域 7 地址	存取长度
从属控制寄存器	ICSCR	R/W	H'FFE7 0000	H'1FF7 0000	8
主控控制寄存器	ICMCR	R/W	H'FFE7 0004	H'1FF7 0004	8
从属状态寄存器	ICSSR	R/(W)*1	H'FFE7 0008	H'1FF7 0008	8
主控状态寄存器	ICMSR	R/(W)*2	H'FFE7 000C	H'1FF7 000C	8
从属中断允许寄存器	ICSIER	R/W	H'FFE7 0010	H'1FF7 0010	8
主控中断允许寄存器	ICMIER	R/W	H'FFE7 0014	H'1FF7 0014	8
时钟控制寄存器	ICCCR	R/W	H'FFE7 0018	H'1FF7 0018	8
从属地址寄存器	ICSAR	R/W	H'FFE7 001C	H'1FF7 001C	8
主控地址寄存器	ICMAR	R/W	H'FFE7 0020	H'1FF7 0020	8
接收数据寄存器	ICRXD	R/W	H'FFE7 0024	H'1FF7 0024	8
发送数据寄存器	ICTXD	R/W	H'FFE7 0024	H'1FF7 0024	8

【注】 *1 为了清除标志，只能给 bit4 ~ 0 写 0。

*2 为了清除标志，只能给 bit6 ~ 0 写 0。

表 16.3 各处理模式的寄存器状态

名称	略称	上电复位	睡眠	待机
从属控制寄存器	ICSCR	H'00	保持	保持
主控控制寄存器	ICMCR	H'x0	保持	保持
从属状态寄存器	ICSSR	H'00	保持	保持
主控状态寄存器	ICMSR	H'00	保持	保持
从属中断允许寄存器	ICSIER	H'00	保持	保持
主控中断允许寄存器	ICMIER	H'00	保持	保持
时钟控制寄存器	ICCCR	H'00	保持	保持
从属地址寄存器	ICSAR	H'00	保持	保持
主控地址寄存器	ICMAR	H'00	保持	保持
接收数据寄存器	ICRXD	H'00	保持	保持
发送数据寄存器	ICTXD	H'00	保持	保持

16.3.1 从属控制寄存器 (ICSCR)

位:	7	6	5	4	3	2	1	0
	—	—	—	—	SDBS	SIE	GCAE	FNA
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 只能写 0。
3	SDBS	0	R/W	从属数据缓冲器选择位 选择数据缓冲器。数据缓冲器有双缓冲器和单缓冲器 2 个模式。 当将此位清除时，选择双缓冲器模式。在接受模式时，如果 2 个缓冲器满并且 SDR 未被清除，SCL 就保持低电平；如果清除 SDR，就解除 SCL 的低电平。 当将此位置位时，选择单缓冲器模式。 从接收数据寄存器接收数据包到清除 SDR 的期间，SCL 保持低电平。 0: 双缓冲器模式 1: 单缓冲器模式
2	SIE	0	R/W	从属接口允许位 作为从属接口，需要将此位置位。在此位为 0 时，对从属接口进行复位。 如果将 MIE 置位，就清除此位。
1	GCAE	0	R/W	通用调用应答允许位 对于主控发送的通用调用地址，如果需从从属设备应答，就必须将此位置 1。
0	FNA	0	R/W	强制非应答位 在从属接收模式中，此位的电平作为应答信号被送到发送设备。在接收数据包期间，清除此位；在数据接收结束时，将此位置位。 在从属接收期间，将强制非应答发送到主控。 当从属设备接收完数据包的最后字节数据时，不启动应答 (nack)，与主控进行通信。主控在接受到 nack 后给总线发送停止条件。即使设定此位也不影响从属地址的应答。

16.3.2 从属状态寄存器 (ICSSR)

通过写 0 进行从属状态寄存器的状态位 (bit0 ~ 4) 的清除。除了 GCAR 位和 STM 位以外, 在写 0 前, 各位保持 1。

位:	7	6	5	4	3	2	1	0
	—	GCAR	STM	SSR	SDE	SDT	SDR	SAR
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*

bit	位名	初始值	R/W	说明
7	—	0	R	保留位 只能写 0。
6	GCAR	0	R	通用调用地址接收位 表示从总线接收到的地址是通用调用地址 (00H)。此状态位不产生中断。 如果 SIE 位为 0 或者将 SSR 位置位, 就由硬件自动将此位清 0。SIE 位是从属控制寄存器的 bit2, SSR 位是此寄存器的 bit4。此位的写操作无效。
5	STM	0	R	从属发送模式位 表示当前从属发送模式的读写操作。1 时为写操作, 0 时为读操作。此状态位不产生中断。 如果 SIE 位为 0 或者将 SSR 位置位, 就由硬件自动将此位清 0。SIE 位是从属控制寄存器的 bit2, SSR 位是此寄存器的 bit4。此位的写操作无效。
4	SSR	0	R/W*	从属停止接收位 停止条件已输出到总线。在停止位期间, 此状态位在 SDA 上升后变为有效。
3	SDE	0	R/W*	从属数据空 发送数据已加载到移位寄存器。在开始发送数据字节时, 将 ICTXD 寄存器的内容加载到能向总线发送数据的移位寄存器。此状态位表示数据加载已结束, ICTXD 寄存器进入就绪状态。在最初的数据位前的 SCL 下降沿, 此状态位变为有效。在单缓冲器模式中, 每次给 ICTXD 寄存器写新的数据时, 都需要清除此位。这是因为在此位保持置位的状态下, 由于停止总线, 即使开始从属发送周期, 从属设备仍然将 SCL 保持为低电平。
2	SDT	0	R/W*	从属数据发送位 字节数据已发送到总线。在最后的的数据位期间, 此位在 SCL 下降后变为有效。
1	SDR	0	R/W*	从属数据接收位 已从总线接收到字节数据。接收数据寄存器中有数据, 为就绪状态。在最后的的数据位期间, 此位在 SCL 下降后变为有效。在单缓冲器模式中, 必须在从 ICRXD 寄存器读取数据后清除此位。 如果将 SDBS 置位, 就在接收数据寄存器开始接收数据包到清除 SDR 的期间, SCL 保持低电平。

bit	位名	初始值	R/W	说明
0	SAR	0	R/W*	从属地址接收位 表示从属设备在总线上认识到自己的地址（该地址由从属地址寄存器定义）。在从属控制寄存器的通用调用应答允许位有效时，即使总线上的地址为通用调用地址，也将此状态位置位。此时，能通过此寄存器的 GCAR 位区分通用调用地址。STM 位表示是读（高电平）还是写（低电平）存取。在最后的地址位期间，此状态位在 SCL 下降后变为有效。在通过软件清除此状态位前，从属设备从 ACK 开始将 SCL 保持为低电平。

【注】 * 可读写。如果写 0，此位就被初始化；忽视写 1。

16.3.3 从属中断允许寄存器 (ICSIER)

位:	7	6	5	4	3	2	1	0
	—	—	—	SSRE	SDEE	SDTE	SDRE	SARE
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 只能写 0。
4	SSRE	0	R/W	从属停止接收中断允许位 0: 禁止产生 SSR 中断 1: 允许产生 SSR 中断
3	SDEE	0	R/W	从属数据空中断允许位 0: 禁止产生 SDE 中断 1: 允许产生 SDE 中断
2	SDTE	0	R/W	从属数据发送中断允许位 0: 禁止产生 SDT 中断 1: 允许产生 SDT 中断
1	SDRE	0	R/W	从属数据接收中断允许位 0: 禁止产生 SDR 中断 1: 允许产生 SDR 中断
0	SARE	0	R/W	从属地址接收中断允许位 0: 禁止产生 SAR 中断 1: 允许产生 SAR 中断

16.3.4 从属地址寄存器 (ICSAR)

位:	7	6	5	4	3	2	1	0
	—	SADD0[6:0]						
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
7	—	0	R	保留位 只能写 0。
6 ~ 0	SADD0[6:0]	全 0	R/W	从属地址 这是在 I ² C 总线上分配给从属设备的固有的 7 位地址。在开始发送数据包时，从属接口确认被送来的从属地址的最初 7 位是否与此地址相同。

16.3.5 主控控制寄存器 (ICMCR)

位:	7	6	5	4	3	2	1	0
	MDBS	FSCL	FSDA	OBPC	MIE	TSBE	FSB	ESG
初始值:	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	MDBS	0	R/W	主控数据缓冲器选择位 选择数据缓冲器。数据缓冲器有双缓冲器和单缓冲器 2 种模式。 当将此位清除时，选择双缓冲器模式。在接收模式时，如果 2 个缓冲器满并且 MDR 未被清除，SCL 就保持低电平；如果清除 MDR，就解除 SCL 的低电平。 当将此位置位时，选择单缓冲器模式。 从接收数据寄存器接收数据包到清除 MDR 的期间，SCL 保持低电平。 0: 双缓冲器模式 1: 单缓冲器模式
6	FSCL	—	R/W	强制 SCL 位 控制 SCL 引脚的状态（读取值反映 SCL 引脚的状态）。如果将 OBPC 置位，此位就直接控制总线上的 SCL。 因为此位直接反映 SCL 引脚的值，所以在读周期中此位的内容根据 SCL 的电平发生变化。
5	FSDA	—	R/W	强制 SDA 位 控制 SDA 引脚的状态（读取值反映 I ² C 总线的忙状态）。如果将 OBPC 置位，此位就直接控制总线上的 SDA。 在读周期中，此位的内容表示 I ² C 总线的忙状态。1 表示 I ² C 总线为忙状态，0 表示就绪状态。
4	OBPC	0	R/W	盖写总线引脚控制位 如果将此位置位，此寄存器的 FSDA 和 FSCL 就直接控制 SDA 和 SCL。用于测试。

位	位名	初始值	R/W	说明
3	MIE	0	R/W	主控接口允许位 在此位置位时，主控接口有效。
2	TSBE	0	R/W	开始字节发送允许位 如果将此位置位，就在发送各开始和重新开始条件后，主控给总线发送开始字节 (H'01)。开始字节用于通过 I ² C 总线接口连接低速单片机。
1	FSB	0	R/W	总线强制停止位 如果将此位置位，主控就在当前的传送结束时给总线发送停止条件。如果 ESG 已被置位，主控就立即发送开始条件，并且开始发送新的数据包。在 ESG 没被置位时，主控进入空闲状态。
0	ESG	0	R/W	允许开始生成位 如果将此位置位，主控就开始发送数据包。如果在 ESG 被置位的情况下总线变为空闲状态，主控就给总线发送开始条件，然后发送从属地址。如果在传送过程中 ESG 被置位，主控就在传送中的数据字节结束时并且在发送从属地址前发送重复开始条件。在发送数据包时，必须在发送从属地址后通过软件清除此位。如果不清除此位，就在各个发送结束后发送重复开始条件。

16.3.6 主控状态寄存器 (ICMSR)

通过写 0 进行主控状态寄存器的状态位 (bit0 ~ 6) 的清除。在写 0 前，各位保持 1。

位:	7	6	5	4	3	2	1	0
	—	MNR	MAL	MST	MDE	MDT	MDR	MAT
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

位	位名	初始值	R/W	说明
7	—	0	R	保留位 只能写 0。
6	MNR	0	R/W*	主控 Nack 接收位 在此位被置位时，表示主控接收到 nack (对于地址或者数据发送的应答)。 (nack 是指在应答周期中总线上的 SDA 为高电平)。
5	MAL	0	R/W*	主控仲裁失败位 如果在多主控系统中此位被置位，表示对于总线上的其他主控，主控仲裁失败。 此时，清除 MIE，主控接口无效。
4	MST	0	R/W*	主控停止发送位 在此位被置位时，表示主控给总线发送停止条件。能根据控制寄存器的强制停止位的设定结果或者在从属设备接收数据包期间接收到从属设备的 nack 进行发送。

位	位名	初始值	R/W	说明
3	MDE	0	R/W*	<p>主控数据空位</p> <p>在开始发送数据字节时，将发送数据寄存器的内容加载到能向总线发送数据的移位寄存器。</p> <p>在此位被置位时，表示数据加载已结束，发送数据寄存器进入就绪状态。</p> <p>在主控发送模式中，MDE 的置位与发送从属地址后的 MAT 位的置位同时发生。此时，必须在清除 ICMCR 的 ESG 位后将 MDE 和 MAT 位清除。清除后重新开始发送数据。</p>
2	MDT	0	R/W*	<p>主控数据发送位</p> <p>字节数据已发送到总线上的从属设备。在最后的的数据位期间，此状态位在 SCL 下降后变为有效。</p>
1	MDR	0	R/W*	<p>主控数据接收位</p> <p>已从总线接收到字节数据，并且接收数据寄存器进入就绪状态。在最后的的数据位期间，此状态位在 SCL 下降后变为有效。在单缓冲器模式中，必须在从接收数据寄存器读数据后清除此状态位。</p> <p>如果将 MDBS 置位，就在接收数据寄存器开始接收数据包到清除 MDR 的期间，SCL 保持低电平。</p> <p>在主控接收模式中，MDR 的置位与发送从属地址后的 MAT 位的置位同时发生。此时，必须在清除 ICMCR 的 ESG 位后将 MDR 和 MAT 位清除。清除后开始接收数据。</p>
0	MAT	0	R/W*	<p>主控地址发送位</p> <p>主控发送从属地址的字节数据包。在地址的 ack 位期间，此位在 SCL 下降后变为有效。</p>

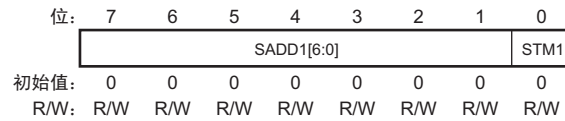
【注】 * 可读写。如果写 0，此位就被初始化；忽视写 1。

16.3.7 主控中断允许寄存器 (ICMIER)

位:	7	6	5	4	3	2	1	0
	—	MNRE	MALE	MSTE	MDEE	MDTE	MDRE	MATE
初始值:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

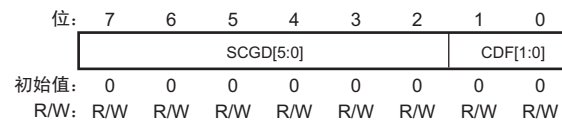
位	位名	初始值	R/W	说明
7	—	0	R	保留位 只能写 0。
6	MNRE	0	R/W	主控 Nack 接收中断允许位 0: 禁止产生 MNR 中断 1: 允许产生 MNR 中断
5	MALE	0	R/W	主控仲裁失去中断允许位 0: 禁止产生 MAL 中断 1: 允许产生 MAL 中断
4	MSTE	0	R/W	主控停止发送中断允许位 0: 禁止产生 MST 中断 1: 允许产生 MST 中断
3	MDEE	0	R/W	主控数据空中断允许位 0: 禁止产生 MDE 中断 1: 允许产生 MDE 中断
2	MDTE	0	R/W	主控数据发送中断允许位 0: 禁止产生 MDT 中断 1: 允许产生 MDT 中断
1	MDRE	0	R/W	主控数据接收中断允许位 0: 禁止产生 MDR 中断 1: 允许产生 MDR 中断
0	MATE	0	R/W	主控地址发送中断允许位 0: 禁止产生 MAT 中断 1: 允许产生 MAT 中断

16.3.8 主控地址寄存器 (ICMAR)



位	位名	初始值	R/W	说明
7 ~ 1	SADD1[6:0]	全 0	R/W	从属地址 是主控要进行通信的从属地址。
0	STM1	0	R/W	从属传送模式位 指定从属设备的运行模式。 将从属运行模式（发送或者接收）设定给与主控发送的从属地址（SADD1）相同的外部从属设备。从属设备根据接收的 STM1 值，由硬件自动设定发送或者接收。 将此位置位时为读操作，清除时为写操作。

16.3.9 时钟控制寄存器 (ICCCR)



位	位名	初始值	R/W	说明
7 ~ 2	SCGD[5:0]	全 0	R/W	SCL 时钟生成分频位 在主控模式运行时，以 SCGD 为分频比，从内部时钟频率生成 SCL 时钟。在从属模式中 SCL 保持低电平（因数据发生上溢而需要停止总线）时，也通过从内部时钟生成的时钟进行运行。因此，无论是主控模式还是从属模式，都需要指定 SCGD。频率的关系如下所示： 公式 2 SCL 速率的计算： $SCL\ freq = IICck / (20 + SCGD * 8)$ IICck: IIC 内部时钟频率 对于 CPU 的速率和 2 种 I ² C 总线速度，CDF 和 SCGD 的推荐值如表 16.4 所示。
1、0	CDF[1:0]	全 0	R/W	时钟分频要素位 I ² C 总线接口模块内的大多数功能块使用的内部时钟是外围时钟分频后的时钟。以 CDF 为分频比，从外围时钟生成 IIC 的内部时钟。 公式 1 IIC 内部时钟频率的计算： $IICck = Pck / (1 + CDF)$ Pck: 外围时钟 对于总线上的 SCL，必须满足 SDA 的准备时间和保持时间的最小值。时钟频率必须保证假信号滤波器能对 50ns（I ² C 快速模式规格定义的）的假信号起作用。 【注】 CDF 的值必须使时钟频率（IICck）低于 20MHz。

表 16.4 CDF 和 SCGD 的推荐值 *

外围时钟频率	100kHz		400kHz	
	CDF	SCGD	CDF	SCGD
50MHz	2	19	2	3
误差	-3.10%		-5.30%	

【注】 * 是对于 SCL 速率的推荐值。

16.3.10 接收 / 发送数据寄存器 (ICRXD/ICTXD)

在读写此寄存器时，存取其他的物理寄存器。在发送数据时，使用 TXD 加载移位寄存器的数据。在从 I²C 总线将数据接收到移位寄存器时，被数据加载到 RXD。

接收数据寄存器 (ICRXD)

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

位	位名	初始值	R/W	说明
7 ~ 0	RXD[7:0]	全 0	R	读接收数据 是主控或者从属设备接收的数据。

发送数据寄存器 (ICTXD)

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

位	位名	初始值	R/W	说明
7 ~ 0	TXD[7:0]	全 0	W	写发送数据 是主控或者从属设备发送的数据。

16.4 运行说明

16.4.1 数据和时钟滤波器

此功能模块用于消除 I²C 总线输入信号中的假信号。消除假信号的宽度最大为 1 个时钟（有关内部时钟频率的详细内容，请参照时钟控制寄存器）。这虽然是快速模式 I²C 总线的速率（400kHz）规格，但是并不违反低速模式的规格。

此功能块对内部时钟进行总线信号的再同步。

16.4.2 时钟发生器

时钟发生器有 2 个功能。第 1 个是根据主控或者从属接口的命令，生成 SCL I²C 总线时钟。第 2 个是控制用于滤波器、主控和从属接口的内部时钟。此时钟用作滤波器、主控或者从属接口的寄存器时钟允许信号。

16.4.3 主控 / 从属接口

此接口独立并行工作。主控接口控制 I²C 总线上的地址和数据传送；从属接口监视 I²C 总线，如果在总线上出现了设定的地址，就进行数据传送。此接口独立地与控制 / 状态寄存器进行通信。只有 1 个中断从 I²C 总线接口模块输出。中断源是主控或者从属设备。

16.4.4 软件状态的联锁

为了对 I²C 总线接口模块尽可能简化软件界面，在主控和从属接口中准备了几个连动状态。此状态位如下所示：

(1) MDR 和 SDR

如果接收数据，就将 MDR 和 SDR 置位。必须在读接收数据寄存器后清除其状态。如果在 MDR 和 SDR 被置位的状态下接收数据，硬件就认为接收寄存器中还有未读的数据，自动将 SCL 保持为低电平，暂停数据传送。对于这种情况，能通过在读接收数据后清除其状态来重新开始传送。

在连续接收数据时，必须在读接收数据寄存器后清除 MDR 和 SDR 的状态。

(2) MDE 和 SDE

即使从属设备或者主控处于给 I²C 总线发送数据（发送数据寄存器的数据）的阶段，如果 MDE 和 SDE 状态位已被置位，SCL 也必须在清除 MDE 和 SDE 前保持低电平。在 MDE 和 SDE 被置位时，表示保存在当前发送数据寄存器中的数据已发送到 I²C 总线。

在给发送数据寄存器（处于能发送下一个字节数据的状态）写数据时，需要通过软件清除此状态位。但是，在给总线发送最初的字节数据时，不需要清除。

(3) MAL

在主控仲裁失败时，将主控状态寄存器的 MAL 位置位，主控控制寄存器的 MIE 位清 0。此时，主控模式无效，I²C 总线接口变为从属模式。当重新开始主控运行时，就在清除 MAL 位后从主控开始数据传送。

(4) SAR

从属接口认为在 I²C 总线上出现了自己的地址时，将 SAR 状态位置位。此时，从属接口在清除 SAR 状态位前将 SCL 保持为低电平。

在对总线进行从属数据发送时，尤其重要。从属接口从发送数据寄存器进行数据发送。软件通过给发送数据寄存器写需要的数据，然后清除 SAR 位，对 SAR 状态进行应答。因此，从属接口能继续存取。

在从属接口接收数据时，软件可能会从接收数据寄存器读取上一次存取时加载的数据，新的存取可能会盖写接收数据寄存器中存在的有效数据。但是，通过使用 SAR 状态位能避免这种情况。如果软件在读完接收数据寄存器中的所有数据后清除 SAR 位（SAR 被置位的情况），就能避免接收数据寄存器的盖写。

16.4.5 I²C 总线数据格式

I²C 总线接口的总线时序如图 16.2 所示，图 16.2 中的符号含义如表 16.5 所示。

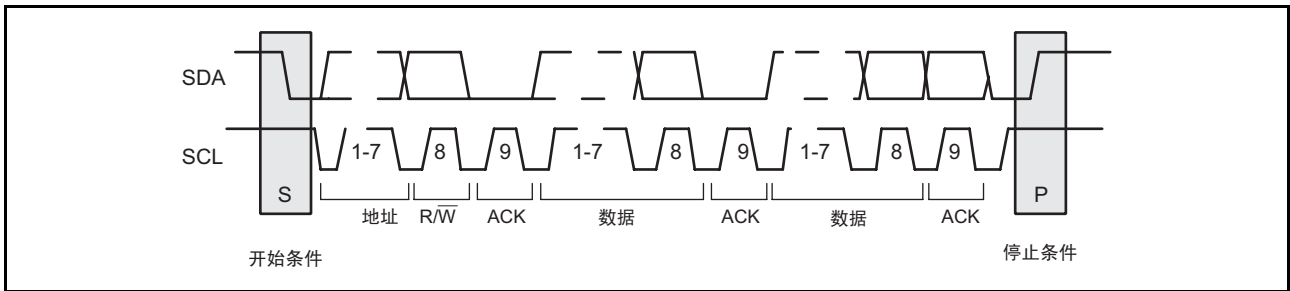


图 16.2 I²C 总线时序

表 16.5 I²C 总线数据格式的符号说明

符号	说明
S	表示开始条件。在 SCL 为高电平的状态下，主控设备将 SDA 从高电平变为低电平。
SLA	表示从属地址。主控选择从属设备。
R/W	表示发送 / 接收方向。为 1 时，数据从从属设备传送到主控设备；为 0 时，数据从主控设备传送到从属设备。
A	表示数据应答。接收设备将 SDA 变为低电平。在发送模式中，从属设备返回数据应答。
DATA	表示发送 / 接收数据。数据长度为 8 位，从 MSB 开始传送。
P	表示停止状态。在 SCL 为高电平的状态下，主控设备将 SDA 从低电平变为高电平。

16.4.6 7 位地址格式

主控到从属设备的数据传送格式（主控数据发送格式）如图 16.3 所示，主控设备读取从属设备的第 2 个及其之后的字节数据的数据传送格式（主控数据接收格式）如图 16.4 所示。

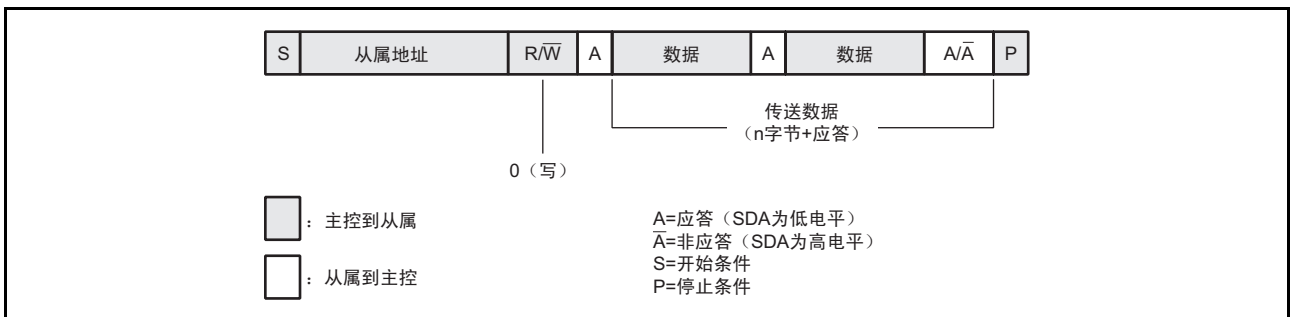


图 16.3 主控数据发送格式

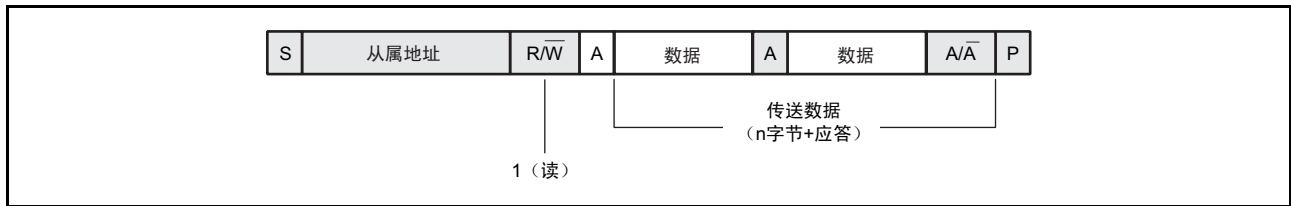


图 16.4 主控数据接收格式

在 1 次传送中改变数据传送方向的复合传送格式如图 16.5 所示。如果在第 1 次传送时改变方向，就发送重复开始条件 (Sr)、从属地址和 R/W。此时，R/W 设定为第 1 次传送方向的反方向。如果主控控制寄存器的允许开始生成位已被置位，就在发送或者接收的最后，主控设备发送重复开始条件。

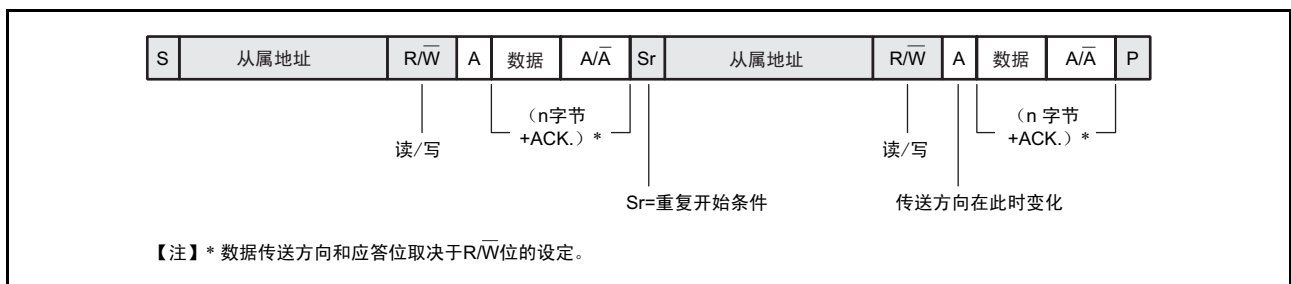


图 16.5 主控传送的复合传送格式

【注】* 重复开始条件：在 SDL 为高电平且 SDA 为低电平时，开始传送。

16.4.7 10 位地址格式

对主控模式支持的 10 位地址传送格式进行说明。

此格式与 7 位地址传送格式相同，有 3 种传送方法。

数据传送格式如图 16.6 所示。在第 1 次传送条件 (S) 后，以 1 个字节输出主控地址寄存器设定的值；以第 2 个字节传送作为从属地址的发送数据寄存器 (TXD) 的设定值；第 3 个字节及其以后的数据传送与 7 位地址数据发送方法相同。

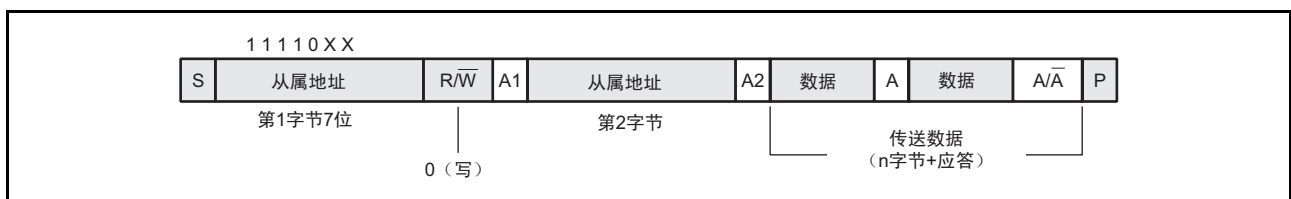


图 16.6 10 位地址的数据发送格式

数据接收格式如图 16.7 所示。

在数据接收格式中，2 个字节的地址发送与前面所述的数据发送相同。然后，发送重复开始条件 (Sr) 和地址寄存器的设定值。此时，STM1 必须为 1 (接收模式)。数据传送与 7 位地址接收格式相同。

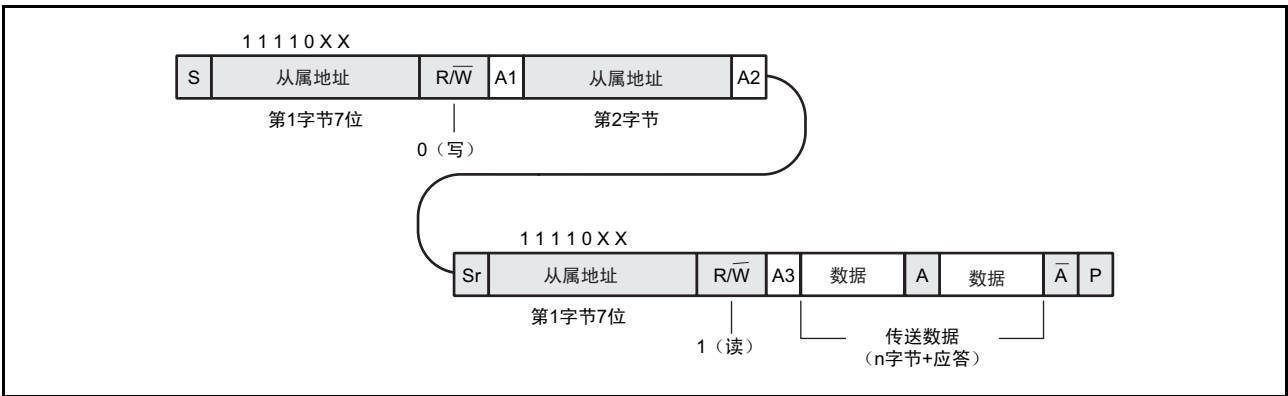


图 16.7 10 位地址的数据接收格式

数据发送 / 接收复合格式如图 16.8 所示。

在数据发送 / 接收复合格式中，在发送最初的 2 个字节的地址后发送数据。然后，发送重复开始条件 (Sr) (代替停止条件 (P))。发送 Sr 后的步骤与通常的数据接收格式相同。

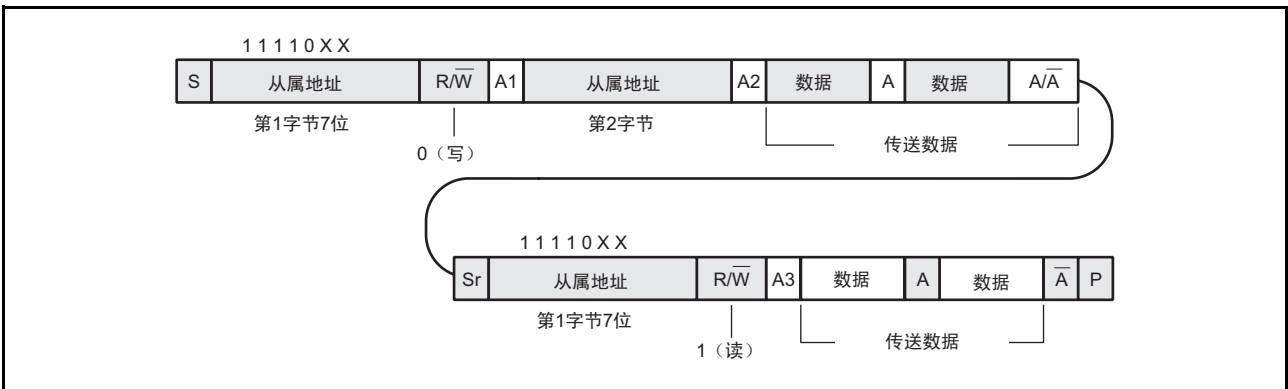


图 16.8 10 位地址数据发送 / 接收复合格式

16.4.8 主控发送的运行

主控发送模式中的发送步骤和运行如下所示，主控发送模式的时序图如图 16.9 所示。将主控控制寄存器的 MD_{BS} 位置位，使 IIC 以单缓冲器模式运行。

1. 在初始状态下，必须根据从属地址、发送数据和发送速度，设定时钟控制寄存器和主控中断允许寄存器。即使在使用主控模式时也需要从属模式，因此必须给从属地址寄存器设定设备地址。
不能在运行过程中更改主控控制寄存器的 MD_{BS} 和从属控制寄存器的 SD_{BS}，否则可能会发生误动作。
2. 必须监视主控控制寄存器的 FSDA 位，确认此位为低电平状态。此状态表示其他的 I²C 设备没有使用总线。在确认后，将主控控制寄存器的 MIE 位和 ESG 位置位，开始主控发送。
3. 在送出发送开始条件、从属地址和数据发送方向的信号后，通过主控状态寄存器的 MAT 位和 MDE 位，在图 16.9 (1) 所示的时序产生中断。然后，必须清除 ESG。为了暂停数据发送，主控设备在清除 MDE 位前将 SCL 保持为低电平。
4. 通过 SAR，在图 16.9 (3) 所示的时序产生中断。如果从属设备的 IRQ 处理延迟，为了暂停数据发送，从属设备就延迟 SCL 的期间 (图 16.9 (7) 所示的时序)。从属设备在第 9 个时钟将 SDA 置为低电平，并且返回 ACK。
5. 数据传送是 8 位加上 1 个 ACK 位，即以 9 位为单位进行的。通过 MDE (bit3)，在数据传送前的第 9 个时钟 (图 16.9 (2) 所示的时序) 产生中断；通过 MDT (bit2)，在 1 个字节数据传送后的第 8 个时钟 (图 16.9 (4) 所示的时序) 产生中断，必须在设定传送数据后清除 MDE；通过从属设备的 SDR (从属接收数据)，在第 8 个时钟 (图 16.9 (6) 所示的时序) 产生中断，必须在从属设备读接收数据后清除 SDR。如果该处理延迟，为了暂停数据发送，从属设备就延长 SCL 的期间 (图 16.9 (8) 所示的时序)。
6. 为了结束数据传送，通过主控状态寄存器的 MNR (bit6)，在从属设备的 ACK 为 1 (Nack) 时的第 9 个时钟产生中断 (图 16.9 (5) 所示的时序)。主控设备接受此 Nack，并且输出数据的传送结束条件。在主控设备结束数据发送时，必须在将主控控制寄存器的 FSB (bit1) 置位后输出暂停条件。如果发送或者接收完字节数据的最终位，IIC 模块就取 FSB 的值，进入停止状态。因此，为了在预先决定的字节数的数据传送后结束通信，需要在开始传送最后的字节数据前设定 FSB。
7. 需要在传送最后的字节数据前设定 FSB。在主控发送模式中设定最后的字节数据后，通过中断或者查询进行 MST (主控的停止发送) 的检查，同时检查 MNR (主控的 NACK 接收)，如果 NACK 被返回，为了重新发送最后的字节数据，需要进入错误程序。

在时钟下降后生成图 16.9 的时序 (1) ~ (6)。

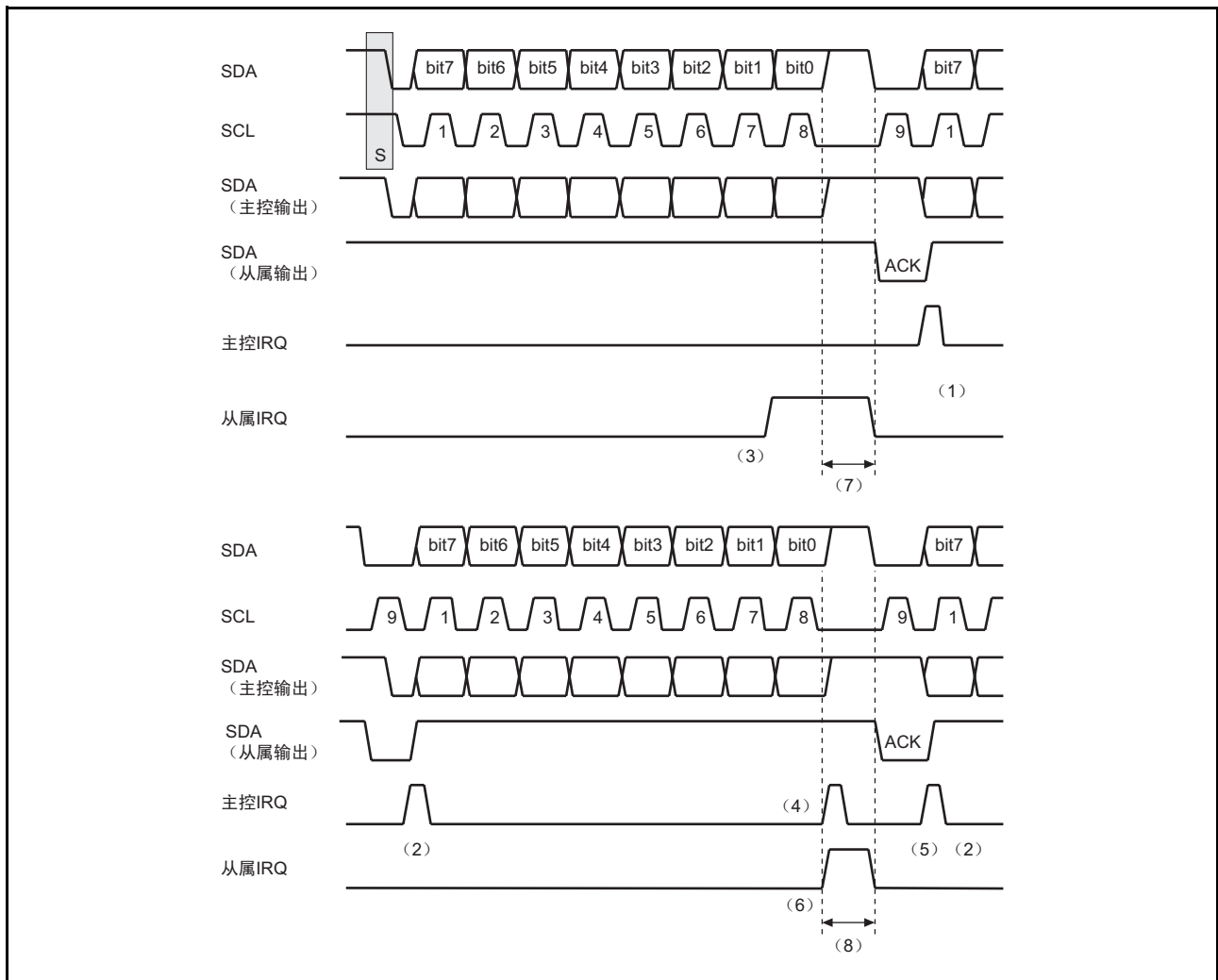


图 16.9 数据传送模式的运行时序

16.4.9 主控接收的运行

主控接收模式中的数据接收步骤和运行如下所示，主控数据接收模式的时序图如图 16.10 所示。将主控控制寄存器的 MDBS 位置位，使 IIC 以单缓冲器模式运行。

1. 在主控数据接收模式中，有关从属地址和数据传送方向的 1 字节信号的工作方式与主控数据传送模式相同。但是，此时的数据传送方向必须选择 1（接收）。
2. 通过数据传送方向的信号，从属设备自动进入数据传送模式，与主控设备的 SCL 时钟同步，发送 1 字节数据。主控设备在第 8 个时钟（图 16.10（2）所示的时序）生成 MDR（bit1）的中断。必须在主控设备读接收数据后清除 MDR 位。如果该中断处理延迟，为了暂停数据发送，主控设备就延长 SCL 期间（图 16.10（3）所示的时序）。
3. 通过表示 1 字节数据传送结束的 SDT（bit2），从属设备在第 8 个时钟（图 16.10（2）所示的时序）产生中断；通过表示数据空的 SDE（bit3），在第 9 个时钟（图 16.10（1）所示的时序）产生中断。必须在给 TXD 写从属发送数据后清除 SDE。

4. 为了结束传送，必须在将主控设备的主控控制寄存器的FSB (bit1) 置位后输出暂停条件。如果发送或者接收完字节数据的最终位，IIC 模块就取FSB 的值，进入停止状态。因此，为了在预先决定的字节数的数据传送后结束通信，需要在开始传送最后的字节数据前设定FSB。在接收完最后的字节数据后，主控接收器结束接收事务，如果最后的字节数据不正确，协议层就通知从属发送方需要重新发送。

在时钟下降后生成图 16.10 的时序 (1) ~ (3)。

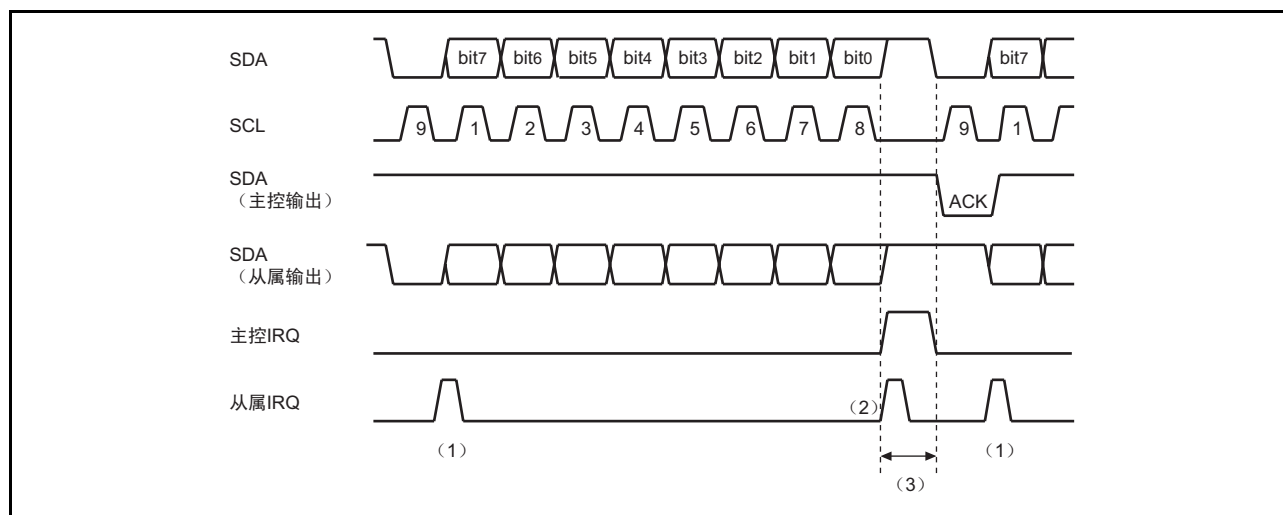


图 16.10 数据接收模式的运行时序

16.5 程序例子

16.5.1 主控发送

为了给 I²C 总线发送数据包，必须按以下步骤设定主控接口：

(1) 时钟控制寄存器的加载

1. 给 SCL 时钟分频 (SCGD) 设定 H'03。
(SCL 频率为 400kHz)
2. 给时钟分频比设定 H'2。
(外围时钟为 50MHz, I²C 内部时钟 IICck 为 16.7MHz)

(2) 主控控制寄存器、第 1 个字节数据和地址的加载

1. 给主控地址寄存器设定要存取的从属地址和 STM1 位 (写模式为 0)。
2. 给发送数据寄存器设定要发送的第 1 个字节数据。
3. 给主控寄存器设定 H'89。
(MDBS=1、MIE=1、ESG=1)

(3) 地址输出的等待

1. 等待主控设备的事件发生 (通过主控状态寄存器的 MAT 位和 MDE 位产生中断)。
2. 给主控控制寄存器设定 H'88 (为了暂停数据发送, 主控设备在清除 MDE 位前将 SCL 保持为低电平)。
在只发送 1 字节数据时, 给主控控制寄存器设定 H'8A (停止生成有效)。
从而在发送 1 字节数据后立即生成停止。
3. 清除 MAT 位。

(4) 字节数据的发送监视

1. 等待主控设备的事件发生 (主控状态寄存器的 MDE 位)。
2. 给发送数据寄存器设定下一个字节数据。
3. 清除 MDE 位。
在设定最后的字节数据后清除 MDE。在开始发送最后的字节数据后生成 MDE。必须在清除 MDE 前给主控控制寄存器设定 H'8A (将强制停止位置位)。

(5) 发送结束的等待

1. 等待主控设备的事件发生 (主控状态寄存器的 MST 位)。
2. 在确认 MNR (Master NACK received) 后清除 MST。

16.5.2 主控接收

为了接受 I²C 总线数据包，必须按以下步骤设定主控接口：

(1) 时钟控制寄存器的加载

1. 给 SCL 时钟分频 (SCGD) 设定 H'03。
(SCL 频率为 400kHz)
2. 给时钟分频比 (CDF) 设定 H'2。
(外围时钟为 50MHz, IIC 内部时钟 IICck 为 16.7MHz)

(2) 主控控制寄存器和地址的加载

1. 给主控地址寄存器设定要存取的从属地址和 STM1 位 (读模式为 1)。
2. 给主控控制寄存器设定 H'89。
(MDBS=1、MIE=1、ESG=1)

(3) 地址输出的等待

1. 等待主控设备的事件发生 (通过主控状态寄存器的 MAT 位和 MDR 位产生中断)。
2. 给主控控制寄存器设定 H'88 (为了暂停数据接收, 主控设备在清除 MDR 位前将 SCL 保持为低电平。)
在只发送 1 字节数据时, 给主控控制寄存器设定 H'8A (停止生成有效)。
从而在接收 1 字节数据时立即生成停止。
3. 清除 MAT 位。

(4) 字节数据的接收监视

1. 等待主控设备的事件发生 (主控状态寄存器的 MDR 位)。
2. 从接收数据寄存器读数据。
如果下一个字节数据是从属设备发送的 (最终-1) 字节的数据, 就产生最终-1 字节的接收中断 (MDR 中断)。
3. 给主控控制寄存器设定 8Ah。
(将强制停止位置位)
4. 清除 MDR 位。

(5) 接收结束的等待

1. 最后字节数据的接收中断 (MDR) 的处理: 读数据, 清除 MDR。
2. 等待主控设备的事件发生 (主控状态寄存器的 MST 位)。
3. 清除 MST。

16.5.3 主控发送 - 重新开始 - 主控接收

为了给 I²C 总线发送数据包、发送重新开始条件以及读取从属设备的数据，必须按以下步骤设定主控接口：

(1) 时钟控制寄存器的加载

1. 给 SCL 时钟分频 (SCGD) 设定 H'03。
(SCL 频率为 400kHz)
2. 给时钟分频比 (CDF) 设定 H'2。
(外围时钟为 50MHz, IIC 内部时钟 IICck 为 16.7MHz)

(2) 主控控制寄存器和地址的加载

1. 给主控地址寄存器设定要存取的从属地址和 STM1 位 (写模式为 0)。
2. 给主控控制寄存器设定 H'89。
(MDBS=1、MIE=1、ESG=1)

(3) 地址输出的等待

1. 等待主控设备的事件发生 (通过主控状态寄存器的 MAT 位和 MDE 位产生中断)。
2. 给主控地址寄存器设定要存取的从属地址和 STM1 位 (读模式为 1)。
如果主控控制寄存器的允许开始生成位还处于置位状态，主控设备就在发送的最后重新发送开始条件。
由于加载了新的地址，所以能改变总线的方向。
3. 清除 MAT 位。

(4) 地址输出的等待

1. 等待主控设备的事件发生 (通过主控状态寄存器的 MAT 位和 MDR 位产生中断)。
2. 给主控控制寄存器设定 H'88。
(为了暂停数据接收，主控设备在清除 MDR 位前将 IIC_SCL 保持为低电平。)
3. 清除 MAT 位。

(5) 字节数据的接收监视

1. 等待主控设备的事件发生 (主控状态寄存器的 MDR 位)。
从接收数据寄存器读数据。
如果下一个字节数据是从属设备发送的 (最终-1) 字节的数据时，就产生最终-1 字节的接收中断 (MDR 中断)。
2. 给主控控制寄存器设定 H'8A。
(将强制停止位置位)
3. 清除 MDR 位。

(6) 接收结束的等待

1. 最后字节数据的接收中断 (MDR) 处理：读数据，清除 MDR。
2. 等待主控设备的事件发生 (主控状态寄存器的 MST 位)。
3. 清除 MST 位。

第 17 章 ATAPI 接口 (ATAPI)

ATAPI 控制器 (ATAPI) 提供 ATA 和 ATAPI 的物理接口, 该设备支持 ATA 任务命令和 ATAPI 包命令。

17.1 特点

- 支持主通道
- 支持主控/从属设备
- 3.3V I/O 接口
- 支持 PIO 模式 0~4、多字 DMA 模式 0~2 和超级 DMA 模式 0~4
- 支持描述符模式

ATAPI 的框图结构如图 17.1 所示。

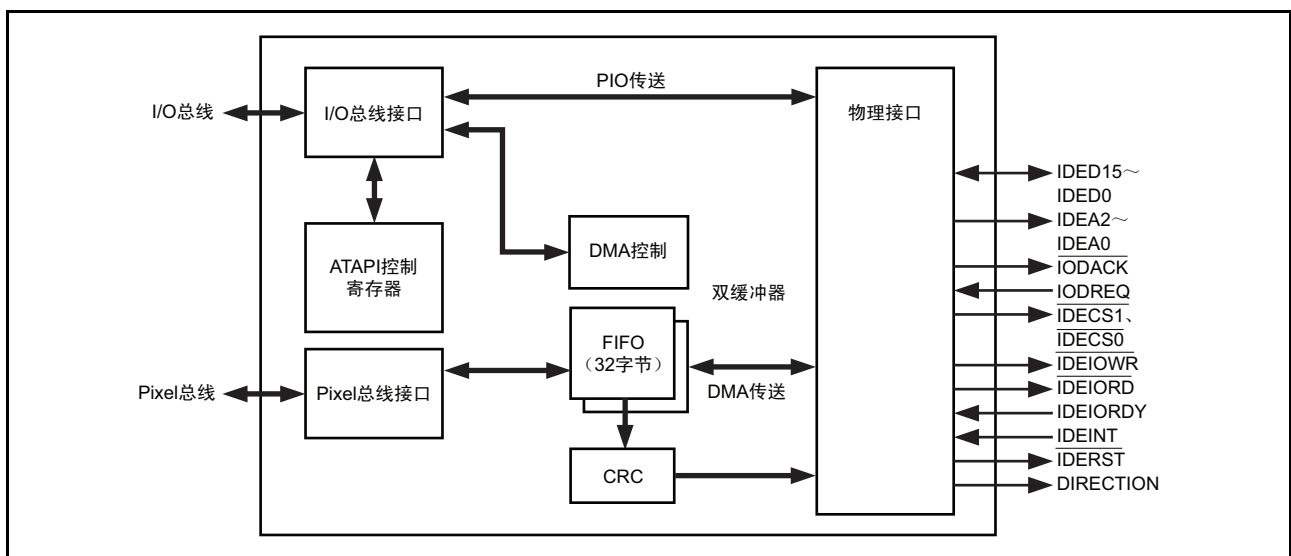


图 17.1 ATAPI 框图

17.2 输入 / 输出引脚

ATAPI 控制器的引脚结构如表 17.1 所示。

ATAPI 控制器的输入 / 输出引脚除了一般的引脚群以外，还有镜像引脚群。无论哪个引脚群一般都进行相同的输入 / 输出操作。通过 GPIO 引脚选择寄存器选择 ATAPI 控制器的引脚。不能混合使用 2 个引脚群。

表 17.1 ATAPI 的引脚结构

信号		(ATAPI 规格)	功能	输入 / 输出
正常输入 / 输出	镜像输入 / 输出			
IDED15 ~ IDED0	IDED15_M ~ IDED0_M	(DD (15:0))	双向数据总线	输入 / 输出
IDEA2 ~ IDEA0	IDEA2_M ~ IDEA0_M	(DA (2:0))	地址总线	输出
IODACK	IODACK_M	(DMACK)	主通道 DMA 应答 (低电平有效)	输出
IODREQ	IODREQ_M	(DMARQ)	主通道 DMA 请求 (高电平有效)	输入
$\overline{\text{IDECS1}}$ 、 $\overline{\text{IDECS0}}$	$\overline{\text{IDECS1_M}}$ 、 $\overline{\text{IDECS0_M}}$	($\overline{\text{CS0}}$ 、 $\overline{\text{CS1}}$)	主通道片选 (低电平有效)	输出
$\overline{\text{IDEIOWR}}$	$\overline{\text{IDEIOWR_M}}$	($\overline{\text{DIOW}}$ 、STOP)	主通道写磁盘 (低电平有效)	输出
$\overline{\text{IDEIORD}}$	$\overline{\text{IDEIORD_M}}$	($\overline{\text{DIOR}}$ 、 $\overline{\text{HDMARDY}}$ 、 $\overline{\text{HSTROBE}}$)	主通道读磁盘 (低电平有效)	输出
IDEIORDY	IDEIORDY_M	(IORDY、 $\overline{\text{DDMARDY}}$ 、 $\overline{\text{DSTROBE}}$)	主通道就绪信号 (高电平有效)	输入
IDEINT	IDEINT_M	(INTRQ)	主通道中断请求 * (高电平有效)	输入
$\overline{\text{IDERST}}$	$\overline{\text{IDERST_M}}$	(RESET)	主通道 ATAPI 设备复位 (低电平有效)	输出
DIRECTION	DIRECTION_M	—	外部电平移位器方向信号 (写设备时为 0)	输出

【注】 * ATAPI 控制器将 ATAPI 设备的中断信号视为电平触发输入。

17.3 寄存器说明

以下的 ATAPI 控制寄存器分配到 SH 寄存器映像空间。

表 17.2 ATA 任务文件寄存器映像
(下列寄存器分配到 ATAPI/ATA 设备, 而没有分配到本 LSI 的 ATAPI 模块。)

地址	读寄存器	写寄存器	引脚地址 ($\overline{\text{IDECS}}[1:0]$ 、 $\text{IDEA}[2:0]$) H: HighLevel L: LowLevel@3.3V I/O	存取长度 *1 (可能位的长度)	寄存器位置
H'FFF0 0000	数据	数据	HL-LLL/HH-XXX (X: Don't care)	32(16)*2	驱动器
H'FFF0 0004	错误	功能	HL-LLH	32(8)*3	驱动器
H'FFF0 0008	扇区计数	扇区计数	HL-LHL	32(8)*3	驱动器
H'FFF0 000C	扇区号	扇区号	HL-LHH	32(8)*3	驱动器
H'FFF0 0010	柱面低位	柱面低位	HL-HLL	32(8)*3	驱动器
H'FFF0 0014	柱面高位	柱面高位	HL-HLH	32(8)*3	驱动器
H'FFF0 0018	设备 / 磁头	设备 / 磁头	HL-HHL	32(8)*3	驱动器
H'FFF0 001C	状态	命令	HL-HHH	32(8)*3	驱动器
H'FFF0 0038	代替状态	设备控制	LH-HHL	32(8)*3	驱动器

【注】 *1 CPU 以长字 (32 位) 为单位存取上述寄存器, 禁止字节存取和字存取。

*2 使用数据总线的 bit15 ~ 0。

*3 使用数据总线的 bit7 ~ 0。

表 17.3 ATAPI 包命令任务文件寄存器映像
(下列寄存器分配到 ATAPI/ATA 设备, 而没有分配到本 LSI 的 ATAPI 模块。)

地址	读寄存器	写寄存器	引脚地址 ($\overline{\text{IDECS}}[1:0]$ 、 $\text{IDEA}[2:0]$)	存取长度 *1 (可能位的长度)	寄存器位置
H'FFF0 0000	数据	数据	HL-LLL	32(16)*2	驱动器
H'FFF0 0004	错误	功能	HL-LLH	32(8)*3	驱动器
H'FFF0 0008	中断源	—	HL-LHL	32(8)*3	驱动器
H'FFF0 000C	—	—	HL-LHH	32(8)*3	驱动器
H'FFF0 0010	字节计数低位	字节计数低位	HL-HLL	32(8)*3	驱动器
H'FFF0 0014	字节计数高位	字节计数高位	HL-HLH	32(8)*3	驱动器
H'FFF0 0018	设备选择	设备选择	HL-HHL	32(8)*3	驱动器
H'FFF0 001C	状态	命令	HL-HHH	32(8)*3	驱动器
H'FFF0 0038	代替状态	设备控制	LH-HHL	32(8)*3	驱动器

【注】 *1 以长字 (32 位) 为单位存取上述寄存器, 禁止字节存取和字存取。

*2 使用数据总线的 bit15 ~ 0。

*3 使用数据总线的 bit7 ~ 0。

表 17.4 ATAPI 控制寄存器映像
(下列寄存器分配到本 LSI 的 ATAPI 模块。)

地址	寄存器名	略称	存取类型	寄存器的存取长度*
H'FFF0 0080	ATAPI 控制	ATAPI_CONTROL	R/W	32
H'FFF0 0084	ATAPI 状态	ATAPI_STATUS	R/W	32
H'FFF0 0088	中断允许	ATAPI_INT_ENABLE	R/W	32
H'FFF0 008C	PIO 时序	ATAPI_PIO_TIMING	R/W	32
H'FFF0 0090	多字 DMA 时序	ATAPI_MULTI_TIMING	R/W	32
H'FFF0 0094	超级 DMA 时序	ATAPI_ULTRA_TIMING	R/W	32
H'FFF0 0098	描述符表基址	ATAPI_DTB_ADR	R/W	32
H'FFF0 009C	DMA 起始地址	ATAPI_DMA_START_ADR	R/W	32
H'FFF0 00A0	DMA 传送计数	ATAPI_DMA_TRANS_CNT	R/W	32
H'FFF0 00A4	ATAPI 控制 2	ATAPI_CONTROL2	R/W	32
H'FFF0 00A8	保留		R	32
H'FFF0 00AC	保留		R	32
H'FFF0 00B0	ATAPI 信号状态	ATAPI_SIG_ST	R	32
H'FFF0 00BC	字节交换	ATAPI_BYTE_SWAP	R/W	32

【注】 * 以长字 (32 位) 为单位存取上述寄存器, 禁止字节存取和字存取。

【寄存器说明的符号说明】

初始值 : 复位后的寄存器值

— : 不定值

R/W : 可读写。能读写值。

R/WC0 : 可读写。写 0 时该位被初始化, 忽视写 1。

R : 可读。只能写 0。

—/W : 可写。读取值为不定值。

控制寄存器和状态寄存器都为高电平有效。

17.3.1 ATAPI 控制寄存器 (ATAPI_CONTROL)

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
	—	—	—	—	—	—	DTCD	—	RESET	M/S	BUSSEL	UDMAEN	DESE	R/W	STOP	START
初始值:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 10	—	全 0	R	保留位 读取值为 0, 只能写 0。

位	位名	初始值	R/W	说明
9	DTCD	0	R/W	在超级 DMA 运行时, 此位控制发生连续设备终止操作的运行模式。即使在接收到设备终止后还达不到传送个数时, 也不进行异常结束, 等待下一个 device 的 DMARQ, 然后重新开始传送。 由于现有的 ATA 设备中存在将设备终止作为暂停处理的设备, 所以即使在接收到设备终止后还达不到传送个数时, 不进行异常结束, 而必须等待下一个 device 的 DMARQ, 然后重新开始传送。该运行模式称为“设备终止连续模式”。 1: 抑制设备终止连续模式 0: 设备终止连续模式
8	—	0	R	保留位 读取值为 0, 只能写 0。
7	RESET	0	R/W	控制 ATAPI 设备的复位。如果将此位置 1, ATAPI 复位信号就有效。IDERST 信号为低电平有效。 当将此位置 1 时, IDERST 信号就变为低电平; 当将此位清 0 时, IDERST 信号就变为高电平。
6	M/S	0	R/W	选择 ATAPI 设备是为主控还是为从属设备。 1: ATAPI 设备为主控。 0: ATAPI 设备为从属设备。
5	BUSSEL	1	R/W	选择 Pixel 总线或者 I/O 总线。 1: 选择 Pixel 总线。 0: 选择 I/O 总线。 【注】 本 LSI 不能将此位置 0, 必须在设定为 1 后才能使用。
4	UDMAEN	0	R/W	超级 DMA 允许位 在使用超级 DMA 时, 必须将此位置 1。在使用多字 DMA 或者 PIOFIFO 模式时, 必须将此位清 0。
3	DESE	0	R/W	控制 descriptor table 的运行模式。 1: Descriptor 功能有效。 0: Descriptor 功能无效。
2	R/W	0	R/W	控制 FIFO 的读写。 1: 读 FIFO (DMA 传送时的 data-in 运行)。 0: 写 FIFO (DMA 传送时的 data-out 运行)。 在从 ATAPI 设备读数据时, 必须将此位置 1。 在将数据写到 ATAPI 设备时, 必须将此位清 0。
1	STOP	0	R/W	控制 DMA 传送的停止。 写时 0: 被忽视。 1: 停止数据传送。 读时 0: 不发行停止命令。 1: 发行数据传送的停止命令。 在开始下一个 DMA 传送时, 必须将此位清 0。 【注】 不能从停止 DMA 传送的地址重新开始传送。
0	START	0	R/W	控制 DMA 传送的开始。 当将此位置 1 时, 就开始 DMA 传送; 当将此位清 0 时, 就被忽视。 写时 0: 被忽视。 1: 开始 DMA 传送。 读时 0: DMA 传送无效。 1: 在 DMA 传送中, 为忙状态。 【注】 在 DMA 激活时, 禁止存取任务文件寄存器。

17.3.2 ATAPI 状态寄存器 (ATAPI_STATUS)

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
	—	—	—	—	—	—	—	SWERR	IFERR	DNEND	DEVTRM	DEVINT	TOUT	ERR	NEND	ACT
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/WC0	R/WC0	R/WC0	R/WC0	R	R/WC0	R/WC0	R/WC0	R

位	位名	初始值	R/W	说明
31 ~ 9	—	全 0	R	保留位 读取值为 0，只能写 0。
8	SWERR	0	R/WC0	软件错误标志位 当此位被置 1 时，表示在 DMA 激活时存取了任务文件寄存器。在 DMA 激活时禁止存取任务寄存器，例如，在超级 DMA 或者多字 DMA 的传送过程中，如果进行 PIO 传送，此位就被置 1。此时，不输出到 LSI 外部，并且忽视存取操作。 通过写 0，此位被复位。
7	IFERR	0	R/WC0	表示检测到 ATAPI 接口协议错误。 <ul style="list-style-type: none"> 超级 DMAdata-in 突发处于主机终止状态，(IDEDREQ=1) 或者 (IDEIORDY=0) 时。 超级 DMAdata-out 突发处于设备终止状态，IDEIORDY=0 时。 开始超级 DMAdata-out 突发时，IDEIORDY=0。 超级 DMAdata-out 突发处于主机终止状态，(IDEDREQ=1) 或者 (IDEIORDY=0) 时。 通过写 0，此位被复位。
6	DNEND	0	R/WC0	表示在描述符模式中全部 DMA 都正常结束。通过写 0，此位被复位。
5	DEVTRM	0	R/WC0	在到达本 ATAPI 模块设定的 DMA 传送字节数之前，如果 ATAPI 设备的超级 DMA 模式结束，此位就被置 1。通过写 0 时，此位被复位。
4	DEVINT	0	R	表示 ATAPI 设备中断 IDEINT 的状态。此位是只读位，在本 LSI 芯片内不保持状态，因此当 IDEINT 为 0 时，此位也被清 0。ATAPI 接口将 ATAPI 设备的中断信号视为电平触发输入。按照 ATAPI 规格，为了清除中断未定状态，在将用于读状态寄存器的 IDEIORD 变为无效电平后，必须在 400ns 以内通过 ATAPI 设备将 IDEINT 变为无效电平。 写无效。
3	TOUT	0	R/WC0	表示检测到 IORDY 超时。如果在 150 个 Pixel 总线时钟周期以上的期间无应答 (IDEIORDY 引脚为低电平)，就检测为超时。 通过写 0，此位被复位。
2	ERR	0	R/WC0	当检测到 DMA 中断时，此位被置 1。 <ul style="list-style-type: none"> 在主机强制结束 DMA 传送时 在 DTCD=1、发生设备终止并且 ACT=0 时，ERR=1。 通过入 0，此位被复位。
1	NEND	0	R/WC0	表示 DMA 正常结束。通过写 0，此位被复位。
0	ACT	0	R	表示 DMA 激活。此位是只读位，在 DMA 传送结束时被清 0。不建议将此位用作中断源。写无效。

17.3.3 中断允许 (ATAPI_INT_ENABLE)

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
	—	—	—	—	—	—	—	iSWERR	iIFERR	iDNEND	iDEVTRM	iDEVINT	iTOUT	iERR	iNEND	iACT
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 9	—	全 0	R	保留位 读取值为 0，只能写 0。
8	iSWERR	0	R/W	SWERR 中断允许位
7	iIFERR	0	R/W	IFERR 中断允许位
6	iDNEND	0	R/W	DNEND 中断允许位
5	iDEVTRM	0	R/W	DEVTRM 中断允许位
4	iDEVINT	0	R/W	DEVINT 中断允许位
3	iTOUT	0	R/W	TOUT 中断允许位
2	iERR	0	R/W	ERR 中断允许位
1	iNEND	0	R/W	NEND 中断允许位
0	iACT	0	R/W	ACT 中断允许位 当 DMA 传送结束时，就自动清除 ACT，因此不建议将此位置 1。

【注】 如果将各位置 1，就允许 ATAPI 状态寄存器各位对应的中断信号。

17.3.4 PIO 时序寄存器 (ATAPI_PIO_TIMING)

存取 ATAPI 设备前，请在此寄存器的以下位设定机器周期数。
机器周期数为 Pixel 总线时钟。

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

位	位名	初始值	R/W	说明
31、30	—	全 0	R	保留位 读取值为 0，只能写 0。
29 ~ 24	pSDCT	0	R/W	设定从属 ATAPI 设备的周期时间。
23 ~ 19	pSDPW	0	R/W	设定从属 ATAPI 设备的 $\overline{\text{IDEIORD}}/\overline{\text{IDEIOWR}}$ 脉冲宽度。
18 ~ 16	pSDST	0	R/W	设定在 PIO 模式中对于从属 ATAPI 设备 $\overline{\text{IDEIORD}}/\overline{\text{IDEIOWR}}$ 的地址准备时间。
15、14	—	全 0	R	保留位 读取值为 0，只能写 0。
13 ~ 8	pMDCT	0	R/W	设定主控 ATAPI 设备的周期时间。
7 ~ 3	pMDPW	0	R/W	设定主控 ATAPI 设备的 $\overline{\text{IDEIORD}}/\overline{\text{IDEIOWR}}$ 脉冲宽度。
2 ~ 0	pMDST	0	R/W	设定在 PIO 模式中对于主控 ATAPI 设备 $\overline{\text{IDEIORD}}/\overline{\text{IDEIOWR}}$ 的地址准备时间。

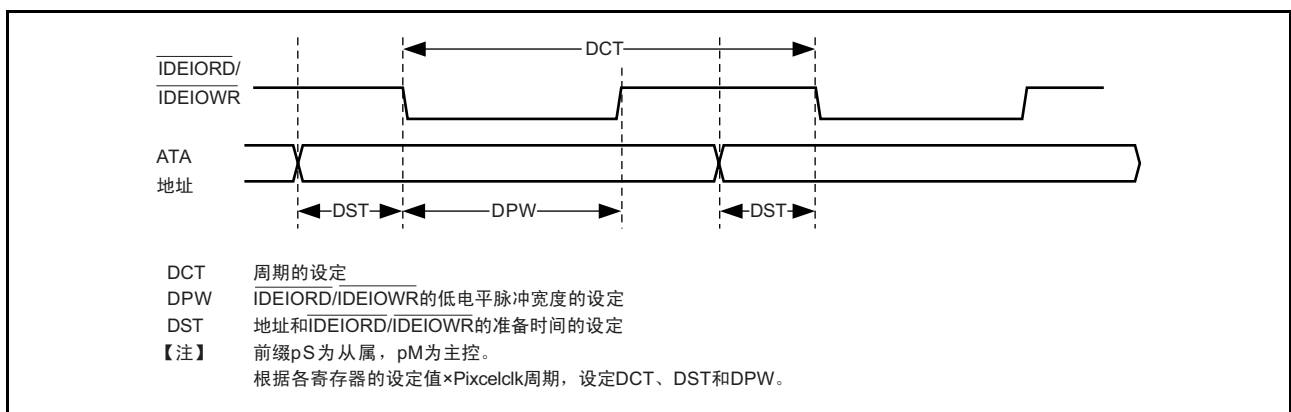


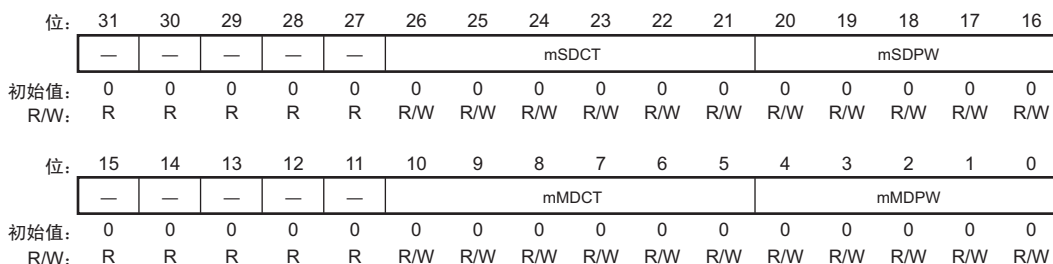
图 17.2 PIO 时序寄存器

• PIO 时序寄存器的设定值表 (主控/从属)

Pixel 总线时钟	模式 0	模式 1	模式 2	模式 3	模式 4
100MHz	H'3DF0	H'28F6	H'22F4	H'134C	H'0D44

17.3.5 多字 DMA 时序寄存器 (ATAPI_MULTI_TIMING)

在存取 ATAPI 设备前，必须给此寄存器的以下位设定机器周期数。



位	位名	初始值	R/W	说明
31 ~ 27	—	全 0	R	保留位 读取值为 0，只能写 0。
26 ~ 21	mSDCT	0	R/W	设定从属 ATAPI 设备的周期时间。
20 ~ 16	mSDPW	0	R/W	设定从属 ATAPI 设备的 $\overline{\text{IDEIORD}}/\overline{\text{IDEIOWR}}$ 脉冲宽度。
15 ~ 11	—	全 0	R	保留位 读取值为 0，只能写 0。
10 ~ 5	mMDCT	0	R/W	设定主控 ATAPI 设备的周期时间。
4 ~ 0	mMDPW	0	R/W	设定主控 ATAPI 设备的 $\overline{\text{IDEIORD}}/\overline{\text{IDEIOWR}}$ 脉冲宽度。

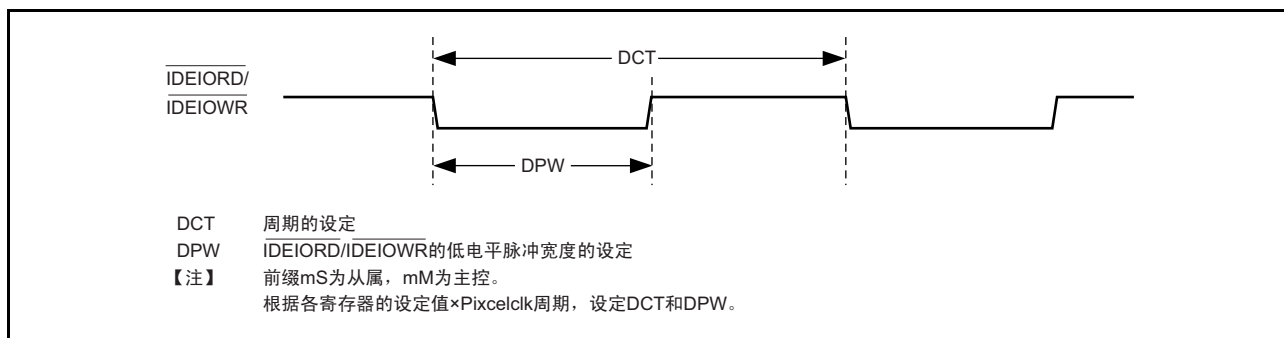


图 17.3 多字 DMA 时序寄存器

• 多字DMA时序寄存器的设定值表

Pixel 总线时钟	模式 0	模式 1	模式 2
100MHz	H'0637	H'0209	H'01A8

17.3.6 超级 DMA 时序寄存器 (ATAPI_ULTRA_TIMING)

在存取 ATAPI 设备前，必须给此寄存器的以下位设定机器周期数。

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

位	位名	初始值	R/W	说明
31 ~ 25	—	全 0	R	保留位 读取值为 0，只能写 0。
24 ~ 21	uSDCT	0	R/W	设定从属 ATAPI 设备的周期时间。
20 ~ 16	uSDRP	0	R/W	设定从 DMARDY (不是 IDEIORDY) 无效到从属 ATAPI 设备产生中断的时间。
15 ~ 9	—	全 0	R	保留位 读取值为 0，只能写 0。
8 ~ 5	uMDCT	0	R/W	设定主控 ATAPI 设备的周期时间。
4 ~ 0	uMDRP	0	R/W	设定从 DMARDY (不是 IDEIORDY) 无效到从属 ATAPI 设备产生中断的时间。

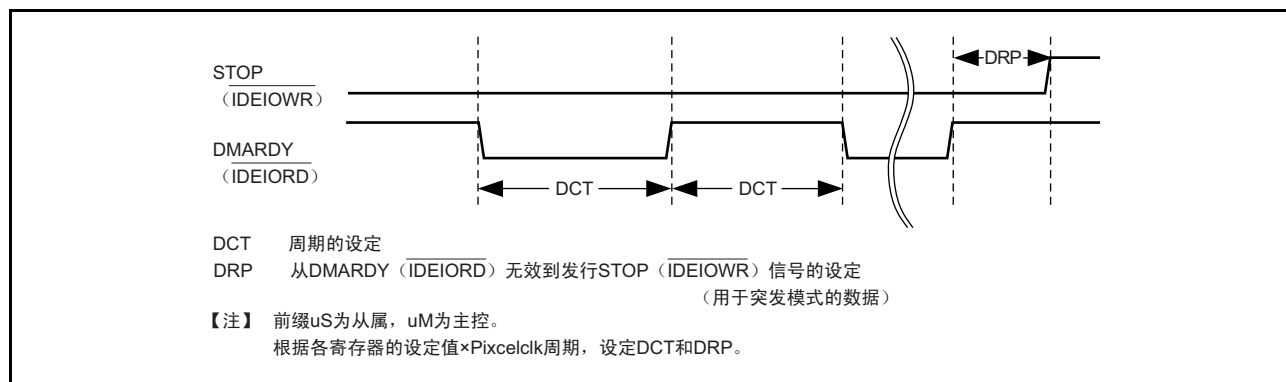


图 17.4 超级 DMA 时序寄存器

• 超级 DMA 时序的设定值表

Pixel 总线时钟	模式 0	模式 1	模式 2	模式 3	模式 4
100MHz	H'0191	H'010E	H'00CB	H'00AB	H'006B

17.3.7 描述符表基址寄存器 (ATAPI_DTB_ADR)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DTBAA[2:0]			DTBA[25:16]									
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/W	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
	DTBA[15:2]														—	—
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	R

位	位名	初始值	R/W	说明
31 ~ 29	—	全 0	R	保留位 读取值为 0，只能写 0。
28 ~ 26	DTBAA[2:0]	0	R/W	设定描述符表基址的 SDRAM 区。设定值如下： 001：设定 SDRAM 区 1。 010：设定 SDRAM 区 2。 上述以外：禁止设定
25 ~ 2	DTBA[25:2]	0	R/W	表示描述符表基址。bit25 ~ 0 用于以字节为单位设定描述符基址。 必须给描述符表确保 64 位地址边界，所以必须将 bit3 和 bit2 置 0。 忽视 bit1 和 bit0。
1、0	—	全 0	R	保留位 读取值为 0，只能写 0。

- 【注】
1. 此寄存器只在将 ATAPI 控制寄存器的 bit5 (BUSSEL) 置 1 时有效。
 2. 本地址在 DMA 激活后不变而保持设定值。
 3. 在 32 位地址模式中，将指定的 32 位地址的低 29 位地址设定到此寄存器的 bit28 ~ bit0。

17.3.8 描述符表

描述符表是以结束标志、描述符 DMA 起始区 / 地址和描述符 DMA 传送计数为 1 组的表格。

存储器内的描述符表映像

地址	数据内容
DTBA	第 1 个结束标志 (bit31=0) 和 DDSTAA/DDSTA 存储区 / 地址
DTBA+4	第 1 个 DDTRC 存储地址
DTBA+8	第 2 个结束标志 (bit31=0) 和 DDSTAA/DDSTA 存储区 / 地址
DTBA+12	第 2 个 DDTRC 存储地址
...	...
DTBA+8×(n-1)	第 n 个结束标志 (bit31=1) 和 DDSTAA/DDSTA 存储区 / 地址
DTBA+8×(n-1)+4	第 n 个 DDTRC 存储地址

17.3.9 结束标志和描述符 DMA 起始地址

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

有效标志和描述符 DMA 起始地址被设定为存储器内的描述符表基址 +m (m: 2 的倍数, 0, 2, 4...)。

bit	位名	初始值	R/W	说明
31	DTEND	—	R/W	控制描述符 DMA 操作的结束。 1: 描述符 DMA 操作结束 当此位为 1 时, 就认为是最后的描述符表。 0: 描述符表有效 当此位为 0 时, 读 DMA 传送计数, 传送 DMA, 并且读下一个描述符表。
30 ~ 29	—	全 0	R	保留位 读取值为 0, 只能写 0。
28 ~ 26	DDSTAA[2:0]	—	R/W	设定描述符操作时 DMA 起始 SDRAM 区。设定值如下: 001: 设定 SDRAM 区 1。 010: 设定 SDRAM 区 2。 上述以外: 禁止设定
25 ~ 2	DDSTA[25:2]	—	R/W	表示描述符表基址。bit25 ~ 0 用于以字节为单位设定描述符基址。 必须给描述符表确保 256 位地址边界, 所以必须将 bit4 ~ 2 置 0。 忽视 bit1 和 bit0。
1、0	—	全 0	R	保留位 读取值为 0, 只能写 0。

17.3.10 描述符 DMA 传送计数

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

描述符 DMA 传送计数被设定为存储器内的描述符表基址 +m (m: 2 的倍数 +1, 1, 3, 5...)。

位	位名	初始值	R/W	说明
31 ~ 29	—	全 0	R	保留位 读取值为 0，只能写 0。
28 ~ 1	DDTRC[28:1]	—	R/W	设定描述符操作时的 DMA 传送计数。 bit28 ~ 0 用于以字节为单位设定 DMA 传送计数。 ATAPI 的数据总线宽度为 16 位（字单位），因此忽视 bit0。
0	—	0	R	保留位 读取值为 0，只能写 0。

17.3.11 DMA 起始地址寄存器 (ATAPI_DMA_START_ADR)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DSTAA[2:0]			DSTA[25:16]									
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/W	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
	DSTA[15:2]														—	—
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	R

位	位名	初始值	R/W	说明
31 ~ 29	—	全 0	R	保留位 读取值为 0，只能写 0。
28 ~ 26	DSTAA[2:0]	0	R/W	设定描述符操作时的 DMA 起始 SDRAM 区。设定值如下： 001：设定 SDRAM 区 1。 010：设定 SDRAM 区 2。 上述以外：禁止设定
25 ~ 2	DSTA[25:2]	0	R/W	设定 DMA 起始地址（表示存储器内的数据传送起始地址）。 bit25 ~ 0 用于以字节为单位设定 DMA 起始地址。必须给 DMA 起始地址确保 256 位地址边界，所以必须将 bit4 ~ 2 置 0。 忽视 bit1 和 bit0。
1、0	—	全 0	R	保留位 读取值为 0，只能写 0。

- 【注】
1. 此寄存器只在将 ATAPI 控制寄存器的 bit5 (BUSSEL) 置 1 时有效。
 2. 本地地址在 DMA 激活后不变而保持设定值。
 3. 在 32 位地址模式中，将指定的 32 位地址和 SDRAM 区设定到此寄存器的 bit28 ~ bit0。

17.3.12 DMA 传送计数寄存器 (ATAPI_DMA_TRANS_CNT)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DTRC[28:16]												
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/W	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
	DTRC[15:1]															—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

位	位名	初始值	R/W	说明
31 ~ 29	—	全 0	R	保留位 读取值为 0，只能写 0。
28 ~ 1	DTRC[28:1]	0	R/W	设定 DMA 传送计数。 bit28 ~ 0 用于以字节为单位设定 DMA 传送计数。 ATAPI 的数据总线宽度为 16 位 (字单位)，因此忽视 bit0。
0	—	0	R	保留位 读取值为 0，只能写 0。

【注】 本计数值在 DMA 激活后不变而保持设定值。

17.3.13 ATAPI 控制 2 寄存器 (ATAPI_CONTROL2)

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WORD SWAP	IFEN
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
31 ~ 2	—	全 0	R	保留位 读取值为 0，只能写 0。
1	WORDSWAP	0	R/W	在允许 Pixel 总线的 32 位总线时，此位控制是否将高 16 位数据和低 16 位数据进行交换。 0: 不进行字交换。以大端法表示 Pixel 总线上的 32 位数据。 1: 在 ATAPI 接口和寄存器 /Pixel 总线之间进行字交换。以小端法表示 Pixel 总线上的 32 位数据。 数据传送时的字交换只在将 ATAPI 控制寄存器的 bit0 置 1 并且 DMA 模式开始时才有效。除了 DMA 以外，全部寄存器的存取都为长字存取。
0	IFEN	0	R/W	控制是否允许 ATAPI 控制器。 0: ATAPI 控制器无效 1: ATAPI 控制器有效 【注】 为 0 时，ATAPI 控制器的 I/O 引脚用作输入引脚，输出引脚为高阻抗状态。

17.3.14 ATAPI 信号状态寄存器 (ATAPI_SIG_ST)

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DDMARDY	DMARQ
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
31 ~ 2	—	全 0	R	保留位 读取值为 0，只能写 0。
1	DDMARDY	—	R	表示 ATAPIDDMARDY (IDEIORDY 反相) 信号状态。
0	DMARQ	—	R	表示 ATAPIDMARQ (IDEDREQ) 信号状态。

17.3.15 字节交换寄存器 (ATAPI_BYTE_SWAP)

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BYTE SWAP
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
31 ~ 1	—	全 0	R	保留位 读取值为 0，只能写 0。
0	BYTESWAP	0	R/W	是控制 ATAPI 接口的高 8 位和低 8 位进行交换的位。 1: 在 APAPI 接口和 Pixel 总线之间进行字节交换。 字节交换只在将 ATAPI 控制寄存器的 bit0 置 1 并且 DMA 模式开始时才有效。

17.3.16 ATAPI 的数据总线调整

IO-BUS侧的数据总线的调整
不会因大端法/小端法而变。
物理总线宽为3:

总线		32位总线				16位总线				8位总线			
存取		31	16	8	0	31	16	8	0	31	16	8	0
大小	地址												
字节	4n 4n+1 4n+2 4n+3	不规定				不规定				不规定			
字	4n 4n+2	不规定				不规定				不规定			
长字	4n	B3	B2	B1	B0	不规定				不规定			

B3: 31~24、B2: 23~16、B1: 15~8、B0: 第7~0位

PIXEL-BUS侧的数据总线的调整
总线宽固定为32位，存取大小为长字。

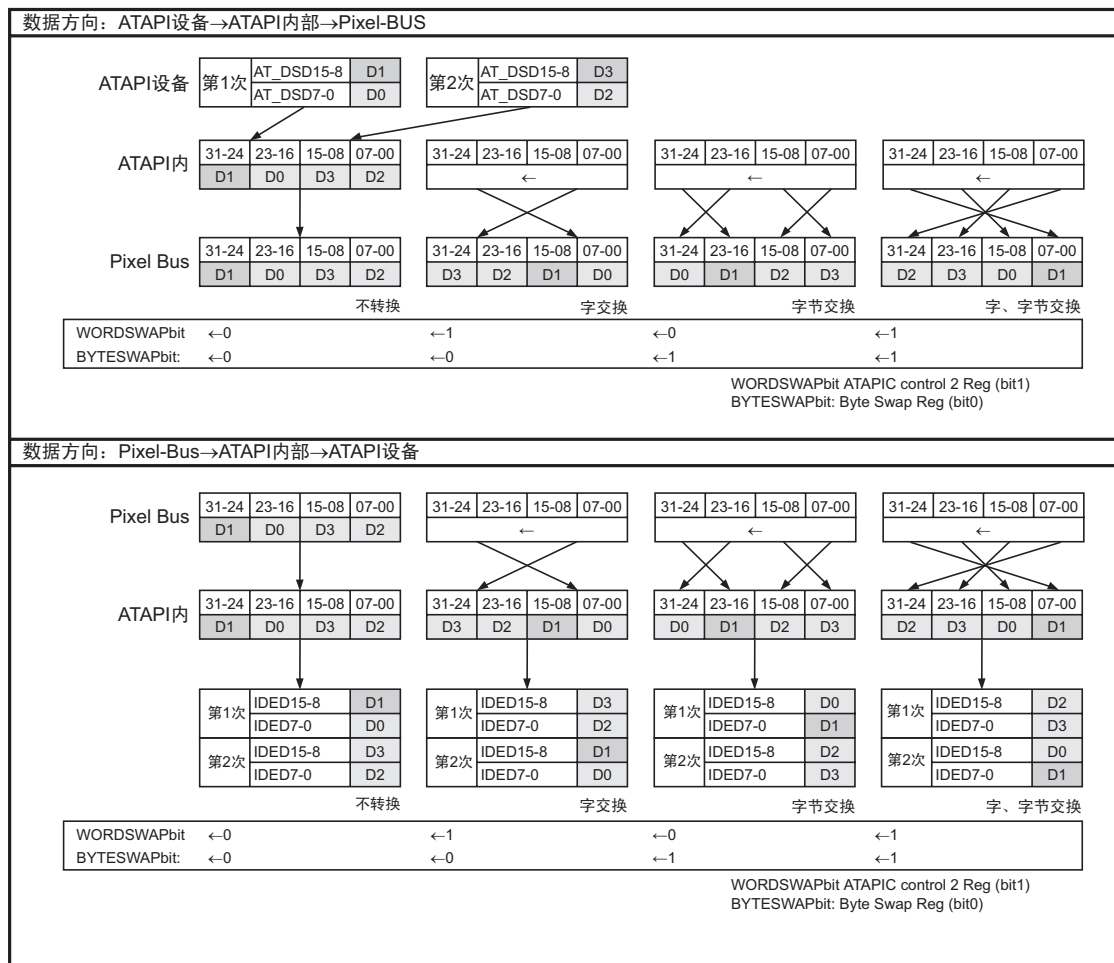


图 17.5 ATAPI 的数据总线调整

17.4 功能说明

ATAPI 控制器支持作为主机的主通道，并且还支持 ATAPI 接口规格书上定义的主控 / 从属结构。ATAPI 接口的读 / 写 FIFO 缓冲器是为了对 PIO 模式和多字 DMA 模式实现最大 16M 字节 / 秒的数据传送而设计的。ATAPI 控制器支持 3.3V I/O 接口。

ATA 任务文件寄存器和 ATAPI 包命令任务文件寄存器分配到 SH 寄存器映像空间。因此，在通过 SH 存取此寄存器时，能通过 DCS[1:0] 引脚和 DSA[2:0] 引脚的寻址，存取 DVD ROM 驱动器等设备内的寄存器。

17.4.1 数据传送模式

ATAPI 接口控制寄存器支持 PIO 传送模式、多字 DMA 传送模式和超级 DMA 传送模式。它启动传送模式以及设定各传送模式的 ATAPI 接口时序。

支持 PIO 模式 0 ~ 4、多字 DMA 模式 0 ~ 2 和超级 DMA 模式 0 ~ 4 (最大 66MB/s)。

Pixel 总线能用于多字 DMA 数据传送和超级 DMA 数据传送，但是 PIO 传送只能使用 I/O 总线。

表 17.5 数据传送模式

数据传送模式 内部操作和内部寄存器	PIO 数据传送	ATA 设备 -Pixel 总线之间的 DMA 数据传送	
		多字 DMA	超级 DMA
FIFO 运行	旁路 *	使用	使用
控制寄存器的 BUSSEL 位	Don't Care	1	1
控制寄存器的 UDMAEN 位	Don't Care	0	1
控制寄存器的 START/STOP 位	Not Used	Used	Used

【注】 * CPU 对 ATA 设备进行 PIO 存取。

本表的 DMA 传送是在 ATAPI 设备和存储器之间进行的传送数据。

17.4.2 描述符功能

ATAPI 控制器能指定在 DMA 数据传送的存储空间中不重复的多个连续存储空间。在描述符表中设定各个 DMA 起始地址和 DMA 传送计数。

17.5 运行步骤

17.5.1 初始化

(1) 接口允许位的设定

必须将 ATAPI 控制 2 寄存器的 IFEN 位置 1。

(2) 时序寄存器的设定

必须给以下寄存器写适当的值。有关这些值，请参照各寄存器的说明。

- PIO 时序寄存器
- 多字 DMA 时序寄存器
- 超级 DMA 时序寄存器

17.5.2 PIO 传送模式的步骤

这是不使用 FIFO 的情况。

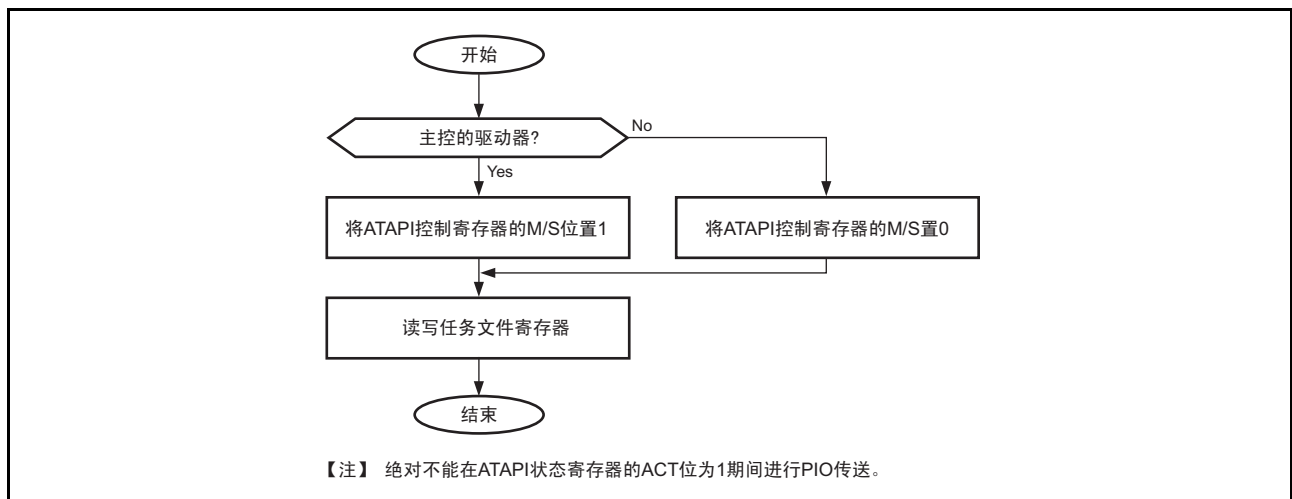


图 17.6 PIO 传送模式的步骤（不使用 FIFO 时）

17.5.3 多字 DMA 传送模式的步骤

这是通过查询并且经 Pixel 总线的存储器传送。

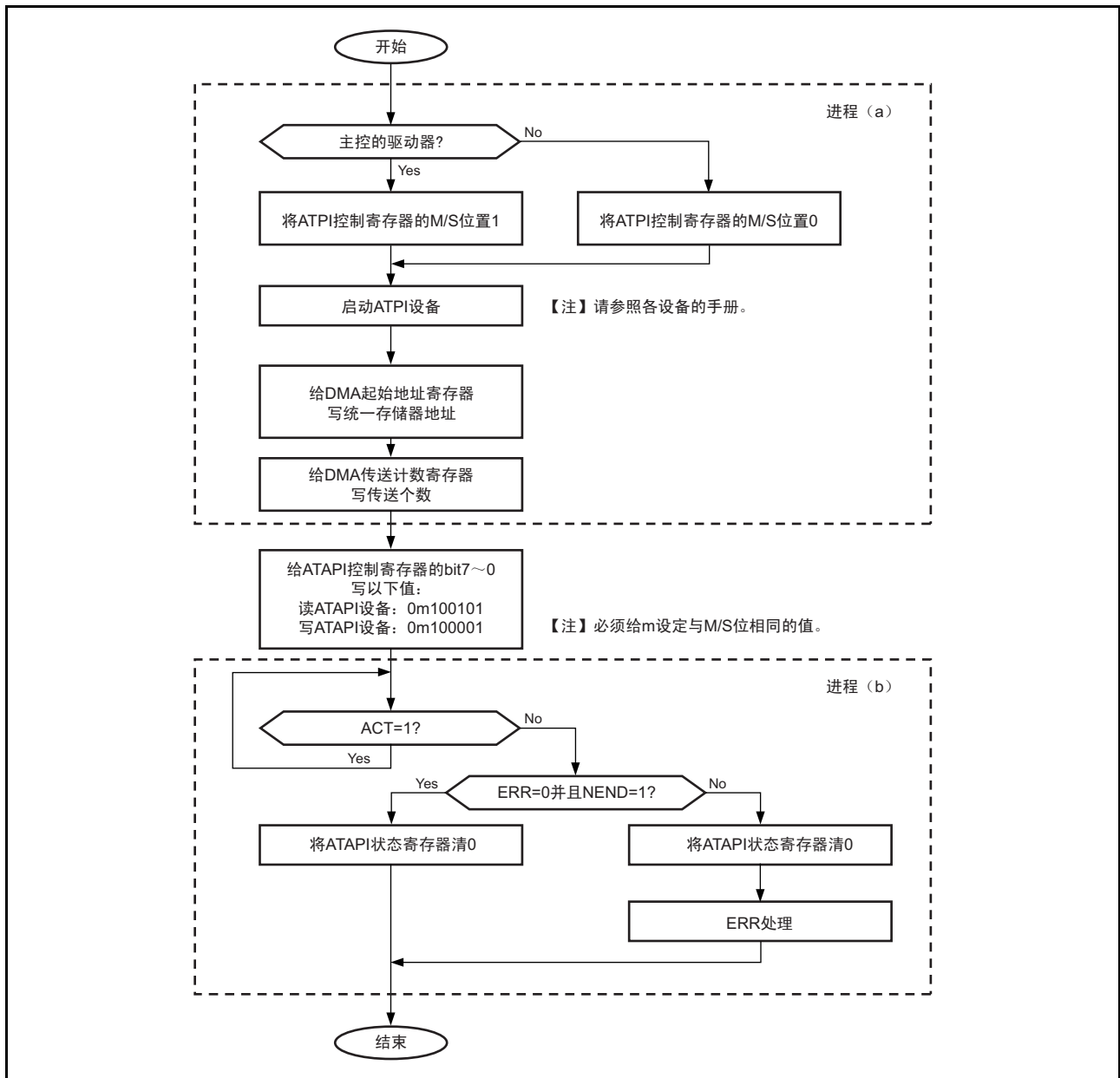


图 17.7 通过查询并且经 Pixel 总线的存储器传送

这是通过中断并且经 Pixel 总线的存储器传送。

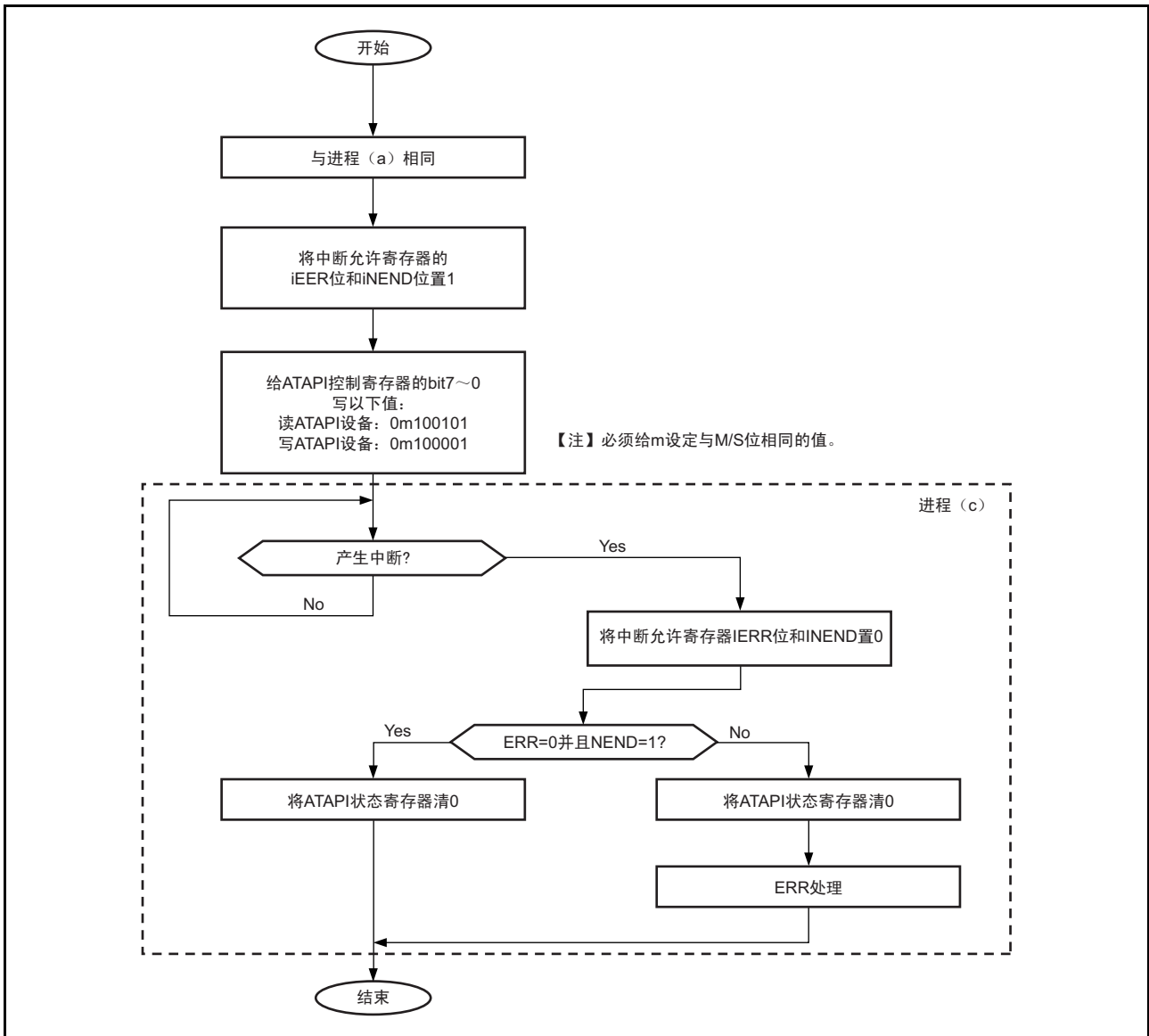


图 17.8 通过中断并且经 Pixel 总线的存储器传送

17.5.4 超级 DMA 传送模式的步骤

这是通过查询并且经 Pixel 总线的存储器传送。

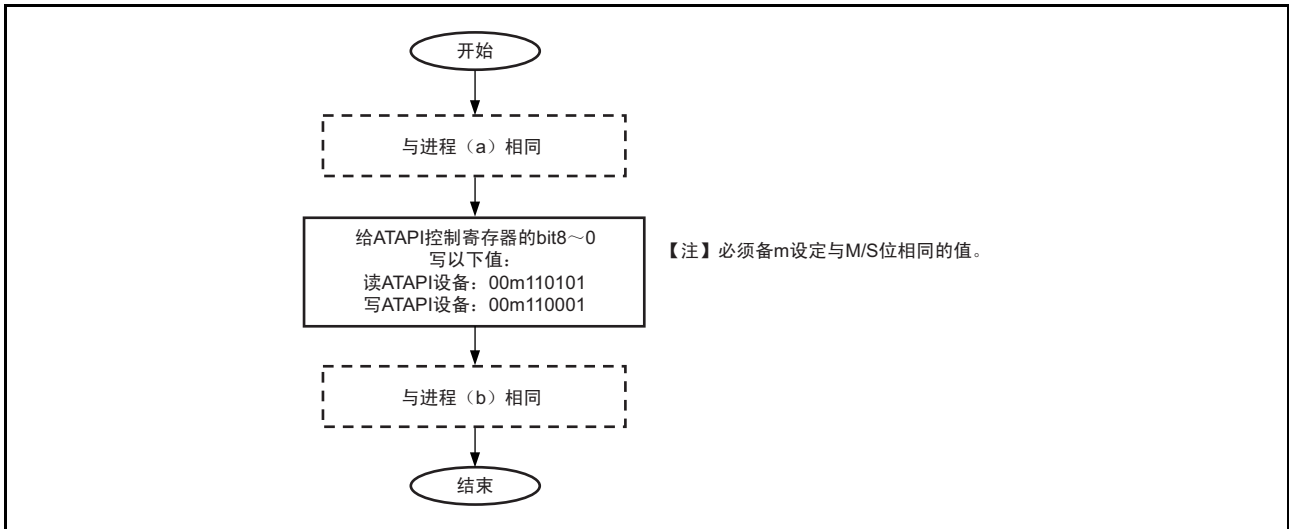


图 17.9 通过查询并且经 Pixel 总线的存储器传送

这是通过中断并且经 Pixel 总线的存储器传送。

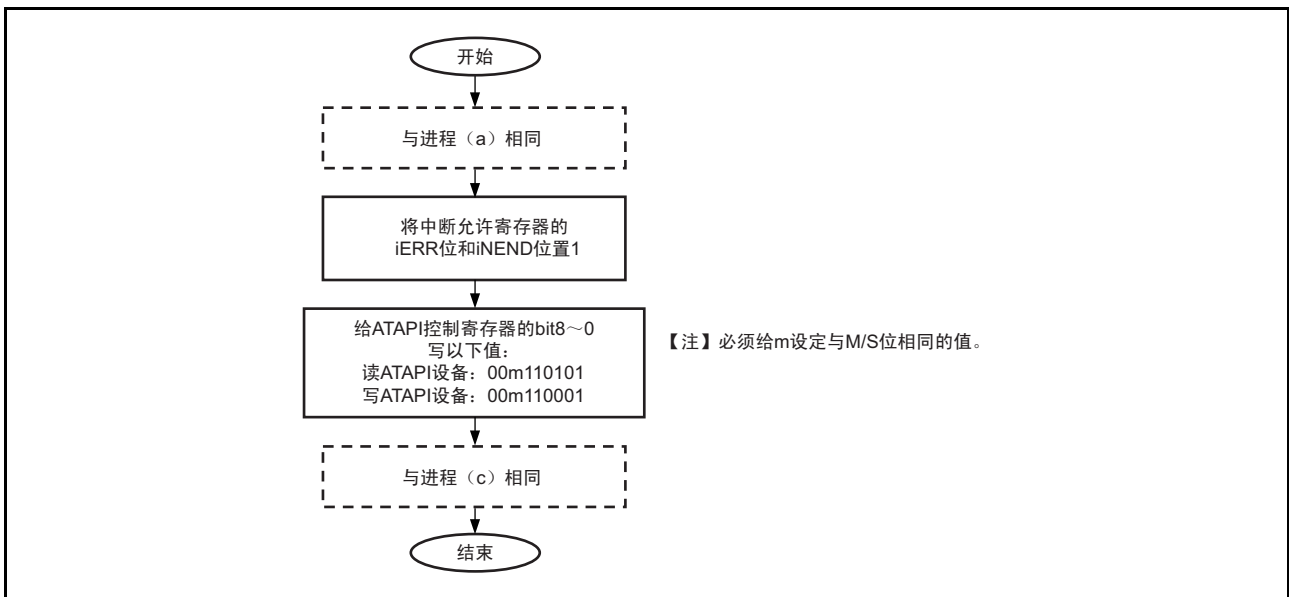


图 17.10 通过中断并且经 Pixel 总线的存储器传送

17.5.5 ATAPI 设备的硬件复位步骤

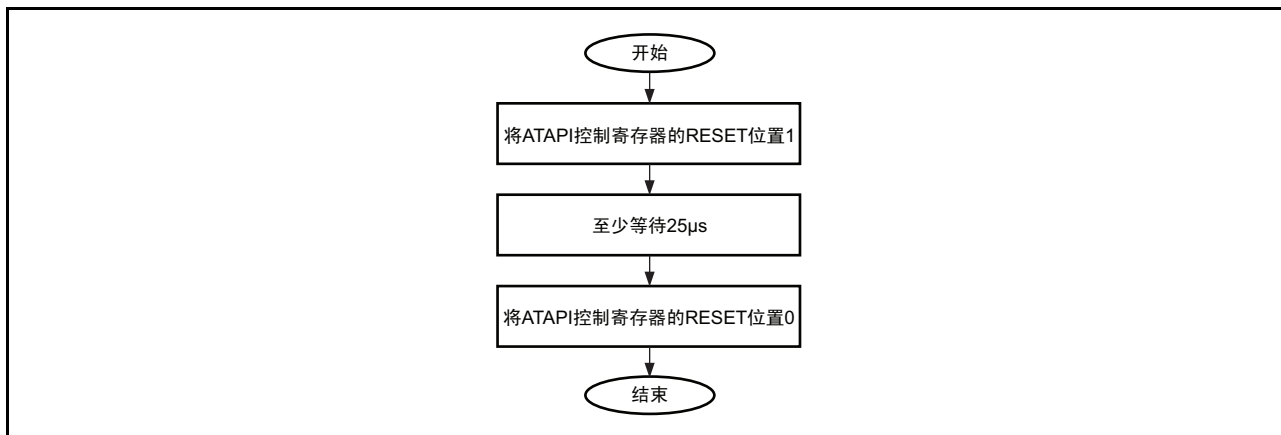


图 17.11 ATAPI 设备的硬件复位步骤

第 18 章 串行音频接口 (SSI)

串行音频接口 (SSI) 是和 Philips 方式等具有兼容性的各种设备进行音频数据发送和接收的模块。不仅对应其他一般格式，还对应多通道模式。

18.1 特点

(1) SSI 模块结构

SSI 模块由以下块构成：

- SSI_DMACH0/1
- SSI_CH0/1/2/3/4/5
- SSI_CLKSEL

(2) SSI 的特点

SSI 模块有以下特点：

- 通道数：6 个通道
- 运行模式：非压缩模式
非压缩模式支持被分割到各通道的串行音频流。
- SSI 模块也能用作发送器或者接收器。
能使用串行总线格式。
- 数据缓冲器和移位寄存器之间是异步传送。
- 能选择用于串行总线接口的时钟分频比。
- 能通过 SSI_DMACH0/1 或者中断控制数据的发送和接收。
- 能选择任意通道的音频时钟。
- 能选择任意通道的串行位时钟/串行字选择信号。

SSI 的框图如图 18.1 所示。

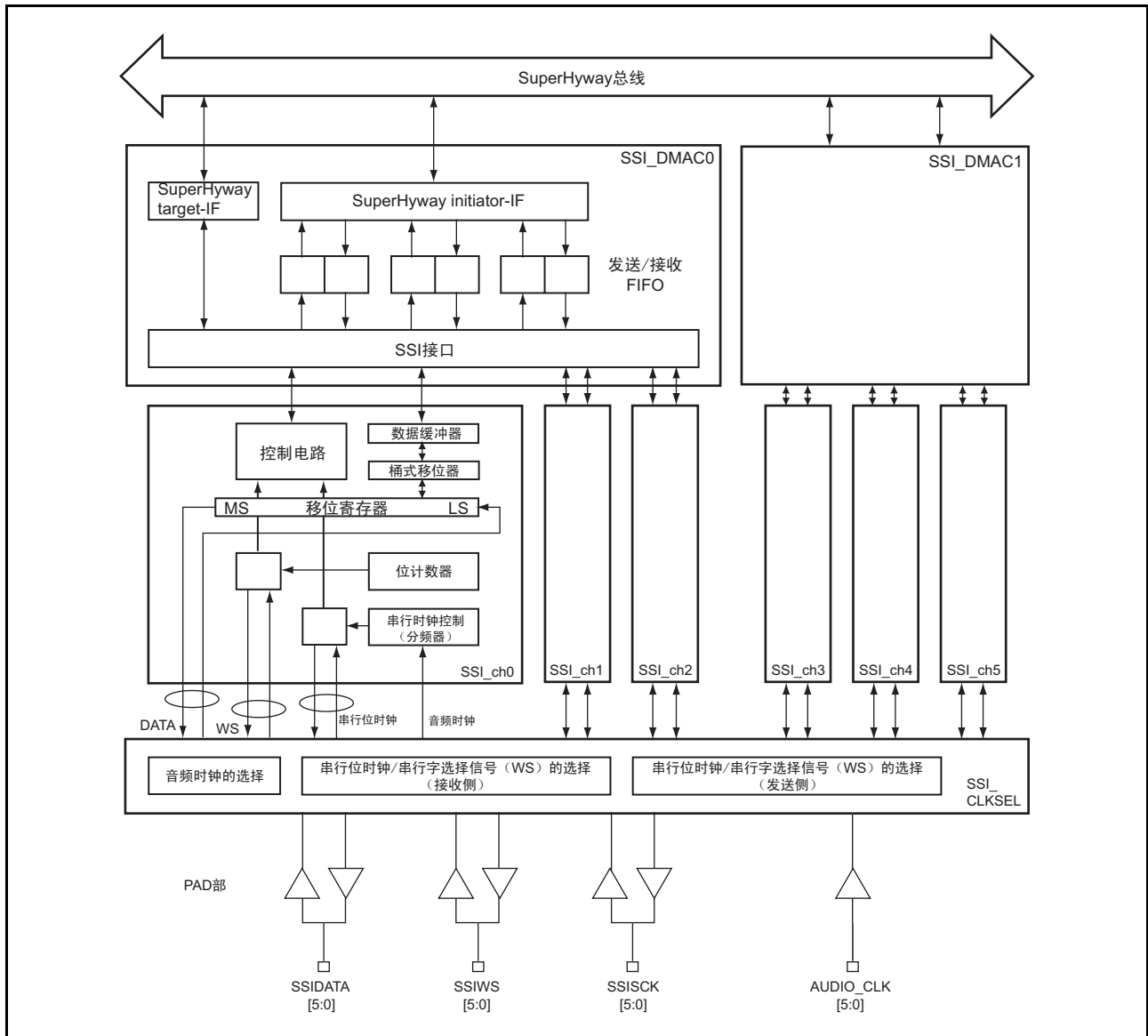


图 18.1 SSI 框图

18.2 输入 / 输出引脚

SSI 的引脚结构如表 18.1 所示。

表 18.1 SSI 的引脚结构

引脚名	功能	输入 / 输出	说明
AUDIO_CLK[5:0]	音频时钟	输入	SSI_CH0 ~ 5 的分频器输入时钟 (过采样时钟)
SSISCK[5:0]	串行位时钟	输入 / 输出	SSI_CH0 ~ 5 的串行位时钟
SSIWS[5:0]	字选择	输入 / 输出	SSI_CH0 ~ 5 的串行字选择信号
SSIDATA[5:0]	串行数据	输入 / 输出	SSI_CH0 ~ 5 的串行数据

18.3 寄存器说明

SSI_DMACH0 的寄存器结构如表 18.2 所示，各处理模式的寄存器状态如表 18.3 所示。

表 18.2 SSI_DMACH0 的寄存器结构

通道	名称	略称	R/W	P4 区地址	区域 7 地址	长度
0	DMA 模式寄存器 0	SSIDMMR0	R/W	H'FF40 1000	H'1F40 1000	32
	RDMA 传送源地址寄存器 0	SSIRDMAADR0	R/W	H'FF40 1008	H'1F40 1008	32
	RDMA 传送字节数寄存器 0	SSIRDMCNTR0	R/W	H'FF40 1010	H'1F40 1010	32
	WDMA 传送目标地址寄存器 0	SSIWDMADR0	R/W	H'FF40 1018	H'1F40 1018	32
	WDMA 传送字节数寄存器 0	SSIWDMCNTR0	R/W	H'FF40 1020	H'1F40 1020	32
	DMA 控制寄存器 0	SSIDMCOR0	R/W	H'FF40 1028	H'1F40 1028	32
	发送暂停块计数器 0	SSISTPBLCNT0	R/W	H'FF40 1030	H'1F40 1030	32
	发送暂停中传送数据寄存器 0	SSISTPDR0	R/W	H'FF40 1038	H'1F40 1038	32
	块计数源寄存器 0	SSIBLCNTR0	R/W	H'FF40 1040	H'1F40 1040	32
	块计数器 0	SSIBLCNT0	R	H'FF40 1048	H'1F40 1048	32
	块 xn 次中断计数源寄存器 0	SSIBLNCNTR0	R/W	H'FF40 1050	H'1F40 1050	32
	块 xn 次计数器 0	SSIBLNCNT0	R	H'FF40 1058	H'1F40 1058	32
1	DMA 模式寄存器 1	SSIDMMR1	R/W	H'FF40 1060	H'1F40 1060	32
	RDMA 传送源地址寄存器 1	SSIRDMAADR1	R/W	H'FF40 1068	H'1F40 1068	32
	RDMA 传送字节数寄存器 1	SSIRDMCNTR1	R/W	H'FF40 1070	H'1F40 1070	32
	WDMA 传送目标地址寄存器 1	SSIWDMADR1	R/W	H'FF40 1078	H'1F40 1078	32
	WDMA 传送字节数寄存器 1	SSIWDMCNTR1	R/W	H'FF40 1080	H'1F40 1080	32
	DMA 控制寄存器 1	SSIDMCOR1	R/W	H'FF40 1088	H'1F40 1088	32
	发送暂停块计数器 1	SSISTPBLCNT1	R/W	H'FF40 1090	H'1F40 1090	32
	发送暂停中传送数据寄存器 1	SSISTPDR1	R/W	H'FF40 1098	H'1F40 1098	32
	块计数源寄存器 1	SSIBLCNTR1	R/W	H'FF40 10A0	H'1F40 10A0	32
	块计数器 1	SSIBLCNT1	R	H'FF40 10A8	H'1F40 10A8	32
	块 xn 次中断计数源寄存器 1	SSIBLNCNTR1	R/W	H'FF40 10B0	H'1F40 10B0	32
	块 xn 次计数器 1	SSIBLNCNT1	R	H'FF40 10B8	H'1F40 10B8	32
2	DMA 模式寄存器 2	SSIDMMR2	R/W	H'FF40 10C0	H'1F40 10C0	32
	RDMA 传送源地址寄存器 2	SSIRDMAADR2	R/W	H'FF40 10C8	H'1F40 10C8	32
	RDMA 传送字节数寄存器 2	SSIRDMCNTR2	R/W	H'FF40 10D0	H'1F40 10D0	32
	WDMA 传送目标地址寄存器 2	SSIWDMADR2	R/W	H'FF40 10D8	H'1F40 10D8	32
	WDMA 传送字节数寄存器 2	SSIWDMCNTR2	R/W	H'FF40 10E0	H'1F40 10E0	32
	DMA 控制寄存器 2	SSIDMCOR2	R/W	H'FF40 10E8	H'1F40 10E8	32
	发送暂停块计数器 2	SSISTPBLCNT2	R/W	H'FF40 10F0	H'1F40 10F0	32
	发送暂停中传送数据寄存器 2	SSISTPDR2	R/W	H'FF40 10F8	H'1F40 10F8	32
	块计数源寄存器 2	SSIBLCNTR2	R/W	H'FF40 1100	H'1F40 1100	32
	块计数器 2	SSIBLCNT2	R	H'FF40 1108	H'1F40 1108	32
	块 xn 次中断计数源寄存器 2	SSIBLNCNTR2	R/W	H'FF40 1110	H'1F40 1110	32
	块 xn 次计数器 2	SSIBLNCNT2	R	H'FF40 1118	H'1F40 1118	32
0 ~ 2 共用	DMA 操作寄存器 0	SSIDMAOR0	R/W	H'FF40 1180	H'1F40 1180	32
	中断状态寄存器 0	SSIDMINTSR0	R/W	H'FF40 1188	H'1F40 1188	32
	中断屏蔽寄存器 0	SSIDMINTMR0	R/W	H'FF40 1190	H'1F40 1190	32

表 18.3 SSI_DMACH0 的各处理模式的寄存器状态

通道	名称	略称	上电复位	睡眠	待机
0	DMA 模式寄存器 0	SSIDMMR0	H'0000 0000	保持	保持
	RDMA 传送源地址寄存器 0	SSIRDMAADR0	H'0000 0000	保持	保持
	RDMA 传送字节数寄存器 0	SSIRDMCNTR0	H'0000 0000	保持	保持
	WDMA 传送目标地址寄存器 0	SSIWDMADR0	H'0000 0000	保持	保持
	WDMA 传送字节数寄存器 0	SSIWDMCNTR0	H'0000 0000	保持	保持
	DMA 控制寄存器 0	SSIDMCOR0	H'0000 0000	保持	保持
	发送暂停块计数器 0	SSISTPBLCNT0	H'0000 0000	保持	保持
	发送暂停中传送数据寄存器 0	SSISTPDR0	H'0000 0000	保持	保持
	块计数源寄存器 0	SSIBLCNTR0	H'0000 0000	保持	保持
	块计数器 0	SSIBLCNT0	H'0000 0000	保持	保持
	块 xn 次中断计数源寄存器 0	SSIBLNCNTR0	H'0000 0000	保持	保持
	块 xn 次计数器 0	SSIBLNCNT0	H'0000 0000	保持	保持
1	DMA 模式寄存器 1	SSIDMMR1	H'0000 0000	保持	保持
	RDMA 传送源地址寄存器 1	SSIRDMAADR1	H'0000 0000	保持	保持
	RDMA 传送字节数寄存器 1	SSIRDMCNTR1	H'0000 0000	保持	保持
	WDMA 传送目标地址寄存器 1	SSIWDMADR1	H'0000 0000	保持	保持
	WDMA 传送字节数寄存器 1	SSIWDMCNTR1	H'0000 0000	保持	保持
	DMA 控制寄存器 1	SSIDMCOR1	H'0000 0000	保持	保持
	发送暂停块计数器 1	SSISTPBLCNT1	H'0000 0000	保持	保持
	发送暂停中传送数据寄存器 1	SSISTPDR1	H'0000 0000	保持	保持
	块计数源寄存器 1	SSIBLCNTR1	H'0000 0000	保持	保持
	块计数器 1	SSIBLCNT1	H'0000 0000	保持	保持
	块 xn 次中断计数源寄存器 1	SSIBLNCNTR1	H'0000 0000	保持	保持
	块 xn 次计数器 1	SSIBLNCNT1	H'0000 0000	保持	保持
2	DMA 模式寄存器 2	SSIDMMR2	H'0000 0000	保持	保持
	RDMA 传送源地址寄存器 2	SSIRDMAADR2	H'0000 0000	保持	保持
	RDMA 传送字节数寄存器 2	SSIRDMCNTR2	H'0000 0000	保持	保持
	WDMA 传送目标地址寄存器 2	SSIWDMADR2	H'0000 0000	保持	保持
	WDMA 传送字节数寄存器 2	SSIWDMCNTR2	H'0000 0000	保持	保持
	DMA 控制寄存器 2	SSIDMCOR2	H'0000 0000	保持	保持
	发送暂停块计数器 2	SSISTPBLCNT2	H'0000 0000	保持	保持
	发送暂停中传送数据寄存器 2	SSISTPDR2	H'0000 0000	保持	保持
	块计数源寄存器 2	SSIBLCNTR2	H'0000 0000	保持	保持
	块计数器 2	SSIBLCNT2	H'0000 0000	保持	保持
	块 xn 次中断计数源寄存器 2	SSIBLNCNTR2	H'0000 0000	保持	保持
	块 xn 次计数器 2	SSIBLNCNT2	H'0000 0000	保持	保持
0 ~ 2 共用	DMA 操作寄存器 0	SSIDMAOR0	H'0000 0000	保持	保持
	中断状态寄存器 0	SSIDMINTSR0	H'0101 0101	保持	保持
	中断屏蔽寄存器 0	SSIDMINTMR0	H'1F1F 1F1F	保持	保持

SSI_DMACH1 的寄存器结构如表 18.4 所示, 各处理模式的寄存器状态如表 18.5 所示。

表 18.4 SSI_DMACH1 的寄存器结构

通道	名称	略称	R/W	P4 区地址	区域 7 地址	长度
3	DMA 模式寄存器 3	SSIDMMR3	R/W	H'FF50 1000	H'1F50 1000	32
	RDMA 传送源地址寄存器 3	SSIRDMAADR3	R/W	H'FF50 1008	H'1F50 1008	32
	RDMA 传送字节数寄存器 3	SSIRDMCNTR3	R/W	H'FF50 1010	H'1F50 1010	32
	WDMA 传送目标地址寄存器 3	SSIWDMADR3	R/W	H'FF50 1018	H'1F50 1018	32
	WDMA 传送字节数寄存器 3	SSIWDMCNTR3	R/W	H'FF50 1020	H'1F50 1020	32
	DMA 控制寄存器 3	SSIDMCOR3	R/W	H'FF50 1028	H'1F50 1028	32
	发送暂停块计数器 3	SSISTPBLCNT3	R/W	H'FF50 1030	H'1F50 1030	32
	发送暂停中传送数据寄存器 3	SSISTPDR3	R/W	H'FF50 1038	H'1F50 1038	32
	块计数源寄存器 3	SSIBLCNTR3	R/W	H'FF50 1040	H'1F50 1040	32
	块计数器 3	SSIBLCNT3	R	H'FF50 1048	H'1F50 1048	32
	块 xn 次中断计数源寄存器 3	SSIBLNCNTR3	R/W	H'FF50 1050	H'1F50 1050	32
	块 xn 次计数器 3	SSIBLNCNT3	R	H'FF50 1058	H'1F50 1058	32
4	DMA 模式寄存器 4	SSIDMMR4	R/W	H'FF50 1060	H'1F50 1060	32
	RDMA 传送源地址寄存器 4	SSIRDMAADR4	R/W	H'FF50 1068	H'1F50 1068	32
	RDMA 传送字节数寄存器 4	SSIRDMCNTR4	R/W	H'FF50 1070	H'1F50 1070	32
	WDMA 传送目标地址寄存器 4	SSIWDMADR4	R/W	H'FF50 1078	H'1F50 1078	32
	WDMA 传送字节数寄存器 4	SSIWDMCNTR4	R/W	H'FF50 1080	H'1F50 1080	32
	DMA 控制寄存器 4	SSIDMCOR4	R/W	H'FF50 1088	H'1F50 1088	32
	发送暂停块计数器 4	SSISTPBLCNT4	R/W	H'FF50 1090	H'FF50 1090	32
	发送暂停中传送数据寄存器 4	SSISTPDR4	R/W	H'FF50 1098	H'FF50 1098	32
	块计数源寄存器 4	SSIBLCNTR4	R/W	H'FF50 10A0	H'FF50 10A0	32
	块计数器 4	SSIBLCNT4	R	H'FF50 10A8	H'FF50 10A8	32
	块 xn 次中断计数源寄存器 4	SSIBLNCNTR4	R/W	H'FF50 10B0	H'FF50 10B0	32
	块 xn 次计数器 4	SSIBLNCNT4	R	H'FF50 10B8	H'FF50 10B8	32
5	DMA 模式寄存器 5	SSIDMMR5	R/W	H'FF50 10C0	H'1F50 10C0	32
	RDMA 传送源地址寄存器 5	SSIRDMAADR5	R/W	H'FF50 10C8	H'1F50 10C8	32
	RDMA 传送字节数寄存器 5	SSIRDMCNTR5	R/W	H'FF50 10D0	H'1F50 10D0	32
	WDMA 传送目标地址寄存器 5	SSIWDMADR5	R/W	H'FF50 10D8	H'1F50 10D8	32
	WDMA 传送字节数寄存器 5	SSIWDMCNTR5	R/W	H'FF50 10E0	H'1F50 10E0	32
	DMA 控制寄存器 5	SSIDMCOR5	R/W	H'FF50 10E8	H'1F50 10E8	32
	发送暂停块计数器 5	SSISTPBLCNT5	R/W	H'FF50 10F0	H'FF50 10F0	32
	发送暂停中传送数据寄存器 5	SSISTPDR5	R/W	H'FF50 10F8	H'1F50 10F8	32
	块计数源寄存器 5	SSIBLCNTR5	R/W	H'FF50 1100	H'1F50 1100	32
	块计数器 5	SSIBLCNT5	R	H'FF50 1108	H'1F50 1108	32
	块 xn 次中断计数源寄存器 5	SSIBLNCNTR5	R/W	H'FF50 1110	H'1F50 1110	32
	块 xn 次计数器 5	SSIBLNCNT5	R	H'FF50 1118	H'1F50 1118	32
3 ~ 5 共用	DMA 操作寄存器 1	SSIDMAOR1	R/W	H'1F50 1180	H'1F50 1180	32
	中断状态寄存器 1	SSIDMINTSR1	R/W	H'1F50 1188	H'1F50 1188	32
	中断屏蔽寄存器 1	SSIDMINTMR1	R/W	H'1F50 1190	H'1F50 1190	32

表 18.5 SSI_DMACH1 的各处理模式的寄存器状态

通道	名称	略称	上电复位	睡眠	待机
3	DMA 模式寄存器 3	SSIDMMR3	H'0000 0000	保持	保持
	RDMA 传送源地址寄存器 3	SSIRDMAADR3	H'0000 0000	保持	保持
	RDMA 传送字节数寄存器 3	SSIRDMCNTR3	H'0000 0000	保持	保持
	WDMA 传送目标地址寄存器 3	SSIWDMADR3	H'0000 0000	保持	保持
	WDMA 传送字节数寄存器 3	SSIWDMCNTR3	H'0000 0000	保持	保持
	DMA 控制寄存器 3	SSIDMCOR3	H'0000 0000	保持	保持
	发送暂停块计数器 3	SSISTPBLCNT3	H'0000 0000	保持	保持
	发送暂停中传送数据寄存器 3	SSISTPDR3	H'0000 0000	保持	保持
	块计数源寄存器 3	SSIBLCNTR3	H'0000 0000	保持	保持
	块计数器 3	SSIBLCNT3	H'0000 0000	保持	保持
	块 xn 次中断计数源寄存器 3	SSIBLNCNTR3	H'0000 0000	保持	保持
	块 xn 次计数器 3	SSIBLNCNT3	H'0000 0000	保持	保持
4	DMA 模式寄存器 4	SSIDMMR4	H'0000 0000	保持	保持
	RDMA 传送源地址寄存器 4	SSIRDMAADR4	H'0000 0000	保持	保持
	RDMA 传送字节数寄存器 4	SSIRDMCNTR4	H'0000 0000	保持	保持
	WDMA 传送目标地址寄存器 4	SSIWDMADR4	H'0000 0000	保持	保持
	WDMA 传送字节数寄存器 4	SSIWDMCNTR4	H'0000 0000	保持	保持
	DMA 控制寄存器 4	SSIDMCOR4	H'0000 0000	保持	保持
	发送暂停块计数器 4	SSISTPBLCNT4	H'0000 0000	保持	保持
	发送暂停中传送数据寄存器 4	SSISTPDR4	H'0000 0000	保持	保持
	块计数源寄存器 4	SSIBLCNTR4	H'0000 0000	保持	保持
	块计数器 4	SSIBLCNT4	H'0000 0000	保持	保持
	块 xn 次中断计数源寄存器 4	SSIBLNCNTR4	H'0000 0000	保持	保持
	块 xn 次计数器 4	SSIBLNCNT4	H'0000 0000	保持	保持
5	DMA 模式寄存器 5	SSIDMMR5	H'0000 0000	保持	保持
	RDMA 传送源地址寄存器 5	SSIRDMAADR5	H'0000 0000	保持	保持
	RDMA 传送字节数寄存器 5	SSIRDMCNTR5	H'0000 0000	保持	保持
	WDMA 传送目标地址寄存器 5	SSIWDMADR5	H'0000 0000	保持	保持
	WDMA 传送字节数寄存器 5	SSIWDMCNTR5	H'0000 0000	保持	保持
	DMA 控制寄存器 5	SSIDMCOR5	H'0000 0000	保持	保持
	发送暂停块计数器 5	SSISTPBLCNT5	H'0000 0000	保持	保持
	发送暂停中传送数据寄存器 5	SSISTPDR5	H'0000 0000	保持	保持
	块计数源寄存器 5	SSIBLCNTR5	H'0000 0000	保持	保持
	块计数器 5	SSIBLCNT5	H'0000 0000	保持	保持
	块 xn 次中断计数源寄存器 5	SSIBLNCNTR5	H'0000 0000	保持	保持
	块 xn 次计数器 5	SSIBLNCNT5	H'0000 0000	保持	保持
3 ~ 5 共用	DMA 操作寄存器 1	SSIDMAOR1	H'0000 0000	保持	保持
	中断状态寄存器 1	SSIDMINTSR1	H'0101 0101	保持	保持
	中断屏蔽寄存器 1	SSIDMINTMR1	H'1F1F 1F1F	保持	保持

SSI_CH0 ~ 5 的寄存器结构如表 18.6 所示，各处理模式的寄存器状态如表 18.7 所示。

表 18.6 SSI_CH0 ~ 5 的寄存器结构

通道	名称	略称	R/W	P4 区地址	区域 7 地址	长度
0	控制寄存器 0	SSICR0	R/W	H'FF40 2000	H'1F40 2000	32
	状态寄存器 0	SSISR0	R/W*	H'FF40 2004	H'1F40 2004	32
	发送数据寄存器 0	SSITDR0	R/W	H'FF40 2008	H'1F40 2008	32
	接收数据寄存器 0	SSIRDR0	R	H'FF40 200C	H'1F40 200C	32
1	控制寄存器 1	SSICR1	R/W	H'FF40 3000	H'1F40 3000	32
	状态寄存器 1	SSISR1	R/W*	H'FF40 3004	H'1F40 3004	32
	发送数据寄存器 1	SSITDR1	R/W	H'FF40 3008	H'1F40 3008	32
	接收数据寄存器 1	SSIRDR1	R	H'FF40 300C	H'1F40 300C	32
2	控制寄存器 2	SSICR2	R/W	H'FF40 4000	H'1F40 4000	32
	状态寄存器 2	SSISR2	R/W*	H'FF40 4004	H'1F40 4004	32
	发送数据寄存器 2	SSITDR2	R/W	H'FF40 4008	H'1F40 4008	32
	接收数据寄存器 2	SSIRDR2	R	H'FF40 400C	H'1F40 400C	32
3	控制寄存器 3	SSICR3	R/W	H'FF50 2000	H'1F50 2000	32
	状态寄存器 3	SSISR3	R/W*	H'FF50 2004	H'1F50 2004	32
	发送数据寄存器 3	SSITDR3	R/W	H'FF50 2008	H'1F50 2008	32
	接收数据寄存器 3	SSIRDR3	R	H'FF50 200C	H'1F50 200C	32
4	控制寄存器 4	SSICR4	R/W	H'FF50 3000	H'1F50 3000	32
	状态寄存器 4	SSISR4	R/W*	H'FF50 3004	H'1F50 3004	32
	发送数据寄存器 4	SSITDR4	R/W	H'FF50 3008	H'1F50 3008	32
	接收数据寄存器 4	SSIRDR4	R	H'FF50 300C	H'1F50 300C	32
5	控制寄存器 5	SSICR5	R/W	H'FF50 4000	H'1F50 4000	32
	状态寄存器 5	SSISR5	R/W*	H'FF50 4004	H'1F50 4004	32
	发送数据寄存器 5	SSITDR5	R/W	H'FF50 4008	H'1F50 4008	32
	接收数据寄存器 5	SSIRDR5	R	H'FF50 400C	H'1F50 400C	32

【注】 * 此寄存器的 bit26 和 bit27 位是可读写位，其他都是只写位。
详细内容请参照“18.3.17 状态寄存器 0~5 (SSISR0~5)”。

表 18.7 SSI_CH0 ~ 5 的各处理模式的寄存器状态

通道	名称	略称	上电复位	睡眠	待机
0	控制寄存器 0	SSICR0	H'0000 0000	保持	保持
	状态寄存器 0	SSISR0	H'0210 A003	保持	保持
	发送数据寄存器 0	SSITDR0	H'0000 0000	保持	保持
	接收数据寄存器 0	SSIRDR0	H'0000 0000	保持	保持
1	控制寄存器 1	SSICR1	H'0000 0000	保持	保持
	状态寄存器 1	SSISR1	H'0210 A003	保持	保持
	发送数据寄存器 1	SSITDR1	H'0000 0000	保持	保持
	接收数据寄存器 1	SSIRDR1	H'0000 0000	保持	保持
2	控制寄存器 2	SSICR2	H'0000 0000	保持	保持
	状态寄存器 2	SSISR2	H'0210 A003	保持	保持
	发送数据寄存器 2	SSITDR2	H'0000 0000	保持	保持
	接收数据寄存器 2	SSIRDR2	H'0000 0000	保持	保持
3	控制寄存器 3	SSICR3	H'0000 0000	保持	保持
	状态寄存器 3	SSISR3	H'0210 A003	保持	保持
	发送数据寄存器 3	SSITDR3	H'0000 0000	保持	保持
	接收数据寄存器 3	SSIRDR3	H'0000 0000	保持	保持
4	控制寄存器 4	SSICR4	H'0000 0000	保持	保持
	状态寄存器 4	SSISR4	H'0210 A003	保持	保持
	发送数据寄存器 4	SSITDR4	H'0000 0000	保持	保持
	接收数据寄存器 4	SSIRDR4	H'0000 0000	保持	保持
5	控制寄存器 5	SSICR5	H'0000 0000	保持	保持
	状态寄存器 5	SSISR5	H'0210 A003	保持	保持
	发送数据寄存器 5	SSITDR5	H'0000 0000	保持	保持
	接收数据寄存器 5	SSIRDR5	H'0000 0000	保持	保持

18.3.1 DMA 模式寄存器 0 ~ 5 (SSIDMMR0 ~ 5)

SSIDMMR0 ~ 5 是 32 位可读写寄存器，设定 SSI_DMAR 的运行模式。

在硬件复位、软件复位和 SSI_DMAR 的软件复位 (SSIDMCOR0 ~ 3 的 DMRST) 时，此寄存器的值被初始化。

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

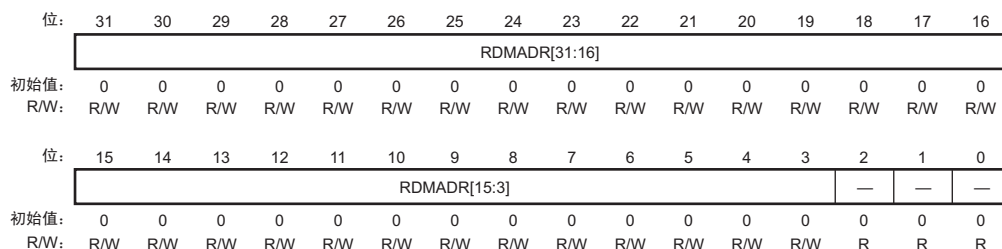
位	位名	初始值	R/W	说明
31 ~ 19	—	全 0	R	保留位 读取值为 0，只能写 0。
18	RDSAM	0	R/W	RDMA 传送源地址模式位 设定 RDMA 传送时的传送源地址的增减。 0: 传送源地址递增 (+4) 1: 传送源地址递减 (-4)
17	—	0	R	保留位 读取值为 0，只能写 0。
16	WDDAM	0	R/W	WDMA 传送目的地址模式位 设定 WDMA 传送时的传送目的地址的增减。 0: 传送目的地址递增 (+4) 1: 传送目的地址递减 (-4)
15 ~ 8	—	全 0	R	保留位 读取值为 0，只能写 0。
7、6	RDMBSZ[1:0]	00	R/W	RDMA 最大突发长度位 设定 RDMA 数据传送时的最大突发长度。 00: 1 个突发 (8 字节) 01: 2 个突发 (16 字节) 10: 4 个突发 (32 字节) 11: 禁止设定
5、4	WDMBSZ[1:0]	00	R/W	WDMA 最大突发长度位 设定 WDMA 数据传送时的最大突发长度。 00: 1 个突发 (8 字节) 01: 2 个突发 (16 字节) 10: 4 个突发 (32 字节) 11: 禁止设定
3 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

18.3.2 RDMA 传送源地址寄存器 0 ~ 5 (SSIRDMDR0 ~ 5)

SSIRDMDR0 ~ 5 是 32 位可读写寄存器，设定 RDMA 传送时的数据传送源的存储器地址。

在硬件复位、软件复位和 SSI_DMIC 的软件复位 (SSIDMCOR0 ~ 3 的 DMRST) 时，此寄存器的值被初始化。

只能在 SSIDMCOR0 ~ 5 的 DMEN=0 时写此寄存器。



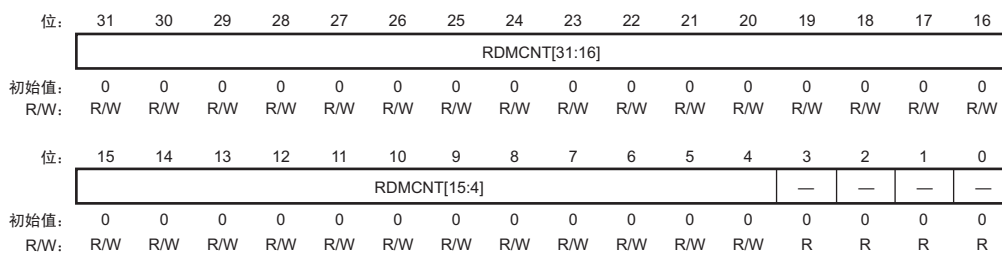
位	位名	初始值	R/W	说明
31 ~ 3	RDMDR[31:3]	全 0	R/W	RDMA 传送源地址 设定 RDMA 传送时的数据传送源的存储器地址。
2 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

18.3.3 RDMA 传送字节数寄存器 0 ~ 5 (SSIRDMCNTR0 ~ 5)

SSIRDMCNTR0 ~ 5 是 32 位可读写寄存器，设定 RDMA 传送时的数据传送字节数。

在硬件复位、软件复位和 SSI_DMIC 的软件复位 (SSIDMCOR0 ~ 3 的 DMRST) 时，此寄存器的值被初始化。

只能在 SSIDMCOR0 ~ 5 的 DMEN=0 时写此寄存器。



位	位名	初始值	R/W	说明
31 ~ 4	RDMCNT[31:4]	全 0	R/W	RDMA 传送字节数 设定 RDMA 传送时的数据传送字节数。 必须根据 RDMA 最大突发长度设定以下值： 1 个突发：8×n[H'08×n] (字节) 2 个突发：16×n[H'10×n] (字节) 4 个突发：32×n[H'20×n] (字节)
3 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

18.3.4 WDMA 传送目的地址寄存器 0 ~ 5 (SSIWDMADR0 ~ 5)

SSIWDMADR0 ~ 5 是 32 位可读写寄存器，设定 WDMA 传送时的数据传送目标的存储器地址。

在硬件复位、软件复位和 SSI_DMIC 的软件复位 (SSIDMCOR0 ~ 3 的 DMRST) 时，此寄存器的值被初始化。

只能在 SSIDMCOR0 ~ 5 的 DMEN=0 时写此寄存器。

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

位	位名	初始值	R/W	说明
31 ~ 3	WDMADR[31:3]	全 0	R/W	WDMA 传送目的地址 设定 WDMA 传送时的数据传送目标的存储器地址。
2 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

18.3.5 WDMA 传送字节数寄存器 0 ~ 5 (SSIWDMCNT0 ~ 5)

SSIWDMCNT0 ~ 5 是 32 位可读写寄存器，设定 WDMA 传送时的数据传送字节数。

在硬件复位、软件复位和 SSI_DMIC 的软件复位 (SSIDMCOR0 ~ 3 的 DMRST) 时，此寄存器的值被初始化。

只能在 SSIDMCOR0 ~ 5 的 DMEN=0 时写此寄存器。

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

位	位名	初始值	R/W	说明
31 ~ 4	WDMCNT[31:4]	全 0	R/W	WDMA 传送字节数 设定 WDMA 传送时的数据传送字节数。 必须根据 WDMA 最大突发长度设定以下值： 1 个突发：8×n[H'08×n] (字节) 2 个突发：16×n[H'10×n] (字节) 4 个突发：32×n[H'20×n] (字节)
3 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

18.3.6 DMA 控制寄存器 0 ~ 5 (SSIDMCOR0 ~ 5)

SSIDMCOR0 ~ 5 是 32 位可读写寄存器，控制 SSI_DMACH0/1 的运行和停止以及选择时钟等。

在硬件复位、软件复位和 SSI_DMACH 的软件复位 (SSIDMCOR0 ~ 3 的 DMRST) 时，此寄存器的值被初始化。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DM RST	TX RST	RX RST	—	—	—	—	—	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	R/W	R/W	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
	—	—	—	SCKSOP[2:0]			SCKSIP[2:0]			SCKS[2:0]			—	RPT MD	TRMD	DMEN
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31	DMRST	0	W	SSI_DMACH0/1 软件复位 将对应 SSI_CH0 ~ 5 的 SSI_DMACH0/1 进行复位，并且中止传送。 读取值为 0。 0: 软件复位无效 1: 软件复位有效
30	TXRST	0	R/W	发送 FIFO 缓冲器复位 对发送 FIFO 缓冲器进行复位。 0: 发送 FIFO 缓冲器复位无效 1: 发送 FIFO 缓冲器复位有效
29	RXRST	0	R/W	接收 FIFO 缓冲器复位 对接收 FIFO 缓冲器进行复位。 0: 接收 FIFO 缓冲器复位无效 1: 接收 FIFO 缓冲器复位有效
28 ~ 13	—	全 0	R	保留位 读取值为 0，只能写 0。
12 ~ 10	SCKSOP[2:0]	000	R/W	串行位时钟和串行字选择信号的输出选择位 选择从 SSI_CH0 ~ 5 输出到外部的串行位时钟和串行字选择信号。 SSIDMCOR0 的情况 000: 输出 SSISCK0/SSIWS0=SSI_CH0 的串行位时钟 / 字选择信号 001: 输出 SSISCK0/SSIWS0=SSI_CH1 的串行位时钟 / 字选择信号 010: 输出 SSISCK0/SSIWS0=SSI_CH2 的串行位时钟 / 字选择信号 100: 输出 SSISCK0/SSIWS0=SSI_CH3 的串行位时钟 / 字选择信号 101: 输出 SSISCK0/SSIWS0=SSI_CH4 的串行位时钟 / 字选择信号 110: 输出 SSISCK0/SSIWS0=SSI_CH5 的串行位时钟 / 字选择信号 011: 禁止设定 111: 禁止设定

位	位名	初始值	R/W	说明
12 ~ 10	SCKSOP[2:0]	000	R/W	<p>SSIDMCOR1 的情况</p> <p>000: 输出 SSISCK1/SSIWS1=SSI_CH1 的串行位时钟 / 字选择信号 001: 输出 SSISCK1/SSIWS1=SSI_CH2 的串行位时钟 / 字选择信号 011: 输出 SSISCK1/SSIWS1=SSI_CH3 的串行位时钟 / 字选择信号 100: 输出 SSISCK1/SSIWS1=SSI_CH4 的串行位时钟 / 字选择信号 101: 输出 SSISCK1/SSIWS1=SSI_CH5 的串行位时钟 / 字选择信号 111: 输出 SSISCK1/SSIWS1=SSI_CH0 的串行位时钟 / 字选择信号 010: 禁止设定 110: 禁止设定</p> <p>SSIDMCOR2 的情况</p> <p>000: 输出 SSISCK2/SSIWS2=SSI_CH2 的串行位时钟 / 字选择信号 010: 输出 SSISCK2/SSIWS2=SSI_CH3 的串行位时钟 / 字选择信号 011: 输出 SSISCK2/SSIWS2=SSI_CH4 的串行位时钟 / 字选择信号 100: 输出 SSISCK2/SSIWS2=SSI_CH5 的串行位时钟 / 字选择信号 110: 输出 SSISCK2/SSIWS2=SSI_CH0 的串行位时钟 / 字选择信号 111: 输出 SSISCK2/SSIWS2=SSI_CH1 的串行位时钟 / 字选择信号 001: 禁止设定 101: 禁止设定</p> <p>SSIDMCOR3 的情况</p> <p>000: 输出 SSISCK3/SSIWS3=SSI_CH3 的串行位时钟 / 字选择信号 001: 输出 SSISCK3/SSIWS3=SSI_CH4 的串行位时钟 / 字选择信号 010: 输出 SSISCK3/SSIWS3=SSI_CH5 的串行位时钟 / 字选择信号 100: 输出 SSISCK3/SSIWS3=SSI_CH0 的串行位时钟 / 字选择信号 101: 输出 SSISCK3/SSIWS3=SSI_CH1 的串行位时钟 / 字选择信号 110: 输出 SSISCK3/SSIWS3=SSI_CH2 的串行位时钟 / 字选择信号 011: 禁止设定 111: 禁止设定</p> <p>SSIDMCOR4 的情况</p> <p>000: 输出 SSISCK4/SSIWS4=SSI_CH4 的串行位时钟 / 字选择信号 001: 输出 SSISCK4/SSIWS4=SSI_CH5 的串行位时钟 / 字选择信号 011: 输出 SSISCK4/SSIWS4=SSI_CH0 的串行位时钟 / 字选择信号 100: 输出 SSISCK4/SSIWS4=SSI_CH1 的串行位时钟 / 字选择信号 101: 输出 SSISCK4/SSIWS4=SSI_CH2 的串行位时钟 / 字选择信号 111: 输出 SSISCK4/SSIWS4=SSI_CH3 的串行位时钟 / 字选择信号 010: 禁止设定 110: 禁止设定</p> <p>SSIDMCOR5 的情况</p> <p>000: 输出 SSISCK5/SSIWS5=SSI_CH5 的串行位时钟 / 字选择信号 010: 输出 SSISCK5/SSIWS5=SSI_CH0 的串行位时钟 / 字选择信号 011: 输出 SSISCK5/SSIWS5=SSI_CH1 的串行位时钟 / 字选择信号 100: 输出 SSISCK5/SSIWS5=SSI_CH2 的串行位时钟 / 字选择信号 110: 输出 SSISCK5/SSIWS5=SSI_CH3 的串行位时钟 / 字选择信号 111: 输出 SSISCK5/SSIWS5=SSI_CH4 的串行位时钟 / 字选择信号 001: 禁止设定 101: 禁止设定</p>

位	位名	初始值	R/W	说明
9 ~ 7	SCKSIP[2:0]	000	R/W	<p>串行位时钟和串行字选择信号的输出选择位 选择从外部输入到 SSI_CH0 ~ 5 的串行位时钟和串行字选择信号。</p> <p>SSIDMCOR0 的情况</p> <p>000: 输入 SSI_CH0 的串行位时钟 / 字选择信号 =SSISCK0/SSIWS0 001: 输入 SSI_CH0 的串行位时钟 / 字选择信号 =SSISCK1/SSIWS1 010: 输入 SSI_CH0 的串行位时钟 / 字选择信号 =SSISCK2/SSIWS2 100: 输入 SSI_CH0 的串行位时钟 / 字选择信号 =SSISCK3/SSIWS3 101: 输入 SSI_CH0 的串行位时钟 / 字选择信号 =SSISCK4/SSIWS4 110: 输入 SSI_CH0 的串行位时钟 / 字选择信号 =SSISCK5/SSIWS5 011: 禁止设定 111: 禁止设定</p> <p>SSIDMCOR1 的情况</p> <p>000: 输入 SSI_CH1 的串行位时钟 / 字选择信号 =SSISCK1/SSIWS1 001: 输入 SSI_CH1 的串行位时钟 / 字选择信号 =SSISCK2/SSIWS2 011: 输入 SSI_CH1 的串行位时钟 / 字选择信号 =SSISCK3/SSIWS3 100: 输入 SSI_CH1 的串行位时钟 / 字选择信号 =SSISCK4/SSIWS4 101: 输入 SSI_CH1 的串行位时钟 / 字选择信号 =SSISCK5/SSIWS5 111: 输入 SSI_CH1 的串行位时钟 / 字选择信号 =SSISCK0/SSIWS0 010: 禁止设定 110: 禁止设定</p> <p>SSIDMCOR2 的情况</p> <p>000: 输入 SSI_CH2 的串行位时钟 / 字选择信号 =SSISCK2/SSIWS2 010: 输入 SSI_CH2 的串行位时钟 / 字选择信号 =SSISCK3/SSIWS3 011: 输入 SSI_CH2 的串行位时钟 / 字选择信号 =SSISCK4/SSIWS4 100: 输入 SSI_CH2 的串行位时钟 / 字选择信号 =SSISCK5/SSIWS5 110: 输入 SSI_CH2 的串行位时钟 / 字选择信号 =SSISCK0/SSIWS0 111: 输入 SSI_CH2 的串行位时钟 / 字选择信号 =SSISCK1/SSIWS1 001: 禁止设定 101: 禁止设定</p> <p>SSIDMCOR3 的情况</p> <p>000: 输入 SSI_CH3 的串行位时钟 / 字选择信号 =SSISCK3/SSIWS3 001: 输入 SSI_CH3 的串行位时钟 / 字选择信号 =SSISCK4/SSIWS4 010: 输入 SSI_CH3 的串行位时钟 / 字选择信号 =SSISCK5/SSIWS5 100: 输入 SSI_CH3 的串行位时钟 / 字选择信号 =SSISCK0/SSIWS0 101: 输入 SSI_CH3 的串行位时钟 / 字选择信号 =SSISCK1/SSIWS1 110: 输入 SSI_CH3 的串行位时钟 / 字选择信号 =SSISCK2/SSIWS2 011: 禁止设定 111: 禁止设定</p>

位	位名	初始值	R/W	说明
9 ~ 7	SCKSIP[2:0]	000	R/W	<p>SSIDMCOR4 的情况</p> <p>000: 输入 SSI_CH4 的串行位时钟 / 字选择信号 =SSISCK4/SSIWS4 001: 输入 SSI_CH4 的串行位时钟 / 字选择信号 =SSISCK5/SSIWS5 011: 输入 SSI_CH4 的串行位时钟 / 字选择信号 =SSISCK0/SSIWS0 100: 输入 SSI_CH4 的串行位时钟 / 字选择信号 =SSISCK1/SSIWS1 101: 输入 SSI_CH4 的串行位时钟 / 选择信号 =SSISCK2/SSIWS2 111: 输入 SSI_CH4 的串行位时钟 / 字选择信号 =SSISCK3/SSIWS3 010: 禁止设定 110: 禁止设定</p> <p>SSIDMCOR5 的情况</p> <p>000: 输入 SSI_CH5 的串行位时钟 / 字选择信号 =SSISCK5/SSIWS5 010: 输入 SSI_CH5 的串行位时钟 / 字选择信号 =SSISCK0/SSIWS0 011: 输入 SSI_CH5 的串行位时钟 / 字选择信号 =SSISCK1/SSIWS1 100: 输入 SSI_CH5 的串行位时钟 / 字选择信号 =SSISCK2/SSIWS2 110: 输入 SSI_CH5 的串行位时钟 / 字选择信号 =SSISCK3/SSIWS3 111: 输入 SSI_CH5 的串行位时钟 / 字选择信号 =SSISCK4/SSIWS4 001: 禁止设定 101: 禁止设定</p>
6 ~ 4	SCKS[2:0]	000	R/W	<p>音频时钟选择位</p> <p>选择从外部输入到 SSI_CH0 ~ 5 的音频时钟。</p> <p>SSIDMCOR0 的情况</p> <p>000: 输入 SSI_CH0 的音频时钟 =AUDIO_CLK0 001: 输入 SSI_CH0 的音频时钟 =AUDIO_CLK1 010: 输入 SSI_CH0 的音频时钟 =AUDIO_CLK2 100: 输入 SSI_CH0 的音频时钟 =AUDIO_CLK3 101: 输入 SSI_CH0 的音频时钟 =AUDIO_CLK4 110: 输入 SSI_CH0 的音频时钟 =AUDIO_CLK5 011: 禁止设定 111: 禁止设定</p> <p>SSIDMCOR1 的情况</p> <p>000: 输入 SSI_CH1 的音频时钟 =AUDIO_CLK1 001: 输入 SSI_CH1 的音频时钟 =AUDIO_CLK2 011: 输入 SSI_CH1 的音频时钟 =AUDIO_CLK3 100: 输入 SSI_CH1 的音频时钟 =AUDIO_CLK4 101: 输入 SSI_CH1 的音频时钟 =AUDIO_CLK5 111: 输入 SSI_CH1 的音频时钟 =AUDIO_CLK0 010: 禁止设定 110: 禁止设定</p>

位	位名	初始值	R/W	说明
6 ~ 4	SCKS[2:0]	000	R/W	<p>SSIDMCOR2 的情况</p> <p>000: 输入 SSI_CH2 的音频时钟 =AUDIO_CLK2 001: 输入 SSI_CH2 的音频时钟 =AUDIO_CLK3 010: 输入 SSI_CH2 的音频时钟 =AUDIO_CLK4 100: 输入 SSI_CH2 的音频时钟 =AUDIO_CLK5 101: 输入 SSI_CH2 的音频时钟 =AUDIO_CLK0 110: 输入 SSI_CH2 的音频时钟 =AUDIO_CLK1 011: 禁止设定 111: 禁止设定</p> <p>SSIDMCOR3 的情况</p> <p>000: 输入 SSI_CH3 的音频时钟 =AUDIO_CLK3 001: 输入 SSI_CH3 的音频时钟 =AUDIO_CLK4 011: 输入 SSI_CH3 的音频时钟 =AUDIO_CLK5 100: 输入 SSI_CH3 的音频时钟 =AUDIO_CLK0 101: 输入 SSI_CH3 的音频时钟 =AUDIO_CLK1 111: 输入 SSI_CH3 的音频时钟 =AUDIO_CLK2 010: 禁止设定 110: 禁止设定</p> <p>SSIDMCOR4 的情况</p> <p>000: 输入 SSI_CH4 的音频时钟 =AUDIO_CLK4 001: 输入 SSI_CH4 的音频时钟 =AUDIO_CLK5 011: 输入 SSI_CH4 的音频时钟 =AUDIO_CLK0 100: 输入 SSI_CH4 的音频时钟 =AUDIO_CLK1 101: 输入 SSI_CH4 的音频时钟 =AUDIO_CLK2 111: 输入 SSI_CH4 的音频时钟 =AUDIO_CLK3 010: 禁止设定 110: 禁止设定</p> <p>SSIDMCOR5 的情况</p> <p>000: 输入 SSI_CH5 的音频时钟 =AUDIO_CLK5 010: 输入 SSI_CH5 的音频时钟 =AUDIO_CLK0 011: 输入 SSI_CH5 的音频时钟 =AUDIO_CLK1 100: 输入 SSI_CH5 的音频时钟 =AUDIO_CLK2 110: 输入 SSI_CH5 的音频时钟 =AUDIO_CLK3 111: 输入 SSI_CH5 的音频时钟 =AUDIO_CLK4 001: 禁止设定 101: 禁止设定</p>
3	—	0	R	<p>保留位</p> <p>读取值为 0, 只能写 0。</p>
2	RPTMD	0	R/W	<p>重复模式位</p> <p>设定正常模式或者重复模式。</p> <p>0: 正常模式 1: 重复模式</p>

位	位名	初始值	R/W	说明
1	TRMD	0	R/W	发送 / 接收模式选择位 设定对应 SSI_CH0 ~ 5 的 SSI_DMAC0/1 的发送模式 / 接收模式。 0: 接收模式 1: 发送模式
0	DMEN	0	R/W	SSI-DMAC 允许位 允许对应 SSI_CH0 ~ 5 的 SSI_DMAC0/1 运行。 0: 停止运行 1: 允许运行

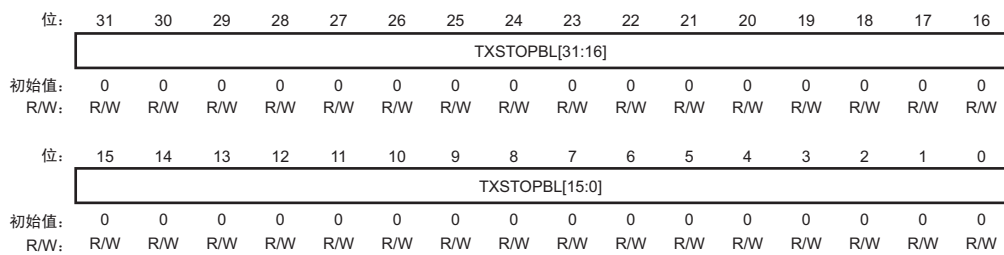
18.3.7 发送暂停块计数器 0 ~ 5 (SSISTPBLCNT0 ~ 5)

SSISTPBLCNT0 ~ 5 是 32 位可读写寄存器，设定在将 SSIDMAOR0 ~ 1 的 TXSTOP0 ~ 5 置 1 后到进入发送暂停状态为止的传送块数。

此寄存器每传送 1 块递减 1，当变为 0 时就进入发送暂停状态。

在硬件复位、软件复位和 SSI_DMAC 的软件复位 (SSIDMCOR0 ~ 3 的 DMRST) 时，此寄存器的值被初始化。

只能在 SSIDMCOR0 ~ 5 的 DMEN=0 时写此寄存器。



位	位名	初始值	R/W	说明
31 ~ 0	TXSTOPBL[31:0]	全 0	R/W	到进入发送暂停状态为止的传送块数 设定在将 SSIDMAOR0 ~ 1 的 TXSTOP0 ~ 5 置 1 后到进入发送暂停状态为止的传送块数。 读取值为设定值 (读不到计数器的值)

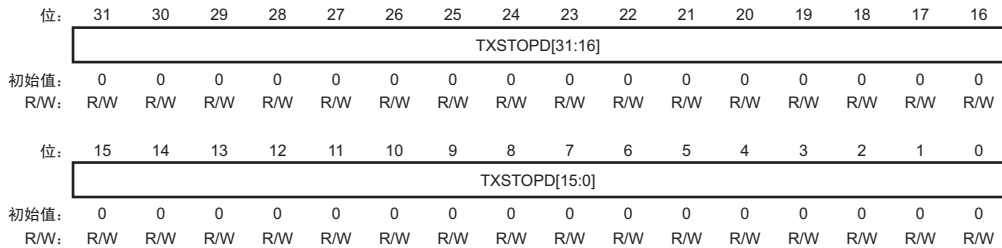
18.3.8 发送暂停中传送数据寄存器 0 ~ 5 (SSISTPDR0 ~ 5)

SSISTPDR0 ~ 5 是 32 位可读写寄存器，设定在发送暂停状态中传送到 SSI_CH0 ~ 5 的数据。

在发送暂停状态中，将此寄存器的设定数据传送到 SSI_CH0 ~ 5，而不将发送 FIFO 缓冲器的数据传送到 SSI_CH0 ~ 5。

在硬件复位、软件复位和 SSI_DMAC 的软件复位 (SSIDMCOR0 ~ 3 的 DMRST) 时，此寄存器的值被初始化。

只能在 SSIDMCOR0 ~ 5 的 DMEN=0 时写此寄存器。



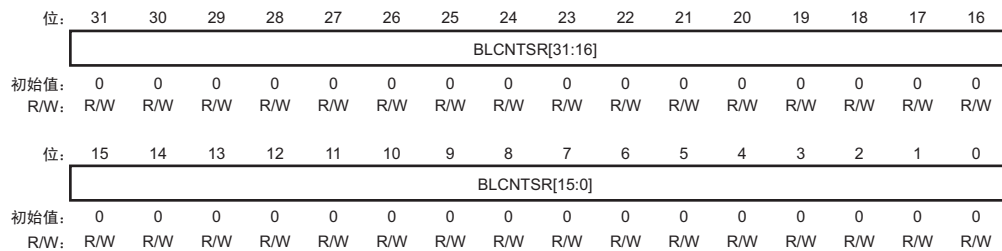
位	位名	初始值	R/W	说明
31 ~ 0	TXSTOPD[31:0]	全 0	R/W	发送暂停状态中的传送数据 设定在发送暂停状态中传送到 SSI_CH0 ~ 5 的数据。

18.3.9 块计数源寄存器 0 ~ 5 (SSIBLCNTR0 ~ 5)

SSIBLCNTR0 ~ 5 是 32 位可读写寄存器，根据传送字节数设定 SSIBLCNTR0 ~ 5 的递增时序。

在硬件复位、软件复位和 SSI_DMAC 的软件复位 (SSIDMCOR0 ~ 3 的 DMRST) 时，此寄存器的值被初始化。

只能在 SSIDMCOR0 ~ 5 的 DMEN=0 时写此寄存器。



位	位名	初始值	R/W	说明
31 ~ 0	BLCNTR[31:0]	全 0	R/W	SSIBLCNTR0 ~ 5 的递增时序 根据传送字节数，设定 SSIBLCNTR0 ~ 5 的递增时序。 必须根据 RDMA 最大突发长度或者 WDMA 最大突发长度设定以下值： 1 个突发：8×n[H'08×n] (字节) 2 个突发：16×n[H'10×n] (字节) 4 个突发：32×n[H'20×n] (字节)

18.3.10 块计数器 0 ~ 5 (SSIBLCNT0 ~ 5)

SSIBLCNT0 ~ 5 是 32 位可读写寄存器，指示以 SSIBLCNTRS0 ~ 5 设定的字节数为 1 个块的传送块数。

在硬件复位、软件复位和 SSI_DMACH 的软件复位 (SSIDMCOR0 ~ 3 的 DMRST) 时，此寄存器的值被初始化。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BLCNT[31:16]															
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
	BLCNT[15:0]															
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
31 ~ 0	BLCNT[31:0]	全 0	R	传送块数 指示以 SSIBLCNTRS0 ~ 5 设定的字节数为 1 个块的传送块数。

18.3.11 模块 xn 次中断计数源寄存器 0 ~ 5 (SSIBLNCNTRS0 ~ 5)

SSIBLNCNTRS0 ~ 5 是 32 位可读写寄存器，根据传送块数设定块 xn 次传送中断的产生时序和 SSIBLNCNT0 ~ 5 的递增时序。

在硬件复位、软件复位和 SSI_DMACH 的软件复位 (SSIDMCOR0 ~ 3 的 DMRST) 时，此寄存器的值被初始化。

只能在 SSIDMCOR0 ~ 5 的 DMEN=0 时写此寄存器。

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

位	位名	初始值	R/W	说明
31 ~ 0	BLNCNTRS[31:0]	全 0	R/W	块 xn 次传送中断产生时序位 根据传送块数，设定块 xn 次传送中断的产生时序和 SSIBLNCNT0 ~ 5 的递增时序。

18.3.12 块 xn 计数器 0 ~ 5 (SSIBLNCNT0 ~ 5)

SSIBLNCNT0 ~ 5 是 32 位可读写寄存器，指示 SSIBLNCNTSR0 ~ 5 设定的每个块数的递增值。

在硬件复位、软件复位和 SSI_DMIC 的软件复位 (SSIDMCOR0 ~ 3 的 DMRST) 时，此寄存器的值被初始化。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BLNCNT[31:16]															
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
	BLNCNT[15:0]															
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
31 ~ 0	BLNCNT[31:0]	全 0	R	传送块 xn 次数 指示 SSIBLNCNTSR0 ~ 5 设定的每个块数的递增值。

18.3.13 DMA 操作寄存器 0 ~ 1 (SSIDMAOR0 ~ 1)

SSIDMAOR0 ~ 1 是 32 位可读写寄存器，设定发送暂停通道的优先顺序和发送 / 接收数据的字节排序。

在硬件复位、软件复位和 SSI_DMIC 的软件复位 (SSIDMCOR0 ~ 3 的 DMRST) 时，此寄存器的值被初始化。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	TX STOP2	TX STOP1	TX STOP0	—	—	PR[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/W	R/W	R/W	R	R	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 7	—	全 0	R	保留位 读取值为 0，只能写 0。
6	TXSTOP2 (TXSTOP5)	0	R/W	SSI_CH2 (CH5) 发送暂停位暂停从 SSI_CH2 (CH5) 的发送 FIFO 缓冲器到 SSI_CH2 (CH5) 的传送，将 SSITXSTPDR2 (5) 设定的数据传送到 SSI_CH2 (CH5)。 0: SSI_CH2 (CH5) 为发送运行 将数据从 SSI_CH2 (CH5) 的发送 FIFO 缓冲器传送到 SSI_CH2 (CH5) 1: SSI_CH2 (CH5) 为发送暂停 将数据从 SSITXSTPDR2 (5) 传送到 SSI_CH2 (CH5)

位	位名	初始值	R/W	说明
5	TXSTOP1 (TXSTOP4)	0	R/W	SSI_CH1 (CH4) 发送暂停位 暂停从 SSI_CH1 (CH4) 的发送 FIFO 缓冲器到 SSI_CH1 (CH4) 的传送, 将 SSITXSTPDR1 (4) 设定的数据传送到 SSI_CH1 (CH4)。 0: SSI_CH1 (CH4) 为发送运行 将数据从 SSI_CH1 (CH4) 的发送 FIFO 缓冲器传送到 SSI_CH1 (CH4) 1: SSI_CH1 (CH4) 为发送暂停 将数据从 SSITXSTPDR1 (4) 传送到 SSI_CH1 (CH4)
4	TXSTOP0 (TXSTOP3)	0	R/W	SSI_CH0 (CH3) 发送暂停位 暂停从 SSI_CH0 (CH3) 的发送 FIFO 缓冲器到 SSI_CH0 (CH3) 的传送, 将 SSITXSTPDR0 (3) 设定的数据传送到 SSI_CH0 (CH3)。 0: SSI_CH0 (CH3) 为发送运行 将数据从 SSI_CH0 (CH3) 的发送 FIFO 缓冲器传送到 SSI_CH0 (CH3) 1: SSI_CH0 (CH3) 为发送暂停 将数据从 SSITXSTPDR0 (3) 传送到 SSI_CH0 (CH3)
3、2	—	全 0	R	保留位 读取值为 0, 只能写 0。
1、0	PR[1:0]	全 0	R/W	优先顺序模式位 1、0 设定多个 SSI (SSI_CH0 ~ 5) 同时发生传送请求时的优先顺序。 00: SSI_CH0>SSI_CH1>SSI_CH2 (SSI_CH3>SSI_CH4>SSI_CH5) 01: SSI_CH0>SSI_CH2>SSI_CH1 (SSI_CH3>SSI_CH5>SSI_CH4) 10: 禁止设定 11: SSI_CH0 ~ 2 (SSI_CH3 ~ 5) 的循环调度

【注】 () 内是 SSIDMAOR1 的情况。

18.3.14 中断状态寄存器 0 ~ 1 (SSIDMINTSR0 ~ 1)

SSIDMINTSR0 ~ 1 是 32 位可读写寄存器，表示 SSI_DMAC0/1 的各种中断源。

如果通过中断屏蔽寄存器 (SSIDMINTMR0 ~ 1) 允许中断时，就产生中断。

通过写“1”，SSIDMINTSR0 ~ 1 的各位被清“0”。忽视写“0”。

在硬件复位、软件复位和 SSI_DMAC 的软件复位 (SSIDMCOR0 ~ 3 的 DMRST) 时，此寄存器的值被初始化。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	BLK END2	BLKN END2	DM END2	TXFIFO FUL2	RXFIFO EMP2
初始值:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	BLK END1	BLKN END1	DM END1	TXFIFO FUL1	RXFIFO EMP1	—	—	—	BLK END0	BLKN END0	DM END0	TXFIFO FUL0	RXFIFO EMP0
初始值:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 21	—	bit24=1 其他 =0	R	保留位 读时 bit24 为 1 而其他位为 0，只能写 0。
20	BLKEND2 (BLKEND5)	0	R/W	块传送结束标志位 2 (5) 表示 SSIBLCNTR2 (5) 所设字节数的数据传送结束。 0: SSIBLCNTR2 (5) 所设字节数的数据传送未结束 1: SSIBLCNTR2 (5) 所设字节数的数据传送结束
19	BLKNEND2 (BLKNEND5)	0	R/W	块 xn 次传送结束标志位 2 (5) 表示 SSIBLNCNTR2 (5) 所设块数的数据传送结束。 0: SSIBLNCNTR2 (5) 所设块数的数据传送未结束 1: SSIBLNCNTR2 (5) 所设块数的数据传送结束
18	DMEND2 (DMEND5)	0	R/W	传送结束标志位 2 (5) 表示 SSIRDMCNTR2 (5) 或者 SSIWDMCNTR2 (5) 所设字节数的数据传送结束。 0: SSIRDMCNTR2 (5) 或者 SSIWDMCNTR2 (5) 所设字节的数据传送未结束 1: SSIRDMCNTR2 (5) 或者 SSIWDMCNTR2 (5) 所设字节的数据传送结束
17	TXFIFOFUL2 (TXFIFOFUL5)	0	R/W	发送 FIFO 缓冲器满标志位 2 (5) 表示 SSI_SH2 (CH5) 的发送 FIFO 缓冲器满。 0: SSI_CH2 (CH5) 的发送 FIFO 缓冲器未 1: SSI_CH2 (CH5) 的发送 FIFO 缓冲器满
16	RXFIFOEMP2 (RXFIFOEMP5)	1	R/W	接收 FIFO 缓冲器空标志位 2 (5) 表示 SSI_CH2 (CH5) 的接收 FIFO 缓冲器空。 0: SSI_CH2 (CH5) 的接收 FIFO 缓冲器未空 1: SSI_CH2 (CH5) 的接收 FIFO 缓冲器空
15 ~ 13	—	全 0	R	保留位 读取值为 0，只能写 0。
12	BLKEND1 (BLKEND4)	0	R/W	块传送结束标志位 1 (4) 表示 SSIBLCNTR1 (4) 所设字节数的数据传送结束。 0: SSIBLCNTR1 (4) 所设字节数的数据传送未结束 1: SSIBLCNTR1 (4) 所设字节数的数据传送结束

位	位名	初始值	R/W	说明
11	BLKNEND1 (BLKNEND4)	0	R/W	块 xn 次传送结束标志位 1 (4) 表示 SSIBLNCNTR1 (4) 所设块数的数据传送结束。 0: SSIBLNCNTR1 (4) 所设块数的数据传送未结束 1: SSIBLNCNTR1 (4) 所设块数的数据传送结束
10	DMEND1 (DMEND4)	0	R/W	传送结束标志位 1 (4) 表示 SSIRDMCNTR1 (4) 或者 SSIWDMCNTR1 (4) 所设字节数的数据传送结束。 0: SSIRDMCNTR1 (4) 或者 SSIWDMCNTR1 (4) 所设字节数的数据传送未结束 1: SSIRDMCNTR1 (4) 或者 SSIWDMCNTR1 (4) 所设字节数的数据传送结束
9	TXFIFOFUL1 (TXFIFOFUL4)	0	R/W	发送 FIFO 缓冲器满标志位 1 (4) 表示 SSI_SH1 (CH4) 的发送 FIFO 缓冲器满。 0: SSI_CH1 (CH4) 的发送 FIFO 缓冲器未 1: SSI_CH1 (CH4) 的发送 FIFO 缓冲器满
8	RXFIFOEMP1 (RXFIFOEMP4)	1	R/W	接收 FIFO 缓冲器空标志位 1 (4) 表示 SSI_CH1 (CH4) 的接收 FIFO 缓冲器空。 0: SSI_CH1 (CH4) 的接收 FIFO 缓冲器未空 1: SSI_CH1 (CH4) 的接收 FIFO 缓冲器空
7 ~ 5	—	全 0	R	保留位 读取值为 0, 只能写 0。
4	BLKEND0 (BLKEND3)	0	R/W	块传送结束标志位 0 (3) 表示 SSIBLNCNTR0 (3) 所设字节数的数据传送结束。 0: SSIBLNCNTR0 (3) 所设字节数的数据传送未结束 1: SSIBLNCNTR0 (3) 所设字节数的数据传送结束
3	BLKNEND0 (BLKNEND3)	0	R/W	块 xn 次传送结束标志位 0 (3) 表示 SSIBLNCNTR0 (3) 所设块数的数据传送结束。 0: SSIBLNCNTR0 (3) 所设块数的数据传送未结束 1: SSIBLNCNTR0 (3) 所设块数的数据传送结束
2	DMEND0 (DMEND3)	0	R/W	传送结束标志位 0 (3) 表示 SSIRDMCNTR0 (3) 或者 SSIWDMCNTR0 (3) 所设字节数的数据传送结束。 0: SSIRDMCNTR0 (3) 或者 SSIWDMCNTR0 (3) 所设字节数的数据传送未结束。 1: SSIRDMCNTR0 (3) 或者 SSIWDMCNTR0 (3) 所设字节数的数据传送结束
1	TXFIFOFUL0 (TXFIFOFUL3)	0	R/W	发送 FIFO 缓冲器满标志位 0 (3) 表示 SSI_CH0 (CH3) 的发送 FIFO 缓冲器满。 0: SSI_CH0 (CH3) 的发送 FIFO 缓冲器未 1: SSI_CH0 (CH3) 的发送 FIFO 缓冲器满
0	RXFIFOEMP0 (RXFIFOEMP3)	1	R/W	接收 FIFO 缓冲器空标志位 0 (3) 表示 SSI_CH0 (CH3) 的接收 FIFO 缓冲器空。 0: SSI_CH0 (CH3) 的接收 FIFO 缓冲器未空 1: SSI_CH0 (CH3) 的接收 FIFO 缓冲器空

【注】 () 内是 SSIDMINTSR1 的情况。

18.3.15 中断屏蔽寄存器 0 ~ 1 (SSIDMINTMR0 ~ 1)

SSIDMINTMR0 ~ 1 是 32 位可读写寄存器，屏蔽 SSI_DMACH0/1 的各种中断源。

在硬件复位、软件复位和 SSI_DMACH 的软件复位 (SSIDMCOR0 ~ 3 的 DMRST) 时，此寄存器的值被初始化。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	BLKEND M2	BLKN ENDM2	DMEND M2	TXFIFO FULM2	RXFIFO EMPM2
初始值:	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	BLKEND M1	BLKN ENDM1	DM ENDM1	TXFIFO FULM1	RXFIFO EMPM1	—	—	—	BLKEND M0	BLKN ENDM0	DMEND M0	TXFIFO FULM0	RXFIFO EMPM0
初始值:	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 21	—	bit31 ~ 29=0 bit28 ~ 24=1	R	保留位 读时 bit31 ~ 29 为 0 而 bit28 ~ 24 为 1，只能写同样的值。
20	BLKENDM2 (BLKENDM5)	1	R/W	BLKEND2 (5) 中断源屏蔽位 设定 BLKEND2 (5) 中断源的屏蔽。 0: 不屏蔽 BLKEND2 (5) 的中断源 1: 屏蔽 BLKEND2 (5) 的中断源
19	BLKNENDM2 (BLKNENDM5)	1	R/W	BLKNEND2 (5) 中断源屏蔽位 设定 BLKNEND2 (5) 中断源的屏蔽。 0: 不屏蔽 BLKNEND2 (5) 的中断源 1: 屏蔽 BLKNEND2 (5) 的中断源
18	DMENDM2 (DMENDM5)	1	R/W	DMEND2 (5) 中断源屏蔽位 设定 DMEND2 (5) 中断源的屏蔽。 0: 不屏蔽 DMEND2 (5) 的中断源 1: 屏蔽 DMEND2 (5) 的中断源
17	TXFIFOFULM2 (TXFIFOFULM5)	1	R/W	TXFIFOFUL2 (5) 中断源屏蔽位 设定 TXFIFOFUL2 (5) 中断源的屏蔽。 0: 不屏蔽 TXFIFOFUL2 (5) 的中断源 1: 屏蔽 TXFIFOFUL2 (5) 的中断源
16	RXFIFOEMPM2 (RXFIFOEMPM5)	1	R/W	RXFIFOEMP2 (5) 中断源屏蔽位 设定 RXFIFOEMP2 (5) 中断源的屏蔽。 0: 不屏蔽 RXFIFOEMP2 (5) 的中断源 1: 屏蔽 RXFIFOEMP2 (5) 的中断源
15 ~ 13	—	全 0	R	保留位 读取值为 0，只能写 0。
12	BLKENDM1 (BLKENDM4)	1	R/W	BLKEND1 (4) 中断源屏蔽位 设定 BLKEND1 (4) 中断源的屏蔽。 0: 不屏蔽 BLKEND1 (4) 的中断源 1: 屏蔽 BLKEND1 (4) 的中断源

位	位名	初始值	R/W	说明
11	BLKNENDM1 (BLKNENDM4)	1	R/W	BLKNEND1 (4) 中断源屏蔽位 设定 BLKNEND1 (4) 中断源的屏蔽。 0: 不屏蔽 BLKNEND1 (4) 的中断源 1: 屏蔽 BLKNEND1 (4) 的中断源
10	DMENDM1 (DMENDM4)	1	R/W	DMEND1 (4) 中断源屏蔽位 设定 DMEND1 (4) 中断源的屏蔽。 0: 不屏蔽 DMEND1 (4) 的中断源 1: 屏蔽 DMEND1 (4) 的中断源
9	TXFIFOFULM1 (TXFIFOFULM4)	1	R/W	TXFIFOFUL1 (4) 中断源屏蔽位 设定 TXFIFOFUL1 (4) 中断源的屏蔽。 0: 不屏蔽 TXFIFOFUL1 (4) 的中断源 1: 屏蔽 TXFIFOFUL1 (4) 的中断源
8	RXFIFOEMP1 (RXFIFOEMP4)	1	R/W	RXFIFOEMP1 (4) 中断源屏蔽位 设定 RXFIFOEMP1 (4) 中断源的屏蔽。 0: 不屏蔽 RXFIFOEMP1 (4) 的中断源 1: 屏蔽 RXFIFOEMP1 (4) 的中断源
7 ~ 5	—	全 0	R	保留位 读取值为 0, 只能写 0。
4	BLKENDM0 (BLKENDM3)	1	R/W	BLKEND0 (3) 中断源屏蔽位 设定 BLKEND0 (3) 中断源的屏蔽。 0: 不屏蔽 BLKEND0 (3) 的中断源 1: 屏蔽 BLKEND0 (3) 的中断源
3	BLKNENDM0 (BLKNENDM3)	1	R/W	BLKNEND0 (3) 中断源屏蔽位 设定 BLKNEND0 (3) 中断源的屏蔽。 0: 不屏蔽 BLKNEND0 (3) 的中断源 1: 屏蔽 BLKNEND0 (3) 的中断源
2	DMENDM0 (DMENDM3)	1	R/W	DMEND0 (3) 中断源屏蔽位 设定 DMEND0 (3) 中断源的屏蔽。 0: 不屏蔽 DMEND0 (3) 的中断源 1: 屏蔽 DMEND0 (3) 的中断源
1	TXFIFOFULM0 (TXFIFOFULM3)	1	R/W	TXFIFOFUL0 (3) 中断源屏蔽位 设定 TXFIFOFUL0 (3) 中断源的屏蔽。 0: 不屏蔽 TXFIFOFUL0 (3) 的中断源 1: 屏蔽 TXFIFOFUL0 (3) 的中断源
0	RXFIFOEMP0 (RXFIFOEMP3)	1	R/W	RXFIFOEMP0 (3) 中断源屏蔽位 设定 RXFIFOEMP0 (3) 中断源的屏蔽。 0: 不屏蔽 RXFIFOEMP0 (3) 的中断源 1: 屏蔽 RXFIFOEMP0 (3) 的中断源

【注】 () 内是 SSIDMINTMR1 的情况。

18.3.16 控制寄存器 0 ~ 5 (SSICR0 ~ 5)

SSICR0 ~ 5 控制中断、选择各极性以及设定运行模式。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DMEN	UIEN	OIEN	IEN	DIEN	CHNL[1:0]	DWL[2:0]			SWL[2:0]			
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL	—	CKDV[2:0]			MUEN	—	TRMD	EN
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 29	—	0	R	保留位 读取值为 0，只能写 0。
28	DMEN	0	R/W	DMA 允许位 允许或者禁止 DMA 请求。 0: 禁止 DMA 请求 1: 允许 DMA 请求
27	UIEN	0	R/W	下溢中断允许位 0: 禁止下溢中断 1: 允许下溢中断
26	OIEN	0	R/W	上溢中断允许位 0: 禁止上溢中断 1: 允许上溢中断
25	IEN	0	R/W	空闲模式中断允许位 0: 禁止空闲模式中断 1: 允许空闲模式中断
24	DIEN	0	R/W	数据中断允许位 0: 禁止数据中断 1: 允许数据中断
23、22	CHNL[1:0]	00	R/W	通道位 表示各系统字的通道数。 00: 各系统字由 1 个通道构成。 01: 各系统字由 2 个通道构成。 10: 各系统字由 3 个通道构成。 11: 各系统字由 4 个通道构成。
21 ~ 19	DWL[2:0]	000	R/W	数据字长度位 表示数据字的位数。 000: 8 位 001: 16 位 010: 18 位 011: 20 位 100: 22 位 101: 24 位 110: 32 位 111: 禁止设定

位	位名	初始值	R/W	说明															
18 ~ 16	SWL[2:0]	000	R/W	系统字长度位 表示系统字的位数。 000: 8 位 001: 16 位 010: 24 位 001: 32 位 100: 48 位 101: 64 位 110: 128 位 111: 256 位															
15	SCKD	0	R/W	串行位时钟方向位 0: 串行位时钟输入, 从属模式 1: 串行位时钟输出, 主模式 【注】 只能设定 (SCKD,SWSD) = (0,0) 和 (1,1)。禁止其他的设定。															
14	SWSD	0	R/W	串行字选择信号 (WS) 方向位 0: 串行字选择输入, 从属模式 1: 串行字选择输出, 主模式 【注】 只能设定 (SCKD,SWSD) = (0,0) 和 (1,1)。禁止其他的设定。															
13	SCKP	0	R/W	串行位时钟极性位 0: SSIWS[5:0] 和 SSIDATA[5:0] 在 SSISCK[5:0] 的下降沿发生变化 (在 SSISCK[5:0] 的上升沿采样) 1: SSIWS[5:0] 和 SSIDATA[5:0] 在 SSISCK[5:0] 的上升沿发生变化 (在 SSISCK[5:0] 的下降沿采样) <table border="1" style="margin-left: 20px;"> <thead> <tr> <th></th> <th>SCKP=0</th> <th>SCKP=1</th> </tr> </thead> <tbody> <tr> <td>接收时 (TRMD=0) SSIDATA[5:0] 输入采样时序</td> <td>SSISCK[5:0] 上升沿</td> <td>SSISCK[5:0] 下降沿</td> </tr> <tr> <td>发送时 (TRMD=1) SSIDATA[5:0] 输出变化时序</td> <td>SSISCK[5:0] 下降沿</td> <td>SSISCK[5:0] 上升沿</td> </tr> <tr> <td>从属模式时 (SWSD=0) SSIWS[5:0] 输入采样时序</td> <td>SSISCK[5:0] 上升沿</td> <td>SSISCK[5:0] 下降沿</td> </tr> <tr> <td>主模式时 (SWSD=1) SSIWS[5:0] 输出变化时序</td> <td>SSISCK[5:0] 下降沿</td> <td>SSISCK[5:0] 上升沿</td> </tr> </tbody> </table>		SCKP=0	SCKP=1	接收时 (TRMD=0) SSIDATA[5:0] 输入采样时序	SSISCK[5:0] 上升沿	SSISCK[5:0] 下降沿	发送时 (TRMD=1) SSIDATA[5:0] 输出变化时序	SSISCK[5:0] 下降沿	SSISCK[5:0] 上升沿	从属模式时 (SWSD=0) SSIWS[5:0] 输入采样时序	SSISCK[5:0] 上升沿	SSISCK[5:0] 下降沿	主模式时 (SWSD=1) SSIWS[5:0] 输出变化时序	SSISCK[5:0] 下降沿	SSISCK[5:0] 上升沿
	SCKP=0	SCKP=1																	
接收时 (TRMD=0) SSIDATA[5:0] 输入采样时序	SSISCK[5:0] 上升沿	SSISCK[5:0] 下降沿																	
发送时 (TRMD=1) SSIDATA[5:0] 输出变化时序	SSISCK[5:0] 下降沿	SSISCK[5:0] 上升沿																	
从属模式时 (SWSD=0) SSIWS[5:0] 输入采样时序	SSISCK[5:0] 上升沿	SSISCK[5:0] 下降沿																	
主模式时 (SWSD=1) SSIWS[5:0] 输出变化时序	SSISCK[5:0] 下降沿	SSISCK[5:0] 上升沿																	
12	SWSP	0	R/W	串行字选择信号 (WS) 极性位 0: SSIWS[5:0] 在第 1 个通道为低电平, 在第 2 个通道为高电平 1: SSIWS[5:0] 在第 1 个通道为高电平, 在第 2 个通道为低电平															
11	SPDP	0	R/W	串行填充极性位 0: 填充位为低电平 1: 填充位为高电平 当 MUEN=1 时, 填充位变为低电平 (优先 MUTE 功能)。															
10	SDTA	0	R/W	串行数据调整位 0: 按串行数据、填充位的顺序进行发送和接收 1: 按填充位、串行数据的顺序进行发送和接收															

位	位名	初始值	R/W	说明
9	PDТА	0	R/W	<p>并行数据调整位</p> <p>当数据字长度为 32、16 或者 8 位时，此位无意义。</p> <p>此位适用于接收模式的 SSIRDRO ~ 5 和发送模式的 SSITDRO ~ 5。</p> <p>0: 并行数据 (SSITDRO ~ 5、SSIRDRO ~ 5) 向左排</p> <p>1: 并行数据 (SSITDRO ~ 5、SSIRDRO ~ 5) 向右排</p> <ul style="list-style-type: none"> • DWL[2:0]=000 (数据字长度为 8 位时)，忽视 PDТА 的设定 串行音频总线使用 SSIRDRO ~ 5 或者 SSITDRO ~ 5 的全部数据位。 每当进行 32 位存取时，发送 / 接收 4 个数据字。 bit7 ~ 0 保存第 1 个数据字，bit15 ~ 8 保存第 2 个数据字，bit23 ~ 16 保存第 3 个数据字，bit31 ~ 24 保存第 4 个数据字。 • DWL[2:0]=001 (数据字长度为 16 位时)，忽视 PDТА 的设定 串行音频总线使用 SSIRDRO ~ 5 或者 SSITDRO ~ 5 的全部数据位。 每当进行 32 位存取时，发送 / 接收 2 个数据字。 bit15 ~ 0 保存第 1 个数据字，bit31 ~ 16 保存第 2 个数据字。 • DWL[2:0]=010、011、100、101 (数据字长度为 18、20、22 或者 24 位时)，PDТА=0 (向左排) SSIRDRO ~ 5 或者 SSITDRO ~ 5 的数据位中被使用的位如下： bit31 ~ bit(32-DWL[2:0] 设定的数据字长度的位数) 即：当 DWL[2:0]=011 时，数据字长度为 20 位，使用 SSIRDRO ~ 5 或者 SSITDRO ~ 5 的 bit31 ~ 12。 其他的所有位被忽视或者为保留位。 • DWL[2:0]=010、011、100、101 (数据字长度为 18、20、22 或者 24 位时)，PDТА=1 (向右排) SSIRDRO ~ 5 或者 SSITDRO ~ 5 的数据位中被使用的位如下： bit(DWL[2:0] 设定的数据字长度的位数 - 1) ~ 位 0 即：当 DWL[2:0]=011 时，数据字长度为 20 位，使用 SSIRDRO ~ 5 或者 SSITDRO ~ 5 的 bit19 ~ 0。 其他的所有位被忽视或者为保留位。 • DWL[2:0]=110 (数据字长度为 32 位时)，忽视 PDТА 的设定 串行音频总线使用 SSIRDRO ~ 5 或者 SSITDRO ~ 5 的全部数据位。
8	DEL	0	R/W	<p>串行数据延迟位</p> <p>0: 在 SSIWS[5:0] 和 SSIDATA[5:0] 之间延迟 1 个时钟周期</p> <p>1: 在 SSIWS[5:0] 和 SSIDATA[5:0] 之间无延迟</p>
7	—	0	R	<p>保留位</p> <p>读取值为 0，只能写 0。</p>

位	位名	初始值	R/W	说明
6 ~ 4	CKDV[2:0]	000	R/W	串行过采样时钟分频比位 设定过采样时钟 AUDIO_CLK[5:0] 和串行位时钟的分频比。 当 SCKD=0 时, 忽视此位。 串行位时钟用于移位寄存器, 由 SSISCK[5:0] 引脚提供。 000: 串行位时钟频率 = 过采样时钟频率 /1 001: 串行位时钟频率 = 过采样时钟频率 /2 010: 串行位时钟频率 = 过采样时钟频率 /4 011: 串行位时钟频率 = 过采样时钟频率 /8 100: 串行位时钟频率 = 过采样时钟频率 /16 101: 串行位时钟频率 = 过采样时钟频率 /6 110: 串行位时钟频率 = 超采样时钟频率 /12 111: 禁止设定
3	MUEN	0	R/W	静音允许位 0: SSI_CH0 ~ 5 不处于静音状态 1: SSI_CH0 ~ 5 处于静音状态
2	—	0	R	保留位 读取值为 0, 只能写 0。
1	TRMD	0	R/W	发送 / 接收模式选择位 0: SSI_CH0 ~ 5 为接收模式 1: SSI_CH0 ~ 5 为发送模式
0	EN	0	R/W	运行允许位 0: 禁止 SSI_CH0 ~ 5 运行 1: 允许 SSI_CH0 ~ 5 运行

18.3.17 状态寄存器 0 ~ 5 (SSISR0 ~ 5)

SSISR0 ~ 5 由表示运行状态的状态标志、表示当前通道号和字号的位构成。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	DMRQ	UIRQ	OIRQ	IIRQ	DIRQ	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0
R/W:	R	R	R	R	R/W*	R/W*	R	R	R	R	R	R	R	R	R	R

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CHNO[1:0]	SWNO	IDST	
初始值:	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1
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 ~ 29	—	0	R	保留位 读取值为 0, 只能写 0。

位	位名	初始值	R/W	说明
28	DMRQ	0	R	<p>DMA 请求状态标志位</p> <p>CPU 能通过本状态标志得知 SSI_CH0 ~ 5 的 DMA 请求状态。</p> <ul style="list-style-type: none"> • TRMD=0 (接收模式) 时 当 DMRQ=1 时, SSIRD0 ~ 5 中有未读数据。 读取 SSIRD0 ~ 5 后, 在接收到下一个未读数据前 DMRQ=0。 • TRMD=1 (发送模式) 时 当 DMRQ=1 时, 为了能继续在串行音频总线上进行发送, SSITDR0 ~ 5 请求写数据。 给 SSITDR0 ~ 5 写数据后, 在发生下一个发送数据的请求前 DMRQ=0。
27	UIRQ	0	R/W*	<p>下溢错误中断状态标志位</p> <p>表示以低于请求速率的速率提供数据。</p> <p>与 UIEN 位的设定无关, 此位被置 1。清 0 时必须写 0。</p> <p>在 UIRQ=1 并且 UIEN=1 时, 产生中断。</p> <ul style="list-style-type: none"> • TRMD=0 (接收模式) 时 当 UIRQ=1 时, 表示在 DMRQ 或者 DIRQ 位指示存在新的未读数据之前已读取 SSIRD0 ~ 5。 此时, 相同的接收数据可能会被主机保存 2 次, 导致多通道数据的破坏。 • TRMD=1 (发送模式) 时 当 UIRQ=1 时, 表示在发送之前没有写 SSITDR0 ~ 5 的发送数据。 此时, 相同的数据可能被多发送 1 次, 导致多通道数据的破坏。 结果, 会输出错误的 SSI 数据, 该错误比接收模式的下溢更为严重。 <p>【注】 当发生下溢错误时, 在写下一个数据前发送数据缓冲器中的数据。</p>
26	OIRQ	0	R/W*	<p>上溢错误中断状态标志位</p> <p>表示以高于请求速率的速率提供数据。</p> <p>与 OIEN 位的设定无关, 此位被置 1。清 0 时必须写 0。</p> <p>在 OIRQ=1 并且 OIEN=1 时, 产生中断。</p> <ul style="list-style-type: none"> • TRMD=0 (接收模式) 时 当 OIRQ=1 时, 表示在给 SSIRD0 ~ 5 写新的未读数据前没有读取以前的未读数据。 此时, 有可能丢失数据, 导致多通道数据的破坏。 • TRMD=1 (发送模式) 时 当 OIRQ=1 时, 表示在将 SSITDR0 ~ 5 中的数据传送给移位寄存器之前已给 SSITDR0 ~ 5 写数据。 此时, 有可能丢失数据, 导致多通道数据的破坏。 <p>【注】 当发生上溢错误时, 从 SSI 接口送来的下一个数据盖写数据缓冲器中的数据。</p>

位	位名	初始值	R/W	说明
25	IIRQ	1	R	<p>空闲模式中断状态标志位</p> <p>表示 SSI_CH0 ~ 5 是否处于空闲状态。</p> <p>为了能进行查询, 与 I IEN 位的设定无关, 此位被置 1。</p> <p>能通过将 I IEN 位清 0 屏蔽中断, 但是即使将此位置 0 也不能清除中断。</p> <p>在 IIRQ=1 并且 I IEN=1 时, 产生中断。</p> <p>0: SSI_CH0 ~ 5 不处于空闲状态</p> <p>1: SSI_CH0 ~ 5 处于空闲状态</p>
24	DIRQ	0	R	<p>数据中断状态标志位</p> <p>表示 SSI_CH0 ~ 5 是否需要读写数据。</p> <p>为了能进行查询, 与 D IEN 位的设定无关, 此位被置 1。</p> <p>能通过将 D IEN 位清 0 可屏蔽中断, 但是即使将此位置 0 也不能清除中断。</p> <p>在 DIRQ=1 并且 D IEN=1 时, 产生中断。</p> <ul style="list-style-type: none"> • TRMD=0 (接收模式) 时 <ul style="list-style-type: none"> 0: SSIRDRO ~ 5 中无未读数据 1: SSIRDRO ~ 5 中有未读数据 • TRMD=1 (发送模式) 时 <ul style="list-style-type: none"> 0: 发送缓冲器满 1: 发送缓冲器空, 请求给 SSITDR0 ~ 5 写数据。
23 ~ 4	—	H'10A00	R	<p>保留位</p> <p>读取值为 H'10A00, 只能写 0。</p>
3、2	CHNO[1:0]	00	R	<p>通道号位</p> <p>表示当前的通道。</p> <ul style="list-style-type: none"> • TRMD=0 (接收模式) 时 <ul style="list-style-type: none"> 表示 SSIRDRO ~ 5 中的当前数据是哪个通道的数据。 当因移位寄存器的传送而更新 SSIRDRO ~ 5 中的数据时, 此值发生变化。 • TRMD=1 (发送模式) 时 <ul style="list-style-type: none"> 表示应给 SSITDR0 ~ 5 写哪个通道的数据。 当数据复制到移位寄存器时, 与是否写 SSITDR0 ~ 5 无关, 此值发生变化。
1	SWNO	1	R	<p>串行字号位</p> <p>表示当前的字号。</p> <ul style="list-style-type: none"> • TRMD=0 (接收模式) 时 <ul style="list-style-type: none"> 表示 SSIRDRO ~ 5 中的当前数据是哪个系统字。 与是否读取 SSIRDRO ~ 5 无关, 当因移位寄存器的传送而更新 SSIRDRO ~ 5 中的数据时, 此值发生变化。 • TRMD=1 (发送模式) 时 <ul style="list-style-type: none"> 表示应给 SSITDR0 ~ 5 写哪个系统字。 当数据复制到移位寄存器时, 与是否写入 SSITDR0 ~ 5 无关, 此值发生变化。

位	位名	初始值	R/W	说明
0	IDST	1	R	空闲模式状态标志位 表示串行总线已处于停止状态。 在 EN=1 并且串行总线处于运行状态时，此位被清除。 当以下条件成立时，此位自动置 1。 <ul style="list-style-type: none"> SSI_CH0 ~ 5 为主控发送器 (SWSD=1 并且 TRMD=1) 时清除 EN 位，当 SSITDR0 ~ 5 的数据从串行数据输入 / 输出引脚 (SSIDATA0 ~ 5) 输出结束时，此位被置 1。 SSI_CH0 ~ 5 为主控接收器 (SWSD=1 并且 TRMD=0) 时清除 EN 位，当结束当前的系统字时，此位被置 1。 SSI_CH0 ~ 5 为从属发送器 / 接收器 (SWSD=0) 时清除 EN 位，当结束当前的系统字时，此位被置 1。 【注】 在结束当前的系统字之前，如果外部设备停止串行总线时钟，此位就不被置位。

【注】 * 可读写。如果写 0，本位就被初始化，忽视写 1。

18.3.18 发送数据寄存器 0 ~ 5 (SSITDR0 ~ 5)

SSITDR0 ~ 5 保存发送的数据。

当出现发送请求时，将此寄存器的数据传送到移位寄存器。在数据字长度不满 32 位时，根据 SSICR0 ~ 5 的 PDTA 控制位的设定进行调整。

能通过读此寄存器获得缓冲器内的数据。

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

18.3.19 接收数据寄存器 0 ~ 5 (SSIRDRO ~ 5)

SSIRDRO ~ 5 保存接收的数据。

每当接收到数据字时，将数据从移位寄存器传送到此寄存器。在数据字长度不满 32 位时，根据 SSICR0 ~ 5 的 PDTA 控制位的设定进行调整。

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

18.4 运行说明

18.4.1 SSI_CLKSEL 的运行说明

根据寄存器的设定，SSI_CLKSEL 将 SSI 接口的各通道的音频时钟和串行位时钟 / 串行字选择信号连接到 SSI_CH0 ~ 5。

- 音频时钟的选择
根据 SSIDMCOR0 ~ 5 的 SCKS[2:0] 位的设定，SSI_CLKSEL 选择 SSI_CH0 ~ 5 使用的音频时钟 (AUDIO_CLK[5:0])。
初始状态为选择相同通道的音频时钟，如：AUDIO_CLK0=SSI_CH0。
- 串行位时钟/串行字选择信号的选择
根据 SSIDMCOR0 ~ 5 的 SCKSIP[2:0] 或者 SCKOP[2:0] 位的设定，SSI_CLKSEL 选择 SSI_CH0 ~ 5 使用的串行位时钟/串行字选择信号 (SSISCK[5:0]/SSIWS[5:0])。
初始状态为选择相同通道的音频时钟，如：SSISCK0=SSI_CH0。

18.4.2 SSI_DMAC0/1 的运行说明

SSI_DMAC0/1 进行 6 个通道的串行音频接口 (SSI_CH0 ~ 5) 和外部存储器 / 内部存储器之间的数据传送。SSI_DMAC0/1 分别有发送和接收的 FIFO 缓冲器 (32 位 × 16 段)，能进行高效率的高速连续通信。

SSI_DMAC0/1 能以任意的块为单位对数据传送量进行计数，并且能以该块为单位进行中断产生和发送暂停等数据传送的管理。

另外，能暂停数据传送，并且在暂停中通过继续给 SSI_CH0 ~ 5 传送任意数据 (例如无声数据)，暂停声音的发出。

- 通道数：
对应 SSI_CH0 ~ 5 发送/接收的 6 个通道
- 传送数据长度：
8 字节 / 16 字节 / 32 字节
- 最大传送字节数：
4,294,967,296 字节
- 总线模式：
周期挪用
- 通道间的优先顺序：
SSI_CH0 ~ 2 和 SSI_CH3 ~ 5 能选择固定优先和循环调度模式。
- 发送 FIFO 缓冲器 / 接收 FIFO 缓冲器：
内置对应 SSI_CH0 ~ 5 发送/接收的 FIFO 缓冲器 (32 位 × 16 段)
- 中断请求：
块传送结束中断
块 xn 次传送结束中断
传送结束中断
发送 FIFO 缓冲器满中断
接收 FIFO 缓冲器空中断
- 软件复位：
能分别对 SSI_CH0 ~ 5 进行软件复位。
能分别对 SSI_CH0 ~ 5 的各发送 FIFO 缓冲器 / 接收 FIFO 缓冲器进行复位。
- 发送暂停：
能选择立即停止或者在任意块单位的数据传送后停止。
在暂停中，自动将任意数据 (例如无声数据) 传送到 SSI_CH0 ~ 5。

18.4.3 SSI_CH0 ~ 5 的运行说明

(1) 总线格式

SSI_CH0 ~ 5 能用作发送器或者接收器，无论哪个模式都能使用多种串行总线格式。
总线格式能选择如表 18.8 所示的 8 个主要模式。

表 18.8 SSI_CH0 ~ 5 总线格式

总线格式	TRMD	SCKD	SWSD	EN	MUEN	DIEN	IEN	OEN	UIEN	DEL	PDTA	SDTA	SPDP	SWSP	SCKP	SWL[2:0]	DWL[2:0]	CHNL[1:0]	
非压缩从属接收器	0	0	0	控制位						结构位									
非压缩从属发送器	1	0	0																
非压缩主控接收器	0	1	1																
非压缩主控发送器	1	1	1																

(2) 非压缩模式

非压缩模式支持被分割到各通道的串行音频流。
不仅支持 Philips、Sony 和松下模式，还对应多种改良版模式。

(a) 从属接收器

在此模式中，能从其他设备接收串行数据。
由外部设备提供用于串行数据流的时钟和串行字选择信号。
在这些信号和设定在 SSI_CH0 ~ 5 的格式不匹配时，不保证运行。

(b) 从属发送器

在此模式中，能给其他设备发送串行数据。
由外部设备提供用于串行数据流的时钟和串行字选择信号。
在这些信号和设定在 SSI_CH0 ~ 5 的格式不匹配时，不保证运行。

(c) 主控接收器

在此模式中，能从其他设备接收串行数据。
从 AUDIO_CLK0 ~ 5 的输入时钟，在内部生成时钟和串行字选择信号。
这些信号的格式取决于 SSI_CH0 ~ 5 的设定。
在从其他设备发送的数据和设定的格式不匹配时，不保证运行。

(d) 主控发送器

在此模式中，能给其他设备发送串行数据。
从 AUDIO_CLK0 ~ 5 的输入时钟，在内部生成时钟和串行字选择信号。
这些信号的格式取决于 SSI_CH0 ~ 5 的设定。

(e) 运行设定的关键字长度

在非压缩模式中，与 SSICR0 ~ 5 字长相关的所有位都有效。
SSI_CH0 ~ 5 可支持多种结构，在此对 Philips、Sony 和松下的格式进行说明。

- Philips 格式

有填充和无填充的 Philips 格式分别如图 18.2 和图 18.3 所示。在数据字的长度小于系统字的长度时，发生填充。

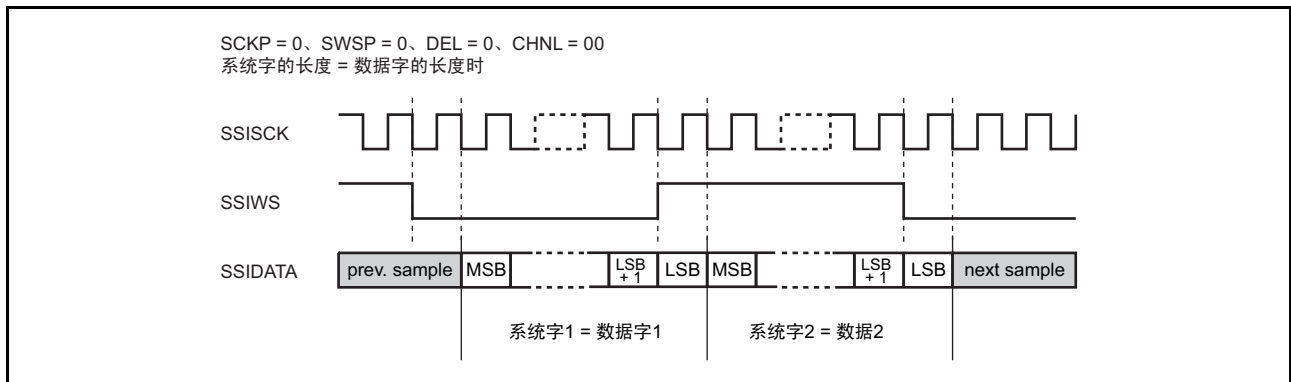


图 18.2 Philips 格式 (无填充)

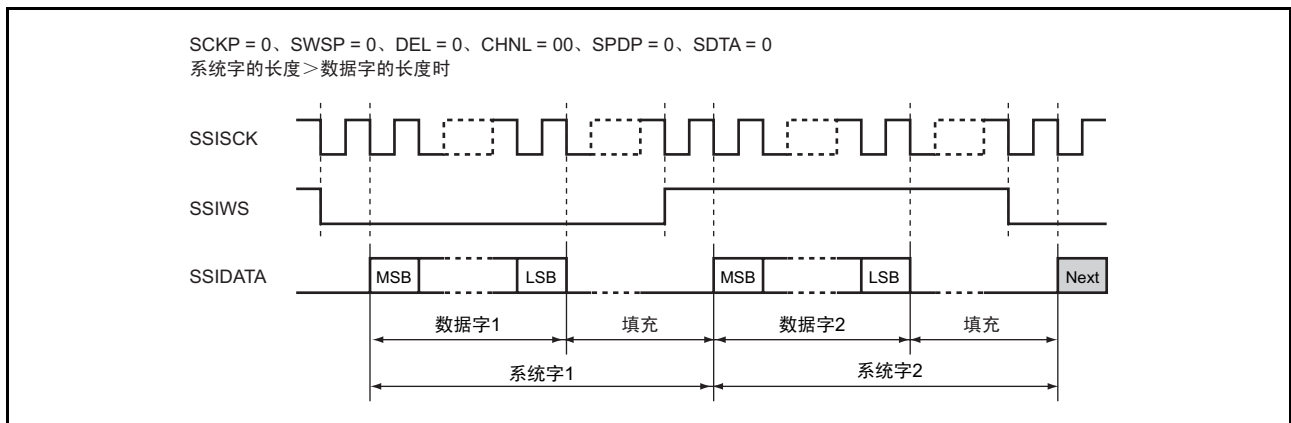


图 18.3 Philips 格式 (有填充)

- Sony 格式

Sony 格式如图 18.4 所示。

这是有填充的例子，在系统字的长度和数据字的长度相同时，可能需要无填充的格式。

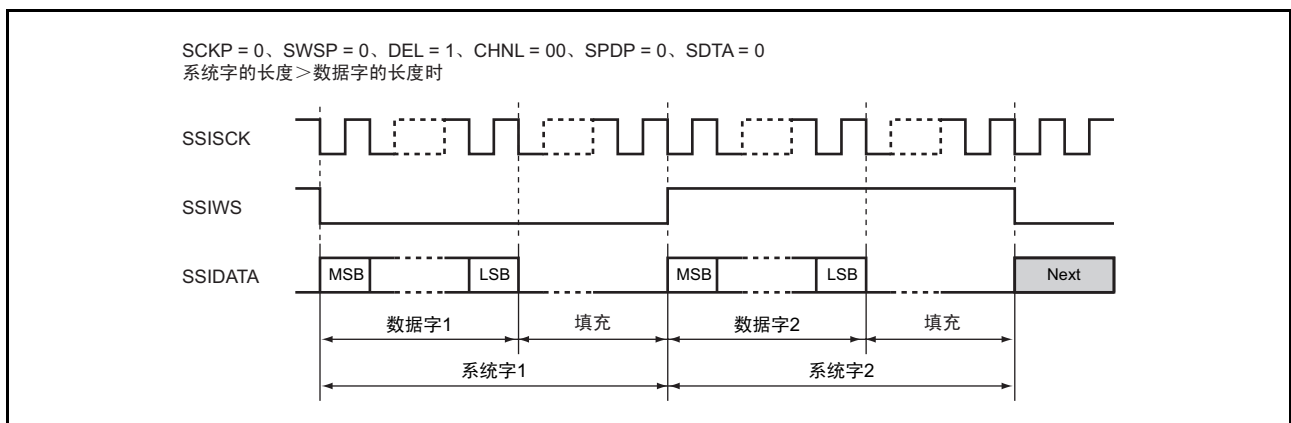


图 18.4 Sony 格式 (按串行数据、填充位的顺序进行发送和接收)

- 松下格式

松下格式如图 18.5 所示。

这是有填充的例子，在系统字的长度和数据字的长度相同时，可能需要无填充的格式。

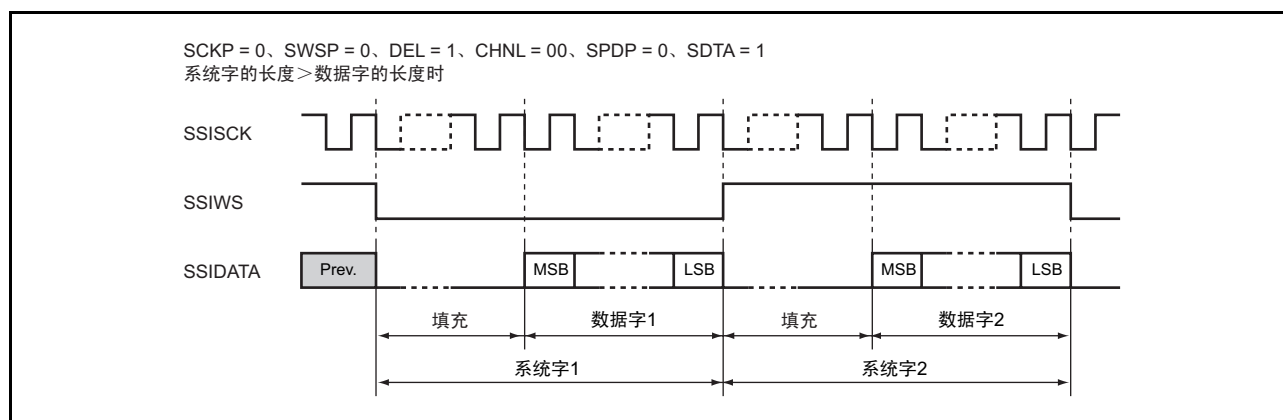


图 18.5 松下格式（按填充位、串行数据的顺序进行发送和接收）

(f) 多通道格式

有些设备类型扩展了 Philips 规格的定义，允许在 2 个系统字中进行 2 个通道以上的传送。

SSI_CH0 ~ 5 使用 CHNL、SWL 和 DWL 位，进行 2、3 和 4 个通道的传送。但是，只限于系统字的长度 (SWL) \geq 数据字的长度 (DWL) \times 通道数 (CHNL) 的情况。

有效的设定和填充位数如表 18.9 所示。

无效的设定用“—”表示。

表 18.9 有效的设定和与填充位数

各系统字的填充位数			DWL[2:0]	000	001	010	011	100	101	110
CHNL[1:0]	各系统字的解码通道	SWL[2:0]	解码后的字长	8	16	18	20	22	24	32
00	1	000	8	0	—	—	—	—	—	—
		001	16	8	0	—	—	—	—	—
		010	24	16	8	6	4	2	0	—
		011	32	24	16	14	12	10	8	0
		100	48	40	32	30	28	26	24	16
		101	64	56	48	46	44	42	40	32
		110	128	120	112	110	108	106	104	96
		111	256	248	240	238	236	234	232	224
01	2	000	8	—	—	—	—	—	—	—
		001	16	0	—	—	—	—	—	—
		010	24	8	—	—	—	—	—	—
		011	32	16	0	—	—	—	—	—
		100	48	32	16	12	8	4	0	—
		101	64	48	32	28	24	20	16	0
		110	128	112	96	92	88	84	80	64
		111	256	240	224	220	216	212	208	192
10	3	000	8	—	—	—	—	—	—	—
		001	16	—	—	—	—	—	—	—
		010	24	0	—	—	—	—	—	—
		011	32	8	—	—	—	—	—	—
		100	48	24	0	—	—	—	—	—
		101	64	40	16	10	4	—	—	—
		110	128	104	80	74	68	62	56	32
		111	256	232	208	202	196	190	184	160
11	4	000	8	—	—	—	—	—	—	—
		001	16	—	—	—	—	—	—	—
		010	24	—	—	—	—	—	—	—
		011	32	0	—	—	—	—	—	—
		100	48	16	—	—	—	—	—	—
		101	64	32	0	—	—	—	—	—
		110	128	96	64	56	48	40	32	0
		111	256	224	192	184	176	168	160	128

当 SSI_CH0 ~ 5 作为发送器运行时，写给 SSITDR0 ~ 5 的各字按写的顺序发送到串行音频总线。
 当 SSI_CH0 ~ 5 作为接收器运行时，串行音频总线接收的各字按 SSIRDR0 ~ 5 接收的顺序读取。
 如图 18.6 ~ 图 18.8 所示，将 2、3 和 4 个通道的数据传送到串行音频总线。

图 18.6 是无填充位的例子，图 18.7 是向左排列的例子，图 18.8 是向右排列的例子。

这些例子都为任意例子。

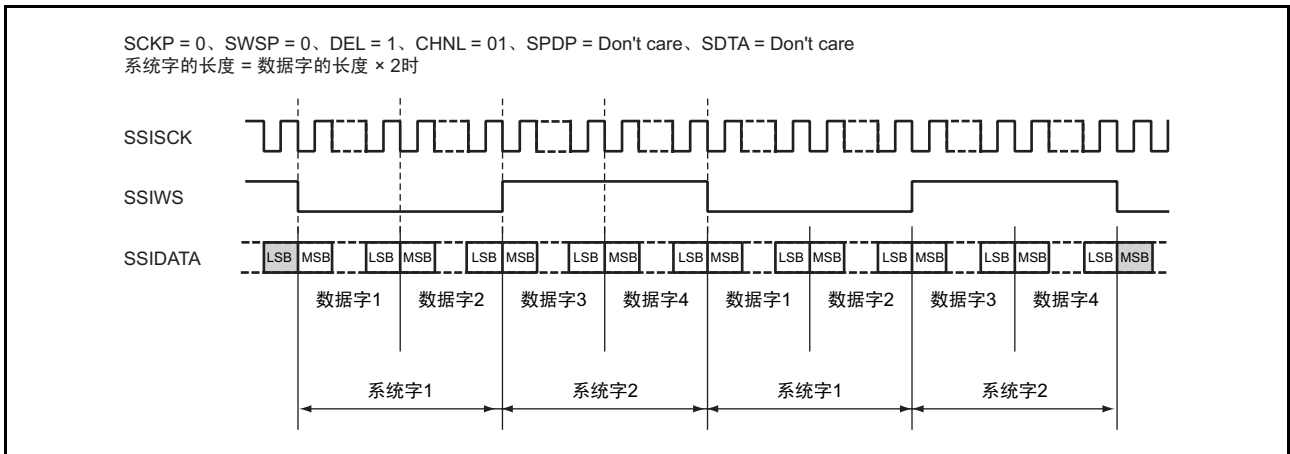


图 18.6 多通道格式 (2 个通道, 无填充)

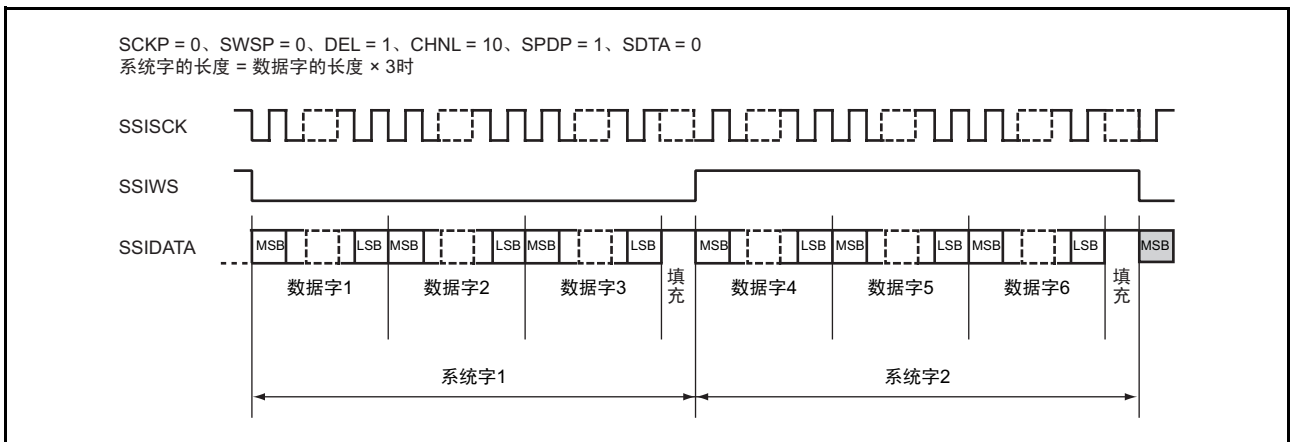


图 18.7 多通道格式 (3 个通道, High 填充)

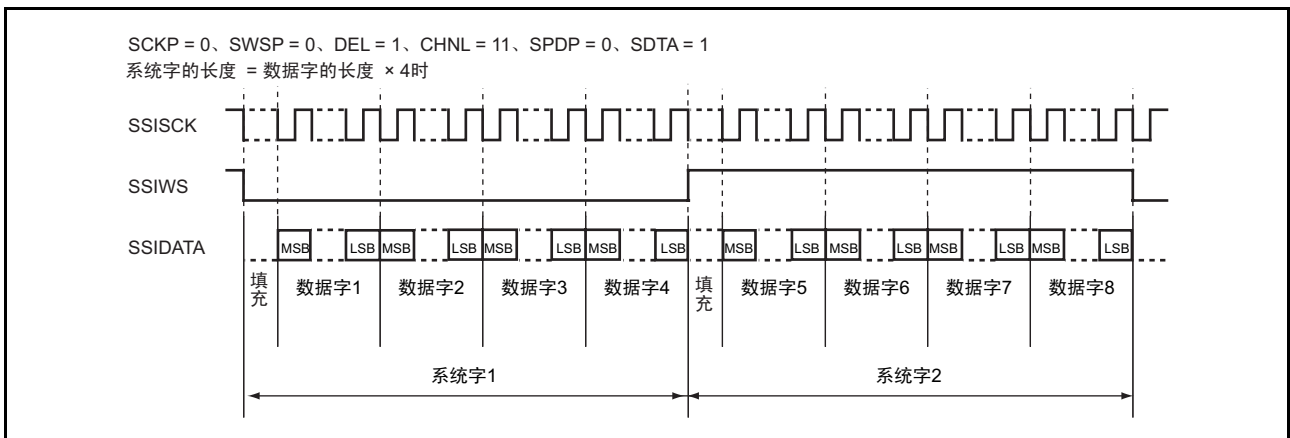


图 18.8 多通道格式
 (按 4 个通道、填充位、串行数据的顺序进行发送和接收, 有填充)

(g) 运行设定的格式设定位

非压缩模式的其他结构位如下所示。这些位不相互排斥，但是根据组合会出现不实用的情况。对这些结构位，参照图 18.9 基本的格式例子进行说明。

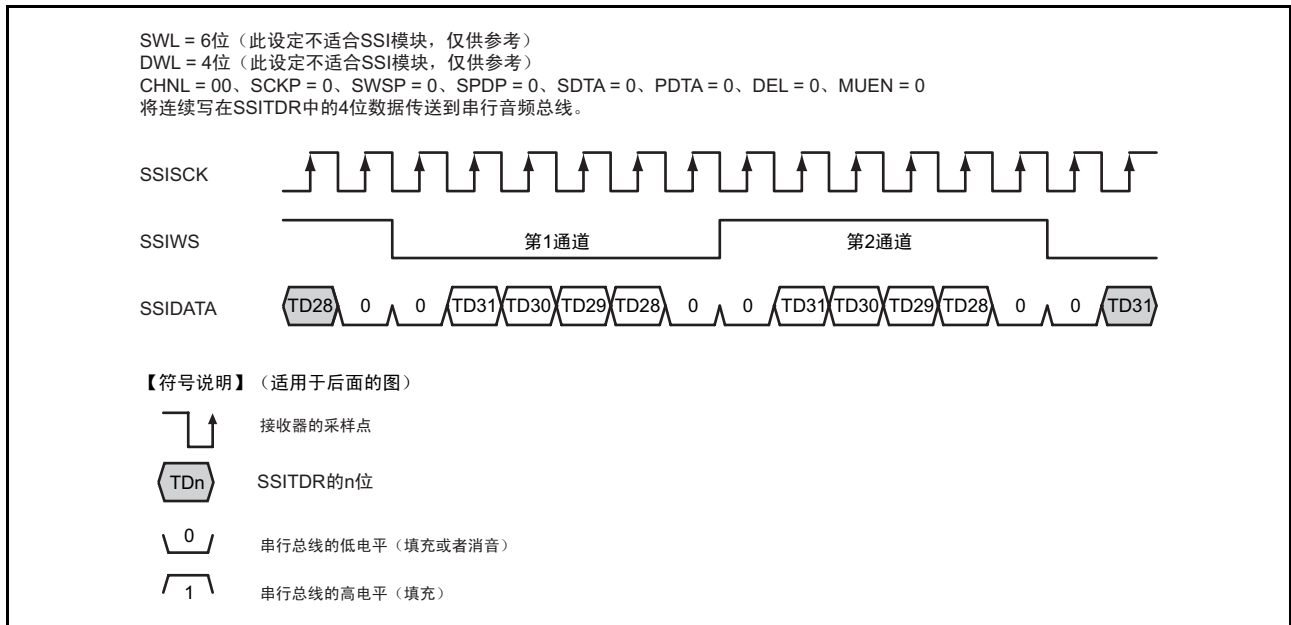


图 18.9 基本格式例子 (发送模式、任意的系统 / 数据字长度)

图 18.9 的例子使用了 6 位系统字和 4 位数据字。

尽管这些设定不能用于 SSI_CHO ~ 5，但是以次为例说明其他设定位。

• 反相时钟

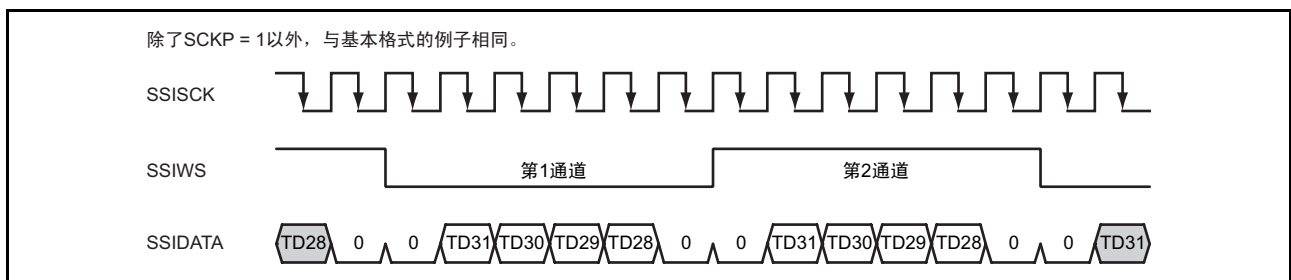


图 18.10 反相时钟

• 反相字选择信号

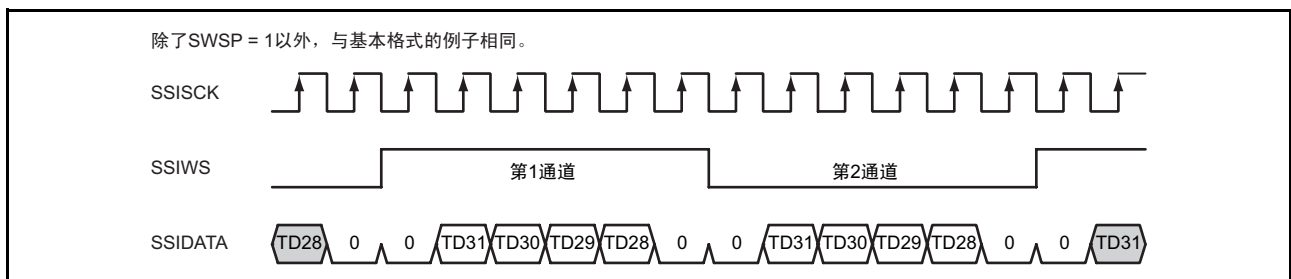


图 18.11 反相字选择信号

- 反相填充极性

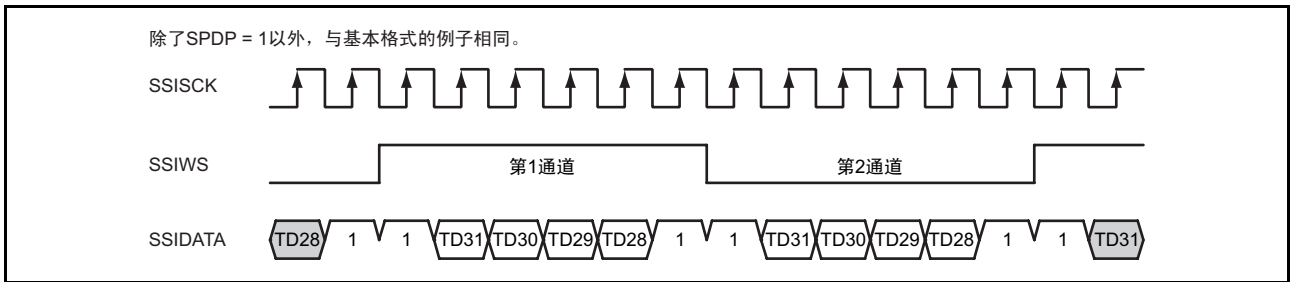


图 18.12 反相填充极性

- 按填充位、串行数据的顺序进行发送和接收，有延迟

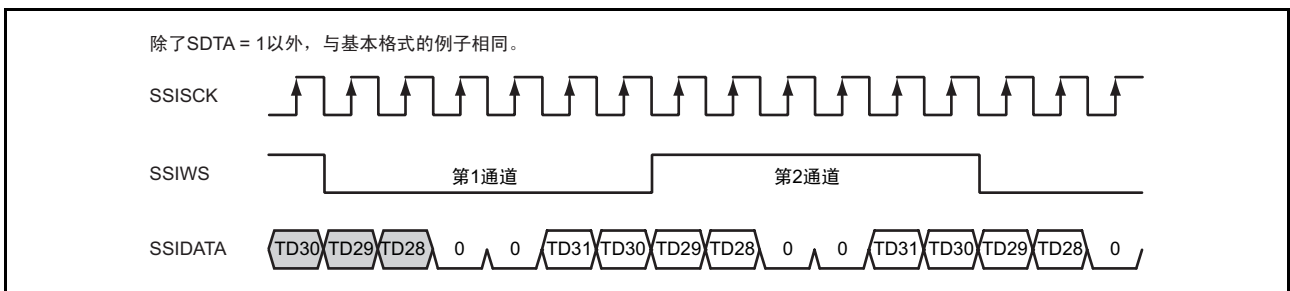


图 18.13 按填充位、串行数据的顺序进行发送和接收，有延迟

- 按填充位、串行数据的顺序进行发送和接收，无延迟

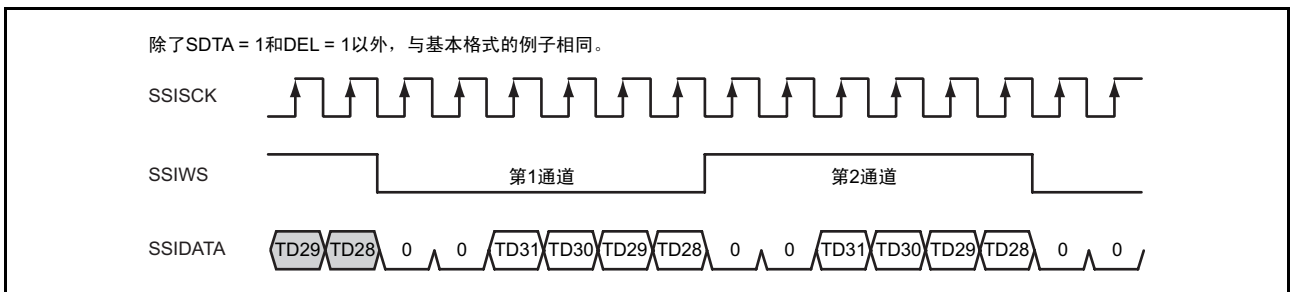


图 18.14 按填充位、串行数据的顺序进行发送和接收，无延迟

- 按串行数据、填充位的顺序进行发送和接收，无延迟

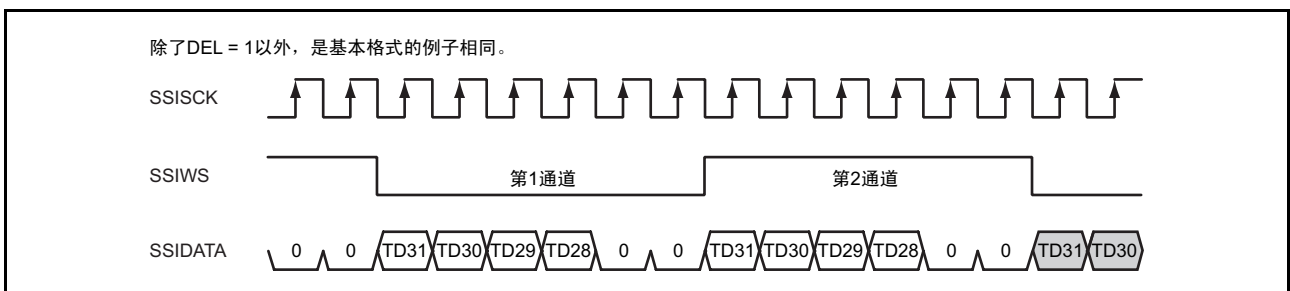


图 18.15 按串行数据、填充位的顺序进行发送和接收，无延迟

- 并行数据向右排列，有延迟

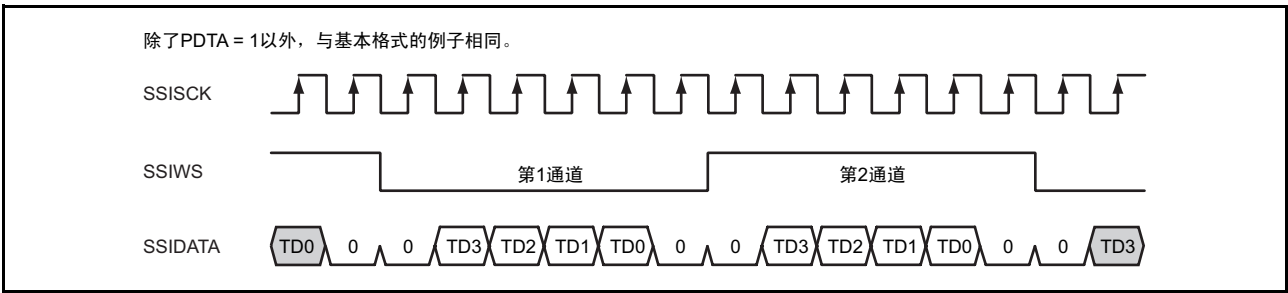


图 18.16 并行数据向右排列，有延迟

- 静音有效

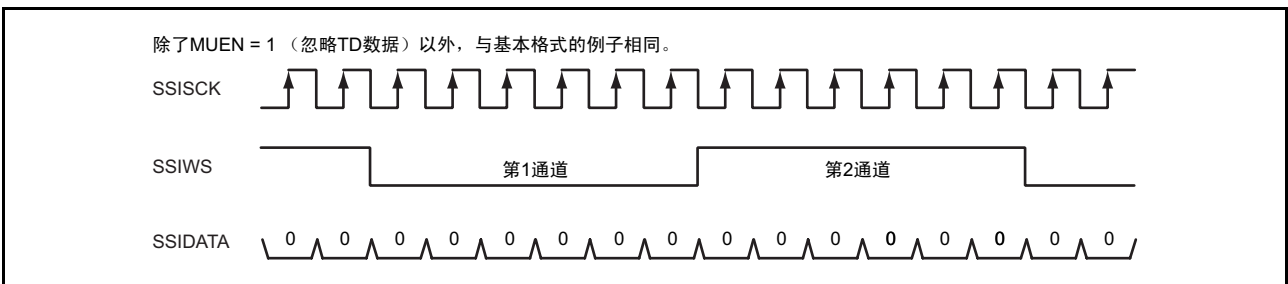


图 18.17 静音有效

(3) 运行模式

运行模式有结构、有效和无效 3 个模式。
运行模式的转移图如图 18.18 所示。

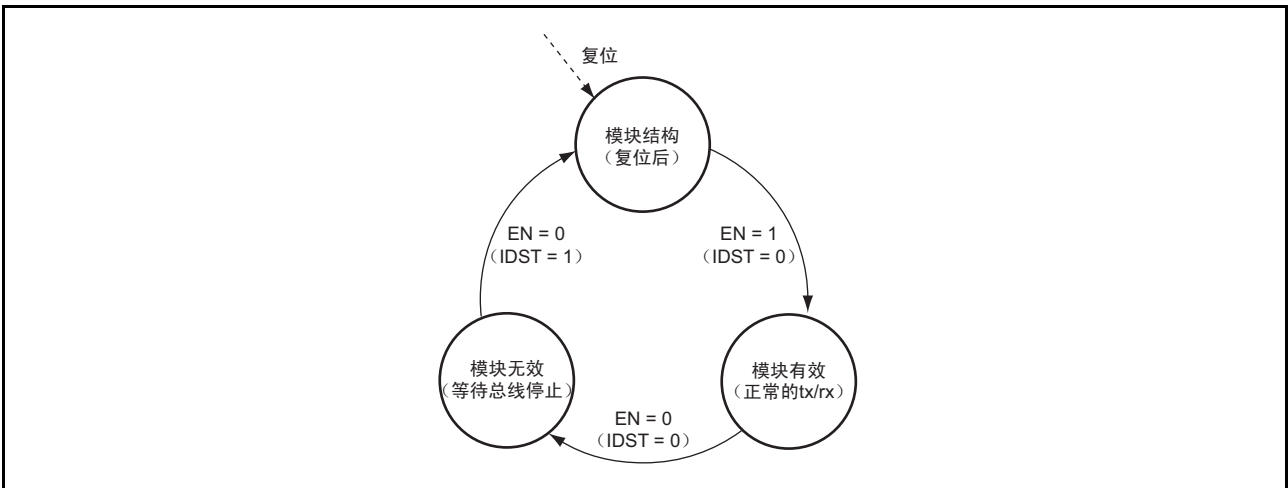


图 18.18 运行模式转移图

- 结构模式
在解除复位后进入此模式。
SSI_CH0 ~ 5 在通过 EN 位变为有效前，需要在此模式中对控制寄存器进行必要的设定。
如果将 EN 位置位，SSI_CH0 ~ 5 就转移到模块有效模式。
- 模块有效模式
此模式的运行取决于选择的运行模式。
详细内容请参照“18.4.3(4) 发送运行”和“18.4.3(5) 接收运行”。

(4) 发送运行

能通过 DMA 或者中断进行发送的控制。

DMA 控制能降低 CPU 负担。在 DMA 控制模式中，通过中断来通知数据下溢 / 上溢的发生以及 SSI_DMACH0 ~ 1 的传送结束。

作为其他的控制方法，还有使用中断的方法。根据需要使 SSI_CH0 ~ 5 提供数据。因为 SSI_CH0 ~ 5 是双缓冲器结构，至少要在每个系统字周期写数据，所以中断控制模式会产生高的负担。

当 SSI_CH0 ~ 5 无效时，在 SSISR0 ~ 5 的 IIRQ 位表示 SSI_CH0 ~ 5 处于空闲状态之前，必须继续提供 SSI 时钟*。

DMA 控制模式和中断控制模式的发送运行分别如图 18.19 和图 18.20 所示。

【注】 * 在 SCKD=0 时，是 SSISCK[5:0] 引脚的输入时钟
在 SCKD=1 时，是 AUDIO_CLK[5:0] 引脚的输入时钟

(a) 使用 SSI_DMACH0 ~ 1 的发送

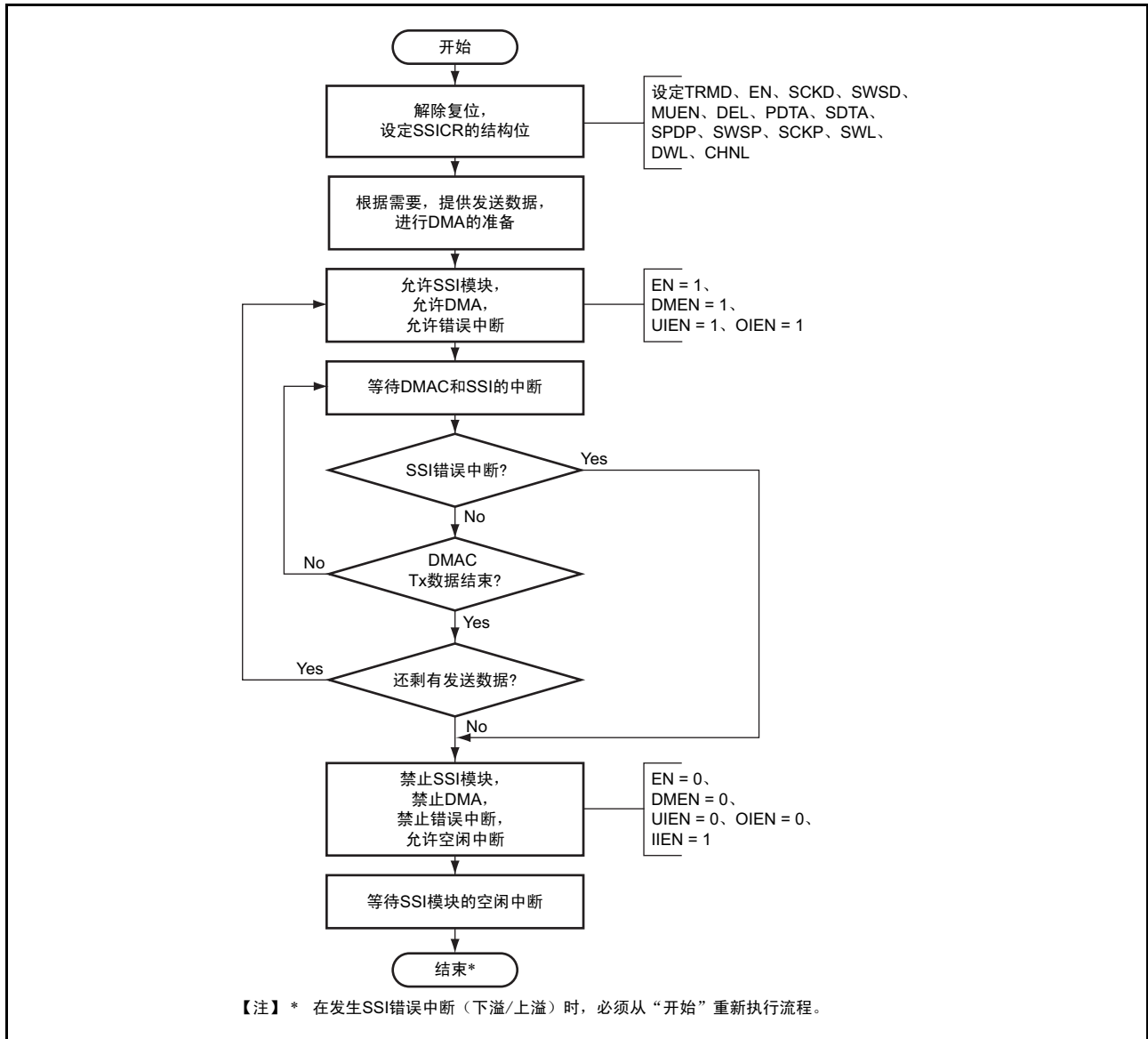


图 18.19 使用 SSI_DMACH0、1 的发送

(b) 使用中断数据流控制的发送

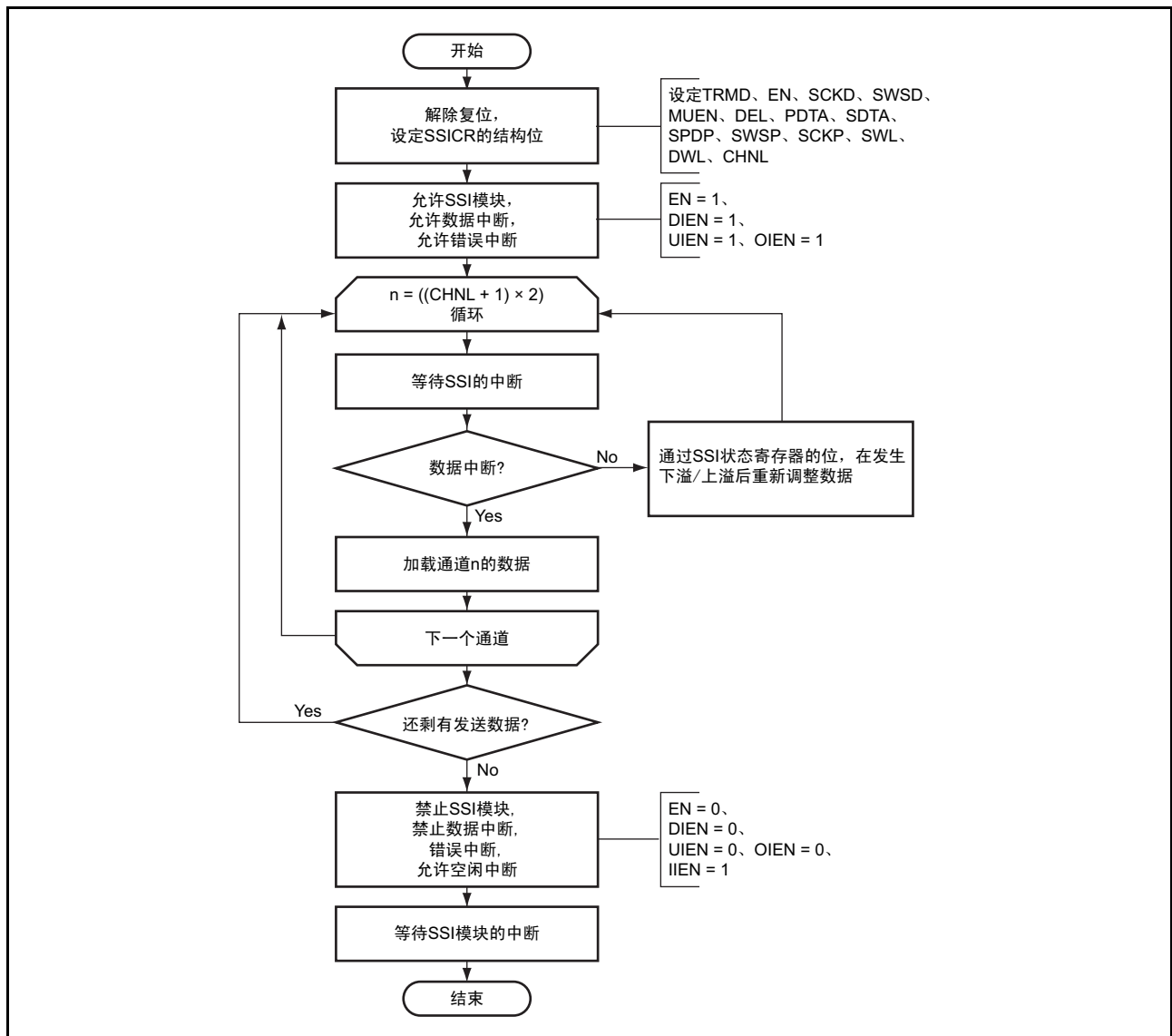


图 18.20 使用中断数据流控制的发送

(5) 接收运行

与发送相同，也能通过 DMA 或者中断进行接收的控制。

各自的运行流程分别如图 18.21 和图 18.22 所示。

当 SSI_CH0 ~ 5 无效时，在 SSISR0 ~ 5 的 IIRQ 位表示 SSI_CH0 ~ 5 处于空闲状态之前，必须继续提供 SSI 时钟*。

【注】* 当 SCKD=0 时，是 SSISCK[5:0] 引脚的输入时钟
当 SCKD=1 时，是 AUDIO_CLK[5:0] 引脚的输入时钟

(a) 使用 SSI_DMAC0 ~ 1 的接收

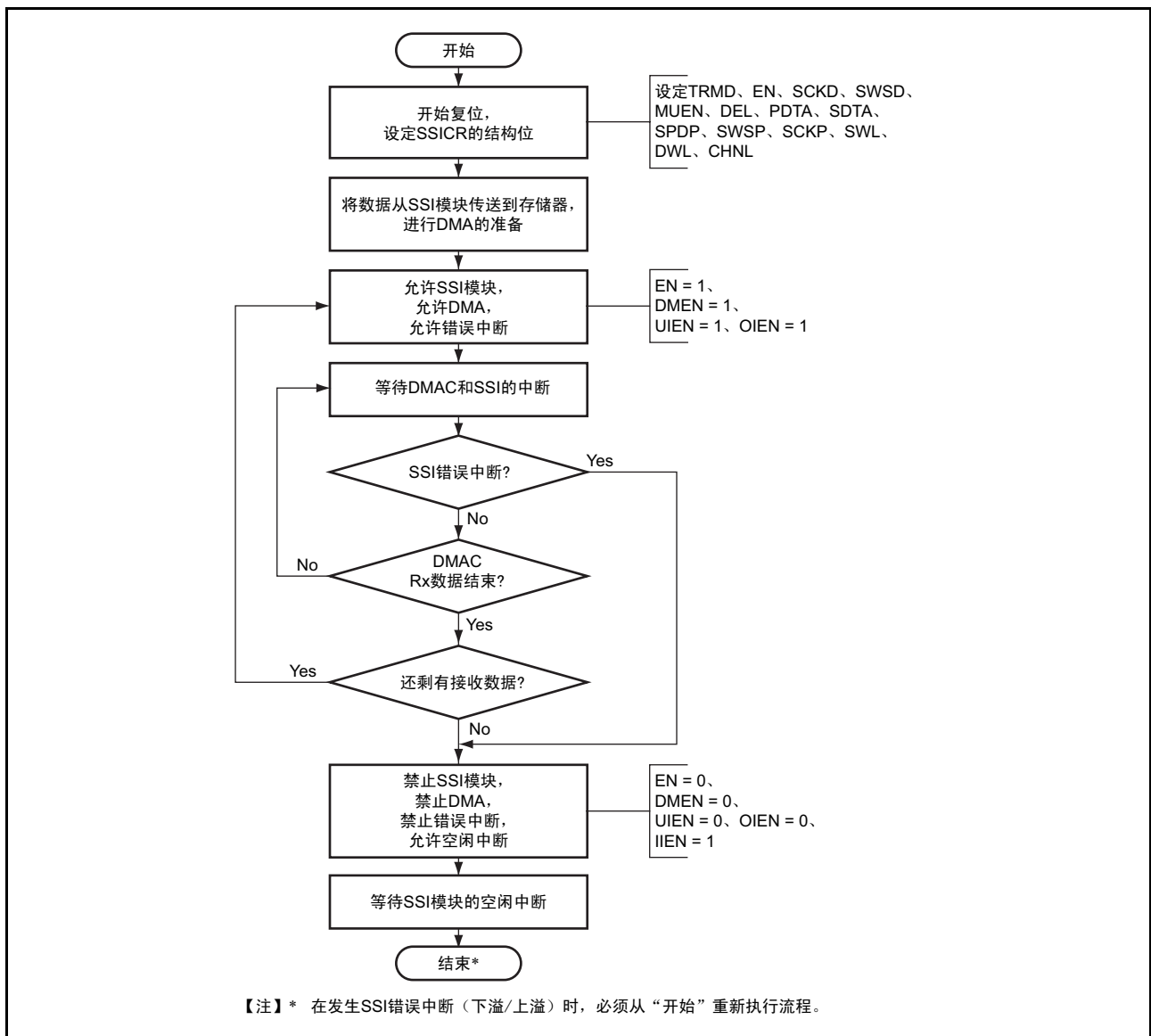


图 18.21 使用 SSI_DMAC0、1 的接收

(b) 使用中断数据流控制的接收

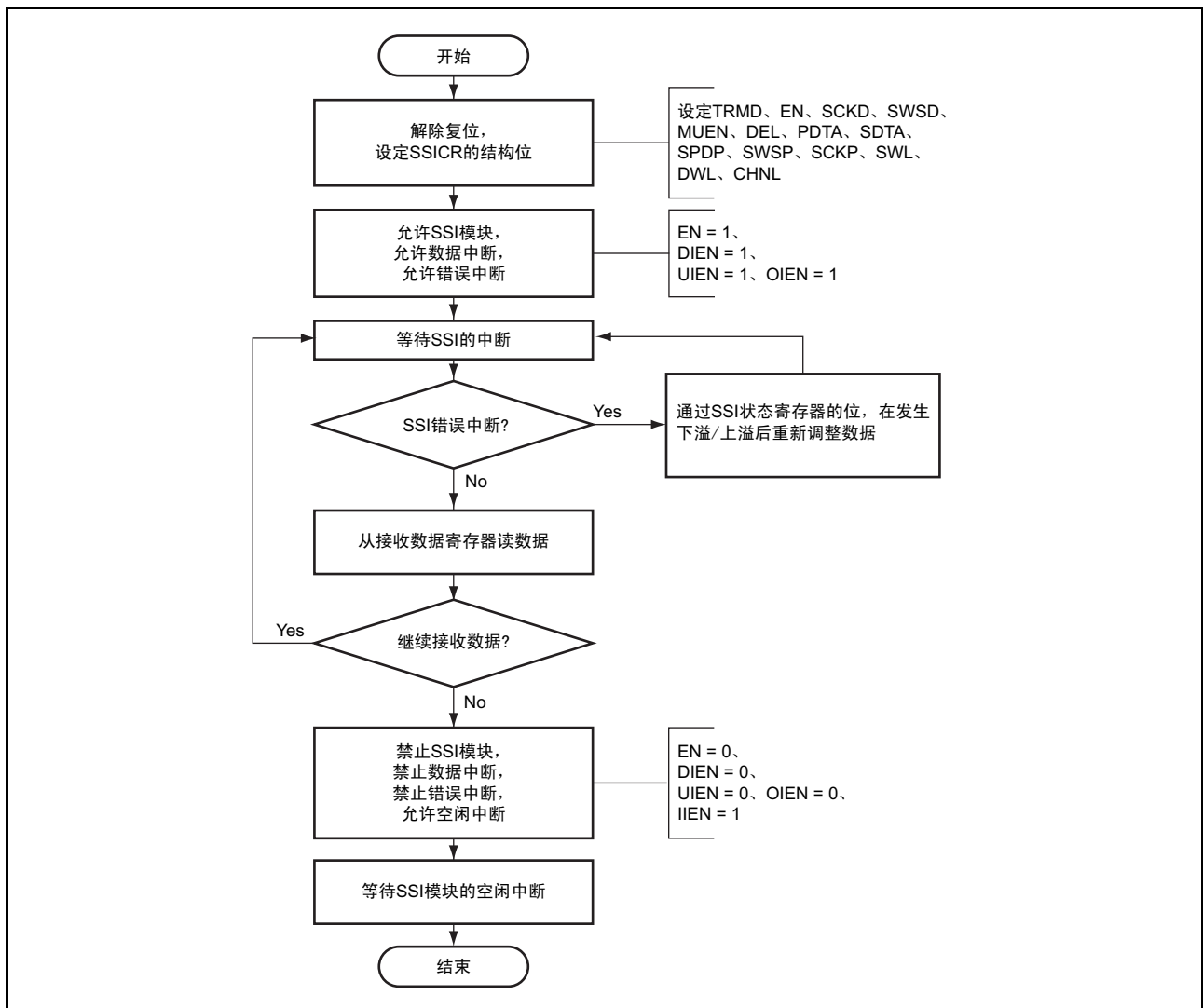


图 18.22 使用中断数据流控制的接收

在下溢或者上溢的条件成立时，能通过 SSISR0 ~ 5 的 CHNO[1:0] 位和 SWNO 位使 SSI_CH0 ~ 5 恢复到条件成立前的状态。

当发生下溢或者上溢时，主 CPU 能通过读通道数和系统字数得知串行音频流到达的位置。

当作为发送器运行时，在到达 SSI_CH0 ~ 5 发送下一个预定数据前，主 CPU 能跳读发送数据，所以能再次和音频数据流同步。当作为接收器运行时，在能保存表示 SSI_CH0 ~ 5 模块进行下一次接收的数据前，主 CPU 通过保存空数据取得接收数据个数的一致性，能再次和音频数据流同步。

(6) 串行位时钟控制

使用串行位时钟功能，控制和选择用于串行总线接口的时钟。

如果将串行位时钟方向设定为输入 (SSICR.SCKD=0)，SSI_CH0 ~ 5 就为时钟从属模式，移位寄存器使用的位时钟为 SSISCK[5:0] 引脚的输入时钟。

如果将串行位时钟方向设定为输出 (SSICR.SCKD=1)，SSI_CH0 ~ 5 就为时钟主控模式，移位寄存器使用的位时钟为 AUDIO_CLK[5:0] 引脚的输入时钟或者其分频时钟。按照 SSICR_0 ~ 5 的串行过采样时钟分频比位 (SSICR.CKDV) 设定的比率，将 AUDIO_CLK[5:0] 引脚的输入时钟分频后的时钟用作移位寄存器的位时钟。

无论上述哪种情况，SSISCK[5:0] 引脚的输出都和位时钟相同。

18.5 使用时的注意事项

18.5.1 接收 DMA 运行中发生上溢时的限制事项

如果在接收 DMA 运行中发生上溢，需要重新启动 SSI_CH0 ~ 5 中的相应模块。SSI_CH0 ~ 5 中的接收缓冲器由 L 通道和 R 通道共用的 32 位寄存器构成。例如，控制寄存器 0 ~ 5 (SSICR0 ~ 5) 的数据字长度 (DWL2 ~ DWL0) 为 32 位并且系统字长度 (SWL2 ~ SWL0) 为 32 位时，如果发生上溢，应该 L 通道接收的数据可能会被 R 通道接收。

在此，当通过上溢错误中断或者上溢错误状态标志 (SSISR0 ~ 5 的 OIRQ 位) 确认上溢后，必须将 SSICR0 ~ 5 的 EN 位和 DMEN 位置 0，禁止 SSI_CH0 ~ 5 的 DMA，使其停止运行（此时，也必须停止 SSI_DMAC0/1 的设定）。

然后，将 OIRQ 位置 0，清除上溢状态，并且再次设定 DMA，重新开始传送。

18.5.2 从属模式中运行时的限制事项

在从属模式中运行时，必须在停止串行字选择信号 (SSIWS[5:0]) 输入前结束数据传送 (SSICR0 ~ 5 的 EN 位 = 0)。

在从属模式时，清除 EN 位（停止传送），通过检测串行字选择信号 (SSIWS[5:0]) 的下降沿结束数据传送。

如果停止串行字选择信号输入，就无法检测串行字选择信号的下降沿，无法正常结束数据传送。

18.5.3 各种寄存器设定时的注意事项

1. 必须考虑 FIFO 缓冲器的大小 (64 字节: 32bit×16 段) 来设定 DMA 模式寄存器 (SSIDMMR) 的 RDMA 最大突发长度 (RDMBSZ) 和 WDMA 最大突发长度 (WDMBSZ)。
2. 必须将 RDMA 传送字节数寄存器 (SSIRDMCNTR) 的字节数设定为 RDMA 最大突发长度 (RDMBSZ) 的倍数。例如: RDMBSZ=4 个突发 (32 字节) 时, 为 32、64、96、128 ···。
也必须将 WDMA 传送字节数寄存器 (SSIWDMCNTR) 的字节数设定为 WDMA 最大突发长度 (WDMBSZ) 的倍数。
另外, 读 RDMA 传送字节数寄存器 (SSIRDMCNTR) 和 WDMA 传送字节数寄存器 (SSIWDMCNTR) 时总是读取设定值。
3. 必须将块计数器源寄存器 (SSIBLCNTR) 的字节数设定为 RDMA 或者 WDMA 最大突发长度的倍数。
4. 不能在 DMA 传送中写发送暂停块计数器 (SSISTPBLCNT) 和发送暂停中传送数据寄存器 (SSISTPDR)。
5. 读 RDMA 传送字节数寄存器 (SSIRDMCNTR) 和 WDMA 传送字节数寄存器 (SSIWDMCNTR) 时总是读取设定值。有关数据传送字节数, 请参照块计数器 (SSIBLCNT) 和块 xn 次计数器 (SSIBLNCNT)。
6. 块计数器 (SSIBLCNT) 和块 xn 次计数器 (SSIBLNCNT) 是只读计数器, 能通过软件复位 (DMRST) 清除。

在发送运行中, 当数据从 SSI-DMAC 的发送 FIFO 传送到 SSI 的数据缓冲器时进行计数; 在接收运行中, 当数据从 SSI 的数据缓冲器传送到 SSI-DMAC 的接收 FIFO 时进行计数。

第 19 章 以太网控制器 (EtherC)

本 LSI 内置符合以太网或者 IEEE802.3 的 MAC (Media Access Control) 层规格的以太网控制器 (EtherC)。EtherC 能通过连接符合此规格的物理层 LSI (PHY-LSI) 进行以太网 /IEEE802.3 帧的发送和接收。本 LSI 内置的以太网控制器有 1 个系统的 MAC 层接口, 并且以太网控制器在本 LSI 内部连接直接存储器存取控制器 (E-DMAC, 用于以太网控制器), 能与存储器进行高速存取。

EtherC 的结构如图 19.1 所示。

19.1 特点

- 进行以太网 /IEEE802.3 帧的发送和接收。
- 支持 10Mbps 和 100Mbps 的传送。
- 支持全双工模式和半双工模式。
- 支持 IEEE802.3u 规格的 MII (Media Independent Interface)。
- 检测 Magic Packet 并且输出 Wake-On-LAN (WOL) 信号。
- 符合 IEEE802.3x 规格的流控制

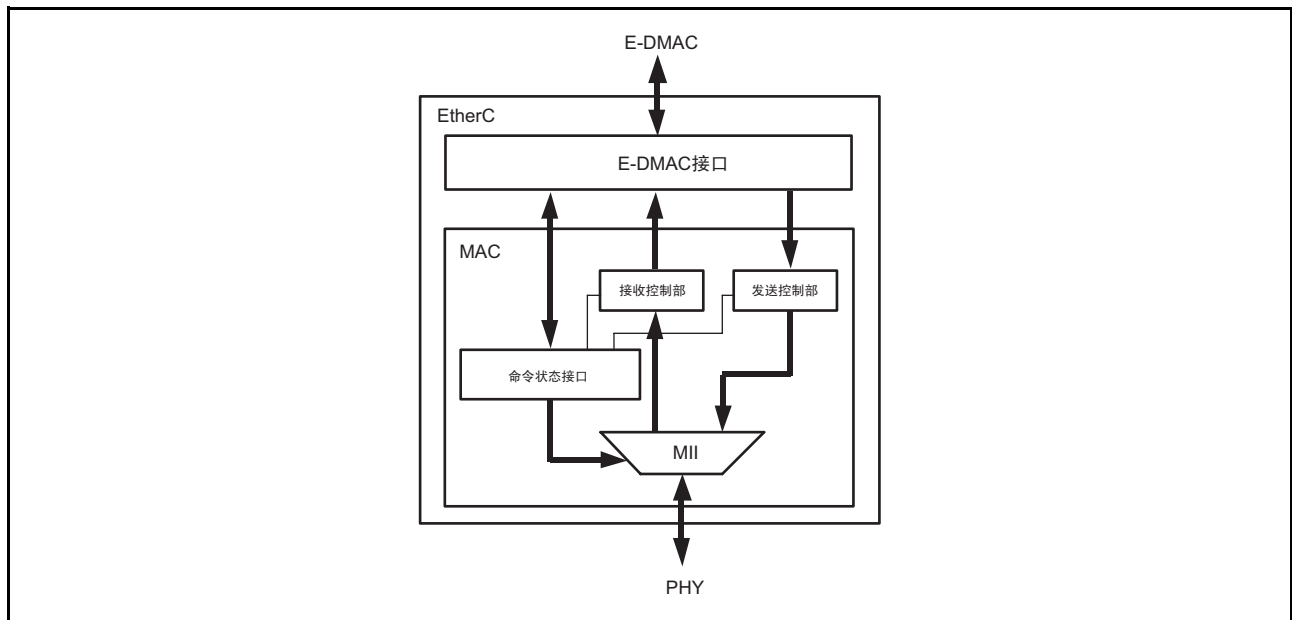


图 19.1 EtherC 的结构

19.2 输入 / 输出引脚

EtherC 的引脚结构如表 19.1 所示。

表 19.1 引脚结构

名称	引脚名	输入 / 输出	说明
发送时钟 *	TX-CLK	输入	是 TX-EN、MII_TXD3 ~ MII_TXD0 和 TX-ER 的时序参照信号。
接收时钟 *	RX-CLK	输入	是 RX-DV、MII_RXD3 ~ MII_RXD0 和 RX-ER 时序参照信号。
发送允许 *	TX-EN	输出	是表示在 MII_TXD3 ~ MII_TXD0 上已准备好发送数据的信号。
发送数据 *	MII_TXD3 ~ MII_TXD0	输出	4 位发送数据
发送错误 *	TX-ER	输出	将发送中的错误通知 PHY-LSI。
接收数据有效 *	RX-DV	输入	是表示在 MII_RXD3 ~ MII_RXD0 上有有效接收数据的信号。
接收数据 *	MII_RXD3 ~ MII_RXD0	输入	4 位接收数据
接收错误 *	RX-ER	输入	识别在数据接收中发生的错误状态。
载波检测 *	CRS	输入	载波检测信号
冲突检测 *	COL	输入	冲突检测信号
管理数据时钟 *	MDC	输出	是用于 MDIO 信息传送的参照时钟信号。
管理数据输入 / 输出 *	MDIO	输入 / 输出	是用于在 STA 和 PHY 之间交换管理信息的双向信号。
连接状态	LNKSTA	输入	是从 PHY-LSI 输入的连接状态
通用外部输出	EXOUT	输出	外部输出引脚
Wake-On-LAN	WOL	输出	是表示 Magic Packet 接收的信号。

【注】 * 符合 IEEE802.3u 的 MII 信号。

19.3 寄存器说明

EtherC 的寄存器结构如表 19.2 所示。

表 19.2 寄存器结构

名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度
Ether 模式寄存器	ECMR	R/W	H'FEF0 0100	H'1EF0 0100	32
EtherC 状态寄存器	ECSR	R/W	H'FEF0 0110	H'1EF0 0110	32
EtherC 中断允许寄存器	ECSIPR	R/W	H'FEF0 0118	H'1EF0 0118	32
接收帧长度上限寄存器	RFLR	R/W	H'FEF0 0108	H'1EF0 0108	32
PHY 部接口寄存器	PIR	R/W	H'FEF0 0120	H'1EF0 0120	32
MAC 地址高位设定寄存器	MAHR	R/W	H'FEF0 01C0	H'1EF0 01C0	32
MAC 地址低位设定寄存器	MALR	R/W	H'FEF0 01C8	H'1EF0 01C8	32
PHY 部状态寄存器	PSR	R	H'FEF0 0128	H'1EF0 0128	32
试发送失败计数寄存器	TROCR	R/W	H'FEF0 01D0	H'1EF0 01D0	32
延迟冲突检测计数寄存器	CDCR	R/W	H'FEF0 01D4	H'1EF0 01D4	32
载波消失计数寄存器	LCCR	R/W	H'FEF0 01D8	H'1EF0 01D8	32
载波未检测计数寄存器	CNDCR	R/W	H'FEF0 01DC	H'1EF0 01DC	32
CRC 错误帧接收计数寄存器	CEFCR	R/W	H'FEF0 01E4	H'1EF0 01E4	32
帧接收错误计数寄存器	FRECR	R/W	H'FEF0 01E8	H'1EF0 01E8	32
64 字节未帧接收计数寄存器	TSFRCR	R/W	H'FEF0 01EC	H'1EF0 01EC	32
指定字节超过帧接收计数寄存器	TLFRCR	R/W	H'FEF0 01F0	H'1EF0 01F0	32
剩余位帧接收计数寄存器	RFCR	R/W	H'FEF0 01F4	H'1EF0 01F4	32
多播地址帧接收计数寄存器	MAFCR	R/W	H'FEF0 01F8	H'1EF0 01F8	32
IPG 设定寄存器	IPGR	R/W	H'FEF0 0150	H'1EF0 0150	32
自动 PAUSE 帧设定寄存器	APR	R/W	H'FEF0 0154	H'1EF0 0154	32
手动 PAUSE 帧设定寄存器	MPR	R/W	H'FEF0 0158	H'1EF0 0158	32
自动 PAUSE 帧再发送次数设定寄存器	TPAUSER	R/W	H'FEF0 0164	H'1EF0 0164	32
随机数生成计数器上限值设定寄存器	RDMLR	R/W	H'FEF0 0140	H'1EF0 0140	32
接收 PAUSE 帧计数器	RFCF	R/W	H'FEF0 0160	H'1EF0 0160	32
PAUSE 帧再发送次数计数器	TPAUSECR	R	H'FEF0 0168	H'1EF0 0168	32
Broadcast 帧接收次数设定寄存器	BCFRR	R	H'FEF0 016C	H'1EF0 016C	32

【注】 * P4 区地址是使用虚拟地址空间 P4 区的地址，区域 7 地址是使用 TLB 存取物理地址空间区域 7 的地址。

EtherC 各处理模式的寄存器状态如表 19.3 所示。

表 19.3 各处理模式的寄存器状态

名称	略称	软件复位时
EtherC 模式寄存器	ECMR	○
EtherC 状态寄存器	ECSR	○
EtherC 中断允许寄存器	ECSIPR	○
接收帧长度上限寄存器	RFLR	○
PHY 部接口寄存器	PIR	○
MAC 地址高位设定寄存器	MAHR	○
MAC 地址低位设定寄存器	MALR	○
PHY 部状态寄存器	PSR	○
试发送失败计数寄存器	TROCR	○
延迟冲突检测计数寄存器	CDCR	○
载波消失计数寄存器	LCCR	○
载波未检测计数寄存器	CNDCR	○
CRC 错误帧接收计数寄存器	CEFCR	○
帧接收错误计数寄存器	FRECR	○
64 字节未帧接收计数寄存器	TSFRCR	○
指定字节超过帧接收计数寄存器	TLFRCR	○
剩余位帧接收计数寄存器	RFRCR	○
多播地址帧接收计数寄存器	MAFCR	○
IPG 设定寄存器	IPGR	○
自动 PAUSE 帧设定寄存器	APR	○
手动 PAUSE 帧设定寄存器	MPR	○
自动 PAUSE 帧再发送次数设定寄存器	TPAUSER	○
随机数生成计数器上限值设定寄存器	RDMLR	○
接收 PAUSE 帧计数器	RFCE	○
PAUSE 帧再发送次数计数器	TPAUSECR	○
Broadcast 帧接收次数设定寄存器	BCFRR	○

【符号说明】

- ：被初始化的寄存器
- ：被保持的寄存器

19.3.1 EtherC 模式寄存器 (ECMR)

ECMR 是 32 位可读写寄存器，指定以太网控制器的运行模式。通常，在复位后的初始设定时设定此寄存器。

在发送和接收功能有效状态下，禁止改写运行模式的设定。切换运行模式时，必须在通过 EDMR 的 SWR 位将 EtherC 和 E-DMAC 恢复到初始状态后，重新设定此寄存器。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	TPC	ZPF	PFR	RXF	TXF
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PRCEF	—	—	MPDE	—	—	RE	TE	—	ILB	—	DM	PRM
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R/W	R	R	R/W	R/W	R	R/W	R	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 21	—	全 0	R	保留位 读取值为 0，只能写 0。
20	TPC	0	R/W	PAUSE 帧发送位 0: 在 PAUSE 期间不发送 PAUSE 帧 1: 在 PAUSE 期间发送 PAUSE 帧
19	ZPF	0	R/W	0 time PAUSE 帧使用允许位 0: TIME 参数为 0 的 PAUSE 帧控制无效 在经过 Timer 值所示时间之前不发送下一个帧。在接收 Timer 值所示时间为 0 的 PAUSE 帧后，放弃 PAUSE 帧。 1: TIME 参数为 0 的 PAUSE 帧控制有效 在未经过 Timer 值所示时间的状态下，如果接收 FIFO 的数据量未达到 FCFTR 设定值，就发送 Timer 值为 0 的自动 PAUSE 帧。在接收 Timer 值所示时间为 0 的 PAUSE 帧后，解除发送等待状态。
18	PFR	0	R/W	PAUSE 帧接收模式位 0: 不将 PAUSE 帧传送到 E-DMAC 1: 将 PAUSE 帧传送到 E-DMAC
17	RXF	0	R/W	接收部流控制的运行模式位 0: PAUSE 帧的检测功能无效 1: 接收部流控制功能有效
16	TXF	0	R/W	发送部流控制的运行模式位 0: PAUSE 帧的检测功能无效 (不发送自动 PAUSE 帧) 1: 发送部流控制功能有效 (根据需要发送自动 PAUSE 帧)
15 ~ 13	—	全 0	R	保留位 读取值为 0，只能写 0。
12	PRCEF	0	R/W	CRC 错误帧接收允许位 0: 将发生 CRC 错误的接收帧作为错误帧 1: 不将发生 CRC 错误的接收帧作为错误帧
11, 10	—	全 0	R	保留位 读取值为 0，只能写 0。

位	位名	初始值	R/W	说明
9	MPDE	0	R/W	Magic Packet 检测允许位 为了允许从以太网的启动, 通过硬件选择是否允许 Magic Packet 检测功能。 0: 不允许 Magic Packet 检测 1: 允许 Magic Packet 检测
8, 7	—	全 0	R	保留位 读取值为 0, 只能写 0。
6	RE	0	R/W	接收允许位 在此位从接收功能有效 (RE=1) 变为无效 (RE=0) 时, 如果有正在接收的帧, 就在该帧接收结束前接收功能有效。 0: 接收功能无效 1: 接收功能有效
5	TE	0	R/W	发送允许位 在此位从发送功能有效 (TE=1) 变为无效 (TE=0) 时, 如果有正在发送的帧, 就在该帧发送结束前发送功能有效。 0: 发送功能无效 1: 发送功能有效
4	—	0	R	保留位 读取值为 0, 只能写 0。
3	ILB	0	R/W	内部回送模式位 指定 EtherC 内部的回送模式。 0: 正常的的数据发送 / 接收 1: 当 DM=1 时, 在 EtherC 的 MAC 内部进行数据回送
2	—	0	R	保留位 读取值为 0, 只能写 0。
1	DM	0	R/W	双向模式 指示 EtherC 的传送模式。 0: 指定半双工传送模式 1: 指定全双工传送模式
0	PRM	0	R/W	混杂模式 如果设定此位, 就能接收全部以太网帧。此全部以太网帧是指全部能接收的帧, 与目的地址、广播地址、多播位等是否有差异无关。 0: EtherC 正常运行 1: EtherC 以混杂模式运行

19.3.2 EtherC 状态寄存器 (ECSR)

ECSR 是 32 位可读写寄存器，表示 EtherC 的内部状态。能通过中断将本状态通知 CPU。如果给 PSRTO、LCHNG、MPD 和 ICD 位写 1，就能清除对应的标志。写 0 时不影响标志。对于产生中断的位，能通过 ECSIPR 寄存器的对应位允许或者禁止该中断。

由本状态寄存器引起的中断反映在 E-DMAC 的 EESR 寄存器的 ECI 位。

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
	—	—	—	—	—	—	—	—	—	—	BFR	PSRTO	—	LCHNG	MPD	ICD
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 6	—	全 0	R	保留位 读取值为 0，只能写 0。
5	BFR	0	R/W	Broadcast 帧连续接收中断 < 中断源 > 位 表示连续接收 Broadcast 帧。
4	PSRTO	0	R/W	PAUSE 帧试发送失败位 在使用流控制重新发送 PAUSE 帧时，表示重新发送次数超过自动 PAUSE 帧再发送次数设定寄存器 (TPAUSER) 的再发送上限值。 0: PAUSE 帧重新发送次数未超过上限值 1: PAUSE 帧重新发送次数超过上限值
3	—	0	R	保留位 读取值为 0，只能写 0。
2	LCHNG	0	R/W	连接信号变化位 表示从 PHY-LSI 输入的 LNKSTA 信号从高电平变到低电平或者从低电平变到高水平。 在确认当前的 Link 状态时，必须参照 PHY 部的状态寄存器 (PSR) 的 LMON 位。 0: 未检测到 LNKSTA 信号的变化 1: 检测到 LNKSTA 信号的变化 (从高电平变到低电平或者从低电平变到高水平)
1	MPD	0	R/W	Magic Packet 检测位 表示在线路上检测到 Magic Packet。 0: 未检测到 Magic Packet 1: 检测到 Magic Packet
0	ICD	0	R/W	非法载波检测位 表示 PHY-LSI 在线路上检测到非法载波。即，在从 PHY-LSI 通知该 LSI 的信号为 RX-DV=0、RX-ER=1 且为 MII-RXD3 ~ 0=1110 时，此位被置位 (参照图 19.4 (6))。 但是，从 PHY-LSI 输入的信号变化快于软件的识别时间时，可能得不到正确的信息。请参照使用的 PHY-LSI 时序。 0: PHY-LSI 在线路上未检测到非法载波 1: PHY-LSI 在线路上检测到非法载波

19.3.3 EtherC 中断允许寄存器 (ECSIPR)

ECSIPR 是 32 位可读写寄存器，允许通过 ECSR 寄存器报告的中断源。各位能允许对应 ECSR 位的中断。

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
	—	—	—	—	—	—	—	—	—	—	BFSIPR	PSRTO IP	—	LCHN GIP	MPDIP	ICDIP
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 6	—	全 0	R	保留位 读取值为 0，只能写 0。
5	BFSIPR	0	R/W	Broadcast 帧连续接收中断允许位 0: 允许由 ECSR 的对应位产生的中断 1: 禁止由 ECSR 的对应位产生的中断
4	PSRTOIP	0	R/W	Pause 帧试发送失败中断允许位 0: 禁止 PSRTO 的中断通知 1: 允许 PSRTO 的中断通知
3	—	0	R	保留位 读取值为 0，只能写 0。
2	LCHNGIP	0	R/W	连接信号变化中断允许位 0: 禁止 LCHNG 的中断通知 1: 允许 LCHNG 的中断通知
1	MPDIP	0	R/W	Magic Packet 检测中断允许位 0: 禁止 MPD 的中断通知 1: 允许 MPD 的中断通知
0	ICDIP	0	R/W	非法载波检测中断允许位 0: 禁止 ICD 的中断通知 1: 允许 ICD 的中断通知

19.3.4 PHY 部接口寄存器 (PIR)

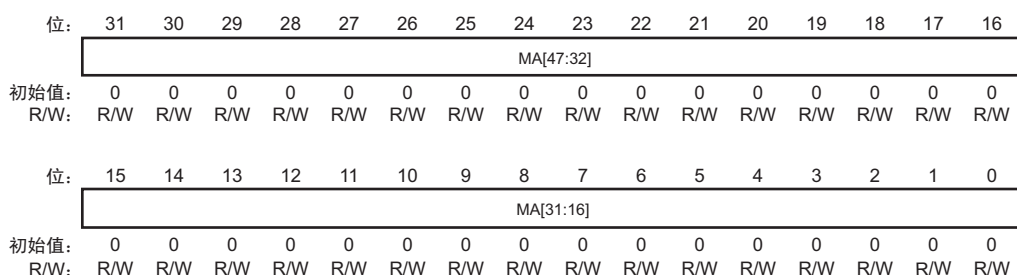
PIR 是 32 位可读写寄存器，提供经由 MII 存取 PHY-LSI 内部寄存器的手段。

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
	—	—	—	—	—	—	—	—	—	—	—	—	MDI	MDO	MMD	MDC
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/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 4	—	全 0	R	保留位 读取值为 0，只能写 0。
3	MDI	不定值	R	MII 管理数据输入位 表示 MDIO 引脚电平。
2	MDO	0	R/W	MII 管理数据输出位 当 MMD 位为 1 时，从 MDIO 引脚输出此位的设定值。
1	MMD	0	R/W	MII 管理模式位 指定 MII 的数据读写方向。 0: 指定读方向 1: 指定写方向
0	MDC	0	R/W	MII 管理数据时钟位 从 MDC 引脚输出此位的设定值，给 MII 提供管理数据时钟。有关 MII 寄存器的存取方法，请参照“19.4.4 MII 寄存器的存取方法”。

19.3.5 MAC 地址高位设定寄存器 (MAHR)

MAHR 是 32 位可读写寄存器，设定 48 位 MAC 地址的高 32 位。通常，在复位后的初始设定时设定此寄存器。在发送和接收功能有效的状态下，禁止改写 MAC 地址的设定。必须在通过 EDMR 的 SWR 位将 EtherC 和 E-DMAC 恢复到初始状态后，重新设定此寄存器。



位	位名	初始值	R/W	说明
31 ~ 0	MA[47:16]	全 0	R/W	MAC 地址位 设定 MAC 地址的高 32 位。 当 MAC 地址为 01-23-45-67-89-AB (16 进制表示) 时，将此寄存器置 H'01234567。

19.3.6 MAC 地址低位设定寄存器 (MALR)

MALR 是 32 位可读写寄存器，设定 48 位 MAC 地址的低 16 位。通常，在复位后的初始设定时设定此寄存器。在发送和接收功能有效的状态下，禁止改写 MAC 地址的设定。必须在通过 EDMR 的 SWR 位将 EtherC 和 E-DMAC 恢复到初始状态后，重新设定此寄存器。

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

位	位名	初始值	R/W	说明
31 ~ 16	—	全 0	R	保留位 读取值为 0，只能写 0。
15 ~ 0	MA[15:0]	全 0	R/W	MAC 地址位 15 ~ 0 设定 MAC 地址的低 16 位。 当 MAC 地址为 01-23-45-67-89-AB（16 进制表示）时，将此寄存器置 H'89AB。

19.3.7 接收帧长度上限寄存器 (RFLR)

RFLR 是 32 位可读写寄存器，以字节为单位指定本 LSI 能接收的最大帧长度。在接收功能有效的状态下，禁止改写此寄存器。

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

位	位名	初始值	R/W	说明
31 ~ 12	—	全 0	R	保留位 读取值为 0，只能写 0。
11 ~ 0	RFL[11:0]	全 0	R/W	接收帧数据长度 11 ~ 0 这里的帧数据是包含从目的地址到 CRC 数据的范围，实际上从目的地址到数据的全部信息传送到存储器，不包含 CRC 数据。当接收超过此位指定值的数据时，舍弃超过设定值部分的数据。 H'000 ~ H'5EE: 1,518 字节 H'5EF: 1,519 字节 H'5F0: 1,520 字节 : : H'7FF: 2,047 字节 H'800 ~ H'FFF: 2,048 字节

19.3.8 PHY 部状态寄存器 (PSR)

PSR 是只读寄存器，能读取 PHY-LSI 的接口信号。

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LMON
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
31 ~ 1	—	全 0	R	保留位 读取值为 0，只能写 0。
0	LMON	不定值	R	LNKSTA 引脚状态位 通过将 PHY-LSI 输出的 Link 信号连接到 LNKSTA 引脚，能读取 Link 状态。有关极性，请参照连接的 PHY-LSI 规格。

19.3.9 试发送失败计数寄存器 (TROCR)

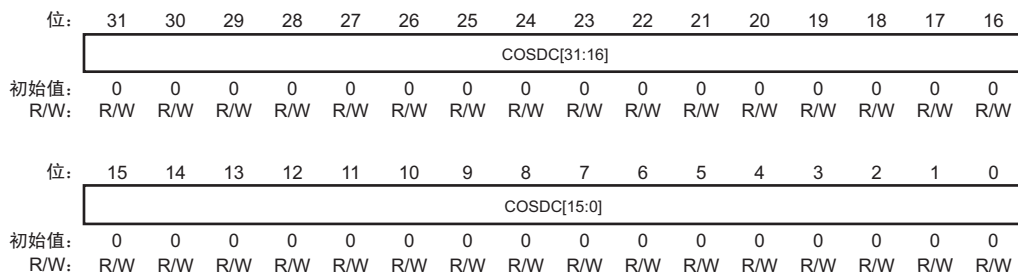
TROCR 是 32 位计数器，表示在发送时（包括重新发送，试发 16 次）无法发送的帧数。当 16 次发送失败时，此寄存器加 1。在此寄存器的值为 H'FFFFFFFF 时，停止递增计数。通过写此寄存器，将计数器的值清 0。能写任意值。

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

位	位名	初始值	R/W	说明
31 ~ 0	TROCR[31:0]	全 0	R/W	试发送失败计数 表示在发送时（包括重新发送，试送 16 次）无法发送的帧数。

19.3.10 延迟冲突检测计数寄存器 (CDCR)

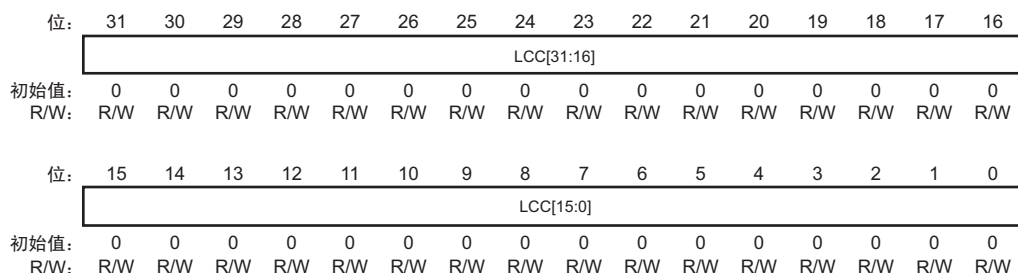
CDCR 是 32 位计数器，表示在开始发送后的全部线路上的延迟冲突次数。当此寄存器的值为 H'FFFFFFFF 时，停止递增计数。通过写此寄存器，将计数器的值清 0。能写任意值。



位	位名	初始值	R/W	说明
31 ~ 0	COSDC[31:0]	全 0	R/W	延迟冲突检测计数 表示开始发送后的全部延迟冲突次数。

19.3.11 载波消失计数寄存器 (LCCR)

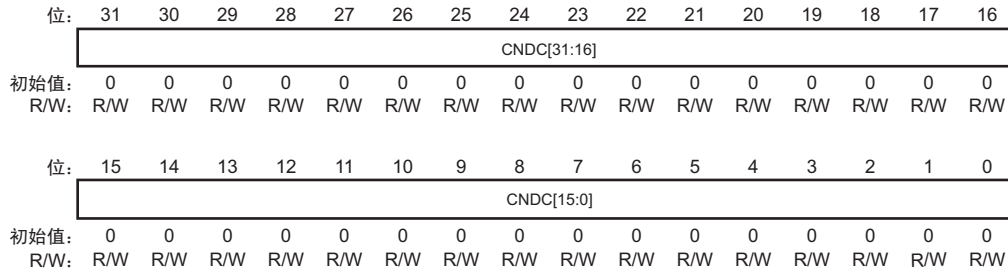
LCCR 是 32 位计数器，表示在发送数据过程中载波消失的次数。当此寄存器的值为 H'FFFFFFFF 时，停止递增计数。通过写此寄存器，将计数器的值清 0。能写任意值。



位	位名	初始值	R/W	说明
31 ~ 0	LCC[31:0]	全 0	R/W	载波消失计数 表示在发送数据过程中载波消失的次数。

19.3.12 载波未检测计数寄存器 (CNDCR)

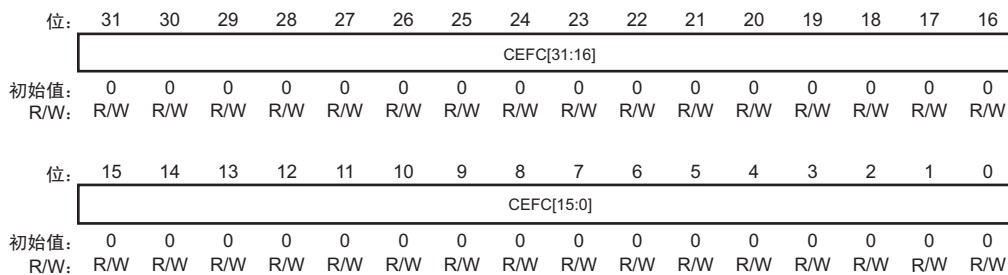
CNDCR 是 32 位计数器，表示在发送前同步码中无法检测载波的次数。当此寄存器的值为 H'FFFFFFFF 时，停止递增计数。通过写此寄存器，将计数器的值清 0。能写任意值。



位	位名	初始值	R/W	说明
31 ~ 0	CNDC[31:0]	全 0	R/W	载波未检测计数 表示未检测到载波的次数。

19.3.13 CRC 错误帧接收计数寄存器 (CEFCR)

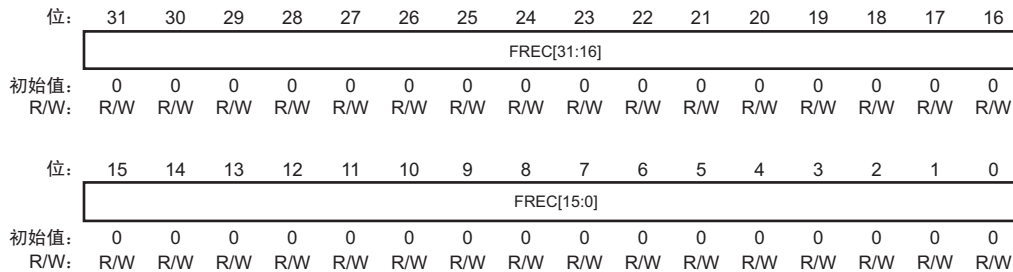
CEFCR 是 32 位计数器，表示发生 CRC 错误的帧的接收次数。当此寄存器的值为 H'FFFFFFFF 时，停止递增计数。通过写此寄存器，将计数器的值清 0。能写任意值。



位	位名	初始值	R/W	说明
31 ~ 0	CEFC[31:0]	全 0	R/W	CRC 错误帧计数 表示发生 CRC 错误的帧的接收次数。

19.3.14 帧接收错误计数寄存器 (FRECR)

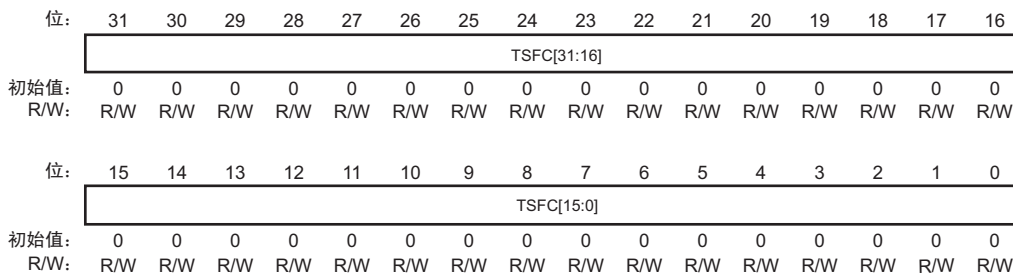
FRECR 是 32 位计数器，表示通过 RX-ER 引脚从 PHY-LSI 输入的错误帧的接收次数。每当 RX-ER 引脚有效时计数器就加 1。当此寄存器的值为 H'FFFFFFFF 时，停止递增计数。通过写此寄存器，将计数器的值清 0。能写任意值。



位	位名	初始值	R/W	说明
31 ~ 0	FRECR[31:0]	全 0	R/W	帧接收错误计数 表示在帧接收过程中发生错误的次数。

19.3.15 64 字节未帧接收计数寄存器 (TSFRCR)

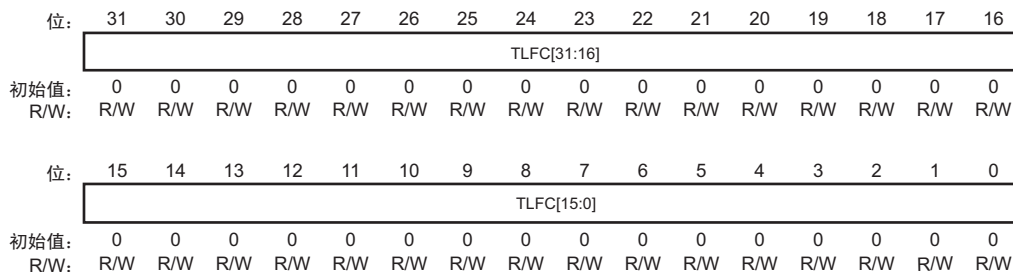
TSFRCR 是 32 位计数器，表示接收到不满 64 字节的帧。当此寄存器的值为 H'FFFFFFFF 时，停止递增计数。通过写此寄存器，将计数器的值清 0。能写任意值。



位	位名	初始值	R/W	说明
31 ~ 0	TSFRCR[31:0]	全 0	R/W	64 字节未帧接收计数 表示接收到未 64 字节的帧的次数。

19.3.16 指定字节帧超过接收计数寄存器 (TLFRCR)

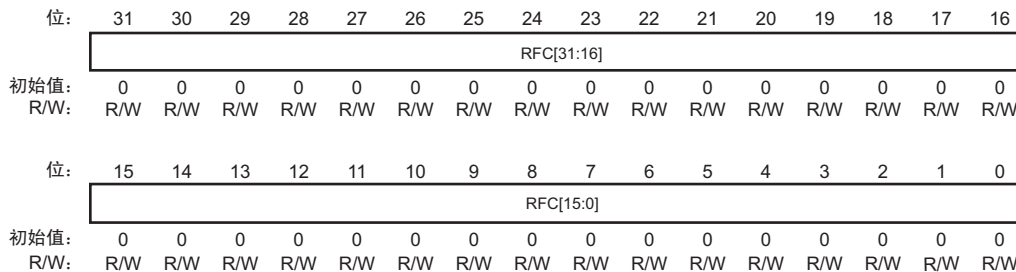
TLFRCR 是 32 位寄存器，表示接收到超过接收帧长度上限寄存器 (RFLR) 所指定值的帧。当此寄存器的值为 H'FFFFFFFF 时，停止递增计数。当接收的帧包含剩余位时，此寄存器不进行递增计数。此时，接收的帧反映在剩余位帧接收计数器寄存器 (RFCR)。通过写此寄存器，将计数器的值清 0。能写任意值。



位	位名	初始值	R/W	说明
31 ~ 0	TLFC[31:0]	全 0	R/W	指定字节超过帧接收计数 表示接收到超过 RFLR 值的帧的次数。

19.3.17 剩余位帧接收计数寄存器 (RFCR)

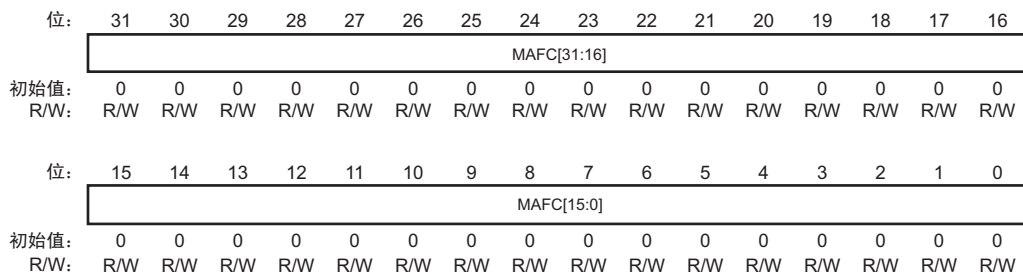
RFCR 是 32 位计数器，表示接收到包含不满 8 位剩余位数据的帧。当此寄存器的值为 H'FFFFFFFF 时，停止递增计数。通过写此寄存器，将计数器的值清 0。能写任意值。



位	位名	初始值	R/W	说明
31 ~ 0	RFC[31:0]	全 0	R/W	剩余位帧接收计数 表示接收到包含剩余位数据的帧的次数。

19.3.18 多播地址帧接收计数寄存器 (MAFCR)

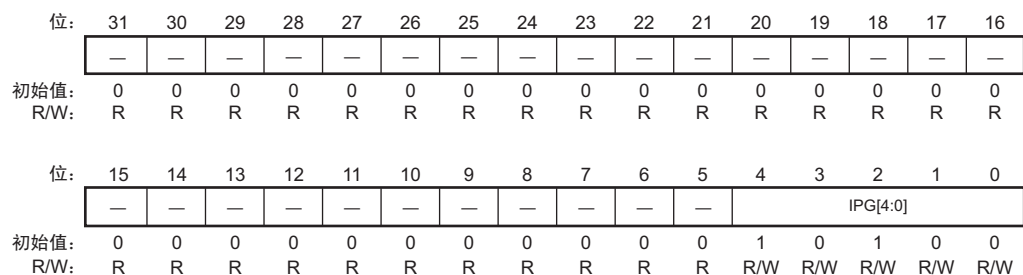
MAFCR 是 32 位计数器，表示接收到指定多播地址的帧。当此寄存器的值为 H'FFFFFFFF 时，停止递增计数。通过写此寄存器，将计数器的值清 0。能写任意值。



位	位名	初始值	R/W	说明
31 ~ 0	MAFC[31:0]	全 0	R/W	多播地址帧计数 表示接收到多播地址帧的次数。

19.3.19 IPG 设定寄存器 (IPGR)

IPGR 是设定 IPG (InterPacketGap) 值的寄存器。在 EtherC 模式寄存器 (ECMR) 的发送 / 接收功能有效的状态下，禁止改写此寄存器的设定 (详细内容请参照 “19.4.6 根据 IPG 设定的运行”)。



位	位名	初始值	R/W	说明
31 ~ 5	—	全 0	R	保留位 读取值为 0，只能写 0。
4 ~ 0	IPG[4:0]	H'14	R/W	Inter Packet Gap 每 4 位时间设定 IPG 值。 H'00: 16 位时间 H'01: 20 位时间 : : H'14: 96 位时间 (初始值) : : H'1F: 140 位时间

19.3.20 自动 PAUSE 帧设定寄存器 (APR)

APR 设定自动 PAUSE 帧的 TIME 参数值。

在发送自动 PAUSE 帧时，将此寄存器设定的值用作 PAUSE 帧的 TIME 参数。

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

位	位名	初始值	R/W	说明
31 ~ 16	—	全 0	R	保留位 读取值为 0，只能写 0。
15 ~ 0	AP[15:0]	全 0	R/W	自动 PAUSE 设定自动 PAUSE 帧的 TIME 参数值。此时，1 位表示 512 位时间。

19.3.21 手动 PAUSE 帧设定寄存器 (MPR)

MPR 设定手动 PAUSE 帧的 TIME 参数值。

在发送手动 PAUSE 帧时，将此寄存器设定的值用作 PAUSE 帧的 TIME 参数。

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

位	位名	初始值	R/W	说明
31 ~ 16	—	全 0	R	保留位 读取值为 0，只能写 0。
15 ~ 0	MP[15:0]	全 0	R/W	手动 PAUSE 设定手动 PAUSE 帧的 TIME 参数值。此时，1 位表示 512 位时间。读取取值为不定值。

19.3.22 自动 PAUSE 帧再发送次数设定寄存器 (TPAUSER)

TPAUSER 设定自动 PAUSE 帧的重新发送次数的上限值。在发送功能有效的状态下，禁止改写此寄存器的设定。

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

位	位名	初始值	R/W	说明
31 ~ 16	—	全 0	R	保留位 读取值为 0，只能写 0。
15 ~ 0	TPAUSE[15:0]	全 0	R/W	自动 PAUSE 帧再发送次数的上限值 H'0000: 不限制重新发送次数 H'0001: 重新发送次数为 1 次 : : H'FFFF: 重新发送次数为 65535 次

19.3.23 随机数生成计数器上限值设定 (RDMLR)

RDMLR 能设定随机数生成部使用的计数器的上限值。

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

位	位名	初始值	R/W	说明
31 ~ 20	—	全 0	R	保留位 读取值为 0，只能写 0。
19 ~ 0	RMD[19:0]	全 0	R/W	随机数生成部使用的计数器的上限值 H'00000: 正常运行的设定值。 H'00001 ~ H'FFFFE: 计数器的上限值为此设定值。

【注】 此寄存器会使 feLic 随机数生成部的运行发生变化，所以在设定非 0 值时，必须注意。

19.3.24 接收 PAUSE 帧计数器 (RFCF)

RFCF 是 PAUSE 帧的接收计数器。

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

位	位名	初始值	R/W	说明
31 ~ 8	—	全 0	R	保留位 读取值为 0，只能写 0。
7 ~ 0	RPAUSE[7:0]	全 0	R	PAUSE 帧的接收次数

19.3.25 PAUSE 帧再发送次数计数器 (TPAUSECR)

TPAUSECR 是表示 PAUSE 帧重新发送次数的计数器。

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

位	位名	初始值	R/W	说明
31 ~ 8	—	全 0	R	保留位 读取值为 0，只能写 0。
7 ~ 0	TXP[7:0]	全 0	R	PAUSE 帧的重新发送次数

19.3.26 Broadcast 帧接收次数设定寄存器 (BCFRR)

BCFRR 是设定 Broadcast 帧的连续接收次数的寄存器。

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

位	位名	初始值	R/W	说明
31 ~ 16	—	全 0	R	保留位 读取值为 0，只能写 0。
15 ~ 0	BCF[15:0]	全 0	R/W	Broadcast 帧的连续接收次数设定 DA 能进行设定次数值的广播地址帧的接收。当接收的次数超过设定值时，放弃超过部分的广播帧。 H'0000：不限制接收次数 H'0001：接收 1 个帧 ： ： H'FFFF：接收 65535 个帧

19.4 运行说明

以太网控制器 (EtherC) 运行的概要如下所示。

以太网控制器 (EtherC) 支持符合 IEEE802.3x 的控制, 能接收和发送使用的 Pause 帧。

19.4.1 发送运行

当从发送 E-DMAC 产生发送请求时, EtherC 发送部将发送数据组合成帧输出到 MII。经由 MII 的发送数据被 PHY-LSI 发送到线路上。EtherC 发送部的状态转移图如图 19.2 所示。

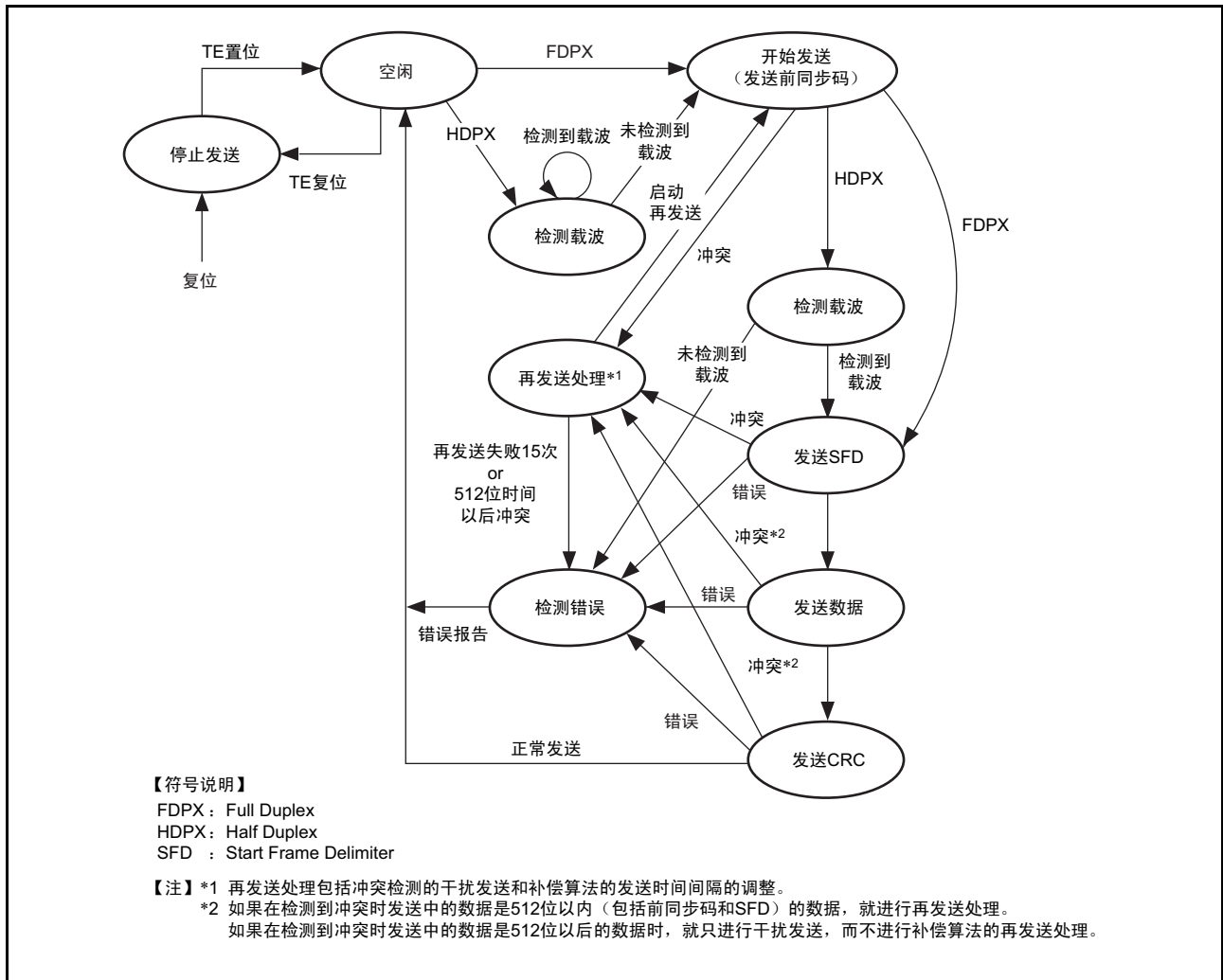


图 19.2 EtherC 发送部的状态转移图

1. 如果将发送允许 (TE) 位置位, 就转移到发送空闲状态。
2. 如果从发送 E-DMAC 产生发送请求, EtherC 就检测载波, 经过帧间隔时间的发送延时, 将前同步码发送到 MII。在选择不需要检测载波的全双工传送模式时, 如果从发送 E-DMAC 产生发送请求, 就立即发送前同步码。
3. 依次发送 SFD、数据和 CRC。在发送结束时, 就从发送 E-DMAC 产生发送结束中断 (TC)。如果在数据发送过程中发生冲突或者未检测到载波, 就将其作为中断源报告。
4. 在经过帧间隔时间后, 转移到空闲状态, 如果还有发送数据, 就继续发送。

19.4.2 接收运行

EtherC 接收部将从 MII 输入的帧分解为前同步码、SFD、数据和 CRC 数据，并且将从 DA（目的地址）到 CRC 数据的所有信息输出到接收 E-DMAC。EtherC 接收部的状态转移图如图 19.3 所示。

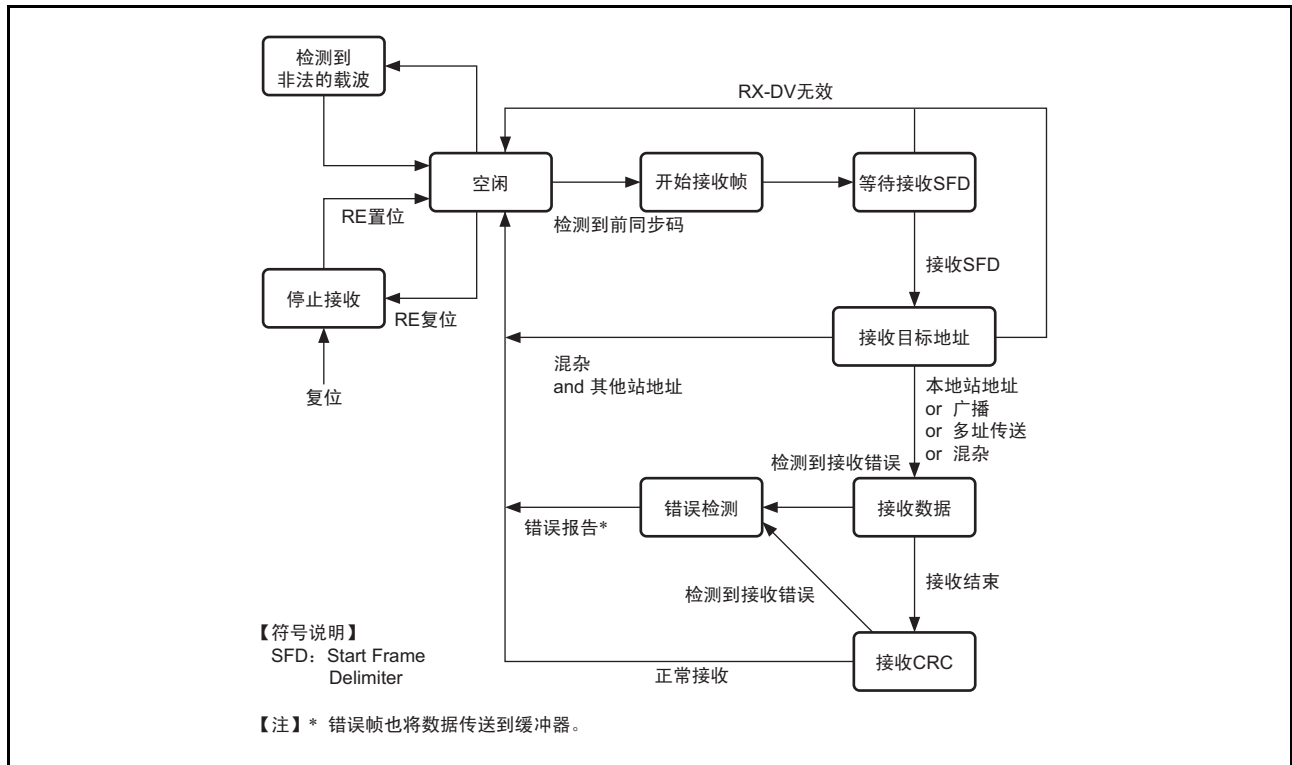


图 19.3 EtherC 接收部的状态转移图

1. 如果将接收允许（RE）位置位，就转移到接收空闲状态。
2. 如果检测到接收包中的前同步码之后的SFD（起始帧分隔符），就开始接收处理。如果是非法帧时，就放弃该帧。
3. 在正常模式中，当帧的目的地址是本LSI、广播帧或者多播帧时，就开始接收数据。在混杂模式中，与帧的种类无关，开始接收数据。
4. 在接收MI1的数据后，进行帧数据部的CRC检查。结果是在将帧数据写到存储器后将状态反映到描述符内。在发生异常时，报告错误状态。
5. 在接收1个帧后，如果EtherC模式寄存器内的接收允许位被置位（RE=1），就准备接收下一个帧。

19.4.3 MII 帧时序

各种 MII 帧的时序如图 19.4 所示

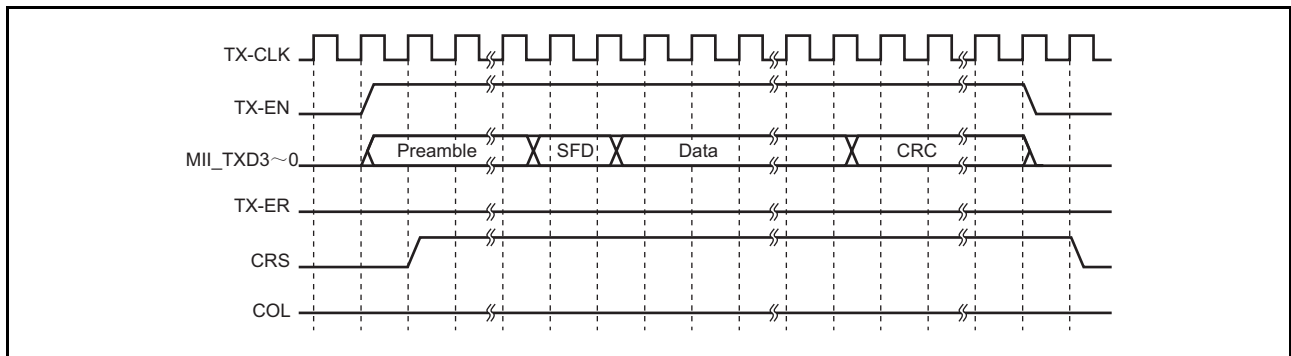


图 19.4 (1) MII 帧的发送时序 (正常发送)

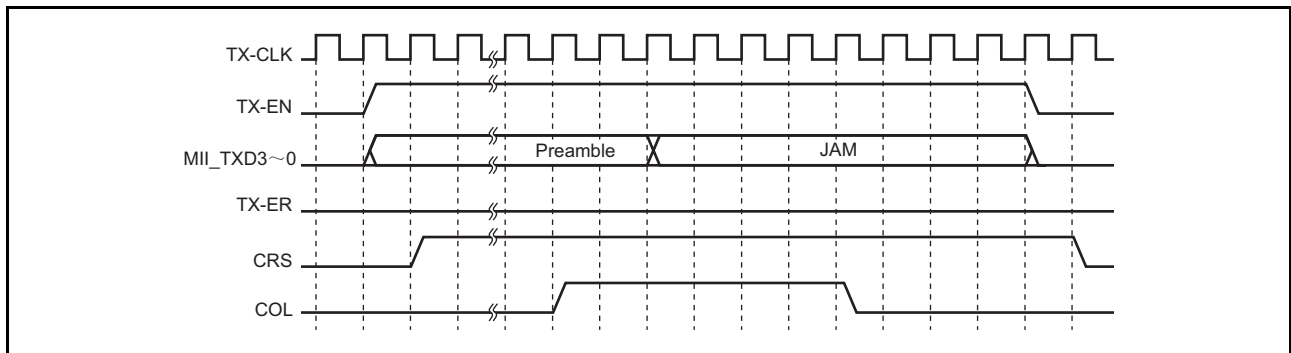


图 19.4 (2) MII 帧的发送时序 (发生冲突)

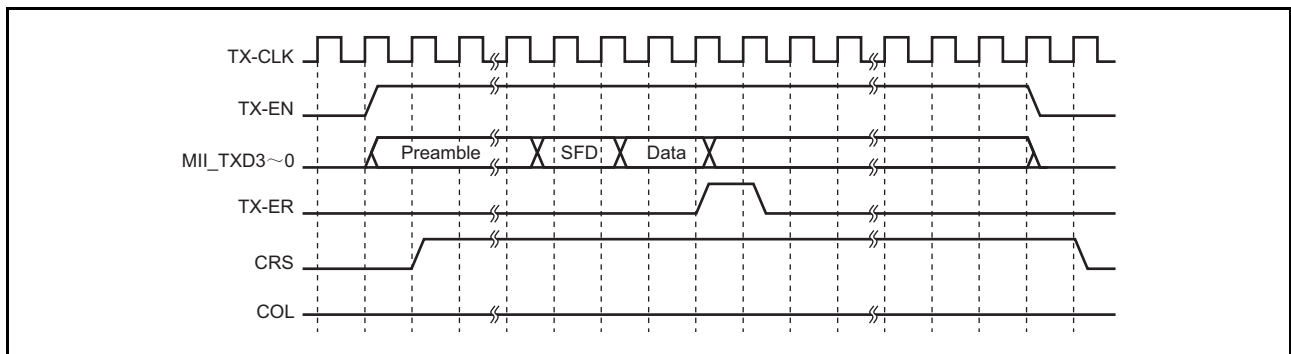


图 19.4 (3) MII 帧的发送时序 (出现发送错误)

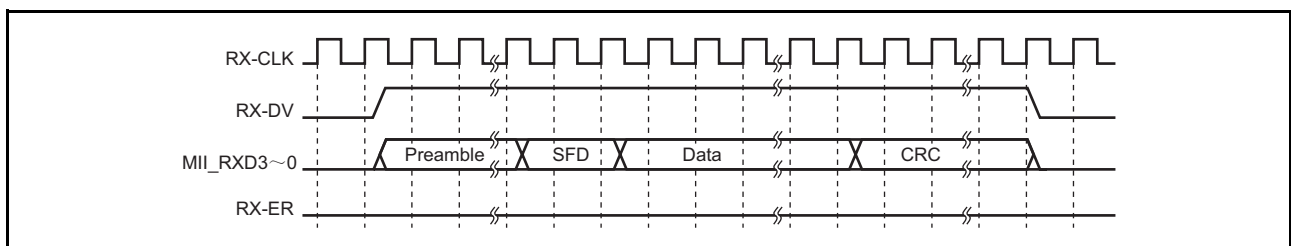


图 19.4 (4) MII 帧的接收时序 (正常接收)

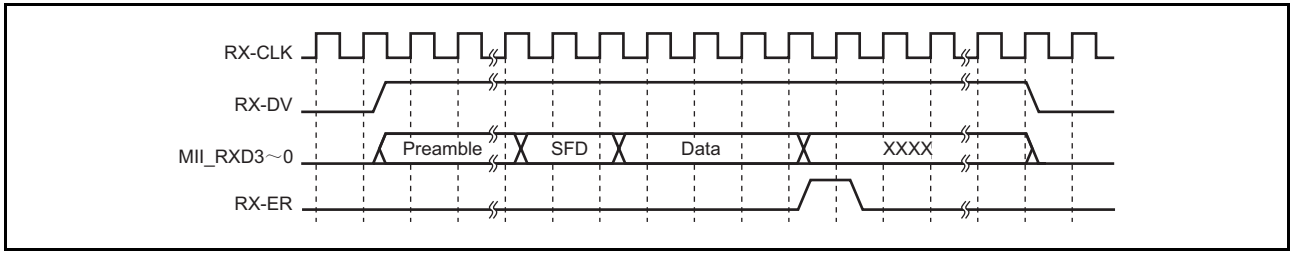


图 19.4 (5) MII 帧的接收时序 (接收错误 (1) 通知接收错误)

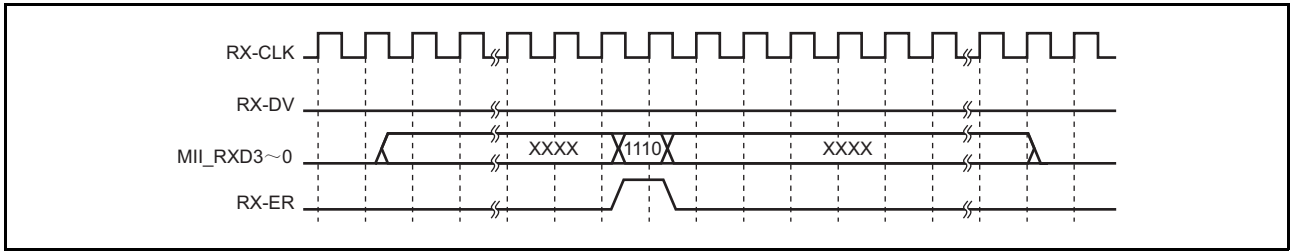


图 19.4 (6) MII 帧的接收时序 (接收错误 (2) 通知载波错误)

19.4.4 MII 寄存器的存取方法

经由本 LSI 的 PHY 部接口寄存器 (PIR) 对 PHY-LSI 内的 MII 寄存器进行存取。按照 IEEE802.3u 规定的 MII 帧格式, 作为串行接口进行连接。

(1) MII 管理帧的格式

MII 管理帧的格式如图 19.5 所示。在存取 MII 寄存器时, 按照 (2) 所示的步骤通过程序来实现管理帧。



图 19.5 MII 管理帧的格式

(2) MII 寄存器的存取步骤

程序经由 PHY 部接口寄存器 (PIR) 对 MII 寄存器进行存取。通过 1 位为单位的写数据、1 位为单位的读数据、总线释放和单独总线释放的组合来实现存取。

MII 寄存器的存取时序例子如图 19.6 所示, 存取时序因 PHY-LSI 种类而不同。

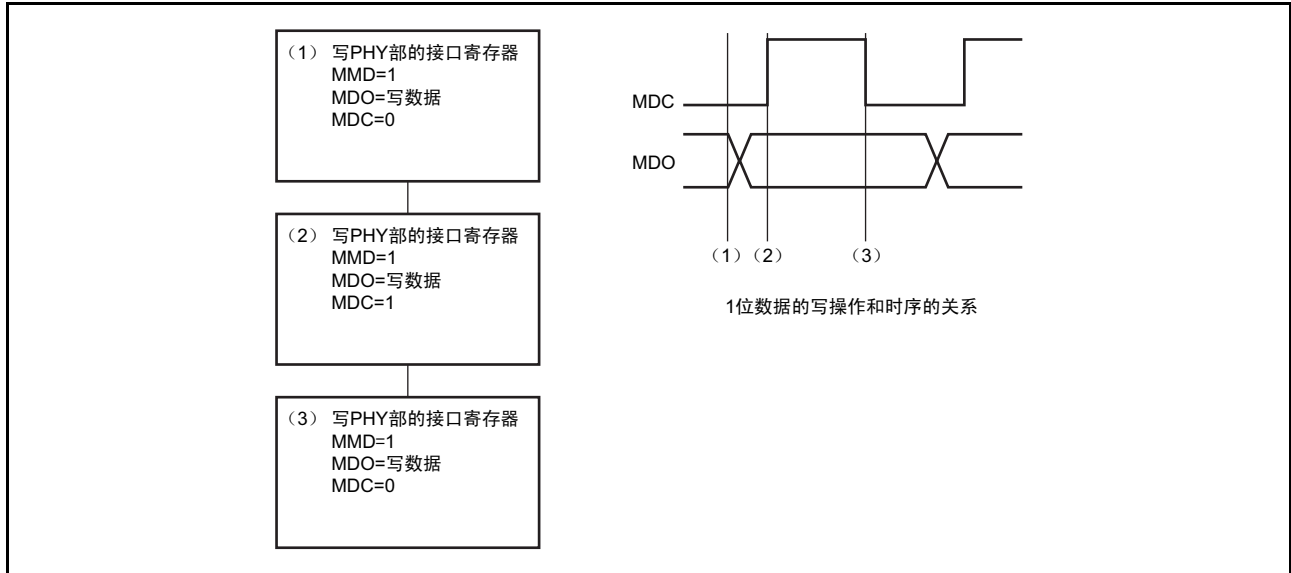


图 19.6 (1) 1 位数据的写流程

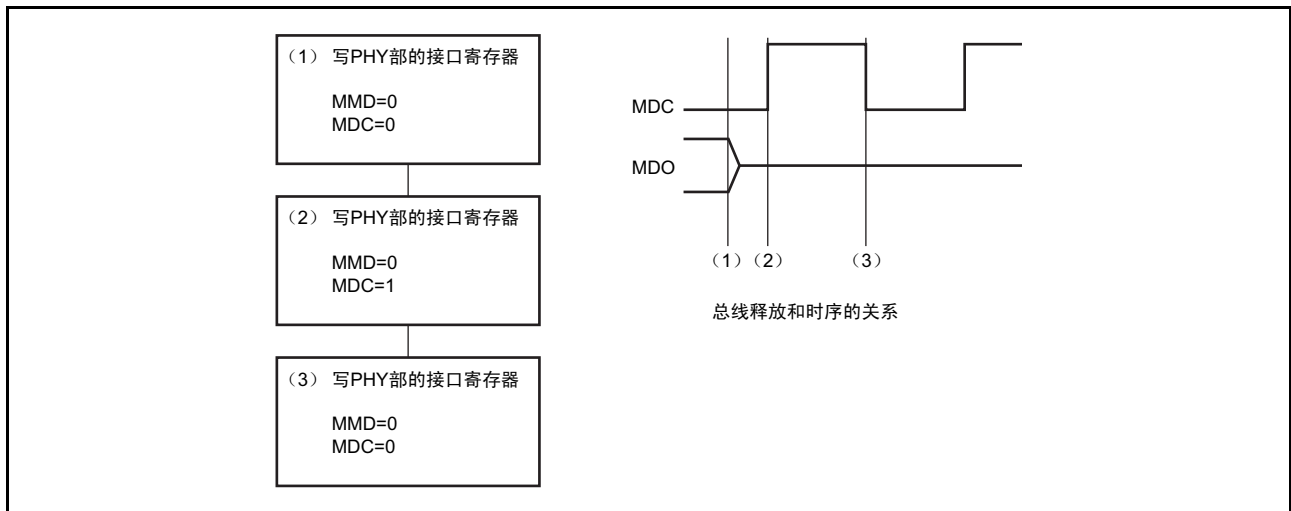


图 19.6 (2) 总线的释放流程 (图 19.5 中读操作的 TA)

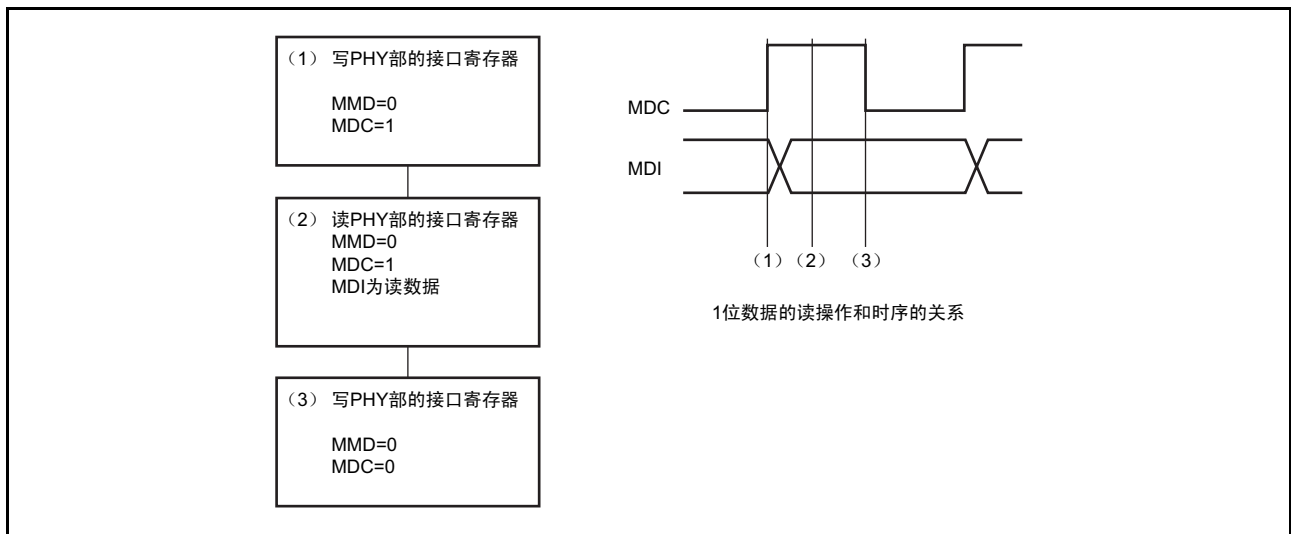


图 19.6 (3) 1 位数据的读流程

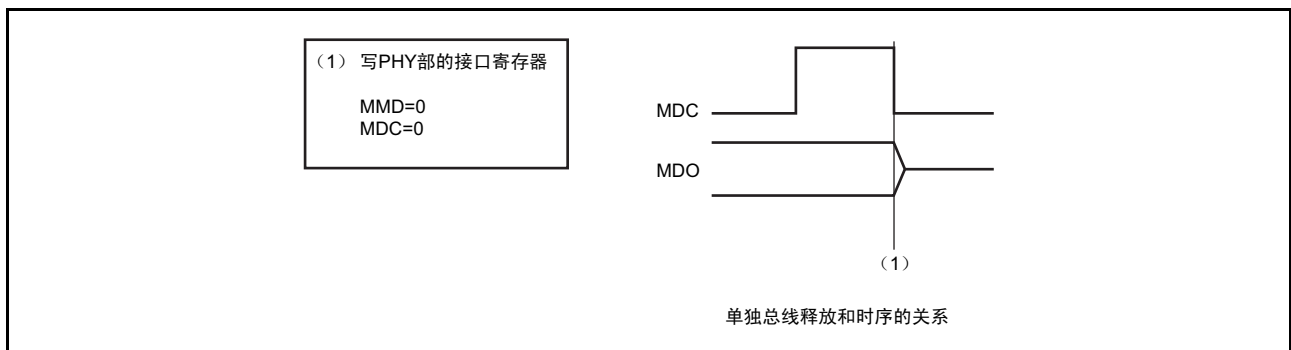


图 19.6 (4) 单独总线的释放流程 (图 19.5 中写操作的 IDLE)

19.4.5 Magic Packet 的检测

EtherC 具有 Magic Packet 的检测功能, 启动从主设备连接到 LAN 的各种外围设备 (WOL: Wake-On-LAN)。因此, 能构筑外围设备接收主设备等发送的 Magic Packet 并且自动启动外围设备的系统。当检测到 Magic Packet 时, 由此前接收的广播包等将数据积累在接收 FIFO, 并且将接收状态报告 EtherC。为了从此中断处理返回到正常运行, 必须通过 E-DMAC 模式寄存器 (EDMR) 的 SWR 位, 进行 EtherC 和 E-DMAC 的初始化。

Magic Packet 与目的地址无关, 进行数据接收。WOL 引脚只有在与 Magic Packet 内的格式所指定地址一致时才有效。有关 Magic Packet 的详细内容, 请参照 AMD 公司的技术资料。

在通过本 LSI 使用 WOL 时, 必须按照以下顺序进行设定:

1. 通过各种中断允许/屏蔽寄存器, 禁止中断源的输出。
2. 设定 EtherC 模式寄存器 (ECMR) 的 Magic Packet 检测允许位 (MPDE)。
3. 将 EtherC 中断允许寄存器 (ECSIPR) 的 Magic Packet 检测中断允许位 (MPDIP) 设定为允许状态。
4. 必要时, 将 CPU 的运行模式设定为睡眠模式或者将外围功能设定为模块待机模式。
5. 如果检测到 Magic Packet, 就通知 CPU, 并且通过 WOL 引脚通知外围 LSI。

19.4.6 根据 IPG 设定的运行

EtherC 具有更改发送帧之间的无发送期间 IPG (Inter Packet Gap) 的功能。通过更改 IPG 设定寄存器 (IPGR) 的设定值, 能提高或者降低传送效率 (与标准值相比)。IEEE802.3 标准规定了 IPG 的设定。在更改设定时, 必须通过同一个网络充分确认各设备是否正常工作。

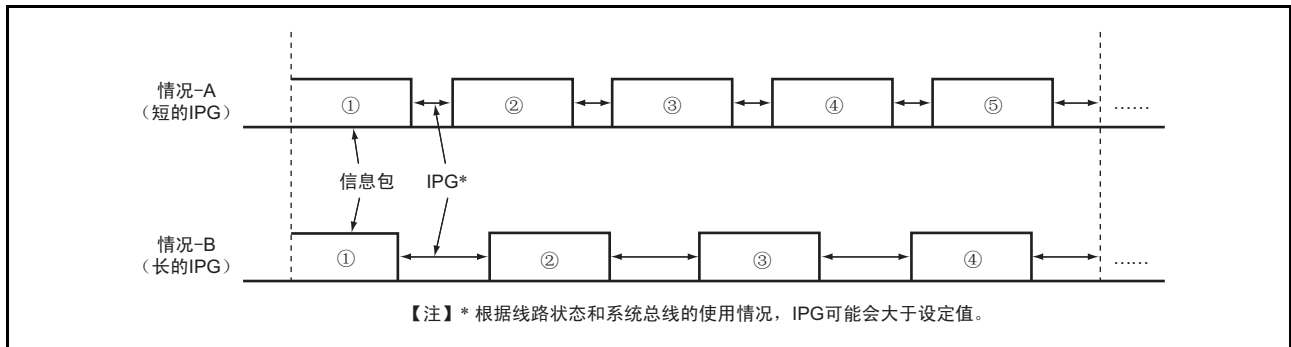


图 19.7 根据 IPG 的变更产生传送效率的不同

19.4.7 流控制

在全双工模式运行时, EtherC 支持符合 IEEE802.3x 的流控制功能。流控制适用于接收和发送运行。在进行流控制时, PAUSE 帧的发送步骤如下:

(1) 自动 PAUSE 帧的发送

对于接收帧, 如果写在接收 FIFO (内置在 E-DMAC) 中的数据量达到 E-DMAC 内部的流控制开始 FIFO 阈值设定寄存器 (FCFTR) 的设定值, 就自动发送 PAUSE 帧。此时, 包含在 PAUSE 帧中的 TIME 参数通过自动 PAUSE 帧设定寄存器 (APR) 进行设定。读接收 FIFO 内的数据, 在数据量未达到 FCFTR 的设定值之前重复发送自动 PAUSE 帧。另外, 也可通过自动 PAUSE 帧再发送次数设定寄存器 (TPAUSER) 设定 PAUSE 帧的重新发送次数的上限值。在这种情况下, 如果接收 FIFO 内的数据量未达到 FCFTR 的设定值或者发送次数未达到 TPAUSER 的设定值, 就重复发送 PAUSE 帧。在 EtherC 模式寄存器 (ECMR) 的 TXF 位为 1 时, 自动 PAUSE 帧的发送有效。

(2) 手动 PAUSE 帧的发送

通过软件的指示, 发送 PAUSE 帧。当将 Timer 值写到手动 PAUSE 帧设定寄存器 (MPR) 时, 就开始发送手动 PAUSE 帧。根据此步骤, PAUSE 帧只发送 1 次。

(3) PAUSE 帧的接收

在接收 PAUSE 帧后, 需要等待 Timer 值所示的时间, 然后发送下一个帧。但是, 正在发送中的帧将继续发送。在 EtherC 模式寄存器 (ECMR) 的 RXF 位为 1 时, PAUSE 帧的接收有效。

19.5 与 PHY-LSI 的连接

与 National Semiconductor Corporation 的 DP83846AVHG 的连接例子如图 19.8 所示。

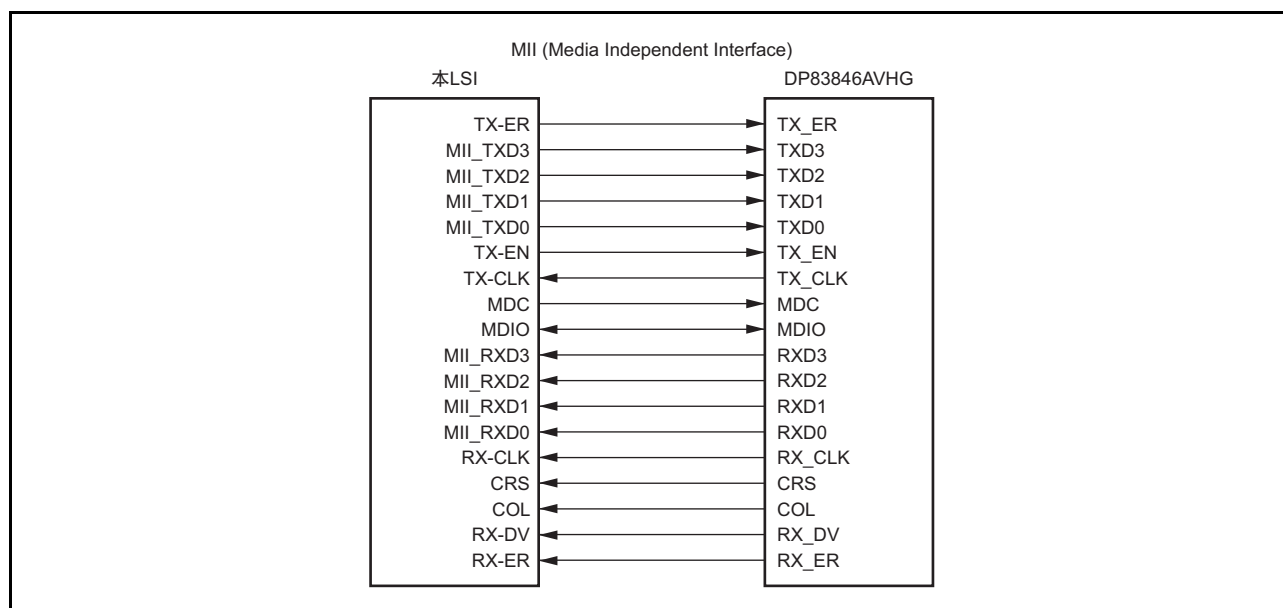


图 19.8 与 DP83846AVHG 的连接例子

19.6 使用时的注意事项

在使用 EtherC 时，必须注意以下事项。

(1) 有关 LCHNG 位的置位条件

即使在 LNKSTA 引脚的输入电平不发生变化的情况下，ECSR 的 LCHNG 位也可能被置位。这是在 GPIO 的 PSEL 选择 LNKSTA 引脚或者通过 EDMR 的 SWR 位解除 EtherC/E-DMAC 的软件复位时 LNKSTA 引脚被输入高电平的情况。

这是因为，在 GPIO 未选择 LNKSTA 引脚或者 EtherC/E-DMAC 处于软件复位中，LSI 内部的 LNKSTA 信号与外部引脚的输入电平无关，被固定为低电平。

为了避免错误地发生连接信号变化中断，必须在清除 LCHNG 位后将 ECSIPR 的 LCHNGIP 位置位。

第 20 章 用于以太网控制器的直接存储器存取控制器（E-DMAC）

本 LSI 内置直接连接以太网控制器（EtherC）的直接存储器存取控制器（E-DMAC）。E-DMAC 使用描述符控制大部分的缓冲器管理，能减轻 CPU 的负担，高效率地控制数据的发送 / 接收。

E-DMAC、存储器中的描述符和发送 / 接收缓冲器的结构如图 20.1 所示。

20.1 特点

- 以描述符管理方式减轻 CPU 负担。
- 通过描述符反映发送 / 接收帧的状态。
- 通过块传送（32 字节单位）提高系统总线的使用效率。
- 能对应单帧和多缓冲方式。
- 通过接收数据的填充，提高软件的处理能力。

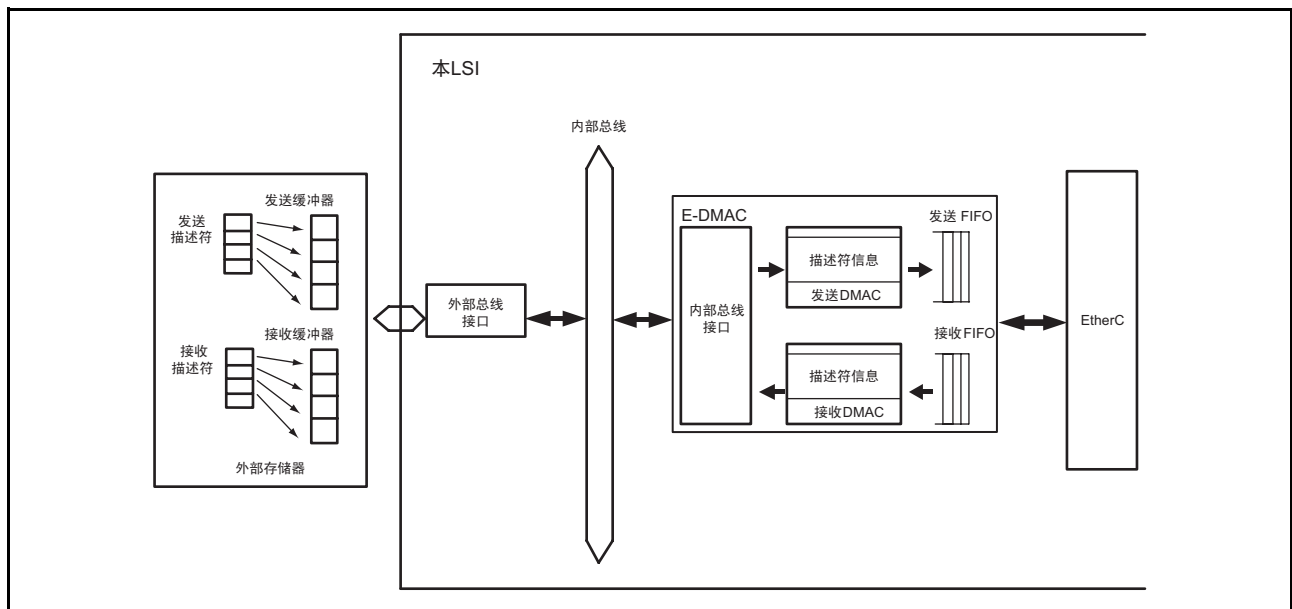


图 20.1 E-DMAC、描述符和缓冲器的结构

20.2 寄存器说明

E-DMAC 的寄存器结构如表 20.1 所示。

表 20.1 寄存器结构

名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度
E-DMAC 模式寄存器	EDMR	R/W	H'FEF0 0000	H'1EF0 0000	32
E-DMAC 发送请求寄存器	EDTRR	R/W	H'FEF0 0008	H'1EF0 0008	32
E-DMAC 接收请求寄存器	EDRRR	R/W	H'FEF0 0010	H'1EF0 0010	32
发送描述符列表起始地址寄存器	TDLAR	R/W	H'FEF0 0018	H'1EF0 0018	32
接收描述符列表起始地址寄存器	RDLAR	R/W	H'FEF0 0020	H'1EF0 0020	32
EtherC/E-DMAC 状态寄存器	EESR	R/W	H'FEF0 0028	H'1EF0 0028	32
EtherC/E-DMAC 状态中断允许寄存器	EESIPR	R/W	H'FEF0 0030	H'1EF0 0030	32
发送 / 接收状态复制指示寄存器	TRSCER	R/W	H'FEF0 0038	H'1EF0 0038	32
丢失帧计数寄存器	RMFCR	R	H'FEF0 0040	H'1EF0 0040	32
发送 FIFO 阈值指定寄存器	TFTR	R/W	H'FEF0 0048	H'1EF0 0048	32
FIFO 容量指定寄存器	FDR	R/W	H'FEF0 0050	H'1EF0 0050	32
接收方式控制寄存器	RMCR	R/W	H'FEF0 0058	H'1EF0 0058	32
发送 FIFO 下溢计数寄存器	TFUCR	R/W	H'FEF0 0064	H'1EF0 0064	32
接收 FIFO 上溢计数寄存器	RFOCR	R/W	H'FEF0 0068	H'1EF0 0068	32
接收缓冲器写地址寄存器	RBWAR	R	H'FEF0 00C8	H'1EF0 00C8	32
接收描述符取地址寄存器	RDFAR	R	H'FEF0 00CC	H'1EF0 00CC	32
发送缓冲器读地址寄存器	TBRAR	R	H'FEF0 00D4	H'1EF0 00D4	32
发送描述符取地址寄存器	TDFAR	R	H'FEF0 00D8	H'1EF0 00D8	32
流控制开始 FIFO 阈值设定寄存器	FCFTR	R/W	H'FEF0 0070	H'1EF0 0070	32
接收数据填充设定寄存器	RPADIR	R/W	H'FEF0 0078	H'1EF0 0078	32
发送中断设定寄存器	TRIMD	R/W	H'FEF0 007C	H'1EF0 007C	32
各输出信号设定寄存器	IOSR	R/W	H'FEF0 006C	H'1EF0 006C	32

【注】 * P4 区地址是使用虚拟地址空间 P4 区的地址，区域 7 地址是使用 TLB 存取物理地址空间区域 7 的地址。

各处理模式的 E-DMAC 寄存器状态如表 20.2 所示。

表 20.2 各处理模式的寄存器状态

名称	简称	软件复位时
E-DMAC 模式寄存器	EDMR	○
E-DMAC 发送请求寄存器	EDTRR	○
E-DMAC 接收请求寄存器	EDRRR	○
发送描述符列表起始地址寄存器	TDLAR	—
接收描述符列表起始地址寄存器	RDLAR	—
EtherC/E-DMAC 状态寄存器	EESR	○
EtherC/E-DMAC 状态中断允许寄存器	EESIPR	○
发送 / 接收状态复制指示寄存器	TRSCER	○
丢失帧计数寄存器	RMFCR	—
发送 FIFO 阈值指定寄存器	TFTR	○
FIFO 容量指定寄存器	FDR	○
接收方式控制寄存器	RMCR	○
发送 FIFO 下溢计数寄存器	TFUCR	—
接收 FIFO 上溢计数寄存器	RFOCR	—
接收缓冲器写地址寄存器	RBWAR	○
接收描述符取地址寄存器	RDFAR	○
发送缓冲器读地址寄存器	TBRAR	○
发送描述符取地址寄存器	TDFAR	○
流控制开始 FIFO 阈值设定寄存器	FCFTR	○
接收数据填充设定寄存器	RPADIR	○
发送中断设定寄存器	TRIMD	○
各输出信号设定寄存器	IOSR	○

【符号说明】

- ：被初始化的寄存器
- ：被保持的寄存器

20.2.1 E-DMAC 模式寄存器 (EDMR)

EDMR 是 32 位可读写寄存器，指定 E-DMAC 的运行模式。通常，在复位后的初始设定时设定此寄存器。当在发送数据的过程中通过此寄存器进行 EtherC 和 E-DMAC 的初始化时，就可能在线路上发送异常数据。在发送和接收功能有效的状态下，禁止改写运行模式的设定。在切换运行模式时，必须在通过软件复位位 (SWR) 将 EtherC 和 E-DMAC 恢复到初始状态后，重新设定此寄存器。完成 EtherC 和 E-DMAC 初始化所需要的时间为 64 个内部总线的时钟 Bφ 周期，所以必须在经过 64 个内部总线的时钟 Bφ 周期后才能存取 EtherC 和 E-DMAC 内的寄存器。

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

位	位名	初始值	R/W	说明
31 ~ 7	—	全 0	R	保留位 读取值为 0，只能写 0。
6	DE	0	R/W	大 / 小端法模式指定位 0: 大端法模式 (长字存取) (初始值) 1: 小端法模式 (长字存取) 发送 / 接收数据有效，发送 / 接收描述符和寄存器无效 (只支持大端法)。
5、4	DL[1:0]	00	R/W	发送 / 接收描述符长度指定位 00: 16 字节 (初始值) 01: 32 字节 10: 64 字节 当设定 11 时，为初始值的 16 字节。
3 ~ 1	—	全 0	R	保留位 读取值为 0，只能写 0。
0	SWR	0	R/W	软件复位 写时 0: 无效 1:

20.2.2 E-DMAC 发送请求寄存器 (EDTRR)

EDTRR 是 32 位可读写寄存器，给 E-DMAC 发布发送指示。在结束 1 帧的发送时，读下一个描述符。如果该描述符内的发送描述符有效位有效，就继续发送；如果发送描述符有效位无效，就清除 TR 位，并且停止发送 DMAC 的运行。

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TR
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
31 ~ 1	—	全 0	R	保留位 读取值为 0，只能写 0。
0	TR	0	R/W	发送请求位 0: 发送停止状态 即使写 0 也不停止发送。通过发送描述符内的有效位进行发送结束的控制。 1: 开始发送 读相应的描述符，发送有效位为 1 的帧。

20.2.3 E-DMAC 接收请求寄存器 (EDRRR)

EDRRR 是 32 位可读写寄存器，给 E-DMAC 发布接收指示。在设定接收请求位时，E-DMAC 读相应的接收描述符。如果描述符内的接收描述符有效位有效，就等待 EtherC 的接收请求，在接收完接收缓冲器分的数据时，E-DMAC 就读下一个描述符，等待帧的接收。此时，如果描述符内的接收描述符有效位无效，就清除 RR 位，并且停止接收 DMAC 的运行。

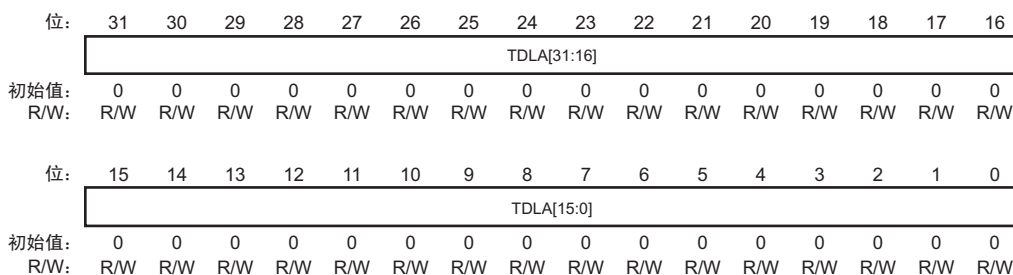
位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RR
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R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

位	位名	初始值	R/W	说明
31 ~ 1	—	全 0	R	保留位 读取值为 0，只能写 0。
0	RR	0	R/W	接收请求位 0: 接收功能无效 * 1: 读接收描述符，并且使 E-DMAC 处于可接收状态。

【注】 * 在接收帧的过程中，如果将接收功能置为无效，接收描述符的回写就无法正常进行，导致此后的接收描述符的读指针出现异常，使 E-DMAC 无法正常运行。此时，为了再次使 E-DMAC 处于可接收状态，必须通过 EDMR 的 SWR 位进行软件复位。在不进行 E-DMAC 的软件复位而使接收功能无效时，通过 ECMR 的 RE 位使接收功能无效，然后结束 E-DMAC 的接收，在确认接收描述符的回写后，必须将此寄存器的接收功能置为无效。

20.2.4 发送描述符列表起始地址寄存器 (TDLAR)

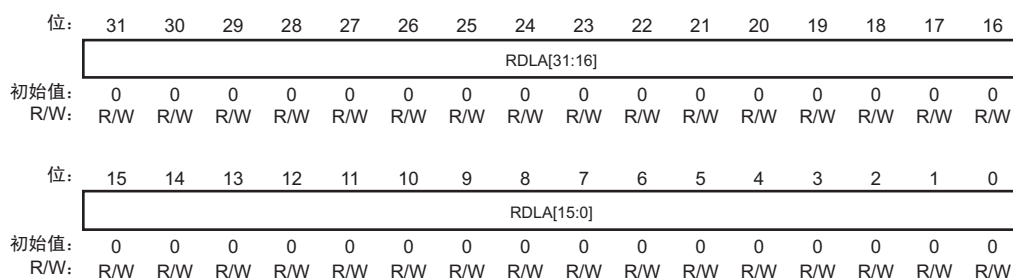
TDLAR 是 32 位可读写寄存器，设定发送描述符列表的起始地址。各描述符具有与 EDMR 的 DL 位表示的描述符长度一致的边界。在发送中，禁止改写此寄存器。必须在发送停止状态下，通过 E-DMAC 发送请求寄存器 (EDTRR) 的 TR 位 (=0) 改写此寄存器。



位	位名	初始值	R/W	说明
31 ~ 0	TDLA[31:0]	全 0	R/W	发送描述符的起始地址位 根据指定的描述符长度，对低位进行以下设定： 16 字节边界：TDLA[3:0]=0000 32 字节边界：TDLA[4:0]=00000 64 字节边界：TDLA[5:0]=000000

20.2.5 接收描述符列表起始地址寄存器 (RDLAR)

RDLAR 是 32 位可读写寄存器，设定接收描述符列表的起始地址。各描述符具有与 EDMR 的 DL 位表示的描述符长度一致的边界。在接收中，禁止改写此寄存器。必须在接收停止状态下，通过 E-DMAC 接收请求寄存器 (EDRRR) 的 RR 位 (=0) 改写此寄存器。



位	位名	初始值	R/W	说明
31 ~ 0	RDLA[31:0]	全 0	R/W	发送描述符的起始地址位 根据指定的描述符长度，对低位进行以下设定： 16 字节边界：RDLA[3:0]=0000 32 字节边界：RDLA[4:0]=00000 64 字节边界：RDLA[5:0]=000000

20.2.6 EtherC/E-DMAC 状态寄存器 (EESR)

EESR 是 32 位可读写寄存器，表示 EtherC 和 E-DMAC 的通信状态。以中断源的形式报告此寄存器的信息。通过写 1 清除各位（但是，bit22 (ECI) 是只读位，即使写 1 也不被清除）；写 0 不影响各位的状态。能通过 EtherC/E-DMAC 状态中断允许寄存器 (EESIPR) 的相应位屏蔽各中断源。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	TWB	—	—	—	TABT	RABT	RFCOF	ADE	ECI	TC	TDE	TFUF	FR	RDE	RFOF
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CND	DLC	CD	TRO	RMAF	—	—	RRF	RTLF	RTSF	PRE	CERF
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	R	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31	—	0	R	保留位 读取值为 0，只能写 0。
30	TWB	0	R/W	回写结束位 表示在帧发送结束后从 E-DMAC 对相应描述符的回写操作结束。该操作只在 TRIMD 的 TIS 位被置 1 时有效。 0: 回写未结束或者无发送指示 1: 回写结束
29 ~ 27	—	全 0	R	保留位 只能写 0。
26	TABT	0	R/W	发送中止检测位 表示在发送帧的过程中因障碍等 EtherC 中止帧的发送。 0: 未中止帧的发送或者未指示发送 1: 中止帧的发送
25	RABT	0	R/W	接收中止检测位 表示在接收帧的过程中因障碍等 EtherC 中止帧的接收。 0: 未中止帧的接收或者未指示接收 1: 中止帧的接收
24	RFCOF	0	R/W	接收帧计数器上溢位 表示接收 FIFO 内的帧计数器发生上溢。 0: 接收帧计数器未发生上溢 1: 接收帧计数器发生上溢
23	ADE	0	R/W	地址错误位 表示 E-DMAC 要传送的存储器地址不正确。 0: 未检测到非法的存储器地址（正常运行） 1: 检测到非法的存储器地址 【注】 在检测到地址错误时，E-DMAC 停止发送和接收。在需要重新开始时，必须在通过 EDMR 的 SWR 位进行软件复位后重新设定。

位	位名	初始值	R/W	说明
22	ECI	0	R	<p>EtherC 状态寄存器源位</p> <p>此位是只读位。当清除 EtherC 的 ECSR 源时，此位也被清除。</p> <p>0: 未检测到 EtherC 状态中断源</p> <p>1: 检测到 EtherC 状态中断源</p>
21	TC	0	R/W	<p>帧发送结束位</p> <p>表示由发送描述符指定的数据已全部从 EtherC 发送。在 1 帧 /1 个缓冲器处理的情况下，当结束 1 帧的发送时，就认为发送结束；在多缓冲器帧处理的情况下，当发送帧的最终数据并且下一个描述符内的发送描述符有效位（TACT）未被置位时，也被认为发送结束，此位变为 1。在帧的发送结束后，E-DMAC 将传送状态回写到该描述符。</p> <p>0: 传送未结束或者未指示传送</p> <p>1: 传送结束</p>
20	TDE	0	R/W	<p>发送描述符空位</p> <p>在多缓冲器帧处理中前一个描述符不为最终帧的情况下，表示在 E-DMAC 读发送描述符时描述符内的发送描述符有效位（TACT）未被置位。可能会发送不完全的帧。</p> <p>0: 检测到发送描述符有效位 TACT=1</p> <p>1: 检测到发送描述符有效位 TACT=0</p> <p>当发送描述符为空（TDE=1）时，必须在进行软件复位后启动发送。此时，从保存在发送描述符列表起始地址寄存器（TDLAR）中的地址开始。</p>
19	TFUF	0	R/W	<p>发送 FIFO 下溢位</p> <p>表示在发送帧的过程中发送 FIFO 发生下溢。在线路上发送不完全的数据。</p> <p>0: 未发生下溢</p> <p>1: 发生下溢</p>
18	FR	0	R/W	<p>帧接收位</p> <p>表示接收帧并且更新接收描述符。每接收 1 帧就将此位置 1。</p> <p>0: 未接收帧</p> <p>1: 已接收帧</p>
17	RDE	0	R/W	<p>接收描述符空位</p> <p>当接收描述符为空（RDE=1）时，在通过将该接收描述符设定为 RACT=1 后启动接收，能重新开始接收。</p> <p>0: 检测到接收描述符有效位 RACT=1</p> <p>1: 检测到接收描述符有效位 RACT=0</p>
16	RFOF	0	R/W	<p>接收 FIFO 上溢位</p> <p>表示在接收帧的过程中接收 FIFO 发生上溢。</p> <p>0: 未发生上溢</p> <p>1: 发生上溢</p>
15 ~ 12	—	全 0	R	<p>保留位</p> <p>只能写 0。</p>
11	CND	0	R/W	<p>载波未检测位</p> <p>表示载波的检测状态。</p> <p>0: 在发送开始时检测到载波</p> <p>1: 未检测到载波</p>

位	位名	初始值	R/W	说明
10	DLC	0	R/W	载波消失检测位 表示在发送帧的过程中检测到载波消失。 0: 未检测到载波消失 1: 检测到载波消失
9	CD	0	R/W	延迟冲突检测位 表示在发送帧的过程中检测到延迟冲突。 0: 未检测到延迟冲突 1: 检测到延迟冲突
8	TRO	0	R/W	试发送失败位 表示在发送帧的过程中发生试发送失败。这是在 EtherC 开始发送后，基于补偿算法，16 次试发送（含 15 次重新发送）都失败。 0: 未检测到试发送失败 1: 检测到试发送失败
7	RMAF	0	R/W	多播地址帧接收位 0: 未接收到多播地址帧 1: 接收到多播地址帧
6、5	—	全 0	R	保留位 只能写 0。
4	RRF	0	R/W	剩余位帧接收位 0: 未接收到剩余位帧 1: 接收到剩余位帧
3	RTLFL	0	R/W	长帧接收错误位 表示接收帧的字节数超过 EtherC 的 RFLR 设定的接收帧长度上限值。 0: 未接收到长帧 1: 接收到长帧
2	RTSF	0	R/W	短帧接收错误位 表示接收不满 64 字节的帧。 0: 未接收到短帧 1: 接收到短帧
1	PRE	0	R/W	PHY-LSI 接收错误位 0: 未检测到 PHY-LSI 接收错误 1: 检测到 PHY-LSI 接收错误
0	CERF	0	R/W	接收帧 CRC 错误位 0: 未检测到 CRC 错误 1: 检测到 CRC 错误

20.2.7 EtherC/E-DMAC 状态中断允许寄存器 (EESIPR)

EESIPR 是 32 位可读写寄存器，是对应 EtherC/E-DMAC 状态寄存器 (EESR) 各位的中断允许寄存器。通过将各位置 1 允许中断。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	TWB IP	—	—	—	TABT IP	RABT IP	RFCOF IP	ADE IP	ECI IP	TC IP	TDE IP	TFUF IP	FR IP	RDE IP	RFOF IP
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R	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
	—	—	—	—	CND IP	DLC IP	CD IP	TRO IP	RMAF IP	—	—	RRF IP	RTL IP	RTSF IP	PRE IP	CERF IP
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	R	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31	—	0	R	保留位 读取值为 0，只能写 0。
30	TWBIP	0	R/W	回写结束中断允许位 0: 禁止回写结束中断 1: 允许回写结束中断
29 ~ 27	—	全 0	R	保留位 只能写 0。
26	TABTIP	0	R/W	发送中止检测中断允许位 0: 禁止发送中止检测中断 1: 允许发送中止检测中断
25	RABTIP	0	R/W	接收中止检测中断允许位 0: 禁止接收中止检测中断 1: 允许接收中止检测中断
24	RFCOFIP	0	R/W	接收帧计数器上溢中断允许位 0: 禁止接收帧计数器上溢中断 1: 允许接收帧计数器上溢中断
23	ADEIP	0	R/W	地址错误中断允许位 0: 禁止地址错误中断 1: 允许地址错误中断
22	ECIIP	0	R/W	EtherC 状态寄存器源中断允许位 0: 禁止 EtherC 状态中断 1: 允许 EtherC 状态中断
21	TCIP	0	R/W	帧发送结束中断允许位 0: 禁止帧发送结束中断 1: 允许帧发送结束中断
20	TDEIP	0	R/W	发送描述符空中断允许位 0: 禁止发送描述符空中断 1: 允许发送描述符空中断
19	TFUFIP	0	R/W	发送 FIFO 下溢中断允许位 0: 禁止下溢中断 1: 允许下溢中断

位	位名	初始值	R/W	说明
18	FRIP	0	R/W	帧接收中断允许位 0: 禁止帧接收中断 1: 允许帧接收中断
17	RDEIP	0	R/W	接收描述符空中断允许位 0: 禁止接收描述符空中断 1: 允许接收描述符空中断
16	RFOFIP	0	R/W	接收 FIFO 上溢中断允许位 0: 禁止上溢中断 1: 允许上溢中断
15 ~ 12	—	全 0	R	保留位 只能写 0。
11	CNDIP	0	R/W	载波未检测中断允许位 0: 禁止载波未检测中断 1: 允许载波未检测中断
10	DLCIP	0	R/W	载波消失检测中断允许位 0: 禁止载波消失检测中断 1: 允许载波消失检测中断
9	CDIP	0	R/W	延迟冲突检测中断允许位 0: 禁止延迟冲突中断 1: 允许延迟冲突中断
8	TROIP	0	R/W	试发送失败中断允许位 0: 禁止试发送失败中断 1: 允许试发送失败中断
7	RMAFIP	0	R/W	多播地址帧接收中断允许位 0: 禁止多播地址帧接收中断 1: 允许多播地址帧接收中断
6、5	—	全 0	R	保留位 只能写 0。
4	RRFIP	0	R/W	剩余位帧接收中断允许位 0: 禁止剩余位帧接收中断 1: 允许剩余位帧接收中断
3	RTLFIP	0	R/W	长帧接收错误中断允许位 0: 禁止长帧接收错误中断 1: 允许长帧接收错误中断
2	RTSFIP	0	R/W	短帧接收错误中断允许位 0: 禁止短帧接收错误中断 1: 允许短帧接收错误中断
1	PREIP	0	R/W	PHY-LSI 接收错误中断允许位 0: 禁止 PHY-LSI 接收错误中断 1: 允许 PHY-LSI 接收错误中断
0	CERFIP	0	R/W	接收帧 CRC 错误中断允许位 0: 禁止 CRC 错误中断 1: 允许 CRC 错误中断

20.2.8 发送 / 接收状态复制指示寄存器 (TRSCER)

TRSCER 指示是否将通过 EtherC/E-DMAC 状态寄存器的各位报告的发送和接收状态信息反映到相应描述符的 TFS25 ~ 0 和 RFS26 ~ 0。如果将对应 EtherC/E-DMAC 状态寄存器 (EESR) 的 bit11 ~ bit0 的此寄存器的各位置 0, 就将发送状态 (EESR 的 bit11 ~ bit8) 反映到发送描述符的 TFS3 ~ TFS0 位, 接收状态 (EESR 的 bit7 ~ bit0) 反映到接收描述符的 RFS7 ~ RFS0 位; 如果将各位置 1, 即使发生相应源, 也不反映到描述符。在 LSI 复位后各位为 0。

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
	—	—	—	—	CND CE	DLC CE	CD CE	TRO CE	RMAF CE	—	—	RRF CE	RTL CE	RTSF CE	PRE CE	CERF CE
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	R	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 12	—	全 0	R	保留位 读取值为 0, 只能写 0。
11	CNDCE	0	R/W	CND 位复制指示位 0: 将 CND 位的状态到发送描述符的 TFS 位。 1: 即使发生相应源, 也不反映到发送描述符的 TFS 位
10	DLCCE	0	R/W	DLC 位复制指示位 0: 将 DLC 位的状态反映到发送描述符的 TFS 位。 1: 即使发生相应源, 也不反映到发送描述符的 TFS 位。
9	CDCE	0	R/W	CD 位复制指示位 0: 将 CD 位的状态反映到发送描述符的 TFS 位。 1: 即使发生相应源, 也不反映到发送描述符的 TFS 位。
8	TROCE	0	R/W	TRO 位复制指示位 0: 将 TRO 位的状态反映到发送描述符的 TFS 位。 1: 即使发生相应源, 也不反映到发送描述符的 TFS 位。
7	RMAFCE	0	R/W	RMAF 位复制指示位 0: 将 RMAF 位的状态反映到接收描述符的 RFS 位。 1: 即使发生相应源, 也不反映到接收描述符的 RFS 位。
6、5	—	全 0	R	保留位 读取值为 0, 只能写 0。
4	RRFCE	0	R/W	RRF 位复制指示位 0: 将 RRF 位的状态反映到接收描述符的 RFS 位。 1: 即使发生相应源, 也不反映到接收描述符的 RFS 位。
3	RTLFCCE	0	R/W	RTLFC 位复制指示位 0: 将 RTLFC 位的状态反映到接收描述符的 RFS 位。 1: 即使发生相应源, 也不反映到接收描述符的 RFS 位。
2	RTSFCE	0	R/W	RTSF 位复制指示位 0: 将 RTSF 位的状态反映到接收描述符的 RFS 位。 1: 即使发生相应源, 也不反映到接收描述符的 RFS 位。

位	位名	初始值	R/W	说明
1	PRECE	0	R/W	PRE 位复制指示位 0: 将反映到接收描述符的 RFS 位反映 PRE 位的状态。 1: 即使发生相应源, 也不反映到接收描述符的 RFS 位。
0	CERFCE	0	R/W	CERF 位复制指示位 0: 将反映到接收描述符的 RFS 位。 1: 即使发生相应源, 也不反映到接收描述符的 RFS 位。

20.2.9 丢失帧计数寄存器 (RMFCR)

RMFCR 是 16 位计数器, 表示在接收过程中接收缓冲器无法保存而被舍弃的帧的个数。如果接收 FIFO 发生上溢, 就舍弃该 FIFO 内的接收帧。此时, 对舍弃的帧数进行递增计数。当此寄存器的值变 H'FFFF 时, 停止计数。通过写此寄存器, 将计数器的值清 0。

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
	MFC[15:0]															
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
31 ~ 16	—	全 0	R	保留位 读取值为 0, 只能写 0。
15 ~ 0	MFC[15:0]	全 0	R	丢失帧计数器 表示在接收时接收缓冲器无法保存而被废弃的帧的个数。

20.2.10 发送 FIFO 阈值指定寄存器 (TFTR)

TFTR 是 32 位可读写寄存器，指定在开始最初发送前的发送 FIFO 的阈值。实际的阈值相当于设定数值的 4 倍。在发送 FIFO 内的数据个数超过此寄存器指定的字节数、发送 FIFO 满或者写 1 帧的数据时，EtherC 开始发送。必须在发送停止状态下设定此寄存器。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TFT[10:0]										
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
31 ~ 11	—	全 0	R	保留位 读取值为 0，只能写 0。
10 ~ 0	TFT[10:0]	全 0	R/W	发送 FIFO 的阈值位 必须将发送 FIFO 的阈值设定为小于 FDR 指定的 FIFO 容量值。 H'000: 存储&正向模式 H'001 ~ H'00C: 禁止设定 H'00D: 52 字节 H'00E: 56 字节 : : H'01F: 124 字节 H'020: 128 字节 : : H'03F: 252 字节 H'040: 256 字节 : : H'07F: 508 字节 H'080: 512 字节 : : H'0FF: 1020 字节 H'100: 1024 字节 : : H'1FF: 2044 字节 H'200: 2048 字节 H'201 ~ H'7FF: 禁止设定

- 【注】
1. 如果在结束写 1 帧数据前开始发送，必须注意下溢的发生。
 2. 当此寄存器的设定值大于发送 / 接收 FIFO 容量时，不保证运行。
 3. 为了防止发生发送下溢，建议设定初始值（存储&正向模式）。

20.2.11 FIFO 容量指定寄存器 (FDR)

FDR 是 32 位可读写寄存器，指定发送和接收 FIFO 的容量。

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
	—	—	—	TFD[4:0]				—	—	—	RFD[4:0]					
初始值:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 13	—	全 0	R	保留位 读取值为 0，只能写 0。
12 ~ 8	TFD[4:0]	00111	R/W	发送 FIFO 的容量位 指定发送 FIFO 的容量。在发送 / 接收开始后，禁止变更设定值。 00000: 256 字节 00001: 512 字节 00010: 768 字节 00011: 1024 字节 00100: 1280 字节 00101: 1536 字节 00110: 1792 字节 00111: 2048 字节 上述以外: 禁止设定
7 ~ 5	—	全 0	R	保留位 读取值为 0，只能写 0。
4 ~ 0	RFD[4:0]	00111	R/W	接收 FIFO 的容量位 指定接收 FIFO 的容量。在发送 / 接收开始后，禁止变更设定值。 00000: 256 字节 00001: 512 字节 00010: 768 字节 00011: 1024 字节 00100: 1280 字节 00101: 1536 字节 00110: 1792 字节 00111: 2048 字节 上述以外: 禁止设定

【注】 当此寄存器的设定值大于发送 / 接收 FIFO 容量时，不保证运行。

20.2.12 接收方式控制寄存器 (RMCR)

RMCR 是 32 位可读写寄存器，指定接收帧时的 EDRRR 的 RR 位的控制方法。必须在接收停止状态下设定此寄存器。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RNC	RNR
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
31 ~ 2	—	全 0	R	保留位 读取值为 0，只能写 0。
1	RNC	0	R/W	接收启动位 non 复位模式指定位 0: nop 1: 将 EDRRR 的接收启动位 (RR) 的复位设定为软件控制。即使取得的描述符的 RACT 位为 0 (接收描述符空)，也不对 EDRRR 的接收启动位 (RR) 进行自复位，继续取接收描述符，并且继续进行接收帧的 DMA 传送。
0	RNR	0	R/W	接收启动位的复位位 0: 在结束 1 帧的接收时，通过硬件对 EDRRR 的接收启动位 (EDRRR.RR) 进行自清除。 控制能以 1 帧为单位。 在接收后续的接收帧时，需要再次设定 EDRRR 的接收启动位。 1: 通过高层软件控制 EDRRR 的接收启动位 (EDRRR.RR)。在将接收启动位 (EDRRR.RR) 置 1 后到将 EDRRR.RR 置 0 前，硬件自动取接收描述符，并且进行帧的接收。能连续接收多个帧。在连续接收时，建议将接收启动位的复位位置 1。但是，如果检测到接收描述符为空，就通过硬件对 EDRRR.RR 进行自清除。

20.2.13 发送 FIFO 下溢计数 (TFUCR)

TFUCR 是指定发送 FIFO 的下溢次数的寄存器。通过写此寄存器，将计数器的值清 0。

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
	UNDER[15:0]															
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 16	—	全 0	R	保留位 读取值为 0，只能写 0。
15 ~ 0	UNDER[15:0]	全 0	R/W	发送 FIFO 的下溢次数位 设定在发送 FIFO 时发生下溢的次数。 如果计数值变为 H'FFFF，就停止计数。

20.2.14 接收 FIFO 上溢计数 (RFOCR)

RFOCR 是指定接收 FIFO 的上溢次数的寄存器。通过写此寄存器，将计数器的值清 0。

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
	OVER[15:0]															
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 16	—	全 0	R	保留位 读取值为 0，只能写 0。
15 ~ 0	OVER[15:0]	全 0	R/W	接收 FIFO 的上溢次数位 设定在接收 FIFO 时发生上溢的次数。 如果计数值变为 H'FFFF，就停止计数。

20.2.15 接收缓冲器写地址寄存器 (RBWAR)

在 E-DMAC 给接收缓冲器写数据时，RBWAR 保存接收缓冲器内为写对象的缓冲器地址。通过监视此寄存器所示的地址，能知道 E-DMAC 正在处理接收缓冲器内的哪一个地址。有可能 E-DMAC 执行的缓冲器写处理和寄存器的读取值不相同。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RBWA[31:16]															
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
	RBWA[15:0]															
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
31 ~ 0	RBWA[31:0]	全 0	R	接收缓冲器的写地址位 此位是只读位，禁止写数据。

20.2.16 接收描述符取地址寄存器 (RDFAR)

在 E-DMAC 从接收描述符取描述符信息时，RDFAR 保存必需的描述符起始地址。通过监视此寄存器所示的地址，能知道 E-DMAC 正在处理哪一个接收描述符信息。有可能 E-DMAC 执行的描述符取处理和寄存器的读取值不相同。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDFA[31:16]															
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
	RDFA[15:0]															
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
31 ~ 0	RDFA[31:0]	全 0	R	接收描述符的取地址位 此位是只读位，禁止写数据。

20.2.17 发送缓冲器读地址寄存器 (TBRAR)

在 E-DMAC 从发送缓冲器读数据时，TBRAR 保存发送缓冲器内为读对象的缓冲器地址。通过监视此寄存器所示的地址，能知道 E-DMAC 正在处理发送缓冲器内的哪一个地址。有可能 E-DMAC 执行的缓冲器读处理和寄存器的读取值不相同。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TBRA[31:16]															
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
	TBRA[15:0]															
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
31 ~ 0	TBRA[31:0]	全 0	R	发送缓冲器的读地址位 此位是只读位，禁止写数据。

20.2.18 发送描述符取地址寄存器 (TDFAR)

在 E-DMAC 从发送描述符取描述符信息时，TDFAR 保存必需的描述符起始地址。通过监视此寄存器所示的地址，能知道 E-DMAC 正在处理哪一个发送描述符信息。有可能 E-DMAC 执行的描述符取处理和寄存器的读取值不相同。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TDFAR[31:16]															
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
	TDFAR[15:0]															
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
31 ~ 0	TDFAR[31:0]	全 0	R	发送描述符的取地址位 此位是只读，禁止写数据。

20.2.19 流控制开始 FIFO 阈值设定寄存器 (FCFTR)

FCFTR 是 32 位可读写寄存器，设定 EtherC 的流控制（设定自动 PAUSE 发送的阈值）。能通过接收 FIFO 的数据容量（RFDO[2:0]）和接收帧数（RFFO[2:0]）设定阈值。以接收 FIFO 的数据容量阈值判断和接收帧数阈值判断的逻辑或为条件，开始流控制。

在根据 RFDO 的设定条件开始流控制时，如果 RFDO 和 FIFO 容量指定寄存器（FDR）设定的接收 FIFO 容量值相同，就以（FIFO 数据容量 - 64）字节开始流控制。例如，在 FDR 的 RFD=1 并且 FCFTR 的 RFDO=1 时，如果将（2048-64）字节的数据保存到接收 FIFO 内，就开始流控制。此寄存器的 RFDO 设定值必须小于等于 FDR 的 RFD 设定值。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	RFFO[2:0]		
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

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

位	位名	初始值	R/W	说明
31 ~ 19	—	全 0	R	保留位 读取值为 0，只能写 0。
18 ~ 16	RFFO[2:0]	111	R/W	接收帧数上溢 BSY 的发送阈值位 000: 在接收 FIFO 内保存了 2 个接收帧时 001: 在接收 FIFO 内保存了 4 个接收帧时 010: 在接收 FIFO 内保存了 6 个接收帧时 : 110: 在接收 FIFO 内保存了 14 个接收帧时 111: 在接收 FIFO 内保存了 16 个接收帧时
15 ~ 3	—	全 0	R	保留位 读取值为 0，只能写 0。
2 ~ 0	RFDO[2:0]	111	R/W	接收 FIFO 上溢 BSY 的发送阈值 000: 在接收 FIFO 内保存了 256-64 字节的数据时 001: 在接收 FIFO 内保存了 512-64 字节的数据时 : 110: 在接收 FIFO 内保存了 1792-64 字节的数据时 111: 在接收 FIFO 内保存了 2048-64 字节的数据时

20.2.20 接收数据填充设定寄存器 (RPADIR)

RPADIR 是 32 位可读写寄存器，设定对接收数据的填充。必须在通过 E-DMAC 模式寄存器 (EDMR) 的 SWR 位进行复位后重新设定此寄存器。

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

位	位名	初始值	R/W	说明
31 ~ 18	—	全 0	R	保留位 读取值为 0，只能写 0。
17、16	PADS[1:0]	全 0	R/W	填充长度位 00: 无填充 01: 插入 1 字节 10: 插入 2 字节 11: 插入 3 字节
15 ~ 6	—	全 0	R	保留位 读取值为 0，只能写 0。
5 ~ 0	PADR[5:0]	全 0	R/W	填充范围位 H'00: 在接收数据的第 1 个字节前插入 PADS 位设置的字节 H'01: 在接收数据的第 2 个字节前插入 PADS 位设置的字节 : H'3E: 在接收数据的第 63 个字节前插入 PADS 位设置的字节 H'3F: 在接收数据的第 64 个字节前插入 PADS 位设置的字节

20.2.21 发送中断设定寄存器 (TRIMD)

TRIMD 是 32 位可读写寄存器，指定是否通过 EESR 的 TWB 位和中断来通知发送运行时各帧的回写结束。

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
	—	—	—	—	—	—	—	—	—	—	—	TIM	—	—	—	TIS
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

位	位名	初始值	R/W	说明
31 ~ 5	—	全 0	R	保留位 读取值为 0，只能写 0。
4	TIM	0	R/W	发送中断模式位 0: 每个发送帧模式 在每个发送帧回写结束时进行中断通知。 1: 中断模式 在由发送描述符 TWBI 位设定的描述符回写结束时进行中断通知。
3 ~ 1	—	全 0	R	保留位 读取值为 0，只能写 0。
0	TIS	0	R/W	发送中断设定位 0: 未设定中断 在 TIM 位指定的模式中，不进行中断通知。 在 TIS 位 =0 时，TIM 位的设定无效。 1: 设定中断 在 TIM 位指定的模式中，以 EESR 的 TWB 位为 1 进行中断通知。

20.2.22 各输出信号设定寄存器 (IOSR)

此位的值被输出到本 LSI 的通用外部输出引脚 (EXOUT)。EXOUT 引脚能用于指示 PHY-LSI 的回送模式等。在通过本功能实现 PHY-LSI 的回送功能时，PHY-LSI 必须有对应 EXOUT 引脚的引脚。

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ELB
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
31 ~ 1	—	全 0	R	保留位 读取值为 0，只能写 0。
0	ELB	0	R/W	外部回送模式位 0: EXOUT 引脚输出高电平 1: EXOUT 引脚输出低电平

20.3 运行说明

E-DMAC 和 EtherC 连接，不经过 CPU 进行数据的发送和接收，与存储器（缓冲器）进行高效率的传送。E-DMAC 读取被称为描述符的控制信息，此描述符对应各缓冲器并且保存缓冲器指针等信息。根据此控制信息，从发送缓冲器读发送数据，并且将接收数据写到接收缓冲器。通过将此描述符进行多个连续排列（描述符列表），能连续进行发送和接收。

20.3.1 描述符列表和数据缓冲器

在开始发送和接收前，通信程序在存储器中建立发送和接收的各描述符列表，然后将此列表的起始地址设定到发送和接收描述符列表起始地址寄存器。

必须将描述符的开始地址设定为 E-DMAC 模式寄存器（EDMR）的 DL 位（描述符长度指定位）指定的地址边界。发送缓冲器的开始地址能设定为长字边界、字边界或者字节边界。

(1) 发送描述符

发送描述符和发送缓冲器的关系如图 20.2 所示。通过此描述符的指示，能将发送帧和发送缓冲器构成 1 帧 1 个缓冲器或者 1 帧多个缓冲器的关系。

在指定发送缓冲器的长度为（TBL）1～16 字节时，需要将缓冲器地址设定为 32 字节边界。在指定发送缓冲器的长度（TBL）为 0 字节时，不保证运行。

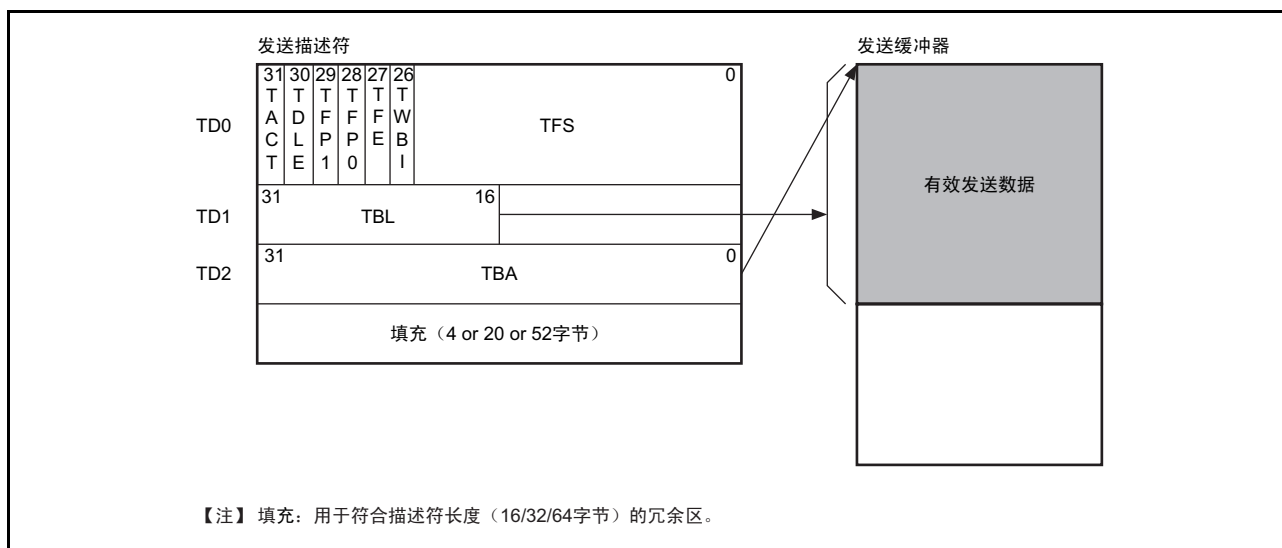


图 20.2 发送描述符和发送缓冲器的关系

(a) 发送描述符 0 (TD0)

TD0 表示发送帧的状态。以此提供帧的发送状态。

(下划线表示回写的位。)

位	位名	初始值	R/W	说明
<u>31</u>	<u>IACT</u>	0	R/W	发送描述符有效位 表示该描述符有效。通过软件将此位置位 (=1)，在结束发送帧的传送时或者因某种原因中止发送时，通过硬件将此位清除 (=0)。
30	TDLE	0	R/W	发送描述符环的结束位 在此位被置位 (=1) 时，表示该描述符为发送描述符环的最后部分。
29 28	TFP1 TFP0	0 0	R/W R/W	发送帧内的位置 1、0 建立发送缓冲器和发送帧的关系。此位和 TBL 位的设定必须维持前后描述符的正确逻辑关系。 00: 维持本描述符指示的发送缓冲器的帧发送 (未结束帧) 01: 本描述符指示的发送缓冲器含有帧的最后部分 (结束帧) 10: 本描述符指示的发送缓冲器为帧的起始部分 (未结束帧) 11: 本描述符指示的发送缓冲器的内容相当于 1 帧 (1 帧 1 个缓冲器)
<u>27</u>	<u>TFE</u>	0	R/W	发送帧错误位 在此位被置位 (=1) 时，表示 TFS 发生错误 (对于 TFS7 ~ TFS0，能通过 TRSCER 屏蔽此位。但是，在 TFS8 被置位时，不能屏蔽 TFS7 ~ TFS0 源。) 1: 发生帧发送中止
26	TWBI	0	R/W	回写结束后中断指示位 (此位在 TRIMD 被置位时有效。) 0: nop 1: 在给本描述符回写结束后产生中断。
<u>25 ~ 0</u>	TFS	全 0	R/W	发送帧的状态位 TFS25 ~ 9[保留 (只写 0)]: TFS8[检测到发送中止]: 当被置位 (=1) 时，表示在发送帧的过程中设定了中止信号 (=1)。 (TFE 置位源) TFS7 ~ 4[保留 (只能写为 0)]: TFS3[检测到无载波 (相当于 EESR 的 CND 位)]、 TFS2[检测到载波消失 (相当于 EESR 的 DLC 位)]、 TFS1[在发送中检测到延迟冲突 (相当于 EESR 的 CD 位)]、 TFS0[试发送失败 (相当于 EESR 的 TRO 位)]: 当被置位 (=1) 时，表示在发送帧的过程中 TFS8 ~ 1 被置位 (=1)。 (TFE 置位源，能通过 TRSCER 设定)

(b) 发送描述符 1 (TD1)

TD1 表示发送缓冲器的长度。

位	位名	初始值	R/W	说明
31 ~ 16	TBL	全 0	R/W	发送缓冲器长度位 表示该发送缓冲器的有效字节的长度。
15 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

(c) 发送描述符 2 (TD2)

TD2 表示该发送缓冲器的起始地址。

位	位名	初始值	R/W	说明
31 ~ 0	TBA	全 0	R/W	发送缓冲器地址位 表示发送缓冲器的起始地址。

(2) 接收描述符

接收描述符和接收缓冲器的关系如图 20.3 所示。必须将接收缓冲器的地址设定为 32 字节边界。
在指定接收缓冲器的长度 (RBL) 为 0 时，不保证运行。

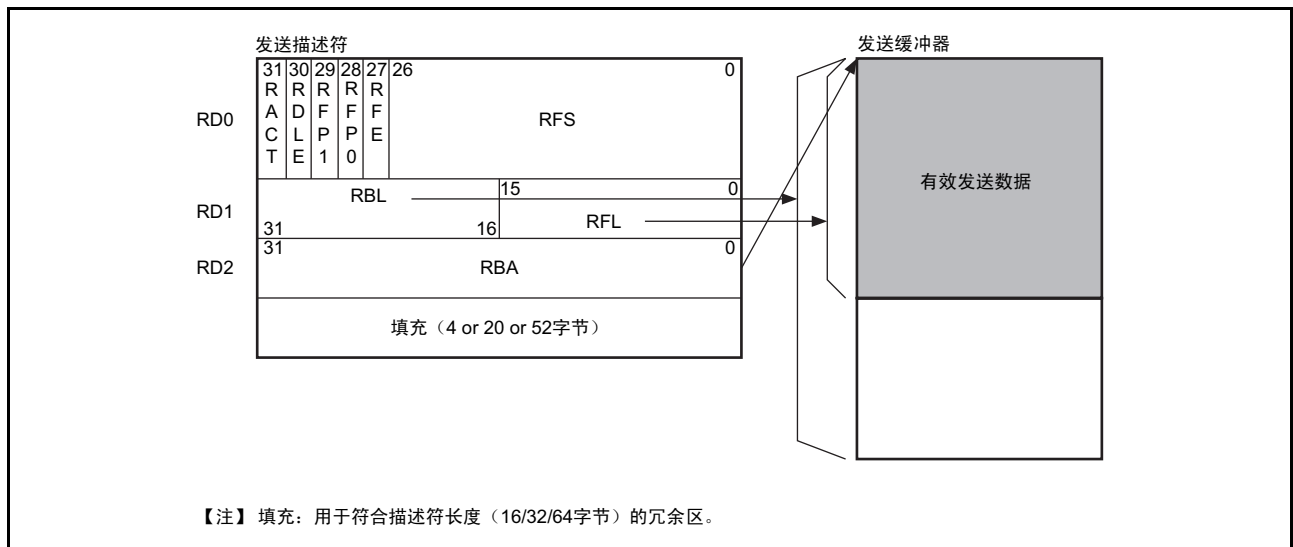


图 20.3 接收描述符和接收缓冲器的关系

(a) 接收描述符 0 (RD0)

RD0 表示接收帧的状态。以此提供帧的接收状态。

(下划线表示回写的位。)

位	位名	初始值	R/W	说明
<u>31</u>	<u>RACT</u>	0	R/W	接收描述符有效位 表示该描述符有效。通过软件设定将此位置位 (=1)，在将接收帧传送到 RD2 所示的缓冲器地址并且结束所有帧的传送时或者在接收缓冲器处于满状态时，通过硬件将此位清除 (=0)。
30	RDLE	0	R/W	接收描述符环的结束位 在此位被置位 (=1) 时，表示该描述符为接收描述符环的最后部分。
<u>29</u> 、 <u>28</u>	<u>RFP[1:0]</u>	00	R/W	接收帧内的位置 建立接收缓冲器和接收帧的关系。 00: 继续进行本描述符指示的接收缓冲器的帧接收 (未结束帧) 01: 本描述符指示的接收缓冲器含有帧的最后部分 (结束帧) 10: 本描述符指示的接收缓冲器为帧的起始部分 (未结束帧) 11: 本描述符指示的接收缓冲器的内容相当于 1 帧 (1 帧 1 个缓冲器)
<u>27</u>	<u>RFE</u>	0	R/W	接收帧错误位 在此位被置位 (=1) 时，表示 RFS 发生错误。 (对于 RFS7 ~ 0，能通过 TRSCER 屏蔽此位。但是，在 RFS8 被置位时，不能屏蔽 RFS7 ~ RFS0 源。)
<u>26 ~ 0</u>	<u>RFS</u>	全 0	R/W	接收帧的状态位 RFS26 ~ 10[保留 (只能写为 0)]； RFS9[接收 FIFO 发生上溢 (相当于 EESR 的 RFOF 位)]： 当被置位 (=1) 时，表示因接收 FIFO 发生上溢，回写中途被中止的帧。(RFE 置位源) RFS8[检测到接收中止]： 当被置位 (=1) 时，表示在接收帧的过程中设定了中止信号 (=1)。 (RFE 置位源) RFS7[接收多播地址帧 (相当于 EESR 的 RMAF 位)]、 RFS6、5[保留 (只能写为 0)]、 RFS4[剩余位帧接收错误 (相当于 EESR 的 RRF 位)]、 RFS3[长帧接收错误 (相当于 EESR 的 RTLf 位)]、 RFS2[短帧接收错误 (相当于 EESR 的 RTSF 位)]、 RFS1[PHY-LSI 接收错误 (相当于 EESR 的 PRE 位)]、 RFS0[检测到接收帧 CRC 错误 (相当于 EESR 的 CERF 位)]： 当被置位 (=1) 时，表示在接收帧的过程中 RFS8 ~ 1 被置位 (=1)。 (RFE 置位源，可通过 TRSCER 设定)

(b) 接收描述符 1 (RD1)

RD1 表示接收缓冲器的长度。
(下划线表示回写的位。)

位	位名	初始值	R/W	说明
31 ~ 16	RBL	全 0	R/W	接收缓冲器长度位 表示该接收缓冲器的字节长度。 必须将缓冲器的长度设定为 32×n 的长度。
<u>15 ~ 0</u>	<u>RFL</u>	全 0	R/W	接收帧长度位 表示保存在缓冲器的接收帧的长度 (字节数)。 不包含 RPADIR 指定的填充部分的字节数。 回写到含帧的最后部分的描述符。

(c) 接收描述符 2 (RD2)

RD2 表示该接收缓冲器的起始地址。

位	位名	初始值	R/W	说明
31 ~ 0	RBA	全 0	R/W	接收缓冲器地址 表示接收缓冲器的起始地址。 必须将缓冲器地址设定为 32 字节边界。

20.3.2 发送功能

如果发送功能有效并且将 E-DMAC 发送请求寄存器 (EDTRR) 的发送请求位 (TR) 置位, E-DMAC 就从发送描述符列表读取前次使用的描述符的下一个描述符 (初始状态是发送描述符起始地址寄存器 (TDLAR) 指定的描述符)。当读取的描述符的 TACT 位有效时, E-DMAC 就从 TD2 指定的发送缓冲器起始地址依次读取发送帧数据, 并且将其传送到 EtherC。EtherC 建立发送帧, 并且开始对 MII 进行发送。在描述符指示的缓冲器长度部分的 DMA 传送结束后, 根据 TFP 的值进行以下处理:

- TFP=00 or 10 (继续帧):
在 DMA 传送结束后进行描述符的回写 (只限 TACT 位)。
- TFP=01 or 11 (结束帧):
在帧发送结束后进行描述符的回写 (TACT 位和状态)。

当读取的描述符的 TACT 位有效时, 就继续读取 E-DMAC 描述符并且进行帧的发送; 当读取 TACT 位无效的描述符时, E-DMAC 就对 EDTRR 的 TR 位进行复位, 并且结束发送处理。

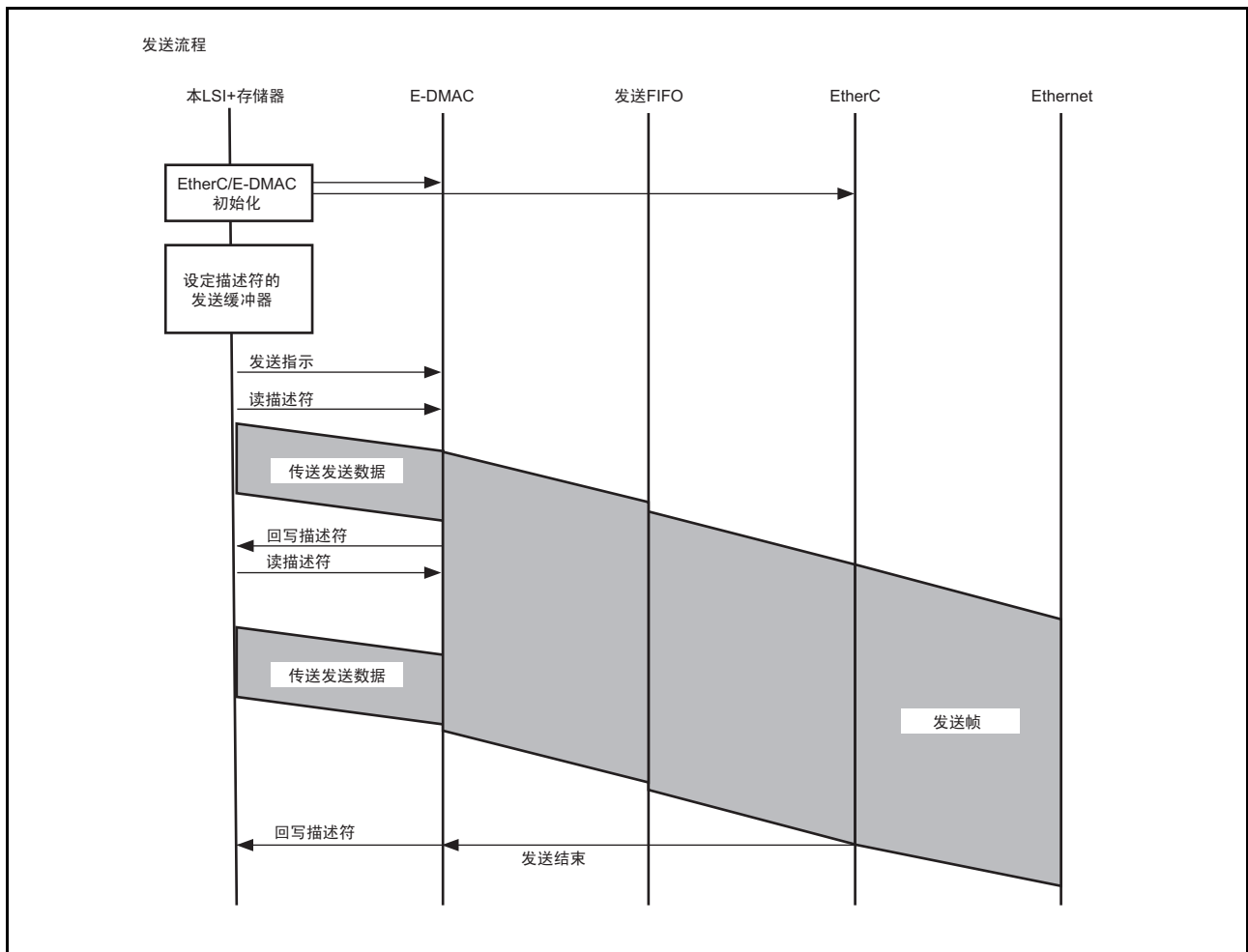


图 20.4 发送流程例子

20.3.3 接收功能

如果接收功能有效并且 CPU 将 E-DMAC 接收请求寄存器 (EDRRR) 的接收请求位 (RR) 置位, E-DMAC 就从接收描述符列表读取前次使用的描述符的下一个描述符 (初始状态是接收描述符起始地址寄存器 (RDLAR) 指定的描述符), 然后进入接收待机状态。如果接收到 RACT 位有效并且是本站的帧时, 就将其传送到 RD2 指定的接收缓冲器。如果接收的帧数据的长度大于 RD1 指示的缓冲器长度时, E-DMAC 就在缓冲器满时回写 (RFP = 10 or 00) 描述符, 然后读取下一个描述符, 继续将数据传送到新的 RD2 指定的接收缓冲器。当结束帧接收时或者因某种错误而中止帧的接收时, 就在回写 (RFP = 11 or 01) 该描述符后结束接收处理, 然后读下一个描述符, 进入接收待机状态。

在连续接收帧时, 必须将接收方式控制寄存器 (RMCR) 的接收控制位 (RNC) 置 1。此位在初始化后为 0。

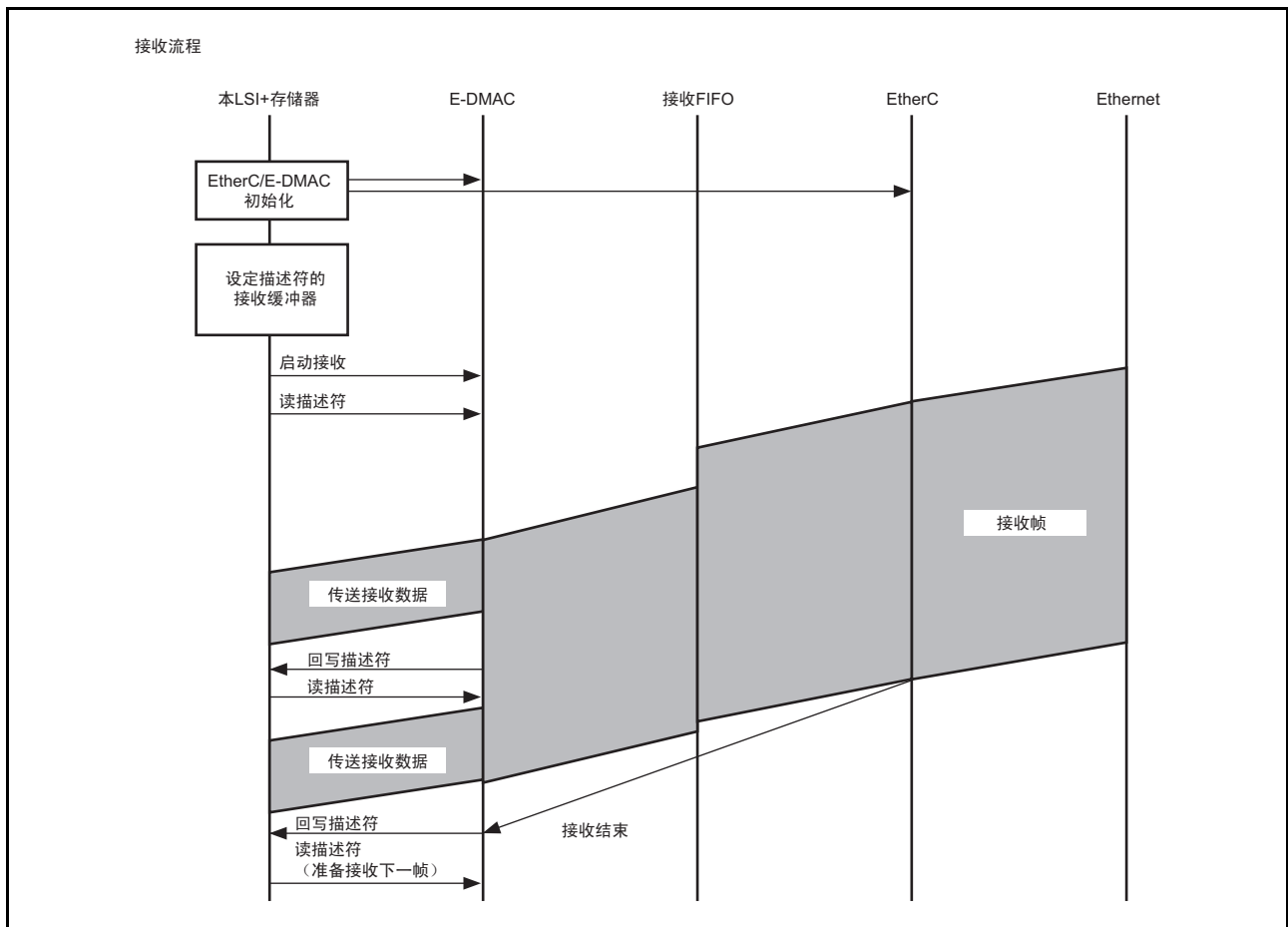


图 20.5 接收流程例子

20.3.4 多缓冲器帧的发送和接收处理

(1) 多缓冲器帧的发送处理

如果在发送多缓冲器帧的过程中发生错误，E-DMAC 就进行如图 20.6 所示的处理。

图中，发送描述符无效（TACT 位为 0）的部分表示全部已正常发送的缓冲数据；发送描述符有效（TACT 位为 1）的部分表示未发送的缓冲数据。在发送描述符有效（TACT 位为 1）的最初描述符部分，如果发生帧发送错误，就立即停止发送，并且将 TACT 位清 0，然后读下一个描述符，通过 TFP1 和 TFP0 位判断发送帧内的位置（继续 [B'00] 或者结束 [B'01]）。如果是继续的描述符，就只将 TACT 位清 0，并且立即读下一个描述符；如果是最后的描述符，就将 TACT 位清 0，并且同时回写 TFE 和 TFS 位。在从发生错误到回写最后的描述符之间，不发送缓冲器中的数据。当通过 EtherC/E-DMAC 状态中断允许寄存器（EESIPR）允许错误中断时，就在回写最后的描述符后立即产生中断。

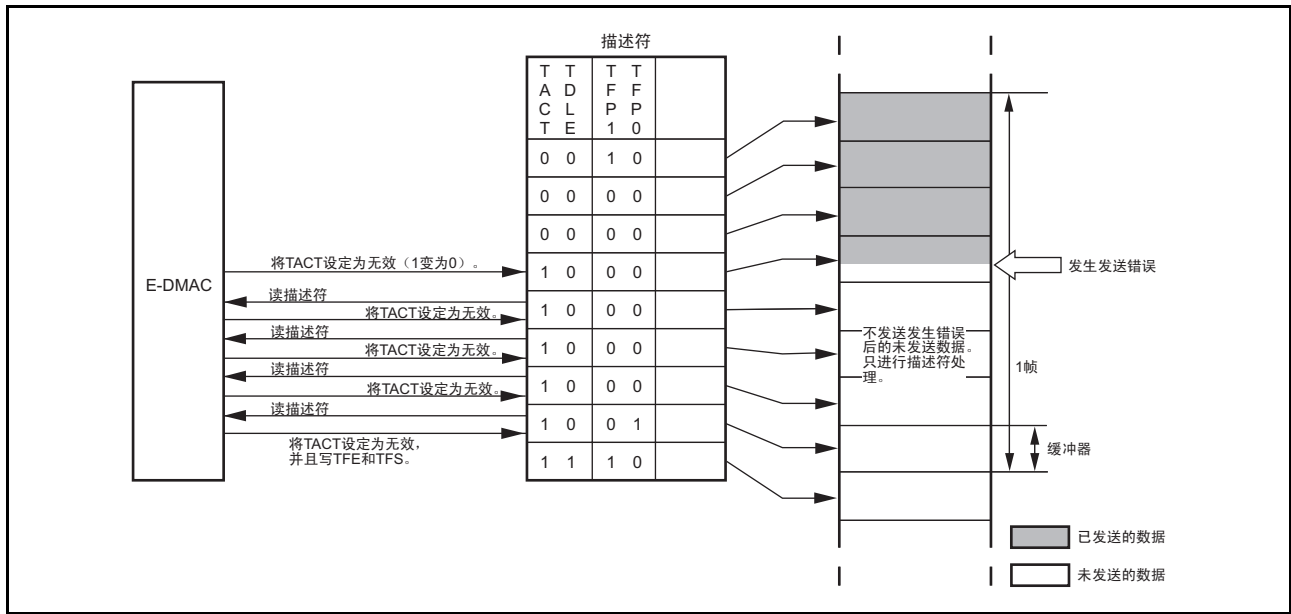


图 20.6 发生发送错误后的 E-DMAC 运行

(2) 多缓冲器帧的接收处理

如果在接收多缓冲器帧的过程中发生错误，E-DMAC 就进行如图 20.7 所示的处理。

图中，接收描述符无效（RACT 位为 0）的部分表示全部已正常接收的缓冲数据；接收描述符有效（RACT 位为 1）的部分表示未接收的缓冲器。在 RACT 位为 1 的最初描述符部分，如果发生帧接收错误，就将状态回写到该描述符。

当通过 EESIPR 允许错误中断时，就在回写后立即产生中断。如果有新的帧接收请求，就从发生错误的缓冲器的下一个缓冲器开始继续接收。

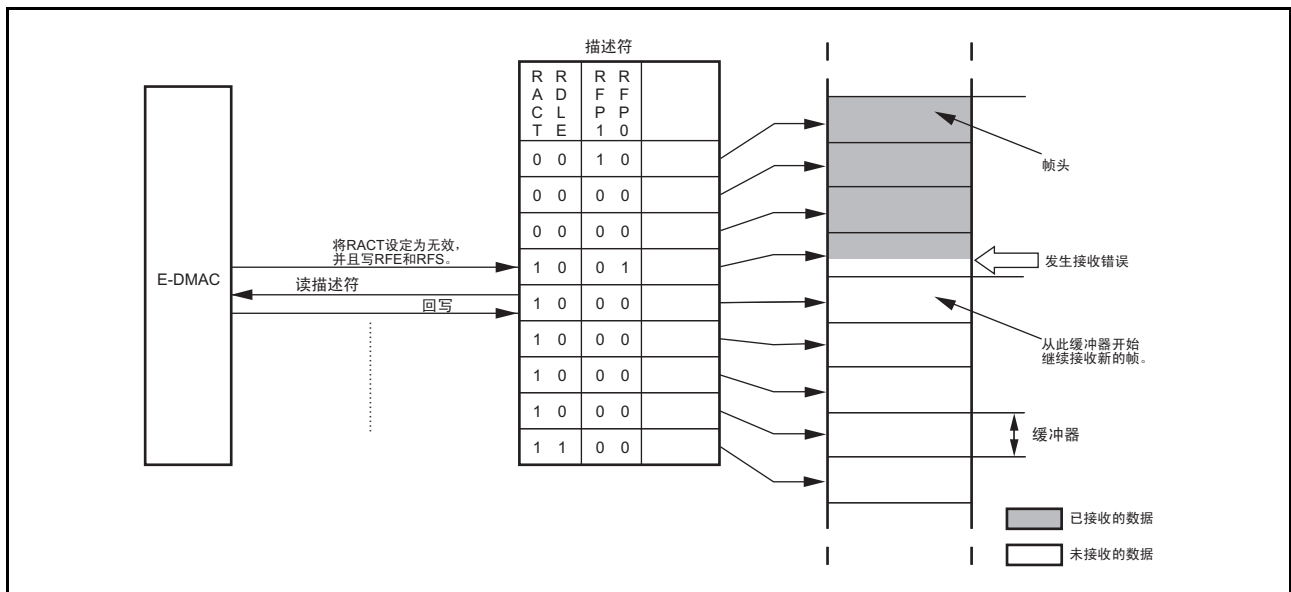


图 20.7 发生接收错误后的 E-DMAC 运行

第 21 章 USB2.0 主机 / 功能模块 (USB)

本模块是具有 USB 主机控制器功能和 USB 功能控制器功能的 USB 控制器。在选择主机控制器功能时，支持 USB (Universal Serial Bus) 规格 2.0 的高速传送、全速传送和低速传送；在选择功能控制器功能时，支持高速传送和全速传送。本模块还内置 USB 收发器，支持 USB 规格定义的所有传送类型。

内置用于数据传送的 5K 字节缓冲存储器，最多可使用 10 个管道。能给管道 1 ~ 9 分配与进行通信的外围设备或者用户系统相符的任意端点号。

21.1 特点

1. 内置支持 USB 高速运行的主机控制器和功能控制器

- 内置 USB 主机控制器和 USB 功能控制器。
- 能通过寄存器的设定进行 USB 主机控制器功能和 USB 功能控制器功能的转换。
- 内置 USB 收发器。

2. 能实现外接元件少并且空间小的安装

- 能直接将 VBUS 信号连接到本模块的输入引脚。
- 内置 D+ 上拉电阻 (功能控制器运行时)。
- 内置 D+、D- 下拉电阻 (主机控制器运行时)。
- 内置 D+、D- 终端电阻 (高速运行时)。
- 内置 D+、D- 输出电阻 (全速运行时)。

3. 支持 USB 通信的全部数据传送类型

- 控制传送
- 批量传送
- 中断传送 (不支持 High Bandwidth)
- 等时传送 (不支持 High Bandwidth)

4. 内部总线接口

- 内置 2 个 DMA 接口通道

5. 管道配置

- 能支持最大 5K 字节的 USB 通信缓冲存储器。
- 最多能选择 10 个管道 (含默认控制管道)。
- 能编程的管道结构
- 能给管道 1 ~ 9 分配任意的端点号。
- 各管道能设定的传送条件如下：
 - 管道 0: 是控制传送专用管道 (默认控制管道: DPC)，固定为 64 字节的单缓冲器。
 - 管道 1、2: 是能选择批量传送或者等时传送的管道，连续传送模式，可编程的缓冲器大小 (最大能指定 2K 字节的双缓冲器)。
 - 管道 3 ~ 5: 是批量传送专用管道，连续传送模式，可编程的缓冲器大小 (最大能指定 2K 字节的双缓冲器)。
 - 管道 6 ~ 9: 是中断传送专用管道，固定为 64 字节的单缓冲器。

6. 选择 USB 主机控制器功能时的特点

- 支持高速传送 (480Mbps)、全速传送 (12Mbps) 和低速传送 (1.5Mbps)。
- 能通过一段集线器，和多个外围设备连接进行通信。
- 自动应答复位握手信号。
- 自动安排 SOF 和信息包的发送时间。
- 能设定等时传送和中断传送的传送间隔。

7. 选择功能控制器功能时的特点

- 支持高速传送 (480Mbps) 和全速传送 (12Mbps)。
- 通过自动应答复位握手信号进行高速运行和全速运行的自动认识。
- 控制传送阶段管理功能
- 设备状态管理功能
- SET_ADDRESS 请求的自动应答功能
- NAK 应答中断功能 (NRDY)
- SOF 内插功能

8. 其他功能

- 通过事务计数的传送结束功能
- BRDY 中断事件通知时序的变更功能 (BFRE)
- 在读 DnFIFO (n=0、1) 端口指定的管道数据后的缓冲存储器自动清除功能 (DCLRM)
- 通过传送结束的应答 PID 的 NAK 设定功能 (SHTNAK)

21.2 输入 / 输出引脚

USB 的引脚结构如表 21.1 所示。

表 21.1 USB 引脚结构

引脚名	名称	输入 / 输出	功能
XIN	USB 晶体 / 外部时钟	输入	连接 USB 工作的晶体振荡器或者外部时钟。
XOUT	USB 晶体振荡器	输出	连接 USB 工作的晶体振荡器。
MODE7	USB 时钟模式控制	输入	选择 USB 时钟模式。 0: 从 XIN 输入外部时钟, 将 XOUT 置为开路。 1: 将 48MHz 晶体振荡器连接 XIN 和 XOUT。
DP	D+ 信号	输入 / 输出	是 USB D+ 信号引脚
DM	D- 信号	输入 / 输出	是 USB D- 信号引脚
VBUS	Vbus 信号	输入	是 USB Vbus 信号引脚
REFRIN	基准输入	输入	通过 5.6K Ω ($\pm 1\%$) 电阻连接模拟接地。
VDD_USB	USB PHY 数字电源	—	是 USB PHY 的数字电源引脚。输入 1.2V。
VSS_USB	USB PHY 数字接地	—	是 USB PHY 的数字接地引脚。输入 0V。
VDDQ_USB	USB PHY 数字电源	—	是 USB PHY 的数字电源引脚。输入 3.3V。
VSSQ_USB	USB PHY 数字接地	—	是 USB PHY 的数字接地引脚。输入 0V。
VDDA_USB	USB PHY 模拟电源	—	是 USB PHY 的模拟电源引脚。输入 1.2V。
VSSA_USB	USB PHY 模拟接地	—	是 USB PHY 的模拟接地引脚。输入 0V。
VDDQA_USB	USB PHY 模拟电源	—	是 USB PHY 的模拟电源引脚。输入 3.3V。
VSSQA_USB	USB PHY 模拟接地	—	是 USB PHY 的模拟接地引脚。输入 0V。
UV12	USB480MHz 电源	—	是 USB 的 480MHz 工作电源引脚。输入 1.2V。
UG1	USB480MHz 接地	—	是 USB 的 480MHz 工作接地引脚。输入 0V。

21.3 寄存器说明

USB 的寄存结构如表 21.2 所示，各种处理模式的寄存器状态如表 21.3 所示。

表 21.2 寄存器结构

寄存器名	略称	R/W	地址	存取长度
系统配置控制寄存器	SYSCFG	R/W	H'FE40 0000	16
CPU 总线等待设定寄存器	BUSWAIT	R/W	H'FE40 0002	16
系统配置状态寄存器	SYSSTS	R	H'FE40 0004	16
设备状态控制寄存器	DVSTCTR	R/W	H'FE40 0008	16
测试模式寄存器	TESTMODE	R/W	H'FE40 000C	16
DMA0-FIFO 总线配置寄存器	D0FBCFG	R/W	H'FE40 0010	16
DMA1-FIFO 总线配置寄存器	D1FBCFG	R/W	H'FE40 0012	16
CFIFO 端口寄存器	CFIFO	R/W	H'FE40 0014	8、16、32
D0FIFO 端口寄存器	D0FIFO	R/W	H'FE40 0018	8、16、32
			H'FE40 0180	
D1FIFO 端口寄存器	D1FIFO	R/W	H'FE40 001C	8、16、32
			H'FE40 01C0	
CFIFO 端口选择寄存器	CFIFOSEL	R/W	H'FE40 0020	16
CFIFO 端口控制寄存器	CFIFOCTR	R/W	H'FE40 0022	16
D0FIFO 端口选择寄存器	D0FIFOSEL	R/W	H'FE40 0028	16
D0FIFO 端口控制寄存器	D0FIFOCTR	R/W	H'FE40 002A	16
D1FIFO 端口选择寄存器	D1FIFOSEL	R/W	H'FE40 002C	16
D1FIFO 端口控制寄存器	D1FIFOCTR	R/W	H'FE40 002E	16
中断允许寄存器 0	INTENB0	R/W	H'FE40 0030	16
中断允许寄存器 1	INTENB1	R/W	H'FE40 0032	16
BRDY 中断允许寄存器	BRDYENB	R/W	H'FE40 0036	16
NRDY 中断允许寄存器	NRDYENB	R/W	H'FE40 0038	16
BEMP 中断允许寄存器	BEMPENB	R/W	H'FE40 003A	16
SOF 输出配置寄存器	SOFCFG	R/W	H'FE40 003C	16
中断状态寄存器 0	INTSTS0	R/W	H'FE40 0040	16
中断状态寄存器 1	INTSTS1	R/W	H'FE40 0042	16
BRDY 中断状态寄存器	BRDYSTS	R/W	H'FE40 0046	16
NRDY 中断状态寄存器	NRDYSTS	R/W	H'FE40 0048	16
BEMP 中断状态寄存器	BEMPSTS	R/W	H'FE40 004A	16
帧号寄存器	FRMNUM	R/W	H'FE40 004C	16
μ 帧号寄存器	UFRMNUM	R/W	H'FE40 004E	16
USB 地址寄存器	USBADDR	R	H'FE40 0050	16
USB 请求类型寄存器	USBREQ	R	H'FE40 0054	16
USB 请求值寄存器	USBVAL	R	H'FE40 0056	16
USB 请求索引寄存器	USBINDX	R	H'FE40 0058	16
USB 请求长度寄存器	USBLENG	R	H'FE40 005A	16
DCP 配置寄存器	DCPCFG	R/W	H'FE40 005C	16

寄存器名	略称	R/W	地址	存取长度
DCP 最大包长度寄存器	DCPMAXP	R/W	H'FE40 005E	16
DCP 控制寄存器	DCPCTR	R/W	H'FE40 0060	16
管道窗口选择寄存器	PIPESEL	R/W	H'FE40 0064	16
管道配置寄存器	PIPECFG	R/W	H'FE40 0068	16
管道缓冲器指定寄存器	PIPEBUF	R/W	H'FE40 006A	16
管道最大包长度寄存器	PIPEMAXP	R/W	H'FE40 006C	16
管道周期控制寄存器	PIPEPERI	R/W	H'FE40 006E	16
管道 1 控制寄存器	PIPE1CTR	R/W	H'FE40 0070	16
管道 2 控制寄存器	PIPE2CTR	R/W	H'FE40 0072	16
管道 3 控制寄存器	PIPE3CTR	R/W	H'FE40 0074	16
管道 4 控制寄存器	PIPE4CTR	R/W	H'FE40 0076	16
管道 5 控制寄存器	PIPE5CTR	R/W	H'FE40 0078	16
管道 6 控制寄存器	PIPE6CTR	R/W	H'FE40 007A	16
管道 7 控制寄存器	PIPE7CTR	R/W	H'FE40 007C	16
管道 8 控制寄存器	PIPE8CTR	R/W	H'FE40 007E	16
管道 9 控制寄存器	PIPE9CTR	R/W	H'FE40 0080	16
管道 1 事务计数允许寄存器	PIPE1TRE	R/W	H'FE40 0090	16
管道 1 事务计数寄存器	PIPE1TRN	R/W	H'FE40 0092	16
管道 2 事务计数允许寄存器	PIPE2TRE	R/W	H'FE40 0094	16
管道 2 事务计数寄存器	PIPE2TRN	R/W	H'FE40 0096	16
管道 3 事务计数允许寄存器	PIPE3TRE	R/W	H'FE40 0098	16
管道 3 事务计数寄存器	PIPE3TRN	R/W	H'FE40 009A	16
管道 4 事务计数允许寄存器	PIPE4TRE	R/W	H'FE40 009C	16
管道 4 事务计数寄存器	PIPE4TRN	R/W	H'FE40 009E	16
管道 5 事务计数允许寄存器	PIPE5TRE	R/W	H'FE40 00A0	16
管道 5 事务计数寄存器	PIPE5TRN	R/W	H'FE40 00A2	16
设备地址 0 配置寄存器	DEVADD0	R/W	H'FE40 00D0	16
设备地址 1 配置寄存器	DEVADD1	R/W	H'FE40 00D2	16
设备地址 2 配置寄存器	DEVADD2	R/W	H'FE40 00D4	16
设备地址 3 配置寄存器	DEVADD3	R/W	H'FE40 00D6	16
设备地址 4 配置寄存器	DEVADD4	R/W	H'FE40 00D8	16
设备地址 5 配置寄存器	DEVADD5	R/W	H'FE40 00DA	16
设备地址 6 配置寄存器	DEVADD6	R/W	H'FE40 00DC	16
设备地址 7 配置寄存器	DEVADD7	R/W	H'FE40 00DE	16
设备地址 8 配置寄存器	DEVADD8	R/W	H'FE40 00E0	16
设备地址 9 配置寄存器	DEVADD9	R/W	H'FE40 00E2	16
设备地址 A 配置寄存器	DEVADDA	R/W	H'FE40 00E4	16

表 21.3 各处理模式的寄存器状态

寄存器略称	上电复位	待机	模块待机	睡眠
SYSCFG	初始化	保持	保持	保持
BUSWAIT	初始化	保持	保持	保持
SYSSTS	初始化	保持	保持	保持
DVSTCTR	初始化	保持	保持	保持
TESTMODE	初始化	保持	保持	保持
D0FBCFG	初始化	保持	保持	保持
D1FBCFG	初始化	保持	保持	保持
CFIFO	初始化	保持	保持	保持
D0FIFO	初始化	保持	保持	保持
D1FIFO	初始化	保持	保持	保持
CFIFOSEL	初始化	保持	保持	保持
CFIFOCTR	初始化	保持	保持	保持
D0FIFOSEL	初始化	保持	保持	保持
D0FIFOCTR	初始化	保持	保持	保持
D1FIFOSEL	初始化	保持	保持	保持
D1FIFOCTR	初始化	保持	保持	保持
INTENB0	初始化	保持	保持	保持
INTENB1	初始化	保持	保持	保持
BRDYENB	初始化	保持	保持	保持
NRDYENB	初始化	保持	保持	保持
BEMPENB	初始化	保持	保持	保持
SOFCFG	初始化	保持	保持	保持
INTSTS0	初始化	保持	保持	保持
INTSTS1	初始化	保持	保持	保持
BRDYSTS	初始化	保持	保持	保持
NRDYSTS	初始化	保持	保持	保持
BEMPSTS	初始化	保持	保持	保持
FRMNUM	初始化	保持	保持	保持
UFRMNUM	初始化	保持	保持	保持
USBADDR	初始化	保持	保持	保持
USBREQ	初始化	保持	保持	保持
USBVAL	初始化	保持	保持	保持
USBINDX	初始化	保持	保持	保持
USBLENG	初始化	保持	保持	保持
DCPCFG	初始化	保持	保持	保持
DCPMAXP	初始化	保持	保持	保持
DCPCTR	初始化	保持	保持	保持
PIPESEL	初始化	保持	保持	保持
PIPECFG	初始化	保持	保持	保持
PIPEBUF	初始化	保持	保持	保持

寄存器略称	上电复位	待机	模块待机	睡眠
PIPEMAXP	初始化	保持	保持	保持
PIPEPERI	初始化	保持	保持	保持
PIPE1CTR	初始化	保持	保持	保持
PIPE2CTR	初始化	保持	保持	保持
PIPE3CTR	初始化	保持	保持	保持
PIPE4CTR	初始化	保持	保持	保持
PIPE5CTR	初始化	保持	保持	保持
PIPE6CTR	初始化	保持	保持	保持
PIPE7CTR	初始化	保持	保持	保持
PIPE8CTR	初始化	保持	保持	保持
PIPE9CTR	初始化	保持	保持	保持
PIPE1TRE	初始化	保持	保持	保持
PIPE1TRN	初始化	保持	保持	保持
PIPE2TRE	初始化	保持	保持	保持
PIPE2TRN	初始化	保持	保持	保持
PIPE3TRE	初始化	保持	保持	保持
PIPE3TRN	初始化	保持	保持	保持
PIPE4TRE	初始化	保持	保持	保持
PIPE4TRN	初始化	保持	保持	保持
PIPE5TRE	初始化	保持	保持	保持
PIPE5TRN	初始化	保持	保持	保持
DEVADD0	初始化	保持	保持	保持
DEVADD1	初始化	保持	保持	保持
DEVADD2	初始化	保持	保持	保持
DEVADD3	初始化	保持	保持	保持
DEVADD4	初始化	保持	保持	保持
DEVADD5	初始化	保持	保持	保持
DEVADD6	初始化	保持	保持	保持
DEVADD7	初始化	保持	保持	保持
DEVADD8	初始化	保持	保持	保持
DEVADD9	初始化	保持	保持	保持
DEVADDA	初始化	保持	保持	保持

21.3.1 系统配置控制寄存器 (SYSCFG)

SYSCFG 是允许高速运行、选择主机控制器功能或者功能控制器功能、控制 DP 和 DM 引脚、允许本模块运行的寄存器。

在上电复位时，此寄存器被初始化。

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

位	位名	初始值	R/W	说明
15 ~ 11	—	全 0	R	保留位 读取值为 0，只能写 0。
10	SCKE	0	R/W	USB 模块时钟允许位 指定是停止还是允许给本模块提供 48MHz 时钟。 0: 停止给 USB 模块提供时钟 1: 允许给 USB 模块提供时钟 在此位为 0 时，只能读写此寄存器和 BUSWAIT 寄存器。 能读 USB 模块内的其他寄存器，但是不能写。
9、8	—	全 0	R	保留位 读取值为 0，只能写 0。
7	HSE	HSE0	R/W	高速运行允许位 0: 禁止高速运行 在选择功能控制器功能时，只允许全速运行 在选择主机控制器功能时，只允许全速运行 1: 允许高速运行 (检测通信速度) 1. 选择主机控制器功能时 在 HSE=0 时，USB 端口进行全速运行；在 HSE=1 时，执行复位握手协议，并且按照执行结果，USB 端口自动进行高速或者全速运行。 必须在检测到连接 (检测到 ATTCH 中断) 后并且在进行 USB 总线复位前 (设定 USBRESET=1 前) 更改此位。 2. 选择功能控制器功能时 在 HSE=0 时，进行全速运行；在 HSE=1 时，执行复位握手协议，并且按照执行结果，自动进行高速或者全速运行。必须在 DPRPU=0 时更改此位。
6	DCFM	0	R/W	控制器功能选择位 选择本模块的功能。 0: 选择功能控制器功能 1: 选择主机控制器功能 必须在 DPRPU=0 并且 DRPD=0 时更改此位。
5	DRPD	0	R/W	D+/D- 线路电阻控制位 在选择主机控制器功能时，指定是禁止还是允许 D+/D- 线路下拉。 0: 禁止下拉 1: 允许下拉 只在选择主机控制器功能时才能将此位置 1；在选择功能控制器功能时，必须将此位置 0。

位	位名	初始值	R/W	说明
4	DPRPU	0	R/W	D+ 线路电阻控制位 在选择功能控制器功能时，指定是禁止还是允许 D+ 线路上拉。 0: 禁止上拉 1: 允许上拉 如果在选择功能控制器功能时将此位置 1，就将 D+ 线路上拉到 3.3V，并且能将已连接状态通知 USB 主机。在此位从 1 变为 0 时，解除 D+ 线路上拉，因此能将已断开状态通知 USB 主机。 只在选择功能控制器功能时才能将此位置 1；在选择主机控制器功能时，必须将此位置 0。
3 ~ 1	—	全 0	R	保留位 读取值为 0，只能写 0。
0	USBE	0	R/W	USB 模块运行允许位 指定是禁止还是允许本模块的运行。 0: 禁止 USB 模块运行 1: 允许 USB 模块运行 在此位从 1 变为 0 时，被初始化的寄存器和位如表 21.4 和表 21.5 所示。 必须在 SCKE=1 时更改此位。 在选择主机控制器功能时，必须在设定 DPRD=1 后消除 LNST 位的颤动，并且在确认 USB 总线状态稳定后设定 USBE=1。

表 21.4 通过 USBE=0 的设定被初始化的寄存器 (选择功能控制器功能时)

寄存器名	位名	备注
SYSSTS	LNST	在选择主机控制器功能时，保持值。
DVSTCTR	RHST	
INTSTS0	DVSQ	在选择主机控制器功能时，保持值。
USBADDR	USBADDR	在选择主机控制器功能时，保持值。
USEREQ	BRequest、bmRequestType	在选择主机控制器功能时，保持值。
USBVAL	wValue	在选择主机控制器功能时，保持值。
USBINDX	wIndex	在选择主机控制器功能时，保持值。
USBLENG	wLength	在选择主机控制器功能时，保持值。

表 21.5 通过 USBE=0 的设定被初始化的寄存器 (选择主机控制器功能时)

寄存器名	位名	备注
DVSTCTR	RHST	
FRMNUM	FRNM	在选择功能控制器功能时，保持值。
UFRMNUM	UFRNM	在选择功能控制器功能时，保持值。

21.3.2 CPU 总线等待寄存器 (BUSWAIT)

BUSWAIT 指定 CPU 存取本模块时的等待数。

即使在 SYSCFG 寄存器的 SCKE 位为 0 时，也能写此寄存器。

在上电复位时，此寄存器被初始化。

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

位	位名	初始值	R/W	说明
15 ~ 4	—	全 0	R	保留位 读取值为 0，只能写 0。
3 ~ 0	BWAIT[3:0]	1111	R/W	CPU 总线等待位 设定存取本模块时的等待数。 0000 ~ 0011: 禁止设定 0100: 4 个等待 (6 个存取周期) 0110: 6 个等待 (8 个存取周期) : 1111: 15 个等待 (17 个存取周期) 初始值

对该控制器的地址 H'04 以后的寄存器进行存取时的存取周期有以下规定：
等待规定：对该控制器的寄存器进行连续存取时的存取周期不能小于 80ns。

为了满足此规定，需要根据输入的 SHwy 时钟频率进行等待控制。
因为初始值为最大值 (17 个时钟周期)，所以必须选择合适的设定值。
SHwy 时钟 MIN 和 MAX 时的 BWAIT 计算方法和设定值如下所示：

[算式]

$$80\text{ns} \div \{(1 \div \text{使用的 SHwy 时钟频率}) \div 10^{-9}\} - 1$$

[SHwy 时钟的 BWAIT 设定值]

<SHwy 时钟>	<BWAIT 设定值>
80MHz	0100
108MHz	0110

21.3.3 系统配置状态寄存器 (SYSSTS)

SYSSTS 监视 USB 数据总线的线路状态 (D+ 及 D- 线路)。

在上电复位或者 USB 复位时, 此寄存器被初始化。

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

位	位名	初始值	R/W	说明
15 ~ 11	—	全 0	R	保留位 读取值为 0, 只能写 0。
10	—	1	R	保留位 读取值为 1, 只能写 1。
9 ~ 2	—	全 0	R	保留位 读取值为 0, 只能写 0。
1、0	LNST[1:0]	不定值 *	R	USB 数据线路状态监视位 表示 USB 数据总线的线路 (D+ 线路和 D- 线路) 状态。USB 数据总线的线路状态如表 21.6 所示。 必须在选择功能控制器功能时进行连接处理 (设定 DPRPU=1) 后或者在选择主机控制器功能时允许下拉 (设定 DRPD=1) 后, 参照此位。

【注】 * 取决于 DP 和 DM 引脚的状态。

表 21.6 USB 数据总线的线路状态表

LNST[1]	LNST[0]	低速运行 (只在选择主机控制器功能时)	全速运行	高速运行	Chirp 运行
0	0	SE0	SE0	Squelch	Squelch
0	1	K-State	K-State	UnSquelch	Chirp J
1	0	J-State	J-State	Invalid	Chirp K
1	1	SE1	SE1	Invalid	Invalid

【符号说明】

Chirp: 在允许高速运行的状态 (SYSCFG.HSE=1) 下, 正在执行复位握手协议 (RHSP)。

Squelch: SE0 或者空闲状态

UnSquelch: 高速 J-State 或者高速 K-State

Chirp J: Chirp J-State

Chirp K: Chirp K-State

Invalid: 无效

21.3.4 设备状态控制寄存器 (DVSTCTR)

DVSTCTR 控制并且确认 USB 数据总线的状态。

在上电复位时，此寄存器被初始化；在 USB 总线复位时，WKUP 位被初始化，RESUME 位为不定值。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	WKUP	RWUPE	USBRST	RESUME	UACT	—	RHST[2:0]		
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W*	R/W	R/W	R/W	R/W	R	R	R	R

位	位名	初始值	R/W	说明
15 ~ 9	—	全 0	R	保留位 读取值为 0，只能写 0。
8	WKUP	0	R/W*	唤醒输出位 在选择功能控制器功能时，指定是禁止还是允许对 USB 总线输出远程唤醒（输出恢复信号）。 0：不输出远程唤醒信号 1：输出远程唤醒信号 本模块控制远程唤醒信号的输出时间。如果将此位置 1，就在输出 10ms 的 K-State 后将此位置 0。 根据 USB 规格，需要在发送远程唤醒信号前至少保持 5ms 的 USB 总线空闲状态。因此，即使在检测到挂起状态后立即将此位置 1，也要等待 2ms 后才输出 K-State。 只在设备处于挂起状态（INTSTS0.DVSQ=1xx）并且 USB 主机允许远程唤醒时，才能将此位置 1。在将此位置 1 时，即使在挂起中，也不能停止内部时钟（必须在 SCKE=1 状态下将 WKUP 置 1。） 在选择主机控制器功能时，必须将此位置 0。
7	RWUPE	0	R/W	唤醒检测允许位 在选择主机控制器功能时，指定是禁止还是允许对下行端口外围设备输出远程唤醒（输出恢复信号）。 0：禁止对下行端口输出远程唤醒 1：允许对下行端口输出远程唤醒 如果将此位置 1，就在检测到远程唤醒信号时对下行端口检测恢复信号（2.5μs 的 K-State），并且进行恢复处理（K-State 驱动）。 当将此位置 0 时，即使检测到从连接到 USB 端口的外围设备发出的远程唤醒信号（K-State），也忽视此信号。 当将此位置 1 时，即使在挂起过程中也不能停止内部时钟（必须设定 SCKE=1 的状态），并且不能从挂起状态进行 USB 总线的复位（USBRST=1）。在 USB Specification2.0 的情况下禁止。 在选择功能控制器功能时，必须将此位置 0。

位	位名	初始值	R/W	说明
6	USBRST	0	R/W	<p>总线复位输出位</p> <p>在选择主机控制器功能时, 控制 USB 总线复位信号的输出。</p> <p>0: 不输出 USB 总线复位信号</p> <p>1: 输出 USB 总线复位信号</p> <p>在选择主机控制器功能时, 如果将此位置 1, 就将 USB 端口驱动到 SE0, 进行 USB 总线复位。此时, 如果 HSE 位为 1, 就执行 Reset Handshake Protocol。</p> <p>在 USBRST=1 期间 (在软件将 USBRST 置 0 前) 继续输出 SE0。必须在 USBRST=1 期间 (USB 总线复位期间) 确保 USB Specification2.0 规定的时间。</p> <p>如果在通信中 (UACT=1) 或者恢复中 (RESUME=1) 将此位置 1, 就在变为 UACT=0 并且 RESUME=0 状态前不开始 USB 总线复位。必须在 USB 总线复位结束 (USBRST=0) 的同时将 UACT 位置 1。</p> <p>在选择功能控制器功能时, 必须将此位置 0。</p>
5	RESUME	0	R/W	<p>恢复输出位</p> <p>在选择主机控制器功能时, 控制恢复信号的输出。</p> <p>0: 不输出恢复信号</p> <p>1: 输出恢复信号</p> <p>如果将此位置 1, 就将端口驱动到 K-State, 并且进行恢复输出。</p> <p>在 RESUME=1 期间 (在软件将 RESUME 置 0 前) 继续输出 K-State。必须在 RESUME=1 期间 (恢复期间) 确保 USB Specification2.0 规定的时间。只能在挂起时将此位置 1。</p> <p>必须在恢复处理结束 (RESUME=0) 的同时将 UACT 位置 1。</p> <p>在选择功能控制器功能时, 必须将此位置 0。</p>
4	UACT	0	R/W	<p>USB 总线允许位</p> <p>在选择主机控制器功能时, 允许 USB 总线运行 (对 USB 总线进行 SOF 或者 μSOF 包的发送控制)。</p> <p>0: 禁止下行端口运行 (禁止发送 SOF/μSOF)</p> <p>1: 允许下行端口运行 (允许发送 SOF/μSOF)</p> <p>如果将此位置 1, 就将 USB 端口设定为 USB 总线允许状态, 并且进行 SOF 输出和数据的发送 / 接收。</p> <p>在软件将 UACT 置 1 后的 1 (μ) 帧时间内开始输出 SOF/μSOF。</p> <p>如果将此位置 0, 本模块就在输出 SOF/μSOF 后转移到空闲状态。</p> <p>在以下情况下, 此位置 0:</p> <ul style="list-style-type: none"> 在通信中 (UACT=1) 检测到 DTCH 中断 在通信中 (UACT=1) 检测到 EOFERR 中断 <p>必须在 USB 复位处理结束 (USBRST=0) 时或者从挂起的恢复处理结束 (RESUME=0) 时将此位置 1。</p> <p>在选择功能控制器功能时, 必须将此位置 0。</p>
3	—	0	R	<p>保留位</p> <p>读取值为 0, 只能写 0。</p>

位	位名	初始值	R/W	说明
2 ~ 0	RHST[2:0]	000	R	<p>复位握手信号位 表示复位握手信号状态。</p> <p>1. 选择主机控制器功能时 000: 通信速度为不定值 (在供电或者断开时) 1xx: 复位握手信号处理中 001: 低速连接时 010: 全速连接时 011: 高速连接时 在通过软件将 USBRST 置 1 后, 此位为 100。 如果对端口设定 HSE=1, 就在本模块从外围设备检测到 ChirpK 时 此位为 111。 在通过软件将 USBRST 置 0 并且本模块结束 SE0 驱动时, 确定 RHST 位的值。</p> <p>2. 选择功能控制器功能时 000: 通信速度为不定值 100: 复位握手信号处理中 010: 全速连接时 011: 高速连接时 在 HSE=1 时, 如果检测到 USB 总线复位, 此位就为 100。此后, 输出 ChirpK, 并且在从 USB 主机检测到 3 次 ChirpJK 时, 此位变为 011。 如果在输出 ChirpK 后的 2.5ms 内不能确定为高速, 此位就变为 010。 在 HSE=0 时, 如果检测到 USB 总线复位, 此位就为 010。 在检测到 USB 总线复位后并且在 RHST 位确定为 010 或者 011 时, 产生 DVST 中断。</p>

【注】 * 只能写 1。

21.3.5 测试模式寄存器 (TESTMODE)

TESTMODE 控制高速运行时的 USB 测试信号输出。

在上电复位时，此寄存器被初始化。

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

位	位名	初始值	R/W	说明
15 ~ 4	—	全 0	R	保留位 读取值为 0，只能写 0。
3 ~ 0	UTST[3:0]	0000	R/W	<p>测试模式位</p> <p>在给此位写数据时，本模块输出高速运行时的 USB 测试信号。本模块的测试模式运行表如表 21.7 所示。</p> <p>1. 选择主机控制器功能时 在 DRPD=1 后能设定此位。本模块对 DRPD=1 并且 UACT=1 的 USB 端口输出波形，对 USB 端口进行高速端接。 — UTST 位的设定步骤如下： 1. 上电复位 2. 启动时钟（晶体振荡以及在 USB PLL 稳定后将 SCKE 位置 1）。 3. DCFM=1、DRPD=1（不需要设定 HSE=1。） 4. USBE=1 5. 给 UTST 位设定对应测试内容的值。 6. 将 UACT 位置 1。 — UTST 位的变更步骤如下： 1.（在上述 6 的状态下）UACT=0、USBE=0 2. USBE=1 3. 给 UTST 位设定对应测试内容的值。 4. 将 UACT 位置 1。</p> <p>在设定 Test_SE0_NAK（1011）时，即使是对 UACT=1 的端口，也不输出 SOF 包。 在设定 Test_Force_Enable（1101）时，本模块对 UACT=1 的端口输出 SOF 包。在设定此模式时，即使检测到高速断开（检测到 DTCH 中断），也不进行随之发生的硬件控制。 在设定 UTST 位时，必须将全部管道的 PID 位设定为 NAK。 在设定测试模式后进行正常的 USB 通信时，必须进行上电复位。</p> <p>2. 选择功能控制器功能时 在高速通信时，根据 USB 主机发出的 SetFeature 请求设定此位。 在将此位置 0001 ~ 0100 时，本模块不转移到挂起状态。</p>

表 21.7 测试模式运行表

测试模式	UTST 位的设定	
	选择功能控制器功能	选择主机控制器功能
正常运行	0000	0000
Test_J	0001	1001
Test_K	0010	1010
Test_SE0_NAK	0011	1011
Test_Packet	0100	1100
Test_Force_Enable	—	1101
Reserved	0101 ~ 0111	1110 ~ 1111

21.3.6 DMA-FIFO 总线配置寄存器 (D0FBCFG、D1FBCFG)

D0FBCFG 和 D1FBCFG 寄存器分别是控制 DMA0-FIFO 和 DMA1-FIFO 总线存取的寄存器。
在上电复位时，此寄存器被初始化。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	DFACC	—	—	—	—	—	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R	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 读取值为 0，只能写 0。
13、12	DFACC	00	R/W	DMA _n -FIFO 缓冲器存取模式位 (n=0, 1) 指定 DMA0-FIFO 或者 DMA1-FIFO 端口的存取模式。 00: 周期挪用模式 (初始值) 01: 16 字节连续存取模式 10: 32 字节连续存取模式 11: 无效
11 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

21.3.7 FIFO 端口寄存器 (CFIFO、D0FIFO、D1FIFO)

CFIFO、D0FIFO、D1FIFO 是用于进行 FIFO 缓冲存储器的数据读写的端口寄存器。

FIFO 端口有 CFIFO、D0FIFO、D1FIFO 共 3 个端口。各 FIFO 端口由进行 FIFO 缓冲存储器的数据读写的端口寄存器 (CFIFO、D0FIFO、D1FIFO)、选择给 FIFO 端口分配管道的选择寄存器 (CFIFOSEL、D0FIFOSEL、D1FIFOSEL) 以及控制寄存器 (CFIFOCTR、D0FIFOCTR、D1FIFOCTR) 构成。

各 FIFO 端口有以下特点:

- 通过 CFIFO 端口存取 DCP 的 FIFO 缓冲器。
- 通过 D0FIFO 或者 D1FIFO 端口, 进行 DMA 传送的 FIFO 缓冲器存取。
- CPU 能存取 D1FIFO 或者 D0FIFO 端口。
- 在使用 FIFO 端口的特有功能时, 不能更改 CURPIPE 位设定的管道号 (选择的管道) (使用 DMA 传送功能等)。
- 构成 FIFO 端口的寄存器群不影响其它 FIFO 端口。
- 不能将同一个管道分配给多个 FIFO 端口。
- FIFO 缓冲器状况有存取权在 CPU 侧和存取权在 SIE 侧的 2 种。当 FIFO 缓冲器的存取权在 SIE 侧时, CPU 不能存取。
- 表 21.2 中的 D0FIFO 端口寄存器和 D1FIFO 端口寄存器所记载的 H'FE40 0180 和 H'FE40 01C0 是用于 16/32 字节连续存取的地址。在 16/32 字节连续存取时, 必须以 32 位为单位进行, 此时能使用任意一种字节排序。

在对 CFIFO 进行 CPU 存取时, 必须存取 H'FE40 0014; 在对 D0FIFO/D1FIFO 进行周期挪用或者 CPU 存取时, 必须存取 H'FE40 0018 或者 H'FE40 001C。但是, 在对 C/D0/D1FIFO 以小端法并且不以 32 位为单位进行存取时, 地址会变。详细内容请参照表 21.8 ~ 表 21.10。

在上电复位时, 此寄存器被初始化。

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

位	位名	初始值	R/W	说明
31 ~ 0	FIFOPORT[31:0]	全 0	R/W	FIFO 端口 通过存取此位, 从 FIFO 缓冲器读接收数据或者给 FIFO 缓冲器写发送数据。 只在各控制寄存器 (CFIFOCTR、D0FIFOCTR 或者 D1FIFOCTR) 的 FRDY 位为 1 时才能存取此寄存器。 此寄存器的有效位因 MBW 位和 BIGEND 位的设定值而不同。有效位如表 21.8 ~ 表 21.10 所示。

表 21.8 32 位存取时的字节排序运行表

BIGEND 位	bit31 ~ 24	bit23 ~ 16	bit15 ~ 8	bit7 ~ 0	存取地址		
					CFIFO	D0FIFO	D1FIFO
0	地址 N+3	地址 N+2	地址 N+1	地址 N+0	H'FE40 0014	H'FE40 0018	H'FE40 001C
1	地址 N+0	地址 N+1	地址 N+2	地址 N+3			

表 21.9 16 位存取时的字节排序运行表

BIGEND 位	bit31 ~ 24	bit23 ~ 16	bit15 ~ 8	bit7 ~ 0	存取地址		
					CFIFO	D0FIFO	D1FIFO
0	写: 无效, 读: 禁止 *		地址 N+1	地址 N+0	H'FE40 0016	H'FE40 001A	H'FE40 001E
1	地址 N+0	地址 N+1	写: 无效, 读: 禁止 *		H'FE40 0014	H'FE40 0018	H'FE40 001C

【注】 * 禁止以字或者字节为单位读无效寄存器。

表 21.10 8 位存取时的字节排序运行表

BIGEND 位	bit31 ~ 24	bit23 ~ 16	bit15 ~ 8	bit7 ~ 0	存取地址		
					CFIFO	D0FIFO	D1FIFO
0	写: 无效, 读: 禁止 *		地址 N+0		H'FE40 0017	H'FE40 001B	H'FE40 001F
1	地址 N+0	写: 无效, 读: 禁止 *			H'FE40 0014	H'FE40 0018	H'FE40 001C

【注】 * 禁止以字或者字节为单位读无效寄存器。

21.3.8 FIFO 端口选择寄存器 (CFIFOSEL、D0FIFOSEL、D1FIFOSEL)

CFIFOSEL、D0FIFOSEL、D1FIFOSEL 选择分配给 FIFO 端口的管道，并且控制各 FIFO 端口的存取。

不能给 CFIFOSEL、D0FIFOSEL、D1FIFOSEL 的 CURPIPE 位指定同一个管道。在 D0FIFOSEL、D1FIFOSEL 寄存器的 CURPIPE 位为 B'000 时，无指定的管道。

在 DMA 传送的允许状态下，必能改变管道号。

在上电复位时，此寄存器被初始化。

(1) CFIFOSEL

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RCNT	REW	—	—	MBW[1:0]	—	BIGEND	—	—	ISEL	—	CURPIPE[3:0]				
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W*	R	R	R/W	R/W	R	R/W	R	R	R/W	R	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15	RCNT	0	R/W	读计数模式位 指定 CFIFOCTR 寄存器的 DTLN 位的读模式。 0: 在读完 CFIFO 的全部接收数据时将 DTLN 位清 0。 (在双缓冲器时, 只读完 1 个缓冲器面时) 1: 每读一个 CFIFO 接收数据时, 对 DTLN 位进行递减计数。
14	REW	0	R/W*	缓冲器指针复位 指定是否对缓冲器指针进行复位。 0: 对缓冲器指针进行复位 1: 不对缓冲器指针进行复位 在选择的管道为接收方向时, 如果在读 FIFO 缓冲器的过程中将此位置 1, 就能读取 FIFO 缓冲器的最初的数据 (在双缓冲器时, 对正在读取的 1 个缓冲器面, 能从最初的数据重新开始读取)。 不能同时设定 REW=1 和 CURPIPE 位的变更。必须在确认 FRDY=1 后设定 REW=1。 对于发送方向的管道, 从 FIFO 缓冲器的最初数据重新写时, 必须使用 BCLR 位。
13、12	—	全 0	R	保留位 读取值为 0, 只能写 0。
11、10	MBW[1:0]	00	R/W	CFIFO 端口存取长度位 指定 CFIFO 端口的存取长度。 00: 8 位 01: 16 位 10: 32 位 11: 禁止设定 在选择的管道为接收方向时, 如果在设定此位后开始读取, 就不能在读完全部数据前更改 MBW 位。必须同时设定 CURPIPE 位和 MBW 位。 在选择的管道为发送方向时, 不能在缓冲存储器的写处理过程中进行 8 位到 16 位 /32 位或者从 16 位到 32 位的存取长度的转换。 即使设定为 8 位 /16 位, 也能通过字节存取的控制进行奇数字节的写操作。
9	—	0	R	保留位 读取值为 0, 只能写 0。

位	位名	初始值	R/W	说明
8	BIGEND	0	R/W	CFIFO 端口字节排序控制位 指定 CFIFO 端口的字节排序。 0: 小端法 1: 大端法
7、6	—	全 0	R	保留位 读取值为 0, 只能写 0。
5	ISEL	0	R/W	DCP 时的 CFIFO 端口存取方向选择位 0: 选择读缓冲存储器 1: 选择写缓冲存储器 在选择的管道为 DCP 时, 如果要更改此位, 就必须在写此位后进行读取, 在确认写的值和读取值相同后进行下一个处理。 如果在存取 FIFO 缓冲器的过程中更改此位的设定, 就保持更改前的存取, 在恢复此位的设定后能继续进行存取。 此位必须和 CURPIPE 位同时设定。
4	—	0	R	保留位 读取值为 0, 只能写 0。
3 ~ 0	CURPIPE[3:0]	0000	R/W	CFIFO 端口存取管道指定位 设定通过 CFIFO 端口进行数据读写的管道号。 0000: DCP 0001: 管道 1 0010: 管道 2 0011: 管道 3 0100: 管道 4 0101: 管道 5 0110: 管道 6 0111: 管道 7 1000: 管道 8 1001: 管道 9 上述以外: 禁止设定 在更改此位时, 必须在写此位后进行读取, 在确认写的值和读取值相同后进行下一个处理。 不能给 CFIFOSEL 寄存器、D0FIFOSEL 寄存器和 D1FIFOSEL 寄存器的 CURPIPE 位设定相同的管道号。 如果在存取 FIFO 缓冲器的过程中更改此位的设定, 就保持更改前的存取, 在恢复此位的设定后能继续进行存取。

【注】 * 读取值为 0。

(2) D0FIFOSEL 和 D1FIFOSEL

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RCNT	REW	DCLRM	DREQE	MBW[1:0]	—	BIG END	—	—	—	—	—	CURPIPE[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	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	RCNT	0	R/W	<p>读计数模式位</p> <p>指定 DnFIFOCTR 寄存器的 DTLN 位的读模式。</p> <p>0: 在读完 DnFIFO 的全部接收数据时将 DTLN 位清 0。 (在双缓冲器时, 只读完 1 个缓冲器面时)</p> <p>1: 每读一个 DnFIFO 接收数据时, 就对 DTLN 位进行递减计数。 在将 BFRE 位置 1 后存取 DnFIFO 时, 必须将此位置 0。</p>
14	REW	0	R/W*1	<p>缓冲器指针复位</p> <p>指定是否对缓冲器指针进行复位</p> <p>0: 不对缓冲器指针进行复位</p> <p>1: 对缓冲器指针进行复位</p> <p>在选择的管道为接收方向时, 如果在读 FIFO 缓冲器的过程中将此位置 1, 就能读取 FIFO 缓冲器的最初的数据 (在双缓冲器时, 对正在读取的 1 个缓冲器面, 能从最初的数据重新开始读取)。</p> <p>不能同时设定 REW=1 和 CURPIPE 位的变更。必须在确认 FRDY=1 后设定 REW=1。</p> <p>如果在将 BFRE 位置 1 后存取 DnFIFO, 就不能在读完短包数据的状态下将此位置 1。</p> <p>对于发送方向的管道, 从 FIFO 缓冲器的最初数据重新写时, 必须使用 BCLR 位。</p>
13	DCLRM	0	R/W	<p>读取所选管道的数据后的缓冲存储器自动清除模式位</p> <p>在读取所选管道的数据后, 指定是禁止还是允许缓冲存储器自动清除。</p> <p>0: 禁止缓冲器自动清除</p> <p>1: 允许缓冲器自动清除</p> <p>如果将此位置 1, 就在分配给所选管道的 FIFO 缓冲器为空的空的状态下接收 Zero-Length packet 时或者在 BFRE=1 的状态下接收短包并且读完数据时, 给 FIFO 缓冲器设定 BCLR=1。</p> <p>在设定 BRDYM=1 后使用本模块时, 必须将此位置 0。</p>
12	DREQE	0	R/W	<p>DMA 传送请求允许位</p> <p>指定是禁止还是允许 DMA 传送请求。</p> <p>0: 禁止 DMA 传送请求</p> <p>1: 允许 DMA 传送请求</p> <p>在允许 DMA 传送请求时, 必须在设定 CURPIPE 位后将此位置 1。 必须在将此位置 0 后更改 CURPIPE 位的设定。</p>

位	位名	初始值	R/W	说明
11、10	MBW[1:0]	00	R/W	<p>FIFO 端口存取长度位 指定 DnFIFO 端口存取长度。</p> <p>00: 8 位 01: 16 位 10: 32 位 11: 禁止设定</p> <p>在选择的管道为接收方向时, 如果在设定此位后开始读取, 就不能在读完全部数据前更改 MBW 位。必须同时设定 CURPIPE 位和 MBW 位。 在选择管道为发送方向时, 不能在缓冲存储器的写处理过程中进行 8 位到 16 位 /32 位或者 16 位到 32 位的存取长度的转换。 即使设定为 8 位 /16 位, 也能通过字节存取的控制进行奇数字节的写操作。</p>
9	—	0	R	<p>保留位 读取值为 0, 只能写 0。</p>
8	BIGEND	0	R/W	<p>FIFO 端口字节排序控制位 指定 DnFIFO 端口的字节排序。</p> <p>0: 小端法 1: 大端法</p>
7 ~ 4	—	全 0	R	<p>保留位 读取值为 0, 只能写 0。</p>
3 ~ 0	CURPIPE[3:0]	0000	R/W	<p>FIFO 端口存取管道指定位 设定通过 D0FIFO/D1FIFO 端口进行数据读写的管道号。</p> <p>0000: 无指定 0001: 管道 1 0010: 管道 2 0011: 管道 3 0100: 管道 4 0101: 管道 5 0110: 管道 6 0111: 管道 7 1000: 管道 8 1001: 管道 9 上述以外: 禁止设定</p> <p>在更改此位时, 必须在写此位后进行读取, 在确认写的值和读取值相同后进行下一个处理。 不能给 CFIFOSEL 寄存器、D0FIFOSEL 寄存器和 D1FIFOSEL 寄存器的 CURPIPE 位设定相同的管道号。 如果在存取 FIFO 缓冲器的过程中更改此位的设定, 就保持更改前的存取, 在恢复此位的设定后能继续进行存取。</p>

【注】 *1 读取值为 0。

21.3.9 FIFO 端口控制寄存器 (CFIFOCTR、D0FIFOCTR、D1FIFOCTR)

CFIFOCTR、D0FIFOCTR、D1FIFOCTR 设定是否写完缓冲存储器、CPU 是否清除缓冲器以及能否存取 FIFO 端口的寄存器。此寄存器支持各 FIFO 端口。

在上电复位时，此寄存器被初始化。

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

位	位名	初始值	R/W	说明
15	BVAL	0	R/W*2	<p>缓冲存储器有效标志位</p> <p>在结束对 CURPIPE 所选管道的 CPU 侧的 FIFO 缓冲器的写操作时，将此位置 1。</p> <p>0: 无效 1: 写完</p> <p>在选择的管道为发送方向时，必须在以下情况下将此位置 1。本模块将 FIFO 缓冲器从 CPU 侧切换到 SIE 侧，并且设定为可发送状态。</p> <ul style="list-style-type: none"> 在发送短包时，必须在写完数据时将此位置 1。 在发送 Zero-Length 包时，必须在对 FIFO 缓冲器写数据前将此位置 1。 必须在给连续传送模式的管道写 MaxPacketSize 的自然数倍数并且小于 BufferSize 的数据后将此位置 1 <p>如果对连续传送模式的管道写 MaxPacketSize 的数据，就将此位置 1，将 FIFO 缓冲器从 CPU 侧转换到 SIE 侧，并且设定为可发送状态。只在 FRDY=1 时才能将此位置 1。</p> <p>在选择的管道为接收方向时，不能将此位置 1。</p>
14	BCLR	0	R/W*1	<p>CPU 缓冲器清除位</p> <p>在清除所选管道的 CPU 侧的 FIFO 缓冲器时，将此位置 1。</p> <p>0: 无效 1: 清除 CPU 侧缓冲存储器</p> <p>在将分配给所选管道的 FIFO 缓冲器设定为双缓冲器时，即使 2 个缓冲器面都为可读状态，也只清除一个 FIFO 缓冲器面。</p> <p>在选择的管道为 DCP 时，不论 FIFO 缓冲器在 CPU 侧还是在 SIE 侧，通过设定 BCLR=1 清除 FIFO 缓冲器。在清除 SIE 侧的缓冲器时，必须在将 DCP 的 PID 位设定为 NAK 后设定 BCLR=1。</p> <p>在选择的管道为发送方向时，如果同时将 BVAL 位和 BCLR 位置 1，就清除以前写的的数据，并将 Zero-Length 包设定为可发送状态。</p> <p>在选择的管道不为 DCP 时，只在 FRDY=1 时才能将此位置 1。</p>

位	位名	初始值	R/W	说明
13	FRDY	0	R	FIFO 端口就绪位 表示 CPU (DMAC) 能否存取 FIFO 端口。 0: 不能存取 FIFO 端口 1: 能存取 FIFO 端口 在以下情况下 FRDY=1, 但是因为没有应该读取的数据, 所以不能从 FIFO 端口读取数据。在这种情况下, 必须在设定 BCLR=1 后清除 FIFO 缓冲器, 并且设定为能发送 / 接收下一个数据的状态。 <ul style="list-style-type: none"> 在分配给所选管道的 FIFO 缓冲器为空的状态下接收 Zero-Length 包。 在 BFRE=1 时接收短包并且读完数据。
12	—	0	R	保留位 读取值为 0, 只能写 0。
11 ~ 0	DTLN[11:0]	H'000	R	接收数据长度位 表示接收数据的长度。 在读 FIFO 缓冲器的过程中, 此位的值因 RCNT 位的设定值而发生如下的变化: <ul style="list-style-type: none"> RCNT=0 时: 在 CPU (DMAC) 读完一个 FIFO 缓冲器面的接收数据前, 用此位表示接收数据的长度。 在 BFRE=1 时, 即使读操作结束, 也在设定 BCLR=1 前保持接收数据的长度。 RCNT=1 时: 每次读取时, 本模块就对 DTLN 位的值进行递减计数。(在 MBW=0 时 -1, 在 MBW=1 时 -2) 在读取一个 FIFO 缓冲器面时 DTLN=0。但是, 当设定为双缓冲器并且在读取一个 FIFO 缓冲器面的接收数据前另一个 FIFO 缓冲器面接收结束时, 就在前 1 个面读取结束后 DTLN 位表示后 1 个面的接收数据长度。在 RCNT=1 时, 如果在读 FIFO 缓冲器的过程中读此位的值, 本模块就在 FIFO 端口读周期后的 150ns 内表示此位的更新值。

【注】 *1 读取值为 0, 只能写 1。

*2 只能写 1。

21.3.10 中断允许寄存器 0 (INTENB0)

INTENB0 指定各中断的屏蔽。本模块在检测到此寄存器中通过软件置 1 的位所对应的中断时, 产生 USB 中断。

与此寄存器的设定值 (禁止 / 允许中断通知) 无关, 在满足各中断源的检测条件时, 本模块将 INTSTS0 寄存器的对应状态位置 1。

在各中断源对应的 INTSTS0 寄存器的状态位为 1 的状态下, 如果通过软件将此寄存器对应的中断允许位从 0 变为 1, 本模块就产生 USB 中断。

在上电复位时, 此寄存器被初始化。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15	VBSE	0	R/W	VBUS 中断允许位 在检测到 VBINT 中断时, 指定是禁止还是允许 USB 的中断输出。 0: 禁止中断输出 1: 允许中断输出
14	RSME	0	R/W	恢复中断允许位 * 在检测到 RESM 中断时, 指定是禁止还是允许 USB 的中断输出。 0: 禁止中断输出 1: 允许中断输出
13	SOFE	0	R/W	允许帧号更新中断允许位 在检测到 SOFR 中断时, 指定是禁止还是允许 USB 的中断输出。 0: 禁止中断输出 1: 允许中断输出
12	DVSE	0	R/W	设备状态转移中断允许位 * 在检测到 DVST 中断时, 指定是禁止还是允许 USB 的中断输出。 0: 禁止中断输出 1: 允许中断输出
11	CTRE	0	R/W	控制传送阶段转移中断允许位 * 在检测到 CTRT 中断时, 指定是禁止还是允许 USB 的中断输出。 0: 禁止中断输出 1: 允许中断输出
10	BEMPE	0	R/W	缓冲器空中断允许位 在检测到 BEMP 中断时, 指定是禁止还是允许 USB 的中断输出。 0: 禁止中断输出 1: 允许中断输出
9	NRDYE	0	R/W	缓冲器未就绪应答中断允许位 在检测到 NRDY 中断时, 指定是禁止还是允许 USB 的中断输出。 0: 禁止中断输出 1: 允许中断输出
8	BRDYE	0	R/W	缓冲器就绪中断允许位 在检测到 BRDY 中断时, 指定是禁止还是允许 USB 的中断输出。 0: 禁止中断输出 1: 允许中断输出
7 ~ 0	—	全 0	R	保留位 读取值为 0, 只能写 0。

【注】 * 只在选择功能控制器功能时才能设定 RSME 位、DVSE 位和 CTRE 位; 在选择主机控制器功能时, 不能允许中断。

21.3.11 中断允许寄存器 1 (INTENB1)

INTENB1 设定选择主机控制器功能时的中断屏蔽。

本模块在检测到此寄存器中通过软件置 1 的位所对应的中断时，产生 USB 中断。

与此寄存器的设定值（禁止 / 允许中断通知）无关，在满足各中断源的检测条件时，本模块将 INTSTS1 寄存器的对应状态位置 1。

在各中断源对应的 INTSTS1 寄存器的状态位为 1 的状态下，如果通过软件将此寄存器对应的中断允许位从 0 变为 1，本模块就产生 USB 中断。

在选择功能控制器功能时，不能允许中断。

在上电复位时，此寄存器被初始化。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	BCHGE	—	DTCHE	ATT CHE	—	—	—	—	EOF ERRE	SIGNE	SACKE	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R/W	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 读取值为 0，只能写 0。
14	BCHGE	0	R/W	USB 总线变化中断允许位 在检测到 BCHG 中断时，指定是禁止还是允许 USB 的中断输出。 0: 禁止中断输出 1: 允许中断输出
13	—	0	R	保留位 读取值为 0，只能写 0。
12	DTCHE	0	R/W	断开检测中断允许位 在检测到 DTCH 中断时，指定是禁止还是允许 USB 的中断输出。 0: 禁止中断输出 1: 允许中断输出
11	ATT CHE	0	R/W	连接检测中断允许位 在检测到 ATT CHE 中断时，指定是禁止还是允许 USB 的中断输出。 0: 禁止中断输出 1: 允许中断输出
10 ~ 7	—	全 0	R	保留位 读取值为 0，只能写 0。
6	EOFERRE	0	R/W	EOF 错误检测中断允许位 在检测到 EOFERR 中断时，指定是禁止还是允许 USB 的中断输出。 0: 禁止中断输出 1: 允许中断输出
5	SIGNE	0	R/W	设置事务错误中断允许位 在检测到 SIGN 中断时，指定是禁止还是允许 USB 的中断输出。 0: 禁止中断输出 1: 允许中断输出
4	SACKE	0	R/W	设置事务正常应答中断允许位 在检测到 SACK 中断时，指定是禁止还是允许 USB 的中断输出。 0: 禁止中断输出 1: 允许中断输出

位	位名	初始值	R/W	说明
3 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

【注】 只在选择主机控制器功能时才能通过 INTENB1 寄存器设定中断允许；在选择功能控制器功能时，不能允许中断。

21.3.12 BRDY 中断允许寄存器 (BRDYENB)

在检测到各管道的 BRDY 中断时，BRDYENB 指定是禁止还是允许将 INTSTS0 寄存器的 BRDY 位置 1。

对于此寄存器中通过软件置 1 的管道，本模块在检测到 BRDY 中断时将 BRDYSTS 寄存器的 PIPEBRDY 位所对应的位置 1，INTSTS0 寄存器的 BRDY 位置 1，并且产生 BRDY 中断。

在 BRDYSTS 寄存器的 PIPEBRDY 位中至少有 1 位为 1 的状态下，如果通过软件将此寄存器对应的中断允许位从 0 变为 1，本模块就产生 BRDY 中断。

在上电复位时，此寄存器被初始化。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIPE9 BRDYE	PIPE8 BRDYE	PIPE7 BRDYE	PIPE6 BRDYE	PIPE5 BRDYE	PIPE4 BRDYE	PIPE3 BRDYE	PIPE2 BRDYE	PIPE1 BRDYE	PIPE0 BRDYE
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 10	—	全 0	R	保留位 读取值为 0，只能写 0。
9	PIPE9BRDYE	0	R/W	管道 9 的 BRDY 中断允许位 0: 禁止中断输出 1: 允许中断输出
8	PIPE8BRDYE	0	R/W	管道 8 的 BRDY 中断允许位 0: 禁止中断输出 1: 允许中断输出
7	PIPE7BRDYE	0	R/W	管道 7 的 BRDY 中断允许位 0: 禁止中断输出 1: 允许中断输出
6	PIPE6BRDYE	0	R/W	管道 6 的 BRDY 中断允许位 0: 禁止中断输出 1: 允许中断输出
5	PIPE5BRDYE	0	R/W	管道 5 的 BRDY 中断允许位 0: 禁止中断输出 1: 允许中断输出
4	PIPE4BRDYE	0	R/W	管道 4 的 BRDY 中断允许位 0: 禁止中断输出 1: 允许中断输出
3	PIPE3BRDYE	0	R/W	管道 3 的 BRDY 中断允许位 0: 禁止中断输出 1: 允许中断输出
2	PIPE2BRDYE	0	R/W	管道 2 的 BRDY 中断允许位 0: 禁止中断输出 1: 允许中断输出

位	位名	初始值	R/W	说明
1	PIPE1BRDYE	0	R/W	管道 1 的 BRDY 中断允许位 0: 禁止中断输出 1: 允许中断输出
0	PIPE0BRDYE	0	R/W	管道 0 的 BRDY 中断允许位 0: 禁止中断输出 1: 允许中断输出

21.3.13 NRDY 中断允许寄存器 (NRDYENB)

在检测到各管道的 NRDY 中断时，NRDYENB 指定是禁止还是允许将 INTSTS0 寄存器的 NRDY 位置 1。

对于此寄存器中通过软件置 1 的管道，本模块在检测到 NRDY 中断源时将 NRDYSTS 寄存器的 PIPENRDY 位对应的位置 1，INTSTS0 寄存器的 NRDY 位置 1，并且产生 NRDY 中断。

在 NRDYSTS 寄存器的 PIPENRDY 位中至少有 1 位为 1 的状态下，如果通过软件将此寄存器对应的中断允许位从 0 变为 1，本模块就产生 NRDY 中断。

在上电复位时，此寄存器被初始化。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIPE9 NRDYE	PIPE8 NRDYE	PIPE7 NRDYE	PIPE6 NRDYE	PIPE5 NRDYE	PIPE4 NRDYE	PIPE3 NRDYE	PIPE2 NRDYE	PIPE1 NRDYE	PIPE0 NRDYE
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 10	—	全 0	R	保留位 读取值为 0，只能写 0。
9	PIPE9NRDYE	0	R/W	管道 9 的 NRDY 中断允许位 0: 禁止中断输出 1: 允许中断输出
8	PIPE8NRDYE	0	R/W	管道 8 的 NRDY 中断允许位 0: 禁止中断输出 1: 允许中断输出
7	PIPE7NRDYE	0	R/W	管道 7 的 NRDY 中断允许位 0: 禁止中断输出 1: 允许中断输出
6	PIPE6NRDYE	0	R/W	管道 6 的 NRDY 中断允许位 0: 禁止中断输出 1: 允许中断输出
5	PIPE5NRDYE	0	R/W	管道 5 的 NRDY 中断允许位 0: 禁止中断输出 1: 允许中断输出
4	PIPE4NRDYE	0	R/W	管道 4 的 NRDY 中断允许位 0: 禁止中断输出 1: 允许中断输出
3	PIPE3NRDYE	0	R/W	管道 3 的 NRDY 中断允许位 0: 禁止中断输出 1: 允许中断输出

位	位名	初始值	R/W	说明
2	PIPE2NRDYE	0	R/W	管道 2 的 NRDY 中断允许位 0: 禁止中断输出 1: 允许中断输出
1	PIPE1NRDYE	0	R/W	管道 1 的 NRDY 中断允许位 0: 禁止中断输出 1: 允许中断输出
0	PIPE0NRDYE	0	R/W	管道 0 的 NRDY 中断允许位 0: 禁止中断输出 1: 允许中断输出

21.3.14 BEMP 中断允许寄存器 (BEMPENB)

在检测到各管道的 BEMP 中断时，BEMPENB 指定是禁止还是允许将 INTSTS0 寄存器的 BEMP 位置 1。

对于此寄存器中通过软件置 1 的管道，本模块在检测到 BEMP 中断源时将 BEMPSTS 寄存器的 PIPEBEMP 位对应的位置 1，INTSTS0 寄存器的 BEMP 位置 1，并且产生 BEMP 中断。

在 BEMPSTS 寄存器的 PIPEBEMP 位中至少有 1 位为 1 的状态下，如果通过软件将此寄存器对应的中断允许位从 0 变为 1，本模块就产生 BEMP 中断。

在上电复位时，此寄存器被初始化。

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

位	位名	初始值	R/W	说明
15 ~ 10	—	全 0	R	保留位 读取值为 0，只能写 0。
9	PIPE9BEMPE	0	R/W	管道 9 的 BEMP 中断允许位 0: 禁止中断输出 1: 允许中断输出
8	PIPE8BEMPE	0	R/W	管道 8 的 BEMP 中断允许位 0: 禁止中断输出 1: 允许中断输出
7	PIPE7BEMPE	0	R/W	管道 7 的 BEMP 中断允许位 0: 禁止中断输出 1: 允许中断输出
6	PIPE6BEMPE	0	R/W	管道 6 的 BEMP 中断允许位 0: 禁止中断输出 1: 允许中断输出
5	PIPE5BEMPE	0	R/W	管道 5 的 BEMP 中断允许位 0: 禁止中断输出 1: 允许中断输出
4	PIPE4BEMPE	0	R/W	管道 4 的 BEMP 中断允许位 0: 禁止中断输出 1: 允许中断输出

位	位名	初始值	R/W	说明
3	PIPE3BEMPE	0	R/W	管道 3 的 BEMP 中断允许位 0: 禁止中断输出 1: 允许中断输出
2	PIPE2BEMPE	0	R/W	管道 2 的 BEMP 中断允许位 0: 禁止中断输出 1: 允许中断输出
1	PIPE1BEMPE	0	R/W	管道 1 的 BEMP 中断允许位 0: 禁止中断输出 1: 允许中断输出
0	PIPE0BEMPE	0	R/W	管道 0 的 BEMP 中断允许位 0: 禁止中断输出 1: 允许中断输出

21.3.15 SOF 控制寄存器 (SOFCFG)

SOFCFG 指定事务的有效期间和 BRDY 中断状态清除时序等。

在上电复位时，此寄存器被初始化。

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

位	位名	初始值	R/W	说明
15 ~ 9	—	全 0	R	保留位 读取值为 0，只能写 0。
8	TRNENSEL	0	R/W	事务有效期限转换位 对于全速或者低速通过程中的端口，指定 1 帧中的发行标记的期间（事务有效期间） 0: 不支持低速 1: 支持低速 此位只在选择主机控制器功能时有效。即使在选择主机控制器功能时，也不影响高速的事务有效期间。 在选择功能控制器功能时，必须将此位置 0。
7	—	全 0	R	保留位 读取值为 0，只能写 0。
6	BRDYM	0	R/W	各管道的 BRDY 中断状态清除时序设定位 指定清除各管道的 BRDY 中断状态的时序。 0: 软件清除 1: 在读写 FIFO 缓冲器时清除状态
5	—	0*	R	保留位 此位为保留位。只能写刚读取的值。 【注】 上电复位后的初始值为 0，必须通过本模块的初始化程序将此位置 1。
4 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

【注】 * 上电复位后的初始值为 0，必须通过本模块的初始化程序将此位置 1。

21.3.16 中断状态寄存器 0 (INTSTS0)

INTSTS0 表示各检测到的中断的状态。

在上电复位时，此寄存器被初始化；在 USB 总线复位时，DVSQ[2:0] 位被初始化。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ[2:0]			VALID	CTSQ[2:0]		
初始值:	0	0	0	0	0	0	0	0	*3	*2	*2	*2	0	0	0	0
R/W:	R/W*7	R/W*7	R/W*7	R/W*7	R/W*7	R	R	R	R	R	R	R	R/W*7	R	R	R

位	位名	初始值	R/W	说明
15	VBINT	0	R/W*7	VBUS 中断状态位 *4*5 0: 未发生 VBUS 中断 1: 发生 VBUS 中断 在检测到 VBUS 引脚输入值的变化 (从高电平变为低电平或者从低电平变为高电平) 时, 将此位置 1。VBSTS 位表示 VBUS 引脚的输入值。在发生 VBINT 中断时, 必须通过软件重复读取 VBSTS 位, 直到读取值数次相同为止, 并且消除颤动。
14	RESM	0	R/W*7	恢复中断状态位 *4*5*6 0: 未发生恢复中断 1: 发生恢复中断 在设定功能控制器功能时, 如果在挂起状态 (DVSQ=1XX) 下检测到 DP 引脚的下降沿, 就将此位置 1。 在选择主机控制器功能时, 读取值无效。
13	SOFR	0	R/W*7	帧号更新中断状态位 *4 0: 未发生 SOF 中断 1: 发生 SOF 中断 1. 设定主机控制器功能时 在通过软件将 UACT 位置 1 的情况下, 在更新帧号时将此位置 1 (每隔 1ms 检测一次此中断。) 2. 设定功能控制器功能时 在更新帧号时, 将此位置 1 (每隔 1ms 检测一次此中断。) 在从 USB 主机发出的 SOF 包破损时, 本模块通过内插来检测 SOFR 中断。
12	DVST	0/1*1	R/W*7	设备状态转移中断状态位 *4*6 0: 未发生设备状态转移中断 1: 发生设备状态转移中断 在设定功能控制器功能时, 如果检测到设备状态的变化, 就更新 DVSQ 值, 并且将此位置 1。 在发生此中断时, 必须在检测到设备状态转到下一个状态前清除状态。 在选择主机控制器功能时, 读取值无效。
11	CTRT	0	R/W*7	控制传送阶段转移中断状态位 *4*6 0: 未产生控制传送阶段转移中断 1: 产生控制传送阶段转移中断 在设定功能控制器功能时, 如果检测到控制传送的阶段转移, 就更新 CTSQ 值, 并且将此位置 1。 在发生此中断时, 必须在检测到控制传送转移到下一个阶段前清除状态。 选择主机控制器功能时, 读取值无效。

位	位名	初始值	R/W	说明
10	BEMP	0	R	<p>缓冲器空中断状态位</p> <p>0: 未发生 BEMP 中断 1: 发生 BEMP 中断</p> <p>在 BEMPENB 寄存器的 PIPEBEMPE 位为 1 的管道对应的 BEMPSTS 寄存器的 PIPEBEMP 位中至少有 1 位为 1 时 (至少对软件允许 BEMP 中断通知的管道中的一个管道, 检测到 BEMP 中断状态时), 将此位置 1。</p> <p>有关 PIPEBEMP 状态的指定条件, 请参照 “21.4.2(3) BEMP 中断”。</p> <p>如果通过软件将由 PIPEBEMPE 位设定为允许的管道对应的全部 PIPEBEMP 位置 0, 就将此位清 0。</p> <p>即使通过软件给此位写 0, 也无法将此位清 0。</p>
9	NRDY	0	R	<p>缓冲器未就绪中断状态位</p> <p>0: 未发生 NRDY 中断 1: 发生 NRDY 中断</p> <p>在 NRDYENB 寄存器的 PIPENRDYE 位为 1 的管道对应的 NRDYSTS 寄存器的 PIPENRDY 位中至少有 1 位为 1 时 (至少对软件允许 NRDY 中断通知的管道中的一个管道, 检测到 NRDY 中断状态时), 将此位置 1。</p> <p>有关 PIPENRDY 状态的指定条件, 请参照 “21.4.2(2) NRDY 中断”。</p> <p>如果通过软件将由 PIPENRDYE 位设定为允许的管道对应的全部 PIPENRDY 位置 0, 就将此位清 0。</p> <p>即使通过软件给此位写 0, 也无法将此位清 0。</p>
8	BRDY	0	R	<p>缓冲器就绪中断状态位</p> <p>表示 BRDY 的中断状态。</p> <p>0: 未发生 BRDY 中断 1: 发生 BRDY 中断</p> <p>在 BRDYENB 寄存器的 PIPEBRDYE 位为 1 的管道对应的 BRDYSTS 寄存器的 PIPEBRDY 位中至少有 1 位为 1 时 (至少对软件允许 BRDY 中断通知的管道中的一个管道, 检测到 BRDY 中断状态时), 将此位置 1。</p> <p>有关 PIPEBRDY 状态的指定条件, 请参照 “21.4.2(1) BRDY 中断”。</p> <p>如果通过软件将由 PIPEBRDYE 位设定为允许的管道对应的全部 PIPEBRDY 位置 0, 就将此位清 0。</p> <p>即使通过软件给此位写 0, 也无法将此位清 0。</p>
7	VBSTS	0/1*3	R	<p>VBUS 输入状态位</p> <p>0: VBUS 引脚为低电平 1: VBUS 引脚为高电平</p>
6 ~ 4	DVSQ[2:0]	000/001*2	R	<p>设备状态位</p> <p>000: 供电状态 001: 默认状态 010: 地址状态 011: 配置状态 1xx: 挂起状态</p> <p>在选择主机控制器功能时, 读取值无效。</p>

位	位名	初始值	R/W	说明
3	VALID	0	R/W*7	USB 请求接收位 0: 未检测到 1: 接收设置包 在选择主机控制器功能时, 读取值无效。
2 ~ 0	CTSQ[2:0]	000	R	控制传送阶段位 000: 空闲或者设置阶段 001: 控制读数据阶段 010: 控制读状态阶段 011: 控制写数据阶段 100: 控制写状态阶段 101: 控制写 (NoData) 状态阶段 110: 控制传送顺序错误 111: 禁止设定 在选择主机控制器功能时, 读取值无效。

【注】 *1 在上电复位时为 B'0；在 USB 总线复位时为 B'1。

*2 在上电复位时为 B'000；在 USB 总线复位时为 B'001。

*3 在 VBUS 引脚为高电平时为 1，在低电平时为 0。

*4 在清除 VBINT 位、RESM 位、SOFR 位、DVST 位或者 CTRT 位时，必须给要清除的位写 0 而给其他位写 1。不能给 0 状态的位写 0。

*5 在时钟停止时 (SCKE=0)，如果检测 VBINT 位和 RESM 位所示的状态发生变化，并且对应的中断为允许状态，就通知中断。必须在允许时钟后，通过软件清除状态。

*6 RESM 位、DVST 位和 CTRT 位的状态只在选择功能控制器功能时发生变化。在选择功能控制器功能时，必须禁止 (0) 对应的中断允许位。

*7 只能写 0。

21.3.17 中断状态寄存器 1 (INTSTS1)

INTSTS1 为确认各中断状态的寄存器。

在选择功能控制器功能时，使用 INTSTS0 寄存器；在选择主机控制器功能时，使用 INTSTS1 寄存器。能通过参照其中一个寄存器确认中断的发生。

只有在选择主机控制器功能时，才能允许此寄存器的各位所示的状态变化中断。

在上电复位时，此寄存器被初始化。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	BCHG	—	DTCH	ATTCH	—	—	—	—	EOF ERR	SIGN	SACK	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W*1	R	R/W*1	R/W*1	R	R	R	R	R/W*1	R/W*1	R/W*1	R	R	R	R

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读取值为 0，只能写 0。

位	位名	初始值	R/W	说明
14	BCHG	0	R/W*1	<p>USB 总线变化中断状态位</p> <p>表示 USB 总线变化的中断状态。</p> <p>0: 未发生 BCHG 中断</p> <p>1: 发生 BCHG 中断</p> <p>在 USB 端口发生全速 / 低速信号电平的状态变化 (从 J-State、K-State 或者 SE0 的任意一种状态变为 J-State、K-State 或者 SE0 任意一种状态) 时, 本模块检测 BCHG 中断, 并且将此位置 1。此时, 如果已通过软件将对应的中断允许位置 1, 就产生中断。</p> <p>SYSSTS0 寄存器的 LNST 位表示 USB 端口当前的输入状态。在 BCHG 引脚发生中断时, 通过软件重复读取 LNST 位, 直到读取值数次相同为止, 消除颤动。</p> <p>即使在内部时钟为停止的状态下也检测 USB 总线变化。</p> <p>在选择功能控制器功能时, 读取值无效。</p>
13	—	0	R	<p>保留位</p> <p>读取值为 0, 只能写 0。</p>
12	DTCH	0	R/W*1	<p>USB 断开检测中断状态位</p> <p>在选择主机控制器功能时, 表示 USB 断开检测的中断状态。</p> <p>0: 未发生 DTCH 中断</p> <p>1: 发生 DTCH 中断</p> <p>在检测到 USB 总线断开时, 本模块检测 DTCH 中断, 并且将此位置 1。此时, 如果已通过软件将对应的中断允许位置 1, 就产生中断。</p> <p>本模块按照 USB 2.0 规格的基准检测总线的断开。</p> <p>在检测到 DTCH 中断之后 (与该中断允许位的设定值无关), 本模块进行以下的硬件控制, 必须通过软件让正在与 USB 端口进行通信的管道全部停止, 并且转移到 USB 端口的连接 (产生 ATTCH 中断) 等待状态。</p> <ul style="list-style-type: none"> • 将检测到 DTCH 中断的端口的 UACT 位更改为 0。 • 将产生 DTCH 中断的端口转移到空闲状态。 <p>在选择功能控制器功能时, 读取值无效。</p>
11	ATTCH	0	R/W*1	<p>ATTCH 中断状态位</p> <p>在选择主机控制器功能时, 表示 ATTCH 的中断状态。</p> <p>0: 未发生 ATTCH 中断</p> <p>1: 发生 ATTCH 中断</p> <p>如果在端口检测到 2.5μs 的全速 / 低速信号电平的 J-State 或者 K-State, 本模块就检测 ATTCH 中断, 并且将此位置 1。此时, 如果已通过软件将对应的中断允许位置 1, 就产生中断。</p> <p>本模块的 ATTCH 中断检测条件如下:</p> <ul style="list-style-type: none"> • 从 K-State、SE0 或者 SE1 变为 J-State, 并且 J-State 持续保持 2.5μs。 • 从 J-State、SE0 或者 SE1 变为 K-State, 并且 K-State 持续保持 2.5μs。 <p>在选择功能控制器功能时, 读取值无效。</p>
10 ~ 7	—	全 0	R	<p>保留位</p> <p>读取值为 0, 只能写 0。</p>

位	位名	初始值	R/W	说明
6	EOFERR	0	R/W*1	<p>EOF 错误检测中断状态位</p> <p>在选择主机控制器功能时，表示 EOFERR 的中断状态。</p> <p>0: 未发生 EOFERR 中断</p> <p>1: 发生 EOFERR 中断</p> <p>如果在 USB 2.0 规格所规定的 EOF2 时检测到通信未结束，本模块就检测 EOFERR 中断，并且将此位置 1。此时，如果已通过软件将对应的中断允许位置 1，就产生 EOFERR 中断。</p> <p>在检测到 EOFERR 中断后（与该中断允许位的设定值无关），本模块进行以下的硬件控制，必须通过软件让正在与 USB 端口进行通信的管道全部停止，并且重新对 USB 端口进行 Enumeration。</p> <ul style="list-style-type: none"> 将检测到 EOFERR 中断的端口的 UACT 位更改 0。 将产生 EOFERR 中断的端口转移到空闲状态。 <p>在选择功能控制器功能时，读取值无效。</p>
5	SIGN	0	R/W*1	<p>设置事务错误中断状态位</p> <p>在选择主机控制器功能时，表示设置事务错误的中断状态。</p> <p>0: 未发生 SIGN 中断</p> <p>1: 发生 SIGN 中断</p> <p>对本模块发行的 SETUP 事务，如果外围设备连续 3 次没有 ACK 应答，本模块就检测 SIGN 中断，并且将此位置 1。此时，如果已通过软件将对应的中断允许位置 1，就产生 SIGN 中断。</p> <p>本模块的 SIGN 中断检测条件是对 3 次连续的 SETUP 事务进行以下任意一种应答时的情况：</p> <ul style="list-style-type: none"> 在外围设备没有任何应答的状态下，检测到超时 ACK 包破损 接收到 ACK 以外的握手信号（NAK、NYET 或者 STALL） <p>在选择功能控制器功能时，读取值无效。</p>
4	SACK	0	R/W*1	<p>设置事务正常应答中断状态位</p> <p>在选择主机控制器功能时，表示设置事务正常应答的中断状态。</p> <p>0: 未发生 SACK 中断</p> <p>1: 发生 SACK 中断</p> <p>对本模块发行的 SETUP 事务，如果接收到外围设备的 ACK 应答，本模块就检测 SACK 中断，并且将此位置 1。此时，如果已通过软件将对应的中断允许位置 1，就产生 SACK 中断。</p>
3 ~ 0	—	全 0	R	<p>保留位</p> <p>读取值为 0，只能写 0。</p>

【注】 *1 在清除此寄存器的各位所示的状态时，只能给要清除的位写 0 而给其他位写 1。

*2 在时钟停止时（SCKE=0），如果检测到 BCHG 位所示的状态变化，并且对应的中断为允许状态，就通知中断。必须在允许时钟后通过软件清除状态。在时钟停止时（SCKE=0），不检测 BCHG 位以外的中断。

21.3.18 BRDY 中断状态寄存器 (BRDYSTS)

BRDYSTS 表示各管道的 BRDY 中断状态。

在上电复位时，此寄存器被初始化。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIPE9 BRDY	PIPE8 BRDY	PIPE7 BRDY	PIPE6 BRDY	PIPE5 BRDY	PIPE4 BRDY	PIPE3 BRDY	PIPE2 BRDY	PIPE1 BRDY	PIPE0 BRDY
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

位	位名	初始值	R/W	说明
15 ~ 10	—	全 0	R	保留位 读取值为 0，只能写 0。
9	PIPE9BRDY	0	R/W*1	管道 9 的 BRDY 中断状态位 *2 0: 未发生中断 1: 发生中断
8	PIPE8BRDY	0	R/W*1	管道 8 的 BRDY 中断状态位 *2 0: 未发生中断 1: 发生中断
7	PIPE7BRDY	0	R/W*1	管道 7 的 BRDY 中断状态位 *2 0: 未发生中断 1: 发生中断
6	PIPE6BRDY	0	R/W*1	管道 6 的 BRDY 中断状态位 *2 0: 未发生中断 1: 发生中断
5	PIPE5BRDY	0	R/W*1	管道 5 的 BRDY 中断状态位 *2 0: 未发生中断 1: 发生中断
4	PIPE4BRDY	0	R/W*1	管道 4 的 BRDY 中断状态位 *2 0: 未发生中断 1: 发生中断
3	PIPE3BRDY	0	R/W*1	管道 3 的 BRDY 中断状态位 *2 0: 未发生中断 1: 发生中断
2	PIPE2BRDY	0	R/W*1	管道 2 的 BRDY 中断状态位 *2 0: 未发生中断 1: 发生中断
1	PIPE1BRDY	0	R/W*1	管道 1 的 BRDY 中断状态位 *2 0: 未发生中断 1: 发生中断
0	PIPE0BRDY	0	R/W*1	管道 0 的 BRDY 中断状态位 *2 0: 未发生中断 1: 发生中断

【注】 *1 在 BRDYM=0 时，如果清除此寄存器的各位所示的状态，就只能给要清除的位写 0 而给其他位写 1。

*2 在 BRDYM=0 时，必须在存取 FIFO 前清除此中断。

21.3.19 NRDY 中断状态寄存器 (NRDYSTS)

NRDYSTS 表示各管道的 NRDY 中断状态。

在上电复位时，此寄存器被初始化。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIPE9 NRDY	PIPE8 NRDY	PIPE7 NRDY	PIPE6 NRDY	PIPE5 NRDY	PIPE4 NRDY	PIPE3 NRDY	PIPE2 NRDY	PIPE1 NRDY	PIPE0 NRDY
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

位	位名	初始值	R/W	说明
15 ~ 10	—	全 0	R	保留位 读取值为 0，只能写 0。
9	PIPE9NRDY	0	R/W*	管道 9 的 NRDY 中断状态位 0: 未发生中断 1: 发生中断
8	PIPE8NRDY	0	R/W*	管道 8 的 NRDY 中断状态位 0: 未发生中断 1: 发生中断
7	PIPE7NRDY	0	R/W*	管道 7 的 NRDY 中断状态位 0: 未发生中断 1: 发生中断
6	PIPE6NRDY	0	R/W*	管道 6 的 NRDY 中断状态位 0: 未发生中断 1: 发生中断
5	PIPE5NRDY	0	R/W*	管道 5 的 NRDY 中断状态位 0: 未发生中断 1: 发生中断
4	PIPE4NRDY	0	R/W*	管道 4 的 NRDY 中断状态位 0: 未发生中断 1: 发生中断
3	PIPE3NRDY	0	R/W*	管道 3 的 NRDY 中断状态位 0: 未发生中断 1: 发生中断
2	PIPE2NRDY	0	R/W*	管道 2 的 NRDY 中断状态位 0: 未发生中断 1: 发生中断
1	PIPE1NRDY	0	R/W*	管道 1 的 NRDY 中断状态位 0: 未发生中断 1: 发生中断
0	PIPE0NRDY	0	R/W*	管道 0 的 NRDY 中断状态位 0: 未发生中断 1: 发生中断

【注】 * 在清除此寄存器的各位所示的状态时，只能给要清除的位写 0 而给其他位写 1。

21.3.20 BEMP 中断状态寄存器 (BEMPSTS)

BEMPSTS 表示各管道的 BEMP 中断状态。

在上电复位时，此寄存器被初始化。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PIPE9 BEMP	PIPE8 BEMP	PIPE7 BEMP	PIPE6 BEMP	PIPE5 BEMP	PIPE4 BEMP	PIPE3 BEMP	PIPE2 BEMP	PIPE1 BEMP	PIPE0 BEMP
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

位	位名	初始值	R/W	说明
15 ~ 10	—	全 0	R	保留位 读取值为 0，只能写 0。
9	PIPE9BEMP	0	R/W*	管道 9 的 BEMP 中断状态位 0: 未发生中断 1: 发生中断
8	PIPE8BEMP	0	R/W*	管道 8 的 BEMP 中断状态位 0: 未发生中断 1: 发生中断
7	PIPE7BEMP	0	R/W*	管道 7 的 BEMP 中断状态位 0: 未发生中断 1: 发生中断
6	PIPE6BEMP	0	R/W*	管道 6 的 BEMP 中断状态位 0: 未发生中断 1: 发生中断
5	PIPE5BEMP	0	R/W*	管道 5 的 BEMP 中断状态位 0: 未发生中断 1: 发生中断
4	PIPE4BEMP	0	R/W*	管道 4 的 BEMP 中断状态位 0: 未发生中断 1: 发生中断
3	PIPE3BEMP	0	R/W*	管道 3 的 BEMP 中断状态位 0: 未发生中断 1: 发生中断
2	PIPE2BEMP	0	R/W*	管道 2 的 BEMP 中断状态位 0: 未发生中断 1: 发生中断
1	PIPE1BEMP	0	R/W*	管道 1 的 BEMP 中断状态位 0: 未发生中断 1: 发生中断
0	PIPE0BEMP	0	R/W*	管道 0 的 BEMP 中断状态位 0: 未发生中断 1: 发生中断

【注】 * 在清除此寄存器的各位所示的状态时，只能给要清除的位写 0 而给其他位写 1。

21.3.21 帧号寄存器 (FRMNUM)

FRMNUM 判断等时错误的通知源以及表示帧号等。

在上电复位时，此寄存器被初始化。

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

位	位名	初始值	R/W	说明
15	OVRN	0	R/W*	<p>超限 / 欠载检测状态位</p> <p>表示是否检测到正在等时传送的管道发生超限 / 欠载错误。</p> <p>0: 无错误 1: 发生错误</p> <p>要将此位清 0 时，能通过软件给此位写 0。此时，必须给此寄存器的其他位写 1。</p> <p>1. 选择主机控制器功能时 在以下任意情况下，此位置 1:</p> <ul style="list-style-type: none"> — 对传送类型是等时发送方向的管道，虽然还没有给 FIFO 缓冲器写完发送数据却到了发行 OUT 标记的时间。 — 对传送类型是等时接收方向的管道，在至少有 1 个 FIFO 缓冲器面不为空的状态下到了发行 IN 标记的时间。 <p>2. 选择功能控制器功能时 在以下任意情况下，此位置 1:</p> <ul style="list-style-type: none"> — 对传送类型是等时发送方向的管道，虽然还没有给 FIFO 缓冲器写完发送数据却接收到 IN Token。 — 对传送类型是等时接收方向的管道，在至少有 1 个 FIFO 缓冲器面不为空的状态下接收到 OUT 标记。
14	CRCE	0	R/W*	<p>接收数据错误位</p> <p>表示检测到正在等时传送的管道发生 CRC 错误或者位填充错误的状态。</p> <p>0: 无错误 1: 发生错误</p> <p>要将此位清 0 时，能通过软件给此位写 0。此时，必须给此寄存器的其他位写 1。</p> <p>1. 选择主机控制器功能时 在检测到 CRC 错误时，产生内部 NRDY 中断请求。</p> <p>2. 选择功能控制器功能时 在检测到 CRC 错误时，不产生内部 NRDY 中断请求。</p>
13 ~ 11	—	全 0	R	<p>保留位</p> <p>读取值为 0，只能写 0。</p>
10 ~ 0	FRNM[10:0]	H'000	R	<p>帧号位</p> <p>在每 1ms 发行 1 次 SOF 时或者接收 SOF 时改写此位，表示最新的帧号。</p> <p>在读此位时，必须重复读取，直到 2 次读取值相同为止。</p>

【注】 * 只能写 0。

21.3.22 μ 帧号寄存器 (UFRMNUM)

UFRMNUM 表示 μ 帧号。

在上电复位时，此寄存器被初始化。

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

位	位名	初始值	R/W	说明
15 ~ 3	—	全 0	R	保留位 读取值为 0，只能写 0。
2 ~ 0	UFRNM[2:0]	000	R	μ 帧位 能确认 μ 帧号。 在高速运行时，此位为 μ 帧号。 在非高速运行时，此位为 B'000。 在读此位时，必须重复读取，直到 2 次读取值相同为止。

21.3.23 USB 地址寄存器 (USBADDR)

USBADDR 表示 USB 地址。

此寄存器只在选择功能控制器功能时有效。在选择主机控制器功能时，必须使用 PIPEMAXP 寄存器的 DEVSEL 位来设定外围设备的地址。

在上电复位或者 USB 总线复位时，此寄存器被初始化。

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

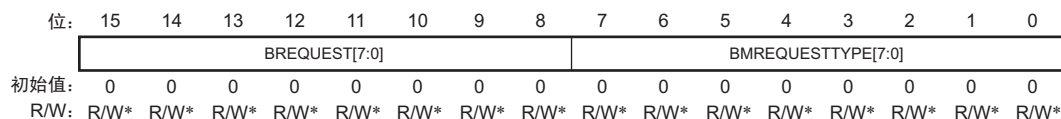
位	位名	初始值	R/W	说明
15 ~ 7	—	全 0	R	保留位 读取值为 0，只能写 0。
6 ~ 0	USBADDR[6:0]	H'00	R	USB 地址 在选择功能控制器功能时，如果正常处理 SET_ADDRESS 请求，就表示由主机分配的 USB 地址。 在检测到 USB 复位时，此位为 H'00。 在选择主机控制器功能时，此位无效。

21.3.24 USB 请求类型寄存器 (USBREQ)

USBREQ 是用于保存控制传送的设置请求的寄存器。

在选择功能控制器功能时，USBREQ 保存接收到的 bRequest 和 bmRequestType 的值；在选择主机控制器功能时，设定要发送的 bRequest 和 bmRequestType 的值。

在上电复位或者 USB 总线复位时，此寄存器被初始化。



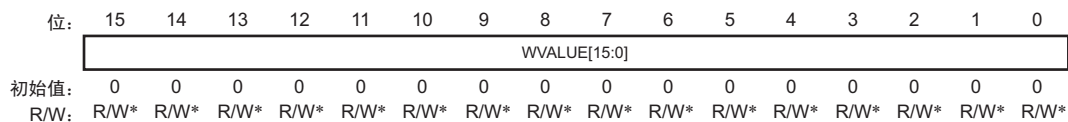
位	位名	初始值	R/W	说明
15 ~ 8	BREQUEST[7:0]	H'00	R/W*	请求位 保存 USB 请求的 bRequest 值。 1. 选择主机控制器功能时 设定要发送的 SETUP 事务的 USB 请求数值。不能在 SUREQ=1 的状态下改写此位。 2. 选择功能控制器功能时 表示 SETUP 事务处理时接收的 USB 请求数值。写无效。
7 ~ 0	BMREQUEST TYPE[7:0]	H'00	R/W*	请求类型位 保存 USB 请求的 bmRequestType 值。 1. 选择主机控制器功能时 设定要发送的 SETUP 事务的 USB 请求数值。不能在 SUREQ=1 的状态下改写此位。 2. 选择功能控制器功能时 表示 SETUP 事务处理时接收的 USB 请求数值。写无效。

【注】 * 在选择功能控制器功能时，是只读位；在选择主机控制器功能时，是可读写位。

21.3.25 USB 请求值寄存器 (USBVAL)

在选择功能控制器功能时，USBVAL 保存接收到的 wValue 值；在选择主机控制器功能时，设定要发送的 wValue 值。

在上电复位或者 USB 总线复位时，此寄存器被初始化。



位	位名	初始值	R/W	说明
15 ~ 0	WVALUE[15:0]	H'0000	R/W*	值 保存 USB 请求的 wValue 值。 1. 选择主机控制器功能时 设定要发送的 SETUP 事务的 USB 请求 wValue 值。不能在 SUREQ=1 的状态下改写此位。 2. 选择功能控制器时 表示 SETUP 事务处理时接收的 USB 请求 wValue 的值。写无效。

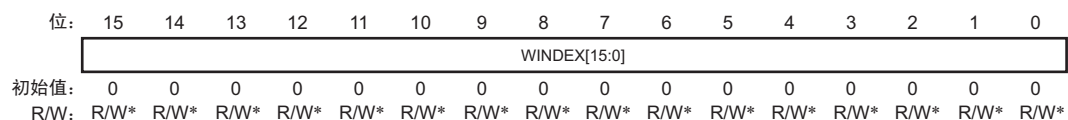
【注】 * 在选择功能控制器功能，是只读位；在选择主机控制器功能时，是可读写位。

21.3.26 USB 请求索引寄存器 (USBINDX)

此寄存器是用于保存控制传送的设置请求的寄存器。

在选择功能控制器功能时，USBINDX 保存接收到的 wIndex 值；在选择主机控制器功能时，设定要发送的 wIndex 值。

在上电复位或者 USB 总线复位时，此寄存器被初始化。



位	位名	初始值	R/W	说明
15 ~ 0	WVALUE[15:0]	H'0000	R/W*	索引 保存 USB 请求的 wIndex 值。 1. 选择主机控制器功能 设定要发送的 SETUP 事务的 USB 请求 wIndex 的值。不能在 SUREQ=1 的状态下改写此位。 2. 选择功能控制器时 表示 SETUP 事务处理时接收的 USB 请求 wIndex 的值。写无效。

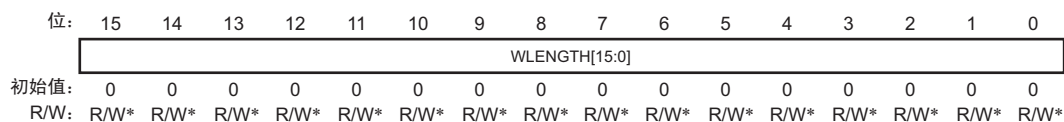
【注】 * 在选择功能控制器功能时，是只读位；在选择主机控制器功能时，是可读写位。

21.3.27 USB 请求长度寄存器 (USBLENG)

此寄存器是用于保存控制传送的设置请求的寄存器。

在选择功能控制器功能时，USBLENG 保存接收到的 wLength 值；在选择主机控制器功能时，设定要发送的 wLength 值。

在上电复位或者 USB 总线复位时，此寄存器被初始化。



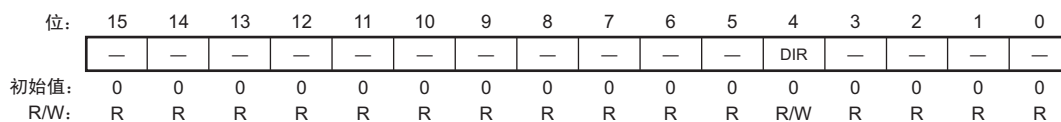
位	位名	初始值	R/W	说明
15 ~ 0	WLENGTH[15:0]	H'0000	R/W*	长度 保存 USB 请求的 wLength 值。 1. 选择主机控制器功能时 设定要发送的 SETUP 事务的 USB 请求 wLength 值。不能在 SUREQ=1 的状态下改写此位。 2. 选择功能控制器时 表示 SETUP 事务处理时接收的 USB 请求 wLength 的值。写无效。

【注】 * 在选择功能控制器功能时，是只读位；在选择主机控制器功能时，是可读写位。

21.3.28 DCP 配置寄存器 (DCPCFG)

DCPCFG 指定默认控制管道 (DCP) 的数据传送方向。

在上电复位时，此寄存器被初始化。



位	位名	初始值	R/W	说明
15 ~ 5	—	全 0	R	保留位 读取值为 0，只能写 0。
4	DIR	0	R/W	传送方向位 在选择主机控制器功能时，设定控制传送的数据阶段和状态阶段的传送方向。 0: 数据接收方向 1: 数据发送方向 在选择功能控制器功能时，必须将此位置 0。
3 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

21.3.29 DCP 最大包长度寄存器 (DCPMAXP)

DCPMAXP 给 DCP 指定最大包长度。

在上电复位时，此寄存器被初始化。

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

位	位名	初始值	R/W	说明
15 ~ 12	DEVSEL[3:0]	0000	R/W	<p>设备选择位</p> <p>在选择主机控制器功能时，指定作为控制传送通信对象的外围设备的地址。</p> <p>0000: 地址 0000 0001: 地址 0001 : 1001: 地址 1001 1010: 地址 1010 上述以外: 禁止设定</p> <p>必须在设定对应此位设定值的 DEVADDn 寄存器后，设定此位。 例如: 在设定 DEVSEL=0010 时，必须给 DEVADD2 寄存器设定地址。 必须在 CSSTS=0、PID=NAK 和 SUREQ=0 时设定此位。 在将 DCP 的 PID 位从 BUF 变为 NAK 后更改此位时，必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是，本模块在将 PID 位变为 NAK 时，不需要通过软件确认 PBUSY 位。 在选择功能控制器功能时，必须将此位置 B'0000。</p>
11 ~ 7	—	全 0	R	<p>保留位</p> <p>读取值为 0，只能写 0。</p>
6 ~ 0	MXPS[6:0]	H'40	R/W	<p>最大包长度</p> <p>将此位设定为 DCP 的最大数据有效负载（最大包长度）。 初始值为 H'40（64 字节）。 必须将 MXPS 位设定为符合 USB 规格的值。 必须在 CSSTS=0、PID=NAK 并且未设定 CURPIPE 位时设定 MXPS 位。 在对应管道的 PID 位从 BUF 变为 NAK 后更改此位时，必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是，本模块在将 PID 位变为 NAK 时，不需要通过软件确认 PBUSY 位。 在 MXPS=0 时，不能写 FIFO 缓冲器，并且不能设定 PID=BUF。</p>

21.3.30 DCP 控制寄存器 (DCPCTR)

DCPCTR 确认 DCP 的缓冲存储器状态、更改并确认 DCP 的数据 PID 顺序位以及设定 DCP 的应答 PID。

在上电复位时，此寄存器被初始化；在 USB 总线复位时，CCPL 和 PID[2:0] 位被初始化。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BSTS	SUREQ	CSCLR	CSCTS	SUREQ CLR	—	—	SQCLR	SQSET	SQMON	PBUSY	PINGE	—	CCPL	PID[1:0]	
初始值:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
R/W:	R	R/W*2	R/W*1	R	R/W*1	R	R	R/W*1	R/W*1	R	R	R/W	R	R/W*1	R/W	R/W

位	位名	初始值	R/W	说明
15	BSTS	0	R	<p>缓冲器状态位</p> <p>表示能否存取 DCP FIFO 缓冲器。</p> <p>0: 不能存取缓冲器</p> <p>1: 能存取缓冲器</p> <p>根据 ISEL 位的设定值, 此位有以下 2 种含义:</p> <ul style="list-style-type: none"> 在 ISEL=0 时, 表示能否读接收数据。 在 ISEL=1 时, 表示能否写发送数据。
14	SUREQ	0	R/W*2	<p>SETUP 标记发送位</p> <p>在选择主机控制器功能时, 通过将此位置 1, 发送设置包。</p> <p>0: 无效</p> <p>1: 发送设置包</p> <p>在 SETUP 事务处理结束后, 本模块产生 SACK 中断或者 SIGN 中断, 并且将此位清 0。</p> <p>如果通过软件将 SUREQCLR 位置 1, 本模块就将此位清 0。</p> <p>在 SETUP 事务处理中, 给 DEVSEL 位、USBREQ 寄存器、USBVAL 寄存器、USBINDX 寄存器和 USBLENG 寄存器设定要发送的 USB 请求后, 必须将此位置 1。</p> <p>在设定 SUREQ=1 前, 必须确认是否将 DCP 的 PID 位设定为 NAK。</p> <p>在将此位置 1 后, 不能在 SETUP 事务处理结束前 (SUREQ=1) 更改 DEVSEL 位、USBREQ 寄存器、USBVAL 寄存器、USBINDX 寄存器和 USBLENG 寄存器的值。</p> <p>只能在发送 SETUP 标记时将此位置 1, 否则必须置 0。</p> <p>在选择功能控制器功能时, 必须将此位置 0。</p>
13	CSCLR	0	R/W*1	<p>分离事务的 C-SPLIT 状态清除位</p> <p>在选择主机控制器功能时, 对于使用分离事务的传送, 能通过将此位置 1 来清除 CSSTS 位。此时, 从 S-SPLIT 重新开始下一次 DCP 传送。</p> <p>0: 无效</p> <p>1: 将 CSSTS 位清 0</p> <p>如果通过软件将此位置 1, 就清除 CSSTS 位。</p> <p>在使用分离事务的传送中, 如果要强制从 S-SPLIT 重新开始下一次传送, 就必须通过软件将此位置 1。正常的分离事务在 C-SPLIT 结束时自动将 CSSTS 位清 0, 因此不需要通过软件进行清除。</p> <p>必须在 UACT=0 并且停止通信时或者在检测到断开时不进行传送的情况下, 通过此位控制 CSSTS 位。</p> <p>在 CSSTS=0 时, 即使将此位置 1, 也保持 CSSTS=0。</p> <p>在选择功能控制器功能时, 必须将此位置 0。</p>
12	CSSTS	0	R	<p>分离事务的 COMPLETE SPLIT (C-SPLIT) 状态位</p> <p>在选择主机控制器功能时, 表示分离事务的 C-SPLIT 状态。</p> <p>0: 正在进行 START-SPLIT (S-SPLIT) 事务处理或者不使用分离事务的设备正在处理中</p> <p>1: 正在进行 C-SPLIT 事务处理</p> <p>在开始 C-SPLIT 时此位为 1, 在检测到 C-SPLIT 结束时此位为 0。</p> <p>在选择功能控制器功能时, 此位的读取值无效。</p>

位	位名	初始值	R/W	说明
11	SUREQCLR	0	R/W*1	<p>SUREQ 位清除</p> <p>在选择主机控制器功能时，能通过将此位置 1 来清除 SUREQ 位。</p> <p>0: 无效</p> <p>1: 将 SUREQ 位清 0</p> <p>此位总是 0。</p> <p>在 SETUP 事务处理中，如果在 SUREQ=1 的状态下停止通信，就必须通过软件将此位置 1。正常的 SETUP 事务在事务处理结束时，自动将 SUREQ 位清 0，因此不需要通过软件进行清除。</p> <p>必须在 UACT=0 并且停止通信时或者在检测到断开时不进行传送的情况下，通过此位控制 SUREQ 位。</p> <p>在选择功能控制器功能时，必须将此位置 0。</p>
10、9	—	全 0	R	<p>保留位</p> <p>读取值为 0，只能写 0。</p>
8	SQCLR	0	R/W*1	<p>交替位清除</p> <p>在 DCP 传送中，能将下一个事务的顺序交替位的期待值设定到 DATA0。</p> <p>0: 无效</p> <p>1: 指定 DATA0</p> <p>此位总是 0。</p> <p>不能同时将 SQCLR 位和 SQSET 位置 1。</p> <p>必须在 CSCTS=0、PID=NAK 并且未设定 CURPIPE 时将此位置 1。</p> <p>在对应管道的 PID 位从 BUF 变为 NAK 后将此位置 1 时，必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是，本模块在将 PID 位变为 NAK 时，不需要通过软件确认 PBUSY 位。</p>
7	SQSET	0	R/W*1	<p>交替位置位</p> <p>在 DCP 传送中，能将下一个事务的顺序交替位的期待值设定到 DATA1。</p> <p>0: 无效</p> <p>1: 指定 DATA1</p> <p>不能同时将 SQCLR 位和 SQSET 位置 1。</p> <p>必须在 CSCTS=0、PID=NAK 并且未设定 CURPIPE 时将此位置 1。</p> <p>在对应管道的 PID 位从 BUF 变为 NAK 后将此位置 1 时，必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是，本模块在将 PID 位变为 NAK 时，不需要通过软件确认 PBUSY 位。</p>
6	SQMON	1	R	<p>顺序交替位监视</p> <p>在 DCP 传送中，表示下一个事务的顺序交替位的期待值。</p> <p>0: DATA0</p> <p>1: DATA1</p> <p>如果事务被正常处理，本模块就将此位取反。但是，如果在接收方向传送时发生 DATA-PID 不一致，就不将此位取反。</p> <p>在选择功能控制器功能时，如果正常接收 SETUP 包，就将此位置 1（将期待值设定到 DATA1）。状态阶段的 IN/OUT 事务不参照此位，即使正常结束也不将此位取反。</p>

位	位名	初始值	R/W	说明
5	PBUSY	0	R	<p>管道忙</p> <p>在 PID 位从 BUF 变为 NAK 时，表示 DCP 事务是否使用 DCP。</p> <p>0: 事务不使用 DCP</p> <p>1: 事务使用 DCP</p> <p>详细内容请参照“21.4.3(1) 管道控制寄存器的切换步骤”。</p> <p>在开始该管道的 USB 事务时，此位从 0 变为 1；在 1 个事务结束时，此位从 1 变为 0。</p> <p>在通过软件设定 PID=NAK 后，能通过读此位来确认管道是否变为能更改设定的状态。</p>
4	PINGE	0	R/W	<p>PING 标记发行允许位</p> <p>在选择主机控制器功能时，如果将此位置 1，在发送方向的传送中，发行 PING 标记，从 PING 事务开始进行发送方向的传送。</p> <p>0: 禁止发行 PING 标记</p> <p>1: 正常 PING 运行</p> <p>如果在 PING 事务处理中检测到 ACK 握手信号，就在下一个事务处理中执行 OUT 事务。</p> <p>如果在 OUT 事务处理中检测到 NAK 握手信号，就在下一个事务处理中执行 PING 事务。</p> <p>在选择主机控制器功能时，如果通过软件将此位置 0，就在发送方向的传送中不发行 PING 标记。所有发送方向的传送都执行 OUT 事务。必须在 CSSTS=0 和 PID=NAK 时更改此位。</p> <p>在对应管道的 PID 位从 BUF 变为 NAK 后将此位置 1 时，必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是，本模块在将 PID 位变为 NAK 时，不需要通过软件确认 PBUSY 位。</p> <p>在选择功能控制器功能时，必须将此位置 0。</p>
3	—	0	R	<p>保留位</p> <p>读取值为 0，只能写 0。</p>
2	CCPL	0	R/W*1	<p>控制传送结束允许位</p> <p>在选择功能控制器功能时，通过将此位置 1 来允许控制传送状态阶段的结束。</p> <p>0: 无效</p> <p>1: 允许控制传送结束</p> <p>在对应的 PID 位为 BUF 时，如果通过软件将此位置 1，就结束控制传送阶段。</p> <p>在控制读传送时，对于 USB 主机的 OUT 事务，发送 ACK 握手信号；在控制写和无数据控制传送时，对于 USB 主机的 IN 事务，发送 Zero-Length 包。但是，当检测到 SET_ADDRESS 请求时，就与此位的设定值无关，在 SETUP 阶段和状态阶段结束之间进行自动应答。</p> <p>在接收到新的 SETUP 包时，将此位从 1 变为 0。</p> <p>在 VALID=1 时，无法通过软件将此位置 1。</p> <p>在选择主机控制器功能时，必须将此位置 0。</p>

位	位名	初始值	R/W	说明
1、0	PID[1:0]	00	R/W	<p>应答 PID</p> <p>在控制传送时，通过此位控制本模块的应答。</p> <p>00: NAK 应答 01: BUF 应答 (根据缓冲器状态) 10: STALL 应答 11: STALL 应答</p> <p>1. 选择主机控制器功能时 按以下顺序将此位从 NAK 变为 BUF:</p> <ul style="list-style-type: none"> — 设定为发送方向时 必须在 UACT=1 和 PID=NAK 的状态下将发送数据写到 FIFO 缓冲器，并且设定 PID=BUF。本模块在设定 PID=BUF 后执行 OUT 事务 (或者 PING 事务)。 — 设定为接收方向时 必须在 UACT=1 和 PID=NAK 的状态下确认 FIFO 缓冲器为空 (设定为空)，并且设定 PID=BUF。本模块在设定 PID=BUF 后执行 IN 事务。 <p>在以下情况下更改此位的值:</p> <ul style="list-style-type: none"> — 在通过软件将此位设定为 BUF 时，如果接收到超过 MaxPacketSize 的数据，就设定 PID=STALL (11)。 — 如果连续 3 次检测到 CRC 错误等接收错误，就设定 PID=NAK。 — 如果接收到 STALL 握手信号，就设定 PID=STALL (11)。 <p>在对选择的管道发行分离事务的 S-SPLIT 后 (CSSTS=1 时)，即使通过软件将此位变为 NAK，也在结束 C-SPLIT 前执行该事务。在 C-SPLIT 结束时，设定 PID=NAK。</p> <p>2. 选择功能控制器功能时 在以下情况下更改此位的值:</p> <ul style="list-style-type: none"> — 在接收到 SETUP 包时，将此位变为 PID=NAK。此时，设定 VALID=1，必须通过软件在设定 VALID=0 后更改此位。 — 在通过软件将此位设定为 BUF 时，如果接收到超过 MaxPacketSize 的数据，就设定 PID=STALL (11)。 — 如果检测到控制传送顺序错误时，就设定 PID=STALL (1x)。 — 如果检测到 USB 总线复位，就设定 PID=NAK。 <p>在 SET_ADDRESS 请求处理 (自动处理) 时，不参照此位的设定值。</p>

【注】 *1 读取值为 0，只能写 1。

*2 只能写 1。

21.3.31 管道窗口选择寄存器 (PIPESEL)

通过 PIPESEL、PIPECFG、PIPEBUF、PIPEMAXP、PIPEPERI、PIPECTR、PIPETRE 和 PIPESTRN 寄存器设定管道 1 ~ 9。

在通过 PIPESEL 寄存器设定要使用的管道后，通过 PIPECFG、PIPEBUF、PIPEMAXP 和 PIPEPERI 寄存器设定各管道的功能。与 PIPESEL 寄存器选择的管道无关，能设定 PIPECTR、PIPETRE 和 PIPESTRN 寄存器。

在上电复位或者 USB 总线复位时，全部管道的寄存器的对应位都被初始化。

在上电复位时，此寄存器被初始化。

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

位	位名	初始值	R/W	说明
15 ~ 4	—	全 0	R	保留位 读取值为 0，只能写 0。
3 ~ 0	PIPESEL[3:0]	0000	R/W	管道窗口选择位 指定与读写对象的 PIPECFG 寄存器、PIPEBUF 寄存器、PIPEMAXP 寄存器和 PIPEPERI 寄存器对应的管道号。 0000: 未选择 0001: 管道 1 0010: 管道 2 0011: 管道 3 0100: 管道 4 0101: 管道 5 0110: 管道 6 0111: 管道 7 1000: 管道 8 1001: 管道 9 上述以外: 禁止设定 能读写此位指定的管道号对应的 PIPECFG、PIPEBUF、PIPEMAXP 和 PIPEPERI 寄存器。 在将此位置 0000 后，PIPECFG 寄存器、PIPEBUF 寄存器、PIPEMAXP 寄存器、PIPEPERI 寄存器和 PIPECTR 寄存器的各位的读取值为 0。 写无效。

21.3.32 管道配置寄存器 (PIPECFG)

此寄存器指定管道 1 ~ 9 的传送类型、缓冲存储器的存取方向和端点号，并且选择是连续传送模式还是非连续传送模式、是单缓冲器还是双缓冲器以及在传送结束时是否禁止管道运行。

在上电复位时，此寄存器被初始化；在 USB 总线复位时，TYPE[1:0] 被初始化。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TYPE[1:0]		—	—	—	BFRE	DBLB	CNTMD	SHT NAK	—	—	DIR	EPNUM[3:0]			
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15、14	TYPE[1:0]	00	R/W	<p>传送类型位 指定给 PIPESEL 位选择的管道的传送类型。</p> <ul style="list-style-type: none"> • 管道 1、2 <ul style="list-style-type: none"> 00: 不使用管道 01: 批量传送 10: 禁止设定 11: 等时传送 • 管道 3 ~ 5 <ul style="list-style-type: none"> 00: 不使用管道 01: 批量传送 10: 禁止设定 11: 禁止设定 • 管道 6 ~ 9 <ul style="list-style-type: none"> 00: 不使用管道 01: 禁止设定 10: 中断传送 11: 禁止设定 <p>在将选择的管道设定为 PID=BUF (开始使用所选管道的 USB 通信) 前, 必须将此位置 00 以外的值。 必须在所选管道的 PID 位为 NAK 状态时更改此位。在所选管道的 PID 位从 BUF 变为 NAK 后更改此位时, 必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是, 本模块在将 PID 位变为 NAK 时, 不需要通过软件确认 PBUSY 位。</p>
13 ~ 11	—	全 0	R	<p>保留位 读取值为 0, 只能写 0。</p>
10	BFRE	0	R/W	<p>BRDY 中断运行指定位 指定本模块向 CPU 发行 BRDY 中断 (与所选管道相关) 的时序。</p> <ul style="list-style-type: none"> 0: 在发送 / 接收数据时产生 BRDY 中断 1: 在读完数据时产生 BRDY 中断 <p>如果通过软件将此位置 1 并且选择的管道用于接收方向, 本模块就检测传送结束, 在读完该信息包时产生 BRDY 中断。 在通过此设定产生 BRDY 中断时, 需要通过软件设定 BCLR=1。在设定 BCLR=1 前分配给所选管道的 FIFO 缓冲器不变为可接收状态。 在通过软件将此位置 1 并且选择的管道用于发送方向时, 本模块不产生 BRDY 中断。 详细内容请参照 “21.4.2 (1) BRDY 中断”。 必须在 CSSTS=0、PID=NAK 并且还未通过 CURPIPE 位设定管道时更改此位。 在进行使用所选管道的 USB 通信后更改此位的设定时, 除了上述 3 个寄存器状态以外, 还必须通过软件连续设定 ACLRM=1 和 ACLRM=0, 清除分配给所选管道的 FIFO 缓冲器。 在所选管道的 PID 位从 BUF 变为 NAK 后更改此位时, 必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是, 本模块在将 PID 位变为 NAK 时, 不需要通过软件确认 PBUSY 位。</p>

位	位名	初始值	R/W	说明
9	DBLB	0	R/W	<p>双缓冲器模式位 指定所选管道使用的 FIFO 缓冲器是单缓冲器还是双缓冲器。</p> <p>0: 单缓冲器 1: 双缓冲器</p> <p>此位在选择管道 1 ~ 5 时有效。</p> <p>在通过软件将此位置 1 后, 本模块将通过 PIPEBUF 寄存器的 BUFSIZE 位指定的 2 个 FIFO 缓冲器面分配给选择的管道。</p> <p>分配给所选管道的 FIFO 缓冲器容量如下: (BUFSIZE+1)×64×(DBLB+1)[字节]</p> <p>在通过软件将此位置 1 并且选择的管道用于发送方向时, 本模块不产生 BRDY 中断。</p> <p>详细内容请参照 PIPEBRDY 中断寄存器。</p> <p>必须在 CSSTS=0、PID=NAK 并且还未通过 CURPIPE 位设定管道号时更改此位。</p> <p>在进行使用所选管道的 USB 通信后更改此位的设定时, 除了上述 3 个寄存器状态以外, 还必须通过软件连续设定 ACLRM=1 和 ACLRM=0, 清除分配给所选管道的 FIFO 缓冲器。</p> <p>在所选管道的 PID 位从 BUF 变为 NAK 后更改此位时, 必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是, 本模块在将 PID 位变为 NAK 时, 不需要通过软件确认 PBUSY 位。</p>
8	CNTMD	0	R/W	<p>连续传送模式位 指定所选管道是否在连续传送模式下进行通信。</p> <p>0: 非连续传送模式 1: 连续传送模式</p> <p>此位在通过 PIPESEL 位选择管道 1 ~ 5 并且选择批量传送 (TYPE=01) 时有效。</p> <p>必须在 CSSTS=0、PID=NAK 并且还未通过 CURPIPE 位设定管道号时更改此位。</p> <p>在进行使用所选管道的 USB 通信后更改此位时, 除了上述 3 个寄存器状态以外, 还必须通过软件连续设定 ACLRM=1 和 ACLRM=0, 清除分配给所选管道的 FIFO 缓冲器。</p> <p>在所选管道的 PID 位从 BUF 变为 NAK 后更改此位时, 必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是, 本模块在将 PID 位变为 NAK 时, 不需要通过软件确认 PBUSY 位。</p>

位	位名	初始值	R/W	说明
7	SHTNAK	0	R/W	<p>传送结束时的管道禁止位</p> <p>所选管道为接收方向时, 指定是否在传送结束时将 PID 变为 NAK。</p> <p>0: 在发送结束时继续管道的使用</p> <p>1: 在传送结束时禁止管道的使用</p> <p>此位在所选管道为管道 1 ~ 管道 5 并且是接收方向时有效。</p> <p>对接收方向的管道, 如果通过软件将此位置 1, 就在认为到所选管道的传送结束时将所选管道对应的 PID 位变为 NAK。在满足以下条件时认为传送结束:</p> <ul style="list-style-type: none"> • 正常接收到短包数据 (包含 Zero-Length) 时 • 使用事务计数器并且正常接收到事务计数器指定数量的信息包时必须 CSSTS=0 和 PID=NAK 时更改此位。 <p>在所选管道的 PID 位从 BUF 变为 NAK 后更改此位时, 必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是, 本模块在将 PID 位变为 NAK 时, 不需要通过软件确认 PBUSY 位。</p> <p>对于发送方向的管道, 必须将此位置 0。</p>
6、5	—	全 0	R	<p>保留位</p> <p>读取值为 0, 只能写 0。</p>
4	DIR	0	R/W	<p>传送方向位</p> <p>指定所选管道的传送方向。</p> <p>0: 接收方向</p> <p>1: 发送方向</p> <p>在通过软件将此位置 0 时, 如果将所选管道设定为接收方向并且将此位置 1, 本模块就将所选管道用于发送方向。</p> <p>必须在 CSSTS=0、PID=NAK 并且还未通过 CURPIPE 位设定管道时更改此位。</p> <p>在进行使用所选管道的 USB 通信后更改此位的设定时, 除了上述 3 个寄存器状态以外, 还必须通过软件连续设定 ACLRM=1 和 ACLRM=0, 清除分配给所选管道的 FIFO 缓冲器。</p> <p>在所选管道的 PID 位从 BUF 变为 NAK 后更改此位时, 必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是, 本模块在将 PID 位变为 NAK 时, 不需要通过软件软件确认 PBUSY 位。</p>
3 ~ 0	EPNUM[3:0]	0000	R/W	<p>端点号</p> <p>指定所选管道的端点号。</p> <p>0000 表示未使用的管道。</p> <p>必须在 CSSTS=0 和 PID=NAK 时更改此位。</p> <p>在所选管道的 PID 位从 BUF 变为 NAK 后更改此位时, 必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是, 本模块在将 PID 位变为 NAK 时, 不需要通过软件确认 PBUSY 位。</p> <p>设定时, DIR 位和 EPNUM 位的设定组合不能与其它管道的设定重复 (EPNUM=0000 的设定可重复。)</p>

21.3.33 管道缓冲器指定寄存器 (PIPEBUF)

PIPEBUF 指定管道 1 ~ 9 的缓冲器大小和缓冲器号。

在上电复位时，此寄存器被初始化。

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

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读取值为 0，只能写 0。
14 ~ 10	BUFSIZE[4:0]	H'00	R/W	缓冲器大小位 指定通过 PIPESEL 位选择的管道的缓冲器大小。以块数为单位，1 个块为 64 字节。 00000 (H'00): 64 字节 00001 (H'01): 128 字节 : : 11111 (H'1F): 2K 字节 在通过软件设定 DBLB=1 时，本模块将通过此位指定的 2 个 FIFO 缓冲器面分配给选择的管道。 分配给所选管道的 FIFO 缓冲器容量如下： (BUFSIZE+1)×64×(DBLB+1)[字节] 根据选择的管道，能给此位设定的值如下： 管道 1 ~ 5: 必须设定 BUFSIZE=H'00 ~ H'1F。 管道 6 ~ 9: 必须设定 BUFSIZE=H'00。 在 CNTMD=1 的状态下使用时，必须将此位设定为 MaxPacketSize 的整数倍的值。 必须在 CSSTS=0、PID=NAK 并且还未通过 CURPIPE 位设定管道时更改此位。 在所选管道的 PID 位从 BUF 变为 NAK 后更改此位时，必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是，本模块在将 PID 位变为 NAK 时，不需要通过软件确认 PBUSY 位。
9、8	—	全 0	R	保留位 读取值为 0，只能写 0。

位	位名	初始值	R/W	说明
7 ~ 0	BUFNMB[7:0]	H'00	R/W	<p>缓冲器号</p> <p>用 H'04 ~ H'4F 指定所选管道的 FIFO 缓冲器号。</p> <p>在将管道 1 ~ 5 设定为所选管道时，能给此位设定适合用户系统的值。</p> <p>BUFNMB=0 ~ 3 为 DCP 专用。</p> <p>BUFNMB=4 为管道 6 专用。</p> <p>但是，在不使用管道 6 时，其它管道能使用。</p> <p>在所选管道为管道 6 时，此位的写无效，自动分配 BUFNMB=4。</p> <p>BUFNMB=5 为管道 7 专用。</p> <p>但是，在不使用管道 7 时，其它管道能使用。</p> <p>在所选管道为管道 7 时，此位的写无效，自动分配 BUFNMB=5。</p> <p>BUFNMB=6 为管道 8 专用。</p> <p>但是，在不使用管道 8 时，其它管道能使用。</p> <p>在所选管道为管道 8 时，此位的写无效，自动分配 BUFNMB=6。</p> <p>BUFNMB=7 为管道 9 专用。</p> <p>但是，在不使用管道 9 时，其它管道能使用。</p> <p>在所选管道为管道 9 时，此位的写无效，自动分配 BUFNMB=7。</p> <p>必须在 CSSTS=0、PID=NAK 并且还未通过 CURPIPE 位设定管道时更改此位。</p> <p>在所选管道的 PID 位从 BUF 变为 NAK 后更改此位时，必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是，本模块在将 PID 位变为 NAK 时，不需要通过软件确认 PBUSY 位。</p>

21.3.34 管道最大包长度寄存器 (PIPEMAXP)

PIPEMAXP 指定管道 1 ~ 9 的最大包长度。

在上电复位时，此寄存器被初始化。

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

位	位名	初始值	R/W	说明
15 ~ 12	DEVSEL[3:0]	00	R/W	<p>设备选择位</p> <p>在选择主机控制器功能时，指示通信对象的外围设备的 USB 设备地址。</p> <p>0000: 地址 0000 0001: 地址 0001 0010: 地址 0010 : 1010: 地址 1010 上述以外: 禁止设定</p> <p>必须在设定对应此位设定值的 DEVADDn (n=0 ~ A) 寄存后，设定此位。</p> <p>例如，在设定 DEVSEL=0010 时，到 DEVADD2 设定地址。</p> <p>在将 PID 从 BUF 变为 NAK 后更改此位的设定时，必须在确认所选管道的 CSSTS=0 和 PBUSY=0 后进行。本模块在将 PID 变为 NAK 时，不需要确认 PBUSY 位。</p> <p>在选择功能控制器功能时，必须将此位置 B'0000。</p>
11	—	0	R	<p>保留位</p> <p>读取值为 0，只能写 0。</p>
10 ~ 0	MXPS[10:0]	*	R/W	<p>最大包长度</p> <p>指定所选管道的最大数据有效负载 (最大包长度)。各管道能设定的值的范围如下所示:</p> <p>管道 1、2: 1 字节 (H'001) ~ 1024 字节 (H'400) 管道 3 ~ 5: 8 字节 (H'008)、16 字节 (H'010)、32 字节 (H'020)、64 字节 (H'040)、512 字节 (H'200) (无 [2:0] 的位。) 管道 6 ~ 9: 1 字节 (H'001) ~ 64 字节 (H'040)</p> <p>必须按各传送类型给 MXPS 位设定符合 USB 规格的值。</p> <p>在分离事务处理中进行等时管道通信时，MXPS 位的设定值不能超过 188 字节。</p> <p>在所选管道的 PID 位从 BUF 变为 NAK 后更改此位时，必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是，本模块在将 PID 位变为 NAK 时，不需要通过软件确认 PBUSY 位。</p> <p>在 MXPS=0 时，不能写 FIFO 缓冲器，并且不能设定 PID=BUF。</p>

【注】 * 未通过 PIPESEL 寄存器的 PIPESEL 位选择管道时为 H'000，否则为 H'040。

21.3.35 管道周期控制寄存器 (PIPEPERI)

管道 1 ~ 9 在等时 IN 传送期间发生间隔错误时, 选择是否运行缓冲器清除功能以及设定间隔错误的检测间隔。

在上电复位时, 此寄存器被初始化。

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

位	位名	初始值	R/W	说明
15 ~ 13	—	全 0	R	保留位 读取值为 0, 只能写 0。
12	IFIS	0	R/W	等时 IN 缓冲器清除位 在 PIPESEL 位选择的管道为等时 IN 传送时, 指定是否清除缓冲器。 0: 不清除缓冲器 1: 清除缓冲器 在选择功能控制器功能时, 如果所选管道的传送类型为等时并且传送方向为 IN 传送, 就在 IITV 位所设各间隔的 (μ) 帧中没有接收到 USB 主机的 IN-Token 时, 自动清除 FIFO 缓冲器。 在设定双缓冲器 (DBLB=1 时) 时, 清除对象是先前使用的 1 个缓冲器面。 在应该接收 IN-Token 的 (μ) 帧之后接收到 SOF 包时, 清除 FIFO 缓冲器。即使 SOF 包破损, 也通过内插功能在应该接收 SOF 时进行清除。 在选择主机控制器功能时或者在所选管道的传送类型不为等时传送的情况下, 必须将此位置 0。
11 ~ 3	—	全 0	R	保留位 读取值为 0, 只能写 0。
2 ~ 0	IITV[2:0]	000	R/W	间隔错误检测间隔位 必须将所选管道的间隔错误检测间隔指定为帧时序的 2 的 n 次方。 选择主机控制器功能时的功能和选择功能控制器功能时的功能不同。 必须在 CSSTS=0、PID=NAK 并且还未通过 CURPIPE 位设定管道时设定此位。 在所选管道的 PID 位从 BUF 变为 NAK 后更改此位时, 必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是, 本模块在将 PID 位变为 NAK 时, 不需要通过软件确认 PBUSY 位。 在设定此位并且进行 USB 通信后要变为其它值时, 必须在设定 PID=NAK 后设定 ACLRM=1, 进行间隔定时器的初始化。 对于管道 3 ~ 5, 此位无效, 必须置 000。

21.3.36 管道 n 控制寄存器 (PIPEnCTR) (n=1 ~ 9)

对于管道 1 ~ 9, 此寄存器确认缓冲存储器状态, 变更和确认数据 PID 顺序位, 并且选择是否设定为自动应答模式和自动清除缓冲器模式以及设定应答 PID。设定时, 与通过 PIPESEL 寄存器选择的管道无关。

在上电复位时, 此寄存器被初始化; 在 USB 总线复位时, PID[1:0] 位被初始化。

(1) PIPEnCTR (n=1 ~ 5)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BSTS	INBUFM	CSCLR	CSSTS	—	AT REPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W*2	R	R	R/W	R/W	R/W*1	R/W*1	R	R	R	R	R	R/W	R/W

位	位名	初始值	R/W	说明
15	BSTS	0	R	缓冲器状态位 表示相应管道的 FIFO 缓冲器状态。 0: CPU 不能存取缓冲器 1: CPU 能存取缓冲器 此位的含义因 DIR、BFRE 和 DCLRM 位的设定值而不同, 如表 21.11 所示。
14	INBUFM	0	R	发送缓冲器监视位 在相应管道为发送方向时, 表示该管道的 FIFO 缓冲器状态。 0: 缓冲存储器中没有能发送的数据 1: 缓冲存储器中有能发送的数据 如果将该管道设定为发送方向 (DIR=1), 就在通过软件 (或者 DMAC) 至少给 FIFO 缓冲器写完 1 个面的数据后将此位置 1。 如果写完的 FIFO 缓冲器面中的数据全部发送结束, 就将此位置 0。如果使用双缓冲器 (DBLB=1 时), 就在发送完 2 个面的数据并且软件 (或者 DMAC) 还未写完 1 个面的数据时将此位置 0。 在将该管道设定为接收方向 (DIR=0) 时, 此位为 BSTS 位的相同值。
13	CSCLR	0	R/W*2	CSPLIT 状态清除位 在选择主机控制器功能时, 如果通过软件将此位置 1, 本模块就将 CSSTS 位清 0。 0: 写无效 1: 清除 CSSTS 位 在使用分离事务的传送中, 如果要强制从 S-SPLIT 重新开始下一次传送, 就必须通过软件将此位置 1。正常的分离事务在 C-SPLIT 结束时自动将 CSSTS 位清 0, 因此不需要通过软件进行清除。 必须在 UACT=0 并且停止通信时或者在检测到断开时不进行传送的情况下, 通过此位控制 CSSTS 位。 在 CSSTS=0 时, 即使将此位置 1, 也保持 CSSTS=0。 在选择功能控制器功能时, 必须将此位置 0。

位	位名	初始值	R/W	说明
12	CSSTS	0	R	<p>CSSTS 状态位</p> <p>在选择主机控制器功能时，此位表示分离事务的 C-SPLIT 状态。</p> <p>0: 正在进行 START-SPLIT (S-SPLIT) 事务处理或者正在进行不使用分离事务的传送</p> <p>1: 正在进行 C-SPLIT 事务处理</p> <p>在开始 C-SPLIT 时此位为 1，在检测到 C-SPLIT 结束时此位为 0。此位只在选择主机控制器功能时有效。</p>
11	—	0	R	<p>保留位</p> <p>读取值为 0，只能写 0。</p>
10	ATREPM	0	R/W	<p>自动应答模式位</p> <p>指定是禁止还是允许相应管道的自动应答。</p> <p>0: 禁止自动应答</p> <p>1: 允许自动应答</p> <p>在选择功能控制器功能时，如果将相应管道的传送类型设定为批量传送，就能将此位置 1。</p> <p>在将此位置 1 后，对于 USB 主机的标记，进行以下应答：</p> <ol style="list-style-type: none"> 该管道为 Bulk-IN 传送 (TYPE=01 并且 DIR=1) 时 如果 ATREPM=1 并且 PID=BUF，就对 IN-Token 发送 Zero-Length 包。 每当接收到 USB 主机的 ACK 时 (1 个事务：接收 IN-Token → 发送 Zero Length 包 → 接收 ACK)，就更新 (交替) 顺序交替位 (DATA-PID)。 不产生 BRDY 中断和 BEMP 中断。 该管道为 Bulk-OUT 传送 (TYPE=01 并且 DIR=0) 时 如果 ATREPM=1 并且 PID=BUF，就对 OUT-Token (或者 PING-Token) 进行 NAK 应答，并且产生 NRDY 中断。 <p>必须在 CSSTS=0 和 PID=NAK 时更改此位。</p> <p>在对应管道的 PID 位从 BUF 变为 NAK 后更改此位时，必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是，本模块在将 PID 位变为 NAK 时，不需要通过软件确认 PBUSY 位。</p> <p>为了以自动应答模式进行 USB 通信，必须在 FIFO 缓冲器为空的状态下将此位置 1。在将此位置 1 进行 USB 通信期间，不能写 FIFO 缓冲器。</p> <p>在该管道的传送类型为等时传送时或者在选择主机控制器功能时，必须将此位置 0。</p>
9	ACLRM	0	R/W	<p>缓冲器自动清除模式位</p> <p>指定是禁止还是允许相应管道的缓冲器自动清除模式。</p> <p>0: 禁止</p> <p>1: 允许 (全部缓冲器被初始化)</p> <p>在要将分配给相应管道的 FIFO 缓冲器的内容全部删除时，必须给 ACLRM 位连续写 1 和 0。</p> <p>给此位连续写 1 和 0 时的清除内容以及需要清除的情况如表 21.12 所示。</p> <p>必须在 CSSTS=0 和 PID=NAK 时更改此位。</p> <p>在对应管道的 PID 位从 BUF 变为 NAK 后更改此位时，必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是，本模块在将 PID 位变为 NAK 时，不需要通过软件确认 PBUSY 位。</p>

位	位名	初始值	R/W	说明
8	SQCLR	0	R/W*1	<p>交替位清除</p> <p>在要清除相应管道的下一个事务的顺序交替位的期待值 (将期待值设定到 DATA0) 时, 将此位置 1。</p> <p>0: 无效 1: 指定 DATA0</p> <p>如果通过软件将此位置 1, 就将相应管道的顺序交替位的期待值设定到 DATA0。此位总是 0。</p> <p>在选择主机控制器功能时, 对于 Bulk-Out 传送管道, 如果将此位置 1, 就从 PING-token 开始进行该管道的下一次传送。</p> <p>必须在 CSCTS=0 和 PID=NAK 时将 SQCLR 位置 1。</p> <p>在对应管道的 PID 位从 BUF 变为 NAK 后将此位置 1 时, 必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是, 本模块在将 PID 位变为 NAK 时, 不需要通过软件确认 PBUSY 位。</p>
7	SQSET	0	R/W*1	<p>交替位置位</p> <p>在要将相应管道的下一个事务的顺序交替位的期待值设定到 DATA1 时, 将此位置 1。</p> <p>0: 无效 1: 指定 DATA1</p> <p>如果通过软件将此位置 1, 就将相应管道的顺序交替位的期待值设定到 DATA1。此位总是 0。</p> <p>必须在 CSSTS=0 和 PID=NAK 时将 SQSET 位置 1。</p> <p>在对应管道的 PID 位从 BUF 变为 NAK 后将此位置 1 时, 必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是, 本模块在将 PID 位变为 NAK 时, 不需要通过软件确认 PBUSY 位。</p>
6	SQMON	0	R	<p>交替位确认</p> <p>表示相应管道的下一个事务的顺序交替位的期待值。</p> <p>0: DATA0 1: DATA1</p> <p>在相应管道的传送类型不为等时传送时, 如果事务被正常处理, 本模块就将此位取反。但是, 如果在接收方向传送时发生 DATA-PID 不一致, 就不将此位取反。</p>
5	PBUSY	0	R	<p>管道忙</p> <p>表示当前事务是否使用相应管道。</p> <p>0: 事务不使用相应管道 1: 事务使用相应管道</p> <p>详细内容请参照“21.4.3(1) 管道控制寄存器的切换步骤”。</p> <p>在开始该管道的 USB 事务时, 此位从 0 变为 1; 在 1 个事务结束时, 此位从 1 变为 0。</p> <p>在通过软件设定 PID=NAK 后, 能通过读此位来确认管道是否变为能更改设定的状态。</p>
4 ~ 2	—	全 0	R	<p>保留位</p> <p>读取值为 0, 只能写 0。</p>

位	位名	初始值	R/W	说明
1、0	PID[1:0]	00	R/W	<p>应答 PID 指定相应管道的下一个事务的应答类型。</p> <p>00: NAK 应答 01: BUF 应答 (根据缓冲器状态) 10: STALL 应答 11: STALL 应答</p> <p>此位的默认值为 NAK。在通过相应管道进行 USB 传送时, 必须将此位更改为 BUF。各 PID 位的设定值所对应的模块基本运行 (通信包中无错误时的运行) 如表 21.13 和表 21.14 所示。</p> <p>当该管道处于 USB 通信过程中时, 如果通过软件将此位从 BUF 变为 NAK, 就必须在写 NAK 后, 通过 PBUSY=1 来确认该管道的 USB 传送已转移到 NAK 状态。但是, 本模块在将此位变为 NAK 时, 不需要通过软件确认 PBUSY 位。</p> <p>在以下情况下更改此位的值:</p> <ul style="list-style-type: none"> • 在该管道为接收方向时并且通过软件将所选管道的 SHTNAK 位置 1 时, 如果本模块认为传送结束, 就设定 PID=NAK。 • 在接收到超过该管道的 MaxPacketSize 有效负载的数据包时, 就设定 PID=STALL (11)。 • 在选择功能控制器功能时, 如果检测到 USB 总线复位, 就设定 PID=NAK。 • 在选择主机控制器功能时, 如果连续 3 次检测到 CRC 错误等接收错误, 就设定 PID=NAK。 • 在选择主机控制器功能时, 如果接收到 STALL 握手信号, 就设定 PID=STALL (11)。 <p>按以下步骤进行此位的设定:</p> <ul style="list-style-type: none"> • 在从 NAK (00) 状态转移到 STALL 状态时, 必须置 10。 • 在从 BUF (01) 状态转移到 STALL 状态时, 必须置 11。 • 在从 STALL (11) 状态转移到 NAK 状态时, 必须写 10 后接着写 00。 • 在从 STALL 状态转移到 BUF 状态时, 必须在变为 NAK 状态后设定为 BUF 状态。

【注】 *1 读取值为 0, 只能写 1。

*2 只能写 1。

表 21.11 BSTS 位的运行

DIR 位	BFRE 位	DCLRM 位	BSTS 位的含义
0	0	0	在能从 FIFO 缓冲器读接收数据时，此位为 1；在读完数据时，此位为 0。
		1	禁止设定。
	1	0	在能从 FIFO 缓冲器读接收数据时，此位为 1；在读完数据后通过软件设定 BCLR=1 时，此位为 0。
		1	在能从 FIFO 缓冲器读接收数据时，此位为 1；在读完数据时，此位为 0。
1	0	0	在能给 FIFO 缓冲器写发送数据时，此位为 1；在写完数据时，此位为 0。
		1	禁止设定。
	1	0	禁止设定。
		1	禁止设定。

表 21.12 ACLRM=1 时的清除内容

编号	通过 ACLRM 位操作的清除内容	需要清除的情况
1	分配给相应管道的 FIFO 缓冲器的全部内容（在设定双缓冲器时，清除 2 个 FIFO 缓冲器面）	
2	在相应管道的传送类型为等时传送时，为间隔计数值。	在对间隔计数值进行复位时
3	与 BFRE 位有关的内部标志	在更改 BFRE 位的设定值时
4	FIFO 缓冲器交替控制	在更改 DBLB 位的设定值时
5	与事务计数有关的内部标志	在强制结束事务计数功能时

表 21.13 PID 位对应的模块运行一览表（选择主机控制器功能时）

PID 位	传送类型	传送方向 (DIR 位)	模块的运行
00 (NAK)	不取决于设定值	不取决于设定值	不发行标记。
01 (BUF)	批量传送或者中断传送	不取决于设定值	如果 UACT=1 并且对应该管道的 FIFO 缓冲器是可发送 / 接收状态，就发行标记。 如果 UACT=0 或者不是可发送 / 接收状态，就不发行标记。
	等时传送	不取决于设定值	与对应该管道的 FIFO 缓冲器的状态无关，发行标记。
10 (STALL) 或者 11 (STALL)	不取决于设定值	不取决于设定值	不发行标记

表 21.14 PID 位对应的模块运行一览表 (选择功能控制器功能时)

PID 位	传送类型	传送方向 (DIR 位)	模块的运行
00 (NAK)	批量传送或者 中断传送	不取决于设定值	对 USB 主机的标记进行 NAK 应答。 但是, 有关 ATREPM=1 时的运行, 请参照 ATREPM 位的说明。
	等时传送	不取决于设定值	对 USB 主机的标记不进行应答。
01 (BUF)	批量传送	接收方向 (DIR=0)	对 USB 主机的 OUT 标记, 如果对应该管道的 FIFO 缓冲器是可接收状态, 就接收数据并且进行 ACK 应答; 如果不是可接收状态, 就进行 NAK 应答。 对 USB 主机的 PING 标记, 如果对应该管道的 FIFO 缓冲器是可接收状态, 就进行 ACK 应答; 如果不是可接收状态, 就进行 NYET 应答。
	中断传送	接收方向 (DIR=0)	对 USB 主机的 OUT 标记, 如果对应该管道的 FIFO 缓冲器是可接收状态, 就接收数据并且进行 ACK 应答; 如果不是可接收状态, 就进行 NAK 应答。
	批量传送或者 中断传送	发送方向 (DIR=1)	如果对应的 FIFO 缓冲器是可发送状态, 就对 USB 主机的标记发送数据; 如果不是可发送状态, 就进行 NAK 应答。
	等时传送	接收方向 (DIR=0)	对 USB 主机的 OUT 标记, 如果对应该管道的 FIFO 缓冲器是可接收状态, 就接收数据; 如果不是可接收状态, 就放弃数据。
发送方向 (DIR=1)		如果对应的 FIFO 缓冲器是可发送状态, 就对 USB 主机的标记发送数据; 如果不是可发送状态, 就发送 Zero-Length 包。	
10 (STALL) 或者	批量传送或者 中断传送	不取决于设定值	对 USB 主机的标记进行 STALL 应答。
11 (STALL)	等时传送	不取决于设定值	对 USB 主机的标记不进行应答。

(2) PIPEnCTR (n=6 ~ 9)

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BSTS	—	CSCLR	CSSTS	—	—	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W*1	R	R	R	R/W	R/W*1	R/W*1	R	R	R	R	R	R/W	R/W

位	位名	初始值	R/W	说明
15	BSTS	0	R	<p>缓冲器状态位</p> <p>表示相应管道的 FIFO 缓冲器状态。</p> <p>0: 不能存取缓冲器</p> <p>1: 能存取缓冲器</p> <p>此位的含义因 DIR、BFRE 和 DCLRM 位的设定值而不同, 如表 21.11 所示。</p>
14	—	0	R	<p>保留位</p> <p>读取值为 0, 只能写 0。</p>
13	CSCLR	0	R/W*1	<p>CSPLIT 状态清除位</p> <p>如果将此位置 1, 就清除相应管道的 CSSTS 位。</p> <p>0: 写无效</p> <p>1: 清除 CSSTS 位</p> <p>在使用分离事务的传送中, 如果要强制从 S-SPLIT 重新开始下一次传送, 就必须通过软件将此位置 1。正常的分离事务在 C-SPLIT 结束时自动将 CSSTS 位清 0, 因此不需要通过软件进行清除。</p> <p>必须在 UACT=0 并且停止通信时或者在检测到断开时不进行传送的情况下, 通过此位控制 CSSTS 位。</p> <p>在 CSSTS=0 时, 即使将此位置 1, 也保持 CSSTS=0。</p> <p>在选择功能控制器功能时, 必须将此位置 0。</p>
12	CSSTS	0	R	<p>CSSTS 状态位</p> <p>在选择主机控制器功能时, 此位表示分离事务的 C-SPLIT 状态。</p> <p>0: 正在进行 START-SPLIT (S-SPLIT) 事务处理或者正在进行不使用分离事务的传送</p> <p>1: 正在进行 C-SPLIT 事务处理</p> <p>在开始 C-SPLIT 时此位为 1, 在检测到 C-SPLIT 结束时此位为 0。</p> <p>此位只在选择主机控制器功能时有效。</p>
11、10	—	全 0	R	<p>保留位</p> <p>读取值为 0, 只能写 0。</p>
9	ACLRM	0	R/W	<p>缓冲器自动清除模式位 *3*4</p> <p>指定是禁止还是允许相应管道的缓冲器自动清除模式。</p> <p>0: 禁止缓冲器自动清除模式</p> <p>1: 允许缓冲器自动清除模式 (全部缓冲器被初始化)</p> <p>在要将分配给相应管道的 FIFO 缓冲器的内容全部删除时, 必须给 ACLRM 位连续写 1 和 0。</p> <p>给此位连续写 1 和 0 时的清除内容以及需要清除的情况如表 21.15 所示。</p> <p>必须在 CSSTS=0、PID=NAK 并且还未通过 CURPIPE 位设定相应管道时更改此位。</p> <p>在对应管道的 PID 位从 BUF 变为 NAK 后更改此位时, 必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是, 本模块在将 PID 位变为 NAK 时, 不需要通过软件确认 PBUSY 位。</p>

位	位名	初始值	R/W	说明
8	SQCLR	0	R/W*1	<p>交替位清除 *3*4</p> <p>在要清除相应管道的下一个事务的顺序交替位的期待值 (将期待值设定到 DATA0) 时, 将此位置 1。</p> <p>0: 无效 1: 指定 DATA0</p> <p>如果通过软件将此位置 1, 就将相应管道的顺序交替位的期待值设定到 DATA0。此位总是 0。</p> <p>在选择主机控制器功能时, 对于 Bulk-Out 传送管道, 如果将此位置 1, 就从 PING-token 开始进行该管道的下一次传送。</p> <p>必须在 CSCTS=0 和 PID=NAK 时将 SQCLR 位置 1。</p> <p>在对应管道的 PID 位从 BUF 变为 NAK 后将此位置 1 时, 必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是, 本模块在将 PID 位变为 NAK 时, 不需要通过软件确认 PBUSY 位。</p>
7	SQSET	0	R/W*1	<p>交替位置位 *3*4</p> <p>在要将相应管道的下一个事务的顺序交替位的期待值设定到 DATA1 时, 将为此置 1。</p> <p>0: 无效 1: 指定 DATA1</p> <p>如果通过软件将此位置 1, 就将相应管道的顺序交替位的期待值设定到 DATA1。此位总是 0。</p> <p>必须在 CSSTS=0 和 PID=NAK 时将 SQSET 位置 1。</p> <p>在对应管道的 PID 位从 BUF 变为 NAK 后将此位置 1 时, 必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是, 本模块在将 PID 位变为 NAK 时, 不需要通过软件确认 PBUSY 位。</p>
6	SQMON	0	R	<p>交替位确认</p> <p>表示相应管道的下一个事务的顺序交替位的期待值。</p> <p>0: DATA0 1: DATA1</p> <p>在相应管道的传送类型不为等时传送时, 如果事务被正常处理, 本模块就将此位取反。但是, 如果在接收方向传送时发生 DATA-PID 不一致, 就不将此位取反。</p>
5	PBUSY	0	R	<p>管道忙</p> <p>表示相应管道是否用于当前 USB 总线。</p> <p>0: 相应管道不用于 USB 总线 1: 相应管道用于 USB 总线</p> <p>在开始该管道的 USB 事务时, 此位从 0 变为 1; 在 1 个事务结束时, 此位从 1 变为 0。</p> <p>在通过软件设定 PID=NAK 后, 能通过读此位来确认管道是否变为能更改设定的状态。</p>
4 ~ 2	—	全 0	R	<p>保留位</p> <p>读取值为 0, 只能写 0。</p>

位	位名	初始值	R/W	说明
1、0	PID[1:0]	00	R/W	<p>应答 PID 指定相应管道的下一个事务的应答类型。</p> <p>00: NAK 应答 01: BUF 应答 (根据缓冲器状态) 10: STALL 应答 11: STALL 应答</p> <p>此位的默认值为 NAK。在通过相应管道进行 USB 传送时, 必须将此位更改为 BUF。各 PID 位的设定值所对应的模块基本运行 (通信包中无错误时的运行) 如表 21.13 和表 21.14 所示。</p> <p>当该管道处于 USB 通信过程中时, 如果通过软件将此位从 BUF 变为 NAK, 就必须在写 NAK 后, 通过 PBUSY=1 确认该管道的 USB 传送已转移到 NAK 状态。但是, 本模块在将此位变为 NAK 时, 不需要通过软件确认 PBUSY 位。</p> <p>在以下情况下更改此位的值:</p> <ul style="list-style-type: none"> • 在该管道为接收方向时并且通过软件将所选管道的 SHTNAK 位置 1 时, 如果本模块认为传送结束, 就设定 PID=NAK。 • 在接收到超过该管道的 MaxPacketSize 有效负载的数据包时, 就设定 PID=STALL (11)。 • 在选择功能控制器功能时, 如果检测到 USB 总线复位, 就设定 PID=NAK。 • 在选择主机控制器功能时, 如果连续 3 次检测到 CRC 错误等接收错误, 就设定 PID=NAK。 • 在选择主机控制器功能时, 如果接收到 STALL 握手信号, 就设定 PID=STALL (11)。 <p>按以下步骤进行此位的设定:</p> <ul style="list-style-type: none"> • 在从 NAK (00) 状态转移到 STALL 状态时, 必须置 10。 • 在从 BUF (01) 状态转移到 STALL 状态时, 必须置 11。 • 在从 STALL (11) 状态转移到 NAK 状态时, 必须写 10 后接着写 00。 • 在从 STALL 状态转移到 BUF 状态时, 必须在变为 NAK 状态后设定为 BUF 状态。

【注】 *1 读取值为 0, 只能写 1。

*2 只能写 1。

*3 在设定 ACLRM 位、SQCLR 位或者 SQSET 位时, 必须满足以下条件:
必须在 CSSTS=0 和 PID=NAK 以及还未通过 CURPIPE 设定所选管道时进行设定。

*4 在 PID 从 BUF 变为 NAK 后更改 ACLRM 位、SQCLR 位或者 SQSET 位的设定时, 必须在确认所选管道的 CSSTS=0 和 PBUSY=0 后进行。在通过 HW 控制将 PID 变为 NAK 时, 不需要确认 PBUSY 位。

表 21.15 ACLRM=1 时的清除内容

编号	通过 ACLRM 位操作的清除内容	需要清除的情况
1	分配给所选管道的 FIFO 缓冲器的全部内容	
2	在选择主机控制器功能并且所选管道的传送类型为中断传送时, 为间隔计数值。	在对间隔计数值进行复位时
3	与 BFRE 位有关的内部标志	在更改 BFRE 位的设定值时
4	与事务计数有关的内部标志	在强制结束事务计数功能时

21.3.37 管道 n 事务计数允许寄存器 (PIPE_nTRE) (n=1 ~ 5)

PIPE_nTRE 指定管道 1 ~ 5 所对应的事务计数器是否有效, 并且指定计数器的清除。
在上电复位时, 此寄存器被初始化。

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

位	位名	初始值	R/W	说明
15 ~ 10	—	全 0	R	保留位 读取值为 0, 只能写 0。
9	TRENB	0	R/W	事务计数器允许位 指定事务计数器是否有效。 0: 事务计数器功能无效 1: 事务计数器功能有效 对于接收管道, 如果通过软件在给 TRNCNT 位设定总信息包的数量后将此位置 1, 就在接收完 TRNCNT 位设定数量的信息包时, 进行以下控制: <ul style="list-style-type: none"> 在使用连续发送 / 接收模式 (CNTMD=1) 时, 即使在接收完成时 FIFO 缓冲器未滿, 也切换给 CPU。 在 SHTNAK=1 时, 如果接收完 TRNCNT 位设定数量的信息包, 就将对应管道的 PID 位变为 NAK。 在 BFRE=1 时, 如果接收完 TRNCNT 位设定数量的信息包并且读完最后的数据, 就产生 BRDY 中断。 对于发送管道, 必须将此位置 0。 在不使用事务计数功能时, 必须将此位置 0。 在使用事务计数功能时, 必须在将此位置 1 前设定 TRNCNT 位。另外, 必须在接收事务计数对象的最初的信息包前将此位置 1。
8	TRCLR	0	R/W	事务计数器清除位 清除对应该管道的事务计数器的当前计数值, 并且将此位置 0。 0: 无效 1: 清除当前计数器
7 ~ 0	—	全 0	R	保留位 读取值为 0, 只能写 0。

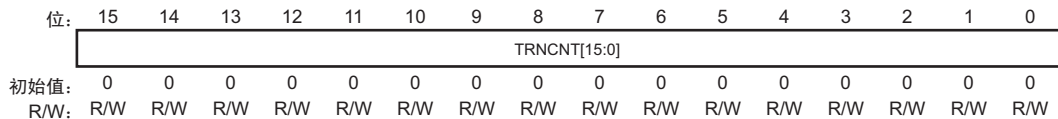
【注】 必须在 CSSTS=0 和 PID=NAK 时更改此寄存器的各位。

在对应管道的 PID 位从 BUF 变为 NAK 后更改各位的设定值时, 必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是, 本模块在将 PID 位变为 NAK 时, 不需要通过软件确认 PBUSY 位。

21.3.38 管道 n 事务计数寄存器 (PIPE_nTRN) (n=1 ~ 5)

PIPE_nTRN 是对应管道 1 ~ 5 的事务计数器。

在上电复位时, 此寄存器被初始化; 在 USB 总线复位时, 保持设定值。



位	位名	初始值	R/W	说明
15 ~ 0	TRNCNT	全 0	R/W	<p>事务计数器</p> <p>写时:</p> <p style="padding-left: 20px;">设定 DMA 传送的事务次数。</p> <p>读时:</p> <p style="padding-left: 20px;">在 TREN_B=0 时, 表示设定的事务次数。</p> <p style="padding-left: 20px;">在 TREN_B=1 时, 表示计数中的事务次数。</p> <p>如果接收时的状态满足以下全部条件, 此计数器就加 1:</p> <ul style="list-style-type: none"> • TREN_B=1 • 在接收信息包时, TRC_NT 设定值 ≠ 当前计数值 +1 • 接收的信息包的有效载荷和 MXPS 位的设定值相同 <p>在满足以下任意条件时, 此计数器就清 0:</p> <ul style="list-style-type: none"> • 满足以下全部条件时 TREN_B=1 <li style="padding-left: 40px;">在接收信息包时, TRC_NT 设定值 = 当前计数值 +1 <li style="padding-left: 40px;">接收的信息包的有效载荷和 MXPS 位的设定值相同 • 满足以下全部条件时 TREN_B=1 <li style="padding-left: 40px;">接收到短包 • 满足以下全部条件时 TREN_B=1 <li style="padding-left: 40px;">通过软件将 TRCLR 位置 1 <p>对于发送管道, 必须将此位置 0。</p> <p>在不使用事务计数功能时, 必须将此位置 0。</p> <p>必须在 CSSTS=0、PID=NAK 和 TREN_B=0 时更改此位。</p> <p>在对应管道的 PID 位从 BUF 变为 NAK 后更改此位时, 必须在确认 CSSTS=0 和 PBUSY=0 后进行。但是, 本模块在将 PID 位变为 NAK 时, 不需要通过软件确认 PBUSY 位。</p> <p>在更改此位的值时, 必须在设定 TREN_B=1 前设定 TRCNT=1。</p>

21.3.39 设备地址 n 配置寄存器 (DEVADDn) (n=0 ~ A)

对于管道 0 ~ A，DEVADDn 指定 HUB（连接通信对象的外围设备）的地址，端口号和通信速度。
在选择主机控制器功能时，必须在开始进行各管道的通信前设定此寄存器的各位。

必须在不存在使用此位设定的有效管道时，更改此寄存器的各位。有效管道是指满足以下 2 个条件的管道。

1. 在通过 DEVSEL 位指定此寄存器时
2. 在将所选管道的 PID 位设定为 BUF 时或者在所选管道为 DCP 并且 SUREQ=1 时
在上电复位时，此寄存器被初始化。

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

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读取值为 0，只能写 0。
14 ~ 11	UPPHUB[3:0]	0000	R/W	通信对象连接的 HUB 寄存器 设定 HUB（连接通信对象的外围设备）的 USB 地址。 0000：外围设备直接连接本 LSI 端口 0001 ~ 1010：HUB 的 USB 地址 1011 ~ 1111：禁止设定 当选择主机控制器功能时，在执行分离事务时参照此位的设定值生成信息包。 在选择功能控制器功能时，必须置 0000。
10 ~ 8	HUBPORT[2:0]	000	R/W	通信对象连接的 HUB 端口 设定 HUB（连接通信对象外围设备）的端口号。 000：外围设备直接连接本 LSI 的端口 001 ~ 111：HUB 的端口号 当选择主机控制器功能时，在执行分离事务时参照此位的设定值生成信息包。 在选择功能控制器功能时，必须置 000。
7、6	USBSPD[1:0]	00	R/W	通信对象设备的传送速度 设定通信对象的外围设备的 USB 传送速度。 00：不使用 DEVADDn 寄存器 01：低速 10：全速 11：高速 当选择主机控制器功能时，参照此位的设定值生成信息包。 在选择功能控制器功能时，必须置 00。
5 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

21.4 运行说明

21.4.1 系统控制和振荡控制

本节说明本模块的初始设定所需的寄存器操作以及功耗控制所需的寄存器。

(1) 复位

本模块的复位种类如表 21.16 所示，有关各复位运行后的寄存器初始状态，请参照“21.3 寄存器说明”。

表 21.16 复位种类一览表

名称	操作
上电复位	从 $\overline{\text{PRESET}}$ 引脚输入低电平。
USB 总线复位	在选择功能控制器功能时，自动检测 D+ 和 D- 线路。

(2) 控制器功能的选择设定

能选择主机控制器功能或者功能控制器功能。通过 SYSCFG 寄存器的 DCFM 位进行控制器功能的选择。必须在上电复位后的初始设定或者在禁止 D+ 上拉 (DPRPU=0) 和 D+/D- 下拉 (DRPD=0) 的状态下进行 DCFM 位的设定。

(3) 高速运行的允许

本模块能通过软件设定 USB 通信速度 (通信位速率)。

在选择主机控制器功能时，能设定为高速运行或者全速 / 低速运行；在选择功能控制器功能时，能选择高速运行或者全速运行。在允许高速运行时，必须将 SYSCFG 寄存器的 HSE 位置 1。如果允许高速运行，本模块就执行复位握手协议，自动设定 USB 通信速度。能通过 DVSTCTR 寄存器的 RHST 位来确认复位握手信号的结果。

在禁止高速运行的情况下，本模块在选择主机控制器功能时进行全速或者低速运行；在选择功能控制器功能时只能进行全速运行。

在选择主机控制器功能时，必须在检测到连接后并且在进行总线复位前进行 HSE 位的变更；在选择功能控制器功能时，必须在禁止 D+ 上拉 (DPRPU=0) 的状态下进行 HSE 位的变更。

(4) USB 数据总线的电阻控制

本模块和 USB 连接器的连接图如图 21.1 所示。

内置 D+ 信号的上拉电阻和 D+/D- 信号的下拉电阻。必须通过 SYSCFG 寄存器的 DPRPU 和 DRPD 位进行各信号的上拉和下拉设定。

能对 D+/D- 信号进行高速运行时的终端电阻和全速运行时的输出电阻进行控制。通过复位握手信号、挂起和恢复的检测，自动切换与主机控制器或者外围设备连接后的内部电阻。

如果选择功能控制器功能并且在与主机控制器进行通信的过程中将 SYSCFG 寄存器的 DPRPU 位置 0，就禁止 USB 数据线的上拉电阻 (或者终端电阻)，因此能将“设备断开”通知 USB 主机。

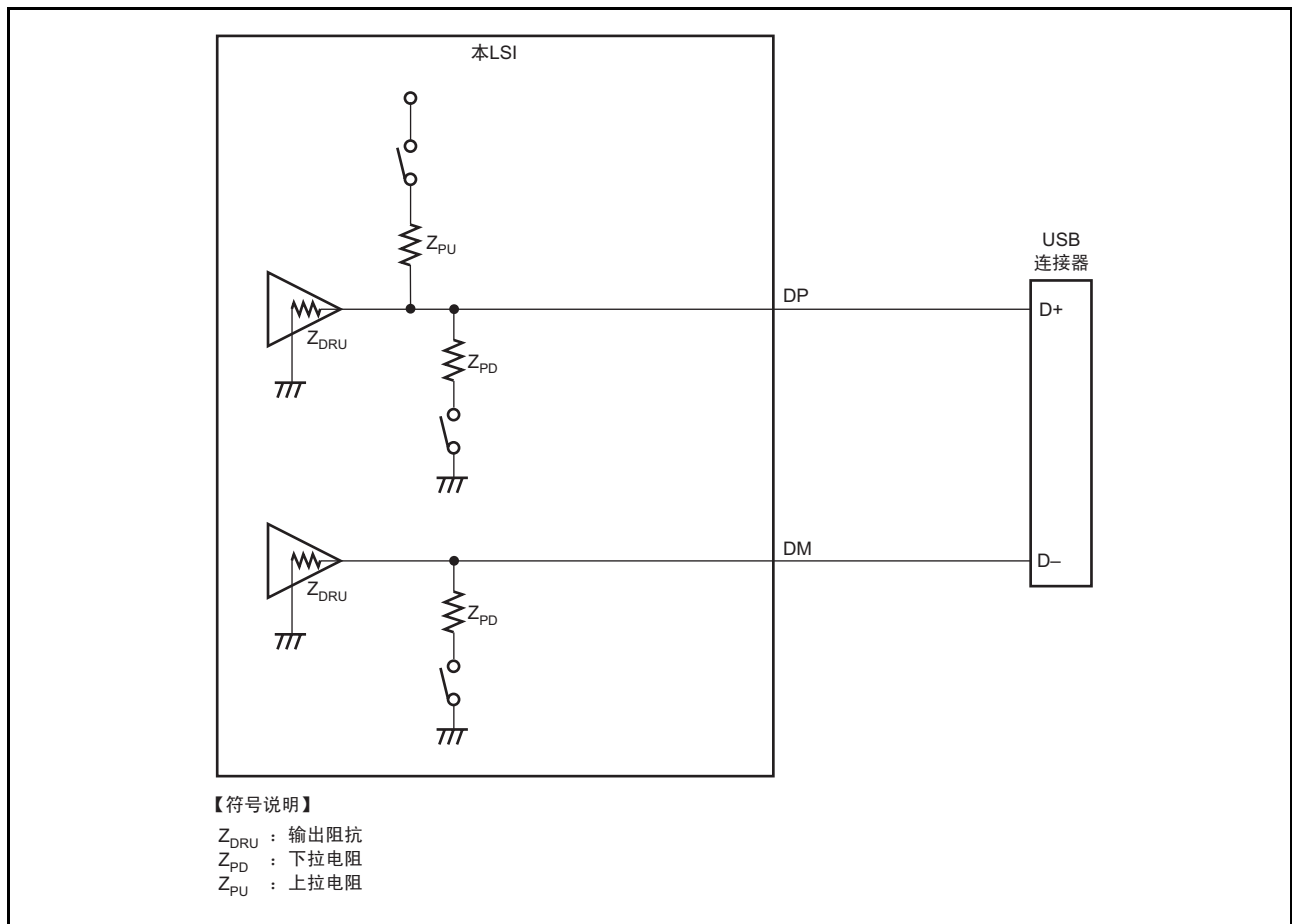


图 21.1 USB 连接器的连接图

21.4.2 中断功能

本模块产生中断的条件如表 21.17 所示。

在这些中断产生条件成立并且通过对应的中断允许寄存器被设定为允许中断输出时，就向中断控制器 (INTC) 请求 USB 中断。

表 21.17 中断产生条件一览表

位	中断名	中断产生条件	产生的功能	相关状态
VBINT	VBUS 中断	<ul style="list-style-type: none"> 检测到 VBUS 输入引脚的状态变化。 (L→H 或者 H→L 的变化) 	主机功能	VBSTS
RESM	恢复中断	<ul style="list-style-type: none"> 在挂起状态下检测到 USB 总线的状态变化。 (J-State→K-State 或者 J-State→SE0) 	功能	—
SOFR	帧号更新中断	<ul style="list-style-type: none"> [在选择主机控制器功能时] 发送完不同帧号的 SOF 包。 [在选择功能控制器功能时] SOFRM=0: 接收到不同帧号的 SOF 包。 SOFRM=1: 因包破损等而不能接收 μ 帧号为 0 的 SOF。 	主机功能	—

位	中断名	中断产生条件	产生的功能	相关状态
DVST	设备状态转移中断	<ul style="list-style-type: none"> 检测到设备状态的转移。 检测到 USB 总线复位。 检测到挂起状态。 接收到 SET_ADDRESS 请求。 接收到 SET_CONFIGURATION 请求。 	功能	DVSQ
CTRT	控制传送阶段转移中断	<ul style="list-style-type: none"> 检测到控制传送的阶段转移。 设置阶段结束。 控制写传送状态阶段转移。 控制读传送状态阶段转移。 控制传送结束。 发生控制传送顺序错误。 	功能	CTSQ
BEMP	缓冲器空中断	<ul style="list-style-type: none"> 发送完缓冲存储器中的全部数据并且缓冲器变空。 接收到超过最大包长度的包。 	主机 功能	BEMPSTS PIPEBEMP
NRDY	缓冲器忙中断	<p>[在选择主机控制器功能时]</p> <ul style="list-style-type: none"> 对于发行的标记, 接收到外围设备的 STALL。 对于发行的标记, 不能正确接收外围设备的应答 (连续 3 次无应答或者连续 3 次发生包接收错误)。 在等时传送时, 发生超限 / 欠载。 <p>[选择功能控制器功能时]</p> <ul style="list-style-type: none"> 对 IN 标记 /OUT 标记 /PING 标记, 应答了 NAK。 在通过等时传送接收数据时, 发生 CRC 错误或者位填充错误。 在通过等时传送接收数据时, 发生超限 / 欠载。 	主机 功能	NRDYSTS PIPENRDY
BRDY	缓冲器就绪中断	<ul style="list-style-type: none"> 缓冲器就绪 (可读写状态) 	主机 功能	BRDYSTS PIPEBRDY
BCHG	总线变化中断	<ul style="list-style-type: none"> 检测到 USB 总线的状态变化。 	主机、功能	—
DTCH	全速运行时的断开检测	<ul style="list-style-type: none"> 在全速运行时, 检测到外围设备断开。 	主机	—
ATTCH	设备连接检测	<ul style="list-style-type: none"> 检测到 USB 总线状态为连续 2.5μs 的 J-STATE 或者连续 2.5μs 的 K-STATE。能用于检测外围设备的连接。 	主机	—
EOFERR	EOF 错误检测	<ul style="list-style-type: none"> 检测到外围设备的 EOF 错误。 	主机	—
SACK	SETUP 正常	<ul style="list-style-type: none"> 接收到设置事务的正常应答 (ACK)。 	主机	—
SIGN	SETUP 错误	<ul style="list-style-type: none"> 连续 3 次检测到设置事务错误 (无应答或者 ACK 包破损)。 	主机	—

【注】 未表示寄存器名的位的寄存器都是 INTSTS0。

本模块的中断结构图如图 21.2 所示。

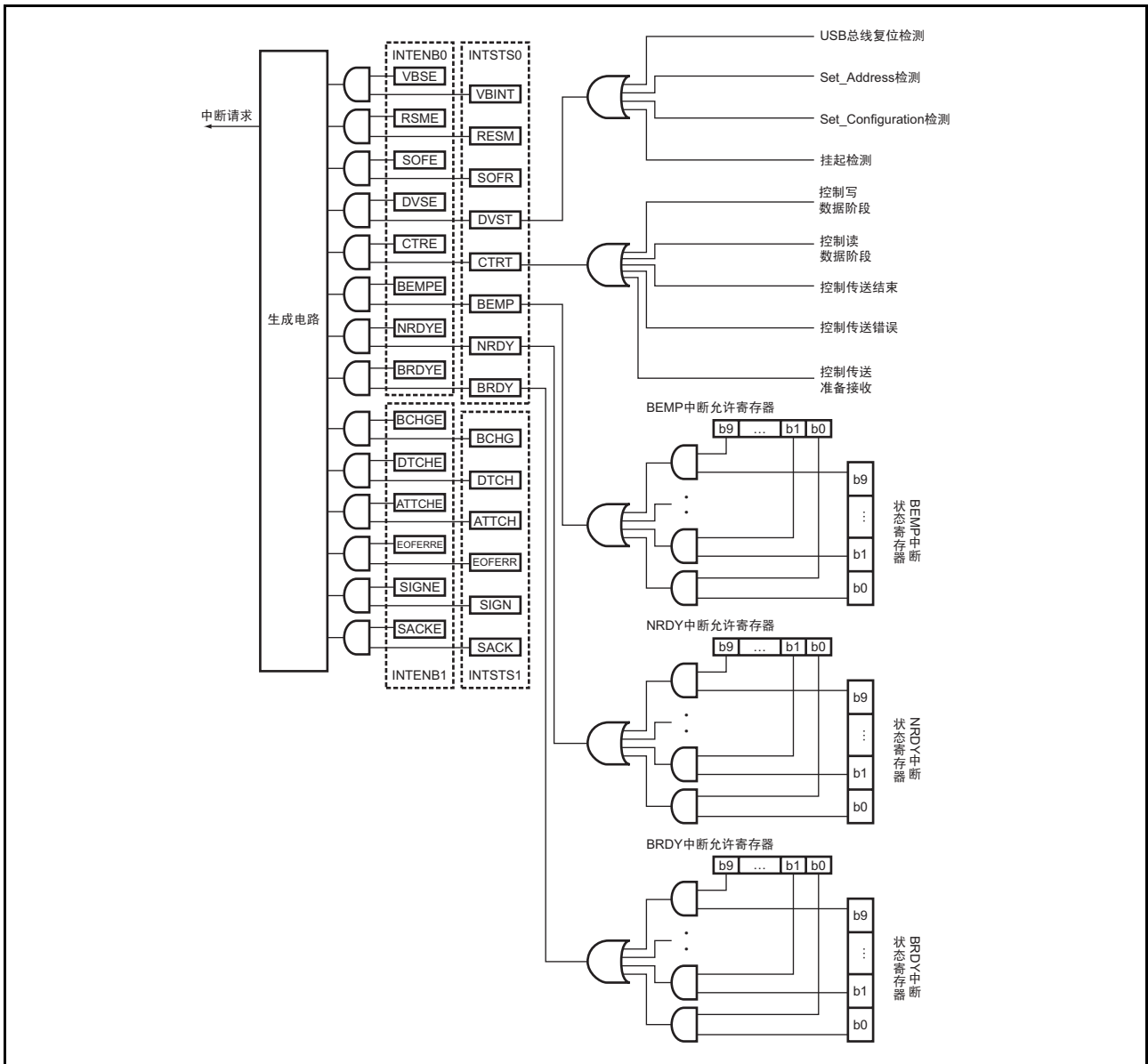


图 21.2 中断结构图

(1) BRDY 中断

在选择主机控制器功能或者功能控制器功能时，都产生 BRDY 中断。在各管道满足下述条件时，将 BRDYSTS 寄存器的对应位置 1。此时，如果已通过软件将相应管道对应的 BRDYENB 寄存器的 PIPEBRDYE 位置 1 并且将 INTENB0 寄存器的 BRDYE 位置 1，就产生 BRDY 中断。

根据 BRDYM 位和各管道的 BFRE 位的设定，BRDY 中断的产生条件和清除方法不同。

BRDYM=0 和 BFRE=0 时在此设定的情况下，BRDY 中断是表示能存取 FIFO 端口的中断。

在满足下述所示的任意一个条件时，产生内部 BRDY 中断请求的触发，并将产生请求触发的管道所对应的 PIPEBRDY 位置 1。

1. 设定为发送方向的管道

- 通过软件将 DIR 位从 0 变为 1。
- 在 CPU 不能写分配给该管道的 FIFO 缓冲器的状态下 (BSTS 位的读取值为 0 时), 该管道的信息包发送结束。
在连续发送/接收模式的情况下, 当一个 FIFO 缓冲器面的数据发送结束时, 产生请求触发。
- 在将 FIFO 缓冲器设定为双缓冲器的情况下, 当写完一个 FIFO 缓冲器时另一个 FIFO 缓冲器变空。
如果在写 FIFO 缓冲器的过程中另一个缓冲器发送结束, 就在当前正在写的缓冲器结束写数据后产生请求触发。
- 等时传送的管道通过硬件清除缓冲器。
- 通过将 ACLRM 位置 1, FIFO 缓冲器从写禁止状态变为可写状态。

对于 DCP (即, 控制传送时的数据发送), 不产生请求触发。

2. 设定为接收方向的管道

- 在 CPU 不能读分配给该管道的 FIFO 缓冲器的状态下 (BSTS 位的读取值为 0 时), 信息包的接收正常结束, 并且 FIFO 缓冲器变为可读状态。
对于数据 PID 不一致的事务, 不产生请求触发。
在连续发送/接收模式的情况下, 当数据长度为 MaxPacketSize 并且缓冲器有空间时, 不产生请求触发。
在接收到短包时, 即使 FIFO 缓冲器有空间也产生请求触发。
在使用事务计数器的情况下, 当接收到设定数量的信息包时, 产生请求触发。此时, 即使 FIFO 缓冲器中有空间也产生请求触发。
- 在 FIFO 缓冲器为双缓冲器的情况下, 当读完一个 FIFO 缓冲器时另一个 FIFO 缓冲器变为可读状态。
如果在读取时另一个缓冲器接收结束, 就在当前正在读的缓冲器结束读数据后产生请求触发。

选择功能控制器功能时, 在控制传送的状态阶段进行通信时不产生该中断。

通过软件将相应管道所对应的 BRDYSTS 寄存器的 PIPEBRDY 位置 0, 能清除该管道的 PIPEBRDY 中断状态。此时, 必须给其它管道对应的位置 1。

必须在存取 FIFO 缓冲器前清除此中断的状态。

(a) BRDYM=0 和 BFRE=1 时

在此设定的情况下, 如果对接收管道读完一次传送的全部数据, 就产生 BRDY 中断, 并且将此寄存器的对应该管道的位置 1。

在以下任一种情况下, 判断为接收到 1 次传送的最后数据。

- 接收到包含 Zero-Length 包的短包。
- 使用事务计数器 (TRNCNT 位) 并且接收到 TRNCNT 位设定数量的信息包。

如果在满足上述判断条件后读完此数据, 就判为读完 1 次传送的全部数据。

如果在 FIFO 缓冲器为空的状态下接收到 Zero-Length 包, 就在将 Zero-Length 包数据交给 CPU 时判断为读完 1 次传送的全部数据。此时, 为了开始下一次传送, 必须通过软件将对应的 FIFOCTR 寄存器的 BCLR 位置 1。

对于发送管道, 此设定不检测 BRDY 中断。

通过软件将相应管道对应的 PIPEBRDY 位置 0, 能清除该管道的 PIPEBRDY 中断状态。此时, 必须给其它管道对应的位置 1。

在使用此模式时, 不能在 1 个传送的数据处理结束前更改 BFRE 位的设定值。如果要在处理过程中更改 BFRE 位, 就必须通过 ACLRM 位清除全部对应管道的 FIFO 缓冲器。

(b) BRDYM=1 和 BFRE=0 时

在此设定的情况下，PIPEBRDY 位的值和各管道的 BSTS 位连动。即，根据 FIFO 缓冲器的状态 BRDY 中断状态变为 1 或者 0。

1. 设定为发送方向的管道

BRDY 中断状态在能给 FIFO 端口写数据的状态下为 1；在不能写状态下为 0。
但是，即使 DCP 的发送管道为可写状态，也不产生 BRDY 中断。

2. 设定为接收方向的管道

BRDY 中断状态在能从 FIFO 端口读数据的状态下为 1；在读完全部数据（不可读状态）时为 0。
如果在 FIFO 缓冲器为空的状态下接收到 Zero-Length 包，就在通过软件设定 BCLR=1 前此位为 1，继续产生 BRDY 中断。
在此设定的情况下，不能通过软件将 PIPEBRDY 位清 0。在 BRDYM=1 时，必须将 BFRE 位全部（全部管道）置 0。

BRDY 中断产生时序图如图 21.3 所示。

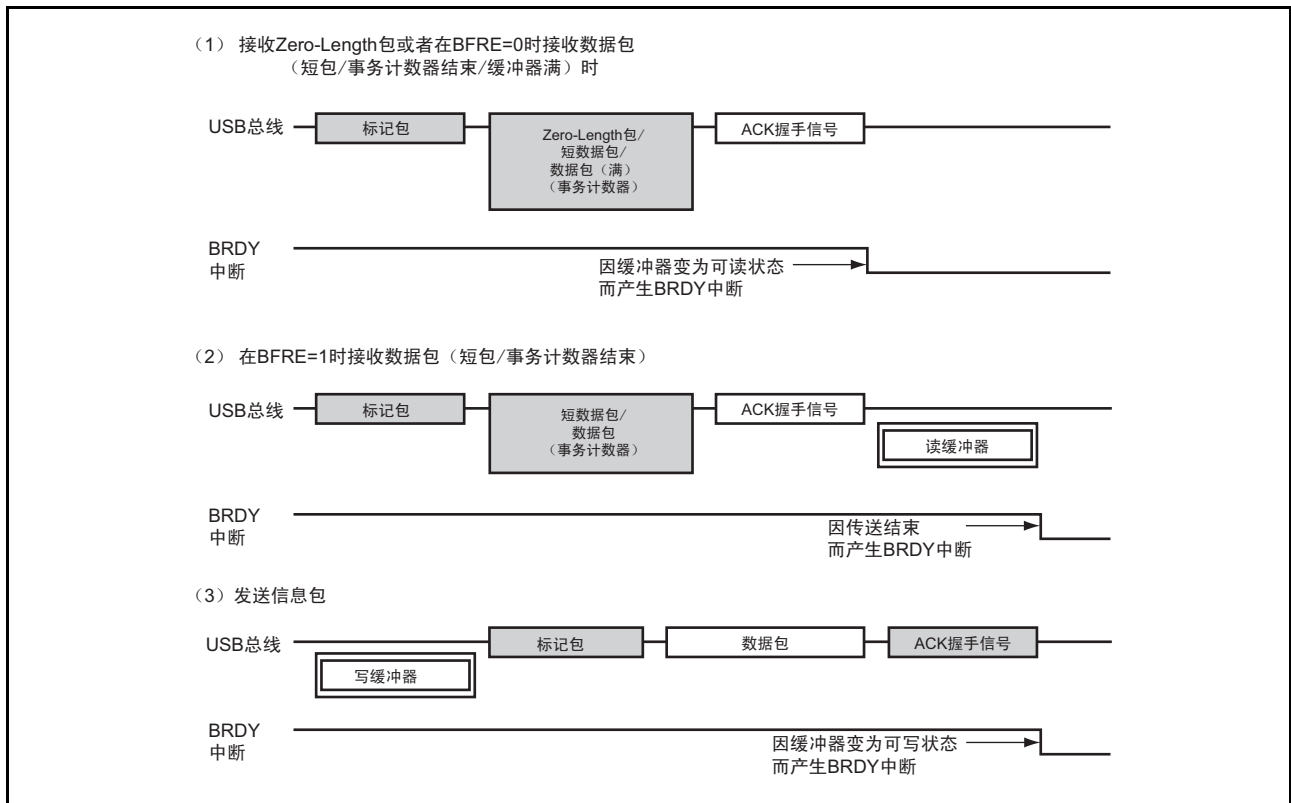


图 21.3 BRDY 中断产生时序图

(2) NRDY 中断

对于由软件设定 PID=BUF 的管道，如果产生内部 NRDY 中断请求，就将 NRDYSTS 寄存器的 PIPENRDY 位的对应位置 1。此时，如果已通过软件将 NRDYENB 寄存器的对应位置 1，就将 INTSTS0 寄存器的 NRDY 位置 1，并且产生 USB 中断。

本模块对管道产生内部 NRDY 中断请求的条件如下。

但是，以下的中断产生条件不适合选择主机控制器功能时的 SETUP 事务。选择主机控制器功能时的 SETUP 事务检测 SACK 中断或者 SIGN 中断。

在选择功能控制器功能并且执行控制传送状态阶段时，不产生中断请求。

(a) 选择主机控制器功能并且不发生分离事务的连接

1. 发送方向的管道

在满足以下任意一个条件时，检测 NRDY 中断：

- 对于等时传送的管道，在 FIFO 缓冲器中没有发送数据的状态下到了发行 OUT 标记的时间。
此时，在 OUT 标记之后继续发送 Zero-Length 包，并且将 PIPENRDY 位的对应位置 1，OVRN 位置 1。
- 对于非等时传送的管道以及 SETUP 事务以外的通信，连续 3 次发生以下任意情况的组合：外围设备无应答（在未检测到外围设备的 Handshake 包的情况下检测到超时）；从外围设备的信息包中检测到错误。
此时，将 PIPENRDY 位的对应位置 1，并且将对应管道的 PID 位变为 NAK。
- 对于 SETUP 事务以外的通信，从外围设备接收到 STALL Handshake（对于 OUT 和 PING 的 STALL）。
此时，将 PIPENRDY 位的对应位置 1，并且将对应管道的 PID 位变为 STALL（11）。

2. 接收方向的管道

- 对于等时传送的管道，在 FIFO 缓冲器中没有空间的状态下到了发行 IN Token 的时间。
此时，放弃接收数据（对于 IN Token），并且将对应该管道的 PIPENRDY 位置 1，OVRN 位置 1。
另外，如果从接收数据（对于 IN Token）中检测到包错误，就将 CRCE 位置 1。
- 对于非等时传送的管道，连续 3 次发生以下任意情况的组合：外围设备对本模块发行的 IN Token 无应答（在未检测到外围设备的 DATA 包的情况下检测到超时）；从外围设备的信息包中检测到错误。
此时，将对应该管道的 PIPENRDY 位置 1，并且将对应管道的 PID 位变为 NAK。
- 对于等时传送的管道，外围设备对 IN Token 无应答（在未检测到外围设备的 DATA 包的情况下检测到超时）或者从外围设备的信息包中检测到错误。
此时，将对应该管道的 PIPENRDY 位置 1（不更改对应管道的 PID 位）。
- 对于等时传送的管道，从接收的数据包中检测到 CRC 错误或者位填充错误。
此时，将对应该管道的 PIPENRDY 位置 1，并且将 CRCE 位置 1。
- 接收到 STALL Handshake。
此时，将对应该管道的 PIPENRDY 位置 1，并且将对应管道的 PID 位变为 STALL。

(b) 选择主机控制器功能并且发生分离事务的连接

1. 发送方向的管道

- 对于等时传送的管道，在 FIFO 缓冲器中没有发送数据的状态下到了发行 OUT 标记的时间。
此时，将发行 Start-Split 事务（S-SPLIT）时的相应管道所对应的 PIPENRDY 位置 1，并且将 OVRN 位置 1。在 OUT 标记之后继续发送 Zero-Length 包。
- 对于非等时传送的管道，连续 3 次发生以下任意情况的组合：HUB 对 S-SPLIT 或者 Complete-Split 事务（C-SPLIT）无应答（在未检测到 HUB 的 Handshake 包的情况下检测到超时）；从 HUB 的信息包中检测到错误。

此时，将对应该管道的 PIPENRDY 位置 1，并且将对应该管道的 PID 位变为 NAK。

如果在发行 C-SPLIT 时检测到 NRDY 中断，就将 CSSTS 位清 0。

- 接收到对 C-SPLIT 的 STALL Handshake。

此时，将对应该管道的 PIPENRDY 位置 1，将对应该管道的 PID 位变为 STALL (11)，并且将 CSSTS 位清 0。

但是，SETUP 事务不检测此中断。

- 对于中断传送的管道，对 microFrame 号=4 的 C-SPLIT，接收到 NYET。

此时，将对应该管道的 PIPENRDY 位置 1，并且将 CSSTS 位清 0（不更改对应管道的 PID 位）。

2. 接收方向的管道

- 对于等时传送的管道，在 FIFO 缓冲器中没有空间的状态下到了发行 IN Token 的时间。

此时，将在发行 S-SPLIT 时的相应管道所对应的 PIPENRDY 位置 1，并且将 OVRN 位置 1。放弃的接收数据（对于 IN Token）。

- 对于批量传送的管道的传送或者 DCP 的 SETUP 事务以外的传送，连续 3 次发生以下任意情况的组合：在发行 S-SPLIT 或者 C-SPLIT 时，HUB 对本模块发行的 IN Token 无应答（在未检测到 HUB 的 DATA 包的情况下检测到超时）；从 HUB 的信息包中检测到错误。

此时，将对应该管道的 PIPENRDY 置 1，并且将对应该管道的 PID 位变为 NAK。如果在发行 C-SPLIT 时发生这种情况，就将 CSSTS 位清 0。

- 对于等时或者中断传送的管道的 C-SPLIT，连续 3 次发生以下任意情况的组合：HUB 对本模块发行的 IN Token 无应答（在未检测到 HUB 的 DATA 包的情况下检测到超时）；从 HUB 的信息包中检测到错误。

如果是中断传送的管道发生这种情况，将对应该管道的 PIPENRDY 位置 1，将对应该管道的 PID 位变为 NAK，并且将 CSSTS 位清 0。

如果是等时传送的管道发生这种情况，将对应该管道的 PIPENRDY 位置 1，并且将 CRCE 位置 1，CSSTS 位清 0（不更改管道的 PID 位）。

- 对于非等时传送的管道的 C-SPLIT，接收到 STALL Handshake。

此时，将对应该管道的 PIPENRDY 位置 1，将对应该管道的 PID 位变为 STALL (11)，并且将 CSSTS 位清 0。

- 对于等时/中断传送的管道的 C-SPLIT，接收到 microFrame=4 的 NYET Handshake。

此时，将对应该管道的 PIPENRDY 位置 1，并且将 CRCE 位置 1，CSSTS 位清 0（不更改管道的 PID 位）。

(c) 选择功能控制器功能

1. 发送方向的管道

- 在 FIFO 缓冲器中没有发送数据的状态下接收到 IN Token。

在接收 IN Token 时，产生 NRDY 中断请求，并且将 PIPENRDY 位置 1。当产生中断的管道的传送类型为等时传送时，就发送 Zero-Length 包，并且将 OVRN 位置 1。

2. 接收方向的管道

- 在 FIFO 缓冲器中没有空间状态下接收到 OUT 标记。

当产生中断的管道的传送类型为等时传送时，就在接收 OUT 标记时产生 NRDY 中断请求，并且将 PIPENRDY 位置 1，OVRN 位置 1。

当产生中断的管道的传送类型不为等时传送时，就在接收 OUT 标记后的数据之后的发送 NAK Handshake 时，产生 NRDY 中断请求，并且将 PIPENRDY 位置 1。

但是，在重新发送（发生 DATA-PID 不一致时）或者 DATA 包中有错误时，不产生 NRDY 中断请求。

- 在 FIFO 缓冲器中没有空间的状态下接收到 PING 标记。
在接收 PING 标记时，产生 NRDY 中断请求，并且将 PIPENRDY 位置 1。
- 对于等时传送的管道，在间隔帧内没有正常接收到标记。
在接收 SOF 时，产生 NRDY 中断请求，并且将 PIPENRDY 位置 1。

选择功能控制器功能时的 NRDY 中断产生时序图如图 21.4 所示。

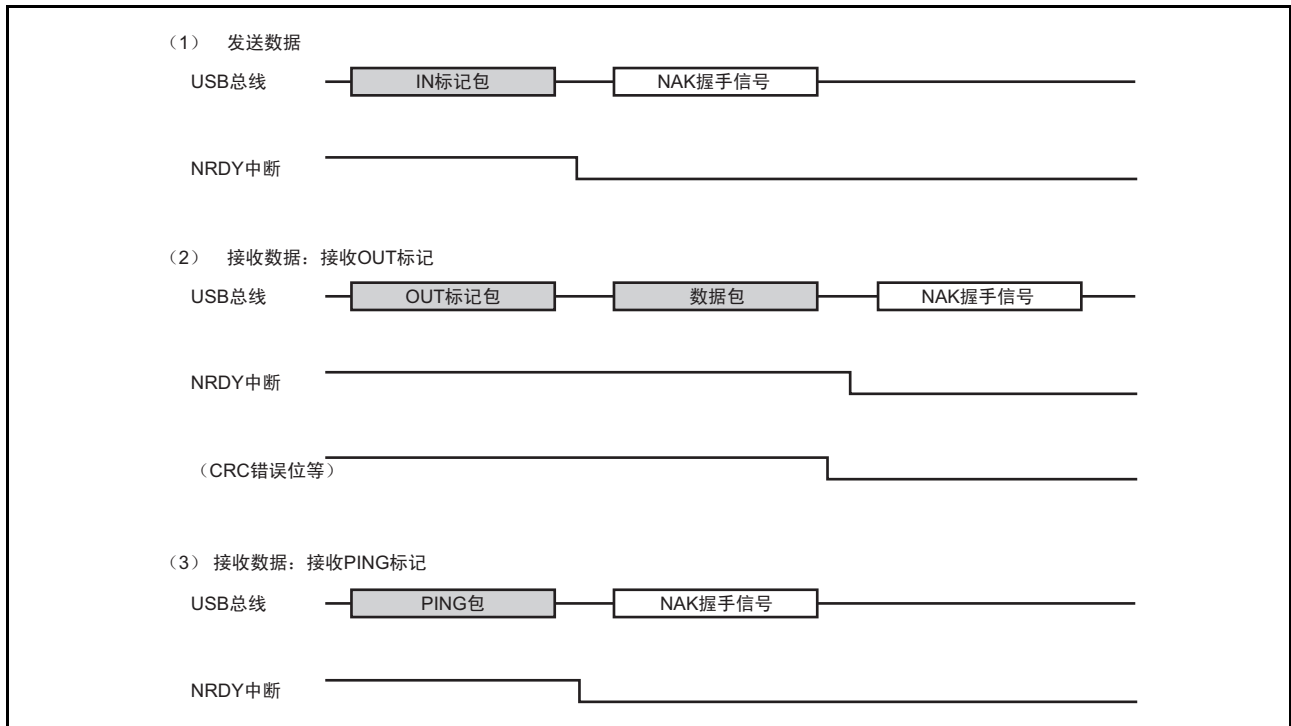


图 21.4 选择功能控制器功能时的 NRDY 中断产生时序图

(3) BEMP 中断

对于由软件设定为 PID=BUF 的管道，如果检测到 BEMP 中断，就将 BEMPSTS 寄存器的 PIPEBEMP 位的对应位置 1。此时，如果已通过软件将 BEMPENB 寄存器的对应位置 1，就将 INTSTS0 寄存器的 BEMP 位置 1，并且产生 USB 中断。

在以下情况下，产生内部 BEMP 中断请求：

1. 发送方向的管道

在发送结束时（包括发送 Zero-Length 包时），对应管道的 FIFO 缓冲器变空。

此时，如果设定为单缓冲器，就在产生 BRDY 中断的同时对 DCP 以外的管道产生内部 BEMP 中断请求。但是，在以下情况下不产生内部 BEMP 中断请求：

- 在设定为双缓冲器并且发送完 1 个缓冲器面的数据时，软件（DMAC）已经开始写 CPU 的 FIFO 缓冲器。
- 通过将 ACLRM 位或者 BCLR 位置 1，已清除（空）缓冲器。
- 在设定功能控制器功能时，已进行控制传送 Status 阶段的 IN 传送（发送 Zero-Length 包）。

2. 接收方向的管道

正常接收到大于MaxPacketSize设定值的数据。

此时，产生BEMP中断请求，并且将PIPEBEMP位的对应位置1，放弃接收数据以及将对应管道的PID位变为STALL（11）。另外，在设定主机控制器功能时不应答；在设定功能控制器功能时进行STALL应答。

但是，以下情况下不产生内部BEMP中断请求：

- 从接收数据检测到CRC错误或者位填充错误等
- 执行SETUP事务

如果给PIPEBEMP位写0，就能清除状态。但是，给PIPEBEMP位的写1无效。

选择功能控制器功能时的BEMP中断产生时序图如图21.5所示。

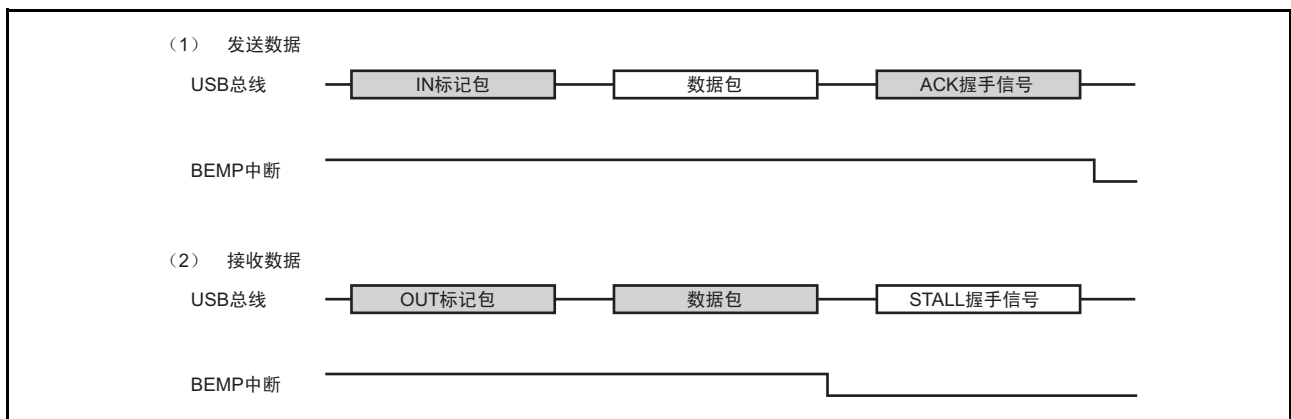


图 21.5 选择功能控制器功能时的 BEMP 中断产生时序图

(4) 设备状态转移中断

本模块的设备状态转移图如图 21.6 所示。本模块管理设备状态，产生设备状态转移中断。但是，通过恢复中断进行检测从挂起的恢复（恢复信号检测）。能通过 INTENB0 寄存器独立设定允许或者禁止设备状态转移中断，能通过 INTSTS0 寄存器的 DVSQ 位确认转移后的设备状态。

在转移到默认状态时，在结束复位握手协议之后，产生设备状态转移中断。

设备状态的管理和设备状态转移中断的产生只在选择功能控制器功能时进行。

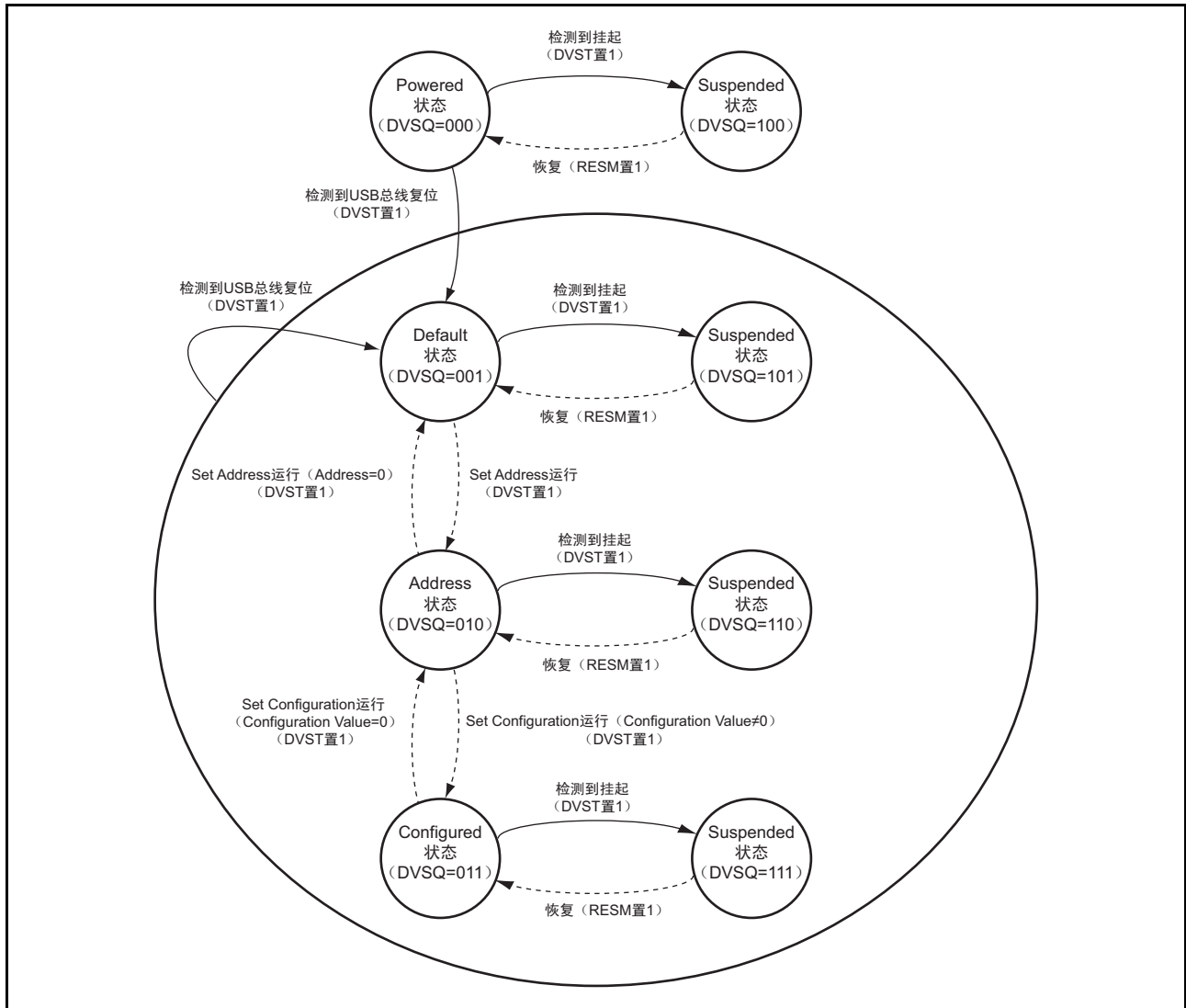


图 21.6 设备状态转移图

(5) 控制传送阶段转移中断

本模块的控制传送阶段转移图如图 21.7 所示。本模块管理控制传送的顺序，产生控制传送阶段转移中断。能通过 INTENB0 寄存器独立设定允许或者禁止控制传送阶段转移中断，能通过 INTSTS0 寄存器的 CTSQ 位确认转移后的传送阶段。

控制传送阶段转移中断的产生只在选择功能控制器功能时进行。

控制传送的顺序错误如下所示。在发生错误时，DCPCTR 寄存器的 PID 位变为 B'1x（应答 STALL）。

1. 控制读传送时

- 对于数据阶段的 IN 标记，在一次也没有进行数据传送的状态下接收到 OUT 或者 PING 标记。

- 在状态阶段接收到IN标记。
 - 在状态阶段接收到DATAPID=DATA0的数据包。
2. 控制写传送时
 - 对于数据阶段的OUT标记，在一次也没有进行ACK应答的状态下接收到IN标记。
 - 在数据阶段接收到DATAPID=DATA0的最初的数据包。
 - 在状态阶段接收到OUT或者PING标记。
 3. 控制写无数据控制传送时
 - 在状态阶段接收到OUT或者PING标记。

在控制写传送数据阶段，如果接收数据的个数超过 USB 请求的 wLength 值，就不能识别是控制传送顺序错误。在控制读传送状态阶段，如果接收到 Zero-Length 包以外的包，就进行 ACK 应答并且正常结束。

当因顺序错误而产生 CTRT 中断时 (SERR=1)，就在系统设定 CTRT=0 (清除中断状态) 前保持 CTSQ=110 的值。因此，在保持 CTSQ=110 的状态下，即使接收到新的 USB 请求，也不产生设置阶段结束的 CTRT 中断 (本模块保持设置阶段结束，在通过软件清除中断状态后产生设置阶段结束中断)。

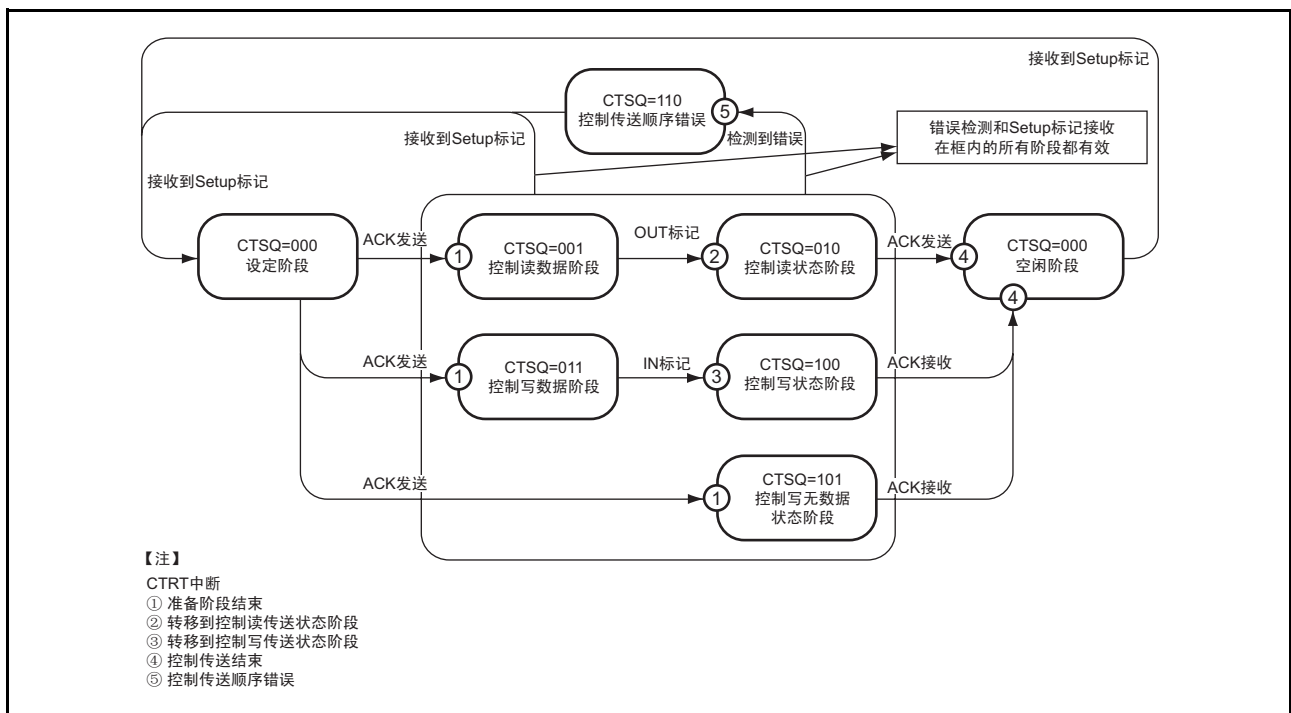


图 21.7 控制传送阶段转移图

(6) 帧更新中断

本模块的 SOFR 中断输出时序例子如图 21.8 所示。

在选择主机控制器功能时，在更新帧号时产生中断；在选择功能控制器功能时，在更新帧号时产生 SOFR 中断。

在选择功能控制器功能时，如果在全速运行中检测到新的 SOF 包，就更新帧号，并且产生 SOFR 中断。但是，如果在高速运行中不为 μ SOF 锁定状态，就不更新帧号，也不产生 SOFR 中断，并且 SOF 的内插功能也不运行。 μ SOF 锁定状态是指连续 2 次接收到无错误并且帧号不同的 μ SOF 包。

μ SOF 锁定监视开始条件和 μ SOF 锁定监视停止条件如下所述：

1. μ SOF 锁定监视开始条件

USBE=1

2. μ SOF 锁定监视停止条件

USB \overline{E} =0、接收到 USB 总线复位或者检测到挂起

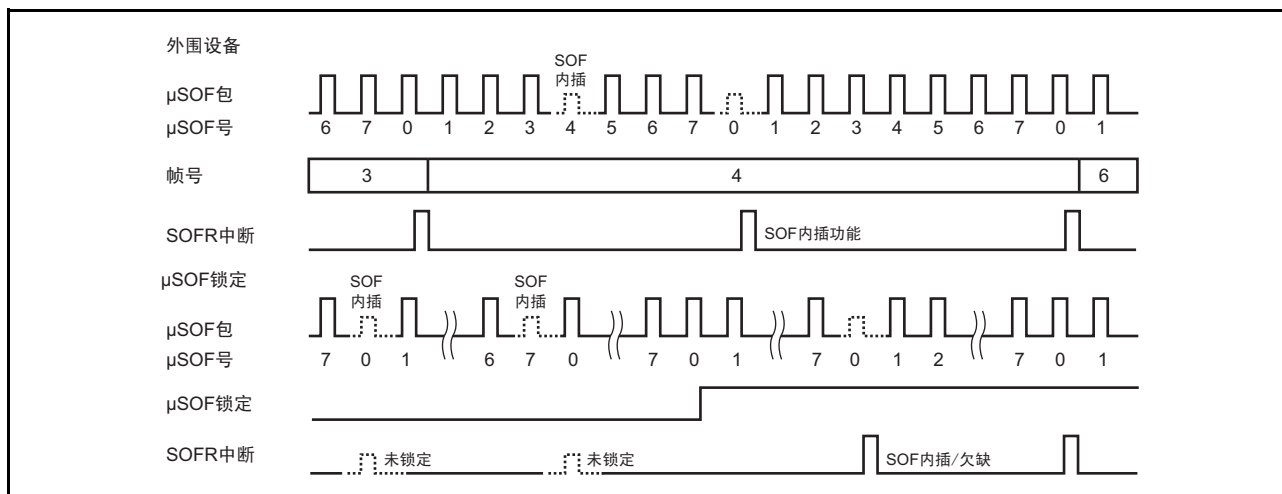


图 21.8 SOFR 中断输出时序例子

(7) VBUS 中断

在 VBUS 引脚发生变化时，产生 VBUS 中断。能通过 INTSTS0 寄存器的 VBSTS 位确认 VBUS 引脚的电平，能通过 VBUS 中断确认主机控制器的连接和断开。但是，在连接主机控制器的状态下启动系统时，VBUS 引脚无变化，所以不产生最初的 VBUS 中断。

(8) 恢复中断

在选择功能控制器功能时，如果在设备状态为挂起状态下 USB 总线状态发生变化 (J-State→K-State 或者 J-State→SE0)，就产生恢复中断。通过恢复中断检测从挂起状态的恢复。

在选择主机控制器功能时，不产生恢复中断。必须使用 BCHG 中断来检测 USB 总线的变化。

(9) BCHG 中断

在 USB 总线状态发生变化时，产生 BCHG 中断。用于在选择主机控制器功能时的外围设备的连接和远程唤醒的检测。在选择主机控制器功能或者功能控制器功能时，都产生 BCHG 中断。

(10) DTCH 中断

在选择主机控制器功能时，如果检测到 USB 总线断开，就产生 DTCH 中断。本模块以 USB Specification 2.0 的基准进行总线断开的检测。

在检测到 DTCH 中断后（与该中断允许位的设定值无关）进行以下的硬件控制。必须通过软件结束正在与该端口进行通信的全部管道的通信，并且转移到该端口的连接（产生 ATTCH 中断）等待状态。

- 将检测到 DTCH 中断的端口的 UACT 位置 0。
- 将产生 DTCH 中断的端口转移到空闲状态。

(11) SACK 中断

在选择主机控制器功能时，如果接收到外围设备对设置包的 ACK 应答，就产生 SACK 中断。能通过 SACK 中断得知设置事务已正常结束。

(12) SIGN 中断

在选择主机控制器功能时，如果不能连续 3 次正常接收到外围设备对设置包的 ACK 应答，就产生 SIGN 中断。能检测外围设备未发送 ACK（无应答）或者 ACK 包破损。

(13) ATTCH 中断

在选择主机控制器功能时，如果在 USB 端口检测到 2.5 μ s 的全速 / 低速信号电平的 J-State 或者 K-State，就产生 ATTCH 中断。ATTCH 中断检测条件如下：

- 从 K-State、SE0 或者 SE1 变为 J-State，并且 J-State 持续保持 2.5 μ s。
- 从 J-State、SE0 或者 SE1 变为 K-State，并且 K-State 持续保持 2.5 μ s。

(14) EOFERR 中断

在 USB 2.0 规格书所规定的 EOF2 时序，如果检测到通信未结束，就产生 EOFERR 中断。

在检测到 EOFERR 中断后（与该中断允许位的设定值无关），进行以下的硬件控制。必须通过软件结束正在与该端口进行通信的全部管道的通信，并且对该端口进行再 Enumeration。

- 将检测到 EOFERR 中断的端口的 UACT 位置 0。
- 将产生 EOFERR 中断的端口转移到空闲状态。

21.4.3 管道控制

本模块的管道设定项目如表 21.18 所示。USB 数据传送需要通过称为端点的逻辑管道进行数据通信。本模块有 10 个用于数据传送的管道，必须按照系统规格设定各管道。

表 21.18 管道设定项目一览表

寄存器名	位名	设定内容	备注
DCPCFG	TYPE	指定传送类型。	管道 1 ~ 9: 能设定
PIPECFG	BFRE	选择 BRDY 中断模式。	管道 1 ~ 5: 能设定
	DBLB	选择双缓冲器。	管道 1 ~ 5: 能设定
	CNTMD	选择连续传送或者非连续传送。	管道 1、2: 能设定 (只在选择批量传送时才能设定) 管道 3 ~ 5: 能设定
	DIR	选择传送方向。	能设定 IN 或者 OUT
	EPNUM	端点号	管道 1 ~ 9: 能设定 在使用管道时, 不能设定为 0000。
	SHTNAK	选择传送结束时的禁止管道。	管道 1、2: 能设定 (只在批量传送选择时才能设定) 管道 3 ~ 5: 能设定
PIPEBUF	BUFSIZE	缓冲存储器大小	DCP: 不能设定 (固定为 256 字节) 管道 1 ~ 5: 能设定 (最多只在设定 2K 字节) 管道 6、9: 不能设定 (固定为 64 字节)
	BUFNMB	缓冲存储器号	DCP: 不能设定 (固定为区域 H'0 ~ H'3) 管道 1 ~ 5: 能设定 (能用区域 H'8 ~ H'7F 指定) 管道 6 ~ 9: 不能设定 (固定为区域 H'4 ~ H'7)
DCPMAXP	DEVSEL	选择设备	只在选择主机控制器功能时参照
PIPEMAXP	MXPS	最大包长度	符合 USB 规格的设置
PIPEPERI	IFIS	清除缓冲器	管道 1、2: 能设定 (只在选择等时传送时) 管道 3 ~ 5: 不能设定 管道 6 ~ 9: 只在选择主机控制器功能时才能设定。
	IITV	间隔计数器	管道 1、2: 能设定 (只在选择等时传送时) 管道 3 ~ 5: 不能设定 管道 6 ~ 9: 只在选择主机控制器功能时才能设定。
DCPCTR	BSTS	缓冲器状态	DCP 通过 ISEL 位进行接收 / 发送缓冲器状态的切换。
PIPEnCTR	INBUFM	监视 IN 缓冲器	管道 3 ~ 5 有此功能。
	SUREQ	请求 SETUP。	只能对 DCP 进行设定。 只在选择主机控制器功能时才能控制。
	SUREQCLR	清除 SUREQ。	只能对 DCP 进行设定。 只在选择主机控制器功能时才能控制。
	CSCLR	清除 CSSTS。	只在选择主机控制器功能时才能控制。
	CSSTS	显示 SPLIT 状态。	只在选择主机控制器功能时才能参照。
	ATREPM	自动应答模式	管道 1 ~ 5: 能设定 只在选择功能控制器功能时才能设定。
	ACLRM	自动清除缓冲器。	管道 1 ~ 9: 能设定
	SQCLR	顺序的清除	清除数据交替位。
	SQSET	顺序的置位	将数据交替位置位。
	SQMON	顺序的确认	确认数据交替位。

寄存器名	位名	设定内容	备注
DCPCTR	PBUSY	管道忙の確認	
PIPEnCTR	PID	应答 PID	请参照“21.4.3(6) 应答 PID”。
PIPEnTRE	TRENB	允许事务计数	管道 1 ~ 5: 能设定
	TRCLR	清除当前事务计数器	管道 1 ~ 5: 能设定
PIPEnTRN	TRNCNT	事务计数器	管道 1 ~ 5: 能设定

(1) 管道控制寄存器的切换步骤

只能在不允许 USB 通信 (PID=NAK) 时改写管道控制寄存器的以下位。

在 USB 通信允许 (PID=BUF) 状态下禁止设定的寄存器的位:

- DCPCFG 寄存器和 DCPMAXP 寄存器的各位
- DCPCTR 寄存器的 SQCLR 位和 SQSET 位
- PIPECFG 寄存器、PIPEBUF 寄存器、PIPEMAXP 寄存器和 PIPEPERI 寄存器的各位
- PIPExCTR 寄存器的 ATREPM 位、ACLRM 位、SQCLR 位和 SQSET 位
- PIPExTRE 寄存器和 PIPExTRN 寄存器的各位

在从 USB 通信允许 (PID=BUF) 状态切换上述位时, 必须按照以下步骤进行:

1. 产生管道控制寄存器的位变更请求。
2. 将该管道的 PID 变为 NAK。
3. 等待该管道的 CSSTS 位变为 0 (只在选择主机控制器功能时)。
4. 等待该管道的 PBUSY 位变为 0。
5. 开始更改管道控制寄存器的位。

只有在还没有给 CFIFOSEL、D0FIFOSEL、D1FIFOSEL 寄存器的 CURPIPE 位设定管道信息时, 才能改写管道控制寄存器的以下位。

在设定 FIFO-PORT 的 CURPIPE 后, 禁止设定的寄存器:

- DCPCFG 寄存器和 DCPMAXP 寄存器的各位
- PIPECFG 寄存器、PIPEBUF 寄存器、PIPEMAXP 寄存器和 PIPEPERI 寄存器的各位

在更改管道信息时, 必须将 CURPIPE 位设定为变更管道。对于 DCP, 必须在修改管道信息后通过 BCLR 清除缓冲器。

(2) 传送类型

通过 PIPEPCFG 寄存器的 TYPE 位设定各管道的传送类型。能给各管道设定的传送类型如下:

- DCP: 不需要设定 (固定为控制传送)。
- 管道 1、2: 必须设定批量传送或者等时传送。
- 管道 3 ~ 5: 必须设定批量传送。
- 管道 6 ~ 9: 必须设定中断传送。

(3) 端点号

通过 PIPEPCFG 寄存器的 EPNUM 位设定各管道的端点号。DCP 固定为端点 0, 其他管道能设定端点 1 ~ 15。

- DCP: 不需要设定 (固定为端点 0)。
- 管道 1 ~ 9: 必须设定 1 ~ 15。但是, DIR 位和 EPNUM 位的组合不能重复。

(4) 最大包长度的设定

通过 DCPMAXP 寄存器和 PIPEMAXP 寄存器的 MXPS 位设定各管道的最大包长度。能给 DCP 和管道 1 ~ 5 设定 USB 规格所定义的所有最大包长度。管道 6 ~ 9 最大为 64 字节, 是最大包长度的上限。必须在开始传送前 (PID=BUF) 设定最大包长度。

- DCP: 在高速运行时, 必须设定 64。
- DCP: 在全速运行时, 必须设定 8、16、32 或者 64。
- 管道 1 ~ 5: 在高速批量传送时, 必须设定 512。
- 管道 1 ~ 5: 在全速批量传送时, 必须设定 8、16、32 或者 64。
- 管道 1、2: 在高速等时传送时, 必须设定 1 ~ 1024 的值。
- 管道 1、2: 在全速等时传送时, 必须设定 1 ~ 1023 的值。
- 管道 6 ~ 9: 必须设定 1 ~ 64 的值。

不支持中断传送和等时传送的 High Bandwidth。

(5) 事务计数器 (管道 1 ~ 5 为读方向)

在以数据包接收方向结束指定次数的事务时, 本模块能认为是传送结束。在 D0FIFO/D1FIFO 端口所选的管道为从缓冲存储器读数据的方向时, 事务计数器工作。事务计数器有指定事务次数的 TRNCNT 寄存器和在内部进行事务计数的当前计数器, 在当前计数器和指定次数相同时, 缓冲存储器为可读状态。能通过 TRCLR 位对事务计数器功能的当前计数器进行初始化, 对事务从头开始重新计数。TRNCNT 寄存器的读取值的信息因 TRENb 位的设定而不同。

- TRENb=0: 能读取被设定的事务计数器的值。
- TRENb=1: 能读取在内部进行计数的当前计数器的值。

TRCLR 位的操作条件如下:

- 在事务计数过程中 PID=BUF 时, 不能清除当前计数器。
- 在缓冲器中有数据剩余时, 不能清除当前计数器。

(6) 应答 PID

通过 DCPCTR 寄存器和 PIPEnCTR 寄存器的 PID 位设定各管道的应答 PID。

在各设定中, 本模块的运行如下:

1. 选择主机控制器功能时的应答 PID 设定

应答 PID 指定事务的执行。

- 设定 NAK: 为管道禁止状态。不执行事务。
- 设定 BUF: 根据缓冲存储器的状况执行事务。
在 OUT 方向时, 如果缓冲存储器中有发送数据, 就发行 OUT 标记。
在 IN 方向时, 如果缓冲存储器中有空间并且为接收允许状态, 就发行 IN 标记。
- 设定 STALL: 为管道禁止状态。不执行事务。

【注】 通过 SUREQ 位设定 DCP 的设置事务。

2. 选择功能控制器功能时的应答 PID 设定

应答 PID 指定对事务 (主机) 的应答。

- 设定 NAK: 对发生的事务总是进行 NAK 应答。
- 设定 BUF: 根据缓冲存储器的状况应答事务。
- 设定 STALL: 对发生的事务总是进行 STALL 应答。

【注】 与 PID 的设定无关, 总是对设置事务进行 ACK 应答, 并且将 USB 请求保存到寄存器。

根据事务结果，可能会发生 PID 位的写操作。在发生以下情况时，本模块写 PID 位。

3. 在选择主机控制器功能时硬件设定应答 PID

- 设定 NAK: 在以下情况下，PID=NAK，自动停止标记的发行。
在等时以外的传送时，产生 NRDY 中断（详细内容请参照 NRDY 中断）。
在批量传送时，在将 PIPECFG 寄存器的 SHTNAK 位置 1 的情况下接收到短包。
在批量传送时，将 SHTNAK 位置 1，结束事务计数器的计数。
- 设定 BUF: 本模块不进行 BUF 的写操作。
- 设定 STALL: 在以下情况下，PID=STALL，自动停止标记的发行。
对发送的标记，接收到应答的 STALL。
接收的数据包超过最大包长度。

4. 在选择功能控制器功能时硬件设定应答 PID

- 设定 NAK: 在以下情况下，PID=NAK，总是对事务进行 NAK 应答。
正常接收到 SETUP 标记（只对 DCP）。
在批量传送时，将 PIPECFG 寄存器的 SHTNAK 位置 1，结束事务计数器的计数或者接收到短包。
- 设定 BUF: 本模块不进行 BUF 的写操作。
- 设定 STALL: 在以下情况下，PID=STALL，总是对事务进行 STALL 应答。
接收的数据包超过最大包长度。
检测到控制传送顺序错误（只对 DCP）。

(7) 数据 PID 顺序位

在控制传送的数据阶段、批量传送和中断传送中，如果进行正常的传送，数据 PID 的顺序位就自动取反。能通过 DCPCTR 寄存器和 PIPEnCTR 寄存器的 SQMON 位确认下一次发送的数据 PID 顺序位。发送数据时，在接收 ACK 握手信号时转换数据位；接收数据时，在发送 ACK 握手信号时转换顺序位。能通过 DCPCTR 寄存器和 PIPEnCTR 寄存器的 SQCLR 位、SQSET 位更改数据 PID 的顺序位。

对于选择功能控制器功能时的控制传送，本模块在阶段转移时自动设定顺序位。在设置阶段结束时由 DATA0 应答；在状态阶段时由 DATA1 应答。因此，不需要通过软件进行设定。对于选择主机控制器功能时的控制传送，在阶段转换时，需要通过软件设定顺序位。

必须注意：无论是选择主机控制器功能还是选择功能控制器功能，在发送或者接收 ClearFeature 请求时，都需要通过软件设定数据 PID 顺序位。

对于设定为等时传送的管道，不能通过 SQSET 位进行顺序位的操作。

(8) 应答 PID=NAK 功能

本模块能通过将 PIPECFG 寄存器的 SHTNAK 位置 1，在接收传送的最后（根据接收的短包或者事务计数器自动认识）数据包时禁止（应答 PID=NAK）管道运行。

通过使用此功能，在缓冲存储器为双缓冲器时，能以传送单位接收数据包。在禁止管道运行后，需要通过软件重新进行管道允许（应答 PID=BUF）的设定。

此功能只能用于批量传送。

(9) 自动应答模式

对于批量传送的管道（管道 1 ~ 5），如果将 PIPEnCTR 寄存器的 ATREPM 位置 1，就变为自动应答模式。在 OUT 传送时（DIR=0）变为 OUT-NAK 模式；在 IN 传送时（DIR=1）变为 Null 自动应答模式。

- **OUT-NAK 模式**

对于批量OUT传送的管道，如果将ATREPM位置1，就对OUT标记或者PING标记进行NAK应答，并且输出NRDY中断。为了从正常模式设定为OUT-NAK模式，必须在管道运行禁止状态（应答PID=NAK）下设定OUT-NAK模式，然后允许管道运行（应答PID=BUF）。在允许管道运行后，OUT-NAK模式有效。但是，如果在设定为禁止管道运行前接受到OUT标记，就正常接收此标记的数据，并且对主机进行ACK应答。

为了从OUT-NAK模式转移到正常模式，必须在管道运行禁止状态（应答PID=NAK）下解除OUT-NAK模式，然后允许管道运行（应答PID=BUF）。在正常模式中，能接收OUT数据，如果缓冲器为可接收数据状态，就对PING标记进行ACK应答。

- **Null自动应答模式**

对于批量IN传送的管道，如果将ATREPM位置1，就继续发送Zero-Length包。

为了从正常模式设定为Null自动应答模式，必须在管道运行禁止状态（应答PID=NAK）下设定Null自动应答模式，然后允许管道运行（应答PID=BUF）。在允许管道运行之后，Null自动应答模式有效。但是，在设定Null自动应答模式时，需要缓冲器为空状态，所以必须确认INBUFM位是否为0。因为当INBUFM位为1时，缓冲器中有数据，所以必须通过ACLRM位将缓冲器清空。在Null自动应答模式的设定过程中，不能给FIFO端口写数据。

为了从Null自动应答模式转移到正常模式，必须使管道运行禁止状态（应答PID=NAK）保持1个Zero-Length包的发送时间（全速时：10 μ s，高速时：3 μ s），然后解除Null自动应答模式。在正常模式中，能给FIFO端口写数据，在允许管道运行（应答PID=BUF）后，能给主机发送信息包。

21.4.4 FIFO 缓冲存储器

(1) FIFO 缓冲存储器的分配

本模块的 FIFO 缓冲存储器映像例子如图 21.9 所示。FIFO 缓冲存储器为 CPU 和本模块共用的区域。对于 FIFO 缓冲存储器的存取权，有系统（CPU）拥有存取权和本模块（SIE）拥有存取权的 2 种情况。

FIFO 缓冲存储器给每个管道设定独立的区域。64 字节为 1 块，通过块起始号和块数（PIPEBUF 寄存器的 BUFNMB 和 BUFSIZE 位）进行设定。

在通过 PIPEnCFG 寄存器的 CNTMD 位选择连续传送模式时，必须将 BUFSIZE 位设定为最大包长度的整数倍。在通过 PIPEnCFG 寄存器的 DBLB 位选择双缓冲器时，给同一管道分配 2 个由 PIPEBUF 寄存器的 BUFSIZE 位指定的存储区面。

缓冲存储器的存取（读写数据）使用 3 个 FIFO 端口。对于分配给 FIFO 端口的管道，通过 C/DnFIFOSEL 寄存器的 CURPIPE 位指定管道号。

能通过 DCPCTR 寄存器和 PIPEnCTR 寄存器的 BSTS 位、INBUFM 位确认各管道的缓冲器状态，能通过 CFIFOCTR 寄存器和 DnFIFOCTR 寄存器的 FRDY 位确认 FIFO 端口的存取权。

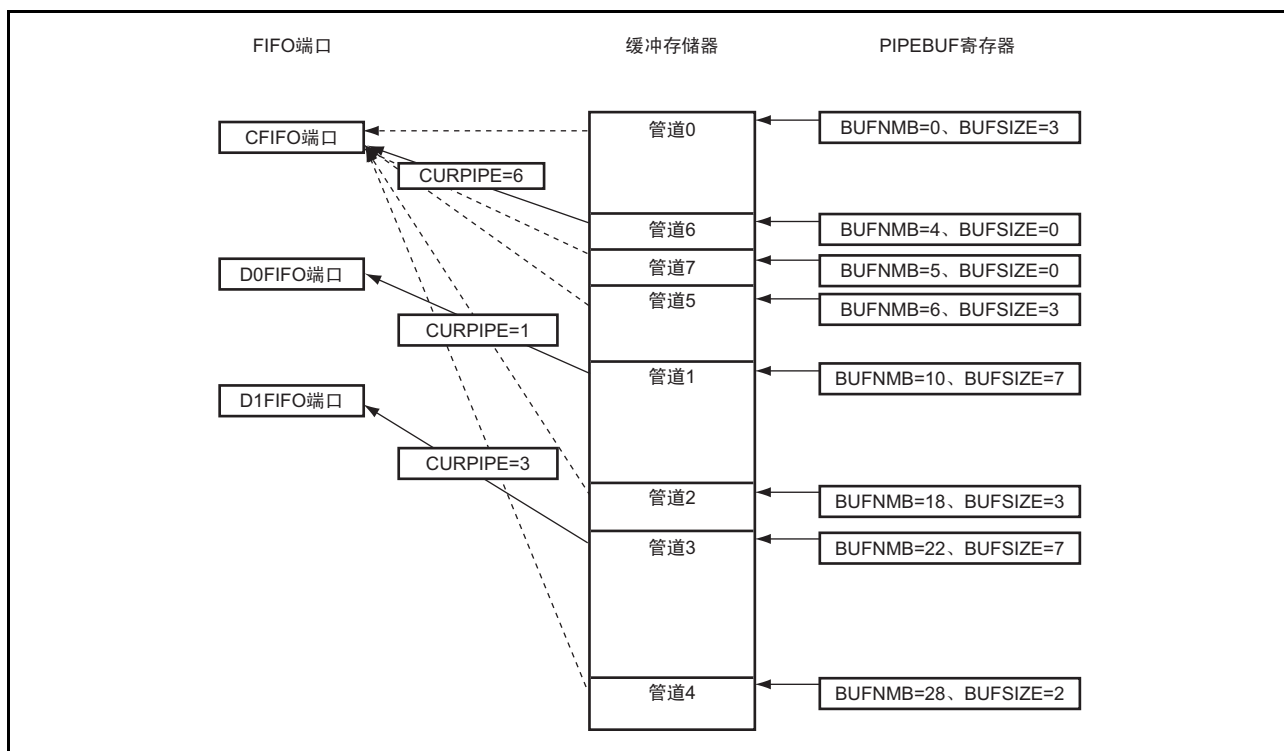


图 21.9 缓冲存储器映像例子

- 缓冲器状态

本模块的缓冲器状态表如表 21.19 和表 21.20 所示。能通过 DCPCTR.BSTS 位和 PIPEnCTR.INBUFM 位确认缓冲存储器状态，能通过 PIPEnCFG 寄存器的 DIR 位或者 CFIFOSEL 寄存器的 ISEL 位（DCP 选择时）指定缓冲存储器的存取方向。

INBUFM 位只在管道 1~5 为发送方向时有效。

在将发送侧的传送管道设定为双缓冲器时，BSTS 位用于判断 CPU 的缓冲器状态，INBUFM 位用于判断 SIE 的缓冲器状态。在因 CPU（DMAC）给 FIFO 端口写数据的速度慢而不能通过 BEMP 中断判断缓冲器为空状态时，能通过 INBUFM 位确认发送结束。

表 21.19 通过 BSTS 位表示的缓冲器状态表

ISEL 或者 DIR	BSTS	缓冲存储器的状态
0 (接收方向)	0	没有接收数据或者正在接收。 不能从 FIFO 端口读数据。
0 (接收方向)	1	有接收数据或者接收到 Zero-Length 包。 能从 FIFO 端口读数据。 但是, 在接收 Zero-Length 包时不能读取, 所以需要清除缓冲器。
1 (发送方向)	0	发送没有结束。 不能给 FIFO 端口写数据。
1 (发送方向)	1	发送结束。 允许 CPU 写。

表 21.20 通过 INBUFM 位表示的缓冲器状态表

DIR	BSTS	缓冲存储器的状态
0 (接收方向)	无效	无效
1 (发送方向)	0	发送完能发送的数据。 没有能发送的数据。
1 (发送方向)	1	已给 FIFO 端口写能发送的数据。 有能发送的数据。

- FIFO 缓冲器的清除

本模块的 FIFO 缓冲存储器清除如表 21.21 所示。能通过 BCLR、DCLRM 和 ACLRM 位清除缓冲存储器。

表 21.21 缓冲器清除一览表

位名	BCLR	DCLRM	ACLRM
寄存器	CFIFOCTR 寄存器 DnFIFOCTR 寄存器	DnFIFOSEL 寄存器	PIPEnCTR 寄存器
功能	清除 CPU 缓冲存储器。	是在读取指定管道的数据后自动清除缓冲存储器的模式。	是放弃全部接收包的自动清除缓冲器模式。
清除方法	通过写 1 进行清除	1: 模式有效 0: 模式无效	1: 模式有效 0: 模式无效

- 缓冲区

本模块的缓冲存储器映像如表 21.22 所示。缓冲存储器有事先分配给管道的专用固定区和用户能设定的用户区。

DCP 的缓冲器是在控制读传送和控制写传送时使用同一个区域的专用固定区。

事先分配管道 6~9 的区域, 在不使用管道 6~9 时, 能作为用户区分配给管道 1~5 使用。

各管道的区域不能重叠。尤其要注意在设定双缓冲器时区域为设定值的 2 倍。

不能用小于最大包长度的设定值指定缓冲器大小。

表 21.22 缓冲存储器映像

缓冲存储器号	缓冲器大小	管道的设定	备注
H'0	64 字节	DCP 专用固定区	单缓冲器
H'1 ~ H'3	—	禁止使用	—
H'4	64 字节	管道 6 用固定区	单缓冲器
H'5	64 字节	管道 7 用固定区	单缓冲器
H'6	64 字节	管道 8 用固定区	单缓冲器
H'7	64 字节	管道 9 用固定区	单缓冲器
H'8 ~ H'4F	最大 5120 字节	管道 1 ~ 5 用户区	能设定双缓冲器, 进行连续传送

- 缓冲器自动清除模式

通过将 PIPEnCTR 寄存器的 ACLRM 位置 1, 放弃所接收的全部数据包。但是, 在接收到正常的数据包时, 对主机控制器进行 ACK 应答。只能在缓冲存储器为读方向时设定此功能。

如果将 ACLRM 位置 1 后接着置 0, 就能与存取方向无关, 清除所选管道的缓冲存储器。

作为硬件的内部顺序执行时间, ACLRM 位的写 1 和写 0 的间隔至少需要 100ns。

- 缓冲存储器规格 (设定单缓冲器/双缓冲器)

管道 1 ~ 5 能通过 PIPEnCFG 寄存器的 DBLB 位选择单缓冲器或者双缓冲器。双缓冲器是给同一管道分配 2 个由 PIPEBUF 寄存器的 BUFSIZE 位指定的存储区面的功能。本模块的缓冲存储器设定例子如图 21.10 所示。

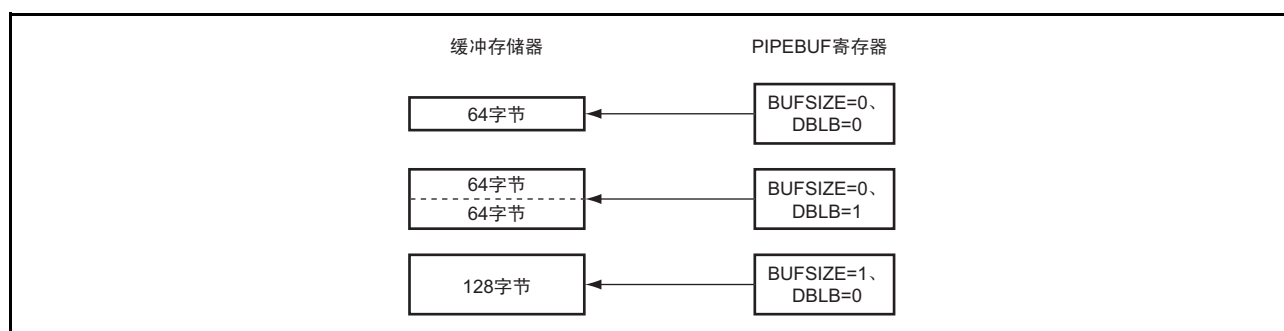


图 21.10 缓冲存储器设定例子

- 缓冲存储器的运行 (设定连续传送)

能通过 PIPEnCFG 寄存器的 CNTMD 位选择连续传送模式或者非连续传送模式。此选择对管道 1 ~ 5 有效。

连续传送模式功能是连续发送/接收多个事务的功能。在设定连续传送模式时, 不向 CPU 产生中断而能进行分配给各管道的缓冲器大小的数据传送。

在连续发送模式中, 根据最大包长度将写数据分割发送。对于不满缓冲器大小的数据 (短包或者最大包尺寸的整数倍小于缓冲器大小), 需要在写发送数据后设定 BVAL=1。

在连续接收模式中, 在接收缓冲器大小的信息包、结束事务计数或者接收短包前不产生中断。

CNTMD 位的设定值和 FIFO 缓冲器的发送/接收结束判断方法如表 21.23 所示。

表 21.23 CNTMD 位的设定值和 FIFO 缓冲器的发送 / 接收结束判断方法

连续传送模式	可读状态和可发送状态的判断方法
非连续传送 (CNTMD=0)	<p>在设定为接收方向时 (DIR=0), FIFO 缓冲器为可读状态的条件: 接收到 1 个包。</p> <p>在设定为发送方向时 (DIR=1), FIFO 缓冲器为可发送状态的条件: 满足以下任意一个条件时</p> <ul style="list-style-type: none"> 通过软件 (或者 DMAC) 将最大包长度的数据写到 FIFO 缓冲器。 通过软件 (或者 DMAC) 将短包的数据 (包括 0 字节时) 写到 FIFO 缓冲器, 并且设定了 BVAL=1。
连续传送 (CNTMD=1)	<p>在设定为接收方向时 (DIR=0), FIFO 缓冲器为可读状态的条件:</p> <ul style="list-style-type: none"> 分配给所选管道的 FIFO 缓冲器接收到的数据的字节数和指定的字节数 ((BUFSIZE+1)×64) 相等。 接收到 Zero-Length 包以外的短包。 在分配给所选管道的 FIFO 缓冲器已保存数据的状态下, 接收到 Zero-Length 包。 接收到由软件给所选管道设定的事务计数器次数的包。 <p>在设定为发送方向时 (DIR=1), FIFO 缓冲器为可发送状态的条件: 满足以下任意一个条件时</p> <ul style="list-style-type: none"> 由软件 (或者 DMAC) 写的的数据个数和分配给所选管道的 1 个 FIFO 缓冲器面大小相等。 通过软件 (或者 DMAC) 将小于分配给所选管道的 1 个 FIFO 缓冲器面大小的数据个数 (包括 0 字节时) 写到 FIFO 缓冲器, 并且设定了 BVAL=1。

本模块的缓冲存储器运行例子如图 21.11 所示。

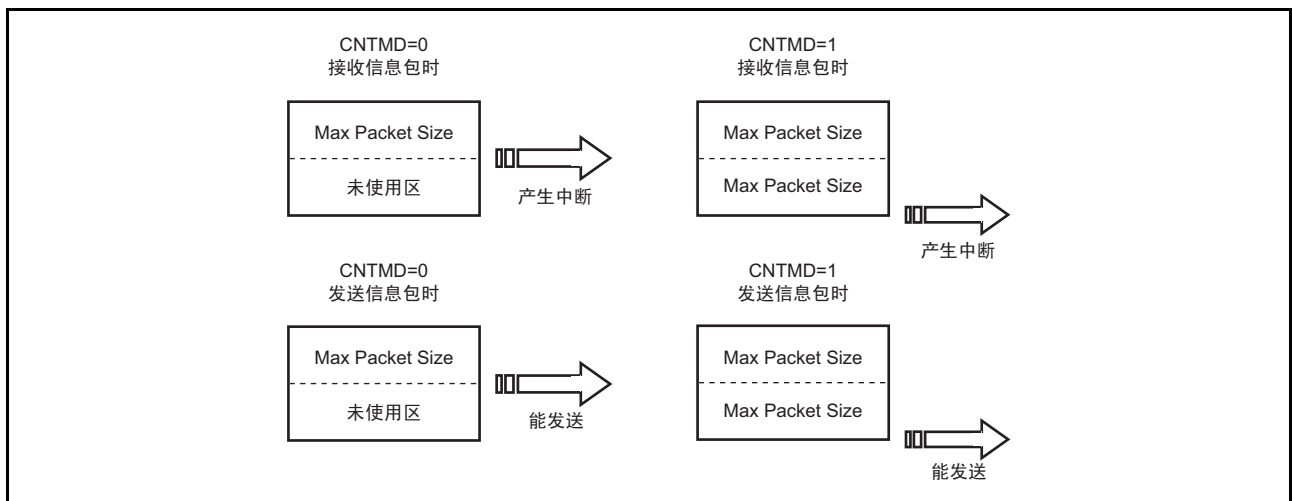


图 21.11 缓冲存储器运行例子

(2) FIFO 端口功能

本模块的 FIFO 端口功能设定如表 21.24 所示。在写数据时，如果写满缓冲器（非连续传送时的最大包长度），就自动进入可发送状态。为了将缓冲器存满（非连续传送时为最大包长度）前的数据变为可发送状态，需要通过 C/DnFIFOCTR 寄存器的 BVAL 位设定写结束。

在发送 Zero-Length 包时，也需要在通过 C/DnFIFOCTR 寄存器的 BCLR 位清除缓冲器后，用 BVAL 位进行写结束的设定。

在读存取时，如果读取全部数据，就自动进入可接收新包的状态。但是，在接收 Zero-Length 包时（DTLN=0），因为无法读取数据，所以需要通过 C/DnFIFOCTR 寄存器的 BCLR 位清除缓冲器。通过 C/DnFIFOCTR 寄存器的 DTLN 位确认接收数据的长度。

表 21.24 FIFO 端口功能设定表

寄存器名	位名	功能	备注
C/DnFIFOSEL	RCNT	选择 DTLN 读模式。	
	REW	对缓冲存储器指针进行复位（再读和再写）	
	DCLRM	在读取指定管道的接收数据后自动清除。	DnFIFO 专用
	DREQE	允许 DMA 传送。	DnFIFO 专用
	MBW	FIFO 端口的存取宽度。	
	BIGEND	选择 FIFO 端口的字节排序。	
	ISEL	FIFO 端口存取方向	DCP 专用
	CURPIPE	选择当前管道。	
C/DnFIFOCTR	BVAL	结束缓冲存储器的写操作。	
	BCLR	清除 CPU 缓冲存储器。	
	DTLN	确认接收数据的长度。	

(a) FIFO 端口的选择

各 FIFO 端口能选择的管道如表 21.25 所示。通过 C/DnFIFOSEL 寄存器的 CURPIPE 位选择要存取的管道。在选择管道后，必须确认是否能正确读取所写的 CURPIPE 值（在读取的内容是上一次管道号时，该控制器表示管道处于变更处理过程中）以及确认 FRDY=1，然后存取 FIFO 端口。

通过 MBW 位选择存取的总线宽度。缓冲存储器存取方向取决于 PIPEnCFG 寄存器的 DIR 位，但是 DCP 取决于 ISEL 位。

表 21.25 不同管道的 FIFO 端口存取表

管道	存取方法	能使用的端口
DCP	CPU 存取	CFIFO 端口寄存器
管道 1 ~ 9	CPU 存取	CFIFO 端口寄存器 D0FIFO/D1FIFO 端口寄存器
	DMA 存取	D0FIFO/D1FIFO 端口寄存器

(b) REW 位

通过 C/DnFIFOSEL 寄存器的 REW 位，能暂停当前存取中的管道存取，对其他管道进行存取，然后再次继续进行当前管道的处理。

如果在设定 C/DnFIFOSEL 寄存器的 CURPIPE 位的同时将 REW 位置 1，并且进行管道选择，缓冲存储器的读写指针就被复位，能从最初的字节开始读写。如果将 REW 位置 0 后选择管道，就不对缓冲存储器的读写指针进行复位，能接着上一次的读写继续进行数据的读写。

为了存取 FIFO 端口，需要在选择管道之后确认 FRDY=1。

(3) DMA 传送 (D0FIFO/D1FIFO 端口)

(a) DMA 传送的概要

对于管道 1 ~ 9，能通过 DMAC 存取 FIFO 端口。在 DMA 设定的管道缓冲器为可存取状态时，输出 DMA 传送请求。

必须通过 DnFIFOSEL 寄存器的 MBW 位选择 FIFO 端口的传送单位，通过 CURPIPE 位选择 DMA 传送的管道。不能在 DMA 传送中更改所选的管道。

(b) DMA 传送结束的自动认识

本模块能通过 DMA 传送结束信号的输入控制，结束 DMA 传送的 FIFO 数据写操作。当采样到传送结束信号时，就将缓冲存储器设定为可发送状态（与设定 BVAL=1 时的状态相同）。

(c) DnFIFO 自动清除模式 (D0FIFO/D1FIFO 端口为读方向)

通过将 DnFIFOSEL 寄存器的 DCLRM 位置 1，在读完缓冲存储器的数据时自动清除所选管道的缓冲存储器。

各设定状态下的接收包和由软件清除缓冲存储器的关系如表 21.26 所示。缓冲器的清除条件因 BFRE 位的设定值而不同，对于需要清除的情况，只要使用 DCLRM 位就能不通过软件进行缓冲器的清除，而且不通过软件能进行 DMA 传送。

只能在缓冲存储器为读方向时设定此功能。

表 21.26 接收包和由软件清除缓冲存储器的关系

寄存器的设定 接收包时的缓冲器状态	DCLRM=0		DCLRM=1	
	BFRE=0	BFRE=1	BFRE=0	BFRE=1
缓冲器满	不需要清除	不需要清除	不需要清除	不需要清除
接收到 Zero-Length 包	需要清除	需要清除	不需要清除	不需要清除
接收到正常的短包	不需要清除	需要清除	不需要清除	不需要清除
事务计数结束	不需要清除	需要清除	不需要清除	不需要清除

21.4.5 控制传送 (DCP)

控制传送的数据阶段的数据传送使用默认控制管道 (DCP)。DCP 的缓冲存储器是控制读和控制写共用的固定区, 为 256 字节的单缓冲器。只能由 CFIFO 端口存取缓冲存储器。

(1) 选择主机控制器功能时的控制传送

(a) 设置阶段

USBREQ 寄存器、USBVAL 寄存器、USBINDX 寄存器和 USBLENG 寄存器是用于发送设置事务的 USB 请求。如果将设置包的数据写给寄存器, 并且将 DCPCTR 寄存器的 SUREQ 位置 1, 就发送设定为设置事务的数据。在事务结束时, SUREQ 位被清 0。在 SUREQ=1 期间, 不能操作上述的 USB 请求寄存器。通过 DCPMAXP 寄存器的 DEVSEL 位指定设置事务的设备地址。

如果发送事务, 就根据外围设备的应答产生中断请求 (INTSTS1 寄存器的 SIGN 位和 SACK 位)。能通过此中断请求确认设置事务的结果。

设置事务的数据包与 DCPCTR 寄存器的 SQMON 位的内容无关, 总是发送 DATA0 数据包 (USB 请求)。

(b) 数据阶段

使用 DCP 缓冲存储器进行数据传送。

必须通过 CFIFOSEL 寄存器的 ISEL 位指定 DCP 缓冲存储器的存取方向, 通过 DCPCFG 寄存器的 DIR 位指定传送方向。

数据阶段的第 1 个数据包需要将数据 PID 作为 DATA1 进行通信。如果通过 DCPCFG 寄存器的 SQSET 位将数据 PID 和 PID 位分别设定到 DATA1 和 BUF, 就执行事务。通过 BRDY 中断和 BEMP 中断检测数据传送的结束。

能通过连续传送的指定进行多个包的数据传送。但是, 必须注意: 在以接收方向进行连续传送时, 只有在缓冲器存满或者接收到短包时才会产生 BRDY 中断 (最大包长度的整数倍并且不超过 256 字节的情况)。

在控制写传送的情况下, 当发送数据为最大包长度的整数倍时, 必须在最后通过软件发送 Zero-Length 包。

(c) 状态阶段

这是与数据阶段反方向的 Zero-Length 包的数据传送。和数据阶段相同, 是使用 DCP 缓冲存储器的数据传送。以数据阶段相同的步骤执行事务。

数据阶段的数据包需要将数据 PID 作为 DATA1 进行通信。必须通过 DCPCFG 寄存器的 SQSET 位将数据 PID 设定到 DATA1。

对于 Zero-Length 包的接收, 必须在发生 BRDY 中断后, 通过 CFIFOCTR 寄存器的 DTLN 位确认接收数据的长度, 然后通过 BCLR 位清除缓冲存储器。

(2) 选择功能控制器功能时的控制传送

(a) 设置阶段

本模块对正常的设置包进行 ACK 应答。设置阶段的本模块运行如下:

1. 在接收到新的设置包时, 将以下位置位:
 - 将 INTSTS0 寄存器的 VALID 位置 1。
 - 将 DCPCTR 寄存器的 PID 位设定为 NAK。
 - 将 DCPCTR 寄存器的 CCPL 位置 0。
2. 在接收到设置包后的数据包时, 将 USB 请求的参数保存到 USBREQ 寄存器、USBVAL 寄存器、USBINDX 寄存器和 USBLENG 寄存器。

必须在设定 VALID=0 后对控制传送进行应答处理。在 VALID=1 的状态下, 不进行 PID=BUF 的设定, 并且也不能结束数据阶段。

通过 VALID 位的功能, 能在控制传送的过程中接收到新的 USB 请求时中止正在处理中的请求处理, 并且对最新请求进行应答。

本模块自动判断接收的 USB 请求的方向位 (bmRequestType 的 bit8) 和请求数据的长度 (wLength), 并且判断控制读传送、控制写传送和控制写无数据传送, 进行阶段转移的管理。对于错误的顺序, 产生控制传送阶段转移中断的顺序错误, 通知软件。有关本模块的阶段管理请参照图 21.7。

(b) 数据阶段

通过 DCP 进行接收的 USB 请求所对应的数据传送。在存取 DCP 缓冲存储器前, 必须通过 CFIFOSEL 寄存器的 ISEL 位指定存取方向。

如果传送数据大于 DCP 缓冲存储器的大小, 就在控制写传送时使用 BRDY 中断进行数据传送, 在控制读传送时使用 BEMP 中断进行数据传送。

高速运行时的控制写传送, 根据缓冲存储器的状况进行 NYET 握手信号应答。

(c) 状态阶段

在 DCPCTR 寄存器的 PID 位为 PID=BUF 时, 通过将 CCPL 位置 1 来结束控制传送。

在进行上述设定后, 根据设置阶段确定的数据传送方向, 本模块自动执行状态阶段。具体运行如下:

- 控制读传送时
发送 Zero-Length 包, 并且接收 USB 主机的 ACK 应答。
- 控制写传送、无数据控制传送时
接收 USB 主机的 Zero-Length 包, 并且发送 ACK 应答。

(d) 控制传送自动应答功能

本模块自动应答正常的 SET_ADDRESS 请求。在 SET_ADDRESS 请求发生以下错误时, 需要通过软件进行应答:

- 控制读传送以外的传送时: bmRequestType≠H'00
- 请求错误时: wIndex≠H'00
- 无数据控制传送以外的传送时: wLength≠H'00
- 请求错误时: wValue > H'7F
- 设备状态错误的控制传送: DVSQ=011 (Configured)

对于 SET_ADDRESS 以外的全部请求, 需要通过对应的软件进行应答。

21.4.6 批量传送 (管道 1 ~ 5)

批量传送能选择缓冲存储器的使用方法 (设定单 / 双缓冲器或者设定连续 / 非连续传送模式)。缓冲存储器大小最大能设定 2K 字节。本模块管理缓冲存储器的状态, 对于 PING 包 / NYET 握手信号进行自动应答。

(1) 选择主机控制器功能时的 PING 包控制

本模块自动发送 OUT 方向的 PING 包。

初始期状态如下所示, 在 PING 包的发送状态下, 接收 ACK 握手信号, 然后发送 OUT 包。当接收到 NAK 或者 NYET 时, 就返回到 PING 包的发送状态。此控制适用于控制传送的数据阶段和状态阶段。

1. 设定 OUT 数据发送模式
2. 发送 PING 包
3. 接收 ACK 握手信号

4. 发送 OUT 数据包
5. 接收 ACK 握手信号 (重复 4 和 5)
6. 发送 OUT 数据包
7. 接收 NAK/NYET 握手信号
8. 发送 PING 包

通过上电复位、NYET/NAK 握手信号的接收、顺序交替位的设定和清除 (SQSET、SQCLR) 以及缓冲器清除 (ACLRM) 的设定, 本模块返回到 PING 包的发送状态。

(2) 选择功能控制器功能时的 NYET 握手信号控制

本模块的 NYET 握手信号应答如表 21.27 所示。按照以下条件进行 NYET 应答, 但是在接收短包时, 进行 ACK 应答而不是 NYET 包应答。此控制还适用于控制写传送的数据阶段。

表 21.27 NYET 握手信号应答表

DCPCTR.PID 位的设定值	缓冲存储器的状态	标记	应答	备注
NAK/STALL	—	SETUP	ACK	—
	—	IN/OUT/PING	NAK/STALL	—
BUF	—	SETUP	ACK	—
	RCV-BRDY1	OUT/PING	ACK	在接收 OUT 标记时, 接收数据包。
	RCV-BRDY2	OUT	NYET	接收数据包, 通知不能接受。
	RCV-BRDY2	OUT (Short)	ACK	接收数据包, 通知能接受。
	RCV-BRDY2	PING	ACK	通知能接收。
	RCV-NRDY	OUT/PING	NAK	通知不能接收。
	TRN-BRDY	IN	DATA0/1	发送数据包。
	TRN-NRDY	IN	NAK	TRN-NRDY

【符号说明】

RCV-BRDY1: 在接收 OUT/PING 标记时, 缓冲存储器有 2 个包以上的空间。

RCV-BRDY2: 在接收 OUT 标记时, 缓冲存储器只有 1 个包的空间。

RCV-NRDY: 在接收 PING 标记时, 缓冲存储器没有空间。

TRN-BRDY: 在接收 IN 标记时, 缓冲存储器中有发送数据。

TRN-NRDY: 在接收 IN 标记时, 缓冲存储器中没有发送数据。

21.4.7 中断传送 (管道 6 ~ 9)

在选择功能控制器功能时, 本模块根据主机控制器管理的周期进行中断传送。在中断传送时, 忽视 (为无应答) PING 包, 并且不发送 NYET 握手信号, 而进行 ACK、NAK、STALL 应答。

在选择主机控制器功能时, 能通过间隔计数器设定标记的发行时序。即使是 OUT 方向的传送, 也不发行 PING 标记, 而发行 OUT 标记。

本模块不支持中断传送的 High-Bandwidth 传送。

(1) 选择主机控制器功能时的中断传送的间隔计数器

在进行中断传送时, 给 PIPEPERI 寄存器的 IITV 位设定事务间隔。该控制器根据设定的间隔发行中断传送的标记。

(a) 计数器的初始化

该控制器对间隔计数器进行初始化的条件如下：

- 上电复位
对 IITV 位进行初始化。
- 通过 ACLRM 进行的缓冲存储器初始化
不对 IITV 位进行初始化，而对计数值进行初始化。通过将 ACLRM 位置 0，从 IITV 的设定值进行计数。

必须注意：在以下情况下，间隔计数器不被初始化：

- USB 总线复位、USB 挂起
不对 IITV 位进行初始化。通过将 UACT 位置 1，从进入 USB 总线复位或者 USB 挂起状态前的值开始计数。

(b) 在产生标记时不能进行发送 / 接收的运行

在以下情况下，即使在产生标记时也不产生标记，而在下一个间隔中尝试执行事务。

- 将 PID 设定为 NAK 或者 STALL
- IN 方向（接收）的传送：在发送标记时缓冲存储器没有空间
- OUT 方向（发送）的传送：在发送标记时缓冲存储器没有发送数据

21.4.8 等时传送（管道 1、2）

本模块的等时传送具有下列功能：

- 等时传送的错误信息通知
- 间隔计数器（由 IITV 位指定）
- 等时 IN 传送数据设置控制（IDLY 功能）
- 等时 IN 传送缓冲器清除功能（由 IFIS 位指定）

本模块不支持等时传送的 High-Bandwidth 传送。

(1) 等时传送的错误检测

本模块具有检测下列错误信息的功能，能通过软件管理等时传送的错误产生。有关本模块检测错误的优先顺序和错误检测的产生中断如表 21.28 和表 21.29 所示。

1. PID 错误
 - 接收包的 PID 不正确
2. CRC 错误、位填充错误
 - 接收包的 CRC 有错误或者位填充不正确
3. 超出最大包长度
 - 接收包的数据长度超出了最大包长度的设定值
4. 超限、欠载错误
 - 选择主机控制器功能时
IN 方向（接收）的传送：在发送标记时缓冲存储器没有空间
OUT 方向（发送）的传送：在发送标记时缓冲存储器没有数据
 - 选择功能控制器功能时
IN 方向（发送）的传送：在接收 IN 标记时缓冲存储器没有数据
OUT 方向（接收）的传送：接收到 OUT 标记，但是缓冲存储器没有空间
5. 间隔错误
在选择功能控制器功能时，如果发生以下情况，就为间隔错误：
 - 等时 IN 传送：在间隔帧中不能接收 IN 标记
 - 等时 OUT 传送：在非间隔帧中接收到 OUT 标记

表 21.28 接收标记时的错误检测

检测的优先顺序	错误	产生的中断和状态
1	PID 错误	主机和功能都不产生中断 (作为破损包被忽视)。
2	CRC 错误、位填充错误	主机和功能都不产生中断 (作为破损包被忽视)。
3	超限、欠载错误	主机和功能都产生 NRDY 中断, 并且将 OVRN 位置位。在选择主机控制器功能时, 不发送标记; 在选择功能控制器功能时, 对 IN 标记发送 Zero-Length 包。对 OUT 标记不接收数据包。
4	间隔错误	在选择功能控制器功能时, 产生 NRDY 中断; 在选择主机控制器功能时, 不产生 NRDY 中断。

表 21.29 接收数据包时的错误检测

检测的优先顺序	错误	产生的中断和状态
1	PID 错误	不产生中断 (作为破损包被忽视)。
2	CRC 错误、位填充错误	主机和功能都产生 NRDY 中断, 并且将 CRCE 位置位。
3	超出最大包长度错误	主机和功能都产生 BEMP 中断, 并且将 PID 设定为 STALL。

(2) DATA-PID

本模块不支持 High-Bandwidth 传送。在选择功能控制器功能时, 对接收的 PID 进行以下的处理:

1. IN 方向

- DATA0: 作为数据包的 PID, 进行发送。
- DATA1: 不发送。
- DATA2: 不发送。
- mData: 不发送。

2. OUT 方向 (全速运行时)

- DATA0: 作为数据包的 PID, 正常接收。
- DATA1: 作为数据包的 PID, 正常接收。
- DATA2: 忽视。
- mData: 忽视。

3. OUT 方向 (高速运行时)

- DATA0: 作为数据包的 PID, 正常接收。
- DATA1: 作为数据包的 PID, 正常接收。
- DATA2: 作为数据包的 PID, 正常接收。
- mData: 作为数据包的 PID, 正常接收。

(3) 间隔计数器

能通过 PIPEPERI 寄存器的 IITV 位设定等时传送的间隔。在选择功能控制器功能时, 通过间隔计数器实现如表 21.30 所示的功能; 在选择主机控制器功能时, 通过间隔计数器生成标记的发行时序。选择主机控制器功能时的间隔计数器运行和中断传送的运行相同。

表 21.30 选择功能控制器功能时的间隔计数器的功能

传送方向	功能	检测条件
IN	发送缓冲器的清除功能	等时 IN 传送时, 在间隔帧中不能正常接收 IN 标记。
OUT	标记未接收的通知	等时 OUT 传送时, 在间隔帧中不能正常接收 OUT 标记。

通过接收 SOF 或者内插 SOF 进行间隔计数, 因此即使 SOF 遭到破损也能保证等时性。能设定的帧间隔为 2^{IITV} 帧或者 $2^{IITV}\mu$ 帧。

(a) 选择功能控制器功能时的计数器的初始化

本模块根据下列条件对间隔计数器进行初始化:

- 上电复位
对 IITV 位进行初始化。
- 通过 ACLRM 进行的缓冲存储器初始化
不对 IITV 位进行初始化, 而对计数值进行初始化。通过将 ACLRM 位置 0, 从 IITV 的设定值开始计数。

在对间隔计数器进行初始化后, 正常传送信息包, 然后如果满足以下条件, 就开始进行的间隔计数。

1. PID=BUF 时, 对于 IN 标记, 在发送数据后接收 SOF
2. PID=BUF 时, 对于 OUT 标记, 在接收数据后接收 SOF

在满足下列条件时, 不对间隔计数器进行初始化:

3. 将 PID 位设定为 NAK 或者 STALL
间隔定时器不停止计数, 在下一个间隔中尝试执行事务。
4. USB 总线复位和 USB 挂起
不对 IITV 位进行初始化。如果接收到 SOF, 就从接收前的值开始计数。

(b) 选择主机控制器功能时的间隔计数和传送控制

本模块根据 IITV 位的设定值控制标记的发行间隔。对选择的管道, 每 2^{IITV} 的 (μ) 帧发行 1 个标记。对高速 HUB 连接的全速 / 低速外围设备进行通信的管道, 每 1ms 帧进行一次间隔计数。从通过软件将 PID 位设定为 BUF 的下一个 (μ) 帧开始进行标记发行间隔的计数。

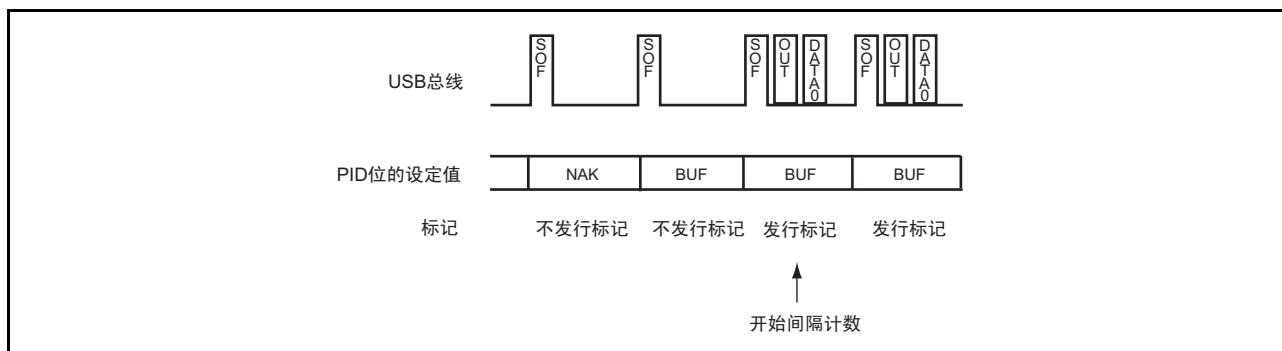


图 21.12 IITV=0 时, 是否发行 Token

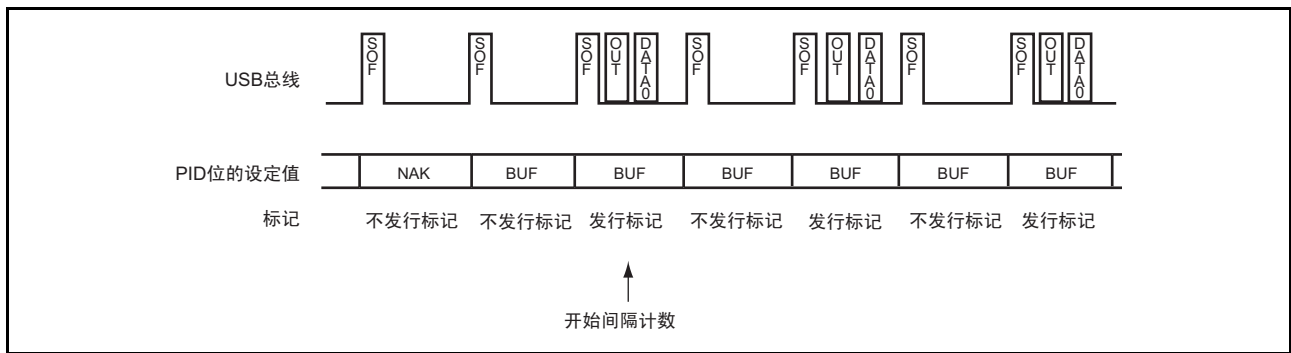


图 21.13 IITV=1 时, 是否发行 Token

在选择管道的传送类型为等时传送时, 本模块除了控制标记的发行间隔以外, 还进行以下运行。即使满足NRDY 中断产生条件, 也发行标记。

1. 选择管道为 Isochronous-IN 传送管道时

发行 In-Token, 在没有正常从外围设备接收到信息包时 (无应答或者包错误等时), 产生NRDY 中断。如果在因FIFO缓冲器满而不能接收数据的状态下 (这是因为软件 (DMAC) 从FIFO缓冲器读数据的速度慢) 到了发行IN-Token的时间, 本模块就将OVRN位置1, 产生NRDY 中断。

2. 选择管道为 Isochronous-OUT 传送管道时

如果在FIFO缓冲器中没有能发送数据的状态下 (这是因为软件 (DMAC) 给FIFO缓冲器写数据的速度慢) 到了发行OUT-TOKEN的时间, 本模块就将OVRN位置1, 产生NRDY 中断, 并且发送Zero-Length包。

标记发行间隔的复位条件如下:

- 本模块进行硬件复位时 (此时, IITV位的设定值也被清0。)
- 通过软件设定 ACLRM=1 时

(c) 选择功能控制器功能时的间隔计数和传送控制

1. 选择管道为 Isochronous-OUT 传送管道时

在IITV位设定的各间隔的 (μ) 帧中没有接收到DATA包时, 该控制器产生NRDY 中断。

在因DATA包发生CRC错误等而不能接收或者因FIFO缓冲器满而不能接收数据时, 产生NRDY 中断。NRDY 中断的产生时序是接收到SOF包时。即使SOF包破损, 也通过内插功能在应该接收SOF时产生中断。

但是, 如果IITV \neq 0, 就在间隔计数开始后的各间隔接收SOF包时, 产生NRDY 中断。

在启动间隔定时器后通过软件将PID位设定为NAK时, 即使接收到SOF包也不产生NRDY 中断。

间隔计数的开始条件因IITV位的设定值而不同。

- 在IITV=0时, 从将所选管道的PID位变为BUF的下一个 (μ) 帧开始间隔计数。
- IITV \neq 0时, 从将所选管道的PID位变为BUF后正常接收完最初的DATA包时开始间隔计数。

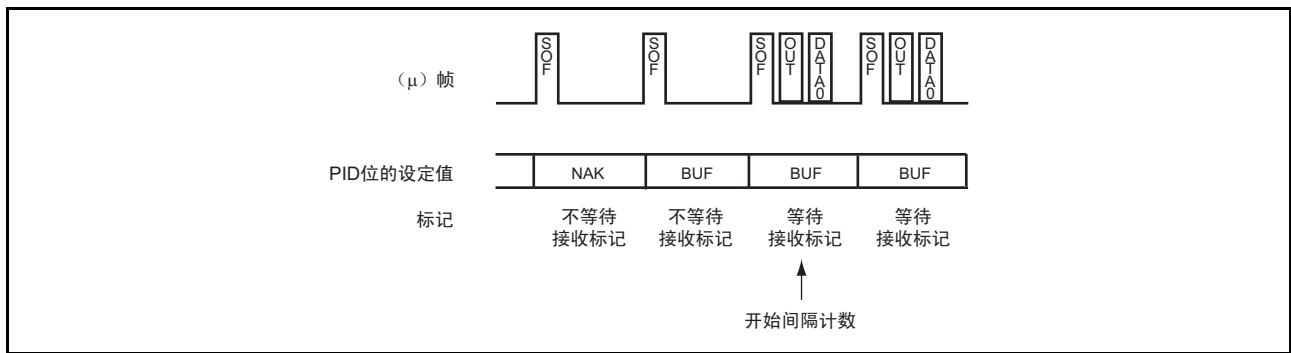


图 21.14 IITV=0 时的 (μ) 帧和是否期待接收 Token 的关系

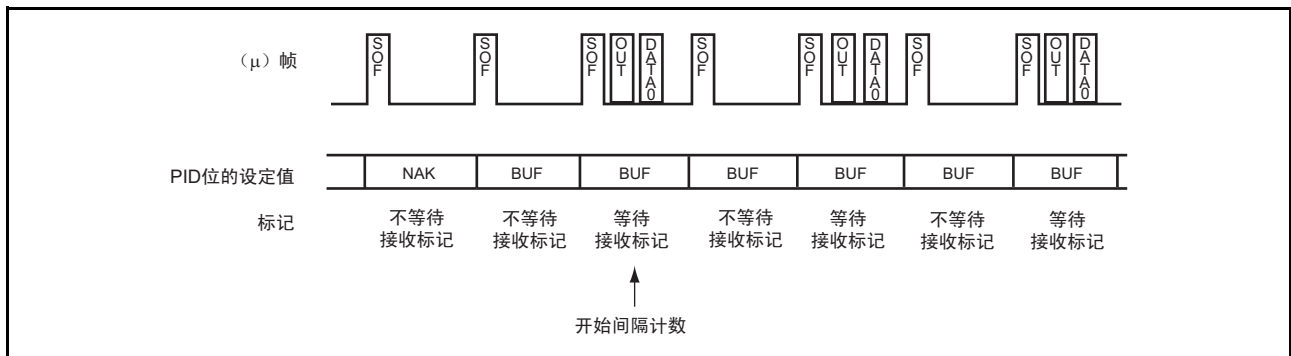


图 21.15 IITV≠0 时的 (μ) 帧和是否期待接收 Token 的关系

2. 选择管道为 Isochronous-IN 传送管道时

与 IFIS=1 一起使用。在 IFIS=0 时，与 IITV 位的设定值无关，对接收的标记进行应答，并且发送数据包。在 IFIS=1 时并且 FIFO 缓冲器中有能发送数据的状态下，如果在 IITV 位设定的各间隔的 (μ) 帧中没有接收到 IN-Token，就清除 FIFO 缓冲器。

即使在因 IN-Token 发生 CRC 错误等的总线错误而不能正常接收时清除 FIFO 缓冲器。

FIFO 缓冲器的清除时序是接收到 SOF 包时。即使 SOF 包破损，也通过内插功能在应该接收 SOF 时清除 FIFO 缓冲器。

间隔计数的开始条件因 IITV 位的设定值而不同（与 OUT 时相同）。

选择功能控制器功能时的间隔计数条件如下：

- 本模块进行硬件复位时（此时，IITV 位的设定值也被清 0。）
- 通过软件设定 ACLRM=1 时
- 本模块检测到 USB 复位时

(4) 选择功能控制器功能时的等时传送的发送数据设置

在选择功能控制器功能时，本模块的等时数据发送在给缓冲存储器写数据后，能通过检测到 SOF 包的下一个帧发送数据包。此功能称为等时传送的发送数据设置功能。通过此功能能指定开始发送的帧。

在缓冲存储器用作双缓冲器时，即使写完 2 个面的缓冲器，可传送状态的缓冲存储器也只为先写完的 1 个面。因此，在同一个帧接收到多个 IN 标记时，被发送的缓冲存储器只为 1 个包的数据。

在接收到 IN 标记时，如果缓冲存储器为可发送状态，就传送数据，且进行正常的应答。但是，如果缓冲存储器为不能发送状态，就发送 Zero-Length 包，并且发生欠载错误。

在 IITV=0（每个帧）时，通过等时传送的发送数据设置功能进行发送的例子如图 21.16 所示。图中带阴影的 Null 表示 Zero-Length 包的发送。

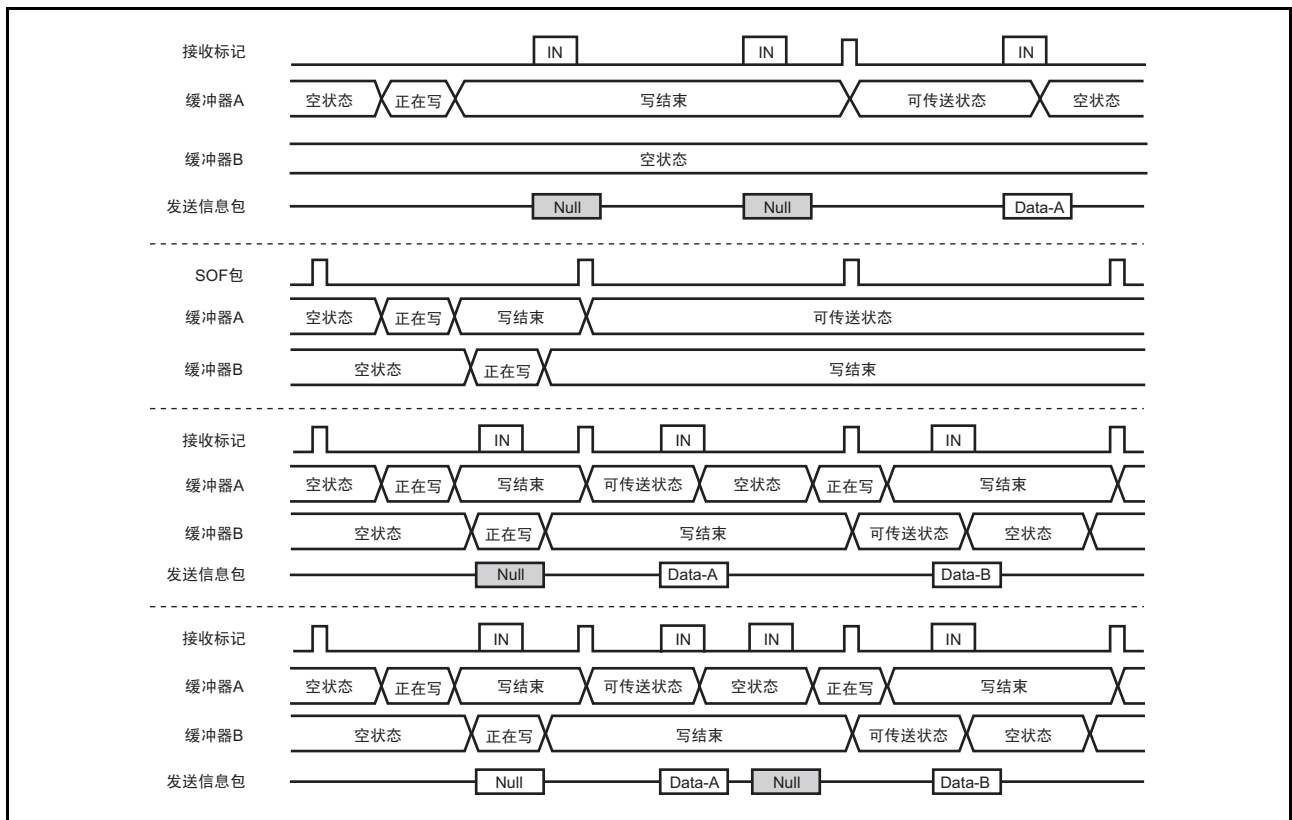


图 21.16 数据设置功能的运行例子

(5) 选择功能控制器功能时的等时传送发送缓冲器清除

在选择功能控制器功能时，如果在进行等时数据发送时的间隔帧中没有接收到 IN 标记，而接收到下一个帧的 SOF 或者 μ SOF 包，就作为 IN 标记破损进行处理，并且清除处于可发送状态的缓冲器，使该缓冲器变为可写状态。

此时，如果使用双缓冲器并且写完 2 个缓冲器面，放弃的缓冲存储器就被看作是同一个间隔帧发送的数据，并且将接收 SOF 或者 μ SOF 包时未放弃的缓冲存储器设定为可传送状态。

缓冲器清除功能的运行开始时序因 IITV 位的设定值而不同。

- IITV=0 时
从管道有效的下一个帧开始运行缓冲器清除。
- IITV \neq 0 时
在最初的正常事务后，运行缓冲器清除。

本模块的缓冲器清除功能的运行例子如图 21.17 所示。但是，对于间隔外的标记（间隔帧之前的标记），根据数据设置状态，发送写数据或者发送作为欠载错误的 Zero-Length 包。

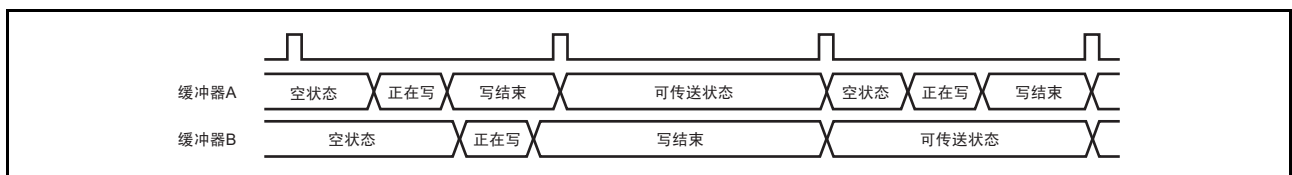


图 21.17 缓冲器清除功能的运行例子

本模块的间隔错误的发生例子如图 21.18 所示。间隔错误有以下 5 种。在图中的①时序，发生间隔错误，并且进行缓冲器的清除。

如果在 IN 传送时发生间隔错误，就进行缓冲器的清除；如果在 OUT 传送时发生间隔错误，就产生 NRDY 中断。

必须根据 OVRN 位判断超限错误和 NRDY 中断（接收包错误等）。

图中是阴影的标记是根据缓冲存储器状态进行的应答。

1. IN 方向

- 如果缓冲器为可传送状态，就传送数据，并且进行正常的应答。
- 如果缓冲器不为可传送状态，就发送 Zero-Length 包并，并且产生欠载错误。

2. OUT 方向

- 如果缓冲器为可接收状态，就接收数据，并且进行正常的应答。
- 如果为缓冲器不为可接收状态，就放弃数据，并且产生超限错误。

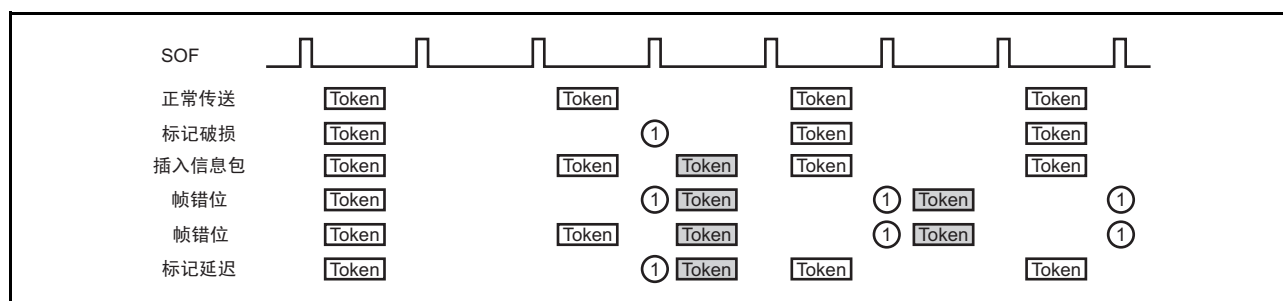


图 21.18 IITV=1 时的间隔错误的发生例子

21.4.9 SOF 内插功能

在选择功能控制器功能时，如果因 SOF 包的破损或者丢失而在 1ms（全速运行时）或者 125 μ s（高速运行时）的间隔内不能接收到 SOF 包，本模块就内插 SOF。在 SYSCFG.USBE=1、SYSCFG.SCKE=1 并且接收 SOF 包时，进行 SOF 内插。在以下情况下，对内插功能进行初始化：

- 上电复位
- USB 总线复位
- 检测到挂起

SOF 内插按以下规格进行：

- 帧间隔（125 μ s 或者 1ms）取决于复位握手协议的结果。
- 在接收 SOF 包前不进行内插。
- 在接收到最初的 SOF 包后，通过 48MHz 的内部时钟对 125 μ s 或者 1ms 进行计数，并且进行内插。
- 在接收第二次以后的 SOF 包后，使用前一次的接收间隔进行内插。
- 在挂起时或者接收到 USB 总线复位时不进行内插。

（在高速运行的挂起转移时，在接收最终包后的 3ms 内继续进行内插。）

本模块根据接收的 SOF 包进行以下运行，但是在 SOF 包丢失时进行 SOF 内插，所以能继续正常运行。

- 帧号和 μ 帧号的更新
- SOFR 中断时序和 μ SOFR 的锁定
- 等时传送的间隔计数

在全速运行时，如果丢失 SOF 包，就不更新 FRMNUM 寄存器的 FRNM 位；如果丢失 μ SOF 包，就更新 UFRMNUM 寄存器的 UFRNM 位。但是，在丢失 UFRNM=000 的 μ SOF 包时，不更新 FRNM 位。此时，即使正常接收到连续的 UFRNM \neq 000 的 μ SOF 包，也不更新 FRNM 位。

21.4.10 管道的安排

(1) 事务发行条件

在选择主机控制器功能时，本模块在设定 UACT=1 后以表 21.31 所示的条件发行事务。

表 21.31 事务发行条件

事务	发行条件				
	DIR	PID	IITV0	缓冲器的状态	SUREQ
设置	—*1	—*1	—*1	—*1	设定 1
控制传送的数据阶段、状态阶段、批量 传送	IN	BUF	无效	有接收区	—*1
	OUT	BUF	无效	有发送数据	—*1
中断传送	IN	BUF	有效	有接收区	—*1
	OUT	BUF	有效	有发送数据	—*1
等时传送	IN	BUF	有效	*2	—*1
	OUT	BUF	有效	*3	—*1

【注】 *1 表中的“—”是表示与标记的发行无关的条件。“有效”表示对于中断传送和等时传送，只在与间隔计数器有关的传送帧中发行。“无效”是表示与间隔计数器无关的条件。

*2 不管有没有接收区，都发行事务。但是，在没有接收区时，放弃接收的数据。

*3 不管有没有发送数据，都发行事务。但是，在没有发送数据时，发送 Zero-Length 包。

(2) 传送的安排

说明本模块的帧内的传送安排方法。在发送 SOF 后，按照以下顺序进行传送：

1. 周期性传送的执行

按照管道 1→管道 2→管道 6→管道 7→管道 8→管道 9 的顺序查找管道，如果有能发行等时传送或者中断传送事务的管道，就发行事务。

2. 控制传送的设置事务

确认 DCP，如果能进行处理设置事务，就发送设置事务。

3. 批量、控制传送数据阶段和状态阶段的执行

按照 DCP→管道 1→管道 2→管道 3→管道 4→管道 5 的顺序查找管道，如果有能发行批量、控制传送数据阶段和控制传送状态阶段事务的管道，就执行事务。

在发行事务后，无论外围设备的应答是 ACK 还是 NAK，都转移到下一个管道的事务。如果帧内有传送的时间，就重复执行 3。

(3) USB 通信的允许

通过将 DVSTCTR 寄存器的 UACT 位置 1，开始 SOF 或者 μ SOF 的发送，并且进入可发行事务的状态。

如果将 UACT 位置 0，就停止 SOF 或者 μ SOF 的发送，进入挂起状态。当将 UACT 位从 1 置为 0 时，就在发送下一个 SOF 或者 μ SOF 后停止。

21.5 使用时的注意事项

21.5.1 USB 模块的启动 / 停止步骤

(1) 启动步骤

必须按照以下步骤启动 USB 模块：

1. 在刷新待机模式时，必须在产生 IRQn 中断或者 NMI 中断后，将低功耗模式的 STBCR 寄存器的 STBY 位置 0，并且返回到正常运行模式。
2. 为了确保振荡稳定时间，必须等待 3 ~ 5ms。
3. 必须将 CPG 的 MSTPCR0 寄存器的 USB 位置 0，SYSCFG 寄存器的 SCKE 位置 1，解除模块待机状态。

(2) 停止步骤

必须按照以下步骤停止 USB 模块：

1. 必须将 SYSCFG 寄存器的 SCKE 位置 0，CPG 的 MSTPCR0 寄存器的 USB 位置 1，将 USB 模块设定为模块待机状态。
2. 必须至少等待 40ns（2 个 48MHz 时钟周期）。
3. 必须将低功耗模式的 STBCR 寄存器的 STBY 位置 1，转换到刷新待机模式。

【注】 必须成对设定 MSTPCR0 寄存器的 USB 位和 SYSCFG 寄存器的 SCKE 位。

第 22 章 LCD 控制器 (LCDC)

LCD 控制器 (LCDC) 采用将显示图像保存在系统存储器的统一图形存储器体系结构。LCDC 模块从系统存储器读数据, 在使用配色存储器确定颜色后, 将数据传送到 LCD 显示屏。能连接除了单片机总线接口方式、NTSC/PAL 方式和 LVDS 接口 * 以外的液晶模块。

【注】 * 通过连接 LVDS 转换 LSI, 能连接 LVDS 接口。

22.1 特点

LCDC 有以下特点:

- 显示屏接口
串行接口方式
支持 STN/Dual STN/TFT 显示屏 (8/12/16/18 位总线) 的数据格式 *1。
- 支持 4/8/15/16bpp (位每像素) 彩色模式。
- 支持 1/2/4/6bpp 灰度模式。
- 支持 16×1 ~ 1024×1024 的液晶显示屏尺寸 *2。
- 24 位彩色配色存储器 (在 24 位中, 16 位有效 R:5/G:6/B:5)
- 通过 8 位 RGB 的 24 位空间调制 FRC, 在容易产生闪烁和阴影的 STN/DSTN 显示屏上实现闪烁较少的 65536 的色彩控制。
- 将连接 CPU 的 SDRAM (区域 1、2) 的一部分用作保存 LCDC 显示数据的 VRAM, 不需要专用的显示存储器。
- 通过 2.4K 字节的大容量行缓冲器, 实现稳定的显示。
- 为了适合液晶显示屏的信号极性, 支持输出信号的电平反相功能。
- 能通过寄存器支持各种数据格式的选择 (设定字节排序, 背景像素方式)。
- 能在用户指定的位置产生中断 (通过控制 VRAM 的更新开始时序进行闪烁的防止)。

【注】 *1 在连接 18 位总线的 TFT 显示屏时, 必须将未接线的低位信号连接到 GND 或者输出数据的最低位。

*2 详细内容请参照 “22.4.1 LCDC 能显示的液晶模块的尺寸”。

LCDC 的框图如图 22.1 所示。

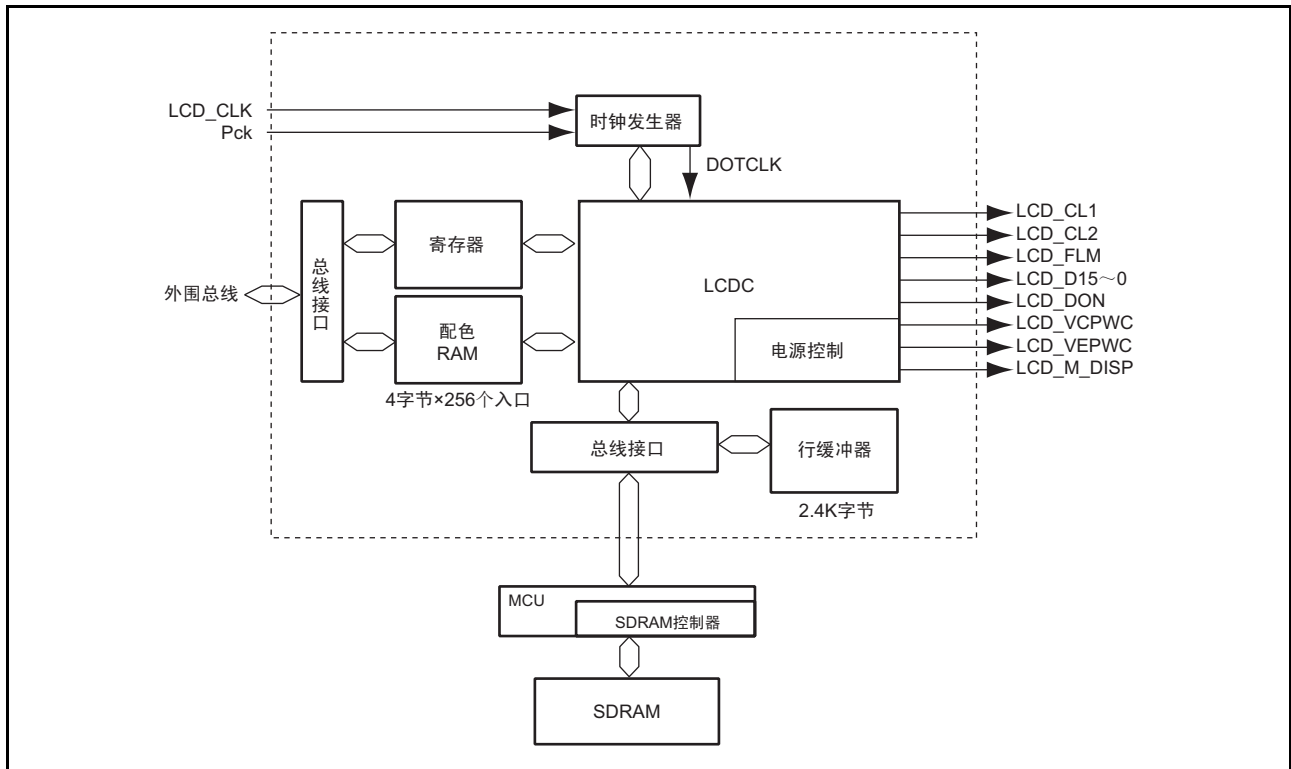


图 22.1 LCDC 的框图

22.2 输入 / 输出引脚

LCDC 的引脚结构如表 22.1 所示。

通过 PFC 引脚选择寄存器进行 LCDC 引脚的选择。

表 22.1 引脚结构

引脚名	输入 / 输出	功能
LCD_D15 ~ 0	输出	用于液晶显示屏的数据
LCD_DON	输出	显示开始信号 (DON)
LCD_CL1	输出	移位时钟 1 (STN/DSTN) / 水平同步信号 (HSYNC)
LCD_CL2	输出	移位时钟 2 (STN/DSTN) / 点时钟 (DOTCLK)
LCD_M_DISP	输出	液晶交流信号 / DISP 信号
LCD_FLM	输出	首行标记 / 垂直同步信号 (VSYNC) (TFT)
LCD_VCPWC	输出	液晶模块电源控制 (VCC)
LCD_VEPWC	输出	液晶模块电源控制 (VEE)
LCD_CLK*	输入	LCD 时钟源输入

【注】 有关与液晶模块的接线规格，请确认“22.5 时钟与 LCD 数据信号例子”和液晶模块的规格。

22.3 寄存器说明

LCDC 的寄存器结构如表 22.2 所示，各处理模式的寄存器状态如表 22.3 所示。

表 22.2 寄存器结构

名称	略称	R/W	P4 区地址	区域 7 地址	存取长度
配色数据寄存器 00 ~ FF	LDPR00 ~ LDPRFF	R/W	H'FFE3 0000 ~ H'FFE3 03FC	H'1FE3 0000 ~ H'1FE3 03FC	32
LCDC 输入时钟寄存器	LDICKR	R/W	H'FFE3 0400	H'1FE3 0400	16
LCDC 模块类型寄存器	LDMTR	R/W	H'FFE3 0402	H'1FE3 0402	16
LCDC 数据格式寄存器	LDDFR	R/W	H'FFE3 0404	H'1FE3 0404	16
LCDC 上部显示屏的数据开始地址寄存器	LDSARU	R/W	H'FFE3 0408	H'1FE3 0408	32
LCDC 下部显示屏的数据开始地址寄存器	LDSARL	R/W	H'FFE3 040C	H'1FE3 040C	32
LCDC 显示屏的数据行地址偏移寄存器	LDLAOR	R/W	H'FFE3 0410	H'1FE3 0410	16
LCDC 配色控制寄存器	LDPALCR	R/W	H'FFE3 0412	H'1FE3 0412	16
LCDC 水平字符数寄存器	LDHCNR	R/W	H'FFE3 0414	H'1FE3 0414	16
LCDC 水平同步信号寄存器	LDHSYNR	R/W	H'FFE3 0416	H'1FE3 0416	16
LCDC 垂直显示行数寄存器	LDVDLNR	R/W	H'FFE3 0418	H'1FE3 0418	16
LCDC 垂直总行数寄存器	LDVTLNR	R/W	H'FFE3 041A	H'1FE3 041A	16
LCDC 垂直同步信号寄存器	LDVSYNR	R/W	H'FFE3 041C	H'1FE3 041C	16
LCDC AC 调制信号交替行数寄存器	LDACLNR	R/W	H'FFE3 041E	H'1FE3 041E	16
LCDC 中断控制寄存器	LDINTR	R/W	H'FFE3 0420	H'1FE3 0420	16
LCDC 电源管理模式寄存器	LDPMMR	R/W	H'FFE3 0424	H'1FE3 0424	16
LCDC 电源顺序期间寄存器	LDPSPR	R/W	H'FFE3 0426	H'1FE3 0426	16
LCDC 控制寄存器	LDCNTR	R/W	H'FFE3 0428	H'1FE3 0428	16
LCDC 用户指定中断控制寄存器	LDUINTR	R/W	H'FFE3 0434	H'1FE3 0434	16
LCDC 用户指定中断行数寄存器	LDUINLNR	R/W	H'FFE3 0436	H'1FE3 0436	16
LCDC 存储器存取间隔数寄存器	LDLIRNR	R/W	H'FFE3 0440	H'1FE3 0440	16

表 22.3 各处理模式的寄存器状态

名称	略称	上电复位	休眠	待机
配色数据寄存器 00 ~ FF	LDPR00 ~ LDPRFF	不定值	保持	保持
LCDC 输入时钟寄存器	LDICKR	H'1101	保持	保持
LCDC 模块类型寄存器	LDMTR	H'0109	保持	保持
LCDC 数据格式寄存器	LDDFR	H'000C	保持	保持
LCDC 上部显示屏的数据开始地址寄存器	LDSARU	H'0400 0000	保持	保持
LCDC 下部显示屏的数据开始地址寄存器	LDSARL	H'0400 0000	保持	保持
LCDC 显示屏的数据行地址偏移量寄存器	LDLAOR	H'0280	保持	保持
LCDC 配色控制寄存器	LDPALCR	H'0000	保持	保持
LCDC 水平字符数寄存器	LDHCNR	H'4F52	保持	保持
LCDC 水平同步信号寄存器	LDHSYNR	H'0050	保持	保持
LCDC 垂直显示行数寄存器	LDVDLNR	H'01DF	保持	保持
LCDC 垂直总行数寄存器	LDVTLNR	H'01DF	保持	保持
LCDC 垂直同步信号寄存器	LDVSYNR	H'01DF	保持	保持
LCDC AC 调制信号交替行数寄存器	LDACLNR	H'000C	保持	保持
LCDC 中断控制寄存器	LDINTR	H'0000	保持	保持
LCDC 电源管理模式寄存器	LDPMMR	H'0010	保持	保持
LCDC 电源顺序期间寄存器	LDPSPR	H'F60F	保持	保持
LCDC 控制寄存器	LDCNTR	H'0000	保持	保持
LCDC 用户指定中断控制寄存器	LDUINTR	H'0000	保持	保持
LCDC 用户指定中断行数寄存器	LDUINLNR	H'004F	保持	保持
LCDC 存储器存取间隔数寄存器	LDLIRNR	H'0000	保持	保持

22.3.1 LCDC 输入时钟寄存器 (LDICKR)

LCDC 能将外围时钟或者外部时钟选择为 LCDC 工作时钟的供给源，并且内置 1/1 ~ 1/32 的分频器，能将分频的时钟用作 LCDC 的工作时钟 (DOTCLK)。LCDC 的输出时钟是从此寄存器选择的工作时钟生成的用于液晶显示屏的同步时钟输出 (LCD_CL2)。在 TFT 显示屏时，LCD_CL2=DOTCLK；在 STN 和 DSTN 显示屏时，LCD_CL2= (DOTCLK/ 输出到液晶显示屏的数据总线宽度) 的频率时钟。在设定 LDICKR 时，与 LCD_CL2 无关，输入到 LCDC 的时钟不能超过外围时钟 (Pck)。

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

位	位名	初始值	R/W	说明
15、14	—	全 0	R	保留位 读取值为 0，只能写 0。
13、12	ICKSEL[1:0]	01	R/W	输入时钟选择位 设定 DOTCLK 的供给源。 00: 禁止设定 01: 选择外围时钟 (Pck) 10: 选择外部时钟 (LCD_CLK) 11: 禁止设定
11 ~ 9	—	全 0	R	保留位 读取值为 0，只能写 0。
8	—	1	R	保留位 读取值为 1，只能写 1。
7、6	—	全 0	R	保留位 读取值为 0，只能写 0。
5 ~ 0	DCDR[5:0]	000001	R/W	时钟分频比 设定输入时钟的分频比。 设定的详细内容请参照表 22.4。

表 22.4 输入 / 输出时钟频率和分频比

DCDR[5:0]	时钟分频比	输入 / 输出时钟频率 (MHz)	
		50.000	54.000
000001	1/1	50.000	54.000
000010	1/2	25.000	27.000
000011	1/3	16.667	18.000
000100	1/4	12.500	13.500
000110	1/6	8.333	9.000
001000	1/8	6.250	6.750
001100	1/12	4.167	4.500
010000	1/16	3.125	3.375
011000	1/24	2.083	2.250
100000	1/32	1.563	1.688

【注】 在设定上述以外的值时，时钟分频比为 1/1 (初始值)。

22.3.2 LCDC 模块类型寄存器 (LDMTR)

为了适合连接的液晶模块的信号极性，LDMTR 设定 LCDC 输出的控制信号和数据信号的极性。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FLM POL	CL1 POL	DISP POL	DPOL	—	MCNT	CL1CNT	CL2CNT	—	—	MIFTYP[5:0]					
初始值:	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15	FLMPOL	0	R/W	FLM (垂直同步) 极性选择位 选择液晶模块 LCD_FLM (垂直同步信号和首行标记) 的极性。 0: LCD_FLM 脉冲为高电平有效 1: LCD_FLM 脉冲为低电平有效
14	CL1POL	0	R/W	CL1 (水平同步) 极性选择位 选择液晶模块 LCD_CL1 (水平同步信号) 的极性。 0: LCD_CL1 脉冲为高电平有效 1: LCD_CL1 脉冲为低电平有效
13	DISPPOL	0	R/W	DISP (允许显示) 极性选择位 选择液晶模块 LCD_M_DISP (允许显示) 的极性。 0: LCD_M_DISP 为高电平有效 1: LCD_M_DISP 为低电平有效
12	DPOL	0	R/W	显示数据的极性选择位 选择液晶模块 LCD_D15 ~ 0 (显示数据) 的极性。支持液晶模块的反映。 0: LCD_D15 ~ 0 为高电平有效。透射式液晶显示屏 1: LCD_D15 ~ 0 为低电平有效。反射型液晶显示屏
11	—	0	R	保留位 读取值为 0, 只能写 0。
10	MCNT	0	R/W	M 信号控制位 设定液晶模块的液晶交流信号的输出。 0: 输出 M (AC 线路调制) 信号 1: 不输出 M 信号
9	CL1CNT	0	R/W	CL1 (水平同步) 控制位 设定垂直回扫期间的 LCD_CL1 输出。 0: 在垂直回扫周期输出 LCD_CL1 1: 在垂直回扫周期不输出 LCD_CL1
8	CL2CNT	1	R/W	CL2 (液晶模块的点时钟) 控制位 设定垂直水平回扫期间的 LCD_CL2 输出。 0: 在垂直水平回扫周期输出 LCD_CL2 1: 在垂直水平回扫周期不输出 LCD_CL2
7、6	—	全 0	R	保留位 读取值为 0, 只能写 0。

位	位名	初始值	R/W	说明
5 ~ 0	MIFTYP[5:0]	001001	R/W	<p>模块接口类型选择位</p> <p>设定液晶显示屏的类型和输出到液晶显示屏的数据总线宽度。液晶显示屏的类型能选择 STN、DSTN 或者 TFT。输出到液晶显示屏的数据总线宽度能选择 4 位、8 位、12 位或者 16 位。当 TFT 液晶显示屏的请求数据总线宽度大于等于 16 位时，必须按照液晶显示屏侧的数据总线宽度进行连接。与 TFT 不同，由于 STN 或者 DSTN 液晶显示屏的显示色数、显示分辨率和输出数据总线宽度的设定不一一对应，因此 16bpp 的显示色数可能使用 8 位数据总线，而 4bpp 的显示色数可能使用 12 位数据总线。这是因为 STN 或者 DSTN 的显示色数并不是由数据总线的位数决定，而是由数据总线的数据规格决定。有关 STN 或者 DSTN 的数据规格，请参照所使用的液晶显示屏的规格说明书。另外，必须按照液晶显示屏的机械接口规格进行输出数据总线宽度的设定。</p> <p>在选择 STN 或者 DSTN 液晶显示屏时，与色彩显示、灰度显示的等级设定无关，通过 LCDC 内的 8 位 RGB 的 24 位间隔调制 FRC 进行显示控制。因此，从 1600 万色中选择 DSPCOLOR 指定的颜色和灰度进行 STN 或者 DSTN 的显示。当使用调色板时，就显示调色板设定的颜色。</p> <p>000000: STN 单色 4 位数据总线模块 000001: STN 单色 8 位数据总线模块 001000: STN 彩色 4 位数据总线模块 001001: STN 彩色 8 位数据总线模块 001010: STN 彩色 12 位数据总线模块 001011: STN 彩色 16 位数据总线模块 010001: DSTN 单色 8 位数据总线模块 010011: DSTN 单色 16 位数据总线模块 011001: DSTN 彩色 8 位数据总线模块 011010: DSTN 彩色 12 位数据总线模块 011011: DSTN 彩色 16 位数据总线模块 101011: TFT 彩色 16 位数据总线模块 上述以外: 禁止设定</p>

22.3.3 LCDC 数据格式寄存器 (LDDFR)

为了适合显示的驱动软件的规格，LDDFR 对 1 字节内的数据进行位调整以及对显示数据的类型和色数进行设定。

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

位	位名	初始值	R/W	说明
15 ~ 9	—	全 0	R	保留位 读取值为 0，只能写 0。
8	PABD	0	R/W	字节数据的像素调整位 设定 1 字节数据内的像素数据的调整种类。与此位的内容无关，调整后的每个像素的数据内容相同。例如：数据 H'05 并不是选择 2 进制的 B'0101 或者 B'1010，而是作为 CPU 的 MOV 指令正常处理的 H'05 (B'0101)。 0：将字节数据设定为大端法 1：将字节数据设定为小端法
7	—	0	R	保留位 读取值为 0，只能写 0。
6 ~ 0	DSPCOLO[6:0]	0001100	R/W	显示颜色选择位 设定显示色数（给 4 ~ 6bpp 的高位置 0）。 有关有“（通过调色板）”记述的显示颜色，实际上通过显示数据选择彩色调色板设定的颜色，并且进行显示。 在旋转显示时，能支持的色数受显示分辨率限制。 0000000：单色、2 灰度、1bpp（通过调色板） 0000001：单色、4 灰度、2bpp（通过调色板） 0000010：单色、16 灰度、4bpp（通过调色板） 0000100：单色、64 灰度、6bpp（通过调色板） 0001010：彩色、16 色、4bpp（通过调色板） 0001100：彩色、256 色、8bpp（通过调色板） 0011101：彩色、32k 色（RGB: 5-5-5）、15bpp 0101101：彩色、64k 色（RGB: 5-6-5）、16bpp 上述以外：禁止设定

22.3.4 LCDC 上部显示屏的数据开始地址寄存器 (LDSARU)

LDSARU 指定液晶显示屏的显示数据 (取到 LCDC) 的开始地址。在使用 DSTN 型的液晶显示屏时, 此寄存器指定上部显示屏的显示数据的开始地址。

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

位	位名	初始值	R/W	说明
31 ~ 28	—	全 0	R	保留位 读取值为 0, 只能写 0。
27	SAU[27]	0	R/W	上部显示屏的显示数据的开始地址 显示数据的开始地址设定在区域 1、2 的 SDRAM 区域内。
26	SAU[26]	1	R/W	
25 ~ 4	SAU[25:4]	全 0	R/W	
3 ~ 0	—	全 0	R	保留位 读取值为 0, 只能写 0。

【注】 LDSARU 的最小调整单位是 512 字节, 必须将低 9 位置 0。

22.3.5 LCDC 下部显示屏的数据开始地址寄存器 (LDSARL)

在使用 DSTN 型的液晶显示屏时, LDSARL 指定下部显示屏的显示数据的开始地址。

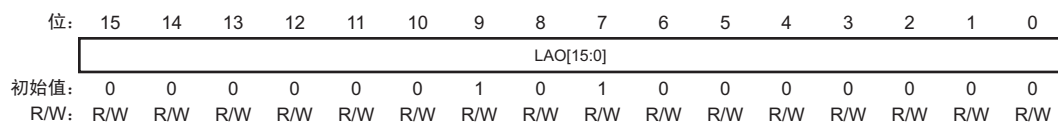
位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	SAL[27:16]											
初始值:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SAL[15:4]												—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W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

位	位名	初始值	R/W	说明
31 ~ 28	—	全 0	R	保留位 读取值为 0, 只能写 0。
27	SAL[27]	0	R/W	下部显示屏的显示数据的开始地址 显示数据的开始地址设定在区域 1、2 的 SDRAM 区域内。 STN、TFT: 不使用 DSTN: 对应下部显示屏的显示数据的开始地址
26	SAL[26]	1	R/W	
25 ~ 4	SAL[25:4]	全 0	R/W	
3 ~ 0	—	全 0	R	保留位 读取值为 0, 只能写 0。

【注】 LDSARL 的最小调整单位是 32 字节, 必须将低 5 位置 0。

22.3.6 LCDC 显示屏的数据行地址偏移寄存器 (LDLAOR)

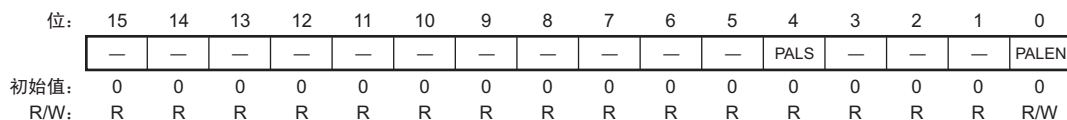
LDLAOR 指定用于 LCDC 读取图像 (图形驱动器认识的图像) 的 Y 坐标增量的地址距离。在 Y 坐标方向增 1 时, LDLAOR 指定要移动的字节地址距离 (移动地址后读存储器的数据), 所以不需要与液晶显示屏的宽度保持一致。在用 $Ax+By+C$ 的公式计算 2 维图像点 (X, Y) 的存储器地址时, 此寄存器就是此公式中的 B。



位	位名	初始值	R/W	说明
15 ~ 0	LAO[15:0]	H'0280	R/W	行地址偏移量 最小调整单位是 32 字节。由于以 32 字节为单位进行处理, 所以必须将各寄存器的低 5 位置 0。另外, 此寄存器的读取值的低 5 位为 0。初始值是连续密集配置 VGA (640×480 点) 显示数据的设定值 (×分辨率=640)。考虑到软件的运行速度, 建议 LDLAOR 的值至少为图像水平宽度的 2 次方。

22.3.7 LCDC 配色控制寄存器 (LDPALCR)

LDPALCR 选择是由 CPU 还是由 LCDC 存取配色存储器。在使用配色存储器进行显示时, 必须设定为正常显示模式; 在改写配色存储器的内容时, 必须设定为彩色调色板设定模式。



位	位名	初始值	R/W	说明
15 ~ 5	—	全 0	R	保留位 读取值为 0, 只能写 0。
4	PALS	0	R	调色板状态位 表示调色板存取权的状态 0: LCDC 使用调色板。正常显示模式 1: 主机 (CPU) 使用调色板。彩色调色板设定模式
3 ~ 1	—	全 0	R	保留位 读取值为 0, 只能写 0。
0	PALEN	0	R/W	调色板读写允许位 请求调色板的存取权。 0: 请求转移到正常显示模式 1: 请求转移到彩色调色板设定模式

22.3.8 配色数据寄存器 00 ~ FF (LDPR00 ~ LDPRFF)

LDPR 是用于存取直接配置 (4 字节 × 256 个地址) 在存储器空间的配色数据的寄存器。在对配色存储器进行存取时, 必须存取此寄存器 (LDPR00 ~ LDPRFF) 中的相应寄存器。每个配色寄存器是指向 3 个 8 位 RGB 区域的 32 位寄存器。有关此彩色调色板的详细规格, 请参照“22.4.2 彩色调色板规格”。

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

位	位名	初始值	R/W	说明
31 ~ 24	—	—	R	保留位
23 ~ 0	PALDnn[23:0]	—	R/W	配色数据 bit18 ~ 16、bit9、bit8、bit2 ~ 0 是 RGB 各调色板内的保留位。不能设定, 但是能根据高位进行扩展使用。

【注】 nn=H'00 ~ H'FF

22.3.9 LCDC 水平字符数寄存器 (LDHCNR)

LDHCNR 是指定液晶模块水平方向 (扫描方向) 的尺寸和包括水平回扫周期的全体扫描宽度的寄存器。

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

位	位名	初始值	R/W	说明
15 ~ 8	HDCN[7:0]	01001111	R	水平显示字符数 设定水平显示的字符数 (字符单位: 8 个点单位)。 必须设定 (显示字符数 - 1) 的值。 (例) 使用宽 640pixel 的液晶模块时 HDCN=(640/8)-1=79=H'4F
7 ~ 0	HTCN[7:0]	01010010	R/W	水平总字符数 设定水平的总字符数 (字符单位: 8 个点)。 必须设定 (总字符数 - 1) 的值。 但是, 最小的水平回扫周期是 3 个字符 (24 个点)。 (例) 使用宽 640pixel 的液晶模块时 HTCN=[(640/8)-1]+3=82=H'52 此时, 水平的总点数为 664 点, 水平回扫周期为 24 点。

- 【注】
- HDCN 和 HTCN 的设定值必须满足 $HTCN \geq HDCN$ 的关系。HTCN 的总字符数必须为偶数 (由于设定减 1 的值, 所以设定值为奇数)。
 - 必须根据显示器的分辨率进行以下 HDCN 的设定。
1bpp 时: 16 的倍数 - 1 [1 行为 128pixel 的倍数]
2bpp 时: 8 的倍数 - 1 [1 行为 64pixel 的倍数]
4bpp 时: 4 的倍数 - 1 [1 行为 32pixel 的倍数]
6bpp/8bpp 时: 2 的倍数 - 1 [1 行为 16pixel 的倍数]

22.3.10 LCDC 水平同步信号寄存器 (LDHSYNR)

LDHSYNR 是指定液晶显示屏模块水平方向（扫描方向）的同步信号时序的寄存器。

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

位	位名	初始值	R/W	说明
15 ~ 12	HSYNW[3:0]	全 0	R/W	水平同步信号宽度 设定水平同步信号（CL1、Hsync）的宽度（字符单位：8 个点）。 必须设定（水平同步信号宽度 -1）的值。 （例）将水平同步信号宽度设定为 8 个点时 HSYNW=(8 个点 / 8 个点 / 字符) - 1 = 0 = H'0
11 ~ 8	—	全 0	R	保留位 读取值为 0，只能写 0。
7 ~ 0	HSYNP[7:0]	0101000 0	R/W	水平同步信号输出位置 设定水平同步信号的输出位置（字符单位：8 个点单位）。 必须设定（水平同步信号输出位置 -1）的值。 （例）使用宽 640pixel 的液晶模块时 HSYNP=[(640/8)+1]-1=80=H'50 此时，从第 648 点到第 655 点，水平同步信号有效。

【注】 必须满足 $HTCN \geq HSYNP + HSYNW + 1$ 和 $HSYNP \geq HDCN + 1$ 的关系。

22.3.11 LCDC 垂直显示行数寄存器 (LDVDLNR)

LDVDLNR 是指定液晶显示屏模块垂直方向（与扫描方向垂直的方向）尺寸的寄存器。在 DSTN 时，与上下显示屏的大小无关，必须至少指定显示屏模块的垂直向向尺寸的偶数（例显示屏为 640×480 时，为 480）。

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

位	位名	初始值	R/W	说明
15 ~ 11	—	全 0	R	保留位 读取值为 0，只能写 0。
10 ~ 0	VDLN[10:0]	0011101111 1	R/W	垂直显示行数 设定垂直方向的显示行数（以行为单位）。 必须设定（显示行数 -1）的值。 （例）使用 480 行的液晶模块时 VDLN=480-1=479=H'1DF

22.3.12 LCDC 垂直总行数寄存器 (LDVTLNR)

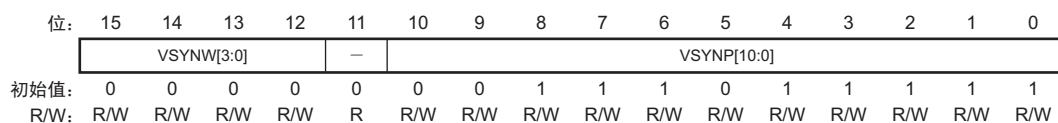
LDVTLNR 是指定包括液晶显示屏模块的垂直回扫周期的全体垂直方向长度的寄存器。

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

位	位名	初始值	R/W	说明
15 ~ 11	—	全 0	R	保留位 读取值为 0，只能写 0。
10 ~ 0	VTLN[10:0]	0011101111 1	R/W	垂直总行数 设定垂直方向的总行数（以行为单位）。 必须设定（总行数 - 1）的值。 最小的垂直总行数为 2 行。 必须满足 $VTLN \geq VDLN$ 和 $VTLN \geq 1$ 。 (例) 使用 480 行的液晶模块并且垂直回扫周期为 0 行时 $VTLN = (480 + 0) - 1 = 479 = H'1DF$

22.3.13 LCDC 垂直同步信号寄存器 (LDVSYNR)

LDVSYNR 是指定液晶模块垂直方向（与扫描方向垂直的方向）的同步信号时序的寄存器。



位	位名	初始值	R/W	说明
15 ~ 12	VSYNW[3:0]	全 0	R/W	垂直同步信号宽度 设定垂直同步信号 (FLM、Vsync) 的宽度 (以行为单位)。 必须设定 (垂直同步信号宽度 -1) 的值。 (例) 将垂直同步信号宽度设定为 1 行时 $VSYNW=(1-1)=0=H'0$
11	—	0	R	保留位 读取值为 0, 只能写 0。
10 ~ 0	VSYNP[10:0]	001110111 11	R/W	垂直同步信号输出位置 设定垂直同步信号 (FLM、Vsync) 的输出位置 (以行为单位)。 必须设定 (垂直同步信号输出位置 -2) 的值。 在 DSTN 时, 必须设定奇数值, 以 (设定值 +1)/2 进行处理。 (例) 使用 480 行的液晶模块并且回扫周期为 0 行 (VTLN=479) 时, 在第 1 行垂直同步信号有效的情况 <ul style="list-style-type: none"> • 单显示时 $VSYNP=[(1-1)+VTLN] \bmod (VTLN+1)=[(1-1)+479 \bmod (479+1)]$ $=479 \bmod 480=479$ $=H'1DF$ • 双显示时 $VSYNP=[(1-1) \times 2 + VTLN] \bmod (VTLN+1)=[(1-1) \times 2 + 479] \bmod (479+1)$ $=479 \bmod 480=479$ $=H'1DF$

22.3.14 LCDC AC 调制信号交替行数寄存器 (LDACLNR)

LDACLNR 是指定液晶模块的 AC 调制信号（液晶交流信号）交替时序的寄存器。

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

位	位名	初始值	R/W	说明
15 ~ 5	—	全 0	R	保留位 读取值为 0，只能写 0。
4 ~ 0	ACLN[4:0]	01100	R/W	AC 行数 设定液晶模块交流信号的交替行数（以行为单位）。 必须设定（交替行数-1）的值。 （例）每 13 行进行 1 次交替时 ACLN=13-1=12=H'0C

【注】 在显示屏的总行数为偶数时，为了在奇数行进行交替，必须设定偶数。

22.3.15 LCDC 中断控制寄存器 (LDINTR)

LDINTR 是指定 Vsync 中断起始点的寄存器。有关中断请参照“22.3.19 LCDC 用户指定中断控制寄存器 (LDUINTR)”和“22.3.20 LCDC 用户指定中断行数寄存器 (LDUINTLNR)”。另外，此寄存器设定的作用和 LCDC 用户指定中断控制寄存器 (LDUINTR) 设定的作用是相互独立的。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MINT EN	FINT EN	VSINT EN	VEINT EN	MINTS	FINTS	VSINTS	VEINTS	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15	MINTEN	0	R/W	存储器存取中断允许位 设定是否在 VRAM 存取（通过 LCDC 进行的 VRAM 存取）的垂直回扫周期的起始点产生中断 0：在 VRAM 存取的垂直回扫周期的起始点不产生中断。 1：在 VRAM 存取的垂直回扫周期的起始点产生中断。
14	FINTEN	0	R/W	帧结束中断允许位 设定是否在帧的最后像素输出到液晶显示屏时产生中断。 0：在帧的最后像素输出到液晶显示屏时，不产生中断。 1：在帧的最后像素输出到液晶显示屏时，产生中断。
13	VSINTEN	0	R/W	Vsync 开始中断允许位 设定是否在 LCDC Vsync 开始时产生中断。 0：在 LCDC Vsync 开始时，不产生中断。 1：在 LCDC Vsync 开始时，产生中断。

位	位名	初始值	R/W	说明
12	VEINTEN	0	R/W	Vsync 结束中断允许位 设定是否在 LCDC Vsync 结束时产生中断。 0: 在 LCDC Vsync 结束时, 不产生中断。 1: 在 LCDC Vsync 结束时, 产生中断。
11	MINTS	0	R/W	存储器存取中断状态位 表示存储器存取中断的处理状态。 在发生 LCDC 存储器存取中断时, 此位为 1 (置位状态)。在存储器存取中断的处理过程中, 必须通过给寄存器写 0 进行清除。 0: 表示 LCDC 未产生存储器存取中断或者接受到处理完存储器存取中断的通知。 1: 表示 LCDC 产生了存储器存取中断, 但是还未接受到处理完的通知。
10	FINTS	0	R/W	帧结束中断状态位 表示帧结束中断的处理状态。 在发生 LCDC 帧结束中断时, 此位为 1 (置位状态)。在帧结束中断的处理过程中, 必须通过给寄存器写 0 进行清除。 0: 表示 LCDC 未产生帧结束中断或者对于产生的帧退出中断接受到处理完帧结束中断的通知。 1: 表示 LCDC 产生了帧结束中断, 但是还未接受到处理完的通知。
9	VSINTS	0	R/W	Vsync 开始中断状态位 表示 LCDC Vsync 开始中断的处理状态。 在发生 LCDC Vsync 开始中断时, 此位为 1 (置位状态)。在 Vsync 开始中断的处理过程中, 必须通过给寄存器写 0 进行清除。 0: 表示 LCDC 未产生 Vsync 开始中断或者接受到处理完 Vsync 开始中断的通知。 1: 表示 LCDC 产生了 Vsync 开始中断, 但是还未接受到处理完的通知。
8	VEINTS	0	R/W	Vsync 结束中断状态位 表示 LCDC Vsync 结束中断的处理状态。 在发生 LCDC Vsync 结束中断时, 此位为 1 (置位状态)。在 Vsync 结束中断的处理过程中, 必须通过给寄存器写 0 进行清除。 0: 表示 LCDC 未产生 Vsync 结束中断或者接受到处理完 Vsync 结束中断的通知。 1: 表示 LCDC 产生了 Vsync 结束中断, 但是还未接受到处理完的通知。
7 ~ 0	—	全 0	R	保留位 读取值为 0, 只能写 0。

22.3.16 LCDC 电源管理模式寄存器 (LDPMMR)

LDPMMR 是设定电源电路（提供给液晶显示屏的电源电路）控制功能的寄存器，此寄存器设定是否使用 LCD_VCPWC 和 LCD_VEPWC 的 2 种电源控制引脚，并且设定电源控制功能的接通和切断。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ONC[3:0]				OFFD[3:0]				—	VCPE	VEPE	DONE	—	—	LPS[1:0]	
初始值:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R

位	位名	初始值	R/W	说明
15 ~ 12	ONC[3:0]	全 0	R/W	LCDC 电源接通顺序周期 在 LCD 模块的电源接通顺序中，以帧周期为单位，设定从 LCD_VEPWC 引脚有效到 LCD_DON 引脚有效的周期。必须设定（周期 - 1）的值。 相当于图 22.4 ~ 22.7 “电源控制顺序和液晶模块的运行状态”的 (c) 周期。详细方法请参照表 22.5（ONA、ONB、OFFD、OFFE 和 OFFF 各寄存器的设定方法是相同的）。
11 ~ 8	OFFD[3:0]	全 0	R/W	LCDC 电源切断顺序周期 在 LCD 模块的电源切断顺序中，以帧周期为单位，设定从 LCD_DON 引脚无效到 LCD_VEPWC 引脚无效的周期。必须设定（周期 - 1）的值。 相当于图 22.4 ~ 图 22.7 “电源控制顺序和液晶模块的运行状态”的 (d) 周期。
7	—	0	R	保留位 读取值为 0，只能写 0。
6	VCPE	0	R/W	LCD_VCPWC 引脚允许位 设定是否使用 LCD_VCPWC 引脚进行电源控制顺序的处理。 0:（无处理）LCD_VCPWC 引脚的输出被屏蔽，固定为低电平。 1:（有处理）按照指定的顺序进行 LCD_VCPWC 引脚的输出控制（有效 / 无效）。
5	VEPE	0	R/W	LCD_VEPWC 引脚允许位 设定是否使用 LCD_VEPWC 引脚进行电源控制顺序的处理。 0:（无处理）LCD_VEPWC 引脚的输出被屏蔽，固定为低电平。 1:（有处理）按照指定的顺序进行 LCD_VEPWC 引脚的输出控制（有效 / 无效）。
4	DONE	1	R/W	LCD_DON 引脚允许位 设定是否使用 LCD_DON 引脚进行电源控制顺序处理的有（无）进行设定。 0:（无处理）LCD_DON 引脚的输出被屏蔽，固定为低电平。 1:（有处理）按照指定的顺序进行 LCD_DON 引脚的输出控制（有效 / 无效）。
3、2	—	全 0	R	保留位 读取值为 0，只能写 0。
1、0	LPS[1:0]	全 0	R	液晶模块电源输入状态位 表示在使用电源控制功能时液晶模块的上电状态。 00: 切断液晶模块的电源 11: 接通液晶模块的电源

22.3.17 LCDC 电源顺序周期寄存器 (LDPSPR)

LDPSPR 是设定电源电路（提供给液晶模块的电源电路）控制功能的寄存器。指定 LCD_VEPWC 和 LCD_VCPWC 引脚及其时序信号的输出开始时序。

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

位	位名	初始值	R/W	说明
15 ~ 12	ONA[3:0]	1111	R/W	LCDC 电源接通顺序周期 在 LCD 模块的电源接通顺序中，以帧周期为单位，设定从 LCD_VCPWC 引脚有效到开始输出显示数据 (LCD_D15 ~ 0) 和时序信号 (LCD_FLM、LCD_CL1、LCD_CL2 和 LCD_M_DISP) 的周期。必须设定 (周期 -1) 的值。 相当于图 22.4 ~ 图 22.7 “ 电源控制顺序和液晶模块的运行状态” 的 (a) 周期。
11 ~ 8	ONB[3:0]	0110	R/W	LCDC 电源接通顺序周期 在 LCD 模块的电源接通顺序中，以帧周期为单位，设定从开始输出显示数据 (LCD_D15 ~ 0) 和时序信号 (LCD_FLM、LCD_CL1、LCD_CL2 和 LCD_M_DISP) 到 LCD_VEPWC 引脚有效的周期。必须设定 (周期 -1) 的值。 相当于图 22.4 ~ 图 22.7 “ 电源控制顺序和液晶模块的运行状态” 的 (b) 周期。
7 ~ 4	OFFE[3:0]	0000	R/W	LCDC 电源切断顺序周期 在液晶模块的电源切断顺序中，以帧周期为单位，设定从 LCD_VEPWC 引脚无效到停止输出显示数据 (LCD_D15 ~ 0) 和时序信号 (LCD_FLM、LCD_CL1、LCD_CL2 和 LCD_M_DISP) 的周期。必须设定 (周期 -1) 的值。 相当于图 22.4 ~ 图 22.7 “ 电源控制顺序和液晶模块的运行状态” 的 (e) 周期。
3 ~ 0	OFFF[3:0]	1111	R/W	LCDC 电源切断顺序周期 在液晶模块的电源切断顺序中，以帧周期为单位，设定从停止输出显示数据 (LCD_D15 ~ 0) 和时序信号 (LCD_FLM、LCD_CL1、LCD_CL2 和 LCD_M_DISP) 到 LCD_VCPWC 引脚无效的周期。必须设定 (周期 -1) 的值。 相当于图 22.4 ~ 图 22.7 “ 电源控制顺序和液晶模块的运行状态” 的 (f) 周期。

22.3.18 LCDC 控制寄存器 (LDCNTR)

LDCNTR 指定是开始还是结束 LCDC 显示运行。

在将 DON2 位和 DON 位都置 1 时，LCDC 开始显示。然后，根据 LDPMMR 和 LDCNTR 设定的顺序接通液晶模块的电源。如果 LPS1、LPS0 位从 B'00 变为 B'11，就结束所定的顺序。不能在所定的顺序结束前操作 DON 位。

在结束 LCDC 显示运行时，将 DON 位置 0。根据 LDPMMR 和 LDCNTR 设定的顺序切断液晶模块的电源，并且停止 LCDC 运行。如果 LPS[1:0] 从 B'11 变为 B'00，就结束所定的顺序。不能在所定的顺序结束前操作 DON 位。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	DON2	—	—	—	DON
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

位	位名	初始值	R/W	说明
15 ~ 5	—	全 0	R	保留位 读取值为 0，只能写 0。
4	DON2	0	R/W	显示开始 2 指示开始 LCDC 显示运行。 0: LCDC 运行 / 结束中 1: 运行开始 此位的读取值为 0。只在开始显示运行时才能将此位置 1。如果不在开始显示运行时将此位置 1，就不保证运行。因为置 1 后自动恢复为 0，所以不需要清 0。
3 ~ 1	—	全 0	R	保留位 读取值为 0，只能写 0。
0	DON	0	R/W	显示开始 指示是开始还是结束 LCDC 显示运行。 能通过参照 LDPMMR 的 LPS[1:0] 的值进行控制顺序的状态确认。 0: LCDC 不运行。非显示模式 1: LCDC 运行。显示模式

- 【注】**
1. 在开始显示时，必须将 LDCNTR 置 H'0011；在结束显示时，必须将 LDCNTR 置 H'0000。不能置其他值。
 2. 如果将 DON2 位置 1，配色 RAM 的数据就为不定值，所以必须在将 DON2 位置 1 后给配色 RAM 设定数据。

22.3.19 LCDC 用户指定中断控制寄存器 (LDUINTR)

LDUINTR 设定是否产生用户指定中断，并且表示其状态。在 LCDC 从 VRAM 读取 LCDC 用户指定中断行数寄存器 (LDUINLNR) 设定的行图像数据时，产生该中断。

LCDC 发行的中断 (LCDCI) 是此寄存器指定的用户指定中断、LCDC 中断控制寄存器 (LDINTR) 指定的存储器存取和 Vsync 中断的 OR 输出。另外，此寄存器和 LCDC 中断控制寄存器 (LDINTR) 的设定分别影响中断的运行。

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

位	位名	初始值	R/W	说明
15 ~ 9	—	全 0	R	保留位 读取值为 0，只能写 0。
8	UINTEN	0	R/W	用户指定中断允许位 设定是否产生 LCDC 用户指定中断。 0：不产生 LCDC 用户指定中断。 1：产生 LCDC 用户指定中断。
7 ~ 1	—	全 0	R	保留位 读取值为 0，只能写 0。
0	UINTS	0	R/W	用户指定中断状态位 在发生 LCDC 用户指定中断时，此位为 1（置位状态）。在用户指定中断的处理过程中，必须通过给寄存器写 0 清除此位。 0：表示 LCDC 未产生用户指定中断或者接受到处理完用户指定中断的通知。 1：表示 LCDC 产生了用户指定中断，但是还未接受到处理完的通知。

【注】 中断的处理流程：

1. 输入中断信号。
2. 读 LDINTR。
3. 如果 MINTS、FINTS、VSINTS 或者 VEINTS=1，该中断就为存储器存取中断、帧结束中断、Vsync 上升中断或者 Vsync 下降中断。针对不同的中断进行处理。
4. 如果 MINTS=FINTS=VSINTS=VEINTS=0，该中断就不是存储器存取中断、帧结束中断、Vsync 上升中断和 Vsync 下降中断。
5. 读 UINTS。
6. 如果 UINTS=1，该中断为用户指定中断。进行用户指定中断的处理。
7. 如果 UINTS=0，该中断不是用户指定中断。进行其它的处理。

22.3.20 LCDC 用户指定中断行数寄存器 (LDUINTLNR)

LDUINTLNR 是设定用户指定中断的发生位置的寄存器。能以水平行为单位进行设定。

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

位	位名	初始值	R/W	说明
15 ~ 11	—	全 0	R	保留位 读取值为 0，只能写 0。
10 ~ 0	UINTLN[10:0]	00001001111	R/W	用户指定中断的发生行数 设定用户指定中断的发生行（以行为单位）。 必须设定（中断发生行数 -1）的值。 （例）在第 80 行发生用户指定中断时： HINTLN=160/2-1=79=H'04F

- 【注】
1. 在使用 STN/TFT 液晶模块时，此寄存器的设定值不能超过 LDVDLNR 垂直显示行数（VDLN）。
 2. 在使用 DSTN 液晶模块时，此寄存器的设定值不能超过 LDVDLNR 垂直显示行数（VDLN）的 1/2。当 LCDC 从 VRAM 读取此寄存器所设行数的下部画面图像数据时，产生用户指定中断。

22.3.21 LCDC 存储器存取间隔数寄存器 (LDLIRNR)

LDLIRNR 是控制 LCDC 读 VRAM 时的总线周期间隔的寄存器。如果给 LDLIRNR 设定 H'00 以外的值，就对 SDRAM 的时钟数进行计数，在计数值与 LDLIRNR 的设定值相同前，LCDC 不存取 VRAM。如果给 LDLIRNR 设定 H'00（初始值），LCDC 就在 1 个 VRAM 存取时钟后存取 VRAM。

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

位	位名	初始值	R/W	说明
15 ~ 8	—	全 0	R	保留位 读取值为 0，只能写 0。
7 ~ 0	LIRN[7:0]	全 0	R/W	LCDC 的 VRAM 读时钟周期间隔 在 LCDC 读 VRAM 的突发总线周期期间，指定 SDRAM 的时钟周期数。 H'00: 1 个时钟周期 H'01: 1 个时钟周期 H'02: 2 个时钟周期 : H'FE: 254 个时钟周期 H'FF: 255 个时钟周期

22.4 运行说明

22.4.1 LCDC 能显示的液晶模块的尺寸

LCDC 能进行 1024×1024 点、16bpp（位每像素）的显示，但是显示的图像保存在与 CPU 共用的 VRAM 中，所以需要在显示前从 VRAM 读数据。

本 LSI 具有最大 16 个突发（32 位宽度）的存储器读操作和 2.4K 字节的行缓冲器，一般很少发生显示问题，但是根据组合的不同，可能会发生一些显示问题。60Hz 帧频时的推荐尺寸为：16bpp 时 320×240 点，8bpp 时 640×480 点。

基准是总线占有率不能超过 40%。

$$\text{总线占有率 (\%)} = \frac{\text{开销系数} \times \text{显示总像素数}((\text{HDCN} + 1) \times 8 \times (\text{VDLN} + 1)) \times \text{帧速率(Hz)} \times \text{色数(bpp)}}{\text{CLKOUT(Hz)} \times \text{总线宽(=32bit)}} \times 100$$

在 CL（CAS 等待时间）=2 的 SDRAM 为 32 位数据总线时，开销系数为 2.000（在 64 位数据总线时为 1.825，无论哪个值都是最佳情况下的理想值）。

有效显示和回扫周期如图 22.2 所示。

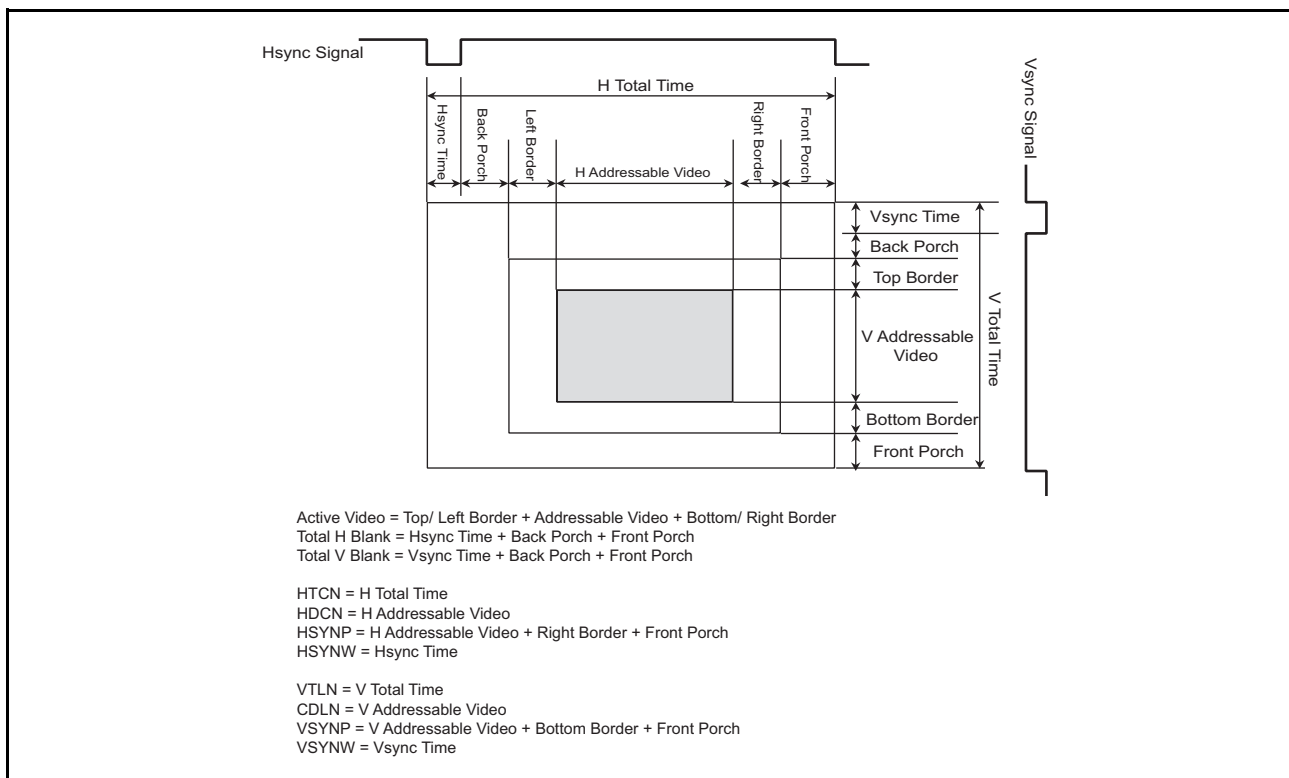


图 22.2 有效显示和回扫周期

22.4.2 彩色调色板规格

(1) 彩色配色寄存器

LCDC 具有每入口输出 24 位数据并且能同时使用 256 个入口的彩色调色板。通过使用该彩色调色板，能同时显示 16M 颜色中的 256 色。

用户能通过以下的步骤随时设定彩色调色板。

1. LDPALCR 的 PALEN 位=0 (初始值)：正常显示运行
2. 存取 LDPALCR，设定 PALEN=1：在 3 个外围时钟周期后，转移到彩色调色板设定模式。
3. 存取 LDPALCR，确认 PALS=1。
4. 存取 LDPR00 ~ FF，给 PALD00 ~ FF 写必要值。
5. 存取 LDPALCR，设定 PALEN=0：在 1 个外围时钟周期后，转移到正常显示模式。

在 LDPALCR 的 PALS=1 期间，LCDC 显示数据的输出引脚 (LCD_D15 ~ 0) 输出 0。

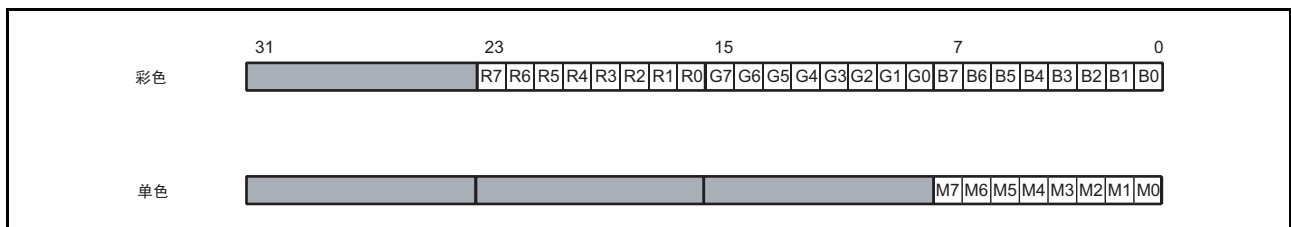


图 22.3 彩色配色数据格式

必须按上图的格式设定 PALDnn 的颜色 / 灰度数据。

在彩色显示时，给 PALDnn[23:16]、PALDnn[15:8] 和 PALDnn[7:0] 分别设定 R 数据、G 数据和 B 数据。对于 PALDnn[18:16]、PALDnn[9:8]、PALDnn[2:0]，虽然存在寄存器的位，但是不存在与其对应的存储器。所以 PALDnn[18:16]、PALDnn[9:8]、PALDnn[2:0] 不能保存调色板的数据，R: 5 位、G: 6 位、B: 5 位有效。在实际使用时，必须写 24 位 (R: 8 位、G: 8 位、B: 8 位) 的数据。当 PALDnn[23:19]、PALDnn[15:10]、PALDnn[7:3] 的值不为 0 时，将 PALDnn[18:16]、PALDnn[9:8]、PALDnn[2:0] 置 0 或者 1；当 PALDnn[23:19]、PALDnn[15:10]、PALDnn[7:3] 的值为 0 时，将 PALDnn[18:16]、PALDnn[9:8]、PALDnn[2:0] 置 0。扩展为 24 位。

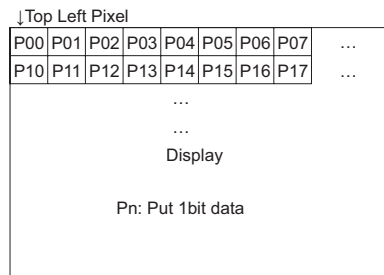
在单色显示时，给 PALDnn[7:3] 设定灰度数据，PALDnn[23:8] 为 don't care。当 PALDnn[7:3] 的值不为 0 时，将 PALDnn[2:0] 置 1；当 PALDnn[7:3] 值为 0 时，将 PALDnn[2:0] 置 0。扩展为 8 位。


【符号说明】 nn: H'00 ~ H'FF

22.4.3 数据格式

1.Packed 1bpp (Pixel Alignment in Byte is Big Endian) [Windows CE Recommended Format]

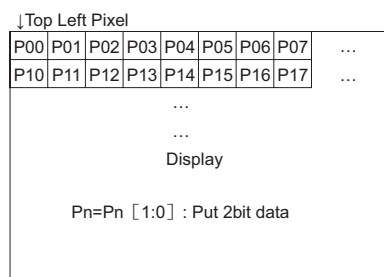
Address	MSB							LSB	[Bit]
	7	6	5	4	3	2	1	0	
+00	P00	P01	P02	P03	P04	P05	P06	P07	(Byte0)
+01	P08								(Byte1)
+02	...								
+LAO+00	P10	P11	P12	P13	P14	P15	P16	P17	
+LAO+01	P18								
+LAO+02	...								
+LAO+03	Display Memory								




LAO: Line Address Offset
 -Unused bits should be 0

2.Packed 2bpp (Pixel Alignment in Byte is Big Endian) [Windows CE Recommended Format]

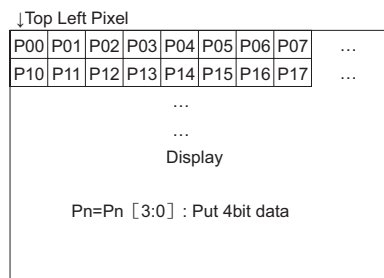
Address	MSB				LSB				[Bit]
	7	6	5	4	3	2	1	0	
+00	P00		P01		P02		P03		(Byte0)
+01	P04		P05		P06		P07		(Byte1)
+02	...								
+LAO+00	P10		P11		P12		P13		
+LAO+01	P14		P15		P16		P17		
+LAO+02	...								
+LAO+03	Display Memory								




LAO: Line Address Offset
 -Unused bits should be 0

3.Packed 4bpp (Pixel Alignment in Byte is Big Endian) [Windows CE Recommended Format]

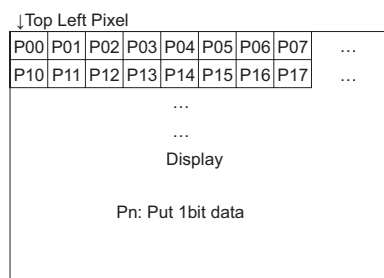
Address	MSB				LSB				[Bit]
	7	6	5	4	3	2	1	0	
+00	P00				P01				(Byte0)
+01	P02				P03				(Byte1)
+02	P04				P05				(Byte2)
+03	...								
+LAO+00	P10				P11				
+LAO+01	P12				P13				
+LAO+02	P14				P15				
+LAO+03	...								
...	Display Memory								




LAO: Line Address Offset
 -Unused bits should be 0

4.Packed 1bpp (Pixel Alignment in Byte is Little Endian)

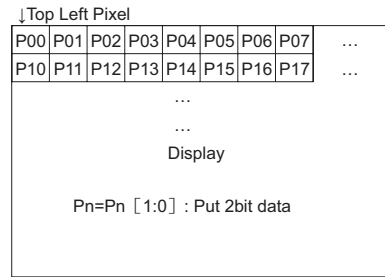
Address	MSB							LSB	[Bit]
	7	6	5	4	3	2	1	0	
+00	P07	P06	P05	P04	P03	P02	P01	P00	(Byte0)
+01								P08	(Byte1)
+02	...								
+LAO+00	P17	P16	P15	P14	P13	P12	P11	P10	
+LAO+01								P18	
+LAO+02	...								
+LAO+03	Display Memory								



LAO: Line Address Offset
 -Unused bits should be 0

5.Packed 2bpp (Pixel Alignment in Byte is Little Endian)

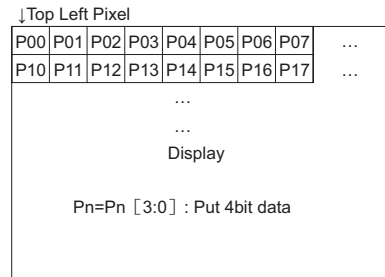
Address	MSB				LSB				[Bit]
	7	6	5	4	3	2	1	0	
+00	P03		P02		P01		P00		(Byte0)
+01	P07		P06		P05		P04		(Byte1)
+02	...								
+03	...								
...	...								
+LAO+00	P13		P12		P11		P10		
+LAO+01	P17		P16		P15		P14		
+LAO+02	...								
+LAO+03	...								
...	Display Memory								



LAO: Line Address Offset
 -Unused bits should be 0

6.Packed 4bpp (Pixel Alignment in Byte is Little Endian)

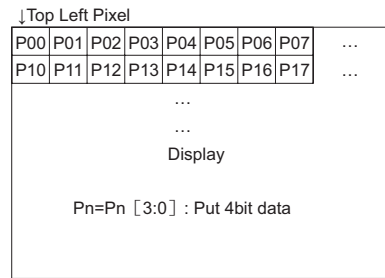
Address	MSB				LSB				[Bit]
	7	6	5	4	3	2	1	0	
+00	P01				P00				(Byte0)
+01	P03				P02				(Byte1)
+02	P05				P04				(Byte2)
+03	...								
...	...								
+LAO+00	P11				P10				
+LAO+01	P13				P12				
+LAO+02	P15				P14				
+LAO+03	...								
...	Display Memory								



LAO: Line Address Offset
 -Unused bits should be 0

7.Unpacked 4bpp [Windows CE Recommended Format]

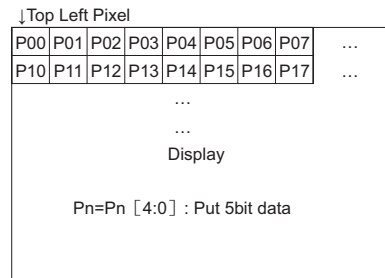
Address	MSB				LSB				[Bit]
	7	6	5	4	3	2	1	0	
+00					P00				(Byte0)
+01					P01				(Byte1)
+02					P02				(Byte2)
+03	...								
...	...								
+LAO+00					P10				
+LAO+01					P11				
+LAO+02					P12				
+LAO+03	...								
...	Display Memory								



LAO: Line Address Offset
 -Unused bits should be 0

8.Unpacked 5bpp [Windows CE Recommended Format]

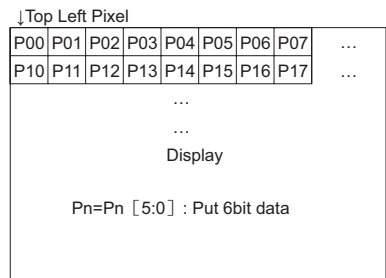
Address	MSB				LSB				[Bit]
	7	6	5	4	3	2	1	0	
+00					P00				(Byte0)
+01					P01				(Byte1)
+02					P02				(Byte2)
+03	...								
...	...								
+LAO+00					P10				
+LAO+01					P11				
+LAO+02					P12				
+LAO+03	...								
...	Display Memory								



LAO: Line Address Offset
 -Unused bits should be 0

9.Unpacked 6bpp [Windows CE Recommended Format]

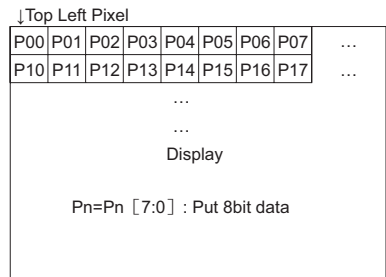
Address	MSB							LSB	[Bit]
	7	6	5	4	3	2	1	0	
+00	P00								(Byte0)
+01	P01								(Byte1)
+02	P02								(Byte2)
+03	...								
...	...								
+LAO+00	P10								
+LAO+01	P11								
+LAO+02	P12								
+LAO+03	...								
...	Display Memory								



LAO: Line Address Offset
 [] -Unused bits should be 0

10.Packed 8bpp [Windows CE Recommended Format]

Address	MSB							LSB	[Bit]
	7	6	5	4	3	2	1	0	
+00	P00								(Byte0)
+01	P01								(Byte1)
+02	P02								(Byte2)
+03	...								
...	...								
+LAO+00	P10								
+LAO+01	P11								
+LAO+02	P12								
+LAO+03	...								
...	Display Memory								



LAO: Line Address Offset
 [] -Unused bits should be 0

11.Unpacked color 15bpp (RGB 555) [Windows CE Recommended Format]

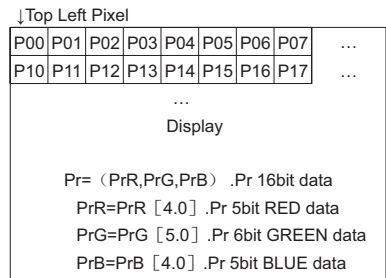
Address	MSB															LSB	[Bit]
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
+00	P00R					P00G					P00B						(Word0)
+02	P01R					P01G					P01B						(Word2)
+04	P02R					P02G					P02B						(Word4)
+06						
...						
+LAO+00	P10R					P10G					P10B						
+LAO+02	P11R					P11G					P11B						
+LAO+04	P12R					P12G					P12B						
+LAO+06						
...	Display Memory										



LAO: Line Address Offset
 [] -Unused bits should be 0

12.Packed color 16bpp (RGB 565) [Windows CE Recommended Format]

Address	MSB															LSB	[Bit]
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
+00	P00R					P00G					P00B						(Word0)
+02	P01R					P01G					P01B						(Word2)
+04	P02R					P02G					P02B						(Word4)
+06						
...						
+LAO+00	P10R					P10G					P10B						
+LAO+02	P11R					P11G					P11B						
+LAO+04	P12R					P12G					P12B						
+LAO+06						
...	Display Memory										



LAO: Line Address Offset
 [] -Unused bits should be 0

22.4.4 显示分辨率的设定

通过 LDHCNR、LDHSYNR、LDVDLNR、LDVTLNR 和 LDVSYNR 设定显示分辨率。通过 LDCNTR 设定 STN 或者 DSTN 显示时的液晶交流周期。这些寄存器的初始值为 VGA (640×480 点)，是 STN 或者 DSTN 显示时的典型的分辨率设定值。

通过 LDICKR 设定要使用的时钟。液晶模块的帧频由尺寸的相关寄存器设定的 1 个画面的显示画面 + 回扫周期 (非显示期间) 以及使用的时钟频率决定。另外，LCDC 具有 Vsync 中断功能，在每个垂直回扫周期的起始点 (正确地说是最后显示行的下一行起始点) 产生中断的。用 LDINTR 设定此功能。

22.4.5 电源控制顺序的处理

通常，液晶模块对于电源的接通和切断需要特定的顺序处理，通过设定 LDPMMR、LDPSPR 和 LDCNTR，能使用液晶电源控制引脚 (LCD_VCPWC、LCD_VEPWC 和 LCD_DON) 进行适合液晶模块需求的电源控制处理。

电源控制顺序的时序图概要如图 22.4 ~ 图 22.7 所示，能设定的电源控制顺序周期的说明如 22.5 所示。

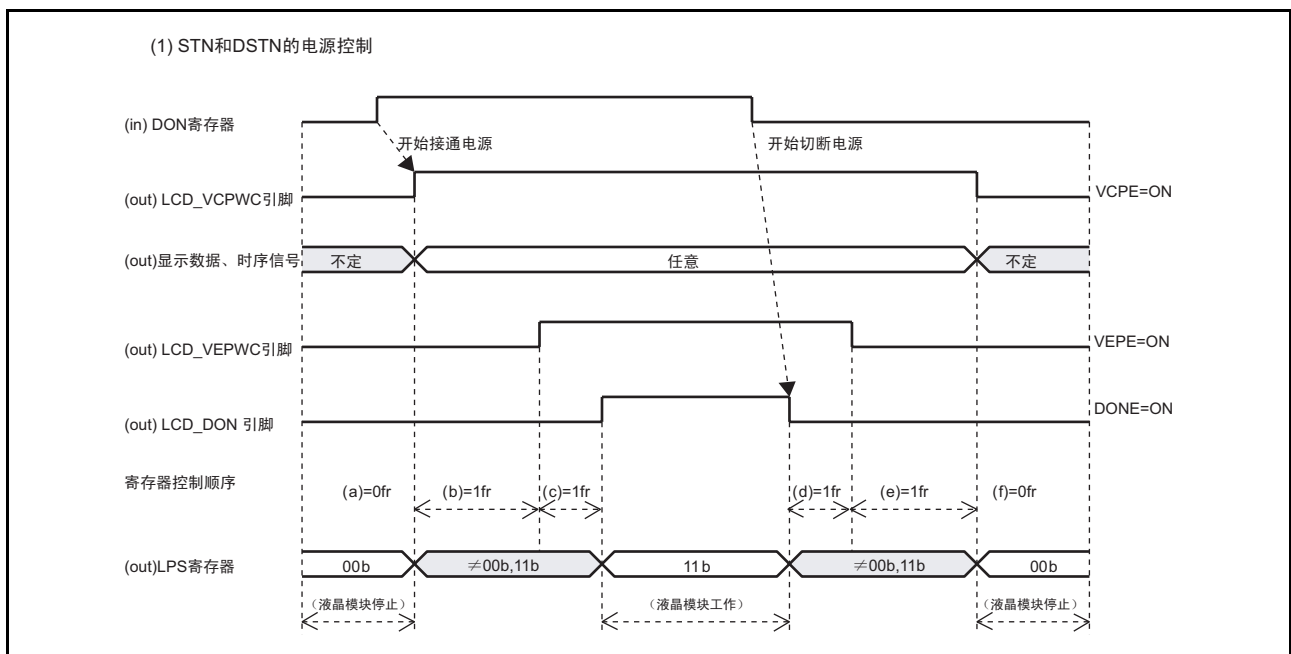


图 22.4 电源控制顺序和液晶模块的运行状态

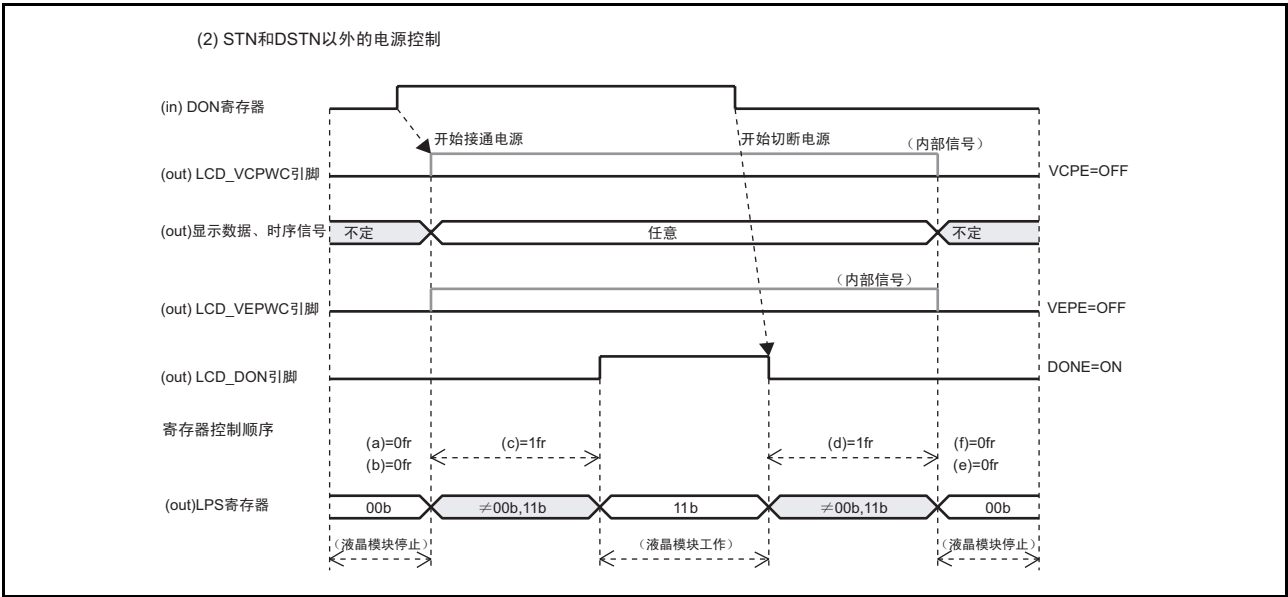


图 22.5 电源控制顺序和液晶模块的运行状态

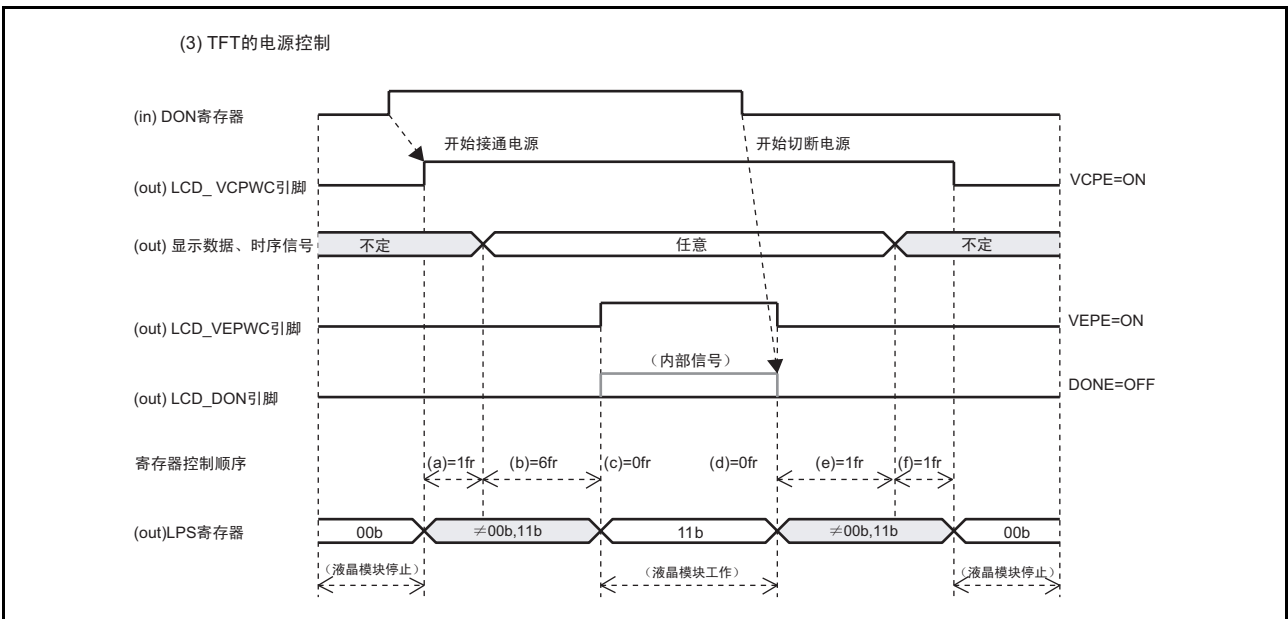


图 22.6 电源控制顺序和液晶模块的运行状态

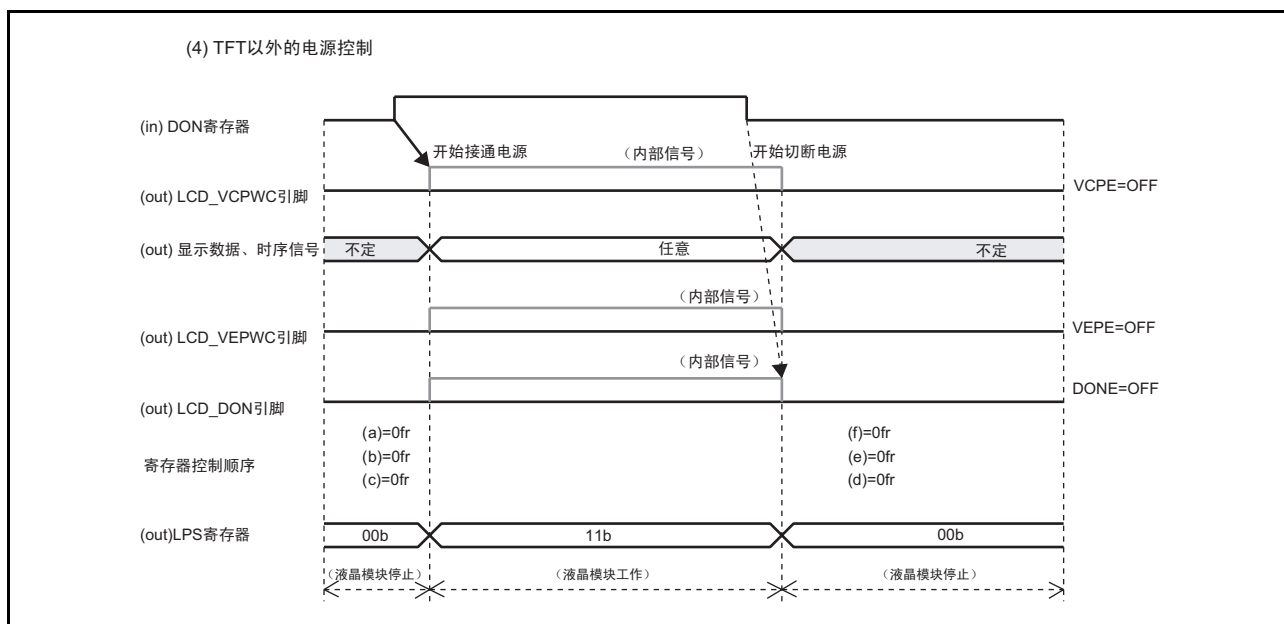


图 22.7 电源控制顺序和液晶模块的运行状态

表 22.5 对于典型的帧频能设定的电源控制顺序周期

ONX、OFFX 寄存器的设定值	帧频	
	120Hz	60Hz
H'F	$(-1+1)/120 = 0.00(\text{ms})$	$(-1+1)/60 = 0.00(\text{ms})$
H'0	$(0+1)/120 = 8.33(\text{ms})$	$(0+1)/60 = 16.67(\text{ms})$
H'1	$(1+1)/120 = 16.67(\text{ms})$	$(1+1)/60 = 33.33(\text{ms})$
H'2	$(2+1)/120 = 25.00(\text{ms})$	$(2+1)/60 = 50.00(\text{ms})$
H'3	$(3+1)/120 = 33.33(\text{ms})$	$(3+1)/60 = 66.67(\text{ms})$
H'4	$(4+1)/120 = 41.67(\text{ms})$	$(4+1)/60 = 83.33(\text{ms})$
H'5	$(5+1)/120 = 50.00(\text{ms})$	$(5+1)/60 = 100.00(\text{ms})$
H'6	$(6+1)/120 = 58.33(\text{ms})$	$(6+1)/60 = 116.67(\text{ms})$
H'7	$(7+1)/120 = 66.67(\text{ms})$	$(7+1)/60 = 133.33(\text{ms})$
H'8	$(8+1)/120 = 75.00(\text{ms})$	$(8+1)/60 = 150.00(\text{ms})$
H'9	$(9+1)/120 = 83.33(\text{ms})$	$(9+1)/60 = 166.67(\text{ms})$
H'A	$(10+1)/120 = 91.67(\text{ms})$	$(10+1)/60 = 183.33(\text{ms})$
H'B	$(11+1)/120 = 100.00(\text{ms})$	$(11+1)/60 = 200.00(\text{ms})$
H'C	$(12+1)/120 = 108.33(\text{ms})$	$(12+1)/60 = 216.67(\text{ms})$
H'D	$(13+1)/120 = 116.67(\text{ms})$	$(13+1)/60 = 233.33(\text{ms})$
H'E	$(14+1)/120 = 125.00(\text{ms})$	$(14+1)/60 = 250.00(\text{ms})$

以帧周期为单位，ONA、ONB、ONC、OFFD、OFFE 和 OFFF 的各寄存器能设定 0 ~ 15 帧的电源控制顺序周期。寄存器的 H'0 ~ H'E 设定表示 1 ~ 15 帧、H'F 设定表示 0 帧。

实际的顺序周期取决于寄存器的设定值和显示帧频率。下表是典型的液晶模块显示帧频率的电源控制顺序周期。

- 显示帧频率为 120Hz 时，给 ONB 寄存器设定 H'6
显示帧频率为 120Hz，所以 1 帧的时间为 $8.33(\text{ms})=1/120(\text{sec})$ ，电源接通顺序周期为 7 帧。此时的顺序周期为 $58.33(\text{ms})=8.33(\text{ms})\times 7$ 。

表 22.6 LCDC 运行模式

模式		功能
显示 ON (LCDC 运行)	寄存器的设定: DON=1	将所定的分辨率、由色数决定的显示数据和时序信号输出到液晶模块。
显示 OFF (LCDC 停止)	寄存器的设定: DON=0	能存取寄存器。 不将所定的分辨率、由色数决定的显示数据和时序信号输出到液晶模块。

表 22.7 液晶模块电源状态

(STN 或者 DSTN 模块时的情况)

状态	逻辑电源	显示数据、时序信号	高压系电源	DON 信号
对应的控制引脚	LCD_VCPWC	LCD_CL2、LCD_CL1、LCD_FLM、 LCD_M_DISP、LCD_D15 ~ 0	LCD_VEPWC	LCD_DON
运行状态	供给	供给	供给	供给
(过渡状态)	供给	供给	供给	
	供给	供给		
	供给			
停止状态				

(为 TFT 模块时)

状态	逻辑电源	显示数据、时序信号	高压系电源
对应的控制引脚	LCD_VCPWC	LCD_CL2、LCD_CL1、LCD_FLM、 LCD_M_DISP、LCD_D15 ~ 0	LCD_VEPWC
运行状态	供给	供给	供给
(过渡状态)	供给	供给	
	供给		
停止状态			

上述是一般情况下的液晶模块运行状态、停止状态下的电源和显示数据、时序信号的供给状态。但是，根据不同的模块，也有在模块内部从逻辑电源电压生成高压电源的情况，所以上述的电源供给未必是必须的。

显示 OFF 模式 (LCDC 停止) 的注意事项:

在通过本 LCDC 使用液晶模块电源控制顺序处理时，如果在显示 ON 状态下切断电源，就无法保证 LCDC 的正常运行。最坏的情况可能会损坏连接的液晶模块。

22.5 时钟与 LCD 数据信号例子

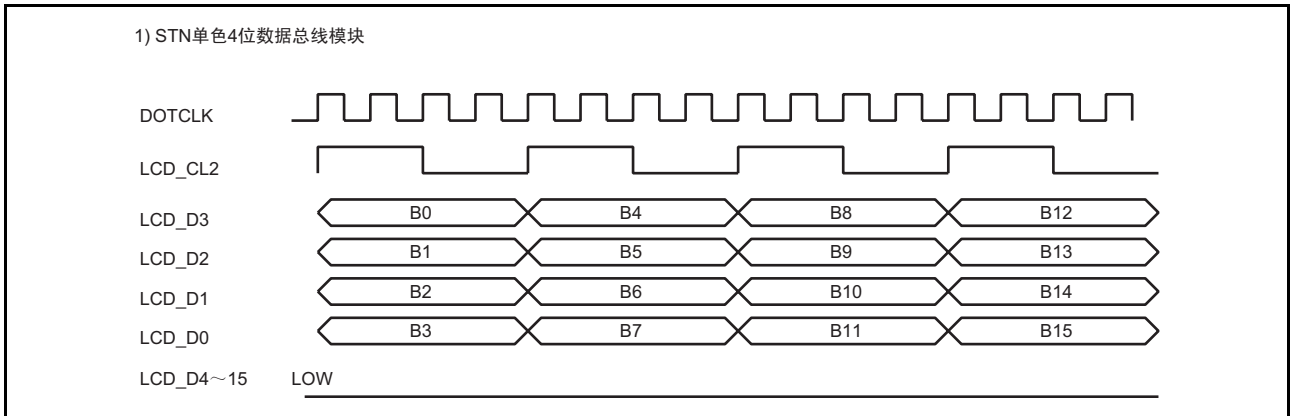


图 22.8 时钟和 LCD 数据信号例子 (STN 单色 4 位数据总线模块)

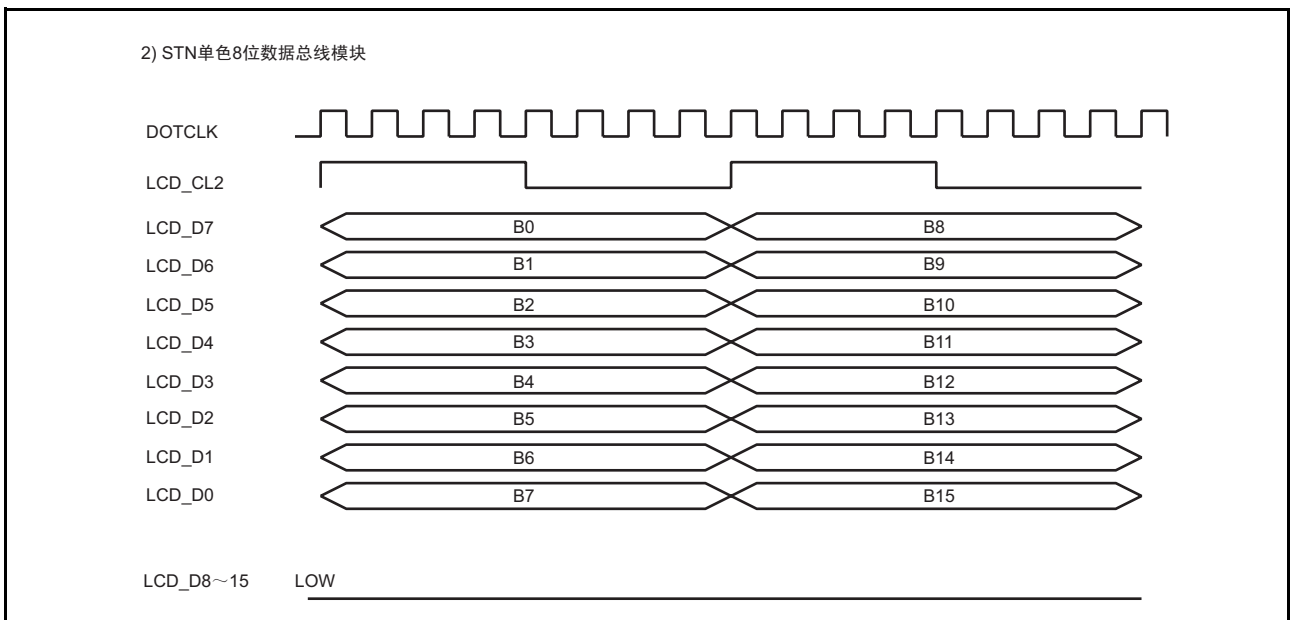


图 22.9 时钟和 LCD 数据信号例子 (STN 单色 8 位数据总线模块)

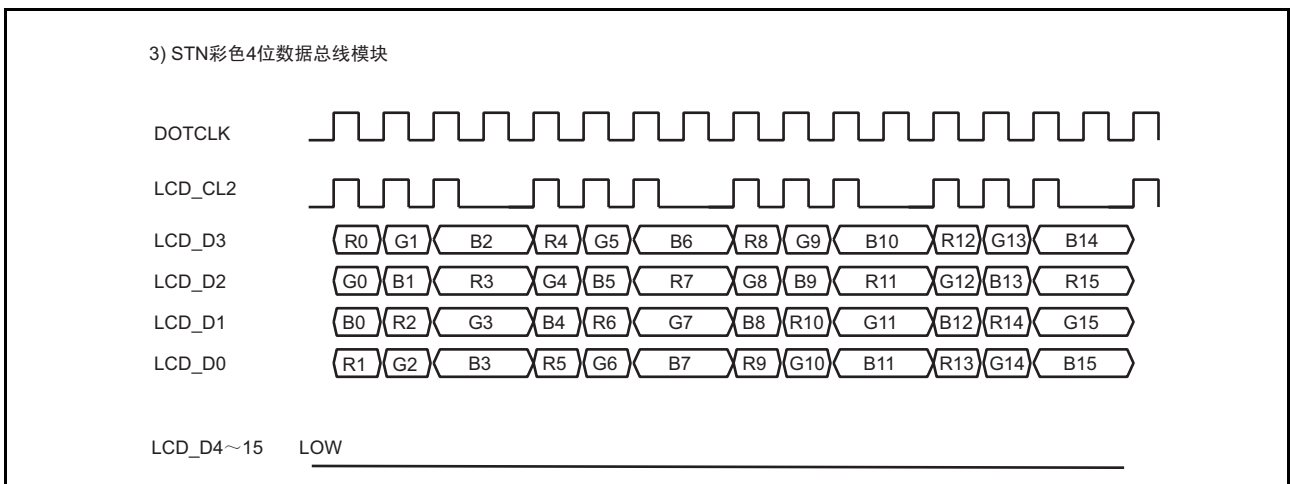


图 22.10 时钟和 LCD 数据信号例子 (STN 彩色 4 位数据总线模块)

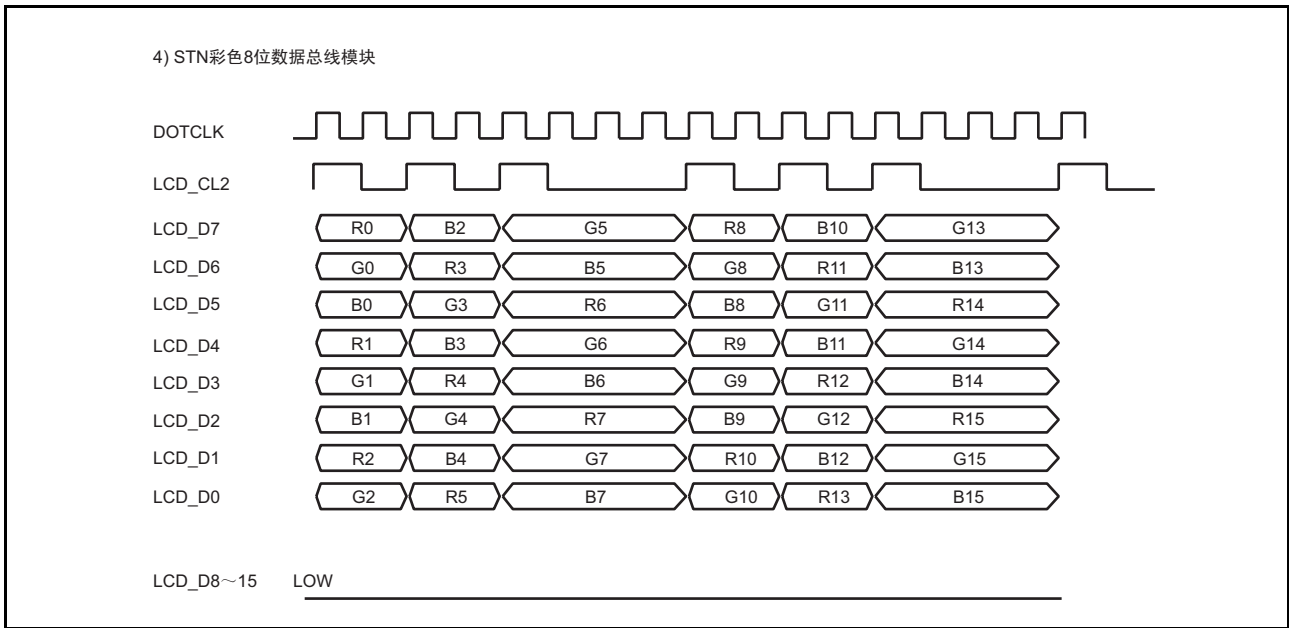


图 22.11 时钟和 LCD 数据信号例子 (STN 彩色 8 位数据总线模块)

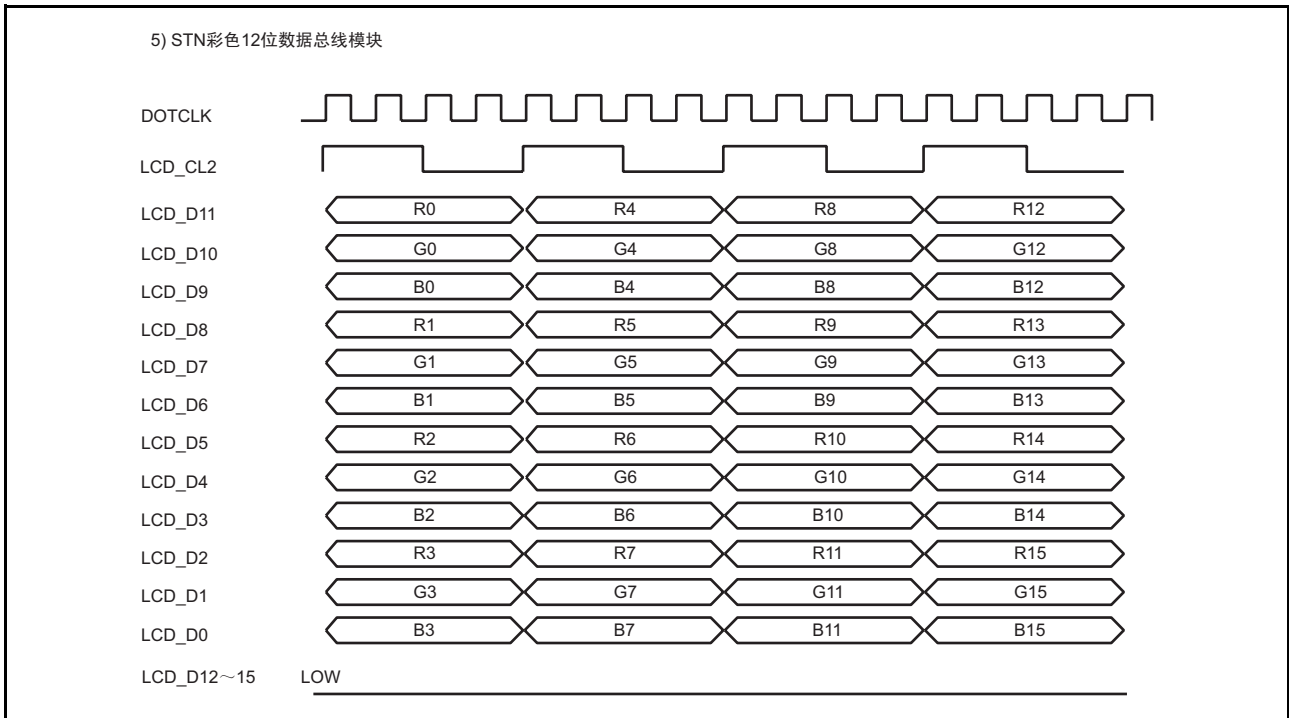


图 22.12 时钟和 LCD 数据信号例子 (STN 彩色 12 位数据总线模块)

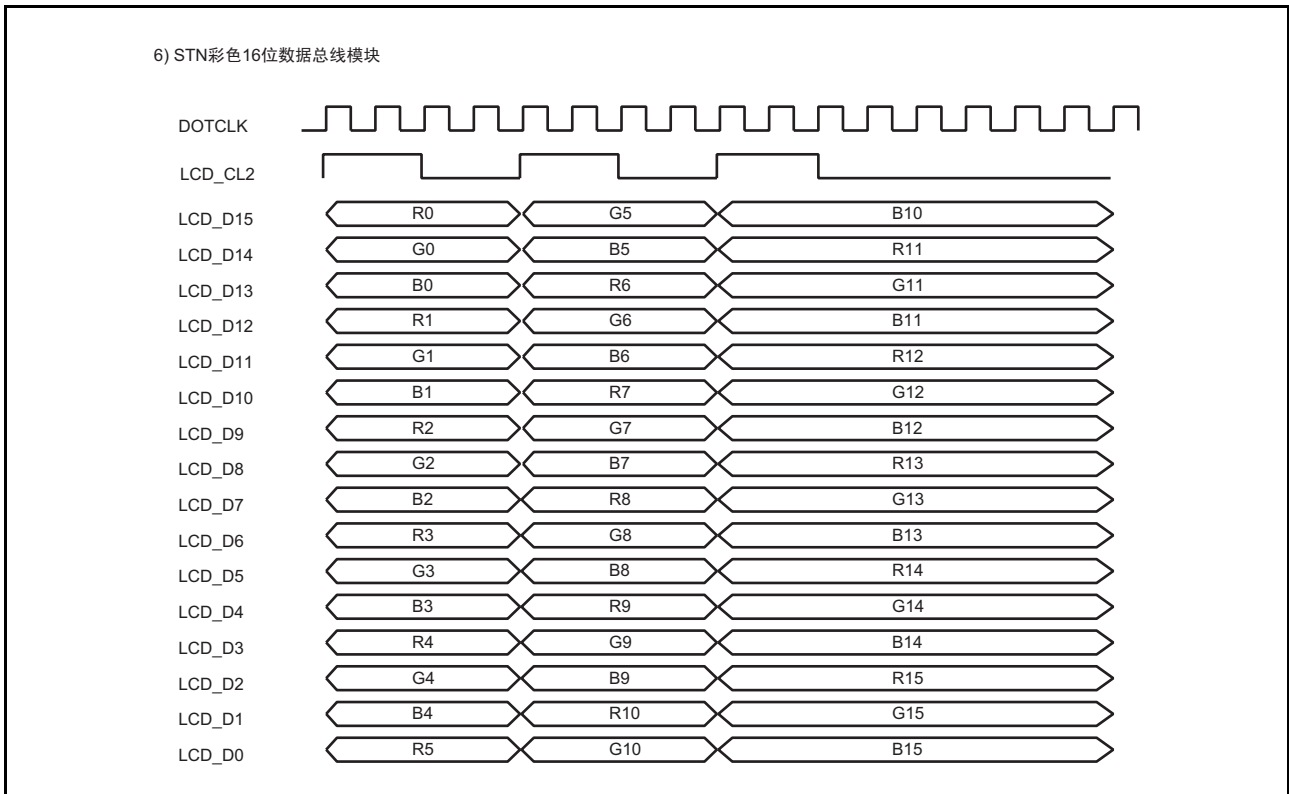


图 22.13 时钟和 LCD 数据信号例子 (STN 彩色 16 位数据总线模块)

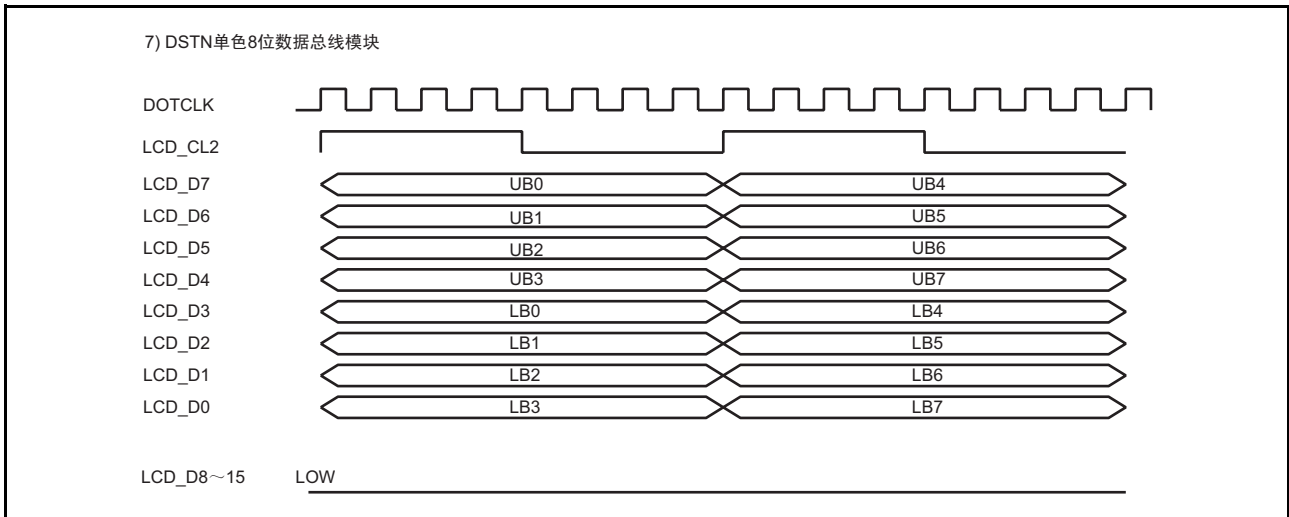


图 22.14 时钟和 LCD 数据信号例子 (DSTN 单色 8 位数据总线模块)

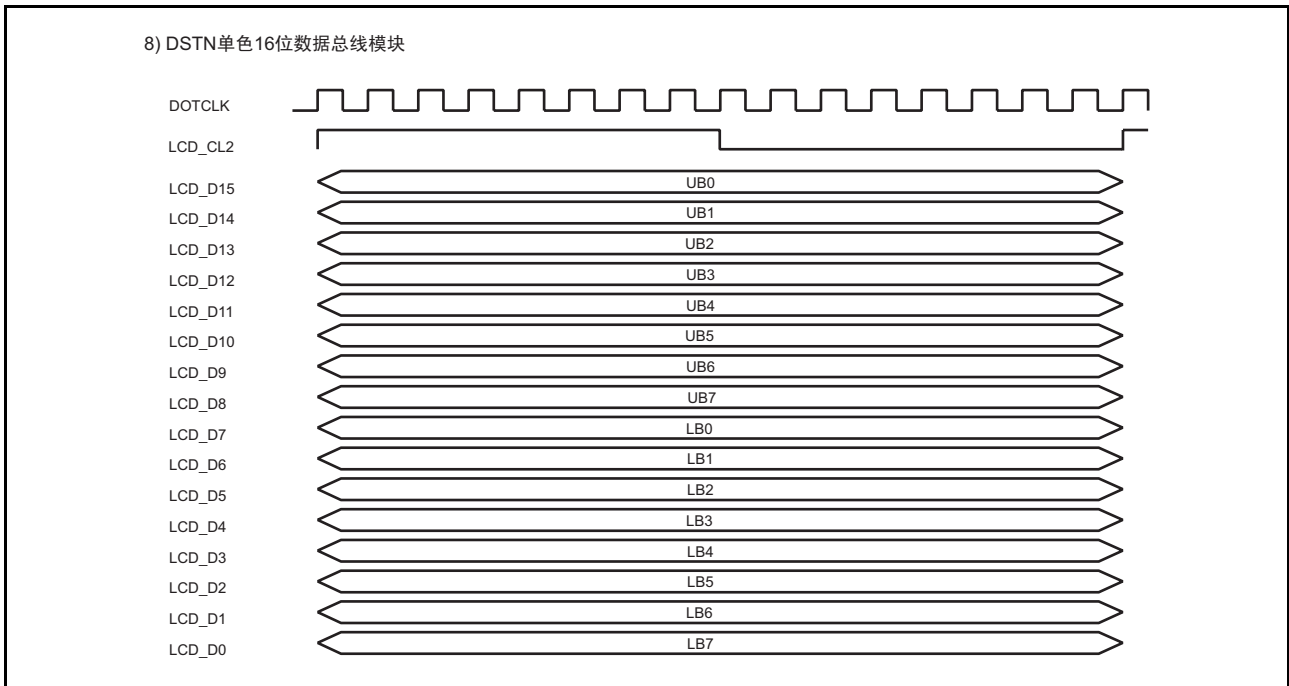


图 22.15 时钟和 LCD 数据信号例子 (DSTN 单色 16 位数据总线模块)

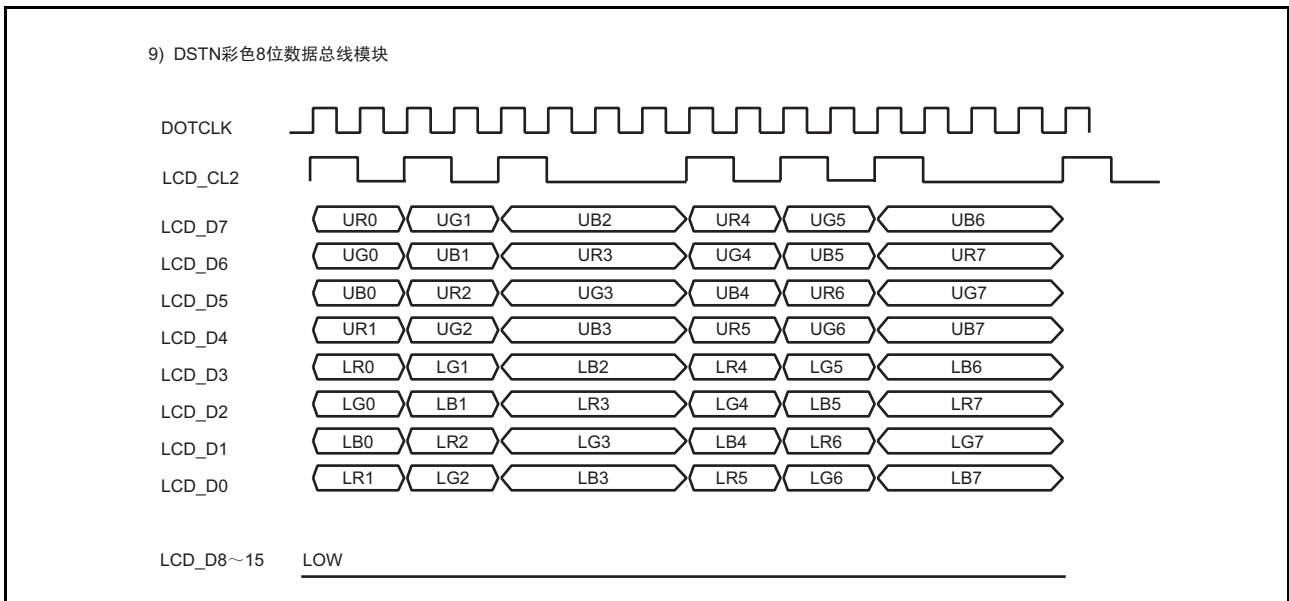


图 22.16 时钟和 LCD 数据信号例子 (DSTN 彩色 8 位数据总线模块)

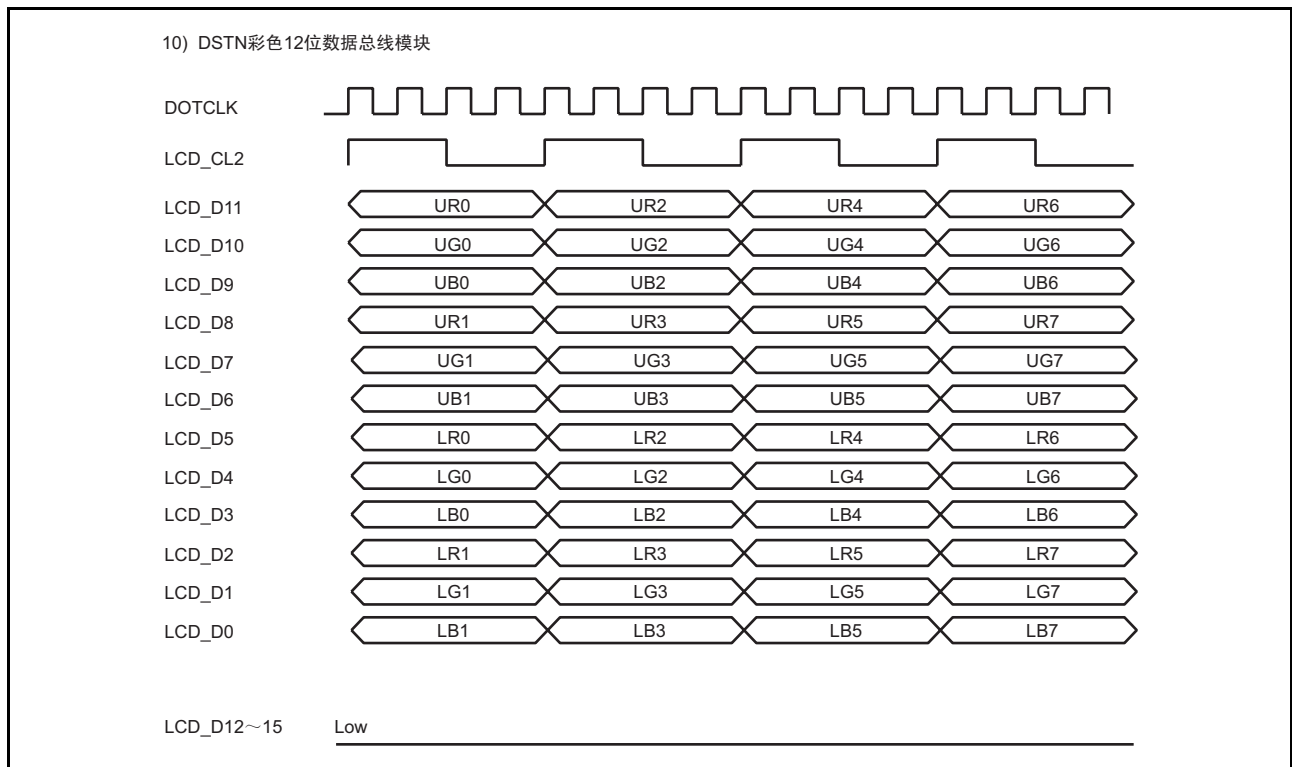


图 22.17 时钟和 LCD 数据信号例子 (DSTN 彩色 12 位数据总线模块)

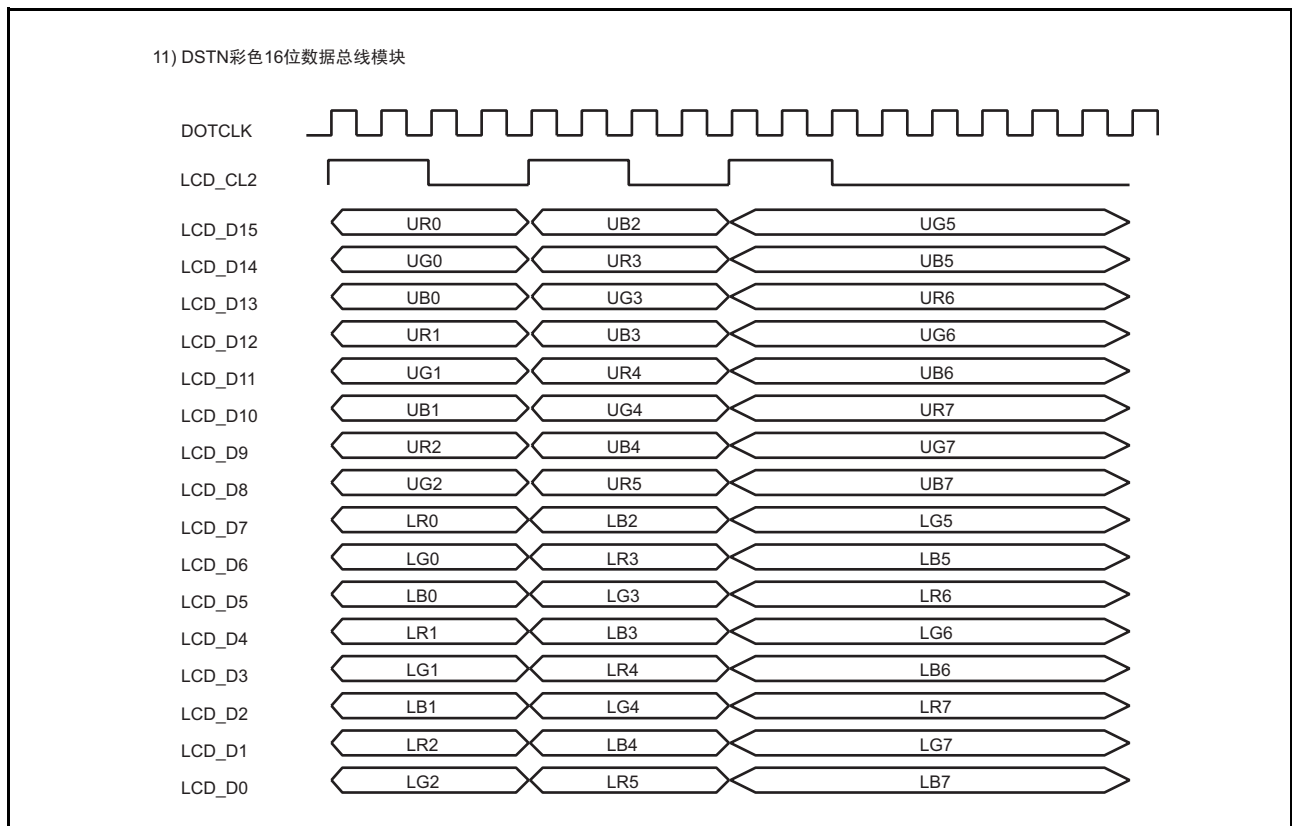


图 22.18 时钟和 LCD 数据信号例子 (DSTN 彩色 16 位数据总线模块)

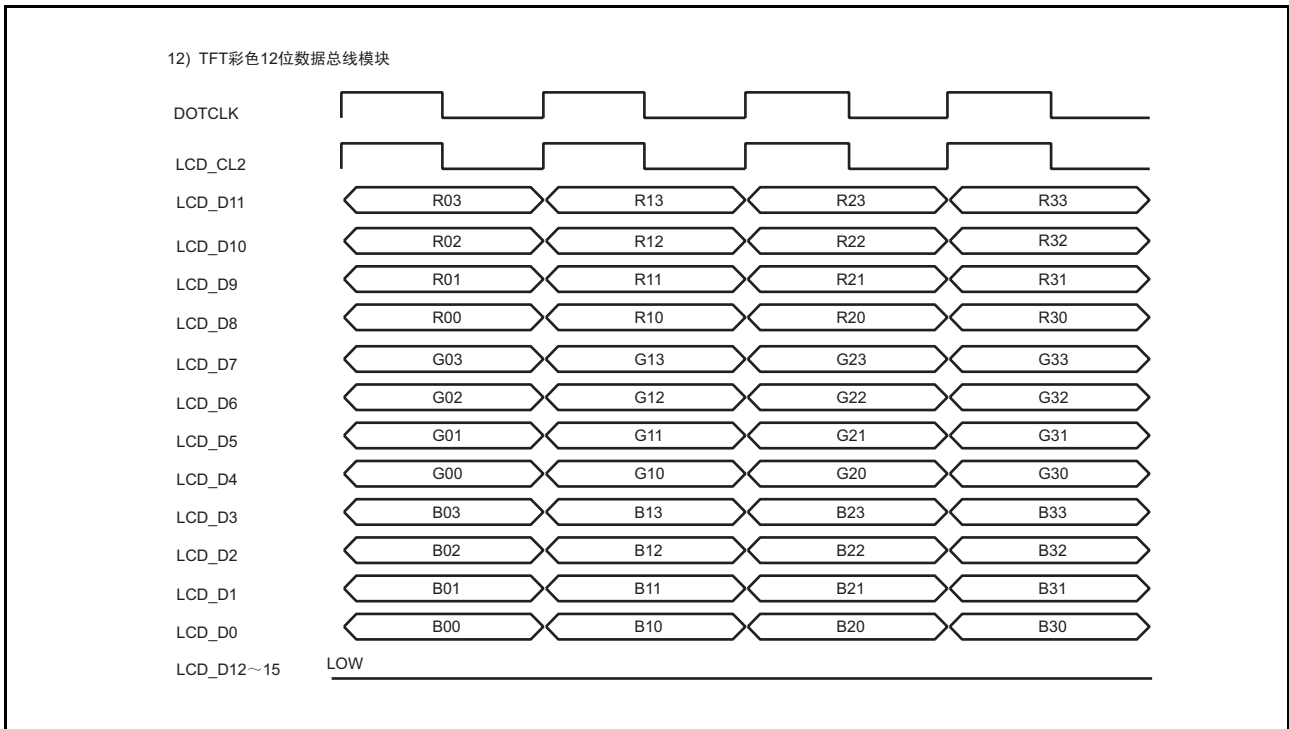


图 22.19 时钟和 LCD 数据信号例子 (TFT 彩色 12 位数据总线模块)

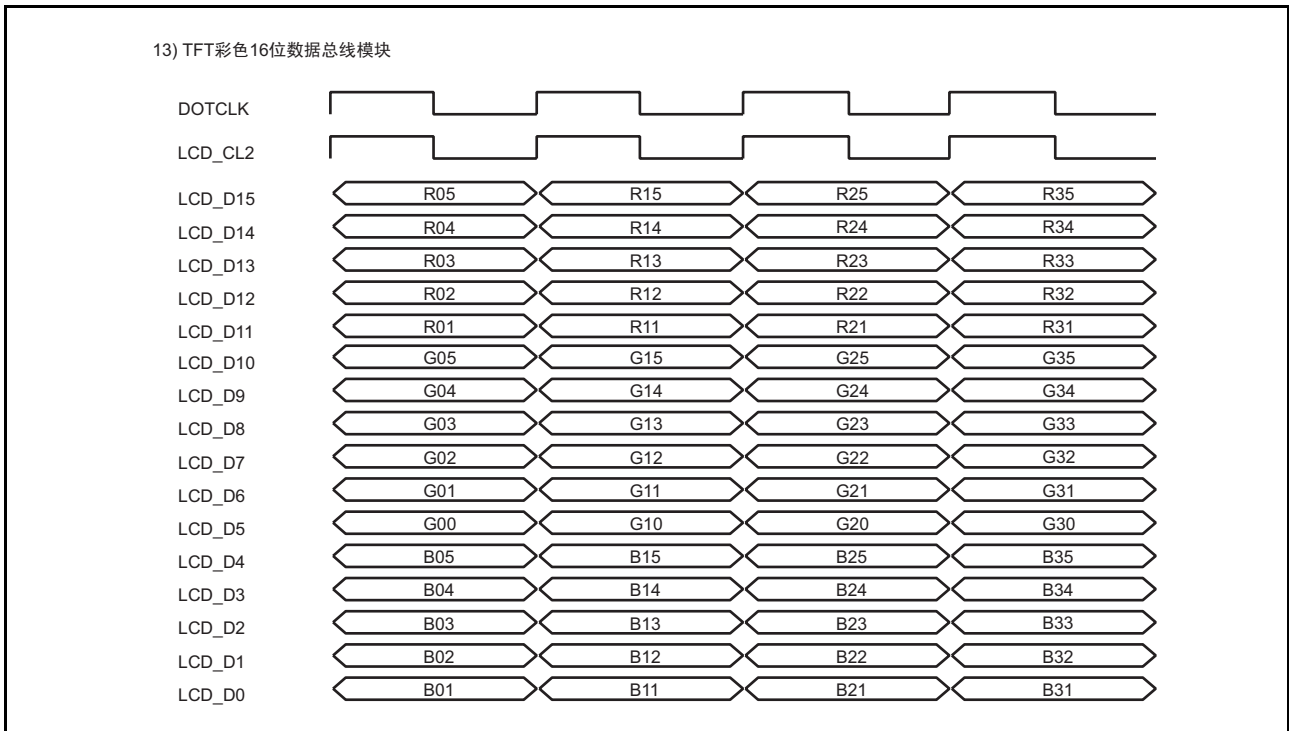


图 22.20 时钟和 LCD 数据信号例子 (TFT 彩色 16 位数据总线模块)

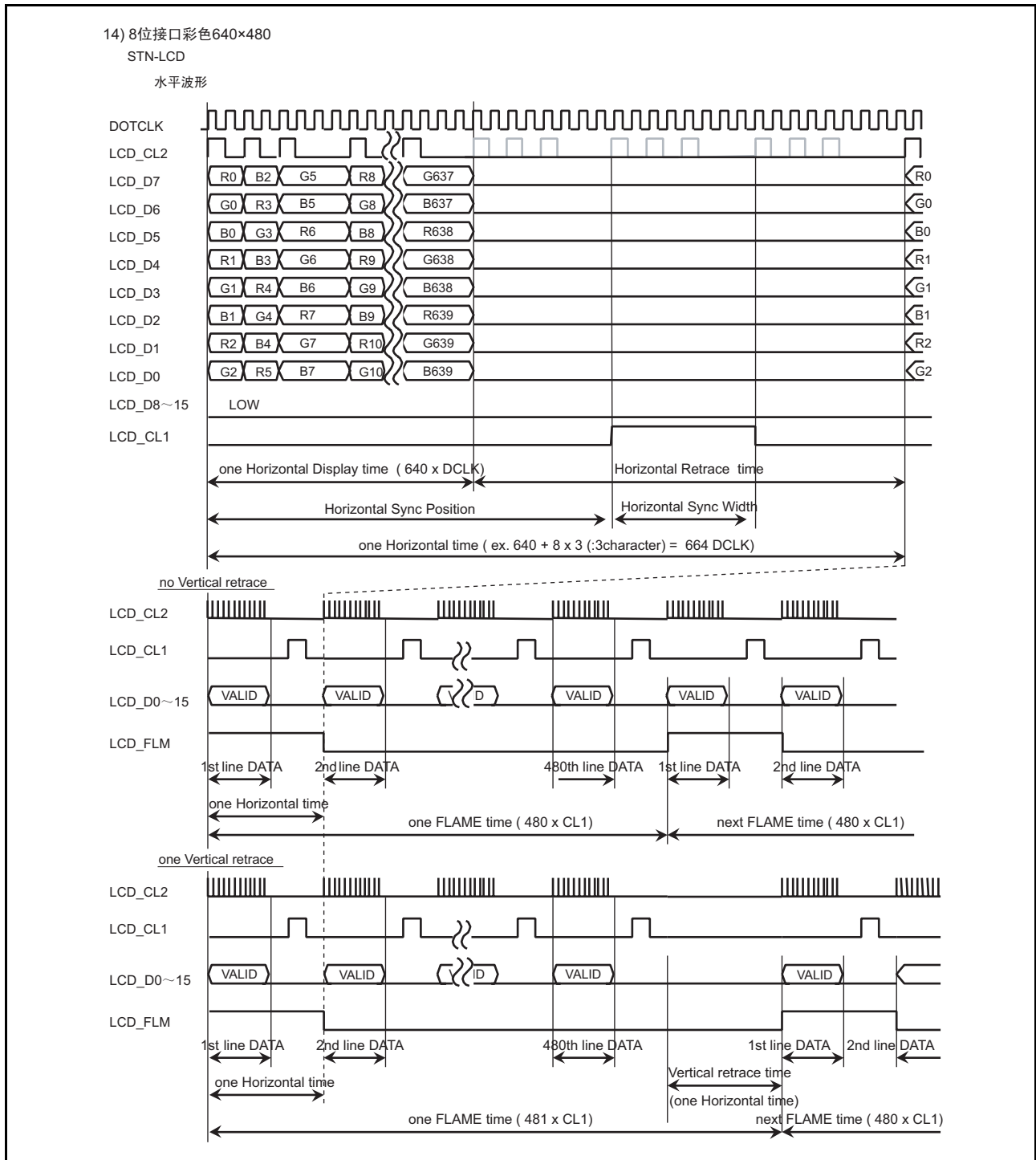


图 22.21 时钟和 LCD 数据信号例子 (8 位接口彩色 640×480)

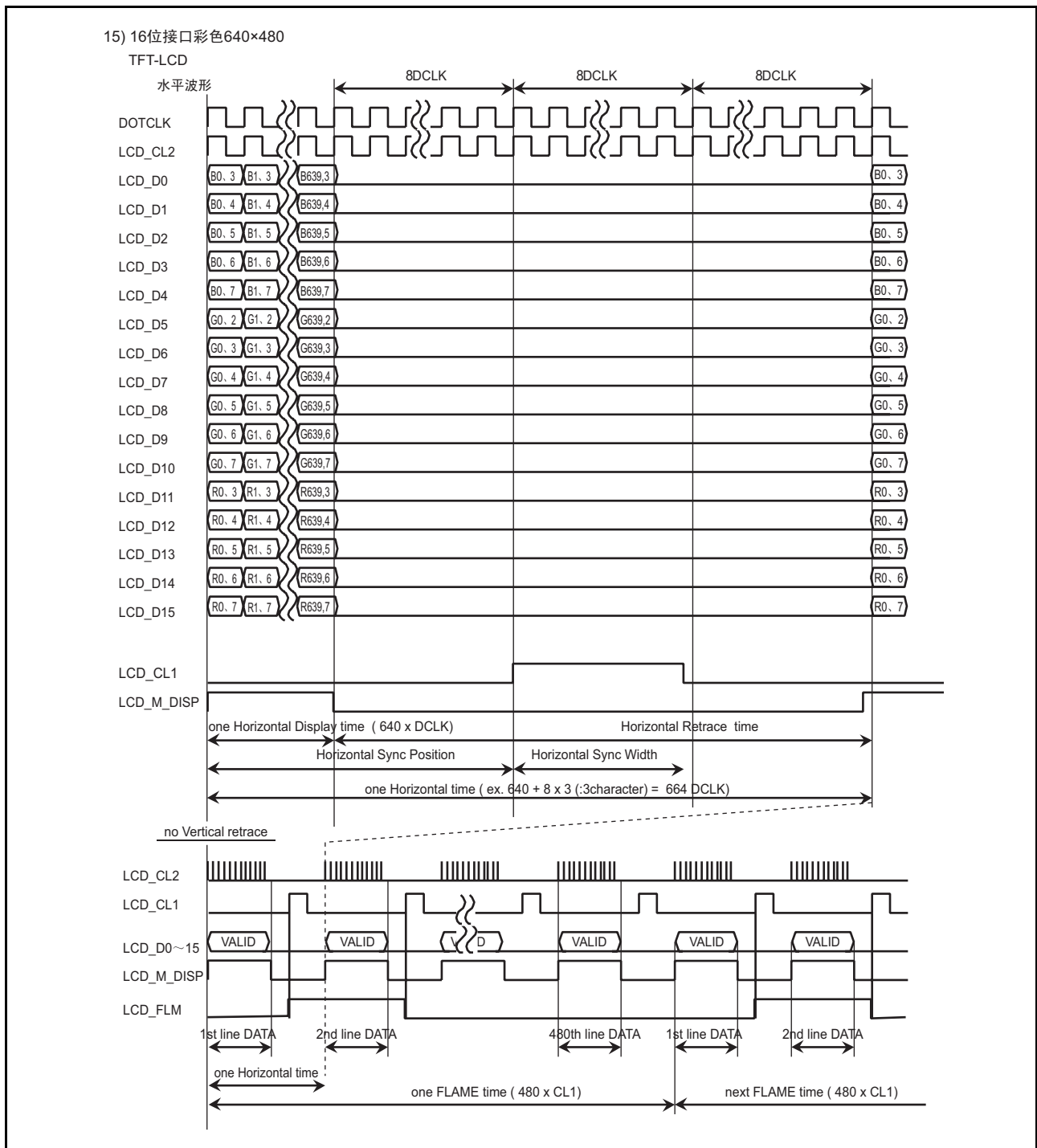


图 22.22 时钟和 LCD 数据信号例子 (16 位接口彩色 640×480)

22.6 使用时的注意事项

22.6.1 显示数据保存的 VRAM (区域 1、2 的 SDRAM) 的存取停止步骤

显示数据保存的 VRAM (区域 1、2 的 SDRAM) 的存取停止步骤如下所示:

- 显示数据保存的 VRAM 的存取停止步骤
 - A. 确认当前的状态, LDPMMR 的 LPS1 和 LPS0 位是否为 1。
 - B. 将 LDCNTR 的 DON 位置 0 (显示 OFF 模式)。
 - C. 确认 LDPMMR 的 LPS1 和 LPS0 位是否为 0。
 - D. 等待 1 帧的显示时间。

在将显示数据保存的 VRAM (区域 1、2 的 SDRAM) 设定为自刷新前, 或者在进入待机模式、模块待机等前, 需要执行此停止步骤。

22.6.2 MCU 请求接受停止时的注意事项

在 MCU 的请求屏蔽设定寄存器 (RQM) 的 bit24 (NMIE) 和 bit16 (LCDC) 都为 1 时, 如果因 NMI 中断而将 NMIFCR 的 NMIFL 位置 1, LCDC 就无法存取显示数据的保存 VRAM (区域 1、2 的 SDRAM) (保留请求)。

因为 LCDC 连续将保存在行缓冲器中的数据输出到 LCD 显示屏的数据引脚, 所以如果行缓冲器变空, LCD 就有可能停止显示。因此, 在行缓冲器变空前, 必须将 NMI 中断设定为无效, 并且将 NMIFL 位清 0。

同样地, 当接收到外部的总线释放请求 (BREQ) 时, LCDC 也无法存取显示数据的保存 VRAM (区域 1、2 的 SDRAM) (保留请求)。

与 NMIF 时的情况相同, 必须在行缓冲器变空前重新获得总线权。

第 23 章 2D 图形引擎 (G2D)

23.1 基本功能

23.1.1 命令、绘图属性一览

表 23.1 命令、绘图属性一览表 (1)

命令	OP CODE								Draw Mode							
	b31	b30	b29	b28	b27	b26	b25	b24	b15	b14	b13	b12	b11	b10	b9	b8
									MTRE	reserve	CLIP	RCLIP	STRANS	DTRANS/ LINKE	WORK/ LREL	SS
POLYGON4A	1	0	0	0	0	0	1	0	MTRE		CLIP	RCLIP	STRANS		WORK	SS
POLYGON4B					0	0	0	1	MTRE		CLIP	RCLIP	STRANS		WORK	SS
POLYGON4C					0	0	0	0	MTRE		CLIP	RCLIP			WORK	
LINEA	1	0	1	1	0	0	1	0	MTRE		CLIP	RCLIP	STRANS			SS(0)
LINEB					0	0	0	1	MTRE		CLIP	RCLIP	STRANS			SS(0)
LINEC					0	0	0	0	MTRE		CLIP	RCLIP		LINKE	LREL	
LINED					0	0	1	1	MTRE		CLIP	RCLIP		LINKE	LREL	
RLINEA					0	1	1	0	MTRE		CLIP	RCLIP	STRANS			SS(0)
RLINEB					0	1	0	1	MTRE		CLIP	RCLIP	STRANS			SS(0)
RLINEC					0	1	0	0	MTRE		CLIP	RCLIP		LINKE	LREL	
RLINED					0	1	1	1	MTRE		CLIP	RCLIP		LINKE	LREL	
FTRAPC	1	1	0	1	0	0	0	0	MTRE		CLIP	RCLIP		LINKE	LREL	
RFTRAPC					0	1	0	0	MTRE		CLIP	RCLIP		LINKE	LREL	
CLRWC	1	1	1	0	0	0	0	0	MTRE		CLIP	RCLIP				
LINEWC	1	1	1	1	0	0	0	0	MTRE		CLIP	RCLIP				
RLINEWC					0	1	0	0	MTRE		CLIP	RCLIP				
BITBLTA	1	0	1	0	0	0	1	0	MTRE		CLIP	RCLIP	STRANS	DTRANS	WORK	SS
BITBLTB					0	0	0	1	MTRE		CLIP	RCLIP	STRANS	DTRANS	WORK	SS
BITBLTC					0	0	0	0	MTRE		CLIP	RCLIP		DTRANS	WORK	
测试模式	1	0	1	0	1	0	0	0	用于内部验证，不能设定。不将命令错误位 (CER) 置位。							

命令	OP CODE								Draw Mode							
									b7	b6	b5	b4	b3	b2	b1	b0
	b31	b30	b29	b28	b27	b26	b25	b24	REL	STYLE/ SRCDIRX	BLKE/ SRCDIRY	NET/EDG/ DSTDIRX	EOS/ DSTDIRY	COOF	AA/ αE	CLKW/ S αE
POLYGON4A	1	0	0	0	0	0	1	0	REL	STYLE	BLKE	NET	EOS	COOF	αE	S αE
POLYGON4B					0	0	0	1	REL	STYLE	BLKE	NET	EOS	COOF	αE	
POLYGON4C					0	0	0	0			BLKE	NET	EOS	COOF	αE	
LINEA	1	0	1	1	0	0	1	0	REL	STYLE(1)		NET	EOS	COOF	AA	
LINEB					0	0	0	1	REL	STYLE(1)		NET	EOS	COOF	AA	
LINEC					0	0	0	0				NET	EOS	COOF	AA	
LINED					0	0	1	1							AA(1)	CLKW
RLINEA					0	1	1	0	REL	STYLE(1)		NET	EOS	COOF	AA	
RLINEB					0	1	0	1	REL	STYLE(1)		NET	EOS	COOF	AA	
RLINEC					0	1	0	0				NET	EOS	COOF	AA	
RLINED					0	1	1	1							AA(1)	CLKW
FTRAPC	1	1	0	1	0	0	0	0			BLKE(1)	EDG	EOS			
RFTRAPC					0	1	0	0			BLKE(1)	EDG	EOS			
CLRWC	1	1	1	0	0	0	0	0			BLKE(1)					
LINEWC	1	1	1	1	0	0	0	0					EOS			
RLINEWC					0	1	0	0					EOS			
BITBLTA	1	0	1	0	0	0	1	0	REL	SRCDIRX	SRCDIRY	DSTDIRX	DSTDIRY	COOF	αE	S αE
BITBLTB					0	0	0	1	REL	SRCDIRX	SRCDIRY	DSTDIRX	DSTDIRY	COOF	αE	
BITBLTC					0	0	0	0				DSTDIRX	DSTDIRY	COOF	αE	
测试模式	1	0	1	0	1	0	0	0	用于内部验证, 不能设定。不将命令错误位 (CER) 置位。							

REL : 只在 SS=0 时有效, 在 SS=1 时必须置 0。

COOF : 只在 16 位 / 像素 (GBM=1) 时有效, 在 8 位 / 像素 (GBM=0) 时必须置 0。

S αE : 只在 ARGB 格式 (SPF=DPF=1) 时有效, 在 RGB 格式 (SPF=DPF=0) 并且 8 位 / 像素 (GBM=0) 时或者在 $\alpha E=0$ 时, 必须置 0。

αE : 只在 16 位 / 像素 (GBM=1) 时有效, 在 8 位 / 像素 (GBM=0) 时必须置 0。

在 POLYGON4A/B/C 命令的情况下, 只在 BLKE=1 时有效, 在 BLK=0 时必须置 0。

在 BITBLTA/B/C 命令的情况下, 只在 ROP 代码 =H'CC 时有效, 在其他 ROP 码时必须置 0。

LREL : 只在 LINKE=1 时有效, 在 LINKE=0 时必须置 0。

STYLE: 在 BLKE=1 或者 (R) LINEA/B 命令时必须置 1。

AA : 在 NET=1 时必须置 0。只在 16 位 / 像素 (GBM=1) 时有效, 在 8 位 / 像素 (GBM=0) 时必须置 0。

在 (R) LINED 命令时必须置 1。

SS : 在 (R) LINEA/B 命令时必须置 0。

BLKE : 在 (R) FTRAPC 和 CLRWC 命令时必须置 1。

阴影部: 不能使用 (必须置 0)。

表 23.2 命令、绘图属性一览表 (2)

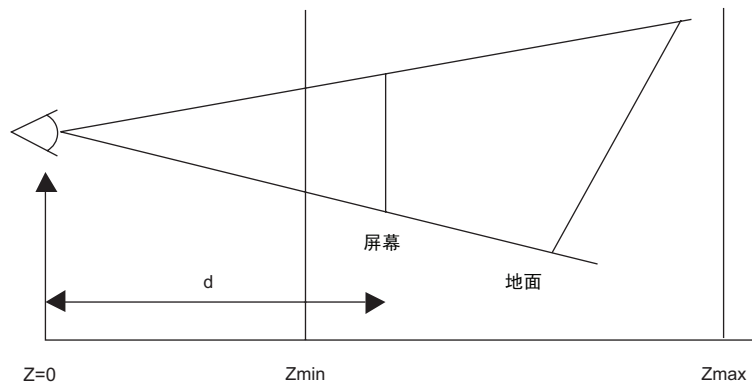
命令	OP CODE								Draw Mode							
	b31	b30	b29	b28	b27	b26	b25	b24	b15	b14	b13	b12	b11	b10	b9	b8
TRAP	0	0	0	0	0	0	0	0								
NOP/INT	0	0	0	0	1	0	0	0	INT							
VBKEM	0	0	0	1	0	0	0	0								
WPR	0	0	0	1	1	0	0	0						LINKE	LREL	
JUMP	0	0	1	0	1	0	0	0								
GOSUB	0	0	1	1	0	0	0	0								
RET	0	0	1	1	1	0	0	0								
LCOFS	0	1	0	0	0	0	0	0								
RLCOFS	0	1	0	0	0	1	0	0								
MOVE	0	1	0	0	1	0	0	0								
RMOVE	0	1	0	0	1	1	0	0								
测试模式	0	1	0	1	0	0	0	0	用于内部验证，不能设定。不将命令错误位 (CER) 置位。							

命令	OP CODE								Draw Mode							
	b31	b30	b29	b28	b27	b26	b25	b24	b7	b6	b5	b4	b3	b2	b1	b0
TRAP	0	0	0	0	0	0	0	0			Flip5	Flip4	Flip3	Flip2	Flip5	Flip1
NOP/INT	0	0	0	0	1	0	0	0	INT NO							
VBKEM	0	0	0	1	0	0	0	0								
WPR	0	0	0	1	1	0	0	0					ByteM3	ByteM2	ByteM1	ByteM0
JUMP	0	0	1	0	1	0	0	0	REL							
GOSUB	0	0	1	1	0	0	0	0	REL							NO
RET	0	0	1	1	1	0	0	0								NO
LCOFS	0	1	0	0	0	0	0	0								
RLCOFS	0	1	0	0	0	1	0	0								
MOVE	0	1	0	0	1	0	0	0								
RMOVE	0	1	0	0	1	1	0	0								
测试模式	0	1	0	1	0	0	0	0	用于内部验证，不能设定。不将命令错误位 (CER) 置位。							

23.1.2 基本功能

(1) 特点

- 装载了坐标变换几何引擎
内置对输入顶点进行坐标变换 (4×4 矩阵运算+Z 裁剪+透视 W 除法运算) 的硬件。
- 扩展了 2D 功能
内置高功能粗线描绘、抗锯齿线描绘、带 ROP/α 混合的 BITBLT 命令。
- 扩充了控制命令功能
扩充了 2 条命令系统: INT 命令、GOSUB/RET 命令, 并且扩充了 WPR 和 TRAP 命令功能
- 在功能性方面与 Q2SD 向上兼容



$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{pmatrix} \quad (1)$$

$$\begin{pmatrix} m00 & m01 & 0 & m03 \\ m10 & m11 & 0 & m13 \\ 0 & 0 & 0 & 0 \\ m20/d & m21/d & 0 & m23/d \end{pmatrix} \quad (2)$$

将上述 9 个参数设定为坐标变换控制寄存器的矩阵参数 A ~ I (MTRAR ~ MTRIR)。

按照 IEEE754 单精度浮点格式设定 MTRAR ~ MTRIR。但是，因为以 32 位定点数（整数部 16 位、尾数部 16 位）进行内部运算，所以必须将参数 A ~ I 设定为 $-2^{15} \leq \text{MTRAR} \sim \text{MTRIR} < 2^{15}$ 。在从单精度浮点转换为 32 位定点数时，如果超过此范围，就进行饱和和处理。在设定参数 A ~ I 时，必须保证矩阵的运算结果 TX、TY、W 不超过以下范围：

$$-H'7FFF.FFFF \leq \text{TX}, \text{TY} \leq H'7FFF.FFFF$$

$$H'0000.0001 \leq W \leq H'7FFF.FFFF$$

如果 $-H'7FFF.FFFF \leq \text{TX}, \text{TY}, W \leq H'7FFF.FFFF$ 范围，就将状态寄存器的矩阵运算错误位 (MTRER) 置位，并且进行饱和和处理。

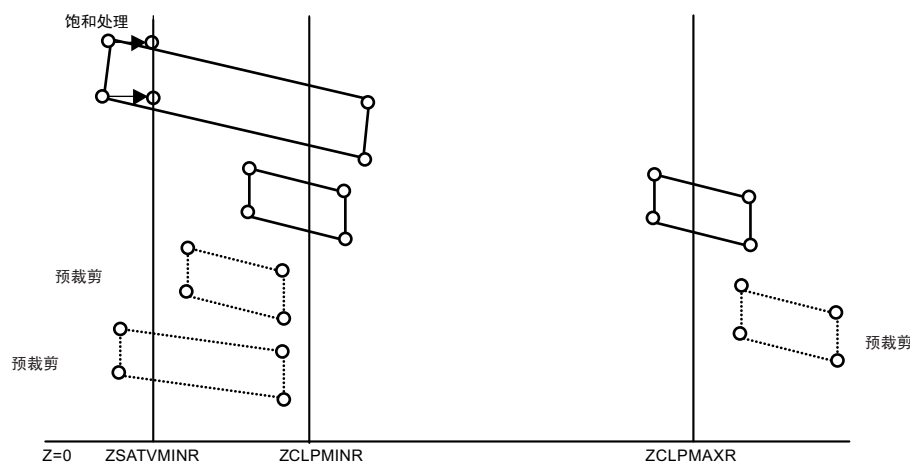
- 【注】**
1. 在 G2D 内部给输入坐标 (X, Y) 加上局部偏移值后进行运算。
 2. 在指定相对坐标的命令时，在 G2D 内部将相对坐标改为绝对坐标后进行矩阵运算。
 3. BITBLT 命令只对中心坐标 (BXC, BYC) 进行矩阵运算，在描绘粗线时只对起点 / 终点坐标进行矩阵运算。
 4. 坐标变换控制寄存器的坐标变换允许位 (GTE) 或者绘图属性的 MTRE 位不能屏蔽矩阵运算错误位 (MTRER)，所以在 GTE=0 或者 1 并且 MTRE=0 时，即使不进行坐标变换，MTRER 也可能被置 1。
因此，从绘图开始到 TRAP 命令期间，除非 GTE=1 并且 MTRE=1，否则就不能使用 MTRER。

(3) Z 裁剪

在绘画图形的所有顶点（参照下述的【注】）小于 Z 裁剪区 MIN 寄存器（ZCLPMINR）的值或者大于 Z 裁剪区 MAX 寄存器（ZCLPMAXR）的值时，进行预裁剪。因为 Z 裁剪使用 W 值进行比较，所以必须用 W 的相应值（在上述例子中为 Z_{min}/d 和 Z_{max}/d ）设定 ZCLPMINR 和 ZCLPMAXR。在未被预裁剪的情况下，当 Z 坐标值小于等于 Z 饱和值 MIN 寄存器（ZSATVMINR）的设定值时，就对 ZSATVMINR 的设定值进行饱和处理。也必须给 ZSATVMINR 设定 W 的相应值。如果进行饱和处理，图形就会变形，所以为了不使 Z 坐标值小于等于 ZSATVMINR 值，必须设定矩阵参数；或者为了使其不出现在屏幕坐标上，必须预先分割绘画图形，使其变为显示列表。按照 IEEE754 单精度浮点格式设定 ZCLPMINR、ZCLPMAXR 和 ZSATVMINR。但是，因为以 32 位定点数（整数部为 16 位、尾数部为 16 位）进行内部运算，所以必须设定为 $2^{-16} \leq ZSATVMINR \leq ZCLPMINR \leq ZCLPMAXR < 2^{15}$ 。如果超过此范围，就进行饱和处理。

为了保持图案的连续性，(R) LINE 类和 (R) LINEW 类命令不进行 Z 预裁剪。因此，描绘屏幕坐标中未出现的图形，可能会发生性能劣化的情况。在不希望性能劣化时，必须将绘图控制寄存器的线预裁剪允许位 (LPCE) 置 1。如果将 LPCE 置 1，就在 2 维裁剪区（系统裁剪区、用户裁剪区和相对用户裁剪区）中以线段为单位进行预裁剪，因此能改善性能。但是，如果中间的线段被预裁剪，图案就不连续（图案从以前描绘的线段的终点开始）。

【注】 在 (R) FTRAPC 命令的情况下，如果边缘描绘有效 (EDG=1)，就不对边缘线进行 Z 预裁剪。



【注】 POLYGON4系、CLRWC: 坐标变换后的4个顶点
 BITBLT系 : 坐标变换后的中心坐标的1个顶点
 (R) FTRAPC : 坐标变换后的外接四边形的4个顶点

(4) 透视 W 除法运算

根据坐标变换控制寄存器的仿射变换允许位 (AFE) 的设定，对矩阵运算后的 TX、TY 进行如下处理：

- AFE位=0时
 输出的X坐标X'和输出的Y坐标Y': $X'=TX/WC+GTROFSX$ 、 $Y'=TY/WC+GTROFSY$ 。
 分别将GTROFSX和GTROFSY设定到坐标变换偏移X寄存器 (GTROFSXR)和坐标变换偏移Y寄存器 (GTROFSYR)。用16位整数 (2的补码) 设定GTROFSX和GTROFSY。
- AFE位=1时
 输出的X坐标X'和输出的Y坐标Y': $X'=TX$ 、 $Y'=TY$ 。不进行Z裁剪、透视W除法运算和偏移加法运算。

在 AFE 位 =0 时，如果 W 除法运算后的 TX/WC 和 TY/WC 超过以下范围，就进行饱和处理。

$$-H'7FFF \leq TX/WC, TY/WC \leq H'7FFF$$

如果偏移 (GTROFSX 和 GTROFSY) 加法运算后的 $TX/WC+GTROFSX$ 和 $TY/WC+GTROFSY$ 超过以下范围, 就进行饱和处理。

$$-H'7FFF \leq TX/WC+GTROFSX, \quad TY/WC+GTROFSY \leq H'7FFF$$

在进行饱和处理后, 命令从饱和处理后的顶点坐标继续执行。

【注】 用 BITBLT 类命令获取的 4 个顶点是变换坐标后的中心坐标值加上宽度 (LW, RW) 和高度 (TH, BH) 的值。在描绘粗线时, 从坐标变换后的起点 / 终点和线宽 (W) 获取 4 个顶点。因此, 这 2 条命令对从基准点 (BITBLT: 中心坐标, 粗线: 起点、终点坐标) 获取的顶点不进行饱和处理, 所以必须注意不能超过绘图坐标。

(5) 坐标变换的流程和饱和处理

按照以下顺序进行坐标变换。

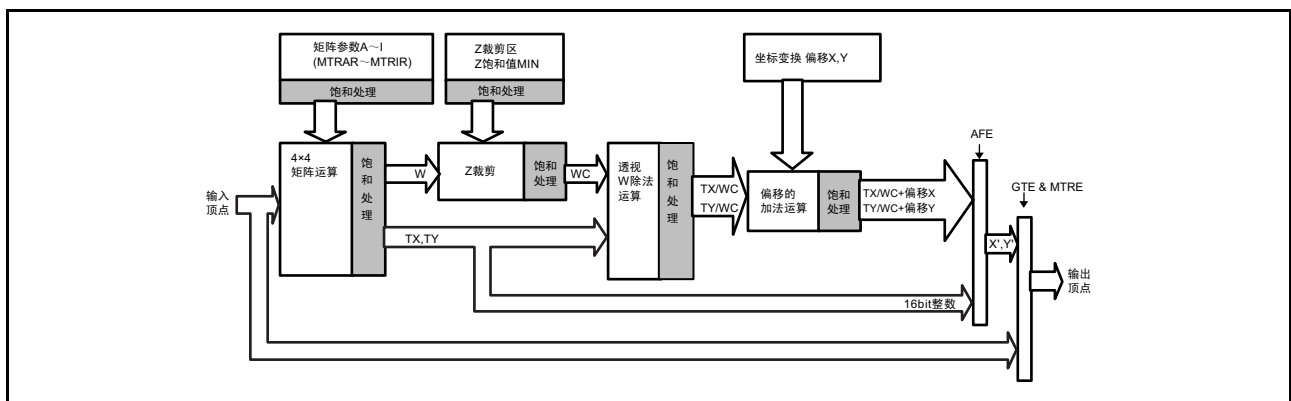


图 23.1 坐标变换的流程图

表 23.3 寄存器的设定参数的范围和饱和处理

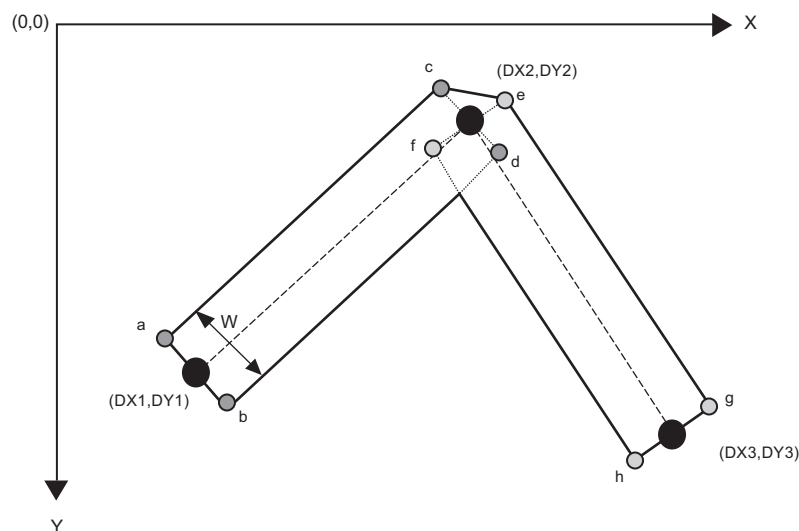
寄存器的设定参数	设定范围	饱和处理
矩阵参数 A ~ I	$-2^{15} \leq \text{矩阵参数 A} \sim \text{I} < 2^{15}$ (IEEE754 单精度浮点)	矩阵参数 A ~ I $\geq 2^{15}$ 时: H'7FFF.FFFF 矩阵参数 A ~ I $< -2^{15}$ 时: -H'8000.0000 (32 位定点)
Z 裁剪区、 Z 饱和值 MIN	$2^{-16} \leq \text{Z 裁剪区、Z 饱和值 MIN} < 2^{15}$ (IEEE754 单精度浮点)	Z 裁剪区、Z 饱和值 MIN $\geq 2^{15}$ 时: H'7FFF.FFFF Z 裁剪区、Z 饱和值 MIN $< 2^{-16}$ 时: H'0000.0001 (32 位定点)
坐标变换偏移 X, Y	$-2^{15} \leq \text{坐标变换偏移 X, Y} \leq 2^{15}-1$ (16 位整数 (2 的补码))	—

表 23.4 运算后的顶点范围和饱和处理

寄存器的设定参数	范围	饱和处理
TX, TY, W	$-H'7FFF.FFFF \leq TX, TY, W \leq H'7FFF.FFFF$ (32 位定点)	TX, TY, W > H'7FFF.FFFF 时: H'7FFF.FFFF TX, TY, W < -H'7FFF.FFFF 时: -H'7FFF.FFFF (32 位定点)
WC	Z 饱和值 MIN $\leq WC \leq H'7FFF.FFFF$ (32 位定点)	WC < Z 饱和值 MIN 时: Z 饱和值 MIN (32 位定点)
TX/WC, TY/WC TX/WC+ 偏移 X, TY/WC+ 偏移 Y	$-H'7FFF \leq TX/WC, TY/WC \leq H'7FFF$ $-H'7FFF \leq TX/WC+ 偏移 X, TY/WC$ $+ 偏移 Y \leq H'7FFF$ (16 位整数)	TX/WC, TY/WC > H'7FFF 时: H'7FFF TX/WC, TY/WC < -H'7FFF 时: -H'7FFF TX/WC+ 偏移 X, TY/WC+ 偏移 Y > H'7FFF 时: H'7FFF TX/WC+ 偏移 X, TY/WC+ 偏移 Y < -H'7FFF 时: -H'7FFF (16 位整数)
X', Y'	$-H'7FFF \leq X', Y' \leq H'7FFF$ (16 位整数)	—
输出的顶点	$-H'8000 \leq \text{输出的顶点} \leq H'7FFF$ (16 位整数)	—

(6) 粗线的描绘

能通过 LINE 类和 RLINE 类命令给线宽 W 设定大于 0 的值进行粗线的描绘。通过起点 / 终点坐标和线宽 W 求得粗线坐标 a、b、c、d，并进行描绘。用 6 位整数部设定 W。如果设定为 0，就描绘一条线宽为 1 的线。和粗线的连接部分能通过绘图控制寄存器 (RCLR) 的连接描绘屏蔽位 (COM) 选择是否进行描绘。在指定粗线时，如果每个线段的起点 / 终点坐标相同，就不进行任何描绘。另外，在描绘粗线时，必须将起点 / 终点坐标设定在逻辑空间的 $-2^{15}+(W+2) \leq x, y \leq 2^{15}-1-(W+2)$ 的范围内。



(7) 抗锯齿

能用于 LINE 类和 RLINE 类命令，以便减少锯齿。通过将绘图属性的抗锯齿位 AA 位置 1 进行抗锯齿处理。在设定抗锯齿时，必须将起点 / 终点坐标设定在逻辑空间的 $-2^{15}+1 \leq x, y \leq 2^{15}-2$ 的范围内。在描绘粗线时，必须将起点 / 终点坐标设定在 $-2^{15}+1+(W+2) \leq x, y \leq 2^{15}-2-(W+2)$ 的范围内。

- 虚线时，对线的间隙不进行抗锯齿处理。

- 在每个线段的起点/终点坐标相同时，(R)LINEA/B/C 命令对线宽为 1 (W=0) 的线不进行抗锯齿处理而描绘 1 个点，对粗线不进行任何描绘。
- 在每个线段的起点/终点坐标相同时，(R)LINED 命令不进行任何描绘。
- 对水平、垂直和倾斜 45 度的线段不进行抗锯齿处理。

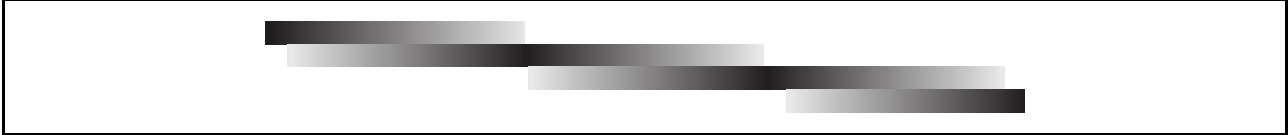


图 23.2 抗锯齿的指定例子

23.1.3 坐标系

G2D 有 4 个 2 维坐标（屏幕坐标、绘图坐标、2 维源坐标和工作坐标）和 1 个 1 维坐标（1 维源坐标）。

屏幕坐标是显示控制的坐标，屏幕坐标的 X 对应显示画面的水平坐标，Y 对应垂直坐标，原点在显示画面的左上方。X 轴的右边和 Y 轴的下边是屏幕坐标的正向坐标。1 个屏幕坐标的数据宽度能选择 16 位（16 位 / 像素）或者 8 位（8 位 / 像素）。

绘图坐标是绘图控制的坐标，是按绘图命令指定的偏移量对屏幕进行进行水平和垂直错位的坐标系。绘图命令在此坐标中进行描绘操作。1 个绘图坐标的数据宽度能选择 16 位（16 位 / 像素）或者 8 位（8 位 / 像素）。

2 维源坐标是绘图控制的坐标，是在执行绘图命令时由绘图命令指定的源（2 维）坐标系，在 SS=1 时使用。1 个 2 维源坐标的数据宽度能选择 16 位（16 位 / 像素）或者 8 位（8 位 / 像素）。

1 维源坐标是绘图控制的坐标，是在执行绘图命令时由绘图命令指定的源（1 维）坐标系，在 SS=0 时使用。1 个 1 维源坐标的数据宽度能选择 1 位（1 位 / 像素）、16 位（16 位 / 像素）或者 8 位（8 位 / 像素）。对 1 个 1 维源指定 1 个物理地址（左上方）、此 1 维源的水平宽度和垂直高度。

工作坐标是绘图控制的坐标，与绘图坐标一一对应，是在执行绘图命令时绘图命令指定的工作坐标系。1 个工作坐标的数据宽度为 1 位。

屏幕坐标的最大值为 X=4095、Y=4095。

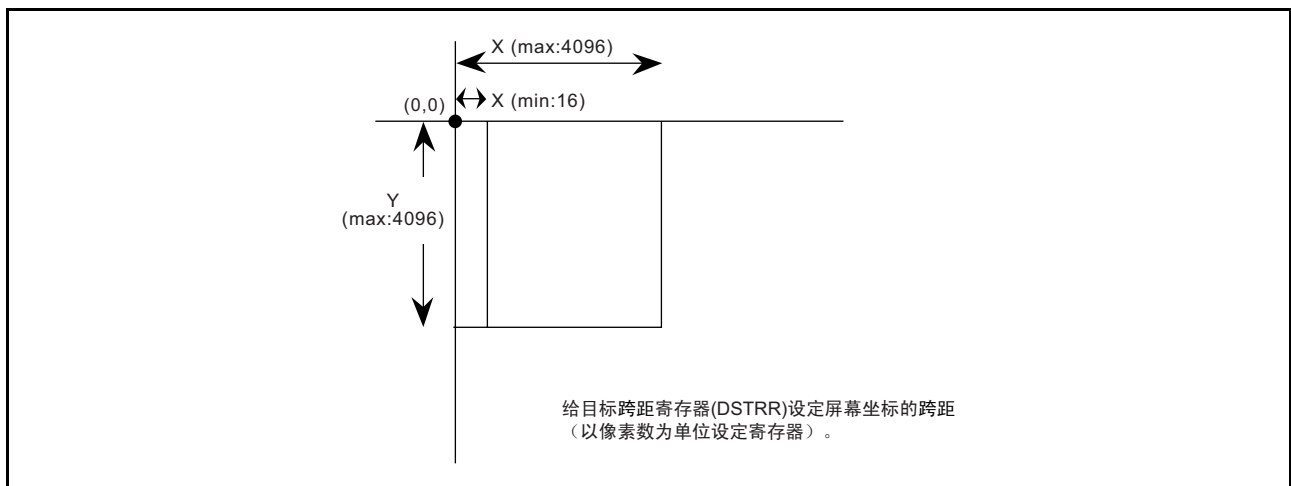


图 23.3 屏幕坐标

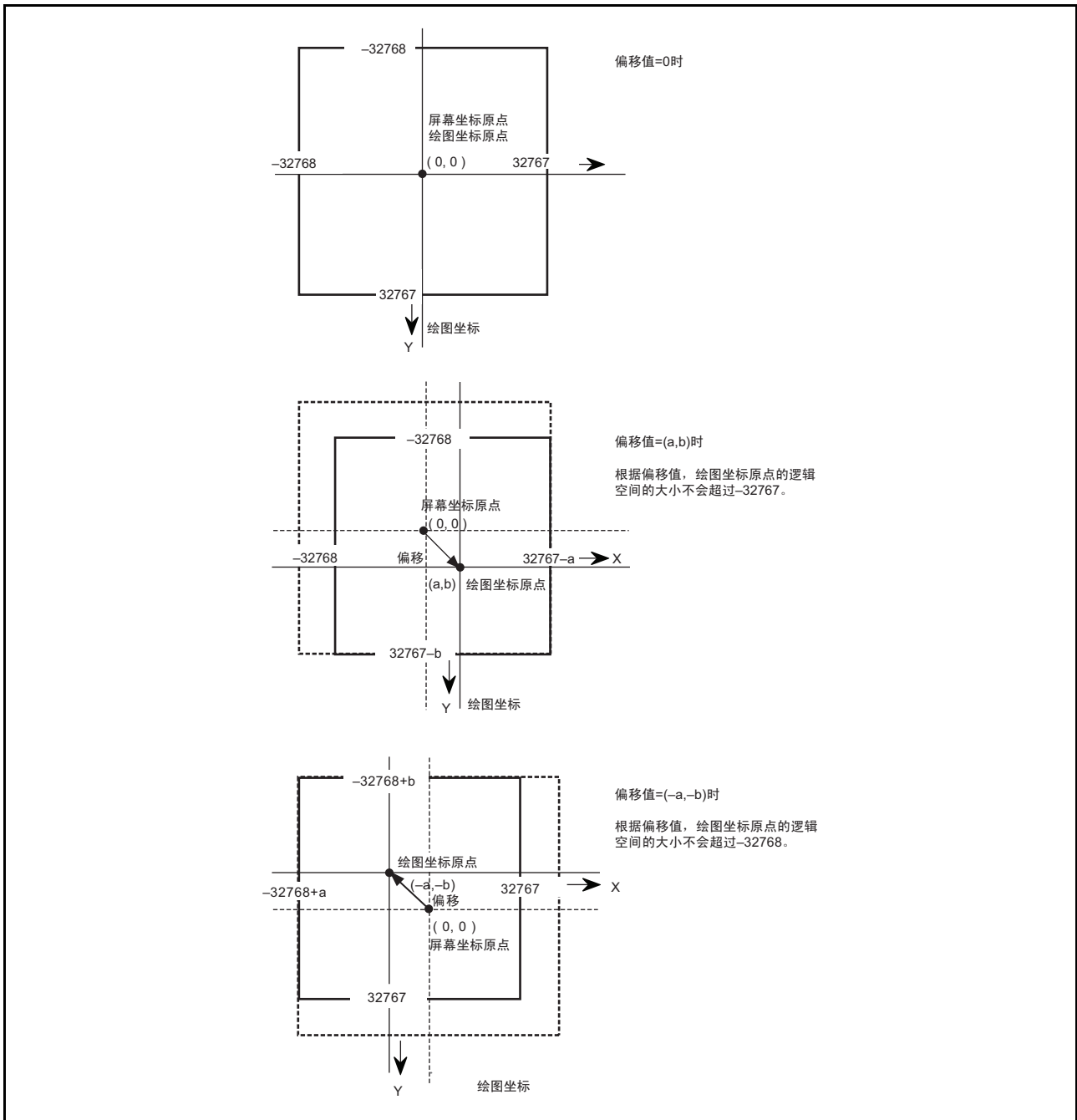


图 23.4 绘图坐标

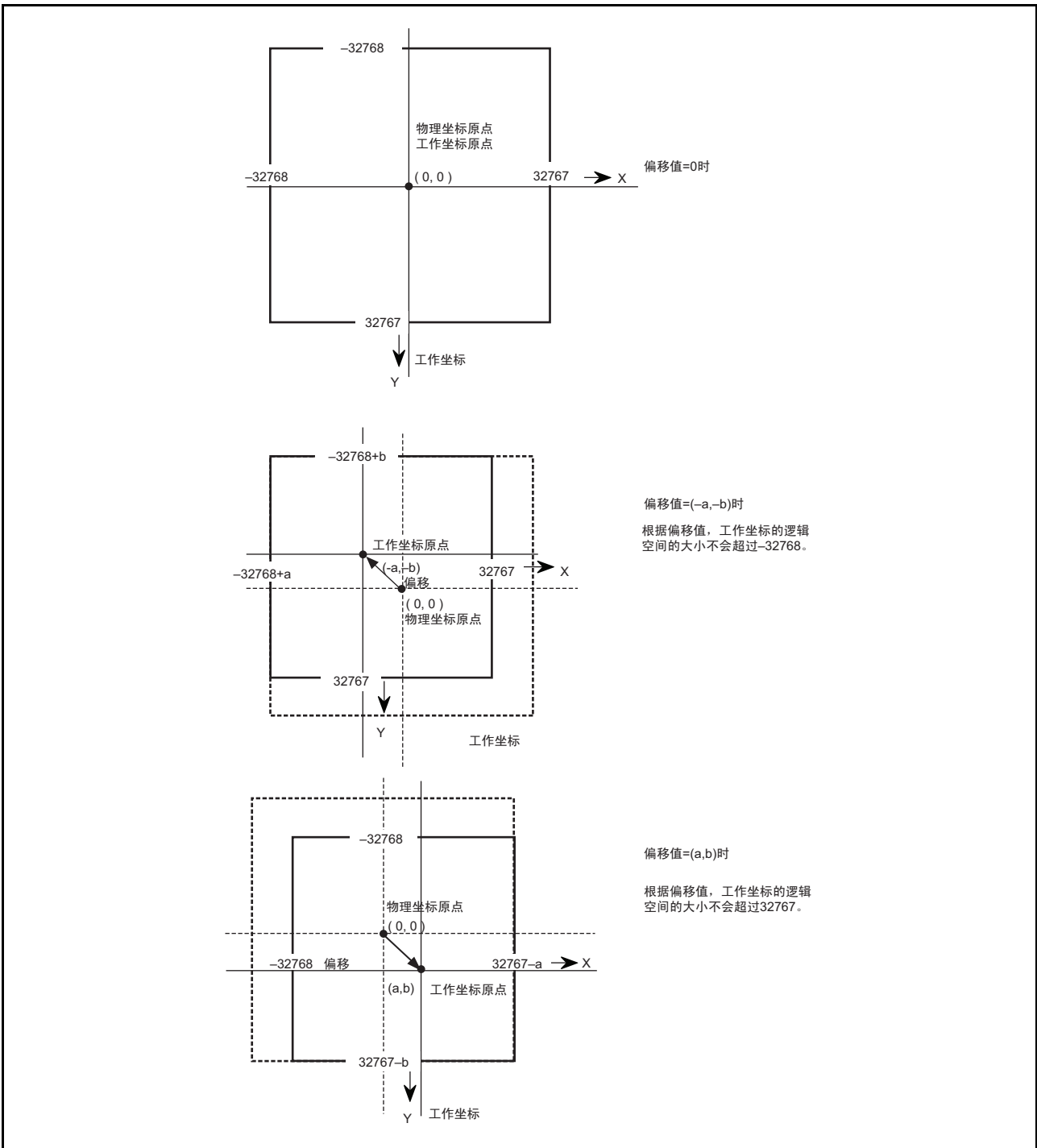


图 23.5 工作坐标

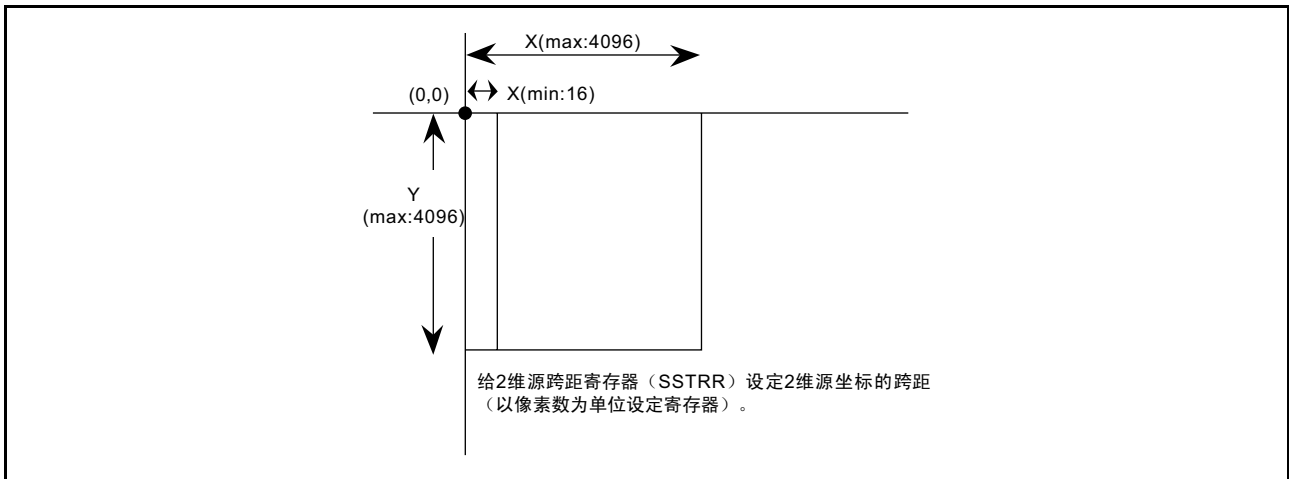


图 23.6 2 维源坐标 (SS=1)

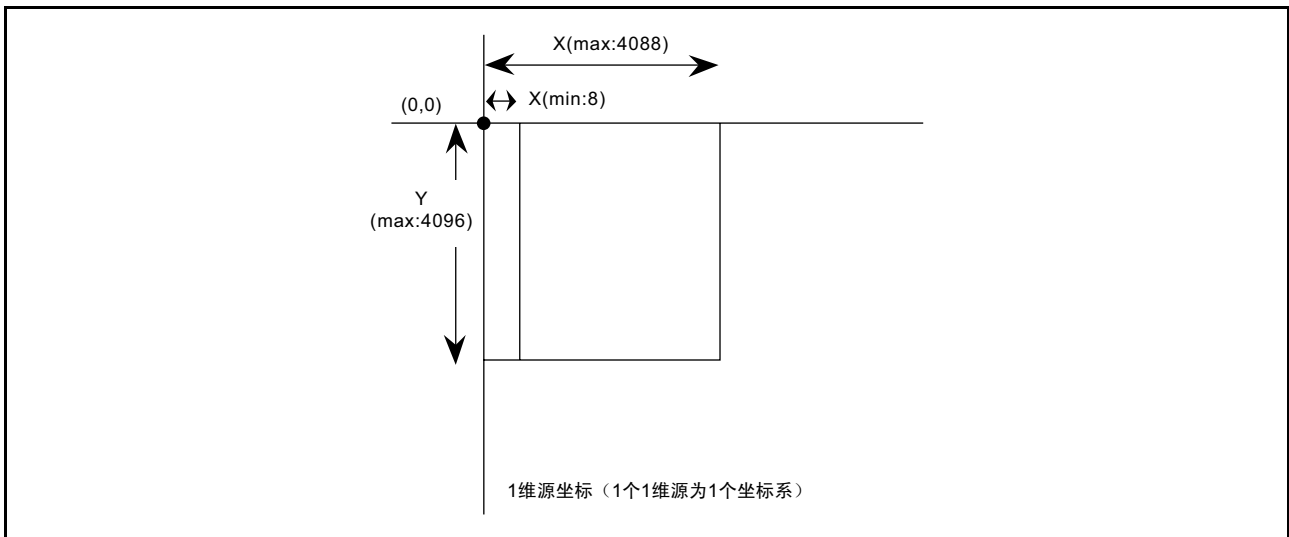
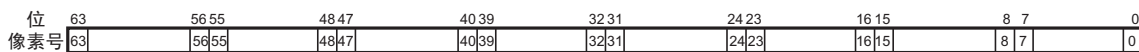


图 23.7 1 维源坐标 (SS=0)

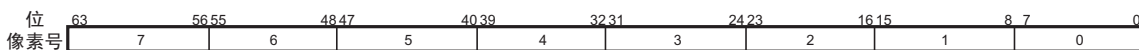
23.1.4 数据格式

●1位/像素数据



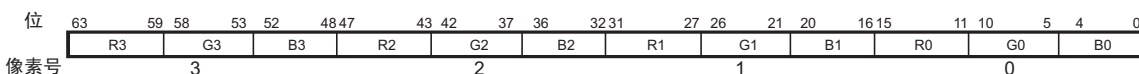
像素号在画面的左侧为0，越向右越大。

●8位/像素数据



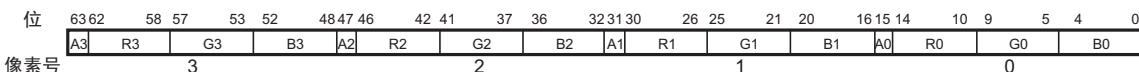
像素号在画面的左侧为0，越向右越大。

●16位/像素数据 (RGB)



像素号在画面的左侧为0，越向右越大。

●16位/像素数据 (ARGB)



像素号在画面的左侧为0，越向右越大。

●32位数据 (显示列表)



23.1.5 绘图属性

(1) 源透明度的指定 (STRANS)

在参照源数据时，能以绘图命令为单位，通过 STRANS 位选择透明或者不透明。在选择透明的情况下，如果绘图控制寄存器 (RCLR) 的源透明色极性位 (STP) 为 0，就在寄存器的值 = 源色时透明；如果 STP 为 1，就在寄存器的值 ≠ 源色时透明。无论 STP 是 1 还是 0，都不描绘此像素。

能使用的命令为 POLYGON4A/4B、LINEA/B、RLINEA/B 和 BITBLTA/B，但是在其他命令的情况下必须将 STRANS 位置 0。在源像素格式为 ARGB 时不比较 A 值。如果将此位置 1，BITBLTA/B 就与 ROP 码无关，总是读源数据。

(2) 目标透明度的指定 (DTRANS)

在参照目标数据时，能以绘图命令为单位，通过 DTRANS 位选择透明或者不透明。在选择透明的情况下，如果绘图控制寄存器 (RCLR) 的目标透明色极性位 (DTP) 为 0，就在寄存器的值 = 目标色时透明；如果 DTP 为 1，就在寄存器的值 ≠ 目标色时透明。无论 DTP 是 1 还是 0，都不描绘此像素。

能使用的命令为 BITBLTA/B/C，但是在其他命令的情况下必须将 DTRANS 位置 0。在目标像素格式为 ARGB 时不比较 A 值。如果将此位置 1，就与 ROP 码无关，总是读目标数据。

(3) 源类型的指定 (STYLE)

能以绘图命令为单位，通过 STYLE 位选择是扩大和缩小源数据还是重复参照源数据。在不指定类型时，根据绘图区的大小扩大和缩小源数据；在指定类型时，根据绘图区的大小重复参照源数据。因此，在重复描绘“口”字图案时，指定此属性。

能使用的命令为 POLYGON4A/4B、LINEA/B 和 RLINER/B，但是在其他命令的情况下必须将 STYLE 位置 0。在将 LINEA/B、RLINER/B 和 POLYGON4A/4B 的 BLKE 设定为 1 时，必须设定 STYLE=1。

在使用 LINEA/B 或者 RLINER/B 命令时，只重复参照源的 X 方向的数据，根据线宽扩大和缩小源的 Y 方向的数据。

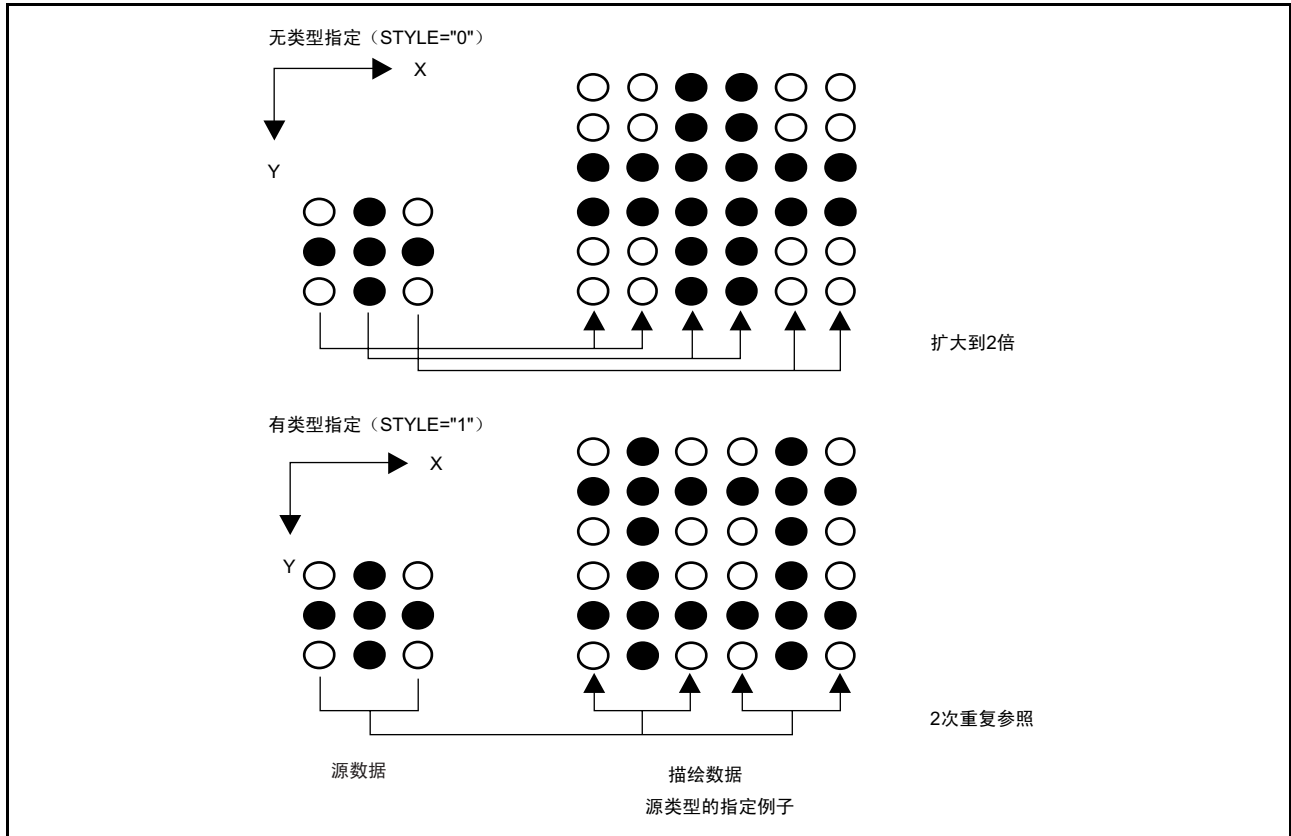


图 23.8 源类型的指定例子

(4) 裁剪的指定 (CLIP)

能进行裁剪区的管理。裁剪区有系统裁剪区、用户裁剪区和相对用户裁剪区。

系统裁剪区是固定绘图范围的区域。系统裁剪区与属性的指定无关，总是有效。

能在系统裁剪区内任意设定用户裁剪区，并且能以绘图命令为单位，通过绘图属性的 CLIP 位选择是否在此区域内进行裁剪。描绘边界。不加上 LCOFS 或者 RLCOFS 命令设定的局部偏移值。设定值必须为 $XMIN < XMAX$ 和 $YMIN < YMAX$ 。

根据屏幕坐标设定裁剪。接通电源后的裁剪范围不确定，所以必须通过 WPR 命令在最初执行的显示列表的开头处设定裁剪区，并使 $XMAX$ 小于目标跨距的设定值。

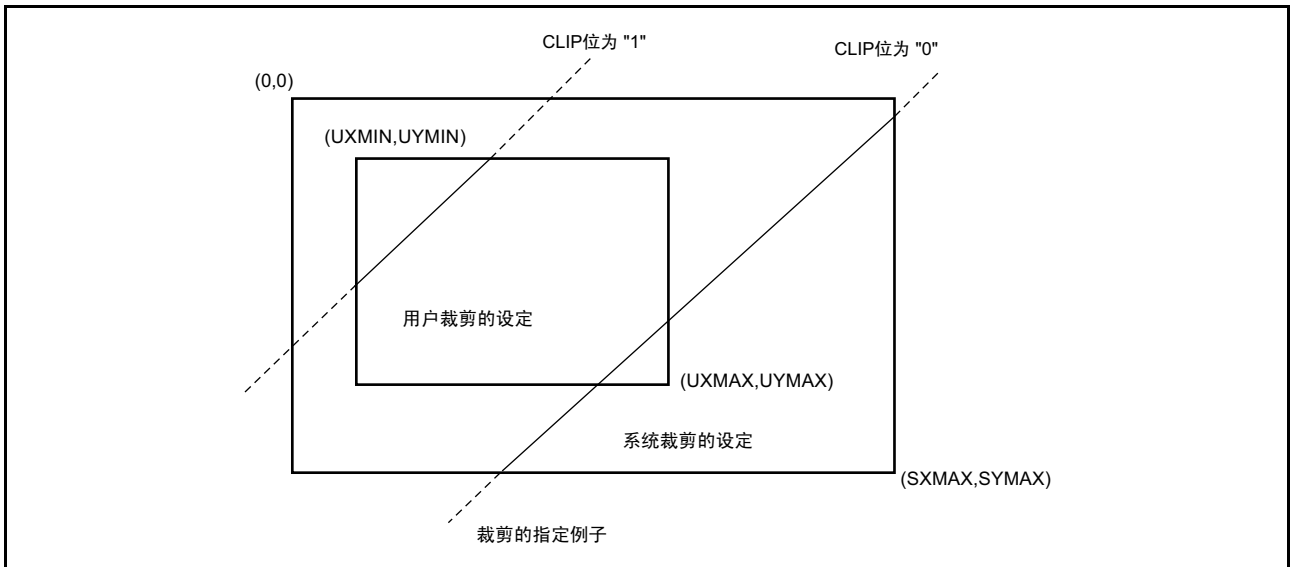


图 23.9 裁剪的指定例子

(5) 相对裁剪的指定 (RCLIP)

能进行裁剪区的管理。裁剪区有系统裁剪区、用户裁剪区和相对用户裁剪区。

系统裁剪区是固定绘图范围的区域。系统裁剪区与属性的指定无关，总是有效。

能通过局部偏移的相对指定，在系统裁剪区内任意设定相对用户裁剪区，并且能以绘图命令为单位，通过绘图属性 RCLIP 位选择是否在此区域内进行裁剪。描绘边界。不加上 LCOFS 或者 RLCOFS 命令设定的局部偏移值。设定值必须为 $XMIN < XMAX$ 和 $YMIN < YMAX$ 的值。

根据屏幕坐标设定裁剪。接通电源后的裁剪范围不确定，所以必须通过 WPR 命令在最初执行的显示列表的开头处设定裁剪区，使 XMAX 小于目标跨距的设定值。如果同时将 RCLIP 和 CLIP 位置 1，就描绘 2 个裁剪区重叠的区域。

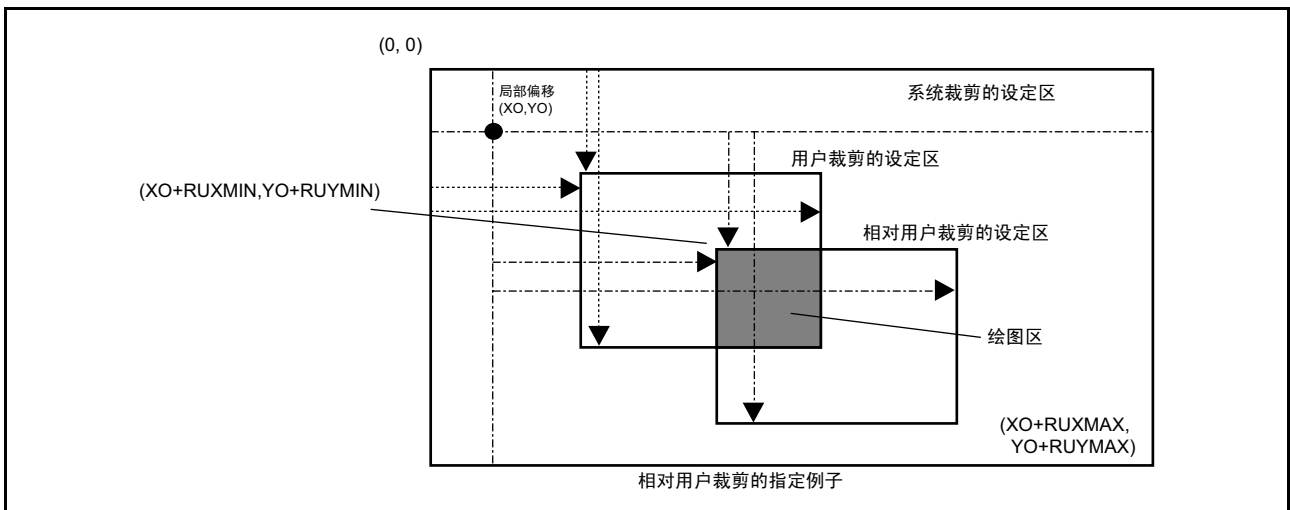


图 23.10 相对用户裁剪的指定例子

在相对用户裁剪区 (XO+RUXMIN, YO+RUYMIN) - (XO+RUXMAX, YO+RUYMAX) 和系统裁剪区重叠时, 进行以下饱和处理:

$$\begin{aligned} XO+RUXMIN < 0 &\rightarrow XO+RUXMIN=0 \\ XO+RUXMAX > SXMAX &\rightarrow XO+RUXMAX=SXMAX \\ YO+RUYMIN < 0 &\rightarrow YO+RUYMIN=0 \\ YO+RUYMAX > SYMAX &\rightarrow YO+RUYMAX=SYMAX \end{aligned}$$

【注】 在设定局部偏移和相对用户裁剪区时, 不能超过以下范围:

$$\begin{aligned} -4096 &\leq XO+RUXMIN \leq 4095 \\ -4096 &\leq YO+RUYMIN \leq 4095 \\ 0 &\leq XO+RUXMAX \leq 8191 \\ 0 &\leq YO+RUYMAX \leq 8191 \end{aligned}$$

在 RCLIP=1 时, 如果相对用户裁剪区满足以下任意 1 个条件, 相对用户裁剪区就在 G2D 内部无效。

(与 RCLIP=0 时操作相同)

$$\begin{aligned} 4095 &< XO+RUXMIN \\ 4095 &< YO+RUYMIN \\ XO+RUXMAX &< 0 \\ YO+RUYMAX &< 0 \end{aligned}$$

(6) NET 描绘的指定 (NET)

能以绘图命令为单位, 通过 NET 位选择是否进行 NET 描绘。NET 描绘是一种描绘功能, 即只描绘绘图坐标的 X+Y=EOS (0: 偶数; 1: 奇数) 为真的坐标的像素。例如: 如果 EOS=0, 就只描绘 Y=0、X=0、2、4、6、8...、Y=1、X=1、3、5、7、9... 的坐标。

通过此功能, 能将描绘的图形和底层相互重叠一半。

能使用的命令为 POLYGON4 类、LINEA/B/C 和 RLINEA/B/C, 但是在其他命令的情况下必须将 NET 位置 0。禁止与抗锯齿允许 (AA) 位一起使用。

(7) 偶数 / 奇数选择的指定 (EOS)

在 EOS 位 =0 时, 选择偶数像素; 在 EOS 位 =1 时, 选择奇数像素。

和 NET 描绘的指定 (NET) 一起使用。在使用 LINEWC 或者 RLINEWC 命令时, 如果 EOS 位 =0, 就在工作坐标中用 0 进行描绘; 如果 EOS 位 =1, 就在工作坐标中用 1 进行描绘。

(8) 工作指定 (WORK)

如果通过 POLYGON4 类或者 BITBLT 类命令在绘图坐标中进行描绘, 就能以绘图命令为单位, 通过 WORK 位选择是否参照 2 值工作数据。

在选择了参照 2 值工作数据时, 如果对应绘图坐标的像素工作数据为 1, 就进行描绘; 如果为 0, 就不描绘。因此, 就能与在工作坐标中描绘图形相同的形式, 在绘图坐标中进行描绘。在工作坐标中进行描绘时, 能选择是通过 FTRAPC、RFTRAPC、LINEWC、RLINEWC 和 CLRWC 命令还是通过 CPU 进行描绘。不能同时通过命令和 CPU 进行存储器绘图的存取。能使用的命令为 POLYGON4 类和 BITBLT 类, 但是在其他命令的情况下必须将 WORK 位置 0。

(9) 源地址的指定 (SS)

能通过 SS 位选择是从 2 维源区域参照源还是从显示列表的 BaseAddress 参数所示的地址参照源。能使用的命令为 POLYGON4A/B 和 BITBLTA/B, 在其他命令的情况下必须将 SS 位置 0。如果设定偏移值, 就从 (TXOFS, TYOFS) 参照源。

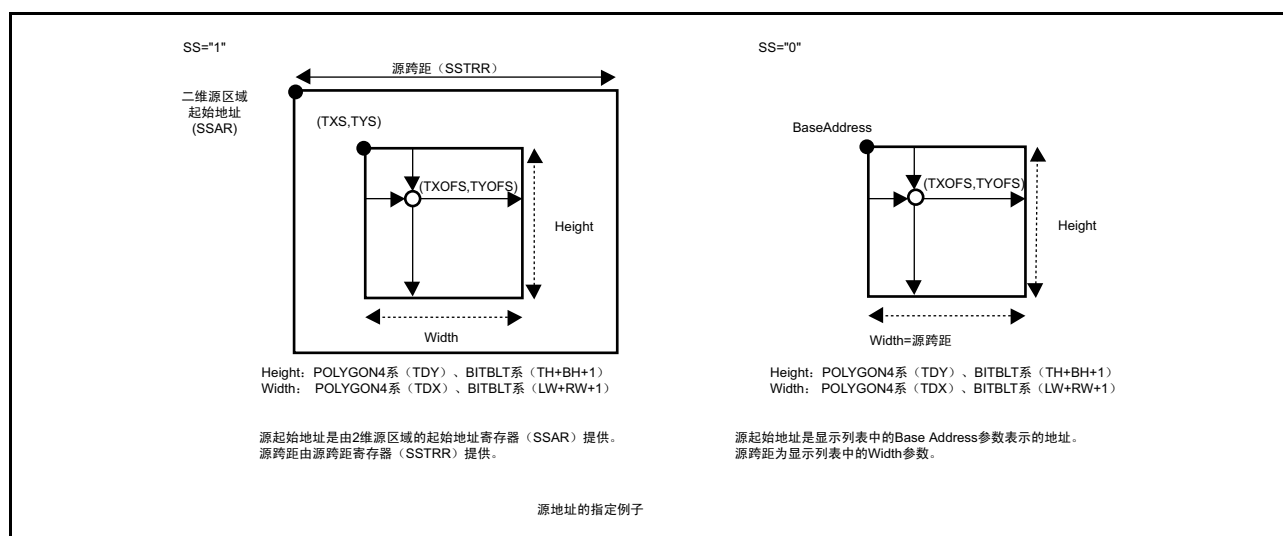


图 23.11 源地址的指定例子

【注】 在 $SS=1$ 时，必须在 $0 \leq TXS \leq SSTR - Width(TDX, LW+RW+1)$ 、 $0 \leq TYS \leq 4096 - Height(TDY, TH+BH+1)$ 的范围内进行设定。

(10) 源坐标相对地址的指定 (REL)

能通过将 POLYGON4A/4B、BITBLTA/B、LINEA/B、RLINEA/B、JUMP 和 GOSUB 命令中的 REL 位置 1，在命令码的前后方进行相对地址的源参照或者转移。在 POLYGON4A 和 BITBLTA 的情况下，必须将 SS 位置 0。在 SS 位为 1 时，不保证运行。

命令码的地址为相对地址的原点（长字地址）。

【注】 在 POLYGON4A/4B、BITBLTA/B、LINEA/B 和 RLINEA/B 命令时，配置命令码的地址（长字：32 位单位）+ 源起始相对地址（长字：32 位单位）必须为 4 个字地址（64 位单位）。

(11) 边缘的描绘 (EDG)

能通过将 FTRAP 和 RFTRAP 命令中的 EDG 位置 1，在对工作区描绘多边形后描绘边缘线。此时，通过 EOS 位指定是用 0 还是用 1 描绘边缘线。

(12) 颜色偏移 (COOF)

能用于 POLYGON4 类、LINEA/B/C、RLINEA/B/C 和 BITBLT 类命令。在目标为 16 位 / 像素时，如果将绘图属性 COOF 位置 1，就给源数据（2 值源为颜色展开后的数据，单色指定时为指定的颜色）加上 COFSR 寄存器的值，然后描绘所得的结果。在 8 位 / 像素时，必须将 COOF 位置 0。在源像素格式为 ARGB 时，A 值不进行运算。

(13) 源方向 X、Y(SRCDIRX, SRCDIRY)

能用于 BITBLTA/B 命令，选择源的扫描方向。

(TXS, TYS)、BaseAddress 与源扫描方向无关，指定矩形源的左上方。

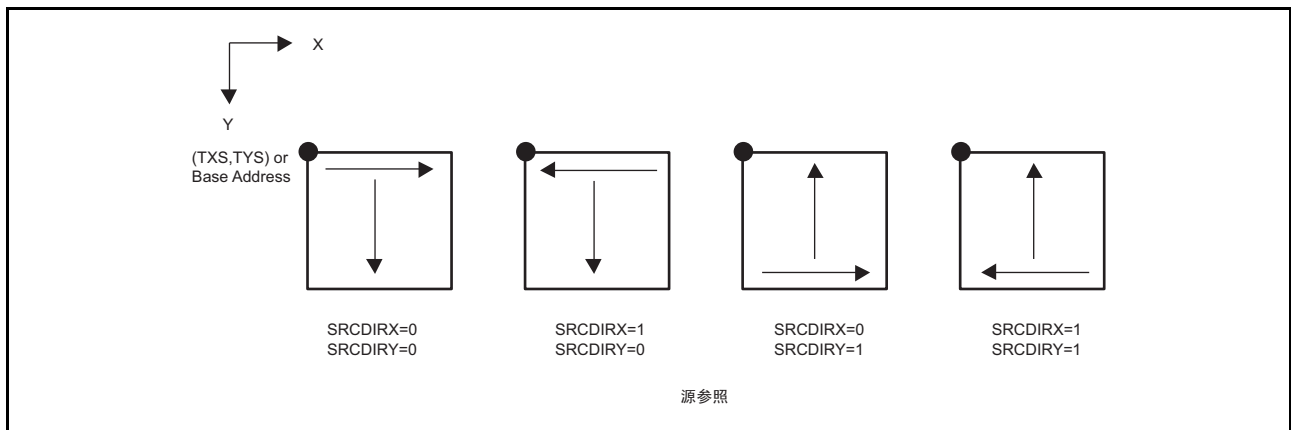


图 23.12 源方向的指定例子

(14) 目标方向 X、Y (DSTDIRX, DSTDIRY)

能用于 BITBLTA/B/C 命令，选择目标的绘图方向。

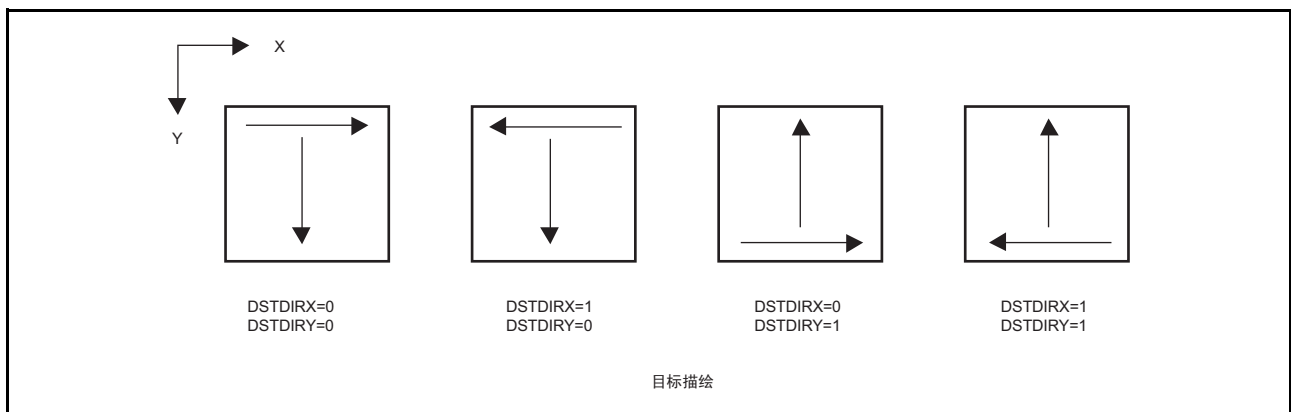


图 23.13 目标方向的指定例子

(15) 抗锯齿允许 (AA)

能用于 LINE 类和 RLINE 类命令，以便减少锯齿。AA 位只在目标为 16 位 / 像素时有效；在 8 位 / 像素时，必须将其置 0。在 LINED 和 RLINED 命令的情况下，必须将 AA 位置 1。不能与 NET 描绘的指定 (NET) 一起使用。

(16) α 混合允许 (αE)

能用于 POLYGON4 类和 BITBLT 类命令。将源数据 (2 值源为颜色展开后的数据，单色指定时为指定的颜色) 和底层数据进行 α 混合以及描绘。 α 值设定在 α 值寄存器 (ALPHR) 中。 αE 位只在目标为 16 位 / 像素时有效，在 8 位 / 像素时必须将其置 0。在 POLYGON4 类命令的情况下，只在 BLKE=1 时有效，在 BLKE=0 时必须将其置 0；在 BITBLT 类命令的情况下，只在 ROP 码 =H'CC (源复制) 时有效，在其他 ROP 码时必须设定 $\alpha E=0$ 。在 ARGB 格式时，不对 A 值进行 α 混合。根据绘图控制寄存器的源 A 值使用位 (SAU) 和 A 值位 (AVALUE) 的设定进行 A 值的描绘。

(17) 源 α 允许 (S α E)

能用于 POLYGON4A 和 BITBLTA 命令，和 α E 配合使用。在 α E=0 时，必须将 S α E 位置 0。在源像素格式 (SPF) 为 1 (ARGB) 时，只有源数据 A 值为 1 的像素才进行 α 混合。0 像素不进行 α 混合，而直接描绘源数据。只在 SPF=1 时有效，在 SPF=0 时必须将 S α E 位置 0。

(18) 块允许 (BLKE)

能用于 POLYGON4 类命令。当 BLKE=1 时，就在内部将输入的顶点坐标 (DX_n, DY_n) 变换为外接矩形的坐标 (DX'_n, DY'_n) 并且进行描绘。在坐标变换时，在内部将变换后的顶点转换为矩形并且进行描绘。这对于在坐标变换后要将图案垂直粘贴时是有效的。BLKE=1 时的绘图方向固定为从左上到右下 (不能上下左右颠倒)。

如果通过 CLRW 命令进行坐标变换，就在内部从输入的左右 X 坐标值和上下 Y 坐标值获取 4 个顶点，并且对 4 个顶点进行坐标变换。然后，在内部将此变换后的 4 个顶点转换为外接矩形并且进行描绘。

如果通过 FTRAPC 和 RFTRAPC 命令进行坐标变换，就在内部从输入的多边形的外接四边形的坐标值获取 4 个顶点，并且对这 4 个顶点进行坐标变换。然后，在内部将此变换后的 4 个顶点转换为外接矩形，并且求得左侧进行描绘。在 CLRW、FTRAPC 和 RFTRAPC 命令的情况下，必须将 BLKE 位置 1。

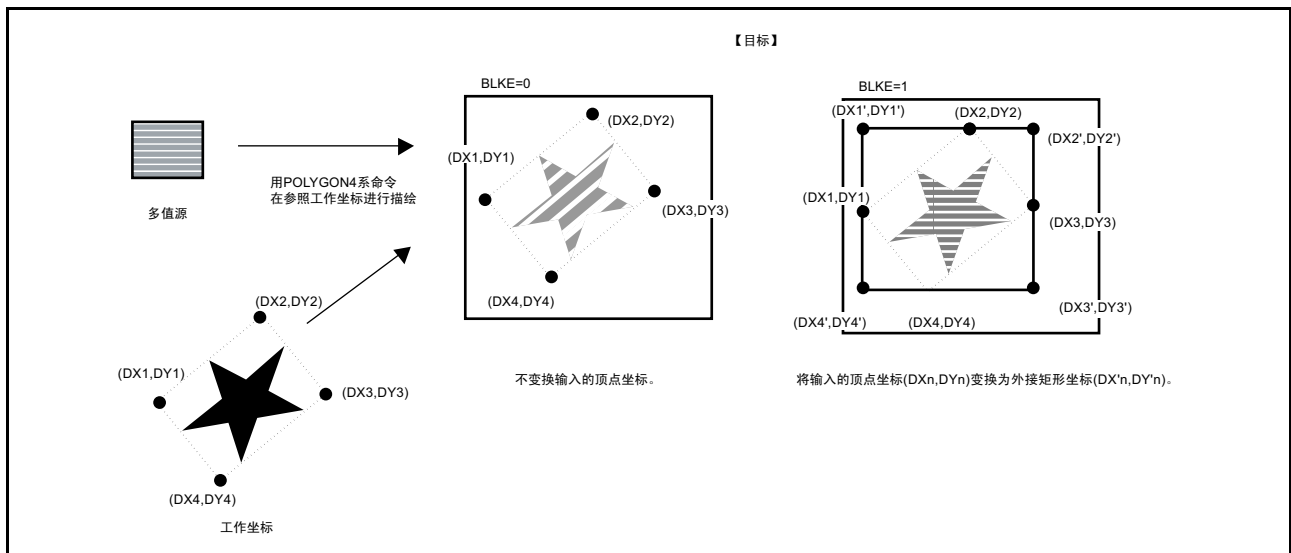


图 23.14 块允许的指定例子

(19) 允许坐标变换 (MTRE)

能用于全部的绘图命令。在坐标变换控制寄存器的坐标变换允许位 (GTE) =1 时，通过将此位置 1 进行输入顶点的坐标变换。

(20) 允许连接的指定 (LINKE)

能用于 LINEC/D、R LINEC/D、FTRAPC、RFTRAPC 和 WPR 命令。通过 LINEC/D、R LINEC/D、FTRAPC、RFTRAPC 命令从 LINK Address 指定的存储器读取顶点坐标，通过 WPR 命令从 LINK Address 指定的存储器读取寄存器的写数据。

由于将 LINK Address 的下一个长字解释为下一个命令码，所以不能将配置 LINK Address 的下一个长字地址指定为 LINK Address 所指的连接目标地址。

(21) 连接地址的相对指定 (LREL)

能用于 LINEC/D、R LINEC/D、FTRAPC、RFTRAPC 和 WPR 命令，和 LINKE 配合使用。在 LINKE=0 时，必须将 LREL 位置 0。将连接目标地址指定为相对地址。命令码的地址为相对地址的原点。

由于将 LINK Address 的下一个长字解释为下一个命令码，所以不能将配置 LINK Address 的下一个长字地址指定为配置命令码的地址 +LINK Address 所指的连接目标地址。

(22) 顺时针方向 (CLKW)

能用于 LINED 和 RLINED 命令。指定 N 顶点的顺序是顺时针还是逆时针方向。在 CLKW=1 时为顺时针方向；在 CLKW=0 时为逆时针方向。

(23) 光栅操作 (ROP)

能用于 BITBLT 类命令。通过 BITBLT 命令参数的 ROP 字段进行指定。

ROP 码	操作
H'00	0
H'11	$\sim(S D)$
H'22	$\sim S \& D$
H'33	$\sim S$
H'44	$S \& \sim D$
H'55	$\sim D$
H'66	$S \wedge D$
H'77	$\sim(S \& D)$
H'88	$S \& D$
H'99	$\sim(S \wedge D)$
H'AA	D
H'BB	$\sim S D$
H'CC	S
H'DD	$S \sim D$
H'EE	$S D$
H'FF	1

在 α 混合允许有效 ($\alpha E=1$) 时，必须设定为 H'CC。在 ARGB 格式时，不对 A 值进行 α 混合以及光栅操作。根据绘图控制寄存器的源 A 值使用位 (SAU) 和 A 值位 (AVALUE) 的设定进行 A 值的描绘。

23.2 显示列表

23.2.1 4 个顶点的屏幕绘图命令

(1) POLYGON4A

(a) 功能

在目标区域中参照多值 (8 或者 16 位 / 像素) 源数据, 进行任意 4 个顶点的描绘。

(b) 命令格式

- SS=1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
OP CODE=1000_0010								Reserve(all 0)								Draw Mode															
0	0	0	0	TXS($0 \leq TXS \leq 4088$)								0	0	0	0	TYS($0 \leq TYS \leq 4095$)															
0	0	0	0	TDX($8 \leq TDX \leq 4095$)								0	0	0	0	TDY($1 \leq TDY \leq 4095$)															
0	0	0	0	TXOFS($0 \leq TXOFS \leq TDX-1$)								0	0	0	0	TYOFS($0 \leq TYOFS \leq TDY-1$)															
符号	DX1($-32768 \leq DX1 \leq 32767$)								符号	DY1($-32768 \leq DY1 \leq 32767$)																					
符号	DX2($-32768 \leq DX2 \leq 32767$)								符号	DY2($-32768 \leq DY2 \leq 32767$)																					
符号	DX3($-32768 \leq DX3 \leq 32767$)								符号	DY3($-32768 \leq DY3 \leq 32767$)																					
符号	DX4($-32768 \leq DX4 \leq 32767$)								符号	DY4($-32768 \leq DY4 \leq 32767$)																					

【注】 $0 \leq TXS \leq SSTRR-TDX$ 、 $0 \leq TYS \leq 4096-TDY$ (SSTRR: 源跨距寄存器的设定值)。

- SS=0 并且 REL=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
OP CODE=1000_0010								Reserve(all 0)								Draw Mode															
0	0	0	Base Address(quad word address)																0	0	0										
0	0	0	0	TDX($8 \leq TDX \leq 4088$)								0	0	0	0	TDY($1 \leq TDY \leq 4095$)															
0	0	0	0	TXOFS($0 \leq TXOFS \leq TDX-1$)								0	0	0	0	TYOFS($0 \leq TYOFS \leq TDY-1$)															
符号	DX1($-32768 \leq DX1 \leq 32767$)								符号	DY1($-32768 \leq DY1 \leq 32767$)																					
符号	DX2($-32768 \leq DX2 \leq 32767$)								符号	DY2($-32768 \leq DY2 \leq 32767$)																					
符号	DX3($-32768 \leq DX3 \leq 32767$)								符号	DY3($-32768 \leq DY3 \leq 32767$)																					
符号	DX4($-32768 \leq DX4 \leq 32767$)								符号	DY4($-32768 \leq DY4 \leq 32767$)																					

- SS=0 并且 REL=1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1000_0010								Reserve(all 0)								Draw Mode															
符号扩展		符号		Base Address(long word address)																0		0									
0		0		TDX($8 \leq TDX \leq 4088$)								0		0		0		0		TDY($1 \leq TDY \leq 4095$)											
0		0		TXOFS($0 \leq TXOFS \leq TDX-1$)								0		0		0		TYOFS($0 \leq TYOFS \leq TDY-1$)													
符号		DX1($-32768 \leq DX1 \leq 32767$)																符号		DY1($-32768 \leq DY1 \leq 32767$)											
符号		DX2($-32768 \leq DX2 \leq 32767$)																符号		DY2($-32768 \leq DY2 \leq 32767$)											
符号		DX3($-32768 \leq DX3 \leq 32767$)																符号		DY3($-32768 \leq DY3 \leq 32767$)											
符号		DX4($-32768 \leq DX4 \leq 32767$)																符号		DY4($-32768 \leq DY4 \leq 32767$)											

【注】 配置命令码的地址（长字：32 位单位）+Base Address（长字：32 位单位）必须为 4 个字地址（64 位单位）。

- CODE
B'10000010
- 绘图属性

参照数据				描绘的坐标系	
多值源	2 值源	2 值工作	指定的颜色	绘图坐标	工作坐标
○		○ (仅 WORK=1)		○	

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	0 固定	CLIP	RCLIP	STRANS	0 固定	WORK	SS	REL	STYLE	BLKE	NET	EOS	COOF	αE	$S\alpha E$

- 命令参数
 - TXS、TYS : 源起始点。给未使用位写 0。
 - Base Address : 源起始绝对地址（4 个字地址，必须给 A31~29 和 A2~A0 写 0。）
源起始相对地址（长字地址，负数为 2 的补码。必须对 A31~A29 用 A28 的符号位进行符号扩展，给 A1~A0 写 0。）

【注】 在 32 位地址模式中，必须将指定的 32 位地址的 bit28~bit3 写到 A28~A3。

- TDX、TDY : 源大小。必须给未使用位写 0。
- DXn、DYn (n=1~4) : 绘图坐标（绝对坐标）。负数为 2 的补码。
- TXOFS、TYOFS : 源偏移。必须给未使用位写 0。

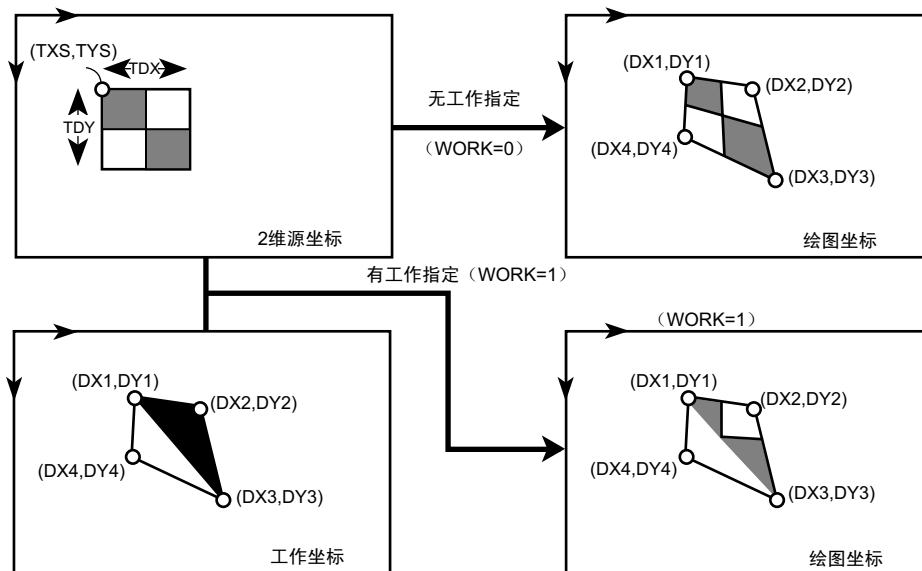
(c) 说明

以任意四边形，将多值（8 或者 16 位 / 像素）源数据传送到绘图坐标。水平扫描源数据，但是在描绘时有时也根据外形进行倾斜扫描。在进行倾斜扫描的描绘时，为了填空需要写 2 次。

在 $SS=0$ 时，必须给 TDX 设定 8 像素的倍数像素数；在 $SS=1$ 时，必须以像素为单位，给 TDX 设定大于等于 8 像素的值。当 TDX 小于 8 像素时，不能正常参照多值源。如果设定 $TXOFS$ 和 $TYOFS$ ，就按偏移量将源数据拆开进行参照。以像素为单位，设定 $TXOFS$ 和 $TYOFS$ 。

1. 当通过绘图属性选择源类型指定 ($STYLE=1$) 时，就不扩大和缩小源数据，而重复参照源数据。
2. 当通过绘图属性选择工作指定 ($WORK=1$) 时，就参照工作坐标（坐标与绘图坐标相同），同时在绘图坐标中描绘工作坐标的像素为 1 的部分。
3. 在 $SS=1$ 时，从 2 维源区域参照源数据；在 $SS=0$ 时，从显示列表的 Base Address 参照源数据。在 $REL=0$ 时，能将源地址指定为绝对地址；在 $REL=1$ 时，能将源地址指定为配置 POLYGON4A 命令的命令码的存储地址的相对地址。
4. 在进行 16 位 / 像素描绘时，如果将绘图属性的 COOF 位置 1，就给多值源数据的值加上 COFSR 寄存器的值，然后描绘所得的结果。此时，通过饱和处理进行运算。在 8 位 / 像素时，必须将 COOF 位置 0。

(d) 例



(2) POLYGON4B

(a) 功能

在目标区域中参照 2 值 (1 位 / 像素) 源数据, 进行任意 4 个顶点的描绘。

(b) 命令格式

- SS=1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1000_0001								Reserve(all 0)								Draw Mode															
Color1								Color0																							
0	0	0	0	TXS($0 \leq TXS \leq 4088$)								0	0	0	0	TYS($0 \leq TYS \leq 4095$)															
0	0	0	0	TDX($8 \leq TDX \leq 4088$)				0	0	0	0	0	0	0	TDY($1 \leq TDY \leq 4095$)																
0	0	0	0	TXOFS($0 \leq TXOFS \leq TDX-1$)								0	0	0	0	TYOFS($0 \leq TYOFS \leq TDY-1$)															
符号	DX1($-32768 \leq DX1 \leq 32767$)								符号	DY1($-32768 \leq DY1 \leq 32767$)																					
符号	DX2($-32768 \leq DX2 \leq 32767$)								符号	DY2($-32768 \leq DY2 \leq 32767$)																					
符号	DX3($-32768 \leq DX3 \leq 32767$)								符号	DY3($-32768 \leq DY3 \leq 32767$)																					
符号	DX4($-32768 \leq DX4 \leq 32767$)								符号	DY4($-32768 \leq DY4 \leq 32767$)																					

【注】 $0 \leq TXS \leq SSTRR-TDX$ 、 $0 \leq TYS \leq 4096-TDY$ (SSTRR: 源跨距寄存器的设定值)。

- SS=0 并且 REL=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1000_0001								Reserve(all 0)								Draw Mode															
Color1								Color0																							
0	0	0	Base Address(quad word address)												0	0	0														
0	0	0	0	TDX($8 \leq TDX \leq 4088$)				0	0	0	0	0	0	0	0	TDY($1 \leq TDY \leq 4095$)															
0	0	0	0	TXOFS($0 \leq TXOFS \leq TDX-1$)								0	0	0	0	TYOFS($0 \leq TYOFS \leq TDY-1$)															
符号	DX1($-32768 \leq DX1 \leq 32767$)								符号	DY1($-32768 \leq DY1 \leq 32767$)																					
符号	DX2($-32768 \leq DX2 \leq 32767$)								符号	DY2($-32768 \leq DY2 \leq 32767$)																					
符号	DX3($-32768 \leq DX3 \leq 32767$)								符号	DY3($-32768 \leq DY3 \leq 32767$)																					
符号	DX4($-32768 \leq DX4 \leq 32767$)								符号	DY4($-32768 \leq DY4 \leq 32767$)																					

- SS=0 并且 REL=1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1000_0001								Reserve(all 0)								Draw Mode															
Color1																Color0															
符号扩展		符号		Base Address(long word address)																								0		0	
0 0 0 0				TDX(8 ≤ TDX ≤ 4088)								0 0 0 0				TDY(1 ≤ TDY ≤ 4095)															
0 0 0 0				TXOFS(0 ≤ TXOFS ≤ TDX-1)								0 0 0 0				TYOFS(0 ≤ TYOFS ≤ TDY-1)															
符号		DX1(-32768 ≤ DX1 ≤ 32767)														符号		DY1(-32768 ≤ DY1 ≤ 32767)													
符号		DX2(-32768 ≤ DX2 ≤ 32767)														符号		DY2(-32768 ≤ DY2 ≤ 32767)													
符号		DX3(-32768 ≤ DX3 ≤ 32767)														符号		DY3(-32768 ≤ DY3 ≤ 32767)													
符号		DX4(-32768 ≤ DX4 ≤ 32767)														符号		DY4(-32768 ≤ DY4 ≤ 32767)													

【注】 配置命令码的地址（长字：32 位单位）+Base Address（长字：32 位单位）必须为 4 个字地址（64 位单位）。

1. CODE
B'10000001
2. 绘图属性

参照数据				描绘的坐标系	
多值源	2 值源	2 值工作	指定的颜色	绘图坐标	工作坐标
	○	○ (仅 WORK=1)		○	

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	0 固定	CLIP	RCLIP	STRANS	0 固定	WORK	SS	REL	STYLE	BLKE	NET	EOS	COOF	αE	0 固定

3. 命令参数
 - TXS、TYS : 源起始点。给未使用位写0。
 - Base Address : 源起始绝对地址（4个字地址，必须给A31~29和A2~A0写0。）
源起始相对地址（长字地址，负数为2的补码。必须对A31~A29用A28的符号位进行符号扩展，给A1~A0写0。）

【注】 在 32 位地址模式中，必须将指定的 32 位地址的 bit28 ~ bit3 写到 A28 ~ A3。

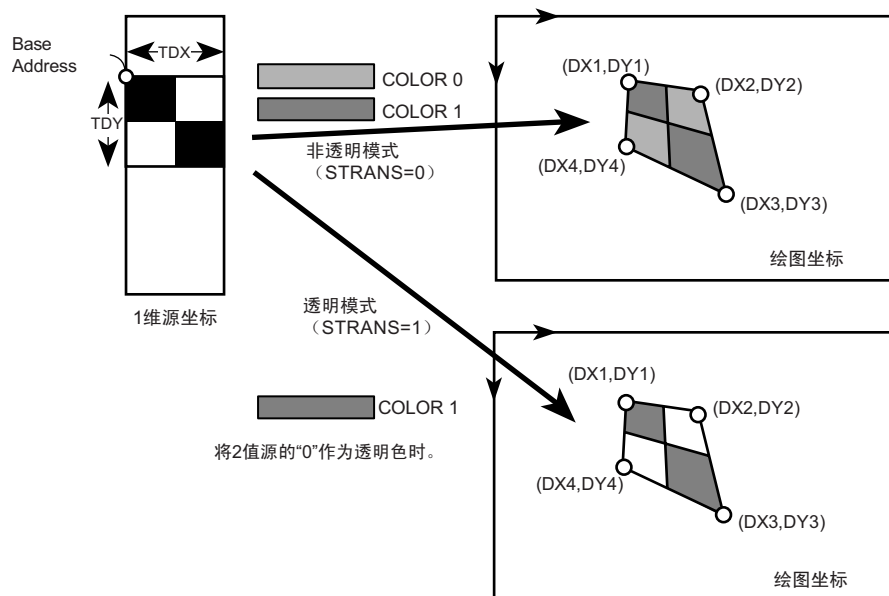
- TDX、TDY : 源大小。必须给未使用位写0。
- DXn、DYn (n=1~4) : 绘图坐标（绝对坐标）。负数为2的补码。
- TXOFS、TYOFS : 源偏移。必须给未使用位写0。
- COLOR0、COLOR1 : 指定8或者16位/像素颜色。对于16位/像素，必须符合目标像素格式。在8位/像素时，必须给高位和低位字节设定相同的值。

(c) 说明

在绘图区域中，以任意四边形以及参数 **COLOR0** 和 **COLOR1** 指定的颜色，描绘 2 值（1 位 / 像素）源数据。对于 8 位 / 像素时的颜色指定（**COLOR0**、1），必须给高位和低位字节设定相同的 8 位数据。水平扫描源数据，但是在描绘时有时也根据外形进行倾斜扫描。在进行倾斜扫描的描绘时，为了填空需要写 2 次。与 **SS** 位的值无关，必须给 **TDX** 设定 8 像素的倍数。如果设定 **TXOFS** 和 **TYOFS**，就按偏移量将源数据拆开进行参照。以像素为单位，设定 **TXOFS** 和 **TYOFS**。

1. 当通过绘图属性选择源类型指定（**STYLE=1**）时，就不扩大和缩小源数据，而重复参照源数据。
2. 当通过绘图属性选择工作指定（**WORK=1**）时，就参照工作坐标（坐标与绘图坐标相同），同时在绘图坐标中描绘工作坐标的像素为 1 的部分。
3. 在 **REL=0** 时，能将源地址指定为绝对地址；在 **REL=1** 时，能将源地址指定为配置 **POLYGON4B** 命令的命令码的存储地址的相对地址。

(d) 例



(3) POLYGON4C

(a) 功能

在绘图坐标中，通过单色的指定进行任意 4 个顶点的描绘。

(b) 命令格式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
OP CODE=1000_0000								Reserve(all 0)								Draw Mode																	
all0								Color																									
符号	DX1(-32768 ≤ DX1 ≤ 32767)																符号	DY1(-32768 ≤ DY1 ≤ 32767)															
符号	DX2(-32768 ≤ DX2 ≤ 32767)																符号	DY2(-32768 ≤ DY2 ≤ 32767)															
符号	DX3(-32768 ≤ DX3 ≤ 32767)																符号	DY3(-32768 ≤ DY3 ≤ 32767)															
符号	DX4(-32768 ≤ DX4 ≤ 32767)																符号	DY4(-32768 ≤ DY4 ≤ 32767)															

1. CODE

B'10000000

2. 绘图属性

参照数据				描绘的坐标系	
多值源	2 值源	2 值工作	指定的颜色	绘图坐标	工作坐标
		○ (仅 WORK=1)	○	○	

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	0 固定	CLIP	RCLIP	0 固定	0 固定	WORK	0 固定	0 固定	0 固定	BLKE	NET	EOS	COOF	αE	0 固定

3. 命令参数

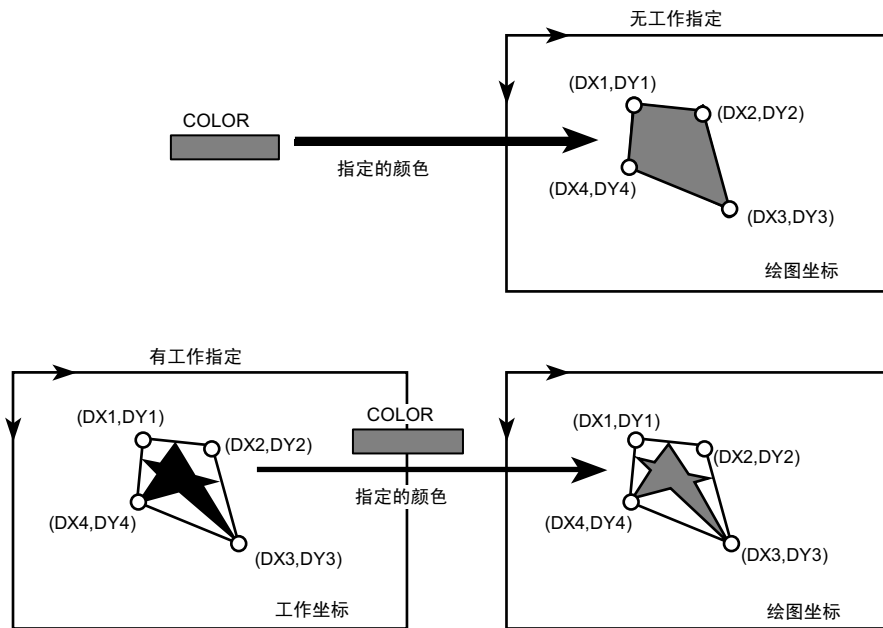
DX_n、DY_n (n=1~4)：绘图坐标（绝对坐标）。负数为 2 的补码。

COLOR：指定 8 或者 16 位/像素颜色。对于 16 位/像素，必须符合目标像素格式。在 8 位/像素时，必须给高位和低位字节设定相同的值。

(c) 说明

在绘图区域中，以 COLOR 参数指定的单色，描绘任意四边形。当通过绘图属性选择工作指定（WORK=1）时，就参照工作坐标（坐标与绘图坐标相同），同时在绘图坐标中描绘工作坐标的像素为 1 的部分。

(d) 例



23.2.2 线的描绘

(1) LINEA

(a) 功能

在目标区域中参照多值（8 或者 16 位 / 像素）源数据，进行任意粗细的折线描绘。

(b) 命令格式

- REL=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1011_0010								Reserve(all 0)								Draw Mode															
0	0	0	Base Address(quad word address)														0	0	0												
0	0	0	0	TDX($8 \leq TDX \leq 4088$)								0	0	0	0	0	0	0	0	TDY($1 \leq TDY \leq 4095$)											
0	0	0	0	TXOFS($0 \leq TXOFS \leq TDX-1$)								n($2 \leq n \leq 65535$)																			
Reserve(all 0)								0	0	0	0	0	0	0	0	0	0	0	0	0	0	W($0,2 \leq W \leq 63$)									
符号	DX1($-32768 \leq DX1 \leq 32767$)														符号	DY1($-32768 \leq DY1 \leq 32767$)															
符号	.														符号	.															
符号	.														符号	.															
符号	DXn($-32768 \leq DXn \leq 32767$)														符号	DYn($-32768 \leq DYn \leq 32767$)															

- 【注】 1. 在 W=0 时，必须设定 TDY=1。
2. 在 n=0、1 时，不保证运行。

- REL=1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1011_0010								Reserve(all 0)								Draw Mode															
符号扩展	符号	Base Address(long word address)														0	0														
0	0	0	0	TDX($8 \leq TDX \leq 4088$)								0	0	0	0	0	0	0	0	TDY($1 \leq TDY \leq 4095$)											
0	0	0	0	TXOFS($0 \leq TXOFS \leq TDX-1$)								n($2 \leq n \leq 65535$)																			
Reserve(all 0)								0	0	0	0	0	0	0	0	0	0	0	0	0	0	W($0,2 \leq W \leq 63$)									
符号	DX1($-32768 \leq DX1 \leq 32767$)														符号	DY1($-32768 \leq DY1 \leq 32767$)															
符号	.														符号	.															
符号	.														符号	.															
符号	DXn($-32768 \leq DXn \leq 32767$)														符号	DYn($-32768 \leq DYn \leq 32767$)															

- 【注】 1. 配置命令码的地址（长字：32 位单位）+Base Address（长字：32 位单位）必须为 4 个字地址（64 位单位）。
2. 在 W=0 时，必须设定 TDY=1。
3. 在 n=0、1 时，不保证运行。

1. CODE
B'10110010
2. 绘图属性

参照数据				描绘的坐标系	
多值源	2 值源	2 值工作	指定的颜色	绘图坐标	工作坐标
○				○	

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	0 固定	CLIP	RCLIP	STRANS	0 固定	0 固定	SS(0)	REL	STYLE(1)	0 固定	NET	EOS	COOF	AA	0 固定

【注】 SS : 必须置 0。

STYLE: 必须置 1。

3. 命令参数

Base Address : 源起始绝对地址 (4 个字地址, 必须给 A31 ~ 29 和 A2 ~ A0 写 0。)
源起始相对地址 (长字地址, 负数为 2 的补码。必须对 A31 ~ A29 用 A28 的符号位进行符号扩展, 给 A1 ~ A0 写 0。)

【注】 在 32 位地址模式中, 必须将指定的 32 位地址的 bit28 ~ bit3 写到 A28 ~ A3。

TDX、TDY : 源大小。必须给未使用位写 0。

TXOFS : 源偏移。必须给未使用位写 0。

n (n=2 ~ 65535) : 顶点数。

W : 线宽。用 6 位整数进行设定, 必须给未使用位写 0。
如果将 W 置 0, 就描绘线宽为 1 的折线。禁止将 W 置 "1"。

DXn (n=2 ~ 65535) : 绘图坐标 (绝对坐标)。负数为 2 的补码。

DYn (n=2 ~ 65535) : 绘图坐标 (绝对坐标)。负数为 2 的补码。

(c) 说明

描绘从顶点 1(DX1, DY1) 到顶点 n(DXn, DYn) 的折线 (经过顶点 2(DX2, DY2)、...、顶点 n-1(DXn-1, DYn-1))。必须给 TDX 设定 8 像素的倍数值。如果设定 TXOFS, 就按偏移量将源数据拆开进行参照。以像素为单位, 设定 TXOFS。只在源的 X 方向重复 STYLE 的图案, 在 Y 方向按照线宽进行扩大和缩小。

如果给 W 设定大于 1 的值, 就能描绘粗线。

- 【注】
1. 对线宽为 1 的线用 8 点进行描绘, 对粗线用 8 点和 4 点进行描绘。
 2. 描绘每个线段的终点。在每个线段的起点和终点坐标相同时, 对线宽为 1 的线描绘 1 个点, 对粗线不进行任何描绘。
 3. 在 AA=1 时
 - 虚线时, 对虚线的间隙不进行抗锯齿处理。
 - 在每个线段的起点 / 终点坐标相同时, 不进行抗锯齿处理。
 - 对水平、垂直和倾斜 45 度的线段不进行抗锯齿处理。

(2) LINEB

(a) 功能

在目标区域中参照 2 值 (1 位 / 像素) 源数据, 进行任意粗细的折线描绘。

(b) 命令格式

- REL=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
OP CODE=1011_0001								Reserve(all 0)								Draw Mode																			
Color1																Color0																			
0	0	0	Base Address(quad word address)																								0	0	0						
0	0	0	0	TDX($8 \leq TDX \leq 4088$)								0	0	0	0	0	0	0	0	TDY($1 \leq TDY \leq 4095$)															
0	0	0	0	TXOFS($0 \leq TXOFS \leq TDX-1$)								n($2 \leq n \leq 65535$)																							
Reserve(all 0)																0	0	0	0	0	0	0	0	0	0	0	0	W($0,2 \leq W \leq 63$)							
符号	DX1($-32768 \leq DX1 \leq 32767$)																符号	DY1($-32768 \leq DY1 \leq 32767$)																	
符号	.																符号	.																	
符号	.																符号	.																	
符号	DXn($-32768 \leq DXn \leq 32767$)																符号	DYn($-32768 \leq DYn \leq 32767$)																	

- 【注】 1. 在 W=0 时, 必须设定 TDY=1。
2. 在 n=0、1 时, 不保证运行。

- REL=1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
OP CODE=1011_0001								Reserve(all 0)								Draw Mode																			
Color1																Color0																			
符号扩展	符号	Base Address(long word address)																								0	0								
0	0	0	0	TDX($8 \leq TDX \leq 4088$)								0	0	0	0	0	0	0	0	TDY($1 \leq TDY \leq 4095$)															
0	0	0	0	TXOFS($0 \leq TXOFS \leq TDX-1$)								n($2 \leq n \leq 65535$)																							
Reserve(all 0)																0	0	0	0	0	0	0	0	0	0	0	0	W($0,2 \leq W \leq 63$)							
符号	DX1($-32768 \leq DX1 \leq 32767$)																符号	DY1($-32768 \leq DY1 \leq 32767$)																	
符号	.																符号	.																	
符号	.																符号	.																	
符号	DXn($-32768 \leq DXn \leq 32767$)																符号	DYn($-32768 \leq DYn \leq 32767$)																	

- 【注】 1. 配置命令码的地址 (长字: 32 位单位) +Base Address (长字: 32 位单位) 必须为 4 个字地址 (64 位单位)。
2. 在 W=0 时, 必须设定 TDY=1。
3. 在 n=0、1 时, 不保证运行。

1. CODE
B'10110001
2. 绘图属性

参照数据				描绘的坐标系	
多值源	2 值源	2 值工作	指定的颜色	绘图坐标	工作坐标
	○			○	

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	0 固定	CLIP	RCLIP	STRANS	0 固定	0 固定	SS(0)	REL	STYLE(1)	0 固定	NET	EOS	COOF	AA	0 固定

【注】 SS : 必须置 0。

STYLE: 必须置 1。

3. 命令参数

Color0、Color1 : 指定 8 或者 16 位/像素颜色。对于 16 位/像素, 必须符合目标像素格式。在 8 位/像素时, 必须给高位和低位字节设定相同的值。

Base Address : 源起始绝对地址 (4 个字地址, 必须给 A31 ~ 29 和 A2 ~ A0 写 0。)
源起始相对地址 (长字地址, 负数为 2 的补码。必须对 A31 ~ A29 用 A28 的符号位进行符号扩展, 给 A1 ~ A0 写 0。)

【注】 在 32 位地址模式中, 必须将指定的 32 位地址的 bit28 ~ bit3 写到 A28 ~ A3。

TDX、TDY : 源大小。必须给未使用位写 0。

TXOFS : 源偏移。必须给未使用位写 0。

n (n=2 ~ 65535) : 顶点数。

W : 线宽。用 6 位整数进行设定, 必须给未使用位写 0。
如果给 W 置 0, 就描绘线宽为 1 的折线。禁止将 W 置 “1”。

DXn (n=2 ~ 65535) : 绘图坐标 (绝对坐标), 负数为 2 的补码。

DYn (n=2 ~ 65535) : 绘图坐标 (绝对坐标), 负数为 2 的补码。

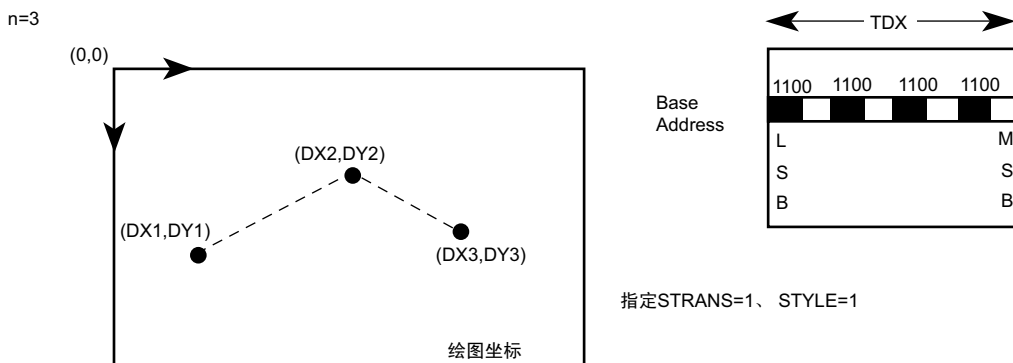
(c) 说明

描绘从顶点 1(DX1, DY1) 到顶点 n(DXn, DYn) 的折线 (经过顶点 2(DX2, DY2)、...、顶点 n-1(DXn-1, DYn-1))。必须给 TDX 设定 8 像素的倍数值。如果设定 TXOFS, 就按偏移量将源数据拆开进行参照。以像素为单位, 设定 TXOFS。只在源的 X 方向重复 STYLE 的图案, 在 Y 方向根据线宽进行扩大和缩小。

如果给 W 设定大于 1 的值, 就能描绘粗线。

- 【注】
1. 对线宽为 1 的线用 8 点进行描绘, 对粗线用 8 点和 4 点进行描绘。
 2. 描绘每个线段的终点。在每个线段的起点/终点坐标相同时, 对线宽为 1 的线描绘 1 个点, 对粗线不进行任何描绘。
 3. 在 AA=1 时
 - 虚线时, 对虚线的间隙不进行抗锯齿处理。
 - 在每个线段的起点/终点坐标相同时, 不进行抗锯齿处理。
 - 对水平、垂直和倾斜 45 度的线段不进行抗锯齿处理。

(d) 例



(3) LINEC

(a) 功能

在目标区域中，通过单色的指定进行任意粗细的折线描绘。

(b) 命令格式

- LINKE=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
OP CODE=1011_0000								Reserve(all 0)								Draw Mode																															
Color																n(2 ≤ n ≤ 65535)																															
Reserve(all 0)																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
符号	DX1(-32768 ≤ DX1 ≤ 32767)															符号	DY1(-32768 ≤ DY1 ≤ 32767)																														
符号	.															符号	.																														
符号	.															符号	.																														
符号	DXn(-32768 ≤ DXn ≤ 32767)															符号	DYn(-32768 ≤ DYn ≤ 32767)																														

【注】 在 n=0、1 时，不保证运行。

- LINKE=1 并且 LREL=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
OP CODE=1011_0000								Reserve(all 0)								Draw Mode																														
Color																n(2 ≤ n ≤ 65535)																														
Reserve(all 0)																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	LINK Address(long word address)														0	0																												

- 【注】
1. 在 n=0、1 时，不保证运行。
 2. 由于将 LINK Address 的下一个长字解释为下一个命令码，所以不能将配置 LINK Address 的下一个长字地址指定为 LINK Address 所指的连接目标地址。

- LINKE=1 并且 LREL=1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
OP CODE=1011_0000								Reserve(all 0)								Draw Mode																															
Color																n(2 ≤ n ≤ 65535)																															
Reserve(all 0)																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
符号扩展		符号		LINK Address(long word address)																								0	0																		

- 【注】 1. 在 n=0、1 时，不保证运行。
 2. 由于将 LINK Address 的下一个长字解释为命令码，所以不能将配置 LINK Address 的下一个长字地址指定为配置命令码的地址 +LINK Address 所指的连接目标地址。

1. CODE
B'10110000
2. 绘图属性

参照数据				描绘的坐标系	
多值源	2 值源	2 值工作	指定的颜色	绘图坐标	工作坐标
			○	○	

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	0 固定	CLIP	RCLIP	0 固定	LINKE	LREL	0 固定	0 固定	0 固定	0 固定	NET	EOS	COOF	AA	0 固定

3. 命令参数
 - Color : 指定 8 或者 16 位/像素颜色。对于 16 位/像素，必须符合目标像素格式。在 8 位/像素时，必须给高位和低位字节设定相同的值。
 - n (n=2 ~ 65535) : 顶点数。
 - W : 线宽。用 6 位整数进行设定，必须给未使用位写 0。
如果将 W 置 0，就描绘线宽为 1 的折线。禁止将 W 置 "1"。
 - DXn (n=2 ~ 65535) : 绘图坐标 (绝对坐标)。负数为 2 的补码。
 - DYn (n=2 ~ 65535) : 绘图坐标 (绝对坐标)。负数为 2 的补码。
 - LINK Address : LINK 绝对地址 (长字地址，必须给 A31 ~ 29、A1 和 A0 写 0。)
LINK 相对地址 (长字地址，负数为 2 的补码。必须对 A31 ~ A29 用 A28 的符号位进行符号扩展，给 A1 ~ A0 写 0。)

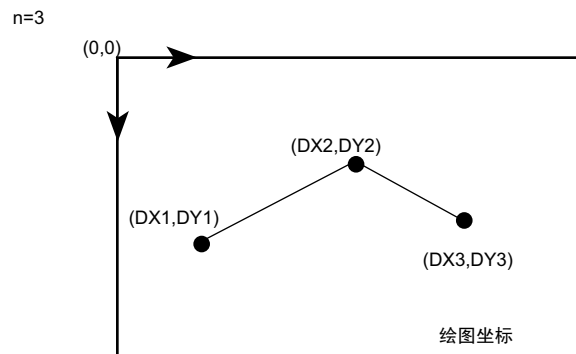
【注】 在 32 位地址模式中，必须将指定的 32 位地址的 bit28 ~ bit3 写到 A28 ~ A3。

(c) 说明

描绘从顶点 1(DX1, DY1) 到顶点 n(DXn, DYn) 的折线 (经过顶点 2(DX2, DY2)、...、顶点 n-1(DXn-1, DYn-1))。如果给 W 设定大于 1 的值，就描绘粗线。在 LINKE=1 时，从 LINK Address 指定的存储器读取顶点坐标。能通过 LREL 位将 LINK Address 指定为绝对地址或者配置 LINEC 命令的命令码的存储地址的相对地址。

- 【注】
1. 对线宽为 1 的线用 8 点进行描绘，对粗线用 8 点和 4 点进行描绘。
 2. 描绘每个线段的终点。在每个线段的起点 / 终点坐标相同时，对线宽为 1 的线描绘 1 个点，对粗线不进行任何描绘。
 3. 在 AA=1 时
 - 在每个线段的起点 / 终点坐标相同时，不进行抗锯齿处理。
 - 对水平、垂直和倾斜 45 度的线段不进行抗锯齿处理。

(d) 例



(4) LINED

(a) 功能

对多边形的外框进行抗锯齿处理。必须在目标为 16 位 / 像素时执行此命令。

(b) 命令格式

- LINKE=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1011_0011								Reserve(all 0)								Draw Mode															
Reserve(all 0)																n(2 ≤ n ≤ 65535)															
符号	DX1(−32768 ≤ DX1 ≤ 32767)														符号	DY1(−32768 ≤ DY1 ≤ 32767)															
符号	.														符号	.															
符号	.														符号	.															
符号	DXn(−32768 ≤ DXn ≤ 32767)														符号	DYn(−32768 ≤ DYn ≤ 32767)															

【注】 在 n=0、1 时，不保证运行。

- LINKE=1 并且 LREL=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1011_0011								Reserve(all 0)								Draw Mode															
Reserve(all 0)								Reserve(all 0)								n(2 ≤ n ≤ 65535)															
0	0	0	LINK Address(long word address)														0	0													

- 【注】 1. 在 n=0、1 时，不保证运行。
2. 由于将 LINK Address 的下一个长字解释为下一个命令码，所以不能将配置 LINK Address 的下一个长字地址指定为 LINK Address 所指的连接目标地址。

- LINKE=1 并且 LREL=1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1011_0011								Reserve(all 0)								Draw Mode															
Reserve(all 0)								Reserve(all 0)								n(2 ≤ n ≤ 65535)															
符号扩展		符号		LINK Address(long word address)														0		0											

- 【注】 1. 在 n=0、1 时，不保证运行。
2. 由于将 LINK Address 的下一个长字解释为下一个命令码，所以不能将配置 LINK Address 的下一个长字地址指定为配置命令码的地址 +LINK Address 所指的连接目标地址。

1. CODE
B'10110011
2. 绘图属性

参照数据				描绘的坐标系	
多值源	2 值源	2 值工作	指定的颜色	绘图坐标	工作坐标
				○	

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	0 固定	CLIP	RCLIP	0 固定	LINKE	LREL	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	AA(1)	CLKW

- 【注】 AA: 必须置 1。

3. 命令参数
 - n (n=2~65535) : 顶点数。
 - DXn (n=2~65535): 绘图坐标 (绝对坐标)。负数为 2 的补码。
 - DYn (n=2~65535): 绘图坐标 (绝对坐标)。负数为 2 的补码。
 - LINK Address : LINK 绝对地址 (长字地址, 必须给 A31~29、A1 和 A0 写 0。)
 - LINK 相对地址 (长字地址, 负数为 2 的补码。必须对 A31~A29 用 A28 的符号位进行符号扩展, 给 A1~A0 写 0。)

- 【注】 在 32 位地址模式中，必须将指定的 32 位地址的 bit28 ~ bit3 写到 A28 ~ A3。

(c) 说明

对工作参照描绘的多边形外框，进行抗锯齿处理。

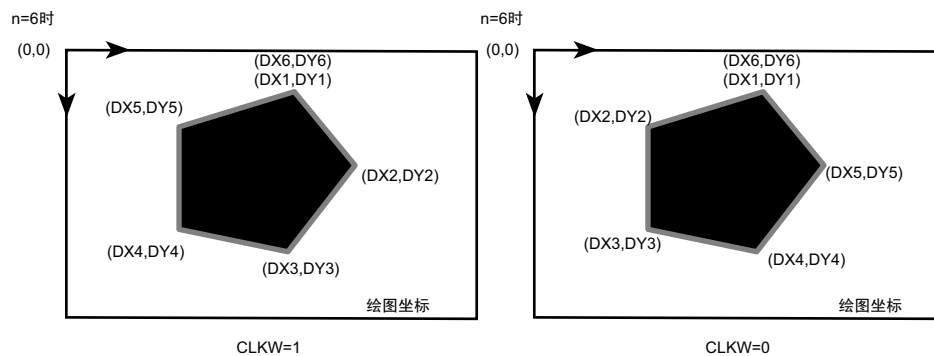
通过 CLKW 位指定 n 顶点的顺序是顺时针还是逆时针方向。在 CLKW=1 时为顺时针方向；在 CLKW=0 时为逆时针方向。通过抗锯齿处理，在顺时针方向时参照绘图方向的左像素；在逆时针方向时参照绘图方向的右像素。当 LINKE=1 时，从 LINK Address 指定的存储器读取顶点坐标。能通过 LREL 位将 LINK Address 指定为绝对地址或者配置 LINED 命令的命令码存储地址的相对地址。

必须在目标为 16 位 / 像素时执行此命令。

必须在 EDG 位 =1 并且 EOS 位 =1 时，通过 (R) FTRAPC 命令描绘工作参照时使用的多边形。

- 【注】**
1. 用 8 点进行描绘。
 2. 不描绘每个线段的终点。在对通过 POLYGON4 类命令描绘的图形外框进行抗锯齿处理时，轨迹可能不一致。
 3. 在每个线段的起点 / 终点坐标相同时，不进行任何描绘。
 4. 在 G2D 内部，对水平、垂直和倾斜 45 度的线段进行预裁剪，所以不进行抗锯齿处理。
 5. 如果参照的像素或者描绘的像素不在裁剪区，就对各像素进行裁剪，而不进行抗锯齿处理。

(d) 例



(5) RLINEA

(a) 功能

在目标区域中参照多值（8 或者 16 位 / 像素）源数据，通过当前指针的相对指定进行任意粗细的折线描绘。

(b) 命令格式

• REL=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
OP CODE=1011_0110								Reserve(all 0)								Draw Mode																			
0	0	0	Base Address(quad word address)																								0	0	0						
0	0	0	0	TDX($8 \leq TDX \leq 4088$)								0	0	0	0	0	0	0	0	TDY($1 \leq TDY \leq 4095$)															
0	0	0	0	TXOFS($0 \leq TXOFS \leq TDX-1$)								n($1 \leq n \leq 65535$)																							
Reserve(all 0)								0	0	0	0	0	0	0	0	0	0	0	0	W($0,2 \leq W \leq 63$)															
符号	DX ₂ ($-128 \leq DX_2 \leq 127$)								符号	DY ₂ ($-128 \leq DY_2 \leq 127$)								符号	DX ₁ ($-128 \leq DX_1 \leq 127$)								符号	DY ₁ ($-128 \leq DY_1 \leq 127$)							
符号	.								符号	.								符号	.								符号	.							
符号	.								符号	.								符号	.								符号	.							
符号	DX _n ($-128 \leq DX_n \leq 127$)								符号	DY _n ($-128 \leq DY_n \leq 127$)								符号	DX _{n-1} ($-128 \leq DX_{n-1} \leq 127$)								符号	DY _{n-1} ($-128 \leq DY_{n-1} \leq 127$)							

- 【注】 1. 在 W=0 时，必须设定 TDY=1。
 2. 在 n=0 时，不保证运行。
 3. 在 n 为奇数时，必须在最后输入虚字 0。

• REL=1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
OP CODE=1011_0110								Reserve(all 0)								Draw Mode																			
符号扩展		符号	Base Address(longword address)																								0	0							
0	0	0	0	TDX($8 \leq TDX \leq 4088$)								0	0	0	0	0	0	0	0	TDY($1 \leq TDY \leq 4095$)															
0	0	0	0	TXOFS($0 \leq TXOFS \leq TDX-1$)								n($1 \leq n \leq 65535$)																							
Reserve(all 0)								0	0	0	0	0	0	0	0	0	0	0	0	W($0,2 \leq W \leq 63$)															
符号	DX ₂ ($-128 \leq DX_2 \leq 127$)								符号	DY ₂ ($-128 \leq DY_2 \leq 127$)								符号	DX ₁ ($-128 \leq DX_1 \leq 127$)								符号	DY ₁ ($-128 \leq DY_1 \leq 127$)							
符号	.								符号	.								符号	.								符号	.							
符号	.								符号	.								符号	.								符号	.							
符号	DX _n ($-128 \leq DX_n \leq 127$)								符号	DY _n ($-128 \leq DY_n \leq 127$)								符号	DX _{n-1} ($-128 \leq DX_{n-1} \leq 127$)								符号	DY _{n-1} ($-128 \leq DY_{n-1} \leq 127$)							

- 【注】 1. 配置命令码的地址（长字：32 位单位）+Base Address（长字：32 位单位）必须为 4 个字地址（64 位单位）。
 2. 在 W=0 时，必须设定 TDY=1。
 3. 在 n=0 时，不保证运行。
 4. 在 n 为奇数时，必须在最后输入虚字 0。

1. CODE
B'10110110
2. 绘图属性

参照数据				描绘的坐标系	
多值源	2 值源	2 值工作	指定的颜色	绘图坐标	工作坐标
○				○	

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	0 固定	CLIP	RCLIP	STRANS	0 固定	0 固定	SS(0)	REL	STYLE(1)	0 固定	NET	EOS	COOF	AA	0 固定

【注】 SS : 必须置 0。

STYLE: 必须置 1。

3. 命令参数

Base Address : 源起始绝对地址 (4 个字地址, 必须给 A31 ~ 29 和 A2 ~ A0 写 0。)
源起始相对地址 (长字地址, 负数为 2 的补码。必须对 A31 ~ A29 用 A28 的符号位进行符号扩展, 给 A1 ~ A0 写 0。)

【注】 在 32 位地址模式中, 必须将指定的 32 位地址的 bit28 ~ bit3 写到 A28 ~ A3。

TDX、TDY : 源大小。必须给未使用位写 0。

TXOFS : 源偏移。必须给未使用位写 0。

n (n=1 ~ 65535) : 顶点数。

W : 线宽。用 6 位整数进行设定, 必须给未使用位写 0。
如果将 W 置 0, 就描绘线宽为 1 的折线。禁止将 W 置 1。

DXn (n=1 ~ 65535) : 绘图坐标 (相对值)。负数为 2 的补码。

DYn (n=1 ~ 65535) : 绘图坐标 (相对值)。负数为 2 的补码。

(c) 说明

将线段 (XC, YC) - (XC+DX1, YC+DY1)、(XC+DX1, YC+DY1) - (XC+DX1+DX2, YC+DY1+DY2)、...、(XC+...+DXn-1, YC+...+DYn-1) - (XC+...+DXn-1+DXn, YC+...+DYn-1+DYn) 的折线描绘到当前指针 (XC, YC) 的相对移动量 (DX, DY) 所指定的坐标。

将坐标最终点保存到当前指针 (XC, YC)。必须给 TDX 设定 8 像素的倍数。如果设定 TXOFS, 就按偏移量将源数据拆开进行参照。以像素为单位, 设定 TXOFS。

只在源的 X 方向重复 STYLE 的图案, 在 Y 方向根据线宽进行扩大和缩小。

如果给 W 设定大于 0 的值, 就描绘粗线。

- 【注】
1. 对线宽为 1 的线用 8 点进行描绘, 对粗线用 8 点和 4 点进行描绘。
 2. 描绘每个线段的终点。在每个线段的起点 / 终点坐标相同时, 对线宽为 1 的线描绘 1 个点, 对粗线不进行任何描绘。
 3. 在 AA=1 时
 - 虚线时, 对虚线的间隙不进行抗锯齿处理。
 - 在每个线段的起点 / 终点坐标相同时, 不进行抗锯齿处理。
 - 对水平、垂直和倾斜 45 度的线段不进行抗锯齿处理。
 4. 将坐标变换前的坐标最终点保存到当前指针 (XC, YC)。

(6) RLINEB

(a) (功能)

在目标区域中参照 2 值 (1 位 / 像素) 源数据, 通过当前指针的相对指定进行任意粗细的折线描绘。

(b) 命令格式

• REL=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
OP CODE=1011_0101								Reserve(all 0)								Draw Mode																			
Color1																Color0																			
0	0	0	Base Address(quad word address)																								0	0	0						
0	0	0	0	TDX($8 \leq TDX \leq 4088$)								0	0	0	0	0	0	0	0	TDY($1 \leq TDY \leq 4095$)															
0	0	0	0	TXOFS($0 \leq TXOFS \leq TDX-1$)								n($1 \leq n \leq 65535$)																							
Reserve(all 0)																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	W($0, 2 \leq W \leq 63$)			
符号	DX2($-128 \leq DX2 \leq 127$)								符号	DY2($-128 \leq DY2 \leq 127$)								符号	DX1($-128 \leq DX1 \leq 127$)								符号	DY1($-128 \leq DY1 \leq 127$)							
符号	.								符号	.								符号	.								符号	.							
符号	.								符号	.								符号	.								符号	.							
符号	DX _n ($-128 \leq DX_n \leq 127$)								符号	DY _n ($-128 \leq DY_n \leq 127$)								符号	DX _{n-1} ($-128 \leq DX_{n-1} \leq 127$)								符号	DY _{n-1} ($-128 \leq DY_{n-1} \leq 127$)							

- 【注】 1. 在 W=0 时, 必须设定 TDY=1。
 2. 在 n=0 时, 不保证运行。
 3. 在 n 为奇数时, 必须在最后输入虚字 0。

• REL=1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
OP CODE=1011_0101								Reserve(all 0)								Draw Mode																			
符号扩展	符号	Base Address(longword address)																								0	0								
0	0	0	0	TDX($8 \leq TDX \leq 4088$)								0	0	0	0	0	0	0	0	TDY($1 \leq TDY \leq 4095$)															
0	0	0	0	TXOFS($0 \leq TXOFS \leq TDX-1$)								n($1 \leq n \leq 65535$)																							
Reserve(all 0)																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	W($0, 2 \leq W \leq 63$)			
符号	DX2($-128 \leq DX2 \leq 127$)								符号	DY2($-128 \leq DY2 \leq 127$)								符号	DX1($-128 \leq DX1 \leq 127$)								符号	DY1($-128 \leq DY1 \leq 127$)							
符号	.								符号	.								符号	.								符号	.							
符号	.								符号	.								符号	.								符号	.							
符号	DX _n ($-128 \leq DX_n \leq 127$)								符号	DY _n ($-128 \leq DY_n \leq 127$)								符号	DX _{n-1} ($-128 \leq DX_{n-1} \leq 127$)								符号	DY _{n-1} ($-128 \leq DY_{n-1} \leq 127$)							

- 【注】 1. 配置命令码的地址 (长字: 32 位单位) +Base Address (长字: 32 位单位) 必须为 4 个字地址 (64 位单位)。
 2. 在 W=0 时, 必须设定 TDY=1。
 3. 在 n=0 时, 不保证运行。
 4. 在 n 为奇数时, 必须在最后输入虚字 0。

1. CODE
B'10110101
2. 绘图属性

参照数据				描绘的坐标系	
多值源	2 值源	2 值工作	指定的颜色	绘图坐标	工作坐标
	○			○	

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	0 固定	CLIP	RCLIP	STRANS	0 固定	0 固定	SS(0)	REL	STYLE(1)	0 固定	NET	EOS	COOF	AA	0 固定

【注】 SS : 必须置 0。

STYLE: 必须置 1。

3. 命令参数

Color0、Color1 : 指定 8 或者 16 位/像素颜色。对于 16 位/像素, 必须符合目标像素格式。在 8 位/像素时, 必须给高位和低位字节设定相同的值。

Base Address : 源起始绝对地址 (4 个字地址, 必须给 A31~29 和 A2~A0 写 0。) 源起始相对地址 (长字地址, 负数为 2 的补码。必须对 A31~A29 用 A28 的符号位进行符号扩展, 给 A1~A0 写 0。)

【注】 在 32 位地址模式中, 必须将指定的 32 位地址的 bit28~bit3 写到 A28~A3。

TDX、TDY : 源大小。必须给未使用位写 0。

TXOFS : 源偏移。必须给未使用位写 0。

n (n=1~65535) : 顶点数。

W : 线宽。用 6 位整数进行设定, 必须给未使用位写 0。
如果将 W 置 0, 就描绘线宽为 1 的折线。禁止将 W 置 1。

DXn (n=1~65535): 绘图坐标 (相对值)。负数为 2 的补码。

DYn (n=1~65535): 绘图坐标 (相对值)。负数为 2 的补码。

(c) 说明

将线段 (XC, YC) - (XC+DX1, YC+DY1)、(XC+DX1, YC+DY1) - (XC+DX1+DX2, YC+DY1+DY2)、...、(XC+...+DXn-1, YC+...+DYn-1) - (XC+...+DXn-1+DXn, YC+...+DYn-1+DYn) 的折线描绘到当前指针 (XC, YC) 的相对移动量 (DX, DY) 所指定的坐标。

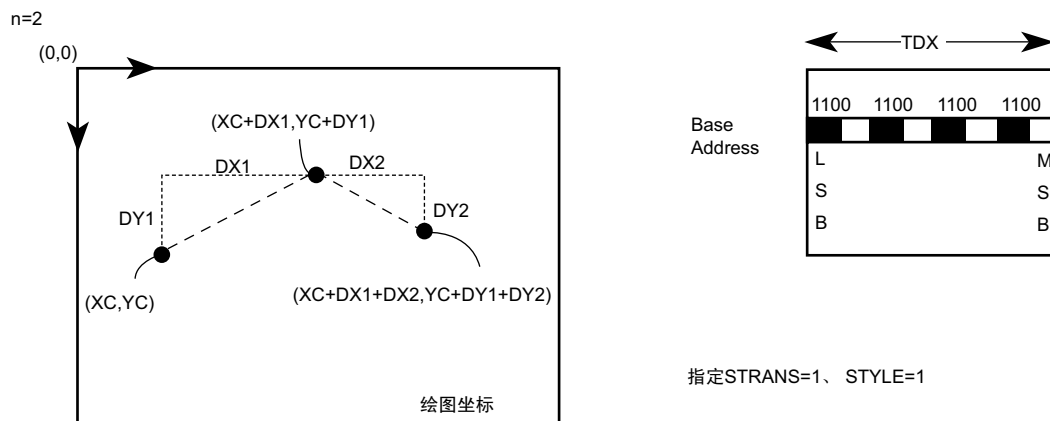
将坐标最终点保存到当前指针 (XC, YC)。必须给 TDX 设定 8 像素的倍数。如果设定 TXOFS, 就按偏移量将源数据拆开进行参照。以像素为单位, 设定 TXOFS。

只在源的 X 方向重复 STYLE 的图案, 在 Y 方向根据线宽进行扩大和缩小。

如果给 W 设定大于 1 的值, 就描绘粗线。

- 【注】
1. 对线宽为 1 的线用 8 点进行描绘, 对粗线用 8 点和 4 点进行描绘。
 2. 描绘每个线段的终点。在每个线段的起点/终点坐标相同时, 对线宽为 1 的线描绘 1 个点, 对粗线不进行任何描绘。
 3. 在 AA=1 时
 - 虚线时, 对虚线的间隙不进行抗锯齿处理。
 - 在每个线段的起点/终点坐标相同时, 不进行抗锯齿处理。
 - 对水平、垂直和倾斜 45 度的线段不进行抗锯齿处理。
 4. 将坐标变换前的坐标最终点保存到当前指针 (XC, YC)。

(d) 例



(7) RLINEC

(a) 功能

在目标区域中，通过单色的指定和当前指针的相对指定进行任意粗细的折线描绘。

(b) 命令格式

- LINKE=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
OP CODE=1011_0100								Reserve(all 0)								Draw Mode																														
Color																n(1 ≤ n ≤ 65535)																														
Reserve(all 0)																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
符号	DX2(-128 ≤ DX2 ≤ 127)								符号	DY2(-128 ≤ DY2 ≤ 127)								符号	DX1(-128 ≤ DX1 ≤ 127)								符号	DY1(-128 ≤ DY1 ≤ 127)																		
符号	.								符号	.								符号	.								符号	.																		
符号	.								符号	.								符号	.								符号	.																		
符号	DX _n (-128 ≤ DX _n ≤ 127)								符号	DY _n (-128 ≤ DY _n ≤ 127)								符号	DX _{n-1} (-128 ≤ DX _{n-1} ≤ 127)								符号	DY _{n-1} (-128 ≤ DY _{n-1} ≤ 127)																		

- 【注】 1. 在 n=0 时，不保证运行。
 2. 在 n 为奇数时，必须在最后输入虚字 0。

- LINKE=1 并且 LREL=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
OP CODE=1011_0100								Reserve(all 0)								Draw Mode																														
Color																n(1 ≤ n ≤ 65535)																														
Reserve(all 0)																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	LINK Address(long word address)													0	0																													

- 【注】 1. 在 n=0 时，不保证运行。

2. 在 n 为奇数时，必须在最后输入虚字 0。
3. 由于将 LINK Address 的下一个长字解释为下一个命令码，所以不能将配置 LINK Address 的下一个长字地址指定为 LINK Address 所指的连接目标地址。

• LINKE=1 并且 LREL=1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																	
OP CODE=1011_0100								Reserve(all 0)								Draw Mode																																
Color																n(1 ≤ n ≤ 65535)																																
Reserve(all 0)																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
符号	扩展	LINK Address(long word address)																									0	0																				

【注】 1. 在 n=0 时，不保证运行。

2. 在 n 为奇数时，必须最后输入虚字 0。
3. 由于将 LINK Address 的下一个长字解释为下一个命令码，所以不能将配置 LINK Address 的下一个长字地址指定为配置命令码的地址 +LINK Address 所指的连接目标地址。

1. CODE
B'10110100
2. 绘图属性

参照数据				描绘的坐标系	
多值源	2 值源	2 值工作	指定的颜色	绘图坐标	工作坐标
			○	○	

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	0 固定	CLIP	RCLIP	0 固定	LINKE	LREL	0 固定	0 固定	0 固定	0 固定	NET	EOS	COOF	AA	0 固定

3. 命令参数

- Color : 指定 8 或者 16 位/像素颜色。对 16 位/像素，必须符合目标像素格式。在 8 位/像素时，必须给高位和低位字节设定相同的值。
- n (n=1 ~ 65535) : 顶点数。
- W : 线宽。用 6 位整数进行设定，必须给未使用位写 0。
如果将 W 置 0，就描绘线宽为 1 的折线。禁止将 W 置 “1”。
- DXn (n=1 ~ 65535) : 绘图坐标 (相对值)。负数为 2 的补码。
- DYn (n=1 ~ 65535) : 绘图坐标 (相对值)。负数为 2 的补码。
- LINK Address : LINK 绝对地址 (长字地址，必须给 A31 ~ 29、A1 和 A0 写 0。)
LINK 相对地址 (长字地址，负数为 2 的补码。必须对 A31 ~ A29 用 A28 的符号位进行符号扩展，给 A1 ~ A0 写 0。)

【注】 在 32 位地址模式中，必须将指定的 32 位地址的 bit28 ~ bit3 写到 A28 ~ A3。

(c) 说明

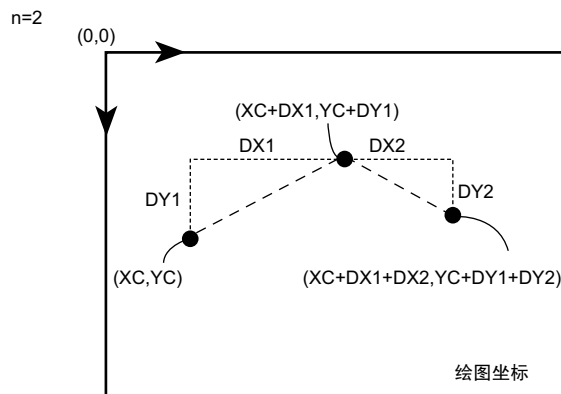
将线段 (XC, YC) - (XC+DX1, YC+DY1)、(XC+DX1, YC+DY1) - (XC+DX1+DX2, YC+DY1+DY2)、…、(XC+…+DXn-1, YC+…+DYn-1) - (XC+…+DXn-1+DXn, YC+…+DYn-1+DYn) 的折线描绘到当前指针 (XC, YC) 的相对移动量 (DX, DY) 所指定的坐标。

如果给 W 设定大于 1 的值, 就描绘粗线。在 LINKE=1 时, 从 LINK Address 指定的存储器读取顶点坐标。能通过 LREL 位将 LINK Address 指定为绝对地址或者配置 RLINEC 命令的命令码的存储地址的相对地址。

将坐标最终点保存到当前指针 (XC, YC)。

- 【注】**
1. 对线宽为 1 的线用 8 点进行描绘, 对粗线用 8 点和 4 点进行描绘。
 2. 描绘每个线段的终点。在每个线段的起点 / 终点坐标相同时, 对线宽为 1 的线描绘 1 个点, 对粗线不进行任何描绘。
 3. 在 AA=1 时
 - 在每个线段的起点/终点坐标相同时, 不进行抗锯齿处理。
 - 对水平、垂直和倾斜45度的线段不进行抗锯齿处理。
 4. 将坐标变换前的坐标最终点保存到当前指针 (XC, YC)。

(d) 例



(8) RLINED

(a) 功能

通过当前指针的相对指定对多边形的外框进行抗锯齿处理。必须在目标为 16 位 / 像素时执行此命令。

(b) 命令格式

- LINKE=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1011_0111								Reserve(all 0)								Draw Mode															
Reserve(all 0)																n(1 ≤ n ≤ 65535)															
符号	DX2(-128 ≤ DX2 ≤ 127)							符号	DY2(-128 ≤ DY2 ≤ 127)							符号	DX1(-128 ≤ DX1 ≤ 127)							符号	DY1(-128 ≤ DY1 ≤ 127)						
符号	.							符号	.							符号	.							符号	.						
符号	.							符号	.							符号	.							符号	.						
符号	DX _n (-128 ≤ DX _n ≤ 127)							符号	DY _n (-128 ≤ DY _n ≤ 127)							符号	DX _{n-1} (-128 ≤ DX _{n-1} ≤ 127)							符号	DY _{n-1} (-128 ≤ DY _{n-1} ≤ 127)						

- 【注】**
1. 在 n=0 时, 不保证运行。
 2. 在 n 为奇数时, 必须在最后输入虚字 0。

- LINKE=1 并且 LREL=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1011_0111								Reserve(all 0)								Draw Mode															
Reserve(all 0)								Reserve(all 0)								n(1 ≤ n ≤ 65535)															
0	0	0	LINK Address(long word address)														0	0													

- 【注】
1. 在 n=0 时，不保证运行。
 2. 在 n 为奇数时，必须在最后输入虚字 0。
 3. 由于将 LINK Address 的下一个长字解释为下一个命令码，所以不能将配置 LINK Address 的下一个长字地址指定为 LINK Address 所指的连接目标地址。

- LINKE=1 并且 LREL=1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1011_0111								Reserve(all 0)								Draw Mode															
Reserve(all 0)								Reserve(all 0)								n(1 ≤ n ≤ 65535)															
符号扩展	符号	LINK Address(long word address)														0	0														

- 【注】
1. 在 n=0 时，不保证运行。
 2. 在 n 为奇数时，必须在最后输入虚字 0。
 3. 由于将 LINK Address 的下一个长字解释为下一个命令码，所以不能将配置 LINK Address 的下一个长字地址指定为配置命令码的地址 +LINK Address 所指的连接目标地址。

1. CODE
B'10110111
2. 绘图属性

参照数据				描绘的坐标系	
多值源	2 值源	2 值工作	指定的颜色	绘图坐标	工作坐标
				○	

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	0 固定	CLIP	RCLIP	0 固定	LINKE	LREL	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	AA(1)	CLKW

- 【注】 AA: 必须置 1。

3. 命令参数
 - n (n=1 ~ 65535) : 顶点数。
 - DXn (n=1 ~ 65535) : 绘图坐标 (相对值)。负数为 2 的补码。
 - DYn (n=1 ~ 65535) : 绘图坐标 (相对值)。负数为 2 的补码。
 - LINK Address : LINK 绝对地址 (长字地址, 必须给 A31 ~ 29、A1 和 A0 写 0。)
LINK 相对地址 (长字地址, 负数为 2 的补码。必须对 A31 ~ A29 用 A28 的符号位进行符号扩展, 给 A1 ~ A0 写 0。)

- 【注】 在 32 位地址模式中, 必须将指定的 32 位地址的 bit28 ~ bit3 写到 A28 ~ A3。

(c) 说明

通过当前指针的相对指定，对工作参照描绘的多边形外框进行抗锯齿处理。

通过 CLKW 位指定 n 顶点的顺序是顺时针还是逆时针方向。在 CLKW=1 时为顺时针方向；在 CLKW=0 时为逆时针方向。通过抗锯齿处理，在顺时针方向时参照绘图方向的左像素，在逆时针方向时参照绘图方向的右像素。当 LINKE=1 时，从 LINK Address 指定的存储器读取顶点坐标。能通过 LREL 位将 LINK Address 指定为绝对地址或者配置 LINED 命令的命令码存储地址的相对地址。

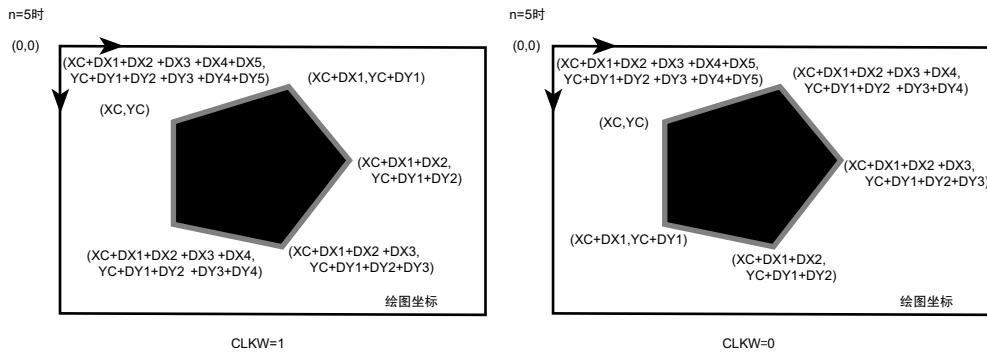
必须在目标为 16 位 / 像素时执行此命令。

必须在 EDG 位 =1 并且 EOS 位 =1 时，通过 (R) FTRAPC 命令描绘工作参照时使用的多边形。

将坐标最终点保存到当前指针 (XC, YC)。

- 【注】**
1. 用 8 点进行描绘。
 2. 不描绘每个线段的终点。在对通过 POLYGON4 类命令描绘的图形外框进行抗锯齿处理时，轨迹可能不一致。
 3. 在每个线段的起点 / 终点坐标相同时，不进行任何描绘。
 4. 在 G2D 内部，对水平、垂直和倾斜 45 度的线段进行预裁剪，所以不进行抗锯齿处理。
 5. 如果参照的像素或者描绘的像素不在裁剪区，就对各像素进行裁剪，而不进行抗锯齿处理。
 6. 将坐标变换前的坐标最终点保存到当前指针 (XC, YC)。

(d) 例



23.2.3 工作屏幕的绘图命令

(1) FTRAPC

(a) 功能

在工作坐标中描绘多边形。

(b) 命令格式

- LINKE=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1101_0000								Reserve(all 0)								Draw Mode															
Reserve(all 0)								Reserve(all 0)								n(2 ≤ n ≤ 65535)															
符号	Xmin(-32768 ≤ Xmin ≤ 32767)															符号	Ymin(-32768 ≤ Ymin ≤ 32767)														
符号	Xmax(-32768 ≤ Xmax ≤ 32767)															符号	Ymax(-32768 ≤ Ymax ≤ 32767)														
符号	DX1(-32768 ≤ DX1 ≤ 32767)															符号	DY1(-32768 ≤ DY1 ≤ 32767)														
符号	.															符号	.														
符号	.															符号	.														
符号	DXn(-32768 ≤ DXn ≤ 32767)															符号	DYn(-32768 ≤ DYn ≤ 32767)														

【注】 1. 在 n=0、1 时，不保证运行。

- LINKE=1 并且 LREL=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1101_0000								Reserve(all 0)								Draw Mode															
Reserve(all 0)								Reserve(all 0)								n(2 ≤ n ≤ 65535)															
符号	Xmin(-32768 ≤ Xmin ≤ 32767)															符号	Ymin(-32768 ≤ Ymin ≤ 32767)														
符号	Xmax(-32768 ≤ Xmax ≤ 32767)															符号	Ymax(-32768 ≤ Ymax ≤ 32767)														
0	0	0	LINK Address(long word address)													0	0														

【注】 1. 在 n=0、1 时，不保证运行。

2. 由于将 LINK Address 的下一个长字解释为下一个命令码，所以不能将配置 LINK Address 的下一个长字地址指定为 LINK Address 所指的连接目标地址。

- LINKE=1 并且 LREL=1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1101_0000								Reserve(all 0)								Draw Mode															
Reserve(all 0)								n(2 ≤ n ≤ 65535)																							
符号	Xmin(-32768 ≤ Xmin ≤ 32767)															符号	Ymin(-32768 ≤ Ymin ≤ 32767)														
符号	Xmax(-32768 ≤ Xmax ≤ 32767)															符号	Ymax(-32768 ≤ Ymax ≤ 32767)														
符号扩展	符号	LINK Address(long word address)															0	0													

- 【注】 1. 在 n=0、1 时，不保证运行。
 2. 由于将 LINK Address 的下一个长字解释为下一个命令码，所以不能将配置 LINK Address 的下一个长字地址指定为配置命令码的地址 +LINK Address 所指的连接目标地址。

1. CODE
B'11010000
2. 绘图属性

参照数据				描绘的坐标系	
多值源	2 值源	2 值工作	指定的颜色	绘图坐标	工作坐标
					○

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	0 固定	CLIP	RCLIP	0 固定	LINKE	LREL	0 固定	0 固定	0 固定	BLKE(1)	EDG	EOS	0 固定	0 固定	0 固定

- 【注】 BLKE: 必须置 1。

3. 命令参数

n (n=2~65535) : 顶点数。

Xmin: 多边形外接四边形的 Xmin 值，工作坐标（绝对坐标）。负数为 2 的补码。

Ymin: 多边形外接四边形的 Ymin 值，工作坐标（绝对坐标）。负数为 2 的补码。

Xmax: 多边形外接四边形的 Xmax 值，工作坐标（绝对坐标）。负数为 2 的补码。

Ymax: 多边形外接四边形的 Ymax 值，工作坐标（绝对坐标）。负数为 2 的补码。

DXn (n=2~65535) : 工作坐标（绝对坐标）。负数为 2 的补码。

DYn (n=2~65535) : 工作坐标（绝对坐标）。负数为 2 的补码。

LINK Address : LINK 绝对地址（长字地址，必须给 A31~29、A1 和 A0 写 0。）

LINK 相对地址（长字地址，负数为 2 的补码。必须对 A31~A29 用 A28 的符号位进行符号扩展，给 A1~A0 写 0。）

- 【注】 在 32 位地址模式中，必须将指定的 32 位地址的 bit28~bit3 写到 A28~A3。

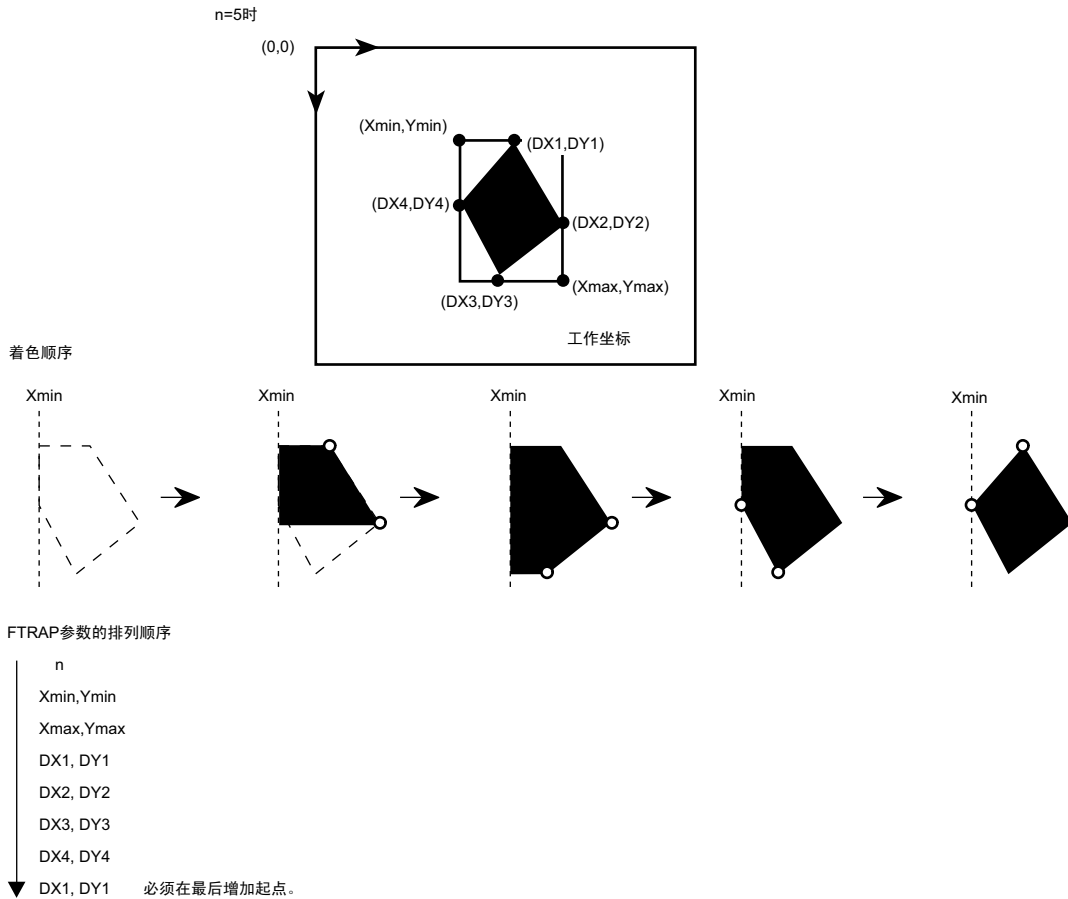
(c) 说明

在工作坐标中描绘 n-1 个顶点的多边形。用 2 值 EOR 对 X=Xmin 为左侧、线段 (DX1, DY1) - (DX2, DY2)、(DX2, DY2) - (DX3, DY3)、...、(DXn-1, DYn-1) - (DXn, DYn) 为右侧、上底和下底平行于 X 轴的 n-1 个梯形进行着色。但是，不描绘下底线。为了使其为封闭图形，必须使 (DXn, DYn)=(DX1, DY1)。通过将绘图属性的 EDG 位置 1，能在着色后描绘边缘线。通过 EOS 位选择线的描绘数据。

如果通过 FTRAPC 命令进行坐标变换，就在内部从输入的多边形的外接四边形的坐标值获取 4 个顶点，并且对这 4 个顶点进行坐标变换。然后，在内部将此变换后的 4 个顶点转换为外接矩形，并且求得左侧进行描绘。

【注】 如果边缘描绘有效 (EDG=1)，就不对边缘线进行 Z 预裁剪。

(d) 例



(2) RFTRAPC

(a) 功能

在工作坐标中，通过当前指针的相对指定进行多边形描绘。

(b) 命令格式

- LINKE=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1101_0100								Reserve(all 0)								Draw Mode															
Reserve(all 0)								n(1 ≤ n ≤ 65535)																							
符号	Xmin(-32768 ≤ Xmin ≤ 32767)														符号	Ymin(-32768 ≤ Ymin ≤ 32767)															
符号	Xmax(-32768 ≤ Xmax ≤ 32767)														符号	Ymax(-32768 ≤ Ymax ≤ 32767)															
符号	DX2(-128 ≤ DX2 ≤ 127)				符号	DY2(-128 ≤ DY2 ≤ 127)				符号	DX1(-128 ≤ DX1 ≤ 127)				符号	DY1(-128 ≤ DY1 ≤ 127)															
符号	.				符号	.				符号	.				符号	.															
符号	.				符号	.				符号	.				符号	.															
符号	DX _n (-128 ≤ DX _n ≤ 127)				符号	DY _n (-128 ≤ DY _n ≤ 127)				符号	DX _{n-1} (-128 ≤ DX _{n-1} ≤ 127)				符号	DY _{n-1} (-128 ≤ DY _{n-1} ≤ 127)															

- 【注】 1. 在 n=0 时，不保证运行。
2. 在 n 为奇数时，必须在最后输入虚字“0”。

- LINKE=1 并且 LREL=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1101_0100								Reserve(all 0)								Draw Mode															
Reserve(all 0)								n(1 ≤ n ≤ 65535)																							
符号	Xmin(-32768 ≤ Xmin ≤ 32767)														符号	Ymin(-32768 ≤ Ymin ≤ 32767)															
符号	Xmax(-32768 ≤ Xmax ≤ 32767)														符号	Ymax(-32768 ≤ Ymax ≤ 32767)															
0	0	0	LINK Address(long word address)														0	0													

- 【注】 1. 在 n=0 时，不保证运行。
2. 在 n 为奇数时，必须在最后输入虚字“0”。
3. 由于将 LINK Address 的下一个长字解释为下一个命令码，所以不能将配置 LINK Address 的下一个长字地址指定为 LINK Address 所指的连接目标地址。

- LINKE=1 并且 LREL=1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1101_0100								Reserve(all 0)								Draw Mode															
Reserve(all 0)								n(1 ≤ n ≤ 65535)																							
符号	Xmin(−32768 ≤ Xmin ≤ 32767)															符号	Ymin(−32768 ≤ Ymin ≤ 32767)														
符号	Xmax(−32768 ≤ Xmax ≤ 32767)															符号	Ymax(−32768 ≤ Ymax ≤ 32767)														
符号扩展	符号	LINK Address(long word address)																									0	0			

- 【注】
1. 在 n=0 时，不保证运行。
 2. 在 n 为奇数时，必须在最后输入虚字“0”。
 3. 由于将 LINK Address 的下一个长字解释为下一个命令码，所以不能将配置 LINK Address 的下一个长字地址指定为配置命令码的地址 +LINK Address 所指的连接目标地址。

1. CODE
B'11010100
2. 绘图属性

参照数据				描绘的坐标系	
多值源	2 值源	2 值工作	指定的颜色	绘图坐标	工作坐标
					○

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	0 固定	CLIP	RCLIP	0 固定	LINKE	LREL	0 固定	0 固定	0 固定	BLKE(1)	EDG	EOS	0 固定	0 固定	0 固定

- 【注】 BLKE: 必须置 1。

3. 命令参数
 - n (n=1 ~ 65535) : 顶点数。
 - Xmin: 多边形的外接四边形的 Xmin 值，工作坐标 (绝对坐标)。负数为 2 的补码。
 - Ymin: 多边形的外接四边形的 Ymin 值，工作坐标 (绝对坐标)。负数为 2 的补码。
 - Xmax: 多边形的外接四边形的 Xmax 值，工作坐标 (绝对坐标)。负数为 2 的补码。
 - Ymax: 多边形的外接四边形的 Ymax 值，工作坐标 (绝对坐标)。负数为 2 的补码。
 - DXn (n=1 ~ 65535) : 工作坐标 (相对坐标)。负数为 2 的补码。
 - DYn (n=1 ~ 65535) : 工作坐标 (相对坐标)。负数为 2 的补码。
 - LINK Address : LINK 绝对地址 (长字地址，必须给 A31 ~ 29、A1 和 A0 写 0。)
 - LINK 相对地址 (长字地址，负数为 2 的补码。必须对 A31 ~ A29 用 A28 的符号位进行符号扩展，给 A1 ~ A0 写 0。)

- 【注】 在 32 位地址模式中，必须将指定的 32 位地址的 bit28 ~ bit3 写到 A28 ~ A3。

(c) 说明

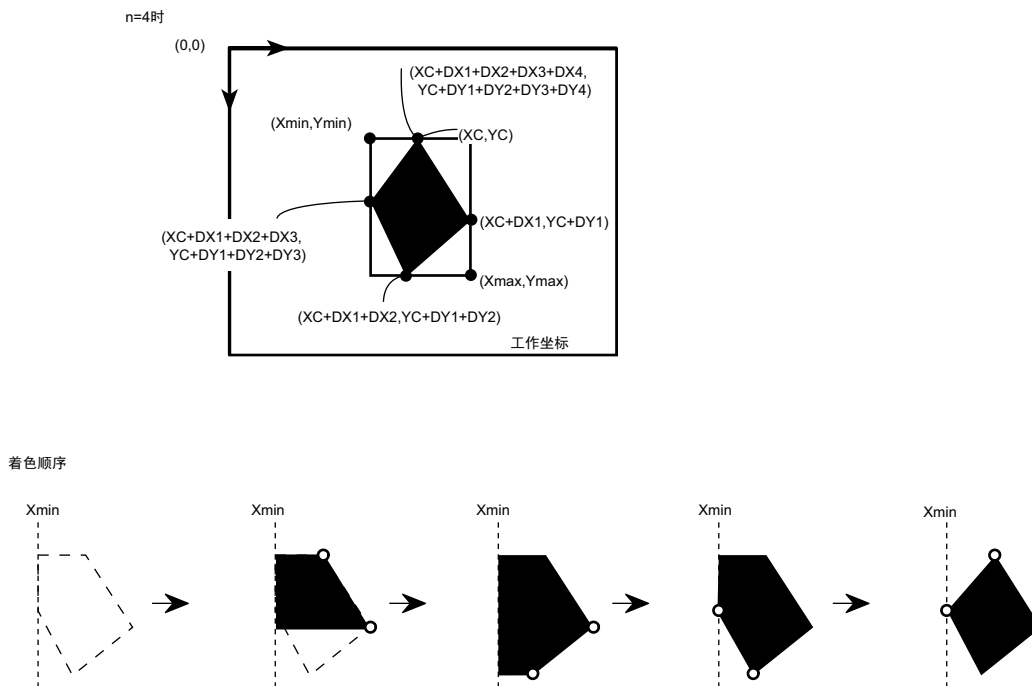
在工作坐标中描绘 $n-1$ 个顶点的多边形。用 2 值 EOR 对 $X=X_{min}$ 为左侧、当前指针 (XC, YC) 的相对移动量 (DX, DY) 所指定坐标的线段 $(XC, YC) - (XC+DX1, YC+DY1)$ 、 $(XC+DX1, YC+DY1) - (XC+DX1+DX2, YC+DY1+DY2)$ 、 \dots 、 $(XC+\dots+DX_{n-1}, YC+\dots+DY_{n-1}) - (XC+\dots+DX_n, YC+\dots+DY_n)$ 为右侧、上底和下底平行于 X 轴的 $n-1$ 个梯形进行着色。但是，不描绘下底线。

将坐标最终点保存到当前指针 (XC, YC) 。为了使其为封闭图形，必须使 $DX1+DX2+DX_n=0$ 和 $DY1+DY2+DY_n=0$ 。通过将绘图属性的 EDG 位置 1，能在着色后描绘边缘线。通过 EOS 位选择线的描绘数据。

如果通过 RFTRAPC 命令进行坐标变换时，就在内部从输入的多边形的外接四边形的坐标值获取 4 个顶点，并且对这 4 个顶点进行坐标变换。然后，在内部将此变换后的 4 个顶点转换为外接矩形，并且求得左侧进行描绘。

- 【注】 1. 将坐标变换前的坐标最终点保存到当前指针 (XC, YC) 。
- 2. 如果边缘描绘有效 (EDG=1)，就不对边缘线进行 Z 预裁剪。

(d) 例



(3) CLRWC

(a) 功能

对工作坐标进行初始化。

(b) 命令格式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1110_0000								Reserve(all 0)								Draw Mode															
符号	Xmin($-32768 \leq Xmin \leq 32767$)														符号	Ymin($-32768 \leq Ymin \leq 32767$)															
符号	Xmax($-32768 \leq Xmax \leq 32767$)														符号	Ymax($-32768 \leq Ymax \leq 32767$)															

1. CODE

B'11100000

2. 绘图属性

参照数据				描绘的坐标系	
多值源	2 值源	2 值工作	指定的颜色	绘图坐标	工作坐标
					○

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	0 固定	CLIP	RCLIP	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	BLKE(1)	0 固定	0 固定	0 固定	0 固定	0 固定

【注】 BLKE: 必须置 1。

3. 命令参数

Xmin、Xmax: 左右 X 坐标值, 工作坐标 (绝对坐标)。负数为 2 的补码。

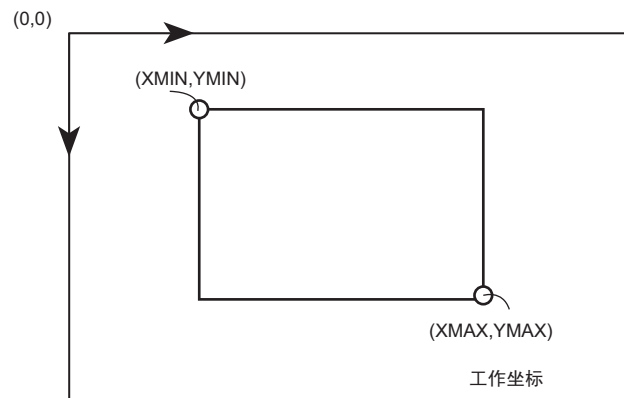
Ymin、Ymax: 上下 Y 坐标值, 工作坐标 (绝对坐标)。负数为 2 的补码。

(c) 说明

用左上角的坐标 (XMIN, YMIN) 和右下角的坐标 (XMAX, YMAX) 指定的区域, 对工作坐标进行初始化。

如果通过 CLRWC 命令进行坐标变换, 就在内部从输入的左右 X 坐标值和上下 Y 坐标值获取 4 个顶点, 并且对这 4 个顶点进行坐标变换。然后, 在内部将此变换后的 4 个顶点转换为外接矩形, 并且进行描绘。

(d) 例



23.2.4 工作线的绘图命令

(1) LINEWC

(a) 功能

在工作坐标中描绘 1 个点宽的实线。

(b) 命令格式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1111_0000								Reserve(all 0)								Draw Mode															
Reserve(all 0)								n(2 ≤ n ≤ 65535)																							
符号	DX1(-32768 ≤ DX1 ≤ 32767)								符号	DY1(-32768 ≤ DY1 ≤ 32767)																					
符号	.								符号	.																					
符号	.								符号	.																					
符号	DXn(-32768 ≤ DXn ≤ 32767)								符号	DYn(-32768 ≤ DYn ≤ 32767)																					

【注】 在 n=0、1 时，不保证运行。

1. CODE
B'11110000
2. 绘图属性

参照数据				描绘的坐标系	
多值源	2 值源	2 值工作	指定的颜色	绘图坐标	工作坐标
			○ (2 值 EOS)		○

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	0 固定	CLIP	RCLIP	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	EOS	0 固定	0 固定	0 固定

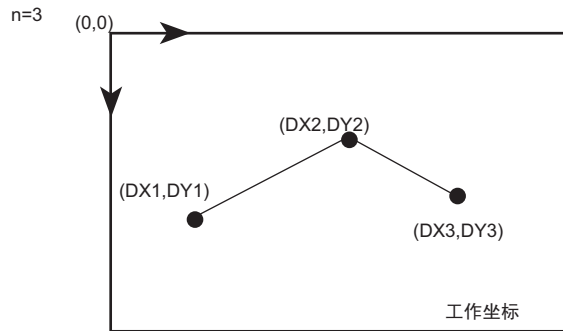
3. 命令参数
 - n (n=2~65535) : 顶点数。
 - DXn (n=2~65535) : 工作坐标 (绝对坐标)。负数为 2 的补码。
 - DYn (n=2~65535) : 工作坐标 (绝对坐标)。负数为 2 的补码。

(c) 说明

在工作坐标中，对从顶点 1(DX1, DY1) 到顶点 n(DXn, DYn) 的折线 (经过顶点 2(DX2, DY2)、…、顶点 n-1(Dxn-1, Dyn-1)) 进行 2 值描绘。通过绘图模式的 EOS 位选择 0 描画或者 1 描画。当选择 EOS 位 =0 时，就在工作坐标中用 0 进行描绘；当选择 EOS 位 =1 时，就在工作坐标中用 1 进行描绘 (用于在工作坐标中进行多边形着色时的边缘线描绘)。

【注】 用 8 点进行描绘。描绘每个线段的终点。

(d) 例



(2) RLINEWC

(a) 功能

在工作坐标中，通过当前指针的相对指定进行 1 个点宽的实线描绘。

(b) 命令格式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1111_0100								Reserve(all 0)								Draw Mode															
Reserve(all 0)								n(1 ≤ n ≤ 65535)																							
符号	DX2(-128 ≤ DX2 ≤ 127)							符号	DY2(-128 ≤ DY2 ≤ 127)							符号	DX1(-128 ≤ DX1 ≤ 127)							符号	DY1(-128 ≤ DY1 ≤ 127)						
符号	.							符号	.							符号	.							符号	.						
符号	.							符号	.							符号	.							符号	.						
符号	DX _n (-128 ≤ DX _n ≤ 127)							符号	DY _n (-128 ≤ DY _n ≤ 127)							符号	DX _{n-1} (-128 ≤ DX _{n-1} ≤ 127)							符号	DY _{n-1} (-128 ≤ DY _{n-1} ≤ 127)						

- 【注】 1. 在 n=0 时，不保证运行。
 2. 在 n 为奇数时，必须在最后输入虚字 0。

- CODE
B'11110100
- 绘图属性

参照数据				描绘的坐标系	
多值源	2 值源	2 值工作	指定的颜色	绘图坐标	工作坐标
			○ (2 值 EOS)		○

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	0 固定	CLIP	RCLIP	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	EOS	0 固定	0 固定	0 固定

3. 命令参数

n ($n=1 \sim 65535$) : 顶点数。

DX_n ($n=1 \sim 65535$) : 工作坐标 (相对坐标)。负数为 2 的补码。

DY_n ($n=1 \sim 65535$) : 工作坐标 (相对坐标)。负数为 2 的补码。

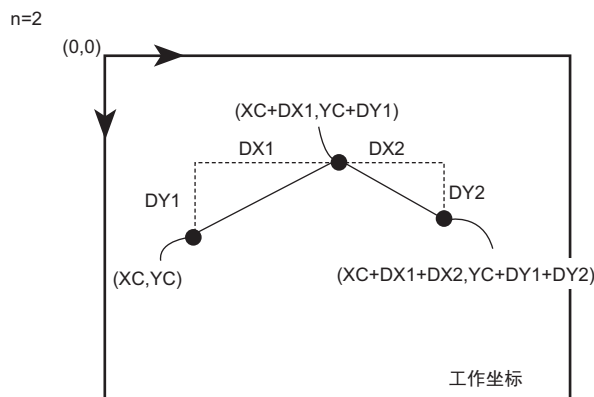
(c) 说明

在工作坐标中, 对当前指针 (XC, YC) 的相对移动量 (DX, DY) 所指定坐标的线段 $(XC, YC) - (XC+DX_1, YC+DY_1)$ 、 $(XC+DX_1, YC+DY_1) - (XC+DX_1+DX_2, YC+DY_1+DY_2)$ 、 \dots 、 $(XC+\dots+DX_{n-1}, YC+\dots+DY_{n-1}) - (XC+\dots+DX_{n-1}+DX_n, YC+\dots+DY_{n-1}+DY_n)$ 的折线进行 2 值描绘。通过绘图属性的 EOS 位选择 0 描画或者 1 描画。当选择 EOS 位 =0 时, 就在工作坐标中用 0 进行描绘; 当选择 EOS 位 =1 时, 就在工作坐标中用 1 进行描绘 (用于在工作坐标中进行多边形着色时的边缘线描绘)。

将坐标最终点保存到当前指针 (XC, YC) 。

- 【注】** 1. 用 8 点进行描绘。描绘每个线段的终点。
2. 将坐标变换之前的坐标最终点保存到当前指针 (XC, YC) 。

(d) 例



23.2.5 矩形的绘图命令

(1) BITBLTA

(a) 功能

将多值 (8 或者 16 位 / 像素) 矩形的源数据传送到目标区域。

(b) 命令格式

- SS=1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1010_0010								Reserve(all 0)								Draw Mode															
Reserve(all 0)								0				0				0				ROP											
0				0				TXS($0 \leq TXS \leq 4088$)								0				0				TYS($0 \leq TYS \leq 4095$)							
0				0				LW($0 \leq LW \leq 4094$)								0				0				RW($0 \leq RW \leq 4094$)							
0				0				TH($0 \leq TH \leq 4094$)								0				0				BH($0 \leq BH \leq 4094$)							
符号 BXC($-32768 \leq BXC \leq 32767$)																符号 BYC($-32768 \leq BYC \leq 32767$)															

- 【注】
1. $0 \leq TXS \leq SSTRR-(LW+RW+1)$ 、 $0 \leq TYS \leq 4096-(TH+BH+1)$ (SSTRR: 源跨距寄存器的设定值)
 2. $8 \leq LW+RW+1 \leq 4095$ 、 $1 \leq TH+BH+1 \leq 4095$
 3. $-32768 \leq BXC-LW \leq 32767$ 、 $-32768 \leq BYC-TH \leq 32767$ 、 $-32768 \leq BXC+RW \leq 32767$ 、
 $-32768 \leq BYC+BH \leq 32767$

- SS=0 并且 REL=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
OP CODE=1010_0010								Reserve(all 0)								Draw Mode																			
Reserve(all 0)								0				0				0				ROP															
0				0				Base Address(quad word address)																0				0				0			
0				0				LW($0 \leq LW \leq 4087$)								0				0				RW($0 \leq RW \leq 4087$)											
0				0				TH($0 \leq TH \leq 4094$)								0				0				BH($0 \leq BH \leq 4094$)											
符号 BXC($-32768 \leq BXC \leq 32767$)																符号 BYC($-32768 \leq BYC \leq 32767$)																			

- 【注】
1. $8 \leq LW+RW+1 \leq 4088$ (8 的倍数)、 $1 \leq TH+BH+1 \leq 4095$
 2. $-32768 \leq BXC-LW \leq 32767$ 、 $-32768 \leq BYC-TH \leq 32767$ 、 $-32768 \leq BXC+RW \leq 32767$ 、
 $-32768 \leq BYC+BH \leq 32767$

- SS=0 并且 REL=1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1010_0010								Reserve(all 0)								Draw Mode															
Reserve(all 0)								0	0	0	0	0	0	0	0	ROP															
符号扩展		符号		Base Address(long word address)																								0	0		
0	0	0	0	LW($0 \leq LW \leq 4087$)								0	0	0	0	RW($0 \leq RW \leq 4087$)															
0	0	0	0	TH($0 \leq TH \leq 4094$)								0	0	0	0	BH($0 \leq BH \leq 4094$)															
符号		BXC($-32768 \leq BXC \leq 32767$)												符号		BYC($-32768 \leq BYC \leq 32767$)															

- 【注】
- $8 \leq LW+RW+1 \leq 4088$ (8 的倍数)、 $1 \leq TH+BH+1 \leq 4095$
 - $-32768 \leq BXC-LW \leq 32767$ 、 $-32768 \leq BYC-TH \leq 32767$ 、 $-32768 \leq BXC+RW \leq 32767$ 、 $-32768 \leq BYC+BH \leq 32767$
 - 配置命令码的地址 (长字: 32 位单位) +Base Address (长字: 32 位单位) 必须为 4 个字地址 (64 位单位)。

- CODE
B'10100010
- 绘图属性

参照数据				描绘的坐标系	
多值源	2 值源	2 值工作	指定的颜色	绘图坐标	工作坐标
○		○ (仅 WORK=1)		○	

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	0 固定	CLIP	RCLIP	STRANS	DTRANS	WORK	SS	REL	SRCDIRX	SRCDIRY	DSTDIRX	DSTDIRY	COOF	αE	$S\alpha E$

- 命令参数
 - TXS、TYS : 源起始点。必须给未使用位写 0。
 - Base Address: 源起始绝对地址 (4 个字地址, 必须给 A31 ~ 29 和 A2 ~ A0 写 0。)
 - 源起始相对地址 (长字地址, 负数为 2 的补码。必须对 A31 ~ A29 用 A28 的符号位进行符号扩展, 给 A1 ~ A0 写 0。)

【注】 在 32 位地址模式中, 必须将指定的 32 位地址的 bit28 ~ bit3 写到 A28 ~ A3。

- BXC、BYC: 中心 X、Y 坐标值, 绘图坐标 (绝对坐标)。负数为 2 的补码。
- LW、RW : Left 宽、Right 宽, 来自 (BXC, BYC) 的相对值, 绘图坐标。必须用像素数进行设定, 给未使用位写 0。
- TH、BH : Top 高、Bottom 高, (BXC, BYC) 的相对值, 绘图坐标。必须用像素数进行设定, 给未使用位写 0。
- ROP : 光栅操作码。

(c) 说明

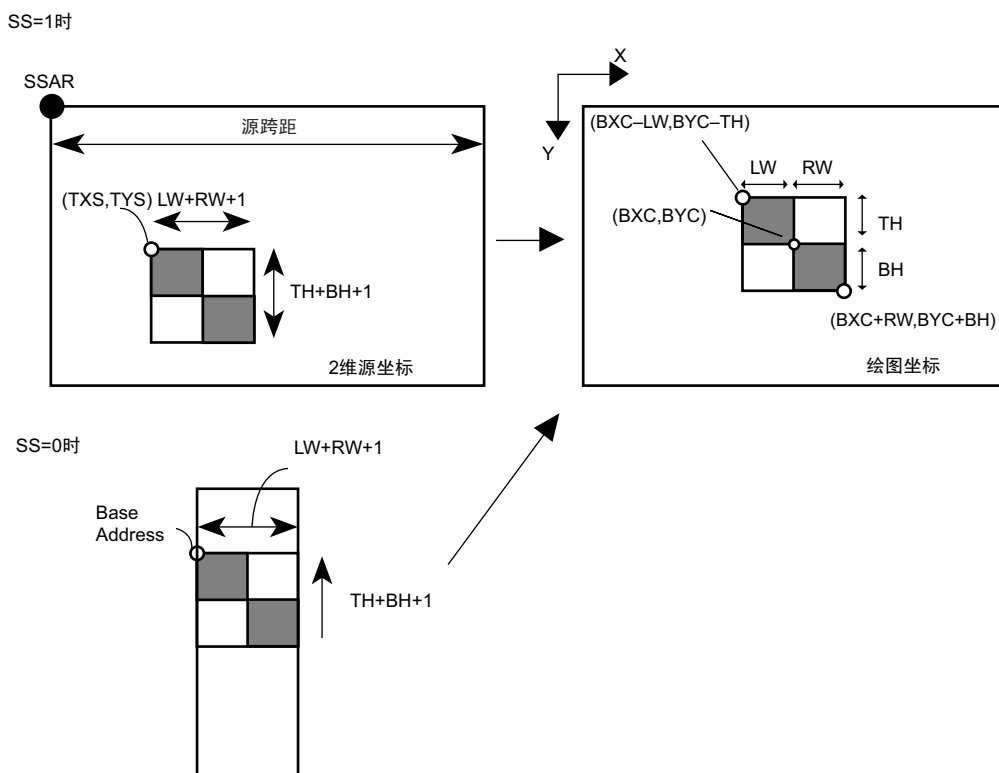
将多值 (8 或者 16 位 / 像素) 矩形的源数据传送到绘图坐标。

在 $SS=0$ 时, $LW+RW+1$ 必须为 8 的倍数; 在 $SS=1$ 时, $LW+RW+1$ 不能小于 8 的值。

1. 当通过绘图属性选择工作指定 ($WORK=1$) 时, 就参照工作坐标 (坐标与绘图坐标相同), 同时在绘图坐标中描绘工作坐标的像素为 1 的部分。
2. 在 $SS=1$ 时, 从 2 维源区域参照源数据; 在 $SS=0$ 时, 从 Base Address 参照源数据。在 $REL=0$ 时, 能将源地址指定为绝对地址; 在 $REL=1$ 时, 能将源地址指定为配置 BITBLTA 命令的命令码的存储地址的相对地址。
3. 在进行 16 位 / 像素描绘时, 如果将绘图属性的 COOF 位置 1, 就给多值源数据的值加上 COFSR 寄存器的值, 然后描绘所得的结果。此时, 通过饱和和处理进行运算。在 8 位 / 像素时, 必须将 COOF 位置 0。
4. 能通过 SRCDIRX, Y 位选择源数据的参照方向。
5. 能通过 DSTDIRX, Y 位选择绘图方向。
6. 如果 $\alpha E=1$, 就将源数据和底层数据进行 α 混合, 并且进行描绘。在 $\alpha E=1$ 时, 必须使 ROP 代码 = H'CC (源复制)。在 ARGB 格式时, 不对 A 值进行 α 混合。根据绘图控制寄存器 (RCLR) 的源 A 值使用位 (SAU) 和 A 值 (AVALUE) 的设定进行 A 值的描绘。 α 混合在 16 位 / 像素时有效。
7. 能进行 16 种光栅操作。在 ARGB 格式时, 不对 A 值进行光栅操作。根据绘图控制寄存器 (RCLR) 的源 A 值使用位 (SAU) 和 A 值 (AVALUE) 的设定进行 A 值的描绘。

【注】 在描绘矩形时, 进行系统裁剪或者 (相对) 用户裁剪。只在中心坐标进行 Z 裁剪。

(d) 例



(2) BITBLTB

(a) 功能

将 2 值 (1 位 / 像素) 矩形的源数据进行颜色展开, 然后传送到目标区域。

(b) 命令格式

- SS=1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																				
OP CODE=1010_0001								Reserve(all 0)								Draw Mode																																																			
Reserve(all 0)								0								0								ROP																																											
Color1																Color0																																																			
0				0				0				0				0				0				0				0				0				TXS($0 \leq \text{TXS} \leq 4088$)				0				0				0				0				TYS($0 \leq \text{TYS} \leq 4095$)											
0				0				0				0				0				0				0				0				0				0				LW($0 \leq \text{LW} \leq 4087$)				0				0				0				0				RW($0 \leq \text{RW} \leq 4087$)							
0				0				0				0				0				0				0				0				0				0				0				TH($0 \leq \text{TH} \leq 4094$)				0				0				0				0				BH($0 \leq \text{BH} \leq 4094$)			
符号																BXC($-32768 \leq \text{BXC} \leq 32767$)																符号																BYC($-32768 \leq \text{BYC} \leq 32767$)																			

- 【注】
1. $0 \leq \text{TXS} \leq \text{SSTR} - (\text{LW} + \text{RW} + 1)$ 、 $0 \leq \text{TYS} \leq 4096 - (\text{TH} + \text{BH} + 1)$ (SSTR: 源跨距寄存器的设定值)
 2. $8 \leq \text{LW} + \text{RW} + 1 \leq 4088$ (8 的倍数)、 $1 \leq \text{TH} + \text{BH} + 1 \leq 4095$
 3. $-32768 \leq \text{BXC} - \text{LW} \leq 32767$ 、 $-32768 \leq \text{BYC} - \text{TH} \leq 32767$ 、 $-32768 \leq \text{BXC} + \text{RW} \leq 32767$ 、 $-32768 \leq \text{BYC} + \text{BH} \leq 32767$

- SS=0 并且 REL=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																								
OP CODE=1010_0001								Reserve(all 0)								Draw Mode																																																							
Reserve(all 0)								0								0								ROP																																															
Color1																Color0																																																							
0				0				0				Base Address(long word address)																0				0				0																																			
0				0				0				0				0				0				0				0				0				0				LW($0 \leq \text{LW} \leq 4087$)				0				0				0				0				RW($0 \leq \text{RW} \leq 4087$)											
0				0				0				0				0				0				0				0				0				0				0				0				TH($0 \leq \text{TH} \leq 4094$)				0				0				0				0				BH($0 \leq \text{BH} \leq 4094$)			
符号																BXC($-32768 \leq \text{BXC} \leq 32767$)																符号																BYC($-32768 \leq \text{BYC} \leq 32767$)																							

- 【注】
1. $8 \leq \text{LW} + \text{RW} + 1 \leq 4088$ (8 的倍数)、 $1 \leq \text{TH} + \text{BH} + 1 \leq 4095$
 2. $-32768 \leq \text{BXC} - \text{LW} \leq 32767$ 、 $-32768 \leq \text{BYC} - \text{TH} \leq 32767$ 、 $-32768 \leq \text{BXC} + \text{RW} \leq 32767$ 、 $-32768 \leq \text{BYC} + \text{BH} \leq 32767$

- SS=0 并且 REL=1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=1010_0010								Reserve(all 0)								Draw Mode															
Reserve(all 0)								0	0	0	0	0	0	0	0	ROP															
Color1								Color0																							
符号扩展		符号		Base Address(long word address)														0	0												
0	0	0	0	LW($0 \leq LW \leq 4087$)								0	0	0	0	RW($0 \leq RW \leq 4087$)															
0	0	0	0	TH($0 \leq TH \leq 4094$)								0	0	0	0	BH($0 \leq BH \leq 4094$)															
符号		BXC($-32768 \leq BXC \leq 32767$)								符号		BYC($-32768 \leq BYC \leq 32767$)																			

- 【注】
1. $8 \leq LW+RW+1 \leq 4088$ (8 的倍数)、 $1 \leq TH+BH+1 \leq 4095$
 2. $-32768 \leq BXC-LW \leq 32767$ 、 $-32768 \leq BYC-TH \leq 32767$ 、 $-32768 \leq BXC+RW \leq 32767$ 、 $-32768 \leq BYC+BH \leq 32767$
 3. 配置命令码的地址 (长字: 32 位单位) +Base Address (长字: 32 位单位) 必须为 4 个字地址 (64 位单位)。

1. CODE

B'10100001

2. 绘图属性

参照数据				描绘的坐标系	
多值源	2 值源	2 值工作	指定的颜色	绘图坐标	工作坐标
	○	○ (仅 WORK=1)		○	○

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	0 固定	CLIP	RCLIP	STRANS	DTRANS	WORK	SS	REL	SRCDIRX	SRCDIRY	DSTDIRX	DSTDIRY	COOF	αE	0 固定

3. 命令参数

TXS、TYS : 源起始点。必须给未使用位写 0。

Base Address : 源起始绝对地址 (4 个字地址, 必须给 A31~29 和 A2~A0 写 0。)

源起始相对地址 (长字地址, 负数为 2 的补码。必须对 A31~A29 用 A28 的符号位进行符号扩展, 给 A1~A0 写 0。)

【注】 在 32 位地址模式中, 必须将指定的 32 位地址的 bit28 ~ bit3 写到 A28 ~ A3。

BXC、BYC : 中心 X、Y 坐标值, 绘图坐标 (绝对坐标)。负数为 2 的补码。

LW、RW : Left 宽、Right 宽、(BXC, BYC) 的相对值、绘图坐标。必须用像素数进行设定, 给未使用位写 0。

TH、BH : Top 高、Bottom 高、(BXC, BYC) 的相对值、绘图坐标。必须用像素数进行设定, 给未使用位写 0。

ROP : 光栅操作码。

COLOR0、COLOR1 : 指定 8 或者 16 位/像素颜色。对于 16 位/像素, 必须符合目标像素格式。在 8 位/像素时, 必须给高位和低位字节设定相同的值。

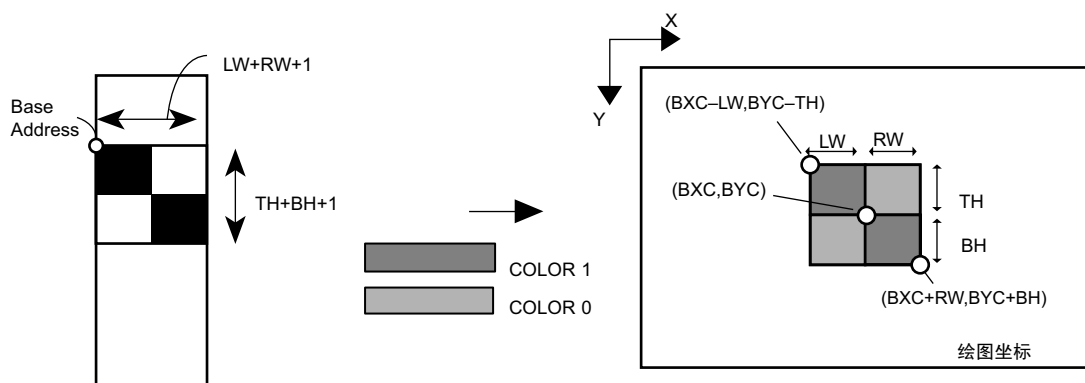
(c) 说明

将 2 值 (1 位 / 像素) 矩形的源数据传送到绘图坐标。
 $LW+RW+1$ 与 SS 位的值无关, 必须为 8 像素的倍数。

1. 当通过绘图属性选择工作指定 ($WORK=1$), 就参照工作坐标 (坐标与绘图坐标相同), 同时在绘图坐标中描绘工作坐标的像素为 1 的部分。
2. 将 2 值源数据线性配置到存储器。在 $REL=0$ 时, 能将源地址指定为绝对地址; 在 $REL=1$ 时, 能将源地址指定为配置 $BITBLTB$ 命令的命令码的存储地址的相对地址。
3. 在进行 16 位 / 像素描绘时, 如果将绘图属性的 $COOF$ 位置 1, 就给 2 值源数据进行颜色展开后的数据加上 $COFSR$ 寄存器的值, 然后描绘所得的结果。此时, 通过饱和处理进行运算。在 8 位 / 像素时, 必须将 $COOF$ 位置 0。
4. 能通过 $SRCDIRX$, Y 位选择源数据的参照方向。
5. 能通过 $DSTDIRX$, Y 位选择绘图方向。
6. 如果 $\alpha E=1$, 就将 2 值源数据进行颜色展开后的数据和底层数据进行 α 混合, 并且进行描绘。在 $\alpha E=1$ 时, 必须使 ROP 代码 = $H'CC$ (源复制)。在 $ARGB$ 格式时, 不对 A 值进行 α 混合。根据绘图控制寄存器 ($RCLR$) 的源 A 值使用位 (SAU) 和 A 值 ($AVALUE$) 的设定进行 A 值的描绘。 α 混合在 16 位 / 像素时有效。
7. 能进行 16 种光栅操作。在 $ARGB$ 格式时, 不对 A 值进行光栅操作。根据绘图控制寄存器 ($RCLR$) 的源 A 值使用位 (SAU) 和 A 值 ($AVALUE$) 的设定进行 A 值的描绘。

【注】 在描绘矩形时, 进行系统裁剪或者 (相对) 用户裁剪。只在中心坐标进行 Z 裁剪。

(d) 例



(3) BITBLTC

(a) 功能

在目标区域中，通过单色的指定进行矩形的描绘。

(b) 命令格式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
OP CODE=1010_0000								Reserve(all 0)								Draw Mode																	
Reserve(all 0)								Reserve(all 0)								0	0	0	0	0	0	0	0	ROP									
Reserve(all 0)								Reserve(all 0)								Color																	
0	0	0	0	LW(0 ≤ LW ≤ 4094)								0	0	0	0	RW(0 ≤ RW ≤ 4094)																	
0	0	0	0	TH(0 ≤ TH ≤ 4094)								0	0	0	0	BH(0 ≤ BH ≤ 4094)																	
符号	BXC(-32768 ≤ BXC ≤ 32767)														符号	BYC(-32768 ≤ BYC ≤ 32767)																	

- 【注】 1. $1 \leq LW+RW+1 \leq 4095$ 、 $1 \leq TH+BH+1 \leq 4095$
 2. $-32768 \leq BXC-LW \leq 32767$ 、 $-32768 \leq BYC-TH \leq 32767$ 、 $-32768 \leq BXC+RW \leq 32767$ 、
 $-32768 \leq BYC+BH \leq 32767$

- CODE
B'10100000
- 绘图属性

参照数据				描绘的坐标系	
多值源	2 值源	2 值工作	指定的颜色	绘图坐标	工作坐标
		○ (仅 WORK=1)	○	○	

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MTRE	0 固定	CLIP	RCLIP	0 固定	DTRANS	WORK	0 固定	0 固定	0 固定	0 固定	DSTDIRX	DSTDIRY	COOF	αE	0 固定

- 命令参数
 - BXC、BYC : 中心 X、Y 坐标值、绘图坐标 (绝对坐标)。负数为 2 的补码。
 - LW、RW : Left 宽、Right 宽、(BXC, BYC) 的相对值、绘图坐标。必须用像素数进行设定，给未使用位写 0。
 - TH、BH : Top 高、Bottom 高、(BXC, BYC) 的相对值、绘图坐标。必须用像素数进行设定，给未使用位写 0。
 - COLOR : 指定 8 或者 16 位/像素颜色。对 16 位/像素，必须符合目标像素格式。在 8 位/像素时，必须给高位和低位字节设定相同的值。
 - ROP : 光栅操作码。

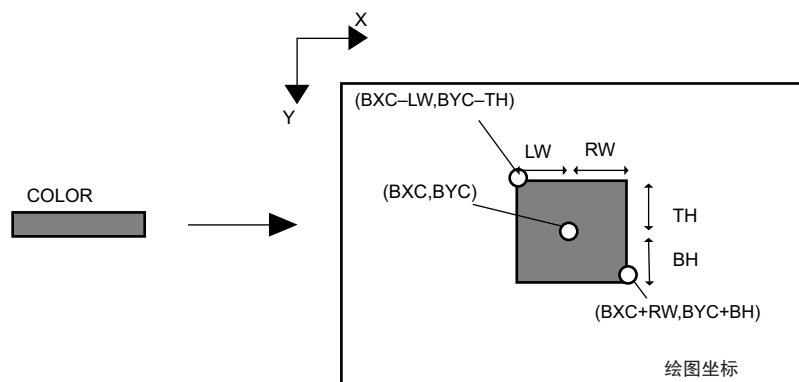
(c) 说明

在目标区域中，通过参数 COLOR 指定的单色进行矩形的描绘。

1. 当通过绘图属性选择工作指定 (WORK=1) 时，就参照工作坐标 (坐标与绘图坐标相同)，同时在绘图坐标中描绘工作坐标的像素为 1 的部分。
2. 在进行 16 位/像素描绘时，如果将绘图属性的 COOF 位置 1，就给指定颜色的值加上 COFSR 寄存器的值，然后描绘所得的结果。此时，通过饱和处理进行运算。在 8 位/像素时，必须将 COOF 位置 0。
3. 能通过 DSTDIRX, Y 位选择绘图方向。
4. 如果 $\alpha E=1$ ，就将指定的颜色数据和底层数据进行 α 混合，并且进行描绘。在 $\alpha E=1$ 时，必须使 ROP 代码=H'CC (源复制)。在 ARGB 格式时，不对 A 值进行 α 混合。根据绘图控制寄存器 (RCLR) 的源 A 值使用位 (SAU) 和 A 值 (AVALUE) 的设定进行 A 值的描绘。 α 混合在 16 位/像素时有效。
5. 能进行 16 种光栅操作。在 ARGB 格式时，不对 A 值进行光栅操作。根据绘图控制寄存器 (RCLR) 的源 A 值使用位 (SAU) 和 A 值 (AVALUE) 的设定进行 A 值的描绘。

【注】 在描绘矩形时，进行系统裁剪或者 (相对) 用户裁剪。只在中心坐标进行 Z 裁剪。

(d) 例



23.2.6 控制命令

(1) MOVE

(a) 功能

设定当前指针。

(b) 命令格式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=0100_1000								Reserve(all 0)								Draw Mode															
XC(-32768 ≤ XC ≤ 32767)																YC(-32768 ≤ YC ≤ 32767)															

1. CODE

B'01001000

2. 绘图属性

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定

3. 命令参数

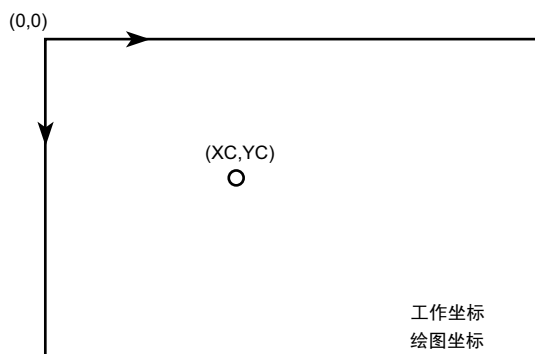
XC: 绘图坐标 (绝对坐标) 或者工作坐标 (绝对坐标)。负数为 2 的补码。

YC: 绘图坐标 (绝对坐标) 或者工作坐标 (绝对坐标)。负数为 2 的补码。

(c) 说明

给当前指针设定 XC 和 YC 加上局部偏移值后的值。用绝对坐标设定 XC 和 YC。当前指针只用于相对类绘图命令，必须在发行 MOVE 命令后连续使用相对类绘图命令。如果在中途使用绝对类绘图命令，就将当前指针用作内部运算的寄存器，使当前指针的值遭到破坏。因此，在再次使用相对类绘图命令前，必须先发行 MOVE 命令。

(d) 例



(2) RMOVE

(a) 功能

给当前指针加上 XC 和 YC。

(b) 命令格式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=0100_1100								Reserve(all 0)								Draw Mode															
XC(-32768 ≤ XC ≤ 32767)																YC(-32768 ≤ YC ≤ 32767)															

1. CODE

B'01001100

2. 绘图属性

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定

3. 命令参数

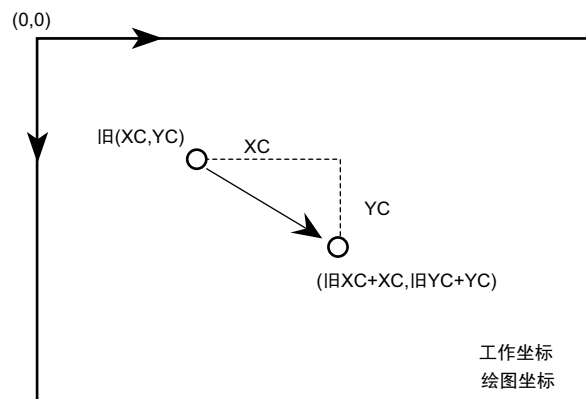
XC: 绘图坐标 (相对坐标) 或者工作坐标 (相对坐标)。负数为 2 的补码。

YC: 绘图坐标 (相对坐标) 或者工作坐标 (相对坐标)。负数为 2 的补码。

(c) 说明

给当前指针加上 XC 和 YC。

(d) 例



(3) LCOFS

(a) 功能

设定目标区域和工作区的偏移值（局部偏移）。

(b) 命令格式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=0100_0000								Reserve(all 0)								Draw Mode															
XO(-32768 ≤ XO ≤ 32767)																YO(-32768 ≤ YO ≤ 32767)															

1. CODE

B'01000000

2. 绘图属性

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定

3. 命令参数

XO: 局部偏移值，绘图坐标（绝对坐标）或者工作坐标（绝对坐标）。负数为2的补码。

YO: 局部偏移值，绘图坐标（绝对坐标）或者工作坐标（绝对坐标）。负数为2的补码。

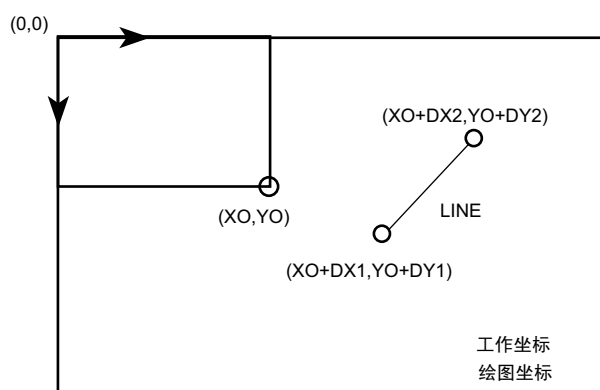
(c) 说明

在设定局部偏移值后，所有绘图命令指定的坐标都加上此偏移值。

必须在显示列表的开头处进行设定（初始值为不定值）。

在要让当前指针反映局部偏移值时，必须在 LCOFS 命令后发行 MOVE 命令。

(d) 例



(4) RLCOFS

(a) 功能

给局部偏移值加上 XO 和 YO。

(b) 命令格式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=0100_0100								Reserve(all 0)								Draw Mode															
XC(-32768 ≤ XO ≤ 32767)																YC(-32768 ≤ YO ≤ 32767)															

1. CODE

B'01000100

2. 绘图属性

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定

3. 命令参数

XO: 局部偏移值, 绘图坐标 (相对坐标) 或者工作坐标 (相对坐标)。负数为 2 的补码。

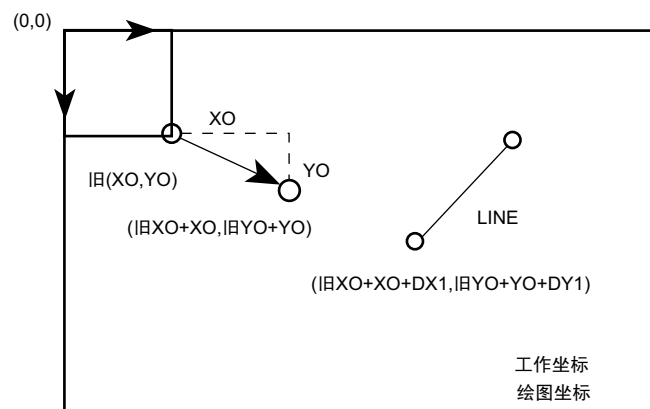
YO: 局部偏移值, 绘图坐标 (相对坐标) 或者工作坐标 (相对坐标)。负数为 2 的补码。

(c) 说明

给局部偏移值加上 XO 和 YO 的值产生新的局部偏移值。在此设定后, 所以绘图命令指定的坐标都加上此偏移值。

在要让当前指针反映局部偏移值时, 必须在通过 LCOFS 或者 RLCOFS 命令设定局部偏移值后执行 MOVE 命令。

(d) 例



(5) WPR

(a) 功能

给特定的地址映像寄存器设定值。

(b) 命令格式

- LINKE=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
OP CODE=0001_1000								Reserve(all 0)								Draw Mode																	
Reserve(all 0)								n-1(0 ≤ n-1 ≤ 255)								0	0	0	0	0	0	0	0	W Reg No									
Data0																																	
•																																	
•																																	
Data n-1																																	

- LINKE=1 并且 LREL=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
OP CODE=0001_1000								Reserve(all 0)								Draw Mode																	
Reserve(all 0)								n-1(0 ≤ n-1 ≤ 255)								0	0	0	0	0	0	0	0	W Reg No									
0	0	0	LINK Address(long word address)																								0	0					

【注】 由于将 LINK Address 的下一个长字解释为下一个命令码，所以不能将配置 LINK Address 的下一个长字地址指定为 LINK Address 所指的连接目标地址。

- LINKE=1 并且 LREL=1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
OP CODE=0001_1000								Reserve(all 0)								Draw Mode																	
Reserve(all 0)								n-1(0 ≤ n-1 ≤ 255)								0	0	0	0	0	0	0	0	W Reg No									
符号扩展		符号	LINK Address(long word address)																								0		0				

【注】 由于将 LINK Address 的下一个长字解释为下一个命令码，所以不能将配置 LINK Address 的下一个长字地址指定为配置命令码的地址 +LINK Address 所指的连接目标地址。

1. CODE
B'00011000
2. 绘图属性

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0 固定	0 固定	0 固定	0 固定	0 固定	LINKE	LREL	0 固定	0 固定	0 固定	0 固定	0 固定	ByteM3	ByteM2	ByteM1	ByteM0

3. 参数

- W reg No : 寄存器编号
 Data n (n=1 ~ 256) : 要写的的数据。
 n-1 : 写数据的个数。
 LINK Address : LINK 绝对地址 (长字地址, 必须给 A31 ~ 29、A1 和 A0 写 0。)
 LINK 相对地址 (长字地址, 负数为 2 的补码。必须对 A31 ~ A29 用 A28 的符号位进行符号扩展, 给 A1 ~ A0 写 0。)

【注】 在 32 位地址模式中, 必须将指定的 32 位地址的 bit28 ~ bit3 写到 A28 ~ A3。

(c) 说明

给地址映像寄存器写数据。给 W reg No 设定寄存器号, 给 Data n 设定要写的的数据。不能与 CPU 的存取发生冲突。

1. 如果设定 LINKE 位, 就从存储器的 LINK Address 读数据, 然后写寄存器。
2. 通过 LREL 位, 能将 LINK Address 指定为绝对地址或者配置 WPR 命令的命令码的存储地址的相对地址。
3. 如果将 ByteM3 ~ 0 位置 “1”, 就能以字节为单位屏蔽寄存器的写操作。

(6) JUMP

(a) 功能

更改显示列表的取地址。

(b) 命令格式

- REL=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=0010_1000								Reserve(all 0)								Draw Mode															
0	0	0	JUMP Address(long word address)														0	0													

- REL=1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=0010_1000								Reserve(all 0)								Draw Mode															
符号扩展		符号	JUMP Address(long word address)														0		0												

1. CODE
B'00101000
2. 绘图属性

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	REL	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定

3. 命令参数

JUMP Address: 转移目标的绝对地址（长字地址，必须给 A31 ~ 29、A1 和 A0 写 0。）

转移目标的相对地址（长字地址，负数为 2 的补码。必须对 A31 ~ A29 用 A28 的符号位进行符号扩展，给 A1 ~ A0 写 0。）

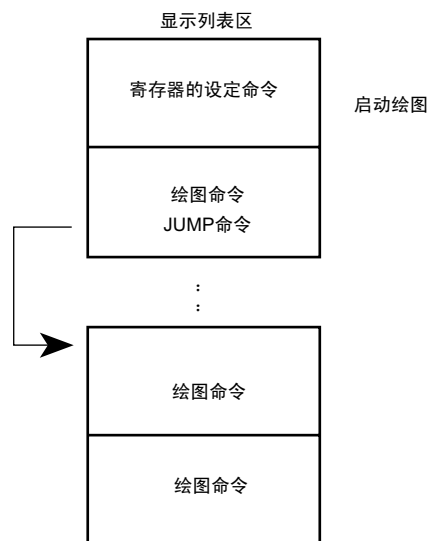
【注】 在 32 位地址模式中，必须将指定的 32 位地址的 bit28 ~ bit3 写到 A28 ~ A3。

(c) 说明

将显示列表的取地址更改为指定的地址。

在 REL=0 时，能将转移目标地址指定为绝对地址。REL=1 时，能将转移目标地址指定为配置命令码的存储地址的相对地址。

(d) 例



(7) GOSUB

(a) 功能

对显示列表进行子程序调用。

(b) 命令格式

- REL=0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=0011_0000								Reserve(all 0)								Draw Mode															
0	0	0	GOSUB Address(long word address)														0	0													

- REL=1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=0011_0000								Reserve(all 0)								Draw Mode															
符号扩展		符号	GOSUB Address(long word address)														0	0													

1. CODE
B'00110000
2. 绘图属性

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	REL	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	No

3. 命令参数

GOSUB Address: 子程序的绝对地址（长字地址，必须给 A31～29、A1 和 A0 写 0。）
子程序的相对地址（长字地址，负数为 2 的补码。必须对 A31～A29 用 A28 的符号位进行符号扩展，给 A1～A0 写 0。）

【注】 在 32 位地址模式中，必须将指定的 32 位地址的 bit28～bit3 写到 A28～A3。

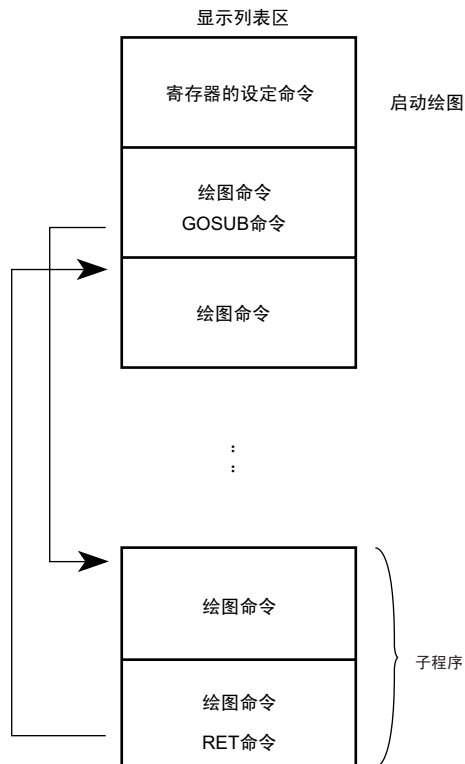
(c) 说明

将显示列表的取地址更改为指定的子程序地址。通过 RET 指令恢复取地址。但是，嵌套层为 1，所以如果在子程序中进行子程序调用，就不能返回。

在 REL=0 时，能将子程序地址指定为绝对地址；在 REL=1 时，能将子程序地址指定为配置命令码的存储地址的相对地址。

在 No 位为 0 时，给返回地址 0 寄存器 (RTN0R) 设定返回地址；为 1 时，给返回地址 1 寄存器 (RTN1R) 设定返回地址。

(d) 例



(8) RET

(a) 功能

通过 GOSUB 命令从子程序调用返回。

(b) 命令格式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=0011_1000								Reserve(all 0)								Draw Mode															

1. CODE
B'00111000
2. 绘图属性

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	No

(c) 说明

将显示列表的取地址恢复为子程序调用源的下一个地址。

在 No 位为 0 时，给地址 0 寄存器 (RTN0R) 设定返回地址；为 1 时，给地址 1 寄存器 (RTN1R) 设定返回地址。

(9) NOP/INT

(a) 功能

不执行任何操作。

(b) 命令格式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=0000_1000								Reserve(all 0)								Draw Mode															

1. CODE

B'00001000

2. 绘图属性

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
INT	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	INT No							

(c) 说明

不进行任何操作，取下一条指令。但是，如果设定 INT 位，就在取此命令后将状态寄存器 (SR) 的 INT 位置“1”，并且将 INT No 保存到中断命令 ID 寄存器 (ICIDR)，然后停止描绘操作。当清除状态寄存器的 INT 位时，就从下一条命令开始重新描绘。

(10) VBKEM

(a) 功能

实现与帧改变时序的同步。

(b) 命令格式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=0001_0000								Reserve(all 0)								Draw Mode															

1. CODE

B'00010000

2. 绘图属性

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定

(c) 说明

如果执行此命令，就在帧改变时序前描绘操作处于待机状态。然后，在经过帧改变时序时，将控制转移到下一条命令。在逐行模式显示或者隔行同步 & 视频模式显示中，帧改变时序是下一个 VSYNC；在隔行同步模式显示中，是下一个帧的起始点。

【注】 只能在手动显示转换模式或者自动绘图模式中使用。

(11) TRAP

(a) 功能

表示显示列表的结束。

(b) 命令格式

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE=0000_0000								Reserve(all 0)								Draw Mode															

1. CODE
B'00000000
2. 绘图属性

Draw Mode															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	0 固定	Flip5	Flip4	Flip3	Flip2	Flip1	Flip0

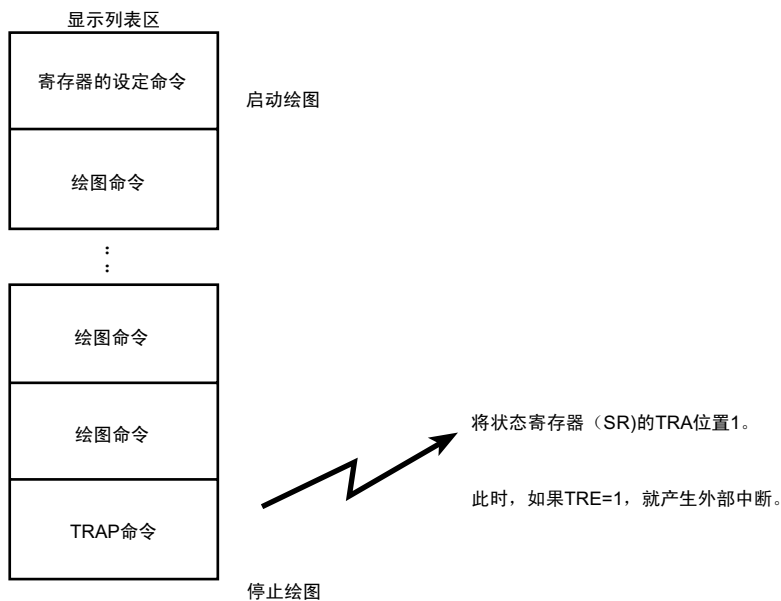
(c) 说明

停止描绘操作，并且将状态寄存器 (SR) 的 TRA 置 1。此时，如果中断允许寄存器 (IER) 的 TRE 被置 1，就向 CPU 产生中断。

在显示列表的最后需要此命令。

如果设定 FlipE5 ~ 0，就翻动对应的平 (只在自动绘图模式时有效)。在逐行模式显示或者隔行同步 & 视频模式显示时，翻动时序是下一个 VSYNC；在隔行同步模式显示时，是下一个帧起始点。

(d) 例



23.3 寄存器规格

在绘图开始后到执行 TRAP 命令期间（用 INT 命令指定的停止绘图期间除外）禁止 CPU 写系统控制寄存器以外的其他寄存器。但是，对于中断允许寄存器，如果 CPU 的写操作与 WPR 命令的写操作发生冲突，就优先 CPU 的写操作。在没有特别指明时，以下的“复位”表示硬件复位和软件复位，硬件复位是指上电复位。

表 23.5 寄存器结构

分类	名称	略称	R/W	WPR*1	P4 区地址 *2	区域 7 地址 *2	存取长度
系统控制	系统控制	SCLR	R/W	×	H'FFEA 0000	H'1FEA 0000	32
	状态	SR	R	×	H'FFEA 0004	H'1FEA 0004	32
	状态寄存器清除	SRCR	W	×	H'FFEA 0008	H'1FEA 0008	32
	中断允许	IER	R/W	○	H'FFEA 000C	H'1FEA 000C	32
	中断命令 ID	ICIDR	R	×	H'FFEA 0010	H'1FEA 0010	32
存储器控制	返回地址 0	RTN0R	R	○	H'FFEA 0040	H'1FEA 0040	32
	返回地址 1	RTN1R	R	○	H'FFEA 0044	H'1FEA 0044	32
	显示列表的起始地址	DLSAR	R/W	×	H'FFEA 0048	H'1FEA 0048	32
	2 维源区域起始地址	SSAR	R/W	○	H'FFEA 004C	H'1FEA 004C	32
	绘图起始地址	RSAR	R/W	○	H'FFEA 0050	H'1FEA 0050	32
	工作区起始地址	WSAR	R/W	○	H'FFEA 0054	H'1FEA 0054	32
	源跨距	SSTRR	R/W	○	H'FFEA 0058	H'1FEA 0058	32
	目标跨距	DSTRR	R/W	○	H'FFEA 005C	H'1FEA 005C	32
	字节排序转换控制	ENDCVR	R/W	×	H'FFEA 0060	H'1FEA 0060	32
颜色控制	源透明色	STCR	R/W	○	H'FFEA 0080	H'1FEA 0080	32
	目标透明色	DTCR	R/W	○	H'FFEA 0084	H'1FEA 0084	32
	α 值	ALPHR	R/W	○	H'FFEA 0088	H'1FEA 0088	32
	颜色偏移	COFSR	R/W	○	H'FFEA 008C	H'1FEA 008C	32
绘图控制	绘图控制	RCLR	R/W	○	H'FFEA 00C0	H'1FEA 00C0	32
	命令状态	CSTR	R	×	H'FFEA 00C4	H'1FEA 00C4	32
	当前指针	CURR	R	×	H'FFEA 00C8	H'1FEA 00C8	32
	局部偏移	LCOR	R	×	H'FFEA 00CC	H'1FEA 00CC	32
	系统裁剪区 MAX	SCLMAR	R	○	H'FFEA 00D0	H'1FEA 00D0	32
	用户裁剪区 MIN	UCLMIR	R	○	H'FFEA 00D4	H'1FEA 00D4	32
	用户裁剪区 MAX	UCLMAR	R	○	H'FFEA 00D8	H'1FEA 00D8	32
	相对用户裁剪区 MIN	RUCLMIR	R	○	H'FFEA 00DC	H'1FEA 00DC	32
	相对用户裁剪区 MAX	RUCLMAR	R	○	H'FFEA 00E0	H'1FEA 00E0	32
	绘图控制 2	RCL2R	R/W	○	H'FFEA 00F0	H'1FEA 00F0	32
	图案偏移	POFSR	R/W	○	H'FFEA 00F8	H'1FEA 00F8	32
坐标变换控制	坐标变换控制	GTRCR	R/W	○	H'FFEA 0100	H'1FEA 0100	32
	矩阵参数 A	MTRAR	R/W	○	H'FFEA 0104	H'1FEA 0104	32
	矩阵参数 B	MTRBR	R/W	○	H'FFEA 0108	H'1FEA 0108	32
	矩阵参数 C	MTRCR	R/W	○	H'FFEA 010C	H'1FEA 010C	32
	矩阵参数 D	MTRDR	R/W	○	H'FFEA 0110	H'1FEA 0110	32
	矩阵参数 E	MTRER	R/W	○	H'FFEA 0114	H'1FEA 0114	32

分类	名称	略称	R/W	WPR*1	P4 区地址 *2	区域 7 地址 *2	存取长度
坐标变换控制	矩阵参数 F	MTRFR	R/W	○	H'FFEA 0118	H'1FEA 0118	32
	矩阵参数 G	MTRGR	R/W	○	H'FFEA 011C	H'1FEA 011C	32
	矩阵参数 H	MTRHR	R/W	○	H'FFEA 0120	H'1FEA 0120	32
	矩阵参数 I	MTRIR	R/W	○	H'FFEA 0124	H'1FEA 0124	32
	坐标变换偏移 X	GTROFSXR	R/W	○	H'FFEA 0128	H'1FEA 0128	32
	坐标变换偏移 Y	GTROFSYR	R/W	○	H'FFEA 012C	H'1FEA 012C	32
	Z 裁剪区 MIN	ZCLPMINR	R/W	○	H'FFEA 0130	H'1FEA 0130	32
	Z 裁剪区 MAX	ZCLPMAXR	R/W	○	H'FFEA 0134	H'1FEA 0134	32
	Z 饱和值 MIN	ZSATVMINR	R/W	○	H'FFEA 0138	H'1FEA 0138	32

【注】 *1 设定 WPR 命令 ○：可，×：不可

*2 P4 区地址是使用虚拟地址空间 P4 区的地址，区域 7 地址是使用 TLB 存取物理地址空间区域的地址。
禁止写指定地址以外的地址，否则不保证运行。

表 23.6 寄存器位结构

分类	寄存器略称	数据																																			
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
系统控制	SCLR																																				
	SRES																																				
	RS																																				
	SR																																				
	VER																																				
	MTRER																																				
	CER																																				
	INT																																				
	TRA																																				
	SRCR																																				
	MTCL																																				
	CECL																																				
	INCL																																				
	TRCL																																				
	IER																																				
	MTE																																				
	CEE																																				
	INE																																				
	TRE																																				
	ICIDR																																				
存储器控制	RTN0R																																	0	0		
	RTN1R																																	0	0		
	DLSAR																																0	0	0	0	
	SSAR																																0	0	0	0	
	RSAR																																0	0	0	0	
	WSAR																																0	0	0	0	
	SSTRR																																0	0	0	0	
	DSTRR																																	0	0	0	0
	ENDCVR																																				
	LWSWAP																																				
	WSWAP																																				
BYTESWAP																																					
BITSWAP																																					
色彩控制	STCR																																				
	STC1																																				
	STC8																																				
	STC16																																				
	DTCR																																				
	DTC8																																				
	DTC16																																				
	ALPHR																																				
	COFSR																																				
	RGB: 565时	COR																																			
	COG																																				
	COB																																				
	ARGB: 1555时	COR																																			
COG																																					
COB																																					

分类	寄存器名称	数据																																			
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
绘图控制	RCLR																																				
	STP																																				
	DTP																																				
	SPF																																				
	DPF																																				
	GBM																																				
	SAU																																				
	AVALUE																																				
	LPCE																																				
	COM																																				
	CSTR																																				
	CURR																																				
	XC																																				
	YC																																				
	LCOR																																				
	XO																																				
	YO																																				
	SCLMAR																																				
	SXMAX																																				
	SYMAX																																				
	UCLMIR																																				
	UXMIN																																				
	UYMIN																																				
	UCLMAR																																				
	UXMAX																																				
	UYMAX																																				
	RUCLMIR																																				
	RUXMIN																																				
	RUYMIN																																				
	RUCLMAR																																				
	RUXMAX																																				
	RUYMAX																																				
RCL2R																																					
DAE																																					
PSTYLE																																					
PXSIZE																																					
PYSIZE																																					
POFSR																																					
POFSX																																					
POFSY																																					
坐标变换控制	GTRCR																																				
	GTE																																				
	AFE																																				
	MTRAR																																				
	MTRBR																																				
	MTRCR																																				
	MTRDR																																				
MTRER																																					

分类	寄存器名称	数据																																	
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
坐标变换控制	MTRFR																																		
	MTRGR																																		
	MTRHR																																		
	MTRIR																																		
	GTROFSXR																																		
	GTROFSYR																																		
	ZCLPMINR																																		
	ZCLPMAXR																																		
	ZSATVMINR																																		

表 23.7 寄存器硬件复位和软件复位时的初始值

	寄存器名称	硬件复位	软件复位	数据																																	
				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
系统控制	SCLR	○		1																															0		
			○	—																															0		
	SR	○	○	1	0	0	0								0																			0	0	0	
	SRCR	○	○												0																				0	0	0
	IER	○	○												0																				0	0	0
ICIDR	×	×																																	*		
存储器控制	RTN0R	×	×			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*		
	RTN1R	×	×			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*		
	DLSAR	×	×			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*		
	SSAR	×	×			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*		
	DSAR	×	×			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*		
	WSAR	×	×			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*		
	SSTRR	×	×																				*	*	*	*	*	*	*	*	*	*	*	*	*		
	DSTRR	×	×																				*	*	*	*	*	*	*	*	*	*	*	*	*		
	ENDCVR	○	○																																0	0	0
色彩控制	STCR	×	×							*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*		
	DTCR	×	×							*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*		
	ALPHR	×	×																																	*	
	COFSR	×	×	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
绘图控制	RCLR	○	○						0	0			0	0			0	0	0																0	0	
	CSTR	×	×			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	CURR	×	×	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	LCOR	×	×	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	SCLMAR	×	×			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	UCLMIR	×	×			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	UCLMAR	×	×			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	RUCLMIR	×	×			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	RUCLMAR	×	×			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	RCL2R	○	○	0	0						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
POF3R	○	○			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

	寄存器略称	硬件复位	软件复位	数据																																
				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
坐标变换控制	GTRCR	○	○	0																															0	
	MTRAR	×	×	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	MTRBR	×	×	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	MTRCR	×	×	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	MTRDR	×	×	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	MTRER	×	×	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	MTRFR	×	×	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	MTRGR	×	×	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	MTRHR	×	×	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	MTRIR	×	×	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	GTROFSXR	×	×																																	
	GTROFSYR	×	×																																	
	ZCLPMIN	×	×	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	ZCLPMAX	×	×	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
	ZSATVMINR	×	×	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	

【符号说明】

- : 不定值。在硬件复位和软件复位时，值保持不变。
- 0 : 在硬件复位和软件复位时，被初始化为0。
- 1 : 在硬件复位时，被初始化为1。
- : 保留位。读取值为0，只能写0。
- * : 保留位。在硬件复位和软件复位时，值保持不变。读取值为不定值，只能写0。
- 0 : 保留位。在硬件复位和软件复位时，被初始化为0。只能写0。
- 1 : 保留位。在硬件复位和软件复位时，被初始化为1。只能写1。

23.3.1 系统控制寄存器

(1) 系统控制寄存器 (SCLR)

偏移: H'000

初始值: H'8000000

系统控制寄存器 (SCLR) 是 32 位可读写寄存器, 设定系统的运行。在硬件复位时, SCLR 被初始化为:

- SRES 位为 1。
- RS 位为 0。
禁止设定 SRES=1 和 RS=1。
- bit31: 软件复位 (SRES)
对 G2D 进行复位。

bit31	说明
SRES	
0	允许进行命令的处理。
1	<p>当进行硬件复位时, SRES=1。(初始值) 在初始化时, 必须置 0。 如果通过软件将此位置 1, 就对描绘操作进行复位。G2D 的寄存器值也被初始化。 除了此寄存器以外, 在 SRES=1 的期间, 不能写其他寄存器。</p> <p>【注】 为了使本 LSI 正确反映软件复位, 需要和存储器的存取相同的写存取反映和确认方法。因此, 在转移到软件复位后, 必须在解除复位前进行以下处理。</p> <ol style="list-style-type: none"> 1. 在 G2D 的优先级与 CPU 相同时, 必须至少虚读 3 次 SDRAM 的任意区域。 2. 在 G2D 为 2 级而 CPU 为 3 级时, 必须至少虚读 1 次 SDRAM 的任意区域。 3. 在 G2D 为 3 级而 CPU 为 2 级时, 必须结束对 G2D 除外的 Level2 和 Level3 模块的 SDRAM 存取。

- bit30~1: 保留位
只能写 0, 读取值为 0。
- bit0: 绘图开始 (RS)
指定绘图开始。在绘图期间 (在绘图开始后到执行 TRAP 命令的期间) 禁止写 1。

bit0	说明
RS	
0	不开始绘图。(初始值)
1	开始绘图。在开始绘图后, 此位被清 0。

(2) 状态寄存器 (SR)

偏移: H'004

初始值: H'8000000

状态寄存器 (SR) 是 32 位只读寄存器, 用于从外部读取内部状态。在复位时, SR 被初始化为:

- Ver 标志为 1000。
- 其他标志为 0。
- bit31~28: 版本标志 (VER)
读取值为 1000。
- bit18: 矩阵运算错误标志 (MTRER)
表示坐标变换矩阵的运算结果 TX、TY、W 超过范围并且进行了饱和处理。

【注】 因为坐标变换控制寄存器的坐标变换允许位 (GTE) 和绘图属性 (MTRE) 不屏蔽 MTRER, 所以在 GTE=0 或者 1 并且 MTRE=0 时, 即使不进行坐标变换, MTRER 也可能被置 1。因此, 从绘图开始到执行 TRAP 命令的期间, 除非 GTE=1 并且 MTRE=1, 否则就不能使用 MTRER。

bit18	说明
MTRER	
0	是正常状态。表示在通过 SCLR 的 SRES 位或者 SRCR 的 MTCL 位清除 MTRER 标志后, 坐标变换矩阵的运算结果 TX、TY、W 未超过范围 (未进行饱和处理)。(初始值)
1	表示坐标变换矩阵的运算结果 TX、TY、W 超过范围, 并且进行了饱和处理。不停止描绘操作。在复位或者清除 SRCR 前, MTRER 标志保持状态。

- bit2: 命令错误标志 (CER)
表示取到错误命令。

bit2	说明
CER	
0	是正常状态。表示在通过 SCLR 的 SRES 位或者 SRCR 的 CECL 位清除 CER 标志后, 未取到错误命令。 错误命令是指命令码的高 8 位未被定义。不检查低 16 位的绘图属性的错误。(初始值)
1	是描绘操作的停止状态。表示在通过 SCLR 的 SRES 位或者 SRCR 的 CECL 位清除 CER 标志后, 取到错误命令, 并且进入描绘操作的停止状态。在重新开始描绘操作时, 必须在软件复位后开始绘图。 在复位或者清除 SRCR 前, CER 标志保持状态。

- bit1: 中断标志 (INT)
表示取到 NOP/INT 命令 (只在绘图属性的 INT 位 =1 时)

bit1	说明
INT	
0	表示在通过 SCLR 的 SRES 位或者 SRCR 的 INCL 位清除 INT 标志后, 未取到 NOP/INT 命令。 (初始值)
1	描绘操作的停止状态。表示在通过 SCLR 的 SRES 位或者 SRCR 的 INCL 位清除 INT 标志后, 取到 NOP/INT 命令, 并且进入描绘操作的停止状态 (只在绘图属性的 INT 位 =1 时)。 如果通过 SRCR 的 INCL 位清除 INT 标志, 就从下一个命令重新开始描绘。 在复位或者清除 SRCR 前, INT 标志保持状态。 【注】 不能在绘图停止期间通过 INT 命令改写显示列表。

- bit0: 陷阱标志 (TRA)
表示命令执行结束。

bit0	说明
TRA	
0	表示在通过 SCLR 的 SRES 位或者 SRCR 的 TRCL 位清除 TRA 标志后, 未取到 TRAP 命令。(初始值)
1	表示命令执行结束或者未执行当前命令。 在复位或者清除 SRCR 前, TRA 标志保持状态。

- bit27~19、17~3: 保留位
读取值为0。

(3) 状态寄存器清除寄存器 (SRCR)

偏移: H'008

初始值: H'0000000

状态寄存器清除寄存器 (SRCR) 是 32 位只写寄存器, 用于清除对应状态寄存器 (SR) 的各标志。在清除 SR 后, SRCR 在内部被清为全 0 (读取值为 0)。

位	位名	略称	说明
18	矩阵运算错误标志清除	MTCL	当将 MTCL 位置 1 时, 就将 SR 的 MTRER 标志清 0。
2	命令错误标志清除	CECL	当将 CECL 位置 1 时, 就将 SR 的 CER 标志清 0。
1	中断标志清除	INCL	当将 INCL 位置 1 时, 就将 SR 的 INT 标志清 0。
0	陷阱标志清除	TRCL	当将 TRCL 位置 1 时, 就将 SR 的 TRA 标志清 0。
31~19、17~3	保留位	—	只能写 0。

(4) 中断允许寄存器 (IER)

偏移: H'00C

初始值: H'0000000

中断允许寄存器 (IER) 是 32 位可读写寄存器, 允许或者禁止对应状态寄存器 (SR) 的各标志中断。如果将 IER 中为 1 的位所对应的 SR 的位置 1, 就向 CPU 请求中断。中断的产生条件如下:

中断产生条件 = a+b+c+d

a=MTRER · MTE

b=CER · CEE

c=INT · INE

d=TRA · TRE

- bit18: 矩阵运算错误标志允许 (MTE)
允许或者禁止 SR 的 MTRER 标志中断。

【注】 因为坐标变换控制寄存器的坐标变换允许位 (GTE) 和绘图属性的 (MTRE) 不屏蔽 MTRER, 所以在 GTE=0 或者 1 并且 MTRE=0 时, 即使不进行坐标变换, MTRER 也可能被置 1, 产生预想外的中断。因此, 从绘图开始到执行 TRAP 命令的期间, 除非 GTE=1 并且 MTRE=1, 否则不能使用 MTEER。

bit18	说明
MTE	
0	禁止 SR 的 MTRER 标志中断。(初始值)
1	允许 SR 的 MTRER 标志中断。

- bit2: 命令错误标志允许 (CEE)
允许或者禁止 SR 的 CER 标志中断。

bit2	说明
CEE	
0	禁止 SR 的 CER 标志中断。(初始值)
1	允许 SR 的 CER 标志中断。

- bit1: 中断标志允许 (INE)
允许或者禁止 SR 的 INT 标志中断。

bit1	说明
INE	
0	禁止 SR 的 INE 标志中断。(初始值)
1	允许 SR 的 INE 标志中断。

- bit0: 陷阱标志允许 (TRE)
允许或者禁止 SR 的 TRA 标志中断。

bit0	说明
TRE	
0	禁止 SR 的 TRA 标志中断。(初始值)
1	允许 SR 的 TRA 标志中断。

- bit31~19、17~3: 保留位
只能写0。

(5) 中断命令 ID 寄存器 (ICIDR)

偏移: H'010

初始值: 不定值

中断命令 ID 寄存器 (ICIDR) 是 32 位只读寄存器, 在取 NOP/INT 命令时, 如果绘图属性的 INT 位被置 1, 就保存绘图属性指定的 ID。未使用位的读取值为 0。在复位时, ICIDR 保持不变。

- bit7~0: 中断命令 ID
- bit31~8: 保留位
读取值为 0。

23.3.2 存储器控制寄存器

(1) 返回地址寄存器 0 (RTN0R)

偏移: H'040

初始值: 不定值

返回地址寄存器 0 (RTN0R) 是 32 位只读寄存器, 设定 GOSUB 命令的绘图属性的 No 位为 0 时的返回地址。RTN0R 所示的地址为长字地址 (A28 ~ A2)。在复位时, RTN0R 保持不变 (未使用位的读取值为不定值)。

(2) 返回地址寄存器 1 (RTN1R)

偏移: H'044

初始值: 不定值

返回地址寄存器 1 (RTN1R) 是 32 位只读寄存器, 设定 GOSUB 命令的绘图属性的 No 位为 1 时的返回地址。RTN1R 所示的地址为长字地址 (A28 ~ A2)。在复位时, RTN1R 保持不变 (未使用位的读取值为不定值)。

(3) 显示列表起始地址寄存器 (DLSAR)

偏移: H'048

初始值: 不定值

显示列表起始地址寄存器 (DLSAR) 是 32 位可读写寄存器, 设定用作显示列表的存储区。以 16 字节为单位, 设定显示列表的起始物理地址 (A28 ~ A0)。在 32 位地址模式中, 必须将指定的 32 位地址的低 29 位地址设定到 bit28 ~ bit0, 给低 4 位写 0。未使用的位只能写 0 (读取值为不定值)。在复位时, DLSAR 保持不变。不能映像到平铺区 (参照“第 11 章 存储器控制单元 (MCU)”)。

(4) 2 维源区域起始地址寄存器 (SSAR)

偏移: H'04C

初始值: 不定值

2 维源区域起始地址寄存器 (SSAR) 是 32 位可读写寄存器, 设定用作 2 维源区域的存储区。此寄存器设定的物理地址为 2 维源坐标的原点物理地址。以 16 字节为单位, 设定 2 维源区域的起始物理地址 (A28 ~ A0)。在 32 位地址模式中, 必须将指定的 32 位地址的低 29 位地址设定到 bit28 ~ bit0, 给低 4 位写 0。未使用位只能写 0 (读取值为不定值)。在复位时, SSAR 保持不变。在映像到平铺寻址区 (参照“第 11 章 存储器控制单元 (MCU)”) 时, 必须给低 9 位置 0 (512 字节单位)。

(5) 绘图起始地址寄存器 (RSAR)

偏移: H'050

初始值: 不定值

绘图起始地址寄存器 (RSAR) 是 32 位可读写寄存器, 设定用作绘图区的存储区。此寄存器设定的物理地址为绘图坐标的原点物理地址。以 16 字节为单位, 设定绘图区的起始物理地址 (A28 ~ A0)。在 32 位地址模式中, 必须将指定的 32 位地址的低 29 位地址设定到 bit28 ~ bit0, 给低 4 位写 0。未使用的位只能写 0 (读取值为不定值)。在复位时, RSAR 保持不变。在映像到平铺寻址区 (参照“第 11 章 存储器控制单元 (MCU)”) 时, 必须将低 9 位置 0 (512 字节单位)。

不能与工作区重叠。

(6) 工作区起始地址寄存器 (WSAR)

偏移: H'054

初始值: 不定值

工作区起始地址寄存器 (WSAR) 是 32 位可读写寄存器, 设定用作工作区的存储区。此寄存器设定的物理地址为工作坐标的原点物理地址。以 16 字节为单位, 设定工作区的起始物理地址 (A28 ~ A0)。在 32 位地址模式中, 必须将指定的 32 位地址的低 29 位地址设定到 bit28 ~ bit0, 给低 4 位写 0。未使用的位只能写 0 (读取值为不定值)。

在复位时, WSAR 保持不变。不能映像到平铺寻址区 (参照“第 11 章 存储器控制单元 (MCU)”)。

不能用工作绘图命令以外的命令对工作区进行描绘。CPU 不能在绘图期间 (从绘图开始后到执行 TRAP 命令 (包含 NOP/INT 命令的停止描绘期间)) 写工作区。不能将通过工作绘图命令描绘的图形用作源数据。

(7) 源跨距寄存器 (SSTR)

偏移: H'058

初始值: 不定值

源跨距寄存器 (SSTR) 是 32 位可读写寄存器, 设定 2 维源区域的跨距。在复位时, SSTR 保持不变。

- bit12 ~ 0: 源跨距 (SSTRIDE)

用像素数指定 2 维源区域的跨距, 必须设定为 $16 \leq \text{SSTRIDE} \leq 4096$ 。必须给低 4 位写 0 (16 像素单位)。

在将使用的 2 维源区域映像到平铺寻址区 (参照“第 11 章 存储器控制单元 (MCU)”) 时, 只能设定 512、1024、2048 或者 4096。

- bit31 ~ 13: 保留位

只能写 0, 读取值为 0。

(8) 目标跨距寄存器 (DSTR)

偏移: H'05C

初始值: 不定值

目标跨距寄存器 (DSTR) 是 32 位可读写寄存器, 设定目标区域的跨距。在复位时, DSTR 保持不变。

- bit12 ~ 0: 目标跨距 (DSTRIDE)

用像素数指定目标区域的跨距, 必须设定为 $256 \leq \text{DSTRIDE} \leq 4096$ 。必须给低 4 位写 0 (16 像素单位)。

在将使用的目标区域映像到平铺寻址区 (参照“第 11 章 存储器控制单元 (MCU)”) 时, 只能设定 512、1024、2048 或者 4096。

- bit31 ~ 13: 保留位

只能写 0, 读取值为 0。

(9) 字节排序转换控制寄存器 (ENDCVR)

偏移: H'060

初始值: H'00000000

字节排序转换控制寄存器 (ENDCVR) 是 32 位可读写寄存器, 设定字节排序的转换方法。

- bit31 ~ 4: 保留位

只能写 0, 读取值为 0。

- bit3: 长字交换 (LWSWAP)
以长字 (32位) 为单位进行数据交换。

bit3	说明
LWSWAP	
0	不交换数据。(初始值)
1	以长字 (32 位) 为单位进行数据交换。

- bit2: 字交换 (WSWAP)
以字 (16位) 为单位进行数据交换。

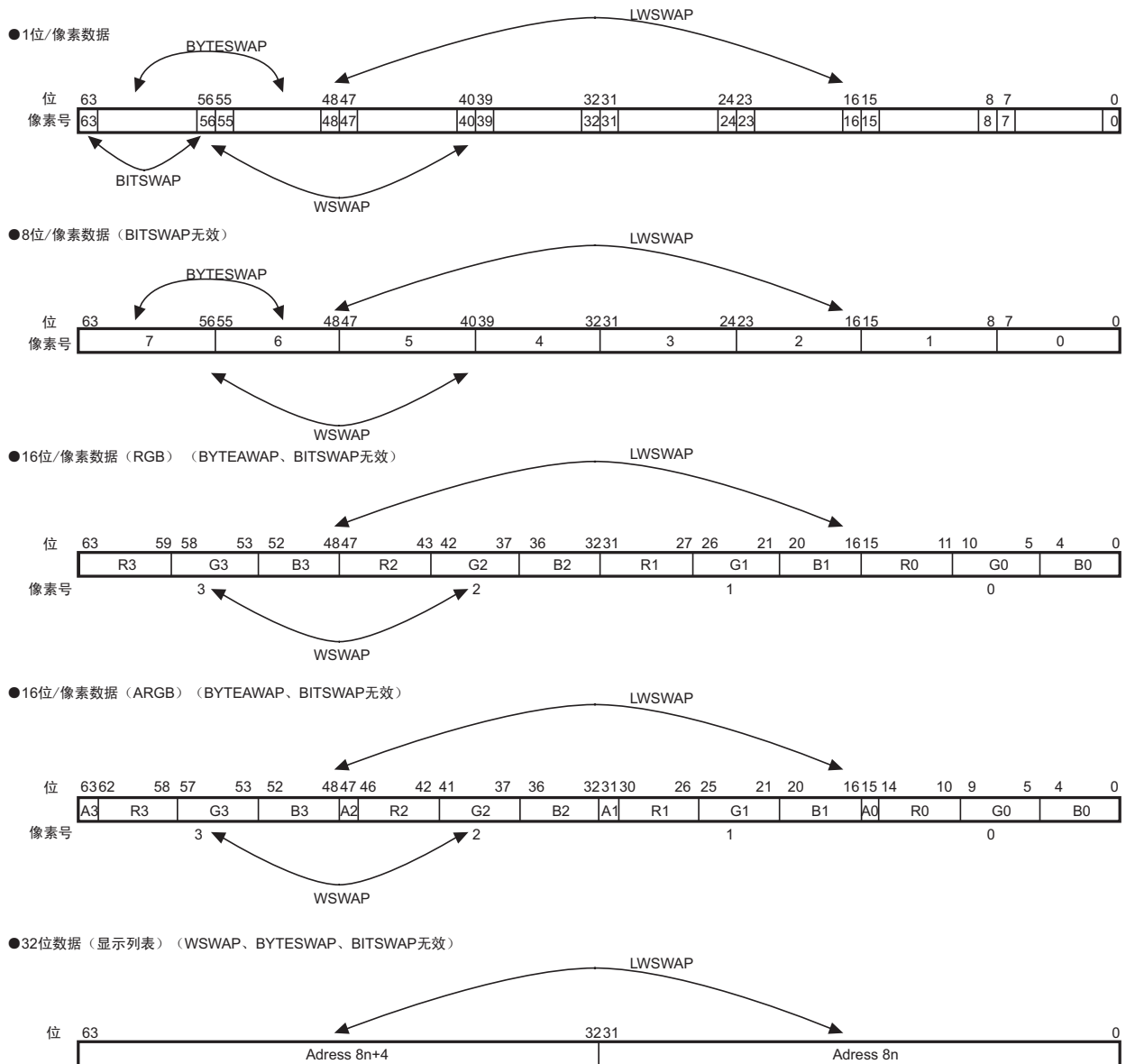
bit2	说明
WSWAP	
0	不交换数据。(初始值)
1	以字 (16 位) 为单位进行数据交换。

- bit1: 字节交换 (BYTESWAP)
以字节 (8位) 为单位进行数据交换。

bit1	说明
BYTESWAP	
0	不交换数据。(初始值)
1	以字 (8 位) 为单位进行数据交换。

- bit0: 位交换 (BITSWAP)
以位为单位进行数据交换。

bit0	说明
BITSWAP	
0	不交换数据。(初始值)
1	以位为单位进行数据交换。



23.3.3 颜色控制寄存器

(1) 源透明色寄存器 (STCR)

偏移: H'080

初始值: 不定值

源透明色寄存器 (STCR) 是 32 位可读写寄存器, 在将绘图属性的 STRANS 位置 1 时, 将源数据和此寄存器设定的颜色进行比较。如果绘图控制寄存器 (RCLR) 的源透明色极性位 (STP) 为 0, 就在此寄存器设定的颜色和源数据相同时透明; 如果 STP 为 1, 就在不不同时透明。无论 STP 是 1 还是 0 都不进行描绘。在复位时, STCR 保持不变。

位	名称	含义
24	STC1	源为 1 位 / 像素时的透明色
23 ~ 16	STC8	源为 8 位 / 像素时的透明色
15 ~ 0	STC16	源为 16 位 / 像素时的透明色

在源为 16 位 / 像素时, 必须使其符合绘图控制寄存器 (RCLR) 的 SPF 的格式。SPF=1 (ARGB=1555) 时, 不比较 A 值。

- bit31 ~ 25: 保留位
只能写 0, 读取值为 0。

(2) 目标透明色寄存器 (DTCR)

偏移: H'084

初始值: 不定值

目标透明色寄存器 (DTCR) 是 32 位可读写寄存器, 在将绘图属性的 DTRANS 位置 1 时, 将目标数据和此寄存器设定的颜色进行比较。如果绘图控制寄存器 (RCLR) 的目标透明色极性位 (DTP) 为 0, 就在此寄存器设定的颜色和目标数据相同时透明; 如果 DTP 为 1, 就在不不同时透明。无论 DTP 是 1 还是 0 都不进行描绘。在复位时, DTCR 保持不变。

位	名称	含义
23 ~ 16	DTC8	目标为 8 位 / 像素时的透明色
15 ~ 0	DTC16	目标为 16 位 / 像素时的透明色

在目标为 16 位 / 像素时, 必须使其符合绘图控制寄存器 (RCLR) 的 DPF 的格式。在 DPF=1 (ARGB=1555) 时, 不比较 A 值。

- bit31 ~ 24: 保留位
只能写 0, 读取值为 0。

(3) α 值寄存器 (ALPHR)

偏移: H'088

初始值: 不定值

α 值寄存器 (ALPHR) 是 32 位可读写寄存器, 设定绘图属性的 α E 位为 1 时的 α 混合值。在复位时, ALPHR 保持不变。对于蓝色和红色成分的混合, α 值的高 5 位有效。对于绿色成分, 在目标像素格式为 RGB 模式时高 6 位有效, 为 ARGB 格式时高 5 位有效。

位	名称	含义
7 ~ 0	ALPH	设定 α 值。

目标 \approx 源 *ALPH/255+ 目标 (1-ALPH/255) (这是 ALPH 为 8 位时的近似表达式)

- bit31 ~ 8: 保留位
只能写 0, 读取值为 0。

(4) 颜色偏移寄存器 (COFSR)

偏移: H'08C

初始值: 不定值

颜色偏移寄存器 (COFSR) 是 32 位可读写寄存器, 在进行 16 位 / 像素描绘时, 如果将绘图属性的 COOF 位置 1, 就给源数据 (2 值源为颜色展开后的数据, 单色指定时为指定的颜色) 加上 COFSR 寄存器的值, 然后描画所得的结果, 通过饱和处理进行运算。在 8 位 / 像素时, 必须将绘图属性的 COOF 位置 0。偏移的各成分作为带符号的整数进行处理。负数时, 必须设定为 2 的补码。在复位时, COFSR 保持不变。

源像素格式为 RGB=565 时 (SPF=0)

位	名称	含义
23 ~ 19	COR (Color Offset R)	颜色偏移的红色成分
15 ~ 10	COG (Color Offset G)	颜色偏移的绿色成分
7 ~ 3	COB (Color Offset B)	颜色偏移的蓝色成分

舍去 bit18 ~ 16、9、8、2 ~ 0。读取值为 0。

源像素格式为 ARGB=1555 时 (SPF=1)

位	名称	含义
23 ~ 19	COR (Color Offset R)	颜色偏移的红色成分
15 ~ 11	COG (Color Offset G)	颜色偏移的绿色成分
7 ~ 3	COB (Color Offset B)	颜色偏移的蓝色成分

舍去 bit18 ~ 16、10 ~ 8、2 ~ 0。读取值为 0。

- bit31 ~ 24: 保留位
只能写 0, 读取值为 0。

23.3.4 绘图控制寄存器

(1) 绘图控制寄存器 (RCLR)

偏移: H'0C0

初始值: H'0000000

绘图控制寄存器 (RCLR) 是 32 位可读写寄存器, 设定绘图属性。

- bit25: 源透明色极性 (STP)
选择是在源数据和源透明色寄存器的设定值相同时透明还是在不相同透明。

bit25	说明
STP	
0	相同时透明。(初始值)
1	不相同透明。

- bit24: 目标透明色极性 (DTP)
选择是在目标数据和目标透明色寄存器的设定值相同时透明还是在不相同透明。

bit24	说明
DTP	
0	相同时透明。(初始值)
1	不相同透明。

- bit21: 源像素格式 (SPF)
指定多值源的像素格式。此位只在多值源为 16 位/像素时有效。在 8 位/像素时, 必须置 0。必须使其符合目标像素格式。

bit21	说明
SPF	
0	为 RGB=565 格式。(初始值)
1	为 ARGB=1555 格式。

- bit20: 目标像素格式 (DPF)
指定目标的像素格式。此位只在目标为 16 位/像素时有效。在 8 位/像素时, 必须置 0。必须使其符合多值源的像素格式。

bit20	说明
DPF	
0	为 RGB=565 格式。(初始值)
1	为 ARGB=1555 格式。

- bit18: 图形位模式 (GBM)
指定多值源和目标的图形位模式。

bit18	说明
GBM	
0	为 8 位 / 像素。(初始值)
1	为 16 位 / 像素。

- bit17: 源 A 值的使用 (SAU)
在源和目标的像素格式为 ARGB 时, 参照源 A 值进行目标 A 值的描绘。
- bit16: A 值 (AVALUE)
在源和目标的像素格式为 ARGB 时, 用为 0 或者 1 描绘目标 A 值。

源 / 目标的像素格式 (SPF/DPF) 和 SAU 位、AVALUE 的关系

SPF	DPF	SAU	AVALUE	说明
0	0	*	*	源 =RGB (565)、目标 =RGB (565)。SAU 和 AVALUE 无效。
0	1	*	*	禁止设定。
1	0	*	*	禁止设定。
1	1	0	0	源 =ARGB (1555)、目标 =ARGB (1555)。用 0 描绘目标 A 值。
		0	1	源 =ARGB (1555)、目标 =ARGB (1555)。用 1 描绘目标 A 值。
		1	*	源 =ARGB (1555)、目标 =ARGB (1555)。参照源 A 值进行目标 A 值的描绘。AVALUE 无效。

【注】 * Don't care

当 SAU=1 时, 在参照 2 值源的命令的情况下, 参照命令参数 Color0 和 Color1 的 A 值; 在单色指定命令的情况下, 参照命令参数 Color 的 A 值。

在 LINED 命令时, 与 SAU 位和 AVALUE 位的设定无关, 回写底层 (目标) 的 A 值。

- bit1: 线预裁剪允许 (LPCE)

用于 (R)LINE 类和 (R)LINEW 类命令。如果将 LPCE 置 1, 就在 2 维裁剪区 (系统裁剪区、用户裁剪区和相对用户裁剪区) 以线段为单位进行预裁剪。但是, 如果中间的线段被预裁剪, 图案就不连续 (图案从以前描绘的线段的终点开始)。

bit1	说明
LPCE	
0	不进行预裁剪。(初始值)
1	在 2 维裁剪区以线段为单位进行预裁剪。

- bit0: 连接描绘屏蔽 (COM)
选择是否描绘粗线的连接部分。

bit0	说明
COM	
0	描绘粗线的连接部分。(初始值)
1	不描绘粗线的连接部分。

- bit31 ~ 26、23、22、19、15 ~ 2: 保留位
只能写0, 读取值为0。

(2) 命令状态寄存器 (CSTR)

偏移: H'0C4

初始值: 不定值

命令状态寄存器 (CSTR) 是 32 位只读寄存器, 保存取到的命令字 (操作码字) 的地址。
CSTR 所示的地址为长字地址 (A28 ~ 2)。未使用位的读取值为 0。在复位时, CSTR 保持不变。

(3) 当前指针寄存器 (CURR)

偏移: H'0C8

初始值: 不定值

当前指针寄存器 (CURR) 是 32 位只读寄存器, 表示指针的当前坐标。
高位字表示指针的 X 坐标 (XC), 低位字表示指针的 Y 坐标 (YC)。
在复位时, CURR 保持不变。

(4) 局部偏移寄存器 (LCOR)

偏移: H'0CC

初始值: 不定值

局部偏移寄存器 (LCOR) 是 32 位只读寄存器, 表示偏移坐标。
高位字表示偏移的 X 坐标 (XO), 低位字表示偏移的 Y 坐标 (YO)。
在复位时, LCOR 保持不变。

(5) 系统裁剪区 MAX 寄存器 (SCLMAR)

偏移: H'0D0

初始值: 不定值

系统裁剪区 MAX 寄存器 (SCLMAR) 是 32 位只读寄存器, 表示系统裁剪坐标的 MAX 值。高位字表示系统裁剪坐标的 XMAX (SXMAX), 低位字表示系统裁剪坐标的 YMAX (SYMAX)。此寄存器的未使用位的读取值为 0。在通过 WPR 命令进行设定时, 必须设定绘图范围的最大值 (最大 4095。但是, 必须设定 SXMAX < DSTRR)。

在复位时, SCLMAR 保持不变。

(6) 用户裁剪区 MIN 寄存器 (UCLMIR)

偏移: H'0D4

初始值: 不定值

用户裁剪区 MIN 寄存器 (UCLMIR) 是 32 位只读寄存器, 表示用户裁剪坐标的 MIN 值。高位字表示用户裁剪坐标的 XMIN (UXMIN), 低位字表示用户裁剪坐标的 YMIN (UYMIN)。此寄存器的未使用位的读取值为 0。在通过 WPR 命令进行设定时, 必须设定为 $0 \leq UXMIN \leq UXMAX \leq SXMAX \leq 4095$ 、 $0 \leq UYMIN \leq UYMAX \leq SYMAX \leq 4095$ 。

在复位时, UCLMIR 保持不变。

(7) 用户裁剪区 MAX 寄存器 (UCLMAR)

偏移: H'0D8

初始值: 不定值

用户裁剪区 MAX 寄存器 (UCLMAR) 是 32 位只读寄存器, 表示用户裁剪坐标的 MAX 值。高位字表示用户裁剪坐标的 X MAX (UX MAX), 低位字表示用户裁剪坐标的 Y MAX (UY MAX)。此寄存器的未使用位的读取值为 0。在通过 WPR 命令进行设定时, 必须设定为 $0 \leq UXMIN \leq UXMAX \leq SXMAX \leq 4095$ 、 $0 \leq UYMIN \leq UYMAX \leq SYMAX \leq 4095$ 。

在复位时, UCLMAR 保持不变。

(8) 相对用户裁剪区 MIN 寄存器 (RUCLMIR)

偏移: H'0DC

初始值: 不定值

相对用户裁剪区 MIN 寄存器 (RUCLMIR) 是 32 位只读寄存器, 表示相对用户裁剪坐标的 MIN 值 (给局部偏移加上的偏移值)。在通过 WPR 命令进行设定时, 必须设定局部偏移的相对坐标。高位字表示相对用户裁剪坐标的 XMIN (RUXMIN), 低位字表示相对用户裁剪坐标的 YMIN (RUYMIN)。此寄存器的未使用位的读取值为 0。在通过 WPR 命令进行设定时, 必须设定为 $0 \leq RUXMIN \leq RUXMAX \leq SXMAX \leq 4095$ 、 $0 \leq RUYMIN \leq RUYMAX \leq SYMAX \leq 4095$ 。有关设定范围的详细内容, 请参照“23.1.5(5) 相对裁剪的指定 (RCLIP)”。在复位时, RUCLMIR 保持不变。

(9) 相对用户裁剪区 MAX 寄存器 (RUCLMAR)

偏移: H'0E0

初始值: 不定值

相对用户裁剪区 MAX 寄存器 (RUCLMAR) 是 32 位只读寄存器, 表示相对用户裁剪坐标的 MAX 值 (给局部偏移加上的偏移值)。在通过 WPR 命令设进行定时, 必须设定局部偏移的相对坐标。高位字表示相对用户裁剪坐标的 X MAX (RUX MAX), 低位字表示相对用户裁剪坐标的 Y MAX (RUY MAX)。此寄存器的未使用位的读取值为 0。在通过 WPR 命令进行设定时, 必须设定为 $0 \leq RUXMIN \leq RUXMAX \leq SXMAX \leq 4095$ 、 $0 \leq RUYMIN \leq RUYMAX \leq SYMAX \leq 4095$ 。有关设定范围的详细内容, 请参照“23.1.5(5) 相对裁剪的指定 (RCLIP)”。在复位时, RUCLMAR 保持不变。

(10) 绘图控制 2 寄存器 (RCL2R)

偏移: H'0F0

初始值: H'00004004

绘图控制 2 寄存器 (RCL2R) 是 32 位可读写寄存器, 设定绘图属性。

- bit21: 目标 α 允许 (DAE)
和 α 混合允许位 (αE) 一起使用。在 ARGB=1555 格式时, 只对目标 (底层) 的 A 值为 1 的像素进行 α 混合。不描绘目标 (底层) 的 A 值为 0 的像素。

bit21	说明
DAE	
0	与目标 (底层) 的 A 值无关, 进行 α 混合。(初始值)
1	只对目标 (底层) 的 A 值为 1 的像素进行 α 混合。

- 【注】**
- 在 RGB=565 格式或者 8 位 / 像素时, 必须置 0。
 - 在非 POLYGON4 类命令时, 必须置 0。
 - 在 α 混合允许 (αE) =0 时, 必须置 0。
 - 因为此位不通过命令进行解码, 所以必须对每个对应的命令进行设定和解除。

- bit20: 图案类型允许 (PSTYLE)
和源类型指定位 (STYLE) 一起使用。以目标坐标为基础, 以图案大小重复生成源图案。

bit20	说明
PSTYLE	
0	图案类型无效。(初始值)
1	以目标坐标为基础, 生成源图案。

- 【注】**
- 必须将源偏移 TXOFS 和 TYOFS 置 0。
 - 在源类型指定 (STYLE) =0 时, 必须置 0。
 - 必须设定源地址指定 (SS) =0。
 - 在非 POLYGON4A/B 命令时, 必须置 0。
 - 因为此位不通过命令进行解码, 所以必须对每个对应的命令进行设定和解除。

- bit19、18: 图案 X 大小 (PXSIZE)
在图案类型允许 (PSTYLE) =1 时, 指定图案 X 的大小。

bit19	bit18	说明
PXSIZE[1]	PXSIZE[0]	
0	0	图案 X 大小 =8 像素。
0	1	图案 X 大小 =16 像素。
1	0	图案 X 大小 =32 像素。
1	1	图案 X 大小 =64 像素。

- 【注】** 必须给源大小 TDX 设定指定的图案 X 大小 (8、16、32、64)。

- bit17、16: 图案 Y 大小 (PYSIZE)
在图案类型允许 (PSTYLE) =1 时, 指定图案 Y 的大小。

bit17	bit16	说明
PYSIZE[1]	PYSIZE[0]	
0	0	图案 Y 大小 =8 像素。
0	1	图案 Y 大小 =16 像素。
1	0	图案 Y 大小 =32 像素。
1	1	图案 Y 大小 =64 像素。

【注】 必须给源大小 TDY 设定指定的图案 Y 大小 (8、16、32、64)。

- bit31 ~ 22、15 ~ 0: 保留位
只能写初始值。

(11) 图案偏移寄存器 (POFSR)

偏移: H'0F8

初始值: H'00000000

图案偏移寄存器 (POFSR) 是 32 位可读写寄存器, 设定图案类型允许 (PSTYLE) =1 时的偏移值。只在图形类型允许 (PSTYLE) =1 时, 参照此设定值。

- bit31 ~ 16: 图案偏移 X (POFSX)
用 16 位整数进行 X 方向的图案偏移值设定。负数时, 必须设定 2 的补码。
- bit15 ~ 0: 图案偏移 Y (POFSY)
用 16 位整数进行 Y 方向的图案偏移值设定。负数时, 必须设定 2 的补码。

23.3.5 坐标变换控制寄存器

(1) 坐标变换控制寄存器 (GTRCR)

偏移: H'100

初始值: H'00000000

坐标变换控制寄存器 (GTRCR) 是 32 位可读写寄存器, 设定是否进行坐标变换的允许位。

- bit31: 坐标变换允许 (GTE)
进行坐标变换。

bit31	说明
GET	
0	不进行坐标变换。绘图属性的 MTRE 无效。(初始值)
1	绘图属性的 MTRE 有效。

- bit0: 仿射变换允许 (AFE)
在坐标变换时, 不进行 W 除法运算和偏移的加法运算。在绘图属性的 MTRE 位=1 并且 GTE 位=1 时有效。

bit0	说明
AFE	
0	矩阵运算后的 TX, TY 除以 WC, 然后加上偏移值, 结果为顶点坐标 X', Y'。 $X'=TX/WC+GTROFSX$ $Y'=TY/WC+GTROFSY$ 给坐标变换偏移 X 寄存器 (GTROFSXR) 和坐标变换偏移 Y 寄存器 (GTROFSYR) 分别设定 GTROFSX 和 GTROFSY。(初始值)
1	矩阵运算后的 TX, TY 为顶点坐标 X', Y'。 $X'=TX$ $Y'=TY$

- bit30~1: 保留位
只能 0, 读取值为 0。

(2) 矩阵参数 A 寄存器 (MTRAR)

偏移: H'104
初始值: 不定值

矩阵参数 A 寄存器 (MTRAR) 是 32 位可读写寄存器, 以 IEEE754 单精度浮点格式, 设定坐标变换时的矩阵参数。但是, 因为以 32 位定点数 (整数部 16 位、尾数部 16 位) 进行内部运算, 所以必须设定为 $-2^{15} \leq MTRAR < 2^{15}$ 。

在复位时, MTRAR 保持不变。

【注】 有关设定范围的详细内容, 请参照 “23.1.2 基本功能 (2) 4×4 矩阵运算~ (5) 坐标变换的流程和饱和处理”。

(3) 矩阵参数 B 寄存器 (MTRBR)

偏移: H'108
初始值: 不定值

矩阵参数 B 寄存器 (MTRBR) 是 32 位可读写寄存器, 以 IEEE754 单精度浮点格式, 设定坐标变换时的矩阵参数。但是, 因为以 32 位定点数 (整数部 16 位、尾数部 16 位) 进行内部运算, 所以必须设定为 $-2^{15} \leq MTRBR < 2^{15}$ 。

在复位时, MTRBR 保持不变。

【注】 有关设定范围的详细内容, 请参照 “23.1.2 基本功能 (2) 4×4 矩阵运算~ (5) 坐标变换的流程和饱和处理”。

(4) 矩阵参数 C 寄存器 (MTRCR)

偏移: H'10C
初始值: 不定值

矩阵参数 C 寄存器 (MTRCR) 是 32 位可读写寄存器, 以 IEEE754 单精度浮点格式, 设定坐标变换时的矩阵参数。但是, 因为以 32 位定点数 (整数部 16 位、尾数部 16 位) 进行内部运算, 所以必须设定为 $-2^{15} \leq MTRCR < 2^{15}$ 。

在复位时, MTRCR 保持不变。

【注】 有关设定范围的详细内容, 请参照 “23.1.2 基本功能 (2) 4×4 矩阵运算~ (5) 坐标变换的流程和饱和处理”。

(5) 矩阵参数 D 寄存器 (MTRDR)

偏移: H'110

初始值: 不定值

矩阵参数 D 寄存器 (MTRDR) 是 32 位可读写寄存器, 以 IEEE754 单精度浮点格式, 设定坐标变换时的矩阵参数。但是, 因为以 32 位定点数 (整数部 16 位、尾数部 16 位) 进行内部运算, 所以必须设定为 $-2^{15} \leq \text{MTRDR} < 2^{15}$ 。

在复位时, MTRDR 保持不变。

【注】 有关设定范围的详细内容, 请参照“23.1.2 基本功能 (2) 4×4 矩阵运算~ (5) 坐标变换的流程和饱和处理”。

(6) 矩阵参数 E 寄存器 (MTRER)

偏移: H'114

初始值: 不定值

矩阵参数 E 寄存器 (MTRER) 是 32 位可读写寄存器, 以 IEEE754 单精度浮点格式, 设定坐标变换时的矩阵参数。但是, 因为以 32 位定点数 (整数部 16 位、尾数部 16 位) 进行内部运算, 所以必须设定为 $-2^{15} \leq \text{MTRER} < 2^{15}$ 。

在复位时, MTRER 保持不变。

【注】 有关设定范围的详细内容, 请参照“23.1.2 基本功能 (2) 4×4 矩阵运算~ (5) 坐标变换的流程和饱和处理”。

(7) 矩阵参数 F 寄存器 (MTRFR)

偏移: H'118

初始值: 不定值

矩阵参数 F 寄存器 (MTRFR) 是 32 位可读写寄存器, 以 IEEE754 单精度浮点格式, 设定坐标变换时的矩阵参数。但是, 因为以 32 位定点数 (整数部 16 位、尾数部 16 位) 进行内部运算, 所以必须设定为 $-2^{15} \leq \text{MTRFR} < 2^{15}$ 。

在复位时, MTRFR 保持不变。

【注】 有关设定范围的详细内容, 请参照“23.1.2 基本功能 (2) 4×4 矩阵运算~ (5) 坐标变换的流程和饱和处理”。

(8) 矩阵参数 G 寄存器 (MTRGR)

偏移: H'11C

初始值: 不定值

矩阵参数 G 寄存器 (MTRGR) 是 32 位可读写寄存器, 以 IEEE754 单精度浮点格式, 设定坐标变换时的矩阵参数。但是, 因为以 32 位定点数 (整数部 16 位、尾数部 16 位) 进行内部运算, 所以必须设定为 $-2^{15} \leq \text{MTRGR} < 2^{15}$ 。

在复位时, MTRGR 保持不变。

【注】 有关设定范围的详细内容, 请参照“23.1.2 基本功能 (2) 4×4 矩阵运算~ (5) 坐标变换的流程和饱和处理”。

(9) 矩阵参数 H 寄存器 (MTRHR)

偏移: H'120

初始值: 不定值

矩阵参数 H 寄存器 (MTRHR) 是 32 位可读写寄存器, 以 IEEE754 单精度浮点格式, 设定坐标变换时的矩阵参数。但是, 因为以 32 位定点数 (整数部 16 位、尾数部 16 位) 进行内部运算, 所以必须设定为 $-2^{15} \leq \text{MTRHR} < 2^{15}$ 。

在复位时, MTRHR 保持不变。

【注】 有关设定范围的详细内容, 请参照 “23.1.2 基本功能 (2) 4x4 矩阵运算 ~ (5) 坐标变换的流程和饱和处理”。

(10) 矩阵参数 I 寄存器 (MTRIR)

偏移: H'124

初始值: 不定值

矩阵参数 I 寄存器 (MTRIR) 是 32 位可读写寄存器, 以 IEEE754 单精度浮点格式, 设定坐标变换时的矩阵参数。但是, 因为以 32 位定点数 (整数部 16 位、尾数部 16 位) 进行内部运算, 所以必须设定为 $-2^{15} \leq \text{MTRIR} < 2^{15}$ 。

在复位时, MTRIR 保持不变。

【注】 有关设定范围相关的详细内容, 请参照 “23.1.2 基本功能 (2) 4x4 矩阵运算 ~ (5) 坐标变换的流程和饱和处理”。

(11) 坐标变换偏移 X 寄存器 (GTROFSXR)

偏移: H'128

初始值: 不定值

坐标变换偏移 X 寄存器 (GTROFSXR) 是 32 位可读写寄存器, 用 16 位整数设定坐标变换时的 X 偏移值。负数时, 用 2 的补码进行设定。

在复位时, GTROFSXR 保持不变。

【注】 有关设定范围的详细内容, 请参照 “23.1.2 基本功能 (2) 4x4 矩阵运算 ~ (5) 坐标变换的流程和饱和处理”。

(12) 坐标变换偏移 Y 寄存器 (GTROFSYR)

偏移: H'12C

初始值: 不定值

坐标变换偏移 Y 寄存器 (GTROFSYR) 是 32 位可读写寄存器, 用 16 位整数设定坐标变换时的 Y 偏移值。负数时, 用 2 的补码进行设定。

在复位时, GTROFSYR 保持不变。

【注】 有关设定范围的详细内容, 请参照 “23.1.2 基本功能 (2) 4x4 矩阵运算 ~ (5) 坐标变换的流程和饱和处理”。

(13) Z 裁剪区 MIN 寄存器 (ZCLPMINR)

偏移: H'130

初始值: 不定值

Z 裁剪区 MIN (ZCLPMINR) 是 32 位可读写寄存器, 以 IEEE754 单精度浮点格式, 设定 Z 裁剪区的 MIN 值。因为用 W 值进行比较, 所以必须将 ZCLPMINR 设定为 W 的相应值。

以 32 位定点数 (整数部 16 位、尾数部 16 位) 进行内部运算, 所以必须设定为 $2^{-16} \leq ZCLPMIN \leq ZCLPMAX < 2^{15}$ 。

在复位时, ZCLPMINR 保持不变。

【注】 有关设定范围的详细内容, 请参照 “23.1.2 基本功能 (2) 4x4 矩阵运算 ~ (5) 坐标变换的流程和饱和处理”。

(14) Z 裁剪区 MAX 寄存器 (ZCLPMAXR)

偏移: H'134

初始值: 不定值

Z 裁剪区 MAX (ZCLPMAXR) 是 32 位可读写寄存器, 以 IEEE754 单精度浮点格式, 设定 Z 裁剪区的 MAX 值。因为用 W 值进行比较, 所以必须将 ZCLPMAXR 设定为 W 的相应值。

以 32 位定点数 (整数部 16 位、尾数部 16 位) 进行内部运算, 所以必须设定为 $2^{-16} \leq ZCLPMIN \leq ZCLPMAX < 2^{15}$ 。

在复位时, ZCLPMAXR 保持不变。

【注】 有关设定范围的详细内容, 请参照 “23.1.2 基本功能 (2) 4x4 矩阵运算 ~ (5) 坐标变换的流程和饱和处理”。

(15) Z 饱和值 MIN 寄存器 (ZSATVMINR)

偏移: H'138

初始值: 不定值

Z 饱和值 MIN (ZSATVMINR) 是 32 位可读写寄存器, 以 IEEE754 单精度浮点格式, 设定 Z 饱和值的 MIN。因为用 W 值进行比较, 所以必须将 ZSATVMINR 设定为 W 的相应值。因为, 以 32 位定点数 (整数部 16 位、尾数部 16 位) 进行内部运算, 所以必须设定为 $2^{-16} \leq ZSATVMIN \leq ZCLPMIN \leq ZCLPMAX < 2^{15}$ 。

在复位时, ZSATVMINR 保持不变。

【注】 有关设定范围的详细内容, 请参照 “23.1.2 基本功能 (2) 4x4 矩阵运算 ~ (5) 坐标变换的流程和饱和处理”。

第 24 章 显示控制器 (VDC2)

24.1 概要

显示控制器 (Video Display Controller 2: VDC2) 的功能是读取保存在外部存储器的 4 个平面的图形图像 (层 1 ~ 层 4) 并且对 4 个平面的图像进行重叠处理的功能。输出 18 位的 RGB (各 6 位) 视频和符合 BTA T-1004 的数字视频。

24.2 特点

项目	功能
工作频率	T-1004 的显示时钟: 54MHz RGB666 的显示时钟: 6.0MHz ~ 36.0MHz 取决于显示屏的大小。
输入图像格式	16 位的 RGB565 逐行方式 (SDRAM)
显示画面尺寸	<ul style="list-style-type: none"> 以逐行方式输出 18 位 RGB 720×480 (NTSC) 720×576 (PAL) 320×240 (QVGA) 640×480 (VGA) 800×480 (WVGA) BTA T-1004 (8;4:4 位格式并行接口) 的 8 位数字输出 (通过 SYNCNT 寄存器的设定, 能选择 RGB 数据的时钟上升沿 / 下降沿的时序) 720×480 (NTSC)
显示画面数	最大 4 个平面 (层 1 ~ 层 4)
α 混合	根据透明度 (α 值) 将层 1 ~ 层 4 进行混合。
色键	对设定的 RGB 色进行色键处理 (能通过 α 值设定透射率)。
输出视频格式	RGB666 逐行方式视频输出 (RGB 各 6 位: 共计 18 位) 输出符合 BTA T-1004 (8;4:4 位格式并行接口) 的 8 位数字视频 (通过 SYNCNT 寄存器的设定, 能选择 RGB 数据的时钟上升沿 / 下降沿的时序)。
同步信号输出	能选择 Vsync、Hsync、数据允许信号、COM/CDE 信号或者 SPL、CLS、SPS、数据允许信号、COM/CDE 信号 (通过 SYNCNT 寄存器的设定, 能选择各信号的时钟上升沿 / 下降沿的时序和极性)。
外部同步模式	能通过外部同步信号 (EX-VSYNC、EX-HSYNC) 和显示屏时钟进行运行 (通过 SYNCNT 寄存器的设定, 能选择外部输入同步信号的时钟上升沿 / 下降沿的时序和极性)。但是, 在使用此模式时, 视频只能输出 RGB666。
色度判断输出	对视频中指定的颜色, 输出判断信号 (CDE)。

24.3 输入 / 输出引脚

表 24.1 输入 / 输出引脚

引脚名	输入 / 输出	名称	说明
DR[5:0]	输出	数字红色数据	是输出视频数据的输出引脚。
DG[5:0]	输出	数字绿色数据	是输出视频数据的输出引脚。
DB[5:0]	输出	数字蓝色数据	是输出视频数据的输出引脚。
VSYNC/SPS	输出	垂直同步信号 / 栅极开始信号	是垂直同步信号 / 栅极开始信号。
HSYNC/SPL	输出	水平同步信号 / 采样开始信号	是水平同步信号 / 采样开始信号。
DE_V/CLS	输出	垂直数据允许信号 / 栅极时钟信号	是垂直数据允许信号 / 栅极时钟信号。
DE_H/DE_C	输出	水平数据允许信号 / 显示允许信号	是水平数据允许信号 / 显示允许信号。
COM/CDE	输出	栅极控制信号 / 色度数据允许信号	是栅极控制信号 / 显示允许信号 (当和寄存器设定的色键对象色相同时, 输出有效电平)。
BT_DATA[7:0]	输出	BTA-T1004 显示数据	是 BTA-T1004 显示数据的输出引脚。
BT_VSYNC	输出	BTA-T1004 垂直同步	是用于 BTA-T1004 的垂直同步信号。
BT_HSYNC	输出	BTA-T1004 水平同步	是用于 BTA-T1004 的水平同步信号。
BT_DE_C	输出	BTA-T1004 显示允许	是 BTA-T1004 显示允许信号。
EX_VSYNC	输入	VSYNC 输入	在外部同步模式时, 输入 VSYNC 信号。
EX_HSYNC	输入	HSYNC 输入	在外部同步模式时, 输入 HSYNC 信号。
DCLKIN	输入	显示屏源极时钟输入	输入用于显示的源极时钟。输入适合显示屏尺寸的频率。
DCLKOUT	输出	显示屏时钟输出	是显示屏时钟的输出引脚。

24.4 VDC2 结构

VDC2 由以下 7 个功能块构成，VDC2 框图如图 24.1 所示。

表 24.2 VDC2 的内部功能块

块名	功能概要
图形部 1 (层 1)	从像素总线读取保存在外部存储器的 Graphics 图像 (RGB565: 层 1)，输出到图形部 2。
图形部 2 (层 2)	从像素总线读取保存在外部存储器的 Graphics 图像 (RGB565: 层 2)，输出到图形部 1，并且进行重叠处理，将处理后的数据输出到图形部 3。
图形部 3 (层 3)	从像素总线读取保存在外部存储器的 Graphics 图像 (RGB565: 层 3)，输出到图形部 2，并且进行重叠处理，将处理后的数据输出到图形部 4。
图形部 4 (层 4)	从像素总线读取保存在外部存储器的 Graphics 图像 (RGB565: 层 4)，输出到图形部 3，并且进行重叠处理，输出处理后的图像数据。
显示控制部	将图形部 4 的输出 (RGB) 进行 YCbCr (4:2:2) 转换，以 BTA T-1004 规格的 8:4:4 并列格式进行输出。输出适合 TFT-LCD 显示屏的控制信号。
输入时序控制部	选择外部输入同步信号的时钟上升沿 / 下降沿的时序和极性。
输出时序控制部	选择同步信号的时钟上升沿 / 下降沿的时序和极性。控制 RGB666 视频输出的时钟上升沿 / 下降沿的时序。

【注】 层 1 ~ 层 4 为相同结构，在最下层 (层 1) 没有 α 混合对象的图像输入。

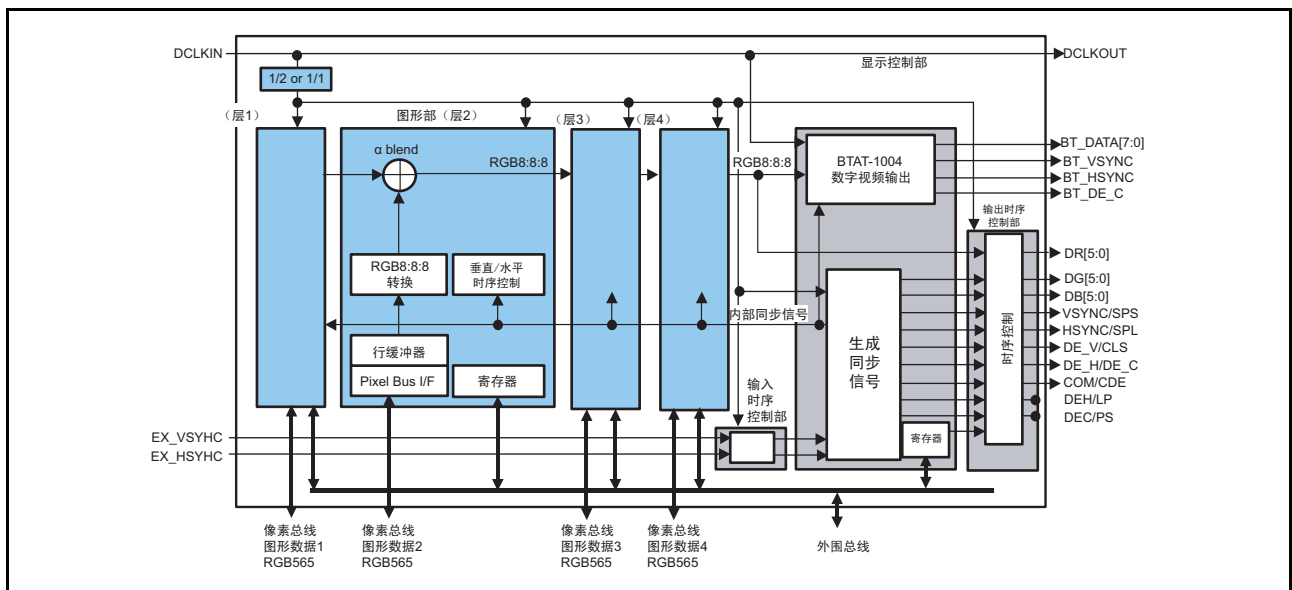


图 24.1 VDC2 框图

24.5 功能说明

24.5.1 图形 (层 1 ~ 层 4)

图形部以 RGB565 (16 位) 格式显示保存在存储区的图像数据, 使用外部输入同步信号或者内部生成的同步信号进行显示控制。图像显示能设定是单面显示还是 2 面、3 面、4 面的重叠显示。在进行重叠面显示时, 通过设定当前图像 (当前层) 的 α 控制区域, 能对下层图形 (下层) 进行透明处理, 透明度能以 $1/256 \times 100\%$ 为单位进行设定。另外, 通过色键处理, 设定指定色的透明度, 能使下层图形 (下层) 的图像透明。例子如图 24.2 所示。

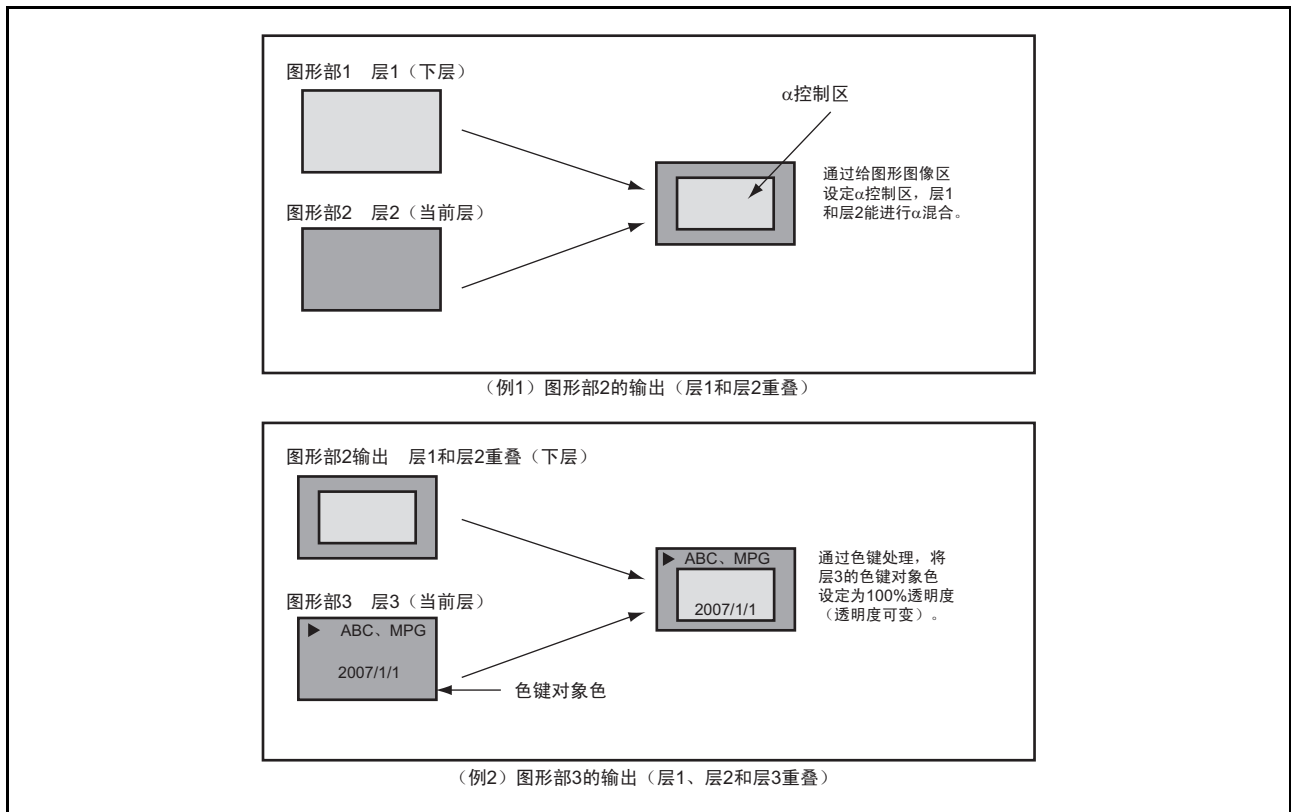


图 24.2 重叠显示的例子

24.5.2 同步信号的生成

能生成的同步信号的格式例子如图 24.3 ~ 图 24.4 所示。能生成和输出 Vsync、Hsync、DEV 和 DEH/DEC。

VSYNC	: 垂直同步信号	SPS	: 栅极开始信号
HSYNC	: 水平同步信号	SPL	: 采样开始信号
DEV	: 垂直数据允许信号	CLS	: 栅极时钟信号
DEH	: 水平数据允许信号	DEC	: 数据允许信号 (合成)
CDE	: 色度数据允许信号	COM	: 栅极控制信号

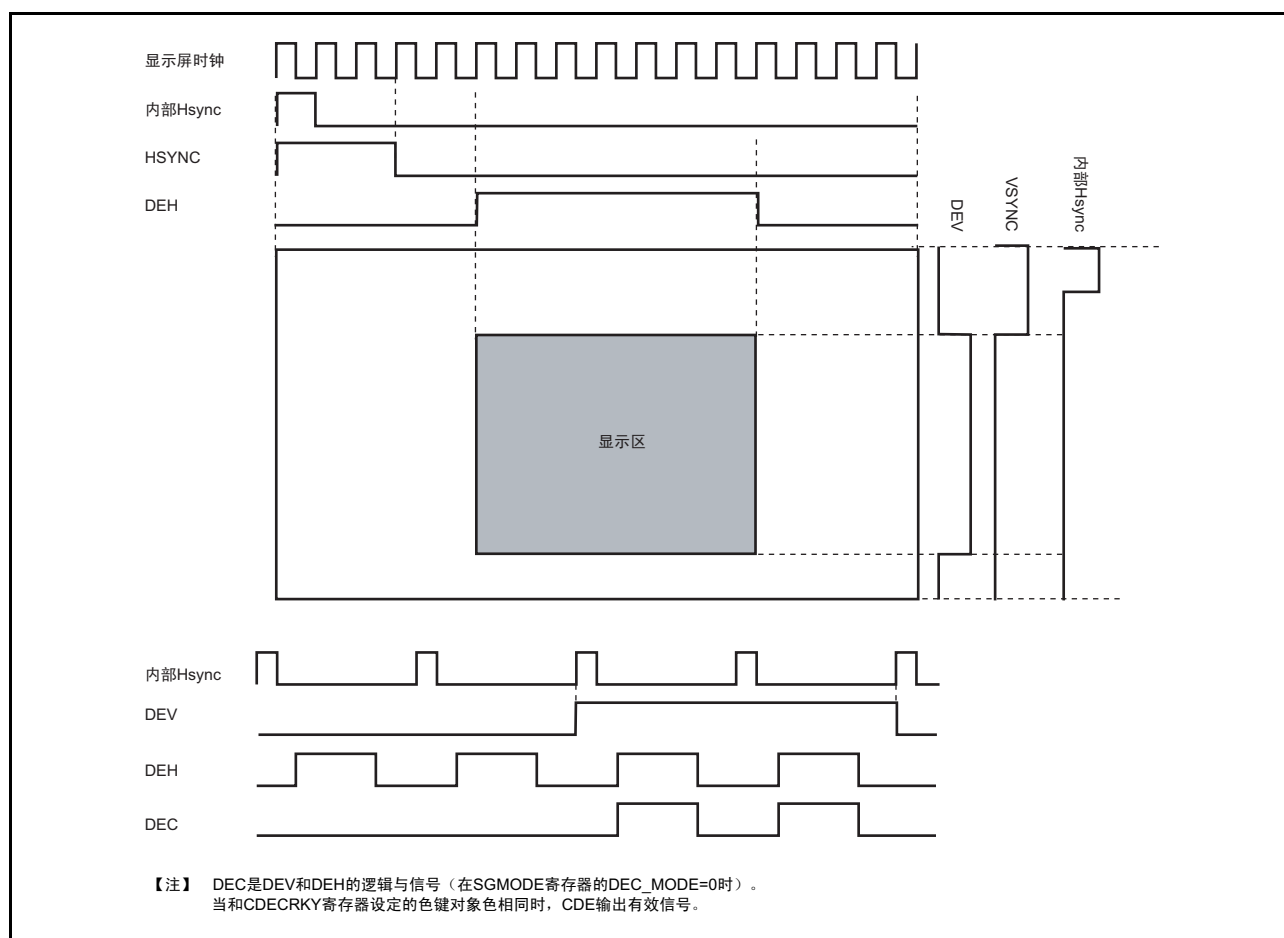


图 24.3 格式（输出 Vsync、Hsync、DEV、DEH、DEC、CDE）

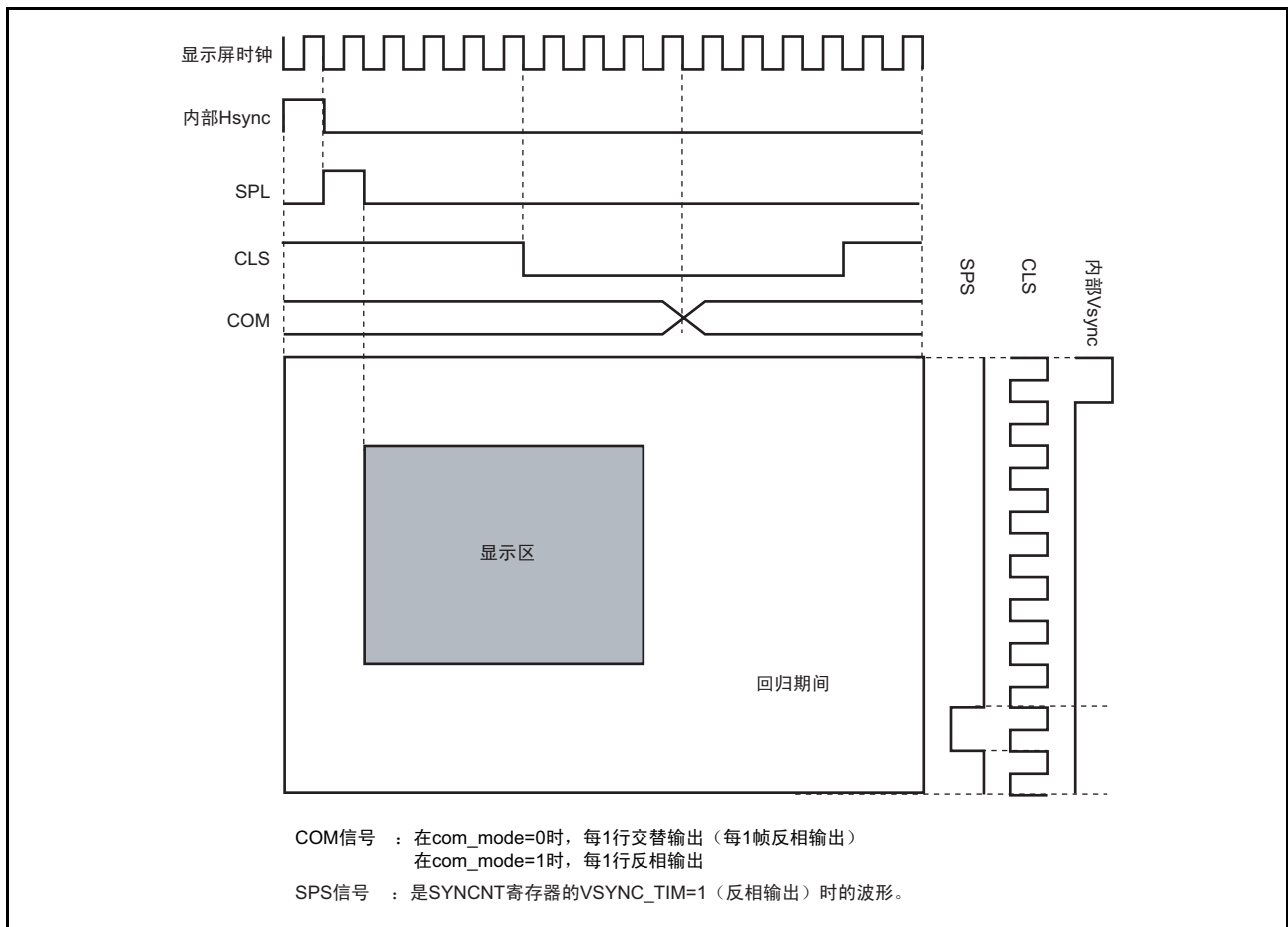


图 24.4 格式 (输出 SPS、SPL、CLS、COM)

24.5.3 外部同步模式

外部同步模式与从 TV、视频等外部同步信号发生电路输入的垂直 / 水平同步信号 (EX-VSYNC 和 EX-HSYNC) 同步进行图形图像的输出。必须将外部同步信号发生电路的垂直 / 水平同步信号和时钟分别输入到 EX-VSYNC、EX-HSYNC 和 DCLKIN 引脚, 并且进行以下同步信号相关寄存器的设定。

(1) 外部同步模式的设定

将 SGMODE 寄存器的 SYNC-SEL 置 1, 进入外部同步模式。在输入的垂直 / 水平同步信号为负极性时, 将 SYNCNT 寄存器的 EX-V-TYPE 和 EX-H-TYPE 置 1, 将输入信号反相。

(2) COM/CDE 引脚的输出设定

将 SGMODE 寄存器的 COM_CDE_SEL 和 CDE_EXE 置 1, 使 CDE 信号输出到 COM/CDE 引脚。当和 CDECRKY 寄存器设定的色键对象色相同时, CDE 信号输出有效电平。能通过 SYNCNT 寄存器的 COM-TYPE 位控制 CDE 信号的极性。

(3) 输入和输出时序的设定

设定输入的垂直 / 水平同步信号的采样时序、RGB 数据和 CDE 信号的输出时序。根据外部同步信号发生电路和显示器的规格, 设定是在 DCLKIN 的上升沿还是在下降沿进行输入 / 输出 (参照 SYNCNT 寄存器)。

在没有输入 EX_VSYNC 信号时，VDC2 停留在垂直回扫周期继续等待 EX_VSYNC 信号（Vsync 不进行自控制）。如果输入 EX_HSYNC 信号，VDC2 就结束水平显示，并且转移到下一行的控制。在没有输入 EX_HSYNC 信号时，停留在水平回扫周期继续等待 EX_HSYNC 信号（Hsync 不进行自控制）。

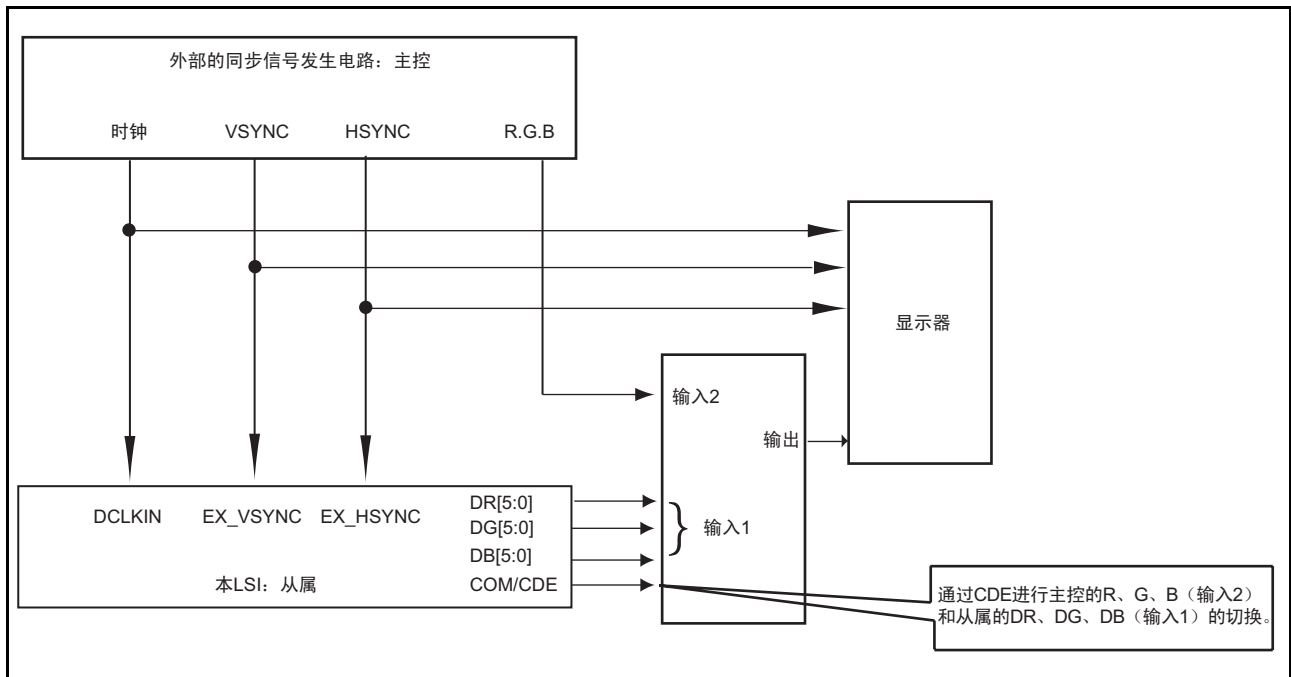


图 24.5 外部同步模式的信号流向

24.5.4 数字视频的输出

在进行 BTA T-1004 数字视频的输出时，生成符合 BTA T-1004（8:4:4 位格式并行接口）规格的 8 位亮度信号（Y）、色差信号（CB、CR）、EAV 和 SAV。

24.5.5 RGB565→YC444 的转换

根据 ITU-R BT601 规格，将 RGB565 转换为 YC444。首先，通过公式①将 RGB565 转换为 RGB888，使 R、G、B 值部超过 0 ~ 255 范围。然后，使用 BT601 规格的公式②（比色法），将 RGB 转换为 YC。

$$R_1 = \frac{255}{31} R$$

$$G_1 = \frac{255}{63} G \quad \dots \quad \textcircled{1}$$

$$B_1 = \frac{255}{31} B$$

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R_1 \\ G_1 \\ B_1 \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad \dots \quad \textcircled{2}$$

24.5.6 YC444→YC422 的转换

将 YC444 转换为 YCbCr422。时序图如图 24.5 所示。通过保持方式进行 YC444→YC422 的色度转换。

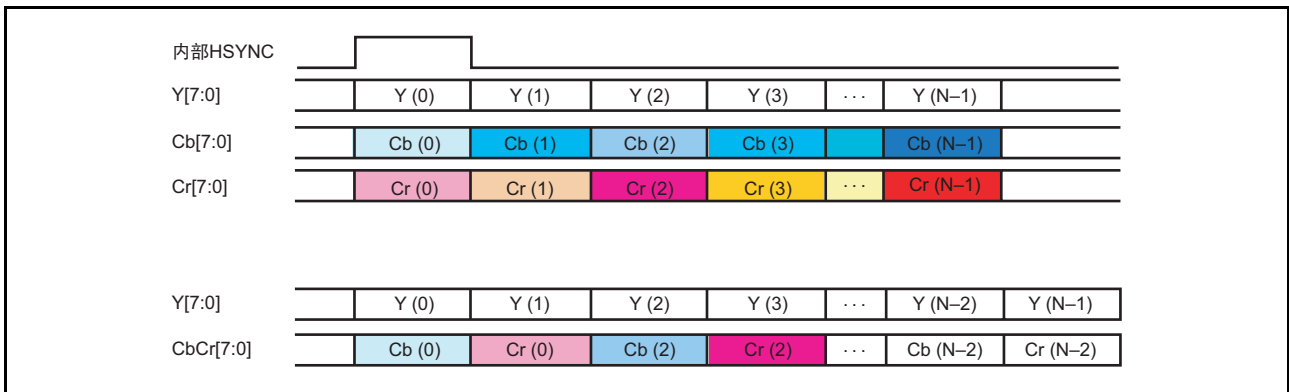


图 24.6 YC444→YC422 的转换时序图

24.5.7 数据允许信号 (合成)

通过 SGMODE 寄存器的 DEC_MODE，能选择是输出图形部生成的数据允许信号（层 1～层 4 的逻辑或）还是输出显示控制部生成的数据允许信号（矩形）。

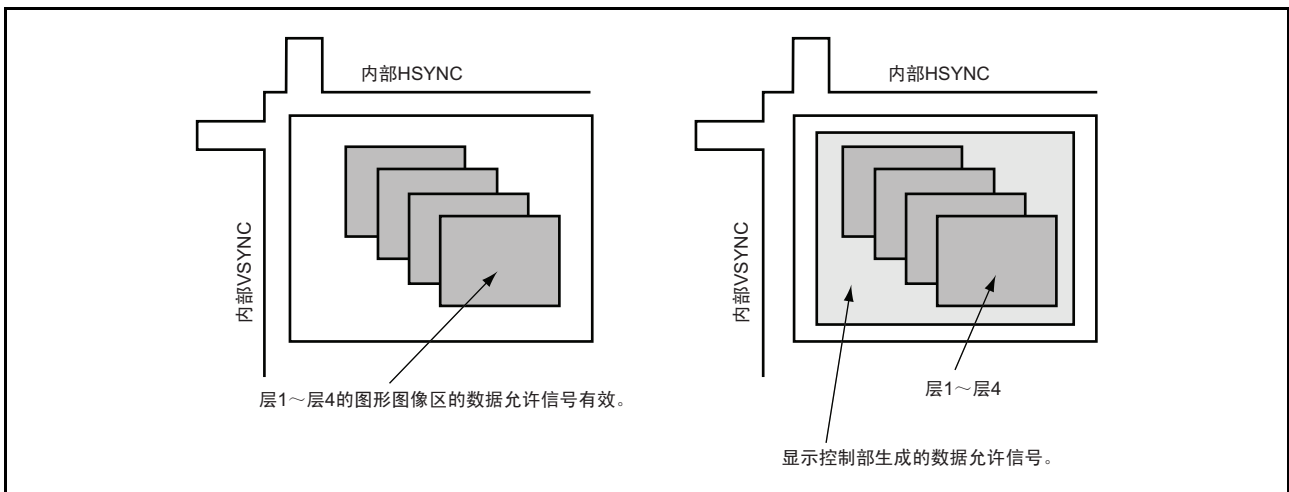


图 24.7 数据允许信号的说明图

24.6 寄存器说明

以下的寄存器设置分配到 SH 寄存器映像空间。

【寄存器说明的符号说明】

初始值 : 复位后的寄存器值

— : 不定值

R/W : 可读写。能读写值。

R/WC0 : 可读写。写 0 时该位被初始化, 忽视写 1。

R/WC1 : 可读写。写 1 时该位被初始化, 忽视写 0。

R : 可读。只能写 0。

—/W : 可写。读取值为不定值。

表 24.3 图形部 1 的寄存器结构

名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度
图形部控制寄存器	GRCMEN1	R/W	H'FFEC 0000	H'1FEC 0000	32/16/8
总线控制寄存器	GRCBUSCNT1	R/W	H'FFEC 0004	H'1FEC 0004	32/16/8
保留	—	R	H'FFEC 0008	H'1FEC 0008	32/16/8
保留	—	R	H'FFEC 000C	H'1FEC 000C	32/16/8
保留	—	R	H'FFEC 0300	H'1FEC 0300	32/16/8
保留	—	R	H'FFEC 0304	H'1FEC 0304	32/16/8
图形图像基址寄存器	GROPSADR1	R/W	H'FFEC 0308	H'1FEC 0308	32/16/8
图形图像区寄存器	GROPSWH1	R/W	H'FFEC 030C	H'1FEC 030C	32/16/8
图形图像行偏移寄存器	GROPSOFST1	R/W	H'FFEC 0310	H'1FEC 0310	32/16/8
图形图像起始位置寄存器	GRODPHV1	R/W	H'FFEC 0314	H'1FEC 0314	32/16/8
保留	—	R	H'FFEC 0318	H'1FEC 0318	32/16/8
保留	—	R	H'FFEC 031C	H'1FEC 031C	32/16/8
保留	—	R	H'FFEC 0320	H'1FEC 0320	32/16/8
保留	—	R	H'FFEC 0324	H'1FEC 0324	32/16/8
保留	—	R	H'FFEC 0328	H'1FEC 0328	32/16/8
图形图像区外的颜色设定寄存器	GROPBASERGB1	R/W	H'FFEC 032C	H'1FEC 032C	32/16/8

【注】 * P4 区地址是使用虚拟地址空间 P4 区的地址, 区域 7 地址是使用 TLB 存取物理地址空间区域 7 的地址。

表 24.4 图形部 2 的寄存器结构

名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度
图形部控制寄存器	GRCMEN2	R/W	H'FFED 0000	H'1FED 0000	32/16/8
总线控制寄存器	GRCBUSCNT2	R/W	H'FFED 0004	H'1FED 0004	32/16/8
保留	—	R	H'FFED 0008	H'1FED 0008	32/16/8
保留	—	R	H'FFED 000C	H'1FED 000C	32/16/8
保留	—	R	H'FFED 0300	H'1FED 0300	32/16/8
保留	—	R	H'FFED 0304	H'1FED 0304	32/16/8
图形图像基址寄存器	GROPSADR2	R/W	H'FFED 0308	H'1FED 0308	32/16/8
图形图像区寄存器	GROPSWH2	R/W	H'FFED 030C	H'1FED 030C	32/16/8

名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度
图形图像行偏移寄存器	GROPSOFST2	R/W	H'FFED 0310	H'1FED 0310	32/16/8
图形图像起始位置寄存器	GRODPHV2	R/W	H'FFED 0314	H'1FED 0314	32/16/8
α 控制区寄存器	GROPEWH2	R/W	H'FFED 0318	H'1FED 0318	32/16/8
α 控制区起始位置寄存器	GROPEDPHV2	R/W	H'FFED 031C	H'1FED 031C	32/16/8
α 控制寄存器	GROPEPDA2	R/W	H'FFED 0320	H'1FED 0320	32/16/8
色键控制寄存器	GROPCRKY0_2	R/W	H'FFED 0324	H'1FED 0324	32/16/8
色度颜色指定寄存器	GROPCRKY1_2	R/W	H'FFED 0328	H'1FED 0328	32/16/8
图形图像区外的颜色设定寄存器	GROPBASERGB2	R/W	H'FFED 032C	H'1FED 032C	32/16/8

【注】 * P4 区地址是使用虚拟地址空间 P4 区的地址，区域 7 地址是使用 TLB 存取物理地址空间区域 7 的地址。

表 24.5 图形部 3 的寄存器结构

名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度
图形部控制寄存器	GRCMEN3	R/W	H'FFEE 0000	H'1FEE 0000	32/16/8
总线控制寄存器	GRCBUSCNT3	R/W	H'FFEE 0004	H'1FEE 0004	32/16/8
保留	—	R	H'FFEE 0008	H'1FEE 0008	32/16/8
保留	—	R	H'FFEE 000C	H'1FEE 000C	32/16/8
保留	—	R	H'FFEE 0300	H'1FEE 0300	32/16/8
保留	—	R	H'FFEE 0304	H'1FEE 0304	32/16/8
图形图像基址寄存器	GROPSADR3	R/W	H'FFEE 0308	H'1FEE 0308	32/16/8
图形图像区寄存器	GROPSWH3	R/W	H'FFEE 030C	H'1FEE 030C	32/16/8
图形图像行偏移寄存器	GROPSOFST3	R/W	H'FFEE 0310	H'1FEE 0310	32/16/8
图形图像起始位置寄存器	GRODPHV3	R/W	H'FFEE 0314	H'1FEE 0314	32/16/8
α 控制区寄存器	GROPEWH3	R/W	H'FFEE 0318	H'1FEE 0318	32/16/8
α 控制区起始位置寄存器	GROPEDPHV3	R/W	H'FFEE 031C	H'1FEE 031C	32/16/8
α 控制寄存器	GROPEPDA3	R/W	H'FFEE 0320	H'1FEE 0320	32/16/8
色键控制寄存器	GROPCRKY0_3	R/W	H'FFEE 0324	H'1FEE 0324	32/16/8
色度颜色指定寄存器	GROPCRKY1_3	R/W	H'FFEE 0328	H'1FEE 0328	32/16/8
图形图像区外的颜色设定寄存器	GROPBASERGB3	R/W	H'FFEE 032C	H'1FEE 032C	32/16/8

【注】 * P4 区地址是使用虚拟地址空间 P4 区的地址，区域 7 地址是使用 TLB 存取物理地址空间的区域 7 的地址。

表 24.6 图形部 4 的寄存器结构

名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度
图形部控制寄存器	GRCMEN4	R/W	H'FFEF 0000	H'1FEF 0000	32/16/8
总线控制寄存器	GRCBUSCNT4	R/W	H'FFEF 0004	H'1FEF 0004	32/16/8
保留	—	R	H'FFEF 0008	H'1FEF 0008	32/16/8
保留	—	R	H'FFEF 000C	H'1FEF 000C	32/16/8
保留	—	R	H'FFEF 0300	H'1FEF 0300	32/16/8
保留	—	R	H'FFEF 0304	H'1FEF 0304	32/16/8
图形图像基址寄存器	GROPSADR4	R/W	H'FFEF 0308	H'1FEF 0308	32/16/8
图形图像区寄存器	GROPSWH4	R/W	H'FFEF 030C	H'1FEF 030C	32/16/8
图形图像行偏移寄存器	GROPSOFST4	R/W	H'FFEF 0310	H'1FEF 0310	32/16/8

名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度
图形图像开始位置寄存器	GRODPHV4	R/W	H'FFEF 0314	H'1FEF 0314	32/16/8
α 控制区域寄存器	GROPEWH4	R/W	H'FFEF 0318	H'1FEF 0318	32/16/8
α 控制区起始位置寄存器	GROPEDPHV4	R/W	H'FFEF 031C	H'1FEF 031C	32/16/8
α 控制寄存器	GROPEDPA4	R/W	H'FFEF 0320	H'1FEF 0320	32/16/8
色键控制寄存器	GROPCRKY0_4	R/W	H'FFEF 0324	H'1FEF 0324	32/16/8
色度颜色指定寄存器	GROPCRKY1_4	R/W	H'FFEF 0328	H'1FEF 0328	32/16/8
图形图像区外的颜色设定寄存器	GROPBASERGB4	R/W	H'FFEF 032C	H'1FEF 032C	32/16/8

【注】 * P4 区地址是使用虚拟地址空间 P4 区的地址，区域 7 地址是使用 TLB 存取物理地址空间区域 7 的地址。

表 24.7 显示控制部的寄存器结构

名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度
SG 模式设定寄存器	SGMODE	R/W	H'FFEB 0000	H'1FEB 0000	32/16/8
中断输出控制寄存器	SGINTCNT	R/W	H'FFEB 0004	H'1FEB 0004	32/16/8
同步信号控制寄存器	SYNCNT	R/W	H'FFEB 0008	H'1FEB 0008	32/16/8
外部输入同步信号时序控制寄存器	EXTSYNCNT	R/W	H'FFEB 000C	H'1FEB 000C	32/16/8
保留	—	R	H'FFEB 0100	H'1FEB 0100	32/16/8
同步信号大小寄存器	SYNSIZE	R/W	H'FFEB 0104	H'1FEB 0104	32/16/8
垂直同步信号时序控制寄存器	VSYNCTIM	R/W	H'FFEB 0108	H'1FEB 0108	32/16/8
水平同步信号时序控制寄存器	HSYNCTIM	R/W	H'FFEB 010C	H'1FEB 010C	32/16/8
栅极时钟信号时序控制寄存器	CLSTIM	R/W	H'FFEB 0110	H'1FEB 0110	32/16/8
采样开始信号时序控制寄存器	SPLTIM	R/W	H'FFEB 0118	H'1FEB 0118	32/16/8
栅极控制信号时序控制寄存器	COMTIM	R/W	H'FFEB 011C	H'1FEB 011C	32/16/8
SGDE 区起始位置寄存器	SGDESTART	R/W	H'FFEB 0120	H'1FEB 0120	32/16/8
SGDE 区大小寄存器	SGDESIZE	R/W	H'FFEB 0124	H'1FEB 0124	32/16/8
CDE 色度颜色指定寄存器	CDECRKY	R/W	H'FFEB 0128	H'1FEB 0128	32/16/8
保留	—	R	H'FFEB 0148	H'1FEB 0148	32/16/8
T-1004 控制寄存器	T1004CNT	R/W	H'FFEB 0200	H'1FEB 0200	32/16/8
T-1004 视频起始位置寄存器	T1004OFFSET	R/W	H'FFEB 0204	H'1FEB 0204	32/16/8
保留	—	R	H'FFEB 0208	H'1FEB 0208	32/16/8
保留	—	R	H'FFEB 020C	H'1FEB 020C	32/16/8

【注】 * P4 区地址是使用虚拟地址空间 P4 区的地址，区域 7 地址是使用 TLB 存取物理地址空间区域 7 的地址。

24.6.1 图形部控制寄存器 (GRCMEN1、2、3、4)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEN	VEN
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
31	WE	0	R/W	寄存器值传送允许位 在此位为 1 时, 与 Vsync 同步传送寄存器 (对象寄存器: H'000 ~ H'31C、H'32C 寄存器) 的值。在通过 Vsync 传送寄存器的值后此位被清 0。
30 ~ 2	—	全 0	R	保留位 读取值为 0, 只能写 0。
1	DEN	0	R/W	图形显示允许位 0: 不允许 1: 允许显示
0	VEN	0	R/W	下层图形显示允许位 0: 不允许 1: 允许显示

表 24.8 显示允许位的说明

DEN	VEN	运行	输出	控制
0	0	不读存储器的图像数据, 也不进行下层图形的显示处理。	全面输出 GROPBASERGB 寄存器设定的颜色 (允许信号无效)。	
0	1	不读存储器的图像数据, 但是进行下层图形的显示处理。	只显示下层图形 (输出下层图形的允许信号)。	只显示下层图形。
1	0	读存储器的图像数据, 但是不进行下层图形的显示处理。	只显示当前图形 (输出当前图形的允许信号)。	只显示当前图形。
1	1	读存储器的图像数据, 并且进行下层图形的显示处理。	处理并显示指定当前图形和下层图形 (输出当前图形和下层图形的允许信号的逻辑或)。	显示当前图形和下层图形。

- 【注】**
1. 当 α 控制区 (参照 GROPEWH 和 GROPEDPHV 寄存器) 大于图形图像区 (参照 GROPSWH 和 GRODPHV 寄存器) 时, 只显示下层图形图像。
 2. 需对每一层进行设定。当高层的 (DEN, VEN)=(0,0) 时, 不输出低层的图形。

24.6.2 总线控制寄存器 (GRCBUSCNT1、2、3、4)

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ENDIAN
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
31 ~ 1	—	全 0	R	保留位 读取值为 0，只能写 0。
0	ENDIAN	0	R/W	设定像素总线的字节排序。 0: 小端法 1: 大端法

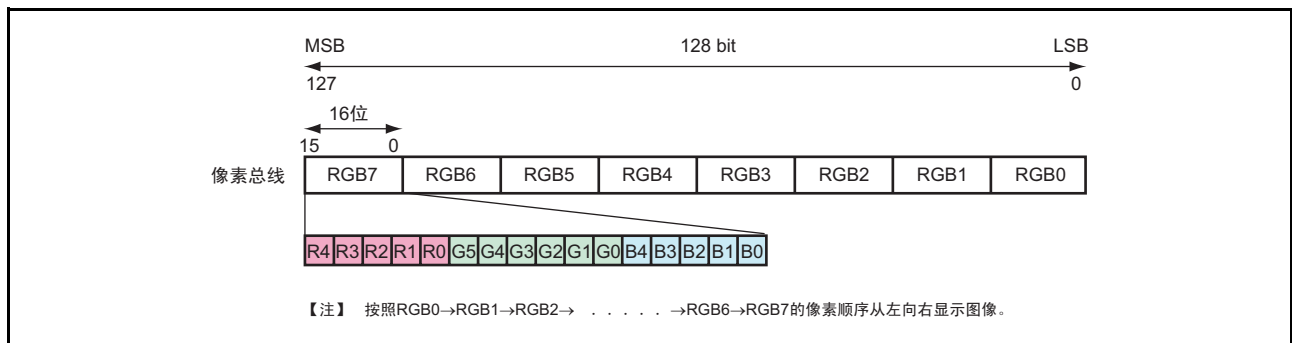


图 24.8 像素总线的字节排序 (ENDIAN=0)

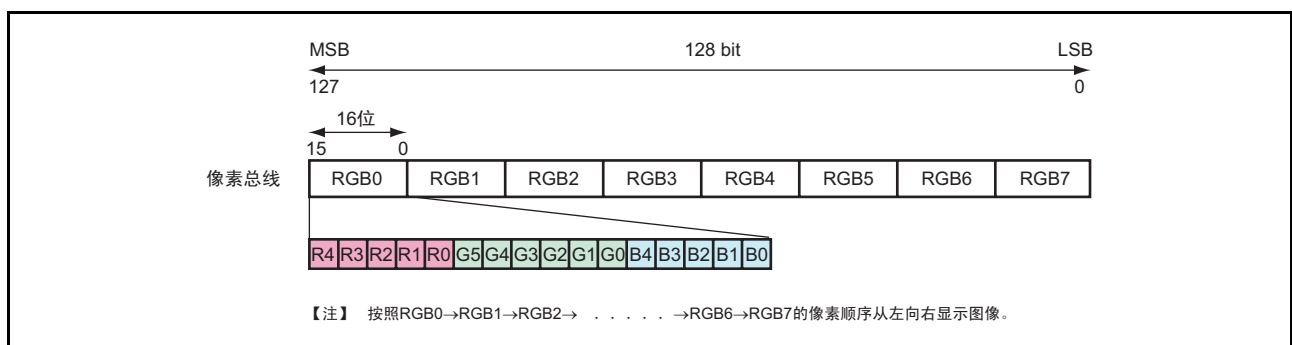


图 24.9 像素总线的字节排序 (ENDIAN=1)

24.6.3 图形图像基址寄存器 (GROPSADR1、2、3、4)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	GROPSADR[28:16]												
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/W	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
	GROPSADR[15:0]															
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 29	—	全 0	R	保留位 读取值为 0，只能写 0。
28 ~ 0	GROPSADR [28:0]	H'0000000	R/W	设定图形图像的读地址。最低位必须固定为 0。

【注】 作为 16 位 RGB 数据处理，所以不能处理跨越 2 字节边界的调整数据。

24.6.4 图形图像区寄存器 (GROPSWH1、2、3、4)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GROPSH[9:0]									
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/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
	—	—	—	—	—	—	GROPSW[9:0]									
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/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 ~ 26	—	全 0	R	保留位 读取值为 0，只能写 0。
25 ~ 16	GROPSH[9:0]	H'000	R/W	设定图形图像区的高度 [行数]。
15 ~ 10	—	全 0	R	保留位 读取值为 0，只能写 0。
9 ~ 0	GROPSW[9:0]	H'000	R/W	设定图形图像区的宽度 [显示屏的时钟周期数]。

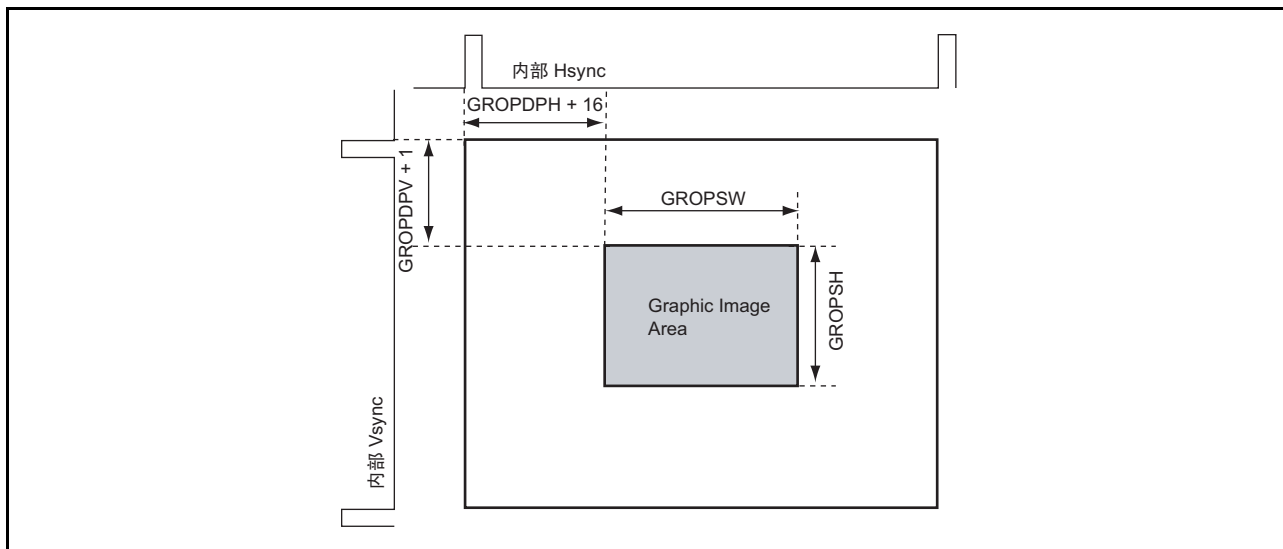


图 24.10 有关于形图像区设定的补充图 (读存储器)

必须在以下范围内设定图形区，否则不保证运行：

(1H 的显示屏时钟数) > GROPSW (宽度) + GRODPH (水平显示起始位置) + (16 显示屏时钟)

(1Frame 的行数) > GROPSH (高度) + GRODPV (垂直显示起始位置) + (1 行)

24.6.5 图形图像行偏移寄存器 (GROPSOFST1、2、3、4)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	GROPSOFST[28:16]												
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/W	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
	GROPSOFST[15:0]															
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 29	—	全 0	R	保留位 读取值为 0，只能写 0。
28 ~ 0	GROPSOFST [28:0]	H'0000000	R/W	设定图形图像的行偏移量。低 4 位必须固定为 0000。

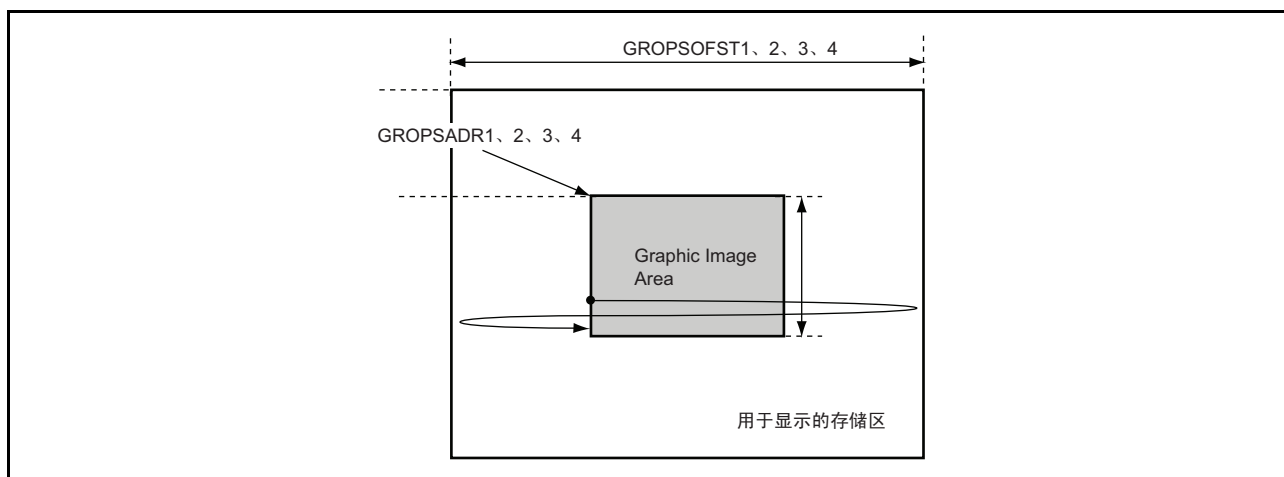


图 24.11 图形图像存储区的补充图

给基址寄存器 (GROPSADR1、2、3、4) 的值加上行偏移量 (GROPSOFST1、2、3、4) × n 的值就是第 n 行的起始 (左端) 地址。

24.6.6 图形图像起始位置寄存器 (GROPDPHV1、2、3、4)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GROPDPV[9:0]									
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/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
	—	—	—	—	—	—	GROPDPH[9:0]									
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/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 ~ 26	—	全 0	R	保留位 读取值为 0，只能写 0。
25 ~ 16	GROPDPV[9:0]	H'000	R/W	设定图形图像区的垂直显示起始位置 [行数]。
15 ~ 10	—	全 0	R	保留位 读取值为 0，只能写 0。
9 ~ 0	GROPDPH[9:0]	H'000	R/W	设定图形图像区的水平显示起始位置 [显示屏的时钟周期数]。

【注】 显示起始位置存在偏移量 (参照图 24.9)。
 垂直方向: GROPDV 寄存器的设定值+1 行
 水平方向: GRODPH 寄存器的设定值+16 个显示屏时钟

24.6.7 α 控制区寄存器 (GROPEWH2、3、4)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GROPEH[9:0]									
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/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
	—	—	—	—	—	—	GROPEW[9:0]									
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/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 ~ 26	—	全 0	R	保留位 读取值为 0，只能写 0。
25 ~ 16	GROPEH[9:0]	H'000	R/W	设定 α 控制区的高度 [行数]。
15 ~ 10	—	全 0	R	保留位 读取值为 0，只能写 0。
9 ~ 0	GROPEW[9:0]	H'000	R/W	设定 α 控制区的宽度 [显示屏的时钟周期数]。

【注】 层 1 为最下层图像，没有 α 控制对象，所以禁止设定。

设定 α 控制区（矩形）的大小。请参照图 24.12。

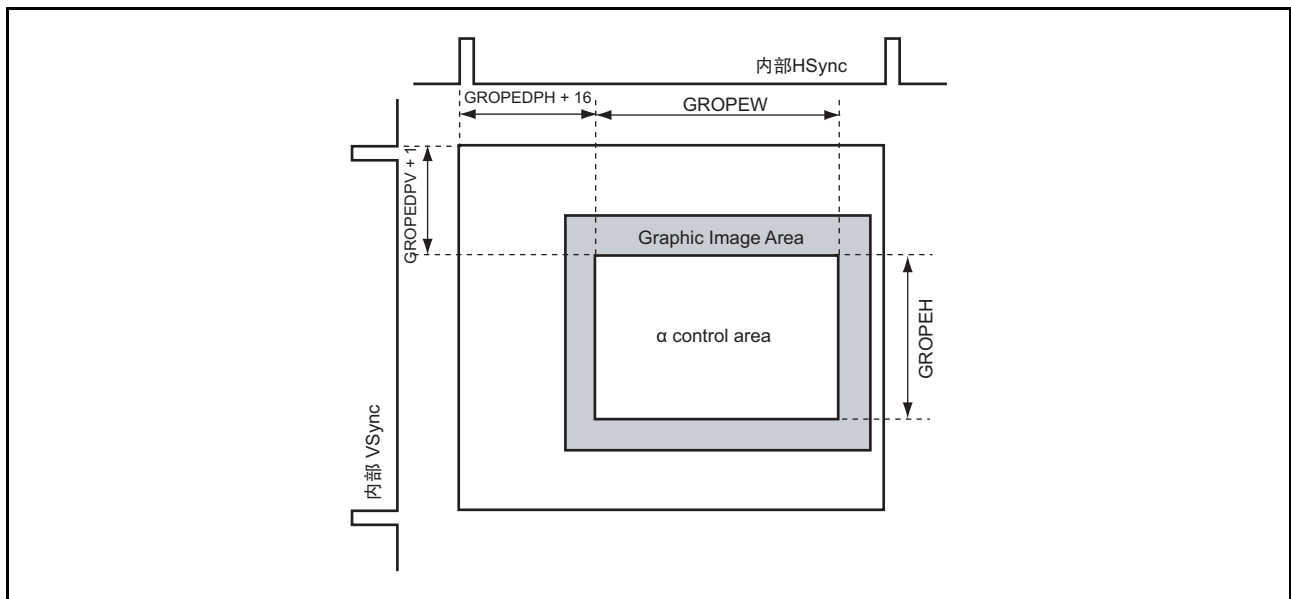
24.6.8 α 控制区起始位置寄存器 (GROPEDPHV2、3、4)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GROPEDPV[9:0]									
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/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
	—	—	—	—	—	—	GROPEDPH[9:0]									
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/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 ~ 26	—	全 0	R	保留位 读取值为 0，只能写 0。
25 ~ 16	GROPEDPV [9:0]	H'000	R/W	设定 α 控制区的垂直起始位置 [行数]。
15 ~ 10	—	全 0	R	保留位 读取值为 0，只能写 0。
9 ~ 0	GROPEDPH [9:0]	H'000	R/W	设定 α 控制区的水平起始位置 [显示屏的时钟周期数]。

【注】 层 1 为最下层图像，没有 α 控制对象，所以禁止设定。

图 24.12 有关 α 控制区设定的补充图

24.6.9 α 控制寄存器 (GROPEDPA2、3、4)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEFA[7:0]								ACOE[7:0]							
初始值:	1	1	1	1	1	1	1	1	0	0	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
	ARATE[7:0]								WE	—	—	AST	—	AMOD[1:0]	AEN	
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 24	DEFA[7:0]	H'FF	R/W	设定 α 值的初始值。
23 ~ 16	ACOE[7:0]	H'00	R/W	设定 α 值的运算系数。用此值对 DEFA 值进行加减运算。
15 ~ 8	ARATE[7:0]	H'00	R/W	设定控制 α 值的帧频 (将 480pVsync 作为计数源)。
7	WE	0	R/W	α 控制的寄存器值传送允许位 在此位为 1 时, 与 Vsync 同步传送寄存器 (H'320 ~ H'328) 的值。 在通过 Vsync 传送寄存器的值后此位被清 0。 0: 禁止传送 1: 允许传送
6、5	—	全 0	R	保留位 读取值为 0, 只能写 0。
4	AST	0	R/W	α 混和处理状态标志位 0: 加减处理结束 1: 加减处理中
3	—	0	R	保留位 读取值为 0, 只能写 0。
2、1	AMOD[1:0]	00	R/W	设定 α 处理模式。 00: α 值初始值 (不变) 01: α 值加法处理 10: α 值减法处理 11: 禁止设定
0	AEN	0	R/W	设定是否允许 α 控制。 0: 不允许 α 控制 (α 值相当于 1) 1: 允许 α 控制

【注】 层 1 为最下层图像, 没有 α 控制对象, 所以禁止设定。

在 AEN=“1”并且 WE=“1”时, 通过 Vsync 时序将 α 值加载到内部。

在 AMOD[1:0]=[00] 时, DEFA 指定的 α 值适用于视频区。

在 AMOD[1:0]=[01] 时, 视频区中的 α 值是按照字段速率给 DEFA 加上 ACOEF 值的结果。当 α 值大于等于 255 时, 停止处理 (淡出)。

在 AMOD[1:0]=[10] 时, 视频区中的 α 值是按照字段速率将 DEFA 值减去 ACOEF 值的结果。当 α 值小于等于 0 时, 停止处理 (淡入)。

表 24.9 α 值的混合比率

α 值 (10 进制)	图形	下层图形
255	256/256	0/256
254	254/256	1/256
253	253/256	2/256
252	252/256	3/256
·	·	·
·	·	·
2	2/256	253/256
1	1/256	254/256
0	0/256	256/256

24.6.10 色键控制寄存器 (GROPCRKY0_2、3、4)

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

位	位名	初始值	R/W	说明
31 ~ 17	—	全 0	R	保留位 读取值为 0, 只能写 0。
16	CKEN	0	R/W	设定是否允许色键操作。 0: 不允许色键操作 1: 允许色键操作
15 ~ 11	CROMAKR[4:0]	00000	R/W	色键对象色 R
10 ~ 5	CROMAKG[5:0]	000000	R/W	色键对象色 G
4 ~ 0	CROMAKB[4:0]	00000	R/W	色键对象色 B

【注】 层 1 为最下层图像, 没有 α 控制对象, 所以禁止设定。

在 GROPEDPA 寄存器的 WE=“1”时, 通过内部 Vsync 时序将色键加载到内部。

通过允许色键操作, 将图形数据 (RGB16 格式) 与 CROMAKR[4:0], CROMAKG[5:0], CROMAKB[4:0] 完全相同的像素转换为色键颜色指定寄存器 (GROPCRKY1) 指定的颜色 (RGB16 格式), 并且通过 ALPHA[7:0] 进行 α 控制。

通过此操作, 能得到在下层图形上显示文字和光标等效果。

24.6.11 色度颜色指定寄存器 (GROPCRKY1_2、3、4)

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

位	位名	初始值	R/W	说明
31 ~ 24	—	全 0	R	保留位 读取值为 0, 只能写 0。
23 ~ 16	ALPHA[7:0]	H'00	R/W	替换后的 α 值
15 ~ 11	R[4:0]	00000	R/W	替换后的 R 值
10 ~ 5	G[5:0]	000000	R/W	替换后的 G 值
4 ~ 0	B[4:0]	00000	R/W	替换后的 B 值

【注】 层 1 为最下层图像, 没有 α 控制对象, 所以禁止设定。

当与色键 RGB 相同时, 设定要替换的颜色信息。

进行以下的 α 运算

输出 $R=R$ (当前图形图像) $\times\alpha+R$ (下层图形图像) $\times(1-\alpha)$

输出 $G=G$ (当前图形图像) $\times\alpha+G$ (下层图形图像) $\times(1-\alpha)$

输出 $B=B$ (当前图形图像) $\times\alpha+B$ (下层图形图像) $\times(1-\alpha)$

24.6.12 图形图像区外的颜色设定寄存器 (GROPBASERGB1、2、3、4)

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
	BASE_R[4:0]					BASE_G[5:0]					BASE_B[4:0]					
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 16	—	全 0	R	保留位 读取值为 0, 只能写 0。
15 ~ 11	BASE_R[4:0]	00000	R/W	图形图像区外的 R 值。
10 ~ 5	BASE_G[5:0]	000000	R/W	图形图像区外的 G 值。
4 ~ 0	BASE_B[4:0]	00000	R/W	图形图像区外的 B 值。

【注】 本功能只在 GRCMEN 寄存器的 VEN=0 时有效。

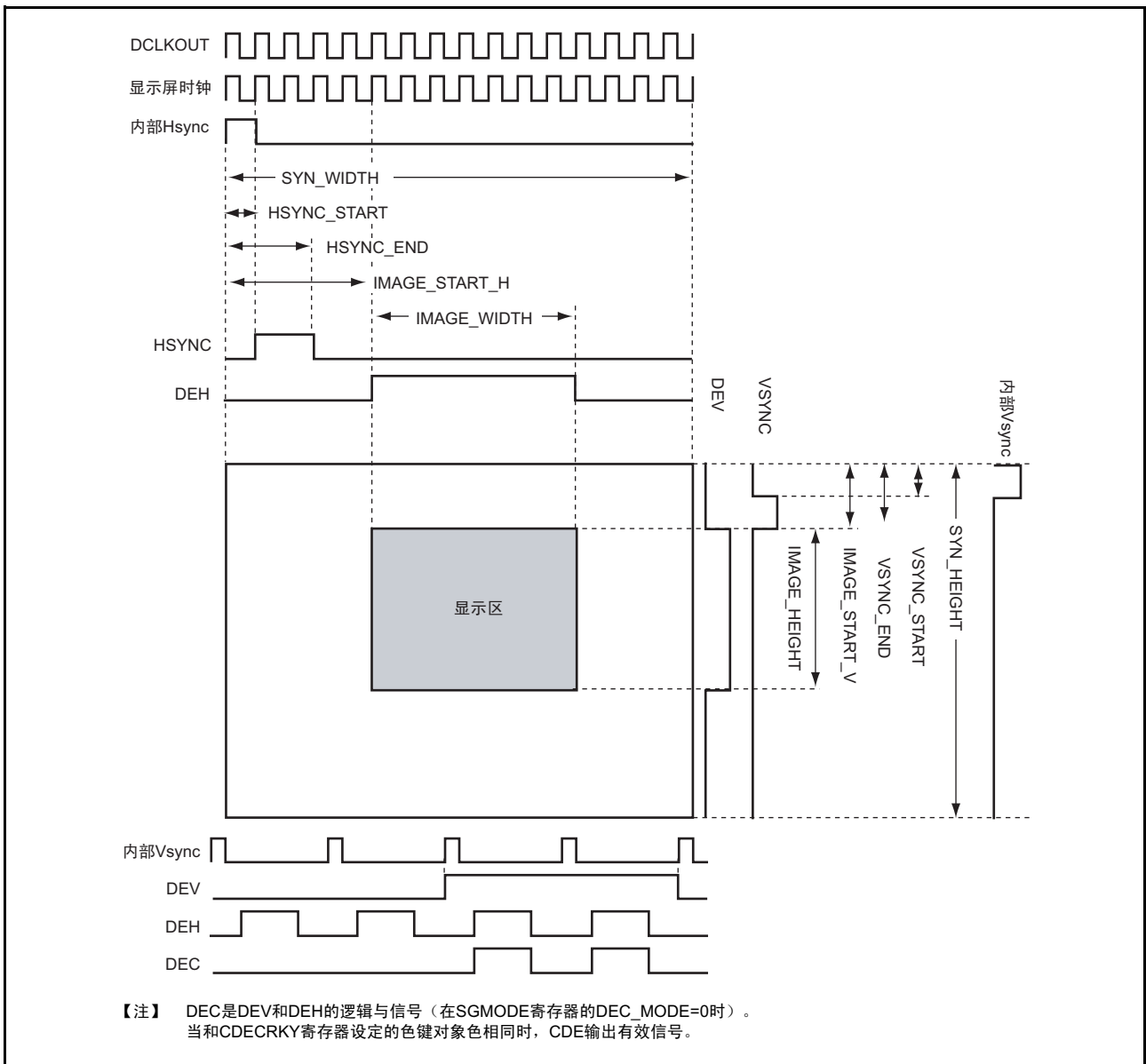


图 24.13 屏幕格式

24.6.13 SG 模式设定寄存器 (SGMODE)

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

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

位	位名	初始值	R/W	说明
31	WE	0	R/W	寄存器值传送允许位 *1 在此位为 1 时, 传送寄存器的值 (对象寄存器为: H'FFEB_0000 ~ H'FFEB_0208 寄存器)。
30 ~ 10	—	全 0	R	保留位 读取值为 0, 只能写 0。
9	COM_CDE_SEL	0	R/W	COM 和 CDE 的输出选择位 0: 输出 COM 信号 1: 输出 CDE 信号
8	CDE_EXE	0	R/W	CDE 的运行允许位 在更新时, 与内部 Vsync 同步。 0: 不允许 CDE 运行 (在 SYNCNT 寄存器的 COM_TYPE 位 =0 时, CDE 总是输出 “0”) 1: 允许 CDE 运行
7 ~ 6	—	全 0	R	保留位 读取值为 0, 只能写 0。
5	DE_SEL	0	R/W	DEH 和 DEC 的输出选择位 0: 输出 DEH (水平方向的数据允许信号) 1: 输出 DEC (水平、垂直方向的合成数据允许信号)
4	DEC_MODE	0	R/W	允许模式选择位 0: 输出 SGDESTART 和 SGDESIZE 寄存器设定的数据允许信号 1: 输出图形部生成的数据允许信号 (合成信号: 各层数据允许信号的逻辑或) *2
3 ~ 2	—	全 0	R	保留位 读取值为 0, 只能写 0。
1	SYNC_SEL	0	R/W	同步信号选择位 0: 使用内部同步信号 1: 使用外部同步信号 (5 个显示屏时钟周期进行延迟)
0	—	0	R	保留位 读取值为 0, 只能写 0。

【注】 *1 必须在更改寄存器值 (H'000 ~ H'208) 前将 WE 位清 0, 在更改寄存器值后将此位置 1。

*2 在将 DEC_MODE 位置 1 时, 还必须将 DE_SEL 位置 1。

24.6.14 中断输出控制寄存器 (SGINTCNT)

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
	—	—	—	—	—	—	—	—	—	—	—	VSYNC MASK	—	—	—	VSYNC STATUS
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/WC0

位	位名	初始值	R/W	说明
31 ~ 5	—	全 0	R	保留位 读取值为 0，只能写 0。
4	VSYNC_MASK	0	R/W	VSYNC 中断屏蔽位 *1 0: 允许中断 1: 屏蔽中断
3 ~ 1	—	全 0	R	保留位 读取值为 0，只能写 0。
0	VSYNC_STATUS	1	R/WC0	VSYNC 中断状态位 *2 0: 有中断 1: 无中断

【注】 *1 在将中断屏蔽位置 1 时，清除中断状态。

*2. 在将中断状态位置 0 时，清除中断状态。

从 VDC 部的 db_n_int_n 信号引脚 (L 有效) 输出 VSYNC_STATUS 值。

24.6.15 同步信号控制寄存器 (SYNCNT)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	RGB_TIM	—	—	EX_V_TIM	EX_H_TIM	—	—	VSYNC_TIM	HSYNC_TIM	DEV_TIM	DEH_TIM	DEC_TIM	COM_TIM
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	EX_V_TYPE	EX_H_TYPE	—	—	VSYNC_TYPE	HSYNC_TYPE	DEV_TYPE	DEH_TYPE	DEC_TYPE	COM_TYPE
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 29	—	全 0	R	保留位 读取值为 0，只能写 0。
28	RGB_TIM	0	R/W	RGB 数据输出时序设定位 0: 在显示屏时钟的上升沿，输出 RGB 数据。 1: 在显示屏时钟的下降沿，输出 RGB 数据。
27 ~ 26	—	全 0	R	保留位 读取值为 0，只能写 0。
25	EX_V_TIM	0	R/W	外部 VSYNC 输入时序设定位 0: 在显示屏时钟的上升沿，锁定外部 VSYNC。 1: 在显示屏时钟的下降沿，锁定外部 VSYNC。
24	EX_H_TIM	0	R/W	外部 HSYNC 输入时序设定位 0: 在显示屏时钟的上升沿，锁定外部 HSYNC。 1: 在显示屏时钟的下降沿，锁定外部 HSYNC。
23 ~ 22	—	全 0	R	保留位 读取值为 0，只能写 0。
21	VSYNC_TIM	0	R/W	VSYNC/SPS 输出时序设定位 0: 在显示屏时钟的上升沿，输出 VSYNC/SPS。 1: 在显示屏时钟的下降沿，输出 VSYNC/SPS。
20	HSYNC_TIM	0	R/W	HSYNC/SPL 输出时序设定位 0: 在显示屏时钟的上升沿，输出 HSYNC/SPL。 1: 在显示屏时钟的下降沿，输出 HSYNC/SPL。
19	DEV_TIM	0	R/W	DEV/CLS 输出时序设定位 0: 在显示屏时钟的上升沿，输出 DEV/CLS。 1: 在显示屏时钟的下降沿，输出 DEV/CLS。
18	DEH_TIM	0	R/W	DEH/LP 输出时序设定位 0: 在显示屏时钟的上升沿，输出 DEH/LP。 1: 在显示屏时钟的下降沿，输出 DEH/LP。
17	DEC_TIM	0	R/W	DEC/PS 输出时序设定位 0: 在显示屏时钟的上升沿，输出 DEC/PS。 1: 在显示屏时钟的下降沿，输出 DEC/PS。
16	COM_TIM	0	R/W	COM/CDE 输出时序设定位 0: 在显示屏时钟的上升沿，输出 COM/CDE。 1: 在显示屏时钟的下降沿，输出 COM/CDE。

位	位名	初始值	R/W	说明
15 ~ 10	—	全 0	R	保留位 读取值为 0，只能写 0。
9	EX_V_TYPE	0	R/W	外部 VSYNC 输入反相控制位 0: 不反相 1: 反相输入
8	EX_H_TYPE	0	R/W	外部 HSYNC 输入反相控制位 0: 不反相 1: 反相输入
7 ~ 6	—	全 0	R	保留位 读取值为 0，只能写 0。
5	VSYNC_TYPE	0	R/W	VSYNC/SPS 反相控制位 0: 不反相 1: 反相输出
4	HSYNC_TYPE	0	R/W	HSYNC/SPL 反相控制位 0: 不反相 1: 反相输出
3	DEV_TYPE	0	R/W	DEV/CLS 反相控制位 0: 不反相 1: 反相输出
2	DEH_TYPE	0	R/W	DEH/LP 反相控制位 0: 不反相 1: 反相输出
1	DEC_TYPE	0	R/W	DEC/PS 反相控制位 0: 不反相 1: 反相输出
0	COM_TYPE	0	R/W	COM/CDE 反相控制位 0: 不反相 1: 反相输出

24.6.16 外部输入同步信号时序控制寄存器 (EXTSYNCNT)

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

位	位名	初始值	R/W	说明
31 ~ 17	—	全 0	R	保留位 读取值为 0，只能写 0。
16	EX_STATUS	0	R	外部 VSYNC 和 HSYNC 输入的相位状态位 0: VSYNC 和 HSYNC 的相位一致 1: VSYNC 和 HSYNC 的相位偏离
15 ~ 6	—	全 0	R	保留位 读取值为 0，只能写 0。
5 ~ 4	EX_V_DLY[1:0]	H'0	R/W	外部 VSYNC 输入延迟位 [点时钟周期] 00: 无延迟 01: 延迟 1 个点时钟周期 10: 延迟 2 个点时钟周期 11: 延迟 3 个点时钟周期
3 ~ 2	—	全 0	R	保留位 读取值为 0，只能写 0。
1 ~ 0	EX_H_DLY[1:0]	H'0	R/W	外部 HSYNC 输入延迟位 [点时钟周期]。 00: 无延迟 01: 延迟 1 个点时钟周期 10: 延迟 2 个点时钟周期 11: 延迟 3 个点时钟周期

外部输入同步信号的输入时序假定了对应 VESA 规格的液晶显示屏的水平和垂直同步信号时序。在对外部输入的同步信号进行内部采样时，此寄存器进行相位调整。

24.6.17 同步信号大小寄存器 (SYNSIZE)

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

位	位名	初始值	R/W	说明
31 ~ 26	—	全 0	R	保留位 读取值为 0, 只能写 0。
25 ~ 16	SYN_HEIGHT [9:0]	H'20D	R/W	设定包含垂直方向回扫周期的高度 [行数]。 初始值: H'20D=525 行
15 ~ 11	—	全 0	R	保留位 读取值为 0, 只能写 0。
10 ~ 0	SYN_WIDTH [10:0]	H'35A	R/W	设定包含水平方向回扫周期的宽度 [显示屏的时钟周期]。 初始值: H'35A=858 点

24.6.18 垂直同步信号时序控制寄存器 (VSYNCTIM)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	VSYNC_START[9:0]									
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/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
	—	—	—	—	—	—	VSYNC_END[9:0]									
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R/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 ~ 26	—	全 0	R	保留位 读取值为 0, 只能写 0。
25 ~ 16	VSYNC_START [9:0]	H'000	R/W	设定内部垂直同步信号到屏幕垂直同步信号 (VSYNC) 被置 1 位置的 间隔 [行数]。
15 ~ 10	—	全 0	R	保留位 读取值为 0, 只能写 0。
9 ~ 0	VSYNC_END [9:0]	H'001	R/W	设定内部垂直同步信号到屏幕垂直同步信号 (VSYNC) 被清 0 位置的 间隔 [行数]。

【注】 必须设定 VSYNC_START ≠ VSYNC_END, 否则不保证运行。

24.6.19 水平同步信号时序控制寄存器 (HSYNTIM)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	HSYNC_START[10:0]										
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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	HSYNC_END[10:0]										
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	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
31 ~ 27	—	全 0	R	保留位 读取值为 0，只能写 0。
26 ~ 16	HSYNC_START [10:0]	H'000	R/W	此位总是被置 H'000。设定内部水平同步信号到屏幕水平同步信号 (HSYNC) 被置 1 位置的间隔 [显示屏的时钟周期数]。
15 ~ 11	—	全 0	R	保留位 读取值为 0，只能写 0。
10 ~ 0	HSYNC_END [10:0]	H'00A	R/W	设定内部水平同步信号到屏幕水平同步信号 (HSYNC) 被清 0 位置的间隔 [显示屏的时钟周期数]。

【注】 必须设定为 HSYNC_START≠HSYNC_END，否则不保证运行。

24.6.20 栅极时钟信号时序控制寄存器 (CLSTIM)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	CLS_START[10:0]										
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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	CLS_END[10:0]										
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
31 ~ 27	—	全 0	R	保留位 读取值为 0，只能写 0。
26 ~ 16	CLS_START[10:0]	H'000	R/W	设定内部水平同步信号到栅极时钟信号 (CLS) 被置 1 位置的间隔 [显示屏的时钟周期数]。
15 ~ 11	—	全 0	R	保留位 读取值为 0，只能写 0。
10 ~ 0	CLS_END[10:0]	H'000	R/W	设定内部水平同步信号到栅极时钟信号 (CLS) 被清 0 位置的间隔 [显示屏的时钟周期数]。

【注】 必须设定为 CLS_START≠CLS_END，否则不保证运行。

24.6.21 采样起始信号时序控制寄存器 (SPLTIM)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SPL_START[10:0]										
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:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SPL_END[10:0]										
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
31 ~ 27	—	全 0	R	保留位 读取值为 0，只能写 0。
26 ~ 16	SPL_START[10:0]	H'000	R/W	设定内部水平同步信号到采样起始信号 (SPL) 被置 1 位置的间隔 [显示屏的时钟周期数]。
15 ~ 11	—	全 0	R	保留位 读取值为 0，只能写 0。
10 ~ 0	SPL_END[10:0]	H'000	R/W	设定内部水平同步信号到采样起始信号 (SPL) 被清 0 位置的间隔 [显示屏的时钟周期数]。

【注】 必须设定为 SPL_START≠SPL_END，否则不保证运行。

24.6.22 栅极控制信号时序控制寄存器 (COMTIM)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	COM_MODE	—	—	—	—	—	COMTIM_V[9:0]									
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/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
	—	—	—	—	—	COMTIM_H[10:0]										
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
31	COM_MODE	0	R/W	选择栅极控制信号 (COM) 的交替模式。 0: 在每 1 行都进行 H→L→H→ 的反相输出。 以帧为单位进行相位的反相 (如果 n 帧是: H→L→H→..., 第 n+1 帧就是 L→H→L→...)。 1: 每 1 帧都进行反相
30 ~ 26	—	全 0	R	保留位 读取值为 0, 只能写 0。
25 ~ 16	COMTIM_V[9:0]	H'000	R/W	设定内部垂直同步信号到栅极控制信号 (COM) 的帧起始位置的间隔 [行数]。在设定 0 时, 第 1 行是帧位置; 在设定 1 时, 第 2 行是帧位置。
15 ~ 11	—	全 0	R	保留位 读取值为 0, 只能写 0。
10 ~ 0	COMTIM_H[10:0]	H'000	R/W	设定内部水平同步信号到栅极控制信号 (COM) 水平方向的交替位置的间隔 [显示屏的时钟周期数]。

【注】 必须设定为 COMTIM_V < SYN_HEIGHT, 否则不保证运行。

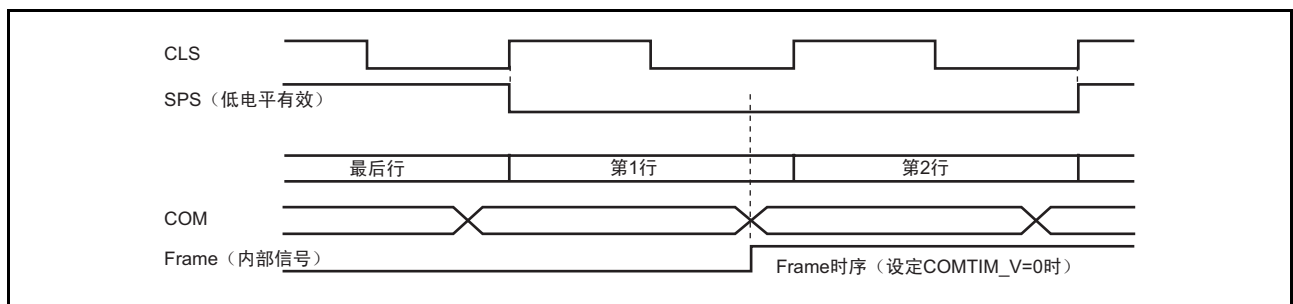


图 24.14 COM 信号的补充说明图

24.6.23 SGDE 区起始位置寄存器 (SGDESTART)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SGDE_START_V[9:0]									
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/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
	—	—	—	—	—	SGDE_START_H[10:0]										
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
31 ~ 26	—	全 0	R	保留位 读取值为 0，只能写 0。
25 ~ 16	SGDE_START_V [9:0]	H'000	R/W	设定内部垂直同步信号到 DE（数据允许信号）垂直方向的输出起始位置的间隔 [行数]。 禁止设定 0。
15 ~ 11	—	全 0	R	保留位 读取值为 0，只能写 0。
10 ~ 0	SGDE_START_H [10:0]	H'000	R/W	设定内部水平同步信号到 DE（数据允许信号）水平方向的输出起始位置的间隔 [显示屏时钟周期数]。

- 【注】 1. 必须设定为 $SYN_HEIGHT > SGDE_HEIGHT + SGDE_START_V$ ，否则不保证运行。
2. 必须设定为 $SYN_WIDTH > SGDE_WIDTH + SGDE_START_H$ ，否则不保证运行。

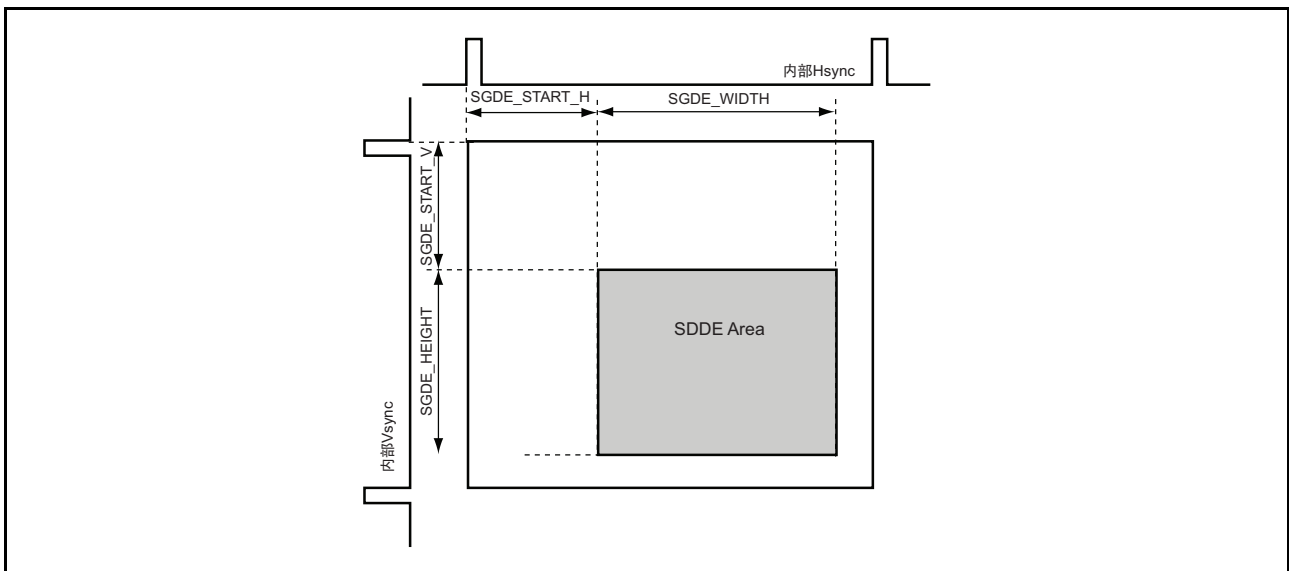


图 24.15 SG 部生成的有关 DE 区设定的补充图

24.6.24 SGDE 区大小寄存器 (SGDESIZE)

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SGDE_HEIGHT[9:0]									
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/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
	—	—	—	—	—	SGDE_WIDTH[10:0]										
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
31 ~ 26	—	全 0	R	保留位 读取值为 0，只能写 0。
25 ~ 16	SGDE_HEIGHT [9:0]	H'000	R/W	设定 DE（数据允许信号）垂直方向的高度 [行数]。
15 ~ 11	—	全 0	R	保留位 读取值为 0，只能写 0。
10 ~ 0	SGDE_WIDTH [10:0]	H'000	R/W	设定 DE（数据允许信号）水平方向的宽度 [显示屏的时钟周期数]。

24.6.25 CDE 色度颜色指定寄存器 (CDECRKY)

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
	CDE_R[4:0]					CDE_G[5:0]					CDE_B[4:0]					
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 16	—	全 0	R	保留位 读取值为 0，只能写 0。
15 ~ 11	CDE_R[4:0]	00000	R/W	CDE 信号的色键对象 R 值。
10 ~ 5	CDE_G[5:0]	000000	R/W	CDE 信号的色键对象 G 值。
4 ~ 0	CDE_B[4:0]	00000	R/W	CDE 信号的色键对象 B 值。

【注】 将图形图像重叠处理后的（层 1+ 层 2+ 层 3+ 层 4）的图像数据和上述的指定色进行比较，生成 CDE 信号（并非进行各层单位的处理。）。

24.6.26 T1004 控制寄存器 (T1004CNT)

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

位	位名	初始值	R/W	说明
31 ~ 3	—	全 0	R	保留位 读取值为 0，只能写 0。
2	VSYNC_TYPE	0	R/W	T-1004 的 VSYNC 极性选择位 0: 正极性 1: 负极性
1	HSYNC_TYPE	0	R/W	T-1004 的 HSYNC 极性选择位 0: 正极性 1: 负极性
0	DEC_TYPE	0	R/W	T-1004 的 DEC (数据允许信号) 极性选择位 0: 正极性 1: 负极性

24.6.27 T1004 视频起始位置寄存器 (T1004OFFSET)

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
	—	—	—	—	—	T1004OFFSET_H[10:0]										
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
31 ~ 11	—	全 0	R	保留位 读取值为 0，只能写 0。
10 ~ 0	T1004OFFSET_H [10:0]	0	R/W	以 2 像素为单位，调整视频信号和消隐期间的水平相位。 在增大设定值时，图像向左移动。最低的 2 位 (bit1、0) 必须固定为 0。

通过表 24.10 的寄存器设定，能在 Active Area 的左上角显示视频。

表 24.10 T-1004 输出的寄存器设定例子

寄存器名	寄存器值	设定内容	
图形部	GRODPHV1、2、3、4	H'0026_0072	对于内部 Vsync, 从 40[行] 输出视频。 对于内部 Hsync, 从 131[显示屏的时钟周期] 输出视频。
显示控制部	SYNSIZE	H'020D_35A	垂直同步信号周期 525[行] 水平同步信号周期 858[显示屏的时钟周期]
	T1004OFFSET	H'0000_0010	调整图像信号和消隐期间的水平相位。 值每增加 H'4, 图像就向左移动 2 个像素。 值每减少 H'4, 图像就向右移动 2 个像素。

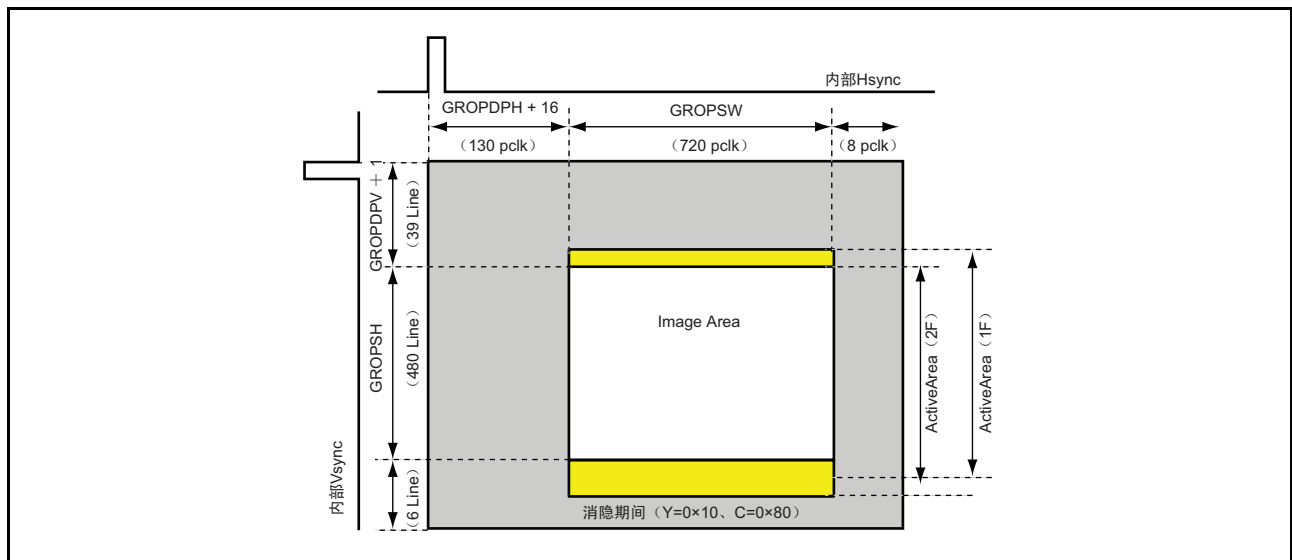


图 24.16 T-1004 输出的视频输出位置

24.7 运行步骤

24.7.1 显示控制部

1. 将寄存器值传送允许位设定为不允许。
必须将SG模式设定寄存器的WE位置0。
2. 设定显示控制部的寄存器。
给表24.9所示的寄存器设定适当的寄存器值。
必须先设定外部引脚的极性。
3. 将寄存器值传送允许位设定为允许。
必须将SG模式设定寄存器的WE位置1。

24.7.2 图形部

1. 将寄存器值传送允许位设定为不允许。
必须将图形块控制寄存器的WE位置0。
必须将 α 控制寄存器的WE位置0。
2. 设定图形部的寄存器
给表24.5~表24.8所示的寄存器设定适当的寄存器值。
3. 将寄存器值传送允许位设定为允许。
必须将图形块控制寄存器的WE位置1。
必须将 α 控制寄存器的WE位置1。
从下一个帧画面开始进行寄存器设定的显示运行。

第 25 章 NAND 闪存控制器 (FLCTL)

NAND 闪存控制器 (FLCTL) 提供 NAND 型闪存的存储器接口, 为了对应闪存特有的读错误, FLCTL 具有 ECC 符号的生成功能和错误检测功能。

【注】 本 LSI 不支持多层单元闪存 (MLC)。

25.1 特点

(1) NAND 型闪存的存储器接口

- 能直接连接 NAND 型闪存的接口
- 以扇区为单元 (512+16 字节) 进行读写, 并且进行 ECC 处理。
在 NAND 型闪存的数据表中, 有将 2048+64 字节的存取单位记为 1 页的产品, 在本书中统一将 512+16 字节记为 1 个扇区。
- 进行以字节为单位的读写。

(2) 存取模式: FLCTL 能选择以下 2 种存取模式。

- 命令存取模式
通过给寄存器指定本 FLCTL 对闪存发行的命令、地址、输入/输出的数据长度, 进行一系列的存取, 实现不进行 ECC 处理的数据的读写和擦除。
- 扇区存取模式
通过指定物理扇区进行以物理扇区为单元的读写, 并且进行 ECC 符号的生成和检查的控制。通过指定扇区数, 能读写连续的物理扇区。

(3) 扇区和管理码

- 1 个扇区由 512 字节的数据和 16 字节的管理码构成。管理码中有 8 字节的 ECC。
- 能以 4 字节单位指定管理码中的 ECC 位置。
- 能给管理码中的 ECC 以外的位置写用户信息。

(4) ECC

- 对于扇区 (数据: 512 字节+管理码:16 字节), 生成 8 字节的 ECC 码并进行错误检查 (但是, 16 字节管理码中的 ECC 的生成和错误检查对象的字节数因设定而不同。)
- 纠错能力: 能修正任意的 3 个错误。
- 在写操作时, 数据和 ECC 前的管理码为 ECC 符号生成对象。ECC 后的管理码不是 ECC 对象。
- 在读操作时, 数据和 ECC 前的管理码为 ECC 错误判断对象。另外, FIFO 内的管理码中的 ECC 不是从闪存读取的 ECC 码, 而是 ECC 电路的判断结果。
- 在发生 ECC 错误时不进行纠错, 必须通过软件进行处理。

(5) 数据错误时

- 在发生编程/擦除错误时, 反映到错误源标志。能指定独立源的中断。
- 在发生读错误时, 管理码内的 ECC 不为 0, 并且反映到 ECC 错误源标志。
- 在发生 ECC 错误时, 进行纠错, 并且指定代替扇区, 根据需要将该块的内容复制到代替扇区。

(6) 数据传送的 FIFO 和数据寄存器

- 内置 224 字节的 FLDTFIFO，用于传送闪存数据。
- 内置 32 字节的 FLECFIFO，用于传送管理码的数据。
- 在从 CPU 和 DMA 进行存取时，有超限/欠载检测标志位。

(7) DMA 传送

- 通过给 DMA 控制器分别指定闪存数据和管理码的传送目标，能将数据和管理码传送到不同的区域。

(8) 存取长度

- 有 32 位存取的寄存器和 8 位存取的寄存器。必须指定存取长度进行读写。

(9) 存取时间

- 本 FLCTL 引脚的工作频率和外围总线的工作频率不同，能通过通用控制寄存器 (FLCMNCR) 的 FCKSEL 位和 QTSEL 位进行指定。
- 在更改 CPG 的设定时，必须在使 FLCTL 进入模块停止的状态下进行。
- 对于 NAND 型闪存， $\overline{\text{FRE}}$ 引脚和 $\overline{\text{FWE}}$ 引脚是以 CPG 指定的引脚工作频率运行的。为了确保准备时间，不能超过所连接存储器的最大工作频率。
- NAND 型闪存引脚的工作时钟 FCLK 使用外围总线的工作时钟 Pck 分频后的时钟。

FLCTL 的框图如图 25.1 所示。

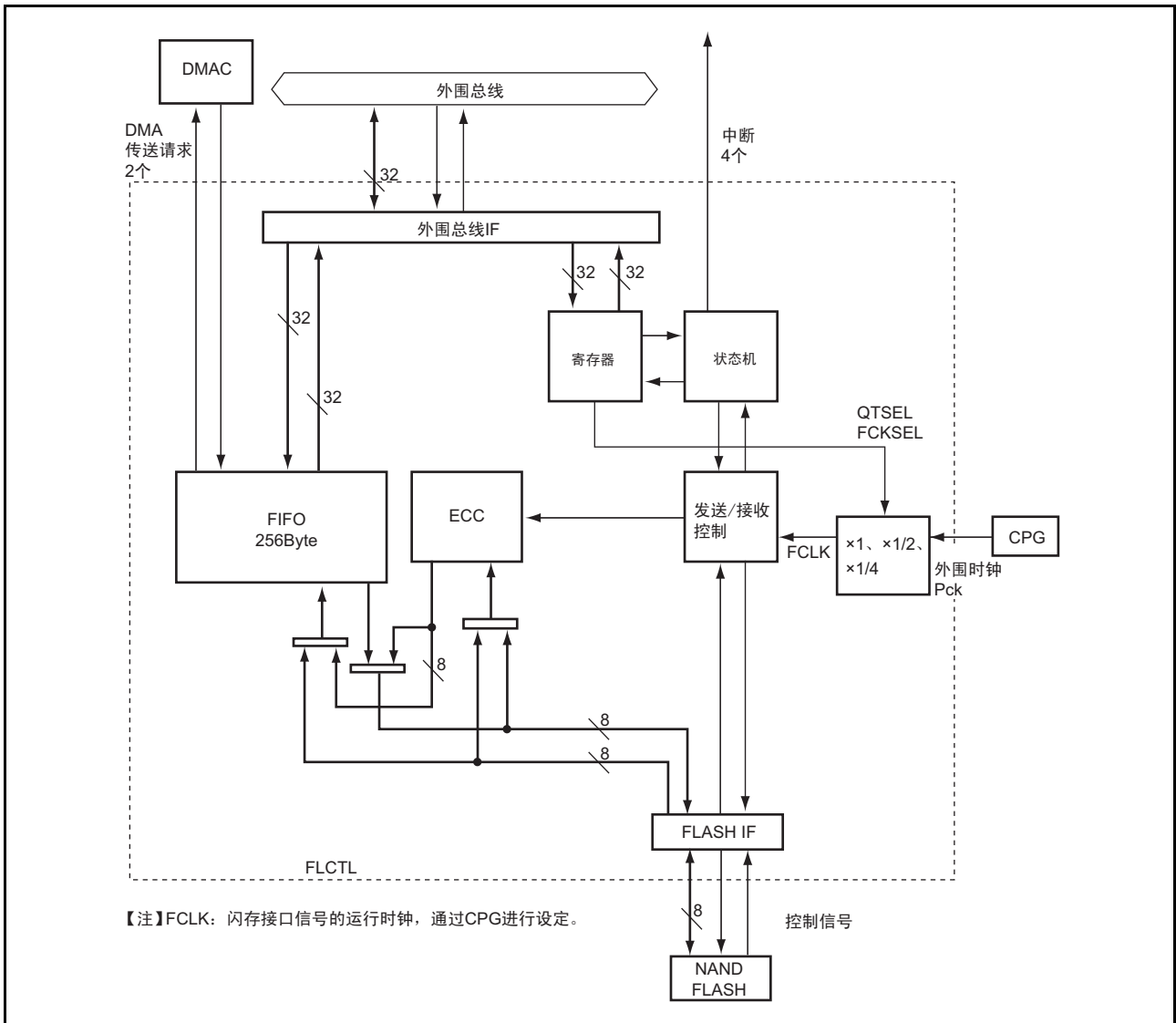


图 25.1 FLCTL 的框图

25.2 输入 / 输出引脚

FLCTL 的引脚结构如表 25.1 所示。

表 25.1 引脚结构

引脚名	功能	输入 / 输出	对应的闪存引脚 (NAND 型)	说明
$\overline{\text{FCE}}$	芯片允许	输出	$\overline{\text{CE}}$	允许本 LSI 连接的闪存。
FD7 ~ FD0	数据输入 / 输出	输入 / 输出	I/O7 ~ I/O0	是命令、地址和数据的输入 / 输出引脚。
FCLE	命令数据允许	输出	CLE	命令锁定允许 (CLE) 在输出命令时为有效电平。
FALE	地址锁定允许	输出	ALE	地址锁定允许 (ALE) 在输出地址时为有效电平。 在输入 / 输出数据时为无效电平。
$\overline{\text{FRE}}$	读允许	输出	$\overline{\text{RE}}$	读允许 ($\overline{\text{RE}}$) 在 $\overline{\text{RE}}$ 的下降沿读数据。
$\overline{\text{FWE}}$	写允许	输出	$\overline{\text{WE}}$	写允许 在 $\overline{\text{WE}}$ 的上升沿, 闪存锁定命令、地址和数据。
FR/B	就绪 / 忙	输入	R/B	就绪 / 忙 高电平表示就绪状态, 低电平表示忙状态。

25.3 寄存器说明

FLCTL 的寄存器结构如表 25.2 所示，各处理模式的寄存器状态如表 25.3 所示。

表 25.2 寄存器结构

寄存器名	略称	R/W	P4 区地址	区域 7 地址	存取长度
共用控制寄存器	FLCMNCR	R/W	H'FFE9 0000	H'1FE9 0000	32
命令控制寄存器	FLCMDCR	R/W	H'FFE9 0004	H'1FE9 0004	32
命令码寄存器	FLCMCDR	R/W	H'FFE9 0008	H'1FE9 0008	32
地址寄存器	FLADR	R/W	H'FFE9 000C	H'1FE9 000C	32
地址寄存器 2	FLADR2	R/W	H'FFE9 003C	H'1FE9 003C	32
数据寄存器	FLDATAR	R/W	H'FFE9 0010	H'1FE9 0010	32
数据计数寄存器	FLDTCNTR	R/W	H'FFE9 0014	H'1FE9 0014	32
中断 DMA 控制寄存器	FLINTDMACR	R/W	H'FFE9 0018	H'1FE9 0018	32
就绪 / 忙超时设定寄存器	FLBSYTMR	R/W	H'FFE9 001C	H'1FE9 001C	32
就绪 / 忙超时计数器	FLBSYCNT	R	H'FFE9 0020	H'1FE9 0020	32
数据 FIFO 寄存器	FLDTFIFO	R/W	H'FFE9 0024/ H'FFE9 0050	H'1FE9 0024/ H'1FE9 0050	32
管理码 FIFO 寄存器	FLECFIFO	R/W	H'FFE9 0028/ H'FFE9 0060	H'1FE9 0028/ H'1FE9 0060	32
传送控制寄存器	FLTRCR	R/W	H'FFE9 002C	H'1FE9 002C	8

表 25.3 各处理模式的寄存器状态

寄存器略称	上电复位	待机	模块待机	睡眠
FLCMNCR	初始化	保持	保持	保持
FLCMDCR	初始化	保持	保持	保持
FLCMCDR	初始化	保持	保持	保持
FLADR	初始化	保持	保持	保持
FLADR2	初始化	保持	保持	保持
FLDATAR	初始化	保持	保持	保持
FLDTCNTR	初始化	保持	保持	保持
FLINTDMACR	初始化	保持	保持	保持
FLBSYTMR	初始化	保持	保持	保持
FLBSYCNT	初始化	保持	保持	保持
FLDTFIFO	初始化	保持	保持	保持
FLECFIFO	初始化	保持	保持	保持
FLTRCR	初始化	保持	保持	保持

25.3.1 共用控制寄存器 (FLCMNCR)

FLCMNCR 是 32 位可读写寄存器，指定闪存 (NAND) 和存取模式等以及指定 $\overline{\text{FCE}}$ 引脚的输出。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	SNAND	QTSEL	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FCKSEL	—	ECCPOS[1:0]	ACM[1:0]	NANDWF	—	—	—	—	—	—	CE0	—	—	TYPESEL	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R	R	R/W

位	位名	初始值	R/W	说明
31 ~ 19	—	全 0	R	保留位 读取值为 0，只能写 0。
18	SNAND	0	R/W	大容量 NAND 闪存选择位 用于指定 2048+64 字节页结构的 1G 位以上的 NAND 闪存。 0: 使用 512+16 字节页结构的闪存 1: 使用 2048+64 字节页结构的 NADA 型闪存 【注】 在 TYPESEL=0 时，禁止设定 1。
17	QTSEL	0	R/W	闪存时钟 4 分频选择位 是闪存内使用的时钟 FCLK 的分频选择位。和 FCKSEL 一起使用。 QTSEL=0、FCKSEL=0: 将 CPG 的时钟 (Pck) 的 1/2 分频后的时钟用作 FCLK。 QTSEL=0、FCKSEL=1: 将 CPG 的时钟 (Pck) 用作 FCLK。 QTSEL=1、FCKSEL=0: 将 CPG 的时钟 (Pck) 的 1/4 分频后的时钟用作 FCLK。 QTSEL=1、FCKSEL=1: 禁止设定
16	—	0	R	1 保留位 读取值为 0，只能写 0。
15	FCKSEL	0	R/W	闪存时钟选择位 是闪存内使用的时钟 FCLK 的分频选择位。 和 QTSEL 一起使用。请参照 QTSEL 的说明部分。
14	—	0	R	保留位 读取值为 0，只能写 0。
13、12	ECCPOS[1:0]	00	R/W	ECC 位置指定位 1、0 指定管理码区内的 ECC 位置 (第 0/4/8 字节)。 00: 在管理码区的第 0 ~ 7 字节存放 ECC 01: 在管理码区的第 4 ~ 11 字节存放 ECC 10: 在管理码区的第 8 ~ 15 字节存放 ECC 11: 禁止设定
11、10	ACM[1:0]	00	R/W	存取模式指定位 1、0 指定存取模式。 00: 命令存取模式 01: 扇区存取模式 10: 禁止设定 11: 禁止设定

位	位名	初始值	R/W	说明
9	NANDWF	0	R/W	NAND 等待插入位 0: 以 1 个 FCLK 周期, 进行地址和数据的输入 / 输出。 1: 以 2 个 FCLK 周期, 进行地址和数据的输入 / 输出。
8 ~ 4	—	全 0	R	保留位 读取值为 0, 只能写 0。
3	CE0	0	R/W	芯片允许位 0 0: 禁止 ($\overline{\text{FCE}}$ 引脚输出高电平。) 1: 允许 ($\overline{\text{FCE}}$ 引脚输出低电平。)
2、1	—	全 0	R	保留位 读取值为 0, 只能写 0。
0	TYPESEL	0	R/W	存储器选择位 0: 保留 1: 选择 NAND 型闪存。 在使用 FLCTL 时, 必须将此位置 1。

25.3.2 命令控制寄存器 (FLCMDCR)

FLCMDCR 是 32 位可读写寄存器, 在命令存取模式中, 能指定是否发行命令和地址, 并且能指定数据输入 / 输出的目标。在扇区存取模式中, 能指定扇区传送次数。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADR CNT2	SCTCNT[19:16]				ADR MD	CDS RC	DOSR	—	—	SEL RW	DOA DR	ADRCNT[1:0]		DOC MD2	DOC MD1
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCTCNT[15:0]															
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31	ADRCNT2	0	R/W	地址发行字节数指定位 指定地址阶段的发行地址数据的字节数。和 ADRCNT[1:0] 位一起使用。 0: 只发行 ADRCNT[1:0] 位指定的字节数的地址。 1: 发行 5 字节的地址 ADRCNT[1:0] 位必须置 00。
30 ~ 27	SCTCNT[19:16]	全 0	R/W	扇区传送次数指定位 [19:16] 是扇区传送次数指定位 SCTCNT[15:0] 位的扩展位。 SCTCNT[19:16] 和 SCTCNT[15:0] 合在一起, 作为 SCTCNT[19:0] 的 20 位计数器运行。
26	ADRMD	0	R/W	扇区存取地址指定位 此位在命令存取模式时无效, 只在扇区存取模式时有效。 0: 地址寄存器的值作为物理扇区号进行处理。 通常, 在存取扇区时使用此值。 1: 地址寄存器的值作为闪存地址进行输出。 【注】 在存取连续的扇区时, 必须置 0。

位	位名	初始值	R/W	说明
25	CDSRC	0	R/W	数据缓冲器指定位 在命令存取模式时, 指定数据阶段的读写数据缓冲器。 0: 将 FLDATAR 指定为数据缓冲器。 1: 将 FLDTFIFO 指定为数据缓冲器。
24	DOSR	0	R/W	读状态选择位 在命令存取模式时, 指定在发行第 2 个命令后是否读状态。 0: 不读状态 1: 读状态
23、22	—	全 0	R	保留位 读取值为 0, 只能写 0。
21	SELRW	0	R/W	数据读写指定位 指定数据阶段的读写方向。 0: 读 1: 写
20	DOADR	0	R/W	地址阶段执行指定位 在命令存取模式时, 指定是否执行地址阶段。 0: 不执行地址阶段 1: 执行地址阶段
19、18	ADRCNT[1:0]	00	R/W	地址发行字节数指定位 指定地址阶段的发行地址数据的字节数。 00: 发行 1 字节的地址 01: 发行 2 字节的地址 10: 发行 3 字节的地址 11: 发行 4 字节的地址
17	DOCMD2	0	R/W	第 2 个命令阶段执行指定位 在命令存取模式时, 指定是否执行第 2 个命令阶段。 0: 不执行第 2 个命令阶段 1: 执行第 2 个命令阶段
16	DOCMD1	0	R/W	第 1 个命令阶段执行指定位 在命令存取模式时, 指定是否执行第 1 个命令阶段。 0: 不执行第 1 个命令阶段 1: 执行第 1 个命令阶段
15 ~ 0	SCTCNT[15:0]	H'0000	R/W	扇区传送次数指定位 [15:0] 在扇区存取模式时, 指定要连续读取的扇区数。在每个扇区传送结束时进行递减计数, 当变为 0 时停止计数。 和 SCTCNT[19:16] 一起使用。 在命令存取模式时, 为 H'0 0001。

25.3.3 命令码寄存器 (FLCMCDR)

FLCMCDR 是 32 位可读写寄存器，指定在指令存取模式和扇区存取模式时发行的命令值。

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
	CMD[15:8]								CMD[7:0]							
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 16	—	全 0	R	保留位 读取值为 0，只能写 0。
15 ~ 8	CMD[15:8]	H'00	R/W	指定在第 2 个命令阶段发行的命令码。
7 ~ 0	CMD[7:0]	H'00	R/W	指定在第 1 个命令阶段发行的命令码。

25.3.4 地址寄存器 (FLADR)

FLADR 是 32 位可读写寄存器，指定在命令存取模式时的地址输出的值。在扇区存取模式时，将物理扇区地址位指定的物理扇区号转换为地址进行输出。

- 命令存取模式时

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

位	位名	初始值	R/W	说明
31 ~ 24	ADR[31:24]	H'00	R/W	第 4 个地址数据位 在命令存取模式时，将输出给闪存的第 4 个数据指定为地址。
23 ~ 16	ADR[23:16]	H'00	R/W	第 3 个地址数据位 在命令存取模式时，将输出给闪存的第 3 个数据指定为地址。
15 ~ 8	ADR[15:8]	H'00	R/W	第 2 个地址数据位 在命令存取模式时，将输出给闪存的第 2 个数据指定为地址。
7 ~ 0	ADR[7:0]	H'00	R/W	第 1 个地址数据位 在命令存取模式时，将输出给闪存的第 1 个数据指定为地址。

- 扇区存取模式时

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	ADR[25:16]									
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/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
	ADR[15:0]															
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 26	—	全 0	R	保留位 读取值为 0，只能写 0。
25 ~ 0	ADR[25:0]	H'0000 0000	R/W	物理扇区地址位 在扇区存取模式时，指定要存取的物理扇区号。 将物理扇区号转换为地址，并且输出到闪存。在 FLCMDCR 的 ADRCNT2 位为 1 时，ADR[25:0] 有效；在 ADRCNT2 位为 0 时，ADR[17:0] 有效。

25.3.5 地址寄存器 2 (FLADR2)

FLADR2 是 32 位可读写寄存器，在 FLCMDCR 的 ADRCNT2 位为 1 时有效。指定在命令存取模式中的地址输出的值。

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
	—	—	—	—	—	—	—	ADR[7:0]								
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 8	—	全 0	R	保留位 读取值为 0，只能写 0。
7 ~ 0	ADR[7:0]	全 0	R/W	第 5 个地址数据位 在命令存取模式时，将输出给闪存的第 5 个数据指定为地址。

25.3.6 数据计数寄存器 (FLDTCNTR)

FLDTCNTR 是 32 位可读写寄存器。指定在命令存取模式时的读写字节数。

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

位	位名	初始值	R/W	说明
31 ~ 24	ECFLW[7:0]	H'00	R	FLECFIFO 存取个数位 表示 FLECFIFO 能读写的长字数。 能用于通过 CPU 进行的 FLECFIFO 读写。 在读 FLECFIFO 时，表示 FLECFIFO 内能读的数据的长字数。 在写 FLECFIFO 时，表示 FLECFIFO 内能写的空的长字数。
23 ~ 16	DTFLW[7:0]	H'00	R	FLDTFIFO 存取个数位 表示 FLDTFIFO 能读写的长字数。 能用于通过 CPU 进行的 FLDTFIFO 读写。 在读 FLDTFIFO 时，表示 FLDTFIFO 内能读的数据的长字数。 在写 FLDTFIFO 时，表示 FLDTFIFO 内能写的空的长字数。
15 ~ 12	—	全 0	R	保留位 读取值为 0，只能写 0。
11 ~ 0	DTCNT[11:0]	H'000	R/W	数据个数指定位 在命令存取模式时，指定读写数据的字节数（最大能指定 2048+64 字节）。

25.3.7 数据寄存器 (FLDATAR)

FLDATAR 是 32 位可读写寄存器。在命令存取模式中，用于保存 FLCMDCR 的 CDSRC 位为 0 时的输入 / 输出数据。

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

位	位名	初始值	R/W	说明
31 ~ 24	DT[31:24]	H'00	R/W	第 4 个数据位 指定从 FD7 ~ FD0 第 4 个输入 / 输出的数据。 在写操作时：指定写数据。 在读操作时：保存读数据。
23 ~ 16	DT[23:16]	H'00	R/W	第 3 个数据位 指定从 FD7 ~ FD0 第 3 个输入 / 输出的数据。 在写操作时：指定写数据。 在读操作时：保存读数据。
15 ~ 8	DT[15:8]	H'00	R/W	第 2 个数据位 指定从 FD7 ~ FD0 第 2 个输入 / 输出的数据。 在写操作时：指定写数据。 在读操作时：保存读数据。
7 ~ 0	DT[7:0]	H'00	R/W	第 1 个数据位 指定从 FD7 ~ FD0 第 1 个输入 / 输出的数据。 在写操作时：指定写数据。 在读操作时：保存读数据。

25.3.8 中断 DMA 控制寄存器 (FLINTDMACR)

FLINTDMACR 是 32 位可读写寄存器，设定是允许还是禁止 DMA 传送请求和中断。FLCTL 在各存取模式运行开始后，对 DMAC 产生传送请求。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	—	—	—	—	—	—	—	ECER INTE	—	—	FIFOTRG [1:0]	AC1 CLR	AC0 CLR	DREQ1 EN	DREQ0 EN		
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	—	—	—	—	—	—	—	EC ERB	ST ERB	BTO ERB	TRR EQF1	TRR EQF0	STER INTE	RBER INTE	TE INTE	TR INTE1	TR INTE0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

位	位名	初始值	R/W	说明
31 ~ 25	—	全 0	R	保留位 读取值为 0，只能写 0。
24	ECERINTE	0	R/W	ECC 错误中断允许位 0: 在发生 ECC 错误时，禁止中断 1: 在发生 ECC 错误时，允许中断
23、22	—	全 0	R	保留位 读取值为 0，只能写 0。
21、20	FIFOTRG[1:0]	00	R/W	FIFO 触发设定位 切换 FIFO 传送请求的发生条件。 在读闪存时 00: 在 FLDTFIFO 存有 4 字节的数据时，向 CPU 中断或者 DMA 传送。 01: 在 FLDTFIFO 存有 16 字节的数据时，向 CPU 请求中断或者 DMA 传送。 10: 在 FLDTFIFO 存有 128 字节的数据时，向 CPU 请求中断或者 DMA 传送。 11: 在 FLDTFIFO 存有 128 字节的数据或存放 16 字节时，向 CPU 请求中断或者 DMA 传送。 在进行闪存编程时 00: 在 FLDTFIFO 有 4 字节以上的空间时，向 CPU 请求中断（不能设定 DMA 传送）。 01: 在 FLDTFIFO 有 16 字节以上的空间时，向 CPU 请求中断或者 DMA 传送。 10: 在 FLDTFIFO 有 128 字节以上的空间时，向 CPU 请求中断（不能设定 DMA 传送）。 11: 在 FLDTFIFO 有 128 字节以上的空间时，向 CPU 请求中断； 在 FLDTFIFO 有 16 字节以上的空间时，向 CPU 请求 DMA 传送。
19	AC1CLR	0	R/W	FLECFIFO 清除位 清除 FLECFIFO。 0: 保持 FLECFIFO 的值。在存取闪存时，必须置 0。 1: 清除 FLECFIFO。在清除后必须置 0。
18	AC0CLR	0	R/W	FLDTFIFO 清除位 清除 FLDTFIFO。 0: 保持 FLDTFIFO 的值。在存取闪存时，必须置 0。 1: 清除 FLDTFIFO 的值。在清除后必须置 0。

位	位名	初始值	R/W	说明
17	DREQ1EN	0	R/W	FLECFIFODMA 请求允许位 选择是允许还是禁止发行 FLECFIFO 区的 DMA 传送请求。 0: 禁止发行 FLECFIFO 区的 DMA 传送请求 1: 允许发行 FLECFIFO 区的 DMA 传送请求
16	DREQ0EN	0	R/W	FLDTFIFODMA 请求允许位 选择是允许还是禁止发行 FLDTFIFO 区的 DMA 传送请求。 0: 禁止发行 FLDTFIFO 区的 DMA 传送请求 1: 允许发行 FLDTFIFO 区的 DMA 传送请求
15 ~ 10	—	全 0	R	保留位 读取值为 0, 只能写 0。
9	ECERB	0	R/W	ECC 错误位 表示 ECC 的错误检查结果。如果在扇区存取模式中读闪存时发生 ECC 错误, 就将此位置 1。 即使此位置 1 也不产生中断。 此位为标志位, 不能写 1。为了清除标志, 只能写 0。 0: 无 ECC 错误 (表示取到的 ECC 全部为 0) 1: 发生 ECC 错误。
8	STERB	0	R/W	状态错误位 表示读状态的结果。如果在读状态时 FLBSYCNT 的 STAT[7:0] 位的特定定位为 1, 就将此位置 1。 此位为标志位, 不能写 1。为了清除标志, 只能写 0。 0: 无状态错误 (表示 FLBSYCNT 的 STAT[7:0] 位的特定定位为 0) 1: 发生状态错误。 有关特定定位, 请参照“25.4.6 状态的读操作”。
7	BTOERB	0	R/W	超时错误位 在发生超时错误时 (FLBSYCNT 的 RBTIMCNT[19:0] 位在递减计数后变为 0), 将此位置 1。 此位为标志位, 不能写 1。为了清除标志, 只能写 0。 0: 无超时错误 1: 发生超时错误
6	TRREQF1	0	R/W	FLECFIFO 传送请求标志位 表示发生了 FLECFIFO 的传送请求。 此位为标志位, 不能写 1。为了清除标志, 只能写 0。 0: 未发生 FLECFIFO 的传送请求 1: 发生 FLECFIFO 的传送请求
5	TRREQF0	0	R/W	FLDTFIFO 传送请求标志位 表示发生了 FLDTFIFO 的传送请求。 此位为标志位, 不能写 1。为了清除标志, 只能写 0。 0: 未发生 FLDTFIFO 的传送请求 1: 发生 FLDTFIFO 的传送请求
4	STERINTE	0	R/W	发生状态错误时的中断允许位 在发生状态错误时, 选择是允许还是禁止向 CPU 请求中断。 0: 在发生状态错误时, 禁止向 CPU 请求中断。 1: 在发生状态错误时, 允许向 CPU 请求中断。

位	位名	初始值	R/W	说明
3	RBERINTE	0	R/W	发生超时错误时的中断允许位 在发生超时错误时，选择是允许还是禁止向 CPU 请求中断。 0: 在发生超时错误时，禁止向 CPU 请求中断。 1: 在发生超时错误时，允许向 CPU 请求中断。
2	TEINTE	0	R/W	传送结束中断允许位 在传送结束时 (FLTRCR 的 TREND 位)，选择是允许还是禁止向 CPU 请求中断。 0: 在传送结束时，禁止向 CPU 请求中断。 1: 在传送结束时，允许向 CPU 请求中断。
1	TRINTE1	0	R/W	对 CPU 的 FLECFIFO 传送请求允许位 在发生 FLECFIFO 的传送请求时，选择是允许还是禁止向 CPU 的中断请求。 0: 在发生 FLECFIFO 的传送请求时，禁止向 CPU 请求中断。 1: 在发生 FLECFIFO 的传送请求时，允许向 CPU 请求中断。 当允许 DMA 传送时，必须将此位置 0。
0	TRINTE0	0	R/W	对 CPU 的 FLDTFIFO 传送请求允许位 在发生 FLDTFIFO 的传送请求时，选择是允许还是禁止向 CPU 请求中断。 0: 在发生 FLDTFIFO 的传送请求时，禁止向 CPU 请求中断。 1: 在发生 FLDTFIFO 的传送请求时，允许向 CPU 请求中断。 当允许 DMA 传送时，必须将此位置 0。

25.3.9 就绪 / 忙超时设定寄存器 (FLBSYTMR)

FLBSYTMR 是 32 位可读写寄存器，设定 $\overline{\text{FR}}/\overline{\text{B}}$ 引脚为忙状态时的超时时间。

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

位	位名	初始值	R/W	说明
31 ~ 20	—	全 0	R	保留位 读取值为 0，只能写 0。
19 ~ 0	RBTMOUT[19:0]	H'00000	R/W	就绪 / 忙超时位 设定忙状态的超时时间 (Pck 的时钟数)。在设定为 0 时，不发生超时。

25.3.10 就绪 / 忙超时计数器 (FLBSYCNT)

FLBSYCNT 是 32 位只读寄存器。

通过读状态，将读取的闪存状态保存到 STAT[7:0] 位。

如果 $\overline{\text{FR/B}}$ 引脚为忙状态，就将 FLBSYTMR 的 RBTMOUT[19:0] 位设定的超时时间复制到 RBTIMCNT[19:0] 位，开始递减计数。如果 RBTIMCNT[19:0] 位的值变为 0，就将 FLINTDMACR 的 BTOERB 位置 1，并且通知发生超时错误。此时，如果通过 FLINTDMACR 的 RBERINTE 位允许中断，就能产生 FLSTE 中断请求。

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

位	位名	初始值	R/W	说明
31 ~ 24	STAT[7:0]	H'00	R	表示闪存的状态读取值。
23 ~ 20	—	全 0	R	保留位 读取值为 0。
19 ~ 0	RBTIMCNT[19:0]	H'00000	R	就绪 / 忙超时计数器位 在 $\overline{\text{FR/B}}$ 引脚为忙状态时，将 FLBSYTMR 的 RBTMOUT[19:0] 位的设定值复制到此位。然后，在 $\overline{\text{FR/B}}$ 引脚为忙状态的期间，对此位的值进行递减计数，如果值变为 0，就发生超时错误。

25.3.11 数据 FIFO 寄存器 (FLDTFIFO)

FLDTFIFO 是用于读写数据 FIFO 区的寄存器。

在进行 DMA 传送时，必须给此寄存器指定数据传送目标（传送源）。

FLCMDCCR 的 SELRW 位指定的读写方向必须和此寄存器的读写方向相同。

在进行 16 字节 DMA 传送时，必须从 16 字节地址边界的地址存取 FLDTFIFO。

必须在通过 FINTDMACR 的 FIFO 清除位清除 FIFO 的值后使用此寄存器。

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

位	位名	初始值	R/W	说明
31 ~ 24	DTFO[31:24]	H'00	R/W	第 1 个数据位 指定从 FD7 ~ FD0 第 1 个输入 / 输出的数据。 在写操作时：指定写数据。 在读操作时：保存读数据。
23 ~ 16	DTFO[23:16]	H'00	R/W	第 2 个数据位 指定从 FD7 ~ FD0 第 2 个输入 / 输出的数据。 在写操作时：指定写数据。 在读操作时：保存读数据。
15 ~ 8	DTFO[15:8]	H'00	R/W	第 3 个数据位 指定从 FD7 ~ FD0 第 3 个输入 / 输出的数据。 在写操作时：指定写数据。 在读操作时：保存读数据。
7 ~ 0	DTFO[7:0]	H'00	R/W	第 4 个数据位 指定从 FD7 ~ FD0 第 4 个输入 / 输出的数据。 在写操作时：指定写数据。 在读操作时：保存读数据。

25.3.12 管理码 FIFO 寄存器 (FLECFIFO)

FLECFIFO 是用于读写管理码 FIFO 区的寄存器。

在进行 DMA 传送时，必须给此寄存器指定管理码的传送目标（传送源）。

FLCMDCR 的 SELRW 位指定的读写方向必须和此寄存器的读写方向相同。

在进行 16 字节 DMA 传送时，必须从 16 字节地址边界的地址存取 FLECFIFO。

必须在通过 FINTDMACR 的 FIFO 清除位清除 FIFO 的值后使用此寄存器。

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

位	位名	初始值	R/W	说明
31 ~ 24	ECFO[31:24]	H'00	R/W	第 1 个数据位 指定从 FD7 ~ FD0 第 1 个输入 / 输出的数据。 在写操作时：指定写数据。 在读操作时：保存读数据。
23 ~ 16	ECFO[23:16]	H'00	R/W	第 2 个数据位 指定从 FD7 ~ FD0 第 2 个输入 / 输出的数据。 在写操作时：指定写数据。 在读操作时：保存读数据。
15 ~ 8	ECFO[15:8]	H'00	R/W	第 3 个数据位 指定从 FD7 ~ FD0 第 3 个输入 / 输出的数据。 在写操作时：指定写数据。 在读操作时：保存读数据。
7 ~ 0	ECFO[7:0]	H'00	R/W	第 4 个数据位 指定从 FD7 ~ FD0 第 4 个输入 / 输出的数据。 在写操作时：指定写数据。 在读操作时：保存读数据。

25.3.13 传送控制寄存器 (FLTRCR)

通过将 TRSTRT 位置 1，开始存取闪存。能通过 TREND 位确认存取是否结束。不能在传送中（从 TRSTRT 位被置 1 到 TREND 位被置 1 之间）强制结束（将 TRSTRT 位置 0）。

位:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TR END	TR STRT
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 读取值为 0，只能写 0。
1	TREND	0	R/W	处理结束标志位 表示指定的存取模式结束处理。 只能写 0。
0	TRSTRT	0	R/W	传送开始位 在 TREND 位为 0 时，如果将 TRSTRT 位从 0 变为 1，就开始存取模式指定位 ACM[1:0] 指定的存取模式的处理。 0: 停止传送 1: 开始传送

25.4 运行说明

25.4.1 运行模式

运行模式有 2 种，在扇区存取模式中进行 ECC 的生成 / 错误检查。

- 命令存取模式
- 扇区存取模式

25.4.2 寄存器的设定步骤

存取闪存时所需的寄存器设定流程如图 25.2 所示。

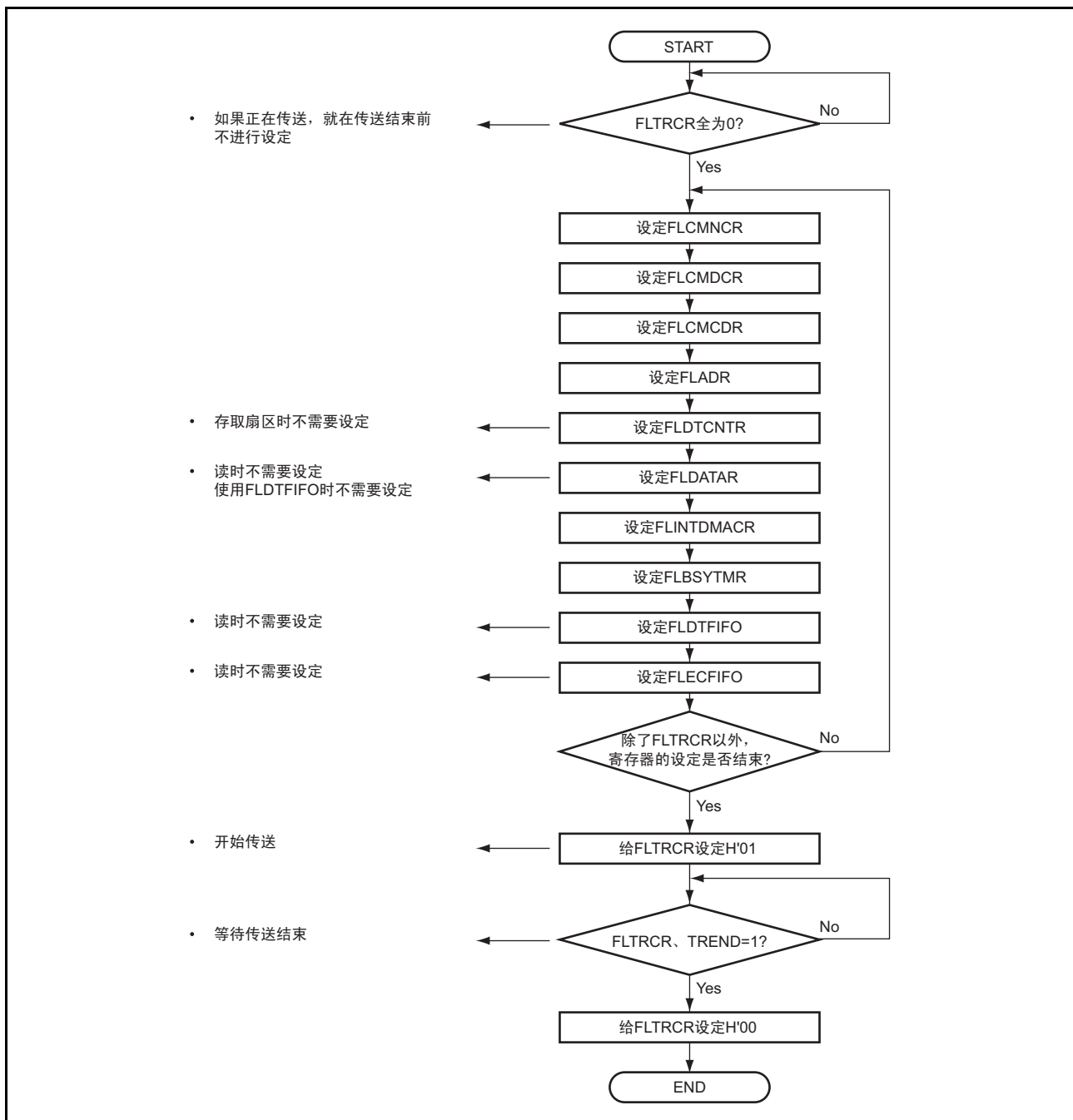


图 25.2 寄存器的设定流程

25.4.3 命令存取模式

命令存取模式是通过设定对闪存发行的命令、地址、读写方向和次数等进行闪存存取的模式。输入 / 输出数据能使用 FLDTFIFO 进行 DMA 传送。

(1) NAND 型闪存的存取

读 NAND 型闪存例子如图 25.3 所示。在此例中，将第 1 个命令指定为 H'00，地址长度指定为 3 字节。这是给数据计数器指定 8 字节的读字节数的情况。

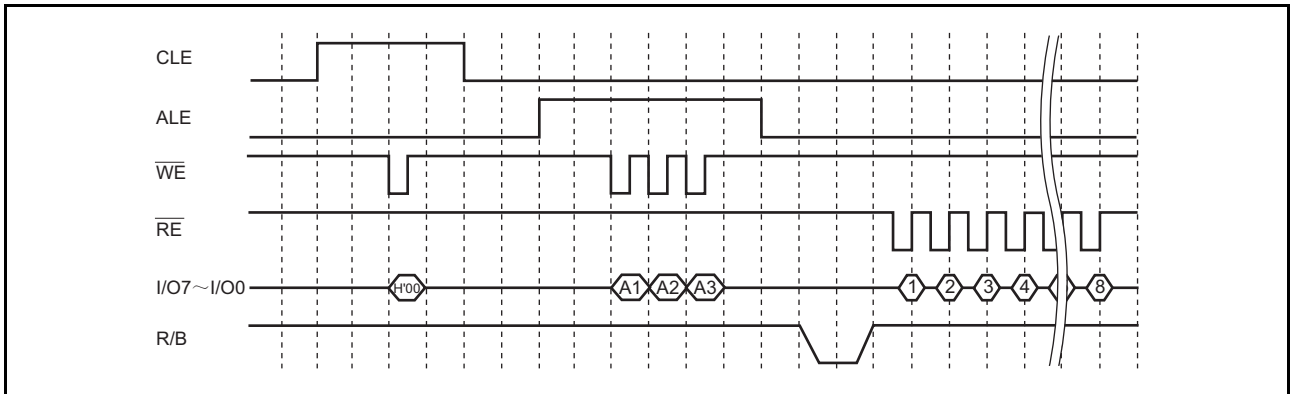


图 25.3 NAND 型闪存的读操作时序

进行 NAND 型闪存编程的波形如图 25.4 和图 25.5 所示。

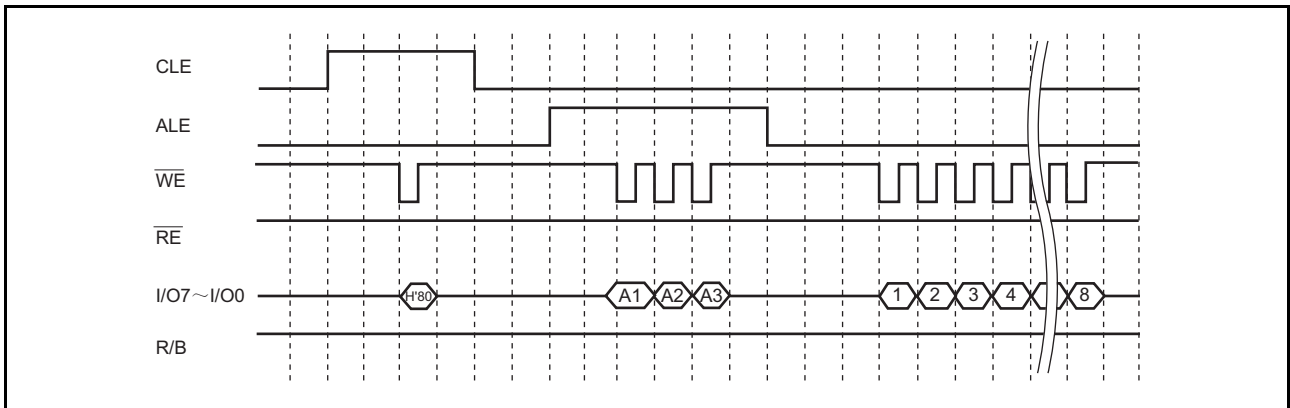


图 25.4 NAND 型闪存的编程时序 (1)

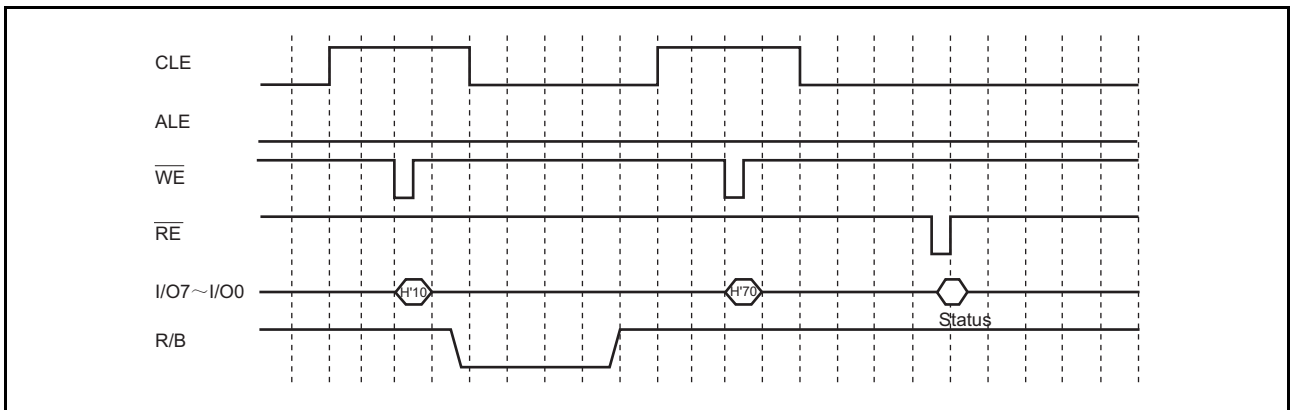


图 25.5 NAND 型闪存的编程时序 (2)

(2) NAND 型闪存 (2048+64 字节) 的存取

读 NAND 型闪存 (2048+64 字节) 的例子如图 25.6 所示。在此例中, 将第 1 个命令指定为 H'00, 第 2 个命令指定为 H'30, 地址长度指定为 4 字节。这是给数据计数器指定 4 字节的读字节数的情况。

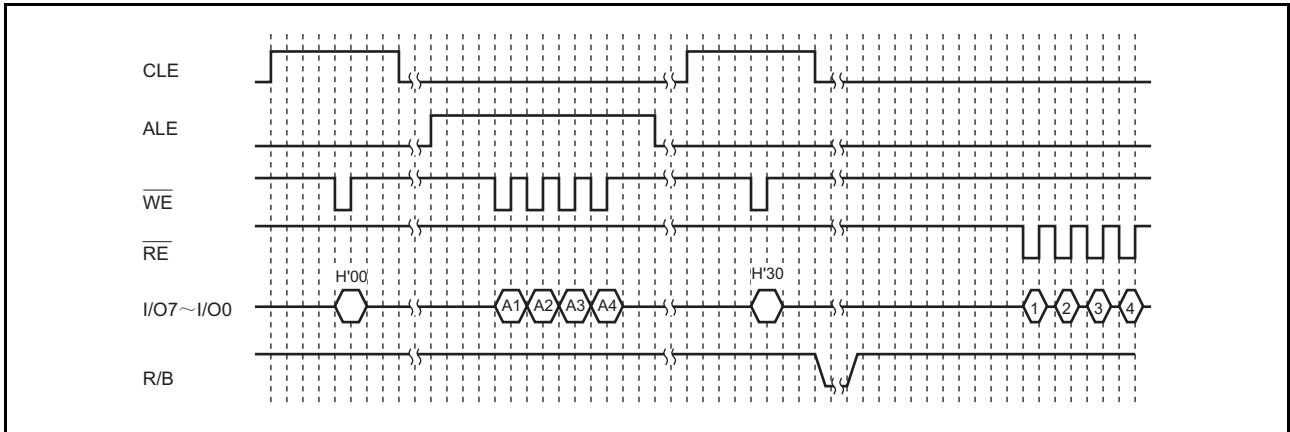


图 25.6 NAND 型闪存的读操作时序

进行 NAND 型闪存 (2048+64 字节) 编程的波形如图 25.7 和图 25.8 所示。

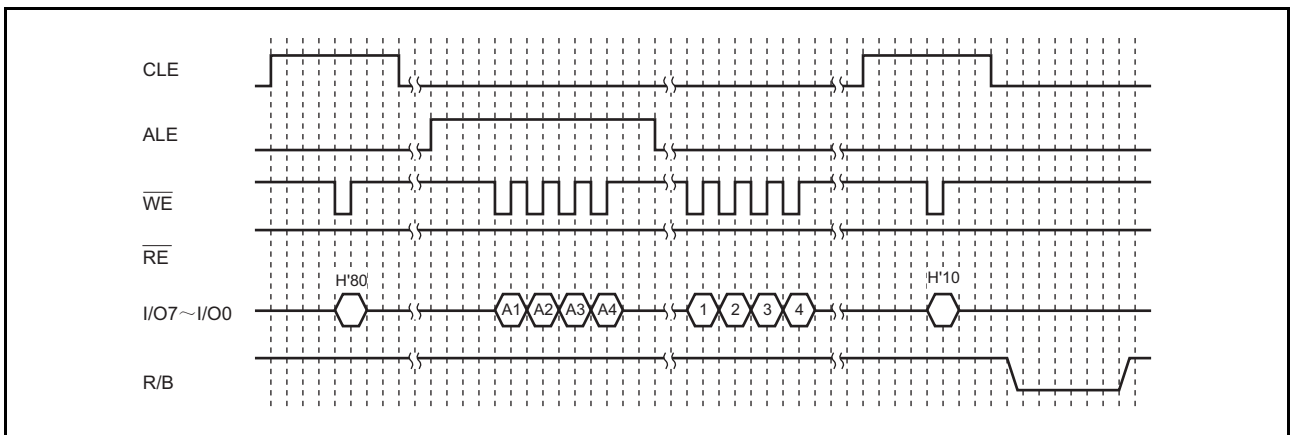


图 25.7 NAND 型闪存的编程时序 (1)

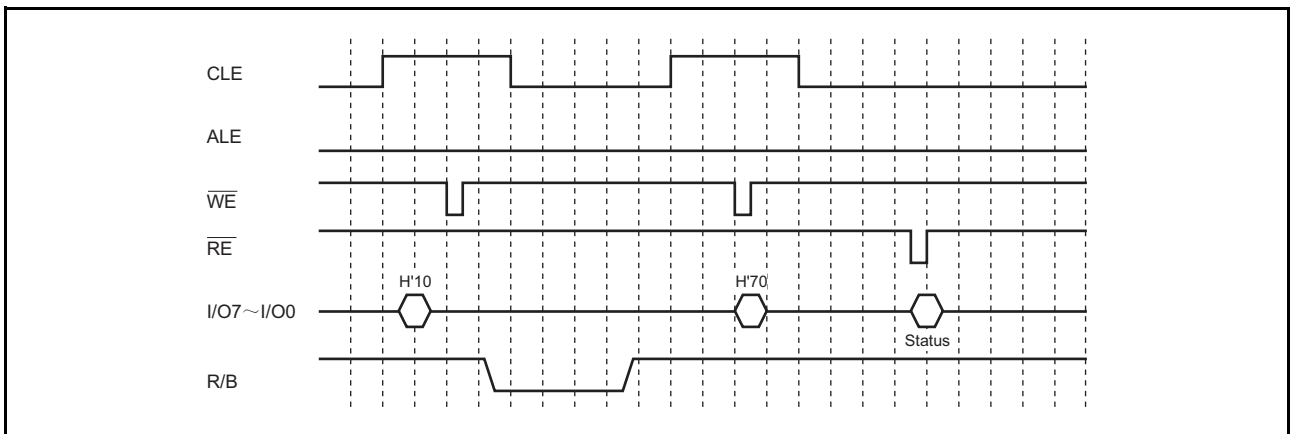


图 25.8 NAND 型闪存的编程时序 (2)

25.4.4 扇区存取模式

在扇区存取模式中，如果指定要存取的物理扇区号，就能进行以扇区为单位的读写。写时进行 ECC 附加处理，读时进行 ECC 错误检查处理。

512 字节的数据保存到 FLDTFIFO，16 字节的管理码保存到 FLECFIFO，所以必须设定 FLINTDMACR 的 DREQ1EN 和 DREQ0EN，分别进行 DMA 传送。

闪存内的扇区（数据 + 管理码）、地址空间的存储器和 DMA 传送关系如图 25.9 所示。

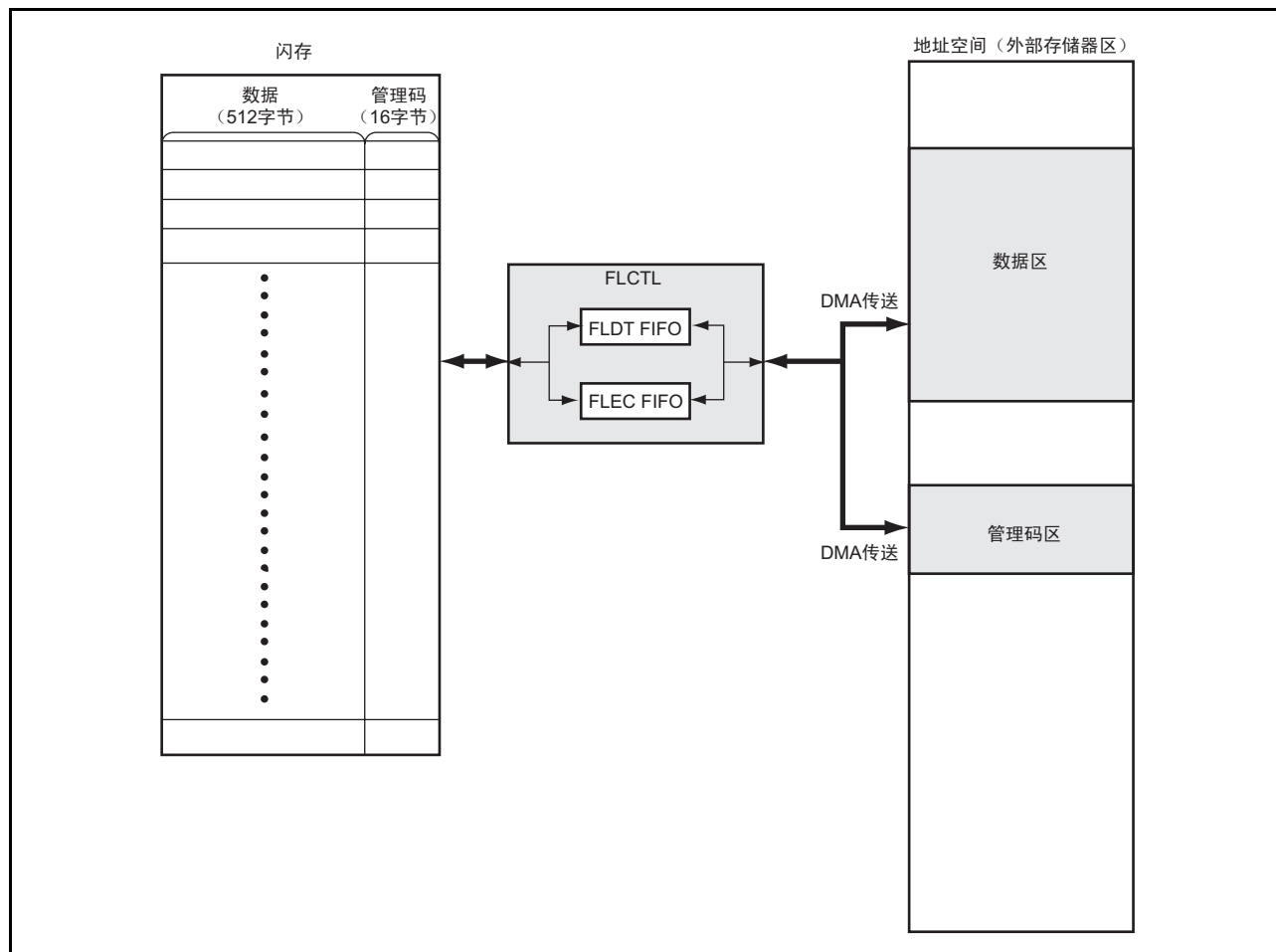


图 25.9 DMA 传送和扇区（数据和管理码）、存储器和 DMA 传送的关系

(1) 物理扇区

NAND 型闪存的物理扇区地址和闪存的地址的关系如图 25.10 所示。

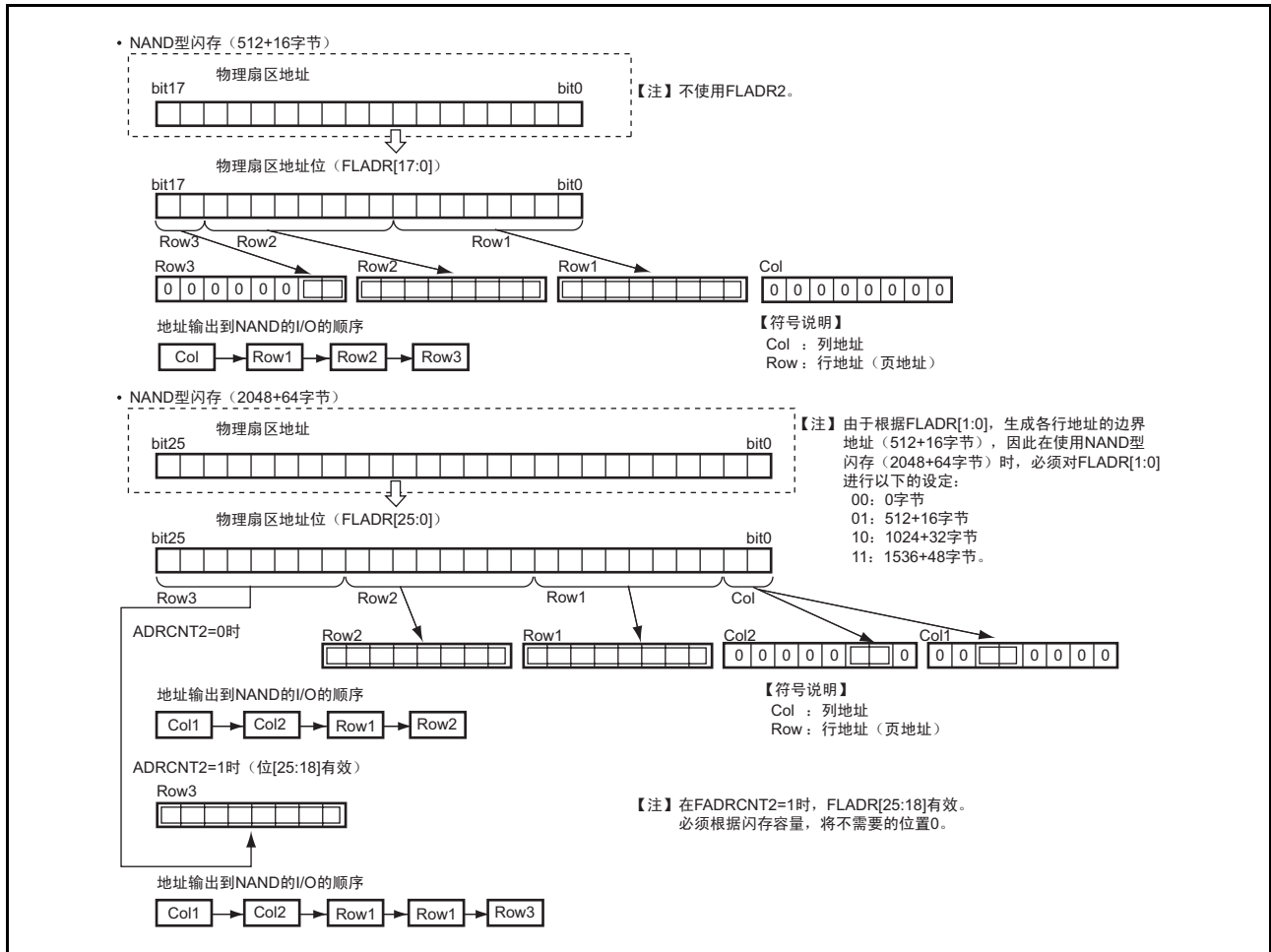


图 25.10 扇区号和 NAND 型闪存的地址展开例子

(2) 连续扇区的存取

如果指定 NAND 型闪存的起始物理扇区和扇区传送次数，就能读写连续的物理扇区。在中途存在不良扇区，并且在传送物理扇区不连续的 0 ~ 40 的逻辑扇区时，物理扇区指定寄存器和扇区传送次数指定寄存器的设定例子如图 25.11 所示。

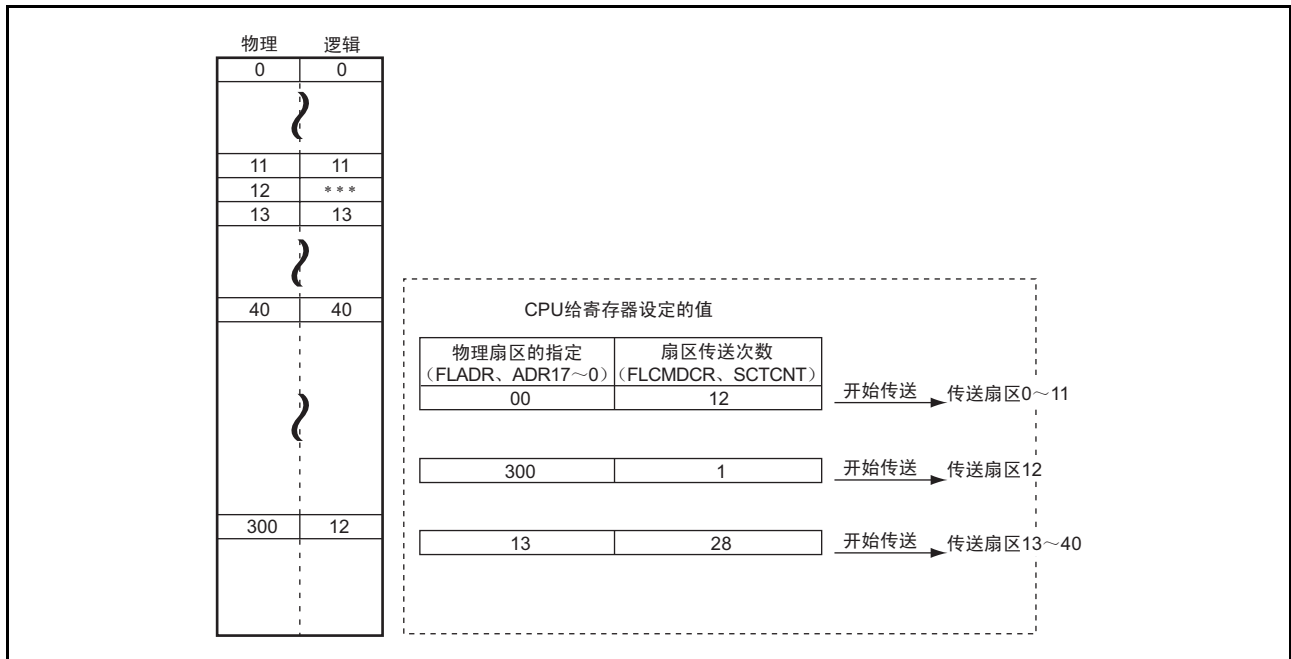


图 25.11 存在不良扇区的扇区存取例子

25.4.5 ECC 的纠错

本 FLCTL 在扇区存取模式的写操作时生成并附加 ECC 代码，在扇区存取模式的读操作时进行 ECC 错误检查，但是不进行纠错。需要通过软件进行纠错处理。

25.4.6 状态的读操作

FLCTL 能读取 NAND 型闪存的状态寄存器的值。NAND 型闪存的状态寄存器的值是从 I/O7 ~ 0 输入的，被保存到 FLBSYCNT 的 STAT[7:0] 位，CPU 能读取 FLBSYCNT 的 STAT[7:0] 位。在状态寄存器的值被保存到 FLBSYCNT 的 STAT[7:0] 位时，如果检查到编程错误或者擦除错误，就将 FLINTDMACR 的 STERB 位置 1。此时，如果 FLINTDMACR 的 STERINTE 位为允许状态，就向 CPU 产生中断。

通过给 NAND 型闪存输入命令 H'70，能读取 NAND 型闪存的状态寄存器。如果将 FLCMDRCR 的 DOSR 位置 1 并且在命令存取模式或者扇区存取模式中进行编程，FLCTL 就自动将 H'70 输入到 NAND 型闪存，进行状态的读取。在读 NAND 型闪存状态时，从 I/O7 ~ 0 输入的状态寄存器各位的含义如表 25.4 所示。

表 25.4 NAND 型闪存的状态

I/O	状态 (definition)	说明
I/O7	编程保护	0: 不能编程 1: 能编程
I/O6	就绪 / 忙	0: 忙状态 1: 就绪状态
I/O5 ~ 1	保留	—
I/O0	编程 / 擦除	0: Pass (成功) 1: Fail (失败)

25.5 中断处理

FLCTL 有 6 种中断源 (状态错误、就绪 / 忙超时错误、ECC 错误, 传送结束, FIFO0 传送请求, FIFO1 传送请求)。全部中断源都有独立的中断标志。如果通过中断允许位允许中断, 就向 CPU 产生独立的中断请求。对 CPU, 状态错误、就绪 / 忙超时错误和 ECC 错误共用 FLSTE 中断。

表 25.5 FLCTL 的中断请求

中断源	中断标志	允许位	意义	优先级
FLSTE 中断	STERB	STERINTE	状态错误	高   低
	BTOERB	RBERINTE	就绪 / 忙超时错误	
	ECERB	ECERINTE	ECC 错误	
FLTEND 中断	TREND	TEINTE	传送结束	
FLTRQ0 中断	TRREQF0	TRINTE0	FIFO0 传送请求	
FLTRQ1 中断	TRREQF1	TRINTE1	FIFO1 传送请求	

【注】 虽然存在 FIFO0 超限 / 欠载和 FIFO1 超限 / 欠载标志, 但不向 CPU 请求中断。

25.6 DMA 传送的设定

FLCTL 能从数据区 FLDTFIFO 和管理码区 FLECFIFO 分别产生 DMA 传送请求。各存取模式的 DMA 传送允许与否如表 25.6 所示。

表 25.6 DMA 传送的设定

	扇区存取模式	命令存取模式
FLDTFIFO	能	能
FLECFIFO	能	不能

有关 DMAC 的设定, 请参照“第 12 章 直接存储器存取控制器 (DMAC)”。

第 26 章 采样速率转换器 (SRC)

采样速率转换器 (SRC) 是转换 WMA/MP3/AAC 等各种解码器生成的数据采样速率的模块。

26.1 特点

- 数据格式：16 位（立体声/单声道）
- 采样速率
输入：能选择 8kHz、11.025kHz、12kHz、16kHz、22.05kHz、24kHz、32kHz、44.1kHz 或者 48kHz。
输出：能选择 32kHz、44.1kHz 或者 48kHz。
- 处理性能：1 个采样输出间隔最大约为 10 μ s（外围总线时钟=54MHz）
- SNR： ≥ 93 db
- 中断源：3 种
输入数据 FIFO 空、输出数据 FIFO 满、输出数据 FIFO 盖写
- DMA 传送源：2 种
输入数据 FIFO 空、输出数据 FIFO 满
- 模块待机模式
在不使用时停止给 SRC 提供时钟，能降低功耗。

SRC 的框图如图 26.1 所示。

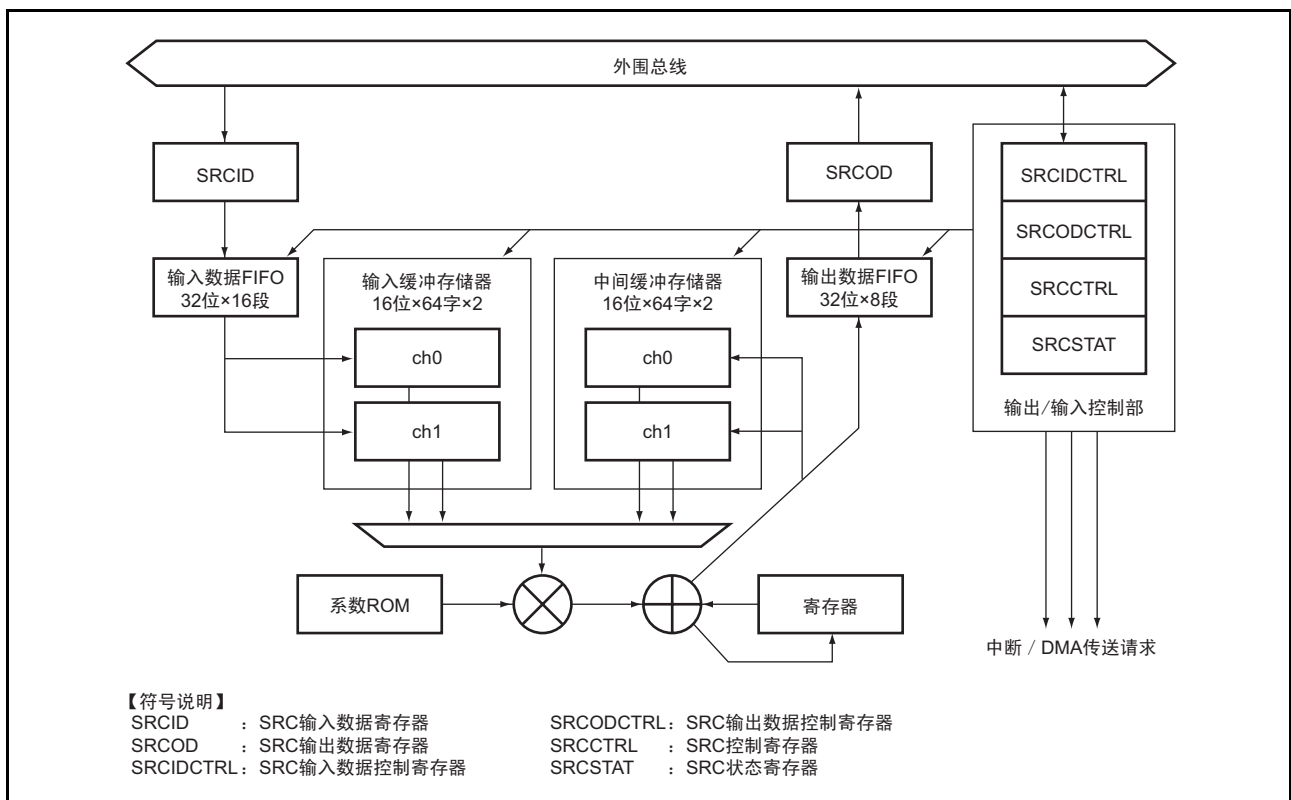


图 26.1 SRC 的框图

26.2 寄存器说明

SRC 中有以下寄存器。

表 26.1 寄存器结构

寄存器名	略称	R/W	P4 区地址	区域 7 地址	存取长度
SRC 输入数据寄存器	SRCID	R/W	H'FFF2 0000	H'1FF2 0000	16、32
SRC 输出数据寄存器	SRCOD	R	H'FFF2 0004	H'1FF2 0004	16、32
SRC 输入数据控制寄存器	SRCIDCTRL	R/W	H'FFF2 0008	H'1FF2 0008	16
SRC 输出数据控制寄存器	SRCODCTRL	R/W	H'FFF2 000A	H'1FF2 000A	16
SRC 控制寄存器	SRCCTRL	R/W	H'FFF2 000C	H'1FF2 000C	16
SRC 状态寄存器	SRCSTAT	R/(W)*	H'FFF2 000E	H'1FF2 000E	16

【注】 * bit15 ~ 3 是只读位。bit2 ~ 0 在读 1 后只能写 0。

表 26.2 各处理模式的寄存器状态

寄存器名	略称	上电复位	睡眠	待机
SRC 输入数据寄存器	SRCID	H'0000 0000	保持	保持
SRC 输出数据寄存器	SRCOD	H'0000 0000	保持	保持
SRC 输入数据控制寄存器	SRCIDCTRL	H'0000	保持	保持
SRC 输出数据控制寄存器	SRCODCTRL	H'0000	保持	保持
SRC 控制寄存器	SRCCTRL	H'0000	保持	保持
SRC 状态寄存器	SRCSTAT	H'0002	保持	保持

26.2.1 SRC 输入数据寄存器 (SRCID)

SRCID 是 32 位可读写寄存器，用于采样速率转换前的数据输入。位的读取值全部为 0。写给 SRCID 的数据被保存到 16 段的输入数据 FIFO。在输入数据 FIFO 的数据个数为 16 时，SRCID 的写操作无效。

为立体声数据时，bit31 ~ 16 保存 ch0 数据，bit15 ~ 0 保存 ch1 数据。为单声道数据时，bit31 ~ 16 保存的数据有效，bit15 ~ 0 保存的数据无效。

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

转换处理的对象数据的调整因 SRCIDCTRL 的 IED 位的设定值而不同，SRCIDCTRL 的 IED 位的设定值和数据的调整关系如表 26.3 所示。

表 26.3 转换数据的调整

IED	ch0[15:8]	ch0[7:0]	ch1[15:8]	ch1[7:0]
0	SRCID[31:24]	SRCID[23:16]	SRCID[15:8]	SRCID[7:0]
1	SRCID[23:16]	SRCID[31:24]	SRCID[7:0]	SRCID[15:8]

26.2.2 SRC 输出数据寄存器 (SRCOD)

SRCOD 是 32 位只读寄存器，用于采样速率转换后的数据输出。能从 SRCOD 读取保存在 8 段输出数据 FIFO 中的数据。

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

SRCOD 内容的调整因 SRCODCTRL 的 OCH 和 OED 位的设定值而不同，SRCODCTRL 的 OCH、OED 位的设定值和 SRCOD 保存的数据的调整关系如表 26.4 所示。

表 26.4 SRCOD 的调整

OCH	OED	SRCOD[31:24]	SRCOD[23:16]	SRCOD[15:8]	SRCOD[7:0]
0	0	ch0[15:8]	ch0[7:0]	ch1[15:8]*2	ch1[7:0]*2
	1	ch0[7:0]	ch0[15:8]	ch1[7:0]*2	ch1[15:8]*2
1*1	0	ch1[15:8]	ch1[7:0]	ch0[15:8]	ch0[7:0]
	1	ch1[7:0]	ch1[15:8]	ch0[7:0]	ch0[15:8]

【注】 *1 在处理单声道数据时，不能设定。

*2 在处理单声道数据时，为无效数据。

*3 SRCCTRL 寄存器的 CL 位置 1 后的读取值为 0，如果在将此位置 1 前进行读取，就不保证读取值。

26.2.3 SRC 输入数据控制寄存器 (SRCIDCTRL)

SRCIDCTRL 是 16 位可读写寄存器，设定输入数据的字节排序格式、中断请求的允许 / 禁止和触发数据的个数。

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

位	位名	初始值	R/W	说明
15 ~ 10	—	全 0	R	保留位 读取值为 0，只能写 0。
9	IED	0	R/W	输入数据的字节排序指定位 指定输入数据的字节排序格式。 0: 大端法 1: 小端法
8	IEN	0	R/W	输入数据空中断允许位 在输入 FIFO 的数据个数小于等于 IFTRG[1:0] 位设定的触发个数并且 SRC 状态寄存器 (SRCSTAT) 的 IINT 位为 1 时，允许或者禁止产生输入数据空中断请求。 0: 禁止产生输入数据空中断请求。 1: 允许产生输入数据空中断请求。
7 ~ 2	—	全 0	R	保留位 读取值为 0，只能写 0。
1、0	IFTRG[1:0]	00	R/W	输入 FIFO 数据触发个数位 指定 SRC 状态寄存器 (SRCSTAT) 的 IINT 位的置位条件。当保存在输入 FIFO 中的输入数据个数小于等于以下所示的设定触发个数时，将 IINT 位置 1。 00: 0 01: 4 10: 8 11: 12

26.2.4 SRC 输出数据控制寄存器 (SRCODCTRL)

SRCODCTRL 是 16 位可读写寄存器，设定输出数据的通道交换、字节排序格式、中断请求的允许 / 禁止和触发数据的个数。

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

位	位名	初始值	R/W	说明
15 ~ 11	—	全 0	R	保留位 读取值为 0，只能写 0。
10	OCH	0	R/W	输出数据通道交换位 指定 SRC 输出数据寄存器 (SRCOD) 的通道交换。 在转换单声道数据时，不能置 1。 0: 不交换通道 (和输入数据的顺序相同) 1: 交换通道 (和输入数据的顺序相反)
9	OED	0	R/W	输出数据的字节排序指定位 指定输出数据的字节排序格式。 0: 大端法 1: 小端法
8	OEN	0	R/W	输出数据满中断允许位 在输出 FIFO 的数据个数大于等于 OFTRG[1:0] 位设定的触发个数并且 SRC 状态寄存器 (SRCSTAT) 的 OINT 位为 1 时，允许或者禁止产生输出数据满中断请求。 0: 禁止产生输出数据满中断请求。 1: 允许产生输出数据满中断请求。
7 ~ 2	—	全 0	R	保留位 读取值为 0，只能写 0。
1、0	OFTRG[1:0]	00	R/W	输出 FIFO 数据触发个数位 指定 SRC 状态寄存器 (SRCSTAT) 的 OINT 位的置位条件。当保存在输出 FIFO 中的输出数据个数大于等于以下所示的设定触发个数时，将 OINT 位置 1。 00: 1 01: 2 10: 4 11: 6

26.2.5 SRC 控制寄存器 (SRCCTRL)

SRCCTRL 是 16 位可读写寄存器，设定模块运行的允许 / 禁止、中断请求的允许 / 禁止、清除处理、内部工作存储器的清除处理和输入 / 输出采样速率。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SRCEN	—	EEN	FL	CL	IFS[3:0]			—	—	OFS[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	R	R/W	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												
15 ~ 13	—	全 0	R	保留位 读取值为 0，只能写 0。												
12	SRCEN	0	R/W	SRC 模块允许位 设定是允许还是禁止 SRC 模块运行。 0：禁止 SRC 模块运行 1：允许 SRC 模块运行 【注】 在 SRCEN=1 时，不能改变下列位的设定值。 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>寄存器名</th><th>位</th><th>位名</th></tr> </thead> <tbody> <tr> <td>SRCIDCTRL</td><td>9</td><td>IED</td></tr> <tr> <td>SRCODCTRL</td><td>10、9</td><td>OCH、OED</td></tr> <tr> <td>SRCCTRL</td><td>7 ~ 4、0</td><td>IFS[3:0]、OFS</td></tr> </tbody> </table>	寄存器名	位	位名	SRCIDCTRL	9	IED	SRCODCTRL	10、9	OCH、OED	SRCCTRL	7 ~ 4、0	IFS[3:0]、OFS
寄存器名	位	位名														
SRCIDCTRL	9	IED														
SRCODCTRL	10、9	OCH、OED														
SRCCTRL	7 ~ 4、0	IFS[3:0]、OFS														
11	—	0	R	保留位 读取值为 0，只能写 0。												
10	EEN	0	R/W	输出 FIFO 盖写中断允许位 在读输出 FIFO 的数据前发生盖写并且 SRC 状态寄存器 (SRCSTAT) 的 OVF 位为 1 时，允许或者禁止产生输出 FIFO 盖写中断请求。 0：禁止产生输出 FIFO 盖写中断请求。 1：允许产生输出 FIFO 盖写中断请求。												
9	FL	0	R/W	内部工作存储器清除位 如果将此位置 1，就开始对保存在输入 FIFO、输入缓冲存储器和中间缓冲存储器中的全部数据进行采样速率的转换（清除处理）。读取值为 0。 在 SRCEN=0 时，即使将此位置 1 也不进行清除处理。在输入缓冲存储器内的数据个数不满 64 的状态下，如果将此位置 1，就得不到有效的输出数据，所以不进行清除处理。												
8	CL	0	R/W	内部工作存储器清除位 如果将此位置 1，就清除输入 FIFO、输出 FIFO、输入缓冲存储器、中间缓冲存储器和累加器。读取值为 0。 必须在执行 SRC 前将此位置 1，清除 SRC。 为了进行清除处理，必须在此位置 1 后等待 32 个 cyc，然后进行下一个处理。在将此位置 1 时，必须设定 IFS[3:0] 和 OFS。												

位	位名	初始值	R/W	说明
7 ~ 4	IFS[3:0]	0000	R/W	输入采样速率 设定输入采样速率。 0000: 8.0kHz 0001: 11.025kHz 0010: 12.0kHz 0011: 禁止设定 0100: 16.0kHz 0101: 22.05kHz 0110: 24.0kHz 0111: 禁止设定 1000: 32.0kHz 1001: 44.1kHz 1010: 48.0kHz 1011: 禁止设定 1100: 禁止设定 1101: 禁止设定 1110: 禁止设定 1111: 禁止设定
3、2	—	全 0	R	保留位 读取值为 0, 只能写 0。
1、0	OFS[1:0]	全 0	R/W	输出采样速率位 设定输出采样速率。 00: 44.1kHz 01: 48.0kHz 10: 32kHz 11: 禁止设定

作为转换结果, 能得到的输出数据个数由以下所示的公式 (A) 或者公式 (B) 求得。IFS、OFS[1:0] 位的设定值和适用的计算公式的关系如表 26.5 所示。

$$\text{输出数据的个数} = \text{输入数据的个数} \times \frac{\text{输出的采样速率}}{\text{输入的采样速率}} \quad \dots \text{(A)}$$

$$\text{输出数据的个数} = \text{输入数据的个数} \times \frac{\text{输出的采样速率}}{\text{输入的采样速率}} - 1 \quad \dots \text{(B)}$$

表 26.5 采样速率的设定和输出数据个数的关系

OFS[1:0] 设定值 (输出采样速率 [kHz])	IFS[3:0] 设定值 (输入采样速率 [kHz])								
	0000 (8.0)	0001 (11.025)	0010 (12.0)	0100 (16.0)	0101 (22.05)	0110 (24.0)	1000 (32.0)	1001 (44.1)	1010 (48.0)
00(44.1)	B	A	A	B	A	A	B	—	A
01(48.0)	B	B	A	B	B	A	B	B	—
10(32.0)	A	B	B	A	B	A	—	B	A

26.2.6 SRC 状态寄存器 (SRCSTAT)

SRCSTAT 是 16 位可读写寄存器，表示输出 FIFO 和输入 FIFO 的数据个数以及各中断源的发生状态和清除处理的执行状态。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OFDN[3:0]			IFDN[4:0]				—	—	FLF	—	OVF	IINT	OINT		
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*

【注】* 只能在读取1后写0。

位	位名	初始值	R/W	说明
15 ~ 12	OFDN[3:0]	0000	R	输出 FIFO 数据计数 表示保存在输出 FIFO 中的数据个数。
11 ~ 7	IFDN[4:0]	00000	R	输入 FIFO 数据计数 表示保存在输入 FIFO 中的数据个数。
6、5	—	全 0	R	保留位 读取值为 0，只能写 0。
4	FLF	0	R	清除处理状态标志位 表示正在执行清除处理。 [清除条件] • 在清除处理结束时 • 在将 SRCCTRL 的 CL 位置 1 时 [置位条件] • 在 SRCCTRL 的 FL 位置 1 时
3	—	全 0	R	保留位 读取值为 0，只能写 0。
2	OVF	0	R/(W)*	输出 FIFO 盖写中断请求标志位 在输出 FIFO 的数据个数为 8 时，表示结束下一个数据的转换处理。 此时，在输出数据 FIFO 变为空（通过读 SRC 输出数据寄存器（SRCOD））前，停止转换处理。 [清除条件] • 在 OVF=1 的状态下读 OVF 后将 OVF 位置 0 时 • 在将 SRCCTRL 的 CL 位置 1 时 [置位条件] • 在输出 FIFO 的数据个数为 8 的状态下，结束下一个数据的转换处理时。
1	IINT	1	R/(W)*	输入 FIFO 空中断请求标志位 表示保存在输入 FIFO 中的数据个数小于等于 SRC 输入数据控制寄存器（SRCIDCTRL）的 IFTRG[1:0] 位设定的触发个数。 [清除条件] • 在 IINT=1 的状态下读 IINT 后将 IINT 位置 0 时 • 通过 DMA 传送，在输入 FIFO 的数据个数超过设定的触发个数时 [设置条件] • 当保存在输入 FIFO 中的数据个数小于等于设定的触发个数时 • 在将 SRCCTRL 的 CL 位置 1 时

位	位名	初始值	R/W	说明
0	OINT	0	R/(W)*	<p>输出 FIFO 满中断请求标志位</p> <p>表示保存在输出 FIFO 中的数据个数大于等于 SRC 输出数据控制寄存器 (SRCODCTRL) 的 OFTRG[1:0] 位设定的触发个数。</p> <p>[清除条件]</p> <ul style="list-style-type: none"> 在 OINT=1 的状态下读 OINT 后将 OINT 位置 0 时 通过 DMA 传送, 输出 FIFO 的数据个数小于设定的触发个数时 <p>[置位条件]</p> <ul style="list-style-type: none"> 当保存在输出 FIFO 中的数据个数大于等于设定的触发个数时

【注】 * 只能在读 1 后写 0。

26.3 运行说明

26.3.1 初始设定

初始设定的步骤如 26.2 所示。

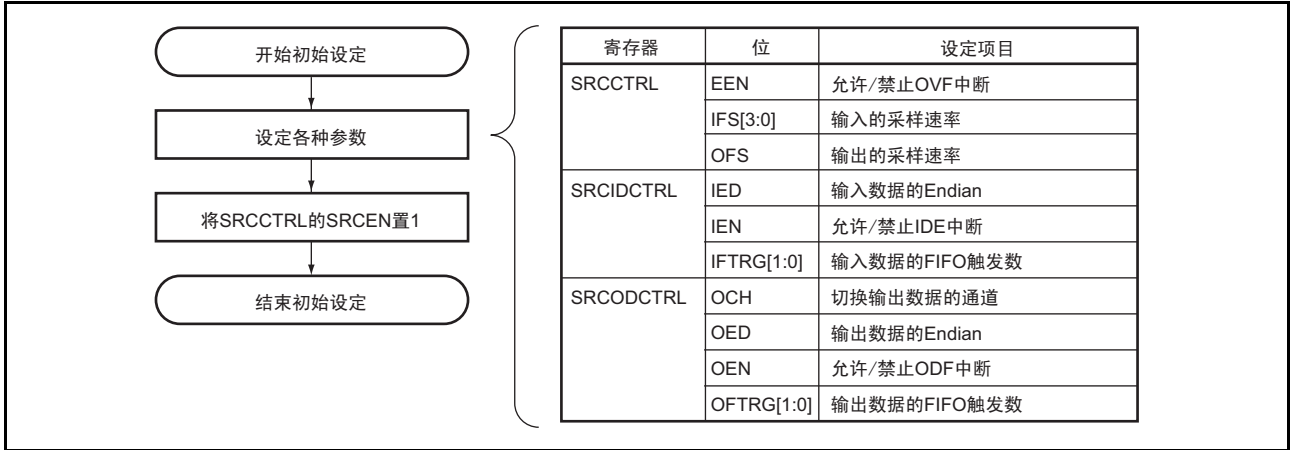


图 26.2 初始设定的步骤例子

26.3.2 数据输入

数据输入的步骤如图 26.3 所示

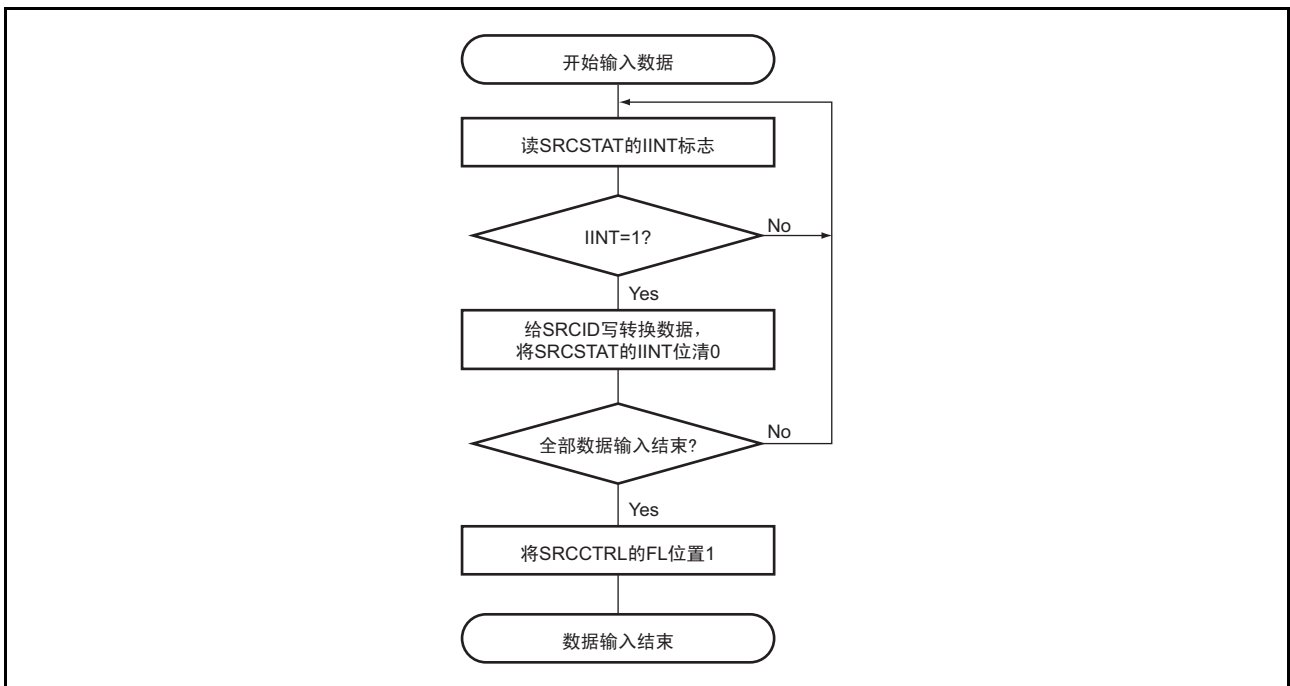


图 26.3 数据输入的步骤例子

(1) 向 CPU 请求中断时

1. 将SRCIDCTRL的IEN位置1。
2. 设定中断控制器。
3. 如果将SRCSTAT的IINT位置1, 就产生IDE中断请求。中断处理程序在SRCSTAT的IINT位为1后将数据写到SRCID, 并且将SRCSTAT的IINT位置0, 然后从中断处理程序返回。
4. 重复执行3, 在全部的数据输入结束后将SRCCTRL的FL位置1。

(2) 通过中断启动 DMAC 时

1. 将DMAC的任一通道设定给SRC的IDEI。
2. 将SRCIDCTRL的IEN位置1。
3. 如果将SRCSTAT的IINT位置1, 就产生IDE中断请求, 并且启动DMAC。通过DMA传送将数据写到SRCID。此时, 如果输入数据FIFO的数据个数超过SRCIDCTRL的IFTRG[1:0]位设定的触发个数, 就清除SRCSTAT的IINT位。
4. 重复执行3, 在全部的数据输入结束后将SRCCTRL的FL位置1。

26.3.3 数据输出

数据输出的步骤如图 26.4 所示。

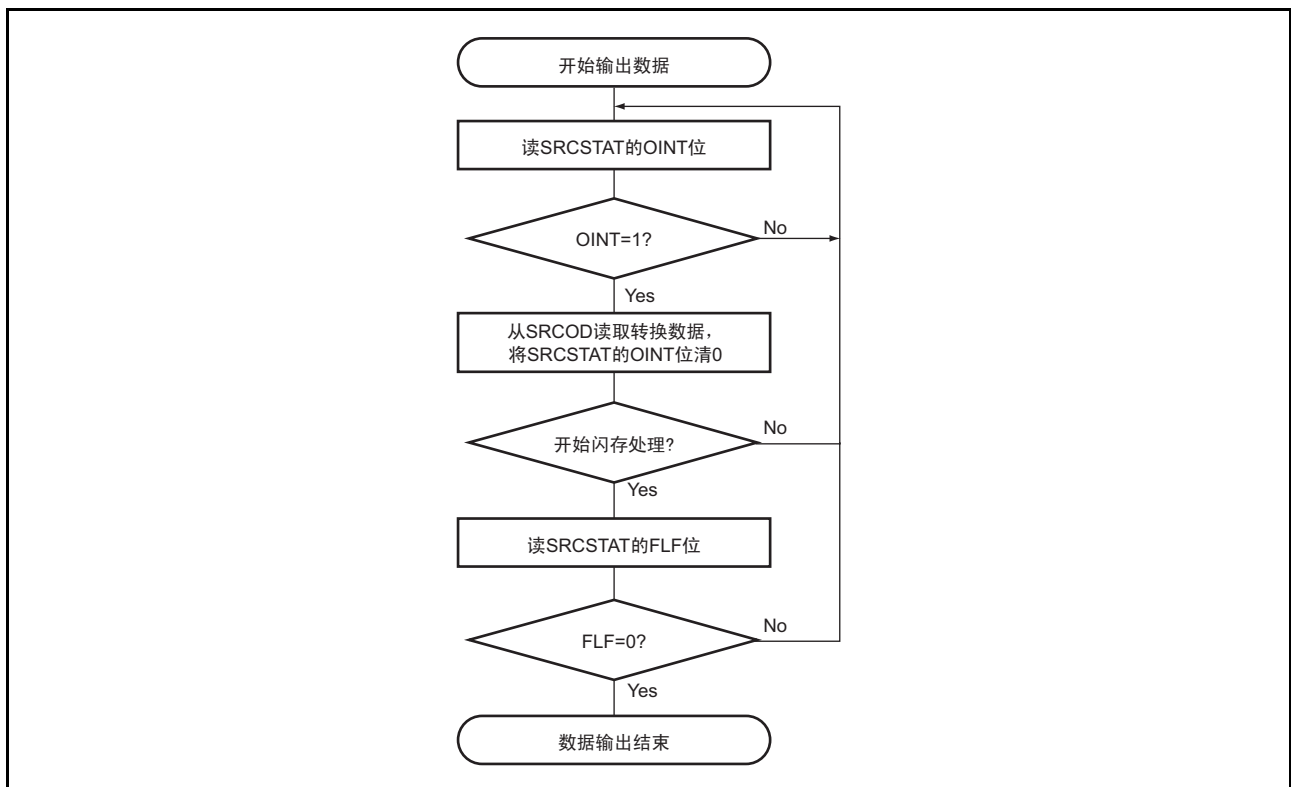


图 26.4 数据输出的步骤例子

(1) 向 CPU 请求中断时

1. 将SRCODCTRL的OEN位置1。
2. 设定中断控制器。
3. 如果将SRCSTAT的OINT位置1, 就产生ODF中断请求。中断处理程序在SRCSTAT的OINT位为1后, 从SRCOD读取数据, 并且将SRCSTAT的OINT位置0, 然后从中断处理程序返回。
4. 在开始清除处理后重复执行3, 直到SRCSTAT的FLF位为0为止。

(2) 通过中断启动 DMAC 时

1. 将DMAC的任一通道设定给SRC的ODFI。
2. 将SRCODCTRL的OEN位置1。
3. 如果将SRCSTAT的OINT位置1, 就产生ODF中断请求, 并且启动DMAC。通过DMA传送, 从SRCOD读取数据。此时, 如果输出数据FIFO的数据个数小于SRCODCTRL的OFTRG[1:0]位设定的触发个数, 就清除SRCSTAT的OINT位。
4. 在开始清除处理后重复执行3, 直到SRCSTAT的FLF位为0为止。

26.4 中断

SRC 的中断源有输入 FIFO 数据空 (IDEI)、输出 FIFO 数据满 (ODFI)、输出 FIFO 盖写 (OVF) 3 种。中断请求的种类和产生的条件如 26.6 所示。

表 26.6 中断请求的种类和产生的条件

中断请求	略称	产生条件	启动 DMAC
输入数据 FIFO 空	IDEI	$(IINT=1) \cdot (IEN=1) \cdot (SRCEN=1)$	可启动
输出数据 FIFO 满	ODFI	$(OINT=1) \cdot (OEN=1) \cdot (SRCEN=1)$	可启动
输出数据 FIFO 重写	OVF	$(OVF=1) \cdot (EEN=1) \cdot (SRCEN=1)$	不可启动

如果中断产生条件成立, CPU 就执行中断异常处理。必须在中断异常处理程序内清除中断源标志。

通过 DMAC 的设定, IDEI 中断和 ODFI 中断能启动 DMAC。通过 DMA 传送将数据写到 SRCID, 当输入数据 FIFO 的数据个数超过设定触发个数时, 就清除 IINT。同样地, 从 SRCOD 读取数据, 当输出数据 FIFO 的数据个数小于设定触发个数时, 就清除 OINT。

26.5 使用时的注意事项

26.5.1 寄存器存取注意事项

如果给 SRCCTRL 的 FL 位写 1，就在 SRCSTAT 的 FLF 位被置位前需要 3 个外围时钟 (Pφ) 周期。另外，因 CPU 不等待寄存器写完就执行后续的指令，所以通过 SRCCTRL 的写指令后的指令无法读到 FLF 位被置位的状态。在要确认清除处理的执行状态时，在 SRCCTRL 的写指令后，虚读 SRCCTRL 或者 SRCSTAT，等待 FLF 的置位。

26.5.2 有关清除处理的注意事项

如果将 SRC 控制寄存器 (SRCCTRL) 的 FL 位置 1，SRC 就在现在为止的输入数据的最后附加 0 数据的同时继续进行转换处理。在音频数据输入结束并且不存在后继数据时，必须进行清除处理。

在执行清除处理后再次进行转换处理时，必须通过以下任意一个操作清除内部工作存储器：

- 将 SRCCTRL 的 CL 位置 1
- 给 SRCCTRL 的 SRCEN 位写 0 后继续写 1。

第 27 章 通用输入 / 输出端口 (GPIO)

27.1 特点

本 LSI 有 10 组 (输入 / 输出: 77 个) 通用端口 (端口 A ~ J)。
通用输入 / 输出端口 (GPIO) 的各端口和外围模块的引脚多路复用。能选择 GPIO 或者外围模块的引脚。
GPIO 有以下特点:

- 各端口的引脚都是多路复用引脚, 能通过端口控制寄存器按引脚进行引脚功能和上拉 MOS 的控制。
- 每个端口都有 1 个用于保存各引脚数据的数据寄存器。
- 支持 GPIO 中断 (端口 A 和端口 B)

能通过通用输入 / 输出端口 (GPIO) 寄存器控制的多路复用引脚如表 27.1 和表 27.2 所示。

表 27.1 能通过端口控制寄存器控制的多路复用引脚一览表

端口	端口功能 (相关模块)	其他功能 1 (相关模块)	其他功能 2 (相关模块)	其他功能 3 (相关模块)	其他功能 4 (相关模块)	GPIO 中断
A	PA7 输入/输出 (端口)	STATUS1 输出 (SYSTEM)	RTS2 输入/输出 (SCIF)	—	—	PINT15 输入 (INTC)
	PA6 输入/输出 (端口)	STATUS0 输出 (SYSTEM)	CTS2 输入/输出 (SCIF)	—	—	PINT14 输入 (INTC)
	PA5 输入/输出 (端口)	FCE 输出 (FLCTL)	—	—	—	PINT13 输入 (INTC)
	PA4 输入/输出 (端口)	FRE 输出 (FLCTL)	—	—	—	PINT12 输入 (INTC)
	PA3 输入/输出 (端口)	FEW 输出 (FLCTL)	—	—	—	PINT11 输入 (INTC)
	PA2 输入/输出 (端口)	TxD2 输出 (SCIF)	—	—	—	PINT10 输入 (INTC)
	PA1 输入/输出 (端口)	RxD2 输入 (SCIF)	—	—	—	PINT9 输入 (INTC)
	PA0 输入/输出 (端口)	SCK2 输入/输出 (SCIF)	—	—	—	PINT8 输入 (INTC)
B	PB7 输入/输出 (端口)	A25 输出 (ADDRESS)	DREQ0 输入 (DMAC)	RTS0 输入/输出 (SCIF)	—	PINT7 输入 (INTC)
	PB6 输入/输出 (端口)	A24 输出 (ADDRESS)	DACK0 输出 (DMAC)	CTS0 输入/输出 (SCIF)	—	PINT6 输入 (INTC)
	PB5 输入/输出 (端口)	A23 输出 (ADDRESS)	DTEND0 输出 (DMAC)	RTS1 输入/输出 (SCIF)	—	PINT5 输入 (INTC)
	PB4 输入/输出 (端口)	A22 输出 (ADDRESS)	CTS1 输入/输出 (SCIF)	—	—	PINT4 输入 (INTC)
	PB3 输入/输出 (端口)	A21 输出 (ADDRESS)	—	—	—	PINT3 输入 (INTC)
	PB2 输入/输出 (端口)	A20 输出 (ADDRESS)	—	—	—	PINT2 输入 (INTC)
	PB1 输入/输出 (端口)	A19 输出 (ADDRESS)	—	—	—	PINT1 输入 (INTC)

端口	端口功能 (相关模块)	其他功能 1 (相关模块)	其他功能 2 (相关模块)	其他功能 3 (相关模块)	其他功能 4 (相关模块)	GPIO 中断
B	PB0 输入/输出 (端口)	A18 输出 (ADDRESS)	—	—	—	$\overline{\text{PINT0}}$ 输入 (INTC)
C	PC7 输入/输出 (端口)	AUDIO_CLK0 输入 (SSI)	—	—	—	—
	PC6 输入/输出 (端口)	AUDIO_CLK1 输入 (SSI)	—	—	—	—
	PC5 输入/输出 (端口)	AUDIO_CLK2 输入 (SSI)	—	—	—	—
	PC4 输入/输出 (端口)	SSIWS2 输入/输出 (SSI)	—	—	—	—
	PC3 输入/输出 (端口)	SSISCK2 输入/输出 (SSI)	—	—	—	—
	PC2 输入/输出 (端口)	SSIDATA2 输入/输出 (SSI)	—	—	—	—
	PC1 输入/输出 (端口)	$\overline{\text{ASEBRK}}$ / BRKACK 输入/输出 (AUD)	TCLK 输入 (TMU)	—	—	—
	PC0 输入/输出 (端口)	FALE 输出 (FLCTL)	—	—	—	—
D	PD7 输入/输出 (端口)	CRS 输入 (EtherC)	IDEA1_M 输出 (ATAPI)	—	—	—
	PD6 输入/输出 (端口)	TX_ER 输出 (EtherC)	$\overline{\text{IDEIOWR}}_M$ 输出 (ATAPI)	—	—	—
	PD5 输入/输出 (端口)	TX_CLK 输入 (EtherC)	IDED15_M 输入/输出 (ATAPI)	—	—	—
	PD4 输入/输出 (端口)	TX_EN 输出 (EtherC)	IDED0_M 输入/输出 (ATAPI)	—	—	—
	PD3 输入/输出 (端口)	MII_TXD0 输出 (EtherC)	SSISCK5 输入/输出 (SSI)	$\overline{\text{IDEIORDY}}_M$ 输入 (ATAPI)	—	—
	PD2 输入/输出 (端口)	MII_TXD1 输出 (EtherC)	SSIWS5 输入/输出 (SSI)	$\overline{\text{IDEIORD}}_M$ 输出 (ATAPI)	—	—
	PD1 输入/输出 (端口)	MII_TXD2 输出 (EtherC)	AUDIO_CLK5 输入 (SSI)	IDEINT_M 输入 (ATAPI)	—	—
	PD0 输入/输出 (端口)	MII_TXD3 输出 (EtherC)	SSIDATA5 输入/输出 (SSI)	$\overline{\text{IODACK}}_M$ 输出 (ATAPI)	—	—
E	PE7 输入/输出 (端口)	COL 输入 (EtherC)	IDEA2_M 输出 (ATAPI)	—	—	—
	PE6 输入/输出 (端口)	RX_ER 输入 (EtherC)	IODREQ_M 输入 (ATAPI)	—	—	—
	PE5 输入/输出 (端口)	RX_CLK 输入 (EtherC)	IDED1_M 输入/输出 (ATAPI)	—	—	—
	PE4 输入/输出 (端口)	RX_DV 输入 (EtherC)	IDED14_M 输入/输出 (ATAPI)	—	—	—

端口	端口功能 (相关模块)	其他功能 1 (相关模块)	其他功能 2 (相关模块)	其他功能 3 (相关模块)	其他功能 4 (相关模块)	GPIO 中断
E	PE3 输入/输出 (端口)	MII_RXD0 输入 (EtherC)	SSIWS4 输入/输出 (SSI)	IDED2_M 输入/输出 (ATAPI)	—	—
	PE2 输入/输出 (端口)	MII_RXD1 输入 (EtherC)	SSISCK4 输入/输出 (SSI)	IDED13_M 输入/输出 (ATAPI)	—	—
	PE1 输入/输出 (端口)	MII_RXD2 输入 (EtherC)	SSIDATA4 输入/输出 (SSI)	IDED3_M 输入/输出 (ATAPI)	—	—
	PE0 输入/输出 (端口)	MII_RXD3 输入 (EtherC)	AUDIO_CLK4 输入 (SSI)	IDED12_M 输入/输出 (ATAPI)	—	—
F	PF7 输入/输出 (端口)	D32 输入/输出 (DATA)	—	—	—	—
	PF6 输入/输出 (端口)	D33 输入/输出 (DATA)	—	—	—	—
	PF5 输入/输出 (端口)	D34 输入/输出 (DATA)	—	—	—	—
	PF4 输入/输出 (端口)	EXOUT 输出 (EtherC)	IDECS1_M 输出 (ATAPI)	—	—	—
	PF3 输入/输出 (端口)	LNKSTA 输入 (EtherC)	IDECS0_M 输出 (ATAPI)	—	—	—
	PF2 输入/输出 (端口)	WOL 输出 (EtherC)	IDEA0_M 输出 (ATAPI)	—	—	—
	PF1 输入/输出 (端口)	MDIO 输入/输出 (EtherC)	IDED11_M 输入/输出 (ATAPI)	—	—	—
	PF0 输入/输出 (端口)	MDC 输出 (EtherC)	IDED4_M 输入/输出 (ATAPI)	—	—	—
G	PG7 输入/输出 (端口)	LCD_DATA15 输出 (LCDC)	DR3 输出 (VDC2)	—	—	—
	PG6 输入/输出 (端口)	LCD_DATA14 输出 (LCDC)	DR2 输出 (VDC2)	—	—	—
	PG5 输入/输出 (端口)	LCD_DATA13 输出 (LCDC)	DR1 输出 (VDC2)	—	—	—
	PG4 输入/输出 (端口)	LCD_DATA12 输出 (LCDC)	DR0 输出 (VDC2)	—	—	—
	PG3 输入/输出 (端口)	LCD_DATA11 输出 (LCDC)	DG5 输出 (VDC2)	—	—	—
	PG2 输入/输出 (端口)	LCD_DATA10 输出 (LCDC)	DG4 输出 (VDC2)	—	—	—
	PG1 输入/输出 (端口)	LCD_DATA9 输出 (LCDC)	DG3 输出 (VDC2)	—	—	—
	PG0 输入/输出 (端口)	LCD_DATA8 输出 (LCDC)	DG2 输出 (VDC2)	—	—	—

端口	端口功能 (相关模块)	其他功能 1 (相关模块)	其他功能 2 (相关模块)	其他功能 3 (相关模块)	其他功能 4 (相关模块)	GPIO 中断
H	PH7 输入/输出 (端口)	AUDIO_CLK3 输入 (SSI)	—	—	—	—
	PH6 输入/输出 (端口)	SSIWS3 输入/输出 (SSI)	—	—	—	—
	PH5 输入/输出 (端口)	SSISCK3 输入/输出 (SSI)	—	—	—	—
	PH4 输入/输出 (端口)	SSIDATA3 输入/输出 (SSI)	—	—	—	—
	PH3 输入/输出 (端口)	LCD_CL2 输出 (LCDC)	DE_V 输出 (VDC2)	—	—	—
	PH2 输入/输出 (端口)	LCD_DON 输出 (LCDC)	DCLKOUT 输出 (VDC2)	—	—	—
	PH1 输入/输出 (端口)	LCD_VCP_WC 输出 (LCDC)	DR4 输出 (VDC2)	—	—	—
	PH0 输入/输出 (端口)	LCD_VEP_WC 输出 (LCDC)	DR5 输出 (VDC2)	—	—	—
I	PI4 输入/输出 (端口)	LCD_DATA7 输出 (LCDC)	DG 输出 (VDC2)	BT_DATA7 输出 (VDC2)	—	—
	PI3 输入/输出 (端口)	LCD_DATA6 输出 (LCDC)	DG0 输出 (VDC2)	BT_DATA6 输出 (VDC2)	—	—
	PI2 输入/输出 (端口)	LCD_DATA5 输出 (LCDC)	DB5 输出 (VDC2)	BT_DATA5 输出 (VDC2)	—	—
	PI1 输入/输出 (端口)	LCD_DATA4 输出 (LCDC)	DB4 输出 (VDC2)	BT_DATA4 输出 (VDC2)	—	—
	PI0 输入/输出 (端口)	PI0 输入/输出 (端口)	COM/CDE 输出 (VDC2)	—	—	—
J	PJ7 输入/输出 (端口)	PJ7 输入/输出 (端口)	—	—	IDED10_M 输入/输出 (ATAPI)	—
	PJ6 输入/输出 (端口)	PJ6 输入/输出 (端口)	—	—	IDED5_M 输入/输出 (ATAPI)	—
	PJ5 输入/输出 (端口)	PJ5 输入/输出 (端口)	—	—	IDED9_M 输入/ 输出 (ATAPI)	—
	PJ4 输入/输出 (端口)	PJ4 输入/输出 (端口)	—	—	IDED6_M 输入/ 输出 (ATAPI)	—
	PJ3 输入/输出 (端口)	PJ3 输入/输出 (端口)	—	—	IDED7_M 输入/输出 (ATAPI)	—
J	PJ2 输入/输出 (端口)	PJ2 输入/输出 (端口)	—	—	IDED8_M 输入/输出 (ATAPI)	—
	PJ1 输入/输出 (端口)	PJ1 输入/输出 (端口)	—	—	IDERST_M 输出 (ATAPI)	—
	PJ0 输入/输出 (端口)	PJ0 输入/输出 (端口)	—	—	DIRECTION_M 输出 (ATAPI)	—

【注】 表中阴影部分的引脚是在复位后能立即使用的引脚。

表 27.2 能通过引脚选择寄存器控制的多路复用引脚一览表

寄存器名	位名	引脚功能 1	引脚功能 2	引脚功能 3	引脚功能 4
PTSEL_K	PTSEL_K7[1:0]	WDTOVF 输出 (SYSTEM)	IRQ1 输入 (INT)	AUDCK 输出 (AUD)	DACK1 输出 (DMAC)
	PTSEL_K6[1:0]	SCK0 输入 / 输出 (SCIF)	AUDSYNC 输出 (AUD)	FCLE 输出 (FLCTL)	—
	PTSEL_K5	SCK1 输入 / 输出 (SCIF)	FR/B 输入 (FLCTL)	—	—
	PTSEL_K4[1:0]	LCD_DATA0 输出 (LCDC)	DB0 输出 (VDC2)	BT_DATA0 输出 (VDC2)	—
	PTSEL_K3[1:0]	LCD_CL1 输出 (LCDC)	HSYNC/SPL* 输入 / 输出 (VDC2)	BT_HSYNC 输出 (VDC2)	—
	PTSEL_K2[1:0]	LCD_CLK 输入 (LCDC)	DCLKIN 输入 (VDC2)	—	—
	PTSEL_K1[1:0]	LCD_FLM 输出 (LCDC)	VSYNC/SPS* 输入 / 输出 (VDC2)	BT_VSYNC 输出 (VDC2)	—
	PTSEL_K0[1:0]	LCD_M_DISP 输出 (LCDC)	DE_H/DE_C 输出 (VDC2)	BT_DE_C 输出 (VDC2)	—
PTSEL_P	PTSEL_P11	RXD0 输入 (SCIF)	AUDATA0 输出 (AUD)	—	—
	PTSEL_P10	TXD0 输出 (SCIF)	AUDATA1 输出 (AUD)	—	—
	PTSEL_P9	RXD1 输入 (SCIF)	AUDATA2 输出 (AUD)	—	—
	PTSEL_P8	TXD1 输出 (SCIF)	AUDATA3 输出 (AUD)	—	—
PTSEL_R	PTSEL_R15	D63 输入 / 输出 (DATA)	IDED1 输入 / 输出 (ATAPI)	—	—
	PTSEL_R14	D62 输入 / 输出 (DATA)	IDED0 输入 / 输出 (ATAPI)	—	—
	PTSEL_R13	D61 输入 / 输出 (DATA)	IDED3 输入 / 输出 (ATAPI)	—	—
	PTSEL_R12	D60 输入 / 输出 (DATA)	IDED2 输入 / 输出 (ATAPI)	—	—
	PTSEL_R11	D59 输入 / 输出 (DATA)	IDED5 输入 / 输出 (ATAPI)	—	—
	PTSEL_R10	D58 输入 / 输出 (DATA)	IDED4 输入 / 输出 (ATAPI)	—	—
	PTSEL_R9	D57 输入 / 输出 (DATA)	IDED7 输入 / 输出 (ATAPI)	—	—
	PTSEL_R8	D56 输入 / 输出 (DATA)	IDED6 输入 / 输出 (ATAPI)	—	—
	PTSEL_R7	D55 输入 / 输出 (DATA)	DIRECTION 输出 (ATAPI)	—	—
	PTSEL_R6	D54 输入 / 输出 (DATA)	IDERST 输出 (ATAPI)	—	—

寄存器名	位名	引脚功能 1	引脚功能 2	引脚功能 3	引脚功能 4
PTSEL_R	PTSEL_R5	D53 输入 / 输出 (DATA)	IDED8 输入 / 输出 (ATAPI)	—	—
	PTSEL_R4	D52 输入 / 输出 (DATA)	IDED9 输入 / 输出 (ATAPI)	—	—
	PTSEL_R3	D51 输入 / 输出 (DATA)	IDED10 输入 / 输出 (ATAPI)	—	—
	PTSEL_R2	D50 输入 / 输出 (DATA)	IDED11 输入 / 输出 (ATAPI)	—	—
	PTSEL_R1	D49 输入 / 输出 (DATA)	IDED12 输入 / 输出 (ATAPI)	—	—
	PTSEL_R0	D48 输入 / 输出 (DATA)	IDED13 输入 / 输出 (ATAPI)	—	—
PTSEL_S	PTSEL_S15	IRQ0 输入 (INT)	DTEND1 输出 (DMAC)	—	—
	PTSEL_S14	IRQOUT 输出 (INT)	DREQ1 输入 (DMAC)	—	—
	PTSEL_S13	D47 输入 / 输出 (DATA)	IDECS0 输出 (ATAPI)	—	—
	PTSEL_S12	D46 输入 / 输出 (DATA)	IDECS1 输出 (ATAPI)	—	—
	PTSEL_S11	D45 输入 / 输出 (DATA)	IODACK 输出 (ATAPI)	—	—
	PTSEL_S10	D44 输入 / 输出 (DATA)	IDEINT 输入 (ATAPI)	—	—
	PTSEL_S9	D43 输入 / 输出 (DATA)	IDEIORDY 输入 (ATAPI)	—	—
	PTSEL_S8	D42 输入 / 输出 (DATA)	IDEIORD 输出 (ATAPI)	—	—
	PTSEL_S7	D41 输入 / 输出 (DATA)	IODREQ 输入 (ATAPI)	—	—
	PTSEL_S6	D40 输入 / 输出 (DATA)	IDEIOWR 输出 (ATAPI)	—	—
	PTSEL_S5	D39 输入 / 输出 (DATA)	IDED14 输入 / 输出 (ATAPI)	—	—
	PTSEL_S4	D38 输入 / 输出 (DATA)	IDED1 输入 / 输出 (ATAPI)	—	—
	PTSEL_S3	D37 输入 / 输出 (DATA)	IDEA1 输出 (ATAPI)	—	—
	PTSEL_S2	D36 输入 / 输出 (DATA)	IDEA2 输出 (ATAPI)	—	—
	PTSEL_S1	D35 输入 / 输出 (DATA)	IDEA0 输出 (ATAPI)	—	—

【注】 表中阴影部分的引脚是在复位后能立即使用的引脚。

* 此功能通过特殊选择寄存器进行输入 / 输出功能的转换。

27.2 寄存器说明

GPIO 的寄存器结构如表 27.3 所示，各处理模式的寄存器状态如表 27.4 所示。

表 27.3 寄存器结构

名称	略称	R/W	P4 区地址 *1	区域 7 地址 *1	存取长度 *2
端口 A 控制寄存器	PTIO_A	R/W	H'FFF1 0000	H'1FF1 0000	16
端口 B 控制寄存器	PTIO_B	R/W	H'FFF1 0004	H'1FF1 0004	16
端口 C 控制寄存器	PTIO_C	R/W	H'FFF1 0008	H'1FF1 0008	16
端口 D 控制寄存器	PTIO_D	R/W	H'FFF1 000C	H'1FF1 000C	16
端口 E 控制寄存器	PTIO_E	R/W	H'FFF1 0010	H'1FF1 0010	16
端口 F 控制寄存器	PTIO_F	R/W	H'FFF1 0014	H'1FF1 0014	16
端口 G 控制寄存器	PTIO_G	R/W	H'FFF1 0018	H'1FF1 0018	16
端口 H 控制寄存器	PTIO_H	R/W	H'FFF1 001C	H'1FF1 001C	16
端口 I 控制寄存器	PTIO_I	R/W	H'FFF1 0020	H'1FF1 0020	16
端口 J 控制寄存器	PTIO_J	R/W	H'FFF1 0024	H'1FF1 0024	16
端口 A 数据寄存器	PTDAT_A	R/W	H'FFF1 0040	H'1FF1 0040	16
端口 B 数据寄存器	PTDAT_B	R/W	H'FFF1 0044	H'1FF1 0044	16
端口 C 数据寄存器	PTDAT_C	R/W	H'FFF1 0048	H'1FF1 0048	16
端口 D 数据寄存器	PTDAT_D	R/W	H'FFF1 004C	H'1FF1 004C	16
端口 E 数据寄存器	PTDAT_E	R/W	H'FFF1 0050	H'1FF1 0050	16
端口 F 数据寄存器	PTDAT_F	R/W	H'FFF1 0054	H'1FF1 0054	16
端口 G 数据寄存器	PTDAT_G	R/W	H'FFF1 0058	H'1FF1 0058	16
端口 H 数据寄存器	PTDAT_H	R/W	H'FFF1 005C	H'1FF1 005C	16
端口 I 数据寄存器	PTDAT_I	R/W	H'FFF1 0060	H'1FF1 0060	16
端口 J 数据寄存器	PTDAT_J	R/W	H'FFF1 0064	H'1FF1 0064	16
输入引脚上拉控制寄存器	PTPUL_SPCL	R/W	H'FFF1 00E0	H'1FF1 00E0	16
引脚选择寄存器 A	PTSEL_A	R/W	H'FFF1 0080	H'1FF1 0080	16
引脚选择寄存器 B	PTSEL_B	R/W	H'FFF1 0084	H'1FF1 0084	16
引脚选择寄存器 C	PTSEL_C	R/W	H'FFF1 0088	H'1FF1 0088	16
引脚选择寄存器 D	PTSEL_D	R/W	H'FFF1 008C	H'1FF1 008C	16
引脚选择寄存器 E	PTSEL_E	R/W	H'FFF1 0090	H'1FF1 0090	16
引脚选择寄存器 F	PTSEL_F	R/W	H'FFF1 0094	H'1FF1 0094	16
引脚选择寄存器 G	PTSEL_G	R/W	H'FFF1 0098	H'1FF1 0098	16
引脚选择寄存器 H	PTSEL_H	R/W	H'FFF1 009C	H'1FF1 009C	16
引脚选择寄存器 I	PTSEL_I	R/W	H'FFF1 00A0	H'1FF1 00A0	16
引脚选择寄存器 J	PTSEL_J	R/W	H'FFF1 00A4	H'1FF1 00A4	16
引脚选择寄存器 K	PTSEL_K	R/W	H'FFF1 00A8	H'1FF1 00A8	16
引脚选择寄存器 P	PTSEL_P	R/W	H'FFF1 00AC	H'1FF1 00AC	16
引脚选择寄存器 R	PTSEL_R	R/W	H'FFF1 00B0	H'1FF1 00B0	16
引脚选择寄存器 S	PTSEL_S	R/W	H'FFF1 00B4	H'1FF1 00B4	16
Hi-Z 寄存器 A	PTHIZ_A	R/W	H'FFF1 00E8	H'1FF1 00E8	16
Hi-Z 寄存器 B	PTHIZ_B	R/W	H'FFF1 00EC	H'1FF1 00EC	16
特殊选择寄存器	PTSEL-SPCL	R/W	H'FFF1 00F0	H'1FF1 00F0	16

【注】 *1 P4 区地址是使用虚拟地址空间 P4 区的地址，区域 7 地址是使用 TLB 存取物理地址空间区域 7 的地址。

*2 只能对寄存器进行 16 位存取。必须以指定的存取长度进行读写。

表 27.4 各处理模式的寄存器状态

名称	略称	上电复位	睡眠	待机
端口 A 控制寄存器	PTIO_A	H'0000	保持	保持
端口 B 控制寄存器	PTIO_B	H'0000	保持	保持
端口 C 控制寄存器	PTIO_C	H'0000	保持	保持
端口 D 控制寄存器	PTIO_D	H'0000	保持	保持
端口 E 控制寄存器	PTIO_E	H'0000	保持	保持
端口 F 控制寄存器	PTIO_F	H'0000	保持	保持
端口 G 控制寄存器	PTIO_G	H'0000	保持	保持
端口 H 控制寄存器	PTIO_H	H'0000	保持	保持
端口 I 控制寄存器	PTIO_I	H'0002	保持	保持
端口 J 控制寄存器	PTIO_J	H'AAAA	保持	保持
端口 A 数据寄存器	PTDAT_A	H'0000	保持	保持
端口 B 数据寄存器	PTDAT_B	H'0000	保持	保持
端口 C 数据寄存器	PTDAT_C	H'0000	保持	保持
端口 D 数据寄存器	PTDAT_D	H'0000	保持	保持
端口 E 数据寄存器	PTDAT_E	H'0000	保持	保持
端口 F 数据寄存器	PTDAT_F	H'0000	保持	保持
端口 G 数据寄存器	PTDAT_G	H'0000	保持	保持
端口 H 数据寄存器	PTDAT_H	H'0000	保持	保持
端口 I 数据寄存器	PTDAT_I	H'0000	保持	保持
端口 J 数据寄存器	PTDAT_J	H'0000	保持	保持
输入引脚上拉控制寄存器	PTPUL_SPCL	H'0000	保持	保持
引脚选择寄存器 A	PTSEL_A	H'0000	保持	保持
引脚选择寄存器 B	PTSEL_B	H'0000	保持	保持
引脚选择寄存器 C	PTSEL_C	H'0000	保持	保持
引脚选择寄存器 D	PTSEL_D	H'0000	保持	保持
引脚选择寄存器 E	PTSEL_E	H'0000	保持	保持
引脚选择寄存器 F	PTSEL_F	H'0000	保持	保持
引脚选择寄存器 G	PTSEL_G	H'0000	保持	保持
引脚选择寄存器 H	PTSEL_H	H'0000	保持	保持
引脚选择寄存器 I	PTSEL_I	H'0000	保持	保持
引脚选择寄存器 J	PTSEL_J	H'0000	保持	保持
引脚选择寄存器 K	PTSEL_K	H'0000	保持	保持
引脚选择寄存器 P	PTSEL_P	H'0000	保持	保持
引脚选择寄存器 R	PTSEL_R	H'0000	保持	保持
引脚选择寄存器 S	PTSEL_S	H'0000	保持	保持
Hi-Z 寄存器 A	PTHIZ_A	H'0000	保持	保持
Hi-Z 寄存器 B	PTHIZ_B	H'0000	保持	保持
特殊选择寄存器	PTSEL_SPCL	H'0000	保持	保持

27.2.1 端口 A 控制寄存器 (PTIO_A)

PTIO_A 是 16 位可读写寄存器，选择引脚功能和输入上拉 MOS 控制。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTIO_A7[1:0]	PTIO_A6[1:0]	PTIO_A5[1:0]	PTIO_A4[1:0]	PTIO_A3[1:0]	PTIO_A2[1:0]	PTIO_A1[1:0]	PTIO_A0[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

位	位名	初始值	R/W	说明
15、14	PTIO_A7[1:0]	00	R/W	PTA7 模式位 00: 其他功能 (STATUS1 和 $\overline{\text{RTS2}}$) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
13、12	PTIO_A6[1:0]	00	R/W	PTA6 模式位 00: 其他功能 (STATUS0 和 $\overline{\text{CTS2}}$) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
11、10	PTIO_A5[1:0]	00	R/W	PTA5 模式位 00: 其他功能 ($\overline{\text{FCE}}$) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
9、8	PTIO_A4[1:0]	00	R/W	PTA4 模式位 00: 其他功能 ($\overline{\text{FRE}}$) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
7、6	PTIO_A3[1:0]	00	R/W	PTA3 模式位 00: 其他功能 ($\overline{\text{FWE}}$) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
5、4	PTIO_A2[1:0]	00	R/W	PTA2 模式位 00: 其他功能 (TxD2) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
3、2	PTIO_A1[1:0]	00	R/W	PTA1 模式位 00: 其他功能 (Rx2D) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
1、0	PTIO_A0[1:0]	00	R/W	PTA0 模式位 00: 其他功能 (SCK2) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)

27.2.2 端口 B 控制寄存器 (PTIO_B)

PTIO_B 是 16 位可读写寄存器，选择引脚功能和输入上拉 MOS 控制。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTIO_B7[1:0]		PTIO_B6[1:0]		PTIO_B5[1:0]		PTIO_B4[1:0]		PTIO_B3[1:0]		PTIO_B2[1:0]		PTIO_B1[1:0]		PTIO_B0[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

位	位名	初始值	R/W	说明
15、14	PTIO_B7[1:0]	00	R/W	PTB7 模式位 00: 其他功能 (A25、 $\overline{\text{DREQ0}}$ 和 $\overline{\text{RTS0}}$) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
13、12	PTIO_B6[1:0]	00	R/W	PTB6 模式位 00: 其他功能 (A24、 $\overline{\text{DACK0}}$ 和 $\overline{\text{CTS0}}$) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
11、10	PTIO_B5[1:0]	00	R/W	PTB5 模式位 00: 其他功能 (A23、 $\overline{\text{DTEND0}}$ 和 $\overline{\text{RTS1}}$) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
9、8	PTIO_B4[1:0]	00	R/W	PTB4 模式位 00: 其他功能 (A22 和 $\overline{\text{CTS1}}$) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
7、6	PTIO_B3[1:0]	00	R/W	PTB3 模式位 00: 其他功能 (A21) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
5、4	PTIO_B2[1:0]	00	R/W	PTB2 模式位 00: 其他功能 (A20) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
3、2	PTIO_B1[1:0]	00	R/W	PTB1 模式位 00: 其他功能 (A19) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
1、0	PTIO_B0[1:0]	00	R/W	PTB0 模式位 00: 其他功能 (A18) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)

27.2.3 端口 C 控制寄存器 (PTIO_C)

PTIO_C 是 16 位可读写寄存器，选择引脚功能和输入上拉 MOS 控制。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTIO_C7[1:0]		PTIO_C6[1:0]		PTIO_C5[1:0]		PTIO_C4[1:0]		PTIO_C3[1:0]		PTIO_C2[1:0]		PTIO_C1[1:0]		PTIO_C0[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

位	位名	初始值	R/W	说明
15、14	PTIO_C7[1:0]	00	R/W	PTC7 模式位 00: 其他功能 (AUDIO_CLK0) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
13、12	PTIO_C6[1:0]	00	R/W	PTC6 模式位 00: 其他功能 (AUDIO_CLK1) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
11、10	PTIO_C5[1:0]	00	R/W	PTC5 模式位 00: 其他功能 (AUDIO_CLK2) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
9、8	PTIO_C4[1:0]	00	R/W	PTC4 模式位 00: 其他功能 (SSIWS2) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
7、6	PTIO_C3[1:0]	00	R/W	PTC3 模式位 00: 其他功能 (SSISCK2) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
5、4	PTIO_C2[1:0]	00	R/W	PTC2 模式位 00: 其他功能 (SSIDATA2) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
3、2	PTIO_C1[1:0]	00	R/W	PTC1 模式位 00: 其他功能 ($\overline{\text{ASEBRKAK}}/\text{BRKACK}, \text{TCLK}$) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
1、0	PTIO_C0[1:0]	00	R/W	PTC0 模式位 00: 其他功能 (FALE) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)

27.2.4 端口 D 控制寄存器 (PTIO_D)

PTIO_D 是 16 位可读写寄存器，选择引脚功能和输入上拉 MOS 控制。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTIO_D7[1:0]		PTIO_D6[1:0]		PTIO_D5[1:0]		PTIO_D4[1:0]		PTIO_D3[1:0]		PTIO_D2[1:0]		PTIO_D1[1:0]		PTIO_D0[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

位	位名	初始值	R/W	说明
15、14	PTIO_D7[1:0]	00	R/W	PTD7 模式位 00: 其他功能 (CRS 和 IDEA1_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
13、12	PTIO_D6[1:0]	00	R/W	PTD6 模式位 00: 其他功能 (TX_ER 和 IDEIOWR_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
11、10	PTIO_D5[1:0]	00	R/W	PTD5 模式位 00: 其他功能 (TX_CLK 和 IDED15_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
9、8	PTIO_D4[1:0]	00	R/W	PTD4 模式位 00: 其他功能 (TX_EN 和 IDED0_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
7、6	PTIO_D3[1:0]	00	R/W	PTD3 模式位 00: 其他功能 (MII_TXD0、SSISCK5 和 IDEIORDY_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
5、4	PTIO_D2[1:0]	00	R/W	PTD2 模式位 00: 其他功能 (MII_TXD1、SSIWS5 和 IDEIORD_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
3、2	PTIO_D1[1:0]	00	R/W	PTD1 模式位 00: 其他功能 (MII_TXD2、AUDIO_CLK5 和 IDEINT_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
1、0	PTIO_D0[1:0]	00	R/W	PTD0 模式位 00: 其他功能 (MII_TXD3、SSIDATA5 和 IODACK_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)

27.2.5 端口 E 控制寄存器 (PTIO_E)

PTIO_E 是 16 位可读写寄存器，选择引脚功能和输入上拉 MOS 控制。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTIO_E7[1:0]		PTIO_E6[1:0]		PTIO_E5[1:0]		PTIO_E4[1:0]		PTIO_E3[1:0]		PTIO_E2[1:0]		PTIO_E1[1:0]		PTIO_E0[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

位	位名	初始值	R/W	说明
15、14	PTIO_E7[1:0]	00	R/W	PTE7 模式位 00: 其他功能 (COL 和 IDEA2_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
13、12	PTIO_E6[1:0]	00	R/W	PTE6 模式位 00: 其他功能 (RX_ER 和 IODREQ_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
11、10	PTIO_E5[1:0]	00	R/W	PTE5 模式位 00: 其他功能 (RX_CLK 和 IDED1_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
9、8	PTIO_E4[1:0]	00	R/W	PTE4 模式位 00: 其他功能 (RX_DV 和 IDED14_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
7、6	PTIO_E3[1:0]	00	R/W	PTE3 模式位 00: 其他功能 (MII_RXD0、SSIWS4 和 IDED2_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
5、4	PTIO_E2[1:0]	00	R/W	PTE2 模式位 00: 其他功能 (MII_RXD1、SSISCK4 和 IDED13_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
3、2	PTIO_E1[1:0]	00	R/W	PTE1 模式位 00: 其他功能 (MII_RXD2、SSIDATA4 和 IDED3_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF): 11: 端口输入 (上拉 MOS: ON)
1、0	PTIO_E0[1:0]	00	R/W	PTE0 模式位 00: 其他功能 (MII_RXD3、AUDIO_CLK4 和 IDED12_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)

27.2.6 端口 F 控制寄存器 (PTIO_F)

PTIO_F 是 16 位可读写寄存器，选择引脚功能和输入上拉 MOS 控制。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTIO_F7[1:0]	PTIO_F6[1:0]	PTIO_F5[1:0]	PTIO_F4[1:0]	PTIO_F3[1:0]	PTIO_F2[1:0]	PTIO_F1[1:0]	PTIO_F0[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

位	位名	初始值	R/W	说明
15、14	PTIO_F7[1:0]	00	R/W	PTF7 模式位 00: 其他功能 (D32) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
13、12	PTIO_F6[1:0]	00	R/W	PTF6 模式位 00: 其他功能 (D33) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
11、10	PTIO_F5[1:0]	00	R/W	PTF5 模式位 00: 其他功能 (D34) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
9、8	PTIO_F4[1:0]	00	R/W	PTF4 模式位 00: 其他功能 (EXOUT 和 $\overline{\text{IDECS1_M}}$) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
7、6	PTIO_F3[1:0]	00	R/W	PTF3 模式位 00: 其他功能 (LNKSTA 和 $\overline{\text{IDECS0_M}}$) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
5、4	PTIO_F2[1:0]	00	R/W	PTF2 模式位 00: 其他功能 (WOL 和 $\overline{\text{IDEA0_M}}$) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
3、2	PTIO_F1[1:0]	00	R/W	PTF1 模式位 00: 其他功能 (MDIO 和 $\overline{\text{IDED11_M}}$) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
1、0	PTIO_F0[1:0]	00	R/W	PTF0 模式位 00: 其他功能 (MDC 和 $\overline{\text{IDED4_M}}$) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)

27.2.7 端口 G 控制寄存器 (PTIO_G)

PTIO_G 是 16 位可读写寄存器，选择引脚功能和输入上拉 MOS 控制。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTIO_G7[1:0]		PTIO_G6[1:0]		PTIO_G5[1:0]		PTIO_G4[1:0]		PTIO_G3[1:0]		PTIO_G2[1:0]		PTIO_G1[1:0]		PTIO_G0[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

位	位名	初始值	R/W	说明
15、14	PTIO_G7[1:0]	00	R/W	PTG7 模式位 00: 其他功能 (LCD_DATA15 和 DR3) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
13、12	PTIO_G6[1:0]	00	R/W	PTG6 模式位 00: 其他功能 (LCD_DATA14 和 DR2) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
11、10	PTIO_G5[1:0]	00	R/W	PTG5 模式位 00: 其他功能 (LCD_DATA13 和 DR1) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
9、8	PTIO_G4[1:0]	00	R/W	PTG4 模式位 00: 其他功能 (LCD_DATA12 和 DR0) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
7、6	PTIO_G3[1:0]	00	R/W	PTG3 模式位 00: 其他功能 (LCD_DATA11 和 DG5) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
5、4	PTIO_G2[1:0]	00	R/W	PTG2 模式位 00: 其他功能 (LCD_DATA10 和 DG4) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
3、2	PTIO_G1[1:0]	00	R/W	PTG1 模式位 00: 其他功能 (LCD_DATA9 和 DG3) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
1、0	PTIO_G0[1:0]	00	R/W	PTG0 模式位 00: 其他功能 (LCD_DATA8 和 DG2) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)

27.2.8 端口 H 控制寄存器 (PTIO_H)

PTIO_H 是 16 位可读写寄存器，选择引脚功能和输入上拉 MOS 控制。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTIO_H7[1:0]	PTIO_H6[1:0]	PTIO_H5[1:0]	PTIO_H4[1:0]	PTIO_H3[1:0]	PTIO_H2[1:0]	PTIO_H1[1:0]	PTIO_H0[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

位	位名	初始值	R/W	说明
15、14	PTIO_H7[1:0]	00	R/W	PTH7 模式位 00: 其他功能 (AUDIO_CLK3) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
13、12	PTIO_H6[1:0]	00	R/W	PTH6 模式位 00: 其他功能 (SSIWS3) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
11、10	PTIO_H5[1:0]	00	R/W	PTH5 模式位 00: 其他功能 (SSISCK3) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
9、8	PTIO_H4[1:0]	00	R/W	PTH4 模式位 00: 其他功能 (SSIDATA3) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
7、6	PTIO_H3[1:0]	00	R/W	PTH3 模式位 00: 其他功能 (LCD_CL2 和 DE_V) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
5、4	PTIO_H2[1:0]	00	R/W	PTH2 模式位 00: 其他功能 (LCD_DON 和 DCLKOUT) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
3、2	PTIO_H1[1:0]	00	R/W	PTH1 模式位 00: 其他功能 (LCD_VCP_WC 和 DR4) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
1、0	PTIO_H0[1:0]	00	R/W	PTH0 模式位 00: 其他功能 (LCD_VEP_WC 和 DR5) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)

27.2.9 端口 I 控制寄存器 (PTIO_I)

PTIO_I 是 16 位可读写寄存器，选择引脚功能和输入上拉 MOS 控制。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PTIO_I4[1:0]	PTIO_I3[1:0]	PTIO_I2[1:0]	PTIO_I1[1:0]	PTIO_I0[1:0]					
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15~10	—	00	R	保留位 读取值为 0，只能写 0。
9、8	PTIO_I4[1:0]	00	R/W	PTI4 模式位 00: 其他功能 (LCD_DATA7、DG1 和 BT_DATA7) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
7、6	PTIO_I3[1:0]	00	R/W	PTI3 模式位 00: 其他功能 (LCD_DATA6、DG0 和 BT_DATA6) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
5、4	PTIO_I2[1:0]	00	R/W	PTI2 模式位 00: 其他功能 (LCD_DATA5、DB5 和 BT_DATA5) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
3、2	PTIO_I1[1:0]	00	R/W	PTI1 模式位 00: 其他功能 (LCD_DATA4、DB4 和 BT_DATA4) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
1、0	PTIO_I0[1:0]	10	R/W	PTI0 模式位 00: 其他功能 (COM/CDE) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)

27.2.10 端口 J 控制寄存器 (PTIO_J)

PTIO_J 是 16 位可读写寄存器，选择引脚功能和输入上拉 MOS 控制。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTIO_J7[1:0]		PTIO_J6[1:0]		PTIO_J5[1:0]		PTIO_J4[1:0]		PTIO_J3[1:0]		PTIO_J2[1:0]		PTIO_J1[1:0]		PTIO_J0[1:0]	
初始值:	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15、14	PTIO_J7[1:0]	10	R/W	PTJ7 模式位 00: 其他功能 (IDED10_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
13、12	PTIO_J6[1:0]	10	R/W	PTJ6 模式位 00: 其他功能 (IDED5_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
11、10	PTIO_J5[1:0]	10	R/W	PTJ5 模式位 00: 其他功能 (IDED9_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
9、8	PTIO_J4[1:0]	10	R/W	PTJ4 模式位 00: 其他功能 (IDED6_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
7、6	PTIO_J3[1:0]	10	R/W	PTJ3 模式位 00: 其他功能 (IDED7_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
5、4	PTIO_J2[1:0]	10	R/W	PTJ2 模式位 00: 其他功能 (IDED8_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
3、2	PTIO_J1[1:0]	10	R/W	PTJ1 模式位 00: 其他功能 ($\overline{\text{IDERST_M}}$) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)
1、0	PTIO_J0[1:0]	10	R/W	PTJ0 模式位 00: 其他功能 (DIRECTION_M) 01: 端口输出 10: 端口输入 (上拉 MOS: OFF) 11: 端口输入 (上拉 MOS: ON)

27.2.11 端口 A 数据寄存器 (PTDAT_A)

PTDAT_A 是 16 位可读写寄存器，保存端口 A 的数据。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PTDAT_A7	PTDAT_A6	PTDAT_A5	PTDAT_A4	PTDAT_A3	PTDAT_A2	PTDAT_A1	PTDAT_A0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 8	—	全 0	R	保留位 其读取值和低 8 位相同，只能写 0。
7	PTDAT_A7	0	R/W	保存用作通用输出端口的引脚输出数据。
6	PTDAT_A6	0	R/W	在引脚功能为通用输出端口时，如果读端口，就能读取对应此寄存器的值。
5	PTDAT_A5	0	R/W	在引脚功能为通用输入端口时，如果读此寄存器，就能读取对应引脚的状态。
4	PTDAT_A4	0	R/W	
3	PTDAT_A3	0	R/W	
2	PTDAT_A2	0	R/W	
1	PTDAT_A1	0	R/W	
0	PTDAT_A0	0	R/W	

27.2.12 端口 B 数据寄存器 (PTDAT_B)

PTDAT_B 是 16 位可读写寄存器，保存端口 B 的数据。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PTDAT_B7	PTDAT_B6	PTDAT_B5	PTDAT_B4	PTDAT_B3	PTDAT_B2	PTDAT_B1	PTDAT_B0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 8	—	全 0	R	保留位 其读取值和低 8 位相同，只能写 0。
7	PTDAT_B7	0	R/W	保存用作通用输出端口的引脚输出数据。
6	PTDAT_B6	0	R/W	在引脚功能为通用输出端口时，如果读端口，就能读取对应此寄存器的值。
5	PTDAT_B5	0	R/W	在引脚功能为通用输入端口时，如果读此寄存器，就能读取对应引脚的状态。
4	PTDAT_B4	0	R/W	在引脚功能为非通用端口时，输入端口的读取值为引脚的状态，写无效；输出端口的读取值为不定值，写无效。
3	PTDAT_B3	0	R/W	
2	PTDAT_B2	0	R/W	
1	PTDAT_B1	0	R/W	
0	PTDAT_B0	0	R/W	

27.2.13 端口 C 数据寄存器 (PTDAT_C)

PTDAT_C 是 16 位可读写寄存器，保存端口 C 的数据。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PTDAT_C7	PTDAT_C6	PTDAT_C5	PTDAT_C4	PTDAT_C3	PTDAT_C2	PTDAT_C1	PTDAT_C0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 8	—	全 0	R	保留位 其读取值和低 8 位相同，只能写 0。
7	PTDAT_C7	0	R/W	保存用作通用输出端口的引脚输出数据。
6	PTDAT_C6	0	R/W	在引脚功能为通用输出端口时，如果读端口，就能读取对应此寄存器的值。
5	PTDAT_C5	0	R/W	在引脚功能为通用输入端口时，如果读此寄存器，就能读取对应引脚的状态。
4	PTDAT_C4	0	R/W	在引脚功能为非通用端口时，输入端口的读取值为引脚的状态，写无效；输出端口的读取值为不定值，写无效。
3	PTDAT_C3	0	R/W	
2	PTDAT_C2	0	R/W	
1	PTDAT_C1	0	R/W	
0	PTDAT_C0	0	R/W	

27.2.14 端口 D 数据寄存器 (PTDAT_D)

PTDAT_D 是 16 位可读写寄存器，保存端口 D 的数据。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PTDAT_D7	PTDAT_D6	PTDAT_D5	PTDAT_D4	PTDAT_D3	PTDAT_D2	PTDAT_D1	PTDAT_D0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 8	—	全 0	R	保留位 其读取值和低 8 位相同，只能写 0。
7	PTDAT_D7	0	R/W	保存用作通用输出端口的引脚输出数据。
6	PTDAT_D6	0	R/W	在引脚功能为通用输出端口时，如果读端口，就能读对应此寄存器的值。
5	PTDAT_D5	0	R/W	在引脚功能为通用输入端口时，如果读此寄存器，就能读取对应引脚的状态。
4	PTDAT_D4	0	R/W	在引脚功能为非通用端口时，输入端口的读取值为引脚的状态，写无效；输出端口的读取值为不定值，写无效。
3	PTDAT_D3	0	R/W	
2	PTDAT_D2	0	R/W	
1	PTDAT_D1	0	R/W	
0	PTDAT_D0	0	R/W	

27.2.15 端口 E 数据寄存器 (PTDAT_E)

PTDAT_E 是 16 位可读写寄存器，保存端口 E 的数据。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PTDAT_E7	PTDAT_E6	PTDAT_E5	PTDAT_E4	PTDAT_E3	PTDAT_E2	PTDAT_E1	PTDAT_E0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 8	—	全 0	R	保留位 其读取值和低 8 位相同，只能写 0。
7	PTDAT_E7	0	R/W	保存用作通用输出端口的引脚输出数据。
6	PTDAT_E6	0	R/W	在引脚功能为通用输出端口时，如果读端口，就能读取对应此寄存器的值。
5	PTDAT_E5	0	R/W	
4	PTDAT_E4	0	R/W	在引脚功能为通用输入端口时，如果读此寄存器，就能读取对应引脚的状态。
3	PTDAT_E3	0	R/W	
2	PTDAT_E2	0	R/W	在引脚功能为非通用端口时，输入端口的读取值为引脚的状态，写无效；输出端口的读取值为不定值，写无效。
1	PTDAT_E1	0	R/W	
0	PTDAT_E0	0	R/W	

27.2.16 端口 F 数据寄存器 (PTDAT_F)

PTDAT_F 是 16 位可读写寄存器，保存端口 F 的数据。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PTDAT_F7	PTDAT_F6	PTDAT_F5	PTDAT_F4	PTDAT_F3	PTDAT_F2	PTDAT_F1	PTDAT_F0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 8	—	全 0	R	保留位 其读取值和低 8 位相同，只能写 0。
7	PTDAT_F7	0	R/W	保存用作通用输出端口的引脚输出数据。
6	PTDAT_F6	0	R/W	在引脚功能为通用输出端口时，如果读端口，就能读取对应此寄存器的值。
5	PTDAT_F5	0	R/W	
4	PTDAT_F4	0	R/W	在引脚功能为通用输入端口时，如果读此寄存器，就能读取对应引脚的状态。
3	PTDAT_F3	0	R/W	
2	PTDAT_F2	0	R/W	在引脚功能为非通用端口时，输入端口的读取值为引脚的状态，写无效；输出端口的读取值为不定值，写无效。
1	PTDAT_F1	0	R/W	
0	PTDAT_F0	0	R/W	

27.2.17 端口 G 数据寄存器 (PTDAT_G)

PTDAT_G 是 16 位可读写寄存器，保存端口 G 的数据。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PTDAT_G7	PTDAT_G6	PTDAT_G5	PTDAT_G4	PTDAT_G3	PTDAT_G2	PTDAT_G1	PTDAT_G0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 8	—	全 0	R	保留位 其读取值和低 8 位相同，只能写 0。
7	PTDAT_G7	0	R/W	保存用作通用输出端口的引脚输出数据。
6	PTDAT_G6	0	R/W	在引脚功能为通用输出端口时，如果读端口，就能读取对应此寄存器的值。
5	PTDAT_G5	0	R/W	
4	PTDAT_G4	0	R/W	在引脚功能为通用输入端口时，如果读此寄存器，就能读取对应引脚的状态。
3	PTDAT_G3	0	R/W	
2	PTDAT_G2	0	R/W	在引脚功能为非通用端口时，输入端口的读取值为引脚的状态，写无效；输出端口的读取值为不定值，写无效。
1	PTDAT_G1	0	R/W	
0	PTDAT_G0	0	R/W	

27.2.18 端口 H 数据寄存器 (PTDAT_H)

PTDAT_H 是 16 位可读写寄存器，保存端口 H 的数据。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PTDAT_H7	PTDAT_H6	PTDAT_H5	PTDAT_H4	PTDAT_H3	PTDAT_H2	PTDAT_H1	PTDAT_H0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 8	—	全 0	R	保留位 其读取值和低 8 位相同，只能写 0。
7	PTDAT_H7	0	R/W	保存用作通用输出端口的引脚输出数据。
6	PTDAT_H6	0	R/W	在引脚功能为通用输出端口时，如果读端口，就能读取对应此寄存器的值。
5	PTDAT_H5	0	R/W	
4	PTDAT_H4	0	R/W	在引脚功能为通用输入端口时，如果读此寄存器，就能读对应引脚的状态。
3	PTDAT_H3	0	R/W	
2	PTDAT_H2	0	R/W	在引脚功能为非通用端口时，输入端口的读取值为引脚的状态，写无效；输出端口的读取值为不定值，写无效。
1	PTDAT_H1	0	R/W	
0	PTDAT_H0	0	R/W	

27.2.19 端口 I 数据寄存器 (PTDAT_I)

PTDAT_I 是 16 位可读写寄存器，保存端口 I 的数据。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	PTDAT_I4	PTDAT_I3	PTDAT_I2	PTDAT_I1	PTDAT_I0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 8	—	全 0	R	保留位 其读取值和低 8 位相同，只能写 0。
7 ~ 5	—	全 0	R	保留位 读取值为 0，只能写 0。
4	PTDAT_I4	0	R/W	保存用作通用输出端口的引脚输出数据。
3	PTDAT_I3	0	R/W	在引脚功能为通用输出端口时，如果读端口，就能读取对应此寄存器的值。 在引脚功能为通用输入端口时，如果读此寄存器，就能读取对应引脚的状态。 在引脚功能为非通用端口时，输入端口的读取值为引脚的状态，写无效；输出端口的读取值为不定值，写无效。
2	PTDAT_I2	0	R/W	
1	PTDAT_I1	0	R/W	
0	PTDAT_I0	0	R/W	

27.2.20 端口 J 数据寄存器 (PTDAT_J)

PTDAT_J 是 16 位可读写寄存器，保存端口 J 的数据。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
	—	—	—	—	—	—	—	—	—	—	—	PTDAT_J7	PTDAT_J6	PTDAT_J5	PTDAT_J4	PTDAT_J3	PTDAT_J2	PTDAT_J1	PTDAT_J0
初始值:	0	0	0	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/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15 ~ 8	—	全 0	R	保留位 其读取值和低 8 位相同，只能写 0。
7	PTDAT_J7	0	R/W	保存用作通用输出端口的引脚输出数据。 在引脚功能为通用输出端口时，如果读端口，就能读取对应此寄存器的值。 在引脚功能为通用输入端口时，如果读此寄存器，就能读取对应引脚的状态。 在引脚功能为非通用端口时，输入端口的读取值为引脚的状态，写无效；输出端口的读取值为不定值，写无效。
6	PTDAT_J6	0	R/W	
5	PTDAT_J5	0	R/W	
4	PTDAT_J4	0	R/W	
3	PTDAT_J3	0	R/W	
2	PTDAT_J2	0	R/W	
1	PTDAT_J1	0	R/W	
0	PTDAT_J0	0	R/W	

27.2.21 输入引脚上拉控制寄存器 (PTPUL_SPCL)

PTPUL_SPCL 是 16 位可读写寄存器。按位控制各自对应的引脚上拉。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PTPUL_	PTPUL_	—	—	—	—	—	—	—	—	—	—	—	—	—
		IRQ1	IRQ0													
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读取值为 0，只能写 0。
14	PTPUL_IRQ1	0	R/W	能设定 IRQ1 引脚的上拉控制。 0: IRQ1 引脚上拉 OFF 1: IRQ1 引脚上拉 ON
13	PTPUL_IRQ0	0	R/W	能设定 IRQ0 引脚的上拉控制。 0: IRQ0 引脚上拉 OFF 1: IRQ0 引脚上拉 ON
12 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

27.2.22 引脚选择寄存器 0 (PTSEL_A)

PTSEL_A 是 16 位可读写寄存器，选择端口 A (PA) 的“其他功能”和多路复用引脚的功能。在使用“其他功能”时，必须在设定 PSEL0 后将端口控制寄存器设定为“其他功能”。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTSEL_A7[1:0]		PTSEL_A6[1:0]		—	PTSEL_A5	—	PTSEL_A4	—	PTSEL_A3	—	PTSEL_A2	—	PTSEL_A1	—	PTSEL_A0
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W

位	位名	初始值	R/W	说明
15、14	PTSEL_A7[1:0]	00	R/W	选择端口 A (PA7) 的功能。 00: STATUS1 01: 选择 $\overline{\text{RTS2}}$ 功能 10: 选择 PA7 功能 11: 禁止设定
13、12	PTSEL_A6[1:0]	00	R/W	选择端口 A (PA6) 的功能。 00: STATUS0 01: 选择 $\overline{\text{CTS2}}$ 功能 10: 选择 PA6 功能 11: 禁止设定
11	—	0	R	保留位 读取值为 0，只能写 0。
10	PTSEL_A5	0	R/W	选择端口 A (PA5) 的功能。 0: 选择 $\overline{\text{FCE}}$ 功能 1: 选择 PA5 功能
9	—	0	R	保留位 读取值为 0，只能写 0。
8	PTSEL_A4	0	R/W	选择端口 A (PA4) 的功能。 0: 选择 $\overline{\text{FRE}}$ 功能 1: 选择 PA4 功能
7	—	0	R	保留位 读取值为 0，只能写 0。
6	PTSEL_A3	0	R/W	选择端口 A (PA3) 的功能。 0: 选择 $\overline{\text{FWE}}$ 功能 1: 选择 PA3 功能
5	—	0	R	保留位 读取值为 0，只能写 0。
4	PTSEL_A2	0	R/W	选择端口 A (PA2) 的功能。 0: 选择 TXD2 功能 1: 选择 PA2 功能
3	—	0	R	保留位 读取值为 0，只能写 0。
2	PTSEL_A1	0	R/W	选择端口 A (PA1) 的功能。 0: 选择 RXD2 功能 1: 选择 PA1 功能
1	—	0	R	保留位 读取值为 0，只能写 0。
0	PTSEL_A0	0	R/W	选择端口 A (PA0) 的功能。 0: 选择 SCK2 功能 1: 选择 PA0 功能

27.2.23 引脚选择寄存器 1 (PTSEL_B)

PTSEL_B 是 16 位可读写寄存器。选择端口 B (PB) 的“其他功能”和多路复用引脚的功能。在使用“其他功能”时，必须在设定 PTSEL_B 后将端口控制寄存器设定为“其他功能”。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTSEL_B7[1:0]	PTSEL_B6[1:0]	PTSEL_B5[1:0]	PTSEL_B4[1:0]	—	PTSEL_B3	—	PTSEL_B2	—	PTSEL_B1	—	PTSEL_B0				
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	R/W	R	R/W	R	R/W	R	R	R/W

位	位名	初始值	R/W	说明
15、14	PTSEL_B7[1:0]	00	R	选择端口 B (PB7) 的功能。 00: 选择 A25 功能 01: 选择 PB7 功能 10: 选择 $\overline{DREQ0}$ 功能 11: 选择 $\overline{RTS0}$ 功能
13、12	PTSEL_B6[1:0]	00	R/W	选择端口 B (PB6) 的功能。 00: 选择 A24 功能 01: 选择 PB6 功能 10: 选择 $\overline{DACK0}$ 功能 11: 选择 $\overline{CTS0}$ 功能
11、10	PTSEL_B5[1:0]	00	R/W	选择端口 B (PB5) 的功能。 00: 选择 A23 功能 01: 选择 PB5 功能 10: 选择 $\overline{DTEND0}$ 功能 11: 选择 $\overline{RTS1}$ 功能
9、8	PTSEL_B4[1:0]	00	R/W	选择端口 B (PB4) 的功能。 00: 选择 A22 功能 01: 选择 PB4 功能 10: 选择 $\overline{CTS1}$ 功能 11: 禁止设定
7	—	0	R	保留位 读取值为 0, 只能写 0。
6	PTSEL_B3	0	R/W	选择端口 B (PB3) 的功能。 0: 选择 A21 功能 1: 选择 PB3 功能
5	—	0	R	保留位 读取值为 0, 只能写 0。
4	PTSEL_B2	全 0	R/W	选择端口 B (PB2) 的功能。 0: 选择 A20 功能 1: 选择 PB2 功能
3	—	0	R	保留位 读取值为 0, 只能写 0。
2	PTSEL_B1	全 0	R/W	选择端口 B (PB1) 的功能。 0: 选择 A19 功能 1: 选择 PB1 功能
1	—	0	R	保留位 读取值为 0, 只能写 0。
0	PTSEL_B0	全 0	R/W	选择端口 B (PB0) 的功能。 0: 选择 A18 功能 1: 选择 PB0 功能

27.2.24 引脚选择寄存器 2 (PTSEL_C)

PTSEL_C 是 16 位可读写寄存器，选择多路复用引脚的功能。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PTSEL_C7	—	PTSEL_C6	—	PTSEL_C5	—	PTSEL_C4	—	PTSEL_C3	—	PTSEL_C2	PTSEL_C1[1:0]	—	PTSEL_C0	
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R/W	R/W	R	R/W

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读取值为 0，只能写 0。
14	PTSEL_C7	0	R/W	选择端口 C (PC7) 的功能。 0: 选择 AUDIO_CLK0 功能 1: 选择 PC7 功能
13	—	0	R	保留位 读取值为 0，只能写 0。
12	PTSEL_C6	0	R/W	选择端口 C (PC6) 的功能。 0: 选择 AUDIO_CLK1 功能 1: 选择 PC6 功能
11	—	0	R	保留位 读取值为 0，只能写 0。
10	PTSEL_C5	0	R/W	选择端口 C (PC5) 的功能。 0: 选择 AUDIO_CLK2 功能 1: 选择 PC5 功能
9	—	0	R	保留位 读取值为 0，只能写 0。
8	PTSEL_C4	0	R/W	选择端口 C (PC4) 的功能。 0: 选择 SSIWS2 功能 1: 选择 PC4 功能
7	—	0	R	保留位 读取值为 0，只能写 0。
6	PTSEL_C3	0	R/W	选择端口 C (PC3) 的功能。 0: 选择 SSISCK2 功能 1: 选择 PC3 功能
5	—	0	R	保留位 读取值为 0，只能写 0。
4	PTSEL_C2	0	R/W	选择端口 C (PC2) 的功能。 0: 选择 SSIDATA2 功能 1: 选择 PC2 功能
3、2	PTSEL_C1[1:0]	00	R/W	选择端口 C (PC1) 的功能。 00: 选择 ASEBRK/BRKACK 功能 01: 选择 TCLK 功能 10: 选择 PC1 功能 00: 禁止设定
1	—	0	R	保留位 读取值为 0，只能写 0。
0	PTSEL_C0	0	R/W	选择端口 C (PC0) 的功能。 0: 选择 FALE 功能 1: 选择 PC0 功能

27.2.25 引脚选择寄存器 3 (PTSEL_D)

PTSEL_D 是 16 位可读写寄存器，选择多路复用引脚的功能。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTSEL_D7[1:0]		PTSEL_D6[1:0]		PTSEL_D5[1:0]		PTSEL_D4[1:0]		PTSEL_D3[1:0]		PTSEL_D2[1:0]		PTSEL_D1[1:0]		PTSEL_D0[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

位	位名	初始值	R/W	说明
15、14	PTSEL_D7[1:0]	00	R/W	选择端口 D (PD7) 的功能。 00: 选择 CRS 功能 01: 选择 PD7 功能 10: 选择 IDEA1_M 功能 11: 禁止设定
13、12	PTSEL_D6[1:0]	00	R/W	选择端口 D (PD6) 的功能。 00: 选择 TX_ER 功能 01: 选择 PD6 功能 10: 选择 IDEIOWR_M 功能 11: 禁止设定
11、10	PTSEL_D5[1:0]	00	R/W	选择端口 D (PD5) 的功能。 00: 选择 TX_CLK 功能 01: 选择 PD5 功能 10: 选择 IDE15_M 功能 11: 禁止设定
9、8	PTSEL_D4[1:0]	00	R/W	选择端口 D (PD4) 的功能。 00: 选择 TX_EN 功能 01: 选择 PD4 功能 10: 选择 IDE0_M 功能 11: 禁止设定
7、6	PTSEL_D3[1:0]	00	R/W	选择端口 D (PD3) 的功能。 00: 选择 MII_TXD0 功能 01: 选择 SSISCK5 功能 10: 选择 IDEIORDY_M 功能 11: 选择 PD3 功能
5、4	PTSEL_D2[1:0]	00	R/W	选择端口 D (PD2) 的功能。 00: 选择 MII_TXD1 功能 01: 选择 SSIWS5 功能 10: 选择 IDEIORD_M 功能 11: 选择 PD2 功能
3、2	PTSEL_D1[1:0]	00	R/W	选择端口 D (PD1) 的功能。 00: 选择 MII_TXD2 功能 01: 选择 AUDIO_CLK5 功能 10: 选择 IDEINT_M 功能 11: 选择 PD1 功能
1、0	PTSEL_D0[1:0]	00	R/W	选择端口 D (PD0) 的功能。 00: 选择 MII_TXD3 功能 01: 选择 SSIDATA5 功能 10: 选择 IODACK_M 功能 11: 选择 PD0 功能

27.2.26 引脚选择寄存器 4 (PTSEL_E)

PTSEL_E 是 16 位可读写寄存器，选择多路复用引脚的功能。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTSEL_E7[1:0]	PTSEL_E6[1:0]	PTSEL_E5[1:0]	PTSEL_E4[1:0]	PTSEL_E3[1:0]	PTSEL_E2[1:0]	PTSEL_E1[1:0]	PTSEL_E0[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

位	位名	初始值	R/W	说明
15、14	PTSEL_E7[1:0]	00	R/W	选择端口 E (PE7) 的功能。 00: 选择 COL 功能 01: 选择 PE7 功能 10: 选择 IDEA2_M 功能 11: 禁止设定
13、12	PTSEL_E6[1:0]	00	R/W	选择端口 E (PE6) 的功能。 00: 选择 RX_ER 功能 01: 选择 PE6 功能 10: 选择 IODREQ_M 功能 11: 禁止设定
11、10	PTSEL_E5[1:0]	00	R/W	选择端口 E (PE5) 的功能。 00: 选择 RX_CLK 功能 01: 选择 PE5 功能 10: 选择 IDED1_M 功能 11: 禁止设定
9、8	PTSEL_E4[1:0]	00	R/W	选择端口 E (PE4) 的功能。 00: 选择 RX_DV 功能 01: 选择 PE4 功能 10: 选择 IDED14_M 功能 11: 禁止设定
7、6	PTSEL_E3[1:0]	00	R/W	选择端口 E (PE3) 的功能。 00: 选择 MII_RXD0 功能 01: 选择 SSIWS4 功能 10: 选择 IDED2_M 功能 11: 选择 PE3 功能
5、4	PTSEL_E2[1:0]	00	R/W	选择端口 E (PE2) 的功能。 00: 选择 MII_RXD1 功能 01: 选择 SSISCK4 功能 10: 选择 IDED13_M 功能 11: 选择 PE2 功能
3、2	PTSEL_E1[1:0]	00	R/W	选择端口 E (PE1) 的功能。 00: 选择 MII_RXD2 功能 01: 选择 SSIDATA4 功能 10: 选择 IDED3_M 功能 11: 选择 PE1 功能
1、0	PTSEL_E0[1:0]	00	R/W	选择端口 E (PE0) 的功能。 00: 选择 MII_RXD3 功能 01: 选择 AUDIO_CLK4 功能 10: 选择 IDED12_M 功能 11: 选择 PE0 功能

27.2.27 引脚选择寄存器 5 (PTSEL_F)

PTSEL_F 是 16 位可读写寄存器，选择多路复用引脚的功能。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PTSEL_F7	—	PTSEL_F6	PTSEL_F5[1:0]	PTSEL_F4[1:0]	PTSEL_F3[1:0]	PTSEL_F2[1:0]	PTSEL_F1[1:0]	PTSEL_F0[1:0]						
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	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 读取值为 0，只能写 0。
14	PTSEL_F7	0	R/W	选择端口 F (PF7) 的功能 0: 选择 D32 功能 1: 选择 PF7 功能
13	—	0	R	保留位 读取值为 0，只能写 0。
12	PTSEL_F6	0	R/W	选择端口 F (PF6) 的功能 0: 选择 D33 功能 1: 选择 PF6 功能
11、10	PTSEL_F5[1:0]	00	R/W	选择端口 F (PF5) 的功能。 00: 选择 D34 功能 01: 选择 PF5 功能 10: 禁止设定 11: 禁止设定
9、8	PTSEL_F4[1:0]	00	R/W	选择端口 F (PF4) 的功能。 00: 选择 EXOUT 功能 01: 选择 PF4 功能 10: 选择 IDECS1_M 功能 11: 禁止设定
7、6	PTSEL_F3[1:0]	00	R/W	选择端口 F (PF3) 的功能。 00: 选择 LNKSTA 功能 01: 选择 PF3 功能 10: 选择 IDECS0_M 功能 11: 禁止设定
5、4	PTSEL_F2[1:0]	00	R/W	选择端口 F (PF2) 的功能。 00: 选择 WOL 功能 01: 选择 PF2 功能 10: 选择 IDEA0_M 功能 11: 禁止设定
3、2	PTSEL_F1[1:0]	00	R/W	选择端口 F (PF1) 的功能。 00: 选择 MDIO 功能 01: 选择 PF1 功能 10: 选择 IDED11_M 功能 11: 禁止设定
1、0	PTSEL_F0[1:0]	00	R/W	选择端口 F (PF0) 的功能。 00: 选择 MDC 功能 01: 选择 PF0 功能 10: 选择 IDED4_M 功能 11: 禁止设定

27.2.28 引脚选择寄存器 6 (PTSEL_G)

PTSEL_G 是 16 位可读写寄存器，选择多路复用引脚的功能。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTSEL_G7[1:0]		PTSEL_G6[1:0]		PTSEL_G5[1:0]		PTSEL_G4[1:0]		PTSEL_G3[1:0]		PTSEL_G2[1:0]		PTSEL_G1[1:0]		PTSEL_G0[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

位	位名	初始值	R/W	说明
15、14	PTSEL_G7[1:0]	00	R/W	选择端口 G (PG7) 的功能。 00: 选择 LCD_DATA15 功能 01: 选择 DR3 功能 10: 选择 PG7 功能 11: 禁止设定
13、12	PTSEL_G6[1:0]	00	R/W	选择端口 G (PG6) 的功能。 00: 选择 LCD_DATA14 功能 01: 选择 DR2 功能 10: 选择 PG6 功能 11: 禁止设定
11、10	PTSEL_G5[1:0]	00	R/W	选择端口 G (PG5) 的功能。 00: 选择 LCD_DATA13 功能 01: 选择 DR1 功能 10: 选择 PG5 功能 11: 禁止设定
9、8	PTSEL_G4[1:0]	00	R/W	选择端口 G (PG4) 的功能。 00: 选择 LCD_DATA12 功能 01: 选择 DR0 功能 10: 选择 PG4 功能 11: 禁止设定
7、6	PTSEL_G3[1:0]	00	R/W	选择端口 G (PG3) 的功能。 00: 选择 LCD_DATA11 功能 01: 选择 DG5 功能 10: 选择 PG3 功能 11: 禁止设定
5、4	PTSEL_G2[1:0]	00	R/W	选择端口 G (PG2) 的功能。 00: 选择 LCD_DATA10 功能 01: 选择 DG4 功能 10: 选择 PG2 功能 11: 禁止设定
3、2	PTSEL_G1[1:0]	00	R/W	选择端口 G (PG1) 的功能。 00: 选择 LCD_DATA9 功能 01: 选择 DG3 功能 10: 选择 PG1 功能 11: 禁止设定
1、0	PTSEL_G0[1:0]	00	R/W	选择端口 G (PG0) 的功能。 00: 选择 LCD_DATA8 功能 01: 选择 DG2 功能 10: 选择 PG0 功能 11: 禁止设定

27.2.29 引脚选择寄存器 7 (PTSEL_H)

PTSEL_H 是 16 位可读写寄存器，选择多路复用引脚的功能。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PTSEL_H7	—	PTSEL_H6	—	PTSEL_H5	—	PTSEL_H4	PTSEL_H3[1:0]	PTSEL_H2[1:0]	PTSEL_H1[1:0]	PTSEL_H0[1:0]				
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15	—	0	R	保留位 读取值为 0，只能写 0。
14	PTSEL_H7	0	R/W	选择端口 H (PH7) 的功能。 0: 选择 AUDIO_CLK3 功能 1: 选择 PH7 功能
13	—	0	R	保留位 读取值为 0，只能写 0。
12	PTSEL_H6	0	R/W	选择端口 H (PH6) 的功能。 0: 选择 SSIWS3 功能 1: 选择 PH6 功能
11	—	0	R	保留位 读取值为 0，只能写 0。
10	PTSEL_H5	0	R/W	选择端口 H (PH5) 的功能。 0: 选择 SSISCK3 功能 1: 选择 PH5 功能
9	—	0	R	保留位 读取值为 0，只能写 0。
8	PTSEL_H4	0	R/W	选择端口 H (PH4) 的功能。 0: 选择 SSIDATA3 功能 1: 选择 PH4 功能
7、6	PTSEL_H3[1:0]	00	R/W	选择端口 H (PH3) 的功能。 00: 选择 LCD_CL2 功能 01: 选择 DE_V 功能 10: 选择 PH3 功能 11: 禁止设定
5、4	PTSEL_H2[1:0]	00	R/W	选择端口 H (PH2) 的功能。 00: 选择 LCD_DON 功能 01: 选择 DCLKOUT 功能 10: 禁止设定 11: 选择 PH2 功能
3、2	PTSEL_H1[1:0]	00	R/W	选择端口 H (PH1) 的功能。 00: 选择 LCD_VCP_WC 功能 01: 选择 DR4 功能 10: 选择 PH1 功能 11: 禁止设定
1、0	PTSEL_H0[1:0]	00	R/W	选择端口 H (PH0) 的功能。 00: 选择 LCD_VEP_WC 功能 01: 选择 DR5 功能 10: 选择 PH0 功能 11: 禁止设定

27.2.30 引脚选择寄存器 8 (PTSEL_I)

PTSEL_I 是 16 位可读写寄存器，选择多路复用引脚的功能。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTSEL_I7[1:0]		PTSEL_I6[1:0]		PTSEL_I5[1:0]		PTSEL_I4[1:0]		PTSEL_I3[1:0]		PTSEL_I2[1:0]		PTSEL_I1[1:0]		PTSEL_I0[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

位	位名	初始值	R/W	说明
15、14	PTSEL_I7[1:0]	00	R/W	选择端口 I (PI7) 的功能。 00: 选择 LCD_DATA3 功能 01: 选择 DB3 功能 10: 选择 BT_DATA3 功能 11: 禁止设定
13、12	PTSEL_I6[1:0]	00	R/W	选择端口 I (PI6) 的功能。 00: 选择 LCD_DATA2 功能 01: 选择 DB2 功能 10: 选择 BT_DATA2 功能 11: 禁止设定
11、10	PTSEL_I5[1:0]	00	R/W	选择端口 I (PI5) 的功能。 00: 选择 LCD_DATA1 功能 01: 选择 DB1 功能 10: 选择 BT_DATA1 功能 11: 禁止设定
9、8	PTSEL_I4[1:0]	00	R/W	选择端口 I (PI4) 的功能。 00: 选择 LCD_DATA7 功能 01: 选择 DG1 功能 10: 选择 BT_DATA7 功能 11: 选择 PI4 功能
7、6	PTSEL_I3[1:0]	00	R/W	选择端口 I (PI3) 的功能。 00: 选择 LCD_DATA6 功能 01: 选择 DG0 功能 10: 选择 BT_DATA6 功能 11: 选择 PI3 功能
5、4	PTSEL_I2[1:0]	00	R/W	选择端口 I (PI2) 的功能。 00: 选择 LCD_DATA5 功能 01: 选择 DB5 功能 10: 选择 BT_DATA5 功能 11: 选择 PI2 功能
3、2	PTSEL_I1[1:0]	00	R/W	选择端口 I (PI1) 的功能。 00: 选择 LCD_DATA4 功能 01: 选择 DB4 功能 10: 选择 BT_DATA4 功能 11: 选择 PI1 功能
1、0	PTSEL_I0[1:0]	00	R/W	选择端口 I (PI0) 的功能。 00: 选择 PI0 功能 01: 选择 COM/CDE 功能 10: 禁止设定 11: 禁止设定

27.2.31 引脚选择寄存器 9 (PTSEL_J)

PTSEL_J 是 16 位可读写寄存器，选择多路复用引脚的功能。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTSEL_J7[1:0]		PTSEL_J6[1:0]		PTSEL_J5[1:0]		PTSEL_J4[1:0]		PTSEL_J3[1:0]		PTSEL_J2[1:0]		PTSEL_J1[1:0]		PTSEL_J0[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

位	位名	初始值	R/W	说明
15、14	PTSEL_J7[1:0]	00	R/W	选择端口 J (PJ7) 的功能。 00: 选择 PJ7 功能 01: 禁止设定 10: 禁止设定 11: 选择 IDED10_M 功能
13、12	PTSEL_J6[1:0]	00	R/W	选择端口 J (PJ6) 的功能。 00: 选择 PJ6 功能 01: 禁止设定 10: 禁止设定 11: 选择 IDED5_M 功能
11、10	PTSEL_J5[1:0]	00	R/W	选择端口 J (PJ5) 的功能。 00: 选择 PJ5 功能 01: 禁止设定 10: 禁止设定 11: 选择 IDED9_M 功能
9、8	PTSEL_J4[1:0]	00	R/W	选择端口 J (PJ4) 的功能。 00: 选择 PJ4 功能 01: 禁止设定 10: 禁止设定 11: 选择 IDED6_M 功能
7、6	PTSEL_J3[1:0]	00	R/W	选择端口 J (PJ3) 的功能。 00: 选择 PJ3 功能 01: 禁止设定 10: 禁止设定 11: 选择 IDED7_M 功能
5、4	PTSEL_J2[1:0]	00	R/W	选择端口 J (PJ2) 的功能。 00: 选择 PJ2 功能 01: 禁止设定 10: 禁止设定 11: 选择 IDED8_M 功能
3、2	PTSEL_J1[1:0]	00	R/W	选择端口 J (PJ1) 的功能。 00: 选择 PJ1 功能 01: 禁止设定 10: 禁止设定 11: 选择 <u>IDERST_M</u> 功能
1、0	PTSEL_J0[1:0]	00	R/W	选择端口 J (PJ0) 的功能。 00: 选择 PJ0 功能 01: 禁止设定 10: 禁止设定 11: 选择 <u>DIRECTION_M</u> 功能

27.2.32 引脚选择寄存器 10 (PTSEL_K)

PTSEL_K 是 16 位可读写寄存器，选择多路复用引脚的功能。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTSEL_K7[1:0]	PTSEL_K6[1:0]	—	PTSEL_K5	PTSEL_K4[1:0]	PTSEL_K3[1:0]	PTSEL_K2[1:0]	PTSEL_K1[1:0]	PTSEL_K0[1:0]							
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
15、14	PTSEL_K7[1:0]	00	R/W	选择端口 K (PK7) 的功能。 00: 选择 $\overline{\text{WDTOVF}}$ 功能 01: 选择 $\overline{\text{IRQ1}}$ 功能 10: 选择 $\overline{\text{AUDCK}}$ 功能 11: 选择 $\overline{\text{DACK1}}$ 功能
13、12	PTSEL_K6[1:0]	00	R/W	选择端口 K (PK6) 的功能。 00: 选择 SCK0 功能 01: 选择 AUDSYNC 功能 10: 选择 FCLE 功能 11: 禁止设定
11	—	0	R	保留位 读取值为 0，只能写 0。
10	PTSEL_K5	0	R/W	选择端口 K (PK5) 的功能。 0: 选择 SCK1 功能 1: 选择 FR/B 功能
9、8	PTSEL_K4[1:0]	00	R/W	选择端口 K (PK4) 的功能。 00: 选择 LCD_DATA0 功能 01: 选择 DB0 功能 10: 选择 BT_DATA0 功能 11: 禁止设定
7、6	PTSEL_K3[1:0]	00	R/W	选择端口 K (PK3) 的功能。 00: 选择 LCD_CL1 功能 01: 选择 HSYNC/SPL 功能 * 10: 选择 BT_HSYNC 功能 11: 禁止设定
5、4	PTSEL_K2[1:0]	00	R/W	选择端口 K (PK2) 的功能。 00: 选择 LCD_CLK 功能 01: 选择 DCLKIN 功能 10: 禁止设定 11: 禁止设定
3、2	PTSEL_K1[1:0]	00	R/W	选择端口 K (PK1) 的功能。 00: 选择 LCD_FLM 功能 01: 选择 VSYNC/SPS 功能 * 10: 选择 BT_VSYNC 功能 11: 禁止设定
1、0	PTSEL_K0[1:0]	00	R/W	选择端口 K (PK0) 的功能。 00: 选择 LCD_M_DISP 功能 01: 选择 DE_C/DE_H 功能 10: 选择 BT_DE_C 功能 11: 禁止设定

【注】 * 此功能通过特殊选择寄存器进行输入 / 输出功能的转换。

27.2.33 引脚选择寄存器 11 (PTSEL_P)

PTSEL_P 是 16 位可读写寄存器，选择多路复用引脚的功能。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PTSEL_11	PTSEL_10	PTSEL_9	PTSEL_8	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15 ~ 12	—	全 0	R	保留位 读取值为 0，只能写 0。
11	PTSEL_P11	0	R/W	选择端口 P (PP11) 的功能。 0: 选择 RXD0 功能 1: 选择 AUDATA0 功能
10	PTSEL_P10	0	R/W	选择端口 P (PP10) 的功能。 0: 选择 TXD0 功能 1: 选择 AUDATA1 功能
9	PTSEL_P9	0	R/W	选择端口 P (PP9) 的功能。 0: 选择 RXD1 功能 1: 选择 AUDATA2 功能
8	PTSEL_P8	0	R/W	选择端口 P (PP8) 的功能。 0: 选择 TXD1 功能 1: 选择 AUDATA3 功能
7 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

27.2.34 引脚选择寄存器 12 (PTSEL_R)

PTSEL_R 是 16 位可读写寄存器，选择多路复用引脚的功能。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTSEL_R15	PTSEL_R14	PTSEL_R13	PTSEL_R12	PTSEL_R11	PTSEL_R10	PTSEL_R9	PTSEL_R8	PTSEL_R7	PTSEL_R6	PTSEL_R5	PTSEL_R4	PTSEL_R3	PTSEL_R2	PTSEL_R1	PTSEL_R0
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	PTSEL_R15	0	R/W	选择端口 P (PR15) 的功能。 0: 选择 D63 功能 1: 选择 IDED1 功能
14	PTSEL_R14	0	R/W	选择端口 P (PR14) 的功能。 0: 选择 D62 功能 1: 选择 IDED0 功能
13	PTSEL_R13	0	R/W	选择端口 P (PR13) 的功能。 0: 选择 D61 功能 1: 选择 IDED3 功能

位	位名	初始值	R/W	说明
12	PTSEL_R12	0	R/W	选择端口 P (PR12) 的功能。 0: 选择 D60 功能 1: 选择 IDED2 功能
11	PTSEL_R11	0	R/W	选择端口 P (PR11) 的功能。 0: 选择 D59 功能 1: 选择 IDED5 功能
10	PTSEL_R10	0	R/W	选择端口 P (PR10) 的功能。 0: 选择 D58 功能 1: 选择 IDED4 功能
9	PTSEL_R9	0	R/W	选择端口 P (PR9) 的功能。 0: 选择 D57 功能 1: 选择 IDED7 功能
8	PTSEL_R8	0	R/W	选择端口 P (PR8) 的功能。 0: 选择 D56 功能 1: 选择 IDED6 功能
7	PTSEL_R7	0	R/W	选择端口 P (PR7) 的功能。 0: 选择 D55 功能 1: 选择 DIRECTION 功能
6	PTSEL_R6	0	R/W	选择端口 P (PR6) 的功能。 0: 选择 D54 功能 1: 选择 IDERST 功能
5	PTSEL_R5	0	R/W	选择端口 P (PR5) 的功能。 0: 选择 D53 功能 1: 选择 IDED8 功能
4	PTSEL_R4	0	R/W	选择端口 P (PR4) 的功能。 0: 选择 D52 功能 1: 选择 IDED9 功能
3	PTSEL_R3	0	R/W	选择端口 P (PR3) 的功能。 0: 选择 D51 功能 1: 选择 IDED10 功能
2	PTSEL_R2	0	R/W	选择端口 P (PR2) 的功能。 0: 选择 D50 功能 1: 选择 IDED11 功能
1	PTSEL_R1	0	R/W	选择端口 P (PR1) 的功能。 0: 选择 D49 功能 1: 选择 IDED12 功能
0	PTSEL_R0	0	R/W	选择端口 P (PR0) 的功能。 0: 选择 D48 功能 1: 选择 IDED13 功能

27.2.35 引脚选择寄存器 13 (PTSEL_S)

PTSEL_S 是 16 位可读写寄存器，选择多路复用引脚的功能。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTSEL_S15	PTSEL_S14	PTSEL_S13	PTSEL_S12	PTSEL_S11	PTSEL_S10	PTSEL_S9	PTSEL_S8	PTSEL_S7	PTSEL_S6	PTSEL_S5	PTSEL_S4	PTSEL_S3	PTSEL_S2	PTSEL_S1	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

位	位名	初始值	R/W	说明
15	PTSEL_S15	0	R/W	选择端口 S (PS15) 的功能。 0: 选择 IRQ0 功能 1: 选择 DTEND1 功能
14	PTSEL_S14	0	R/W	选择端口 S (PS14) 的功能。 0: 选择 IRQOUT 功能 1: 选择 DREQ1 功能
13	PTSEL_S13	0	R/W	选择端口 S (PS13) 的功能。 0: 选择 D47 功能 1: 选择 IDECS0 功能
12	PTSEL_S12	0	R/W	选择端口 S (PS12) 的功能。 0: 选择 D46 功能 1: 选择 IDECS1 功能
11	PTSEL_S11	0	R/W	选择端口 S (PS11) 的功能。 0: 选择 D45 功能 1: 选择 IODACK 功能
10	PTSEL_S10	0	R/W	选择端口 S (PS10) 的功能。 0: 选择 D44 功能 1: 选择 IODINT 功能
9	PTSEL_S9	0	R/W	选择端口 S (PS9) 的功能。 0: 选择 D43 功能 1: 选择 IDEIORDY 功能
8	PTSEL_S8	0	R/W	选择端口 S (PS8) 的功能。 0: 选择 D42 功能 1: 选择 IDEIORD 功能
7	PTSEL_S7	0	R/W	选择端口 S (PS7) 的功能。 0: 选择 D41 功能 1: 选择 IODREQ 功能
6	PTSEL_S6	0	R/W	选择端口 S (PS6) 的功能。 0: 选择 D40 功能 1: 选择 IDEIOWR 功能
5	PTSEL_S5	0	R/W	选择端口 S (PS5) 的功能。 0: 选择 D39 功能 1: 选择 IDED14 功能
4	PTSEL_S4	0	R/W	选择端口 S (PS4) 的功能。 0: 选择 D38 功能 1: 选择 IDED15 功能

位	位名	初始值	R/W	说明
3	PTSEL_S3	0	R/W	选择端口 S (PS3) 的功能。 0: 选择 D37 功能 1: 选择 IDEA1 功能
2	PTSEL_S2	0	R/W	选择端口 S (PS2) 的功能。 0: 选择 D36 功能 1: 选择 IDEA2 功能
1	PTSEL_S1	0	R/W	选择端口 S (PS1) 的功能。 0: 选择 D35 功能 1: 选择 IDEA0 功能
0	—	0	R	保留位 读取值为 0, 只能写 0。

27.2.36 HI-Z 寄存器 A (PTHIZ_A)

PTHIZ_A 是 16 位可读写寄存器，将模块引脚设定为高阻抗状态。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTHIZ_ATA	PTHIZ_TMU	PTHIZ_LCD	PTHIZ_IIC	PTHIZ_FLCTL	PTHIZ_DMACH	PTHIZ_SCIO	PTHIZ_SC1	PTHIZ_SC2	PTHIZ_ETH	PTHIZ_VDC2	PTHIZ_USB	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

位	位名	初始值	R/W	说明
15	PTHIZ_ATA	0	R/W	将 ATAPI 的引脚设定为高阻抗状态。 0: 通常 1: 高阻抗状态
14	PTHIZ_TMU	0	R/W	将 TMU 的引脚设定为高阻抗状态。 0: 通常 1: 高阻抗状态
13	PTHIZ_LCD	0	R/W	将 LCD 的引脚设定为高阻抗状态。 0: 通常 1: 高阻抗状态
12	PTHIZ_IIC	0	R/W	将 IIC 的引脚设定为高阻抗状态。 0: 通常 1: 高阻抗状态
11	PTHIZ_FLCTL	0	R/W	将 FLCTL 的引脚设定为高阻抗状态。 0: 通常 1: 高阻抗状态
10	PTHIZ_DMACH	0	R/W	将 DMACH 的引脚设定为高阻抗状态。 0: 通常 1: 高阻抗状态
9	PTHIZ_SCIO	0	R/W	将 SCIF 通道 0 的引脚设定为高阻抗状态。 0: 通常 1: 高阻抗状态
8	PTHIZ_SC1	0	R/W	将 SCIF 通道 1 的引脚设定为高阻抗状态。 0: 通常 1: 高阻抗状态
7	PTHIZ_SC2	0	R/W	将 SCIF 通道 2 的引脚设定为高阻抗状态。 0: 通常 1: 高阻抗状态
6	PTHIZ_ETH	0	R/W	将 EtherC 的引脚设定为高阻抗状态。 0: 通常 1: 高阻抗状态
5	PTHIZ_VDC2	0	R/W	将 VDC2 的引脚设定为高阻抗状态。 0: 通常 1: 高阻抗状态
4	PTHIZ_USB	0	R/W	将 USB 的引脚设定为高阻抗状态。 0: 通常 1: 高阻抗状态
3 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

27.2.37 HI-Z 寄存器 B (PTHIZ_B)

PTHIZ_B 是 16 位可读写寄存器，将模块引脚设定为高阻抗状态。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTHIZ_SSI0	PTHIZ_SSI1	PTHIZ_SSI2	PTHIZ_SSI3	PTHIZ_SSI4	PTHIZ_SSI5	—	—	—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
15	PTHIZ_SSI0	0	R/W	将 SSI 通道 0 的引脚设定为高阻抗状态。 0: 通常 1: 高阻抗状态
14	PTHIZ_SSI1	0	R/W	将 SSI 通道 1 的引脚设定为高阻抗状态。 0: 通常 1: 高阻抗状态
13	PTHIZ_SSI2	0	R/W	将 SSI 通道 2 的引脚设定为高阻抗状态。 0: 通常 1: 高阻抗状态
12	PTHIZ_SSI3	0	R/W	将 SSI 通道 3 的引脚设定为高阻抗状态。 0: 通常 1: 高阻抗状态
11	PTHIZ_SSI4	0	R/W	将 SSI 通道 4 的引脚设定为高阻抗状态。 0: 通常 1: 高阻抗状态
10	PTHIZ_SSI5	0	R/W	将 SSI 通道 5 的引脚设定为高阻抗状态。 0: 通常 1: 高阻抗状态
9 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

27.2.38 特殊选择寄存器 (PTSEL_SPCL)

PTSEL_SPCL 是 16 位可读写寄存器，选择 HSYNC 和 VSYNC 的输入 / 输出功能。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PTSEL_VSYNC	PTSEL_HSYNC
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 读取值为 0，只能写 0。
1	PTSEL_VSYNC	0	R/W	选择 VSYNC 的功能。 0: 选择 VSYNC/SPS 功能 1: 选择 EX_VSYNC 功能
0	PTSEL_HSYNC	0	R/W	选择 HSYNC 的功能。 0: 选择 HSYNC/SPL 功能 1: 选择 EX_HSYNC 功能

27.3 使用例子

举例说明通用输入 / 输出端口 (GPIO) 的设定步骤。

27.3.1 端口功能的选择

在选择端口功能时，首先必须按照以下内容设定端口输出功能或者端口输入功能；然后，通过引脚选择寄存器 (PTSEL_A ~ PTSEL_J) 选择要使用的端口功能。

在错误地设定端口的输入 / 输出功能的状态下，如果将引脚选择寄存器设定为端口功能，就可能发生信号冲突等问题，所以必须注意。

27.3.2 端口输出功能

在设定端口输出功能时，必须将端口控制寄存器 (PTIO_A ~ PTIO_J) 的 2 个对应位置 B'01，将端口数据寄存器 (PTDAT_A ~ PTDAT_J) 的对应位的值输出到引脚。

此时，对于设定为端口输出功能的引脚，上拉控制寄存器 (PTPUL_AB ~ PTUPL_IJ) 的设定无效。

27.3.3 端口输入功能

在设定端口输入功能时，如果不使用上拉 MOS，就必须将端口控制寄存器 (PTIO_A ~ PTIO_J) 的 2 个对应位置 B'10；如果使用上拉 MOS，就必须置 B'11。能从端口数据寄存器 (PTDAT_A ~ PTDAT_J) 的对应位读取引脚的输入值。

27.3.4 外围模块功能

在设定外围模块功能时，首先必须通过引脚选择寄存器 (PTSEL_A ~ PTSEL_J) 选择要使用的模块，然后将端口控制寄存器 (PTIO_A ~ PTIO_J) 的 2 个对应位置 B'00。

第 28 章 低功耗模式

在功耗模式中，停止内部模块的部分功能和 CPU 功能，所以能降低功耗。

28.1 特点

- 支持刷新待机模式
- 支持睡眠模式/模块待机功能

28.1.1 功耗模式的种类

功耗模式有以下模式和功能：

- 睡眠模式
- 刷新待机模式
- 模块待机功能

各模式的 CPU 和内部模块等的状态如表 28.1 所示。

表 28.1 低功耗模式的状态

低功耗模式	转移方法	CPG	CPU	内部存储器	内部外围模块	引脚	SDRAM	解除方法	S1*	S0*
睡眠模式	在 STBCR 的 STBY=0 时执行 SLEEP 指令。	运行	停止 保持寄存器的内容	保持	运行	保持	AR 或者 SR	(1) 中断 (2) 上电复位	1	0
刷新待机模式	在 STBCR 的 STBY=1 时执行 SLEEP 指令。	运行	停止 保持寄存器的内容	停止 保持内容	停止	保持 CLKOUT 运行	SR	(1) NMI、IRQ (2) 上电复位	0	1
模块待机功能	将 MSTPCR 的相应位置 1。	运行	运行	保持	指定的模块停止	保持	AR 或者 SR	将 MSTPCR 的相应位置 0。	0	0
上电复位	将 $\overline{\text{PRESET}}$ 引脚设定为低电平。	初始状态	初始状态	初始状态	初始状态	初始状态	初始状态	—	1	1
正常状态		运行	运行	运行	运行	运行	运行	—	0	0

【注】 * S1 和 S0 分别表示 STATUS1 引脚和 STATUS0 引脚的输出状态。

28.2 输入 / 输出引脚

功耗模式的相关引脚如表 28.2 所示。

表 28.2 引脚结构

引脚名	功能	输入 / 输出	说明
STATUS1	处理状态 1	输出	表示本 LSI 的运行状态。
STATUS0	处理状态 0		STATUS[1:0] 运行状态 H H : 上电复位 H L : 睡眠模式 L L : 正常运行 L H : 刷新待机模式

28.3 寄存器说明

低功耗模式的寄存器结构如表 28.3 所示，各处理模式的寄存器状态如表 28.4 所示。

表 28.3 寄存器结构

名称	略称	R/W	P4 区地址	区域 7 地址	存取长度
待机控制寄存器	STBCR	R/W	H'FFC8 0020	H'1FC8 0020	32
模块停止寄存器 0	MSTPCR0	R/W	H'FFC8 0030	H1FC8 0030	32
模块停止寄存器 1	MSTPCR1	R/W	H'FFC8 0038	H'1FC8 0038	32

【注】 有关待机控制寄存器，请参照表 9.1。

表 28.4 各处理模式的寄存器状态

名称	略称	上电复位	睡眠	待机
待机控制寄存器	STBCR	H'0000 0000	保持	保持
模块停止寄存器 0	MSTPCR0	H'0000 0000	保持	保持
模块停止寄存器 1	MSTPCR1	H'0000 0000	保持	保持

【注】 有关待机控制寄存器，请参照表 9.1。

28.3.1 待机控制寄存器 (STBCR)

STBCR 是 32 位可读写寄存器，能指定执行 SLEEP 指令后的低功耗模式。

STBCR 只能进行长字存取。

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
	—	—	—	—	—	—	—	—	STBY	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
31 ~ 8	—	全 0	R	保留位 读取值为 0，只能写 0。
7	STBY	0	R/W	待机位 指定在执行 SLEEP 指令后是转移到睡眠模式还是转移到刷新待机模式。 0: 转移到睡眠模式 1: 转移到刷新待机模式 【注】 在通过中断从刷新待机模式返回时，必须将此位清 0。
6 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

28.3.2 模块停止寄存器 0 (MSTPCR0)

MSTPCR0 是 32 位可读写寄存器，能分别指定各模块的运行和停止。

MSTPCR0 只能进行长字存取。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	INTC	DMAC	—	H-UDI	—	UBC	—
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	R/W	R	R/W	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LDCD	—	TMU	FLCTL	—	SCIF2	SCIF1	SCIF0	ETHER	IIC	ATAPI	G2D	—	VDC2	—	USB
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R/W

位	位名	初始值	R/W	说明
31 ~ 23	—	全 0	R	保留位 读取值为 0，只能写 0。
22	INTC	0	R/W	INTC 模块停止位 当将 INTC 位置 1 时，就停止向 INTC 提供时钟。 0: INTC 运行 1: 停止向 INTC 提供时钟

位	位名	初始值	R/W	说明
21	DMAC	0	R/W	DMAC 模块停止位 当将 DMAC 位置 1 时, 就停止向 DMAC 提供时钟。 0: DMAC 运行 1: 停止向 DMAC 提供时钟
20	—	0	R	保留位 读取值为 0, 只能写 0。
19	H-UDI	0	R/W	H-UDI 模块停止位 当将 H-UDI 位置 1 时, 就停止向 H-UDI 提供时钟。 0: H-UDI 运行 1: 停止向 H-UDI 提供时钟
18	—	0	R	保留位 读取值为 0, 只能写 0。
17	UBC	0	R/W	UBC 模块停止位 当将 UBC 位置 1 时, 就停止向 UBC 提供时钟。 0: UBC 运行 1: 停止向 UBC 提供时钟
16	—	0	R	保留位 读取值为 0, 只能写 0。
15	LCDC	0	R/W	LCDC 模块停止位 当将 LCDC 位置 1 时, 就停止向 LCDC 提供时钟。 0: LCDC 运行 1: 停止向 LCDC 提供时钟
14	—	0	R	保留位 读取值为 0, 只能写 0。
13	TMU	0	R/W	TMU 模块停止位 当将 TMU 位置 1 时, 就停止向 TMU 提供时钟。 0: TMU 运行 1: 停止向 TMU 提供时钟
12	FLCTL	0	R/W	FLCTL 模块停止位 当将 FLCTL 位置 1 时, 就停止向 FLCT 提供时钟。 0: FLCTL 运行 1: 停止向 FLCTL 提供时钟
11	—	0	R	保留位 读取值为 0, 只能写 0。
10	SCIF2	0	R/W	SCIF2 模块停止位 当将 SCIF2 位置 1 时, 就停止向 SCIF2 提供时钟。 0: SCIF2 运行 1: 停止向 SCIF2 提供时钟
9	SCIF1	0	R/W	SCIF1 模块停止位 当将 SCIF1 位置 1 时, 就停止向 SCIF1 提供时钟。 0: SCIF1 运行 1: 停止向 SCIF1 提供时钟

位	位名	初始值	R/W	说明
8	SCIF0	0	R/W	SCIF0 模块停止位 当将 SCIF0 位置 1 时, 就停止向 SCIF0 提供时钟。 0: SCIF0 运行 1: 停止向 SCIF0 提供时钟
7	ETHER	0	R/W	ETHER 模块停止位 当将 ETHER 位置 1 时, 就停止向 ETHER 提供时钟。 0: ETHER 运行 1: 停止向 ETHER 提供时钟
6	IIC	0	R/W	IIC 模块停止位 当将 IIC 位置 1 时, 就停止向 IIC 提供时钟。 0: IIC 运行 1: 停止向 IIC 提供时钟
5	ATAPI	0	R/W	ATAPI 模块停止位 当将 ATAPI 位置 1 时, 就停止向 ATAPI 提供时钟。 0: ATAPI 运行 1: 停止向 ATAPI 提供时钟
4	G2D	0	R/W	G2D 模块停止位 当将 G2D 位置 1 时, 就停止向 G2D 提供时钟。 0: G2D 运行 1: 停止向 G2D 提供时钟
3	—	0	R	保留位 读取值为 0, 只能写 0。
2	VDC2	0	R/W	VDC2 模块停止位 当将 VDC2 位置 1 时, 就停止向 VDC2 提供时钟。 0: VDC2 运行 1: 停止向 VDC2 提供时钟
1	—	0	R	保留位 读取值为 0, 只能写 0。
0	USB	0	R/W	USB 模块停止位 当将 USB 位置 1 时, 就停止向 USB 提供时钟。 0: USB 运行 1: 停止向 USB 提供时钟

28.3.3 模块停止寄存器 1 (MSTPCR1)

MSTPCR1 是 32 位可读写寄存器，能分别指定各模块的运行和停止。

MSTPCR1 只能进行长字存取。

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

位	位名	初始值	R/W	说明
31	SRC	0	R/W	SRC 模块停止位 当将 SRC 位置 1 时，就停止向 SRC 提供时钟。 0: SRC 运行 1: 停止向 SRC 提供时钟
30 ~ 26	—	0	R	保留位 读取值为 0，只能写 0。
25	SSI_B	0	R/W	SSI_B 模块停止位 当将 SSI_B 位置 1 时，就停止向 SSI_B 提供时钟。 0: SSI_B 运行 1: 停止向 SSI_B 提供时钟
24	SSI_A	0	R/W	SSI_A 模块停止位 当将 SSI_A 位置 1 时，就停止向 SSI_A 提供时钟。 0: SSI_A 运行 1: 停止向 SSI_A 提供时钟
23 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

28.4 睡眠模式

28.4.1 睡眠模式的转移

在 STBCR 寄存器的 STBY 位为 0 的状态下，如果执行 SLEEP 指令，就从程序执行状态转移到睡眠模式。在执行 SLEEP 指令后，CPU 停止运行，但是保持 CPU 的寄存器内容。

CPU 以外的内部模块继续运行，并且 CLKOUT 引脚继续输出时钟。

在睡眠模式中，STATUS1 引脚输出高电平，STATUS0 引脚输出低电平。

28.4.2 睡眠模式的解除

能通过中断（NMI、IRQ1、IRQ0 和内部模块）和复位解除睡眠模式。

在睡眠模式中，即使 SR 寄存器的 BL 位为 1 也接受中断。必须根据需要，在执行 SLEEP 指令前将 SPC 和 SSR 压栈。

(1) 通过中断解除睡眠模式

如果发生 NMI、IRQ1、IRQ0 或者内部模块的中断，就解除睡眠模式，并且进行中断异常处理。对应中断源的异常码被设定到 INTEVT 寄存器。

(2) 通过复位解除睡眠模式

通过 $\overline{\text{PRESET}}$ 引脚的上电复位和看门狗定时器上溢所产生的上电复位解除睡眠模式。

28.5 刷新待机模式

28.5.1 刷新待机模式的转移

在 STBCR 的 STBY 位为 1 的状态下，如果执行 SLEEP 指令，就从程序执行状态转移到刷新待机模式。在刷新待机模式中，CPU、时钟和内部外围模块都停止运行。但是，MCU 和 CPG 运行，CLKOUT 引脚继续输出时钟。

保持 CPU 和高速缓存的寄存器内容，而一部分外围模块的寄存器被初始化。

向刷新待机模式的转移步骤如下所示：

1. 将 STBCR 的 STBY 位置 1。
2. 执行 SLEEP 指令。
3. 进入刷新待机模式，LSI 内部的时钟停止，从 STATUS0 引脚输出高电平。

28.5.2 刷新待机模式的解除

能通过中断（NMI、IRQ1 和 IRQ0）和复位解除刷新待机模式。

(1) 通过中断解除刷新待机模式

如果检测到 NMI 或者 IRQ 中断，就解除刷新待机模式，STATUS0 引脚变为低电平，并且进行中断异常处理。对应中断源的异常码被设定到 INTEVT 寄存器。在转移到中断处理程序后，必须清除 STBCR 寄存器的 STBY 位。在刷新待机模式中，即使 SR 寄存器的 BL 位为 1 也接受中断，所以必须根据需要，在执行 SLEEP 指令前将 SPC 和 SSR 压栈。

(2) 通过复位解除刷新待机模式

通过 $\overline{\text{PRESET}}$ 引脚的上电复位解除刷新待机模式。

28.6 模块待机功能

28.6.1 模块待机功能的转移

通过将模块停止寄存器的各位置 1，就停止向各位对应的内部模块提供时钟。

模块待机状态中的各模块继续保持转移到模块待机状态前的状态，并且寄存器的设定值和外部引脚也保持停止前的状态。当从模块待机状态返回时，就从停止前的状态开始运行。

28.6.2 模块待机功能的解除

通过将模块停止寄存器的各位清 0 或者上电复位，解除模块待机功能。

28.7 STATUS 引脚的变化时序

28.7.1 复位时

请参照“29.5.1 PRESET 引脚的上电复位”。

28.7.2 解除睡眠模式时

(1) 在睡眠模式中发生中断

STATUS 引脚的变化时序如图 28.1 所示。

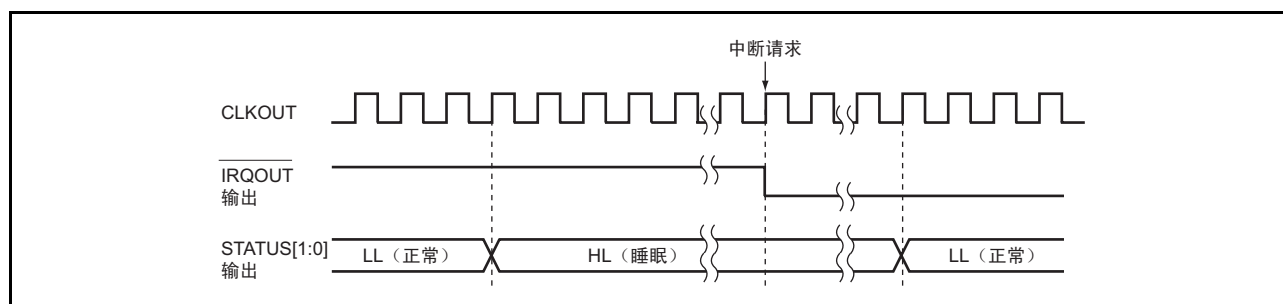


图 28.1 在睡眠模式中发生中断时的 STATUS 输出

第 29 章 复位和看门狗定时器 (WDT)

复位和看门狗定时器 (WDT) 由复位控制部和看门狗定时器控制部构成, 控制上电复位时的顺序以及本 LSI 内部和外部外围设备的复位。

WDT 是 1 个通道的定时器, 能用作看门狗定时器或者间隔定时器。

29.1 特点

- 在一定周期内使用计数的定时器, 实现系统失控的监视功能。
- 有通过计数器上溢进行内部复位的看门狗定时器模式和产生中断的间隔定时器模式。
- 在看门狗定时器模式中, 通过计数器上溢进行内部模块复位, 并输出 WDTOVF 信号。
- 在间隔定时器模式中, 通过计数器上溢产生间隔定时器中断。
- 为了使 WDT 的相关寄存器不容易改写, 给高 8 位设定了代码值。
- 计数器上溢的最大时间约为 21 秒 (外围时钟: Pck=50MHz 时)。

复位和看门狗定时器 (WDT) 的框图如图 29.1 所示。

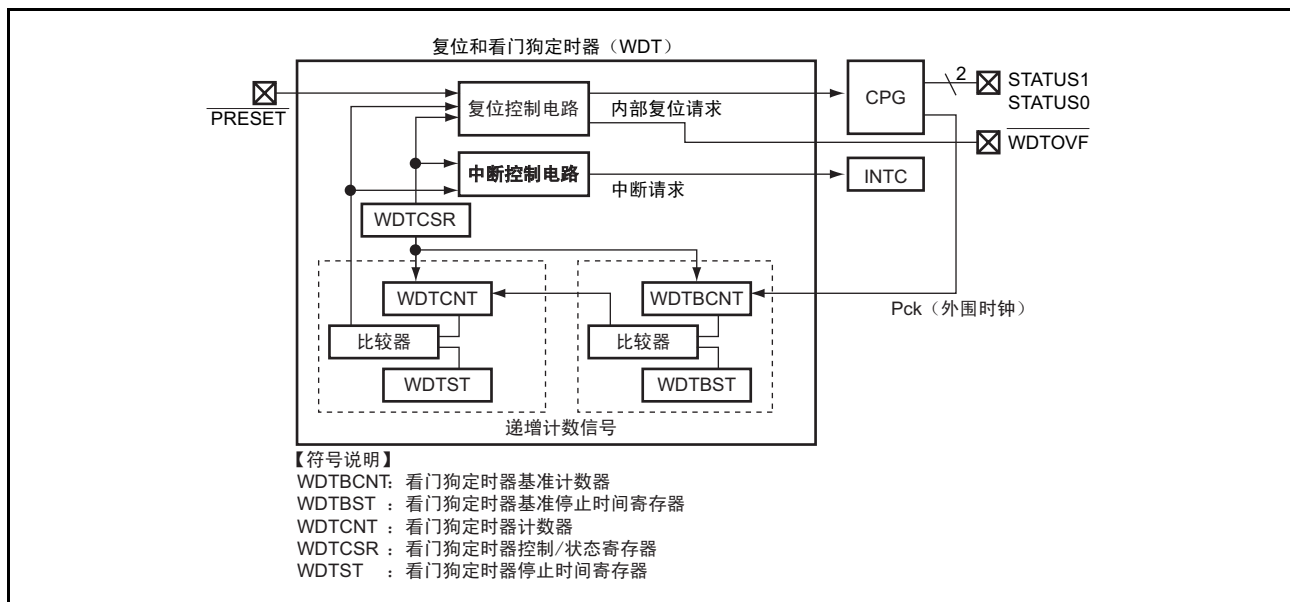


图 29.1 框图

29.2 输入 / 输出引脚

复位和 WDT 的引脚结构及其功能如表 29.1 所示。

表 29.1 复位和 WDT 的引脚结构及其功能

引脚名	功能	输入 / 输出	说明
$\overline{\text{PRESET}}$	上电复位的输入引脚	输入	通过给引脚输入低电平, 转移到上电复位状态。
STATUS1 STATUS0	状态输出	输出	表示处理器的运行状态。 HH: 复位 HL: 睡眠模式 LH: 刷新待机模式 LL: 正常运行 STATUS1 和 STATUS0 引脚是分别与其他功能多路复用的引脚。
$\overline{\text{WDTOVF}}$	看门狗定时器上溢	输出	输出看门狗定时器模式时的计数器上溢信号。

29.3 寄存器说明

WDT 的寄存器结构如表 29.2 所示, 各处理模式的寄存器状态如表 29.3 所示。

表 29.2 寄存器结构

名称	略称	R/W	P4 区地址	区域 7 地址	存取长度
看门狗定时器停止时间寄存器	WDTST	R/W	H'FFCC 0000	H'1FCC 0000	32
看门狗定时器控制 / 状态寄存器	WDTCSR	R/W	H'FFCC 0004	H'1FCC 0004	32
看门狗定时器基准停止时间寄存器	WDTBST	R/W	H'FFCC 0008	H'1FCC 0008	32
看门狗定时器计数器	WDCNT	R	H'FFCC 0010	H'1FCC 0010	32
看门狗定时器基准计数器	WDTBCNT	R	H'FFCC 0018	H'1FCC 0018	32

表 29.3 各处理模式的寄存器状态

名称	略称	上电复位		睡眠	待机
		通过 $\overline{\text{PRESET}}$ 引脚	通过 WDT/H-UDI		
看门狗定时器停止时间寄存器	WDTST	H'0000 0000	保持	保持	保持
看门狗定时器控制 / 状态寄存器	WDTCSR	H'0000 0000	保持	保持	保持
看门狗定时器基准停止时间寄存器	WDTBST	H'0000 0000	保持	保持	保持
看门狗定时器计数器	WDCNT	H'0000 0000	保持	保持	保持
看门狗定时器基准计数器	WDTBCNT	H'0000 0000	保持	保持	保持

29.3.1 看门狗定时器停止时间寄存器 (WDTST)

WDTST 是 32 位可读写寄存器，设定 WDTCNT 的上溢值。当设定 H'0000 0001 时，为上溢的最小时间；当设定 H'0000 0000 时，为上溢的最大时间。

必须将高位字节设定为代码值 H'5A，并且以长字为单位写 WDTST。读取的代码值为 0。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	代码值 (H'5A)								—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	WDTST											
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 24	代码值	全 0	R/W	代码值 (H'5A) 读取值为 0，写此寄存器时只能写 H'5A。
23 ~ 12	—	全 0	R	保留位 读取值为 0，只能写 0。
11 ~ 0	WDTST	H'000	R/W	WDTCNT 的上溢值

29.3.2 看门狗定时器控制 / 状态寄存器 (WDTCSR)

WDTCSR 是 32 位可读写寄存器，由选择定时器模式的位和上溢标志构成。

必须将高位字节设定为代码值 H'A5，并且以长字为单位写 WDTCSR。读取的代码值为 0。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	代码值 (H'A5)								—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R
位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TME	WT/IT	—	WOVF	IOVF	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R	R	R

位	位名	初始值	R/W	说明
31 ~ 24	代码值	全 0	R/W	代码值 (H'A5) 读取值为 0，写此寄存器时只能写 H'A5。
23 ~ 8	—	全 0	R	保留位 读取值为 0，只能写 0。
7	TME	0	R/W	定时器允许位 设定定时器运行的开始或者停止。 0: 停止递增计数 1: 开始递增计数
6	WT/IT	0	R/W	定时器模式选择位 指定是用作看门狗定时器还是用作间隔定时器。如果在 WDT 的运行中改写 WT/IT，递增计数可能会不正确。 0: 间隔定时器模式 1: 看门狗定时器模式
5	—	0	R	保留位 读取值为 0，只能写 0。
4	WOVF	0	R/W	看门狗定时器上溢位 在看门狗定时器模式中，表示 WDTCNT 发生上溢。在间隔定时器模式中，不被置位。 0: 无上溢 1: 在看门狗定时器模式中，WDTCNT 发生上溢。
3	IOVF	0	R/W	间隔定时器上溢位 在间隔定时器模式中，表示 WDTCNT 发生上溢。 在看门狗定时器模式中，不被置位。 0: 无上溢 1: 在间隔定时器模式中，WDTCNT 发生上溢。
2 ~ 0	—	全 0	R	保留位 读取值为 0，只能写 0。

29.3.3 看门狗定时器基准停止时间寄存器 (WDTBST)

WDTBST 是 32 位可读写寄存器，进行 WDTBCNT 的清除。

通过以长字为单位将代码值 H'55 写到 WDTBST 的高位字节，进行 WDTBCNT 的清除。WDTBST 的读取值 0。

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	代码值 (H'55)								—	—	—	—	—	—	—	—
初始值:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R
位:	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

29.3.4 看门狗定时器计数器 (WDTCNT)

WDTCNT 是 32 位只读寄存器，通过 WDTBCNT 的上溢进行递增计数。当 WDTCNT 发生上溢时，在看门狗定时器模式时产生所选的复位，而在间隔定时器模式时产生中断。

WDTCNT 的写操作无效。

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

29.3.5 看门狗定时器基准计数器 (WDTBCNT)

WDTBCNT 是 32 位只读寄存器，通过外围时钟 (Pck) 进行递增计数。当 WDTBCNT 发生上溢时，WDTCNT 就进行递增计数，并且将 WDTBCNT 清除为 H'0000 0000。

WDTBCNT 的写操作无效。

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

29.4 运行说明

29.4.1 复位请求

复位有上电复位，以下说明中断发生源。

(1) 上电复位

1. 中断发生源:

- $\overline{\text{PRESET}}$ 引脚输入低电平。
- 在 WDTCSR 的 WT/IT 位为 1 的状态下，WDTCNT 发生上溢。
- 发生 H-UDI 复位。

详细内容请参照“第 31 章 用户调试接口 (H-UDI)”。

2. 转移目标地址: H'A000 0000

3. 转移时的操作:

将异常码 H'000 设定到 EXPEVT。进行 VBR 和 SR 的初始化，并且转移到 PC=H'A000 0000。

通过初始化，VBR 寄存器被设定为 H'0000 0000；SR 的 MD、RB 和 BL 位被置 1，FD 位被清 0，并且中断屏蔽级 (IMASK3 ~ IMASK0) 被设定为 B'1111。

进行 CPU 和外围模块的初始化。详细内容请参照各章的寄存器说明。

必须在上电时给 $\overline{\text{PRESET}}$ 引脚输入低电平。另外，为了进行 H-UDI 的初始化，需要给 $\overline{\text{TRST}}$ 引脚输入低电平。

```
Power_on_reset()  
{  
    EXPEVT = H'0000 0000;  
    VBR = H'0000 0000;  
    SR.MD =1;  
    SR.RB =1;  
    SR.BL =1;  
    SR.(I0-I3)= B'1111;  
    SR.FD =0;  
    Initialize_CPU();  
    Initialize_Module(PowerOn);  
    PC = H'A000 0000;  
}
```

29.4.2 看门狗定时器模式的使用方法

1. 给 WDTST 设定 WDTCNT 的上溢时间。
2. 将 WDTCSR 的 WT/IT 位置 1。
3. 如果将 WDTCSR 的 TME 位置 1，就开始 WDT 计数器的计数。
4. 在看门狗定时器模式中，为了不使 WDTCNT 发生上溢，必须定期清除 WDTCNT 或者 WDTBCNT。清除方法请参照“29.4.5 WDT 计数器的清除方法”。
5. 如果 WDTCNT 发生上溢，就将 WDTCSR 的 WOVF 标志置 1，并且产生上电复位。

29.4.3 间隔定时器模式的使用方法

在间隔定时器模式中，每当计数器发生上溢，就产生间隔定时器中断。因此，能每隔一定时间产生中断。

1. 给 WDTST 设定 WDCNT 的上溢时间。
2. 将 WDTCSR 的 WT/IT 位置 0。
3. 如果将 WDTCSR 的 TME 位置 1，就开始 WDT 计数器的计数。
4. 如果 WDCNT 发生上溢，就将 WDTCSR 的 IOVF 标志置 1，并且产生间隔定时器中断请求。此时，WDCNT 和 WDTBCNT 继续进行计数。

29.4.4 WDT 发生的上溢时间

WDCNT 和 WDTBCNT 的关系如图 29.2 所示。

此图是间隔定时器模式的例子，在 WDCNT 发生上溢后还继续递增计数。

在看门狗定时器模式中，复位解除后将 WDCNT 和 WDTBCNT 清 0，重新开始递增计数。

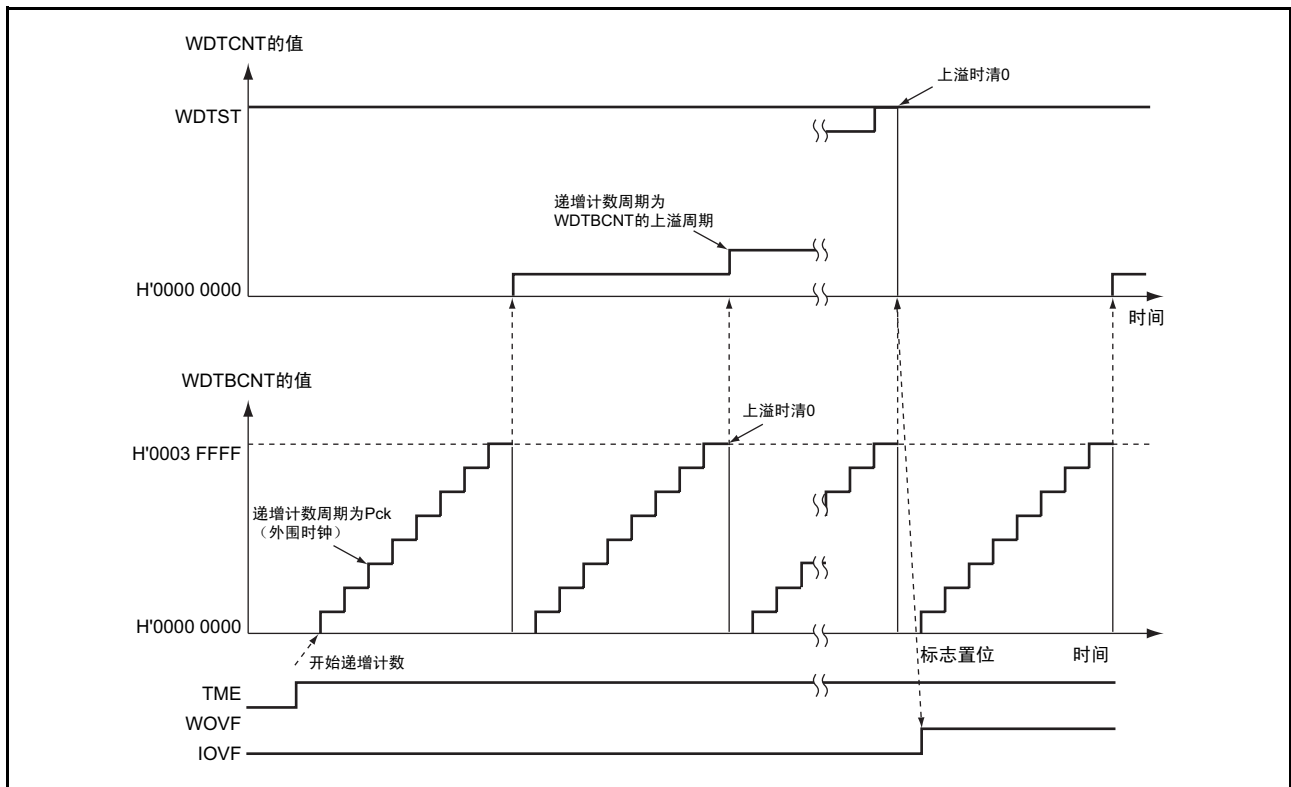


图 29.2 WDT 的递增计数运行（间隔定时器模式的例子）

WDTBCNT 是 18 位计数器，通过外围时钟（Pck）进行递增计数。通过将 H'55 写到 WDTBST 的 bit31 ~ 24 进行 WDTBCNT 的清除。在外围时钟为 50MHz 时，WDTBCNT 的上溢时间约为：

$$2^{18}[\text{bit}] \times 1/50[\text{MHz}] = 5.243[\text{ms}]。$$

WDCNT 是 12 位计数器，在 WDTBCNT 发生上溢时进行递增计数。当给 WDTST 的全部位写 0 时，为上溢的最大时间。在外围时钟为 50MHz 时，上溢的最大时间约为：

$$2^{12}[\text{bit}] \times 5.243[\text{ms}] = 21.47[\text{s}]。$$

当给 WDTST 写 H'0000 0001 时，上溢的最小时间为：

$$2^1[\text{bit}] \times 5.243[\text{ms}] = 5.243[\text{ms}]。$$

29.4.5 WDT 计数器的清除方法

通过以长字为单位将代码值写到 WDTBST 进行 WDTBCNT 的清除，给 WDTST 设定上溢值进行 WDTCNT 的清除。

29.5 复位中的引脚时序

29.5.1 PRESET 引脚的上电复位

为了在转移到上电复位状态时对 PLL 进行复位，需要确保 PLL 的振荡稳定时间。PLL 振荡稳定时间是 PLL1 振荡稳定时间和 PLL2 振荡稳定时间的合计时间。

在使 PRESET 引脚输入从低电平变为高电平后到经过复位保持时间前，LSI 内部继续保持复位状态。复位保持时间至少为 EXTAL 引脚输入的 10240 个时钟周期。

(1) 上电顺序

必须在上电时给 PRESET 引脚输入低电平。另外，为了进行 H-UDI 的初始化，需要给 TRST 引脚输入低电平。

表示复位状态的 STATUS[1:0] 输出时序为异步输出，而表示正常状态的时序是外围时钟 (Pck) 的同步输出，不与 EXTAL 引脚输入、CLKOUT 引脚输出同步。

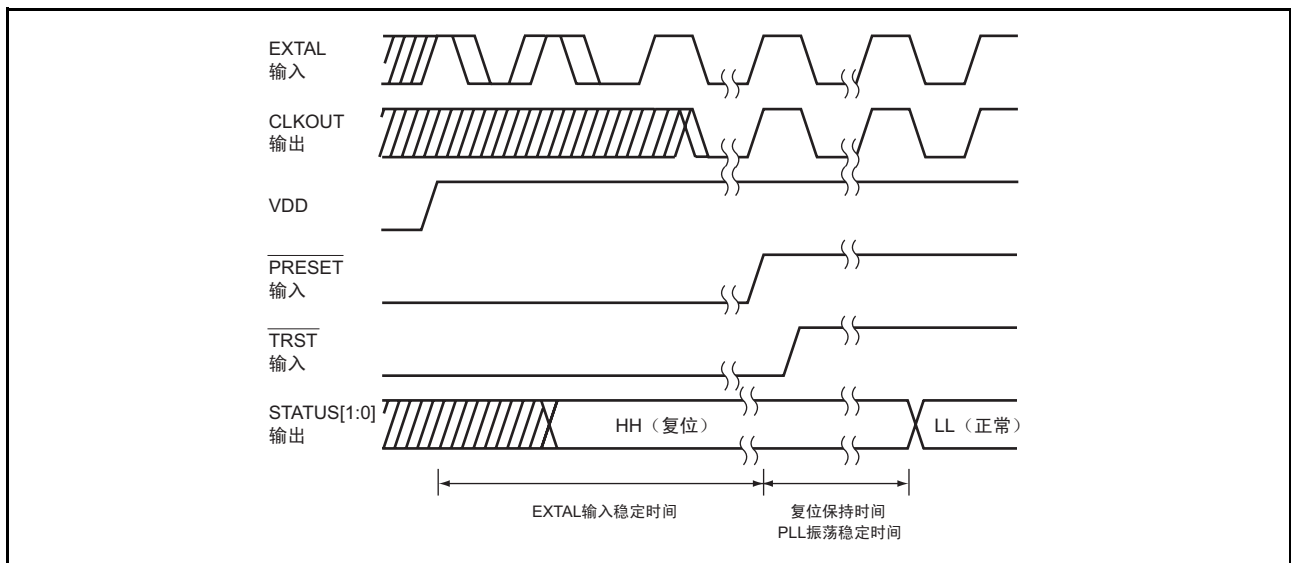
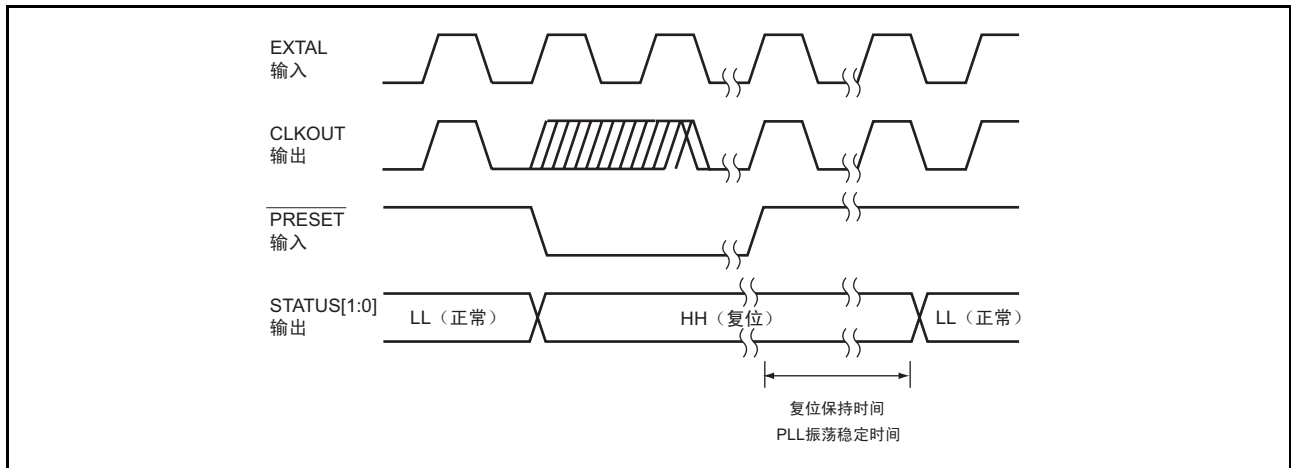


图 29.3 上电顺序

(2) 正常运行中的 $\overline{\text{PRESET}}$ 的上电复位

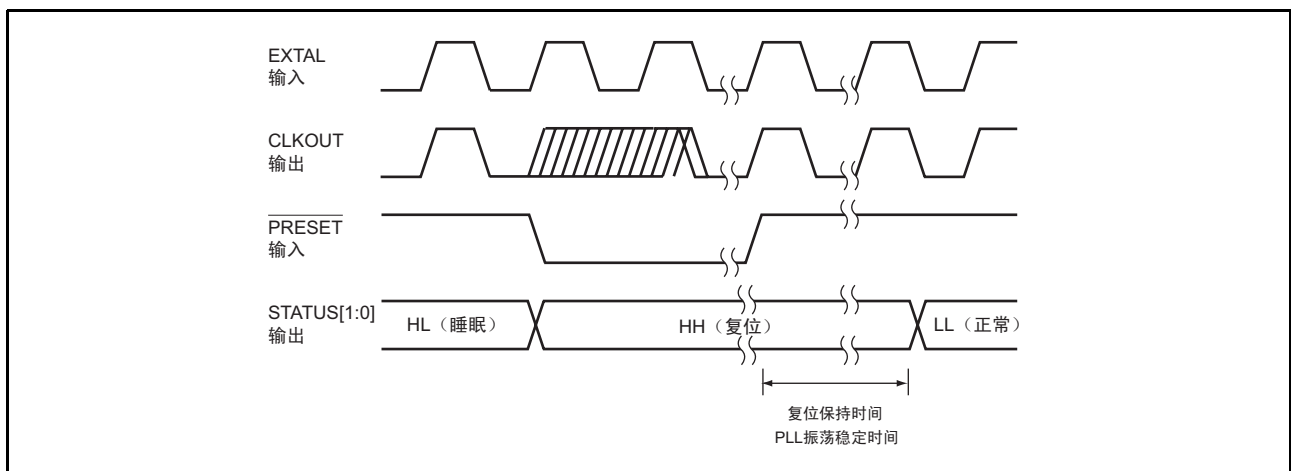
在正常运行中，如果给 $\overline{\text{PRESET}}$ 引脚输入低电平进行上电复位，就需要确保 PLL 的振荡稳定时间。

表示复位状态的 STATUS[1:0] 输出时序为异步输出，而表示正常状态的时序是外围时钟 (Pck) 的同步输出，不与 EXTAL 引脚输入、CLKOUT 引脚输出同步。

图 29.4 正常运行中的 $\overline{\text{PRESET}}$ 引脚的上电复位(3) 睡眠模式中的 $\overline{\text{PRESET}}$ 引脚的上电复位

在睡眠模式中，如果给 $\overline{\text{PRESET}}$ 引脚输入低电平进行上电复位，就需要确保 PLL 的振荡稳定时间。

表示复位状态的 STATUS[1:0] 输出时序为异步输出，而表示正常状态的时序是外围时钟 (Pck) 的同步输出，不与 EXTAL 引脚输入、CLKOUT 引脚输出同步。

图 29.5 睡眠模式中的 $\overline{\text{PRESET}}$ 引脚的上电复位

29.5.2 通过 WDT 上溢产生的上电复位

通过 WDT 上溢产生的上电复位的时间 (WDT 复位时间) 至少为 EXTAL 引脚输入的 9 个时钟周期和 18 个外围时钟 (Pck) 周期。

另外, 从 WDT 发生上溢到转移到上电复位状态为止的时间 (WDT 复位请求时间) 至少为 EXTAL 引脚输入的 1 个时钟周期和 5 个外围时钟 (Pck) 周期。

(1) 正常运行的 WDT 上溢的上电复位

表示复位状态和正常状态的 STATUS[1:0] 输出时序是外围时钟 (Pck) 的同步输出, 而不与 EXTAL 引脚输入、CLKOUT 引脚输出同步。

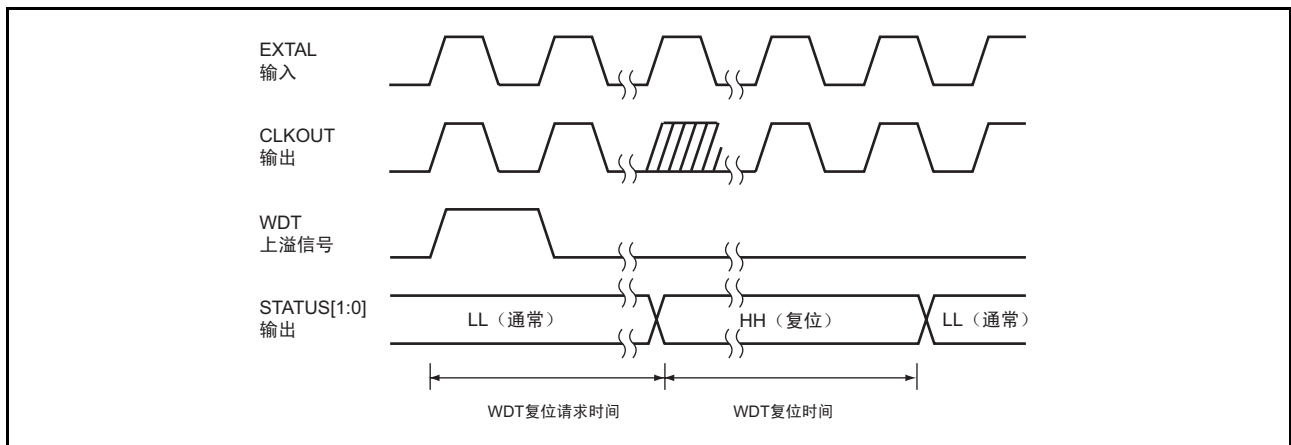


图 29.6 正常运行中的 WDT 上溢的上电复位

(2) 睡眠模式中的 WDT 上溢的上电复位

表示复位状态和正常状态的 STATUS[1:0] 输出时序是外围时钟 (Pck) 的同步输出, 而不与 EXTAL 引脚输入、CLKOUT 引脚输出同步。

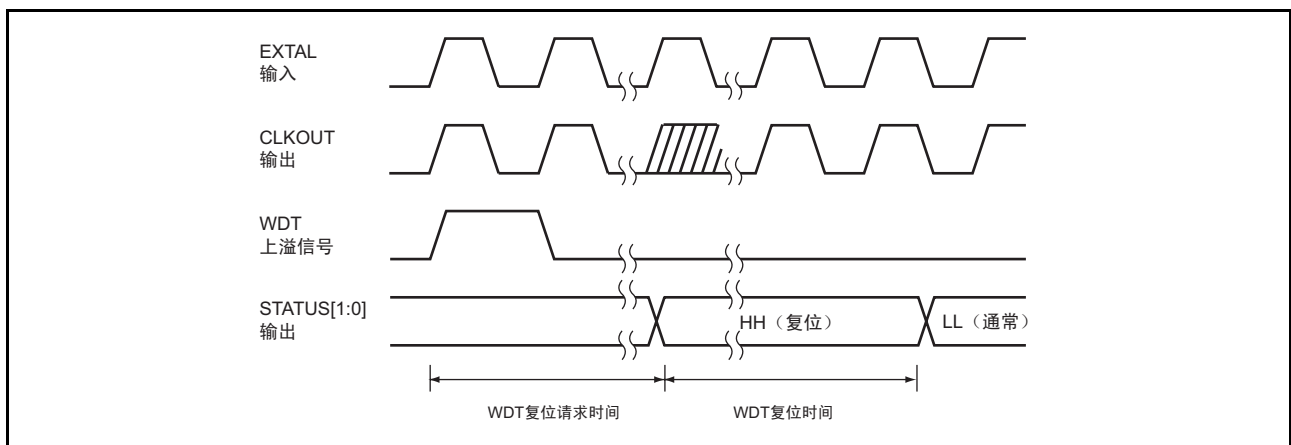


图 29.7 睡眠模式中的 WDT 上溢的上电复位

第 30 章 用户断点控制器 (UBC)

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

30.1 特点

1. 能设定以下的断点比较条件:

断点通道数: 2个通道 (通道0和通道1)

能将用户断点条件设定为通道0和通道1独立或者连续的条件 (连续断点是指: 通道0的断点条件成立后通道1的断点条件成立, 或者在通道1的断点条件成立后通道0的断点条件成立, 并且两者不发生在同一个总线周期。)

- 地址

在对由 ASID 和 32 位地址构成的 40 位地址进行比较时, 能对 ASID 选择是比较全部位还是屏蔽全部位。

对于 32 位地址, 能按位进行屏蔽。用户能用低 12 位 (4K 字节页)、低 10 位 (1K 字节页) 或者任意大小的页进行地址屏蔽。

- 数据

只能对通道1进行32位的屏蔽。

- 总线周期

取指令的断点 (PC 断点) 或者存取操作数的断点

- 读写操作

- 操作数长度

支持字节、字、长字和4个字。

2. 能执行用户指定的用户断点条件异常处理程序。
3. 对于 PC 断点, 能指定是将断点设定在指令执行前还是设定在指令执行后。
4. 作为断点条件 (只对通道1), 最多能指定 $2^{12}-1$ 次的重复次数。

UBC 的框图如图 30.1 所示。

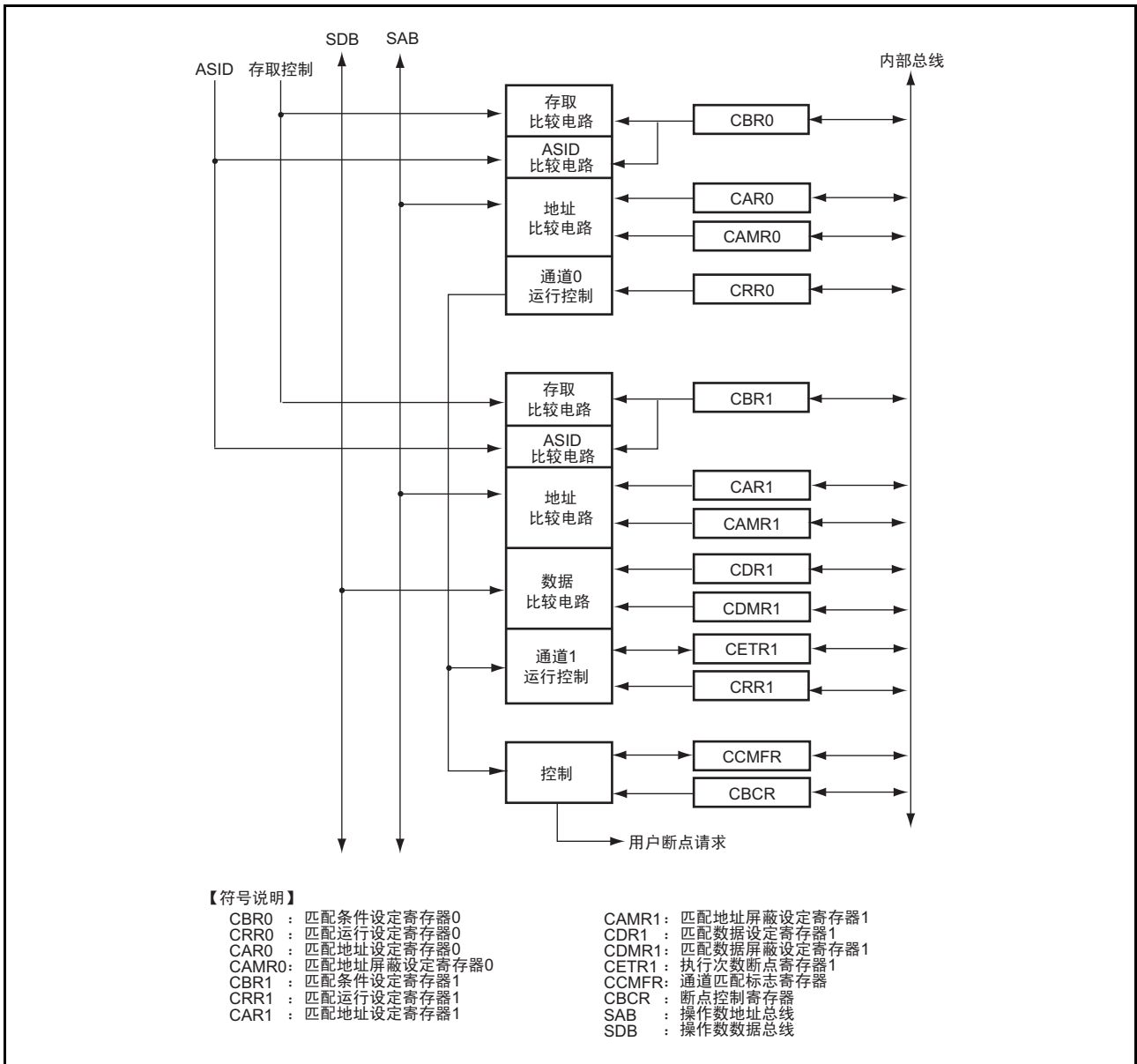


图 30.1 UBC 的框图

30.2 寄存器说明

UBC 中有以下寄存器。

表 30.1 寄存器结构

名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度
匹配条件设定寄存器 0	CBR0	R/W	H'FF20 0000	H'1F20 0000	32
匹配运行设定寄存器 0	CRR0	R/W	H'FF20 0004	H'1F20 0004	32
匹配地址设定寄存器 0	CAR0	R/W	H'FF20 0008	H'1F20 0008	32
匹配地址屏蔽设定寄存器 0	CAMR0	R/W	H'FF20 000C	H'1F20 000C	32
匹配条件设定寄存器 1	CBR1	R/W	H'FF20 0020	H'1F20 0020	32
匹配运行设定寄存器 1	CRR1	R/W	H'FF20 0024	H'1F20 0024	32
匹配地址设定寄存器 1	CAR1	R/W	H'FF20 0028	H'1F20 0028	32
匹配地址屏蔽设定寄存器 1	CAMR1	R/W	H'FF20 002C	H'1F20 002C	32
匹配数据设定寄存器 1	CDR1	R/W	H'FF20 0030	H'1F20 0030	32
匹配数据屏蔽设定寄存器 1	CDMR1	R/W	H'FF20 0034	H'1F20 0034	32
执行次数断点寄存器 1	CETR1	R/W	H'FF20 0038	H'1F20 0038	32
通道匹配标志寄存器	CCMFR	R/W	H'FF20 0600	H'1F20 0600	32
断点控制寄存器	CBCR	R/W	H'FF20 0620	H'1F20 0620	32

【注】 * P4 区地址是使用虚拟地址空间 P4 区的地址，区域 7 地址是使用 TLB 存取物理地址空间区域 7 的地址。

表 30.2 各处理状态的寄存器状态

名称	略称	上电复位	睡眠	待机
匹配条件设定寄存器 0	CBR0	H'2000 0000	保持	保持
匹配运行设定寄存器 0	CRR0	H'0000 2000	保持	保持
匹配地址设定寄存器 0	CAR0	不定值	保持	保持
匹配地址屏蔽设定寄存器 0	CAMR0	不定值	保持	保持
匹配条件设定寄存器 1	CBR1	H'2000 0000	保持	保持
匹配运行设定寄存器 1	CRR1	H'0000 2000	保持	保持
匹配地址设定寄存器 1	CAR1	不定值	保持	保持
匹配地址屏蔽设定寄存器 1	CAMR1	不定值	保持	保持
匹配数据设定寄存器 1	CDR1	不定值	保持	保持
匹配数据屏蔽设定寄存器 1	CDMR1	不定值	保持	保持
执行次数断点寄存器 1	CETR1	不定值	保持	保持
通道匹配标志寄存器	CCMFR	H'0000 0000	保持	保持
断点控制寄存器	CBCR	H'0000 0000	保持	保持

存取长度必须与控制寄存器的长度相同。如果长度不同，就在寄存器的写操作时不执行写操作，而在读操作时返回不定值。从执行控制寄存器的改写指令到实际反映被写值的期间，可能不产生希望的断点。为了得知控制寄存器的更改时序，必须读取最后被写的的数据。从而后续指令对刚写的寄存器值有效。

30.2.1 匹配条件设定寄存器 0、1 (CBR0、CBR1)

CBR0 和 CBR1 是 32 位可读写寄存器，分别指定通道 0 和通道 1 的断点条件。能设定的断点条件是：

1. 是否有匹配标志
2. 是否有 ASID 及其值
3. 是否有数值
4. 操作数长度
5. 是否有执行次数
6. 总线
7. 取指令或者存取操作数
8. 读写操作

• CBR0

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MFE	AIE	MFI						AIV							
初始值:	0	0	1	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
	—	SZ				—	—	—	—	CD	ID	—	RW	CE		
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/W	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31	MFE	0	R/W	匹配标志允许位 指定在匹配条件中是否包含此寄存器的 MFI 位指定的匹配标志的值。 在指定的匹配标志的值为 1 时，满足条件。 0: 匹配条件不包含匹配标志，不检查。 1: 匹配条件包含匹配标志。
30	AIE	0	R/W	ASID 允许位 指定在匹配条件中是否包含此寄存器的 AIV 位指定的 ASID。 0: 匹配条件不包含 ASID，不检查。 1: 匹配条件包含 ASID。
29 ~ 24	MFI	100000	R/W	匹配标志指定位 指定作为匹配条件的匹配标志。 000000: CCMFR 寄存器的 MF0 位 000001: CCMFR 寄存器的 MF1 位 其他: 保留 (禁止设定) 【注】 初始值为保留 (禁止设定) 状态。在将 CBR0[0] 位置 1 时，必须设定 000000 或者 000001。另外，必须注意: 在 CCRMFR 寄存器的 MF0 位为 0 时，如果将此寄存器的 MFE 位置 1，MFI 位置 000000，就不会命中通道 0。
23 ~ 16	AIV	H'00	R/W	ASID 指定位 指定作为匹配条件的 ASID 值。
15	—	0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。

位	位名	初始值	R/W	说明
14 ~ 12	SZ	全 0	R/W	<p>操作数长度选择位</p> <p>指定作为匹配条件的操作数长度。此位只在以操作数存取周期为对象时有效。</p> <p>000: 匹配条件不包含操作数长度, 不检查 (任何操作数长度都为对象) *1。</p> <p>001: 以字节存取为对象</p> <p>010: 以字存取为对象</p> <p>011: 以长字存取为对象</p> <p>100: 以 4 个字存取为对象 *3</p> <p>其他: 保留 (禁止设定)</p>
11 ~ 8	—	全 0	R	<p>保留位</p> <p>有关此位的读写, 请参照“产品使用上的注意事项”。</p>
7、6	CD	00	R/W	<p>总线选择位</p> <p>指定作为匹配条件的总线。此位只在以操作数存取周期为对象时有效。</p> <p>00: 在存取操作数时, 以操作数总线为对象。</p> <p>其他: 保留 (禁止设定)</p>
5、4	ID	00	R/W	<p>取指令 / 存取操作数选择位</p> <p>指定作为匹配条件的取指令周期或者操作数存取周期。</p> <p>00: 以取指令周期或者操作数存取周期为对象</p> <p>01: 以取指令周期为对象</p> <p>10: 操作数存取周期为对象</p> <p>11: 以取指令周期或者操作数存取周期为对象</p>
3	—	0	R	<p>保留位</p> <p>有关此位的读写, 请参照“产品使用上的注意事项”。</p>
2、1	RW	00	R/W	<p>总线指令选择位</p> <p>指定作为匹配条件的读周期或者写周期。此位只在以操作数存取周期为对象时有效。</p> <p>00: 以读周期或者写周期为对象</p> <p>01: 以读周期为对象</p> <p>10: 以写周期为对象</p> <p>11: 以读周期或者写周期为对象</p>
0	CE	0	R/W	<p>通道允许位</p> <p>指定通道是否有效。在此位置 0 时, 其他位的设定全部无效。</p> <p>0: 通道无效</p> <p>1: 通道有效</p>

• CBR1

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MFE	AIE	MFI						AIV							
初始值:	0	0	1	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
	DBE	SZ			ETBE	—	—	—	CD	ID		—	RW	CE		
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	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31	MFE	0	R/W	匹配标志允许位 指定是否在匹配条件中包含此寄存器的 MFI 位指定的匹配标志值。在指定的匹配标志值为 1 时，满足条件。 0: 匹配条件不包含匹配标志，不检查。 1: 匹配条件包含匹配标志。
30	AIE	0	R/W	ASID 允许位 指定在匹配条件中是否包含此寄存器的 AIV 位指定的 ASID。 0: 匹配条件不包含 ASID，不检查。 1: 匹配条件包含 ASID
29 ~ 24	MFI	100000	R/W	匹配标志指定位 指定作为匹配条件的匹配标志。 000000: CCMFR 寄存器的 MF0 位 000001: CCMFR 寄存器的 MF1 位 其他: 保留 (禁止设定) 【注】 初始值为保留 (禁止设定) 状态。在将 CBR1[0] 位置 1 时，必须设定 000000 或者 000001。另外，必须注意: 在 CCRMFR 寄存器的 MF1 位为 0 时，如果将此寄存器的 MFE 位置 1，MFI 位置 000001，就不会命中通道 1。
23 ~ 16	AIV	全 0	R/W	ASID 指定位 指定作为匹配条件的 ASID 值。
15	DBE	0	R/W	数值允许位 *2 指定在匹配条件中是否包含数值。此位只在以操作数存取周期为对象时有效。 0: 匹配条件不包含数值，不检查。 1: 匹配条件包含数值。
14 ~ 12	SZ	000	R/W	操作数长度选择位 指定作为匹配条件的操作数长度。此位只在以操作数存取周期为对象时有效。 000: 匹配条件不包含操作数长度，不检查 (任何操作数长度都为对象) *1。 001: 以字节存取为对象 010: 以字存取为对象 011: 以长字存取为对象 100: 以 4 个字存取为对象 *3 其他: 保留 (禁止设定)

位	位名	初始值	R/W	说明
11	ETBE	0	R/W	<p>执行次数允许位</p> <p>指定在匹配条件中是否包含执行次数。在此位为 1 时，如果匹配条件成立的次数和 CETR1 寄存器指定的值相等，就产生 CRR1 寄存器指定的运行。</p> <p>0: 匹配条件不包含执行次数，不检查。 1: 匹配条件包含执行次数。</p>
10 ~ 8	—	全 0	R	<p>保留位</p> <p>有关此位的读写，请参照“产品使用上的注意事项”。</p>
7、6	CD	00	R/W	<p>总线选择位</p> <p>指定作为匹配条件的总线。此位只在以操作数存取周期为对象时有效。</p> <p>00: 在存取操作数时，以操作数总线为对象。 其他: 保留 (禁止设定)</p>
5、4	ID	00	R/W	<p>取指令 / 存取操作数选择位</p> <p>指定作为匹配条件的取指令周期或者操作数存取周期。</p> <p>00: 以取指令周期或者操作数存取周期为对象 01: 以取指令周期为对象 10: 以操作数存取周期为对象 11: 以取指令周期或者操作数存取周期为对象</p>
3	—	0	R	<p>保留位</p> <p>有关此位的读写，请参照“产品使用上的注意事项”。</p>
2、1	RW	00	R/W	<p>总线命令选择位</p> <p>指定作为匹配条件的读周期或者写周期。此位只在以操作数存取周期为对象时有效。</p> <p>00: 以读周期或者写周期为对象 01: 以读周期为对象 10: 以写周期对象 11: 以读周期或者写周期为对象</p>
0	CE	0	R/W	<p>通道允许位</p> <p>指定通道是否有效。在此位置 0 时，其他位的设定全部无效。</p> <p>0: 通道无效 1: 通道有效</p>

【注】 *1 在匹配条件包含数值时，必须指定操作数长度。

*2 OCBI 指令被处理为无数值的长字写存取， PREF、OCBP 和 OCBWB 被处理为无数值的长字读存取。在对这些指令设定匹配条件时，不能包含数值。

*3 在指定 4 个字存取并且匹配条件包含数值时，64 位数据的高 32 位和低 32 位分别与匹配数据设定寄存器和匹配数据屏蔽设定寄存器的内容进行比较。

30.2.2 匹配运行设定寄存器 0、1 (CRR0、CRR1)

CRR0 和 CRR1 是 32 位可读写寄存器，分别指定通道 0 和通道 1 满足匹配条件时的运行。能设定的运行内容是：

1. 取指令周期的断点时序
2. 是否暂停

- CRR0

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PCB	BIE
初始值:	0	0	1	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
31 ~ 14	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。
13	—	1	R	保留位 读取值为 1，只能写 1。
12 ~ 2	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。
1	PCB	0	R/W	PC 断点选择位 指定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。对于取指令周期以外的断点，此位无效。 0: 将 PC 断点设定在指令执行前。 1: 将 PC 断点设定在指令执行后。
0	BIE	0	R/W	暂停允许位 指定在通道的匹配条件成立时是否暂停。 0: 不暂停 1: 暂停

- CRR1

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PCB	BIE
初始值:	0	0	1	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
31 ~ 14	—	全 0	R	保留位 有关此位的读写, 请参照“产品使用上的注意事项”。
13	—	1	R	保留位 读取值为 1, 只能写 1。
12 ~ 2	—	全 0	R	保留位 有关此位的读写, 请参照“产品使用上的注意事项”。
1	PCB	0	R/W	PC 断点选择位 指定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。对于取指令周期以外的断点, 此位无效。 0: 将 PC 断点设定在指令执行前 1: 将 PC 断点设定在指令执行后
0	BIE	0	R/W	暂停允许位 指定在通道的匹配条件成立时是否暂停。 0: 不暂停 1: 暂停

30.2.3 匹配地址设定寄存器 0、1 (CAR0、CAR1)

CAR0 和 CAR1 是 32 位可读写寄存器, 指定通道 0 和通道 1 的断点条件的虚拟地址。

• CAR0

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

位	位名	初始值	R/W	说明
31 ~ 0	CA	不定值	R/W	比较地址 指定断点条件的地址。 在通过 CBR0 寄存器指定操作数总线时, 给 CA[31:0] 指定 SAB 的地址。

• CAR1

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

位	位名	初始值	R/W	说明
31 ~ 0	CA	不定值	R/W	比较地址 指定作为断点条件的地址。 在通过 CBR1 寄存器指定操作数总线时，给 CA[31:0] 指定 SAB 的地址。

30.2.4 匹配地址屏蔽设定寄存器 0、1 (CAMR0、CAMR1)

CAMR0 和 CAMR1 是 32 位可读写寄存器，指定对应通道的匹配地址设定寄存器的地址位中的屏蔽位（将要屏蔽的位置 1）。

• CAMR0

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

位	位名	初始值	R/W	说明
31 ~ 0	CAM	不定值	R/W	比较地址屏蔽位 指定 CAR0 寄存器的地址位中的屏蔽位（将要屏蔽的位置 1）。 0: 断点条件包含地址位 CA[n]。 1: 屏蔽地址位 CA[n]，断点条件不包含地址位 CA[n]。 【注】 n=31 ~ 0

• CAMR1

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

位	位名	初始值	R/W	说明
31 ~ 0	CAM	不定值	R/W	比较地址屏蔽 指定 CAR1 寄存器的地址位中的屏蔽位（将要屏蔽的位置 1）。 0: 断点条件包含地址位 CA[n]。 1: 屏蔽地址位 CA[n]，断点条件不包含地址位 CA[n]。 【注】 n=31 ~ 0

30.2.5 匹配数据设定寄存器 1 (CDR1)

CDR1 是 32 位可读写寄存器，指定通道 1 的断点条件的数值。

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

位	位名	初始值	R/W	说明
31 ~ 0	CD	不定值	R/W	比较数值 指定断点条件的数值。 在通过 CBR1 寄存器指定操作数总线时，给 CD[31:0] 指定 SDB 的数值。

表 30.3 匹配数据设定寄存器的指定

用 CBR1 进行总线选择	CD[31:24]	CD[23:16]	CD[15:8]	CD[7:0]
操作数总线 (字节)	Don't care			SDB7 ~ 0
操作数总线 (字)	Don't care		SDB15 ~ 0	
操作数总线 (长字)	SDB31 ~ 0			

- 【注】**
1. 在匹配条件包含数值时，必须指定操作数长度。
 2. OCBI 指令被处理为无数值的长字写存取，PREF、OCBP 和 OCBWB 被处理为无数值的长字读存取。在对这些指令设定匹配条件时，不能包含数值。
 3. 在指定 4 个字存取并且匹配条件包含数值时，64 位数据的高 32 位和低 32 位分别与匹配数据设定寄存器和匹配数据屏蔽设定寄存器的内容进行比较。

30.2.6 匹配数据屏蔽设定寄存器 1 (CDMR1)

CDMR1 是 32 位可读写寄存器，指定匹配数据设定寄存器的数值中的屏蔽位（将要屏蔽的位置 1）。

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

位	位名	初始值	R/W	说明
31 ~ 0	CDM	不定值	R/W	比较数值屏蔽位 指定 CDR1 寄存器的数值中的屏蔽位（将要屏蔽的位置 1）。 0: 断点条件包含数值位 CD[n]。 1: 屏蔽数值位 CD[n]，断点条件不包含数值位 CD[n]。 【注】 n=31 ~ 0

30.2.7 执行次数断点寄存器 1 (CETR1)

CETR1 是 32 位可读写寄存器，指定暂停前的通道命中次数，能指定的最大值为 $2^{12}-1$ 。在匹配条件设定寄存器指定了匹配条件的包含执行次数时，通道每命中一次此寄存器的值就减 1，如果在寄存器值变为 H'001 后又命中就暂停。

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
	—	—	—	—	CET											
初始值:	0	0	0	0	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

位	位名	初始值	R/W	说明
31 ~ 12	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。
11 ~ 0	CET	不定值	R/W	执行次数 指定断点条件的执行次数。

30.2.8 通道匹配标志寄存器 (CCMFR)

CCMFR 是 32 位可读写寄存器，表示各通道的匹配条件是否成立。如果通道的匹配条件成立，就将对应的标志位置 1。在清除标志时，将要清除的位置 0 而将其他位置 1 的数据写到此寄存器（将写的值和当前值进行逻辑与，然后写寄存器）。能通过此匹配标志实现多个通道的连续运行。

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MF1	MF0
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
31 ~ 2	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。
1	MF1	0	R/W	通道 1 条件成立标志位 当通道 1 设定的匹配条件成立时，就将此标志置 1。如果将此位置 0，就清除标志。 0: 通道 1 的匹配条件不成立 1: 通道 1 的匹配条件成立
0	MF0	0	R/W	通道 0 条件成立标志位 当通道 0 设定的匹配条件成立时，就将此标志置 1。如果将此位置 0，就清除标志。 0: 通道 0 的匹配条件不成立 1: 通道 0 的匹配条件成立

30.2.9 断点控制寄存器 (CBCR)

CBCR 是 32 位可读写寄存器，指定是否使用用户断点调试支持功能。有关用户断点调试支持功能的详细内容，请参照“30.4 用户断点调试支持功能”。

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UBDE
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
31 ~ 1	—	全 0	R	保留位 有关此位的读写，请参照“产品使用上的注意事项”。
0	UBDE	0	R/W	用户断点调试支持功能允许位 指定是否使用用户断点调试支持功能。 0: 不使用调试支持功能 1: 使用调试支持功能

30.3 运行说明

30.3.1 有关存取的术语说明

所谓“取指令”是指获取指令的存取。例如，在执行转移指令后，获取转移目标的指令。所谓“操作数存取”是指由指令执行的任意存储器存取。例如，对指令 `MOV.W @(disp,PC), Rn` 的地址 ($PC+disp \times 2+4$) 的存取。“数据”用于“地址”的对照。

所有的操作数存取分为读存取和写存取。在以下指令时需要特别注意：

- `PREF`、`OCBP` 和 `OCBWB` 指令：作为读存取处理。
- `MOVCA.L` 和 `OCBI` 指令：作为写存取处理。
- `TAS.B` 指令：作为 1 个读存取或者 1 个写存取处理。

`PREF`、`OCBP`、`OCBWB` 和 `OCBI` 指令的操作数存取是无数值的存取，在对这些指令设定匹配条件时，不能包含数值。

所有的操作数存取都要定义操作数长度。操作数长度有：字节、字、长字和 4 个字。对于 `PREF`、`OCBP`、`OCBWB`、`MOVCA.L` 和 `OCBI` 指令的操作数存取，操作数长度被定义为长字。

30.3.2 用户断点的运行流程

从设定断点条件到用户断点异常处理的运行流程如下所示：

1. 通过匹配条件设定寄存器 (`CBR0` 或者 `CBR1`) 指定作为匹配条件的操作数长度、总线、取指令/操作数存取、读写条件；通过匹配地址设定寄存器 (`CAR0` 或者 `CAR1`) 指定断点地址；通过匹配地址屏蔽设定寄存器 (`CAMR0` 或者 `CAMR1`) 指定地址的屏蔽条件。在匹配条件包含 `ASID` 时，将匹配条件设定寄存器的 `AIE` 位置位，并且通过 `AIV` 位指定 `ASID`；在匹配条件包含数值时，将匹配条件设定寄存器的 `DBE` 位置位，并且通过匹配数据设定寄存器 (`CDR1`) 指定断点数据，通过匹配数据屏蔽设定寄存器 (`CDMR1`) 指定数据的屏蔽条件；在匹配条件包含执行次数时，将匹配条件设定寄存器的 `ETBE` 位置位，并且通过执行次数断点寄存器 (`CETR1`) 指定执行次数的条件。在设定连续断点时，将匹配条件设定寄存器的 `MFE` 位置位，并且通过 `MFI` 位指定第 1 通道号。
2. 通过匹配运行设定寄存器 (`CRR0` 或者 `CRR1`) 指定匹配条件成立时是否暂停，指定取指令条件成立时的断点位置。在匹配条件设定寄存器的 `CE` 位以外的位和其它寄存器的设定完成后，必须将匹配条件设定寄存器的 `CE` 位置位，并且再次读取匹配条件设定寄存器。通过此步骤，能保证控制寄存器的设定值从后续指令开始有效。复位后，如果通过初始状态的控制寄存器将匹配条件设定寄存器的 `CE` 位置位，就有可能发生意想不到的暂停。
3. 如果匹配条件成立，就将通道匹配标志寄存器 (`CCMFR`) 的对应条件成立标志 (`MF1` 或者 `MF0`) 置位，并且根据匹配运行设定寄存器 (`CRR0`、`CRR1`) 的设定内容，向 CPU 请求暂停。对于暂停请求，CPU 的处理因 `SR` 寄存器的 `BL` 位的值而不同，在 `BL` 位为 0 时，接受暂停请求，进行预先决定的异常处理；在 `BL` 位为 1 时，不进行异常处理。
4. 能使用对应的条件成立标志 (`MF1` 或者 `MF0`)，检查匹配条件是否成立。标志在条件成立时被置位，但不自动清除。为了能再次使用标志，必须通过存储器存储指令，将通道匹配标志寄存器 (`CCMFR`) 的标志位置 0。
5. 通道 0 和通道 1 设定的断点条件有可能几乎同时成立。此时，即使只向 CPU 请求 1 个暂停，对应这些断点的 2 个条件成立标志也可能被置位。
6. 在 `SR` 寄存器的 `BL` 位为 1 时，不接受任何暂停请求。但是，进行条件判断，在条件成立时，将条件成立标志置位。
7. 如果指定顺序条件，就在各通道的匹配条件成立时将对应的条件成立标志置位。在第 1 通道的匹配条件成立而第 2 通道的匹配条件不成立时，为了清除第 1 通道的匹配状态，必须清除第 1 通道的条件成立标志。

30.3.3 取指令周期的断点

1. 如果给匹配条件设定寄存器 (CBR0或者CBR1) 设定取指令周期, 匹配条件就是取指令周期。为了在匹配条件成立时暂停, 必须将对应该通道的匹配运行寄存器 (CRR0或者CRR1) 的BIE位置位。能通过PCB位指定将断点设定在指令执行前还是在指令执行后。在将取指令周期指定为匹配条件时, 必须将匹配地址设定寄存器 (CAR0或者CAR1) 的LSB位清0。在此位置1时不暂停。
2. 如果将断点设定在取指令周期中的指令执行前, 就在取指令后并且在执行前暂停。因此, 此功能不能用于因溢出 (在转移或者断点转移中取的没被执行的指令) 而取的指令。有关指令执行前的断点与其它异常的优先顺序, 请参照“第5章 异常处理”。如果对延迟转移指令的延迟槽指定指令执行前的断点, 就在延迟转移指令执行前暂停。但是, 不能对RTE指令的延迟槽指定指令执行前的断点。
3. 如果将断点设定在取指令周期中的指令执行后, 就在匹配条件成立的指令执行后并且在下一条指令执行前暂停。与指令执行前的断点相同, 不能用于因溢出而取的指令。有关指令执行后的断点与其它异常的优先顺序, 请参照“第5章 异常处理”。如果对延迟转移指令的延迟槽指定指令执行后的断点, 就不会在转移目标的第一条指令执行前暂停。
4. 如果将取指令周期指定为通道1的匹配条件, 匹配条件设定寄存器CBR1的DBE位就无效, 忽视匹配数据设定寄存器CDR1和匹配数据屏蔽设定寄存器CDMR1的设定。因此, 不能对取指令周期的断点设定数值。

30.3.4 操作数存取周期的断点

1. 对操作数存取周期的断点, 匹配条件设定寄存器 (CBR0或者CBR1) 的操作数长度指定和条件比较对象地址的关系如下所示。

表 30.4 指定操作数长度和比较对象地址

操作数长度的选择	比较地址位
4 个字	地址位 A31 ~ A3
长字	地址位 A31 ~ A2
字	地址位 A31 ~ A1
字节	地址位 A31 ~ A0
不包含在匹配条件中	4 个字存取时的地址位 A31 ~ A3 长字存取时的地址位 A31 ~ A2 字存取时的地址位 A31 ~ A1 字节存取时的地址位 A31 ~ A0

例如, 在给匹配地址设定寄存器 (CAR0或者CAR1) 设定地址H'0000 1003时, 匹配条件成立的操作数存取周期 (假设其它条件都成立时) 包含以下含义:

对地址H'00001000进行长字存取

对地址H'00001002进行字存取

对地址H'00001003进行字节存取

2. 在通道 1 的匹配条件中包含数值时
当匹配条件包含数值时，需要通过匹配条件设定寄存器 (CBR1) 的操作数长度选择位 (SZ 位) 选择 4 个字、长字、字或者字节，并且设定匹配数据设定寄存器 (CDR1) 和匹配数据屏蔽设定寄存器 (CDMR1)。此时，如果地址条件和数据条件成立，匹配条件就成立。对应字节存取、字存取、长字存取的数值和屏蔽分别设定在 CDR1 和 CDMR1 的 bit7~0、bit15~0、bit31~0。在 4 个字存取时，64 位数据分为高 32 位和低 32 位，作为 2 个 32 位数据进行条件比较。当 2 个 32 位数据中的任意一个满足匹配条件时，匹配条件就成立。
3. PREF、OCBP、OCBWB 和 OCBI 指令的操作数存取是无数值的存取。如果这些指令的匹配条件包含数值，匹配条件就不成立。
4. 如果选择操作数总线，就在条件成立的指令执行结束后并且在下一条指令执行前暂停。但是，在匹配条件包含数值时，也可能在从条件成立的指令开始执行数条指令后暂停，并且无法确定暂停的指令。另外，如果延迟转移指令或者该延迟槽中发生此类断点，就不会在转移目标的第一条指令执行前暂停。不能对 RTE 指令的延迟槽设定操作数断点。另外，在匹配条件包含数值时，不能在 RTE 指令的 1~6 条指令前设定断点。

30.3.5 连续断点

1. 通过设定匹配条件设定寄存器 (CBR0 和 CBR1) 的 MFE 位和 MFI 位，能指定连续条件（在通道 0 的匹配条件成立后通道 1 的匹配条件成立，或者相反）。对于第 1 通道，必须清除匹配条件设定寄存器的 MFE 位和匹配运行设定寄存器的 BIE 位；对于第 2 通道，必须将匹配条件设定寄存器的 MFE 位置位，并且通过 MFI 位指定第 1 通道号。通过第 2 通道的匹配运行设定寄存器的 BIE 位指定在连续条件成立时是否暂停。如果指定连续条件，就在各通道的匹配条件成立时将对应的条件成立标志置位。在第 1 通道的匹配条件成立而第 2 通道的匹配条件不成立时，为了清除第 1 通道的匹配状态，必须清除第 1 通道的条件成立标志。
2. 在指定连续条件时，也能对于通道 1 指定执行次数断点条件。
3. 在第 1 通道和第 2 通道的匹配条件成立时序接近时，可能会出现不保证连续条件的情况。
 - 第 1 通道和第 2 通道都在取指令周期匹配条件成立

指令 B 在指令 A 之后	与设定同样的地址等效。不能使用此设定。
指令 B 在指令 A 的 1 条指令之后	不保证连续运行。
指令 B 在指令 A 的 2 条或者 2 条以上指令之后	保证连续运行。

- 第 1 通道在取指令周期匹配条件成立，第 2 通道在操作数存取周期匹配条件成立

指令 B 在指令 A 之后或者指令 A 的 1 条指令之后	不保证连续运行。
指令 B 在指令 A 的 2 条或者 2 条以上指令之后	保证连续运行。

- 第 1 通道在操作数存取周期匹配条件成立，第 2 通道在取指令周期匹配条件成立

指令 B 在指令 A 的 0~5 条指令之后	不保证连续运行。
指令 B 在指令 A 的 6 条或者 6 条以上指令之后	保证连续运行。

- 第 1 通道和第 2 通道都在操作数存取周期匹配条件成立

指令 B 在指令 A 的 0~5 条指令之后	不保证连续运行。
指令 B 在指令 A 的 6 条或者 6 条以上指令之后	保证连续运行。

30.3.6 被保存的程序计数器的值

在暂停时，将要继续执行的指令的地址保存到 SPC，并且转换到异常处理状态。除了匹配条件包含数值的情况以外，能确定暂停的指令。

1. 将取指令周期（指令执行前）指定为匹配条件时
将匹配条件成立的指令地址保存到 SPC，不执行条件成立的指令而在指令执行前暂停。但是，如果在延迟槽指令时条件成立，就将延迟转移指令的地址保存到 SPC。
2. 将取指令周期（指令执行后）指定为匹配条件时
将匹配条件成立的指令的下一条指令地址保存到 SPC，执行条件成立的指令，并且在下一条指令执行前暂停。如果在延迟转移指令或者该延迟槽指令时条件成立，就执行其指令，并且将转移目标地址保存到 SPC。
3. 将操作数存取（只限地址）指定为匹配条件时
将断点条件成立的指令的下一条指令地址保存到 SPC，执行条件成立的指令，并且在下一条指令执行前暂停。但是，如果在延迟槽指令时条件成立，就将转移目标地址保存到 SPC。
4. 将操作数存取（地址+数据）指定为匹配条件时
如果给匹配条件追加数值，就执行匹配条件成立的指令，并且在执行 1 条指令～6 条指令的任意指令前发生用户断点暂停，将该指令地址保存到 SPC。无法准确地确定暂停的位置。如果在延迟槽指令时条件成立，就将转移目标地址保存到 SPC。如果条件成立的指令的下一条指令是转移指令，就有可能在执行转移指令或者延迟槽指令后暂停。此时，也将转移目标地址保存到 SPC。

30.4 用户断点调试支持功能

通过使用用户断点调试支持功能，能改变 CPU 接受用户断点请求时的转移目标地址。如果将断点控制寄存器 CBCR 的 UBDE 位置 1，就不转移到 [VBR+ 偏移量] 所示的地址而转移到 DBR 所示的地址。用户断点调试支持功能的流程图如图 30.2 所示。

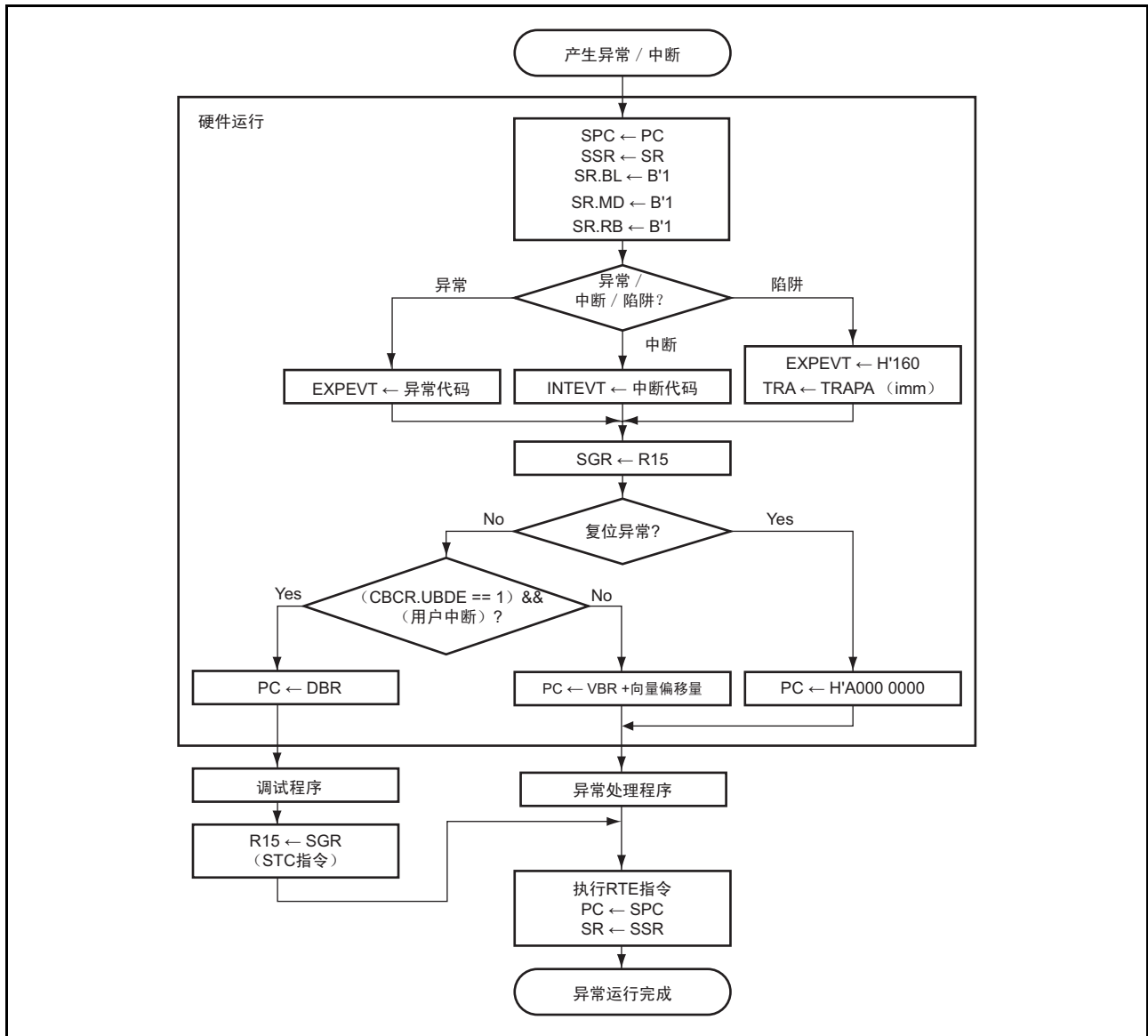


图 30.2 用户断点调试支持功能的流程图

30.5 用户断点的使用例子

(1) 指定为取指令周期的匹配条件

- 例子 1-1

寄存器设定:

```
CBR0=H'00000013/CRR0=H'00002003/CAR0=H'00000404/CAMR0=H'00000000/  
CBR1=H'00000013/CRR1=H'00002001/CAR1=H'000008010/CAMR1=H'00000006/  
CDR1=H'00000000/CDMR1=H'00000000/CETR1=H'00000000/  
CBCR=H'00000000
```

指定条件: 通道0/通道1的独立条件

【通道0】

地址: H'00000404, 地址屏蔽: H'00000000

总线周期: 取指令 (指令执行后)

条件不包含 ASID

【通道1】

地址: H'000008010, 地址屏蔽: H'00000006

数据: H'00000000, 数据屏蔽: H'00000000, 执行次数: H'00000000

总线周期: 取指令 (指令执行前)

条件不包含 ASID、数值和执行次数

在执行地址 H'00000404 的指令后或者在执行地址 H'000008010 ~ H'000008016 的指令前发生用户断点暂停。

- 例 1-2

寄存器设定:

```
CBR0=H'40800013/CRR0=H'00002000/CAR0=H'00037226/CAMR0=H'00000000/  
CBR1=H'C0700013/CRR1=H'00002001/CAR1=H'0003722E/CAMR1=H'00000000/  
CDR1=H'00000000/CDMR1=H'00000000/CETR1=H'00000000/  
CBCR=H'00000000
```

指定条件: 通道0→通道1的连续模式

【通道0】

地址: H'00037226, 地址屏蔽: H'00000000, ASID: H'80

总线周期: 取指令 (指令执行前)

【通道1】

地址: H'0003722E, 地址屏蔽: H'00000000, ASID: H'70

数据: H'00000000, 数据屏蔽: H'00000000, 执行次数: H'00000000

总线周期: 取指令 (指令执行前)

条件不包含数值和执行次数

在 ASID=H'80 并且执行地址 H'00037226 的指令后以及在 ASID=H'70 并且执行地址 H'0003722E 的指令前发生用户断点暂停。

• 例 1-3

寄存器设定:

CBR0=H'00000013/CRR0=H'00002001/CAR0=H'00027128/CAMR0=H'00000000/

CBR1=H'00000013/CRR1=H'00002001/CAR1=H'00031415/CAMR1=H'00000000/

CDR1=H'00000000/CDMR1=H'00000000/CETR1=H'00000000/

CBCR=H'00000000

指定条件: 通道0/通道1的独立条件

【通道0】

地址: H'00027128, 地址屏蔽: H'00000000

总线周期: 取指令 (指令执行前)

条件不包含 ASID

【通道1】

地址: H'00031415, 地址屏蔽: H'00000000

数据: H'00000000, 数据屏蔽: H'00000000, 执行次数: H'00000000

总线周期: 取指令 (指令执行前)

条件不包含 ASID、数值和执行次数

通道0在执行地址H'00027128的指令前发生用户断点暂停。通道1因为对偶数地址取指令, 所以不发生用户断点暂停。

• 例 1-4

寄存器设定:

CBR0=H'40800013/CRR0=H'00002000/CAR0=H'00037226/CAMR0=H'00000000/

CBR1=H'C0700013/CRR1=H'00002001/CAR1=H'0003722E/CAMR1=H'00000000/

CDR1=H'00000000/CDMR1=H'00000000/CETR1=H'00000000/

CBCR=H'00000000

指定条件: 通道0→通道1的连续模式

【通道0】

地址: H'00037226, 地址屏蔽: H'00000000, ASID=H'80

总线周期: 取指令 (指令执行前)

【通道1】

地址: H'0003722E, 地址屏蔽: H'00000000, ASID=H'70

数据: H'00000000, 数据屏蔽: H'00000000, 执行次数: H'00000000

总线周期: 取指令 (指令执行前)

条件不包含数值和执行次数

在ASID=H'80并且执行地址H'00037226的指令后以及在ASID=H'70并且执行地址H'0003722E的指令前发生用户断点暂停。

• 例 1-5

寄存器设定:

CBR0=H'00000013/CRR0=H'00002001/CAR0=H'00000500/CAMR0=H'00000000/

CBR1=H'00000813/CRR1=H'00002001/CAR1=H'00001000/CAMR1=H'00000000/

CDR1=H'00000000/CDMR1=H'00000000/CETR1=H'00000005/

CBCR=H'00000000

指定条件: 通道0/通道1的独立条件

【通道0】

地址: H'00000500, 地址屏蔽: H'00000000

总线周期: 取指令 (指令执行前)

条件不包含 ASID

【通道1】

地址: H'00001000, 地址屏蔽: H'00000000

数据: H'00000000, 数据屏蔽: H'00000000, 执行次数: H'00000005

总线周期: 取指令 (指令执行前)

执行次数: 5次

条件不包含 ASID 和数值

通道0在执行地址H'00000500的指令前发生用户断点暂停。通道1在执行4次地址H'00001000的指令后并且在第5次执行指令前发生用户断点暂停。

• 例 1-6

寄存器设定:

CBR0=H'40800013/CRR0=H'00002003/CAR0=H'00008404/CAMR0=H'00000FFF/

CBR1=H'40700013/CRR1=H'00002001/CAR1=H'00008010/CAMR1=H'00000006/

CDR1=H'00000000/CDMR1=H'00000000/CETR1=H'00000000/

CBCR=H'00000000

指定条件: 通道0/通道1的独立条件

【通道0】

地址: H'00008404, 地址屏蔽: H'00000FFF, ASID: H'80

总线周期: 取指令 (指令执行后)

【通道1】

地址: H'00008010, 地址屏蔽: H'00000006, ASID: H'70

数据: H'00000000, 数据屏蔽: H'00000000, 执行次数: H'00000000

总线周期: 取指令 (指令执行前)

条件不包含数值和执行次数

在ASID=H'80并且执行地址H'00008000~H'00008FFE的指令后或者在ASID=H'70并且执行地址H'00008010~H'00008016的指令前发生用户断点暂停。

(2) 指定为操作数存取周期的匹配条件

• 例2-1

寄存器设定:

CBR0=H'40800023/CRR0=H'00002001/CAR0=H'00123456/CAMR0=H'00000000/

CBR1=H'4070A025/CRR1=H'00002001/CAR1=H'000ABCDE/CAMR1=H'000000FF/

CDR1=H'0000A512/CDMR1=H'00000000/CETR1=H'00000000/

CBCR=H'00000000

指定条件: 通道0/通道1的独立条件

【通道0】

地址: H'00123456, 地址屏蔽: H'00000000, ASID: H'80

总线周期: 操作数总线/操作数存取/读取 (条件不包含操作数长度)

【通道1】

地址: H'000ABCDE, 地址屏蔽: H'000000FF, ASID: H'70

数据: H'0000A512, 数据屏蔽: H'00000000, 执行次数: H'00000000

总线周期: 操作数总线/操作数存取/写入/字

条件不包含执行次数

通道0在ASID=H'80的状态下以长字为单位读地址H'00123454、以字为单位读地址H'00123456或者以字节为单位读地址H'00123456时发生用户断点暂停。通道1在ASID=H'70并且给H'000ABC00~H'000ABCFE写字数据H'A512时发生用户断点暂停。

30.6 使用时的注意事项

1. 在执行改写 UBC 寄存器值的指令后到实际反映其值的期间，可能不发生期望的断点。在更改 UBC 寄存器后并且在执行断点对象指令前，必须执行以下 a~c 中的任意一项：
 - a. 在读取被更改的寄存器后，必须通过 RTE 指令进行转移（读寄存器和 RTE 指令不需要连续）。
 - b. 在读取被更改的寄存器后，必须对任意的地址（也可以是不能缓存的区域）执行 ICBI 指令（读寄存器和 ICBI 指令不需要连续）。
 - c. 在更改 UBC 寄存器前，必须先设定 IRMC.R1=0（初始值），然后按写寄存器→读寄存器→写寄存器（相同的值）的顺序进行寄存器的更新。另外，在更改多个 UBC 寄存器时，不需要每一个寄存器而只需要最后更改的寄存器并且按照上述的步骤进行。
2. CRR0 和 CRR1 的 PCB 位只在将取指令指定为匹配条件时有效。
3. 如果设定连续条件，就在第 1 通道的匹配条件成立后第 2 通道的匹配条件成立时连续条件成立。因而，即使将匹配条件设定为通道 0 和通道 1 的条件匹配在相同的总线周期同时成立，连续条件也不成立，并且不暂停。
4. 对于 SLEEP 指令，不能在执行以取指令周期为匹配条件的指令后发生断点暂停。另外，不能在 SLEEP 指令的 1~5 条指令前发生以操作数存取为匹配条件的断点暂停。
5. 在用户断点和其它异常发生在同一指令时，通过规定的优先顺序进行判断。有关优先顺序，请参照“第 5 章 异常处理”。如果发生更高优先级的异常，就不发生用户断点暂停。
 - 比其他任何异常优先接受指令执行前的断点。
 - 在指令执行后的断点或者数据存取的断点与优先级更高的再执行型异常（包括指令执行前的断点）同时发生时，接受再执行型的异常，并且不将表示条件成立的标志置位。在通过异常处理解除再执行型的异常源并且结束同一指令的重新执行时发生断暂停，并且将对应的标志置位。
 - 在指令执行后的断点或者操作数存取的断点与优先级更高的完成型异常（TRAPA）同时发生时，不产生断点暂停，但是将表示条件成立的标志置位。
6. 在通道 0 和通道 1 独立的匹配条件同时成立或者双方的结果 SPC 值相同时，只发生一次用户断点暂停，但是 2 个通道的条件成立标志都被置位。例如，
地址 110 的指令（在通道 0，取指令执行后的断点）→SPC=112，CCMFR.MF0=1
地址 112 的指令（在通道 1，取指令执行前的断点）→SPC=112，CCMFR.MF1=1
7. 不能对 RTE 指令的延迟槽指令设定指令执行前断点和操作数断点。另外，在操作数断点包含数值时，不能在 RTE 指令的 1~6 条指令前发生断点暂停。
8. 对于 2 个以上执行状态的指令，如果再执行型异常和执行后断点发生竞争，就产生再执行型异常。在这种情况下，当断点条件成立时，CCMFR.MF0（或者 CCMFR.MF1）位有可能被置 1 也有可能不被置 1。

第 31 章 用户调试接口 (H-UDI)

用户调试接口 (H-UDI) 是符合 JTAG (IEEE1149.1) 的串行输入 / 输出接口, 用于连接仿真器。

31.1 特点

H-UDI 是符合 IEEE1149.1 “IEEE Standard Test Access Port and Boundary-Scan Architecture” 的串行输入 / 输出接口, 用于连接仿真器。在使用仿真器时, 不能使用此接口的功能。连接仿真器的方法请参照各仿真器的手册。

H-UDI 有 6 个引脚 (TCK、TMS、TDI、TDO、 $\overline{\text{TRST}}$ 和 $\overline{\text{ASEBRK/BRKACK}}$), $\overline{\text{ASEBRK/BRKACK}}$ 除外的引脚功能和串行传送协议符合 JTAG 规格。用于仿真器的引脚有 6 个信号 (AUDSYNC、ADUCK 和 AUDATA3 ~ AUDATA0), 芯片模式指定引脚有 1 个信号 (MPMD)。

本 LSI 的 H-UDI 分为用于边界扫描的 TAP 控制器和控制 H-UDI 功能的 TAP 控制器。在接通电源或者 $\overline{\text{TRST}}$ 有效时, 选择边界扫描 TAP 控制器, 所以在要使用 H-UDI 功能时需要输入切换命令。另外, CPU 不能存取边界扫描 TAP 控制器。

H-UDI 的框图如图 31.1 所示。

H-UDI 电路内部有 TAP 控制器以及 SDBPR、SDBSR、SDIR 和 SDINT 共 4 个寄存器。这 4 个寄存器分别是支持 JTAG 旁路模式的寄存器、支持 JTAG 边界扫描模式的寄存器、用于命令的寄存器以及用于 H-UDI 中断的寄存器。能从 TDI 和 TDO 引脚直接存取 SDIR。

TAP (Test Access Port) 控制器、控制寄存器和边界扫描 TAP 控制器与芯片的复位引脚相互独立, 通过将 $\overline{\text{TRST}}$ 引脚设定为低电平或者在将 TMS 引脚置 1 后给 TCK 至少输入 5 个以上的时钟周期进行复位。其它电路通过正常的复位进行初始化。

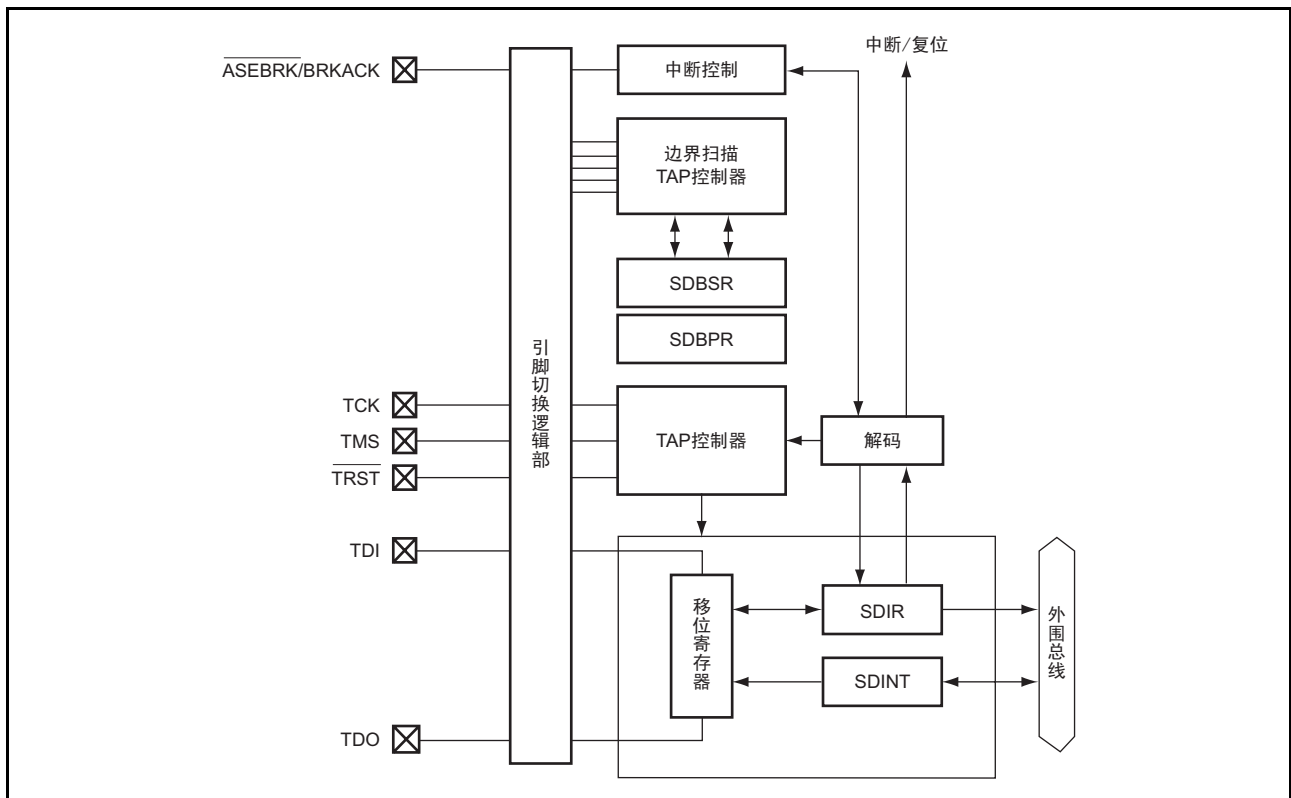


图 31.1 H-UDI 的框图

31.2 输入 / 输出引脚

H-UDI 的引脚结构如表 31.1 所示。

表 31.1 H-UDI 的引脚结构

引脚名	功能	输入 / 输出	说明	未使用时
TCK	时钟引脚	输入	和 JTAG 的串行时钟输入引脚相同。与此信号同步，将数据从数据输入引脚 TDI 输入到 H-UDI 电路，或者从数据输出引脚 TDO 读数据。	开路 *1
TMS	模式引脚	输入	是模式选择输入引脚。通过与 TCK 同步改变此信号，决定 TDI 输入的数据含义。此协议符合 JTAG (IEEE Std 1149.1) 规格。	开路 *1
$\overline{\text{TRST}}$ *2	复位引脚	输入	是对 H-UDI 进行复位的输入引脚。与 TCK 异步接受此信号，为低电平时对 JTAG 接口电路进行复位。无论是否使用 JTAG，必须在接通电源时将 $\overline{\text{TRST}}$ 设定为一定时期的低电平。这与 IEEE 规格不同。	固定接地或者连接 $\overline{\text{PRESET}}$ *3
TDI	数据输入引脚	输入	是数据输入引脚。通过与 TCK 同步改变此信号，将数据传送到 H-UDI 电路。	开路 *1
TDO	数据输出引脚	输出	是数据输出引脚。通过与 TCK 同步读此信号，从 H-UDI 电路读数据。	开路
ASEBRK /BRKACK	用于仿真器的引脚	输入 / 输出	是仿真器专用的引脚	开路 *1
AUDSYNC AUDCK AUDATA3 ~ AUDATA0	用于仿真器的引脚	输出	是仿真器专用的引脚	开路
MPMD	芯片模式指定引脚	输入	指定本 LSI 的运行模式是仿真支持模式 (MPMD= 低电平) 还是芯片模式 (MPMD= 高电平)。	开路 *1

【注】 *1 在芯片内被上拉。在设计能使用仿真器的电路板的情况下或者在通过 H-UDI 使用中断和复位的情况下，也能外接上拉电阻。

*2 在设计能使用仿真器的电路板的情况下或者在通过 H-UDI 使用中断和复位的情况下，必须在接通电源时将 $\overline{\text{PRESET}}$ 重复的期间将 $\overline{\text{TRST}}$ 设定为低电平，并且使 $\overline{\text{TRST}}$ 能独立控制。

*3 固定接地或者连接与 $\overline{\text{PRESET}}$ 的相同 (或相同的动作) 信号。但是，在固定接地时存在以下的问题。因为 $\overline{\text{TRST}}$ 在片内被上拉，所以在从外部固定为接地时会流过微小的电流，电源值取决于端口引脚上拉电阻的规定。此电流不影响芯片的工作，但是会消耗无用的功率。

在设定 TCK 或者本 LSI 的 CPG 时，必须使 TCK 的频率低于本 LSI 的外围时钟频率。

31.3 边界扫描 TAP 控制器 (IDCODE、EXTEST、SAMPLE/PRELOAD、BYPASS、CLAMP、HIGHZ)

本 LSI 的 H-UDI 分为用于边界扫描的 TAP 控制器和控制 H-UDI 复位和 H-UDI 中断功能的 TAP 控制器。在接通电源或者 $\overline{\text{TRST}}$ 有效时，用于边界扫描的 TAP 控制器有效，能使用 JTAG 规定的边界扫描功能。另外，通过输入 H-UDI 切换命令，能使用 H-UDI 复位和 H-UDI 中断功能。但是，本 LSI 有以下限制事项：

- 时钟相关信号 ($\overline{\text{EXTAL}}$ 、XTAL、XOUT 和 XIN) 不属于边界扫描对象范围。
- 复位相关信号 ($\overline{\text{PRESET}}$) 不属于边界扫描对象范围。
- H-UDI 相关信号 (TCK、TDI、TDO、TMS、 $\overline{\text{TRST}}$ 和 MPMD) 不属于边界扫描对象范围。
- USB 相关信号 (DM、DP、VBUS 和 REFRIN) 不属于边界扫描对象范围。
- 在执行边界扫描 (IDCODE、EXTEST、SAMPLE/PRELOAD、BYPASS、CLAMP、HIGHZ、H-UDI 切换命令) 时，TCK 的最大频率为 2MHz。
- H-UDI (外部控制器) 对边界扫描 TAP 控制器的存取长度为 4 位。

边界扫描 TAP 控制器的支持命令如下所示。

【注】 在边界扫描时，必须将 $\overline{\text{PRESET}}$ 引脚固定为高电平。另外，从边界扫描 TAP 控制器切换到 H-UDI 的顺序如图 31.2 所示。

表 31.2 边界扫描 TAP 控制器的支持命令

bit3	bit2	bit1	bit0	说明
1	1	1	1	BYPASS
0	0	0	0	EXTEST
0	0	0	1	SAMPLE/PRELOAD
0	1	0	0	IDCODE
0	1	1	0	CLAMP
0	1	1	1	HIGHZ
0	0	1	1	H-UDI 切换命令
上述以外				禁止设定

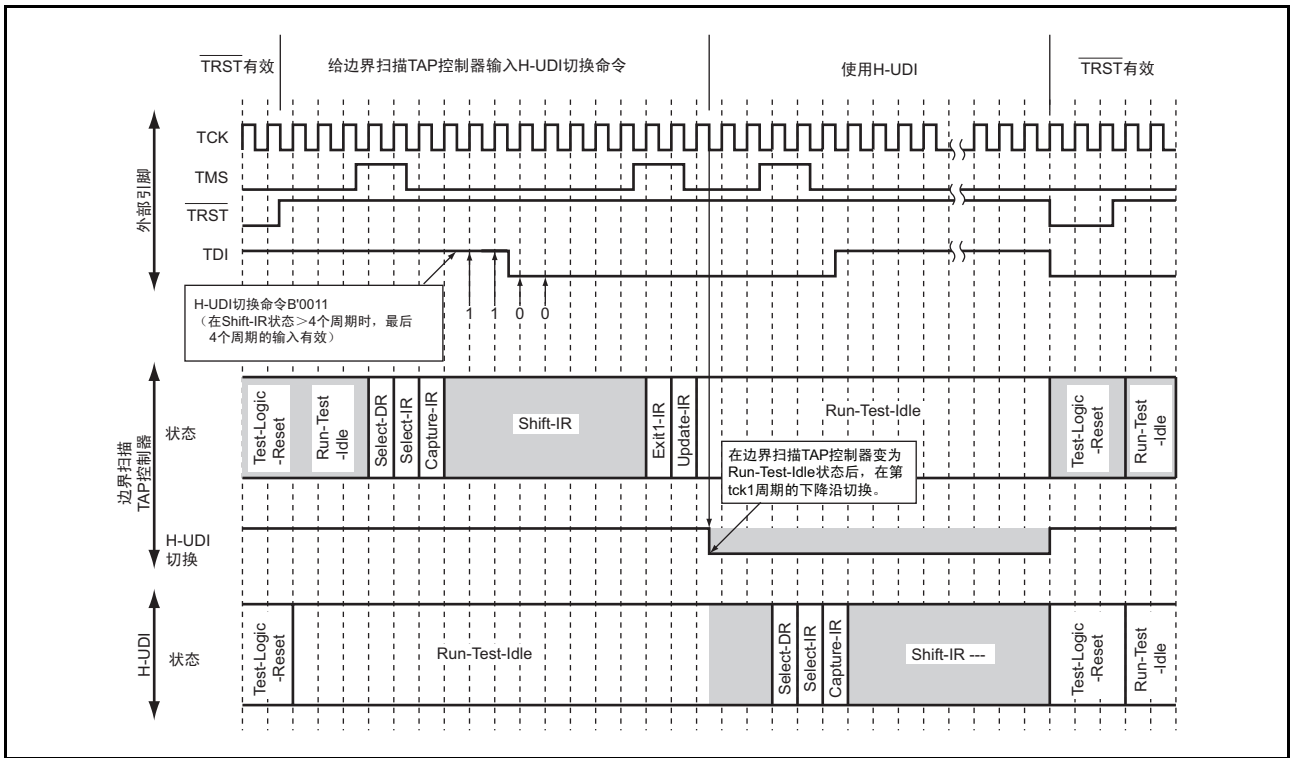


图 31.2 从边界扫描 TAP 控制器切换到 H-UDI 的顺序

31.4 寄存器说明

H-UDI 的寄存器结构如表 31.3 和表 31.4 所示，各处理模式的寄存器状态如表 31.5 所示。

表 31.3 寄存器结构 (1)

名称	略称	CPU				
		R/W	P4 区地址 *1	区域 7 地址 *1	存取长度	初始值 *2
指令寄存器	SDIR	R	H'FC11 0000	H'1C11 0000	16	H'0EFF
中断源寄存器	SDINT	R/W	H'FC11 0018	H'1C11 0018	16	H'0000
边界扫描寄存器	SDBSR	—	—	—	—	—
旁路寄存器	SDBPR	—	—	—	—	不定值

【注】 *1 P4 区地址是使用虚拟地址空间 P4 区的地址，区域 7 地址是使用 TLB 存取物理地址空间区域 7 的地址。

*2 在 $\overline{\text{TRST}}$ 为低电平或者 TAP 为 Test-Logic-Reset 状态时，寄存器被初始化。

表 31.4 寄存器结构 (2)

名称	略称	H-UDI 引脚		
		R/W	存取长度	初始值 *1
指令寄存器	SDIR	R/W	32	H'FFFF FFFD (固定值 *2)
中断源寄存器	SDINT	W*3	32	H'0000 0000
边界扫描寄存器	SDBSR	—	—	—
旁路寄存器	SDBPR	R/W	1	不定值

【注】 *1 在 $\overline{\text{TRST}}$ 引脚为低电平或者 TAP 为 Test-Logic-Reset 状态时，寄存器被初始化。

*2 从 H-UDI 读取的值为固定值 (H'FFFF FFFD)。

*3 能通过 H-UDI 中断命令将最低有效位置 1。

表 31.5 各处理状态的寄存器状态

总称	略称	上电复位	睡眠	待机
指令寄存器	SDIR	H'0EFF	保持	保持
中断源寄存器	SDINT	H'0000	保持	保持

31.4.1 指令寄存器 (SDIR)

SDIR 是 16 位只读寄存器 (CPU 能读取)，由串行输入 (TDI) 设定值 (命令)。通过 $\overline{\text{TRST}}$ 或者在 TAP 的 Test Logic Reset 状态时，对 SDIR 进行初始化。与 CPU 的模式无关，能从 H-UDI 写数据。另外，在设定保留命令时，不保证运行。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TI								—	—	—	—	—	—	—	—
初始值:	0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	1
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	TI	00001110	R	测试指令位 7 ~ 0 0110xxxx: H-UDI 复位无效 0111xxxx: H-UDI 复位有效 101xxxxx: H-UDI 中断 00001110: 初始状态 上述以外: 禁止设定
7 ~ 0	—	全 1	R	保留位 读取值为 1。

【注】 即使进行 H-UDI 复位，CPG 和 WDT 模块也不被初始化。

31.4.2 中断源寄存器 (SDINT)

SDINT 是 16 位可读写寄存器 (CPU 能读写)。在从 H-UDI 引脚给 SDIR 设定 “H-UDI 中断” (Update-IR) 命令时, INTREQ 位为 1。在 SDIR 为 “H-UDI 中断” 命令期间, SDINT 连接在 H-UDI 引脚的 TDI 和 TDO 之间, 能作为 32 位寄存器进行读取。此时, 高 16 位为 0、低 16 位为 SDINT。

只能通过 CPU 给 INTREQ 位写 0, 因为在此位为 1 的期间继续发行中断请求, 所以必须在中断处理程序中将此位清 0。通过 $\overline{\text{TRST}}$ 或者在 TAP 的 Test Logic Reset 状态时, 对此寄存器进行初始化。

位:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INTREQ
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 有关此位的读写, 请参照 “产品使用上的注意事项”。
0	INTREQ	0	R/W	中断请求位 表示是否通过 “H-UDI 中断” 命令产生中断请求。通过 CPU 给此位写 0, 能清除中断请求。在给此位写 1 时, 保持先前的值。

31.4.3 旁路寄存器 (SDBPR)

旁路寄存器 (SDBPR) 是用于支持 JTAG 的旁路模式的 1 位寄存器。如果给边界扫描 TAP 控制器设定 BYPASS 命令, 就在 TDI 和 TDO 之间连接 SDBPR。CPU 不能进行存取。即使在上电复位或者 $\overline{\text{TRST}}$ 有效时, 此寄存器也不被初始化。但是, 在 Capture-DR 状态时, 被初始化为 0。

31.4.4 边界扫描寄存器 (SDBSR)

边界扫描寄存器 (SDBSR) 是用于支持 JTAG 的边界扫描模式的寄存器。为了控制外部输入 / 输出引脚, 此寄存器是配置到 PAD 的移位寄存器, 能用 SAMPLE/PRELOAD、EXTEST 命令进行符合 JTAG (IEEE Std 1149.1) 规格的边界扫描测试。无论芯片为何种模式, CPU 都不能进行存取。即使在上电复位或者 $\overline{\text{TRST}}$ 有效时, 此寄存器也不被初始化。

表 31.6 边界扫描寄存器的结构

号码	引脚名	类型
From TDI		
603	SCK0/AUDSYNC/FCLE	CONTROL
602	SCK0/AUDSYNC/FCLE	OUTPUT
601	SCK0/AUDSYNC/FCLE	INPUT
600	LCD_VEP_WC/DR5/PH0	CONTROL
599	LCD_VEP_WC/DR5/PH0	OUTPUT
598	LCD_VEP_WC/DR5/PH0	INPUT
597	LCD_FLM/VSYNC/SPS/EX_VSYNC/BT_VSYNC	CONTROL
596	LCD_FLM/VSYNC/SPS/EX_VSYNC/BT_VSYNC	OUTPUT
595	LCD_FLM/VSYNC/SPS/EX_VSYNC/BT_VSYNC	INPUT
594	LCD_CL1/HSYNC/SPL/EX_HSYNC/BT_HSYNC	CONTROL
593	LCD_CL1/HSYNC/SPL/EX_HSYNC/BT_HSYNC	OUTPUT
592	LCD_CL1/HSYNC/SPL/EX_HSYNC/BT_HSYNC	INPUT
591	LCD_M_DISP/DE_C/DE_H/BT_DE_C	CONTROL
590	LCD_M_DISP/DE_C/DE_H/BT_DE_C	OUTPUT
589	LCD_M_DISP/DE_C/DE_H/BT_DE_C	INPUT
588	LCD_VCP_WC/DR4/PH1	CONTROL
587	LCD_VCP_WC/DR4/PH1	OUTPUT
586	LCD_VCP_WC/DR4/PH1	INPUT
585	LCD_CLK/DCLKIN	CONTROL
584	LCD_CLK/DCLKIN	OUTPUT
583	LCD_CLK/DCLKIN	INPUT
582	LCD_DATA15/DR3/PG7	CONTROL
581	LCD_DATA15/DR3/PG7	OUTPUT
580	LCD_DATA15/DR3/PG7	INPUT
579	LCD_DATA14/DR2/PG6	CONTROL
578	LCD_DATA14/DR2/PG6	OUTPUT
577	LCD_DATA14/DR2/PG6	INPUT
576	LCD_DATA13/DR1/PG5	CONTROL
575	LCD_DATA13/DR1/PG5	OUTPUT
574	LCD_DATA13/DR1/PG5	INPUT
573	LCD_DATA12/DR0/PG4	CONTROL
572	LCD_DATA12/DR0/PG4	OUTPUT
571	LCD_DATA12/DR0/PG4	INPUT
570	LCD_DATA11/DG5/PG3	CONTROL
569	LCD_DATA11/DG5/PG3	OUTPUT
568	LCD_DATA11/DG5/PG3	INPUT
567	LCD_DATA10/DG4/PG2	CONTROL
566	LCD_DATA10/DG4/PG2	OUTPUT

号码	引脚名	类型
565	LCD_DATA10/DG4/PG2	INPUT
564	LCD_DATA9/DG3/PG1	CONTROL
563	LCD_DATA9/DG3/PG1	OUTPUT
562	LCD_DATA9/DG3/PG1	INPUT
561	LCD_DATA8/DG2/PG0	CONTROL
560	LCD_DATA8/DG2/PG0	OUTPUT
559	LCD_DATA8/DG2/PG0	INPUT
558	LCD_DATA7/DG1/BT_DATA7/PI4	CONTROL
557	LCD_DATA7/DG1/BT_DATA7/PI4	OUTPUT
556	LCD_DATA7/DG1/BT_DATA7/PI4	INPUT
555	LCD_DATA6/DG0/BT_DATA6/PI3	CONTROL
554	LCD_DATA6/DG0/BT_DATA6/PI3	OUTPUT
553	LCD_DATA6/DG0/BT_DATA6/PI3	INPUT
552	LCD_DATA5/DB5/BT_DATA5/PI2	CONTROL
551	LCD_DATA5/DB5/BT_DATA5/PI2	OUTPUT
550	LCD_DATA5/DB5/BT_DATA5/PI2	INPUT
549	LCD_DATA4/DB4/BT_DATA4/PI1	CONTROL
548	LCD_DATA4/DB4/BT_DATA4/PI1	OUTPUT
547	LCD_DATA4/DB4/BT_DATA4/PI1	INPUT
546	LCD_DATA3/DB3/BT_DATA3	CONTROL
545	LCD_DATA3/DB3/BT_DATA3	OUTPUT
544	LCD_DATA3/DB3/BT_DATA3	INPUT
543	LCD_DATA2/DB2/BT_DATA2	CONTROL
542	LCD_DATA2/DB2/BT_DATA2	OUTPUT
541	LCD_DATA2/DB2/BT_DATA2	INPUT
540	LCD_DATA1/DB1/BT_DATA1	CONTROL
539	LCD_DATA1/DB1/BT_DATA1	OUTPUT
538	LCD_DATA1/DB1/BT_DATA1	INPUT
537	LCD_DATA0/DB0/BT_DATA0	CONTROL
536	LCD_DATA0/DB0/BT_DATA0	OUTPUT
535	LCD_DATA0/DB0/BT_DATA0	INPUT
534	LCD_CL2/DE_V/PH3	CONTROL
533	LCD_CL2/DE_V/PH3	OUTPUT
532	LCD_CL2/DE_V/PH3	INPUT
531	LCD_DON/DCLKOUT/PH2	CONTROL
530	LCD_DON/DCLKOUT/PH2	OUTPUT
529	LCD_DON/DCLKOUT/PH2	INPUT
528	PI0/COM/CDE	CONTROL
527	PI0/COM/CDE	OUTPUT
526	PI0/COM/CDE	INPUT

号码	引脚名	类型
525	RDY	INPUT
524	—	INTERNAL
523	NMI	INPUT
522	$\overline{\text{BACK}}$	CONTROL
521	$\overline{\text{BACK}}$	OUTPUT
520	$\overline{\text{BACK}}$	INPUT
519	$\overline{\text{RD}}$	CONTROL
518	$\overline{\text{RD}}$	OUTPUT
517	$\overline{\text{RD}}$	INPUT
516	$\overline{\text{CS3}}$	CONTROL
515	$\overline{\text{CS3}}$	OUTPUT
514	$\overline{\text{CS3}}$	INPUT
513	$\overline{\text{BREQ}}$	INPUT
512	$\overline{\text{BS}}$	CONTROL
511	$\overline{\text{BS}}$	OUTPUT
510	CS0	CONTROL
509	CS0	OUTPUT
508	CS0	INPUT
507	$\overline{\text{ASEBRK/BRKACK/TCLK/PC1}}$	CONTROL
506	$\overline{\text{ASEBRK/BRKACK/TCLK/PC1}}$	OUTPUT
505	$\overline{\text{ASEBRK/BRKACK/TCLK/PC1}}$	INPUT
504	$\overline{\text{A25/PB7/DREQ0/RTS0}}$	CONTROL
503	$\overline{\text{A25/PB7/DREQ0/RTS0}}$	OUTPUT
502	$\overline{\text{A25/PB7/DREQ0/RTS0}}$	INPUT
501	$\overline{\text{A24/PB6/DACK0/CTS0}}$	CONTROL
500	$\overline{\text{A24/PB6/DACK0/CTS0}}$	OUTPUT
499	$\overline{\text{A24/PB6/DACK0/CTS0}}$	INPUT
498	A17	CONTROL
497	A17	OUTPUT
496	A17	INPUT
495	$\overline{\text{A23/PB5/DTEND0/RTS1}}$	CONTROL
494	$\overline{\text{A23/PB5/DTEND0/RTS1}}$	OUTPUT
493	$\overline{\text{A23/PB5/DTEND0/RTS1}}$	INPUT
492	A21/PB3	CONTROL
491	A21/PB3	OUTPUT
490	A21/PB3	INPUT
489	A20/PB2	CONTROL
488	A20/PB2	OUTPUT
487	A20/PB2	INPUT
486	$\overline{\text{A22/PB4/CTS1}}$	CONTROL

号码	引脚名	类型
485	A22/PB4/CTS1	OUTPUT
484	A22/PB4/CTS1	INPUT
483	A19/PB1	CONTROL
482	A19/PB1	OUTPUT
481	A19/PB1	INPUT
480	A18/PB0	CONTROL
479	A18/PB0	OUTPUT
478	A18/PB0	INPUT
477	D15	CONTROL
476	D15	OUTPUT
475	D15	INPUT
474	D14	CONTROL
473	D14	OUTPUT
472	D14	INPUT
471	D1	CONTROL
470	D1	OUTPUT
469	D1	INPUT
468	D0	CONTROL
467	D0	OUTPUT
466	D0	INPUT
465	D13	CONTROL
464	D13	OUTPUT
463	D13	INPUT
462	D12	CONTROL
461	D12	OUTPUT
460	D12	INPUT
459	D3	CONTROL
458	D3	OUTPUT
457	D3	INPUT
456	D2	CONTROL
455	D2	OUTPUT
454	D2	INPUT
453	D11	CONTROL
452	D11	OUTPUT
451	D11	INPUT
450	D10	CONTROL
449	D10	OUTPUT
448	D10	INPUT
447	D5	CONTROL
446	D5	OUTPUT

号码	引脚名	类型
445	D5	INPUT
444	D4	CONTROL
443	D4	OUTPUT
442	D4	INPUT
441	D9	CONTROL
440	D9	OUTPUT
439	D9	INPUT
438	D6	CONTROL
437	D6	OUTPUT
436	D6	INPUT
435	D7	CONTROL
434	D7	OUTPUT
433	D7	INPUT
432	D8	CONTROL
431	D8	OUTPUT
430	D8	INPUT
429	DQMLL	CONTROL
428	DQMLL	OUTPUT
427	DQMLL	INPUT
426	DQMUL	CONTROL
425	DQMUL	OUTPUT
424	DQMUL	INPUT
423	DQMUU	CONTROL
422	DQMUU	OUTPUT
421	DQMUU	INPUT
420	D16	CONTROL
419	D16	OUTPUT
418	D16	INPUT
417	DQMLU	CONTROL
416	DQMLU	OUTPUT
415	DQMLU	INPUT
414	D17	CONTROL
413	D17	OUTPUT
412	D17	INPUT
411	D18	CONTROL
410	D18	OUTPUT
409	D18	INPUT
408	D19	CONTROL
407	D19	OUTPUT
406	D19	INPUT

号码	引脚名	类型
405	D31	CONTROL
404	D31	OUTPUT
403	D31	INPUT
402	D30	CONTROL
401	D30	OUTPUT
400	D30	INPUT
399	D20	CONTROL
398	D20	OUTPUT
397	D20	INPUT
396	D21	CONTROL
395	D21	OUTPUT
394	D21	INPUT
393	D22	CONTROL
392	D22	OUTPUT
391	D22	INPUT
390	D28	CONTROL
389	D28	OUTPUT
388	D28	INPUT
387	D29	CONTROL
386	D29	OUTPUT
385	D29	INPUT
384	A15	CONTROL
383	A15	OUTPUT
382	A15	INPUT
381	D23	CONTROL
380	D23	OUTPUT
379	D23	INPUT
378	D26	CONTROL
377	D26	OUTPUT
376	D26	INPUT
375	A13	CONTROL
374	A13	OUTPUT
373	A16	CONTROL
372	A16	OUTPUT
371	A16	INPUT
370	D27	CONTROL
369	D27	OUTPUT
368	D27	INPUT
367	D24	CONTROL
366	D24	OUTPUT

号码	引脚名	类型
365	D24	INPUT
364	A10	CONTROL
363	A10	OUTPUT
362	A14	CONTROL
361	A14	OUTPUT
360	D25	CONTROL
359	D25	OUTPUT
358	D25	INPUT
357	A4	CONTROL
356	A4	OUTPUT
355	A11	CONTROL
354	A11	OUTPUT
353	A5	CONTROL
352	A5	OUTPUT
351	R/W	CONTROL
350	R/W	OUTPUT
349	A8	CONTROL
348	A8	OUTPUT
347	A12	CONTROL
346	A12	OUTPUT
345	CKE	CONTROL
344	CKE	OUTPUT
343	RAS	CONTROL
342	RAS	OUTPUT
341	CLKOUT	CONTROL
340	CLKOUT	OUTPUT
339	A9	CONTROL
338	A9	OUTPUT
337	A6	CONTROL
336	A6	OUTPUT
335	A7	CONTROL
334	A7	OUTPUT
333	CAS	CONTROL
332	CAS	OUTPUT
331	CS1	CONTROL
330	CS1	OUTPUT
329	CS1	INPUT
328	CS2	CONTROL
327	CS2	OUTPUT
326	CS2	INPUT

号码	引脚名	类型
325	A0	CONTROL
324	A0	OUTPUT
323	D47/ $\overline{\text{IDECS0}}$	CONTROL
322	D47/ $\overline{\text{IDECS0}}$	OUTPUT
321	D47/ $\overline{\text{IDECS0}}$	INPUT
320	A3	CONTROL
319	A3	OUTPUT
318	A1	CONTROL
317	A1	OUTPUT
316	D45/ $\overline{\text{IODACK}}$	CONTROL
315	D45/ $\overline{\text{IODACK}}$	OUTPUT
314	D45/ $\overline{\text{IODACK}}$	INPUT
313	D46/ $\overline{\text{IDECS1}}$	CONTROL
312	D46/ $\overline{\text{IDECS1}}$	OUTPUT
311	D46/ $\overline{\text{IDECS1}}$	INPUT
310	D33/PF6	CONTROL
309	D33/PF6	OUTPUT
308	D33/PF6	INPUT
307	A2	CONTROL
306	A2	OUTPUT
305	D44/ $\overline{\text{IDEINT}}$	CONTROL
304	D44/ $\overline{\text{IDEINT}}$	OUTPUT
303	D44/ $\overline{\text{IDEINT}}$	INPUT
302	D43/ $\overline{\text{IDEIORDY}}$	CONTROL
301	D43/ $\overline{\text{IDEIORDY}}$	OUTPUT
300	D43/ $\overline{\text{IDEIORDY}}$	INPUT
299	D42/ $\overline{\text{IDEIORD}}$	CONTROL
298	D42/ $\overline{\text{IDEIORD}}$	OUTPUT
297	D42/ $\overline{\text{IDEIORD}}$	INPUT
296	D32/PF7	CONTROL
295	D32/PF7	OUTPUT
294	D32/PF7	INPUT
293	D40/ $\overline{\text{IDEIOWR}}$	CONTROL
292	D40/ $\overline{\text{IDEIOWR}}$	OUTPUT
291	D40/ $\overline{\text{IDEIOWR}}$	INPUT
290	D41/ $\overline{\text{IODREQ}}$	CONTROL
289	D41/ $\overline{\text{IODREQ}}$	OUTPUT
288	D41/ $\overline{\text{IODREQ}}$	INPUT
287	D35/ $\overline{\text{IDEA0}}$	CONTROL
286	D35/ $\overline{\text{IDEA0}}$	OUTPUT

号码	引脚名	类型
285	D35/IDEA0	INPUT
284	D37/IDEA1	CONTROL
283	D37/IDEA1	OUTPUT
282	D37/IDEA1	INPUT
281	D39/IDED14	CONTROL
280	D39/IDED14	OUTPUT
279	D39/IDED14	INPUT
278	D34/PF5	CONTROL
277	D34/PF5	OUTPUT
276	D34/PF5	INPUT
275	D36/IDEA2	CONTROL
274	D36/IDEA2	OUTPUT
273	D36/IDEA2	INPUT
272	D63/IDED1	CONTROL
271	D63/IDED1	OUTPUT
270	D63/IDED1	INPUT
269	D38/IDED15	CONTROL
268	D38/IDED15	OUTPUT
267	D38/IDED15	INPUT
266	D62/IDED0	CONTROL
265	D62/IDED0	OUTPUT
264	D62/IDED0	INPUT
263	$\overline{\text{WE2}}/\text{DQM64UL}$	CONTROL
262	$\overline{\text{WE2}}/\text{DQM64UL}$	OUTPUT
261	$\overline{\text{WE0}}/\text{DQM64LL}$	CONTROL
260	$\overline{\text{WE0}}/\text{DQM64LL}$	OUTPUT
259	D60/IDED2	CONTROL
258	D60/IDED2	OUTPUT
257	D60/IDED2	INPUT
256	$\overline{\text{WE3}}/\text{DQM64UU}$	CONTROL
255	$\overline{\text{WE3}}/\text{DQM64UU}$	OUTPUT
254	$\overline{\text{WE1}}/\text{DQM64LU}$	CONTROL
253	$\overline{\text{WE1}}/\text{DQM64LU}$	OUTPUT
252	D61/IDED3	CONTROL
251	D61/IDED3	OUTPUT
250	D61/IDED3	INPUT
249	D48/IDED13	CONTROL
248	D48/IDED13	OUTPUT
247	D48/IDED13	INPUT
246	D59/IDED5	CONTROL

号码	引脚名	类型
245	D59/IDED5	OUTPUT
244	D59/IDED5	INPUT
243	D58/IDED4	CONTROL
242	D58/IDED4	OUTPUT
241	D58/IDED4	INPUT
240	D49/IDED12	CONTROL
239	D49/IDED12	OUTPUT
238	D49/IDED12	INPUT
237	D51/IDED10	CONTROL
236	D51/IDED10	OUTPUT
235	D51/IDED10	INPUT
234	D50/IDED11	CONTROL
233	D50/IDED11	OUTPUT
232	D50/IDED11	INPUT
231	D56/IDED6	CONTROL
230	D56/IDED6	OUTPUT
229	D56/IDED6	INPUT
228	D52/IDED9	CONTROL
227	D52/IDED9	OUTPUT
226	D52/IDED9	INPUT
225	D53/IDED8	CONTROL
224	D53/IDED8	OUTPUT
223	D53/IDED8	INPUT
222	D57/IDED7	CONTROL
221	D57/IDED7	OUTPUT
220	D57/IDED7	INPUT
219	D54/ $\overline{\text{IDERST}}$	CONTROL
218	D54/ $\overline{\text{IDERST}}$	OUTPUT
217	D54/ $\overline{\text{IDERST}}$	INPUT
216	D55/DIRECTION	CONTROL
215	D55/DIRECTION	OUTPUT
214	D55/DIRECTION	INPUT
213	WOL/PF2/IDEA0_M	CONTROL
212	WOL/PF2/IDEA0_M	OUTPUT
211	WOL/PF2/IDEA0_M	INPUT
210	SSISCK2/PC3	CONTROL
209	SSISCK2/PC3	OUTPUT
208	SSISCK2/PC3	INPUT
207	SSIDATA2/PC2	CONTROL
206	SSIDATA2/PC2	OUTPUT

号码	引脚名	类型
205	SSIDATA2/PC2	INPUT
204	SSIWS2/PC4	CONTROL
203	SSIWS2/PC4	OUTPUT
202	SSIWS2/PC4	INPUT
201	LNKSTA/PF3/ $\overline{\text{IDECS0_M}}$	CONTROL
200	LNKSTA/PF3/ $\overline{\text{IDECS0_M}}$	OUTPUT
199	LNKSTA/PF3/ $\overline{\text{IDECS0_M}}$	INPUT
198	EXOUT/PF4/ $\overline{\text{IDECS1_M}}$	CONTROL
197	EXOUT/PF4/ $\overline{\text{IDECS1_M}}$	OUTPUT
196	EXOUT/PF4/ $\overline{\text{IDECS1_M}}$	INPUT
195	AUDIO_CLK2/PC5	CONTROL
194	AUDIO_CLK2/PC5	OUTPUT
193	AUDIO_CLK2/PC5	INPUT
192	CRS/PD7/IDEA1_M	CONTROL
191	CRS/PD7/IDEA1_M	OUTPUT
190	CRS/PD7/IDEA1_M	INPUT
189	COL/PE7/IDEA2_M	CONTROL
188	COL/PE7/IDEA2_M	OUTPUT
187	COL/PE7/IDEA2_M	INPUT
186	TX_ER/PD6/ $\overline{\text{IDEIOWR_M}}$	CONTROL
185	TX_ER/PD6/ $\overline{\text{IDEIOWR_M}}$	OUTPUT
184	TX_ER/PD6/ $\overline{\text{IDEIOWR_M}}$	INPUT
183	MII_TXD3/SSIDATA5/ $\overline{\text{IODACK_M}}$ /PD0	CONTROL
182	MII_TXD3/SSIDATA5/ $\overline{\text{IODACK_M}}$ /PD0	OUTPUT
181	MII_TXD3/SSIDATA5/ $\overline{\text{IODACK_M}}$ /PD0	INPUT
180	MII_TXD2/AUDIO_CLK5/IDEINT_M/PD1	CONTROL
179	MII_TXD2/AUDIO_CLK5/IDEINT_M/PD1	OUTPUT
178	MII_TXD2/AUDIO_CLK5/IDEINT_M/PD1	INPUT
177	RX_ER/PE6/IODREQ_M	CONTROL
176	RX_ER/PE6/IODREQ_M	OUTPUT
175	RX_ER/PE6/IODREQ_M	INPUT
174	MII_TXD1/SSIWS5/ $\overline{\text{IDEIORD_M}}$ /PD2	CONTROL
173	MII_TXD1/SSIWS5/ $\overline{\text{IDEIORD_M}}$ /PD2	OUTPUT
172	MII_TXD1/SSIWS5/ $\overline{\text{IDEIORD_M}}$ /PD2	INPUT
171	SSIDATA3/PH4	CONTROL
170	SSIDATA3/PH4	OUTPUT
169	SSIDATA3/PH4	INPUT
168	MII_TXD0/SSISCK5/IDEIORDY_M/PD3	CONTROL
167	MII_TXD0/SSISCK5/IDEIORDY_M/PD3	OUTPUT
166	MII_TXD0/SSISCK5/IDEIORDY_M/PD3	INPUT

号码	引脚名	类型
165	TX_EN/PD4/IDED0_M	CONTROL
164	TX_EN/PD4/IDED0_M	OUTPUT
163	TX_EN/PD4/IDED0_M	INPUT
162	SSIWS3/PH6	CONTROL
161	SSIWS3/PH6	OUTPUT
160	SSIWS3/PH6	INPUT
159	TX_CLK/PD5/IDED15_M	CONTROL
158	TX_CLK/PD5/IDED15_M	OUTPUT
157	TX_CLK/PD5/IDED15_M	INPUT
156	RX_CLK/PE5/IDED1_M	CONTROL
155	RX_CLK/PE5/IDED1_M	OUTPUT
154	RX_CLK/PE5/IDED1_M	INPUT
153	RX_DV/PE4/IDED14_M	CONTROL
152	RX_DV/PE4/IDED14_M	OUTPUT
151	RX_DV/PE4/IDED14_M	INPUT
150	SSISCK3/PH5	CONTROL
149	SSISCK3/PH5	OUTPUT
148	SSISCK3/PH5	INPUT
147	IRQ0/DTEND1	CONTROL
146	IRQ0/DTEND1	OUTPUT
145	IRQ0/DTEND1	INPUT
144	MII_RXD0/SSIWS4/IDED2_M/PE3	CONTROL
143	MII_RXD0/SSIWS4/IDED2_M/PE3	OUTPUT
142	MII_RXD0/SSIWS4/IDED2_M/PE3	INPUT
141	MII_RXD1/SSISCK4/IDED13_M/PE2	CONTROL
140	MII_RXD1/SSISCK4/IDED13_M/PE2	OUTPUT
139	MII_RXD1/SSISCK4/IDED13_M/PE2	INPUT
138	AUDIO_CLK3/PH7	CONTROL
137	AUDIO_CLK3/PH7	OUTPUT
136	AUDIO_CLK3/PH7	INPUT
135	MII_RXD2/SSIDATA4/IDED3_M/PE1	CONTROL
134	MII_RXD2/SSIDATA4/IDED3_M/PE1	OUTPUT
133	MII_RXD2/SSIDATA4/IDED3_M/PE1	INPUT
132	IRQOUT/DREQ1	CONTROL
131	IRQOUT/DREQ1	OUTPUT
130	IRQOUT/DREQ1	INPUT
129	MII_RXD3/AUDIO_CLK4/IDED12_M/PE0	CONTROL
128	MII_RXD3/AUDIO_CLK4/IDED12_M/PE0	OUTPUT
127	MII_RXD3/AUDIO_CLK4/IDED12_M/PE0	INPUT
126	MDC/PF0/IDED4_M	CONTROL

号码	引脚名	类型
125	MDC/PF0/IDED4_M	OUTPUT
124	MDC/PF0/IDED4_M	INPUT
123	MDIO/PF1/IDED11_M	CONTROL
122	MDIO/PF1/IDED11_M	OUTPUT
121	MDIO/PF1/IDED11_M	INPUT
120	AUDIO_CLK0/PC7	CONTROL
119	AUDIO_CLK0/PC7	OUTPUT
118	AUDIO_CLK0/PC7	INPUT
117	SSIWS0	CONTROL
116	SSIWS0	OUTPUT
115	SSIWS0	INPUT
114	STATUS1/RTS2/PA7	CONTROL
113	STATUS1/RTS2/PA7	OUTPUT
112	STATUS1/RTS2/PA7	INPUT
111	SSISCK0	CONTROL
110	SSISCK0	OUTPUT
109	SSISCK0	INPUT
108	AUDIO_CLK1/PC6	CONTROL
107	AUDIO_CLK1/PC6	OUTPUT
106	AUDIO_CLK1/PC6	INPUT
105	STATUS0/CTS2/PA6	CONTROL
104	STATUS0/CTS2/PA6	OUTPUT
103	STATUS0/CTS2/PA6	INPUT
102	SSIDATA0	CONTROL
101	SSIDATA0	OUTPUT
100	SSIDATA0	INPUT
99	SSISCK1	CONTROL
98	SSISCK1	OUTPUT
97	SSISCK1	INPUT
96	PJ7/IDED10_M	CONTROL
95	PJ7/IDED10_M	OUTPUT
94	PJ7/IDED10_M	INPUT
93	SSIWS1	CONTROL
92	SSIWS1	OUTPUT
91	SSIWS1	INPUT
90	PJ6/IDED5_M	CONTROL
89	PJ6/IDED5_M	OUTPUT
88	PJ6/IDED5_M	INPUT
87	FRE/PA4	CONTROL
86	FRE/PA4	OUTPUT

号码	引脚名	类型
85	FRE/PA4	INPUT
84	SSIDATA1	CONTROL
83	SSIDATA1	OUTPUT
82	SSIDATA1	INPUT
81	PJ5/IDED9_M	CONTROL
80	PJ5/IDED9_M	OUTPUT
79	PJ5/IDED9_M	INPUT
78	PJ4/IDED6_M	CONTROL
77	PJ4/IDED6_M	OUTPUT
76	PJ4/IDED6_M	INPUT
75	PJ2/IDED8_M	CONTROL
74	PJ2/IDED8_M	OUTPUT
73	PJ2/IDED8_M	INPUT
72	PJ3/IDED7_M	CONTROL
71	PJ3/IDED7_M	OUTPUT
70	PJ3/IDED7_M	INPUT
69	FWE/PA3	CONTROL
68	FWE/PA3	OUTPUT
67	FWE/PA3	INPUT
66	FCE/PA5	CONTROL
65	FCE/PA5	OUTPUT
64	FCE/PA5	INPUT
63	PJ1/IDERST_M	CONTROL
62	PJ1/IDERST_M	OUTPUT
61	PJ1/IDERST_M	INPUT
60	PJ0/DIRECTION_M	CONTROL
59	PJ0/DIRECTION_M	OUTPUT
58	PJ0/DIRECTION_M	INPUT
57	MODE7/FD6	CONTROL
56	MODE7/FD6	OUTPUT
55	MODE7/FD6	INPUT
54	FALE/PC0	CONTROL
53	FALE/PC0	OUTPUT
52	FALE/PC0	INPUT
51	MODE3/FD3	CONTROL
50	MODE3/FD3	OUTPUT
49	MODE3/FD3	INPUT
48	MODE5/FD5	CONTROL
47	MODE5/FD5	OUTPUT
46	MODE5/FD5	INPUT

号码	引脚名	类型
45	TXD2/PA2	CONTROL
44	TXD2/PA2	OUTPUT
43	TXD2/PA2	INPUT
42	MODE2/FD2	CONTROL
41	MODE2/FD2	OUTPUT
40	MODE2/FD2	INPUT
39	MODE4/FD4	CONTROL
38	MODE4/FD4	OUTPUT
37	MODE4/FD4	INPUT
36	MODE8/FD7	CONTROL
35	MODE8/FD7	OUTPUT
34	MODE8/FD7	INPUT
33	MODE1/FD1	CONTROL
32	MODE1/FD1	OUTPUT
31	MODE1/FD1	INPUT
30	RXD2/PA1	CONTROL
29	RXD2/PA1	OUTPUT
28	RXD2/PA1	INPUT
27	SCK2/PA0	CONTROL
26	SCK2/PA0	OUTPUT
25	SCK2/PA0	INPUT
24	SCL	OUTPUT
23	SCL	INPUT
22	SDA	OUTPUT
21	SDA	INPUT
20	RXD1/AUDATA2	CONTROL
19	RXD1/AUDATA2	OUTPUT
18	RXD1/AUDATA2	INPUT
17	WDTOVF/IRQ1/AUDCK/DACK1	CONTROL
16	WDTOVF/IRQ1/AUDCK/DACK1	OUTPUT
15	WDTOVF/IRQ1/AUDCK/DACK1	INPUT
14	MODE0/FD0	CONTROL
13	MODE0/FD0	OUTPUT
12	MODE0/FD0	INPUT
11	RXD0/AUDATA0	CONTROL
10	RXD0/AUDATA0	OUTPUT
9	RXD0/AUDATA0	INPUT
8	TXD1/AUDATA3	CONTROL
7	TXD1/AUDATA3	OUTPUT
6	TXD1/AUDATA3	INPUT

号码	引脚名	类型
5	TXD0/AUDATA1	CONTROL
4	TXD0/AUDATA1	OUTPUT
3	TXD0/AUDATA1	INPUT
2	SCK1/FR \bar{B}	CONTROL
1	SCK1/FR \bar{B}	OUTPUT
0	SCK1/FR \bar{B}	INPUT
To TDO		

31.5 运行说明

31.5.1 TAP 控制

TAP 控制电路的内部状态如图 31.3 所示，符合 JTAG 规格的状态转移。

- 转移条件是 TCK 上升沿的 TMS 值。
- 在 TCK 的上升沿对 TDI 值进行采样，在 TCK 的下降沿对 TDI 值进行移位。
- TDO 值在 TCK 的下降沿发生变化。另外，不在 Shift-DR 和 Shift-IR 状态时，TDO 为高阻抗状态。
- 通过 TRST=0，与 TCK 异步转移到 Test-Logic-Reset 状态。

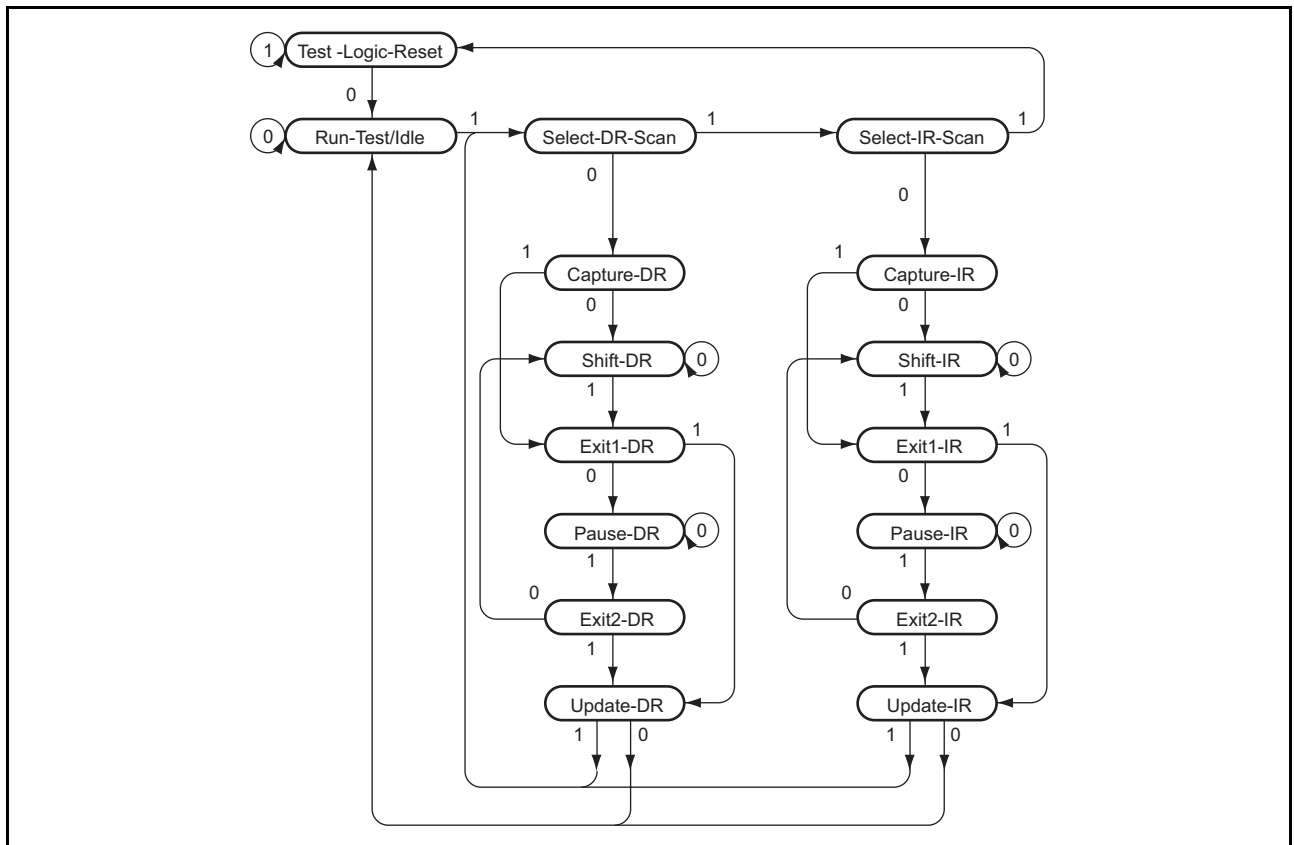


图 31.3 TAP 控制状态转移图

31.5.2 H-UDI 复位

能通过 SDIR 命令进行上电复位。通过在从 H-UDI 引脚发送 H-UDI 复位有效命令后发送 H-UDI 复位无效命令进行复位（参照图 31.4）。H-UDI 复位有效命令和 H-UDI 复位无效命令之间的所需时间与上电复位时的复位引脚保持低电平的时间相同。

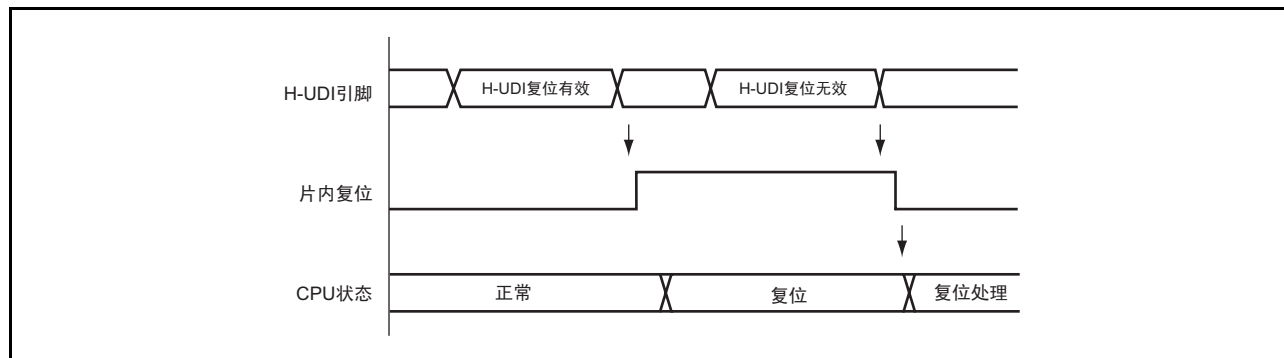


图 31.4 H-UDI 复位

31.5.3 H-UDI 中断

H-UDI 中断功能是通过 H-UDI 给 SDIR 设定命令值而产生的中断功能。

如果在设定命令 (Update-IR) 后 SDINT 寄存器的 INTREQ 位变为 1, H-UDI 中断请求信号就输出有效电平。只要不通过软件将 INTREQ 位置 0, 中断请求信号就不会变为无效, 所以中断请求不会丢失。在 SDIR 为 H-UDI 中断命令的期间, SDINT 连接在 TDI 和 TDO 之间。

31.6 注意事项

除非通过 $\overline{\text{TRST}}$ 有效或者在 TAP 的 Test-Logic-Reset 状态时进行初始化, 只要 H-UDI 不写其它命令, 被设定的 SDIR 命令就不会改变。

H-UDI 用于连接仿真器。因此, 如果使用仿真器, 就不能使用 H-UDI 的功能。

第 32 章 寄存器一览

32.1 寄存器地址一览

- 以功能模块为单位，按照手册的章节号的顺序记载本 LSI 的内部 I/O 寄存器。
- 禁止存取一览表中未记载的保留地址，否则不能保证存取时的运行及其后继运行。
- 在 16 位或者 32 位地址时，以大端法为前提，记载 MSB 侧的地址。
- 存取长度表示位数。
- 有关各寄存器的详细内容，请参照相关章节的寄存器说明。

表 32.1 寄存器结构

模块	名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度	备注
异常处理	TRAPA 异常寄存器	TRA	R/W	H'FF00 0020	H'1F00 0020	32	
	异常事件寄存器	EXPEVT	R/W	H'FF00 0024	H'1F00 0024	32	
	中断事件寄存器	INTEVT	R/W	H'FF00 0028	H'1F00 0028	32	
	非支持检测异常寄存器	EXPMASK	R/W	H'FF00 0004	H'1F00 0004	32	
MMU	页表入口高位寄存器	PTEH	R/W	H'FF00 0000	H'1F00 0000	32	
	页表入口低位寄存器	PTEL	R/W	H'FF00 0004	H'1F00 0004	32	
	转换表基址寄存器	TTB	R/W	H'FF00 0008	H'1F00 0008	32	
	TLB 异常地址寄存器	TEA	R/W	H'FF00 000C	H'1F00 000C	32	
	MMU 控制寄存器	MMUCR	R/W	H'FF00 0010	H'1F00 0010	32	
	页表入口辅助寄存器	PTEA	R/W	H'FF00 0034	H'1F00 0034	32	
	物理地址空间控制寄存器	PASCR	R/W	H'FF00 0070	H'1F00 0070	32	
	重取指令禁止控制寄存器	IRMCR	R/W	H'FF00 0078	H'1F00 0078	32	
高速缓存	高速缓存控制寄存器	CCR	R/W	H'FF00 001C	H'1F00 001C	32	
	队列地址控制寄存器 0	QACR0	R/W	H'FF00 0038	H'1F00 0038	32	
	队列地址控制寄存器 1	QACR1	R/W	H'FF00 003C	H'1F00 003C	32	
	内部存储器控制寄存器	RAMCR	R/W	H'FF00 0074	H'1F00 0074	32	
内部存储器	内部存储器控制寄存器	RAMCR	R/W	H'FF00 0074	H'1F00 0074	32	
CPG	频率控制寄存器	FRQCR	R	H'FFC8 0000	H'1FC8 0000	32	
	PLL 控制寄存器	PLLCR	R/W	H'FFC8 0024	H'1FC8 0024	32	
	VDC2 时钟控制寄存器	VDC2CLKCR	R/W	H'FFC8 0004	H'1FC8 0004	32	
WDT	看门狗定时器停止定时寄存器	WDTST	R/W	H'FFCC 0000	H'1FCC 0000	32	
	看门狗定时器控制 / 状态寄存器	WDTCSR	R/W	H'FFCC 0004	H'1FCC 0004	32	
	看门狗定时器基准停止定时寄存器	WDTBST	R/W	H'FFCC 0008	H'1FCC 0008	32	
	看门狗定时器计数器	WDTCNT	R	H'FFCC 0010	H'1FCC 0010	32	
	看门狗定时器基准计数器	WDTBCNT	R	H'FFCC 0018	H'1FCC 0018	32	
DMAC	DMA 源地址寄存器 0	SAR0	R/W	H'FF60 8020	H'1F60 8020	32*3	
	DMA 目标地址寄存器 0	DAR0	R/W	H'FF60 8024	H'1F60 8024	32*3	
	DMA 传送计数寄存器 0	TCR0	R/W	H'FF60 8028	H'1F60 8028	32*3	
	DMA 通道控制寄存器 0	CHCR0	R/W*1	H'FF60 802C	H'1F60 802C	32*3	
	DMA 源地址寄存器 1	SAR1	R/W	H'FF60 8030	H'1F60 8030	32*3	

模块	名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度	备注
DMAC	DMA 目标地址寄存器 1	DAR1	R/W	H'FF60 8034	H'1F60 8034	32*3	
	DMA 传送计数寄存器 1	TCR1	R/W	H'FF60 8038	H'1F60 8038	32*3	
	DMA 通道控制寄存器 1	CHCR1	R/W*1	H'FF60 803C	H'1F60 803C	32*3	
	DMA 源地址寄存器 2	SAR2	R/W	H'FF60 8040	H'1F60 8040	32*3	
	DMA 目标地址寄存器 2	DAR2	R/W	H'FF60 8044	H'1F60 8044	32*3	
	DMA 传送计数寄存器 2	TCR2	R/W	H'FF60 8048	H'1F60 8048	32*3	
	DMA 通道控制寄存器 2	CHCR2	R/W*1	H'FF60 804C	H'1F60 804C	32*3	
	DMA 源地址寄存器 3	SAR3	R/W	H'FF60 8050	H'1F60 8050	32*3	
	DMA 目标地址寄存器 3	DAR3	R/W	H'FF60 8054	H'1F60 8054	32*3	
	DMA 传送计数寄存器 3	TCR3	R/W	H'FF60 8058	H'1F60 8058	32*3	
	DMA 通道控制寄存器 3	CHCR3	R/W*1	H'FF60 805C	H'1F60 805C	32*3	
	DMA 操作寄存器 0	DMAOR0	R/W*2	H'FF60 8060	H'1F60 8060	32*3	
	DMA 源地址寄存器 4	SAR4	R/W	H'FF60 8070	H'1F60 8070	32*3	
	DMA 目标地址寄存器 4	DAR4	R/W	H'FF60 8074	H'1F60 8074	32*3	
	DMA 传送计数寄存器 4	TCR4	R/W	H'FF60 8078	H'1F60 8078	32*3	
	DMA 通道控制寄存器 4	CHCR4	R/W*1	H'FF60 807C	H'1F60 807C	32*3	
	DMA 源地址寄存器 5	SAR5	R/W	H'FF60 8080	H'1F60 8080	32*3	
	DMA 目标地址寄存器 5	DAR5	R/W	H'FF60 8084	H'1F60 8084	32*3	
	DMA 传送计数寄存器 5	TCR5	R/W	H'FF60 8088	H'1F60 8088	32*3	
	DMA 通道控制寄存器 5	CHCR5	R/W*1	H'FF60 808C	H'1F60 808C	32*3	
	DMA 源地址寄存器 B0	SARB0	R/W	H'FF60 8120	H'1F60 8120	32*3	
	DMA 目标地址寄存器 B0	DARB0	R/W	H'FF60 8124	H'1F60 8124	32*3	
	DMA 传送计数寄存器 B0	TCRB0	R/W	H'FF60 8128	H'1F60 8128	32*3	
	DMA 源地址寄存器 B1	SARB1	R/W	H'FF60 8130	H'1F60 8130	32*3	
	DMA 目标地址寄存器 B1	DARB1	R/W	H'FF60 8134	H'1F60 8134	32*3	
	DMA 传送计数寄存器 B1	TCRB1	R/W	H'FF60 8138	H'1F60 8138	32*3	
	DMA 源地址寄存器 B2	SARB2	R/W	H'FF60 8140	H'1F60 8140	32*3	
	DMA 目标地址寄存器 B2	DARB2	R/W	H'FF60 8144	H'1F60 8144	32*3	
	DMA 传送计数寄存器 B2	TCRB2	R/W	H'FF60 8148	H'1F60 8148	32*3	
	DMA 源地址寄存器 B3	SARB3	R/W	H'FF60 8150	H'1F60 8150	32*3	
	DMA 目标地址寄存器 B3	DARB3	R/W	H'FF60 8154	H'1F60 8154	32*3	
DMA 传送计数寄存器 B3	TCRB3	R/W	H'FF60 8158	H'1F60 8158	32*3		
DMA 扩展资源选择器 0	DMARS0	R/W	H'FF60 9000	H'1F60 9000	16*3		
DMA 扩展资源选择器 1	DMARS1	R/W	H'FF60 9004	H'1F60 9004	16*3		
DMA 扩展资源选择器 2	DMARS2	R/W	H'FF60 9008	H'1F60 9008	16*3		
INTC	中断控制寄存器 0	ICR0	R/W	H'FFD0 0000	H'1FD0 0000	32	
	中断控制寄存器 1	ICR1	R/W	H'FFD0 001C	H'1FD0 001C	32	
	中断优先顺序设定寄存器	INTPRI	R/W	H'FFD0 0010	H'1FD0 0010	32	
	中断源寄存器	INTREQ	R/W	H'FFD0 0024	H'1FD0 0024	32	
	中断屏蔽寄存器	INTMSK	R/W	H'FFD0 0044	H'1FD0 0044	32	

模块	名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度	备注
INTC	中断屏蔽清除寄存器	INTMSKCLR	R/W	H'FFD0 0064	H'1FD0 0064	32	
	NMI 标志控制寄存器	NMIFCR	R/W	H'FFD0 00C0	H'1FD0 00C0	32	
	用户中断屏蔽级设定寄存器	USERIMASK	R/W	H'FFD3 0000	H'1FD3 0000	32	
	中断优先顺序设定寄存器 0	INT2PRI0	R/W	H'FFD4 0000	H'1FD4 0000	32	
	中断优先顺序设定寄存器 1	INT2PRI1	R/W	H'FFD4 0004	H'1FD4 0004	32	
	中断优先顺序设定寄存器 2	INT2PRI2	R/W	H'FFD4 0008	H'1FD4 0008	32	
	中断优先顺序设定寄存器 3	INT2PRI3	R/W	H'FFD4 000C	H'1FD4 000C	32	
	中断优先顺序设定寄存器 4	INT2PRI4	R/W	H'FFD4 0010	H'1FD4 0010	32	
	中断优先顺序设定寄存器 5	INT2PRI5	R/W	H'FFD4 0014	H'1FD4 0014	32	
	中断优先顺序设定寄存器 6	INT2PRI6	R/W	H'FFD4 0018	H'1FD4 0018	32	
	中断优先顺序设定寄存器 7	INT2PRI7	R/W	H'FFD4 001C	H'1FD4 001C	32	
	中断优先顺序设定寄存器 8	INT2PRI8	R/W	H'FFD4 00A0	H'1FD4 00A0	32	
	中断优先顺序设定寄存器 9	INT2PRI9	R/W	H'FFD4 00A4	H'1FD4 00A4	32	
	中断优先顺序设定寄存器 10	INT2PRI10	R/W	H'FFD4 00A8	H'1FD4 00A8	32	
	中断优先顺序设定寄存器 11	INT2PRI11	R/W	H'FFD4 00AC	H'1FD4 00AC	32	
	中断优先顺序设定寄存器 12	INT2PRI12	R/W	H'FFD4 00B0	H'1FD4 00B0	32	
	中断源寄存器 0 (不影响屏蔽状态)	INT2A0	R	H'FFD4 0030	H'1FD4 0030	32	
	中断源寄存器 01 (不影响屏蔽状态)	INT2A01	R	H'FFD4 00C0	H'1FD4 00C0	32	
	中断源寄存器 1 (影响屏蔽状态)	INT2A1	R	H'FFD4 0034	H'1FD4 0034	32	
	中断源寄存器 11 (影响屏蔽状态)	INT2A11	R	H'FFD4 00C4	H'1FD4 00C4	32	
	中断屏蔽寄存器	INT2MSKR	R/W	H'FFD4 0038	H'1FD4 0038	32	
	中断屏蔽寄存器 1	INT2MSKR1	R/W	H'FFD4 00D0	H'1FD4 00D0	32	
	中断屏蔽清除寄存器	INT2MSKCR	W	H'FFD4 003C	H'1FD4 003C	32	
	中断屏蔽清除寄存器 1	INT2MSKCR1	W	H'FFD4 00D4	H'1FD4 00D4	32	
	各模块中断源寄存器 0	INT2B0	R	H'FFD4 0040	H'1FD4 0040	32	
	各模块中断源寄存器 2	INT2B2	R	H'FFD4 0048	H'1FD4 0048	32	
	各模块中断源寄存器 3	INT2B3	R	H'FFD4 004C	H'1FD4 004C	32	
	各模块中断源寄存器 4	INT2B4	R	H'FFD4 0050	H'1FD4 0050	32	
	各模块中断源寄存器 5	INT2B5	R	H'FFD4 0054	H'1FD4 0054	32	
各模块中断源寄存器 6	INT2B6	R	H'FFD4 0058	H'1FD4 0058	32		
各模块中断源寄存器 7	INT2B7	R	H'FFD4 005C	H'1FD4 005C	32		
GPIO 中断设定寄存器	INT2GPIC	R/W	H'FFD4 0090	H'1FD4 0090	32		
TMU	定时器输出控制寄存器	TOCR	R/W	H'FFD8 0000	H'1FD8 0000	8	
	定时器启动寄存器 0	TSTR0	R/W	H'FFD8 0004	H'1FD8 0004	8	
	定时器常数寄存器 0	TCOR0	R/W	H'FFD8 0008	H'1FD8 0008	32	
	定时器计数器 0	TCNT0	R/W	H'FFD8 000C	H'1FD8 000C	32	
	定时器控制寄存器 0	TCR0	R/W	H'FFD8 0010	H'1FD8 0010	16	

模块	名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度	备注
TMU	定时器常数寄存器 1	TCOR1	R/W	H'FFD8 0014	H'1FD8 0014	32	
	定时器计数器 1	TCNT1	R/W	H'FFD8 0018	H'1FD8 0018	32	
	定时器控制寄存器 1	TCR1	R/W	H'FFD8 001C	H'1FD8 001C	16	
	定时器常数寄存器 2	TCOR2	R/W	H'FFD8 0020	H'1FD8 0020	32	
	定时器计数器 2	TCNT2	R/W	H'FFD8 0024	H'1FD8 0024	32	
	定时器控制寄存器 2	TCR2	R/W	H'FFD8 0028	H'1FD8 0028	16	
	输入捕捉寄存器 2	TCPR2	R	H'FFD8 002C	H'1FD8 002C	32	
	定时器起始寄存器 1	TSTR1	R/W	H'FFDC 0004	H'1FDC 0004	8	
	定时器常数寄存器 3	TCOR3	R/W	H'FFDC 0008	H'1FDC 0008	32	
	定时器计数器 3	TCNT3	R/W	H'FFDC 000C	H'1FDC 000C	32	
	定时器控制寄存器 3	TCR3	R/W	H'FFDC 0010	H'1FDC 0010	16	
	定时器常数寄存器 4	TCOR4	R/W	H'FFDC 0014	H'1FDC 0014	32	
	定时器计数器 4	TCNT4	R/W	H'FFDC 0018	H'1FDC 0018	32	
	定时器控制寄存器 4	TCR4	R/W	H'FFDC 001C	H'1FDC 001C	16	
	定时器常数寄存器 5	TCOR5	R/W	H'FFDC 0020	H'1FDC 0020	32	
	定时器计数器 5	TCNT5	R/W	H'FFDC 0024	H'1FDC 0024	32	
	定时器控制寄存器 5	TCR5	R/W	H'FFDC 0028	H'1FDC 0028	16	
SCIF	串行模式寄存器 _0	SCSMR_0	R/W	H'FFE0 0000	H'1FE0 0000	16	
	位速率寄存器 _0	SCBRR_0	R/W	H'FFE0 0004	H'1FE0 0004	8	
	串行控制寄存器 _0	SCSCR_0	R/W	H'FFE0 0008	H'1FE0 0008	16	
	发送 FIFO 数据寄存器 _0	SCFTDR_0	W	H'FFE0 000C	H'1FE0 000C	8	
	串行状态寄存器 _0	SCFSR_0	R/(W)*4	H'FFE0 0010	H'1FE0 0010	16	
	接收 FIFO 数据寄存器 _0	SCFRDR_0	R	H'FFE0 0014	H'1FE0 0014	8	
	FIFO 控制寄存器 _0	SCFCR_0	R/W	H'FFE0 0018	H'1FE0 0018	16	
	FIFO 数据计数置位寄存器 _0	SCFDR_0	R	H'FFE0 001C	H'1FE0 001C	16	
	串行端口寄存器 _0	SCSPTR_0	R/W	H'FFE0 0020	H'1FE0 0020	16	
	线路状态寄存器 _0	SCLSR_0	R/(W)*5	H'FFE0 0024	H'1FE0 0024	16	
	串行扩展模式寄存器 _0	SCEMR_0	R/W	H'FFE0 0028	H'1FE0 0028	16	
	串行模式寄存器 _1	SCSMR_1	R/W	H'FFE1 0000	H'1FE1 0000	16	
	位速率寄存器 _1	SCBRR_1	R/W	H'FFE1 0004	H'1FE1 0004	8	
	串行控制寄存器 _1	SCSCR_1	R/W	H'FFE1 0008	H'1FE1 0008	16	
	发送 FIFO 数据寄存器 _1	SCFTDR_1	W	H'FFE1 000C	H'1FE1 000C	8	
	串行状态寄存器 _1	SCFSR_1	R/(W)*4	H'FFE1 0010	H'1FE1 0010	16	
	接收 FIFO 数据寄存器 _1	SCFRDR_1	R	H'FFE1 0014	H'1FE1 0014	8	
	FIFO 控制寄存器 _1	SCFCR_1	R/W	H'FFE1 0018	H'1FE1 0018	16	
	FIFO 数据计数置位寄存器 _1	SCFDR_1	R	H'FFE1 001C	H'1FE1 001C	16	
	串行端口寄存器 _1	SCSPTR_1	R/W	H'FFE1 0020	H'1FE1 0020	16	
	线路状态寄存器 _1	SCLSR_1	R/(W)*5	H'FFE1 0024	H'1FE1 0024	16	
串行扩展模式寄存器 _1	SCEMR_1	R/W	H'FFE1 0028	H'1FE1 0028	16		
串行模式寄存器 _2	SCSMR_2	R/W	H'FFE2 0000	H'1FE2 0000	16		

模块	名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度	备注
SCIF	位速率寄存器 _2	SCBRR_2	R/W	H'FFE2 0004	H'1FE2 0004	8	
	串行控制寄存器 _2	SCSCR_2	R/W	H'FFE2 0008	H'1FE2 0008	16	
	发送 FIFO 数据寄存器 _2	SCFTDR_2	W	H'FFE2 000C	H'1FE2 000C	8	
	串行状态寄存器 _2	SCFSR_2	R/(W)*4	H'FFE2 0010	H'1FE2 0010	16	
	接收 FIFO 数据寄存器 _2	SCFRDR_2	R	H'FFE2 0014	H'1FE2 0014	8	
	FIFO 控制寄存器 _2	SCFCR_2	R/W	H'FFE2 0018	H'1FE2 0018	16	
	FIFO 数据计数置位寄存器 _2	SCFDR_2	R	H'FFE2 001C	H'1FE2 001C	16	
	串行端口寄存器 _2	SCSPTR_2	R/W	H'FFE2 0020	H'1FE2 0020	16	
	线路状态寄存器 _2	SCLSR_2	R/(W)*5	H'FFE2 0024	H'1FE2 0024	16	
	串行扩展模式寄存器 _2	SCEMR_2	R/W	H'FFE2 0028	H'1FE2 0028	16	
IIC	从属控制寄存器	ICSCR	R/W	H'FFE7 0000	H'1FF7 0000	8	
	主控控制寄存器	ICMCR	R/W	H'FFE7 0004	H'1FF7 0004	8	
	从属状态寄存器	ICSSR	R/(W)*6	H'FFE7 0008	H'1FF7 0008	8	
	主控状态寄存器	ICMSR	R/(W)*7	H'FFE7 000C	H'1FF7 000C	8	
	从属中断允许寄存器	ICSIER	R/W	H'FFE7 0010	H'1FF7 0010	8	
	主控中断允许寄存器	ICMIER	R/W	H'FFE7 0014	H'1FF7 0014	8	
	时钟控制寄存器	ICCCR	R/W	H'FFE7 0018	H'1FF7 0018	8	
	从属地址寄存器	ICSAR	R/W	H'FFE7 001C	H'1FF7 001C	8	
	主控地址寄存器	ICMAR	R/W	H'FFE7 0020	H'1FF7 0020	8	
	接收数据寄存器	ICRXD	R/W	H'FFE7 0024	H'1FF7 0024	8	
	发送数据寄存器	ICTXD	R/W	H'FFE7 0024	H'1FF7 0024	8	
SSI_DMAC0	DMA 模式寄存器 0	SSIDMMR0	R/W	H'FF40 1000	H'1F40 1000	32	
	RDMA 传送源地址寄存器 0	SSIRDMDR0	R/W	H'FF40 1008	H'1F40 1008	32	
	RDMA 传送字节寄存器 0	SSIRDMCNR0	R/W	H'FF40 1010	H'1F40 1010	32	
	WDMA 传送源地址寄存器 0	SSIWDMADR0	R/W	H'FF40 1018	H'1F40 1018	32	
	WDMA 传送字节寄存器 0	SSIWDMCNR0	R/W	H'FF40 1020	H'1F40 1020	32	
	DMA 控制寄存器 0	SSIDMCOR0	R/W	H'FF40 1028	H'1F40 1028	32	
	发送暂停块计数器 0	SSISTPBLCNT0	R/W	H'FF40 1030	H'1F40 1030	32	
	发送暂停中传送数据寄存器 0	SSISTPDR0	R/W	H'FF40 1038	H'1F40 1038	32	
	块计数源寄存器 0	SSIBLCNTR0	R/W	H'FF40 1040	H'1F40 1040	32	
	块计数器 0	SSIBLCNT0	R/W	H'FF40 1048	H'1F40 1048	32	
	块 xn 次中断计数源寄存器 0	SSIBLNCNTR0	R/W	H'FF40 1050	H'1F40 1050	32	
	块 xn 次计数器 0	SSIBLNCNT0	R/W	H'FF40 1058	H'1F40 1058	32	
	DMA 模式寄存器 1	SSIDMMR1	R/W	H'FF40 1060	H'1F40 1060	32	
	RDMA 传送源地址寄存器 1	SSIRDMDR1	R/W	H'FF40 1068	H'1F40 1068	32	
	RDMA 传送字节寄存器 1	SSIRDMCNR1	R/W	H'FF40 1070	H'1F40 1070	32	
	WDMA 传送源地址寄存器 1	SSIWDMADR1	R/W	H'FF40 1078	H'1F40 1078	32	
	WDMA 传送字节寄存器 1	SSIWDMCNR1	R/W	H'FF40 1080	H'1F40 1080	32	
	DMA 控制寄存器 1	SSIDMCOR1	R/W	H'FF40 1088	H'1F40 1088	32	
发送暂停块计数器 1	SSISTPBLCNT1	R/W	H'FF40 1090	H'1F40 1090	32		

模块	名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度	备注
SSI_DMAC0	发送暂停中传送数据寄存器 1	SSISTPDR1	R/W	H'FF40 1098	H'1F40 1098	32	
	块计数源寄存器 1	SSIBLCNTR1	R/W	H'FF40 10A0	H'1F40 10A0	32	
	块计数器 1	SSIBLCNT1	R/W	H'FF40 10A8	H'1F40 10A8	32	
	块 xn 次中断计数源寄存器 1	SSIBLNCNTR1	R/W	H'FF40 10B0	H'1F40 10B0	32	
	块 xn 次计数器 1	SSIBLNCNT1	R/W	H'FF40 10B8	H'1F40 10B8	32	
	DMA 模式寄存器 2	SSIDMMR2	R/W	H'FF40 10C0	H'1F40 10C0	32	
	RDMA 传送源地址寄存器 2	SSIRDMAADR2	R/W	H'FF40 10C8	H'1F40 10C8	32	
	RDMA 传送字节寄存器 2	SSIRDMCNTR2	R/W	H'FF40 10D0	H'1F40 10D0	32	
	WDMA 传送源地址寄存器 2	SSIWDMAADR2	R/W	H'FF40 10D8	H'1F40 10D8	32	
	WDMA 传送字节寄存器 2	SSIWDMCNTR2	R/W	H'FF40 10E0	H'1F40 10E0	32	
	DMA 控制寄存器 2	SSIDMCOR2	R/W	H'FF40 10E8	H'1F40 10E8	32	
	发送暂停块计数器 2	SSISTPBLCNT2	R/W	H'FF40 10F0	H'1F40 10F0	32	
	发送暂停中传送数据寄存器 2	SSISTPDR2	R/W	H'FF40 10F8	H'1F40 10F8	32	
	块计数源寄存器 2	SSIBLCNTR2	R/W	H'FF40 1100	H'1F40 1100	32	
	块计数器 2	SSIBLCNT2	R/W	H'FF40 1108	H'1F40 1108	32	
	块 xn 次中断计数源寄存器 2	SSIBLNCNTR2	R/W	H'FF40 1110	H'1F40 1110	32	
	块 xn 次计数器 2	SSIBLNCNT2	R/W	H'FF40 1118	H'1F40 1118	32	
	DMA 操作寄存器 0	SSIDMAOR0	R/W	H'FF40 1180	H'1F40 1180	32	
	中断状态寄存器 0	SSIDMINTSR0	R/W	H'FF40 1188	H'1F40 1188	32	
中断屏蔽寄存器 0	SSIDMINTMR0	R/W	H'FF40 1190	H'1F40 1190	32		
SSI_DMAC1	DMA 模式寄存器 3	SSIDMMR3	R/W	H'FF50 1000	H'1F50 1000	32	
	RDMA 传送源地址寄存器 3	SSIRDMAADR3	R/W	H'FF50 1008	H'1F50 1008	32	
	RDMA 传送字节寄存器 3	SSIRDMCNTR3	R/W	H'FF50 1010	H'1F50 1010	32	
	WDMA 传送源地址寄存器 3	SSIWDMAADR3	R/W	H'FF50 1018	H'1F50 1018	32	
	WDMA 传送字节寄存器 3	SSIWDMCNTR3	R/W	H'FF50 1020	H'1F50 1020	32	
	DMA 控制寄存器 3	SSIDMCOR3	R/W	H'FF50 1028	H'1F50 1028	32	
	发送暂停块计数器 3	SSISTPBLCNT3	R/W	H'FF50 1030	H'1F50 1030	32	
	发送暂停中传送数据寄存器 3	SSISTPDR3	R/W	H'FF50 1038	H'1F50 1038	32	
	块计数源寄存器 3	SSIBLCNTR3	R/W	H'FF50 1040	H'1F50 1040	32	
	块计数器 3	SSIBLCNT3	R/W	H'FF50 1048	H'1F50 1048	32	
	块 xn 次中断计数源寄存器 3	SSIBLNCNTR3	R/W	H'FF50 1050	H'1F50 1050	32	
	块 xn 次计数器 3	SSIBLNCNT3	R/W	H'FF50 1058	H'1F50 1058	32	
	DMA 模式寄存器 4	SSIDMMR4	R/W	H'FF50 1060	H'1F50 1060	32	
	RDMA 传送源地址寄存器 4	SSIRDMAADR4	R/W	H'FF50 1068	H'1F50 1068	32	
	RDMA 传送字节寄存器 4	SSIRDMCNTR4	R/W	H'FF50 1070	H'1F50 1070	32	
	WDMA 传送源地址寄存器 4	SSIWDMAADR4	R/W	H'FF50 1078	H'1F50 1078	32	
	WDMA 传送字节寄存器 4	SSIWDMCNTR4	R/W	H'FF50 1080	H'1F50 1080	32	
	DMA 控制寄存器 4	SSIDMCOR4	R/W	H'FF50 1088	H'1F50 1088	32	
	发送暂停块计数器 4	SSISTPBLCNT4	R/W	H'FF50 1090	H'1F50 1090	32	
	发送暂停中传送数据寄存器 4	SSISTPDR4	R/W	H'FF50 1098	H'1F50 1098	32	

模块	名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度	备注
SSI_DMAC1	块计数源寄存器 4	SSIBLCNTR4	R/W	H'FF50 10A0	H'1F50 10A0	32	
	块计数器 4	SSIBLCNT4	R/W	H'FF50 10A8	H'1F50 10A8	32	
	块 xn 次中断计数源寄存器 4	SSIBLNCNTR4	R/W	H'FF50 10B0	H'1F50 10B0	32	
	块 xn 次计数器 4	SSIBLNCNT4	R/W	H'FF50 10B8	H'1F50 10B8	32	
	DMA 模式寄存器 5	SSIDMMR5	R/W	H'FF50 10C0	H'1F50 10C0	32	
	RDMA 传送源地址寄存器 5	SSIRDMDR5	R/W	H'FF50 10C8	H'1F50 10C8	32	
	RDMA 传送字节寄存器 5	SSIRDMCNTR5	R/W	H'FF50 10D0	H'1F50 10D0	32	
	WDMA 传送源地址寄存器 5	SSIWDMADR5	R/W	H'FF50 10D8	H'1F50 10D8	32	
	WDMA 传送字节寄存器 5	SSIWDMCNTR5	R/W	H'FF50 10E0	H'1F50 10E0	32	
	DMA 控制寄存器 5	SSIDMCOR5	R/W	H'FF50 10E8	H'1F50 10E8	32	
	发送暂停块计数器 5	SSISTPBLCNT5	R/W	H'FF50 10F0	H'1F50 10F0	32	
	发送暂停中传送数据寄存器 5	SSISTPDR5	R/W	H'FF50 10F8	H'1F50 10F8	32	
	块计数源寄存器 5	SSIBLCNTR5	R/W	H'FF50 1100	H'1F50 1100	32	
	块计数器 5	SSIBLCNT5	R/W	H'FF50 1108	H'1F50 1108	32	
	块 xn 次中断计数源寄存器 5	SSIBLNCNTR5	R/W	H'FF50 1110	H'1F50 1110	32	
	块 xn 次计数器 5	SSIBLNCNT5	R/W	H'FF50 1118	H'1F50 1118	32	
	DMA 操作寄存器 1	SSIDMAOR1	R/W	H'FF50 1180	H'1F50 1180	32	
	中断状态寄存器 1	SSIDMINTSR1	R/W	H'FF50 1188	H'1F50 1188	32	
中断屏蔽寄存器 1	SSIDMINTMR1	R/W	H'FF50 1190	H'1F50 1190	32		
SSI_CH0 ~ 5	控制寄存器 0	SSICR0	R/W	H'FF40 2000	H'1F40 2000	32	
	状态寄存器 0	SSISR0	R/W*8	H'FF40 2004	H'1F40 2004	32	
	发送数据寄存器 0	SSITDR0	R/W	H'FF40 2008	H'1F40 2008	32	
	接收数据寄存器 0	SSIRDR0	R	H'FF40 200C	H'1F40 200C	32	
	控制寄存器 1	SSICR1	R/W	H'FF40 3000	H'1F40 3000	32	
	状态寄存器 1	SSISR1	R/W*8	H'FF40 3004	H'1F40 3004	32	
	发送数据寄存器 1	SSITDR1	R/W	H'FF40 3008	H'1F40 3008	32	
	接收数据寄存器 1	SSIRDR1	R	H'FF40 300C	H'1F40 300C	32	
	控制寄存器 2	SSICR2	R/W	H'FF40 4000	H'1F40 4000	32	
	状态寄存器 2	SSISR2	R/W*8	H'FF40 4004	H'1F40 4004	32	
	发送数据寄存器 2	SSITDR2	R/W	H'FF40 4008	H'1F40 4008	32	
	接收数据寄存器 2	SSIRDR2	R	H'FF40 400C	H'1F40 400C	32	
	控制寄存器 3	SSICR3	R/W	H'FF50 2000	H'1F50 2000	32	
	状态寄存器 3	SSISR3	R/W*8	H'FF50 2004	H'1F50 2004	32	
	发送数据寄存器 3	SSITDR3	R/W	H'FF50 2008	H'1F50 2008	32	
	接收数据寄存器 3	SSIRDR3	R	H'FF50 200C	H'1F50 200C	32	
	控制寄存器 4	SSICR4	R/W	H'FF50 3000	H'1F50 3000	32	
	状态寄存器 4	SSISR4	R/W*8	H'FF50 3004	H'1F50 3004	32	
	发送数据寄存器 4	SSITDR4	R/W	H'FF50 3008	H'1F50 3008	32	
	接收数据寄存器 4	SSIRDR4	R	H'FF50 300C	H'1F50 300C	32	
控制寄存器 5	SSICR5	R/W	H'FF50 4000	H'1F50 4000	32		

模块	名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度	备注
SSL_CH0 ~ 5	状态寄存器 5	SSISR5	R/W*8	H'FF50 4004	H'1F50 4004	32	
	发送数据寄存器 5	SSITDR5	R/W	H'FF50 4008	H'1F50 4008	32	
	接收数据寄存器 5	SSIRDR5	R	H'FF50 400C	H'1F50 400C	32	
EtherC	EtherC 模式寄存器	ECMR	R/W	H'FEF0 0100*	H'1EF0 0100*	32	
	EtherC 状态寄存器	ECSR	R/W	H'FEF0 0110*	H'1EF0 0110*	32	
	EtherC 中断允许寄存器	ECSIPR	R/W	H'FEF0 0118*	H'1EF0 0118*	32	
	接收帧长度上限寄存器	RFLR	R/W	H'FEF0 0108*	H'1EF0 0108*	32	
	PHY 部接口寄存器	PIR	R/W	H'FEF0 0120*	H'1EF0 0120*	32	
	MAC 地址高位设定寄存器	MAHR	R/W	H'FEF0 01C0*	H'1EF0 01C0*	32	
	MAC 地址低位设定寄存器	MALR	R/W	H'FEF0 01C8*	H'1EF0 01C8*	32	
	PHY 部状态寄存器	PSR	R	H'FEF0 0128*	H'1EF0 0128*	32	
	试发送失败计数寄存器	TROCR	R/W	H'FEF0 01D0*	H'1EF0 01D0*	32	
	延迟冲突检测计数寄存器	CDCR	R/W	H'FEF0 01D4*	H'1EF0 01D4*	32	
	载波消失计数寄存器	LCCR	R/W	H'FEF0 01D8*	H'1EF0 01D8*	32	
	载波未检测计数寄存器	CNDCR	R/W	H'FEF0 01DC*	H'1EF0 01DC*	32	
	CRC 错误帧接收计数寄存器	CEFCR	R/W	H'FEF0 01E4*	H'1EF0 01E4*	32	
	帧接收错误计数寄存器	FRECR	R/W	H'FEF0 01E8*	H'1EF0 01E8*	32	
	64 字节未帧接收计数寄存器	TSFRRCR	R/W	H'FEF0 01EC*	H'1EF0 01EC*	32	
	指定字节超帧接收计数寄存器	TLFRRCR	R/W	H'FEF0 01F0*	H'1EF0 01F0*	32	
	剩余位帧接收计数寄存器	RFCR	R/W	H'FEF0 01F4*	H'1EF0 01F4*	32	
	多播地址帧接收计数寄存器	MAFCR	R/W	H'FEF0 01F8*	H'1EF0 01F8*	32	
	IPG 设定寄存器	IPGR	R/W	H'FEF0 0150*	H'1EF0 0150*	32	
	自动 PAUSE 帧设定寄存器	APR	R/W	H'FEF0 0154*	H'1EF0 0154*	32	
	手动 PAUSE 帧设定寄存器	MPR	R/W	H'FEF0 0158*	H'1EF0 0158*	32	
	自动 PAUSE 帧再发送次数设定寄存器	TPAUSER	R/W	H'FEF0 0164*	H'1EF0 0164*	32	
	随机数生成计数器上限值设定寄存器	RDMLR	R/W	H'FEF0 0140*	H'1EF0 0140*	32	
	接收 PAUSE 帧计数器	RFCF	R/W	H'FEF0 0160*	H'1EF0 0160*	32	
	PAUSE 帧再发送次数计数器	TPAUSECR	R/W	H'FEF0 0168*	H'1EF0 0168*	32	
	Broadcast 帧接收次数设定寄存器	BCFRR	R/W	H'FEF0 016C*	H'1EF0 016C*	32	
E-DMAC	E-DMAC 模式寄存器	EDMR	R/W	H'FEF0 0000*	H'1EF0 0000*	32	
	E-DMAC 发送请求寄存器	EDTRR	R/W	H'FEF0 0008*	H'1EF0 0008*	32	
	E-DMAC 接收请求寄存器	EDRRR	R/W	H'FEF0 0010*	H'1EF0 0010*	32	
	发送描述符列表起始地址寄存器	TDLAR	R/W	H'FEF0 0018*	H'1EF0 0018*	32	
	接收描述符列表起始地址寄存器	RDLAR	R/W	H'FEF0 0020*	H'1EF0 0020*	32	
	EtherC/E-DMAC 状态寄存器	EESR	R/W	H'FEF0 0028*	H'1EF0 0028*	32	
	EtherC/E-DMAC 状态中断允许寄存器	EESIPR	R/W	H'FEF0 0030*	H'1EF0 0030*	32	
	发送 / 接收状态复制指示寄存器	TRSCER	R/W	H'FEF0 0038*	H'1EF0 0038*	32	

模块	名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度	备注
E-DMAC	丢失帧计数寄存器	RMFCR	R	H'FEF0 0040*	H'1EF0 0040*	32	
	发送 FIFO 阈值指定寄存器	TFTR	R/W	H'FEF0 0048*	H'1EF0 0048*	32	
	FIFO 容量指定寄存器	FDR	R/W	H'FEF0 0050*	H'1EF0 0050*	32	
	接收方式控制寄存器	RMCR	R/W	H'FEF0 0058*	H'1EF0 0058*	32	
	发送 FIFO 下溢计数寄存器	TFUCR	R/W	H'FEF0 0064*	H'1EF0 0064*	32	
	接收 FIFO 上溢计数寄存器	RFOCR	R/W	H'FEF0 0068	H'1EF0 0068	32	
	接收缓冲器写地址寄存器	RBWAR	R	H'FEF0 00C8*	H'1EF0 00C8*	32	
	接收描述符取地址寄存器	RDFAR	R	H'FEF0 00CC*	H'1EF0 00CC*	32	
	发送缓冲器读地址寄存器	TBRAR	R	H'FEF0 00D4*	H'1EF0 00D4*	32	
	发送描述符取地址寄存器	TDFAR	R	H'FEF0 00D8*	H'1EF0 00D8*	32	
	流控制开始 FIFO 阈值设定寄存器	FCFTR	R/W	H'FEF0 0070*	H'1EF0 0070*	32	
	接收数据填充设定寄存器	RPADIR	R/W	H'FEF0 0078*	H'1EF0 0078*	32	
	发送中断设定寄存器	TRIMD	R/W	H'FEF0 007C*	H'1EF0 007C*	32	
	各输出信号设定寄存器	IOSR	R/W	H'FEF0 006C*	H'1EF0 006C*	32	
USB	系统配置控制寄存器	SYSCFG	R/W	H'FE40 0000		16	
	CPU 总线等待设定寄存器	BUSWAIT	R/W	H'FE40 0002		16	
	系统配置状态寄存器	SYSSTS	R	H'FE40 0004		16	
	设备状态控制寄存器	DVSTCTR	R/W	H'FE40 0008		16	
	测试模式寄存器	TESTMODE	R/W	H'FE40 000C		16	
	DMA0-FIFO 总线配置寄存器	D0FBCFG	R/W	H'FE40 0010		16	
	DMA1-FIFO 总线配置寄存器	D1FBCFG	R/W	H'FE40 0012		16	
	CFIFO 端口寄存器	CFIFO	R/W	H'FE40 0014		8、16、32	
	D0FIFO 端口寄存器		R/W	H'FE40 0018		8、16、32	
				H'FE40 0180			
	D1FIFO 端口寄存器		R/W	H'FE40 001C		8、16、32	
				H'FE40 01C0			
	CFIFO 端口选择寄存器	CFIFOSEL	R/W	H'FE40 0020		16	
	CFIFO 端口控制寄存器	CFIFOCTR	R/W	H'FE40 0022		16	
	D0FIFO 端口选择寄存器	D0FIFOSEL	R/W	H'FE40 0028		16	
	D0FIFO 端口控制寄存器	D0FIFOCTR	R/W	H'FE40 002A		16	
	D1FIFO 端口选择寄存器	D1FIFOSEL	R/W	H'FE40 002C		16	
	D1FIFO 端口控制寄存器	D1FIFOCTR	R/W	H'FE40 002E		16	
	中断允许寄存器 0	INTENB0	R/W	H'FE40 0030		16	
	中断允许寄存器 1	INTENB1	R/W	H'FE40 0032		16	
	BRDY 中断允许寄存器	BRDYENB	R/W	H'FE40 0036		16	
	NRDY 中断允许寄存器	NRDYENB	R/W	H'FE40 0038		16	
	BEMP 中断允许寄存器	BEMPENB	R/W	H'FE40 003A		16	
	SOF 输出配置寄存器	SOFCFG	R/W	H'FE40 003C		16	
中断状态寄存器 0	INTSTS0	R/W	H'FE40 0040		16		
中断状态寄存器 1	INTSTS1	R/W	H'FE40 0042		16		

模块	名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度	备注
USB	BRDY 中断状态寄存器	BRDYSTS	R/W	H'FE40 0046		16	
	NRDY 中断状态寄存器	NRDYSTS	R/W	H'FE40 0048		16	
	BEMP 中断状态寄存器	BEMPSTS	R/W	H'FE40 004A		16	
	帧号寄存器	FRMNUM	R/W	H'FE40 004C		16	
	μ 帧号寄存器	UFRMNUM	R/W	H'FE40 004E		16	
	USB 地址寄存器	USBADDR	R	H'FE40 0050		16	
	USB 请求类型寄存器	USBREQ	R	H'FE40 0054		16	
	USB 请求值寄存器	USBVAL	R	H'FE40 0056		16	
	USB 请求索引寄存器	USBINDX	R	H'FE40 0058		16	
	USB 请求长度寄存器	USBLENG	R	H'FE40 005A		16	
	DCP 配置寄存器	DCPCFG	R/W	H'FE40 005C		16	
	DCP 最大包长度寄存器	DCPMAXP	R/W	H'FE40 005E		16	
	DCP 控制寄存器	DCPCTR	R/W	H'FE40 0060		16	
	管道窗口选择寄存器	PIPESEL	R/W	H'FE40 0064		16	
	管道配置寄存器	PIPECFG	R/W	H'FE40 0068		16	
	管道缓冲器指定寄存器	PIPEBUF	R/W	H'FE40 006A		16	
	管道最大包长度寄存器	PIPEMAXP	R/W	H'FE40 006C		16	
	管道周期控制寄存器	PIPEPERI	R/W	H'FE40 006E		16	
	管道 1 控制寄存器	PIPE1CTR	R/W	H'FE40 0070		16	
	管道 2 控制寄存器	PIPE2CTR	R/W	H'FE40 0072		16	
	管道 3 控制寄存器	PIPE3CTR	R/W	H'FE40 0074		16	
	管道 4 控制寄存器	PIPE4CTR	R/W	H'FE40 0076		16	
	管道 5 控制寄存器	PIPE5CTR	R/W	H'FE40 0078		16	
	管道 6 控制寄存器	PIPE6CTR	R/W	H'FE40 007A		16	
	管道 7 控制寄存器	PIPE7CTR	R/W	H'FE40 007C		16	
	管道 8 控制寄存器	PIPE8CTR	R/W	H'FE40 007E		16	
	管道 9 控制寄存器	PIPE9CTR	R/W	H'FE40 0080		16	
	管道 1 事务计数允许寄存器	PIPE1TRE	R/W	H'FE40 0090		16	
	管道 1 事务计数寄存器	PIPE1TRN	R/W	H'FE40 0092		16	
	管道 2 事务计数允许寄存器	PIPE2TRE	R/W	H'FE40 0094		16	
	管道 2 事务计数寄存器	PIPE2TRN	R/W	H'FE40 0096		16	
	管道 3 事务计数允许寄存器	PIPE3TRE	R/W	H'FE40 0098		16	
	管道 3 处理计数寄存器	PIPE3TRN	R/W	H'FE40 009A		16	
	管道 4 事务计数允许寄存器	PIPE4TRE	R/W	H'FE40 009C		16	
	管道 4 事务计数寄存器	PIPE4TRN	R/W	H'FE40 009E		16	
	管道 5 事务计数允许寄存器	PIPE5TRE	R/W	H'FE40 00A0		16	
	管道 5 事务计数寄存器	PIPE5TRN	R/W	H'FE40 00A2		16	
	设备地址 0 配置寄存器	DEVADD0	R/W	H'FE40 00D0		16	
	设备地址 1 配置寄存器	DEVADD1	R/W	H'FE40 00D2		16	
	设备地址 2 配置寄存器	DEVADD2	R/W	H'FE40 00D4		16	

模块	名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度	备注
USB	设备地址 3 配置寄存器	DEVADD3	R/W	H'FE40 00D6		16	
	设备地址 4 配置寄存器	DEVADD4	R/W	H'FE40 00D8		16	
	设备地址 5 配置寄存器	DEVADD5	R/W	H'FE40 00DA		16	
	设备地址 6 配置寄存器	DEVADD6	R/W	H'FE40 00DC		16	
	设备地址 7 配置寄存器	DEVADD7	R/W	H'FE40 00DE		16	
	设备地址 8 配置寄存器	DEVADD8	R/W	H'FE40 00E0		16	
	设备地址 9 配置寄存器	DEVADD9	R/W	H'FE40 00E2		16	
	设备地址 A 配置寄存器	DEVADDA	R/W	H'FE40 00E4		16	
LCDC	配色数据寄存器 00 ~ FF	LDPR00 ~ LDPRFF	R/W	H'FFE3 0000 ~ H'FFE3 03FC	H'1FE3 0000 ~ H'1FE3 03FC	32	
	LCDC 输入时钟寄存器	LDICKR	R/W	H'FFE3 0400	H'1FE3 0400	16	
	LCDC 模块类型寄存器	LDMTR	R/W	H'FFE3 0402	H'1FE3 0402	16	
	LCDC 数据格式寄存器	LDDFR	R/W	H'FFE3 0404	H'1FE3 0404	16	
	LCDC 上部显示屏的数据开始地址寄存器	LDSARU	R/W	H'FFE3 0408	H'1FE3 0408	32	
	LCDC 下部显示屏的数据开始地址寄存器	LDSARL	R/W	H'FFE3 040C	H'1FE3 040C	32	
	LCDC 显示屏的数据行地址偏移寄存器	LDLAOR	R/W	H'FFE3 0410	H'1FE3 0410	16	
	LCDC 配色控制寄存器	LDPALCR	R/W	H'FFE3 0412	H'1FE3 0412	16	
	LCDC 水平字符数寄存器	LDHCNR	R/W	H'FFE3 0414	H'1FE3 0414	16	
	LCDC 水平同步信号寄存器	LDHSYNR	R/W	H'FFE3 0416	H'1FE3 0416	16	
	LCDC 垂直显示行数寄存器	LDVDLNR	R/W	H'FFE3 0418	H'1FE3 0418	16	
	LCDC 垂直总行数寄存器	LDVTLNR	R/W	H'FFE3 041A	H'1FE3 041A	16	
	LCDC 垂直同步信号寄存器	LDVSYNR	R/W	H'FFE3 041C	H'1FE3 041C	16	
	LCDC AC 调制信号交替行数寄存器	LDACLNR	R/W	H'FFE3 041E	H'1FE3 041E	16	
	LCDC 中断控制寄存器	LDINTR	R/W	H'FFE3 0420	H'1FE3 0420	16	
	LCDC 电源控制模式寄存器	LDPMMR	R/W	H'FFE3 0424	H'1FE3 0424	16	
	LCDC 电源顺序期间寄存器	LDPSPR	R/W	H'FFE3 0426	H'1FE3 0426	16	
	LCDC 控制寄存器	LDCNTR	R/W	H'FFE3 0428	H'1FE3 0428	16	
	LCDC 用户指定中断控制寄存器	LDUINTR	R/W	H'FFE3 0434	H'1FE3 0434	16	
	LCDC 用户指定中断行数寄存器	LDUINLNR	R/W	H'FFE3 0436	H'1FE3 0436	16	
LCDC 存储器存取间隔数寄存器	LDLIRNR	R/W	H'FFE3 0440	H'1FE3 0440	16		
VDC2 图形部 1	图形部控制寄存器	GRCMEN1	R/W	H'FFEC 0000*	H'1FEC 0000*	32/16/8	
	总线控制寄存器	GRCBUSCNT1	R/W	H'FFEC 0004*	H'1FEC 0004*	32/16/8	
	保留	—	R	H'FFEC 0008*	H'1FEC 0008*	32/16/8	
	保留	—	R	H'FFEC 000C*	H'1FEC 000C*	32/16/8	
	保留	—	R	H'FFEC 0300*	H'1FEC 0300*	32/16/8	
	保留	—	R	H'FFEC 0304*	H'1FEC 0304*	32/16/8	
	图形图像基址寄存器	GROPSADR1	R/W	H'FFEC 0308*	H'1FEC 0308*	32/16/8	

模块	名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度	备注
VDC2 图形部 1	图形图像区寄存器	GROPSWH1	R/W	H'FFEC 030C*	H'1FEC 030C*	32/16/8	
	图形图像行偏移寄存器	GROPSOFST1	R/W	H'FFEC 0310*	H'1FEC 0310*	32/16/8	
	图形图像起始位置寄存器	GROPDPHV1	R/W	H'FFEC 0314*	H'1FEC 0314*	32/16/8	
	保留	—	R	H'FFEC 0318*	H'1FEC 0318*	32/16/8	
	保留	—	R	H'FFEC 031C*	H'1FEC 031C*	32/16/8	
	保留	—	R	H'FFEC 0320*	H'1FEC 0320*	32/16/8	
	保留	—	R	H'FFEC 0324*	H'1FEC 0324*	32/16/8	
	保留	—	R	H'FFEC 0328*	H'1FEC 0328*	32/16/8	
	图形图像区外的颜色设定寄存器	GROPBASE RGB1	R/W	H'FFEC 032C*	H'1FEC 032C*	32/16/8	
VDC2 图形部 2	图形部控制寄存器	GRCMEN2	R/W	H'FFED 0000*	H'1FED 0000*	32/16/8	
	总线控制寄存器	GRCBUSCNT2	R/W	H'FFED 0004*	H'1FED 0004*	32/16/8	
	保留	—	R	H'FFED 0008*	H'1FED 0008*	32/16/8	
	保留	—	R	H'FFED 000C*	H'1FED 000C*	32/16/8	
	保留	—	R	H'FFED 0300*	H'1FED 0300*	32/16/8	
	保留	—	R	H'FFED 0304*	H'1FED 0304*	32/16/8	
	图形图像基址寄存器	GROPSADR2	R/W	H'FFED 0308*	H'1FED 0308*	32/16/8	
	图形图像区寄存器	GROPSWH2	R/W	H'FFED 030C*	H'1FED 030C*	32/16/8	
	图形图像行偏移寄存器	GROPSOFST2	R/W	H'FFED 0310*	H'1FED 0310*	32/16/8	
	图形图像起始位置寄存器	GROPDPHV2	R/W	H'FFED 0314*	H'1FED 0314*	32/16/8	
	α 控制区寄存器	GROPEWH2	R/W	H'FFED 0318*	H'1FED 0318*	32/16/8	
	α 控制区起始位置寄存器	GROPEDPHV2	R/W	H'FFED 031C*	H'1FED 031C*	32/16/8	
	α 控制寄存器	GROPEDPA2	R/W	H'FFED 0320*	H'1FED 0320*	32/16/8	
	色键控制寄存器	GROPCRKY0_2	R/W	H'FFED 0324*	H'1FED 0324*	32/16/8	
	色度颜色指定寄存器	GROPCRKY1_2	R/W	H'FFED 0328*	H'1FED 0328*	32/16/8	
	图形图像区外的颜色设定寄存器	GROPBASE RGB2	R/W	H'FFED 032C*	H'1FED 032C*	32/16/8	
	VDC2 图形部 3	图形部控制寄存器	GRCMEN3	R/W	H'FFEE 0000*	H'1FEE 0000*	32/16/8
总线控制寄存器		GRCBUSCNT3	R/W	H'FFEE 0004*	H'1FEE 0004*	32/16/8	
保留		—	R	H'FFEE 0008*	H'1FEE 0008*	32/16/8	
保留		—	R	H'FFEE 000C*	H'1FEE 000C*	32/16/8	
保留		—	R	H'FFEE 0300*	H'1FEE 0300*	32/16/8	
保留		—	R	H'FFEE 0304*	H'1FEE 0304*	32/16/8	
图形图像基址寄存器		GROPSADR3	R/W	H'FFEE 0308*	H'1FEE 0308*	32/16/8	
图形图像区寄存器		GROPSWH3	R/W	H'FFEE 030C*	H'1FEE 030C*	32/16/8	
图形图像行偏移寄存器		GROPSOFST3	R/W	H'FFEE 0310*	H'1FEE 0310*	32/16/8	
图形图像起始位置寄存器		GROPDPHV3	R/W	H'FFEE 0314*	H'1FEE 0314*	32/16/8	
α 控制区寄存器		GROPEWH3	R/W	H'FFEE 0318*	H'1FEE 0318*	32/16/8	
α 控制区起始位置寄存器		GROPEDPHV3	R/W	H'FFEE 031C*	H'1FEE 031C*	32/16/8	
α 控制寄存器		GROPEDPA3	R/W	H'FFEE 0320*	H'1FEE 0320*	32/16/8	
色键控制寄存器		GROPCRKY0_3	R/W	H'FFEE 0324*	H'1FEE 0324*	32/16/8	

模块	名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度	备注
VDC2 图形部 3	色度颜色指定寄存器	GROPCRKY1_3	R/W	H'FFEE 0328*	H'1FEE 0328*	32/16/8	
	图形图像区外的颜色设定寄存器	GROPBASE RGB3	R/W	H'FFEE 032C*	H'1FEE 032C*	32/16/8	
VDC2 图形部 4	图形部控制寄存器	GRCMEN4	R/W	H'FFEF 0000*	H'1FEF 0000*	32/16/8	
	总线控制寄存器	GRCBUSCNT4	R/W	H'FFEF 0004*	H'1FEF 0004*	32/16/8	
	保留	—	R	H'FFEF 0008*	H'1FEF 0008*	32/16/8	
	保留	—	R	H'FFEF 000C*	H'1FEF 000C*	32/16/8	
	保留	—	R	H'FFEF 0300*	H'1FEF 0300*	32/16/8	
	保留	—	R	H'FFEF0304*	H'1FEF 0304*	32/16/8	
	图形图像基址寄存器	GROPSADR4	R/W	H'FFEF 0308*	H'1FEF 0308*	32/16/8	
	图形图像区寄存器	GROPSWH4	R/W	H'FFEF 030C*	H'1FEF 030C*	32/16/8	
	图形图像行偏移寄存器	GROPSOFST4	R/W	H'FFEF 0310*	H'1FEF 0310*	32/16/8	
	图形图像开起始位置寄存器	GROPDPHV4	R/W	H'FFEF 0314*	H'1FEF 0314*	32/16/8	
	α 控制区寄存器	GROPEWH4	R/W	H'FFEF 0318*	H'1FEF 0318*	32/16/8	
	α 控制区开始位置寄存器	GROPEDPHV4	R/W	H'FFEF 031C*	H'1FEF 031C*	32/16/8	
	α 控制寄存器	GROPEDPA4	R/W	H'FFEF 0320*	H'1FEF 0320*	32/16/8	
	色键控制寄存器	GROPCRKY0_4	R/W	H'FFEF 0324*	H'1FEF 0324*	32/16/8	
	色度颜色指定寄存器	GROPCRKY1_4	R/W	H'FFEF 0328*	H'1FEF 0328*	32/16/8	
	图形图像区外的颜色设定寄存器	GROPBASE RGB4	R/W	H'FFEF 032C*	H'1FEF 032C*	32/16/8	
VDC2 显示 控制部	SG 模式设定寄存器	SGMODE	R/W	H'FFEB 0000*	H'1FEB 0000*	32/16/8	
	中断输出控制寄存器	SGINTCNT	R/W	H'FFEB 0004*	H'1FEB 0004*	32/16/8	
	同步信号控制寄存器	SYNCNT	R/W	H'FFEB 0008*	H'1FEB 0008*	32/16/8	
	外部输入同步信号时序控制寄存器	EXTSYNCNT	R/W	H'FFEB 000C*	H'1FEB 000C*	32/16/8	
	保留	—	R	H'FFEB 0100*	H'1FEB 0100*	32/16/8	
	同步信号大小寄存器	SYNSIZE	R/W	H'FFEB0104*	H'1FEB0104*	32/16/8	
	垂直同步信号时序控制寄存器	VSYNCTIM	R/W	H'FFEB 0108*	H'1FEB 0108*	32/16/8	
	水平同步信号时序控制寄存器	HSYNCTIM	R/W	H'FFEB 010C*	H'1FEB 010C*	32/16/8	
	栅极时钟信号时序控制寄存器	CLSTIM	R/W	H'FFEB 0110*	H'1FEB 0110*	32/16/8	
	采样开始信号时序控制寄存器	SPLTIM	R/W	H'FFEB 0118*	H'1FEB 0118*	32/16/8	
	栅极控制信号时序控制寄存器	COMTIM	R/W	H'FFEB 011C*	H'1FEB 011C*	32/16/8	
	SGDE 区起始位置寄存器	SGDESTART	R/W	H'FFEB 0120*	H'1FEB 0120*	32/16/8	
	SGDE 区大小寄存器	SGDESIZE	R/W	H'FFEB 0124*	H'1FEB 0124*	32/16/8	
	CDE 色度颜色指定寄存器	CDECRKY	R/W	H'FFEB 0128*	H'1FEB 0128*	32/16/8	
	保留	—	R	H'FFEB 0148*	H'1FEB 0148*	32/16/8	
	T-1004 控制寄存器	T1004CNT	R/W	H'FFEB 0200*	H'1FEB 0200*	32/16/8	
	T-1004 视频起始位置寄存器	T1004OFFSET	R/W	H'FFEB 0204*	H'1FEB 0204*	32/16/8	
	保留	—	R	H'FFEB 0208*	H'1FEB 0208*	32/16/8	
	保留	—	R	H'FFEB 020C*	H'1FEB 020C*	32/16/8	

模块	名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度	备注
FLCTL	共用控制寄存器	FLCMNCR	R/W	H'FFE9 0000	H'1FE9 0000	32	
	命令控制寄存器	FLCMDRCR	R/W	H'FFE9 0004	H'1FE9 0004	32	
	命令码寄存器	FLCMCDR	R/W	H'FFE9 0008	H'1FE9 0008	32	
	地址寄存器	FLADR	R/W	H'FFE9 000C	H'1FE9 000C	32	
	地址寄存器 2	FLADR2	R/W	H'FFE9 003C	H'1FE9 003C	32	
	数据寄存器	FLDATAR	R/W	H'FFE9 0010	H'1FE9 0010	32	
	数据计数寄存器	FLDTCNTR	R/W	H'FFE9 0014	H'1FE9 0014	32	
	中断 DMA 控制寄存器	FLINTDMACR	R/W	H'FFE9 0018	H'1FE9 0018	32	
	就绪 / 忙超时设定寄存器	FLBSYTMR	R/W	H'FFE9 001C	H'1FE9 001C	32	
	就绪 / 忙超时计数器	FLBSYCNT	R	H'FFE9 0020	H'1FE9 0020	32	
	数据 FIFO 寄存器	FLDTFIFO	R/W	H'FFE9 0024/ H'FFE9 0050	H'1FE9 0024/ H'1FE9 0050	32	
	管理码 FIFO 寄存器	FLECFIFO	R/W	H'FFE9 0028/ H'FFE9 0060	H'1FE9 0028/ H'1FE9 0060	32	
	传送控制寄存器	FLTRCR	R/W	H'FFE9 002C	H'1FE9 002C	8	
SRC	SRC 输入数据寄存器	SRCID	R/W	H'FFF3 0000	H'1FF3 0000	16、32	
	SRC 输出数据寄存器	SRCOD	R	H'FFF3 0004	H'1FF3 0004	16、32	
	SRC 输入数据控制寄存器	SRCIDCTRL	R/W	H'FFF3 0008	H'1FF3 0008	16	
	SRC 输出数据控制寄存器	SRCODCTRL	R/W	H'FFF3 000A	H'1FF3 000A	16	
	SRC 控制寄存器	SRCCTRL	R/W	H'FFF3 000C	H'1FF3 000C	16	
	SRC 状态寄存器	SRCSTAT	R/(W)*9	H'FFF3 000E	H'1FF3 000E	16	
GPIO	端口 A 控制寄存器	PTIO_A	R/W	H'FFF1 0000*	H'1FF1 0000*	16*10	
	端口 B 控制寄存器	PTIO_B	R/W	H'FFF1 0004*	H'1FF1 0004*	16*10	
	端口 C 控制寄存器	PTIO_C	R/W	H'FFF1 0008*	H'1FF1 0008*	16*10	
	端口 D 控制寄存器	PTIO_D	R/W	H'FFF1 000C*	H'1FF1 000C*	16*10	
	端口 E 控制寄存器	PTIO_E	R/W	H'FFF1 0010*	H'1FF1 0010*	16*10	
	端口 F 控制寄存器	PTIO_F	R/W	H'FFF1 0014*	H'1FF1 0014*	16*10	
	端口 G 控制寄存器	PTIO_G	R/W	H'FFF1 0018*	H'1FF1 0018*	16*10	
	端口 H 控制寄存器	PTIO_H	R/W	H'FFF1 001C*	H'FFF1 001C*	16*10	
	端口 I 控制寄存器	PTIO_I	R/W	H'FFF1 0020*	H'FFF1 0020*	16*10	
	端口 J 控制寄存器	PTIO_J	R/W	H'FFF1 0024*	H'FFF1 0024*	16*10	
	端口 A 数据寄存器	PTDAT_A	R/W	H'FFF1 0040*	H'1FF1 0040*	16*10	
	端口 B 数据寄存器	PTDAT_B	R/W	H'FFF1 0044*	H'1FF1 0044*	16*10	
	端口 C 数据寄存器	PTDAT_C	R/W	H'FFF1 0048*	H'1FF1 0048*	16*10	
	端口 D 数据寄存器	PTDAT_D	R/W	H'FFF1 004C*	H'1FF1 004C*	16*10	
	端口 E 数据寄存器	PTDAT_E	R/W	H'FFF1 0050*	H'1FF1 0050*	16*10	
	端口 F 数据寄存器	PTDAT_F	R/W	H'FFF1 0054*	H'1FF1 0054*	16*10	
	端口 G 数据寄存器	PTDAT_G	R/W	H'FFF1 0058*	H'1FF1 0058*	16*10	
	端口 H 数据寄存器	PTDAT_H	R/W	H'FFF1 005C*	H'1FF1 005C*	16*10	
	端口 I 数据寄存器	PTDAT_I	R/W	H'FFF1 0060*	H'1FF1 0060*	16*10	
	端口 J 数据寄存器	PTDAT_J	R/W	H'FFF10064*	H'1FF1 0064*	16*10	

模块	名称	略称	R/W	P4 区地址 *	区域 7 地址 *	存取长度	备注
GPIO	输入引脚上拉控制寄存器	PTPUL_SPCL	R/W	H'FFF1 00E0*	H'1FF1 00E0*	16*10	
	引脚选择寄存器 A	PTSEL_A	R/W	H'FFF1 0080*	H'1FF1 0080*	16*10	
	引脚选择寄存器 B	PTSEL_B	R/W	H'FFF1 0084*	H'1FF1 0084*	16*10	
	引脚选择寄存器 C	PTSEL_C	R/W	H'FFF1 0088*	H'1FF1 0088*	16*10	
	引脚选择寄存器 D	PTSEL_D	R/W	H'FFF1 008C*	H'1FF1 008C*	16*10	
	引脚选择寄存器 E	PTSEL_E	R/W	H'FFF1 0090*	H'1FF1 0090*	16*10	
	引脚选择寄存器 F	PTSEL_F	R/W	H'FFF1 0094*	H'1FF1 0094*	16*10	
	引脚选择寄存器 G	PTSEL_G	R/W	H'FFF1 0098*	H'1FF1 0098*	16*10	
	引脚选择寄存器 H	PTSEL_H	R/W	H'FFF1 009C*	H'1FF1 009C*	16*10	
	引脚选择寄存器 I	PTSEL_I	R/W	H'FFF1 00A0*	H'1FF1 00A0*	16*10	
	引脚选择寄存器 J	PTSEL_J	R/W	H'FFF1 00A4*	H'1FF1 00A4*	16*10	
	引脚选择寄存器 K	PTSEL_K	R/W	H'FFF1 00A8*	H'1FF1 00A8*	16*10	
	引脚选择寄存器 P	PTSEL_P	R/W	H'FFF1 00AC*	H'1FF1 00AC*	16*10	
	引脚选择寄存器 R	PTSEL_R	R/W	H'FFF1 00B0*	H'1FF1 00B0*	16*10	
	引脚选择寄存器 S	PTSEL_S	R/W	H'FFF1 00B4*	H'1FF1 00B4*	16*10	
	Hi-Z 寄存器 A	PTHIZ_A	R/W	H'FFF1 00E8*	H'1FF1 00E8*	16*10	
	Hi-Z 寄存器 B	PTHIZ_B	R/W	H'FFF1 00EC*	H'1FF1 00EC*	16*10	
	特殊选择寄存器	PTSEL_SPCL	R/W	H'FFF 100F0	H'1FF1 00F0	16*10	
低功耗模式	待机控制寄存器 *11	STBCR	R/W	H'FFC8 0020	H'1FC8 0020	32	
	模块停止寄存器 0	MSTPCR0	R/W	H'FFC8 0030	H'1FC8 0030	32	
	模块停止寄存器 1	MSTPCR1	R/W	H'FFC8 0038	H'1FC8 0038	32	
UBC	匹配条件设定寄存器 0	CBR0	R/W	H'FF20 0000	H'1F20 0000	32	
	匹配运行设定寄存器 0	CRR0	R/W	H'FF20 0004	H'1F20 0004	32	
	匹配地址设定寄存器 0	CAR0	R/W	H'FF20 0008	H'1F20 0008	32	
	匹配地址屏蔽设定寄存器 0	CAMR0	R/W	H'FF20 000C	H'1F20 000C	32	
	匹配条件设定寄存器 1	CBR1	R/W	H'FF20 0020	H'1F20 0020	32	
	匹配运行设定寄存器 1	CRR1	R/W	H'FF20 0024	H'1F20 0024	32	
	匹配地址设定寄存器 1	CAR1	R/W	H'FF20 0028	H'1F20 0028	32	
	匹配地址屏蔽设定寄存器 1	CAMR1	R/W	H'FF20 002C	H'1F20 002C	32	
	匹配数据设定寄存器 1	CDR1	R/W	H'FF20 0030	H'1F20 0030	32	
	匹配数据屏蔽设定寄存器 1	CDMR1	R/W	H'FF20 0034	H'1F20 0034	32	
	执行次数断点寄存器 1	CETR1	R/W	H'FF20 0038	H'1F20 0038	32	
	通道匹配标志寄存器	CCMFR	R/W	H'FF20 0600	H'1F20 0600	32	
	断点控制寄存器	CBCR	R/W	H'FF20 0620	H'1F20 0620	32	

【注】 * P4 区地址是使用虚拟地址空间 P4 区的地址，区域 7 地址是使用 TLB 存取物理地址空间区域 7 的地址。

- *1 为了清除标志，CHCR 的 HE 和 TE 位只能在读 1 后写 0。
- *2 为了清除标志，DMAOR 的 AE 和 NMIF 位只能在读 1 后写 0。
- *3 必须以指定的存取长度进行存取。
- *4 为了清除标志，只能写 0。bit15 ~ 8、3、2 是只读位，不能写数据。
- *5 为了清除标志，只能写 0。bit15 ~ 1 是只读位，不能写数据。
- *6 为了清除标志，只能给 bit4 ~ 0 写 0。
- *7 为了清除标志，只能给 bit6 ~ 0 写 0。

- *8 此寄存器的 bit26 和 bit27 是可读写位，其他是只读位。详细内容请参照“18.3.17 状态寄存器 0 ~ 5 (SSISR0 ~ 5)”。
- *9 bit15 ~ 3 是只读位。bit2 ~ 0 只能在读 1 后写 0。
- *10 只能对寄存器进行 16 位存取。必须以指定的存取长度进行读写。
- *11 有关待机控制寄存器，请参照图 10.1。

模块	名称	略称	R/W	P4 区地址	区域 7 地址	存取长度	备注 (初始值)
MCU	版本控制寄存器	VCR	R/W	H'FF80 0000		32	H'0B04 0000 0000 0000
	存储器接口模式寄存器	MIM	R/W	H'FF80 0008		32	H'0000 0000 061A 0x40
	SDRAM 控制寄存器	SCR	R/W	H'FF80 0010		32	H'0000 0000 0000 0000
	SDRAM 时序寄存器	STR	R/W	H'FF80 0018		32	H'0000 0000 00FF FFE7
	SDRAM 行属性寄存器	SDRA	R/W	H'FF80 0030		32	H'0000 0000 0000 0200
	SDRAM 模式寄存器	SDMR	R/W	H'FFAx xxxx		32	—
	仲裁模式寄存器	AMR	R/W	H'FF80 0200		32	H'0000 0000 0400 0000
	线性平铺转换控制寄存器	LTC0	R/W	H'FF80 0100		32	H'0000 0000 0000 0000
	线性平铺区起始地址寄存器	LTAD0	R/W	H'FF80 0108		32	H'0000 0000 0000 0000
	线性平铺区起始地址屏蔽寄存器	LTAM0	R/W	H'FF80 0110		32	H'0000 0000 0000 0000
	线性平铺转换控制寄存器	LTC1	R/W	H'FF80 0118		32	H'0000 0000 0000 0000
	线性平铺区起始地址寄存器	LTAD1	R/W	H'FF80 0120		32	H'0000 0000 0000 0000
	线性平铺区起始地址屏蔽寄存器	LTAM1	R/W	H'FF80 0128		32	H'0000 0000 0000 0000
	线性平铺转换控制寄存器	LTC2	R/W	H'FF80 0130		32	H'0000 0000 0000 0000
	线性平铺区起始地址寄存器	LTAD2	R/W	H'FF80 0138		32	H'0000 0000 0000 0000
	线性平铺区起始地址屏蔽寄存器	LTAM2	R/W	H'FF80 0140		32	H'0000 0000 0000 0000
	线性平铺转换控制寄存器	LTC3	R/W	H'FF80 0148		32	H'0000 0000 0000 0000
线性平铺区起始地址寄存器	LTAD3	R/W	H'FF80 0150		32	H'0000 0000 0000 0000	

模块	名称	略称	R/W	P4 区地址	区域 7 地址	存取长度	备注 (初始值)
MCU	线性平铺区起始地址屏蔽寄存器	LTAM3	R/W	H'FF80 0158		32	H'0000 0000 0000 0000
	线性平铺转换控制寄存器	LTC4	R/W	H'FF80 0160		32	H'0000 0000 0000 0000
	线性平铺区起始地址寄存器	LTAD4	R/W	H'FF80 0168		32	H'0000 0000 0000 0000
	线性平铺区起始地址屏蔽寄存器	LTAM4	R/W	H'FF80 0170		32	H'0000 0000 0000 0000
	线性平铺转换控制寄存器	LTC5	R/W	H'FF80 0178		32	H'0000 0000 0000 0000
	线性平铺区起始地址寄存器	LTAD5	R/W	H'FF80 0180		32	H'0000 0000 0000 0000
	线性平铺区起始地址屏蔽寄存器	LTAM5	R/W	H'FF80 0188		32	H'0000 0000 0000 0000
	线性平铺转换控制寄存器	LTC6	R/W	H'FF80 0190		32	H'0000 0000 0000 0000
	线性平铺区起始地址寄存器	LTAD6	R/W	H'FF80 0198		32	H'0000 0000 0000 0000
	线性平铺区起始地址屏蔽寄存器	LTAM6	R/W	H'FF80 01A0		32	H'0000 0000 0000 0000
	线性平铺转换控制寄存器	LTC7	R/W	H'FF80 01A8		32	H'0000 0000 0000 0000
	线性平铺区起始地址寄存器	LTAD7	R/W	H'FF80 01B0		32	H'0000 0000 0000 0000
	线性平铺区起始地址屏蔽寄存器	LTAM7	R/W	H'FF80 01B8		32	H'0000 0000 0000 0000
	请求屏蔽设定寄存器	RQM	R/W	H'FF80 0218		32	H'0000 0000 0000 0000
	总线控制寄存器	BCR	R/W	H'FF80 1000		32	H'0000 0000 3800 0000
	CS0 总线控制寄存器	CS0BCR	R/W	H'FF80 2000		32	H'0000 0000 7777 7x80
	CS0 等待控制寄存器	CS0WCR	R/W	H'FF80 2008		32	H'0000 0000 7777 770F
	CS3 总线控制寄存器	CS3BCR	R/W	H'FF80 2030		32	H'0000 0000 7777 7380
CS3 等待控制寄存器	CS3WCR	R/W	H'FF80 2038		32	H'0000 0000 7777 770F	
ATAPI	ATAPI 状态	ATAPI_STATUS	R/W	H'FFF0 0084		32	
	中断允许	ATAPI_INT_ENABLE	R/W	H'FFF0 0088		32	
	PIO 时序	ATAPI_PIO_TIMING	R/W	H'FFF0 008C		32	
	多字 DMA 时序	ATAPI_MULTI_TIMING	R/W	H'FFF0 0090		32	
	超级 DMA 时序	ATAPI_ULTRA_TIMING	R/W	H'FFF0 0094		32	

模块	名称	略称	R/W	P4 区地址	区域 7 地址	存取长度	备注 (初始值)
ATAPI	描述符表基址	ATAPI_DTBA_ADR	R/W	H'FFF0 0098		32	
	DMA 起始地址	ATAPI_DMA_START_ADR	R/W	H'FFF0 009C		32	
	DMA 传送计数	ATAPI_DMA_TRANS_CNT	R/W	H'FFF0 00A0		32	
	ATAPI 控制 2	ATAPI_CONTROL2	R/W	H'FFF0 00A4		32	
	保留		R	H'FFF0 00A8		32	
	保留		R	H'FFF0 00AC		32	
	ATAPI 信号状态	ATAPI_SIG_ST	R	H'FFF0 00B0		32	
	字节交换	ATAPI_BYTE_SWAP	R/W	H'FFF0 00BC		32	

【注】 * 必须以长字（32 位）为单位存取上述寄存器，禁止字节存取和字存取。

模块	名称	略称	R/W	P4 区地址 *2	区域 7 地址 *2	存取长度	备注 (WPR)*1
G2D	系统控制寄存器	SCLR	R/W	H'FFEA 0000	H'1FEA 0000	32	×
	状态寄存器	SR	R	H'FFEA 0004	H'1FEA 0004	32	×
	状态寄存器清除寄存器	SRCLR	W	H'FFEA 0008	H'1FEA 0008	32	×
	中断允许寄存器	IER	R/W	H'FFEA 000C	H'1FEA 000C	32	○
	中断命令 ID 寄存器	ICIDR	R	H'FFEA 0010	H'1FEA 0010	32	×
	返回地址寄存器 0	RTN0R	R	H'FFEA 0040	H'1FEA 0040	32	○
	返回地址寄存器 1	RTN1R	R	H'FFEA 0044	H'1FEA 0044	32	○
	显示列表起始地址寄存器	DLSAR	R/W	H'FFEA 0048	H'1FEA 0048	32	×
	2 维源区域起始地址寄存器	SSAR	R/W	H'FFEA 004C	H'1FEA 004C	32	○
	绘图起始地址寄存器	RSAR	R/W	H'FFEA 0050	H'1FEA 0050	32	○
	工作区起始地址寄存器	WSAR	R/W	H'FFEA 0054	H'1FEA 0054	32	○
	源跨距寄存器	SSTRR	R/W	H'FFEA 0058	H'1FEA 0058	32	○
	目标跨距寄存器	DSTRR	R/W	H'FFEA 005C	H'1FEA 005C	32	○
	字节排序转换控制寄存器	ENDCVR	R/W	H'FFEA 0060	H'1FEA 0060	32	×
	源透明色寄存器	STCR	R/W	H'FFEA 0080	H'1FEA 0080	32	○
	目标透明色寄存器	DTCR	R/W	H'FFEA 0084	H'1FEA 0084	32	○
	α 值寄存器	ALPHR	R/W	H'FFEA 0088	H'1FEA 0088	32	○
	颜色偏移寄存器	COFSR	R/W	H'FFEA 008C	H'1FEA 008C	32	○
	绘图控制寄存器	RCLR	R/W	H'FFEA 00C0	H'1FEA 00C0	32	○
	命令状态寄存器	CSTR	R	H'FFEA 00C4	H'1FEA 00C4	32	×
	当前指针寄存器	CURR	R	H'FFEA 00C8	H'1FEA 00C8	32	×
	局部偏移寄存器	LCOR	R	H'FFEA 00CC	H'1FEA 00CC	32	×
	系统裁剪区 MAX 寄存器	SCLMAR	R	H'FFEA 00D0	H'1FEA 00D0	32	○
	用户裁剪区 MIN 寄存器	UCLMIR	R	H'FFEA 00D4	H'1FEA 00D4	32	○
	用户裁剪区 MAX 寄存器	UCLMAR	R	H'FFEA 00D8	H'1FEA 00D8	32	○
	相对用户裁剪区 MIN 寄存器	RUCLMIR	R	H'FFEA 00DC	H'1FEA 00DC	32	○
	相对用户裁剪区 MAX 寄存器	RUCLMAR	R	H'FFEA 00E0	H'1FEA 00E0	32	○
	绘图控制 2 寄存器	RCL2R	R/W	H'FFEA 00F0	H'1FEA 00F0	32	○
	图案偏移寄存器	POFSR	R/W	H'FFEA 00F8	H'1FEA 00F8	32	○

模块	名称	略称	R/W	P4 区地址 *2	区域 7 地址 *2	存取长度	备注 (WPR)*1
G2D	坐标变换控制寄存器	GTRCR	R/W	H'FFEA 0100	H'1FEA 0100	32	○
	矩阵参数 A 寄存器	MTRAR	R/W	H'FFEA 0104	H'1FEA 0104	32	○
	矩阵参数 B 寄存器	MTRBR	R/W	H'FFEA 0108	H'1FEA 0108	32	○
	矩阵参数 C 寄存器	MTRCR	R/W	H'FFEA 010C	H'1FEA 010C	32	○
	矩阵参数 D 寄存器	MTRDR	R/W	H'FFEA 0110	H'1FEA 0110	32	○
	矩阵参数 E 寄存器	MTRER	R/W	H'FFEA 0114	H'1FEA 0114	32	○
	矩阵参数 F 寄存器	MTRFR	R/W	H'FFEA 0118	H'1FEA 0118	32	○
	矩阵参数 G 寄存器	MTRGR	R/W	H'FFEA 011C	H'1FEA 011C	32	○
	矩阵参数 H 寄存器	MTRHR	R/W	H'FFEA 0120	H'1FEA 0120	32	○
	矩阵参数 I 寄存器	MTRIR	R/W	H'FFEA 0124	H'1FEA 0124	32	○
	坐标变换偏移 X 寄存器	GTROFSXR	R/W	H'FFEA 0128	H'1FEA 0128	32	○
	坐标变换偏移 Y 寄存器	GTROFSYR	R/W	H'FFEA 012C	H'1FEA 012C	32	○
	Z 裁剪区 MIN 寄存器	ZCLPMINR	R/W	H'FFEA 0130	H'1FEA 0130	32	○
	Z 裁剪区 MAX 寄存器	ZCLPMAXR	R/W	H'FFEA 0134	H'1FEA 0134	32	○
Z 饱和值 MIN 寄存器	ZSATVMINR	R/W	H'FFEA 0138	H'1FEA 0138	32	○	

【注】 *1 WPR 命令的设定 ○：可，×：不可

*2 P4 区地址是使用虚拟地址空间 P4 区的地址，区域 7 地址是使用 TLB 存取物理地址空间区域 7 的地址。必须写指定的地址，否则不保证运行。

模块	名称	略称	R/W	P4 区地址 *1	区域 7 地址 *1	存取长度	备注 (初始值)*2
H-JDI	指令寄存器	SDIR	R	H'FC11 0000	H'1C11 0000	16	H'0EFF
	中断源寄存器	SDINT	R/W	H'FC11 0018	H'1C11 0018	16	H'0000
	边界扫描寄存器	SDBSR	—	—	—	—	—
	旁路寄存器	SDBPR	—	—	—	—	不定值

【注】 *1 P4 区地址是使用虚拟地址空间 P4 区的地址，区域 7 地址是使用 TLB 存取物理地址空间区域 7 的地址。

*2 在 TRST 为低电平或者 TAP 为 Test-Logic-Reset 状态时，寄存器被初始化。

32.2 各处理模式的寄存器状态

表 32.2 各处理模式的寄存器状态 (1)

模块	寄存器略称	上电复位	睡眠	待机
异常处理	TRA	不定值	保持	保持
	EXPEVT	H'0000 0000	保持	保持
	INTEVT	不定值	保持	保持
	EXPMASK	H'0000 0000	保持	保持
MMU	PTEH	不定值	保持	保持
	PTEL	不定值	保持	保持
	TTB	不定值	保持	保持
	TEA	不定值	保持	保持
	MMUCR	H'0000 0000	保持	保持
	PTEA	H'0000 xxx0	保持	保持
	PASCR	H'0000 0000	保持	保持
	IRMCR	H'0000 0000	保持	保持
高速缓存	CCR	H'0000 0000	保持	保持
	QACR0	不定值	保持	保持
	QACR1	不定值	保持	保持
	RAMCR	H'0000 0000	保持	保持
内部存储器	RAMCR	H'0000 0000	保持	保持
CPG	FRQCR	H'x032 0044*1	保持	保持
	PLLCR	H'0000 E001	保持	保持
	VDC2CLKCR	H'0000 0080	保持	保持
DMAC	SAR0	不定值	保持	保持
	DAR0	不定值	保持	保持
	TCR0	不定值	保持	保持
	CHCR0	H'4000 0000	保持	保持
	SAR1	不定值	保持	保持
	DAR1	不定值	保持	保持
	TCR1	不定值	保持	保持
	CHCR1	H'4000 0000	保持	保持
	SAR2	不定值	保持	保持
	DAR2	不定值	保持	保持
	TCR2	不定值	保持	保持
	CHCR2	H'4000 0000	保持	保持
	SAR3	不定值	保持	保持
	DAR3	不定值	保持	保持
	TCR3	不定值	保持	保持
	CHCR3	H'4000 0000	保持	保持
	DMAOR0	H'0000	保持	保持
SAR4	不定值	保持	保持	

模块	寄存器略称	上电复位	睡眠	待机
DMAC	DAR4	不定值	保持	保持
	TCR4	不定值	保持	保持
	CHCR4	H'4000 0000	保持	保持
	SAR5	不定值	保持	保持
	DAR5	不定值	保持	保持
	TCR5	不定值	保持	保持
	CHCR5	H'4000 0000	保持	保持
	SARB0	不定值	保持	保持
	DARB0	不定值	保持	保持
	TCRB0	不定值	保持	保持
	SARB1	不定值	保持	保持
	DARB1	不定值	保持	保持
	TCRB1	不定值	保持	保持
	SARB2	不定值	保持	保持
	DARB2	不定值	保持	保持
	TCRB2	不定值	保持	保持
	SARB3	不定值	保持	保持
	DARB3	不定值	保持	保持
	TCRB3	不定值	保持	保持
	DMARS0	H'0000	保持	保持
DMARS1	H'0000	保持	保持	
DMARS2	H'0000	保持	保持	
MCU	VCR	H'0B04 0000 0000 0000	保持	保持
	MIM	H'0000 0000 061A 0x40	保持	保持
	SCR	H'0000 0000 0000 0000	保持	保持
	STR	H'0000 0000 00FF FFE7	保持	保持
	SDRA	H'0000 0000 0000 0200	保持	保持
	SDMR	—	保持	保持
	AMR	H'0000 0000 0400 0000	保持	保持
	LTC0	H'0000 0000 0000 0000	保持	保持
	LTAD0	H'0000 0000 0000 0000	保持	保持
	LTAM0	H'0000 0000 0000 0000	保持	保持
	LTC1	H'0000 0000 0000 0000	保持	保持
	LTAD1	H'0000 0000 0000 0000	保持	保持
	LTAM1	H'0000 0000 0000 0000	保持	保持
	LTC2	H'0000 0000 0000 0000	保持	保持
	LTAD2	H'0000 0000 0000 0000	保持	保持
	LTAM2	H'0000 0000 0000 0000	保持	保持
	LTC3	H'0000 0000 0000 0000	保持	保持
	LTAD3	H'0000 0000 0000 0000	保持	保持
LTAM3	H'0000 0000 0000 0000	保持	保持	

模块	寄存器略称	上电复位	睡眠	待机
MCU	LTC4	H'0000 0000 0000 0000	保持	保持
	LTAD4	H'0000 0000 0000 0000	保持	保持
	LTAM4	H'0000 0000 0000 0000	保持	保持
	LTC5	H'0000 0000 0000 0000	保持	保持
	LTAD5	H'0000 0000 0000 0000	保持	保持
	LTAM5	H'0000 0000 0000 0000	保持	保持
	LTC6	H'0000 0000 0000 0000	保持	保持
	LTAD6	H'0000 0000 0000 0000	保持	保持
	LTAM6	H'0000 0000 0000 0000	保持	保持
	LTC7	H'0000 0000 0000 0000	保持	保持
	LTAD7	H'0000 0000 0000 0000	保持	保持
	LTAM7	H'0000 0000 0000 0000	保持	保持
	RQM	H'0000 0000 0000 0000	保持	保持
	BCR	H'0000 0000 3800 0000	保持	保持
	CS0BCR	H'0000 0000 7777 7x80	保持	保持
	CS0WCR	H'0000 0000 7777 770F	保持	保持
	CS3BCR	H'0000 0000 7777 7380	保持	保持
	CS3WCR	H'0000 0000 7777 770F	保持	保持
INTC	ICR0	H'x000 0000	保持	保持
	ICR1	H'0000 0000	保持	保持
	INTPRI	H'0000 0000	保持	保持
	INTREQ	H'0000 0000	保持	保持
	INTMSK	H'FF00 0000	保持	保持
	INTMSKCLR	H'0000 0000	保持	保持
	NMIFCR	H'x000 0000	保持	保持
	USERIMASK	H'0000 0000	保持	保持
	INT2PRI0	H'0000 0000	保持	保持
	INT2PRI1	H'0000 0000	保持	保持
	INT2PRI2	H'0000 0000	保持	保持
	INT2PRI3	H'0000 0000	保持	保持
	INT2PRI4	H'0000 0000	保持	保持
	INT2PRI5	H'0000 0000	保持	保持
	INT2PRI6	H'0000 0000	保持	保持
	INT2PRI7	H'0000 0000	保持	保持
	INT2PRI8	H'0000 0000	保持	保持
	INT2PRI9	H'0000 0000	保持	保持
	INT2PRI10	H'0000 0000	保持	保持
	INT2PRI11	H'0000 0000	保持	保持
	INT2PRI12	H'0000 0000	保持	保持
INT2A0	H'xxxx xxxx	保持	保持	
INT2A01	H'xxxx xxxx	保持	保持	

模块	寄存器略称	上电复位	睡眠	待机
INTC	INT2A1	H'0000 0000	保持	保持
	INT2A11	H'0000 0000	保持	保持
	INT2MSKR	H'FFFF FFFF	保持	保持
	INT2MSKR1	H'FFFF FFFF	保持	保持
	INT2MSKCR	H'0000 0000	保持	保持
	INT2MSKCR1	H'0000 0000	保持	保持
	INT2B0	H'xxxx xxxx	保持	保持
	INT2B2	H'xxxx xxxx	保持	保持
	INT2B3	H'xxxx xxxx	保持	保持
	INT2B4	H'xxxx xxxx	保持	保持
	INT2B5	H'xxxx xxxx	保持	保持
	INT2B6	H'xxxx xxxx	保持	保持
	INT2B7	H'xxxx xxxx	保持	保持
	INT2GPIC	H'0000 0000	保持	保持
TMU	TOCR	H'00	保持	保持
	TSTR0	H'00	保持	保持
	TCOR0	H'FFFF FFFF	保持	保持
	TCNT0	H'FFFF FFFF	保持	保持
	TCR0	H'0000	保持	保持
	TCOR1	H'FFFF FFFF	保持	保持
	TCNT1	H'FFFF FFFF	保持	保持
	TCR1	H'0000	保持	保持
	TCOR2	H'FFFF FFFF	保持	保持
	TCNT2	H'FFFF FFFF	保持	保持
	TCR2	H'0000	保持	保持
	TCPR2	H'xxxx xxxx	保持	保持
	TSTR1	H'00	保持	保持
	TCOR3	H'FFFF FFFF	保持	保持
	TCNT3	H'FFFF FFFF	保持	保持
	TCR3	H'0000	保持	保持
	TCOR4	H'FFFF FFFF	保持	保持
	TCNT4	H'FFFF FFFF	保持	保持
	TCR4	H'0000	保持	保持
	TCOR5	H'FFFF FFFF	保持	保持
TCNT5	H'FFFF FFFF	保持	保持	
TCR5	H'0000	保持	保持	
SCIF	SCSMR_0	H'0000	保持	保持
	SCBRR_0	H'FF	保持	保持
	SCSCR_0	H'0000	保持	保持
	SCFTDR_0	不定值	保持	保持
	SCFSR_0	H'0060	保持	保持

模块	寄存器略称	上电复位	睡眠	待机
SCIF	SCFRDR_0	不定值	保持	保持
	SCFCR_0	H'0000	保持	保持
	SCFDR_0	H'0000	保持	保持
	SCSPTR_0	H'0050	保持	保持
	SCLSR_0	H'0000	保持	保持
	SCEMR_0	H'0000	保持	保持
	SCSMR_1	H'0000	保持	保持
	SCBRR_1	H'FF	保持	保持
	SCSCR_1	H'0000	保持	保持
	SCFTDR_1	不定值	保持	保持
	SCFSR_1	H'0060	保持	保持
	SCFRDR_1	不定值	保持	保持
	SCFCR_1	H'0000	保持	保持
	SCFDR_1	H'0000	保持	保持
	SCSPTR_1	H'0050	保持	保持
	SCLSR_1	H'0000	保持	保持
	SCEMR_1	H'0000	保持	保持
	SCSMR_2	H'0000	保持	保持
	SCBRR_2	H'FF	保持	保持
	SCSCR_2	H'0000	保持	保持
	SCFTDR_2	不定值	保持	保持
	SCFSR_2	H'0060	保持	保持
	SCFRDR_2	不定值	保持	保持
	SCFCR_2	H'0000	保持	保持
SCFDR_2	H'0000	保持	保持	
SCSPTR_2	H'0050	保持	保持	
SCLSR_2	H'0000	保持	保持	
SCEMR_2	H'0000	保持	保持	
IIC	ICSCR	H'00	保持	保持
	ICMCR	H'x0	保持	保持
	ICSSR	H'00	保持	保持
	ICMSR	H'00	保持	保持
	ICSIER	H'00	保持	保持
	ICMIER	H'00	保持	保持
	ICCCR	H'00	保持	保持
	ICSAR	H'00	保持	保持
	ICMAR	H'00	保持	保持
	ICRXD	H'00	保持	保持
	ICTXD	H'00	保持	保持
ATAPI	ATAPI_CONTROL	H'0000 0020	保持	保持
	ATAPI_STATUS	H'0000 0000	保持	保持

模块	寄存器略称	上电复位	睡眠	待机
ATAPI	ATAPI_INT_ENABLE	H'0000 0000	保持	保持
	ATAPI_PIO_TIMING	H'0000 0000	保持	保持
	ATAPI_MULTI_TIMING	H'0000 0000	保持	保持
	ATAPI_ULTRA_TIMING	H'0000 0000	保持	保持
	ATAPI_DTBA_ADR	H'0000 0000	保持	保持
	ATAPI_DMA_START_ADR	H'0000 0000	保持	保持
	ATAPI_DMA_TRANS_CNT	H'0000 0000	保持	保持
	ATAPI_CONTROL2	H'0000 0000	保持	保持
	ATAPI_SIG_ST	H'0000 000x	保持	保持
	ATAPI_BYTE_SWAP	H'0000 0000	保持	保持
SSI_DMAC0	SSIDMMR0	H'0000 0000	保持	保持
	SSIRDMAADR0	H'0000 0000	保持	保持
	SSIRDMCNTR0	H'0000 0000	保持	保持
	SSIWMAADR0	H'0000 0000	保持	保持
	SSIWMCNTR0	H'0000 0000	保持	保持
	SSIDMCR0	H'0000 0000	保持	保持
	SSISTPBLCNT0	H'0000 0000	保持	保持
	SSISTPDR0	H'0000 0000	保持	保持
	SSIBLCNTR0	H'0000 0000	保持	保持
	SSIBLCNT0	H'0000 0000	保持	保持
	SSIBLNCNTR0	H'0000 0000	保持	保持
	SSIBLNCNT0	H'0000 0000	保持	保持
	SSIDMMR1	H'0000 0000	保持	保持
	SSIRDMAADR1	H'0000 0000	保持	保持
	SSIRDMCNTR1	H'0000 0000	保持	保持
	SSIWMAADR1	H'0000 0000	保持	保持
	SSIWMCNTR1	H'0000 0000	保持	保持
	SSIDMCR1	H'0000 0000	保持	保持
	SSISTPBLCNT1	H'0000 0000	保持	保持
	SSISTPDR1	H'0000 0000	保持	保持
	SSIBLCNTR1	H'0000 0000	保持	保持
	SSIBLCNT1	H'0000 0000	保持	保持
	SSIBLNCNTR1	H'0000 0000	保持	保持
	SSIBLNCNT1	H'0000 0000	保持	保持
	SSIDMMR2	H'0000 0000	保持	保持
	SSIRDMAADR2	H'0000 0000	保持	保持
	SSIRDMCNTR2	H'0000 0000	保持	保持
	SSIWMAADR2	H'0000 0000	保持	保持
	SSIWMCNTR2	H'0000 0000	保持	保持
	SSIDMCR2	H'0000 0000	保持	保持
SSISTPBLCNT2	H'0000 0000	保持	保持	

模块	寄存器略称	上电复位	睡眠	待机
SSI_DMAC0	SSISTPDR2	H'0000 0000	保持	保持
	SSIBLCNTR2	H'0000 0000	保持	保持
	SSIBLCNT2	H'0000 0000	保持	保持
	SSIBLNCNTR2	H'0000 0000	保持	保持
	SSIBLNCNT2	H'0000 0000	保持	保持
	SSIDMAOR0	H'0000 0000	保持	保持
	SSIDMINTSR0	H'0101 0101	保持	保持
	SSIDMINTMR0	H'1F1F 1F1F	保持	保持
SSI_DMAC1	SSIDMMR3	H'0000 0000	保持	保持
	SSIRDMADR3	H'0000 0000	保持	保持
	SSIRDMCNR3	H'0000 0000	保持	保持
	SSIWDMADR3	H'0000 0000	保持	保持
	SSIWDMCNR3	H'0000 0000	保持	保持
	SSIDMCOR3	H'0000 0000	保持	保持
	SSISTPBLCNT3	H'0000 0000	保持	保持
	SSISTPDR3	H'0000 0000	保持	保持
	SSIBLCNTR3	H'0000 0000	保持	保持
	SSIBLCNT3	H'0000 0000	保持	保持
	SSIBLNCNTR3	H'0000 0000	保持	保持
	SSIBLNCNT3	H'0000 0000	保持	保持
	SSIDMMR4	H'0000 0000	保持	保持
	SSIRDMADR4	H'0000 0000	保持	保持
	SSIRDMCNR4	H'0000 0000	保持	保持
	SSIWDMADR4	H'0000 0000	保持	保持
	SSIWDMCNR4	H'0000 0000	保持	保持
	SSIDMCOR4	H'0000 0000	保持	保持
	SSISTPBLCNT4	H'0000 0000	保持	保持
	SSISTPDR4	H'0000 0000	保持	保持
	SSIBLCNTR4	H'0000 0000	保持	保持
	SSIBLCNT4	H'0000 0000	保持	保持
	SSIBLNCNTR4	H'0000 0000	保持	保持
	SSIBLNCNT4	H'0000 0000	保持	保持
	SSIDMMR5	H'0000 0000	保持	保持
	SSIRDMADR5	H'0000 0000	保持	保持
	SSIRDMCNR5	H'0000 0000	保持	保持
	SSIWDMADR5	H'0000 0000	保持	保持
	SSIWDMCNR5	H'0000 0000	保持	保持
	SSIDMCOR5	H'0000 0000	保持	保持
	SSISTPBLCNT5	H'0000 0000	保持	保持
	SSISTPDR5	H'0000 0000	保持	保持
SSIBLCNTR5	H'0000 0000	保持	保持	

模块	寄存器略称	上电复位	睡眠	待机
SSI_DMAC1	SSIBLCNT5	H'0000 0000	保持	保持
	SSIBLNCNTR5	H'0000 0000	保持	保持
SSI_DMAC1	SSIBLNCNT5	H'0000 0000	保持	保持
	SSIDMAOR1	H'0000 0000	保持	保持
	SSIDMINTSR1	H'0101 0101	保持	保持
	SSIDMINTMR1	H'1F1F 1F1F	保持	保持
SSI_CH0 ~ 5	SSICR0	H'0000 0000	保持	保持
	SSISR0	H'0210 A003	保持	保持
	SSITDR0	H'0000 0000	保持	保持
	SSIRDR0	H'0000 0000	保持	保持
	SSICR1	H'0000 0000	保持	保持
	SSISR1	H'0210 A003	保持	保持
	SSITDR1	H'0000 0000	保持	保持
	SSIRDR1	H'0000 0000	保持	保持
	SSICR2	H'0000 0000	保持	保持
	SSISR2	H'0210 A003	保持	保持
	SSITDR2	H'0000 0000	保持	保持
	SSIRDR2	H'0000 0000	保持	保持
	SSICR3	H'0000 0000	保持	保持
	SSISR3	H'0210 A003	保持	保持
	SSITDR3	H'0000 0000	保持	保持
	SSIRDR3	H'0000 0000	保持	保持
	SSICR4	H'0000 0000	保持	保持
	SSISR4	H'0210 A003	保持	保持
	SSITDR4	H'0000 0000	保持	保持
	SSIRDR4	H'0000 0000	保持	保持
SSICR5	H'0000 0000	保持	保持	
SSISR5	H'0210 A003	保持	保持	
SSITDR5	H'0000 0000	保持	保持	
SSIRDR5	H'0000 0000	保持	保持	
EtherC	ECMR	H'0000 0000	保持	保持
	ECSR	H'0000 0000	保持	保持
	ECSIPR	H'0000 0000	保持	保持
	RFLR	H'0000 0000	保持	保持
	PIR	H'0000 000x	保持	保持
	MAHR	H'0000 0000	保持	保持
	MALR	H'0000 0000	保持	保持
	PSR	H'0000 000x	保持	保持
	TROCR	H'0000 000x	保持	保持
	CDCR	H'0000 0000	保持	保持
	LCCR	H'0000 0000	保持	保持

模块	寄存器略称	上电复位	睡眠	待机
EtherC	CNDCCR	H'0000 0000	保持	保持
	CEFCR	H'0000 0000	保持	保持
	FRECR	H'0000 0000	保持	保持
	TSFRCR	H'0000 0000	保持	保持
	TLFRCR	H'0000 0000	保持	保持
	RFCR	H'0000 0000	保持	保持
	MAFCR	H'0000 0000	保持	保持
	IPGR	H'0000 0000	保持	保持
	APR	H'0000 0000	保持	保持
	MPR	H'0000 0000	保持	保持
	TPAUSER	H'0000 0000	保持	保持
	BCFRR	H'0000 0000	保持	保持
EDMAC	EDMR	H'0000 0000	保持	保持
	EDTRR	H'0000 0000	保持	保持
	EDRRR	H'0000 0000	保持	保持
	TDLAR	H'0000 0000	保持	保持
	RDLAR	H'0000 0000	保持	保持
	EESR	H'0000 0000	保持	保持
	EESIPR	H'0000 0000	保持	保持
	TRSCER	H'0000 0000	保持	保持
	RMFCR	H'0000 0000	保持	保持
	TFTR	H'0000 0000	保持	保持
	FDR	H'0000 0707	保持	保持
	RMCR	H'0000 0000	保持	保持
	TFUCR	H'0000 0000	保持	保持
	RFOCR	H'0000 0000	保持	保持
	RBWAR	H'0000 0000	保持	保持
	RDFAR	H'0000 0000	保持	保持
	TBRAR	H'0000 0000	保持	保持
	TDFAR	H'0000 0000	保持	保持
	FCFTR	H'0007 0007	保持	保持
	RPADIR	H'0000 0000	保持	保持
TRIMD	H'0000 0000	保持	保持	
USB	SYSCFG	H'0000	保持	保持
	BUSWAIT	H'000F	保持	保持
	SYSSTS	H'040x	保持	保持
	DVSTCTR	H'0000	保持	保持
	TESTMODE	H'0000	保持	保持
	D0FBCFG	H'0000	保持	保持
	D1FBCFG	H'0000	保持	保持
	CFIFO	H'0000 0000	保持	保持

模块	寄存器略称	上电复位	睡眠	待机
USB	D0FIFO	H'0000 0000	保持	保持
	D1FIFO	H'0000 0000	保持	保持
	CFIFOSEL	H'0000	保持	保持
	CFIFOCTR	H'0000	保持	保持
	D0FIFOSEL	H'0000	保持	保持
	D0FIFOCTR	H'0000	保持	保持
	D1FIFOSEL	H'0000	保持	保持
	D1FIFOCTR	H'0000	保持	保持
	INTENB0	H'0000	保持	保持
	INTENB1	H'0000	保持	保持
	BRDYENB	H'0000	保持	保持
	NRDYENB	H'0000	保持	保持
	BEMPENB	H'0000	保持	保持
	SOFCFG	H'0000	保持	保持
	INTSTS0	H'0000	保持	保持
	INTSTS1	H'0000	保持	保持
	BRDYSTS	H'0000	保持	保持
	NRDYSTS	H'0000	保持	保持
	BEMPSTS	H'0000	保持	保持
	FRMNUM	H'0000	保持	保持
	UFRMNUM	H'0000	保持	保持
	USBADDR	H'0000	保持	保持
	USBREQ	H'0000	保持	保持
	USBVAL	H'0000	保持	保持
	USBINDX	H'0000	保持	保持
	USBLENG	H'0000	保持	保持
	DCPCFG	H'0000	保持	保持
	DCPMAXP	H'0040	保持	保持
	DCPCTR	H'0040	保持	保持
	PIPESEL	H'0000	保持	保持
	PIPECFG	H'0000	保持	保持
	PIPEBUF	H'0000	保持	保持
	PIPEMAXP	H'0000	保持	保持
	PIPEPERI	H'0000	保持	保持
PIPE1CTR	H'0000	保持	保持	
PIPE2CTR	H'0000	保持	保持	
PIPE3CTR	H'0000	保持	保持	
PIPE4CTR	H'0000	保持	保持	
PIPE5CTR	H'0000	保持	保持	
PIPE6CTR	H'0000	保持	保持	
PIPE7CTR	H'0000	保持	保持	

模块	寄存器略称	上电复位	睡眠	待机
USB	PIPE8CTR	H'0000	保持	保持
	PIPE9CTR	H'0000	保持	保持
	PIPE1TRE	H'0000	保持	保持
	PIPE1TRN	H'0000	保持	保持
	PIPE2TRE	H'0000	保持	保持
	PIPE2TRN	H'0000	保持	保持
	PIPE3TRE	H'0000	保持	保持
	PIPE3TRN	H'0000	保持	保持
	PIPE4TRE	H'0000	保持	保持
	PIPE4TRN	H'0000	保持	保持
	PIPE5TRE	H'0000	保持	保持
	PIPE5TRN	H'0000	保持	保持
	DEVADD0	H'0000	保持	保持
	DEVADD1	H'0000	保持	保持
	DEVADD2	H'0000	保持	保持
	DEVADD3	H'0000	保持	保持
	DEVADD4	H'0000	保持	保持
	DEVADD5	H'0000	保持	保持
	DEVADD6	H'0000	保持	保持
	DEVADD7	H'0000	保持	保持
DEVADD8	H'0000	保持	保持	
DEVADD9	H'0000	保持	保持	
DEVADDA	H'0000	保持	保持	
LCD	LDPR00 ~ LDPRFF	不定值	保持	保持
	LDICKR	H'1101	保持	保持
	LDMTR	H'0109	保持	保持
	LDDFR	H'000C	保持	保持
	LDSARU	H'04000000	保持	保持
	LDSARL	H'04000000	保持	保持
	LDLAOR	H'0280	保持	保持
	LDPALCR	H'0000	保持	保持
	LDHCNR	H'4F52	保持	保持
	LDHSYNR	H'0050	保持	保持
	LDVDLNR	H'01DF	保持	保持
	LDVTLNR	H'01DF	保持	保持
	LDVSYNR	H'01DF	保持	保持
	LDACLNR	H'000C	保持	保持
	LDINTR	H'0000	保持	保持
	LDPMMR	H'0010	保持	保持
	LDPSPR	H'F60F	保持	保持
LDCNTR	H'0000	保持	保持	

模块	寄存器略称	上电复位	睡眠	待机
LCDC	LDUINTR	H'0000	保持	保持
	LDUINTLNR	H'004F	保持	保持
	LDLIRNR	H'0000	保持	保持
G2D	SCLR	H'8000 0000	保持	保持
	SR	不定值	保持	保持
	SRCR	H'0000 0000	保持	保持
	IER	H'0000 0000	保持	保持
	ICIDR	不定值	保持	保持
	RTN0R	不定值	保持	保持
	RTN1R	不定值	保持	保持
	DLSAR	不定值	保持	保持
	SSAR	不定值	保持	保持
	RSAR	不定值	保持	保持
	WSAR	不定值	保持	保持
	SSTRR	不定值	保持	保持
	DSTRR	不定值	保持	保持
	ENDCVR	H'0000 0000	保持	保持
	STCR	不定值	保持	保持
	DTCR	不定值	保持	保持
	ALPHR	不定值	保持	保持
	COFSR	不定值	保持	保持
	RCLR	H'0000 0000	保持	保持
	CSTR	不定值	保持	保持
	CURR	不定值	保持	保持
	LCOR	不定值	保持	保持
	SCLMAR	不定值	保持	保持
	UCLMIR	不定值	保持	保持
	UCLMAR	不定值	保持	保持
	RUCLMIR	不定值	保持	保持
	RUCLMAR	不定值	保持	保持
	RCL2R	H'0000 4004	保持	保持
	POFSR	H'0000 0000	保持	保持
	GTRCR	H'0000 0000	保持	保持
	MTRAR	不定值	保持	保持
	MTRBR	不定值	保持	保持
	MTRCR	不定值	保持	保持
	MTRDR	不定值	保持	保持
MTRER	不定值	保持	保持	
MTRFR	不定值	保持	保持	
MTRGR	不定值	保持	保持	
MTRHR	不定值	保持	保持	

模块	寄存器略称	上电复位	睡眠	待机
G2D	MTRIR	不定值	保持	保持
	GTROFSXR	不定值	保持	保持
	GTROFSYR	不定值	保持	保持
	ZCLPMINR	不定值	保持	保持
	ZCLPMAXR	不定值	保持	保持
	ZSATVMINR	不定值	保持	保持
VDC2 图形部 1	GRCMEN1	H'0000 0000	保持	保持
	GRCBUSCNT1	H'0000 0000	保持	保持
	GROPSADR1	H'0000 0000	保持	保持
	GROPSWH1	H'0000 0000	保持	保持
	GROPSOFST1	H'0000 0000	保持	保持
	GRODPHV1	H'0000 0000	保持	保持
VDC2 图形部 2	GROPBASERGB1	H'0000 0000	保持	保持
	GRCMEN2	H'0000 0000	保持	保持
	GRCBUSCNT2	H'0000 0000	保持	保持
	GROPSADR2	H'0000 0000	保持	保持
	GROPSWH2	H'0000 0000	保持	保持
	GROPSOFST2	H'0000 0000	保持	保持
	GRODPHV2	H'0000 0000	保持	保持
	GROPEWH2	H'0000 0000	保持	保持
	GROPEDPHV2	H'0000 0000	保持	保持
	GROPEDPA2	H'0000 0000	保持	保持
	GROPCRKY0_2	H'0000 0000	保持	保持
	GROPCRKY1_2	H'0000 0000	保持	保持
GROPBASERGB2	H'0000 0000	保持	保持	
VDC2 图形部 3	GRCMEN3	H'0000 0000	保持	保持
	GRCBUSCNT3	H'0000 0000	保持	保持
	GROPSADR3	H'0000 0000	保持	保持
	GROPSWH3	H'0000 0000	保持	保持
	GROPSOFST3	H'0000 0000	保持	保持
	GRODPHV3	H'0000 0000	保持	保持
	GROPEWH3	H'0000 0000	保持	保持
	GROPEDPHV3	H'0000 0000	保持	保持
	GROPEDPA3	H'0000 0000	保持	保持
	GROPCRKY0_3	H'0000 0000	保持	保持
	GROPCRKY1_3	H'0000 0000	保持	保持
	GROPBASERGB3	H'0000 0000	保持	保持
VDC2 图形部 4	GRCMEN4	H'0000 0000	保持	保持
	GRCBUSCNT4	H'0000 0000	保持	保持
	GROPSADR4	H'0000 0000	保持	保持
	GROPSWH4	H'0000 0000	保持	保持

模块	寄存器略称	上电复位	睡眠	待机
VDC2 图形部 4	GROPSOFST4	H'0000 0000	保持	保持
	GRODPHV4	H'0000 0000	保持	保持
	GROPEWH4	H'0000 0000	保持	保持
	GROPEDPHV4	H'0000 0000	保持	保持
	GROPEDPA4	H'0000 0000	保持	保持
	GROPCRKY0_4	H'0000 0000	保持	保持
	GROPCRKY1_4	H'0000 0000	保持	保持
	GROPBASERGB4	H'0000 0000	保持	保持
VDC2 显示控制部	SGMODE	H'0000 0000	保持	保持
	SGINTCNT	H'0000 0000	保持	保持
	SYNCNT	H'0000 0000	保持	保持
	EXTSYNCNT	H'0000 0000	保持	保持
	SYNSIZE	H'020D 035A	保持	保持
	VSYNCTIM	H'0000 0001	保持	保持
	HSYNCTIM	H'0000 000A	保持	保持
	CLSTIM	H'0000 0000	保持	保持
	SPLTIM	H'0000 0000	保持	保持
	COMTIM	H'0000 0000	保持	保持
	SGDESTART	H'0000 0000	保持	保持
	SGDESIZE	H'0000 0000	保持	保持
	CDECRKY	H'0000 0000	保持	保持
	T1004CNT	H'0000 0000	保持	保持
T1004OFFSET	H'0000 0000	保持	保持	
FLCTL	FLCMNCR	初始化	保持	保持
	FLCMDCR	初始化	保持	保持
	FLCMCDR	初始化	保持	保持
	FLADR	初始化	保持	保持
	FLADR2	初始化	保持	保持
	FLDATAR	初始化	保持	保持
	FLDTCNTR	初始化	保持	保持
	FLINTDMACR	初始化	保持	保持
	FLBSYTMR	初始化	保持	保持
	FLBSYCNT	初始化	保持	保持
	FLDTFIFO	初始化	保持	保持
	FLECFIFO	初始化	保持	保持
	FLTRCR	初始化	保持	保持
SRC	SRCID	H'0000 0000	保持	保持
	SRCOD	H'0000 0000	保持	保持
	SRCIDCTRL	H'0000	保持	保持
	SRCODCTRL	H'0000	保持	保持
	SRCCTRL	H'0000	保持	保持

模块	寄存器略称	上电复位	睡眠	待机
SRC	SRCSTAT	H'0002	保持	保持
GPIO	PTIO_A	H'0000	保持	保持
	PTIO_B	H'0000	保持	保持
	PTIO_C	H'0000	保持	保持
	PTIO_D	H'0000	保持	保持
	PTIO_E	H'0000	保持	保持
	PTIO_F	H'0000	保持	保持
	PTIO_G	H'0000	保持	保持
	PTIO_H	H'0000	保持	保持
	PTIO_I	H'0002	保持	保持
	PTIO_J	H'AAAA	保持	保持
	PTDAT_A	H'0000	保持	保持
	PTDAT_B	H'0000	保持	保持
	PTDAT_C	H'0000	保持	保持
	PTDAT_D	H'0000	保持	保持
	PTDAT_E	H'0000	保持	保持
	PTDAT_F	H'0000	保持	保持
	PTDAT_G	H'0000	保持	保持
	PTDAT_H	H'0000	保持	保持
	PTDAT_I	H'0000	保持	保持
	PTDAT_J	H'0000	保持	保持
	PTPUL_SPCL	H'0000	保持	保持
	PTSEL_A	H'0000	保持	保持
	PTSEL_B	H'0000	保持	保持
	PTSEL_C	H'0000	保持	保持
	PTSEL_D	H'0000	保持	保持
	PTSEL_E	H'0000	保持	保持
	PTSEL_F	H'0000	保持	保持
	PTSEL_G	H'0000	保持	保持
	PTSEL_H	H'0000	保持	保持
	PTSEL_I	H'0000	保持	保持
	PTSEL_J	H'0000	保持	保持
	PTSEL_K	H'0000	保持	保持
PTSEL_P	H'0000	保持	保持	
PTSEL_R	H'0000	保持	保持	
PTSEL_S	H'0000	保持	保持	
PTHIZ_A	H'0000	保持	保持	
PTHIZ_B	H'0000	保持	保持	
低功耗模式 *2	STBCR	H'0000 0000	保持	保持
	MSTPCR0	H'0000 0000	保持	保持
	MSTPCR1	H'0000 0000	保持	保持

模块	寄存器略称	上电复位	睡眠	待机
UBC	CBR0	H'2000 0000	保持	保持
	CRR0	H'0000 2000	保持	保持
	CAR0	不定值	保持	保持
	CAMR0	不定值	保持	保持
	CBR1	H'2000 0000	保持	保持
	CRR1	H'0000 2000	保持	保持
	CAR1	不定值	保持	保持
	CAMR1	不定值	保持	保持
	CDR1	不定值	保持	保持
	CDMR1	不定值	保持	保持
	CETR1	不定值	保持	保持
	CCMFR	H'0000 0000	保持	保持
	CBCR	H'0000 0000	保持	保持
H-UDI	SDIR	H'0EFF	保持	保持
	SDINT	H'0000	保持	保持

【符号说明】

○：被初始化的寄存器

—：被保持的寄存器

【注】 *1 是通过时钟运行模式（由 MODE0、MODE1 和 MODE2 设定）设定的初始值。

*2 有关待机控制寄存器，请参照图 10.1（图 10.1：CPG 框图）。

模块	寄存器略称	上电复位		睡眠	待机
		通过 PRESET 引脚	通过 WDT/H-UDI		
WDT	WDTST	H'0000 0000	保持	保持	保持
	WDTCSR	H'0000 0000	保持	保持	保持
	WDTBST	H'0000 0000	保持	保持	保持
	WDTCNT	H'0000 0000	保持	保持	保持
	WDTBCNT	H'0000 0000	保持	保持	保持

第 33 章 电特性

33.1 绝对最大额定值 *1*2

表 33.1 绝对最大额定值

项目	符号	额定值	单位
电源电压 (IO)	V_{DDQ} 、 V_{DDQ_USB}	-0.3 ~ 4.6	V
电源电压 (内部)	V_{DD} 、 V_{DD_PLL1} 、 V_{DD_PLL2} 、 V_{DD_USB} 、 $UV12$	-0.3 ~ 1.7	V
电源电压 (模拟 3.3V 系列)	V_{DDQA_USB}	-0.3 ~ 4.6	V
电源电压 (模拟 1.2V 系列)	V_{DDA_USB}	-0.3 ~ 1.7	V
输入电压	V_{in}	-0.3 ~ $V_{DDQ}+0.3$ *3	V
Vbus 输入电压	V_{bus}	-0.3 ~ 5.5	V
工作环境温度	T_{opr}	-20 ~ 85 (普通规格产品)	°C
		-40 ~ 85 (大温度范围规格产品)	
保存温度	T_{stg}	-55 ~ 125	°C

【注】 *1 如果使用 LSI 时超过最大额定值，就会造成 LSI 的永久性破坏。

*2 如果全部的 VSS 不连接 GND，就会造成 LSI 的永久性破坏。

*3 上限不能超过电源电压。

33.2 接通和切断电源的顺序

接通和切断电源的顺序及其推荐值如下所示：

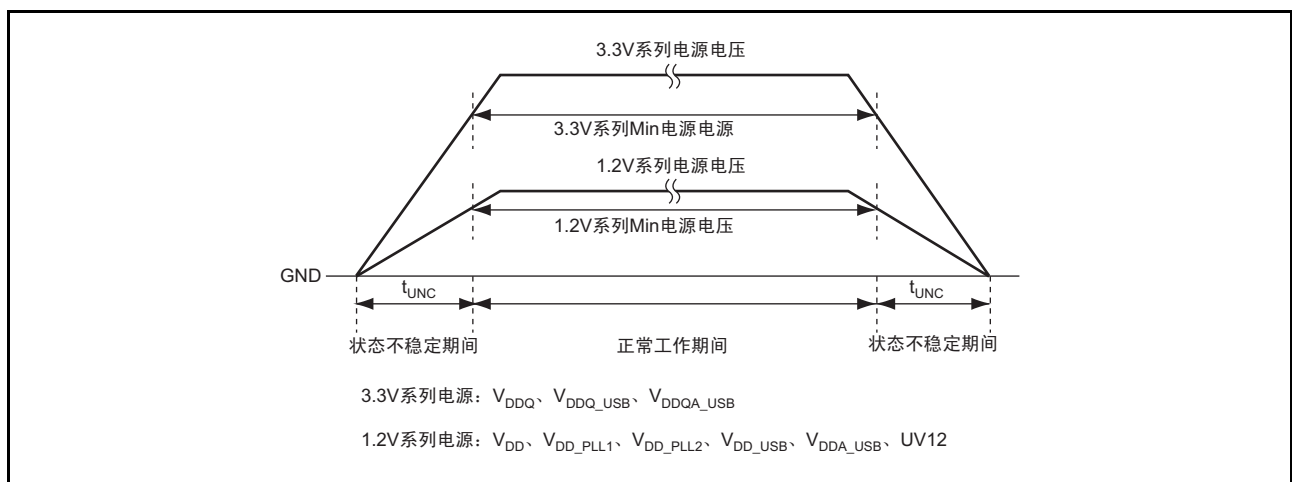


图 33.1 接通和切断电源的顺序

表 33.2 接通和切断电源的容许时间

项目	符号	最大容许值	单位
状态的不稳定时间	t_{unc}	100	ms

【注】 上述是最大容许值，而不是要求严密设定的值。

3.3V 系列电源和 1.2V 系列电源的接通和切断顺序无论哪个在先都没有问题，但是在后接通的电压达到 Min. 电压前或者先切断的电压低于 Min. 电压后，都进入状态不稳定期间。在此期间，引脚状态和内部状态都为不稳定状态。在设计系统时，必须注意：不能因该状态引起系统误动作。

33.3 DC 特性

表 33.3 DC 特性 [共用项目]

$T_a = -20 \sim 85^\circ\text{C}$ 、 $-40 \sim 85^\circ\text{C}$

项目		符号	Min.	Typ.	Max.	单位	测量条件
电源电压 (IO)		V_{DDQ} 、 V_{DDQ_USB}	3.0	3.3	3.6	V	
电源电压 (内部)		V_{DD} 、 V_{DD_PLL1} 、 V_{DD_PLL2} 、 V_{DD_USB} 、 $UV12$	1.15	1.25	1.35	V	
电源电压 (模拟 3.3V 系列)		V_{DDQA_USB}	3.0	3.3	3.6	V	
电源电压 (模拟 1.2V 系列)		V_{DDA_USB}	1.15	1.25	1.35	V	
消耗电流	正常运行时	$I_{DDQ} + I_{DDQ_USB}$	—	—	150	mA	I _{ck} =324MHz SH _{ck} =B _{ck} =108MHz P _{ck} =54MHz
		$I_{DD} + I_{DD_PLL1} + I_{DD_PLL2}$ $+ I_{DD_USB} + UV12$	—	—	850	mA	
	睡眠模式时	$I_{DDQ} + I_{DDQ_USB}$	—	—	150	mA	
		$I_{DD} + I_{DD_PLL1} + I_{DD_PLL2}$ $+ I_{DD_USB} + UV12$	—	—	650	mA	
	刷新待机时	$I_{DDQ} + I_{DDQ_USB}$	—	—	15	mA	
		$I_{DD} + I_{DD_PLL1} +$ $I_{DD_PLL2} + I_{DD_USB} + UV12$	—	—	450	mA	
消耗电流 (USB)	运行时	I_{DDQA_USB}	—	—	15	mA	
		I_{DDA_USB}	—	—	20	mA	
	刷新待机时	I_{DDQA_USB}	—	—	600	μA	
		I_{DDA_USB}	—	—	600	μA	

表 33.4 DC 特性 [USB 收发器和 I²C 的相关引脚除外]

Ta=-20 ~ 85°C、-40 ~ 85°C

项目	符号	Min.	Typ.	Max.	单位	测量条件	
输入电压	PRESET、TRST、NMI、 MODE8/FD7、MODE7/FD6、MODE5/FD5、 MEDE4/FD4、MODE3/FD3、MODE2/FD2、 MODE1/FD1、MODE0/FD0、 IRQ0/DTEND1、 WDTOVF/IRQ1/AUDCK/DACK1、 D44/IDEINT、 MII_TXD2/AUDIO_CLK5/IDEINT_M/PD1、 STATUS1/RTS2/PA7、 STATUS0/CTS2/PA6、FCE/PA5、 FRE/PA4、FWE/PA3、TXD2/PA2、 RXD2/PA1、SCK2/PA0、 A25/PB7/DREQ0/RTS0、 A24/PB6/DACK0/CTS0、 A23/PB5/DTEND0/RTS1、A22/PB4/CTS1、 A21/PB3、A20/PB2、A19/PB1、A18/PB0、 ASEBRKAK/BRKACK/TCLK/PC1 EXTAL、XIN	V _{IH}	V _{CCQ} ×0.9	—	V _{CCQ} +0.3	V	V _{CCQ} =3.0 ~ 3.6V
	其他输入引脚		2	—	V _{CCQ} +0.3		
输入电压	PRESET、TRST、NMI、 MODE8/FD7、MODE7/FD6、MODE5/FD5、 MEDE4/FD4、MODE3/FD3、MODE2/FD2、 MODE1/FD1、MODE0/FD0、 IRQ0/DTEND1、 WDTOVF/IRQ1/AUDCK/DACK1、 D44/IDEINT、 MII_TXD2/AUDIO_CLK5/IDEINT_M/PD1、 STATUS1/RTS2/PA7、 STATUS0/CTS2/PA6、FCE/PA5、 FRE/PA4、FWE/PA3、TXD2/PA2、 RXD2/PA1、SCK2/PA0、 A25/PB7/DREQ0/RTS0、 A24/PB6/DACK0/CTS0、 A23/PB5/DTEND0/RTS1、A22/PB4/CTS1、 A21/PB3、A20/PB2、A19/PB1、A18/PB0、 ASEBRKAK/BRKACK/TCLK/PC1 EXTAL、XIN	V _{IL}	-0.3	—	V _{CCQ} ×0.1	V	V _{CCQ} =3.0 ~ 3.6V
	其他输入引脚	V _{IL}	-0.3	—	V _{CCQ} ×0.2	V	
输入漏泄电流	全部的输入引脚	I _{in}	—	—	1	μA	V _{IN} =0.5 ~ V _{CCQ} -0.5V
三态漏泄电流	输入 / 输出、全部的输出引脚 (OFF 状态)	I _{sti}	—	—	1		
输出电压	全部的输出引脚	V _{OH}	2.4	—	—	V	V _{CCQ} =3.0V I _{OH} =-2mA
	全部的输出引脚	V _{OL}	—	—	0.55		V _{DDQ} =3.0V I _{OL} =2mA

项目		符号	Min.	Typ.	Max.	单位	测量条件
上拉电阻	全部的引脚	R_{pull}	20	60	180	k Ω	
引脚电容	其他	C_L	—	—	10	pF	

【注】 消耗电流值是在 V_{IH} (Min.) = $V_{CCQ} - 0.5V$ 和 V_{IL} (Max.) = $0.5V$ 并且全部的输出引脚没有负载状态下的值。

表 33.5 DC 特性 [I²C 的相关引脚]

条件: 3.3V 系列电源 = 3.0 ~ 3.6V、1.2V 系列电源 = 1.15 ~ 1.35V、 $T_a = -20 \sim 85^\circ C$ 、 $-40 \sim 85^\circ C$

项目	符号	Min.	Typ.	Max.	单位	测量条件
电源电压	V_{CCQ}	3.0	3.3	3.6	V	
输入高电平电压	V_{IH}	$V_{CCQ} \times 0.7$	—	$V_{CCQ} + 0.3$	V	
输入低电平电压	V_{IL}	-0.3	—	$V_{CCQ} \times 0.3$	V	
输出低电平电压	V_{OL}	—	—	0.4	V	$I_{OL} = 3mA$ 时
输出低电平容许电流	I_{OL}	—	—	10	mA	

【注】 I²C 的相关引脚: SCL、SDA (漏极开路引脚)

表 33.6 DC 特性 [USB 的相关引脚 (1)]

条件: 3.3V 系列电源 = 3.0 ~ 3.6V、1.2V 系列电源 = 1.15 ~ 1.35V、 $T_a = -20 \sim 85^\circ C$ 、 $-40 \sim 85^\circ C$

项目	符号	Min.	Typ.	Max.	单位	测量条件
基准电阻	R_{REF}	5.6k $\Omega \pm 1\%$			V	
输入高电平电压 (VBUS)	V_{IH}	4.02	—	5.25	V	
输入低电平电压 (VBUS)	V_{IL}	0.0	—	1.0	V	
输入高电平电压 (XIN)	V_{IH}	$V_{DDQ} - 0.5$	—	$V_{DDQ} + 0.3$	V	
输入低电平电压 (XIN)	V_{IL}	-0.3	—	0.5	V	

表 33.6 特性 [USB 的相关引脚 (2) 全速 / 高速共用项目]

条件: 3.3V 系列电源 = 3.0 ~ 3.6V、1.2V 系列电源 = 1.15 ~ 1.35V、 $T_a = -20 \sim 85^\circ C$ 、 $-40 \sim 85^\circ C$

项目	符号	Min.	Typ.	Max.	单位	测量条件
DP 上拉电阻 (选择功能控制器功能时)	R_{pu}	0.900	—	1.575	k Ω	空闲时
		1.425	—	3.090	k Ω	发送 / 接收时
DP、DM 上拉电阻 (选择主机控制器功能时)	R_{pd}	14.25	—	24.80	k Ω	

表 33.6 DC 特性 [USB 的相关引脚 (3) 全速项目]

条件: 3.3V 系列电源 =3.0 ~ 3.6V、1.2V 系列电源 =1.15 ~ 1.35V、 $T_a = -20 \sim 85^\circ\text{C}$ 、 $-40 \sim 85^\circ\text{C}$

项目	符号	Min.	Typ.	Max.	单位	测量条件
输入高电平电压	V_{IH}	2.0	—	—	V	
输入低电平电压	V_{IL}	—	—	0.8	V	
差动输入灵敏度	V_{DI}	0.2	—	—	V	$ (DP)-(DM) $
差动共模范围	V_{CM}	0.8	—	2.5	V	
输出高电平电压	V_{OH}	2.8	—	3.6	V	$I_{OH}=5\text{mA}$
输出低电平电压	V_{OL}	0.0	—	0.3	V	$I_{OL}=5\text{mA}$
单端接收器阈值电压	V_{SE}	0.8	—	2.0	V	
交叉电压	V_{CRS}	1.3	—	2.0	V	$C_L=50\text{pF}$

【注】 USB 的相关引脚: DP、DM

表 33.6 DC 特性 [USB 的相关引脚 (4) 高速项目]

条件: 3.3V 系列电源 =3.0 ~ 3.6V、1.2V 系列电源 =1.15 ~ 1.35V、 $T_a = -20 \sim 85^\circ\text{C}$ 、 $-40 \sim 85^\circ\text{C}$

项目	符号	Min.	Typ.	Max.	单位	测量条件
差动输入灵敏度	V_{HSDI}	0.15	—	—	V	
静噪检测阈值电压 (差动电压)	V_{HSSQ}	100	—	150	mV	
共模电压范围	V_{HSCM}	-50	—	500	mV	
空闲状态	V_{HSOI}	-10.0	—	10.0	mV	
输出高电平电压	V_{HSOH}	360	—	440	mV	
输出低电平电压	V_{HSOL}	-10.0	—	10.0	mV	
Chirp J 输出电压 (差分)	V_{CHIRPJ}	700	—	1000	mV	
Chirp K 输出电压 (差分)	V_{CHIRPK}	-900	—	-500	mV	

【注】 USB 的相关引脚: DP、DM

表 33.7 DC 特性 [USB 的相关引脚 (5) 低速项目]

条件: 3.3V 系列电源 =3.0 ~ 3.6V、1.2V 系列电源 =1.15 ~ 1.35V、 $T_a = -20 \sim 85^\circ\text{C}$ 、 $-40 \sim 85^\circ\text{C}$

项目	符号	Min.	Typ.	Max.	单位	测量条件
输出高电平电压	V_{LSOH}	2.8	—	—	V	$I_{OH}=200\mu\text{A}$
输出低电平电压	V_{LSOL}	—	—	0.3	V	$I_{OL}=2\text{mA}$

【注】 USB 的相关引脚: DP、DM

表 33.8 输出容许电流值

条件: 3.3V 系列电源 =3.0 ~ 3.6V、1.2V 系列电源 =1.15 ~ 1.35V、 $T_a = -20 \sim 85^\circ\text{C}$ 、 $-40 \sim 85^\circ\text{C}$

项目	符号	Min.	Typ.	Max.	单位
输出低电平容许电流	I_{OL}	—	—	2	mA
输出低电平容许电流 (总和)	ΣI_{OL}	—	—	120	
输出高电平容许电流	$-I_{OL}$	—	—	2	mA
输出高电平容许电流 (总和)	$\Sigma -I_{OH} $	—	—	40	

【注】 为了确保 LSI 的可靠性, 输出的电流值不能超过表 33.8 的值。

33.4 AC 特性

本 LSI 的输入原则上是时钟同步输入。只要没有特殊要求，就必须保持各输入信号的准备时间和保持时间。

33.4.1 时钟 / 控制信号时序

表 33.9 时钟 / 控制信号时序

条件：3.3V 系列电源 = 3.0 ~ 3.6V、1.2V 系列电源 = 1.15 ~ 1.35V、 $T_a = -20 \sim 85^\circ\text{C}$ 、 $-40 \sim 85^\circ\text{C}$

项目	符号	Min.	Max.	单位	参照图
EXTAL 时钟输入频率 *1	f_{EX}	24	32.4	MHz	
EXTAL 时钟输入的周期时间	t_{EXcyc}	30.8	42	ns	33.2
EXTAL 时钟输入低电平脉宽	t_{EXL}	7	—	ns	33.2
EXTAL 时钟输入高电平脉宽	t_{EXH}	7	—	ns	33.2
EXTAL 时钟输入的上升时间	t_{EXr}	—	4	ns	33.2
EXTAL 时钟输入的下降时间	t_{EXf}	—	4	ns	33.2
CLKOUT 时钟输出 *2	t_{OP}	80	108	MHz	
CLKOUT 时钟输出的周期时间	$t_{CLKOUTcyc}$	9.26	12.5	ns	33.3
CLKOUT 时钟输出低电平脉宽	$t_{CLKOUTL}$	2	—	ns	33.3
CLKOUT 时钟输出高电平脉宽	$t_{CLKOUTH}$	2	—	ns	33.3
CLKOUT 时钟输出的上升时间	$t_{CLKOUTr}$	—	3	ns	33.3
CLKOUT 时钟输出的下降时间	$t_{CLKOUTf}$	—	3	ns	33.3
MDn 复位准备时间	t_{MDRS}	30	—	ms	33.5
MDn 复位保持时间	t_{MDRH}	20	—	ns	33.5
PRESET 有效时间	t_{RESPW}	30	—	ms	33.4、33.5
上电振荡稳定时间	t_{OSC}	60	—	μs	33.4
TRST 复位保持时间	t_{TRSTRH}	20	—	ns	33.4

【注】 *1. 在晶体振荡器连接 EXTAL 和 XTAL 时，最大频率为 32.4MHz。另外，在使用 3 维谐波晶体振荡器时，需要外接谐波电路。

*2. CLKOUT 引脚的最大连接负载电容为 50pF。

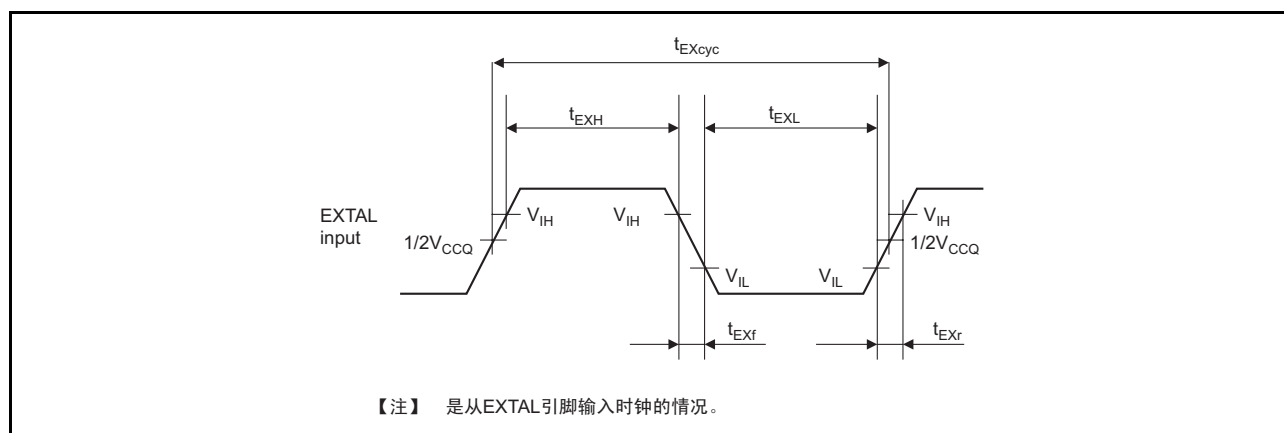


图 33.2 EXTAL 时钟的输入时序

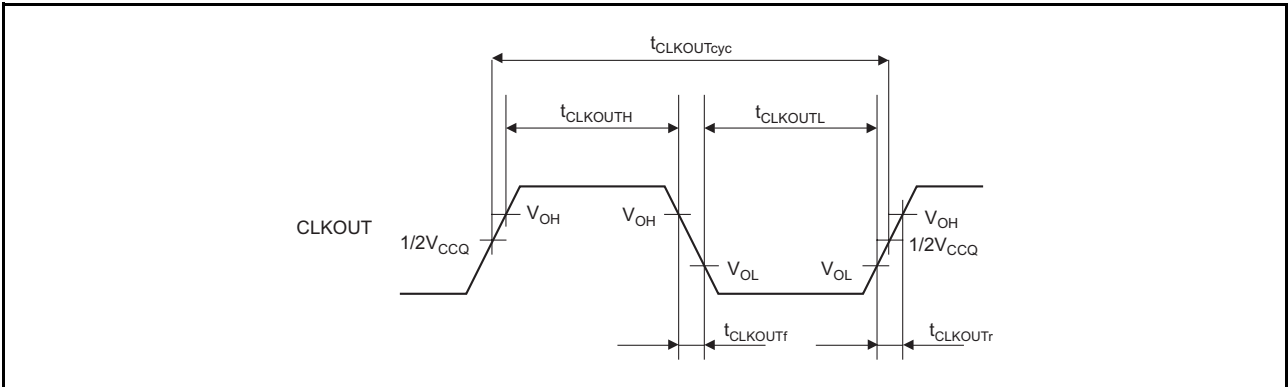


图 33.3 CLKOUT 时钟的输出时序 (1)

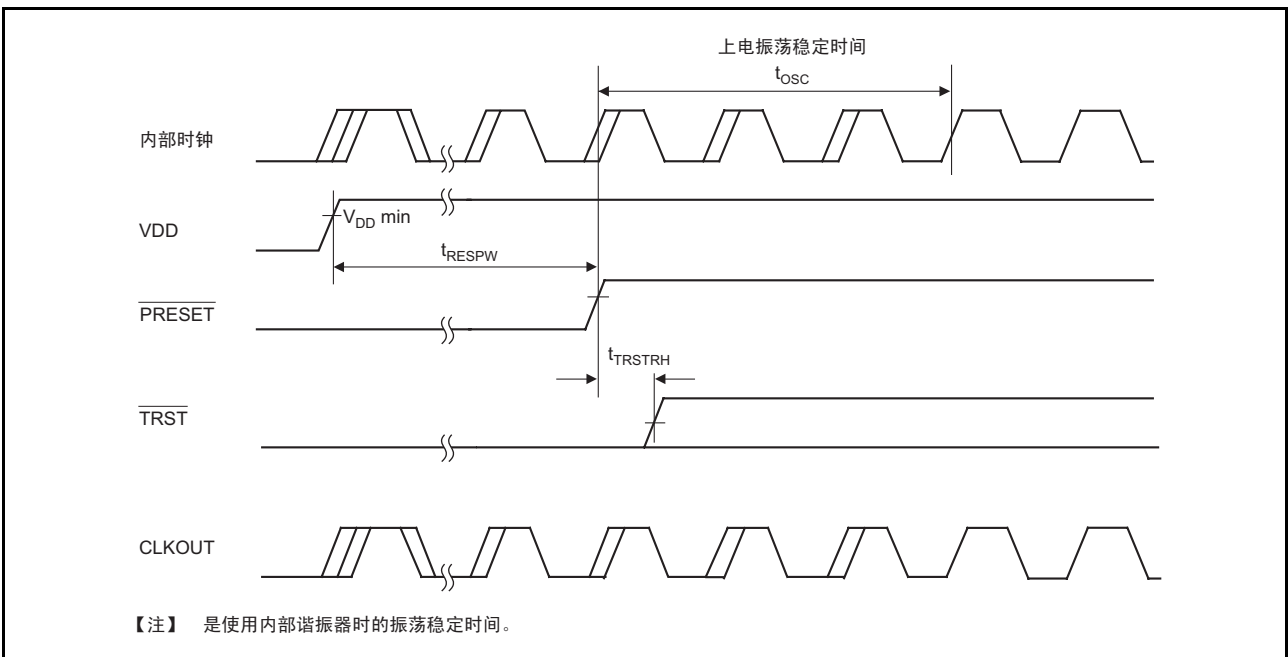


图 33.4 上电时的振荡稳定时间

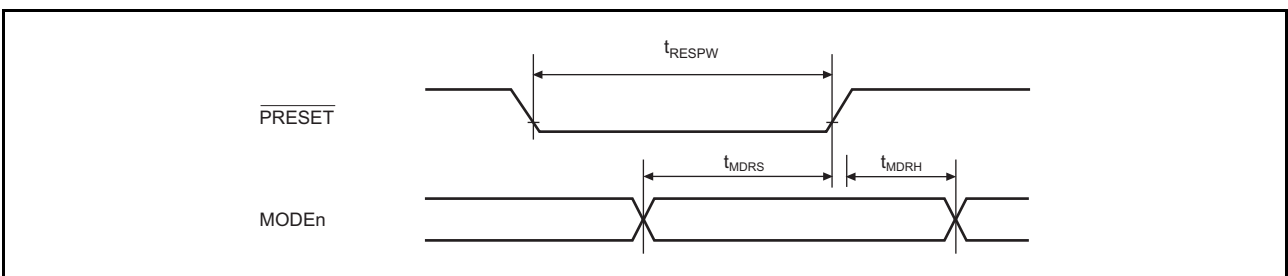


图 33.5 MODE 引脚的准备时间和保持时序

33.4.2 控制信号时序

表 33.10 控制信号时序

条件：3.3V 系列电源 = 3.0 ~ 3.6V、1.2V 系列电源 = 1.15 ~ 1.35V、 $T_a = -20 \sim 85^\circ\text{C}$ 、 $-40 \sim 85^\circ\text{C}$

项目	符号	Min.	Max.	单位	参照图
WDTOVF 延迟时间	t_{WOVD}	—	$t_{\text{cyc}}+9$	ns	33.6
STATUS0 和 STATUS1 延迟时间	t_{STD}	—	$t_{\text{cyc}}+10$	ns	33.6
BREQ 准备时间	t_{BREQS}	3	—	ns	33.7
BREQ 保持时间	t_{BREQH}	1.5	—	ns	33.7
BACK 延迟时间	t_{BACKD}	—	7	ns	33.7
总线三态延迟时间	t_{BOFF}	—	10	ns	33.7
总线缓冲器的接通时间	t_{BON}	—	10	ns	33.7

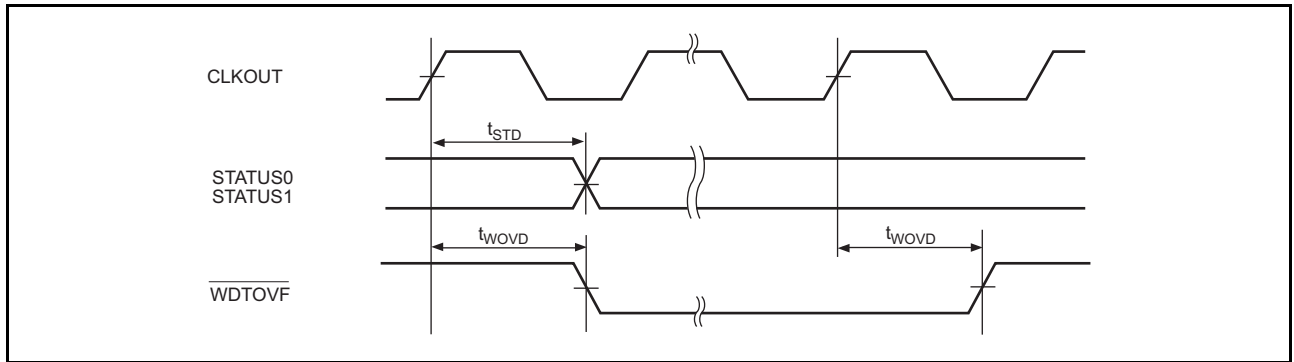
【注】 t_{cyc} 表示 1 个 CLKOUT 时钟周期的时间。

图 33.6 待机时的引脚驱动时序

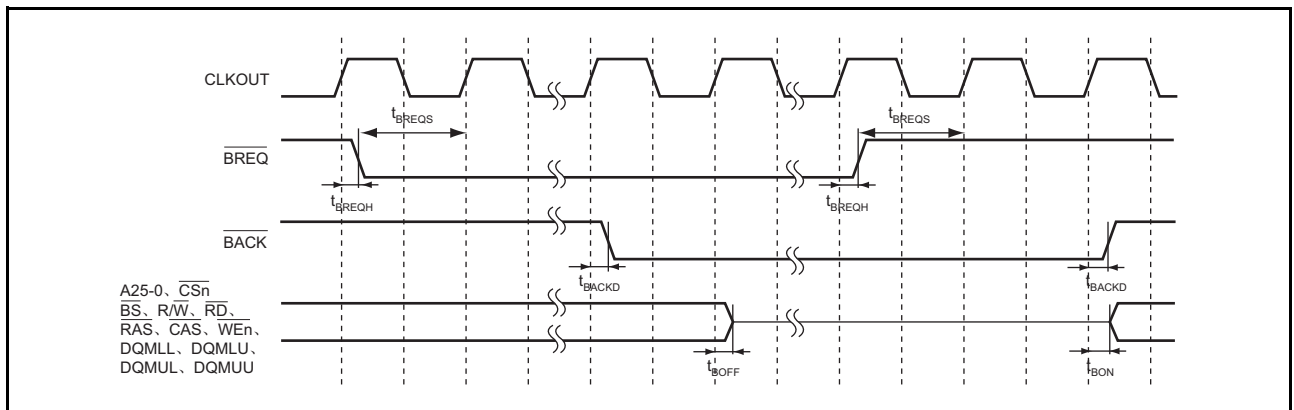


图 33.7 控制信号时序

33.4.3 总线时序

表 33.11 总线时序

条件：3.3V 系列电源 =3.0 ~ 3.6V、1.2V 系列电源 =1.15 ~ 1.35V、 $T_a = -20 \sim 85^\circ\text{C}$ 、 $-40 \sim 85^\circ\text{C}$

项目	符号	Min.	Max.	单位	备注	参照图
地址延迟时间	t_{AD}	1.0	7.0	ns		33.8 ~ 33.11
\overline{BS} 延迟时间	t_{BSD}	1.0	7.0	ns		33.8 ~ 33.11
\overline{CSn} 延迟时间	t_{CSD}	1.0	7.0	ns		33.8 ~ 33.11
$\overline{R/W}$ 延迟时间	t_{RWD}	1.0	7.0	ns		33.8 ~ 33.11
\overline{RD} 延迟时间	t_{RSD}	1.0	7.0	ns		33.8 ~ 33.11
读数据的准备时间	t_{RDS}	3.0	—	ns		33.8 ~ 33.11
读数据的保持时间	t_{RDH}	1.5	—	ns		33.8 ~ 33.11
\overline{WEn} 延迟时间 (下降时) *	t_{WEDF}	—	7.0	ns		33.8 ~ 33.11
\overline{WEn} 延迟时间	t_{WED1}	1.0	7.0	ns		33.8 ~ 33.11
写数据的延迟时间	t_{WDD}	1.0	7.0	ns		33.8 ~ 33.11
\overline{RDY} 准备时间	t_{RDYS}	3.0	—	ns		33.9、33.10
\overline{RDY} 保持时间	t_{RDYH}	1.5	—	ns		33.9、33.10
\overline{RAS} 延迟时间	t_{RASD}	1.0	7.0	ns		33.12 ~ 33.23
\overline{CAS} 延迟时间	t_{CASD}	1.0	7.0	ns		33.12 ~ 33.23
\overline{CKE} 延迟时间	t_{CKED}	1.0	7.0	ns		33.12 ~ 33.23
\overline{DQM} 延迟时间	t_{DQMD}	1.0	7.0	ns		33.12 ~ 33.23

【注】 * 表示对于 CLKOUT 下降沿的延迟时间。

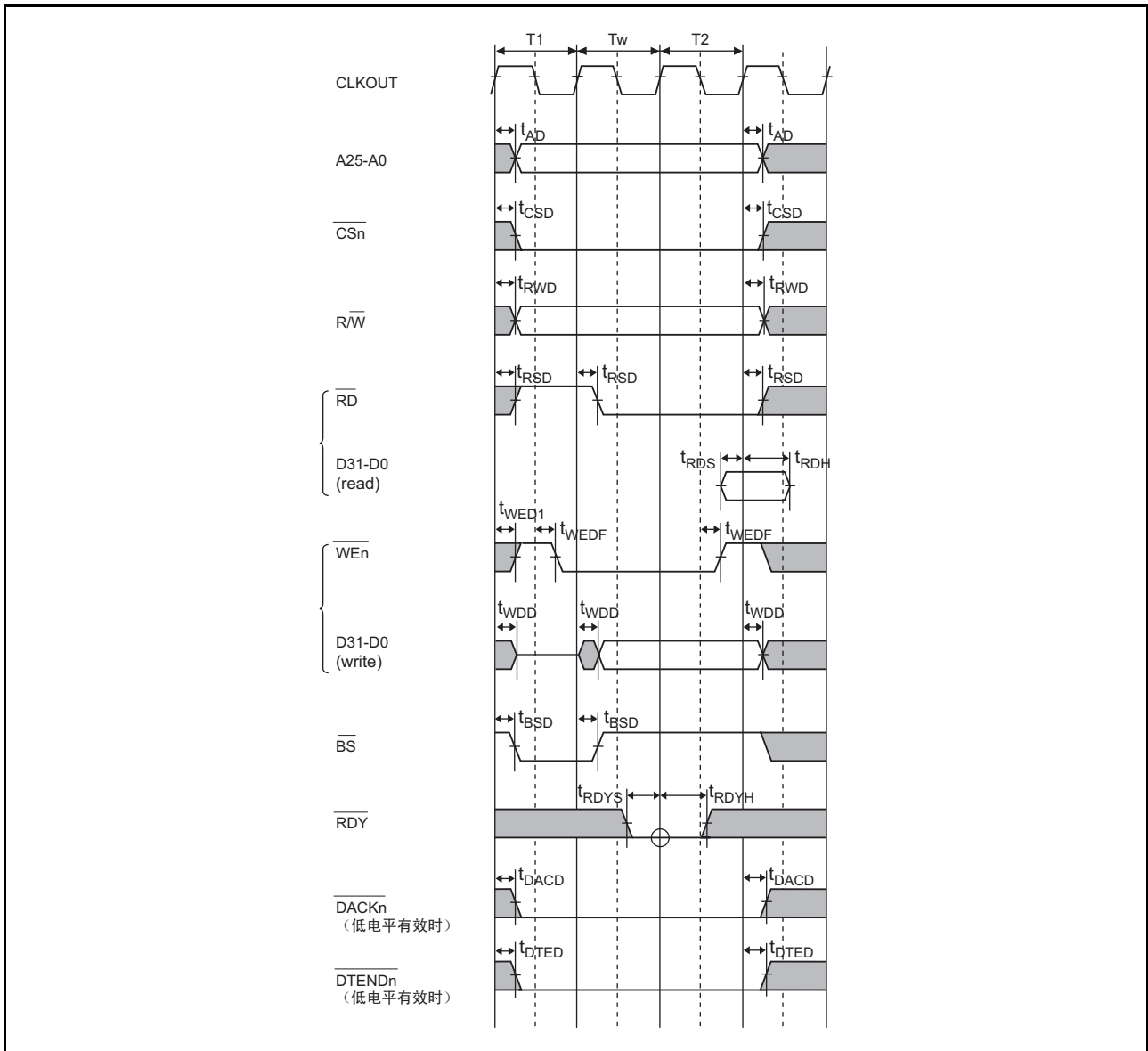


图 33.9 SRAM 总线周期 基本总线周期 (内部 1 个等待)

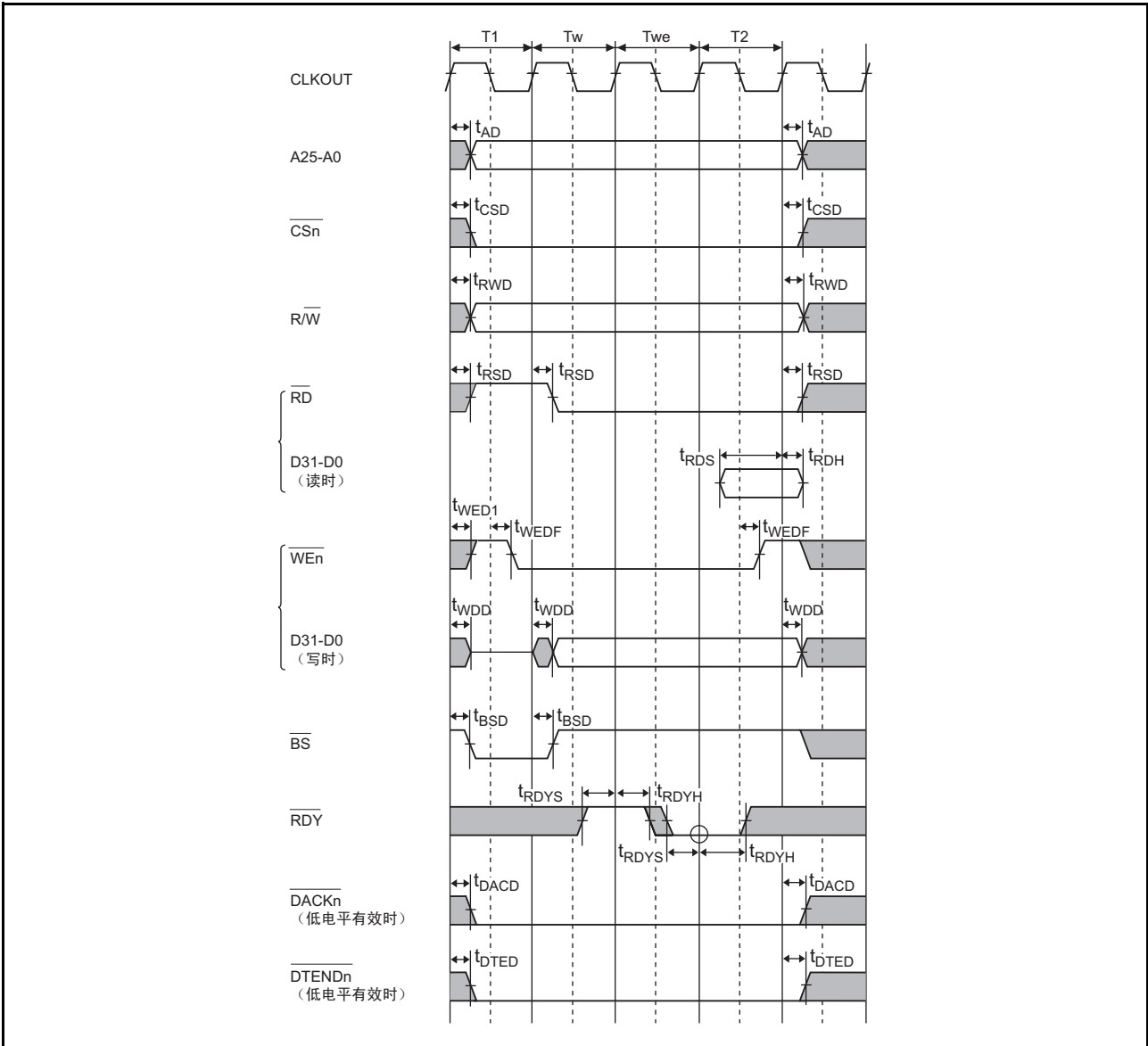


图 33.10 SRAM 总线周期 基本总线周期
(内部等待 + 外部 1 个等待)

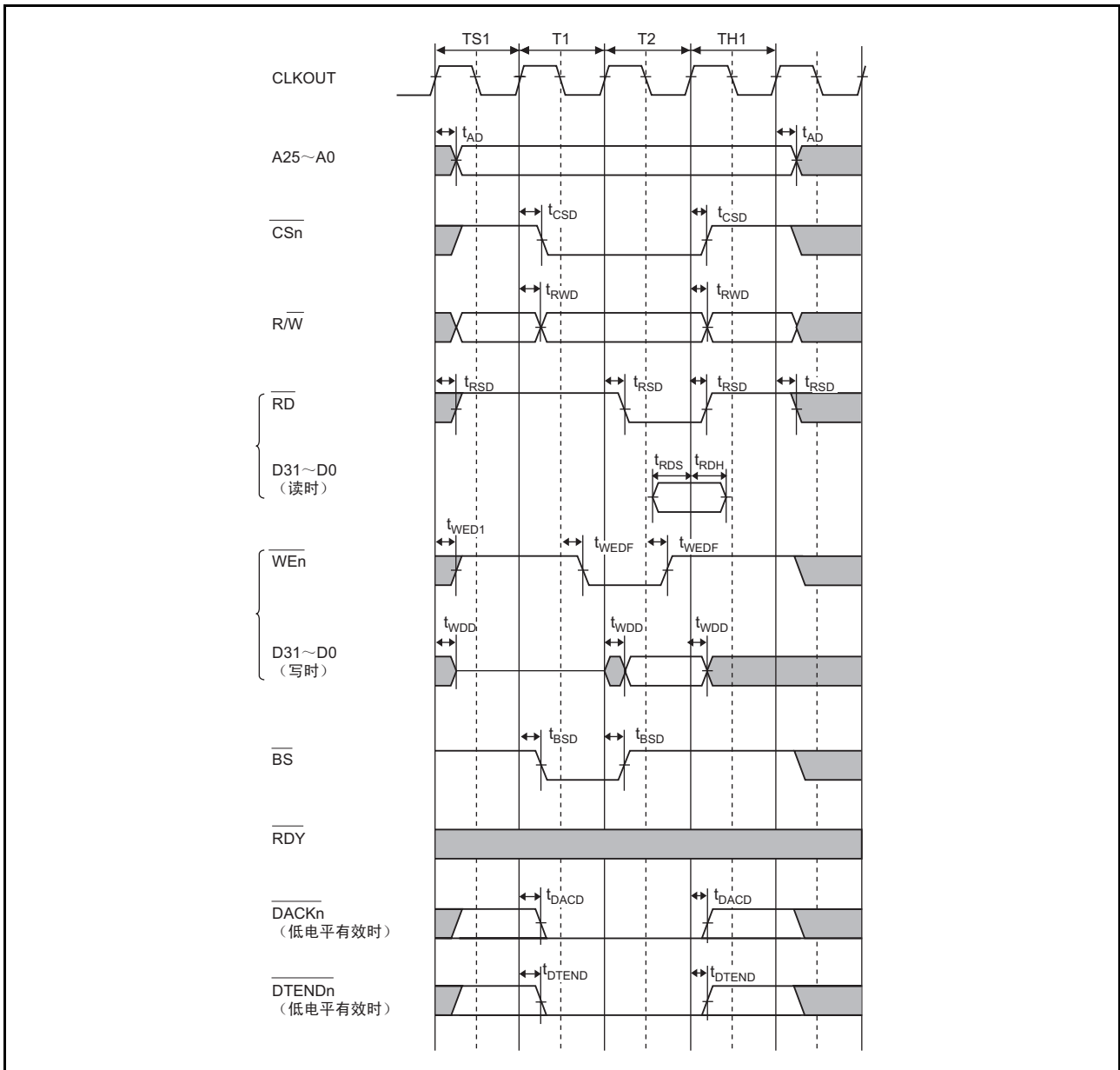


图 33.11 SRAM 总线周期 基本总线周期
(无等待, 插入地址准备时间 / 保持时间, AnS=1, AnH=1)

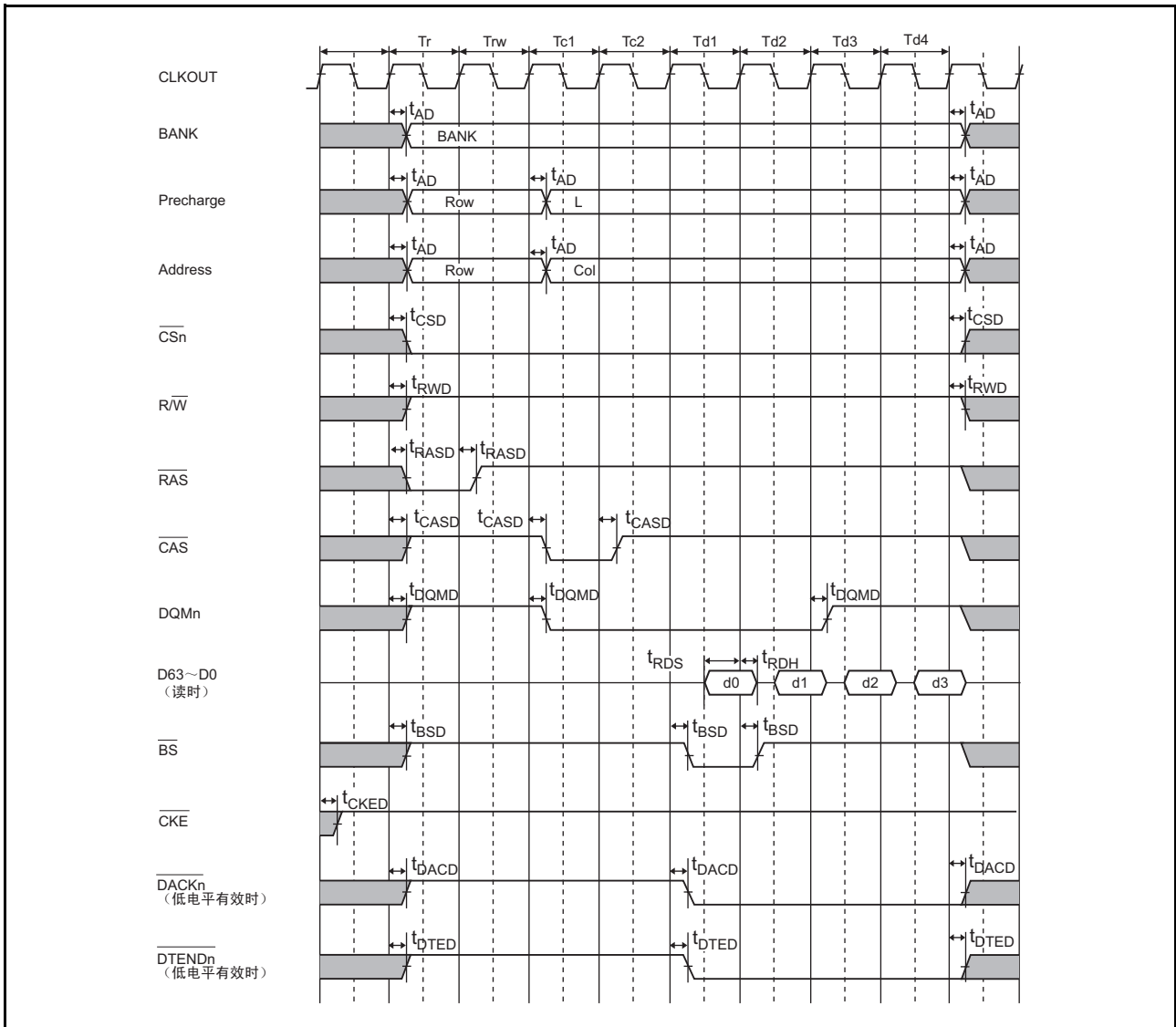


图 33.12 SDRAM 总线周期 存储体打开模式 读总线周期 (ACT-READ)
 (BOMODE[1:0]=00、SCL[2:0]=000、SRCD=0、CAS 等待时间 =2cyc、IRCD=2cyc)

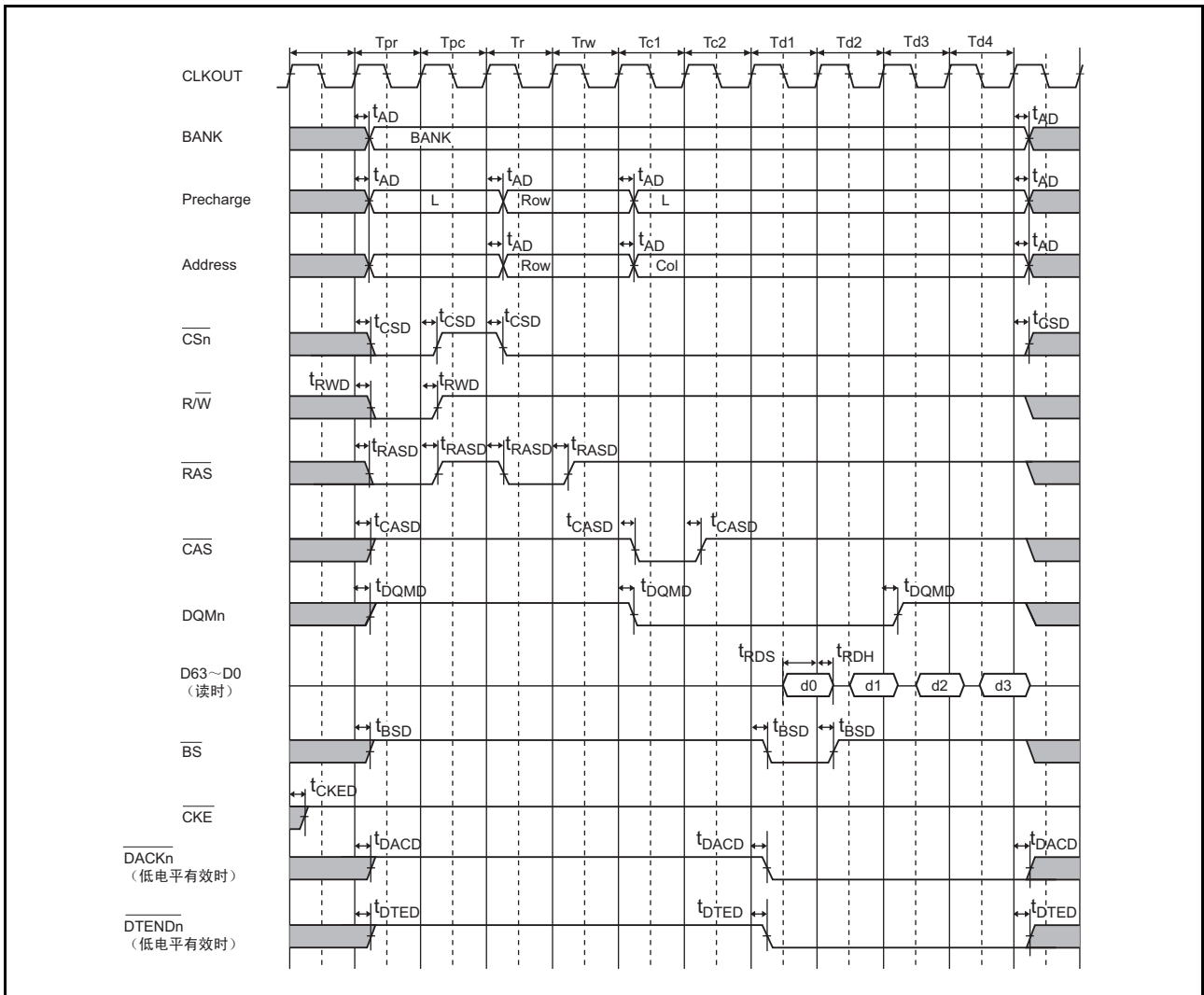


图 33.13 SDRAM 总线周期 存储体打开模式 预充电读总线周期
 (PRE-ACT-READ) (BOMODE[1:0]=00、SRP[1:0]=00、SCL[2:0]=000、SRCD=0、IRP=2cyc、CAS 等待
 时间=2cyc、IRCD=2cyc)

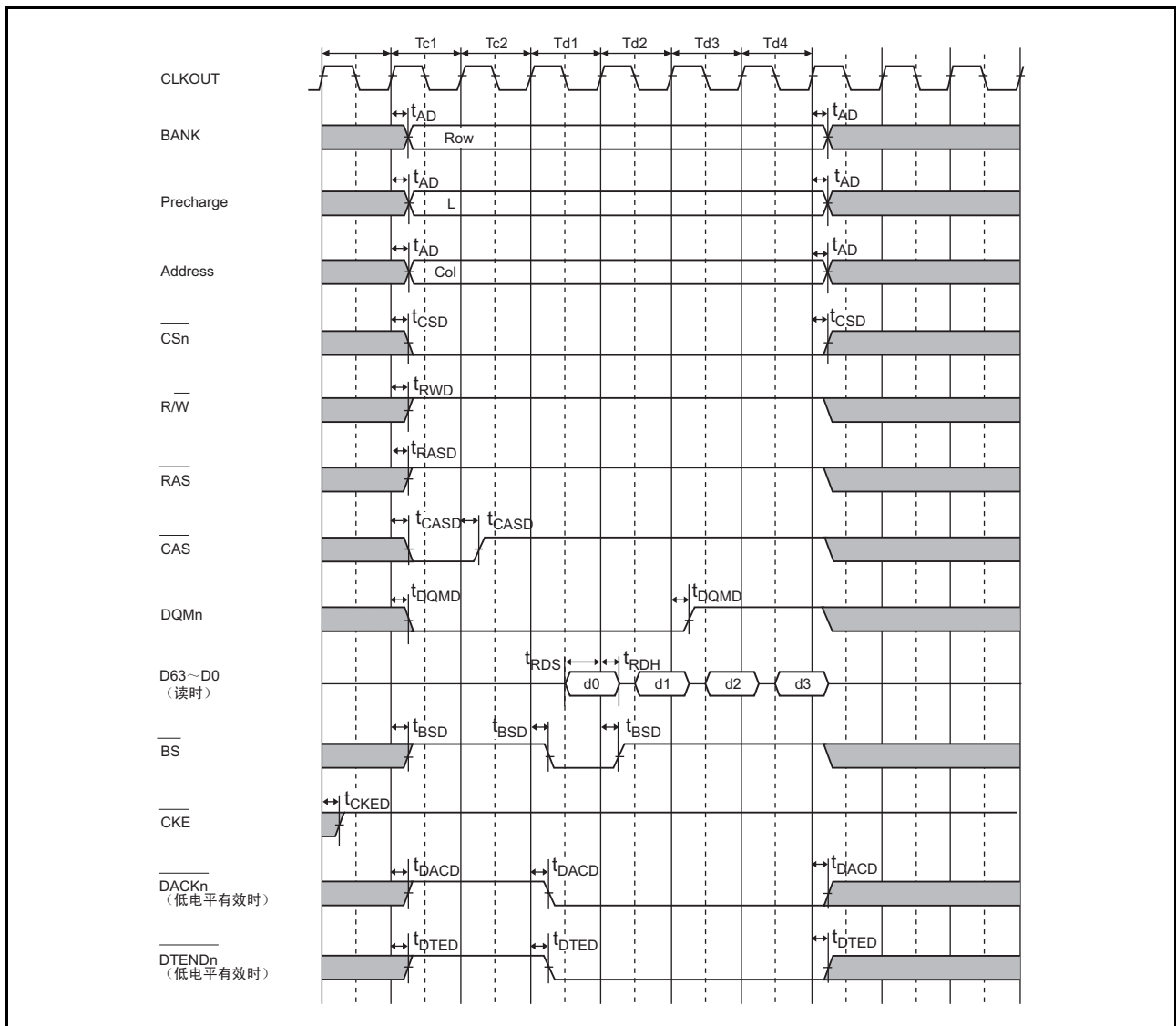


图 33.14 SDRAM 总线周期 存储体打开模式 读总线周期 (READ)
 (BOMODE[1:0]=00、SCL[2:0]=000、SRCD=0、CAS 等待时间 =2cyc、IRCD=2cyc)

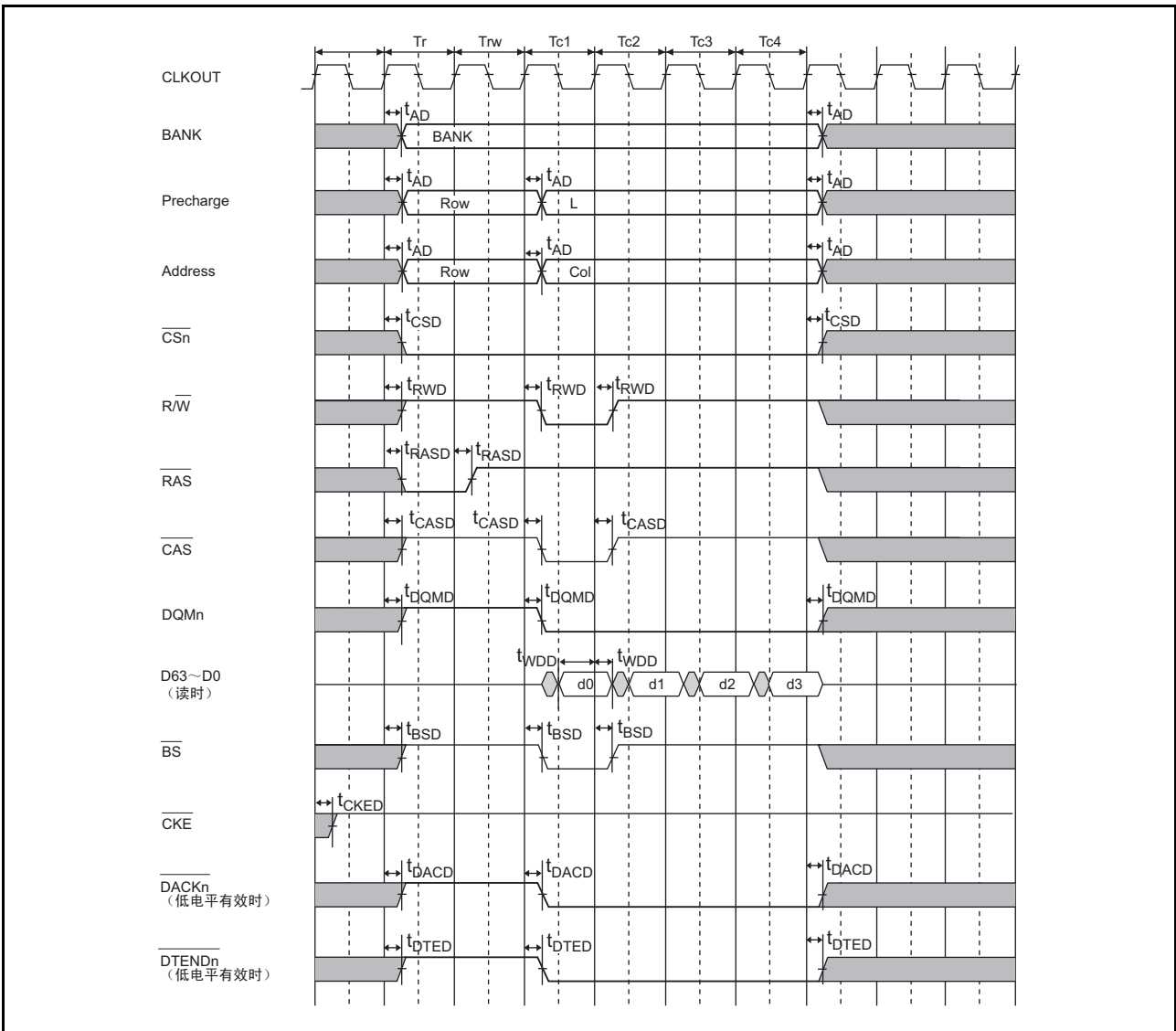


图 33.15 SDRAM 总线周期 存储体打开模式 写总线周期 (ACT-WRITE)
(BOMODE[1:0]=00、SRCD=0、IRCD=2cyc)

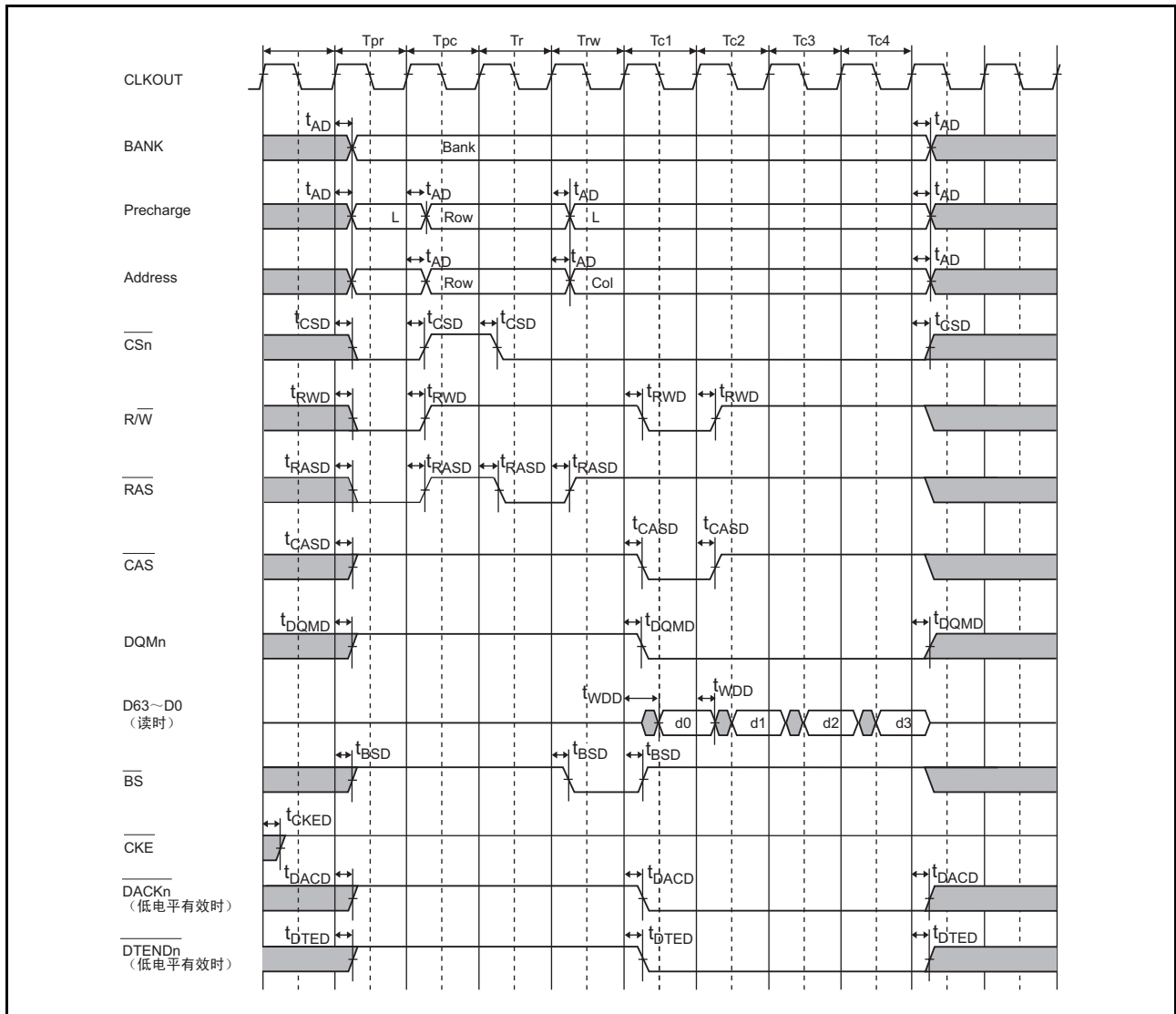


图 33.16 SDRAM 总线周期 存储体打开模式 预充电写总线周期 (PRE-ACT-SRITE) (BOMODE[1:0]=00、SRP[1:0]=00、SRCD=0、IRP=2cyc、IRCD=2cyc)

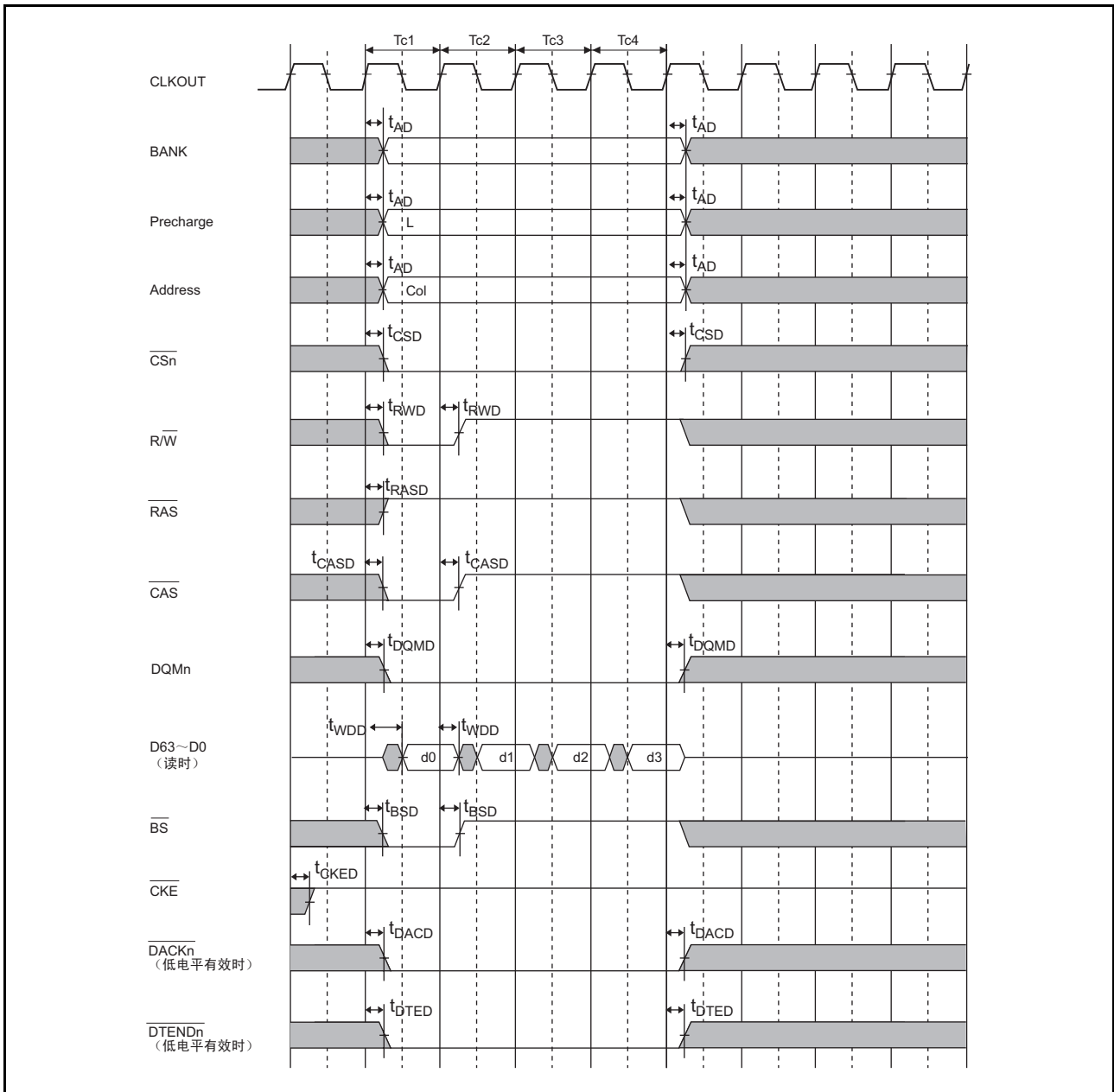


图 33.17 SDRAM 总线周期 存储体打开模式 写总线周期 (WRITE)
 (BOMODE[1:0]=00、SRCD=0、IRCD=2cyc)

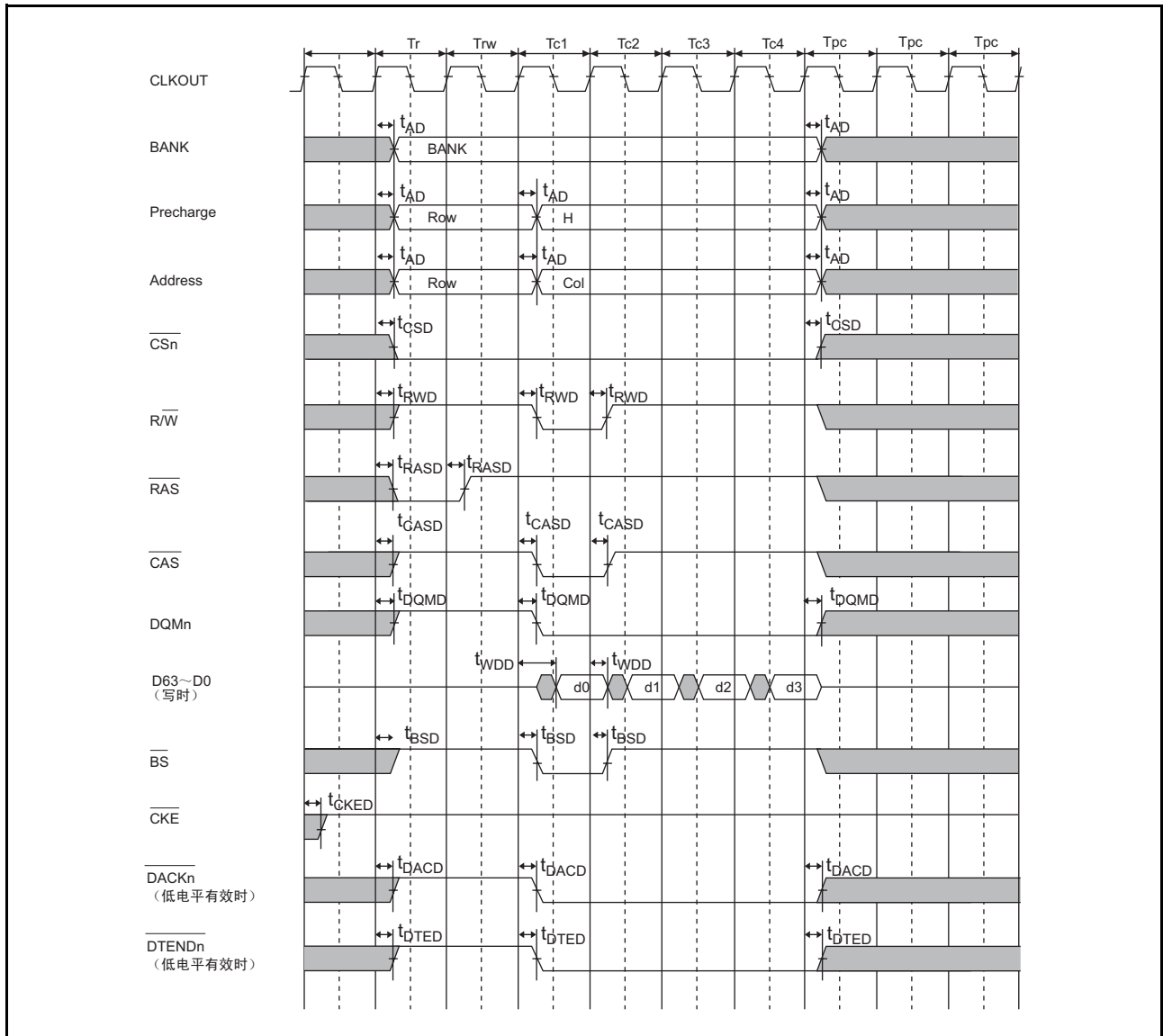


图 33.19 SDRAM 总线周期 存储体打开模式 写总线周期 (ACT-WRITEA)
 (BOMODE[1:0]=00、SWR[1:0]=00、SRP[1:0]=00、SRCD=0、IDAL=4cyc、IRCD=2cyc)

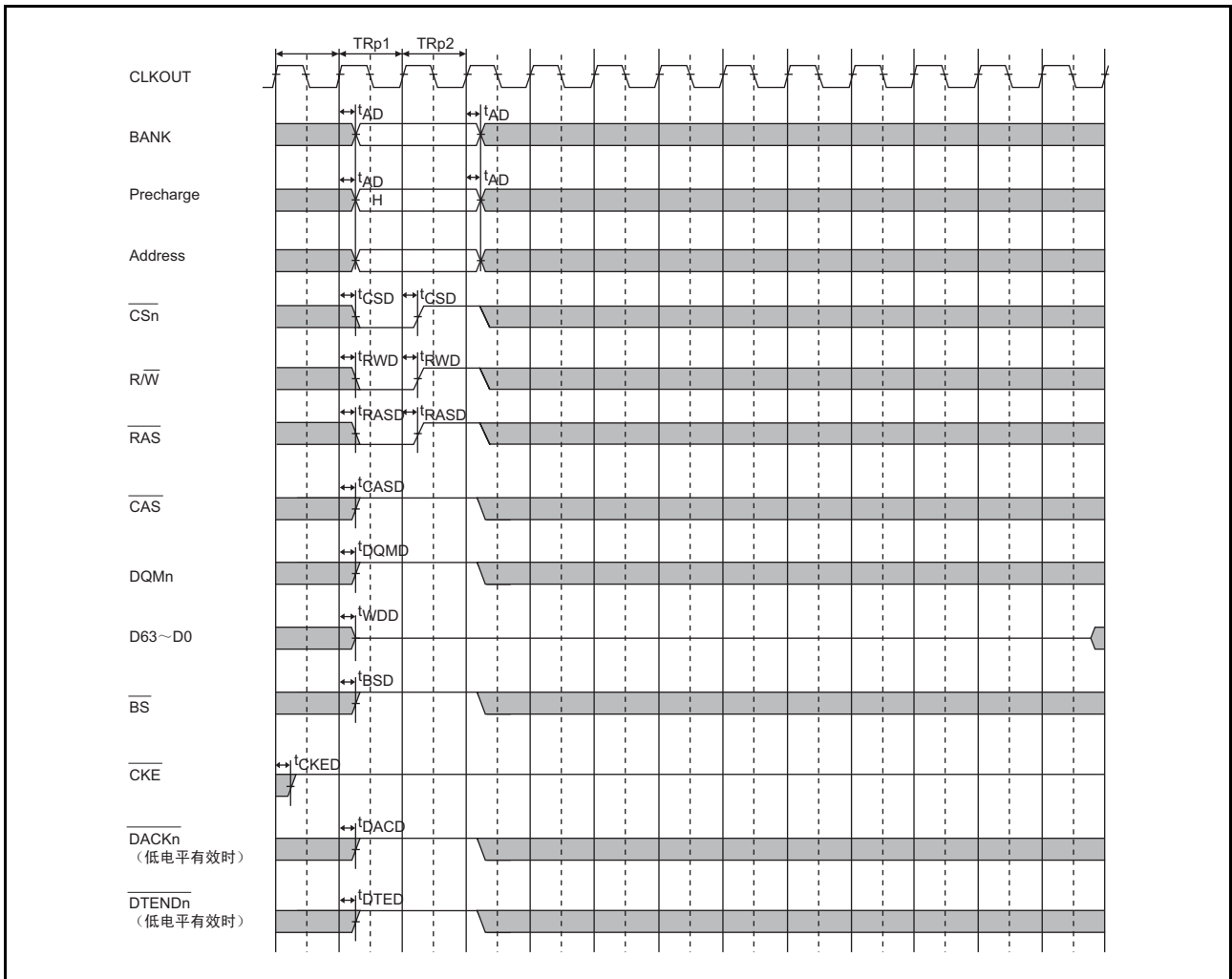


图 33.20 SDRAM 总线周期 预充电模式 (PALL) (SRP[1:0]=00、IRP=2cyc)

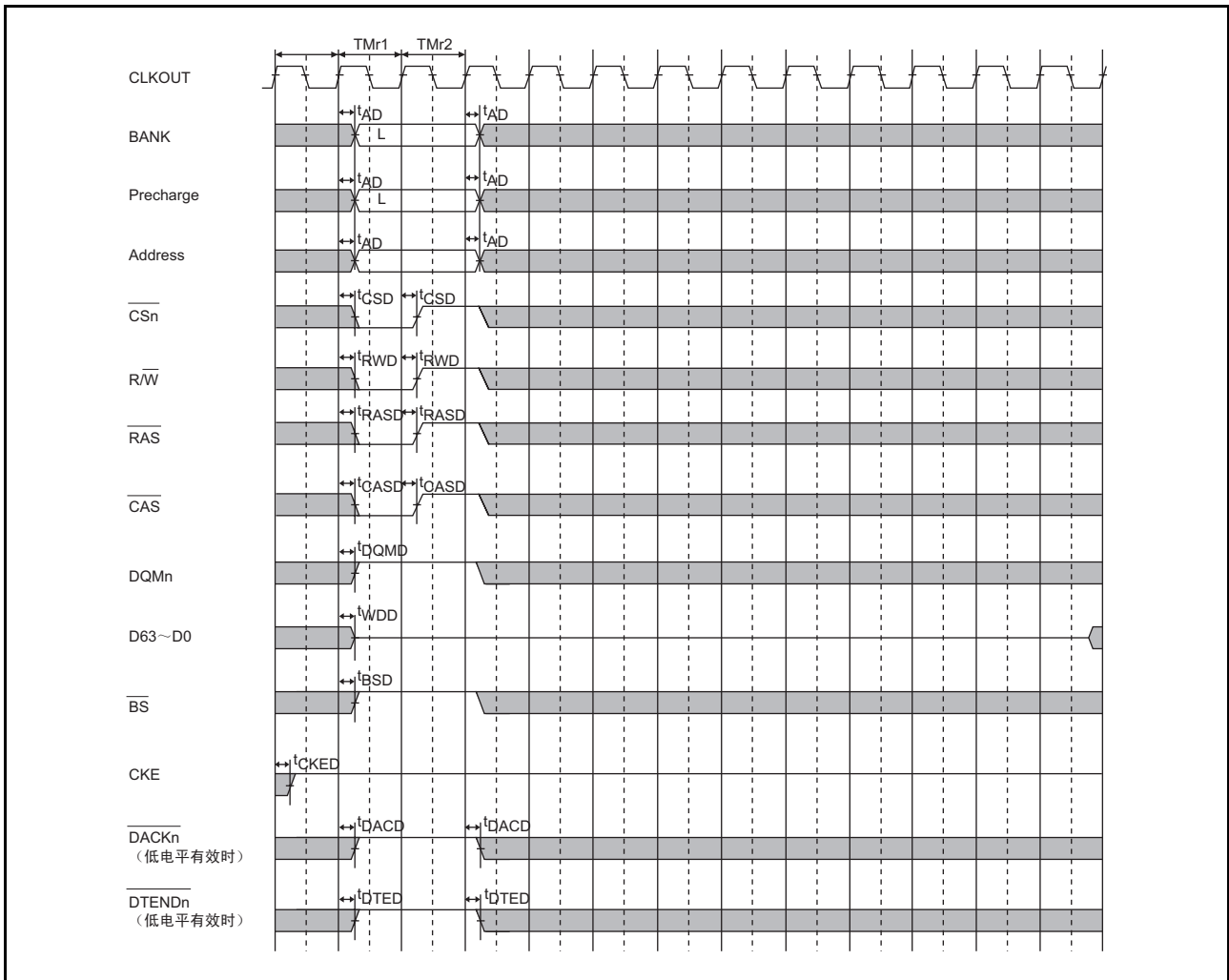


图 33.21 SDRAM 总线周期 模式寄存器设置周期 (MRS)

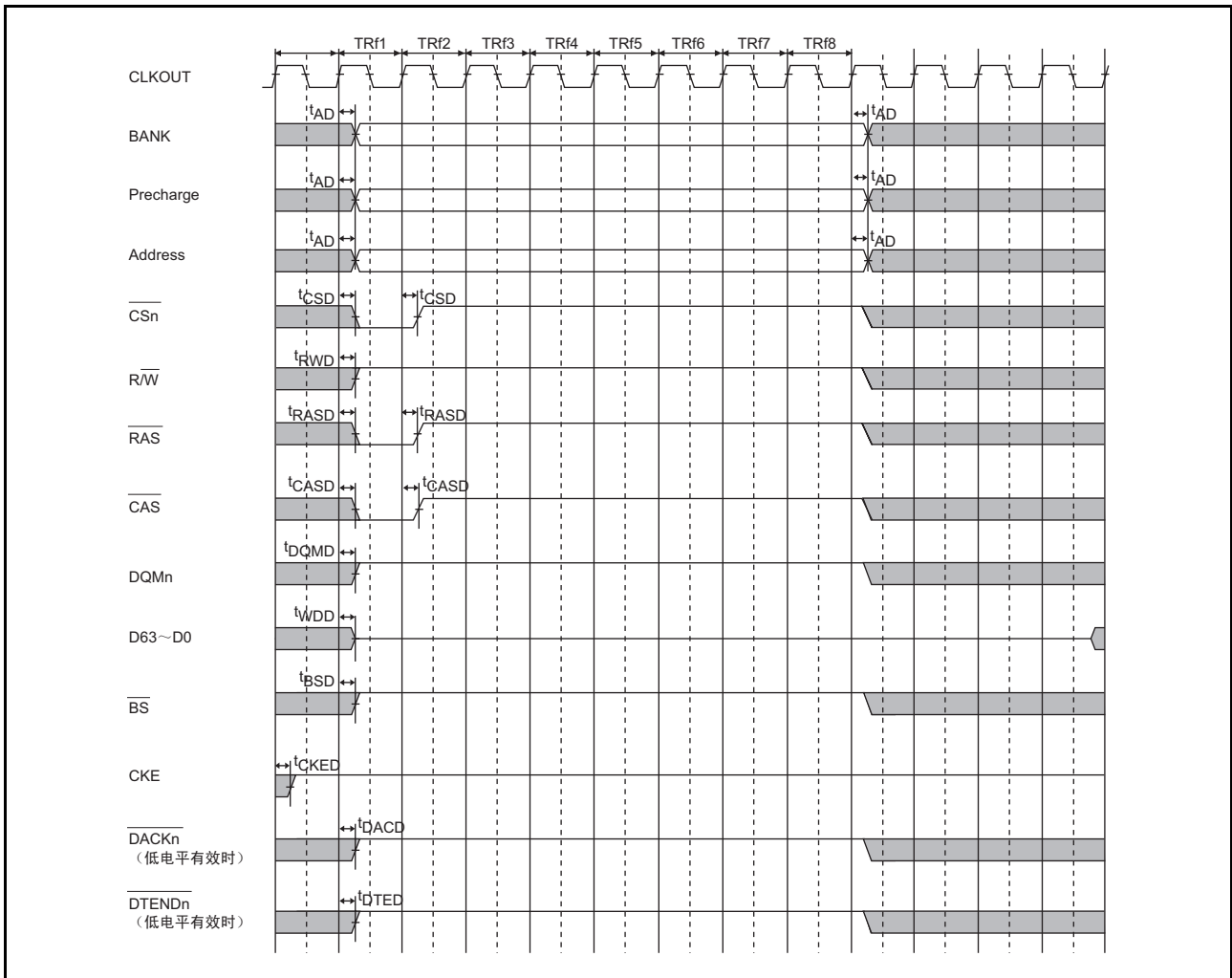


图 33.22 SDRAM 总线周期 自动刷新周期 (REF) (SRFC[2:0]=000、IRC=8cyc)

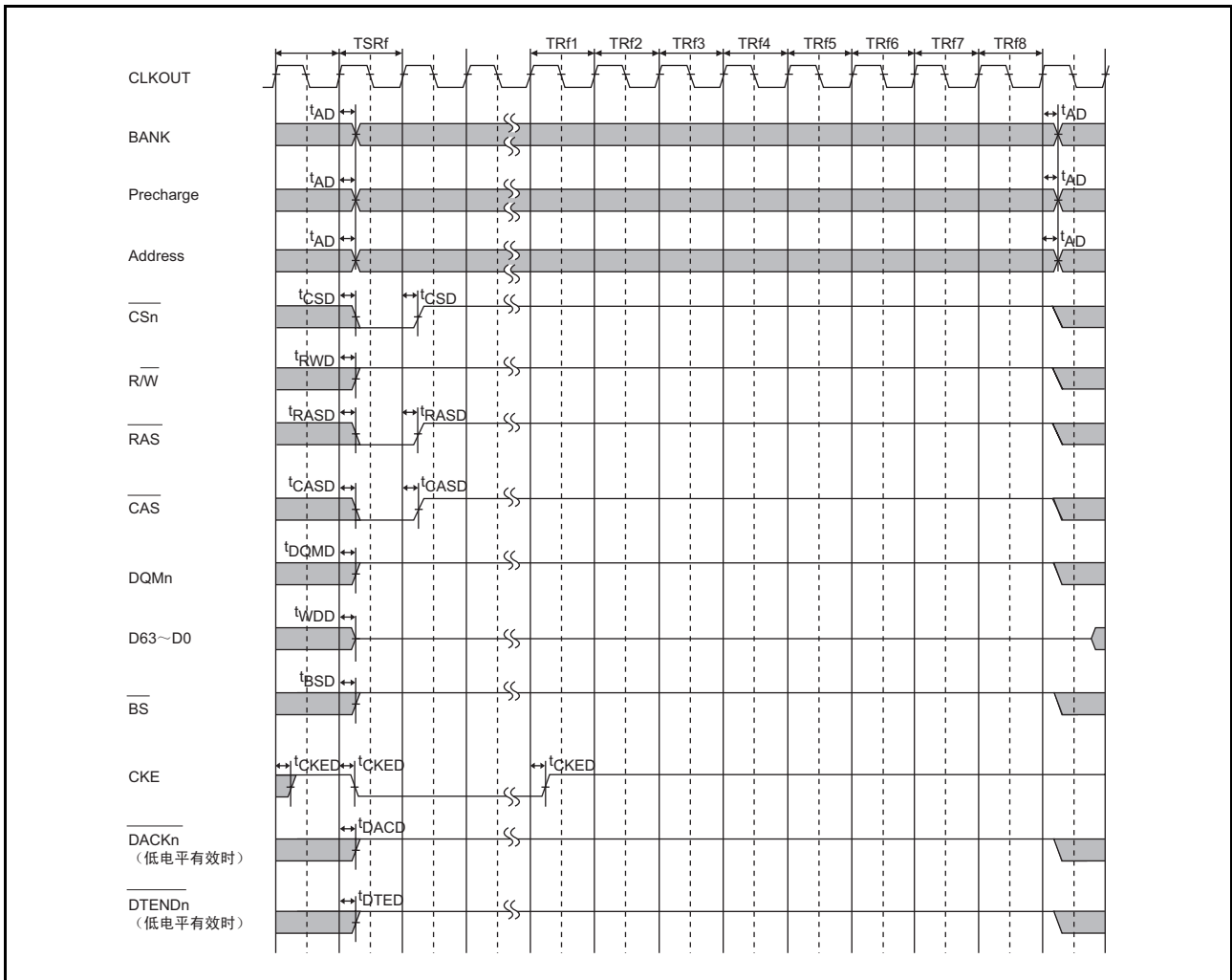


图 33.23 SDRAM 总线周期 自刷新周期 (SREF)

33.4.4 INTC 模块的信号时序

表 33.12 INTC 模块的信号时序

条件：3.3V 系列电源 = 3.0 ~ 3.6V、1.2V 系列电源 = 1.15 ~ 1.35V、 $T_a = -20 \sim 85^\circ\text{C}$ 、 $-40 \sim 85^\circ\text{C}$

项目	符号	Min.	Max.	单位	参照图	备注
NMI 脉宽 (High)	t_{NMIH}	5	—	t_{cyc}	33.24	正常模式时 睡眠模式时
NMI 脉宽 (Low)	t_{NMIL}	5	—	t_{cyc}	33.24	正常模式时 睡眠模式时
IRQ1 和 IRQ0 准备时间	t_{IRQS}	8	—	ns	33.25	IRQ 输入
IRQ1 和 IRQ0 保持时间	t_{IRQH}	3	—	ns	33.25	IRQ 输入
$\overline{\text{PINTn}}$ 中断准备时间	t_{GPIOS}	15	—	ns	33.25	GPIO 中断输入
$\overline{\text{PINTn}}$ 中断保持时间	t_{GPIOH}	8	—	ns	33.25	GPIO 中断输入
IRQOUT 输出延迟时间	t_{IRQOD}	—	13	ns	33.25	IRQOUT 输出

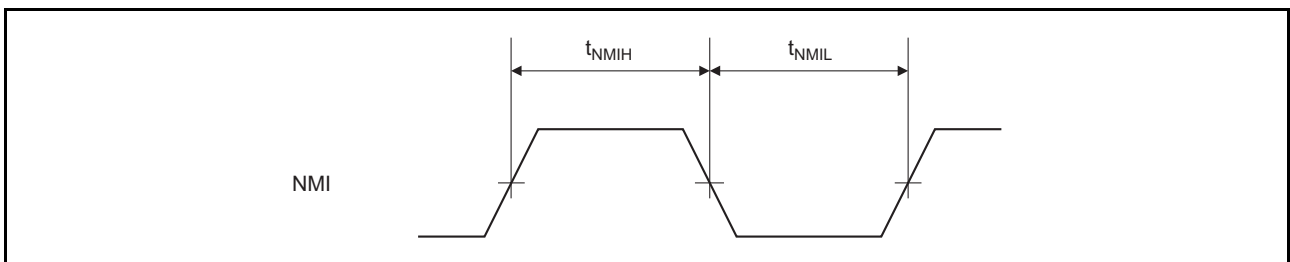
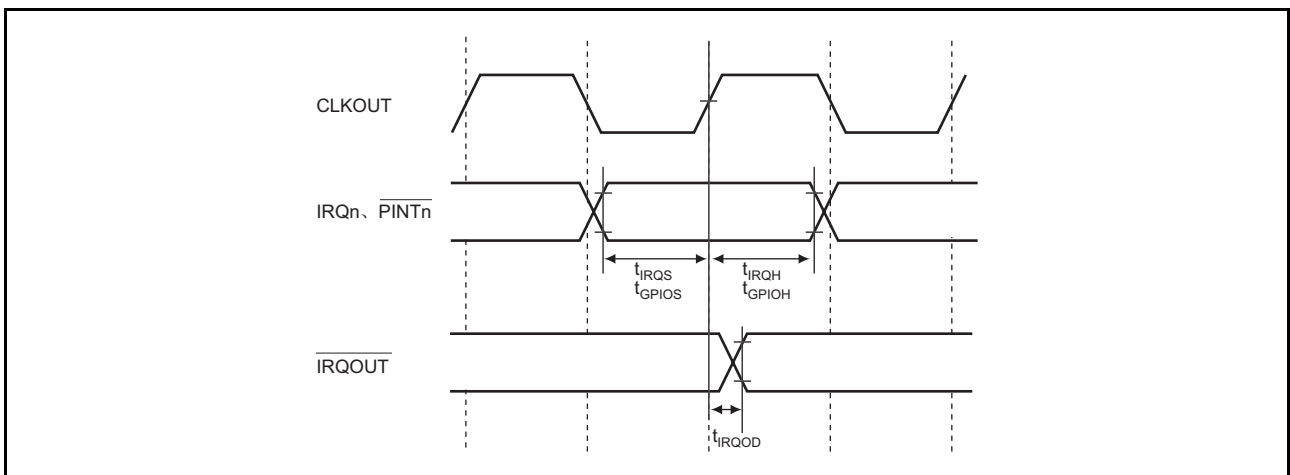
【注】 t_{cyc} 表示 1 个 CLKOUT 时钟周期的时间。

图 33.24 NMI 输入时序

图 33.25 IRQ 时序、 $\overline{\text{PINTn}}$ 输入时序和 $\overline{\text{IRQOUT}}$ 输出时序

33.4.5 DMAC 模块的信号时序

表 33.13 DMAC 模块的信号时序

条件: 3.3V 系列电源 = 3.0 ~ 3.6V、1.2V 系列电源 = 1.15 ~ 1.35V、 $T_a = -20 \sim 85^\circ\text{C}$ 、 $-40 \sim 85^\circ\text{C}$

模块	项目	符号	Min.	Max.	单位	参照图
DMAC	DREQn 准备时间	t_{DRQS}	6	—	ns	33.26
	DREQn 保持时间	t_{DRQH}	4	—	ns	33.26
	DTENDn 延迟时间	t_{DTED}	1.0	8	ns	33.26、33.8 ~ 33.23
	DACKn 延迟时间	t_{DADC}	1.0	8	ns	33.26、33.8 ~ 33.23

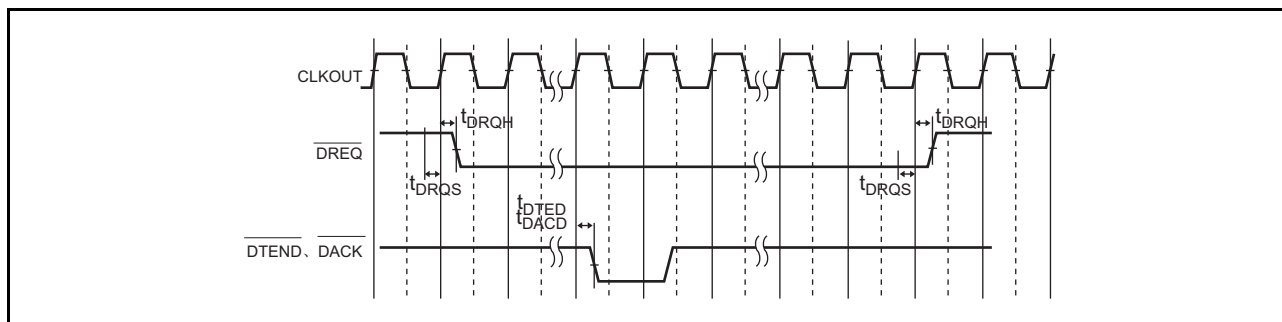


图 33.26 DREQ/DTEND/DACK 时序

33.4.6 TMU 模块的信号时序

表 33.14 TMU 模块的信号时序

条件: 3.3V 系列电源 = 3.0 ~ 3.6V、1.2V 系列电源 = 1.15 ~ 1.35V、 $T_a = -20 \sim 85^\circ\text{C}$ 、 $-40 \sim 85^\circ\text{C}$

模块	项目	符号	Min.	Max.	单位	参照图
TMU	定时器时钟脉宽 (High)	t_{TCLKWH}	4	—	t_{Pcyc}	33.27
	定时器时钟脉宽 (Low)	t_{TCLKWL}	4	—	t_{Pcyc}	33.27
	定时器时钟上升时间	t_{TCLKr}	—	0.8	t_{Pcyc}	33.27
	定时器时钟下降时间	t_{TCLKf}	—	0.8	t_{Pcyc}	33.27

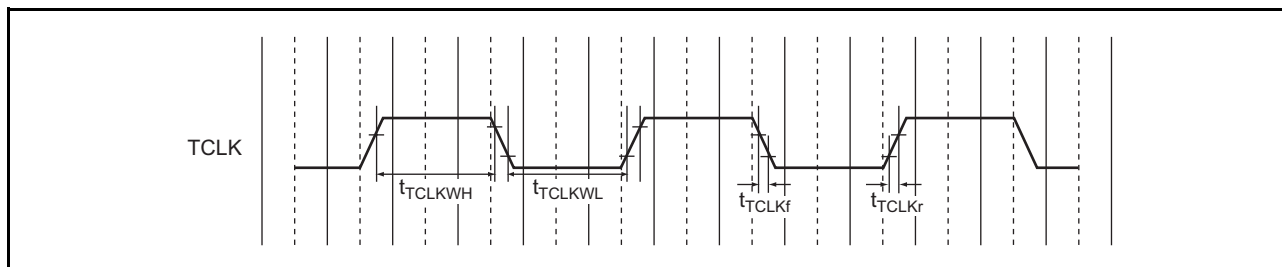
【注】 t_{Pcyc} 表示 1 个外围时钟 (Pck) 周期的时间。

图 33.27 TCLK 输入时序

33.4.7 IIC 模块的信号时序

表 33.15 IIC 模块的信号时序

条件: 3.3V 系列电源 = 3.0 ~ 3.6V、1.2V 系列电源 = 1.15 ~ 1.35V、 $T_a = -20 \sim 85^\circ\text{C}$ 、 $-40 \sim 85^\circ\text{C}$

项目	符号	min.	typ.	max.	单位	参照图
SCL 频率	t_{ICYC}	0	—	400	kHz	图 33.28、33.29 $R_P \cdot C_B = 257 \times 10^{-9}$ $\sim 275 \times 10^{-9} [\Omega \cdot \text{pF}]$ $V_{\text{PU}} = 3.3\text{V}$
SCL/SDA 下降时间	t_{ICF}	—	—	300	ns	
SDA 总线自由时间	t_{ICBF}	1.3	—	—	μs	
SCL 开始条件保持时间	t_{ICH}	0.6	—	—	μs	
SCL 重复开始条件的准备时间	t_{ICS}	0.6	—	—	μs	
SDA 停止条件准备时间	t_{ICST}	0.6	—	—	μs	
SDA 准备时间	t_{DAS}	100	—	—	ns	
SDA 保持时间	t_{ICDH}	0	—	0.9	μs	

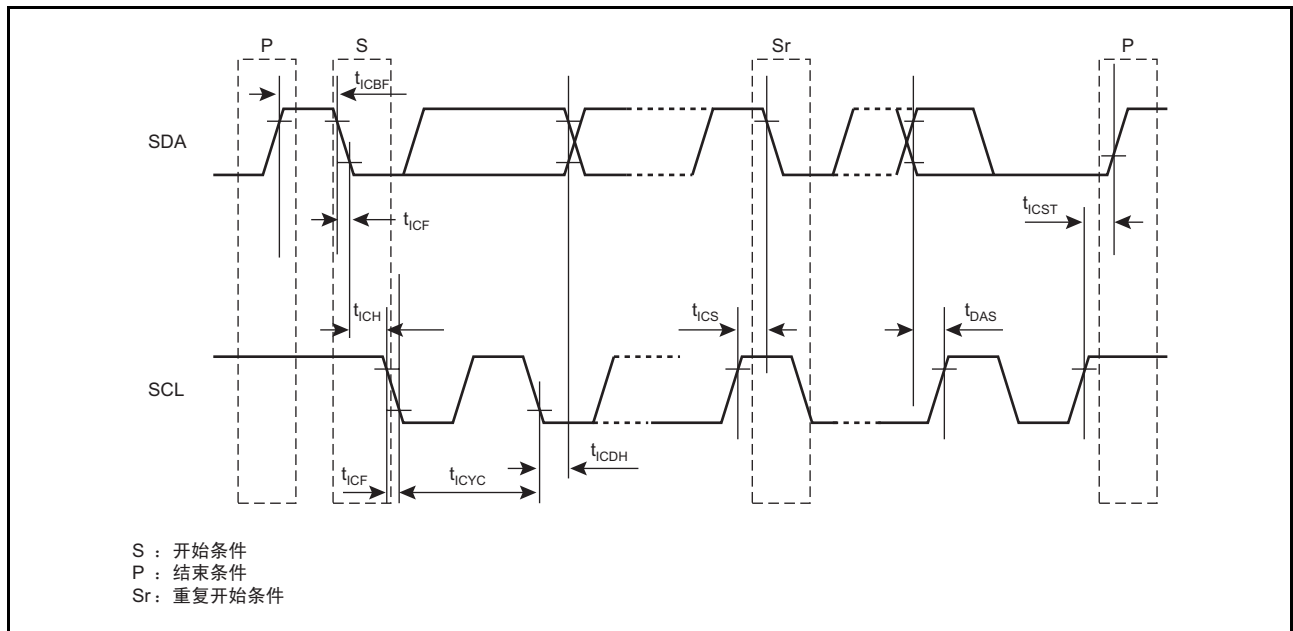


图 33.28 IIC 时序

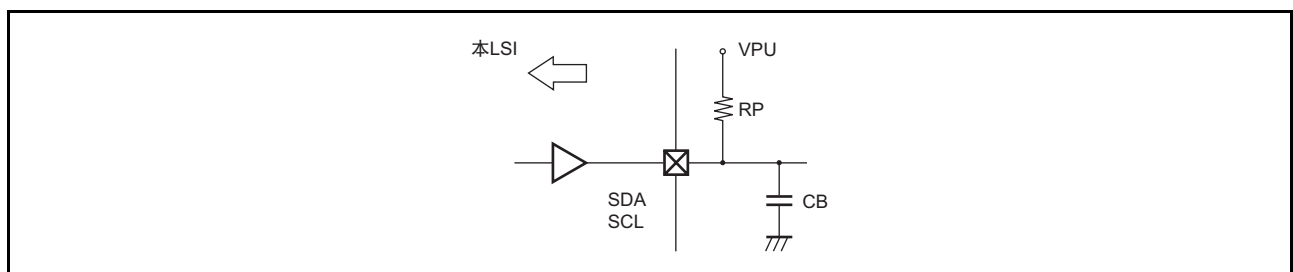


图 33.29 AC 特性负载条件

33.4.8 SCIF 模块的信号时序

表 33.16 SCIF 模块的信号时序

条件：3.3V 系列电源 =3.0 ~ 3.6V、1.2V 系列电源 =1.15 ~ 1.35V、 $T_a = -20 \sim 85^\circ\text{C}$ 、 $-40 \sim 85^\circ\text{C}$

项目	符号	Min.	Max.	单位	参照图
输入时钟周期	时钟同步	12	—	t_{pcyc}	33.30
	异步	4	—	t_{pcyc}	33.30
输入时钟上升时间	t_{SCKr}	—	1.5	t_{pcyc}	33.30
输入时钟下降时间	t_{SCKf}	—	1.5	t_{pcyc}	33.30
输入时钟宽度	t_{SCKW}	0.4	0.6	t_{Scyc}	33.30
发送数据延迟时间（时钟同步）	t_{TXD}	—	$3t_{\text{pcyc}}+15$	t_{pcyc}	33.31
接收数据准备时间（时钟同步）	t_{RXS}	$4t_{\text{pcyc}}+15$	—	ns	33.31
接收数据保持时间（时钟同步）	t_{RXH}	100	—	ns	33.31

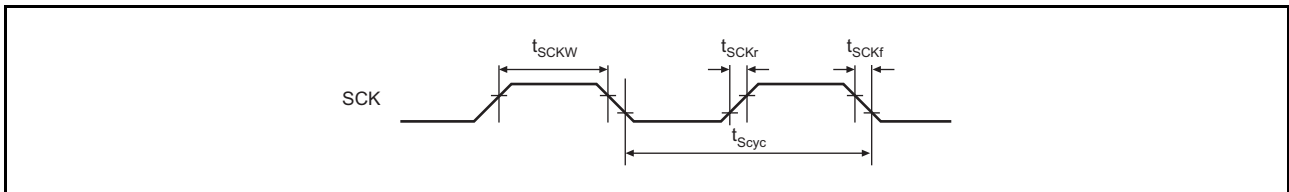
【注】 t_{pcyc} 表示 1 个外围时钟（Pck）周期的时间。

图 33.30 SCK 输入时钟时序

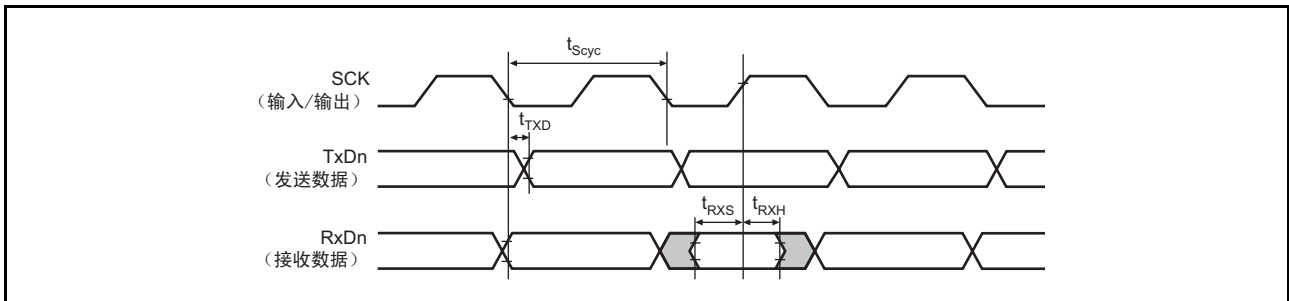


图 33.31 时钟同步模式时的 SCIF 输入 / 输出时序

33.4.9 SSI 模块的信号时序

表 33.17 SSI 模块的信号时序

条件：3.3V 系列电源 =3.0 ~ 3.6V、1.2V 系列电源 =1.15 ~ 1.35V、 $T_a = -20 \sim 85^\circ\text{C}$ 、 $-40 \sim 85^\circ\text{C}$

项目	符号	min	typ	max	单位	备注	参照图
输出时钟周期	t_O	160	—	3364	ns	输出	33.32
输入时钟周期	t_I	160	—	3364	ns	输入	
时钟 High	t_{HC}	40	—	—	ns	双向	
时钟 Low	t_{LC}	40	—	—	ns		
时钟上升时间	t_{RC}	—	—	20	ns	输出 (100pF)	
延迟	t_{DTR}	—	—	30	ns	发送	33.33、33.34
准备时间	t_{SR}	20	—	—	ns	接收	33.35、33.36
保持时间	t_{HTR}	10	—	—	ns	接收	33.35、33.36
AUDIO_CLK 频率	f_{AUDIO}	3.072	—	24.576	MHz		33.37

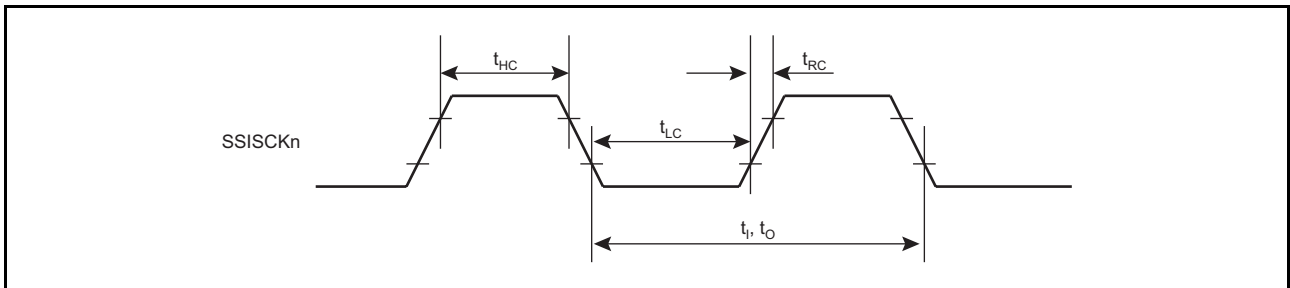


图 33.32 时钟输入 / 输出时序

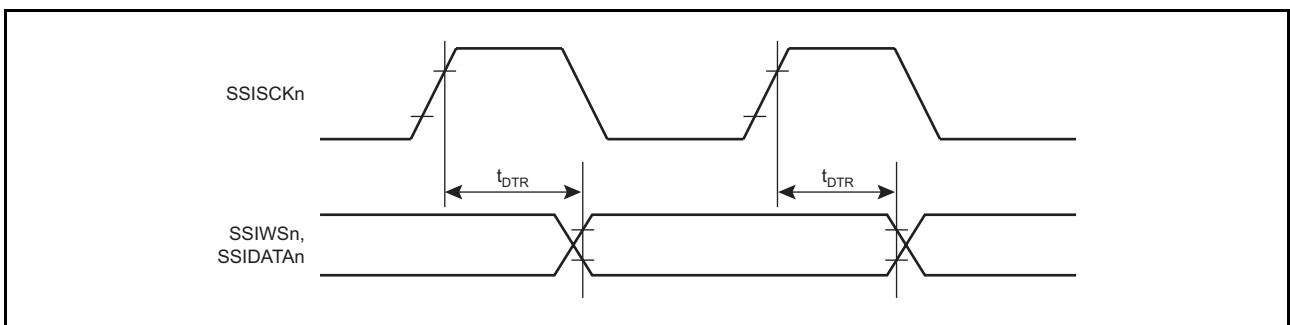


图 33.33 SSI 发送时序 (1)

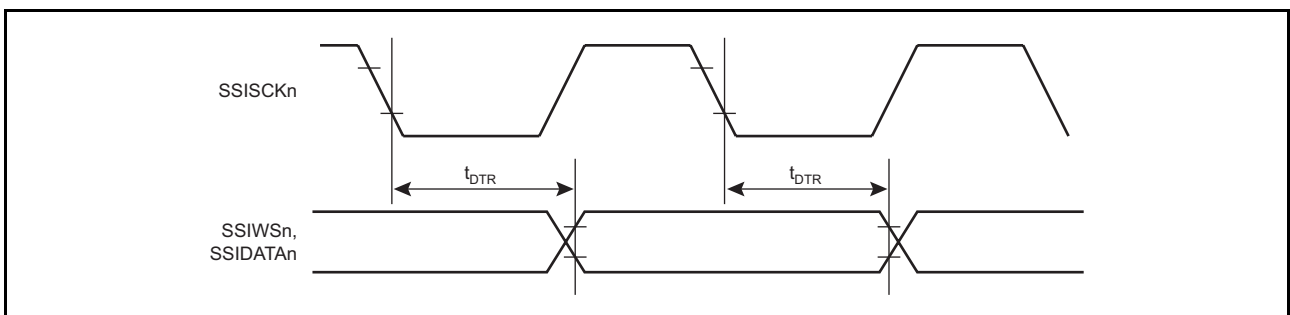


图 33.34 SSI 发送时序 (2)

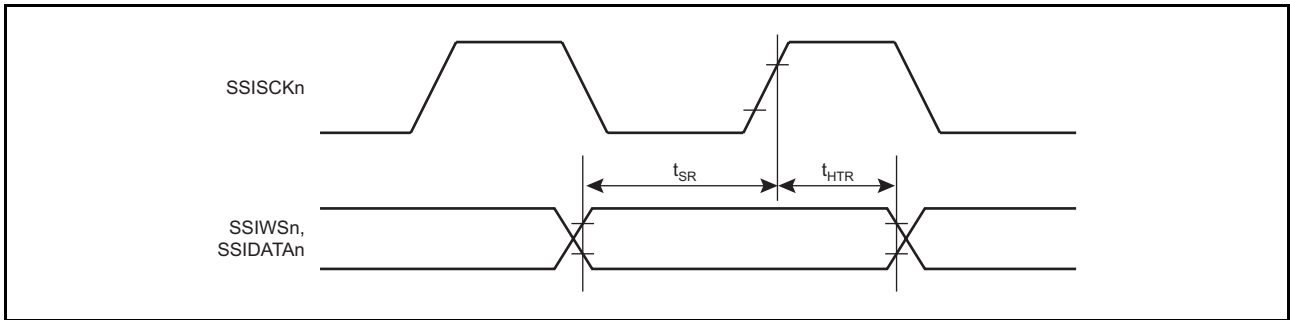


图 33.35 SSI 接收时序 (1)

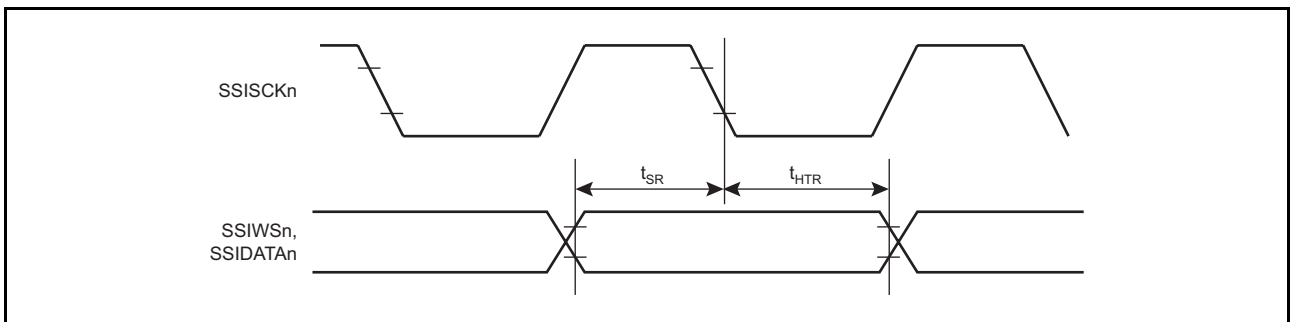


图 33.36 SSI 接收时序 (2)

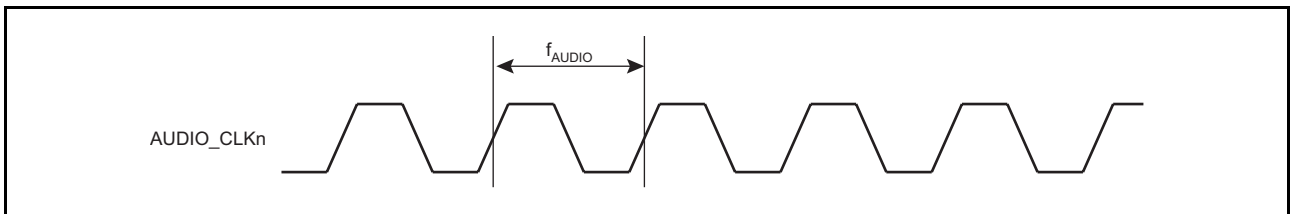


图 33.37 AUDIO_CLK 时序

33.4.10 ATAPI 接口模块的信号时序

表 33.18 ATAPI 接口 PIO 传送的寄存器存取时序

条件: 3.3V 系列电源 =3.0 ~ 3.6V、1.2V 系列电源 =1.15 ~ 1.35V、Ta=-20 ~ 85°C、-40 ~ 85°C

PIO 传送的寄存器存取时的符号、项目和条件 (max/min)			Mode0 ns	Mode1 ns	Mode2 ns	Mode3 ns	Mode4 ns	参照图
t0	周期时间	(min)	600	383	330	180	120	33.38
t1	地址准备时间	(min)	70	50	30	30	25	
t2	$\overline{\text{IDEIORD}}/\overline{\text{IDEIOWR}}$ 脉宽 8-bit	(min)	290	290	290	80	70	
t2i	$\overline{\text{IDEIORD}}/\overline{\text{IDEIOWR}}$ 恢复时间	(min)	—	—	—	70	25	
t3	$\overline{\text{IDEIOWR}}$ 数据准备时间	(min)	60	45	30	30	20	
t4	$\overline{\text{IDEIOWR}}$ 数据保持时间	(min)	30	20	15	10	10	
t5	$\overline{\text{IDEIORD}}$ 数据准备时间	(min)	50	35	20	20	20	
t6	$\overline{\text{IDEIORD}}$ 数据保持时间	(min)	5	5	5	5	5	
t6z	$\overline{\text{IDEIORD3}}$ 状态延迟时间	(max)	30	30	30	30	30	
t9	地址保持时间	(min)	20	15	10	10	10	
tRD	$\overline{\text{IDEIORDY}}$ 读数据的有效时间	(min)	0	0	0	0	0	
tA	$\overline{\text{IDEIORDY}}$ 准备时间		35	35	35	35	35	
tB	$\overline{\text{IDEIORDY}}$ 脉冲时间	(max)	1250	1250	1250	1250	1250	
tC	从 $\overline{\text{IDEIORDY}}$ 引脚无效到高阻抗的时间	(max)	5	5	5	5	5	

表 33.19 ATAPI 接口 PIO 的数据传送时序

PIO 传送的寄存器存取时的符号、项目和条件 (max/min)			Mode0 ns	Mode1 ns	Mode2 ns	Mode3 ns	Mode4 ns	参照图
t0	周期时间	(min)	600	383	240	180	120	33.38
t1	地址准备时间	(min)	70	50	30	30	25	
t2	$\overline{\text{IDEIORD}}/\overline{\text{IDEIOWR}}$ 脉宽 8-bit	(min)	290	290	290	80	70	
t2i	$\overline{\text{IDEIORD}}/\overline{\text{IDEIOWR}}$ 恢复时间	(min)	—	—	—	70	25	
t3	$\overline{\text{IDEIOWR}}$ 数据准备时间	(min)	60	45	30	30	20	
t4	$\overline{\text{IDEIOWR}}$ 数据保持时间	(min)	30	20	15	10	10	
t5	$\overline{\text{IDEIORD}}$ 数据准备时间	(min)	50	35	20	20	20	
t6	$\overline{\text{IDEIORD}}$ 数据保持时间	(min)	5	5	5	5	5	
t6z	$\overline{\text{IDEIORD3}}$ 状态延迟时间	(max)	30	30	30	30	30	
t9	地址保持时间	(min)	20	15	10	10	10	
tRD	$\overline{\text{IDEIORDY}}$ 读数据的有效时间	(min)	0	0	0	0	0	
tA	$\overline{\text{IDEIORDY}}$ 准备时间		35	35	35	35	35	
tB	$\overline{\text{IDEIORDY}}$ 脉冲时间	(max)	1250	1250	1250	1250	1250	
tC	从 $\overline{\text{IDEIORDY}}$ 引脚无效到高阻抗的时间	(max)	5	5	5	5	5	

表 33.20 ATAPI 接口 多字传送时序

多字传送的符号、项目和条件 (max/min)			Mode0 ns	Mode1 ns	Mode2 ns	参照图
t0	周期时间	(min)	480	150	120	33.40 ~ 33.42
tD	$\overline{\text{IDEIORD}}/\overline{\text{IDEIOWR}}$ 脉宽	(min)	215	80	70	33.39 ~ 33.42
tE	$\overline{\text{IDEIORD}}$ 数据存取时间	(max)	150	60	50	
tF	$\overline{\text{IDEIORD}}$ 数据保持时间	(min)	5	5	5	
tG	$\overline{\text{IDEIORD}}/\overline{\text{IDEIOWR}}$ 数据准备时间	(min)	100	30	20	
tH	$\overline{\text{IDEIOWR}}$ 数据保持时间	(min)	20	15	10	
tI	$\overline{\text{IODACK}}$ 准备时间	(min)	0	0	0	33.39
tJ	$\overline{\text{IODACK}}$ 保持时间	(min)	20	5	5	33.41、33.42
tKR	$\overline{\text{IDEIORD}}$ 引脚无效脉宽	(min)	50	50	25	33.40 ~ 33.42
tKW	$\overline{\text{IDEIOWR}}$ 引脚无效脉宽	(min)	215	50	25	33.40 ~ 33.42
tLR	$\overline{\text{IDEIORD}} \cdot \overline{\text{IODREQ}}$ 延迟时间	(max)	120	40	35	33.41
tLW	$\overline{\text{IDEIOWR}} \cdot \overline{\text{IODREQ}}$ 延迟时间	(max)	40	40	35	
tM	$\overline{\text{IDECS}}[1:0]$ 准备时间	(min)	50	30	25	33.39
tN	$\overline{\text{IDECS}}[1:0]$ 保持时间	(min)	15	10	10	33.41、33.42
tZ	$\overline{\text{IODACK3}}$ 状态延迟时间	(max)	20	25	25	

表 33.21 ATAPI 接口 超极 DMA 传送时序

超级 DMA 传送 的符号	Mode0 ns		Mode1 ns		Mode2 ns		Mode3 ns		Mode4 ns		参照图
	min	max	min	max	min	max	min	max	min	max	
t2CYCTYP	240		160		120		90		60		33.44
tCYC	112		73		54		39		25		33.44、33.49
t2CYC	230		153		115		86		57		
tDS	15		10		7		7		5		
tDH	5		5		5		5		5		
tDVS	70		48		31		20		6.7		
tDVH	6.2		6.2		6.2		6.2		6.2		33.43、33.44、 33.48、33.49
tCS	15		10		7		7		5		
tCH	5		5		5		5		5		
tCVS	70		48		31		20		6.7		
tCVH	6.2		6.2		6.2		6.2		6.2		33.46、33.47、 33.51、33.52
tZFS	0		0		0		0		0		33.43
tDZFS	70		48		31		20		6.7		33.43、33.48
tFS		230		200		170		130		120	33.43
tLI	0	150	0	150	0	150	0	100	0	100	33.46 ~ 33.48 33.51、33.52
tMLI	20		20		20		20		20		33.46、33.47、 33.51、33.52
tUI	0		0		0		0		0		33.43、33.48
tAZ		10		10		10		10		10	33.43、33.46、 33.47
tZAH	20		20		20		20		20		33.46、33.47
tZAD	0		0		0		0		0		33.43
tENV	20	70	20	70	20	70	20	55	20	55	33.43、33.48
tRFS		75		70		60		60		60	33.45、33.47、 33.50、33.52
tRP	160		125		100		100		100		
tIORDYZ		20		20		20		20		20	33.46、33.47、 33.51、33.52
tZIORDY	0		0		0		0		0		33.43、33.48
tACK	20		20		20		20		20		33.43、33.46 ~ 33.48、33.51、 33.52
tSS	50		50		50		50		50		33.46、33.51

表 33.22 ATAPI 接口 有关超级 DMA 传送时序的符号

符号	备注
t2CYCTYP	平均周期时间 (2 个周期)
tCYC	周期时间
t2CYC	最小周期时间 (2 个周期)
tDS	数据准备时间 (接收方)
tDH	数据保持时间 (接收方)
tDVS	数据准备时间 (发送方)
tDVH	数据保持时间 (发送方)
tCS	CRC 数据准备时间 (接收方)
tCH	CRC 数据保持时间 (接收方)
tCVS	CRC 数据准备时间 (发送方)
tCVH	CRC 数据保持时间 (发送方)
tZFS	从选通状态到驱动状态的准备时间 (发送方)
tDZFS	从驱动状态到最初的选通状态的准备时间 (发送方)
tFS	最初的选通时间
tLI	有限制的互锁时间
tMLI	最小互锁时间
tUI	无限制的互锁时间
tAZ	输出释放时间
tZAH	输出延迟时间
tZAD	输出确定时间 (从释放开始)
tENV	包络时间
tRFS	最后的选通时间
tRP	到 STOP 有效或者 DMARQ 无效为止的时间
tIORDYZ	到释放 IORDY 为止的时间
tZIORDY	到驱动 STROBE 为止的时间
tACK	DMACK 准备时间 / 保持时间
tSS	STROBE STOP 时间

表 33.23 ATAPI 接口 DIRECTION 的时序

项目和符号	Mode0 ns		Mode1 ns		Mode2 ns		Mode3 ns		Mode4 ns		参照图
	min	max	min	max	min	max	min	max	min	max	
PIO 写操作时的 DIRECTION 下降延 迟时间 tDIRECTION_WF	65	74	45	54	25	34	25	34	25	34	33.53
PIO 写操作时的 DIRECTION 上升延 迟时间 tDIRECTION_WR	47	55	47	55	47	55	47	55	47	55	
多字 DMA 数据输出 DIRECTION 下降延 迟时间 tMDIRECTION_F	-3	5	-3	5	-3	5	—	—	—	—	33.55
多字 DMA 数据输出 DIRECTION 上升延 迟时间 tMDIRECTION_R	7	15	7	15	7	15	—	—	—	—	
超级 DMA 数据输入 CRC 发送时的 DIRECTION 下降延 迟时间 tUDIRECTION_F (CRC)	116	124	76	84	56	64	46	54	36	44	33.57、33.58
超级 DMA 数据输入 CRC 发送时的 DIRECTION 上升延 迟时间 tUDIRECTION_R (CRC)	17	25	17	25	17	25	17	25	17	25	
超级 DMA 数据输出 时的 DIRECTION 下降延迟时间 tUDIRECTION_F	25	34	25	34	25	34	25	34	38	43	33.59
超级 DMA 数据输出 时的 DIRECTION 上升延迟时间 tUDIRECTION_R	48	55	48	55	48	55	48	55	48	55	33.60、33.61
从 DIRECTION 下 降到启动 IDE 数 据总线的时间 tDON	9	15	9	15	9	15	9	15	18	22	33.53、33.55、 33.57 ~ 33.59
从 IDE 数据总线 OFF 状态到 DIRECTION 上升的 时间 tDOFF	6	14	6	14	6	14	6	14	6	14	33.53、33.55、 33.57、33.58、 33.60、33.61

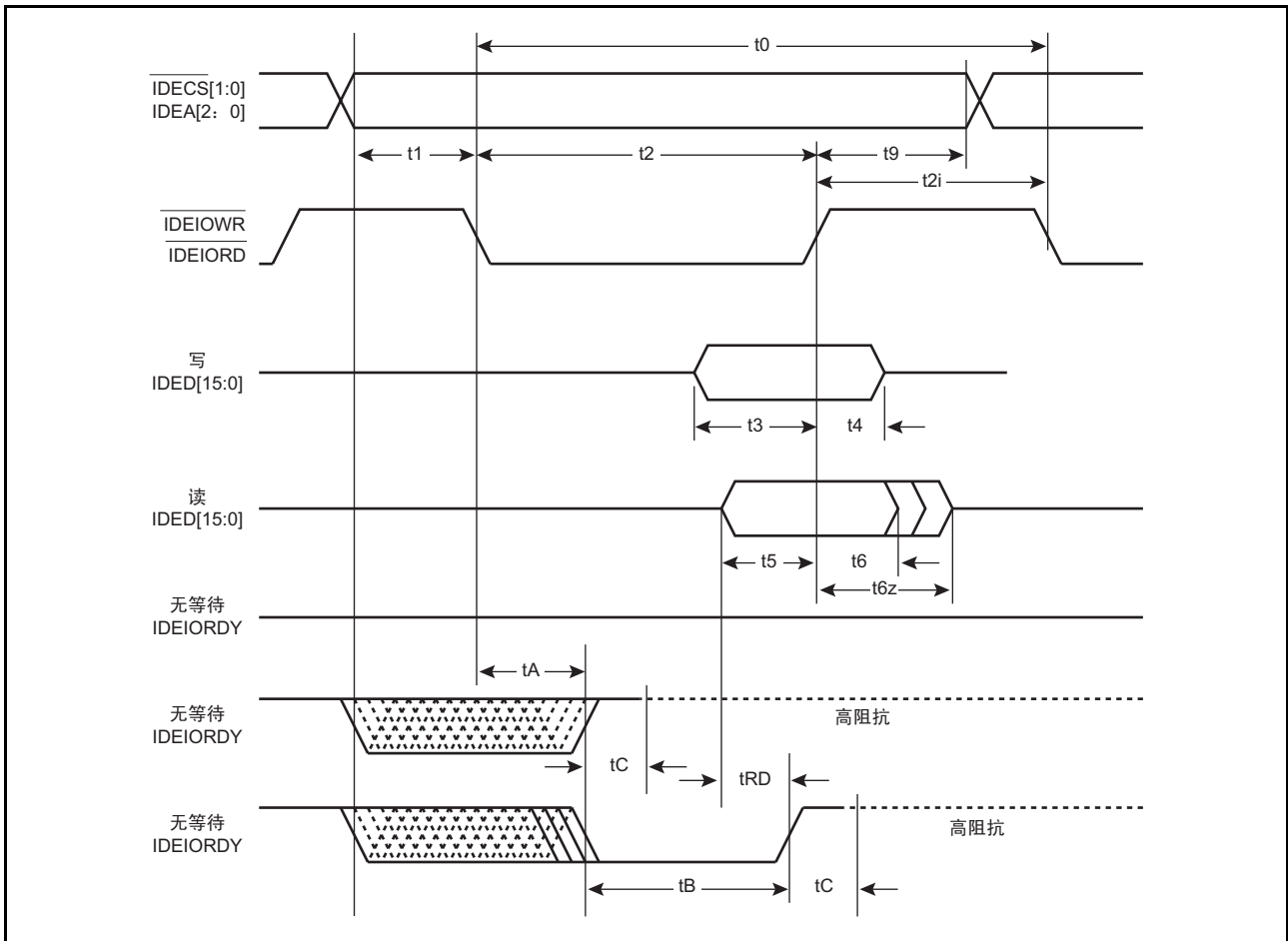


图 33.38 设备间的 PIO 数据传送和寄存器传送

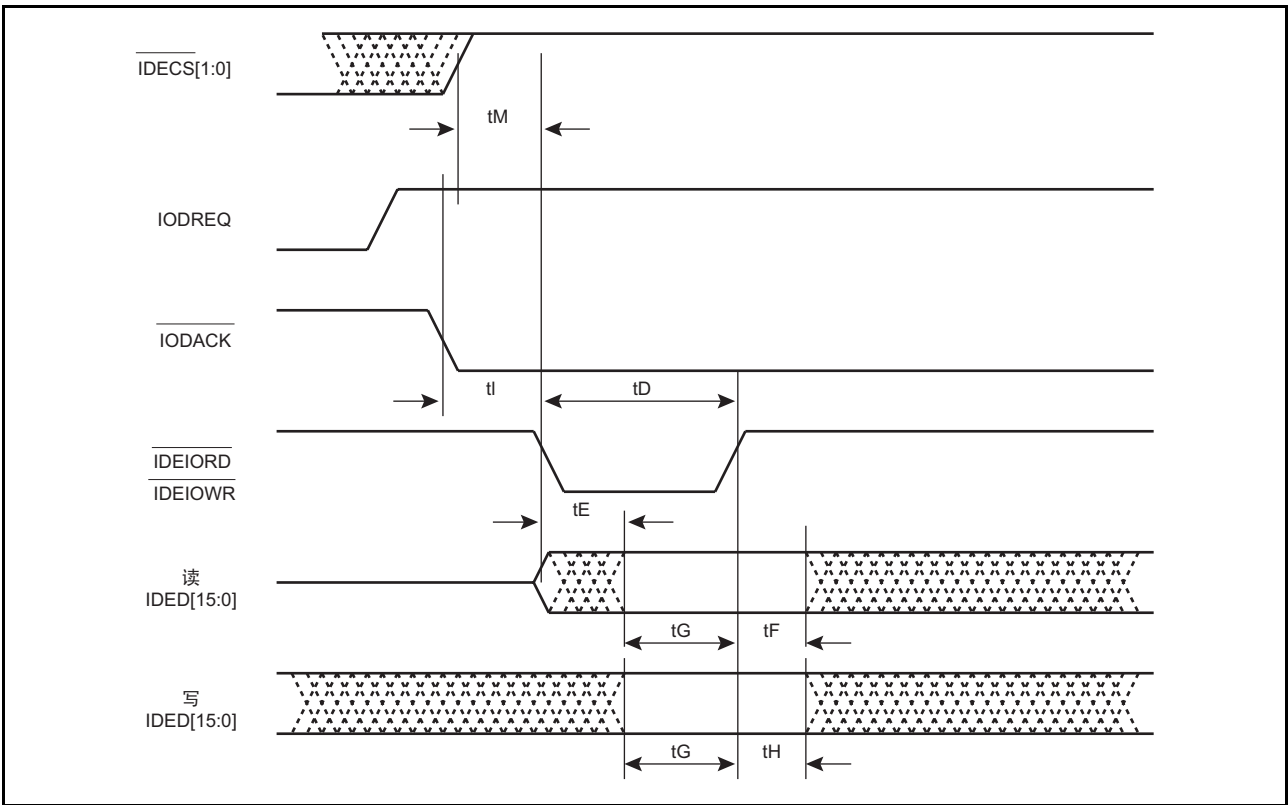


图 33.39 多字 DMA 数据的传送开始

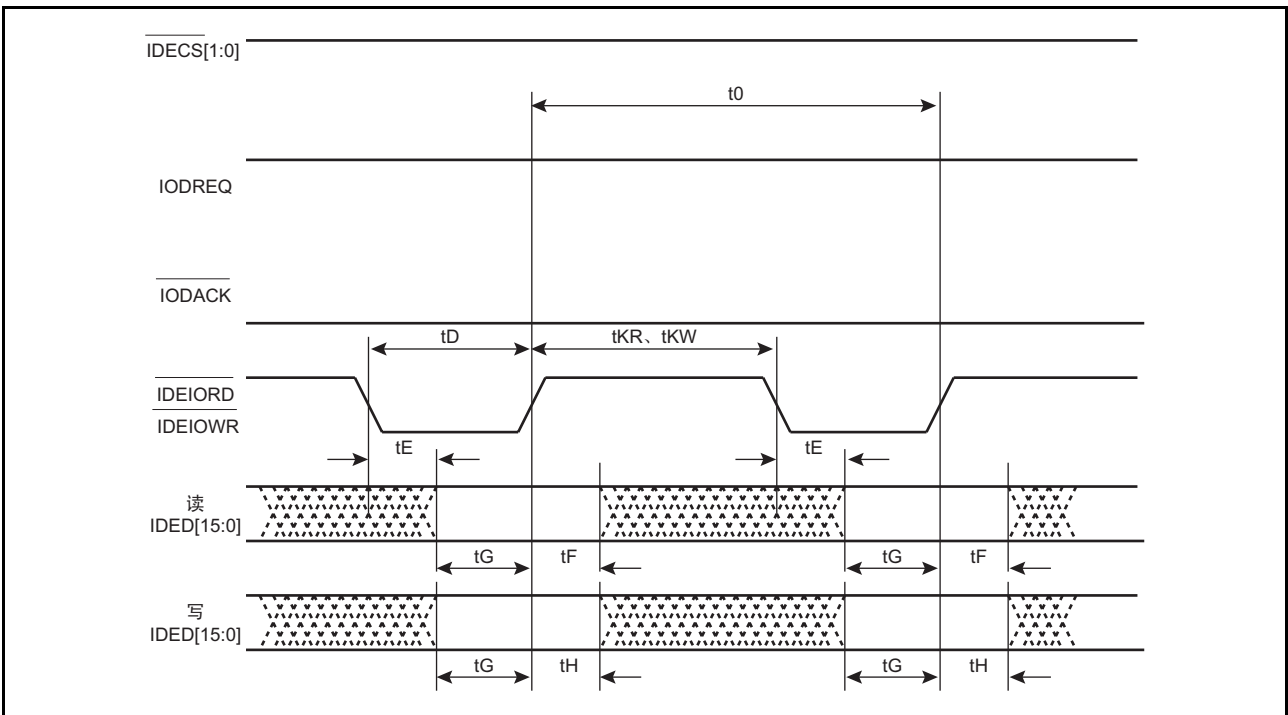


图 33.40 多字 DMA 数据传送

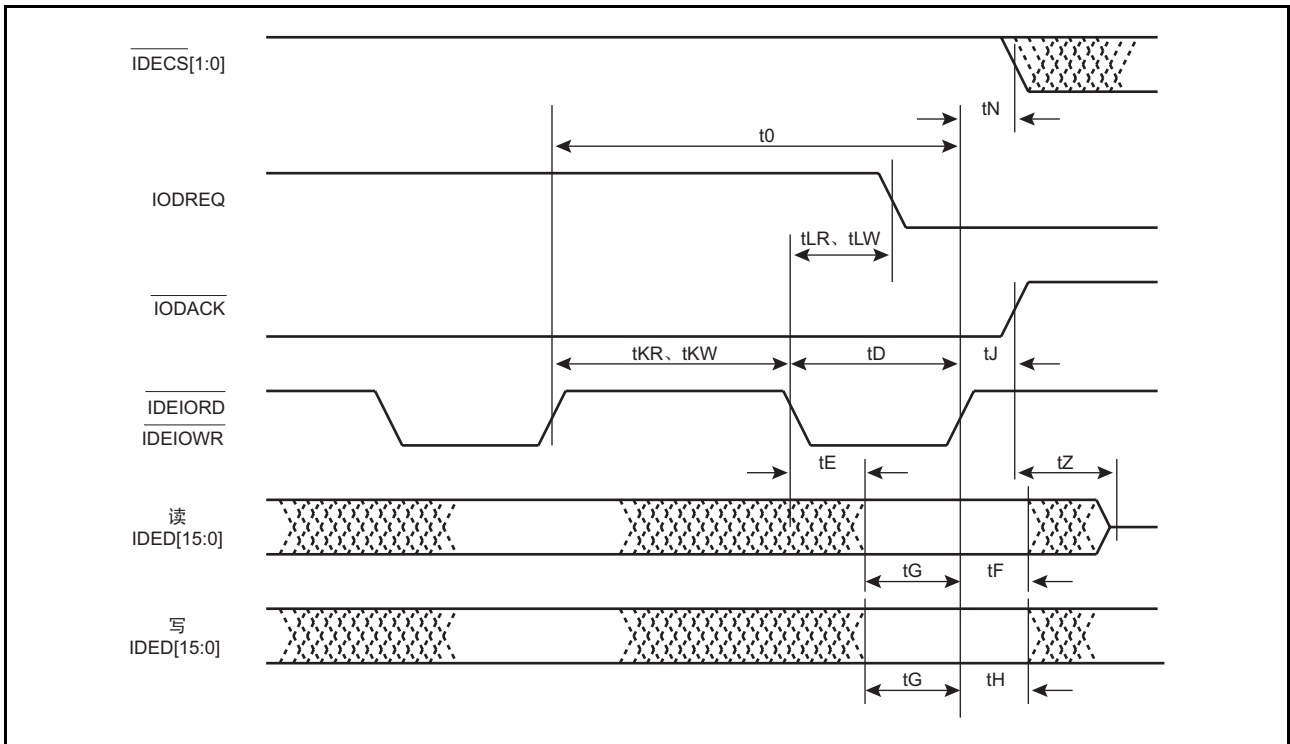


图 33.41 设备的多字 DMA 数据的传送结束

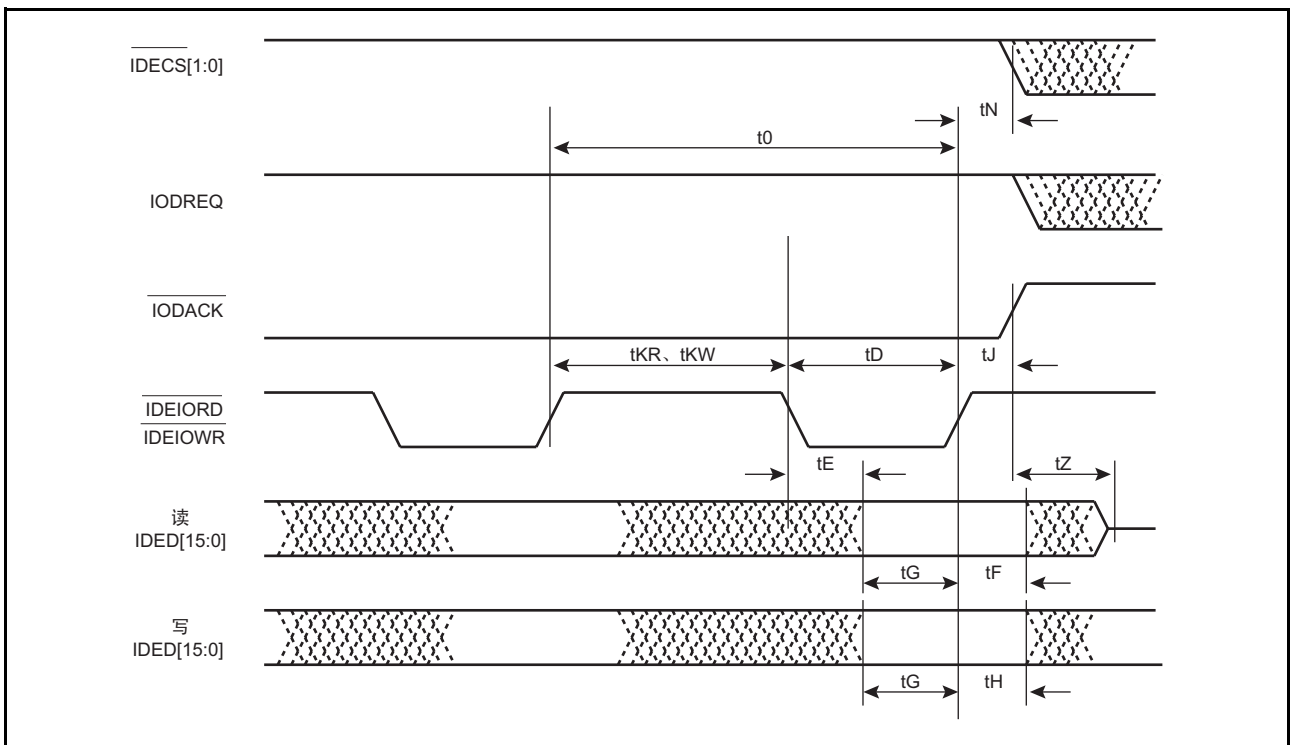


图 33.42 主机的多字 DMA 数据的传送结束

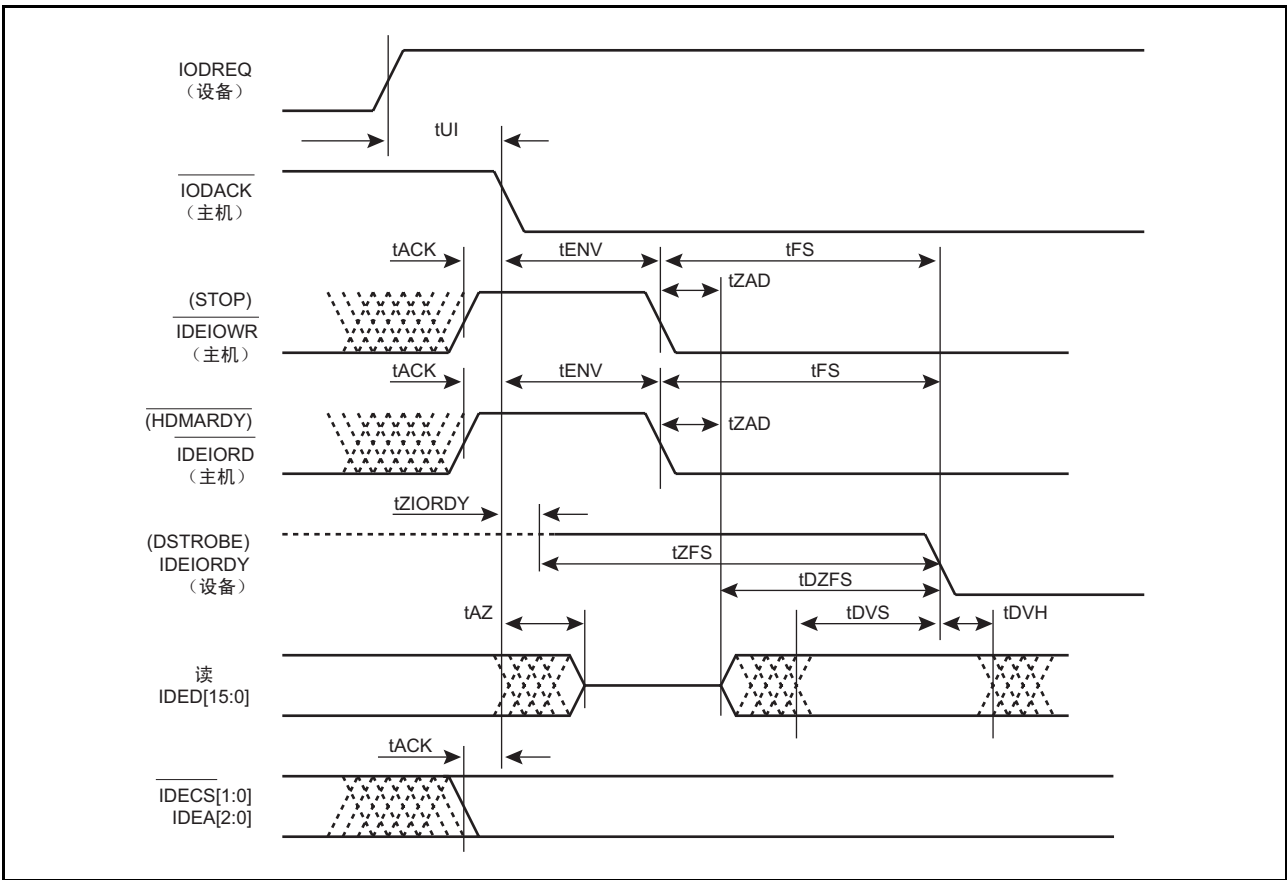


图 33.43 UltraDMA 数据的输入突发开始

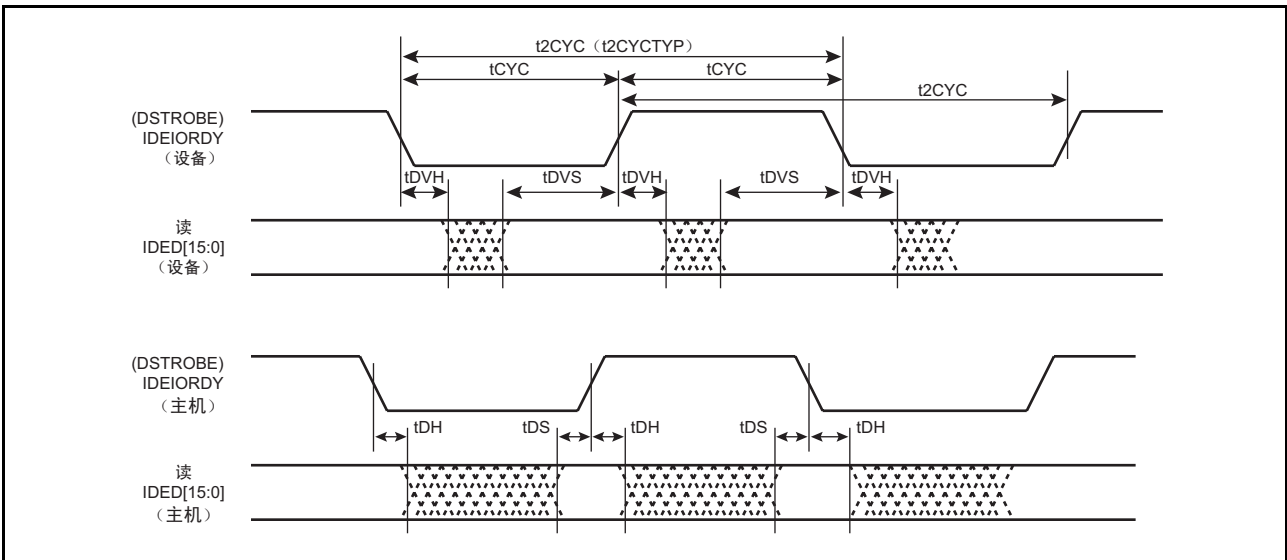


图 33.44 UltraDMA 数据的输入突发

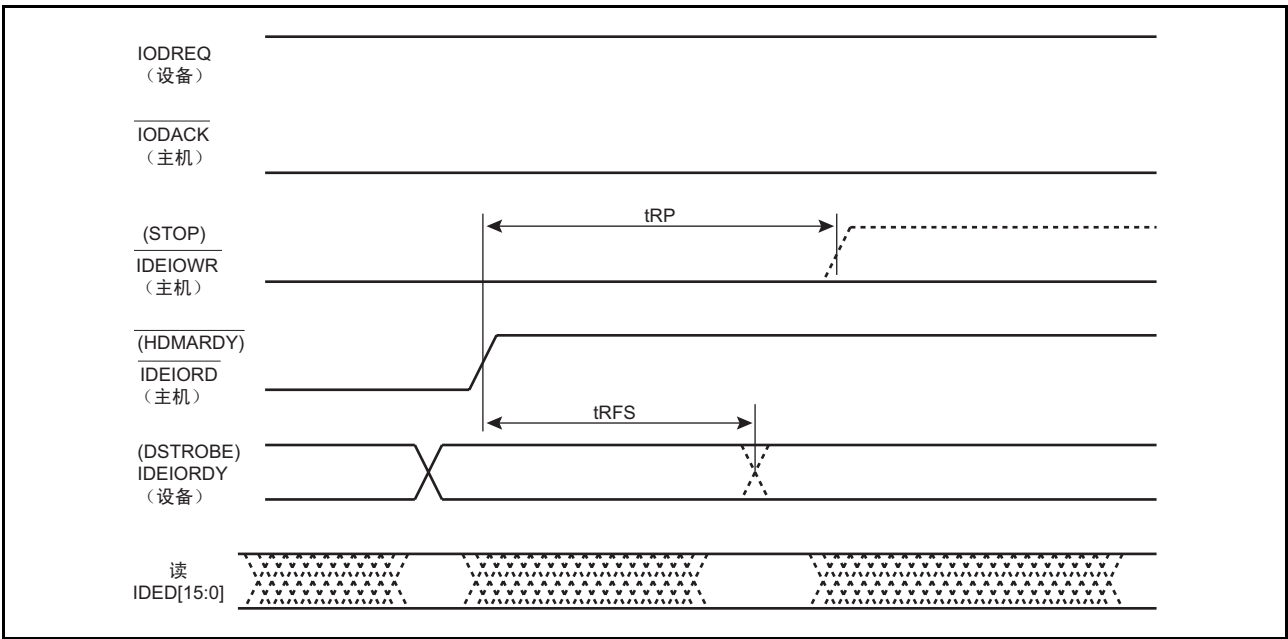


图 33.45 主机的 UltraDMA 数据的输入突发暂停

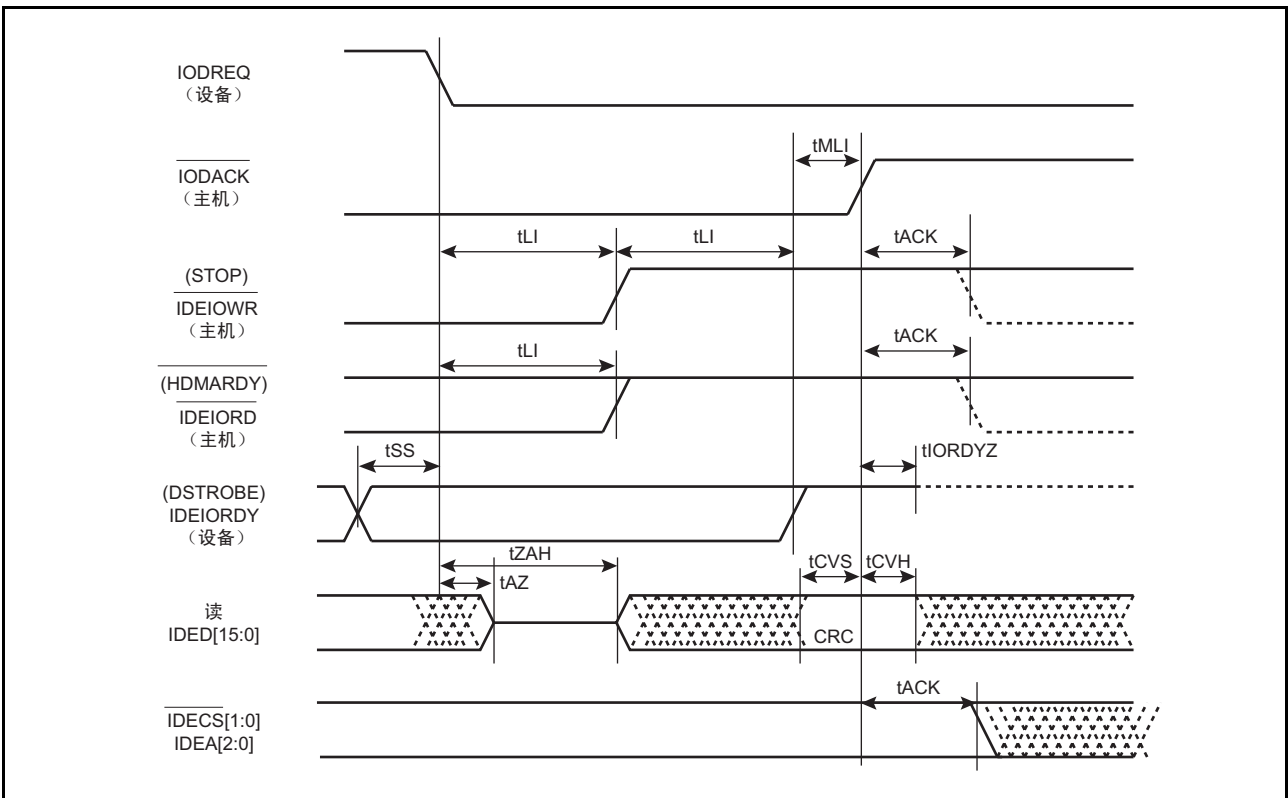


图 33.46 设备的 UltraDMA 数据的输入突发结束

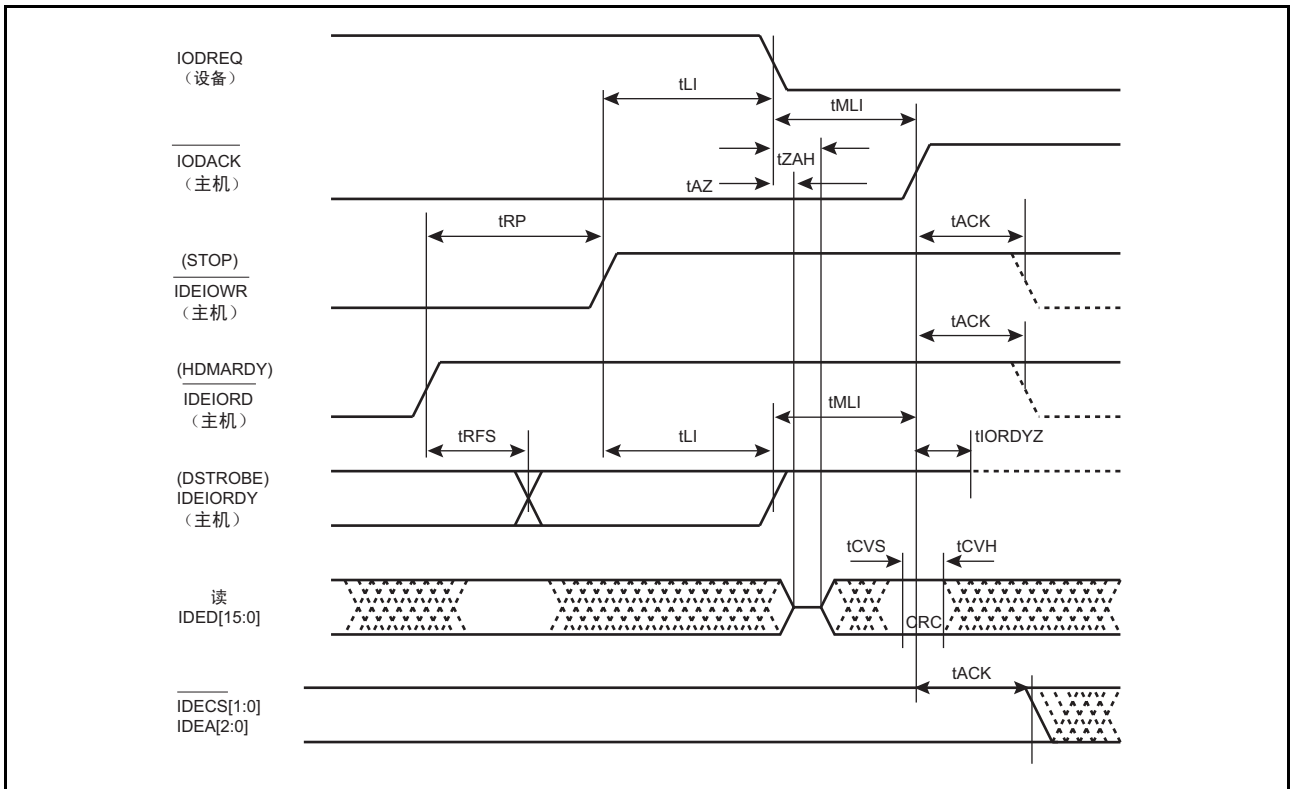


图 33.47 主机的 UltraDMA 数据的输入突发结束

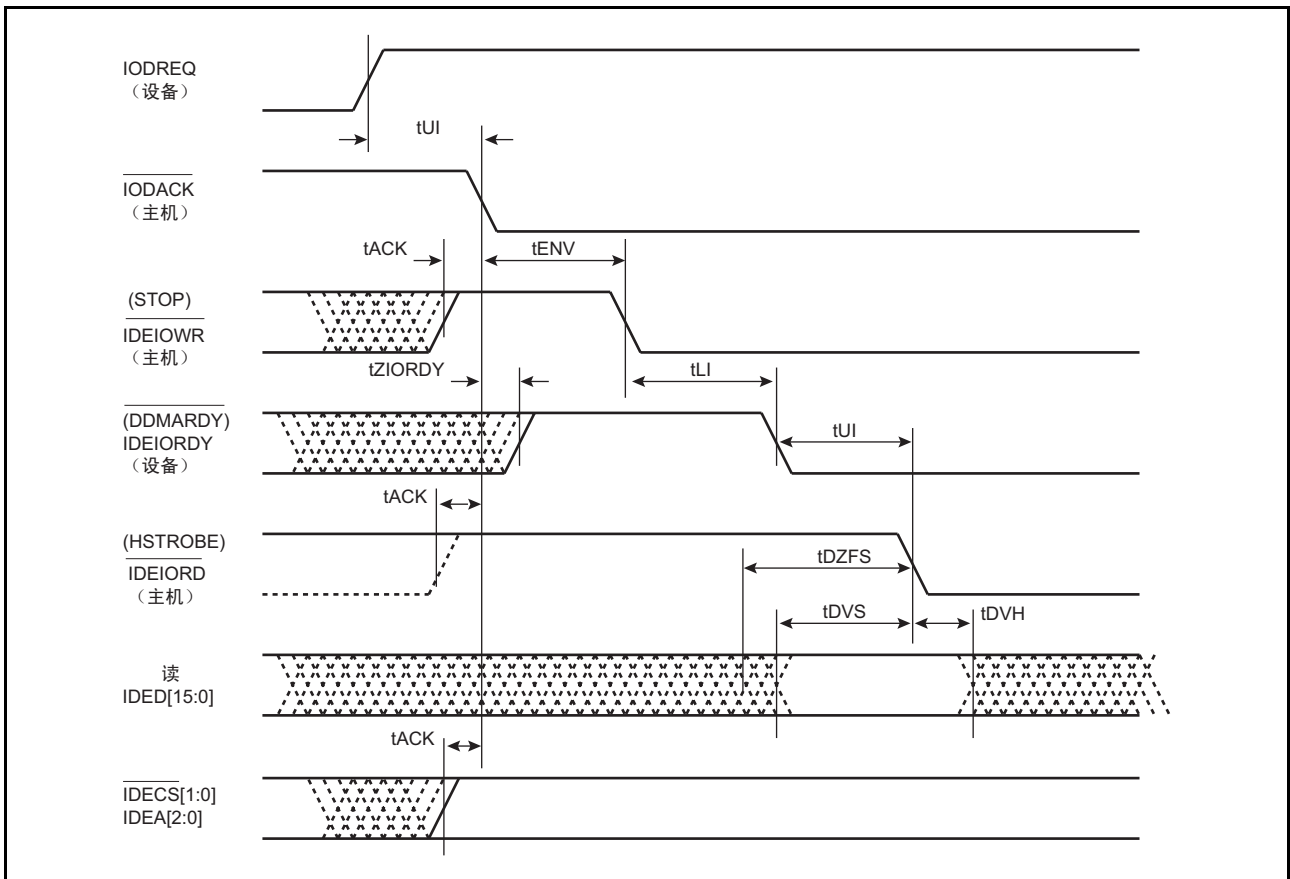


图 33.48 UltraDMA 数据的输出突发开始

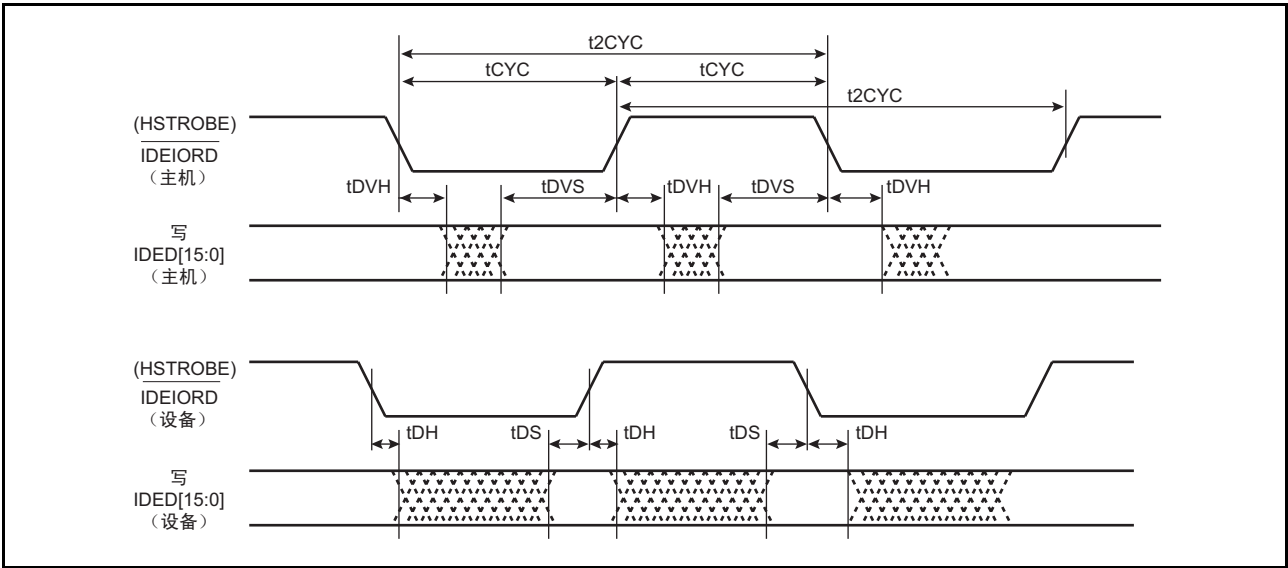


图 33.49 UltraDMA 数据的输出突发

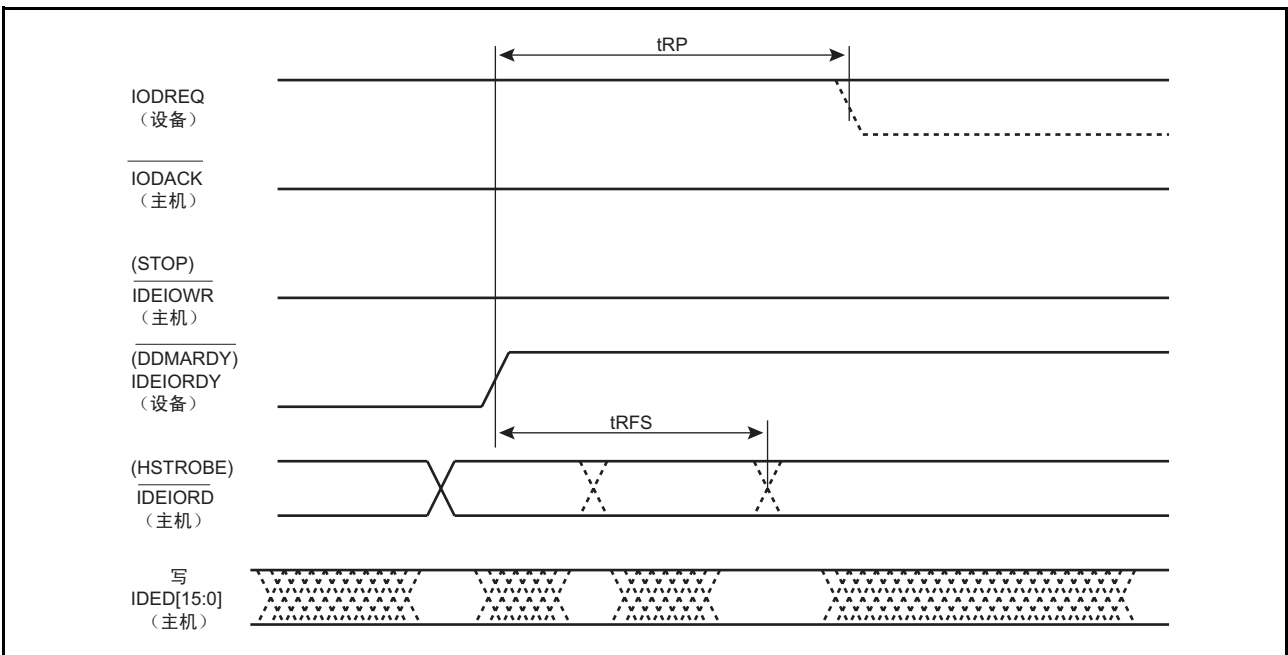


图 33.50 设备的 UltraDMA 数据的输出突发暂停

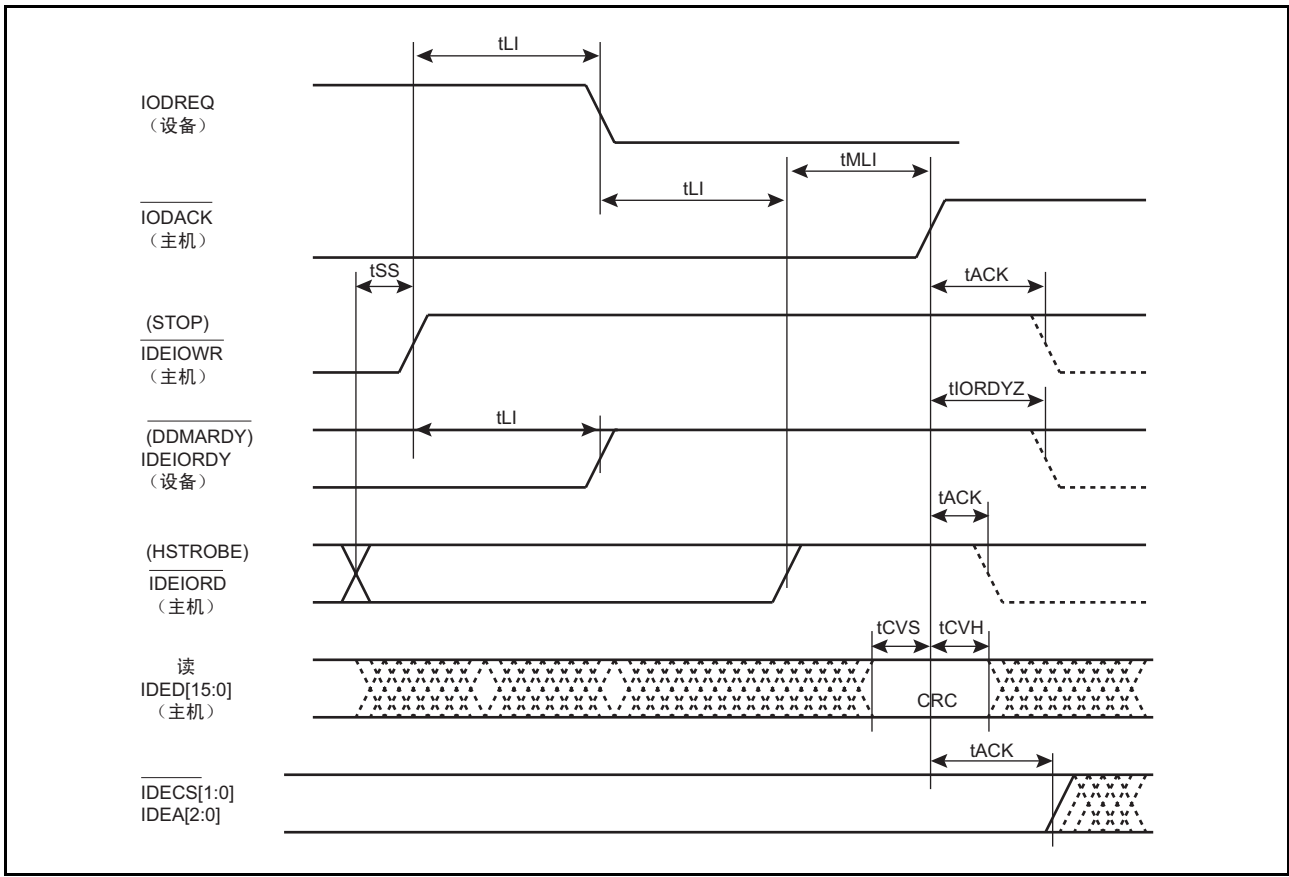


图 33.51 主机的 UltraDMA 数据的输出突发结束

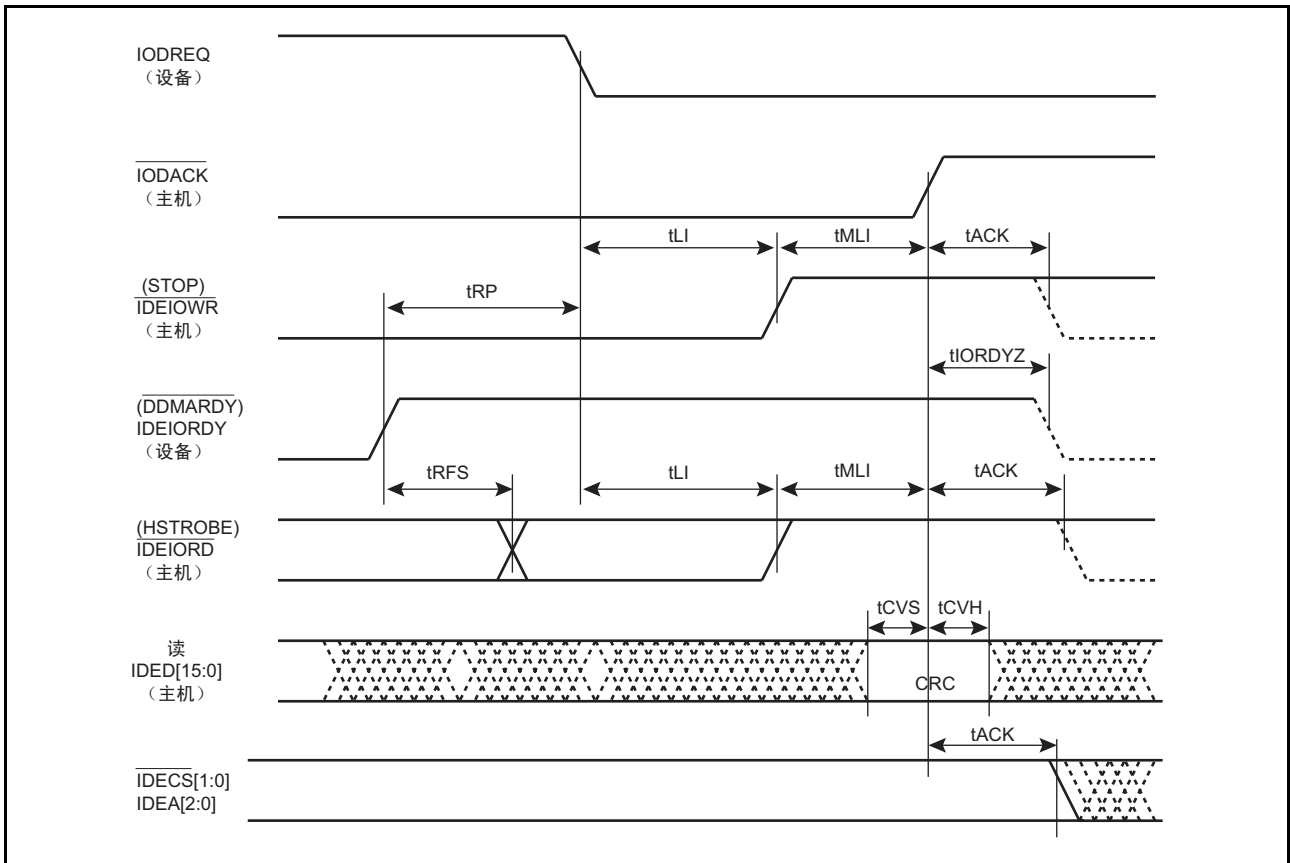


图 33.52 设备的 UltraDMA 数据的输出突发结束

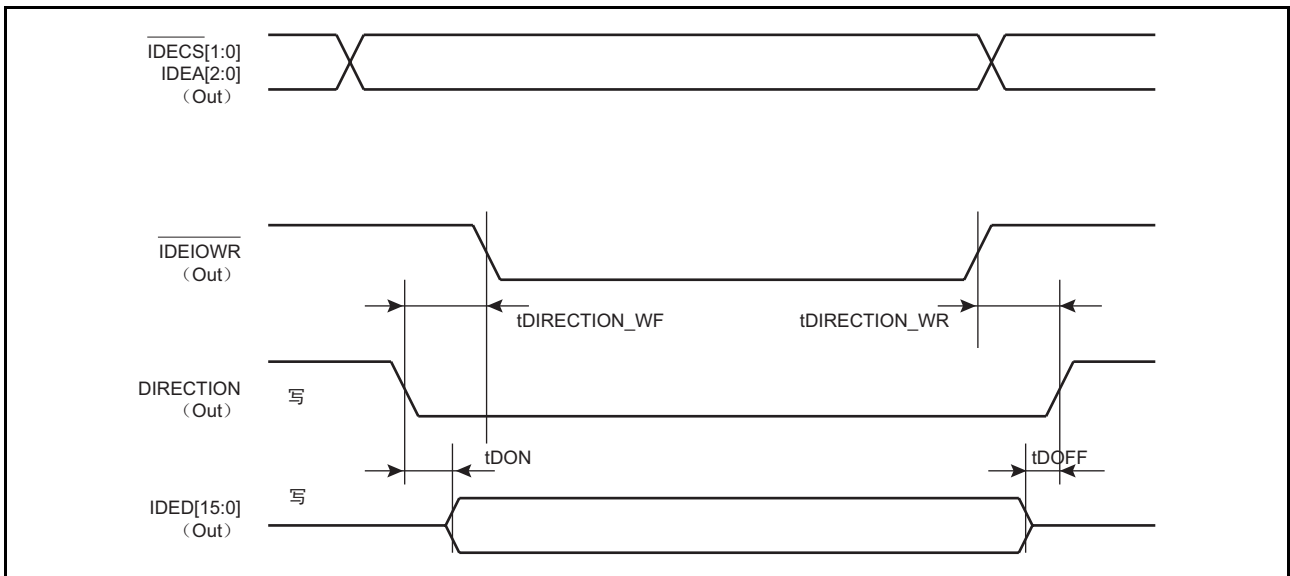


图 33.53 PIO 数据传送到设备 (DIRECTION)

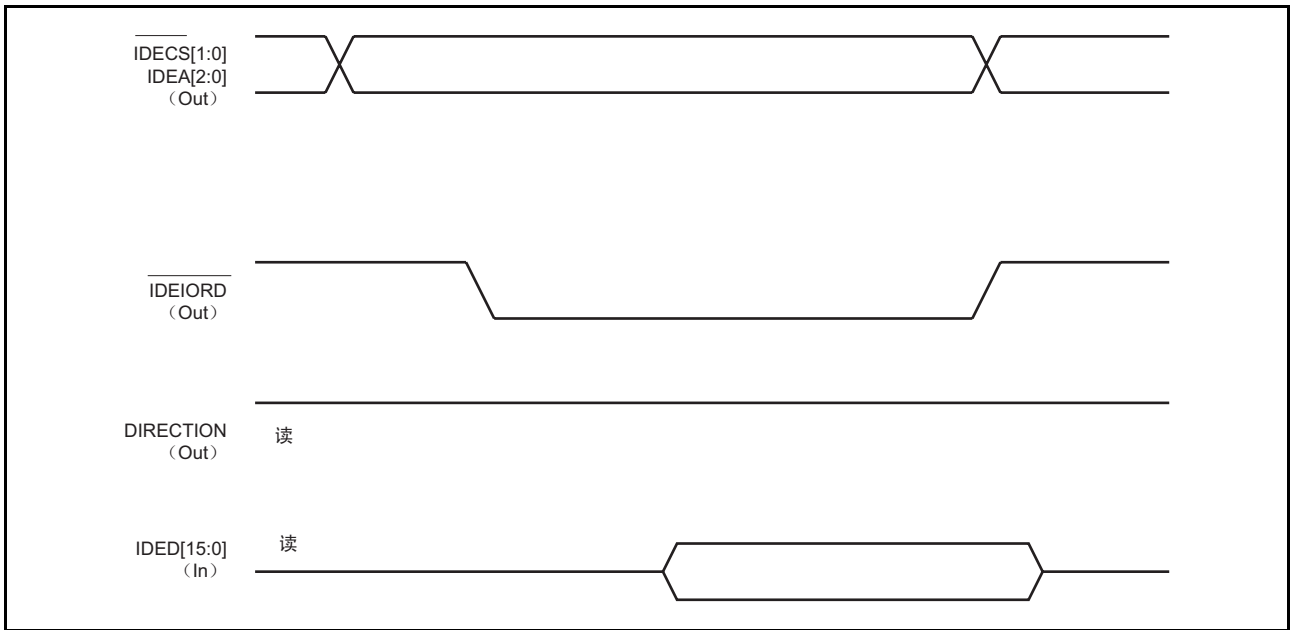


图 33.54 从设备传送 PIO 数据 (DIRECTION)

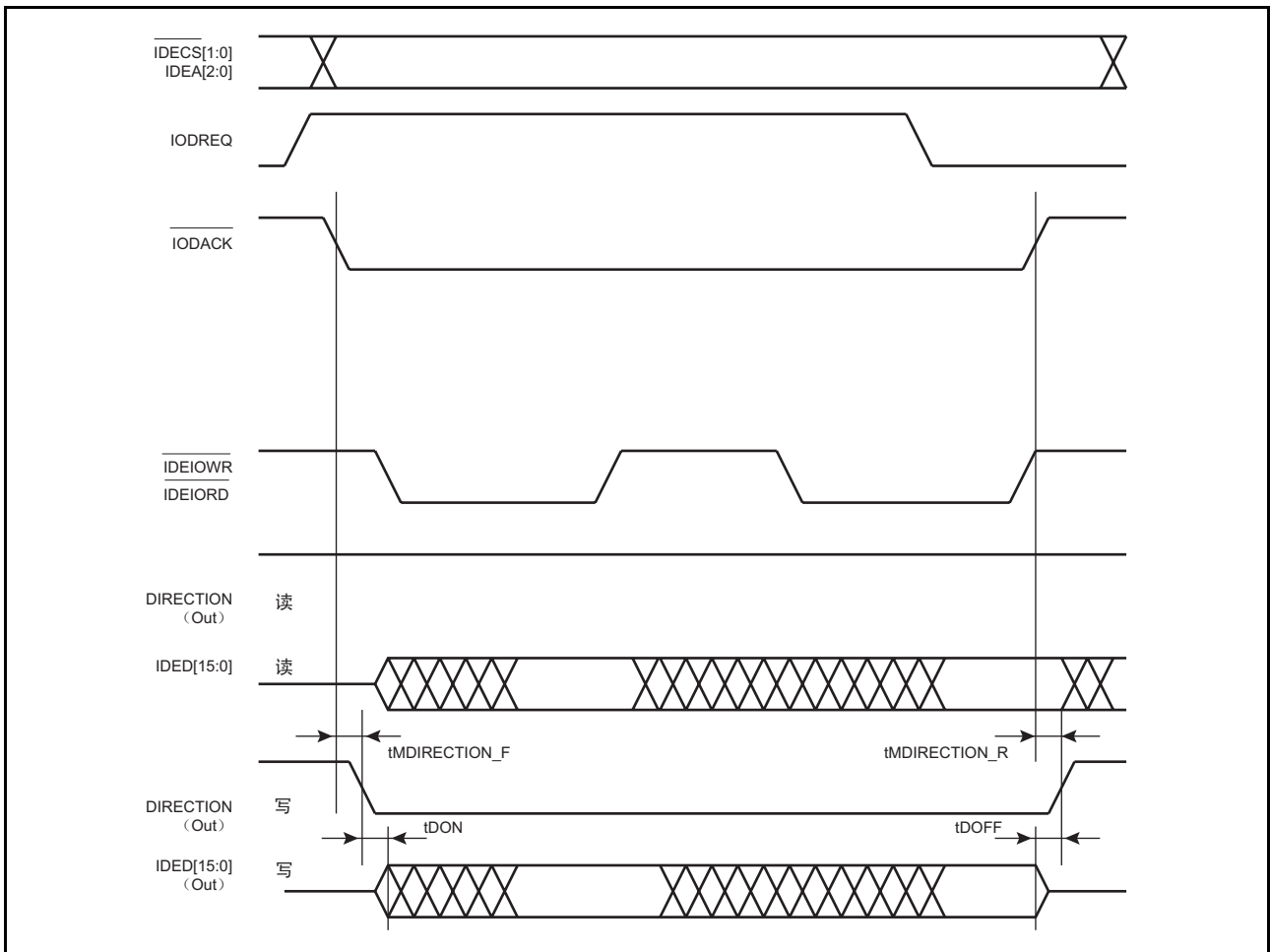


图 33.55 多字 DMA 传送 (DIRECTION)

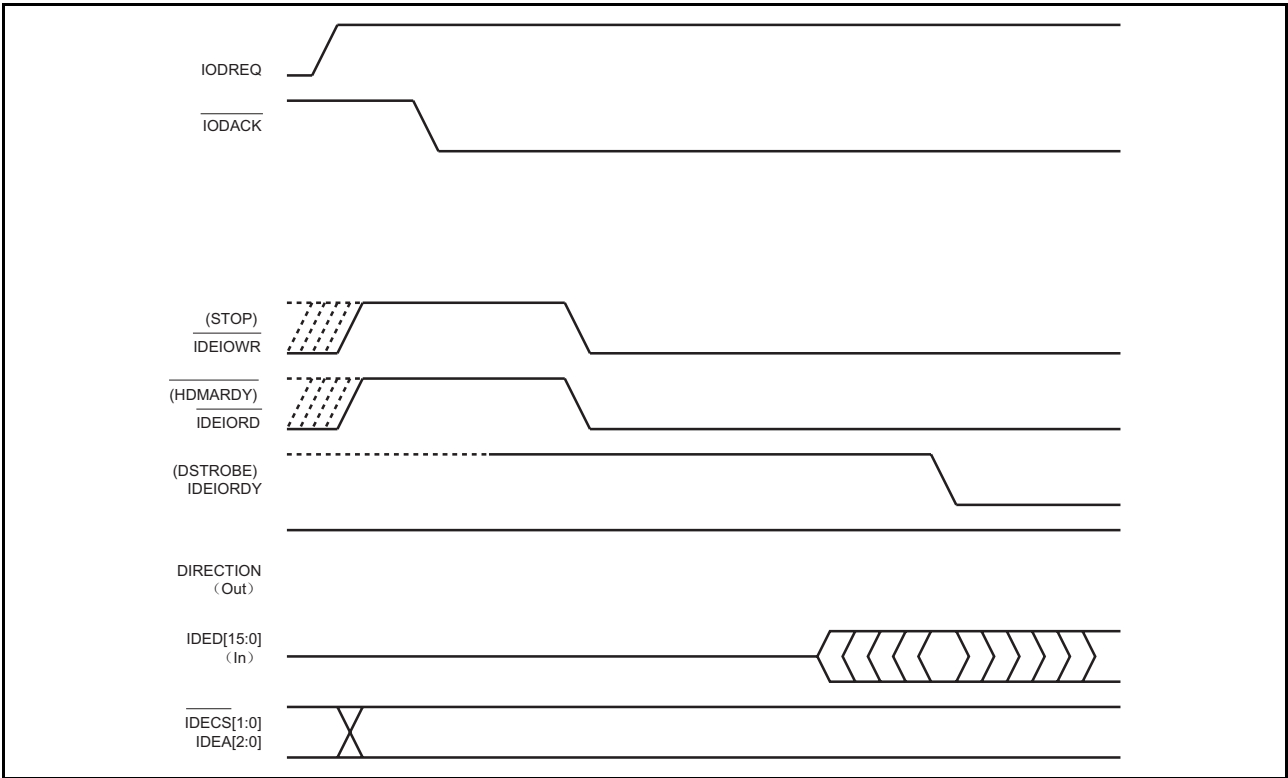


图 33.56 UltraDMA 传送数据的输入突发开始 (DIRECTION)

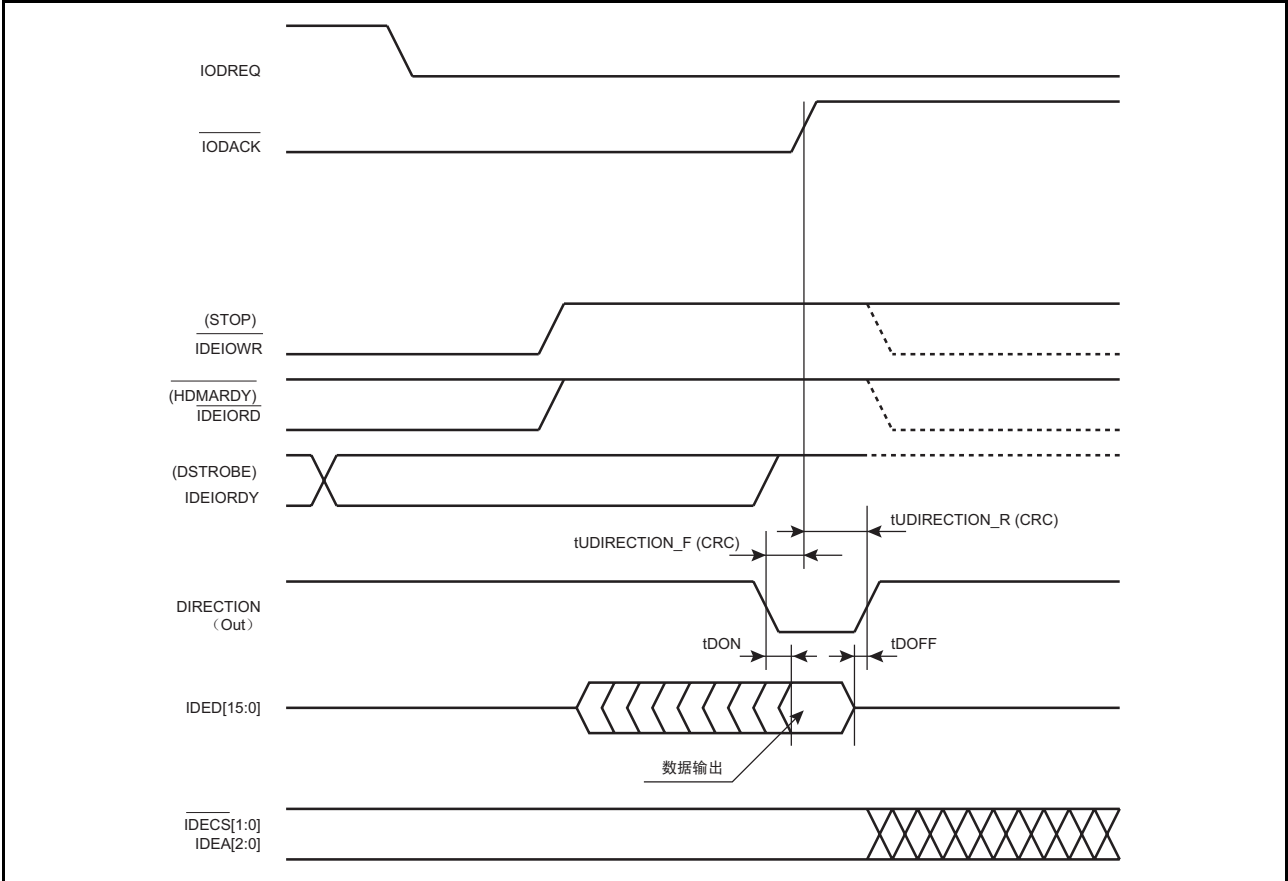


图 33.57 设备的 UltraDMA 传送数据的输入突发结束 (DIRECTION)

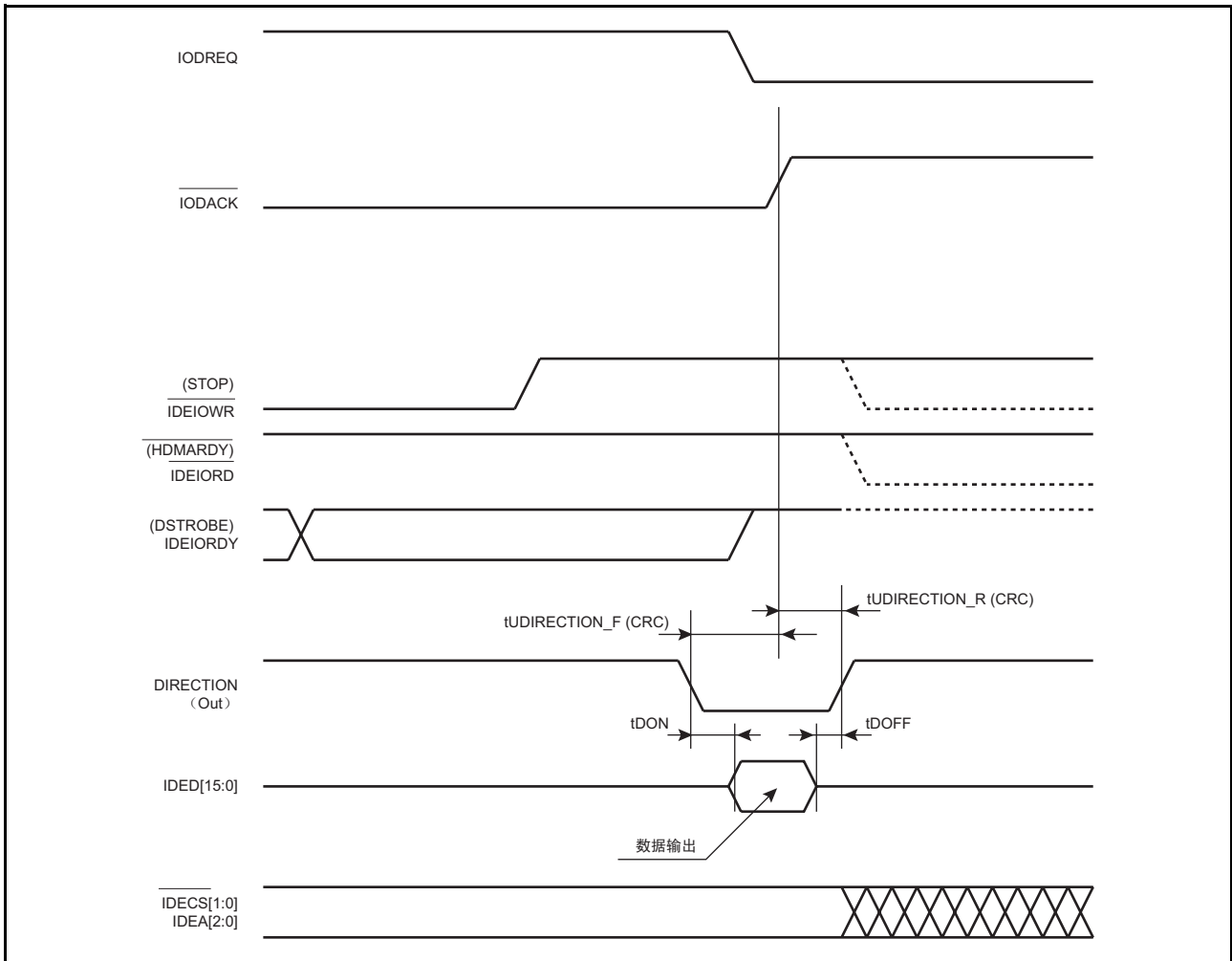


图 33.58 主机的 UltraDMA 传送数据的输入突发结束 (DIRECTION)

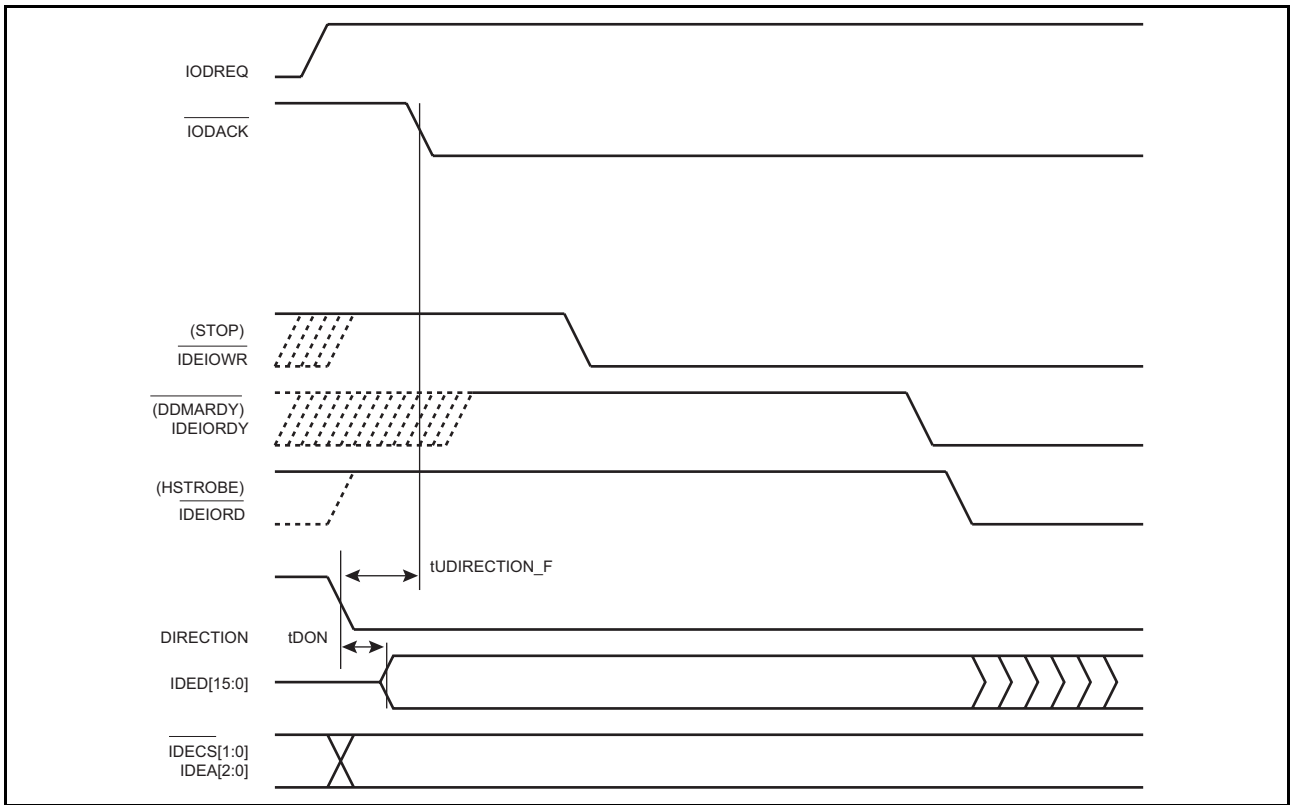


图 33.59 UltraDMA 传送数据的输出突发开始 (DIRECTION)

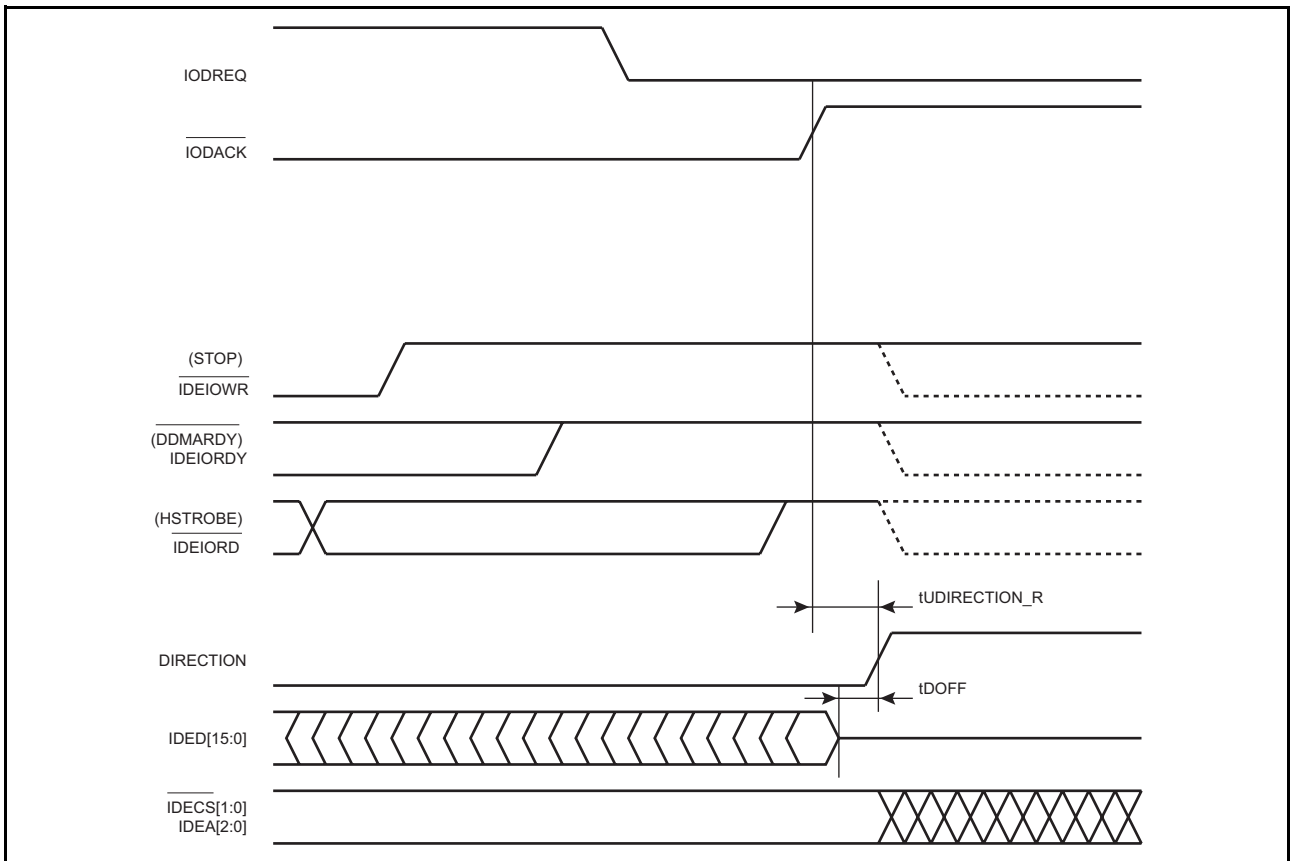


图 33.60 主机的 UltraDMA 传送数据的输出突发结束 (DIRECTION)

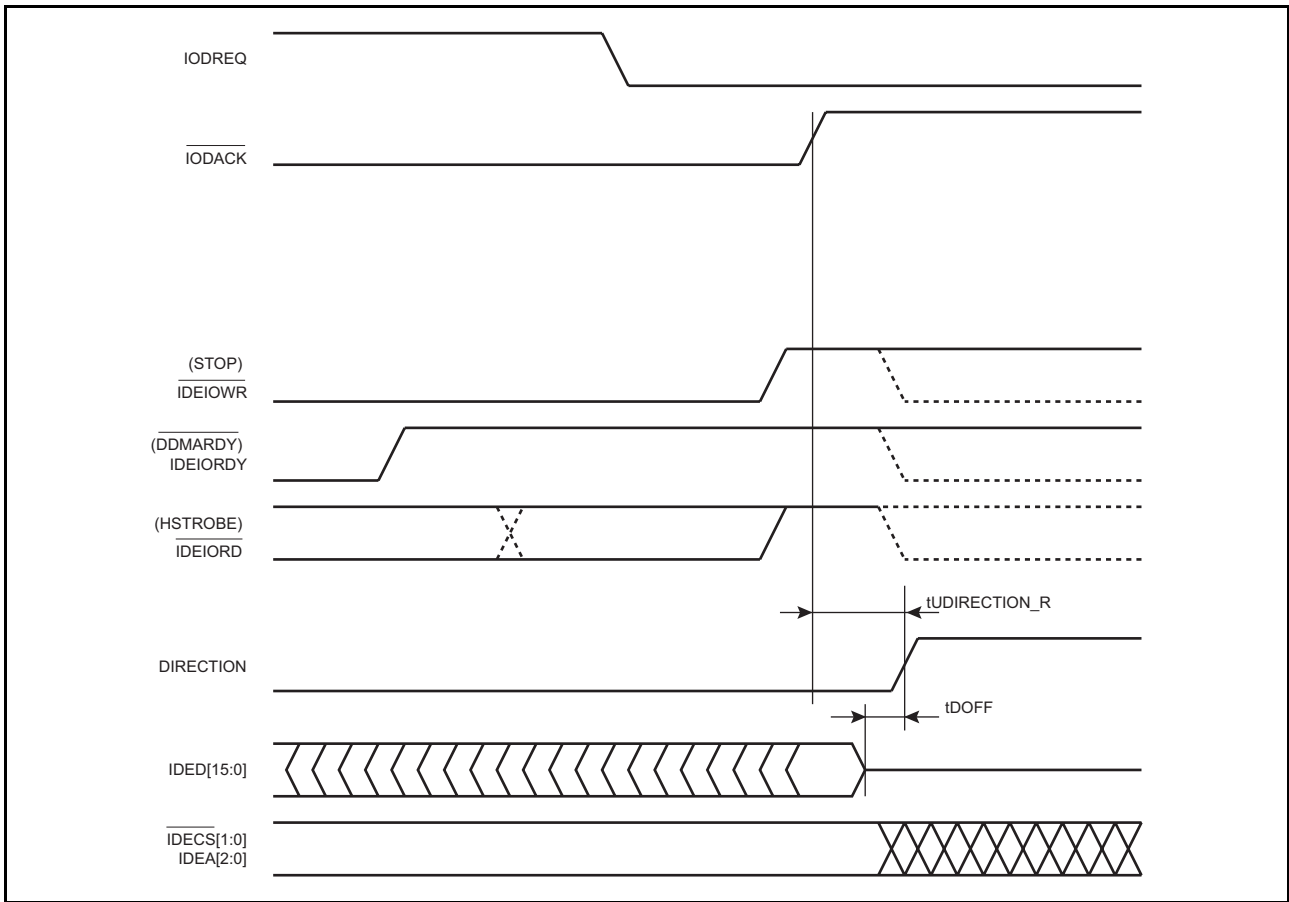


图 33.61 设备的 UltraDMA 传送数据的输出突发结束 (DIRECTION)

33.4.11 USB 模块的信号时序

表 33.24 USB 模块的时钟时序

条件：3.3V 系列电源 =3.0 ~ 3.6V、1.2V 系列电源 =1.15 ~ 1.35V、 $T_a=-20 \sim 85^{\circ}\text{C}$ 、 $-40 \sim 85^{\circ}\text{C}$

项目	符号	Min	Max	单位	参照图
USB_CLK 外部输入的时钟频率 (48MHz)	t_{FREQ}	47.9	48.1	MHz	33.62
时钟上升时间	t_{R48}	—	2	ns	
时钟下降时间	t_{F48}	—	2	ns	
占空比 ($t_{\text{HIGH}}/t_{\text{LOW}}$)	t_{DUTY}	90	110	%	

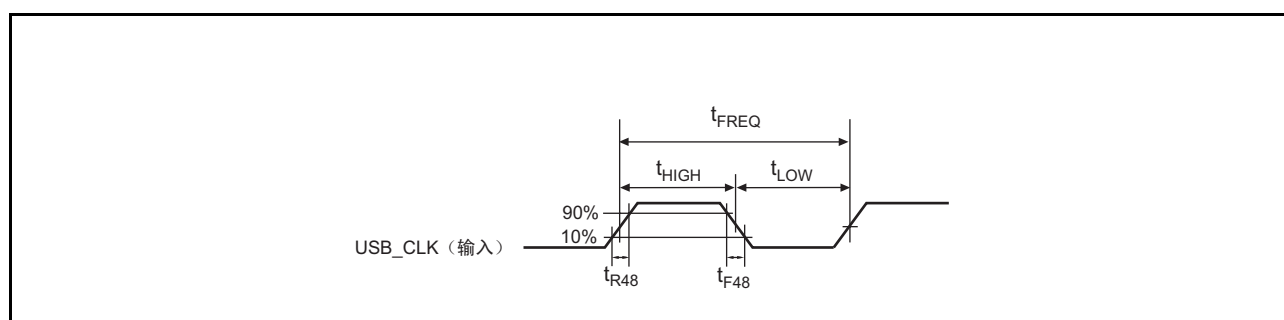


图 33.62 USB 时钟时序

表 33.25 USB 电特性 (全速)

项目	符号	Min	Max	单位	状态 *1
转移时间 (上升) *2	t_{R}	4	20	ns	CL=50pF
转移时间 (下降) *2	t_{F}	4	20	ns	CL=50pF
上升 / 下降时间的匹配	t_{RFM}	90	111	%	(TR/TF)
输出信号的交叉电压	V_{CRS}	1.3	2.0	V	—

【注】 是连接用于边沿控制的 $C_{\text{EDGE}}=47\text{pF}$ 和串联电阻 $R_s=45\Omega$ 时的测量值。

*1 在没有指定的情况下，是 CL=50pF 条件的测量值。

*2 是信号电压 10% ~ 90% 内的测量值。

表 33.26 USB 电特性 (低速)

项目	符号	Min	Max	单位	状态
转移时间 (上升) *	t_{R}	75	300	ns	
转移时间 (下降) *	t_{F}	75	300	ns	
上升 / 下降时间的匹配	t_{RFM}	80	125	%	(TR/TF)
输出信号的交叉电压	V_{CRS}	1.3	2.0	V	—

【注】 是连接用于边沿控制的 $C_{\text{EDGE}}=47\text{pF}$ 和串联电阻 $R_s=22\Omega$ 时的测量值。

* 是信号电压 10% ~ 90% 内的测量值。

33.4.12 GPIO 信号时序

表 33.27 GPIO 信号时序

条件：3.3V 系列电源 =3.0 ~ 3.6V、1.2V 系列电源 =1.15 ~ 1.35V、 $T_a=-20 \sim 85^{\circ}\text{C}$ 、 $-40 \sim 85^{\circ}\text{C}$

项目	符号	Min	Max	单位	参照图
GPIO 输出的延迟时间	t_{IOPD}	—	17	ns	33.63
GPIO 输入的准备时间	t_{IOPS}	17	—	ns	
GPIO 输入的保持时间	t_{IOPH}	$t_{CLKOUTcyc}$	—	ns	

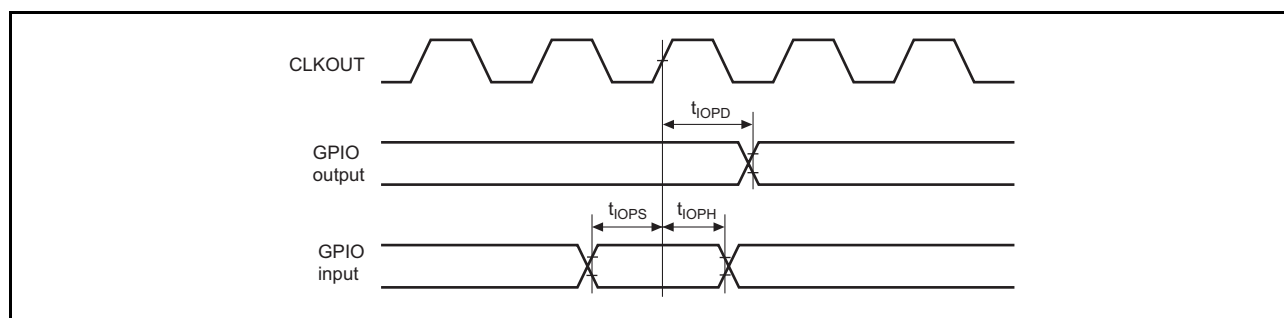


图 33.63 GPIO 时序

33.4.13 H-UDI 模块的信号时序

表 33.28 H-UDI 模块的信号时序

条件：3.3V 系列电源 =3.0 ~ 3.6V、1.2V 系列电源 =1.15 ~ 1.35V、 $T_a=-20 \sim 85^{\circ}\text{C}$ 、 $-40 \sim 85^{\circ}\text{C}$

模块	项目	符号	Min.	Max.	单位	参照图
H-UDI	输入时钟周期	t_{TCKcyc}	50	—	ns	33.64、33.66
	输入时钟脉宽 (High)	t_{TCKH}	15	—	ns	33.64
	输入时钟脉宽 (Low)	t_{TCKL}	15	—	ns	33.64
	输入时钟的上升时间	t_{TCKr}	—	10	ns	33.64
	输入时钟的下降时间	t_{TCKf}	—	10	ns	33.64
	ASEBRKAK/BRKACK 准备时间	$t_{ASEBRKS}$	10	—	t_{cyc}	33.65
	ASEBRKAK/BRKACK 保持时间	$t_{ASEBRKH}$	10	—	t_{cyc}	33.65
	TDI/TMS 准备时间	t_{TDIS}	15	—	ns	33.66
	TDI/TMS 保持时间	t_{TDIH}	15	—	ns	33.66
	TDO 数据延迟时间	t_{TDO}	0	15	ns	33.66
	ASEBRKAK/BRKACK 脉宽	t_{PINBRK}	2	—	t_{Pcyc}	33.67

- 【注】 1. t_{cyc} 表示 1 个 CLKOUT 时钟周期的时间。
2. t_{Pcyc} 表示 1 个外围时钟 (Pck) 周期的时间。

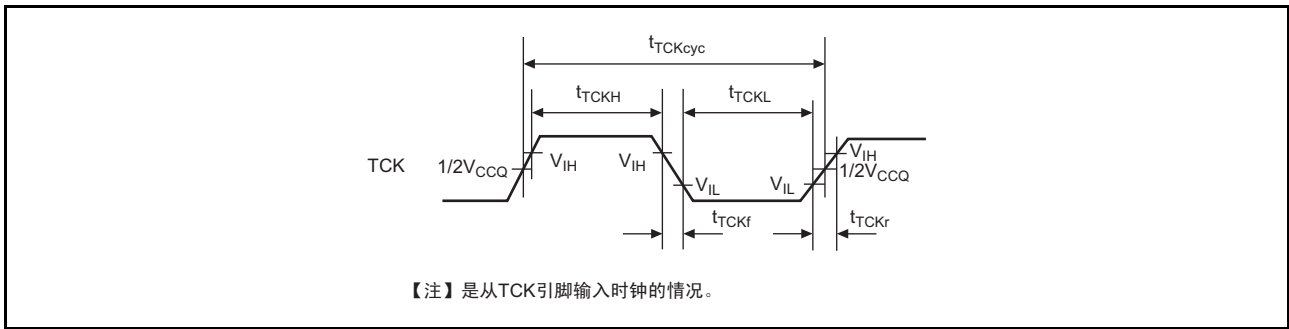


图 33.64 TCK 输入时序

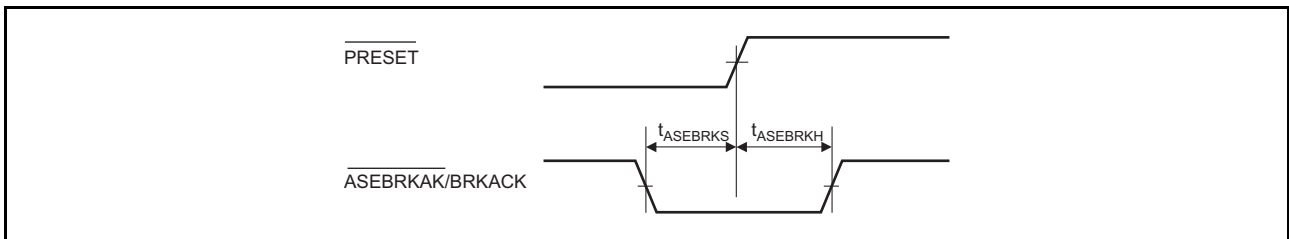


图 33.65 \overline{PRESET} 保持时序

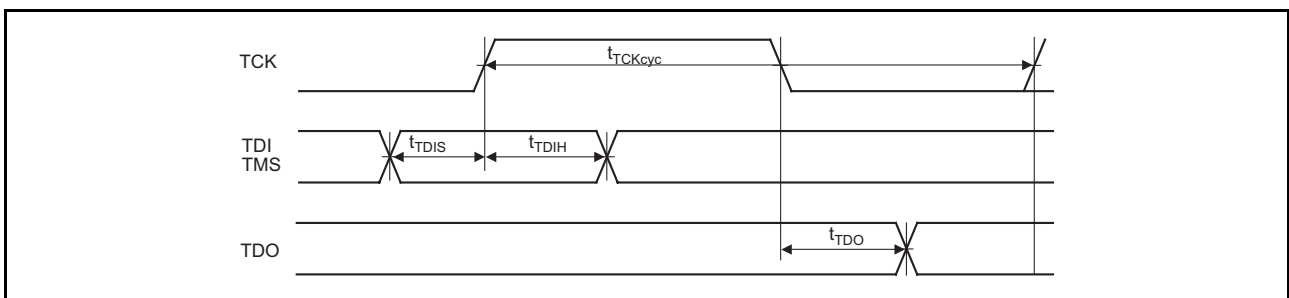


图 33.66 H-UDI 数据传送时序

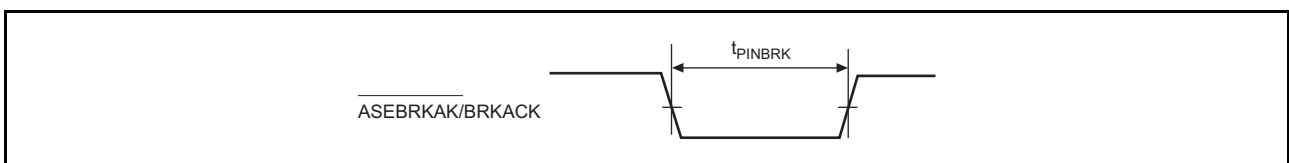


图 33.67 $\overline{ASEBRKAK/BRKACK}$ 引脚中止时序

33.4.14 EtherC 模块的信号时序

表 33.29 以太网控制器时序 (MII)

条件: 3.3V 系列电源 = 3.0 ~ 3.6V、1.2V 系列电源 = 1.15 ~ 1.35V、 $T_a = -20 \sim 85^\circ\text{C}$ 、 $-40 \sim 85^\circ\text{C}$

项目	符号	Min.	Typ.	Max.	单位	参照图
TX-CLK 周期时间	t_{TCYC}	40	—	—	ns	33.68
TX-EN 输出的延迟时间	t_{TEND}	1	—	20		
MII_TXD[3:0] 输出的延迟时间	T_{MTDD}	1	—	20	33.69	
CRS 准备时间	t_{CRSS}	10	—	—		
CRS 保持时间	t_{CRSH}	10	—	—		
COL 准备时间	t_{COLS}	10	—	—		
COL 保持时间	t_{COLH}	10	—	—		
RX-CLK 周期时间	t_{RCYC}	40	—	—		33.70
RX-DV 准备时间	t_{RDVS}	10	—	—		
RX-DV 保持时间	t_{RDVH}	10	—	—		
MII_RXD[3:0] 准备时间	T_{MRDS}	10	—	—		
MII_RXD[3:0] 保持时间	T_{MRDH}	10	—	—		33.71
RX-ER 准备时间	t_{RERS}	10	—	—		
RX-ER 保持时间	t_{RERH}	10	—	—	33.72	
MDIO 准备时间	t_{MDIOS}	10	—	—		
MDIO 保持时间	t_{MDIOH}	10	—	—	33.73	
MDIO 输出数据的保持时间*	t_{MDIODH}	5	—	18		
WOL 输出的延迟时间	T_{WOLD}	1	—	20	33.74	
EXOUT 输出的延迟时间	T_{EXOUTD}	1	—	20	33.75	

【注】* 用户需要通过程序进行满足此规定的设定。

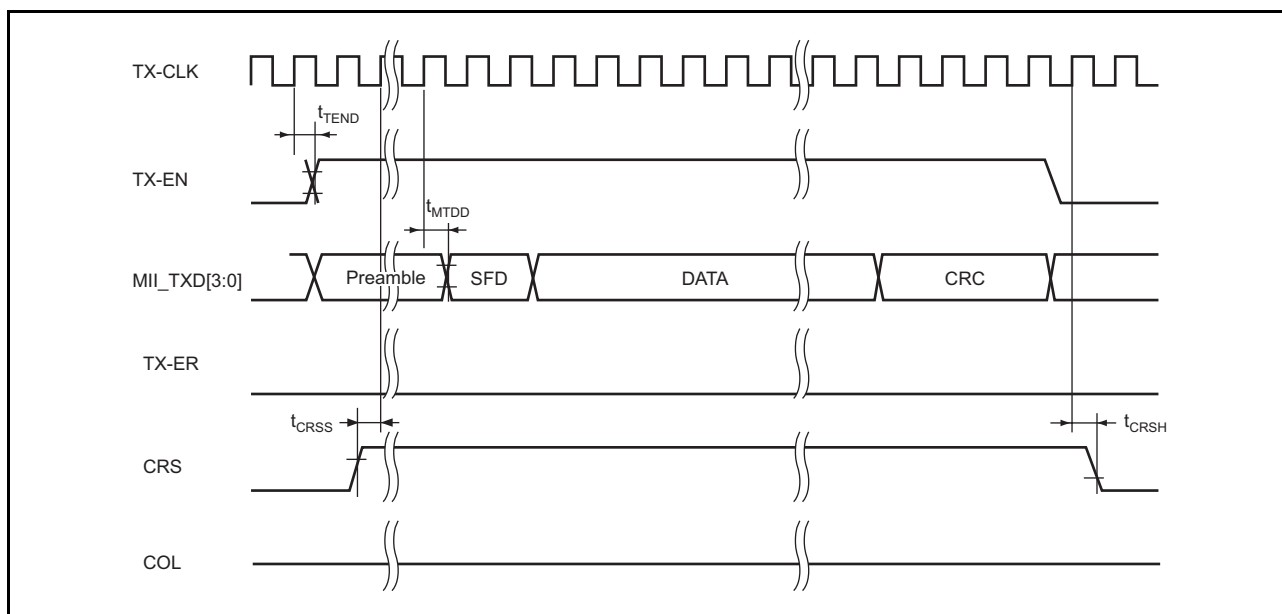


图 33.68 MII 发送时序 (正常运行时)

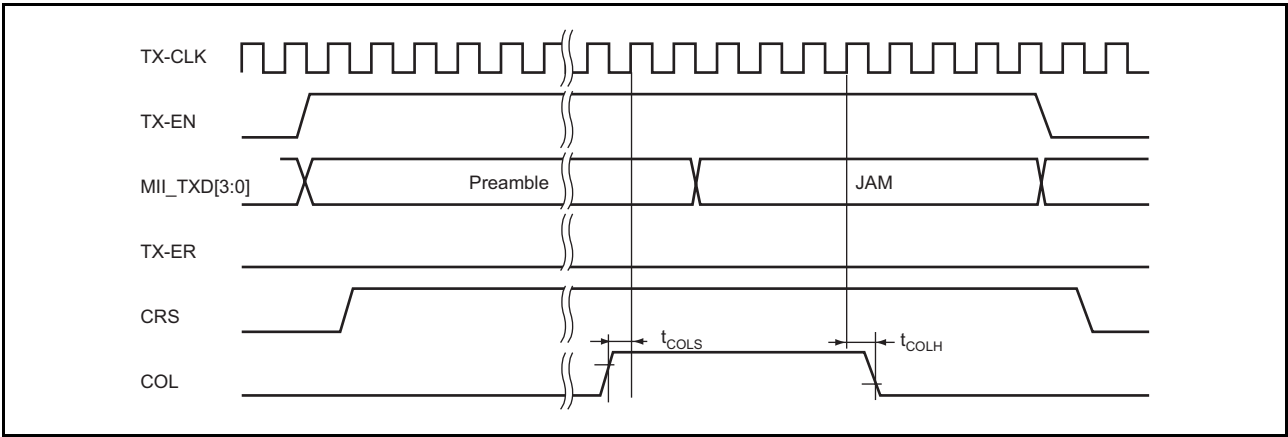


图 33.69 MII 发送时序（发生冲突时）

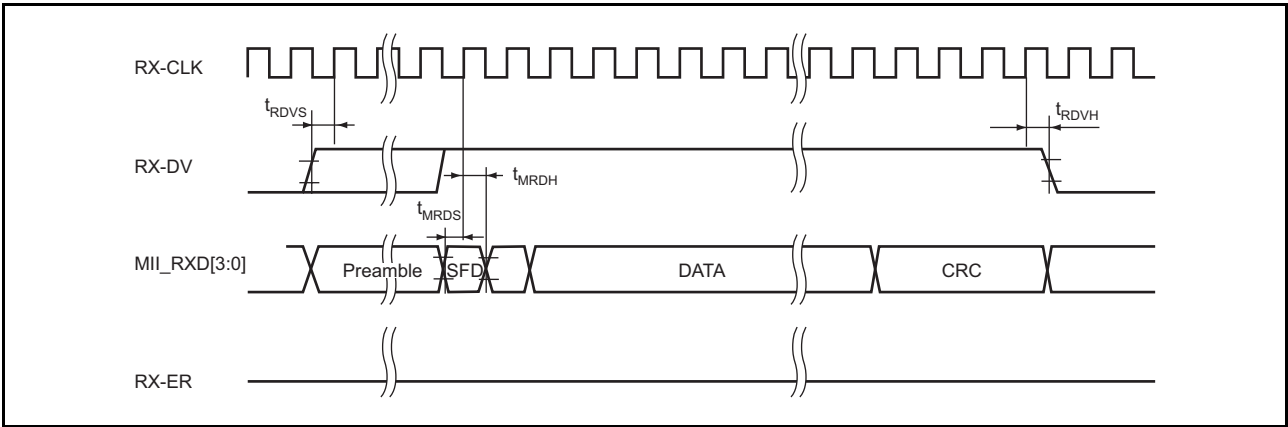


图 33.70 MII 接收时序（正常运行时）

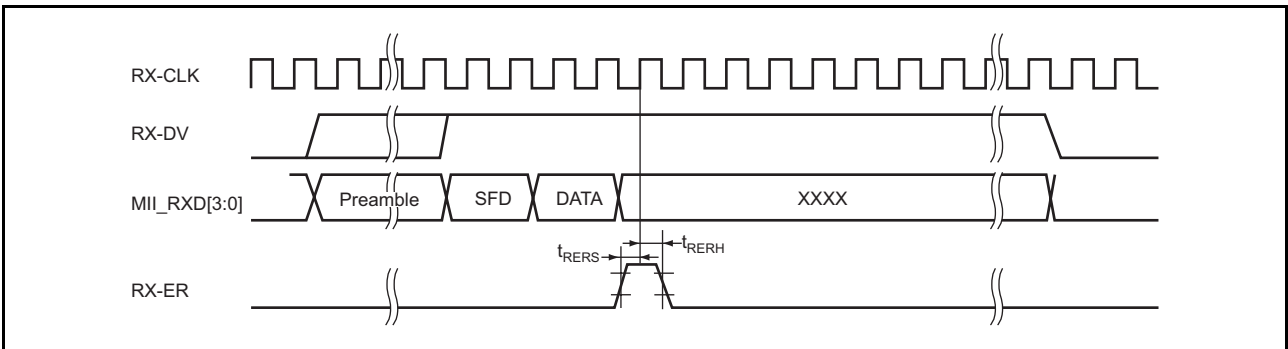


图 33.71 MII 接收时序（发生错误时）

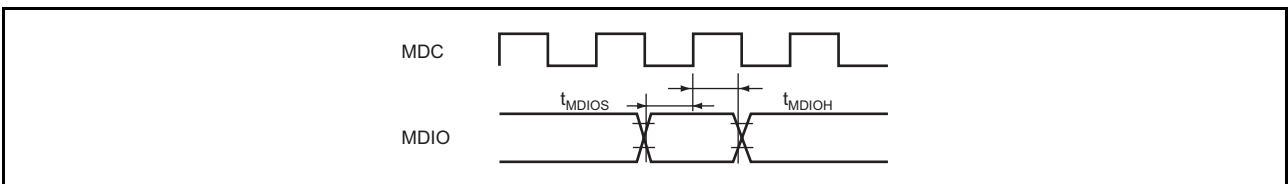


图 33.72 MDIO 输入时序

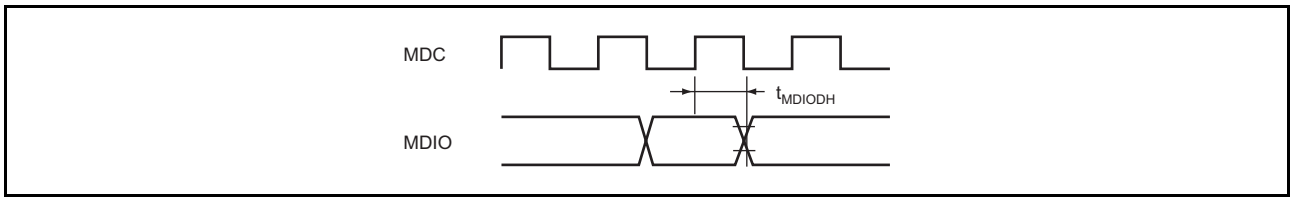


图 33.73 MDIO 输出时序

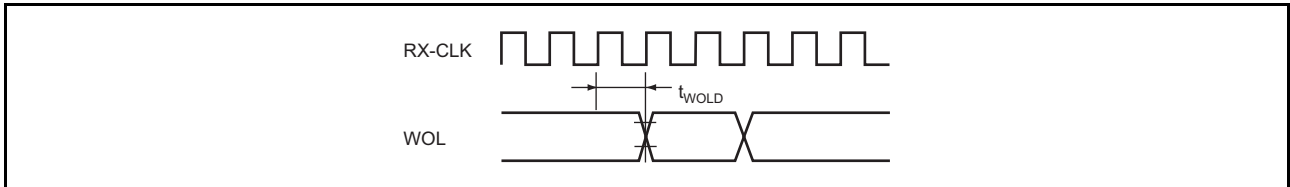


图 33.74 WOL 输出时序

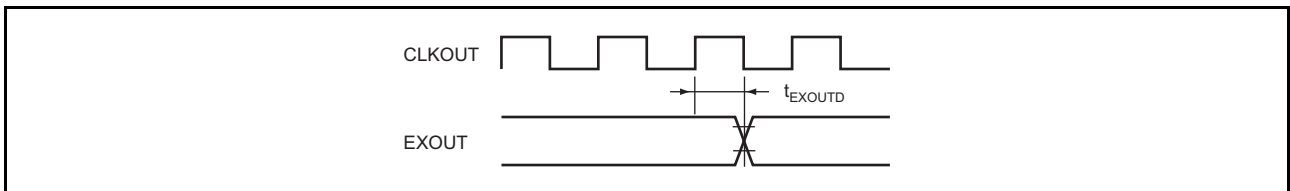


图 33.75 EXOUT 输出时序

33.4.15 FLCTL 模块的信号时序

表 33.30 NAND 型闪存的接口时序

条件：3.3V 系列电源 = 3.0 ~ 3.6V、1.2V 系列电源 = 1.15 ~ 1.35V、 $T_a = -20 \sim 85^\circ\text{C}$ 、 $-40 \sim 85^\circ\text{C}$

项目	符号	Min.	Max.	单位	参照图
命令输出的准备时间	t_{NCDS}	$2 \times t_{\text{fcyc}} - 10$	—	ns	33.76、33.80
命令输出的保持时间	t_{NCDH}	$1.5 \times t_{\text{fcyc}} - 5$	—	ns	
数据输出的准备时间	t_{NDOS}	$0.5 \times t_{\text{fcyc}} - 5$	—	ns	33.76、33.77、 33.79、33.80
数据输出的保持时间	t_{NDOH}	$0.5 \times t_{\text{fcyc}} - 10$	—	ns	
命令 - 地址的转移时间 1	t_{NCDAD1}	$1.5 \times t_{\text{fcyc}} - 10$	—	ns	33.76、33.77
命令 - 地址的转移时间 2	t_{NCDAD2}	$2 \times t_{\text{fcyc}} - 10$	—	ns	33.77
$\overline{\text{FWE}}$ 周期时间	t_{NWC}	$t_{\text{fcyc}} - 5$	—	ns	33.77、33.79
$\overline{\text{FWE}}$ 低电平脉宽	t_{NWP}	$0.5 \times t_{\text{fcyc}} - 5$	—	ns	33.76、33.77、 33.79、33.80
$\overline{\text{FWE}}$ 高电平脉宽	t_{NWH}	$0.5 \times t_{\text{fcyc}} - 5$	—	ns	
地址 - 就绪 / 忙 - 转移时间	t_{NADRB}	—	$32 \times t_{\text{pcyc}}$	ns	33.77、33.78
就绪 / 忙 - 读数据的转移时间 1	t_{NRBDR1}	$1.5 \times t_{\text{fcyc}}$	—	ns	33.78
就绪 / 忙 - 读数据的转移时间 2	t_{NRBDR2}	$32 \times t_{\text{pcyc}}$	—	ns	
$\overline{\text{FRE}}$ 周期时间	t_{NSCC}	$t_{\text{fcyc}} - 5$	—	ns	
$\overline{\text{FRE}}$ 低电平脉宽	t_{NSP}	$0.5 \times t_{\text{fcyc}} - 5$	—	ns	33.78、33.80
$\overline{\text{FRE}}$ 高电平脉宽	t_{NSPH}	$0.5 \times t_{\text{fcyc}} - 5$	—	ns	33.78
读数据的准备时间	t_{NRDS}	14	—	ns	33.78、33.80
读数据的保持时间	t_{NRDH}	0	—	ns	33.78、33.80
写数据的准备时间	t_{NDWS}	$32 \times t_{\text{pcyc}}$	—	ns	33.79
命令 - 读状态的转移时间	t_{NCDSR}	$4 \times t_{\text{fcyc}}$	—	ns	33.80
命令输出 OFF - 读状态的转移时间	t_{NCDFSR}	$3.5 \times t_{\text{fcyc}}$	—	ns	
读状态的准备时间	t_{NSTS}	$2.5 \times t_{\text{fcyc}}$	—	ns	

【注】 t_{fcyc} 为 1 个 FLCTL 时钟周期的时间。

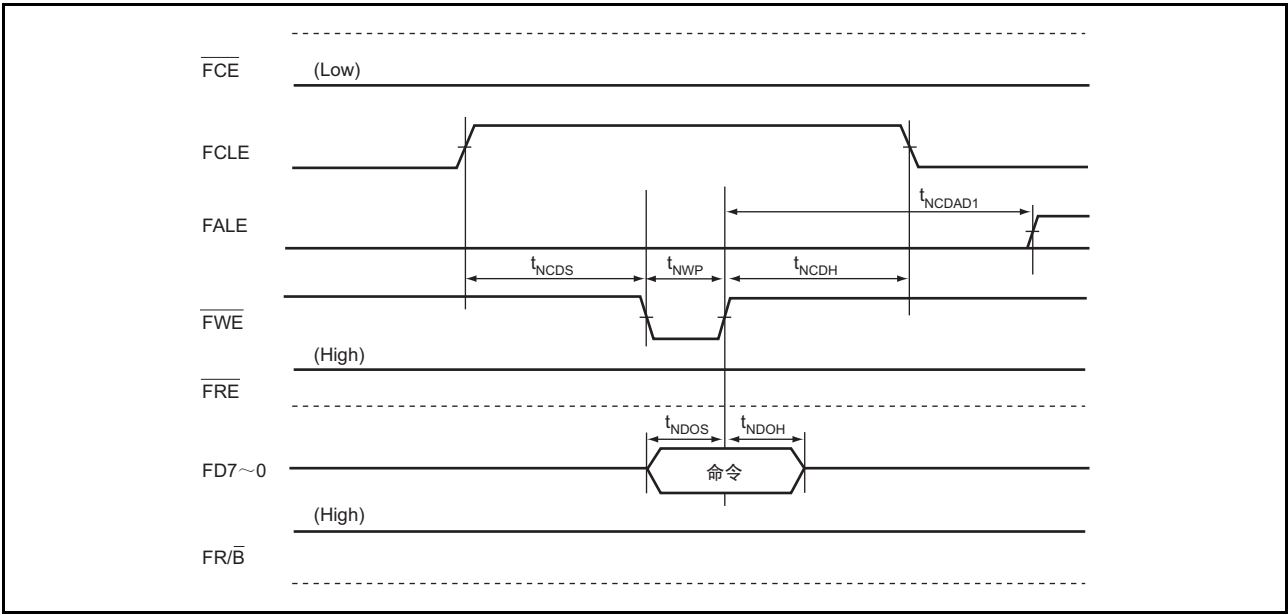


图 33.76 NAND 型闪存 的命令发行时序

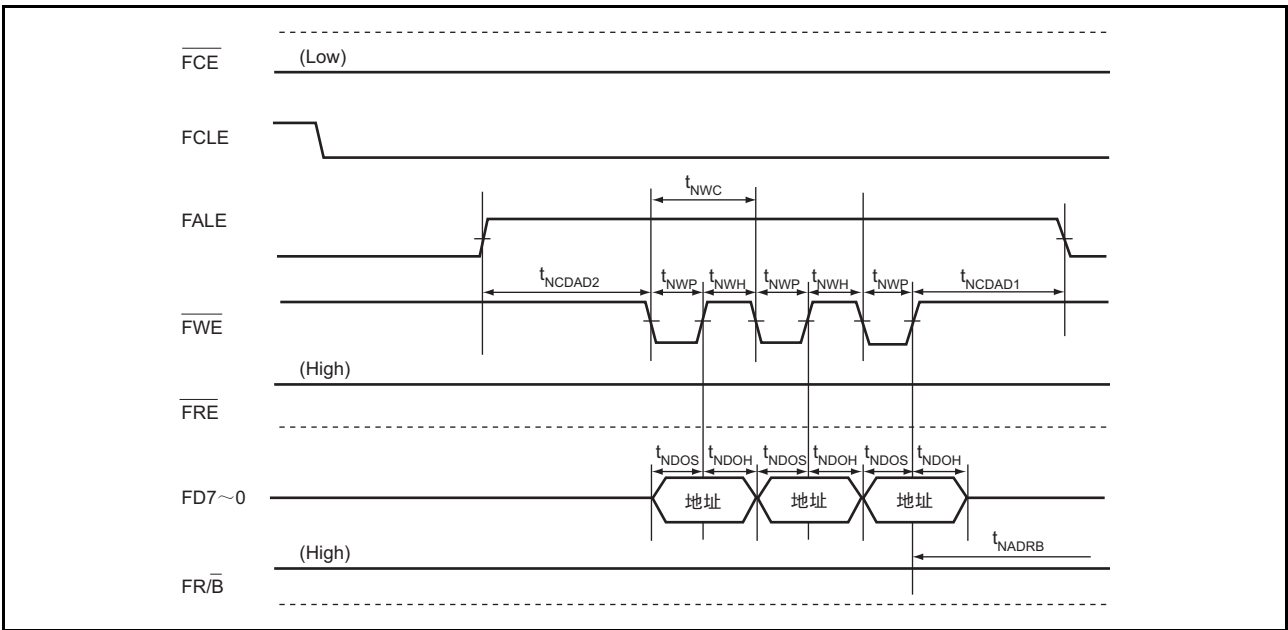


图 33.77 NAND 型闪存的地址发行时序

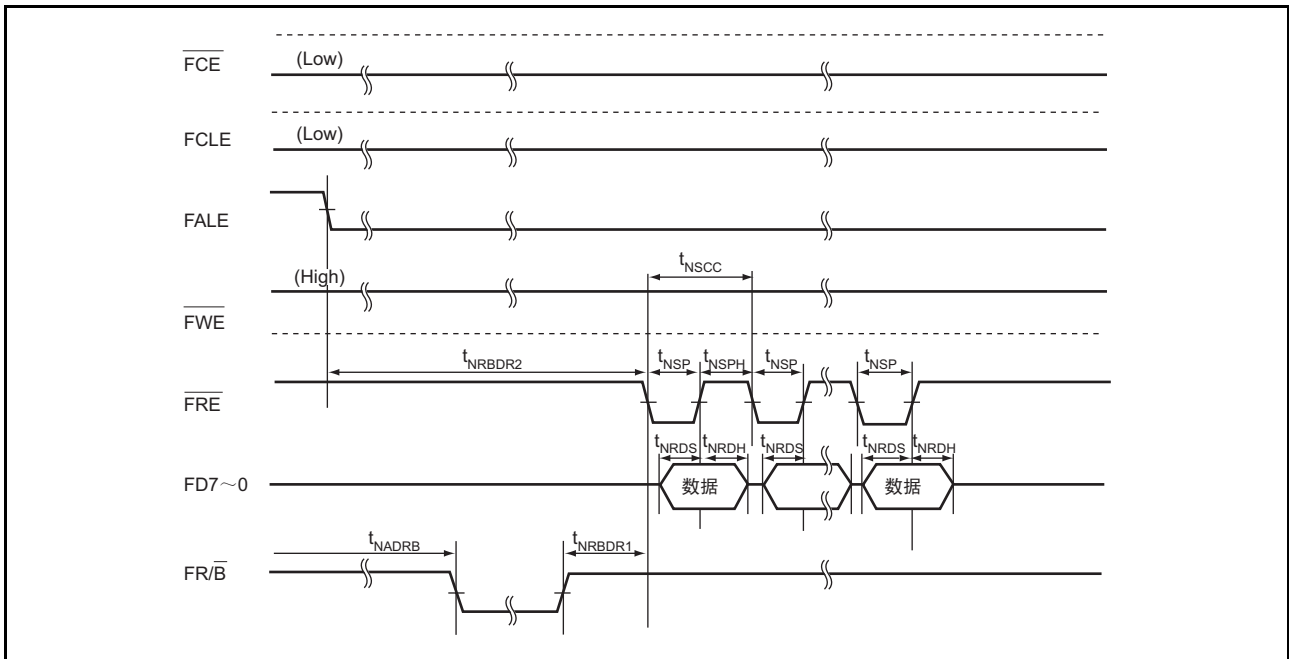


图 33.78 NAND 型闪存的数据读时序

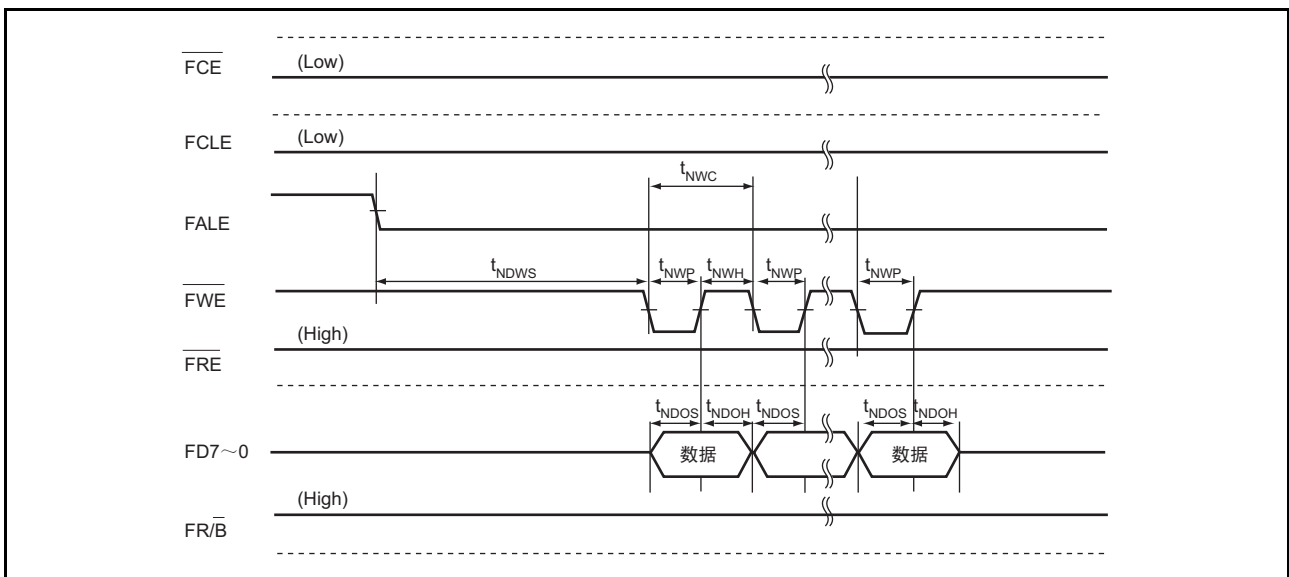


图 33.79 NAND 型闪存的数据写时序

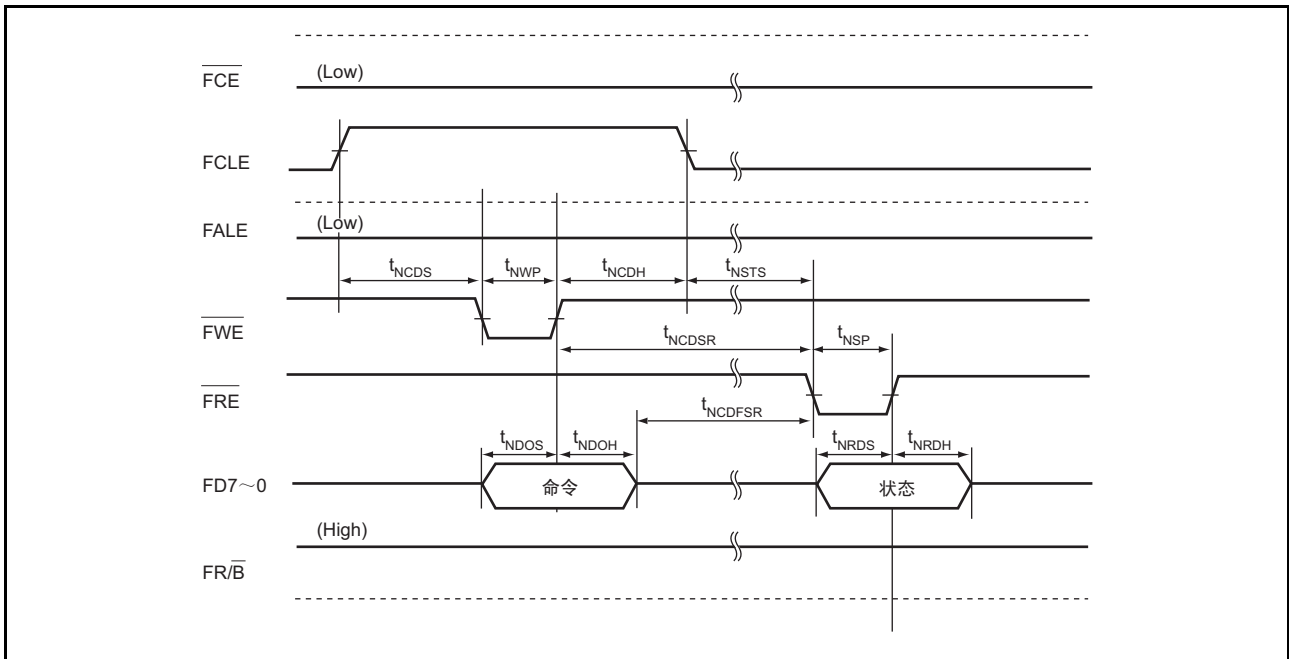


图 33.80 NAND 型闪存的状态读时序

33.4.16 LCDC 模块的信号时序

表 33.31 LCDC 模块的信号时序

条件: 3.3V 系列电源 =3.0 ~ 3.6V、1.2V 系列电源 =1.15 ~ 1.35V、 $T_a = -20 \sim 85^\circ\text{C}$ 、 $-40 \sim 85^\circ\text{C}$

项目	符号	Min.	Max.	单位	参照图
LCD_CLK 输入时钟频率	t_{FREQ}	—	54	MHz	33.81
LCD_CLK 输入时钟的上升时间	t_r	—	3	ns	
LCD_CLK 输入时钟的下降时间	t_f	—	3	ns	
LCD_CLK 输入时钟的占空比	t_{DUTY}	90	110	%	
时钟 (LCD_CL2) 周期时间	t_{CC}	25	—	ns	
时钟 (LCD_CL2) 高电平脉宽	t_{CHW}	7	—	ns	
时钟 (LCD_CL2) 低电平脉宽	t_{CLW}	7	—	ns	
时钟 (LCD_CL2) 转移时间 (上升 / 下降)	t_{CT}	—	3	ns	
数据 (LCD_DATA) 延迟时间	t_{DDdo}	-3.5	3	ns	
表示允许 (LCD_M_DISP) 延迟时间	t_{IDdo}	-3.5	3	ns	
水平同步信号 (LCD_CL1) 延迟时间	t_{HDdo}	-3.5	3	ns	
垂直同步信号 (LCD_FLM) 延迟时间	t_{VDdo}	-3.5	3	ns	

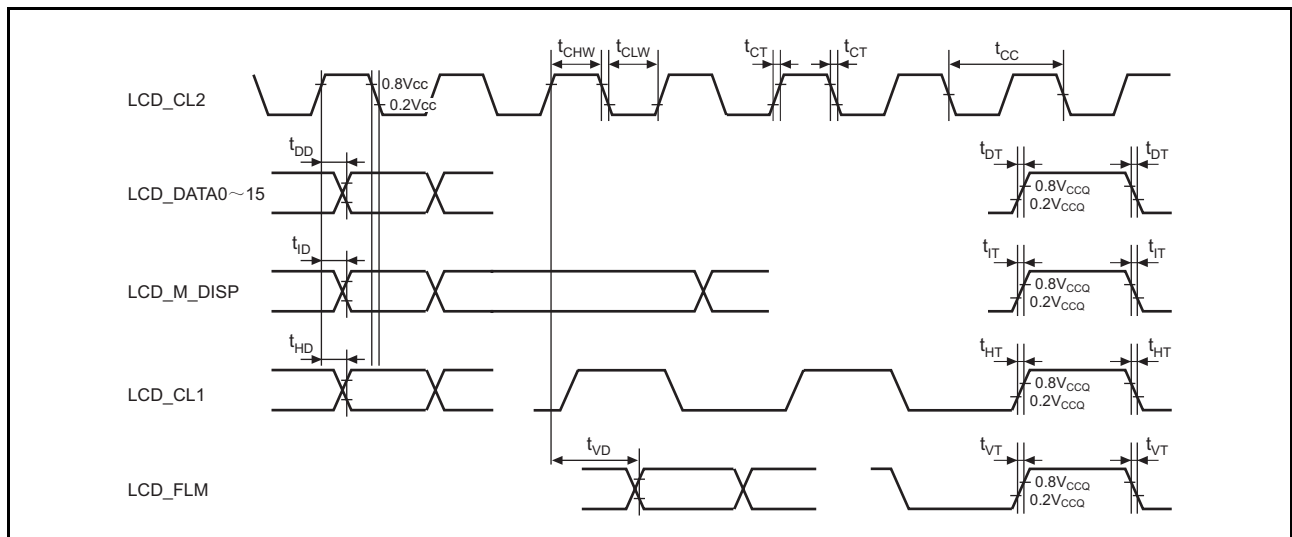


图 33.81 LCDC 模块的信号时序

33.4.17 VDC2 模块的信号时序

表 33.32 VDC2 模块的信号时序

条件：3.3V 系电源 = 3.0 ~ 3.6V、1.2V 系电源 = 1.15 ~ 1.35V、 $T_a = -20 \sim 85^\circ\text{C}$ 、 $-40 \sim 85^\circ\text{C}$

项目	符号	min	typ	max	单位	备注	参照图
输出时钟周期	t_O	18.5	—	158	ns	输出	33.82
输入时钟周期	t_I	18.5	—	158	ns	输入	
时钟 High	t_{HC}	6	—	—	ns	输入 / 输出	
时钟 Low	t_{LC}	6	—	—	ns	输入 / 输出	
时钟上升时间	t_{RC}	—	—	3	ns	输出 (100pF)	
延迟	内部同步模式	t_{DTRI}	—	5	ns	发送	33.83、33.84
	外部同步模式	t_{DTRO}	—	20	ns		
准备时间	t_{SR}	5	—	—	ns	接收	33.85、33.86
保持时间	t_{HTR}	5	—	—	ns	接收	33.85 ~ 33.86

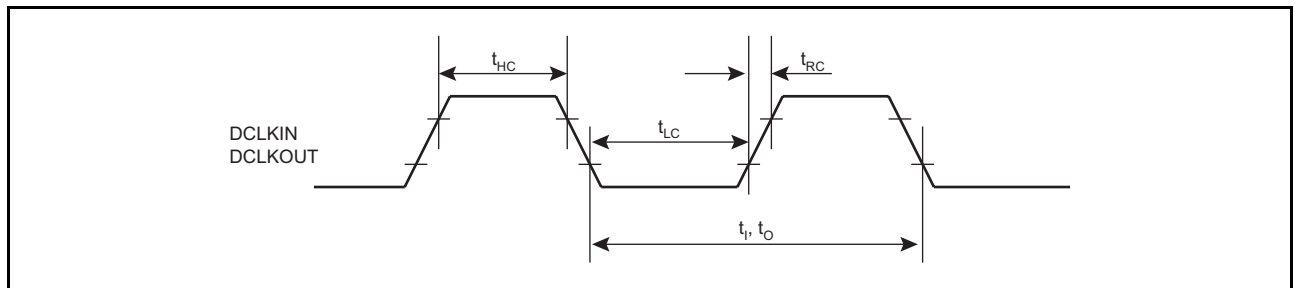


图 33.82 时钟的输入 / 输出时序

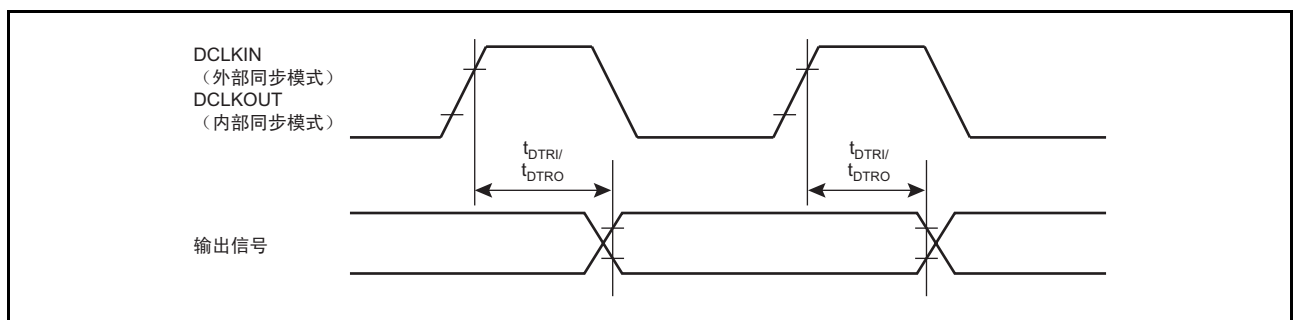


图 33.83 VDC2 发送时序 (1)

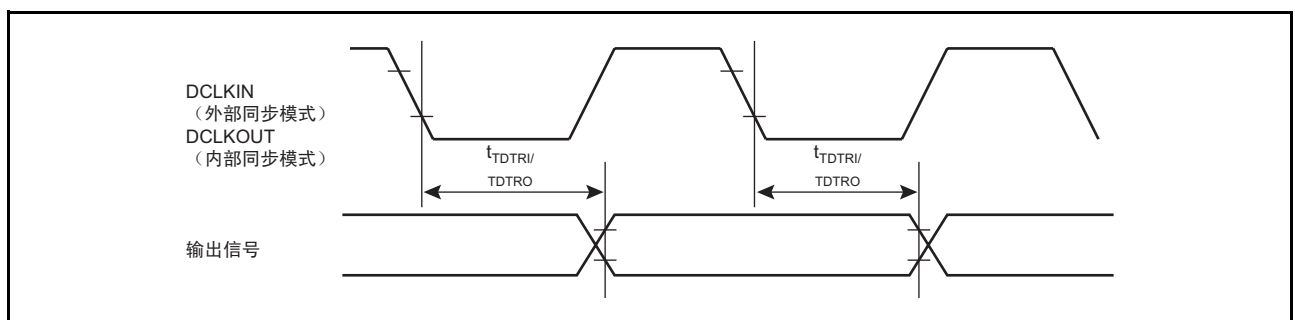


图 33.84 VDC2 发送时序 (2)

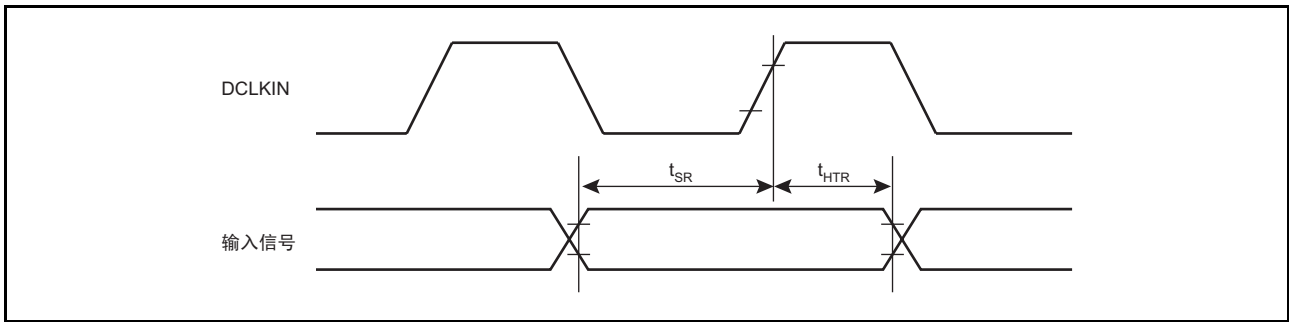


图 33.85 VDC2 接收时序 (1)

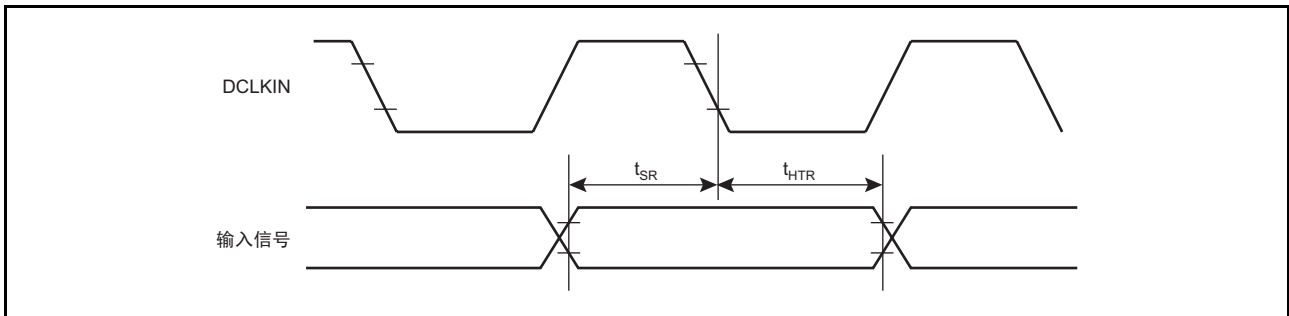


图 33.86 VDC2 接收时序 (2)

33.5 AC 特性的测量条件

AC 特性的测量条件如下所示：

- 输入/输出信号基准电平： $V^*/2$
- 输入脉冲电平： $V_{SSQ} \sim V^*$
- 输入上升和下降时间：1ns

【注】 V^* ： V_{DDQ} ($V_{DDQ}=3.0 \sim 3.6V$)

输出负载电路如图 33.87 所示。

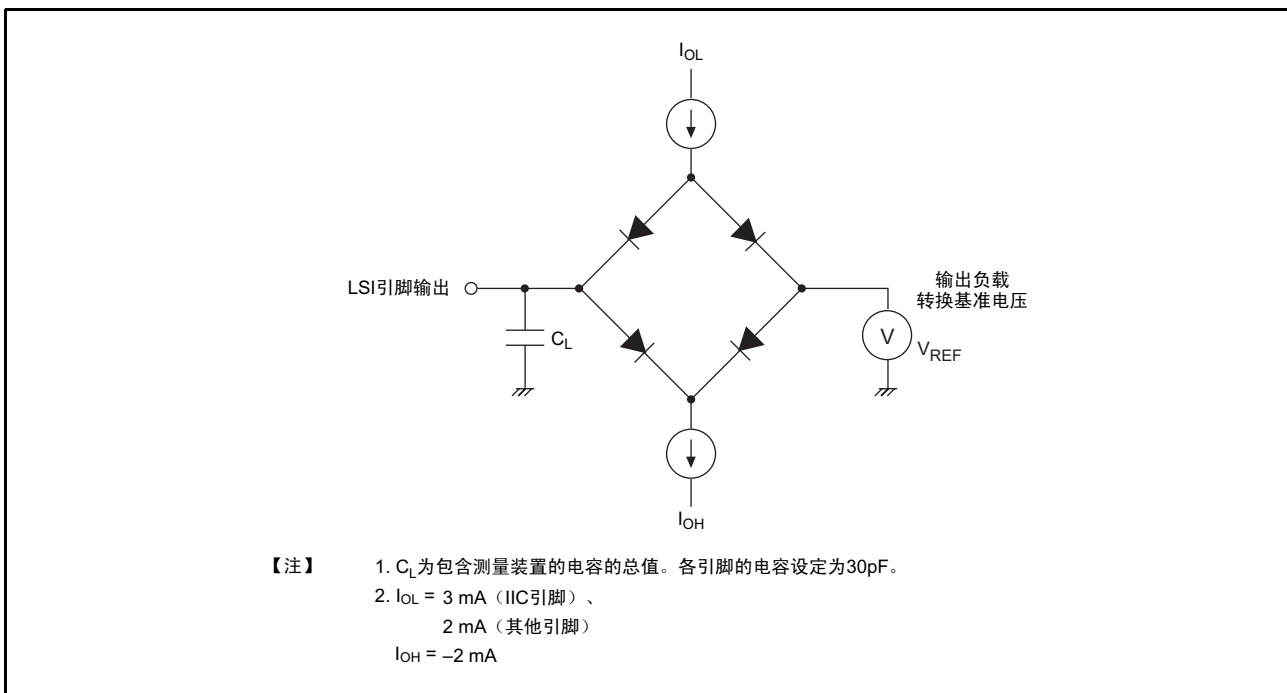


图 33.87 输出负载电路

附录

附录 A. CPU 运行模式寄存器 (CPUOPM)

CPUOPM 用于切换 CPU 的运行模式。能以 32 位长度从 P4 区的 H'FF2F0000 或者区域地址的 H'1F2F0000 读写此寄存器。在写此寄存器时，必须给保留位写初始值，否则就不保证运行。

不能通过 CPU 以外的 SuperHyway 总线主控进行存取，而必须通过 CPU 的存储指令进行 CPUOPM 的更新。必须在更新 CPUOPM 后，读一次 CPUOPM，然后执行以下 1 项或者 2 项：

1. 通过 RTE 指令进行转移。
2. 对任意地址（也包含不能缓存的区域）执行 ICBI 指令。

在执行后，就能保证 CPU 在运行时使用 CPUOPM 更新后的值。

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
	—	—	—	—	—	—	—	—	—	—	RABD	—	INTMU	—	—	—
初始值:	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R	R

位	位名	初始值	R/W	说明
31 ~ 6	—	H'000000F	R	保留位 只能写初始值。
5	RABD	1	R/W	子程序返回预测执行位 0: 从子程序返回时，预测性地发行取指令。当此位置 0 时，请参照“附录 C. 子程序返回的预测执行”。 1: 从子程序返回时，不预测性地发行取指令。
4	—	0	R	保留位 只能写初始值。
3	INTMU	0	R/W	中断运行模式切换位 0: 即使接受中断，SR.IMASK 的值也不变。 1: 当接受中断时，根据接受的电平自动设定 SR.IMASK 的值。
2 ~ 0	—	000	R	保留位 只能写初始值。

附录 B. 指令预取及其副作用

SH-4A 内部有用于保存预取指令的缓冲器，不断进行指令的预取。因此，不能在各存储空间最后 64 字节区域配置程序，否则有可能会发生超出存储区的预取指令的总线存取。

由此产生问题的例子如下所示。

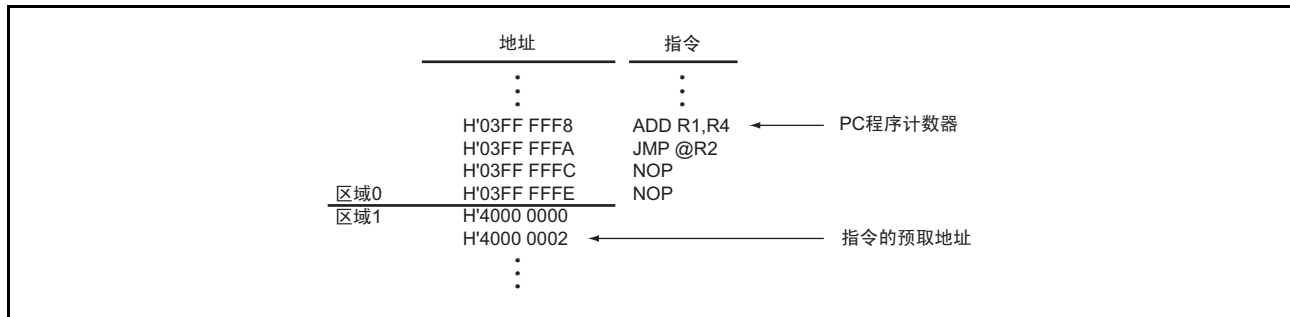


图 B.1 指令的预取

在图 B.1 中，假设了对 PC（程序计数器）指向的指令（ADD）和地址 H'0400 0002 的指令同时取指令的情况，并且假设了程序在执行后续的 JMP 指令和延迟槽指令后转移到区域 1 以外的区域。

此时，程序流程有可能发生对意外区域 1 的总线存取（指令的预取）。

(1) 预取指令的副作用

1. 预取指令引起的外部总线存取可能会导致连接此区域的 FIFO 等外部设备的误动作。
2. 对预取指令引起的外部总线请求，如果没有应答的设备，就可能导致死机。

(2) 回避方法

1. 通过使用 MMU，能避免这些非法取指令。
2. 只要不在各区域的最后 64 字节区域配置程序，就能避免此问题。

附录 C. 子程序返回的预测执行

SH-4A 内部具有从子程序返回时预测性地发行取指令的结构。在从子程序返回时，通过预测性地发行取指令，能缩短返回时的执行周期。将 CPU 运行模式寄存器（CPUOPM）的 bit5（RABD）置 0 时，此功能有效。但是，如果从子程序返回时预测性地发行取指令，就有可能对程序不应该存储的地址进行取指令。其结果是：有可能发生对意外区域的总线存取或者因发生内部指令地址错误而引发误动作。有关发生对意外区域的总线存取时的副作用，请参照“附录 B.(1) 预取指令的副作用”。

使用条件：

在将子程序返回预测执行功能设定为有效的情况下，对于 JSR/BSR/BSRF 指令设定在 PR 中的返回地址，必须使用 RTS 指令从子程序返回，这样能抑制存取不应该存储的地址，避免误动作。

附录 D. 版本寄存器 (PVR、PRR)

SH-4A 有表示处理器内核版本和产品版本的只读寄存器。如果使用这些寄存器的值，就能从软件区别处理器版本和产品版本，构筑扩展性高的系统。

【注】 必须屏蔽 PVR 寄存器的 bit7 ~ bit0 和 PRR 寄存器的 bit3 ~ bit0 的值，不能影响软件。

表 D.1 寄存器结构

名称	略称	R/W	P4 区地址	区域 7 地址	长度
处理器版本寄存器	PVR	R	H'FF00 0030	H'1F00 0030	32
产品寄存器	PRR	R	H'FF00 0044	H'1F00 0044	32

• PVR

位:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CHIP								VER							
初始值:	0	0	0	1	0	0	0	0	0	0	1	1	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
	CUT								—	—	—	—	—	—	—	—
初始值:	0	0	0	0	1	0	0	0	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

位	位名	初始值	R/W	说明
31 ~ 24	CHIP	H'10	R	表示处理器族的类别。 对于 SH-4A 系列，值一定是 H'10。
23 ~ 16	VER	H'30	R	表示版本。 当 SH-4A 系列进行大幅度的功能扩展时，值有变更。
15 ~ 8	CUT	H'08	R	表示版本。 当 SH-4A 系列进行小规模修改时，值有变更。
7 ~ 0	—	不定值	R	不定值 在读后，必须通过软件进行屏蔽后再使用。

• PRR

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
	Product								CUT				—	—	—	—
初始值:	0	0	0	1	0	0	0	1	—	—	—	—	—	—	—	—
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 ~ 16	—	全 0	R	固定为全 0。
15 ~ 8	Product	H'11	R	表示产品类别。
7 ~ 4	CUT	不定值	R	表示版本。 当产品进行小规模修改时，值有变更。
3 ~ 0	—	不定值	R	不定值。 在读后，必须通过软件进行屏蔽后再使用。

附录 E. 引脚状态

表 E.1 复位、低功耗状态、总线释放状态下的引脚状态

BGAball no.	引脚名	I/O	上电复位	刷新待机	睡眠	HiZ 控制时 *	总线 释放时
B1	XIN	XI	XI	XI	XI	XI	XI
C1	XOUT	XO	XO	XO	XO	XO	XO
D3	WOL/PF2/IDEA0_M	O/I/O	O	K/G/K	O/I/O	Z/I/O	O/I/O
E3	SSISCK2/PC3	IO/IO	I	K/G	IO/IO	Z/IO	IO/IO
F4	SSIDATA2/PC2	IO/IO	I	K/G	IO/IO	Z/IO	IO/IO
G4	SSIWS2/PC4	IO/IO	I	K/G	IO/IO	Z/IO	IO/IO
D2	LNKSTA/PF3/IDECS0_M	I/IO/O	I	I/G/K	I/IO/O	Z/IO/Z	I/IO/O
D1	EXOUT/PF4/IDECS1_M	O/I/O/O	O	K/G/K	O/I/O/O	Z/IO/Z	O/I/O/O
F3	AUDIO_CLK2/PC5	I/IO	I	I/G	I/IO	Z/IO	I/IO
E2	CRS/PD7/IDEA1_M	I/IO/O	I	I/G/K	I/IO/O	Z/IO/Z	I/IO/O
E1	COL/PE7/IDEA2_M	I/IO/O	I	I/G/K	I/IO/O	Z/IO/Z	I/IO/O
G3	TX_ER/PD6/IDEIOWR_M	O/I/O/O	O	K/G/K	O/I/O/O	Z/IO/Z	O/I/O/O
F2	MII_TXD3/SSIDATA5/ IODACK_M/PD0	O/I/O/O/IO	O	K/K/K/G	O/I/O/O/IO	Z/Z/Z/IO	O/I/O/O/IO
F1	MII_TXD2/AUDIO_CLK5/ IDEINT_M/PD1	O/I/I/IO	O	K/I/I/G	O/I/I/IO	Z/Z/Z/IO	O/I/I/IO
H4	MPMD	I	PI	PI	PI	PI	PI
H3	RX_ER/PE6/IODREQ_M	I/IO/I	I	I/G/I	I/IO/I	Z/IO/Z	I/IO/I
G2	MII_TXD1/SSIWS5/IDEIORD_M/ PD2	O/I/O/O/IO	O	K/K/K/G	O/I/O/O/IO	Z/Z/Z/IO	O/I/O/O/IO
J4	SSIDATA3/PH4	IO/IO	I	K/G	IO/IO	Z/IO	IO/IO
G1	MII_TXD0/SSISCK5/ IDEIORDY_M/PD3	O/I/O/I/IO	O	K/K/I/G	O/I/O/I/IO	Z/Z/Z/IO	O/I/O/I/IO
H2	TX_EN/PD4/IDED0_M	O/I/O/IO	O	K/G/IO	O/I/O/IO	Z/IO/Z	O/I/O/IO
J3	SSIWS3/PH6	IO/IO	I	K/G	IO/IO	Z/IO	IO/IO
H1	TX_CLK/PD5/IDED15_M	I/IO/IO	I	I/G/IO	I/IO/IO	Z/IO/Z	I/IO/IO
J2	RX_CLK/PE5/IDED1_M	I/IO/IO	I	I/G/IO	I/IO/IO	Z/IO/Z	I/IO/IO
J1	RX_DV/PE4/IDED14_M	I/IO/IO	I	I/G/IO	I/IO/IO	Z/IO/Z	I/IO/IO
K3	SSISCK3/PH5	IO/IO	I	K/G	IO/IO	Z/IO	IO/IO
K4	IRQ0/DTEND1	I/O	I	I/O	I/O	I/Z	I/O
K2	MII_RXD0/SSIWS4/IDED2_M/PE3	I/IO/IO/IO	I	I/K/IO/G	I/IO/IO/IO	Z/Z/Z/IO	I/IO/IO/IO
K1	MII_RXD1/SSISCK4/IDED13_M/PE2	I/IO/IO/IO	I	I/K/IO/G	I/IO/IO/IO	Z/Z/Z/IO	I/IO/IO/IO
L3	AUDIO_CLK3/PH7	I/IO	I	I/G	I/IO	Z/IO	I/IO
L2	MII_RXD2/SSIDATA4/IDED3_M/ PE1	I/IO/IO/IO	I	I/K/IO/G	I/IO/IO/IO	Z/Z/Z/IO	I/IO/IO/IO
L4	IRQOUT/DREQ1	O/I	O	K/I	O/I	O/Z	O/I
L1	MII_RXD3/AUDIO_CLK4/ IDED12_M/PE0	I/I/IO/IO	I	I/I/IO/G	I/I/IO/IO	Z/Z/Z/IO	I/I/IO/IO
M2	MDC/PF0/IDED4_M	O/I/O/IO	O	K/G/IO	O/I/O/IO	Z/IO/Z	O/I/O/IO
M1	MDIO/PF1/IDED11_M	IO/IO/IO	I	K/G/IO	IO/IO/IO	Z/IO/Z	IO/IO/IO
N1	AUDIO_CLK0/PC7	I/IO	I	I/G	I/IO	Z/IO	I/IO

BGAball no.	引脚名	I/O	上电复位	刷新待机	睡眠	HiZ 控制时 *	总线 释放时
M3	SSIWS0	IO	I	K	IO	Z	IO
M4	STATUS1/RTS2/PA7	O/IO/IO	L	L/K/G	H/IO/IO	L/Z/IO	H/IO/IO
N2	SSISCK0	IO	I	K	IO	Z	IO
P1	AUDIO_CLK1/PC6	I/IO	I	I/G	I/IO	Z/IO	I/IO
N4	STATUS0/CTS2/PA6	O/IO/IO	L	H/K/G	L/IO/IO	L/Z/IO	L/IO/IO
N3	SSIDATA0	IO	I	K	IO	Z	IO
P2	SSISCK1	IO	I	K	IO	Z	IO
R1	PJ7/IDED10_M	IO/IO	I	G/IO	IO/IO	IO/Z	IO/IO
P3	SSIWS1	IO	I	K	IO	Z	IO
R2	PJ6/IDED5_M	IO/IO	I	G/IO	IO/IO	IO/Z	IO/IO
P4	FRE/PA4	O/IO	O	K/G	O/IO	Z/IO	O/IO
R3	SSIDATA1	IO	I	K	IO	Z	IO
T1	PJ5/IDED9_M	IO/IO	I	G/IO	IO/IO	IO/Z	IO/IO
T2	PJ4/IDED6_M	IO/IO	I	G/IO	IO/IO	IO/Z	IO/IO
U1	PJ2/IDED8_M	IO/IO	I	G/IO	IO/IO	IO/Z	IO/IO
U2	PJ3/IDED7_M	IO/IO	I	G/IO	IO/IO	IO/Z	IO/IO
T3	FWE/PA3	O/IO	O	K/G	O/IO	Z/IO	O/IO
R4	FCE/PA5	O/IO	O	K/G	O/IO	Z/IO	O/IO
V1	PJ1/IDERST_M	IO/O	I	G/K	IO/O	IO/Z	IO/O
V2	PJ0/DIRECTION_M	IO/O	I	G/K	IO/O	IO/Z	IO/O
U3	MODE7/FD6	I/IO	Z	-/K	-/IO	-/Z	-/IO
T4	FALE/PC0	O/IO	O	K/G	O/IO	Z/IO	O/IO
W1	MODE3/FD3	I/IO	Z	-/K	-/IO	-/Z	-/IO
V3	MODE5/FD5	I/IO	Z	-/K	-/IO	-/Z	-/IO
Y1	TXD2/PA2	O/IO	Z	K/G	O/IO	Z/IO	O/IO
W2	MODE2/FD2	I/IO	Z	-/K	-/IO	-/Z	-/IO
V4	MODE4/FD4	I/IO	Z	-/K	-/IO	-/Z	-/IO
U4	MODE8/FD7	I/IO	Z	-/K	-/IO	-/Z	-/IO
W3	MODE1/FD1	I/IO	Z	-/K	-/IO	-/Z	-/IO
Y2	RXD2/PA1	I/IO	I	I/G	I/IO	Z/IO	I/IO
AA1	SCK2/PA0	IO/IO	I	K/G	IO/IO	Z/IO	IO/IO
AB3	SDA	IO	Z	IO	IO	Z	IO
AB2	SCL	IO	Z	IO	IO	Z	IO
Y4	RXD1/AUDATA2	I/IO	I	I/K	I/IO	Z/IO	I/IO
W5	WDTOVF/IRQ1/AUDCK/DACK1	O/I/IO/O	O	O/I/K/O	O/I/IO/O	O/I/IO/Z	O/I/IO/O
AA3	MODE0/FD0	I/IO	Z	-/K	-/IO	-/Z	-/IO
Y5	RXD0/AUDATA0	I/IO	I	I/K	I/IO	Z/IO	I/IO
AA4	TXD1/AUDATA3	O/IO	Z	K/K	O/IO	Z/IO	O/IO
W6	TDO	O	Z	O	O	O	O
AA5	TXD0/AUDATA1	O/IO	Z	K/K	O/IO	Z/IO	O/IO
AB4	SCK1/FR/B	IO/I	I	K/I	IO/I	Z/Z	IO/I
Y6	TMS	I	PI	PI	PI	PI	PI
W7	TRST	I	PI	PI	PI	PI	PI
AA6	TDI	I	PI	PI	PI	PI	PI

BGAball no.	引脚名	I/O	上电复位	刷新待机	睡眠	HiZ 控制时 *	总线 释放时
AB5	SCK0/AUDSYNC/FCLE	IO/IO/O	I	K/K/K	IO/IO/O	Z/IO/Z	IO/IO/O
AB6	TCK	I	PI	PI	PI	PI	PI
Y8	LCD_VEP_WC/DR5/PH0	O/O/IO	O	K/K/G	O/O/IO	Z/Z/IO	O/O/IO
AA8	LCD_FLM/VSYNC/SPS/ EX_VSYNC/BT_VSYNC	O/IO/O	O	K/K/K	O/IO/O	Z/Z/Z	O/IO/O
AB8	LCD_CL1/HSYNC/SPL/ EX_HSYNC/BT_HSYNC	O/IO/O	O	K/K/K	O/IO/O	Z/Z/Z	O/IO/O
AA9	LCD_M_DISP/DE_C/DE_H/ BT_DE_C	O/O/O	O	K/K/K	O/O/O	Z/Z/Z	O/O/O
Y9	LCD_VCP_WC/DR4/PH1	O/O/IO	O	K/K/G	O/O/IO	Z/Z/IO	O/O/IO
AB9	LCD_CLK/DCLKIN	I/I	I	I/I	I/I	Z/Z	I/I
Y10	LCD_DATA15/DR3/PG7	O/O/IO	O	K/K/G	O/O/IO	Z/O/IO	O/O/IO
AA10	LCD_DATA14/DR2/PG6	O/O/IO	O	K/K/G	O/O/IO	Z/O/IO	O/O/IO
AB10	LCD_DATA13/DR1/PG5	O/O/IO	O	K/K/G	O/O/IO	Z/O/IO	O/O/IO
Y11	LCD_DATA12/DR0/PG4	O/O/IO	O	K/K/G	O/O/IO	Z/O/IO	O/O/IO
AA11	LCD_DATA11/DG5/PG3	O/O/IO	O	K/K/G	O/O/IO	Z/O/IO	O/O/IO
AB11	LCD_DATA10/DG4/PG2	O/O/IO	O	K/K/G	O/O/IO	Z/O/IO	O/O/IO
Y12	LCD_DATA9/DG3/PG1	O/O/IO	O	K/K/G	O/O/IO	Z/O/IO	O/O/IO
AA12	LCD_DATA8/DG2/PG0	O/O/IO	O	K/K/G	O/O/IO	Z/O/IO	O/O/IO
AB12	LCD_DATA7/DG1/BT_DATA7/PI4	O/O/O/IO	O	K/K/K/G	O/O/O/IO	Z/Z/Z/IO	O/O/O/IO
Y13	LCD_DATA6/DG0/BT_DATA6/PI3	O/O/O/IO	O	K/K/K/G	O/O/O/IO	Z/Z/Z/IO	O/O/O/IO
AA13	LCD_DATA5/DB5/BT_DATA5/PI2	O/O/O/IO	O	K/K/K/G	O/O/O/IO	Z/Z/Z/IO	O/O/O/IO
AB13	LCD_DATA4/DB4/BT_DATA4/PI1	O/O/O/IO	O	K/K/K/G	O/O/O/IO	Z/Z/Z/IO	O/O/O/IO
Y14	LCD_DATA3/DB3/BT_DATA3	O/O/O	O	K/K/K	O/O/O	Z/Z/Z	O/O/O
AA14	LCD_DATA2/DB2/BT_DATA2	O/O/O	O	K/K/K	O/O/O	Z/Z/Z	O/O/O
AB14	LCD_DATA1/DB1/BT_DATA1	O/O/O	O	K/K/K	O/O/O	Z/Z/Z	O/O/O
AA15	LCD_DATA0/DB0/BT_DATA0	O/O/O	O	K/K/K	O/O/O	Z/Z/Z	O/O/O
Y15	LCD_CL2/DE_V/PH3	O/O/IO	O	K/K/G	O/O/IO	Z/Z/IO	O/O/IO
AB15	LCD_DON/DCLKOUT/PH2	O/O/IO	O	K/K/G	O/O/IO	Z/Z/IO	O/O/IO
Y16	PI0/COM/CDE	IO/O	I	G/K	IO/O	IO/Z	IO/O
AB17	RDY	I	I	MI	MI	MI	MI
AA17	NMI	I	I	I	I	I	I
AA18	BACK	O	O	K	O	O	O
Y17	RD	O	O	K	O	O	MZ
W17	CS3	O	O	K	O	O	MZ
Y19	BREQ	I	I	MI	MI	MI	MI
AB21	EXTAL	XI	XI	XI	XI	XI	XI
AB22	XTAL	XO	XO	XO	XO	XO	XO
Y20	BS	O	O	K	O	O	MZ
Y22	PRESET	I	I	I	I	I	I
W18	CS0	O	O	K	O	O	MZ
W21	ASEBRK/BRKACK/TCLK/PC1	O/I/IO	A	A/I/G	IO/I/IO	IO/Z/IO	IO/I/IO
V19	A25/PB7/DREQ0/RTS0	O/IO/I/IO	O	K/G/I/K	O/IO/I/IO	O/IO/Z/Z	MZ/IO/I/IO
W22	A24/PB6/DACK0/CTS0	O/IO/O/IO	O	K/G/O/K	O/IO/O/IO	O/IO/Z/Z	MZ/IO/O/IO

BGAball no.	引脚名	I/O	上电复位	刷新待机	睡眠	HiZ 控制时 *	总线 释放时
U19	A17	O	O	K	O	O	MZ
V21	A23/PB5/DTEND0/RTS1	O/IO/O/IO	O	K/G/O/K	O/IO/O/IO	O/IO/Z/Z	MZ/IO/O/IO
V20	A21/PB3	O/IO	O	K/G	O/IO	O/IO	MZ/IO
U22	A20/PB2	O/IO	O	K/G	O/IO	O/IO	MZ/IO
V22	A22/PB4/CTS1	O/IO/IO	O	K/G/K	O/IO/IO	O/IO/Z	MZ/IO/IO
U21	A19/PB1	O/IO	O	K/G	O/IO	O/IO	MZ/IO
U20	A18/PB0	O/IO	O	K/G	O/IO	O/IO	MZ/IO
T19	D15	IO	PZ	MZ	IO	IO	MZ
T20	D14	IO	PZ	MZ	IO	IO	MZ
T21	D1	IO	PZ	MZ	IO	IO	MZ
T22	D0	IO	PZ	MZ	IO	IO	MZ
R19	D13	IO	PZ	MZ	IO	IO	MZ
R20	D12	IO	PZ	MZ	IO	IO	MZ
R21	D3	IO	PZ	MZ	IO	IO	MZ
R22	D2	IO	PZ	MZ	IO	IO	MZ
P19	D11	IO	PZ	MZ	IO	IO	MZ
P20	D10	IO	PZ	MZ	IO	IO	MZ
P21	D5	IO	PZ	MZ	IO	IO	MZ
P22	D4	IO	PZ	MZ	IO	IO	MZ
N19	D9	IO	PZ	MZ	IO	IO	MZ
N22	D6	IO	PZ	MZ	IO	IO	MZ
N21	D7	IO	PZ	MZ	IO	IO	MZ
N20	D8	IO	PZ	MZ	IO	IO	MZ
M22	DQMLL	O	O	K	O	O	MZ
M21	DQMUL	O	O	K	O	O	MZ
M19	DQMUU	O	O	K	O	O	MZ
L22	D16	IO	PZ	MZ	IO	IO	MZ
M20	DQMLU	O	O	K	O	O	MZ
L21	D17	IO	PZ	MZ	IO	IO	MZ
K22	D18	IO	PZ	MZ	IO	IO	MZ
K21	D19	IO	PZ	MZ	IO	IO	MZ
L19	D31	IO	PZ	MZ	IO	IO	MZ
L20	D30	IO	PZ	MZ	IO	IO	MZ
J22	D20	IO	PZ	MZ	IO	IO	MZ
J21	D21	IO	PZ	MZ	IO	IO	MZ
H22	D22	IO	PZ	MZ	IO	IO	MZ
K20	D28	IO	PZ	MZ	IO	IO	MZ
K19	D29	IO	PZ	MZ	IO	IO	MZ
G22	A15	O	O	K	O	O	MZ
H21	D23	IO	PZ	MZ	IO	IO	MZ
J20	D26	IO	PZ	MZ	IO	IO	MZ
F22	A13	O	O	K	O	O	MZ
G21	A16	O	O	K	O	O	MZ
J19	D27	IO	PZ	MZ	IO	IO	MZ

BGAball no.	引脚名	I/O	上电复位	刷新待机	睡眠	HiZ 控制时 *	总线 释放时
H20	D24	IO	PZ	MZ	IO	IO	MZ
E22	A10	O	O	K	O	O	MZ
F21	A14	O	O	K	O	O	MZ
H19	D25	IO	PZ	MZ	IO	IO	MZ
D22	A4	O	O	K	O	O	MZ
G20	A11	O	O	K	O	O	MZ
E21	A5	O	O	K	O	O	MZ
D21	R/W	O	O	H	O	O	MZ
F20	A8	O	O	K	O	O	MZ
G19	A12	O	O	K	O	O	MZ
C22	CKE	O	O	K	O	O	MZ
C21	RAS	O	O	H	O	O	MZ
B22	CLKOUT	O	O	C	O	O	O
F19	A9	O	O	K	O	O	MZ
E20	A6	O	O	K	O	O	MZ
E19	A7	O	O	K	O	O	MZ
D20	CAS	O	O	K	O	O	MZ
A21	CS1	O	O	K	O	O	MZ
B20	CS2	O	O	K	O	O	MZ
C19	A0	O	O	K	O	O	MZ
D18	D47/IDECs0	IO/O	PZ	MZ/K	IO/O	IO/Z	MZ/O
A20	A3	O	O	K	O	O	MZ
B19	A1	O	O	K	O	O	MZ
D17	D45/IODACK	IO/O	PZ	MZ/K	IO/O	IO/Z	MZ/O
C18	D46/IDECs1	IO/O	PZ	MZ/K	IO/O	IO/Z	MZ/O
B18	D33/PF6	IO/IO	PZ	MZ/G	IO/IO	IO/IO	MZ/IO
A19	A2	O	O	K	O	O	MZ
C17	D44/IDEINT	IO/I	PZ	MZ/I	IO/I	IO/Z	MZ/I
D16	D43/IDEIORDY	IO/I	PZ	MZ/I	IO/I	IO/Z	MZ/I
C16	D42/IDEIORD	IO/O	PZ	MZ/K	IO/O	IO/Z	MZ/O
A18	D32/PF7	IO/IO	PZ	MZ/G	IO/IO	IO/IO	MZ/IO
C15	D40/IDEIOWR	IO/O	PZ	MZ/K	IO/O	IO/Z	MZ/O
D15	D41/IODREQ	IO/I	PZ	MZ/I	IO/I	IO/Z	MZ/I
B17	D35/IDEA0	IO/O	PZ	MZ/K	IO/O	IO/Z	MZ/O
B16	D37/IDEA1	IO/O	PZ	MZ/K	IO/O	IO/Z	MZ/O
B15	D39/IDED14	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
A17	D34/PF5	IO/IO	PZ	MZ/G	IO/IO	IO/IO	MZ/IO
A16	D36/IDEA2	IO/O	PZ	MZ/K	IO/O	IO/Z	MZ/O
D14	D63/IDED1	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
A15	D38/IDED15	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
C14	D62/IDED0	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
B14	WE2/DQM64UL	O/O	O	K/K	O/O	O/O	MZ/MZ
A14	WE0/DQM64LL	O/O	O	K/K	O/O	O/O	MZ/MZ
C13	D60/IDED2	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO

BGAball no.	引脚名	I/O	上电复位	刷新待机	睡眠	HiZ 控制时*	总线 释放时
B13	WE3/DQM64UU	O/O	O	K/K	O/O	O/O	MZ/MZ
A13	WE1/DQM64LU	O/O	O	K/K	O/O	O/O	MZ/MZ
D13	D61/IDED3	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
A12	D48/IDED13	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
D12	D59/IDED5	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
C12	D58/IDED4	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
B12	D49/IDED12	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
B11	D51/IDED10	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
A11	D50/IDED11	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
C11	D56/IDED6	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
A10	D52/IDED9	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
B10	D53/IDED8	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
D11	D57/IDED7	IO/IO	PZ	MZ/IO	IO/IO	IO/Z	MZ/IO
C10	D54/IDERST	IO/O	PZ	MZ/K	IO/O	IO/Z	MZ/O
D10	D55/DIRECTION	IO/O	PZ	MZ/K	IO/O	IO/Z	MZ/O
A7	DM	AIO	Z	Z	AIO	Z	AIO
A6	DP	AIO	Z	Z	AIO	Z	AIO
B6	VBUS	AI	AI	AI	AI	AI	AI
E8	REFRIN	AI	AI	AI	AI	AI	AI

【注】 * 表示设定 GPIO 的 Hi-Z 寄存器 A、B (PTHIZ_A、B) 的对应位时的引脚状态。

【符号说明】

- I : 输入
- O : 输出
- IO : 输入/输出
- XI : XTAL 输入
- XO : XTAL 输出
- AI : 模拟输入
- AIO : 模拟输入/输出
- Z : 高阻抗
- PI : 通过内部上拉电阻进行上拉的输入
- PZ : 通过内部上拉电阻进行上拉的高阻抗。
- H : 高电平输出
- L : 低电平输出
- K : 输入和输出引脚都保持转移到刷新待机模式前的状态。
- MI : 输入缓冲器 ON, 通过 MCU 寄存器的设定进行内部上拉电阻 ON/OFF 的控制。
- MZ : 输入缓冲器 OFF、输出缓冲器 OFF, 通过 MCU 寄存器的设定进行内部上拉电阻 ON/OFF 的控制。
- A : 内部上拉电阻 ON, 当 MPMD=低电平时, 通过寄存器进行 I/O 控制, 当 TRST=低电平或者 MPMD=高电平时为输入。
- C : 根据 CPG 寄存器的设定, 为时钟输出或者低电平输出。
- G : 通过 GPIO 寄存器的设定进行输入/输出和内部上拉电阻 ON/OFF 的控制。
- GPI : 输入缓冲器 ON、输出缓冲器 OFF, 通过 GPIO 寄存器的设定进行内部上拉电阻 ON/OFF 的控制。
- GPZ : 输入缓冲器 OFF、输出缓冲器 OFF, 通过 GPIO 寄存器的设定进行内部上拉电阻 ON/OFF 的控制。
- : 不可选择

附录 F. 未使用引脚的处理

表 F.1 未使用引脚的处理

BGA ball no.	引脚名	未使用引脚的处理 *4
B1	XIN	下拉
C1	XOUT	开路
D3	WOL/PF2/IDEA0_M	开路
E3	SSISCK2/PC3	上拉
F4	SSIDATA2/PC2	上拉
G4	SSIWS2/PC4	上拉
D2	LNKSTA/PF3/IDECS0_M	下拉
D1	EXOUT/PF4/IDECS1_M	开路
F3	AUDIO_CLK2/PC5	上拉
E2	CRS/PD7/IDEA1_M	下拉
E1	COL/PE7/IDEA2_M	下拉
G3	TX_ER/PD6/IDEIOWR_M	开路
F2	MII_TXD3/SSIDATA5/IODACK_M/PD0	开路
F1	MII_TXD2/AUDIO_CLK5/IDEINT_M/PD1	开路
H4	MPMD	上拉 *2
H3	RX_ER/PE6/IODREQ_M	下拉
G2	MII_TXD1/SSIWS5/IDEIORD_M/PD2	开路
J4	SSIDATA3/PH4	上拉
G1	MII_TXD0/SSISCK5/IDEIORDY_M/PD3	开路
H2	TX_EN/PD4/IDED0_M	开路
J3	SSIWS3/PH6	上拉
H1	TX_CLK/PD5/IDED15_M	下拉
J2	RX_CLK/PE5/IDED1_M	下拉
J1	RX_DV/PE4/IDED14_M	下拉
K3	SSISCK3/PH5	上拉
K4	IRQ0/DTEND1	上拉
K2	MII_RXD0/SSIWS4/IDED2_M/PE3	下拉
K1	MII_RXD1/SSISCK4/IDED13_M/PE2	下拉
L3	AUDIO_CLK3/PH7	上拉
L2	MII_RXD2/SSIDATA4/IDED3_M/PE1	下拉
L4	IRQOUT/DREQ1	开路
L1	MII_RXD3/AUDIO_CLK4/IDED12_M/PE0	下拉
M2	MDC/PF0/IDED4_M	开路
M1	MDIO/PF1/IDED11_M	下拉
N1	AUDIO_CLK0/PC7	上拉
M3	SSIWS0	上拉
M4	STATUS1/RTS2/PA7	开路

BGA ball no.	引脚名	未使用引脚的处理 *4
N2	SSISCK0	上拉
P1	AUDIO_CLK1/PC6	上拉
N4	STATUS0/CTS2/PA6	开路
N3	SSIDATA0	上拉
P2	SSISCK1	上拉
R1	PJ7/IDED10_M	上拉
P3	SSIWS1	上拉
R2	PJ6/IDED5_M	上拉
P4	FRE/PA4	开路
R3	SSIDATA1	上拉
T1	PJ5/IDED9_M	上拉
T2	PJ4/IDED6_M	上拉
U1	PJ2/IDED8_M	上拉
U2	PJ3/IDED7_M	上拉
T3	FWE/PA3	开路
R4	FCE/PA5	开路
V1	PJ1/IDERST_M	上拉
V2	PJ0/DIRECTION_M	上拉
U3	MODE7/FD6	下拉
T4	FALE/PC0	开路
W1	MODE3/FD3	必须使用
V3	MODE5/FD5	必须使用
Y1	TXD2/PA2	开路
W2	MODE2/FD2	必须使用
V4	MODE4/FD4	必须使用
U4	MODE8/FD7	必须使用
W3	MODE1/FD1	必须使用
Y2	RXD2/PA1	上拉
AA1	SCK2/PA0	上拉
AB3	SDA	开路
AB2	SCL	开路
Y4	RXD1/AUDATA2	上拉
W5	WDTOVF/IRQ1/AUDCK/DACK1	开路
AA3	MODE0/FD0	必须使用
Y5	RXD0/AUDATA0	上拉
AA4	TXD1/AUDATA3	开路
W6	TDO	开路 *2
AA5	TXD0/AUDATA1	开路
AB4	SCK1/FR \bar{B}	上拉
Y6	TMS	开路 *2

BGA ball no.	引脚名	未使用引脚的处理 *4
W7	TRST	固定接地或者连接 PRESET*2*3
AA6	TDI	开路 *2
AB5	SCK0/AUDSYNC/FCLE	上拉
AB6	TCK	开路 *2
Y8	LCD_VEP_WC/DR5/PH0	开路
AA8	LCD_FLM/VSYNC/SPS/EX_VSYNC/BT_VSYNC	开路
AB8	LCD_CL1/HSYNC/SPL/EX_HSYNC/BT_HSYNC	开路
AA9	LCD_M_DISP/DE_C/DE_H/BT_DE_C	开路
Y9	LCD_VCP_WC/DR4/PH1	开路
AB9	LCD_CLK/DCLKIN	上拉
Y10	LCD_DATA15/DR3/PG7	开路
AA10	LCD_DATA14/DR2/PG6	开路
AB10	LCD_DATA13/DR1/PG5	开路
Y11	LCD_DATA12/DR0/PG4	开路
AA11	LCD_DATA11/DG5/PG3	开路
AB11	LCD_DATA10/DG4/PG2	开路
Y12	LCD_DATA9/DG3/PG1	开路
AA12	LCD_DATA8/DG2/PG0	开路
AB12	LCD_DATA7/DG1/BT_DATA7/PI4	开路
Y13	LCD_DATA6/DG0/BT_DATA6/PI3	开路
AA13	LCD_DATA5/DB5/BT_DATA5/PI2	开路
AB13	LCD_DATA4/DB4/BT_DATA4/PI1	开路
Y14	LCD_DATA3/DB3/BT_DATA3	开路
AA14	LCD_DATA2/DB2/BT_DATA2	开路
AB14	LCD_DATA1/DB1/BT_DATA1	开路
AA15	LCD_DATA0/DB0/BT_DATA0	开路
Y15	LCD_CL2/DE_V/PH3	开路
AB15	LCD_DON/DCLKOUT/PH2	开路
Y16	PI0/COM/CDE	上拉
AB17	RDY	下拉 *1
AA17	NMI	上拉
AA18	BACK	开路
Y17	RD	开路
W17	CS3	开路
Y19	BREQ	上拉
AB21	EXTAL	必须使用
AB22	XTAL	开路
Y20	BS	开路
Y22	PRESET	必须使用
W18	CS0	必须使用

BGA ball no.	引脚名	未使用引脚的处理 *4
W21	ASEBRKAK/BRKACK/TCLK/PC1	开路 *2
V19	A25/PB7/DREQ0/RTS0	开路
W22	A24/PB6/DACK0/CTS0	开路
U19	A17	开路
V21	A23/PB5/DTEND0/RTS1	开路
V20	A21/PB3	开路
U22	A20/PB2	开路
V22	A22/PB4/CTS1	开路
U21	A19/PB1	开路
U20	A18/PB0	开路
T19	D15	开路
T20	D14	开路
T21	D1	必须使用
T22	D0	必须使用
R19	D13	开路
R20	D12	开路
R21	D3	必须使用
R22	D2	必须使用
P19	D11	开路
P20	D10	开路
P21	D5	必须使用
P22	D4	必须使用
N19	D9	开路
N22	D6	必须使用
N21	D7	必须使用
N20	D8	开路
M22	DQMLL	开路
M21	DQMUL	开路
M19	DQMUU	开路
L22	D16	开路
M20	DQMLU	开路
L21	D17	开路
K22	D18	开路
K21	D19	开路
L19	D31	开路
L20	D30	开路
J22	D20	开路
J21	D21	开路
H22	D22	开路
K20	D28	开路

BGA ball no.	引脚名	未使用引脚的处理 *4
K19	D29	开路
G22	A15	开路
H21	D23	开路
J20	D26	开路
F22	A13	开路
G21	A16	开路
J19	D27	开路
H20	D24	开路
E22	A10	开路
F21	A14	开路
H19	D25	开路
D22	A4	开路
G20	A11	开路
E21	A5	开路
D21	R/W	开路
F20	A8	开路
G19	A12	开路
C22	CKE	开路
C21	RAS	开路
B22	CLKOUT	开路
F19	A9	开路
E20	A6	开路
E19	A7	开路
D20	CAS	开路
A21	CS1	开路
B20	CS2	开路
C19	A0	开路
D18	D47/IDECS0	开路
A20	A3	开路
B19	A1	开路
D17	D45/IODACK	开路
C18	D46/IDECS1	开路
B18	D33/PF6	开路
A19	A2	开路
C17	D44/IDEINT	开路
D16	D43/IDEIORDY	开路
C16	D42/IDEIORD	开路
A18	D32/PF7	开路
C15	D40/IDEIOWR	开路
D15	D41/IODREQ	开路

BGA ball no.	引脚名	未使用引脚的处理 *4
B17	D35/IDEA0	开路
B16	D37/IDEA1	开路
B15	D39/IDED14	开路
A17	D34/PF5	开路
A16	D36/IDEA2	开路
D14	D63/IDED1	开路
A15	D38/IDED15	开路
C14	D62/IDED0	开路
B14	$\overline{\text{WE2}}/\text{DQM64UL}$	开路
A14	$\overline{\text{WE0}}/\text{DQM64LL}$	开路
C13	D60/IDED2	开路
B13	$\overline{\text{WE3}}/\text{DQM64UU}$	开路
A13	$\overline{\text{WE1}}/\text{DQM64LU}$	开路
D13	D61/IDED3	开路
A12	D48/IDED13	开路
D12	D59/IDED5	开路
C12	D58/IDED4	开路
B12	D49/IDED12	开路
B11	D51/IDED10	开路
A11	D50/IDED11	开路
C11	D56/IDED6	开路
A10	D52/IDED9	开路
B10	D53/IDED8	开路
D11	D57/IDED7	开路
C10	D54/ $\overline{\text{IDERST}}$	开路
D10	D55/DIRECTION	开路
A7	DM	开路
A6	DP	开路
B6	VBUS	开路
E8	REFRIN	开路

【注】 *1 上电复位后，在 LSI 内部引脚被上拉（初始状态）。必须将 MCU 的 BCR.IPUP 位置 1，使上拉电阻 OFF。但是，因为在此位被置 1 时 $\overline{\text{BREQ}}$ 引脚的上拉电阻也为 OFF，所以在使用 $\overline{\text{BREQ}}$ 引脚时，必须在电路板上进行上拉处理。

*2 在设计能使用仿真器的电路板时，必须遵照仿真器的指示。

*3 在设计不使用仿真器的电路板时，必须固定接地或者连接 $\overline{\text{PRESET}}$ 的相同信号。但是，在设定在固定接地的情况下，因为 $\overline{\text{TRST}}$ 在 LSI 内部被上拉，所以会流过微小的电流。电流值取决于端口引脚上拉电阻的规定。此电流不影响芯片的工作，但是会消耗无用的功率。

*4 未使用引脚的处理是以上电复位后的引脚状态为前提。

附录 G. 型号一览表

产品型号	产品目录的型号	工作温度	焊球成分	PKG 代码	SDHI
R5S77640N300BG	R5S77640N300BG	-20 ~ +85°C	无铅	PRBG0404GA-A	未装载
R5S77640P300BG	R5S77640P300BG	-40 ~ +85°C	无铅	PRBG0404GA-A	未装载
R5S77641N300BG	R5S77641N300BG	-20 ~ +85°C	无铅	PRBG0404GA-A	装载*
R5S77641P300BG	R5S77641P300BG	-40 ~ +85°C	无铅	PRBG0404GA-A	装载*

【注】 * 本产品装载了 SD 主机接口 (SDHI)
有关 SD 主机接口 (SDHI)，需要在缔结了保密协议后才于公开。详细内容请向本公司营业负责人确认。

附录 H. 封装尺寸图

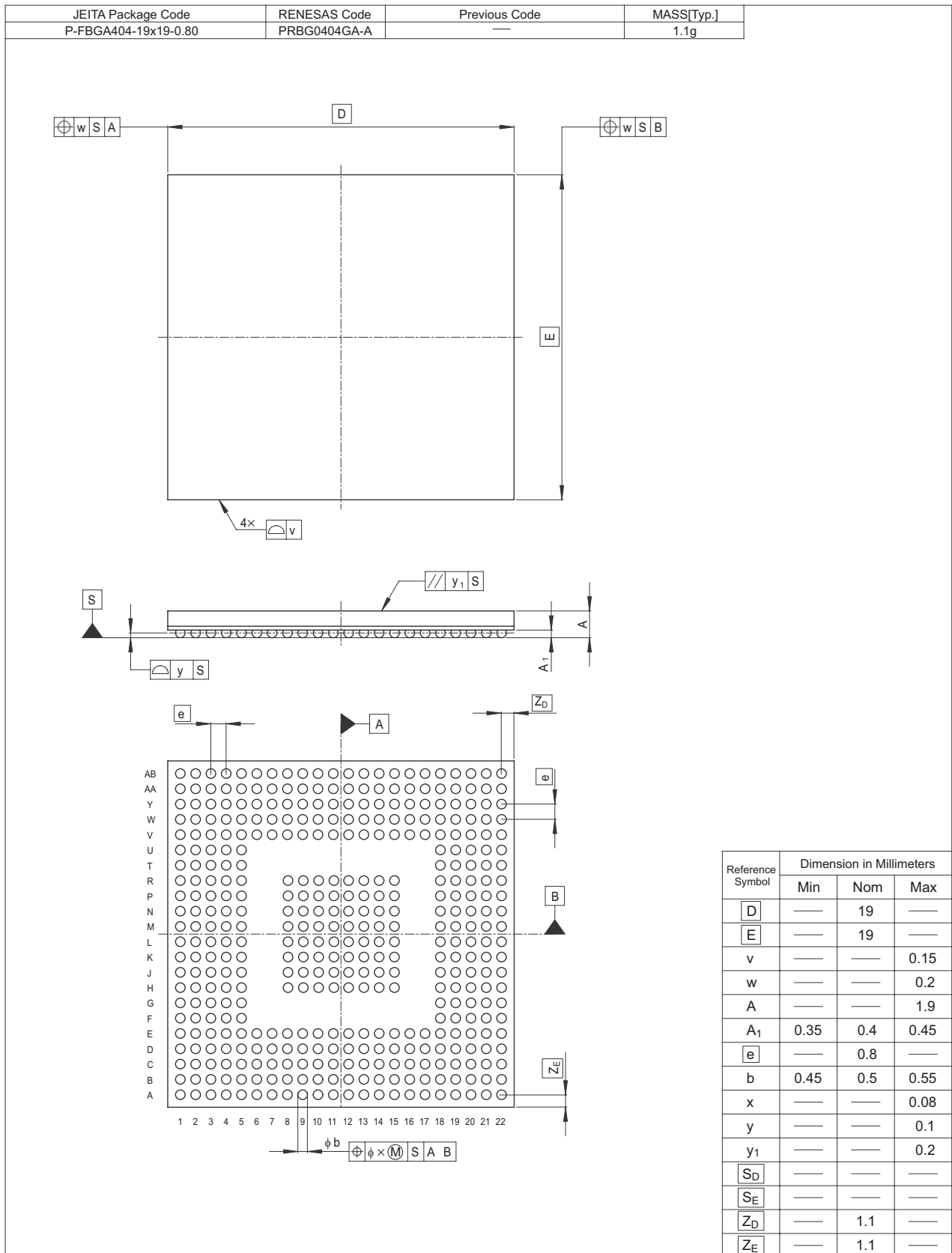


图 H.1 封装尺寸图

索引

数字

10 位地址格式	359
2D 图形引擎 (G2D)	637
4 × 4 矩阵运算	640
7 位地址格式	358

A

ATAPI	367
ATTCH 中断	574

B

BCHG 中断	573
BEMP 中断	569
BRDY 中断	564
被 0 除	90
编程模块	18
编程模型	18
边界扫描 TAP 控制器	899

C

采样速率转换器 (SRC)	801
槽 FPU 禁止异常	78
槽非法指令异常	77
操作数存取周期的断点	888
处理模式	18
初始页写异常	72
粗线的描绘	644
存储器管理单元	94
存储器控制器单元 (MCU)	158
存储器映像寄存器	28
存储器映像 TLB 的结构	123

D

大端法	28
DTCH 中断	573
单精度浮点寄存器	22
单精度浮点扩展寄存器	22
单精度浮点扩展寄存器矩阵	22
单精度浮点向量寄存器	22
单精度数据对的传送	93
单虚拟存储模式	99
等时传送	589
低功耗模式	856
低功耗状态	29
地址转换	98
地址转换方式	109, 114
电源控制顺序的处理	624
定点传送指令	36
定时器单元 (TMU)	293

多缓冲器帧的发送和接收处理	492
多重虚拟存储模式	99
多重中断	289

E

ECC 错误检查	799
ECC 码	799
EOFERR 中断	574
EtherC 发送部	456
EtherC 接收部	457

F

FIFO 缓冲存储器	580
FIFO 内置型串行通信接口 (SCIF)	305
FPU 错误	90
FPU 的相关系统寄存器	19
FPU 异常	79, 90
FPU 异常处理	91
发送描述符 0 (TD0)	487
发送描述符 1 (TD1)	488
发送描述符 2 (TD2)	488
发行速率	56
非规格化数	85
浮点单精度指令	41
浮点单元 (FPU)	83
浮点寄存器	19, 21, 86
浮点控制指令	42
浮点数格式	83
浮点双精度指令	42
符号扩展	28
复位和看门狗定时器 (WDT)	864
复位状态	29

G

高速缓存	129
根据 IPG 设定的运行	462
共用 TLB (UTLB) 的结构	107, 111
管道的安排	596
管道控制	575

H

H-UDI 复位	70
恢复中断	573

I

I ² C 总线接口 (IIC)	346
I ² C 总线数据格式	358
IRQ 中断	283

J

寄存器

ALPHR.....	728	D1FIFOCTR.....	515
AMR.....	173	D1FIFOSEL.....	513
APR.....	452	DBR.....	25
ATAPI_BYTE_SWAP.....	381	DCPCFG.....	535
ATAPI_CONTROL.....	370	DCPCTR.....	536
ATAPI_CONTROL2.....	380	DCPMAXP.....	536
ATAPI_DMA_START_ADR.....	379	DEVADDn (n=0 ~ A).....	560
ATAPI_DMA_TRANS_CNT.....	380	DLSAR.....	723
ATAPI_DTB_ADR.....	377	DSTRR.....	724
ATAPI_INT_ENABLE.....	373	DTCR.....	727
ATAPI_MULTI_TIMING.....	375	DVSTCTR.....	504
ATAPI_PIO_TIMING.....	374	ECMR.....	440
ATAPI_SIG_ST.....	381	ECSIPR.....	443
ATAPI_STATUS.....	372	ECSR.....	442
ATAPI_ULTRA_TIMING.....	376	EDMR.....	467
BCFRR.....	455	EDRRR.....	468
BCR.....	178	EDTRR.....	467
BEMPENB.....	521	EESIPR.....	473
BEMPSTS.....	530	EESR.....	470
BRDYENB.....	519	ENDCVR.....	724
BRDYSTS.....	528	EXPEVT.....	64
BUSWAIT.....	502	EXPMASK.....	65
CAMR0.....	883	FCFTR.....	483
CAMR1.....	883	FDR.....	478
CAR0.....	882	FLADR.....	783
CAR1.....	882	FLADR2.....	784
CBCR.....	886	FLBSYCNT.....	790
CBR0.....	877	FLBSYTMR.....	789
CBR1.....	877	FLCMCDR.....	783
CCMFR.....	886	FLCMDCR.....	781
CCR.....	132	FLCMNCR.....	780
CDCR.....	447	FLDATAR.....	786
CDMR1.....	885	FLDTFIFO.....	791
CDR1.....	884	FLECFIFO.....	792
CEFCR.....	448	FLINTDMACR.....	787
CETR1.....	885	FLTRCR.....	793
CFIFO.....	509	FPSCR.....	25, 88
CFIFOCTR.....	515	FPUL.....	27, 89
CFIFOSEL.....	511	FRECR.....	449
CNDCR.....	448	FRMNUM.....	531
COFSR.....	728	FRQCR.....	155
CPUOPM.....	1019	GBR.....	25
CRR0.....	881	GTRCR.....	734
CRR1.....	881	GTROFSXR.....	737
CS0BCR.....	180	GTROFSYR.....	737
CS3BCR.....	185	ICCCR.....	355
CSnWCR.....	182	ICIDR.....	722
CSTR.....	731	ICMAR.....	355
CURR.....	731	ICMCR.....	351
D0FBCFG.....	508	ICMIER.....	354
D0FIFO.....	509	ICMSR.....	352
D0FIFOCTR.....	515	ICR0.....	263
D0FIFOSEL.....	513	ICR1.....	264
D1FBCFG.....	508	ICRXD.....	356
D1FIFO.....	509	ICSAR.....	351
		ICSCR.....	348
		ICSIER.....	350

ICSSR.....	349	MMUCR.....	102
ICTXD.....	356	MPR.....	452
IER.....	721	MSTPCR0.....	858
INT2A0.....	269	MSTPCR1.....	861
INT2A01.....	271	MTRAR.....	735
INT2A1.....	272	MTRBR.....	735
INT2A11.....	273	MTRCR.....	735
INT2B.....	278	MTRDR.....	736
INT2GPIC.....	281	MTRER.....	736
INT2MSKCR.....	276	MTRFR.....	736
INT2MSKCR1.....	277	MTRGR.....	736
INT2MSKR.....	273	MTRHR.....	737
INT2MSKR1.....	275	MTRIR.....	737
INT2PRI.....	269	NMIFCR.....	267
INTENB0.....	516	NRDYENB.....	520
INTENB1.....	518	NRDYSTS.....	529
INTEVT.....	65	PC.....	25
INTMSK.....	266	PIPEBUF.....	545
INTMSKCLR.....	266	PIPECFG.....	541
INTPRI.....	264	PIPEMAXP.....	547
INTREQ.....	265	PIPEnCTR (n=1 ~ 9).....	549
INTSTS0.....	523	PIPEnTRE (n=1 ~ 5).....	558
INTSTS1.....	525	PIPEnTRN (n=1 ~ 5).....	559
IOSR.....	485	PIPEPERI.....	548
IPGR.....	451	PIPESEL.....	541
IRMCR.....	106	PIR.....	443
LCCR.....	447	PLLCR.....	156
LCOR.....	731	POFSR.....	734
LDACLNR.....	612	PR.....	25
LDCNTR.....	616	PRR.....	1022
LDDFR.....	605	PSR.....	446
LDHCNR.....	608	PTEA.....	104
LDHSYNR.....	609	PTEH.....	100
LDICKR.....	602	PTEL.....	101
LDINTR.....	612	PVR.....	1022
LDLAOR.....	607	QACR0.....	133
LDLIRNR.....	618	QACR1.....	133
LDMTR.....	603	RAMCR.....	134, 148
LDPALCR.....	607	RBWAR.....	481
LDPMMR.....	614	RCL2R.....	732
LDPSPR.....	615	RCLR.....	729
LDSARL.....	606	RDFAR.....	481
LDSARU.....	606	RDLAR.....	469
LDVDLNR.....	609	RDMLR.....	453
LDUINTLNR.....	618	RFCF.....	454
LDUINTR.....	617	RFCR.....	450
LDVSYNR.....	611	RFLR.....	445
LDVTLNR.....	610	RFOCR.....	480
LTADn.....	175	RMCR.....	479
LTAMn.....	176	RMFCR.....	476
LTCn.....	174	RPADIR.....	484
MACH.....	25	RQM.....	177
MACL.....	25	RSAR.....	723
MAFCR.....	451	RTN0R.....	723
MAHR.....	444	RTN1R.....	723
MALR.....	444	RUCLMAR.....	732
MIM.....	166	RUCLMIR.....	732

SCBRR	317	TTB	102
SCEMR	326	VBR	25
SCFCR	321	UCLMAR	732
SCFDR	323	UCLMIR	731
SCFRDR	309	VCR	165
SCFSR	314	VDC2CLKCR	157
SCFTDR	309	WDTBCNT	868
SCLMAR	731	WDTBST	868
SCLR	719	WDTCNT	868
SCLSR	325	WDTCSR	867
SCR	168	WDTST	866
SCRSR	309	UFRMNUM	532
SCSCR	311	WSAR	724
SCSMR	310	USBADDR	532
SCSPTR	323	USBINDX	534
SCTSR	309	USBLENG	535
SDRA	171	USBREQ	533
SGR	25	USBVAL	534
SOFCFG	522	USERIMASK	267
SPC	25	ZCLPMAXR	738
SR	23, 719	ZCLPMINR	738
SRCCTRL	806	ZSATVMINR	738
SRCID	802	几何运算指令	92
SRCIDCTRL	804	加载 / 存储体系结构	31
SRCOD	803	接收描述符 0 (RD0)	489
SRCODCTRL	805	接收描述符 1 (RD1)	490
SRCR	721	接收描述符 2 (RD2)	490
SRCSTAT	808		
SSAR	723		
SSR	25	K	
SSTRR	724	抗锯齿	644
STBCR	858	控制寄存器	19
STCR	727	控制传送阶段转移中断	571
STR	169		
SYSCFG	500		
SYSSTS	503	L	
TBRAR	482	LCDC 模块信号时序	1015
TCNTn	299	LCD 控制器 (LCDC)	598
TCORn	298	流控制	462
TCPR2	301	流水线操作	44
TCRn	299	逻辑运算指令	38
TDFAR	482		
TDLAR	469		
TEA	102	M	
TESTMODE	507	Magic Packet 的检测	461
TFTR	477	MII 寄存器的存取方法	459
TFUCR	480	MII 帧时序	458
TLFRCR	450	命令存取模式	795
TOCR	297	模块待机功能	863
TPAUSECR	454		
TPAUSER	453		
TRA	64	N	
TRIMD	484	NMI 中断	283
TROCR	446	NMI (非屏蔽中断)	80
TRSCER	475	NRDY 中断	567
TSFRCR	449	NYET 握手信号应答表	588
TSTRn	297		

内部存储器	147
内部外围模块的中断	283

P

批量传送	587
------------	-----

Q

取指令周期的断点	888
----------------	-----

S

SACK 中断	573
SCBRR 设定值的计算式	317
SCIF 中断源	343
SIGN 中断	574
SOF 内插功能	595
扇区存取模式	797
上电复位	70, 869
上溢	90
设备状态转移中断	571
舍入	90
事务发行条件	596
时钟同步模式的运行	336
时钟振荡器 (CPG)	151
数据地址错误	74
数据 PID 顺序位	578
数据 TLB 保护违反异常	72
数据 TLB 多重命中异常	70
数据 TLB 未命中异常	71
刷新待机模式	862
双精度浮点寄存器	22
睡眠模式	862
算术运算指令	37

T

TAP 控制	918
TCNT 的计数时序	302
TLB 扩展模式	111
T 位	31
特权模式	18
通用寄存器	18
透视 W 除法运算	642

U

USB2.0 主机 / 功能模块 (USB)	494
USB 数据总线的电阻控制	561

V

VBUS 中断	573
---------------	-----

W

物理地址空间	98
无条件陷阱	76
无效运算	90

X

系统寄存器	19
系统控制指令	40
下溢	90
显示分辨率的设定	624
显示 OFF 模式 (LCDC 停止) 的注意事项	627
向量地址	67
小端法	28
虚拟地址空间	95
选择功能控制器功能时的控制传送	586
选择主机控制器功能时的控制传送	586
寻址方式	32

Y

延迟槽	31
延迟转移	31
液晶模块电源状态	627
一般 FPU 禁止异常	78
一般 FPU 禁止异常 / 槽 FPU 禁止异常	90
一般非法指令异常	76
一般中断请求	80
异步模式的接收数据采样时序和接收容限	344
异步模式的运行	328
异常 / 中断码	67
异常处理	63, 66
异常流程	68
以太网控制器 (EtherC)	436
移位指令	39
应答 PID	577
用户调试接口 (H-UDI)	897
用户断点控制器	874
用户断点运行流程	887
用户模式	19
用于以太网控制器的 直接存储器存取控制器 (E-DMAC)	464
有效地址	32
与 PHY-LSI 的连接	463

Z

Z 裁剪	642
帧更新中断	572
指令地址错误	75
指令高速缓存的通路预测	138
指令 TLB 保护违反异常	73
指令 TLB 多重命中异常	70
指令 TLB 未命中异常	71
指令 TLB (ITLB) 的结构	108, 113
指令系统	31

指令执行后的用户断点	79
指令执行前的用户断点	79
指令执行状态	29
执行状态	56
中断传送	588
中止的检测和处理	344
中止信号的发送	344
转移指令	39
自动重加载计数	302
坐标变换的流程和饱和处理	643

修订记录	SH7764 群硬件手册
------	--------------

Rev.	发行日	修订内容	
		页	修订处
1.00	2008.09.25	—	初版发行

**瑞萨 32 位 RISC 单片机
硬件手册
SH7764 群**

Publication Date: Rev1.00, Sep. 25, 2008
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan



RENESAS SALES OFFICES

<http://www.renesas.com>

Refer to "<http://www.renesas.com/en/network>" for the latest and detailed information.

Renesas Technology America, Inc.

450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.

Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7858/7898

Renesas Technology Hong Kong Ltd.

7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2377-3473

Renesas Technology Taiwan Co., Ltd.

10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

Renesas Technology Singapore Pte. Ltd.

1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.

Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd

Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510



SH7764群



瑞萨电子株式会社

RCJ09B0059-0100